

#|

Copyright (C) 1987, 1994 by William R. Bevier. All Rights Reserved.

You may copy and distribute verbatim copies of this Nqthm-1992 event script as you receive it, in any medium, including embedding it verbatim in derivative works, provided that you conspicuously and appropriately publish on each copy a valid copyright notice "Copyright (C) 1987, 1994 by William R. Bevier. All Rights Reserved."

NO WARRANTY

William R. Bevier PROVIDES ABSOLUTELY NO WARRANTY. THE EVENT SCRIPT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE SCRIPT IS WITH YOU. SHOULD THE SCRIPT PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL William R. Bevier BE LIABLE TO YOU FOR ANY DAMAGES, ANY LOST PROFITS, LOST MONIES, OR OTHER SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THIS SCRIPT (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY THIRD PARTIES), EVEN IF YOU HAVE ADVISED US OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.''

|#

```
;  
; Kit Script for NQTHM-1992  
; Bill Bevier  
; Computational Logic, Inc.  
; bevier@cli.com  
;  
; This script is derived from the original Kit proof script. The  
; original script contains the events entered by the author, and  
; extends over some 62 files. The original runs on a version of NQTHM  
; modified by the author to handle deftheory events and hints, and to  
; handle opening up of constant functions in a certain way.  
;  
; This script was derived by replaying the original in PC-NQTHM, and  
; simultaneously writing out a translation of each event that would  
; allow the script to play in NQTHM-1992. The main differences between  
; the events which appear here and those in the original are:  
;
```

```

; 1. Constant expressions are expanded in place here. The body of no
; definition or lemma is modified from the original other than the
; replacement of a constant expression by its value.
; 2. A few extra deftheory events and enable-theory hints are
; required in this script, primarily for controlling *1* functions.
;

```

EVENT: Start with the initial **nqthm** theory.

THEOREM: plus-right-id2
 $(y \notin \mathbf{N}) \rightarrow ((x + y) = \text{fix}(x))$

EVENT: Disable plus-right-id2.

THEOREM: plus-add1
 $(x + (1 + y))$
 $= \text{if } y \in \mathbf{N} \text{ then } 1 + (x + y)$
 $\text{else } 1 + x \text{ endif}$

EVENT: Disable plus-add1.

THEOREM: commutativity2-of-plus
 $(x + (y + z)) = (y + (x + z))$

EVENT: Disable commutativity2-of-plus.

THEOREM: commutativity-of-plus
 $(x + y) = (y + x)$

EVENT: Disable commutativity-of-plus.

THEOREM: associativity-of-plus
 $((x + y) + z) = (x + (y + z))$

EVENT: Disable associativity-of-plus.

THEOREM: plus-equal-0
 $((a + b) = '0) = ((a \simeq 0) \wedge (b \simeq 0))$

EVENT: Disable plus-equal-0.

THEOREM: difference-x-x

$$(x - x) = '0$$

EVENT: Disable difference-x-x.

THEOREM: difference-plus

$$(((x + y) - x) = \text{fix}(y)) \wedge (((y + x) - x) = \text{fix}(y))$$

EVENT: Disable difference-plus.

THEOREM: plus-cancellation

$$((a + b) = (a + c)) = (\text{fix}(b) = \text{fix}(c))$$

EVENT: Disable plus-cancellation.

THEOREM: difference-0

$$(y \not\prec x) \rightarrow ((x - y) = '0)$$

EVENT: Disable difference-0.

THEOREM: equal-difference-0

$EVENT: Disable equal-difference-0.$

THEOREM: difference-cancellation-0

$$(x = (x - y)) = ((x \in \mathbf{N}) \wedge ((x = '0) \vee (y \simeq 0)))$$

EVENT: Disable difference-cancellation-0.

THEOREM: difference-cancellation-1

$$\begin{aligned} & ((x - y) = (z - y)) \\ & = \text{if } x < y \text{ then } y \not\prec z \\ & \quad \text{elseif } z < y \text{ then } y \not\prec x \\ & \quad \text{else } \text{fix}(x) = \text{fix}(z) \text{ endif} \end{aligned}$$

EVENT: Disable difference-cancellation-1.

THEOREM: times-zero2

$$(y \notin \mathbf{N}) \rightarrow ((x * y) = '0)$$

EVENT: Disable times-zero2.

THEOREM: distributivity-of-times-over-plus

$$(x * (y + z)) = ((x * y) + (x * z))$$

EVENT: Disable distributivity-of-times-over-plus.

THEOREM: times-add1

$$\begin{aligned} &(x * (1 + y)) \\ = &\text{ if } y \in \mathbf{N} \text{ then } x + (x * y) \\ &\text{ else fix } (x) \text{ endif} \end{aligned}$$

EVENT: Disable times-add1.

THEOREM: commutativity-of-times

$$(x * y) = (y * x)$$

EVENT: Disable commutativity-of-times.

THEOREM: commutativity2-of-times

$$(x * (y * z)) = (y * (x * z))$$

EVENT: Disable commutativity2-of-times.

THEOREM: associativity-of-times

$$((x * y) * z) = (x * (y * z))$$

EVENT: Disable associativity-of-times.

THEOREM: equal-times-0

$$((x * y) = '0) = ((x \simeq 0) \vee (y \simeq 0))$$

EVENT: Disable equal-times-0.

DEFINITION:

$$\begin{aligned} &\text{exp}(i, j) \\ = &\text{ if } j \simeq 0 \text{ then } '1 \\ &\text{ else } i * \text{exp}(i, j - 1) \text{ endif} \end{aligned}$$

THEOREM: exp-plus

$$\text{exp}(i, j + k) = (\text{exp}(i, j) * \text{exp}(i, k))$$

EVENT: Disable exp-plus.

THEOREM: equal-lessp

$$\begin{aligned} & ((x < y) = z) \\ & = \text{if } x < y \text{ then } '1*\text{true} = z \\ & \quad \text{else } '1*\text{false} = z \text{ endif} \end{aligned}$$

EVENT: Disable equal-lessp.

THEOREM: difference-elim

$$((y \in \mathbf{N}) \wedge (y \not< x)) \rightarrow ((x + (y - x)) = y)$$

THEOREM: remainder-quotient

$$((x \bmod y) + (y * (x \div y))) = \text{fix}(x)$$

EVENT: Disable remainder-quotient.

THEOREM: remainder-wrt-1

$$(y \bmod '1) = '0$$

EVENT: Disable remainder-wrt-1.

THEOREM: remainder-wrt-12

$$(x \notin \mathbf{N}) \rightarrow ((y \bmod x) = \text{fix}(y))$$

EVENT: Disable remainder-wrt-12.

THEOREM: lessp-remainder2

$$((x \bmod y) < y) = (y \not< 0)$$

EVENT: Disable lessp-remainder2.

THEOREM: remainder-x-x

$$(x \bmod x) = '0$$

EVENT: Disable remainder-x-x.

THEOREM: remainder-quotient-elim

$$((y \not< 0) \wedge (x \in \mathbf{N})) \rightarrow (((x \bmod y) + (y * (x \div y))) = x)$$

THEOREM: lessp-times-1

$$(i \not< 0) \rightarrow ((i * j) \not< j)$$

EVENT: Disable lessp-times-1.

THEOREM: lessp-times-2

$$(i \neq 0) \rightarrow ((j * i) \neq j)$$

EVENT: Disable lessp-times-2.

THEOREM: lessp-quotient1

$$((i \div j) < i) = ((i \neq 0) \wedge ((j \simeq 0) \vee (j \neq '1)))$$

EVENT: Disable lessp-quotient1.

THEOREM: lessp-remainder1

$$((x \mathbf{mod} y) < x) = ((y \neq 0) \wedge ((x \neq 0) \wedge (x \neq y)))$$

EVENT: Disable lessp-remainder1.

THEOREM: difference-plus1

$$((x + y) - x) = \text{fix}(y)$$

EVENT: Disable difference-plus1.

THEOREM: difference-plus2

$$((y + x) - x) = \text{fix}(y)$$

EVENT: Disable difference-plus2.

THEOREM: difference-plus-cancelation

$$((x + y) - (x + z)) = (y - z)$$

EVENT: Disable difference-plus-cancelation.

THEOREM: times-difference

$$(x * (c - w)) = ((c * x) - (w * x))$$

EVENT: Disable times-difference.

DEFINITION: divides(x, y) = $((y \mathbf{mod} x) \simeq 0)$

THEOREM: divides-times

$$((x * z) \mathbf{mod} z) = '0$$

EVENT: Disable divides-times.

THEOREM: difference-plus3

$$((b + (a + c)) - a) = (b + c)$$

EVENT: Disable difference-plus3.

THEOREM: difference-add1-cancellation

$$((1 + (y + z)) - z) = (1 + y)$$

EVENT: Disable difference-add1-cancellation.

THEOREM: remainder-add1

$$((y \neq 0) \wedge (y \neq '1)) \rightarrow (((1 + (x * y)) \mathbf{mod} y) \neq '0)$$

EVENT: Disable remainder-add1.

THEOREM: divides-plus-rewrite1

$$\begin{aligned} &(((x \mathbf{mod} z) = '0) \wedge ((y \mathbf{mod} z) = '0)) \\ \rightarrow &(((x + y) \mathbf{mod} z) = '0) \end{aligned}$$

EVENT: Disable divides-plus-rewrite1.

THEOREM: divides-plus-rewrite2

$$\begin{aligned} &(((x \mathbf{mod} z) = '0) \wedge ((y \mathbf{mod} z) \neq '0)) \\ \rightarrow &(((x + y) \mathbf{mod} z) \neq '0) \end{aligned}$$

EVENT: Disable divides-plus-rewrite2.

THEOREM: divides-plus-rewrite

$$\begin{aligned} &((x \mathbf{mod} z) = '0) \\ \rightarrow &(((x + y) \mathbf{mod} z) = '0) = ((y \mathbf{mod} z) = '0) \end{aligned}$$

EVENT: Disable divides-plus-rewrite.

THEOREM: lessp-plus-cancelation

$$((x + y) < (x + z)) = (y < z)$$

EVENT: Disable lessp-plus-cancelation.

THEOREM: divides-plus-rewrite-commuted

$$\begin{aligned} &((x \mathbf{mod} z) = '0) \\ \rightarrow &(((y + x) \mathbf{mod} z) = '0) = ((y \mathbf{mod} z) = '0) \end{aligned}$$

EVENT: Disable divides-plus-rewrite-commuted.

THEOREM: euclid

$$\begin{aligned} & ((x \bmod z) = '0) \\ \rightarrow & (((y - x) \bmod z) = '0) \\ & = \text{if } x < y \text{ then } (y \bmod z) = '0 \\ & \quad \text{else } '*1*true \text{ endif} \end{aligned}$$

EVENT: Disable euclid.

THEOREM: lessp-times-cancellation

$$((x * z) < (y * z)) = ((z \neq 0) \wedge (x < y))$$

EVENT: Disable lessp-times-cancellation.

THEOREM: lessp-plus-cancellation3

$$(y < (x + y)) = (x \neq 0)$$

EVENT: Disable lessp-plus-cancellation3.

THEOREM: quotient-times1

$$\begin{aligned} & ((y \in \mathbf{N}) \wedge ((x \in \mathbf{N}) \wedge ((x \neq '0) \wedge \text{divides}(x, y)))) \\ \rightarrow & ((x * (y \div x)) = y) \end{aligned}$$

EVENT: Disable quotient-times1.

THEOREM: quotient-lessp

$$((x \neq 0) \wedge (x < y)) \rightarrow ((y \div x) \neq '0)$$

EVENT: Disable quotient-lessp.

THEOREM: times-id-iff-1

$$(z = (w * z)) = ((z \in \mathbf{N}) \wedge ((z = '0) \vee (w = '1)))$$

EVENT: Disable times-id-iff-1.

THEOREM: divides-times1

$$(a = (z * y)) \rightarrow ((a \bmod z) = '0)$$

EVENT: Disable divides-times1.

THEOREM: times-identity1

$$\begin{aligned} & ((y \in \mathbf{N}) \wedge ((y \neq '1) \wedge ((y \neq '0) \wedge (x \neq '0)))) \\ & \rightarrow (x \neq (x * y)) \end{aligned}$$

EVENT: Disable times-identity1.

THEOREM: times-identity

$$(x = (x * y)) = ((x = '0) \vee ((x \in \mathbf{N}) \wedge (y = '1)))$$

EVENT: Disable times-identity.

THEOREM: quotient-divides

$$((y \in \mathbf{N}) \wedge ((x * (y \div x)) \neq y)) \rightarrow ((y \bmod x) \neq '0)$$

EVENT: Disable quotient-divides.

THEOREM: remainder-times

$$((y * x) \bmod y) = '0$$

EVENT: Disable remainder-times.

THEOREM: quotient-times

$$\begin{aligned} & ((y * x) \div y) \\ & = \text{if } y \simeq 0 \text{ then } '0 \\ & \quad \text{else fix } (x) \text{ endif} \end{aligned}$$

EVENT: Disable quotient-times.

THEOREM: distributivity-of-divides

$$((a \neq 0) \wedge \text{divides}(a, w)) \rightarrow ((c * (w \div a)) = ((c * w) \div a))$$

EVENT: Disable distributivity-of-divides.

THEOREM: if-times-then-divides

$$((c \neq 0) \wedge (\neg \text{divides}(c, x))) \rightarrow ((c * y) \neq x)$$

EVENT: Disable if-times-then-divides.

THEOREM: times-equal-1

$$\begin{aligned} & ((a * b) = '1) \\ & = ((a \neq '0) \\ & \quad \wedge ((b \neq '0) \end{aligned}$$

$$\begin{aligned} &\wedge ((a \in \mathbf{N}) \\ &\quad \wedge ((b \in \mathbf{N}) \\ &\quad \quad \wedge (((a - 1) = '0) \\ &\quad \quad \quad \wedge ((b - 1) = '0)))))) \end{aligned}$$

EVENT: Disable times-equal-1.

THEOREM: divides-implies-times
 $((a \neq 0) \wedge ((c \in \mathbf{N}) \wedge ((a * c) = b)))$
 $\rightarrow ((c = (b \div a)) = '1 * true)$

EVENT: Disable divides-implies-times.

THEOREM: difference-1
 $(x - '1) = (x - 1)$

EVENT: Disable difference-1.

THEOREM: difference-2
 $((1 + (1 + x)) - '2) = \text{fix}(x)$

EVENT: Disable difference-2.

THEOREM: half-plus
 $((x + (x + y)) \div '2) = (x + (y \div '2))$

EVENT: Disable half-plus.

THEOREM: times-1
 $('1 * x) = \text{fix}(x)$

EVENT: Disable times-1.

THEOREM: exp-of-0
 $\text{exp}('0, k)$
 $= \text{if } k \simeq 0 \text{ then } '1$
 $\quad \text{else } '0 \text{ endif}$

EVENT: Disable exp-of-0.

THEOREM: exp-of-1
 $\text{exp}('1, k) = '1$

EVENT: Disable exp-of-1.

THEOREM: exp-by-0
 $\text{exp}(x, '0) = '1$

EVENT: Disable exp-by-0.

THEOREM: exp-times
 $\text{exp}(i * j, k) = (\text{exp}(i, k) * \text{exp}(j, k))$

EVENT: Disable exp-times.

THEOREM: exp-exp
 $\text{exp}(\text{exp}(i, j), k) = \text{exp}(i, j * k)$

EVENT: Disable exp-exp.

THEOREM: remainder-plus-times-1
 $((x + (i * j)) \bmod j) = (x \bmod j)$

EVENT: Disable remainder-plus-times-1.

THEOREM: remainder-plus-times-2
 $((x + (j * i)) \bmod j) = (x \bmod j)$

EVENT: Disable remainder-plus-times-2.

THEOREM: remainder-times-1
 $((b * (a * c)) \bmod a) = '0$

EVENT: Disable remainder-times-1.

THEOREM: remainder-of-1
 $('1 \bmod x)$
 $= \text{if } x = '1 \text{ then } '0$
 $\text{else } '1 \text{ endif}$

EVENT: Disable remainder-of-1.

THEOREM: remainder-difference-times
 $((p * x) - (p * y)) \bmod p = '0$

EVENT: Disable remainder-difference-times.

THEOREM: lessp-remainder-divisor
 $(y \neq 0) \rightarrow ((x \bmod y) < y)$

EVENT: Disable lessp-remainder-divisor.

EVENT: Let us define the theory *cl-arithmetic* to consist of the following events: lessp-remainder-divisor, remainder-difference-times, remainder-of-1, remainder-times-1, remainder-plus-times-2, remainder-plus-times-1, exp-exp, exp-times, exp-by-0, exp-of-1, exp-of-0, times-1, half-plus, difference-2, difference-1, divides-implies-times, times-equal-1, if-times-then-divides, distributivity-of-divides, quotient-times, remainder-times, quotient-divides, times-identity, times-identity1, divides-times1, times-id-iff-1, quotient-lessp, quotient-times1, lessp-plus-cancellation3, lessp-times-cancellation, euclid, divides-plus-rewrite-commuted, lessp-plus-cancellation, divides-plus-rewrite, divides-plus-rewrite2, divides-plus-rewrite1, remainder-add1, difference-add1-cancellation, difference-plus3, divides-times, divides, times-difference, difference-plus-cancellation, difference-plus2, difference-plus1, lessp-remainder1, lessp-quotient1, lessp-times-2, lessp-times-1, remainder-quotient-elim, remainder-x-x, lessp-remainder2, remainder-wrt-12, remainder-wrt-1, remainder-quotient, difference-elim, equal-lessp, exp-plus, equal-times-0, associativity-of-times, commutativity2-of-times, commutativity-of-times, times-add1, distributivity-of-times-over-plus, times-zero2, difference-cancellation-1, difference-cancellation-0, equal-difference-0, difference-0, plus-cancellation, difference-plus, difference-x-x, plus-equal-0, associativity-of-plus, commutativity-of-plus, commutativity2-of-plus, plus-add1, plus-right-id2.

DEFINITION:
number-and-list-induction (n, l)
= **if** $n \simeq 0$ **then** '0
 elseif listp (l) **then** number-and-list-induction ($n - 1, \text{cdr } (l)$)
 else '0 **endif**

DEFINITION:
double-number-induction (x, y)
= **if** $(x \simeq 0) \vee (y \simeq 0)$ **then** '0
 else double-number-induction ($x - 1, y - 1$) **endif**

DEFINITION:
double-number-double-list-induction ($i, j, l1, l2$)
= **if** listp ($l1$) \wedge listp ($l2$)
 then if $(i \simeq 0) \vee (j \simeq 0)$ **then** '0
 else double-number-double-list-induction ($i - 1,$

$j - 1,$
 $\text{cdr}(l1),$
 $\text{cdr}(l2)) \text{ endif}$

else '0 endif

THEOREM: plus-0
 $(a + '0) = \text{fix}(a)$

EVENT: Disable plus-0.

THEOREM: plus-1
 $('1 + x) = (1 + x)$

EVENT: Disable plus-1.

THEOREM: plus-0-arg1
 $('0 + x) = \text{fix}(x)$

EVENT: Disable plus-0-arg1.

THEOREM: plus-difference-cancellation
if $b < a$ **then** $'*1*false$
else $'*1*true$ **endif**
 $\rightarrow ((a + (b - a)) = \text{fix}(b))$

EVENT: Disable plus-difference-cancellation.

THEOREM: canonicalize-plus-terms1
 $(1 + ((1 + a) + b)) = ((1 + (1 + a)) + b)$

EVENT: Disable canonicalize-plus-terms1.

THEOREM: canonicalize-plus-terms2
 $((1 + a) + ((1 + b) + c)) = (((1 + a) + (1 + b)) + c)$

EVENT: Disable canonicalize-plus-terms2.

THEOREM: canonicalize-plus-terms3
 $((a * b) + ((1 + c) + d)) = ((1 + c) + (d + (a * b)))$

EVENT: Disable canonicalize-plus-terms3.

THEOREM: plus-plus-difference

$$\begin{aligned} & ((i \in \mathbf{N}) \\ & \wedge ((n \in \mathbf{N}) \\ & \quad \wedge \text{if } n < i \text{ then } \text{'*1*false} \\ & \quad \quad \text{else } \text{'*1*true endif})) \\ \rightarrow & ((i + (j + (n - i))) = (n + j)) \end{aligned}$$

EVENT: Disable plus-plus-difference.

THEOREM: plus-times-reduction

$$(a + (b * a)) = ((1 + b) * a)$$

EVENT: Disable plus-times-reduction.

THEOREM: plus-times-sub1-reduction

$$(b \neq 0) \rightarrow ((a + ((b - 1) * a)) = (b * a))$$

EVENT: Disable plus-times-sub1-reduction.

THEOREM: plus-commutativity-associativity-crock

$$(b + (a + c)) = (a + (b + c))$$

EVENT: Disable plus-commutativity-associativity-crock.

THEOREM: distributivity-of-times-over-plus-backwards

$$((a * b) + (a * c)) = (a * (b + c))$$

EVENT: Disable distributivity-of-times-over-plus-backwards.

THEOREM: difference-is-zero

$$(a < b) \rightarrow ((a - b) = '0)$$

EVENT: Disable difference-is-zero.

THEOREM: difference-by-larger-number

$$((x - 1) - x) = '0$$

EVENT: Disable difference-by-larger-number.

THEOREM: difference-add1-arg1

$$\begin{aligned} & \text{if } a < b \text{ then } \text{'*1*false} \\ & \text{else } \text{'*1*true endif} \\ \rightarrow & (((1 + a) - b) = (1 + (a - b))) \end{aligned}$$

EVENT: Disable difference-add1-arg1.

THEOREM: difference-add1-arg2
 $(b < a) \rightarrow ((a - (1 + b)) = ((a - b) - 1))$

EVENT: Disable difference-add1-arg2.

THEOREM: sub1-difference
 $((a - b) - 1) = ((a - 1) - b)$

EVENT: Disable sub1-difference.

THEOREM: difference-sub1-arg1
 $(b < a) \rightarrow (((a - 1) - b) = ((a - b) - 1))$

EVENT: Disable difference-sub1-arg1.

THEOREM: difference-sub1-arg2
 $((b \neq 0)$
 \wedge **if** $a < b$ **then** $'*1*false$
 else $'*1>true$ **endif**)
 $\rightarrow ((a - (b - 1)) = (1 + (a - b)))$

EVENT: Disable difference-sub1-arg2.

THEOREM: difference-plus-cancellation2-instance
if $a < c$ **then** $'*1*false$
else $'*1>true$ **endif**
 $\rightarrow (((1 + a) + b) - (1 + c)) = ((a - c) + b)$

EVENT: Disable difference-plus-cancellation2-instance.

THEOREM: difference-difference
 $((a - b) - c) = (a - (b + c))$

EVENT: Disable difference-difference.

THEOREM: yet-another-difference-plus-crock
 $((n + (b + a)) - a) = (n + b)$

EVENT: Disable yet-another-difference-plus-crock.

THEOREM: non-zero-difference
 $((a \in \mathbf{N}) \wedge (a < b)) \rightarrow ((b - a) \neq '0)$

EVENT: Disable non-zero-difference.

THEOREM: rewrite-non-zero-difference-as-lessp
 $((a - b) \neq '0) = (b < a)$

EVENT: Disable rewrite-non-zero-difference-as-lessp.

THEOREM: rewrite-zero-difference-as-equality
if $a < b$ **then** **'*1*false**
else **'*1>true** **endif**
 $\rightarrow (((a - b) = '0) = (\text{fix}(a) = \text{fix}(b)))$

EVENT: Disable rewrite-zero-difference-as-equality.

THEOREM: regroup-fact
(if $a < c$ **then** **'*1*false**
else **'*1>true** **endif**
 \wedge **if** $b < d$ **then** **'*1*false**
else **'*1>true** **endif)**
 $\rightarrow (((a + b) - (c + d)) = ((a - c) + (b - d)))$

EVENT: Disable regroup-fact.

THEOREM: difference-equals-0
 $((a \in \mathbf{N})$
 $\wedge ((b \in \mathbf{N})$
 \wedge **if** $a < b$ **then** **'*1*false**
else **'*1>true** **endif))**
 $\rightarrow (((a - b) = '0) = (a = b))$

EVENT: Disable difference-equals-0.

THEOREM: times-zero3
 $(x * '0) = '0$

EVENT: Disable times-zero3.

THEOREM: distributivity-of-times-over-difference
 $((a * b) - (a * c)) = (a * (b - c))$

EVENT: Disable distributivity-of-times-over-difference.

THEOREM: exp-2-never-0
 $(\text{'0} < \exp(\text{'2}, i)) = \text{'1*true}$

EVENT: Disable exp-2-never-0.

THEOREM: exp-2-never-0-linear
 $\text{'0} < \exp(\text{'2}, i)$

EVENT: Disable exp-2-never-0-linear.

THEOREM: remainder-noop
 $((a \in \mathbf{N}) \wedge ((a < b) \wedge (b \neq 0))) \rightarrow ((a \mathbf{mod} b) = a)$

EVENT: Disable remainder-noop.

THEOREM: remainder-add1-casesplit
 $(a < b)$
 $\rightarrow (((1 + a) \mathbf{mod} b)$
 $= \mathbf{if} (1 + a) = b \mathbf{then} \text{'0}$
 $\mathbf{else} 1 + a \mathbf{endif})$

EVENT: Disable remainder-add1-casesplit.

THEOREM: zerop-remainder-difference
 $((a \mathbf{mod} b) = \text{'0}) \rightarrow (((a - b) \mathbf{mod} b) = \text{'0})$

EVENT: Disable zerop-remainder-difference.

THEOREM: quotient-x-x
 $(x \neq 0) \rightarrow ((x \div x) = \text{'1})$

EVENT: Disable quotient-x-x.

THEOREM: remainder-times-other-way
 $((b * a) \mathbf{mod} a) = \text{'0}$

EVENT: Disable remainder-times-other-way.

THEOREM: quotient-times-other-way

$((b * a) \div a)$
 = **if** $a \simeq 0$ **then** '0
 else fix(b) **endif**

EVENT: Disable quotient-times-other-way.

THEOREM: quotient-difference
 $(i \not\leq j) \rightarrow (((i - j) \div j) = ((i \div j) - 1))$

EVENT: Disable quotient-difference.

THEOREM: quotient-plus
 $(a \not\leq 0) \rightarrow (((a + b) \div a) = (1 + (b \div a)))$

EVENT: Disable quotient-plus.

THEOREM: quotient-plus-times
 $(c < b) \rightarrow (((c + (b * a)) \div b) = \text{fix}(a))$

EVENT: Disable quotient-plus-times.

THEOREM: leq-times
 $((a \not\leq 0)$
 \wedge **if** $c < b$ **then** '*1*false
 else '*1*true **endif**)
 \rightarrow **if** $(a * c) < (a * b)$ **then** '*1*false
 else '*1*true **endif**

THEOREM: lessp-remainder-sub1-crock
 $(x < n) \rightarrow ((x - 1) < n)$

THEOREM: leq-plus
if $c < (a + b)$ **then** '*1*false
 else '*1*true **endif**
 \rightarrow **if** $c < a$ **then** '*1*false
 else '*1*true **endif**

THEOREM: leq-plus1
if $c < (a + b)$ **then** '*1*false
 else '*1*true **endif**
 \rightarrow (**if** $c < a$ **then** '*1*false
 else '*1*true **endif**
 \wedge **if** $c < b$ **then** '*1*false
 else '*1*true **endif**)

THEOREM: leq-plus2
if $c < (a + b)$ **then** `'*1*false`
else `'*1>true` **endif**
 $\rightarrow ((c < a) = \text{'*1*false})$

THEOREM: leq-casesplit
 $((a \in \mathbf{N}) \wedge (b \in \mathbf{N}))$
 $\rightarrow (\text{if } b < a \text{ then } \text{'*1*false}$
 else `'*1>true` **endif**
 $= ((a < b) \vee (a = b)))$

THEOREM: integer-inequality-casesplit
 $((i \in \mathbf{N}) \wedge (j \in \mathbf{N})) \rightarrow ((i \neq j) = ((i < j) \vee (j < i)))$

THEOREM: integer-equality-crock
 $((a \in \mathbf{N}) \wedge ((b \in \mathbf{N}) \wedge ((a < '1) \wedge (b < '1)))) \rightarrow (a = b)$

THEOREM: lessp-remainder-remainder
 $((c \neq 0) \wedge ((b \bmod c) \neq 0))$
 $\rightarrow (((a \bmod (b \bmod c)) < c) = \text{'*1>true})$

EVENT: Disable lessp-remainder-remainder.

THEOREM: lessp-times
 $((a \neq 0) \wedge (b < c)) \rightarrow ((a * b) < (a * c))$

EVENT: Disable lessp-times.

THEOREM: leq-plus-times
 $((b < c) \wedge (a \neq 0))$
 $\rightarrow (\text{if } (a * c) < (a + (a * b)) \text{ then } \text{'*1*false}$
 else `'*1>true` **endif**

THEOREM: lessp-difference
 $((a \neq 0) \wedge (a < b)) \rightarrow ((b - a) < b)$

THEOREM: remainder-difference-non-zero
 $((a \neq 0) \wedge ((b \in \mathbf{N}) \wedge (a < b))) \rightarrow (((b - a) \bmod b) \neq 0)$

EVENT: Disable remainder-difference-non-zero.

THEOREM: sum-zero-implies-addends-zero
 $((a + b) = '0) \rightarrow ((a \simeq 0) \wedge (b \simeq 0))$

THEOREM: remainder-zero-implies-leq-fact
 $((a \bmod b) = '0) \wedge (a \neq 0) \rightarrow ((a < b) = '*1*false)$

EVENT: Disable remainder-zero-implies-leq-fact.

DEFINITION:
lessp-quotient-induction (a, b, c)
= **if** $b \simeq 0$ **then** '0
 elseif $c \simeq 0$ **then** '0
 elseif $a < c$ **then** '0
 else lessp-quotient-induction $(a - c, b - 1, c)$ **endif**

THEOREM: lessp-quotient-remainder-crock
 $((a < (b * c))$
 $\wedge ((b \neq '0)$
 $\wedge ((b \in \mathbf{N}) \wedge ((c \neq '0) \wedge ((c \in \mathbf{N}) \wedge (a \in \mathbf{N}))))))$
 $\rightarrow ((a \div c) < b)$

EVENT: Disable lessp-quotient-remainder-crock.

THEOREM: lessp-quotient-remainder
 $((b \neq 0) \wedge (c \neq 0)) \rightarrow (((a \bmod (b * c)) \div c) < \text{fix}(b))$

EVENT: Disable lessp-quotient-remainder.

THEOREM: lessp-quotient-times
 $(a < (b * c)) \rightarrow (((a \div c) < b) = '*1*true)$

EVENT: Disable lessp-quotient-times.

DEFINITION:
 $\min(a, b)$
= **if** $a < b$ **then** a
 else b **endif**

THEOREM: lessp-transitivity
 $((a < b) \wedge (b < c)) \rightarrow (a < c)$

THEOREM: remainder-plus
 $((a + n) \bmod a) = (n \bmod a)$

EVENT: Disable remainder-plus.

THEOREM: remainder-divides
 $(\text{divides}(a, u) \wedge (u \in \mathbf{N})) \rightarrow (((u + n) \bmod a) = (n \bmod a))$

EVENT: Disable remainder-divides.

THEOREM: remainder-remainder
 $\text{divides}(a, b) \rightarrow (((n \bmod b) \bmod a) = (n \bmod a))$

EVENT: Disable remainder-remainder.

THEOREM: lessp-quotient
 $(n < (a * b)) \rightarrow ((n \div a) < b)$

DEFINITION:
lessp-quotient-quotient-induction (n, a, b)
= **if** $n \simeq 0$ **then** '0
 elseif $a < n$ **then** '0
 elseif $b < n$ **then** '0
 else lessp-quotient-quotient-induction $(n, a - n, b - n)$ **endif**

THEOREM: lessp-quotient-quotient
 $((n \neq 0) \wedge (((b \bmod n) = '0) \wedge (a < b)))$
 $\rightarrow (((a \div n) < (b \div n)) = '*1*true)$

EVENT: Disable lessp-quotient-quotient.

THEOREM: lessp-times-quotient
 $((n \neq 0) \wedge ((i \in \mathbf{N}) \wedge ((i < j) \wedge ((j \bmod n) = '0))))$
 $\rightarrow (((n * (i \div n)) < j) = '*1*true)$

EVENT: Disable lessp-times-quotient.

THEOREM: not-lessp-times-plus
 $((a \in \mathbf{N})$
 $\wedge ((v \in \mathbf{N})$
 $\wedge ((v < n)$
 $\wedge ((b \in \mathbf{N})$
 $\wedge ((n \neq '0)$
 $\wedge ((n \in \mathbf{N})$
 $\wedge ((v + (a * n))$
 $< (b * n))))))))$
 $\rightarrow ((b * n) \not< ((a * n) + n))$

THEOREM: not-lessp-times-plus-instance

$$\begin{aligned} & ((x \in \mathbf{N}) \\ & \wedge ((v \in \mathbf{N}) \\ & \quad \wedge ((v < n) \\ & \quad \quad \wedge ((z \in \mathbf{N}) \\ & \quad \quad \quad \wedge ((n \neq '0) \\ & \quad \quad \quad \quad \wedge ((n \in \mathbf{N}) \\ & \quad \quad \quad \quad \quad \wedge ((n * z) \\ & \quad \quad \quad \quad \quad \quad < (n + (n * x)))))))))) \\ \rightarrow & (((v + (n * x)) < (n * z)) = '*1*false) \end{aligned}$$

EVENT: Disable not-lessp-times-plus-instance.

THEOREM: not-lessp-plus-times-quotient

$$\begin{aligned} & ((n \neq 0) \wedge ((i \in \mathbf{N}) \wedge ((i < j) \wedge ((j \bmod n) = '0)))) \\ \rightarrow & ((j < ((n * (i \div n)) + n)) = '*1*false) \end{aligned}$$

EVENT: Disable not-lessp-plus-times-quotient.

THEOREM: quotient-plus-times-remainder

$$(n \neq 0) \rightarrow (((n * a) + (b \bmod n)) \div n) = \text{fix}(a)$$

EVENT: Disable quotient-plus-times-remainder.

THEOREM: remainder-plus-times-remainder

$$(((n * a) + (b \bmod n)) \bmod n) = (b \bmod n)$$

EVENT: Disable remainder-plus-times-remainder.

THEOREM: remainder-non-zero

$$((n \neq 0) \wedge (n < m)) \rightarrow ('0 < (n \bmod m))$$

THEOREM: plus-not-equal

$$(\text{fix}(b) \neq \text{fix}(c)) \rightarrow ((a + b) \neq (a + c))$$

EVENT: Disable plus-not-equal.

THEOREM: difference-plus4

$$((a + (b + c)) - (b + c)) = \text{fix}(a)$$

EVENT: Disable difference-plus4.

THEOREM: remainder-remainder-inverse
 $\text{divides}(a, b) \rightarrow ((n \bmod a) = ((n \bmod b) \bmod a))$

THEOREM: associativity-of-plus-inverse
 $(a + (b + c)) = ((a + b) + c)$

DEFINITION:
 $\text{length}(l)$
= **if** listp(l) **then** $1 + \text{length}(\text{cdr}(l))$
 else '0 **endif**

DEFINITION:
 $\text{insert}(i, l)$
= **if** listp(l)
 then if $i < \text{car}(l)$ **then** $\text{cons}(i, l)$
 else $\text{cons}(\text{car}(l), \text{insert}(i, \text{cdr}(l)))$ **endif**
 else $\text{cons}(i, l)$ **endif**

DEFINITION:
 $\text{occurrences}(x, l)$
= **if** listp(l)
 then if $x = \text{car}(l)$ **then** $1 + \text{occurrences}(x, \text{cdr}(l))$
 else $\text{occurrences}(x, \text{cdr}(l))$ **endif**
 else '0 **endif**

DEFINITION:
 $\text{remove}(x, l)$
= **if** listp(l)
 then if $x = \text{car}(l)$ **then** $\text{cdr}(l)$
 else $\text{cons}(\text{car}(l), \text{remove}(x, \text{cdr}(l)))$ **endif**
 else l **endif**

DEFINITION:
 $\text{permutation}(a, b)$
= **if** listp(a)
 then $(\text{car}(a) \in b) \wedge \text{permutation}(\text{cdr}(a), \text{remove}(\text{car}(a), b))$
 else $b \simeq \text{nil}$ **endif**

DEFINITION:
 $\text{plistp}(l)$
= **if** listp(l) **then** $\text{plistp}(\text{cdr}(l))$
 else $l = \text{'nil}$ **endif**

DEFINITION:
 $\text{setp}(l)$
= **if** listp(l) **then** $(\text{car}(l) \notin \text{cdr}(l)) \wedge \text{setp}(\text{cdr}(l))$
 else $l = \text{'nil}$ **endif**

DEFINITION:

```
sorted (l)
=  if listp (l)
    then if listp (cdr (l))
          then if car (cdr (l)) < car (l) then '*1*false
                else sorted (cdr (l)) endif
          else '*1>true endif
    else '*1>true endif
```

THEOREM: not-listp-remove

$$(\neg \text{listp } (l)) \rightarrow (\neg \text{listp } (\text{remove } (x, \text{cons } (x, l))))$$

EVENT: Disable not-listp-remove.

THEOREM: length-insert

$$\text{length } (\text{insert } (i, l)) = (1 + \text{length } (l))$$

EVENT: Disable length-insert.

THEOREM: length-remove

```
length (remove (x, l))
=  if x ∈ l then length (l) - 1
    else length (l) endif
```

EVENT: Disable length-remove.

THEOREM: length-permutation

$$\text{permutation } (a, b) \rightarrow (\text{length } (a) = \text{length } (b))$$

THEOREM: number-of-occurrences-of-set-member-is-1

$$(\text{setp } (l) \wedge (x \in l)) \rightarrow (\text{occurrences } (x, l) = '1)$$

THEOREM: plistp-remove

$$\text{plistp } (\text{remove } (x, l)) = \text{plistp } (l)$$

EVENT: Disable plistp-remove.

THEOREM: plistp-set

$$\text{setp } (l) \rightarrow \text{plistp } (l)$$

EVENT: Disable plistp-set.

THEOREM: insert-remove

$$(\text{sorted } (l) \wedge (x \neq y)) \\ \rightarrow (\text{insert } (x, \text{remove } (y, l)) = \text{remove } (y, \text{insert } (x, l)))$$

EVENT: Disable insert-remove.

THEOREM: member-insert
 $(x \in l) \rightarrow (x \in \text{insert}(i, l))$

EVENT: Disable member-insert.

THEOREM: not-member-insert
 $((x \notin l) \wedge (i \neq x)) \rightarrow (x \notin \text{insert}(i, l))$

EVENT: Disable not-member-insert.

THEOREM: member-remove1
 $(x \in \text{remove}(y, l)) \rightarrow (x \in l)$

THEOREM: member-remove2
 $((x \in l) \wedge (x \neq y)) \rightarrow (x \in \text{remove}(y, l))$

EVENT: Disable member-remove2.

THEOREM: not-member-remove1
 $\text{setp}(l) \rightarrow (x \notin \text{remove}(x, l))$

EVENT: Disable not-member-remove1.

THEOREM: not-member-remove2
 $(x \notin l) \rightarrow (x \notin \text{remove}(y, l))$

EVENT: Disable not-member-remove2.

THEOREM: not-member-permutation
 $(\text{permutation}(a, b) \wedge (x \notin b)) \rightarrow (x \notin a)$

THEOREM: member-permutation
 $(\text{permutation}(a, b) \wedge (x \in b)) \rightarrow (x \in a)$

THEOREM: remove-remove
 $(x \neq y) \rightarrow (\text{remove}(y, \text{remove}(x, l)) = \text{remove}(x, \text{remove}(y, l)))$

EVENT: Disable remove-remove.

THEOREM: sorted-insert
 $(\text{sorted}(l) \wedge (i \notin l)) \rightarrow \text{sorted}(\text{insert}(i, l))$

EVENT: Disable sorted-insert.

THEOREM: sorted-remove
 $\text{sorted}(l) \rightarrow \text{sorted}(\text{remove}(x, l))$

EVENT: Disable sorted-remove.

THEOREM: not-permutation
 $(\text{listp}(b) \wedge (\text{car}(b) \notin a)) \rightarrow (\neg \text{permutation}(a, b))$

EVENT: Disable not-permutation.

THEOREM: permutation-right-cons1
 $(\text{listp}(b) \wedge (\text{car}(b) \in a))$
 $\rightarrow (\text{permutation}(a, b) = \text{permutation}(\text{remove}(\text{car}(b), a), \text{cdr}(b)))$

EVENT: Disable permutation-right-cons1.

THEOREM: permutation-right-cons
 $\text{permutation}(a, \text{cons}(x, b))$
 $= \text{if } x \in a \text{ then } \text{permutation}(\text{remove}(x, a), b)$
 $\text{else } *1* \text{false endif}$

EVENT: Disable permutation-right-cons.

THEOREM: commutativity-of-permutation
 $\text{permutation}(b, a) = \text{permutation}(a, b)$

EVENT: Disable commutativity-of-permutation.

THEOREM: member-permutation2
 $(\text{permutation}(b, a) \wedge (x \in b)) \rightarrow (x \in a)$

THEOREM: permutation-reflexivity
 $\text{permutation}(l, l)$

EVENT: Disable permutation-reflexivity.

THEOREM: permutation-append-car
 $\text{permutation}(\text{append}(l, \text{cons}(x, \text{'nil})), \text{cons}(x, l))$

EVENT: Disable permutation-append-car.

THEOREM: permutation-append-insert
(permutation (a, b) \wedge (sorted (b) \wedge ((x \notin a) \wedge (x \notin b))))
 \rightarrow permutation (append (a, cons (x, 'nil)), insert (x, b))

EVENT: Disable permutation-append-insert.

THEOREM: permutation-remove-remove
(permutation (a, b) \wedge ((x \in a) \wedge (x \in b)))
 \rightarrow permutation (remove (x, a), remove (x, b))

EVENT: Disable permutation-remove-remove.

DEFINITION:
transitivity-of-permutation-induction (a, b, c)
= **if** listp (a)
 then transitivity-of-permutation-induction (cdr (a),
 remove (car (a), b),
 remove (car (a), c))
 else '0 **endif**

THEOREM: transitivity-of-permutation-base-case
((a \simeq nil) \wedge (permutation (a, b) \wedge permutation (b, c))) \rightarrow permutation (a, c)

THEOREM: transitivity-of-permutation-induction-step
(listp (a)
 \wedge (((permutation (cdr (a), remove (car (a), b))
 \wedge permutation (remove (car (a), b), remove (car (a), c)))
 \rightarrow permutation (cdr (a), remove (car (a), c)))
 \wedge (permutation (a, b) \wedge permutation (b, c))))
 \rightarrow permutation (a, c)

THEOREM: transitivity-of-permutation
(permutation (a, b) \wedge permutation (b, c)) \rightarrow permutation (a, c)

THEOREM: setp-remove
setp (a) \rightarrow setp (remove (x, a))

EVENT: Disable setp-remove.

DEFINITION:
setp-permutation-induction (a, b)
= **if** listp (a) **then** setp-permutation-induction (cdr (a), remove (car (a), b))
 else '0 **endif**

THEOREM: setp-permutation-base-case

$$((\neg \text{listp}(a)) \wedge \text{plistp}(a)) \rightarrow \text{setp}(a)$$

THEOREM: setp-permutation-induction-step

$$\begin{aligned} & (\text{listp}(a) \\ & \wedge (\text{setp}(b) \\ & \quad \wedge (\text{plistp}(a) \\ & \quad \quad \wedge (\text{permutation}(a, b) \\ & \quad \quad \quad \wedge ((\text{setp}(\text{remove}(\text{car}(a), b)) \\ & \quad \quad \quad \quad \wedge (\text{plistp}(\text{cdr}(a)) \\ & \quad \quad \quad \quad \quad \wedge \text{permutation}(\text{cdr}(a), \\ & \quad \quad \quad \quad \quad \quad \text{remove}(\text{car}(a), b)))))) \\ & \quad \rightarrow \text{setp}(\text{cdr}(a)))))) \\ & \rightarrow \text{setp}(a) \end{aligned}$$

THEOREM: setp-permutation

$$(\text{setp}(b) \wedge (\text{plistp}(a) \wedge \text{permutation}(a, b))) \rightarrow \text{setp}(a)$$

THEOREM: setp-insert

$$(\text{setp}(l) \wedge (i \notin l)) \rightarrow \text{setp}(\text{insert}(i, l))$$

EVENT: Disable setp-insert.

DEFINITION:

```
getnth(n, l)
= if listp(l)
  then if n ≈ 0 then car(l)
        else getnth(n - 1, cdr(l)) endif
  else '0 endif
```

DEFINITION:

```
putnth(v, n, l)
= if listp(l)
  then if n ≈ 0 then cons(v, cdr(l))
        else cons(car(l), putnth(v, n - 1, cdr(l))) endif
  else l endif
```

DEFINITION:

```
getseg(n, k, l)
= if k ≈ 0 then 'nil
  else cons(getnth(n, l), getseg(1 + n, k - 1, l)) endif
```

DEFINITION:

```
putseg(s, n, l)
= if listp(s) then putnth(car(s), n, putseg(cdr(s), 1 + n, l))
  else l endif
```

DEFINITION:

$\text{fixlength}(n, l, \text{default})$

```
= if listp(l)
  then if  $n \simeq 0$  then 'nil
    else cons(car(l), fixlength( $n - 1$ , cdr(l), default)) endif
  elseif  $n \simeq 0$  then 'nil
  else cons(default, fixlength( $n - 1$ , l, default)) endif
```

DEFINITION:

$\text{firstn}(n, l)$

```
= if listp(l)
  then if  $n \simeq 0$  then 'nil
    else cons(car(l), firstn( $n - 1$ , cdr(l))) endif
  else 'nil endif
```

DEFINITION:

$\text{nthcdr}(n, l)$

```
= if listp(l)
  then if  $n \simeq 0$  then l
    else nthcdr( $n - 1$ , cdr(l)) endif
  else l endif
```

THEOREM: non-zero-length-implies-listp

$(\text{length}(l) \neq 0) \rightarrow \text{listp}(l)$

EVENT: Disable non-zero-length-implies-listp.

THEOREM: listp-implies-non-zero-length

$\text{listp}(l) \rightarrow (\text{length}(l) \neq '0)$

EVENT: Disable listp-implies-non-zero-length.

THEOREM: plist-of-non-zero-length-is-a-list

$(\text{plist}(a) \wedge (\text{length}(a) \neq '0)) \rightarrow \text{listp}(a)$

EVENT: Disable plist-of-non-zero-length-is-a-list.

THEOREM: listp-append

$\text{listp}(\text{append}(a, b)) = (\text{listp}(a) \vee \text{listp}(b))$

EVENT: Disable listp-append.

THEOREM: listp-append-left

$\text{listp}(a) \rightarrow \text{listp}(\text{append}(a, b))$

EVENT: Disable listp-append-left.

THEOREM: listp-getseg
 $(k \neq 0) \rightarrow \text{listp}(\text{getseg}(n, k, l))$

EVENT: Disable listp-getseg.

THEOREM: listp-putnth
 $\text{listp}(\text{putnth}(v, n, l)) = \text{listp}(l)$

EVENT: Disable listp-putnth.

THEOREM: listp-putseg
 $\text{listp}(\text{putseg}(s, n, l)) = \text{listp}(l)$

EVENT: Disable listp-putseg.

THEOREM: rewrite-car-to-getnth-0
 $\text{car}(l) = \text{getnth}(0, l)$

EVENT: Disable rewrite-car-to-getnth-0.

THEOREM: car-append-casesplit
 $\text{car}(\text{append}(a, b))$
 $= \text{if listp}(a) \text{ then } \text{car}(a)$
 $\text{else } \text{car}(b) \text{ endif}$

EVENT: Disable car-append-casesplit.

THEOREM: car-append
 $\text{listp}(a) \rightarrow (\text{car}(\text{append}(a, b)) = \text{car}(a))$

EVENT: Disable car-append.

THEOREM: car-getseg
 $(k \neq 0) \rightarrow (\text{car}(\text{getseg}(n, k, l)) = \text{getnth}(n, l))$

EVENT: Disable car-getseg.

THEOREM: car-putnth-zero
 $(\text{listp}(l) \wedge (n \simeq 0)) \rightarrow (\text{car}(\text{putnth}(v, n, l)) = v)$

EVENT: Disable car-putnth-zero.

THEOREM: car-putseg-zero
 $((n \simeq 0) \wedge (\text{listp}(s) \wedge \text{listp}(l))) \rightarrow (\text{car}(\text{putseg}(s, n, l)) = \text{car}(s))$

EVENT: Disable car-putseg-zero.

THEOREM: cdr-append
 $\text{cdr}(\text{append}(a, b))$
 $=$ **if** $\text{listp}(a)$ **then** $\text{append}(\text{cdr}(a), b)$
 else $\text{cdr}(b)$ **endif**

EVENT: Disable cdr-append.

THEOREM: cdr-getseg
 $(k \neq 0) \rightarrow (\text{cdr}(\text{getseg}(n, k, l)) = \text{getseg}(1 + n, k - 1, l))$

EVENT: Disable cdr-getseg.

THEOREM: cdr-is-nil-when-length-is-1
 $(\text{plistp}(l) \wedge (\text{length}(l) = '1))$
 $\rightarrow ((\text{cdr}(l) = 'nil) = '*1*true)$

EVENT: Disable cdr-is-nil-when-length-is-1.

THEOREM: plist-of-length-0
 $(\text{plistp}(l) \wedge (\text{length}(l) = '0)) \rightarrow ((l = 'nil) = '*1*true)$

EVENT: Disable plist-of-length-0.

THEOREM: plistp-append
 $\text{plistp}(b) \rightarrow \text{plistp}(\text{append}(a, b))$

EVENT: Disable plistp-append.

THEOREM: plistp-list1
 $\text{plistp}(\text{cons}(x, 'nil))$

EVENT: Disable plistp-list1.

THEOREM: plistp-list2
 $\text{plistp}(\text{cons}(x, \text{cons}(y, 'nil)))$

EVENT: Disable plistp-list2.

THEOREM: plistp-putnth
 $\text{plistp}(l) \rightarrow \text{plistp}(\text{putnth}(v, i, l))$

EVENT: Disable plistp-putnth.

THEOREM: plistp-getseg
 $\text{plistp}(\text{getseg}(n, k, l))$

EVENT: Disable plistp-getseg.

THEOREM: plistp-putseg
 $\text{plistp}(l) \rightarrow \text{plistp}(\text{putseg}(s, n, l))$

EVENT: Disable plistp-putseg.

THEOREM: plistp-fixlength
 $\text{plistp}(\text{fixlength}(n, l, v))$

EVENT: Disable plistp-fixlength.

THEOREM: length-append
 $\text{length}(\text{append}(a, b)) = \text{length}(a) + \text{length}(b)$

EVENT: Disable length-append.

THEOREM: length-putnth
 $\text{length}(\text{putnth}(n, v, l)) = \text{length}(l)$

EVENT: Disable length-putnth.

THEOREM: length-getseg
 $\text{length}(\text{getseg}(n, k, l)) = \text{fix}(k)$

EVENT: Disable length-getseg.

THEOREM: length-putseg
 $\text{length}(\text{putseg}(s, n, l)) = \text{length}(l)$

EVENT: Disable length-putseg.

THEOREM: length-fixlength
 $\text{length}(\text{fixlength}(n, l, \text{default})) = \text{fix}(n)$

EVENT: Disable length-fixlength.

THEOREM: append-non-list-left
 $(\neg \text{listp}(a)) \rightarrow (\text{append}(a, b) = b)$

EVENT: Disable append-non-list-left.

THEOREM: append-non-list-left-corollary
 $(\text{length}(a) = '0) \rightarrow (\text{append}(a, b) = b)$

EVENT: Disable append-non-list-left-corollary.

THEOREM: append-nil-on-right
 $\text{plistp}(a) \rightarrow (\text{append}(a, 'nil) = a)$

EVENT: Disable append-nil-on-right.

THEOREM: append-cdr
 $\text{listp}(a) \rightarrow (\text{append}(\text{cdr}(a), b) = \text{cdr}(\text{append}(a, b)))$

EVENT: Disable append-cdr.

THEOREM: append-crock
 $\text{append}(\text{cons}(a, 'nil), \text{append}(b, c)) = \text{append}(\text{cons}(a, b), c)$

EVENT: Disable append-crock.

THEOREM: associativity-of-append
 $\text{append}(\text{append}(a, b), c) = \text{append}(a, \text{append}(b, c))$

EVENT: Disable associativity-of-append.

THEOREM: append-fixlength
 $\text{append}(\text{fixlength}(a, 'nil, \text{default}), \text{fixlength}(b, 'nil, \text{default}))$
 $= \text{fixlength}(a + b, 'nil, \text{default})$

EVENT: Disable append-fixlength.

THEOREM: getnth-0-2
 $\text{getnth}('0, \text{cons}(a, \text{cons}(b, 'nil))) = a$

EVENT: Disable getnth-0-2.

THEOREM: getnth-1-2
 $\text{getnth}('1, \text{cons}(a, \text{cons}(b, 'nil))) = b$

EVENT: Disable getnth-1-2.

THEOREM: list-getnth-0
 $(\text{listp}(l) \wedge (\text{length}(l) = '1)) \rightarrow (\text{cons}(\text{getnth}('0, l), 'nil) = l)$

EVENT: Disable list-getnth-0.

THEOREM: open-up-getnth
 $\text{getnth}(1 + n, \text{cons}(a, b)) = \text{getnth}(n, b)$

EVENT: Disable open-up-getnth.

THEOREM: open-up-getnth1
 $(n \neq 0) \rightarrow (\text{getnth}(n, \text{cons}(a, b)) = \text{getnth}(n - 1, b))$

EVENT: Disable open-up-getnth1.

THEOREM: open-up-getnth-on-zero
 $\text{getnth}('0, \text{cons}(a, b)) = a$

EVENT: Disable open-up-getnth-on-zero.

THEOREM: getnth-with-non-number-index
 $(n \notin \mathbf{N}) \rightarrow (\text{getnth}(n, l) = \text{getnth}('0, l))$

EVENT: Disable getnth-with-non-number-index.

THEOREM: getnth-append-for-length-left
 $\text{getnth}(\text{length}(a), \text{append}(a, b))$
 $= \text{if listp}(b) \text{ then } \text{car}(b)$
 $\text{else } '0 \text{ endif}$

EVENT: Disable getnth-append-for-length-left.

THEOREM: getnth-append-casesplit
 $\text{getnth}(n, \text{append}(a, b))$
 $= \text{if } n < \text{length}(a) \text{ then } \text{getnth}(n, a)$
 $\text{else } \text{getnth}(n - \text{length}(a), b) \text{ endif}$

EVENT: Disable getnth-append-casesplit.

THEOREM: getnth-append-right
if $n < \text{length}(a)$ **then** '*1*false
else '*1*true **endif**
 $\rightarrow (\text{getnth}(n, \text{append}(a, b)) = \text{getnth}(n - \text{length}(a), b))$

EVENT: Disable getnth-append-right.

THEOREM: getnth-append-left
 $(n < \text{length}(a)) \rightarrow (\text{getnth}(n, \text{append}(a, b)) = \text{getnth}(n, a))$

EVENT: Disable getnth-append-left.

THEOREM: getnth-putnth-casesplit
 $\text{getnth}(i, \text{putnth}(v, j, l))$
 $=$ **if** $i < \text{length}(l)$
 then if $\text{fix}(i) = \text{fix}(j)$ **then** v
 else $\text{getnth}(i, l)$ **endif**
 else $\text{getnth}(i, l)$ **endif**

EVENT: Disable getnth-putnth-casesplit.

THEOREM: getnth-putnth-coincidence
 $(n < \text{length}(l)) \rightarrow (\text{getnth}(n, \text{putnth}(v, n, l)) = v)$

EVENT: Disable getnth-putnth-coincidence.

THEOREM: getnth-putnth-non-interference
 $(\text{fix}(i) \neq \text{fix}(j)) \rightarrow (\text{getnth}(i, \text{putnth}(v, j, l)) = \text{getnth}(i, l))$

EVENT: Disable getnth-putnth-non-interference.

THEOREM: getnth-getseg-zero-index
 $((k \neq 0) \wedge (i \simeq 0)) \rightarrow (\text{getnth}(i, \text{getseg}(n, k, l)) = \text{getnth}(n, l))$

EVENT: Disable getnth-getseg-zero-index.

DEFINITION:
 $\text{getnth-getseg-induction}(i, n, k)$
 $=$ **if** $(i \simeq 0) \vee (k \simeq 0)$ **then** '0
 else $\text{getnth-getseg-induction}(i - 1, 1 + n, k - 1)$ **endif**

THEOREM: getnth-getseg
 $(i < k) \rightarrow (\text{getnth}(i, \text{getseg}(n, k, l)) = \text{getnth}(n + i, l))$

EVENT: Disable getnth-getseg.

THEOREM: getnth-getseg-inverse
 $(i < k) \rightarrow (\text{getnth}(n + i, l) = \text{getnth}(i, \text{getseg}(n, k, l)))$

EVENT: Disable getnth-getseg-inverse.

THEOREM: getnth-after-putseg
if $i < (n + \text{length}(s))$ **then** **'*1*false**
else **'*1*true** **endif**
 $\rightarrow (\text{getnth}(i, \text{putseg}(s, n, l)) = \text{getnth}(i, l))$

EVENT: Disable getnth-after-putseg.

THEOREM: getnth-before-putseg
 $(i < n) \rightarrow (\text{getnth}(i, \text{putseg}(s, n, l)) = \text{getnth}(i, l))$

EVENT: Disable getnth-before-putseg.

THEOREM: getnth-putseg-coincidence
(if $i < n$ **then** **'*1*false**
else **'*1*true** **endif**
 $\wedge (((i - n) < \text{length}(s)) \wedge (i < \text{length}(l)))$
 $\rightarrow (\text{getnth}(i, \text{putseg}(s, n, l)) = \text{getnth}(i - n, s))$

EVENT: Disable getnth-putseg-coincidence.

THEOREM: open-putnth-casesplit
 $\text{putnth}(v, n, \text{cons}(a, b))$
 $=$ **if** $n \simeq 0$ **then** $\text{cons}(v, b)$
else $\text{cons}(a, \text{putnth}(v, n - 1, b))$ **endif**

EVENT: Disable open-putnth-casesplit.

THEOREM: open-up-putnth1
 $(n \neq 0) \rightarrow (\text{putnth}(v, n, \text{cons}(a, b)) = \text{cons}(a, \text{putnth}(v, n - 1, b)))$

EVENT: Disable open-up-putnth1.

THEOREM: putnth-into-nil
 $\text{putnth}(v, i, \text{'nil}) = \text{'nil}$

EVENT: Disable putnth-into-nil.

THEOREM: open-up-putnth-on-zero
 $\text{putnth}(val, '0, \text{cons}(a, b)) = \text{cons}(val, b)$

EVENT: Disable open-up-putnth-on-zero.

THEOREM: putnth-with-non-number-index
 $(i \notin \mathbf{N}) \rightarrow (\text{putnth}(v, i, l) = \text{putnth}(v, '0, l))$

EVENT: Disable putnth-with-non-number-index.

THEOREM: putnth-has-no-effect
 $(n \not\prec \text{length}(l)) \rightarrow (\text{putnth}(v, n, l) = l)$

EVENT: Disable putnth-has-no-effect.

THEOREM: putnth-cons1
 $(n \simeq 0) \rightarrow (\text{putnth}(v, n, \text{cons}(a, b)) = \text{cons}(v, b))$

EVENT: Disable putnth-cons1.

THEOREM: putnth-cons2
 $(n \not\prec 0) \rightarrow (\text{putnth}(v, n, \text{cons}(a, b)) = \text{cons}(a, \text{putnth}(v, n - 1, b)))$

EVENT: Disable putnth-cons2.

THEOREM: putnth-shift
 $(i < \text{length}(l))$
 $\rightarrow (\text{putnth}(v, i, l) = \text{append}(\text{firstn}(i, l), \text{cons}(v, \text{nthcdr}(1 + i, l))))$

EVENT: Disable putnth-shift.

THEOREM: putnth-append-casesplit
 $\text{putnth}(v, n, \text{append}(a, b))$
 $= \text{if } n < \text{length}(a) \text{ then } \text{append}(\text{putnth}(v, n, a), b)$
 $\text{else } \text{append}(a, \text{putnth}(v, n - \text{length}(a), b)) \text{ endif}$

EVENT: Disable putnth-append-casesplit.

THEOREM: putnth-append-right

if $n < \text{length}(a)$ **then** '*1*false
else '*1*true endif

$\rightarrow (\text{putnth}(v, n, \text{append}(a, b)) = \text{append}(a, \text{putnth}(v, n - \text{length}(a), b)))$

EVENT: Disable putnth-append-right.

THEOREM: putnth-append-left

$(n < \text{length}(a)) \rightarrow (\text{putnth}(v, n, \text{append}(a, b)) = \text{append}(\text{putnth}(v, n, a), b))$

EVENT: Disable putnth-append-left.

THEOREM: putnth-getnth-coincidence

$\text{putnth}(\text{getnth}(i, l), i, l) = l$

EVENT: Disable putnth-getnth-coincidence.

THEOREM: putnth-getnth-coincidence1

$(\text{getnth}(i, l1) = \text{getnth}(i, l2))$

$\rightarrow (\text{putnth}(\text{getnth}(i, l1), i, \text{putnth}(v, i, l2)) = l2)$

EVENT: Disable putnth-getnth-coincidence1.

THEOREM: putnth-putnth-casesplit

$((i \in \mathbf{N})$

$\wedge ((j \in \mathbf{N})$

$\wedge \text{if } j < i \text{ then } \text{'*1*false}$
 $\text{else } \text{'*1*true endif})$

$\rightarrow (\text{putnth}(b, j, \text{putnth}(a, i, l))$

$= \text{if } i = j \text{ then } \text{putnth}(b, j, l)$
 $\text{else } \text{putnth}(a, i, \text{putnth}(b, j, l)) \text{ endif})$

EVENT: Disable putnth-putnth-casesplit.

THEOREM: putnth-coincidence

$(\text{fix}(i) = \text{fix}(j)) \rightarrow (\text{putnth}(b, j, \text{putnth}(a, i, l)) = \text{putnth}(b, j, l))$

EVENT: Disable putnth-coincidence.

THEOREM: putnth-non-interference

$(\text{fix}(i) \neq \text{fix}(j))$

$\rightarrow (\text{putnth}(b, j, \text{putnth}(a, i, l)) = \text{putnth}(a, i, \text{putnth}(b, j, l)))$

THEOREM: putnth-non-interference-corollary1

$$\begin{aligned} & (\text{fix } (i) < (1 + j)) \\ \rightarrow & (\text{putnth } (b, 1 + j, \text{putnth } (a, i, l)) = \text{putnth } (a, i, \text{putnth } (b, 1 + j, l))) \end{aligned}$$

EVENT: Disable putnth-non-interference-corollary1.

THEOREM: putnth-non-interference-corollary2

$$\begin{aligned} & (\text{fix } (i) < (j + k)) \\ \rightarrow & (\text{putnth } (b, j + k, \text{putnth } (a, i, l)) = \text{putnth } (a, i, \text{putnth } (b, j + k, l))) \end{aligned}$$

EVENT: Disable putnth-non-interference-corollary2.

THEOREM: putnth-putseg-coincidence-proof

$$\begin{aligned} & ((n \in \mathbf{N}) \\ & \wedge ((i \in \mathbf{N}) \\ & \quad \wedge (\text{if } i < n \text{ then } '*1*false \\ & \quad \quad \text{else } '*1*true \text{ endif} \\ & \quad \quad \wedge ((i - n) < \text{length}(s)))))) \\ \rightarrow & (\text{putnth } (v, i, \text{putseg } (s, n, l)) = \text{putseg } (\text{putnth } (v, i - n, s), n, l)) \end{aligned}$$

EVENT: Disable putnth-putseg-coincidence-proof.

THEOREM: putseg-with-non-number-index

$$(n \notin \mathbf{N}) \rightarrow (\text{putseg } (s, n, l) = \text{putseg } (s, '0, l))$$

EVENT: Disable putseg-with-non-number-index.

THEOREM: putnth-putseg-coincidence

$$\begin{aligned} & (\text{if } i < n \text{ then } '*1*false \\ & \quad \text{else } '*1*true \text{ endif} \\ & \quad \wedge ((i - n) < \text{length}(s))) \\ \rightarrow & (\text{putnth } (v, i, \text{putseg } (s, n, l)) = \text{putseg } (\text{putnth } (v, i - n, s), n, l)) \end{aligned}$$

EVENT: Disable putnth-putseg-coincidence.

THEOREM: putnth-putseg-coincidence-inverse

$$\begin{aligned} & ((i < \text{length}(s)) \wedge (i \in \mathbf{N})) \\ \rightarrow & (\text{putseg } (\text{putnth } (v, i, s), n, l) = \text{putnth } (v, i + n, \text{putseg } (s, n, l))) \end{aligned}$$

EVENT: Disable putnth-putseg-coincidence-inverse.

THEOREM: putnth-after-putseg-proof

```

((i ∈ N)
 ∧ ((n ∈ N)
   ∧ (if i < n then '*1*false
      else '*1*true endif
      ∧ if (i - n) < length(s) then '*1*false
      else '*1*true endif)))
→ (putnth(v, i, putseg(s, n, l)) = putseg(s, n, putnth(v, i, l)))

```

EVENT: Disable putnth-after-putseg-proof.

```

THEOREM: putnth-after-putseg
(if i < n then '*1*false
 else '*1*true endif
 ∧ if (i - n) < length(s) then '*1*false
 else '*1*true endif)
→ (putnth(v, i, putseg(s, n, l)) = putseg(s, n, putnth(v, i, l)))

```

EVENT: Disable putnth-after-putseg.

```

THEOREM: getseg-of-length-zero
(k ≈ 0) → (getseg(n, k, l) = 'nil)

```

EVENT: Disable getseg-of-length-zero.

```

THEOREM: getseg-add1-length
getseg(n, 1 + k, l) = append(getseg(n, k, l), cons(getnth(n + k, l), 'nil))

```

EVENT: Disable getseg-add1-length.

```

THEOREM: getseg-with-non-number-index
(n ∉ N) → (getseg(n, k, l) = getseg('0, k, l))

```

EVENT: Disable getseg-with-non-number-index.

```

THEOREM: getseg-add1-cons
getseg(1 + n, k, cons(a, b)) = getseg(n, k, b)

```

EVENT: Disable getseg-add1-cons.

```

THEOREM: getseg-length-1
getseg(n, '1, l) = cons(getnth(n, l), 'nil)

```


THEOREM: getseg-length-1-rule
 $(k = '1) \rightarrow (\text{getseg}(n, k, l) = \text{cons}(\text{getnth}(n, l), 'nil))$

EVENT: Disable getseg-length-1-rule.

THEOREM: getseg-add1-length2
 $((n \in \mathbf{N})$
 \wedge **if** $\text{length}(l) < (1 + n)$ **then** $'*1*false$
else $'*1>true$ **endif**)
 $\rightarrow (\text{getseg}(n, 1 + k, l) = \text{cons}(\text{getnth}(n, l), \text{getseg}(1 + n, k, l)))$

EVENT: Disable getseg-add1-length2.

THEOREM: getseg-append-right
if $n < \text{length}(a)$ **then** $'*1*false$
else $'*1>true$ **endif**
 $\rightarrow (\text{getseg}(n, k, \text{append}(a, b)) = \text{getseg}(n - \text{length}(a), k, b))$

EVENT: Disable getseg-append-right.

THEOREM: getseg-across-append
 $((n \in \mathbf{N})$
 \wedge (**if** $\text{length}(a) < n$ **then** $'*1*false$
else $'*1>true$ **endif**
 \wedge **if** $(n + k) < \text{length}(a)$ **then** $'*1*false$
else $'*1>true$ **endif**)
 $\rightarrow (\text{getseg}(n, k, \text{append}(a, b))$
 $= \text{append}(\text{getseg}(n, \text{length}(a) - n, a),$
 $\text{getseg}('0, k - (\text{length}(a) - n), b))$

EVENT: Disable getseg-across-append.

THEOREM: getseg-append-left
if $\text{length}(a) < (n + k)$ **then** $'*1*false$
else $'*1>true$ **endif**
 $\rightarrow (\text{getseg}(n, k, \text{append}(a, b)) = \text{getseg}(n, k, a))$

EVENT: Disable getseg-append-left.

THEOREM: getseg-after-putnth
 $(n < i) \rightarrow (\text{getseg}(i, k, \text{putnth}(v, n, l)) = \text{getseg}(i, k, l))$

EVENT: Disable getseg-after-putnth.

THEOREM: getseg-before-putnth

```
if  $n < (i + k)$  then '1*false
else '1*true endif
→ (getseg( $i, k, \text{putnth}(v, n, l)$ ) = getseg( $i, k, l$ ))
```

EVENT: Disable getseg-before-putnth.

THEOREM: getseg-putnth-coincidence

```
(if  $n < i$  then '1*false
else '1*true endif
∧ ((( $n - i$ ) <  $k$ ) ∧ ( $n < \text{length}(l)$ )))
→ (getseg( $i, k, \text{putnth}(v, n, l)$ ) = putnth( $v, n - i, \text{getseg}(i, k, l)$ ))
```

EVENT: Disable getseg-putnth-coincidence.

DEFINITION:

```
integer-induction-instance( $n, k$ )
= if  $k \simeq 0$  then '0
  else integer-induction-instance( $1 + n, k - 1$ ) endif
```

THEOREM: getseg-getseg

```
if  $k < (i + j)$  then '1*false
else '1*true endif
→ (getseg( $i, j, \text{getseg}(n, k, l)$ ) = getseg( $n + i, j, l$ ))
```

EVENT: Disable getseg-getseg.

THEOREM: getseg-after-putseg

```
if  $i < (n + \text{length}(s))$  then '1*false
else '1*true endif
→ (getseg( $i, k, \text{putseg}(s, n, l)$ ) = getseg( $i, k, l$ ))
```

EVENT: Disable getseg-after-putseg.

THEOREM: getseg-before-putseg

```
if  $n < (i + k)$  then '1*false
else '1*true endif
→ (getseg( $i, k, \text{putseg}(s, n, l)$ ) = getseg( $i, k, l$ ))
```

EVENT: Disable getseg-before-putseg.

THEOREM: getseg-within-putseg

```
(if  $i < n$  then '1*false
```

```

else '*1*true endif
 $\wedge$  (if length( $l$ ) < ( $n$  + length( $s$ )) then '*1*false
      else '*1*true endif
       $\wedge$  (if ( $n$  + length( $s$ )) < ( $i$  +  $k$ ) then '*1*false
            else '*1*true endif))
 $\rightarrow$  (getseg( $i$ ,  $k$ , putseg( $s$ ,  $n$ ,  $l$ )) = getseg( $i - n$ ,  $k$ ,  $s$ ))

```

EVENT: Disable getseg-within-putseg.

DEFINITION:

```

getseg-putseg-induction( $i$ ,  $k$ ,  $s$ ,  $n$ )
= if  $k \simeq 0$  then '0
  elseif listp( $s$ )
  then getseg-putseg-induction( $1 + i$ ,  $k - 1$ , cdr( $s$ ),  $1 + n$ )
  else '0 endif

```

THEOREM: putseg-has-no-effect

$(n \not\prec \text{length}(l)) \rightarrow (\text{putseg}(s, n, l) = l)$

EVENT: Disable putseg-has-no-effect.

THEOREM: putseg-cons2

$(n \not\prec 0) \rightarrow (\text{putseg}(s, n, \text{cons}(a, b)) = \text{cons}(a, \text{putseg}(s, n - 1, b)))$

EVENT: Disable putseg-cons2.

THEOREM: getseg-around-putseg-proof

```

(( $i \in \mathbf{N}$ )
 $\wedge$  (( $n \in \mathbf{N}$ )
       $\wedge$  (if  $n < i$  then '*1*false
            else '*1*true endif
             $\wedge$  (if length( $l$ ) < ( $n$  + length( $s$ )) then '*1*false
                  else '*1*true endif
                   $\wedge$  (if ( $i + k$ ) < ( $n$  + length( $s$ ))
                        then '*1*false
                        else '*1*true endif))))))
 $\rightarrow$  (getseg( $i$ ,  $k$ , putseg( $s$ ,  $n$ ,  $l$ )) = putseg( $s$ ,  $n - i$ , getseg( $i$ ,  $k$ ,  $l$ )))

```

EVENT: Disable getseg-around-putseg-proof.

THEOREM: numberp-casesplit

$(a \in \mathbf{N}) \vee (a \notin \mathbf{N})$

THEOREM: getseg-around-putseg

```
(if n < i then '*1*false
  else '*1*true endif
  ∧ (if length(l) < (n + length(s)) then '*1*false
    else '*1*true endif
    ∧ (if (i + k) < (n + length(s)) then '*1*false
      else '*1*true endif))
  → (getseg(i, k, putseg(s, n, l)) = putseg(s, n - i, getseg(i, k, l)))
```

EVENT: Disable getseg-around-putseg.

THEOREM: putseg-nil-segment

```
putseg('nil, n, l) = l
```

EVENT: Disable putseg-nil-segment.

THEOREM: putseg-non-list-segment

```
(¬ listp(s)) → (putseg(s, n, l) = l)
```

EVENT: Disable putseg-non-list-segment.

THEOREM: open-up-putseg-on-segment-cons

```
((n ∈ ℕ)
  ∧ (if length(l) < (n + length(cons(a, b))) then '*1*false
    else '*1*true endif)
  → (putseg(cons(a, b), n, l) = putnth(a, n, putseg(b, '1 + n, l)))
```

EVENT: Disable open-up-putseg-on-segment-cons.

THEOREM: putseg-cons1

```
((n ≃ 0) ∧ listp(s))
  → (putseg(s, n, cons(a, b)) = cons(car(s), putseg(cdr(s), n, b)))
```

EVENT: Disable putseg-cons1.

THEOREM: putseg-getseg-coincidence

```
(i = n) → (putseg(getseg(n, k, l), i, l) = l)
```

EVENT: Disable putseg-getseg-coincidence.

THEOREM: equality-of-getnth-backchain-proof

```
(if i < n then '*1*false
```

else '1*true endif
 \wedge ((($i - n < k$) \wedge (getseg($n, k, l1$) = getseg($n, k, l2$))))
 \rightarrow (getnth($i, l1$) = getnth($i, l2$))

DEFINITION:

nested-intervals($n1, k1, n2, k2$)
 = **(if** $n2 < n1$ **then** '1*false
 else '1*true **endif**
 \wedge **if** ($n1 + k1 < (n2 + k2)$) **then** '1*false
 else '1*true **endif**)

DEFINITION:

strange-induction(a, b, c, d)
 = **if** ($a \simeq 0$) \vee ($b \simeq 0$) **then** '0
 else strange-induction($a - 1, b - 1, 1 + c, 1 + d$) **endif**

THEOREM: plus-arg2-must-be-zero

(if $i < n$ **then** '1*false
 else '1*true **endif**
 \wedge **if** $n < (i + j)$ **then** '1*false
 else '1*true **endif**)
 \rightarrow ($j \simeq 0$)

EVENT: Disable plus-arg2-must-be-zero.

THEOREM: getseg-must-be-nil

(if $i < n$ **then** '1*false
 else '1*true **endif**
 \wedge **if** $n < (i + j)$ **then** '1*false
 else '1*true **endif**)
 \rightarrow (getseg(i, j, l) = 'nil)

EVENT: Disable getseg-must-be-nil.

THEOREM: equality-of-getseg-backchain-base-case

((($j \simeq 0$) \vee ($k \simeq 0$))
 \wedge (($i \in \mathbf{N}$)
 \wedge (($n \in \mathbf{N}$)
 \wedge (nested-intervals(n, k, i, j)
 \wedge (getseg($n, k, l1$) = getseg($n, k, l2$))))))
 \rightarrow (getseg($i, j, l1$) = getseg($i, j, l2$))

THEOREM: equality-of-getseg-backchain-induction-step-case1

(($i \in \mathbf{N}$))

$$\begin{aligned}
& \wedge ((n \in \mathbf{N}) \\
& \quad \wedge ((j \neq 0) \\
& \quad \quad \wedge ((k \neq 0) \\
& \quad \quad \quad \wedge (((\text{nested-intervals}(1 + n, k - 1, 1 + i, j - 1) \\
& \quad \quad \quad \quad \wedge (\text{getseg}(1 + n, k - 1, l1) \\
& \quad \quad \quad \quad \quad = \text{getseg}(1 + n, k - 1, l2))) \\
& \quad \quad \quad \rightarrow (\text{getseg}(1 + i, j - 1, l1) \\
& \quad \quad \quad \quad = \text{getseg}(1 + i, j - 1, l2))) \\
& \quad \quad \wedge (\text{nested-intervals}(n, k, i, j) \\
& \quad \quad \quad \wedge ((\text{getseg}(n, k, l1) \\
& \quad \quad \quad \quad = \text{getseg}(n, k, l2)) \\
& \quad \quad \quad \quad \wedge (i = n)))))) \\
& \rightarrow (\text{getseg}(i, j, l1) = \text{getseg}(i, j, l2))
\end{aligned}$$

THEOREM: equality-of-getseg-backchain-induction-step-case2

$$\begin{aligned}
& ((i \in \mathbf{N}) \\
& \quad \wedge ((n \in \mathbf{N}) \\
& \quad \quad \wedge ((j \neq 0) \\
& \quad \quad \quad \wedge ((k \neq 0) \\
& \quad \quad \quad \quad \wedge (((\text{nested-intervals}(1 + n, k - 1, 1 + i, j - 1) \\
& \quad \quad \quad \quad \quad \wedge (\text{getseg}(1 + n, k - 1, l1) \\
& \quad \quad \quad \quad \quad \quad = \text{getseg}(1 + n, k - 1, l2))) \\
& \quad \quad \quad \quad \rightarrow (\text{getseg}(1 + i, j - 1, l1) \\
& \quad \quad \quad \quad \quad = \text{getseg}(1 + i, j - 1, l2))) \\
& \quad \quad \quad \wedge (\text{nested-intervals}(n, k, i, j) \\
& \quad \quad \quad \quad \wedge ((\text{getseg}(n, k, l1) \\
& \quad \quad \quad \quad \quad = \text{getseg}(n, k, l2)) \\
& \quad \quad \quad \quad \quad \wedge (n < i)))))) \\
& \rightarrow (\text{getseg}(i, j, l1) = \text{getseg}(i, j, l2))
\end{aligned}$$

THEOREM: equality-of-getseg-backchain-induction-step

$$\begin{aligned}
& ((i \in \mathbf{N}) \\
& \quad \wedge ((n \in \mathbf{N}) \\
& \quad \quad \wedge ((j \neq 0) \\
& \quad \quad \quad \wedge ((k \neq 0) \\
& \quad \quad \quad \quad \wedge (((\text{nested-intervals}(1 + n, k - 1, 1 + i, j - 1) \\
& \quad \quad \quad \quad \quad \wedge (\text{getseg}(1 + n, k - 1, l1) \\
& \quad \quad \quad \quad \quad \quad = \text{getseg}(1 + n, k - 1, l2))) \\
& \quad \quad \quad \quad \rightarrow (\text{getseg}(1 + i, j - 1, l1) \\
& \quad \quad \quad \quad \quad = \text{getseg}(1 + i, j - 1, l2))) \\
& \quad \quad \quad \wedge (\text{nested-intervals}(n, k, i, j) \\
& \quad \quad \quad \quad \wedge (\text{getseg}(n, k, l1) \\
& \quad \quad \quad \quad \quad = \text{getseg}(n, k, l2)))))) \\
& \rightarrow (\text{getseg}(i, j, l1) = \text{getseg}(i, j, l2))
\end{aligned}$$

THEOREM: equality-of-getseg-backchain-proof

$$\begin{aligned} & ((i \in \mathbf{N}) \\ & \wedge ((n \in \mathbf{N}) \\ & \quad \wedge (\text{nested-intervals}(n, k, i, j) \\ & \quad \quad \wedge (\text{getseg}(n, k, l1) = \text{getseg}(n, k, l2)))))) \\ \rightarrow & (\text{getseg}(i, j, l1) = \text{getseg}(i, j, l2)) \end{aligned}$$

THEOREM: equality-of-getseg-backchain

$$\begin{aligned} & (\text{nested-intervals}(n, k, i, j) \wedge (\text{getseg}(n, k, l1) = \text{getseg}(n, k, l2))) \\ \rightarrow & (\text{getseg}(i, j, l1) = \text{getseg}(i, j, l2)) \end{aligned}$$

DEFINITION: $\text{getnth2}(i, j, l) = \text{getnth}(j, \text{getnth}(i, l))$

DEFINITION:

$$\text{putnth2}(v, i, j, l) = \text{putnth}(\text{putnth}(v, j, \text{getnth}(i, l)), i, l)$$

THEOREM: plistp-putnth2

$$\text{plistp}(l) \rightarrow \text{plistp}(\text{putnth2}(v, i, j, l))$$

EVENT: Disable plistp-putnth2.

THEOREM: length-putnth2

$$\text{length}(\text{putnth2}(v, i, j, l)) = \text{length}(l)$$

EVENT: Disable length-putnth2.

THEOREM: getnth-putnth-coincidence1

$$\begin{aligned} & ((\text{fix}(i) = \text{fix}(j)) \wedge (i < \text{length}(l))) \\ \rightarrow & (\text{getnth}(i, \text{putnth}(v, j, l)) = v) \end{aligned}$$

EVENT: Disable getnth-putnth-coincidence1.

THEOREM: getnth2-putnth2-coincidence

$$\begin{aligned} & ((i < \text{length}(l)) \wedge (j < \text{length}(\text{getnth}(i, l)))) \\ \rightarrow & (\text{getnth2}(i, j, \text{putnth2}(v, i, j, l)) = v) \end{aligned}$$

EVENT: Disable getnth2-putnth2-coincidence.

THEOREM: getnth2-putnth2-non-interference1

$$(\text{fix}(i) \neq \text{fix}(x)) \rightarrow (\text{getnth2}(i, j, \text{putnth2}(v, x, y, l)) = \text{getnth2}(i, j, l))$$

EVENT: Disable getnth2-putnth2-non-interference1.

THEOREM: getnth2-putnth2-non-interference2
((fix (i) = fix (x)) \wedge ((i < length (l)) \wedge (fix (j) \neq fix (y))))
 \rightarrow (getnth2 (i, j, putnth2 (v, x, y, l)) = getnth2 (i, j, l))

EVENT: Disable getnth2-putnth2-non-interference2.

THEOREM: fixlength-of-length-0
(n \simeq 0) \rightarrow (fixlength (n, l, x) = 'nil)

EVENT: Disable fixlength-of-length-0.

THEOREM: fixlength-idempotence
fixlength (n, fixlength (n, l, a), b) = fixlength (n, l, a)

EVENT: Disable fixlength-idempotence.

THEOREM: fixlength-noop
(plistp (l) \wedge (n = length (l))) \rightarrow (fixlength (n, l, default) = l)

EVENT: Disable fixlength-noop.

DEFINITION:
reverse (l)
= **if** plistp (l) **then** append (reverse (cdr (l)), cons (car (l), 'nil))
 else 'nil **endif**

THEOREM: plistp-reverse
plistp (reverse (a))

EVENT: Disable plistp-reverse.

THEOREM: reverse-append
reverse (append (a, b)) = append (reverse (b), reverse (a))

EVENT: Disable reverse-append.

THEOREM: reverse-list1
reverse (cons (a, 'nil)) = cons (a, 'nil)

EVENT: Disable reverse-list1.

THEOREM: reverse-list2
reverse (cons (a, cons (a, 'nil))) = cons (a, cons (a, 'nil))

EVENT: Disable reverse-list2.

THEOREM: reverse-reverse
 $\text{plistp}(l) \rightarrow (\text{reverse}(\text{reverse}(l)) = l)$

EVENT: Disable reverse-reverse.

DEFINITION: $\text{finite-numberp}(n, lub) = ((n \in \mathbf{N}) \wedge (n < lub))$

DEFINITION:
 $\text{number-listp}(l)$
 $=$ **if** $\text{listp}(l)$ **then** $(\text{car}(l) \in \mathbf{N}) \wedge \text{number-listp}(\text{cdr}(l))$
else **'*1*true endif**

DEFINITION:
 $\text{finite-number-listp}(l, lub)$
 $=$ **if** $\text{listp}(l)$
then $\text{finite-numberp}(\text{car}(l), lub) \wedge \text{finite-number-listp}(\text{cdr}(l), lub)$
else **'*1*true endif**

DEFINITION:
 $\text{non-zero-listp}(l)$
 $=$ **if** $\text{listp}(l)$ **then** $(\text{car}(l) \neq 0) \wedge \text{non-zero-listp}(\text{cdr}(l))$
else **'*1*true endif**

THEOREM: non-1-finite-number-is-zero
 $(\text{finite-numberp}(a, '2) \wedge (a \neq '1)) \rightarrow (a = '0)$

THEOREM: finite-numberp-member
 $(\text{finite-number-listp}(b, lub) \wedge (x \in b)) \rightarrow \text{finite-numberp}(x, lub)$

EVENT: Disable finite-numberp-member.

THEOREM: a-finite-number-is-lessp-than-its-lub
 $\text{finite-numberp}(n, lub) \rightarrow ((n < lub) = '*1*true)$

EVENT: Disable a-finite-number-is-lessp-than-its-lub.

THEOREM: a-finite-number-is-a-number
 $\text{finite-numberp}(n, lub) \rightarrow (n \in \mathbf{N})$

THEOREM: finite-numberp-from-smaller-lub
 $(\text{finite-numberp}(n, i) \wedge (i < j)) \rightarrow \text{finite-numberp}(n, j)$

THEOREM: finite-number-sub1
 $\text{finite-numberp}(a, b) \rightarrow \text{finite-numberp}(a - 1, b)$

EVENT: Disable finite-number-sub1.

THEOREM: finite-numberp-remainder
 $(b \neq 0) \rightarrow \text{finite-numberp}(a \bmod b, b)$

EVENT: Disable finite-numberp-remainder.

THEOREM: a-finite-number-list-is-a-number-list
 $\text{finite-number-listp}(l, lub) \rightarrow \text{number-listp}(l)$

THEOREM: finite-numberp-getnth
 $(\text{finite-number-listp}(l, lub) \wedge (i < \text{length}(l)))$
 $\rightarrow \text{finite-numberp}(\text{getnth}(i, l), lub)$

EVENT: Disable finite-numberp-getnth.

THEOREM: numberp-getnth
 $(\text{finite-number-listp}(l, lub) \wedge (i < \text{length}(l))) \rightarrow (\text{getnth}(i, l) \in \mathbf{N})$

EVENT: Disable numberp-getnth.

THEOREM: lessp-getnth
 $(\text{finite-number-listp}(l, lub) \wedge (i < \text{length}(l)))$
 $\rightarrow ((\text{getnth}(i, l) < lub) = \text{'*1*true})$

EVENT: Disable lessp-getnth.

THEOREM: lessp-getnth-linear
 $(\text{finite-number-listp}(l, lub) \wedge (i < \text{length}(l)))$
 $\rightarrow (\text{getnth}(i, l) < lub)$

EVENT: Disable lessp-getnth-linear.

THEOREM: element-of-non-zero-list-is-a-number
 $(\text{non-zero-listp}(l) \wedge (i < \text{length}(l))) \rightarrow (\text{getnth}(i, l) \in \mathbf{N})$

EVENT: Disable element-of-non-zero-list-is-a-number.

THEOREM: element-of-non-zero-list-is-non-zero
 $(\text{non-zero-listp}(l) \wedge (i < \text{length}(l))) \rightarrow (\text{getnth}(i, l) \neq '0)$

EVENT: Disable element-of-non-zero-list-is-non-zero.

THEOREM: finite-number-listp-list
 $\text{finite-numberp}(n, lub) \rightarrow \text{finite-number-listp}(\text{cons}(n, \text{'nil}), lub)$

EVENT: Disable finite-number-listp-list.

THEOREM: finite-number-listp-append
 $(\text{finite-number-listp}(a, lub) \wedge \text{finite-number-listp}(b, lub))$
 $\rightarrow \text{finite-number-listp}(\text{append}(a, b), lub)$

EVENT: Disable finite-number-listp-append.

THEOREM: finite-number-listp-remove
 $\text{finite-number-listp}(l, lub) \rightarrow \text{finite-number-listp}(\text{remove}(x, l), lub)$

EVENT: Disable finite-number-listp-remove.

THEOREM: finite-number-listp-permutation
 $(\text{finite-number-listp}(b, lub) \wedge \text{permutation}(a, b))$
 $\rightarrow \text{finite-number-listp}(a, lub)$

EVENT: Disable finite-number-listp-permutation.

THEOREM: finite-number-listp-putnth
 $(\text{finite-number-listp}(l, lub) \wedge \text{finite-numberp}(v, lub))$
 $\rightarrow \text{finite-number-listp}(\text{putnth}(v, n, l), lub)$

EVENT: Disable finite-number-listp-putnth.

THEOREM: finite-number-listp-getseg
 $(\text{finite-number-listp}(l, lub)$
 $\wedge \text{if length}(l) < (n + k) \text{ then '1*false}$
 $\text{else '1*true endif})$
 $\rightarrow \text{finite-number-listp}(\text{getseg}(n, k, l), lub)$

EVENT: Disable finite-number-listp-getseg.

THEOREM: finite-number-listp-putseg
 $(\text{finite-number-listp}(l, lub) \wedge \text{finite-number-listp}(s, lub))$
 $\rightarrow \text{finite-number-listp}(\text{putseg}(s, n, l), lub)$

EVENT: Disable finite-number-listp-putseg.

THEOREM: finite-number-listp-fixlength
(finite-number-listp (l , lub) \wedge finite-numberp ($default$, lub))
 \rightarrow finite-number-listp (fixlength (n , l , $default$), lub)

EVENT: Disable finite-number-listp-fixlength.

THEOREM: finite-number-listp-for-zero-list
($lub \neq 0$) \rightarrow finite-number-listp (fixlength (n , 'nil', '0'), lub)

EVENT: Disable finite-number-listp-for-zero-list.

THEOREM: finite-number-listp-from-smaller-lub
(finite-number-listp (l , i) \wedge ($i < j$)) \rightarrow finite-number-listp (l , j)

THEOREM: non-zero-listp-list
($a \neq 0$) \rightarrow non-zero-listp (cons (a , 'nil'))

EVENT: Disable non-zero-listp-list.

THEOREM: non-zero-listp-append
(non-zero-listp (a) \wedge non-zero-listp (b)) \rightarrow non-zero-listp (append (a , b))

EVENT: Disable non-zero-listp-append.

THEOREM: non-zero-listp-putnth
(non-zero-listp (l) \wedge ($v \neq 0$)) \rightarrow non-zero-listp (putnth (v , n , l))

EVENT: Disable non-zero-listp-putnth.

THEOREM: non-zero-listp-getseg
(non-zero-listp (l)
 \wedge **if** length (l) $<$ ($n + k$) **then** '*1*false
 else '*1*true **endif**)
 \rightarrow non-zero-listp (getseg (n , k , l))

EVENT: Disable non-zero-listp-getseg.

THEOREM: non-zero-listp-putseg
(non-zero-listp (l) \wedge non-zero-listp (s)) \rightarrow non-zero-listp (putseg (s , n , l))

EVENT: Disable non-zero-listp-putseg.

THEOREM: non-zero-listp-fixlength
 $(\text{non-zero-listp}(l) \wedge (\text{default} \neq 0))$
 $\rightarrow \text{non-zero-listp}(\text{fixlength}(n, l, \text{default}))$

EVENT: Disable non-zero-listp-fixlength.

THEOREM: car-nthcdr
 $(n < \text{length}(l)) \rightarrow (\text{car}(\text{nthcdr}(n, l)) = \text{getnth}(n, l))$

EVENT: Disable car-nthcdr.

THEOREM: cdr-nthcdr
 $(n < \text{length}(l)) \rightarrow (\text{cdr}(\text{nthcdr}(n, l)) = \text{nthcdr}(1 + n, l))$

EVENT: Disable cdr-nthcdr.

THEOREM: listp-nthcdr
 $(n < \text{length}(l)) \rightarrow \text{listp}(\text{nthcdr}(n, l))$

EVENT: Disable listp-nthcdr.

THEOREM: plistp-firstn
 $\text{plistp}(\text{firstn}(n, l))$

EVENT: Disable plistp-firstn.

THEOREM: plistp-nthcdr
 $\text{plistp}(\text{nthcdr}(n, l)) = \text{plistp}(l)$

EVENT: Disable plistp-nthcdr.

THEOREM: length-firstn
if $\text{length}(l) < n$ **then** `'*1*false`
else `'*1>true` **endif**
 $\rightarrow (\text{length}(\text{firstn}(n, l)) = \text{fix}(n))$

EVENT: Disable length-firstn.

THEOREM: length-nthcdr
 $\text{length}(\text{nthcdr}(n, l)) = (\text{length}(l) - n)$

EVENT: Disable length-nthcdr.

THEOREM: firstn-zero
 $(k \simeq 0) \rightarrow (\text{firstn}(k, l) = \text{'nil'})$

EVENT: Disable firstn-zero.

THEOREM: nthcdr-zero
 $(n \simeq 0) \rightarrow (\text{nthcdr}(n, l) = l)$

EVENT: Disable nthcdr-zero.

THEOREM: nthcdr-with-large-n
 $(\text{plistsp}(l)$
 $\wedge \text{ if } n < \text{length}(l) \text{ then '1*false}$
 $\text{ else '1*true endif})$
 $\rightarrow (\text{nthcdr}(n, l) = \text{'nil'})$

EVENT: Disable nthcdr-with-large-n.

THEOREM: firstn-for-length
 $(\text{plistsp}(l) \wedge (n = \text{length}(l))) \rightarrow (\text{firstn}(n, l) = l)$

EVENT: Disable firstn-for-length.

THEOREM: append-firstn-nthcdr
 $\text{append}(\text{firstn}(i, l), \text{nthcdr}(i, l)) = l$

EVENT: Disable append-firstn-nthcdr.

THEOREM: firstn-firstn
 $(i < n) \rightarrow (\text{firstn}(i, \text{firstn}(n, l)) = \text{firstn}(i, l))$

EVENT: Disable firstn-firstn.

THEOREM: firstn-nthcdr
 $\text{plistsp}(l) \rightarrow (\text{firstn}(\text{length}(l) - k, \text{nthcdr}(k, l)) = \text{nthcdr}(k, l))$

EVENT: Disable firstn-nthcdr.

THEOREM: nthcdr-append-right

```

if  $n < \text{length}(a)$  then '*1*false
else '*1>true endif
→  $(\text{nthcdr}(n, \text{append}(a, b)) = \text{nthcdr}(n - \text{length}(a), b))$ 

```

EVENT: Disable nthcdr-append-right.

```

THEOREM: nthcdr-append-left
if  $\text{length}(a) < n$  then '*1*false
else '*1>true endif
→  $(\text{nthcdr}(n, \text{append}(a, b)) = \text{append}(\text{nthcdr}(n, a), b))$ 

```

EVENT: Disable nthcdr-append-left.

```

DEFINITION:
tablep( $n, l$ )
= if listp( $l$ )
  then plistp(car( $l$ ))
    ∧ ((length(car( $l$ )) =  $n$ ) ∧ tablep( $n, \text{cdr}(l)$ ))
  else  $l = \text{'nil}$  endif

```

```

THEOREM: length-of-nthcdr-goes-down
(listp( $l$ ) ∧ ( $n \neq 0$ ))
→ ((length(nthcdr( $n, l$ )) < length( $l$ )) = '*1>true)

```

EVENT: Disable length-of-nthcdr-goes-down.

EVENT: Enable length-of-nthcdr-goes-down.

```

DEFINITION:
table( $n, l$ )
= if  $n \simeq 0$  then  $l$ 
  elseif listp( $l$ ) then cons(getseg('0,  $n, l$ ), table( $n, \text{nthcdr}(n, l)$ ))
  else  $\text{'nil}$  endif

```

EVENT: Disable length-of-nthcdr-goes-down.

```

DEFINITION:
flatten( $table$ )
= if listp( $table$ ) then append(car( $table$ ), flatten(cdr( $table$ )))
  else  $\text{'nil}$  endif

```

```

THEOREM: plistp-table
plistp( $l$ ) → plistp(table( $n, l$ ))

```

EVENT: Disable plistp-table.

THEOREM: length-table

$$\begin{aligned} & ((n \neq 0) \wedge ((\text{length}(l) \bmod n) = '0)) \\ \rightarrow & (\text{length}(\text{table}(n, l)) = (\text{length}(l) \div n)) \end{aligned}$$

EVENT: Disable length-table.

THEOREM: tablep-table

$$(n \neq 0) \rightarrow \text{tablep}(n, \text{table}(n, l))$$

EVENT: Disable tablep-table.

THEOREM: length-getnth-from-table

$$(\text{tablep}(n, l) \wedge (i < \text{length}(l))) \rightarrow (\text{length}(\text{getnth}(i, l)) = n)$$

THEOREM: plistp-getnth-from-table

$$(\text{tablep}(n, l) \wedge (i < \text{length}(l))) \rightarrow \text{plistp}(\text{getnth}(i, l))$$

THEOREM: length-getnth-table

$$\begin{aligned} & ((n \neq 0) \wedge (((\text{length}(l) \bmod n) = '0) \wedge (i < (\text{length}(l) \div n)))) \\ \rightarrow & (\text{length}(\text{getnth}(i, \text{table}(n, l))) = n) \end{aligned}$$

EVENT: Disable length-getnth-table.

THEOREM: plistp-getnth-table

$$\begin{aligned} & ((n \neq 0) \wedge (((\text{length}(l) \bmod n) = '0) \wedge (i < (\text{length}(l) \div n)))) \\ \rightarrow & \text{plistp}(\text{getnth}(i, \text{table}(n, l))) \end{aligned}$$

EVENT: Disable plistp-getnth-table.

THEOREM: getseg-equals-firstn-nthcdr

$$\begin{aligned} & (\text{if } \text{length}(l) < n \text{ then } '*1*false \\ & \quad \text{else } '*1*true \text{ endif} \\ & \wedge \text{if } (\text{length}(l) - n) < k \text{ then } '*1*false \\ & \quad \text{else } '*1*true \text{ endif}) \\ \rightarrow & (\text{getseg}(n, k, l) = \text{firstn}(k, \text{nthcdr}(n, l))) \end{aligned}$$

EVENT: Disable getseg-equals-firstn-nthcdr.

THEOREM: getseg-for-length

$$(\text{plistp}(l) \wedge (k = \text{length}(l))) \rightarrow (\text{getseg}('0, k, l) = l)$$

EVENT: Disable getseg-for-length.

THEOREM: plistp-flatten
plistp (flatten (table))

EVENT: Disable plistp-flatten.

THEOREM: length-flatten
 $((n \neq 0) \wedge \text{tablep}(n, l)) \rightarrow (\text{length}(\text{flatten}(l)) = (\text{length}(l) * n))$

EVENT: Disable length-flatten.

THEOREM: table-append
 $((n \neq 0) \wedge ((\text{length}(a) \bmod n) = '0))$
 $\rightarrow (\text{table}(n, \text{append}(a, b)) = \text{append}(\text{table}(n, a), \text{table}(n, b)))$

EVENT: Disable table-append.

THEOREM: nthcdr-for-length
plistp (l) \rightarrow (nthcdr (length (l), l) = 'nil)

EVENT: Disable nthcdr-for-length.

THEOREM: table-for-length
 $(\text{plistp}(l) \wedge (\text{length}(l) \neq 0)) \rightarrow (\text{table}(\text{length}(l), l) = \text{cons}(l, 'nil))$

EVENT: Disable table-for-length.

THEOREM: table-flatten
 $((n \neq 0) \wedge \text{tablep}(n, \text{table})) \rightarrow (\text{table}(n, \text{flatten}(\text{table})) = \text{table})$

EVENT: Disable table-flatten.

THEOREM: listp-table
 $(n \neq 0) \rightarrow (\text{listp}(\text{table}(n, l)) = \text{listp}(l))$

EVENT: Disable listp-table.

DEFINITION:
integer-nthcdr-induction (i, n, l)
= **if** $n \simeq 0$ **then** '0

```

elseif listp(l)
then if i  $\simeq$  0 then '0
      else integer-nthcdr-induction(i - 1, n, nthcdr(n, l)) endif
else '0 endif

```

THEOREM: getnth-nthcdr
 $\text{getnth}(i, \text{nthcdr}(n, l)) = \text{getnth}(n + i, l)$

EVENT: Disable getnth-nthcdr.

THEOREM: getseg-nthcdr-proof
 $(i \in \mathbf{N}) \rightarrow (\text{getseg}(i, k, \text{nthcdr}(n, l)) = \text{getseg}(n + i, k, l))$

EVENT: Disable getseg-nthcdr-proof.

THEOREM: getseg-nthcdr
 $\text{getseg}(i, k, \text{nthcdr}(n, l)) = \text{getseg}(n + i, k, l)$

EVENT: Disable getseg-nthcdr.

THEOREM: getnth-table
 $((i * n) < \text{length}(l) \wedge (n \neq 0))$
 $\rightarrow (\text{getnth}(i, \text{table}(n, l)) = \text{getseg}(i * n, n, l))$

EVENT: Disable getnth-table.

THEOREM: getnth-table-inverse
 $((i * n) < \text{length}(l) \wedge (n \neq 0))$
 $\rightarrow (\text{getseg}(i * n, n, l) = \text{getnth}(i, \text{table}(n, l)))$

EVENT: Disable getnth-table-inverse.

THEOREM: rewrite-getnth-from-table
 $((n \neq 0)$
 $\wedge ((i < n)$
 $\wedge (\text{if } \text{length}(l) < (n + (n * j)) \text{ then } '*1*false$
 $\text{else } '*1*true \text{ endif}))$
 $\rightarrow (\text{getnth}(i + (n * j), l) = \text{getnth}(i, \text{getnth}(j, \text{table}(n, l))))$

EVENT: Disable rewrite-getnth-from-table.

THEOREM: rewrite-getnth-from-table0

```

((n ≠ 0)
 ∧  if length(l) < (n + (n * j)) then '*1*false
    else '*1*true endif)
→  (getnth(n * j, l) = getnth('0, getnth(j, table(n, l))))

```

EVENT: Disable rewrite-getnth-from-table0.

THEOREM: nthcdr-nthcdr
 $\text{nthcdr}(i, \text{nthcdr}(j, l)) = \text{nthcdr}(j + i, l)$

EVENT: Disable nthcdr-nthcdr.

THEOREM: putnth-nthcdr
 $\text{putnth}(v, i, \text{nthcdr}(n, l)) = \text{nthcdr}(n, \text{putnth}(v, n + i, l))$

EVENT: Disable putnth-nthcdr.

THEOREM: putseg-nthcdr-proof
 $(i \in \mathbf{N}) \rightarrow (\text{putseg}(s, i, \text{nthcdr}(n, l)) = \text{nthcdr}(n, \text{putseg}(s, n + i, l)))$

EVENT: Disable putseg-nthcdr-proof.

THEOREM: putseg-nthcdr
 $\text{putseg}(s, i, \text{nthcdr}(n, l)) = \text{nthcdr}(n, \text{putseg}(s, n + i, l))$

EVENT: Disable putseg-nthcdr.

THEOREM: nthcdr-putnth
 $(n < i) \rightarrow (\text{nthcdr}(i, \text{putnth}(v, n, l)) = \text{nthcdr}(i, l))$

EVENT: Disable nthcdr-putnth.

THEOREM: nthcdr-putseg
if $i < (n + \text{length}(s))$ **then** '*1*false
else '*1*true **endif**
→ $(\text{nthcdr}(i, \text{putseg}(s, n, l)) = \text{nthcdr}(i, l))$

EVENT: Disable nthcdr-putseg.

THEOREM: table-putseg-subcase
 $(\text{plistp}(s)$
 $\wedge (\text{length}(s) \neq 0)$

\wedge **if** length(l) < length(s) **then** '*1*false
else '*1*true **endif**)
 \rightarrow (table(length(s), putseg(s , '0, l))
= cons(s , table(length(s), nthcdr(length(s), l))))

EVENT: Disable table-putseg-subcase.

THEOREM: table-putseg
(plistp(s)
 \wedge ((length(s) = n)
 \wedge (($n \neq 0$)
 \wedge **if** length(l) < (($n * i$) + length(s))
then '*1*false
else '*1*true **endif**))
 \rightarrow (table(n , putseg(s , $n * i$, l)) = putnth(s , i , table(n , l)))

EVENT: Disable table-putseg.

THEOREM: table-putseg-inverse
(plistp(s)
 \wedge ((length(s) = n)
 \wedge (($n \neq 0$)
 \wedge **if** length(l) < (($n * i$) + length(s))
then '*1*false
else '*1*true **endif**))
 \rightarrow (putnth(s , i , table(n , l)) = table(n , putseg(s , $n * i$, l)))

EVENT: Disable table-putseg-inverse.

THEOREM: table-putnth
(($n \neq 0$)
 \wedge (($i \in \mathbf{N}$) \wedge (($i < \text{length}(l)$) \wedge ((length(l) **mod** n) = '0))))
 \rightarrow (table(n , putnth(v , i , l))
= putnth(putnth(v , i **mod** n , getnth($i \div n$, table(n , l))),
 $i \div n$,
table(n , l)))

EVENT: Disable table-putnth.

DEFINITION:
finite-number-tablep(n , l , lub)
= **if** listp(l)
then plistp(car(l))

$$\begin{aligned} & \wedge (\text{finite-number-listp}(\text{car}(l), \text{lub}) \\ & \quad \wedge ((\text{length}(\text{car}(l)) = n) \\ & \quad \quad \wedge \text{finite-number-tablep}(n, \text{cdr}(l), \text{lub}))) \\ \text{else } l = \text{'nil} \text{ endif} \end{aligned}$$

THEOREM: a-finite-number-table-is-a-table
 $\text{finite-number-tablep}(n, l, \text{lub}) \rightarrow \text{tablep}(n, l)$

EVENT: Disable a-finite-number-table-is-a-table.

THEOREM: finite-number-listp-nil
 $\text{finite-number-listp}(\text{'nil}, \text{lub})$

EVENT: Disable finite-number-listp-nil.

THEOREM: finite-number-listp-flatten
 $((n \neq 0) \wedge \text{finite-number-tablep}(n, l, \text{lub}))$
 $\rightarrow \text{finite-number-listp}(\text{flatten}(l), \text{lub})$

EVENT: Disable finite-number-listp-flatten.

THEOREM: finite-number-tablep-list
 $(\text{plistp}(l) \wedge ((\text{length}(l) = n) \wedge \text{finite-number-listp}(l, \text{lub})))$
 $\rightarrow \text{finite-number-tablep}(n, \text{cons}(l, \text{'nil}), \text{lub})$

EVENT: Disable finite-number-tablep-list.

THEOREM: finite-number-tablep-append
 $(\text{finite-number-tablep}(n, l1, \text{lub}) \wedge \text{finite-number-tablep}(n, l2, \text{lub}))$
 $\rightarrow \text{finite-number-tablep}(n, \text{append}(l1, l2), \text{lub})$

EVENT: Disable finite-number-tablep-append.

THEOREM: getnth-plus-1-is-cadr-nthcdr
 $\text{getnth}(\text{'1} + n, l) = \text{car}(\text{cdr}(\text{nthcdr}(n, l)))$

EVENT: Disable getnth-plus-1-is-cadr-nthcdr.

THEOREM: cadr-firstn
 $(\text{'1} < n) \rightarrow (\text{car}(\text{cdr}(\text{firstn}(n, l))) = \text{car}(\text{cdr}(l)))$

EVENT: Disable cadr-firstn.

THEOREM: getnth-plus-1-is-cadr-getseg
 $(('1 < k$
 \wedge (if length(l) < n then '*1*false
 \quad else '*1*true endif
 $\quad \wedge$ if (length(l) - n) < k then '*1*false
 $\quad \quad$ else '*1*true endif))
 \rightarrow (getnth('1 + n , l) = car(cdr(getseg(n , k , l))))

THEOREM: getnth-plus-1-is-cadr-getseg-inverse
 $(('1 < k$
 \wedge (if length(l) < n then '*1*false
 \quad else '*1*true endif
 $\quad \wedge$ if (length(l) - n) < k then '*1*false
 $\quad \quad$ else '*1*true endif))
 \rightarrow (car(cdr(getseg(n , k , l))) = getnth('1 + n , l))

EVENT: Disable getnth-plus-1-is-cadr-getseg-inverse.

THEOREM: firstn-fixlength
(if $n < i$ then '*1*false
 \quad else '*1*true endif
 \wedge if length(l) < i then '*1*false
 \quad else '*1*true endif)
 \rightarrow (firstn(i , fixlength(n , l , default))) = firstn(i , l)

EVENT: Disable firstn-fixlength.

THEOREM: getseg-fixlength
(plistp(l) \wedge (($n \neq 0$) \wedge (length(l) < n)))
 \rightarrow (getseg('0, length(l), fixlength(n , l , default))
 \quad = getseg('0, length(l), l))

EVENT: Disable getseg-fixlength.

THEOREM: number-listp-finite-number-listp
finite-number-listp(l , lub) \rightarrow number-listp(l)

DEFINITION: nonlast(l) = getseg('0, length(l) - 1, l)

THEOREM: length-nonlast
length(nonlast(l)) = (length(l) - 1)

EVENT: Disable length-nonlast.

THEOREM: plistp-nonlast
plistp (nonlast (l))

EVENT: Disable plistp-nonlast.

THEOREM: finite-number-listp-nonlast
finite-number-listp (l , lub) \rightarrow finite-number-listp (nonlast (l), lub)

EVENT: Disable finite-number-listp-nonlast.

THEOREM: difference-sub1-plus-crock
 $((a \neq 0) \wedge (b \neq 0)) \rightarrow (((a + b) - 1) - a) = (b - 1)$

EVENT: Disable difference-sub1-plus-crock.

THEOREM: nonlast-append
plistp (a)
 \rightarrow (nonlast (append (a , b)))
= **if** length (b) = '0 **then** nonlast (a)
 else append (a , nonlast (b)) **endif**

EVENT: Disable nonlast-append.

DEFINITION:
incr-mod (n , lub)
= **if** $(1 + n) < lub$ **then** $1 + n$
 else '0 **endif**

DEFINITION:
decr-mod (n , lub)
= **if** $n \simeq 0$ **then** $lub - 1$
 else $n - 1$ **endif**

THEOREM: decr-mod-size
 $((n \neq 0) \wedge (a < n)) \rightarrow (\text{decr-mod} (a, n) < n)$

EVENT: Disable decr-mod-size.

THEOREM: incr-mod-equals-remainder-add1
 $(n < lub) \rightarrow (\text{incr-mod} (n, lub) = ((1 + n) \bmod lub))$

EVENT: Disable incr-mod-equals-remainder-add1.

THEOREM: incr-mod-decr-mod
 $(n < lub) \rightarrow (\text{incr-mod}(\text{decr-mod}(n, lub), lub) = \text{fix}(n))$

EVENT: Disable incr-mod-decr-mod.

THEOREM: lessp-incr-mod
 $(n < lub) \rightarrow (\text{incr-mod}(n, lub) < lub)$

EVENT: Disable lessp-incr-mod.

DEFINITION: $\text{qfirst}(table) = \text{car}(table)$

DEFINITION: $\text{enq}(item, table) = \text{append}(table, \text{cons}(item, 'nil))$

DEFINITION: $\text{deq}(table) = \text{cdr}(table)$

DEFINITION: $\text{qempty}(table) = (\text{length}(table) = '0)$

DEFINITION: $\text{qfull}(table, max) = (\text{length}(table) \not< max)$

DEFINITION: $\text{qreplace}(item, queue) = \text{enq}(item, \text{nonlast}(queue))$

THEOREM: plistp-enq
 $\text{plistp}(\text{enq}(item, l))$

EVENT: Disable plistp-enq.

THEOREM: plistp-deq
 $(\text{plistp}(queue) \wedge (\neg \text{qempty}(queue))) \rightarrow \text{plistp}(\text{deq}(queue))$

EVENT: Disable plistp-deq.

THEOREM: plistp-qreplace
 $\text{plistp}(\text{qreplace}(item, queue))$

EVENT: Disable plistp-qreplace.

THEOREM: length-enq
 $\text{length}(\text{enq}(item, queue)) = (1 + \text{length}(queue))$

EVENT: Disable length-enq.

THEOREM: length-non-empty-queue
 $(\neg \text{qempty}(queue)) \rightarrow (\text{length}(queue) \neq '0)$

EVENT: Disable length-non-empty-queue.

THEOREM: length-deq
 $(\neg \text{qempty}(queue)) \rightarrow (\text{length}(\text{deq}(queue)) = (\text{length}(queue) - 1))$

EVENT: Disable length-deq.

THEOREM: length-qreplace
 $(\neg \text{qempty}(queue)) \rightarrow (\text{length}(\text{qreplace}(item, queue)) = \text{length}(queue))$

EVENT: Disable length-qreplace.

THEOREM: not-qempty-enq
 $\neg \text{qempty}(\text{enq}(item, queue))$

EVENT: Disable not-qempty-enq.

THEOREM: not-qfullp-deq
(if $n < \text{length}(queue)$ **then** **'*1*false**
else **'*1>true** **endif**
 $\wedge (\neg \text{qempty}(queue))$
 $\rightarrow (\neg \text{qfullp}(\text{deq}(queue), n))$

EVENT: Disable not-qfullp-deq.

THEOREM: qfirst-enq
 $(\neg \text{qempty}(queue)) \rightarrow (\text{qfirst}(\text{enq}(x, queue)) = \text{qfirst}(queue))$

EVENT: Disable qfirst-enq.

THEOREM: finite-number-listp-enq
 $(\text{finite-number-listp}(queue, lub) \wedge \text{finite-numberp}(item, lub))$
 $\rightarrow \text{finite-number-listp}(\text{enq}(item, queue), lub)$

EVENT: Disable finite-number-listp-enq.

THEOREM: finite-number-listp-deq
 $(\text{finite-number-listp}(queue, lub) \wedge (\neg \text{qempty}(queue)))$
 $\rightarrow \text{finite-number-listp}(\text{deq}(queue), lub)$

EVENT: Disable finite-number-listp-deq.

THEOREM: finite-number-listp-qreplace
(finite-number-listp (*queue*, *lub*) \wedge finite-numberp (*item*, *lub*))
 \rightarrow finite-number-listp (qreplace (*item*, *queue*), *lub*)

EVENT: Disable finite-number-listp-qreplace.

THEOREM: finite-numberp-qfirst
(finite-number-listp (*queue*, *lub*) \wedge (\neg qempty (*queue*)))
 \rightarrow finite-numberp (qfirst (*queue*), *lub*)

EVENT: Disable finite-numberp-qfirst.

THEOREM: numberp-qfirst
(\neg qempty (*queue*) \wedge number-listp (*queue*)) \rightarrow (qfirst (*queue*) $\in \mathbf{N}$)

EVENT: Disable numberp-qfirst.

THEOREM: deq-is-remove
(setp (*l*) \wedge ('0 < length (*l*))) \rightarrow (deq (*l*) = remove (qfirst (*l*), *l*))

EVENT: Disable deq-is-remove.

THEOREM: member-qfirst-queue
(\neg qempty (*queue*)) \rightarrow (qfirst (*queue*) \in *queue*)

EVENT: Disable member-qfirst-queue.

THEOREM: permutation-enq-deq
(\neg qempty (*queue*)) \rightarrow permutation (enq (qfirst (*queue*), deq (*queue*)), *queue*)

EVENT: Disable permutation-enq-deq.

DEFINITION: QHEAD-FIELD = '0

DEFINITION: QTAIL-FIELD = '1

DEFINITION: QCURRLENGTH-FIELD = '2

DEFINITION: QMAXLENGTH-FIELD = '3

DEFINITION: QARRAY-FIELD = '4

DEFINITION:

$\text{delta}(a, b, \text{max})$
= **if if** $a < b$ **then** $'1*$ **false**
 else $'1*$ **true** **endif** **then** $(\text{max} - a) + b$
 else $b - a$ **endif**

THEOREM: delta-a-a

$(a < \text{max}) \rightarrow (\text{delta}(a, a, \text{max}) = \text{max})$

EVENT: Disable delta-a-a.

THEOREM: delta-incr-mod-a

$((a < \text{max}) \wedge (b < \text{max}))$
 $\rightarrow (\text{delta}(\text{incr-mod}(a, \text{max}), b, \text{max})$
 = **if** $\text{delta}(a, b, \text{max}) = '1$ **then** max
 else $\text{delta}(a, b, \text{max}) - 1$ **endif**)

EVENT: Disable delta-incr-mod-a.

THEOREM: delta-incr-mod-b

$((a < \text{max}) \wedge (b < \text{max}))$
 $\rightarrow (\text{delta}(a, \text{incr-mod}(b, \text{max}), \text{max})$
 = **if** $\text{fix}(a) = \text{fix}(b)$ **then** $'1$
 else $1 + \text{delta}(a, b, \text{max})$ **endif**)

EVENT: Disable delta-incr-mod-b.

THEOREM: delta-equals-max-iff-a-equals-b

$((a \in \mathbf{N}) \wedge ((b \in \mathbf{N}) \wedge ((a < \text{max}) \wedge (b < \text{max}))))$
 $\rightarrow ((\text{delta}(a, b, \text{max}) = \text{max}) = (a = b))$

EVENT: Disable delta-equals-max-iff-a-equals-b.

THEOREM: leq-delta

$((a \in \mathbf{N}) \wedge ((b \in \mathbf{N}) \wedge ((a < \text{max}) \wedge (b < \text{max}))))$
 \rightarrow **if** $\text{max} < \text{delta}(a, b, \text{max})$ **then** $'1*$ **false**
 else $'1*$ **true** **endif**

DEFINITION:

$\text{array-qindex-relation}(\text{queue})$
= $(\text{delta}(\text{getnth}('0, \text{queue}), \text{getnth}('1, \text{queue}), \text{getnth}('3, \text{queue}))$
 = **if** $\text{getnth}('2, \text{queue}) \simeq 0$ **then** $\text{getnth}('3, \text{queue})$
 else $\text{getnth}('2, \text{queue})$ **endif**)

DEFINITION:

$$\begin{aligned} & \text{array-queuep}(queue) \\ = & \text{ (plistp}(queue) \\ & \wedge ((\text{length}(queue) = ('4 + \text{getnth}('3, queue))) \\ & \wedge ((\text{getnth}('0, queue) \in \mathbf{N}) \\ & \wedge ((\text{getnth}('1, queue) \in \mathbf{N}) \\ & \wedge ((\text{getnth}('2, queue) \in \mathbf{N}) \\ & \wedge ((\text{getnth}('3, queue) \neq 0) \\ & \wedge ((\text{getnth}('0, queue) \\ & \quad < \text{getnth}('3, \\ & \quad \quad queue)) \\ & \wedge ((\text{getnth}('1, \\ & \quad \quad queue) \\ & \quad < \text{getnth}('3, \\ & \quad \quad \quad queue)) \\ & \wedge ((\text{getnth}('2, \\ & \quad \quad queue) \\ & \quad < (1 + \text{getnth}('3, \\ & \quad \quad \quad \quad queue))) \\ & \wedge \text{array-qindex-relation}(queue))))))))) \end{aligned}$$

THEOREM: plistp-array-queue
 $\text{array-queuep}(queue) \rightarrow \text{plistp}(queue)$

EVENT: Disable plistp-array-queue.

THEOREM: length-array-queue
 $\text{array-queuep}(queue) \rightarrow (\text{length}(queue) = ('4 + \text{getnth}('3, queue)))$

EVENT: Disable length-array-queue.

THEOREM: numberp-qhead
 $\text{array-queuep}(queue) \rightarrow (\text{getnth}('0, queue) \in \mathbf{N})$

EVENT: Disable numberp-qhead.

THEOREM: numberp-qtail
 $\text{array-queuep}(queue) \rightarrow (\text{getnth}('1, queue) \in \mathbf{N})$

EVENT: Disable numberp-qtail.

THEOREM: numberp-qcurrlength
 $\text{array-queuep}(queue) \rightarrow (\text{getnth}('2, queue) \in \mathbf{N})$

EVENT: Disable numberp-qcurrenlength.

THEOREM: numberp-qmaxlength
array-queuep(*queue*) \rightarrow (getnth('3, *queue*) $\in \mathbf{N}$)

EVENT: Disable numberp-qmaxlength.

THEOREM: qmaxlength-non-zero
array-queuep(*queue*) \rightarrow (getnth('3, *queue*) \neq '0)

EVENT: Disable qmaxlength-non-zero.

THEOREM: qmaxlength-greater-than-zero
array-queuep(*queue*) \rightarrow ('0 < getnth('3, *queue*))

EVENT: Disable qmaxlength-greater-than-zero.

THEOREM: lessp-qhead-qmaxlength
array-queuep(*queue*) \rightarrow (getnth('0, *queue*) < getnth('3, *queue*))

EVENT: Disable lessp-qhead-qmaxlength.

THEOREM: lessp-qtail-qmaxlength
array-queuep(*queue*) \rightarrow (getnth('1, *queue*) < getnth('3, *queue*))

EVENT: Disable lessp-qtail-qmaxlength.

THEOREM: lessp-qcurrenlength-add1-qmaxlength
array-queuep(*queue*) \rightarrow (getnth('2, *queue*) < (1 + getnth('3, *queue*)))

EVENT: Disable lessp-qcurrenlength-add1-qmaxlength.

THEOREM: plistp-array-qarray
plistp(getseg('4, length(*queue*) - '4, *queue*))

EVENT: Disable plistp-array-qarray.

THEOREM: length-array-qarray
array-queuep(*queue*)
 \rightarrow (length(getseg('4, length(*queue*) - '4, *queue*)) = getnth('3, *queue*))

EVENT: Disable length-array-qarray.

DEFINITION:

$$\begin{aligned} & \text{array-enq}(item, queue) \\ = & \text{putnth}(\text{incr-mod}(\text{getnth}('1, queue), \text{getnth}('3, queue)), \\ & \quad '1, \\ & \quad \text{putnth}(1 + \text{getnth}('2, queue), \\ & \quad \quad '2, \\ & \quad \quad \text{putnth}(item, '4 + \text{getnth}('1, queue), queue))) \end{aligned}$$

DEFINITION:

$$\begin{aligned} & \text{array-deq}(queue) \\ = & \text{putnth}(\text{incr-mod}(\text{getnth}('0, queue), \text{getnth}('3, queue)), \\ & \quad '0, \\ & \quad \text{putnth}(\text{getnth}('2, queue) - 1, '2, queue)) \end{aligned}$$

DEFINITION:

$$\text{array-qfirst}(queue) = \text{getnth}('4 + \text{getnth}('0, queue), queue)$$

DEFINITION:

$$\text{array-qfullp}(queue) = (\text{getnth}('2, queue) = \text{getnth}('3, queue))$$

$$\text{DEFINITION: } \text{array-qempty}(queue) = (\text{getnth}('2, queue) \simeq 0)$$

THEOREM: length-array-enq

$$\text{length}(\text{array-enq}(item, queue)) = \text{length}(queue)$$

EVENT: Disable length-array-enq.

THEOREM: plistp-array-enq

$$\text{plistp}(queue) \rightarrow \text{plistp}(\text{array-enq}(item, queue))$$

EVENT: Disable plistp-array-enq.

THEOREM: array-qhead-array-enq

$$\text{getnth}('0, \text{array-enq}(item, queue)) = \text{getnth}('0, queue)$$

EVENT: Disable array-qhead-array-enq.

THEOREM: array-qtail-array-enq

$$\begin{aligned} & \text{array-queuep}(queue) \\ \rightarrow & (\text{getnth}('1, \text{array-enq}(item, queue)) \\ & = \text{incr-mod}(\text{getnth}('1, queue), \text{getnth}('3, queue))) \end{aligned}$$

EVENT: Disable array-qtail-array-enq.

THEOREM: array-qcurrenlength-array-enq
array-queuep (*queue*)
 \rightarrow (getnth ('2, array-enq (*item*, *queue*)) = (1 + getnth ('2, *queue*)))

EVENT: Disable array-qcurrenlength-array-enq.

THEOREM: array-qmaxlength-array-enq
getnth ('3, array-enq (*item*, *queue*)) = getnth ('3, *queue*)

EVENT: Disable array-qmaxlength-array-enq.

THEOREM: array-qarray-array-enq
array-queuep (*queue*)
 \rightarrow (getseg ('4, getnth ('3, *queue*), array-enq (*item*, *queue*))
= putnth (*item*,
getnth ('1, *queue*),
getseg ('4, getnth ('3, *queue*), *queue*)))

EVENT: Disable array-qarray-array-enq.

THEOREM: array-queuep-array-enq
(array-queuep (*queue*) \wedge (\neg array-qfullp (*queue*)))
 \rightarrow array-queuep (array-enq (*item*, *queue*))

EVENT: Disable array-queuep-array-enq.

THEOREM: length-array-deq
length (array-deq (*queue*)) = length (*queue*)

EVENT: Disable length-array-deq.

THEOREM: plistp-array-deq
plistp (*queue*) \rightarrow plistp (array-deq (*queue*))

EVENT: Disable plistp-array-deq.

THEOREM: array-qhead-array-deq
array-queuep (*queue*)
 \rightarrow (getnth ('0, array-deq (*queue*))
= incr-mod (getnth ('0, *queue*), getnth ('3, *queue*)))

EVENT: Disable array-qhead-array-deq.

THEOREM: array-qtail-array-deq
 $\text{getnth}('1, \text{array-deq}(queue)) = \text{getnth}('1, queue)$

EVENT: Disable array-qtail-array-deq.

THEOREM: array-qcurrenlength-array-deq
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{getnth}('2, \text{array-deq}(queue)) = (\text{getnth}('2, queue) - 1))$

EVENT: Disable array-qcurrenlength-array-deq.

THEOREM: array-qmaxlength-array-deq
 $\text{getnth}('3, \text{array-deq}(queue)) = \text{getnth}('3, queue)$

EVENT: Disable array-qmaxlength-array-deq.

THEOREM: array-qarray-array-deq
 $\text{getseg}('4, \text{getnth}('3, queue), \text{array-deq}(queue))$
 $= \text{getseg}('4, \text{getnth}('3, queue), queue)$

EVENT: Disable array-qarray-array-deq.

THEOREM: array-queuep-array-deq
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow \text{array-queuep}(\text{array-deq}(queue))$

EVENT: Disable array-queuep-array-deq.

THEOREM: not-array-qempty-array-enq
 $\text{array-queuep}(queue) \rightarrow (\neg \text{array-qempty}(\text{array-enq}(item, queue)))$

EVENT: Disable not-array-qempty-array-enq.

THEOREM: not-array-qfullp-array-deq
 $\text{array-queuep}(queue) \rightarrow (\neg \text{array-qfullp}(\text{array-deq}(queue)))$

EVENT: Disable not-array-qfullp-array-deq.

THEOREM: array-qcurrenlength-in-a-full-queue
 $(\text{array-queuep}(queue) \wedge \text{array-qfullp}(queue))$
 $\rightarrow (\text{getnth}('2, queue) = \text{getnth}('3, queue))$

EVENT: Disable array-qcurrenlength-in-a-full-queue.

THEOREM: array-qcurrenlength-in-a-non-full-queue
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qfullp}(queue)))$
 $\rightarrow (\text{getnth}(2, queue) < \text{getnth}(3, queue))$

EVENT: Disable array-qcurrenlength-in-a-non-full-queue.

THEOREM: array-qcurrenlength-in-an-empty-queue
 $(\text{array-queuep}(queue) \wedge \text{array-qemptyyp}(queue))$
 $\rightarrow (\text{getnth}(2, queue) = 0)$

EVENT: Disable array-qcurrenlength-in-an-empty-queue.

THEOREM: array-qcurrenlength-in-a-non-empty-queue
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qemptyyp}(queue)))$
 $\rightarrow (\text{getnth}(2, queue) \neq 0)$

EVENT: Disable array-qcurrenlength-in-a-non-empty-queue.

DEFINITION:
 $\text{array-nonlast}(queue)$
 $= \text{putnth}(\text{decr-mod}(\text{getnth}(1, queue), \text{getnth}(3, queue)),$
 $\quad 1,$
 $\quad \text{putnth}(\text{getnth}(2, queue) - 1, 2, queue))$

THEOREM: length-array-nonlast
 $\text{length}(\text{array-nonlast}(queue)) = \text{length}(queue)$

EVENT: Disable length-array-nonlast.

THEOREM: plistp-array-nonlast
 $\text{plistp}(queue) \rightarrow \text{plistp}(\text{array-nonlast}(queue))$

EVENT: Disable plistp-array-nonlast.

THEOREM: array-qhead-array-nonlast
 $\text{getnth}(0, \text{array-nonlast}(queue)) = \text{getnth}(0, queue)$

EVENT: Disable array-qhead-array-nonlast.

THEOREM: array-qtail-array-nonlast
array-queuep (*queue*)
→ (getnth ('1, array-nonlast (*queue*))
= decr-mod (getnth ('1, *queue*), getnth ('3, *queue*)))

EVENT: Disable array-qtail-array-nonlast.

THEOREM: array-qcurrenlength-array-nonlast
array-queuep (*queue*)
→ (getnth ('2, array-nonlast (*queue*)) = (getnth ('2, *queue*) - 1))

EVENT: Disable array-qcurrenlength-array-nonlast.

THEOREM: array-qmaxlength-array-nonlast
getnth ('3, array-nonlast (*queue*)) = getnth ('3, *queue*)

EVENT: Disable array-qmaxlength-array-nonlast.

THEOREM: array-qarray-array-nonlast
getseg ('4, getnth ('3, *queue*), array-nonlast (*queue*))
= getseg ('4, getnth ('3, *queue*), *queue*)

EVENT: Disable array-qarray-array-nonlast.

THEOREM: delta-sub1-b-case1
((*a* ∈ **N**)
∧ ((*b* ∈ **N**)
∧ ((*b* ≠ '0)
∧ ((*a* < *max*)
∧ ((*b* < *max*)
∧ ((delta (*a*, *b*, *max*) - 1) ≠ '0))))))
→ (delta (*a*, *b* - 1, *max*) = (delta (*a*, *b*, *max*) - 1))

EVENT: Disable delta-sub1-b-case1.

THEOREM: delta-sub1-b-case2
((*a* ∈ **N**) ∧ ((*a* < *max*) ∧ ((delta (*a*, '0, *max*) - 1) ≠ '0)))
→ (delta (*a*, *max* - 1, *max*) = (delta (*a*, '0, *max*) - 1))

EVENT: Disable delta-sub1-b-case2.

```

(PROVE-LEMMA
ARRAY-QUEUEP-ARRAY-NONLAST-CROCK1
(REWRITE)
(IMPLIES
(AND
(PLISTP QUEUE)
(AND
(EQUAL (LENGTH QUEUE)
(PLUS '4 (GETNTH '3 QUEUE)))
(AND
(NUMBERP (GETNTH '0 QUEUE))
(AND
(NUMBERP (GETNTH '1 QUEUE))
(AND
(NOT (EQUAL (GETNTH '3 QUEUE) '0))
(AND (NUMBERP (GETNTH '3 QUEUE))
(AND (LESSP (GETNTH '0 QUEUE)
(GETNTH '3 QUEUE))
(AND (LESSP (GETNTH '1 QUEUE)
(GETNTH '3 QUEUE))
(AND (LESSP (SUB1 (GETNTH '2 QUEUE))
(GETNTH '3 QUEUE))
(AND (EQUAL (DELTA (GETNTH '0 QUEUE)
(GETNTH '1 QUEUE)
(GETNTH '3 QUEUE))
(GETNTH '2 QUEUE))
(AND (NOT (EQUAL (GETNTH '2 QUEUE) '0))
(AND (EQUAL (SUB1 (GETNTH '2 QUEUE)) '0)
(NOT (EQUAL (GETNTH '1 QUEUE)
'0))))))))))))))
(EQUAL (GETNTH '0 QUEUE)
(SUB1 (GETNTH '1 QUEUE))))
((ENABLE DELTA)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T)))

```

EVENT: Disable array-queuep-array-nonlast-crock1.

```

(PROVE-LEMMA
ARRAY-QUEUEP-ARRAY-NONLAST-CROCK2
(REWRITE)

```

```

(IMPLIES
(AND
(PLISTP QUEUE)
(AND
(EQUAL (LENGTH QUEUE)
(PLUS '4 (GETNTH '3 QUEUE)))
(AND
(NUMBERP (GETNTH '0 QUEUE))
(AND
(NUMBERP (GETNTH '1 QUEUE))
(AND
(NOT (EQUAL (GETNTH '3 QUEUE) '0))
(AND (NUMBERP (GETNTH '3 QUEUE))
(AND (LESSP (GETNTH '0 QUEUE)
(GETNTH '3 QUEUE))
(AND (LESSP (GETNTH '1 QUEUE)
(GETNTH '3 QUEUE))
(AND (LESSP (SUB1 (GETNTH '2 QUEUE))
(GETNTH '3 QUEUE))
(AND (EQUAL (DELTA (GETNTH '0 QUEUE)
(GETNTH '1 QUEUE)
(GETNTH '3 QUEUE))
(GETNTH '2 QUEUE))
(AND (NOT (EQUAL (GETNTH '2 QUEUE) '0))
(AND (EQUAL (SUB1 (GETNTH '2 QUEUE)) '0)
(EQUAL (GETNTH '1 QUEUE)
'0))))))))))))))
(EQUAL (GETNTH '0 QUEUE)
(SUB1 (GETNTH '3 QUEUE))))
((ENABLE LENGTH DELTA)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable array-queuep-array-nonlast-crock2.

THEOREM: array-queuep-array-nonlast
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow \text{array-queuep}(\text{array-nonlast}(queue))$

EVENT: Disable array-queuep-array-nonlast.

DEFINITION:
 $\text{array-qreplace}(item, queue) = \text{array-enq}(item, \text{array-nonlast}(queue))$

THEOREM: length-array-qreplace
 $\text{length}(\text{array-qreplace}(item, queue)) = \text{length}(queue)$

EVENT: Disable length-array-qreplace.

THEOREM: plistp-array-qreplace
 $\text{plistp}(queue) \rightarrow \text{plistp}(\text{array-qreplace}(item, queue))$

EVENT: Disable plistp-array-qreplace.

THEOREM: array-qhead-array-qreplace
 $\text{getnth}('0, \text{array-qreplace}(item, queue)) = \text{getnth}('0, queue)$

EVENT: Disable array-qhead-array-qreplace.

THEOREM: array-qtail-array-qreplace
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{getnth}('1, \text{array-qreplace}(item, queue)) = \text{getnth}('1, queue))$

EVENT: Disable array-qtail-array-qreplace.

THEOREM: array-qcurrlength-array-qreplace
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{getnth}('2, \text{array-qreplace}(item, queue)) = \text{getnth}('2, queue))$

EVENT: Disable array-qcurrlength-array-qreplace.

THEOREM: array-qmaxlength-array-qreplace
 $\text{getnth}('3, \text{array-qreplace}(item, queue)) = \text{getnth}('3, queue)$

EVENT: Disable array-qmaxlength-array-qreplace.

THEOREM: array-qarray-array-qreplace
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{getseg}('4, \text{getnth}('3, queue), \text{array-qreplace}(item, queue))$
 $= \text{putnth}(item,$
 $\quad \text{decr-mod}(\text{getnth}('1, queue), \text{getnth}('3, queue)),$
 $\quad \text{getseg}('4, \text{getnth}('3, queue), queue))$

EVENT: Disable array-qarray-array-qreplace.

THEOREM: not-array-qfullp-array-nonlast
 $\text{array-queuep}(queue) \rightarrow (\neg \text{array-qfullp}(\text{array-nonlast}(queue)))$

EVENT: Disable not-array-qfullp-array-nonlast.

THEOREM: array-queuep-array-qreplace
(array-queuep (*queue*) \wedge (\neg array-qempty (*queue*)))
 \rightarrow array-queuep (array-qreplace (*item*, *queue*))

EVENT: Disable array-queuep-array-qreplace.

EVENT: Let us define the theory *array-queue-facts* to consist of the following events: array-qcurrlength-in-an-empty-queue, array-qcurrlength-in-a-non-empty-queue, array-qcurrlength-in-a-non-full-queue, array-qcurrlength-in-a-full-queue, not-array-qfullp-array-deq, not-array-qempty-array-enq, array-queuep-array-qreplace, array-qmaxlength-array-qreplace, array-qcurrlength-array-qreplace, array-qtail-array-qreplace, array-qhead-array-qreplace, plistp-array-qreplace, length-array-qreplace, array-queuep-array-deq, array-qmaxlength-array-deq, array-qcurrlength-array-deq, array-qtail-array-deq, array-qhead-array-deq, plistp-array-deq, length-array-deq, array-queuep-array-enq, array-qmaxlength-array-enq, array-qcurrlength-array-enq, array-qtail-array-enq, array-qhead-array-enq, plistp-array-enq, length-array-enq, length-array-qarray, plistp-array-qarray, lessp-qcurrlength-add1-qmaxlength, lessp-qtail-qmaxlength, lessp-qhead-qmaxlength, qmaxlength-greater-than-zero, qmaxlength-non-zero, numberp-qmaxlength, numberp-qcurrlength, numberp-qtail, numberp-qhead.

THEOREM: array-qfullp-implies-not-array-qempty
(array-queuep (*queue*) \wedge array-qfullp (*queue*))
 \rightarrow (\neg array-qempty (*queue*))

THEOREM: array-qhead-equals-array-qtail-in-an-empty-queue
(array-queuep (*queue*) \wedge array-qempty (*queue*))
 \rightarrow (getnth ('0, *queue*) = getnth ('1, *queue*))

EVENT: Disable array-qhead-equals-array-qtail-in-an-empty-queue.

THEOREM: array-qhead-equals-array-qtail-iff-qfull-or-qempty
array-queuep (*queue*)
 \rightarrow ((getnth ('0, *queue*) = getnth ('1, *queue*))
= (array-qfullp (*queue*) \vee array-qempty (*queue*)))

EVENT: Disable array-qhead-equals-array-qtail-iff-qfull-or-qempty.

DEFINITION:
delta-segment (*a*, *b*, *s*)

```

= if  $a < b$  then '*1*false
  else '*1*true endif
  then append (getseg ( $a$ , length ( $s$ ) -  $a$ ,  $s$ ), getseg ('0,  $b$ ,  $s$ ))
  else getseg ( $a$ ,  $b - a$ ,  $s$ ) endif

```

THEOREM: delta-equals-length-delta-segment
 $((a < \text{length}(s)) \wedge (b < \text{length}(s)))$
 $\rightarrow (\text{length}(\text{delta-segment}(a, b, s)) = \text{delta}(a, b, \text{length}(s)))$

EVENT: Disable delta-equals-length-delta-segment.

THEOREM: plistp-delta-segment
 plistp (delta-segment (a , b , s))

EVENT: Disable plistp-delta-segment.

THEOREM: finite-number-listp-delta-segment
 (finite-number-listp (s , lub)
 $\wedge ((b \in \mathbf{N})$
 $\wedge (\text{if } \text{length}(s) < b \text{ then } '*1*false$
 $\text{else } '*1*true \text{ endif}$
 $\wedge ((a \in \mathbf{N})$
 $\wedge (\text{if } \text{length}(s) < a \text{ then } '*1*false$
 $\text{else } '*1*true \text{ endif})))$)
 \rightarrow finite-number-listp (delta-segment (a , b , s), lub)

EVENT: Disable finite-number-listp-delta-segment.

THEOREM: length-delta-segment
 $((a \in \mathbf{N}) \wedge ((b \in \mathbf{N}) \wedge ((a < \text{length}(s)) \wedge (b < \text{length}(s))))))$
 \rightarrow **if** $\text{length}(s) < \text{length}(\text{delta-segment}(a, b, s))$ **then** '*1*false
else '*1*true **endif**

DEFINITION:
 mapup-queue ($queue$)
 = **if** array-qempty ($queue$) **then** 'nil
else delta-segment (getnth ('0, $queue$),
 getnth ('1, $queue$),
 getseg ('4, getnth ('3, $queue$), $queue$)) **endif**

THEOREM: finite-number-listp-mapup-queue
 (array-queuep ($queue$)
 \wedge finite-number-listp (getseg ('4, getnth ('3, $queue$), $queue$), lub))
 \rightarrow finite-number-listp (mapup-queue ($queue$), lub)

EVENT: Disable finite-number-listp-mapup-queue.

THEOREM: plistp-mapup-queue
plistp (mapup-queue (*queue*))

EVENT: Disable plistp-mapup-queue.

THEOREM: length-mapup-queue
array-queuep (*queue*)
→ **if** getnth ('3, *queue*) < length (mapup-queue (*queue*)) **then** '*1*false
 else '*1*true **endif**

THEOREM: delta-segment-incr-mod-b
(plistp (*s*)
 ∧ ((*a* ∈ **N**)
 ∧ ((*b* ∈ **N**)
 ∧ ((*a* < *n*) ∧ ((*b* < *n*) ∧ (*n* = length (*s*)))))))
→ (delta-segment (*a*, incr-mod (*b*, *n*), *s*)
 = **if** *a* = *b* **then** cons (getnth (*a*, *s*), 'nil)
 else append (delta-segment (*a*, *b*, *s*),
 cons (getnth (*b*, *s*), 'nil)) **endif**)

EVENT: Disable delta-segment-incr-mod-b.

THEOREM: delta-segment-putnth-non-interference
((*a* ∈ **N**)
 ∧ ((*b* ∈ **N**)
 ∧ ((*a* < length (*s*) ∧ ((*b* < length (*s*) ∧ (*a* ≠ *b*))))))
→ (delta-segment (*a*, *b*, putnth (*v*, *b*, *s*)) = delta-segment (*a*, *b*, *s*))

EVENT: Disable delta-segment-putnth-non-interference.

THEOREM: correctness-of-array-enq
(array-queuep (*queue*) ∧ (¬ array-qfullp (*queue*)))
→ (mapup-queue (array-enq (*item*, *queue*)) = enq (*item*, mapup-queue (*queue*)))

EVENT: Disable correctness-of-array-enq.

THEOREM: delta-segment-incr-mod-a
(plistp (*s*)
 ∧ ((*a* ∈ **N**)
 ∧ ((*b* ∈ **N**))

$$\begin{aligned}
& \wedge ((a < n) \wedge ((b < n) \wedge (n = \text{length}(s)))))) \\
\rightarrow & (\text{delta-segment}(\text{incr-mod}(a, n), b, s) \\
& = \text{if } \text{delta}(a, b, n) = '1 \text{ then } \text{delta-segment}(b, b, s) \\
& \text{ else } \text{cdr}(\text{delta-segment}(a, b, s)) \text{ endif})
\end{aligned}$$

EVENT: Disable delta-segment-incr-mod-a.

THEOREM: array-qemptyp-array-deq-when-delta-is-1

$$\begin{aligned}
& (\text{array-queuep}(queue) \\
& \wedge ((\neg \text{array-qemptyp}(queue) \\
& \wedge (\text{delta}(\text{getnth}('0, queue), \text{getnth}('1, queue), \text{getnth}('3, queue)) \\
& = '1))) \\
\rightarrow & \text{array-qemptyp}(\text{array-deq}(queue))
\end{aligned}$$

EVENT: Disable array-qemptyp-array-deq-when-delta-is-1.

THEOREM: array-qemptyp-array-deq-when-delta-is-1-alternative

$$\begin{aligned}
& (\text{array-queuep}(queue) \\
& \wedge ((\neg \text{array-qemptyp}(queue)) \wedge \text{array-qemptyp}(\text{array-deq}(queue)))) \\
\rightarrow & (\text{delta}(\text{getnth}('0, queue), \text{getnth}('1, queue), \text{getnth}('3, queue)) = '1)
\end{aligned}$$

EVENT: Disable array-qemptyp-array-deq-when-delta-is-1-alternative.

THEOREM: correctness-of-array-deq-case-3

$$\begin{aligned}
& (\text{array-queuep}(queue) \\
& \wedge ((\neg \text{array-qemptyp}(queue)) \wedge \text{array-qemptyp}(\text{array-deq}(queue)))) \\
\rightarrow & (\text{cdr}(\text{delta-segment}(\text{getnth}('0, queue), \\
& \text{getnth}('1, queue), \\
& \text{getseg}('4, \text{getnth}('3, queue), queue))) \\
& = 'nil)
\end{aligned}$$

EVENT: Disable correctness-of-array-deq-case-3.

THEOREM: array-qmaxlength-must-be-1

$$\begin{aligned}
& (\text{array-queuep}(queue) \\
& \wedge (\text{array-qfullp}(queue) \wedge \text{array-qemptyp}(\text{array-deq}(queue)))) \\
\rightarrow & (\text{getnth}('3, queue) = '1)
\end{aligned}$$

EVENT: Disable array-qmaxlength-must-be-1.

THEOREM: length-delta-segment-a-a

$$\begin{aligned}
& (\text{array-queuep}(queue)
\end{aligned}$$

$$\begin{aligned}
& \wedge (\text{array-qfullp}(\text{queue}) \wedge \text{array-qempty}(\text{array-deq}(\text{queue}))) \\
\rightarrow & (\text{length}(\text{delta-segment}(\text{getnth}('0, \text{queue}), \\
& \qquad \qquad \qquad \text{getnth}('0, \text{queue}), \\
& \qquad \qquad \qquad \text{getseg}('4, \text{getnth}('3, \text{queue}), \text{queue}))) \\
& = '1)
\end{aligned}$$

EVENT: Disable length-delta-segment-a-a.

THEOREM: correctness-of-array-deq-case-1

$$\begin{aligned}
& (\text{array-queuep}(\text{queue}) \\
& \wedge (\text{array-qfullp}(\text{queue}) \wedge \text{array-qempty}(\text{array-deq}(\text{queue}))) \\
\rightarrow & (\text{cdr}(\text{delta-segment}(\text{getnth}('0, \text{queue}), \\
& \qquad \qquad \qquad \text{getnth}('0, \text{queue}), \\
& \qquad \qquad \qquad \text{getseg}('4, \text{getnth}('3, \text{queue}), \text{queue}))) \\
& = 'nil)
\end{aligned}$$

EVENT: Disable correctness-of-array-deq-case-1.

THEOREM: correctness-of-array-deq

$$\begin{aligned}
& (\text{array-queuep}(\text{queue}) \wedge (\neg \text{array-qempty}(\text{queue}))) \\
\rightarrow & (\text{mapup-queue}(\text{array-deq}(\text{queue})) = \text{deq}(\text{mapup-queue}(\text{queue})))
\end{aligned}$$

EVENT: Disable correctness-of-array-deq.

THEOREM: correctness-of-array-qfirst

$$\begin{aligned}
& (\text{array-queuep}(\text{queue}) \wedge (\neg \text{array-qempty}(\text{queue}))) \\
\rightarrow & (\text{array-qfirst}(\text{queue}) = \text{qfirst}(\text{mapup-queue}(\text{queue})))
\end{aligned}$$

EVENT: Disable correctness-of-array-qfirst.

THEOREM: correctness-of-array-qempty

$$\begin{aligned}
& \text{array-queuep}(\text{queue}) \\
\rightarrow & (\text{array-qempty}(\text{queue}) = \text{qempty}(\text{mapup-queue}(\text{queue})))
\end{aligned}$$

EVENT: Disable correctness-of-array-qempty.

THEOREM: correctness-of-array-qfullp

$$\begin{aligned}
& (\text{array-queuep}(\text{queue}) \wedge (\text{max} = \text{getnth}('3, \text{queue}))) \\
\rightarrow & (\text{array-qfullp}(\text{queue}) = \text{qfullp}(\text{mapup-queue}(\text{queue}), \text{max}))
\end{aligned}$$

THEOREM: length-delta-segment-decr-mod-arg1

$$\begin{aligned}
& ((a \in \mathbf{N}) \wedge ((b = \text{length}(s)) \wedge (a < \text{length}(s)))) \\
\rightarrow & (\text{length}(\text{delta-segment}(\text{decr-mod}(a, b), a, s)) = '1)
\end{aligned}$$

EVENT: Disable length-delta-segment-decr-mod-arg1.

THEOREM: nonlast-of-length-1
 $(\text{length}(l) = '1) \rightarrow (\text{nonlast}(l) = \text{'nil})$

EVENT: Disable nonlast-of-length-1.

THEOREM: correctness-of-array-nonlast-case1
 $(\text{array-queuep}(queue)$
 $\wedge ((\neg \text{array-qempty}(queue)) \wedge \text{array-qempty}(\text{array-nonlast}(queue))))$
 $\rightarrow (\text{nonlast}(\text{delta-segment}(\text{getnth}('0, queue),$
 $\text{getnth}('1, queue),$
 $\text{getseg}('4, \text{getnth}('3, queue), queue)))$
 $= \text{'nil})$

EVENT: Disable correctness-of-array-nonlast-case1.

THEOREM: delta-segment-decr-mod-arg2
 $((a \in \mathbf{N})$
 $\wedge ((a < \text{length}(s))$
 $\wedge ((b \in \mathbf{N})$
 $\wedge ((b < \text{length}(s)) \wedge (a \neq \text{decr-mod}(b, \text{length}(s))))))$
 $\rightarrow (\text{delta-segment}(a, \text{decr-mod}(b, \text{length}(s)), s)$
 $= \text{nonlast}(\text{delta-segment}(a, b, s)))$

EVENT: Disable delta-segment-decr-mod-arg2.

THEOREM: correctness-of-array-nonlast-case2
 $(\text{array-queuep}(queue)$
 $\wedge ((\neg \text{array-qempty}(queue))$
 $\wedge (\neg \text{array-qempty}(\text{array-nonlast}(queue))))$
 $\rightarrow (\text{delta-segment}(\text{getnth}('0, queue),$
 $\text{decr-mod}(\text{getnth}('1, queue), \text{getnth}('3, queue)),$
 $\text{getseg}('4, \text{getnth}('3, queue), queue))$
 $= \text{nonlast}(\text{delta-segment}(\text{getnth}('0, queue),$
 $\text{getnth}('1, queue),$
 $\text{getseg}('4, \text{getnth}('3, queue), queue))))$

EVENT: Disable correctness-of-array-nonlast-case2.

THEOREM: correctness-of-array-nonlast
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{mapup-queue}(\text{array-nonlast}(queue)) = \text{nonlast}(\text{mapup-queue}(queue)))$

EVENT: Disable correctness-of-array-nonlast.

THEOREM: correctness-of-array-qreplace
 $(\text{array-queuep}(queue) \wedge (\neg \text{array-qempty}(queue)))$
 $\rightarrow (\text{mapup-queue}(\text{array-qreplace}(item, queue))$
 $= \text{qreplace}(item, \text{mapup-queue}(queue)))$

EVENT: Disable correctness-of-array-qreplace.

THEOREM: length-mapup-queue-is-qcurrlength
 $\text{array-queuep}(queue) \rightarrow (\text{length}(\text{mapup-queue}(queue)) = \text{getnth}(2, queue))$

EVENT: Disable length-mapup-queue-is-qcurrlength.

DEFINITION: $\text{qfirst2}(i, j, qtable) = \text{qfirst}(\text{getnth2}(i, j, qtable))$

DEFINITION:
 $\text{enq2}(msg, i, j, qtable) = \text{putnth2}(\text{enq}(msg, \text{getnth2}(i, j, qtable)), i, j, qtable)$

DEFINITION:
 $\text{deq2}(i, j, qtable) = \text{putnth2}(\text{deq}(\text{getnth2}(i, j, qtable)), i, j, qtable)$

DEFINITION:
 $\text{qfullp2}(i, j, qtable, max) = \text{qfullp}(\text{getnth2}(i, j, qtable), max)$

DEFINITION: $\text{qempty2}(i, j, qtable) = \text{qempty}(\text{getnth2}(i, j, qtable))$

THEOREM: array-qempty-implies-not-array-qfullp
 $(\text{array-queuep}(queue) \wedge \text{array-qempty}(queue))$
 $\rightarrow (\neg \text{array-qfullp}(queue))$

THEOREM: array-qfirst-array-enq
 $(\text{array-queuep}(queue)$
 $\wedge ((\neg \text{array-qempty}(queue)) \wedge (\neg \text{array-qfullp}(queue))))$
 $\rightarrow (\text{array-qfirst}(\text{array-enq}(item, queue)) = \text{array-qfirst}(queue))$

EVENT: Disable array-qfirst-array-enq.

THEOREM: array-qfirst-array-enq1
 $(\text{array-queuep}(queue) \wedge \text{array-qempty}(queue))$
 $\rightarrow (\text{array-qfirst}(\text{array-enq}(item, queue)) = item)$

EVENT: Disable array-qfirst-array-enq1.

DEFINITION: TM-WORDSIZE = '16

DEFINITION: TM-WORDLUB = '65536

DEFINITION:

alu-result (*value*, *carry*) = cons (*value*, cons (*carry*, 'nil))

DEFINITION: alu-value (*alu-result*) = car (*alu-result*)

DEFINITION: alu-carry (*alu-result*) = car (cdr (*alu-result*))

DEFINITION:

tm-alu-decr (*a*)

= **if** *a* \simeq 0 **then** '65535 *1*true
 else alu-result ((*a* - 1) mod '65536, '1*false) **endif**

DEFINITION:

tm-alu-decr-mod (*a*, *b*)

= alu-result (decr-mod (*a* mod '65536, *b* mod '65536), '1*false)

DEFINITION:

tm-alu-difference (*a*, *b*)

= **if** *b* < *a*
 then alu-result (('65536 - (*a* - *b*)) mod '65536, '1*true)
 else alu-result ((*b* - *a*) mod '65536, '1*false) **endif**

DEFINITION:

tm-alu-incr (*a*)

= **if** (1 + *a*) < '65536 **then** alu-result (1 + *a*, '1*false)
 else '0 *1*true **endif**

DEFINITION:

tm-alu-incr-mod (*a*, *b*)

= alu-result (incr-mod (*a* mod '65536, *b* mod '65536), '1*false)

DEFINITION:

tm-alu-mod (*a*, *b*)

= **if** (*b* mod '65536) \simeq 0 **then** alu-result (*a* mod '65536, '1*false)
 else alu-result (*a* mod (*b* mod '65536), '1*false) **endif**

DEFINITION:

tm-alu-mult (*a*, *b*)

= **if** (*a* * *b*) < '65536 **then** alu-result (*a* * *b*, '1*false)
 else alu-result ((*a* * *b*) mod '65536, '1*true) **endif**

DEFINITION:

tm-alu-plus(a, b)

= **if** ($a + b$) < '65536 **then** alu-result($a + b$, '*1*false)
 else alu-result($(a + b) \bmod '65536$, '*1*true) **endif**

THEOREM: finite-numberp-alu-value-tm-alu-decr
finite-numberp(alu-value(tm-alu-decr(a)), '65536)

EVENT: Disable finite-numberp-alu-value-tm-alu-decr.

THEOREM: lessp-alu-value-tm-alu-decr-wordsiz
(alu-value(tm-alu-decr(a)) < '65536) = '*1*true

EVENT: Disable lessp-alu-value-tm-alu-decr-wordsiz.

THEOREM: lessp-alu-value-tm-alu-decr-wordsiz-linear
alu-value(tm-alu-decr(a)) < '65536

EVENT: Disable lessp-alu-value-tm-alu-decr-wordsiz-linear.

THEOREM: numberp-alu-value-tm-alu-decr
alu-value(tm-alu-decr(a)) $\in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-decr.

THEOREM: tm-alu-decr-value-identical-to-ideal-result
(($a \neq 0$) \wedge ($a < '65536$)) \rightarrow (alu-value(tm-alu-decr(a)) = ($a - 1$))

EVENT: Disable tm-alu-decr-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-decr-mod
finite-numberp(alu-value(tm-alu-decr-mod(a, b)), '65536)

EVENT: Disable finite-numberp-alu-value-tm-alu-decr-mod.

THEOREM: lessp-alu-value-tm-alu-decr-mod-wordsiz
(alu-value(tm-alu-decr-mod(a, b)) < '65536) = '*1*true

EVENT: Disable lessp-alu-value-tm-alu-decr-mod-wordsiz.

THEOREM: lessp-alu-value-tm-alu-decr-mod-wordsiz-linear
alu-value(tm-alu-decr-mod(a, b)) < '65536

EVENT: Disable lessp-alu-value-tm-alu-decr-mod-wordsize-linear.

THEOREM: numberp-alu-value-tm-alu-decr-mod
 $\text{alu-value}(\text{tm-alu-decr-mod}(a, b)) \in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-decr-mod.

THEOREM: tm-alu-decr-mod-value-identical-to-ideal-result
 $((a < '65536) \wedge (b < '65536))$
 $\rightarrow (\text{alu-value}(\text{tm-alu-decr-mod}(a, b)) = \text{decr-mod}(a, b))$

EVENT: Disable tm-alu-decr-mod-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-incr
 $\text{finite-numberp}(\text{alu-value}(\text{tm-alu-incr}(a)), '65536)$

EVENT: Disable finite-numberp-alu-value-tm-alu-incr.

THEOREM: lessp-alu-value-tm-alu-incr-wordsize
 $(\text{alu-value}(\text{tm-alu-incr}(a)) < '65536) = '*1*\text{true}$

EVENT: Disable lessp-alu-value-tm-alu-incr-wordsize.

THEOREM: lessp-alu-value-tm-alu-incr-wordsize-linear
 $\text{alu-value}(\text{tm-alu-incr}(a)) < '65536$

EVENT: Disable lessp-alu-value-tm-alu-incr-wordsize-linear.

THEOREM: numberp-alu-value-tm-alu-incr
 $\text{alu-value}(\text{tm-alu-incr}(a)) \in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-incr.

THEOREM: tm-alu-incr-value-identical-to-ideal-result
 $((1 + a) < '65536) \rightarrow (\text{alu-value}(\text{tm-alu-incr}(a)) = (1 + a))$

EVENT: Disable tm-alu-incr-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-incr-mod
 $\text{finite-numberp}(\text{alu-value}(\text{tm-alu-incr-mod}(a, b)), '65536)$

EVENT: Disable finite-numberp-alu-value-tm-alu-incr-mod.

THEOREM: lessp-alu-value-tm-alu-incr-mod-wordsize
 $(\text{alu-value}(\text{tm-alu-incr-mod}(a, b)) < '65536) = '*1*true$

EVENT: Disable lessp-alu-value-tm-alu-incr-mod-wordsize.

THEOREM: lessp-alu-value-tm-alu-incr-mod-wordsize-linear
 $\text{alu-value}(\text{tm-alu-incr-mod}(a, b)) < '65536$

EVENT: Disable lessp-alu-value-tm-alu-incr-mod-wordsize-linear.

THEOREM: numberp-alu-value-tm-alu-incr-mod
 $\text{alu-value}(\text{tm-alu-incr-mod}(a, b)) \in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-incr-mod.

THEOREM: tm-alu-incr-mod-value-identical-to-ideal-result
 $((a < '65536) \wedge (b < '65536))$
 $\rightarrow (\text{alu-value}(\text{tm-alu-incr-mod}(a, b)) = \text{incr-mod}(a, b))$

EVENT: Disable tm-alu-incr-mod-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-mod
 $\text{finite-numberp}(\text{alu-value}(\text{tm-alu-mod}(a, b)), '65536)$

EVENT: Disable finite-numberp-alu-value-tm-alu-mod.

THEOREM: lessp-alu-value-tm-alu-mod-wordsize
 $(\text{alu-value}(\text{tm-alu-mod}(a, b)) < '65536) = '*1*true$

EVENT: Disable lessp-alu-value-tm-alu-mod-wordsize.

THEOREM: lessp-alu-value-tm-alu-mod-wordsize-linear
 $\text{alu-value}(\text{tm-alu-mod}(a, b)) < '65536$

EVENT: Disable lessp-alu-value-tm-alu-mod-wordsize-linear.

THEOREM: numberp-alu-value-tm-alu-mod
 $\text{alu-value}(\text{tm-alu-mod}(a, b)) \in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-mod.

THEOREM: tm-alu-mod-value-identical-to-ideal-result
(($b \neq 0$) \wedge ($b < '65536$))
 \rightarrow (alu-value (tm-alu-mod (a , b)) = ($a \bmod b$))

EVENT: Disable tm-alu-mod-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-mult
finite-numberp (alu-value (tm-alu-mult (a , b)), '65536)

EVENT: Disable finite-numberp-alu-value-tm-alu-mult.

THEOREM: lessp-alu-value-tm-alu-mult-wordsiz
(alu-value (tm-alu-mult (a , b)) < '65536) = '*1*true

EVENT: Disable lessp-alu-value-tm-alu-mult-wordsiz.

THEOREM: lessp-alu-value-tm-alu-mult-wordsiz-linear
alu-value (tm-alu-mult (a , b)) < '65536

EVENT: Disable lessp-alu-value-tm-alu-mult-wordsiz-linear.

THEOREM: numberp-alu-value-tm-alu-mult
alu-value (tm-alu-mult (a , b)) $\in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-mult.

THEOREM: tm-alu-mult-value-identical-to-ideal-result
(($a * b$) < '65536) \rightarrow (alu-value (tm-alu-mult (a , b)) = ($a * b$))

EVENT: Disable tm-alu-mult-value-identical-to-ideal-result.

THEOREM: finite-numberp-alu-value-tm-alu-plus
finite-numberp (alu-value (tm-alu-plus (a , b)), '65536)

EVENT: Disable finite-numberp-alu-value-tm-alu-plus.

THEOREM: lessp-alu-value-tm-alu-plus-wordsiz
(alu-value (tm-alu-plus (a , b)) < '65536) = '*1*true

EVENT: Disable lessp-alu-value-tm-alu-plus-wordsiz.

THEOREM: lessp-alu-value-tm-alu-plus-wordsiz-linear
alu-value (tm-alu-plus (a , b)) < '65536

EVENT: Disable lessp-alu-value-tm-alu-plus-wordsizes-linear.

THEOREM: numberp-alu-value-tm-alu-plus
alu-value (tm-alu-plus (a , b)) $\in \mathbf{N}$

EVENT: Disable numberp-alu-value-tm-alu-plus.

THEOREM: tm-alu-plus-value-identical-to-ideal-result
(($a + b$) < '65536) \rightarrow (alu-value (tm-alu-plus (a , b)) = ($a + b$))

EVENT: Disable tm-alu-plus-value-identical-to-ideal-result.

EVENT: Let us define the theory *tm-alu-arithmetic* to consist of the following events: tm-alu-plus-value-identical-to-ideal-result, numberp-alu-value-tm-alu-plus, lessp-alu-value-tm-alu-plus-wordsizes-linear, lessp-alu-value-tm-alu-plus-wordsizes, finite-numberp-alu-value-tm-alu-plus, tm-alu-mult-value-identical-to-ideal-result, numberp-alu-value-tm-alu-mult, lessp-alu-value-tm-alu-mult-wordsizes-linear, lessp-alu-value-tm-alu-mult-wordsizes, finite-numberp-alu-value-tm-alu-mult, tm-alu-mod-value-identical-to-ideal-result, numberp-alu-value-tm-alu-mod, lessp-alu-value-tm-alu-mod-wordsizes-linear, lessp-alu-value-tm-alu-mod-wordsizes, finite-numberp-alu-value-tm-alu-mod, tm-alu-incr-mod-value-identical-to-ideal-result, numberp-alu-value-tm-alu-incr-mod, lessp-alu-value-tm-alu-incr-mod-wordsizes-linear, lessp-alu-value-tm-alu-incr-mod-wordsizes, finite-numberp-alu-value-tm-alu-incr-mod, tm-alu-incr-value-identical-to-ideal-result, numberp-alu-value-tm-alu-incr, lessp-alu-value-tm-alu-incr-wordsizes-linear, lessp-alu-value-tm-alu-incr-wordsizes, finite-numberp-alu-value-tm-alu-incr, tm-alu-decr-mod-value-identical-to-ideal-result, numberp-alu-value-tm-alu-decr-mod, lessp-alu-value-tm-alu-decr-mod-wordsizes-linear, lessp-alu-value-tm-alu-decr-mod-wordsizes, finite-numberp-alu-value-tm-alu-decr-mod, tm-alu-decr-value-identical-to-ideal-result, numberp-alu-value-tm-alu-decr, lessp-alu-value-tm-alu-decr-wordsizes-linear, lessp-alu-value-tm-alu-decr-wordsizes, finite-numberp-alu-value-tm-alu-decr.

DEFINITION: tm-incr (a) = ((1 + a) **mod** '65536)

DEFINITION:
tm-incrn (n , a)
= **if** $n \simeq 0$ **then** a
 else tm-incr (tm-incrn ($n - 1$, a)) **endif**

DEFINITION:
tm-decr (a)
= **if** $a \simeq 0$ **then** '65535
 else $a - 1$ **endif**

THEOREM: finite-numberp-tm-incr
finite-numberp (tm-incr (*a*), '65536)

EVENT: Disable finite-numberp-tm-incr.

THEOREM: finite-numberp-tm-incrn
finite-numberp (*a*, '65536) → finite-numberp (tm-incrn (*n*, *a*), '65536)

EVENT: Disable finite-numberp-tm-incrn.

THEOREM: finite-numberp-tm-decr
finite-numberp (*a*, '65536) → finite-numberp (tm-decr (*a*), '65536)

EVENT: Disable finite-numberp-tm-decr.

EVENT: Add the shell *tm*, with recognizer function symbol *tm-shellp* and 14 accessors: *tm-memory*, with type restriction (none-of) and default value zero; *tm-regs*, with type restriction (none-of) and default value zero; *tm-cc*, with type restriction (none-of) and default value zero; *tm-error*, with type restriction (none-of) and default value zero; *tm-svcflag*, with type restriction (none-of) and default value zero; *tm-svcid*, with type restriction (none-of) and default value zero; *tm-base*, with type restriction (none-of) and default value zero; *tm-limit*, with type restriction (none-of) and default value zero; *tm-slimit*, with type restriction (none-of) and default value zero; *tm-svmode*, with type restriction (none-of) and default value zero; *tm-rvstate*, with type restriction (none-of) and default value zero; *tm-clock*, with type restriction (none-of) and default value zero; *tm-iports*, with type restriction (none-of) and default value zero; *tm-oports*, with type restriction (none-of) and default value zero.

DEFINITION: TM-REGSIZE = '3

DEFINITION: TM-CHARSIZE = '8

DEFINITION: TM-CCSIZE = '2

DEFINITION: TM-ERRORSIZE = '6

DEFINITION: TM-SVCFLAGSIZE = '1

DEFINITION: TM-SVCIDSIZE = '7

DEFINITION: TM-CCLUB = '4

DEFINITION: TM-ERRORLUB = '64

DEFINITION: TM-SVCFLAGLUB = '2

DEFINITION: TM-SVCIDLUB = '128

DEFINITION: TM-REGLLENGTH = '8

DEFINITION: TM-MEMLLENGTH = '65536

DEFINITION: TM-CHARLUB = '256

DEFINITION: TM-PORT-LENGTH = '16

DEFINITION: TM-PC-ADDRESS = '0

DEFINITION: TM-SP-ADDRESS = '1

DEFINITION:

tm-set-memory (*memory*, *tm*)

= tm (*memory*,
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*),
tm-slimit (*tm*),
tm-svmode (*tm*),
tm-rwstate (*tm*),
tm-clock (*tm*),
tm-iports (*tm*),
tm-oports (*tm*))

DEFINITION:

tm-set-regs (*regs*, *tm*)

= tm (tm-memory (*tm*),
regs,
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*),
tm-slimit (*tm*),
tm-svmode (*tm*),
tm-rwstate (*tm*),

$\text{tm-clock}(tm)$,
 $\text{tm-iports}(tm)$,
 $\text{tm-oports}(tm)$)

DEFINITION:

$\text{tm-set-cc}(cc, tm)$
 $= \text{tm}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $cc,$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$
 $\text{tm-svcid}(tm),$
 $\text{tm-base}(tm),$
 $\text{tm-limit}(tm),$
 $\text{tm-slimit}(tm),$
 $\text{tm-svmode}(tm),$
 $\text{tm-rwstate}(tm),$
 $\text{tm-clock}(tm),$
 $\text{tm-iports}(tm),$
 $\text{tm-oports}(tm))$

DEFINITION:

$\text{tm-set-error}(error, tm)$
 $= \text{tm}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $error,$
 $\text{tm-svcflag}(tm),$
 $\text{tm-svcid}(tm),$
 $\text{tm-base}(tm),$
 $\text{tm-limit}(tm),$
 $\text{tm-slimit}(tm),$
 $\text{tm-svmode}(tm),$
 $\text{tm-rwstate}(tm),$
 $\text{tm-clock}(tm),$
 $\text{tm-iports}(tm),$
 $\text{tm-oports}(tm))$

DEFINITION:

$\text{tm-set-svcflag}(svcflag, tm)$
 $= \text{tm}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $svcflag,$

$\text{tm-svcid}(tm)$,
 $\text{tm-base}(tm)$,
 $\text{tm-limit}(tm)$,
 $\text{tm-slimit}(tm)$,
 $\text{tm-svmode}(tm)$,
 $\text{tm-rwstate}(tm)$,
 $\text{tm-clock}(tm)$,
 $\text{tm-iports}(tm)$,
 $\text{tm-oports}(tm)$

DEFINITION:

$\text{tm-set-svcid}(svcid, tm)$
 $= \text{tm}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$
 $svcid,$
 $\text{tm-base}(tm),$
 $\text{tm-limit}(tm),$
 $\text{tm-slimit}(tm),$
 $\text{tm-svmode}(tm),$
 $\text{tm-rwstate}(tm),$
 $\text{tm-clock}(tm),$
 $\text{tm-iports}(tm),$
 $\text{tm-oports}(tm))$

DEFINITION:

$\text{tm-set-base}(base, tm)$
 $= \text{tm}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$
 $\text{tm-svcid}(tm),$
 $base,$
 $\text{tm-limit}(tm),$
 $\text{tm-slimit}(tm),$
 $\text{tm-svmode}(tm),$
 $\text{tm-rwstate}(tm),$
 $\text{tm-clock}(tm),$
 $\text{tm-iports}(tm),$
 $\text{tm-oports}(tm))$

DEFINITION:

```

tm-set-limit (limit, tm)
= tm (tm-memory (tm),
      tm-regs (tm),
      tm-cc (tm),
      tm-error (tm),
      tm-svcflag (tm),
      tm-svcid (tm),
      tm-base (tm),
      limit,
      tm-slimit (tm),
      tm-svmode (tm),
      tm-rwstate (tm),
      tm-clock (tm),
      tm-iports (tm),
      tm-oports (tm))

```

DEFINITION:

```

tm-set-slimit (slimit, tm)
= tm (tm-memory (tm),
      tm-regs (tm),
      tm-cc (tm),
      tm-error (tm),
      tm-svcflag (tm),
      tm-svcid (tm),
      tm-base (tm),
      tm-limit (tm),
      slimit,
      tm-svmode (tm),
      tm-rwstate (tm),
      tm-clock (tm),
      tm-iports (tm),
      tm-oports (tm))

```

DEFINITION:

```

tm-set-svmode (svmode, tm)
= tm (tm-memory (tm),
      tm-regs (tm),
      tm-cc (tm),
      tm-error (tm),
      tm-svcflag (tm),
      tm-svcid (tm),
      tm-base (tm),
      tm-limit (tm),
      tm-slimit (tm),
      svmode)

```

$svmode,$
 $tm\text{-}rwstate(tm),$
 $tm\text{-}clock(tm),$
 $tm\text{-}iports(tm),$
 $tm\text{-}oports(tm)$

DEFINITION:

$tm\text{-}set\text{-}rwstate(rwstate, tm)$
 $= tm(tm\text{-}memory(tm),$
 $tm\text{-}regs(tm),$
 $tm\text{-}cc(tm),$
 $tm\text{-}error(tm),$
 $tm\text{-}svcflag(tm),$
 $tm\text{-}svcid(tm),$
 $tm\text{-}base(tm),$
 $tm\text{-}limit(tm),$
 $tm\text{-}slimit(tm),$
 $tm\text{-}svmode(tm),$
 $rwstate,$
 $tm\text{-}clock(tm),$
 $tm\text{-}iports(tm),$
 $tm\text{-}oports(tm))$

DEFINITION:

$tm\text{-}set\text{-}clock(clock, tm)$
 $= tm(tm\text{-}memory(tm),$
 $tm\text{-}regs(tm),$
 $tm\text{-}cc(tm),$
 $tm\text{-}error(tm),$
 $tm\text{-}svcflag(tm),$
 $tm\text{-}svcid(tm),$
 $tm\text{-}base(tm),$
 $tm\text{-}limit(tm),$
 $tm\text{-}slimit(tm),$
 $tm\text{-}svmode(tm),$
 $tm\text{-}rwstate(tm),$
 $clock,$
 $tm\text{-}iports(tm),$
 $tm\text{-}oports(tm))$

DEFINITION:

$tm\text{-}set\text{-}iports(iports, tm)$
 $= tm(tm\text{-}memory(tm),$
 $tm\text{-}regs(tm),$
 $tm\text{-}cc(tm),$

$tm_error(tm)$,
 $tm_svcflag(tm)$,
 $tm_svcid(tm)$,
 $tm_base(tm)$,
 $tm_limit(tm)$,
 $tm_slimit(tm)$,
 $tm_svmode(tm)$,
 $tm_rwstate(tm)$,
 $tm_clock(tm)$,
 $iports$,
 $tm_oport(tm)$

DEFINITION:

$tm_set_oport(oports, tm)$
 $= tm(tm_memory(tm),$
 $tm_regs(tm),$
 $tm_cc(tm),$
 $tm_error(tm),$
 $tm_svcflag(tm),$
 $tm_svcid(tm),$
 $tm_base(tm),$
 $tm_limit(tm),$
 $tm_slimit(tm),$
 $tm_svmode(tm),$
 $tm_rwstate(tm),$
 $tm_clock(tm),$
 $tm_iports(tm),$
 $oports)$

DEFINITION: $tm_pc(tm) = getnth('0, tm_regs(tm))$

DEFINITION:

$tm_set_pc(pc, tm) = tm_set_regs(putnth(pc, '0, tm_regs(tm)), tm)$

DEFINITION: $tm_sp(tm) = getnth('1, tm_regs(tm))$

DEFINITION:

$tm_set_sp(sp, tm) = tm_set_regs(putnth(sp, '1, tm_regs(tm)), tm)$

DEFINITION: $TM_NON_ZERO_NO_CARRY_CONDITION = '0$

DEFINITION: $TM_ZERO_NO_CARRY_CONDITION = '1$

DEFINITION: $TM_NON_ZERO_CARRY_CONDITION = '2$

DEFINITION: $TM_ZERO_CARRY_CONDITION = '3$

DEFINITION:

```
tm-cc-value (alu-result)
=  if alu-value (alu-result)  $\simeq$  0
   then if falsep (alu-carry (alu-result)) then '1
     else '3 endif
   elseif falsep (alu-carry (alu-result)) then '0
     else '2 endif
```

DEFINITION: TM-CC-DIVISOR = '4

DEFINITION: TM-ERROR-DIVISOR = '256

DEFINITION: TM-SVCFLAG-DIVISOR = '512

DEFINITION:

```
tm-pack-psw (cc, error, svcflag, svcid)
=  (cc + ((error * '4)
        + ((svcflag * '256) + (svcid * '512))))
```

THEOREM: finite-numberp-tm-pack-psw1

```
(finite-numberp (cc, '4)
   $\wedge$  (finite-numberp (error, '64)
     $\wedge$  (finite-numberp (svcflag, '2)  $\wedge$  finite-numberp (svcid, '128))))
 $\rightarrow$  finite-numberp (tm-pack-psw (cc, error, svcflag, svcid), '65536)
```

EVENT: Disable finite-numberp-tm-pack-psw1.

DEFINITION: tm-unpack-cc (*psw*) = (*psw* **mod** '4)

DEFINITION: tm-unpack-error (*psw*) = ((*psw* **mod** '256) \div '4)

DEFINITION:

tm-unpack-svcflag (*psw*) = ((*psw* **mod** '512) \div '256)

DEFINITION: tm-unpack-svcid (*psw*) = (*psw* \div '512)

THEOREM: finite-numberp-tm-unpack-cc

```
finite-numberp (tm-unpack-cc (psw), '4)
```

EVENT: Disable finite-numberp-tm-unpack-cc.

THEOREM: finite-numberp-tm-unpack-error

```
finite-numberp (tm-unpack-error (psw), '64)
```

EVENT: Disable finite-numberp-tm-unpack-error.

THEOREM: finite-numberp-tm-unpack-svcflag
finite-numberp (tm-unpack-svcflag (*psw*), '2)

EVENT: Disable finite-numberp-tm-unpack-svcflag.

THEOREM: finite-numberp-tm-unpack-svcid
finite-numberp (*psw*, '65536) \rightarrow finite-numberp (tm-unpack-svcid (*psw*), '128)

EVENT: Disable finite-numberp-tm-unpack-svcid.

DEFINITION: TM-REGISTER-SAVE-AREA-ADDR = '0

DEFINITION: TM-CLOCK-NEW-PC-ADDR = '3

DEFINITION: TM-ERROR-NEW-PC-ADDR = '4

DEFINITION: TM-SVC-NEW-PC-ADDR = '5

DEFINITION: TM-INPUT-NEW-PC-ADDR = '6

DEFINITION: TM-OUTPUT-NEW-PC-ADDR = '7

DEFINITION: TM-SVCID-ADDR = '8

DEFINITION: TM-INPUT-DEVID-ADDR = '8

DEFINITION: TM-INPUT-CHAR-ADDR = '9

DEFINITION: TM-OUTPUT-DEVID-ADDR = '9

DEFINITION: TM-NO-ERROR = '0

DEFINITION: TM-HALT-STATUS = '1

DEFINITION: TM-OPCODE-ERROR = '2

DEFINITION: TM-PRIVILEGE-ERROR = '3

DEFINITION: TM-STACK-OVERFLOW-ERROR = '4

DEFINITION: TM-STACK-UNDERFLOW-ERROR = '5

DEFINITION: TM-ADDRESS-ERROR = '6

DEFINITION: TM-PC-ADDRESS-ERROR = '7

DEFINITION: TM-SUPERVISOR-MODE = '1

DEFINITION: TM-USER-MODE = '0

DEFINITION: tm-in-supervisor-mode(*tm*) = (tm-svmode(*tm*) = '1)

DEFINITION: TM-WAIT-STATE = '1

DEFINITION: TM-RUN-STATE = '0

DEFINITION: tm-waiting(*tm*) = (tm-rwstate(*tm*) = '1)

DEFINITION: TM-SVC = '1

DEFINITION: TM-NO-SVC = '0

DEFINITION: TM-CLEARED-SVCFLAG = '0

EVENT: Add the shell *tm-iport*, with recognizer function symbol *tm-iportp* and 3 accessors: *tm-iinterrupt-flag*, with type restriction (none-of) and default value zero; *tm-ierror-flag*, with type restriction (none-of) and default value zero; *tm-ichar*, with type restriction (none-of) and default value zero.

EVENT: Add the shell *tm-oport*, with recognizer function symbol *tm-oportp* and 3 accessors: *tm-ointerrupt-flag*, with type restriction (none-of) and default value zero; *tm-obusy-flag*, with type restriction (none-of) and default value zero; *tm-ochar*, with type restriction (none-of) and default value zero.

DEFINITION:

tm-iport-errorp(*id*, *ports*) = (tm-ierror-flag(getnth(*id*, *ports*)) = '1)

DEFINITION:

tm-clear-input-interrupt(*id*, *ports*)

= putnth(tm-iport('0,
 tm-ierror-flag(getnth(*id*, *ports*)),
 tm-ichar(getnth(*id*, *ports*))),
 id,
 ports)

DEFINITION:

tm-post-input-interrupt(*char*, *id*, *ports*)

= putnth(tm-iport('1, tm-iinterrupt-flag(getnth(*id*, *ports*)), *char*),
 id,
 ports)

DEFINITION:

tm-oport-idlep(*id*, *ports*) = (tm-obusy-flag(getnth(*id*, *ports*)) = '0)

DEFINITION:

```
tm-start-output (char, id, ports)  
= putnth (tm-oport ('0, '1, char mod '256), id, ports)
```

DEFINITION:

```
tm-post-output-interrupt (id, ports)  
= putnth ('(*1*quote tm-oport 1 0 0), id, ports)
```

DEFINITION:

```
tm-clear-output-interrupt (id, ports)  
= putnth (tm-oport ('0,  
                    tm-obusy-flag (getnth (id, ports)),  
                    tm-ochar (getnth (id, ports))),  
          id,  
          ports)
```

DEFINITION:

```
real-addr (source, num) = cons (source, cons (num, 'nil))
```

DEFINITION: real-addr-source (*real-addr*) = car (*real-addr*)

DEFINITION: real-addr-num (*real-addr*) = car (cdr (*real-addr*))

DEFINITION: tm-arg-mode (*x*) = car (*x*)

DEFINITION: tm-arg-datum (*x*) = car (cdr (*x*))

DEFINITION: tm-arg-index (*x*) = car (cdr (cdr (*x*)))

DEFINITION: tm-fix-word (*n*) = (*n mod* '65536)

DEFINITION: tm-fix-reg-address (*n*) = (*n mod* '8)

DEFINITION:

```
tm-compute-address (arg, tm)  
= if tm-arg-mode (arg)  $\simeq$  0  
  then real-addr ('0,  
                 tm-fix-word (tm-arg-datum (arg) + tm-arg-index (arg)))  
  elseif tm-arg-mode (arg) = '1  
  then real-addr ('1, tm-fix-reg-address (tm-arg-datum (arg)))  
  elseif tm-arg-mode (arg) = '2  
  then real-addr ('2,  
                 tm-fix-word (tm-arg-datum (arg) + tm-arg-index (arg)))  
  else real-addr ('2,  
                 tm-fix-word (getnth (tm-fix-reg-address (tm-arg-datum (arg)),  
                                     tm-regs (tm))  
                               + tm-arg-index (arg))) endif
```

DEFINITION:

```
tm-good-address (addr, tm)
= if real-addr-source (addr)  $\simeq$  0
  then finite-numberp (real-addr-num (addr), '65536)
  elseif real-addr-source (addr) = '1
  then finite-numberp (real-addr-num (addr), '8)
  elseif tm-in-supervisor-mode (tm)
  then finite-numberp (real-addr-num (addr), '65536)
  else finite-numberp (real-addr-num (addr),
    min ('65536 - tm-base (tm), tm-limit (tm))) endif
```

DEFINITION:

```
tm-incr-address (addr)
= if real-addr-source (addr)  $\simeq$  0
  then real-addr (real-addr-source (addr), tm-incr (real-addr-num (addr)))
  elseif real-addr-source (addr) = '1
  then real-addr (real-addr-source (addr),
    (1 + real-addr-num (addr)) mod '8)
  else real-addr (real-addr-source (addr),
    tm-incr (real-addr-num (addr))) endif
```

DEFINITION:

```
tm-incrn-address (n, a)
= if n  $\simeq$  0 then a
  else tm-incr-address (tm-incrn-address (n - 1, a)) endif
```

DEFINITION:

```
tm-fetch-from-memory (addr, tm)
= if tm-in-supervisor-mode (tm) then getnth (addr, tm-memory (tm))
  else getnth (tm-base (tm) + addr, tm-memory (tm)) endif
```

DEFINITION:

```
tm-store-in-memory (value, addr, tm)
= if tm-in-supervisor-mode (tm)
  then tm-set-memory (putnth (value, addr, tm-memory (tm)), tm)
  else tm-set-memory (putnth (value,
    tm-base (tm) + addr,
    tm-memory (tm)),
  tm) endif
```

DEFINITION:

```
tm-fetch-from-regmem (addr, tm) = getnth (addr, tm-regs (tm))
```

DEFINITION:

```
tm-store-in-regmem (value, addr, tm)
= tm-set-regs (putnth (value, addr, tm-regs (tm)), tm)
```

DEFINITION:

tm-fetch (*addr*, *tm*)

= **if** real-addr-source (*addr*) \simeq 0 **then** real-addr-num (*addr*)
elseif real-addr-source (*addr*) = '1
then tm-fetch-from-regmem (real-addr-num (*addr*), *tm*)
else tm-fetch-from-memory (real-addr-num (*addr*), *tm*) **endif**

DEFINITION:

tm-store (*value*, *addr*, *tm*)

= **if** real-addr-source (*addr*) \simeq 0 **then** *tm*
elseif real-addr-source (*addr*) = '1
then tm-store-in-regmem (*value*, real-addr-num (*addr*), *tm*)
else tm-store-in-memory (*value*, real-addr-num (*addr*), *tm*) **endif**

DEFINITION: tm-stack-overflowp (*tm*) = (tm-sp (*tm*) \simeq 0)

DEFINITION:

tm-stack-underflowp (*tm*)

= **if** tm-in-supervisor-mode (*tm*) **then** (1 + tm-sp (*tm*)) $\not\leq$ tm-slimit (*tm*)
else (1 + tm-sp (*tm*))
 $\not\leq$ min ('65536 - tm-base (*tm*), tm-limit (*tm*)) **endif**

THEOREM: tm-stack-underflowp-guards-good-address

((\neg tm-stack-underflowp (*tm*)) \wedge finite-numberp (tm-slimit (*tm*), '65536))
 \rightarrow tm-good-address (real-addr ('2, 1 + tm-sp (*tm*)), *tm*)

EVENT: Disable tm-stack-underflowp-guards-good-address.

DEFINITION:

tm-execute-add (*addr1*, *addr2*, *tm*)

= tm-store (alu-value (tm-alu-plus (tm-fetch (*addr1*, *tm*), tm-fetch (*addr2*, *tm*))),
addr1,
tm-set-cc (tm-cc-value (tm-alu-plus (tm-fetch (*addr1*, *tm*),
tm-fetch (*addr2*, *tm*))),
tm))

DEFINITION:

tm-execute-branch (*addr*, *tm*) = tm-set-pc (tm-fetch (*addr*, *tm*), *tm*)

DEFINITION:

tm-execute-branch-on-zero (*addr*, *tm*)

= **if** tm-cc (*tm*) = '1 **then** tm-set-pc (tm-fetch (*addr*, *tm*), *tm*)
else *tm* **endif**

DEFINITION:

```
tm-execute-branch-not-zero (addr, tm)
=  if tm-cc (tm) = '1 then tm
    else tm-set-pc (tm-fetch (addr, tm), tm) endif
```

DEFINITION:

```
tm-execute-call (addr, tm)
=  if tm-stack-overflowp (tm) then tm-set-error ('4, tm)
    elseif tm-good-address (real-addr ('2, tm-sp (tm)), tm)
    then tm-store (tm-pc (tm),
                  real-addr ('2, tm-sp (tm)),
                  tm-set-pc (tm-fetch (addr, tm),
                              tm-set-sp (tm-decr (tm-sp (tm)), tm)))
    else tm-set-error ('6, tm) endif
```

DEFINITION:

```
tm-execute-compare (addr1, addr2, tm)
=  tm-set-cc (tm-cc-value (tm-alu-difference (tm-fetch (addr1, tm),
                                             tm-fetch (addr2, tm))),
             tm)
```

DEFINITION:

```
tm-execute-decr (addr, tm)
=  tm-store (alu-value (tm-alu-decr (tm-fetch (addr, tm))),
            addr,
            tm-set-cc (tm-cc-value (tm-alu-decr (tm-fetch (addr, tm))), tm))
```

DEFINITION:

```
tm-execute-decr-mod (addr1, addr2, tm)
=  tm-store (alu-value (tm-alu-decr-mod (tm-fetch (addr1, tm),
                                           tm-fetch (addr2, tm))),
            addr1,
            tm-set-cc (tm-cc-value (tm-alu-decr-mod (tm-fetch (addr1, tm),
                                                       tm-fetch (addr2, tm))),
                    tm))
```

DEFINITION:

```
tm-execute-incr (addr, tm)
=  tm-store (alu-value (tm-alu-incr (tm-fetch (addr, tm))),
            addr,
            tm-set-cc (tm-cc-value (tm-alu-incr (tm-fetch (addr, tm))), tm))
```

DEFINITION:

```
tm-execute-incr-mod (addr1, addr2, tm)
=  tm-store (alu-value (tm-alu-incr-mod (tm-fetch (addr1, tm),
```



```

                                tm-fetch (addr2, tm)),
addr1,
tm-set-cc (tm-cc-value (tm-alu-incr-mod (tm-fetch (addr1, tm),
                                tm-fetch (addr2, tm))),
tm))

```

DEFINITION:

```

tm-execute-load-base (addr, tm)
= if tm-in-supervisor-mode (tm) then tm-set-base (tm-fetch (addr, tm), tm)
  else tm-set-error ('3, tm) endif

```

DEFINITION:

```

tm-execute-load-limit (addr, tm)
= if tm-in-supervisor-mode (tm) then tm-set-limit (tm-fetch (addr, tm), tm)
  else tm-set-error ('3, tm) endif

```

DEFINITION:

```

tm-execute-load-psw (addr, tm)
= if tm-in-supervisor-mode (tm)
  then tm-set-svmode ('0,
tm-set-pc (tm-fetch (addr, tm),
tm-set-sp (tm-fetch (tm-incrn-address ('1,
                                addr),
                                tm),
tm-set-cc (tm-unpack-cc (tm-fetch (tm-incrn-address ('2,
                                addr),
                                tm)),
tm-set-error (tm-unpack-error (tm-fetch (tm-incrn-address ('3,
                                addr),
                                tm)),
tm-set-svcflag (tm-unpack-svcflag (tm-fetch (tm-incrn-address ('4,
                                addr),
                                tm)),
tm-set-svcid (tm-unpack-svcid (tm-fetch (tm-incrn-address ('5,
                                addr),
                                tm)))))))))
  else tm-set-error ('3, tm) endif

```

DEFINITION:

```

tm-execute-mod (addr1, addr2, tm)
= tm-store (alu-value (tm-alu-mod (tm-fetch (addr1, tm), tm-fetch (addr2, tm))),
addr1,
tm-set-cc (tm-cc-value (tm-alu-mod (tm-fetch (addr1, tm),
tm-fetch (addr2, tm))),
tm))

```

DEFINITION:

tm-execute-move (*addr1*, *addr2*, *tm*) = tm-store (tm-fetch (*addr2*, *tm*), *addr1*, *tm*)

DEFINITION:

tm-execute-multiply (*addr1*, *addr2*, *tm*)
= tm-store (alu-value (tm-alu-mult (tm-fetch (*addr1*, *tm*), tm-fetch (*addr2*, *tm*))),
 addr1,
 tm-set-cc (tm-cc-value (tm-alu-mult (tm-fetch (*addr1*, *tm*),
 tm-fetch (*addr2*, *tm*))),
 tm))

DEFINITION:

tm-execute-post-output-interrupt (*addr*, *tm*)
= **if** tm-in-supervisor-mode (*tm*)
 then tm-set-oports (tm-post-output-interrupt (tm-fetch (*addr*, *tm*),
 tm-oports (*tm*)),
 tm)
 else tm-set-error ('3, *tm*) **endif**

DEFINITION:

tm-execute-return (*tm*)
= **if** tm-stack-underflowp (*tm*) **then** tm-set-error ('5, *tm*)
 else tm-set-pc (tm-fetch (real-addr ('2, tm-incr (tm-sp (*tm*))), *tm*),
 tm-set-sp (tm-incr (tm-sp (*tm*)), *tm*)) **endif**

DEFINITION:

tm-execute-run (*tm*)
= **if** tm-in-supervisor-mode (*tm*) **then** tm-set-rwstate ('0, *tm*)
 else tm-set-error ('3, *tm*) **endif**

DEFINITION:

tm-execute-set-clock (*addr*, *tm*)
= **if** tm-in-supervisor-mode (*tm*) **then** tm-set-clock (tm-fetch (*addr*, *tm*), *tm*)
 else tm-set-error ('3, *tm*) **endif**

DEFINITION:

tm-execute-start-output (*addr1*, *addr2*, *tm*)
= **if** tm-in-supervisor-mode (*tm*)
 then tm-set-oports (tm-start-output (tm-fetch (*addr2*, *tm*),
 tm-fetch (*addr1*, *tm*),
 tm-oports (*tm*)),
 tm)
 else tm-set-error ('3, *tm*) **endif**

DEFINITION:

```

tm-execute-svc(addr, tm)
= if tm-in-supervisor-mode(tm) then tm
  else tm-set-svcflag('1,
                    tm-set-svcid(tm-fetch(addr, tm) mod '128,
                    tm)) endif

```

DEFINITION:

```

tm-execute-svc-return(addr, tm)
= if tm-in-supervisor-mode(tm)
  then tm-set-svmode('0,
                    tm-set-pc(tm-fetch(addr, tm),
                    tm-set-sp(tm-fetch(tm-incrn-address('1,
                    addr),
                    tm),
                    tm-set-cc(tm-unpack-cc(tm-fetch(tm-incrn-address('2,
                    addr),
                    tm),
                    tm-set-error(tm-unpack-error(tm-fetch(tm-incrn-address('3,
                    tm),
                    tm-set-svcflag('0,
                    tm-set-svcid(tm-unpack-cc(tm-fetch(tm-incrn-address(tm)))))))))
  else tm-set-error('3, tm) endif

```

DEFINITION:

```

tm-execute-test-iport(addr, tm)
= if tm-in-supervisor-mode(tm)
  then if tm-iport-errorp(tm-fetch(addr, tm) mod '16, tm-iports(tm))
    then tm-set-cc('1, tm)
    else tm-set-cc('0, tm) endif
  else tm-set-error('3, tm) endif

```

DEFINITION:

```

tm-execute-test-oport(addr, tm)
= if tm-in-supervisor-mode(tm)
  then if tm-oport-idlep(tm-fetch(addr, tm) mod '16, tm-oports(tm))
    then tm-set-cc('1, tm)
    else tm-set-cc('0, tm) endif
  else tm-set-error('3, tm) endif

```

DEFINITION:

```

tm-execute-wait(tm)

```

```

= if tm-in-supervisor-mode (tm)
  then tm-set-rwstate ('1, tm-set-svmode('0, tm))
  else tm-set-error('3, tm) endif

DEFINITION: TM-OPCODE-SIZE = '6

DEFINITION: TM-MODE-SIZE = '2

DEFINITION: TM-INDEX-SIZE = '3

DEFINITION: TM-OPCODE-DIVISOR = '64

DEFINITION: TM-MODE1-DIVISOR = '256

DEFINITION: TM-INDEX1-DIVISOR = '2048

DEFINITION: TM-MODE2-DIVISOR = '8192

DEFINITION: tm-opcode (instruction) = (instruction mod '64)

DEFINITION:
tm-mode1 (instruction) = ((instruction mod '256) ÷ '64)

DEFINITION:
tm-index1 (instruction) = ((instruction mod '2048) ÷ '256)

DEFINITION:
tm-mode2 (instruction) = ((instruction mod '8192) ÷ '2048)

DEFINITION: tm-index2 (instruction) = (instruction ÷ '8192)

DEFINITION:
tm-nullary-instructionp (instruction)
= ((tm-opcode (instruction) mod '4) = '0)

DEFINITION:
tm-unary-instructionp (instruction)
= ((tm-opcode (instruction) mod '2) = '0)

DEFINITION:
tm-datum1 (instruction-list) = car (cdr (instruction-list))

DEFINITION:
tm-datum2 (instruction-list) = car (cdr (cdr (instruction-list)))

DEFINITION:
tm-arg1 (instruction-list)
= cons (tm-mode1 (car (instruction-list)),
       cons (tm-datum1 (instruction-list),
            cons (tm-index1 (car (instruction-list)), 'nil)))

```

DEFINITION:
tm-arg2 (*instruction-list*)
= cons (tm-mode2 (car (*instruction-list*)),
cons (tm-datum2 (*instruction-list*),
cons (tm-index2 (car (*instruction-list*)), 'nil)))

DEFINITION: TM-RETURN-OPCODE = '0

DEFINITION: TM-WAIT-OPCODE = '4

DEFINITION: TM-RUN-OPCODE = '8

DEFINITION:
tm-execute-nullary (*opcode*, *tm*)
= **if** *opcode* = '0 **then** tm-execute-return (*tm*)
elseif *opcode* = '8 **then** tm-execute-run (*tm*)
elseif *opcode* = '4 **then** tm-execute-wait (*tm*)
else tm-set-error ('2, *tm*) **endif**

DEFINITION: TM-BR-OPCODE = '2

DEFINITION: TM-BRNZ-OPCODE = '6

DEFINITION: TM-BRZ-OPCODE = '10

DEFINITION: TM-CALL-OPCODE = '14

DEFINITION: TM-DECR-OPCODE = '18

DEFINITION: TM-INCR-OPCODE = '22

DEFINITION: TM-LBASE-OPCODE = '26

DEFINITION: TM-LLIMIT-OPCODE = '30

DEFINITION: TM-LPSW-OPCODE = '34

DEFINITION: TM-POST-OPCODE = '38

DEFINITION: TM-SVC-OPCODE = '42

DEFINITION: TM-SVC-RETURN-OPCODE = '46

DEFINITION: TM-TESTI-OPCODE = '50

DEFINITION: TM-TESTO-OPCODE = '54

DEFINITION: TM-TIME-OPCODE = '58

DEFINITION:

```
tm-execute-unary(opcode, addr, tm)
=  if ¬ tm-good-address(addr, tm) then tm-set-error('6, tm)
    elseif opcode = '2 then tm-execute-branch(addr, tm)
    elseif opcode = '10 then tm-execute-branch-on-zero(addr, tm)
    elseif opcode = '6 then tm-execute-branch-not-zero(addr, tm)
    elseif opcode = '14 then tm-execute-call(addr, tm)
    elseif opcode = '18 then tm-execute-decr(addr, tm)
    elseif opcode = '22 then tm-execute-incr(addr, tm)
    elseif opcode = '26 then tm-execute-load-base(addr, tm)
    elseif opcode = '30 then tm-execute-load-limit(addr, tm)
    elseif opcode = '34 then tm-execute-load-psw(addr, tm)
    elseif opcode = '58 then tm-execute-set-clock(addr, tm)
    elseif opcode = '50 then tm-execute-test-iport(addr, tm)
    elseif opcode = '54 then tm-execute-test-oport(addr, tm)
    elseif opcode = '42 then tm-execute-svc(addr, tm)
    elseif opcode = '46 then tm-execute-svc-return(addr, tm)
    elseif opcode = '38 then tm-execute-post-output-interrupt(addr, tm)
    else tm-set-error('2, tm) endif
```

DEFINITION: TM-ADD-OPCODE = '1

DEFINITION: TM-COMPARE-OPCODE = '3

DEFINITION: TM-DECR-MOD-OPCODE = '5

DEFINITION: TM-INCR-MOD-OPCODE = '7

DEFINITION: TM-MOD-OPCODE = '9

DEFINITION: TM-MOVE-OPCODE = '11

DEFINITION: TM-MULT-OPCODE = '13

DEFINITION: TM-STOUT-OPCODE = '15

DEFINITION:

```
tm-execute-binary(opcode, addr1, addr2, tm)
=  if (¬ tm-good-address(addr1, tm))
    ∨ (¬ tm-good-address(addr2, tm)) then tm-set-error('6, tm)
    elseif opcode = '1 then tm-execute-add(addr1, addr2, tm)
    elseif opcode = '3 then tm-execute-compare(addr1, addr2, tm)
    elseif opcode = '5 then tm-execute-decr-mod(addr1, addr2, tm)
    elseif opcode = '7 then tm-execute-incr-mod(addr1, addr2, tm)
    elseif opcode = '9 then tm-execute-mod(addr1, addr2, tm)
    elseif opcode = '11 then tm-execute-move(addr1, addr2, tm)
```

```

elseif opcode = '13 then tm-execute-multiply (addr1, addr2, tm)
elseif opcode = '15 then tm-execute-start-output (addr1, addr2, tm)
else tm-set-error ('2, tm) endif

```

DEFINITION:

```

tm-execute-instruction (instruction, tm)
= if length (instruction) = '1
  then tm-execute-nullary (tm-opcode (car (instruction)), tm)
  elseif length (instruction) = '2
  then tm-execute-unary (tm-opcode (car (instruction)),
                        tm-compute-address (tm-arg1 (instruction), tm),
                        tm)
  elseif length (instruction) = '3
  then tm-execute-binary (tm-opcode (car (instruction)),
                          tm-compute-address (tm-arg1 (instruction), tm),
                          tm-compute-address (tm-arg2 (instruction), tm),
                          tm)
  else tm-set-error ('2, tm) endif

```

DEFINITION:

```

tm-fetch-opcode (tm) = tm-fetch (real-addr ('2, tm-pc (tm)), tm)

```

DEFINITION:

```

tm-fetch-arg1 (tm) = tm-fetch (real-addr ('2, tm-incrn ('1, tm-pc (tm))), tm)

```

DEFINITION:

```

tm-fetch-arg2 (tm) = tm-fetch (real-addr ('2, tm-incrn ('2, tm-pc (tm))), tm)

```

DEFINITION:

```

tm-execute (opcode, tm)
= if tm-nullary-instructionp (opcode)
  then tm-execute-instruction (cons (opcode, 'nil),
                                tm-set-pc (tm-incrn ('1, tm-pc (tm)), tm))
  elseif tm-good-address (real-addr ('2, tm-incrn ('1, tm-pc (tm))), tm)
  then if tm-unary-instructionp (opcode)
    then tm-execute-instruction (cons (opcode,
                                       cons (tm-fetch-arg1 (tm), 'nil)),
                                 tm-set-pc (tm-incrn ('2, tm-pc (tm)), tm))
    elseif tm-good-address (real-addr ('2, tm-incrn ('2, tm-pc (tm))), tm)
    then tm-execute-instruction (cons (opcode,
                                       cons (tm-fetch-arg1 (tm),
                                             cons (tm-fetch-arg2 (tm),
                                                  'nil))),
                                 tm-set-pc (tm-incrn ('3, tm-pc (tm)), tm))
  else tm-set-error ('7, tm) endif
else tm-set-error ('7, tm) endif

```

DEFINITION:

```
tm-decrement-clock (tm)
= if tm-clock (tm)  $\simeq$  0 then tm
  elseif  $\neg$  tm-in-supervisor-mode (tm)
  then tm-set-clock (tm-decr (tm-clock (tm)), tm)
  else tm endif
```

DEFINITION:

```
tm-good-pc-address (tm) = tm-good-address (real-addr ('2, tm-pc (tm)), tm)
```

DEFINITION:

```
tm-fetch-execute (tm)
= if tm-good-pc-address (tm)
  then tm-execute (tm-fetch-opcode (tm), tm-decrement-clock (tm))
  else tm-set-error ('7, tm-decrement-clock (tm)) endif
```

DEFINITION:

```
tm-fetch-new-pc-on-interrupt (addr, tm)
= tm-set-pc (tm-fetch-from-memory (addr, tm), tm)
```

DEFINITION:

```
tm-store-old-psw-on-interrupt (addr, tm)
= tm-store (tm-pc (tm),
            real-addr ('2, addr),
            tm-store (tm-sp (tm),
                    tm-incrn-address ('1, real-addr ('2, addr)),
                    tm-store (tm-pack-psw (tm-cc (tm),
                                           tm-error (tm),
                                           tm-svflag (tm),
                                           tm-svcid (tm)),
                    tm-incrn-address ('2, real-addr ('2, addr)),
                    tm)))
```

DEFINITION:

```
tm-clock-interruptp (tm)
= (( $\neg$  tm-in-supervisor-mode (tm))  $\wedge$  (tm-clock (tm)  $\simeq$  0))
```

DEFINITION:

```
tm-execute-clock-interrupt (tm)
= tm-set-sp (tm-decr (tm-slimit (tm)),
            tm-fetch-new-pc-on-interrupt ('3,
            tm-store-old-psw-on-interrupt ('0,
            tm-set-svmode ('1,
                           tm))))
```

DEFINITION: tm-errorp (*tm*) = (tm-error (*tm*) \neq '0)

DEFINITION:

```
tm-execute-error-interrupt (tm)
= tm-set-sp (tm-decr (tm-slimit (tm)),
            tm-set-error ('0,
                          tm-fetch-new-pc-on-interrupt ('4,
                                                         tm-store-old-psw-on-interrupt ('0,
                                                                 tm-set-svmode ('1,
                                                                 tm))),
```

DEFINITION:

```
tm-svc-interruptp (tm)
= ((¬ tm-in-supervisor-mode (tm)) ∧ (tm-svcflag (tm) = '1))
```

DEFINITION:

```
tm-store-svcid-on-interrupt (tm) = tm-store (tm-svcid (tm), '2 8), tm)
```

DEFINITION:

```
tm-execute-svc-interrupt (tm)
= tm-set-sp (tm-decr (tm-slimit (tm)),
            tm-fetch-new-pc-on-interrupt ('5,
                                         tm-store-svcid-on-interrupt (tm-store-old-psw-on-interrupt ('0,
                                                                 tm-se
```

DEFINITION: $tm\text{-overflow-char}(char) = ('256 + char)$

DEFINITION:

```
tm-some-input-interruptp (ports)
= if listp (ports)
  then (tm-iinterrupt-flag (car (ports)) = '1)
    ∨ tm-some-input-interruptp (cdr (ports))
  else '*1*false endif
```

DEFINITION:

```
tm-input-interruptp (tm)
= ((¬ tm-in-supervisor-mode (tm))
   ∧ tm-some-input-interruptp (tm-iports (tm)))
```

DEFINITION:

```
tm-interrupting-input-port (ports)
= if listp (ports)
  then if tm-iinterrupt-flag (car (ports)) = '1 then '0
    else 1 + tm-interrupting-input-port (cdr (ports)) endif
  else '0 endif
```

DEFINITION:

```
tm-store-interrupting-input-device (id, tm)
= tm-store (id,
            '(2 8),
            tm-store (tm-ichar (getnth (id, tm-iports (tm))), '(2 9), tm))
```

DEFINITION:

```
tm-execute-input-interrupt (tm)
= tm-set-sp (tm-decr (tm-slimit (tm)),
            tm-set-error ('0,
                          tm-fetch-new-pc-on-interrupt ('6,
                                                         tm-set-iports (tm-clear-input-interrupt (tm-interrupting-input-device (tm-iports (tm))),
                                                         tm-store-interrupting-input-device (tm-iports (tm))),
                                                         tm-iports (tm))),
            tm-iports (tm))
```

DEFINITION:

```
tm-some-output-interruptp (ports)
= if listp (ports)
  then (tm-ointerrupt-flag (car (ports)) = '1)
    ∨ tm-some-output-interruptp (cdr (ports))
  else '*1*false endif
```

DEFINITION:

```
tm-output-interruptp (tm)
= ((¬ tm-in-supervisor-mode (tm))
   ∧ tm-some-output-interruptp (tm-oports (tm)))
```

DEFINITION:

```
tm-interrupting-output-port (ports)
= if listp (ports)
  then if tm-ointerrupt-flag (car (ports)) = '1 then '0
    else 1 + tm-interrupting-output-port (cdr (ports)) endif
  else '0 endif
```

DEFINITION:

```
tm-store-interrupting-output-device (id, tm) = tm-store (id, '(2 9), tm)
```

DEFINITION:

```
tm-execute-output-interrupt (tm)
= tm-set-sp (tm-decr (tm-slimit (tm)),
            tm-set-error ('0,
```

```

tm-fetch-new-pc-on-interrupt ('7,
                                tm-set-oports (tm-clear-output-interrupt (tm-interrupt
                                                tm-oports
                                                tm-store-interrupting-output-device

```

EVENT: Add the shell *tm-device-input-event*, with recognizer function symbol *tm-device-input-eventp* and 2 accessors: *tm-idevid*, with type restriction (none-of) and default value zero; *tm-idatum*, with type restriction (none-of) and default value zero.

EVENT: Add the shell *tm-device-output-event*, with recognizer function symbol *tm-device-output-eventp* and 1 accessor: *tm-odevid*, with type restriction (none-of) and default value zero.

DEFINITION:

```

tm-post-interrupt (event, tm)
=  if tm-device-input-eventp (event)
    then tm-set-iports (tm-post-input-interrupt (tm-idatum (event)
                                                    mod '256,
                                                    tm-idevid (event) mod '16,
                                                    tm-iports (tm)),
                    tm)
    elseif tm-device-output-eventp (event)
    then tm-set-oports (tm-post-output-interrupt (tm-odevid (event)
                                                    mod '16,
                                                    tm-oports (tm)),
                    tm)
    else tm endif

```

DEFINITION:

```

tm-step (tm)
=  if tm-input-interruptp (tm) then tm-execute-input-interrupt (tm)
    elseif tm-output-interruptp (tm) then tm-execute-output-interrupt (tm)
    elseif tm-waiting (tm) then tm
    elseif tm-errorp (tm) then tm-execute-error-interrupt (tm)
    elseif tm-clock-interruptp (tm) then tm-execute-clock-interrupt (tm)
    elseif tm-svc-interruptp (tm) then tm-execute-svc-interrupt (tm)
    else tm-fetch-execute (tm) endif

```

DEFINITION:

```

tm-processor (tm, oracle)
=  if listp (oracle)
    then tm-processor (tm-step (tm-post-interrupt (car (oracle), tm)),
                        cdr (oracle))
    else tm endif

```

THEOREM: tm-processor-with-null-oracle
tm-processor (*tm*, 'nil) = *tm*

EVENT: Disable tm-processor-with-null-oracle.

```

THEOREM: open-up-tm-processor
tm-processor (tm, cons (a, b))
=  tm-processor (tm-step (tm-post-interrupt (a, tm)), b)

```

EVENT: Disable open-up-tm-processor.

```

THEOREM: tm-processor-append-oracle
tm-processor (tm, append (a, b)) = tm-processor (tm-processor (tm, a), b)

```

EVENT: Disable tm-processor-append-oracle.

```

THEOREM: finite-numberp-tm-cc-value
finite-numberp (tm-cc-value (x), '4)

```

EVENT: Disable finite-numberp-tm-cc-value.

```

THEOREM: lessp-tm-cc-value-tm-cclub
(tm-cc-value (x) < '4) = '*1*true

```

EVENT: Disable lessp-tm-cc-value-tm-cclub.

```

THEOREM: lessp-tm-cc-value-tm-cclub-linear
tm-cc-value (x) < '4

```

EVENT: Disable lessp-tm-cc-value-tm-cclub-linear.

```

THEOREM: tm-cc-value-on-compare-equal
(finite-numberp (a, '65536) ∧ (finite-numberp (b, '65536) ∧ (a = b)))
→ (tm-cc-value (tm-alu-difference (a, b)) = '1)

```

EVENT: Disable tm-cc-value-on-compare-equal.

THEOREM: tm-cc-value-on-compare-greaterp
(($b < a$) \wedge ($a < '65536$))
 \rightarrow (tm-cc-value (tm-alu-difference (a , b)) = '2)

EVENT: Disable tm-cc-value-on-compare-greaterp.

THEOREM: tm-cc-value-on-compare-lessp
(($a < b$) \wedge ($b < '65536$))
 \rightarrow (tm-cc-value (tm-alu-difference (a , b)) = '0)

EVENT: Disable tm-cc-value-on-compare-lessp.

THEOREM: tm-cc-value-on-compare-not-equal
(finite-numberp (a , '65536) \wedge (finite-numberp (b , '65536) \wedge ($a \neq b$)))
 \rightarrow ((tm-cc-value (tm-alu-difference (a , b)) = '1) = '*1*false)

EVENT: Disable tm-cc-value-on-compare-not-equal.

THEOREM: tm-cc-value-on-compare-not-equal-to-zero
(finite-numberp (a , '65536) \wedge ($a \neq '0$))
 \rightarrow (tm-cc-value (tm-alu-difference (a , '0)) = '2)

EVENT: Disable tm-cc-value-on-compare-not-equal-to-zero.

THEOREM: numberp-tm-fix-word
tm-fix-word (n) $\in \mathbf{N}$

EVENT: Disable numberp-tm-fix-word.

THEOREM: tm-fix-word-size
(tm-fix-word (n) < '65536) = '*1*true

EVENT: Disable tm-fix-word-size.

THEOREM: tm-fix-word-size-linear
tm-fix-word (n) < '65536

EVENT: Disable tm-fix-word-size-linear.

THEOREM: tm-fix-word-noop
($n < '65536$) \rightarrow (tm-fix-word (n) = fix (n))

EVENT: Disable tm-fix-word-noop.

THEOREM: numberp-tm-fix-reg-address
tm-fix-reg-address (n) $\in \mathbf{N}$

EVENT: Disable numberp-tm-fix-reg-address.

THEOREM: tm-fix-reg-address-size
(tm-fix-reg-address (n) < '8) = '*1*true

EVENT: Disable tm-fix-reg-address-size.

THEOREM: tm-fix-reg-address-size-linear
tm-fix-reg-address (n) < '8

EVENT: Disable tm-fix-reg-address-size-linear.

THEOREM: tm-fix-reg-address-noop
(n < '8) \rightarrow (tm-fix-reg-address (n) = fix (n))

EVENT: Disable tm-fix-reg-address-noop.

DEFINITION:

good-tm-iport (x)
= (tm-iportp (x)
 \wedge (finite-numberp (tm-iinterrupt-flag (x), '2)
 \wedge (finite-numberp (tm-ierror-flag (x), '2)
 \wedge finite-numberp (tm-ichar (x), '256))))

DEFINITION:

good-tm-iport-array (l)
= **if** listp (l)
 then good-tm-iport (car (l)) \wedge good-tm-iport-array (cdr (l))
 else '*1*true **endif**

DEFINITION:

good-tm-oport (x)
= (tm-oportp (x)
 \wedge (finite-numberp (tm-ointerrupt-flag (x), '2)
 \wedge (finite-numberp (tm-obusy-flag (x), '2)
 \wedge finite-numberp (tm-ochar (x), '256))))

```

DEFINITION:
good-tm-oport-array (l)
=  if listp (l)
    then good-tm-oport (car (l)) ^ good-tm-oport-array (cdr (l))
    else '*1*true endif

```

```

(DEFN GOOD-TM
  (TM)
  (AND
    (TM-SHELLP TM)
    (AND
      (PLISTP (TM-MEMORY TM))
      (AND
        (EQUAL (LENGTH (TM-MEMORY TM))
          '65536)
        (AND
          (FINITE-NUMBER-LISTP (TM-MEMORY TM)
            '65536)
          (AND
            (PLISTP (TM-REGS TM))
            (AND
              (EQUAL (LENGTH (TM-REGS TM)) '8)
              (AND
                (FINITE-NUMBER-LISTP (TM-REGS TM)
                  '65536)
                (AND
                  (FINITE-NUMBERP (TM-CC TM) '4)
                  (AND
                    (FINITE-NUMBERP (TM-ERROR TM) '64)
                    (AND
                      (FINITE-NUMBERP (TM-SVCFLAG TM) '2)
                      (AND
                        (FINITE-NUMBERP (TM-SVCID TM) '128)
                        (AND
                          (FINITE-NUMBERP (TM-BASE TM) '65536)
                          (AND
                            (FINITE-NUMBERP (TM-LIMIT TM) '65536)
                            (AND
                              (FINITE-NUMBERP (TM-SLIMIT TM)
                                '65536)
                              (AND
                                (FINITE-NUMBERP (TM-SVMODE TM) '2)

```

```

(AND
  (FINITE-NUMBERP (TM-RWSTATE TM) '2)
  (AND
    (FINITE-NUMBERP (TM-CLOCK TM) '65536)
    (AND
      (PLISTP (TM-IPOINTS TM))
      (AND
        (EQUAL (LENGTH (TM-IPOINTS TM)) '16)
        (AND
          (GOOD-TM-IPOINT-ARRAY (TM-IPOINTS TM))
          (AND
            (PLISTP (TM-OPOINTS TM))
            (AND
              (EQUAL (LENGTH (TM-OPOINTS TM)) '16)
              (GOOD-TM-OPOINT-ARRAY
                (TM-OPOINTS TM))))))))))))))
NIL)

```

THEOREM: tm-shellp-tm
 $\text{good-tm}(tm) \rightarrow \text{tm-shellp}(tm)$

EVENT: Disable tm-shellp-tm.

THEOREM: plistp-tm-memory
 $\text{good-tm}(tm) \rightarrow \text{plistp}(\text{tm-memory}(tm))$

EVENT: Disable plistp-tm-memory.

THEOREM: length-tm-memory
 $\text{good-tm}(tm) \rightarrow (\text{length}(\text{tm-memory}(tm)) = '65536)$

EVENT: Disable length-tm-memory.

THEOREM: finite-number-listp-tm-memory
 $\text{good-tm}(tm) \rightarrow \text{finite-number-listp}(\text{tm-memory}(tm), '65536)$

EVENT: Disable finite-number-listp-tm-memory.

THEOREM: numberp-getnth-from-tm-memory
 $(\text{good-tm}(tm) \wedge (i < '65536)) \rightarrow (\text{getnth}(i, \text{tm-memory}(tm)) \in \mathbf{N})$

EVENT: Disable numberp-getnth-from-tm-memory.

THEOREM: plistp-tm-regs
 $\text{good-tm}(tm) \rightarrow \text{plistp}(\text{tm-regs}(tm))$

EVENT: Disable plistp-tm-regs.

THEOREM: length-tm-regs
 $\text{good-tm}(tm) \rightarrow (\text{length}(\text{tm-regs}(tm)) = '8)$

EVENT: Disable length-tm-regs.

THEOREM: finite-number-listp-tm-regs
 $\text{good-tm}(tm) \rightarrow \text{finite-number-listp}(\text{tm-regs}(tm), '65536)$

EVENT: Disable finite-number-listp-tm-regs.

THEOREM: numberp-getnth-from-tm-regs
 $(\text{good-tm}(tm) \wedge (i < '8)) \rightarrow (\text{getnth}(i, \text{tm-regs}(tm)) \in \mathbf{N})$

EVENT: Disable numberp-getnth-from-tm-regs.

THEOREM: finite-numberp-tm-cc
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-cc}(tm), '4)$

EVENT: Disable finite-numberp-tm-cc.

THEOREM: finite-numberp-tm-error
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-error}(tm), '64)$

EVENT: Disable finite-numberp-tm-error.

THEOREM: finite-numberp-tm-base
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-base}(tm), '65536)$

EVENT: Disable finite-numberp-tm-base.

THEOREM: finite-numberp-tm-limit
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-limit}(tm), '65536)$

EVENT: Disable finite-numberp-tm-limit.

THEOREM: finite-numberp-tm-slimit
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-slimit}(tm), '65536)$

EVENT: Disable finite-numberp-tm-slimit.

THEOREM: finite-numberp-tm-svmode
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-svmode}(tm), '2)$

EVENT: Disable finite-numberp-tm-svmode.

THEOREM: finite-numberp-tm-clock
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-clock}(tm), '65536)$

EVENT: Disable finite-numberp-tm-clock.

THEOREM: finite-numberp-tm-pc
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-pc}(tm), '65536)$

EVENT: Disable finite-numberp-tm-pc.

THEOREM: finite-numberp-tm-sp
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-sp}(tm), '65536)$

EVENT: Disable finite-numberp-tm-sp.

THEOREM: finite-numberp-tm-pack-psw
 $\text{good-tm}(tm)$
 $\rightarrow \text{finite-numberp}(\text{tm-pack-psw}(\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$
 $\text{tm-svcid}(tm)),$
 $'65536)$

EVENT: Disable finite-numberp-tm-pack-psw.

THEOREM: finite-numberp-tm-svcflag
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-svcflag}(tm), '2)$

EVENT: Disable finite-numberp-tm-svcflag.

THEOREM: finite-numberp-tm-svcid
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-svcid}(tm), '128)$

EVENT: Disable finite-numberp-tm-svcid.

THEOREM: finite-numberp-tm-rwstate
 $\text{good-tm}(tm) \rightarrow \text{finite-numberp}(\text{tm-rwstate}(tm), '2)$

EVENT: Disable finite-numberp-tm-rwstate.

THEOREM: plistp-tm-iports
 $\text{good-tm}(tm) \rightarrow \text{plistp}(\text{tm-iports}(tm))$

EVENT: Disable plistp-tm-iports.

THEOREM: length-tm-iports
 $\text{good-tm}(tm) \rightarrow (\text{length}(\text{tm-iports}(tm)) = '16)$

EVENT: Disable length-tm-iports.

THEOREM: good-tm-iport-array-tm-iports
 $\text{good-tm}(tm) \rightarrow \text{good-tm-iport-array}(\text{tm-iports}(tm))$

EVENT: Disable good-tm-iport-array-tm-iports.

THEOREM: plistp-tm-oports
 $\text{good-tm}(tm) \rightarrow \text{plistp}(\text{tm-oports}(tm))$

EVENT: Disable plistp-tm-oports.

THEOREM: length-tm-oports
 $\text{good-tm}(tm) \rightarrow (\text{length}(\text{tm-oports}(tm)) = '16)$

EVENT: Disable length-tm-oports.

THEOREM: good-tm-oport-array-tm-oports
 $\text{good-tm}(tm) \rightarrow \text{good-tm-oport-array}(\text{tm-oports}(tm))$

EVENT: Disable good-tm-oport-array-tm-oports.

THEOREM: good-tm-iport-getnth
 $(\text{good-tm-iport-array}(l) \wedge (i < \text{length}(l)))$
 $\rightarrow \text{good-tm-iport}(\text{getnth}(i, l))$

EVENT: Disable good-tm-iport-getnth.

THEOREM: good-tm-iport-array-putnth
(good-tm-iport-array (l) \wedge good-tm-iport ($iport$))
 \rightarrow good-tm-iport-array (putnth ($iport$, n , l))

EVENT: Disable good-tm-iport-array-putnth.

THEOREM: good-tm-iport-getnth-from-tm-iports
(good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow good-tm-iport (getnth (id , tm-iports (tm)))

EVENT: Disable good-tm-iport-getnth-from-tm-iports.

THEOREM: good-tm-iport-array-putnth-into-tm-iports
(good-tm (tm) \wedge good-tm-iport ($iport$))
 \rightarrow good-tm-iport-array (putnth ($iport$, n , tm-iports (tm)))

EVENT: Disable good-tm-iport-array-putnth-into-tm-iports.

THEOREM: good-tm-oport-getnth
(good-tm-oport-array (l) \wedge ($i < \text{length}(l)$))
 \rightarrow good-tm-oport (getnth (i , l))

EVENT: Disable good-tm-oport-getnth.

THEOREM: good-tm-oport-array-putnth
(good-tm-oport-array (l) \wedge good-tm-oport ($oport$))
 \rightarrow good-tm-oport-array (putnth ($oport$, n , l))

EVENT: Disable good-tm-oport-array-putnth.

THEOREM: good-tm-oport-getnth-from-tm-oports
(good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow good-tm-oport (getnth (id , tm-oports (tm)))

EVENT: Disable good-tm-oport-getnth-from-tm-oports.

THEOREM: good-tm-oport-array-putnth-into-tm-oports
(good-tm (tm) \wedge good-tm-oport ($oport$))
 \rightarrow good-tm-oport-array (putnth ($oport$, n , tm-oports (tm)))

EVENT: Disable good-tm-oport-array-putnth-into-tm-oports.

THEOREM: size-of-tm-interrupting-input-port
tm-some-input-interruptp (l)
 \rightarrow (tm-interrupting-input-port (l) < length (l))

EVENT: Disable size-of-tm-interrupting-input-port.

THEOREM: finite-numberp-tm-interrupting-input-port
(good-tm (tm) \wedge tm-some-input-interruptp (tm-iports (tm)))
 \rightarrow finite-numberp (tm-interrupting-input-port (tm-iports (tm)), '16)

EVENT: Disable finite-numberp-tm-interrupting-input-port.

THEOREM: finite-numberp-tm-interrupting-input-port-tm-wordlub
(good-tm (tm) \wedge tm-some-input-interruptp (tm-iports (tm)))
 \rightarrow finite-numberp (tm-interrupting-input-port (tm-iports (tm)), '65536)

EVENT: Disable finite-numberp-tm-interrupting-input-port-tm-wordlub.

THEOREM: good-tm-iport-getnth-properties
(good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow (tm-iportp (getnth (id , tm-iports (tm)))
 \wedge (finite-numberp (tm-iinterrupt-flag (getnth (id , tm-iports (tm))),
'2)
 \wedge (finite-numberp (tm-ierror-flag (getnth (id ,
tm-iports (tm))),
'2)
 \wedge finite-numberp (tm-ichar (getnth (id ,
tm-iports (tm))),
'256))))))

EVENT: Disable good-tm-iport-getnth-properties.

THEOREM: size-of-tm-interrupting-output-port
tm-some-output-interruptp (l)
 \rightarrow (tm-interrupting-output-port (l) < length (l))

EVENT: Disable size-of-tm-interrupting-output-port.

THEOREM: finite-numberp-tm-interrupting-output-port
(good-tm (tm) \wedge tm-some-output-interruptp (tm-oports (tm)))
 \rightarrow finite-numberp (tm-interrupting-output-port (tm-oports (tm)), '16)

EVENT: Disable finite-numberp-tm-interrupting-output-port.

THEOREM: finite-numberp-tm-interrupting-output-port-tm-wordlub
 (good-tm (tm) \wedge tm-some-output-interruptp (tm-oports (tm)))
 \rightarrow finite-numberp (tm-interrupting-output-port (tm-oports (tm)), '65536)

EVENT: Disable finite-numberp-tm-interrupting-output-port-tm-wordlub.

THEOREM: good-tm-oport-getnth-properties
 (good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow (tm-oportp (getnth (id , tm-oports (tm)))
 \wedge (finite-numberp (tm-ointerrupt-flag (getnth (id , tm-oports (tm))),
 '2)
 \wedge (finite-numberp (tm-obusy-flag (getnth (id , tm-oports (tm))),
 '2)
 \wedge finite-numberp (tm-ochar (getnth (id ,
 tm-oports (tm))),
 '256))))))

EVENT: Disable good-tm-oport-getnth-properties.

EVENT: Let us define the theory *tm-shells* to consist of the following events:
 tm-device-output-event, tm-device-input-event, tm-oport, tm-iport, tm.

THEOREM: good-tm-iports-tm-clear-input-interrupt
 (good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow (plistp (tm-clear-input-interrupt (id , tm-iports (tm)))
 \wedge ((length (tm-clear-input-interrupt (id , tm-iports (tm))) = '16)
 \wedge good-tm-iport-array (tm-clear-input-interrupt (id ,
 tm-iports (tm))))))

EVENT: Disable good-tm-iports-tm-clear-input-interrupt.

THEOREM: good-tm-oports-tm-clear-output-interrupt
 (good-tm (tm) \wedge finite-numberp (id , '16))
 \rightarrow (plistp (tm-clear-output-interrupt (id , tm-oports (tm)))
 \wedge ((length (tm-clear-output-interrupt (id , tm-oports (tm)))
 = '16)
 \wedge good-tm-oport-array (tm-clear-output-interrupt (id ,
 tm-oports (tm))))))

EVENT: Disable good-tm-oports-tm-clear-output-interrupt.

THEOREM: finite-numberp-tm-ichar-getnth-crock

$(\text{good-tm}(tm) \wedge \text{finite-numberp}(id, '16))$
 $\rightarrow \text{finite-numberp}(\text{tm-ichar}(\text{getnth}(id, \text{tm-iports}(tm))), '65536)$

EVENT: Disable finite-numberp-tm-ichar-getnth-crock.

THEOREM: finite-numberp-tm-overflow-char
 $\text{finite-numberp}(char, '256)$
 $\rightarrow \text{finite-numberp}(\text{tm-overflow-char}(char), '512)$

THEOREM: finite-numberp-tm-overflow-char-tm-wordlub
 $\text{finite-numberp}(char, '256)$
 $\rightarrow \text{finite-numberp}(\text{tm-overflow-char}(char), '65536)$

EVENT: Disable finite-numberp-tm-overflow-char-tm-wordlub.

THEOREM: tm-iport-error-flag-is-zero
 $(\text{good-tm}(tm)$
 $\wedge ((id \in \mathbf{N})$
 $\wedge ((id < '16) \wedge (\neg \text{tm-iport-errorp}(id, \text{tm-iports}(tm))))))$
 $\rightarrow (\text{tm-ierror-flag}(\text{getnth}(id, \text{tm-iports}(tm))) = '0)$

EVENT: Disable tm-iport-error-flag-is-zero.

THEOREM: finite-numberp-tm-interrupting-input-port-crock
 $(\text{good-tm}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow ((\text{tm-interrupting-input-port}(\text{tm-iports}(os)) \in \mathbf{N})$
 $\wedge (\text{tm-interrupting-input-port}(\text{tm-iports}(os)) < '16))$

EVENT: Disable finite-numberp-tm-interrupting-input-port-crock.

THEOREM: tm-ichar-facts
 $(\text{good-tm}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow ((('256$
 $\quad + \text{tm-ichar}(\text{getnth}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \quad \quad \text{tm-iports}(os))))$
 $\quad < '65536)$
 $\quad = '*1*true)$

EVENT: Disable tm-ichar-facts.

THEOREM: finite-numberp-tm-interrupting-output-port-crock
 $(\text{good-tm}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow ((\text{tm-interrupting-output-port}(\text{tm-oports}(os)) \in \mathbf{N})$
 $\wedge (\text{tm-interrupting-output-port}(\text{tm-oports}(os)) < '16))$

EVENT: Disable finite-numberp-tm-interrupting-output-port-crock.

DEFINITION: TM-CPU-LENGTH = '9

DEFINITION: tm-r0(*tm*) = getnth('0, tm-regs(*tm*))

DEFINITION: tm-r1(*tm*) = getnth('1, tm-regs(*tm*))

DEFINITION: tm-r2(*tm*) = getnth('2, tm-regs(*tm*))

DEFINITION: tm-r3(*tm*) = getnth('3, tm-regs(*tm*))

DEFINITION: tm-r4(*tm*) = getnth('4, tm-regs(*tm*))

DEFINITION: tm-r5(*tm*) = getnth('5, tm-regs(*tm*))

DEFINITION: tm-r6(*tm*) = getnth('6, tm-regs(*tm*))

DEFINITION: tm-r7(*tm*) = getnth('7, tm-regs(*tm*))

DEFINITION:

```
tm-cpu(tm)
= cons(tm-r0(tm),
      cons(tm-r1(tm),
          cons(tm-r2(tm),
              cons(tm-r3(tm),
                  cons(tm-r4(tm),
                      cons(tm-r5(tm),
                          cons(tm-r6(tm),
                              cons(tm-r7(tm),
                                  cons(tm-pack-psw(tm-cc(tm),
                                      tm-error(tm),
                                      tm-svcflag(tm),
                                      tm-svcid(tm)),
                                      'nil))))))))))
```

DEFINITION:

```
good-cpu(cpu)
= (plistp(cpu)
   ^ ((length(cpu) = '9) ^ finite-number-listp(cpu, '65536)))
```

DEFINITION:

```
good-cpu-list(l)
= if listp(l) then good-cpu(car(l)) ^ good-cpu-list(cdr(l))
  else '*1*true endif
```


THEOREM: open-up-finite-number-listp
finite-number-listp (cons (a, b), lub)
= (finite-numberp (a, lub) \wedge finite-number-listp (b, lub))

EVENT: Disable open-up-finite-number-listp.

THEOREM: finite-number-listp-on-non-list
(\neg listp (x)) \rightarrow (finite-number-listp (x, lub) = '1*true)

EVENT: Disable finite-number-listp-on-non-list.

THEOREM: length-tm-cpu
good-tm (tm) \rightarrow (length (tm-cpu (tm)) = '9)

EVENT: Disable length-tm-cpu.

THEOREM: plistp-tm-cpu
plistp (tm-cpu (tm))

EVENT: Disable plistp-tm-cpu.

THEOREM: finite-number-listp-tm-cpu
good-tm (tm) \rightarrow finite-number-listp (tm-cpu (tm), '65536)

EVENT: Disable finite-number-listp-tm-cpu.

THEOREM: good-cpu-tm-cpu
good-tm (tm) \rightarrow good-cpu (tm-cpu (tm))

EVENT: Disable good-cpu-tm-cpu.

THEOREM: good-cpu-getnth
(good-cpu-list (l) \wedge (n < length (l))) \rightarrow good-cpu (getnth (n, l))

EVENT: Disable good-cpu-getnth.

THEOREM: good-cpu-list-putnth
(good-cpu-list (l) \wedge good-cpu (cpu)) \rightarrow good-cpu-list (putnth (cpu, i, l))

EVENT: Disable good-cpu-list-putnth.

THEOREM: list-of-getnths8
 $(\text{plistp}(l) \wedge (\text{length}(l) = '8))$
 \rightarrow (cons (getnth ('0, l),
 cons (getnth ('1, l),
 cons (getnth ('2, l),
 cons (getnth ('3, l),
 cons (getnth ('4, l),
 cons (getnth ('5, l),
 cons (getnth ('6, l),
 cons (getnth ('7, l), 'nil))))))))))
= l)

EVENT: Disable list-of-getnths8.

THEOREM: cpu-regs-segment
 $\text{good-tm}(tm) \rightarrow (\text{getseg}('0, '8, \text{tm-cpu}(tm)) = \text{tm-regs}(tm))$

EVENT: Disable cpu-regs-segment.

EVENT: Let us define the theory *good-tm-properties* to consist of the following events: cpu-regs-segment, good-cpu-list-putnth, good-cpu-getnth, good-cpu-tm-cpu, finite-numberp-tm-pack-psw, finite-number-listp-tm-cpu, numberp-getnth-from-tm-regs, numberp-getnth-from-tm-memory, finite-numberp-tm-pc, finite-numberp-tm-sp, finite-numberp-tm-cc, finite-numberp-tm-error, finite-numberp-tm-svcflag, finite-numberp-tm-svcid, finite-numberp-tm-rwstate, plistp-tm-iports, length-tm-iports, good-tm-iport-array-tm-iports, plistp-tm-oports, length-tm-oports, good-tm-oport-array-tm-oports, good-tm-iport-getnth-from-tm-iports, good-tm-iport-array-putnth-into-tm-iports, good-tm-oport-getnth-from-tm-oports, good-tm-oport-array-putnth-into-tm-oports, finite-numberp-tm-clock, finite-numberp-tm-svmode, finite-numberp-tm-slimit, finite-numberp-tm-limit, finite-numberp-tm-base, finite-number-listp-tm-regs, length-tm-regs, plistp-tm-regs, finite-number-listp-tm-memory, length-tm-memory, plistp-tm-memory, tm-shellp-tm, tm-fix-reg-address-noop, tm-fix-reg-address-size-linear, tm-fix-reg-address-size, numberp-tm-fix-reg-address, tm-fix-word-noop, tm-fix-word-size-linear, tm-fix-word-size, numberp-tm-fix-word, tm-cc-value-on-compare-lessp, tm-cc-value-on-compare-greaterp, finite-numberp-tm-pack-psw, finite-numberp-tm-overflow-char-tm-wordlub, finite-numberp-tm-ichar-getnth-crock, finite-numberp-tm-interrupting-input-port, finite-numberp-tm-interrupting-input-port-tm-wordlub, good-tm-iport-getnth-properties, finite-numberp-tm-interrupting-output-port, finite-numberp-tm-interrupting-output-port-tm-wordlub, good-tm-oport-getnth-properties.

EVENT: Let us define the theory *tm-definition* to consist of the following events: tm-processor, tm-step, tm-post-interrupt, tm-execute-output-interrupt, tm-store-

interrupting-output-device, tm-output-interruptp, tm-execute-input-interrupt, tm-store-interrupting-input-device, tm-input-interruptp, tm-overflow-char, tm-execute-svc-interrupt, tm-store-svcid-on-interrupt, tm-svc-interruptp, tm-execute-error-interrupt, tm-errorp, tm-execute-clock-interrupt, tm-clock-interruptp, tm-store-old-psw-on-interrupt, tm-fetch-new-pc-on-interrupt, tm-fetch-execute, tm-good-pc-address, tm-decrement-clock, tm-execute, tm-fetch-arg2, tm-fetch-arg1, tm-fetch-opcode, tm-execute-instruction, tm-execute-binary, tm-execute-unary, tm-execute-nullary, tm-arg2, tm-arg1, tm-datum2, tm-datum1, tm-unary-instructionp, tm-nullary-instructionp, tm-index2, tm-mode2, tm-index1, tm-mode1, tm-opcode, tm-execute-svc-return, tm-execute-svc, tm-execute-start-output, tm-execute-test-iport, tm-execute-test-oport, tm-execute-post-output-interrupt, tm-execute-wait, tm-execute-set-clock, tm-execute-run, tm-execute-return, tm-execute-multiply, tm-execute-move, tm-execute-mod, tm-execute-load-psw, tm-execute-load-limit, tm-execute-load-base, tm-execute-incr-mod, tm-execute-incr, tm-execute-decr-mod, tm-execute-decr, tm-execute-compare, tm-execute-call, tm-execute-branch-not-zero, tm-execute-branch-on-zero, tm-execute-branch, tm-execute-add, tm-stack-underflowp, tm-stack-overflowp, tm-store, tm-fetch, tm-store-in-regmem, tm-fetch-from-regmem, tm-store-in-memory, tm-fetch-from-memory, tm-incrn-address, tm-incr-address, tm-good-address, tm-compute-address, tm-arg-index, tm-arg-datum, tm-arg-mode, real-addr-num, real-addr-source, real-addr, tm-clear-output-interrupt, tm-post-output-interrupt, tm-start-output, tm-oport-idlep, tm-post-input-interrupt, tm-clear-input-interrupt, tm-iport-errorp, tm-no-svc, tm-svc, tm-waiting, tm-in-supervisor-mode, tm-user-mode, tm-supervisor-over, tm-pc-address-error, tm-address-error, tm-stack-underflow-error, tm-stack-overflow-error, tm-privilege-error, tm-opcode-error, tm-halt-status, tm-no-error, tm-unpack-svcid, tm-unpack-svcflag, tm-unpack-error, tm-unpack-cc, tm-pack-psw, tm-zero-carry-condition, tm-non-zero-carry-condition, tm-zero-no-carry-condition, tm-non-zero-no-carry-condition, tm-set-sp, tm-sp, tm-set-pc, tm-pc, tm-set-oports, tm-set-iports, tm-set-clock, tm-set-rwstate, tm-set-svmode, tm-set-slimit, tm-set-limit, tm-set-base, tm-set-svcid, tm-set-svcflag, tm-set-error, tm-set-cc, tm-set-regs, tm-set-memory, tm-decr, tm-incrn, tm-incr, tm-r0, tm-r1, tm-r2, tm-r3, tm-r4, tm-r5, tm-r6, tm-r7, tm-shells.

THEOREM: finite-numberp-tm-fetch-from-memory-in-supervisor-mode
 $(\text{good-tm}(tm) \wedge (\text{tm-in-supervisor-mode}(tm) \wedge \text{finite-numberp}(n, '65536)))$
 $\rightarrow \text{finite-numberp}(\text{tm-fetch-from-memory}(n, tm), '65536)$

EVENT: Disable finite-numberp-tm-fetch-from-memory-in-supervisor-mode.

THEOREM: finite-numberp-min-difference-memlength-base-limit
 $\text{good-tm}(tm)$
 $\rightarrow \text{finite-numberp}(\min('65536 - \text{tm-base}(tm), \text{tm-limit}(tm)), '65536)$

EVENT: Disable finite-numberp-min-difference-memlength-base-limit.

THEOREM: lessp-plus-base-addr-memlength
 (good-tm (*tm*)
 ∧ ((real-addr-source (*addr*) ≠ 0)
 ∧ ((real-addr-source (*addr*) ≠ '1)
 ∧ ((¬ tm-in-supervisor-mode (*tm*)
 ∧ tm-good-address (*addr*, *tm*))))))
 → ((tm-base (*tm*) + real-addr-num (*addr*) < '65536)

EVENT: Disable lessp-plus-base-addr-memlength.

THEOREM: finite-numberp-tm-fetch
 (good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*)
 → finite-numberp (tm-fetch (*addr*, *tm*), '65536)

EVENT: Disable finite-numberp-tm-fetch.

(PROVE-LEMMA
 TM-STORE-PRESERVES-ALL-BUT-MEMORY-AND-REGS
 (REWRITE)
 (AND
 (EQUAL (TM-CC (TM-STORE VALUE ADDR TM))
 (TM-CC TM))
 (AND
 (EQUAL (TM-ERROR (TM-STORE VALUE ADDR TM))
 (TM-ERROR TM))
 (AND
 (EQUAL (TM-BASE (TM-STORE VALUE ADDR TM))
 (TM-BASE TM))
 (AND
 (EQUAL (TM-LIMIT (TM-STORE VALUE ADDR TM))
 (TM-LIMIT TM))
 (AND
 (EQUAL (TM-SLIMIT (TM-STORE VALUE ADDR TM))
 (TM-SLIMIT TM))
 (AND (EQUAL (TM-SVMODE (TM-STORE VALUE ADDR TM))
 (TM-SVMODE TM))
 (AND (EQUAL (TM-SVCFLAG (TM-STORE VALUE ADDR TM))
 (TM-SVCFLAG TM))
 (AND (EQUAL (TM-SVCID (TM-STORE VALUE ADDR TM))
 (TM-SVCID TM))
 (AND (EQUAL (TM-RWSTATE (TM-STORE VALUE ADDR TM))

```

(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-STORE VALUE ADDR TM))
            (TM-CLOCK TM))
      (AND (EQUAL (TM-IPOINTS (TM-STORE VALUE ADDR TM))
                  (TM-IPOINTS TM))
            (EQUAL (TM-OPOINTS (TM-STORE VALUE ADDR TM))
                    (TM-OPOINTS TM)))))))))
((ENABLE TM-STORE-IN-MEMORY TM-STORE-IN-REGMEM TM-SET-REGS TM-SET-MEMORY
         TM-STORE)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable tm-store-preserves-all-but-memory-and-regs.

```

THEOREM: tm-good-memory-tm-store
(good-tm (tm) ∧ finite-numberp (value, '65536))
→ (plistp (tm-memory (tm-store (value, addr, tm)))
     ∧ ((length (tm-memory (tm-store (value, addr, tm))) = '65536)
        ∧ finite-number-listp (tm-memory (tm-store (value, addr, tm)),
                                '65536)))

```

EVENT: Disable tm-good-memory-tm-store.

```

THEOREM: tm-good-regs-tm-store
(good-tm (tm) ∧ finite-numberp (value, '65536))
→ (plistp (tm-regs (tm-store (value, addr, tm)))
     ∧ ((length (tm-regs (tm-store (value, addr, tm))) = '8)
        ∧ finite-number-listp (tm-regs (tm-store (value, addr, tm)),
                                '65536)))

```

EVENT: Disable tm-good-regs-tm-store.

```

THEOREM: good-tm-tm-store
(good-tm (tm) ∧ finite-numberp (value, '65536))
→ good-tm (tm-store (value, addr, tm))

```

EVENT: Disable good-tm-tm-store.

```

THEOREM: good-tm-tm-set-regs
(good-tm (tm)
  ∧ (finite-number-listp (regs, '65536)
     ∧ (plistp (regs) ∧ (length (regs) = '8))))
→ good-tm (tm-set-regs (regs, tm))

```

EVENT: Disable good-tm-tm-set-regs.

```
(PROVE-LEMMA
ACCESS-TM-SET-REGS
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-REGS REGS TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-REGS REGS TM))
REGS)
(AND
(EQUAL (TM-CC (TM-SET-REGS REGS TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-REGS REGS TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-REGS REGS TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-REGS REGS TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-REGS REGS TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-REGS REGS TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-REGS REGS TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-SET-REGS REGS TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-REGS REGS TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-REGS REGS TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPORTS (TM-SET-REGS REGS TM))
(TM-IPORTS TM))
```

```

(AND (EQUAL (TM-OPOINTS (TM-SET-REGS REGS TM))
            (TM-OPOINTS TM))
      (AND (EQUAL (TM-PC (TM-SET-REGS REGS TM))
                  (GETNTH '0 REGS))
            (EQUAL (TM-SP (TM-SET-REGS REGS TM))
                    (GETNTH '1 REGS)))))))))
(ENABLE TM-SET-REGS TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-regs.

```

THEOREM: good-tm-tm-set-memory
(good-tm (tm)
  ∧ (finite-number-listp (memory, '65536)
    ∧ (plistp (memory) ∧ (length (memory) = '65536))))
→ good-tm (tm-set-memory (memory, tm))

```

EVENT: Disable good-tm-tm-set-memory.

```

(PROVE-LEMMA
ACCESS-TM-SET-MEMORY
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-MEMORY MEMORY TM))
        MEMORY)
(AND
(EQUAL (TM-REGS (TM-SET-MEMORY MEMORY TM))
        (TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-MEMORY MEMORY TM))
        (TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-MEMORY MEMORY TM))
        (TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-MEMORY MEMORY TM))
        (TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-MEMORY MEMORY TM))
        (TM-LIMIT TM))

```

```

(AND
  (EQUAL (TM-SLIMIT (TM-SET-MEMORY MEMORY TM))
    (TM-SLIMIT TM))
  (AND
    (EQUAL (TM-SVMODE (TM-SET-MEMORY MEMORY TM))
      (TM-SVMODE TM))
    (AND
      (EQUAL (TM-SVCFLAG (TM-SET-MEMORY MEMORY TM))
        (TM-SVCFLAG TM))
      (AND
        (EQUAL (TM-SVCID (TM-SET-MEMORY MEMORY TM))
          (TM-SVCID TM))
        (AND (EQUAL (TM-RWSTATE (TM-SET-MEMORY MEMORY TM))
          (TM-RWSTATE TM))
          (AND (EQUAL (TM-CLOCK (TM-SET-MEMORY MEMORY TM))
            (TM-CLOCK TM))
            (AND (EQUAL (TM-IPOINTS (TM-SET-MEMORY MEMORY TM))
              (TM-IPOINTS TM))
              (AND (EQUAL (TM-OPOINTS (TM-SET-MEMORY MEMORY TM))
                (TM-OPOINTS TM))
                (AND (EQUAL (TM-PC (TM-SET-MEMORY MEMORY TM))
                  (TM-PC TM))
                  (EQUAL (TM-SP (TM-SET-MEMORY MEMORY TM))
                    (TM-SP TM))))))))))))))
  ((ENABLE TM-SET-MEMORY TM-PC TM-SP)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-memory.

THEOREM: good-tm-tm-set-cc

$(\text{good-tm}(tm) \wedge \text{finite-numberp}(cc, '4)) \rightarrow \text{good-tm}(\text{tm-set-cc}(cc, tm))$

EVENT: Disable good-tm-tm-set-cc.

```

(PROVE-LEMMA
  ACCESS-TM-SET-CC
  (REWRITE)
  (AND
    (EQUAL (TM-MEMORY (TM-SET-CC CC TM))
      (TM-MEMORY TM))

```



```

(AND
  (EQUAL (TM-REGS (TM-SET-CC CC TM))
    (TM-REGS TM))
  (AND
    (EQUAL (TM-CC (TM-SET-CC CC TM)) CC)
    (AND
      (EQUAL (TM-ERROR (TM-SET-CC CC TM))
        (TM-ERROR TM))
      (AND
        (EQUAL (TM-BASE (TM-SET-CC CC TM))
          (TM-BASE TM))
        (AND
          (EQUAL (TM-LIMIT (TM-SET-CC CC TM))
            (TM-LIMIT TM))
          (AND
            (EQUAL (TM-SLIMIT (TM-SET-CC CC TM))
              (TM-SLIMIT TM))
            (AND
              (EQUAL (TM-SVMODE (TM-SET-CC CC TM))
                (TM-SVMODE TM))
              (AND (EQUAL (TM-SVCFLAG (TM-SET-CC CC TM))
                (TM-SVCFLAG TM))
                (AND (EQUAL (TM-SVCID (TM-SET-CC CC TM))
                  (TM-SVCID TM))
                  (AND (EQUAL (TM-RWSTATE (TM-SET-CC CC TM))
                    (TM-RWSTATE TM))
                    (AND (EQUAL (TM-CLOCK (TM-SET-CC CC TM))
                      (TM-CLOCK TM))
                      (AND (EQUAL (TM-IPOINTS (TM-SET-CC CC TM))
                        (TM-IPOINTS TM))
                        (AND (EQUAL (TM-OPOINTS (TM-SET-CC CC TM))
                          (TM-OPOINTS TM))
                          (AND (EQUAL (TM-PC (TM-SET-CC CC TM))
                            (TM-PC TM))
                            (EQUAL (TM-SP (TM-SET-CC CC TM))
                              (TM-SP TM))))))))))))))))))
    ((ENABLE TM-SET-CC TM-PC TM-SP)
     (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
     (DISABLE-THEORY T)))

```

EVENT: Disable access-tm-set-cc.

THEOREM: good-tm-tm-set-error
 (good-tm (*tm*) \wedge finite-numberp (*error*, '64))

→ good-tm (tm-set-error (*error*, *tm*))

EVENT: Disable good-tm-tm-set-error.

```
(PROVE-LEMMA
ACCESS-TM-SET-ERROR
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-ERROR ERROR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-ERROR ERROR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-ERROR ERROR TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-ERROR ERROR TM))
ERROR)
(AND
(EQUAL (TM-BASE (TM-SET-ERROR ERROR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-ERROR ERROR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-ERROR ERROR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-ERROR ERROR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-ERROR ERROR TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-SET-ERROR ERROR TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-ERROR ERROR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-ERROR ERROR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-SET-ERROR ERROR TM))
(TM-IPOINTS TM))
```

```

(AND (EQUAL (TM-OPOINTS (TM-SET-ERROR ERROR TM))
            (TM-OPOINTS TM))
      (AND (EQUAL (TM-PC (TM-SET-ERROR ERROR TM))
                (TM-PC TM))
            (EQUAL (TM-SP (TM-SET-ERROR ERROR TM))
                    (TM-SP TM)))))))))
(ENABLE TM-SET-ERROR TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-error.

THEOREM: good-tm-tm-set-base
 (good-tm (*tm*) \wedge finite-numberp (*base*, '65536))
 \rightarrow good-tm (tm-set-base (*base*, *tm*))

EVENT: Disable good-tm-tm-set-base.

```

(PROVE-LEMMA
 ACCESS-TM-SET-BASE
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-SET-BASE BASE TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-SET-BASE BASE TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-SET-BASE BASE TM))
           (TM-CC TM))
    (AND
     (EQUAL (TM-ERROR (TM-SET-BASE BASE TM))
            (TM-ERROR TM))
     (AND
      (EQUAL (TM-BASE (TM-SET-BASE BASE TM))
             BASE)
      (AND
       (EQUAL (TM-LIMIT (TM-SET-BASE BASE TM))
              (TM-LIMIT TM))
       (AND
        (EQUAL (TM-SLIMIT (TM-SET-BASE BASE TM))
               (TM-SLIMIT TM))

```

```

(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-BASE BASE TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-BASE BASE TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-SET-BASE BASE TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-BASE BASE TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-BASE BASE TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-SET-BASE BASE TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-SET-BASE BASE TM))
(TM-OPOINTS TM))
(AND (EQUAL (TM-PC (TM-SET-BASE BASE TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-SET-BASE BASE TM))
(TM-SP TM))))))))))))))
((ENABLE TM-SET-BASE TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-base.

THEOREM: good-tm-tm-set-limit
 $(\text{good-tm}(tm) \wedge \text{finite-numberp}(limit, '65536))$
 $\rightarrow \text{good-tm}(\text{tm-set-limit}(limit, tm))$

EVENT: Disable good-tm-tm-set-limit.

```

(PROVE-LEMMA
ACCESS-TM-SET-LIMIT
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-LIMIT LIMIT TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-LIMIT LIMIT TM))

```

```

(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-LIMIT LIMIT TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-LIMIT LIMIT TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-LIMIT LIMIT TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-LIMIT LIMIT TM))
LIMIT)
(AND
(EQUAL (TM-SLIMIT (TM-SET-LIMIT LIMIT TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-LIMIT LIMIT TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-LIMIT LIMIT TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-SET-LIMIT LIMIT TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-LIMIT LIMIT TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-LIMIT LIMIT TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-SET-LIMIT LIMIT TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPORTS (TM-SET-LIMIT LIMIT TM))
(TM-OPORTS TM))
(AND (EQUAL (TM-PC (TM-SET-LIMIT LIMIT TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-SET-LIMIT LIMIT TM))
(TM-SP TM))))))))))))))
((ENABLE TM-SET-LIMIT TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-limit.

THEOREM: good-tm-tm-set-slimit
(good-tm (*tm*) \wedge finite-numberp (*slimit*, '65536))

→ good-tm (tm-set-slimit (*slimit*, *tm*))

EVENT: Disable good-tm-tm-set-slimit.

```
(PROVE-LEMMA
ACCESS-TM-SET-SLIMIT
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-SLIMIT SLIMIT TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-SLIMIT SLIMIT TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-SLIMIT SLIMIT TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-SLIMIT SLIMIT TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-SLIMIT SLIMIT TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-SLIMIT SLIMIT TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-SLIMIT SLIMIT TM))
SLIMIT)
(AND
(EQUAL (TM-SVMODE (TM-SET-SLIMIT SLIMIT TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-SLIMIT SLIMIT TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-SET-SLIMIT SLIMIT TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-SLIMIT SLIMIT TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-SLIMIT SLIMIT TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPORTS (TM-SET-SLIMIT SLIMIT TM))
```

```

(TM-IPORTS TM))
(AND (EQUAL (TM-OPOINTS (TM-SET-SLIMIT SLIMIT TM))
            (TM-OPOINTS TM))
      (AND (EQUAL (TM-PC (TM-SET-SLIMIT SLIMIT TM))
                  (TM-PC TM))
            (EQUAL (TM-SP (TM-SET-SLIMIT SLIMIT TM))
                    (TM-SP TM)))))))))
((ENABLE TM-SET-SLIMIT TM-PC TM-SP)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-slimit.

THEOREM: good-tm-tm-set-svmode
 $(\text{good-tm}(tm) \wedge \text{finite-numberp}(svmode, '2))$
 $\rightarrow \text{good-tm}(\text{tm-set-svmode}(svmode, tm))$

EVENT: Disable good-tm-tm-set-svmode.

```

(PROVE-LEMMA
 ACCESS-TM-SET-SVMODE
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-SET-SVMODE SVMODE TM))
          (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-SET-SVMODE SVMODE TM))
           (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-SET-SVMODE SVMODE TM))
            (TM-CC TM))
    (AND
     (EQUAL (TM-ERROR (TM-SET-SVMODE SVMODE TM))
             (TM-ERROR TM))
     (AND
      (EQUAL (TM-BASE (TM-SET-SVMODE SVMODE TM))
              (TM-BASE TM))
      (AND
       (EQUAL (TM-LIMIT (TM-SET-SVMODE SVMODE TM))
               (TM-LIMIT TM))
       (AND
        (

```

```

(EQUAL (TM-SLIMIT (TM-SET-SVMODE SVMODE TM))
 (TM-SLIMIT TM))
(AND
 (EQUAL (TM-SVMODE (TM-SET-SVMODE SVMODE TM))
 SVMODE)
 (AND
 (EQUAL (TM-SVCFLAG (TM-SET-SVMODE SVMODE TM))
 (TM-SVCFLAG TM))
 (AND
 (EQUAL (TM-SVCID (TM-SET-SVMODE SVMODE TM))
 (TM-SVCID TM))
 (AND (EQUAL (TM-RWSTATE (TM-SET-SVMODE SVMODE TM))
 (TM-RWSTATE TM))
 (AND (EQUAL (TM-CLOCK (TM-SET-SVMODE SVMODE TM))
 (TM-CLOCK TM))
 (AND (EQUAL (TM-IPORTS (TM-SET-SVMODE SVMODE TM))
 (TM-IPORTS TM))
 (AND (EQUAL (TM-OPORTS (TM-SET-SVMODE SVMODE TM))
 (TM-OPORTS TM))
 (AND (EQUAL (TM-PC (TM-SET-SVMODE SVMODE TM))
 (TM-PC TM))
 (EQUAL (TM-SP (TM-SET-SVMODE SVMODE TM))
 (TM-SP TM))))))))))))))
((ENABLE TM-SET-SVMODE TM-PC TM-SP)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-svmode.

THEOREM: good-tm-tm-set-svcflag
 $(\text{good-tm}(tm) \wedge \text{finite-numberp}(svcflag, '2))$
 $\rightarrow \text{good-tm}(\text{tm-set-svcflag}(svcflag, tm))$

EVENT: Disable good-tm-tm-set-svcflag.

```

(PROVE-LEMMA
 ACCESS-TM-SET-SVCFLAG
 (REWRITE)
 (AND
 (EQUAL (TM-MEMORY (TM-SET-SVCFLAG SVCFLAG TM))
 (TM-MEMORY TM))

```



```

(AND
  (EQUAL (TM-REGS (TM-SET-SVCFLAG SVCFLAG TM))
    (TM-REGS TM))
  (AND
    (EQUAL (TM-CC (TM-SET-SVCFLAG SVCFLAG TM))
      (TM-CC TM))
    (AND
      (EQUAL (TM-ERROR (TM-SET-SVCFLAG SVCFLAG TM))
        (TM-ERROR TM))
      (AND
        (EQUAL (TM-BASE (TM-SET-SVCFLAG SVCFLAG TM))
          (TM-BASE TM))
        (AND
          (EQUAL (TM-LIMIT (TM-SET-SVCFLAG SVCFLAG TM))
            (TM-LIMIT TM))
          (AND
            (EQUAL (TM-SLIMIT (TM-SET-SVCFLAG SVCFLAG TM))
              (TM-SLIMIT TM))
            (AND
              (EQUAL (TM-SVMODE (TM-SET-SVCFLAG SVCFLAG TM))
                (TM-SVMODE TM))
              (AND
                (EQUAL (TM-SVCFLAG (TM-SET-SVCFLAG SVCFLAG TM))
                  SVCFLAG)
                (AND
                  (EQUAL (TM-SVCID (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-SVCID TM))
                  (AND (EQUAL (TM-RWSTATE (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-RWSTATE TM))
                    (AND (EQUAL (TM-CLOCK (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-CLOCK TM))
                    (AND (EQUAL (TM-IPOINTS (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-IPOINTS TM))
                    (AND (EQUAL (TM-OPOINTS (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-OPOINTS TM))
                    (AND (EQUAL (TM-PC (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-PC TM))
                    (EQUAL (TM-SP (TM-SET-SVCFLAG SVCFLAG TM))
                    (TM-SP TM))))))))))))))))))
  ((ENABLE TM-SET-SVCFLAG TM-PC TM-SP)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-svcflag.

THEOREM: good-tm-tm-set-svcid
(good-tm (*tm*) \wedge finite-numberp (*svcid*, '128))
→ good-tm (tm-set-svcid (*svcid*, *tm*))

EVENT: Disable good-tm-tm-set-svcid.

```
(PROVE-LEMMA
ACCESS-TM-SET-SVCID
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-SVCID SVCID TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-SVCID SVCID TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-SVCID SVCID TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-SVCID SVCID TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-SVCID SVCID TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-SVCID SVCID TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-SVCID SVCID TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-SVCID SVCID TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-SVCID SVCID TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-SET-SVCID SVCID TM))
SVCID)
(AND (EQUAL (TM-RWSTATE (TM-SET-SVCID SVCID TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-SVCID SVCID TM))
(TM-CLOCK TM))
```

```

(AND (EQUAL (TM-IPOINTS (TM-SET-SVCID SVCID TM))
            (TM-IPOINTS TM))
      (AND (EQUAL (TM-OPOINTS (TM-SET-SVCID SVCID TM))
                (TM-OPOINTS TM))
            (AND (EQUAL (TM-PC (TM-SET-SVCID SVCID TM))
                      (TM-PC TM))
                  (EQUAL (TM-SP (TM-SET-SVCID SVCID TM))
                        (TM-SP TM))))))))))
(ENABLE TM-SET-SVCID TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-svcid.

THEOREM: good-tm-tm-set-rwstate
 $(\text{good-tm}(tm) \wedge \text{finite-numberp}(rwstate, '2))$
 $\rightarrow \text{good-tm}(\text{tm-set-rwstate}(rwstate, tm))$

EVENT: Disable good-tm-tm-set-rwstate.

```

(PROVE-LEMMA
 ACCESS-TM-SET-RWSTATE
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-SET-RWSTATE RWSTATE TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-SET-RWSTATE RWSTATE TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-SET-RWSTATE RWSTATE TM))
           (TM-CC TM))
    (AND
     (EQUAL (TM-ERROR (TM-SET-RWSTATE RWSTATE TM))
            (TM-ERROR TM))
     (AND
      (EQUAL (TM-BASE (TM-SET-RWSTATE RWSTATE TM))
             (TM-BASE TM))
      (AND
       (EQUAL (TM-LIMIT (TM-SET-RWSTATE RWSTATE TM))
              (TM-LIMIT TM))

```

```

(AND
  (EQUAL (TM-SLIMIT (TM-SET-RWSTATE RWSTATE TM))
    (TM-SLIMIT TM))
  (AND
    (EQUAL (TM-SVMODE (TM-SET-RWSTATE RWSTATE TM))
      (TM-SVMODE TM))
    (AND
      (EQUAL (TM-SVCFLAG (TM-SET-RWSTATE RWSTATE TM))
        (TM-SVCFLAG TM))
      (AND
        (EQUAL (TM-SVCID (TM-SET-RWSTATE RWSTATE TM))
          (TM-SVCID TM))
        (AND (EQUAL (TM-RWSTATE (TM-SET-RWSTATE RWSTATE TM))
          RWSTATE)
          (AND (EQUAL (TM-CLOCK (TM-SET-RWSTATE RWSTATE TM))
            (TM-CLOCK TM))
            (AND (EQUAL (TM-IPOINTS (TM-SET-RWSTATE RWSTATE TM))
              (TM-IPOINTS TM))
              (AND (EQUAL (TM-OPOINTS (TM-SET-RWSTATE RWSTATE TM))
                (TM-OPOINTS TM))
                (AND (EQUAL (TM-PC (TM-SET-RWSTATE RWSTATE TM))
                  (TM-PC TM))
                  (EQUAL (TM-SP (TM-SET-RWSTATE RWSTATE TM))
                    (TM-SP TM))))))))))))))
  ((ENABLE TM-SET-RWSTATE TM-PC TM-SP)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-rwstate.

THEOREM: good-tm-tm-set-clock
 $(\text{good-tm}(tm) \wedge \text{finite-numberp}(clock, '65536))$
 $\rightarrow \text{good-tm}(\text{tm-set-clock}(clock, tm))$

EVENT: Disable good-tm-tm-set-clock.

```

(PROVE-LEMMA
  ACCESS-TM-SET-CLOCK
  (REWRITE)
  (AND
    (EQUAL (TM-MEMORY (TM-SET-CLOCK CLOCK TM))

```

```

(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-CLOCK CLOCK TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-CLOCK CLOCK TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-CLOCK CLOCK TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-CLOCK CLOCK TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-CLOCK CLOCK TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-CLOCK CLOCK TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-CLOCK CLOCK TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-CLOCK CLOCK TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-SET-CLOCK CLOCK TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-CLOCK CLOCK TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-CLOCK CLOCK TM))
CLOCK)
(AND (EQUAL (TM-IPOINTS (TM-SET-CLOCK CLOCK TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPORTS (TM-SET-CLOCK CLOCK TM))
(TM-OPORTS TM))
(AND (EQUAL (TM-PC (TM-SET-CLOCK CLOCK TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-SET-CLOCK CLOCK TM))
(TM-SP TM))))))))))))))
((ENABLE TM-SET-CLOCK TM-PC TM-SP)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-clock.

THEOREM: good-tm-tm-set-ports
 (good-tm (*tm*)
 ^ (plistp (*iports*)
 ^ ((length (*iports*) = '16) ^ good-tm-iport-array (*iports*)))
 → good-tm (tm-set-ports (*iports*, *tm*))

EVENT: Disable good-tm-tm-set-ports.

```
(PROVE-LEMMA
ACCESS-TM-SET-IPORTS
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-SET-IPORTS IPORTS TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-SET-IPORTS IPORTS TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-SET-IPORTS IPORTS TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-SET-IPORTS IPORTS TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-SET-IPORTS IPORTS TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-SET-IPORTS IPORTS TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-SET-IPORTS IPORTS TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-SET-IPORTS IPORTS TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-SET-IPORTS IPORTS TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-SET-IPORTS IPORTS TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-SET-IPORTS IPORTS TM))
```

```

(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-SET-IPORTS IPORTS TM))
            (TM-CLOCK TM))
      (AND (EQUAL (TM-IPORTS (TM-SET-IPORTS IPORTS TM))
                  IPORTS)
            (AND (EQUAL (TM-OPORTS (TM-SET-IPORTS IPORTS TM))
                        (TM-OPORTS TM))
                  (AND (EQUAL (TM-PC (TM-SET-IPORTS IPORTS TM))
                              (TM-PC TM))
                       (EQUAL (TM-SP (TM-SET-IPORTS IPORTS TM))
                              (TM-SP TM))))))))))
((ENABLE TM-SET-IPORTS TM-PC TM-SP)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-iptables.

THEOREM: good-tm-tm-set-oports

```

(good-tm (tm)
  ^ (plstp (oports)
    ^ ((length (oports) = '16) ^ good-tm-oport-array (oports)))
→ good-tm (tm-set-oports (oports, tm))

```

EVENT: Disable good-tm-tm-set-oports.

```

(PROVE-LEMMA
 ACCESS-TM-SET-OPOINTS
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-SET-OPOINTS OPOINTS TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-SET-OPOINTS OPOINTS TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-SET-OPOINTS OPOINTS TM))
           (TM-CC TM))
    (AND
     (EQUAL (TM-ERROR (TM-SET-OPOINTS OPOINTS TM))
            (TM-ERROR TM))
     (AND

```

```

(EQUAL (TM-BASE (TM-SET-OPOINTS OPOINTS TM))
      (TM-BASE TM))
(AND
  (EQUAL (TM-LIMIT (TM-SET-OPOINTS OPOINTS TM))
        (TM-LIMIT TM))
  (AND
    (EQUAL (TM-SLIMIT (TM-SET-OPOINTS OPOINTS TM))
          (TM-SLIMIT TM))
    (AND
      (EQUAL (TM-SVMODE (TM-SET-OPOINTS OPOINTS TM))
            (TM-SVMODE TM))
      (AND
        (EQUAL (TM-SVCFLAG (TM-SET-OPOINTS OPOINTS TM))
              (TM-SVCFLAG TM))
        (AND
          (EQUAL (TM-SVCID (TM-SET-OPOINTS OPOINTS TM))
                (TM-SVCID TM))
          (AND (EQUAL (TM-RWSTATE (TM-SET-OPOINTS OPOINTS TM))
                    (TM-RWSTATE TM))
              (AND (EQUAL (TM-CLOCK (TM-SET-OPOINTS OPOINTS TM))
                        (TM-CLOCK TM))
                  (AND (EQUAL (TM-IPOINTS (TM-SET-OPOINTS OPOINTS TM))
                            (TM-IPOINTS TM))
                      (AND (EQUAL (TM-OPOINTS (TM-SET-OPOINTS OPOINTS TM))
                                OPOINTS)
                          (AND (EQUAL (TM-PC (TM-SET-OPOINTS OPOINTS TM))
                                    (TM-PC TM))
                              (EQUAL (TM-SP (TM-SET-OPOINTS OPOINTS TM))
                                    (TM-SP TM)))))))))))))))))
((ENABLE TM-SET-OPOINTS TM-PC TM-SP)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable access-tm-set-oports.

THEOREM: good-tm-tm-set-pc

$(\text{good-tm}(tm) \wedge \text{finite-numberp}(pc, '65536)) \rightarrow \text{good-tm}(\text{tm-set-pc}(pc, tm))$

EVENT: Disable good-tm-tm-set-pc.

(PROVE-LEMMA


```

ACCESS-TM-SET-PC
(REWRITE)
(AND
  (EQUAL (TM-MEMORY (TM-SET-PC PC TM))
    (TM-MEMORY TM))
  (AND
    (EQUAL (TM-CC (TM-SET-PC PC TM))
      (TM-CC TM))
    (AND
      (EQUAL (TM-ERROR (TM-SET-PC PC TM))
        (TM-ERROR TM))
      (AND
        (EQUAL (TM-BASE (TM-SET-PC PC TM))
          (TM-BASE TM))
        (AND
          (EQUAL (TM-LIMIT (TM-SET-PC PC TM))
            (TM-LIMIT TM))
          (AND
            (EQUAL (TM-SLIMIT (TM-SET-PC PC TM))
              (TM-SLIMIT TM))
            (AND (EQUAL (TM-SVMODE (TM-SET-PC PC TM))
              (TM-SVMODE TM))
              (AND (EQUAL (TM-SVCFLAG (TM-SET-PC PC TM))
                (TM-SVCFLAG TM))
                (AND (EQUAL (TM-SVCID (TM-SET-PC PC TM))
                  (TM-SVCID TM))
                  (AND (EQUAL (TM-RWSTATE (TM-SET-PC PC TM))
                    (TM-RWSTATE TM))
                    (AND (EQUAL (TM-CLOCK (TM-SET-PC PC TM))
                      (TM-CLOCK TM))
                      (AND (EQUAL (TM-IPOINTS (TM-SET-PC PC TM))
                        (TM-IPOINTS TM))
                        (AND (EQUAL (TM-OPOINTS (TM-SET-PC PC TM))
                          (TM-OPOINTS TM))
                          (EQUAL (TM-SP (TM-SET-PC PC TM))
                            (TM-SP TM))))))))))))))))))
  ((ENABLE TM-SET-PC TM-PC TM-SP TM-SET-REGS
    A-FINITE-NUMBER-IS-LESSP-THAN-ITS-LUB GETNTH-PUTNTH-NON-INTERFERENCE)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T)))

```

EVENT: Disable access-tm-set-pc.

THEOREM: good-tm-tm-set-sp

(good-tm (*tm*) ∧ finite-numberp (*sp*, '65536)) → good-tm (tm-set-sp (*sp*, *tm*))
 EVENT: Disable good-tm-tm-set-sp.

```
(PROVE-LEMMA
ACCESS-TM-SET-SP
(REWRITE)
(AND
  (EQUAL (TM-MEMORY (TM-SET-SP SP TM))
    (TM-MEMORY TM))
  (AND
    (EQUAL (TM-CC (TM-SET-SP SP TM))
      (TM-CC TM))
    (AND
      (EQUAL (TM-ERROR (TM-SET-SP SP TM))
        (TM-ERROR TM))
      (AND
        (EQUAL (TM-BASE (TM-SET-SP SP TM))
          (TM-BASE TM))
        (AND
          (EQUAL (TM-LIMIT (TM-SET-SP SP TM))
            (TM-LIMIT TM))
          (AND
            (EQUAL (TM-SLIMIT (TM-SET-SP SP TM))
              (TM-SLIMIT TM))
            (AND (EQUAL (TM-SVMODE (TM-SET-SP SP TM))
              (TM-SVMODE TM))
              (AND (EQUAL (TM-SVCFLAG (TM-SET-SP SP TM))
                (TM-SVCFLAG TM))
                (AND (EQUAL (TM-SVCID (TM-SET-SP SP TM))
                  (TM-SVCID TM))
                  (AND (EQUAL (TM-RWSTATE (TM-SET-SP SP TM))
                    (TM-RWSTATE TM))
                    (AND (EQUAL (TM-CLOCK (TM-SET-SP SP TM))
                      (TM-CLOCK TM))
                      (AND (EQUAL (TM-IPOINTS (TM-SET-SP SP TM))
                        (TM-IPOINTS TM))
                        (AND (EQUAL (TM-OPOINTS (TM-SET-SP SP TM))
                          (TM-OPOINTS TM))
                          (EQUAL (TM-PC (TM-SET-SP SP TM))
                            (TM-PC TM))))))))))))))))))
  ((ENABLE TM-SET-SP TM-PC TM-SP TM-SET-REGS
```

A-FINITE-NUMBER-IS-LESSP-THAN-ITS-LUB GETNTH-PUTNTH-NON-INTERFERENCE)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

EVENT: Disable access-tm-set-sp.

THEOREM: good-tm-tm-execute-add
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-add}(addr1, addr2, tm))$

EVENT: Disable good-tm-tm-execute-add.

THEOREM: good-tm-tm-execute-branch
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-branch}(addr, tm))$

EVENT: Disable good-tm-tm-execute-branch.

THEOREM: good-tm-tm-execute-branch-on-zero
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-branch-on-zero}(addr, tm))$

EVENT: Disable good-tm-tm-execute-branch-on-zero.

THEOREM: good-tm-tm-execute-branch-not-zero
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-branch-not-zero}(addr, tm))$

EVENT: Disable good-tm-tm-execute-branch-not-zero.

THEOREM: good-tm-tm-execute-call
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-call}(addr, tm))$

EVENT: Disable good-tm-tm-execute-call.

THEOREM: good-tm-tm-execute-compare
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-compare}(addr1, addr2, tm))$

EVENT: Disable good-tm-tm-execute-compare.

THEOREM: good-tm-tm-execute-decr
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-decr}(addr, tm))$

EVENT: Disable good-tm-tm-execute-decr.

THEOREM: good-tm-tm-execute-decr-mod
(good-tm(tm) \wedge (tm-good-address($addr1$, tm) \wedge tm-good-address($addr2$, tm)))
 \rightarrow good-tm(tm-execute-decr-mod($addr1$, $addr2$, tm))

EVENT: Disable good-tm-tm-execute-decr-mod.

THEOREM: good-tm-tm-execute-incr
good-tm(tm) \rightarrow good-tm(tm-execute-incr($addr$, tm))

EVENT: Disable good-tm-tm-execute-incr.

THEOREM: good-tm-tm-execute-incr-mod
(good-tm(tm) \wedge (tm-good-address($addr1$, tm) \wedge tm-good-address($addr2$, tm)))
 \rightarrow good-tm(tm-execute-incr-mod($addr1$, $addr2$, tm))

EVENT: Disable good-tm-tm-execute-incr-mod.

THEOREM: good-tm-tm-execute-load-base
(good-tm(tm) \wedge tm-good-address($addr$, tm))
 \rightarrow good-tm(tm-execute-load-base($addr$, tm))

EVENT: Disable good-tm-tm-execute-load-base.

THEOREM: good-tm-tm-execute-load-limit
(good-tm(tm) \wedge tm-good-address($addr$, tm))
 \rightarrow good-tm(tm-execute-load-limit($addr$, tm))

EVENT: Disable good-tm-tm-execute-load-limit.

DEFINITION:

tm-good-supervisor-address($addr$)
= **if** real-addr-source($addr$) \simeq 0
 then finite-numberp(real-addr-num($addr$), '65536)
 elseif real-addr-source($addr$) = '1
 then finite-numberp(real-addr-num($addr$), '8)
 else finite-numberp(real-addr-num($addr$), '65536) **endif**

THEOREM: finite-numberp-tm-fetch-in-supervisor-mode
(good-tm(tm)
 \wedge (tm-in-supervisor-mode(tm) \wedge tm-good-supervisor-address($addr$)))
 \rightarrow finite-numberp(tm-fetch($addr$, tm), '65536)

EVENT: Disable finite-numberp-tm-fetch-in-supervisor-mode.

THEOREM: tm-good-supervisor-address-tm-incr-address
tm-good-supervisor-address (tm-incr-address (*addr*))

EVENT: Disable tm-good-supervisor-address-tm-incr-address.

THEOREM: tm-good-supervisor-address-tm-incrn-address
tm-good-supervisor-address (*addr*)
→ tm-good-supervisor-address (tm-incrn-address (*n*, *addr*))

EVENT: Disable tm-good-supervisor-address-tm-incrn-address.

THEOREM: tm-good-supervisor-address-tm-good-address
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ tm-good-supervisor-address (*addr*)

EVENT: Disable tm-good-supervisor-address-tm-good-address.

THEOREM: good-tm-tm-execute-load-psw
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-load-psw (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-load-psw.

THEOREM: good-tm-tm-execute-mod
good-tm (*tm*) → good-tm (tm-execute-mod (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-mod.

THEOREM: good-tm-tm-execute-move
(good-tm (*tm*) ∧ tm-good-address (*addr2*, *tm*))
→ good-tm (tm-execute-move (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-move.

THEOREM: good-tm-tm-execute-multiply
good-tm (*tm*) → good-tm (tm-execute-multiply (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-multiply.

THEOREM: finite-numberp-add1-tm-sp
(good-tm (tm) \wedge (\neg tm-stack-underflowp (tm)))
 \rightarrow finite-numberp (1 + tm-sp (tm), '65536)

EVENT: Disable finite-numberp-add1-tm-sp.

THEOREM: tm-incr-sp-is-add1-sp
(good-tm (tm) \wedge (\neg tm-stack-underflowp (tm)))
 \rightarrow (tm-incr (tm-sp (tm)) = (1 + tm-sp (tm)))

EVENT: Disable tm-incr-sp-is-add1-sp.

THEOREM: good-tm-tm-execute-return
good-tm (tm) \rightarrow good-tm (tm-execute-return (tm))

EVENT: Disable good-tm-tm-execute-return.

THEOREM: good-tm-tm-execute-test-iport
good-tm (tm) \rightarrow good-tm (tm-execute-test-iport ($addr$, tm))

EVENT: Disable good-tm-tm-execute-test-iport.

THEOREM: good-tm-tm-execute-test-oport
good-tm (tm) \rightarrow good-tm (tm-execute-test-oport ($addr$, tm))

EVENT: Disable good-tm-tm-execute-test-oport.

THEOREM: good-tm-tm-execute-set-clock
(good-tm (tm) \wedge tm-good-address ($addr$, tm))
 \rightarrow good-tm (tm-execute-set-clock ($addr$, tm))

EVENT: Disable good-tm-tm-execute-set-clock.

THEOREM: good-tm-tm-execute-run
good-tm (tm) \rightarrow good-tm (tm-execute-run (tm))

EVENT: Disable good-tm-tm-execute-run.

THEOREM: good-tm-tm-execute-wait
good-tm (tm) \rightarrow good-tm (tm-execute-wait (tm))

EVENT: Disable good-tm-tm-execute-wait.

THEOREM: good-tm-oports-tm-post-output-interrupt
 $(\text{plistp}(l) \wedge \text{good-tm-oport-array}(l))$
 $\rightarrow (\text{plistp}(\text{tm-post-output-interrupt}(id, l))$
 $\quad \wedge ((\text{length}(\text{tm-post-output-interrupt}(id, l)) = \text{length}(l))$
 $\quad \wedge \text{good-tm-oport-array}(\text{tm-post-output-interrupt}(id, l))))$

EVENT: Disable good-tm-oports-tm-post-output-interrupt.

THEOREM: good-tm-tm-execute-post-output-interrupt
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-post-output-interrupt}(addr, tm))$

EVENT: Disable good-tm-tm-execute-post-output-interrupt.

THEOREM: good-tm-oports-tm-start-output
 $(\text{plistp}(l) \wedge \text{good-tm-oport-array}(l))$
 $\rightarrow (\text{plistp}(\text{tm-start-output}(char, id, l))$
 $\quad \wedge ((\text{length}(\text{tm-start-output}(char, id, l)) = \text{length}(l))$
 $\quad \wedge \text{good-tm-oport-array}(\text{tm-start-output}(char, id, l))))$

EVENT: Disable good-tm-oports-tm-start-output.

THEOREM: good-tm-tm-execute-start-output
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-start-output}(addr1, addr2, tm))$

EVENT: Disable good-tm-tm-execute-start-output.

THEOREM: good-tm-tm-execute-svc
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-svc}(addr, tm))$

EVENT: Disable good-tm-tm-execute-svc.

THEOREM: good-tm-tm-execute-svc-return
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(addr, tm))$
 $\rightarrow \text{good-tm}(\text{tm-execute-svc-return}(addr, tm))$

EVENT: Disable good-tm-tm-execute-svc-return.

THEOREM: good-tm-tm-execute-nullary
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-nullary}(opcode, tm))$

EVENT: Disable good-tm-tm-execute-nullary.

THEOREM: good-tm-tm-execute-unary
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-unary}(opcode, addr, tm))$

EVENT: Disable good-tm-tm-execute-unary.

THEOREM: good-tm-tm-execute-binary
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-binary}(opcode, addr1, addr2, tm))$

EVENT: Disable good-tm-tm-execute-binary.

THEOREM: good-tm-tm-execute-instruction
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-instruction}(instruction, tm))$

EVENT: Disable good-tm-tm-execute-instruction.

THEOREM: plus-tm-base-tm-limit-tm-set-pc
 $(\text{tm-base}(\text{tm-set-pc}(pc, tm)) + \text{tm-limit}(\text{tm-set-pc}(pc, tm)))$
 $= (\text{tm-base}(tm) + \text{tm-limit}(tm))$

EVENT: Disable plus-tm-base-tm-limit-tm-set-pc.

THEOREM: finite-numberp-tm-good-address
 $(\text{good-tm}(tm) \wedge \text{tm-good-address}(\text{real-addr}(2, n), tm))$
 $\rightarrow \text{finite-numberp}(n, 65536)$

EVENT: Disable finite-numberp-tm-good-address.

THEOREM: good-tm-tm-execute
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute}(instruction, tm))$

EVENT: Disable good-tm-tm-execute.

THEOREM: good-tm-tm-decrement-clock
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-decrement-clock}(tm))$

EVENT: Disable good-tm-tm-decrement-clock.

THEOREM: plus-tm-base-tm-limit-tm-decrement-clock
 $(\text{tm-base}(\text{tm-decrement-clock}(tm)) + \text{tm-limit}(\text{tm-decrement-clock}(tm)))$
 $= (\text{tm-base}(tm) + \text{tm-limit}(tm))$

EVENT: Disable plus-tm-base-tm-limit-tm-decrement-clock.


```

(PROVE-LEMMA
TM-DECREMENT-CLOCK-PRESERVES-ALL-BUT-CLOCK
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-DECREMENT-CLOCK TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-DECREMENT-CLOCK TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-DECREMENT-CLOCK TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-DECREMENT-CLOCK TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-DECREMENT-CLOCK TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-DECREMENT-CLOCK TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-DECREMENT-CLOCK TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-DECREMENT-CLOCK TM))
(TM-SVMODE TM))
(AND (EQUAL (TM-SVCFLAG (TM-DECREMENT-CLOCK TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-DECREMENT-CLOCK TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-DECREMENT-CLOCK TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-IPOINTS (TM-DECREMENT-CLOCK TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPORTS (TM-DECREMENT-CLOCK TM))
(TM-OPORTS TM))
(AND (EQUAL (TM-PC (TM-DECREMENT-CLOCK TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-DECREMENT-CLOCK TM))
(TM-SP TM))))))))))))))
((ENABLE TM-DECREMENT-CLOCK ACCESS-TM-SET-CLOCK)

```

(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

EVENT: Disable tm-decrement-clock-preserves-all-but-clock.

THEOREM: tm-good-address-tm-decrement-clock
 $\text{tm-good-address}(addr, \text{tm-decrement-clock}(tm)) = \text{tm-good-address}(addr, tm)$

EVENT: Disable tm-good-address-tm-decrement-clock.

THEOREM: good-tm-tm-fetch-execute
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-fetch-execute}(tm))$

EVENT: Disable good-tm-tm-fetch-execute.

THEOREM: tm-good-address-tm-store
 $\text{tm-good-address}(addr, \text{tm-store}(value, addr, tm)) = \text{tm-good-address}(addr, tm)$

EVENT: Disable tm-good-address-tm-store.

THEOREM: good-tm-tm-execute-clock-interrupt
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-clock-interrupt}(tm))$

EVENT: Disable good-tm-tm-execute-clock-interrupt.

THEOREM: good-tm-tm-execute-error-interrupt
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-error-interrupt}(tm))$

EVENT: Disable good-tm-tm-execute-error-interrupt.

THEOREM: good-tm-tm-execute-svc-interrupt
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-execute-svc-interrupt}(tm))$

EVENT: Disable good-tm-tm-execute-svc-interrupt.

THEOREM: good-tm-tm-execute-input-interrupt
 $(\text{good-tm}(tm) \wedge \text{tm-some-input-interruptp}(\text{tm-ports}(tm)))$
 $\rightarrow \text{good-tm}(\text{tm-execute-input-interrupt}(tm))$

EVENT: Disable good-tm-tm-execute-input-interrupt.

THEOREM: good-tm-tm-execute-output-interrupt
 $(\text{good-tm}(tm) \wedge \text{tm-some-output-interruptp}(\text{tm-ports}(tm)))$
 $\rightarrow \text{good-tm}(\text{tm-execute-output-interrupt}(tm))$

EVENT: Disable good-tm-tm-execute-output-interrupt.

THEOREM: good-tm-tm-step
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-step}(tm))$

EVENT: Disable good-tm-tm-step.

THEOREM: good-tm-ports-tm-post-input-interrupt
 $(\text{plistp}(l)$
 $\wedge (\text{finite-numberp}(id, \text{length}(l))$
 $\wedge (\text{finite-numberp}(char, '256) \wedge \text{good-tm-iport-array}(l)))$
 $\rightarrow (\text{plistp}(\text{tm-post-input-interrupt}(char, id, l))$
 $\wedge ((\text{length}(\text{tm-post-input-interrupt}(char, id, l)) = \text{length}(l))$
 $\wedge \text{good-tm-iport-array}(\text{tm-post-input-interrupt}(char, id, l))))$

EVENT: Disable good-tm-ports-tm-post-input-interrupt.

THEOREM: good-tm-tm-post-interrupt
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-post-interrupt}(event, tm))$

EVENT: Disable good-tm-tm-post-interrupt.

THEOREM: good-tm-tm-processor
 $\text{good-tm}(tm) \rightarrow \text{good-tm}(\text{tm-processor}(tm, n))$

EVENT: Disable good-tm-tm-processor.

THEOREM: registers-untouched-by-tm-execute-add
 $(\text{tm-error}(\text{tm-execute-add}(addr1, addr2, tm)) = \text{tm-error}(tm))$
 $\wedge ((\text{tm-base}(\text{tm-execute-add}(addr1, addr2, tm)) = \text{tm-base}(tm))$
 $\wedge ((\text{tm-limit}(\text{tm-execute-add}(addr1, addr2, tm)) = \text{tm-limit}(tm))$
 $\wedge ((\text{tm-slimit}(\text{tm-execute-add}(addr1, addr2, tm))$
 $= \text{tm-slimit}(tm))$
 $\wedge ((\text{tm-svmode}(\text{tm-execute-add}(addr1, addr2, tm))$
 $= \text{tm-svmode}(tm))$
 $\wedge ((\text{tm-svcflag}(\text{tm-execute-add}(addr1,$
 $addr2,$
 $tm)))$

$$\begin{aligned}
&= \text{tm-svcflag}(tm) \\
\wedge & ((\text{tm-svcid}(\text{tm-execute-add}(addr1, \\
&\quad\quad\quad addr2, \\
&\quad\quad\quad tm))) \\
&= \text{tm-svcid}(tm) \\
\wedge & ((\text{tm-rwstate}(\text{tm-execute-add}(addr1, \\
&\quad\quad\quad addr2, \\
&\quad\quad\quad tm))) \\
&= \text{tm-rwstate}(tm) \\
\wedge & ((\text{tm-clock}(\text{tm-execute-add}(addr1, \\
&\quad\quad\quad addr2, \\
&\quad\quad\quad tm))) \\
&= \text{tm-clock}(tm) \\
\wedge & ((\text{tm-iptables}(\text{tm-execute-add}(addr1, \\
&\quad\quad\quad addr2, \\
&\quad\quad\quad tm))) \\
&= \text{tm-iptables}(tm) \\
\wedge & (\text{tm-oports}(\text{tm-execute-add}(addr1, \\
&\quad\quad\quad addr2, \\
&\quad\quad\quad tm))) \\
&= \text{tm-oports}(tm)))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-add.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-BRANCH
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-BRANCH ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-BRANCH ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-EXECUTE-BRANCH ADDR TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-BRANCH ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-BRANCH ADDR TM))
(TM-LIMIT TM))

```

```

(AND
  (EQUAL (TM-SLIMIT (TM-EXECUTE-BRANCH ADDR TM))
    (TM-SLIMIT TM))
  (AND
    (EQUAL (TM-SVMODE (TM-EXECUTE-BRANCH ADDR TM))
      (TM-SVMODE TM))
    (AND
      (EQUAL (TM-SVCFLAG (TM-EXECUTE-BRANCH ADDR TM))
        (TM-SVCFLAG TM))
      (AND
        (EQUAL (TM-SVCID (TM-EXECUTE-BRANCH ADDR TM))
          (TM-SVCID TM))
        (AND (EQUAL (TM-RWSTATE (TM-EXECUTE-BRANCH ADDR TM))
          (TM-RWSTATE TM))
          (AND (EQUAL (TM-CLOCK (TM-EXECUTE-BRANCH ADDR TM))
            (TM-CLOCK TM))
            (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-BRANCH ADDR TM))
              (TM-IPOINTS TM))
              (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-BRANCH ADDR TM))
                (TM-OPOINTS TM))
                (EQUAL (TM-SP (TM-EXECUTE-BRANCH ADDR TM))
                  (TM-SP TM))))))))))))))
  ((ENABLE TM-EXECUTE-BRANCH ACCESS-TM-SET-PC)
  (ENABLE-THEORY GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-branch.

```

(PROVE-LEMMA
  REGISTERS-UNTOUCHED-BY-TM-EXECUTE-BRANCH-ON-ZERO
  (REWRITE)
  (AND
    (EQUAL (TM-MEMORY (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
      (TM-MEMORY TM))
    (AND
      (EQUAL (TM-CC (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
        (TM-CC TM))
      (AND
        (EQUAL (TM-ERROR (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
          (TM-ERROR TM))
        (AND
          (EQUAL (TM-BASE (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
            (TM-BASE TM))

```

```

(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-RWSTATE TM))
(AND
(EQUAL (TM-CLOCK (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-OPOINTS TM))
(EQUAL (TM-SP (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
(TM-SP TM)))))))))))))
((ENABLE TM-EXECUTE-BRANCH-ON-ZERO ACCESS-TM-SET-PC)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-branch-on-zero.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-BRANCH-NOT-ZERO
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))

```

```

(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-RWSTATE TM))
(AND
(EQUAL (TM-CLOCK (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-OPOINTS TM))
(EQUAL (TM-SP (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
(TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-BRANCH-NOT-ZERO ACCESS-TM-SET-PC)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-branch-not-zero.

THEOREM: registers-untouched-by-tm-execute-call
 $(\text{tm-cc}(\text{tm-execute-call}(addr, tm)) = \text{tm-cc}(tm))$
 $\wedge ((\text{tm-base}(\text{tm-execute-call}(addr, tm)) = \text{tm-base}(tm))$
 $\wedge ((\text{tm-limit}(\text{tm-execute-call}(addr, tm)) = \text{tm-limit}(tm))$


```

(AND
  (EQUAL (TM-SLIMIT (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
    (TM-SLIMIT TM))
  (AND
    (EQUAL (TM-SVMODE (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
      (TM-SVMODE TM))
    (AND
      (EQUAL (TM-SVCFLAG (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
        (TM-SVCFLAG TM))
      (AND
        (EQUAL (TM-SVCID (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
          (TM-SVCID TM))
        (AND
          (EQUAL (TM-RWSTATE (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
            (TM-RWSTATE TM))
          (AND
            (EQUAL (TM-CLOCK (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
              (TM-CLOCK TM))
            (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
              (TM-IPOINTS TM))
              (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
                (TM-OPOINTS TM))
                (AND (EQUAL (TM-PC (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
                  (TM-PC TM))
                  (EQUAL (TM-SP (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
                    (TM-SP TM))))))))))))))
  ((ENABLE TM-EXECUTE-COMPARE ACCESS-TM-SET-CC)
  (ENABLE-THEORY GROUND-ZERO)
  (DISABLE-THEORY T)))

```

EVENT: Disable registers-untouched-by-tm-execute-compare.

THEOREM: registers-untouched-by-tm-execute-decr

$$\begin{aligned}
& (\text{tm-error} (\text{tm-execute-decr} (\text{addr}, \text{tm})) = \text{tm-error} (\text{tm})) \\
\wedge & ((\text{tm-base} (\text{tm-execute-decr} (\text{addr}, \text{tm})) = \text{tm-base} (\text{tm})) \\
& \wedge ((\text{tm-limit} (\text{tm-execute-decr} (\text{addr}, \text{tm})) = \text{tm-limit} (\text{tm})) \\
& \quad \wedge ((\text{tm-slimit} (\text{tm-execute-decr} (\text{addr}, \text{tm})) \\
& \quad \quad = \text{tm-slimit} (\text{tm})) \\
& \quad \wedge ((\text{tm-svmode} (\text{tm-execute-decr} (\text{addr}, \text{tm})) \\
& \quad \quad = \text{tm-svmode} (\text{tm})) \\
& \quad \wedge ((\text{tm-svcflag} (\text{tm-execute-decr} (\text{addr}, \text{tm})) \\
& \quad \quad = \text{tm-svcflag} (\text{tm})) \\
& \quad \wedge ((\text{tm-svcid} (\text{tm-execute-decr} (\text{addr}, \\
& \quad \quad \quad \text{tm}))
\end{aligned}$$

$$\begin{aligned}
&= \text{tm-svcid}(tm) \\
\wedge & ((\text{tm-rwstate}(\text{tm-execute-decr}(addr, \\
& \hspace{10em} tm))) \\
&= \text{tm-rwstate}(tm) \\
\wedge & ((\text{tm-clock}(\text{tm-execute-decr}(addr, \\
& \hspace{10em} tm))) \\
&= \text{tm-clock}(tm) \\
\wedge & ((\text{tm-iports}(\text{tm-execute-decr}(addr, \\
& \hspace{10em} tm))) \\
&= \text{tm-iports}(tm) \\
\wedge & (\text{tm-oports}(\text{tm-execute-decr}(addr, \\
& \hspace{10em} tm))) \\
&= \text{tm-oports}(tm)))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-decr.

THEOREM: registers-untouched-by-tm-execute-decr-mod

$$\begin{aligned}
&(\text{tm-error}(\text{tm-execute-decr-mod}(addr1, addr2, tm)) = \text{tm-error}(tm)) \\
\wedge & ((\text{tm-base}(\text{tm-execute-decr-mod}(addr1, addr2, tm)) = \text{tm-base}(tm)) \\
& \wedge ((\text{tm-limit}(\text{tm-execute-decr-mod}(addr1, addr2, tm)) \\
& \hspace{2em} = \text{tm-limit}(tm)) \\
& \wedge ((\text{tm-slimit}(\text{tm-execute-decr-mod}(addr1, addr2, tm)) \\
& \hspace{2em} = \text{tm-slimit}(tm)) \\
& \wedge ((\text{tm-svmode}(\text{tm-execute-decr-mod}(addr1, \\
& \hspace{10em} addr2, \\
& \hspace{10em} tm))) \\
& \hspace{2em} = \text{tm-svmode}(tm)) \\
\wedge & ((\text{tm-svcflag}(\text{tm-execute-decr-mod}(addr1, \\
& \hspace{10em} addr2, \\
& \hspace{10em} tm))) \\
& \hspace{2em} = \text{tm-svcflag}(tm)) \\
\wedge & ((\text{tm-svcid}(\text{tm-execute-decr-mod}(addr1, \\
& \hspace{10em} addr2, \\
& \hspace{10em} tm))) \\
& \hspace{2em} = \text{tm-svcid}(tm)) \\
\wedge & ((\text{tm-rwstate}(\text{tm-execute-decr-mod}(addr1, \\
& \hspace{10em} addr2, \\
& \hspace{10em} tm))) \\
& \hspace{2em} = \text{tm-rwstate}(tm)) \\
\wedge & ((\text{tm-clock}(\text{tm-execute-decr-mod}(addr1, \\
& \hspace{10em} addr2, \\
& \hspace{10em} tm))) \\
& \hspace{2em} = \text{tm-clock}(tm)) \\
\wedge & ((\text{tm-iports}(\text{tm-execute-decr-mod}(addr1,
\end{aligned}$$

$$\begin{aligned}
& \text{addr2,} \\
& \text{tm))} \\
& = \text{tm-iports}(tm) \\
& \wedge (\text{tm-oports}(\text{tm-execute-decr-mod}(\text{addr1}, \\
& \text{addr2}, \\
& \text{tm})) \\
& = \text{tm-oports}(tm)))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-decr-mod.

THEOREM: registers-untouched-by-tm-execute-incr

$$\begin{aligned}
& (\text{tm-error}(\text{tm-execute-incr}(\text{addr}, \text{tm})) = \text{tm-error}(tm)) \\
& \wedge ((\text{tm-base}(\text{tm-execute-incr}(\text{addr}, \text{tm})) = \text{tm-base}(tm)) \\
& \quad \wedge ((\text{tm-limit}(\text{tm-execute-incr}(\text{addr}, \text{tm})) = \text{tm-limit}(tm)) \\
& \quad \quad \wedge ((\text{tm-slimit}(\text{tm-execute-incr}(\text{addr}, \text{tm})) \\
& \quad \quad \quad = \text{tm-slimit}(tm)) \\
& \quad \quad \wedge ((\text{tm-svmode}(\text{tm-execute-incr}(\text{addr}, \text{tm})) \\
& \quad \quad \quad = \text{tm-svmode}(tm)) \\
& \quad \quad \wedge ((\text{tm-svcflag}(\text{tm-execute-incr}(\text{addr}, \text{tm})) \\
& \quad \quad \quad = \text{tm-svcflag}(tm)) \\
& \quad \quad \wedge ((\text{tm-svcid}(\text{tm-execute-incr}(\text{addr}, \\
& \quad \quad \quad \text{tm})) \\
& \quad \quad \quad = \text{tm-svcid}(tm)) \\
& \quad \quad \wedge ((\text{tm-rwstate}(\text{tm-execute-incr}(\text{addr}, \\
& \quad \quad \quad \text{tm})) \\
& \quad \quad \quad = \text{tm-rwstate}(tm)) \\
& \quad \quad \wedge ((\text{tm-clock}(\text{tm-execute-incr}(\text{addr}, \\
& \quad \quad \quad \text{tm})) \\
& \quad \quad \quad = \text{tm-clock}(tm)) \\
& \quad \quad \wedge ((\text{tm-iports}(\text{tm-execute-incr}(\text{addr}, \\
& \quad \quad \quad \text{tm})) \\
& \quad \quad \quad = \text{tm-iports}(tm)) \\
& \quad \quad \wedge (\text{tm-oports}(\text{tm-execute-incr}(\text{addr}, \\
& \quad \quad \quad \text{tm})) \\
& \quad \quad \quad = \text{tm-oports}(tm)))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-incr.

THEOREM: registers-untouched-by-tm-execute-incr-mod

$$\begin{aligned}
& (\text{tm-error}(\text{tm-execute-incr-mod}(\text{addr1}, \text{addr2}, \text{tm})) = \text{tm-error}(tm)) \\
& \wedge ((\text{tm-base}(\text{tm-execute-incr-mod}(\text{addr1}, \text{addr2}, \text{tm})) = \text{tm-base}(tm)) \\
& \quad \wedge ((\text{tm-limit}(\text{tm-execute-incr-mod}(\text{addr1}, \text{addr2}, \text{tm})) \\
& \quad \quad = \text{tm-limit}(tm))
\end{aligned}$$

$$\begin{aligned}
& \wedge ((\text{tm-slimit}(\text{tm-execute-incr-mod}(\text{addr1}, \text{addr2}, \text{tm})) \\
& \quad = \text{tm-slimit}(\text{tm})) \\
& \quad \wedge ((\text{tm-svmode}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-svmode}(\text{tm})) \\
& \quad \wedge ((\text{tm-svcflag}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-svcflag}(\text{tm})) \\
& \quad \wedge ((\text{tm-svcid}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-svcid}(\text{tm})) \\
& \quad \wedge ((\text{tm-rwstate}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-rwstate}(\text{tm})) \\
& \quad \wedge ((\text{tm-clock}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-clock}(\text{tm})) \\
& \quad \wedge ((\text{tm-iports}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-iports}(\text{tm})) \\
& \quad \wedge (\text{tm-oports}(\text{tm-execute-incr-mod}(\text{addr1}, \\
& \quad \quad \quad \text{addr2}, \\
& \quad \quad \quad \text{tm})) \\
& \quad = \text{tm-oports}(\text{tm}))))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-incr-mod.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-LOAD-BASE
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-LOAD-BASE ADDR TM))
(TM-MEMORY TM))
(AND

```

```

(EQUAL (TM-REGS (TM-EXECUTE-LOAD-BASE ADDR TM))
      (TM-REGS TM))
(AND
  (EQUAL (TM-CC (TM-EXECUTE-LOAD-BASE ADDR TM))
        (TM-CC TM))
  (AND
    (EQUAL (TM-BASE (TM-EXECUTE-LOAD-BASE ADDR TM))
          (TM-BASE TM))
    (AND
      (EQUAL (TM-LIMIT (TM-EXECUTE-LOAD-BASE ADDR TM))
            (TM-LIMIT TM))
      (AND
        (EQUAL (TM-SLIMIT (TM-EXECUTE-LOAD-BASE ADDR TM))
              (TM-SLIMIT TM))
        (AND
          (EQUAL (TM-SVMODE (TM-EXECUTE-LOAD-BASE ADDR TM))
                (TM-SVMODE TM))
          (AND
            (EQUAL (TM-SVCFLAG (TM-EXECUTE-LOAD-BASE ADDR TM))
                  (TM-SVCFLAG TM))
            (AND
              (EQUAL (TM-SVCID (TM-EXECUTE-LOAD-BASE ADDR TM))
                    (TM-SVCID TM))
              (AND
                (EQUAL (TM-RWSTATE (TM-EXECUTE-LOAD-BASE ADDR TM))
                      (TM-RWSTATE TM))
                (AND (EQUAL (TM-CLOCK (TM-EXECUTE-LOAD-BASE ADDR TM))
                          (TM-CLOCK TM))
                    (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-LOAD-BASE ADDR TM))
                              (TM-IPOINTS TM))
                        (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-LOAD-BASE ADDR TM))
                                  (TM-OPOINTS TM))
                            (AND (EQUAL (TM-PC (TM-EXECUTE-LOAD-BASE ADDR TM))
                                      (TM-PC TM))
                                (EQUAL (TM-SP (TM-EXECUTE-LOAD-BASE ADDR TM))
                                      (TM-SP TM))))))))))))))))))
      ((ENABLE TM-EXECUTE-LOAD-BASE ACCESS-TM-SET-ERROR)
       (ENABLE-THEORY GROUND-ZERO)
       (DISABLE-THEORY T)))

```

EVENT: Disable registers-untouched-by-tm-execute-load-base.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-LOAD-LIMIT
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPORTS (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-OPORTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-LOAD-LIMIT ADDR TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-LOAD-LIMIT ADDR TM))

```

```

(TM-SP TM)))))))))))))
((ENABLE TM-EXECUTE-LOAD-LIMIT ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-load-limit.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-LOAD-PSW
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-LOAD-PSW ADDR TM))
(TM-RWSTATE TM))

```

```

(AND (EQUAL (TM-CLOCK (TM-EXECUTE-LOAD-PSW ADDR TM))
            (TM-CLOCK TM))
      (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-LOAD-PSW ADDR TM))
                (TM-IPOINTS TM))
            (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-LOAD-PSW ADDR TM))
                    (TM-OPOINTS TM))
                  (AND (EQUAL (TM-PC (TM-EXECUTE-LOAD-PSW ADDR TM))
                          (TM-PC TM))
                      (EQUAL (TM-SP (TM-EXECUTE-LOAD-PSW ADDR TM))
                          (TM-SP TM))))))))))
(ENABLE TM-EXECUTE-LOAD-PSW ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-load-psw.

THEOREM: registers-untouched-by-tm-execute-mod

```

(tm-error (tm-execute-mod (addr1, addr2, tm)) = tm-error (tm))
^ ((tm-base (tm-execute-mod (addr1, addr2, tm)) = tm-base (tm))
   ^ ((tm-limit (tm-execute-mod (addr1, addr2, tm)) = tm-limit (tm))
      ^ ((tm-slimit (tm-execute-mod (addr1, addr2, tm))
          = tm-slimit (tm))
         ^ ((tm-svmode (tm-execute-mod (addr1, addr2, tm))
            = tm-svmode (tm))
            ^ ((tm-svcflag (tm-execute-mod (addr1,
                                           addr2,
                                           tm))
                = tm-svcflag (tm))
               ^ ((tm-svcid (tm-execute-mod (addr1,
                                           addr2,
                                           tm))
                   = tm-svcid (tm))
                  ^ ((tm-rwstate (tm-execute-mod (addr1,
                                                  addr2,
                                                  tm))
                      = tm-rwstate (tm))
                     ^ ((tm-clock (tm-execute-mod (addr1,
                                                    addr2,
                                                    tm))
                         = tm-clock (tm))
                        ^ ((tm-iports (tm-execute-mod (addr1,
                                                       addr2,
                                                       tm))
                            = tm-iports (tm))

```


$$\begin{aligned} & \wedge (\text{tm-oport}(\text{tm-execute-mod}(\text{addr1}, \\ & \qquad \qquad \qquad \text{addr2}, \\ & \qquad \qquad \qquad \text{tm})) \\ & = \text{tm-oport}(\text{tm})) \end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-mod.

```
(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-MOVE
(REWRITE)
(AND
(EQUAL (TM-CC (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-IPOINTS TM))
(EQUAL (TM-OPOINTS (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
(TM-OPOINTS TM)))))))))))))
((ENABLE TM-EXECUTE-MOVE TM-STORE-PRESERVES-ALL-BUT-MEMORY-AND-REGS)
```

(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

EVENT: Disable registers-untouched-by-tm-execute-move.

THEOREM: registers-untouched-by-tm-execute-multiply
(tm-error (tm-execute-multiply (*addr1*, *addr2*, *tm*)) = tm-error (*tm*))
 \wedge ((tm-base (tm-execute-multiply (*addr1*, *addr2*, *tm*)) = tm-base (*tm*))
 \wedge ((tm-limit (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-limit (*tm*))
 \wedge ((tm-slimit (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-slimit (*tm*))
 \wedge ((tm-svmode (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-svmode (*tm*))
 \wedge ((tm-svcflag (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-svcflag (*tm*))
 \wedge ((tm-svcid (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-svcid (*tm*))
 \wedge ((tm-rwstate (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-rwstate (*tm*))
 \wedge ((tm-clock (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-clock (*tm*))
 \wedge ((tm-iports (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-iports (*tm*))
 \wedge (tm-oports (tm-execute-multiply (*addr1*,
addr2,
tm))
= tm-oports (*tm*))))))))))

EVENT: Disable registers-untouched-by-tm-execute-multiply.

THEOREM: registers-untouched-by-tm-execute-return

$$\begin{aligned}
& (\text{tm-memory } (\text{tm-execute-return } (tm)) = \text{tm-memory } (tm)) \\
\wedge & \left((\text{tm-cc } (\text{tm-execute-return } (tm)) = \text{tm-cc } (tm)) \right. \\
& \quad \wedge \left((\text{tm-base } (\text{tm-execute-return } (tm)) = \text{tm-base } (tm)) \right. \\
& \quad \quad \wedge \left((\text{tm-limit } (\text{tm-execute-return } (tm)) = \text{tm-limit } (tm)) \right. \\
& \quad \quad \quad \wedge \left((\text{tm-slimit } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad = \text{tm-slimit } (tm)) \\
& \quad \quad \quad \quad \wedge \left((\text{tm-svmode } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad = \text{tm-svmode } (tm)) \\
& \quad \quad \quad \quad \quad \wedge \left((\text{tm-svcflag } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad = \text{tm-svcflag } (tm)) \\
& \quad \quad \quad \quad \quad \quad \wedge \left((\text{tm-svcid } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad = \text{tm-svcid } (tm)) \\
& \quad \quad \quad \quad \quad \quad \quad \wedge \left((\text{tm-rwstate } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad = \text{tm-rwstate } (tm)) \\
& \quad \quad \quad \quad \quad \quad \quad \quad \wedge \left((\text{tm-clock } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad = \text{tm-clock } (tm)) \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \wedge \left((\text{tm-iports } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad = \text{tm-iports } (tm)) \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \wedge \left((\text{tm-oports } (\text{tm-execute-return } (tm)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad = \text{tm-oports } (tm))))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-return.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-SET-CLOCK
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-SET-CLOCK ADDR TM))

```

```

(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-OPOINTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-SET-CLOCK ADDR TM))
(TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-SET-CLOCK ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-set-clock.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-RUN
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-RUN TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-RUN TM))

```

```

(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-RUN TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-RUN TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-RUN TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-RUN TM))
(TM-SVMODE TM))
(AND (EQUAL (TM-SVCFLAG (TM-EXECUTE-RUN TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-EXECUTE-RUN TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-EXECUTE-RUN TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-RUN TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-RUN TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-RUN TM))
(TM-OPOINTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-RUN TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-RUN TM))
(TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-RUN ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-run.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-WAIT
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-WAIT TM))

```

```

(TM-MEMORY TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-WAIT TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-WAIT TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-WAIT TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-WAIT TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-WAIT TM))
(TM-SVMODE TM))
(AND (EQUAL (TM-SVCFLAG (TM-EXECUTE-WAIT TM))
(TM-SVCFLAG TM))
(AND (EQUAL (TM-SVCID (TM-EXECUTE-WAIT TM))
(TM-SVCID TM))
(AND (EQUAL (TM-RWSTATE (TM-EXECUTE-WAIT TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-WAIT TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-WAIT TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-WAIT TM))
(TM-OPOINTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-WAIT TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-WAIT TM))
(TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-WAIT ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-wait.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-TEST-IPORT
(REWRITE)
(IMPLIES

```

```

(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
  (EQUAL (TM-MEMORY (TM-EXECUTE-TEST-IPOINT ADDR TM))
    (TM-MEMORY TM))
  (AND
    (EQUAL (TM-REGS (TM-EXECUTE-TEST-IPOINT ADDR TM))
      (TM-REGS TM))
    (AND
      (EQUAL (TM-CC (TM-EXECUTE-TEST-IPOINT ADDR TM))
        (TM-CC TM))
      (AND
        (EQUAL (TM-BASE (TM-EXECUTE-TEST-IPOINT ADDR TM))
          (TM-BASE TM))
        (AND
          (EQUAL (TM-LIMIT (TM-EXECUTE-TEST-IPOINT ADDR TM))
            (TM-LIMIT TM))
          (AND
            (EQUAL (TM-SLIMIT (TM-EXECUTE-TEST-IPOINT ADDR TM))
              (TM-SLIMIT TM))
            (AND
              (EQUAL (TM-SVMODE (TM-EXECUTE-TEST-IPOINT ADDR TM))
                (TM-SVMODE TM))
              (AND
                (EQUAL (TM-SVCFLAG (TM-EXECUTE-TEST-IPOINT ADDR TM))
                  (TM-SVCFLAG TM))
                (AND
                  (EQUAL (TM-SVCID (TM-EXECUTE-TEST-IPOINT ADDR TM))
                    (TM-SVCID TM))
                  (AND
                    (EQUAL (TM-RWSTATE (TM-EXECUTE-TEST-IPOINT ADDR TM))
                      (TM-RWSTATE TM))
                    (AND (EQUAL (TM-CLOCK (TM-EXECUTE-TEST-IPOINT ADDR TM))
                      (TM-CLOCK TM))
                      (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-TEST-IPOINT ADDR TM))
                        (TM-IPOINTS TM))
                        (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-TEST-IPOINT ADDR TM))
                          (TM-OPOINTS TM))
                          (AND (EQUAL (TM-PC (TM-EXECUTE-TEST-IPOINT ADDR TM))
                            (TM-PC TM))
                            (EQUAL (TM-SP (TM-EXECUTE-TEST-IPOINT ADDR TM))
                              (TM-SP TM))))))))))))))))))
  ((ENABLE TM-EXECUTE-TEST-IPOINT ACCESS-TM-SET-ERROR)
  (ENABLE-THEORY GROUND-ZERO)
  (DISABLE-THEORY T)))

```

EVENT: Disable registers-untouched-by-tm-execute-test-iport.

```
(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-TEST-OPORT
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-TEST-OPORT ADDR TM))
(TM-IPOINTS TM))
```



```

(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-TEST-OPOINT ADDR TM))
            (TM-OPOINTS TM))
      (AND (EQUAL (TM-PC (TM-EXECUTE-TEST-OPOINT ADDR TM))
                (TM-PC TM))
            (EQUAL (TM-SP (TM-EXECUTE-TEST-OPOINT ADDR TM))
                  (TM-SP TM)))))))))
((ENABLE TM-EXECUTE-TEST-OPOINT ACCESS-TM-SET-ERROR)
 (ENABLE-THEORY GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-test-oport.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-START-OUTPUT
(REWRITE)
(IMPLIES
 (NOT (TM-IN-SUPERVISOR-MODE TM))
 (AND
  (EQUAL (TM-MEMORY (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
        (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
         (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
          (TM-CC TM))
    (AND
     (EQUAL (TM-BASE (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
           (TM-BASE TM))
     (AND
      (EQUAL (TM-LIMIT (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
            (TM-LIMIT TM))
      (AND
       (EQUAL (TM-SLIMIT (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
             (TM-SLIMIT TM))
       (AND
        (EQUAL (TM-SVMODE (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
              (TM-SVMODE TM))
        (AND
         (EQUAL (TM-SVCFLAG (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
               (TM-SVCFLAG TM))
         (AND
          (AND

```

```

(EQUAL (TM-SVCID (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
      (TM-SVCID TM))
(AND
  (EQUAL (TM-RWSTATE (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
        (TM-RWSTATE TM))
  (AND (EQUAL (TM-CLOCK (TM-EXECUTE-START-OUTPUT ADDR1 ADDR2 TM))
              (TM-CLOCK TM))
        (AND (EQUAL (TM-IPOINTS (TM-EXECUTE-START-OUTPUT ADDR1
                                                                    ADDR2 TM))
                    (TM-IPOINTS TM))
              (AND (EQUAL (TM-OPOINTS (TM-EXECUTE-START-OUTPUT ADDR1
                                                                    ADDR2
                                                                    TM))
                    (TM-OPOINTS TM))
                    (AND (EQUAL (TM-PC (TM-EXECUTE-START-OUTPUT ADDR1
                                                                    ADDR2
                                                                    TM))
                                (TM-PC TM))
                          (EQUAL (TM-SP (TM-EXECUTE-START-OUTPUT ADDR1
                                                                    ADDR2
                                                                    TM))
                                (TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-START-OUTPUT ACCESS-TM-SET-ERROR)
 (ENABLE-THEORY GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-start-output.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-POST-OUTPUT-INTERRUPT
(REWRITE)
(IMPLIES
  (NOT (TM-IN-SUPERVISOR-MODE TM))
  (AND
    (EQUAL (TM-MEMORY (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
          (TM-MEMORY TM))
    (AND
      (EQUAL (TM-REGS (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
            (TM-REGS TM))
      (AND
        (EQUAL (TM-CC (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
              (TM-CC TM))

```

```

(AND
  (EQUAL (TM-BASE (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
    (TM-BASE TM))
  (AND
    (EQUAL (TM-LIMIT (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
      (TM-LIMIT TM))
    (AND
      (EQUAL (TM-SLIMIT (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
        (TM-SLIMIT TM))
      (AND
        (EQUAL (TM-SVMODE (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
          (TM-SVMODE TM))
        (AND
          (EQUAL (TM-SVCFLAG (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
            (TM-SVCFLAG TM))
          (AND
            (EQUAL (TM-SVCID (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
              (TM-SVCID TM))
            (AND
              (EQUAL (TM-RWSTATE (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
                (TM-RWSTATE TM))
              (AND
                (EQUAL (TM-CLOCK (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
                  (TM-CLOCK TM))
                (AND
                  (EQUAL (TM-IPOINTS (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
                    (TM-IPOINTS TM))
                  (AND (EQUAL (TM-OPORTS (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR
                    TM))
                    (TM-OPORTS TM))
                    (AND (EQUAL (TM-PC (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR
                    TM))
                    (TM-PC TM))
                    (EQUAL (TM-SP (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR
                    TM))
                    (TM-SP TM))))))))))))))))))
  ((ENABLE TM-EXECUTE-POST-OUTPUT-INTERRUPT ACCESS-TM-SET-ERROR)
  (ENABLE-THEORY GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-post-output-interrupt.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-SVC
(REWRITE)
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-SVC ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-SVC ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-SVC ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-ERROR (TM-EXECUTE-SVC ADDR TM))
(TM-ERROR TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-SVC ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-SVC ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-SVC ADDR TM))
(TM-SLIMIT TM))
(AND (EQUAL (TM-SVMODE (TM-EXECUTE-SVC ADDR TM))
(TM-SVMODE TM))
(AND (EQUAL (TM-RWSTATE (TM-EXECUTE-SVC ADDR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-SVC ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-SVC ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPOINTS (TM-EXECUTE-SVC ADDR TM))
(TM-OPOINTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-SVC ADDR TM))
(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-SVC ADDR TM))
(TM-SP TM))))))))))))))
((ENABLE TM-EXECUTE-SVC ACCESS-TM-SET-SVCID ACCESS-TM-SET-SVCFLAG)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable registers-untouched-by-tm-execute-svc.

```

(PROVE-LEMMA
REGISTERS-UNTOUCHED-BY-TM-EXECUTE-SVC-RETURN
(REWRITE)
(IMPLIES
(NOT (TM-IN-SUPERVISOR-MODE TM))
(AND
(EQUAL (TM-MEMORY (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-MEMORY TM))
(AND
(EQUAL (TM-REGS (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-REGS TM))
(AND
(EQUAL (TM-CC (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-CC TM))
(AND
(EQUAL (TM-BASE (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-BASE TM))
(AND
(EQUAL (TM-LIMIT (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-LIMIT TM))
(AND
(EQUAL (TM-SLIMIT (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-SLIMIT TM))
(AND
(EQUAL (TM-SVMODE (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-SVMODE TM))
(AND
(EQUAL (TM-SVCFLAG (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-SVCFLAG TM))
(AND
(EQUAL (TM-SVCID (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-SVCID TM))
(AND
(EQUAL (TM-RWSTATE (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-RWSTATE TM))
(AND (EQUAL (TM-CLOCK (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-CLOCK TM))
(AND (EQUAL (TM-IPOINTS (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-IPOINTS TM))
(AND (EQUAL (TM-OPORTS (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-OPORTS TM))
(AND (EQUAL (TM-PC (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-PC TM))

```

(TM-PC TM))
(EQUAL (TM-SP (TM-EXECUTE-SVC-RETURN ADDR TM))
(TM-SP TM)))))))))

((ENABLE TM-EXECUTE-SVC-RETURN ACCESS-TM-SET-ERROR)
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

EVENT: Disable registers-untouched-by-tm-execute-svc-return.

THEOREM: registers-untouched-by-tm-execute-nullary

(\neg tm-in-supervisor-mode (tm))
 \rightarrow ((tm-base (tm-execute-nullary ($opcode$, tm)) = tm-base (tm))
 \wedge ((tm-limit (tm-execute-nullary ($opcode$, tm)) = tm-limit (tm))
 \wedge ((tm-slimit (tm-execute-nullary ($opcode$, tm))
= tm-slimit (tm))
 \wedge ((tm-svmode (tm-execute-nullary ($opcode$, tm))
= tm-svmode (tm))
 \wedge ((tm-svcflag (tm-execute-nullary ($opcode$,
 tm))
= tm-svcflag (tm))
 \wedge ((tm-svcid (tm-execute-nullary ($opcode$,
 tm))
= tm-svcid (tm))
 \wedge ((tm-rwstate (tm-execute-nullary ($opcode$,
 tm))
= tm-rwstate (tm))
 \wedge ((tm-clock (tm-execute-nullary ($opcode$,
 tm))
= tm-clock (tm))
 \wedge ((tm-iports (tm-execute-nullary ($opcode$,
 tm))
= tm-iports (tm))
 \wedge (tm-oports (tm-execute-nullary ($opcode$,
 tm))
= tm-oports (tm)))))))))

EVENT: Disable registers-untouched-by-tm-execute-nullary.

THEOREM: registers-untouched-by-tm-execute-unary

(\neg tm-in-supervisor-mode (tm))
 \rightarrow ((tm-base (tm-execute-unary ($opcode$, $addr$, tm)) = tm-base (tm))
 \wedge ((tm-limit (tm-execute-unary ($opcode$, $addr$, tm)) = tm-limit (tm))
 \wedge ((tm-slimit (tm-execute-unary ($opcode$, $addr$, tm))

$$\begin{aligned}
&= \text{tm-slimit}(tm) \\
\wedge & ((\text{tm-svmode}(\text{tm-execute-unary}(opcode, addr, tm))) \\
&= \text{tm-svmode}(tm)) \\
&\wedge ((\text{tm-rwstate}(\text{tm-execute-unary}(opcode, \\
&\quad addr, \\
&\quad tm))) \\
&= \text{tm-rwstate}(tm)) \\
&\wedge ((\text{tm-clock}(\text{tm-execute-unary}(opcode, \\
&\quad addr, \\
&\quad tm))) \\
&= \text{tm-clock}(tm)) \\
&\wedge ((\text{tm-iports}(\text{tm-execute-unary}(opcode, \\
&\quad addr, \\
&\quad tm))) \\
&= \text{tm-iports}(tm)) \\
&\wedge (\text{tm-oports}(\text{tm-execute-unary}(opcode, \\
&\quad addr, \\
&\quad tm))) \\
&= \text{tm-oports}(tm)))))))))
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-unary.

THEOREM: registers-untouched-by-tm-execute-binary

$$\begin{aligned}
&(\neg \text{tm-in-supervisor-mode}(tm)) \\
\rightarrow & ((\text{tm-base}(\text{tm-execute-binary}(opcode, addr1, addr2, tm)) = \text{tm-base}(tm)) \\
&\wedge ((\text{tm-limit}(\text{tm-execute-binary}(opcode, addr1, addr2, tm)) \\
&= \text{tm-limit}(tm)) \\
&\wedge ((\text{tm-slimit}(\text{tm-execute-binary}(opcode, addr1, addr2, tm)) \\
&= \text{tm-slimit}(tm)) \\
&\wedge ((\text{tm-svmode}(\text{tm-execute-binary}(opcode, \\
&\quad addr1, \\
&\quad addr2, \\
&\quad tm))) \\
&= \text{tm-svmode}(tm)) \\
&\wedge ((\text{tm-svcflag}(\text{tm-execute-binary}(opcode, \\
&\quad addr1, \\
&\quad addr2, \\
&\quad tm))) \\
&= \text{tm-svcflag}(tm)) \\
&\wedge ((\text{tm-svcid}(\text{tm-execute-binary}(opcode, \\
&\quad addr1, \\
&\quad addr2, \\
&\quad tm))) \\
&= \text{tm-svcid}(tm))
\end{aligned}$$

$$\begin{aligned}
& \wedge \left(\left(\text{tm-rwstate} \left(\text{tm-execute-binary} \left(\text{opcode}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \text{addr1}, \\
& \qquad \qquad \qquad \text{addr2}, \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& = \text{tm-rwstate} (tm) \\
& \wedge \left(\left(\text{tm-clock} \left(\text{tm-execute-binary} \left(\text{opcode}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \text{addr1}, \\
& \qquad \qquad \qquad \text{addr2}, \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& = \text{tm-clock} (tm) \\
& \wedge \left(\left(\text{tm-iports} \left(\text{tm-execute-binary} \left(\text{opcode}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \text{addr1}, \\
& \qquad \qquad \qquad \text{addr2}, \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& = \text{tm-iports} (tm) \\
& \wedge \left(\left(\text{tm-oports} \left(\text{tm-execute-binary} \left(\text{opcode}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \text{addr1}, \\
& \qquad \qquad \qquad \text{addr2}, \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& = \text{tm-oports} (tm)
\end{aligned}$$

EVENT: Disable registers-untouched-by-tm-execute-binary.

THEOREM: registers-untouched-by-tm-execute-instruction

$$\begin{aligned}
& (\neg \text{tm-in-supervisor-mode} (tm)) \\
& \rightarrow \left(\left(\text{tm-base} \left(\text{tm-execute-instruction} \left(\text{instruction}, \text{tm} \right) \right) = \text{tm-base} (tm) \right) \right. \\
& \quad \wedge \left(\left(\text{tm-limit} \left(\text{tm-execute-instruction} \left(\text{instruction}, \text{tm} \right) \right) \right. \right. \\
& \qquad = \text{tm-limit} (tm) \\
& \quad \wedge \left(\left(\text{tm-slimit} \left(\text{tm-execute-instruction} \left(\text{instruction}, \text{tm} \right) \right) \right. \right. \\
& \qquad = \text{tm-slimit} (tm) \\
& \quad \wedge \left(\left(\text{tm-svmode} \left(\text{tm-execute-instruction} \left(\text{instruction}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& \qquad = \text{tm-svmode} (tm) \\
& \quad \wedge \left(\left(\text{tm-rwstate} \left(\text{tm-execute-instruction} \left(\text{instruction}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& \qquad = \text{tm-rwstate} (tm) \\
& \quad \wedge \left(\left(\text{tm-clock} \left(\text{tm-execute-instruction} \left(\text{instruction}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& \qquad = \text{tm-clock} (tm) \\
& \quad \wedge \left(\left(\text{tm-iports} \left(\text{tm-execute-instruction} \left(\text{instruction}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right) \\
& \qquad = \text{tm-iports} (tm) \\
& \quad \wedge \left(\left(\text{tm-oports} \left(\text{tm-execute-instruction} \left(\text{instruction}, \right. \right. \right. \right. \\
& \qquad \qquad \qquad \left. \left. \left. \left. \text{tm} \right) \right) \right) \right)
\end{aligned}$$

$$= \text{tm-oports}(tm))))))))))$$

EVENT: Disable registers-untouched-by-tm-execute-instruction.

THEOREM: registers-untouched-by-tm-execute
 $(\neg \text{tm-in-supervisor-mode}(tm))$
 $\rightarrow ((\text{tm-base}(\text{tm-execute}(opcode, tm)) = \text{tm-base}(tm))$
 $\wedge ((\text{tm-limit}(\text{tm-execute}(opcode, tm)) = \text{tm-limit}(tm))$
 $\wedge ((\text{tm-slimit}(\text{tm-execute}(opcode, tm)) = \text{tm-slimit}(tm))$
 $\wedge ((\text{tm-svmode}(\text{tm-execute}(opcode, tm))$
 $= \text{tm-svmode}(tm))$
 $\wedge ((\text{tm-rwstate}(\text{tm-execute}(opcode, tm))$
 $= \text{tm-rwstate}(tm))$
 $\wedge ((\text{tm-clock}(\text{tm-execute}(opcode,$
 $tm))$
 $= \text{tm-clock}(tm))$
 $\wedge ((\text{tm-iports}(\text{tm-execute}(opcode,$
 $tm))$
 $= \text{tm-iports}(tm))$
 $\wedge (\text{tm-oports}(\text{tm-execute}(opcode,$
 $tm))$
 $= \text{tm-oports}(tm))))))))))$

EVENT: Disable registers-untouched-by-tm-execute.

THEOREM: registers-untouched-by-tm-fetch-execute
 $(\neg \text{tm-in-supervisor-mode}(tm))$
 $\rightarrow ((\text{tm-base}(\text{tm-fetch-execute}(tm)) = \text{tm-base}(tm))$
 $\wedge ((\text{tm-limit}(\text{tm-fetch-execute}(tm)) = \text{tm-limit}(tm))$
 $\wedge ((\text{tm-slimit}(\text{tm-fetch-execute}(tm)) = \text{tm-slimit}(tm))$
 $\wedge ((\text{tm-svmode}(\text{tm-fetch-execute}(tm))$
 $= \text{tm-svmode}(tm))$
 $\wedge ((\text{tm-rwstate}(\text{tm-fetch-execute}(tm))$
 $= \text{tm-rwstate}(tm))$
 $\wedge ((\text{tm-iports}(\text{tm-fetch-execute}(tm))$
 $= \text{tm-iports}(tm))$
 $\wedge (\text{tm-oports}(\text{tm-fetch-execute}(tm))$
 $= \text{tm-oports}(tm))))))))))$

EVENT: Disable registers-untouched-by-tm-fetch-execute.

THEOREM: tm-clock-tm-fetch-execute
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge (\neg \text{tm-clock-interruptp}(tm)))$
 $\rightarrow (\text{tm-clock}(\text{tm-fetch-execute}(tm)) = (\text{tm-clock}(tm) - 1))$

EVENT: Disable tm-clock-tm-fetch-execute.

DEFINITION:

base-limit (*base*, *limit*) = cons (*base*, cons (*limit*, 'nil))

DEFINITION: base (*base-limit*) = car (*base-limit*)

DEFINITION: limit (*base-limit*) = car (cdr (*base-limit*))

DEFINITION:

disjoint (*b1*, *l1*, *b2*, *l2*)
= (if *b2* < (*b1* + *l1*) then '*1*false
 else '*1*true endif
 ∨ if *b1* < (*b2* + *l2*) then '*1*false
 else '*1*true endif)

THEOREM: symmetry-of-disjoint

disjoint (*b2*, *l2*, *b1*, *l1*) = disjoint (*b1*, *l1*, *b2*, *l2*)

EVENT: Disable symmetry-of-disjoint.

DEFINITION:

disjoint-everywhere (*b*, *l*, *table*)
= if listp (*table*)
 then disjoint (*b*, *l*, base (car (*table*)), limit (car (*table*)))
 ∧ disjoint-everywhere (*b*, *l*, cdr (*table*))
 else '*1*true endif

DEFINITION:

mutually-disjoint (*table*)
= if listp (*table*)
 then disjoint-everywhere (base (car (*table*)),
 limit (car (*table*)),
 cdr (*table*))
 ∧ mutually-disjoint (cdr (*table*))
 else '*1*true endif

DEFINITION:

finite-segment-tablep (*table*, *max*)
= if listp (*table*)
 then if *max* < (base (car (*table*)) + limit (car (*table*)))
 then '*1*false
 else '*1*true endif
 ∧ finite-segment-tablep (cdr (*table*), *max*)
 else '*1*true endif

THEOREM: disjoint-getnth
 $(\text{disjoint-everywhere}(b, l, \text{table}) \wedge (i < \text{length}(\text{table})))$
 $\rightarrow \text{disjoint}(b, l, \text{base}(\text{getnth}(i, \text{table})), \text{limit}(\text{getnth}(i, \text{table})))$

EVENT: Disable disjoint-getnth.

THEOREM: tm-good-address-tm-set-cc
 $\text{tm-good-address}(\text{addr}, \text{tm-set-cc}(cc, tm)) = \text{tm-good-address}(\text{addr}, tm)$

EVENT: Disable tm-good-address-tm-set-cc.

THEOREM: tm-good-address-tm-set-pc
 $\text{tm-good-address}(\text{addr}, \text{tm-set-pc}(pc, tm)) = \text{tm-good-address}(\text{addr}, tm)$

EVENT: Disable tm-good-address-tm-set-pc.

THEOREM: tm-good-address-tm-set-sp
 $\text{tm-good-address}(\text{addr}, \text{tm-set-sp}(sp, tm)) = \text{tm-good-address}(\text{addr}, tm)$

EVENT: Disable tm-good-address-tm-set-sp.

THEOREM: tm-store-in-memory-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm))$
 $\wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm))$
 $\wedge \text{finite-numberp}(\text{addr},$
 $\quad \text{min}(65536 - \text{tm-base}(tm), \text{tm-limit}(tm))))))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-store-in-memory}(\text{value}, \text{addr}, tm)))$
 $\quad = \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-store-in-memory-protection-theorem.

THEOREM: tm-store-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm))$
 $\wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \wedge \text{tm-good-address}(\text{addr}, tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-store}(\text{value}, \text{addr}, tm)))$
 $\quad = \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-store-protection-theorem.

THEOREM: tm-execute-add-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm))$
 $\wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$

$$\begin{aligned} & \wedge \text{tm-good-address}(addr1, tm)) \\ \rightarrow & (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-add}(addr1, addr2, tm))) \\ & = \text{getseg}(b, l, \text{tm-memory}(tm))) \end{aligned}$$

EVENT: Disable tm-execute-add-protection-theorem.

THEOREM: tm-execute-branch-protection-theorem
 $\text{tm-memory}(\text{tm-execute-branch}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-branch-protection-theorem.

THEOREM: tm-execute-branch-on-zero-protection-theorem
 $\text{tm-memory}(\text{tm-execute-branch-on-zero}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-branch-on-zero-protection-theorem.

THEOREM: tm-execute-branch-not-zero-protection-theorem
 $\text{tm-memory}(\text{tm-execute-branch-not-zero}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-branch-not-zero-protection-theorem.

THEOREM: tm-execute-call-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm))$
 $\wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \wedge \text{tm-good-address}(addr, tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-call}(addr, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-call-protection-theorem.

THEOREM: tm-execute-compare-protection-theorem
 $\text{tm-memory}(\text{tm-execute-compare}(addr1, addr2, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-compare-protection-theorem.

THEOREM: tm-execute-decr-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm))$
 $\wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \wedge \text{tm-good-address}(addr, tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-decr}(addr, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-decr-protection-theorem.

THEOREM: tm-execute-decr-mod-protection-theorem

$$\begin{aligned} & ((\neg \text{tm-in-supervisor-mode}(tm)) \\ & \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \\ & \quad \wedge \text{tm-good-address}(addr1, tm))) \\ \rightarrow & (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-decr-mod}(addr1, addr2, tm))) \\ & = \text{getseg}(b, l, \text{tm-memory}(tm))) \end{aligned}$$

EVENT: Disable tm-execute-decr-mod-protection-theorem.

THEOREM: tm-execute-incr-protection-theorem

$$\begin{aligned} & ((\neg \text{tm-in-supervisor-mode}(tm)) \\ & \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \wedge \text{tm-good-address}(addr, tm))) \\ \rightarrow & (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-incr}(addr, tm))) \\ & = \text{getseg}(b, l, \text{tm-memory}(tm))) \end{aligned}$$

EVENT: Disable tm-execute-incr-protection-theorem.

THEOREM: tm-execute-incr-mod-protection-theorem

$$\begin{aligned} & ((\neg \text{tm-in-supervisor-mode}(tm)) \\ & \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \\ & \quad \wedge \text{tm-good-address}(addr1, tm))) \\ \rightarrow & (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-incr-mod}(addr1, addr2, tm))) \\ & = \text{getseg}(b, l, \text{tm-memory}(tm))) \end{aligned}$$

EVENT: Disable tm-execute-incr-mod-protection-theorem.

THEOREM: tm-execute-load-base-protection-theorem

$$\text{tm-memory}(\text{tm-execute-load-base}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-load-base-protection-theorem.

THEOREM: tm-execute-load-limit-protection-theorem

$$\text{tm-memory}(\text{tm-execute-load-limit}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-load-limit-protection-theorem.

THEOREM: tm-execute-load-psw-protection-theorem

$$\text{tm-memory}(\text{tm-execute-load-psw}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-load-psw-protection-theorem.

THEOREM: tm-execute-mod-protection-theorem

$$\begin{aligned}
& ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \quad \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \\
& \quad \quad \wedge \text{tm-good-address}(addr1, tm))) \\
& \rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-mod}(addr1, addr2, tm))) \\
& \quad = \text{getseg}(b, l, \text{tm-memory}(tm)))
\end{aligned}$$

EVENT: Disable tm-execute-mod-protection-theorem.

THEOREM: tm-execute-move-protection-theorem

$$\begin{aligned}
& ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \quad \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \\
& \quad \quad \wedge \text{tm-good-address}(addr1, tm))) \\
& \rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-move}(addr1, addr2, tm))) \\
& \quad = \text{getseg}(b, l, \text{tm-memory}(tm)))
\end{aligned}$$

EVENT: Disable tm-execute-move-protection-theorem.

THEOREM: tm-execute-multiply-protection-theorem

$$\begin{aligned}
& ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \quad \wedge (\text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)) \\
& \quad \quad \wedge \text{tm-good-address}(addr1, tm))) \\
& \rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-multiply}(addr1, addr2, tm))) \\
& \quad = \text{getseg}(b, l, \text{tm-memory}(tm)))
\end{aligned}$$

EVENT: Disable tm-execute-multiply-protection-theorem.

THEOREM: tm-execute-return-protection-theorem

$$\text{tm-memory}(\text{tm-execute-return}(tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-return-protection-theorem.

THEOREM: tm-execute-test-iptest-protection-theorem

$$\text{tm-memory}(\text{tm-execute-test-iptest}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-test-iptest-protection-theorem.

THEOREM: tm-execute-test-oport-protection-theorem

$$\text{tm-memory}(\text{tm-execute-test-oport}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-test-oport-protection-theorem.

THEOREM: tm-execute-set-clock-protection-theorem

$$\text{tm-memory}(\text{tm-execute-set-clock}(addr, tm)) = \text{tm-memory}(tm)$$

EVENT: Disable tm-execute-set-clock-protection-theorem.

THEOREM: tm-execute-run-protection-theorem
 $\text{tm-memory}(\text{tm-execute-run}(tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-run-protection-theorem.

THEOREM: tm-execute-wait-protection-theorem
 $\text{tm-memory}(\text{tm-execute-wait}(tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-wait-protection-theorem.

THEOREM: tm-execute-start-output-protection-theorem
 $\text{tm-memory}(\text{tm-execute-start-output}(addr1, addr2, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-start-output-protection-theorem.

THEOREM: tm-execute-post-output-interrupt-protection-theorem
 $\text{tm-memory}(\text{tm-execute-post-output-interrupt}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-post-output-interrupt-protection-theorem.

THEOREM: tm-execute-svc-protection-theorem
 $\text{tm-memory}(\text{tm-execute-svc}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-svc-protection-theorem.

THEOREM: tm-execute-svc-return-protection-theorem
 $\text{tm-memory}(\text{tm-execute-svc-return}(addr, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-svc-return-protection-theorem.

THEOREM: tm-execute-nullary-protection-theorem
 $\text{tm-memory}(\text{tm-execute-nullary}(opcode, tm)) = \text{tm-memory}(tm)$

EVENT: Disable tm-execute-nullary-protection-theorem.

THEOREM: tm-execute-unary-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-unary}(opcode, addr, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-unary-protection-theorem.

THEOREM: tm-execute-binary-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-binary}(opcode, addr1, addr2, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-binary-protection-theorem.

THEOREM: tm-execute-instruction-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute-instruction}(instruction, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-instruction-protection-theorem.

THEOREM: tm-execute-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-execute}(opcode, tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-execute-protection-theorem.

THEOREM: tm-fetch-execute-protection-theorem
 $((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{disjoint}(b, l, \text{tm-base}(tm), \text{tm-limit}(tm)))$
 $\rightarrow (\text{getseg}(b, l, \text{tm-memory}(\text{tm-fetch-execute}(tm)))$
 $= \text{getseg}(b, l, \text{tm-memory}(tm)))$

EVENT: Disable tm-fetch-execute-protection-theorem.

THEOREM: length-tm-memory-tm-store
 $\text{length}(\text{tm-memory}(\text{tm-store}(value, addr, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-store.

THEOREM: length-tm-memory-tm-execute-add
 $\text{length}(\text{tm-memory}(\text{tm-execute-add}(addr1, addr2, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-add.

THEOREM: length-tm-memory-tm-execute-branch
 $\text{length}(\text{tm-memory}(\text{tm-execute-branch}(addr, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-branch.

THEOREM: length-tm-memory-tm-execute-branch-on-zero
 $\text{length}(\text{tm-memory}(\text{tm-execute-branch-on-zero}(addr, tm)))$
 $= \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-branch-on-zero.

THEOREM: length-tm-memory-tm-execute-branch-not-zero
 $\text{length}(\text{tm-memory}(\text{tm-execute-branch-not-zero}(addr, tm)))$
 $= \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-branch-not-zero.

THEOREM: length-tm-memory-tm-execute-call
 $\text{length}(\text{tm-memory}(\text{tm-execute-call}(addr, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-call.

THEOREM: length-tm-memory-tm-execute-compare
 $\text{length}(\text{tm-memory}(\text{tm-execute-compare}(addr1, addr2, tm)))$
 $= \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-compare.

THEOREM: length-tm-memory-tm-execute-decr
 $\text{length}(\text{tm-memory}(\text{tm-execute-decr}(addr, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-decr.

THEOREM: length-tm-memory-tm-execute-decr-mod
 $\text{length}(\text{tm-memory}(\text{tm-execute-decr-mod}(addr1, addr2, tm)))$
 $= \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-decr-mod.

THEOREM: length-tm-memory-tm-execute-incr
 $\text{length}(\text{tm-memory}(\text{tm-execute-incr}(addr, tm))) = \text{length}(\text{tm-memory}(tm))$

EVENT: Disable length-tm-memory-tm-execute-incr.

THEOREM: length-tm-memory-tm-execute-incr-mod
length (tm-memory (tm-execute-incr-mod (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-incr-mod.

THEOREM: length-tm-memory-tm-execute-load-base
length (tm-memory (tm-execute-load-base (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-load-base.

THEOREM: length-tm-memory-tm-execute-load-limit
length (tm-memory (tm-execute-load-limit (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-load-limit.

THEOREM: length-tm-memory-tm-execute-load-psw
length (tm-memory (tm-execute-load-psw (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-load-psw.

THEOREM: length-tm-memory-tm-execute-mod
length (tm-memory (tm-execute-mod (*addr1*, *addr2*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-mod.

THEOREM: length-tm-memory-tm-execute-move
length (tm-memory (tm-execute-move (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-move.

THEOREM: length-tm-memory-tm-execute-multiply
length (tm-memory (tm-execute-multiply (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-multiply.

THEOREM: length-tm-memory-tm-execute-return
length (tm-memory (tm-execute-return (*tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-return.

THEOREM: length-tm-memory-tm-execute-test-iport
 $\text{length}(\text{tm-memory}(\text{tm-execute-test-iport}(\text{addr}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-test-iport.

THEOREM: length-tm-memory-tm-execute-test-oport
 $\text{length}(\text{tm-memory}(\text{tm-execute-test-oport}(\text{addr}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-test-oport.

THEOREM: length-tm-memory-tm-execute-set-clock
 $\text{length}(\text{tm-memory}(\text{tm-execute-set-clock}(\text{addr}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-set-clock.

THEOREM: length-tm-memory-tm-execute-run
 $\text{length}(\text{tm-memory}(\text{tm-execute-run}(\text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-run.

THEOREM: length-tm-memory-tm-execute-wait
 $\text{length}(\text{tm-memory}(\text{tm-execute-wait}(\text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-wait.

THEOREM: length-tm-memory-tm-execute-post-output-interrupt
 $\text{length}(\text{tm-memory}(\text{tm-execute-post-output-interrupt}(\text{addr}, \text{tm})))$
 $= \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-post-output-interrupt.

THEOREM: length-tm-memory-tm-execute-start-output
 $\text{length}(\text{tm-memory}(\text{tm-execute-start-output}(\text{addr1}, \text{addr2}, \text{tm})))$
 $= \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-start-output.

THEOREM: length-tm-memory-tm-execute-svc
 $\text{length}(\text{tm-memory}(\text{tm-execute-svc}(\text{addr}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-svc.

THEOREM: length-tm-memory-tm-execute-svc-return
 $\text{length}(\text{tm-memory}(\text{tm-execute-svc-return}(\text{addr}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-svc-return.

THEOREM: length-tm-memory-tm-execute-nullary
 $\text{length}(\text{tm-memory}(\text{tm-execute-nullary}(\text{opcode}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-nullary.

THEOREM: length-tm-memory-tm-execute-unary
 $\text{length}(\text{tm-memory}(\text{tm-execute-unary}(\text{opcode}, \text{addr}, \text{tm})))$
 $= \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-unary.

THEOREM: length-tm-memory-tm-execute-binary
 $\text{length}(\text{tm-memory}(\text{tm-execute-binary}(\text{opcode}, \text{addr1}, \text{addr2}, \text{tm})))$
 $= \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-binary.

THEOREM: length-tm-memory-tm-execute-instruction
 $\text{length}(\text{tm-memory}(\text{tm-execute-instruction}(\text{instruction}, \text{tm})))$
 $= \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute-instruction.

THEOREM: length-tm-memory-tm-execute
 $\text{length}(\text{tm-memory}(\text{tm-execute}(\text{instruction}, \text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-execute.

THEOREM: length-tm-memory-tm-fetch-execute
 $\text{length}(\text{tm-memory}(\text{tm-fetch-execute}(\text{tm}))) = \text{length}(\text{tm-memory}(\text{tm}))$

EVENT: Disable length-tm-memory-tm-fetch-execute.

(DEFN
GOOD-ADDRESS-SPACE

```

(X MEMLENGTH)
(AND
  (TM-SHELLP X)
  (AND
    (NUMBERP MEMLENGTH)
    (AND
      (IF (LESSP '65536 MEMLENGTH)
          '*1*FALSE
          '*1*TRUE)
      (AND
        (PLISTP (TM-MEMORY X))
        (AND
          (FINITE-NUMBER-LISTP (TM-MEMORY X)
                               '65536)
          (AND
            (AND
              (EQUAL (LENGTH (TM-MEMORY X))
                     MEMLENGTH)
              (AND
                (PLISTP (TM-REGS X))
                (AND (FINITE-NUMBER-LISTP (TM-REGS X)
                                           '65536)
                    (AND (EQUAL (LENGTH (TM-REGS X)) '8)
                        (AND (FINITE-NUMBERP (TM-CC X) '4)
                            (AND (FINITE-NUMBERP (TM-ERROR X) '64)
                                (AND (FINITE-NUMBERP (TM-SVCFLAG X) '2)
                                    (AND (FINITE-NUMBERP (TM-SVCID X) '128)
                                        (AND (EQUAL (TM-BASE X) '0)
                                            (AND (EQUAL (TM-LIMIT X) MEMLENGTH)
                                                (EQUAL (TM-SVMODE X)
                                                    '0))))))))))))))))))
          NIL)

```

THEOREM: tm-shellp-address-space
 good-address-space (*address-space*, *memlength*) \rightarrow tm-shellp (*address-space*)

EVENT: Disable tm-shellp-address-space.

THEOREM: non-number-memlength-gives-bad-address-space
 (*memlength* $\notin \mathbf{N}$) \rightarrow (\neg good-address-space (*address-space*, *memlength*))

EVENT: Disable non-number-memlength-gives-bad-address-space.

THEOREM: address-space-memlength
 good-address-space (*address-space*, *memlength*)

→ **if** '65536 < *memlength* **then** '*1***false**
 else '*1***true** **endif**

THEOREM: plistp-address-space-memory
good-address-space (*address-space*, *memlength*)
→ plistp (tm-memory (*address-space*))

EVENT: Disable plistp-address-space-memory.

THEOREM: finite-number-listp-address-space-memory
good-address-space (*address-space*, *memlength*)
→ finite-number-listp (tm-memory (*address-space*), '65536)

EVENT: Disable finite-number-listp-address-space-memory.

THEOREM: length-address-space-memory
good-address-space (*address-space*, *memlength*)
→ (length (tm-memory (*address-space*)) = *memlength*)

EVENT: Disable length-address-space-memory.

THEOREM: plistp-address-space-regs
good-address-space (*address-space*, *memlength*)
→ plistp (tm-regs (*address-space*))

EVENT: Disable plistp-address-space-regs.

THEOREM: finite-number-listp-address-space-regs
good-address-space (*address-space*, *memlength*)
→ finite-number-listp (tm-regs (*address-space*), '65536)

EVENT: Disable finite-number-listp-address-space-regs.

THEOREM: length-address-space-regs
good-address-space (*address-space*, *memlength*)
→ (length (tm-regs (*address-space*)) = '8)

EVENT: Disable length-address-space-regs.

THEOREM: finite-numberp-address-space-cc
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-cc (*address-space*), '4)

EVENT: Disable finite-numberp-address-space-cc.

THEOREM: finite-numberp-address-space-error
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-error (*address-space*), '64)

EVENT: Disable finite-numberp-address-space-error.

THEOREM: finite-numberp-address-space-svcflag
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-svcflag (*address-space*), '2)

EVENT: Disable finite-numberp-address-space-svcflag.

THEOREM: finite-numberp-address-space-svcid
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-svcid (*address-space*), '128)

EVENT: Disable finite-numberp-address-space-svcid.

THEOREM: finite-numberp-address-space-pc
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-pc (*address-space*), '65536)

EVENT: Disable finite-numberp-address-space-pc.

THEOREM: finite-numberp-address-space-sp
good-address-space (*address-space*, *memlength*)
→ finite-numberp (tm-sp (*address-space*), '65536)

EVENT: Disable finite-numberp-address-space-sp.

THEOREM: tm-base-address-space
good-address-space (*address-space*, *memlength*)
→ (tm-base (*address-space*) = '0)

EVENT: Disable tm-base-address-space.

THEOREM: tm-limit-address-space
good-address-space (*address-space*, *memlength*)
→ (tm-limit (*address-space*) = *memlength*)

EVENT: Disable tm-limit-address-space.

THEOREM: tm-svmode-address-space
good-address-space (*address-space*, *memlength*)
→ (tm-svmode (*address-space*) = '0)

EVENT: Disable tm-svmode-address-space.

THEOREM: tm-in-supervisor-mode-good-address-space
good-address-space (*address-space*, *memlength*)
→ (tm-in-supervisor-mode (*address-space*) = '*1*false)

EVENT: Disable tm-in-supervisor-mode-good-address-space.

EVENT: Let us define the theory *good-address-space-properties* to consist of the following events: tm-in-supervisor-mode-good-address-space, non-number-memlength-gives-bad-address-space, tm-svmode-address-space, tm-limit-address-space, tm-base-address-space, finite-numberp-address-space-svcid, finite-numberp-address-space-svcflag, finite-numberp-address-space-error, finite-numberp-address-space-cc, length-address-space-regs, finite-number-listp-address-space-regs, plistp-address-space-regs, length-address-space-memory, finite-number-listp-address-space-memory, plistp-address-space-memory, address-space-memlength, tm-shellp-address-space, finite-numberp-address-space-pc, finite-numberp-address-space-sp.

THEOREM: good-address-space-base-limit-property
good-address-space (*address-space*, *memlength*)
→ (min ('65536 - tm-base (*address-space*), tm-limit (*address-space*))
= tm-limit (*address-space*))

EVENT: Disable good-address-space-base-limit-property.

THEOREM: min-tm-memlength-memlength-in-good-address-space
good-address-space (*address-space*, *memlength*)
→ (min ('65536, *memlength*) = *memlength*)

EVENT: Disable min-tm-memlength-memlength-in-good-address-space.

THEOREM: good-memory-address-implies-non-zero-limit-for-address-space
(good-address-space (*address-space*, *memlength*)
∧ tm-good-address (real-addr ('2, *pc*), *address-space*))
→ (tm-limit (*address-space*) ≠ 0)

THEOREM: finite-numberp-tm-good-address-for-address-space
 (good-address-space (*address-space*, *memlength*)
 ∧ tm-good-address (real-addr ('2, *n*), *address-space*))
 → finite-numberp (*n*, *memlength*)

EVENT: Disable finite-numberp-tm-good-address-for-address-space.

THEOREM: finite-numberp-tm-fetch-from-address-space
 (good-address-space (*address-space*, *memlength*)
 ∧ tm-good-address (*addr*, *address-space*))
 → finite-numberp (tm-fetch (*addr*, *address-space*), '65536)

EVENT: Disable finite-numberp-tm-fetch-from-address-space.

THEOREM: address-space-good-memory-tm-store
 (good-address-space (*address-space*, *memlength*)
 ∧ finite-numberp (*value*, '65536))
 → (plistp (tm-memory (tm-store (*value*, *addr*, *address-space*)))
 ∧ ((length (tm-memory (tm-store (*value*, *addr*, *address-space*)))
 = fix (*memlength*))
 ∧ finite-number-listp (tm-memory (tm-store (*value*,
addr,
address-space)),
 '65536)))

EVENT: Disable address-space-good-memory-tm-store.

THEOREM: address-space-good-regs-tm-store
 (good-address-space (*address-space*, *memlength*)
 ∧ finite-numberp (*value*, '65536))
 → (plistp (tm-regs (tm-store (*value*, *addr*, *address-space*)))
 ∧ ((length (tm-regs (tm-store (*value*, *addr*, *address-space*))) = '8)
 ∧ finite-number-listp (tm-regs (tm-store (*value*,
addr,
address-space)),
 '65536)))

EVENT: Disable address-space-good-regs-tm-store.

THEOREM: good-address-space-tm-store
 (good-address-space (*address-space*, *memlength*)
 ∧ finite-numberp (*value*, '65536))
 → good-address-space (tm-store (*value*, *addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-store.

THEOREM: good-address-space-tm-set-cc
(good-address-space (*address-space*, *memlength*) \wedge finite-numberp (*cc*, '4))
 \rightarrow good-address-space (tm-set-cc (*cc*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-cc.

THEOREM: good-address-space-tm-set-error
(good-address-space (*address-space*, *memlength*) \wedge finite-numberp (*error*, '64))
 \rightarrow good-address-space (tm-set-error (*error*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-error.

THEOREM: good-address-space-tm-set-svcflag
(good-address-space (*address-space*, *memlength*) \wedge finite-numberp (*svcflag*, '2))
 \rightarrow good-address-space (tm-set-svcflag (*svcflag*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-svcflag.

THEOREM: good-address-space-tm-set-svcid
(good-address-space (*address-space*, *memlength*)
 \wedge finite-numberp (*svcid*, '128))
 \rightarrow good-address-space (tm-set-svcid (*svcid*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-svcid.

THEOREM: good-address-space-tm-set-pc
(good-address-space (*address-space*, *memlength*) \wedge finite-numberp (*pc*, '65536))
 \rightarrow good-address-space (tm-set-pc (*pc*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-pc.

THEOREM: good-address-space-tm-set-sp
(good-address-space (*address-space*, *memlength*) \wedge finite-numberp (*sp*, '65536))
 \rightarrow good-address-space (tm-set-sp (*sp*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-set-sp.

THEOREM: good-address-space-tm-execute-add
good-address-space (*address-space*, *memlength*)
 \rightarrow good-address-space (tm-execute-add (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-add.

THEOREM: good-address-space-tm-execute-branch
(good-address-space (*address-space*, *memlength*)
 \wedge tm-good-address (*addr*, *address-space*))
→ good-address-space (tm-execute-branch (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-branch.

THEOREM: good-address-space-tm-execute-branch-on-zero
(good-address-space (*address-space*, *memlength*)
 \wedge tm-good-address (*addr*, *address-space*))
→ good-address-space (tm-execute-branch-on-zero (*addr*, *address-space*),
 memlength)

EVENT: Disable good-address-space-tm-execute-branch-on-zero.

THEOREM: good-address-space-tm-execute-branch-not-zero
(good-address-space (*address-space*, *memlength*)
 \wedge tm-good-address (*addr*, *address-space*))
→ good-address-space (tm-execute-branch-not-zero (*addr*, *address-space*),
 memlength)

EVENT: Disable good-address-space-tm-execute-branch-not-zero.

THEOREM: good-address-space-tm-execute-call
(good-address-space (*address-space*, *memlength*)
 \wedge tm-good-address (*addr*, *address-space*))
→ good-address-space (tm-execute-call (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-call.

THEOREM: good-address-space-tm-execute-compare
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-compare (*addr1*, *addr2*, *address-space*),
 memlength)

EVENT: Disable good-address-space-tm-execute-compare.

THEOREM: good-address-space-tm-execute-decr
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-decr (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-decr.

THEOREM: good-address-space-tm-execute-decr-mod
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-decr-mod (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-decr-mod.

THEOREM: good-address-space-tm-execute-incr
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-incr (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-incr.

THEOREM: good-address-space-tm-execute-incr-mod
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-incr-mod (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-incr-mod.

THEOREM: good-address-space-tm-execute-load-base
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-load-base (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-load-base.

THEOREM: good-address-space-tm-execute-load-limit
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-load-limit (*addr*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-load-limit.

THEOREM: good-address-space-tm-execute-load-psw
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-load-psw (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-load-psw.

THEOREM: good-address-space-tm-execute-mod
 good-address-space (*address-space*, *memlength*)
 \rightarrow good-address-space (tm-execute-mod (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-mod.

THEOREM: good-address-space-tm-execute-move
 (good-address-space (*address-space*, *memlength*)
 \wedge tm-good-address (*addr2*, *address-space*))
 \rightarrow good-address-space (tm-execute-move (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-move.

THEOREM: good-address-space-tm-execute-multiply
 good-address-space (*address-space*, *memlength*)
 \rightarrow good-address-space (tm-execute-multiply (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-multiply.

THEOREM: finite-numberp-add1-address-space-sp
 (good-address-space (*address-space*, *memlength*)
 \wedge (\neg tm-stack-underflowp (*address-space*)))
 \rightarrow finite-numberp (1 + tm-sp (*address-space*), '65536)

EVENT: Disable finite-numberp-add1-address-space-sp.

THEOREM: min-memlength-tm-memlength-in-good-address-space
 good-address-space (*address-space*, *memlength*)
 \rightarrow (min ('65536, *memlength*) = *memlength*)

THEOREM: finite-numberp-tm-incr-memlength
 ((*memlength* \in \mathbf{N})
 \wedge ((min ('65536, *memlength*) = *memlength*) \wedge (*b* < (*memlength* - 1))))
 \rightarrow finite-numberp (tm-incr (*b*), *memlength*)

EVENT: Disable finite-numberp-tm-incr-memlength.

THEOREM: tm-good-address-tm-incr-tm-sp-for-address-space
 (good-address-space (*address-space*, *memlength*)
 \wedge (\neg tm-stack-underflowp (*address-space*)))

→ $\text{tm-good-address}(\text{real-addr}('2, \text{tm-incr}(\text{tm-sp}(\text{address-space}))),$
 $\text{address-space})$

EVENT: Disable tm-good-address-tm-incr-tm-sp-for-address-space.

THEOREM: good-address-space-tm-execute-return
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-return}(\text{address-space}), \text{memlength})$

EVENT: Disable good-address-space-tm-execute-return.

THEOREM: good-address-space-tm-execute-set-clock
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-set-clock}(\text{addr}, \text{address-space}), \text{memlength})$

EVENT: Disable good-address-space-tm-execute-set-clock.

THEOREM: good-address-space-tm-execute-run
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-run}(\text{address-space}), \text{memlength})$

EVENT: Disable good-address-space-tm-execute-run.

THEOREM: good-address-space-tm-execute-wait
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-wait}(\text{address-space}), \text{memlength})$

EVENT: Disable good-address-space-tm-execute-wait.

THEOREM: good-address-space-tm-execute-test-iport
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-test-iport}(\text{addr}, \text{address-space}),$
 $\text{memlength})$

EVENT: Disable good-address-space-tm-execute-test-iport.

THEOREM: good-address-space-tm-execute-test-oport
 $\text{good-address-space}(\text{address-space}, \text{memlength})$
→ $\text{good-address-space}(\text{tm-execute-test-oport}(\text{addr}, \text{address-space}),$
 $\text{memlength})$

EVENT: Disable good-address-space-tm-execute-test-oport.

THEOREM: good-address-space-tm-execute-start-output
 $\text{good-address-space}(address\text{-space}, memlength)$
 $\rightarrow \text{good-address-space}(\text{tm-execute-start-output}(addr1, addr2, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-start-output.

THEOREM: good-address-space-tm-execute-post-output-interrupt
 $\text{good-address-space}(address\text{-space}, memlength)$
 $\rightarrow \text{good-address-space}(\text{tm-execute-post-output-interrupt}(addr, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-post-output-interrupt.

THEOREM: good-address-space-tm-execute-svc
 $(\text{good-address-space}(address\text{-space}, memlength)$
 $\wedge \text{tm-good-address}(addr, address\text{-space}))$
 $\rightarrow \text{good-address-space}(\text{tm-execute-svc}(addr, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-svc.

THEOREM: good-address-space-tm-execute-svc-return
 $\text{good-address-space}(address\text{-space}, memlength)$
 $\rightarrow \text{good-address-space}(\text{tm-execute-svc-return}(addr, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-svc-return.

THEOREM: good-address-space-tm-execute-nullary
 $\text{good-address-space}(address\text{-space}, memlength)$
 $\rightarrow \text{good-address-space}(\text{tm-execute-nullary}(opcode, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-nullary.

THEOREM: good-address-space-tm-execute-unary
 $\text{good-address-space}(address\text{-space}, memlength)$
 $\rightarrow \text{good-address-space}(\text{tm-execute-unary}(opcode, addr, address\text{-space}), memlength)$

EVENT: Disable good-address-space-tm-execute-unary.

THEOREM: good-address-space-tm-execute-binary
 good-address-space (*address-space*, *memlength*)
 → good-address-space (tm-execute-binary (*opcode*, *addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-binary.

THEOREM: good-address-space-tm-execute-instruction
 good-address-space (*address-space*, *memlength*)
 → good-address-space (tm-execute-instruction (*instruction*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-instruction.

THEOREM: good-address-space-tm-execute
 good-address-space (*address-space*, *memlength*)
 → good-address-space (tm-execute (*instruction*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute.

THEOREM: good-address-space-tm-decrement-clock
 good-address-space (*address-space*, *memlength*)
 → good-address-space (tm-decrement-clock (*address-space*), *memlength*)

EVENT: Disable good-address-space-tm-decrement-clock.

THEOREM: good-address-space-tm-fetch-execute
 good-address-space (*address-space*, *memlength*)
 → good-address-space (tm-fetch-execute (*address-space*), *memlength*)

EVENT: Disable good-address-space-tm-fetch-execute.

DEFINITION:
 mapup-address-space (*memory*, *regs*, *cc*, *error*, *svcflag*, *svcid*, *base*, *limit*)
 = tm (getseg (*base*, *limit*, *memory*),
 regs,
 cc,
 error,
 svcflag,
 svcid,
 '0',
 fix (*limit*),


```
'0,
'0,
'0,
'0,
'0,
'0)
```

```
(PROVE-LEMMA GOOD-ADDRESS-SPACE-MAPUP-ADDRESS-SPACE
(REWRITE)
(IMPLIES
(AND
(FINITE-NUMBER-LISTP MEMORY '65536)
(AND (NUMBERP LIMIT)
(AND (IF (LESSP '65536 (LENGTH MEMORY))
'*1*FALSE
'*1*TRUE)
(AND (PLISTP REGS)
(AND (FINITE-NUMBER-LISTP REGS '65536)
(AND (EQUAL (LENGTH REGS) '8)
(AND (FINITE-NUMBERP CC '4)
(AND (FINITE-NUMBERP ERROR '64)
(AND (FINITE-NUMBERP SVCFLAG '2)
(AND (FINITE-NUMBERP SVCID '128)
(IF (LESSP (LENGTH MEMORY)
(PLUS BASE LIMIT))
'*1*FALSE
'*1*TRUE))))))))))
(GOOD-ADDRESS-SPACE (MAPUP-ADDRESS-SPACE MEMORY REGS CC ERROR SVCFLAG SVCID
BASE LIMIT)
LIMIT))
((ENABLE GOOD-ADDRESS-SPACE MAPUP-ADDRESS-SPACE PLISTP-GETSEG LENGTH-GETSEG
FINITE-NUMBER-LISTP-GETSEG)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T)))
```

EVENT: Disable good-address-space-mapup-address-space.

THEOREM: tm-fetch-from-memory-mapup-address-space
(good-tm (*tm*)
 $\wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \wedge (addr < \text{tm-limit}(tm)))$
 $\rightarrow (\text{tm-fetch-from-memory}(addr,$
mapup-address-space (tm-memory (*tm*),

$$\begin{aligned}
& \text{tm-regs}(tm), \\
& \text{tm-cc}(tm), \\
& \text{tm-error}(tm), \\
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{tm-fetch-from-memory}(addr, tm)
\end{aligned}$$

EVENT: Disable tm-fetch-from-memory-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: tm-fetch-from-regmem-mapup-address-space} \\
& \text{good-tm}(tm) \\
\rightarrow & (\text{tm-fetch-from-regmem}(addr, \\
& \qquad \qquad \qquad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \qquad \qquad \text{tm-regs}(tm), \\
& \qquad \qquad \qquad \text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm), \\
& \qquad \qquad \qquad \text{tm-base}(tm), \\
& \qquad \qquad \qquad \text{tm-limit}(tm))) \\
= & \text{tm-fetch-from-regmem}(addr, tm)
\end{aligned}$$

EVENT: Disable tm-fetch-from-regmem-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: lessp-good-memory-address-limit} \\
& (\text{good-tm}(tm) \\
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \quad \wedge ((\text{real-addr-source}(addr) \neq 0) \\
& \quad \quad \wedge ((\text{real-addr-source}(addr) \neq '1) \\
& \quad \quad \quad \wedge \text{tm-good-address}(addr, tm)))))) \\
\rightarrow & (\text{real-addr-num}(addr) < \text{tm-limit}(tm))
\end{aligned}$$

$$\begin{aligned}
& \text{THEOREM: tm-fetch-mapup-address-space} \\
& (\text{good-tm}(tm) \\
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{tm-good-address}(addr, tm))) \\
\rightarrow & (\text{tm-fetch}(addr, \\
& \qquad \qquad \qquad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \qquad \qquad \text{tm-regs}(tm), \\
& \qquad \qquad \qquad \text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-regs } (tm), \\
& \text{tm-cc } (tm), \\
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm)) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-store-in-regmem } (value, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-regs } (\text{tm-store-in-regmem } (value, \text{addr}, tm)), \\
& \text{tm-cc } (\text{tm-store-in-regmem } (value, \text{addr}, tm)), \\
& \text{tm-error } (\text{tm-store-in-regmem } (value, \text{addr}, tm)), \\
& \text{tm-svcflag } (\text{tm-store-in-regmem } (value, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-svcid } (\text{tm-store-in-regmem } (value, \text{addr}, tm)), \\
& \text{tm-base } (\text{tm-store-in-regmem } (value, \text{addr}, tm)), \\
& \text{tm-limit } (\text{tm-store-in-regmem } (value, \text{addr}, tm))))
\end{aligned}$$

EVENT: Disable tm-store-in-regmem-commutes-with-mapup-address-space.

THEOREM: tm-store-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm } (tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base } (tm) + \text{tm-limit } (tm)) \text{ then } '*1*\text{false} \\
& \text{else } '*1*\text{true} \text{ endif} \\
& \wedge ((\neg \text{tm-in-supervisor-mode } (tm)) \\
& \wedge \text{tm-good-address } (\text{addr}, tm)))) \\
\rightarrow & (\text{tm-store } (value, \\
& \text{addr}, \\
& \text{mapup-address-space } (\text{tm-memory } (tm), \\
& \text{tm-regs } (tm), \\
& \text{tm-cc } (tm), \\
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm))) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-store } (value, \text{addr}, tm)), \\
& \text{tm-regs } (\text{tm-store } (value, \text{addr}, tm)), \\
& \text{tm-cc } (\text{tm-store } (value, \text{addr}, tm)), \\
& \text{tm-error } (\text{tm-store } (value, \text{addr}, tm)), \\
& \text{tm-svcflag } (\text{tm-store } (value, \text{addr}, tm)), \\
& \text{tm-svcid } (\text{tm-store } (value, \text{addr}, tm)),
\end{aligned}$$

tm-base (tm-store (*value*, *addr*, *tm*)),
tm-limit (tm-store (*value*, *addr*, *tm*)))

EVENT: Disable tm-store-commutes-with-mapup-address-space.

THEOREM: tm-set-pc-commutes-with-mapup-address-space

tm-set-pc (*pc*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-set-pc (*pc*, *tm*)),
tm-regs (tm-set-pc (*pc*, *tm*)),
tm-cc (tm-set-pc (*pc*, *tm*)),
tm-error (tm-set-pc (*pc*, *tm*)),
tm-svcflag (tm-set-pc (*pc*, *tm*)),
tm-svcid (tm-set-pc (*pc*, *tm*)),
tm-base (tm-set-pc (*pc*, *tm*)),
tm-limit (tm-set-pc (*pc*, *tm*)))

EVENT: Disable tm-set-pc-commutes-with-mapup-address-space.

THEOREM: tm-set-sp-commutes-with-mapup-address-space

tm-set-sp (*sp*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-set-sp (*sp*, *tm*)),
tm-regs (tm-set-sp (*sp*, *tm*)),
tm-cc (tm-set-sp (*sp*, *tm*)),
tm-error (tm-set-sp (*sp*, *tm*)),
tm-svcflag (tm-set-sp (*sp*, *tm*)),
tm-svcid (tm-set-sp (*sp*, *tm*)),
tm-base (tm-set-sp (*sp*, *tm*)),
tm-limit (tm-set-sp (*sp*, *tm*)))

EVENT: Disable tm-set-sp-commutes-with-mapup-address-space.

THEOREM: tm-set-cc-commutes-with-mapup-address-space

$$\begin{aligned} & \text{tm-set-cc}(cc, \\ & \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\ & \quad \quad \text{tm-regs}(tm), \\ & \quad \quad \text{tm-cc}(tm), \\ & \quad \quad \text{tm-error}(tm), \\ & \quad \quad \text{tm-svcflag}(tm), \\ & \quad \quad \text{tm-svcid}(tm), \\ & \quad \quad \text{tm-base}(tm), \\ & \quad \quad \text{tm-limit}(tm))) \\ = & \quad \text{mapup-address-space}(\text{tm-memory}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-regs}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-cc}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-error}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-svcflag}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-svcid}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-base}(\text{tm-set-cc}(cc, tm)), \\ & \quad \text{tm-limit}(\text{tm-set-cc}(cc, tm))) \end{aligned}$$

EVENT: Disable tm-set-cc-commutes-with-mapup-address-space.

THEOREM: tm-set-error-commutes-with-mapup-address-space

$$\begin{aligned} & \text{tm-set-error}(error, \\ & \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\ & \quad \quad \text{tm-regs}(tm), \\ & \quad \quad \text{tm-cc}(tm), \\ & \quad \quad \text{tm-error}(tm), \\ & \quad \quad \text{tm-svcflag}(tm), \\ & \quad \quad \text{tm-svcid}(tm), \\ & \quad \quad \text{tm-base}(tm), \\ & \quad \quad \text{tm-limit}(tm))) \\ = & \quad \text{mapup-address-space}(\text{tm-memory}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-regs}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-cc}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-error}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-svcflag}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-svcid}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-base}(\text{tm-set-error}(error, tm)), \\ & \quad \text{tm-limit}(\text{tm-set-error}(error, tm))) \end{aligned}$$

EVENT: Disable tm-set-error-commutes-with-mapup-address-space.

THEOREM: tm-set-svcflag-commutes-with-mapup-address-space

$$\begin{aligned}
& \text{tm-set-svcflag} (svcflag, \\
& \quad \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \quad \quad \text{tm-regs} (tm), \\
& \quad \quad \text{tm-cc} (tm), \\
& \quad \quad \text{tm-error} (tm), \\
& \quad \quad \text{tm-svcflag} (tm), \\
& \quad \quad \text{tm-svcid} (tm), \\
& \quad \quad \text{tm-base} (tm), \\
& \quad \quad \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-regs} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-cc} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-error} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-svcflag} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-svcid} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-base} (\text{tm-set-svcflag} (svcflag, tm)), \\
& \quad \text{tm-limit} (\text{tm-set-svcflag} (svcflag, tm)))
\end{aligned}$$

EVENT: Disable tm-set-svcflag-commutes-with-mapup-address-space.

THEOREM: tm-set-svcid-commutes-with-mapup-address-space

$$\begin{aligned}
& \text{tm-set-svcid} (svcid, \\
& \quad \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \quad \quad \text{tm-regs} (tm), \\
& \quad \quad \text{tm-cc} (tm), \\
& \quad \quad \text{tm-error} (tm), \\
& \quad \quad \text{tm-svcflag} (tm), \\
& \quad \quad \text{tm-svcid} (tm), \\
& \quad \quad \text{tm-base} (tm), \\
& \quad \quad \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-regs} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-cc} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-error} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-svcflag} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-svcid} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-base} (\text{tm-set-svcid} (svcid, tm)), \\
& \quad \text{tm-limit} (\text{tm-set-svcid} (svcid, tm)))
\end{aligned}$$

EVENT: Disable tm-set-svcid-commutes-with-mapup-address-space.

THEOREM: tm-in-supervisor-mode-tm-set-cc

$$\text{tm-in-supervisor-mode} (\text{tm-set-cc} (cc, tm)) = \text{tm-in-supervisor-mode} (tm)$$

EVENT: Disable tm-in-supervisor-mode-tm-set-cc.

THEOREM: tm-in-supervisor-mode-tm-set-svcflag
 $\text{tm-in-supervisor-mode}(\text{tm-set-svcflag}(svcflag, tm))$
 $= \text{tm-in-supervisor-mode}(tm)$

EVENT: Disable tm-in-supervisor-mode-tm-set-svcflag.

THEOREM: tm-in-supervisor-mode-tm-set-svcid
 $\text{tm-in-supervisor-mode}(\text{tm-set-svcid}(svcid, tm)) = \text{tm-in-supervisor-mode}(tm)$

EVENT: Disable tm-in-supervisor-mode-tm-set-svcid.

THEOREM: tm-in-supervisor-mode-tm-set-sp
 $\text{tm-in-supervisor-mode}(\text{tm-set-sp}(sp, tm)) = \text{tm-in-supervisor-mode}(tm)$

EVENT: Disable tm-in-supervisor-mode-tm-set-sp.

THEOREM: tm-in-supervisor-mode-tm-set-pc
 $\text{tm-in-supervisor-mode}(\text{tm-set-pc}(pc, tm)) = \text{tm-in-supervisor-mode}(tm)$

EVENT: Disable tm-in-supervisor-mode-tm-set-pc.

THEOREM: tm-limit-mapup-address-space
 $\text{tm-limit}(\text{mapup-address-space}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$
 $\text{tm-svcid}(tm),$
 $\text{tm-base}(tm),$
 $\text{tm-limit}(tm)))$
 $= \text{fix}(\text{tm-limit}(tm))$

EVENT: Disable tm-limit-mapup-address-space.

THEOREM: tm-base-mapup-address-space
 $\text{tm-base}(\text{mapup-address-space}(\text{tm-memory}(tm),$
 $\text{tm-regs}(tm),$
 $\text{tm-cc}(tm),$
 $\text{tm-error}(tm),$
 $\text{tm-svcflag}(tm),$

$$\begin{aligned}
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{'0}
\end{aligned}$$

EVENT: Disable tm-base-mapup-address-space.

$$\begin{aligned}
\text{THEOREM: plus-tm-base-tm-limit-tm-set-cc} \\
& (\text{tm-base}(\text{tm-set-cc}(cc, tm)) + \text{tm-limit}(\text{tm-set-cc}(cc, tm))) \\
= & (\text{tm-base}(tm) + \text{tm-limit}(tm))
\end{aligned}$$

EVENT: Disable plus-tm-base-tm-limit-tm-set-cc.

$$\begin{aligned}
\text{THEOREM: plus-tm-base-tm-limit-tm-set-sp} \\
& (\text{tm-base}(\text{tm-set-sp}(sp, tm)) + \text{tm-limit}(\text{tm-set-sp}(sp, tm))) \\
= & (\text{tm-base}(tm) + \text{tm-limit}(tm))
\end{aligned}$$

EVENT: Disable plus-tm-base-tm-limit-tm-set-sp.

$$\begin{aligned}
\text{THEOREM: tm-execute-add-commutes-with-mapup-address-space} \\
& (\text{good-tm}(tm) \\
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \wedge (\text{if } \text{'65536} < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \\
& \quad \text{then } \text{'*1*false} \\
& \quad \text{else } \text{'*1*true endif} \\
& \wedge (\text{tm-good-address}(addr1, tm) \\
& \quad \wedge \text{tm-good-address}(addr2, tm)))))) \\
\rightarrow & (\text{tm-execute-add}(addr1, \\
& \quad addr2, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \quad \text{tm-regs}(tm), \\
& \quad \quad \text{tm-cc}(tm), \\
& \quad \quad \text{tm-error}(tm), \\
& \quad \quad \text{tm-svcflag}(tm), \\
& \quad \quad \text{tm-svcid}(tm), \\
& \quad \quad \text{tm-base}(tm), \\
& \quad \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-add}(addr1, addr2, tm)), \\
& \quad \text{tm-regs}(\text{tm-execute-add}(addr1, addr2, tm)), \\
& \quad \text{tm-cc}(\text{tm-execute-add}(addr1, addr2, tm)), \\
& \quad \text{tm-error}(\text{tm-execute-add}(addr1, addr2, tm)), \\
& \quad \text{tm-svcflag}(\text{tm-execute-add}(addr1, addr2, tm)), \\
& \quad \text{tm-svcid}(\text{tm-execute-add}(addr1, addr2, tm))),
\end{aligned}$$

$$\begin{aligned} & \text{tm-base}(\text{tm-execute-add}(\text{addr1}, \text{addr2}, \text{tm})), \\ & \text{tm-limit}(\text{tm-execute-add}(\text{addr1}, \text{addr2}, \text{tm}))) \end{aligned}$$

EVENT: Disable tm-execute-add-commutes-with-mapup-address-space.

THEOREM: tm-execute-branch-commutes-with-mapup-address-space

$$\begin{aligned} & \text{good-tm}(\text{tm}) \\ & \wedge ((\neg \text{tm-in-supervisor-mode}(\text{tm})) \wedge \text{tm-good-address}(\text{addr}, \text{tm})) \\ & \rightarrow (\text{tm-execute-branch}(\text{addr}, \\ & \qquad \qquad \qquad \text{mapup-address-space}(\text{tm-memory}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-regs}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-cc}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-error}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-svcflag}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-svcid}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-base}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-limit}(\text{tm}))) \\ & = \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-regs}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-cc}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-error}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-svcflag}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-svcid}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-base}(\text{tm-execute-branch}(\text{addr}, \text{tm})), \\ & \qquad \text{tm-limit}(\text{tm-execute-branch}(\text{addr}, \text{tm}))) \end{aligned}$$

EVENT: Disable tm-execute-branch-commutes-with-mapup-address-space.

THEOREM: tm-cc-mapup-address-space

$$\begin{aligned} & \text{good-tm}(\text{tm}) \\ & \rightarrow (\text{tm-cc}(\text{mapup-address-space}(\text{tm-memory}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-regs}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-cc}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-error}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-svcflag}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-svcid}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-base}(\text{tm}), \\ & \qquad \qquad \qquad \text{tm-limit}(\text{tm}))) \\ & = \text{tm-cc}(\text{tm}) \end{aligned}$$

EVENT: Disable tm-cc-mapup-address-space.

THEOREM: tm-error-mapup-address-space

$$\begin{aligned}
& \text{good-tm}(tm) \\
& \rightarrow (\text{tm-error}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \qquad \qquad \text{tm-regs}(tm), \\
& \qquad \qquad \qquad \text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm), \\
& \qquad \qquad \qquad \text{tm-base}(tm), \\
& \qquad \qquad \qquad \text{tm-limit}(tm))) \\
& = \text{tm-error}(tm)
\end{aligned}$$

EVENT: Disable tm-error-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: tm-svcflag-mapup-address-space} \\
& \text{good-tm}(tm) \\
& \rightarrow (\text{tm-svcflag}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \qquad \qquad \text{tm-regs}(tm), \\
& \qquad \qquad \qquad \text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm), \\
& \qquad \qquad \qquad \text{tm-base}(tm), \\
& \qquad \qquad \qquad \text{tm-limit}(tm))) \\
& = \text{tm-svcflag}(tm)
\end{aligned}$$

EVENT: Disable tm-svcflag-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: tm-svcid-mapup-address-space} \\
& \text{good-tm}(tm) \\
& \rightarrow (\text{tm-svcid}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \qquad \qquad \text{tm-regs}(tm), \\
& \qquad \qquad \qquad \text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm), \\
& \qquad \qquad \qquad \text{tm-base}(tm), \\
& \qquad \qquad \qquad \text{tm-limit}(tm))) \\
& = \text{tm-svcid}(tm)
\end{aligned}$$

EVENT: Disable tm-svcid-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: tm-pc-mapup-address-space} \\
& \text{good-tm}(tm)
\end{aligned}$$

$$\begin{aligned}
&\rightarrow (\text{tm-pc}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
&\quad \text{tm-regs}(tm), \\
&\quad \text{tm-cc}(tm), \\
&\quad \text{tm-error}(tm), \\
&\quad \text{tm-svcflag}(tm), \\
&\quad \text{tm-svcid}(tm), \\
&\quad \text{tm-base}(tm), \\
&\quad \text{tm-limit}(tm))) \\
&= \text{tm-pc}(tm)
\end{aligned}$$

EVENT: Disable tm-pc-mapup-address-space.

THEOREM: tm-sp-mapup-address-space

$$\begin{aligned}
&\text{good-tm}(tm) \\
&\rightarrow (\text{tm-sp}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
&\quad \text{tm-regs}(tm), \\
&\quad \text{tm-cc}(tm), \\
&\quad \text{tm-error}(tm), \\
&\quad \text{tm-svcflag}(tm), \\
&\quad \text{tm-svcid}(tm), \\
&\quad \text{tm-base}(tm), \\
&\quad \text{tm-limit}(tm))) \\
&= \text{tm-sp}(tm)
\end{aligned}$$

EVENT: Disable tm-sp-mapup-address-space.

THEOREM: tm-execute-branch-on-zero-commutes-with-mapup-address-space

$$\begin{aligned}
&(\text{good-tm}(tm) \\
&\wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{tm-good-address}(addr, tm))) \\
&\rightarrow (\text{tm-execute-branch-on-zero}(addr, \\
&\quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
&\quad \text{tm-regs}(tm), \\
&\quad \text{tm-cc}(tm), \\
&\quad \text{tm-error}(tm), \\
&\quad \text{tm-svcflag}(tm), \\
&\quad \text{tm-svcid}(tm), \\
&\quad \text{tm-base}(tm), \\
&\quad \text{tm-limit}(tm))) \\
&= \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-branch-on-zero}(addr, \\
&\quad \text{tm}), \\
&\quad \text{tm-regs}(\text{tm-execute-branch-on-zero}(addr, tm)), \\
&\quad \text{tm-cc}(\text{tm-execute-branch-on-zero}(addr, tm)), \\
&\quad \text{tm-error}(\text{tm-execute-branch-on-zero}(addr,
\end{aligned}$$

$$\begin{aligned}
& \text{tm-svcflag}(\text{tm-execute-branch-on-zero}(addr, tm)), \\
& \text{tm-svcid}(\text{tm-execute-branch-on-zero}(addr, tm)), \\
& \text{tm-base}(\text{tm-execute-branch-on-zero}(addr, tm)), \\
& \text{tm-limit}(\text{tm-execute-branch-on-zero}(addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-branch-on-zero-commutes-with-mapup-address-space.

THEOREM: tm-execute-branch-not-zero-commutes-with-mapup-address-space
(good-tm (tm))

$$\begin{aligned}
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \wedge \text{tm-good-address}(addr, tm)) \\
& \rightarrow (\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \qquad \text{tm-regs}(tm), \\
& \qquad \text{tm-cc}(tm), \\
& \qquad \text{tm-error}(tm), \\
& \qquad \text{tm-svcflag}(tm), \\
& \qquad \text{tm-svcid}(tm), \\
& \qquad \text{tm-base}(tm), \\
& \qquad \text{tm-limit}(tm))) \\
& = \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-regs}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-cc}(\text{tm-execute-branch-not-zero}(addr, tm)), \\
& \text{tm-error}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-svcflag}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-svcid}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-base}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm-limit}(\text{tm-execute-branch-not-zero}(addr, \\
& \qquad \text{tm}))))))
\end{aligned}$$

EVENT: Disable tm-execute-branch-not-zero-commutes-with-mapup-address-space.

THEOREM: tm-stack-overflowp-mapup-address-space
good-tm (tm)

\rightarrow (tm-stack-overflowp (mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= tm-stack-overflowp (*tm*))

EVENT: Disable tm-stack-overflowp-mapup-address-space.

THEOREM: tm-good-address-mapup-address-space
(good-tm (*tm*)
 \wedge (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
 \wedge (\neg tm-in-supervisor-mode (*tm*))))
 \rightarrow (tm-good-address (*addr*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= tm-good-address (*addr*, *tm*))

EVENT: Disable tm-good-address-mapup-address-space.

THEOREM: tm-execute-call-commutes-with-mapup-address-space
(good-tm (*tm*)
 \wedge (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
 \wedge ((\neg tm-in-supervisor-mode (*tm*)
 \wedge tm-good-address (*addr*, *tm*))))
 \rightarrow (tm-execute-call (*addr*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),

$$\begin{aligned}
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm))) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-regs } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-cc } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-error } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-svcflag } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-svcid } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-base } (\text{tm-execute-call } (addr, tm)), \\
& \text{tm-limit } (\text{tm-execute-call } (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-call-commutes-with-mapup-address-space.

$$\begin{aligned}
& \text{THEOREM: tm-execute-compare-commutes-with-mapup-address-space} \\
& (\text{good-tm } (tm) \\
& \wedge ((\neg \text{tm-in-supervisor-mode } (tm)) \\
& \wedge (\text{tm-good-address } (addr1, tm) \wedge \text{tm-good-address } (addr2, tm)))) \\
\rightarrow & (\text{tm-execute-compare } (addr1, \\
& \text{addr2,} \\
& \text{mapup-address-space } (\text{tm-memory } (tm), \\
& \text{tm-regs } (tm), \\
& \text{tm-cc } (tm), \\
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm))) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-compare } (addr1, \\
& \text{addr2,} \\
& \text{tm)), \\
& \text{tm-regs } (\text{tm-execute-compare } (addr1, addr2, tm)), \\
& \text{tm-cc } (\text{tm-execute-compare } (addr1, addr2, tm)), \\
& \text{tm-error } (\text{tm-execute-compare } (addr1, \\
& \text{addr2,} \\
& \text{tm)),} \\
& \text{tm-svcflag } (\text{tm-execute-compare } (addr1, \\
& \text{addr2,} \\
& \text{tm)),} \\
& \text{tm-svcid } (\text{tm-execute-compare } (addr1, \\
& \text{addr2,} \\
& \text{tm)),} \\
& \text{tm-base } (\text{tm-execute-compare } (addr1, addr2, tm)), \\
& \text{tm-limit } (\text{tm-execute-compare } (addr1, \\
& \text{addr2,}
\end{aligned}$$

$tm))))$

EVENT: Disable tm-execute-compare-commutes-with-mapup-address-space.

THEOREM: tm-execute-decr-commutes-with-mapup-address-space
(good-tm (tm)
 \wedge (**if** '65536 < (tm-base (tm) + tm-limit (tm)) **then** '*1*false
 else '*1*true **endif**
 \wedge ((\neg tm-in-supervisor-mode (tm))
 \wedge tm-good-address ($addr$, tm))))
→ (tm-execute-decr ($addr$,
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),
 tm-svcflag (tm),
 tm-svcid (tm),
 tm-base (tm),
 tm-limit (tm)))
 = mapup-address-space (tm-memory (tm-execute-decr ($addr$, tm)),
 tm-regs (tm-execute-decr ($addr$, tm)),
 tm-cc (tm-execute-decr ($addr$, tm)),
 tm-error (tm-execute-decr ($addr$, tm)),
 tm-svcflag (tm-execute-decr ($addr$, tm)),
 tm-svcid (tm-execute-decr ($addr$, tm)),
 tm-base (tm-execute-decr ($addr$, tm)),
 tm-limit (tm-execute-decr ($addr$, tm))))

EVENT: Disable tm-execute-decr-commutes-with-mapup-address-space.

THEOREM: tm-execute-decr-mod-commutes-with-mapup-address-space
(good-tm (tm)
 \wedge (**if** '65536 < (tm-base (tm) + tm-limit (tm)) **then** '*1*false
 else '*1*true **endif**
 \wedge ((\neg tm-in-supervisor-mode (tm))
 \wedge (tm-good-address ($addr1$, tm)
 \wedge tm-good-address ($addr2$, tm))))
→ (tm-execute-decr-mod ($addr1$,
 $addr2$,
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),

$$\begin{aligned}
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-regs}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-cc}(\text{tm-execute-decr-mod}(addr1, addr2, tm)), \\
& \text{tm-error}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-svcflag}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-svcid}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-base}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-limit}(\text{tm-execute-decr-mod}(addr1, \\
& \text{addr2}, \\
& tm))))
\end{aligned}$$

EVENT: Disable tm-execute-decr-mod-commutes-with-mapup-address-space.

THEOREM: tm-execute-incr-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \text{ else } '*1*true \text{ endif} \\
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \wedge \text{tm-good-address}(addr, tm)))) \\
\rightarrow & (\text{tm-execute-incr}(addr, \\
& \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \text{tm-regs}(tm), \\
& \text{tm-cc}(tm), \\
& \text{tm-error}(tm), \\
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)))
\end{aligned}$$

$$\begin{aligned}
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-regs} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-cc} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-error} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-svcflag} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-svcid} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-base} (\text{tm-execute-incr} (addr, tm)), \\
& \text{tm-limit} (\text{tm-execute-incr} (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-incr-commutes-with-mapup-address-space.

THEOREM: tm-execute-incr-mod-commutes-with-mapup-address-space
(good-tm (tm))

$$\begin{aligned}
& \wedge (\mathbf{if} \ '65536 < (\text{tm-base} (tm) + \text{tm-limit} (tm)) \ \mathbf{then} \ '1*\mathbf{false} \\
& \quad \mathbf{else} \ '1*\mathbf{true} \ \mathbf{endif}) \\
& \wedge ((\neg \text{tm-in-supervisor-mode} (tm)) \\
& \quad \wedge (\text{tm-good-address} (addr1, tm) \\
& \quad \quad \wedge \text{tm-good-address} (addr2, tm)))) \\
\rightarrow & (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \quad \quad \quad \quad \text{tm-regs} (tm), \\
& \quad \quad \quad \quad \text{tm-cc} (tm), \\
& \quad \quad \quad \quad \text{tm-error} (tm), \\
& \quad \quad \quad \quad \text{tm-svcflag} (tm), \\
& \quad \quad \quad \quad \text{tm-svcid} (tm), \\
& \quad \quad \quad \quad \text{tm-base} (tm), \\
& \quad \quad \quad \quad \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-regs} (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-cc} (\text{tm-execute-incr-mod} (addr1, addr2, tm)), \\
& \quad \text{tm-error} (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-svcflag} (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-svcid} (\text{tm-execute-incr-mod} (addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)),
\end{aligned}$$

$$\begin{aligned} & \text{tm-base (tm-execute-incr-mod (addr1,} \\ & \hspace{10em} \text{addr2,} \\ & \hspace{10em} \text{tm)),} \\ & \text{tm-limit (tm-execute-incr-mod (addr1,} \\ & \hspace{10em} \text{addr2,} \\ & \hspace{10em} \text{tm))))} \end{aligned}$$

EVENT: Disable tm-execute-incr-mod-commutes-with-mapup-address-space.

THEOREM: tm-in-supervisor-mode-mapup-address-space
tm-in-supervisor-mode (mapup-address-space (*memory*,
regs,
cc,
error,
svcflag,
svcid,
base,
limit))

= '1*false

EVENT: Disable tm-in-supervisor-mode-mapup-address-space.

THEOREM: tm-execute-load-base-commutes-with-mapup-address-space
(good-tm (*tm*) \wedge (\neg tm-in-supervisor-mode (*tm*)))
 \rightarrow (tm-execute-load-base (*addr*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))

= mapup-address-space (tm-memory (tm-execute-load-base (*addr*, *tm*)),
tm-regs (tm-execute-load-base (*addr*, *tm*)),
tm-cc (tm-execute-load-base (*addr*, *tm*)),
tm-error (tm-execute-load-base (*addr*, *tm*)),
tm-svcflag (tm-execute-load-base (*addr*, *tm*)),
tm-svcid (tm-execute-load-base (*addr*, *tm*)),
tm-base (tm-execute-load-base (*addr*, *tm*)),
tm-limit (tm-execute-load-base (*addr*, *tm*))))

EVENT: Disable tm-execute-load-base-commutes-with-mapup-address-space.

THEOREM: tm-execute-load-limit-commutes-with-mapup-address-space
 (good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))
 \rightarrow (tm-execute-load-limit ($addr$,
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),
 tm-svcflag (tm),
 tm-svcid (tm),
 tm-base (tm),
 tm-limit (tm)))
 = mapup-address-space (tm-memory (tm-execute-load-limit ($addr$, tm)),
 tm-regs (tm-execute-load-limit ($addr$, tm)),
 tm-cc (tm-execute-load-limit ($addr$, tm)),
 tm-error (tm-execute-load-limit ($addr$, tm)),
 tm-svcflag (tm-execute-load-limit ($addr$, tm)),
 tm-svcid (tm-execute-load-limit ($addr$, tm)),
 tm-base (tm-execute-load-limit ($addr$, tm)),
 tm-limit (tm-execute-load-limit ($addr$, tm))))

EVENT: Disable tm-execute-load-limit-commutes-with-mapup-address-space.

THEOREM: tm-execute-load-psw-commutes-with-mapup-address-space
 (good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))
 \rightarrow (tm-execute-load-psw ($addr$,
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),
 tm-svcflag (tm),
 tm-svcid (tm),
 tm-base (tm),
 tm-limit (tm)))
 = mapup-address-space (tm-memory (tm-execute-load-psw ($addr$, tm)),
 tm-regs (tm-execute-load-psw ($addr$, tm)),
 tm-cc (tm-execute-load-psw ($addr$, tm)),
 tm-error (tm-execute-load-psw ($addr$, tm)),
 tm-svcflag (tm-execute-load-psw ($addr$, tm)),
 tm-svcid (tm-execute-load-psw ($addr$, tm)),
 tm-base (tm-execute-load-psw ($addr$, tm)),
 tm-limit (tm-execute-load-psw ($addr$, tm))))

EVENT: Disable tm-execute-load-psw-commutes-with-mapup-address-space.

$$\begin{aligned}
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-regs}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-cc}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-error}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-svcid}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-base}(\text{tm-execute-move}(addr1, addr2, tm)), \\
& \text{tm-limit}(\text{tm-execute-move}(addr1, addr2, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-move-commutes-with-mapup-address-space.

THEOREM: tm-execute-multiply-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \\
& \quad \text{then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \wedge (\text{tm-good-address}(addr1, tm) \\
& \quad \wedge \text{tm-good-address}(addr2, tm)))))) \\
\rightarrow & (\text{tm-execute-multiply}(addr1, \\
& \quad addr2, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \quad \text{tm-regs}(tm), \\
& \quad \quad \text{tm-cc}(tm), \\
& \quad \quad \text{tm-error}(tm), \\
& \quad \quad \text{tm-svcflag}(tm), \\
& \quad \quad \text{tm-svcid}(tm), \\
& \quad \quad \text{tm-base}(tm), \\
& \quad \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-multiply}(addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \text{tm-regs}(\text{tm-execute-multiply}(addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \text{tm-cc}(\text{tm-execute-multiply}(addr1, addr2, tm)), \\
& \text{tm-error}(\text{tm-execute-multiply}(addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \text{tm-svcflag}(\text{tm-execute-multiply}(addr1, \\
& \quad \quad \quad addr2,
\end{aligned}$$

$$\begin{aligned}
& \text{tm-svcid}(\text{tm-execute-multiply}(addr1, \\
& \quad \text{tm}), \\
& \quad \text{addr2}, \\
& \quad \text{tm}), \\
& \text{tm-base}(\text{tm-execute-multiply}(addr1, \\
& \quad \text{tm}), \\
& \quad \text{addr2}, \\
& \quad \text{tm}), \\
& \text{tm-limit}(\text{tm-execute-multiply}(addr1, \\
& \quad \text{tm}), \\
& \quad \text{addr2}, \\
& \quad \text{tm})))
\end{aligned}$$

EVENT: Disable tm-execute-multiply-commutes-with-mapup-address-space.

THEOREM: tm-stack-underflowp-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))) \\
& \rightarrow (\text{tm-stack-underflowp}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \text{tm-regs}(tm), \\
& \quad \text{tm-cc}(tm), \\
& \quad \text{tm-error}(tm), \\
& \quad \text{tm-svcflag}(tm), \\
& \quad \text{tm-svcid}(tm), \\
& \quad \text{tm-base}(tm), \\
& \quad \text{tm-limit}(tm))) \\
& = \text{tm-stack-underflowp}(tm))
\end{aligned}$$

EVENT: Disable tm-stack-underflowp-mapup-address-space.

THEOREM: tm-execute-return-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))) \\
& \rightarrow (\text{tm-execute-return}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \text{tm-regs}(tm), \\
& \quad \text{tm-cc}(tm), \\
& \quad \text{tm-error}(tm), \\
& \quad \text{tm-svcflag}(tm), \\
& \quad \text{tm-svcid}(tm), \\
& \quad \text{tm-base}(tm), \\
& \quad \text{tm-limit}(tm)))
\end{aligned}$$

$$\begin{aligned}
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-return} (tm)), \\
& \text{tm-regs} (\text{tm-execute-return} (tm)), \\
& \text{tm-cc} (\text{tm-execute-return} (tm)), \\
& \text{tm-error} (\text{tm-execute-return} (tm)), \\
& \text{tm-svcflag} (\text{tm-execute-return} (tm)), \\
& \text{tm-svcid} (\text{tm-execute-return} (tm)), \\
& \text{tm-base} (\text{tm-execute-return} (tm)), \\
& \text{tm-limit} (\text{tm-execute-return} (tm)))
\end{aligned}$$

EVENT: Disable tm-execute-return-commutes-with-mapup-address-space.

THEOREM: tm-execute-set-clock-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-set-clock} (addr, \\
& \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm), \\
& \text{tm-base} (tm), \\
& \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-regs} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-cc} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-error} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-svcflag} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-svcid} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-base} (\text{tm-execute-set-clock} (addr, tm)), \\
& \text{tm-limit} (\text{tm-execute-set-clock} (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-set-clock-commutes-with-mapup-address-space.

THEOREM: tm-execute-run-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-run} (\text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm), \\
& \text{tm-base} (tm), \\
& \text{tm-limit} (tm)))
\end{aligned}$$

$$\begin{aligned}
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-run} (tm)), \\
& \text{tm-regs} (\text{tm-execute-run} (tm)), \\
& \text{tm-cc} (\text{tm-execute-run} (tm)), \\
& \text{tm-error} (\text{tm-execute-run} (tm)), \\
& \text{tm-svcflag} (\text{tm-execute-run} (tm)), \\
& \text{tm-svcid} (\text{tm-execute-run} (tm)), \\
& \text{tm-base} (\text{tm-execute-run} (tm)), \\
& \text{tm-limit} (\text{tm-execute-run} (tm)))
\end{aligned}$$

EVENT: Disable tm-execute-run-commutes-with-mapup-address-space.

THEOREM: tm-execute-wait-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-wait} (\text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm), \\
& \text{tm-base} (tm), \\
& \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-wait} (tm)), \\
& \text{tm-regs} (\text{tm-execute-wait} (tm)), \\
& \text{tm-cc} (\text{tm-execute-wait} (tm)), \\
& \text{tm-error} (\text{tm-execute-wait} (tm)), \\
& \text{tm-svcflag} (\text{tm-execute-wait} (tm)), \\
& \text{tm-svcid} (\text{tm-execute-wait} (tm)), \\
& \text{tm-base} (\text{tm-execute-wait} (tm)), \\
& \text{tm-limit} (\text{tm-execute-wait} (tm)))
\end{aligned}$$

EVENT: Disable tm-execute-wait-commutes-with-mapup-address-space.

THEOREM: tm-execute-test-iptort-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-test-iptort} (addr, \\
& \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm), \\
& \text{tm-base} (tm), \\
& \text{tm-limit} (tm)))
\end{aligned}$$

$$\begin{aligned}
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-regs} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-cc} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-error} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-svcflag} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-svcid} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-base} (\text{tm-execute-test-iport} (addr, tm)), \\
& \text{tm-limit} (\text{tm-execute-test-iport} (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-test-iport-commutes-with-mapup-address-space.

THEOREM: tm-execute-test-oport-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-test-oport} (addr, \\
& \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm), \\
& \text{tm-base} (tm), \\
& \text{tm-limit} (tm))) \\
= & \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-regs} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-cc} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-error} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-svcflag} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-svcid} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-base} (\text{tm-execute-test-oport} (addr, tm)), \\
& \text{tm-limit} (\text{tm-execute-test-oport} (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-test-oport-commutes-with-mapup-address-space.

THEOREM: tm-execute-start-output-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-start-output} (addr1, \\
& addr2, \\
& \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \text{tm-regs} (tm), \\
& \text{tm-cc} (tm), \\
& \text{tm-error} (tm), \\
& \text{tm-svcflag} (tm), \\
& \text{tm-svcid} (tm),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm))) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-regs } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-cc } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-error } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-svcflag } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-svcid } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-base } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm})), \\
& \text{tm-limit } (\text{tm-execute-start-output } (addr1, \\
& \text{addr2,} \\
& \text{tm}))))
\end{aligned}$$

EVENT: Disable tm-execute-start-output-commutes-with-mapup-address-space.

THEOREM: tm-execute-post-output-interrupt-commutes-with-mapup-address-space
(good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))

$$\begin{aligned}
\rightarrow & (\text{tm-execute-post-output-interrupt } (addr, \\
& \text{mapup-address-space } (\text{tm-memory } (tm), \\
& \text{tm-regs } (tm), \\
& \text{tm-cc } (tm), \\
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm))) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-post-output-interrupt } (addr, \\
& \text{tm})), \\
& \text{tm-regs } (\text{tm-execute-post-output-interrupt } (addr, \\
& \text{tm})),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-cc} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})), \\
& \text{tm-error} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})), \\
& \text{tm-svcflag} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})), \\
& \text{tm-svcid} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})), \\
& \text{tm-base} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})), \\
& \text{tm-limit} (\text{tm-execute-post-output-interrupt} (addr, \\
& \quad \quad \quad \text{tm})))
\end{aligned}$$

EVENT: Disable tm-execute-post-output-interrupt-commutes-with-mapup-address-space.

THEOREM: tm-execute-svc-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm} (tm)) \\
& \wedge ((\neg \text{tm-in-supervisor-mode} (tm)) \wedge \text{tm-good-address} (addr, tm)) \\
& \rightarrow (\text{tm-execute-svc} (addr, \\
& \quad \quad \quad \text{mapup-address-space} (\text{tm-memory} (tm), \\
& \quad \quad \quad \quad \text{tm-regs} (tm), \\
& \quad \quad \quad \quad \text{tm-cc} (tm), \\
& \quad \quad \quad \quad \text{tm-error} (tm), \\
& \quad \quad \quad \quad \text{tm-svcflag} (tm), \\
& \quad \quad \quad \quad \text{tm-svcid} (tm), \\
& \quad \quad \quad \quad \text{tm-base} (tm), \\
& \quad \quad \quad \quad \text{tm-limit} (tm))) \\
& = \text{mapup-address-space} (\text{tm-memory} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-regs} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-cc} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-error} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-svcflag} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-svcid} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-base} (\text{tm-execute-svc} (addr, tm)), \\
& \quad \quad \quad \text{tm-limit} (\text{tm-execute-svc} (addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-svc-commutes-with-mapup-address-space.

THEOREM: tm-execute-svc-return-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm} (tm) \wedge (\neg \text{tm-in-supervisor-mode} (tm))) \\
& \rightarrow (\text{tm-execute-svc-return} (addr, \\
& \quad \quad \quad \text{mapup-address-space} (\text{tm-memory} (tm),
\end{aligned}$$

```

tm-regs (tm),
tm-cc (tm),
tm-error (tm),
tm-svcflag (tm),
tm-svcid (tm),
tm-base (tm),
tm-limit (tm)))
= mapup-address-space (tm-memory (tm-execute-svc-return (addr, tm)),
tm-regs (tm-execute-svc-return (addr, tm)),
tm-cc (tm-execute-svc-return (addr, tm)),
tm-error (tm-execute-svc-return (addr, tm)),
tm-svcflag (tm-execute-svc-return (addr, tm)),
tm-svcid (tm-execute-svc-return (addr, tm)),
tm-base (tm-execute-svc-return (addr, tm)),
tm-limit (tm-execute-svc-return (addr, tm))))

```

EVENT: Disable tm-execute-svc-return-commutes-with-mapup-address-space.

DEFINITION:

```

tm-execute-nullary2 (opcode, tm)
= if opcode = '8 then tm-execute-run (tm)
  elseif opcode = '4 then tm-execute-wait (tm)
  else tm-set-error ('2, tm) endif

```

DEFINITION:

```

tm-execute-nullary1 (opcode, tm)
= if opcode = '0 then tm-execute-return (tm)
  else tm-execute-nullary2 (opcode, tm) endif

```

THEOREM: tm-execute-nullary1-equals-tm-execute-nullary

```

tm-execute-nullary1 (opcode, tm) = tm-execute-nullary (opcode, tm)

```

EVENT: Disable tm-execute-nullary1-equals-tm-execute-nullary.

THEOREM: tm-execute-nullary2-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (if '65536 < (tm-base (tm) + tm-limit (tm)) then '*1*false
    else '*1*true endif
    ∧ (¬ tm-in-supervisor-mode (tm))))
→ (tm-execute-nullary2 (opcode,
  mapup-address-space (tm-memory (tm),
    tm-regs (tm),
    tm-cc (tm),
    tm-error (tm),

```

$$\begin{aligned}
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-regs}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-cc}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-error}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-svcid}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-base}(\text{tm-execute-nullary2}(opcode, tm)), \\
& \text{tm-limit}(\text{tm-execute-nullary2}(opcode, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-nullary2-commutes-with-mapup-address-space.

THEOREM: tm-execute-nullary1-commutes-with-mapup-address-space
(good-tm(tm))

$$\begin{aligned}
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm))) \\
\rightarrow & (\text{tm-execute-nullary1}(opcode, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \text{tm-regs}(tm), \\
& \quad \text{tm-cc}(tm), \\
& \quad \text{tm-error}(tm), \\
& \quad \text{tm-svcflag}(tm), \\
& \quad \text{tm-svcid}(tm), \\
& \quad \text{tm-base}(tm), \\
& \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-regs}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-cc}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-error}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-svcid}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-base}(\text{tm-execute-nullary1}(opcode, tm)), \\
& \text{tm-limit}(\text{tm-execute-nullary1}(opcode, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-nullary1-commutes-with-mapup-address-space.

THEOREM: tm-execute-nullary-commutes-with-mapup-address-space
(good-tm(tm))

$$\wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false$$

```

    else '*1*true endif
    ∧ (¬ tm-in-supervisor-mode (tm)))
→ (tm-execute-nullary (opcode,
    mapup-address-space (tm-memory (tm),
    tm-regs (tm),
    tm-cc (tm),
    tm-error (tm),
    tm-svcflag (tm),
    tm-svcid (tm),
    tm-base (tm),
    tm-limit (tm)))
    = mapup-address-space (tm-memory (tm-execute-nullary (opcode, tm)),
    tm-regs (tm-execute-nullary (opcode, tm)),
    tm-cc (tm-execute-nullary (opcode, tm)),
    tm-error (tm-execute-nullary (opcode, tm)),
    tm-svcflag (tm-execute-nullary (opcode, tm)),
    tm-svcid (tm-execute-nullary (opcode, tm)),
    tm-base (tm-execute-nullary (opcode, tm)),
    tm-limit (tm-execute-nullary (opcode, tm))))

```

EVENT: Disable tm-execute-nullary-commutes-with-mapup-address-space.

DEFINITION:

```

tm-execute-unary8 (opcode, addr, tm)
= if opcode = '50 then tm-execute-test-iport (addr, tm)
  elseif opcode = '54 then tm-execute-test-oport (addr, tm)
  else tm-set-error ('2, tm) endif

```

DEFINITION:

```

tm-execute-unary7 (opcode, addr, tm)
= if opcode = '46 then tm-execute-svc-return (addr, tm)
  elseif opcode = '38 then tm-execute-post-output-interrupt (addr, tm)
  else tm-execute-unary8 (opcode, addr, tm) endif

```

DEFINITION:

```

tm-execute-unary6 (opcode, addr, tm)
= if opcode = '58 then tm-execute-set-clock (addr, tm)
  elseif opcode = '42 then tm-execute-svc (addr, tm)
  else tm-execute-unary7 (opcode, addr, tm) endif

```

DEFINITION:

```

tm-execute-unary5 (opcode, addr, tm)
= if opcode = '30 then tm-execute-load-limit (addr, tm)
  elseif opcode = '34 then tm-execute-load-psw (addr, tm)
  else tm-execute-unary6 (opcode, addr, tm) endif

```

DEFINITION:

```
tm-execute-unary4(opcode, addr, tm)
= if opcode = '22 then tm-execute-incr(addr, tm)
  elseif opcode = '26 then tm-execute-load-base(addr, tm)
  else tm-execute-unary5(opcode, addr, tm) endif
```

DEFINITION:

```
tm-execute-unary3(opcode, addr, tm)
= if opcode = '6 then tm-execute-branch-not-zero(addr, tm)
  elseif opcode = '18 then tm-execute-decr(addr, tm)
  else tm-execute-unary4(opcode, addr, tm) endif
```

DEFINITION:

```
tm-execute-unary2(opcode, addr, tm)
= if opcode = '10 then tm-execute-branch-on-zero(addr, tm)
  elseif opcode = '14 then tm-execute-call(addr, tm)
  else tm-execute-unary3(opcode, addr, tm) endif
```

DEFINITION:

```
tm-execute-unary1(opcode, addr, tm)
= if  $\neg$  tm-good-address(addr, tm) then tm-set-error('6, tm)
  elseif opcode = '2 then tm-execute-branch(addr, tm)
  else tm-execute-unary2(opcode, addr, tm) endif
```

THEOREM: tm-execute-unary1-equals-tm-execute-unary
tm-execute-unary1(*opcode*, *addr*, *tm*) = tm-execute-unary(*opcode*, *addr*, *tm*)

EVENT: Disable tm-execute-unary1-equals-tm-execute-unary.

THEOREM: tm-execute-unary8-commutes-with-mapup-address-space
(good-tm(*tm*))

```
 $\wedge$  (tm-good-address(addr, tm)
   $\wedge$  (if '65536 < (tm-base(tm) + tm-limit(tm))
    then '*1*false
    else '*1*true endif
     $\wedge$  ( $\neg$  tm-in-supervisor-mode(tm))))))
 $\rightarrow$  (tm-execute-unary8(opcode,
  addr,
  mapup-address-space(tm-memory(tm),
    tm-regs(tm),
    tm-cc(tm),
    tm-error(tm),
    tm-svcflag(tm),
    tm-svcid(tm),
```


$$\begin{aligned}
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-unary8}(opcode, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-regs}(\text{tm-execute-unary8}(opcode, addr, tm)), \\
& \text{tm-cc}(\text{tm-execute-unary8}(opcode, addr, tm)), \\
& \text{tm-error}(\text{tm-execute-unary8}(opcode, addr, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-unary8}(opcode, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-svcid}(\text{tm-execute-unary8}(opcode, addr, tm)), \\
& \text{tm-base}(\text{tm-execute-unary8}(opcode, addr, tm)), \\
& \text{tm-limit}(\text{tm-execute-unary8}(opcode, addr, tm))))
\end{aligned}$$

EVENT: Disable tm-execute-unary8-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary7-commutes-with-mapup-address-space
(good-tm(tm))

$$\begin{aligned}
& \wedge (\text{tm-good-address}(addr, tm) \\
& \quad \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \\
& \quad \quad \text{then } '*1*false \\
& \quad \quad \text{else } '*1*true \text{ endif} \\
& \quad \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))))) \\
\rightarrow & (\text{tm-execute-unary7}(opcode, \\
& \text{addr}, \\
& \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \text{tm-regs}(tm), \\
& \text{tm-cc}(tm), \\
& \text{tm-error}(tm), \\
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-unary7}(opcode, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-regs}(\text{tm-execute-unary7}(opcode, addr, tm)), \\
& \text{tm-cc}(\text{tm-execute-unary7}(opcode, addr, tm)), \\
& \text{tm-error}(\text{tm-execute-unary7}(opcode, addr, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-unary7}(opcode, \\
& \text{addr}, \\
& tm)), \\
& \text{tm-svcid}(\text{tm-execute-unary7}(opcode, addr, tm)),
\end{aligned}$$

tm-base (tm-execute-unary7 (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary7 (*opcode*, *addr*, *tm*)))

EVENT: Disable tm-execute-unary7-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary6-commutes-with-mapup-address-space

```
(good-tm (tm)
  ∧ (tm-good-address (addr, tm)
    ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
      then '*1*false
      else '*1*true endif
      ∧ (¬ tm-in-supervisor-mode (tm))))))
→ (tm-execute-unary6 (opcode,
  addr,
  mapup-address-space (tm-memory (tm),
    tm-regs (tm),
    tm-cc (tm),
    tm-error (tm),
    tm-svcflag (tm),
    tm-svcid (tm),
    tm-base (tm),
    tm-limit (tm)))
  = mapup-address-space (tm-memory (tm-execute-unary6 (opcode,
    addr,
    tm)),
    tm-regs (tm-execute-unary6 (opcode, addr, tm)),
    tm-cc (tm-execute-unary6 (opcode, addr, tm)),
    tm-error (tm-execute-unary6 (opcode, addr, tm)),
    tm-svcflag (tm-execute-unary6 (opcode,
      addr,
      tm)),
    tm-svcid (tm-execute-unary6 (opcode, addr, tm)),
    tm-base (tm-execute-unary6 (opcode, addr, tm)),
    tm-limit (tm-execute-unary6 (opcode, addr, tm))))
```

EVENT: Disable tm-execute-unary6-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary5-commutes-with-mapup-address-space

```
(good-tm (tm)
  ∧ (tm-good-address (addr, tm)
    ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
      then '*1*false
      else '*1*true endif
```

$$\begin{aligned}
& \wedge (\neg \text{tm-in-supervisor-mode}(tm))) \\
\rightarrow & (\text{tm-execute-unary5}(opcode, \\
& \quad addr, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \quad \text{tm-regs}(tm), \\
& \quad \quad \text{tm-cc}(tm), \\
& \quad \quad \text{tm-error}(tm), \\
& \quad \quad \text{tm-svcflag}(tm), \\
& \quad \quad \text{tm-svcid}(tm), \\
& \quad \quad \text{tm-base}(tm), \\
& \quad \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-unary5}(opcode, \\
& \quad \quad \quad addr, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-regs}(\text{tm-execute-unary5}(opcode, addr, tm)), \\
& \quad \text{tm-cc}(\text{tm-execute-unary5}(opcode, addr, tm)), \\
& \quad \text{tm-error}(\text{tm-execute-unary5}(opcode, addr, tm)), \\
& \quad \text{tm-svcflag}(\text{tm-execute-unary5}(opcode, \\
& \quad \quad \quad addr, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-svcid}(\text{tm-execute-unary5}(opcode, addr, tm)), \\
& \quad \text{tm-base}(\text{tm-execute-unary5}(opcode, addr, tm)), \\
& \quad \text{tm-limit}(\text{tm-execute-unary5}(opcode, addr, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-unary5-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary4-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{tm-good-address}(addr, tm) \\
& \quad \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \\
& \quad \quad \text{then } '*1*false \\
& \quad \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))) \\
\rightarrow & (\text{tm-execute-unary4}(opcode, \\
& \quad addr, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \quad \text{tm-regs}(tm), \\
& \quad \quad \text{tm-cc}(tm), \\
& \quad \quad \text{tm-error}(tm), \\
& \quad \quad \text{tm-svcflag}(tm), \\
& \quad \quad \text{tm-svcid}(tm), \\
& \quad \quad \text{tm-base}(tm), \\
& \quad \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-unary4}(opcode,
\end{aligned}$$

$$\begin{aligned}
& \text{addr,} \\
& \text{tm}), \\
& \text{tm-regs (tm-execute-unary4 (opcode, addr, tm)),} \\
& \text{tm-cc (tm-execute-unary4 (opcode, addr, tm)),} \\
& \text{tm-error (tm-execute-unary4 (opcode, addr, tm)),} \\
& \text{tm-svcflag (tm-execute-unary4 (opcode,} \\
& \quad \text{addr,} \\
& \quad \text{tm)),} \\
& \text{tm-svcid (tm-execute-unary4 (opcode, addr, tm)),} \\
& \text{tm-base (tm-execute-unary4 (opcode, addr, tm)),} \\
& \text{tm-limit (tm-execute-unary4 (opcode, addr, tm))})
\end{aligned}$$

EVENT: Disable tm-execute-unary4-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary3-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm (tm)} \\
& \wedge (\text{tm-good-address (addr, tm)} \\
& \quad \wedge (\text{if '65536 < (tm-base (tm) + tm-limit (tm))} \\
& \quad \quad \text{then '*1*false} \\
& \quad \quad \text{else '*1*true endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode (tm)}))) \\
\rightarrow & (\text{tm-execute-unary3 (opcode,} \\
& \quad \text{addr,} \\
& \quad \text{mapup-address-space (tm-memory (tm),} \\
& \quad \quad \text{tm-regs (tm),} \\
& \quad \quad \text{tm-cc (tm),} \\
& \quad \quad \text{tm-error (tm),} \\
& \quad \quad \text{tm-svcflag (tm),} \\
& \quad \quad \text{tm-svcid (tm),} \\
& \quad \quad \text{tm-base (tm),} \\
& \quad \quad \text{tm-limit (tm))}) \\
= & \text{mapup-address-space (tm-memory (tm-execute-unary3 (opcode,} \\
& \quad \text{addr,} \\
& \quad \text{tm)),} \\
& \text{tm-regs (tm-execute-unary3 (opcode, addr, tm)),} \\
& \text{tm-cc (tm-execute-unary3 (opcode, addr, tm)),} \\
& \text{tm-error (tm-execute-unary3 (opcode, addr, tm)),} \\
& \text{tm-svcflag (tm-execute-unary3 (opcode,} \\
& \quad \text{addr,} \\
& \quad \text{tm)),} \\
& \text{tm-svcid (tm-execute-unary3 (opcode, addr, tm)),} \\
& \text{tm-base (tm-execute-unary3 (opcode, addr, tm)),} \\
& \text{tm-limit (tm-execute-unary3 (opcode, addr, tm))})
\end{aligned}$$

EVENT: Disable tm-execute-unary3-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary2-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (tm-good-address (addr, tm)
    ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
      then '*1*false
      else '*1*true endif
      ∧ (¬ tm-in-supervisor-mode (tm))))))
→ (tm-execute-unary2 (opcode,
  addr,
  mapup-address-space (tm-memory (tm),
    tm-regs (tm),
    tm-cc (tm),
    tm-error (tm),
    tm-svcflag (tm),
    tm-svcid (tm),
    tm-base (tm),
    tm-limit (tm)))
  = mapup-address-space (tm-memory (tm-execute-unary2 (opcode,
    addr,
    tm)),
  tm-regs (tm-execute-unary2 (opcode, addr, tm)),
  tm-cc (tm-execute-unary2 (opcode, addr, tm)),
  tm-error (tm-execute-unary2 (opcode, addr, tm)),
  tm-svcflag (tm-execute-unary2 (opcode,
    addr,
    tm)),
  tm-svcid (tm-execute-unary2 (opcode, addr, tm)),
  tm-base (tm-execute-unary2 (opcode, addr, tm)),
  tm-limit (tm-execute-unary2 (opcode, addr, tm))))

```

EVENT: Disable tm-execute-unary2-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary1-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
    then '*1*false
    else '*1*true endif
    ∧ (¬ tm-in-supervisor-mode (tm))))
→ (tm-execute-unary1 (opcode,
  addr,
  mapup-address-space (tm-memory (tm),
    tm-regs (tm),

```

$$\begin{aligned}
& \text{tm-cc}(tm), \\
& \text{tm-error}(tm), \\
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-unary1}(opcode, \\
& \quad \text{addr}, \\
& \quad tm)), \\
& \text{tm-regs}(\text{tm-execute-unary1}(opcode, addr, tm)), \\
& \text{tm-cc}(\text{tm-execute-unary1}(opcode, addr, tm)), \\
& \text{tm-error}(\text{tm-execute-unary1}(opcode, addr, tm)), \\
& \text{tm-svcflag}(\text{tm-execute-unary1}(opcode, \\
& \quad \text{addr}, \\
& \quad tm)), \\
& \text{tm-svcid}(\text{tm-execute-unary1}(opcode, addr, tm)), \\
& \text{tm-base}(\text{tm-execute-unary1}(opcode, addr, tm)), \\
& \text{tm-limit}(\text{tm-execute-unary1}(opcode, addr, tm))))
\end{aligned}$$

EVENT: Disable tm-execute-unary1-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary-commutes-with-mapup-address-space
(good-tm(tm))
 \wedge (**if** '65536 < (tm-base(tm) + tm-limit(tm)) **then** '*1*false
else '*1*true **endif**
 \wedge (\neg tm-in-supervisor-mode(tm)))
 \rightarrow (tm-execute-unary($opcode$,
 $addr$,
mapup-address-space(tm-memory(tm),
tm-regs(tm),
tm-cc(tm),
tm-error(tm),
tm-svcflag(tm),
tm-svcid(tm),
tm-base(tm),
tm-limit(tm)))
= mapup-address-space(tm-memory(tm-execute-unary($opcode$, $addr$, tm)),
tm-regs(tm-execute-unary($opcode$, $addr$, tm)),
tm-cc(tm-execute-unary($opcode$, $addr$, tm)),
tm-error(tm-execute-unary($opcode$, $addr$, tm)),
tm-svcflag(tm-execute-unary($opcode$,
 $addr$,
 tm)),
tm-svcid(tm-execute-unary($opcode$, $addr$, tm)),

tm-base (tm-execute-unary (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary (*opcode*, *addr*, *tm*)))

EVENT: Disable tm-execute-unary-commutes-with-mapup-address-space.

DEFINITION:

tm-execute-binary6 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** *opcode* = '7 **then** tm-execute-incr-mod (*addr1*, *addr2*, *tm*)
else tm-set-error ('2, *tm*) **endif**

DEFINITION:

tm-execute-binary5 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** *opcode* = '15 **then** tm-execute-start-output (*addr1*, *addr2*, *tm*)
elseif *opcode* = '5 **then** tm-execute-decr-mod (*addr1*, *addr2*, *tm*)
else tm-execute-binary6 (*opcode*, *addr1*, *addr2*, *tm*) **endif**

DEFINITION:

tm-execute-binary4 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** *opcode* = '11 **then** tm-execute-move (*addr1*, *addr2*, *tm*)
elseif *opcode* = '13 **then** tm-execute-multiply (*addr1*, *addr2*, *tm*)
else tm-execute-binary5 (*opcode*, *addr1*, *addr2*, *tm*) **endif**

DEFINITION:

tm-execute-binary3 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** *opcode* = '3 **then** tm-execute-compare (*addr1*, *addr2*, *tm*)
elseif *opcode* = '9 **then** tm-execute-mod (*addr1*, *addr2*, *tm*)
else tm-execute-binary4 (*opcode*, *addr1*, *addr2*, *tm*) **endif**

DEFINITION:

tm-execute-binary2 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** *opcode* = '1 **then** tm-execute-add (*addr1*, *addr2*, *tm*)
else tm-execute-binary3 (*opcode*, *addr1*, *addr2*, *tm*) **endif**

DEFINITION:

tm-execute-binary1 (*opcode*, *addr1*, *addr2*, *tm*)
= **if** (\neg tm-good-address (*addr1*, *tm*))
 \vee (\neg tm-good-address (*addr2*, *tm*)) **then** tm-set-error ('6, *tm*)
else tm-execute-binary2 (*opcode*, *addr1*, *addr2*, *tm*) **endif**

THEOREM: tm-execute-binary1-equals-tm-execute-binary

tm-execute-binary1 (*opcode*, *addr1*, *addr2*, *tm*)
= tm-execute-binary (*opcode*, *addr1*, *addr2*, *tm*)

EVENT: Disable tm-execute-binary1-equals-tm-execute-binary.

THEOREM: tm-execute-binary6-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm } (tm) \\
& \wedge (\text{tm-good-address } (addr1, tm) \\
& \quad \wedge (\text{tm-good-address } (addr2, tm) \\
& \quad \quad \wedge (\text{if } '65536 < (\text{tm-base } (tm) + \text{tm-limit } (tm)) \\
& \quad \quad \quad \text{then } '*1*\text{false} \\
& \quad \quad \quad \text{else } '*1*\text{true endif} \\
& \quad \quad \quad \wedge (\neg \text{tm-in-supervisor-mode } (tm)))))) \\
\rightarrow & (\text{tm-execute-binary6 } (opcode, \\
& \quad addr1, \\
& \quad addr2, \\
& \quad \text{mapup-address-space } (\text{tm-memory } (tm), \\
& \quad \quad \text{tm-regs } (tm), \\
& \quad \quad \text{tm-cc } (tm), \\
& \quad \quad \text{tm-error } (tm), \\
& \quad \quad \text{tm-svcflag } (tm), \\
& \quad \quad \text{tm-svcid } (tm), \\
& \quad \quad \text{tm-base } (tm), \\
& \quad \quad \text{tm-limit } (tm))) \\
& = \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-regs } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-cc } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-error } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-svcflag } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-svcid } (\text{tm-execute-binary6 } (opcode, \\
& \quad \quad \quad addr1, \\
& \quad \quad \quad addr2, \\
& \quad \quad \quad tm)), \\
& \quad \text{tm-base } (\text{tm-execute-binary6 } (opcode,
\end{aligned}$$

$addr1,$
 $addr2,$
 $tm),$
 $tm-limit (tm-execute-binary6 (opcode,$
 $addr1,$
 $addr2,$
 $tm))))$

EVENT: Disable tm-execute-binary6-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary5-commutes-with-mapup-address-space
(good-tm (tm))

\wedge (tm-good-address ($addr1, tm$)
 \wedge (tm-good-address ($addr2, tm$)
 \wedge (**if** '65536 < (tm-base (tm) + tm-limit (tm))
then '*1*false
else '*1>true **endif**
 \wedge (\neg tm-in-supervisor-mode (tm))))))
 \rightarrow (tm-execute-binary5 ($opcode,$
 $addr1,$
 $addr2,$
mapup-address-space (tm-memory (tm),
tm-regs (tm),
tm-cc (tm),
tm-error (tm),
tm-svcflag (tm),
tm-svcid (tm),
tm-base (tm),
tm-limit (tm)))
= mapup-address-space (tm-memory (tm-execute-binary5 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
tm-regs (tm-execute-binary5 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
tm-cc (tm-execute-binary5 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
tm-error (tm-execute-binary5 ($opcode,$
 $addr1,$
 $addr2,$

```

tm-svcflag (tm-execute-binary5 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-svcid (tm-execute-binary5 (opcode,
                              addr1,
                              addr2,
                              tm)),
tm-base (tm-execute-binary5 (opcode,
                             addr1,
                             addr2,
                             tm)),
tm-limit (tm-execute-binary5 (opcode,
                              addr1,
                              addr2,
                              tm))))

```

EVENT: Disable tm-execute-binary5-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary4-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (tm-good-address (addr1, tm)
     ∧ (tm-good-address (addr2, tm)
        ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
              then '*1*false
              else '*1*true endif
         ∧ (¬ tm-in-supervisor-mode (tm))))))
→ (tm-execute-binary4 (opcode,
                       addr1,
                       addr2,
                       mapup-address-space (tm-memory (tm),
                                             tm-regs (tm),
                                             tm-cc (tm),
                                             tm-error (tm),
                                             tm-svcflag (tm),
                                             tm-svcid (tm),
                                             tm-base (tm),
                                             tm-limit (tm)))
                          tm-regs (tm-execute-binary4 (opcode,
                                                           addr1,
                                                           addr2,
                                                           tm)))
    tm-regs (tm-execute-binary4 (opcode,

```

```

                                addr1,
                                addr2,
                                tm)),
tm-cc (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-error (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-svcflag (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-svcid (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-base (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-limit (tm-execute-binary4 (opcode,
                                addr1,
                                addr2,
                                tm))))

```

EVENT: Disable tm-execute-binary4-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary3-commutes-with-mapup-address-space
(good-tm (*tm*)

```

  ∧ (tm-good-address (addr1, tm)
    ∧ (tm-good-address (addr2, tm)
      ∧ (if '65536 < (tm-base (tm) + tm-limit (tm))
            then '*1*false
            else '*1*true endif
        ∧ (¬ tm-in-supervisor-mode (tm))))))

```

```

→ (tm-execute-binary3 (opcode,
                        addr1,
                        addr2,
                        mapup-address-space (tm-memory (tm),
                                             tm-regs (tm),
                                             tm-cc (tm),

```

$$\begin{aligned}
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm)) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-regs } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-cc } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-error } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-svcflag } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-svcid } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-base } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm)), \\
& \text{tm-limit } (\text{tm-execute-binary3 } (opcode, \\
& \text{addr1}, \\
& \text{addr2}, \\
& tm))))))
\end{aligned}$$

EVENT: Disable tm-execute-binary3-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary2-commutes-with-mapup-address-space
(good-tm (tm))
 \wedge (tm-good-address ($addr1$, tm)
 \wedge (tm-good-address ($addr2$, tm)))

$$\begin{aligned}
& \wedge \text{ (if '65536} < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \\
& \quad \text{then '*1*false} \\
& \quad \text{else '*1>true endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))))) \\
\rightarrow & (\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \text{addr1}, \\
& \quad \text{addr2}, \\
& \quad \text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \quad \text{tm-regs}(tm), \\
& \quad \quad \text{tm-cc}(tm), \\
& \quad \quad \text{tm-error}(tm), \\
& \quad \quad \text{tm-svcflag}(tm), \\
& \quad \quad \text{tm-svcid}(tm), \\
& \quad \quad \text{tm-base}(tm), \\
& \quad \quad \text{tm-limit}(tm))) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \text{addr1}, \\
& \quad \text{addr2}, \\
& \quad \text{tm})), \\
& \quad \text{tm-regs}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-cc}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-error}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-svcflag}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-svcid}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-base}(\text{tm-execute-binary2}(\text{opcode}, \\
& \quad \quad \text{addr1}, \\
& \quad \quad \text{addr2}, \\
& \quad \quad \text{tm})), \\
& \quad \text{tm-limit}(\text{tm-execute-binary2}(\text{opcode},
\end{aligned}$$

$addr1,$
 $addr2,$
 $tm))))$

EVENT: Disable tm-execute-binary2-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary1-commutes-with-mapup-address-space
 (good-tm (tm))

\wedge (**if** '65536 < (tm-base (tm) + tm-limit (tm)) **then** '*1*false
else '*1*true **endif**
 \wedge (\neg tm-in-supervisor-mode (tm)))

\rightarrow (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),
 tm-svcflag (tm),
 tm-svcid (tm),
 tm-base (tm),
 tm-limit (tm))
)
)

 $=$ mapup-address-space (tm-memory (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
 tm-regs (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
 tm-cc (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
 tm-error (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
 tm-svcflag (tm-execute-binary1 ($opcode,$
 $addr1,$
 $addr2,$
 tm)),
 tm-svcid (tm-execute-binary1 ($opcode,$
 $addr1,$

```

                                addr2,
                                tm)),
tm-base (tm-execute-binary1 (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-limit (tm-execute-binary1 (opcode,
                                addr1,
                                addr2,
                                tm))))

```

EVENT: Disable tm-execute-binary1-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary-commutes-with-mapup-address-space
(good-tm (*tm*)
 \wedge (if '65536 < (tm-base (*tm*) + tm-limit (*tm*)) then '*1*false
else '*1*true endif
 \wedge (\neg tm-in-supervisor-mode (*tm*))))
 \rightarrow (tm-execute-binary (*opcode*,
addr1,
addr2,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*))
= mapup-address-space (tm-memory (tm-execute-binary (*opcode*,
addr1,
addr2,
tm)),
tm-regs (tm-execute-binary (*opcode*,
addr1,
addr2,
tm)),
tm-cc (tm-execute-binary (*opcode*,
addr1,
addr2,
tm)),
tm-error (tm-execute-binary (*opcode*,
addr1,
addr2,

```

tm-svcflag (tm-execute-binary (opcode,
                                addr1,
                                addr2,
                                tm)),
tm-svcid (tm-execute-binary (opcode,
                              addr1,
                              addr2,
                              tm)),
tm-base (tm-execute-binary (opcode,
                             addr1,
                             addr2,
                             tm)),
tm-limit (tm-execute-binary (opcode,
                              addr1,
                              addr2,
                              tm))))

```

EVENT: Disable tm-execute-binary-commutes-with-mapup-address-space.

THEOREM: tm-compute-address-mapup-address-space
 (good-tm (tm) \wedge (\neg tm-in-supervisor-mode (tm)))
 \rightarrow (tm-compute-address (arg ,
 mapup-address-space (tm-memory (tm),
 tm-regs (tm),
 tm-cc (tm),
 tm-error (tm),
 tm-svcflag (tm),
 tm-svcid (tm),
 tm-base (tm),
 tm-limit (tm)))
 = tm-compute-address (arg , tm))

EVENT: Disable tm-compute-address-mapup-address-space.

DEFINITION:
 tm-execute-instruction2 ($instruction$, tm)
 = **if** length ($instruction$) = '3
 then tm-execute-binary (tm-opcode (car ($instruction$)),
 tm-compute-address (tm-arg1 ($instruction$), tm),
 tm-compute-address (tm-arg2 ($instruction$), tm),
 tm)
 else tm-set-error ('2, tm) **endif**

DEFINITION:

```

tm-execute-instruction1 (instruction, tm)
=  if length (instruction) = '1
    then tm-execute-nullary (tm-opcode (car (instruction)), tm)
    elseif length (instruction) = '2
    then tm-execute-unary (tm-opcode (car (instruction)),
                          tm-compute-address (tm-arg1 (instruction), tm),
                          tm)
    else tm-execute-instruction2 (instruction, tm) endif

```

THEOREM: tm-execute-instruction1-equals-tm-execute-instruction
tm-execute-instruction1 (*instruction*, *tm*)
= tm-execute-instruction (*instruction*, *tm*)

EVENT: Disable tm-execute-instruction1-equals-tm-execute-instruction.

THEOREM: tm-execute-instruction2-commutes-with-mapup-address-space
(good-tm (*tm*)

```

  ∧ (if '65536 < (tm-base (tm) + tm-limit (tm)) then '*1*false
     else '*1*true endif
     ∧ (¬ tm-in-supervisor-mode (tm)))
→  (tm-execute-instruction2 (instruction,
                             mapup-address-space (tm-memory (tm),
                                                  tm-regs (tm),
                                                  tm-cc (tm),
                                                  tm-error (tm),
                                                  tm-svcflag (tm),
                                                  tm-svcid (tm),
                                                  tm-base (tm),
                                                  tm-limit (tm))
                             tm-execute-instruction2 (instruction,
                                                         tm)),
    tm-regs (tm-execute-instruction2 (instruction,
                                       tm)),
    tm-cc (tm-execute-instruction2 (instruction,
                                     tm)),
    tm-error (tm-execute-instruction2 (instruction,
                                        tm)),
    tm-svcflag (tm-execute-instruction2 (instruction,
                                          tm)),
    tm-svcid (tm-execute-instruction2 (instruction,
                                        tm)),
    tm-base (tm-execute-instruction2 (instruction,
                                       tm)),

```

tm-limit (tm-execute-instruction2 (*instruction*,
tm))))

EVENT: Disable tm-execute-instruction2-commutes-with-mapup-address-space.

THEOREM: tm-execute-instruction1-commutes-with-mapup-address-space

(good-tm (*tm*)
 \wedge (if '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
 \wedge (\neg tm-in-supervisor-mode (*tm*))))
 \rightarrow (tm-execute-instruction1 (*instruction*,
mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-instruction1 (*instruction*,
tm)),
tm-regs (tm-execute-instruction1 (*instruction*,
tm)),
tm-cc (tm-execute-instruction1 (*instruction*,
tm)),
tm-error (tm-execute-instruction1 (*instruction*,
tm)),
tm-svcflag (tm-execute-instruction1 (*instruction*,
tm)),
tm-svcid (tm-execute-instruction1 (*instruction*,
tm)),
tm-base (tm-execute-instruction1 (*instruction*,
tm)),
tm-limit (tm-execute-instruction1 (*instruction*,
tm))))

EVENT: Disable tm-execute-instruction1-commutes-with-mapup-address-space.

THEOREM: tm-execute-instruction-commutes-with-mapup-address-space

(good-tm (*tm*)
 \wedge (if '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
 \wedge (\neg tm-in-supervisor-mode (*tm*))))

tm-base (tm-decrement-clock (*tm*)),
tm-limit (tm-decrement-clock (*tm*)))

EVENT: Disable tm-decrement-clock-commutes-with-mapup-address-space.

THEOREM: tm-in-supervisor-mode-tm-decrement-clock
tm-in-supervisor-mode (tm-decrement-clock (*tm*)) = tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-decrement-clock.

THEOREM: tm-fetch-arg1-mapup-address-space
(good-tm (*tm*)
 \wedge ((\neg tm-in-supervisor-mode (*tm*))
 \wedge tm-good-address (real-addr ('2, tm-incrn ('1, tm-pc (*tm*))), *tm*)))
 \rightarrow (tm-fetch-arg1 (mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= tm-fetch-arg1 (*tm*))

EVENT: Disable tm-fetch-arg1-mapup-address-space.

THEOREM: tm-fetch-arg2-mapup-address-space
(good-tm (*tm*)
 \wedge ((\neg tm-in-supervisor-mode (*tm*))
 \wedge tm-good-address (real-addr ('2, tm-incrn ('2, tm-pc (*tm*))), *tm*)))
 \rightarrow (tm-fetch-arg2 (mapup-address-space (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= tm-fetch-arg2 (*tm*))

EVENT: Disable tm-fetch-arg2-mapup-address-space.

DEFINITION:

```

tm-execute2(opcode, tm)
=  if tm-unary-instructionp(opcode)
    then tm-execute-instruction(cons(opcode,
                                     cons(tm-fetch-arg1(tm), 'nil)),
                                 tm-set-pc(tm-incrn('2, tm-pc(tm)), tm))
    elseif tm-good-address(real-addr('2, tm-incrn('2, tm-pc(tm))), tm)
    then tm-execute-instruction(cons(opcode,
                                     cons(tm-fetch-arg1(tm),
                                             cons(tm-fetch-arg2(tm), 'nil))),
                                 tm-set-pc(tm-incrn('3, tm-pc(tm)), tm))
    else tm-set-error('7, tm) endif

```

DEFINITION:

```

tm-execute1(opcode, tm)
=  if tm-nullary-instructionp(opcode)
    then tm-execute-instruction(cons(opcode, 'nil),
                                   tm-set-pc(tm-incrn('1, tm-pc(tm)), tm))
    elseif tm-good-address(real-addr('2, tm-incrn('1, tm-pc(tm))), tm)
    then tm-execute2(opcode, tm)
    else tm-set-error('7, tm) endif

```

THEOREM: tm-execute1-equals-tm-execute

tm-execute1(*opcode*, *tm*) = tm-execute(*opcode*, *tm*)

EVENT: Disable tm-execute1-equals-tm-execute.

THEOREM: tm-execute2-commutes-with-mapup-address-space

```

(good-tm(tm)
 ∧ (if '65536 < (tm-base(tm) + tm-limit(tm)) then '*1*false
   else '*1*true endif
 ∧ ((¬ tm-in-supervisor-mode(tm))
   ∧ tm-good-address(real-addr('2,
                               tm-incrn('1, tm-pc(tm))),
                       tm)))
→ (tm-execute2(instruction,
               mapup-address-space(tm-memory(tm),
                                     tm-regs(tm),
                                     tm-cc(tm),
                                     tm-error(tm),
                                     tm-svcflag(tm),
                                     tm-svcid(tm),
                                     tm-base(tm),
                                     tm-limit(tm)))
   = mapup-address-space(tm-memory(tm-execute2(instruction, tm)),

```

```

tm-regs (tm-execute2 (instruction, tm)),
tm-cc (tm-execute2 (instruction, tm)),
tm-error (tm-execute2 (instruction, tm)),
tm-svcflag (tm-execute2 (instruction, tm)),
tm-svcid (tm-execute2 (instruction, tm)),
tm-base (tm-execute2 (instruction, tm)),
tm-limit (tm-execute2 (instruction, tm)))

```

EVENT: Disable tm-execute2-commutes-with-mapup-address-space.

THEOREM: tm-execute1-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (if '65536 < (tm-base (tm) + tm-limit (tm)) then '*1*false
      else '*1*true endif
    ∧ (¬ tm-in-supervisor-mode (tm))))
→ (tm-execute1 (instruction,
                mapup-address-space (tm-memory (tm),
                                     tm-regs (tm),
                                     tm-cc (tm),
                                     tm-error (tm),
                                     tm-svcflag (tm),
                                     tm-svcid (tm),
                                     tm-base (tm),
                                     tm-limit (tm)))
    = mapup-address-space (tm-memory (tm-execute1 (instruction, tm)),
                           tm-regs (tm-execute1 (instruction, tm)),
                           tm-cc (tm-execute1 (instruction, tm)),
                           tm-error (tm-execute1 (instruction, tm)),
                           tm-svcflag (tm-execute1 (instruction, tm)),
                           tm-svcid (tm-execute1 (instruction, tm)),
                           tm-base (tm-execute1 (instruction, tm)),
                           tm-limit (tm-execute1 (instruction, tm))))

```

EVENT: Disable tm-execute1-commutes-with-mapup-address-space.

THEOREM: tm-execute-commutes-with-mapup-address-space

```

(good-tm (tm)
  ∧ (if '65536 < (tm-base (tm) + tm-limit (tm)) then '*1*false
      else '*1*true endif
    ∧ (¬ tm-in-supervisor-mode (tm))))
→ (tm-execute (instruction,
                mapup-address-space (tm-memory (tm),
                                     tm-regs (tm),

```

$$\begin{aligned}
& \text{tm-cc}(tm), \\
& \text{tm-error}(tm), \\
& \text{tm-svcflag}(tm), \\
& \text{tm-svcid}(tm), \\
& \text{tm-base}(tm), \\
& \text{tm-limit}(tm)) \\
= & \text{mapup-address-space}(\text{tm-memory}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-regs}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-cc}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-error}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-svcflag}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-svcid}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-base}(\text{tm-execute}(instruction, tm)), \\
& \text{tm-limit}(\text{tm-execute}(instruction, tm)))
\end{aligned}$$

EVENT: Disable tm-execute-commutes-with-mapup-address-space.

THEOREM: tm-fetch-opcode-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge ((\neg \text{tm-in-supervisor-mode}(tm)) \\
& \quad \quad \wedge \text{tm-good-address}(\text{real-addr}('2, \text{tm-pc}(tm), tm)))))) \\
\rightarrow & (\text{tm-fetch-opcode}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \text{tm-regs}(tm), \\
& \quad \text{tm-cc}(tm), \\
& \quad \text{tm-error}(tm), \\
& \quad \text{tm-svcflag}(tm), \\
& \quad \text{tm-svcid}(tm), \\
& \quad \text{tm-base}(tm), \\
& \quad \text{tm-limit}(tm))) \\
= & \text{tm-fetch-opcode}(tm)
\end{aligned}$$

EVENT: Disable tm-fetch-opcode-mapup-address-space.

THEOREM: tm-fetch-execute-commutes-with-mapup-address-space

$$\begin{aligned}
& (\text{good-tm}(tm) \\
& \wedge (\text{if } '65536 < (\text{tm-base}(tm) + \text{tm-limit}(tm)) \text{ then } '*1*false \\
& \quad \text{else } '*1*true \text{ endif} \\
& \quad \wedge (\neg \text{tm-in-supervisor-mode}(tm)))) \\
\rightarrow & (\text{tm-fetch-execute}(\text{mapup-address-space}(\text{tm-memory}(tm), \\
& \quad \text{tm-regs}(tm), \\
& \quad \text{tm-cc}(tm),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-error } (tm), \\
& \text{tm-svcflag } (tm), \\
& \text{tm-svcid } (tm), \\
& \text{tm-base } (tm), \\
& \text{tm-limit } (tm)) \\
= & \text{mapup-address-space } (\text{tm-memory } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-regs } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-cc } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-error } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-svcflag } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-svcid } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-base } (\text{tm-fetch-execute } (tm)), \\
& \text{tm-limit } (\text{tm-fetch-execute } (tm)))
\end{aligned}$$

EVENT: Disable tm-fetch-execute-commutes-with-mapup-address-space.

DEFINITION:

$$\text{deq-ith-buffer } (id, buffers) = \text{putnth } (\text{deq } (\text{getnth } (id, buffers)), id, buffers)$$

DEFINITION:

$$\begin{aligned}
& \text{enq-ith-buffer } (item, id, buffers) \\
= & \text{putnth } (\text{enq } (item, \text{getnth } (id, buffers)), id, buffers)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{qreplace-ith-buffer } (item, id, buffers) \\
= & \text{putnth } (\text{qreplace } (item, \text{getnth } (id, buffers)), id, buffers)
\end{aligned}$$

EVENT: Add the shell *task*, with recognizer function symbol *task-shellp* and 2 accessors: *task-pstate*, with type restriction (none-of) and default value zero; *task-channels*, with type restriction (none-of) and default value zero.

EVENT: Let us define the theory *task-shell* to consist of the following events: *task*.

DEFINITION: $\text{task-ibuffers } (task) = \text{car } (\text{task-channels } (task))$

DEFINITION: $\text{task-obuffers } (task) = \text{car } (\text{cdr } (\text{task-channels } (task)))$

DEFINITION:

$$\text{task-mbuffers } (task) = \text{car } (\text{cdr } (\text{cdr } (\text{task-channels } (task))))$$

DEFINITION: $\text{TASK-IBUFFER-CAPACITY} = '4$

DEFINITION: $\text{TASK-OBUFFER-CAPACITY} = '4$

DEFINITION: $\text{TASK-MBUFFER-CAPACITY} = '4$

DEFINITION: $\text{OS-SVC-SEND-ID} = '0$

DEFINITION: $\text{OS-SVC-RECEIVE-ID} = '1$

DEFINITION: $\text{OS-SVC-TYO-ID} = '2$

DEFINITION: $\text{OS-SVC-TYI-ID} = '3$

DEFINITION:

$\text{task-send-instructionp}(task)$
 $= ((\text{tm-svcid}(\text{task-pstate}(task)) \bmod '4) = '0)$

DEFINITION:

$\text{task-receive-instructionp}(task)$
 $= ((\text{tm-svcid}(\text{task-pstate}(task)) \bmod '4) = '1)$

DEFINITION:

$\text{task-tyo-instructionp}(task)$
 $= ((\text{tm-svcid}(\text{task-pstate}(task)) \bmod '4) = '2)$

DEFINITION:

$\text{task-tyi-instructionp}(task)$
 $= ((\text{tm-svcid}(\text{task-pstate}(task)) \bmod '4) = '3)$

DEFINITION:

$\text{task-update-control}(pstate) = \text{tm-set-svcflag}('0, pstate)$

DEFINITION:

$\text{task-communicationp}(task) = \text{tm-svc-interruptp}(\text{task-pstate}(task))$

DEFINITION: $\text{os-srcid}(task, max) = (\text{tm-r2}(task) \bmod max)$

DEFINITION: $\text{os-destid}(task, max) = (\text{tm-r2}(task) \bmod max)$

DEFINITION: $\text{os-idevid}(task, max) = (\text{tm-r2}(task) \bmod max)$

DEFINITION: $\text{os-odevid}(task, max) = (\text{tm-r2}(task) \bmod max)$

DEFINITION: $\text{os-message}(task) = \text{tm-r3}(task)$

DEFINITION:

$\text{os-store-message}(msg, task) = \text{tm-store}(msg, '(1 3), task)$

DEFINITION:

$\text{task-srcid}(task)$
 $= \text{os-srcid}(\text{task-pstate}(task), \text{length}(\text{task-mbuffers}(task)))$

DEFINITION:

task-destid (*task*)
= os-destid (task-pstate (*task*), length (task-mbuffers (*task*)))

DEFINITION:

task-idevid (*task*)
= os-idevid (task-pstate (*task*), length (task-ibuffers (*task*)))

DEFINITION:

task-odevid (*task*)
= os-odevid (task-pstate (*task*), length (task-ibuffers (*task*)))

DEFINITION: task-message (*task*) = os-message (task-pstate (*task*))

DEFINITION:

task-store-message (*msg*, *pstate*) = os-store-message (*msg*, *pstate*)

DEFINITION:

task-fetch-execute (*pstate*) = tm-fetch-execute (*pstate*)

DEFINITION:

task-execute-send (*msg*, *srcid*, *destid*, *task*)
= **if** qfullp2 (*srcid*, *destid*, task-mbuffers (*task*), '4) **then** *task*
 else task (task-update-control (task-pstate (*task*)),
 cons (task-ibuffers (*task*),
 cons (task-obuffers (*task*),
 cons (enq2 (*msg*,
 srcid,
 destid,
 task-mbuffers (*task*)),
 'nil)))) **endif**

DEFINITION:

task-execute-recv (*srcid*, *destid*, *task*)
= **if** qempty2 (*srcid*, *destid*, task-mbuffers (*task*)) **then** *task*
 else task (task-update-control (task-store-message (qfirst2 (*srcid*,
 destid,
 task-mbuffers (*task*)),
 task-pstate (*task*))),
 cons (task-ibuffers (*task*),
 cons (task-obuffers (*task*),
 cons (deq2 (*srcid*, *destid*, task-mbuffers (*task*)),
 'nil)))) **endif**

DEFINITION:

```

task-execute-output (char, id, task)
= if qfullp (getnth (id, task-obuffers (task)), '4) then task
  else task (task-update-control (task-pstate (task)),
            cons (task-ibuffers (task),
                  cons (enq-ith-buffer (char, id, task-obuffers (task)),
                        cons (task-mbuffers (task), 'nil)))) endif

```

DEFINITION:

```

task-execute-input (id, task)
= if qempty (getnth (id, task-ibuffers (task))) then task
  else task (task-update-control (task-store-message (qfirst (getnth (id,
                                                                    task-ibuffers (task))),
                                                                    task-pstate (task))),
            cons (deq-ith-buffer (id, task-ibuffers (task)),
                  cons (task-obuffers (task),
                        cons (task-mbuffers (task), 'nil)))) endif

```

DEFINITION:

```

task-communication-step (task, i)
= if task-send-instructionp (task)
  then task-execute-send (task-message (task), i, task-destid (task), task)
  elseif task-receive-instructionp (task)
  then task-execute-receive (task-srcid (task), i, task)
  elseif task-tyo-instructionp (task)
  then task-execute-output (task-message (task), i, task)
  else task-execute-input (i, task) endif

```

DEFINITION:

```

task-private-step (task)
= task (task-fetch-execute (task-pstate (task)), task-channels (task))

```

DEFINITION:

```

task-step (task, i)
= if task-communicationp (task) then task-communication-step (task, i)
  else task-private-step (task) endif

```

DEFINITION:

```

task-update-channels (task, channels) = task (task-pstate (task), channels)

```

DEFINITION: task-activep (*x*) = (*x* = '*1*true)

DEFINITION:

```

task-processor (task, i, oracle)
= if listp (oracle)
  then if task-activep (car (oracle))

```

```

then task-processor (task-step (task, i), i, cdr (oracle))
else task-processor (task-update-channels (task, car (oracle)),
i,
cdr (oracle)) endif
else task endif

```

THEOREM: finite-numberp-os-message
good-address-space (*machine*, *memlength*)
→ finite-numberp (os-message (*machine*), '65536)

EVENT: Disable finite-numberp-os-message.

THEOREM: good-address-space-os-store-message
(good-address-space (*tm*, *memlength*) ∧ finite-numberp (*msg*, '65536))
→ good-address-space (os-store-message (*msg*, *tm*), *memlength*)

EVENT: Disable good-address-space-os-store-message.

THEOREM: good-address-space-os-store-message1
(good-address-space (*task*, length (tm-memory (*task*)))
∧ finite-numberp (*msg*, '65536))
→ good-address-space (os-store-message (*msg*, *task*), length (tm-memory (*task*)))

EVENT: Disable good-address-space-os-store-message1.

THEOREM: length-tm-memory-os-store-message
length (tm-memory (os-store-message (*msg*, *task*))) = length (tm-memory (*task*))

EVENT: Disable length-tm-memory-os-store-message.

DEFINITION:
good-task-buffer (*buffer*, *capacity*)
= (plistp (*buffer*)
∧ ((length (*buffer*) < (1 + *capacity*))
∧ finite-number-listp (*buffer*, '65536)))

DEFINITION:
good-task-buffer-list (*l*, *capacity*)
= **if** listp (*l*)
then good-task-buffer (car (*l*), *capacity*)
∧ good-task-buffer-list (cdr (*l*), *capacity*)
else *l* = 'nil **endif**

DEFINITION:

good-task-buffer-table (l , $length$, $capacity$)
= **if** listp (l)
 then plistp (car (l))
 \wedge ((length (car (l)) = $length$)
 \wedge (good-task-buffer-list (car (l), $capacity$)
 \wedge good-task-buffer-table (cdr (l),
 $length$,
 $capacity$)))
 else $l = 'nil$ **endif**

DEFINITION:

good-task ($task$, $ilength$, $olength$, $mlength$)
= (task-shellp ($task$)
 \wedge (good-address-space (task-pstate ($task$),
 length (tm-memory (task-pstate ($task$))))
 \wedge ((length (task-ibuffers ($task$)) = $ilength$)
 \wedge (good-task-buffer-list (task-ibuffers ($task$),
 '4)
 \wedge ((length (task-obuffers ($task$))
 = $olength$)
 \wedge (good-task-buffer-list (task-obuffers ($task$),
 '4)
 \wedge ((length (task-mbuffers ($task$))
 = $mlength$)
 \wedge good-task-buffer-table (task-mbuffers ($task$),
 $mlength$,
 '4)))))))))

THEOREM: good-task-buffer-properties

good-task-buffer ($buffer$, $capacity$)
 \rightarrow (plistp ($buffer$)
 \wedge ((length ($buffer$) < (1 + $capacity$))
 \wedge finite-number-listp ($buffer$, '65536)))

EVENT: Disable good-task-buffer-properties.

THEOREM: good-task-buffer-getnth

(good-task-buffer-list (l , $capacity$) \wedge ($i < \text{length}(l)$)
 \rightarrow good-task-buffer (getnth (i , l), $capacity$))

EVENT: Disable good-task-buffer-getnth.

THEOREM: plistp-good-task-buffer-list

good-task-buffer-list (l , $capacity$) \rightarrow plistp (l)

EVENT: Disable plistp-good-task-buffer-list.

THEOREM: good-task-buffer-list-putnth
(good-task-buffer-list (l , $capacity$)
 \wedge (plistp ($buffer$)
 \wedge ((length ($buffer$) $<$ (1 + $capacity$))
 \wedge finite-number-listp ($buffer$, '65536))))
 \rightarrow good-task-buffer-list (putnth ($buffer$, i , l), $capacity$)

EVENT: Disable good-task-buffer-list-putnth.

THEOREM: good-task-buffer-list-getnth
(good-task-buffer-table (l , $length$, $capacity$) \wedge ($i <$ length (l)))
 \rightarrow (plistp (getnth (i , l))
 \wedge ((length (getnth (i , l)) = $length$)
 \wedge good-task-buffer-list (getnth (i , l), $capacity$)))

EVENT: Disable good-task-buffer-list-getnth.

THEOREM: good-task-buffer-getnth2
(good-task-buffer-table (l , $length$, $capacity$)
 \wedge (($i <$ length (l) \wedge ($j <$ length)))
 \rightarrow good-task-buffer (getnth2 (i , j , l), $capacity$)

EVENT: Disable good-task-buffer-getnth2.

THEOREM: good-task-buffer-table-putnth
(good-task-buffer-table (l , $length$, $capacity$)
 \wedge (plistp (x)
 \wedge ((length (x) = $length$)
 \wedge good-task-buffer-list (x , $capacity$))))
 \rightarrow good-task-buffer-table (putnth (x , i , l), $length$, $capacity$)

EVENT: Disable good-task-buffer-table-putnth.

THEOREM: good-task-buffer-table-putnth2
(good-task-buffer-table (l , $length$, $capacity$)
 \wedge (($i <$ length (l)
 \wedge (plistp ($buffer$)
 \wedge ((length ($buffer$) $<$ (1 + $capacity$)
 \wedge finite-number-listp ($buffer$, '65536))))))
 \rightarrow good-task-buffer-table (putnth2 ($buffer$, i , j , l), $length$, $capacity$)

EVENT: Disable good-task-buffer-table-putnth2.

THEOREM: plistp-good-task-buffer-table
good-task-buffer-table(l , $length$, $capacity$) \rightarrow plistp(l)

EVENT: Disable plistp-good-task-buffer-table.

EVENT: Add the shell ak , with recognizer function symbol $ak-shellp$ and 10 accessors: $ak-pstates$, with type restriction (none-of) and default value zero; $ak-ibuffers$, with type restriction (none-of) and default value zero; $ak-obuffers$, with type restriction (none-of) and default value zero; $ak-mbuffers$, with type restriction (none-of) and default value zero; $ak-readyq$, with type restriction (none-of) and default value zero; $ak-status$, with type restriction (none-of) and default value zero; $ak-rwstate$, with type restriction (none-of) and default value zero; $ak-clock$, with type restriction (none-of) and default value zero; $ak-iports$, with type restriction (none-of) and default value zero; $ak-oports$, with type restriction (none-of) and default value zero.

EVENT: Let us define the theory $ak-shell$ to consist of the following events: ak .

DEFINITION: $AK-TASKIDLUB = '16$

DEFINITION: $AK-TIME-SLICE = '1000$

DEFINITION: $ak-taskid(ak) = qfirst(ak-readyq(ak))$

DEFINITION:

$ak-send-instructionp(ak)$
 $= ((tm-svcid(getnth(ak-taskid(ak), ak-pstates(ak)))) \bmod '4) = '0)$

DEFINITION:

$ak-receive-instructionp(ak)$
 $= ((tm-svcid(getnth(ak-taskid(ak), ak-pstates(ak)))) \bmod '4) = '1)$

DEFINITION:

$ak-tyo-instructionp(ak)$
 $= ((tm-svcid(getnth(ak-taskid(ak), ak-pstates(ak)))) \bmod '4) = '2)$

DEFINITION:

$ak-tyi-instructionp(ak)$
 $= ((tm-svcid(getnth(ak-taskid(ak), ak-pstates(ak)))) \bmod '4) = '3)$

DEFINITION:

$ak-update-control(id, pstates)$
 $= putnth(tm-set-svcflag('0, getnth(id, pstates)), id, pstates)$

DEFINITION:

ak-communicationp (*ak*)
= tm-svc-interruptp (getnth (ak-taskid (*ak*), ak-pstates (*ak*)))

DEFINITION:

ak-errorp (*ak*) = tm-errorp (getnth (ak-taskid (*ak*), ak-pstates (*ak*)))

DEFINITION:

ak-svc-interruptp (*ak*)
= tm-svc-interruptp (getnth (ak-taskid (*ak*), ak-pstates (*ak*)))

DEFINITION:

ak-destid (*ak*) = os-destid (getnth (ak-taskid (*ak*), ak-pstates (*ak*)), '16)

DEFINITION:

ak-srcid (*ak*) = os-srcid (getnth (ak-taskid (*ak*), ak-pstates (*ak*)), '16)

DEFINITION:

ak-idevid (*ak*) = os-idevid (getnth (ak-taskid (*ak*), ak-pstates (*ak*)), '16)

DEFINITION:

ak-odevid (*ak*) = os-odevid (getnth (ak-taskid (*ak*), ak-pstates (*ak*)), '16)

DEFINITION:

ak-message (*ak*) = os-message (getnth (ak-taskid (*ak*), ak-pstates (*ak*)))

DEFINITION:

ak-store-message (*msg*, *taskid*, *pstates*)
= putnth (os-store-message (*msg*, getnth (*taskid*, *pstates*)), *taskid*, *pstates*)

DEFINITION:

ak-fetch-execute (*id*, *pstates*)
= putnth (tm-fetch-execute (getnth (*id*, *pstates*)), *id*, *pstates*)

DEFINITION:

ak-oport-idlep (*id*, *oport*) = tm-oport-idlep (*id*, *oport*)

DEFINITION:

ak-post-output-interrupt (*id*, *oport*) = tm-post-output-interrupt (*id*, *oport*)

DEFINITION:

ak-clear-input-interrupt (*id*, *oport*) = tm-clear-input-interrupt (*id*, *oport*)

DEFINITION:

ak-clear-output-interrupt (*id*, *oport*)
= tm-clear-output-interrupt (*id*, *oport*)

DEFINITION:

$\text{ak-start-output}(char, id, oports) = \text{tm-start-output}(char, id, oports)$

DEFINITION:

$\text{ak-input-interruptp}(ak) = \text{tm-some-input-interruptp}(\text{ak-iports}(ak))$

DEFINITION:

$\text{ak-output-interruptp}(ak) = \text{tm-some-output-interruptp}(\text{ak-oports}(ak))$

DEFINITION:

$\text{ak-interrupting-input-port}(iports) = \text{tm-interrupting-input-port}(iports)$

DEFINITION:

$\text{ak-interrupting-output-port}(oports) = \text{tm-interrupting-output-port}(oports)$

DEFINITION: $\text{AK-STATUS-FLAG-FIELD} = '0$

DEFINITION: $\text{AK-STATUS-TASKID-FIELD} = '1$

DEFINITION: $\text{AK-READY-STATUS} = '0$

DEFINITION: $\text{AK-ERROR-STATUS} = '1$

DEFINITION: $\text{AK-SEND-STATUS} = '2$

DEFINITION: $\text{AK-RECEIVE-STATUS} = '3$

DEFINITION: $\text{AK-OUTPUT-STATUS} = '4$

DEFINITION: $\text{AK-INPUT-STATUS} = '5$

DEFINITION: $\text{AK-STATUS-LUB} = '6$

DEFINITION: $\text{AK-STATUS-LENGTH} = '2$

DEFINITION:

$\text{ak-ready-statusp}(taskid, ak)$
 $= (\text{getnth}('0, \text{getnth}(taskid, \text{ak-status}(ak))) = '0)$

DEFINITION:

$\text{ak-waiting-to-sendp}(srcid, destid, ak)$
 $= ((\text{getnth}('0, \text{getnth}(srcid, \text{ak-status}(ak))) = '2)$
 $\quad \wedge (\text{getnth}('1, \text{getnth}(srcid, \text{ak-status}(ak))) = destid))$

DEFINITION:

$\text{ak-waiting-to-receivep}(srcid, destid, ak)$
 $= ((\text{getnth}('0, \text{getnth}(destid, \text{ak-status}(ak))) = '3)$
 $\quad \wedge (\text{getnth}('1, \text{getnth}(destid, \text{ak-status}(ak))) = srcid))$

DEFINITION:

```
ak-waiting-to-outputp(id, ak)  
= (getnth('0, getnth(id, ak-status(ak))) = '4)
```

DEFINITION:

```
ak-waiting-to-inputp(id, ak)  
= (getnth('0, getnth(id, ak-status(ak))) = '5)
```

DEFINITION: AK-WAIT-STATE = '1

DEFINITION: AK-RUN-STATE = '0

DEFINITION: ak-waiting(*ak*) = (ak-rwstate(*ak*) = '1)

DEFINITION:

```
ak-dispatcher(ak)  
= ak(ak-pstates(ak),  
      ak-ibuffers(ak),  
      ak-obuffers(ak),  
      ak-mbuffers(ak),  
      ak-readyq(ak),  
      ak-status(ak),  
      if qempty(ak-readyq(ak)) then '1  
      else '0 endif,  
      if qempty(ak-readyq(ak)) then ak-clock(ak)  
      else '1000 endif,  
      ak-iports(ak),  
      ak-oports(ak))
```

DEFINITION:

```
ak-clock-interrupt-handler(ak)  
= ak-dispatcher(ak(ak-pstates(ak),  
                  ak-ibuffers(ak),  
                  ak-obuffers(ak),  
                  ak-mbuffers(ak),  
                  enq(ak-taskid(ak), deq(ak-readyq(ak))),  
                  ak-status(ak),  
                  ak-rwstate(ak),  
                  ak-clock(ak),  
                  ak-iports(ak),  
                  ak-oports(ak)))
```

DEFINITION:

```
ak-error-handler(ak)  
= ak-dispatcher(ak(ak-pstates(ak),
```

```

ak-ibuffers (ak),
ak-obuffers (ak),
ak-mbuffers (ak),
deq (ak-readyq (ak)),
putnth (' (1 0), ak-taskid (ak), ak-status (ak)),
ak-rwstate (ak),
ak-clock (ak),
ak-iports (ak),
ak-oports (ak))

```

DEFINITION:

```

ak-block-send (srcid, destid, ak)
= ak-dispatcher (ak (ak-pstates (ak),
                    ak-ibuffers (ak),
                    ak-obuffers (ak),
                    ak-mbuffers (ak),
                    deq (ak-readyq (ak)),
                    putnth (cons ('2, cons (destid, 'nil)),
                            srcid,
                            ak-status (ak)),
                    ak-rwstate (ak),
                    ak-clock (ak),
                    ak-iports (ak),
                    ak-oports (ak)))

```

DEFINITION:

```

ak-execute-send-to-buffer (msg, srcid, destid, ak)
= ak (ak-update-control (srcid, ak-pstates (ak)),
      ak-ibuffers (ak),
      ak-obuffers (ak),
      enq2 (msg, srcid, destid, ak-mbuffers (ak)),
      if ak-waiting-to-receivep (srcid, destid, ak)
      then enq (destid, ak-readyq (ak))
      else ak-readyq (ak) endif,
      if ak-waiting-to-receivep (srcid, destid, ak)
      then putnth (' (0 0), destid, ak-status (ak))
      else ak-status (ak) endif,
      ak-rwstate (ak),
      ak-clock (ak),
      ak-iports (ak),
      ak-oports (ak))

```

DEFINITION:

```

ak-execute-send (msg, srcid, destid, ak)
= if qfullp2 (srcid, destid, ak-mbuffers (ak), '4)

```

```

then ak-block-send (srcid, destid, ak)
else ak-execute-send-to-buffer (msg, srcid, destid, ak) endif

```

DEFINITION:

```

ak-block-receive (srcid, destid, ak)
= ak-dispatcher (ak (ak-pstates (ak),
                    ak-ibuffers (ak),
                    ak-obuffers (ak),
                    ak-mbuffers (ak),
                    deq (ak-readyq (ak)),
                    putnth (cons ('3, cons (srcid, 'nil)),
                            destid,
                            ak-status (ak)),
                    ak-rwstate (ak),
                    ak-clock (ak),
                    ak-iports (ak),
                    ak-oports (ak)))

```

DEFINITION:

```

ak-execute-receive-from-buffer (srcid, destid, ak)
= ak (ak-update-control (destid,
                        ak-store-message (qfirst2 (srcid,
                                                  destid,
                                                  ak-mbuffers (ak)),
                                                  destid,
                                                  ak-pstates (ak))),
      ak-ibuffers (ak),
      ak-obuffers (ak),
      deq2 (srcid, destid, ak-mbuffers (ak)),
      if ak-waiting-to-sendp (srcid, destid, ak)
      then enq (srcid, ak-readyq (ak))
      else ak-readyq (ak) endif,
      if ak-waiting-to-sendp (srcid, destid, ak)
      then putnth ('(0 0), srcid, ak-status (ak))
      else ak-status (ak) endif,
      ak-rwstate (ak),
      ak-clock (ak),
      ak-iports (ak),
      ak-oports (ak))

```

DEFINITION:

```

ak-execute-receive (srcid, destid, ak)
= if qempty2 (srcid, destid, ak-mbuffers (ak))
  then ak-block-receive (srcid, destid, ak)
  else ak-execute-receive-from-buffer (srcid, destid, ak) endif

```

DEFINITION:

```
ak-block-output (id, ak)
= ak-dispatcher (ak (ak-pstates (ak),
                    ak-ibuffers (ak),
                    ak-obuffers (ak),
                    ak-mbuffers (ak),
                    deq (ak-readyq (ak)),
                    putnth ('(4 0), id, ak-status (ak)),
                    ak-rwstate (ak),
                    ak-clock (ak),
                    ak-iports (ak),
                    ak-oports (ak)))
```

DEFINITION:

```
ak-execute-output-to-buffer (char, id, ak)
= ak (ak-update-control (id, ak-pstates (ak)),
     ak-ibuffers (ak),
     enq-ith-buffer (char, id, ak-obuffers (ak)),
     ak-mbuffers (ak),
     ak-readyq (ak),
     ak-status (ak),
     ak-rwstate (ak),
     ak-clock (ak),
     ak-iports (ak),
     if ak-oport-idle (id, ak-oports (ak))
     then ak-post-output-interrupt (id, ak-oports (ak))
     else ak-oports (ak) endif)
```

DEFINITION:

```
ak-execute-output (char, id, ak)
= if qfullp (getnth (id, ak-obuffers (ak)), '4) then ak-block-output (id, ak)
  else ak-execute-output-to-buffer (char, id, ak) endif
```

DEFINITION:

```
ak-block-input (id, ak)
= ak-dispatcher (ak (ak-pstates (ak),
                    ak-ibuffers (ak),
                    ak-obuffers (ak),
                    ak-mbuffers (ak),
                    deq (ak-readyq (ak)),
                    putnth ('(5 0), id, ak-status (ak)),
                    ak-rwstate (ak),
                    ak-clock (ak),
                    ak-iports (ak),
                    ak-oports (ak)))
```

DEFINITION:

```
ak-execute-input-from-buffer (id, ak)
= ak (ak-update-control (id,
                        ak-store-message (qfirst (getnth (id,
                                                         ak-ibuffers (ak))),
                                           id,
                                           ak-pstates (ak))),
      putnth (deq (getnth (id, ak-ibuffers (ak))), id, ak-ibuffers (ak)),
      ak-obuffers (ak),
      ak-mbuffers (ak),
      ak-readyq (ak),
      ak-status (ak),
      ak-rwstate (ak),
      ak-clock (ak),
      ak-iports (ak),
      ak-oports (ak))
```

DEFINITION:

```
ak-execute-input (id, ak)
= if qempty (getnth (id, ak-ibuffers (ak))) then ak-block-input (id, ak)
  else ak-execute-input-from-buffer (id, ak) endif
```

DEFINITION:

```
ak-svc-handler (ak)
= if ak-send-instructionp (ak)
  then ak-execute-send (ak-message (ak), ak-taskid (ak), ak-destid (ak), ak)
  elseif ak-receive-instructionp (ak)
  then ak-execute-recv (ak-srcid (ak), ak-taskid (ak), ak)
  elseif ak-tyo-instructionp (ak)
  then ak-execute-output (ak-message (ak), ak-taskid (ak), ak)
  else ak-execute-input (ak-taskid (ak), ak) endif
```

DEFINITION:

```
ak-private-step (ak)
= ak (ak-fetch-execute (ak-taskid (ak), ak-pstates (ak)),
      ak-ibuffers (ak),
      ak-obuffers (ak),
      ak-mbuffers (ak),
      ak-readyq (ak),
      ak-status (ak),
      ak-rwstate (ak),
      ak-clock (ak) - 1,
      ak-iports (ak),
      ak-oports (ak))
```

DEFINITION:

```
ak-update-ibuffer (id, ak)
=  if qfullp (getnth (id, ak-ibuffers (ak)), '4)
    then qreplace-ith-buffer (tm-overflow-char (tm-ichar (getnth (id,
                                                                    ak-iports (ak))),
                                                                    id,
                                                                    ak-ibuffers (ak))
    elseif tm-iport-errorp (id, ak-iports (ak))
    then enq-ith-buffer (tm-overflow-char (tm-ichar (getnth (id, ak-iports (ak))),
                                                                    id,
                                                                    ak-ibuffers (ak))
    else enq-ith-buffer (tm-ichar (getnth (id, ak-iports (ak))),
                                                                    id,
                                                                    ak-ibuffers (ak)) endif
```

DEFINITION:

```
ak-waiting-input-interrupt-handler (id, ak)
=  ak-dispatcher (ak (ak-pstates (ak),
                    ak-update-ibuffer (id, ak),
                    ak-obuffers (ak),
                    ak-mbuffers (ak),
                    if ak-waiting-to-inputp (id, ak)
                    then enq (id, ak-readyq (ak))
                    else ak-readyq (ak) endif,
                    if ak-waiting-to-inputp (id, ak)
                    then putnth ('(0 0), id, ak-status (ak))
                    else ak-status (ak) endif,
                    ak-rwstate (ak),
                    ak-clock (ak),
                    ak-clear-input-interrupt (id, ak-iports (ak)),
                    ak-oports (ak)))
```

DEFINITION:

```
ak-running-input-interrupt-handler (id, ak)
=  ak (ak-pstates (ak),
    ak-update-ibuffer (id, ak),
    ak-obuffers (ak),
    ak-mbuffers (ak),
    if ak-waiting-to-inputp (id, ak) then enq (id, ak-readyq (ak))
    else ak-readyq (ak) endif,
    if ak-waiting-to-inputp (id, ak)
    then putnth ('(0 0), id, ak-status (ak))
    else ak-status (ak) endif,
    ak-rwstate (ak),
```

```

    ak-clock (ak),
    ak-clear-input-interrupt (id, ak-iports (ak)),
    ak-oports (ak)

```

DEFINITION:

```

ak-input-interrupt-handler (id, ak)
=  if ak-waiting (ak) then ak-waiting-input-interrupt-handler (id, ak)
    else ak-running-input-interrupt-handler (id, ak) endif

```

DEFINITION:

```

ak-waiting-output-interrupt-handler (id, ak)
=  ak-dispatcher (ak (ak-pstates (ak),
    ak-ibuffers (ak),
    if qempty (getnth (id, ak-obuffers (ak)))
    then ak-obuffers (ak)
    else deq-ith-buffer (id, ak-obuffers (ak)) endif,
    ak-mbuffers (ak),
    if ak-waiting-to-outputp (id, ak)
    then enq (id, ak-readyq (ak))
    else ak-readyq (ak) endif,
    if ak-waiting-to-outputp (id, ak)
    then putnth ('(0 0), id, ak-status (ak))
    else ak-status (ak) endif,
    ak-rwstate (ak),
    ak-clock (ak),
    ak-iports (ak),
    if qempty (getnth (id, ak-obuffers (ak)))
    then ak-clear-output-interrupt (id, ak-oports (ak))
    else ak-start-output (qfirst (getnth (id,
    ak-obuffers (ak))),
    id,
    ak-oports (ak)) endif)

```

DEFINITION:

```

ak-running-output-interrupt-handler (id, ak)
=  ak (ak-pstates (ak),
    ak-ibuffers (ak),
    if qempty (getnth (id, ak-obuffers (ak))) then ak-obuffers (ak)
    else deq-ith-buffer (id, ak-obuffers (ak)) endif,
    ak-mbuffers (ak),
    if ak-waiting-to-outputp (id, ak) then enq (id, ak-readyq (ak))
    else ak-readyq (ak) endif,
    if ak-waiting-to-outputp (id, ak)
    then putnth ('(0 0), id, ak-status (ak))
    else ak-status (ak) endif,

```



```

ak-rwstate (ak),
ak-clock (ak),
ak-iports (ak),
if qempty (getnth (id, ak-obuffers (ak)))
then ak-clear-output-interrupt (id, ak-oports (ak))
else ak-start-output (qfirst (getnth (id, ak-obuffers (ak))),
                       id,
                       ak-oports (ak)) endif

```

DEFINITION:

```

ak-output-interrupt-handler (id, ak)
= if ak-waiting (ak) then ak-waiting-output-interrupt-handler (id, ak)
  else ak-running-output-interrupt-handler (id, ak) endif

```

DEFINITION: $\text{ak-clock-interruptp} (ak) = (\text{ak-clock} (ak) \simeq 0)$

DEFINITION:

```

ak-step (ak)
= if ak-input-interruptp (ak)
  then ak-input-interrupt-handler (ak-interrupting-input-port (ak-iports (ak)),
                                   ak)
  elseif ak-output-interruptp (ak)
  then ak-output-interrupt-handler (ak-interrupting-output-port (ak-oports (ak)),
                                   ak)
  elseif ak-waiting (ak) then ak
  elseif ak-errorp (ak) then ak-error-handler (ak)
  elseif ak-clock-interruptp (ak) then ak-clock-interrupt-handler (ak)
  elseif ak-svc-interruptp (ak) then ak-svc-handler (ak)
  else ak-private-step (ak) endif

```

DEFINITION:

```

ak-post-interrupt (interrupt, ak)
= ak (ak-pstates (ak),
      ak-ibuffers (ak),
      ak-obuffers (ak),
      ak-mbuffers (ak),
      ak-readyq (ak),
      ak-status (ak),
      ak-rwstate (ak),
      ak-clock (ak),
      if tm-device-input-eventp (interrupt)
      then tm-post-input-interrupt (tm-idatum (interrupt) mod '256,
                                   tm-idevid (interrupt) mod '16,
                                   ak-iports (ak))
      else ak-iports (ak) endif,

```

```

if tm-device-output-eventp (interrupt)
then tm-post-output-interrupt (tm-odevid (interrupt) mod '16,
                               ak-oportp (ak))
else ak-oportp (ak) endif

```

DEFINITION:

```

ak-processor (ak, oracle)
= if listp (oracle)
  then ak-processor (ak-step (ak-post-interrupt (car (oracle), ak)),
                    cdr (oracle))
  else ak endif

```

THEOREM: good-tm-iport-array-putnth1

```

(good-tm-iport-array (l)
  $\wedge$  (tm-iportp (iport)
       $\wedge$  (finite-numberp (tm-iinterrupt-flag (iport), '2)
           $\wedge$  (finite-numberp (tm-ierror-flag (iport), '2)
               $\wedge$  finite-numberp (tm-ichar (iport), '256))))))
 $\rightarrow$  good-tm-iport-array (putnth (iport, n, l))

```

EVENT: Disable good-tm-iport-array-putnth1.

THEOREM: good-tm-oport-array-putnth1

```

(good-tm-oport-array (l)
  $\wedge$  (tm-oportp (oport)
       $\wedge$  (finite-numberp (tm-ointerrupt-flag (oport), '2)
           $\wedge$  (finite-numberp (tm-obusy-flag (oport), '2)
               $\wedge$  finite-numberp (tm-ochar (oport), '256))))))
 $\rightarrow$  good-tm-oport-array (putnth (oport, n, l))

```

EVENT: Disable good-tm-oport-array-putnth1.

DEFINITION:

```

good-address-space-list (l)
= if listp (l)
  then good-address-space (car (l), length (tm-memory (car (l))))
     $\wedge$  good-address-space-list (cdr (l))
  else l = 'nil endif

```

THEOREM: plistp-good-address-space-list

```

good-address-space-list (l)  $\rightarrow$  plistp (l)

```

EVENT: Disable plistp-good-address-space-list.

THEOREM: good-address-space-getnth
 (good-address-space-list (l) \wedge ($i < \text{length}(l)$))
 \rightarrow good-address-space (getnth (i, l), length (tm-memory (getnth (i, l))))

EVENT: Disable good-address-space-getnth.

THEOREM: good-address-space-list-putnth
 (good-address-space-list (l)
 \wedge good-address-space ($task$, length (tm-memory ($task$))))
 \rightarrow good-address-space-list (putnth ($task, i, l$))

EVENT: Disable good-address-space-list-putnth.

DEFINITION:
 good-status ($task\text{-}status$)
 = (plistp ($task\text{-}status$)
 \wedge ((length ($task\text{-}status$) = '2)
 \wedge (finite-numberp (getnth ('0, $task\text{-}status$), '6)
 \wedge finite-numberp (getnth ('1, $task\text{-}status$), '16))))

DEFINITION:
 good-status-list ($status$)
 = **if** listp ($status$)
 then good-status (car ($status$)) \wedge good-status-list (cdr ($status$))
 else $status = \text{'nil}$ **endif**

THEOREM: plistp-good-status-list
 good-status-list (l) \rightarrow plistp (l)

EVENT: Disable plistp-good-status-list.

THEOREM: good-status-getnth
 (good-status-list (l) \wedge ($i < \text{length}(l)$)) \rightarrow good-status (getnth (i, l))

EVENT: Disable good-status-getnth.

THEOREM: good-status-list-putnth
 (good-status-list (l)
 \wedge (plistp ($status$)
 \wedge ((length ($status$) = '2)
 \wedge (finite-numberp (getnth ('0, $status$), '6)
 \wedge finite-numberp (getnth ('1, $status$), '16))))))
 \rightarrow good-status-list (putnth ($status, i, l$))

EVENT: Disable good-status-list-putnth.

EVENT: Let us define the theory *good-ak-construction* to consist of the following events: good-task-buffer-table-putnth2, good-task-buffer-table-putnth, good-task-buffer-getnth2, good-task-buffer-list-getnth, good-task-buffer-list-putnth, plistp-good-task-buffer-list, good-task-buffer-getnth, good-status-list-putnth, good-status-getnth, plistp-good-status-list, good-tm-iport-getnth, good-tm-oport-getnth, good-tm-iport-array-putnth1, good-tm-oport-array-putnth1, good-address-space-getnth, good-address-space-list-putnth, plistp-good-address-space-list, a-finite-number-is-lessp-than-its-lub, length-putnth, length-putnth2, plistp-putnth, plistp-putnth2.

DEFINITION:

status-flag-readyp (*task-status*) = (getnth ('0, *task-status*) = '0)

DEFINITION:

index-ready-set (*n*, *status*)
= **if** listp (*status*)
 then if status-flag-readyp (car (*status*))
 then cons (fix (*n*), index-ready-set (1 + *n*, cdr (*status*)))
 else index-ready-set (1 + *n*, cdr (*status*)) **endif**
 else 'nil **endif**

DEFINITION:

ak-ready-set (*ak*) = index-ready-set ('0, ak-status (*ak*))

```
(DEFN GOOD-AK
  (AK)
  (AND
    (AK-SHELLP AK)
    (AND
      (EQUAL (LENGTH (AK-PSTATES AK)) '16)
      (AND
        (GOOD-ADDRESS-SPACE-LIST (AK-PSTATES AK))
        (AND
          (EQUAL (LENGTH (AK-IBUFFERS AK)) '16)
          (AND
            (GOOD-TASK-BUFFER-LIST (AK-IBUFFERS AK)
                                   '4)
            (AND
              (EQUAL (LENGTH (AK-OBUFFERS AK)) '16)
              (AND
```

```

(GOOD-TASK-BUFFER-LIST (AK-OBUFFERS AK)
                        '4)
(AND
 (EQUAL (LENGTH (AK-MBUFFERS AK)) '16)
 (AND
  (GOOD-TASK-BUFFER-TABLE (AK-MBUFFERS AK)
                           '16
                           '4)
  (AND
   (PLISTP (AK-READYQ AK))
   (AND
    (LESSP (LENGTH (AK-READYQ AK)) '17)
    (AND
     (FINITE-NUMBER-LISTP (AK-READYQ AK)
                           '16)
     (AND
      (EQUAL (LENGTH (AK-STATUS AK)) '16)
      (AND
       (GOOD-STATUS-LIST (AK-STATUS AK))
       (AND
        (FINITE-NUMBERP (AK-RWSTATE AK) '2)
        (AND
         (FINITE-NUMBERP (AK-CLOCK AK) '65536)
         (AND
          (PLISTP (AK-IPOINTS AK))
          (AND
           (EQUAL (LENGTH (AK-IPOINTS AK)) '16)
           (AND
            (GOOD-TM-IPOINT-ARRAY (AK-IPOINTS AK))
            (AND
             (PLISTP (AK-OPOINTS AK))
             (AND
              (EQUAL (LENGTH (AK-OPOINTS AK)) '16)
              (AND
               (GOOD-TM-OPOINT-ARRAY (AK-OPOINTS AK))
               (AND
                (PERMUTATION (AK-READYQ AK)
                             (AK-READY-SET AK))
                (IFF (AK-WAITING AK)
                     (QEMPTYP (AK-READYQ AK))))))))))))))))))))))))))
NIL)

```

THEOREM: ak-shellp-ak
good-ak(*ak*) → ak-shellp(*ak*)

EVENT: Disable ak-shellp-ak.

THEOREM: length-ak-pstates
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-pstates}(ak)) = '16)$

EVENT: Disable length-ak-pstates.

THEOREM: good-address-space-list-ak-pstates
 $\text{good-ak}(ak) \rightarrow \text{good-address-space-list}(\text{ak-pstates}(ak))$

EVENT: Disable good-address-space-list-ak-pstates.

THEOREM: good-address-space-getnth-from-ak
 $(\text{good-ak}(ak) \wedge (i < '16))$
 $\rightarrow \text{good-address-space}(\text{getnth}(i, \text{ak-pstates}(ak)),$
 $\text{length}(\text{tm-memory}(\text{getnth}(i, \text{ak-pstates}(ak))))))$

EVENT: Disable good-address-space-getnth-from-ak.

THEOREM: length-ak-ibuffers
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-ibuffers}(ak)) = '16)$

EVENT: Disable length-ak-ibuffers.

THEOREM: good-task-buffer-list-ak-ibuffers
 $\text{good-ak}(ak) \rightarrow \text{good-task-buffer-list}(\text{ak-ibuffers}(ak), '4)$

EVENT: Disable good-task-buffer-list-ak-ibuffers.

THEOREM: plistp-ak-ibuffers
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-ibuffers}(ak))$

EVENT: Disable plistp-ak-ibuffers.

THEOREM: good-task-buffer-getnth-from-ak-ibuffers
 $(\text{good-ak}(ak) \wedge \text{finite-numberp}(id, '16))$
 $\rightarrow \text{good-task-buffer}(\text{getnth}(id, \text{ak-ibuffers}(ak)), '4)$

EVENT: Disable good-task-buffer-getnth-from-ak-ibuffers.

THEOREM: length-ak-obuffers
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-obuffers}(ak)) = '16)$

EVENT: Disable length-ak-obuffers.

THEOREM: good-task-buffer-list-ak-obuffers
 $\text{good-ak}(ak) \rightarrow \text{good-task-buffer-list}(\text{ak-obuffers}(ak), '4)$

EVENT: Disable good-task-buffer-list-ak-obuffers.

THEOREM: plistp-ak-obuffers
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-obuffers}(ak))$

EVENT: Disable plistp-ak-obuffers.

THEOREM: good-task-buffer-getnth-from-ak-obuffers
 $(\text{good-ak}(ak) \wedge \text{finite-numberp}(id, '16))$
 $\rightarrow \text{good-task-buffer}(\text{getnth}(id, \text{ak-obuffers}(ak)), '4)$

EVENT: Disable good-task-buffer-getnth-from-ak-obuffers.

THEOREM: length-ak-mbuffers
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-mbuffers}(ak)) = '16)$

EVENT: Disable length-ak-mbuffers.

THEOREM: good-task-buffer-table-ak-mbuffers
 $\text{good-ak}(ak) \rightarrow \text{good-task-buffer-table}(\text{ak-mbuffers}(ak), '16, '4)$

EVENT: Disable good-task-buffer-table-ak-mbuffers.

THEOREM: plistp-ak-mbuffers
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-mbuffers}(ak))$

EVENT: Disable plistp-ak-mbuffers.

THEOREM: good-task-buffer-getnth2-from-ak-mbuffers
 $(\text{good-ak}(ak) \wedge ((i < '16) \wedge (j < '16)))$
 $\rightarrow \text{good-task-buffer}(\text{getnth2}(i, j, \text{ak-mbuffers}(ak)), '4)$

EVENT: Disable good-task-buffer-getnth2-from-ak-mbuffers.

THEOREM: finite-numberp-ak-taskid
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak)))$
 $\rightarrow \text{finite-numberp}(\text{ak-taskid}(ak), '16)$

EVENT: Disable finite-numberp-ak-taskid.

THEOREM: plistp-ak-readyq
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-readyq}(ak))$

EVENT: Disable plistp-ak-readyq.

THEOREM: length-ak-readyq
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-readyq}(ak)) < '17)$

EVENT: Disable length-ak-readyq.

THEOREM: finite-number-listp-ak-readyq
 $\text{good-ak}(ak) \rightarrow \text{finite-number-listp}(\text{ak-readyq}(ak), '16)$

EVENT: Disable finite-number-listp-ak-readyq.

THEOREM: length-ak-status
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-status}(ak)) = '16)$

EVENT: Disable length-ak-status.

THEOREM: good-status-list-ak-status
 $\text{good-ak}(ak) \rightarrow \text{good-status-list}(\text{ak-status}(ak))$

EVENT: Disable good-status-list-ak-status.

THEOREM: plistp-ak-status
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-status}(ak))$

EVENT: Disable plistp-ak-status.

THEOREM: good-status-getnth-from-ak-status
 $(\text{good-ak}(ak) \wedge (i < '16)) \rightarrow \text{good-status}(\text{getnth}(i, \text{ak-status}(ak)))$

EVENT: Disable good-status-getnth-from-ak-status.

THEOREM: finite-numberp-ak-rwstate
 $\text{good-ak}(ak) \rightarrow \text{finite-numberp}(\text{ak-rwstate}(ak), '2)$

EVENT: Disable finite-numberp-ak-rwstate.

THEOREM: finite-numberp-ak-clock
 $\text{good-ak}(ak) \rightarrow \text{finite-numberp}(\text{ak-clock}(ak), '65536)$

EVENT: Disable finite-numberp-ak-clock.

THEOREM: plistp-ak-iports
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-iports}(ak))$

EVENT: Disable plistp-ak-iports.

THEOREM: length-ak-iports
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-iports}(ak)) = '16)$

EVENT: Disable length-ak-iports.

THEOREM: good-tm-iport-array-ak-iports
 $\text{good-ak}(ak) \rightarrow \text{good-tm-iport-array}(\text{ak-iports}(ak))$

EVENT: Disable good-tm-iport-array-ak-iports.

THEOREM: good-tm-iport-getnth-from-ak-iports
 $(\text{good-ak}(ak) \wedge (i < '16)) \rightarrow \text{good-tm-iport}(\text{getnth}(i, \text{ak-iports}(ak)))$

EVENT: Disable good-tm-iport-getnth-from-ak-iports.

THEOREM: plistp-ak-oports
 $\text{good-ak}(ak) \rightarrow \text{plistp}(\text{ak-oports}(ak))$

EVENT: Disable plistp-ak-oports.

THEOREM: length-ak-oports
 $\text{good-ak}(ak) \rightarrow (\text{length}(\text{ak-oports}(ak)) = '16)$

EVENT: Disable length-ak-oports.

THEOREM: good-tm-oport-array-ak-oports
 $\text{good-ak}(ak) \rightarrow \text{good-tm-oport-array}(\text{ak-oports}(ak))$

EVENT: Disable good-tm-oport-array-ak-oports.

THEOREM: good-tm-oport-getnth-from-ak-oports
 $(\text{good-ak}(ak) \wedge (i < '16)) \rightarrow \text{good-tm-oport}(\text{getnth}(i, \text{ak-oports}(ak)))$

EVENT: Disable good-tm-oport-getnth-from-ak-oports.

THEOREM: permutation-ak-readyq-ak-ready-set
 $\text{good-ak}(ak) \rightarrow \text{permutation}(\text{ak-readyq}(ak), \text{ak-ready-set}(ak))$

EVENT: Disable permutation-ak-readyq-ak-ready-set.

THEOREM: ak-waiting-iff-qempty-ak-readyq
 $\text{good-ak}(ak) \rightarrow (\text{ak-waiting}(ak) \leftrightarrow \text{qempty}(\text{ak-readyq}(ak)))$

EVENT: Disable ak-waiting-iff-qempty-ak-readyq.

EVENT: Let us define the theory *good-ak-properties* to consist of the following events: ak-waiting-iff-qempty-ak-readyq, permutation-ak-readyq-ak-ready-set, good-tm-oport-getnth-from-ak-oports, good-tm-oport-array-ak-oports, length-ak-oports, plistp-ak-oports, good-tm-iport-getnth-from-ak-iports, good-tm-iport-array-ak-iports, length-ak-iports, plistp-ak-iports, finite-numberp-ak-clock, finite-numberp-ak-rwstate, good-status-getnth-from-ak-status, plistp-ak-status, good-status-list-ak-status, length-ak-status, finite-number-listp-ak-readyq, length-ak-readyq, plistp-ak-readyq, finite-numberp-ak-taskid, good-task-buffer-getnth2-from-ak-mbuffers, plistp-ak-mbuffers, plistp-good-task-buffer-table, good-task-buffer-table-ak-mbuffers, length-ak-mbuffers, good-task-buffer-getnth-from-ak-obuffers, plistp-ak-obuffers, good-task-buffer-list-ak-obuffers, length-ak-obuffers, good-task-buffer-getnth-from-ak-ibuffers, plistp-ak-ibuffers, good-task-buffer-list-ak-ibuffers, length-ak-ibuffers, good-address-space-getnth-from-ak, good-address-space-list-ak-pstates, length-ak-pstates, ak-shellp-ak.

EVENT: Let us define the theory *queue-facts* to consist of the following events: not-qempty-enq, not-qfullp-deq, length-enq, length-deq, length-qreplace, qfirst-enq, plistp-enq, plistp-deq, plistp-qreplace, finite-numberp-qfirst, finite-number-listp-enq, finite-number-listp-deq, finite-number-listp-qreplace.

THEOREM: not-member-index-ready-set
 $(i < n) \rightarrow (i \notin \text{index-ready-set}(n, \text{status}))$

EVENT: Disable not-member-index-ready-set.

THEOREM: setp-index-ready-set
 $\text{setp}(\text{index-ready-set}(n, \text{status}))$

EVENT: Disable setp-index-ready-set.

DEFINITION:

set-status-induction (i, n, l)

= **if** $i \simeq 0$ **then** '0
 elseif listp (l) **then** set-status-induction ($i - 1, 1 + n, \text{cdr } (l)$)
 else '0 **endif**

THEOREM: index-ready-set-putnth-unready-status

$((i < \text{length } (l))$
 $\wedge (\text{status-flag-readyp } (\text{getnth } (i, l)) \wedge (\neg \text{status-flag-readyp } (v))))$
 $\rightarrow (\text{index-ready-set } (n, \text{putnth } (v, i, l))$
 $= \text{remove } (n + i, \text{index-ready-set } (n, l))$)

EVENT: Disable index-ready-set-putnth-unready-status.

THEOREM: not-status-flag-readyp-list

$(\text{flag} \neq '0) \rightarrow (\neg \text{status-flag-readyp } (\text{cons } (\text{flag}, \text{cons } (\text{taskid}, 'nil))))$

EVENT: Disable not-status-flag-readyp-list.

THEOREM: status-flag-readyp-iff-member-index-ready-set1

$(i < \text{length } (\text{status}))$
 $\rightarrow (\text{status-flag-readyp } (\text{getnth } (i, \text{status}))$
 $= ((n + i) \in \text{index-ready-set } (n, \text{status}))$)

THEOREM: status-flag-readyp-iff-member-index-ready-set

$((i \in \mathbf{N}) \wedge (i < \text{length } (\text{status})))$
 $\rightarrow (\text{status-flag-readyp } (\text{getnth } (i, \text{status}))$
 $= (i \in \text{index-ready-set } ('0, \text{status}))$)

EVENT: Disable status-flag-readyp-iff-member-index-ready-set.

THEOREM: status-flag-readyp-getnth

$(\text{permutation } (\text{queue}, \text{index-ready-set } ('0, \text{status}))$
 $\wedge ((\neg \text{qemptyp } (\text{queue}))$
 $\wedge (\text{number-listp } (\text{queue}) \wedge (\text{qfirst } (\text{queue}) < \text{length } (\text{status}))))$
 $\rightarrow \text{status-flag-readyp } (\text{getnth } (\text{qfirst } (\text{queue}), \text{status}))$

EVENT: Disable status-flag-readyp-getnth.

THEOREM: permutation-deq-index-ready-set-putnth

$(\text{permutation } (\text{queue}, \text{index-ready-set } ('0, \text{status}))$
 $\wedge (\text{number-listp } (\text{queue}))$
 $\wedge (\text{plistp } (\text{queue}))$

$$\begin{aligned} & \wedge ((\neg \text{qempty}(queue)) \\ & \quad \wedge ((\text{taskid} = \text{qfirst}(queue)) \\ & \quad \quad \wedge ((\text{flag} \neq '0) \\ & \quad \quad \quad \wedge (\text{taskid} < \text{length}(status)))))) \\ \rightarrow & \text{permutation}(\text{deq}(queue), \\ & \quad \text{index-ready-set}('0, \\ & \quad \quad \text{putnth}(\text{cons}(\text{flag}, \text{cons}(x, 'nil)), \\ & \quad \quad \quad \text{taskid}, \\ & \quad \quad \quad \text{status}))) \end{aligned}$$

EVENT: Disable permutation-deq-index-ready-set-putnth.

DEFINITION:

```
number-of-ready-tasks (status)
=  if listp (status)
    then if status-flag-readyp (car (status))
          then 1 + number-of-ready-tasks (cdr (status))
          else number-of-ready-tasks (cdr (status)) endif
    else '0 endif
```

DEFINITION:

```
number-of-unready-tasks (status)
=  if listp (status)
    then if status-flag-readyp (car (status))
          then number-of-unready-tasks (cdr (status))
          else 1 + number-of-unready-tasks (cdr (status)) endif
    else '0 endif
```

THEOREM: equal-length-status-sum-of-ready-and-unready-tasks

```
length (status)
= (number-of-ready-tasks (status) + number-of-unready-tasks (status))
```

THEOREM: leq-number-of-ready-tasks-length-status

```
if length (status) < number-of-ready-tasks (status) then '*1*false
else '*1*true endif
```

THEOREM: number-of-unready-tasks-non-zero

```
((i < length (status)) ∧ (¬ status-flag-readyp (getnth (i, status))))
→ ('0 < number-of-unready-tasks (status))
```

EVENT: Disable number-of-unready-tasks-non-zero.

THEOREM: not-equal-number-of-ready-tasks-length-status

```
((i < length (status)) ∧ (¬ status-flag-readyp (getnth (i, status))))
→ (number-of-ready-tasks (status) ≠ length (status))
```

THEOREM: lessp-number-of-ready-tasks-length-status
 $((i < \text{length}(status)) \wedge (\neg \text{status-flag-readyp}(\text{getnth}(i, status))))$
 $\rightarrow (\text{number-of-ready-tasks}(status) < \text{length}(status))$

THEOREM: number-of-ready-tasks-equals-length-index-ready-set
 $\text{number-of-ready-tasks}(status) = \text{length}(\text{index-ready-set}(n, status))$

THEOREM: status-flag-readyp-list
 $\text{status-flag-readyp}(\text{cons}('0, \text{cons}(x, 'nil)))$

EVENT: Disable status-flag-readyp-list.

THEOREM: lessp-car-index-ready-set
 $((i < n) \wedge \text{listp}(\text{index-ready-set}(n, l)))$
 $\rightarrow (i < \text{car}(\text{index-ready-set}(n, l)))$

THEOREM: lessp-car-index-ready-set-instance
 $\text{listp}(\text{index-ready-set}(1 + n, l)) \rightarrow (n < \text{car}(\text{index-ready-set}(1 + n, l)))$

EVENT: Disable lessp-car-index-ready-set-instance.

THEOREM: rewrite-insert-to-cons-crock1
 $\text{listp}(\text{index-ready-set}(1 + n, z))$
 $\rightarrow (\text{insert}(n, \text{index-ready-set}(1 + n, z))$
 $\quad = \text{cons}(n, \text{index-ready-set}(1 + n, z)))$

THEOREM: rewrite-insert-to-cons-crock2
 $(\neg \text{listp}(\text{index-ready-set}(1 + n, z)))$
 $\rightarrow (\text{insert}(n, \text{index-ready-set}(1 + n, z))$
 $\quad = \text{cons}(n, \text{index-ready-set}(1 + n, z)))$

THEOREM: rewrite-insert-to-cons-crock
 $\text{insert}(n, \text{index-ready-set}(1 + n, z)) = \text{cons}(n, \text{index-ready-set}(1 + n, z))$

EVENT: Disable rewrite-insert-to-cons-crock.

THEOREM: index-ready-set-putnth-ready-status
 $((i < \text{length}(l))$
 $\wedge ((\neg \text{status-flag-readyp}(\text{getnth}(i, l))) \wedge \text{status-flag-readyp}(v)))$
 $\rightarrow (\text{index-ready-set}(n, \text{putnth}(v, i, l))$
 $\quad = \text{insert}(n + i, \text{index-ready-set}(n, l)))$

EVENT: Disable index-ready-set-putnth-ready-status.

THEOREM: sorted-index-ready-set
 sorted (index-ready-set (n , l))

EVENT: Disable sorted-index-ready-set.

THEOREM: permutation-enq-index-ready-set-putnth
 (permutation ($queue$, index-ready-set ('0, $status$))
 \wedge ((\neg status-flag-readyp (getnth ($taskid$, $status$)))
 \wedge (plistp ($queue$)
 \wedge (($taskid \in \mathbf{N}$) \wedge ($taskid < \text{length}(status)$))))))
 \rightarrow permutation (enq ($taskid$, $queue$),
 index-ready-set ('0,
 putnth (cons ('0, cons (x , 'nil)),
 $taskid$,
 $status$)))

EVENT: Disable permutation-enq-index-ready-set-putnth.

THEOREM: number-of-ready-tasks-equals-length-ak-readyq
 permutation (ak-readyq (ak), ak-ready-set (ak))
 \rightarrow (number-of-ready-tasks (ak-status (ak)) = length (ak-readyq (ak)))

EVENT: Disable number-of-ready-tasks-equals-length-ak-readyq.

THEOREM: ak-waiting-to-sendp-implies-readyq-not-full
 (good-ak (ak)
 \wedge (ak-waiting-to-sendp ($srcid$, $destid$, ak) \wedge finite-numberp ($srcid$, '16)))
 \rightarrow (\neg qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-receivep-implies-readyq-not-full
 (good-ak (ak)
 \wedge (ak-waiting-to-receivep ($srcid$, $destid$, ak)
 \wedge finite-numberp ($destid$, '16)))
 \rightarrow (\neg qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-outputp-implies-readyq-not-full
 (good-ak (ak) \wedge (ak-waiting-to-outputp (id , ak) \wedge finite-numberp (id , '16)))
 \rightarrow (\neg qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-inputp-implies-readyq-not-full
 (good-ak (ak) \wedge (ak-waiting-to-inputp (id , ak) \wedge finite-numberp (id , '16)))
 \rightarrow (\neg qfullp (ak-readyq (ak), '16))

THEOREM: good-ak-ak-error-handler
 (good-ak (ak) \wedge (\neg ak-waiting (ak))) \rightarrow good-ak (ak-error-handler (ak))

EVENT: Disable good-ak-ak-error-handler.

THEOREM: good-ak-ak-clock-interrupt-handler
(good-ak(ak) \wedge (\neg ak-waiting(ak)))
 \rightarrow good-ak(ak-clock-interrupt-handler(ak))

EVENT: Disable good-ak-ak-clock-interrupt-handler.

THEOREM: good-ak-ak-block-send
(good-ak(ak)
 \wedge (\neg ak-waiting(ak)
 \wedge (($srcid = ak-taskid(ak)$) \wedge finite-numberp($destid, '16$))))
 \rightarrow good-ak(ak-block-send($srcid, destid, ak$))

EVENT: Disable good-ak-ak-block-send.

THEOREM: ak-waiting-to-receivep-implies-not-status-flag-readyp
ak-waiting-to-receivep($srcid, destid, ak$)
 \rightarrow (\neg status-flag-readyp(getnth($destid, ak-status(ak)$)))

THEOREM: good-ak-ak-execute-send-to-buffer
(good-ak(ak)
 \wedge (\neg ak-waiting(ak)
 \wedge ((\neg qfullp2($srcid, destid, ak-mbuffers(ak), '4$)
 \wedge (finite-numberp($srcid, '16$)
 \wedge (finite-numberp($destid, '16$)
 \wedge finite-numberp($msg, '65536$))))))
 \rightarrow good-ak(ak-execute-send-to-buffer($msg, srcid, destid, ak$))

EVENT: Disable good-ak-ak-execute-send-to-buffer.

THEOREM: good-ak-ak-execute-send
(good-ak(ak)
 \wedge (\neg ak-waiting(ak)
 \wedge (finite-numberp($destid, '16$)
 \wedge finite-numberp($msg, '65536$))))
 \rightarrow good-ak(ak-execute-send($msg, ak-taskid(ak), destid, ak$))

EVENT: Disable good-ak-ak-execute-send.

THEOREM: good-ak-ak-block-receive
(good-ak(ak))

$$\begin{aligned} & \wedge ((\neg \text{ak-waiting}(ak)) \\ & \quad \wedge ((\text{destid} = \text{ak-taskid}(ak)) \wedge \text{finite-numberp}(srcid, '16)))) \\ \rightarrow & \text{good-ak}(\text{ak-block-receive}(srcid, \text{destid}, ak)) \end{aligned}$$

EVENT: Disable good-ak-ak-block-receive.

THEOREM: ak-waiting-to-sendp-implies-not-status-flag-readyp
 ak-waiting-to-sendp(*srcid*, *destid*, *ak*)
 $\rightarrow (\neg \text{status-flag-readyp}(\text{getnth}(srcid, \text{ak-status}(ak))))$

THEOREM: good-status-properties
 good-status(*status*)
 $\rightarrow (\text{plistp}(status) \wedge ((\text{length}(status) = '2) \wedge (\text{finite-numberp}(\text{getnth}('0, status), '6) \wedge \text{finite-numberp}(\text{getnth}('1, status), '16))))$

EVENT: Disable good-status-properties.

THEOREM: good-ak-ak-execute-receive-from-buffer
 (good-ak(*ak*)
 $\wedge ((\neg \text{ak-waiting}(ak)) \wedge ((\neg \text{qempty2}(srcid, \text{destid}, \text{ak-mbuffers}(ak)) \wedge (\text{finite-numberp}(srcid, '16) \wedge \text{finite-numberp}(\text{destid}, '16))))))$
 $\rightarrow \text{good-ak}(\text{ak-execute-receive-from-buffer}(srcid, \text{destid}, ak))$

EVENT: Disable good-ak-ak-execute-receive-from-buffer.

THEOREM: good-ak-ak-execute-receive
 (good-ak(*ak*) $\wedge ((\neg \text{ak-waiting}(ak)) \wedge \text{finite-numberp}(srcid, '16))$)
 $\rightarrow \text{good-ak}(\text{ak-execute-receive}(srcid, \text{ak-taskid}(ak), ak))$

EVENT: Disable good-ak-ak-execute-receive.

THEOREM: good-ak-ak-block-output
 (good-ak(*ak*) $\wedge ((\neg \text{ak-waiting}(ak)) \wedge (id = \text{ak-taskid}(ak)))$)
 $\rightarrow \text{good-ak}(\text{ak-block-output}(id, ak))$

EVENT: Disable good-ak-ak-block-output.

THEOREM: good-ak-obuffers-properties
 good-ak(*ak*)
 $\rightarrow (\text{plistp}(\text{ak-obuffers}(ak)) \wedge (\text{length}(\text{ak-obuffers}(ak)) = '16))$

EVENT: Disable good-ak-obuffers-properties.

THEOREM: good-ak-ak-execute-output-to-buffer
(good-ak(ak)
 \wedge (\neg ak-waiting(ak))
 \wedge ((\neg qfullp(getnth(id , ak-obuffers(ak)), '4))
 \wedge (finite-numberp(id , '16)
 \wedge finite-numberp($char$, '65536))))))
 \rightarrow good-ak(ak-execute-output-to-buffer($char$, id , ak)))

EVENT: Disable good-ak-ak-execute-output-to-buffer.

THEOREM: good-ak-ak-execute-output
(good-ak(ak) \wedge (\neg ak-waiting(ak)) \wedge finite-numberp($char$, '65536)))
 \rightarrow good-ak(ak-execute-output($char$, ak-taskid(ak), ak)))

EVENT: Disable good-ak-ak-execute-output.

THEOREM: good-ak-ak-block-input
(good-ak(ak) \wedge (\neg ak-waiting(ak)) \wedge ($id =$ ak-taskid(ak)))
 \rightarrow good-ak(ak-block-input(id , ak)))

EVENT: Disable good-ak-ak-block-input.

THEOREM: good-ak-ibuffers-properties
good-ak(ak)
 \rightarrow (plistp(ak-ibuffers(ak)) \wedge (length(ak-ibuffers(ak)) = '16))

EVENT: Disable good-ak-ibuffers-properties.

THEOREM: good-ak-ak-execute-input-from-buffer
(good-ak(ak)
 \wedge (\neg ak-waiting(ak))
 \wedge (finite-numberp(id , '16)
 \wedge (\neg qempty(getnth(id , ak-ibuffers(ak))))))
 \rightarrow good-ak(ak-execute-input-from-buffer(id , ak)))

EVENT: Disable good-ak-ak-execute-input-from-buffer.

THEOREM: good-ak-ak-execute-input
(good-ak(ak) \wedge (\neg ak-waiting(ak)))
 \rightarrow good-ak(ak-execute-input(ak-taskid(ak), ak)))

EVENT: Disable good-ak-ak-execute-input.

THEOREM: finite-numberp-ak-srcid
 $\text{good-ak}(ak) \rightarrow \text{finite-numberp}(\text{ak-srcid}(ak), '16)$

EVENT: Disable finite-numberp-ak-srcid.

THEOREM: finite-numberp-ak-destid
 $\text{good-ak}(ak) \rightarrow \text{finite-numberp}(\text{ak-destid}(ak), '16)$

EVENT: Disable finite-numberp-ak-destid.

THEOREM: finite-numberp-ak-message
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak)))$
 $\rightarrow \text{finite-numberp}(\text{ak-message}(ak), '65536)$

EVENT: Disable finite-numberp-ak-message.

THEOREM: good-ak-ak-svc-handler
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak))) \rightarrow \text{good-ak}(\text{ak-svc-handler}(ak))$

EVENT: Disable good-ak-ak-svc-handler.

THEOREM: good-ak-ak-private-step
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak))) \rightarrow \text{good-ak}(\text{ak-private-step}(ak))$

EVENT: Disable good-ak-ak-private-step.

THEOREM: ak-waiting-to-inputp-implies-not-status-flag-readyp
 $\text{ak-waiting-to-inputp}(id, ak)$
 $\rightarrow (\neg \text{status-flag-readyp}(\text{getnth}(id, \text{ak-status}(ak))))$

THEOREM: ak-waiting-to-outputp-implies-not-status-flag-readyp
 $\text{ak-waiting-to-outputp}(id, ak)$
 $\rightarrow (\neg \text{status-flag-readyp}(\text{getnth}(id, \text{ak-status}(ak))))$

THEOREM: good-tm-iport-properties
 $\text{good-tm-iport}(port)$
 $\rightarrow (\text{tm-iportp}(port)$
 $\quad \wedge (\text{finite-numberp}(\text{tm-iinterrupt-flag}(port), '2)$
 $\quad \wedge (\text{finite-numberp}(\text{tm-ierror-flag}(port), '2)$
 $\quad \wedge \text{finite-numberp}(\text{tm-ichar}(port), '256))))$

EVENT: Disable good-tm-iport-properties.

THEOREM: good-ak-ak-waiting-input-interrupt-handler-case1
(good-ak (ak)
 \wedge (finite-numberp (id , '16)
 \wedge (ak-waiting (ak) \wedge ak-waiting-to-inputp (id , ak))))
 \rightarrow good-ak (ak-waiting-input-interrupt-handler (id , ak)))

THEOREM: good-ak-ak-waiting-input-interrupt-handler-case2
(good-ak (ak)
 \wedge (finite-numberp (id , '16)
 \wedge (ak-waiting (ak) \wedge (\neg ak-waiting-to-inputp (id , ak))))))
 \rightarrow good-ak (ak-waiting-input-interrupt-handler (id , ak)))

THEOREM: good-ak-ak-waiting-input-interrupt-handler
(good-ak (ak) \wedge (finite-numberp (id , '16) \wedge ak-waiting (ak)))
 \rightarrow good-ak (ak-waiting-input-interrupt-handler (id , ak)))

EVENT: Disable good-ak-ak-waiting-input-interrupt-handler.

THEOREM: good-ak-ak-running-input-interrupt-handler-case1
(good-ak (ak)
 \wedge (finite-numberp (id , '16)
 \wedge ((\neg ak-waiting (ak)) \wedge ak-waiting-to-inputp (id , ak))))
 \rightarrow good-ak (ak-running-input-interrupt-handler (id , ak)))

THEOREM: good-ak-ak-running-input-interrupt-handler-case2
(good-ak (ak)
 \wedge (finite-numberp (id , '16)
 \wedge ((\neg ak-waiting (ak)) \wedge (\neg ak-waiting-to-inputp (id , ak))))))
 \rightarrow good-ak (ak-running-input-interrupt-handler (id , ak)))

THEOREM: good-ak-ak-running-input-interrupt-handler
(good-ak (ak) \wedge (finite-numberp (id , '16) \wedge (\neg ak-waiting (ak))))
 \rightarrow good-ak (ak-running-input-interrupt-handler (id , ak)))

EVENT: Disable good-ak-ak-running-input-interrupt-handler.

THEOREM: finite-numberp-interrupting-ak-input-port
(good-ak (ak) \wedge tm-some-input-interruptp (ak-ports (ak)))
 \rightarrow finite-numberp (tm-interrupting-input-port (ak-ports (ak)), '16)

EVENT: Disable finite-numberp-interrupting-ak-input-port.

THEOREM: good-ak-ak-input-interrupt-handler
 $(\text{good-ak}(ak) \wedge \text{ak-input-interruptp}(ak))$
 $\rightarrow \text{good-ak}(\text{ak-input-interrupt-handler}(\text{ak-interrupting-input-port}(\text{ak-iports}(ak)),$
 $ak))$

EVENT: Disable good-ak-ak-input-interrupt-handler.

THEOREM: good-tm-oport-properties
 $\text{good-tm-oport}(port)$
 $\rightarrow (\text{tm-oportp}(port)$
 $\wedge (\text{finite-numberp}(\text{tm-ointerrupt-flag}(port), '2)$
 $\wedge (\text{finite-numberp}(\text{tm-obusy-flag}(port), '2)$
 $\wedge \text{finite-numberp}(\text{tm-ochar}(port), '256))))$

EVENT: Disable good-tm-oport-properties.

THEOREM: good-ak-ak-waiting-output-interrupt-handler-case1
 $(\text{good-ak}(ak)$
 $\wedge (\text{finite-numberp}(id, '16)$
 $\wedge (\text{ak-waiting}(ak) \wedge \text{ak-waiting-to-outputp}(id, ak))))$
 $\rightarrow \text{good-ak}(\text{ak-waiting-output-interrupt-handler}(id, ak))$

THEOREM: good-ak-ak-waiting-output-interrupt-handler-case2
 $(\text{good-ak}(ak)$
 $\wedge (\text{finite-numberp}(id, '16)$
 $\wedge (\text{ak-waiting}(ak) \wedge (\neg \text{ak-waiting-to-outputp}(id, ak))))))$
 $\rightarrow \text{good-ak}(\text{ak-waiting-output-interrupt-handler}(id, ak))$

THEOREM: good-ak-ak-waiting-output-interrupt-handler
 $(\text{good-ak}(ak) \wedge (\text{finite-numberp}(id, '16) \wedge \text{ak-waiting}(ak)))$
 $\rightarrow \text{good-ak}(\text{ak-waiting-output-interrupt-handler}(id, ak))$

EVENT: Disable good-ak-ak-waiting-output-interrupt-handler.

THEOREM: good-ak-ak-running-output-interrupt-handler-case1
 $(\text{good-ak}(ak)$
 $\wedge (\text{finite-numberp}(id, '16)$
 $\wedge ((\neg \text{ak-waiting}(ak)) \wedge \text{ak-waiting-to-outputp}(id, ak))))$
 $\rightarrow \text{good-ak}(\text{ak-running-output-interrupt-handler}(id, ak))$

THEOREM: good-ak-ak-running-output-interrupt-handler-case2
 $(\text{good-ak}(ak)$
 $\wedge (\text{finite-numberp}(id, '16)$
 $\wedge ((\neg \text{ak-waiting}(ak)) \wedge (\neg \text{ak-waiting-to-outputp}(id, ak))))))$
 $\rightarrow \text{good-ak}(\text{ak-running-output-interrupt-handler}(id, ak))$

THEOREM: good-ak-ak-running-output-interrupt-handler
 $(\text{good-ak}(ak) \wedge (\text{finite-numberp}(id, '16) \wedge (\neg \text{ak-waiting}(ak))))$
 $\rightarrow \text{good-ak}(\text{ak-running-output-interrupt-handler}(id, ak))$

EVENT: Disable good-ak-ak-running-output-interrupt-handler.

THEOREM: finite-numberp-interrupting-ak-output-port
 $(\text{good-ak}(ak) \wedge \text{tm-some-output-interruptp}(\text{ak-oports}(ak)))$
 $\rightarrow \text{finite-numberp}(\text{tm-interrupting-output-port}(\text{ak-oports}(ak)), '16)$

EVENT: Disable finite-numberp-interrupting-ak-output-port.

THEOREM: good-ak-ak-output-interrupt-handler
 $(\text{good-ak}(ak) \wedge \text{ak-output-interruptp}(ak))$
 $\rightarrow \text{good-ak}(\text{ak-output-interrupt-handler}(\text{ak-interrupting-output-port}(\text{ak-oports}(ak)), ak))$

EVENT: Disable good-ak-ak-output-interrupt-handler.

THEOREM: good-ak-ak-step
 $\text{good-ak}(ak) \rightarrow \text{good-ak}(\text{ak-step}(ak))$

EVENT: Disable good-ak-ak-step.

THEOREM: good-ak-ak-post-input-interrupt
 $(\text{good-ak}(ak) \wedge \text{tm-device-input-eventp}(interrupt))$
 $\rightarrow \text{good-ak}(\text{ak-post-interrupt}(interrupt, ak))$

EVENT: Disable good-ak-ak-post-input-interrupt.

THEOREM: good-ak-ak-post-output-interrupt
 $(\text{good-ak}(ak) \wedge \text{tm-device-output-eventp}(interrupt))$
 $\rightarrow \text{good-ak}(\text{ak-post-interrupt}(interrupt, ak))$

EVENT: Disable good-ak-ak-post-output-interrupt.

THEOREM: good-ak-ak-post-interrupt-noop
 $(\text{good-ak}(ak)$
 $\wedge ((\neg \text{tm-device-input-eventp}(interrupt))$
 $\wedge (\neg \text{tm-device-output-eventp}(interrupt))))$
 $\rightarrow \text{good-ak}(\text{ak-post-interrupt}(interrupt, ak))$

EVENT: Disable good-ak-ak-post-interrupt-noop.

THEOREM: good-ak-ak-post-interrupt
 $\text{good-ak}(ak) \rightarrow \text{good-ak}(\text{ak-post-interrupt}(interrupt, ak))$

EVENT: Disable good-ak-ak-post-interrupt.

THEOREM: good-ak-ak-processor
 $\text{good-ak}(ak) \rightarrow \text{good-ak}(\text{ak-processor}(ak, oracle))$

EVENT: Disable good-ak-ak-processor.

THEOREM: ak-pstates-ak-clock-interrupt-handler
 $\text{ak-pstates}(\text{ak-clock-interrupt-handler}(ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-clock-interrupt-handler.

THEOREM: ak-pstates-ak-error-handler
 $\text{ak-pstates}(\text{ak-error-handler}(ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-error-handler.

THEOREM: ak-pstates-ak-block-send
 $\text{ak-pstates}(\text{ak-block-send}(srcid, destid, ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-block-send.

THEOREM: length-ak-pstates-ak-execute-send-to-buffer
 $\text{length}(\text{ak-pstates}(\text{ak-execute-send-to-buffer}(msg, srcid, destid, ak)))$
 $= \text{length}(\text{ak-pstates}(ak))$

EVENT: Disable length-ak-pstates-ak-execute-send-to-buffer.

THEOREM: length-ak-pstates-ak-execute-send
 $\text{length}(\text{ak-pstates}(\text{ak-execute-send}(msg, srcid, destid, ak)))$
 $= \text{length}(\text{ak-pstates}(ak))$

EVENT: Disable length-ak-pstates-ak-execute-send.

THEOREM: ak-pstates-ak-block-receive
 $\text{ak-pstates}(\text{ak-block-receive}(srcid, destid, ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-block-receive.

THEOREM: length-ak-pstates-ak-execute-receive-from-buffer
length (ak-pstates (ak-execute-receive-from-buffer (*srcid*, *destid*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-receive-from-buffer.

THEOREM: length-ak-pstates-ak-execute-receive
length (ak-pstates (ak-execute-receive (*srcid*, *destid*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-receive.

THEOREM: ak-pstates-ak-block-output
ak-pstates (ak-block-output (*id*, *ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-block-output.

THEOREM: length-ak-pstates-ak-execute-output-to-buffer
length (ak-pstates (ak-execute-output-to-buffer (*char*, *id*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-output-to-buffer.

THEOREM: length-ak-pstates-ak-execute-output
length (ak-pstates (ak-execute-output (*char*, *id*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-output.

THEOREM: ak-pstates-ak-block-input
ak-pstates (ak-block-input (*id*, *ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-block-input.

THEOREM: length-ak-pstates-ak-execute-input-from-buffer
length (ak-pstates (ak-execute-input-from-buffer (*id*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-input-from-buffer.

THEOREM: length-ak-pstates-ak-execute-input
length (ak-pstates (ak-execute-input (*id*, *ak*))) = length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-input.

THEOREM: length-ak-pstates-ak-svc-handler
 $\text{length}(\text{ak-pstates}(\text{ak-svc-handler}(ak))) = \text{length}(\text{ak-pstates}(ak))$

EVENT: Disable length-ak-pstates-ak-svc-handler.

THEOREM: ak-pstates-ak-input-interrupt-handler
 $\text{ak-pstates}(\text{ak-input-interrupt-handler}(id, ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-input-interrupt-handler.

THEOREM: ak-pstates-ak-output-interrupt-handler
 $\text{ak-pstates}(\text{ak-output-interrupt-handler}(id, ak)) = \text{ak-pstates}(ak)$

EVENT: Disable ak-pstates-ak-output-interrupt-handler.

THEOREM: length-ak-pstates-ak-private-step
 $\text{length}(\text{ak-pstates}(\text{ak-private-step}(ak))) = \text{length}(\text{ak-pstates}(ak))$

EVENT: Disable length-ak-pstates-ak-private-step.

THEOREM: length-ak-pstates-ak-step
 $\text{length}(\text{ak-pstates}(\text{ak-step}(ak))) = \text{length}(\text{ak-pstates}(ak))$

EVENT: Disable length-ak-pstates-ak-step.

DEFINITION:
 $\text{ak-channels}(ak)$
 $= \text{cons}(\text{ak-ibuffers}(ak),$
 $\quad \text{cons}(\text{ak-obuffers}(ak), \text{cons}(\text{ak-mbuffers}(ak), \text{'nil'})))$

DEFINITION:
 $\text{project}(i, ak) = \text{task}(\text{getnth}(i, \text{ak-pstates}(ak)), \text{ak-channels}(ak))$

DEFINITION:
 $\text{control-oracle-step}(i, ak)$
 $= \text{if } \text{ak-input-interruptp}(ak)$
 $\quad \text{then } \text{ak-channels}(\text{ak-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{ak-iports}(ak)),$
 $\quad \quad \quad ak))$
 $\quad \text{elseif } \text{ak-output-interruptp}(ak)$
 $\quad \text{then } \text{ak-channels}(\text{ak-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{ak-oports}(ak)),$
 $\quad \quad \quad ak))$


```

                                ak))
elseif ak-waiting(ak) then ak-channels(ak)
elseif ak-errorp(ak) then ak-channels(ak-error-handler(ak))
elseif ak-clock-interruptp(ak)
then ak-channels(ak-clock-interrupt-handler(ak))
elseif ak-svc-interruptp(ak)
then if i = ak-taskid(ak) then '*1*true
      else ak-channels(ak-svc-handler(ak)) endif
elseif i = ak-taskid(ak) then '*1*true
else ak-channels(ak-private-step(ak)) endif

```

DEFINITION:

```

control-oracle(i, ak, oracle)
= if listp(oracle)
  then cons(control-oracle-step(i, ak-post-interrupt(car(oracle), ak)),
            control-oracle(i,
                            ak-step(ak-post-interrupt(car(oracle), ak)),
                            cdr(oracle)))
  else 'nil endif

```

THEOREM: access-ak-post-interrupt

```

(ak-pstates(ak-post-interrupt(interrupt, ak)) = ak-pstates(ak))
 $\wedge$  ((ak-ibuffers(ak-post-interrupt(interrupt, ak)) = ak-ibuffers(ak))
       $\wedge$  ((ak-obuffers(ak-post-interrupt(interrupt, ak))
          = ak-obuffers(ak))
       $\wedge$  ((ak-mbuffers(ak-post-interrupt(interrupt, ak))
          = ak-mbuffers(ak))
       $\wedge$  (ak-taskid(ak-post-interrupt(interrupt, ak))
          = ak-taskid(ak))))))

```

EVENT: Disable access-ak-post-interrupt.

THEOREM: access-ak-input-interrupt-handler

```

(ak-pstates(ak-input-interrupt-handler(id, ak)) = ak-pstates(ak))
 $\wedge$  ((ak-obuffers(ak-input-interrupt-handler(id, ak)) = ak-obuffers(ak))
       $\wedge$  (ak-mbuffers(ak-input-interrupt-handler(id, ak))
          = ak-mbuffers(ak)))

```

EVENT: Disable access-ak-input-interrupt-handler.

THEOREM: access-ak-output-interrupt-handler

```

(ak-pstates(ak-output-interrupt-handler(id, ak)) = ak-pstates(ak))
 $\wedge$  ((ak-ibuffers(ak-output-interrupt-handler(id, ak)) = ak-ibuffers(ak))
       $\wedge$  (ak-mbuffers(ak-output-interrupt-handler(id, ak))
          = ak-mbuffers(ak)))

```

EVENT: Disable access-ak-output-interrupt-handler.

THEOREM: access-ak-error-handler

$$\begin{aligned}
& (\text{ak-pstates}(\text{ak-error-handler}(ak)) = \text{ak-pstates}(ak)) \\
\wedge & ((\text{ak-ibuffers}(\text{ak-error-handler}(ak)) = \text{ak-ibuffers}(ak)) \\
& \quad \wedge ((\text{ak-obuffers}(\text{ak-error-handler}(ak)) = \text{ak-obuffers}(ak)) \\
& \quad \quad \wedge (\text{ak-mbuffers}(\text{ak-error-handler}(ak)) \\
& \quad \quad \quad = \text{ak-mbuffers}(ak))))))
\end{aligned}$$

EVENT: Disable access-ak-error-handler.

THEOREM: ak-private-step-non-interference

$$\begin{aligned}
& (\text{good-ak}(ak) \\
& \quad \wedge (\text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak))) \\
& \quad \quad \wedge ((i \neq \text{ak-taskid}(ak)) \wedge (\neg \text{ak-waiting}(ak)))))) \\
\rightarrow & (\text{getnth}(i, \text{ak-pstates}(\text{ak-private-step}(ak))) \\
& \quad = \text{getnth}(i, \text{ak-pstates}(ak)))
\end{aligned}$$

EVENT: Disable ak-private-step-non-interference.

THEOREM: ak-svc-handler-non-interference

$$\begin{aligned}
& (\text{good-ak}(ak) \\
& \quad \wedge (\text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak))) \\
& \quad \quad \wedge ((i \neq \text{ak-taskid}(ak)) \wedge (\neg \text{ak-waiting}(ak)))))) \\
\rightarrow & (\text{getnth}(i, \text{ak-pstates}(\text{ak-svc-handler}(ak))) \\
& \quad = \text{getnth}(i, \text{ak-pstates}(ak)))
\end{aligned}$$

EVENT: Disable ak-svc-handler-non-interference.

THEOREM: ak-implements-non-active-task-step

$$\begin{aligned}
& (\text{good-ak}(ak) \\
& \quad \wedge (\text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak))) \\
& \quad \quad \wedge (\neg \text{task-activep}(\text{control-oracle-step}(i, \\
& \quad \quad \quad \text{ak-post-interrupt}(\text{car}(\text{oracle}), \\
& \quad \quad \quad \text{ak})))))) \\
\rightarrow & (\text{project}(i, \text{ak-step}(\text{ak-post-interrupt}(\text{car}(\text{oracle}), \text{ak}))) \\
& \quad = \text{task-update-channels}(\text{project}(i, \text{ak}), \\
& \quad \quad \text{control-oracle-step}(i, \\
& \quad \quad \quad \text{ak-post-interrupt}(\text{car}(\text{oracle}), \\
& \quad \quad \quad \text{ak}))))
\end{aligned}$$

EVENT: Disable ak-implements-non-active-task-step.

THEOREM: ak-execute-receive-implements-task-execute-receive
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak)))$
 \rightarrow (task (getnth (ak-taskid (ak),
 ak-pstates (ak-execute-receive (ak-srcid (ak),
 ak-taskid (ak),
 ak))),
 ak-channels (ak-execute-receive (ak-srcid (ak), ak-taskid (ak), ak)))
 $=$ task-execute-receive (task-srcid (task (getnth (ak-taskid (ak),
 ak-pstates (ak)),
 ak-channels (ak))),
 ak-taskid (ak),
 task (getnth (ak-taskid (ak), ak-pstates (ak)),
 ak-channels (ak))))

EVENT: Disable ak-execute-receive-implements-task-execute-receive.

THEOREM: ak-execute-send-implements-task-execute-send
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak)))$
 \rightarrow (task (getnth (ak-taskid (ak),
 ak-pstates (ak-execute-send (ak-message (ak),
 ak-taskid (ak),
 ak-destid (ak),
 ak))),
 ak-channels (ak-execute-send (ak-message (ak),
 ak-taskid (ak),
 ak-destid (ak),
 ak))))
 $=$ task-execute-send (task-message (task (getnth (ak-taskid (ak),
 ak-pstates (ak)),
 ak-channels (ak))),
 ak-taskid (ak),
 task-destid (task (getnth (ak-taskid (ak),
 ak-pstates (ak)),
 ak-channels (ak))),
 task (getnth (ak-taskid (ak), ak-pstates (ak)),
 ak-channels (ak))))

EVENT: Disable ak-execute-send-implements-task-execute-send.

THEOREM: ak-execute-input-implements-task-execute-input
 $(\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak)))$
 \rightarrow (task (getnth (ak-taskid (ak),
 ak-pstates (ak-execute-input (ak-taskid (ak), ak))),

$$\begin{aligned}
& \text{ak-channels}(\text{ak-execute-input}(\text{ak-taskid}(ak), ak)) \\
= & \text{task-execute-input}(\text{ak-taskid}(ak), \\
& \text{task}(\text{getnth}(\text{ak-taskid}(ak), \text{ak-pstates}(ak)), \\
& \text{ak-channels}(ak)))
\end{aligned}$$

EVENT: Disable ak-execute-input-implements-task-execute-input.

$$\begin{aligned}
\text{THEOREM: ak-execute-output-implements-task-execute-output} \\
& (\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak))) \\
\rightarrow & (\text{task}(\text{getnth}(\text{ak-taskid}(ak), \\
& \text{ak-pstates}(\text{ak-execute-output}(\text{ak-message}(ak), \\
& \text{ak-taskid}(ak), \\
& ak))), \\
& \text{ak-channels}(\text{ak-execute-output}(\text{ak-message}(ak), \text{ak-taskid}(ak), ak))) \\
= & \text{task-execute-output}(\text{task-message}(\text{task}(\text{getnth}(\text{ak-taskid}(ak), \\
& \text{ak-pstates}(ak)), \\
& \text{ak-channels}(ak))), \\
& \text{ak-taskid}(ak), \\
& \text{task}(\text{getnth}(\text{ak-taskid}(ak), \text{ak-pstates}(ak)), \\
& \text{ak-channels}(ak)))
\end{aligned}$$

EVENT: Disable ak-execute-output-implements-task-execute-output.

$$\begin{aligned}
\text{THEOREM: ak-svc-handler-implements-task-communication-step} \\
& (\text{good-ak}(ak) \wedge (\neg \text{ak-waiting}(ak))) \\
\rightarrow & (\text{task}(\text{getnth}(\text{ak-taskid}(ak), \text{ak-pstates}(\text{ak-svc-handler}(ak))), \\
& \text{ak-channels}(\text{ak-svc-handler}(ak))) \\
= & \text{task-communication-step}(\text{task}(\text{getnth}(\text{ak-taskid}(ak), \\
& \text{ak-pstates}(ak)), \\
& \text{ak-channels}(ak)), \\
& \text{ak-taskid}(ak))
\end{aligned}$$

EVENT: Disable ak-svc-handler-implements-task-communication-step.

$$\begin{aligned}
\text{THEOREM: ak-private-step-implements-task-fetch-execute} \\
& (\text{good-ak}(ak) \\
& \wedge ((\neg \text{ak-waiting}(ak)) \\
& \wedge (\text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak))) \\
& \wedge (i = \text{ak-taskid}(ak)))))) \\
\rightarrow & (\text{task}(\text{getnth}(i, \text{ak-pstates}(\text{ak-private-step}(ak))), \\
& \text{ak-channels}(\text{ak-private-step}(ak))) \\
= & \text{task}(\text{tm-fetch-execute}(\text{getnth}(i, \text{ak-pstates}(ak))), \\
& \text{ak-channels}(ak))
\end{aligned}$$

EVENT: Disable ak-private-step-implements-task-fetch-execute.

THEOREM: ak-implements-active-task-step

$$\begin{aligned} & (\text{good-ak}(ak) \\ & \wedge (\text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak))) \\ & \quad \wedge \text{task-activep}(\text{control-oracle-step}(i, \\ & \quad \quad \quad \text{ak-post-interrupt}(\text{car}(oracle), \\ & \quad \quad \quad \quad \quad \quad \quad ak)))))) \\ \rightarrow & (\text{project}(i, \text{ak-step}(\text{ak-post-interrupt}(\text{car}(oracle), ak))) \\ & = \text{task-step}(\text{project}(i, ak), i)) \end{aligned}$$

EVENT: Disable ak-implements-active-task-step.

THEOREM: ak-implements-parallel-tasks

$$\begin{aligned} & (\text{good-ak}(ak) \wedge \text{finite-numberp}(i, \text{length}(\text{ak-pstates}(ak)))) \\ \rightarrow & (\text{project}(i, \text{ak-processor}(ak, oracle)) \\ & = \text{task-processor}(\text{project}(i, ak), i, \text{control-oracle}(i, ak, oracle))) \end{aligned}$$

EVENT: Disable ak-implements-parallel-tasks.

DEFINITION:

ASM-OPCODE-ALIST

```
= '(return . 0)
    (wait . 4)
    (run . 8)
    (br . 2)
    (brz . 10)
    (brnz . 6)
    (call . 14)
    (decr . 18)
    (decrm . 5)
    (incr . 22)
    (incrm . 7)
    (lbase . 26)
    (llimit . 30)
    (lpsw . 34)
    (post . 38)
    (svc . 42)
    (svcr . 46)
    (time . 58)
    (add . 1)
    (compare . 3)
    (mod . 9)
```

```

(move . 11)
(mult . 13)
(stout . 15)
(testi . 50)
(testo . 54)

```

DEFINITION:

```

append-n-times (n, list)
= if n  $\simeq$  0 then 'nil
  else append (list, append-n-times (n - 1, list)) endif

```

DEFINITION:

```

lookup (x, alist)
= if listp (alist)
  then if listp (car (alist))  $\wedge$  (x = car (car (alist)))
    then cdr (car (alist))
    else lookup (x, cdr (alist)) endif
  else 'nil endif

```

DEFINITION:

```

boundp (x, alist)
= if listp (alist)
  then if listp (car (alist))
    then if x = car (car (alist)) then '*1*true
      else boundp (x, cdr (alist)) endif
    else boundp (x, cdr (alist)) endif
  else '*1*false endif

```

DEFINITION:

```

number-and-litatom-listp (x)
= if listp (x)
  then ((car (x)  $\in$   $\mathbf{N}$ )  $\vee$  litatom (car (x)))
     $\wedge$  number-and-litatom-listp (cdr (x))
  else '*1*true endif

```

DEFINITION:

```

dc-formp (x)
= (listp (x)
   $\wedge$  ((car (x) = 'dc)
     $\wedge$  ((car (cdr (x))  $\in$   $\mathbf{N}$ )
       $\wedge$  ((car (cdr (cdr (x)))  $\in$   $\mathbf{N}$ )
         $\vee$  (litatom (car (cdr (cdr (x))))))
         $\vee$  (listp (car (cdr (cdr (x))))))
         $\vee$  number-and-litatom-listp (car (cdr (cdr (x))))))))))

```

DEFINITION:

```
translate-dc-arg (arg, symtab)
= if arg ∈ N then arg
  elseif litatom (arg)
  then if boundp (arg, symtab) then lookup (arg, symtab)
    else arg endif
  else arg endif
```

DEFINITION:

```
translate-dc-arglist (arglist, symtab)
= if listp (arglist)
  then cons (translate-dc-arg (car (arglist), symtab),
            translate-dc-arglist (cdr (arglist), symtab))
  else 'nil endif
```

DEFINITION:

```
translate-dc (dc-form, symtab)
= if dc-formp (dc-form)
  then append-n-times (car (cdr (dc-form)),
                      translate-dc-arglist (if listp (car (cdr (cdr (dc-form))))
                                             then car (cdr (cdr (dc-form)))
                                             else cons (car (cdr (cdr (dc-form))),
                                                         'nil) endif,
                      symtab))
  else dc-form endif
```

DEFINITION:

```
asm-instruction-opcode (instruction)
= lookup (car (instruction),
         '(return . 0)
         (wait . 4)
         (run . 8)
         (br . 2)
         (brz . 10)
         (brnz . 6)
         (call . 14)
         (decr . 18)
         (decrm . 5)
         (incr . 22)
         (incrm . 7)
         (lbase . 26)
         (llimit . 30)
         (lpsw . 34)
         (post . 38)
         (svc . 42))
```

```

(svcr . 46)
(time . 58)
(add . 1)
(compare . 3)
(mod . 9)
(move . 11)
(mult . 13)
(stout . 15)
(testi . 50)
(testo . 54))

```

DEFINITION:

```

asm-instruction-model (instruction, symtab)
= if listp (car (cdr (instruction)))
  then if boundp (car (car (cdr (instruction))), symtab)
    then lookup (car (car (cdr (instruction))), symtab)
    else car (car (cdr (instruction))) endif
  else '0 endif

```

DEFINITION:

```

asm-instruction-arg1 (instruction, symtab)
= if listp (car (cdr (instruction)))
  then if boundp (car (cdr (car (cdr (instruction)))), symtab)
    then lookup (car (cdr (car (cdr (instruction)))), symtab)
    else car (cdr (car (cdr (instruction)))) endif
  elseif boundp (car (cdr (instruction)), symtab)
    then lookup (car (cdr (instruction)), symtab)
    else car (cdr (instruction)) endif

```

DEFINITION:

```

asm-instruction-index1 (instruction, symtab)
= if listp (car (cdr (instruction)))
  then if boundp (car (cdr (cdr (car (cdr (instruction))))), symtab)
    then lookup (car (cdr (cdr (car (cdr (instruction))))), symtab)
    else car (cdr (cdr (car (cdr (instruction)))))) endif
  else '0 endif

```

DEFINITION:

```

asm-instruction-mode2 (instruction, symtab)
= if listp (car (cdr (cdr (instruction))))
  then if boundp (car (car (cdr (cdr (instruction)))), symtab)
    then lookup (car (car (cdr (cdr (instruction))))), symtab)
    else car (cdr (cdr (instruction)))) endif
  else '0 endif

```


DEFINITION:

```
asm-instruction-arg2(instruction, symtab)
=  if listp(car(cdr(cdr(instruction))))
    then if boundp(car(cdr(car(cdr(cdr(instruction))))), symtab)
        then lookup(car(cdr(car(cdr(cdr(instruction))))), symtab)
        else car(cdr(car(cdr(cdr(instruction)))) endif
    elseif boundp(car(cdr(cdr(instruction))), symtab)
        then lookup(car(cdr(cdr(instruction))), symtab)
        else car(cdr(cdr(instruction))) endif
```

DEFINITION:

```
asm-instruction-index2(instruction, symtab)
=  if listp(car(cdr(cdr(instruction))))
    then if boundp(car(cdr(cdr(car(cdr(cdr(instruction))))), symtab)
        then lookup(car(cdr(cdr(car(cdr(cdr(instruction))))), symtab)
        else car(cdr(cdr(car(cdr(cdr(instruction)))))) endif
    else '0 endif
```

DEFINITION:

```
asm-good-arg(arg, mode)
=  if (mode = '0) ∨ (mode = '2) then finite-numberp(arg, '65536)
    else finite-numberp(arg, '8) endif
```

DEFINITION:

```
asm-good-instruction(instruction, symtab)
=  (boundp(car(instruction),
            '(return . 0)
            (wait . 4)
            (run . 8)
            (br . 2)
            (brz . 10)
            (brnz . 6)
            (call . 14)
            (decr . 18)
            (decrm . 5)
            (incr . 22)
            (incrm . 7)
            (lbase . 26)
            (llimit . 30)
            (lpsw . 34)
            (post . 38)
            (svc . 42)
            (svcr . 46)
            (time . 58)
            (add . 1)
```

```

      (compare . 3)
      (mod . 9)
      (move . 11)
      (mult . 13)
      (stout . 15)
      (testi . 50)
      (testo . 54)))
^ ((length (instruction)
   = if tm-nullary-instructionp (lookup (car (instruction),
                                           '(return
                                             . 0)
                                             (wait . 4)
                                             (run . 8)
                                             (br . 2)
                                             (brz . 10)
                                             (brnz . 6)
                                             (call . 14)
                                             (decr . 18)
                                             (decrm . 5)
                                             (incr . 22)
                                             (incrm . 7)
                                             (lbase
                                              . 26)
                                             (llimit
                                              . 30)
                                             (lpsw . 34)
                                             (post . 38)
                                             (svc . 42)
                                             (svcr . 46)
                                             (time . 58)
                                             (add . 1)
                                             (compare
                                              . 3)
                                             (mod . 9)
                                             (move . 11)
                                             (mult . 13)
                                             (stout
                                              . 15)
                                             (testi
                                              . 50)
                                             (testo
                                              . 54))))))
then '1
elseif tm-unary-instructionp (lookup (car (instruction),

```

```
'(return
  . 0)
  (wait . 4)
  (run . 8)
  (br . 2)
  (brz . 10)
  (brnz . 6)
  (call
    . 14)
  (decr
    . 18)
  (decrm
    . 5)
  (incr
    . 22)
  (incrm
    . 7)
  (lbase
    . 26)
  (llimit
    . 30)
  (lpsw
    . 34)
  (post
    . 38)
  (svc . 42)
  (svcr
    . 46)
  (time
    . 58)
  (add . 1)
  (compare
    . 3)
  (mod . 9)
  (move
    . 11)
  (mult
    . 13)
  (stout
    . 15)
  (testi
    . 50)
  (testo
    . 54))))
```

```

then '2
else '3 endif
 $\wedge$  (finite-numberp (asm-instruction-mode1 (instruction,
                                             symtab),
                                             '4)
 $\wedge$  (finite-numberp (asm-instruction-mode2 (instruction,
                                             symtab),
                                             '4)
 $\wedge$  (asm-good-arg (asm-instruction-arg1 (instruction,
                                          symtab),
                                          asm-instruction-mode1 (instruction,
                                                                  symtab))
 $\wedge$  (asm-good-arg (asm-instruction-arg2 (instruction,
                                          symtab),
                                          asm-instruction-mode2 (instruction,
                                                                  symtab))
 $\wedge$  (finite-numberp (asm-instruction-index1 (instruction,
                                              symtab),
                                              '8)
 $\wedge$  finite-numberp (asm-instruction-index2 (instruction,
                                              symtab),
                                              '8)))))))))

```

DEFINITION:

```

asm-pack-word1 (instruction, symtab)
= if length (instruction) = '1
  then asm-instruction-opcode (instruction)
  elseif length (instruction) = '2
  then asm-instruction-opcode (instruction)
    + ((asm-instruction-mode1 (instruction, symtab) * '64)
      + (asm-instruction-index1 (instruction, symtab)
        * '256))
  else asm-instruction-opcode (instruction)
    + ((asm-instruction-mode1 (instruction, symtab)
      * '64)
      + ((asm-instruction-index1 (instruction, symtab)
        * '256)
      + ((asm-instruction-mode2 (instruction,
                                symtab)
        * '2048)
      + (asm-instruction-index2 (instruction,
                                symtab)
        * '8192)))))) endif

```

DEFINITION:

```

asm-pack-instruction (instruction, symtab)
= if length (instruction) = '1
  then cons (asm-pack-word1 (instruction, symtab), 'nil)
  elseif length (instruction) = '2
  then cons (asm-pack-word1 (instruction, symtab),
            cons (asm-instruction-arg1 (instruction, symtab), 'nil))
  else cons (asm-pack-word1 (instruction, symtab),
            cons (asm-instruction-arg1 (instruction, symtab),
                  cons (asm-instruction-arg2 (instruction, symtab),
                        'nil))) endif

```

DEFINITION:

```

translate-instruction (instruction, symtab)
= if asm-good-instruction (instruction, symtab)
  then asm-pack-instruction (instruction, symtab)
  else instruction endif

```

DEFINITION:

```

translate-source (source, symtab)
= if listp (source)
  then if litatom (car (source))
    then translate-source (cdr (source), symtab)
    elseif listp (car (source))  $\wedge$  (car (car (source))) = 'dcl
    then translate-source (cdr (source), symtab)
    elseif listp (car (source))  $\wedge$  (car (car (source))) = 'dc
    then append (translate-dc (car (source), symtab),
                 translate-source (cdr (source), symtab))
    else append (translate-instruction (car (source), symtab),
                 translate-source (cdr (source), symtab)) endif
  else 'nil endif

```

DEFINITION:

```

incr-dc-position (dc-form)
= if listp (car (cdr (cdr (dc-form))))
  then car (cdr (dc-form)) * length (car (cdr (cdr (dc-form))))
  else car (cdr (dc-form)) endif

```

DEFINITION:

```

incr-instruction-position (instruction)
= if boundp (car (instruction),
              '(return . 0)
              (wait . 4)
              (run . 8)
              (br . 2)
              (brz . 10))

```

```

(brnz . 6)
(call . 14)
(decr . 18)
(decrm . 5)
(incr . 22)
(incrm . 7)
(lbase . 26)
(llimit . 30)
(lpsw . 34)
(post . 38)
(svc . 42)
(svcr . 46)
(time . 58)
(add . 1)
(compare . 3)
(mod . 9)
(move . 11)
(mult . 13)
(stout . 15)
(testi . 50)
(testo . 54))
then if tm-nullary-instructionp (lookup (car (instruction),
'((return . 0)
  (wait . 4)
  (run . 8)
  (br . 2)
  (brz . 10)
  (brnz . 6)
  (call . 14)
  (decr . 18)
  (decrm . 5)
  (incr . 22)
  (incrm . 7)
  (lbase . 26)
  (llimit . 30)
  (lpsw . 34)
  (post . 38)
  (svc . 42)
  (svcr . 46)
  (time . 58)
  (add . 1)
  (compare . 3)
  (mod . 9)
  (move . 11)

```

```

                                (mult . 13)
                                (stout . 15)
                                (testi . 50)
                                (testo . 54))) then '1
elseif tm-unary-instructionp (lookup (car (instruction),
                                '(return . 0)
                                (wait . 4)
                                (run . 8)
                                (br . 2)
                                (brz . 10)
                                (brnz . 6)
                                (call . 14)
                                (decr . 18)
                                (decrm . 5)
                                (incr . 22)
                                (incrm . 7)
                                (lbase . 26)
                                (llimit . 30)
                                (lpsw . 34)
                                (post . 38)
                                (svc . 42)
                                (svcr . 46)
                                (time . 58)
                                (add . 1)
                                (compare . 3)
                                (mod . 9)
                                (move . 11)
                                (mult . 13)
                                (stout . 15)
                                (testi . 50)
                                (testo . 54))))
                                then '2
                                else '3 endif
else '0 endif

```

DEFINITION:

```

build-symtab (source, symtab, position)
= if listp (source)
  then if litatom (car (source))
    then build-symtab (cdr (source),
                      cons (cons (car (source), position), symtab),
                      position)
    elseif listp (car (source)  $\wedge$  (car (car (source)) = 'dcl)
    then build-symtab (cdr (source),

```

```

                                cons (cons (car (cdr (car (source))),
                                                car (cdr (cdr (car (source))))),
                                        symtab),
                                position)
elseif listp (car (source))  $\wedge$  (car (car (source)) = 'dc)
then build-symtab (cdr (source),
                    symtab,
                    position + incr-dc-position (car (source)))
else build-symtab (cdr (source),
                    symtab,
                    position + incr-instruction-position (car (source))) endif
else symtab endif

```

DEFINITION:

```

find-duplicate-symbol (symtab, already-seen)
= if listp (symtab)
  then if listp (car (symtab))
    then if car (car (symtab))  $\in$  already-seen then car (car (symtab))
      else find-duplicate-symbol (cdr (symtab),
                                cons (car (car (symtab)),
                                      already-seen)) endif
    else find-duplicate-symbol (cdr (symtab), already-seen) endif
  else '*1*false endif

```

DEFINITION:

```

assembly-errorp (assembler-result)
= ( $\neg$  (plistp (assembler-result)
               $\wedge$  ((length (assembler-result) = '2)
                    $\wedge$  (car (cdr (assembler-result)) = '*1*true))))

```

DEFINITION:

```

find-non-finite-number (l)
= if listp (l)
  then if finite-numberp (car (l), '65536)
    then find-non-finite-number (cdr (l))
    else car (l) endif
  else 'nil endif

```

DEFINITION:

```

assemble2 (memory)
= if finite-number-tablep ('100, table ('100, memory), '65536)
  then cons (memory, '*1*true)
  else cons ('nil,
             cons (cons ('assembly,
                         cons ('error,

```



```

                                cons (find-non-finite-number (memory),
                                        'nil))),
'nil)) endif

```

DEFINITION:

```

assemble1 (source, symtab)
= if find-duplicate-symbol (symtab, 'nil)
  then cons ('nil,
            cons (cons ('assembly,
                      cons ('error,
                            cons ('duplicate,
                                  cons ('label,
                                        cons (find-duplicate-symbol (symtab,
                                                                    'nil),
                                                                    'nil))))),
            'nil))
  else assemble2 (translate-source (source, symtab)) endif

```

DEFINITION:

```

assemble (source) = assemble1 (source, build-symtab (source, 'nil, '0))

```

EVENT: Let us define the theory *assembler* to consist of the following events: assemble, assemble1, assemble2, find-non-finite-number, assembly-errorp, find-duplicate-symbol, build-symtab, incr-instruction-position, incr-dc-position, translate-source, translate-instruction, asm-pack-instruction, asm-pack-word1, asm-good-instruction, asm-good-arg, asm-instruction-index2, asm-instruction-arg2, asm-instruction-mode2, asm-instruction-index1, asm-instruction-arg1, asm-instruction-mode1, asm-instruction-opcode, translate-dc, translate-dc-arglist, translate-dc-arg, dc-formp, number-and-litatom-listp, boundp, append-n-times, asm-opcode-alist.

DEFINITION:

```

reverse-alist (alist)
= if listp (alist)
  then cons (cons (cdr (car (alist)), car (car (alist))),
            reverse-alist (cdr (alist)))
  else alist endif

```

DEFINITION:

ASM-REVERSE-OPCODE-ALIST

```

= '( (0 . return)
      (4 . wait)
      (8 . run)
      (2 . br)

```

```

(10 . brz)
(6 . brnz)
(14 . call)
(18 . decr)
(5 . decrm)
(22 . incr)
(7 . incrm)
(26 . lbase)
(30 . llimit)
(34 . lpsw)
(38 . post)
(42 . svc)
(46 . svcr)
(58 . time)
(1 . add)
(3 . compare)
(9 . mod)
(11 . move)
(13 . mult)
(15 . stout)
(50 . testi)
(54 . testo))

```

DEFINITION:

```

asm-symbolic-opcode(n)
= lookup(tm-opcode(n),
        '(0 . return)
          (4 . wait)
          (8 . run)
          (2 . br)
          (10 . brz)
          (6 . brnz)
          (14 . call)
          (18 . decr)
          (5 . decrm)
          (22 . incr)
          (7 . incrm)
          (26 . lbase)
          (30 . llimit)
          (34 . lpsw)
          (38 . post)
          (42 . svc)
          (46 . svcr)
          (58 . time)

```

```
(1 . add)
(3 . compare)
(9 . mod)
(11 . move)
(13 . mult)
(15 . stout)
(50 . testi)
(54 . testo))
```

DEFINITION: OS-READYQ-LENGTH = '20

DEFINITION: OS-TASK-TABLE-LENGTH = '144

DEFINITION: OS-SEGMENT-TABLE-LENGTH = '32

DEFINITION: OS-STATUS-TABLE-LENGTH = '32

DEFINITION: OS-IBUFFER-LENGTH = '8

DEFINITION: OS-IBUFFERS-LENGTH = '128

DEFINITION: OS-OBUFFER-LENGTH = '8

DEFINITION: OS-OBUFFERS-LENGTH = '128

DEFINITION: OS-MBUFFER-LENGTH = '8

DEFINITION: OS-MBUFFERS-LENGTH = '2048

DEFINITION: OS-MBUFFERS-LENGTH-QUARTER = '512

THEOREM: os-mbuffers-length-is-divisible-by-4
'*1*true

DEFINITION:

OS-SOURCE

```
= '(dcl r0 0)
    (dcl r1 1)
    (dcl r2 2)
    (dcl r3 3)
    (dcl r4 4)
    (dcl r5 5)
    (dcl r6 6)
    (dcl r7 7)
    (dcl interrupt-pc-field 0)
    (dcl interrupt-sp-field 1)
    (dcl interrupt-flag-field 2)
```

```
(dcl svcid-addr 8)
(dcl input-devid-addr 8)
(dcl input-char-addr 9)
(dcl output-devid-addr 9)
(dcl charlub 256)
(dcl send-svcid 0)
(dcl receive-svcid 1)
(dcl tyo-svcid 2)
(dcl tyi-svcid 3)
(dcl task-table-length 144)
(dcl task-table-entry-length 9)
(dcl pc-field 0)
(dcl sp-field 1)
(dcl r2-field 2)
(dcl r3-field 3)
(dcl r4-field 4)
(dcl r5-field 5)
(dcl r6-field 6)
(dcl r7-field 7)
(dcl flag-field 8)
(dcl readyq-length 20)
(dcl qhead-field 0)
(dcl qtail-field 1)
(dcl qcurrlength-field 2)
(dcl qmaxlength-field 3)
(dcl qarray-field 4)
(dcl segment-table-length 32)
(dcl base-field 0)
(dcl limit-field 1)
(dcl status-entry-length 2)
(dcl status-flag-field 0)
(dcl status-taskid-field 1)
(dcl ready-status 0)
(dcl error-status 1)
(dcl send-status 2)
(dcl receive-status 3)
(dcl output-status 4)
(dcl input-status 5)
(dcl input-buffer-length 8)
(dcl output-buffer-length 8)
(dcl message-buffer-length 8)
(dcl source-multiplier 128)
(dcl dest-multiplier 8)
(dcl taskidlub 16)
```

```
reg-save-area
(dc 3 0)
clock-new-pc
(dc 1 clock-interrupt-handler)
error-new-pc
(dc 1 error-interrupt-handler)
svc-new-pc
(dc 1 svc-interrupt-handler)
input-new-pc
(dc 1 input-interrupt-handler)
output-new-pc
(dc 1 output-interrupt-handler)
interrupt-data
(dc 2 0)
branch-address
(dc 1 0)
time-slice
(dc 1 1000)
current-taskid
(dc 1 0)
temp-r2
(dc 1 0)
temp-r3
(dc 1 0)
task-table
(dc 144 0)
segment-table
(dc 32 0)
readyq
(dc 20 0)
status-table
(dc 32 0)
ibuffers
(dc 128 0)
obuffers
(dc 128 0)
mbuffers
(dc 512 0)
(dc 512 0)
(dc 512 0)
(dc 512 0)
save-state
(move (2 temp-r2) (1 r2))
(move (2 temp-r3) (1 r3))
```

```

(move (1 r3) readyq)
(call qfirst)
save-state-return
(mult (1 r2) task-table-entry-length)
(add (1 r2) task-table)
(move
  (3 r2 pc-field)
  (2 reg-save-area interrupt-pc-field))
(move
  (3 r2 sp-field)
  (2 reg-save-area interrupt-sp-field))
(move (3 r2 r2-field) (2 temp-r2))
(move (3 r2 r3-field) (2 temp-r3))
(move (3 r2 r4-field) (1 r4))
(move (3 r2 r5-field) (1 r5))
(move (3 r2 r6-field) (1 r6))
(move (3 r2 r7-field) (1 r7))
(add (1 r2) flag-field)
(move
  (3 r2)
  (2 reg-save-area interrupt-flag-field))
(move (1 r2) (2 temp-r2))
(move (1 r3) (2 temp-r3))
(return)
restore-state
(move (1 r3) (1 r2))
(mult (1 r2) task-table-entry-length)
(add (1 r2) task-table)
(mult (1 r3) 2)
(add (1 r3) segment-table)
(lbase (3 r3 base-field))
(llimit (3 r3 limit-field))
(move (1 r3) (1 r2))
(add (1 r3) flag-field)
(move
  (2 reg-save-area interrupt-pc-field)
  (3 r2 pc-field))
(move
  (2 reg-save-area interrupt-sp-field)
  (3 r2 sp-field))
(move
  (2 reg-save-area interrupt-flag-field)
  (3 r3))
(move (1 r7) (3 r2 r7-field))

```

```

(move (1 r6) (3 r2 r6-field))
(move (1 r5) (3 r2 r5-field))
(move (1 r4) (3 r2 r4-field))
(move (1 r3) (3 r2 r3-field))
(move (1 r2) (3 r2 r2-field))
(return)
clock-interrupt-handler
(call save-state)
trace-label1
(move (1 r3) readyq)
(call qfirst)
(call dequeue)
(call enqueue)
trace-label2
(br dispatcher)
error-interrupt-handler
(call save-state)
trace-label3
(move (1 r3) readyq)
(call qfirst)
(call dequeue)
trace-label4
(mult (1 r2) status-entry-length)
(add (1 r2) status-table)
(move (3 r2 status-flag-field) error-status)
(move (3 r2 status-taskid-field) 0)
(br dispatcher)
svc-interrupt-handler
(call save-state)
trace-label5
(mod (2 svcid-addr) 4)
(compare (2 svcid-addr) send-svcid)
(brz send-svc-handler)
(compare (2 svcid-addr) receive-svcid)
(brz receive-svc-handler)
(compare (2 svcid-addr) tyo-svcid)
(brz tyo-svc-handler)
(br tyi-svc-handler)
send-svc-handler
(move (1 r6) (1 r2))
(mod (1 r6) taskidlub)
(move (1 r7) (1 r3))
(move (1 r3) readyq)
(call qfirst)

```

```

trace-label6
(move (2 current-taskid) (1 r2))
(move (1 r4) (1 r6))
(mult (1 r4) dest-multiplier)
(move (1 r3) (1 r2))
(mult (1 r3) source-multiplier)
(add (1 r3) (1 r4))
(add (1 r3) mbuffers)
trace-label7
(call qfullp)
(brz block-send)
(move (1 r2) (1 r7))
(call enqueue)
trace-label8
(move (1 r3) (1 r6))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(compare (3 r3 status-flag-field) receive-status)
(brnz svc-resume-task)
(compare
  (3 r3 status-taskid-field)
  (2 current-taskid))
(brnz svc-resume-task)
(move (3 r3 status-flag-field) ready-status)
(move (3 r3 status-taskid-field) 0)
(move (1 r2) (1 r6))
(move (1 r3) readyq)
(call enqueue)
trace-label9
(br svc-resume-task)
block-send
(move (1 r3) readyq)
(call dequeue)
trace-label10
(move (1 r3) (2 current-taskid))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(move (3 r3 status-flag-field) send-status)
(move (3 r3 status-taskid-field) (1 r6))
(br dispatcher)
receive-svc-handler
(move (1 r6) (1 r2))
(mod (1 r6) taskidlub)
(move (1 r3) readyq)

```



```

(call qfirst)
trace-label11
(move (2 current-taskid) (1 r2))
(move (1 r4) (1 r2))
(mult (1 r4) dest-multiplier)
(move (1 r3) (1 r6))
(mult (1 r3) source-multiplier)
(add (1 r3) (1 r4))
(add (1 r3) mbuffers)
trace-label12
(call qempty)
(brz block-receive)
(call qfirst)
(call dequeue)
trace-label13
(move (1 r3) (2 current-taskid))
(mult (1 r3) task-table-entry-length)
(add (1 r3) task-table)
(move (3 r3 r3-field) (1 r2))
trace-label14
(move (1 r3) (1 r6))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(compare (3 r3 status-flag-field) send-status)
(brnz svc-resume-task)
(compare
  (3 r3 status-taskid-field)
  (2 current-taskid))
(brnz svc-resume-task)
(move (3 r3 status-flag-field) ready-status)
(move (3 r3 status-taskid-field) 0)
(move (1 r2) (1 r6))
(move (1 r3) readyq)
(call enqueue)
trace-label15
(br svc-resume-task)
block-receive
(move (1 r3) readyq)
(call dequeue)
trace-label16
(move (1 r3) (2 current-taskid))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(move (3 r3 status-flag-field) receive-status)

```

```

(move (3 r3 status-taskid-field) (1 r6))
(br dispatcher)
tyo-svc-handler
(move (1 r7) (1 r3))
(move (1 r3) readyq)
(call qfirst)
trace-label17
(move (2 current-taskid) (1 r2))
(move (1 r3) (1 r2))
(mult (1 r3) output-buffer-length)
(add (1 r3) obuffers)
trace-label18
(call qfullp)
(brz block-tyo)
(move (1 r2) (1 r7))
(call enqueue)
trace-label19
(testo (2 current-taskid))
(brnz svc-resume-task)
(post (2 current-taskid))
(br svc-resume-task)
block-tyo
(move (1 r3) readyq)
(call dequeue)
trace-label20
(move (1 r3) (2 current-taskid))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(move (3 r3 status-flag-field) output-status)
(move (3 r3 status-taskid-field) 0)
(br dispatcher)
tyi-svc-handler
(move (1 r3) readyq)
(call qfirst)
trace-label21
(move (2 current-taskid) (1 r2))
(move (1 r3) (1 r2))
(mult (1 r3) input-buffer-length)
(add (1 r3) ibuffers)
trace-label22
(call qempty)
(brz block-tyi)
(call qfirst)
(call dequeue)

```

```

trace-label23
(move (1 r3) (2 current-taskid))
(mult (1 r3) task-table-entry-length)
(add (1 r3) task-table)
(move (3 r3 r3-field) (1 r2))
(br svc-resume-task)
block-tyi
(move (1 r3) readyq)
(call dequeue)
trace-label24
(move (1 r3) (2 current-taskid))
(mult (1 r3) status-entry-length)
(add (1 r3) status-table)
(move (3 r3 status-flag-field) input-status)
(move (3 r3 status-taskid-field) 0)
(br dispatcher)
input-interrupt-handler
(move (2 branch-address) dispatcher)
(move (2 temp-r3) (1 r3))
(move (1 r3) readyq)
(call qempty)
trace-label25
(brz iih-skip-save-state)
(move (2 branch-address) resume-task)
(move (1 r3) (2 temp-r3))
(call save-state)
trace-label26 iih-skip-save-state
(move (1 r5) (2 input-devid-addr))
(mult (1 r5) status-entry-length)
(add (1 r5) status-table)
(compare (3 r5 status-flag-field) input-status)
(brnz check-for-full-input-buffer)
(move (1 r2) (2 input-devid-addr))
(move (1 r3) readyq)
(call enqueue)
trace-label27
(move (3 r5 status-flag-field) ready-status)
(move (3 r5 status-taskid-field) 0)
trace-label28 check-for-full-input-buffer
(move (1 r3) (2 input-devid-addr))
(mult (1 r3) input-buffer-length)
(add (1 r3) ibuffers)
(call qfullp)
(brnz check-for-iport-error)

```

```

(move (1 r2) (2 input-char-addr))
(add (1 r2) charlub)
(call qreplace)
trace-label29
(br (2 branch-address))
check-for-iport-error
(testi (2 input-devid-addr))
(brnz enqueue-input-character)
(move (1 r2) (2 input-char-addr))
(add (1 r2) charlub)
(call enqueue)
trace-label30
(br (2 branch-address))
enqueue-input-character
(move (1 r2) (2 input-char-addr))
(call enqueue)
trace-label31
(br (2 branch-address))
output-interrupt-handler
(move (2 branch-address) dispatcher)
(move (2 temp-r3) (1 r3))
(move (1 r3) readyq)
(call qempty)
trace-label32
(brz oih-skip-save-state)
(move (2 branch-address) resume-task)
(move (1 r3) (2 temp-r3))
(call save-state)
trace-label33 oih-skip-save-state
(move (1 r5) (2 output-devid-addr))
(mult (1 r5) status-entry-length)
(add (1 r5) status-table)
(compare (3 r5 status-flag-field) output-status)
(brnz check-for-empty-output-buffer)
(move (1 r2) (2 output-devid-addr))
(move (1 r3) readyq)
(call enqueue)
trace-label34
(move (3 r5 status-flag-field) ready-status)
(move (3 r5 status-taskid-field) 0)
trace-label35 check-for-empty-output-buffer
(move (1 r3) (2 output-devid-addr))
(mult (1 r3) output-buffer-length)
(add (1 r3) obuffers)

```

```

(call qempty)
(brz (2 branch-address))
(call qfirst)
(stout (2 output-devid-addr) (1 r2))
(call dequeue)
trace-label36
(br (2 branch-address))
dispatcher
(move (1 r3) readyq)
(call qempty)
dispatcher-trace-label1
(brz readyq-empty)
(call qfirst)
(call restore-state)
dispatcher-trace-label2
(time (2 time-slice 0))
(lpsw (2 reg-save-area))
readyq-empty
(wait)
pc-after-wait svc-resume-task
(move (1 r3) readyq)
(call qfirst)
(call restore-state)
svc-resume-task-trace-label1
(svcr (2 reg-save-area))
resume-task
(move (1 r3) readyq)
(call qfirst)
(call restore-state)
resume-task-trace-label1
(lpsw (2 reg-save-area))
enqueue
(move (1 r4) (1 r3))
(add (1 r4) qarray-field)
(add (1 r4) (3 r3 qtail-field))
(move (3 r4) (1 r2))
(incr (3 r3 qcurrlength-field))
(incrm (3 r3 qtail-field) (3 r3 qmaxlength-field))
(return)
qreplace
(move (1 r4) (3 r3 qtail-field))
(decrm (1 r4) (3 r3 qmaxlength-field))
(add (1 r4) (1 r3))
(add (1 r4) qarray-field)

```

```

(move (3 r4) (1 r2))
(return)
dequeue
(decr (3 r3 qcurrlength-field))
(incrm (3 r3 qhead-field) (3 r3 qmaxlength-field))
(return)
qfirst
(move (1 r2) (1 r3))
(add (1 r2) qarray-field)
(add (1 r2) (3 r3 qhead-field))
(move (1 r2) (3 r2))
(return)
qempty
(compare (3 r3 qcurrlength-field) 0)
(return)
qfullp
(compare
  (3 r3 qcurrlength-field)
  (3 r3 qmaxlength-field))
(return)
end-of-os-source)

```

THEOREM: clean-assembly
 '*1*true

DEFINITION:
 OS-SYMTAB
 = '(end-of-os-source . 3301)
 (qfullp . 3297)
 (qempty . 3293)
 (qfirst . 3280)
 (dequeue . 3274)
 (qreplace . 3258)
 (enqueue . 3240)
 (resume-task-trace-label1 . 3238)
 (resume-task . 3231)
 (svc-resume-task-trace-label1 . 3229)
 (svc-resume-task . 3222)
 (pc-after-wait . 3222)
 (readyq-empty . 3221)
 (dispatcher-trace-label2 . 3217)
 (dispatcher-trace-label1 . 3211)
 (dispatcher . 3206)
 (trace-label36 . 3204)

(check-for-empty-output-buffer . 3184)
(trace-label35 . 3184)
(trace-label34 . 3178)
(oih-skip-save-state . 3156)
(trace-label33 . 3156)
(trace-label32 . 3146)
(output-interrupt-handler . 3135)
(trace-label31 . 3133)
(enqueue-input-character . 3128)
(trace-label30 . 3126)
(check-for-iptort-error . 3114)
(trace-label29 . 3112)
(check-for-full-input-buffer . 3091)
(trace-label28 . 3091)
(trace-label27 . 3085)
(iih-skip-save-state . 3063)
(trace-label26 . 3063)
(trace-label25 . 3053)
(input-interrupt-handler . 3042)
(trace-label24 . 3025)
(block-tyi . 3020)
(trace-label23 . 3006)
(trace-label22 . 2998)
(trace-label21 . 2986)
(tyi-svc-handler . 2981)
(trace-label20 . 2964)
(block-tyo . 2959)
(trace-label19 . 2951)
(trace-label18 . 2942)
(trace-label17 . 2930)
(tyo-svc-handler . 2922)
(trace-label16 . 2905)
(block-receive . 2900)
(trace-label15 . 2898)
(trace-label14 . 2865)
(trace-label13 . 2853)
(trace-label12 . 2845)
(trace-label11 . 2824)
(receive-svc-handler . 2813)
(trace-label10 . 2796)
(block-send . 2791)
(trace-label9 . 2789)
(trace-label8 . 2756)
(trace-label7 . 2747)

(trace-label6 . 2726)
(send-svc-handler . 2712)
(trace-label5 . 2692)
(svc-interrupt-handler . 2690)
(trace-label4 . 2676)
(trace-label3 . 2669)
(error-interrupt-handler . 2667)
(trace-label2 . 2665)
(trace-label1 . 2656)
(clock-interrupt-handler . 2654)
(restore-state . 2601)
(save-state-return . 2558)
(save-state . 2547)
(mbuffers . 499)
(obuffers . 371)
(ibuffers . 243)
(status-table . 211)
(readyq . 191)
(segment-table . 159)
(task-table . 15)
(temp-r3 . 14)
(temp-r2 . 13)
(current-taskid . 12)
(time-slice . 11)
(branch-address . 10)
(interrupt-data . 8)
(output-new-pc . 7)
(input-new-pc . 6)
(svc-new-pc . 5)
(error-new-pc . 4)
(clock-new-pc . 3)
(reg-save-area . 0)
(taskidlub . 16)
(dest-multiplier . 8)
(source-multiplier . 128)
(message-buffer-length . 8)
(output-buffer-length . 8)
(input-buffer-length . 8)
(input-status . 5)
(output-status . 4)
(receive-status . 3)
(send-status . 2)
(error-status . 1)
(ready-status . 0)

(status-taskid-field . 1)
(status-flag-field . 0)
(status-entry-length . 2)
(limit-field . 1)
(base-field . 0)
(segment-table-length . 32)
(qarray-field . 4)
(qmaxlength-field . 3)
(qcurlength-field . 2)
(qtail-field . 1)
(qhead-field . 0)
(readyq-length . 20)
(flag-field . 8)
(r7-field . 7)
(r6-field . 6)
(r5-field . 5)
(r4-field . 4)
(r3-field . 3)
(r2-field . 2)
(sp-field . 1)
(pc-field . 0)
(task-table-entry-length . 9)
(task-table-length . 144)
(tyi-svcid . 3)
(tyo-svcid . 2)
(receive-svcid . 1)
(send-svcid . 0)
(charlub . 256)
(output-devid-addr . 9)
(input-char-addr . 9)
(input-devid-addr . 8)
(svcid-addr . 8)
(interrupt-flag-field . 2)
(interrupt-sp-field . 1)
(interrupt-pc-field . 0)
(r7 . 7)
(r6 . 6)
(r5 . 5)
(r4 . 4)
(r3 . 3)
(r2 . 2)
(r1 . 1)
(r0 . 0))

```

DEFINITION: OS-TASK-TABLE-ADDRESS = '15
DEFINITION: OS-READYQ-ADDRESS = '191
DEFINITION: OS-SEGMENT-TABLE-ADDRESS = '159
DEFINITION: OS-STATUS-TABLE-ADDRESS = '211
DEFINITION: OS-IBUFFERS-ADDRESS = '243
DEFINITION: OS-OBUFFERS-ADDRESS = '371
DEFINITION: OS-MBUFFERS-ADDRESS = '499
DEFINITION: OS-CLOCK-INTERRUPT-HANDLER-ADDRESS = '2654
DEFINITION: OS-ERROR-HANDLER-ADDRESS = '2667
DEFINITION: OS-SVC-HANDLER-ADDRESS = '2690
DEFINITION: OS-INPUT-INTERRUPT-HANDLER-ADDRESS = '3042
DEFINITION: OS-OUTPUT-INTERRUPT-HANDLER-ADDRESS = '3135
DEFINITION: OS-CODE-ADDRESS = '2547
DEFINITION: OS-SAVE-STATE-RETURN-ADDRESS = '2558
DEFINITION: OS-DISPATCHER-RETURN-ADDRESS = 'nil
DEFINITION: OS-TIME-SLICE-ADDRESS = '11
DEFINITION: OS-CODE-LENGTH = '754

DEFINITION:
nthcdr-n-times(n, k, l)
= if n  $\simeq$  0 then l
  else nthcdr-n-times(n - 1, k, nthcdr(k, l)) endif

DEFINITION:
firstn-n-times(n, k, l)
= if n  $\simeq$  0 then 'nil
  else append(firstn(k, l), firstn-n-times(n - 1, k, nthcdr(k, l))) endif

DEFINITION:
firstn-with-small-recursion(n, l, max)
= append(firstn(n mod max, l),
  firstn-n-times(n  $\div$  max, max, nthcdr(n mod max, l)))

```

DEFINITION:

$\text{nthcdr-with-small-recursion}(n, l, \text{max})$
 $= \text{nthcdr}(n \bmod \text{max}, \text{nthcdr-n-times}(n \div \text{max}, \text{max}, l))$

DEFINITION:

$\text{getseg-with-small-recursion}(n, k, l, \text{max})$
 $= \text{firstn-with-small-recursion}(k,$
 $\qquad \qquad \qquad \text{nthcdr-with-small-recursion}(n, l, \text{max}),$
 $\qquad \qquad \qquad \text{max})$

DEFINITION:

OS-MACHINE-CODE

```
= '(2187 13 2 2187 14 3 75 3 191 14 3280 77 2 9 65 2 15
    4299 2 0 12747 2 0 4811 2 13 5067 2 14 3275 2 4 3531
    2 5 3787 2 6 4043 2 7 65 2 8 20683 2 0 4171 2 13
    4171 3 14 0 2123 3 2 77 2 9 65 2 15 77 3 2 65 3 159
    218 3 478 3 2123 3 2 65 3 8 6283 0 2 14731 0 2 6795
    0 3 63563 7 2 55371 6 2 47179 5 2 38987 4 2 30795 3
    2 22603 2 2 0 14 2547 75 3 191 14 3280 14 3274 14
    3240 2 3206 14 2547 75 3 191 14 3280 14 3274 77 2 2
    65 2 211 203 2 1 459 2 0 2 3206 14 2547 137 8 4 131
    8 0 10 2712 131 8 1 10 2813 131 8 2 10 2922 2 2981
    2123 6 2 73 6 16 2123 7 3 75 3 191 14 3280 2187 12 2
    2123 4 6 77 4 8 2123 3 2 77 3 128 2113 3 4 65 3 499
    14 3297 10 2791 2123 2 7 14 3240 2123 3 6 77 3 2 65
    3 211 195 3 3 6 3222 4547 3 12 6 3222 203 3 0 459 3
    0 2123 2 6 75 3 191 14 3240 2 3222 75 3 191 14 3274
    4171 3 12 77 3 2 65 3 211 203 3 2 2507 3 6 2 3206
    2123 6 2 73 6 16 75 3 191 14 3280 2187 12 2 2123 4 2
    77 4 8 2123 3 6 77 3 128 2113 3 4 65 3 499 14 3293
    10 2900 14 3280 14 3274 4171 3 12 77 3 9 65 3 15
    3019 3 2 2123 3 6 77 3 2 65 3 211 195 3 2 6 3222
    4547 3 12 6 3222 203 3 0 459 3 0 2123 2 6 75 3 191
    14 3240 2 3222 75 3 191 14 3274 4171 3 12 77 3 2 65
    3 211 203 3 3 2507 3 6 2 3206 2123 7 3 75 3 191 14
    3280 2187 12 2 2123 3 2 77 3 8 65 3 371 14 3297 10
    2959 2123 2 7 14 3240 182 12 6 3222 166 12 2 3222 75
    3 191 14 3274 4171 3 12 77 3 2 65 3 211 203 3 4 459
    3 0 2 3206 75 3 191 14 3280 2187 12 2 2123 3 2 77 3
    8 65 3 243 14 3293 10 3020 14 3280 14 3274 4171 3 12
    77 3 9 65 3 15 3019 3 2 2 3222 75 3 191 14 3274 4171
    3 12 77 3 2 65 3 211 203 3 5 459 3 0 2 3206 139 10
    3206 2187 14 3 75 3 191 14 3293 10 3063 139 10 3231
    4171 3 14 14 2547 4171 5 8 77 5 2 65 5 211 195 5 5 6
```

```

3091 4171 2 8 75 3 191 14 3240 203 5 0 459 5 0 4171
3 8 77 3 8 65 3 243 14 3297 6 3114 4171 2 9 65 2 256
14 3258 130 10 178 8 6 3128 4171 2 9 65 2 256 14
3240 130 10 4171 2 9 14 3240 130 10 139 10 3206 2187
14 3 75 3 191 14 3293 10 3156 139 10 3231 4171 3 14
14 2547 4171 5 9 77 5 2 65 5 211 195 5 4 6 3184 4171
2 9 75 3 191 14 3240 203 5 0 459 5 0 4171 3 9 77 3 8
65 3 371 14 3293 138 10 14 3280 2191 9 2 14 3274 130
10 75 3 191 14 3293 10 3221 14 3280 14 2601 186 11
162 0 4 75 3 191 14 3280 14 2601 174 0 75 3 191 14
3280 14 2601 162 0 2123 4 3 65 4 4 14401 4 3 2251 4
2 726 3 31175 3 3 0 14411 4 3 30789 4 3 2113 4 3 65
4 4 2251 4 2 0 722 3 30919 3 3 0 2123 2 3 65 2 4
6209 2 3 6219 2 2 0 707 3 0 0 31427 3 3 0)

```

DEFINITION: $OS-LIMIT = '3311$

DEFINITION:

$os-task-table(os) = getseg('15, '144, tm-memory(os))$

DEFINITION: $os-readyq(os) = getseg('191, '20, tm-memory(os))$

DEFINITION:

$os-segment-table(os) = getseg('159, '32, tm-memory(os))$

DEFINITION:

$os-status-table(os) = getseg('211, '32, tm-memory(os))$

DEFINITION:

$os-ibuffers(os) = getseg('243, '128, tm-memory(os))$

DEFINITION:

$os-obuffers(os) = getseg('371, '128, tm-memory(os))$

DEFINITION:

$os-mbuffers(os) = getseg('499, '2048, tm-memory(os))$

DEFINITION: $os-code(os) = getseg('2547, '754, tm-memory(os))$

THEOREM: $length-os-task-table$

$length(os-task-table(os)) = '144$

EVENT: Disable $length-os-task-table$.

THEOREM: $plistp-os-task-table$

$plistp(os-task-table(os))$

EVENT: Disable plistp-os-task-table.

THEOREM: length-os-readyq
 $\text{length}(\text{os-readyq}(os)) = '20$

EVENT: Disable length-os-readyq.

THEOREM: plistp-os-readyq
 $\text{plistp}(\text{os-readyq}(os))$

EVENT: Disable plistp-os-readyq.

THEOREM: length-os-segment-table
 $\text{length}(\text{os-segment-table}(os)) = '32$

EVENT: Disable length-os-segment-table.

THEOREM: plistp-os-segment-table
 $\text{plistp}(\text{os-segment-table}(os))$

EVENT: Disable plistp-os-segment-table.

THEOREM: length-os-status-table
 $\text{length}(\text{os-status-table}(os)) = '32$

EVENT: Disable length-os-status-table.

THEOREM: plistp-os-status-table
 $\text{plistp}(\text{os-status-table}(os))$

EVENT: Disable plistp-os-status-table.

THEOREM: length-os-ibuffers
 $\text{length}(\text{os-ibuffers}(os)) = '128$

EVENT: Disable length-os-ibuffers.

THEOREM: plistp-os-ibuffers
 $\text{plistp}(\text{os-ibuffers}(os))$

EVENT: Disable plistp-os-ibuffers.

THEOREM: length-os-obuffers
length (os-obuffers (*os*)) = '128

EVENT: Disable length-os-obuffers.

THEOREM: plistp-os-obuffers
plistp (os-obuffers (*os*))

EVENT: Disable plistp-os-obuffers.

THEOREM: length-os-mbuffers
length (os-mbuffers (*os*)) = '2048

EVENT: Disable length-os-mbuffers.

THEOREM: plistp-os-mbuffers
plistp (os-mbuffers (*os*))

EVENT: Disable plistp-os-mbuffers.

DEFINITION: OS-CURRENT-TASKID-ADDRESS = '12

DEFINITION: OS-TRACE-LABEL1 = '2656

DEFINITION: OS-TRACE-LABEL2 = '2665

DEFINITION: OS-TRACE-LABEL3 = '2669

DEFINITION: OS-TRACE-LABEL4 = '2676

DEFINITION: OS-TRACE-LABEL5 = '2692

DEFINITION: OS-TRACE-LABEL6 = '2726

DEFINITION: OS-TRACE-LABEL7 = '2747

DEFINITION: OS-TRACE-LABEL8 = '2756

DEFINITION: OS-TRACE-LABEL9 = '2789

DEFINITION: OS-TRACE-LABEL10 = '2796

DEFINITION: OS-SVC-RECEIVE-HANDLER-ADDRESS = '2813

DEFINITION: OS-TRACE-LABEL11 = '2824

DEFINITION: OS-TRACE-LABEL12 = '2845
DEFINITION: OS-TRACE-LABEL13 = '2853
DEFINITION: OS-TRACE-LABEL14 = '2865
DEFINITION: OS-TRACE-LABEL15 = '2898
DEFINITION: OS-TRACE-LABEL16 = '2905
DEFINITION: OS-SVC-TYO-HANDLER-ADDRESS = '2922
DEFINITION: OS-TRACE-LABEL17 = '2930
DEFINITION: OS-TRACE-LABEL18 = '2942
DEFINITION: OS-TRACE-LABEL19 = '2951
DEFINITION: OS-TRACE-LABEL20 = '2964
DEFINITION: OS-SVC-TYI-HANDLER-ADDRESS = '2981
DEFINITION: OS-TRACE-LABEL21 = '2986
DEFINITION: OS-TRACE-LABEL22 = '2998
DEFINITION: OS-TRACE-LABEL23 = '3006
DEFINITION: OS-TRACE-LABEL24 = '3025
DEFINITION: OS-TRACE-LABEL25 = '3053
DEFINITION: OS-TRACE-LABEL26 = '3063
DEFINITION: OS-TRACE-LABEL27 = '3085
DEFINITION: OS-TRACE-LABEL28 = '3091
DEFINITION: OS-TRACE-LABEL29 = '3112
DEFINITION: OS-TRACE-LABEL30 = '3126
DEFINITION: OS-TRACE-LABEL31 = '3133
DEFINITION: OS-TRACE-LABEL32 = '3146
DEFINITION: OS-TRACE-LABEL33 = '3156
DEFINITION: OS-TRACE-LABEL34 = '3178

DEFINITION: OS-TRACE-LABEL35 = '3184

DEFINITION: OS-TRACE-LABEL36 = '3204

DEFINITION: OS-DISPATCHER-TRACE-LABEL1 = '3211

DEFINITION: OS-DISPATCHER-TRACE-LABEL2 = '3217

DEFINITION: OS-SVC-RESUME-TASK-ADDRESS = '3222

DEFINITION: OS-SVC-RESUME-TASK-TRACE-LABEL1 = '3229

DEFINITION: OS-RESUME-TASK-ADDRESS = '3231

DEFINITION: OS-RESUME-TASK-TRACE-LABEL1 = '3238

DEFINITION: OS-PC-AFTER-WAIT = '3222

DEFINITION: OS-TEMP-R2-ADDRESS = '13

DEFINITION: OS-TEMP-R3-ADDRESS = '14

DEFINITION: OS-DISPATCHER-ADDRESS = '3206

DEFINITION: OS-DISPATCHER-PC-AFTER-WAIT = '3222

DEFINITION: OS-DISPATCHER-QEMPTY-RETURN-ADDRESS = '3211

DEFINITION: OS-RESTORE-STATE-ADDRESS = '2601

THEOREM: length-os-code
length (os-code (os)) = '754

EVENT: Disable length-os-code.

THEOREM: plistp-os-code
plistp (os-code (os))

EVENT: Disable plistp-os-code.

DEFINITION:
finite-number-queuep (queue, capacity, lub)
= (array-queuep (queue)
 \wedge ((getnth ('3, queue) = capacity)
 \wedge finite-number-listp (getseg ('4, capacity, queue), lub)))

THEOREM: array-queuep-finite-number-queue
finite-number-queuep (*queue*, *capacity*, *lub*) \rightarrow array-queuep (*queue*)

EVENT: Disable array-queuep-finite-number-queue.

THEOREM: finite-number-queue-qmaxlength-field
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow (getnth ('3, *queue*) = *capacity*)

EVENT: Disable finite-number-queue-qmaxlength-field.

THEOREM: finite-number-listp-qarray
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow finite-number-listp (getseg ('4, *capacity*, *queue*), *lub*)

EVENT: Disable finite-number-listp-qarray.

THEOREM: length-finite-number-queue
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow (length (*queue*) = ('4 + *capacity*))

THEOREM: plistp-finite-number-queue
finite-number-queuep (*queue*, *capacity*, *lub*) \rightarrow plistp (*queue*)

THEOREM: finite-number-queue-qhead-size
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow ((getnth ('0, *queue*) \in \mathbf{N}) \wedge (getnth ('0, *queue*) $<$ *capacity*))

EVENT: Disable finite-number-queue-qhead-size.

THEOREM: finite-number-queue-qtail-size
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow ((getnth ('1, *queue*) \in \mathbf{N}) \wedge (getnth ('1, *queue*) $<$ *capacity*))

EVENT: Disable finite-number-queue-qtail-size.

THEOREM: finite-number-queue-qcurrenlength-size
finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow ((getnth ('2, *queue*) \in \mathbf{N}) \wedge (getnth ('2, *queue*) $<$ (1 + *capacity*)))

EVENT: Disable finite-number-queue-qcurrenlength-size.

THEOREM: finite-number-queue-qmaxlength-value
 finite-number-queuep(*queue*, *capacity*, *lub*)
 \rightarrow (getnth('3, *queue*) = *capacity*)

EVENT: Disable finite-number-queue-qmaxlength-value.

THEOREM: finite-number-queue-qarray-after-enq
 finite-number-queuep(*queue*, *capacity*, *lub*)
 \rightarrow (getseg('4, *capacity*, array-enq(*item*, *queue*))
 = putnth(*item*, getnth('1, *queue*), getseg('4, *capacity*, *queue*)))

EVENT: Disable finite-number-queue-qarray-after-enq.

THEOREM: finite-number-queuep-array-enq
 (finite-number-queuep(*queue*, *capacity*, *lub*)
 \wedge ((\neg array-qfullp(*queue*) \wedge finite-numberp(*item*, *lub*)))
 \rightarrow finite-number-queuep(array-enq(*item*, *queue*), *capacity*, *lub*))

EVENT: Disable finite-number-queuep-array-enq.

THEOREM: finite-number-queue-qarray-after-deq
 finite-number-queuep(*queue*, *capacity*, *lub*)
 \rightarrow (getseg('4, *capacity*, array-deq(*queue*)) = getseg('4, *capacity*, *queue*))

EVENT: Disable finite-number-queue-qarray-after-deq.

THEOREM: finite-number-queuep-array-deq
 (finite-number-queuep(*queue*, *capacity*, *lub*) \wedge (\neg array-qempty(*queue*)))
 \rightarrow finite-number-queuep(array-deq(*queue*), *capacity*, *lub*)

EVENT: Disable finite-number-queuep-array-deq.

THEOREM: array-qfirst-finite-number-queue
 (finite-number-queuep(*queue*, *capacity*, *lub*) \wedge (\neg array-qempty(*queue*)))
 \rightarrow ((array-qfirst(*queue*) \in \mathbf{N}) \wedge (array-qfirst(*queue*) < *lub*))

EVENT: Disable array-qfirst-finite-number-queue.

THEOREM: remainder-array-qfirst-finite-number-queue
 (finite-number-queuep(*queue*, *capacity*, *lub*) \wedge (\neg array-qempty(*queue*)))
 \rightarrow ((array-qfirst(*queue*) **mod** *lub*) = array-qfirst(*queue*))

EVENT: Disable remainder-array-qfirst-finite-number-queue.

EVENT: Let us define the theory *finite-number-queuep-facts* to consist of the following events: finite-number-queue-qmaxlength-value, finite-number-queue-qcurrlength-size, finite-number-queue-qtail-size, finite-number-queue-qhead-size, remainder-array-qfirst-finite-number-queue, array-qfirst-finite-number-queue, finite-number-queuep-array-deq, finite-number-queue-qarray-after-deq, finite-number-queuep-array-enq, finite-number-queue-qarray-after-enq, plistp-finite-number-queue, length-finite-number-queue, finite-number-listp-qarray, finite-number-queue-qmaxlength-field, array-queuep-finite-number-queue.

DEFINITION:

```
finite-number-queue-listp (list, capacity, lub)
= if listp (list)
  then finite-number-queuep (car (list), capacity, lub)
     $\wedge$  finite-number-queue-listp (cdr (list), capacity, lub)
  else '*1*true endif
```

THEOREM: finite-number-queuep-getnth
(finite-number-queue-listp (*l*, *capacity*, *lub*) \wedge (*i* < length (*l*)))
→ finite-number-queuep (getnth (*i*, *l*), *capacity*, *lub*)

EVENT: Disable finite-number-queuep-getnth.

THEOREM: finite-number-queue-listp-putnth
(finite-number-queue-listp (*l*, *capacity*, *lub*)
 \wedge finite-number-queuep (*queue*, *capacity*, *lub*))
→ finite-number-queue-listp (putnth (*queue*, *i*, *l*), *capacity*, *lub*)

EVENT: Disable finite-number-queue-listp-putnth.

DEFINITION:

```
finite-number-queue-tablep (n, l, capacity, lub)
= if listp (l)
  then plistp (car (l))
     $\wedge$  ((length (car (l)) = n)
       $\wedge$  (finite-number-queue-listp (car (l), capacity, lub)
         $\wedge$  finite-number-queue-tablep (n,
          cdr (l),
          capacity,
          lub)))
  else '*1*true endif
```

THEOREM: finite-number-queue-listp-getnth
 (finite-number-queue-tablep ($n, l, capacity, lub$) \wedge ($i < \text{length}(l)$))
 \rightarrow (plistp (getnth (i, l))
 \wedge ((length (getnth (i, l)) = n)
 \wedge finite-number-queue-listp (getnth (i, l), $capacity, lub$)))

EVENT: Disable finite-number-queue-listp-getnth.

THEOREM: finite-number-queuep-getnth2
 (finite-number-queue-tablep ($n, l, capacity, lub$)
 \wedge (($i < \text{length}(l)$) \wedge ($j < n$)))
 \rightarrow finite-number-queuep (getnth2 (i, j, l), $capacity, lub$)

EVENT: Disable finite-number-queuep-getnth2.

THEOREM: finite-number-queue-tablep-putnth
 (finite-number-queue-tablep ($n, l, capacity, lub$)
 \wedge (finite-number-queue-listp ($s, capacity, lub$)
 \wedge (plistp (s) \wedge (length (s) = n))))
 \rightarrow finite-number-queue-tablep ($n, \text{putnth} (s, i, l), capacity, lub$)

EVENT: Disable finite-number-queue-tablep-putnth.

THEOREM: finite-number-queue-tablep-putnth2
 (finite-number-queue-tablep ($n, l, capacity, lub$)
 \wedge (($i < \text{length}(l)$) \wedge finite-number-queuep ($queue, capacity, lub$)))
 \rightarrow finite-number-queue-tablep ($n, \text{putnth2} (queue, i, j, l), capacity, lub$)

EVENT: Disable finite-number-queue-tablep-putnth2.

THEOREM: finite-number-queue-listp-nthcdr
 finite-number-queue-listp ($l, capacity, lub$)
 \rightarrow finite-number-queue-listp (nthcdr (n, l), $capacity, lub$)

EVENT: Disable finite-number-queue-listp-nthcdr.

THEOREM: finite-number-queue-listp-firstn
 finite-number-queue-listp ($l, capacity, lub$)
 \rightarrow finite-number-queue-listp (firstn (n, l), $capacity, lub$)

EVENT: Disable finite-number-queue-listp-firstn.

THEOREM: finite-number-queue-listp-getseg
 (if length(l) < ($n + k$) then '1*false
 else '1*true endif
 ∧ finite-number-queue-listp(l , *capacity*, *lub*)
 → finite-number-queue-listp(getseg(n , k , l), *capacity*, *lub*)

EVENT: Disable finite-number-queue-listp-getseg.

THEOREM: divisor-not-lessp
 (($n \neq 0$) ∧ (($a \neq 0$) ∧ divides(n , a))) → ($a \not< n$)

THEOREM: divides-not-lessp-instance
 (($n \neq 0$) ∧ (listp(l) ∧ ((length(l) mod n) = '0)))
 → ((length(l) < n) = '1*false)

EVENT: Disable divides-not-lessp-instance.

THEOREM: finite-number-queue-tablep-table
 (($n \neq 0$)
 ∧ (((length(l) mod n) = '0)
 ∧ finite-number-queue-listp(l , *capacity*, *lub*)))
 → finite-number-queue-tablep(n , table(n , l), *capacity*, *lub*)

EVENT: Disable finite-number-queue-tablep-table.

DEFINITION: os-current-taskid(os) = array-qfirst(os-readyq(os))

DEFINITION:
 os-ready-set(os) = index-ready-set('0, table('2, os-status-table(os)))

DEFINITION:
 os-readyq-qarray(os)
 = getseg('4, getnth('3, os-readyq(os)), os-readyq(os))

```
(DEFN GOOD-OS
  (OS)
  (AND
    (GOOD-TM OS)
    (AND
      (EQUAL (TM-SLIMIT OS) '3311)
      (AND
        (EQUAL (OS-CODE OS)
```

'(2187 13 2 2187 14 3 75 3 191 14 3280 77 2 9 65 2 15 4299 2 0
12747 2 0 4811 2 13 5067 2 14 3275 2 4 3531 2 5 3787 2 6
4043 2 7 65 2 8 20683 2 0 4171 2 13 4171 3 14 0 2123 3 2 77
2 9 65 2 15 77 3 2 65 3 159 218 3 478 3 2123 3 2 65 3 8
6283 0 2 14731 0 2 6795 0 3 63563 7 2 55371 6 2 47179 5 2
38987 4 2 30795 3 2 22603 2 2 0 14 2547 75 3 191 14 3280 14
3274 14 3240 2 3206 14 2547 75 3 191 14 3280 14 3274 77 2 2
65 2 211 203 2 1 459 2 0 2 3206 14 2547 137 8 4 131 8 0 10
2712 131 8 1 10 2813 131 8 2 10 2922 2 2981 2123 6 2 73 6
16 2123 7 3 75 3 191 14 3280 2187 12 2 2123 4 6 77 4 8 2123
3 2 77 3 128 2113 3 4 65 3 499 14 3297 10 2791 2123 2 7 14
3240 2123 3 6 77 3 2 65 3 211 195 3 3 6 3222 4547 3 12 6
3222 203 3 0 459 3 0 2123 2 6 75 3 191 14 3240 2 3222 75 3
191 14 3274 4171 3 12 77 3 2 65 3 211 203 3 2 2507 3 6 2
3206 2123 6 2 73 6 16 75 3 191 14 3280 2187 12 2 2123 4 2
77 4 8 2123 3 6 77 3 128 2113 3 4 65 3 499 14 3293 10 2900
14 3280 14 3274 4171 3 12 77 3 9 65 3 15 3019 3 2 2123 3 6
77 3 2 65 3 211 195 3 2 6 3222 4547 3 12 6 3222 203 3 0 459
3 0 2123 2 6 75 3 191 14 3240 2 3222 75 3 191 14 3274 4171
3 12 77 3 2 65 3 211 203 3 3 2507 3 6 2 3206 2123 7 3 75 3
191 14 3280 2187 12 2 2123 3 2 77 3 8 65 3 371 14 3297 10
2959 2123 2 7 14 3240 182 12 6 3222 166 12 2 3222 75 3 191
14 3274 4171 3 12 77 3 2 65 3 211 203 3 4 459 3 0 2 3206 75
3 191 14 3280 2187 12 2 2123 3 2 77 3 8 65 3 243 14 3293 10
3020 14 3280 14 3274 4171 3 12 77 3 9 65 3 15 3019 3 2 2
3222 75 3 191 14 3274 4171 3 12 77 3 2 65 3 211 203 3 5 459
3 0 2 3206 139 10 3206 2187 14 3 75 3 191 14 3293 10 3063
139 10 3231 4171 3 14 14 2547 4171 5 8 77 5 2 65 5 211 195
5 5 6 3091 4171 2 8 75 3 191 14 3240 203 5 0 459 5 0 4171 3
8 77 3 8 65 3 243 14 3297 6 3114 4171 2 9 65 2 256 14 3258
130 10 178 8 6 3128 4171 2 9 65 2 256 14 3240 130 10 4171 2
9 14 3240 130 10 139 10 3206 2187 14 3 75 3 191 14 3293 10
3156 139 10 3231 4171 3 14 14 2547 4171 5 9 77 5 2 65 5 211
195 5 4 6 3184 4171 2 9 75 3 191 14 3240 203 5 0 459 5 0
4171 3 9 77 3 8 65 3 371 14 3293 138 10 14 3280 2191 9 2 14
3274 130 10 75 3 191 14 3293 10 3221 14 3280 14 2601 186 11
162 0 4 75 3 191 14 3280 14 2601 174 0 75 3 191 14 3280 14
2601 162 0 2123 4 3 65 4 4 14401 4 3 2251 4 2 726 3 31175 3
3 0 14411 4 3 30789 4 3 2113 4 3 65 4 4 2251 4 2 0 722 3
30919 3 3 0 2123 2 3 65 2 4 6209 2 3 6219 2 2 0 707 3 0 0
31427 3 3 0))

(AND
(EQUAL (GETNTH '3 (TM-MEMORY OS))
'2654)

```

(AND
  (EQUAL (GETNTH '4 (TM-MEMORY OS))
    '2667)
  (AND
    (EQUAL (GETNTH '5 (TM-MEMORY OS))
      '2690)
    (AND
      (EQUAL (GETNTH '6 (TM-MEMORY OS))
        '3042)
      (AND
        (EQUAL (GETNTH '7 (TM-MEMORY OS))
          '3135)
        (AND
          (EQUAL (GETNTH '11 (TM-MEMORY OS))
            '1000)
          (AND
            (NOT (TM-IN-SUPERVISOR-MODE OS))
            (AND
              (GOOD-CPU-LIST (TABLE '9 (OS-TASK-TABLE OS)))
              (AND
                (GOOD-STATUS-LIST (TABLE '2 (OS-STATUS-TABLE OS)))
                (AND
                  (FINITE-NUMBER-QUEUEP (OS-READYQ OS))
                  '16
                  '16)
                (AND
                  (MUTUALLY-DISJOINT (TABLE '2 (OS-SEGMENT-TABLE OS)))
                  (AND
                    (DISJOINT-EVERYWHERE '0
                      '3311
                      (TABLE '2 (OS-SEGMENT-TABLE OS)))
                    (AND
                      (FINITE-SEGMENT-TABLEP (TABLE '2 (OS-SEGMENT-TABLE OS))
                        '65536)
                      (AND
                        (FINITE-NUMBER-QUEUE-LISTP (TABLE '8 (OS-IBUFFERS OS))
                          '4
                          '65536)
                        (AND
                          (FINITE-NUMBER-QUEUE-LISTP (TABLE '8 (OS-OBUFFERS OS))
                            '4
                            '65536)
                          (AND
                            (FINITE-NUMBER-QUEUE-LISTP (TABLE '8 (OS-MBUFFERS OS))

```

```

'4
'65536)
(AND
(PERMUTATION (MAPUP-QUEUE (OS-READYQ OS))
(OS-READY-SET OS))
(AND
(IFF (TM-WAITING OS)
(AARRAY-QEMPTY (OS-READYQ OS)))
(IMPLIES
(NOT (TM-WAITING OS))
(AND
(EQUAL
(TM-BASE OS)
(BASE (GETNTH (OS-CURRENT-TASKID OS)
(TABLE '2 (OS-SEGMENT-TABLE OS))))))
(EQUAL
(TM-LIMIT OS)
(LIMIT
(GETNTH
(OS-CURRENT-TASKID OS)
(TABLE '2
(OS-SEGMENT-TABLE OS))))))))))))))))))))))))))))))))))))
NIL)

```

THEOREM: good-tm-os
good-os (*os*) → good-tm (*os*)

EVENT: Disable good-tm-os.

THEOREM: tm-slimit-os
good-os (*os*) → (tm-slimit (*os*) = '3311)

EVENT: Disable tm-slimit-os.

THEOREM: os-code-os
good-os (*os*)
→ (os-code (*os*)
= '(2187 13 2 2187 14 3 75 3 191 14 3280 77 2 9 65
2 15 4299 2 0 12747 2 0 4811 2 13 5067 2 14
3275 2 4 3531 2 5 3787 2 6 4043 2 7 65 2 8
20683 2 0 4171 2 13 4171 3 14 0 2123 3 2 77 2
9 65 2 15 77 3 2 65 3 159 218 3 478 3 2123 3
2 65 3 8 6283 0 2 14731 0 2 6795 0 3 63563 7
2 55371 6 2 47179 5 2 38987 4 2 30795 3 2

22603 2 2 0 14 2547 75 3 191 14 3280 14 3274
14 3240 2 3206 14 2547 75 3 191 14 3280 14
3274 77 2 2 65 2 211 203 2 1 459 2 0 2 3206 14
2547 137 8 4 131 8 0 10 2712 131 8 1 10 2813
131 8 2 10 2922 2 2981 2123 6 2 73 6 16 2123
7 3 75 3 191 14 3280 2187 12 2 2123 4 6 77 4
8 2123 3 2 77 3 128 2113 3 4 65 3 499 14 3297
10 2791 2123 2 7 14 3240 2123 3 6 77 3 2 65 3
211 195 3 3 6 3222 4547 3 12 6 3222 203 3 0
459 3 0 2123 2 6 75 3 191 14 3240 2 3222 75 3
191 14 3274 4171 3 12 77 3 2 65 3 211 203 3 2
2507 3 6 2 3206 2123 6 2 73 6 16 75 3 191 14
3280 2187 12 2 2123 4 2 77 4 8 2123 3 6 77 3
128 2113 3 4 65 3 499 14 3293 10 2900 14 3280
14 3274 4171 3 12 77 3 9 65 3 15 3019 3 2 2123
3 6 77 3 2 65 3 211 195 3 2 6 3222 4547 3 12
6 3222 203 3 0 459 3 0 2123 2 6 75 3 191 14
3240 2 3222 75 3 191 14 3274 4171 3 12 77 3 2
65 3 211 203 3 3 2507 3 6 2 3206 2123 7 3 75
3 191 14 3280 2187 12 2 2123 3 2 77 3 8 65 3
371 14 3297 10 2959 2123 2 7 14 3240 182 12 6
3222 166 12 2 3222 75 3 191 14 3274 4171 3 12
77 3 2 65 3 211 203 3 4 459 3 0 2 3206 75 3
191 14 3280 2187 12 2 2123 3 2 77 3 8 65 3 243
14 3293 10 3020 14 3280 14 3274 4171 3 12 77
3 9 65 3 15 3019 3 2 2 3222 75 3 191 14 3274
4171 3 12 77 3 2 65 3 211 203 3 5 459 3 0 2
3206 139 10 3206 2187 14 3 75 3 191 14 3293 10
3063 139 10 3231 4171 3 14 14 2547 4171 5 8 77
5 2 65 5 211 195 5 5 6 3091 4171 2 8 75 3 191
14 3240 203 5 0 459 5 0 4171 3 8 77 3 8 65 3
243 14 3297 6 3114 4171 2 9 65 2 256 14 3258
130 10 178 8 6 3128 4171 2 9 65 2 256 14 3240
130 10 4171 2 9 14 3240 130 10 139 10 3206
2187 14 3 75 3 191 14 3293 10 3156 139 10 3231
4171 3 14 14 2547 4171 5 9 77 5 2 65 5 211 195
5 4 6 3184 4171 2 9 75 3 191 14 3240 203 5 0
459 5 0 4171 3 9 77 3 8 65 3 371 14 3293 138
10 14 3280 2191 9 2 14 3274 130 10 75 3 191 14
3293 10 3221 14 3280 14 2601 186 11 162 0 4 75
3 191 14 3280 14 2601 174 0 75 3 191 14 3280
14 2601 162 0 2123 4 3 65 4 4 14401 4 3 2251
4 2 726 3 31175 3 3 0 14411 4 3 30789 4 3 2113
4 3 65 4 4 2251 4 2 0 722 3 30919 3 3 0 2123

2 3 65 2 4 6209 2 3 6219 2 2 0 707 3 0 0 31427
3 3 0))

EVENT: Disable os-code-os.

THEOREM: os-clock-new-pc
 $\text{good-os}(os) \rightarrow (\text{getnth}(3, \text{tm-memory}(os)) = 2654)$

EVENT: Disable os-clock-new-pc.

THEOREM: os-error-new-pc
 $\text{good-os}(os) \rightarrow (\text{getnth}(4, \text{tm-memory}(os)) = 2667)$

EVENT: Disable os-error-new-pc.

THEOREM: os-svc-new-pc
 $\text{good-os}(os) \rightarrow (\text{getnth}(5, \text{tm-memory}(os)) = 2690)$

EVENT: Disable os-svc-new-pc.

THEOREM: os-input-new-pc
 $\text{good-os}(os) \rightarrow (\text{getnth}(6, \text{tm-memory}(os)) = 3042)$

EVENT: Disable os-input-new-pc.

THEOREM: os-output-new-pc
 $\text{good-os}(os) \rightarrow (\text{getnth}(7, \text{tm-memory}(os)) = 3135)$

EVENT: Disable os-output-new-pc.

THEOREM: os-time-slice
 $\text{good-os}(os) \rightarrow (\text{getnth}(11, \text{tm-memory}(os)) = 1000)$

EVENT: Disable os-time-slice.

THEOREM: os-not-in-supervisor-mode
 $\text{good-os}(os) \rightarrow (\neg \text{tm-in-supervisor-mode}(os))$

EVENT: Disable os-not-in-supervisor-mode.

THEOREM: tm-base-os

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{tm-base}(os) \\
& = \text{base}(\text{getnth}(\text{os-current-taskid}(os), \\
& \quad \text{table}('2, \text{os-segment-table}(os))))))
\end{aligned}$$

EVENT: Disable tm-base-os.

THEOREM: tm-limit-os

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{tm-limit}(os) \\
& = \text{limit}(\text{getnth}(\text{os-current-taskid}(os), \\
& \quad \text{table}('2, \text{os-segment-table}(os))))))
\end{aligned}$$

EVENT: Disable tm-limit-os.

THEOREM: good-cpu-list-os-task-table

$$\text{good-os}(os) \rightarrow \text{good-cpu-list}(\text{table}('9, \text{os-task-table}(os)))$$

EVENT: Disable good-cpu-list-os-task-table.

THEOREM: good-status-list-os-status-table

$$\text{good-os}(os) \rightarrow \text{good-status-list}(\text{table}('2, \text{os-status-table}(os)))$$

EVENT: Disable good-status-list-os-status-table.

THEOREM: finite-number-queuep-os-readyq

$$\text{good-os}(os) \rightarrow \text{finite-number-queuep}(\text{os-readyq}(os), '16, '16)$$

EVENT: Disable finite-number-queuep-os-readyq.

THEOREM: mutually-disjoint-os-segment-table

$$\text{good-os}(os) \rightarrow \text{mutually-disjoint}(\text{table}('2, \text{os-segment-table}(os)))$$

EVENT: Disable mutually-disjoint-os-segment-table.

THEOREM: disjoint-everywhere-os-segment-table

$$\text{good-os}(os) \rightarrow \text{disjoint-everywhere}('0, '3311, \text{table}('2, \text{os-segment-table}(os)))$$

EVENT: Disable disjoint-everywhere-os-segment-table.

THEOREM: finite-segment-tablep-os-segment-table
good-os (*os*)
→ finite-segment-tablep (table ('2, os-segment-table (*os*)), '65536)

EVENT: Disable finite-segment-tablep-os-segment-table.

THEOREM: finite-number-queue-listp-os-ibuffers
good-os (*os*)
→ finite-number-queue-listp (table ('8, os-ibuffers (*os*)), '4, '65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers.

THEOREM: finite-number-queue-listp-os-obuffers
good-os (*os*)
→ finite-number-queue-listp (table ('8, os-obuffers (*os*)), '4, '65536)

EVENT: Disable finite-number-queue-listp-os-obuffers.

THEOREM: finite-number-queue-listp-os-mbuffers
good-os (*os*)
→ finite-number-queue-listp (table ('8, os-mbuffers (*os*)), '4, '65536)

EVENT: Disable finite-number-queue-listp-os-mbuffers.

THEOREM: finite-number-queue-tablep-os-mbuffers
good-os (*os*)
→ finite-number-queue-tablep ('16,
table ('16, table ('8, os-mbuffers (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queue-tablep-os-mbuffers.

THEOREM: permutation-os-readyq-qarray-os-ready-set
good-os (*os*) → permutation (mapup-queue (os-readyq (*os*)), os-ready-set (*os*))

EVENT: Disable permutation-os-readyq-qarray-os-ready-set.

THEOREM: tm-waiting-iff-array-qemptyp-os-readyq
good-os (*os*) → (tm-waiting (*os*) ↔ array-qemptyp (os-readyq (*os*)))

EVENT: Disable tm-waiting-iff-array-qemptyp-os-readyq.

DEFINITION: $\text{os-taskid}(os) = \text{array-qfirst}(\text{os-readyq}(os))$

THEOREM: $\text{finite-numberp-os-taskid}$
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{finite-numberp}(\text{os-taskid}(os), '16)$

EVENT: Disable $\text{finite-numberp-os-taskid}$.

EVENT: Let us define the theory *good-os-properties* to consist of the following events: $\text{finite-numberp-os-taskid}$, $\text{tm-waiting-iff-array-qempty-os-readyq}$, $\text{permutation-os-readyq-qarray-os-ready-set}$, $\text{finite-number-queue-tablep-os-mbuffers}$, $\text{finite-number-queue-listp-os-mbuffers}$, $\text{finite-number-queue-listp-os-obuffers}$, $\text{finite-number-queue-listp-os-ibuffers}$, $\text{finite-segment-tablep-os-segment-table}$, $\text{disjoint-everywhere-os-segment-table}$, $\text{mutually-disjoint-os-segment-table}$, $\text{finite-number-queuep-os-readyq}$, $\text{good-status-list-os-status-table}$, $\text{good-cpu-list-os-task-table}$, tm-limit-os , tm-base-os , $\text{os-not-in-supervisor-mode}$, os-time-slice , os-output-new-pc , os-input-new-pc , os-svc-new-pc , os-error-new-pc , os-clock-new-pc , os-code-os , tm-slimit-os , good-tm-os .

THEOREM: $\text{length-finite-number-queue-corollary}$
 $\text{finite-number-queuep}(queue, '16, '16) \rightarrow (\text{length}(queue) = '20)$

EVENT: Disable $\text{length-finite-number-queue-corollary}$.

THEOREM: $\text{plistsp-finite-number-queue-corollary}$
 $\text{finite-number-queuep}(queue, '16, '16) \rightarrow \text{plistsp}(queue)$

EVENT: Disable $\text{plistsp-finite-number-queue-corollary}$.

THEOREM: $\text{array-queuep-finite-number-queue-corollary}$
 $\text{finite-number-queuep}(queue, '16, '16) \rightarrow \text{array-queuep}(queue)$

EVENT: Disable $\text{array-queuep-finite-number-queue-corollary}$.

THEOREM: $\text{finite-number-queue-qhead-size-corollary}$
 $\text{finite-number-queuep}(queue, '16, '16)$
 $\rightarrow ((\text{getnth}('0, queue) \in \mathbf{N}) \wedge (\text{getnth}('0, queue) < '16))$

EVENT: Disable $\text{finite-number-queue-qhead-size-corollary}$.

THEOREM: $\text{finite-number-queue-qtail-size-corollary}$
 $\text{finite-number-queuep}(queue, '16, '16)$
 $\rightarrow ((\text{getnth}('1, queue) \in \mathbf{N}) \wedge (\text{getnth}('1, queue) < '16))$

EVENT: Disable finite-number-queue-qtail-size-corollary.

THEOREM: finite-number-queue-qcurlength-size-corollary
finite-number-queuep(*queue*, '16, '16)
→ ((getnth('2, *queue*) ∈ **N**) ∧ (getnth('2, *queue*) < '17))

EVENT: Disable finite-number-queue-qcurlength-size-corollary.

THEOREM: finite-number-queue-qmaxlength-value-corollary
finite-number-queuep(*queue*, '16, '16) → (getnth('3, *queue*) = '16)

EVENT: Disable finite-number-queue-qmaxlength-value-corollary.

THEOREM: finite-number-queuep-array-enq-finite-number-queue-corollary
(finite-number-queuep(*queue*, '16, '16)
∧ ((¬ array-qfullp(*queue*)) ∧ ((*item* ∈ **N**) ∧ (*item* < '16))))
→ finite-number-queuep(array-enq(*item*, *queue*), '16, '16)

EVENT: Disable finite-number-queuep-array-enq-finite-number-queue-corollary.

THEOREM: finite-number-queuep-array-deq-finite-number-queue-corollary
(finite-number-queuep(*queue*, '16, '16) ∧ (¬ array-qempty(*queue*)))
→ finite-number-queuep(array-deq(*queue*), '16, '16)

EVENT: Disable finite-number-queuep-array-deq-finite-number-queue-corollary.

THEOREM: array-qfirst-finite-number-queue-corollary
(finite-number-queuep(*queue*, '16, '16) ∧ (¬ array-qempty(*queue*)))
→ ((array-qfirst(*queue*) ∈ **N**) ∧ (array-qfirst(*queue*) < '16))

EVENT: Disable array-qfirst-finite-number-queue-corollary.

THEOREM: remainder-array-qfirst-finite-number-queue-corollary
(finite-number-queuep(*queue*, '16, '16) ∧ (¬ array-qempty(*queue*)))
→ ((array-qfirst(*queue*) **mod** '16) = array-qfirst(*queue*))

EVENT: Disable remainder-array-qfirst-finite-number-queue-corollary.

EVENT: Let us define the theory *finite-number-queue-corollaries* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary, array-qfirst-finite-number-queue-corollary, finite-number-queuep-array-deq-finite-number-queue-corollary, finite-number-queuep-array-enq-finite-number-queue-corollary,

finite-number-queue-qmaxlength-value-corollary, finite-number-queue-qcurrenlength-size-corollary, finite-number-queue-qtail-size-corollary, finite-number-queue-qhead-size-corollary, array-queuep-finite-number-queue-corollary, plistp-finite-number-queue-corollary, length-finite-number-queue-corollary.

THEOREM: length-finite-number-queue-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536) \rightarrow (length(*queue*) = '8)

EVENT: Disable length-finite-number-queue-corollary-for-mbuffers.

THEOREM: plistp-finite-number-queue-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536) \rightarrow plistp(*queue*)

EVENT: Disable plistp-finite-number-queue-corollary-for-mbuffers.

THEOREM: array-queuep-finite-number-queue-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536) \rightarrow array-queuep(*queue*)

EVENT: Disable array-queuep-finite-number-queue-corollary-for-mbuffers.

THEOREM: finite-number-queue-qhead-size-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536)
 \rightarrow ((getnth('0, *queue*) \in \mathbf{N}) \wedge (getnth('0, *queue*) < '4))

EVENT: Disable finite-number-queue-qhead-size-corollary-for-mbuffers.

THEOREM: finite-number-queue-qtail-size-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536)
 \rightarrow ((getnth('1, *queue*) \in \mathbf{N}) \wedge (getnth('1, *queue*) < '4))

EVENT: Disable finite-number-queue-qtail-size-corollary-for-mbuffers.

THEOREM: finite-number-queue-qcurrenlength-size-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536)
 \rightarrow ((getnth('2, *queue*) \in \mathbf{N}) \wedge (getnth('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrenlength-size-corollary-for-mbuffers.

THEOREM: finite-number-queue-qmaxlength-value-corollary-for-mbuffers
finite-number-queuep(*queue*, '4, '65536) \rightarrow (getnth('3, *queue*) = '4)

EVENT: Disable finite-number-queue-qmaxlength-value-corollary-for-mbuffers.

THEOREM: finite-number-queuep-array-enq-finite-number-queue-corollary-for-mbuffers
 $(\text{finite-number-queuep}(queue, '4, '65536)$
 $\wedge ((\neg \text{array-qfullp}(queue)) \wedge ((item \in \mathbf{N}) \wedge (item < '65536))))$
 $\rightarrow \text{finite-number-queuep}(\text{array-enq}(item, queue), '4, '65536)$

EVENT: Disable finite-number-queuep-array-enq-finite-number-queue-corollary-for-mbuffers.

THEOREM: finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers
 $(\text{finite-number-queuep}(queue, '4, '65536) \wedge (\neg \text{array-qemptyyp}(queue)))$
 $\rightarrow \text{finite-number-queuep}(\text{array-deq}(queue), '4, '65536)$

EVENT: Disable finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers.

THEOREM: array-qfirst-finite-number-queue-corollary-for-mbuffers
 $(\text{finite-number-queuep}(queue, '4, '65536) \wedge (\neg \text{array-qemptyyp}(queue)))$
 $\rightarrow ((\text{array-qfirst}(queue) \in \mathbf{N}) \wedge (\text{array-qfirst}(queue) < '65536))$

EVENT: Disable array-qfirst-finite-number-queue-corollary-for-mbuffers.

THEOREM: remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers
 $(\text{finite-number-queuep}(queue, '4, '65536) \wedge (\neg \text{array-qemptyyp}(queue)))$
 $\rightarrow ((\text{array-qfirst}(queue) \bmod '65536) = \text{array-qfirst}(queue))$

EVENT: Disable remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers.

THEOREM: finite-number-queue-listp-getnth-corollary
 $(\text{finite-number-queue-tablep}('16, l, '4, '65536) \wedge (i < \text{length}(l)))$
 $\rightarrow (\text{plistp}(\text{getnth}(i, l))$
 $\wedge ((\text{length}(\text{getnth}(i, l)) = '16)$
 $\wedge \text{finite-number-queue-listp}(\text{getnth}(i, l), '4, '65536)))$

EVENT: Disable finite-number-queue-listp-getnth-corollary.

THEOREM: finite-number-queuep-getnth2-corollary
 $(\text{finite-number-queue-tablep}('16, l, '4, '65536)$
 $\wedge ((i < \text{length}(l)) \wedge (j < '16)))$
 $\rightarrow \text{finite-number-queuep}(\text{getnth2}(i, j, l), '4, '65536)$

EVENT: Disable finite-number-queuep-getnth2-corollary.

THEOREM: finite-number-queue-tablep-putnth-corollary
 (finite-number-queue-tablep ('16, l, '4, '65536)
 ∧ (finite-number-queue-listp (s, '4, '65536)
 ∧ (plistp (s) ∧ (length (s) = '16))))
 → finite-number-queue-tablep ('16, putnth (s, i, l), '4, '65536)

EVENT: Disable finite-number-queue-tablep-putnth-corollary.

THEOREM: finite-number-queue-tablep-putnth2-corollary
 (finite-number-queue-tablep ('16, l, '4, '65536)
 ∧ ((i < length (l)) ∧ finite-number-queuep (queue, '4, '65536)))
 → finite-number-queue-tablep ('16, putnth2 (queue, i, j, l), '4, '65536)

EVENT: Disable finite-number-queue-tablep-putnth2-corollary.

EVENT: Let us define the theory *finite-number-queue-tablep-corollaries* to consist of the following events: finite-number-queue-tablep-putnth2-corollary, finite-number-queue-tablep-putnth-corollary, finite-number-queuep-getnth2-corollary, finite-number-queue-listp-getnth-corollary, remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers, array-qfirst-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-mbuffers, finite-number-queue-qmaxlength-value-corollary-for-mbuffers, finite-number-queue-qcurrlength-size-corollary-for-mbuffers, finite-number-queue-qtail-size-corollary-for-mbuffers, finite-number-queue-qhead-size-corollary-for-mbuffers, array-queuep-finite-number-queue-corollary-for-mbuffers, plistp-finite-number-queue-corollary-for-mbuffers, length-finite-number-queue-corollary-for-mbuffers.

THEOREM: number-of-ready-tasks-equals-length-os-readyq
 permutation (mapup-queue (os-readyq (os)), os-ready-set (os))
 → (number-of-ready-tasks (table ('2, os-status-table (os)))
 = length (mapup-queue (os-readyq (os))))

EVENT: Disable number-of-ready-tasks-equals-length-os-readyq.

THEOREM: os-waiting-to-sendp-implies-readyq-not-full
 (good-os (os)
 ∧ ((getnth ('2 * destid, os-status-table (os)) = '2)
 ∧ finite-numberp (destid, '16)))
 → (¬ array-qfullp (os-readyq (os)))

THEOREM: os-waiting-to-receivep-implies-readyq-not-full
 (good-os (os)

\wedge ((getnth ('2 * *destid*, os-status-table (*os*)) = '3)
 \wedge finite-numberp (*destid*, '16))
 \rightarrow (\neg array-qfullp (os-readyq (*os*)))

THEOREM: os-waiting-to-inputp-implies-readyq-not-full
 (good-os (*os*)
 \wedge ((getnth ('2 * *destid*, os-status-table (*os*)) = '5)
 \wedge finite-numberp (*destid*, '16))
 \rightarrow (\neg array-qfullp (os-readyq (*os*)))

THEOREM: os-waiting-to-outputp-implies-readyq-not-full
 (good-os (*os*)
 \wedge ((getnth ('2 * *destid*, os-status-table (*os*)) = '4)
 \wedge finite-numberp (*destid*, '16))
 \rightarrow (\neg array-qfullp (os-readyq (*os*)))

THEOREM: finite-number-queue-qarray-after-qreplace
 finite-number-queuep (*queue*, *capacity*, *lub*)
 \rightarrow (getseg ('4, *capacity*, array-qreplace (*item*, *queue*))
 $=$ putnth (*item*,
 decr-mod (getnth ('1, *queue*), getnth ('3, *queue*)),
 getseg ('4, *capacity*, *queue*))

THEOREM: finite-number-queuep-array-qreplace
 (finite-number-queuep (*queue*, *capacity*, *lub*)
 \wedge ((\neg array-qempty (*queue*)) \wedge finite-numberp (*item*, *lub*)))
 \rightarrow finite-number-queuep (array-qreplace (*item*, *queue*), *capacity*, *lub*)

EVENT: Disable finite-number-queuep-array-qreplace.

THEOREM: permutation-os-readyq-qarray-os-ready-set-corollary
 good-os (*os*)
 \rightarrow permutation (mapup-queue (os-readyq (*os*)),
 index-ready-set ('0, table ('2, os-status-table (*os*)))

EVENT: Disable permutation-os-readyq-qarray-os-ready-set-corollary.

DEFINITION:
 mapup-cpu (*taskid*, *os*)
 $=$ **if** tm-waiting (*os*) **then** getnth (*taskid*, table ('9, os-task-table (*os*)))
elseif *taskid* = os-current-taskid (*os*) **then** tm-cpu (*os*)
else getnth (*taskid*, table ('9, os-task-table (*os*))) **endif**

DEFINITION:
 mapup-regs (*taskid*, *os*) = getseg ('0, '8, mapup-cpu (*taskid*, *os*))

DEFINITION:
mapup-cc (*taskid*, *os*) = tm-unpack-cc (getnth ('8, mapup-cpu (*taskid*, *os*)))

DEFINITION:
mapup-error (*taskid*, *os*)
= tm-unpack-error (getnth ('8, mapup-cpu (*taskid*, *os*)))

DEFINITION:
mapup-svcflag (*taskid*, *os*)
= tm-unpack-svcflag (getnth ('8, mapup-cpu (*taskid*, *os*)))

DEFINITION:
mapup-svcid (*taskid*, *os*)
= tm-unpack-svcid (getnth ('8, mapup-cpu (*taskid*, *os*)))

DEFINITION:
mapup-base (*taskid*, *os*)
= base (getnth (*taskid*, table ('2, os-segment-table (*os*))))

DEFINITION:
mapup-limit (*taskid*, *os*)
= limit (getnth (*taskid*, table ('2, os-segment-table (*os*))))

DEFINITION:
mapup-task (*taskid*, *os*)
= mapup-address-space (tm-memory (*os*),
 mapup-regs (*taskid*, *os*),
 mapup-cc (*taskid*, *os*),
 mapup-error (*taskid*, *os*),
 mapup-svcflag (*taskid*, *os*),
 mapup-svcid (*taskid*, *os*),
 mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*))

DEFINITION:
mapup-tasks (*taskid*, *os*)
= **if** *taskid* < '16
 then cons (mapup-task (*taskid*, *os*), mapup-tasks (1 + *taskid*, *os*))
 else 'nil **endif**

DEFINITION: mapup-os-tasks (*os*) = mapup-tasks ('0, *os*)

DEFINITION:
mapup-queue-list (*l*)
= **if** listp (*l*) **then** cons (mapup-queue (car (*l*)), mapup-queue-list (cdr (*l*)))
 else 'nil **endif**

DEFINITION:

$\text{mapup-os-ibuffers}(os) = \text{mapup-queue-list}(\text{table}('8, \text{os-ibuffers}(os)))$

DEFINITION:

$\text{mapup-os-obuffers}(os) = \text{mapup-queue-list}(\text{table}('8, \text{os-obuffers}(os)))$

DEFINITION:

$\text{mapup-os-mbuffers}(os)$
 $= \text{table}('16, \text{mapup-queue-list}(\text{table}('8, \text{os-mbuffers}(os))))$

DEFINITION:

$\text{mapup-os}(os)$
 $= \text{ak}(\text{mapup-os-tasks}(os),$
 $\text{mapup-os-ibuffers}(os),$
 $\text{mapup-os-obuffers}(os),$
 $\text{mapup-os-mbuffers}(os),$
 $\text{mapup-queue}(\text{os-readyq}(os)),$
 $\text{table}('2, \text{os-status-table}(os)),$
 $\text{tm-rwstate}(os),$
 $\text{tm-clock}(os),$
 $\text{tm-ports}(os),$
 $\text{tm-oports}(os))$

THEOREM: $\text{length-mapup-queue-list}$

$\text{length}(\text{mapup-queue-list}(l)) = \text{length}(l)$

EVENT: Disable $\text{length-mapup-queue-list}$.

THEOREM: $\text{getnth-mapup-queue-list}$

$(i < \text{length}(l))$

$\rightarrow (\text{getnth}(i, \text{mapup-queue-list}(l)) = \text{mapup-queue}(\text{getnth}(i, l)))$

EVENT: Disable $\text{getnth-mapup-queue-list}$.

THEOREM: $\text{mapup-queue-list-putnth}$

$\text{mapup-queue-list}(\text{putnth}(v, i, l))$

$= \text{putnth}(\text{mapup-queue}(v), i, \text{mapup-queue-list}(l))$

EVENT: Disable $\text{mapup-queue-list-putnth}$.

THEOREM: $\text{mapup-queue-getnth}$

$(\text{finite-number-queue-listp}(l, \text{capacity}, \text{lub}) \wedge (id < \text{length}(l)))$

$\rightarrow (\text{mapup-queue}(\text{getnth}(id, l)) = \text{getnth}(id, \text{mapup-queue-list}(l)))$

EVENT: Disable mapup-queue-getnth.

THEOREM: length-mapup-tasks
 $\text{length}(\text{mapup-tasks}(taskid, os)) = ('16 - taskid)$

EVENT: Disable length-mapup-tasks.

THEOREM: length-mapup-os-tasks
 $\text{length}(\text{mapup-os-tasks}(os)) = '16$

EVENT: Disable length-mapup-os-tasks.

THEOREM: length-mapup-cpu
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{length}(\text{mapup-cpu}(taskid, os)) = '9)$

EVENT: Disable length-mapup-cpu.

THEOREM: finite-number-listp-mapup-cpu
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow \text{finite-number-listp}(\text{mapup-cpu}(taskid, os), '65536)$

EVENT: Disable finite-number-listp-mapup-cpu.

THEOREM: numberp-mapup-base
 $(\text{good-os}(os) \wedge (taskid < '16)) \rightarrow (\text{mapup-base}(taskid, os) \in \mathbf{N})$

EVENT: Disable numberp-mapup-base.

THEOREM: numberp-mapup-limit
 $(\text{good-os}(os) \wedge (taskid < '16)) \rightarrow (\text{mapup-limit}(taskid, os) \in \mathbf{N})$

EVENT: Disable numberp-mapup-limit.

THEOREM: plistp-mapup-regs
 $\text{plistp}(\text{mapup-regs}(taskid, os))$

EVENT: Disable plistp-mapup-regs.

THEOREM: length-mapup-regs
 $\text{length}(\text{mapup-regs}(taskid, os)) = '8$

EVENT: Disable length-mapup-regs.

THEOREM: finite-number-listp-mapup-regs
(good-os(*os*) \wedge (*taskid* < '16))
→ finite-number-listp (mapup-regs (*taskid*, *os*), '65536)

EVENT: Disable finite-number-listp-mapup-regs.

THEOREM: finite-numberp-mapup-cc
finite-numberp (mapup-cc (*taskid*, *os*), '4)

EVENT: Disable finite-numberp-mapup-cc.

THEOREM: finite-numberp-mapup-error
finite-numberp (mapup-error (*taskid*, *os*), '64)

EVENT: Disable finite-numberp-mapup-error.

THEOREM: finite-numberp-mapup-svcflag
finite-numberp (mapup-svcflag (*taskid*, *os*), '2)

EVENT: Disable finite-numberp-mapup-svcflag.

THEOREM: finite-numberp-mapup-svcid
(good-os(*os*) \wedge (*taskid* < '16))
→ finite-numberp (mapup-svcid (*taskid*, *os*), '128)

EVENT: Disable finite-numberp-mapup-svcid.

THEOREM: leq-plus-base-limit-getnth
(finite-segment-tablep (*l*, *max*) \wedge (*i* < length(*l*)))
→ **if** *max* < (base (getnth (*i*, *l*)) + limit (getnth (*i*, *l*))) **then** '*1*false
else '*1*true **endif**

THEOREM: leq-plus-base-limit-getnth-corollary
(finite-segment-tablep (*l*, '65536) \wedge (*i* < length(*l*)))
→ ('65536 $\not<$ (base (getnth (*i*, *l*)) + limit (getnth (*i*, *l*))))

EVENT: Disable leq-plus-base-limit-getnth-corollary.

THEOREM: leq-plus-mapup-base-mapup-limit
(good-os(*os*) \wedge (*taskid* < '16))
→ ('65536 $\not<$ (mapup-base (*taskid*, *os*) + mapup-limit (*taskid*, *os*)))

EVENT: Disable leq-plus-mapup-base-mapup-limit.

THEOREM: length-tm-memory-mapup-task
(good-os(*os*) \wedge (*taskid* < '16))
 \rightarrow (length(tm-memory(mapup-task(*taskid*, *os*))) = mapup-limit(*taskid*, *os*))

EVENT: Disable length-tm-memory-mapup-task.

THEOREM: good-address-space-mapup-task
(good-os(*os*) \wedge finite-numberp(*taskid*, '16))
 \rightarrow good-address-space(mapup-task(*taskid*, *os*),
length(tm-memory(mapup-task(*taskid*, *os*))))

EVENT: Disable good-address-space-mapup-task.

THEOREM: good-address-space-list-mapup-tasks
(good-os(*os*) \wedge finite-numberp(*taskid*, '16))
 \rightarrow good-address-space-list(mapup-tasks(*taskid*, *os*))

EVENT: Disable good-address-space-list-mapup-tasks.

THEOREM: good-address-space-list-mapup-os-tasks
good-os(*os*) \rightarrow good-address-space-list(mapup-os-tasks(*os*))

EVENT: Disable good-address-space-list-mapup-os-tasks.

THEOREM: length-mapup-os-ibuffers
length(mapup-os-ibuffers(*os*)) = '16

EVENT: Disable length-mapup-os-ibuffers.

THEOREM: good-task-buffer-mapup-queue
finite-number-queuep(*queue*, *capacity*, '65536)
 \rightarrow good-task-buffer(mapup-queue(*queue*), *capacity*)

EVENT: Disable good-task-buffer-mapup-queue.

THEOREM: good-task-buffer-list-mapup-queue-list
finite-number-queue-list(*l*, *capacity*, '65536)
 \rightarrow good-task-buffer-list(mapup-queue-list(*l*), *capacity*)

EVENT: Disable good-task-buffer-list-mapup-queue-list.

THEOREM: good-task-buffer-list-mapup-os-ibuffers
good-os (*os*) → good-task-buffer-list (mapup-os-ibuffers (*os*), '4)

EVENT: Disable good-task-buffer-list-mapup-os-ibuffers.

THEOREM: length-mapup-os-obuffers
length (mapup-os-obuffers (*os*)) = '16

EVENT: Disable length-mapup-os-obuffers.

THEOREM: good-task-buffer-list-mapup-os-obuffers
good-os (*os*) → good-task-buffer-list (mapup-os-obuffers (*os*), '4)

EVENT: Disable good-task-buffer-list-mapup-os-obuffers.

THEOREM: length-mapup-os-mbuffers
length (mapup-os-mbuffers (*os*)) = '16

EVENT: Disable length-mapup-os-mbuffers.

THEOREM: good-task-buffer-list-nthcdr
good-task-buffer-list (*l*, *capacity*)
→ good-task-buffer-list (nthcdr (*n*, *l*), *capacity*)

EVENT: Disable good-task-buffer-list-nthcdr.

THEOREM: good-task-buffer-list-firstn
good-task-buffer-list (*l*, *capacity*)
→ good-task-buffer-list (firstn (*n*, *l*), *capacity*)

EVENT: Disable good-task-buffer-list-firstn.

THEOREM: good-task-buffer-list-getseg
(good-task-buffer-list (*l*, *capacity*)
∧ (if length (*l*) < *n* then '*1*false
 else '*1*true endif
 ∧ if (length (*l*) - *n*) < *k* then '*1*false
 else '*1*true endif))
→ good-task-buffer-list (getseg (*n*, *k*, *l*), *capacity*)

EVENT: Disable good-task-buffer-list-getseg.

THEOREM: good-task-buffer-table-table
 (good-task-buffer-list (l , $capacity$)
 \wedge (((length(l) **mod** n) = '0) \wedge ($n \neq 0$)))
 \rightarrow good-task-buffer-table (table (n , l), n , $capacity$)

EVENT: Disable good-task-buffer-table-table.

THEOREM: good-task-buffer-table-mapup-os-mbuffers
 good-os (os) \rightarrow good-task-buffer-table (mapup-os-mbuffers (os), '16, '4)

EVENT: Disable good-task-buffer-table-mapup-os-mbuffers.

THEOREM: finite-number-listp-mapup-finite-number-queue
 finite-number-queuep ($queue$, $capacity$, lub)
 \rightarrow finite-number-listp (mapup-queue ($queue$), lub)

THEOREM: finite-number-listp-mapup-finite-number-queue-corollary
 good-os (os) \rightarrow finite-number-listp (mapup-queue (os-readyq (os)), '16)

EVENT: Disable finite-number-listp-mapup-finite-number-queue-corollary.

THEOREM: ak-ready-set-mapup-os
 ak-ready-set (mapup-os (os)) = os-ready-set (os)

EVENT: Disable ak-ready-set-mapup-os.

THEOREM: good-ak-mapup-os
 good-os (os) \rightarrow good-ak (mapup-os (os))

EVENT: Disable good-ak-mapup-os.

THEOREM: not-qemptyp-mapup-queue-os-readyq
 (good-os (os) \wedge (\neg tm-waiting (os)))
 \rightarrow (\neg qemptyp (mapup-queue (os-readyq (os))))

EVENT: Disable not-qemptyp-mapup-queue-os-readyq.

THEOREM: lessp-qfirst-mapup-queue-os-readyq
 (good-os (os) \wedge (\neg tm-waiting (os)))
 \rightarrow (qfirst (mapup-queue (os-readyq (os))) < '16)

EVENT: Disable lessp-qfirst-mapup-queue-os-readyq.

THEOREM: good-os-tm-set-error
(good-os(*os*) ∧ finite-numberp(*error*, '64))
→ good-os(tm-set-error(*error*, *os*))

EVENT: Disable good-os-tm-set-error.

THEOREM: good-os-tm-set-pc
(good-os(*os*) ∧ finite-numberp(*pc*, '65536)) → good-os(tm-set-pc(*pc*, *os*))

EVENT: Disable good-os-tm-set-pc.

THEOREM: good-os-tm-set-sp
(good-os(*os*) ∧ finite-numberp(*sp*, '65536)) → good-os(tm-set-sp(*sp*, *os*))

EVENT: Disable good-os-tm-set-sp.

THEOREM: good-os-tm-set-cc
(good-os(*os*) ∧ finite-numberp(*cc*, '4)) → good-os(tm-set-cc(*cc*, *os*))

EVENT: Disable good-os-tm-set-cc.

THEOREM: good-os-tm-set-svcflag
(good-os(*os*) ∧ finite-numberp(*svcflag*, '2))
→ good-os(tm-set-svcflag(*svcflag*, *os*))

EVENT: Disable good-os-tm-set-svcflag.

THEOREM: good-os-tm-set-svcid
(good-os(*os*) ∧ finite-numberp(*svcid*, '128))
→ good-os(tm-set-svcid(*svcid*, *os*))

EVENT: Disable good-os-tm-set-svcid.

THEOREM: good-os-tm-store-in-regmem
(good-os(*os*) ∧ finite-numberp(*value*, '65536))
→ good-os(tm-store-in-regmem(*value*, *addr*, *os*))

EVENT: Disable good-os-tm-store-in-regmem.

THEOREM: not-equal-plus-base-limit-0
(good-tm(*tm*)
∧ ((¬ tm-in-supervisor-mode(*tm*)))

$$\begin{aligned}
& \wedge (\text{tm-good-address}(addr, tm) \\
& \quad \wedge ((\text{real-addr-source}(addr) \neq 0) \\
& \quad \quad \wedge (\text{real-addr-source}(addr) \neq '1)))) \\
\rightarrow & ((\text{tm-base}(tm) + \text{tm-limit}(tm)) \neq '0)
\end{aligned}$$

THEOREM: equality-of-getseg-backchain-corollary

$$\begin{aligned}
& (\text{good-tm}(os) \\
& \quad \wedge ((\neg \text{tm-in-supervisor-mode}(os)) \\
& \quad \quad \wedge (\text{nested-intervals}('0, '3311, i, j) \\
& \quad \quad \quad \wedge (\text{disjoint-everywhere}('0, \\
& \quad \quad \quad \quad '3311, \\
& \quad \quad \quad \quad \text{table}('2, \text{os-segment-table}(os)))) \\
& \quad \quad \wedge ((\text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \quad \quad < \text{length}(\text{table}('2, \\
& \quad \quad \quad \quad \text{os-segment-table}(os)))) \\
& \quad \quad \wedge ((\text{tm-base}(os) \\
& \quad \quad \quad = \text{base}(\text{getnth}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \quad \quad \quad \quad \text{table}('2, \\
& \quad \quad \quad \quad \text{os-segment-table}(os)))))) \\
& \quad \quad \wedge ((\text{tm-limit}(os) \\
& \quad \quad \quad = \text{limit}(\text{getnth}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \quad \quad \quad \quad \text{table}('2, \\
& \quad \quad \quad \quad \text{os-segment-table}(os)))))) \\
& \quad \quad \wedge (\text{tm-good-address}(addr, \\
& \quad \quad \quad \quad os) \\
& \quad \quad \quad \quad \wedge ((\text{real-addr-source}(addr) \neq 0) \\
& \quad \quad \quad \quad \quad \wedge (\text{real-addr-source}(addr) \\
& \quad \quad \quad \quad \quad \neq '1))))))))) \\
\rightarrow & (\text{getseg}(i, \\
& \quad \quad j, \\
& \quad \quad \text{putnth}(value, \\
& \quad \quad \quad \text{tm-base}(os) + \text{real-addr-num}(addr), \\
& \quad \quad \quad \text{tm-memory}(os))) \\
& = \text{getseg}(i, j, \text{tm-memory}(os)))
\end{aligned}$$

EVENT: Disable equality-of-getseg-backchain-corollary.

THEOREM: equality-of-getnth-backchain-corollary

$$\begin{aligned}
& (\text{good-tm}(os) \\
& \quad \wedge ((\neg \text{tm-in-supervisor-mode}(os)) \\
& \quad \quad \wedge (\text{if } i < '0 \text{ then } '*1*false \\
& \quad \quad \quad \text{else } '*1*true \text{ endif} \\
& \quad \quad \wedge (((i - '0) < '3311) \\
& \quad \quad \quad \wedge (\text{disjoint-everywhere}('0,
\end{aligned}$$

$$\begin{aligned}
& \text{'3311,} \\
& \text{table ('2,} \\
& \quad \text{os-segment-table (os))} \\
\wedge & \left(\left(\text{array-qfirst (os-readyq (os))} \right. \right. \\
& \quad \left. \left. < \text{length (table ('2,} \right. \right. \\
& \quad \quad \left. \left. \text{os-segment-table (os))} \right) \right) \\
\wedge & \left(\left(\text{tm-base (os)} \right. \right. \\
& \quad \left. \left. = \text{base (getnth (array-qfirst (os-readyq (os)),} \right. \right. \\
& \quad \quad \left. \left. \text{table ('2,} \right. \right. \\
& \quad \quad \left. \left. \text{os-segment-table (os))} \right) \right) \\
\wedge & \left(\left(\text{tm-limit (os)} \right. \right. \\
& \quad \left. \left. = \text{limit (getnth (array-qfirst (os-readyq (os)),} \right. \right. \\
& \quad \quad \left. \left. \text{table ('2,} \right. \right. \\
& \quad \quad \left. \left. \text{os-segment-table (os))} \right) \right) \\
\wedge & \left(\text{tm-good-address (addr,} \right. \\
& \quad \left. \text{os)} \right. \\
& \quad \wedge \left(\left(\text{real-addr-source (addr) } \neq 0 \right) \right. \\
& \quad \quad \left. \wedge \left(\text{real-addr-source (addr)} \right. \right. \\
& \quad \quad \quad \left. \left. \neq \text{'1} \right) \right) \right) \\
\rightarrow & \left(\text{getnth (i,} \right. \\
& \quad \left. \text{putnth (value,} \right. \\
& \quad \quad \left. \text{tm-base (os) + real-addr-num (addr),} \right. \\
& \quad \quad \left. \text{tm-memory (os))} \right) \\
& = \text{getnth (i, tm-memory (os))}
\end{aligned}$$

EVENT: Disable equality-of-getnth-backchain-corollary.

THEOREM: good-os-tm-store-in-memory

$$\begin{aligned}
& \text{(good-os (os)} \\
& \quad \wedge \left(\left(\neg \text{tm-waiting (os)} \right) \right. \\
& \quad \quad \wedge \left(\left(\text{real-addr-source (addr) } \neq 0 \right) \right. \\
& \quad \quad \quad \wedge \left(\left(\text{real-addr-source (addr) } \neq \text{'1} \right) \right. \\
& \quad \quad \quad \quad \wedge \left(\text{tm-good-address (addr, os)} \right. \\
& \quad \quad \quad \quad \quad \left. \wedge \text{finite-numberp (value, '65536)} \right) \right) \right) \\
\rightarrow & \text{good-os (tm-store-in-memory (value, real-addr-num (addr), os))}
\end{aligned}$$

EVENT: Disable good-os-tm-store-in-memory.

THEOREM: good-os-tm-store

$$\begin{aligned}
& \text{(good-os (os)} \\
& \quad \wedge \left(\left(\neg \text{tm-waiting (os)} \right) \right. \\
& \quad \quad \left. \wedge \left(\text{tm-good-address (addr, os) } \wedge \text{finite-numberp (value, '65536)} \right) \right) \\
\rightarrow & \text{good-os (tm-store (value, addr, os))}
\end{aligned}$$

EVENT: Disable good-os-tm-store.

THEOREM: good-os-tm-execute-add
(good-os(os) \wedge (\neg tm-waiting(os)) \wedge tm-good-address($addr1$, os))
 \rightarrow good-os(tm-execute-add($addr1$, $addr2$, os))

EVENT: Disable good-os-tm-execute-add.

THEOREM: good-os-tm-execute-branch
(good-os(os) \wedge tm-good-address($addr$, os))
 \rightarrow good-os(tm-execute-branch($addr$, os))

EVENT: Disable good-os-tm-execute-branch.

THEOREM: good-os-tm-execute-branch-on-zero
(good-os(os) \wedge tm-good-address($addr$, os))
 \rightarrow good-os(tm-execute-branch-on-zero($addr$, os))

EVENT: Disable good-os-tm-execute-branch-on-zero.

THEOREM: good-os-tm-execute-branch-not-zero
(good-os(os) \wedge tm-good-address($addr$, os))
 \rightarrow good-os(tm-execute-branch-not-zero($addr$, os))

EVENT: Disable good-os-tm-execute-branch-not-zero.

THEOREM: good-os-tm-execute-call
(good-os(os) \wedge (\neg tm-waiting(os)) \wedge tm-good-address($addr$, os))
 \rightarrow good-os(tm-execute-call($addr$, os))

EVENT: Disable good-os-tm-execute-call.

THEOREM: good-os-tm-execute-compare
(good-os(os) \wedge tm-good-address($addr1$, os))
 \rightarrow good-os(tm-execute-compare($addr1$, $addr2$, os))

EVENT: Disable good-os-tm-execute-compare.

THEOREM: good-os-tm-execute-decr
(good-os(os) \wedge (\neg tm-waiting(os)) \wedge tm-good-address($addr$, os))
 \rightarrow good-os(tm-execute-decr($addr$, os))

EVENT: Disable good-os-tm-execute-decr.

THEOREM: good-os-tm-execute-decr-mod
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-good-address}(addr1, os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-decr-mod}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-decr-mod.

THEOREM: good-os-tm-execute-incr
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-good-address}(addr, os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-incr}(addr, os))$

EVENT: Disable good-os-tm-execute-incr.

THEOREM: good-os-tm-execute-incr-mod
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-good-address}(addr1, os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-incr-mod}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-incr-mod.

THEOREM: good-os-tm-execute-load-base
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-load-base}(addr, os))$

EVENT: Disable good-os-tm-execute-load-base.

THEOREM: good-os-tm-execute-load-limit
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-load-limit}(addr, os))$

EVENT: Disable good-os-tm-execute-load-limit.

THEOREM: good-os-tm-execute-load-psw
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-load-psw}(addr, os))$

EVENT: Disable good-os-tm-execute-load-psw.

THEOREM: good-os-tm-execute-mod
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-good-address}(addr1, os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-mod}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-mod.

THEOREM: good-os-tm-execute-move
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\text{tm-good-address}(addr1, os) \wedge \text{tm-good-address}(addr2, os))))$
 $\rightarrow \text{good-os}(\text{tm-execute-move}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-move.

THEOREM: good-os-tm-execute-multiply
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-good-address}(addr1, os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-multiply}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-multiply.

THEOREM: good-os-tm-execute-return
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-return}(os))$

EVENT: Disable good-os-tm-execute-return.

THEOREM: good-os-tm-execute-post-output-interrupt
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-post-output-interrupt}(addr, os))$

EVENT: Disable good-os-tm-execute-post-output-interrupt.

THEOREM: good-os-tm-execute-run
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \rightarrow \text{good-os}(\text{tm-execute-run}(os))$

EVENT: Disable good-os-tm-execute-run.

THEOREM: good-os-tm-execute-set-clock
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-set-clock}(addr, os))$

EVENT: Disable good-os-tm-execute-set-clock.

THEOREM: good-os-tm-execute-start-output
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-start-output}(addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-start-output.

THEOREM: good-os-tm-execute-svc
 $(\text{good-os}(os) \wedge \text{tm-good-address}(addr, os))$
 $\rightarrow \text{good-os}(\text{tm-execute-svc}(addr, os))$

EVENT: Disable good-os-tm-execute-svc.

THEOREM: good-os-tm-execute-svc-return
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-svc-return}(addr, os))$

EVENT: Disable good-os-tm-execute-svc-return.

THEOREM: good-os-tm-execute-test-iport
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-test-iport}(addr, os))$

EVENT: Disable good-os-tm-execute-test-iport.

THEOREM: good-os-tm-execute-test-oport
 $\text{good-os}(os) \rightarrow \text{good-os}(\text{tm-execute-test-oport}(addr, os))$

EVENT: Disable good-os-tm-execute-test-oport.

THEOREM: good-os-tm-execute-wait
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \rightarrow \text{good-os}(\text{tm-execute-wait}(os))$

EVENT: Disable good-os-tm-execute-wait.

THEOREM: good-os-tm-execute-nullary
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-nullary}(opcode, os))$

EVENT: Disable good-os-tm-execute-nullary.

THEOREM: good-os-tm-execute-unary
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-unary}(opcode, addr, os))$

EVENT: Disable good-os-tm-execute-unary.

THEOREM: good-os-tm-execute-binary
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-binary}(opcode, addr1, addr2, os))$

EVENT: Disable good-os-tm-execute-binary.

THEOREM: good-os-tm-execute-instruction
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{good-os}(\text{tm-execute-instruction}(instruction, os))$

EVENT: Disable good-os-tm-execute-instruction.

THEOREM: good-os-tm-execute
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow good-os(tm-execute(*instruction*, *os*))

EVENT: Disable good-os-tm-execute.

THEOREM: good-os-tm-decrement-clock
(good-os(*os*) \wedge (\neg tm-waiting(*os*))) \rightarrow good-os(tm-decrement-clock(*os*))

EVENT: Disable good-os-tm-decrement-clock.

THEOREM: good-os-tm-fetch-execute
(good-os(*os*) \wedge (\neg tm-waiting(*os*))) \rightarrow good-os(tm-fetch-execute(*os*))

EVENT: Disable good-os-tm-fetch-execute.

THEOREM: good-os-tm-set-iports
(good-os(*os*)
 \wedge (plistp(*iports*)
 \wedge ((length(*iports*) = '16) \wedge good-tm-iport-array(*iports*))))
 \rightarrow good-os(tm-set-iports(*iports*, *os*))

EVENT: Disable good-os-tm-set-iports.

THEOREM: good-os-tm-set-oports
(good-os(*os*)
 \wedge (plistp(*oports*)
 \wedge ((length(*oports*) = '16) \wedge good-tm-oport-array(*oports*))))
 \rightarrow good-os(tm-set-oports(*oports*, *os*))

EVENT: Disable good-os-tm-set-oports.

THEOREM: good-os-tm-post-interrupt
good-os(*os*) \rightarrow good-os(tm-post-interrupt(*interrupt*, *os*))

EVENT: Disable good-os-tm-post-interrupt.

DEFINITION:
disjoint-segments-induction(*i*, *j*, *l*)
= **if** *i* \simeq 0 **then** '0

```

elseif listp (l) then disjoint-segments-induction (i - 1, j - 1, cdr (l))
else '0 endif

```

THEOREM: mutually-disjoint-segments-are-disjoint-case1

```

(mutually-disjoint (l)
  ∧ ((i < length (l) ∧ ((j < length (l) ∧ (fix (i) < fix (j))))))
→ disjoint (base (getnth (i, l)),
            limit (getnth (i, l)),
            base (getnth (j, l)),
            limit (getnth (j, l)))

```

EVENT: Disable mutually-disjoint-segments-are-disjoint-case1.

THEOREM: mutually-disjoint-segments-are-disjoint

```

(mutually-disjoint (l)
  ∧ ((i < length (l) ∧ ((j < length (l) ∧ (fix (i) ≠ fix (j))))))
→ disjoint (base (getnth (i, l)),
            limit (getnth (i, l)),
            base (getnth (j, l)),
            limit (getnth (j, l)))

```

EVENT: Disable mutually-disjoint-segments-are-disjoint.

THEOREM: numberp-os-current-taskid

```

(good-os (os) ∧ (¬ tm-waiting (os)))
→ ((os-current-taskid (os) ∈ N) ∧ (os-current-taskid (os) < '16))

```

EVENT: Disable numberp-os-current-taskid.

THEOREM: tm-fetch-execute-mapup-address-space-separation

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)
      ∧ ((taskid ∈ N
          ∧ ((taskid < '16)
              ∧ (taskid ≠ os-current-taskid (os)))))))
→ (mapup-address-space (tm-memory (tm-fetch-execute (os)),
                                regs,
                                cc,
                                error,
                                svcflag,
                                svcid,
                                base (getnth (taskid,
                                              table ('2, os-segment-table (os))))),

```

$$\begin{aligned}
& \text{limit}(\text{getnth}(taskid, \\
& \quad \text{table}('2, \text{os-segment-table}(os)))) \\
= & \text{mapup-address-space}(\text{tm-memory}(os), \\
& \quad \text{regs}, \\
& \quad \text{cc}, \\
& \quad \text{error}, \\
& \quad \text{svcflag}, \\
& \quad \text{svcid}, \\
& \quad \text{base}(\text{getnth}(taskid, \\
& \quad \quad \text{table}('2, \\
& \quad \quad \quad \text{os-segment-table}(os))))), \\
& \quad \text{limit}(\text{getnth}(taskid, \\
& \quad \quad \text{table}('2, \\
& \quad \quad \quad \text{os-segment-table}(os))))))
\end{aligned}$$

EVENT: Disable tm-fetch-execute-mapup-address-space-separation.

THEOREM: getnth-tm-reglength-tm-cpu
 $\text{getnth}('8, \text{tm-cpu}(tm))$
 $= \text{tm-pack-psw}(\text{tm-cc}(tm), \text{tm-error}(tm), \text{tm-svcflag}(tm), \text{tm-svcid}(tm))$

EVENT: Disable getnth-tm-reglength-tm-cpu.

THEOREM: os-ibuffers-tm-fetch-execute
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{os-ibuffers}(\text{tm-fetch-execute}(os)) = \text{os-ibuffers}(os))$

EVENT: Disable os-ibuffers-tm-fetch-execute.

THEOREM: os-obuffers-tm-fetch-execute
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{os-obuffers}(\text{tm-fetch-execute}(os)) = \text{os-obuffers}(os))$

EVENT: Disable os-obuffers-tm-fetch-execute.

THEOREM: os-mbuffers-tm-fetch-execute
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{os-mbuffers}(\text{tm-fetch-execute}(os)) = \text{os-mbuffers}(os))$

EVENT: Disable os-mbuffers-tm-fetch-execute.

THEOREM: os-readyq-tm-fetch-execute
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{os-readyq}(\text{tm-fetch-execute}(os)) = \text{os-readyq}(os))$

EVENT: Disable os-readyq-tm-fetch-execute.

THEOREM: os-status-table-tm-fetch-execute
(good-os(os) \wedge (\neg tm-waiting(os)))
 \rightarrow (os-status-table(tm-fetch-execute(os)) = os-status-table(os))

EVENT: Disable os-status-table-tm-fetch-execute.

THEOREM: os-task-table-tm-fetch-execute
(good-os(os) \wedge (\neg tm-waiting(os)))
 \rightarrow (os-task-table(tm-fetch-execute(os)) = os-task-table(os))

EVENT: Disable os-task-table-tm-fetch-execute.

THEOREM: os-segment-table-tm-fetch-execute
(good-os(os) \wedge (\neg tm-waiting(os)))
 \rightarrow (os-segment-table(tm-fetch-execute(os)) = os-segment-table(os))

EVENT: Disable os-segment-table-tm-fetch-execute.

THEOREM: os-current-taskid-tm-fetch-execute
(good-os(os) \wedge (\neg tm-waiting(os)))
 \rightarrow (os-current-taskid(tm-fetch-execute(os)) = os-current-taskid(os))

EVENT: Disable os-current-taskid-tm-fetch-execute.

THEOREM: tm-waiting-tm-fetch-execute
good-os(os) \rightarrow (tm-waiting(tm-fetch-execute(os)) = tm-waiting(os))

EVENT: Disable tm-waiting-tm-fetch-execute.

EVENT: Let us define the theory *good-os-properties-crock* to consist of the following events: finite-numberp-os-taskid, tm-waiting-iff-array-qemptyp-os-readyq, permutation-os-readyq-qarray-os-ready-set, finite-number-queue-tablep-os-mbuffers, finite-number-queue-listp-os-obuffers, finite-number-queue-listp-os-ibuffers, finite-segment-tablep-os-segment-table, disjoint-everywhere-os-segment-table, mutually-disjoint-os-segment-table, finite-number-queuep-os-readyq, good-status-list-os-status-table, good-cpu-list-os-task-table, os-not-in-supervisor-mode, os-time-slice, os-output-new-pc, os-input-new-pc, os-svc-new-pc, os-error-new-pc, os-clock-new-pc, os-code-os, tm-slimit-os, good-tm-os.

THEOREM: times-commutativity-associativity-crock

$$(a * (c * b)) = ((a * b) * c)$$

EVENT: Disable times-commutativity-associativity-crock.

THEOREM: tm-unpack-cc-tm-pack-psw

$$\text{finite-numberp}(cc, '4)$$

$$\rightarrow (\text{tm-unpack-cc}(\text{tm-pack-psw}(cc, error, svcflag, svcid)) = cc)$$

EVENT: Disable tm-unpack-cc-tm-pack-psw.

THEOREM: remainder-plus-plus-times

$$((a + (b + (j * i))) \mathbf{mod} j) = ((a + b) \mathbf{mod} j)$$

EVENT: Disable remainder-plus-plus-times.

THEOREM: tm-unpack-error-tm-pack-psw

$$(\text{finite-numberp}(cc, '4) \wedge \text{finite-numberp}(error, '64))$$

$$\rightarrow (\text{tm-unpack-error}(\text{tm-pack-psw}(cc, error, svcflag, svcid)) = error)$$

EVENT: Disable tm-unpack-error-tm-pack-psw.

THEOREM: remainder-plus-plus-plus-times

$$((a + (b + (c + (i * j)))) \mathbf{mod} j) = ((a + (b + c)) \mathbf{mod} j)$$

EVENT: Disable remainder-plus-plus-plus-times.

THEOREM: quotient-plus-plus-times

$$((c + d) < b) \rightarrow (((c + (d + (b * a))) \div b) = \text{fix}(a))$$

EVENT: Disable quotient-plus-plus-times.

THEOREM: tm-unpack-svcflag-tm-pack-psw

$$(\text{finite-numberp}(cc, '4)$$

$$\wedge (\text{finite-numberp}(error, '64) \wedge \text{finite-numberp}(svcflag, '2)))$$

$$\rightarrow (\text{tm-unpack-svcflag}(\text{tm-pack-psw}(cc, error, svcflag, svcid)) = svcflag)$$

EVENT: Disable tm-unpack-svcflag-tm-pack-psw.

THEOREM: quotient-plus-plus-plus-times

$$((c + (d + e)) < b)$$

$$\rightarrow (((c + (d + (e + (b * a)))) \div b) = \text{fix}(a))$$

EVENT: Disable quotient-plus-plus-plus-times.

THEOREM: tm-unpack-svcid-tm-pack-psw
(finite-numberp (*cc*, '4)
 \wedge (finite-numberp (*error*, '64)
 \wedge (finite-numberp (*svcflag*, '2) \wedge finite-numberp (*svcid*, '128))))
→ (tm-unpack-svcid (tm-pack-psw (*cc*, *error*, *svcflag*, *svcid*)) = *svcid*)

EVENT: Disable tm-unpack-svcid-tm-pack-psw.

THEOREM: mapup-current-task-tm-fetch-execute
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
→ (mapup-task (os-current-taskid (*os*), tm-fetch-execute (*os*))
 = tm-fetch-execute (mapup-task (os-current-taskid (*os*), *os*)))

EVENT: Disable mapup-current-task-tm-fetch-execute.

THEOREM: mapup-task-separation
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((*taskid* \in \mathbf{N})
 \wedge ((*taskid* < '16)
 \wedge (*taskid* \neq os-current-taskid (*os*)))))))
→ (mapup-task (*taskid*, tm-fetch-execute (*os*)) = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-separation.

THEOREM: mapup-task-tm-fetch-execute-casesplit
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
→ (mapup-task (*taskid*, tm-fetch-execute (*os*))
 = **if** *taskid* = os-current-taskid (*os*)
 then tm-fetch-execute (mapup-task (os-current-taskid (*os*), *os*))
 else mapup-task (*taskid*, *os*) **endif**)

EVENT: Disable mapup-task-tm-fetch-execute-casesplit.

THEOREM: mapup-tasks-tm-fetch-execute-case1
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (os-current-taskid (*os*) < *taskid*))))
→ (mapup-tasks (*taskid*, tm-fetch-execute (*os*)) = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-tm-fetch-execute-case1.

THEOREM: getnth-mapup-tasks

$$\begin{aligned} & ((i < '16) \\ & \wedge (\text{if } i < j \text{ then } '*1*\text{false} \\ & \quad \text{else } '*1*\text{true} \text{ endif} \\ & \quad \wedge ((j \in \mathbf{N}) \wedge (i \in \mathbf{N})))) \\ \rightarrow & (\text{getnth}(i - j, \text{mapup-tasks}(j, os)) = \text{mapup-task}(i, os)) \end{aligned}$$

THEOREM: getnth-mapup-tasks-corollary2

$$\begin{aligned} & ((i < '16) \wedge (i \in \mathbf{N})) \\ \rightarrow & (\text{getnth}('0, \text{mapup-tasks}(i, os)) = \text{mapup-task}(i, os)) \end{aligned}$$

EVENT: Disable getnth-mapup-tasks-corollary2.

THEOREM: mapup-tasks-tm-fetch-execute-case2

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge ((taskid \in \mathbf{N}) \\ & \quad \quad \wedge (\text{if } \text{os-current-taskid}(os) < taskid \text{ then } '*1*\text{false} \\ & \quad \quad \quad \text{else } '*1*\text{true} \text{ endif})))) \\ \rightarrow & (\text{mapup-tasks}(taskid, \text{tm-fetch-execute}(os)) \\ & \quad = \text{putnth}(\text{tm-fetch-execute}(\text{getnth}(\text{os-current-taskid}(os) \\ & \quad \quad \quad - taskid, \\ & \quad \quad \quad \text{mapup-tasks}(taskid, os))), \\ & \quad \quad \quad \text{os-current-taskid}(os) - taskid, \\ & \quad \quad \quad \text{mapup-tasks}(taskid, os))) \end{aligned}$$

EVENT: Disable mapup-tasks-tm-fetch-execute-case2.

THEOREM: difference-zero-arg2

$$(n - '0) = \text{fix}(n)$$

EVENT: Disable difference-zero-arg2.

THEOREM: mapup-os-tasks-tm-fetch-execute

$$\begin{aligned} & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ \rightarrow & (\text{mapup-os-tasks}(\text{tm-fetch-execute}(os)) \\ & \quad = \text{putnth}(\text{tm-fetch-execute}(\text{getnth}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \\ & \quad \quad \quad \text{mapup-os-tasks}(os))), \\ & \quad \quad \quad \text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \\ & \quad \quad \quad \text{mapup-os-tasks}(os))) \end{aligned}$$

EVENT: Disable mapup-os-tasks-tm-fetch-execute.

THEOREM: correctness-of-tm-fetch-execute
(good-os(*os*) \wedge ((\neg tm-waiting(*os*)) \wedge (\neg tm-clock-interruptp(*os*))))
 \rightarrow (mapup-os(tm-fetch-execute(*os*)) = ak-private-step(mapup-os(*os*)))

EVENT: Disable correctness-of-tm-fetch-execute.

THEOREM: contract-array-enq
(*maxlength* = getnth('3, *queue*))
 \rightarrow (putnth(incr-mod(getnth('1, *queue*), *maxlength*),
 '1,
 putnth(1 + getnth('2, *queue*),
 '2,
 putnth(*item*, '4 + getnth('1, *queue*), *queue*)))
= array-enq(*item*, *queue*))

EVENT: Disable contract-array-enq.

THEOREM: contract-array-enq-instance
(array-queuep(*queue*) \wedge array-qempty(*queue*))
 \rightarrow ((*maxlength* = getnth('3, *queue*))
 \rightarrow (putnth(incr-mod(getnth('1, *queue*), *maxlength*),
 '1,
 putnth('1,
 '2,
 putnth(*item*, '4 + getnth('1, *queue*), *queue*)))
= array-enq(*item*, *queue*)))

EVENT: Disable contract-array-enq-instance.

THEOREM: contract-array-deq
(*maxlength* = getnth('3, *queue*))
 \rightarrow (putnth(incr-mod(getnth('0, *queue*), *maxlength*),
 '0,
 putnth(getnth('2, *queue*) - 1, '2, *queue*))
= array-deq(*queue*))

EVENT: Disable contract-array-deq.

THEOREM: contract-array-qfirst
getnth('4 + getnth('0, *queue*), *queue*) = array-qfirst(*queue*)

EVENT: Disable contract-array-qfirst.

THEOREM: contract-array-qreplace
(array-queuep (*queue*)
 $\wedge ((\neg \text{array-qempty}(queue)) \wedge (\text{maxlength} = \text{getnth}('3, queue)))$)
 $\rightarrow (\text{putnth}(item, '4 + \text{decr-mod}(\text{getnth}('1, queue), \text{maxlength}), queue)$
 $= \text{array-qreplace}(item, queue)$)

EVENT: Disable contract-array-qreplace.

EVENT: Let us define the theory *contract-queues* to consist of the following events: contract-array-qfirst, contract-array-enq-instance, contract-array-qreplace, contract-array-enq, contract-array-deq.

DEFINITION: $\text{tick}(n) = \text{fixlength}(n, 'nil, 'tick)$

THEOREM: tick-plus
 $\text{tick}(a + b) = \text{append}(\text{tick}(a), \text{tick}(b))$

THEOREM: getnth-os-code
good-os (*os*)
 $\rightarrow (\text{getnth}(n, \text{os-code}(os))$
 $= \text{getnth}(n,$
 '(2187 13 2 2187 14 3 75 3 191 14 3280 77
 2 9 65 2 15 4299 2 0 12747 2 0 4811 2 13
 5067 2 14 3275 2 4 3531 2 5 3787 2 6 4043
 2 7 65 2 8 20683 2 0 4171 2 13 4171 3 14
 0 2123 3 2 77 2 9 65 2 15 77 3 2 65 3 159
 218 3 478 3 2123 3 2 65 3 8 6283 0 2
 14731 0 2 6795 0 3 63563 7 2 55371 6 2
 47179 5 2 38987 4 2 30795 3 2 22603 2 2
 0 14 2547 75 3 191 14 3280 14 3274 14
 3240 2 3206 14 2547 75 3 191 14 3280 14
 3274 77 2 2 65 2 211 203 2 1 459 2 0 2
 3206 14 2547 137 8 4 131 8 0 10 2712 131
 8 1 10 2813 131 8 2 10 2922 2 2981 2123
 6 2 73 6 16 2123 7 3 75 3 191 14 3280
 2187 12 2 2123 4 6 77 4 8 2123 3 2 77 3
 128 2113 3 4 65 3 499 14 3297 10 2791
 2123 2 7 14 3240 2123 3 6 77 3 2 65 3 211
 195 3 3 6 3222 4547 3 12 6 3222 203 3 0
 459 3 0 2123 2 6 75 3 191 14 3240 2 3222
 75 3 191 14 3274 4171 3 12 77 3 2 65 3

211 203 3 2 2507 3 6 2 3206 2123 6 2 73
 6 16 75 3 191 14 3280 2187 12 2 2123 4 2
 77 4 8 2123 3 6 77 3 128 2113 3 4 65 3
 499 14 3293 10 2900 14 3280 14 3274 4171
 3 12 77 3 9 65 3 15 3019 3 2 2123 3 6 77
 3 2 65 3 211 195 3 2 6 3222 4547 3 12 6
 3222 203 3 0 459 3 0 2123 2 6 75 3 191
 14 3240 2 3222 75 3 191 14 3274 4171 3
 12 77 3 2 65 3 211 203 3 3 2507 3 6 2
 3206 2123 7 3 75 3 191 14 3280 2187 12 2
 2123 3 2 77 3 8 65 3 371 14 3297 10 2959
 2123 2 7 14 3240 182 12 6 3222 166 12 2
 3222 75 3 191 14 3274 4171 3 12 77 3 2
 65 3 211 203 3 4 459 3 0 2 3206 75 3 191
 14 3280 2187 12 2 2123 3 2 77 3 8 65 3
 243 14 3293 10 3020 14 3280 14 3274 4171
 3 12 77 3 9 65 3 15 3019 3 2 2 3222 75 3
 191 14 3274 4171 3 12 77 3 2 65 3 211 203
 3 5 459 3 0 2 3206 139 10 3206 2187 14 3
 75 3 191 14 3293 10 3063 139 10 3231 4171
 3 14 14 2547 4171 5 8 77 5 2 65 5 211 195
 5 5 6 3091 4171 2 8 75 3 191 14 3240 203
 5 0 459 5 0 4171 3 8 77 3 8 65 3 243 14
 3297 6 3114 4171 2 9 65 2 256 14 3258 130
 10 178 8 6 3128 4171 2 9 65 2 256 14 3240
 130 10 4171 2 9 14 3240 130 10 139 10
 3206 2187 14 3 75 3 191 14 3293 10 3156
 139 10 3231 4171 3 14 14 2547 4171 5 9
 77 5 2 65 5 211 195 5 4 6 3184 4171 2 9
 75 3 191 14 3240 203 5 0 459 5 0 4171 3
 9 77 3 8 65 3 371 14 3293 138 10 14 3280
 2191 9 2 14 3274 130 10 75 3 191 14 3293
 10 3221 14 3280 14 2601 186 11 162 0 4
 75 3 191 14 3280 14 2601 174 0 75 3 191
 14 3280 14 2601 162 0 2123 4 3 65 4 4
 14401 4 3 2251 4 2 726 3 31175 3 3 0
 14411 4 3 30789 4 3 2113 4 3 65 4 4 2251
 4 2 0 722 3 30919 3 3 0 2123 2 3 65 2 4
 6209 2 3 6219 2 2 0 707 3 0 0 31427 3 3
 0)))

EVENT: Disable getnth-os-code.

EVENT: Let us define the theory *tracing-lemmas* to consist of the following

events: getnth-os-code, open-up-tm-processor, tick, tm-processor-with-null-oracle.

EVENT: Let us define the theory *os-segment-lengths* to consist of the following events: length-os-task-table, plistp-os-task-table, length-os-segment-table, plistp-os-segment-table, length-os-readyq, plistp-os-readyq, length-os-status-table, plistp-os-status-table, length-os-ibuffers, plistp-os-ibuffers, length-os-obuffers, plistp-os-obuffers, length-os-mbuffers, plistp-os-mbuffers, length-os-code, plistp-os-code.

EVENT: Let us define the theory *trace-arithmetic* to consist of the following events: canonicalize-plus-terms1, canonicalize-plus-terms2, canonicalize-plus-terms3, commutativity-of-plus, commutativity-of-times, difference-x-x, difference-plus2, difference-plus-cancellation2-instance, plus-0-arg1, times-1, yet-another-difference-plus-crock, lessp-remainder2.

EVENT: Let us define the theory *finite-number-queue-corollaries1* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary, array-qfirst-finite-number-queue-corollary, finite-number-queuep-array-deq-finite-number-queue-corollary, finite-number-queuep-array-enq-finite-number-queue-corollary, finite-number-queue-qmaxlength-value-corollary, finite-number-queue-qcurrlength-size-corollary, finite-number-queue-qtail-size-corollary, finite-number-queue-qhead-size-corollary, array-queuep-finite-number-queue-corollary.

EVENT: Let us define the theory *good-os-properties1* to consist of the following events: finite-numberp-os-taskid, permutation-os-readyq-qarray-os-ready-set, finite-number-queue-listp-os-mbuffers, finite-number-queue-tablep-os-mbuffers, finite-number-queue-listp-os-obuffers, finite-number-queue-listp-os-ibuffers, finite-segment-tablep-os-segment-table, disjoint-everywhere-os-segment-table, mutually-disjoint-os-segment-table, finite-number-queuep-os-readyq, good-status-list-os-status-table, good-cpu-list-os-task-table, tm-limit-os, tm-base-os, os-not-in-supervisor-mode, os-time-slice, os-output-new-pc, os-input-new-pc, os-svc-new-pc, os-error-new-pc, os-clock-new-pc, tm-slimit-os, good-tm-os.

EVENT: Let us define the theory *memory-reference* to consist of the following events: getnth-putnth-coincidence, getnth-putnth-non-interference, getnth-after-putseg, getnth-before-putseg, getnth-putseg-coincidence, putnth-coincidence, putnth-non-interference-corollary1, putnth-non-interference-corollary2, putnth-after-putseg, putnth-putseg-coincidence, length-getseg, length-putseg, length-putnth.

EVENT: Let us define the theory *trace-array-queue-facts* to consist of the following events: array-qcurrlength-in-an-empty-queue, array-qcurrlength-in-a-non-

empty-queue, array-qcurrenlength-in-a-non-full-queue, array-qcurrenlength-in-a-full-queue, not-array-qfullp-array-deq, not-array-qemptyyp-array-enq, array-queuep-array-deq, plistp-array-deq, length-array-deq, array-queuep-array-enq, plistp-array-enq, length-array-enq, length-array-qarray, plistp-array-qarray, lessp-qcurrenlength-add1-qmaxlength, lessp-qtail-qmaxlength, lessp-qhead-qmaxlength, qmaxlength-greater-than-zero, qmaxlength-non-zero, numberp-qmaxlength, numberp-qcurrenlength, numberp-qtail, numberp-qhead.

DEFINITION: OS-READYQ-QARRAY-ADDRESS = '195

THEOREM: os-readyq-address-facts

$$\begin{aligned} & \text{finite-number-queuep}(queue, '16, '16) \\ \rightarrow & \left(\left(\left('195 + \text{getnth}('0, queue) \right) < '65536 \right) = '*1*true \right) \\ & \wedge \left(\left(\left('195 + \text{getnth}('1, queue) \right) < '65536 \right) = '*1*true \right) \end{aligned}$$

EVENT: Disable os-readyq-address-facts.

THEOREM: tm-pack-psw-tm-unpack-fields1

$$\begin{aligned} & \text{tm-pack-psw}(\text{tm-unpack-cc}(n), \\ & \quad \text{tm-unpack-error}(n), \\ & \quad \text{tm-unpack-svcflag}(n), \\ & \quad \text{tm-unpack-svcid}(n)) \\ = & \left((n \bmod '256) \right. \\ & \quad \left. + \left(('256 * ((n \bmod '512) \div '256)) \right. \right. \\ & \quad \left. \left. + ('512 * (n \div '512)) \right) \right) \end{aligned}$$

EVENT: Disable tm-pack-psw-tm-unpack-fields1.

THEOREM: tm-pack-psw-tm-unpack-fields2

$$\begin{aligned} & \left((n \bmod '256) \right. \\ & \quad \left. + \left(('256 * ((n \bmod '512) \div '256)) \right. \right. \\ & \quad \left. \left. + ('512 * (n \div '512)) \right) \right) \\ = & \left((n \bmod '512) + ('512 * (n \div '512)) \right) \end{aligned}$$

EVENT: Disable tm-pack-psw-tm-unpack-fields2.

THEOREM: tm-pack-psw-tm-unpack-fields

$$\begin{aligned} & (n \in \mathbf{N}) \\ \rightarrow & \left(\text{tm-pack-psw}(\text{tm-unpack-cc}(n), \right. \\ & \quad \text{tm-unpack-error}(n), \\ & \quad \text{tm-unpack-svcflag}(n), \\ & \quad \left. \text{tm-unpack-svcid}(n) \right) \\ = & n) \end{aligned}$$

EVENT: Disable tm-pack-psw-tm-unpack-fields.

THEOREM: list-of-getnths9
(plistp (l) \wedge (length (l) = '9))
→ (cons (getnth ('0, l),
 cons (getnth ('1, l),
 cons (getnth ('2, l),
 cons (getnth ('3, l),
 cons (getnth ('4, l),
 cons (getnth ('5, l),
 cons (getnth ('6, l),
 cons (getnth ('7, l),
 cons (getnth ('8, l), 'nil))))))))))

= l)

EVENT: Disable list-of-getnths9.

```
(PROVE-LEMMA CONTRACT-CPU
(REWRITE)
(IMPLIES
(GOOD-CPU CPU)
(EQUAL
(CONS
(GETNTH '0 CPU)
(CONS
(GETNTH '1 CPU)
(CONS
(GETNTH '2 CPU)
(CONS
(GETNTH '3 CPU)
(CONS
(GETNTH '4 CPU)
(CONS
(GETNTH '5 CPU)
(CONS (GETNTH '6 CPU)
(CONS (GETNTH '7 CPU)
(CONS (TM-PACK-PSW (TM-UNPACK-CC (GETNTH '8 CPU))
(TM-UNPACK-ERROR (GETNTH '8 CPU))
(TM-UNPACK-SVCFLAG (GETNTH '8 CPU))
(TM-UNPACK-SVCID (GETNTH '8 CPU)))
'NIL))))))))))
```

```

CPU))
((ENABLE TM-PACK-PSW-TM-UNPACK-FIELDS GOOD-CPU LIST-OF-GETNTHS9
  FINITE-NUMBERP-GETNTH OPEN-UP-GETNTH OPEN-UP-GETNTH-ON-ZERO)
 (USE (A-FINITE-NUMBER-IS-A-NUMBER (N (GETNTH '8 CPU))
    (LUB '65536)))
(ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable contract-cpu.

```

THEOREM: list-of-putnth8
(plistp (l) ^ (length (l) = '8))
→ (putnth (a,
           '0,
           putnth (b,
                  '1,
                  putnth (c,
                          '2,
                          putnth (d,
                                  '3,
                                  putnth (w,
                                          '4,
                                          putnth (x,
                                                  '5,
                                                  putnth (y,
                                                          '6,
                                                          putnth (z, '7, l))))))))))
= cons (a,
        cons (b,
              cons (c,
                    cons (d, cons (w, cons (x, cons (y, cons (z, 'nil))))))))))

```

EVENT: Disable list-of-putnth8.

```

THEOREM: contract-list-of-getnth8
((n ∈ N)
 ^ if length (l) < ('8 + n) then '*1*false
   else '*1*true endif)
→ (cons (getnth (n, l),
          cons (getnth ('1 + n, l),
                cons (getnth ('2 + n, l),
                      cons (getnth ('3 + n, l),
                              cons (getnth ('4 + n, l),

```

$$\begin{aligned}
& \text{cons}(\text{getnth}('5 + n, l), \\
& \quad \text{cons}(\text{getnth}('6 + n, l), \\
& \quad \quad \text{cons}(\text{getnth}('7 + n, l), 'nil))))))))) \\
= & \text{getseg}(n, '8, l)
\end{aligned}$$

EVENT: Disable contract-list-of-getnth8.

THEOREM: contract-sequence-of-putnth8

$$\begin{aligned}
& ((n \in \mathbf{N}) \\
& \wedge \text{if length}(l) < ('9 + n) \text{ then } '*1*\text{false} \\
& \quad \text{else } '*1*\text{true} \text{ endif}) \\
\rightarrow & (\text{putnth}(v0, \\
& \quad n, \\
& \quad \text{putnth}(v1, \\
& \quad \quad '1 + n, \\
& \quad \quad \text{putnth}(v2, \\
& \quad \quad \quad '2 + n, \\
& \quad \quad \quad \text{putnth}(v3, \\
& \quad \quad \quad \quad '3 + n, \\
& \quad \quad \quad \quad \text{putnth}(v4, \\
& \quad \quad \quad \quad \quad '4 + n, \\
& \quad \quad \quad \quad \quad \text{putnth}(v5, \\
& \quad \quad \quad \quad \quad \quad '5 + n, \\
& \quad \quad \quad \quad \quad \quad \text{putnth}(v6, \\
& \quad \quad \quad \quad \quad \quad \quad '6 + n, \\
& \quad \quad \quad \quad \quad \quad \quad \text{putnth}(v7, \\
& \quad \quad \quad \quad \quad \quad \quad \quad '7 + n, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \text{putnth}(v8, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad '8 \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad + n, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad l))))))))) \\
= & \text{putseg}(\text{cons}(v0, \\
& \quad \text{cons}(v1, \\
& \quad \quad \text{cons}(v2, \\
& \quad \quad \quad \text{cons}(v3, \\
& \quad \quad \quad \quad \text{cons}(v4, \\
& \quad \quad \quad \quad \quad \text{cons}(v5, \\
& \quad \quad \quad \quad \quad \quad \text{cons}(v6, \\
& \quad \quad \quad \quad \quad \quad \quad \text{cons}(v7, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \text{cons}(v8, 'nil))))))))) \\
& \quad n, \\
& \quad l))
\end{aligned}$$

EVENT: Disable contract-sequence-of-putnth8.

```

(PROVE-LEMMA
CONTRACT-PUTNTHS-INTO-TASK-TABLE
(REWRITE)
(IMPLIES
  (AND (GOOD-TM OS) (LESSP ID '16))
  (EQUAL
    (PUTNTH
      (GETNTH '0 (TM-REGS OS))
      (TIMES '9 ID)
      (PUTNTH
        (GETNTH '1 (TM-REGS OS))
        (PLUS '1 (TIMES '9 ID))
        (PUTNTH
          (GETNTH '2 (TM-REGS OS))
          (PLUS '2 (TIMES '9 ID))
          (PUTNTH
            (GETNTH '3 (TM-REGS OS))
            (PLUS '3 (TIMES '9 ID))
            (PUTNTH
              (GETNTH '4 (TM-REGS OS))
              (PLUS '4 (TIMES '9 ID))
              (PUTNTH
                (GETNTH '5 (TM-REGS OS))
                (PLUS '5 (TIMES '9 ID))
                (PUTNTH
                  (GETNTH '6 (TM-REGS OS))
                  (PLUS '6 (TIMES '9 ID))
                  (PUTNTH (GETNTH '7 (TM-REGS OS))
                    (PLUS '7 (TIMES '9 ID))
                    (PUTNTH (PLUS (TM-CC OS)
                      (PLUS (TIMES '4 (TM-ERROR OS))
                        (PLUS (TIMES '256 (TM-SVCFLAG OS))
                          (TIMES '512 (TM-SVCID OS))))))
                    (PLUS '8 (TIMES '9 ID))
                    (OS-TASK-TABLE OS))))))))))
    (PUTSEG (TM-CPU OS)
      (TIMES '9 ID)
      (OS-TASK-TABLE OS))))
((ENABLE TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4 TM-R5 TM-R6 TM-R7 TM-PACK-PSW
  COMMUTATIVITY-OF-PLUS COMMUTATIVITY-OF-TIMES
  CONTRACT-SEQUENCE-OF-PUTNTHS8 LENGTH-OS-TASK-TABLE)
(ENABLE-THEORY GOOD-TM-PROPERTIES GROUND-ZERO)

```



```
(DISABLE PLUS TIMES)
(DISABLE-THEORY T))
```

EVENT: Disable contract-putnth-into-task-table.

```
(PROVE-LEMMA
CONTRACT-PUTNTHS-INTO-REGMEM
(REWRITE)
(IMPLIES
  (AND (GOOD-TM OS)
        (AND (EQUAL (LENGTH TASK-TABLE) '144)
              (AND (NUMBERP TASKID)
                    (LESSP TASKID '16))))))
(EQUAL
  (PUTNTH
   (GETNTH (TIMES '9 TASKID) TASK-TABLE)
   '0
  (PUTNTH
   (GETNTH (PLUS '1 (TIMES '9 TASKID))
            TASK-TABLE)
   '1
  (PUTNTH
   (GETNTH (PLUS '2 (TIMES '9 TASKID))
            TASK-TABLE)
   '2
  (PUTNTH
   (GETNTH (PLUS '3 (TIMES '9 TASKID))
            TASK-TABLE)
   '3
  (PUTNTH (GETNTH (PLUS '4 (TIMES '9 TASKID))
                  TASK-TABLE)
   '4
  (PUTNTH (GETNTH (PLUS '5 (TIMES '9 TASKID))
                  TASK-TABLE)
   '5
  (PUTNTH (GETNTH (PLUS '6 (TIMES '9 TASKID))
                  TASK-TABLE)
   '6
  (PUTNTH (GETNTH (PLUS '7 (TIMES '9 TASKID))
                  TASK-TABLE)
   '7
  (TM-REGS OS))))))))))
```

```

(GETSEG '0
      '8
      (GETNTH TASKID
        (TABLE '9 TASK-TABLE))))
((ENABLE-THEORY GOOD-TM-PROPERTIES GROUND-ZERO)
 (ENABLE LIST-OF-PUTNTHS8 LENGTH-TABLE CONTRACT-LIST-OF-GETNTHS8 GETNTH-TABLE
   COMMUTATIVITY-OF-TIMES GETSEG-GETSEG PLUS-0)
 (DISABLE PLUS TIMES)
 (DISABLE-THEORY T))

```

EVENT: Disable contract-putnths-into-regmem.

```

(PROVE-LEMMA
 OS-TASK-TABLE-ADDRESS-FACTS
 (REWRITE)
 (IMPLIES
  (LESSP ID '16)
  (AND
   (EQUAL (LESSP (PLUS '15 (TIMES '9 ID))
                '65536)
            '*1*TRUE)
    (AND (EQUAL (LESSP (PLUS '16 (TIMES '9 ID))
                      '65536)
                '*1*TRUE)
         (AND (EQUAL (LESSP (PLUS '17 (TIMES '9 ID))
                          '65536)
                    '*1*TRUE)
              (AND (EQUAL (LESSP (PLUS '18 (TIMES '9 ID))
                                '65536)
                        '*1*TRUE)
                  (AND (EQUAL (LESSP (PLUS '19 (TIMES '9 ID))
                                    '65536)
                            '*1*TRUE)
                      (AND (EQUAL (LESSP (PLUS '20 (TIMES '9 ID))
                                        '65536)
                                '*1*TRUE)
                          (AND (EQUAL (LESSP (PLUS '21 (TIMES '9 ID))
                                            '65536)
                                    '*1*TRUE)
                              (AND (EQUAL (LESSP (PLUS '22 (TIMES '9 ID))
                                                '65536)
                                        '*1*TRUE)

```

```
(EQUAL (LESSP (PLUS '23 (TIMES '9 ID))
             '65536)
        '*1*TRUE))))))
```

```
((ENABLE-THEORY GROUND-ZERO)
 (DISABLE-THEORY T))
```

EVENT: Disable os-task-table-address-facts.

DEFINITION:

```
os-new-task-table (os)
= putseg (tm-cpu (os),
          '9 * array-qfirst (os-readyq (os)),
          os-task-table (os))
```

DEFINITION:

```
os-new-regs (taskid, tm)
= getseg ('0, '8, getnth (taskid, table ('9, os-new-task-table (tm))))
```

DEFINITION:

```
os-new-flags (taskid, tm)
= getnth ('8, getnth (taskid, table ('9, os-new-task-table (tm))))
```

DEFINITION:

```
os-new-cc (taskid, tm) = tm-unpack-cc (os-new-flags (taskid, tm))
```

DEFINITION:

```
os-new-error (taskid, tm) = tm-unpack-error (os-new-flags (taskid, tm))
```

DEFINITION:

```
os-new-svcflag (taskid, tm) = tm-unpack-svcflag (os-new-flags (taskid, tm))
```

DEFINITION:

```
os-new-svcid (taskid, tm) = tm-unpack-svcid (os-new-flags (taskid, tm))
```

DEFINITION:

```
os-new-base (taskid, os)
= base (getnth (taskid, table ('2, os-segment-table (os))))
```

DEFINITION:

```
os-new-limit (taskid, os)
= limit (getnth (taskid, table ('2, os-segment-table (os))))
```

EVENT: Let us define the theory *os-new-cpu* to consist of the following events: os-new-base, os-new-cc, os-new-error, os-new-flags, os-new-limit, os-new-regs, os-new-svcflag, os-new-svcid.

THEOREM: good-cpu-list-os-new-task-table
 (good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow good-cpu-list(table('9, os-new-task-table(*os*)))

EVENT: Disable good-cpu-list-os-new-task-table.

THEOREM: os-segment-table-address-facts
 (*id* < '16)
 \rightarrow (((('159 + ('2 * *id*) < '65536) = '*1*true)
 \wedge (((('160 + ('2 * *id*) < '65536) = '*1*true)
 \wedge (((('2 * *id*) < '32) = '*1*true)
 \wedge (('1 + ('2 * *id*) < '32)
 = '*1*true))))))

EVENT: Disable os-segment-table-address-facts.

THEOREM: contract-os-new-base-expression
 (('2 * *taskid*) < '32)
 \rightarrow (getnth('2 * *taskid*, os-segment-table(*os*))
 = base(getnth(*taskid*, table('2, os-segment-table(*os*))))))

EVENT: Disable contract-os-new-base-expression.

THEOREM: contract-os-new-limit-expression
 (('1 + ('2 * *taskid*) < '32)
 \rightarrow (getnth('1 + ('2 * *taskid*), os-segment-table(*os*))
 = limit(getnth(*taskid*, table('2, os-segment-table(*os*))))))

EVENT: Disable contract-os-new-limit-expression.

EVENT: Let us define the theory *access-os-segment-table* to consist of the following events: os-segment-table-address-facts, contract-os-new-base-expression, contract-os-new-limit-expression.

THEOREM: contract-sequence-of-putnth2
 ((*n* \in \mathbf{N})
 \wedge **if** length(*l*) < ('2 + *n*) **then** '*1*false
else '*1*true **endif**)
 \rightarrow (putnth(*a*, *n*, putnth(*b*, '1 + *n*, *l*))
 = putseg(cons(*a*, cons(*b*, 'nil)), *n*, *l*))

EVENT: Disable contract-sequence-of-putnth2.

THEOREM: contract-putnth-into-status-table

$(i < '16)$

\rightarrow (putnth(a , $'2 * i$, putnth(b , $'1 + ('2 * i)$, os-status-table(os)))
= putseg(cons(a , cons(b , 'nil)), $'2 * i$, os-status-table(os)))

EVENT: Disable contract-putnth-into-status-table.

THEOREM: os-status-table-address-facts

$(id < '16)$

\rightarrow (((('2 * id) < '65536) = '*1>true)
 \wedge (((('211 + ('2 * id)) < '65536) = '*1>true)
 \wedge (((('212 + ('2 * id)) < '65536)
= '*1>true)
 \wedge (((('2 * id) < '32) = '*1>true)
 \wedge (((('1 + ('2 * id)) < '32)
= '*1>true)
 \wedge (((('1 + ('2 * id))
< '32)
= '*1>true)
 \wedge (('32
< ('2
+ ('2
* id)))
= '*1>false))))))

EVENT: Disable os-status-table-address-facts.

THEOREM: access-os-status-table-crock

(good-tm(os) \wedge ($id < '16$))

\rightarrow ((getnth($'2 * id$, os-status-table(os)) $\in \mathbf{N}$)
 \wedge ((getnth($'2 * id$, os-status-table(os)) < '65536)
 \wedge ((getnth($'1 + ('2 * id$), os-status-table(os)) $\in \mathbf{N}$)
 \wedge (getnth($'1 + ('2 * id$,
os-status-table(os))
< '65536))))))

EVENT: Disable access-os-status-table-crock.

THEOREM: finite-numberp-getnth-os-status-table

(good-tm(os) \wedge ($i < '32$))

\rightarrow finite-numberp(getnth(i , os-status-table(os)), '65536)

EVENT: Let us define the theory *access-os-status-table* to consist of the following

events: contract-putnth-into-status-table, os-status-table-address-facts, access-os-status-table-crock.

DEFINITION:

os-update-status(*id*, *status*, *flag*, *os*)
= putseg (cons (*status*, cons (*flag*, 'nil)), '2 * *id*, os-status-table (*os*))

THEOREM: re-associate-times

$$((1 + a) * ((1 + b) * c)) = (((1 + a) * (1 + b)) * c)$$

EVENT: Disable re-associate-times.

THEOREM: limited-associativity-of-plus

$$(((1 + a) + (b + c)) + d) = ((1 + a) + ((b + c) + d))$$

EVENT: Disable limited-associativity-of-plus.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-mbuffers* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers, array-qfirst-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-mbuffers, finite-number-queue-qmaxlength-value-corollary-for-mbuffers, finite-number-queue-qcurrlength-size-corollary-for-mbuffers, finite-number-queue-qtail-size-corollary-for-mbuffers, finite-number-queue-qhead-size-corollary-for-mbuffers, array-queuep-finite-number-queue-corollary-for-mbuffers, plistp-finite-number-queue-corollary-for-mbuffers, length-finite-number-queue-corollary-for-mbuffers.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-mbuffers1* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers, array-qfirst-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-mbuffers, finite-number-queue-qcurrlength-size-corollary-for-mbuffers, finite-number-queue-qtail-size-corollary-for-mbuffers, finite-number-queue-qhead-size-corollary-for-mbuffers, array-queuep-finite-number-queue-corollary-for-mbuffers, plistp-finite-number-queue-corollary-for-mbuffers, length-finite-number-queue-corollary-for-mbuffers.

DEFINITION: OS-MBUFFER-SOURCE-MULTIPLIER = '128

DEFINITION: OS-MBUFFER-DEST-MULTIPLIER = '8

DEFINITION: OS-MBUFFERS-ADDRESS-PLUS-QTAIL-FIELD = '500

DEFINITION: OS-MBUFFERS-ADDRESS-PLUS-QCURRLENGTH-FIELD = '501

DEFINITION: OS-MBUFFERS-ADDRESS-PLUS-QMAXLENGTH-FIELD = '502

DEFINITION: OS-MBUFFERS-ADDRESS-PLUS-QARRAY-FIELD = '503

DEFINITION:

os-current-mbuffer (*sid*, *did*, *os*)
= getnth (('16 * *sid*) + *did*, table ('8, os-mbuffers (*os*)))

THEOREM: finite-number-queuep-current-mbuffer
(good-os (*os*) ∧ ((*sid* < '16) ∧ (*did* < '16)))
→ finite-number-queuep (getnth (('16 * *sid*) + *did*,
table ('8, os-mbuffers (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queuep-current-mbuffer.

THEOREM: finite-number-queuep-current-mbuffer-for-receive
(good-os (*os*) ∧ ((*sid* < '16) ∧ (*did* < '16)))
→ finite-number-queuep (getnth (*did* + ('16 * *sid*),
table ('8, os-mbuffers (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queuep-current-mbuffer-for-receive.

THEOREM: re-associate-times1
('8 * (*b* + ('16 * *a*))) = (('8 * *b*) + ('128 * *a*))

EVENT: Disable re-associate-times1.

THEOREM: rewrite-os-mbuffer-address
(('8 * (getnth ('2, tm-regs (*os*)) mod '16))
+ ('128 * array-qfirst (os-readyq (*os*))))
= ('8
* ((getnth ('2, tm-regs (*os*)) mod '16)
+ ('16 * array-qfirst (os-readyq (*os*))))))

EVENT: Disable rewrite-os-mbuffer-address.

THEOREM: rewrite-os-mbuffer-address1

```

(( '8 * array-qfirst (os-readyq (os)))
 + ( '128 * (getnth ('2, tm-regs (os)) mod '16)))
= ( '8
  * (array-qfirst (os-readyq (os))
    + ( '16 * (getnth ('2, tm-regs (os)) mod '16))))

```

EVENT: Disable rewrite-os-mbuffer-address1.

THEOREM: os-mbuffers-address-facts1

```

(i < '16)
→ (((('8 * i) < '65536) = '*1>true)
   ∧ (((('128 * i) < '2048) = '*1>true)
      ∧ (((('128 * i) < '65536) = '*1>true)
         ∧ (('16 * i) < '256) = '*1>true))))

```

EVENT: Disable os-mbuffers-address-facts1.

```

(PROVE-LEMMA
 OS-MBUFFERS-ADDRESS-FACTS2
 (REWRITE)
 (IMPLIES
 (AND (LESSP I '16) (LESSP J '16))
 (AND
 (EQUAL (LESSP (PLUS (TIMES '8 J) (TIMES '128 I))
 '65536)
 '*1*TRUE)
 (AND
 (EQUAL (LESSP (PLUS '500
 (PLUS (TIMES '8 J) (TIMES '128 I)))
 '65536)
 '*1*TRUE)
 (AND
 (EQUAL (LESSP (PLUS '499
 (PLUS (TIMES '8 J) (TIMES '128 I)))
 '65536)
 '*1*TRUE)
 (AND
 (EQUAL (LESSP (PLUS '501
 (PLUS (TIMES '8 J) (TIMES '128 I)))
 '65536)
 '*1*TRUE)

```



```

(AND
  (EQUAL (LESSP (PLUS '502
                    (PLUS (TIMES '8 J) (TIMES '128 I)))
                    '65536)
        '*1*TRUE)
  (AND
    (EQUAL (LESSP (PLUS '503
                      (PLUS (TIMES '8 J) (TIMES '128 I)))
                      '65536)
          '*1*TRUE)
    (AND
      (EQUAL (LESSP (PLUS '1
                        (PLUS (TIMES '8 J) (TIMES '128 I)))
                        '2048)
            '*1*TRUE)
      (AND (EQUAL (LESSP (PLUS '2
                            (PLUS (TIMES '8 J) (TIMES '128 I)))
                            '2048)
                '*1*TRUE)
            (AND (EQUAL (LESSP (PLUS '3
                                (PLUS (TIMES '8 J) (TIMES '128 I)))
                                '2048)
                    '*1*TRUE)
                (EQUAL (LESSP (PLUS '4
                                (PLUS (TIMES '8 J) (TIMES '128 I)))
                                '2048)
                    '*1*TRUE))))))))))
((ENABLE COMMUTATIVITY-OF-PLUS)
 (ENABLE-THEORY GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable os-mbuffers-address-facts2.

THEOREM: os-mbuffers-address-facts3

```

((i < '16) ^ (j < '16))
→ (((('16 * j) + i) < '256) = '*1>true)
   ^ (((('8 * (i + ('16 * j))) < '2048) = '*1>true)
      ^ (((('8 * (('16 * j) + i)) < '2048)
         = '*1>true)
        ^ (((('2048
              < ('8
                + ('8
                  * (('16 * j) + i))))
           = '*1>false)

```

$$\begin{aligned}
& \wedge \left(\left(\left(\left('8 * j \right) + \left('128 * i \right) \right) \right. \right. \\
& \quad \left. \left. < '2048 \right) \right. \\
& \quad \left. = '*1*true \right) \\
& \wedge \left(\left(\left('2048 \right. \right. \right. \\
& \quad \left. \left. < \left('8 \right. \right. \right. \\
& \quad \quad \left. + \left('8 \right. \right. \\
& \quad \quad \quad \left. * \left(i + \left('16 \right. \right. \right. \\
& \quad \quad \quad \quad \left. * j \right) \right) \right) \right) \\
& \quad \left. = '*1*false \right) \\
& \wedge \left(\left(\left(i + \left('16 * j \right) \right) \right. \right. \\
& \quad \left. \left. < '256 \right) \right. \\
& \quad \left. = '*1*true \right) \right) \right)
\end{aligned}$$

EVENT: Disable os-mbuffers-address-facts3.

THEOREM: finite-numberp-getnth-from-current-mbuffer
(good-tm(*os*) \wedge ((*sid* < '16) \wedge ((*did* < '16) \wedge (*i* < '8))))
 \rightarrow finite-numberp(getnth(*i*,
getnth(('16 * *sid*) + *did*,
table('8, os-mbuffers(*os*))),
'65536))

THEOREM: finite-numberp-getnth-from-current-mbuffer1
(good-tm(*os*) \wedge ((*sid* < '16) \wedge ((*did* < '16) \wedge (*i* < '8))))
 \rightarrow ((getnth(*i*, getnth(('16 * *sid*) + *did*, table('8, os-mbuffers(*os*))) $\in \mathbf{N}$)
 \wedge (getnth(*i*,
getnth(('16 * *sid*) + *did*,
table('8, os-mbuffers(*os*)))
< '65536))

EVENT: Disable finite-numberp-getnth-from-current-mbuffer1.

THEOREM: access-current-os-mbuffer-fields
(good-os(*os*) \wedge ((*sid* < '16) \wedge (*did* < '16)))
 \rightarrow ((getnth('1,
getnth(('16 * *sid*) + *did*, table('8, os-mbuffers(*os*)))
< '65535)
 \wedge ((getnth('2,
getnth(('16 * *sid*) + *did*,
table('8, os-mbuffers(*os*)))
< '65535)
 \wedge (getnth('3,
getnth(('16 * *sid*) + *did*,
table('8, os-mbuffers(*os*)))

< '65535)))

EVENT: Disable access-current-os-mbuffer-fields.

THEOREM: lessp-plus-fact-for-mbuffers1
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*) ∧ ((*sid* < '16) ∧ (*did* < '16))))
 → (((('4
 + ((('8 * *did*) + ('128 * *sid*))
 + getnth ('1,
 getnth (('16 * *sid*) + *did*,
 table ('8, os-mbuffers (*os*))))))
 < '2048)
 = '*1>true)

EVENT: Disable lessp-plus-fact-for-mbuffers1.

THEOREM: lessp-plus-fact-for-mbuffers2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*) ∧ ((*sid* < '16) ∧ (*did* < '16))))
 → (((('503
 + ((('8 * *did*) + ('128 * *sid*))
 + getnth ('1,
 getnth (('16 * *sid*) + *did*,
 table ('8, os-mbuffers (*os*))))))
 < '65536)
 = '*1>true)

EVENT: Disable lessp-plus-fact-for-mbuffers2.

THEOREM: index-within-length-of-mbuffer
 (if '4 < *i* then '*1>false
 else '*1>true endif
 ∧ finite-number-queuep (getnth (('16 * array-qfirst (os-readyq (*os*)))
 + (getnth ('2, tm-regs (*os*)) mod '16),
 table ('8, os-mbuffers (*os*))),
 '4,
 '65536))
 → (((*i* + getnth ('1,
 getnth (('16 * array-qfirst (os-readyq (*os*)))
 + (getnth ('2, tm-regs (*os*)) mod '16),
 table ('8, os-mbuffers (*os*))))
 < '8)
 = '*1>true)

EVENT: Disable index-within-length-of-mbuffer.

THEOREM: lessp-plus-fact-for-mbuffers-for-receive1
(good-os(*os*)
∧ ((¬ tm-waiting(*os*) ∧ ((*sid* < '16) ∧ (*did* < '16))))
→ (((('4
+ ((('8 * *did*) + ('128 * *sid*))
+ getnth('0,
getnth(*did* + ('16 * *sid*),
table('8, os-mbuffers(*os*))))))
< '2048)
= '*1>true)

EVENT: Disable lessp-plus-fact-for-mbuffers-for-receive1.

THEOREM: lessp-plus-fact-for-mbuffers-for-receive2
(good-os(*os*)
∧ ((¬ tm-waiting(*os*) ∧ ((*sid* < '16) ∧ (*did* < '16))))
→ (((('503
+ ((('8 * *did*) + ('128 * *sid*))
+ getnth('0,
getnth(*did* + ('16 * *sid*),
table('8, os-mbuffers(*os*))))))
< '65536)
= '*1>true)

EVENT: Disable lessp-plus-fact-for-mbuffers-for-receive2.

THEOREM: access-current-os-mbuffer-fields-for-receive
(good-os(*os*) ∧ ((*i* < '16) ∧ (*j* < '16)))
→ ((getnth('0, getnth(*i* + ('16 * *j*), table('8, os-mbuffers(*os*))))
< '65535)
∧ ((getnth('2,
getnth(*i* + ('16 * *j*), table('8, os-mbuffers(*os*))))
< '65535)
∧ (getnth('3,
getnth(*i* + ('16 * *j*),
table('8, os-mbuffers(*os*))))
< '65535)))

EVENT: Disable access-current-os-mbuffer-fields-for-receive.

THEOREM: index-within-length-of-mbuffer-for-receive

```

(if '4 < i then '*1*false
else '*1*true endif
  ∧ finite-number-queuep (getnth (array-qfirst (os-readyq (os))
                                + ('16
                                   * (getnth ('2, tm-regs (os))
                                             mod '16)),
                                table ('8, os-mbuffers (os))),
                                '4,
                                '65536))
→ (((i + getnth ('0,
                getnth (array-qfirst (os-readyq (os))
                        + ('16
                           * (getnth ('2, tm-regs (os))
                                     mod '16)),
                table ('8, os-mbuffers (os))))))
    < '8)
  = '*1*true)

```

EVENT: Disable index-within-length-of-mbuffer-for-receive.

THEOREM: access-current-mbuffer

```

((sid < '16) ∧ ((did < '16) ∧ (i < '8)))
→ (getnth (i + (('8 * did) + ('128 * sid)), os-mbuffers (os))
    = getnth (i, os-current-mbuffer (sid, did, os)))

```

EVENT: Disable access-current-mbuffer.

THEOREM: contract-putnth-into-current-mbuffer

```

(((i + getnth ('1, os-current-mbuffer (sid, did, os))) < '8)
 ∧ ((sid < '16) ∧ (did < '16)))
→ (putnth (value,
           i + (('8 * did) + ('128 * sid))
           + getnth ('1,
                    getnth (('16 * sid) + did,
                            table ('8, os-mbuffers (os))))),
    os-mbuffers (os))
= putseg (putnth (value,
                 i + getnth ('1, os-current-mbuffer (sid, did, os)),
                 os-current-mbuffer (sid, did, os)),
         ('8 * did) + ('128 * sid),
         os-mbuffers (os)))

```

EVENT: Disable contract-putnth-into-current-mbuffer.

THEOREM: associativity-of-plus-crock

$$\begin{aligned}
& ('4 \\
& + ((('8 * i) \\
& \quad + (('128 * j) \\
& \quad \quad + \text{getnth} (('8 * i) + ('128 * j), \text{os-mbuffers}(os)))))) \\
= & ((('8 * i) \\
& + (('128 * j) \\
& \quad + ('4 \\
& \quad \quad + \text{getnth} (('8 * i) + ('128 * j), \\
& \quad \quad \quad \text{os-mbuffers}(os))))))
\end{aligned}$$

EVENT: Disable associativity-of-plus-crock.

THEOREM: contract-getnth-from-current-mbuffer-for-receive

$$\begin{aligned}
& (((('4 + \text{getnth} ('0, \text{os-current-mbuffer}(sid, did, os))) < '8) \\
& \wedge ((sid < '16) \wedge (did < '16))) \\
\rightarrow & (\text{getnth} ('4 \\
& \quad + (((('8 * did) + ('128 * sid)) \\
& \quad \quad + \text{getnth} ('0, \\
& \quad \quad \quad \text{getnth}(did + ('16 * sid), \\
& \quad \quad \quad \quad \text{table}('8, \text{os-mbuffers}(os)))))), \\
& \quad \text{os-mbuffers}(os)) \\
= & \text{getnth} ('4 + \text{getnth} ('0, \text{os-current-mbuffer}(sid, did, os)), \\
& \quad \text{os-current-mbuffer}(sid, did, os))
\end{aligned}$$

EVENT: Disable contract-getnth-from-current-mbuffer-for-receive.

THEOREM: access-current-mbuffer-qhead

$$\begin{aligned}
& ((sid < '16) \wedge (did < '16)) \\
\rightarrow & (\text{getnth} (('8 * did) + ('128 * sid), \text{os-mbuffers}(os)) \\
& = \text{getnth} ('0, \text{os-current-mbuffer}(sid, did, os)))
\end{aligned}$$

EVENT: Disable access-current-mbuffer-qhead.

THEOREM: contract-putnth-into-current-mbuffer-for-receive1

$$\begin{aligned}
& ((i \in \mathbf{N}) \wedge ((i < '4) \wedge ((sid < '16) \wedge (did < '16)))) \\
\rightarrow & (\text{putnth}(value, \\
& \quad i + (('8 * did) + ('128 * sid)), \\
& \quad \text{os-mbuffers}(os)) \\
= & \text{putseg}(\text{putnth}(value, i, \text{os-current-mbuffer}(sid, did, os)), \\
& \quad ('8 * did) + ('128 * sid), \\
& \quad \text{os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-mbuffer-for-receive1.

DEFINITION:

os-mbuffers-with-enqueued-message (*os*)
= putseg (array-enq (getnth ('3, tm-regs (*os*)),
os-current-mbuffer (array-qfirst (os-readyq (*os*)),
getnth ('2, tm-regs (*os*)) **mod** '16,
os)),
('8 * (getnth ('2, tm-regs (*os*)) **mod** '16))
+ ('128 * array-qfirst (os-readyq (*os*))),
os-mbuffers (*os*))

THEOREM: length-os-mbuffers-with-enqueued-message
length (os-mbuffers-with-enqueued-message (*os*)) = '2048

EVENT: Disable length-os-mbuffers-with-enqueued-message.

THEOREM: plistp-os-mbuffers-with-enqueued-message
plistp (os-mbuffers-with-enqueued-message (*os*))

EVENT: Disable plistp-os-mbuffers-with-enqueued-message.

DEFINITION:

os-mbuffers-with-dequeued-message (*os*)
= putseg (array-deq (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)),
('8 * array-qfirst (os-readyq (*os*)))
+ ('128 * (getnth ('2, tm-regs (*os*)) **mod** '16)),
os-mbuffers (*os*))

THEOREM: length-os-mbuffers-with-dequeued-message
length (os-mbuffers-with-dequeued-message (*os*)) = '2048

EVENT: Disable length-os-mbuffers-with-dequeued-message.

THEOREM: plistp-os-mbuffers-with-dequeued-message
plistp (os-mbuffers-with-dequeued-message (*os*))

EVENT: Disable plistp-os-mbuffers-with-dequeued-message.

THEOREM: correctness-of-array-qfullp-mbuffers-instance
(array-queuep (*queue*) \wedge ('4 = getnth ('3, *queue*)))
 \rightarrow (array-qfullp (*queue*) = qfullp (mapup-queue (*queue*), '4))

EVENT: Disable correctness-of-array-qfullp-mbuffers-instance.

THEOREM: mapup-queue-getnth2-os-mbuffers

$$\begin{aligned} & ((i < '16) \wedge (j < '16)) \\ \rightarrow & (\text{mapup-queue}(\text{getnth2}(i, j, \text{table}('16, \text{table}('8, \text{os-mbuffers}(os)))))) \\ & = \text{getnth2}(i, j, \text{mapup-os-mbuffers}(os)) \end{aligned}$$

EVENT: Disable mapup-queue-getnth2-os-mbuffers.

THEOREM: os-current-mbuffer-rewrite-to-getnth2

$$\begin{aligned} & ((sid < '16) \wedge (did < '16)) \\ \rightarrow & (\text{getnth}('16 * sid + did, \text{table}('8, \text{os-mbuffers}(os))) \\ & = \text{getnth2}(sid, did, \text{table}('16, \text{table}('8, \text{os-mbuffers}(os)))) \end{aligned}$$

EVENT: Disable os-current-mbuffer-rewrite-to-getnth2.

THEOREM: limited-associativity-of-plus2

$$(((1 + a) + b) + c) = ((1 + a) + (b + c))$$

EVENT: Disable limited-associativity-of-plus2.

THEOREM: length-finite-number-queue-corollary-for-ibuffers

$$\text{finite-number-queuep}(queue, '4, '65536) \rightarrow (\text{length}(queue) = '8)$$

EVENT: Disable length-finite-number-queue-corollary-for-ibuffers.

THEOREM: plistp-finite-number-queue-corollary-for-ibuffers

$$\text{finite-number-queuep}(queue, '4, '65536) \rightarrow \text{plistp}(queue)$$

EVENT: Disable plistp-finite-number-queue-corollary-for-ibuffers.

THEOREM: array-queuep-finite-number-queue-corollary-for-ibuffers

$$\text{finite-number-queuep}(queue, '4, '65536) \rightarrow \text{array-queuep}(queue)$$

EVENT: Disable array-queuep-finite-number-queue-corollary-for-ibuffers.

THEOREM: finite-number-queue-qhead-size-corollary-for-ibuffers

$$\begin{aligned} & \text{finite-number-queuep}(queue, '4, '65536) \\ \rightarrow & ((\text{getnth}('0, queue) \in \mathbf{N}) \wedge (\text{getnth}('0, queue) < '4)) \end{aligned}$$

EVENT: Disable finite-number-queue-qhead-size-corollary-for-ibuffers.

THEOREM: finite-number-queue-qtail-size-corollary-for-ibuffers

$$\begin{aligned} & \text{finite-number-queuep}(queue, '4, '65536) \\ \rightarrow & ((\text{getnth}('1, queue) \in \mathbf{N}) \wedge (\text{getnth}('1, queue) < '4)) \end{aligned}$$

EVENT: Disable finite-number-queue-qtail-size-corollary-for-ibuffers.

THEOREM: finite-number-queue-qcurrenlength-size-corollary-for-ibuffers
finite-number-queuep(*queue*, '4, '65536)
→ ((getnth('2, *queue*) ∈ **N**) ∧ (getnth('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrenlength-size-corollary-for-ibuffers.

THEOREM: finite-number-queue-qmaxlength-value-corollary-for-ibuffers
finite-number-queuep(*queue*, '4, '65536) → (getnth('3, *queue*) = '4)

EVENT: Disable finite-number-queue-qmaxlength-value-corollary-for-ibuffers.

THEOREM: finite-number-queuep-array-enq-finite-number-queue-corollary-for-ibuffers
(finite-number-queuep(*queue*, '4, '65536)
∧ ((¬ array-qfullp(*queue*)) ∧ ((*item* ∈ **N**) ∧ (*item* < '65536))))
→ finite-number-queuep(array-enq(*item*, *queue*), '4, '65536)

EVENT: Disable finite-number-queuep-array-enq-finite-number-queue-corollary-for-ibuffers.

THEOREM: finite-number-queuep-array-deq-finite-number-queue-corollary-for-ibuffers
(finite-number-queuep(*queue*, '4, '65536) ∧ (¬ array-qemptytyp(*queue*)))
→ finite-number-queuep(array-deq(*queue*), '4, '65536)

EVENT: Disable finite-number-queuep-array-deq-finite-number-queue-corollary-for-ibuffers.

THEOREM: array-qfirst-finite-number-queue-corollary-for-ibuffers
(finite-number-queuep(*queue*, '4, '65536) ∧ (¬ array-qemptytyp(*queue*)))
→ ((array-qfirst(*queue*) ∈ **N**) ∧ (array-qfirst(*queue*) < '65536))

EVENT: Disable array-qfirst-finite-number-queue-corollary-for-ibuffers.

THEOREM: remainder-array-qfirst-finite-number-queue-corollary-for-ibuffers
(finite-number-queuep(*queue*, '4, '65536) ∧ (¬ array-qemptytyp(*queue*)))
→ ((array-qfirst(*queue*) **mod** '65536) = array-qfirst(*queue*))

EVENT: Disable remainder-array-qfirst-finite-number-queue-corollary-for-ibuffers.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-ibuffers* to consist of the following events: remainder-array-qfirst-finite-number-queue-

corollary-for-ibuffers, array-qfirst-finite-number-queue-corollary-for-ibuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-ibuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-ibuffers, finite-number-queue-qmaxlength-value-corollary-for-ibuffers, finite-number-queue-qcurrlength-size-corollary-for-ibuffers, finite-number-queue-qtail-size-corollary-for-ibuffers, finite-number-queue-qhead-size-corollary-for-ibuffers, array-queuep-finite-number-queue-corollary-for-ibuffers, plistp-finite-number-queue-corollary-for-ibuffers, length-finite-number-queue-corollary-for-ibuffers.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-ibuffers1* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary-for-ibuffers, array-qfirst-finite-number-queue-corollary-for-ibuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-ibuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-ibuffers, finite-number-queue-qcurrlength-size-corollary-for-ibuffers, finite-number-queue-qtail-size-corollary-for-ibuffers, finite-number-queue-qhead-size-corollary-for-ibuffers, array-queuep-finite-number-queue-corollary-for-ibuffers, plistp-finite-number-queue-corollary-for-ibuffers, length-finite-number-queue-corollary-for-ibuffers.

DEFINITION: OS-IBUFFERS-ADDRESS-PLUS-QTAIL-FIELD = '244

DEFINITION: OS-IBUFFERS-ADDRESS-PLUS-QCURRENLENGTH-FIELD = '245

DEFINITION: OS-IBUFFERS-ADDRESS-PLUS-QMAXLENGTH-FIELD = '246

DEFINITION: OS-IBUFFERS-ADDRESS-PLUS-QARRAY-FIELD = '247

DEFINITION:

os-current-ibuffer (*id*, *os*) = getnth (*id*, table ('8, os-ibuffers (*os*)))

THEOREM: finite-number-queuep-current-ibuffer

(good-os (*os*) \wedge (*id* < '16))

\rightarrow finite-number-queuep (getnth (*id*, table ('8, os-ibuffers (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queuep-current-ibuffer.

THEOREM: finite-number-queuep-array-qreplace-corollary-for-ibuffers

(finite-number-queuep (*queue*, '4, '65536)

\wedge ((\neg array-qempty (*queue*) \wedge finite-numberp (*item*, '65536)))

\rightarrow finite-number-queuep (array-qreplace (*item*, *queue*), '4, '65536)

EVENT: Disable finite-number-queuep-array-qreplace-corollary-for-ibuffers.

```

(PROVE-LEMMA
OS-IBUFFERS-ADDRESS-FACTS
(REWRITE)
(IMPLIES
(LESSP I '16)
(AND
(EQUAL (LESSP (TIMES '8 I) '128)
'*1*TRUE)
(AND
(EQUAL (LESSP '128 (PLUS '8 (TIMES '8 I)))
'*1*FALSE)
(AND
(EQUAL (LESSP (TIMES '8 I) '65536)
'*1*TRUE)
(AND
(EQUAL (LESSP (PLUS '243 (TIMES '8 I))
'65536)
'*1*TRUE)
(AND
(EQUAL (LESSP (PLUS '244 (TIMES '8 I))
'65536)
'*1*TRUE)
(AND
(EQUAL (LESSP (PLUS '245 (TIMES '8 I))
'65536)
'*1*TRUE)
(AND (EQUAL (LESSP (PLUS '246 (TIMES '8 I))
'65536)
'*1*TRUE)
(AND (EQUAL (LESSP (PLUS '247 (TIMES '8 I))
'65536)
'*1*TRUE)
(AND (EQUAL (LESSP (PLUS '1 (TIMES '8 I)) '128)
'*1*TRUE)
(AND (EQUAL (LESSP (PLUS '2 (TIMES '8 I)) '128)
'*1*TRUE)
(AND (EQUAL (LESSP (PLUS '3 (TIMES '8 I)) '128)
'*1*TRUE)
(EQUAL (LESSP (PLUS '4 (TIMES '8 I)) '128)
'*1*TRUE)))))))))))))
((ENABLE-THEORY GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable os-ibuffers-address-facts.

THEOREM: lessp-plus-facts-for-ibuffers
(good-os(*os*) \wedge (\neg tm-waiting(*os*) \wedge (*id* < '16)))
 \rightarrow (((('4
+ (('8 * *id*)
+ getnth('0, getnth(*id*, table('8, os-ibuffers(*os*))))))
< '128)
= '*1*true)
 \wedge (((('247
+ (('8 * *id*)
+ getnth('0,
getnth(*id*, table('8, os-ibuffers(*os*))))))
< '65536)
= '*1*true))

EVENT: Disable lessp-plus-facts-for-ibuffers.

THEOREM: index-within-length-of-ibuffer
(if '4 < *i* then '*1*false
else '*1*true endif
 \wedge finite-number-queuep(getnth(*id*, table('8, os-ibuffers(*os*)),
'4,
'65536))
 \rightarrow (((*i* + getnth('0, getnth(*id*, table('8, os-ibuffers(*os*)))) < '8)
= '*1*true)

EVENT: Disable index-within-length-of-ibuffer.

THEOREM: lessp-qmaxlength-field-os-current-ibuffer-tm-wordlub
(good-tm(*os*) \wedge (*id* < '16))
 \rightarrow (getnth('3, getnth(*id*, table('8, os-ibuffers(*os*)))) < '65536)

EVENT: Disable lessp-qmaxlength-field-os-current-ibuffer-tm-wordlub.

THEOREM: qreplace-address-fact
(good-os(*os*) \wedge tm-some-input-interruptp(tm-iports(*os*)))
 \rightarrow (((('243
+ (('8 * tm-interrupting-input-port(tm-iports(*os*)))
+ decr-mod(getnth('1,
getnth(tm-interrupting-input-port(tm-iports(*os*)),
table('8, os-ibuffers(*os*))))),

$$\begin{aligned}
& \text{'4})) \\
& < \text{'65536}) \\
& = \text{'*1*true}) \\
& \wedge \text{((((('247} \\
& \quad + \text{ (('8} \\
& \quad \quad * \text{tm-interrupting-input-port (tm-iports (os))}) \\
& \quad \quad + \text{decr-mod (getnth ('1,} \\
& \quad \quad \quad \text{getnth (tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \quad \quad \text{table ('8,} \\
& \quad \quad \quad \text{os-ibuffers (os))}),} \\
& \quad \quad \quad \text{'4})) \\
& < \text{'65536}) \\
& = \text{'*1*true}) \\
& \wedge \text{((((('4} \\
& \quad + \text{ (('8} \\
& \quad \quad * \text{tm-interrupting-input-port (tm-iports (os))}) \\
& \quad \quad + \text{decr-mod (getnth ('1,} \\
& \quad \quad \quad \text{getnth (tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \quad \quad \text{table ('8,} \\
& \quad \quad \quad \text{os-ibuffers (os))}),} \\
& \quad \quad \quad \text{'4})) \\
& < \text{'128}) \\
& = \text{'*1*true}))
\end{aligned}$$

EVENT: Disable qreplace-address-fact.

THEOREM: more-ibuffer-address-facts

$$\begin{aligned}
& (\text{good-os (os) } \wedge \text{tm-some-input-interruptp (tm-iports (os))}) \\
& \rightarrow \text{((((('243} \\
& \quad + \text{ (('8 * tm-interrupting-input-port (tm-iports (os))}) \\
& \quad \quad + \text{getnth ('1,} \\
& \quad \quad \quad \text{getnth (tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \quad \quad \text{table ('8, os-ibuffers (os))}))} \\
& < \text{'65536}) \\
& = \text{'*1*true}) \\
& \wedge \text{((((('247} \\
& \quad + \text{ (('8} \\
& \quad \quad * \text{tm-interrupting-input-port (tm-iports (os))}) \\
& \quad \quad + \text{getnth ('1,} \\
& \quad \quad \quad \text{getnth (tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \quad \quad \text{table ('8, os-ibuffers (os))}))} \\
& < \text{'65536}) \\
& = \text{'*1*true}) \\
& \wedge \text{((((('4}
\end{aligned}$$

$$\begin{aligned}
& + ((\text{'8} \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os))) \\
& \quad + \text{getnth}(\text{'1}, \\
& \quad \quad \text{getnth}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{table}(\text{'8}, \\
& \quad \quad \quad \quad \text{os-ibuffers}(os)))))) \\
& < \text{'128}) \\
& = \text{'*1*true}))
\end{aligned}$$

EVENT: Disable more-ibuffer-address-facts.

THEOREM: access-current-ibuffer

$$\begin{aligned}
& ((id < \text{'16}) \wedge (i < \text{'8})) \\
& \rightarrow (\text{getnth}(i + (\text{'8} * id), \text{os-ibuffers}(os)) \\
& \quad = \text{getnth}(i, \text{os-current-ibuffer}(id, os)))
\end{aligned}$$

EVENT: Disable access-current-ibuffer.

THEOREM: contract-getnth-from-current-ibuffer

$$\begin{aligned}
& (((\text{'4} + \text{getnth}(\text{'0}, \text{os-current-ibuffer}(id, os))) < \text{'8}) \wedge (id < \text{'16})) \\
& \rightarrow (\text{getnth}(\text{'4} \\
& \quad + ((\text{'8} * id) \\
& \quad + \text{getnth}(\text{'0}, \\
& \quad \quad \text{getnth}(id, \text{table}(\text{'8}, \text{os-ibuffers}(os))))), \\
& \quad \text{os-ibuffers}(os)) \\
& = \text{getnth}(\text{'4} + \text{getnth}(\text{'0}, \text{os-current-ibuffer}(id, os)), \\
& \quad \text{os-current-ibuffer}(id, os))
\end{aligned}$$

EVENT: Disable contract-getnth-from-current-ibuffer.

THEOREM: access-current-ibuffer-qhead

$$\begin{aligned}
& (id < \text{'16}) \\
& \rightarrow (\text{getnth}(\text{'8} * id, \text{os-ibuffers}(os)) \\
& \quad = \text{getnth}(\text{'0}, \text{os-current-ibuffer}(id, os)))
\end{aligned}$$

EVENT: Disable access-current-ibuffer-qhead.

THEOREM: contract-putnth-into-current-ibuffer

$$\begin{aligned}
& ((i \in \mathbf{N}) \wedge ((i < \text{'4}) \wedge (id < \text{'16}))) \\
& \rightarrow (\text{putnth}(value, i + (\text{'8} * id), \text{os-ibuffers}(os)) \\
& \quad = \text{putseg}(\text{putnth}(value, i, \text{os-current-ibuffer}(id, os)), \\
& \quad \quad \text{'8} * id, \\
& \quad \quad \text{os-ibuffers}(os)))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-ibuffer.

THEOREM: contract-putnth-into-current-ibuffer1

$$\begin{aligned}
& (((i + \text{decr-mod}(\text{getnth}('1, \text{os-current-ibuffer}(id, os)), '4)) < '8) \\
& \wedge (id < '16)) \\
\rightarrow & (\text{putnth}(value, \\
& \quad i + (('8 * id) \\
& \quad + \text{decr-mod}(\text{getnth}('1, \\
& \quad \quad \text{getnth}(id, \\
& \quad \quad \text{table}('8, \\
& \quad \quad \text{os-ibuffers}(os))), \\
& \quad \quad '4)), \\
& \quad \text{os-ibuffers}(os)) \\
= & \text{putseg}(\text{putnth}(value, \\
& \quad i + \text{decr-mod}(\text{getnth}('1, \\
& \quad \quad \text{os-current-ibuffer}(id, os)), \\
& \quad \quad '4), \\
& \quad \text{os-current-ibuffer}(id, os)), \\
& \quad '8 * id, \\
& \quad \text{os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-ibuffer1.

THEOREM: contract-putnth-into-current-ibuffer2

$$\begin{aligned}
& (((i + \text{getnth}('1, \text{os-current-ibuffer}(id, os))) < '8) \wedge (id < '16)) \\
\rightarrow & (\text{putnth}(value, \\
& \quad i + (('8 * id) \\
& \quad + \text{getnth}('1, \\
& \quad \quad \text{getnth}(id, \text{table}('8, \text{os-ibuffers}(os))))), \\
& \quad \text{os-ibuffers}(os)) \\
= & \text{putseg}(\text{putnth}(value, \\
& \quad i + \text{getnth}('1, \text{os-current-ibuffer}(id, os)), \\
& \quad \text{os-current-ibuffer}(id, os)), \\
& \quad '8 * id, \\
& \quad \text{os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-ibuffer2.

DEFINITION:

$$\begin{aligned}
& \text{os-ibuffers-with-dequeued-character}(os) \\
= & \text{putseg}(\text{array-deq}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), os)), \\
& \quad '8 * \text{os-current-taskid}(os), \\
& \quad \text{os-ibuffers}(os))
\end{aligned}$$

THEOREM: length-os-ibuffers-with-dequeued-character
length (os-ibuffers-with-dequeued-character (*os*)) = '128

EVENT: Disable length-os-ibuffers-with-dequeued-character.

THEOREM: plistp-os-ibuffers-with-dequeued-character
plistp (os-ibuffers-with-dequeued-character (*os*))

EVENT: Disable plistp-os-ibuffers-with-dequeued-character.

DEFINITION:

os-ibuffers-with-qreplaced-overflow-character (*os*)
= putseg (array-qreplace (tm-overflow-char (tm-ichar (getnth (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*)))),
getnth (tm-interrupting-input-port (tm-iports (*os*)),
table ('8, os-ibuffers (*os*))),
'8 * tm-interrupting-input-port (tm-iports (*os*)),
os-ibuffers (*os*))

THEOREM: length-os-ibuffers-with-qreplaced-overflow-character
length (os-ibuffers-with-qreplaced-overflow-character (*os*)) = '128

EVENT: Disable length-os-ibuffers-with-qreplaced-overflow-character.

THEOREM: plistp-os-ibuffers-with-qreplaced-overflow-character
plistp (os-ibuffers-with-qreplaced-overflow-character (*os*))

EVENT: Disable plistp-os-ibuffers-with-qreplaced-overflow-character.

DEFINITION:

os-ibuffers-with-enqueued-overflow-character (*os*)
= putseg (array-enq (tm-overflow-char (tm-ichar (getnth (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))),
getnth (tm-interrupting-input-port (tm-iports (*os*)),
table ('8, os-ibuffers (*os*))),
'8 * tm-interrupting-input-port (tm-iports (*os*)),
os-ibuffers (*os*))

THEOREM: length-os-ibuffers-with-enqueued-overflow-character
length (os-ibuffers-with-enqueued-overflow-character (*os*)) = '128

EVENT: Disable length-os-ibuffers-with-enqueued-overflow-character.

THEOREM: plistp-os-ibuffers-with-enqueued-overflow-character
plistp (os-ibuffers-with-enqueued-overflow-character (*os*))

EVENT: Disable plistp-os-ibuffers-with-enqueued-overflow-character.

DEFINITION:

os-ibuffers-with-enqueued-character (*os*)
= putseg (array-enq (tm-ichar (getnth (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))),
getnth (tm-interrupting-input-port (tm-iports (*os*)),
table ('8, os-ibuffers (*os*))),
'8 * tm-interrupting-input-port (tm-iports (*os*)),
os-ibuffers (*os*))

THEOREM: length-os-ibuffers-with-enqueued-character
length (os-ibuffers-with-enqueued-character (*os*)) = '128

EVENT: Disable length-os-ibuffers-with-enqueued-character.

THEOREM: plistp-os-ibuffers-with-enqueued-character
plistp (os-ibuffers-with-enqueued-character (*os*))

EVENT: Disable plistp-os-ibuffers-with-enqueued-character.

THEOREM: mapup-queue-getnth-for-ibuffers
(good-os (*os*) \wedge (*id* < '16))
 \rightarrow (mapup-queue (getnth (*id*, table ('8, os-ibuffers (*os*))))
= getnth (*id*, mapup-os-ibuffers (*os*)))

EVENT: Disable mapup-queue-getnth-for-ibuffers.

THEOREM: length-finite-number-queue-corollary-for-obuffers
finite-number-queuep (*queue*, '4, '65536) \rightarrow (length (*queue*) = '8)

EVENT: Disable length-finite-number-queue-corollary-for-obuffers.

THEOREM: plistp-finite-number-queue-corollary-for-obuffers
finite-number-queuep (*queue*, '4, '65536) \rightarrow plistp (*queue*)

EVENT: Disable plistp-finite-number-queue-corollary-for-obuffers.

THEOREM: array-queuep-finite-number-queue-corollary-for-obuffers
finite-number-queuep (*queue*, '4, '65536) \rightarrow array-queuep (*queue*)

EVENT: Disable array-queueep-finite-number-queue-corollary-for-obuffers.

THEOREM: finite-number-queue-qhead-size-corollary-for-obuffers
finite-number-queueep(*queue*, '4, '65536)
→ ((getnth('0, *queue*) ∈ **N**) ∧ (getnth('0, *queue*) < '4))

EVENT: Disable finite-number-queue-qhead-size-corollary-for-obuffers.

THEOREM: finite-number-queue-qtail-size-corollary-for-obuffers
finite-number-queueep(*queue*, '4, '65536)
→ ((getnth('1, *queue*) ∈ **N**) ∧ (getnth('1, *queue*) < '4))

EVENT: Disable finite-number-queue-qtail-size-corollary-for-obuffers.

THEOREM: finite-number-queue-qcurrenlength-size-corollary-for-obuffers
finite-number-queueep(*queue*, '4, '65536)
→ ((getnth('2, *queue*) ∈ **N**) ∧ (getnth('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrenlength-size-corollary-for-obuffers.

THEOREM: finite-number-queue-qmaxlength-value-corollary-for-obuffers
finite-number-queueep(*queue*, '4, '65536) → (getnth('3, *queue*) = '4)

EVENT: Disable finite-number-queue-qmaxlength-value-corollary-for-obuffers.

THEOREM: finite-number-queueep-array-enq-finite-number-queue-corollary-for-obuffers
(finite-number-queueep(*queue*, '4, '65536)
∧ ((¬ array-qfullp(*queue*)) ∧ ((*item* ∈ **N**) ∧ (*item* < '65536))))
→ finite-number-queueep(array-enq(*item*, *queue*), '4, '65536)

EVENT: Disable finite-number-queueep-array-enq-finite-number-queue-corollary-for-obuffers.

THEOREM: finite-number-queueep-array-deq-finite-number-queue-corollary-for-obuffers
(finite-number-queueep(*queue*, '4, '65536) ∧ (¬ array-qempty(*queue*)))
→ finite-number-queueep(array-deq(*queue*), '4, '65536)

EVENT: Disable finite-number-queueep-array-deq-finite-number-queue-corollary-for-obuffers.

THEOREM: array-qfirst-finite-number-queue-corollary-for-obuffers
(finite-number-queueep(*queue*, '4, '65536) ∧ (¬ array-qempty(*queue*)))
→ ((array-qfirst(*queue*) ∈ **N**) ∧ (array-qfirst(*queue*) < '65536))

EVENT: Disable array-qfirst-finite-number-queue-corollary-for-obuffers.

THEOREM: remainder-array-qfirst-finite-number-queue-corollary-for-obuffers
(finite-number-queuep (*queue*, '4, '65536) \wedge (\neg array-qempty (*queue*)))
 \rightarrow ((array-qfirst (*queue*) **mod** '65536) = array-qfirst (*queue*))

EVENT: Disable remainder-array-qfirst-finite-number-queue-corollary-for-obuffers.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-obuffers* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary-for-obuffers, array-qfirst-finite-number-queue-corollary-for-obuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-obuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-obuffers, finite-number-queue-qmaxlength-value-corollary-for-obuffers, finite-number-queue-qcurrlength-size-corollary-for-obuffers, finite-number-queue-qtail-size-corollary-for-obuffers, finite-number-queue-qhead-size-corollary-for-obuffers, array-queuep-finite-number-queue-corollary-for-obuffers, plistp-finite-number-queue-corollary-for-obuffers, length-finite-number-queue-corollary-for-obuffers.

EVENT: Let us define the theory *finite-number-queue-corollaries-for-obuffers1* to consist of the following events: remainder-array-qfirst-finite-number-queue-corollary-for-obuffers, array-qfirst-finite-number-queue-corollary-for-obuffers, finite-number-queuep-array-deq-finite-number-queue-corollary-for-obuffers, finite-number-queuep-array-enq-finite-number-queue-corollary-for-obuffers, finite-number-queue-qcurrlength-size-corollary-for-obuffers, finite-number-queue-qtail-size-corollary-for-obuffers, finite-number-queue-qhead-size-corollary-for-obuffers, array-queuep-finite-number-queue-corollary-for-obuffers, plistp-finite-number-queue-corollary-for-obuffers, length-finite-number-queue-corollary-for-obuffers.

DEFINITION: OS-OBUFFERS-ADDRESS-PLUS-QTAIL-FIELD = '372

DEFINITION: OS-OBUFFERS-ADDRESS-PLUS-QCURRENLENGTH-FIELD = '373

DEFINITION: OS-OBUFFERS-ADDRESS-PLUS-QMAXLENGTH-FIELD = '374

DEFINITION: OS-OBUFFERS-ADDRESS-PLUS-QARRAY-FIELD = '375

DEFINITION:

os-current-obuffer (*id*, *os*) = getnth (*id*, table ('8, os-obuffers (*os*)))

THEOREM: finite-number-queuep-current-obuffer

(good-os (*os*) \wedge (*id* < '16))

\rightarrow finite-number-queuep (getnth (*id*, table ('8, os-obuffers (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queuep-current-obuffer.

```
(PROVE-LEMMA
OS-OBUFFERS-ADDRESS-FACTS
(REWRITE)
(IMPLIES
  (LESSP I '16)
  (AND
    (EQUAL (LESSP (TIMES '8 I) '65536)
            '*1*TRUE)
    (AND
      (EQUAL (LESSP (PLUS '371 (TIMES '8 I))
                  '65536)
              '*1*TRUE)
      (AND
        (EQUAL (LESSP (PLUS '372 (TIMES '8 I))
                  '65536)
                '*1*TRUE)
        (AND
          (EQUAL (LESSP (PLUS '373 (TIMES '8 I))
                  '65536)
                  '*1*TRUE)
          (AND (EQUAL (LESSP (PLUS '374 (TIMES '8 I))
                          '65536)
                      '*1*TRUE)
                (AND (EQUAL (LESSP (PLUS '375 (TIMES '8 I))
                              '65536)
                          '*1*TRUE)
                    (AND (EQUAL (LESSP (PLUS '1 (TIMES '8 I)) '128)
                                '*1*TRUE)
                        (AND (EQUAL (LESSP (PLUS '2 (TIMES '8 I)) '128)
                                    '*1*TRUE)
                            (AND (EQUAL (LESSP (PLUS '3 (TIMES '8 I)) '128)
                                        '*1*TRUE)
                                (EQUAL (LESSP (PLUS '4 (TIMES '8 I)) '128)
                                        '*1*TRUE))))))))))))))
  ((ENABLE-THEORY GROUND-ZERO)
   (DISABLE-THEORY T)))
```

EVENT: Disable os-obuffers-address-facts.

THEOREM: lessp-plus-facts-for-obuffers
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (id < '16)))$
 $\rightarrow (((('4$
 $\quad + (('8 * id$
 $\quad \quad + \text{getnth}('1, \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os))))))$
 $< '128)$
 $= '*1*true)$
 $\wedge (((('375$
 $\quad + (('8 * id$
 $\quad \quad + \text{getnth}('1,$
 $\quad \quad \quad \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os))))))$
 $< '65536)$
 $= '*1*true))$

EVENT: Disable lessp-plus-facts-for-obuffers.

THEOREM: lessp-qmaxlength-field-os-current-obuffer-tm-wordlub
 $(\text{good-tm}(os) \wedge (id < '16))$
 $\rightarrow (\text{getnth}('3, \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os)))) < '65536)$

EVENT: Disable lessp-qmaxlength-field-os-current-obuffer-tm-wordlub.

THEOREM: obuffer-address-facts
 $(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow (((('375$
 $\quad + (('8 * \text{tm-interrupting-output-port}(\text{tm-oports}(os)))$
 $\quad \quad + \text{getnth}('0,$
 $\quad \quad \quad \text{getnth}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \quad \quad \text{table}('8, \text{os-obuffers}(os))))))$
 $< '65536)$
 $= '*1*true)$

EVENT: Disable obuffer-address-facts.

THEOREM: access-current-obuffer
 $((id < '16) \wedge (i < '8))$
 $\rightarrow (\text{getnth}(i + ('8 * id), \text{os-obuffers}(os))$
 $\quad = \text{getnth}(i, \text{os-current-obuffer}(id, os)))$

EVENT: Disable access-current-obuffer.

THEOREM: contract-getnth-from-current-obuffer
 $((('4 + \text{getnth}('1, \text{os-current-obuffer}(id, os))) < '8) \wedge (id < '16))$

$$\begin{aligned}
& \rightarrow (\text{getnth}('4 \\
& \quad + (('8 * id \\
& \quad \quad + \text{getnth}('1, \\
& \quad \quad \quad \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os))))), \\
& \quad \text{os-obuffers}(os)) \\
& = \text{getnth}('4 + \text{getnth}('1, \text{os-current-obuffer}(id, os)), \\
& \quad \text{os-current-obuffer}(id, os))
\end{aligned}$$

EVENT: Disable contract-getnth-from-current-obuffer.

THEOREM: contract-putnth-into-current-obuffer

$$\begin{aligned}
& (((i + \text{getnth}('1, \text{os-current-obuffer}(id, os))) < '8) \wedge (id < '16)) \\
& \rightarrow (\text{putnth}(value, \\
& \quad i + (('8 * id \\
& \quad \quad + \text{getnth}('1, \\
& \quad \quad \quad \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os))))), \\
& \quad \text{os-obuffers}(os)) \\
& = \text{putseg}(\text{putnth}(value, \\
& \quad \quad i + \text{getnth}('1, \text{os-current-obuffer}(id, os)), \\
& \quad \quad \text{os-current-obuffer}(id, os)), \\
& \quad '8 * id, \\
& \quad \text{os-obuffers}(os))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-obuffer.

THEOREM: contract-putnth-into-current-obuffer1

$$\begin{aligned}
& ((i \in \mathbf{N}) \wedge ((i < '4) \wedge (id < '16))) \\
& \rightarrow (\text{putnth}(value, i + '8 * id, \text{os-obuffers}(os)) \\
& \quad = \text{putseg}(\text{putnth}(value, i, \text{os-current-obuffer}(id, os)), \\
& \quad \quad '8 * id, \\
& \quad \quad \text{os-obuffers}(os))
\end{aligned}$$

EVENT: Disable contract-putnth-into-current-obuffer1.

THEOREM: contract-getnth-from-current-obuffer1

$$\begin{aligned}
& ((('4 + \text{getnth}('0, \text{os-current-obuffer}(id, os))) < '8) \wedge (id < '16)) \\
& \rightarrow (\text{getnth}('4 \\
& \quad + (('8 * id \\
& \quad \quad + \text{getnth}('0, \\
& \quad \quad \quad \text{getnth}(id, \text{table}('8, \text{os-obuffers}(os))))), \\
& \quad \text{os-obuffers}(os)) \\
& = \text{getnth}('4 + \text{getnth}('0, \text{os-current-obuffer}(id, os)), \\
& \quad \text{os-current-obuffer}(id, os))
\end{aligned}$$

EVENT: Disable contract-getnth-from-current-obuffer1.

THEOREM: access-current-obuffer-qhead
($id < '16$)
 \rightarrow (getnth ('8 * id , os-obuffers (os))
= getnth ('0, os-current-obuffer (id , os)))

EVENT: Disable access-current-obuffer-qhead.

DEFINITION:
os-obuffers-with-enqueued-character (os)
= putseg (array-enq (getnth ('3, tm-regs (os)),
os-current-obuffer (os-current-taskid (os), os)),
'8 * os-current-taskid (os),
os-obuffers (os))

THEOREM: length-os-obuffers-with-enqueued-character
length (os-obuffers-with-enqueued-character (os)) = '128

EVENT: Disable length-os-obuffers-with-enqueued-character.

THEOREM: plistp-os-obuffers-with-enqueued-character
plistp (os-obuffers-with-enqueued-character (os))

EVENT: Disable plistp-os-obuffers-with-enqueued-character.

DEFINITION:
os-obuffers-with-dequeued-character (os)
= putseg (array-deq (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
 os)),
'8 * tm-interrupting-output-port (tm-oports (os)),
os-obuffers (os))

THEOREM: length-os-obuffers-with-dequeued-character
length (os-obuffers-with-dequeued-character (os)) = '128

EVENT: Disable length-os-obuffers-with-dequeued-character.

THEOREM: plistp-os-obuffers-with-dequeued-character
plistp (os-obuffers-with-dequeued-character (os))

EVENT: Disable plistp-os-obuffers-with-dequeued-character.

THEOREM: restore-regs
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 \rightarrow (getseg('0,
'8,
getseg('9 * array-qfirst(os-readyq(os)),
'9,
putseg(tm-cpu(os),
'9 * array-qfirst(os-readyq(os)),
os-task-table(os))))
= tm-regs(os))

EVENT: Disable restore-regs.

THEOREM: restore-regs1
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 \rightarrow (getseg('0,
'8,
getnth(array-qfirst(os-readyq(os)),
table('9,
putseg(tm-cpu(os),
'9 * array-qfirst(os-readyq(os)),
os-task-table(os))))))
= tm-regs(os))

EVENT: Disable restore-regs1.

THEOREM: restore-cc
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 \rightarrow (((getnth('8 + ('9 * array-qfirst(os-readyq(os))),
putseg(tm-cpu(os),
'9 * array-qfirst(os-readyq(os)),
os-task-table(os)))
mod '4)
= tm-cc(os))

EVENT: Disable restore-cc.

THEOREM: restore-error
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 \rightarrow (((getnth('8 + ('9 * array-qfirst(os-readyq(os))),
putseg(tm-cpu(os),
'9 * array-qfirst(os-readyq(os)),
os-task-table(os)))

$$\begin{aligned}
& \mathbf{mod} \ '256) \\
& \div \ '4) \\
& = \ \text{tm-error} (os)
\end{aligned}$$

EVENT: Disable restore-error.

$$\begin{aligned}
& \text{THEOREM: restore-svcflag} \\
& (\text{good-os} (os) \wedge (\neg \text{tm-waiting} (os))) \\
& \rightarrow \ ((\text{getnth} ('8 + ('9 * \text{array-qfirst} (\text{os-readyq} (os))), \\
& \qquad \qquad \text{putseg} (\text{tm-cpu} (os), \\
& \qquad \qquad \qquad '9 * \text{array-qfirst} (\text{os-readyq} (os)), \\
& \qquad \qquad \qquad \text{os-task-table} (os))) \\
& \qquad \mathbf{mod} \ '512) \\
& \qquad \div \ '256) \\
& \qquad = \ \text{tm-svcflag} (os)
\end{aligned}$$

EVENT: Disable restore-svcflag.

$$\begin{aligned}
& \text{THEOREM: restore-svcid} \\
& (\text{good-os} (os) \wedge (\neg \text{tm-waiting} (os))) \\
& \rightarrow \ ((\text{getnth} ('8 + ('9 * \text{array-qfirst} (\text{os-readyq} (os))), \\
& \qquad \qquad \text{putseg} (\text{tm-cpu} (os), \\
& \qquad \qquad \qquad '9 * \text{array-qfirst} (\text{os-readyq} (os)), \\
& \qquad \qquad \qquad \text{os-task-table} (os))) \\
& \qquad \div \ '512) \\
& \qquad = \ \text{tm-svcid} (os)
\end{aligned}$$

EVENT: Disable restore-svcid.

$$\begin{aligned}
& \text{DEFINITION:} \\
& \text{os-task-table-with-delivered-character} (os) \\
& = \ \text{putseg} (\text{putnth} (\text{array-qfirst} (\text{getnth} (\text{os-current-taskid} (os), \\
& \qquad \qquad \qquad \qquad \qquad \qquad \text{table} ('8, \text{os-ibuffers} (os))), \\
& \qquad \qquad \qquad \qquad \qquad \qquad '3, \\
& \qquad \qquad \qquad \qquad \qquad \qquad \text{tm-cpu} (os)), \\
& \qquad \qquad \qquad '9 * \text{array-qfirst} (\text{os-readyq} (os)), \\
& \qquad \text{os-task-table} (os))
\end{aligned}$$

$$\begin{aligned}
& \text{THEOREM: length-os-task-table-with-delivered-character} \\
& \text{length} (\text{os-task-table-with-delivered-character} (os)) = '144
\end{aligned}$$

EVENT: Disable length-os-task-table-with-delivered-character.

THEOREM: plistp-os-task-table-with-delivered-character
 plistp (os-task-table-with-delivered-character (*os*))

EVENT: Disable plistp-os-task-table-with-delivered-character.

THEOREM: restore-regs-for-tyi-svc-return
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (getseg ('0,
 '8,
 getnth (array-qfirst (os-readyq (*os*)),
 table ('9, os-task-table-with-delivered-character (*os*))))
 = putnth (array-qfirst (getnth (os-current-taskid (*os*),
 table ('8, os-ibuffers (*os*))),
 '3,
 tm-regs (*os*)))

EVENT: Disable restore-regs-for-tyi-svc-return.

THEOREM: restore-cc-for-tyi-svc-return
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow ((getnth ('8 + ('9 * array-qfirst (os-readyq (*os*))),
 os-task-table-with-delivered-character (*os*))
 mod '4)
 = tm-cc (*os*))

EVENT: Disable restore-cc-for-tyi-svc-return.

THEOREM: restore-error-for-tyi-svc-return
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (((getnth ('8 + ('9 * array-qfirst (os-readyq (*os*))),
 os-task-table-with-delivered-character (*os*))
 mod '256)
 \div '4)
 = tm-error (*os*))

EVENT: Disable restore-error-for-tyi-svc-return.

THEOREM: restore-svcid-for-tyi-svc-return
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow ((getnth ('8 + ('9 * array-qfirst (os-readyq (*os*))),
 os-task-table-with-delivered-character (*os*))
 \div '512)
 = tm-svcid (*os*))

EVENT: Disable restore-svcid-for-tyi-svc-return.

DEFINITION:

os-task-table-with-delivered-message (*os*)
= putseg (putnth (array-qfirst (getnth (array-qfirst (os-readyq (*os*))
+ ('16
* (getnth ('2,
tm-regs (*os*))
mod '16)),
table ('8, os-mbuffers (*os*))),
'3,
tm-cpu (*os*)),
'9 * array-qfirst (os-readyq (*os*)),
os-task-table (*os*))

THEOREM: length-os-task-table-with-delivered-message
length (os-task-table-with-delivered-message (*os*)) = '144

EVENT: Disable length-os-task-table-with-delivered-message.

THEOREM: plistp-os-task-table-with-delivered-message
plistp (os-task-table-with-delivered-message (*os*))

EVENT: Disable plistp-os-task-table-with-delivered-message.

THEOREM: restore-regs-for-receive-svc-return
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (getseg ('0,
'8,
getnth (array-qfirst (os-readyq (*os*)),
table ('9, os-task-table-with-delivered-message (*os*)))
= putnth (array-qfirst (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*,
os)),
'3,
tm-regs (*os*)))

EVENT: Disable restore-regs-for-receive-svc-return.

THEOREM: restore-cc-for-receive-svc-return
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow ((getnth ('8 + ('9 * array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-message (*os*))

$$\begin{aligned} & \text{mod } '4) \\ & = \text{tm-cc}(os) \end{aligned}$$

EVENT: Disable restore-cc-for-receive-svc-return.

$$\begin{aligned} & \text{THEOREM: restore-error-for-receive-svc-return} \\ & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ & \rightarrow (((\text{getnth}('8 + ('9 * \text{array-qfirst}(\text{os-readyq}(os))), \\ & \quad \text{os-task-table-with-delivered-message}(os)) \\ & \quad \text{mod } '256) \\ & \quad \div '4) \\ & = \text{tm-error}(os)) \end{aligned}$$

EVENT: Disable restore-error-for-receive-svc-return.

$$\begin{aligned} & \text{THEOREM: restore-svcid-for-receive-svc-return} \\ & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ & \rightarrow (((\text{getnth}('8 + ('9 * \text{array-qfirst}(\text{os-readyq}(os))), \\ & \quad \text{os-task-table-with-delivered-message}(os)) \\ & \quad \div '512) \\ & = \text{tm-svcid}(os)) \end{aligned}$$

EVENT: Disable restore-svcid-for-receive-svc-return.

EVENT: Let us define the theory *executable-fns* to consist of the following events: *1*greaterp, *1*leq, *1*geq, *1*member, *1*iff, *1*ord-lessp, *1*ordinalp, *1*assoc, *1*pairlist, *1*subrp, *1*apply-subr, *1*formals, *1*body, *1*fix-cost, *1*strip-cars, *1*sum-cdrs, *1*v&c\$, *1*v&c-apply\$, *1*eval\$, *1*quantifier-initial-value, *1*add-to-set, *1*append, *1*max, *1*union, *1*quantifier-operation, *1*for, *1*exp, *1*divides, *1*number-and-list-induction, *1*double-number-induction, *1*double-number-double-list-induction, *1*lessp-quotient-induction, *1*min, *1*lessp-quotient-quotient-induction, *1*length, *1*insert, *1*occurrences, *1*remove, *1*permutation, *1*plistp, *1*setp, *1*sorted, *1*transitivity-of-permutation-induction, *1*setp-permutation-induction, *1*getnth, *1*putnth, *1*getseg, *1*putseg, *1*fixlength, *1*firstn, *1*nthcdr, *1*getnth-getseg-induction, *1*integer-induction-instance, *1*getseg-putseg-induction, *1*nested-intervals, *1*strange-induction, *1*getnth2, *1*putnth2, *1*reverse, *1*finite-numberp, *1*number-listp, *1*finite-number-listp, *1*non-zero-listp, *1*tablep, *1*table, *1*flatten, *1*integer-nthcdr-induction, *1*finite-number-tablep, *1*non-last, *1*incr-mod, *1*decr-mod, *1*qfirst, *1*enq, *1*deq, *1*qemptyp, *1*qfullp, *1*qreplace, *1*qhead-field, *1*qtail-field, *1*qcurrlength-field, *1*qmaxlength-field, *1*qarray-field, *1*delta, *1*array-qindex-relation, *1*array-queuep, *1*array-enq, *1*array-deq, *1*array-qfirst, *1*array-qfullp, *1*array-qemptyp, *1*array-nonlast, *1*array-qreplace, *1*delta-segment, *1*mapup-queue, *1*qfirst2, *1*enq2,

*1*deq2, *1*qfullp2, *1*qempty2, *1*tm-wordsize, *1*tm-wordlub, *1*alu-
 result, *1*alu-value, *1*alu-carry, *1*tm-alu-decr, *1*tm-alu-decr-mod, *1*tm-
 alu-difference, *1*tm-alu-incr, *1*tm-alu-incr-mod, *1*tm-alu-mod, *1*tm-alu-
 mult, *1*tm-alu-plus, *1*tm-incr, *1*tm-incrn, *1*tm-decr, *1*tm-shellp, *1*tm-
 memory, *1*tm-regs, *1*tm-cc, *1*tm-error, *1*tm-svcflag, *1*tm-svcid, *1*tm-
 base, *1*tm-limit, *1*tm-slimit, *1*tm-svmode, *1*tm-rwstate, *1*tm-clock, *1*tm-
 iports, *1*tm-oports, *1*tm-regsize, *1*tm-charsize, *1*tm-ccsize, *1*tm-
 errorsize, *1*tm-svcflagsize, *1*tm-svcidsize, *1*tm-cclub, *1*tm-errorlub, *1*tm-
 svcflaglub, *1*tm-svcidlub, *1*tm-reglength, *1*tm-memlength, *1*tm-charlub, *1*tm-
 port-length, *1*tm-pc-address, *1*tm-sp-address, *1*tm-set-memory, *1*tm-
 set-regs, *1*tm-set-cc, *1*tm-set-error, *1*tm-set-svcflag, *1*tm-set-svcid, *1*tm-
 set-base, *1*tm-set-limit, *1*tm-set-slimit, *1*tm-set-svmode, *1*tm-set-rwstate, *1*tm-
 set-clock, *1*tm-set-iports, *1*tm-set-oports, *1*tm-pc, *1*tm-set-pc, *1*tm-sp, *1*tm-
 set-sp, *1*tm-non-zero-no-carry-condition, *1*tm-zero-no-carry-
 condition, *1*tm-non-zero-carry-condition, *1*tm-zero-carry-condition, *1*tm-
 cc-value, *1*tm-cc-divisor, *1*tm-error-divisor, *1*tm-svcflag-divisor, *1*tm-
 pack-psw, *1*tm-unpack-cc, *1*tm-unpack-error, *1*tm-unpack-svcflag, *1*tm-
 unpack-svcid, *1*tm-register-save-area-addr, *1*tm-clock-new-pc-addr, *1*tm-
 error-new-pc-addr, *1*tm-svc-new-pc-addr, *1*tm-input-new-pc-addr, *1*tm-
 output-new-pc-addr, *1*tm-svcid-addr, *1*tm-input-devid-addr, *1*tm-input-
 char-addr, *1*tm-output-devid-addr, *1*tm-no-error, *1*tm-halt-status, *1*tm-
 opcode-error, *1*tm-privilege-error, *1*tm-stack-overflow-error, *1*tm-stack-
 underflow-error, *1*tm-address-error, *1*tm-pc-address-error, *1*tm-supervisor-
 mode, *1*tm-user-mode, *1*tm-in-supervisor-mode, *1*tm-wait-state, *1*tm-
 run-state, *1*tm-waiting, *1*tm-svc, *1*tm-no-svc, *1*tm-cleared-svcflag, *1*tm-
 iportp, *1*tm-iinterrupt-flag, *1*tm-ierror-flag, *1*tm-ichar, *1*tm-oportp, *1*tm-
 ointerrupt-flag, *1*tm-obusy-flag, *1*tm-ochar, *1*tm-iport-errorp, *1*tm-clear-
 input-interrupt, *1*tm-post-input-interrupt, *1*tm-oport-idlep, *1*tm-start-output, *1*tm-
 post-output-interrupt, *1*tm-clear-output-interrupt, *1*real-addr, *1*real-
 addr-source, *1*real-addr-num, *1*tm-arg-mode, *1*tm-arg-datum, *1*tm-arg-
 index, *1*tm-fix-word, *1*tm-fix-reg-address, *1*tm-compute-address, *1*tm-
 good-address, *1*tm-incr-address, *1*tm-incrn-address, *1*tm-fetch-from-memory, *1*tm-
 store-in-memory, *1*tm-fetch-from-regmem, *1*tm-store-in-regmem, *1*tm-
 fetch, *1*tm-store, *1*tm-stack-overflowp, *1*tm-stack-underflowp, *1*tm-execute-
 add, *1*tm-execute-branch, *1*tm-execute-branch-on-zero, *1*tm-execute-branch-
 not-zero, *1*tm-execute-call, *1*tm-execute-compare, *1*tm-execute-decr, *1*tm-
 execute-decr-mod, *1*tm-execute-incr, *1*tm-execute-incr-mod, *1*tm-execute-
 load-base, *1*tm-execute-load-limit, *1*tm-execute-load-psw, *1*tm-execute-
 mod, *1*tm-execute-move, *1*tm-execute-multiply, *1*tm-execute-post-output-
 interrupt, *1*tm-execute-return, *1*tm-execute-run, *1*tm-execute-set-clock, *1*tm-
 execute-start-output, *1*tm-execute-svc, *1*tm-execute-svc-return, *1*tm-
 execute-test-iport, *1*tm-execute-test-oport, *1*tm-execute-wait, *1*tm-opcode-
 size, *1*tm-mode-size, *1*tm-index-size, *1*tm-opcode-divisor, *1*tm-mode1-
 divisor, *1*tm-index1-divisor, *1*tm-mode2-divisor, *1*tm-opcode, *1*tm-mode1,

$*1*tm-index1, *1*tm-mode2, *1*tm-index2, *1*tm-nullary-instructionp, *1*tm-unary-instructionp, *1*tm-datum1, *1*tm-datum2, *1*tm-arg1, *1*tm-arg2, *1*tm-return-opcode, *1*tm-wait-opcode, *1*tm-run-opcode, *1*tm-execute-nullary, *1*tm-br-opcode, *1*tm-brnz-opcode, *1*tm-brz-opcode, *1*tm-call-opcode, *1*tm-decr-opcode, *1*tm-incr-opcode, *1*tm-lbase-opcode, *1*tm-llimit-opcode, *1*tm-lpsw-opcode, *1*tm-post-opcode, *1*tm-svc-opcode, *1*tm-svc-return-opcode, *1*tm-testi-opcode, *1*tm-testo-opcode, *1*tm-time-opcode, *1*tm-execute-unary, *1*tm-add-opcode, *1*tm-compare-opcode, *1*tm-decr-mod-opcode, *1*tm-incr-mod-opcode, *1*tm-mod-opcode, *1*tm-move-opcode, *1*tm-mult-opcode, *1*tm-stout-opcode, *1*tm-execute-binary, *1*tm-execute-instruction, *1*tm-fetch-opcode, *1*tm-fetch-arg1, *1*tm-fetch-arg2, *1*tm-execute, *1*tm-decrement-clock, *1*tm-good-pc-address, *1*tm-fetch-execute, *1*tm-fetch-new-pc-on-interrupt, *1*tm-store-old-psw-on-interrupt, *1*tm-clock-interruptp, *1*tm-execute-clock-interrupt, *1*tm-errorp, *1*tm-execute-error-interrupt, *1*tm-svc-interruptp, *1*tm-store-svcid-on-interrupt, *1*tm-execute-svc-interrupt, *1*tm-overflow-char, *1*tm-some-input-interruptp, *1*tm-input-interruptp, *1*tm-interrupting-input-port, *1*tm-store-interrupting-input-device, *1*tm-execute-input-interrupt, *1*tm-some-output-interruptp, *1*tm-output-interruptp, *1*tm-interrupting-output-port, *1*tm-store-interrupting-output-device, *1*tm-execute-output-interrupt, *1*tm-device-input-eventp, *1*tm-idevid, *1*tm-idatum, *1*tm-device-output-eventp, *1*tm-odevid, *1*tm-post-interrupt, *1*tm-step, *1*tm-processor.$

DEFINITION:

os-intended-clock-interrupt (tm)

$= \text{tm-set-memory}(\text{putnth}(\text{tm-pc}(tm),$
 $\quad '0,$
 $\quad \text{putnth}(\text{tm-sp}(tm),$
 $\quad \quad '1,$
 $\quad \quad \text{putnth}(\text{tm-pack-psw}(\text{tm-cc}(tm),$
 $\quad \quad \quad \text{tm-error}(tm),$
 $\quad \quad \quad \text{tm-svcflag}(tm),$
 $\quad \quad \quad \text{tm-svcid}(tm)),$
 $\quad \quad '2,$
 $\quad \quad \text{putseg}(\text{os-task-table}(tm),$
 $\quad \quad \quad '15,$
 $\quad \quad \quad \text{putseg}(\text{os-segment-table}(tm),$
 $\quad \quad \quad \quad '159,$
 $\quad \quad \quad \quad \text{putseg}(\text{os-readyq}(tm),$
 $\quad \quad \quad \quad \quad '191,$
 $\quad \quad \quad \quad \quad \text{putseg}(\text{os-code}(tm),$
 $\quad \quad \quad \quad \quad \quad '2547,$
 $\quad \quad \quad \quad \quad \quad \text{tm-memory}(tm))))))))),$

```

tm-set-pc ('2654,
           tm-set-sp ('3310,
                     tm-set-svmode ('1,
                                    tm-set-error ('0,
                                                  tm-set-rwstate ('0,
                                                                tm))))))

```

THEOREM: equivalence-of-clock-interrupt-expressions
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow (\text{tm-execute-clock-interrupt}(os) = \text{os-intended-clock-interrupt}(os))$

EVENT: Disable equivalence-of-clock-interrupt-expressions.

DEFINITION:

```

os-time-for-clock-interrupt-handler-thru-save-state(os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-CLOCK-INTERRUPT-HANDLER-THRU-SAVE-STATE
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
              (TM-ERROR OS)
              (TM-SVCFLAG OS)
              (TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)

```

```

'15
(PUTSEG (OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2656
'3310
(TM-MEMORY OS))))))))))
(PUTNTH '2656
'0
(PUTNTH '3310
'1
(PUTNTH (TM-R2 OS)
'2
(PUTNTH (TM-R3 OS)
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
(TIMES '9
(ARRAY-QFIRST (OS-READYQ OS))))
'8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-clock-interrupt-handler-thru-save-state
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-clock-interrupt}(os),$
 $\text{os-time-for-clock-interrupt-handler-thru-save-state}(os))$
 $= \text{os-clock-interrupt-handler-thru-save-state}(os))$

EVENT: Disable trace-clock-interrupt-handler-thru-save-state.

DEFINITION:

```
os-time-for-clock-interrupt-handler-to-dispatcher (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-CLOCK-INTERRUPT-HANDLER-TO-DISPATCHER
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG (OS-SEGMENT-TABLE OS)
'159
(PUTSEG (ARRAY-ENQ (ARRAY-QFIRST (OS-READYQ OS))
(Array-DEQ (OS-READYQ OS)))
'191
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2665
'3310
(TM-MEMORY OS))))))))))
(PUTNTH '3206
```

```

'0
(PUTNTH '3310
  '1
    (PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
      '2
        (PUTNTH '191
          '3
            (PUTNTH (PLUS '195
              (GETNTH '1
                (ARRAY-DEQ (OS-READYQ OS))))
              '4
                (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1
  (ARRAY-DEQ (OS-READYQ OS)))
  '16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS)
NIL)

```

THEOREM: trace-clock-interrupt-handler-to-dispatcher
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-clock-interrupt-handler-thru-save-state}(os),$
 $\text{os-time-for-clock-interrupt-handler-to-dispatcher}(os))$
 $= \text{os-clock-interrupt-handler-to-dispatcher}(os))$

EVENT: Disable trace-clock-interrupt-handler-to-dispatcher.

DEFINITION:
 $\text{os-time-for-clock-interrupt-handler-to-restore-state}(os)$
 $= \text{'(tick tick tick tick tick tick tick tick tick tick tick)}$

(DEFN OS-CLOCK-INTERRUPT-HANDLER-TO-RESTORE-STATE

```

(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG (OS-SEGMENT-TABLE OS)
'159
(PUTSEG (ARRAY-ENQ (ARRAY-QFIRST (OS-READYQ OS))
(ARRAY-DEQ (OS-READYQ OS)))
'191
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3217
'3310
(TM-MEMORY OS))))))))))
(PUTNTH '2601
'0
(PUTNTH '3309
'1
(PUTNTH (ARRAY-QFIRST (ARRAY-ENQ (ARRAY-QFIRST (OS-READYQ OS))
(ARRAY-DEQ (OS-READYQ OS))))
'2
(PUTNTH '191
'3

```

```

(PUTNTH (PLUS '195
          (GETNTH '1
                  (ARRAY-DEQ (OS-READYQ OS))))))
'4
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS '195
                (GETNTH '0
                        (ARRAY-ENQ (ARRAY-QFIRST (OS-READYQ OS))
                                   (ARRAY-DEQ (OS-READYQ OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS)
NIL)

```

THEOREM: trace-clock-interrupt-handler-to-restore-state
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-clock-interrupt-handler-to-dispatcher}(os),$
 $\text{os-time-for-clock-interrupt-handler-to-restore-state}(os))$
 $= \text{os-clock-interrupt-handler-to-restore-state}(os))$

EVENT: Disable trace-clock-interrupt-handler-to-restore-state.

DEFINITION:
 $\text{os-clock-new-readyq}(os)$
 $= \text{array-enq}(\text{array-qfirst}(\text{os-readyq}(os)), \text{array-deq}(\text{os-readyq}(os)))$

DEFINITION:
 $\text{os-time-for-clock-interrupt-handler-to-finish}(os)$
 $= \text{'(tick tick tick tick tick tick tick tick tick tick}$
 $\text{tick tick tick tick tick tick tick tick tick tick}$
 tick)

(DEFN OS-CLOCK-INTERRUPT-HANDLER
(OS)

```

(TM
(PUTNTH
  (GETNTH (TIMES '9
            (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS)))
            (OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
  (GETNTH (PLUS '1
              (TIMES '9
                    (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))))
            (OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
  (GETNTH (PLUS '8
              (TIMES '9
                    (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))))
            (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG (OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-CLOCK-NEW-READYQ OS)
'191
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3217
'3310
(TM-MEMORY OS))))))))))
(OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
OS)
(OS-NEW-CC (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
OS)
(OS-NEW-ERROR (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
OS)

```

```

(OS-NEW-SVCFLAG (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                OS)
(OS-NEW-SVCID (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
              OS)
(OS-NEW-BASE (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
             OS)
(OS-NEW-LIMIT (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
              OS)
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-clock-interrupt-handler-to-finish
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-clock-interrupt-handler-to-restore-state}(os),$
 $\text{os-time-for-clock-interrupt-handler-to-finish}(os))$
 $= \text{os-clock-interrupt-handler}(os))$

EVENT: Disable trace-clock-interrupt-handler-to-finish.

DEFINITION:

```

os-cih-time1(os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)

```

DEFINITION:

```

os-cih-time2(os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)

```

DEFINITION:

```

os-time-for-clock-interrupt-handler(os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick)

```

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick

```

THEOREM: rewrite-clock-trace1

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-clock-interrupt}(os), \\
& \quad \text{os-time-for-clock-interrupt-handler}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-clock-interrupt}(os), \\
& \quad \text{os-time-for-clock-interrupt-handler-thru-save-state}(os)), \\
& \quad \text{os-cih-time1}(os))
\end{aligned}$$

EVENT: Disable rewrite-clock-trace1.

THEOREM: rewrite-clock-trace2

$$\begin{aligned}
& \text{tm-processor}(\text{os-clock-interrupt-handler-thru-save-state}(os), \text{os-cih-time1}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-clock-interrupt-handler-thru-save-state}(os), \\
& \quad \text{os-time-for-clock-interrupt-handler-to-dispatcher}(os)), \\
& \quad \text{os-cih-time2}(os))
\end{aligned}$$

EVENT: Disable rewrite-clock-trace2.

THEOREM: rewrite-clock-trace3

$$\begin{aligned}
& \text{tm-processor}(\text{os-clock-interrupt-handler-to-dispatcher}(os), \text{os-cih-time2}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-clock-interrupt-handler-to-dispatcher}(os), \\
& \quad \text{os-time-for-clock-interrupt-handler-to-restore-state}(os)), \\
& \quad \text{os-time-for-clock-interrupt-handler-to-finish}(os))
\end{aligned}$$

EVENT: Disable rewrite-clock-trace3.

THEOREM: trace-clock-interrupt-handler

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-clock-interrupt}(os), \\
& \quad \text{os-time-for-clock-interrupt-handler}(os)) \\
= & \text{os-clock-interrupt-handler}(os))
\end{aligned}$$

EVENT: Disable trace-clock-interrupt-handler.

DEFINITION:

$$\begin{aligned}
& \text{os-intended-error-interrupt}(tm) \\
= & \text{tm-set-memory}(\text{putnth}(\text{tm-pc}(tm),
\end{aligned}$$

```

'0,
putnth (tm-sp (tm),
      '1,
      putnth (tm-pack-psw (tm-cc (tm),
                        tm-error (tm),
                        tm-svcflag (tm),
                        tm-svcid (tm)),
            '2,
            putseg (os-task-table (tm),
                  '15,
                  putseg (os-segment-table (tm),
                        '159,
                        putseg (os-readyq (tm),
                              '191,
                              putseg (os-status-table (tm),
                                    '211,
                                    putseg (os-code (tm),
                                            '2547,
                                            tm-memory (tm))))))))),
tm-set-pc ('2667,
          tm-set-sp ('3310,
                    tm-set-svmode ('1,
                                    tm-set-error ('0,
                                                  tm-set-rwstate ('0,
                                                                tm))))))

```

THEOREM: equivalence-of-error-interrupt-expressions
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-execute-error-interrupt}(os) = \text{os-intended-error-interrupt}(os))$

EVENT: Disable equivalence-of-error-interrupt-expressions.

DEFINITION:

```

os-time-for-error-handler-thru-save-state (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-ERROR-HANDLER-THRU-SAVE-STATE
  (OS)
  (TM
   (PUTNTH

```



```

(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2669
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '2669
'0
(PUTNTH '3310
'1
(PUTNTH (TM-R2 OS)
'2
(PUTNTH (TM-R3 OS)
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15

```

```

(TIMES '9
  (ARRAY-QFIRST (OS-READYQ OS)))
'8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS)
NIL)

```

THEOREM: trace-error-handler-thru-save-state
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-error-interrupt}(os),$
 $\text{os-time-for-error-handler-thru-save-state}(os))$
 $= \text{os-error-handler-thru-save-state}(os))$

EVENT: Disable trace-error-handler-thru-save-state.

DEFINITION:
os-time-for-error-handler-to-dispatcher(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

```

(DEFN OS-ERROR-HANDLER-TO-DISPATCHER
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)

```

```

                                (TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'1
'0
OS)
'211
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2676
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH (PLUS '211
(TIMES '2
(ARRAY-QFIRST (OS-READYQ OS))))
'2
(PUTNTH '191 '3 (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '2
(ARRAY-QFIRST (OS-READYQ OS)))
'211))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)

```

```

(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-error-handler-to-dispatcher
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-error-handler-thru-save-state}(os),$
 $\text{os-time-for-error-handler-to-dispatcher}(os))$
 $= \text{os-error-handler-to-dispatcher}(os))$

EVENT: Disable trace-error-handler-to-dispatcher.

DEFINITION:
 $\text{os-time-for-error-handler-to-wait-state}(os)$
 $= \text{'(tick tick tick tick tick tick)}$

```

(DEFN OS-ERROR-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14

```

```

(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'1
'0
OS)
'211
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3211
'3310
(TM-MEMORY OS))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (PLUS '211
(TIMES '2
(ARRAY-QFIRST (OS-READYQ OS))))
'2
(PUTNTH '191 '3 (TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-error-handler-to-wait-state

```

(good-os (os)
  ^ ((¬ tm-waiting (os)) ∧ array-qempty (array-deq (os-readyq (os))))
  → (tm-processor (os-error-handler-to-dispatcher (os),
                  os-time-for-error-handler-to-wait-state (os))
     = os-error-handler-path1 (os))

```

EVENT: Disable trace-error-handler-to-wait-state.

DEFINITION:

```

os-time-for-error-handler-to-restore-state (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

```

(DEFN OS-ERROR-HANDLER-TO-RESTORE-STATE
  (OS)
  (TM
    (PUTNTH
      (TM-RO OS)
      '0
    (PUTNTH
      (TM-R1 OS)
      '1
    (PUTNTH
      (TM-PACK-PSW (TM-CC OS)
                   (TM-ERROR OS)
                   (TM-SVCFLAG OS)
                   (TM-SVCID OS))
      '2
    (PUTNTH
      (TM-R2 OS)
      '13
    (PUTNTH
      (TM-R3 OS)
      '14
    (PUTSEG
      (OS-NEW-TASK-TABLE OS)
      '15
    (PUTSEG
      (OS-SEGMENT-TABLE OS)
      '159
    (PUTSEG (ARRAY-DEQ (OS-READYQ OS))
            '191

```

```

(PUTSEG (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'1
'0
OS)
'211
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3217
'3310
(TM-MEMORY OS)))))))))
(PUTNTH '2601
'0
(PUTNTH '3309
'1
(PUTNTH (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
'2
(PUTNTH '191 '3 (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS '195
(GETNTH '0
(ARRAY-DEQ (OS-READYQ OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-error-handler-to-restore-state

$$\begin{aligned}
& \text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow & (\text{tm-processor}(\text{os-error-handler-to-dispatcher}(os), \\
& \quad \text{os-time-for-error-handler-to-restore-state}(os)) \\
& = \text{os-error-handler-to-restore-state}(os))
\end{aligned}$$

EVENT: Disable trace-error-handler-to-restore-state.

DEFINITION:

os-time-for-error-handler-to-finish(*os*)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick)
```

```
(DEFN OS-ERROR-HANDLER-PATH2
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
          (TIMES '9
                (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
          (TIMES '9
                (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'1
```



```

'0
OS)
'211
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3217
'3310
(TM-MEMORY OS)))))))))))))
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-BASE (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-LIMIT (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-error-handler-to-finish

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\neg \text{array-qempty } (\text{array-deq } (\text{os-readyq } (os)))))) \\
\rightarrow & (\text{tm-processor } (\text{os-error-handler-to-restore-state } (os), \\
& \quad \text{os-time-for-error-handler-to-finish } (os)) \\
& = \text{os-error-handler-path2 } (os))
\end{aligned}$$

EVENT: Disable trace-error-handler-to-finish.

DEFINITION:

os-error-time1 (*os*)

= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick)

DEFINITION:

os-time-for-error-handler-path1 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

THEOREM: rewrite-error-trace1

tm-processor (os-intended-error-interrupt (*os*),
os-time-for-error-handler-path1 (*os*))
= tm-processor (tm-processor (os-intended-error-interrupt (*os*),
os-time-for-error-handler-thru-save-state (*os*)),
os-error-time1 (*os*))

EVENT: Disable rewrite-error-trace1.

THEOREM: rewrite-error-trace2

tm-processor (os-error-handler-thru-save-state (*os*), os-error-time1 (*os*))
= tm-processor (tm-processor (os-error-handler-thru-save-state (*os*),
os-time-for-error-handler-to-dispatcher (*os*)),
os-time-for-error-handler-to-wait-state (*os*))

EVENT: Disable rewrite-error-trace2.

THEOREM: trace-error-handler-path1

(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*) \wedge array-qempty (array-deq (os-readyq (*os*))))))
 \rightarrow (tm-processor (os-intended-error-interrupt (*os*),
os-time-for-error-handler-path1 (*os*))
= os-error-handler-path1 (*os*))

EVENT: Disable trace-error-handler-path1.

DEFINITION:

os-error-time2 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

DEFINITION:

```
os-error-time3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)
```

DEFINITION:

```
os-time-for-error-handler-path2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick)
```

THEOREM: rewrite-error-trace3

```
tm-processor (os-intended-error-interrupt (os),
              os-time-for-error-handler-path2 (os))
= tm-processor (tm-processor (os-intended-error-interrupt (os),
                             os-time-for-error-handler-thru-save-state (os)),
               os-error-time2 (os))
```

EVENT: Disable rewrite-error-trace3.

THEOREM: rewrite-error-trace4

```
tm-processor (os-error-handler-thru-save-state (os), os-error-time2 (os))
= tm-processor (tm-processor (os-error-handler-thru-save-state (os),
                             os-time-for-error-handler-to-dispatcher (os)),
               os-error-time3 (os))
```

EVENT: Disable rewrite-error-trace4.

THEOREM: rewrite-error-trace5

```
tm-processor (os-error-handler-to-dispatcher (os), os-error-time3 (os))
= tm-processor (tm-processor (os-error-handler-to-dispatcher (os),
                             os-time-for-error-handler-to-restore-state (os)),
               os-time-for-error-handler-to-finish (os))
```

EVENT: Disable rewrite-error-trace5.

THEOREM: trace-error-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\neg \text{array-qempty} (\text{array-deq } (\text{os-readyq } (os)))))) \\
\rightarrow & (\text{tm-processor } (\text{os-intended-error-interrupt } (os), \\
& \quad \text{os-time-for-error-handler-path2 } (os)) \\
& = \text{os-error-handler-path2 } (os))
\end{aligned}$$

EVENT: Disable trace-error-handler-path2.

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-error-handler } (os) \\
= & \text{ if } \text{array-qempty} (\text{array-deq } (\text{os-readyq } (os))) \\
& \text{ then } \text{os-time-for-error-handler-path1 } (os) \\
& \text{ else } \text{os-time-for-error-handler-path2 } (os) \text{ endif}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-error-handler } (os) \\
= & \text{ if } \text{array-qempty} (\text{array-deq } (\text{os-readyq } (os))) \\
& \text{ then } \text{os-error-handler-path1 } (os) \\
& \text{ else } \text{os-error-handler-path2 } (os) \text{ endif}
\end{aligned}$$

THEOREM: trace-os-error-handler

$$\begin{aligned}
& (\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{tm-processor } (\text{os-intended-error-interrupt } (os), \\
& \quad \text{os-time-for-error-handler } (os)) \\
& = \text{os-error-handler } (os))
\end{aligned}$$

EVENT: Disable trace-os-error-handler.

DEFINITION:

$$\begin{aligned}
& \text{os-intended-svc-send-interrupt } (tm) \\
= & \text{tm-set-memory } (\text{putnth } (\text{tm-pc } (tm), \\
& \quad '0, \\
& \quad \text{putnth } (\text{tm-sp } (tm), \\
& \quad \quad '1, \\
& \quad \quad \text{putnth } (\text{tm-pack-psw } (\text{tm-cc } (tm), \\
& \quad \quad \quad \text{tm-error } (tm), \\
& \quad \quad \quad \text{tm-svcflag } (tm), \\
& \quad \quad \quad \text{tm-svcid } (tm)), \\
& \quad \quad \quad '2, \\
& \quad \quad \quad \text{putnth } (\text{tm-svcid } (tm), \\
& \quad \quad \quad \quad '8, \\
& \quad \quad \quad \quad \text{putseg } (\text{os-task-table } (tm), \\
& \quad \quad \quad \quad \quad '15,
\end{aligned}$$

```

tm-set-pc ('2690,
           tm-set-sp ('3310,
                      tm-set-svmode ('1,
                                     tm-set-error ('0,
                                                    tm-set-rwstate ('0,
                                                                    tm-memor
putseg (os-segment-table (tm),
       '159,
putseg (os-readyq (tm),
       '191,
putseg (os-status-table (tm),
       '211,
putseg (os-mbuffers (tm),
       '499,
putseg (os-code (tm),
       '2547,
tm-memor

```

THEOREM: equivalence-of-svc-send-interrupt-expressions
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow (\text{tm-execute-svc-interrupt}(os) = \text{os-intended-svc-send-interrupt}(os))$

EVENT: Disable equivalence-of-svc-send-interrupt-expressions.

DEFINITION:

```

os-time-for-svc-send-handler-thru-save-state (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-THRU-SAVE-STATE
  (OS)
  (TM
   (PUTNTH
    (TM-RO OS)
    '0
   (PUTNTH
    (TM-R1 OS)
    '1
   (PUTNTH
    (TM-PACK-PSW (TM-CC OS)
                  (TM-ERROR OS)

```

```

                (TM-SVCFLAG OS)
                (TM-SVCID OS))
'2
(PUTNTH
(TM-SVCID OS)
'8
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2692
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2692
'0
(PUTNTH '3310
'1
(PUTNTH (TM-R2 OS)
'2
(PUTNTH (TM-R3 OS)
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
(TIMES '9
(ARRAY-QFIRST (OS-READYQ OS))))))

```

```

'8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-thru-save-state
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (tm-processor (os-intended-svc-send-interrupt(*os*),
os-time-for-svc-send-handler-thru-save-state(*os*))
= os-svc-send-handler-thru-save-state(*os*))

EVENT: Disable trace-svc-send-handler-thru-save-state.

DEFINITION:
os-time-for-svc-send-handler-to-qfullp-test(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

```

(DEFN OS-SVC-SEND-HANDLER-TO-QFULLP-TEST
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2

```

```

(PUTNTH '0
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (OS-READYQ OS)
'191
(PUTSEG
  (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2726
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2747
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH (PLUS '499
(PUTNTH (PLUS (TIMES '8
(REMAINDER (TM-R2 OS) '16))
(TIMES '128
(ARRAY-QFIRST (OS-READYQ OS))))))
'3

```



```

(PUTNTH (TIMES '8
        (REMAINDER (TM-R2 OS) '16))
  '4
  (PUTNTH (REMAINDER (TM-R2 OS) '16)
    '6
    (PUTNTH (TM-R3 OS)
      '7
      (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS (TIMES '8
                                (REMAINDER (GETNTH '2 (TM-REGS OS)
                                                  '16))
                                (TIMES '128
                                      (ARRAY-QFIRST (OS-READYQ OS))))
          '499))
  '0
  (TM-SVCFLAG OS)
  (TM-SVCID OS)
  (TM-BASE OS)
  (TM-LIMIT OS)
  (TM-SLIMIT OS)
  '1
  '0
  (TM-CLOCK OS)
  (TM-IPOINTS OS)
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-qfullp-test
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '0)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-send-handler-thru-save-state}(os),$
 $\quad \text{os-time-for-svc-send-handler-to-qfullp-test}(os))$
 $= \text{os-svc-send-handler-to-qfullp-test}(os))$

EVENT: Disable trace-svc-send-handler-to-qfullp-test.

DEFINITION:

```

os-time-for-svc-send-handler-to-dispatcher(os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-TO-DISPATCHER
  (OS)

```

```

(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '0
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'2
(OS-DESTID OS '16)
OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558

```

```

'3309
(PUTNTH '2796
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH (PLUS '211
(TIMES '2
(ARRAY-QFIRST (OS-READYQ OS))))
'3
(PUTNTH (TIMES '8
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'6
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS)))))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '2
(ARRAY-QFIRST (OS-READYQ OS)))
'211))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-dispatcher
(good-os (*os*)
 \wedge (\neg tm-waiting (*os*))
 \wedge (((tm-svcid (*os*) **mod** '4) = '0)

```

      ^ array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                         os-destid (os, '16),
                                         os))))))
→ (tm-processor (os-svc-send-handler-to-qfullp-test (os),
              os-time-for-svc-send-handler-to-dispatcher (os))
   = os-svc-send-handler-to-dispatcher (os))

```

EVENT: Disable trace-svc-send-handler-to-dispatcher.

DEFINITION:

```

os-time-for-svc-send-handler-to-wait-state (os)
= '(tick tick tick tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-PATH1
 (OS)
 (TM
  (PUTNTH
   (TM-R0 OS)
   '0
  (PUTNTH
   (TM-R1 OS)
   '1
  (PUTNTH
   (TM-PACK-PSW (TM-CC OS)
                 (TM-ERROR OS)
                 (TM-SVCFLAG OS)
                 (TM-SVCID OS))
   '2
  (PUTNTH '0
   '8
  (PUTNTH
   (ARRAY-QFIRST (OS-READYQ OS))
   '12
  (PUTNTH
   (TM-R2 OS)
   '13
  (PUTNTH
   (TM-R3 OS)
   '14
  (PUTSEG
   (OS-NEW-TASK-TABLE OS)
   '15

```

```

(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'2
(OS-DESTID OS '16)
OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3211
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH '191
'3
(PUTNTH (TIMES '8
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'6
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)

```

```

(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-wait-state

```

(good-os (os)
  ^ ((¬ tm-waiting (os)) ^ array-qempty (array-deq (os-readyq (os))))))
→ (tm-processor (os-svc-send-handler-to-dispatcher (os),
                os-time-for-svc-send-handler-to-wait-state (os))
   = os-svc-send-handler-path1 (os))

```

EVENT: Disable trace-svc-send-handler-to-wait-state.

DEFINITION:

```

os-time-for-svc-send-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

(DEFN OS-SVC-SEND-HANDLER-PATH2

```

(OS)
(TM
 (PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
  (OS-NEW-TASK-TABLE OS))
 '0
 (PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
  (OS-NEW-TASK-TABLE OS))
 '1
 (PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9

```

```

                                (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
                                (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '0
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
  (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
                    '2
                    (OS-DESTID OS '16)
                    OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3217
'3310
(TM-MEMORY OS))))))))))))))
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)

```

```

(OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-BASE (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-LIMIT (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-lpsw

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
    ^ (¬ array-qempty (array-deq (os-readyq (os)))))
→ (tm-processor (os-svc-send-handler-to-dispatcher (os),
  os-time-for-svc-send-handler-to-lpsw (os))
  = os-svc-send-handler-path2 (os))

```

EVENT: Disable trace-svc-send-handler-to-lpsw.

DEFINITION:

```

os-time-for-svc-send-handler-thru-enqueue (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-THRU-ENQUEUE
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)

```



```

                (TM-ERROR OS)
                (TM-SVCFLAG OS)
                (TM-SVCID OS))
'2
(PUTNTH '0
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (OS-READYQ OS)
'191
(PUTSEG
  (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2756
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2756
'0
(PUTNTH '3310
'1
(PUTNTH
  (TM-R3 OS)
'2
(PUTNTH

```

```

(PLUS '499
  (PLUS (TIMES '8
    (REMAINDER (GETNTH '2 (TM-REGS OS))
      '16))
    (TIMES '128
      (ARRAY-QFIRST (OS-READYQ OS))))))
'3
(PUTNTH
  (PLUS '503
    (PLUS (PLUS (TIMES '8
      (REMAINDER (GETNTH '2 (TM-REGS OS))
        '16))
      (TIMES '128
        (ARRAY-QFIRST (OS-READYQ OS))))
      (GETNTH '1
        (GETNTH (PLUS (TIMES '16
          (ARRAY-QFIRST (OS-READYQ OS)))
            (REMAINDER (GETNTH '2 (TM-REGS OS))
              '16))
          (TABLE '8 (OS-MBUFFERS OS))))))
    (TABLE '8 (OS-MBUFFERS OS))))))
'4
(PUTNTH (REMAINDER (TM-R2 OS) '16)
  '6
  (PUTNTH (TM-R3 OS)
    '7
    (TM-REGS OS))))))
(TM-CC-VALUE
  (TM-ALU-INCR-MOD (GETNTH '1
    (GETNTH (PLUS (TIMES '16
      (ARRAY-QFIRST (OS-READYQ OS)))
      (REMAINDER (GETNTH '2 (TM-REGS OS))
        '16))
      (TABLE '8 (OS-MBUFFERS OS))))
    (GETNTH '3
      (GETNTH (PLUS (TIMES '16
        (ARRAY-QFIRST (OS-READYQ OS)))
        (REMAINDER (GETNTH '2 (TM-REGS OS))
          '16))
        (TABLE '8 (OS-MBUFFERS OS))))))
    '0
    (TM-SVCFLAG OS)
    (TM-SVCID OS)
    (TM-BASE OS)
    (TM-LIMIT OS)

```

```

(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-thru-enqueue

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (((tm-svcid (os) mod '4) = '0)
        ^ (¬ array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                                os-destid (os,
                                                            '16),
                                                            os))))))
  → (tm-processor (os-svc-send-handler-to-qfullp-test (os),
                 os-time-for-svc-send-handler-thru-enqueue (os))
     = os-svc-send-handler-thru-enqueue (os))

```

EVENT: Disable trace-svc-send-handler-thru-enqueue.

DEFINITION:

```

os-time-for-svc-send-handler-to-svc-resume-task1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

(DEFN OS-SVC-SEND-HANDLER-TO-SVC-RESUME-TASK1

```

(OS)
(TM
 (PUTNTH
  (TM-R0 OS)
  '0
 (PUTNTH
  (TM-R1 OS)
  '1
 (PUTNTH
  (TM-PACK-PSW (TM-CC OS)
               (TM-ERROR OS)
               (TM-SVCFLAG OS)
               (TM-SVCID OS))
  '2

```

```

(PUTNTH '0
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (ARRAY-ENQ (OS-DESTID OS '16)
              (OS-READYQ OS))
'191
(PUTSEG
  (OS-UPDATE-STATUS (OS-DESTID OS '16)
                    '0
                    '0
                    OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2789
'3310
(TM-MEMORY OS))))))))))))))

(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'2
(PUTNTH '191

```

```

'3
(PUTNTH (PLUS '195
          (GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
                    '16)
'6
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
                              '16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS)
NIL)

```

THEOREM: trace-svc-send-handler-to-svc-resume-task1

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ ((getnth ('2 * os-destid (os, '16), os-status-table (os))
                = '3)
        ^ (getnth ('1 + ('2 * os-destid (os, '16)),
                    os-status-table (os))
              = os-current-taskid (os))))))
→ (tm-processor (os-svc-send-handler-thru-enqueue (os),
               os-time-for-svc-send-handler-to-svc-resume-task1 (os))
   = os-svc-send-handler-to-svc-resume-task1 (os))

```

EVENT: Disable trace-svc-send-handler-to-svc-resume-task1.

DEFINITION:

```

os-time-for-svc-send-handler-to-svc-resume-task2 (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-TO-SVC-RESUME-TASK2
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '0
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558

```

```

'3309
(PUTNTH '2756
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-R3 OS)
'2
(PUTNTH
(PLUS '211
(TIMES '2 (OS-DESTID OS '16)))
'3
(PUTNTH
(PLUS '503
(PLUS (PLUS (TIMES '8
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
(TIMES '128
(ARRAY-QFIRST (OS-READYQ OS))))
(GETNTH '1
(GETNTH (PLUS (TIMES '16
(ARRAY-QFIRST (OS-READYQ OS)))
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
(TABLE '8 (OS-MBUFFERS OS))))))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'6
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE (GETNTH (TIMES '2
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
(OS-STATUS-TABLE OS))
'3))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)

```

```

(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-svc-resume-task2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
     ∧ (getnth ('2 * os-destid (os, '16), os-status-table (os))
        ≠ '3)))
→ (tm-processor (os-svc-send-handler-thru-enqueue (os),
                os-time-for-svc-send-handler-to-svc-resume-task2 (os))
   = os-svc-send-handler-to-svc-resume-task2 (os))

```

EVENT: Disable trace-svc-send-handler-to-svc-resume-task2.

DEFINITION:

```

os-time-for-svc-send-handler-to-svc-resume-task3 (os)
= '(tick tick tick tick tick tick tick)

```

(DEFN OS-SVC-SEND-HANDLER-TO-SVC-RESUME-TASK3

```

(OS)
(TM
  (PUTNTH
    (TM-R0 OS)
    '0
  (PUTNTH
    (TM-R1 OS)
    '1
  (PUTNTH
    (TM-PACK-PSW (TM-CC OS)
                  (TM-ERROR OS)
                  (TM-SVCFLAG OS)
                  (TM-SVCID OS))
    '2
  (PUTNTH '0
    '8

```



```

(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
  '12
(PUTNTH
  (TM-R2 OS)
  '13
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (OS-READYQ OS)
  '191
(PUTSEG
  (OS-STATUS-TABLE OS)
  '211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
  '499
(PUTSEG (OS-CODE OS)
  '2547
(PUTNTH '2558
  '3309
(PUTNTH '2756
  '3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
  '0
(PUTNTH '3310
  '1
(PUTNTH
  (TM-R3 OS)
  '2
(PUTNTH
  (PLUS '211
    (TIMES '2 (OS-DESTID OS '16)))
  '3
(PUTNTH
  (PLUS '503
    (PLUS (PLUS (TIMES '8

```

```

(REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
(TIMES '128
 (ARRAY-QFIRST (OS-READYQ OS)))
(GETNTH '1
 (GETNTH (PLUS (TIMES '16
 (ARRAY-QFIRST (OS-READYQ OS)))
 (REMAINDER (GETNTH '2 (TM-REGS OS))
'16))
 (TABLE '8 (OS-MBUFFERS OS))))))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'6
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE (GETNTH (PLUS '1
(TIMES '2
(REMAINDER (GETNTH '2 (TM-REGS OS))
'16)))
(OS-STATUS-TABLE OS))
 (ARRAY-QFIRST (OS-READYQ OS))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-svc-resume-task3

```

(good-os (os)
 ^ ((¬ tm-waiting (os))
    ^ ((getnth ('2 * os-destid (os, '16), os-status-table (os))
              = '3)
       ^ (getnth ('1 + ('2 * os-destid (os, '16)),
                  os-status-table (os))

```

```

→ (tm-processor (os-svc-send-handler-thru-enqueue (os),
            ≠ os-current-taskid (os))))
            os-time-for-svc-send-handler-to-svc-resume-task3 (os))
= os-svc-send-handler-to-svc-resume-task3 (os)

```

EVENT: Disable trace-svc-send-handler-to-svc-resume-task3.

DEFINITION:

```

os-time-for-svc-send-handler-to-svcr (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-SVC-SEND-HANDLER-PATH3
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
        (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
        (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '0
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH

```

```

(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (ARRAY-ENQ (OS-DESTID OS '16)
              (OS-READYQ OS))
'191
(PUTSEG
  (OS-UPDATE-STATUS (OS-DESTID OS '16)
                    '0
                    '0
                    OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
'499
  (PUTSEG (OS-CODE OS)
'2547
    (PUTNTH '2558
'3309
      (PUTNTH '3229
'3310
        (TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-svcr1
(good-os(*os*))

```

^ ((¬ tm-waiting (os))
  ^ ((getnth ('2 * os-destid (os, '16), os-status-table (os))
      = '3)
     ^ (getnth ('1 + ('2 * os-destid (os, '16)),
                os-status-table (os))
        = os-current-taskid (os))))
→ (tm-processor (os-svc-send-handler-to-svc-resume-task1 (os),
                os-time-for-svc-send-handler-to-svcr (os))
    = os-svc-send-handler-path3 (os))

```

EVENT: Disable trace-svc-send-handler-to-svcr1.

```

(DEFN
OS-SVC-SEND-HANDLER-PATH4
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '0
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH

```

```

(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-ENQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3229
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-send-handler-to-svcr2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-send-handler-to-svc-resume-task2}(os),$
 $\text{os-time-for-svc-send-handler-to-svcr}(os))$
 $= \text{os-svc-send-handler-path4}(os))$

EVENT: Disable trace-svc-send-handler-to-svcr2.

THEOREM: trace-svc-send-handler-to-svcr3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-processor (os-svc-send-handler-to-svc-resume-task3 (*os*),
 os-time-for-svc-send-handler-to-svcr (*os*))
 = os-svc-send-handler-path4 (*os*))

EVENT: Disable trace-svc-send-handler-to-svcr3.

DEFINITION:
 os-svc-send-time1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick)

DEFINITION:
 os-svc-send-time2 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick)

DEFINITION:
 os-time-for-svc-send-handler-path1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick)

THEOREM: rewrite-svc-send-trace1
 tm-processor (os-intended-svc-send-interrupt (*os*),
 os-time-for-svc-send-handler-path1 (*os*))
 = tm-processor (tm-processor (os-intended-svc-send-interrupt (*os*),
 os-time-for-svc-send-handler-thru-save-state (*os*)),
 os-svc-send-time1 (*os*))

EVENT: Disable rewrite-svc-send-trace1.

THEOREM: rewrite-svc-send-trace2
 tm-processor (os-svc-send-handler-thru-save-state (*os*), os-svc-send-time1 (*os*))
 = tm-processor (tm-processor (os-svc-send-handler-thru-save-state (*os*),

os-time-for-svc-send-handler-to-qfullp-test (*os*),
os-svc-send-time2 (*os*)

EVENT: Disable rewrite-svc-send-trace2.

THEOREM: rewrite-svc-send-trace3

tm-processor (os-svc-send-handler-to-qfullp-test (*os*), os-svc-send-time2 (*os*))
= tm-processor (tm-processor (os-svc-send-handler-to-qfullp-test (*os*),
os-time-for-svc-send-handler-to-dispatcher (*os*)),
os-time-for-svc-send-handler-to-wait-state (*os*))

EVENT: Disable rewrite-svc-send-trace3.

THEOREM: trace-svc-send-handler-path1

(good-os (*os*)
 \wedge (\neg tm-waiting (*os*)
 \wedge (((tm-svcid (*os*) **mod** '4) = '0)
 \wedge (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os))
 \wedge array-qempty (array-deq (os-readyq (*os*)))))))))
 \rightarrow (tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-path1 (*os*))
= os-svc-send-handler-path1 (*os*))

EVENT: Disable trace-svc-send-handler-path1.

DEFINITION:

os-svc-send-time3 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-svc-send-time4 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-svc-send-handler-path2 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick)

THEOREM: rewrite-svc-send-trace4

tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-path2 (*os*))
= tm-processor (tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-thru-save-state (*os*)),
os-svc-send-time3 (*os*))

EVENT: Disable rewrite-svc-send-trace4.

THEOREM: rewrite-svc-send-trace5

tm-processor (os-svc-send-handler-thru-save-state (*os*), os-svc-send-time3 (*os*))
= tm-processor (tm-processor (os-svc-send-handler-thru-save-state (*os*),
os-time-for-svc-send-handler-to-qfullp-test (*os*)),
os-svc-send-time4 (*os*))

EVENT: Disable rewrite-svc-send-trace5.

THEOREM: rewrite-svc-send-trace6

tm-processor (os-svc-send-handler-to-qfullp-test (*os*), os-svc-send-time4 (*os*))
= tm-processor (tm-processor (os-svc-send-handler-to-qfullp-test (*os*),
os-time-for-svc-send-handler-to-dispatcher (*os*)),
os-time-for-svc-send-handler-to-lpsw (*os*))

EVENT: Disable rewrite-svc-send-trace6.

THEOREM: trace-svc-send-handler-path2

(good-os (*os*)
∧ (¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '0)
∧ (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),

$$\begin{aligned} & \text{os-destid}(os, '16), \\ & os) \\ \rightarrow & \quad \wedge (\neg \text{array-qemptyp}(\text{array-deq}(\text{os-readyq}(os)))))) \\ & (\text{tm-processor}(\text{os-intended-svc-send-interrupt}(os), \\ & \quad \text{os-time-for-svc-send-handler-path2}(os)) \\ = & \quad \text{os-svc-send-handler-path2}(os) \end{aligned}$$

EVENT: Disable trace-svc-send-handler-path2.

DEFINITION:

os-svc-send-time5(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

DEFINITION:

os-svc-send-time6(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

DEFINITION:

os-svc-send-time7(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-svc-send-handler-path3(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

```

THEOREM: rewrite-svc-send-trace7

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-svc-send-interrupt}(os), \\
& \quad \text{os-time-for-svc-send-handler-path3}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-svc-send-interrupt}(os), \\
& \quad \text{os-time-for-svc-send-handler-thru-save-state}(os)), \\
& \quad \text{os-svc-send-time5}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-send-trace7.

THEOREM: rewrite-svc-send-trace8

$$\begin{aligned}
& \text{tm-processor}(\text{os-svc-send-handler-thru-save-state}(os), \text{os-svc-send-time5}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-svc-send-handler-thru-save-state}(os), \\
& \quad \text{os-time-for-svc-send-handler-to-qfullp-test}(os)), \\
& \quad \text{os-svc-send-time6}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-send-trace8.

THEOREM: rewrite-svc-send-trace9

$$\begin{aligned}
& \text{tm-processor}(\text{os-svc-send-handler-to-qfullp-test}(os), \text{os-svc-send-time6}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-svc-send-handler-to-qfullp-test}(os), \\
& \quad \text{os-time-for-svc-send-handler-thru-enqueue}(os)), \\
& \quad \text{os-svc-send-time7}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-send-trace9.

THEOREM: rewrite-svc-send-trace10

$$\begin{aligned}
& \text{tm-processor}(\text{os-svc-send-handler-thru-enqueue}(os), \text{os-svc-send-time7}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-svc-send-handler-thru-enqueue}(os), \\
& \quad \text{os-time-for-svc-send-handler-to-svc-resume-task1}(os)), \\
& \quad \text{os-time-for-svc-send-handler-to-svcr}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-send-trace10.

THEOREM: trace-svc-send-handler-path3

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (((tm-svcid (os) mod '4) = '0)
        ^ ((¬ array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                                    os-destid (os,
                                                    '16),
                                                    os)))
           ^ ((getnth ('2 * os-destid (os, '16),
                       os-status-table (os))
              = '3)
              ^ (getnth ('1
                          + ('2
                            * os-destid (os,
                            '16)),
                          os-status-table (os))
                 = os-current-taskid (os))))))))
→ (tm-processor (os-intended-svc-send-interrupt (os),
               os-time-for-svc-send-handler-path3 (os))
   = os-svc-send-handler-path3 (os))

```

EVENT: Disable trace-svc-send-handler-path3.

DEFINITION:

```

os-svc-send-time8 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick)

```

DEFINITION:

```

os-svc-send-time9 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick)

```

DEFINITION:

```

os-svc-send-time10 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick)

```

DEFINITION:

```

os-time-for-svc-send-handler-path4a (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick)

```

THEOREM: rewrite-svc-send-trace11

```

tm-processor (os-intended-svc-send-interrupt (os),
              os-time-for-svc-send-handler-path4a (os))
= tm-processor (tm-processor (os-intended-svc-send-interrupt (os),
                             os-time-for-svc-send-handler-thru-save-state (os)),
                os-svc-send-time8 (os))

```

EVENT: Disable rewrite-svc-send-trace11.

THEOREM: rewrite-svc-send-trace12

```

tm-processor (os-svc-send-handler-thru-save-state (os), os-svc-send-time8 (os))
= tm-processor (tm-processor (os-svc-send-handler-thru-save-state (os),
                             os-time-for-svc-send-handler-to-qfullp-test (os)),
                os-svc-send-time9 (os))

```

EVENT: Disable rewrite-svc-send-trace12.

THEOREM: rewrite-svc-send-trace13

```

tm-processor (os-svc-send-handler-to-qfullp-test (os), os-svc-send-time9 (os))
= tm-processor (tm-processor (os-svc-send-handler-to-qfullp-test (os),
                             os-time-for-svc-send-handler-thru-enqueue (os)),
                os-svc-send-time10 (os))

```

EVENT: Disable rewrite-svc-send-trace13.

THEOREM: rewrite-svc-send-trace14

```

tm-processor (os-svc-send-handler-thru-enqueue (os), os-svc-send-time10 (os))

```

= tm-processor (tm-processor (os-svc-send-handler-thru-enqueue (os),
os-time-for-svc-send-handler-to-svc-resume-task2 (os)),
os-time-for-svc-send-handler-to-svcr (os))

EVENT: Disable rewrite-svc-send-trace14.

THEOREM: trace-svc-send-handler-path4a

(good-os (os)
 \wedge (\neg tm-waiting (os))
 \wedge (((tm-svcid (os) **mod** '4) = '0)
 \wedge (\neg array-qfullp (os-current-mbuffer (os-current-taskid (os),
os-destid (os,
'16),
os)))
 \wedge (getnth ('2 * os-destid (os, '16),
os-status-table (os))
 \neq '3))))))
 \rightarrow (tm-processor (os-intended-svc-send-interrupt (os),
os-time-for-svc-send-handler-path4a (os))
= os-svc-send-handler-path4 (os))

EVENT: Disable trace-svc-send-handler-path4a.

DEFINITION:

os-svc-send-time11 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-svc-send-time12 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-svc-send-time13 (os)
= '(tick tick tick tick tick tick tick tick tick tick

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-svc-send-handler-path4b (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)

```

THEOREM: rewrite-svc-send-trace15

```

tm-processor (os-intended-svc-send-interrupt (os),
              os-time-for-svc-send-handler-path4b (os))
= tm-processor (tm-processor (os-intended-svc-send-interrupt (os),
                             os-time-for-svc-send-handler-thru-save-state (os)),
               os-svc-send-time11 (os))

```

EVENT: Disable rewrite-svc-send-trace15.

THEOREM: rewrite-svc-send-trace16

```

tm-processor (os-svc-send-handler-thru-save-state (os), os-svc-send-time11 (os))
= tm-processor (tm-processor (os-svc-send-handler-thru-save-state (os),
                             os-time-for-svc-send-handler-to-qfullp-test (os)),
               os-svc-send-time12 (os))

```

EVENT: Disable rewrite-svc-send-trace16.

THEOREM: rewrite-svc-send-trace17

```

tm-processor (os-svc-send-handler-to-qfullp-test (os), os-svc-send-time12 (os))
= tm-processor (tm-processor (os-svc-send-handler-to-qfullp-test (os),
                             os-time-for-svc-send-handler-thru-enqueue (os)),
               os-svc-send-time13 (os))

```

EVENT: Disable rewrite-svc-send-trace17.

THEOREM: rewrite-svc-send-trace18

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-send-handler-thru-enqueue}(os), \text{os-svc-send-time13}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-send-handler-thru-enqueue}(os), \\ & \quad \text{os-time-for-svc-send-handler-to-svc-resume-task3}(os), \\ & \quad \text{os-time-for-svc-send-handler-to-svcr}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-send-trace18.

THEOREM: trace-svc-send-handler-path4b

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \\ & \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\ & \quad \quad \quad \text{os-destid}(os, \\ & \quad \quad \quad \quad '16), \\ & \quad \quad \quad \quad \quad os))) \\ & \quad \quad \wedge ((\text{getnth}('2 * \text{os-destid}(os, '16), \\ & \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad \quad = '3) \\ & \quad \quad \wedge (\text{getnth}('1 \\ & \quad \quad \quad \quad + ('2 \\ & \quad \quad \quad \quad \quad * \text{os-destid}(os, \\ & \quad \quad \quad \quad \quad \quad '16)), \\ & \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad \quad \neq \text{os-current-taskid}(os)))))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-send-interrupt}(os), \\ & \quad \text{os-time-for-svc-send-handler-path4b}(os)) \\ = & \text{os-svc-send-handler-path4}(os) \end{aligned}$$

EVENT: Disable trace-svc-send-handler-path4b.

DEFINITION:

$$\begin{aligned} & \text{os-time-for-svc-send-handler-path4}(os) \\ = & \text{if } \text{getnth}('2 * \text{os-destid}(os, '16), \text{os-status-table}(os)) \neq '3 \\ & \quad \text{then } \text{os-time-for-svc-send-handler-path4a}(os) \\ & \quad \text{else } \text{os-time-for-svc-send-handler-path4b}(os) \text{ endif} \end{aligned}$$

THEOREM: trace-svc-send-handler-path4

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \\ & \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\ & \quad \quad \quad \text{os-destid}(os, \\ & \quad \quad \quad \quad '16), \\ & \quad \quad \quad \quad \quad os))) \end{aligned}$$

$$\begin{aligned}
& \wedge (\neg ((\text{getnth}('2 * \text{os-destid}(os, '16), \\
& \quad \text{os-status-table}(os)) \\
& \quad = '3) \\
& \wedge (\text{getnth}('1 \\
& \quad + ('2 \\
& \quad \quad * \text{os-destid}(os, \\
& \quad \quad \quad '16)), \\
& \quad \quad \text{os-status-table}(os)) \\
& \quad = \text{os-current-taskid}(os)))))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-svc-send-interrupt}(os), \\
& \quad \text{os-time-for-svc-send-handler-path4}(os)) \\
& = \text{os-svc-send-handler-path4}(os))
\end{aligned}$$

EVENT: Disable trace-svc-send-handler-path4.

DEFINITION:

```

os-time-for-svc-send-handler (os)
=  if array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                         os-destid (os, '16),
                                         os))
    then if array-qempty (array-deq (os-readyq (os)))
          then os-time-for-svc-send-handler-path1 (os)
          else os-time-for-svc-send-handler-path2 (os) endif
    elseif (getnth ('2 * os-destid (os, '16), os-status-table (os))
              = '3)
          ^ (getnth ('1 + ('2 * os-destid (os, '16)),
                    os-status-table (os))
              = os-current-taskid (os))
    then os-time-for-svc-send-handler-path3 (os)
    else os-time-for-svc-send-handler-path4 (os) endif

```

DEFINITION:

```

os-svc-send-handler (os)
=  if array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                         os-destid (os, '16),
                                         os))
    then if array-qempty (array-deq (os-readyq (os)))
          then os-svc-send-handler-path1 (os)
          else os-svc-send-handler-path2 (os) endif
    elseif (getnth ('2 * os-destid (os, '16), os-status-table (os))
              = '3)
          ^ (getnth ('1 + ('2 * os-destid (os, '16)),
                    os-status-table (os))
              = os-current-taskid (os))

```

then os-svc-send-handler-path3 (*os*)
else os-svc-send-handler-path4 (*os*) **endif**

THEOREM: trace-svc-send-handler
(good-os (*os*) \wedge (\neg tm-waiting (*os*)) \wedge ((tm-svcid (*os*) **mod** '4) = '0)))
 \rightarrow (tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler (*os*))
= os-svc-send-handler (*os*))

EVENT: Disable trace-svc-send-handler.

DEFINITION:
os-intended-svc-receive-interrupt (*tm*)
= os-intended-svc-send-interrupt (*tm*)

THEOREM: equivalence-of-svc-receive-interrupt-expressions
(good-os (*os*) \wedge (\neg tm-waiting (*os*)) \wedge (\neg tm-errorp (*os*)))
 \rightarrow (tm-execute-svc-interrupt (*os*)
= os-intended-svc-receive-interrupt (*os*))

EVENT: Disable equivalence-of-svc-receive-interrupt-expressions.

DEFINITION:
os-time-for-svc-receive-handler-thru-save-state (*os*)
= os-time-for-svc-send-handler-thru-save-state (*os*)

DEFINITION:
os-svc-receive-handler-thru-save-state (*os*)
= os-svc-send-handler-thru-save-state (*os*)

THEOREM: trace-svc-receive-handler-thru-save-state
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-thru-save-state (*os*))
= os-svc-receive-handler-thru-save-state (*os*))

EVENT: Disable trace-svc-receive-handler-thru-save-state.

DEFINITION:
os-time-for-svc-receive-handler-to-qempty-test (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

```

(DEFN OS-SVC-RECEIVE-HANDLER-TO-QEMPTYP-TEST
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '1
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS OS)
'499
(PUTSEG (OS-CODE OS)
'2547

```

```

(PUTNTH '2558
  '3309
    (PUTNTH '2824
      '3310
        (TM-MEMORY OS))))))))))))))
(PUTNTH '2845
  '0
    (PUTNTH '3310
      '1
        (PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
          '2
            (PUTNTH (PLUS '499
              (PLUS (TIMES '8
                (ARRAY-QFIRST (OS-READYQ OS)))
                (TIMES '128
                  (REMAINDER (TM-R2 OS) '16))))
          '3
            (PUTNTH (TIMES '8
              (ARRAY-QFIRST (OS-READYQ OS)))
            '4
              (PUTNTH (REMAINDER (TM-R2 OS) '16)
                '6
                  (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS (TIMES '8
  (ARRAY-QFIRST (OS-READYQ OS)))
  (TIMES '128
    (REMAINDER (GETNTH '2 (TM-REGS OS)
      '16))))
  '499))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS)
NIL)

```

THEOREM: trace-svc-receive-handler-to-qempty-test
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '1)))$

```

→ (tm-processor (os-svc-receive-handler-thru-save-state (os),
                os-time-for-svc-receive-handler-to-qempty-test (os))
    = os-svc-receive-handler-to-qempty-test (os))

```

EVENT: Disable trace-svc-receive-handler-to-qempty-test.

DEFINITION:

```

os-time-for-svc-receive-handler-to-dispatcher (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-TO-DISPATCHER
  (OS)
  (TM
    (PUTNTH
      (TM-R0 OS)
      '0
    (PUTNTH
      (TM-R1 OS)
      '1
    (PUTNTH
      (TM-PACK-PSW (TM-CC OS)
                   (TM-ERROR OS)
                   (TM-SVCFLAG OS)
                   (TM-SVCID OS))
      '2
    (PUTNTH '1
      '8
    (PUTNTH
      (ARRAY-QFIRST (OS-READYQ OS))
      '12
    (PUTNTH
      (TM-R2 OS)
      '13
    (PUTNTH
      (TM-R3 OS)
      '14
    (PUTSEG
      (OS-NEW-TASK-TABLE OS)
      '15
    (PUTSEG
      (OS-SEGMENT-TABLE OS)

```

```

'159
(PUTSEG
  (ARRAY-DEQ (OS-READYQ OS))
  '191
  (PUTSEG
    (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
      '3
      (OS-SRCID OS '16)
      OS)
    '211
    (PUTSEG (OS-MBUFFERS OS)
      '499
      (PUTSEG (OS-CODE OS)
        '2547
        (PUTNTH '2558
          '3309
          (PUTNTH '2905
            '3310
            (TM-MEMORY OS)))))))))))))
(PUTNTH '3206
  '0
  (PUTNTH '3310
    '1
    (PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
      '2
      (PUTNTH (PLUS '211
        (TIMES '2
          (ARRAY-QFIRST (OS-READYQ OS))))
        '3
        (PUTNTH (TIMES '8
          (ARRAY-QFIRST (OS-READYQ OS)))
          '4
          (PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
            '16)
            '6
            (TM-REGS OS))))))
      (TM-CC-VALUE (TM-ALU-PLUS (TIMES '2
        (ARRAY-QFIRST (OS-READYQ OS)))
        '211))
      '0
      (TM-SVCFLAG OS)
      (TM-SVCID OS)
      (TM-BASE OS)
      (TM-LIMIT OS)

```

```

(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-dispatcher

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (((tm-svcid (os) mod '4) = '1)
        ^ array-qempty (os-current-mbuffer (os-srcid (os, '16),
                                             os-current-taskid (os),
                                             os))))))
→ (tm-processor (os-svc-receive-handler-to-qempty-test (os),
                os-time-for-svc-receive-handler-to-dispatcher (os))
   = os-svc-receive-handler-to-dispatcher (os))

```

EVENT: Disable trace-svc-receive-handler-to-dispatcher.

DEFINITION:

```

os-time-for-svc-receive-handler-to-wait-state (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '1
'8

```

```

(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
  '12
(PUTNTH
  (TM-R2 OS)
  '13
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (ARRAY-DEQ (OS-READYQ OS))
  '191
(PUTSEG
  (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
    '3
    (OS-SRCID OS '16)
    OS)
  '211
  (PUTSEG (OS-MBUFFERS OS)
    '499
    (PUTSEG (OS-CODE OS)
      '2547
      (PUTNTH '2558
        '3309
        (PUTNTH '3211
          '3310
          (TM-MEMORY OS))))))))))))))
(PUTNTH '3222
  '0
  (PUTNTH '3310
    '1
    (PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
      '2
      (PUTNTH '191
        '3
        (PUTNTH (TIMES '8
          (ARRAY-QFIRST (OS-READYQ OS)))
          '4

```



```

(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
                  '16)
        '6
        (TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-wait-state

```

(good-os (os)
  ^ ((¬ tm-waiting (os)) ∧ array-qempty (array-deq (os-readyq (os))))))
→ (tm-processor (os-svc-receive-handler-to-dispatcher (os),
                os-time-for-svc-receive-handler-to-wait-state (os))
   = os-svc-receive-handler-path1 (os))

```

EVENT: Disable trace-svc-receive-handler-to-wait-state.

DEFINITION:

```

os-time-for-svc-receive-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-PATH2
  (OS)
  (TM
   (PUTNTH
    (GETNTH (TIMES '9
              (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
            (OS-NEW-TASK-TABLE OS))
    '0

```

```

(PUTNTH
  (GETNTH (PLUS '1
            (TIMES '9
              (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))))
    (OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
  (GETNTH (PLUS '8
            (TIMES '9
              (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))))
    (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '1
  '8
  (PUTNTH
    (ARRAY-QFIRST (OS-READYQ OS))
    '12
    (PUTNTH
      (TM-R2 OS)
      '13
      (PUTNTH
        (TM-R3 OS)
        '14
        (PUTSEG
          (OS-NEW-TASK-TABLE OS)
          '15
          (PUTSEG
            (OS-SEGMENT-TABLE OS)
            '159
            (PUTSEG
              (ARRAY-DEQ (OS-READYQ OS))
              '191
              (PUTSEG
                (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
                '3
                (OS-SRCID OS '16)
                OS)
                '211
                (PUTSEG (OS-MBUFFERS OS)
                '499
                (PUTSEG (OS-CODE OS)
                '2547
                (PUTNTH '2558
                '3309

```

```

(PUTNTH '3217
      '3310
      (TM-MEMORY OS))))))))))))))
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-BASE (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-LIMIT (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-lpsw

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (¬ array-qempty (array-deq (os-readyq (os)))))
→ (tm-processor (os-svc-receive-handler-to-dispatcher (os),
                os-time-for-svc-receive-handler-to-lpsw (os))
   = os-svc-receive-handler-path2 (os))

```

EVENT: Disable trace-svc-receive-handler-to-lpsw.

DEFINITION:

```

os-time-for-svc-receive-handler-thru-delivery (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-THRU-DELIVERY
  (OS)

```

```

(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '1
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2853
'3310

```

```

(TM-MEMORY OS))))))))))))))
(PUTNTH '2865
'0
(PUTNTH '3310
'1
(PUTNTH
  (ARRAY-QFIRST (GETNTH (PLUS (ARRAY-QFIRST (OS-READYQ OS))
    (TIMES '16
      (REMAINDER (GETNTH '2 (TM-REGS OS))
        '16)))
    (TABLE '8 (OS-MBUFFERS OS))))))
'2
(PUTNTH (PLUS '15
  (TIMES '9
    (ARRAY-QFIRST (OS-READYQ OS))))))
'3
(PUTNTH (TIMES '8
  (ARRAY-QFIRST (OS-READYQ OS))))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
  '16)
  '6
  (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '9
  (ARRAY-QFIRST (OS-READYQ OS)))
  '15))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-thru-delivery

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
    ^ (((tm-svcid (os) mod '4) = '1)
      ^ (¬ array-qemptyp (os-current-mbuffer (os-srcid (os),

```

```

                                '16),
                                os-current-taskid (os),
                                os))))))
→ (tm-processor (os-svc-receive-handler-to-qempty-test (os),
                os-time-for-svc-receive-handler-thru-delivery (os))
    = os-svc-receive-handler-thru-delivery (os))

```

EVENT: Disable trace-svc-receive-handler-thru-delivery.

DEFINITION:

```

os-time-for-svc-receive-handler-to-svc-resume-task1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-TO-SVC-RESUME-TASK1
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
              (TM-ERROR OS)
              (TM-SVCFLAG OS)
              (TM-SVCID OS))
'2
(PUTNTH '1
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)

```

```

'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (OS-SRCID OS '16)
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (OS-SRCID OS '16)
'0
'0
OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
'499
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2898
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'2
(PUTNTH '191
'3
(PUTNTH (PLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
'16)
'6
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
'16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)

```

```

(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-svc-resume-task1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
     ∧ ((getnth ('2 * os-srcid (os, '16), os-status-table (os))
                = '2)
        ∧ (getnth ('1 + ('2 * os-srcid (os, '16)),
                    os-status-table (os))
              = os-current-taskid (os))))))
→ (tm-processor (os-svc-receive-handler-thru-delivery (os),
                 os-time-for-svc-receive-handler-to-svc-resume-task1 (os))
    = os-svc-receive-handler-to-svc-resume-task1 (os))

```

EVENT: Disable trace-svc-receive-handler-to-svc-resume-task1.

DEFINITION:

```

os-time-for-svc-receive-handler-to-svc-resume-task2 (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-SVC-RECEIVE-HANDLER-TO-SVC-RESUME-TASK2
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))

```



```

'2
(PUTNTH '1
  '8
  (PUTNTH
    (ARRAY-QFIRST (OS-READYQ OS))
    '12
    (PUTNTH
      (TM-R2 OS)
      '13
      (PUTNTH
        (TM-R3 OS)
        '14
        (PUTSEG
          (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)
          '15
          (PUTSEG
            (OS-SEGMENT-TABLE OS)
            '159
            (PUTSEG
              (OS-READYQ OS)
              '191
              (PUTSEG
                (OS-STATUS-TABLE OS)
                '211
                (PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
                  '499
                  (PUTSEG (OS-CODE OS)
                    '2547
                    (PUTNTH '2558
                      '3309
                      (PUTNTH '2853
                        '3310
                        (TM-MEMORY OS))))))))))))))
(PUTNTH '3222
  '0
  (PUTNTH '3310
    '1
    (PUTNTH
      (ARRAY-QFIRST (GETNTH (PLUS (ARRAY-QFIRST (OS-READYQ OS))
        (TIMES '16
          (REMAINDER (GETNTH '2 (TM-REGS OS))
            '16)))
        (TABLE '8 (OS-MBUFFERS OS))))
      '2

```

```

(PUTNTH (PLUS '211
          (TIMES '2
              (REMAINDER (GETNTH '2 (TM-REGS OS))
                          '16)))
        '3
  (PUTNTH (TIMES '8
          (ARRAY-QFIRST (OS-READYQ OS)))
        '4
    (PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
                      '16)
            '6
      (TM-REGS OS))))))
(TM-CC-VALUE
  (TM-ALU-DIFFERENCE (GETNTH (TIMES '2
                              (REMAINDER (GETNTH '2 (TM-REGS OS))
                                          '16))
                            (OS-STATUS-TABLE OS))
                    '2))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-svc-resume-task2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{getnth } ('2 * \text{os-srcid } (os), '16), \text{os-status-table } (os)) \\
& \quad \quad \neq '2))) \\
\rightarrow & (\text{tm-processor } (\text{os-svc-receive-handler-thru-delivery } (os), \\
& \quad \quad \quad \text{os-time-for-svc-receive-handler-to-svc-resume-task2 } (os)) \\
& \quad = \text{os-svc-receive-handler-to-svc-resume-task2 } (os))
\end{aligned}$$

EVENT: Disable trace-svc-receive-handler-to-svc-resume-task2.

DEFINITION:

os-time-for-svc-receive-handler-to-svc-resume-task3 (*os*)

```
= '(tick tick tick tick tick tick tick)
```

```
(DEFN OS-SVC-RECEIVE-HANDLER-TO-SVC-RESUME-TASK3
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '1
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
'499
```

```

(PUTSEG (OS-CODE OS)
        '2547
        (PUTNTH '2558
          '3309
          (PUTNTH '2853
            '3310
            (TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
  (ARRAY-QFIRST (GETNTH (PLUS (ARRAY-QFIRST (OS-READYQ OS))
                              (TIMES '16
                                (REMAINDER (GETNTH '2 (TM-REGS OS))
                                              '16))))
                (TABLE '8 (OS-MBUFFERS OS))))
'2
(PUTNTH (PLUS '211
            (TIMES '2
              (REMAINDER (GETNTH '2 (TM-REGS OS))
                          '16)))
'3
(PUTNTH (TIMES '8
        (ARRAY-QFIRST (OS-READYQ OS)))
'4
(PUTNTH (REMAINDER (GETNTH '2 (TM-REGS OS))
                  '16)
'6
        (TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE (GETNTH (PLUS '1
                              (TIMES '2
                                (REMAINDER (GETNTH '2 (TM-REGS OS))
                                              '16)))
                          (OS-STATUS-TABLE OS))
  (ARRAY-QFIRST (OS-READYQ OS))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1

```

```

'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-svc-resume-task3

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ ((getnth ('2 * os-srcid (os, '16), os-status-table (os))
                = '2)
        ^ (getnth ('1 + ('2 * os-srcid (os, '16)),
                   os-status-table (os))
            ≠ os-current-taskid (os))))))
→ (tm-processor (os-svc-receive-handler-thru-delivery (os),
                 os-time-for-svc-receive-handler-to-svc-resume-task3 (os))
    = os-svc-receive-handler-to-svc-resume-task3 (os))

```

EVENT: Disable trace-svc-receive-handler-to-svc-resume-task3.

DEFINITION:

```

os-time-for-svc-receive-handler-to-svcr (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick)

```

(DEFN OS-SVC-RECEIVE-HANDLER-PATH3

```

(OS)
(TM
 (PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
          (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
 '0
 (PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
          (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
 '1
 (PUTNTH
  (GETNTH (PLUS '8

```

```

(TIMES '9
  (ARRAY-QFIRST (OS-READYQ OS)))
(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
'2
(PUTNTH '1
  '8
  (PUTNTH
    (ARRAY-QFIRST (OS-READYQ OS))
    '12
    (PUTNTH
      (TM-R2 OS)
      '13
      (PUTNTH
        (TM-R3 OS)
        '14
        (PUTSEG
          (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)
          '15
          (PUTSEG
            (OS-SEGMENT-TABLE OS)
            '159
            (PUTSEG
              (ARRAY-ENQ (OS-SRCID OS '16)
                (OS-READYQ OS))
              '191
              (PUTSEG
                (OS-UPDATE-STATUS (OS-SRCID OS '16)
                  '0
                  '0
                  OS)
                '211
                (PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
                  '499
                  (PUTSEG (OS-CODE OS)
                    '2547
                    (PUTNTH '2558
                      '3309
                      (PUTNTH '3229
                        '3310
                        (TM-MEMORY OS))))))))))))))
(PUTNTH (ARRAY-QFIRST (OS-CURRENT-MBUFFER (OS-SRCID OS '16)
  (OS-CURRENT-TASKID OS)
  OS))
'3

```

```

(TM-REGS OS))
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-svcr1

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ ((getnth ('2 * os-srcid (os, '16), os-status-table (os))
                = '2)
        ^ (getnth ('1 + ('2 * os-srcid (os, '16)),
                   os-status-table (os))
            = os-current-taskid (os))))))
→ (tm-processor (os-svc-receive-handler-to-svc-resume-task1 (os),
                  os-time-for-svc-receive-handler-to-svcr (os))
    = os-svc-receive-handler-path3 (os))

```

EVENT: Disable trace-svc-receive-handler-to-svcr1.

```

(DEFN
OS-SVC-RECEIVE-HANDLER-PATH4
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
          (TIMES '9
              (ARRAY-QFIRST (OS-READYQ OS))))))

```

```

(OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
'1
(PUTNTH
  (GETNTH (PLUS '8
    (TIMES '9
      (ARRAY-QFIRST (OS-READYQ OS))))
    (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS))
'2
(PUTNTH '1
  '8
  (PUTNTH
    (ARRAY-QFIRST (OS-READYQ OS))
  '12
  (PUTNTH
    (TM-R2 OS)
  '13
  (PUTNTH
    (TM-R3 OS)
  '14
  (PUTSEG
    (OS-TASK-TABLE-WITH-DELIVERED-MESSAGE OS)
  '15
  (PUTSEG
    (OS-SEGMENT-TABLE OS)
  '159
  (PUTSEG
    (OS-READYQ OS)
  '191
  (PUTSEG
    (OS-STATUS-TABLE OS)
  '211
  (PUTSEG (OS-MBUFFERS-WITH-DEQUEUED-MESSAGE OS)
    '499
    (PUTSEG (OS-CODE OS)
      '2547
      (PUTNTH '2558
        '3309
        (PUTNTH '3229
          '3310
          (TM-MEMORY OS))))))))))))))
(PUTNTH (ARRAY-QFIRST (OS-CURRENT-MBUFFER (OS-SRCID OS '16)
  (OS-CURRENT-TASKID OS)
  OS))
'3

```



```

(TM-REGS OS))
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-receive-handler-to-svcr2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-receive-handler-to-svc-resume-task2}(os),$
 $\text{os-time-for-svc-receive-handler-to-svcr}(os))$
 $= \text{os-svc-receive-handler-path4}(os))$

EVENT: Disable trace-svc-receive-handler-to-svcr2.

THEOREM: trace-svc-receive-handler-to-svcr3
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-receive-handler-to-svc-resume-task3}(os),$
 $\text{os-time-for-svc-receive-handler-to-svcr}(os))$
 $= \text{os-svc-receive-handler-path4}(os))$

EVENT: Disable trace-svc-receive-handler-to-svcr3.

DEFINITION:
 $\text{os-svc-receive-time1}(os)$
 $= \text{'(tick tick)}$

DEFINITION:
 $\text{os-svc-receive-time2}(os)$
 $= \text{'(tick tick)}$

DEFINITION:

os-time-for-svc-receive-handler-path1 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

THEOREM: rewrite-svc-receive-trace1

tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-path1 (*os*))
= tm-processor (tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-thru-save-state (*os*)),
os-svc-receive-time1 (*os*))

EVENT: Disable rewrite-svc-receive-trace1.

THEOREM: rewrite-svc-receive-trace2

tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-svc-receive-time1 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-time-for-svc-receive-handler-to-qempty-test (*os*)),
os-svc-receive-time2 (*os*))

EVENT: Disable rewrite-svc-receive-trace2.

THEOREM: rewrite-svc-receive-trace3

tm-processor (os-svc-receive-handler-to-qempty-test (*os*),
os-svc-receive-time2 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qempty-test (*os*),
os-time-for-svc-receive-handler-to-dispatcher (*os*)),
os-time-for-svc-receive-handler-to-wait-state (*os*))

EVENT: Disable rewrite-svc-receive-trace3.

THEOREM: trace-svc-receive-handler-path1

(good-os (*os*)
∧ (¬ tm-waiting (*os*)
∧ (((tm-svcid (*os*) **mod** '4) = '1)
∧ (array-qempty (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),

$$\begin{aligned}
& \rightarrow (\text{tm-processor} (\text{os-intended-svc-receive-interrupt} (os), \\
& \quad \text{os-time-for-svc-receive-handler-path1} (os)) \\
& = \text{os-svc-receive-handler-path1} (os))
\end{aligned}$$

EVENT: Disable trace-svc-receive-handler-path1.

DEFINITION:

$$\begin{aligned}
& \text{os-svc-receive-time3} (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-svc-receive-time4} (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-svc-receive-handler-path2} (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick)
\end{aligned}$$

THEOREM: rewrite-svc-receive-trace4

$$\begin{aligned}
& \text{tm-processor} (\text{os-intended-svc-receive-interrupt} (os), \\
& \quad \text{os-time-for-svc-receive-handler-path2} (os)) \\
& = \text{tm-processor} (\text{tm-processor} (\text{os-intended-svc-receive-interrupt} (os), \\
& \quad \text{os-time-for-svc-receive-handler-thru-save-state} (os)), \\
& \quad \text{os-svc-receive-time3} (os))
\end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace4.

THEOREM: rewrite-svc-receive-trace5
tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-svc-receive-time3 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-time-for-svc-receive-handler-to-qemptyyp-test (*os*)),
os-svc-receive-time4 (*os*))

EVENT: Disable rewrite-svc-receive-trace5.

THEOREM: rewrite-svc-receive-trace6
tm-processor (os-svc-receive-handler-to-qemptyyp-test (*os*),
os-svc-receive-time4 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyyp-test (*os*),
os-time-for-svc-receive-handler-to-dispatcher (*os*)),
os-time-for-svc-receive-handler-to-lpsw (*os*))

EVENT: Disable rewrite-svc-receive-trace6.

THEOREM: trace-svc-receive-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '1)
∧ (array-qemptyyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os))
∧ (¬ array-qemptyyp (array-deq (os-readyq (*os*))))))))))
→ (tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-path2 (*os*))
= os-svc-receive-handler-path2 (*os*))

EVENT: Disable trace-svc-receive-handler-path2.

DEFINITION:
os-svc-receive-time5 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

```

DEFINITION:

```

os-svc-receive-time6 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick)

```

DEFINITION:

```

os-svc-receive-time7 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-svc-receive-handler-path3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

THEOREM: rewrite-svc-receive-trace7

```

tm-processor (os-intended-svc-receive-interrupt (os),
             os-time-for-svc-receive-handler-path3 (os))
= tm-processor (tm-processor (os-intended-svc-receive-interrupt (os),
                           os-time-for-svc-receive-handler-thru-save-state (os)),
               os-svc-receive-time5 (os))

```

EVENT: Disable rewrite-svc-receive-trace7.

THEOREM: rewrite-svc-receive-trace8

$$\begin{aligned}
& \text{tm-processor} (\text{os-svc-receive-handler-thru-save-state} (os), \\
& \quad \text{os-svc-receive-time5} (os)) \\
= & \text{tm-processor} (\text{tm-processor} (\text{os-svc-receive-handler-thru-save-state} (os), \\
& \quad \text{os-time-for-svc-receive-handler-to-qempty-test} (os)), \\
& \quad \text{os-svc-receive-time6} (os))
\end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace8.

THEOREM: rewrite-svc-receive-trace9

$$\begin{aligned}
& \text{tm-processor} (\text{os-svc-receive-handler-to-qempty-test} (os), \\
& \quad \text{os-svc-receive-time6} (os)) \\
= & \text{tm-processor} (\text{tm-processor} (\text{os-svc-receive-handler-to-qempty-test} (os), \\
& \quad \text{os-time-for-svc-receive-handler-thru-delivery} (os)), \\
& \quad \text{os-svc-receive-time7} (os))
\end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace9.

THEOREM: rewrite-svc-receive-trace10

$$\begin{aligned}
& \text{tm-processor} (\text{os-svc-receive-handler-thru-delivery} (os), \\
& \quad \text{os-svc-receive-time7} (os)) \\
= & \text{tm-processor} (\text{tm-processor} (\text{os-svc-receive-handler-thru-delivery} (os), \\
& \quad \text{os-time-for-svc-receive-handler-to-svc-resume-task1} (os)), \\
& \quad \text{os-time-for-svc-receive-handler-to-svcr} (os))
\end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace10.

THEOREM: trace-svc-receive-handler-path3

$$\begin{aligned}
& (\text{good-os} (os) \\
& \quad \wedge ((\neg \text{tm-waiting} (os)) \\
& \quad \quad \wedge (((\text{tm-srcid} (os) \bmod '4) = '1) \\
& \quad \quad \quad \wedge ((\neg \text{array-qempty} (\text{os-current-mbuffer} (\text{os-srcid} (os), \\
& \quad \quad \quad \quad \quad \quad \quad '16), \\
& \quad \quad \quad \quad \quad \quad \quad \text{os-current-taskid} (os), \\
& \quad \quad \quad \quad \quad \quad \quad os))) \\
& \quad \quad \quad \wedge ((\text{getnth} ('2 * \text{os-srcid} (os, '16), \\
& \quad \quad \quad \quad \quad \quad \quad \text{os-status-table} (os)) \\
& \quad \quad \quad \quad = '2) \\
& \quad \quad \quad \wedge (\text{getnth} ('1 \\
& \quad \quad \quad \quad \quad \quad \quad + ('2 \\
& \quad \quad \quad \quad \quad \quad \quad \quad * \text{os-srcid} (os, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad '16)), \\
& \quad \quad \quad \quad \quad \quad \quad \text{os-status-table} (os))
\end{aligned}$$

```

= os-current-taskid (os))))))
→ (tm-processor (os-intended-svc-receive-interrupt (os),
              os-time-for-svc-receive-handler-path3 (os))
    = os-svc-receive-handler-path3 (os))

```

EVENT: Disable trace-svc-receive-handler-path3.

DEFINITION:

```

os-svc-receive-time8 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

DEFINITION:

```

os-svc-receive-time9 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick)

```

DEFINITION:

```

os-svc-receive-time10 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

DEFINITION:

```

os-time-for-svc-receive-handler-path4a (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

```

THEOREM: rewrite-svc-receive-trace11

```

tm-processor (os-intended-svc-receive-interrupt (os),
              os-time-for-svc-receive-handler-path4a (os))
= tm-processor (tm-processor (os-intended-svc-receive-interrupt (os),
                             os-time-for-svc-receive-handler-thru-save-state (os)),
               os-svc-receive-time8 (os))

```

EVENT: Disable rewrite-svc-receive-trace11.

THEOREM: rewrite-svc-receive-trace12

```

tm-processor (os-svc-receive-handler-thru-save-state (os),
              os-svc-receive-time8 (os))
= tm-processor (tm-processor (os-svc-receive-handler-thru-save-state (os),
                             os-time-for-svc-receive-handler-to-qempty-test (os)),
               os-svc-receive-time9 (os))

```

EVENT: Disable rewrite-svc-receive-trace12.

THEOREM: rewrite-svc-receive-trace13

```

tm-processor (os-svc-receive-handler-to-qempty-test (os),
              os-svc-receive-time9 (os))
= tm-processor (tm-processor (os-svc-receive-handler-to-qempty-test (os),
                             os-time-for-svc-receive-handler-thru-delivery (os)),
               os-svc-receive-time10 (os))

```

EVENT: Disable rewrite-svc-receive-trace13.

THEOREM: rewrite-svc-receive-trace14

```

tm-processor (os-svc-receive-handler-thru-delivery (os),
              os-svc-receive-time10 (os))
= tm-processor (tm-processor (os-svc-receive-handler-thru-delivery (os),
                             os-time-for-svc-receive-handler-to-svc-resume-task2 (os)),
               os-time-for-svc-receive-handler-to-svcr (os))

```

EVENT: Disable rewrite-svc-receive-trace14.

THEOREM: trace-svc-receive-handler-path4a

```

(good-os (os)
  ∧ (¬ tm-waiting (os))
  ∧ (((tm-svcid (os) mod '4) = '1)

```


$$\begin{aligned}
& \wedge \left((\neg \text{array-qempty}(\text{os-current-mbuffer}(\text{os-srcid}(\text{os}, \text{'16}), \text{os-current-taskid}(\text{os}), \text{os}))) \right. \\
& \quad \wedge \left(\text{getnth}(\text{'2} * \text{os-srcid}(\text{os}, \text{'16}), \text{os-status-table}(\text{os})) \right. \\
& \quad \quad \left. \neq \text{'2} \right) \left. \right) \\
\rightarrow & \left(\text{tm-processor}(\text{os-intended-svc-receive-interrupt}(\text{os}), \text{os-time-for-svc-receive-handler-path4a}(\text{os})) \right. \\
& \quad \left. = \text{os-svc-receive-handler-path4}(\text{os}) \right)
\end{aligned}$$

EVENT: Disable trace-svc-receive-handler-path4a.

DEFINITION:

os-svc-receive-time11(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

DEFINITION:

os-svc-receive-time12(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick)

DEFINITION:

os-svc-receive-time13(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

DEFINITION:

os-time-for-svc-receive-handler-path4b(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick

```

THEOREM: rewrite-svc-receive-trace15

$$\begin{aligned} & \text{tm-processor}(\text{os-intended-svc-receive-interrupt}(os), \\ & \quad \text{os-time-for-svc-receive-handler-path4b}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-intended-svc-receive-interrupt}(os), \\ & \quad \text{os-time-for-svc-receive-handler-thru-save-state}(os)), \\ & \quad \text{os-svc-receive-time11}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace15.

THEOREM: rewrite-svc-receive-trace16

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-receive-handler-thru-save-state}(os), \\ & \quad \text{os-svc-receive-time11}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-receive-handler-thru-save-state}(os), \\ & \quad \text{os-time-for-svc-receive-handler-to-qempty-test}(os)), \\ & \quad \text{os-svc-receive-time12}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace16.

THEOREM: rewrite-svc-receive-trace17

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-receive-handler-to-qempty-test}(os), \\ & \quad \text{os-svc-receive-time12}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-receive-handler-to-qempty-test}(os), \\ & \quad \text{os-time-for-svc-receive-handler-thru-delivery}(os)), \\ & \quad \text{os-svc-receive-time13}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace17.

THEOREM: rewrite-svc-receive-trace18

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-receive-handler-thru-delivery}(os), \\ & \quad \text{os-svc-receive-time13}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-receive-handler-thru-delivery}(os), \\ & \quad \text{os-time-for-svc-receive-handler-to-svc-resume-task3}(os)), \\ & \quad \text{os-time-for-svc-receive-handler-to-svc}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-receive-trace18.

THEOREM: trace-svc-receive-handler-path4b

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '1) \\
& \quad \quad \wedge ((\neg \text{array-qempty} (\text{os-current-mbuffer } (\text{os-srcid } (os), \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \text{os-current-taskid } (os), \\
& \quad \quad \quad os))) \\
& \quad \quad \wedge ((\text{getnth } ('2 * \text{os-srcid } (os), '16), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '2) \\
& \quad \quad \wedge (\text{getnth } ('1 \\
& \quad \quad \quad \quad + ('2 \\
& \quad \quad \quad \quad \quad * \text{os-srcid } (os, \\
& \quad \quad \quad \quad \quad \quad '16)), \\
& \quad \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad \neq \text{os-current-taskid } (os)))))) \\
& \rightarrow (\text{tm-processor } (\text{os-intended-svc-receive-interrupt } (os), \\
& \quad \quad \text{os-time-for-svc-receive-handler-path4b } (os)) \\
& \quad = \text{os-svc-receive-handler-path4 } (os))
\end{aligned}$$

EVENT: Disable trace-svc-receive-handler-path4b.

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-svc-receive-handler-path4 } (os) \\
& = \text{if } \text{getnth } ('2 * \text{os-srcid } (os), '16), \text{os-status-table } (os)) \neq '2 \\
& \quad \text{then } \text{os-time-for-svc-receive-handler-path4a } (os) \\
& \quad \text{else } \text{os-time-for-svc-receive-handler-path4b } (os) \text{ endif}
\end{aligned}$$

THEOREM: trace-svc-receive-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '1) \\
& \quad \quad \wedge ((\neg \text{array-qempty} (\text{os-current-mbuffer } (\text{os-srcid } (os), \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \text{os-current-taskid } (os), \\
& \quad \quad \quad os))) \\
& \quad \quad \wedge (\neg ((\text{getnth } ('2 * \text{os-srcid } (os), '16), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '2) \\
& \quad \quad \wedge (\text{getnth } ('1 \\
& \quad \quad \quad \quad + ('2 \\
& \quad \quad \quad \quad \quad * \text{os-srcid } (os, \\
& \quad \quad \quad \quad \quad \quad '16)),
\end{aligned}$$

$$\begin{aligned} & \text{os-status-table}(os) \\ & = \text{os-current-taskid}(os) \end{aligned} \text{)))))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-receive-interrupt}(os), \\ & \text{os-time-for-svc-receive-handler-path4}(os)) \\ & = \text{os-svc-receive-handler-path4}(os))$$

EVENT: Disable trace-svc-receive-handler-path4.

DEFINITION:

$$\begin{aligned} & \text{os-time-for-svc-receive-handler}(os) \\ = & \text{if array-qempty}(os\text{-current-mbuffer}(os\text{-srcid}(os, '16), \\ & \text{os-current-taskid}(os), \\ & os)) \\ & \text{then if array-qempty}(array\text{-deq}(os\text{-readyq}(os))) \\ & \quad \text{then os-time-for-svc-receive-handler-path1}(os) \\ & \quad \text{else os-time-for-svc-receive-handler-path2}(os) \text{ endif} \\ & \text{elseif}(\text{getnth}('2 * os\text{-srcid}(os, '16), os\text{-status-table}(os)) = '2) \\ & \quad \wedge (\text{getnth}('1 + ('2 * os\text{-srcid}(os, '16)), \\ & \quad \text{os-status-table}(os)) \\ & \quad = \text{os-current-taskid}(os)) \\ & \text{then os-time-for-svc-receive-handler-path3}(os) \\ & \text{else os-time-for-svc-receive-handler-path4}(os) \text{ endif} \end{aligned}$$

DEFINITION:

$$\begin{aligned} & \text{os-svc-receive-handler}(os) \\ = & \text{if array-qempty}(os\text{-current-mbuffer}(os\text{-srcid}(os, '16), \\ & \text{os-current-taskid}(os), \\ & os)) \\ & \text{then if array-qempty}(array\text{-deq}(os\text{-readyq}(os))) \\ & \quad \text{then os-svc-receive-handler-path1}(os) \\ & \quad \text{else os-svc-receive-handler-path2}(os) \text{ endif} \\ & \text{elseif}(\text{getnth}('2 * os\text{-srcid}(os, '16), os\text{-status-table}(os)) = '2) \\ & \quad \wedge (\text{getnth}('1 + ('2 * os\text{-srcid}(os, '16)), \\ & \quad \text{os-status-table}(os)) \\ & \quad = \text{os-current-taskid}(os)) \\ & \text{then os-svc-receive-handler-path3}(os) \\ & \text{else os-svc-receive-handler-path4}(os) \text{ endif} \end{aligned}$$

THEOREM: trace-svc-receive-handler

$$\begin{aligned} & (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '1))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-receive-interrupt}(os), \\ & \text{os-time-for-svc-receive-handler}(os)) \\ & = \text{os-svc-receive-handler}(os)) \end{aligned}$$

EVENT: Disable trace-svc-receive-handler.

DEFINITION:

```

os-intended-svc-tyi-interrupt (tm)
=  tm-set-memory (putnth (tm-pc (tm),
                          '0,
                          putnth (tm-sp (tm),
                                  '1,
                                  putnth (tm-pack-psw (tm-cc (tm),
                                                         tm-error (tm),
                                                         tm-svcflag (tm),
                                                         tm-svcid (tm)),
                                      '2,
                                      putnth (tm-svcid (tm),
                                              '8,
                                              putseg (os-task-table (tm),
                                                      '15,
                                                      putseg (os-segment-table (tm),
                                                              '159,
                                                              putseg (os-readyq (tm),
                                                                      '191,
                                                                      putseg (os-status-table (tm),
                                                                              '211,
                                                                              putseg (os-ibuffers (tm),
                                                                                      '243,
                                                                                      putseg (os-code (tm),
                                                                                              '2547,
                                                                                              tm-memor
tm-set-pc ('2690,
          tm-set-sp ('3310,
          tm-set-svmode ('1,
          tm-set-error ('0,
          tm-set-rwstate ('0,
                          tm)))))))))

```

THEOREM: equivalence-of-svc-tyi-interrupt-expressions

```

(good-os (os) ∧ ((¬ tm-waiting (os)) ∧ (¬ tm-errorp (os))))
→ (tm-execute-svc-interrupt (os) = os-intended-svc-tyi-interrupt (os))

```

EVENT: Disable equivalence-of-svc-tyi-interrupt-expressions.

DEFINITION:

```

os-time-for-svc-tyi-handler-thru-save-state (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

```

(DEFN OS-SVC-TYI-HANDLER-THRU-SAVE-STATE
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-SVCID OS)
'8
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2692

```

```

'3310
(TM-MEMORY OS)))))))))))))

(PUTNTH '2692
  '0
  (PUTNTH '3310
    '1
    (PUTNTH (TM-R2 OS)
      '2
      (PUTNTH (TM-R3 OS)
        '3
        (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
  (TIMES '9
    (ARRAY-QFIRST (OS-READYQ OS))))
  '8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyi-handler-thru-save-state
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-svc-tyi-interrupt}(os),$
 $\text{os-time-for-svc-tyi-handler-thru-save-state}(os))$
 $= \text{os-svc-tyi-handler-thru-save-state}(os))$

EVENT: Disable trace-svc-tyi-handler-thru-save-state.

DEFINITION:
 $\text{os-time-for-svc-tyi-handler-to-qempty-test}(os)$
 $= \text{'(tick tick tick tick tick tick tick tick tick tick tick tick tick tick tick tick tick)}$

(DEFN OS-SVC-TYI-HANDLER-TO-QEMPTY-TEST

```

(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '3
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2986

```



```

'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2998
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH (PLUS '243
(TIMES '8
(ARRAY-QFIRST (OS-READYQ OS))))
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '8
(ARRAY-QFIRST (OS-READYQ OS))
'243))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS)
NIL)

```

THEOREM: trace-svc-tyi-handler-to-qempty-test
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '3)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-tyi-handler-thru-save-state}(os),$
 $\text{os-time-for-svc-tyi-handler-to-qempty-test}(os))$
 $= \text{os-svc-tyi-handler-to-qempty-test}(os))$

EVENT: Disable trace-svc-tyi-handler-to-qempty-test.

DEFINITION:

os-time-for-svc-tyi-handler-to-dispatcher(*os*)
 $= '(\text{tick tick tick tick tick tick tick tick tick tick tick tick tick tick tick tick})$

```

(DEFN OS-SVC-TYI-HANDLER-TO-DISPATCHER
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '3
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'5
'0
OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)

```

```

'2547
(PUTNTH '2558
'3309
(PUTNTH '3025
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH (PLUS '211
(TIMES '2
(ARRAY-QFIRST (OS-READYQ OS))))
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '2
(ARRAY-QFIRST (OS-READYQ OS)))
'211))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyi-handler-to-dispatcher

$$\begin{aligned}
& \text{(good-os } (os) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge \text{array-qemptytyp } (\text{os-current-ibuffer } (\text{os-current-taskid } (os), os)))) \\
\rightarrow & \text{(tm-processor } (\text{os-svc-tyi-handler-to-qemptytyp-test } (os), \\
& \quad \text{os-time-for-svc-tyi-handler-to-dispatcher } (os)) \\
& = \text{os-svc-tyi-handler-to-dispatcher } (os))
\end{aligned}$$

EVENT: Disable trace-svc-tyi-handler-to-dispatcher.

DEFINITION:

```
os-time-for-svc-tyi-handler-to-wait-state (os)
= '(tick tick tick tick tick tick)
```

```
(DEFN OS-SVC-TYI-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(TM-RO OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '3
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'5
'0
```

```

                                OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3211
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH '191 '3 (TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyi-handler-to-wait-state
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge array-qempty (array-deq (os-readyq (*os*))))))
 \rightarrow (tm-processor (os-svc-tyi-handler-to-dispatcher (*os*),
os-time-for-svc-tyi-handler-to-wait-state (*os*))
= os-svc-tyi-handler-path1 (*os*))

EVENT: Disable trace-svc-tyi-handler-to-wait-state.

DEFINITION:
os-time-for-svc-tyi-handler-to-lpsw (*os*)
= '(tick tick tick tick tick tick tick tick tick tick

```
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick)
```

```
(DEFN OS-SVC-TYI-HANDLER-PATH2
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
        (TIMES '9
          (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
        (TIMES '9
          (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '3
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
```

```

      (ARRAY-DEQ (OS-READYQ OS))
      '191
      (PUTSEG
        (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
          '5
          '0
          OS)
        '211
        (PUTSEG (OS-IBUFFERS OS)
          '243
          (PUTSEG (OS-CODE OS)
            '2547
            (PUTNTH '2558
              '3309
              (PUTNTH '3217
                '3310
                (TM-MEMORY OS))))))))))
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-BASE (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (OS-NEW-LIMIT (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS)
      (TM-SLIMIT OS)
      '0
      '0
      '1000
      (TM-IPORTS OS)
      (TM-OPORTS OS))
      NIL)

```

THEOREM: trace-svc-tyi-handler-to-lpsw

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ (¬ array-qempty (array-deq (os-readyq (os)))))
  → (tm-processor (os-svc-tyi-handler-to-dispatcher (os),

```

```
os-time-for-svc-tyi-handler-to-lpsw (os))
= os-svc-tyi-handler-path2 (os))
```

EVENT: Disable trace-svc-tyi-handler-to-lpsw.

DEFINITION:

```
os-time-for-svc-tyi-handler-thru-delivery (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-SVC-TYI-HANDLER-THRU-DELIVERY
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '3
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE-WITH-DELIVERED-CHARACTER OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
```



```

(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3006
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (GETNTH (ARRAY-QFIRST (OS-READYQ OS))
(TABLE '8 (OS-IBUFFERS OS))))
'2
(PUTNTH (PLUS '15
(TIMES '9
(ARRAY-QFIRST (OS-READYQ OS))))
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '9
(ARRAY-QFIRST (OS-READYQ OS)))
'15))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyi-handler-thru-delivery

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (¬ array-qempty (os-current-ibuffer (os-current-taskid (os),
                                             os))))))
→ (tm-processor (os-svc-tyi-handler-to-qempty-test (os),
               os-time-for-svc-tyi-handler-thru-delivery (os))
   = os-svc-tyi-handler-thru-delivery (os))

```

EVENT: Disable trace-svc-tyi-handler-thru-delivery.

DEFINITION:

```

os-time-for-svc-tyi-handler-to-svc (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-SVC-TYI-HANDLER-PATH3
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-TASK-TABLE-WITH-DELIVERED-CHARACTER OS))
'0
(PUTNTH
(GETNTH (PLUS '1
            (TIMES '9
              (ARRAY-QFIRST (OS-READYQ OS))))
(OS-TASK-TABLE-WITH-DELIVERED-CHARACTER OS))
'1
(PUTNTH
(GETNTH (PLUS '8
            (TIMES '9
              (ARRAY-QFIRST (OS-READYQ OS))))
(OS-TASK-TABLE-WITH-DELIVERED-CHARACTER OS))
'2
(PUTNTH '3
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH

```

```

(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE-WITH-DELIVERED-CHARACTER OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3229
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH (ARRAY-QFIRST (GETNTH (OS-CURRENT-TASKID OS)
(TABLE '8 (OS-IBUFFERS OS))))
'3
(TM-REGS OS))
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyi-handler-to-svcr
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-processor (os-svc-tyi-handler-thru-delivery (*os*),
 os-time-for-svc-tyi-handler-to-svcr (*os*))
 = os-svc-tyi-handler-path3 (*os*))

EVENT: Disable trace-svc-tyi-handler-to-svcr.

DEFINITION:
 os-svc-tyi-time1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick)

DEFINITION:
 os-svc-tyi-time2 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick)

DEFINITION:
 os-time-for-svc-tyi-handler-path1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick)

THEOREM: rewrite-svc-tyi-trace1
 tm-processor (os-intended-svc-tyi-interrupt (*os*),
 os-time-for-svc-tyi-handler-path1 (*os*))
 = tm-processor (tm-processor (os-intended-svc-tyi-interrupt (*os*),
 os-time-for-svc-tyi-handler-thru-save-state (*os*)),
 os-svc-tyi-time1 (*os*))

EVENT: Disable rewrite-svc-tyi-trace1.

THEOREM: rewrite-svc-tyi-trace2
 tm-processor (os-svc-tyi-handler-thru-save-state (*os*), os-svc-tyi-time1 (*os*))
 = tm-processor (tm-processor (os-svc-tyi-handler-thru-save-state (*os*),
 os-time-for-svc-tyi-handler-to-qempty-test (*os*)),
 os-svc-tyi-time2 (*os*))

EVENT: Disable rewrite-svc-tyi-trace2.

THEOREM: rewrite-svc-tyi-trace3

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-tyi-handler-to-qempty-test}(os), \text{os-svc-tyi-time2}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyi-handler-to-qempty-test}(os), \\ & \quad \text{os-time-for-svc-tyi-handler-to-dispatcher}(os)), \\ & \quad \text{os-time-for-svc-tyi-handler-to-wait-state}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyi-trace3.

THEOREM: trace-svc-tyi-handler-path1

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '3) \\ & \quad \quad \wedge (\text{array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-tyi-interrupt}(os), \\ & \quad \text{os-time-for-svc-tyi-handler-path1}(os)) \\ = & \text{os-svc-tyi-handler-path1}(os)) \end{aligned}$$

EVENT: Disable trace-svc-tyi-handler-path1.

DEFINITION:

$$\begin{aligned} & \text{os-svc-tyi-time3}(os) \\ = & \text{'(tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick)} \end{aligned}$$

DEFINITION:

$$\begin{aligned} & \text{os-svc-tyi-time4}(os) \\ = & \text{'(tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick)} \end{aligned}$$

DEFINITION:

$$\text{os-time-for-svc-tyi-handler-path2}(os)$$

```

= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

THEOREM: rewrite-svc-tyi-trace4

```

tm-processor (os-intended-svc-tyi-interrupt (os),
             os-time-for-svc-tyi-handler-path2 (os))
= tm-processor (tm-processor (os-intended-svc-tyi-interrupt (os),
                           os-time-for-svc-tyi-handler-thru-save-state (os)),
               os-svc-tyi-time3 (os))

```

EVENT: Disable rewrite-svc-tyi-trace4.

THEOREM: rewrite-svc-tyi-trace5

```

tm-processor (os-svc-tyi-handler-thru-save-state (os), os-svc-tyi-time3 (os))
= tm-processor (tm-processor (os-svc-tyi-handler-thru-save-state (os),
                           os-time-for-svc-tyi-handler-to-qempty-test (os)),
               os-svc-tyi-time4 (os))

```

EVENT: Disable rewrite-svc-tyi-trace5.

THEOREM: rewrite-svc-tyi-trace6

```

tm-processor (os-svc-tyi-handler-to-qempty-test (os), os-svc-tyi-time4 (os))
= tm-processor (tm-processor (os-svc-tyi-handler-to-qempty-test (os),
                           os-time-for-svc-tyi-handler-to-dispatcher (os)),
               os-time-for-svc-tyi-handler-to-lpsw (os))

```

EVENT: Disable rewrite-svc-tyi-trace6.

THEOREM: trace-svc-tyi-handler-path2

```

(good-os (os)
  ∧ (¬ tm-waiting (os))
  ∧ (((tm-svcid (os) mod '4) = '3)
     ∧ (array-qempty (os-current-ibuffer (os-current-taskid (os),
                                           os))
        ∧ (¬ array-qempty (array-deq (os-readyq (os))))))))

```

→ (tm-processor (os-intended-svc-tyi-interrupt (os),
os-time-for-svc-tyi-handler-path2 (os))
= os-svc-tyi-handler-path2 (os))

EVENT: Disable trace-svc-tyi-handler-path2.

DEFINITION:

os-svc-tyi-time5 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

DEFINITION:

os-svc-tyi-time6 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-svc-tyi-handler-path3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

THEOREM: rewrite-svc-tyi-trace7

tm-processor (os-intended-svc-tyi-interrupt (os),
os-time-for-svc-tyi-handler-path3 (os))
= tm-processor (tm-processor (os-intended-svc-tyi-interrupt (os),
os-time-for-svc-tyi-handler-thru-save-state (os)),
os-svc-tyi-time5 (os))

EVENT: Disable rewrite-svc-tyi-trace7.

THEOREM: rewrite-svc-tyi-trace8

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-tyi-handler-thru-save-state}(os), \text{os-svc-tyi-time5}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyi-handler-thru-save-state}(os), \\ & \quad \text{os-time-for-svc-tyi-handler-to-qempty-test}(os)), \\ & \quad \text{os-svc-tyi-time6}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyi-trace8.

THEOREM: rewrite-svc-tyi-trace9

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-tyi-handler-to-qempty-test}(os), \text{os-svc-tyi-time6}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyi-handler-to-qempty-test}(os), \\ & \quad \text{os-time-for-svc-tyi-handler-thru-delivery}(os)), \\ & \quad \text{os-time-for-svc-tyi-handler-to-svcr}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyi-trace9.

THEOREM: trace-svc-tyi-handler-path3

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '3) \\ & \quad \quad \wedge (\neg \text{array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), \\ & \quad \quad \quad \text{os})))))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-tyi-interrupt}(os), \\ & \quad \text{os-time-for-svc-tyi-handler-path3}(os)) \\ = & \text{os-svc-tyi-handler-path3}(os)) \end{aligned}$$

EVENT: Disable trace-svc-tyi-handler-path3.

DEFINITION:

$$\begin{aligned} & \text{os-time-for-svc-tyi-handler}(os) \\ = & \text{if array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), os)) \\ & \text{then if array-qempty}(\text{array-deq}(\text{os-readyq}(os))) \\ & \quad \text{then os-time-for-svc-tyi-handler-path1}(os) \\ & \quad \text{else os-time-for-svc-tyi-handler-path2}(os) \text{ endif} \\ & \text{else os-time-for-svc-tyi-handler-path3}(os) \text{ endif} \end{aligned}$$

DEFINITION:

$$\begin{aligned} & \text{os-svc-tyi-handler}(os) \\ = & \text{if array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), os)) \\ & \text{then if array-qempty}(\text{array-deq}(\text{os-readyq}(os))) \\ & \quad \text{then os-svc-tyi-handler-path1}(os) \\ & \quad \text{else os-svc-tyi-handler-path2}(os) \text{ endif} \\ & \text{else os-svc-tyi-handler-path3}(os) \text{ endif} \end{aligned}$$

THEOREM: trace-svc-tyi-handler

$$\begin{aligned}
& (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '3))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-svc-tyi-interrupt}(os), \\
& \qquad \text{os-time-for-svc-tyi-handler}(os)) \\
= & \text{os-svc-tyi-handler}(os)
\end{aligned}$$

EVENT: Disable trace-svc-tyi-handler.

DEFINITION:

$$\begin{aligned}
& \text{os-intended-svc-tyo-interrupt}(tm) \\
= & \text{tm-set-memory}(\text{putnth}(\text{tm-pc}(tm), \\
& \qquad '0, \\
& \qquad \text{putnth}(\text{tm-sp}(tm), \\
& \qquad \qquad '1, \\
& \qquad \qquad \text{putnth}(\text{tm-pack-psw}(\text{tm-cc}(tm), \\
& \qquad \qquad \qquad \text{tm-error}(tm), \\
& \qquad \qquad \qquad \text{tm-svcflag}(tm), \\
& \qquad \qquad \qquad \text{tm-svcid}(tm)), \\
& \qquad \qquad \qquad '2, \\
& \qquad \qquad \text{putnth}(\text{tm-svcid}(tm), \\
& \qquad \qquad \qquad '8, \\
& \qquad \qquad \text{putseg}(\text{os-task-table}(tm), \\
& \qquad \qquad \qquad '15, \\
& \qquad \qquad \text{putseg}(\text{os-segment-table}(tm), \\
& \qquad \qquad \qquad '159, \\
& \qquad \qquad \text{putseg}(\text{os-readyq}(tm), \\
& \qquad \qquad \qquad '191, \\
& \qquad \qquad \text{putseg}(\text{os-status-table}(tm), \\
& \qquad \qquad \qquad '211, \\
& \qquad \qquad \text{putseg}(\text{os-obuffers}(tm), \\
& \qquad \qquad \qquad '371, \\
& \qquad \qquad \text{putseg}(\text{os-code}(tm), \\
& \qquad \qquad \qquad '2547, \\
& \qquad \qquad \qquad \text{tm-memor} \\
& \qquad \text{tm-set-pc}('2690, \\
& \qquad \qquad \text{tm-set-sp}('3310, \\
& \qquad \qquad \qquad \text{tm-set-svmode}('1, \\
& \qquad \qquad \qquad \qquad \text{tm-set-error}('0, \\
& \qquad \qquad \qquad \qquad \qquad \text{tm-set-rwstate}('0, \\
& \qquad \qquad \qquad \qquad \qquad \qquad tm))))))
\end{aligned}$$

THEOREM: equivalence-of-svc-tyo-interrupt-expressions

$$\begin{aligned}
& (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os)))) \\
\rightarrow & (\text{tm-execute-svc-interrupt}(os) = \text{os-intended-svc-tyo-interrupt}(os))
\end{aligned}$$

EVENT: Disable equivalence-of-svc-tyo-interrupt-expressions.

DEFINITION:

os-time-for-svc-tyo-handler-thru-save-state (*os*)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick)
```

(DEFN OS-SVC-TYO-HANDLER-THRU-SAVE-STATE

(OS)

(TM

(PUTNTH

(TM-R0 OS)

'0

(PUTNTH

(TM-R1 OS)

'1

(PUTNTH

(TM-PACK-PSW (TM-CC OS)

(TM-ERROR OS)

(TM-SVCFLAG OS)

(TM-SVCID OS))

'2

(PUTNTH

(TM-SVCID OS)

'8

(PUTNTH

(TM-R2 OS)

'13

(PUTNTH

(TM-R3 OS)

'14

(PUTSEG

(OS-NEW-TASK-TABLE OS)

'15

(PUTSEG

(OS-SEGMENT-TABLE OS)

'159

(PUTSEG

(OS-READYQ OS)

'191

(PUTSEG (OS-STATUS-TABLE OS)

```

'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2692
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2692
'0
(PUTNTH '3310
'1
(PUTNTH (TM-R2 OS)
'2
(PUTNTH (TM-R3 OS)
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
(TIMES '9
(ARRAY-QFIRST (OS-READYQ OS))))
'8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-thru-save-state
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os),$
 $\text{os-time-for-svc-tyo-handler-thru-save-state}(os))$
 $= \text{os-svc-tyo-handler-thru-save-state}(os))$

EVENT: Disable trace-svc-tyo-handler-thru-save-state.

DEFINITION:

os-time-for-svc-tyo-handler-to-qfullp-test(*os*)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-SVC-TYO-HANDLER-TO-QFULLP-TEST
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
              (TM-ERROR OS)
              (TM-SVCFLAG OS)
              (TM-SVCID OS))
'2
(PUTNTH '2
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
```

```

'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2930
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2942
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH (PLUS '371
(TIMES '8
(ARRAY-QFIRST (OS-READYQ OS))))
'3
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (TIMES '8
(ARRAY-QFIRST (OS-READYQ OS)))
'371))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-qfullp-test
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{tm-svcid}(os) \bmod '4) = '2)))$
 $\rightarrow (\text{tm-processor}(\text{os-svc-tyo-handler-thru-save-state}(os),$
 $\text{os-time-for-svc-tyo-handler-to-qfullp-test}(os))$
 $= \text{os-svc-tyo-handler-to-qfullp-test}(os))$

EVENT: Disable trace-svc-tyo-handler-to-qfullp-test.

DEFINITION:

os-time-for-svc-tyo-handler-to-dispatcher (*os*)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick)
```

(DEFN OS-SVC-TYO-HANDLER-TO-DISPATCHER

(OS)

(TM

(PUTNTH

(TM-R0 OS)

'0

(PUTNTH

(TM-R1 OS)

'1

(PUTNTH

(TM-PACK-PSW (TM-CC OS)

(TM-ERROR OS)

(TM-SVCFLAG OS)

(TM-SVCID OS))

'2

(PUTNTH '2

'8

(PUTNTH

(ARRAY-QFIRST (OS-READYQ OS))

'12

(PUTNTH

(TM-R2 OS)

'13

(PUTNTH

(TM-R3 OS)

'14

(PUTSEG

(OS-NEW-TASK-TABLE OS)

'15

(PUTSEG

(OS-SEGMENT-TABLE OS)

'159

(PUTSEG

(ARRAY-DEQ (OS-READYQ OS))

'191

```

(PUTSEG
  (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
    '4
    '0
    OS)
  '211
  (PUTSEG (OS-OBUFFERS OS)
    '371
    (PUTSEG (OS-CODE OS)
      '2547
      (PUTNTH '2558
        '3309
        (PUTNTH '2964
          '3310
          (TM-MEMORY OS)))))))))))))
(PUTNTH '3206
  '0
  (PUTNTH '3310
    '1
    (PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
      '2
      (PUTNTH (PLUS '211
        (TIMES '2
          (ARRAY-QFIRST (OS-READYQ OS))))
        '3
        (PUTNTH (GETNTH '3 (TM-REGS OS))
          '7
          (TM-REGS OS))))))
    (TM-CC-VALUE (TM-ALU-PLUS (TIMES '2
      (ARRAY-QFIRST (OS-READYQ OS)))
      '211))
    '0
    (TM-SVCFLAG OS)
    (TM-SVCID OS)
    (TM-BASE OS)
    (TM-LIMIT OS)
    (TM-SLIMIT OS)
    '1
    '0
    (TM-CLOCK OS)
    (TM-IPORTS OS)
    (TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-dispatcher
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
 ∧ (((tm-svcid (*os*) **mod** '4) = '2)
 ∧ array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))))))
 → (tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*),
 os-time-for-svc-tyo-handler-to-dispatcher (*os*))
 = os-svc-tyo-handler-to-dispatcher (*os*))

EVENT: Disable trace-svc-tyo-handler-to-dispatcher.

DEFINITION:
 os-time-for-svc-tyo-handler-to-wait-state (*os*)
 = '(tick tick tick tick tick tick)

```
(DEFN OS-SVC-TYO-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(TM-R0 OS)
'0
(PUTNTH
(TM-R1 OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH '2
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
```



```

(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
'4
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3211
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH (ARRAY-QFIRST (OS-READYQ OS))
'2
(PUTNTH '191
'3
(PUTNTH (GETNTH '3 (TM-REGS OS))
'7
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1

```

```

(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-wait-state

```

(good-os (os)
  ^ ((¬ tm-waiting (os)) ∧ array-qempty (array-deq (os-readyq (os))))))
→ (tm-processor (os-svc-tyo-handler-to-dispatcher (os),
                os-time-for-svc-tyo-handler-to-wait-state (os))
   = os-svc-tyo-handler-path1 (os))

```

EVENT: Disable trace-svc-tyo-handler-to-wait-state.

DEFINITION:

```

os-time-for-svc-tyo-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

(DEFN OS-SVC-TYO-HANDLER-PATH2

```

(OS)
(TM
 (PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
  (OS-NEW-TASK-TABLE OS))
 '0
 (PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
  (OS-NEW-TASK-TABLE OS))
 '1
 (PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9
                    (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS))))
  (OS-NEW-TASK-TABLE OS))
 '2
 (PUTNTH '2

```

```

'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (ARRAY-DEQ (OS-READYQ OS))
'191
(PUTSEG
  (OS-UPDATE-STATUS (ARRAY-QFIRST (OS-READYQ OS))
    '4
    '0
    OS)
'211
(PUTSEG (OS-OBUFFERS OS)
  '371
  (PUTSEG (OS-CODE OS)
    '2547
    (PUTNTH '2558
      '3309
      (PUTNTH '3217
        '3310
        (TM-MEMORY OS)))))))))))))
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)
(OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
  OS)

```

```

(OS-NEW-BASE (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
 OS)
(OS-NEW-LIMIT (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
 OS)
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-lpsw

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ (¬ array-qempty (array-deq (os-readyq (os)))))
→ (tm-processor (os-svc-tyo-handler-to-dispatcher (os),
              os-time-for-svc-tyo-handler-to-lpsw (os))
   = os-svc-tyo-handler-path2 (os))

```

EVENT: Disable trace-svc-tyo-handler-to-lpsw.

DEFINITION:

```

os-time-for-svc-tyo-handler-thru-enqueue (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

(DEFN OS-SVC-TYO-HANDLER-THRU-ENQUEUE

```

(OS)
(TM
 (PUTNTH
  (TM-R0 OS)
  '0
 (PUTNTH
  (TM-R1 OS)
  '1
 (PUTNTH
  (TM-PACK-PSW (TM-CC OS)
               (TM-ERROR OS)
               (TM-SVCFLAG OS)
               (TM-SVCID OS))
  '2

```

```

(PUTNTH '2
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (OS-READYQ OS)
'191
(PUTSEG
  (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '2951
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '2951
'0
(PUTNTH '3310
'1
(PUTNTH
  (GETNTH '3 (TM-REGS OS))
'2
(PUTNTH
  (PLUS '371
    (TIMES '8
      (ARRAY-QFIRST (OS-READYQ OS))))
'3

```

```

(PUTNTH (PLUS '375
          (PLUS (TIMES '8
                  (ARRAY-QFIRST (OS-READYQ OS)))
                (GETNTH '1
                        (GETNTH (ARRAY-QFIRST (OS-READYQ OS))
                                (TABLE '8 (OS-OBUFFERS OS))))))
        '4
(PUTNTH (GETNTH '3 (TM-REGS OS))
        '7
        (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                     (GETNTH (ARRAY-QFIRST (OS-READYQ OS))
                             (TABLE '8 (OS-OBUFFERS OS))))
 (GETNTH '3
         (GETNTH (ARRAY-QFIRST (OS-READYQ OS))
                 (TABLE '8 (OS-OBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPORTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-thru-enqueue

$$\begin{aligned}
& \text{(good-os } (os) \\
& \wedge \text{ ((}\neg \text{ tm-waiting } (os)) \\
& \quad \wedge \text{ (((tm-svcid } (os) \text{ mod } '4) = '2) \\
& \quad \quad \wedge \text{ (}\neg \text{ array-qfullp (os-current-obuffer (os-current-taskid } (os), \\
& \quad \quad \quad \text{os)))))) \\
& \rightarrow \text{ (tm-processor (os-svc-tyo-handler-to-qfullp-test } (os), \\
& \quad \quad \text{os-time-for-svc-tyo-handler-thru-enqueue } (os)) \\
& \quad = \text{ os-svc-tyo-handler-thru-enqueue } (os))
\end{aligned}$$

EVENT: Disable trace-svc-tyo-handler-thru-enqueue.

DEFINITION:

os-time-for-svc-tyo-handler-to-svcr1 (*os*)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick)
```

```
(DEFN OS-SVC-TYO-HANDLER-PATH3
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
        (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
        (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '2
'8
(PUTNTH
(ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
```

```

(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3229
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-POST-OUTPUT-INTERRUPT (ARRAY-QFIRST (OS-READYQ OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-svc1

```

(good-os (os)
  ^ ((¬ tm-waiting (os))
     ^ tm-oport-idlep (array-qfirst (os-readyq (os)), tm-oports (os))))
→ (tm-processor (os-svc-tyo-handler-thru-enqueue (os),
              os-time-for-svc-tyo-handler-to-svc1 (os))
   = os-svc-tyo-handler-path3 (os))

```

EVENT: Disable trace-svc-tyo-handler-to-svc1.

DEFINITION:

```

os-time-for-svc-tyo-handler-to-svc2 (os)
= '(tick tick tick tick tick tick tick tick tick

```



```
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-SVC-TYO-HANDLER-PATH4
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
           (OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
  (GETNTH (PLUS '1
              (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
  (GETNTH (PLUS '8
              (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH '2
'8
(PUTNTH
  (ARRAY-QFIRST (OS-READYQ OS))
'12
(PUTNTH
  (TM-R2 OS)
'13
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (OS-READYQ OS)
```

```

'191
(PUTSEG
  (OS-STATUS-TABLE OS)
  '211
  (PUTSEG (OS-OBUFFERS-WITH-ENQUEUED-CHARACTER OS)
    '371
    (PUTSEG (OS-CODE OS)
      '2547
      (PUTNTH '2558
        '3309
        (PUTNTH '3229
          '3310
          (TM-MEMORY OS)))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
'0
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-svc-tyo-handler-to-svcr2

$$\begin{aligned}
& \text{(good-os } (os) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\neg \text{tm-oport-idlep } (\text{array-qfirst } (\text{os-readyq } (os)), \\
& \quad \quad \quad \text{tm-oports } (os)))))) \\
& \rightarrow (\text{tm-processor } (\text{os-svc-tyo-handler-thru-enqueue } (os), \\
& \quad \quad \quad \text{os-time-for-svc-tyo-handler-to-svcr2 } (os)) \\
& \quad = \text{os-svc-tyo-handler-path4 } (os))
\end{aligned}$$

EVENT: Disable trace-svc-tyo-handler-to-svcr2.

DEFINITION:

```

os-svc-tyo-time1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick

```

tick tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-svc-tyo-time2 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

DEFINITION:

os-time-for-svc-tyo-handler-path1 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

THEOREM: rewrite-svc-tyo-trace1

tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path1 (*os*))
= tm-processor (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-thru-save-state (*os*)),
os-svc-tyo-time1 (*os*))

EVENT: Disable rewrite-svc-tyo-trace1.

THEOREM: rewrite-svc-tyo-trace2

tm-processor (os-svc-tyo-handler-thru-save-state (*os*), os-svc-tyo-time1 (*os*))
= tm-processor (tm-processor (os-svc-tyo-handler-thru-save-state (*os*),
os-time-for-svc-tyo-handler-to-qfullp-test (*os*)),
os-svc-tyo-time2 (*os*))

EVENT: Disable rewrite-svc-tyo-trace2.

THEOREM: rewrite-svc-tyo-trace3

tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*), os-svc-tyo-time2 (*os*))
= tm-processor (tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*),
os-time-for-svc-tyo-handler-to-dispatcher (*os*)),
os-time-for-svc-tyo-handler-to-wait-state (*os*))

EVENT: Disable rewrite-svc-tyo-trace3.

THEOREM: trace-svc-tyo-handler-path1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ (((tm-svcid (os) mod '4) = '2)
      ∧ (array-qfullp (os-current-obuffer (os-current-taskid (os),
                                          os))
        ∧ array-qempty (array-deq (os-readyq (os)))))))
→ (tm-processor (os-intended-svc-tyo-interrupt (os),
              os-time-for-svc-tyo-handler-path1 (os))
  = os-svc-tyo-handler-path1 (os))

```

EVENT: Disable trace-svc-tyo-handler-path1.

DEFINITION:

```

os-svc-tyo-time3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-svc-tyo-time4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-svc-tyo-handler-path2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick
  tick tick)

```

THEOREM: rewrite-svc-tyo-trace4

$$\begin{aligned} & \text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\ & \quad \text{os-time-for-svc-tyo-handler-path2}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\ & \quad \text{os-time-for-svc-tyo-handler-thru-save-state}(os)), \\ & \quad \text{os-svc-tyo-time3}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace4.

THEOREM: rewrite-svc-tyo-trace5

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-tyo-handler-thru-save-state}(os), \text{os-svc-tyo-time3}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyo-handler-thru-save-state}(os), \\ & \quad \text{os-time-for-svc-tyo-handler-to-qfullp-test}(os)), \\ & \quad \text{os-svc-tyo-time4}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace5.

THEOREM: rewrite-svc-tyo-trace6

$$\begin{aligned} & \text{tm-processor}(\text{os-svc-tyo-handler-to-qfullp-test}(os), \text{os-svc-tyo-time4}(os)) \\ = & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyo-handler-to-qfullp-test}(os), \\ & \quad \text{os-time-for-svc-tyo-handler-to-dispatcher}(os)), \\ & \quad \text{os-time-for-svc-tyo-handler-to-lpsw}(os)) \end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace6.

THEOREM: trace-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '2) \\ & \quad \quad \wedge (\text{array-qfullp}(\text{os-current-obuffer}(\text{os-current-taskid}(os), \\ & \quad \quad \quad \text{os})) \\ & \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))))))) \\ \rightarrow & (\text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\ & \quad \text{os-time-for-svc-tyo-handler-path2}(os)) \\ = & \text{os-svc-tyo-handler-path2}(os)) \end{aligned}$$

EVENT: Disable trace-svc-tyo-handler-path2.

DEFINITION:

$$\begin{aligned} & \text{os-svc-tyo-time5}(os) \\ = & '(\text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick}) \end{aligned}$$

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

```

DEFINITION:

```

os-svc-tyo-time6 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-svc-tyo-handler-path3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

THEOREM: rewrite-svc-tyo-trace7

```

tm-processor (os-intended-svc-tyo-interrupt (os),
              os-time-for-svc-tyo-handler-path3 (os))
= tm-processor (tm-processor (os-intended-svc-tyo-interrupt (os),
                             os-time-for-svc-tyo-handler-thru-save-state (os)),
               os-svc-tyo-time5 (os))

```

EVENT: Disable rewrite-svc-tyo-trace7.

THEOREM: rewrite-svc-tyo-trace8

```

tm-processor (os-svc-tyo-handler-thru-save-state (os), os-svc-tyo-time5 (os))
= tm-processor (tm-processor (os-svc-tyo-handler-thru-save-state (os),
                             os-time-for-svc-tyo-handler-to-qfullp-test (os)),
               os-svc-tyo-time6 (os))

```

EVENT: Disable rewrite-svc-tyo-trace8.

THEOREM: rewrite-svc-tyo-trace9

```

tm-processor (os-svc-tyo-handler-to-qfullp-test (os), os-svc-tyo-time6 (os))
= tm-processor (tm-processor (os-svc-tyo-handler-to-qfullp-test (os),
                             os-svc-tyo-time6 (os)),
               os-svc-tyo-time6 (os))

```

os-time-for-svc-tyo-handler-thru-enqueue (*os*),
os-time-for-svc-tyo-handler-to-svcr1 (*os*)

EVENT: Disable rewrite-svc-tyo-trace9.

THEOREM: trace-svc-tyo-handler-path3

(good-os (*os*)
 \wedge (\neg tm-waiting (*os*)
 \wedge (((tm-svcid (*os*) **mod** '4) = '2)
 \wedge (\neg array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os)))
 \wedge tm-oport-idlep (array-qfirst (os-readyq (*os*),
tm-oports (*os*))))))
 \rightarrow (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path3 (*os*)
= os-svc-tyo-handler-path3 (*os*))

EVENT: Disable trace-svc-tyo-handler-path3.

DEFINITION:

os-svc-tyo-time7 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick)

DEFINITION:

os-svc-tyo-time8 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick)

DEFINITION:

os-time-for-svc-tyo-handler-path4 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

```

THEOREM: rewrite-svc-tyo-trace10

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\
& \quad \text{os-time-for-svc-tyo-handler-path4}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\
& \quad \text{os-time-for-svc-tyo-handler-thru-save-state}(os)), \\
& \quad \text{os-svc-tyo-time7}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace10.

THEOREM: rewrite-svc-tyo-trace11

$$\begin{aligned}
& \text{tm-processor}(\text{os-svc-tyo-handler-thru-save-state}(os), \text{os-svc-tyo-time7}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyo-handler-thru-save-state}(os), \\
& \quad \text{os-time-for-svc-tyo-handler-to-qfullp-test}(os)), \\
& \quad \text{os-svc-tyo-time8}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace11.

THEOREM: rewrite-svc-tyo-trace12

$$\begin{aligned}
& \text{tm-processor}(\text{os-svc-tyo-handler-to-qfullp-test}(os), \text{os-svc-tyo-time8}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-svc-tyo-handler-to-qfullp-test}(os), \\
& \quad \text{os-time-for-svc-tyo-handler-thru-enqueue}(os)), \\
& \quad \text{os-time-for-svc-tyo-handler-to-svcr2}(os))
\end{aligned}$$

EVENT: Disable rewrite-svc-tyo-trace12.

THEOREM: trace-svc-tyo-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '2) \\
& \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-obuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \text{os}))) \\
& \quad \quad \quad \wedge (\neg \text{tm-oport-idlep}(\text{array-qfirst}(\text{os-readyq}(os), \\
& \quad \quad \quad \text{tm-oports}(os))))))) \\
& \rightarrow (\text{tm-processor}(\text{os-intended-svc-tyo-interrupt}(os), \\
& \quad \text{os-time-for-svc-tyo-handler-path4}(os)) \\
& \quad = \text{os-svc-tyo-handler-path4}(os))
\end{aligned}$$

EVENT: Disable trace-svc-tyo-handler-path4.

DEFINITION:

```
os-time-for-svc-tyo-handler (os)
=  if array-qfullp (os-current-obuffer (os-current-taskid (os), os))
    then if array-qemptytp (array-deq (os-readyq (os)))
        then os-time-for-svc-tyo-handler-path1 (os)
        else os-time-for-svc-tyo-handler-path2 (os) endif
    elseif tm-oport-idlep (array-qfirst (os-readyq (os)), tm-oports (os))
    then os-time-for-svc-tyo-handler-path3 (os)
    else os-time-for-svc-tyo-handler-path4 (os) endif
```

DEFINITION:

```
os-svc-tyo-handler (os)
=  if array-qfullp (os-current-obuffer (os-current-taskid (os), os))
    then if array-qemptytp (array-deq (os-readyq (os)))
        then os-svc-tyo-handler-path1 (os)
        else os-svc-tyo-handler-path2 (os) endif
    elseif tm-oport-idlep (array-qfirst (os-readyq (os)), tm-oports (os))
    then os-svc-tyo-handler-path3 (os)
    else os-svc-tyo-handler-path4 (os) endif
```

THEOREM: trace-svc-tyo-handler

```
(good-os (os)  $\wedge$  (( $\neg$  tm-waiting (os))  $\wedge$  ((tm-svcid (os) mod '4) = '2)))
 $\rightarrow$  (tm-processor (os-intended-svc-tyo-interrupt (os),
                  os-time-for-svc-tyo-handler (os))
      =  os-svc-tyo-handler (os))
```

EVENT: Disable trace-svc-tyo-handler.

(DEFN

OS-INTENDED-INPUT-INTERRUPT

(TM)

(TM-SET-MEMORY

(PUTNTH

(TM-PC TM)

'0

(PUTNTH

(TM-SP TM)

'1

(PUTNTH

(TM-PACK-PSW (TM-CC TM)

(TM-ERROR TM)

(TM-SVCFLAG TM)

```

(TM-SVCID TM))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS TM))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS TM))
(TM-IPOINTS TM)))
'9
(PUTSEG
(OS-TASK-TABLE TM)
'15
(PUTSEG (OS-SEGMENT-TABLE TM)
'159
(PUTSEG (OS-READYQ TM)
'191
(PUTSEG (OS-STATUS-TABLE TM)
'211
(PUTSEG (OS-IBUFFERS TM)
'243
(PUTSEG (OS-CODE TM)
'2547
(TM-MEMORY TM))))))))))
(TM-SET-PC '3042
(TM-SET-SP '3310
(TM-SET-SVMODE '1
(TM-SET-ERROR '0
(TM-SET-RWSTATE '0
(TM-SET-IPOINTS
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS TM))
(TM-IPOINTS TM))
TM))))))
NIL)

```

THEOREM: equivalence-of-input-interrupt-expressions

good-os (*os*)

→ (tm-execute-input-interrupt (*os*) = os-intended-input-interrupt (*os*))

EVENT: Disable equivalence-of-input-interrupt-expressions.

DEFINITION:

os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*)

= '(tick tick tick tick tick tick tick)

```

(DEFN OS-WAITING-INPUT-HANDLER-THRU-READYQ-EMPTY-TEST
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3053

```

```

'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3063
  '0
  (PUTNTH '3310
    '1
    (PUTNTH '191 '3 (TM-REGS OS))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-thru-readyq-empty-test
 (good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)) \wedge tm-waiting (*os*)))
 \rightarrow (tm-processor (os-intended-input-interrupt (*os*),
 os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*))
 = os-waiting-input-handler-thru-readyq-empty-test (*os*))

EVENT: Disable trace-waiting-input-handler-thru-readyq-empty-test.

DEFINITION:
 os-time-for-waiting-input-handler-to-ibuffer-full-test (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick)

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-IBUFFER-FULL-TEST
  (OS)
  (TM
  (PUTNTH
    (TM-PC OS)
    '0

```

```

(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3085
'3310
(TM-MEMORY OS))))))))))))))

```

```

(PUTNTH '3091
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'2
(PUTNTH '191
'3
(PUTNTH
(PPLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
'16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-ibuffer-full-test

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ (tm-waiting (os)
      ^ (getnth ('2
        * tm-interrupting-input-port (tm-iports (os)),
        os-status-table (os))
        = '5))))
→ (tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
  os-time-for-waiting-input-handler-to-ibuffer-full-test (os))

```

= os-waiting-input-handler-to-ibuffer-full-test (os))

EVENT: Disable trace-waiting-input-handler-to-ibuffer-full-test.

DEFINITION:

os-time-for-waiting-input-handler-to-dispatcher1 (os)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick)
```

```
(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER1
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
```

```

(PUTSEG
  (ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    (OS-READYQ OS))
  '191
  (PUTSEG
    (OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
      '0
      '0
      OS)
    '211
    (PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
      '243
      (PUTSEG (OS-CODE OS)
        '2547
        (PUTNTH '3112
          '3310
          (TM-MEMORY OS)))))))))))))
(PUTNTH '3206
  '0
  (PUTNTH '3310
    '1
    (PUTNTH
      (PLUS '256
        (TM-ICHR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
          (TM-IPOINTS OS))))
      '2
      (PUTNTH
        (PLUS '243
          (TIMES '8
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        '3
        (PUTNTH
          (PLUS '247
            (PLUS
              (TIMES '8
                (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
              (DECR-MOD (GETNTH '1
                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                  (TABLE '8 (OS-IBUFFERS OS))))
                '4)))
          '4
          (PUTNTH (PLUS '211
            (TIMES '2
              (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))

```



```

'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-PLUS
(PLUS '243
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher1

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ (tm-waiting (os)
      ^ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                         os))))))
→ (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
                    os-time-for-waiting-input-handler-to-dispatcher1 (os))
  = os-waiting-input-handler-to-dispatcher1 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher1.

DEFINITION:

```

os-time-for-waiting-input-handler-to-dispatcher2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER2
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(TM-IPORTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'0
'0
OS)

```

```

'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3126
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS)))))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
(GETNTH '3
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)

```

```

(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher2

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ (tm-waiting (os)
      ^ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                os)))
        ^ tm-iptort-errorp (tm-interrupting-input-port (tm-iports (os)),
                      tm-iports (os))))))
→ (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
                  os-time-for-waiting-input-handler-to-dispatcher2 (os))
  = os-waiting-input-handler-to-dispatcher2 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher2.

DEFINITION:

```

os-time-for-waiting-input-handler-to-dispatcher3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER3
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)

```

```

                (TM-ERROR OS)
                (TM-SVCFLAG OS)
                (TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3133
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH

```

```

(TM-ICHR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                (TM-IPOINTS OS)))
'2
(PUTNTH
 (PLUS '243
  (TIMES '8
   (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
 (PLUS '247
  (PLUS (TIMES '8
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
        (GETNTH '1
                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                        (TABLE '8 (OS-IBUFFERS OS)))))))
'4
(PUTNTH (PLUS '211
          (TIMES '2
           (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                       (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                               (TABLE '8 (OS-IBUFFERS OS))))
                  (GETNTH '3
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                          (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher3
(good-os (*os*))

```

 $\wedge$  (tm-some-input-interruptp (tm-iports (os))
       $\wedge$  (tm-waiting (os)
             $\wedge$  (( $\neg$  array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                         os)))
                   $\wedge$  ( $\neg$  tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                           tm-iports (os))))))
 $\rightarrow$  (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
                       os-time-for-waiting-input-handler-to-dispatcher3 (os))
           = os-waiting-input-handler-to-dispatcher3 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher3.

DEFINITION:

```

os-time-for-waiting-input-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)

```

```

(DEFN OS-WAITING-INPUT-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(OS-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
        (TIMES '9
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
        (TIMES '9
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8

```

```

(PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    (TM-IPOINTS OS)))
  '9
(PUTNTH '3206
  '10
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    (OS-READYQ OS))
  '191
(PUTSEG
  (OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    '0
    '0
    OS)
  '211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
  '243
  (PUTSEG (OS-CODE OS)
    '2547
    (PUTNTH '3217
      '3310
      (TM-MEMORY OS)))))))))))))
(GETSEG '0
  '8
  (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    (TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS '8
  (TIMES '9
    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
  (OS-TASK-TABLE OS))
  '4)
(QUOTIENT
  (REMAINDER
    (GETNTH (PLUS '8

```



```

                                (TIMES '9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
                                (OS-TASK-TABLE OS))
'256)
'4)
(QUOTIENT
 (REMAINDER
  (GETNTH (PLUS '8
            (TIMES '9
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
            (OS-TASK-TABLE OS))
          '512)
  '256)
 (QUOTIENT (GETNTH (PLUS '8
                    (TIMES '9
                    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
                    (OS-TASK-TABLE OS))
          '512)
  (BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (TM-SLIMIT OS)
  '0
  '0
  '1000
  (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                             (TM-IPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-lpsw1

```

(good-os (os)
 ^ (tm-some-input-interruptp (tm-iports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher1 (os),
                 os-time-for-waiting-input-handler-to-lpsw (os))
   = os-waiting-input-handler-path1 (os))

```

EVENT: Disable trace-waiting-input-handler-to-lpsw1.

```

(DEFN
 OS-WAITING-INPUT-HANDLER-PATH2

```

```

(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
            (OS-TASK-TABLE OS))
'0
(PUTNTH
  (GETNTH (PLUS '1
              (TIMES '9
                (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
            (OS-TASK-TABLE OS))
'1
(PUTNTH
  (GETNTH (PLUS '8
              (TIMES '9
                (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
            (OS-TASK-TABLE OS))
'2
(PUTNTH
  (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                    (TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
  (TM-R3 OS)
'14
(PUTSEG
  (OS-TASK-TABLE OS)
'15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
'159
(PUTSEG
  (ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
              (OS-READYQ OS))
'191
(PUTSEG
  (OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'0

```

```

                                '0
                                OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3217
'3310
(TM-MEMORY OS)))))))))))))
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'4)
(QUOTIENT
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'256)
'4)
(QUOTIENT
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'512)
'256)
(QUOTIENT (GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
(OS-TASK-TABLE OS))
'512)
(BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '2 (OS-SEGMENT-TABLE OS))))
(LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '2 (OS-SEGMENT-TABLE OS))))

```

```

(TM-SLIMIT OS)
'0
'0
'1000
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))

(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-lpsw2

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher2 (os),
                 os-time-for-waiting-input-handler-to-lpsw (os))
   = os-waiting-input-handler-path2 (os))

```

EVENT: Disable trace-waiting-input-handler-to-lpsw2.

```

(DEFN
OS-WAITING-INPUT-HANDLER-PATH3
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
          (OS-TASK-TABLE OS))
  '0
(PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
          (OS-TASK-TABLE OS))
  '1
(PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9
                    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
          (OS-TASK-TABLE OS))
  '2
(PUTNTH
  (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  '8

```

```

(PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                   (TM-IPOINTS OS)))
  '9
(PUTNTH '3206
  '10
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
              (OS-READYQ OS))
  '191
(PUTSEG
  (OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                    '0
                    '0
                    OS)
  '211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
  '243
(PUTSEG (OS-CODE OS)
  '2547
(PUTNTH '3217
  '3310
(TM-MEMORY OS))))))))))))))
(GETSEG '0
  '8
  (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
          (TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS '8
                    (TIMES '9
                      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
              (OS-TASK-TABLE OS))
  '4)
(QUOTIENT
  (REMAINDER
    (GETNTH (PLUS '8

```

```

(TIMES '9
  (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(OS-TASK-TABLE OS))
'256)
'4)
(QUOTIENT
 (REMAINDER
  (GETNTH (PLUS '8
    (TIMES '9
      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
      (OS-TASK-TABLE OS)))
    '512)
  '256)
 (QUOTIENT (GETNTH (PLUS '8
  (TIMES '9
    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
    (OS-TASK-TABLE OS)))
    '512)
 (BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  (TABLE '2 (OS-SEGMENT-TABLE OS))))
 (LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  (TABLE '2 (OS-SEGMENT-TABLE OS))))
 (TM-SLIMIT OS)
 '0
 '0
 '1000
 (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  (TM-IPOINTS OS))
 (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-lpsw3

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher3 (os),
  os-time-for-waiting-input-handler-to-lpsw (os))
  = os-waiting-input-handler-path3 (os))

```

EVENT: Disable trace-waiting-input-handler-to-lpsw3.

DEFINITION:

```

os-waiting-input-time1 (os)
= ' (tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick tick

```

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-waiting-input-time2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-waiting-input-handler-path1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick)

```

THEOREM: rewrite-waiting-input-trace1

```

tm-processor (os-intended-input-interrupt (os),
              os-time-for-waiting-input-handler-path1 (os))
= tm-processor (tm-processor (os-intended-input-interrupt (os),
                             os-time-for-waiting-input-handler-thru-readyq-empty-test (os)),
               os-waiting-input-time1 (os))

```

EVENT: Disable rewrite-waiting-input-trace1.

THEOREM: rewrite-waiting-input-trace2

```

tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
              os-waiting-input-time1 (os))
= tm-processor (tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
                             os-time-for-waiting-input-handler-to-ibuffer-full-test (os)),
               os-waiting-input-time2 (os))

```

EVENT: Disable rewrite-waiting-input-trace2.

THEOREM: rewrite-waiting-input-trace3

```

tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
              os-waiting-input-time2 (os))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
                             os-time-for-waiting-input-handler-to-dispatcher1 (os)),
               os-time-for-waiting-input-handler-to-lpsw (os))

```

EVENT: Disable rewrite-waiting-input-trace3.

THEOREM: trace-waiting-input-handler-path1

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
     ^ (tm-waiting (os)
        ^ ((getnth ('2
                    * tm-interrupting-input-port (tm-iports (os)),
                    os-status-table (os))
            = '5)
         ^ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                             os))))))
→ (tm-processor (os-intended-input-interrupt (os),
                os-time-for-waiting-input-handler-path1 (os))
   = os-waiting-input-handler-path1 (os))

```

EVENT: Disable trace-waiting-input-handler-path1.

DEFINITION:

```

os-waiting-input-time3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-waiting-input-time4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick)

```

DEFINITION:


```

os-time-for-waiting-input-handler-path2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick)

```

THEOREM: rewrite-waiting-input-trace4

```

tm-processor (os-intended-input-interrupt (os),
             os-time-for-waiting-input-handler-path2 (os))
= tm-processor (tm-processor (os-intended-input-interrupt (os),
                            os-time-for-waiting-input-handler-thru-readyq-empty-test (os)),
               os-waiting-input-time3 (os))

```

EVENT: Disable rewrite-waiting-input-trace4.

THEOREM: rewrite-waiting-input-trace5

```

tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
             os-waiting-input-time3 (os))
= tm-processor (tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
                            os-time-for-waiting-input-handler-to-ibuffer-full-test (os)),
               os-waiting-input-time4 (os))

```

EVENT: Disable rewrite-waiting-input-trace5.

THEOREM: rewrite-waiting-input-trace6

```

tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
             os-waiting-input-time4 (os))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (os),
                            os-time-for-waiting-input-handler-to-dispatcher2 (os)),
               os-time-for-waiting-input-handler-to-lpsw (os))

```

EVENT: Disable rewrite-waiting-input-trace6.

THEOREM: trace-waiting-input-handler-path2

```

(good-os (os)
 & (tm-some-input-interruptp (tm-iports (os))
   & (tm-waiting (os)
     & ((getnth ('2
                 * tm-interrupting-input-port (tm-iports (os))),

```

$$\begin{aligned}
& \text{os-status-table}(os) \\
= & \text{'5)} \\
\wedge & ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \text{os}))) \\
& \wedge \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \text{tm-iports}(os)))))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \text{os-time-for-waiting-input-handler-path2}(os)) \\
= & \text{os-waiting-input-handler-path2}(os)
\end{aligned}$$

EVENT: Disable trace-waiting-input-handler-path2.

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-input-time5}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-input-time6}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-waiting-input-handler-path3}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick)}
\end{aligned}$$

THEOREM: rewrite-waiting-input-trace7
 $\text{tm-processor}(\text{os-intended-input-interrupt}(os),$


```

DEFINITION:
os-time-for-waiting-input-handler-to-ibuffer-full-test2 (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-IBUFFER-FULL-TEST2
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)

```

```

'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3053
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3091
'0
(PUTNTH '3310
'1
(PUTNTH '191
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE
(GETNTH (TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(OS-STATUS-TABLE OS))
'5))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-ibuffer-full-test2

```

(good-os (os)
^ (tm-some-input-interruptp (tm-iports (os))
^ (tm-waiting (os)
^ (getnth ('2
* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))

```

```

→ (tm-processor (os-waiting-input-handler-thru-readyq-empty-test (os),
                    os-time-for-waiting-input-handler-to-ibuffer-full-test2 (os))
    = os-waiting-input-handler-to-ibuffer-full-test2 (os))

```

EVENT: Disable trace-waiting-input-handler-to-ibuffer-full-test2.

DEFINITION:

```

os-time-for-waiting-input-handler-to-dispatcher4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick)

```

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER4
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15

```

```

(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3112
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS)))))))))

```

```

(TM-CC-VALUE
 (TM-ALU-PLUS
  (PLUS '243
   (PLUS
    (TIMES '8
     (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
    (DECR-MOD (GETNTH '1
                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                        (TABLE '8 (OS-IBUFFERS OS))))
                '4)))
   '4))
 '0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
 '1
 '0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                          (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher4

```

(good-os (os)
 ^ (tm-some-input-interruptp (tm-iports (os))
   ^ (tm-waiting (os)
     ^ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                       os))))))
→ (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (os),
                  os-time-for-waiting-input-handler-to-dispatcher4 (os))
   = os-waiting-input-handler-to-dispatcher4 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher4.

DEFINITION:

```

os-time-for-waiting-input-handler-to-dispatcher5 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```



```

(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER5
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3126
'3310

```

```

(TM-MEMORY OS))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
(GETNTH '3
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)

```

```

(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher5

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ (tm-waiting (os)
      ^ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                                    os)))
        ^ tm-iptort-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                                tm-iports (os))))))
→ (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (os),
                  os-time-for-waiting-input-handler-to-dispatcher5 (os))
  = os-waiting-input-handler-to-dispatcher5 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher5.

DEFINITION:

```

os-time-for-waiting-input-handler-to-dispatcher6 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

(DEFN OS-WAITING-INPUT-HANDLER-TO-DISPATCHER6

```

(OS)
(TM
 (PUTNTH
  (TM-PC OS)
  '0
 (PUTNTH
  (TM-SP OS)
  '1
 (PUTNTH
  (TM-PACK-PSW (TM-CC OS)
                (TM-ERROR OS)
                (TM-SVCFLAG OS)
                (TM-SVCID OS))
  '2
 (PUTNTH
  (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  '8

```

```

(PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                   (TM-IPOINTS OS)))
  '9
(PUTNTH '3206
  '10
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG (OS-READYQ OS)
  '191
(PUTSEG (OS-STATUS-TABLE OS)
  '211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
  '243
(PUTSEG (OS-CODE OS)
  '2547
(PUTNTH '3133
  '3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3206
  '0
(PUTNTH '3310
  '1
(PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                   (TM-IPOINTS OS)))
  '2
(PUTNTH
  (PLUS '243
    (TIMES '8
      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
  '3
(PUTNTH
  (PLUS '247
    (PLUS (TIMES '8
      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
    (GETNTH '1

```

```

                                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                (TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
          (TIMES '2
                (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))))
'5
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                (TABLE '8 (OS-IBUFFERS OS))))
                    (GETNTH '3
                        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                (TABLE '8 (OS-IBUFFERS OS)))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-dispatcher6

```

(good-os (os)
  ∧ (tm-some-input-interruptp (tm-iports (os))
     ∧ (tm-waiting (os)
        ∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                                    os)))
           ∧ (¬ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                                tm-iports (os)))))))
→ (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (os),
                   os-time-for-waiting-input-handler-to-dispatcher6 (os))
   = os-waiting-input-handler-to-dispatcher6 (os))

```

EVENT: Disable trace-waiting-input-handler-to-dispatcher6.

DEFINITION:

```
os-time-for-waiting-input-handler-to-wait (os)
= '(tick tick tick tick tick tick)
```

```
(DEFN OS-WAITING-INPUT-HANDLER-PATH4
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
```

```

'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3211
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH '191
'3
(PUTNTH
(PLUS '247
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS)))))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))

```

(TM-OPOINTS OS))
NIL)

THEOREM: trace-waiting-input-handler-to-wait1

(good-os (*os*)
^ (tm-some-input-interruptp (tm-iports (*os*)) ^ tm-waiting (*os*)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher4 (*os*),
os-time-for-waiting-input-handler-to-wait (*os*))
= os-waiting-input-handler-path4 (*os*))

EVENT: Disable trace-waiting-input-handler-to-wait1.

(DEFN
OS-WAITING-INPUT-HANDLER-PATH5
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)


```

'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3211
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH '191
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS)))))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)

```

```

(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))

(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-wait2

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher5 (os),
                  os-time-for-waiting-input-handler-to-wait (os))
  = os-waiting-input-handler-path5 (os))

```

EVENT: Disable trace-waiting-input-handler-to-wait2.

```

(DEFN
OS-WAITING-INPUT-HANDLER-PATH6
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
              (TM-ERROR OS)
              (TM-SVCFLAG OS)
              (TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                  (TM-IPOINTS OS)))

```

```

'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3211
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'2
(PUTNTH '191
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))

```

```

'5
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-input-handler-to-wait3

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-input-handler-to-dispatcher6 (os),
  os-time-for-waiting-input-handler-to-wait (os))
  = os-waiting-input-handler-path6 (os))

```

EVENT: Disable trace-waiting-input-handler-to-wait3.

DEFINITION:

```

os-waiting-input-time7 (os)
= '(tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-waiting-input-time8 (os)
= '(tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick
  tick tick tick)

```

DEFINITION:

```

os-time-for-waiting-input-handler-path4 (os)
= '(tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick
  tick tick tick tick)

```

THEOREM: rewrite-waiting-input-trace10

$$\begin{aligned} & \text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-waiting-input-handler-path4} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-waiting-input-handler-thru-readyq-empty-test} (os)), \\ & \quad \text{os-waiting-input-time7} (os)) \end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace10.

THEOREM: rewrite-waiting-input-trace11

$$\begin{aligned} & \text{tm-processor} (\text{os-waiting-input-handler-thru-readyq-empty-test} (os), \\ & \quad \text{os-waiting-input-time7} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-waiting-input-handler-thru-readyq-empty-test} (os), \\ & \quad \text{os-time-for-waiting-input-handler-to-ibuffer-full-test2} (os)), \\ & \quad \text{os-waiting-input-time8} (os)) \end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace11.

THEOREM: rewrite-waiting-input-trace12

$$\begin{aligned} & \text{tm-processor} (\text{os-waiting-input-handler-to-ibuffer-full-test2} (os), \\ & \quad \text{os-waiting-input-time8} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-waiting-input-handler-to-ibuffer-full-test2} (os), \\ & \quad \text{os-time-for-waiting-input-handler-to-dispatcher4} (os)), \\ & \quad \text{os-time-for-waiting-input-handler-to-wait} (os)) \end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace12.

THEOREM: trace-waiting-input-handler-path4

$$\begin{aligned} & (\text{good-os} (os) \\ & \quad \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \quad \wedge (\text{tm-waiting} (os) \\ & \quad \quad \quad \wedge ((\text{getnth} ('2 \\ & \quad \quad \quad \quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \quad \text{os-status-table} (os)) \\ & \quad \quad \quad \neq '5) \\ & \quad \quad \quad \wedge \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \quad \text{os})))))) \\ \rightarrow & (\text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-waiting-input-handler-path4} (os)) \\ = & \text{os-waiting-input-handler-path4} (os)) \end{aligned}$$

EVENT: Disable trace-waiting-input-handler-path4.

DEFINITION:

os-waiting-input-time9(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick)

DEFINITION:

os-waiting-input-time10(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

DEFINITION:

os-time-for-waiting-input-handler-path5(*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick)

THEOREM: rewrite-waiting-input-trace13

tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path5 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*)),
os-waiting-input-time9 (*os*))

EVENT: Disable rewrite-waiting-input-trace13.

THEOREM: rewrite-waiting-input-trace14

tm-processor (os-waiting-input-handler-thru-readyq-empty-test (*os*),
os-waiting-input-time9 (*os*))
= tm-processor (tm-processor (os-waiting-input-handler-thru-readyq-empty-test (*os*),
os-time-for-waiting-input-handler-to-ibuffer-full-test2 (*os*)),
os-waiting-input-time10 (*os*))

EVENT: Disable rewrite-waiting-input-trace14.

THEOREM: rewrite-waiting-input-trace15

tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-waiting-input-time10 (*os*))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-time-for-waiting-input-handler-to-dispatcher5 (*os*)),
os-time-for-waiting-input-handler-to-wait (*os*))

EVENT: Disable rewrite-waiting-input-trace15.

THEOREM: trace-waiting-input-handler-path5

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
& \quad \wedge (\text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \neq '5) \\
& \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os}))) \\
& \quad \quad \quad \wedge \text{tm-iport-errorp } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{tm-iports } (os)))))) \\
& \rightarrow (\text{tm-processor } (\text{os-intended-input-interrupt } (os), \\
& \quad \quad \text{os-time-for-waiting-input-handler-path5 } (os)) \\
& \quad = \text{os-waiting-input-handler-path5 } (os))
\end{aligned}$$

EVENT: Disable trace-waiting-input-handler-path5.

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-input-time11 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-input-time12 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-waiting-input-handler-path6 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick)
\end{aligned}$$

THEOREM: rewrite-waiting-input-trace16

$$\begin{aligned}
& \text{tm-processor } (\text{os-intended-input-interrupt } (os), \\
& \quad \text{os-time-for-waiting-input-handler-path6 } (os)) \\
& = \text{tm-processor } (\text{tm-processor } (\text{os-intended-input-interrupt } (os), \\
& \quad \quad \text{os-time-for-waiting-input-handler-thru-readyq-empty-test } (os)), \\
& \quad \text{os-waiting-input-time11 } (os))
\end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace16.

THEOREM: rewrite-waiting-input-trace17

$$\begin{aligned} & \text{tm-processor} (\text{os-waiting-input-handler-thru-readyq-empty-test} (os), \\ & \quad \text{os-waiting-input-time11} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-waiting-input-handler-thru-readyq-empty-test} (os), \\ & \quad \text{os-time-for-waiting-input-handler-to-ibuffer-full-test2} (os)), \\ & \quad \text{os-waiting-input-time12} (os)) \end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace17.

THEOREM: rewrite-waiting-input-trace18

$$\begin{aligned} & \text{tm-processor} (\text{os-waiting-input-handler-to-ibuffer-full-test2} (os), \\ & \quad \text{os-waiting-input-time12} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-waiting-input-handler-to-ibuffer-full-test2} (os), \\ & \quad \text{os-time-for-waiting-input-handler-to-dispatcher6} (os)), \\ & \quad \text{os-time-for-waiting-input-handler-to-wait} (os)) \end{aligned}$$

EVENT: Disable rewrite-waiting-input-trace18.

THEOREM: trace-waiting-input-handler-path6

$$\begin{aligned} & (\text{good-os} (os) \\ & \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \wedge (\text{tm-waiting} (os) \\ & \quad \quad \wedge ((\text{getnth} ('2 \\ & \quad \quad \quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os-status-table} (os)) \\ & \quad \quad \neq '5) \\ & \quad \quad \wedge ((\neg \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge (\neg \text{tm-iport-errorp} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{tm-iports} (os)))))))))) \\ \rightarrow & (\text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-waiting-input-handler-path6} (os)) \\ = & \text{os-waiting-input-handler-path6} (os)) \end{aligned}$$

EVENT: Disable trace-waiting-input-handler-path6.

DEFINITION:

$$\begin{aligned} & \text{os-time-for-waiting-input-handler} (os) \\ = & \text{if getnth} ('2 * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \text{os-status-table} (os)) \end{aligned}$$


```

= '5
then if array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                     os))
    then os-time-for-waiting-input-handler-path1 (os)
    elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                     tm-iports (os))
    then os-time-for-waiting-input-handler-path2 (os)
    else os-time-for-waiting-input-handler-path3 (os) endif
elseif array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                     os))
then os-time-for-waiting-input-handler-path4 (os)
elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                     tm-iports (os))
then os-time-for-waiting-input-handler-path5 (os)
else os-time-for-waiting-input-handler-path6 (os) endif

```

DEFINITION:

```

os-waiting-input-handler (os)
= if getnth ('2 * tm-interrupting-input-port (tm-iports (os)),
            os-status-table (os))
    = '5
    then if array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                     os))
        then os-waiting-input-handler-path1 (os)
        elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                     tm-iports (os))
        then os-waiting-input-handler-path2 (os)
        else os-waiting-input-handler-path3 (os) endif
    elseif array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                     os))
    then os-waiting-input-handler-path4 (os)
    elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                     tm-iports (os))
    then os-waiting-input-handler-path5 (os)
    else os-waiting-input-handler-path6 (os) endif

```

THEOREM: trace-waiting-input-handler

```

(good-os (os)
  ∧ (tm-some-input-interruptp (tm-iports (os)) ∧ tm-waiting (os)))
→ (tm-processor (os-intended-input-interrupt (os),
                os-time-for-waiting-input-handler (os))
   = os-waiting-input-handler (os))

```

EVENT: Disable trace-waiting-input-handler.

DEFINITION:

```
os-time-for-running-input-handler-thru-save-state(os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick)
```

```
(DEFN OS-RUNNING-INPUT-HANDLER-THRU-SAVE-STATE
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
              (TM-ERROR OS)
              (TM-SVCFLAG OS)
              (TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                  (TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
```

```

'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3063
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3063
'0
(PUTNTH '3310
'1
(PUTNTH (GETNTH '2 (TM-REGS OS))
'2
(PUTNTH (GETNTH '3 (TM-REGS OS))
'3
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
(TIMES '9
(ARRAY-QFIRST (OS-READYQ OS))))
'8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-thru-save-state

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ (¬ tm-waiting (os))))
→ (tm-processor (os-intended-input-interrupt (os),
                os-time-for-running-input-handler-thru-save-state (os))
   = os-running-input-handler-thru-save-state (os))

```

EVENT: Disable trace-running-input-handler-thru-save-state.

DEFINITION:

```

os-time-for-running-input-handler-to-ibuffer-full-test (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-IBUFFER-FULL-TEST
  (OS)
  (TM
    (PUTNTH
      (TM-PC OS)
      '0
    (PUTNTH
      (TM-SP OS)
      '1
    (PUTNTH
      (TM-PACK-PSW (TM-CC OS)
                    (TM-ERROR OS)
                    (TM-SVCFLAG OS)
                    (TM-SVCID OS))
      '2
    (PUTNTH
      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
      '8
    (PUTNTH
      (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                        (TM-IPOINTS OS)))
      '9
    (PUTNTH '3231
      '10
    (PUTNTH
      (TM-R2 OS)
      '13
    (PUTNTH
      (TM-R3 OS)

```

```

'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3085
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3091
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'2
(PUTNTH '191
'3
(PUTNTH
(PLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))

```

```

(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
                               '16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))

(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-ibuffer-full-test

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                 * tm-interrupting-input-port (tm-iports (os)),
                 os-status-table (os))
         = '5))))
→ (tm-processor (os-running-input-handler-thru-save-state (os),
                 os-time-for-running-input-handler-to-ibuffer-full-test (os))
  = os-running-input-handler-to-ibuffer-full-test (os))

```

EVENT: Disable trace-running-input-handler-to-ibuffer-full-test.

DEFINITION:

```

os-time-for-running-input-handler-to-resume-task1 (os)
= '(tick tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK1
  (OS)
  (TM
   (PUTNTH
    (TM-PC OS)
    '0
    (PUTNTH

```

```

(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558

```

```

'3309
(PUTNTH '3112
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-PLUS
(PLUS '243
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4))

```



```

'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-resume-task1

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                         os))))))
→ (tm-processor (os-running-input-handler-to-ibuffer-full-test (os),
                  os-time-for-running-input-handler-to-resume-task1 (os))
  = os-running-input-handler-to-resume-task1 (os))

```

EVENT: Disable trace-running-input-handler-to-resume-task1.

DEFINITION:

```

os-time-for-running-input-handler-to-resume-task2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK2
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)

```

```

                (TM-ERROR OS)
                (TM-SVCFLAG OS)
                (TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(TM-IPORTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3126
'3310
(TM-MEMORY OS)))))))))))))))))

```

```

(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(PUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PUS '247
(PUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
(GETNTH '3
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))

```

```

(TM-IPORTS OS))
(TM-OPOINTS OS))
NIL)

THEOREM: trace-running-input-handler-to-resume-task2
(good-os (os)
  ∧ (tm-some-input-interruptp (tm-ports (os))
    ∧ ((¬ tm-waiting (os))
      ∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-ports (os)),
        os)))
        ∧ tm-ipt-errorp (tm-interrupting-input-port (tm-ports (os)),
          tm-ports (os))))))
→ (tm-processor (os-running-input-handler-to-ibuffer-full-test (os),
  os-time-for-running-input-handler-to-resume-task2 (os))
  = os-running-input-handler-to-resume-task2 (os))

```

EVENT: Disable trace-running-input-handler-to-resume-task2.

DEFINITION:

```

os-time-for-running-input-handler-to-resume-task3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK3
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'8
(PUTNTH

```

```

(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
              (TM-IPORTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
 (ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
             (OS-READYQ OS))
'191
(PUTSEG
 (OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                    '0
                    '0
                    OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3133
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                  (TM-IPORTS OS)))
'2

```

```

(PUTNTH
  (PLUS '243
    (TIMES '8
      (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
  '3
  (PUTNTH
    (PLUS '247
      (PLUS (TIMES '8
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
        (GETNTH '1
          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
            (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
        (TIMES '2
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        '5
        (TM-REGS OS))))))
(TM-CC-VALUE
  (TM-ALU-INCR-MOD (GETNTH '1
    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
      (TABLE '8 (OS-IBUFFERS OS))))
    (GETNTH '3
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '8 (OS-IBUFFERS OS))))))
  '0
  (TM-SVCFLAG OS)
  (TM-SVCID OS)
  (TM-BASE OS)
  (TM-LIMIT OS)
  (TM-SLIMIT OS)
  '1
  '0
  (TM-CLOCK OS)
  (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    (TM-IPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-resume-task3

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os))),

```

```

                                os)))
      ∧ (¬ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                         tm-iports (os))))))
→ (tm-processor (os-running-input-handler-to-ibuffer-full-test (os),
                   os-time-for-running-input-handler-to-resume-task3 (os))
    = os-running-input-handler-to-resume-task3 (os))

```

EVENT: Disable trace-running-input-handler-to-resume-task3.

DEFINITION:

```

os-time-for-running-input-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-PATH1
  (OS)
  (TM
   (PUTNTH
    (GETNTH (TIMES '9
             (ARRAY-QFIRST (OS-READYQ OS)))
            (OS-NEW-TASK-TABLE OS))
    '0
   (PUTNTH
    (GETNTH (PLUS '1
                  (TIMES '9
                       (ARRAY-QFIRST (OS-READYQ OS))))
            (OS-NEW-TASK-TABLE OS))
    '1
   (PUTNTH
    (GETNTH (PLUS '8
                  (TIMES '9
                       (ARRAY-QFIRST (OS-READYQ OS))))
            (OS-NEW-TASK-TABLE OS))
    '2
   (PUTNTH
    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    '8
   (PUTNTH
    (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                     (TM-IPOINTS OS)))
    '9

```

```

(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)

```



```

(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-lpsw1

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                * tm-interrupting-input-port (tm-iports (os)),
                os-status-table (os))
        = '5))))
→ (tm-processor (os-running-input-handler-to-resume-task1 (os),
                 os-time-for-running-input-handler-to-lpsw (os))
  = os-running-input-handler-path1 (os))

```

EVENT: Disable trace-running-input-handler-to-lpsw1.

```

(DEFN
OS-RUNNING-INPUT-HANDLER-PATH2
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
            (TIMES '9
              (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
            (TIMES '9
              (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))

```

```

'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(TM-IPORTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)

```

```

(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))

(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-lpsw2

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                * tm-interrupting-input-port (tm-iports (os)),
                os-status-table (os))
        = '5))))
→ (tm-processor (os-running-input-handler-to-resume-task2 (os),
                  os-time-for-running-input-handler-to-lpsw (os))
  = os-running-input-handler-path2 (os))

```

EVENT: Disable trace-running-input-handler-to-lpsw2.

```

(DEFN
OS-RUNNING-INPUT-HANDLER-PATH3
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
          (OS-NEW-TASK-TABLE OS))
  '0
(PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
          (OS-NEW-TASK-TABLE OS))
  '1
(PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9

```

```

                (ARRAY-QFIRST (OS-READYQ OS))))
        (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(TM-IPORTS OS))))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)

```

```

(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))

(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-lpsw3

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                * tm-interrupting-input-port (tm-iports (os)),
                os-status-table (os))
        = '5))))
→ (tm-processor (os-running-input-handler-to-resume-task3 (os),
                 os-time-for-running-input-handler-to-lpsw (os))
  = os-running-input-handler-path3 (os))

```

EVENT: Disable trace-running-input-handler-to-lpsw3.

DEFINITION:

```

os-running-input-time1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick)

```

DEFINITION:

```

os-running-input-time2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick

```

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-running-input-handler-path1 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick)

```

THEOREM: rewrite-running-input-trace1

```

tm-processor (os-intended-input-interrupt (os),
              os-time-for-running-input-handler-path1 (os))
= tm-processor (tm-processor (os-intended-input-interrupt (os),
                             os-time-for-running-input-handler-thru-save-state (os)),
               os-running-input-time1 (os))

```

EVENT: Disable rewrite-running-input-trace1.

THEOREM: rewrite-running-input-trace2

```

tm-processor (os-running-input-handler-thru-save-state (os),
              os-running-input-time1 (os))
= tm-processor (tm-processor (os-running-input-handler-thru-save-state (os),
                             os-time-for-running-input-handler-to-ibuffer-full-test (os)),
               os-running-input-time2 (os))

```

EVENT: Disable rewrite-running-input-trace2.

THEOREM: rewrite-running-input-trace3

```

tm-processor (os-running-input-handler-to-ibuffer-full-test (os),
              os-running-input-time2 (os))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test (os),
                             os-time-for-running-input-handler-to-resume-task1 (os)),
               os-time-for-running-input-handler-to-lpsw (os))

```

EVENT: Disable rewrite-running-input-trace3.

THEOREM: trace-running-input-handler-path1

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ ((getnth ('2
                * tm-interrupting-input-port (tm-iports (os)),
                os-status-table (os))
          = '5)
        ^ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                          os))))))
→ (tm-processor (os-intended-input-interrupt (os),
               os-time-for-running-input-handler-path1 (os))
   = os-running-input-handler-path1 (os))

```

EVENT: Disable trace-running-input-handler-path1.

DEFINITION:

```

os-running-input-time3 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick)

```

DEFINITION:

```

os-running-input-time4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

```

os-time-for-running-input-handler-path2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

THEOREM: rewrite-running-input-trace4

$$\begin{aligned} & \text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-running-input-handler-path2} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-running-input-handler-thru-save-state} (os)), \\ & \quad \text{os-running-input-time3} (os)) \end{aligned}$$

EVENT: Disable rewrite-running-input-trace4.

THEOREM: rewrite-running-input-trace5

$$\begin{aligned} & \text{tm-processor} (\text{os-running-input-handler-thru-save-state} (os), \\ & \quad \text{os-running-input-time3} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-running-input-handler-thru-save-state} (os), \\ & \quad \text{os-time-for-running-input-handler-to-ibuffer-full-test} (os)), \\ & \quad \text{os-running-input-time4} (os)) \end{aligned}$$

EVENT: Disable rewrite-running-input-trace5.

THEOREM: rewrite-running-input-trace6

$$\begin{aligned} & \text{tm-processor} (\text{os-running-input-handler-to-ibuffer-full-test} (os), \\ & \quad \text{os-running-input-time4} (os)) \\ = & \text{tm-processor} (\text{tm-processor} (\text{os-running-input-handler-to-ibuffer-full-test} (os), \\ & \quad \text{os-time-for-running-input-handler-to-resume-task2} (os)), \\ & \quad \text{os-time-for-running-input-handler-to-lpsw} (os)) \end{aligned}$$

EVENT: Disable rewrite-running-input-trace6.

THEOREM: trace-running-input-handler-path2

$$\begin{aligned} & (\text{good-os} (os) \\ & \quad \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \quad \wedge ((\neg \text{tm-waiting} (os)) \\ & \quad \quad \quad \wedge ((\text{getnth} ('2 \\ & \quad \quad \quad \quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \quad \text{os-status-table} (os)) \\ & \quad \quad \quad = '5) \\ & \quad \quad \quad \wedge ((\neg \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \quad \wedge \text{tm-ipt-errorp} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \quad \text{tm-iports} (os))))))))) \\ \rightarrow & (\text{tm-processor} (\text{os-intended-input-interrupt} (os), \\ & \quad \text{os-time-for-running-input-handler-path2} (os)) \\ = & \text{os-running-input-handler-path2} (os)) \end{aligned}$$

EVENT: Disable trace-running-input-handler-path2.

DEFINITION:

os-running-input-time5 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

DEFINITION:

os-running-input-time6 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-running-input-handler-path3 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

THEOREM: rewrite-running-input-trace7

tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path3 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-thru-save-state (*os*)),
os-running-input-time5 (*os*))

EVENT: Disable rewrite-running-input-trace7.

THEOREM: rewrite-running-input-trace8

tm-processor (os-running-input-handler-thru-save-state (*os*),
os-running-input-time5 (*os*))
= tm-processor (tm-processor (os-running-input-handler-thru-save-state (*os*),

os-time-for-running-input-handler-to-ibuffer-full-test (*os*),
os-running-input-time6 (*os*)

EVENT: Disable rewrite-running-input-trace8.

THEOREM: rewrite-running-input-trace9
tm-processor (os-running-input-handler-to-ibuffer-full-test (*os*),
os-running-input-time6 (*os*))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test (*os*),
os-time-for-running-input-handler-to-resume-task3 (*os*)),
os-time-for-running-input-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-input-trace9.

THEOREM: trace-running-input-handler-path3
(good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
= '5)
 \wedge ((\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))
 \wedge (\neg tm-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*)))))))))
 \rightarrow (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path3 (*os*))
= os-running-input-handler-path3 (*os*))

EVENT: Disable trace-running-input-handler-path3.

DEFINITION:
os-time-for-running-input-handler-to-ibuffer-full-test2 (*os*)
= '(tick tick tick tick)

```
(DEFN OS-RUNNING-INPUT-HANDLER-TO-IBUFFER-FULL-TEST2
(OS)
(TM
(PUTNTH
(TM-PC OS)
```

```

'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3063

```

```

'3310
(TM-MEMORY OS))))))))))))))

(PUTNTH '3091
'0
(PUTNTH '3310
'1
(PUTNTH
(GETNTH '2 (TM-REGS OS))
'2
(PUTNTH (GETNTH '3 (TM-REGS OS))
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE
(GETNTH (TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(OS-STATUS-TABLE OS))
'5))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-ibuffer-full-test2

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
        * tm-interrupting-input-port (tm-iports (os)),
        os-status-table (os))
        ≠ '5))))

```

```

→ (tm-processor (os-running-input-handler-thru-save-state (os),
                 os-time-for-running-input-handler-to-ibuffer-full-test2 (os))
    = os-running-input-handler-to-ibuffer-full-test2 (os))

```

EVENT: Disable trace-running-input-handler-to-ibuffer-full-test2.

DEFINITION:

```

os-time-for-running-input-handler-to-resume-task4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK4
  (OS)
  (TM
   (PUTNTH
    (TM-PC OS)
    '0
   (PUTNTH
    (TM-SP OS)
    '1
   (PUTNTH
    (TM-PACK-PSW (TM-CC OS)
                  (TM-ERROR OS)
                  (TM-SVCFLAG OS)
                  (TM-SVCID OS))
    '2
   (PUTNTH
    (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
    '8
   (PUTNTH
    (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                     (TM-IPOINTS OS)))
    '9
   (PUTNTH '3231
    '10
   (PUTNTH
    (TM-R2 OS)
    '13
   (PUTNTH
    (TM-R3 OS)
    '14
   (PUTSEG

```

```

(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3112
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(DECR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
'4)))
'4

```

```

(PUTNTH (PLUS '211
          (TIMES '2
              (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        '5
        (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-PLUS
  (PLUS '243
    (PLUS
      (TIMES '8
        (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
      (DECR-MOD (GETNTH '1
                    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                          (TABLE '8 (OS-IBUFFERS OS))))
                '4)))
    '4))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                          (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-resume-task4

```

(good-os (os)
  ∧ (tm-some-input-interruptp (tm-iports (os))
    ∧ ((¬ tm-waiting (os))
      ∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                          os))))))
→ (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (os),
                    os-time-for-running-input-handler-to-resume-task4 (os))
  = os-running-input-handler-to-resume-task4 (os))

```

EVENT: Disable trace-running-input-handler-to-resume-task4.

DEFINITION:

os-time-for-running-input-handler-to-resume-task5 (os)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK5
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
```



```

(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3126
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(PLUS '256
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
(GETNTH '3

```

```

                                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                (TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                           (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-resume-task5

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                                    os)))
        ^ tm-iptort-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                            tm-iports (os))))))
→ (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (os),
                    os-time-for-running-input-handler-to-resume-task5 (os))
  = os-running-input-handler-to-resume-task5 (os))

```

EVENT: Disable trace-running-input-handler-to-resume-task5.

DEFINITION:

```

os-time-for-running-input-handler-to-resume-task6 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-INPUT-HANDLER-TO-RESUME-TASK6
  (OS)
  (TM
    (PUTNTH
      (TM-PC OS)
      '0
      (PUTNTH

```

```

(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3133
'3310
(TM-MEMORY OS))))))))))))))

```

```

(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS)))
'2
(PUTNTH
(PLUS '243
(TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'3
(PUTNTH
(PLUS '247
(PLUS (TIMES '8
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS)))
(GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD (GETNTH '1
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))
(GETNTH '3
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '8 (OS-IBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))

```

```
(TM-OPOINTS OS))
NIL)
```

THEOREM: trace-running-input-handler-to-resume-task6

```
(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os))
    ^ ((¬ tm-waiting (os))
      ^ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                                                    os)))
        ^ (¬ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                            tm-iports (os)))))))
  → (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (os),
                    os-time-for-running-input-handler-to-resume-task6 (os))
    = os-running-input-handler-to-resume-task6 (os))
```

EVENT: Disable trace-running-input-handler-to-resume-task6.

```
(DEFN
OS-RUNNING-INPUT-HANDLER-PATH4
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
'8
(PUTNTH
```

```

(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
              (TM-IPOINTS OS)))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-QREPLACED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))

(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))

```

```

(TM-IPORTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-lpsw4

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ (¬ tm-waiting (os))))
→ (tm-processor (os-running-input-handler-to-resume-task4 (os),
                 os-time-for-running-input-handler-to-lpsw (os))
    = os-running-input-handler-path4 (os))

```

EVENT: Disable trace-running-input-handler-to-lpsw4.

```

(DEFN
OS-RUNNING-INPUT-HANDLER-PATH5
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (ARRAY-QFIRST (OS-READYQ OS)))
(OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
          (TIMES '9
            (ARRAY-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
'8
(PUTNTH
(TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                  (TM-IPORTS OS)))
'9
(PUTNTH '3231

```

```

'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-OVERFLOW-CHARACTER OS)
'243
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```


THEOREM: trace-running-input-handler-to-lpsw5

(good-os (*os*)
 ^ (tm-some-input-interruptp (tm-iports (*os*)) ^ (¬ tm-waiting (*os*))))
 → (tm-processor (os-running-input-handler-to-resume-task5 (*os*),
 os-time-for-running-input-handler-to-lpsw (*os*))
 = os-running-input-handler-path5 (*os*))

EVENT: Disable trace-running-input-handler-to-lpsw5.

```
(DEFN
OS-RUNNING-INPUT-HANDLER-PATH6
(OS)
(TM
 (PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
           (OS-NEW-TASK-TABLE OS))
  '0
 (PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
  '1
 (PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
  '2
 (PUTNTH
  (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  '8
 (PUTNTH
  (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                   (TM-IPOINTS OS)))
  '9
 (PUTNTH '3231
  '10
 (PUTNTH
  (TM-R2 OS)
  '13
```

```

(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (OS-READYQ OS)
  '191
(PUTSEG
  (OS-STATUS-TABLE OS)
  '211
  (PUTSEG (OS-IBUFFERS-WITH-ENQUEUED-CHARACTER OS)
    '243
    (PUTSEG (OS-CODE OS)
      '2547
      (PUTNTH '2558
        '3309
        (PUTNTH '3238
          '3310
          (TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
  (TM-IPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-input-handler-to-lpsw6

```

(good-os (os)
  ^ (tm-some-input-interruptp (tm-iports (os)) ^ (¬ tm-waiting (os))))
→ (tm-processor (os-running-input-handler-to-resume-task6 (os),

```

$$\begin{aligned}
& \text{os-time-for-running-input-handler-to-lpsw}(os) \\
= & \text{os-running-input-handler-path6}(os)
\end{aligned}$$

EVENT: Disable trace-running-input-handler-to-lpsw6.

DEFINITION:

$$\begin{aligned}
& \text{os-running-input-time7}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-running-input-time8}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-running-input-handler-path4}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick)}
\end{aligned}$$

THEOREM: rewrite-running-input-trace10

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \quad \text{os-time-for-running-input-handler-path4}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \quad \text{os-time-for-running-input-handler-thru-save-state}(os)), \\
& \quad \text{os-running-input-time7}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace10.

THEOREM: rewrite-running-input-trace11

$$\text{tm-processor}(\text{os-running-input-handler-thru-save-state}(os),$$

$$\begin{aligned}
& \text{os-running-input-time7}(os) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-running-input-handler-thru-save-state}(os), \\
& \text{os-time-for-running-input-handler-to-ibuffer-full-test2}(os)), \\
& \text{os-running-input-time8}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace11.

$$\begin{aligned}
& \text{THEOREM: rewrite-running-input-trace12} \\
& \text{tm-processor}(\text{os-running-input-handler-to-ibuffer-full-test2}(os), \\
& \text{os-running-input-time8}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-running-input-handler-to-ibuffer-full-test2}(os), \\
& \text{os-time-for-running-input-handler-to-resume-task4}(os)), \\
& \text{os-time-for-running-input-handler-to-lpsw}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace12.

$$\begin{aligned}
& \text{THEOREM: trace-running-input-handler-path4} \\
& (\text{good-os}(os) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \neq '5) \\
& \quad \quad \wedge \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os))))))))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \text{os-time-for-running-input-handler-path4}(os)) \\
= & \text{os-running-input-handler-path4}(os))
\end{aligned}$$

EVENT: Disable trace-running-input-handler-path4.

$$\begin{aligned}
& \text{DEFINITION:} \\
& \text{os-running-input-time9}(os) \\
= & '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick)
\end{aligned}$$

$$\begin{aligned}
& \text{DEFINITION:} \\
& \text{os-running-input-time10}(os)
\end{aligned}$$

```

= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

DEFINITION:

os-time-for-running-input-handler-path5 (*os*)

```

= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

THEOREM: rewrite-running-input-trace13

```

tm-processor (os-intended-input-interrupt (os),
              os-time-for-running-input-handler-path5 (os))
= tm-processor (tm-processor (os-intended-input-interrupt (os),
                          os-time-for-running-input-handler-thru-save-state (os)),
               os-running-input-time9 (os))

```

EVENT: Disable rewrite-running-input-trace13.

THEOREM: rewrite-running-input-trace14

```

tm-processor (os-running-input-handler-thru-save-state (os),
              os-running-input-time9 (os))
= tm-processor (tm-processor (os-running-input-handler-thru-save-state (os),
                          os-time-for-running-input-handler-to-ibuffer-full-test2 (os)),
               os-running-input-time10 (os))

```

EVENT: Disable rewrite-running-input-trace14.

THEOREM: rewrite-running-input-trace15

```

tm-processor (os-running-input-handler-to-ibuffer-full-test2 (os),
              os-running-input-time10 (os))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (os),
                          os-time-for-running-input-handler-to-resume-task5 (os)),
               os-time-for-running-input-handler-to-lpsw (os))

```

EVENT: Disable rewrite-running-input-trace15.

THEOREM: trace-running-input-handler-path5

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \neq '5) \\
& \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os}))) \\
& \quad \quad \quad \wedge \text{tm-iport-errorp } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{tm-iports } (os)))))) \\
& \rightarrow (\text{tm-processor } (\text{os-intended-input-interrupt } (os), \\
& \quad \quad \text{os-time-for-running-input-handler-path5 } (os)) \\
& \quad = \text{os-running-input-handler-path5 } (os))
\end{aligned}$$

EVENT: Disable trace-running-input-handler-path5.

DEFINITION:

$$\begin{aligned}
& \text{os-running-input-time11 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-running-input-time12 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-running-input-handler-path6 } (os) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick)
\end{aligned}$$

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

```

THEOREM: rewrite-running-input-trace16

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \quad \text{os-time-for-running-input-handler-path6}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-input-interrupt}(os), \\
& \quad \text{os-time-for-running-input-handler-thru-save-state}(os)), \\
& \quad \text{os-running-input-time11}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace16.

THEOREM: rewrite-running-input-trace17

$$\begin{aligned}
& \text{tm-processor}(\text{os-running-input-handler-thru-save-state}(os), \\
& \quad \text{os-running-input-time11}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-running-input-handler-thru-save-state}(os), \\
& \quad \text{os-time-for-running-input-handler-to-ibuffer-full-test2}(os)), \\
& \quad \text{os-running-input-time12}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace17.

THEOREM: rewrite-running-input-trace18

$$\begin{aligned}
& \text{tm-processor}(\text{os-running-input-handler-to-ibuffer-full-test2}(os), \\
& \quad \text{os-running-input-time12}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-running-input-handler-to-ibuffer-full-test2}(os), \\
& \quad \text{os-time-for-running-input-handler-to-resume-task6}(os)), \\
& \quad \text{os-time-for-running-input-handler-to-lpsw}(os))
\end{aligned}$$

EVENT: Disable rewrite-running-input-trace18.

THEOREM: trace-running-input-handler-path6

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\text{getnth}('2 \\
& \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \text{os-status-table}(os)) \\
& \quad \neq '5) \\
& \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \text{os}))) \\
& \quad \wedge (\neg \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \text{tm-iports}(os))))))))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-input-interrupt}(os),
\end{aligned}$$

os-time-for-running-input-handler-path6 (*os*)
 = os-running-input-handler-path6 (*os*)

EVENT: Disable trace-running-input-handler-path6.

DEFINITION:

```
os-time-for-running-input-handler (os)
=  if getnth ('2 * tm-interrupting-input-port (tm-iports (os)),
            os-status-table (os))
    =  '5
    then if array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                             os))
        then os-time-for-running-input-handler-path1 (os)
        elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                             tm-iports (os))
        then os-time-for-running-input-handler-path2 (os)
        else os-time-for-running-input-handler-path3 (os) endif
    elseif array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                             os))
        then os-time-for-running-input-handler-path4 (os)
        elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                             tm-iports (os))
        then os-time-for-running-input-handler-path5 (os)
        else os-time-for-running-input-handler-path6 (os) endif
```

DEFINITION:

```
os-running-input-handler (os)
=  if getnth ('2 * tm-interrupting-input-port (tm-iports (os)),
            os-status-table (os))
    =  '5
    then if array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                             os))
        then os-running-input-handler-path1 (os)
        elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                             tm-iports (os))
        then os-running-input-handler-path2 (os)
        else os-running-input-handler-path3 (os) endif
    elseif array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
                                             os))
        then os-running-input-handler-path4 (os)
        elseif tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                             tm-iports (os))
        then os-running-input-handler-path5 (os)
        else os-running-input-handler-path6 (os) endif
```


THEOREM: trace-running-input-handler

$$\begin{aligned}
& (\text{good-os } (os) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \wedge (\neg \text{tm-waiting } (os)))) \\
\rightarrow & (\text{tm-processor } (\text{os-intended-input-interrupt } (os), \\
& \qquad \text{os-time-for-running-input-handler } (os)) \\
& = \text{os-running-input-handler } (os))
\end{aligned}$$

EVENT: Disable trace-running-input-handler.

DEFINITION:

os-intended-output-interrupt (tm)

$$\begin{aligned}
= & \text{tm-set-memory } (\text{putnth } (\text{tm-pc } (tm), \\
& \qquad '0, \\
& \qquad \text{putnth } (\text{tm-sp } (tm), \\
& \qquad \qquad '1, \\
& \qquad \qquad \text{putnth } (\text{tm-pack-psw } (\text{tm-cc } (tm), \\
& \qquad \qquad \qquad \text{tm-error } (tm), \\
& \qquad \qquad \qquad \text{tm-svcflag } (tm), \\
& \qquad \qquad \qquad \text{tm-svcid } (tm)), \\
& \qquad \qquad \qquad '2, \\
& \qquad \text{putnth } (\text{tm-interrupting-output-port } (\text{tm-oports } (tm)), \\
& \qquad \qquad '9, \\
& \qquad \text{putseg } (\text{os-task-table } (tm), \\
& \qquad \qquad '15, \\
& \qquad \text{putseg } (\text{os-segment-table } (tm), \\
& \qquad \qquad '159, \\
& \qquad \text{putseg } (\text{os-readyq } (tm), \\
& \qquad \qquad '191, \\
& \qquad \text{putseg } (\text{os-status-table } (tm), \\
& \qquad \qquad '211, \\
& \qquad \text{putseg } (\text{os-obuffers } (tm), \\
& \qquad \qquad '371, \\
& \qquad \text{putseg } (\text{os-code } (tm), \\
& \qquad \qquad '2547, \\
& \qquad \text{tm-memory } (tm))) \\
& \text{tm-set-pc } ('3135, \\
& \qquad \text{tm-set-sp } ('3310, \\
& \qquad \qquad \text{tm-set-svmode } ('1, \\
& \qquad \qquad \qquad \text{tm-set-error } ('0, \\
& \qquad \qquad \qquad \qquad \text{tm-set-rwstate } ('0, \\
& \qquad \qquad \qquad \qquad \qquad \text{tm-set-oports } (\text{tm-clear } (tm))))))
\end{aligned}$$

THEOREM: equivalence-of-output-interrupt-expressions

good-os (*os*)
→ (tm-execute-output-interrupt (*os*) = os-intended-output-interrupt (*os*))

EVENT: Disable equivalence-of-output-interrupt-expressions.

DEFINITION:

os-time-for-waiting-output-handler-thru-readyq-empty-test (*os*)
= '(tick tick tick tick tick tick tick)

```
(DEFN OS-WAITING-OUTPUT-HANDLER-THRU-READYQ-EMPTY-TEST
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
```

```

'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3146
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3156
'0
(PUTNTH '3310
'1
(PUTNTH '191 '3 (TM-REGS OS)))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-thru-readyq-empty-test
(good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)) \wedge tm-waiting (*os*)))
 \rightarrow (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-waiting-output-handler-thru-readyq-empty-test (*os*))
= os-waiting-output-handler-thru-readyq-empty-test (*os*))

EVENT: Disable trace-waiting-output-handler-thru-readyq-empty-test.

DEFINITION:
os-time-for-waiting-output-handler-to-obuffer-empty-test (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick)

```

(DEFN OS-WAITING-OUTPUT-HANDLER-TO-OBUFFER-EMPTY-TEST
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3178

```

```

'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3184
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'2
(PUTNTH '191
'3
(PUTNTH
(PLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS)))))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
'16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-obuffer-empty-test

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ (tm-waiting (os)
      ^ (getnth ('2
        * tm-interrupting-output-port (tm-oports (os)),
        os-status-table (os))
        = '4))))))

```

```

→ (tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
                 os-time-for-waiting-output-handler-to-obuffer-empty-test (os))
    = os-waiting-output-handler-to-obuffer-empty-test (os))

```

EVENT: Disable trace-waiting-output-handler-to-obuffer-empty-test.

DEFINITION:

```

os-time-for-waiting-output-handler-to-dispatcher1 (os)
= '(tick tick tick tick tick tick tick)

```

```

(DEFN OS-WAITING-OUTPUT-HANDLER-TO-DISPATCHER1
 (OS)
 (TM
  (PUTNTH
   (TM-PC OS)
   '0
  (PUTNTH
   (TM-SP OS)
   '1
  (PUTNTH
   (TM-PACK-PSW (TM-CC OS)
                 (TM-ERROR OS)
                 (TM-SVCFLAG OS)
                 (TM-SVCID OS))
   '2
  (PUTNTH
   (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
   '9
  (PUTNTH '3206
   '10
  (PUTNTH
   (TM-R3 OS)
   '14
  (PUTSEG
   (OS-TASK-TABLE OS)
   '15
  (PUTSEG
   (OS-SEGMENT-TABLE OS)
   '159
  (PUTSEG
   (ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
              (OS-READYQ OS))

```

```

'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3195
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'2
(PUTNTH
(PLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH
(PLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS)))))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0

```

```

(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                             (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-dispatcher1

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ (tm-waiting (os)
      ^ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                          os))))))
→ (tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),
                  os-time-for-waiting-output-handler-to-dispatcher1 (os))
  = os-waiting-output-handler-to-dispatcher1 (os))

```

EVENT: Disable trace-waiting-output-handler-to-dispatcher1.

DEFINITION:

```

os-time-for-waiting-output-handler-to-dispatcher2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

(DEFN OS-WAITING-OUTPUT-HANDLER-TO-DISPATCHER2

```

(OS)
(TM
  (PUTNTH
    (TM-PC OS)
    '0
  (PUTNTH
    (TM-SP OS)
    '1
  (PUTNTH
    (TM-PACK-PSW (TM-CC OS)
                  (TM-ERROR OS)
                  (TM-SVCFLAG OS)
                  (TM-SVCID OS))
    '2
  (PUTNTH
    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    '9
  (PUTNTH '3206

```



```

'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3204
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
(ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '8 (OS-OBUFFERS OS))))
'2
(PUTNTH
(PLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH
(PLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4

```

```

(PUTNTH (PLUS '211
          (TIMES '2
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
        '5
        (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD
  (GETNTH '0
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
             (TABLE '8 (OS-OBUFFERS OS))))
  (GETNTH '3
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
             (TABLE '8 (OS-OBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-START-OUTPUT
 (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                       (TABLE '8 (OS-OBUFFERS OS))))
 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
 (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-waiting-output-handler-to-dispatcher2

```

(good-os (os)
  ∧ (tm-some-output-interruptp (tm-oports (os))
    ∧ (tm-waiting (os)
      ∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                          os))))))
→ (tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),
                  os-time-for-waiting-output-handler-to-dispatcher2 (os))
  = os-waiting-output-handler-to-dispatcher2 (os))

```

EVENT: Disable trace-waiting-output-handler-to-dispatcher2.

DEFINITION:

os-time-for-waiting-output-handler-to-lpsw (os)

```
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick)
```

```
(DEFN OS-WAITING-OUTPUT-HANDLER-PATH1
(OS)
(TM
(PUTNTH
(GETNTH (TIMES '9
        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS)))
(OS-TASK-TABLE OS))
'0
(PUTNTH
(GETNTH (PLUS '1
        (TIMES '9
        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
        (TIMES '9
        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
```

```

'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3217
'3310
(TM-MEMORY OS)))))))))))))
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'4)
(QUOTIENT
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'256)
'4)
(QUOTIENT
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'512)
'256)
(QUOTIENT (GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))

```

```

'512)
(BASE (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
              (TABLE '2 (OS-SEGMENT-TABLE OS))))
(LIMIT (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
              (TABLE '2 (OS-SEGMENT-TABLE OS))))
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                           (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-lpsw1

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-output-handler-to-dispatcher1 (os),
                 os-time-for-waiting-output-handler-to-lpsw (os))
   = os-waiting-output-handler-path1 (os))

```

EVENT: Disable trace-waiting-output-handler-to-lpsw1.

```

(DEFN
OS-WAITING-OUTPUT-HANDLER-PATH2
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS)))
          (OS-TASK-TABLE OS))
  '0
(PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
          (OS-TASK-TABLE OS))
  '1
(PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9
                    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))

```

```

(OS-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3217
'3310
(TM-MEMORY OS)))))))))))))
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER
(GETNTH (PLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))
'4)
(QUOTIENT

```

```

(REMAINDER
  (GETNTH (PLUS '8
            (TIMES '9
                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
    (OS-TASK-TABLE OS))
  '256)
'4)
(QUOTIENT
 (REMAINDER
  (GETNTH (PLUS '8
            (TIMES '9
                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
    (OS-TASK-TABLE OS))
  '512)
'256)
(QUOTIENT (GETNTH (PLUS '8
                  (TIMES '9
                        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
    (OS-TASK-TABLE OS))
  '512)
(BASE (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TABLE '2 (OS-SEGMENT-TABLE OS))))
(LIMIT (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TABLE '2 (OS-SEGMENT-TABLE OS))))
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPOINTS OS)
(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-lpsw2

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-output-handler-to-dispatcher2 (os),
  os-time-for-waiting-output-handler-to-lpsw (os))
  = os-waiting-output-handler-path2 (os))

```

EVENT: Disable trace-waiting-output-handler-to-lpsw2.

```

DEFINITION:
os-time-for-waiting-output-handler-to-obuffer-empty-test2 (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-WAITING-OUTPUT-HANDLER-TO-OBUFFER-EMPTY-TEST2
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3146

```



```

'3310
(TM-MEMORY OS)))))))))))))

(PUTNTH '3184
'0
(PUTNTH '3310
'1
(PUTNTH '191
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-DIFFERENCE
(GETNTH (TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS)))
(OS-STATUS-TABLE OS))
'4))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-obuffer-empty-test2

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ (tm-waiting (os)
      ^ (getnth ('2
        * tm-interrupting-output-port (tm-oports (os)),
        os-status-table (os))
        ≠ '4))))
→ (tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
  os-time-for-waiting-output-handler-to-obuffer-empty-test2 (os))
  = os-waiting-output-handler-to-obuffer-empty-test2 (os))

```

EVENT: Disable trace-waiting-output-handler-to-obuffer-empty-test2.

DEFINITION:

os-time-for-waiting-output-handler-to-dispatcher3 (os)
= '(tick tick tick tick tick tick tick)

```
(DEFN OS-WAITING-OUTPUT-HANDLER-TO-DISPATCHER3
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
```

```

'2547
(PUTNTH '3195
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH (PLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-dispatcher3

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ (tm-waiting (os)
      ^ array-qemptyyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
        os))))))
→ (tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (os),
  os-time-for-waiting-output-handler-to-dispatcher3 (os))
  = os-waiting-output-handler-to-dispatcher3 (os))

```

EVENT: Disable trace-waiting-output-handler-to-dispatcher3.

DEFINITION:

```
os-time-for-waiting-output-handler-to-dispatcher4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick)
```

```
(DEFN OS-WAITING-OUTPUT-HANDLER-TO-DISPATCHER4
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
```

```

(PUTNTH '3204
      '3310
      (TM-MEMORY OS)))))))))))))
(PUTNTH '3206
'0
(PUTNTH '3310
'1
(PUTNTH
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                        (TABLE '8 (OS-OBUFFERS OS))))
'2
(PUTNTH
  (PLUS '371
    (TIMES '8
      (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH (PLUS '211
          (TIMES '2
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD
  (GETNTH '0
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
            (TABLE '8 (OS-OBUFFERS OS))))
  (GETNTH '3
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
            (TABLE '8 (OS-OBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                        (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))

```

NIL)

THEOREM: trace-waiting-output-handler-to-dispatcher4

(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
 ∧ (tm-waiting (*os*)
 ∧ (¬ array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
 os))))))
→ (tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (*os*),
 os-time-for-waiting-output-handler-to-dispatcher4 (*os*))
 = os-waiting-output-handler-to-dispatcher4 (*os*))

EVENT: Disable trace-waiting-output-handler-to-dispatcher4.

DEFINITION:

os-time-for-waiting-output-handler-to-wait (*os*)
= '(tick tick tick tick tick tick)

(DEFN OS-WAITING-OUTPUT-HANDLER-PATH3

(OS)
(TM
 (PUTNTH
 (TM-PC OS)
 '0
 (PUTNTH
 (TM-SP OS)
 '1
 (PUTNTH
 (TM-PACK-PSW (TM-CC OS)
 (TM-ERROR OS)
 (TM-SVCFLAG OS)
 (TM-SVCID OS))
 '2
 (PUTNTH
 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
 '9
 (PUTNTH '3206
 '10
 (PUTNTH
 (TM-R3 OS)
 '14
 (PUTSEG

```

(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3211
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH '191
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-wait1
(good-os (*os*))

\wedge (tm-some-output-interruptp (tm-oports (*os*)) \wedge tm-waiting (*os*))
 \rightarrow (tm-processor (os-waiting-output-handler-to-dispatcher3 (*os*),
os-time-for-waiting-output-handler-to-wait (*os*))
= os-waiting-output-handler-path3 (*os*))

EVENT: Disable trace-waiting-output-handler-to-wait1.

```

(DEFN
OS-WAITING-OUTPUT-HANDLER-PATH4
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3206
'10
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG (OS-READYQ OS)
'191
(PUTSEG (OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)

```



```

'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '3211
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3222
'0
(PUTNTH '3310
'1
(PUTNTH
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
'2
(PUTNTH '191
'3
(PUTNTH (PLUS '211
  (TIMES '2
    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'1
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-waiting-output-handler-to-wait2

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os)) ^ tm-waiting (os)))
→ (tm-processor (os-waiting-output-handler-to-dispatcher4 (os),
  os-time-for-waiting-output-handler-to-wait (os))

```

= os-waiting-output-handler-path4 (os))

EVENT: Disable trace-waiting-output-handler-to-wait2.

DEFINITION:

os-waiting-output-time1 (os)

= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-waiting-output-time2 (os)

= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-waiting-output-handler-path1 (os)

= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

THEOREM: rewrite-waiting-output-trace1

tm-processor (os-intended-output-interrupt (os),
os-time-for-waiting-output-handler-path1 (os))
= tm-processor (tm-processor (os-intended-output-interrupt (os),
os-time-for-waiting-output-handler-thru-readyq-empty-test (os)),
os-waiting-output-time1 (os))

EVENT: Disable rewrite-waiting-output-trace1.

THEOREM: rewrite-waiting-output-trace2

tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
os-waiting-output-time1 (os))
= tm-processor (tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
os-time-for-waiting-output-handler-to-obuffer-empty-test (os)),
os-waiting-output-time2 (os))

EVENT: Disable rewrite-waiting-output-trace2.

THEOREM: rewrite-waiting-output-trace3

$$\begin{aligned} & \text{tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),} \\ & \quad \text{os-waiting-output-time2 (os))} \\ = & \text{tm-processor (tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),} \\ & \quad \text{os-time-for-waiting-output-handler-to-dispatcher1 (os)),} \\ & \quad \text{os-time-for-waiting-output-handler-to-lpsw (os))} \end{aligned}$$

EVENT: Disable rewrite-waiting-output-trace3.

THEOREM: trace-waiting-output-handler-path1

$$\begin{aligned} & (\text{good-os (os)} \\ & \wedge (\text{tm-some-output-interruptp (tm-oports (os))} \\ & \quad \wedge (\text{tm-waiting (os)} \\ & \quad \quad \wedge ((\text{getnth ('2} \\ & \quad \quad \quad * \text{tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os-status-table (os))} \\ & \quad \quad = '4) \\ & \quad \quad \wedge \text{array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os)))))) \\ \rightarrow & (\text{tm-processor (os-intended-output-interrupt (os),} \\ & \quad \text{os-time-for-waiting-output-handler-path1 (os))} \\ = & \text{os-waiting-output-handler-path1 (os)} \end{aligned}$$

EVENT: Disable trace-waiting-output-handler-path1.

DEFINITION:

$$\begin{aligned} & \text{os-waiting-output-time3 (os)} \\ = & \text{'(tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick)} \end{aligned}$$

DEFINITION:

$$\begin{aligned} & \text{os-waiting-output-time4 (os)} \\ = & \text{'(tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick} \\ & \quad \text{tick tick tick tick tick tick tick tick tick tick)} \end{aligned}$$

```

tick tick tick tick tick tick tick tick tick tick
tick tick)

```

DEFINITION:

```

os-time-for-waiting-output-handler-path2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick)

```

THEOREM: rewrite-waiting-output-trace4

```

tm-processor (os-intended-output-interrupt (os),
              os-time-for-waiting-output-handler-path2 (os))
= tm-processor (tm-processor (os-intended-output-interrupt (os),
                             os-time-for-waiting-output-handler-thru-readyq-empty-test (os)),
               os-waiting-output-time3 (os))

```

EVENT: Disable rewrite-waiting-output-trace4.

THEOREM: rewrite-waiting-output-trace5

```

tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
              os-waiting-output-time3 (os))
= tm-processor (tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
                             os-time-for-waiting-output-handler-to-obuffer-empty-test (os)),
               os-waiting-output-time4 (os))

```

EVENT: Disable rewrite-waiting-output-trace5.

THEOREM: rewrite-waiting-output-trace6

```

tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),
              os-waiting-output-time4 (os))
= tm-processor (tm-processor (os-waiting-output-handler-to-obuffer-empty-test (os),
                             os-time-for-waiting-output-handler-to-dispatcher2 (os)),
               os-time-for-waiting-output-handler-to-lpsw (os))

```

EVENT: Disable rewrite-waiting-output-trace6.

THEOREM: trace-waiting-output-handler-path2

```

(good-os (os))

```

$$\begin{aligned}
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oport}(\text{os})) \\
& \quad \wedge (\text{tm-waiting}(\text{os}) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oport}(\text{os})), \\
& \quad \quad \quad \text{os-status-table}(\text{os})) \\
& \quad \quad = '4) \\
& \quad \quad \wedge (\neg \text{array-qemptyp}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oport}(\text{os})), \\
& \quad \quad \quad \text{os})))))) \\
& \rightarrow (\text{tm-processor}(\text{os-intended-output-interrupt}(\text{os}), \\
& \quad \quad \text{os-time-for-waiting-output-handler-path2}(\text{os})) \\
& \quad = \text{os-waiting-output-handler-path2}(\text{os}))
\end{aligned}$$

EVENT: Disable trace-waiting-output-handler-path2.

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-output-time5}(\text{os}) \\
& = '(tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-waiting-output-time6}(\text{os}) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick)
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-waiting-output-handler-path3}(\text{os}) \\
& = '(tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick tick tick tick tick tick \\
& \quad tick tick tick tick tick)
\end{aligned}$$

THEOREM: rewrite-waiting-output-trace7

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-output-interrupt}(\text{os}), \\
& \quad \text{os-time-for-waiting-output-handler-path3}(\text{os})) \\
& = \text{tm-processor}(\text{tm-processor}(\text{os-intended-output-interrupt}(\text{os}), \\
& \quad \quad \text{os-time-for-waiting-output-handler-thru-readyq-empty-test}(\text{os})), \\
& \quad \quad \text{os-waiting-output-time5}(\text{os}))
\end{aligned}$$

EVENT: Disable rewrite-waiting-output-trace7.

THEOREM: rewrite-waiting-output-trace8

$$\begin{aligned}
& \text{tm-processor}(\text{os-waiting-output-handler-thru-readyq-empty-test}(\text{os}), \\
& \quad \text{os-waiting-output-time5}(\text{os})) \\
& = \text{tm-processor}(\text{tm-processor}(\text{os-waiting-output-handler-thru-readyq-empty-test}(\text{os}), \\
& \quad \quad \text{os-time-for-waiting-output-handler-to-obuffer-empty-test2}(\text{os})), \\
& \quad \quad \text{os-waiting-output-time6}(\text{os}))
\end{aligned}$$

EVENT: Disable rewrite-waiting-output-trace8.

THEOREM: rewrite-waiting-output-trace9
tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (os),
os-waiting-output-time6 (os))
= tm-processor (tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (os),
os-time-for-waiting-output-handler-to-dispatcher3 (os)),
os-time-for-waiting-output-handler-to-wait (os))

EVENT: Disable rewrite-waiting-output-trace9.

THEOREM: trace-waiting-output-handler-path3
(good-os (os)
∧ (tm-some-output-interruptp (tm-oports (os))
∧ (tm-waiting (os)
∧ ((getnth ('2
* tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
≠ '4)
∧ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
os))))))
→ (tm-processor (os-intended-output-interrupt (os),
os-time-for-waiting-output-handler-path3 (os))
= os-waiting-output-handler-path3 (os))

EVENT: Disable trace-waiting-output-handler-path3.

DEFINITION:
os-waiting-output-time7 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

DEFINITION:
os-waiting-output-time8 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

DEFINITION:
os-time-for-waiting-output-handler-path4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

```

tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

```

THEOREM: rewrite-waiting-output-trace10
tm-processor (os-intended-output-interrupt (*os*),
os-time-for-waiting-output-handler-path4 (*os*))
= tm-processor (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-waiting-output-handler-thru-readyq-empty-test (*os*)),
os-waiting-output-time7 (*os*))

EVENT: Disable rewrite-waiting-output-trace10.

THEOREM: rewrite-waiting-output-trace11
tm-processor (os-waiting-output-handler-thru-readyq-empty-test (*os*),
os-waiting-output-time7 (*os*))
= tm-processor (tm-processor (os-waiting-output-handler-thru-readyq-empty-test (*os*),
os-time-for-waiting-output-handler-to-obuffer-empty-test2 (*os*)),
os-waiting-output-time8 (*os*))

EVENT: Disable rewrite-waiting-output-trace11.

THEOREM: rewrite-waiting-output-trace12
tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (*os*),
os-waiting-output-time8 (*os*))
= tm-processor (tm-processor (os-waiting-output-handler-to-obuffer-empty-test2 (*os*),
os-time-for-waiting-output-handler-to-dispatcher4 (*os*)),
os-time-for-waiting-output-handler-to-wait (*os*))

EVENT: Disable rewrite-waiting-output-trace12.

THEOREM: trace-waiting-output-handler-path4
(good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge (tm-waiting (*os*)
 \wedge ((getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
 \neq '4)
 \wedge (\neg array-qemptyyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))))))
 \rightarrow (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-waiting-output-handler-path4 (*os*))
= os-waiting-output-handler-path4 (*os*))

EVENT: Disable trace-waiting-output-handler-path4.

DEFINITION:

```
os-time-for-waiting-output-handler (os)
=  if getnth ('2 * tm-interrupting-output-port (tm-oports (os)),
             os-status-table (os))
   =  '4
   then if array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                           os))
        then os-time-for-waiting-output-handler-path1 (os)
        else os-time-for-waiting-output-handler-path2 (os) endif
   elseif array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                           os))
        then os-time-for-waiting-output-handler-path3 (os)
        else os-time-for-waiting-output-handler-path4 (os) endif
```

DEFINITION:

```
os-waiting-output-handler (os)
=  if getnth ('2 * tm-interrupting-output-port (tm-oports (os)),
             os-status-table (os))
   =  '4
   then if array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                           os))
        then os-waiting-output-handler-path1 (os)
        else os-waiting-output-handler-path2 (os) endif
   elseif array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                           os))
        then os-waiting-output-handler-path3 (os)
        else os-waiting-output-handler-path4 (os) endif
```

THEOREM: trace-waiting-output-handler

```
(good-os (os)
   $\wedge$  (tm-some-output-interruptp (tm-oports (os))  $\wedge$  tm-waiting (os)))
 $\rightarrow$  (tm-processor (os-intended-output-interrupt (os),
                  os-time-for-waiting-output-handler (os))
  =  os-waiting-output-handler (os))
```

EVENT: Disable trace-waiting-output-handler.

DEFINITION:

```
os-time-for-running-output-handler-thru-save-state (os)
=  '(tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick tick)
```


tick tick tick tick)

```
(DEFN OS-RUNNING-OUTPUT-HANDLER-THRU-SAVE-STATE
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
```

```

(PUTSEG (OS-CODE OS)
  '2547
  (PUTNTH '2558
    '3309
    (PUTNTH '3156
      '3310
      (TM-MEMORY OS))))))))))))))
(PUTNTH '3156
  '0
  (PUTNTH '3310
    '1
    (PUTNTH (GETNTH '2 (TM-REGS OS))
      '2
      (PUTNTH (GETNTH '3 (TM-REGS OS))
        '3
        (TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-PLUS (PLUS '15
  (TIMES '9
    (ARRAY-QFIRST (OS-READYQ OS))))
  '8))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-thru-save-state

$$\begin{aligned}
& \text{(good-os } (os) \\
& \wedge \text{ (tm-some-output-interruptp (tm-oports } (os)) \wedge (\neg \text{tm-waiting } (os)))) \\
\rightarrow & \text{ (tm-processor (os-intended-output-interrupt } (os), \\
& \quad \text{os-time-for-running-output-handler-thru-save-state } (os)) \\
& = \text{ os-running-output-handler-thru-save-state } (os))
\end{aligned}$$

EVENT: Disable trace-running-output-handler-thru-save-state.

DEFINITION:

```

os-time-for-running-output-handler-to-obuffer-empty-test (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-OBUFFER-EMPTY-TEST
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))

```

```

'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3178
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3184
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'2
(PUTNTH '191
'3
(PUTNTH
(PPLUS '195
(GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE (TM-ALU-INCR-MOD (GETNTH '1 (OS-READYQ OS))
'16))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))

```

```

                                (TM-OPOINTS OS))
NIL)

THEOREM: trace-running-output-handler-to-obuffer-empty-test
(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                * tm-interrupting-output-port (tm-oports (os)),
                os-status-table (os))
        = '4))))
→ (tm-processor (os-running-output-handler-thru-save-state (os),
                 os-time-for-running-output-handler-to-obuffer-empty-test (os))
  = os-running-output-handler-to-obuffer-empty-test (os))

```

EVENT: Disable trace-running-output-handler-to-obuffer-empty-test.

DEFINITION:

```

os-time-for-running-output-handler-to-resume-task1 (os)
= '(tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-RESUME-TASK1
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH

```

```

(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3195
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'2
(PUTNTH
(PLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH
(PLUS '195

```

```

      (GETNTH '1 (OS-READYQ OS))
    '4
    (PUTNTH (PLUS '211
      (TIMES '2
        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
      '5
      (TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-resume-task1

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ array-qemptyyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
        os))))))
→ (tm-processor (os-running-output-handler-to-obuffer-empty-test (os),
  os-time-for-running-output-handler-to-resume-task1 (os))
  = os-running-output-handler-to-resume-task1 (os))

```

EVENT: Disable trace-running-output-handler-to-resume-task1.

DEFINITION:

```

os-time-for-running-output-handler-to-resume-task2 (os)
= '(tick tick tick tick tick tick tick tick tick tick
  tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-RESUME-TASK2
  (OS)
  (TM

```

```

(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558

```



```

'3309
(PUTNTH '3204
'3310
(TM-MEMORY OS)))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
'2
(PUTNTH
  (PLUS '371
    (TIMES '8
      (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH
  (PLUS '195
    (GETNTH '1 (OS-READYQ OS)))
'4
(PUTNTH (PLUS '211
  (TIMES '2
    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS)))))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD
  (GETNTH '0
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
      (TABLE '8 (OS-OBUFFERS OS))))
  (GETNTH '3
    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
      (TABLE '8 (OS-OBUFFERS OS))))))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)

```

```

(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                        (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-resume-task2

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                          os))))))
→ (tm-processor (os-running-output-handler-to-obuffer-empty-test (os),
                    os-time-for-running-output-handler-to-resume-task2 (os))
  = os-running-output-handler-to-resume-task2 (os))

```

EVENT: Disable trace-running-output-handler-to-resume-task2.

DEFINITION:

```

os-time-for-running-output-handler-to-lpsw (os)
= '(tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick tick tick
   tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-PATH1
  (OS)
  (TM
    (PUTNTH
      (GETNTH (TIMES '9
                (ARRAY-QFIRST (OS-READYQ OS)))
              (OS-NEW-TASK-TABLE OS))
      '0
      (PUTNTH
        (GETNTH (PLUS '1
                      (TIMES '9
                        (ARRAY-QFIRST (OS-READYQ OS))))
                  (OS-NEW-TASK-TABLE OS))
        '1
        (PUTNTH
          (GETNTH (PLUS '8
                        (TIMES '9

```

```

                                (ARRAY-QFIRST (OS-READYQ OS)))
                                (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)

```

```

(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                             (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-running-output-handler-to-lpsw1

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
                 * tm-interrupting-output-port (tm-oports (os)),
                 os-status-table (os))
          = '4))))
→ (tm-processor (os-running-output-handler-to-resume-task1 (os),
                  os-time-for-running-output-handler-to-lpsw (os))
  = os-running-output-handler-path1 (os))

```

EVENT: Disable trace-running-output-handler-to-lpsw1.

```

(DEFN
OS-RUNNING-OUTPUT-HANDLER-PATH2
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
          (OS-NEW-TASK-TABLE OS))
  '0
(PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
          (OS-NEW-TASK-TABLE OS))
  '1
(PUTNTH
  (GETNTH (PLUS '8

```

```

                (TIMES '9
                (ARRAY-QFIRST (OS-READYQ OS))))
                (OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(ARRAY-ENQ (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(OS-READYQ OS))
'191
(PUTSEG
(OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'0
'0
OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)

```

```

(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-running-output-handler-to-lpsw2

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ (getnth ('2
        * tm-interrupting-output-port (tm-oports (os)),
        os-status-table (os))
        = '4))))
→ (tm-processor (os-running-output-handler-to-resume-task2 (os),
  os-time-for-running-output-handler-to-lpsw (os))
  = os-running-output-handler-path2 (os))

```

EVENT: Disable trace-running-output-handler-to-lpsw2.

DEFINITION:

```

os-time-for-running-output-handler-to-obuffer-empty-test2 (os)
= '(tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-OBUFFER-EMPTY-TEST2
  (OS)
  (TM
    (PUTNTH
      (TM-PC OS)
      '0
    (PUTNTH
      (TM-SP OS)
      '1

```

```

(PUTNTH
  (TM-PACK-PSW (TM-CC OS)
    (TM-ERROR OS)
    (TM-SVCFLAG OS)
    (TM-SVCID OS))
  '2
(PUTNTH
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  '9
(PUTNTH '3231
  '10
(PUTNTH
  (TM-R2 OS)
  '13
(PUTNTH
  (TM-R3 OS)
  '14
(PUTSEG
  (OS-NEW-TASK-TABLE OS)
  '15
(PUTSEG
  (OS-SEGMENT-TABLE OS)
  '159
(PUTSEG
  (OS-READYQ OS)
  '191
(PUTSEG
  (OS-STATUS-TABLE OS)
  '211
(PUTSEG (OS-OBUFFERS OS)
  '371
(PUTSEG (OS-CODE OS)
  '2547
(PUTNTH '2558
  '3309
(PUTNTH '3156
  '3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3184
  '0
(PUTNTH '3310
  '1
(PUTNTH
  (GETNTH '2 (TM-REGS OS))

```

```

'2
(PUTNTH
  (GETNTH '3 (TM-REGS OS))
  '3
  (PUTNTH (PLUS '211
    (TIMES '2
      (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
    '5
    (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-DIFFERENCE
  (GETNTH (TIMES '2
    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS)))
    (OS-STATUS-TABLE OS))
  '4))
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-obuffer-empty-test2

```

(good-os (os)
  ∧ (tm-some-output-interruptp (tm-oports (os))
    ∧ ((¬ tm-waiting (os))
      ∧ (getnth ('2
        * tm-interrupting-output-port (tm-oports (os)),
        os-status-table (os))
        ≠ '4))))
→ (tm-processor (os-running-output-handler-thru-save-state (os),
  os-time-for-running-output-handler-to-obuffer-empty-test2 (os))
  = os-running-output-handler-to-obuffer-empty-test2 (os))

```

EVENT: Disable trace-running-output-handler-to-obuffer-empty-test2.

DEFINITION:


```
os-time-for-running-output-handler-to-resume-task3(os)
= '(tick tick tick tick tick tick tick)
```

```
(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-RESUME-TASK3
(OS)
(TM
(PUTNTH
(TM-PC OS)
'0
(PUTNTH
(TM-SP OS)
'1
(PUTNTH
(TM-PACK-PSW (TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
```

```

'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3195
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(GETNTH '2 (TM-REGS OS))
'2
(PUTNTH
(PPLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
'1
'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPORTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-resume-task3
(good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((\neg tm-waiting (*os*)))

```

      ∧ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oport (os)),
                                          os))))))
→ (tm-processor (os-running-output-handler-to-obuffer-empty-test2 (os),
                      os-time-for-running-output-handler-to-resume-task3 (os))
    = os-running-output-handler-to-resume-task3 (os))

```

EVENT: Disable trace-running-output-handler-to-resume-task3.

DEFINITION:

```

os-time-for-running-output-handler-to-resume-task4 (os)
= '(tick tick tick tick tick tick tick tick tick tick
    tick tick tick tick tick tick tick tick tick)

```

```

(DEFN OS-RUNNING-OUTPUT-HANDLER-TO-RESUME-TASK4
  (OS)
  (TM
    (PUTNTH
      (TM-PC OS)
      '0
    (PUTNTH
      (TM-SP OS)
      '1
    (PUTNTH
      (TM-PACK-PSW (TM-CC OS)
                    (TM-ERROR OS)
                    (TM-SVCFLAG OS)
                    (TM-SVCID OS))
      '2
    (PUTNTH
      (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
      '9
    (PUTNTH '3231
      '10
    (PUTNTH
      (TM-R2 OS)
      '13
    (PUTNTH
      (TM-R3 OS)
      '14
    (PUTSEG
      (OS-NEW-TASK-TABLE OS)
      '15

```

```

(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3204
'3310
(TM-MEMORY OS))))))))))))))
(PUTNTH '3231
'0
(PUTNTH '3310
'1
(PUTNTH
(ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '8 (OS-OBUFFERS OS))))
'2
(PUTNTH
(PLUS '371
(TIMES '8
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'3
(PUTNTH (PLUS '211
(TIMES '2
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
'5
(TM-REGS OS))))))
(TM-CC-VALUE
(TM-ALU-INCR-MOD
(GETNTH '0
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '8 (OS-OBUFFERS OS))))
(GETNTH '3
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '8 (OS-OBUFFERS OS))))))

```

```

'0
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'1
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-resume-task4

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os))
    ^ ((¬ tm-waiting (os))
      ^ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
        os))))))
→ (tm-processor (os-running-output-handler-to-obuffer-empty-test2 (os),
  os-time-for-running-output-handler-to-resume-task4 (os))
  = os-running-output-handler-to-resume-task4 (os))

```

EVENT: Disable trace-running-output-handler-to-resume-task4.

```

(DEFN
OS-RUNNING-OUTPUT-HANDLER-PATH3
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
    (ARRAY-QFIRST (OS-READYQ OS)))
    (OS-NEW-TASK-TABLE OS))
'0
(PUTNTH
  (GETNTH (PLUS '1
    (TIMES '9
      (ARRAY-QFIRST (OS-READYQ OS))))

```

```

(OS-NEW-TASK-TABLE OS))
'1
(PUTNTH
(GETNTH (PLUS '8
(TIMES '9
(Array-QFIRST (OS-READYQ OS))))
(OS-NEW-TASK-TABLE OS))
'2
(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)

```

```

(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPOINTS OS)
(TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                             (TM-OPOINTS OS)))
NIL)

```

THEOREM: trace-running-output-handler-to-lpsw3

```

(good-os (os)
  ^ (tm-some-output-interruptp (tm-oports (os)) ^ (¬ tm-waiting (os))))
→ (tm-processor (os-running-output-handler-to-resume-task3 (os),
                  os-time-for-running-output-handler-to-lpsw (os))
   = os-running-output-handler-path3 (os))

```

EVENT: Disable trace-running-output-handler-to-lpsw3.

```

(DEFN
OS-RUNNING-OUTPUT-HANDLER-PATH4
(OS)
(TM
(PUTNTH
  (GETNTH (TIMES '9
           (ARRAY-QFIRST (OS-READYQ OS)))
           (OS-NEW-TASK-TABLE OS))
  '0
(PUTNTH
  (GETNTH (PLUS '1
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
  '1
(PUTNTH
  (GETNTH (PLUS '8
               (TIMES '9
                    (ARRAY-QFIRST (OS-READYQ OS))))
           (OS-NEW-TASK-TABLE OS))
  '2

```

```

(PUTNTH
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
'9
(PUTNTH '3231
'10
(PUTNTH
(TM-R2 OS)
'13
(PUTNTH
(TM-R3 OS)
'14
(PUTSEG
(OS-NEW-TASK-TABLE OS)
'15
(PUTSEG
(OS-SEGMENT-TABLE OS)
'159
(PUTSEG
(OS-READYQ OS)
'191
(PUTSEG
(OS-STATUS-TABLE OS)
'211
(PUTSEG (OS-OBUFFERS-WITH-DEQUEUED-CHARACTER OS)
'371
(PUTSEG (OS-CODE OS)
'2547
(PUTNTH '2558
'3309
(PUTNTH '3238
'3310
(TM-MEMORY OS))))))))))))))
(TM-REGS OS)
(TM-CC OS)
(TM-ERROR OS)
(TM-SVCFLAG OS)
(TM-SVCID OS)
(TM-BASE OS)
(TM-LIMIT OS)
(TM-SLIMIT OS)
'0
'0
(TM-CLOCK OS)
(TM-IPORTS OS)

```



```

(TM-START-OUTPUT
  (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
    (TABLE '8 (OS-OBUFFERS OS))))
  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
  (TM-OPOINTS OS))
NIL)

```

THEOREM: trace-running-output-handler-to-lpsw4
 (good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)) \wedge (\neg tm-waiting (*os*))))
 \rightarrow (tm-processor (os-running-output-handler-to-resume-task4 (*os*),
 os-time-for-running-output-handler-to-lpsw (*os*))
 = os-running-output-handler-path4 (*os*))

EVENT: Disable trace-running-output-handler-to-lpsw4.

DEFINITION:
 os-running-output-time1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick)

DEFINITION:
 os-running-output-time2 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick)

DEFINITION:
 os-time-for-running-output-handler-path1 (*os*)
 = '(tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick tick tick tick tick
 tick tick tick tick tick tick)

THEOREM: rewrite-running-output-trace1

$$\begin{aligned} & \text{tm-processor (os-intended-output-interrupt (os),} \\ & \quad \text{os-time-for-running-output-handler-path1 (os))} \\ = & \text{tm-processor (tm-processor (os-intended-output-interrupt (os),} \\ & \quad \text{os-time-for-running-output-handler-thru-save-state (os)),} \\ & \quad \text{os-running-output-time1 (os))} \end{aligned}$$

EVENT: Disable rewrite-running-output-trace1.

THEOREM: rewrite-running-output-trace2

$$\begin{aligned} & \text{tm-processor (os-running-output-handler-thru-save-state (os),} \\ & \quad \text{os-running-output-time1 (os))} \\ = & \text{tm-processor (tm-processor (os-running-output-handler-thru-save-state (os),} \\ & \quad \text{os-time-for-running-output-handler-to-obuffer-empty-test (os)),} \\ & \quad \text{os-running-output-time2 (os))} \end{aligned}$$

EVENT: Disable rewrite-running-output-trace2.

THEOREM: rewrite-running-output-trace3

$$\begin{aligned} & \text{tm-processor (os-running-output-handler-to-obuffer-empty-test (os),} \\ & \quad \text{os-running-output-time2 (os))} \\ = & \text{tm-processor (tm-processor (os-running-output-handler-to-obuffer-empty-test (os),} \\ & \quad \text{os-time-for-running-output-handler-to-resume-task1 (os)),} \\ & \quad \text{os-time-for-running-output-handler-to-lpsw (os))} \end{aligned}$$

EVENT: Disable rewrite-running-output-trace3.

THEOREM: trace-running-output-handler-path1

$$\begin{aligned} & (\text{good-os (os)} \\ & \wedge (\text{tm-some-output-interruptp (tm-oports (os))} \\ & \quad \wedge ((\neg \text{tm-waiting (os)}) \\ & \quad \quad \wedge ((\text{getnth ('2} \\ & \quad \quad \quad * \text{tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os-status-table (os))} \\ & \quad \quad = '4) \\ & \quad \wedge \text{array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os))))) \\ \rightarrow & (\text{tm-processor (os-intended-output-interrupt (os),} \\ & \quad \text{os-time-for-running-output-handler-path1 (os))} \\ = & \text{os-running-output-handler-path1 (os))} \end{aligned}$$

EVENT: Disable trace-running-output-handler-path1.

DEFINITION:

os-running-output-time3 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

DEFINITION:

os-running-output-time4 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick)

DEFINITION:

os-time-for-running-output-handler-path2 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

THEOREM: rewrite-running-output-trace4

tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path2 (*os*))
= tm-processor (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-thru-save-state (*os*)),
os-running-output-time3 (*os*))

EVENT: Disable rewrite-running-output-trace4.

THEOREM: rewrite-running-output-trace5

tm-processor (os-running-output-handler-thru-save-state (*os*),
os-running-output-time3 (*os*))
= tm-processor (tm-processor (os-running-output-handler-thru-save-state (*os*),

os-time-for-running-output-handler-to-obuffer-empty-test (*os*),
os-running-output-time4 (*os*)

EVENT: Disable rewrite-running-output-trace5.

THEOREM: rewrite-running-output-trace6
tm-processor (os-running-output-handler-to-obuffer-empty-test (*os*),
os-running-output-time4 (*os*))
= tm-processor (tm-processor (os-running-output-handler-to-obuffer-empty-test (*os*),
os-time-for-running-output-handler-to-resume-task2 (*os*)),
os-time-for-running-output-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-output-trace6.

THEOREM: trace-running-output-handler-path2
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((¬ tm-waiting (*os*))
∧ ((getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
= '4)
∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))))))
→ (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path2 (*os*))
= os-running-output-handler-path2 (*os*))

EVENT: Disable trace-running-output-handler-path2.

DEFINITION:
os-running-output-time5 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick)

DEFINITION:
os-running-output-time6 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick)

DEFINITION:

os-time-for-running-output-handler-path3 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick tick
tick tick tick tick)

THEOREM: rewrite-running-output-trace7

tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path3 (*os*))
= tm-processor (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-thru-save-state (*os*)),
os-running-output-time5 (*os*))

EVENT: Disable rewrite-running-output-trace7.

THEOREM: rewrite-running-output-trace8

tm-processor (os-running-output-handler-thru-save-state (*os*),
os-running-output-time5 (*os*))
= tm-processor (tm-processor (os-running-output-handler-thru-save-state (*os*),
os-time-for-running-output-handler-to-obuffer-empty-test2 (*os*)),
os-running-output-time6 (*os*))

EVENT: Disable rewrite-running-output-trace8.

THEOREM: rewrite-running-output-trace9

tm-processor (os-running-output-handler-to-obuffer-empty-test2 (*os*),
os-running-output-time6 (*os*))
= tm-processor (tm-processor (os-running-output-handler-to-obuffer-empty-test2 (*os*),
os-time-for-running-output-handler-to-resume-task3 (*os*)),
os-time-for-running-output-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-output-trace9.

THEOREM: trace-running-output-handler-path3

(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((¬ tm-waiting (*os*))
∧ ((getnth ('2

$$\begin{aligned}
& * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \text{os-status-table}(os) \\
\neq & \text{'4)} \\
\wedge & \text{array-qemptyp}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& os)))))) \\
\rightarrow & (\text{tm-processor}(\text{os-intended-output-interrupt}(os), \\
& \text{os-time-for-running-output-handler-path3}(os)) \\
= & \text{os-running-output-handler-path3}(os)
\end{aligned}$$

EVENT: Disable trace-running-output-handler-path3.

DEFINITION:

$$\begin{aligned}
& \text{os-running-output-time7}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-running-output-time8}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick)}
\end{aligned}$$

DEFINITION:

$$\begin{aligned}
& \text{os-time-for-running-output-handler-path4}(os) \\
= & \text{'(tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick tick tick tick tick} \\
& \text{tick tick tick tick tick tick)}
\end{aligned}$$

THEOREM: rewrite-running-output-trace10

$$\begin{aligned}
& \text{tm-processor}(\text{os-intended-output-interrupt}(os), \\
& \text{os-time-for-running-output-handler-path4}(os)) \\
= & \text{tm-processor}(\text{tm-processor}(\text{os-intended-output-interrupt}(os),
\end{aligned}$$

os-time-for-running-output-handler-thru-save-state (*os*),
os-running-output-time7 (*os*)

EVENT: Disable rewrite-running-output-trace10.

THEOREM: rewrite-running-output-trace11
tm-processor (os-running-output-handler-thru-save-state (*os*),
os-running-output-time7 (*os*))
= tm-processor (tm-processor (os-running-output-handler-thru-save-state (*os*),
os-time-for-running-output-handler-to-obuffer-empty-test2 (*os*)),
os-running-output-time8 (*os*))

EVENT: Disable rewrite-running-output-trace11.

THEOREM: rewrite-running-output-trace12
tm-processor (os-running-output-handler-to-obuffer-empty-test2 (*os*),
os-running-output-time8 (*os*))
= tm-processor (tm-processor (os-running-output-handler-to-obuffer-empty-test2 (*os*),
os-time-for-running-output-handler-to-resume-task4 (*os*)),
os-time-for-running-output-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-output-trace12.

THEOREM: trace-running-output-handler-path4
(good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
 \neq '4)
 \wedge (\neg array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))))))
 \rightarrow (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path4 (*os*))
= os-running-output-handler-path4 (*os*))

EVENT: Disable trace-running-output-handler-path4.

DEFINITION:
os-time-for-running-output-handler (*os*)
= **if** getnth ('2 * tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))

```

= '4
then if array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                         os))
    then os-time-for-running-output-handler-path1 (os)
    else os-time-for-running-output-handler-path2 (os) endif
elseif array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                         os))
    then os-time-for-running-output-handler-path3 (os)
    else os-time-for-running-output-handler-path4 (os) endif

```

DEFINITION:

```

os-running-output-handler (os)
= if getnth ('2 * tm-interrupting-output-port (tm-oports (os)),
             os-status-table (os))
  = '4
  then if array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                         os))
    then os-running-output-handler-path1 (os)
    else os-running-output-handler-path2 (os) endif
  elseif array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                         os))
    then os-running-output-handler-path3 (os)
    else os-running-output-handler-path4 (os) endif

```

THEOREM: trace-running-output-handler

```

(good-os (os)
  ∧ (tm-some-output-interruptp (tm-oports (os)) ∧ (¬ tm-waiting (os))))
→ (tm-processor (os-intended-output-interrupt (os),
                os-time-for-running-output-handler (os))
   = os-running-output-handler (os))

```

EVENT: Disable trace-running-output-handler.

EVENT: Let us define the theory *access-os-memory* to consist of the following events: getnth-before-putseg, getnth-putnth-non-interference, getseg-after-putnth, getseg-after-putseg, getseg-before-putnth, getseg-before-putseg, getseg-for-length, getseg-getseg, getseg-within-putseg, length-array-deq, length-array-enq, length-getseg, length-putnth, length-putseg, plistp-array-deq, plistp-array-enq, plistp-getseg, plistp-putnth, plistp-putseg.

THEOREM: good-tm-os-intended-clock-interrupt

```

(good-os (os) ∧ ((¬ tm-waiting (os)) ∧ (¬ tm-errorp (os))))
→ good-tm (os-intended-clock-interrupt (os))

```


EVENT: Disable good-tm-os-intended-clock-interrupt.

THEOREM: good-tm-os-clock-interrupt-handler
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow \text{good-tm}(\text{os-clock-interrupt-handler}(os))$

EVENT: Disable good-tm-os-clock-interrupt-handler.

THEOREM: tm-slimit-os-clock-interrupt-handler
 $\text{tm-slimit}(\text{os-clock-interrupt-handler}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-clock-interrupt-handler.

THEOREM: os-code-os-clock-interrupt-handler
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-clock-interrupt-handler}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-clock-interrupt-handler.

THEOREM: tm-interrupt-addresses-os-clock-interrupt-handler
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5,$
 $\text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\text{tm-memory}(\text{os-clock-interrupt-handler}(os)))$
 $= \text{getnth}('11,$
 $\text{tm-memory}(os))))))$

EVENT: Disable tm-interrupt-addresses-os-clock-interrupt-handler.

THEOREM: not-tm-in-supervisor-mode-os-clock-interrupt-handler
 $\neg \text{tm-in-supervisor-mode}(\text{os-clock-interrupt-handler}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-clock-interrupt-handler.

THEOREM: os-task-table-os-clock-interrupt-handler
good-tm (*os*)
→ (os-task-table (os-clock-interrupt-handler (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-clock-interrupt-handler.

THEOREM: os-segment-table-os-clock-interrupt-handler
good-tm (*os*)
→ (os-segment-table (os-clock-interrupt-handler (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-clock-interrupt-handler.

THEOREM: os-readyq-os-clock-interrupt-handler
good-tm (*os*)
→ (os-readyq (os-clock-interrupt-handler (*os*))
= os-clock-new-readyq (*os*))

EVENT: Disable os-readyq-os-clock-interrupt-handler.

THEOREM: finite-number-queuep-os-clock-new-readyq
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ finite-number-queuep (os-clock-new-readyq (*os*), '16, '16)

EVENT: Disable finite-number-queuep-os-clock-new-readyq.

THEOREM: not-array-qemptyp-os-clock-new-readyq
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (¬ array-qemptyp (os-clock-new-readyq (*os*)))

EVENT: Disable not-array-qemptyp-os-clock-new-readyq.

THEOREM: os-ibuffers-os-clock-interrupt-handler
good-tm (*os*)
→ (os-ibuffers (os-clock-interrupt-handler (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-clock-interrupt-handler.

THEOREM: os-obuffers-os-clock-interrupt-handler

good-tm (*os*)

→ (os-obuffers (os-clock-interrupt-handler (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-clock-interrupt-handler.

THEOREM: os-mbuffers-os-clock-interrupt-handler

good-tm (*os*)

→ (os-mbuffers (os-clock-interrupt-handler (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-clock-interrupt-handler.

THEOREM: os-status-table-os-clock-interrupt-handler

good-tm (*os*)

→ (os-status-table (os-clock-interrupt-handler (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-clock-interrupt-handler.

THEOREM: os-ready-set-os-clock-interrupt-handler

good-tm (*os*)

→ (os-ready-set (os-clock-interrupt-handler (*os*)) = os-ready-set (*os*))

EVENT: Disable os-ready-set-os-clock-interrupt-handler.

THEOREM: permutation-os-clock-new-readyq-os-ready-set

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))

→ permutation (mapup-queue (os-clock-new-readyq (*os*)),
os-ready-set (os-clock-interrupt-handler (*os*)))

EVENT: Disable permutation-os-clock-new-readyq-os-ready-set.

THEOREM: tm-base-os-clock-interrupt-handler

tm-base (os-clock-interrupt-handler (*os*))

= os-new-base (array-qfirst (os-clock-new-readyq (*os*)), *os*)

EVENT: Disable tm-base-os-clock-interrupt-handler.

THEOREM: tm-limit-os-clock-interrupt-handler

tm-limit (os-clock-interrupt-handler (*os*))

= os-new-limit (array-qfirst (os-clock-new-readyq (*os*)), *os*)

EVENT: Disable tm-limit-os-clock-interrupt-handler.

THEOREM: tm-base-tm-limit-os-clock-interrupt-handler
good-tm (*os*)
→ ((tm-base (os-clock-interrupt-handler (*os*))
= base (getnth (os-current-taskid (os-clock-interrupt-handler (*os*)),
table ('2, os-segment-table (*os*))))))
∧ (tm-limit (os-clock-interrupt-handler (*os*))
= limit (getnth (os-current-taskid (os-clock-interrupt-handler (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-clock-interrupt-handler.

THEOREM: not-tm-waiting-os-clock-interrupt-handler
¬ tm-waiting (os-clock-interrupt-handler (*os*))

EVENT: Disable not-tm-waiting-os-clock-interrupt-handler.

THEOREM: good-os-os-clock-interrupt-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-os (os-clock-interrupt-handler (*os*))

EVENT: Disable good-os-os-clock-interrupt-handler.

THEOREM: good-tm-os-intended-error-interrupt
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-tm (os-intended-error-interrupt (*os*))

EVENT: Disable good-tm-os-intended-error-interrupt.

THEOREM: good-tm-os-error-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ array-qempty (array-deq (os-readyq (*os*))))
→ good-tm (os-error-handler-path1 (*os*))

EVENT: Disable good-tm-os-error-handler-path1.

THEOREM: tm-slimit-os-error-handler-path1
tm-slimit (os-error-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-error-handler-path1.

THEOREM: os-code-os-error-handler-path1
good-tm (*os*) → (os-code (os-error-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-error-handler-path1.

THEOREM: tm-interrupt-addresses-os-error-handler-path1

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-error-handler-path1}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-error-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-error-handler-path1

$$\neg \text{tm-in-supervisor-mode}(\text{os-error-handler-path1}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-error-handler-path1.

THEOREM: os-task-table-os-error-handler-path1

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-error-handler-path1}(os)) = \text{os-new-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-error-handler-path1.

THEOREM: os-segment-table-os-error-handler-path1

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-error-handler-path1}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-error-handler-path1.

THEOREM: os-readyq-os-error-handler-path1

good-tm (*os*)
 \rightarrow (os-readyq (os-error-handler-path1 (*os*)) = array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-error-handler-path1.

THEOREM: os-ibuffers-os-error-handler-path1
good-tm (*os*)
 \rightarrow (os-ibuffers (os-error-handler-path1 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-error-handler-path1.

THEOREM: os-obuffers-os-error-handler-path1
good-tm (*os*)
 \rightarrow (os-obuffers (os-error-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-error-handler-path1.

THEOREM: os-mbuffers-os-error-handler-path1
good-tm (*os*)
 \rightarrow (os-mbuffers (os-error-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-error-handler-path1.

THEOREM: os-status-table-os-error-handler-path1
good-tm (*os*)
 \rightarrow (os-status-table (os-error-handler-path1 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)), '1, '0, *os*))

EVENT: Disable os-status-table-os-error-handler-path1.

EVENT: Let us define the theory *executable-theory* to consist of the following events: *1*greaterp, *1*leq, *1*geq, *1*member, *1*iff, *1*ord-lessp, *1*ordinalp, *1*assoc, *1*pairlist, *1*subrp, *1*apply-subr, *1*formals, *1*body, *1*fix-cost, *1*strip-cars, *1*sum-cdrs, *1*v&c\$, *1*v&c-apply\$, *1*eval\$, *1*quantifier-initial-value, *1*add-to-set, *1*append, *1*max, *1*union, *1*quantifier-operation, *1*for, *1*exp, *1*divides, *1*min, *1*length, *1*insert, *1*occurrences, *1*remove, *1*permutation, *1*plistp, *1*setp, *1*sorted, *1*getnth, *1*putnth, *1*getseg, *1*putseg, *1*fixlength, *1*firstn, *1*nthcdr, *1*nested-intervals, *1*getnth2, *1*putnth2, *1*reverse, *1*finite-numberp, *1*number-listp, *1*finite-number-listp, *1*non-zero-listp, *1*tablep, *1*table, *1*flatten, *1*finite-number-tablep, *1*nonlast, *1*incr-mod, *1*decr-mod, *1*qfirst, *1*enq, *1*deq, *1*qemp-ty, *1*qfullp, *1*qreplace, *1*qhead-field, *1*qtail-field, *1*qcurrenlength-field,

*1*qmaxlength-field, *1*qarray-field, *1*delta, *1*array-qindex-relation, *1*array-queuep, *1*array-enq, *1*array-deq, *1*array-qfirst, *1*array-qfullp, *1*array-qempty, *1*array-nonlast, *1*array-qreplace, *1*delta-segment, *1*mapup-queue, *1*qfirst2, *1*enq2, *1*deq2, *1*qfullp2, *1*qempty2.

THEOREM: good-status-list-os-status-table-os-error-handler-path1
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow good-status-list (table ('2,
 os-status-table (os-error-handler-path1 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-error-handler-path1.

THEOREM: os-ready-set-os-error-handler-path1
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (os-ready-set (os-error-handler-path1 (*os*))
 = index-ready-set ('0,
 table ('2,
 os-update-status (array-qfirst (os-readyq (*os*)),
 '1,
 '0,
 os))))

EVENT: Disable os-ready-set-os-error-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-error-handler-path1
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow permutation (mapup-queue (array-deq (os-readyq (*os*))),
 os-ready-set (os-error-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-error-handler-path1.

THEOREM: tm-waiting-os-error-handler-path1
 tm-waiting (os-error-handler-path1 (*os*))

EVENT: Disable tm-waiting-os-error-handler-path1.

THEOREM: good-os-os-error-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge array-qempty (array-deq (os-readyq (*os*))))))
 \rightarrow good-os (os-error-handler-path1 (*os*))

EVENT: Disable good-os-os-error-handler-path1.

THEOREM: good-tm-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qempty (array-deq (os-readyq (*os*))))))
 \rightarrow good-tm (os-error-handler-path2 (*os*)))

EVENT: Disable good-tm-os-error-handler-path2.

THEOREM: tm-slimit-os-error-handler-path2
 tm-slimit (os-error-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-error-handler-path2.

THEOREM: os-code-os-error-handler-path2
 good-tm (*os*) \rightarrow (os-code (os-error-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-error-handler-path2.

THEOREM: tm-interrupt-addresses-os-error-handler-path2
 good-tm (*os*)
 \rightarrow ((getnth ('3, tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('3, tm-memory (*os*)))
 \wedge ((getnth ('4, tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('4, tm-memory (*os*)))
 \wedge ((getnth ('5, tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('5, tm-memory (*os*)))
 \wedge ((getnth ('6,
 tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('6, tm-memory (*os*)))
 \wedge ((getnth ('7,
 tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('7, tm-memory (*os*)))
 \wedge (getnth ('11,
 tm-memory (os-error-handler-path2 (*os*)))
 = getnth ('11,
 tm-memory (*os*))))))))))

EVENT: Disable tm-interrupt-addresses-os-error-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-error-handler-path2
 \neg tm-in-supervisor-mode (os-error-handler-path2 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-error-handler-path2.

THEOREM: os-task-table-os-error-handler-path2

good-tm (*os*)

→ (os-task-table (os-error-handler-path2 (*os*)) = os-new-task-table (*os*))

EVENT: Disable os-task-table-os-error-handler-path2.

THEOREM: os-segment-table-os-error-handler-path2

good-tm (*os*)

→ (os-segment-table (os-error-handler-path2 (*os*))

= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-error-handler-path2.

THEOREM: os-readyq-os-error-handler-path2

good-tm (*os*)

→ (os-readyq (os-error-handler-path2 (*os*)) = array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-error-handler-path2.

THEOREM: os-ibuffers-os-error-handler-path2

good-tm (*os*)

→ (os-ibuffers (os-error-handler-path2 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-error-handler-path2.

THEOREM: os-obuffers-os-error-handler-path2

good-tm (*os*)

→ (os-obuffers (os-error-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-error-handler-path2.

THEOREM: os-mbuffers-os-error-handler-path2

good-tm (*os*)

→ (os-mbuffers (os-error-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-error-handler-path2.

THEOREM: os-status-table-os-error-handler-path2

good-tm (*os*)

→ (os-status-table (os-error-handler-path2 (*os*))

= os-update-status (array-qfirst (os-readyq (*os*)), '1, '0, *os*))

EVENT: Disable os-status-table-os-error-handler-path2.

THEOREM: good-status-list-os-status-table-os-error-handler-path2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{good-status-list}(\text{table}('2,$
 $\qquad\qquad\qquad \text{os-status-table}(\text{os-error-handler-path2}(os))))$

EVENT: Disable good-status-list-os-status-table-os-error-handler-path2.

THEOREM: os-ready-set-os-error-handler-path2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{os-ready-set}(\text{os-error-handler-path2}(os))$
 $\quad = \text{index-ready-set}('0,$
 $\qquad\qquad\qquad \text{table}('2,$
 $\qquad\qquad\qquad \text{os-update-status}(\text{array-qfirst}(\text{os-readyq}(os)),$
 $\qquad\qquad\qquad \quad '1,$
 $\qquad\qquad\qquad \quad '0,$
 $\qquad\qquad\qquad \quad os))))$

EVENT: Disable os-ready-set-os-error-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-error-handler-path2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow \text{permutation}(\text{mapup-queue}(\text{array-deq}(\text{os-readyq}(os))),$
 $\qquad\qquad\qquad \text{os-ready-set}(\text{os-error-handler-path2}(os)))$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-error-handler-path2.

THEOREM: tm-base-os-error-handler-path2
 $\text{tm-base}(\text{os-error-handler-path2}(os))$
 $= \text{os-new-base}(\text{array-qfirst}(\text{array-deq}(\text{os-readyq}(os))), os)$

EVENT: Disable tm-base-os-error-handler-path2.

THEOREM: tm-limit-os-error-handler-path2
 $\text{tm-limit}(\text{os-error-handler-path2}(os))$
 $= \text{os-new-limit}(\text{array-qfirst}(\text{array-deq}(\text{os-readyq}(os))), os)$

EVENT: Disable tm-limit-os-error-handler-path2.

THEOREM: tm-base-tm-limit-os-error-handler-path2
 $\text{good-tm}(os)$
 $\rightarrow ((\text{tm-base}(\text{os-error-handler-path2}(os))$
 $\quad = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-error-handler-path2}(os))),$

$$\begin{aligned} & \text{table ('2, os-segment-table (os))}) \\ \wedge & \text{ (tm-limit (os-error-handler-path2 (os))} \\ & = \text{ limit (getnth (os-current-taskid (os-error-handler-path2 (os)),} \\ & \text{ table ('2, os-segment-table (os))}))} \end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-error-handler-path2.

THEOREM: not-tm-waiting-os-error-handler-path2
 $\neg \text{tm-waiting (os-error-handler-path2 (os))}$

EVENT: Disable not-tm-waiting-os-error-handler-path2.

THEOREM: good-os-os-error-handler-path2
 $(\text{good-os (os)}$
 $\wedge ((\neg \text{tm-waiting (os)})$
 $\wedge (\neg \text{array-qempty (array-deq (os-readyq (os)))))$
 $\rightarrow \text{good-os (os-error-handler-path2 (os))}$

EVENT: Disable good-os-os-error-handler-path2.

THEOREM: good-os-os-error-handler
 $(\text{good-os (os)} \wedge (\neg \text{tm-waiting (os)})) \rightarrow \text{good-os (os-error-handler (os))}$

EVENT: Disable good-os-os-error-handler.

THEOREM: good-tm-os-intended-svc-send-interrupt
 $(\text{good-os (os)} \wedge ((\neg \text{tm-waiting (os)}) \wedge (\neg \text{tm-errorp (os)})))$
 $\rightarrow \text{good-tm (os-intended-svc-send-interrupt (os))}$

EVENT: Disable good-tm-os-intended-svc-send-interrupt.

THEOREM: good-tm-os-svc-send-handler-path1
 $(\text{good-os (os)}$
 $\wedge ((\neg \text{tm-waiting (os)})$
 $\wedge ((\neg \text{tm-errorp (os)})$
 $\wedge (((\text{tm-svcid (os)} \bmod '4) = '0)$
 $\wedge (\text{array-qfullp (os-current-mbuffer (os-current-taskid (os),$
 os-destid (os,
 '16),
 os))
 $\wedge \text{array-qempty (array-deq (os-readyq (os)))))$
 $\rightarrow \text{good-tm (os-svc-send-handler-path1 (os))}$

EVENT: Disable good-tm-os-svc-send-handler-path1.

THEOREM: tm-slimit-os-svc-send-handler-path1
 $\text{tm-slimit}(\text{os-svc-send-handler-path1}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-svc-send-handler-path1.

THEOREM: os-code-os-svc-send-handler-path1
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-send-handler-path1}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-svc-send-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path1
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os))$
 $\wedge ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os))$
 $\wedge ((\text{getnth}('6,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os))$
 $\wedge ((\text{getnth}('7,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os))$
 $\wedge (\text{getnth}('11,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path1}(os)))$
 $= \text{getnth}('11,$
 $\quad \text{tm-memory}(os))))))))))$

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path1
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-send-handler-path1}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-send-handler-path1.

THEOREM: os-task-table-os-svc-send-handler-path1
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-send-handler-path1}(os))$
 $= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-svc-send-handler-path1.

THEOREM: os-segment-table-os-svc-send-handler-path1
good-tm (*os*)
→ (os-segment-table (os-svc-send-handler-path1 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-svc-send-handler-path1.

THEOREM: os-readyq-os-svc-send-handler-path1
good-tm (*os*)
→ (os-readyq (os-svc-send-handler-path1 (*os*))
= array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-send-handler-path1.

THEOREM: os-ibuffers-os-svc-send-handler-path1
good-tm (*os*)
→ (os-ibuffers (os-svc-send-handler-path1 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-send-handler-path1.

THEOREM: os-obuffers-os-svc-send-handler-path1
good-tm (*os*)
→ (os-obuffers (os-svc-send-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-send-handler-path1.

THEOREM: os-mbuffers-os-svc-send-handler-path1
good-tm (*os*)
→ (os-mbuffers (os-svc-send-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-send-handler-path1.

THEOREM: os-status-table-os-svc-send-handler-path1
good-tm (*os*)
→ (os-status-table (os-svc-send-handler-path1 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)),
'2,
os-destid (*os*, '16),
os))

EVENT: Disable os-status-table-os-svc-send-handler-path1.

THEOREM: os-ready-set-os-svc-send-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (os-ready-set (os-svc-send-handler-path1 (*os*))
= index-ready-set ('0,
putnth (cons ('2,
cons (os-destid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-svc-send-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-send-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path1.

THEOREM: tm-waiting-os-svc-send-handler-path1
tm-waiting (os-svc-send-handler-path1 (*os*))

EVENT: Disable tm-waiting-os-svc-send-handler-path1.

THEOREM: good-os-os-svc-send-handler-path1
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg tm-errorp (*os*))
 \wedge (((tm-svcid (*os*) **mod** '4) = '0)
 \wedge (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*,
'16),
os))
 \wedge array-qempty (array-deq (os-readyq (*os*))))))))))
 \rightarrow good-os (os-svc-send-handler-path1 (*os*))

EVENT: Disable good-os-os-svc-send-handler-path1.

THEOREM: good-tm-os-svc-send-handler-path2
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{tm-errorp}(os)) \\
& \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \\
& \quad \quad \wedge (\text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \text{os-destid}(os, \\
& \quad \quad \quad '16), \\
& \quad \quad \quad os))) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))))) \\
\rightarrow & \text{good-tm}(\text{os-svc-send-handler-path2}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-svc-send-handler-path2.

THEOREM: tm-slimit-os-svc-send-handler-path2
 $\text{tm-slimit}(\text{os-svc-send-handler-path2}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-svc-send-handler-path2.

THEOREM: os-code-os-svc-send-handler-path2
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-send-handler-path2}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-svc-send-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path2
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\quad \text{tm-memory}(\text{os-svc-send-handler-path2}(os)))$
 $= \text{getnth}('11,$
 $\quad \text{tm-memory}(os)))))))))$

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path2
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-send-handler-path2}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-send-handler-path2.

THEOREM: os-task-table-os-svc-send-handler-path2
good-tm (*os*)
→ (os-task-table (os-svc-send-handler-path2 (*os*)))
= os-new-task-table (*os*)

EVENT: Disable os-task-table-os-svc-send-handler-path2.

THEOREM: os-segment-table-os-svc-send-handler-path2
good-tm (*os*)
→ (os-segment-table (os-svc-send-handler-path2 (*os*)))
= os-segment-table (*os*)

EVENT: Disable os-segment-table-os-svc-send-handler-path2.

THEOREM: os-readyq-os-svc-send-handler-path2
good-tm (*os*)
→ (os-readyq (os-svc-send-handler-path2 (*os*)))
= array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-send-handler-path2.

THEOREM: os-ibuffers-os-svc-send-handler-path2
good-tm (*os*)
→ (os-ibuffers (os-svc-send-handler-path2 (*os*))) = os-ibuffers (*os*)

EVENT: Disable os-ibuffers-os-svc-send-handler-path2.

THEOREM: os-obuffers-os-svc-send-handler-path2
good-tm (*os*)
→ (os-obuffers (os-svc-send-handler-path2 (*os*))) = os-obuffers (*os*)

EVENT: Disable os-obuffers-os-svc-send-handler-path2.

THEOREM: os-mbuffers-os-svc-send-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-svc-send-handler-path2 (*os*))) = os-mbuffers (*os*)

EVENT: Disable os-mbuffers-os-svc-send-handler-path2.

THEOREM: os-status-table-os-svc-send-handler-path2
good-tm (*os*)
→ (os-status-table (os-svc-send-handler-path2 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)),
'2,
os-destid (*os*, '16),
os))

EVENT: Disable os-status-table-os-svc-send-handler-path2.

THEOREM: os-ready-set-os-svc-send-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-send-handler-path2 (*os*))
= index-ready-set ('0,
putnth (cons ('2,
cons (os-destid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-svc-send-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-send-handler-path2 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path2.

THEOREM: tm-base-os-svc-send-handler-path2
tm-base (os-svc-send-handler-path2 (*os*))
= os-new-base (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-base-os-svc-send-handler-path2.

THEOREM: tm-limit-os-svc-send-handler-path2
tm-limit (os-svc-send-handler-path2 (*os*))
= os-new-limit (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-limit-os-svc-send-handler-path2.

THEOREM: tm-base-tm-limit-os-svc-send-handler-path2
good-tm (*os*)

$$\begin{aligned} \rightarrow & ((\text{tm-base}(\text{os-svc-send-handler-path2}(os)) \\ & = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path2}(os)), \\ & \quad \text{table}('2, \text{os-segment-table}(os)))))) \\ \wedge & (\text{tm-limit}(\text{os-svc-send-handler-path2}(os)) \\ & = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path2}(os)), \\ & \quad \text{table}('2, \text{os-segment-table}(os)))))) \end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-svc-send-handler-path2.

THEOREM: not-tm-waiting-os-svc-send-handler-path2
 $\neg \text{tm-waiting}(\text{os-svc-send-handler-path2}(os))$

EVENT: Disable not-tm-waiting-os-svc-send-handler-path2.

THEOREM: good-os-os-svc-send-handler-path2
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\neg \text{tm-errorp}(os)) \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \wedge (\text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \text{os-destid}(os, '16), os)) \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))))))$
 $\rightarrow \text{good-os}(\text{os-svc-send-handler-path2}(os))$

EVENT: Disable good-os-os-svc-send-handler-path2.

THEOREM: good-tm-os-svc-send-handler-path3
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\neg \text{tm-errorp}(os)) \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \text{os-destid}(os, '16), os))) \wedge ((\text{getnth}('2 * \text{os-destid}(os, '16), \text{os-status-table}(os)) = '3) \wedge (\text{getnth}('1 + ('2 * \text{os-destid}(os,$

'16)),
 os-status-table (os)
 = os-current-taskid (os))))))

→ good-tm (os-svc-send-handler-path3 (os))

EVENT: Disable good-tm-os-svc-send-handler-path3.

THEOREM: tm-slimit-os-svc-send-handler-path3
 tm-slimit (os-svc-send-handler-path3 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-svc-send-handler-path3.

THEOREM: os-code-os-svc-send-handler-path3
 good-tm (os) → (os-code (os-svc-send-handler-path3 (os)) = os-code (os))

EVENT: Disable os-code-os-svc-send-handler-path3.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path3
 good-tm (os)
 → ((getnth ('3, tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('3, tm-memory (os)))
 ∧ ((getnth ('4, tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('4, tm-memory (os)))
 ∧ ((getnth ('5, tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('5, tm-memory (os)))
 ∧ ((getnth ('6,
 tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('6, tm-memory (os)))
 ∧ ((getnth ('7,
 tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('7, tm-memory (os)))
 ∧ (getnth ('11,
 tm-memory (os-svc-send-handler-path3 (os)))
 = getnth ('11,
 tm-memory (os))))))

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path3
 ¬ tm-in-supervisor-mode (os-svc-send-handler-path3 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-svc-send-handler-path3.

THEOREM: os-task-table-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-task-table (os-svc-send-handler-path3 (*os*)))
 = os-new-task-table (*os*)

EVENT: Disable os-task-table-os-svc-send-handler-path3.

THEOREM: os-segment-table-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-segment-table (os-svc-send-handler-path3 (*os*)))
 = os-segment-table (*os*)

EVENT: Disable os-segment-table-os-svc-send-handler-path3.

THEOREM: os-readyq-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-readyq (os-svc-send-handler-path3 (*os*)))
 = array-enq (os-destid (*os*, '16), os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-send-handler-path3.

THEOREM: os-ibuffers-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-ibuffers (os-svc-send-handler-path3 (*os*))) = os-ibuffers (*os*)

EVENT: Disable os-ibuffers-os-svc-send-handler-path3.

THEOREM: os-obuffers-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-obuffers (os-svc-send-handler-path3 (*os*))) = os-obuffers (*os*)

EVENT: Disable os-obuffers-os-svc-send-handler-path3.

THEOREM: os-mbuffers-os-svc-send-handler-path3
 good-tm (*os*)
 → (os-mbuffers (os-svc-send-handler-path3 (*os*)))
 = os-mbuffers-with-enqueued-message (*os*)

EVENT: Disable os-mbuffers-os-svc-send-handler-path3.

THEOREM: finite-number-queue-listp-os-mbuffers-with-enqueued-message
 (good-os (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \text{os-destid}(os, '16), \\
& \quad \quad \quad os)))))) \\
\rightarrow & \text{finite-number-queue-listp}(\text{table}('8, \\
& \quad \quad \quad \text{os-mbuffers-with-enqueued-message}(os), \\
& \quad \quad \quad '4, \\
& \quad \quad \quad '65536))
\end{aligned}$$

EVENT: Disable finite-number-queue-listp-os-mbuffers-with-enqueued-message.

THEOREM: os-status-table-os-svc-send-handler-path3
good-tm(*os*)
 \rightarrow (os-status-table(os-svc-send-handler-path3(*os*))
= putseg('0 0),
'2 * os-destid(*os*, '16),
os-status-table(*os*)))

EVENT: Disable os-status-table-os-svc-send-handler-path3.

THEOREM: os-ready-set-os-svc-send-handler-path3
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (os-ready-set(os-svc-send-handler-path3(*os*))
= index-ready-set('0,
putnth('0 0),
os-destid(*os*, '16),
table('2, os-status-table(*os*))))))

EVENT: Disable os-ready-set-os-svc-send-handler-path3.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path3
(good-os(*os*)
 \wedge ((\neg tm-waiting(*os*))
 \wedge (getnth('2 * os-destid(*os*, '16), os-status-table(*os*))
= '3)))
 \rightarrow permutation(mapup-queue(array-enq(os-destid(*os*, '16), os-readyq(*os*)),
os-ready-set(os-svc-send-handler-path3(*os*))))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path3.

THEOREM: not-tm-waiting-os-svc-send-handler-path3
 \neg tm-waiting(os-svc-send-handler-path3(*os*))

EVENT: Disable not-tm-waiting-os-svc-send-handler-path3.

THEOREM: tm-base-os-svc-send-handler-path3
 $\text{tm-base}(\text{os-svc-send-handler-path3}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-svc-send-handler-path3.

THEOREM: tm-limit-os-svc-send-handler-path3
 $\text{tm-limit}(\text{os-svc-send-handler-path3}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-svc-send-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-send-handler-path3
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\text{getnth}('2 * \text{os-destid}(os, '16), \text{os-status-table}(os)) = '3)))$
 $\rightarrow ((\text{tm-base}(\text{os-svc-send-handler-path3}(os)) = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path3}(os)), \text{table}('2, \text{os-segment-table}(os)))))) \wedge (\text{tm-limit}(\text{os-svc-send-handler-path3}(os)) = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path3}(os)), \text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-svc-send-handler-path3.

THEOREM: finite-numberp-os-destid
 $(\text{os-destid}(os, '16) \in \mathbf{N})$
 $\wedge ((\text{os-destid}(os, '16) < '16) \wedge \text{finite-numberp}(\text{os-destid}(os, '16), '16))$

EVENT: Disable finite-numberp-os-destid.

THEOREM: good-os-os-svc-send-handler-path3
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\neg \text{tm-errorp}(os)) \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \text{os-destid}(os, '16), os)))$

$$\begin{aligned}
& \wedge ((\text{getnth}('2 * \text{os-destid}(os, '16), \\
& \quad \text{os-status-table}(os)) \\
& \quad = '3) \\
& \wedge (\text{getnth}('1 \\
& \quad + ('2 \\
& \quad \quad * \text{os-destid}(os, \\
& \quad \quad \quad '16)), \\
& \quad \quad \text{os-status-table}(os)) \\
& \quad = \text{os-current-taskid}(os)))))) \\
\rightarrow & \text{good-os}(\text{os-svc-send-handler-path3}(os))
\end{aligned}$$

EVENT: Disable good-os-os-svc-send-handler-path3.

THEOREM: good-tm-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{tm-errorp}(os)) \\
& \quad \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \\
& \quad \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \quad \text{os-destid}(os, \\
& \quad \quad \quad \quad \quad '16), \\
& \quad \quad \quad \quad \quad os)))))) \\
& \wedge (\neg ((\text{getnth}('2 \\
& \quad \quad * \text{os-destid}(os, \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = '3) \\
& \quad \wedge (\text{getnth}('1 \\
& \quad \quad + ('2 \\
& \quad \quad \quad * \text{os-destid}(os, \\
& \quad \quad \quad \quad '16)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = \text{os-current-taskid}(os))))))))) \\
\rightarrow & \text{good-tm}(\text{os-svc-send-handler-path4}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-svc-send-handler-path4.

THEOREM: tm-slimit-os-svc-send-handler-path4

$$\text{tm-slimit}(\text{os-svc-send-handler-path4}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-svc-send-handler-path4.

THEOREM: os-code-os-svc-send-handler-path4

$$\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-send-handler-path4}(os)) = \text{os-code}(os))$$

EVENT: Disable os-code-os-svc-send-handler-path4.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-svc-send-handler-path4}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path4

$$\rightarrow \text{tm-in-supervisor-mode}(\text{os-svc-send-handler-path4}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-send-handler-path4.

THEOREM: os-task-table-os-svc-send-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-svc-send-handler-path4}(os)) \\ & = \text{os-new-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-svc-send-handler-path4.

THEOREM: os-segment-table-os-svc-send-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-svc-send-handler-path4}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-svc-send-handler-path4.

THEOREM: os-readyq-os-svc-send-handler-path4

good-tm (*os*)
→ (os-readyq (os-svc-send-handler-path4 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-svc-send-handler-path4.

THEOREM: os-ibuffers-os-svc-send-handler-path4
good-tm (*os*)
→ (os-ibuffers (os-svc-send-handler-path4 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-send-handler-path4.

THEOREM: os-obuffers-os-svc-send-handler-path4
good-tm (*os*)
→ (os-obuffers (os-svc-send-handler-path4 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-send-handler-path4.

THEOREM: os-mbuffers-os-svc-send-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-svc-send-handler-path4 (*os*))
= os-mbuffers-with-enqueued-message (*os*))

EVENT: Disable os-mbuffers-os-svc-send-handler-path4.

THEOREM: os-status-table-os-svc-send-handler-path4
good-tm (*os*)
→ (os-status-table (os-svc-send-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-svc-send-handler-path4.

THEOREM: os-ready-set-os-svc-send-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-send-handler-path4 (*os*)) = os-ready-set (*os*))

EVENT: Disable os-ready-set-os-svc-send-handler-path4.

THEOREM: not-tm-waiting-os-svc-send-handler-path4
¬ tm-waiting (os-svc-send-handler-path4 (*os*))

EVENT: Disable not-tm-waiting-os-svc-send-handler-path4.

THEOREM: tm-base-os-svc-send-handler-path4
tm-base (os-svc-send-handler-path4 (os)) = tm-base (os)

EVENT: Disable tm-base-os-svc-send-handler-path4.

THEOREM: tm-limit-os-svc-send-handler-path4
tm-limit (os-svc-send-handler-path4 (os)) = tm-limit (os)

EVENT: Disable tm-limit-os-svc-send-handler-path4.

THEOREM: tm-base-tm-limit-os-svc-send-handler-path4
(good-os (os) \wedge (\neg tm-waiting (os)))
 \rightarrow (((tm-base (os-svc-send-handler-path4 (os))
= base (getnth (os-current-taskid (os-svc-send-handler-path4 (os)),
table ('2, os-segment-table (os)))))
 \wedge (tm-limit (os-svc-send-handler-path4 (os))
= limit (getnth (os-current-taskid (os-svc-send-handler-path4 (os)),
table ('2, os-segment-table (os)))))

EVENT: Disable tm-base-tm-limit-os-svc-send-handler-path4.

THEOREM: good-os-os-svc-send-handler-path4
(good-os (os)
 \wedge (\neg tm-waiting (os))
 \wedge (\neg tm-errorp (os))
 \wedge (((tm-svcid (os) **mod** '4) = '0)
 \wedge (\neg array-qfullp (os-current-mbuffer (os-current-taskid (os),
os-destid (os,
'16),
os)))
 \wedge (\neg ((getnth ('2
* os-destid (os,
'16),
os-status-table (os))
= '3)
 \wedge (getnth ('1
+ ('2
* os-destid (os,
'16),
os-status-table (os))
= os-current-taskid (os))))))))))
 \rightarrow good-os (os-svc-send-handler-path4 (os))

EVENT: Disable good-os-os-svc-send-handler-path4.

THEOREM: good-os-os-svc-send-handler
 $(\text{good-os } (os)$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge ((\neg \text{tm-errorp } (os)) \wedge ((\text{tm-svcid } (os) \bmod '4) = '0))))$
 $\rightarrow \text{good-os } (\text{os-svc-send-handler } (os))$

EVENT: Disable good-os-os-svc-send-handler.

THEOREM: good-tm-os-intended-svc-receive-interrupt
 $(\text{good-os } (os) \wedge ((\neg \text{tm-waiting } (os)) \wedge (\neg \text{tm-errorp } (os))))$
 $\rightarrow \text{good-tm } (\text{os-intended-svc-receive-interrupt } (os))$

EVENT: Disable good-tm-os-intended-svc-receive-interrupt.

THEOREM: good-tm-os-svc-receive-handler-path1
 $(\text{good-os } (os)$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge ((\neg \text{tm-errorp } (os))$
 $\wedge (((\text{tm-svcid } (os) \bmod '4) = '1)$
 $\wedge (\text{array-qemptyp } (\text{os-current-mbuffer } (\text{os-srcid } (os,$
 $'16),$
 $\text{os-current-taskid } (os),$
 $os)))$
 $\wedge \text{array-qemptyp } (\text{array-deq } (\text{os-readyq } (os))))))$
 $\rightarrow \text{good-tm } (\text{os-svc-receive-handler-path1 } (os))$

EVENT: Disable good-tm-os-svc-receive-handler-path1.

THEOREM: tm-slimit-os-svc-receive-handler-path1
 $\text{tm-slimit } (\text{os-svc-receive-handler-path1 } (os)) = \text{tm-slimit } (os)$

EVENT: Disable tm-slimit-os-svc-receive-handler-path1.

THEOREM: os-code-os-svc-receive-handler-path1
 $\text{good-tm } (os) \rightarrow (\text{os-code } (\text{os-svc-receive-handler-path1 } (os)) = \text{os-code } (os))$

EVENT: Disable os-code-os-svc-receive-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-receive-handler-path1
 $\text{good-tm } (os)$
 $\rightarrow ((\text{getnth } ('3, \text{tm-memory } (\text{os-svc-receive-handler-path1 } (os)))$
 $= \text{getnth } ('3, \text{tm-memory } (os)))$

$$\begin{aligned}
& \wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-receive-handler-path1}(os))) \\
& \quad = \text{getnth}('4, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('5, \\
& \quad \quad \text{tm-memory}(\text{os-svc-receive-handler-path1}(os))) \\
& \quad \quad = \text{getnth}('5, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('6, \\
& \quad \quad \text{tm-memory}(\text{os-svc-receive-handler-path1}(os))) \\
& \quad \quad = \text{getnth}('6, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('7, \\
& \quad \quad \text{tm-memory}(\text{os-svc-receive-handler-path1}(os))) \\
& \quad \quad = \text{getnth}('7, \text{tm-memory}(os))) \\
& \quad \wedge (\text{getnth}('11, \\
& \quad \quad \text{tm-memory}(\text{os-svc-receive-handler-path1}(os))) \\
& \quad \quad = \text{getnth}('11, \\
& \quad \quad \quad \text{tm-memory}(os)))))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path1
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-receive-handler-path1}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-receive-handler-path1.

THEOREM: os-task-table-os-svc-receive-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-task-table}(\text{os-svc-receive-handler-path1}(os)) \\
= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-svc-receive-handler-path1.

THEOREM: os-segment-table-os-svc-receive-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-segment-table}(\text{os-svc-receive-handler-path1}(os)) \\
= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-svc-receive-handler-path1.

THEOREM: os-readyq-os-svc-receive-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-readyq}(\text{os-svc-receive-handler-path1}(os)) \\
= \text{array-deq}(\text{os-readyq}(os)))$

EVENT: Disable os-readyq-os-svc-receive-handler-path1.

THEOREM: os-ibuffers-os-svc-receive-handler-path1
good-tm (*os*)
→ (os-ibuffers (os-svc-receive-handler-path1 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-receive-handler-path1.

THEOREM: os-obuffers-os-svc-receive-handler-path1
good-tm (*os*)
→ (os-obuffers (os-svc-receive-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-receive-handler-path1.

THEOREM: os-mbuffers-os-svc-receive-handler-path1
good-tm (*os*)
→ (os-mbuffers (os-svc-receive-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-receive-handler-path1.

THEOREM: os-status-table-os-svc-receive-handler-path1
good-tm (*os*)
→ (os-status-table (os-svc-receive-handler-path1 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)),
'3,
os-srcid (*os*, '16),
os))

EVENT: Disable os-status-table-os-svc-receive-handler-path1.

THEOREM: os-ready-set-os-svc-receive-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-receive-handler-path1 (*os*))
= index-ready-set ('0,
putnth (cons ('3, cons (os-srcid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-svc-receive-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-receive-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path1.

THEOREM: tm-waiting-os-svc-receive-handler-path1
tm-waiting (os-svc-receive-handler-path1 (os))

EVENT: Disable tm-waiting-os-svc-receive-handler-path1.

THEOREM: good-os-os-svc-receive-handler-path1
(good-os (os)
 \wedge (\neg tm-waiting (os))
 \wedge (\neg tm-errorp (os))
 \wedge (((tm-svcid (os) **mod** '4) = '1)
 \wedge (array-qempty (os-current-mbuffer (os-srcid (os,
 '16),
 os-current-taskid (os),
 os))
 \wedge array-qempty (array-deq (os-readyq (os)))))))))
→ good-os (os-svc-receive-handler-path1 (os))

EVENT: Disable good-os-os-svc-receive-handler-path1.

THEOREM: good-tm-os-svc-receive-handler-path2
(good-os (os)
 \wedge (\neg tm-waiting (os))
 \wedge (\neg tm-errorp (os))
 \wedge (((tm-svcid (os) **mod** '4) = '1)
 \wedge (array-qempty (os-current-mbuffer (os-srcid (os,
 '16),
 os-current-taskid (os),
 os))
 \wedge (\neg array-qempty (array-deq (os-readyq (os)))))))))
→ good-tm (os-svc-receive-handler-path2 (os))

EVENT: Disable good-tm-os-svc-receive-handler-path2.

THEOREM: tm-slimit-os-svc-receive-handler-path2
tm-slimit (os-svc-receive-handler-path2 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-svc-receive-handler-path2.

THEOREM: os-code-os-svc-receive-handler-path2
good-tm (os) → (os-code (os-svc-receive-handler-path2 (os)) = os-code (os))

EVENT: Disable os-code-os-svc-receive-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-receive-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \\ & \quad \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-svc-receive-handler-path2}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path2

$$\neg \text{tm-in-supervisor-mode}(\text{os-svc-receive-handler-path2}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-receive-handler-path2.

THEOREM: os-task-table-os-svc-receive-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-svc-receive-handler-path2}(os)) \\ & = \text{os-new-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-svc-receive-handler-path2.

THEOREM: os-segment-table-os-svc-receive-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-svc-receive-handler-path2}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-svc-receive-handler-path2.

THEOREM: os-readyq-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-readyq (os-svc-receive-handler-path2 (*os*))
= array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-receive-handler-path2.

THEOREM: os-ibuffers-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-ibuffers (os-svc-receive-handler-path2 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-receive-handler-path2.

THEOREM: os-obuffers-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-obuffers (os-svc-receive-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-receive-handler-path2.

THEOREM: os-mbuffers-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-svc-receive-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-receive-handler-path2.

THEOREM: os-status-table-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-status-table (os-svc-receive-handler-path2 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)),
'3,
os-srcid (*os*, '16),
os))

EVENT: Disable os-status-table-os-svc-receive-handler-path2.

THEOREM: os-ready-set-os-svc-receive-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-receive-handler-path2 (*os*))
= index-ready-set ('0,
putnth (cons ('3, cons (os-srcid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-svc-receive-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path2
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-receive-handler-path2 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path2.

THEOREM: tm-base-os-svc-receive-handler-path2
tm-base (os-svc-receive-handler-path2 (*os*))
= os-new-base (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-base-os-svc-receive-handler-path2.

THEOREM: tm-limit-os-svc-receive-handler-path2
tm-limit (os-svc-receive-handler-path2 (*os*))
= os-new-limit (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-limit-os-svc-receive-handler-path2.

THEOREM: tm-base-tm-limit-os-svc-receive-handler-path2
good-tm (*os*)
 \rightarrow ((tm-base (os-svc-receive-handler-path2 (*os*))
= base (getnth (os-current-taskid (os-svc-receive-handler-path2 (*os*)),
table ('2, os-segment-table (*os*))))))
 \wedge (tm-limit (os-svc-receive-handler-path2 (*os*))
= limit (getnth (os-current-taskid (os-svc-receive-handler-path2 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-receive-handler-path2.

THEOREM: not-tm-waiting-os-svc-receive-handler-path2
 \neg tm-waiting (os-svc-receive-handler-path2 (*os*))

EVENT: Disable not-tm-waiting-os-svc-receive-handler-path2.

THEOREM: good-os-os-svc-receive-handler-path2
(good-os (*os*)
 \wedge (\neg tm-waiting (*os*)
 \wedge (\neg tm-errorp (*os*)))

$$\begin{aligned}
& \wedge \left(((\text{tm-svcid}(os) \bmod '4) = '1) \right. \\
& \quad \wedge \left(\text{array-qempty}(\text{os-current-mbuffer}(\text{os-srcid}(os), \right. \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \left. \text{os-current-taskid}(os), \right. \\
& \quad \quad \quad \left. os) \right) \\
& \quad \wedge \left(\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \right) \\
\rightarrow & \text{good-os}(\text{os-svc-receive-handler-path2}(os))
\end{aligned}$$

EVENT: Disable good-os-os-svc-receive-handler-path2.

THEOREM: good-tm-os-svc-receive-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge \left((\neg \text{tm-waiting}(os)) \right. \\
& \quad \wedge \left((\neg \text{tm-errorp}(os)) \right. \\
& \quad \quad \wedge \left(((\text{tm-svcid}(os) \bmod '4) = '1) \right. \\
& \quad \quad \quad \wedge \left((\neg \text{array-qempty}(\text{os-current-mbuffer}(\text{os-srcid}(os), \right. \\
& \quad \quad \quad \quad \quad \quad \quad '16), \\
& \quad \quad \quad \quad \quad \quad \quad \left. \text{os-current-taskid}(os), \right. \\
& \quad \quad \quad \quad \quad \quad \quad \left. os) \right) \\
& \quad \quad \quad \wedge \left((\text{getnth}('2 * \text{os-srcid}(os, '16), \right. \\
& \quad \quad \quad \quad \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad \quad \quad \quad \quad = '2) \\
& \quad \quad \quad \quad \wedge \left(\text{getnth}('1 \right. \\
& \quad \quad \quad \quad \quad \quad \quad + ('2 \\
& \quad \quad \quad \quad \quad \quad \quad \quad * \text{os-srcid}(os, \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad '16), \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \left. \text{os-status-table}(os)) \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \left. = \text{os-current-taskid}(os) \right) \right) \right) \\
& \quad \quad \quad \left. \right) \\
\rightarrow & \text{good-tm}(\text{os-svc-receive-handler-path3}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-svc-receive-handler-path3.

THEOREM: tm-slimit-os-svc-receive-handler-path3

$$\text{tm-slimit}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-svc-receive-handler-path3.

THEOREM: os-code-os-svc-receive-handler-path3

$$\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-receive-handler-path3}(os)) = \text{os-code}(os))$$

EVENT: Disable os-code-os-svc-receive-handler-path3.

THEOREM: tm-interrupt-addresses-os-svc-receive-handler-path3

$$\begin{aligned}
& \text{good-tm}(os) \\
\rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('3, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('4, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('5, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('5, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('6, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('6, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('7, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('7, \text{tm-memory}(os))) \\
\wedge & (\text{getnth}('11, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path3}(os))) \\
& = \text{getnth}('11, \\
& \quad \text{tm-memory}(os))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path3
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-receive-handler-path3}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-receive-handler-path3.

THEOREM: os-task-table-os-svc-receive-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-receive-handler-path3}(os))$
 $= \text{os-task-table-with-delivered-message}(os))$

EVENT: Disable os-task-table-os-svc-receive-handler-path3.

THEOREM: lessp-crock-for-good-cpu-list-os-task-table-with-delivered-message
 $(n < '16) \rightarrow (('144 < ('9 + ('9 * n))) = '*1*false)$

EVENT: Disable lessp-crock-for-good-cpu-list-os-task-table-with-delivered-message.

THEOREM: good-cpu-list-os-task-table-with-delivered-message
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\neg \text{array-qemptyp}(\text{os-current-mbuffer}(\text{os-srcid}(os, '16),$
 $\text{os-current-taskid}(os),$

→ good-cpu-list (table ('9, os-task-table-with-delivered-message (os))))

EVENT: Disable good-cpu-list-os-task-table-with-delivered-message.

THEOREM: os-segment-table-os-svc-receive-handler-path3
good-tm (os)
→ (os-segment-table (os-svc-receive-handler-path3 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-svc-receive-handler-path3.

THEOREM: os-readyq-os-svc-receive-handler-path3
good-tm (os)
→ (os-readyq (os-svc-receive-handler-path3 (os))
= array-enq (os-srcid (os, '16), os-readyq (os)))

EVENT: Disable os-readyq-os-svc-receive-handler-path3.

THEOREM: os-ibuffers-os-svc-receive-handler-path3
good-tm (os)
→ (os-ibuffers (os-svc-receive-handler-path3 (os)) = os-ibuffers (os))

EVENT: Disable os-ibuffers-os-svc-receive-handler-path3.

THEOREM: os-obuffers-os-svc-receive-handler-path3
good-tm (os)
→ (os-obuffers (os-svc-receive-handler-path3 (os)) = os-obuffers (os))

EVENT: Disable os-obuffers-os-svc-receive-handler-path3.

THEOREM: os-mbuffers-os-svc-receive-handler-path3
good-tm (os)
→ (os-mbuffers (os-svc-receive-handler-path3 (os))
= os-mbuffers-with-dequeued-message (os))

EVENT: Disable os-mbuffers-os-svc-receive-handler-path3.

THEOREM: finite-number-queue-listp-os-mbuffers-with-dequeued-message
(good-os (os)
∧ ((¬ tm-waiting (os))
∧ (¬ array-qempty (os-current-mbuffer (os-srcid (os, '16),

$$\begin{aligned} & \text{os-current-taskid}(os), \\ & os)))))) \\ \rightarrow & \text{finite-number-queue-listp}(\text{table}('8, \\ & \text{os-mbuffers-with-dequeued-message}(os), \\ & '4, \\ & '65536)) \end{aligned}$$

EVENT: Disable finite-number-queue-listp-os-mbuffers-with-dequeued-message.

THEOREM: os-status-table-os-svc-receive-handler-path3
good-tm(*os*)
 \rightarrow (os-status-table(os-svc-receive-handler-path3(*os*))
= os-update-status(os-srcid(*os*, '16), '0, '0, *os*))

EVENT: Disable os-status-table-os-svc-receive-handler-path3.

THEOREM: os-ready-set-os-svc-receive-handler-path3
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (os-ready-set(os-svc-receive-handler-path3(*os*))
= index-ready-set('0,
putnth('0 0),
os-srcid(*os*, '16),
table('2, os-status-table(*os*))))

EVENT: Disable os-ready-set-os-svc-receive-handler-path3.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path3
(good-os(*os*)
 \wedge ((\neg tm-waiting(*os*)
 \wedge (getnth('2 * os-srcid(*os*, '16), os-status-table(*os*)
= '2))))
 \rightarrow permutation(mapup-queue(array-enq(os-srcid(*os*, '16), os-readyq(*os*)),
os-ready-set(os-svc-receive-handler-path3(*os*))))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path3.

THEOREM: not-tm-waiting-os-svc-receive-handler-path3
 \neg tm-waiting(os-svc-receive-handler-path3(*os*))

EVENT: Disable not-tm-waiting-os-svc-receive-handler-path3.

THEOREM: tm-base-os-svc-receive-handler-path3
tm-base(os-svc-receive-handler-path3(*os*)) = tm-base(*os*)

EVENT: Disable tm-base-os-svc-receive-handler-path3.

THEOREM: tm-limit-os-svc-receive-handler-path3
 $\text{tm-limit}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-svc-receive-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-receive-handler-path3
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{getnth}('2 * \text{os-srcid}(os, '16), \text{os-status-table}(os))$
 $= '2)))$
 $\rightarrow ((\text{tm-base}(\text{os-svc-receive-handler-path3}(os))$
 $= \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-receive-handler-path3}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$
 $\wedge (\text{tm-limit}(\text{os-svc-receive-handler-path3}(os))$
 $= \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-receive-handler-path3}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-svc-receive-handler-path3.

THEOREM: finite-numberp-os-srcid
 $(\text{os-srcid}(os, '16) \in \mathbf{N})$
 $\wedge ((\text{os-srcid}(os, '16) < '16)$
 $\wedge \text{finite-numberp}(\text{os-srcid}(os, '16), '16))$

EVENT: Disable finite-numberp-os-srcid.

THEOREM: good-os-os-svc-receive-handler-path3
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge (((\text{tm-svcid}(os) \bmod '4) = '1)$
 $\wedge ((\neg \text{array-qempty}(\text{os-current-mbuffer}(\text{os-srcid}(os,$
 $'16),$
 $\text{os-current-taskid}(os),$
 $os))))))$
 $\wedge ((\text{getnth}('2 * \text{os-srcid}(os, '16),$
 $\text{os-status-table}(os))$
 $= '2)$
 $\wedge (\text{getnth}('1$
 $+ ('2$

$$\begin{aligned} & * \text{os-srcid}(os, \\ & \quad '16)), \\ & \text{os-status-table}(os) \\ & = \text{os-current-taskid}(os)))))) \\ \rightarrow & \text{good-os}(\text{os-svc-receive-handler-path3}(os)) \end{aligned}$$

EVENT: Disable good-os-os-svc-receive-handler-path3.

THEOREM: good-tm-os-svc-receive-handler-path4

$$\begin{aligned} & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge ((\neg \text{tm-errorp}(os)) \\ & \quad \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '1) \\ & \quad \quad \quad \wedge ((\neg \text{array-qempty}(os-current-mbuffer(\text{os-srcid}(os, \\ & \quad \quad \quad \quad '16), \\ & \quad \quad \quad \quad \text{os-current-taskid}(os), \\ & \quad \quad \quad \quad os)))) \\ & \quad \quad \wedge (\neg ((\text{getnth}('2 \\ & \quad \quad \quad \quad * \text{os-srcid}(os, \\ & \quad \quad \quad \quad \quad '16), \\ & \quad \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad \quad = '2) \\ & \quad \quad \wedge (\text{getnth}('1 \\ & \quad \quad \quad \quad + ('2 \\ & \quad \quad \quad \quad \quad * \text{os-srcid}(os, \\ & \quad \quad \quad \quad \quad \quad '16)), \\ & \quad \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad \quad = \text{os-current-taskid}(os))))))))) \\ \rightarrow & \text{good-tm}(\text{os-svc-receive-handler-path4}(os)) \end{aligned}$$

EVENT: Disable good-tm-os-svc-receive-handler-path4.

THEOREM: tm-slimit-os-svc-receive-handler-path4

$$\text{tm-slimit}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-svc-receive-handler-path4.

THEOREM: os-code-os-svc-receive-handler-path4

$$\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-receive-handler-path4}(os)) = \text{os-code}(os))$$

EVENT: Disable os-code-os-svc-receive-handler-path4.

THEOREM: tm-interrupt-addresses-os-svc-receive-handler-path4

$$\begin{aligned}
& \text{good-tm}(os) \\
\rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('3, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('4, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('5, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('5, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('6, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('6, \text{tm-memory}(os))) \\
\wedge & ((\text{getnth}('7, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('7, \text{tm-memory}(os))) \\
\wedge & (\text{getnth}('11, \\
& \quad \text{tm-memory}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{getnth}('11, \\
& \quad \text{tm-memory}(os))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path4
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-receive-handler-path4}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-receive-handler-path4.

THEOREM: os-task-table-os-svc-receive-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-receive-handler-path4}(os))$
 $= \text{os-task-table-with-delivered-message}(os))$

EVENT: Disable os-task-table-os-svc-receive-handler-path4.

THEOREM: os-segment-table-os-svc-receive-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-segment-table}(\text{os-svc-receive-handler-path4}(os))$
 $= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-svc-receive-handler-path4.

THEOREM: os-readyq-os-svc-receive-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-readyq}(\text{os-svc-receive-handler-path4}(os)) = \text{os-readyq}(os))$

EVENT: Disable os-readyq-os-svc-receive-handler-path4.

THEOREM: os-ibuffers-os-svc-receive-handler-path4
good-tm (*os*)
→ (os-ibuffers (os-svc-receive-handler-path4 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-receive-handler-path4.

THEOREM: os-obuffers-os-svc-receive-handler-path4
good-tm (*os*)
→ (os-obuffers (os-svc-receive-handler-path4 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-receive-handler-path4.

THEOREM: os-mbuffers-os-svc-receive-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-svc-receive-handler-path4 (*os*))
= os-mbuffers-with-dequeued-message (*os*))

EVENT: Disable os-mbuffers-os-svc-receive-handler-path4.

THEOREM: os-status-table-os-svc-receive-handler-path4
good-tm (*os*)
→ (os-status-table (os-svc-receive-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-svc-receive-handler-path4.

THEOREM: os-ready-set-os-svc-receive-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-receive-handler-path4 (*os*)) = os-ready-set (*os*))

EVENT: Disable os-ready-set-os-svc-receive-handler-path4.

THEOREM: not-tm-waiting-os-svc-receive-handler-path4
¬ tm-waiting (os-svc-receive-handler-path4 (*os*))

EVENT: Disable not-tm-waiting-os-svc-receive-handler-path4.

THEOREM: tm-base-os-svc-receive-handler-path4
tm-base (os-svc-receive-handler-path4 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-svc-receive-handler-path4.

THEOREM: tm-limit-os-svc-receive-handler-path4
 $\text{tm-limit}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-svc-receive-handler-path4.

THEOREM: tm-base-tm-limit-os-svc-receive-handler-path4
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow ((\text{tm-base}(\text{os-svc-receive-handler-path4}(os))$
 $= \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-receive-handler-path4}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$
 $\wedge (\text{tm-limit}(\text{os-svc-receive-handler-path4}(os))$
 $= \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-receive-handler-path4}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-svc-receive-handler-path4.

THEOREM: good-os-os-svc-receive-handler-path4
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge (((\text{tm-srcid}(os) \bmod '4) = '1)$
 $\wedge ((\neg \text{array-qempty}(\text{os-current-mbuffer}(\text{os-srcid}(os,$
 $'16),$
 $\text{os-current-taskid}(os),$
 $os))))$
 $\wedge (\neg ((\text{getnth}('2$
 $* \text{os-srcid}(os,$
 $'16),$
 $\text{os-status-table}(os))$
 $= '2)$
 $\wedge (\text{getnth}('1$
 $+ ('2$
 $* \text{os-srcid}(os,$
 $'16)),$
 $\text{os-status-table}(os))$
 $= \text{os-current-taskid}(os))))))))))$
 $\rightarrow \text{good-os}(\text{os-svc-receive-handler-path4}(os))$

EVENT: Disable good-os-os-svc-receive-handler-path4.

THEOREM: good-os-os-svc-receive-handler

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge ((\neg \text{tm-errorp } (os)) \wedge ((\text{tm-svcid } (os) \bmod '4) = '1)))) \\
& \rightarrow \text{good-os } (os\text{-svc-receive-handler } (os))
\end{aligned}$$

EVENT: Disable good-os-os-svc-receive-handler.

THEOREM: good-tm-os-intended-svc-tyi-interrupt
 $(\text{good-os } (os) \wedge ((\neg \text{tm-waiting } (os)) \wedge (\neg \text{tm-errorp } (os))))$
 $\rightarrow \text{good-tm } (os\text{-intended-svc-tyi-interrupt } (os))$

EVENT: Disable good-tm-os-intended-svc-tyi-interrupt.

THEOREM: good-tm-os-svc-tyi-handler-path1
 $(\text{good-os } (os)$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge ((\neg \text{tm-errorp } (os))$
 $\quad \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '3)$
 $\quad \quad \quad \wedge (\text{array-qemptyp } (os\text{-current-ibuffer } (os\text{-current-taskid } (os),$
 $\quad \quad \quad \quad \quad \quad \quad \quad \quad \quad os)))$
 $\quad \quad \quad \quad \quad \quad \quad \wedge \text{array-qemptyp } (os\text{-readyq } (os))))))$
 $\rightarrow \text{good-tm } (os\text{-svc-tyi-handler-path1 } (os))$

EVENT: Disable good-tm-os-svc-tyi-handler-path1.

THEOREM: tm-slimit-os-svc-tyi-handler-path1
 $\text{tm-slimit } (os\text{-svc-tyi-handler-path1 } (os)) = \text{tm-slimit } (os)$

EVENT: Disable tm-slimit-os-svc-tyi-handler-path1.

THEOREM: os-code-os-svc-tyi-handler-path1
 $\text{good-tm } (os) \rightarrow (os\text{-code } (os\text{-svc-tyi-handler-path1 } (os)) = os\text{-code } (os))$

EVENT: Disable os-code-os-svc-tyi-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path1
 $\text{good-tm } (os)$
 $\rightarrow ((\text{getnth } ('3, \text{tm-memory } (os\text{-svc-tyi-handler-path1 } (os)))$
 $\quad = \text{getnth } ('3, \text{tm-memory } (os)))$
 $\quad \wedge ((\text{getnth } ('4, \text{tm-memory } (os\text{-svc-tyi-handler-path1 } (os)))$
 $\quad \quad = \text{getnth } ('4, \text{tm-memory } (os)))$
 $\quad \wedge ((\text{getnth } ('5, \text{tm-memory } (os\text{-svc-tyi-handler-path1 } (os)))$

$$\begin{aligned}
&= \text{getnth}('5, \text{tm-memory}(os)) \\
\wedge & \left(\text{getnth}('6, \right. \\
& \quad \left. \text{tm-memory}(\text{os-svc-tyi-handler-path1}(os)) \right) \\
&= \text{getnth}('6, \text{tm-memory}(os)) \\
\wedge & \left(\text{getnth}('7, \right. \\
& \quad \left. \text{tm-memory}(\text{os-svc-tyi-handler-path1}(os)) \right) \\
&= \text{getnth}('7, \text{tm-memory}(os)) \\
\wedge & \left(\text{getnth}('11, \right. \\
& \quad \left. \text{tm-memory}(\text{os-svc-tyi-handler-path1}(os)) \right) \\
&= \text{getnth}('11, \\
& \quad \text{tm-memory}(os))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path1
 \rightarrow tm-in-supervisor-mode(os-svc-tyi-handler-path1(os))

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyi-handler-path1.

THEOREM: os-task-table-os-svc-tyi-handler-path1
good-tm(os)
 \rightarrow (os-task-table(os-svc-tyi-handler-path1(os))
= os-new-task-table(os))

EVENT: Disable os-task-table-os-svc-tyi-handler-path1.

THEOREM: os-segment-table-os-svc-tyi-handler-path1
good-tm(os)
 \rightarrow (os-segment-table(os-svc-tyi-handler-path1(os))
= os-segment-table(os))

EVENT: Disable os-segment-table-os-svc-tyi-handler-path1.

THEOREM: os-readyq-os-svc-tyi-handler-path1
good-tm(os)
 \rightarrow (os-readyq(os-svc-tyi-handler-path1(os)) = array-deq(os-readyq(os)))

EVENT: Disable os-readyq-os-svc-tyi-handler-path1.

THEOREM: os-ibuffers-os-svc-tyi-handler-path1
good-tm(os)
 \rightarrow (os-ibuffers(os-svc-tyi-handler-path1(os)) = os-ibuffers(os))

EVENT: Disable os-ibuffers-os-svc-tyi-handler-path1.

THEOREM: os-obuffers-os-svc-tyi-handler-path1

good-tm (*os*)

→ (os-obuffers (os-svc-tyi-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-tyi-handler-path1.

THEOREM: os-mbuffers-os-svc-tyi-handler-path1

good-tm (*os*)

→ (os-mbuffers (os-svc-tyi-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyi-handler-path1.

THEOREM: os-status-table-os-svc-tyi-handler-path1

good-tm (*os*)

→ (os-status-table (os-svc-tyi-handler-path1 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)), '5, '0, *os*))

EVENT: Disable os-status-table-os-svc-tyi-handler-path1.

THEOREM: os-ready-set-os-svc-tyi-handler-path1

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))

→ (os-ready-set (os-svc-tyi-handler-path1 (*os*))
= index-ready-set ('0,
putnth ('(5 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-svc-tyi-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path1

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))

→ permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-tyi-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path1.

THEOREM: tm-waiting-os-svc-tyi-handler-path1

tm-waiting (os-svc-tyi-handler-path1 (*os*))

EVENT: Disable tm-waiting-os-svc-tyi-handler-path1.

THEOREM: good-os-os-svc-tyi-handler-path1

$$\begin{aligned} & \text{(good-os (os))} \\ & \wedge ((\neg \text{tm-waiting (os)}) \\ & \quad \wedge ((\neg \text{tm-errorp (os)}) \\ & \quad \quad \wedge (((\text{tm-svcid (os) mod '4} = '3) \\ & \quad \quad \quad \wedge (\text{array-qempty (os-current-ibuffer (os-current-taskid (os),} \\ & \quad \quad \quad \quad \text{os})) \\ & \quad \quad \quad \quad \wedge \text{array-qempty (array-deq (os-readyq (os)))))))) \\ \rightarrow & \text{good-os (os-svc-tyi-handler-path1 (os))} \end{aligned}$$

EVENT: Disable good-os-os-svc-tyi-handler-path1.

THEOREM: good-tm-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{(good-os (os))} \\ & \wedge ((\neg \text{tm-waiting (os)}) \\ & \quad \wedge ((\neg \text{tm-errorp (os)}) \\ & \quad \quad \wedge (((\text{tm-svcid (os) mod '4} = '3) \\ & \quad \quad \quad \wedge (\text{array-qempty (os-current-ibuffer (os-current-taskid (os),} \\ & \quad \quad \quad \quad \text{os})) \\ & \quad \quad \quad \quad \wedge (\neg \text{array-qempty (array-deq (os-readyq (os)))))))) \\ \rightarrow & \text{good-tm (os-svc-tyi-handler-path2 (os))} \end{aligned}$$

EVENT: Disable good-tm-os-svc-tyi-handler-path2.

THEOREM: tm-slimit-os-svc-tyi-handler-path2

$$\text{tm-slimit (os-svc-tyi-handler-path2 (os))} = \text{tm-slimit (os)}$$

EVENT: Disable tm-slimit-os-svc-tyi-handler-path2.

THEOREM: os-code-os-svc-tyi-handler-path2

$$\text{good-tm (os)} \rightarrow \text{os-code (os-svc-tyi-handler-path2 (os))} = \text{os-code (os)}$$

EVENT: Disable os-code-os-svc-tyi-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{good-tm (os)} \\ \rightarrow & ((\text{getnth ('3, tm-memory (os-svc-tyi-handler-path2 (os))})} \\ & \quad = \text{getnth ('3, tm-memory (os))}) \\ & \wedge ((\text{getnth ('4, tm-memory (os-svc-tyi-handler-path2 (os))})} \\ & \quad = \text{getnth ('4, tm-memory (os))}) \\ & \wedge ((\text{getnth ('5, tm-memory (os-svc-tyi-handler-path2 (os))})} \\ & \quad = \text{getnth ('5, tm-memory (os))}) \\ & \wedge ((\text{getnth ('6,} \end{aligned}$$

$$\begin{aligned}
& \text{tm-memory (os-svc-tyi-handler-path2 (os))} \\
= & \text{getnth ('6, tm-memory (os))} \\
\wedge & ((\text{getnth ('7,} \\
& \quad \text{tm-memory (os-svc-tyi-handler-path2 (os))} \\
= & \text{getnth ('7, tm-memory (os))} \\
\wedge & (\text{getnth ('11,} \\
& \quad \text{tm-memory (os-svc-tyi-handler-path2 (os))} \\
= & \text{getnth ('11,} \\
& \quad \text{tm-memory (os))}))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path2
 \neg tm-in-supervisor-mode (os-svc-tyi-handler-path2 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyi-handler-path2.

THEOREM: os-task-table-os-svc-tyi-handler-path2
good-tm (os)
 \rightarrow (os-task-table (os-svc-tyi-handler-path2 (os))
= os-new-task-table (os))

EVENT: Disable os-task-table-os-svc-tyi-handler-path2.

THEOREM: os-segment-table-os-svc-tyi-handler-path2
good-tm (os)
 \rightarrow (os-segment-table (os-svc-tyi-handler-path2 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-svc-tyi-handler-path2.

THEOREM: os-readyq-os-svc-tyi-handler-path2
good-tm (os)
 \rightarrow (os-readyq (os-svc-tyi-handler-path2 (os)) = array-deq (os-readyq (os)))

EVENT: Disable os-readyq-os-svc-tyi-handler-path2.

THEOREM: os-ibuffers-os-svc-tyi-handler-path2
good-tm (os)
 \rightarrow (os-ibuffers (os-svc-tyi-handler-path2 (os)) = os-ibuffers (os))

EVENT: Disable os-ibuffers-os-svc-tyi-handler-path2.

THEOREM: os-obuffers-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & \text{os-obuffers}(\text{os-svc-tyi-handler-path2}(os)) = \text{os-obuffers}(os) \end{aligned}$$

EVENT: Disable os-obuffers-os-svc-tyi-handler-path2.

THEOREM: os-mbuffers-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & \text{os-mbuffers}(\text{os-svc-tyi-handler-path2}(os)) = \text{os-mbuffers}(os) \end{aligned}$$

EVENT: Disable os-mbuffers-os-svc-tyi-handler-path2.

THEOREM: os-status-table-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & \text{os-status-table}(\text{os-svc-tyi-handler-path2}(os)) \\ & = \text{os-update-status}(\text{array-qfirst}(\text{os-readyq}(os)), '5, '0, os) \end{aligned}$$

EVENT: Disable os-status-table-os-svc-tyi-handler-path2.

THEOREM: os-ready-set-os-svc-tyi-handler-path2

$$\begin{aligned} & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ \rightarrow & \text{os-ready-set}(\text{os-svc-tyi-handler-path2}(os)) \\ & = \text{index-ready-set}('0, \\ & \quad \text{putnth}('5\ 0), \\ & \quad \text{array-qfirst}(\text{os-readyq}(os)), \\ & \quad \text{table}('2, \text{os-status-table}(os))) \end{aligned}$$

EVENT: Disable os-ready-set-os-svc-tyi-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path2

$$\begin{aligned} & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ \rightarrow & \text{permutation}(\text{mapup-queue}(\text{array-deq}(\text{os-readyq}(os))), \\ & \quad \text{os-ready-set}(\text{os-svc-tyi-handler-path2}(os))) \end{aligned}$$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path2.

THEOREM: tm-base-os-svc-tyi-handler-path2

$$\begin{aligned} & \text{tm-base}(\text{os-svc-tyi-handler-path2}(os)) \\ = & \text{os-new-base}(\text{array-qfirst}(\text{array-deq}(\text{os-readyq}(os))), os) \end{aligned}$$

EVENT: Disable tm-base-os-svc-tyi-handler-path2.

THEOREM: tm-limit-os-svc-tyi-handler-path2

$$\text{tm-limit}(\text{os-svc-tyi-handler-path2}(os))$$

$$= \text{os-new-limit}(\text{array-qfirst}(\text{array-deq}(\text{os-readyq}(os))), os)$$

EVENT: Disable tm-limit-os-svc-tyi-handler-path2.

THEOREM: tm-base-tm-limit-os-svc-tyi-handler-path2

$$\text{good-tm}(os)$$

$$\rightarrow ((\text{tm-base}(\text{os-svc-tyi-handler-path2}(os))$$

$$= \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path2}(os)),$$

$$\text{table}('2, \text{os-segment-table}(os))))))$$

$$\wedge (\text{tm-limit}(\text{os-svc-tyi-handler-path2}(os))$$

$$= \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path2}(os)),$$

$$\text{table}('2, \text{os-segment-table}(os))))))$$

EVENT: Disable tm-base-tm-limit-os-svc-tyi-handler-path2.

THEOREM: not-tm-waiting-os-svc-tyi-handler-path2

$$\neg \text{tm-waiting}(\text{os-svc-tyi-handler-path2}(os))$$

EVENT: Disable not-tm-waiting-os-svc-tyi-handler-path2.

THEOREM: good-os-os-svc-tyi-handler-path2

$$(\text{good-os}(os)$$

$$\wedge ((\neg \text{tm-waiting}(os))$$

$$\wedge ((\neg \text{tm-errorp}(os))$$

$$\wedge (((\text{tm-svcid}(os) \bmod '4) = '3)$$

$$\wedge (\text{array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os),$$

$$os))))))$$

$$\wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))$$

$$\rightarrow \text{good-os}(\text{os-svc-tyi-handler-path2}(os))$$

EVENT: Disable good-os-os-svc-tyi-handler-path2.

THEOREM: good-tm-os-svc-tyi-handler-path3

$$(\text{good-os}(os)$$

$$\wedge ((\neg \text{tm-waiting}(os))$$

$$\wedge ((\neg \text{tm-errorp}(os))$$

$$\wedge (((\text{tm-svcid}(os) \bmod '4) = '3)$$

$$\wedge (\neg \text{array-qempty}(\text{os-current-ibuffer}(\text{os-current-taskid}(os),$$

$$os))))))$$

$$\rightarrow \text{good-tm}(\text{os-svc-tyi-handler-path3}(os))$$

EVENT: Disable good-tm-os-svc-tyi-handler-path3.

THEOREM: tm-slimit-os-svc-tyi-handler-path3
 $\text{tm-slimit}(\text{os-svc-tyi-handler-path3}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-svc-tyi-handler-path3.

THEOREM: os-code-os-svc-tyi-handler-path3
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-tyi-handler-path3}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-svc-tyi-handler-path3.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\quad \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\quad \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\quad \text{tm-memory}(\text{os-svc-tyi-handler-path3}(os)))$
 $= \text{getnth}('11,$
 $\quad \text{tm-memory}(os))))))))))$

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path3
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-tyi-handler-path3}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyi-handler-path3.

THEOREM: os-task-table-os-svc-tyi-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-tyi-handler-path3}(os))$
 $= \text{os-task-table-with-delivered-character}(os))$

EVENT: Disable os-task-table-os-svc-tyi-handler-path3.

THEOREM: good-cpu-list-os-task-table-with-delivered-character
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (\neg array-qempty (os-current-ibuffer (os-current-taskid (*os*),
 os))))))
 \rightarrow good-cpu-list (table ('9, os-task-table-with-delivered-character (*os*)))

EVENT: Disable good-cpu-list-os-task-table-with-delivered-character.

THEOREM: os-segment-table-os-svc-tyi-handler-path3
good-tm (*os*)
 \rightarrow (os-segment-table (os-svc-tyi-handler-path3 (*os*))
 = os-segment-table (*os*))

EVENT: Disable os-segment-table-os-svc-tyi-handler-path3.

THEOREM: os-readyq-os-svc-tyi-handler-path3
good-tm (*os*) \rightarrow (os-readyq (os-svc-tyi-handler-path3 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-svc-tyi-handler-path3.

THEOREM: os-ibuffers-os-svc-tyi-handler-path3
good-tm (*os*)
 \rightarrow (os-ibuffers (os-svc-tyi-handler-path3 (*os*))
 = os-ibuffers-with-dequeued-character (*os*))

EVENT: Disable os-ibuffers-os-svc-tyi-handler-path3.

THEOREM: finite-number-queue-listp-os-ibuffers-with-dequeued-character
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (\neg array-qempty (os-current-ibuffer (os-current-taskid (*os*),
 os))))))
 \rightarrow finite-number-queue-listp (table ('8,
 os-ibuffers-with-dequeued-character (*os*)),
 '4,
 '65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-with-dequeued-character.

THEOREM: os-obuffers-os-svc-tyi-handler-path3

$$\text{good-tm}(os) \rightarrow (\text{os-obuffers}(\text{os-svc-tyi-handler-path3}(os)) = \text{os-obuffers}(os))$$

EVENT: Disable os-obuffers-os-svc-tyi-handler-path3.

THEOREM: os-mbuffers-os-svc-tyi-handler-path3

$$\text{good-tm}(os) \rightarrow (\text{os-mbuffers}(\text{os-svc-tyi-handler-path3}(os)) = \text{os-mbuffers}(os))$$

EVENT: Disable os-mbuffers-os-svc-tyi-handler-path3.

THEOREM: os-status-table-os-svc-tyi-handler-path3

$$\begin{aligned} \text{good-tm}(os) \\ \rightarrow (\text{os-status-table}(\text{os-svc-tyi-handler-path3}(os)) \\ = \text{os-status-table}(os)) \end{aligned}$$

EVENT: Disable os-status-table-os-svc-tyi-handler-path3.

THEOREM: not-tm-waiting-os-svc-tyi-handler-path3

$$\neg \text{tm-waiting}(\text{os-svc-tyi-handler-path3}(os))$$

EVENT: Disable not-tm-waiting-os-svc-tyi-handler-path3.

THEOREM: tm-base-os-svc-tyi-handler-path3

$$\text{tm-base}(\text{os-svc-tyi-handler-path3}(os)) = \text{tm-base}(os)$$

EVENT: Disable tm-base-os-svc-tyi-handler-path3.

THEOREM: tm-limit-os-svc-tyi-handler-path3

$$\text{tm-limit}(\text{os-svc-tyi-handler-path3}(os)) = \text{tm-limit}(os)$$

EVENT: Disable tm-limit-os-svc-tyi-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-tyi-handler-path3

$$\begin{aligned} & (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\ \rightarrow & ((\text{tm-base}(\text{os-svc-tyi-handler-path3}(os)) \\ & = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path3}(os)), \\ & \quad \text{table}('2, \text{os-segment-table}(os)))))) \\ \wedge & (\text{tm-limit}(\text{os-svc-tyi-handler-path3}(os)) \\ & = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path3}(os)), \\ & \quad \text{table}('2, \text{os-segment-table}(os)))))) \end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-svc-tyi-handler-path3.

THEOREM: good-os-os-svc-tyi-handler-path3

$$\begin{aligned} & (\text{good-os } (os) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\neg \text{tm-errorp } (os)) \\ & \quad \quad \quad \wedge (((\text{tm-svcid } (os) \mathbf{mod } '4) = '3) \\ & \quad \quad \quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-ibuffer } (\text{os-current-taskid } (os), \\ & \quad \quad \quad \quad \quad \quad \quad \quad os)))))) \\ & \rightarrow \text{good-os } (\text{os-svc-tyi-handler-path3 } (os)) \end{aligned}$$

EVENT: Disable good-os-os-svc-tyi-handler-path3.

THEOREM: good-os-os-svc-tyi-handler

$$\begin{aligned} & (\text{good-os } (os) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\neg \text{tm-errorp } (os)) \wedge ((\text{tm-svcid } (os) \mathbf{mod } '4) = '3)))) \\ & \rightarrow \text{good-os } (\text{os-svc-tyi-handler } (os)) \end{aligned}$$

EVENT: Disable good-os-os-svc-tyi-handler.

THEOREM: good-tm-os-intended-svc-tyo-interrupt

$$\begin{aligned} & (\text{good-os } (os) \wedge ((\neg \text{tm-waiting } (os)) \wedge (\neg \text{tm-errorp } (os)))) \\ & \rightarrow \text{good-tm } (\text{os-intended-svc-tyo-interrupt } (os)) \end{aligned}$$

EVENT: Disable good-tm-os-intended-svc-tyo-interrupt.

THEOREM: good-tm-os-svc-tyo-handler-path1

$$\begin{aligned} & (\text{good-os } (os) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\neg \text{tm-errorp } (os)) \\ & \quad \quad \quad \wedge (((\text{tm-svcid } (os) \mathbf{mod } '4) = '2) \\ & \quad \quad \quad \quad \wedge (\text{array-qfullp } (\text{os-current-obuffer } (\text{os-current-taskid } (os), \\ & \quad \quad \quad \quad \quad \quad \quad \quad os)) \\ & \quad \quad \quad \quad \quad \quad \quad \wedge \text{array-qempty } (\text{array-deq } (\text{os-readyq } (os)))))) \\ & \rightarrow \text{good-tm } (\text{os-svc-tyo-handler-path1 } (os)) \end{aligned}$$

EVENT: Disable good-tm-os-svc-tyo-handler-path1.

THEOREM: tm-slimit-os-svc-tyo-handler-path1

$$\text{tm-slimit } (\text{os-svc-tyo-handler-path1 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-svc-tyo-handler-path1.

THEOREM: os-code-os-svc-tyo-handler-path1
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-svc-tyo-handler-path1}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-svc-tyo-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path1
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\quad \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\quad \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\quad \text{tm-memory}(\text{os-svc-tyo-handler-path1}(os)))$
 $= \text{getnth}('11,$
 $\quad \text{tm-memory}(os)))))))))$

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path1
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-tyo-handler-path1}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyo-handler-path1.

THEOREM: os-task-table-os-svc-tyo-handler-path1
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-tyo-handler-path1}(os))$
 $= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-svc-tyo-handler-path1.

THEOREM: os-segment-table-os-svc-tyo-handler-path1
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-segment-table}(\text{os-svc-tyo-handler-path1}(os))$
 $= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-svc-tyo-handler-path1.

THEOREM: os-readyq-os-svc-tyo-handler-path1

good-tm (*os*)

→ (os-readyq (os-svc-tyo-handler-path1 (*os*)) = array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-tyo-handler-path1.

THEOREM: os-ibuffers-os-svc-tyo-handler-path1

good-tm (*os*)

→ (os-ibuffers (os-svc-tyo-handler-path1 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-tyo-handler-path1.

THEOREM: os-obuffers-os-svc-tyo-handler-path1

good-tm (*os*)

→ (os-obuffers (os-svc-tyo-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-tyo-handler-path1.

THEOREM: os-mbuffers-os-svc-tyo-handler-path1

good-tm (*os*)

→ (os-mbuffers (os-svc-tyo-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyo-handler-path1.

THEOREM: os-status-table-os-svc-tyo-handler-path1

good-tm (*os*)

→ (os-status-table (os-svc-tyo-handler-path1 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)), '4, '0, *os*))

EVENT: Disable os-status-table-os-svc-tyo-handler-path1.

THEOREM: os-ready-set-os-svc-tyo-handler-path1

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))

→ (os-ready-set (os-svc-tyo-handler-path1 (*os*))
= index-ready-set ('0,
putnth ('4 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable os-ready-set-os-svc-tyo-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-tyo-handler-path1
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 \rightarrow permutation (mapup-queue (array-deq (os-readyq (os))),
os-ready-set (os-svc-tyo-handler-path1 (os)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-tyo-handler-path1.

THEOREM: tm-waiting-os-svc-tyo-handler-path1
 $\text{tm-waiting}(os\text{-svc-tyo-handler-path1}(os))$

EVENT: Disable tm-waiting-os-svc-tyo-handler-path1.

THEOREM: good-os-os-svc-tyo-handler-path1
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge (((\text{tm-svcid}(os) \bmod '4) = '2)$
 $\wedge (\text{array-qfullp}(\text{os-current-obuffer}(\text{os-current-taskid}(os),$
 $os)))$
 $\wedge \text{array-qemptyyp}(\text{array-deq}(\text{os-readyq}(os))))))$
 \rightarrow good-os (os-svc-tyo-handler-path1 (os))

EVENT: Disable good-os-os-svc-tyo-handler-path1.

THEOREM: good-tm-os-svc-tyo-handler-path2
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge (((\text{tm-svcid}(os) \bmod '4) = '2)$
 $\wedge (\text{array-qfullp}(\text{os-current-obuffer}(\text{os-current-taskid}(os),$
 $os)))$
 $\wedge (\neg \text{array-qemptyyp}(\text{array-deq}(\text{os-readyq}(os))))))$
 \rightarrow good-tm (os-svc-tyo-handler-path2 (os))

EVENT: Disable good-tm-os-svc-tyo-handler-path2.

THEOREM: tm-slimit-os-svc-tyo-handler-path2
 $\text{tm-slimit}(os\text{-svc-tyo-handler-path2}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-svc-tyo-handler-path2.

THEOREM: os-code-os-svc-tyo-handler-path2
 $\text{good-tm}(os) \rightarrow (\text{os-code}(os\text{-svc-tyo-handler-path2}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-svc-tyo-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-svc-tyo-handler-path2}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path2

$$\rightarrow \text{tm-in-supervisor-mode}(\text{os-svc-tyo-handler-path2}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyo-handler-path2.

THEOREM: os-task-table-os-svc-tyo-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-svc-tyo-handler-path2}(os)) \\ & = \text{os-new-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-svc-tyo-handler-path2.

THEOREM: os-segment-table-os-svc-tyo-handler-path2

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-svc-tyo-handler-path2}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-svc-tyo-handler-path2.

THEOREM: os-readyq-os-svc-tyo-handler-path2

good-tm (*os*)
 \rightarrow (os-readyq (os-svc-tyo-handler-path2 (*os*)) = array-deq (os-readyq (*os*)))

EVENT: Disable os-readyq-os-svc-tyo-handler-path2.

THEOREM: os-ibuffers-os-svc-tyo-handler-path2
good-tm (*os*)
 \rightarrow (os-ibuffers (os-svc-tyo-handler-path2 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-tyo-handler-path2.

THEOREM: os-obuffers-os-svc-tyo-handler-path2
good-tm (*os*)
 \rightarrow (os-obuffers (os-svc-tyo-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-tyo-handler-path2.

THEOREM: os-mbuffers-os-svc-tyo-handler-path2
good-tm (*os*)
 \rightarrow (os-mbuffers (os-svc-tyo-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyo-handler-path2.

THEOREM: os-status-table-os-svc-tyo-handler-path2
good-tm (*os*)
 \rightarrow (os-status-table (os-svc-tyo-handler-path2 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)), '4, '0, *os*))

EVENT: Disable os-status-table-os-svc-tyo-handler-path2.

THEOREM: os-ready-set-os-svc-tyo-handler-path2
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (os-ready-set (os-svc-tyo-handler-path2 (*os*))
= index-ready-set ('0,
putnth ('(4 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable os-ready-set-os-svc-tyo-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-tyo-handler-path2
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-tyo-handler-path2 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-tyo-handler-path2.

THEOREM: tm-base-os-svc-tyo-handler-path2
tm-base (os-svc-tyo-handler-path2 (os))
= os-new-base (array-qfirst (array-deq (os-readyq (os))), os)

EVENT: Disable tm-base-os-svc-tyo-handler-path2.

THEOREM: tm-limit-os-svc-tyo-handler-path2
tm-limit (os-svc-tyo-handler-path2 (os))
= os-new-limit (array-qfirst (array-deq (os-readyq (os))), os)

EVENT: Disable tm-limit-os-svc-tyo-handler-path2.

THEOREM: tm-base-tm-limit-os-svc-tyo-handler-path2
good-tm (os)
→ ((tm-base (os-svc-tyo-handler-path2 (os))
= base (getnth (os-current-taskid (os-svc-tyo-handler-path2 (os)),
table ('2, os-segment-table (os))))))
∧ (tm-limit (os-svc-tyo-handler-path2 (os))
= limit (getnth (os-current-taskid (os-svc-tyo-handler-path2 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-svc-tyo-handler-path2.

THEOREM: not-tm-waiting-os-svc-tyo-handler-path2
¬ tm-waiting (os-svc-tyo-handler-path2 (os))

EVENT: Disable not-tm-waiting-os-svc-tyo-handler-path2.

THEOREM: good-os-os-svc-tyo-handler-path2
(good-os (os)
∧ ((¬ tm-waiting (os))
∧ ((¬ tm-errorp (os))
∧ (((tm-svcid (os) mod '4) = '2)
∧ (array-qfullp (os-current-obuffer (os-current-taskid (os),
os))
∧ (¬ array-qempty (array-deq (os-readyq (os))))))))))
→ good-os (os-svc-tyo-handler-path2 (os))

EVENT: Disable good-os-os-svc-tyo-handler-path2.

THEOREM: good-tm-os-svc-tyo-handler-path3
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
 ∧ ((¬ tm-errorp (*os*))
 ∧ (((tm-svcid (*os*) **mod** '4) = '2)
 ∧ ((¬ array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os)))
 ∧ tm-oport-idlep (array-qfirst (os-readyq (*os*),
 tm-oports (*os*))))))))))
 → good-tm (os-svc-tyo-handler-path3 (*os*))

EVENT: Disable good-tm-os-svc-tyo-handler-path3.

THEOREM: tm-slimit-os-svc-tyo-handler-path3
 tm-slimit (os-svc-tyo-handler-path3 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyo-handler-path3.

THEOREM: os-code-os-svc-tyo-handler-path3
 good-tm (*os*) → (os-code (os-svc-tyo-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyo-handler-path3.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path3
 good-tm (*os*)
 → ((getnth ('3, tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('4, tm-memory (*os*)))
 ∧ ((getnth ('5, tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('5, tm-memory (*os*)))
 ∧ ((getnth ('6,
 tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('6, tm-memory (*os*)))
 ∧ ((getnth ('7,
 tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('7, tm-memory (*os*)))
 ∧ (getnth ('11,
 tm-memory (os-svc-tyo-handler-path3 (*os*)))
 = getnth ('11,
 tm-memory (*os*))))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path3
 $\neg \text{tm-in-supervisor-mode}(\text{os-svc-tyo-handler-path3}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyo-handler-path3.

THEOREM: os-task-table-os-svc-tyo-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-svc-tyo-handler-path3}(os))$
 $= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-svc-tyo-handler-path3.

THEOREM: os-segment-table-os-svc-tyo-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-segment-table}(\text{os-svc-tyo-handler-path3}(os))$
 $= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-svc-tyo-handler-path3.

THEOREM: os-readyq-os-svc-tyo-handler-path3
 $\text{good-tm}(os) \rightarrow (\text{os-readyq}(\text{os-svc-tyo-handler-path3}(os)) = \text{os-readyq}(os))$

EVENT: Disable os-readyq-os-svc-tyo-handler-path3.

THEOREM: os-ibuffers-os-svc-tyo-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-ibuffers}(\text{os-svc-tyo-handler-path3}(os)) = \text{os-ibuffers}(os))$

EVENT: Disable os-ibuffers-os-svc-tyo-handler-path3.

THEOREM: os-obuffers-os-svc-tyo-handler-path3
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-obuffers}(\text{os-svc-tyo-handler-path3}(os))$
 $= \text{os-obuffers-with-enqueued-character}(os))$

EVENT: Disable os-obuffers-os-svc-tyo-handler-path3.

THEOREM: finite-number-queue-listp-os-obuffers-with-enqueued-character
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\neg \text{array-qfullp}(\text{os-current-obuffer}(\text{os-current-taskid}(os),$
 $os))))))$

→ finite-number-queue-listp (table ('8,
os-obuffers-with-enqueued-character (os)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-obuffers-with-enqueued-character.

THEOREM: os-mbuffers-os-svc-tyo-handler-path3
good-tm (os)
→ (os-mbuffers (os-svc-tyo-handler-path3 (os)) = os-mbuffers (os))

EVENT: Disable os-mbuffers-os-svc-tyo-handler-path3.

THEOREM: os-status-table-os-svc-tyo-handler-path3
good-tm (os)
→ (os-status-table (os-svc-tyo-handler-path3 (os))
= os-status-table (os))

EVENT: Disable os-status-table-os-svc-tyo-handler-path3.

THEOREM: not-tm-waiting-os-svc-tyo-handler-path3
¬ tm-waiting (os-svc-tyo-handler-path3 (os))

EVENT: Disable not-tm-waiting-os-svc-tyo-handler-path3.

THEOREM: tm-base-os-svc-tyo-handler-path3
tm-base (os-svc-tyo-handler-path3 (os)) = tm-base (os)

EVENT: Disable tm-base-os-svc-tyo-handler-path3.

THEOREM: tm-limit-os-svc-tyo-handler-path3
tm-limit (os-svc-tyo-handler-path3 (os)) = tm-limit (os)

EVENT: Disable tm-limit-os-svc-tyo-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-tyo-handler-path3
(good-os (os) ∧ (¬ tm-waiting (os)))
→ (((tm-base (os-svc-tyo-handler-path3 (os))
= base (getnth (os-current-taskid (os-svc-tyo-handler-path3 (os)),
table ('2, os-segment-table (os))))))
∧ (tm-limit (os-svc-tyo-handler-path3 (os))
= limit (getnth (os-current-taskid (os-svc-tyo-handler-path3 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-svc-tyo-handler-path3.

THEOREM: good-os-os-svc-tyo-handler-path3

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{tm-errorp } (os)) \\ & \quad \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '2) \\ & \quad \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-obuffer } (\text{os-current-taskid } (os), \\ & \quad \quad \quad \quad \quad \quad \quad os)))) \\ & \quad \quad \quad \wedge \text{tm-oport-idlep } (\text{array-qfirst } (\text{os-readyq } (os), \\ & \quad \quad \quad \quad \quad \quad \quad \text{tm-oports } (os)))))))) \\ \rightarrow & \text{good-os } (\text{os-svc-tyo-handler-path3 } (os)) \end{aligned}$$

EVENT: Disable good-os-os-svc-tyo-handler-path3.

THEOREM: good-tm-os-svc-tyo-handler-path4

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{tm-errorp } (os)) \\ & \quad \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '2) \\ & \quad \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-obuffer } (\text{os-current-taskid } (os), \\ & \quad \quad \quad \quad \quad \quad \quad os)))) \\ & \quad \quad \quad \wedge (\neg \text{tm-oport-idlep } (\text{array-qfirst } (\text{os-readyq } (os), \\ & \quad \quad \quad \quad \quad \quad \quad \text{tm-oports } (os)))))))) \\ \rightarrow & \text{good-tm } (\text{os-svc-tyo-handler-path4 } (os)) \end{aligned}$$

EVENT: Disable good-tm-os-svc-tyo-handler-path4.

THEOREM: tm-slimit-os-svc-tyo-handler-path4

$$\text{tm-slimit } (\text{os-svc-tyo-handler-path4 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-svc-tyo-handler-path4.

THEOREM: os-code-os-svc-tyo-handler-path4

$$\text{good-tm } (os) \rightarrow (\text{os-code } (\text{os-svc-tyo-handler-path4 } (os)) = \text{os-code } (os))$$

EVENT: Disable os-code-os-svc-tyo-handler-path4.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path4

$$\begin{aligned} & \text{good-tm } (os) \\ \rightarrow & ((\text{getnth } ('3, \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\ & \quad = \text{getnth } ('3, \text{tm-memory } (os))) \end{aligned}$$

$$\begin{aligned}
& \wedge ((\text{getnth } '4, \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
& \quad = \text{getnth } '4, \text{tm-memory } (os)) \\
& \wedge ((\text{getnth } '5, \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
& \quad = \text{getnth } '5, \text{tm-memory } (os)) \\
& \wedge ((\text{getnth } '6, \\
& \quad \quad \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
& \quad = \text{getnth } '6, \text{tm-memory } (os)) \\
& \wedge ((\text{getnth } '7, \\
& \quad \quad \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
& \quad = \text{getnth } '7, \text{tm-memory } (os)) \\
& \wedge (\text{getnth } '11, \\
& \quad \quad \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
& \quad = \text{getnth } '11, \\
& \quad \quad \text{tm-memory } (os))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path4
 $\neg \text{tm-in-supervisor-mode } (\text{os-svc-tyo-handler-path4 } (os))$

EVENT: Disable not-tm-in-supervisor-mode-os-svc-tyo-handler-path4.

THEOREM: os-task-table-os-svc-tyo-handler-path4
good-tm (os)
 $\rightarrow (\text{os-task-table } (\text{os-svc-tyo-handler-path4 } (os)))$
 $= \text{os-new-task-table } (os)$

EVENT: Disable os-task-table-os-svc-tyo-handler-path4.

THEOREM: os-segment-table-os-svc-tyo-handler-path4
good-tm (os)
 $\rightarrow (\text{os-segment-table } (\text{os-svc-tyo-handler-path4 } (os)))$
 $= \text{os-segment-table } (os)$

EVENT: Disable os-segment-table-os-svc-tyo-handler-path4.

THEOREM: os-readyq-os-svc-tyo-handler-path4
good-tm (os) $\rightarrow (\text{os-readyq } (\text{os-svc-tyo-handler-path4 } (os))) = \text{os-readyq } (os)$

EVENT: Disable os-readyq-os-svc-tyo-handler-path4.

THEOREM: os-ibuffers-os-svc-tyo-handler-path4
good-tm (os)
 $\rightarrow (\text{os-ibuffers } (\text{os-svc-tyo-handler-path4 } (os))) = \text{os-ibuffers } (os)$

EVENT: Disable os-ibuffers-os-svc-tyo-handler-path4.

THEOREM: os-obuffers-os-svc-tyo-handler-path4
good-tm (*os*)
→ (os-obuffers (os-svc-tyo-handler-path4 (*os*))
= os-obuffers-with-enqueued-character (*os*))

EVENT: Disable os-obuffers-os-svc-tyo-handler-path4.

THEOREM: os-mbuffers-os-svc-tyo-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-svc-tyo-handler-path4 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyo-handler-path4.

THEOREM: os-status-table-os-svc-tyo-handler-path4
good-tm (*os*)
→ (os-status-table (os-svc-tyo-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-svc-tyo-handler-path4.

THEOREM: not-tm-waiting-os-svc-tyo-handler-path4
¬ tm-waiting (os-svc-tyo-handler-path4 (*os*))

EVENT: Disable not-tm-waiting-os-svc-tyo-handler-path4.

THEOREM: tm-base-os-svc-tyo-handler-path4
tm-base (os-svc-tyo-handler-path4 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-svc-tyo-handler-path4.

THEOREM: tm-limit-os-svc-tyo-handler-path4
tm-limit (os-svc-tyo-handler-path4 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-svc-tyo-handler-path4.

THEOREM: tm-base-tm-limit-os-svc-tyo-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-svc-tyo-handler-path4 (*os*))
= base (getnth (os-current-taskid (os-svc-tyo-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))

$$\begin{aligned} &\wedge \text{ (tm-limit (os-svc-tyo-handler-path4 (os))} \\ &\quad = \text{ limit (getnth (os-current-taskid (os-svc-tyo-handler-path4 (os)),} \\ &\quad \quad \text{table ('2, os-segment-table (os))))))} \end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-svc-tyo-handler-path4.

THEOREM: good-os-os-svc-tyo-handler-path4

$$\begin{aligned} &\text{(good-os (os)} \\ &\quad \wedge \text{ ((}\neg \text{ tm-waiting (os))} \\ &\quad \quad \wedge \text{ ((}\neg \text{ tm-errorp (os))} \\ &\quad \quad \quad \wedge \text{ (((tm-svcid (os) mod '4) = '2)} \\ &\quad \quad \quad \quad \wedge \text{ ((}\neg \text{ array-qfullp (os-current-obuffer (os-current-taskid (os),} \\ &\quad \quad \quad \quad \quad \text{os))}} \\ &\quad \quad \quad \quad \quad \quad \wedge \text{ ((}\neg \text{ tm-oport-idlep (array-qfirst (os-readyq (os),} \\ &\quad \quad \quad \quad \quad \quad \quad \text{tm-oports (os))))))))) \\ &\rightarrow \text{ good-os (os-svc-tyo-handler-path4 (os))} \end{aligned}$$

EVENT: Disable good-os-os-svc-tyo-handler-path4.

THEOREM: good-os-os-svc-tyo-handler

$$\begin{aligned} &\text{(good-os (os)} \\ &\quad \wedge \text{ ((}\neg \text{ tm-waiting (os))} \\ &\quad \quad \wedge \text{ ((}\neg \text{ tm-errorp (os))} \wedge \text{ ((tm-svcid (os) mod '4) = '2))))) \\ &\rightarrow \text{ good-os (os-svc-tyo-handler (os))} \end{aligned}$$

EVENT: Disable good-os-os-svc-tyo-handler.

THEOREM: good-tm-os-intended-input-interrupt

$$\begin{aligned} &\text{(good-os (os) } \wedge \text{ tm-some-input-interruptp (tm-iports (os))} \\ &\rightarrow \text{ good-tm (os-intended-input-interrupt (os))} \end{aligned}$$

EVENT: Disable good-tm-os-intended-input-interrupt.

THEOREM: finite-number-queuep-array-enq-interrupting-input-port

$$\begin{aligned} &\text{(good-os (os)} \\ &\quad \wedge \text{ (tm-waiting (os) } \wedge \text{ tm-some-input-interruptp (tm-iports (os))}) \\ &\rightarrow \text{ finite-number-queuep (array-enq (tm-interrupting-input-port (tm-iports (os)),} \\ &\quad \quad \text{os-readyq (os)),} \\ &\quad \quad \quad '16, \\ &\quad \quad \quad '16)} \end{aligned}$$

EVENT: Disable finite-number-queuep-array-enq-interrupting-input-port.

THEOREM: good-tm-os-waiting-input-handler-path1

(good-os (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ (tm-waiting (*os*)
 ∧ ((getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 = '5)
 ∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
 os))))))
 → good-tm (os-waiting-input-handler-path1 (*os*))

EVENT: Disable good-tm-os-waiting-input-handler-path1.

THEOREM: tm-slimit-os-waiting-input-handler-path1

tm-slimit (os-waiting-input-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-input-handler-path1.

THEOREM: os-code-os-waiting-input-handler-path1

good-tm (*os*)
 → (os-code (os-waiting-input-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-input-handler-path1.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path1

good-tm (*os*)
 → ((getnth ('3, tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('4, tm-memory (*os*)))
 ∧ ((getnth ('5,
 tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('5, tm-memory (*os*)))
 ∧ ((getnth ('6,
 tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('6, tm-memory (*os*)))
 ∧ ((getnth ('7,
 tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('7, tm-memory (*os*)))
 ∧ (getnth ('11,
 tm-memory (os-waiting-input-handler-path1 (*os*)))
 = getnth ('11,
 tm-memory (*os*))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path1
→ tm-in-supervisor-mode (os-waiting-input-handler-path1 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path1.

THEOREM: os-task-table-os-waiting-input-handler-path1
good-tm (os)
→ (os-task-table (os-waiting-input-handler-path1 (os))
= os-task-table (os))

EVENT: Disable os-task-table-os-waiting-input-handler-path1.

THEOREM: os-segment-table-os-waiting-input-handler-path1
good-tm (os)
→ (os-segment-table (os-waiting-input-handler-path1 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-waiting-input-handler-path1.

THEOREM: os-readyq-os-waiting-input-handler-path1
good-tm (os)
→ (os-readyq (os-waiting-input-handler-path1 (os))
= array-enq (tm-interrupting-input-port (tm-iports (os)),
os-readyq (os)))

EVENT: Disable os-readyq-os-waiting-input-handler-path1.

THEOREM: os-ibuffers-os-waiting-input-handler-path1
good-tm (os)
→ (os-ibuffers (os-waiting-input-handler-path1 (os))
= os-ibuffers-with-qreplaced-overflow-character (os))

EVENT: Disable os-ibuffers-os-waiting-input-handler-path1.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path1
(good-os (os)
∧ (tm-some-input-interruptp (tm-iports (os))
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))
→ finite-number-queue-listp (table ('8,

$$\begin{aligned} & \text{os-ibuffers (os-waiting-input-handler-path1 (os))}, \\ & '4, \\ & '65536) \end{aligned}$$

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path1.

THEOREM: os-obuffers-os-waiting-input-handler-path1
good-tm (os)
 \rightarrow (os-obuffers (os-waiting-input-handler-path1 (os)) = os-obuffers (os))

EVENT: Disable os-obuffers-os-waiting-input-handler-path1.

THEOREM: os-mbuffers-os-waiting-input-handler-path1
good-tm (os)
 \rightarrow (os-mbuffers (os-waiting-input-handler-path1 (os)) = os-mbuffers (os))

EVENT: Disable os-mbuffers-os-waiting-input-handler-path1.

THEOREM: os-status-table-os-waiting-input-handler-path1
good-tm (os)
 \rightarrow (os-status-table (os-waiting-input-handler-path1 (os))
= os-update-status (tm-interrupting-input-port (tm-iports (os)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-waiting-input-handler-path1.

THEOREM: good-status-list-os-status-table-os-waiting-input-handler-path1
(good-os (os) \wedge tm-some-input-interruptp (tm-iports (os)))
 \rightarrow good-status-list (table ('2,
os-status-table (os-waiting-input-handler-path1 (os))))

EVENT: Disable good-status-list-os-status-table-os-waiting-input-handler-path1.

THEOREM: os-ready-set-os-waiting-input-handler-path1
(good-os (os) \wedge tm-some-input-interruptp (tm-iports (os)))
 \rightarrow (os-ready-set (os-waiting-input-handler-path1 (os))
= index-ready-set ('0,
putnth ('(0 0),
tm-interrupting-input-port (tm-iports (os)),
table ('2, os-status-table (os))))

EVENT: Disable os-ready-set-os-waiting-input-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge (getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 = '5))))))
 \rightarrow permutation (mapup-queue (array-enq (tm-interrupting-input-port (tm-iports (*os*)),
 os-readyq (*os*))),
 os-ready-set (os-waiting-input-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path1.

THEOREM: not-tm-waiting-os-waiting-input-handler-path1
 \neg tm-waiting (os-waiting-input-handler-path1 (*os*))

EVENT: Disable not-tm-waiting-os-waiting-input-handler-path1.

THEOREM: tm-base-os-waiting-input-handler-path1
tm-base (os-waiting-input-handler-path1 (*os*))
= base (getnth (tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-segment-table (*os*))))

EVENT: Disable tm-base-os-waiting-input-handler-path1.

THEOREM: tm-limit-os-waiting-input-handler-path1
tm-limit (os-waiting-input-handler-path1 (*os*))
= limit (getnth (tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-segment-table (*os*))))

EVENT: Disable tm-limit-os-waiting-input-handler-path1.

THEOREM: tm-base-tm-limit-os-waiting-input-handler-path1
(good-os (*os*) \wedge tm-waiting (*os*))
 \rightarrow ((tm-base (os-waiting-input-handler-path1 (*os*))
 = base (getnth (os-current-taskid (os-waiting-input-handler-path1 (*os*)),
 table ('2, os-segment-table (*os*))))))
 \wedge (tm-limit (os-waiting-input-handler-path1 (*os*)))

$$= \text{limit} (\text{getnth} (\text{os-current-taskid} (\text{os-waiting-input-handler-path1} (os)), \\ \text{table} ('2, \text{os-segment-table} (os))))))$$

EVENT: Disable tm-base-tm-limit-os-waiting-input-handler-path1.

THEOREM: good-os-os-waiting-input-handler-path1

$$\begin{aligned} & (\text{good-os} (os) \\ & \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \wedge (\text{tm-waiting} (os) \\ & \quad \quad \wedge ((\text{getnth} ('2 \\ & \quad \quad \quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os-status-table} (os)) \\ & \quad \quad = '5) \\ & \quad \quad \wedge \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os})))))) \\ \rightarrow & \text{good-os} (\text{os-waiting-input-handler-path1} (os)) \end{aligned}$$

EVENT: Disable good-os-os-waiting-input-handler-path1.

THEOREM: good-tm-os-waiting-input-handler-path2

$$\begin{aligned} & (\text{good-os} (os) \\ & \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \wedge (\text{tm-waiting} (os) \\ & \quad \quad \wedge ((\text{getnth} ('2 \\ & \quad \quad \quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os-status-table} (os)) \\ & \quad \quad = '5) \\ & \quad \quad \wedge ((\neg \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge \text{tm-iport-errorp} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad \text{tm-iports} (os)))))) \\ \rightarrow & \text{good-tm} (\text{os-waiting-input-handler-path2} (os)) \end{aligned}$$

EVENT: Disable good-tm-os-waiting-input-handler-path2.

THEOREM: tm-slimit-os-waiting-input-handler-path2

$$\text{tm-slimit} (\text{os-waiting-input-handler-path2} (os)) = \text{tm-slimit} (os)$$

EVENT: Disable tm-slimit-os-waiting-input-handler-path2.

THEOREM: os-code-os-waiting-input-handler-path2

$$\begin{aligned} & \text{good-tm} (os) \\ \rightarrow & (\text{os-code} (\text{os-waiting-input-handler-path2} (os)) = \text{os-code} (os)) \end{aligned}$$

EVENT: Disable os-code-os-waiting-input-handler-path2.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path2
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-waiting-input-handler-path2 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path2
¬ tm-in-supervisor-mode (os-waiting-input-handler-path2 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path2.

THEOREM: os-task-table-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-task-table (os-waiting-input-handler-path2 (*os*))
= os-task-table (*os*))

EVENT: Disable os-task-table-os-waiting-input-handler-path2.

THEOREM: os-segment-table-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-segment-table (os-waiting-input-handler-path2 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-waiting-input-handler-path2.

THEOREM: os-readyq-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-readyq (os-waiting-input-handler-path2 (*os*))
= array-enq (tm-interrupting-input-port (tm-iports (*os*)),
os-readyq (*os*)))

EVENT: Disable os-readyq-os-waiting-input-handler-path2.

THEOREM: os-ibuffers-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-ibuffers (os-waiting-input-handler-path2 (*os*))
= os-ibuffers-with-enqueued-overflow-character (*os*))

EVENT: Disable os-ibuffers-os-waiting-input-handler-path2.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path2
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-waiting-input-handler-path2 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path2.

THEOREM: os-obuffers-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-obuffers (os-waiting-input-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-input-handler-path2.

THEOREM: os-mbuffers-os-waiting-input-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-waiting-input-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-input-handler-path2.

THEOREM: os-status-table-os-waiting-input-handler-path2
good-tm (*os*)

$$\begin{aligned} \rightarrow & \text{(os-status-table (os-waiting-input-handler-path2 (os))} \\ & = \text{os-update-status (tm-interrupting-input-port (tm-iports (os)),} \\ & \quad '0,} \\ & \quad '0,} \\ & \quad os)) \end{aligned}$$

EVENT: Disable os-status-table-os-waiting-input-handler-path2.

THEOREM: good-status-list-os-status-table-os-waiting-input-handler-path2
 $(\text{good-os (os)} \wedge \text{tm-some-input-interruptp (tm-iports (os))})$
 $\rightarrow \text{good-status-list (table ('2, os-status-table (os-waiting-input-handler-path2 (os))))}$

EVENT: Disable good-status-list-os-status-table-os-waiting-input-handler-path2.

THEOREM: os-ready-set-os-waiting-input-handler-path2
 $(\text{good-os (os)} \wedge \text{tm-some-input-interruptp (tm-iports (os))})$
 $\rightarrow \text{(os-ready-set (os-waiting-input-handler-path2 (os))}$
 $= \text{index-ready-set ('0, putnth ('(0 0), tm-interrupting-input-port (tm-iports (os)), table ('2, os-status-table (os))))})$

EVENT: Disable os-ready-set-os-waiting-input-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path2
 $(\text{good-os (os)} \wedge (\text{tm-waiting (os)} \wedge (\text{tm-some-input-interruptp (tm-iports (os))} \wedge (\text{getnth ('2 * tm-interrupting-input-port (tm-iports (os)), os-status-table (os)) = '5)))))$
 $\rightarrow \text{permutation (mapup-queue (array-enq (tm-interrupting-input-port (tm-iports (os)), os-readyq (os)), os-ready-set (os-waiting-input-handler-path2 (os))))}$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path2.

THEOREM: not-tm-waiting-os-waiting-input-handler-path2
 $\neg \text{tm-waiting (os-waiting-input-handler-path2 (os))}$

EVENT: Disable not-tm-waiting-os-waiting-input-handler-path2.

THEOREM: tm-base-os-waiting-input-handler-path2
tm-base (os-waiting-input-handler-path2 (os))
= base (getnth (tm-interrupting-input-port (tm-iports (os)),
table ('2, os-segment-table (os))))

EVENT: Disable tm-base-os-waiting-input-handler-path2.

THEOREM: tm-limit-os-waiting-input-handler-path2
tm-limit (os-waiting-input-handler-path2 (os))
= limit (getnth (tm-interrupting-input-port (tm-iports (os)),
table ('2, os-segment-table (os))))

EVENT: Disable tm-limit-os-waiting-input-handler-path2.

THEOREM: tm-base-tm-limit-os-waiting-input-handler-path2
(good-os (os) \wedge tm-waiting (os))
 \rightarrow ((tm-base (os-waiting-input-handler-path2 (os))
= base (getnth (os-current-taskid (os-waiting-input-handler-path2 (os)),
table ('2, os-segment-table (os))))))
 \wedge (tm-limit (os-waiting-input-handler-path2 (os))
= limit (getnth (os-current-taskid (os-waiting-input-handler-path2 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-waiting-input-handler-path2.

THEOREM: good-os-os-waiting-input-handler-path2
(good-os (os)
 \wedge (tm-some-input-interruptp (tm-iports (os))
 \wedge (tm-waiting (os)
 \wedge ((getnth ('2
* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
= '5)
 \wedge ((\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os)))
 \wedge tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
tm-iports (os)))))))))
 \rightarrow good-os (os-waiting-input-handler-path2 (os))

EVENT: Disable good-os-os-waiting-input-handler-path2.

THEOREM: good-tm-os-waiting-input-handler-path3

$$\begin{aligned}
& \text{good-os}(os) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge (\text{tm-waiting}(os) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = '5) \\
& \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os}))) \\
& \quad \quad \wedge (\neg \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{tm-iports}(os))))))))) \\
& \rightarrow \text{good-tm}(\text{os-waiting-input-handler-path3}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-waiting-input-handler-path3.

THEOREM: tm-slimit-os-waiting-input-handler-path3

$$\text{tm-slimit}(\text{os-waiting-input-handler-path3}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-waiting-input-handler-path3.

THEOREM: os-code-os-waiting-input-handler-path3

$$\begin{aligned}
& \text{good-tm}(os) \\
& \rightarrow (\text{os-code}(\text{os-waiting-input-handler-path3}(os)) = \text{os-code}(os))
\end{aligned}$$

EVENT: Disable os-code-os-waiting-input-handler-path3.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path3

$$\begin{aligned}
& \text{good-tm}(os) \\
& \rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-waiting-input-handler-path3}(os))) \\
& \quad = \text{getnth}('3, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-waiting-input-handler-path3}(os))) \\
& \quad \quad = \text{getnth}('4, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('5, \\
& \quad \quad \quad \text{tm-memory}(\text{os-waiting-input-handler-path3}(os))) \\
& \quad \quad \quad = \text{getnth}('5, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('6, \\
& \quad \quad \quad \text{tm-memory}(\text{os-waiting-input-handler-path3}(os))) \\
& \quad \quad \quad = \text{getnth}('6, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('7, \\
& \quad \quad \quad \text{tm-memory}(\text{os-waiting-input-handler-path3}(os))) \\
& \quad \quad \quad = \text{getnth}('7, \text{tm-memory}(os))) \\
& \quad \quad \wedge (\text{getnth}('11,
\end{aligned}$$

$$\begin{aligned}
& \text{tm-memory (os-waiting-input-handler-path3 (os))} \\
= & \text{getnth ('11,} \\
& \text{tm-memory (os))}))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path3
 \neg tm-in-supervisor-mode (os-waiting-input-handler-path3 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path3.

THEOREM: os-task-table-os-waiting-input-handler-path3
good-tm (os)
 \rightarrow (os-task-table (os-waiting-input-handler-path3 (os))
= os-task-table (os))

EVENT: Disable os-task-table-os-waiting-input-handler-path3.

THEOREM: os-segment-table-os-waiting-input-handler-path3
good-tm (os)
 \rightarrow (os-segment-table (os-waiting-input-handler-path3 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-waiting-input-handler-path3.

THEOREM: os-readyq-os-waiting-input-handler-path3
good-tm (os)
 \rightarrow (os-readyq (os-waiting-input-handler-path3 (os))
= array-enq (tm-interrupting-input-port (tm-iports (os)),
os-readyq (os)))

EVENT: Disable os-readyq-os-waiting-input-handler-path3.

THEOREM: os-ibuffers-os-waiting-input-handler-path3
good-tm (os)
 \rightarrow (os-ibuffers (os-waiting-input-handler-path3 (os))
= os-ibuffers-with-enqueued-character (os))

EVENT: Disable os-ibuffers-os-waiting-input-handler-path3.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path3
(good-os (os))

$$\begin{aligned} & \wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ & \quad \wedge (\neg \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \quad \quad os)))))) \\ \rightarrow & \text{finite-number-queue-listp} (\text{table} ('8, \\ & \quad \quad \quad \text{os-ibuffers} (\text{os-waiting-input-handler-path3} (os)), \\ & \quad \quad \quad '4, \\ & \quad \quad \quad '65536)) \end{aligned}$$

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path3.

THEOREM: os-obuffers-os-waiting-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-obuffers} (\text{os-waiting-input-handler-path3} (os)) = \text{os-obuffers} (os))$

EVENT: Disable os-obuffers-os-waiting-input-handler-path3.

THEOREM: os-mbuffers-os-waiting-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-mbuffers} (\text{os-waiting-input-handler-path3} (os)) = \text{os-mbuffers} (os))$

EVENT: Disable os-mbuffers-os-waiting-input-handler-path3.

THEOREM: os-status-table-os-waiting-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-status-table} (\text{os-waiting-input-handler-path3} (os)) \\ = \text{os-update-status} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ \quad \quad \quad '0, \\ \quad \quad \quad '0, \\ \quad \quad \quad os))$

EVENT: Disable os-status-table-os-waiting-input-handler-path3.

THEOREM: good-status-list-os-status-table-os-waiting-input-handler-path3
(good-os (os) \wedge tm-some-input-interruptp (tm-iports (os)))
 $\rightarrow \text{good-status-list} (\text{table} ('2, \\ \quad \quad \quad \text{os-status-table} (\text{os-waiting-input-handler-path3} (os))))$

EVENT: Disable good-status-list-os-status-table-os-waiting-input-handler-path3.

THEOREM: os-ready-set-os-waiting-input-handler-path3
(good-os (os) \wedge tm-some-input-interruptp (tm-iports (os)))

$$\begin{aligned} &\rightarrow (\text{os-ready-set } (\text{os-waiting-input-handler-path3 } (os)) \\ &= \text{index-ready-set } ('0, \\ &\quad \text{putnth } ('(0\ 0), \\ &\quad \quad \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ &\quad \quad \text{table } ('2, \text{os-status-table } (os)))) \end{aligned}$$

EVENT: Disable os-ready-set-os-waiting-input-handler-path3.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path3

$$\begin{aligned} &(\text{good-os } (os) \\ &\wedge (\text{tm-waiting } (os) \\ &\quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\ &\quad \quad \wedge (\text{getnth } ('2 \\ &\quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ &\quad \quad \quad \text{os-status-table } (os)) \\ &\quad \quad = '5)))) \\ &\rightarrow \text{permutation } (\text{mapup-queue } (\text{array-enq } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ &\quad \quad \quad \text{os-readyq } (os))), \\ &\quad \quad \text{os-ready-set } (\text{os-waiting-input-handler-path3 } (os))) \end{aligned}$$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path3.

THEOREM: not-tm-waiting-os-waiting-input-handler-path3

$$\neg \text{tm-waiting } (\text{os-waiting-input-handler-path3 } (os))$$

EVENT: Disable not-tm-waiting-os-waiting-input-handler-path3.

THEOREM: tm-base-os-waiting-input-handler-path3

$$\begin{aligned} &\text{tm-base } (\text{os-waiting-input-handler-path3 } (os)) \\ &= \text{base } (\text{getnth } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ &\quad \quad \text{table } ('2, \text{os-segment-table } (os)))) \end{aligned}$$

EVENT: Disable tm-base-os-waiting-input-handler-path3.

THEOREM: tm-limit-os-waiting-input-handler-path3

$$\begin{aligned} &\text{tm-limit } (\text{os-waiting-input-handler-path3 } (os)) \\ &= \text{limit } (\text{getnth } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ &\quad \quad \text{table } ('2, \text{os-segment-table } (os)))) \end{aligned}$$

EVENT: Disable tm-limit-os-waiting-input-handler-path3.

THEOREM: tm-base-tm-limit-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge \text{tm-waiting}(os)) \\
\rightarrow & ((\text{tm-base}(\text{os-waiting-input-handler-path3}(os)) \\
& = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-waiting-input-handler-path3}(os)), \\
& \quad \text{table}('2, \text{os-segment-table}(os)))))) \\
\wedge & (\text{tm-limit}(\text{os-waiting-input-handler-path3}(os)) \\
& = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-waiting-input-handler-path3}(os)), \\
& \quad \text{table}('2, \text{os-segment-table}(os))))))
\end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-waiting-input-handler-path3.

THEOREM: good-os-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
\wedge & (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge (\text{tm-waiting}(os) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = '5) \\
& \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os})))) \\
& \quad \quad \wedge (\neg \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{tm-iports}(os))))))))) \\
\rightarrow & \text{good-os}(\text{os-waiting-input-handler-path3}(os))
\end{aligned}$$

EVENT: Disable good-os-os-waiting-input-handler-path3.

THEOREM: good-tm-os-waiting-input-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
\wedge & (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge (\text{tm-waiting}(os) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \neq '5) \\
& \quad \quad \wedge \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os})))))) \\
\rightarrow & \text{good-tm}(\text{os-waiting-input-handler-path4}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-waiting-input-handler-path4.

THEOREM: tm-slimit-os-waiting-input-handler-path4

$$\text{tm-slimit}(\text{os-waiting-input-handler-path4}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-waiting-input-handler-path4.

THEOREM: os-code-os-waiting-input-handler-path4

good-tm (*os*)

→ (os-code (os-waiting-input-handler-path4 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-input-handler-path4.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path4

good-tm (*os*)

→ ((getnth ('3, tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('3, tm-memory (*os*)))

∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('4, tm-memory (*os*)))

∧ ((getnth ('5,

tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('5, tm-memory (*os*)))

∧ ((getnth ('6,

tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('6, tm-memory (*os*)))

∧ ((getnth ('7,

tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('7, tm-memory (*os*)))

∧ (getnth ('11,

tm-memory (os-waiting-input-handler-path4 (*os*)))

= getnth ('11,

tm-memory (*os*))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path4

→ tm-in-supervisor-mode (os-waiting-input-handler-path4 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path4.

THEOREM: os-task-table-os-waiting-input-handler-path4

good-tm (*os*)

→ (os-task-table (os-waiting-input-handler-path4 (*os*))

= os-task-table (*os*))

EVENT: Disable os-task-table-os-waiting-input-handler-path4.

THEOREM: os-segment-table-os-waiting-input-handler-path4

good-tm (*os*)

→ (os-segment-table (os-waiting-input-handler-path4 (*os*))

= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-waiting-input-handler-path4.

THEOREM: os-readyq-os-waiting-input-handler-path4
good-tm (*os*)
→ (os-readyq (os-waiting-input-handler-path4 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-waiting-input-handler-path4.

THEOREM: os-ibuffers-os-waiting-input-handler-path4
good-tm (*os*)
→ (os-ibuffers (os-waiting-input-handler-path4 (*os*))
= os-ibuffers-with-qreplaced-overflow-character (*os*))

EVENT: Disable os-ibuffers-os-waiting-input-handler-path4.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path4
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-waiting-input-handler-path4 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path4.

THEOREM: os-obuffers-os-waiting-input-handler-path4
good-tm (*os*)
→ (os-obuffers (os-waiting-input-handler-path4 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-input-handler-path4.

THEOREM: os-mbuffers-os-waiting-input-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-waiting-input-handler-path4 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-input-handler-path4.

THEOREM: os-status-table-os-waiting-input-handler-path4
good-tm (*os*)

→ (os-status-table (os-waiting-input-handler-path4 (os))
= os-status-table (os))

EVENT: Disable os-status-table-os-waiting-input-handler-path4.

THEOREM: tm-waiting-os-waiting-input-handler-path4
tm-waiting (os-waiting-input-handler-path4 (os))

EVENT: Disable tm-waiting-os-waiting-input-handler-path4.

THEOREM: good-os-os-waiting-input-handler-path4
(good-os (os)

∧ (tm-some-input-interruptp (tm-iports (os))

∧ (tm-waiting (os)

∧ ((getnth ('2

* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))

≠ '5)

∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))))

→ good-os (os-waiting-input-handler-path4 (os))

EVENT: Disable good-os-os-waiting-input-handler-path4.

THEOREM: good-tm-os-waiting-input-handler-path5
(good-os (os)

∧ (tm-some-input-interruptp (tm-iports (os))

∧ (tm-waiting (os)

∧ ((getnth ('2

* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))

≠ '5)

∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))

∧ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
tm-iports (os))))))

→ good-tm (os-waiting-input-handler-path5 (os))

EVENT: Disable good-tm-os-waiting-input-handler-path5.

THEOREM: tm-slimit-os-waiting-input-handler-path5
tm-slimit (os-waiting-input-handler-path5 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-waiting-input-handler-path5.

THEOREM: os-code-os-waiting-input-handler-path5

good-tm (*os*)

→ (os-code (os-waiting-input-handler-path5 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-input-handler-path5.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path5

good-tm (*os*)

→ ((getnth ('3, tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('3, tm-memory (*os*)))

∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('4, tm-memory (*os*)))

∧ ((getnth ('5,
tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('5, tm-memory (*os*)))

∧ ((getnth ('6,
tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('6, tm-memory (*os*)))

∧ ((getnth ('7,
tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('7, tm-memory (*os*)))

∧ (getnth ('11,
tm-memory (os-waiting-input-handler-path5 (*os*)))
= getnth ('11,
tm-memory (*os*))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path5.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path5

→ tm-in-supervisor-mode (os-waiting-input-handler-path5 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path5.

THEOREM: os-task-table-os-waiting-input-handler-path5

good-tm (*os*)

→ (os-task-table (os-waiting-input-handler-path5 (*os*))
= os-task-table (*os*))

EVENT: Disable os-task-table-os-waiting-input-handler-path5.

THEOREM: os-segment-table-os-waiting-input-handler-path5

$\text{good-tm}(os)$
 $\rightarrow (\text{os-segment-table}(\text{os-waiting-input-handler-path5}(os)))$
 $= \text{os-segment-table}(os)$

EVENT: Disable os-segment-table-os-waiting-input-handler-path5.

THEOREM: os-readyq-os-waiting-input-handler-path5
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-readyq}(\text{os-waiting-input-handler-path5}(os))) = \text{os-readyq}(os)$

EVENT: Disable os-readyq-os-waiting-input-handler-path5.

THEOREM: os-ibuffers-os-waiting-input-handler-path5
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-ibuffers}(\text{os-waiting-input-handler-path5}(os)))$
 $= \text{os-ibuffers-with-enqueued-overflow-character}(os)$

EVENT: Disable os-ibuffers-os-waiting-input-handler-path5.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path5
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge (\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os))),$
 $os))))))$
 $\rightarrow \text{finite-number-queue-listp}(\text{table}('8,$
 $\text{os-ibuffers}(\text{os-waiting-input-handler-path5}(os)),$
 $'4,$
 $'65536)$

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path5.

THEOREM: os-obuffers-os-waiting-input-handler-path5
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-obuffers}(\text{os-waiting-input-handler-path5}(os))) = \text{os-obuffers}(os)$

EVENT: Disable os-obuffers-os-waiting-input-handler-path5.

THEOREM: os-mbuffers-os-waiting-input-handler-path5
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-mbuffers}(\text{os-waiting-input-handler-path5}(os))) = \text{os-mbuffers}(os)$

EVENT: Disable os-mbuffers-os-waiting-input-handler-path5.

THEOREM: os-status-table-os-waiting-input-handler-path5
good-tm (*os*)
→ (os-status-table (os-waiting-input-handler-path5 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-waiting-input-handler-path5.

THEOREM: tm-waiting-os-waiting-input-handler-path5
tm-waiting (os-waiting-input-handler-path5 (*os*))

EVENT: Disable tm-waiting-os-waiting-input-handler-path5.

THEOREM: good-os-os-waiting-input-handler-path5
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (tm-waiting (*os*)
∧ ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
≠ '5)
∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))
∧ tm-ipt-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*)))))))))
→ good-os (os-waiting-input-handler-path5 (*os*))

EVENT: Disable good-os-os-waiting-input-handler-path5.

THEOREM: good-tm-os-waiting-input-handler-path6
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (tm-waiting (*os*)
∧ ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
≠ '5)
∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))
∧ (¬ tm-ipt-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*)))))))))
→ good-tm (os-waiting-input-handler-path6 (*os*))

EVENT: Disable good-tm-os-waiting-input-handler-path6.

THEOREM: tm-slimit-os-waiting-input-handler-path6
 $\text{tm-slimit}(\text{os-waiting-input-handler-path6}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-waiting-input-handler-path6.

THEOREM: os-code-os-waiting-input-handler-path6
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-code}(\text{os-waiting-input-handler-path6}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-waiting-input-handler-path6.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path6
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5,$
 $\text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\text{tm-memory}(\text{os-waiting-input-handler-path6}(os)))$
 $= \text{getnth}('11,$
 $\text{tm-memory}(os)))))))))$

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path6.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path6
 $\neg \text{tm-in-supervisor-mode}(\text{os-waiting-input-handler-path6}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-input-handler-path6.

THEOREM: os-task-table-os-waiting-input-handler-path6
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-waiting-input-handler-path6}(os))$
 $= \text{os-task-table}(os))$

EVENT: Disable os-task-table-os-waiting-input-handler-path6.

THEOREM: os-segment-table-os-waiting-input-handler-path6
good-tm (*os*)
→ (os-segment-table (os-waiting-input-handler-path6 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-waiting-input-handler-path6.

THEOREM: os-readyq-os-waiting-input-handler-path6
good-tm (*os*)
→ (os-readyq (os-waiting-input-handler-path6 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-waiting-input-handler-path6.

THEOREM: os-ibuffers-os-waiting-input-handler-path6
good-tm (*os*)
→ (os-ibuffers (os-waiting-input-handler-path6 (*os*))
= os-ibuffers-with-enqueued-character (*os*))

EVENT: Disable os-ibuffers-os-waiting-input-handler-path6.

THEOREM: finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path6
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-waiting-input-handler-path6 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path6.

THEOREM: os-obuffers-os-waiting-input-handler-path6
good-tm (*os*)
→ (os-obuffers (os-waiting-input-handler-path6 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-input-handler-path6.

THEOREM: os-mbuffers-os-waiting-input-handler-path6

good-tm (*os*)
 \rightarrow (os-mbuffers (os-waiting-input-handler-path6 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-input-handler-path6.

THEOREM: os-status-table-os-waiting-input-handler-path6
good-tm (*os*)
 \rightarrow (os-status-table (os-waiting-input-handler-path6 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-waiting-input-handler-path6.

THEOREM: tm-waiting-os-waiting-input-handler-path6
tm-waiting (os-waiting-input-handler-path6 (*os*))

EVENT: Disable tm-waiting-os-waiting-input-handler-path6.

THEOREM: good-os-os-waiting-input-handler-path6
(good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge (tm-waiting (*os*)
 \wedge ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*)
 \neq '5)
 \wedge ((\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))
 \wedge (\neg tm-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))))))))))
 \rightarrow good-os (os-waiting-input-handler-path6 (*os*))

EVENT: Disable good-os-os-waiting-input-handler-path6.

THEOREM: good-os-os-waiting-input-handler
(good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)) \wedge tm-waiting (*os*)))
 \rightarrow good-os (os-waiting-input-handler (*os*))

EVENT: Disable good-os-os-waiting-input-handler.

THEOREM: finite-number-queuep-array-enq-interrupting-input-port2
(good-os (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge (\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = '5)))) \\
\rightarrow & \text{finite-number-queuep}(\text{array-enq}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-readyq}(os)), \\
& \quad \quad '16, \\
& \quad '16)
\end{aligned}$$

EVENT: Disable finite-number-queuep-array-enq-interrupting-input-port2.

THEOREM: good-tm-os-running-input-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad = '5) \\
& \quad \quad \quad \wedge \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \quad \text{os))))))))) \\
\rightarrow & \text{good-tm}(\text{os-running-input-handler-path1}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-running-input-handler-path1.

THEOREM: tm-slimit-os-running-input-handler-path1

$$\text{tm-slimit}(\text{os-running-input-handler-path1}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-running-input-handler-path1.

THEOREM: os-code-os-running-input-handler-path1

$$\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-running-input-handler-path1}(os)) = \text{os-code}(os))$$

EVENT: Disable os-code-os-running-input-handler-path1.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path1

$$\begin{aligned}
& \text{good-tm}(os) \\
\rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
& \quad = \text{getnth}('3, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-running-input-handler-path1}(os)))
\end{aligned}$$

$$\begin{aligned}
&= \text{getnth}('4, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('5, \\
&\quad \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
&= \text{getnth}('5, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('6, \\
&\quad \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
&= \text{getnth}('6, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('7, \\
&\quad \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
&= \text{getnth}('7, \text{tm-memory}(os)) \\
\wedge &(\text{getnth}('11, \\
&\quad \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
&= \text{getnth}('11, \\
&\quad \text{tm-memory}(os)))))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path1
 $\neg \text{tm-in-supervisor-mode}(\text{os-running-input-handler-path1}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path1.

THEOREM: os-task-table-os-running-input-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-task-table}(\text{os-running-input-handler-path1}(os))$
 $= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-running-input-handler-path1.

THEOREM: os-segment-table-os-running-input-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-segment-table}(\text{os-running-input-handler-path1}(os))$
 $= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-running-input-handler-path1.

THEOREM: os-readyq-os-running-input-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-readyq}(\text{os-running-input-handler-path1}(os))$
 $= \text{array-enq}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-readyq}(os))$

EVENT: Disable os-readyq-os-running-input-handler-path1.

THEOREM: os-ibuffers-os-running-input-handler-path1
good-tm (*os*)
→ (os-ibuffers (os-running-input-handler-path1 (*os*))
= os-ibuffers-with-qreplaced-overflow-character (*os*))

EVENT: Disable os-ibuffers-os-running-input-handler-path1.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path1
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-running-input-handler-path1 (*os*)),
'4,
'65536))

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path1.

THEOREM: os-obuffers-os-running-input-handler-path1
good-tm (*os*)
→ (os-obuffers (os-running-input-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path1.

THEOREM: os-mbuffers-os-running-input-handler-path1
good-tm (*os*)
→ (os-mbuffers (os-running-input-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path1.

THEOREM: os-status-table-os-running-input-handler-path1
good-tm (*os*)
→ (os-status-table (os-running-input-handler-path1 (*os*))
= os-update-status (tm-interrupting-input-port (tm-iports (*os*)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-running-input-handler-path1.

THEOREM: good-status-list-os-status-table-os-running-input-handler-path1

$(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow \text{good-status-list}(\text{table}('2,$
 $\qquad\qquad\qquad \text{os-status-table}(\text{os-running-input-handler-path1}(os))))$

EVENT: Disable good-status-list-os-status-table-os-running-input-handler-path1.

THEOREM: os-ready-set-os-running-input-handler-path1
 $(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow (\text{os-ready-set}(\text{os-running-input-handler-path1}(os))$
 $\quad = \text{index-ready-set}('0,$
 $\qquad\qquad\qquad \text{putnth}('0\ 0),$
 $\qquad\qquad\qquad \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\qquad\qquad\qquad \text{table}('2, \text{os-status-table}(os))))$

EVENT: Disable os-ready-set-os-running-input-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-input-handler-path1
 $(\text{good-os}(os)$
 $\quad \wedge ((\neg \text{tm-waiting}(os))$
 $\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\quad \quad \quad \wedge (\text{getnth}('2$
 $\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \quad \quad \quad \text{os-status-table}(os))$
 $\quad \quad \quad = '5))))$
 $\rightarrow \text{permutation}(\text{mapup-queue}(\text{array-enq}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\qquad\qquad\qquad \text{os-readyq}(os))),$
 $\qquad\qquad\qquad \text{os-ready-set}(\text{os-running-input-handler-path1}(os)))$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-input-handler-path1.

THEOREM: not-tm-waiting-os-running-input-handler-path1
 $\neg \text{tm-waiting}(\text{os-running-input-handler-path1}(os))$

EVENT: Disable not-tm-waiting-os-running-input-handler-path1.

THEOREM: tm-base-os-running-input-handler-path1
 $\text{tm-base}(\text{os-running-input-handler-path1}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-running-input-handler-path1.

THEOREM: tm-limit-os-running-input-handler-path1
 $\text{tm-limit}(\text{os-running-input-handler-path1}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-input-handler-path1.

THEOREM: tm-base-tm-limit-os-running-input-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge (\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '5)))) \\
\rightarrow & ((\text{tm-base } (\text{os-running-input-handler-path1 } (os)) \\
& \quad = \text{base } (\text{getnth } (\text{os-current-taskid } (\text{os-running-input-handler-path1 } (os)), \\
& \quad \quad \text{table } ('2, \text{os-segment-table } (os)))))) \\
& \wedge (\text{tm-limit } (\text{os-running-input-handler-path1 } (os)) \\
& \quad = \text{limit } (\text{getnth } (\text{os-current-taskid } (\text{os-running-input-handler-path1 } (os)), \\
& \quad \quad \text{table } ('2, \text{os-segment-table } (os))))))
\end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path1.

THEOREM: good-os-os-running-input-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '5) \\
& \quad \quad \quad \wedge \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \quad \text{os})))))) \\
\rightarrow & \text{good-os } (\text{os-running-input-handler-path1 } (os))
\end{aligned}$$

EVENT: Disable good-os-os-running-input-handler-path1.

THEOREM: good-tm-os-running-input-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '5) \\
& \quad \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \quad \text{os}))))))
\end{aligned}$$

$$\begin{aligned} & \wedge \text{tm-iptort-errorp}(\text{tm-interrupting-input-port}(\text{tm-iptorts}(os)), \\ & \quad \text{tm-iptorts}(os)))))) \\ \rightarrow & \text{good-tm}(\text{os-running-input-handler-path2}(os)) \end{aligned}$$

EVENT: Disable good-tm-os-running-input-handler-path2.

THEOREM: tm-slimit-os-running-input-handler-path2
 $\text{tm-slimit}(\text{os-running-input-handler-path2}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-running-input-handler-path2.

THEOREM: os-code-os-running-input-handler-path2
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-code}(\text{os-running-input-handler-path2}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-running-input-handler-path2.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path2
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5,$
 $\quad \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\quad \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\quad \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\quad \text{tm-memory}(\text{os-running-input-handler-path2}(os)))$
 $= \text{getnth}('11,$
 $\quad \text{tm-memory}(os))))))$

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path2
 $\neg \text{tm-in-supervisor-mode}(\text{os-running-input-handler-path2}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path2.

THEOREM: os-task-table-os-running-input-handler-path2
good-tm (*os*)
→ (os-task-table (os-running-input-handler-path2 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-running-input-handler-path2.

THEOREM: os-segment-table-os-running-input-handler-path2
good-tm (*os*)
→ (os-segment-table (os-running-input-handler-path2 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-running-input-handler-path2.

THEOREM: os-readyq-os-running-input-handler-path2
good-tm (*os*)
→ (os-readyq (os-running-input-handler-path2 (*os*))
= array-enq (tm-interrupting-input-port (tm-iports (*os*)),
os-readyq (*os*)))

EVENT: Disable os-readyq-os-running-input-handler-path2.

THEOREM: os-ibuffers-os-running-input-handler-path2
good-tm (*os*)
→ (os-ibuffers (os-running-input-handler-path2 (*os*))
= os-ibuffers-with-enqueued-overflow-character (*os*))

EVENT: Disable os-ibuffers-os-running-input-handler-path2.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path2
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-running-input-handler-path2 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path2.

THEOREM: os-obuffers-os-running-input-handler-path2

good-tm (*os*)
 \rightarrow (os-obuffers (os-running-input-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path2.

THEOREM: os-mbuffers-os-running-input-handler-path2
good-tm (*os*)
 \rightarrow (os-mbuffers (os-running-input-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path2.

THEOREM: os-status-table-os-running-input-handler-path2
good-tm (*os*)
 \rightarrow (os-status-table (os-running-input-handler-path2 (*os*))
= os-update-status (tm-interrupting-input-port (tm-iports (*os*)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-running-input-handler-path2.

THEOREM: good-status-list-os-status-table-os-running-input-handler-path2
(good-os (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*)))
 \rightarrow good-status-list (table ('2,
os-status-table (os-running-input-handler-path2 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-running-input-handler-path2.

THEOREM: os-ready-set-os-running-input-handler-path2
(good-os (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*)))
 \rightarrow (os-ready-set (os-running-input-handler-path2 (*os*))
= index-ready-set ('0,
putnth ('(0 0),
tm-interrupting-input-port (tm-iports (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable os-ready-set-os-running-input-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-input-handler-path2
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*)))

$$\begin{aligned}
& \wedge \text{ (getnth ('2} \\
& \quad * \text{ tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \text{os-status-table (os))} \\
& \quad = \text{ '5))}) \\
\rightarrow & \text{ permutation (mapup-queue (array-enq (tm-interrupting-input-port (tm-iports (os)),} \\
& \quad \text{os-readyq (os))),} \\
& \quad \text{os-ready-set (os-running-input-handler-path2 (os)))}
\end{aligned}$$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-input-handler-path2.

THEOREM: not-tm-waiting-os-running-input-handler-path2
 $\neg \text{ tm-waiting (os-running-input-handler-path2 (os))}$

EVENT: Disable not-tm-waiting-os-running-input-handler-path2.

THEOREM: tm-base-os-running-input-handler-path2
 $\text{tm-base (os-running-input-handler-path2 (os)) = tm-base (os)}$

EVENT: Disable tm-base-os-running-input-handler-path2.

THEOREM: tm-limit-os-running-input-handler-path2
 $\text{tm-limit (os-running-input-handler-path2 (os)) = tm-limit (os)}$

EVENT: Disable tm-limit-os-running-input-handler-path2.

THEOREM: tm-base-tm-limit-os-running-input-handler-path2
 (good-os (os))
 $\wedge \text{ (tm-some-input-interruptp (tm-iports (os))}$
 $\quad \wedge \text{ ((}\neg \text{ tm-waiting (os))}$
 $\quad \quad \wedge \text{ (getnth ('2}$
 $\quad \quad \quad * \text{ tm-interrupting-input-port (tm-iports (os)),}$
 $\quad \quad \quad \text{os-status-table (os))}$
 $\quad \quad = \text{ '5))})$
 $\rightarrow \text{ ((tm-base (os-running-input-handler-path2 (os))}$
 $\quad = \text{ base (getnth (os-current-taskid (os-running-input-handler-path2 (os)),}$
 $\quad \quad \text{table ('2, os-segment-table (os))))}$
 $\quad \wedge \text{ (tm-limit (os-running-input-handler-path2 (os))}$
 $\quad = \text{ limit (getnth (os-current-taskid (os-running-input-handler-path2 (os)),}$
 $\quad \quad \text{table ('2, os-segment-table (os))))})$

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path2.

THEOREM: good-os-os-running-input-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad = '5) \\ & \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge \text{tm-iport-errorp } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os)))))) \\ & \rightarrow \text{good-os } (\text{os-running-input-handler-path2 } (os)) \end{aligned}$$

EVENT: Disable good-os-os-running-input-handler-path2.

THEOREM: good-tm-os-running-input-handler-path3

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad = '5) \\ & \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge (\neg \text{tm-iport-errorp } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os)))))) \\ & \rightarrow \text{good-tm } (\text{os-running-input-handler-path3 } (os)) \end{aligned}$$

EVENT: Disable good-tm-os-running-input-handler-path3.

THEOREM: tm-slimit-os-running-input-handler-path3

$$\text{tm-slimit } (\text{os-running-input-handler-path3 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-running-input-handler-path3.

THEOREM: os-code-os-running-input-handler-path3

$$\begin{aligned} & \text{good-tm } (os) \\ & \rightarrow (\text{os-code } (\text{os-running-input-handler-path3 } (os)) = \text{os-code } (os)) \end{aligned}$$

EVENT: Disable os-code-os-running-input-handler-path3.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path3

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \\ & \quad \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-running-input-handler-path3}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path3

$$\neg \text{tm-in-supervisor-mode}(\text{os-running-input-handler-path3}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path3.

THEOREM: os-task-table-os-running-input-handler-path3

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-running-input-handler-path3}(os)) \\ & = \text{os-new-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-running-input-handler-path3.

THEOREM: os-segment-table-os-running-input-handler-path3

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-running-input-handler-path3}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-running-input-handler-path3.

THEOREM: os-readyq-os-running-input-handler-path3

$$\text{good-tm}(os)$$

$$\begin{aligned} &\rightarrow (\text{os-readyq} (\text{os-running-input-handler-path3} (os)) \\ &\quad = \text{array-enq} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ &\quad \quad \text{os-readyq} (os))) \end{aligned}$$

EVENT: Disable os-readyq-os-running-input-handler-path3.

THEOREM: os-ibuffers-os-running-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-ibuffers} (\text{os-running-input-handler-path3} (os)) \\ = \text{os-ibuffers-with-enqueued-character} (os))$

EVENT: Disable os-ibuffers-os-running-input-handler-path3.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path3
(good-os (os)
 $\wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os)) \\ \wedge (\neg \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ os))))))$
 $\rightarrow \text{finite-number-queue-listp} (\text{table} ('8, \\ \quad \quad \quad \text{os-ibuffers} (\text{os-running-input-handler-path3} (os)), \\ '4, \\ '65536))$

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path3.

THEOREM: os-obuffers-os-running-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-obuffers} (\text{os-running-input-handler-path3} (os)) = \text{os-obuffers} (os))$

EVENT: Disable os-obuffers-os-running-input-handler-path3.

THEOREM: os-mbuffers-os-running-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-mbuffers} (\text{os-running-input-handler-path3} (os)) = \text{os-mbuffers} (os))$

EVENT: Disable os-mbuffers-os-running-input-handler-path3.

THEOREM: os-status-table-os-running-input-handler-path3
good-tm (os)
 $\rightarrow (\text{os-status-table} (\text{os-running-input-handler-path3} (os)) \\ = \text{os-update-status} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)),$

'0,
'0,
os)

EVENT: Disable os-status-table-os-running-input-handler-path3.

THEOREM: good-status-list-os-status-table-os-running-input-handler-path3
 (good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
 → good-status-list (table ('2,
 os-status-table (os-running-input-handler-path3 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-running-input-handler-path3.

THEOREM: os-ready-set-os-running-input-handler-path3
 (good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
 → (os-ready-set (os-running-input-handler-path3 (*os*))
 = index-ready-set ('0,
 putnth ('(0 0),
 tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-status-table (*os*))))))

EVENT: Disable os-ready-set-os-running-input-handler-path3.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-input-handler-path3
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*)
 ∧ (getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*)
 = '5))))))
 → permutation (mapup-queue (array-enq (tm-interrupting-input-port (tm-iports (*os*)),
 os-readyq (*os*))),
 os-ready-set (os-running-input-handler-path3 (*os*))))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-input-handler-path3.

THEOREM: not-tm-waiting-os-running-input-handler-path3
 ¬ tm-waiting (os-running-input-handler-path3 (*os*))

EVENT: Disable not-tm-waiting-os-running-input-handler-path3.

THEOREM: tm-base-os-running-input-handler-path3
 $\text{tm-base}(\text{os-running-input-handler-path3}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-running-input-handler-path3.

THEOREM: tm-limit-os-running-input-handler-path3
 $\text{tm-limit}(\text{os-running-input-handler-path3}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-input-handler-path3.

THEOREM: tm-base-tm-limit-os-running-input-handler-path3
 $(\text{good-os}(os) \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\text{getnth}('2 * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{os-status-table}(os)) = '5))))))$
 $\rightarrow ((\text{tm-base}(\text{os-running-input-handler-path3}(os)) = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path3}(os)), \text{table}('2, \text{os-segment-table}(os)))))) \wedge (\text{tm-limit}(\text{os-running-input-handler-path3}(os)) = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path3}(os)), \text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path3.

THEOREM: good-os-os-running-input-handler-path3
 $(\text{good-os}(os) \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{getnth}('2 * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{os-status-table}(os)) = '5) \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), os))) \wedge (\neg \text{tm-ipt-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{tm-iports}(os))))))))))$
 $\rightarrow \text{good-os}(\text{os-running-input-handler-path3}(os))$

EVENT: Disable good-os-os-running-input-handler-path3.

THEOREM: good-tm-os-running-input-handler-path4

(good-os (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ ((¬ tm-waiting (*os*))
 ∧ ((getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 ≠ '5)
 ∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
 os))))))
 → good-tm (os-running-input-handler-path4 (*os*))

EVENT: Disable good-tm-os-running-input-handler-path4.

THEOREM: tm-slimit-os-running-input-handler-path4

tm-slimit (os-running-input-handler-path4 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-running-input-handler-path4.

THEOREM: os-code-os-running-input-handler-path4

good-tm (*os*)
 → (os-code (os-running-input-handler-path4 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-input-handler-path4.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path4

good-tm (*os*)
 → ((getnth ('3, tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('4, tm-memory (*os*)))
 ∧ ((getnth ('5,
 tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('5, tm-memory (*os*)))
 ∧ ((getnth ('6,
 tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('6, tm-memory (*os*)))
 ∧ ((getnth ('7,
 tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('7, tm-memory (*os*)))
 ∧ (getnth ('11,
 tm-memory (os-running-input-handler-path4 (*os*)))
 = getnth ('11,
 tm-memory (*os*))))))

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path4
→ tm-in-supervisor-mode (os-running-input-handler-path4 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path4.

THEOREM: os-task-table-os-running-input-handler-path4
good-tm (os)
→ (os-task-table (os-running-input-handler-path4 (os))
= os-new-task-table (os))

EVENT: Disable os-task-table-os-running-input-handler-path4.

THEOREM: os-segment-table-os-running-input-handler-path4
good-tm (os)
→ (os-segment-table (os-running-input-handler-path4 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-running-input-handler-path4.

THEOREM: os-readyq-os-running-input-handler-path4
good-tm (os)
→ (os-readyq (os-running-input-handler-path4 (os)) = os-readyq (os))

EVENT: Disable os-readyq-os-running-input-handler-path4.

THEOREM: os-ibuffers-os-running-input-handler-path4
good-tm (os)
→ (os-ibuffers (os-running-input-handler-path4 (os))
= os-ibuffers-with-qreplaced-overflow-character (os))

EVENT: Disable os-ibuffers-os-running-input-handler-path4.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path4
(good-os (os)
∧ (tm-some-input-interruptp (tm-iports (os))
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-running-input-handler-path4 (os))),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path4.

THEOREM: os-obuffers-os-running-input-handler-path4
good-tm (*os*)
→ (os-obuffers (os-running-input-handler-path4 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path4.

THEOREM: os-mbuffers-os-running-input-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-running-input-handler-path4 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path4.

THEOREM: os-status-table-os-running-input-handler-path4
good-tm (*os*)
→ (os-status-table (os-running-input-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-running-input-handler-path4.

THEOREM: not-tm-waiting-os-running-input-handler-path4
¬ tm-waiting (os-running-input-handler-path4 (*os*))

EVENT: Disable not-tm-waiting-os-running-input-handler-path4.

THEOREM: tm-base-os-running-input-handler-path4
tm-base (os-running-input-handler-path4 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-input-handler-path4.

THEOREM: tm-limit-os-running-input-handler-path4
tm-limit (os-running-input-handler-path4 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-input-handler-path4.

THEOREM: tm-base-tm-limit-os-running-input-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-running-input-handler-path4 (*os*))
= base (getnth (os-current-taskid (os-running-input-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable os-code-os-running-input-handler-path5.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path5
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-running-input-handler-path5 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))))

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path5.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path5
¬ tm-in-supervisor-mode (os-running-input-handler-path5 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path5.

THEOREM: os-task-table-os-running-input-handler-path5
good-tm (*os*)
→ (os-task-table (os-running-input-handler-path5 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-running-input-handler-path5.

THEOREM: os-segment-table-os-running-input-handler-path5
good-tm (*os*)
→ (os-segment-table (os-running-input-handler-path5 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-running-input-handler-path5.

THEOREM: os-readyq-os-running-input-handler-path5
good-tm (*os*)
→ (os-readyq (os-running-input-handler-path5 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-running-input-handler-path5.

THEOREM: os-ibuffers-os-running-input-handler-path5
good-tm (*os*)
→ (os-ibuffers (os-running-input-handler-path5 (*os*))
= os-ibuffers-with-enqueued-overflow-character (*os*))

EVENT: Disable os-ibuffers-os-running-input-handler-path5.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path5
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-ibuffers (os-running-input-handler-path5 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path5.

THEOREM: os-obuffers-os-running-input-handler-path5
good-tm (*os*)
→ (os-obuffers (os-running-input-handler-path5 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path5.

THEOREM: os-mbuffers-os-running-input-handler-path5
good-tm (*os*)
→ (os-mbuffers (os-running-input-handler-path5 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path5.

THEOREM: os-status-table-os-running-input-handler-path5
good-tm (*os*)
→ (os-status-table (os-running-input-handler-path5 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-running-input-handler-path5.

THEOREM: not-tm-waiting-os-running-input-handler-path5
 $\neg \text{tm-waiting}(\text{os-running-input-handler-path5}(os))$

EVENT: Disable not-tm-waiting-os-running-input-handler-path5.

THEOREM: tm-base-os-running-input-handler-path5
 $\text{tm-base}(\text{os-running-input-handler-path5}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-running-input-handler-path5.

THEOREM: tm-limit-os-running-input-handler-path5
 $\text{tm-limit}(\text{os-running-input-handler-path5}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-input-handler-path5.

THEOREM: tm-base-tm-limit-os-running-input-handler-path5
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow ((\text{tm-base}(\text{os-running-input-handler-path5}(os))$
 $= \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path5}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$
 $\wedge (\text{tm-limit}(\text{os-running-input-handler-path5}(os))$
 $= \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path5}(os)),$
 $\text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path5.

THEOREM: good-os-os-running-input-handler-path5
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\neq '5)$
 $\wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad os))))$
 $\wedge \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{tm-iports}(os))))))$
 $\rightarrow \text{good-os}(\text{os-running-input-handler-path5}(os))$

EVENT: Disable good-os-os-running-input-handler-path5.

THEOREM: good-tm-os-running-input-handler-path6

$$\begin{aligned}
& \text{good-os}(os) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \neq '5) \\
& \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad os))) \\
& \quad \quad \wedge (\neg \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \quad \quad \text{tm-iports}(os))))))))) \\
& \rightarrow \text{good-tm}(\text{os-running-input-handler-path6}(os))
\end{aligned}$$

EVENT: Disable good-tm-os-running-input-handler-path6.

THEOREM: tm-slimit-os-running-input-handler-path6

$$\text{tm-slimit}(\text{os-running-input-handler-path6}(os)) = \text{tm-slimit}(os)$$

EVENT: Disable tm-slimit-os-running-input-handler-path6.

THEOREM: os-code-os-running-input-handler-path6

$$\begin{aligned}
& \text{good-tm}(os) \\
& \rightarrow (\text{os-code}(\text{os-running-input-handler-path6}(os)) = \text{os-code}(os))
\end{aligned}$$

EVENT: Disable os-code-os-running-input-handler-path6.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path6

$$\begin{aligned}
& \text{good-tm}(os) \\
& \rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-running-input-handler-path6}(os))) \\
& \quad = \text{getnth}('3, \text{tm-memory}(os))) \\
& \quad \wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-running-input-handler-path6}(os))) \\
& \quad \quad = \text{getnth}('4, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('5, \\
& \quad \quad \quad \text{tm-memory}(\text{os-running-input-handler-path6}(os))) \\
& \quad \quad \quad = \text{getnth}('5, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('6, \\
& \quad \quad \quad \text{tm-memory}(\text{os-running-input-handler-path6}(os))) \\
& \quad \quad \quad = \text{getnth}('6, \text{tm-memory}(os))) \\
& \quad \quad \wedge ((\text{getnth}('7, \\
& \quad \quad \quad \text{tm-memory}(\text{os-running-input-handler-path6}(os))) \\
& \quad \quad \quad = \text{getnth}('7, \text{tm-memory}(os))) \\
& \quad \quad \quad \wedge (\text{getnth}('11,
\end{aligned}$$

$$\begin{aligned}
& \text{tm-memory (os-running-input-handler-path6 (os))} \\
= & \text{getnth ('11,} \\
& \text{tm-memory (os))}))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path6.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path6
 \neg tm-in-supervisor-mode (os-running-input-handler-path6 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-running-input-handler-path6.

THEOREM: os-task-table-os-running-input-handler-path6
good-tm (os)
 \rightarrow (os-task-table (os-running-input-handler-path6 (os))
= os-new-task-table (os))

EVENT: Disable os-task-table-os-running-input-handler-path6.

THEOREM: os-segment-table-os-running-input-handler-path6
good-tm (os)
 \rightarrow (os-segment-table (os-running-input-handler-path6 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-running-input-handler-path6.

THEOREM: os-readyq-os-running-input-handler-path6
good-tm (os)
 \rightarrow (os-readyq (os-running-input-handler-path6 (os)) = os-readyq (os))

EVENT: Disable os-readyq-os-running-input-handler-path6.

THEOREM: os-ibuffers-os-running-input-handler-path6
good-tm (os)
 \rightarrow (os-ibuffers (os-running-input-handler-path6 (os))
= os-ibuffers-with-enqueued-character (os))

EVENT: Disable os-ibuffers-os-running-input-handler-path6.

THEOREM: finite-number-queue-listp-os-ibuffers-os-running-input-handler-path6
(good-os (os)
 \wedge (tm-some-input-interruptp (tm-iports (os))
 \wedge (\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os))),

$$\rightarrow \text{finite-number-queue-listp}(\text{table}('8, \text{os}))))))$$

$$\text{os-ibuffers}(\text{os-running-input-handler-path6}(\text{os})),$$

$$'4,$$

$$'65536)$$

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-running-input-handler-path6.

THEOREM: os-obuffers-os-running-input-handler-path6
 good-tm (*os*)

$$\rightarrow (\text{os-obuffers}(\text{os-running-input-handler-path6}(\text{os})) = \text{os-obuffers}(\text{os}))$$

EVENT: Disable os-obuffers-os-running-input-handler-path6.

THEOREM: os-mbuffers-os-running-input-handler-path6
 good-tm (*os*)

$$\rightarrow (\text{os-mbuffers}(\text{os-running-input-handler-path6}(\text{os})) = \text{os-mbuffers}(\text{os}))$$

EVENT: Disable os-mbuffers-os-running-input-handler-path6.

THEOREM: os-status-table-os-running-input-handler-path6
 good-tm (*os*)

$$\rightarrow (\text{os-status-table}(\text{os-running-input-handler-path6}(\text{os})) = \text{os-status-table}(\text{os}))$$

EVENT: Disable os-status-table-os-running-input-handler-path6.

THEOREM: not-tm-waiting-os-running-input-handler-path6

$$\neg \text{tm-waiting}(\text{os-running-input-handler-path6}(\text{os}))$$

EVENT: Disable not-tm-waiting-os-running-input-handler-path6.

THEOREM: tm-base-os-running-input-handler-path6

$$\text{tm-base}(\text{os-running-input-handler-path6}(\text{os})) = \text{tm-base}(\text{os})$$

EVENT: Disable tm-base-os-running-input-handler-path6.

THEOREM: tm-limit-os-running-input-handler-path6

$$\text{tm-limit}(\text{os-running-input-handler-path6}(\text{os})) = \text{tm-limit}(\text{os})$$

EVENT: Disable tm-limit-os-running-input-handler-path6.

THEOREM: tm-base-tm-limit-os-running-input-handler-path6
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow ((tm-base (os-running-input-handler-path6 (*os*))
 = base (getnth (os-current-taskid (os-running-input-handler-path6 (*os*)),
 table ('2, os-segment-table (*os*))))))
 \wedge (tm-limit (os-running-input-handler-path6 (*os*))
 = limit (getnth (os-current-taskid (os-running-input-handler-path6 (*os*)),
 table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path6.

THEOREM: good-os-os-running-input-handler-path6
 (good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*)
 \neq '5)
 \wedge ((\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
 os))))
 \wedge (\neg tm-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),
 tm-iports (*os*))))))))))
 \rightarrow good-os (os-running-input-handler-path6 (*os*))

EVENT: Disable good-os-os-running-input-handler-path6.

THEOREM: good-os-os-running-input-handler
 (good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)) \wedge (\neg tm-waiting (*os*))))
 \rightarrow good-os (os-running-input-handler (*os*))

EVENT: Disable good-os-os-running-input-handler.

THEOREM: good-tm-os-intended-output-interrupt
 (good-os (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*)))
 \rightarrow good-tm (os-intended-output-interrupt (*os*))

EVENT: Disable good-tm-os-intended-output-interrupt.

THEOREM: finite-number-queuep-array-enq-interrupting-output-port
 (good-os (*os*)
 \wedge (tm-waiting (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*))))

→ finite-number-queuep (array-enq (tm-interrupting-output-port (tm-oports (*os*)),
os-readyq (*os*)),
'16,
'16)

EVENT: Disable finite-number-queuep-array-enq-interrupting-output-port.

THEOREM: good-tm-os-waiting-output-handler-path1
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (tm-waiting (*os*)
∧ ((getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*)
= '4)
∧ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))))))
→ good-tm (os-waiting-output-handler-path1 (*os*))

EVENT: Disable good-tm-os-waiting-output-handler-path1.

THEOREM: tm-slimit-os-waiting-output-handler-path1
tm-slimit (os-waiting-output-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-output-handler-path1.

THEOREM: os-code-os-waiting-output-handler-path1
good-tm (*os*)
→ (os-code (os-waiting-output-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-output-handler-path1.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path1
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-output-handler-path1 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-output-handler-path1 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-output-handler-path1 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-output-handler-path1 (*os*)))

$$\begin{aligned}
&= \text{getnth}('6, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('7, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path1}(os))) \\
&= \text{getnth}('7, \text{tm-memory}(os)) \\
\wedge &(\text{getnth}('11, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path1}(os))) \\
&= \text{getnth}('11, \\
&\quad \text{tm-memory}(os))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path1
 $\neg \text{tm-in-supervisor-mode}(\text{os-waiting-output-handler-path1}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-output-handler-path1.

THEOREM: os-task-table-os-waiting-output-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-task-table}(\text{os-waiting-output-handler-path1}(os)) = \text{os-task-table}(os))$

EVENT: Disable os-task-table-os-waiting-output-handler-path1.

THEOREM: os-segment-table-os-waiting-output-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-segment-table}(\text{os-waiting-output-handler-path1}(os)) = \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-waiting-output-handler-path1.

THEOREM: os-readyq-os-waiting-output-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-readyq}(\text{os-waiting-output-handler-path1}(os)) = \text{array-enq}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \text{os-readyq}(os)))$

EVENT: Disable os-readyq-os-waiting-output-handler-path1.

THEOREM: os-ibuffers-os-waiting-output-handler-path1
good-tm(*os*)
 $\rightarrow (\text{os-ibuffers}(\text{os-waiting-output-handler-path1}(os)) = \text{os-ibuffers}(os))$

EVENT: Disable os-ibuffers-os-waiting-output-handler-path1.

THEOREM: os-obuffers-os-waiting-output-handler-path1
 good-tm (*os*)
 \rightarrow (os-obuffers (os-waiting-output-handler-path1 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-output-handler-path1.

THEOREM: os-mbuffers-os-waiting-output-handler-path1
 good-tm (*os*)
 \rightarrow (os-mbuffers (os-waiting-output-handler-path1 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-output-handler-path1.

THEOREM: os-status-table-os-waiting-output-handler-path1
 good-tm (*os*)
 \rightarrow (os-status-table (os-waiting-output-handler-path1 (*os*))
 = os-update-status (tm-interrupting-output-port (tm-oports (*os*)),
 '0,
 '0,
 os))

EVENT: Disable os-status-table-os-waiting-output-handler-path1.

THEOREM: good-status-list-os-status-table-os-waiting-output-handler-path1
 (good-os (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*)))
 \rightarrow good-status-list (table ('2,
 os-status-table (os-waiting-output-handler-path1 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-waiting-output-handler-path1.

THEOREM: os-ready-set-os-waiting-output-handler-path1
 (good-os (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*)))
 \rightarrow (os-ready-set (os-waiting-output-handler-path1 (*os*))
 = index-ready-set ('0,
 putnth ('(0 0),
 tm-interrupting-output-port (tm-oports (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable os-ready-set-os-waiting-output-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path1
 (good-os (*os*)
 \wedge (tm-waiting (*os*)))

$$\begin{aligned}
& \wedge (\text{tm-some-output-interruptp} (\text{tm-oports} (os)) \\
& \quad \wedge (\text{getnth} ('2 \\
& \quad \quad * \text{tm-interrupting-output-port} (\text{tm-oports} (os)), \\
& \quad \quad \text{os-status-table} (os)) \\
& \quad = '4))) \\
\rightarrow & \text{permutation} (\text{mapup-queue} (\text{array-enq} (\text{tm-interrupting-output-port} (\text{tm-oports} (os)), \\
& \quad \quad \text{os-readyq} (os))), \\
& \quad \quad \text{os-ready-set} (\text{os-waiting-output-handler-path1} (os)))
\end{aligned}$$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path1.

THEOREM: not-tm-waiting-os-waiting-output-handler-path1
 $\neg \text{tm-waiting} (\text{os-waiting-output-handler-path1} (os))$

EVENT: Disable not-tm-waiting-os-waiting-output-handler-path1.

THEOREM: tm-base-os-waiting-output-handler-path1
 $\text{tm-base} (\text{os-waiting-output-handler-path1} (os))$
 $= \text{base} (\text{getnth} (\text{tm-interrupting-output-port} (\text{tm-oports} (os)),$
 $\quad \text{table} ('2, \text{os-segment-table} (os))))$

EVENT: Disable tm-base-os-waiting-output-handler-path1.

THEOREM: tm-limit-os-waiting-output-handler-path1
 $\text{tm-limit} (\text{os-waiting-output-handler-path1} (os))$
 $= \text{limit} (\text{getnth} (\text{tm-interrupting-output-port} (\text{tm-oports} (os)),$
 $\quad \text{table} ('2, \text{os-segment-table} (os))))$

EVENT: Disable tm-limit-os-waiting-output-handler-path1.

THEOREM: tm-base-tm-limit-os-waiting-output-handler-path1
 $(\text{good-os} (os) \wedge \text{tm-waiting} (os))$
 $\rightarrow ((\text{tm-base} (\text{os-waiting-output-handler-path1} (os))$
 $\quad = \text{base} (\text{getnth} (\text{os-current-taskid} (\text{os-waiting-output-handler-path1} (os)),$
 $\quad \quad \text{table} ('2, \text{os-segment-table} (os))))))$
 $\wedge (\text{tm-limit} (\text{os-waiting-output-handler-path1} (os))$
 $\quad = \text{limit} (\text{getnth} (\text{os-current-taskid} (\text{os-waiting-output-handler-path1} (os)),$
 $\quad \quad \text{table} ('2, \text{os-segment-table} (os))))))$

EVENT: Disable tm-base-tm-limit-os-waiting-output-handler-path1.

THEOREM: good-os-os-waiting-output-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\
& \quad \wedge (\text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2) \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad = '4) \\
& \quad \quad \wedge \text{array-qemptyp } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os})))))) \\
& \rightarrow \text{good-os } (\text{os-waiting-output-handler-path1 } (os))
\end{aligned}$$

EVENT: Disable good-os-os-waiting-output-handler-path1.

THEOREM: good-tm-os-waiting-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\
& \quad \wedge (\text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2) \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad = '4) \\
& \quad \quad \wedge (\neg \text{array-qemptyp } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os})))))) \\
& \rightarrow \text{good-tm } (\text{os-waiting-output-handler-path2 } (os))
\end{aligned}$$

EVENT: Disable good-tm-os-waiting-output-handler-path2.

THEOREM: tm-slimit-os-waiting-output-handler-path2

$$\text{tm-slimit } (\text{os-waiting-output-handler-path2 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-waiting-output-handler-path2.

THEOREM: os-code-os-waiting-output-handler-path2

$$\begin{aligned} & \text{good-tm } (os) \\ & \rightarrow (\text{os-code } (\text{os-waiting-output-handler-path2 } (os)) = \text{os-code } (os)) \end{aligned}$$

EVENT: Disable os-code-os-waiting-output-handler-path2.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path2

$$\begin{aligned}
& \text{good-tm } (os) \\
& \rightarrow ((\text{getnth } ('3, \text{tm-memory } (\text{os-waiting-output-handler-path2 } (os))) \\
& \quad = \text{getnth } ('3, \text{tm-memory } (os))) \\
& \quad \wedge ((\text{getnth } ('4, \text{tm-memory } (\text{os-waiting-output-handler-path2 } (os)))
\end{aligned}$$

$$\begin{aligned}
&= \text{getnth}('4, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('5, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path2}(os))) \\
&= \text{getnth}('5, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('6, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path2}(os))) \\
&= \text{getnth}('6, \text{tm-memory}(os)) \\
\wedge &((\text{getnth}('7, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path2}(os))) \\
&= \text{getnth}('7, \text{tm-memory}(os)) \\
\wedge &(\text{getnth}('11, \\
&\quad \text{tm-memory}(\text{os-waiting-output-handler-path2}(os))) \\
&= \text{getnth}('11, \\
&\quad \text{tm-memory}(os)))))))))
\end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path2
 $\neg \text{tm-in-supervisor-mode}(\text{os-waiting-output-handler-path2}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-output-handler-path2.

THEOREM: os-task-table-os-waiting-output-handler-path2
good-tm(*os*)
 $\rightarrow (\text{os-task-table}(\text{os-waiting-output-handler-path2}(os))$
 $= \text{os-task-table}(os))$

EVENT: Disable os-task-table-os-waiting-output-handler-path2.

THEOREM: os-segment-table-os-waiting-output-handler-path2
good-tm(*os*)
 $\rightarrow (\text{os-segment-table}(\text{os-waiting-output-handler-path2}(os))$
 $= \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-waiting-output-handler-path2.

THEOREM: os-readyq-os-waiting-output-handler-path2
good-tm(*os*)
 $\rightarrow (\text{os-readyq}(\text{os-waiting-output-handler-path2}(os))$
 $= \text{array-enq}(\text{tm-interrupting-output-port}(\text{tm-ports}(os)),$
 $\quad \text{os-readyq}(os))$

EVENT: Disable os-readyq-os-waiting-output-handler-path2.

THEOREM: os-ibuffers-os-waiting-output-handler-path2
good-tm (*os*)
→ (os-ibuffers (os-waiting-output-handler-path2 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-waiting-output-handler-path2.

THEOREM: os-obuffers-os-waiting-output-handler-path2
good-tm (*os*)
→ (os-obuffers (os-waiting-output-handler-path2 (*os*))
= os-obuffers-with-dequeued-character (*os*))

EVENT: Disable os-obuffers-os-waiting-output-handler-path2.

THEOREM: finite-number-queue-listp-os-obuffers-os-waiting-output-handler-path2
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-obuffers (os-waiting-output-handler-path2 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-obuffers-os-waiting-output-handler-path2.

THEOREM: os-mbuffers-os-waiting-output-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-waiting-output-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-output-handler-path2.

THEOREM: os-status-table-os-waiting-output-handler-path2
good-tm (*os*)
→ (os-status-table (os-waiting-output-handler-path2 (*os*))
= os-update-status (tm-interrupting-output-port (tm-oports (*os*)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-waiting-output-handler-path2.

THEOREM: good-status-list-os-status-table-os-waiting-output-handler-path2

$(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow \text{good-status-list}(\text{table}('2,$
 $\qquad\qquad\qquad \text{os-status-table}(\text{os-waiting-output-handler-path2}(os))))$

EVENT: Disable good-status-list-os-status-table-os-waiting-output-handler-path2.

THEOREM: os-ready-set-os-waiting-output-handler-path2
 $(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow (\text{os-ready-set}(\text{os-waiting-output-handler-path2}(os))$
 $\quad = \text{index-ready-set}('0,$
 $\qquad\qquad\qquad \text{putnth}('0\ 0,$
 $\qquad\qquad\qquad\qquad\qquad \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\qquad\qquad\qquad\qquad\qquad \text{table}('2, \text{os-status-table}(os))))$

EVENT: Disable os-ready-set-os-waiting-output-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path2
 $(\text{good-os}(os)$
 $\wedge (\text{tm-waiting}(os)$
 $\quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\quad\quad \wedge (\text{getnth}('2$
 $\quad\quad\quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad\quad\quad \text{os-status-table}(os))$
 $\quad\quad = '4))))$
 $\rightarrow \text{permutation}(\text{mapup-queue}(\text{array-enq}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\qquad\qquad\qquad \text{os-readyq}(os))),$
 $\qquad\qquad\qquad \text{os-ready-set}(\text{os-waiting-output-handler-path2}(os)))$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path2.

THEOREM: not-tm-waiting-os-waiting-output-handler-path2
 $\neg \text{tm-waiting}(\text{os-waiting-output-handler-path2}(os))$

EVENT: Disable not-tm-waiting-os-waiting-output-handler-path2.

THEOREM: tm-base-os-waiting-output-handler-path2
 $\text{tm-base}(\text{os-waiting-output-handler-path2}(os))$
 $= \text{base}(\text{getnth}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\qquad\qquad\qquad \text{table}('2, \text{os-segment-table}(os))))$

EVENT: Disable tm-base-os-waiting-output-handler-path2.

THEOREM: tm-limit-os-waiting-output-handler-path2

$$\begin{aligned} & \text{tm-limit (os-waiting-output-handler-path2 (os))} \\ &= \text{limit (getnth (tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \text{table ('2, os-segment-table (os))))} \end{aligned}$$

EVENT: Disable tm-limit-os-waiting-output-handler-path2.

THEOREM: tm-base-tm-limit-os-waiting-output-handler-path2

$$\begin{aligned} & (\text{good-os (os)} \wedge \text{tm-waiting (os)}) \\ & \rightarrow ((\text{tm-base (os-waiting-output-handler-path2 (os))} \\ & \quad = \text{base (getnth (os-current-taskid (os-waiting-output-handler-path2 (os)),} \\ & \quad \quad \text{table ('2, os-segment-table (os))))}) \\ & \wedge (\text{tm-limit (os-waiting-output-handler-path2 (os))} \\ & \quad = \text{limit (getnth (os-current-taskid (os-waiting-output-handler-path2 (os)),} \\ & \quad \quad \text{table ('2, os-segment-table (os))))})) \end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-waiting-output-handler-path2.

THEOREM: good-os-os-waiting-output-handler-path2

$$\begin{aligned} & (\text{good-os (os)} \\ & \wedge (\text{tm-some-output-interruptp (tm-oports (os))} \\ & \quad \wedge (\text{tm-waiting (os)} \\ & \quad \quad \wedge ((\text{getnth ('2} \\ & \quad \quad \quad * \text{tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os-status-table (os)} \\ & \quad \quad \quad = '4) \\ & \quad \quad \wedge (\neg \text{array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os)))))))))) \\ & \rightarrow \text{good-os (os-waiting-output-handler-path2 (os))} \end{aligned}$$

EVENT: Disable good-os-os-waiting-output-handler-path2.

THEOREM: good-tm-os-waiting-output-handler-path3

$$\begin{aligned} & (\text{good-os (os)} \\ & \wedge (\text{tm-some-output-interruptp (tm-oports (os))} \\ & \quad \wedge (\text{tm-waiting (os)} \\ & \quad \quad \wedge ((\text{getnth ('2} \\ & \quad \quad \quad * \text{tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os-status-table (os)} \\ & \quad \quad \quad \neq '4) \\ & \quad \quad \wedge \text{array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),} \\ & \quad \quad \quad \text{os)))))))))) \\ & \rightarrow \text{good-tm (os-waiting-output-handler-path3 (os))} \end{aligned}$$

EVENT: Disable good-tm-os-waiting-output-handler-path3.

THEOREM: tm-slimit-os-waiting-output-handler-path3
 $\text{tm-slimit}(\text{os-waiting-output-handler-path3}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-waiting-output-handler-path3.

THEOREM: os-code-os-waiting-output-handler-path3
 $\text{good-tm}(os) \rightarrow (\text{os-code}(\text{os-waiting-output-handler-path3}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-waiting-output-handler-path3.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path3
 $\text{good-tm}(os) \rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('3, \text{tm-memory}(os))) \wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('4, \text{tm-memory}(os))) \wedge ((\text{getnth}('5, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('5, \text{tm-memory}(os))) \wedge ((\text{getnth}('6, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('6, \text{tm-memory}(os))) \wedge ((\text{getnth}('7, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('7, \text{tm-memory}(os))) \wedge (\text{getnth}('11, \text{tm-memory}(\text{os-waiting-output-handler-path3}(os))) = \text{getnth}('11, \text{tm-memory}(os))))))))))$

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path3
 $\neg \text{tm-in-supervisor-mode}(\text{os-waiting-output-handler-path3}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-output-handler-path3.

THEOREM: os-task-table-os-waiting-output-handler-path3
 $\text{good-tm}(os)$

→ (os-task-table (os-waiting-output-handler-path3 (os))
= os-task-table (os))

EVENT: Disable os-task-table-os-waiting-output-handler-path3.

THEOREM: os-segment-table-os-waiting-output-handler-path3
good-tm (os)
→ (os-segment-table (os-waiting-output-handler-path3 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-waiting-output-handler-path3.

THEOREM: os-readyq-os-waiting-output-handler-path3
good-tm (os)
→ (os-readyq (os-waiting-output-handler-path3 (os)) = os-readyq (os))

EVENT: Disable os-readyq-os-waiting-output-handler-path3.

THEOREM: os-ibuffers-os-waiting-output-handler-path3
good-tm (os)
→ (os-ibuffers (os-waiting-output-handler-path3 (os)) = os-ibuffers (os))

EVENT: Disable os-ibuffers-os-waiting-output-handler-path3.

THEOREM: os-obuffers-os-waiting-output-handler-path3
good-tm (os)
→ (os-obuffers (os-waiting-output-handler-path3 (os)) = os-obuffers (os))

EVENT: Disable os-obuffers-os-waiting-output-handler-path3.

THEOREM: os-mbuffers-os-waiting-output-handler-path3
good-tm (os)
→ (os-mbuffers (os-waiting-output-handler-path3 (os)) = os-mbuffers (os))

EVENT: Disable os-mbuffers-os-waiting-output-handler-path3.

THEOREM: os-status-table-os-waiting-output-handler-path3
good-tm (os)
→ (os-status-table (os-waiting-output-handler-path3 (os))
= os-status-table (os))

EVENT: Disable os-status-table-os-waiting-output-handler-path3.

THEOREM: tm-waiting-os-waiting-output-handler-path3
tm-waiting (os-waiting-output-handler-path3 (os))

EVENT: Disable tm-waiting-os-waiting-output-handler-path3.

THEOREM: good-os-os-waiting-output-handler-path3
(good-os (os)
 \wedge (tm-some-output-interruptp (tm-oports (os))
 \wedge (tm-waiting (os)
 \wedge ((getnth ('2
* tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
 \neq '4)
 \wedge array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
os)))))))))
 \rightarrow good-os (os-waiting-output-handler-path3 (os))

EVENT: Disable good-os-os-waiting-output-handler-path3.

THEOREM: good-tm-os-waiting-output-handler-path4
(good-os (os)
 \wedge (tm-some-output-interruptp (tm-oports (os))
 \wedge (tm-waiting (os)
 \wedge ((getnth ('2
* tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
 \neq '4)
 \wedge (\neg array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
os)))))))))
 \rightarrow good-tm (os-waiting-output-handler-path4 (os))

EVENT: Disable good-tm-os-waiting-output-handler-path4.

THEOREM: tm-slimit-os-waiting-output-handler-path4
tm-slimit (os-waiting-output-handler-path4 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-waiting-output-handler-path4.

THEOREM: os-code-os-waiting-output-handler-path4
good-tm (os)
 \rightarrow (os-code (os-waiting-output-handler-path4 (os)) = os-code (os))

EVENT: Disable os-code-os-waiting-output-handler-path4.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & ((\text{getnth}('3, \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('3, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('4, \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('4, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('5, \\ & \quad \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('5, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('6, \\ & \quad \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('6, \text{tm-memory}(os))) \\ \wedge & ((\text{getnth}('7, \\ & \quad \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('7, \text{tm-memory}(os))) \\ \wedge & (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-waiting-output-handler-path4}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os))))))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path4

$$\neg \text{tm-in-supervisor-mode}(\text{os-waiting-output-handler-path4}(os))$$

EVENT: Disable not-tm-in-supervisor-mode-os-waiting-output-handler-path4.

THEOREM: os-task-table-os-waiting-output-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-task-table}(\text{os-waiting-output-handler-path4}(os)) \\ & = \text{os-task-table}(os)) \end{aligned}$$

EVENT: Disable os-task-table-os-waiting-output-handler-path4.

THEOREM: os-segment-table-os-waiting-output-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-segment-table}(\text{os-waiting-output-handler-path4}(os)) \\ & = \text{os-segment-table}(os)) \end{aligned}$$

EVENT: Disable os-segment-table-os-waiting-output-handler-path4.

THEOREM: os-readyq-os-waiting-output-handler-path4

$$\begin{aligned} & \text{good-tm}(os) \\ \rightarrow & (\text{os-readyq}(\text{os-waiting-output-handler-path4}(os)) = \text{os-readyq}(os)) \end{aligned}$$

EVENT: Disable os-readyq-os-waiting-output-handler-path4.

THEOREM: os-ibuffers-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-ibuffers (os-waiting-output-handler-path4 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-waiting-output-handler-path4.

THEOREM: os-obuffers-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-obuffers (os-waiting-output-handler-path4 (*os*))
= os-obuffers-with-dequeued-character (*os*))

EVENT: Disable os-obuffers-os-waiting-output-handler-path4.

THEOREM: finite-number-queue-listp-os-obuffers-os-waiting-output-handler-path4
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (¬ array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-obuffers (os-waiting-output-handler-path4 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-obuffers-os-waiting-output-handler-path4.

THEOREM: os-mbuffers-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-mbuffers (os-waiting-output-handler-path4 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-output-handler-path4.

THEOREM: os-status-table-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-status-table (os-waiting-output-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-waiting-output-handler-path4.

THEOREM: tm-waiting-os-waiting-output-handler-path4
tm-waiting (os-waiting-output-handler-path4 (*os*))

EVENT: Disable tm-waiting-os-waiting-output-handler-path4.

THEOREM: good-os-os-waiting-output-handler-path4

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\ & \quad \wedge (\text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad \quad \neq '4) \\ & \quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os})))))) \\ & \rightarrow \text{good-os } (\text{os-waiting-output-handler-path4 } (os)) \end{aligned}$$

EVENT: Disable good-os-os-waiting-output-handler-path4.

THEOREM: good-os-os-waiting-output-handler

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \wedge \text{tm-waiting } (os)) \\ & \rightarrow \text{good-os } (\text{os-waiting-output-handler } (os)) \end{aligned}$$

EVENT: Disable good-os-os-waiting-output-handler.

THEOREM: finite-number-queuep-array-enq-interrupting-output-port2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \quad \wedge (\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad \quad = '4)))) \\ & \rightarrow \text{finite-number-queuep } (\text{array-enq } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-readyq } (os)), \\ & \quad \quad \quad '16, \\ & \quad \quad \quad '16) \end{aligned}$$

EVENT: Disable finite-number-queuep-array-enq-interrupting-output-port2.

THEOREM: good-tm-os-running-output-handler-path1

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \end{aligned}$$

$$\begin{aligned}
& \wedge \left(\left(\text{getnth}('2 \right. \right. \\
& \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \quad \quad \left. \text{os-status-table}(os) \right. \\
& \quad \quad \quad = '4) \\
& \quad \wedge \left. \left. \left. \text{array-qemptyp}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \right. \right. \right. \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \left. \left. \left. os \right) \right) \right) \right)
\end{aligned}$$

→ good-tm(os-running-output-handler-path1(os))

EVENT: Disable good-tm-os-running-output-handler-path1.

THEOREM: tm-slimit-os-running-output-handler-path1
tm-slimit(os-running-output-handler-path1(os)) = tm-slimit(os)

EVENT: Disable tm-slimit-os-running-output-handler-path1.

THEOREM: os-code-os-running-output-handler-path1
good-tm(os)
→ (os-code(os-running-output-handler-path1(os)) = os-code(os))

EVENT: Disable os-code-os-running-output-handler-path1.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path1
good-tm(os)
→ ((getnth('3, tm-memory(os-running-output-handler-path1(os)))
= getnth('3, tm-memory(os)))
 \wedge ((getnth('4, tm-memory(os-running-output-handler-path1(os)))
= getnth('4, tm-memory(os)))
 \wedge ((getnth('5,
tm-memory(os-running-output-handler-path1(os)))
= getnth('5, tm-memory(os)))
 \wedge ((getnth('6,
tm-memory(os-running-output-handler-path1(os)))
= getnth('6, tm-memory(os)))
 \wedge ((getnth('7,
tm-memory(os-running-output-handler-path1(os)))
= getnth('7, tm-memory(os)))
 \wedge (getnth('11,
tm-memory(os-running-output-handler-path1(os)))
= getnth('11,
tm-memory(os)))))))))

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path1
→ tm-in-supervisor-mode (os-running-output-handler-path1 (os))

EVENT: Disable not-tm-in-supervisor-mode-os-running-output-handler-path1.

THEOREM: os-task-table-os-running-output-handler-path1
good-tm (os)
→ (os-task-table (os-running-output-handler-path1 (os))
= os-new-task-table (os))

EVENT: Disable os-task-table-os-running-output-handler-path1.

THEOREM: os-segment-table-os-running-output-handler-path1
good-tm (os)
→ (os-segment-table (os-running-output-handler-path1 (os))
= os-segment-table (os))

EVENT: Disable os-segment-table-os-running-output-handler-path1.

THEOREM: os-readyq-os-running-output-handler-path1
good-tm (os)
→ (os-readyq (os-running-output-handler-path1 (os))
= array-enq (tm-interrupting-output-port (tm-ports (os)),
os-readyq (os)))

EVENT: Disable os-readyq-os-running-output-handler-path1.

THEOREM: os-ibuffers-os-running-output-handler-path1
good-tm (os)
→ (os-ibuffers (os-running-output-handler-path1 (os)) = os-ibuffers (os))

EVENT: Disable os-ibuffers-os-running-output-handler-path1.

THEOREM: os-obuffers-os-running-output-handler-path1
good-tm (os)
→ (os-obuffers (os-running-output-handler-path1 (os)) = os-obuffers (os))

EVENT: Disable os-obuffers-os-running-output-handler-path1.

THEOREM: os-mbuffers-os-running-output-handler-path1
good-tm (os)
→ (os-mbuffers (os-running-output-handler-path1 (os)) = os-mbuffers (os))

EVENT: Disable os-mbuffers-os-running-output-handler-path1.

THEOREM: os-status-table-os-running-output-handler-path1
good-tm (*os*)
→ (os-status-table (os-running-output-handler-path1 (*os*))
= os-update-status (tm-interrupting-output-port (tm-oports (*os*)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-running-output-handler-path1.

THEOREM: good-status-list-os-status-table-os-running-output-handler-path1
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ good-status-list (table ('2,
os-status-table (os-running-output-handler-path1 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-running-output-handler-path1.

THEOREM: os-ready-set-os-running-output-handler-path1
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (os-ready-set (os-running-output-handler-path1 (*os*))
= index-ready-set ('0,
putnth ('0 0),
tm-interrupting-output-port (tm-oports (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable os-ready-set-os-running-output-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-output-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
= '4))))))
→ permutation (mapup-queue (array-enq (tm-interrupting-output-port (tm-oports (*os*)),
os-readyq (*os*))),
os-ready-set (os-running-output-handler-path1 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-output-handler-path1.

THEOREM: not-tm-waiting-os-running-output-handler-path1
 $\neg \text{tm-waiting}(\text{os-running-output-handler-path1}(os))$

EVENT: Disable not-tm-waiting-os-running-output-handler-path1.

THEOREM: tm-base-os-running-output-handler-path1
 $\text{tm-base}(\text{os-running-output-handler-path1}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-running-output-handler-path1.

THEOREM: tm-limit-os-running-output-handler-path1
 $\text{tm-limit}(\text{os-running-output-handler-path1}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-output-handler-path1.

THEOREM: tm-base-tm-limit-os-running-output-handler-path1
 $(\text{good-os}(os) \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\text{getnth}('2 * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \text{os-status-table}(os)) = '4)))) \rightarrow ((\text{tm-base}(\text{os-running-output-handler-path1}(os)) = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-running-output-handler-path1}(os)), \text{table}('2, \text{os-segment-table}(os)))))) \wedge (\text{tm-limit}(\text{os-running-output-handler-path1}(os)) = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-running-output-handler-path1}(os)), \text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path1.

THEOREM: good-os-os-running-output-handler-path1
 $(\text{good-os}(os) \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{getnth}('2 * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \text{os-status-table}(os)) = '4) \wedge \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), os)))))) \rightarrow \text{good-os}(\text{os-running-output-handler-path1}(os))$

EVENT: Disable good-os-os-running-output-handler-path1.

THEOREM: good-tm-os-running-output-handler-path2

$$\begin{aligned} & (\text{good-os } (os) \\ & \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad = '4) \\ & \quad \wedge (\neg \text{array-qemptyp } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os}))))))))) \\ & \rightarrow \text{good-tm } (\text{os-running-output-handler-path2 } (os)) \end{aligned}$$

EVENT: Disable good-tm-os-running-output-handler-path2.

THEOREM: tm-slimit-os-running-output-handler-path2

$$\text{tm-slimit } (\text{os-running-output-handler-path2 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-running-output-handler-path2.

THEOREM: os-code-os-running-output-handler-path2

$$\begin{aligned} & \text{good-tm } (os) \\ & \rightarrow (\text{os-code } (\text{os-running-output-handler-path2 } (os)) = \text{os-code } (os)) \end{aligned}$$

EVENT: Disable os-code-os-running-output-handler-path2.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path2

$$\begin{aligned} & \text{good-tm } (os) \\ & \rightarrow ((\text{getnth } ('3, \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\ & \quad = \text{getnth } ('3, \text{tm-memory } (os))) \\ & \quad \wedge ((\text{getnth } ('4, \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\ & \quad \quad = \text{getnth } ('4, \text{tm-memory } (os))) \\ & \quad \quad \wedge ((\text{getnth } ('5, \\ & \quad \quad \quad \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\ & \quad \quad \quad = \text{getnth } ('5, \text{tm-memory } (os))) \\ & \quad \quad \wedge ((\text{getnth } ('6, \\ & \quad \quad \quad \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\ & \quad \quad \quad = \text{getnth } ('6, \text{tm-memory } (os))) \\ & \quad \quad \wedge ((\text{getnth } ('7, \\ & \quad \quad \quad \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\ & \quad \quad \quad = \text{getnth } ('7, \text{tm-memory } (os)))))) \end{aligned}$$

$$\begin{aligned} & \wedge (\text{getnth}('11, \\ & \quad \text{tm-memory}(\text{os-running-output-handler-path2}(os))) \\ & = \text{getnth}('11, \\ & \quad \text{tm-memory}(os)))))) \end{aligned}$$

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path2
 $\neg \text{tm-in-supervisor-mode}(\text{os-running-output-handler-path2}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-running-output-handler-path2.

THEOREM: os-task-table-os-running-output-handler-path2
 good-tm(*os*)
 $\rightarrow (\text{os-task-table}(\text{os-running-output-handler-path2}(os)) \\ = \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-running-output-handler-path2.

THEOREM: os-segment-table-os-running-output-handler-path2
 good-tm(*os*)
 $\rightarrow (\text{os-segment-table}(\text{os-running-output-handler-path2}(os)) \\ = \text{os-segment-table}(os))$

EVENT: Disable os-segment-table-os-running-output-handler-path2.

THEOREM: os-readyq-os-running-output-handler-path2
 good-tm(*os*)
 $\rightarrow (\text{os-readyq}(\text{os-running-output-handler-path2}(os)) \\ = \text{array-enq}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ \text{os-readyq}(os)))$

EVENT: Disable os-readyq-os-running-output-handler-path2.

THEOREM: os-ibuffers-os-running-output-handler-path2
 good-tm(*os*)
 $\rightarrow (\text{os-ibuffers}(\text{os-running-output-handler-path2}(os)) = \text{os-ibuffers}(os))$

EVENT: Disable os-ibuffers-os-running-output-handler-path2.

THEOREM: os-obuffers-os-running-output-handler-path2
 good-tm(*os*)
 $\rightarrow (\text{os-obuffers}(\text{os-running-output-handler-path2}(os)) \\ = \text{os-obuffers-with-dequeued-character}(os))$

EVENT: Disable os-obuffers-os-running-output-handler-path2.

THEOREM: finite-number-queue-listp-os-obuffers-os-running-output-handler-path2

$$\begin{aligned}
 & (\text{good-os } (os) \\
 & \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
 & \quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
 & \quad \quad \quad os)))))) \\
 \rightarrow & \text{finite-number-queue-listp } (\text{table } ('8, \\
 & \quad \quad \quad \text{os-obuffers } (\text{os-running-output-handler-path2 } (os))), \\
 & \quad \quad '4, \\
 & \quad \quad '65536)
 \end{aligned}$$

EVENT: Disable finite-number-queue-listp-os-obuffers-os-running-output-handler-path2.

THEOREM: os-mbuffers-os-running-output-handler-path2

$$\begin{aligned}
 & \text{good-tm } (os) \\
 \rightarrow & (\text{os-mbuffers } (\text{os-running-output-handler-path2 } (os)) = \text{os-mbuffers } (os))
 \end{aligned}$$

EVENT: Disable os-mbuffers-os-running-output-handler-path2.

THEOREM: os-status-table-os-running-output-handler-path2

$$\begin{aligned}
 & \text{good-tm } (os) \\
 \rightarrow & (\text{os-status-table } (\text{os-running-output-handler-path2 } (os)) \\
 & \quad = \text{os-update-status } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
 & \quad \quad '0, \\
 & \quad \quad '0, \\
 & \quad \quad os))
 \end{aligned}$$

EVENT: Disable os-status-table-os-running-output-handler-path2.

THEOREM: good-status-list-os-status-table-os-running-output-handler-path2

$$\begin{aligned}
 & (\text{good-os } (os) \wedge \text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\
 \rightarrow & \text{good-status-list } (\text{table } ('2, \\
 & \quad \quad \quad \text{os-status-table } (\text{os-running-output-handler-path2 } (os))))
 \end{aligned}$$

EVENT: Disable good-status-list-os-status-table-os-running-output-handler-path2.

THEOREM: os-ready-set-os-running-output-handler-path2

$$\begin{aligned}
 & (\text{good-os } (os) \wedge \text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\
 \rightarrow & (\text{os-ready-set } (\text{os-running-output-handler-path2 } (os)) \\
 & \quad = \text{index-ready-set } ('0,
 \end{aligned}$$

$\text{putnth}('0\ 0),$
 $\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{table}('2, \text{os-status-table}(os))$

EVENT: Disable os-ready-set-os-running-output-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-output-handler-path2

$(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\wedge (\text{getnth}('2$
 $\quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '4))))$
 $\rightarrow \text{permutation}(\text{mapup-queue}(\text{array-enq}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os-readyq}(os))),$
 $\quad \text{os-ready-set}(\text{os-running-output-handler-path2}(os)))$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-output-handler-path2.

THEOREM: not-tm-waiting-os-running-output-handler-path2

$\neg \text{tm-waiting}(\text{os-running-output-handler-path2}(os))$

EVENT: Disable not-tm-waiting-os-running-output-handler-path2.

THEOREM: tm-base-os-running-output-handler-path2

$\text{tm-base}(\text{os-running-output-handler-path2}(os)) = \text{tm-base}(os)$

EVENT: Disable tm-base-os-running-output-handler-path2.

THEOREM: tm-limit-os-running-output-handler-path2

$\text{tm-limit}(\text{os-running-output-handler-path2}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-output-handler-path2.

THEOREM: tm-base-tm-limit-os-running-output-handler-path2

$(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{getnth}('2$
 $\quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$

$$\begin{aligned}
& \text{os-status-table } (os) \\
& = \text{'4})) \\
\rightarrow & ((\text{tm-base } (\text{os-running-output-handler-path2 } (os)) \\
& = \text{base } (\text{getnth } (\text{os-current-taskid } (\text{os-running-output-handler-path2 } (os)), \\
& \quad \text{table } ('2, \text{os-segment-table } (os)))))) \\
& \wedge (\text{tm-limit } (\text{os-running-output-handler-path2 } (os)) \\
& = \text{limit } (\text{getnth } (\text{os-current-taskid } (\text{os-running-output-handler-path2 } (os)), \\
& \quad \text{table } ('2, \text{os-segment-table } (os))))))
\end{aligned}$$

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path2.

THEOREM: good-os-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad = \text{'4}) \\
& \quad \quad \wedge (\neg \text{array-qemptyp } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os))))))))) \\
\rightarrow & \text{good-os } (\text{os-running-output-handler-path2 } (os))
\end{aligned}$$

EVENT: Disable good-os-os-running-output-handler-path2.

THEOREM: good-tm-os-running-output-handler-path3

$$\begin{aligned}
& (\text{good-os } (os) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \neq \text{'4}) \\
& \quad \quad \wedge \text{array-qemptyp } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os))))))))) \\
\rightarrow & \text{good-tm } (\text{os-running-output-handler-path3 } (os))
\end{aligned}$$

EVENT: Disable good-tm-os-running-output-handler-path3.

THEOREM: tm-slimit-os-running-output-handler-path3

$$\text{tm-slimit } (\text{os-running-output-handler-path3 } (os)) = \text{tm-slimit } (os)$$

EVENT: Disable tm-slimit-os-running-output-handler-path3.

THEOREM: os-code-os-running-output-handler-path3
good-tm (*os*)
→ (os-code (os-running-output-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-output-handler-path3.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path3
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-running-output-handler-path3 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))))

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path3
¬ tm-in-supervisor-mode (os-running-output-handler-path3 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-running-output-handler-path3.

THEOREM: os-task-table-os-running-output-handler-path3
good-tm (*os*)
→ (os-task-table (os-running-output-handler-path3 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-running-output-handler-path3.

THEOREM: os-segment-table-os-running-output-handler-path3
good-tm (*os*)
→ (os-segment-table (os-running-output-handler-path3 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-running-output-handler-path3.

THEOREM: os-readyq-os-running-output-handler-path3
good-tm (*os*)
→ (os-readyq (os-running-output-handler-path3 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-running-output-handler-path3.

THEOREM: os-ibuffers-os-running-output-handler-path3
good-tm (*os*)
→ (os-ibuffers (os-running-output-handler-path3 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-running-output-handler-path3.

THEOREM: os-obuffers-os-running-output-handler-path3
good-tm (*os*)
→ (os-obuffers (os-running-output-handler-path3 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-output-handler-path3.

THEOREM: os-mbuffers-os-running-output-handler-path3
good-tm (*os*)
→ (os-mbuffers (os-running-output-handler-path3 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-output-handler-path3.

THEOREM: os-status-table-os-running-output-handler-path3
good-tm (*os*)
→ (os-status-table (os-running-output-handler-path3 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-running-output-handler-path3.

THEOREM: not-tm-waiting-os-running-output-handler-path3
¬ tm-waiting (os-running-output-handler-path3 (*os*))

EVENT: Disable not-tm-waiting-os-running-output-handler-path3.

THEOREM: tm-base-os-running-output-handler-path3
tm-base (os-running-output-handler-path3 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-output-handler-path3.

THEOREM: tm-limit-os-running-output-handler-path3
 $\text{tm-limit}(\text{os-running-output-handler-path3}(os)) = \text{tm-limit}(os)$

EVENT: Disable tm-limit-os-running-output-handler-path3.

THEOREM: tm-base-tm-limit-os-running-output-handler-path3
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow ((\text{tm-base}(\text{os-running-output-handler-path3}(os))$
 $\quad = \text{base}(\text{getnth}(\text{os-current-taskid}(\text{os-running-output-handler-path3}(os)),$
 $\quad \quad \text{table}('2, \text{os-segment-table}(os))))))$
 $\wedge (\text{tm-limit}(\text{os-running-output-handler-path3}(os))$
 $\quad = \text{limit}(\text{getnth}(\text{os-current-taskid}(\text{os-running-output-handler-path3}(os)),$
 $\quad \quad \text{table}('2, \text{os-segment-table}(os))))))$

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path3.

THEOREM: good-os-os-running-output-handler-path3
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\quad \wedge ((\neg \text{tm-waiting}(os))$
 $\quad \quad \wedge ((\text{getnth}('2$
 $\quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \quad \quad \text{os-status-table}(os))$
 $\quad \quad \quad \neq '4)$
 $\quad \quad \wedge \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \quad \quad \text{os))))))$
 $\rightarrow \text{good-os}(\text{os-running-output-handler-path3}(os))$

EVENT: Disable good-os-os-running-output-handler-path3.

THEOREM: good-tm-os-running-output-handler-path4
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\quad \wedge ((\neg \text{tm-waiting}(os))$
 $\quad \quad \wedge ((\text{getnth}('2$
 $\quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \quad \quad \text{os-status-table}(os))$
 $\quad \quad \quad \neq '4)$
 $\quad \quad \wedge (\neg \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \quad \quad \text{os))))))$
 $\rightarrow \text{good-tm}(\text{os-running-output-handler-path4}(os))$

EVENT: Disable good-tm-os-running-output-handler-path4.

THEOREM: tm-slimit-os-running-output-handler-path4
 $\text{tm-slimit}(\text{os-running-output-handler-path4}(os)) = \text{tm-slimit}(os)$

EVENT: Disable tm-slimit-os-running-output-handler-path4.

THEOREM: os-code-os-running-output-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-code}(\text{os-running-output-handler-path4}(os)) = \text{os-code}(os))$

EVENT: Disable os-code-os-running-output-handler-path4.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow ((\text{getnth}('3, \text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('3, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('4, \text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('4, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('5,$
 $\text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('5, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('6,$
 $\text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('6, \text{tm-memory}(os)))$
 $\wedge ((\text{getnth}('7,$
 $\text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('7, \text{tm-memory}(os)))$
 $\wedge (\text{getnth}('11,$
 $\text{tm-memory}(\text{os-running-output-handler-path4}(os)))$
 $= \text{getnth}('11,$
 $\text{tm-memory}(os))))))))))$

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path4
 $\neg \text{tm-in-supervisor-mode}(\text{os-running-output-handler-path4}(os))$

EVENT: Disable not-tm-in-supervisor-mode-os-running-output-handler-path4.

THEOREM: os-task-table-os-running-output-handler-path4
 $\text{good-tm}(os)$
 $\rightarrow (\text{os-task-table}(\text{os-running-output-handler-path4}(os))$
 $= \text{os-new-task-table}(os))$

EVENT: Disable os-task-table-os-running-output-handler-path4.

THEOREM: os-segment-table-os-running-output-handler-path4
good-tm (*os*)
→ (os-segment-table (os-running-output-handler-path4 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-running-output-handler-path4.

THEOREM: os-readyq-os-running-output-handler-path4
good-tm (*os*)
→ (os-readyq (os-running-output-handler-path4 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-running-output-handler-path4.

THEOREM: os-ibuffers-os-running-output-handler-path4
good-tm (*os*)
→ (os-ibuffers (os-running-output-handler-path4 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-running-output-handler-path4.

THEOREM: os-obuffers-os-running-output-handler-path4
good-tm (*os*)
→ (os-obuffers (os-running-output-handler-path4 (*os*))
= os-obuffers-with-dequeued-character (*os*))

EVENT: Disable os-obuffers-os-running-output-handler-path4.

THEOREM: finite-number-queue-listp-os-obuffers-os-running-output-handler-path4
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (¬ array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))
→ finite-number-queue-listp (table ('8,
os-obuffers (os-running-output-handler-path4 (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-obuffers-os-running-output-handler-path4.

THEOREM: os-mbuffers-os-running-output-handler-path4

good-tm (*os*)
 \rightarrow (os-mbuffers (os-running-output-handler-path4 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-output-handler-path4.

THEOREM: os-status-table-os-running-output-handler-path4
good-tm (*os*)
 \rightarrow (os-status-table (os-running-output-handler-path4 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-running-output-handler-path4.

THEOREM: not-tm-waiting-os-running-output-handler-path4
 \neg tm-waiting (os-running-output-handler-path4 (*os*))

EVENT: Disable not-tm-waiting-os-running-output-handler-path4.

THEOREM: tm-base-os-running-output-handler-path4
tm-base (os-running-output-handler-path4 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-output-handler-path4.

THEOREM: tm-limit-os-running-output-handler-path4
tm-limit (os-running-output-handler-path4 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-output-handler-path4.

THEOREM: tm-base-tm-limit-os-running-output-handler-path4
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (((tm-base (os-running-output-handler-path4 (*os*))
= base (getnth (os-current-taskid (os-running-output-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))
 \wedge (tm-limit (os-running-output-handler-path4 (*os*))
= limit (getnth (os-current-taskid (os-running-output-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path4.

THEOREM: good-os-os-running-output-handler-path4
(good-os (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge (\neg tm-waiting (*os*)))

THEOREM: equivalence-of-svc-interrupt-expressions
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow (\text{tm-execute-svc-interrupt}(os) = \text{os-intended-svc-interrupt}(os))$

EVENT: Disable equivalence-of-svc-interrupt-expressions.

THEOREM: lessp-4-crock
 $((a \in \mathbf{N})$
 $\wedge ((a < '4) \wedge ((a \neq '0) \wedge ((a \neq '1) \wedge (a \neq '2))))))$
 $\rightarrow (a = '3)$

THEOREM: remainder-4-equals-3
 $((n \bmod '4) \neq '0) \wedge (((n \bmod '4) \neq '1) \wedge ((n \bmod '4) \neq '2)))$
 $\rightarrow ((n \bmod '4) = '3)$

EVENT: Disable remainder-4-equals-3.

THEOREM: trace-svc-interrupt-handler
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-svc-interrupt}(os),$
 $\quad \text{os-time-for-svc-handler}(os))$
 $= \text{os-svc-handler}(os))$

EVENT: Disable trace-svc-interrupt-handler.

THEOREM: good-os-os-svc-handler
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow \text{good-os}(\text{os-svc-handler}(os))$

EVENT: Disable good-os-os-svc-handler.

DEFINITION:
 $\text{os-input-interrupt-handler}(os)$
 $= \text{if } \text{tm-waiting}(os) \text{ then } \text{os-waiting-input-handler}(os)$
 $\quad \text{else } \text{os-running-input-handler}(os) \text{ endif}$

DEFINITION:
 $\text{os-time-for-input-interrupt-handler}(os)$
 $= \text{if } \text{tm-waiting}(os) \text{ then } \text{os-time-for-waiting-input-handler}(os)$
 $\quad \text{else } \text{os-time-for-running-input-handler}(os) \text{ endif}$

THEOREM: trace-input-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-ports}(os)))$

$$\begin{aligned} &\rightarrow (\text{tm-processor}(\text{os-intended-input-interrupt}(os), \\ &\quad \text{os-time-for-input-interrupt-handler}(os)) \\ &= \text{os-input-interrupt-handler}(os) \end{aligned}$$

EVENT: Disable trace-input-interrupt-handler.

THEOREM: good-os-os-input-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow \text{good-os}(\text{os-input-interrupt-handler}(os))$

EVENT: Disable good-os-os-input-interrupt-handler.

DEFINITION:

$\text{os-output-interrupt-handler}(os)$
 $= \text{if tm-waiting}(os) \text{ then os-waiting-output-handler}(os)$
 $\quad \text{else os-running-output-handler}(os) \text{ endif}$

DEFINITION:

$\text{os-time-for-output-interrupt-handler}(os)$
 $= \text{if tm-waiting}(os) \text{ then os-time-for-waiting-output-handler}(os)$
 $\quad \text{else os-time-for-running-output-handler}(os) \text{ endif}$

THEOREM: trace-output-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow (\text{tm-processor}(\text{os-intended-output-interrupt}(os),$
 $\quad \text{os-time-for-output-interrupt-handler}(os))$
 $= \text{os-output-interrupt-handler}(os))$

EVENT: Disable trace-output-interrupt-handler.

THEOREM: good-os-os-output-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow \text{good-os}(\text{os-output-interrupt-handler}(os))$

EVENT: Disable good-os-os-output-interrupt-handler.

DEFINITION:

$\text{os-step}(os)$
 $= \text{if tm-input-interruptp}(os) \text{ then os-input-interrupt-handler}(os)$
 $\quad \text{elseif tm-output-interruptp}(os) \text{ then os-output-interrupt-handler}(os)$
 $\quad \text{elseif tm-waiting}(os) \text{ then } os$
 $\quad \text{elseif tm-errorp}(os) \text{ then os-error-handler}(os)$
 $\quad \text{elseif tm-clock-interruptp}(os) \text{ then os-clock-interrupt-handler}(os)$
 $\quad \text{elseif tm-svc-interruptp}(os) \text{ then os-svc-handler}(os)$
 $\quad \text{else tm-fetch-execute}(os) \text{ endif}$

DEFINITION:

```
os-processor (os, oracle)  
= if listp (oracle)  
  then os-processor (os-step (tm-post-interrupt (car (oracle), os)),  
                    cdr (oracle))  
  else os endif
```

THEOREM: good-os-os-step

$\text{good-os} (os) \rightarrow \text{good-os} (\text{os-step} (os))$

EVENT: Disable good-os-os-step.

THEOREM: good-os-os-processor

$\text{good-os} (os) \rightarrow \text{good-os} (\text{os-processor} (os, oracle))$

EVENT: Disable good-os-os-processor.

DEFINITION:

```
timed-tm-step (tm)  
= if tm-input-interruptp (tm)  
  then tm-processor (tm-execute-input-interrupt (tm),  
                  os-time-for-input-interrupt-handler (tm))  
  elseif tm-output-interruptp (tm)  
  then tm-processor (tm-execute-output-interrupt (tm),  
                  os-time-for-output-interrupt-handler (tm))  
  elseif tm-waiting (tm) then tm  
  elseif tm-errorp (tm)  
  then tm-processor (tm-execute-error-interrupt (tm),  
                  os-time-for-error-handler (tm))  
  elseif tm-clock-interruptp (tm)  
  then tm-processor (tm-execute-clock-interrupt (tm),  
                  os-time-for-clock-interrupt-handler (tm))  
  elseif tm-svc-interruptp (tm)  
  then tm-processor (tm-execute-svc-interrupt (tm),  
                  os-time-for-svc-handler (tm))  
  else tm-fetch-execute (tm) endif
```

DEFINITION:

```
timed-tm-processor (tm, oracle)  
= if listp (oracle)  
  then timed-tm-processor (timed-tm-step (tm-post-interrupt (car (oracle),  
                                                             tm)),  
                          cdr (oracle))  
  else tm endif
```

DEFINITION:

```
os-oracle-step (event, os)
=  if tm-input-interruptp (os)
    then cons (event, os-time-for-input-interrupt-handler (os))
    elseif tm-output-interruptp (os)
    then cons (event, os-time-for-output-interrupt-handler (os))
    elseif tm-waiting (os) then cons (event, 'nil)
    elseif tm-errorp (os) then cons (event, os-time-for-error-handler (os))
    elseif tm-clock-interruptp (os)
    then cons (event, os-time-for-clock-interrupt-handler (os))
    elseif tm-svc-interruptp (os)
    then cons (event, os-time-for-svc-handler (os))
    else cons (event, 'nil) endif
```

DEFINITION:

```
os-oracle (os, oracle)
=  if listp (oracle)
    then append (os-oracle-step (car (oracle),
                                tm-post-interrupt (car (oracle), os)),
                os-oracle (timed-tm-step (tm-post-interrupt (car (oracle), os)),
                            cdr (oracle)))
    else oracle endif
```

THEOREM: tm-processor-oracle-distributes

tm-processor (*tm*, append (*a*, *b*)) = tm-processor (tm-processor (*tm*, *a*), *b*)

EVENT: Disable tm-processor-oracle-distributes.

THEOREM: tm-processor-implements-timed-tm-step

tm-processor (*tm*, os-oracle-step (*x*, tm-post-interrupt (*x*, *tm*)))
= timed-tm-step (tm-post-interrupt (*x*, *tm*))

EVENT: Disable tm-processor-implements-timed-tm-step.

THEOREM: tm-implements-timed-tm

tm-processor (*tm*, os-oracle (*tm*, *oracle*)) = timed-tm-processor (*tm*, *oracle*)

EVENT: Disable tm-implements-timed-tm.

DEFINITION:

```
btimed-tm-processor (tm, oracle)
=  if listp (oracle)
    then timed-tm-step (tm-post-interrupt (car (oracle),
```

b timed-tm-processor (*tm*,
cdr (*oracle*)))

else *tm* endif

THEOREM: timed-tm-implements-btimed-tm

timed-tm-processor (*tm*, *oracle*) = b timed-tm-processor (*tm*, reverse (*oracle*))

EVENT: Disable timed-tm-implements-btimed-tm.

DEFINITION:

bos-processor (*os*, *oracle*)

= **if** listp (*oracle*)

then os-step (tm-post-interrupt (car (*oracle*),
bos-processor (*os*, cdr (*oracle*))))

else *os* endif

THEOREM: bos-implements-os

bos-processor (*os*, reverse (*oracle*)) = os-processor (*os*, *oracle*)

EVENT: Disable bos-implements-os.

THEOREM: timed-tm-step-implements-os-step

good-os (*os*) → (timed-tm-step (*os*) = os-step (*os*))

EVENT: Disable timed-tm-step-implements-os-step.

THEOREM: good-os-bos-processor

good-os (*os*) → good-os (bos-processor (*os*, *oracle*))

EVENT: Disable good-os-bos-processor.

THEOREM: good-os-timed-tm-step

good-os (*os*) → good-os (timed-tm-step (*os*))

EVENT: Disable good-os-timed-tm-step.

THEOREM: good-os-btimed-tm-processor

good-os (*os*) → good-os (b timed-tm-processor (*os*, *oracle*))

EVENT: Disable good-os-btimed-tm-processor.

THEOREM: b timed-tm-implements-bos

good-os (*os*)

→ (b timed-tm-processor (*os*, *oracle*) = bos-processor (*os*, *oracle*))

EVENT: Disable btimed-tm-implements-bos.

THEOREM: tm-implements-os

good-os (*os*)
→ (tm-processor (*os*, os-oracle (*os*, *oracle*)) = os-processor (*os*, *oracle*))

EVENT: Disable tm-implements-os.

THEOREM: mapup-base-os-clock-interrupt-handler

good-os (*os*)
→ (mapup-base (*taskid*, os-clock-interrupt-handler (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-clock-interrupt-handler.

THEOREM: mapup-limit-os-clock-interrupt-handler

good-os (*os*)
→ (mapup-limit (*taskid*, os-clock-interrupt-handler (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-clock-interrupt-handler.

THEOREM: getseg-tm-memory-os-clock-interrupt-handler

(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-clock-interrupt-handler (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-clock-interrupt-handler.

(PROVE-LEMMA
TM-CPU-OS-CLOCK-INTERRUPT-HANDLER-CROCK
(REWRITE)
(IMPLIES
(AND (GOOD-OS OS)
(NOT (TM-WAITING OS)))
(EQUAL

```

(TM-CPU (OS-CLOCK-INTERRUPT-HANDLER OS))
(CONS
  (GETNTH '0
    (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
      OS))
  (CONS
    (GETNTH '1
      (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
        OS))
    (CONS
      (GETNTH '2
        (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
          OS))
      (CONS
        (GETNTH '3
          (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
            OS))
        (CONS
          (GETNTH '4
            (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
              OS))
          (CONS
            (GETNTH '5
              (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                OS))
            (CONS
              (GETNTH '6
                (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                  OS))
              (CONS
                (GETNTH '7
                  (OS-NEW-REGS (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                    OS))
                (CONS
                  (TM-PACK-PSW
                    (OS-NEW-CC (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                      OS)
                    (OS-NEW-ERROR (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                      OS)
                    (OS-NEW-SVCFLAG (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                      OS)
                    (OS-NEW-SVCID (ARRAY-QFIRST (OS-CLOCK-NEW-READYQ OS))
                      OS))
                  'NIL)))))))))))))

```


((ENABLE OS-CLOCK-INTERRUPT-HANDLER TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
 TM-R5 TM-R6 TM-R7)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T)))

EVENT: Disable tm-cpu-os-clock-interrupt-handler-crock.

THEOREM: tm-cpu-os-clock-interrupt-handler-case1
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ (array-qfirst (os-clock-new-readyq (*os*)
 = array-qfirst (os-readyq (*os*))))))
 → (tm-cpu (os-clock-interrupt-handler (*os*)
 = getnth (os-current-taskid (os-clock-interrupt-handler (*os*)),
 table ('9,
 os-task-table (os-clock-interrupt-handler (*os*))))))

THEOREM: tm-cpu-os-clock-interrupt-handler-case2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ (array-qfirst (os-clock-new-readyq (*os*)
 ≠ array-qfirst (os-readyq (*os*))))))
 → (tm-cpu (os-clock-interrupt-handler (*os*)
 = getnth (os-current-taskid (os-clock-interrupt-handler (*os*)),
 table ('9,
 os-task-table (os-clock-interrupt-handler (*os*))))))

THEOREM: tm-cpu-os-clock-interrupt-handler
 (good-os (*os*) ∧ (¬ tm-waiting (*os*)))
 → (tm-cpu (os-clock-interrupt-handler (*os*)
 = getnth (os-current-taskid (os-clock-interrupt-handler (*os*)),
 table ('9,
 os-task-table (os-clock-interrupt-handler (*os*))))))

EVENT: Disable tm-cpu-os-clock-interrupt-handler.

THEOREM: mapup-cpu-os-clock-interrupt-handler
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*) ∧ ((*taskid* ∈ \mathbf{N}) ∧ (*taskid* < '16))))
 → (mapup-cpu (*taskid*, os-clock-interrupt-handler (*os*)
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-clock-interrupt-handler.

THEOREM: mapup-regs-os-clock-interrupt-handler
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs } (taskid, \text{os-clock-interrupt-handler } (os))$
 $= \text{mapup-regs } (taskid, os))$

EVENT: Disable mapup-regs-os-clock-interrupt-handler.

THEOREM: mapup-cc-os-clock-interrupt-handler
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cc } (taskid, \text{os-clock-interrupt-handler } (os))$
 $= \text{mapup-cc } (taskid, os))$

EVENT: Disable mapup-cc-os-clock-interrupt-handler.

THEOREM: mapup-error-os-clock-interrupt-handler
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-error } (taskid, \text{os-clock-interrupt-handler } (os))$
 $= \text{mapup-error } (taskid, os))$

EVENT: Disable mapup-error-os-clock-interrupt-handler.

THEOREM: mapup-svcflag-os-clock-interrupt-handler
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-svcflag } (taskid, \text{os-clock-interrupt-handler } (os))$
 $= \text{mapup-svcflag } (taskid, os))$

EVENT: Disable mapup-svcflag-os-clock-interrupt-handler.

THEOREM: mapup-svcid-os-clock-interrupt-handler
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-svcid } (taskid, \text{os-clock-interrupt-handler } (os))$
 $= \text{mapup-svcid } (taskid, os))$

EVENT: Disable mapup-svcid-os-clock-interrupt-handler.

THEOREM: mapup-task-os-clock-interrupt-handler
 (good-os (*os*)

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-clock-interrupt-handler.

$$\begin{aligned}
\text{THEOREM: mapup-tasks-os-clock-interrupt-handler} \\
& (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (taskid \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-clock-interrupt-handler.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-clock-interrupt-handler} \\
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-clock-interrupt-handler.

$$\begin{aligned}
\text{THEOREM: mapup-os-ibuffers-os-clock-interrupt-handler} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-clock-interrupt-handler.

$$\begin{aligned}
\text{THEOREM: mapup-os-obuffers-os-clock-interrupt-handler} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-clock-interrupt-handler.

$$\begin{aligned}
\text{THEOREM: mapup-os-mbuffers-os-clock-interrupt-handler} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-clock-interrupt-handler}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-clock-interrupt-handler.

THEOREM: mapup-queue-os-readyq-os-clock-interrupt-handler

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-queue}(\text{os-readyq}(\text{os-clock-interrupt-handler}(os))) \\
& = \text{enq}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \\
& \quad \text{deq}(\text{mapup-queue}(\text{os-readyq}(os))))
\end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: mapup-os-status-table-os-clock-interrupt-handler} \\
& \text{good-os}(os) \\
\rightarrow & (\text{table}('2, \text{os-status-table}(\text{os-clock-interrupt-handler}(os))) \\
& = \text{table}('2, \text{os-status-table}(os)))
\end{aligned}$$

EVENT: Disable mapup-os-status-table-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: tm-rwstate-os-clock-interrupt-handler} \\
& \text{tm-rwstate}(\text{os-clock-interrupt-handler}(os)) = '0
\end{aligned}$$

EVENT: Disable tm-rwstate-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: tm-clock-os-clock-interrupt-handler} \\
& \text{tm-clock}(\text{os-clock-interrupt-handler}(os)) = '1000
\end{aligned}$$

EVENT: Disable tm-clock-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: tm-iports-os-clock-interrupt-handler} \\
& \text{tm-iports}(\text{os-clock-interrupt-handler}(os)) = \text{tm-iports}(os)
\end{aligned}$$

EVENT: Disable tm-iports-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: tm-oports-os-clock-interrupt-handler} \\
& \text{tm-oports}(\text{os-clock-interrupt-handler}(os)) = \text{tm-oports}(os)
\end{aligned}$$

EVENT: Disable tm-oports-os-clock-interrupt-handler.

$$\begin{aligned}
& \text{THEOREM: correctness-of-clock-interrupt-handler} \\
& (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os)))) \\
\rightarrow & (\text{mapup-os}(\text{os-clock-interrupt-handler}(os)) \\
& = \text{ak-clock-interrupt-handler}(\text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-clock-interrupt-handler.

THEOREM: mapup-base-os-error-handler-path1
 good-os (*os*)
 \rightarrow (mapup-base (*taskid*, os-error-handler-path1 (*os*)))
 $=$ mapup-base (*taskid*, *os*)

EVENT: Disable mapup-base-os-error-handler-path1.

THEOREM: mapup-limit-os-error-handler-path1
 good-os (*os*)
 \rightarrow (mapup-limit (*taskid*, os-error-handler-path1 (*os*)))
 $=$ mapup-limit (*taskid*, *os*)

EVENT: Disable mapup-limit-os-error-handler-path1.

THEOREM: getseg-tm-memory-os-error-handler-path1
 (good-os (*os*) \wedge (*taskid* < '16))
 \rightarrow (getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (os-error-handler-path1 (*os*)))
 $=$ getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-error-handler-path1.

THEOREM: mapup-cpu-os-error-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
 \rightarrow (mapup-cpu (*taskid*, os-error-handler-path1 (*os*)))
 $=$ mapup-cpu (*taskid*, *os*)

EVENT: Disable mapup-cpu-os-error-handler-path1.

THEOREM: mapup-regs-os-error-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
 \rightarrow (mapup-regs (*taskid*, os-error-handler-path1 (*os*)))
 $=$ mapup-regs (*taskid*, *os*)

EVENT: Disable mapup-regs-os-error-handler-path1.

THEOREM: mapup-cc-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
& \rightarrow (\text{mapup-cc } (\text{taskid}, \text{os-error-handler-path1 } (os)) = \text{mapup-cc } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-error-handler-path1.

THEOREM: mapup-error-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
& \rightarrow (\text{mapup-error } (\text{taskid}, \text{os-error-handler-path1 } (os)) \\
& \quad = \text{mapup-error } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-error-handler-path1.

THEOREM: mapup-svcflag-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
& \rightarrow (\text{mapup-svcflag } (\text{taskid}, \text{os-error-handler-path1 } (os)) \\
& \quad = \text{mapup-svcflag } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-error-handler-path1.

THEOREM: mapup-svcid-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
& \rightarrow (\text{mapup-svcid } (\text{taskid}, \text{os-error-handler-path1 } (os)) \\
& \quad = \text{mapup-svcid } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-error-handler-path1.

THEOREM: mapup-task-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
& \rightarrow (\text{mapup-task } (\text{taskid}, \text{os-error-handler-path1 } (os)) \\
& \quad = \text{mapup-task } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-error-handler-path1.

THEOREM: mapup-tasks-os-error-handler-path1

$$\begin{aligned}
& (\text{good-os } (os) \wedge ((\neg \text{tm-waiting } (os)) \wedge (\text{taskid} \in \mathbf{N}))) \\
& \rightarrow (\text{mapup-tasks } (\text{taskid}, \text{os-error-handler-path1 } (os)) \\
& \quad = \text{mapup-tasks } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-error-handler-path1.

THEOREM: mapup-os-tasks-os-error-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-error-handler-path1 (*os*)) = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-error-handler-path1.

THEOREM: mapup-os-ibuffers-os-error-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-ibuffers (os-error-handler-path1 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-error-handler-path1.

THEOREM: mapup-os-obuffers-os-error-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-error-handler-path1 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-error-handler-path1.

THEOREM: mapup-os-mbuffers-os-error-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-error-handler-path1 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-error-handler-path1.

THEOREM: mapup-queue-os-readyq-os-error-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-error-handler-path1 (*os*)))
= deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-error-handler-path1.

THEOREM: mapup-os-status-table-os-error-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-error-handler-path1 (*os*)))
= putnth ('(1 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-error-handler-path1.

THEOREM: tm-rwstate-os-error-handler-path1
 $\text{tm-rwstate}(\text{os-error-handler-path1}(os)) = '1$

EVENT: Disable tm-rwstate-os-error-handler-path1.

THEOREM: tm-clock-os-error-handler-path1
 $\text{tm-clock}(\text{os-error-handler-path1}(os)) = \text{tm-clock}(os)$

EVENT: Disable tm-clock-os-error-handler-path1.

THEOREM: tm-iports-os-error-handler-path1
 $\text{tm-iports}(\text{os-error-handler-path1}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-error-handler-path1.

THEOREM: tm-oports-os-error-handler-path1
 $\text{tm-oports}(\text{os-error-handler-path1}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-error-handler-path1.

THEOREM: correctness-of-os-error-handler-path1
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))$
 $\rightarrow (\text{mapup-os}(\text{os-error-handler-path1}(os)) = \text{ak-error-handler}(\text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-error-handler-path1.

THEOREM: mapup-base-os-error-handler-path2
 $\text{good-os}(os) \rightarrow (\text{mapup-base}(taskid, \text{os-error-handler-path2}(os)) = \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-error-handler-path2.

THEOREM: mapup-limit-os-error-handler-path2
 $\text{good-os}(os) \rightarrow (\text{mapup-limit}(taskid, \text{os-error-handler-path2}(os)) = \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-error-handler-path2.

THEOREM: getseg-tm-memory-os-error-handler-path2
(good-os (*os*) \wedge (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-error-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-error-handler-path2.

```
(PROVE-LEMMA
  TM-CPU-OS-ERROR-HANDLER-PATH2-CROCK
  (REWRITE)
  (IMPLIES
    (AND (GOOD-OS OS)
          (NOT (TM-WAITING OS))))
  (EQUAL
    (TM-CPU (OS-ERROR-HANDLER-PATH2 OS))
    (CONS
      (GETNTH '0
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                     OS))
      (CONS
        (GETNTH '1
          (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                       OS))
        (CONS
          (GETNTH '2
            (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                          OS))
          (CONS
            (GETNTH '3
              (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                            OS))
            (CONS
              (GETNTH '4
                (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                              OS))
```

```

(CONS
  (GETNTH '5
    (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
      OS))
  (CONS
    (GETNTH '6
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS))
    (CONS
      (GETNTH '7
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS))
      (CONS
        (TM-PACK-PSW
          (OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS))
        'NIL)))))))))
((ENABLE OS-ERROR-HANDLER-PATH2 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4 TM-R5
  TM-R6 TM-R7)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-error-handler-path2-crock.

THEOREM: tm-cpu-os-error-handler-path2-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ (array-qfirst (array-deq (os-readyq (os)))
      = array-qfirst (os-readyq (os))))
→ (tm-cpu (os-error-handler-path2 (os))
  = getnth (os-current-taskid (os-error-handler-path2 (os)),
    table ('9, os-task-table (os-error-handler-path2 (os))))))

```

THEOREM: tm-cpu-os-error-handler-path2-case2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qemptyp (array-deq (os-readyq (os))))
      ∧ (array-qfirst (array-deq (os-readyq (os)))
        = array-qfirst (os-readyq (os))))))

```

$$\begin{aligned}
& \neq \text{array-qfirst}(\text{os-readyq}(os))) \\
\rightarrow & (\text{tm-cpu}(\text{os-error-handler-path2}(os)) \\
& = \text{getnth}(\text{os-current-taskid}(\text{os-error-handler-path2}(os)), \\
& \quad \text{table}('9, \text{os-task-table}(\text{os-error-handler-path2}(os))))))
\end{aligned}$$

THEOREM: tm-cpu-os-error-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow & (\text{tm-cpu}(\text{os-error-handler-path2}(os)) \\
& = \text{getnth}(\text{os-current-taskid}(\text{os-error-handler-path2}(os)), \\
& \quad \text{table}('9, \text{os-task-table}(\text{os-error-handler-path2}(os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-error-handler-path2.

THEOREM: mapup-cpu-os-error-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-error-handler-path2}(os)) \\
& = \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-error-handler-path2.

THEOREM: mapup-regs-os-error-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-regs}(\text{taskid}, \text{os-error-handler-path2}(os)) \\
& = \text{mapup-regs}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-error-handler-path2.

THEOREM: mapup-cc-os-error-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cc}(\text{taskid}, \text{os-error-handler-path2}(os)) = \text{mapup-cc}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-error-handler-path2.

THEOREM: mapup-error-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-error-handler-path2 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-error-handler-path2.

THEOREM: mapup-svcflag-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcflag (*taskid*, os-error-handler-path2 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-error-handler-path2.

THEOREM: mapup-svcid-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcid (*taskid*, os-error-handler-path2 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-error-handler-path2.

THEOREM: mapup-task-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-task (*taskid*, os-error-handler-path2 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-error-handler-path2.

THEOREM: mapup-tasks-os-error-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge (taskid \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-error-handler-path2}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-error-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-error-handler-path2} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-error-handler-path2}(os)) = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-error-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-ibuffers-os-error-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-error-handler-path2}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-error-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-obuffers-os-error-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-error-handler-path2}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-error-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-mbuffers-os-error-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-error-handler-path2}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-error-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-queue-os-readyq-os-error-handler-path2} \\
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-queue}(\text{os-readyq}(\text{os-error-handler-path2}(os))) \\
& = \text{deq}(\text{mapup-queue}(\text{os-readyq}(os))))
\end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-error-handler-path2.

THEOREM: mapup-os-status-table-os-error-handler-path2
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{table}('2, \text{os-status-table}(\text{os-error-handler-path2}(os)))$
 $= \text{putnth}('1\ 0,$
 $\quad \text{array-qfirst}(\text{os-readyq}(os)),$
 $\quad \text{table}('2, \text{os-status-table}(os)))$

EVENT: Disable mapup-os-status-table-os-error-handler-path2.

THEOREM: tm-clock-os-error-handler-path2
 $\text{tm-clock}(\text{os-error-handler-path2}(os)) = '1000$

EVENT: Disable tm-clock-os-error-handler-path2.

THEOREM: tm-iports-os-error-handler-path2
 $\text{tm-iports}(\text{os-error-handler-path2}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-error-handler-path2.

THEOREM: tm-oports-os-error-handler-path2
 $\text{tm-oports}(\text{os-error-handler-path2}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-error-handler-path2.

THEOREM: tm-rwstate-os-error-handler-path2
 $\text{tm-rwstate}(\text{os-error-handler-path2}(os)) = '0$

EVENT: Disable tm-rwstate-os-error-handler-path2.

THEOREM: correctness-of-os-error-handler-path2
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))))$
 $\rightarrow (\text{mapup-os}(\text{os-error-handler-path2}(os))$
 $= \text{ak-error-handler}(\text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-error-handler-path2.

THEOREM: correctness-of-os-error-handler
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{mapup-os}(\text{os-error-handler}(os)) = \text{ak-error-handler}(\text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-error-handler.

THEOREM: mapup-base-os-svc-send-handler-path1
 good-os (*os*)
 \rightarrow (mapup-base (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-send-handler-path1.

THEOREM: mapup-limit-os-svc-send-handler-path1
 good-os (*os*)
 \rightarrow (mapup-limit (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-send-handler-path1.

THEOREM: getseg-tm-memory-os-svc-send-handler-path1
 (good-os (*os*) \wedge (*taskid* < '16))
 \rightarrow (getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (os-svc-send-handler-path1 (*os*)))
 = getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-send-handler-path1.

THEOREM: mapup-cpu-os-svc-send-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
 \rightarrow (mapup-cpu (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-send-handler-path1.

THEOREM: mapup-regs-os-svc-send-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
 \rightarrow (mapup-regs (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-send-handler-path1.

THEOREM: mapup-cc-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cc } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-cc } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-send-handler-path1.

THEOREM: mapup-error-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-error } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-error } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-send-handler-path1.

THEOREM: mapup-svcflag-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-svcflag } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-svcflag } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-svc-send-handler-path1.

THEOREM: mapup-svcid-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-svcid } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-svcid } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-svcid-os-svc-send-handler-path1.

THEOREM: mapup-task-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-task } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-task } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-send-handler-path1.

THEOREM: mapup-tasks-os-svc-send-handler-path1

$$\begin{aligned}
& (\text{good-os } (os) \wedge ((\neg \text{tm-waiting } (os)) \wedge (\text{taskid} \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks } (\text{taskid}, \text{os-svc-send-handler-path1 } (os))) \\
& = \text{mapup-tasks } (\text{taskid}, os)
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-send-handler-path1.

THEOREM: mapup-os-tasks-os-svc-send-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-send-handler-path1 (*os*)) = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path1.

THEOREM: mapup-os-ibuffers-os-svc-send-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-ibuffers (os-svc-send-handler-path1 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-send-handler-path1.

THEOREM: mapup-os-obuffers-os-svc-send-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-svc-send-handler-path1 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-send-handler-path1.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-send-handler-path1 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path1.

THEOREM: mapup-queue-os-readyq-os-svc-send-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-send-handler-path1 (*os*)))
= deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-send-handler-path1.

THEOREM: mapup-os-status-table-os-svc-send-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-send-handler-path1 (*os*)))
= putnth (cons ('2, cons (os-destid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

→ (mapup-base (*taskid*, os-svc-send-handler-path2 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-send-handler-path2.

THEOREM: mapup-limit-os-svc-send-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-send-handler-path2 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-send-handler-path2.

THEOREM: getseg-tm-memory-os-svc-send-handler-path2
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-send-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-send-handler-path2.

```
(PROVE-LEMMA
  TM-CPU-OS-SVC-SEND-HANDLER-PATH2-CROCK
  (REWRITE)
  (IMPLIES
    (AND (GOOD-OS OS)
          (NOT (TM-WAITING OS))))
  (EQUAL
    (TM-CPU (OS-SVC-SEND-HANDLER-PATH2 OS))
    (CONS
      (GETNTH '0
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                     OS))
      (CONS
        (GETNTH '1
          (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                      OS))
        (CONS
          (GETNTH '2
```

```

(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '3
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '4
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '5
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '6
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '7
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(TM-PACK-PSW
(OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS)
(OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
'NIL)))))))))
((ENABLE OS-SVC-SEND-HANDLER-PATH2 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4 TM-R5
TM-R6 TM-R7)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-svc-send-handler-path2-crock.

THEOREM: tm-cpu-os-svc-send-handler-path2-case1
(good-os (*os*)
 \wedge (\neg tm-waiting (*os*)
 \wedge (array-qfirst (array-deq (os-readyq (*os*))))

$$\begin{aligned}
&= \text{array-qfirst}(\text{os-readyq}(os))) \\
\rightarrow &(\text{tm-cpu}(\text{os-svc-send-handler-path2}(os)) \\
&= \text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path2}(os)), \\
&\quad \text{table}('9, \text{os-task-table}(\text{os-svc-send-handler-path2}(os))))))
\end{aligned}$$

THEOREM: tm-cpu-os-svc-send-handler-path2-case2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\wedge ((\neg \text{tm-waiting}(os)) \\
&\quad \wedge ((\neg \text{array-qemptytyp}(\text{array-deq}(\text{os-readyq}(os)))) \\
&\quad \wedge (\text{array-qfirst}(\text{array-deq}(\text{os-readyq}(os))) \\
&\quad \quad \neq \text{array-qfirst}(\text{os-readyq}(os)))))) \\
\rightarrow &(\text{tm-cpu}(\text{os-svc-send-handler-path2}(os)) \\
&= \text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path2}(os)), \\
&\quad \text{table}('9, \text{os-task-table}(\text{os-svc-send-handler-path2}(os))))))
\end{aligned}$$

THEOREM: tm-cpu-os-svc-send-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\wedge ((\neg \text{tm-waiting}(os)) \\
&\quad \wedge (\neg \text{array-qemptytyp}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow &(\text{tm-cpu}(\text{os-svc-send-handler-path2}(os)) \\
&= \text{getnth}(\text{os-current-taskid}(\text{os-svc-send-handler-path2}(os)), \\
&\quad \text{table}('9, \text{os-task-table}(\text{os-svc-send-handler-path2}(os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-svc-send-handler-path2.

THEOREM: mapup-cpu-os-svc-send-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\wedge ((\neg \text{tm-waiting}(os)) \\
&\quad \wedge ((\neg \text{array-qemptytyp}(\text{array-deq}(\text{os-readyq}(os)))) \\
&\quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow &(\text{mapup-cpu}(\text{taskid}, \text{os-svc-send-handler-path2}(os)) \\
&= \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path2.

THEOREM: mapup-regs-os-svc-send-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\wedge ((\neg \text{tm-waiting}(os)) \\
&\quad \wedge ((\neg \text{array-qemptytyp}(\text{array-deq}(\text{os-readyq}(os)))) \\
&\quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow &(\text{mapup-regs}(\text{taskid}, \text{os-svc-send-handler-path2}(os)) \\
&= \text{mapup-regs}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-send-handler-path2.

THEOREM: mapup-cc-os-svc-send-handler-path2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ ((¬ array-qempty (array-deq (os-readyq (*os*)))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-cc (*taskid*, os-svc-send-handler-path2 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-send-handler-path2.

THEOREM: mapup-error-os-svc-send-handler-path2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ ((¬ array-qempty (array-deq (os-readyq (*os*)))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-error (*taskid*, os-svc-send-handler-path2 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-send-handler-path2.

THEOREM: mapup-svcflag-os-svc-send-handler-path2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ ((¬ array-qempty (array-deq (os-readyq (*os*)))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-svcflag (*taskid*, os-svc-send-handler-path2 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-svc-send-handler-path2.

THEOREM: mapup-svcid-os-svc-send-handler-path2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)
 ∧ ((¬ array-qempty (array-deq (os-readyq (*os*)))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-svcid (*taskid*, os-svc-send-handler-path2 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-send-handler-path2.

THEOREM: mapup-task-os-svc-send-handler-path2
 (good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-svc-send-handler-path2}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-send-handler-path2.

THEOREM: mapup-tasks-os-svc-send-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \quad \quad \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-send-handler-path2}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-send-handler-path2.

THEOREM: mapup-os-tasks-os-svc-send-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-svc-send-handler-path2}(os)) = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path2.

THEOREM: mapup-os-ibuffers-os-svc-send-handler-path2

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-svc-send-handler-path2}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-svc-send-handler-path2.

THEOREM: mapup-os-obuffers-os-svc-send-handler-path2

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-svc-send-handler-path2}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-svc-send-handler-path2.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path2

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-svc-send-handler-path2}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-send-handler-path2
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (mapup-queue(os-readyq(os-svc-send-handler-path2(*os*)))
= deq(mapup-queue(os-readyq(*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-send-handler-path2.

THEOREM: mapup-os-status-table-os-svc-send-handler-path2
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (table('2, os-status-table(os-svc-send-handler-path2(*os*)))
= putnth(cons('2, cons(os-destid(*os*, '16), 'nil)),
array-qfirst(os-readyq(*os*)),
table('2, os-status-table(*os*))))

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path2.

THEOREM: tm-clock-os-svc-send-handler-path2
tm-clock(os-svc-send-handler-path2(*os*)) = '1000

EVENT: Disable tm-clock-os-svc-send-handler-path2.

THEOREM: tm-iports-os-svc-send-handler-path2
tm-iports(os-svc-send-handler-path2(*os*)) = tm-iports(*os*)

EVENT: Disable tm-iports-os-svc-send-handler-path2.

THEOREM: tm-oports-os-svc-send-handler-path2
tm-oports(os-svc-send-handler-path2(*os*)) = tm-oports(*os*)

EVENT: Disable tm-oports-os-svc-send-handler-path2.

THEOREM: tm-rwstate-os-svc-send-handler-path2
tm-rwstate(os-svc-send-handler-path2(*os*)) = '0

EVENT: Disable tm-rwstate-os-svc-send-handler-path2.

THEOREM: correctness-of-os-svc-send-handler-path2
(good-os(*os*)
 \wedge ((\neg tm-waiting(*os*))
 \wedge ((\neg tm-errorp(*os*)))

$$\begin{aligned}
& \wedge \quad (((\text{tm-svcid}(os) \bmod '4) = '0) \\
& \quad \wedge \quad (\text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \text{os-destid}(os, \\
& \quad \quad \quad '16), \\
& \quad \quad \quad os))) \\
& \quad \wedge \quad (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow & \quad (\text{mapup-os}(\text{os-svc-send-handler-path2}(os)) \\
& \quad = \quad \text{ak-execute-send}(\text{os-message}(os), \\
& \quad \quad \quad \text{os-current-taskid}(os), \\
& \quad \quad \quad \text{os-destid}(os, '16), \\
& \quad \quad \quad \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-svc-send-handler-path2.

THEOREM: mapup-base-os-svc-send-handler-path3

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & \quad (\text{mapup-base}(taskid, \text{os-svc-send-handler-path3}(os)) \\
& \quad = \quad \text{mapup-base}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-base-os-svc-send-handler-path3.

THEOREM: mapup-limit-os-svc-send-handler-path3

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & \quad (\text{mapup-limit}(taskid, \text{os-svc-send-handler-path3}(os)) \\
& \quad = \quad \text{mapup-limit}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-limit-os-svc-send-handler-path3.

THEOREM: getseg-tm-memory-os-svc-send-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (taskid < '16)) \\
\rightarrow & \quad (\text{getseg}(\text{mapup-base}(taskid, os), \\
& \quad \quad \quad \text{mapup-limit}(taskid, os), \\
& \quad \quad \quad \text{tm-memory}(\text{os-svc-send-handler-path3}(os))) \\
& \quad = \quad \text{getseg}(\text{mapup-base}(taskid, os), \\
& \quad \quad \quad \text{mapup-limit}(taskid, os), \\
& \quad \quad \quad \text{tm-memory}(os)))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-svc-send-handler-path3.

THEOREM: tm-cpu-os-svc-send-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & \quad (\text{tm-cpu}(\text{os-svc-send-handler-path3}(os)) \\
& \quad = \quad \text{putnth}(\text{tm-pack-psw}(\text{tm-cc}(os), \text{tm-error}(os), '0, \text{tm-svcid}(os)),
\end{aligned}$$

'8,
tm-cpu(*os*))

EVENT: Disable tm-cpu-os-svc-send-handler-path3.

THEOREM: array-qfirst-os-readyq-os-svc-send-handler-path3
(good-os(*os*)
 \wedge (\neg tm-waiting(*os*) \wedge (\neg array-qfullp(os-readyq(*os*))))
 \rightarrow (array-qfirst(os-readyq(os-svc-send-handler-path3(*os*)))
= array-qfirst(os-readyq(*os*)))

EVENT: Disable array-qfirst-os-readyq-os-svc-send-handler-path3.

THEOREM: mapup-cpu-os-svc-send-handler-path3-case1
(good-os(*os*)
 \wedge (\neg tm-waiting(*os*) \wedge (\neg array-qfullp(os-readyq(*os*))))
 \rightarrow (mapup-cpu(array-qfirst(os-readyq(*os*)), os-svc-send-handler-path3(*os*))
= putnth(tm-pack-psw(mapup-cc(array-qfirst(os-readyq(*os*)), *os*),
mapup-error(array-qfirst(os-readyq(*os*)), *os*),
'0,
mapup-svcid(array-qfirst(os-readyq(*os*)), *os*)),
'8,
mapup-cpu(array-qfirst(os-readyq(*os*)), *os*))

EVENT: Disable mapup-cpu-os-svc-send-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-send-handler-path3-case2
(good-os(*os*)
 \wedge (\neg tm-waiting(*os*)
 \wedge (\neg array-qfullp(os-readyq(*os*)))
 \wedge ((*taskid* \in \mathbf{N}
 \wedge (*taskid* < '16)
 \wedge (*taskid* \neq array-qfirst(os-readyq(*os*))))))))
 \rightarrow (mapup-cpu(*taskid*, os-svc-send-handler-path3(*os*))
= mapup-cpu(*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-send-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-send-handler-path3
(good-os(*os*)
 \wedge (\neg tm-waiting(*os*)
 \wedge (\neg array-qfullp(os-readyq(*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))

$$\begin{aligned}
&\rightarrow (\text{mapup-cpu}(taskid, \text{os-svc-send-handler-path3}(os)) \\
&= \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \\
&\quad \text{then putnth}(\text{tm-pack-psw}(\text{mapup-cc}(taskid, os), \\
&\quad\quad\quad \text{mapup-error}(taskid, os), \\
&\quad\quad\quad '0, \\
&\quad\quad\quad \text{mapup-svcid}(taskid, os)), \\
&\quad\quad\quad '8, \\
&\quad\quad\quad \text{mapup-cpu}(taskid, os)) \\
&\quad \text{else mapup-cpu}(taskid, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path3.

THEOREM: mapup-regs-os-svc-send-handler-path3

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os)) \\
&\quad\quad \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
&\quad\quad\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
&\rightarrow (\text{mapup-regs}(taskid, \text{os-svc-send-handler-path3}(os)) \\
&= \text{mapup-regs}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-send-handler-path3.

THEOREM: mapup-cc-os-svc-send-handler-path3

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os)) \\
&\quad\quad \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
&\quad\quad\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
&\rightarrow (\text{mapup-cc}(taskid, \text{os-svc-send-handler-path3}(os)) \\
&= \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-send-handler-path3.

THEOREM: mapup-error-os-svc-send-handler-path3

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os)) \\
&\quad\quad \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
&\quad\quad\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
&\rightarrow (\text{mapup-error}(taskid, \text{os-svc-send-handler-path3}(os)) \\
&= \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-send-handler-path3.

THEOREM: mapup-svcflag-os-svc-send-handler-path3

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qfullp (os-readyq (os)))
      ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))))
→ (mapup-svcflag (taskid, os-svc-send-handler-path3 (os))
  = if taskid = array-qfirst (os-readyq (os)) then '0
    else mapup-svcflag (taskid, os) endif)

```

EVENT: Disable mapup-svcflag-os-svc-send-handler-path3.

THEOREM: mapup-svcid-os-svc-send-handler-path3

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qfullp (os-readyq (os)))
      ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))))
→ (mapup-svcid (taskid, os-svc-send-handler-path3 (os))
  = mapup-svcid (taskid, os))

```

EVENT: Disable mapup-svcid-os-svc-send-handler-path3.

THEOREM: mapup-task-os-svc-send-handler-path3

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qfullp (os-readyq (os)))
      ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))))
→ (mapup-task (taskid, os-svc-send-handler-path3 (os))
  = if taskid = array-qfirst (os-readyq (os))
    then tm-set-svcflag ('0, mapup-task (taskid, os))
    else mapup-task (taskid, os) endif)

```

EVENT: Disable mapup-task-os-svc-send-handler-path3.

THEOREM: mapup-tasks-os-svc-send-handler-path3-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qfullp (os-readyq (os)))
      ∧ ((taskid ∈ ℕ)
        ∧ (array-qfirst (os-readyq (os)) < taskid))))))
→ (mapup-tasks (taskid, os-svc-send-handler-path3 (os))
  = mapup-tasks (taskid, os))

```

EVENT: Disable mapup-tasks-os-svc-send-handler-path3-case1.

THEOREM: mapup-tasks-os-svc-send-handler-path3-case2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge ((\neg \text{array-qfullp } (\text{os-readyq } (os))) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \quad \wedge \text{if array-qfirst } (\text{os-readyq } (os)) < \text{taskid} \\
& \quad \quad \quad \quad \text{then } '*1*\text{false} \\
& \quad \quad \quad \quad \text{else } '*1*\text{true endif})))))) \\
\rightarrow & (\text{mapup-tasks } (\text{taskid}, \text{os-svc-send-handler-path3 } (os)) \\
& = \text{putnth } (\text{tm-set-svcflag } ('0, \\
& \quad \quad \quad \text{getnth } (\text{array-qfirst } (\text{os-readyq } (os)) \\
& \quad \quad \quad \quad - \text{taskid}, \\
& \quad \quad \quad \quad \text{mapup-tasks } (\text{taskid}, os)), \\
& \quad \quad \text{array-qfirst } (\text{os-readyq } (os)) - \text{taskid}, \\
& \quad \quad \text{mapup-tasks } (\text{taskid}, os)))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-send-handler-path3-case2.

THEOREM: mapup-os-tasks-os-svc-send-handler-path3-support

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge (\neg \text{array-qfullp } (\text{os-readyq } (os)))) \\
\rightarrow & (\text{mapup-os-tasks } (\text{os-svc-send-handler-path3 } (os)) \\
& = \text{putnth } (\text{tm-set-svcflag } ('0, \\
& \quad \quad \quad \text{getnth } (\text{array-qfirst } (\text{os-readyq } (os)), \\
& \quad \quad \quad \quad \text{mapup-os-tasks } (os)), \\
& \quad \quad \text{array-qfirst } (\text{os-readyq } (os)), \\
& \quad \quad \text{mapup-os-tasks } (os)))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path3-support.

THEOREM: mapup-os-tasks-os-svc-send-handler-path3

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{getnth } ('2 * \text{os-destid } (os), '16), \text{os-status-table } (os)) \\
& \quad \quad = '3)) \\
\rightarrow & (\text{mapup-os-tasks } (\text{os-svc-send-handler-path3 } (os)) \\
& = \text{putnth } (\text{tm-set-svcflag } ('0, \\
& \quad \quad \quad \text{getnth } (\text{qfirst } (\text{mapup-queue } (\text{os-readyq } (os))), \\
& \quad \quad \quad \quad \text{mapup-os-tasks } (os)), \\
& \quad \quad \text{qfirst } (\text{mapup-queue } (\text{os-readyq } (os))), \\
& \quad \quad \text{mapup-os-tasks } (os)))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path3.

THEOREM: mapup-os-ibuffers-os-svc-send-handler-path3
good-os (*os*)
→ (mapup-os-ibuffers (os-svc-send-handler-path3 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-send-handler-path3.

THEOREM: mapup-os-obuffers-os-svc-send-handler-path3
good-os (*os*)
→ (mapup-os-obuffers (os-svc-send-handler-path3 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-send-handler-path3.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path3-support
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)
∧ (¬ array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os))))))
→ (mapup-os-mbuffers (os-svc-send-handler-path3 (*os*))
= enq2 (tm-r3 (*os*),
array-qfirst (os-readyq (*os*),
os-destid (*os*, '16),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path3-support.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)
∧ (¬ array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os))))))
→ (mapup-os-mbuffers (os-svc-send-handler-path3 (*os*))
= enq2 (tm-r3 (*os*),
qfirst (mapup-queue (os-readyq (*os*)),
os-destid (*os*, '16),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path3.

THEOREM: mapup-queue-os-readyq-os-svc-send-handler-path3

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{getnth } ('2 * \text{os-destid } (os, '16), \text{os-status-table } (os)) \\
& \quad \quad = '3))) \\
\rightarrow & (\text{mapup-queue } (\text{os-readyq } (\text{os-svc-send-handler-path3 } (os))) \\
& \quad = \text{enq } (\text{os-destid } (os, '16), \text{mapup-queue } (\text{os-readyq } (os))))
\end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: mapup-os-status-table-os-svc-send-handler-path3} \\
& (\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{table } ('2, \text{os-status-table } (\text{os-svc-send-handler-path3 } (os))) \\
& \quad = \text{putnth } ('(0 0), \\
& \quad \quad \text{os-destid } (os, '16), \\
& \quad \quad \text{table } ('2, \text{os-status-table } (os))))
\end{aligned}$$

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: tm-clock-os-svc-send-handler-path3} \\
& \text{tm-clock } (\text{os-svc-send-handler-path3 } (os)) = \text{tm-clock } (os)
\end{aligned}$$

EVENT: Disable tm-clock-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: tm-iports-os-svc-send-handler-path3} \\
& \text{tm-iports } (\text{os-svc-send-handler-path3 } (os)) = \text{tm-iports } (os)
\end{aligned}$$

EVENT: Disable tm-iports-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: tm-oports-os-svc-send-handler-path3} \\
& \text{tm-oports } (\text{os-svc-send-handler-path3 } (os)) = \text{tm-oports } (os)
\end{aligned}$$

EVENT: Disable tm-oports-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: tm-rwstate-os-svc-send-handler-path3} \\
& (\text{good-tm } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{tm-rwstate } (\text{os-svc-send-handler-path3 } (os)) = \text{tm-rwstate } (os))
\end{aligned}$$

EVENT: Disable tm-rwstate-os-svc-send-handler-path3.

$$\begin{aligned}
& \text{THEOREM: ak-waiting-to-receivep-mapup-os} \\
& (\text{good-os } (os) \\
& \quad \wedge ((\neg \text{tm-waiting } (os))
\end{aligned}$$

$$\begin{aligned}
& \wedge ((\neg \text{tm-errorp}(os)) \\
& \quad \wedge ((\text{getnth}('2 * \text{os-destid}(os, '16), \\
& \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = '3) \\
& \quad \wedge (\text{getnth}('1 \\
& \quad \quad + ('2 * \text{os-destid}(os, '16)), \\
& \quad \quad \text{os-status-table}(os)) \\
& \quad \quad = \text{os-current-taskid}(os)))))) \\
\rightarrow & \text{ak-waiting-to-receivep}(\text{os-current-taskid}(os), \\
& \quad \text{os-destid}(os, '16), \\
& \quad \text{mapup-os}(os))
\end{aligned}$$

EVENT: Disable ak-waiting-to-receivep-mapup-os.

THEOREM: correctness-of-os-svc-send-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\neg \text{tm-errorp}(os)) \\
& \quad \quad \quad \wedge (((\text{tm-svcid}(os) \bmod '4) = '0) \\
& \quad \quad \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-current-mbuffer}(\text{os-current-taskid}(os), \\
& \quad \quad \quad \quad \quad \quad \quad \text{os-destid}(os, \\
& \quad \quad \quad \quad \quad \quad \quad \quad '16), \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad os)))))) \\
& \quad \quad \quad \wedge ((\text{getnth}('2 * \text{os-destid}(os, '16), \\
& \quad \quad \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad \quad \quad = '3) \\
& \quad \quad \quad \wedge (\text{getnth}('1 \\
& \quad \quad \quad \quad \quad + ('2 \\
& \quad \quad \quad \quad \quad \quad * \text{os-destid}(os, \\
& \quad \quad \quad \quad \quad \quad \quad '16)), \\
& \quad \quad \quad \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad \quad \quad = \text{os-current-taskid}(os))))))))) \\
\rightarrow & (\text{mapup-os}(\text{os-svc-send-handler-path3}(os)) \\
& \quad = \text{ak-execute-send}(\text{os-message}(os), \\
& \quad \quad \text{os-current-taskid}(os), \\
& \quad \quad \text{os-destid}(os, '16), \\
& \quad \quad \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-svc-send-handler-path3.

THEOREM: mapup-base-os-svc-send-handler-path4

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-base}(\text{taskid}, \text{os-svc-send-handler-path4}(os)) \\
& \quad = \text{mapup-base}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-base-os-svc-send-handler-path4.

THEOREM: mapup-limit-os-svc-send-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-send-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-send-handler-path4.

THEOREM: getseg-tm-memory-os-svc-send-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-send-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-send-handler-path4.

THEOREM: tm-cpu-os-svc-send-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-cpu (os-svc-send-handler-path4 (*os*))
= putnth (tm-pack-psw (tm-cc (*os*), tm-error (*os*), '0, tm-svcid (*os*)),
'8,
tm-cpu (*os*)))

EVENT: Disable tm-cpu-os-svc-send-handler-path4.

THEOREM: mapup-cpu-os-svc-send-handler-path4-case1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-cpu (array-qfirst (os-readyq (*os*)), os-svc-send-handler-path4 (*os*))
= putnth (tm-pack-psw (mapup-cc (array-qfirst (os-readyq (*os*)), *os*),
mapup-error (array-qfirst (os-readyq (*os*)), *os*),
'0,
mapup-svcid (array-qfirst (os-readyq (*os*)), *os*)),
'8,
mapup-cpu (array-qfirst (os-readyq (*os*)), *os*)))

EVENT: Disable mapup-cpu-os-svc-send-handler-path4-case1.

THEOREM: mapup-cpu-os-svc-send-handler-path4-case2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \wedge ((\text{taskid} < '16) \\
& \quad \quad \quad \wedge (\text{taskid} \neq \text{array-qfirst } (\text{os-readyq } (os)))))) \\
\rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-svc-send-handler-path4 } (os)) \\
& = \text{mapup-cpu } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path4-case2.

THEOREM: mapup-cpu-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-svc-send-handler-path4 } (os)) \\
& = \text{if } \text{taskid} = \text{array-qfirst } (\text{os-readyq } (os)) \\
& \quad \text{then putnth } (\text{tm-pack-psw } (\text{mapup-cc } (\text{taskid}, os), \\
& \quad \quad \text{mapup-error } (\text{taskid}, os), \\
& \quad \quad '0, \\
& \quad \quad \text{mapup-svcid } (\text{taskid}, os)), \\
& \quad '8, \\
& \quad \text{mapup-cpu } (\text{taskid}, os)) \\
& \text{else mapup-cpu } (\text{taskid}, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path4.

THEOREM: mapup-regs-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-regs } (\text{taskid}, \text{os-svc-send-handler-path4 } (os)) \\
& = \text{mapup-regs } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-send-handler-path4.

THEOREM: mapup-cc-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cc } (\text{taskid}, \text{os-svc-send-handler-path4 } (os)) \\
& = \text{mapup-cc } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-send-handler-path4.

THEOREM: mapup-error-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os } (os))
\end{aligned}$$

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-svc-send-handler-path4}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-send-handler-path4.

THEOREM: mapup-svcflag-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-svc-send-handler-path4}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \text{ then } '0 \\
& \quad \text{else } \text{mapup-svcflag}(taskid, os) \text{ endif})
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-svc-send-handler-path4.

THEOREM: mapup-svcid-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-svc-send-handler-path4}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-svc-send-handler-path4.

THEOREM: mapup-task-os-svc-send-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-svc-send-handler-path4}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then } \text{tm-set-svcflag}('0, \text{mapup-task}(taskid, os)) \\
& \quad \text{else } \text{mapup-task}(taskid, os) \text{ endif})
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-send-handler-path4.

THEOREM: mapup-tasks-os-svc-send-handler-path4-case1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (\text{array-qfirst}(\text{os-readyq}(os)) < taskid)))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-send-handler-path4}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-send-handler-path4-case1.

THEOREM: mapup-tasks-os-svc-send-handler-path4-case2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((taskid ∈ ℕ)
      ∧ if array-qfirst (os-readyq (os)) < taskid
        then '*1*false
        else '*1*true endif)))
→ (mapup-tasks (taskid, os-svc-send-handler-path4 (os))
  = putnth (tm-set-svcflag ('0,
    getnth (array-qfirst (os-readyq (os))
      - taskid,
      mapup-tasks (taskid, os))),
    array-qfirst (os-readyq (os)) - taskid,
    mapup-tasks (taskid, os)))

```

EVENT: Disable mapup-tasks-os-svc-send-handler-path4-case2.

```

THEOREM: mapup-os-tasks-os-svc-send-handler-path4-support
(good-os (os) ∧ (¬ tm-waiting (os)))
→ (mapup-os-tasks (os-svc-send-handler-path4 (os))
  = putnth (tm-set-svcflag ('0,
    getnth (array-qfirst (os-readyq (os)),
      mapup-os-tasks (os))),
    array-qfirst (os-readyq (os)),
    mapup-os-tasks (os)))

```

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path4-support.

```

THEOREM: mapup-os-tasks-os-svc-send-handler-path4
(good-os (os) ∧ (¬ tm-waiting (os)))
→ (mapup-os-tasks (os-svc-send-handler-path4 (os))
  = putnth (tm-set-svcflag ('0,
    getnth (qfirst (mapup-queue (os-readyq (os))),
      mapup-os-tasks (os))),
    qfirst (mapup-queue (os-readyq (os))),
    mapup-os-tasks (os)))

```

EVENT: Disable mapup-os-tasks-os-svc-send-handler-path4.

```

THEOREM: mapup-os-ibuffers-os-svc-send-handler-path4
good-os (os)
→ (mapup-os-ibuffers (os-svc-send-handler-path4 (os))
  = mapup-os-ibuffers (os))

```

EVENT: Disable mapup-os-ibuffers-os-svc-send-handler-path4.

THEOREM: mapup-os-obuffers-os-svc-send-handler-path4
 good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-svc-send-handler-path4 (*os*))
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-send-handler-path4.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path4-support
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
 os-destid (*os*, '16),
os))))))
 \rightarrow (mapup-os-mbuffers (os-svc-send-handler-path4 (*os*))
 = enq2 (tm-r3 (*os*),
 array-qfirst (os-readyq (*os*)),
 os-destid (*os*, '16),
 mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path4-support.

THEOREM: mapup-os-mbuffers-os-svc-send-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
 os-destid (*os*, '16),
os))))))
 \rightarrow (mapup-os-mbuffers (os-svc-send-handler-path4 (*os*))
 = enq2 (tm-r3 (*os*),
 qfirst (mapup-queue (os-readyq (*os*)),
 os-destid (*os*, '16),
 mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path4.

THEOREM: mapup-queue-os-readyq-os-svc-send-handler-path4
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-send-handler-path4 (*os*)))
 = mapup-queue (os-readyq (*os*)))

EVENT: Disable mapup-queue-os-readyq-os-svc-send-handler-path4.

THEOREM: mapup-os-status-table-os-svc-send-handler-path4

$(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{table}('2, \text{os-status-table}(\text{os-svc-send-handler-path4}(os)))$
 $\quad = \text{table}('2, \text{os-status-table}(os)))$

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path4.

THEOREM: tm-clock-os-svc-send-handler-path4
 $\text{tm-clock}(\text{os-svc-send-handler-path4}(os)) = \text{tm-clock}(os)$

EVENT: Disable tm-clock-os-svc-send-handler-path4.

THEOREM: tm-iports-os-svc-send-handler-path4
 $\text{tm-iports}(\text{os-svc-send-handler-path4}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-svc-send-handler-path4.

THEOREM: tm-oports-os-svc-send-handler-path4
 $\text{tm-oports}(\text{os-svc-send-handler-path4}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-svc-send-handler-path4.

THEOREM: tm-rwstate-os-svc-send-handler-path4
 $(\text{good-tm}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-rwstate}(\text{os-svc-send-handler-path4}(os)) = \text{tm-rwstate}(os))$

EVENT: Disable tm-rwstate-os-svc-send-handler-path4.

THEOREM: not-ak-waiting-to-receivep-mapup-os

$(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\quad \wedge ((\neg \text{tm-errorp}(os))$
 $\quad \quad \wedge (\neg ((\text{getnth}('2 * \text{os-destid}(os, '16),$
 $\quad \quad \quad \text{os-status-table}(os))$
 $\quad \quad \quad = '3)$
 $\quad \wedge (\text{getnth}('1$
 $\quad \quad \quad + ('2$
 $\quad \quad \quad \quad * \text{os-destid}(os,$
 $\quad \quad \quad \quad \quad \text{'16)),$
 $\quad \quad \quad \quad \text{os-status-table}(os))$
 $\quad \quad \quad = \text{os-current-taskid}(os))))))$
 $\rightarrow (\neg \text{ak-waiting-to-receivep}(\text{os-current-taskid}(os),$
 $\quad \text{os-destid}(os, '16),$
 $\quad \text{mapup-os}(os)))$

$\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(taskid, \text{os-svc-receive-handler-path1}(os)))$
 $= \text{mapup-base}(taskid, os)$

EVENT: Disable mapup-base-os-svc-receive-handler-path1.

THEOREM: mapup-limit-os-svc-receive-handler-path1
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-svc-receive-handler-path1}(os)))$
 $= \text{mapup-limit}(taskid, os)$

EVENT: Disable mapup-limit-os-svc-receive-handler-path1.

THEOREM: getseg-tm-memory-os-svc-receive-handler-path1
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(\text{os-svc-receive-handler-path1}(os)))$
 $= \text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(os))$

EVENT: Disable getseg-tm-memory-os-svc-receive-handler-path1.

THEOREM: mapup-cpu-os-svc-receive-handler-path1
 $(\text{good-os}(os)$
 $\quad \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu}(taskid, \text{os-svc-receive-handler-path1}(os)))$
 $= \text{mapup-cpu}(taskid, os)$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path1.

THEOREM: mapup-regs-os-svc-receive-handler-path1
 $(\text{good-os}(os)$
 $\quad \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs}(taskid, \text{os-svc-receive-handler-path1}(os)))$
 $= \text{mapup-regs}(taskid, os)$

EVENT: Disable mapup-regs-os-svc-receive-handler-path1.

THEOREM: mapup-cc-os-svc-receive-handler-path1
 $(\text{good-os}(os)$

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-receive-handler-path1.

$$\begin{aligned}
& \text{THEOREM: mapup-error-os-svc-receive-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-receive-handler-path1.

$$\begin{aligned}
& \text{THEOREM: mapup-svcflag-os-svc-receive-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-svc-receive-handler-path1.

$$\begin{aligned}
& \text{THEOREM: mapup-svcid-os-svc-receive-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-svc-receive-handler-path1.

$$\begin{aligned}
& \text{THEOREM: mapup-task-os-svc-receive-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-receive-handler-path1.

$$\begin{aligned}
& \text{THEOREM: mapup-tasks-os-svc-receive-handler-path1} \\
& (\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (taskid \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-receive-handler-path1}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-receive-handler-path1.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-receive-handler-path1 (*os*)))
= mapup-os-tasks (*os*)

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path1.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-ibuffers (os-svc-receive-handler-path1 (*os*)))
= mapup-os-ibuffers (*os*)

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path1.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-svc-receive-handler-path1 (*os*)))
= mapup-os-obuffers (*os*)

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path1.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-receive-handler-path1 (*os*)))
= mapup-os-mbuffers (*os*)

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path1.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-receive-handler-path1 (*os*))))
= deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path1.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path1
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-receive-handler-path1 (*os*))))
= putnth (cons ('3, cons (os-srcid (*os*, '16), 'nil)),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable mapup-base-os-svc-receive-handler-path2.

THEOREM: mapup-limit-os-svc-receive-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-receive-handler-path2 (*os*)))
= mapup-limit (*taskid*, *os*)

EVENT: Disable mapup-limit-os-svc-receive-handler-path2.

THEOREM: getseg-tm-memory-os-svc-receive-handler-path2
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-receive-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-receive-handler-path2.

```
(PROVE-LEMMA
  TM-CPU-OS-SVC-RECEIVE-HANDLER-PATH2-CROCK
  (REWRITE)
  (IMPLIES
    (AND (GOOD-OS OS)
          (NOT (TM-WAITING OS))))
  (EQUAL
    (TM-CPU (OS-SVC-RECEIVE-HANDLER-PATH2 OS))
    (CONS
      (GETNTH '0
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                     OS))
      (CONS
        (GETNTH '1
          (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                       OS))
        (CONS
          (GETNTH '2
            (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                          OS))
          (CONS
```

```

(GETNTH '3
  (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
    OS))
(CONS
  (GETNTH '4
    (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
      OS))
  (CONS
    (GETNTH '5
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS))
    (CONS
      (GETNTH '6
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS))
      (CONS
        (GETNTH '7
          (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS))
        (CONS
          (TM-PACK-PSW
            (OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
              OS)
            (OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
              OS)
            (OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
              OS)
            (OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
              OS))
          'NIL))))))))))
((ENABLE OS-SVC-RECEIVE-HANDLER-PATH2 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
  TM-R5 TM-R6 TM-R7)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-svc-receive-handler-path2-crock.

THEOREM: tm-cpu-os-svc-receive-handler-path2-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ (array-qfirst (array-deq (os-readyq (os)))
      = array-qfirst (os-readyq (os))))))
→ (tm-cpu (os-svc-receive-handler-path2 (os))
  = getnth (os-current-taskid (os-svc-receive-handler-path2 (os))),

```

table ('9,
os-task-table (os-svc-receive-handler-path2 (os))))

THEOREM: tm-cpu-os-svc-receive-handler-path2-case2
 (good-os (os)
 \wedge ((\neg tm-waiting (os))
 \wedge ((\neg array-qempty (array-deq (os-readyq (os))))
 \wedge (array-qfirst (array-deq (os-readyq (os)))
 \neq array-qfirst (os-readyq (os))))))
 \rightarrow (tm-cpu (os-svc-receive-handler-path2 (os))
 $=$ getnth (os-current-taskid (os-svc-receive-handler-path2 (os)),
table ('9,
os-task-table (os-svc-receive-handler-path2 (os))))))

THEOREM: tm-cpu-os-svc-receive-handler-path2
 (good-os (os)
 \wedge ((\neg tm-waiting (os))
 \wedge (\neg array-qempty (array-deq (os-readyq (os))))))
 \rightarrow (tm-cpu (os-svc-receive-handler-path2 (os))
 $=$ getnth (os-current-taskid (os-svc-receive-handler-path2 (os)),
table ('9,
os-task-table (os-svc-receive-handler-path2 (os))))))

EVENT: Disable tm-cpu-os-svc-receive-handler-path2.

THEOREM: mapup-cpu-os-svc-receive-handler-path2
 (good-os (os)
 \wedge ((\neg tm-waiting (os))
 \wedge ((\neg array-qempty (array-deq (os-readyq (os))))
 \wedge (($taskid \in \mathbf{N}$) \wedge ($taskid < '16$))))))
 \rightarrow (mapup-cpu ($taskid$, os-svc-receive-handler-path2 (os))
 $=$ mapup-cpu ($taskid$, os))

EVENT: Disable mapup-cpu-os-svc-receive-handler-path2.

THEOREM: mapup-regs-os-svc-receive-handler-path2
 (good-os (os)
 \wedge ((\neg tm-waiting (os))
 \wedge ((\neg array-qempty (array-deq (os-readyq (os))))
 \wedge (($taskid \in \mathbf{N}$) \wedge ($taskid < '16$))))))
 \rightarrow (mapup-regs ($taskid$, os-svc-receive-handler-path2 (os))
 $=$ mapup-regs ($taskid$, os))

EVENT: Disable mapup-regs-os-svc-receive-handler-path2.

THEOREM: mapup-cc-os-svc-receive-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-svc-receive-handler-path2 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-receive-handler-path2.

THEOREM: mapup-error-os-svc-receive-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-svc-receive-handler-path2 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-receive-handler-path2.

THEOREM: mapup-svcflag-os-svc-receive-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcflag (*taskid*, os-svc-receive-handler-path2 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-svc-receive-handler-path2.

THEOREM: mapup-svcid-os-svc-receive-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcid (*taskid*, os-svc-receive-handler-path2 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-receive-handler-path2.

THEOREM: mapup-task-os-svc-receive-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-receive-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-tasks-os-svc-receive-handler-path2} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \quad \quad \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-receive-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-os-tasks-os-svc-receive-handler-path2} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-os-obuffers-os-svc-receive-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-svc-receive-handler-path2}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path2
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (mapup-queue(os-readyq(os-svc-receive-handler-path2(*os*)))
= deq(mapup-queue(os-readyq(*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path2.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path2
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (table('2, os-status-table(os-svc-receive-handler-path2(*os*)))
= putnth(cons('3, cons(os-srcid(*os*, '16), 'nil)),
array-qfirst(os-readyq(*os*)),
table('2, os-status-table(*os*))))

EVENT: Disable mapup-os-status-table-os-svc-receive-handler-path2.

THEOREM: tm-clock-os-svc-receive-handler-path2
tm-clock(os-svc-receive-handler-path2(*os*)) = '1000

EVENT: Disable tm-clock-os-svc-receive-handler-path2.

THEOREM: tm-iports-os-svc-receive-handler-path2
tm-iports(os-svc-receive-handler-path2(*os*)) = tm-iports(*os*)

EVENT: Disable tm-iports-os-svc-receive-handler-path2.

THEOREM: tm-oports-os-svc-receive-handler-path2
tm-oports(os-svc-receive-handler-path2(*os*)) = tm-oports(*os*)

EVENT: Disable tm-oports-os-svc-receive-handler-path2.

THEOREM: tm-rwstate-os-svc-receive-handler-path2
tm-rwstate(os-svc-receive-handler-path2(*os*)) = '0

EVENT: Disable tm-rwstate-os-svc-receive-handler-path2.

THEOREM: correctness-of-os-svc-receive-handler-path2
(good-os(*os*)
 \wedge ((\neg tm-waiting(*os*))
 \wedge ((\neg tm-errorp(*os*)))

$$\begin{aligned}
& os)), \\
& '3, \\
& \text{putnth} (\text{tm-pack-psw} (\text{tm-cc} (os), \\
& \qquad \qquad \qquad \text{tm-error} (os), \\
& \qquad \qquad \qquad '0, \\
& \qquad \qquad \qquad \text{tm-svcid} (os)), \\
& '8, \\
& \text{tm-cpu} (os)))
\end{aligned}$$

EVENT: Disable tm-cpu-os-svc-receive-handler-path3.

THEOREM: array-qfirst-os-readyq-os-svc-receive-handler-path3
(good-os (os)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge (\neg \text{array-qfullp} (\text{os-readyq} (os))))$
 $\rightarrow (\text{array-qfirst} (\text{os-readyq} (\text{os-svc-receive-handler-path3} (os)))$
 $= \text{array-qfirst} (\text{os-readyq} (os)))$

EVENT: Disable array-qfirst-os-readyq-os-svc-receive-handler-path3.

THEOREM: mapup-cpu-os-svc-receive-handler-path3-case1
(good-os (os)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge (\neg \text{array-qfullp} (\text{os-readyq} (os))))$
 $\rightarrow (\text{mapup-cpu} (\text{array-qfirst} (\text{os-readyq} (os)),$
 $\qquad \qquad \qquad \text{os-svc-receive-handler-path3} (os))$
 $= \text{putnth} (\text{array-qfirst} (\text{os-current-mbuffer} (\text{os-srcid} (os), '16),$
 $\qquad \qquad \qquad \text{os-current-taskid} (os),$
 $\qquad \qquad \qquad os)),$
 $'3,$
 $\text{putnth} (\text{tm-pack-psw} (\text{mapup-cc} (\text{array-qfirst} (\text{os-readyq} (os)),$
 $\qquad \qquad \qquad os),$
 $\qquad \qquad \qquad \text{mapup-error} (\text{array-qfirst} (\text{os-readyq} (os)),$
 $\qquad \qquad \qquad os),$
 $'0,$
 $\text{mapup-svcid} (\text{array-qfirst} (\text{os-readyq} (os)),$
 $\qquad \qquad \qquad os)),$
 $'8,$
 $\text{mapup-cpu} (\text{array-qfirst} (\text{os-readyq} (os)), os)))$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-receive-handler-path3-case2
(good-os (os)
 $\wedge ((\neg \text{tm-waiting} (os))$

$$\begin{aligned}
& \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
& \quad \wedge ((taskid \in \mathbf{N}) \\
& \quad \quad \wedge ((taskid < '16) \\
& \quad \quad \quad \wedge (taskid \neq \text{array-qfirst}(\text{os-readyq}(os))))))))) \\
\rightarrow & (\text{mapup-cpu}(taskid, \text{os-svc-receive-handler-path3}(os)) \\
& = \text{mapup-cpu}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-receive-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-cpu}(taskid, \text{os-svc-receive-handler-path3}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then putnth}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os, '16), \\
& \quad \quad \quad \text{os-current-taskid}(os), \\
& \quad \quad \quad \quad os)), \\
& \quad \quad '3, \\
& \quad \quad \text{putnth}(\text{tm-pack-psw}(\text{mapup-cc}(taskid, os), \\
& \quad \quad \quad \text{mapup-error}(taskid, os), \\
& \quad \quad \quad '0, \\
& \quad \quad \quad \text{mapup-svcid}(taskid, os)), \\
& \quad \quad '8, \\
& \quad \quad \text{mapup-cpu}(taskid, os))) \\
& \quad \text{else mapup-cpu}(taskid, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path3.

THEOREM: mapup-regs-os-svc-receive-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge ((\neg \text{array-qfullp}(\text{os-readyq}(os))) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-regs}(taskid, \text{os-svc-receive-handler-path3}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then putnth}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os, '16), \\
& \quad \quad \quad \text{os-current-taskid}(os), \\
& \quad \quad \quad \quad os)), \\
& \quad \quad '3, \\
& \quad \quad \text{mapup-regs}(taskid, os)) \\
& \quad \text{else mapup-regs}(taskid, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-receive-handler-path3.

THEOREM: mapup-cc-os-svc-receive-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qfullp (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-cc (*taskid*, os-svc-receive-handler-path3 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-receive-handler-path3.

THEOREM: mapup-error-os-svc-receive-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qfullp (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-error (*taskid*, os-svc-receive-handler-path3 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-receive-handler-path3.

THEOREM: mapup-svcflag-os-svc-receive-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qfullp (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcflag (*taskid*, os-svc-receive-handler-path3 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*)) **then** '0
 else mapup-svcflag (*taskid*, *os*) **endif**)

EVENT: Disable mapup-svcflag-os-svc-receive-handler-path3.

THEOREM: mapup-svcid-os-svc-receive-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qfullp (os-readyq (*os*)))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcid (*taskid*, os-svc-receive-handler-path3 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-receive-handler-path3.

THEOREM: mapup-task-os-svc-receive-handler-path3

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
      ∧ ((¬ array-qfullp (os-readyq (os)))
          ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))))
→ (mapup-task (taskid, os-svc-receive-handler-path3 (os))
    = if taskid = array-qfirst (os-readyq (os))
      then os-store-message (array-qfirst (os-current-mbuffer (os-srcid (os,
                                                                    '16),
                                                                    os-current-taskid (os),
                                                                    os)),
                            tm-set-svcflag ('0,
                                            mapup-task (taskid, os)))
      else mapup-task (taskid, os) endif)

```

EVENT: Disable mapup-task-os-svc-receive-handler-path3.

THEOREM: mapup-tasks-os-svc-receive-handler-path3-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
      ∧ ((¬ array-qfullp (os-readyq (os)))
          ∧ ((taskid ∈ ℕ)
              ∧ (array-qfirst (os-readyq (os)) < taskid))))))
→ (mapup-tasks (taskid, os-svc-receive-handler-path3 (os))
    = mapup-tasks (taskid, os))

```

EVENT: Disable mapup-tasks-os-svc-receive-handler-path3-case1.

THEOREM: mapup-tasks-os-svc-receive-handler-path3-case2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
      ∧ ((¬ array-qfullp (os-readyq (os)))
          ∧ ((taskid ∈ ℕ)
              ∧ if array-qfirst (os-readyq (os)) < taskid
                then '*1*false
                else '*1>true endif))))))
→ (mapup-tasks (taskid, os-svc-receive-handler-path3 (os))
    = putnth (os-store-message (array-qfirst (os-current-mbuffer (os-srcid (os,
                                                                    '16),
                                                                    os-current-taskid (os),
                                                                    os)),
                            tm-set-svcflag ('0,
                                            getnth (array-qfirst (os-readyq (os))

```

$$\begin{aligned}
& \text{array-qfirst}(\text{os-readyq}(os) - \text{taskid}, \\
& \text{mapup-tasks}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-receive-handler-path3-case2.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path3-support

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{array-qfullp}(\text{os-readyq}(os)))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-svc-receive-handler-path3}(os)) \\
& = \text{putnth}(\text{os-store-message}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os), \\
& \hspace{15em} '16), \\
& \hspace{15em} \text{os-current-taskid}(os), \\
& \hspace{15em} os)), \\
& \hspace{10em} \text{tm-set-svcflag}('0, \\
& \hspace{15em} \text{getnth}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \hspace{15em} \text{mapup-os-tasks}(os))), \\
& \text{array-qfirst}(\text{os-readyq}(os)), \\
& \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path3-support.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{getnth}('2 * \text{os-srcid}(os), '16), \text{os-status-table}(os)) \\
& \quad = '2))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-svc-receive-handler-path3}(os)) \\
& = \text{putnth}(\text{os-store-message}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os), \\
& \hspace{15em} '16), \\
& \hspace{15em} \text{os-current-taskid}(os), \\
& \hspace{15em} os)), \\
& \hspace{10em} \text{tm-set-svcflag}('0, \\
& \hspace{15em} \text{getnth}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \\
& \hspace{15em} \text{mapup-os-tasks}(os))), \\
& \text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \\
& \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path3.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path3

$\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-ibuffers } (\text{os-svc-receive-handler-path3 } (os)))$
 $= \text{mapup-os-ibuffers } (os)$

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path3.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path3
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-svc-receive-handler-path3 } (os)))$
 $= \text{mapup-os-obuffers } (os)$

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path3.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path3-support
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge (\neg \text{array-qempty } (\text{os-current-mbuffer } (\text{os-srcid } (os, '16),$
 $\quad \quad \quad \text{os-current-taskid } (os),$
 $\quad \quad \quad os))))))$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-svc-receive-handler-path3 } (os)))$
 $= \text{deq2 } (\text{os-srcid } (os, '16),$
 $\quad \text{array-qfirst } (\text{os-readyq } (os)),$
 $\quad \text{mapup-os-mbuffers } (os))$

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path3-support.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path3
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge (\neg \text{array-qempty } (\text{os-current-mbuffer } (\text{os-srcid } (os, '16),$
 $\quad \quad \quad \text{os-current-taskid } (os),$
 $\quad \quad \quad os))))))$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-svc-receive-handler-path3 } (os)))$
 $= \text{deq2 } (\text{os-srcid } (os, '16),$
 $\quad \text{qfirst } (\text{mapup-queue } (\text{os-readyq } (os))),$
 $\quad \text{mapup-os-mbuffers } (os))$

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path3.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path3
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge (\text{getnth } ('2 * \text{os-srcid } (os, '16), \text{os-status-table } (os)))$

$$\begin{aligned} &= \text{'2})) \\ \rightarrow & (\text{mapup-queue}(\text{os-readyq}(\text{os-svc-receive-handler-path3}(os))) \\ &= \text{enq}(\text{os-srcid}(os, \text{'16}), \text{mapup-queue}(\text{os-readyq}(os)))) \end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path3.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path3
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{table}(\text{'2}, \text{os-status-table}(\text{os-svc-receive-handler-path3}(os)))$
 $= \text{putnth}(\text{'0 0},$
 $\text{os-srcid}(os, \text{'16}),$
 $\text{table}(\text{'2}, \text{os-status-table}(os)))$

EVENT: Disable mapup-os-status-table-os-svc-receive-handler-path3.

THEOREM: tm-clock-os-svc-receive-handler-path3
 $\text{tm-clock}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-clock}(os)$

EVENT: Disable tm-clock-os-svc-receive-handler-path3.

THEOREM: tm-iports-os-svc-receive-handler-path3
 $\text{tm-iports}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-svc-receive-handler-path3.

THEOREM: tm-oports-os-svc-receive-handler-path3
 $\text{tm-oports}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-svc-receive-handler-path3.

THEOREM: tm-rwstate-os-svc-receive-handler-path3
 $(\text{good-tm}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-rwstate}(\text{os-svc-receive-handler-path3}(os)) = \text{tm-rwstate}(os))$

EVENT: Disable tm-rwstate-os-svc-receive-handler-path3.

THEOREM: ak-waiting-to-sendp-mapup-os
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge ((\text{getnth}(\text{'2 * os-srcid}(os, \text{'16}),$
 $\text{os-status-table}(os)))$

→ (mapup-base (*taskid*, os-svc-receive-handler-path4 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-receive-handler-path4.

THEOREM: mapup-limit-os-svc-receive-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-receive-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-receive-handler-path4.

THEOREM: getseg-tm-memory-os-svc-receive-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-receive-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-receive-handler-path4.

THEOREM: tm-cpu-os-svc-receive-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-cpu (os-svc-receive-handler-path4 (*os*))
= putnth (array-qfirst (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)),
'3,
putnth (tm-pack-psw (tm-cc (*os*),
tm-error (*os*),
'0,
tm-svcid (*os*)),
'8,
tm-cpu (*os*))))

EVENT: Disable tm-cpu-os-svc-receive-handler-path4.

THEOREM: mapup-cpu-os-svc-receive-handler-path4-case1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-cpu (array-qfirst (os-readyq (*os*)),
os-svc-receive-handler-path4 (*os*)))

$$\begin{aligned}
= & \text{putnth}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os, '16), \\
& \qquad \qquad \qquad \text{os-current-taskid}(os), \\
& \qquad \qquad \qquad os)), \\
& '3, \\
& \text{putnth}(\text{tm-pack-psw}(\text{mapup-cc}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \qquad \qquad \qquad os), \\
& \qquad \qquad \qquad \text{mapup-error}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \qquad \qquad \qquad os), \\
& '0, \\
& \qquad \qquad \text{mapup-svcid}(\text{array-qfirst}(\text{os-readyq}(os)), \\
& \qquad \qquad \qquad os)), \\
& '8, \\
& \text{mapup-cpu}(\text{array-qfirst}(\text{os-readyq}(os)), os)))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path4-case1.

THEOREM: mapup-cpu-os-svc-receive-handler-path4-case2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \wedge ((\text{taskid} < '16) \\
& \quad \quad \quad \wedge (\text{taskid} \neq \text{array-qfirst}(\text{os-readyq}(os))))))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-svc-receive-handler-path4}(os)) \\
& = \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path4-case2.

THEOREM: mapup-cpu-os-svc-receive-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-svc-receive-handler-path4}(os)) \\
& = \text{if } \text{taskid} = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then } \text{putnth}(\text{array-qfirst}(\text{os-current-mbuffer}(\text{os-srcid}(os, '16), \\
& \qquad \qquad \qquad \text{os-current-taskid}(os), \\
& \qquad \qquad \qquad os)), \\
& \quad '3, \\
& \quad \text{putnth}(\text{tm-pack-psw}(\text{mapup-cc}(\text{taskid}, os), \\
& \qquad \qquad \qquad \text{mapup-error}(\text{taskid}, os), \\
& \qquad \qquad \qquad '0, \\
& \qquad \qquad \qquad \text{mapup-svcid}(\text{taskid}, os)), \\
& \quad '8, \\
& \quad \text{mapup-cpu}(\text{taskid}, os)) \\
& \text{else } \text{mapup-cpu}(\text{taskid}, os) \text{ endif}
\end{aligned}$$

$$\begin{aligned}
& \text{tm-set-svcflag} ('0, \\
& \quad \text{getnth} (\text{array-qfirst} (\text{os-readyq} (os)) \\
& \quad \quad - \text{taskid}, \\
& \quad \quad \text{mapup-tasks} (\text{taskid}, \\
& \quad \quad \quad os))), \\
& \text{array-qfirst} (\text{os-readyq} (os)) - \text{taskid}, \\
& \text{mapup-tasks} (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-receive-handler-path4-case2.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path4-support
(good-os (os) \wedge (\neg tm-waiting (os)))

$$\begin{aligned}
\rightarrow & (\text{mapup-os-tasks} (\text{os-svc-receive-handler-path4} (os)) \\
& = \text{putnth} (\text{os-store-message} (\text{array-qfirst} (\text{os-current-mbuffer} (\text{os-srcid} (os), \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \text{os-current-taskid} (os), \\
& \quad \quad \quad os)),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-set-svcflag} ('0, \\
& \quad \text{getnth} (\text{array-qfirst} (\text{os-readyq} (os)), \\
& \quad \quad \text{mapup-os-tasks} (os))), \\
& \text{array-qfirst} (\text{os-readyq} (os)), \\
& \text{mapup-os-tasks} (os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path4-support.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path4
(good-os (os) \wedge (\neg tm-waiting (os)))

$$\begin{aligned}
\rightarrow & (\text{mapup-os-tasks} (\text{os-svc-receive-handler-path4} (os)) \\
& = \text{putnth} (\text{os-store-message} (\text{array-qfirst} (\text{os-current-mbuffer} (\text{os-srcid} (os), \\
& \quad \quad \quad '16), \\
& \quad \quad \quad \text{os-current-taskid} (os), \\
& \quad \quad \quad os)),
\end{aligned}$$

$$\begin{aligned}
& \text{tm-set-svcflag} ('0, \\
& \quad \text{getnth} (\text{qfirst} (\text{mapup-queue} (\text{os-readyq} (os))), \\
& \quad \quad \text{mapup-os-tasks} (os))), \\
& \text{qfirst} (\text{mapup-queue} (\text{os-readyq} (os))), \\
& \text{mapup-os-tasks} (os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path4.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path4
good-os (os)

$$\begin{aligned}
\rightarrow & (\text{mapup-os-ibuffers} (\text{os-svc-receive-handler-path4} (os)) \\
& = \text{mapup-os-ibuffers} (os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path4.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path4
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-svc-receive-handler-path4 } (os))$
 $\quad = \text{mapup-os-obuffers } (os))$

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path4.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path4-support
 $(\text{good-os } (os)$
 $\quad \wedge ((\neg \text{tm-waiting } (os))$
 $\quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-mbuffer } (\text{os-srcid } (os, '16),$
 $\quad \quad \quad \text{os-current-taskid } (os),$
 $\quad \quad \quad \text{os}))))))$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-svc-receive-handler-path4 } (os))$
 $\quad = \text{deq2 } (\text{os-srcid } (os, '16),$
 $\quad \quad \text{array-qfirst } (\text{os-readyq } (os)),$
 $\quad \quad \text{mapup-os-mbuffers } (os))$

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path4-support.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path4
 $(\text{good-os } (os)$
 $\quad \wedge ((\neg \text{tm-waiting } (os))$
 $\quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-mbuffer } (\text{os-srcid } (os, '16),$
 $\quad \quad \quad \text{os-current-taskid } (os),$
 $\quad \quad \quad \text{os}))))))$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-svc-receive-handler-path4 } (os))$
 $\quad = \text{deq2 } (\text{os-srcid } (os, '16),$
 $\quad \quad \text{qfirst } (\text{mapup-queue } (\text{os-readyq } (os))),$
 $\quad \quad \text{mapup-os-mbuffers } (os))$

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path4.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path4
 $(\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os)))$
 $\rightarrow (\text{mapup-queue } (\text{os-readyq } (\text{os-svc-receive-handler-path4 } (os)))$
 $\quad = \text{mapup-queue } (\text{os-readyq } (os)))$

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path4.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{table}('2, \text{os-status-table}(\text{os-svc-receive-handler-path4}(os))) \\
& = \text{table}('2, \text{os-status-table}(os)))
\end{aligned}$$

EVENT: Disable mapup-os-status-table-os-svc-receive-handler-path4.

THEOREM: tm-clock-os-svc-receive-handler-path4
 $\text{tm-clock}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-clock}(os)$

EVENT: Disable tm-clock-os-svc-receive-handler-path4.

THEOREM: tm-iports-os-svc-receive-handler-path4
 $\text{tm-iports}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-svc-receive-handler-path4.

THEOREM: tm-oports-os-svc-receive-handler-path4
 $\text{tm-oports}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-svc-receive-handler-path4.

THEOREM: tm-rwstate-os-svc-receive-handler-path4
 $(\text{good-tm}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-rwstate}(\text{os-svc-receive-handler-path4}(os)) = \text{tm-rwstate}(os))$

EVENT: Disable tm-rwstate-os-svc-receive-handler-path4.

THEOREM: not-ak-waiting-to-sendp-mapup-os
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\neg \text{tm-errorp}(os))$
 $\wedge (\neg ((\text{getnth}('2 * \text{os-srcid}(os, '16),$
 $\text{os-status-table}(os))$
 $= '2)$
 $\wedge (\text{getnth}('1$
 $+ ('2$
 $* \text{os-srcid}(os, '16),$
 $\text{os-status-table}(os))$
 $= \text{os-current-taskid}(os))))))$
 $\rightarrow (\neg \text{ak-waiting-to-sendp}(\text{os-srcid}(os, '16),$
 $\text{os-current-taskid}(os),$
 $\text{mapup-os}(os)))$

$$= \text{mapup-base}(taskid, os)$$

EVENT: Disable mapup-base-os-svc-tyi-handler-path1.

THEOREM: mapup-limit-os-svc-tyi-handler-path1
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-svc-tyi-handler-path1}(os)))$
 $= \text{mapup-limit}(taskid, os)$

EVENT: Disable mapup-limit-os-svc-tyi-handler-path1.

THEOREM: getseg-tm-memory-os-svc-tyi-handler-path1
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(\text{os-svc-tyi-handler-path1}(os)))$
 $= \text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(os))$

EVENT: Disable getseg-tm-memory-os-svc-tyi-handler-path1.

THEOREM: mapup-cpu-os-svc-tyi-handler-path1
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu}(taskid, \text{os-svc-tyi-handler-path1}(os)))$
 $= \text{mapup-cpu}(taskid, os)$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path1.

THEOREM: mapup-regs-os-svc-tyi-handler-path1
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs}(taskid, \text{os-svc-tyi-handler-path1}(os)))$
 $= \text{mapup-regs}(taskid, os)$

EVENT: Disable mapup-regs-os-svc-tyi-handler-path1.

THEOREM: mapup-cc-os-svc-tyi-handler-path1
 $(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cc}(taskid, \text{os-svc-tyi-handler-path1}(os)))$
 $= \text{mapup-cc}(taskid, os)$

EVENT: Disable mapup-cc-os-svc-tyi-handler-path1.

THEOREM: mapup-error-os-svc-tyi-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$)
→ (mapup-error (*taskid*, os-svc-tyi-handler-path1 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-tyi-handler-path1.

THEOREM: mapup-svcflag-os-svc-tyi-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$)
→ (mapup-svcflag (*taskid*, os-svc-tyi-handler-path1 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-svc-tyi-handler-path1.

THEOREM: mapup-svcid-os-svc-tyi-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$)
→ (mapup-svcid (*taskid*, os-svc-tyi-handler-path1 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-tyi-handler-path1.

THEOREM: mapup-task-os-svc-tyi-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$)
→ (mapup-task (*taskid*, os-svc-tyi-handler-path1 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-svc-tyi-handler-path1.

THEOREM: mapup-tasks-os-svc-tyi-handler-path1
(good-os (*os*) $\wedge ((\neg \text{tm-waiting} (os)) \wedge (taskid \in \mathbf{N}))$)
→ (mapup-tasks (*taskid*, os-svc-tyi-handler-path1 (*os*))
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-svc-tyi-handler-path1.

THEOREM: mapup-os-tasks-os-svc-tyi-handler-path1

$(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{mapup-os-tasks}(\text{os-svc-tyi-handler-path1}(os)) = \text{mapup-os-tasks}(os))$

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path1.

THEOREM: mapup-os-ibuffers-os-svc-tyi-handler-path1
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-os-ibuffers}(\text{os-svc-tyi-handler-path1}(os))$
 $= \text{mapup-os-ibuffers}(os))$

EVENT: Disable mapup-os-ibuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-os-obuffers-os-svc-tyi-handler-path1
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-os-obuffers}(\text{os-svc-tyi-handler-path1}(os))$
 $= \text{mapup-os-obuffers}(os))$

EVENT: Disable mapup-os-obuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-os-mbuffers-os-svc-tyi-handler-path1
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-os-mbuffers}(\text{os-svc-tyi-handler-path1}(os))$
 $= \text{mapup-os-mbuffers}(os))$

EVENT: Disable mapup-os-mbuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-queue-os-readyq-os-svc-tyi-handler-path1
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{mapup-queue}(\text{os-readyq}(\text{os-svc-tyi-handler-path1}(os)))$
 $= \text{deq}(\text{mapup-queue}(\text{os-readyq}(os))))$

EVENT: Disable mapup-queue-os-readyq-os-svc-tyi-handler-path1.

THEOREM: mapup-os-status-table-os-svc-tyi-handler-path1
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{table}('2, \text{os-status-table}(\text{os-svc-tyi-handler-path1}(os)))$
 $= \text{putnth}('5\ 0,$
 $\quad \text{array-qfirst}(\text{os-readyq}(os)),$
 $\quad \text{table}('2, \text{os-status-table}(os)))$

EVENT: Disable mapup-os-status-table-os-svc-tyi-handler-path1.

EVENT: Disable mapup-limit-os-svc-tyi-handler-path2.

THEOREM: getseg-tm-memory-os-svc-tyi-handler-path2
(good-os(*os*) \wedge (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (os-svc-tyi-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-tyi-handler-path2.

```
(PROVE-LEMMA
TM-CPU-OS-SVC-TYI-HANDLER-PATH2-CROCK
(REWRITE)
(IMPLIES
(AND (GOOD-OS OS)
      (NOT (TM-WAITING OS))))
(EQUAL
(TM-CPU (OS-SVC-TYI-HANDLER-PATH2 OS))
(CONS
(GETNTH '0
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '1
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '2
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '3
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
(CONS
(GETNTH '4
(OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
OS))
```

```

(CONS
  (GETNTH '5
    (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
      OS))
  (CONS
    (GETNTH '6
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS))
    (CONS
      (GETNTH '7
        (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS))
      (CONS
        (TM-PACK-PSW
          (OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS)
          (OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
            OS))
        'NIL)))))))))
((ENABLE OS-SVC-TYI-HANDLER-PATH2 TM-CPU TM-RO TM-R1 TM-R2 TM-R3 TM-R4 TM-R5
  TM-R6 TM-R7)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-svc-tyi-handler-path2-crock.

THEOREM: tm-cpu-os-svc-tyi-handler-path2-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ (array-qfirst (array-deq (os-readyq (os)))
      = array-qfirst (os-readyq (os))))
→ (tm-cpu (os-svc-tyi-handler-path2 (os))
  = getnth (os-current-taskid (os-svc-tyi-handler-path2 (os)),
    table ('9, os-task-table (os-svc-tyi-handler-path2 (os))))))

```

THEOREM: tm-cpu-os-svc-tyi-handler-path2-case2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
    ∧ ((¬ array-qemptyp (array-deq (os-readyq (os))))
      ∧ (array-qfirst (array-deq (os-readyq (os)))

```


$$\begin{aligned}
& \neq \text{array-qfirst}(\text{os-readyq}(os))) \\
\rightarrow & (\text{tm-cpu}(\text{os-svc-tyi-handler-path2}(os)) \\
& = \text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path2}(os)), \\
& \quad \text{table}('9, \text{os-task-table}(\text{os-svc-tyi-handler-path2}(os))))))
\end{aligned}$$

THEOREM: tm-cpu-os-svc-tyi-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))))) \\
\rightarrow & (\text{tm-cpu}(\text{os-svc-tyi-handler-path2}(os)) \\
& = \text{getnth}(\text{os-current-taskid}(\text{os-svc-tyi-handler-path2}(os)), \\
& \quad \text{table}('9, \text{os-task-table}(\text{os-svc-tyi-handler-path2}(os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-svc-tyi-handler-path2.

THEOREM: mapup-cpu-os-svc-tyi-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path2.

THEOREM: mapup-regs-os-svc-tyi-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-regs}(\text{taskid}, \text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-regs}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-tyi-handler-path2.

THEOREM: mapup-cc-os-svc-tyi-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cc}(\text{taskid}, \text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-cc}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-tyi-handler-path2.

THEOREM: mapup-error-os-svc-tyi-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-svc-tyi-handler-path2 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-tyi-handler-path2.

THEOREM: mapup-svcflag-os-svc-tyi-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcflag (*taskid*, os-svc-tyi-handler-path2 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-svc-tyi-handler-path2.

THEOREM: mapup-svcid-os-svc-tyi-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcid (*taskid*, os-svc-tyi-handler-path2 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-tyi-handler-path2.

THEOREM: mapup-task-os-svc-tyi-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qempty (array-deq (os-readyq (*os*))))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-task (*taskid*, os-svc-tyi-handler-path2 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-svc-tyi-handler-path2.

THEOREM: mapup-tasks-os-svc-tyi-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge ((\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os)))) \\
& \quad \wedge (taskid \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-tyi-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-svc-tyi-handler-path2} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{array-deq}(\text{os-readyq}(os))))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-svc-tyi-handler-path2}(os)) = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-ibuffers-os-svc-tyi-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-svc-tyi-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-obuffers-os-svc-tyi-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-svc-tyi-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-os-mbuffers-os-svc-tyi-handler-path2} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-svc-tyi-handler-path2}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-svc-tyi-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-queue-os-readyq-os-svc-tyi-handler-path2} \\
& (\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-queue}(\text{os-readyq}(\text{os-svc-tyi-handler-path2}(os))) \\
& = \text{deq}(\text{mapup-queue}(\text{os-readyq}(os))))
\end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-svc-tyi-handler-path2.

THEOREM: mapup-base-os-svc-tyi-handler-path3
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-tyi-handler-path3 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-tyi-handler-path3.

THEOREM: mapup-limit-os-svc-tyi-handler-path3
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-tyi-handler-path3 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyi-handler-path3.

THEOREM: getseg-tm-memory-os-svc-tyi-handler-path3
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-tyi-handler-path3.

THEOREM: tm-cpu-os-svc-tyi-handler-path3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-cpu (os-svc-tyi-handler-path3 (*os*))
= putnth (array-qfirst (os-current-ibuffer (os-current-taskid (*os*),
os)),
'3,
putnth (tm-pack-psw (tm-cc (*os*),
tm-error (*os*),
'0,
tm-svcid (*os*)),
'8,
tm-cpu (*os*))))

EVENT: Disable tm-cpu-os-svc-tyi-handler-path3.

THEOREM: mapup-cpu-os-svc-tyi-handler-path3-case1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-cpu (array-qfirst (os-readyq (*os*)), os-svc-tyi-handler-path3 (*os*))

$$\begin{aligned}
= & \text{putnth}(\text{array-qfirst}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), \\
& \hspace{15em} os)), \\
& \quad '3, \\
& \quad \text{putnth}(\text{tm-pack-psw}(\text{mapup-cc}(\text{array-qfirst}(\text{os-readyq}(os), \\
& \hspace{15em} os), \\
& \hspace{15em} \text{mapup-error}(\text{array-qfirst}(\text{os-readyq}(os), \\
& \hspace{15em} os), \\
& \hspace{15em} '0, \\
& \hspace{15em} \text{mapup-svcid}(\text{array-qfirst}(\text{os-readyq}(os), \\
& \hspace{15em} os))), \\
& \quad '8, \\
& \quad \text{mapup-cpu}(\text{array-qfirst}(\text{os-readyq}(os), os))))))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-tyi-handler-path3-case2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \wedge ((\text{taskid} < '16) \\
& \quad \quad \quad \wedge (\text{taskid} \neq \text{array-qfirst}(\text{os-readyq}(os))))))))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-svc-tyi-handler-path3}(os)) \\
& = \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-tyi-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-svc-tyi-handler-path3}(os)) \\
& = \text{if } \text{taskid} = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then } \text{putnth}(\text{array-qfirst}(\text{os-current-ibuffer}(\text{os-current-taskid}(os), \\
& \hspace{15em} os)), \\
& \quad \quad '3, \\
& \quad \quad \text{putnth}(\text{tm-pack-psw}(\text{mapup-cc}(\text{taskid}, os), \\
& \hspace{15em} \text{mapup-error}(\text{taskid}, os), \\
& \hspace{15em} '0, \\
& \hspace{15em} \text{mapup-svcid}(\text{taskid}, os))), \\
& \quad \quad '8, \\
& \quad \quad \text{mapup-cpu}(\text{taskid}, os))) \\
& \quad \text{else } \text{mapup-cpu}(\text{taskid}, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path3.

EVENT: Disable mapup-svcid-os-svc-tyi-handler-path3.

THEOREM: mapup-task-os-svc-tyi-handler-path3

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-task (taskid, os-svc-tyi-handler-path3 (os))
   = if taskid = array-qfirst (os-readyq (os))
     then os-store-message (array-qfirst (os-current-ibuffer (os-current-taskid (os),
                                                                os)),
                           tm-set-svcflag ('0,
                                           mapup-task (taskid, os)))
     else mapup-task (taskid, os) endif)

```

EVENT: Disable mapup-task-os-svc-tyi-handler-path3.

THEOREM: mapup-tasks-os-svc-tyi-handler-path3-case1

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
     ∧ ((taskid ∈ ℕ) ∧ (array-qfirst (os-readyq (os)) < taskid))))
→ (mapup-tasks (taskid, os-svc-tyi-handler-path3 (os))
   = mapup-tasks (taskid, os))

```

EVENT: Disable mapup-tasks-os-svc-tyi-handler-path3-case1.

THEOREM: mapup-tasks-os-svc-tyi-handler-path3-case2

```

(good-os (os)
  ∧ ((¬ tm-waiting (os))
     ∧ ((taskid ∈ ℕ)
        ∧ if array-qfirst (os-readyq (os)) < taskid
          then '*1*false
          else '*1*true endif)))
→ (mapup-tasks (taskid, os-svc-tyi-handler-path3 (os))
   = putnth (os-store-message (array-qfirst (os-current-ibuffer (os-current-taskid (os),
                                                                os)),
                           tm-set-svcflag ('0,
                                           getnth (array-qfirst (os-readyq (os))
                                                  - taskid,
                                                  mapup-tasks (taskid,
                                                                os))),
                           array-qfirst (os-readyq (os)) - taskid,
                           mapup-tasks (taskid, os)))

```

EVENT: Disable mapup-tasks-os-svc-tyi-handler-path3-case2.

THEOREM: mapup-os-tasks-os-svc-tyi-handler-path3-support
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-tyi-handler-path3 (*os*))
 = putnth (os-store-message (array-qfirst (os-current-ibuffer (os-current-taskid (*os*),
os)),
 tm-set-svcflag ('0,
 getnth (array-qfirst (os-readyq (*os*)),
 mapup-os-tasks (*os*))),
 array-qfirst (os-readyq (*os*)),
 mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path3-support.

THEOREM: mapup-os-tasks-os-svc-tyi-handler-path3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-tyi-handler-path3 (*os*))
 = putnth (os-store-message (array-qfirst (os-current-ibuffer (os-current-taskid (*os*),
os)),
 tm-set-svcflag ('0,
 getnth (qfirst (mapup-queue (os-readyq (*os*)),
 mapup-os-tasks (*os*))),
 qfirst (mapup-queue (os-readyq (*os*))),
 mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path3.

THEOREM: mapup-os-ibuffers-os-svc-tyi-handler-path3
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qempty (os-current-ibuffer (os-current-taskid (*os*),
os))))))
 \rightarrow (mapup-os-ibuffers (os-svc-tyi-handler-path3 (*os*))
 = deq-ith-buffer (os-current-taskid (*os*), mapup-os-ibuffers (*os*)))

EVENT: Disable mapup-os-ibuffers-os-svc-tyi-handler-path3.

THEOREM: mapup-os-obuffers-os-svc-tyi-handler-path3
 good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-svc-tyi-handler-path3 (*os*))
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-tyi-handler-path3.

THEOREM: mapup-os-mbuffers-os-svc-tyi-handler-path3
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-tyi-handler-path3 (*os*)))
 $=$ mapup-os-mbuffers (*os*)

EVENT: Disable mapup-os-mbuffers-os-svc-tyi-handler-path3.

THEOREM: mapup-queue-os-readyq-os-svc-tyi-handler-path3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-tyi-handler-path3 (*os*))))
 $=$ mapup-queue (os-readyq (*os*)))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyi-handler-path3.

THEOREM: mapup-os-status-table-os-svc-tyi-handler-path3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-tyi-handler-path3 (*os*))))
 $=$ table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-svc-tyi-handler-path3.

THEOREM: tm-clock-os-svc-tyi-handler-path3
 tm-clock (os-svc-tyi-handler-path3 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-tyi-handler-path3.

THEOREM: tm-iports-os-svc-tyi-handler-path3
 tm-iports (os-svc-tyi-handler-path3 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyi-handler-path3.

THEOREM: tm-oports-os-svc-tyi-handler-path3
 tm-oports (os-svc-tyi-handler-path3 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-tyi-handler-path3.

THEOREM: tm-rwstate-os-svc-tyi-handler-path3
 (good-tm (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-rwstate (os-svc-tyi-handler-path3 (*os*))) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-tyi-handler-path3.

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path1.

THEOREM: mapup-cpu-os-svc-tyo-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
→ (mapup-cpu (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path1.

THEOREM: mapup-regs-os-svc-tyo-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
→ (mapup-regs (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-tyo-handler-path1.

THEOREM: mapup-cc-os-svc-tyo-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
→ (mapup-cc (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-tyo-handler-path1.

THEOREM: mapup-error-os-svc-tyo-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
→ (mapup-error (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-tyo-handler-path1.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path1
(good-os (*os*)
 $\wedge ((\neg \text{tm-waiting} (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
→ (mapup-svcflag (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path1.

THEOREM: mapup-svcid-os-svc-tyo-handler-path1
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-svcid } (taskid, \text{os-svc-tyo-handler-path1 } (os))$
 $= \text{mapup-svcid } (taskid, os))$

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path1.

THEOREM: mapup-task-os-svc-tyo-handler-path1
 (good-os (*os*)
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-task } (taskid, \text{os-svc-tyo-handler-path1 } (os))$
 $= \text{mapup-task } (taskid, os))$

EVENT: Disable mapup-task-os-svc-tyo-handler-path1.

THEOREM: mapup-tasks-os-svc-tyo-handler-path1
 (good-os (*os*) $\wedge ((\neg \text{tm-waiting } (os)) \wedge (taskid \in \mathbf{N}))$)
 $\rightarrow (\text{mapup-tasks } (taskid, \text{os-svc-tyo-handler-path1 } (os))$
 $= \text{mapup-tasks } (taskid, os))$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path1.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path1
 (good-os (*os*) $\wedge (\neg \text{tm-waiting } (os))$)
 $\rightarrow (\text{mapup-os-tasks } (\text{os-svc-tyo-handler-path1 } (os)) = \text{mapup-os-tasks } (os))$

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path1.

THEOREM: mapup-os-ibuffers-os-svc-tyo-handler-path1
 good-os (*os*)
 $\rightarrow (\text{mapup-os-ibuffers } (\text{os-svc-tyo-handler-path1 } (os))$
 $= \text{mapup-os-ibuffers } (os))$

EVENT: Disable mapup-os-ibuffers-os-svc-tyo-handler-path1.

THEOREM: mapup-os-obuffers-os-svc-tyo-handler-path1
 good-os (*os*)
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-svc-tyo-handler-path1 } (os))$
 $= \text{mapup-os-obuffers } (os))$

EVENT: Disable mapup-os-obuffers-os-svc-tyo-handler-path1.

THEOREM: mapup-os-mbuffers-os-svc-tyo-handler-path1
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-tyo-handler-path1 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-tyo-handler-path1.

THEOREM: mapup-queue-os-readyq-os-svc-tyo-handler-path1
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-tyo-handler-path1 (*os*)))
 = deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyo-handler-path1.

THEOREM: mapup-os-status-table-os-svc-tyo-handler-path1
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-tyo-handler-path1 (*os*)))
 = putnth ('(4 0),
 array-qfirst (os-readyq (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-svc-tyo-handler-path1.

THEOREM: tm-rwstate-os-svc-tyo-handler-path1
 tm-rwstate (os-svc-tyo-handler-path1 (*os*)) = '1

EVENT: Disable tm-rwstate-os-svc-tyo-handler-path1.

THEOREM: tm-clock-os-svc-tyo-handler-path1
 tm-clock (os-svc-tyo-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-tyo-handler-path1.

THEOREM: tm-iports-os-svc-tyo-handler-path1
 tm-iports (os-svc-tyo-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyo-handler-path1.

THEOREM: tm-oports-os-svc-tyo-handler-path1
 tm-oports (os-svc-tyo-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-tyo-handler-path1.

THEOREM: mapup-queue-getnth-for-obuffers
 (good-os (*os*) \wedge (*id* < '16))
 \rightarrow (mapup-queue (getnth (*id*, table ('8, os-obuffers (*os*))))
 = getnth (*id*, mapup-os-obuffers (*os*)))

EVENT: Disable mapup-queue-getnth-for-obuffers.

THEOREM: correctness-of-array-qfullp-obuffers-instance
 (array-queuep (*queue*) \wedge ('4 = getnth ('3, *queue*)))
 \rightarrow (array-qfullp (*queue*) = qfullp (mapup-queue (*queue*), '4))

EVENT: Disable correctness-of-array-qfullp-obuffers-instance.

THEOREM: correctness-of-os-svc-tyo-handler-path1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge ((\neg tm-errorp (*os*)
 \wedge (((tm-svcid (*os*) **mod** '4) = '2)
 \wedge (array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))
 \wedge array-qempty (array-deq (os-readyq (*os*))))))))))
 \rightarrow (mapup-os (os-svc-tyo-handler-path1 (*os*)
 = ak-execute-output (os-message (*os*),
 os-current-taskid (*os*),
 mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyo-handler-path1.

THEOREM: mapup-base-os-svc-tyo-handler-path2
 good-os (*os*)
 \rightarrow (mapup-base (*taskid*, os-svc-tyo-handler-path2 (*os*)
 = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-tyo-handler-path2.

THEOREM: mapup-limit-os-svc-tyo-handler-path2
 good-os (*os*)
 \rightarrow (mapup-limit (*taskid*, os-svc-tyo-handler-path2 (*os*)
 = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyo-handler-path2.

THEOREM: getseg-tm-memory-os-svc-tyo-handler-path2

```

(good-os(os) ∧ (taskid < '16))
→ (getseg (mapup-base (taskid, os),
    mapup-limit (taskid, os),
    tm-memory (os-svc-tyo-handler-path2 (os)))
= getseg (mapup-base (taskid, os),
    mapup-limit (taskid, os),
    tm-memory (os))

```

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path2.

```

(PROVE-LEMMA
TM-CPU-OS-SVC-TYO-HANDLER-PATH2-CROCK
(REWRITE)
(IMPLIES
(AND (GOOD-OS OS)
      (NOT (TM-WAITING OS))))
(EQUAL
(TM-CPU (OS-SVC-TYO-HANDLER-PATH2 OS))
(CONS
(GETNTH '0
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))
(CONS
(GETNTH '1
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))
(CONS
(GETNTH '2
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))
(CONS
(GETNTH '3
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))
(CONS
(GETNTH '4
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))
(CONS
(GETNTH '5
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
                  OS))

```



```

(CONS
  (GETNTH '6
    (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
      OS))
  (CONS
    (GETNTH '7
      (OS-NEW-REGS (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
        OS))
    (CONS
      (TM-PACK-PSW
        (OS-NEW-CC (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS)
        (OS-NEW-ERROR (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS)
        (OS-NEW-SVCFLAG (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS)
        (OS-NEW-SVCID (ARRAY-QFIRST (ARRAY-DEQ (OS-READYQ OS)))
          OS))
      'NIL)))))))))
((ENABLE OS-SVC-TYO-HANDLER-PATH2 TM-CPU TM-RO TM-R1 TM-R2 TM-R3 TM-R4 TM-R5
  TM-R6 TM-R7)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-svc-tyo-handler-path2-crock.

THEOREM: tm-cpu-os-svc-tyo-handler-path2-case1
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (array-qfirst (array-deq (os-readyq (*os*)))
 $=$ array-qfirst (os-readyq (*os*))))))
 \rightarrow (tm-cpu (os-svc-tyo-handler-path2 (*os*))
 $=$ getnth (os-current-taskid (os-svc-tyo-handler-path2 (*os*)),
 table ('9, os-task-table (os-svc-tyo-handler-path2 (*os*))))))

THEOREM: tm-cpu-os-svc-tyo-handler-path2-case2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((\neg array-qemptyp (array-deq (os-readyq (*os*)))
 \wedge (array-qfirst (array-deq (os-readyq (*os*)))
 \neq array-qfirst (os-readyq (*os*))))))
 \rightarrow (tm-cpu (os-svc-tyo-handler-path2 (*os*))
 $=$ getnth (os-current-taskid (os-svc-tyo-handler-path2 (*os*)),
 table ('9, os-task-table (os-svc-tyo-handler-path2 (*os*))))))

THEOREM: tm-cpu-os-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge (\neg \text{array-qemptytyp } (\text{array-deq } (\text{os-readyq } (os)))))) \\ \rightarrow & (\text{tm-cpu } (\text{os-svc-tyo-handler-path2 } (os))) \\ & = \text{getnth } (\text{os-current-taskid } (\text{os-svc-tyo-handler-path2 } (os)), \\ & \quad \text{table } ('9, \text{os-task-table } (\text{os-svc-tyo-handler-path2 } (os)))) \end{aligned}$$

EVENT: Disable tm-cpu-os-svc-tyo-handler-path2.

THEOREM: mapup-cpu-os-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{array-qemptytyp } (\text{array-deq } (\text{os-readyq } (os)))) \\ & \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-svc-tyo-handler-path2 } (os))) \\ & = \text{mapup-cpu } (\text{taskid}, os) \end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path2.

THEOREM: mapup-regs-os-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{array-qemptytyp } (\text{array-deq } (\text{os-readyq } (os)))) \\ & \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-regs } (\text{taskid}, \text{os-svc-tyo-handler-path2 } (os))) \\ & = \text{mapup-regs } (\text{taskid}, os) \end{aligned}$$

EVENT: Disable mapup-regs-os-svc-tyo-handler-path2.

THEOREM: mapup-cc-os-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{array-qemptytyp } (\text{array-deq } (\text{os-readyq } (os)))) \\ & \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-cc } (\text{taskid}, \text{os-svc-tyo-handler-path2 } (os))) \\ & = \text{mapup-cc } (\text{taskid}, os) \end{aligned}$$

EVENT: Disable mapup-cc-os-svc-tyo-handler-path2.

THEOREM: mapup-error-os-svc-tyo-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \end{aligned}$$

$$\begin{aligned}
& \wedge ((\neg \text{array-qemptyp} (\text{array-deq} (\text{os-readyq} (os)))) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error} (taskid, \text{os-svc-tyo-handler-path2} (os)) \\
& = \text{mapup-error} (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-tyo-handler-path2.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path2

$$\begin{aligned}
& (\text{good-os} (os) \\
& \quad \wedge ((\neg \text{tm-waiting} (os)) \\
& \quad \quad \wedge ((\neg \text{array-qemptyp} (\text{array-deq} (\text{os-readyq} (os)))) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcflag} (taskid, \text{os-svc-tyo-handler-path2} (os)) \\
& = \text{mapup-svcflag} (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path2.

THEOREM: mapup-svcid-os-svc-tyo-handler-path2

$$\begin{aligned}
& (\text{good-os} (os) \\
& \quad \wedge ((\neg \text{tm-waiting} (os)) \\
& \quad \quad \wedge ((\neg \text{array-qemptyp} (\text{array-deq} (\text{os-readyq} (os)))) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid} (taskid, \text{os-svc-tyo-handler-path2} (os)) \\
& = \text{mapup-svcid} (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path2.

THEOREM: mapup-task-os-svc-tyo-handler-path2

$$\begin{aligned}
& (\text{good-os} (os) \\
& \quad \wedge ((\neg \text{tm-waiting} (os)) \\
& \quad \quad \wedge ((\neg \text{array-qemptyp} (\text{array-deq} (\text{os-readyq} (os)))) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task} (taskid, \text{os-svc-tyo-handler-path2} (os)) \\
& = \text{mapup-task} (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-tyo-handler-path2.

THEOREM: mapup-tasks-os-svc-tyo-handler-path2

$$\begin{aligned}
& (\text{good-os} (os) \\
& \quad \wedge ((\neg \text{tm-waiting} (os)) \\
& \quad \quad \wedge ((\neg \text{array-qemptyp} (\text{array-deq} (\text{os-readyq} (os)))) \\
& \quad \quad \quad \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks} (taskid, \text{os-svc-tyo-handler-path2} (os)) \\
& = \text{mapup-tasks} (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path2.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path2
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qempty (array-deq (os-readyq (*os*))))))
→ (mapup-os-tasks (os-svc-tyo-handler-path2 (*os*)) = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path2.

THEOREM: mapup-os-ibuffers-os-svc-tyo-handler-path2
good-os (*os*)
→ (mapup-os-ibuffers (os-svc-tyo-handler-path2 (*os*))
 = mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-tyo-handler-path2.

THEOREM: mapup-os-obuffers-os-svc-tyo-handler-path2
good-os (*os*)
→ (mapup-os-obuffers (os-svc-tyo-handler-path2 (*os*))
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-tyo-handler-path2.

THEOREM: mapup-os-mbuffers-os-svc-tyo-handler-path2
good-os (*os*)
→ (mapup-os-mbuffers (os-svc-tyo-handler-path2 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-tyo-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-tyo-handler-path2
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
→ (mapup-queue (os-readyq (os-svc-tyo-handler-path2 (*os*)))
 = deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyo-handler-path2.

THEOREM: mapup-os-status-table-os-svc-tyo-handler-path2
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
→ (table ('2, os-status-table (os-svc-tyo-handler-path2 (*os*)))
 = putnth ('(4 0),

array-qfirst (os-readyq (os)),
table ('2, os-status-table (os)))

EVENT: Disable mapup-os-status-table-os-svc-tyo-handler-path2.

THEOREM: tm-clock-os-svc-tyo-handler-path2
tm-clock (os-svc-tyo-handler-path2 (os)) = '1000

EVENT: Disable tm-clock-os-svc-tyo-handler-path2.

THEOREM: tm-iports-os-svc-tyo-handler-path2
tm-iports (os-svc-tyo-handler-path2 (os)) = tm-iports (os)

EVENT: Disable tm-iports-os-svc-tyo-handler-path2.

THEOREM: tm-oports-os-svc-tyo-handler-path2
tm-oports (os-svc-tyo-handler-path2 (os)) = tm-oports (os)

EVENT: Disable tm-oports-os-svc-tyo-handler-path2.

THEOREM: tm-rwstate-os-svc-tyo-handler-path2
tm-rwstate (os-svc-tyo-handler-path2 (os)) = '0

EVENT: Disable tm-rwstate-os-svc-tyo-handler-path2.

THEOREM: correctness-of-os-svc-tyo-handler-path2
(good-os (os)
 \wedge ((\neg tm-waiting (os))
 \wedge ((\neg tm-errorp (os))
 \wedge (((tm-svcid (os) **mod** '4) = '2)
 \wedge (array-qfullp (os-current-obuffer (os-current-taskid (os),
os)))
 \wedge (\neg array-qempty (array-deq (os-readyq (os))))))))))
 \rightarrow (mapup-os (os-svc-tyo-handler-path2 (os))
= ak-execute-output (os-message (os),
os-current-taskid (os),
mapup-os (os)))

EVENT: Disable correctness-of-os-svc-tyo-handler-path2.

THEOREM: mapup-base-os-svc-tyo-handler-path3
good-os (os)
 \rightarrow (mapup-base (taskid, os-svc-tyo-handler-path3 (os))
= mapup-base (taskid, os))

EVENT: Disable mapup-base-os-svc-tyo-handler-path3.

THEOREM: mapup-limit-os-svc-tyo-handler-path3
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-tyo-handler-path3 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyo-handler-path3.

THEOREM: getseg-tm-memory-os-svc-tyo-handler-path3
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-tyo-handler-path3 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path3.

THEOREM: tm-cpu-os-svc-tyo-handler-path3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-cpu (os-svc-tyo-handler-path3 (*os*))
= putnth (tm-pack-psw (tm-cc (*os*), tm-error (*os*), '0, tm-svcid (*os*)),
'8,
tm-cpu (*os*)))

EVENT: Disable tm-cpu-os-svc-tyo-handler-path3.

THEOREM: mapup-cpu-os-svc-tyo-handler-path3-case1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-cpu (array-qfirst (os-readyq (*os*)), os-svc-tyo-handler-path3 (*os*))
= putnth (tm-pack-psw (mapup-cc (array-qfirst (os-readyq (*os*)), *os*),
mapup-error (array-qfirst (os-readyq (*os*)), *os*),
'0,
mapup-svcid (array-qfirst (os-readyq (*os*)), *os*)),
'8,
mapup-cpu (array-qfirst (os-readyq (*os*)), *os*)))

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-tyo-handler-path3-case2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \wedge ((\text{taskid} < '16) \\
& \quad \quad \quad \wedge (\text{taskid} \neq \text{array-qfirst } (\text{os-readyq } (os)))))) \\
\rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-svc-tyo-handler-path3 } (os)) \\
& = \text{mapup-cpu } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-svc-tyo-handler-path3 } (os)) \\
& = \text{if } \text{taskid} = \text{array-qfirst } (\text{os-readyq } (os)) \\
& \quad \text{then putnth } (\text{tm-pack-psw } (\text{mapup-cc } (\text{taskid}, os), \\
& \quad \quad \text{mapup-error } (\text{taskid}, os), \\
& \quad \quad '0, \\
& \quad \quad \text{mapup-svcid } (\text{taskid}, os)), \\
& \quad '8, \\
& \quad \text{mapup-cpu } (\text{taskid}, os)) \\
& \text{else mapup-cpu } (\text{taskid}, os) \text{ endif}
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path3.

THEOREM: mapup-regs-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-regs } (\text{taskid}, \text{os-svc-tyo-handler-path3 } (os)) \\
& = \text{mapup-regs } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-svc-tyo-handler-path3.

THEOREM: mapup-cc-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow & (\text{mapup-cc } (\text{taskid}, \text{os-svc-tyo-handler-path3 } (os)) \\
& = \text{mapup-cc } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-svc-tyo-handler-path3.

THEOREM: mapup-error-os-svc-tyo-handler-path3

$$(\text{good-os } (os))$$

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-svc-tyo-handler-path3}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-svc-tyo-handler-path3.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-svc-tyo-handler-path3}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \text{ then } '0 \\
& \quad \text{else } \text{mapup-svcflag}(taskid, os) \text{ endif})
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path3.

THEOREM: mapup-svcid-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-svc-tyo-handler-path3}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path3.

THEOREM: mapup-task-os-svc-tyo-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-svc-tyo-handler-path3}(os)) \\
& = \text{if } taskid = \text{array-qfirst}(\text{os-readyq}(os)) \\
& \quad \text{then } \text{tm-set-svcflag}('0, \text{mapup-task}(taskid, os)) \\
& \quad \text{else } \text{mapup-task}(taskid, os) \text{ endif})
\end{aligned}$$

EVENT: Disable mapup-task-os-svc-tyo-handler-path3.

THEOREM: mapup-tasks-os-svc-tyo-handler-path3-case1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (\text{array-qfirst}(\text{os-readyq}(os)) < taskid)))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-svc-tyo-handler-path3}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path3-case1.

THEOREM: mapup-tasks-os-svc-tyo-handler-path3-case2

$(\text{good-os}(os)$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((taskid \in \mathbf{N})$
 $\wedge \text{if array-qfirst}(os\text{-readyq}(os)) < taskid$
 $\text{then } '*1*false$
 $\text{else } '*1*true \text{ endif}))$
 $\rightarrow (\text{mapup-tasks}(taskid, os\text{-svc-tyo-handler-path3}(os))$
 $= \text{putnth}(\text{tm-set-svcflag}('0,$
 $\text{getnth}(\text{array-qfirst}(os\text{-readyq}(os))$
 $\quad - \quad taskid,$
 $\text{mapup-tasks}(taskid, os)),$
 $\text{array-qfirst}(os\text{-readyq}(os)) - taskid,$
 $\text{mapup-tasks}(taskid, os)))$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path3-case2.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path3-support
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{mapup-os-tasks}(os\text{-svc-tyo-handler-path3}(os))$
 $= \text{putnth}(\text{tm-set-svcflag}('0,$
 $\text{getnth}(\text{array-qfirst}(os\text{-readyq}(os)),$
 $\text{mapup-os-tasks}(os)),$
 $\text{array-qfirst}(os\text{-readyq}(os)),$
 $\text{mapup-os-tasks}(os)))$

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path3-support.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path3
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{mapup-os-tasks}(os\text{-svc-tyo-handler-path3}(os))$
 $= \text{putnth}(\text{tm-set-svcflag}('0,$
 $\text{getnth}(\text{qfirst}(\text{mapup-queue}(os\text{-readyq}(os))),$
 $\text{mapup-os-tasks}(os)),$
 $\text{qfirst}(\text{mapup-queue}(os\text{-readyq}(os))),$
 $\text{mapup-os-tasks}(os)))$

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path3.

THEOREM: mapup-os-ibuffers-os-svc-tyo-handler-path3
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-os-ibuffers}(os\text{-svc-tyo-handler-path3}(os))$
 $= \text{mapup-os-ibuffers}(os))$

EVENT: Disable mapup-os-ibuffers-os-svc-tyo-handler-path3.

THEOREM: mapup-os-obuffers-os-svc-tyo-handler-path3
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))))))
 \rightarrow (mapup-os-obuffers (os-svc-tyo-handler-path3 (*os*)
 = enq-ith-buffer (os-message (*os*),
 os-current-taskid (*os*),
 mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-svc-tyo-handler-path3.

THEOREM: mapup-os-mbuffers-os-svc-tyo-handler-path3
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-tyo-handler-path3 (*os*)
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-tyo-handler-path3.

THEOREM: mapup-queue-os-readyq-os-svc-tyo-handler-path3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-tyo-handler-path3 (*os*)))
 = mapup-queue (os-readyq (*os*)))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyo-handler-path3.

THEOREM: mapup-os-status-table-os-svc-tyo-handler-path3
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-tyo-handler-path3 (*os*)))
 = table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-svc-tyo-handler-path3.

THEOREM: tm-clock-os-svc-tyo-handler-path3
 tm-clock (os-svc-tyo-handler-path3 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-tyo-handler-path3.

THEOREM: tm-iports-os-svc-tyo-handler-path3
 tm-iports (os-svc-tyo-handler-path3 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyo-handler-path3.

THEOREM: tm-oports-os-svc-tyo-handler-path3
 $\text{tm-oports (os-svc-tyo-handler-path3 (os))}$
 $= \text{tm-post-output-interrupt (array-qfirst (os-readyq (os)), tm-oports (os))}$

EVENT: Disable tm-oports-os-svc-tyo-handler-path3.

THEOREM: tm-rwstate-os-svc-tyo-handler-path3
 $(\text{good-tm (os)} \wedge (\neg \text{tm-waiting (os)}))$
 $\rightarrow (\text{tm-rwstate (os-svc-tyo-handler-path3 (os))} = \text{tm-rwstate (os)})$

EVENT: Disable tm-rwstate-os-svc-tyo-handler-path3.

THEOREM: correctness-of-os-svc-tyo-handler-path3
 $(\text{good-os (os)}$
 $\wedge ((\neg \text{tm-waiting (os)})$
 $\wedge ((\neg \text{tm-errorp (os)})$
 $\wedge (((\text{tm-svcid (os)} \bmod '4) = '2)$
 $\wedge ((\neg \text{array-qfullp (os-current-obuffer (os-current-taskid (os),$
 $\text{os}))}$
 $\wedge \text{tm-oport-idlep (array-qfirst (os-readyq (os)),$
 $\text{tm-oports (os))}))))))$
 $\rightarrow (\text{mapup-os (os-svc-tyo-handler-path3 (os))}$
 $= \text{ak-execute-output (os-message (os),$
 $\text{os-current-taskid (os),}$
 $\text{mapup-os (os))})$

EVENT: Disable correctness-of-os-svc-tyo-handler-path3.

THEOREM: mapup-base-os-svc-tyo-handler-path4
 good-os (os)
 $\rightarrow (\text{mapup-base (taskid, os-svc-tyo-handler-path4 (os))}$
 $= \text{mapup-base (taskid, os)})$

EVENT: Disable mapup-base-os-svc-tyo-handler-path4.

THEOREM: mapup-limit-os-svc-tyo-handler-path4
 good-os (os)
 $\rightarrow (\text{mapup-limit (taskid, os-svc-tyo-handler-path4 (os))}$
 $= \text{mapup-limit (taskid, os)})$

EVENT: Disable mapup-limit-os-svc-tyo-handler-path4.

THEOREM: getseg-tm-memory-os-svc-tyo-handler-path4

$$\begin{aligned}
& (\text{good-os } (os) \wedge (\text{taskid} < '16)) \\
\rightarrow & (\text{getseg } (\text{mapup-base } (taskid, os), \\
& \quad \text{mapup-limit } (taskid, os), \\
& \quad \text{tm-memory } (\text{os-svc-tyo-handler-path4 } (os))) \\
= & \text{getseg } (\text{mapup-base } (taskid, os), \\
& \quad \text{mapup-limit } (taskid, os), \\
& \quad \text{tm-memory } (os))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path4.

$$\begin{aligned}
\text{THEOREM: tm-cpu-os-svc-tyo-handler-path4} \\
& (\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{tm-cpu } (\text{os-svc-tyo-handler-path4 } (os)) \\
= & \text{putnth } (\text{tm-pack-psw } (\text{tm-cc } (os), \text{tm-error } (os), '0, \text{tm-svcid } (os)), \\
& \quad '8, \\
& \quad \text{tm-cpu } (os))
\end{aligned}$$

EVENT: Disable tm-cpu-os-svc-tyo-handler-path4.

$$\begin{aligned}
\text{THEOREM: mapup-cpu-os-svc-tyo-handler-path4-case1} \\
& (\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{mapup-cpu } (\text{array-qfirst } (\text{os-readyq } (os)), \text{os-svc-tyo-handler-path4 } (os)) \\
= & \text{putnth } (\text{tm-pack-psw } (\text{mapup-cc } (\text{array-qfirst } (\text{os-readyq } (os)), os), \\
& \quad \text{mapup-error } (\text{array-qfirst } (\text{os-readyq } (os)), os), \\
& \quad '0, \\
& \quad \text{mapup-svcid } (\text{array-qfirst } (\text{os-readyq } (os)), os)), \\
& \quad '8, \\
& \quad \text{mapup-cpu } (\text{array-qfirst } (\text{os-readyq } (os)), os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path4-case1.

$$\begin{aligned}
\text{THEOREM: mapup-cpu-os-svc-tyo-handler-path4-case2} \\
& (\text{good-os } (os) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \\
& \quad \quad \quad \wedge ((\text{taskid} < '16) \\
& \quad \quad \quad \quad \wedge (\text{taskid} \neq \text{array-qfirst } (\text{os-readyq } (os)))))) \\
\rightarrow & (\text{mapup-cpu } (taskid, \text{os-svc-tyo-handler-path4 } (os)) \\
= & \text{mapup-cpu } (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path4-case2.

THEOREM: mapup-cpu-os-svc-tyo-handler-path4

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-cpu (taskid, os-svc-tyo-handler-path4 (os))
   =  if taskid = array-qfirst (os-readyq (os))
      then putnth (tm-pack-psw (mapup-cc (taskid, os),
                                     mapup-error (taskid, os),
                                     '0,
                                     mapup-svcid (taskid, os)),
                  '8,
                  mapup-cpu (taskid, os))
      else mapup-cpu (taskid, os) endif)

```

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path4.

THEOREM: mapup-regs-os-svc-tyo-handler-path4

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-regs (taskid, os-svc-tyo-handler-path4 (os))
   =  mapup-regs (taskid, os))

```

EVENT: Disable mapup-regs-os-svc-tyo-handler-path4.

THEOREM: mapup-cc-os-svc-tyo-handler-path4

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-cc (taskid, os-svc-tyo-handler-path4 (os))
   =  mapup-cc (taskid, os))

```

EVENT: Disable mapup-cc-os-svc-tyo-handler-path4.

THEOREM: mapup-error-os-svc-tyo-handler-path4

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-error (taskid, os-svc-tyo-handler-path4 (os))
   =  mapup-error (taskid, os))

```

EVENT: Disable mapup-error-os-svc-tyo-handler-path4.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path4

```

(good-os (os)
  ∧ ((¬ tm-waiting (os)) ∧ ((taskid ∈ ℕ) ∧ (taskid < '16))))
→ (mapup-svcflag (taskid, os-svc-tyo-handler-path4 (os))
   =  if taskid = array-qfirst (os-readyq (os)) then '0
      else mapup-svcflag (taskid, os) endif)

```

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path4.

THEOREM: mapup-svcid-os-svc-tyo-handler-path4
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-svc-tyo-handler-path4 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path4.

THEOREM: mapup-task-os-svc-tyo-handler-path4
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))
→ (mapup-task (*taskid*, os-svc-tyo-handler-path4 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*))
 then tm-set-svcflag ('0, mapup-task (*taskid*, *os*))
 else mapup-task (*taskid*, *os*) **endif**)

EVENT: Disable mapup-task-os-svc-tyo-handler-path4.

THEOREM: mapup-tasks-os-svc-tyo-handler-path4-case1
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (array-qfirst (os-readyq (*os*)) < *taskid*))))
→ (mapup-tasks (*taskid*, os-svc-tyo-handler-path4 (*os*))
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path4-case1.

THEOREM: mapup-tasks-os-svc-tyo-handler-path4-case2
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge ((*taskid* \in \mathbf{N})
 \wedge **if** array-qfirst (os-readyq (*os*)) < *taskid*
 then '*1*false
 else '*1*true **endif**)))
→ (mapup-tasks (*taskid*, os-svc-tyo-handler-path4 (*os*))
 = putnth (tm-set-svcflag ('0,
 getnth (array-qfirst (os-readyq (*os*))
 - *taskid*,
 mapup-tasks (*taskid*, *os*))),
 array-qfirst (os-readyq (*os*)) - *taskid*,
 mapup-tasks (*taskid*, *os*)))

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path4-case2.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path4-support
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-tyo-handler-path4 (*os*))
= putnth (tm-set-svcflag ('0,
getnth (array-qfirst (os-readyq (*os*)),
mapup-os-tasks (*os*))),
array-qfirst (os-readyq (*os*)),
mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path4-support.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path4
(good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-os-tasks (os-svc-tyo-handler-path4 (*os*))
= putnth (tm-set-svcflag ('0,
getnth (qfirst (mapup-queue (os-readyq (*os*)),
mapup-os-tasks (*os*))),
qfirst (mapup-queue (os-readyq (*os*))),
mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-tyo-handler-path4.

THEOREM: mapup-os-ibuffers-os-svc-tyo-handler-path4
good-os (*os*)
 \rightarrow (mapup-os-ibuffers (os-svc-tyo-handler-path4 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-tyo-handler-path4.

THEOREM: mapup-os-obuffers-os-svc-tyo-handler-path4
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (\neg array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))))))
 \rightarrow (mapup-os-obuffers (os-svc-tyo-handler-path4 (*os*))
= enq-ith-buffer (os-message (*os*),
os-current-taskid (*os*),
mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-svc-tyo-handler-path4.

THEOREM: mapup-os-mbuffers-os-svc-tyo-handler-path4
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-svc-tyo-handler-path4 (*os*)))
 = mapup-os-mbuffers (*os*)

EVENT: Disable mapup-os-mbuffers-os-svc-tyo-handler-path4.

THEOREM: mapup-queue-os-readyq-os-svc-tyo-handler-path4
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (mapup-queue (os-readyq (os-svc-tyo-handler-path4 (*os*))))
 = mapup-queue (os-readyq (*os*)))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyo-handler-path4.

THEOREM: mapup-os-status-table-os-svc-tyo-handler-path4
 (good-os (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (table ('2, os-status-table (os-svc-tyo-handler-path4 (*os*))))
 = table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-svc-tyo-handler-path4.

THEOREM: tm-clock-os-svc-tyo-handler-path4
 tm-clock (os-svc-tyo-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-tyo-handler-path4.

THEOREM: tm-iports-os-svc-tyo-handler-path4
 tm-iports (os-svc-tyo-handler-path4 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyo-handler-path4.

THEOREM: tm-oports-os-svc-tyo-handler-path4
 tm-oports (os-svc-tyo-handler-path4 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-tyo-handler-path4.

THEOREM: tm-rwstate-os-svc-tyo-handler-path4
 (good-tm (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-rwstate (os-svc-tyo-handler-path4 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-tyo-handler-path4.

THEOREM: correctness-of-os-svc-tyo-handler-path4

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{tm-errorp } (os)) \\ & \quad \quad \wedge (((\text{tm-svcid } (os) \bmod '4) = '2) \\ & \quad \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-obuffer } (\text{os-current-taskid } (os), \\ & \quad \quad \quad \quad \quad \quad \quad os)))))) \\ & \quad \quad \quad \wedge (\neg \text{tm-oport-idlep } (\text{array-qfirst } (\text{os-readyq } (os)), \\ & \quad \quad \quad \quad \quad \quad \quad \text{tm-oports } (os)))))) \\ \rightarrow & (\text{mapup-os } (\text{os-svc-tyo-handler-path4 } (os)) \\ & = \text{ak-execute-output } (\text{os-message } (os), \\ & \quad \quad \quad \text{os-current-taskid } (os), \\ & \quad \quad \quad \text{mapup-os } (os))) \end{aligned}$$

EVENT: Disable correctness-of-os-svc-tyo-handler-path4.

THEOREM: correctness-of-os-svc-tyo-handler

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge ((\neg \text{tm-errorp } (os)) \wedge ((\text{tm-svcid } (os) \bmod '4) = '2)))) \\ \rightarrow & (\text{mapup-os } (\text{os-svc-tyo-handler } (os)) \\ & = \text{ak-execute-output } (\text{os-message } (os), \\ & \quad \quad \quad \text{os-current-taskid } (os), \\ & \quad \quad \quad \text{mapup-os } (os))) \end{aligned}$$

EVENT: Disable correctness-of-os-svc-tyo-handler.

THEOREM: mapup-base-os-waiting-input-handler-path1

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-base } (taskid, \text{os-waiting-input-handler-path1 } (os)) \\ & = \text{mapup-base } (taskid, os)) \end{aligned}$$

EVENT: Disable mapup-base-os-waiting-input-handler-path1.

THEOREM: mapup-limit-os-waiting-input-handler-path1

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-limit } (taskid, \text{os-waiting-input-handler-path1 } (os)) \\ & = \text{mapup-limit } (taskid, os)) \end{aligned}$$

EVENT: Disable mapup-limit-os-waiting-input-handler-path1.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path1

$$(\text{good-os } (os) \wedge (taskid < '16))$$

```

→ (getseg (mapup-base (taskid, os),
    mapup-limit (taskid, os),
    tm-memory (os-waiting-input-handler-path1 (os)))
=  getseg (mapup-base (taskid, os),
    mapup-limit (taskid, os),
    tm-memory (os))

```

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path1.

```

(PROVE-LEMMA
TM-CPU-OS-WAITING-INPUT-HANDLER-PATH1-CROCK
(REWRITE)
(EQUAL
(TM-CPU (OS-WAITING-INPUT-HANDLER-PATH1 OS))
(CONS
(GETNTH '0
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '1
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '2
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '3
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '4
(GETSEG '0
'8

```

```

                                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                                (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '5
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '6
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '7
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (TM-PACK-PSW
    (TM-UNPACK-CC
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-ERROR
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCFLAG
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCID
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS))))
    'NIL)))))))))
((ENABLE OS-WAITING-INPUT-HANDLER-PATH1 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4

```

TM-R5 TM-R6 TM-R7 TM-UNPACK-CC TM-UNPACK-ERROR TM-UNPACK-SVCFLAG
 TM-UNPACK-SVCID)
 (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
 (DISABLE-THEORY T))

EVENT: Disable tm-cpu-os-waiting-input-handler-path1-crock.

THEOREM: tm-cpu-os-waiting-input-handler-path1
 (good-os (*os*)
 \wedge (tm-waiting (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*))))
 \rightarrow (tm-cpu (os-waiting-input-handler-path1 (*os*))
 $=$ getnth (os-current-taskid (os-waiting-input-handler-path1 (*os*)),
 table ('9,
 os-task-table (os-waiting-input-handler-path1 (*os*))))))

EVENT: Disable tm-cpu-os-waiting-input-handler-path1.

THEOREM: mapup-cpu-os-waiting-input-handler-path1
 (good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cpu (*taskid*, os-waiting-input-handler-path1 (*os*))
 $=$ mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path1.

THEOREM: mapup-regs-os-waiting-input-handler-path1
 (good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-regs (*taskid*, os-waiting-input-handler-path1 (*os*))
 $=$ mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-input-handler-path1.

THEOREM: mapup-cc-os-waiting-input-handler-path1
 (good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-waiting-input-handler-path1 (*os*))
 $=$ mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-input-handler-path1.

THEOREM: mapup-error-os-waiting-input-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-error (*taskid*, os-waiting-input-handler-path1 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-input-handler-path1.

THEOREM: mapup-svcflag-os-waiting-input-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcflag (*taskid*, os-waiting-input-handler-path1 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path1.

THEOREM: mapup-svcid-os-waiting-input-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcid (*taskid*, os-waiting-input-handler-path1 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-input-handler-path1.

THEOREM: mapup-task-os-waiting-input-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-task (*taskid*, os-waiting-input-handler-path1 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-input-handler-path1.

THEOREM: mapup-tasks-os-waiting-input-handler-path1

THEOREM: mapup-queue-os-readyq-os-waiting-input-handler-path1
 $(\text{good-os}(os) \wedge \text{tm-waiting}(os))$
 $\rightarrow (\text{mapup-queue}(\text{os-readyq}(\text{os-waiting-input-handler-path1}(os))))$
 $= \text{enq}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\text{mapup-queue}(\text{os-readyq}(os)))$

EVENT: Disable mapup-queue-os-readyq-os-waiting-input-handler-path1.

THEOREM: mapup-os-status-table-os-waiting-input-handler-path1
 $(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow (\text{table}('2, \text{os-status-table}(\text{os-waiting-input-handler-path1}(os))))$
 $= \text{putnth}('0\ 0),$
 $\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\text{table}('2, \text{os-status-table}(os)))$

EVENT: Disable mapup-os-status-table-os-waiting-input-handler-path1.

THEOREM: tm-rwstate-os-waiting-input-handler-path1
 $\text{tm-rwstate}(\text{os-waiting-input-handler-path1}(os)) = '0$

EVENT: Disable tm-rwstate-os-waiting-input-handler-path1.

THEOREM: tm-clock-os-waiting-input-handler-path1
 $\text{tm-clock}(\text{os-waiting-input-handler-path1}(os)) = '1000$

EVENT: Disable tm-clock-os-waiting-input-handler-path1.

THEOREM: tm-iports-os-waiting-input-handler-path1
 $\text{tm-iports}(\text{os-waiting-input-handler-path1}(os))$
 $= \text{tm-clear-input-interrupt}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\text{tm-iports}(os))$

EVENT: Disable tm-iports-os-waiting-input-handler-path1.

THEOREM: tm-oports-os-waiting-input-handler-path1
 $\text{tm-oports}(\text{os-waiting-input-handler-path1}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-waiting-input-handler-path1.

THEOREM: ak-waiting-to-inputp-mapup-os
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)))$

$$\begin{aligned} & \wedge (\text{getnth} ('2 * \text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \text{os-status-table} (os)) \\ & \quad = '5))) \\ \rightarrow & \text{ak-waiting-to-inputp} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)), \\ & \quad \text{mapup-os} (os)) \end{aligned}$$

EVENT: Disable ak-waiting-to-inputp-mapup-os.

THEOREM: correctness-of-array-qfullp-ibuffers-instance
 $(\text{array-queuep} (queue) \wedge ('4 = \text{getnth} ('3, queue)))$
 $\rightarrow (\text{array-qfullp} (queue) = \text{qfullp} (\text{mapup-queue} (queue), '4))$

EVENT: Disable correctness-of-array-qfullp-ibuffers-instance.

THEOREM: correctness-of-os-waiting-input-handler-path1
 $(\text{good-os} (os)$
 $\wedge (\text{tm-some-input-interruptp} (\text{tm-iports} (os))$
 $\wedge (\text{tm-waiting} (os)$
 $\wedge ((\text{getnth} ('2$
 $\quad * \text{tm-interrupting-input-port} (\text{tm-iports} (os)),$
 $\quad \text{os-status-table} (os))$
 $\quad = '5)$
 $\wedge \text{array-qfullp} (\text{os-current-ibuffer} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)),$
 $\quad \text{os}))))))$
 $\rightarrow (\text{mapup-os} (\text{os-waiting-input-handler-path1} (os))$
 $\quad = \text{ak-waiting-input-interrupt-handler} (\text{tm-interrupting-input-port} (\text{tm-iports} (os)),$
 $\quad \text{mapup-os} (os)))$

EVENT: Disable correctness-of-os-waiting-input-handler-path1.

THEOREM: mapup-base-os-waiting-input-handler-path2
 $\text{good-os} (os)$
 $\rightarrow (\text{mapup-base} (taskid, \text{os-waiting-input-handler-path2} (os))$
 $\quad = \text{mapup-base} (taskid, os))$

EVENT: Disable mapup-base-os-waiting-input-handler-path2.

THEOREM: mapup-limit-os-waiting-input-handler-path2
 $\text{good-os} (os)$
 $\rightarrow (\text{mapup-limit} (taskid, \text{os-waiting-input-handler-path2} (os))$
 $\quad = \text{mapup-limit} (taskid, os))$

EVENT: Disable mapup-limit-os-waiting-input-handler-path2.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path2
 (good-os (*os*) \wedge (*taskid* < '16))
 \rightarrow (getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (os-waiting-input-handler-path2 (*os*)))
 = getseg (mapup-base (*taskid*, *os*),
 mapup-limit (*taskid*, *os*),
 tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path2.

```
(PROVE-LEMMA
  TM-CPU-OS-WAITING-INPUT-HANDLER-PATH2-CROCK
  (REWRITE)
  (EQUAL
    (TM-CPU (OS-WAITING-INPUT-HANDLER-PATH2 OS))
    (CONS
      (GETNTH '0
        (GETSEG '0
          '8
          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
            (TABLE '9 (OS-TASK-TABLE OS))))))
      (CONS
        (GETNTH '1
          (GETSEG '0
            '8
            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
              (TABLE '9 (OS-TASK-TABLE OS))))))
        (CONS
          (GETNTH '2
            (GETSEG '0
              '8
              (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                (TABLE '9 (OS-TASK-TABLE OS))))))
          (CONS
            (GETNTH '3
              (GETSEG '0
                '8
                (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
                  (TABLE '9 (OS-TASK-TABLE OS))))))
            (CONS
              (GETNTH '4
```

```

        (GETSEG '0
          '8
          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
            (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '5
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '6
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '7
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (TM-PACK-PSW
    (TM-UNPACK-CC
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-ERROR
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCFLAG
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCID
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS))))))

```

```

      'NIL)))))))))
((ENABLE OS-WAITING-INPUT-HANDLER-PATH2 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
  TM-R5 TM-R6 TM-R7 TM-UNPACK-CC TM-UNPACK-ERROR TM-UNPACK-SVCFLAG
  TM-UNPACK-SVCID)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-waiting-input-handler-path2-crock.

THEOREM: tm-cpu-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-waiting } (os) \wedge \text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\
\rightarrow & (\text{tm-cpu } (\text{os-waiting-input-handler-path2 } (os)) \\
& = \text{getnth } (\text{os-current-taskid } (\text{os-waiting-input-handler-path2 } (os)), \\
& \quad \text{table } ('9, \\
& \quad \quad \text{os-task-table } (\text{os-waiting-input-handler-path2 } (os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-waiting-input-handler-path2.

THEOREM: mapup-cpu-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-waiting } (os) \\
& \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cpu } (\text{taskid}, \text{os-waiting-input-handler-path2 } (os)) \\
& = \text{mapup-cpu } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-waiting-input-handler-path2.

THEOREM: mapup-regs-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-waiting } (os) \\
& \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-regs } (\text{taskid}, \text{os-waiting-input-handler-path2 } (os)) \\
& = \text{mapup-regs } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-waiting-input-handler-path2.

THEOREM: mapup-cc-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-waiting } (os) \\
& \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))
\end{aligned}$$

$$\begin{aligned}
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-waiting-input-handler-path2}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-waiting-input-handler-path2.

THEOREM: mapup-error-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-waiting-input-handler-path2}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-waiting-input-handler-path2.

THEOREM: mapup-svcflag-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-waiting-input-handler-path2}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path2.

THEOREM: mapup-svcid-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-waiting-input-handler-path2}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-waiting-input-handler-path2.

THEOREM: mapup-task-os-waiting-input-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-waiting-input-handler-path2}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-waiting-input-handler-path2.

THEOREM: mapup-tasks-os-waiting-input-handler-path2

$$\begin{aligned} & (\text{good-os } (os) \\ & \wedge (\text{tm-waiting } (os) \\ & \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \wedge (\text{taskid} \in \mathbf{N})))) \\ \rightarrow & (\text{mapup-tasks } (\text{taskid}, \text{os-waiting-input-handler-path2 } (os)) \\ & = \text{mapup-tasks } (\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-tasks-os-waiting-input-handler-path2.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path2

$$\begin{aligned} & (\text{good-os } (os) \\ & \wedge (\text{tm-waiting } (os) \wedge \text{tm-some-input-interruptp } (\text{tm-iports } (os)))) \\ \rightarrow & (\text{mapup-os-tasks } (\text{os-waiting-input-handler-path2 } (os)) \\ & = \text{mapup-os-tasks } (os)) \end{aligned}$$

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path2.

THEOREM: mapup-os-ibuffers-os-waiting-input-handler-path2

$$\begin{aligned} & (\text{good-os } (os) \\ & \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\ & \quad \wedge (\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad os)))))) \\ \rightarrow & (\text{mapup-os-ibuffers } (\text{os-waiting-input-handler-path2 } (os)) \\ & = \text{enq-ith-buffer } (\text{tm-overflow-char } (\text{tm-ichar } (\text{getnth } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os))), \\ & \quad \quad \quad \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{mapup-os-ibuffers } (os))) \end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-waiting-input-handler-path2.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path2

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-os-obuffers } (\text{os-waiting-input-handler-path2 } (os)) \\ & = \text{mapup-os-obuffers } (os)) \end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path2.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path2

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-os-mbuffers } (\text{os-waiting-input-handler-path2 } (os)) \\ & = \text{mapup-os-mbuffers } (os)) \end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path2.

THEOREM: mapup-queue-os-readyq-os-waiting-input-handler-path2
(good-os(*os*) ∧ tm-waiting(*os*))
→ (mapup-queue(os-readyq(os-waiting-input-handler-path2(*os*)))
= enq(tm-interrupting-input-port(tm-iports(*os*)),
mapup-queue(os-readyq(*os*))))

EVENT: Disable mapup-queue-os-readyq-os-waiting-input-handler-path2.

THEOREM: mapup-os-status-table-os-waiting-input-handler-path2
(good-os(*os*) ∧ tm-some-input-interruptp(tm-iports(*os*)))
→ (table('2, os-status-table(os-waiting-input-handler-path2(*os*)))
= putnth('0 0),
tm-interrupting-input-port(tm-iports(*os*)),
table('2, os-status-table(*os*))))

EVENT: Disable mapup-os-status-table-os-waiting-input-handler-path2.

THEOREM: tm-rwstate-os-waiting-input-handler-path2
tm-rwstate(os-waiting-input-handler-path2(*os*)) = '0

EVENT: Disable tm-rwstate-os-waiting-input-handler-path2.

THEOREM: tm-clock-os-waiting-input-handler-path2
tm-clock(os-waiting-input-handler-path2(*os*)) = '1000

EVENT: Disable tm-clock-os-waiting-input-handler-path2.

THEOREM: tm-iports-os-waiting-input-handler-path2
tm-iports(os-waiting-input-handler-path2(*os*))
= tm-clear-input-interrupt(tm-interrupting-input-port(tm-iports(*os*)),
tm-iports(*os*))

EVENT: Disable tm-iports-os-waiting-input-handler-path2.

THEOREM: tm-oports-os-waiting-input-handler-path2
tm-oports(os-waiting-input-handler-path2(*os*)) = tm-oports(*os*)

EVENT: Disable tm-oports-os-waiting-input-handler-path2.

THEOREM: correctness-of-os-waiting-input-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))) \\ & \quad \wedge (\text{tm-waiting } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2) \\ & \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad = '5) \\ & \quad \quad \wedge ((\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{os}))) \\ & \quad \quad \quad \wedge \text{tm-iport-errorp } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os)))))) \\ \rightarrow & (\text{mapup-os } (\text{os-waiting-input-handler-path2 } (os))) \\ & = \text{ak-waiting-input-interrupt-handler } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \text{mapup-os } (os)) \end{aligned}$$

EVENT: Disable correctness-of-os-waiting-input-handler-path2.

THEOREM: mapup-base-os-waiting-input-handler-path3

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-base } (taskid, \text{os-waiting-input-handler-path3 } (os))) \\ & = \text{mapup-base } (taskid, os) \end{aligned}$$

EVENT: Disable mapup-base-os-waiting-input-handler-path3.

THEOREM: mapup-limit-os-waiting-input-handler-path3

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-limit } (taskid, \text{os-waiting-input-handler-path3 } (os))) \\ & = \text{mapup-limit } (taskid, os) \end{aligned}$$

EVENT: Disable mapup-limit-os-waiting-input-handler-path3.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path3

$$\begin{aligned} & (\text{good-os } (os) \wedge (taskid < '16)) \\ \rightarrow & (\text{getseg } (\text{mapup-base } (taskid, os), \\ & \quad \text{mapup-limit } (taskid, os), \\ & \quad \text{tm-memory } (\text{os-waiting-input-handler-path3 } (os)))) \\ & = \text{getseg } (\text{mapup-base } (taskid, os), \\ & \quad \text{mapup-limit } (taskid, os), \\ & \quad \text{tm-memory } (os)) \end{aligned}$$

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path3.


```

(GETNTH '6
  (GETSEG '0
    '8
    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
      (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (GETNTH '7
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (TM-PACK-PSW
    (TM-UNPACK-CC
      (GETNTH (PLUS '8
        (TIMES '9
          (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
        (OS-TASK-TABLE OS)))
      (TM-UNPACK-ERROR
        (GETNTH (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
          (OS-TASK-TABLE OS)))
      (TM-UNPACK-SVCFLAG
        (GETNTH (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
          (OS-TASK-TABLE OS)))
      (TM-UNPACK-SVCID
        (GETNTH (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-INPUT-PORT (TM-IPOINTS OS))))
          (OS-TASK-TABLE OS))))
    'NIL)))))))))
((ENABLE OS-WAITING-INPUT-HANDLER-PATH3 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
  TM-R5 TM-R6 TM-R7 TM-UNPACK-CC TM-UNPACK-ERROR TM-UNPACK-SVCFLAG
  TM-UNPACK-SVCID)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-waiting-input-handler-path3-crock.

THEOREM: tm-cpu-os-waiting-input-handler-path3
 (good-os (*os*))

$$\begin{aligned}
& \wedge (\text{tm-waiting}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os))) \\
\rightarrow & (\text{tm-cpu}(\text{os-waiting-input-handler-path3}(os)) \\
& = \text{getnth}(\text{os-current-taskid}(\text{os-waiting-input-handler-path3}(os)), \\
& \quad \text{table}('9, \\
& \quad \quad \text{os-task-table}(\text{os-waiting-input-handler-path3}(os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-waiting-input-handler-path3.

THEOREM: mapup-cpu-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cpu}(\text{taskid}, \text{os-waiting-input-handler-path3}(os)) \\
& = \text{mapup-cpu}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-waiting-input-handler-path3.

THEOREM: mapup-regs-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-regs}(\text{taskid}, \text{os-waiting-input-handler-path3}(os)) \\
& = \text{mapup-regs}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-waiting-input-handler-path3.

THEOREM: mapup-cc-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cc}(\text{taskid}, \text{os-waiting-input-handler-path3}(os)) \\
& = \text{mapup-cc}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-waiting-input-handler-path3.

THEOREM: mapup-error-os-waiting-input-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16))))))
\end{aligned}$$

$$\begin{aligned} &\rightarrow (\text{mapup-error}(\text{taskid}, \text{os-waiting-input-handler-path3}(\text{os})) \\ &= \text{mapup-error}(\text{taskid}, \text{os})) \end{aligned}$$

EVENT: Disable mapup-error-os-waiting-input-handler-path3.

THEOREM: mapup-svcflag-os-waiting-input-handler-path3

$$\begin{aligned} &(\text{good-os}(\text{os}) \\ &\wedge (\text{tm-waiting}(\text{os}) \\ &\quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(\text{os})) \\ &\quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-svcflag}(\text{taskid}, \text{os-waiting-input-handler-path3}(\text{os})) \\ &= \text{mapup-svcflag}(\text{taskid}, \text{os})) \end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path3.

THEOREM: mapup-svcid-os-waiting-input-handler-path3

$$\begin{aligned} &(\text{good-os}(\text{os}) \\ &\wedge (\text{tm-waiting}(\text{os}) \\ &\quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(\text{os})) \\ &\quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-svcid}(\text{taskid}, \text{os-waiting-input-handler-path3}(\text{os})) \\ &= \text{mapup-svcid}(\text{taskid}, \text{os})) \end{aligned}$$

EVENT: Disable mapup-svcid-os-waiting-input-handler-path3.

THEOREM: mapup-task-os-waiting-input-handler-path3

$$\begin{aligned} &(\text{good-os}(\text{os}) \\ &\wedge (\text{tm-waiting}(\text{os}) \\ &\quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(\text{os})) \\ &\quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-task}(\text{taskid}, \text{os-waiting-input-handler-path3}(\text{os})) \\ &= \text{mapup-task}(\text{taskid}, \text{os})) \end{aligned}$$

EVENT: Disable mapup-task-os-waiting-input-handler-path3.

THEOREM: mapup-tasks-os-waiting-input-handler-path3

$$\begin{aligned} &(\text{good-os}(\text{os}) \\ &\wedge (\text{tm-waiting}(\text{os}) \\ &\quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(\text{os})) \wedge (\text{taskid} \in \mathbf{N})))) \\ &\rightarrow (\text{mapup-tasks}(\text{taskid}, \text{os-waiting-input-handler-path3}(\text{os})) \\ &= \text{mapup-tasks}(\text{taskid}, \text{os})) \end{aligned}$$

EVENT: Disable mapup-tasks-os-waiting-input-handler-path3.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path3
 (good-os (*os*)
 \wedge (tm-waiting (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*))))
 \rightarrow (mapup-os-tasks (os-waiting-input-handler-path3 (*os*))
 = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path3.

THEOREM: mapup-os-ibuffers-os-waiting-input-handler-path3
 (good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge (\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
 \rightarrow (mapup-os-ibuffers (os-waiting-input-handler-path3 (*os*))
 = enq-ith-buffer (tm-ichar (getnth (tm-interrupting-input-port (tm-iports (*os*)),
 tm-iports (*os*))),
 tm-interrupting-input-port (tm-iports (*os*)),
 mapup-os-ibuffers (*os*)))

EVENT: Disable mapup-os-ibuffers-os-waiting-input-handler-path3.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path3
 good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-waiting-input-handler-path3 (*os*))
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path3.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path3
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-waiting-input-handler-path3 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path3.

THEOREM: mapup-queue-os-readyq-os-waiting-input-handler-path3
 (good-os (*os*) \wedge tm-waiting (*os*))
 \rightarrow (mapup-queue (os-readyq (os-waiting-input-handler-path3 (*os*))
 = enq (tm-interrupting-input-port (tm-iports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-waiting-input-handler-path3.

THEOREM: mapup-os-status-table-os-waiting-input-handler-path3
 (good-os (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*)))
 \rightarrow (table ('2, os-status-table (os-waiting-input-handler-path3 (*os*)))
 $=$ putnth ('(0 0),
 tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-waiting-input-handler-path3.

THEOREM: tm-rwstate-os-waiting-input-handler-path3
 tm-rwstate (os-waiting-input-handler-path3 (*os*)) = '0

EVENT: Disable tm-rwstate-os-waiting-input-handler-path3.

THEOREM: tm-clock-os-waiting-input-handler-path3
 tm-clock (os-waiting-input-handler-path3 (*os*)) = '1000

EVENT: Disable tm-clock-os-waiting-input-handler-path3.

THEOREM: tm-iports-os-waiting-input-handler-path3
 tm-iports (os-waiting-input-handler-path3 (*os*))
 $=$ tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
 tm-iports (*os*))

EVENT: Disable tm-iports-os-waiting-input-handler-path3.

THEOREM: tm-oports-os-waiting-input-handler-path3
 tm-oports (os-waiting-input-handler-path3 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-waiting-input-handler-path3.

THEOREM: correctness-of-os-waiting-input-handler-path3
 (good-os (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge (tm-waiting (*os*)
 \wedge ((getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 $=$ '5)
 \wedge ((\neg array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
 os)))
 \wedge (\neg tm-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),

$$\begin{aligned} & \text{tm-iports}(os))))))))) \\ \rightarrow & (\text{mapup-os}(\text{os-waiting-input-handler-path3}(os)) \\ & = \text{ak-waiting-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\ & \text{mapup-os}(os)) \end{aligned}$$

EVENT: Disable correctness-of-os-waiting-input-handler-path3.

THEOREM: mapup-base-os-waiting-input-handler-path4
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(taskid, \text{os-waiting-input-handler-path4}(os))$
 $= \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-waiting-input-handler-path4.

THEOREM: mapup-limit-os-waiting-input-handler-path4
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-waiting-input-handler-path4}(os))$
 $= \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-waiting-input-handler-path4.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path4
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg}(\text{mapup-base}(taskid, os),$
 $\text{mapup-limit}(taskid, os),$
 $\text{tm-memory}(\text{os-waiting-input-handler-path4}(os)))$
 $= \text{getseg}(\text{mapup-base}(taskid, os),$
 $\text{mapup-limit}(taskid, os),$
 $\text{tm-memory}(os))$

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path4.

THEOREM: mapup-cpu-os-waiting-input-handler-path4
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu}(taskid, \text{os-waiting-input-handler-path4}(os))$
 $= \text{mapup-cpu}(taskid, os))$

EVENT: Disable mapup-cpu-os-waiting-input-handler-path4.

THEOREM: mapup-regs-os-waiting-input-handler-path4
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs}(taskid, \text{os-waiting-input-handler-path4}(os))$
 $= \text{mapup-regs}(taskid, os))$

EVENT: Disable mapup-regs-os-waiting-input-handler-path4.

THEOREM: mapup-cc-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-cc(*taskid*, *os*)

EVENT: Disable mapup-cc-os-waiting-input-handler-path4.

THEOREM: mapup-error-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-error(*taskid*, *os*)

EVENT: Disable mapup-error-os-waiting-input-handler-path4.

THEOREM: mapup-svcflag-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-svcflag(*taskid*, *os*)

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path4.

THEOREM: mapup-svcid-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-svcid(*taskid*, *os*)

EVENT: Disable mapup-svcid-os-waiting-input-handler-path4.

THEOREM: mapup-task-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-task(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-task(*taskid*, *os*)

EVENT: Disable mapup-task-os-waiting-input-handler-path4.

THEOREM: mapup-tasks-os-waiting-input-handler-path4
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks(*taskid*, os-waiting-input-handler-path4(*os*)))
= mapup-tasks(*taskid*, *os*)

EVENT: Disable mapup-tasks-os-waiting-input-handler-path4.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path4
 $(\text{good-os } (os) \wedge \text{tm-waiting } (os))$
 $\rightarrow (\text{mapup-os-tasks } (\text{os-waiting-input-handler-path4 } (os)))$
 $= \text{mapup-os-tasks } (os)$

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path4.

THEOREM: mapup-os-ibuffers-os-waiting-input-handler-path4
 $(\text{good-os } (os)$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))$
 $\wedge \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)),$
 $os))))$
 $\rightarrow (\text{mapup-os-ibuffers } (\text{os-waiting-input-handler-path4 } (os)))$
 $= \text{qreplace-ith-buffer } (\text{tm-overflow-char } (\text{tm-ichar } (\text{getnth } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)),$
 $\text{tm-iports } (os))))),$
 $\text{tm-interrupting-input-port } (\text{tm-iports } (os)),$
 $\text{mapup-os-ibuffers } (os))$

EVENT: Disable mapup-os-ibuffers-os-waiting-input-handler-path4.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path4
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-waiting-input-handler-path4 } (os)))$
 $= \text{mapup-os-obuffers } (os)$

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path4.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path4
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-waiting-input-handler-path4 } (os)))$
 $= \text{mapup-os-mbuffers } (os)$

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path4.

THEOREM: tm-rwstate-os-waiting-input-handler-path4
 $\text{tm-rwstate } (\text{os-waiting-input-handler-path4 } (os)) = '1$

EVENT: Disable tm-rwstate-os-waiting-input-handler-path4.

THEOREM: tm-clock-os-waiting-input-handler-path4
 $\text{tm-clock } (\text{os-waiting-input-handler-path4 } (os)) = \text{tm-clock } (os)$

EVENT: Disable tm-clock-os-waiting-input-handler-path4.

THEOREM: tm-iports-os-waiting-input-handler-path4
tm-iports (os-waiting-input-handler-path4 (os))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (os)),
tm-iports (os))

EVENT: Disable tm-iports-os-waiting-input-handler-path4.

THEOREM: tm-oports-os-waiting-input-handler-path4
tm-oports (os-waiting-input-handler-path4 (os)) = tm-oports (os)

EVENT: Disable tm-oports-os-waiting-input-handler-path4.

THEOREM: not-ak-waiting-to-inputp-mapup-os
(good-os (os)
∧ (tm-some-input-interruptp (tm-iports (os))
∧ (getnth ('2 * tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
≠ '5)))
→ (¬ ak-waiting-to-inputp (tm-interrupting-input-port (tm-iports (os)),
mapup-os (os)))

EVENT: Disable not-ak-waiting-to-inputp-mapup-os.

THEOREM: correctness-of-os-waiting-input-handler-path4
(good-os (os)
∧ (tm-some-input-interruptp (tm-iports (os))
∧ (tm-waiting (os)
∧ ((getnth ('2
* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
≠ '5)
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))))
→ (mapup-os (os-waiting-input-handler-path4 (os))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (os)),
mapup-os (os)))

EVENT: Disable correctness-of-os-waiting-input-handler-path4.

THEOREM: mapup-base-os-waiting-input-handler-path5

$\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(taskid, \text{os-waiting-input-handler-path5}(os)))$
 $= \text{mapup-base}(taskid, os)$

EVENT: Disable mapup-base-os-waiting-input-handler-path5.

THEOREM: mapup-limit-os-waiting-input-handler-path5
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-waiting-input-handler-path5}(os)))$
 $= \text{mapup-limit}(taskid, os)$

EVENT: Disable mapup-limit-os-waiting-input-handler-path5.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path5
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(\text{os-waiting-input-handler-path5}(os))))$
 $= \text{getseg}(\text{mapup-base}(taskid, os),$
 $\quad \text{mapup-limit}(taskid, os),$
 $\quad \text{tm-memory}(os))$

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path5.

THEOREM: mapup-cpu-os-waiting-input-handler-path5
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu}(taskid, \text{os-waiting-input-handler-path5}(os)))$
 $= \text{mapup-cpu}(taskid, os)$

EVENT: Disable mapup-cpu-os-waiting-input-handler-path5.

THEOREM: mapup-regs-os-waiting-input-handler-path5
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs}(taskid, \text{os-waiting-input-handler-path5}(os)))$
 $= \text{mapup-regs}(taskid, os)$

EVENT: Disable mapup-regs-os-waiting-input-handler-path5.

THEOREM: mapup-cc-os-waiting-input-handler-path5
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cc}(taskid, \text{os-waiting-input-handler-path5}(os)))$
 $= \text{mapup-cc}(taskid, os)$

EVENT: Disable mapup-cc-os-waiting-input-handler-path5.

THEOREM: mapup-error-os-waiting-input-handler-path5
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error(*taskid*, os-waiting-input-handler-path5(*os*)))
= mapup-error(*taskid*, *os*)

EVENT: Disable mapup-error-os-waiting-input-handler-path5.

THEOREM: mapup-svcflag-os-waiting-input-handler-path5
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag(*taskid*, os-waiting-input-handler-path5(*os*)))
= mapup-svcflag(*taskid*, *os*)

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path5.

THEOREM: mapup-svcid-os-waiting-input-handler-path5
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid(*taskid*, os-waiting-input-handler-path5(*os*)))
= mapup-svcid(*taskid*, *os*)

EVENT: Disable mapup-svcid-os-waiting-input-handler-path5.

THEOREM: mapup-task-os-waiting-input-handler-path5
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-task(*taskid*, os-waiting-input-handler-path5(*os*)))
= mapup-task(*taskid*, *os*)

EVENT: Disable mapup-task-os-waiting-input-handler-path5.

THEOREM: mapup-tasks-os-waiting-input-handler-path5
(good-os(*os*) ∧ (tm-waiting(*os*) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks(*taskid*, os-waiting-input-handler-path5(*os*)))
= mapup-tasks(*taskid*, *os*)

EVENT: Disable mapup-tasks-os-waiting-input-handler-path5.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path5
(good-os(*os*) ∧ tm-waiting(*os*))
→ (mapup-os-tasks(os-waiting-input-handler-path5(*os*)))
= mapup-os-tasks(*os*)

EVENT: Disable tm-iports-os-waiting-input-handler-path5.

THEOREM: tm-oports-os-waiting-input-handler-path5
 $\text{tm-oports}(\text{os-waiting-input-handler-path5}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-waiting-input-handler-path5.

THEOREM: correctness-of-os-waiting-input-handler-path5
 $(\text{good-os}(os) \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge (\text{tm-waiting}(os) \wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\neq '5)$
 $\wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os}))$
 $\wedge \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{tm-iports}(os))))))$
 $\rightarrow (\text{mapup-os}(\text{os-waiting-input-handler-path5}(os))$
 $\quad = \text{ak-waiting-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-waiting-input-handler-path5.

THEOREM: mapup-base-os-waiting-input-handler-path6
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(taskid, \text{os-waiting-input-handler-path6}(os))$
 $\quad = \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-waiting-input-handler-path6.

THEOREM: mapup-limit-os-waiting-input-handler-path6
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-waiting-input-handler-path6}(os))$
 $\quad = \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-waiting-input-handler-path6.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path6
 $(\text{good-os}(os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg}(\text{mapup-base}(taskid, os),$

$$\begin{aligned}
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (os\text{-waiting-input-handler-path6 } (os))) \\
= & \text{getseg } (\text{mapup-base } (taskid, os), \\
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (os))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path6.

$$\begin{aligned}
& \text{THEOREM: mapup-cpu-os-waiting-input-handler-path6} \\
& (\text{good-os } (os) \wedge (\text{tm-waiting } (os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-cpu } (taskid, os\text{-waiting-input-handler-path6 } (os))) \\
& = \text{mapup-cpu } (taskid, os)
\end{aligned}$$

EVENT: Disable mapup-cpu-os-waiting-input-handler-path6.

$$\begin{aligned}
& \text{THEOREM: mapup-regs-os-waiting-input-handler-path6} \\
& (\text{good-os } (os) \wedge (\text{tm-waiting } (os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-regs } (taskid, os\text{-waiting-input-handler-path6 } (os))) \\
& = \text{mapup-regs } (taskid, os)
\end{aligned}$$

EVENT: Disable mapup-regs-os-waiting-input-handler-path6.

$$\begin{aligned}
& \text{THEOREM: mapup-cc-os-waiting-input-handler-path6} \\
& (\text{good-os } (os) \wedge (\text{tm-waiting } (os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-cc } (taskid, os\text{-waiting-input-handler-path6 } (os))) \\
& = \text{mapup-cc } (taskid, os)
\end{aligned}$$

EVENT: Disable mapup-cc-os-waiting-input-handler-path6.

$$\begin{aligned}
& \text{THEOREM: mapup-error-os-waiting-input-handler-path6} \\
& (\text{good-os } (os) \wedge (\text{tm-waiting } (os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error } (taskid, os\text{-waiting-input-handler-path6 } (os))) \\
& = \text{mapup-error } (taskid, os)
\end{aligned}$$

EVENT: Disable mapup-error-os-waiting-input-handler-path6.

$$\begin{aligned}
& \text{THEOREM: mapup-svcflag-os-waiting-input-handler-path6} \\
& (\text{good-os } (os) \wedge (\text{tm-waiting } (os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag } (taskid, os\text{-waiting-input-handler-path6 } (os))) \\
& = \text{mapup-svcflag } (taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path6.

THEOREM: mapup-svcid-os-waiting-input-handler-path6
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16))))$
 $\rightarrow (\text{mapup-svcid}(\text{taskid}, \text{os-waiting-input-handler-path6}(os))$
 $= \text{mapup-svcid}(\text{taskid}, os))$

EVENT: Disable mapup-svcid-os-waiting-input-handler-path6.

THEOREM: mapup-task-os-waiting-input-handler-path6
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16))))$
 $\rightarrow (\text{mapup-task}(\text{taskid}, \text{os-waiting-input-handler-path6}(os))$
 $= \text{mapup-task}(\text{taskid}, os))$

EVENT: Disable mapup-task-os-waiting-input-handler-path6.

THEOREM: mapup-tasks-os-waiting-input-handler-path6
 $(\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge (\text{taskid} \in \mathbf{N})))$
 $\rightarrow (\text{mapup-tasks}(\text{taskid}, \text{os-waiting-input-handler-path6}(os))$
 $= \text{mapup-tasks}(\text{taskid}, os))$

EVENT: Disable mapup-tasks-os-waiting-input-handler-path6.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path6
 $(\text{good-os}(os) \wedge \text{tm-waiting}(os))$
 $\rightarrow (\text{mapup-os-tasks}(\text{os-waiting-input-handler-path6}(os))$
 $= \text{mapup-os-tasks}(os))$

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path6.

THEOREM: mapup-os-ibuffers-os-waiting-input-handler-path6
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge (\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $os))))))$
 $\rightarrow (\text{mapup-os-ibuffers}(\text{os-waiting-input-handler-path6}(os))$
 $= \text{enq-ith-buffer}(\text{tm-ichar}(\text{getnth}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\text{tm-iports}(os))),$
 $\text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\text{mapup-os-ibuffers}(os))$

EVENT: Disable mapup-os-ibuffers-os-waiting-input-handler-path6.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path6

$\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-waiting-input-handler-path6 } (os)))$
 $= \text{mapup-os-obuffers } (os)$

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path6.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path6
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-waiting-input-handler-path6 } (os)))$
 $= \text{mapup-os-mbuffers } (os)$

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path6.

THEOREM: tm-rwstate-os-waiting-input-handler-path6
 $\text{tm-rwstate } (\text{os-waiting-input-handler-path6 } (os)) = '1$

EVENT: Disable tm-rwstate-os-waiting-input-handler-path6.

THEOREM: tm-clock-os-waiting-input-handler-path6
 $\text{tm-clock } (\text{os-waiting-input-handler-path6 } (os)) = \text{tm-clock } (os)$

EVENT: Disable tm-clock-os-waiting-input-handler-path6.

THEOREM: tm-iports-os-waiting-input-handler-path6
 $\text{tm-iports } (\text{os-waiting-input-handler-path6 } (os))$
 $= \text{tm-clear-input-interrupt } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)),$
 $\text{tm-iports } (os))$

EVENT: Disable tm-iports-os-waiting-input-handler-path6.

THEOREM: tm-oports-os-waiting-input-handler-path6
 $\text{tm-oports } (\text{os-waiting-input-handler-path6 } (os)) = \text{tm-oports } (os)$

EVENT: Disable tm-oports-os-waiting-input-handler-path6.

THEOREM: correctness-of-os-waiting-input-handler-path6
 $(\text{good-os } (os))$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)))$
 $\wedge (\text{tm-waiting } (os))$
 $\wedge ((\text{getnth } ('2$
 $\quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)),$
 $\quad \text{os-status-table } (os))$

$$\begin{aligned}
& \neq '5) \\
& \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \hspace{10em} os))) \\
& \quad \wedge (\neg \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \hspace{10em} \text{tm-iports}(os)))))) \\
\rightarrow & (\text{mapup-os}(\text{os-waiting-input-handler-path6}(os)) \\
& = \text{ak-waiting-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \hspace{10em} \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-waiting-input-handler-path6.

THEOREM: correctness-of-os-waiting-input-handler

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-os}(\text{os-waiting-input-handler}(os)) \\
& = \text{ak-waiting-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \hspace{10em} \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-waiting-input-handler.

THEOREM: mapup-base-os-running-input-handler-path1

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-base}(taskid, \text{os-running-input-handler-path1}(os)) \\
& = \text{mapup-base}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-base-os-running-input-handler-path1.

THEOREM: mapup-limit-os-running-input-handler-path1

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-limit}(taskid, \text{os-running-input-handler-path1}(os)) \\
& = \text{mapup-limit}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-limit-os-running-input-handler-path1.

THEOREM: getseg-tm-memory-os-running-input-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \wedge (taskid < '16)) \\
\rightarrow & (\text{getseg}(\text{mapup-base}(taskid, os), \\
& \quad \text{mapup-limit}(taskid, os), \\
& \quad \text{tm-memory}(\text{os-running-input-handler-path1}(os))) \\
& = \text{getseg}(\text{mapup-base}(taskid, os), \\
& \quad \text{mapup-limit}(taskid, os), \\
& \quad \text{tm-memory}(os)))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-running-input-handler-path1.

THEOREM: tm-cpu-os-running-input-handler-path1-crock
 $\text{tm-cpu}(\text{os-running-input-handler-path1}(os)) = \text{tm-cpu}(os)$

EVENT: Disable tm-cpu-os-running-input-handler-path1-crock.

THEOREM: tm-cpu-os-running-input-handler-path1
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge (\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '5))))$
 $\rightarrow (\text{tm-cpu}(\text{os-running-input-handler-path1}(os))$
 $\quad = \text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path1}(os)),$
 $\quad \text{table}('9,$
 $\quad \text{os-task-table}(\text{os-running-input-handler-path1}(os))))))$

EVENT: Disable tm-cpu-os-running-input-handler-path1.

THEOREM: mapup-cpu-os-running-input-handler-path1
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '5)$
 $\wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16))))))$
 $\rightarrow (\text{mapup-cpu}(\text{taskid}, \text{os-running-input-handler-path1}(os))$
 $\quad = \text{mapup-cpu}(\text{taskid}, os))$

EVENT: Disable mapup-cpu-os-running-input-handler-path1.

THEOREM: mapup-regs-os-running-input-handler-path1
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$

$$\begin{aligned}
& \text{os-status-table}(os) \\
& = '5 \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-regs}(taskid, \text{os-running-input-handler-path1}(os)) \\
& = \text{mapup-regs}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-running-input-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-cc-os-running-input-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-running-input-handler-path1}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-running-input-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-error-os-running-input-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-running-input-handler-path1}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-running-input-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-svcflag-os-running-input-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os))
\end{aligned}$$

$$\begin{aligned}
&= '5) \\
&\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcflag}(taskid, \text{os-running-input-handler-path1}(os)) \\
&= \text{mapup-svcflag}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path1.

$$\begin{aligned}
&\text{THEOREM: mapup-svcid-os-running-input-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcid}(taskid, \text{os-running-input-handler-path1}(os)) \\
&= \text{mapup-svcid}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-input-handler-path1.

$$\begin{aligned}
&\text{THEOREM: mapup-task-os-running-input-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-task}(taskid, \text{os-running-input-handler-path1}(os)) \\
&= \text{mapup-task}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-task-os-running-input-handler-path1.

$$\begin{aligned}
&\text{THEOREM: mapup-tasks-os-running-input-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5)
\end{aligned}$$

$$\begin{aligned} & \wedge (taskid \in \mathbf{N})))))) \\ \rightarrow & \text{ (mapup-tasks (taskid, os-running-input-handler-path1 (os))} \\ & = \text{ mapup-tasks (taskid, os))} \end{aligned}$$

EVENT: Disable mapup-tasks-os-running-input-handler-path1.

$$\begin{aligned} \text{THEOREM: mapup-os-tasks-os-running-input-handler-path1} \\ & \text{(good-os (os)} \\ & \wedge ((\neg \text{tm-waiting (os)} \\ & \wedge (\text{tm-some-input-interruptp (tm-iports (os))} \\ & \wedge (\text{getnth ('2} \\ & \qquad \qquad \qquad * \text{tm-interrupting-input-port (tm-iports (os)),} \\ & \qquad \qquad \qquad \text{os-status-table (os))} \\ & \qquad \qquad \qquad = '5)))))) \\ \rightarrow & \text{ (mapup-os-tasks (os-running-input-handler-path1 (os))} \\ & = \text{ mapup-os-tasks (os))} \end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-input-handler-path1.

$$\begin{aligned} \text{THEOREM: mapup-os-ibuffers-os-running-input-handler-path1} \\ & \text{(good-os (os)} \\ & \wedge (\text{tm-some-input-interruptp (tm-iports (os))} \\ & \wedge \text{array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),} \\ & \qquad \qquad \qquad \text{os))})) \\ \rightarrow & \text{ (mapup-os-ibuffers (os-running-input-handler-path1 (os))} \\ & = \text{qreplace-ith-buffer (tm-overflow-char (tm-ichar (getnth (tm-interrupting-input-port (tm-iports (os)),} \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{tm-iports (os))}),} \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{tm-interrupting-input-port (tm-iports (os)),} \\ & \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \qquad \text{mapup-os-ibuffers (os))})} \end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-input-handler-path1.

$$\begin{aligned} \text{THEOREM: mapup-os-obuffers-os-running-input-handler-path1} \\ & \text{good-os (os)} \\ \rightarrow & \text{ (mapup-os-obuffers (os-running-input-handler-path1 (os))} \\ & = \text{ mapup-os-obuffers (os))} \end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path1.

$$\begin{aligned} \text{THEOREM: mapup-os-mbuffers-os-running-input-handler-path1} \\ & \text{good-os (os)} \\ \rightarrow & \text{ (mapup-os-mbuffers (os-running-input-handler-path1 (os))} \\ & = \text{ mapup-os-mbuffers (os))} \end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path1.

THEOREM: mapup-queue-os-readyq-os-running-input-handler-path1
(good-os (*os*)
 \wedge (\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge (getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 = '5))))
 \rightarrow (mapup-queue (os-readyq (os-running-input-handler-path1 (*os*)))
 = enq (tm-interrupting-input-port (tm-iports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-running-input-handler-path1.

THEOREM: mapup-os-status-table-os-running-input-handler-path1
(good-os (*os*) \wedge tm-some-input-interruptp (tm-iports (*os*)))
 \rightarrow (table ('2, os-status-table (os-running-input-handler-path1 (*os*)))
 = putnth ('(0 0),
 tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-running-input-handler-path1.

THEOREM: tm-rwstate-os-running-input-handler-path1
(good-tm (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-rwstate (os-running-input-handler-path1 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-input-handler-path1.

THEOREM: tm-clock-os-running-input-handler-path1
tm-clock (os-running-input-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-input-handler-path1.

THEOREM: tm-iports-os-running-input-handler-path1
tm-iports (os-running-input-handler-path1 (*os*))
 = tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
 tm-iports (*os*))

EVENT: Disable tm-iports-os-running-input-handler-path1.

THEOREM: tm-oports-os-running-input-handler-path1
 $\text{tm-oports}(\text{os-running-input-handler-path1}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-running-input-handler-path1.

THEOREM: correctness-of-os-running-input-handler-path1
 $(\text{good-os}(os) \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{getnth}('2 * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{os-status-table}(os)) = '5) \wedge \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os), os)))))))$
 $\rightarrow (\text{mapup-os}(\text{os-running-input-handler-path1}(os)) = \text{ak-running-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-running-input-handler-path1.

THEOREM: mapup-base-os-running-input-handler-path2
 $\text{good-os}(os) \rightarrow (\text{mapup-base}(taskid, \text{os-running-input-handler-path2}(os)) = \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-running-input-handler-path2.

THEOREM: mapup-limit-os-running-input-handler-path2
 $\text{good-os}(os) \rightarrow (\text{mapup-limit}(taskid, \text{os-running-input-handler-path2}(os)) = \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-running-input-handler-path2.

THEOREM: getseg-tm-memory-os-running-input-handler-path2
 $(\text{good-os}(os) \wedge (taskid < '16)) \rightarrow (\text{getseg}(\text{mapup-base}(taskid, os), \text{mapup-limit}(taskid, os), \text{tm-memory}(\text{os-running-input-handler-path2}(os))) = \text{getseg}(\text{mapup-base}(taskid, os), \text{mapup-limit}(taskid, os), \text{tm-memory}(os)))$

EVENT: Disable getseg-tm-memory-os-running-input-handler-path2.

THEOREM: tm-cpu-os-running-input-handler-path2-crock
 $\text{tm-cpu}(\text{os-running-input-handler-path2}(os)) = \text{tm-cpu}(os)$

EVENT: Disable tm-cpu-os-running-input-handler-path2-crock.

THEOREM: tm-cpu-os-running-input-handler-path2
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge (\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '5))))$
 $\rightarrow (\text{tm-cpu}(\text{os-running-input-handler-path2}(os))$
 $\quad = \text{getnth}(\text{os-current-taskid}(\text{os-running-input-handler-path2}(os)),$
 $\quad \text{table}('9,$
 $\quad \text{os-task-table}(\text{os-running-input-handler-path2}(os))))))$

EVENT: Disable tm-cpu-os-running-input-handler-path2.

THEOREM: mapup-cpu-os-running-input-handler-path2
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '5)$
 $\wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16))))))$
 $\rightarrow (\text{mapup-cpu}(\text{taskid}, \text{os-running-input-handler-path2}(os))$
 $\quad = \text{mapup-cpu}(\text{taskid}, os))$

EVENT: Disable mapup-cpu-os-running-input-handler-path2.

THEOREM: mapup-regs-os-running-input-handler-path2
(good-os(os))
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),$

$$\begin{aligned}
& \text{os-status-table}(os) \\
& = '5 \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-regs}(taskid, \text{os-running-input-handler-path2}(os)) \\
& = \text{mapup-regs}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-running-input-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-cc-os-running-input-handler-path2} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-running-input-handler-path2}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-running-input-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-error-os-running-input-handler-path2} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-running-input-handler-path2}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-running-input-handler-path2.

$$\begin{aligned}
\text{THEOREM: mapup-svcflag-os-running-input-handler-path2} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os))
\end{aligned}$$

$$\begin{aligned}
&= '5) \\
&\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcflag}(taskid, \text{os-running-input-handler-path2}(os)) \\
&= \text{mapup-svcflag}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path2.

THEOREM: mapup-svcid-os-running-input-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcid}(taskid, \text{os-running-input-handler-path2}(os)) \\
&= \text{mapup-svcid}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-input-handler-path2.

THEOREM: mapup-task-os-running-input-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-task}(taskid, \text{os-running-input-handler-path2}(os)) \\
&= \text{mapup-task}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-task-os-running-input-handler-path2.

THEOREM: mapup-tasks-os-running-input-handler-path2

$$\begin{aligned}
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '5)
\end{aligned}$$

$$\begin{aligned} & \wedge (taskid \in \mathbf{N})))))) \\ \rightarrow & (\text{mapup-tasks}(taskid, \text{os-running-input-handler-path2}(os)) \\ & = \text{mapup-tasks}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-tasks-os-running-input-handler-path2.

$$\begin{aligned} \text{THEOREM: mapup-os-tasks-os-running-input-handler-path2} \\ & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\ & \wedge (\text{getnth}('2 \\ & \qquad \qquad \qquad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\ & \qquad \qquad \qquad \text{os-status-table}(os)) \\ & \qquad \qquad \qquad = '5)))))) \\ \rightarrow & (\text{mapup-os-tasks}(\text{os-running-input-handler-path2}(os)) \\ & = \text{mapup-os-tasks}(os)) \end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-input-handler-path2.

$$\begin{aligned} \text{THEOREM: mapup-os-ibuffers-os-running-input-handler-path2} \\ & (\text{good-os}(os) \\ & \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\ & \wedge (\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\ & \qquad \qquad \qquad os)))))) \\ \rightarrow & (\text{mapup-os-ibuffers}(\text{os-running-input-handler-path2}(os)) \\ & = \text{enq-ith-buffer}(\text{tm-overflow-char}(\text{tm-ichar}(\text{getnth}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\ & \qquad \qquad \qquad \text{tm-iports}(os))), \\ & \qquad \qquad \qquad \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\ & \qquad \qquad \qquad \text{mapup-os-ibuffers}(os))) \end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-input-handler-path2.

$$\begin{aligned} \text{THEOREM: mapup-os-obuffers-os-running-input-handler-path2} \\ & \text{good-os}(os) \\ \rightarrow & (\text{mapup-os-obuffers}(\text{os-running-input-handler-path2}(os)) \\ & = \text{mapup-os-obuffers}(os)) \end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path2.

$$\begin{aligned} \text{THEOREM: mapup-os-mbuffers-os-running-input-handler-path2} \\ & \text{good-os}(os) \\ \rightarrow & (\text{mapup-os-mbuffers}(\text{os-running-input-handler-path2}(os)) \\ & = \text{mapup-os-mbuffers}(os)) \end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path2.

THEOREM: mapup-queue-os-readyq-os-running-input-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ (getnth ('2
 * tm-interrupting-input-port (tm-iports (*os*)),
 os-status-table (*os*))
 = '5))))))
→ (mapup-queue (os-readyq (os-running-input-handler-path2 (*os*)))
 = enq (tm-interrupting-input-port (tm-iports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-running-input-handler-path2.

THEOREM: mapup-os-status-table-os-running-input-handler-path2
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (table ('2, os-status-table (os-running-input-handler-path2 (*os*)))
 = putnth ('(0 0),
 tm-interrupting-input-port (tm-iports (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-running-input-handler-path2.

THEOREM: tm-rwstate-os-running-input-handler-path2
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-input-handler-path2 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-input-handler-path2.

THEOREM: tm-clock-os-running-input-handler-path2
tm-clock (os-running-input-handler-path2 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-input-handler-path2.

THEOREM: tm-iports-os-running-input-handler-path2
tm-iports (os-running-input-handler-path2 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
 tm-iports (*os*))

EVENT: Disable tm-iports-os-running-input-handler-path2.

THEOREM: tm-oports-os-running-input-handler-path2
 $\text{tm-oports}(\text{os-running-input-handler-path2}(os)) = \text{tm-oports}(os)$

EVENT: Disable tm-oports-os-running-input-handler-path2.

THEOREM: correctness-of-os-running-input-handler-path2
 $(\text{good-os}(os) \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge ((\neg \text{tm-waiting}(os)) \wedge ((\text{getnth}('2 * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{os-status-table}(os)) = '5) \wedge ((\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), os))) \wedge \text{tm-iport-errorp}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{tm-iports}(os))))))))))$
 $\rightarrow (\text{mapup-os}(\text{os-running-input-handler-path2}(os)) = \text{ak-running-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-running-input-handler-path2.

THEOREM: mapup-base-os-running-input-handler-path3
 $\text{good-os}(os) \rightarrow (\text{mapup-base}(taskid, \text{os-running-input-handler-path3}(os)) = \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-running-input-handler-path3.

THEOREM: mapup-limit-os-running-input-handler-path3
 $\text{good-os}(os) \rightarrow (\text{mapup-limit}(taskid, \text{os-running-input-handler-path3}(os)) = \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-running-input-handler-path3.

THEOREM: getseg-tm-memory-os-running-input-handler-path3
 $(\text{good-os}(os) \wedge (taskid < '16)) \rightarrow (\text{getseg}(\text{mapup-base}(taskid, os), \text{mapup-limit}(taskid, os), \text{tm-memory}(\text{os-running-input-handler-path3}(os))) = \text{getseg}(\text{mapup-base}(taskid, os),$

mapup-limit (*taskid*, *os*),
tm-memory (*os*))

EVENT: Disable getseg-tm-memory-os-running-input-handler-path3.

THEOREM: tm-cpu-os-running-input-handler-path3-crock
tm-cpu (os-running-input-handler-path3 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path3-crock.

THEOREM: tm-cpu-os-running-input-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge (getnth ('2
 \quad * tm-interrupting-input-port (tm-iports (*os*)),
 \quad os-status-table (*os*)
 \quad = '5))))))
 \rightarrow (tm-cpu (os-running-input-handler-path3 (*os*)
 $=$ getnth (os-current-taskid (os-running-input-handler-path3 (*os*)),
 \quad table ('9,
 \quad os-task-table (os-running-input-handler-path3 (*os*))))))

EVENT: Disable tm-cpu-os-running-input-handler-path3.

THEOREM: mapup-cpu-os-running-input-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((getnth ('2
 \quad * tm-interrupting-input-port (tm-iports (*os*)),
 \quad os-status-table (*os*)
 \quad = '5)
 \quad \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16)))))))))
 \rightarrow (mapup-cpu (*taskid*, os-running-input-handler-path3 (*os*)
 $=$ mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-input-handler-path3.

THEOREM: mapup-regs-os-running-input-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*))

$$\begin{aligned}
& \wedge ((\text{getnth } '2 \\
& \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \text{os-status-table } (os)) \\
& \quad = '5) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-regs } (taskid, \text{os-running-input-handler-path3 } (os)) \\
& = \text{mapup-regs } (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-running-input-handler-path3.

THEOREM: mapup-cc-os-running-input-handler-path3

$$\begin{aligned}
& (\text{good-os } (os) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\
& \quad \quad \quad \wedge ((\text{getnth } '2 \\
& \quad \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad \quad = '5) \\
& \quad \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-cc } (taskid, \text{os-running-input-handler-path3 } (os)) \\
& = \text{mapup-cc } (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-running-input-handler-path3.

THEOREM: mapup-error-os-running-input-handler-path3

$$\begin{aligned}
& (\text{good-os } (os) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\
& \quad \quad \quad \wedge ((\text{getnth } '2 \\
& \quad \quad \quad \quad * \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\
& \quad \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad \quad = '5) \\
& \quad \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-error } (taskid, \text{os-running-input-handler-path3 } (os)) \\
& = \text{mapup-error } (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-running-input-handler-path3.

THEOREM: mapup-svcflag-os-running-input-handler-path3

$$\begin{aligned}
& (\text{good-os } (os) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\
& \quad \quad \quad \wedge ((\text{getnth } '2
\end{aligned}$$

$$\begin{aligned}
& * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \text{os-status-table}(os) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-svcid-os-running-input-handler-path3} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-task-os-running-input-handler-path3} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \text{os-status-table}(os)) \\
& = '5) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-tasks-os-running-input-handler-path3} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& * \text{tm-interrupting-input-port}(\text{tm-iports}(os)),
\end{aligned}$$

$$\begin{aligned}
& \text{os-status-table}(os) \\
& = '5) \\
& \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-running-input-handler-path3} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge (\text{getnth}('2 \\
& \quad * \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '5)))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-ibuffers-os-running-input-handler-path3} \\
& (\text{good-os}(os) \\
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \wedge (\neg \text{array-qfullp}(\text{os-current-ibuffer}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad os)))))) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-running-input-handler-path3}(os)) \\
& = \text{enq-ith-buffer}(\text{tm-ichar}(\text{getnth}(\text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{tm-iports}(os))), \\
& \quad \text{tm-interrupting-input-port}(\text{tm-iports}(os)), \\
& \quad \text{mapup-os-ibuffers}(os)))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-obuffers-os-running-input-handler-path3} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-running-input-handler-path3}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-mbuffers-os-running-input-handler-path3} \\
& \text{good-os}(os)
\end{aligned}$$

→ (mapup-os-mbuffers (os-running-input-handler-path3 (os))
= mapup-os-mbuffers (os))

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path3.

THEOREM: mapup-queue-os-readyq-os-running-input-handler-path3
(good-os (os)
∧ ((¬ tm-waiting (os))
∧ (tm-some-input-interruptp (tm-iports (os))
∧ (getnth ('2
* tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
= '5))))))
→ (mapup-queue (os-readyq (os-running-input-handler-path3 (os)))
= enq (tm-interrupting-input-port (tm-iports (os)),
mapup-queue (os-readyq (os))))

EVENT: Disable mapup-queue-os-readyq-os-running-input-handler-path3.

THEOREM: mapup-os-status-table-os-running-input-handler-path3
(good-os (os) ∧ tm-some-input-interruptp (tm-iports (os)))
→ (table ('2, os-status-table (os-running-input-handler-path3 (os)))
= putnth ('(0 0),
tm-interrupting-input-port (tm-iports (os)),
table ('2, os-status-table (os))))

EVENT: Disable mapup-os-status-table-os-running-input-handler-path3.

THEOREM: tm-rwstate-os-running-input-handler-path3
(good-tm (os) ∧ (¬ tm-waiting (os)))
→ (tm-rwstate (os-running-input-handler-path3 (os)) = tm-rwstate (os))

EVENT: Disable tm-rwstate-os-running-input-handler-path3.

THEOREM: tm-clock-os-running-input-handler-path3
tm-clock (os-running-input-handler-path3 (os)) = tm-clock (os)

EVENT: Disable tm-clock-os-running-input-handler-path3.

THEOREM: tm-iports-os-running-input-handler-path3
tm-iports (os-running-input-handler-path3 (os))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (os)),
tm-iports (os))

$$\begin{aligned}
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (\text{os-running-input-handler-path4 } (os))) \\
= & \text{getseg } (\text{mapup-base } (taskid, os), \\
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (os))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-running-input-handler-path4.

THEOREM: tm-cpu-os-running-input-handler-path4-crock
 $\text{tm-cpu } (\text{os-running-input-handler-path4 } (os)) = \text{tm-cpu } (os)$

EVENT: Disable tm-cpu-os-running-input-handler-path4-crock.

THEOREM: tm-cpu-os-running-input-handler-path4
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge \text{tm-some-input-interruptp } (\text{tm-iports } (os)))$
 $\rightarrow (\text{tm-cpu } (\text{os-running-input-handler-path4 } (os)))$
 $= \text{getnth } (\text{os-current-taskid } (\text{os-running-input-handler-path4 } (os)),$
 $\text{table } ('9,$
 $\text{os-task-table } (\text{os-running-input-handler-path4 } (os))))$

EVENT: Disable tm-cpu-os-running-input-handler-path4.

THEOREM: mapup-cpu-os-running-input-handler-path4
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))$
 $\wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu } (taskid, \text{os-running-input-handler-path4 } (os)))$
 $= \text{mapup-cpu } (taskid, os)$

EVENT: Disable mapup-cpu-os-running-input-handler-path4.

THEOREM: mapup-regs-os-running-input-handler-path4
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))$
 $\wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs } (taskid, \text{os-running-input-handler-path4 } (os)))$
 $= \text{mapup-regs } (taskid, os)$

EVENT: Disable mapup-regs-os-running-input-handler-path4.

THEOREM: mapup-cc-os-running-input-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-running-input-handler-path4 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-input-handler-path4.

THEOREM: mapup-error-os-running-input-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-running-input-handler-path4 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-input-handler-path4.

THEOREM: mapup-svcflag-os-running-input-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcflag (*taskid*, os-running-input-handler-path4 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-running-input-handler-path4.

THEOREM: mapup-svcid-os-running-input-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-input-interruptp (tm-iports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcid (*taskid*, os-running-input-handler-path4 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-running-input-handler-path4.

THEOREM: mapup-task-os-running-input-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

THEOREM: mapup-os-mbuffers-os-running-input-handler-path4
good-os (*os*)
→ (mapup-os-mbuffers (os-running-input-handler-path4 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path4.

THEOREM: tm-rwstate-os-running-input-handler-path4
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-input-handler-path4 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-input-handler-path4.

THEOREM: tm-clock-os-running-input-handler-path4
tm-clock (os-running-input-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-input-handler-path4.

THEOREM: tm-iports-os-running-input-handler-path4
tm-iports (os-running-input-handler-path4 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))

EVENT: Disable tm-iports-os-running-input-handler-path4.

THEOREM: tm-oports-os-running-input-handler-path4
tm-oports (os-running-input-handler-path4 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-running-input-handler-path4.

THEOREM: correctness-of-os-running-input-handler-path4
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((¬ tm-waiting (*os*))
∧ ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*)
≠ '5)
∧ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
→ (mapup-os (os-running-input-handler-path4 (*os*))
= ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler-path4.

THEOREM: mapup-base-os-running-input-handler-path5
good-os (*os*)
→ (mapup-base (*taskid*, os-running-input-handler-path5 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-input-handler-path5.

THEOREM: mapup-limit-os-running-input-handler-path5
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-input-handler-path5 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-input-handler-path5.

THEOREM: getseg-tm-memory-os-running-input-handler-path5
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-input-handler-path5 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-input-handler-path5.

THEOREM: tm-cpu-os-running-input-handler-path5-crock
tm-cpu (os-running-input-handler-path5 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path5-crock.

THEOREM: tm-cpu-os-running-input-handler-path5
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (tm-cpu (os-running-input-handler-path5 (*os*))
= getnth (os-current-taskid (os-running-input-handler-path5 (*os*)),
table ('9,
os-task-table (os-running-input-handler-path5 (*os*))))))

EVENT: Disable tm-cpu-os-running-input-handler-path5.

THEOREM: mapup-cpu-os-running-input-handler-path5
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cpu (*taskid*, os-running-input-handler-path5 (*os*))
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-input-handler-path5.

THEOREM: mapup-regs-os-running-input-handler-path5
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-regs (*taskid*, os-running-input-handler-path5 (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-input-handler-path5.

THEOREM: mapup-cc-os-running-input-handler-path5
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-running-input-handler-path5 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-input-handler-path5.

THEOREM: mapup-error-os-running-input-handler-path5
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-running-input-handler-path5 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-input-handler-path5.

THEOREM: mapup-svcflag-os-running-input-handler-path5
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-running-input-handler-path5}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path5.

$$\begin{aligned}
\text{THEOREM: mapup-svcid-os-running-input-handler-path5} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-running-input-handler-path5}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-input-handler-path5.

$$\begin{aligned}
\text{THEOREM: mapup-task-os-running-input-handler-path5} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-running-input-handler-path5}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-running-input-handler-path5.

$$\begin{aligned}
\text{THEOREM: mapup-tasks-os-running-input-handler-path5} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-input-interruptp}(\text{tm-iports}(os)) \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-running-input-handler-path5}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-running-input-handler-path5.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-running-input-handler-path5} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-running-input-handler-path5}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-input-handler-path5.

THEOREM: mapup-os-ibuffers-os-running-input-handler-path5

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os)) \\ & \quad \wedge (\neg \text{array-qfullp } (\text{os-current-ibuffer } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad os)))))) \\ \rightarrow & (\text{mapup-os-ibuffers } (\text{os-running-input-handler-path5 } (os)) \\ & = \text{enq-ith-buffer } (\text{tm-overflow-char } (\text{tm-ichar } (\text{getnth } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os))), \\ & \quad \quad \quad \text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{mapup-os-ibuffers } (os))) \end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-input-handler-path5.

THEOREM: mapup-os-obuffers-os-running-input-handler-path5

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-os-obuffers } (\text{os-running-input-handler-path5 } (os)) \\ & = \text{mapup-os-obuffers } (os)) \end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path5.

THEOREM: mapup-os-mbuffers-os-running-input-handler-path5

$$\begin{aligned} & \text{good-os } (os) \\ \rightarrow & (\text{mapup-os-mbuffers } (\text{os-running-input-handler-path5 } (os)) \\ & = \text{mapup-os-mbuffers } (os)) \end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path5.

THEOREM: tm-rwstate-os-running-input-handler-path5

$$\begin{aligned} & (\text{good-tm } (os) \wedge (\neg \text{tm-waiting } (os))) \\ \rightarrow & (\text{tm-rwstate } (\text{os-running-input-handler-path5 } (os)) = \text{tm-rwstate } (os)) \end{aligned}$$

EVENT: Disable tm-rwstate-os-running-input-handler-path5.

THEOREM: tm-clock-os-running-input-handler-path5

$$\text{tm-clock } (\text{os-running-input-handler-path5 } (os)) = \text{tm-clock } (os)$$

EVENT: Disable tm-clock-os-running-input-handler-path5.

THEOREM: tm-iports-os-running-input-handler-path5

$$\begin{aligned} & \text{tm-iports } (\text{os-running-input-handler-path5 } (os)) \\ = & \text{tm-clear-input-interrupt } (\text{tm-interrupting-input-port } (\text{tm-iports } (os)), \\ & \quad \quad \quad \text{tm-iports } (os)) \end{aligned}$$

EVENT: Disable tm-iports-os-running-input-handler-path5.

THEOREM: tm-oports-os-running-input-handler-path5
tm-oports (os-running-input-handler-path5 (os)) = tm-oports (os)

EVENT: Disable tm-oports-os-running-input-handler-path5.

THEOREM: correctness-of-os-running-input-handler-path5
(good-os (os)
 ∧ (tm-some-input-interruptp (tm-iports (os))
 ∧ ((¬ tm-waiting (os))
 ∧ ((getnth ('2
 * tm-interrupting-input-port (tm-iports (os)),
 os-status-table (os))
 ≠ '5)
 ∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
 os)))
 ∧ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
 tm-iports (os)))))))))
→ (mapup-os (os-running-input-handler-path5 (os))
 = ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (os)),
 mapup-os (os)))

EVENT: Disable correctness-of-os-running-input-handler-path5.

THEOREM: mapup-base-os-running-input-handler-path6
good-os (os)
→ (mapup-base (taskid, os-running-input-handler-path6 (os))
 = mapup-base (taskid, os))

EVENT: Disable mapup-base-os-running-input-handler-path6.

THEOREM: mapup-limit-os-running-input-handler-path6
good-os (os)
→ (mapup-limit (taskid, os-running-input-handler-path6 (os))
 = mapup-limit (taskid, os))

EVENT: Disable mapup-limit-os-running-input-handler-path6.

THEOREM: getseg-tm-memory-os-running-input-handler-path6
(good-os (os) ∧ (taskid < '16))
→ (getseg (mapup-base (taskid, os),

$$\begin{aligned}
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (os\text{-running-input-handler-path6 } (os))) \\
= & \text{getseg } (\text{mapup-base } (taskid, os), \\
& \text{mapup-limit } (taskid, os), \\
& \text{tm-memory } (os))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-running-input-handler-path6.

THEOREM: tm-cpu-os-running-input-handler-path6-crock
 $\text{tm-cpu } (os\text{-running-input-handler-path6 } (os)) = \text{tm-cpu } (os)$

EVENT: Disable tm-cpu-os-running-input-handler-path6-crock.

THEOREM: tm-cpu-os-running-input-handler-path6
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os)) \wedge \text{tm-some-input-interruptp } (\text{tm-iports } (os)))$
 $\rightarrow (\text{tm-cpu } (os\text{-running-input-handler-path6 } (os)))$
 $= \text{getnth } (os\text{-current-taskid } (os\text{-running-input-handler-path6 } (os)),$
 $\text{table } ('9,$
 $\text{os-task-table } (os\text{-running-input-handler-path6 } (os))))$

EVENT: Disable tm-cpu-os-running-input-handler-path6.

THEOREM: mapup-cpu-os-running-input-handler-path6
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))$
 $\wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-cpu } (taskid, os\text{-running-input-handler-path6 } (os)))$
 $= \text{mapup-cpu } (taskid, os)$

EVENT: Disable mapup-cpu-os-running-input-handler-path6.

THEOREM: mapup-regs-os-running-input-handler-path6
 $(\text{good-os } (os))$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\wedge (\text{tm-some-input-interruptp } (\text{tm-iports } (os))$
 $\wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16))))$
 $\rightarrow (\text{mapup-regs } (taskid, os\text{-running-input-handler-path6 } (os)))$
 $= \text{mapup-regs } (taskid, os)$

EVENT: Disable mapup-regs-os-running-input-handler-path6.

THEOREM: mapup-cc-os-running-input-handler-path6
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-running-input-handler-path6 (*os*)
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-input-handler-path6.

THEOREM: mapup-error-os-running-input-handler-path6
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-running-input-handler-path6 (*os*)
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-input-handler-path6.

THEOREM: mapup-svcflag-os-running-input-handler-path6
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcflag (*taskid*, os-running-input-handler-path6 (*os*)
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-running-input-handler-path6.

THEOREM: mapup-svcid-os-running-input-handler-path6
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-input-interruptp (tm-iports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-svcid (*taskid*, os-running-input-handler-path6 (*os*)
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-running-input-handler-path6.

THEOREM: mapup-task-os-running-input-handler-path6
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

THEOREM: mapup-os-mbuffers-os-running-input-handler-path6
good-os (*os*)
→ (mapup-os-mbuffers (os-running-input-handler-path6 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path6.

THEOREM: tm-rwstate-os-running-input-handler-path6
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-input-handler-path6 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-input-handler-path6.

THEOREM: tm-clock-os-running-input-handler-path6
tm-clock (os-running-input-handler-path6 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-input-handler-path6.

THEOREM: tm-iports-os-running-input-handler-path6
tm-iports (os-running-input-handler-path6 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))

EVENT: Disable tm-iports-os-running-input-handler-path6.

THEOREM: tm-oports-os-running-input-handler-path6
tm-oports (os-running-input-handler-path6 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-running-input-handler-path6.

THEOREM: correctness-of-os-running-input-handler-path6
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((¬ tm-waiting (*os*))
∧ ((getnth ('2
* tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*)
≠ '5)
∧ ((¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))
∧ (¬ tm-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))))))))))

$$\begin{aligned} \rightarrow & \text{ (mapup-os (os-running-input-handler-path6 (os))} \\ & = \text{ ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (os)),} \\ & \text{ mapup-os (os))} \end{aligned}$$

EVENT: Disable correctness-of-os-running-input-handler-path6.

THEOREM: correctness-of-os-running-input-handler

$$\begin{aligned} & \text{(good-os (os)} \\ & \wedge \text{ (tm-some-input-interruptp (tm-iports (os)) \wedge (\neg \text{tm-waiting (os)}))} \\ \rightarrow & \text{ (mapup-os (os-running-input-handler (os))} \\ & = \text{ ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (os)),} \\ & \text{ mapup-os (os))} \end{aligned}$$

EVENT: Disable correctness-of-os-running-input-handler.

THEOREM: mapup-base-os-waiting-output-handler-path1

$$\begin{aligned} & \text{good-os (os)} \\ \rightarrow & \text{ (mapup-base (taskid, os-waiting-output-handler-path1 (os))} \\ & = \text{ mapup-base (taskid, os)} \end{aligned}$$

EVENT: Disable mapup-base-os-waiting-output-handler-path1.

THEOREM: mapup-limit-os-waiting-output-handler-path1

$$\begin{aligned} & \text{good-os (os)} \\ \rightarrow & \text{ (mapup-limit (taskid, os-waiting-output-handler-path1 (os))} \\ & = \text{ mapup-limit (taskid, os)} \end{aligned}$$

EVENT: Disable mapup-limit-os-waiting-output-handler-path1.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path1

$$\begin{aligned} & \text{(good-os (os) \wedge (taskid < '16))} \\ \rightarrow & \text{ (getseg (mapup-base (taskid, os),} \\ & \text{ mapup-limit (taskid, os),} \\ & \text{ tm-memory (os-waiting-output-handler-path1 (os))} \\ = & \text{ getseg (mapup-base (taskid, os),} \\ & \text{ mapup-limit (taskid, os),} \\ & \text{ tm-memory (os))} \end{aligned}$$

EVENT: Disable getseg-tm-memory-os-waiting-output-handler-path1.

```

(PROVE-LEMMA
  TM-CPU-OS-WAITING-OUTPUT-HANDLER-PATH1-CROCK
  (REWRITE)
  (EQUAL
    (TM-CPU (OS-WAITING-OUTPUT-HANDLER-PATH1 OS))
    (CONS
      (GETNTH '0
        (GETSEG '0
          '8
          (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
            (TABLE '9 (OS-TASK-TABLE OS))))))
      (CONS
        (GETNTH '1
          (GETSEG '0
            '8
            (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
              (TABLE '9 (OS-TASK-TABLE OS))))))
        (CONS
          (GETNTH '2
            (GETSEG '0
              '8
              (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                (TABLE '9 (OS-TASK-TABLE OS))))))
          (CONS
            (GETNTH '3
              (GETSEG '0
                '8
                (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                  (TABLE '9 (OS-TASK-TABLE OS))))))
            (CONS
              (GETNTH '4
                (GETSEG '0
                  '8
                  (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                    (TABLE '9 (OS-TASK-TABLE OS))))))
              (CONS
                (GETNTH '5
                  (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
                      (TABLE '9 (OS-TASK-TABLE OS))))))
                (CONS
                  (GETNTH '6
                    (GETSEG '0

```

```

'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))
(CONS
(GETNTH '7
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(TM-PACK-PSW
(TM-UNPACK-CC
(GETNTH
(PPLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS)))
(TM-UNPACK-ERROR
(GETNTH
(PPLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS)))
(TM-UNPACK-SVCFLAG
(GETNTH
(PPLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS)))
(TM-UNPACK-SVCID
(GETNTH
(PPLUS '8
(TIMES '9
(TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
(OS-TASK-TABLE OS))))
'NIL)))))))))
((ENABLE OS-WAITING-OUTPUT-HANDLER-PATH1 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
TM-R5 TM-R6 TM-R7 TM-UNPACK-CC TM-UNPACK-ERROR TM-UNPACK-SVCFLAG
TM-UNPACK-SVCID)
(ENABLE-THEORY TM-SHELLS GROUND-ZERO)
(DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-waiting-output-handler-path1-crock.

THEOREM: tm-cpu-os-waiting-output-handler-path1
 (good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*))))
 → (tm-cpu (os-waiting-output-handler-path1 (*os*))
 = getnth (os-current-taskid (os-waiting-output-handler-path1 (*os*)),
 table ('9,
 os-task-table (os-waiting-output-handler-path1 (*os*))))))

EVENT: Disable tm-cpu-os-waiting-output-handler-path1.

THEOREM: mapup-cpu-os-waiting-output-handler-path1
 (good-os (*os*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-cpu (*taskid*, os-waiting-output-handler-path1 (*os*))
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-output-handler-path1.

THEOREM: mapup-regs-os-waiting-output-handler-path1
 (good-os (*os*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-regs (*taskid*, os-waiting-output-handler-path1 (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-output-handler-path1.

THEOREM: mapup-cc-os-waiting-output-handler-path1
 (good-os (*os*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
 → (mapup-cc (*taskid*, os-waiting-output-handler-path1 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-output-handler-path1.

THEOREM: mapup-error-os-waiting-output-handler-path1
 (good-os (*os*)
 ∧ (tm-waiting (*os*))

$$\begin{aligned}
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-waiting-output-handler-path1}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-waiting-output-handler-path1.

THEOREM: mapup-svcflag-os-waiting-output-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-waiting-output-handler-path1}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path1.

THEOREM: mapup-svcid-os-waiting-output-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-waiting-output-handler-path1}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-waiting-output-handler-path1.

THEOREM: mapup-task-os-waiting-output-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-waiting-output-handler-path1}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-waiting-output-handler-path1.

THEOREM: mapup-tasks-os-waiting-output-handler-path1

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-waiting}(os) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-waiting-output-handler-path1}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-waiting-output-handler-path1.

THEOREM: mapup-os-tasks-os-waiting-output-handler-path1
(good-os (*os*)
 \wedge (tm-waiting (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*))))
 \rightarrow (mapup-os-tasks (os-waiting-output-handler-path1 (*os*))
 = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path1.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-ibuffers (os-waiting-output-handler-path1 (*os*))
 = mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path1.

THEOREM: mapup-os-obuffers-os-waiting-output-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-obuffers (os-waiting-output-handler-path1 (*os*))
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path1.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path1
good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-waiting-output-handler-path1 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path1.

THEOREM: mapup-queue-os-readyq-os-waiting-output-handler-path1
(good-os (*os*) \wedge tm-waiting (*os*))
 \rightarrow (mapup-queue (os-readyq (os-waiting-output-handler-path1 (*os*)))
 = enq (tm-interrupting-output-port (tm-oports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-waiting-output-handler-path1.

THEOREM: mapup-os-status-table-os-waiting-output-handler-path1
(good-os (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*)))
 \rightarrow (table ('2, os-status-table (os-waiting-output-handler-path1 (*os*)))

$$= \text{putnth}('0\ 0), \\ \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ \text{table}('2, \text{os-status-table}(os)))$$

EVENT: Disable mapup-os-status-table-os-waiting-output-handler-path1.

THEOREM: tm-rwstate-os-waiting-output-handler-path1
 $\text{tm-rwstate}(\text{os-waiting-output-handler-path1}(os)) = '0$

EVENT: Disable tm-rwstate-os-waiting-output-handler-path1.

THEOREM: tm-clock-os-waiting-output-handler-path1
 $\text{tm-clock}(\text{os-waiting-output-handler-path1}(os)) = '1000$

EVENT: Disable tm-clock-os-waiting-output-handler-path1.

THEOREM: tm-iports-os-waiting-output-handler-path1
 $\text{tm-iports}(\text{os-waiting-output-handler-path1}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-waiting-output-handler-path1.

THEOREM: tm-oports-os-waiting-output-handler-path1
 $\text{tm-oports}(\text{os-waiting-output-handler-path1}(os)) \\ = \text{tm-clear-output-interrupt}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ \text{tm-oports}(os))$

EVENT: Disable tm-oports-os-waiting-output-handler-path1.

THEOREM: ak-waiting-to-outputp-mapup-os
 $(\text{good-os}(os) \\ \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ \wedge (\text{getnth}('2 * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ \text{os-status-table}(os)) \\ = '4))) \\ \rightarrow \text{ak-waiting-to-outputp}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ \text{mapup-os}(os))$

EVENT: Disable ak-waiting-to-outputp-mapup-os.

THEOREM: correctness-of-os-waiting-output-handler-path1
 $(\text{good-os}(os) \\ \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$


```

(EQUAL
(TM-CPU (OS-WAITING-OUTPUT-HANDLER-PATH2 OS))
(CONS
(GETNTH '0
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '1
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '2
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '3
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '4
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '5
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
(GETNTH '6
(GETSEG '0
'8
(GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
(TABLE '9 (OS-TASK-TABLE OS))))))

```

```

(CONS
  (GETNTH '7
    (GETSEG '0
      '8
      (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))
        (TABLE '9 (OS-TASK-TABLE OS))))))
(CONS
  (TM-PACK-PSW
    (TM-UNPACK-CC
      (GETNTH
        (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-ERROR
      (GETNTH
        (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCFLAG
      (GETNTH
        (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
        (OS-TASK-TABLE OS)))
    (TM-UNPACK-SVCID
      (GETNTH
        (PLUS '8
          (TIMES '9
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPOINTS OS))))
        (OS-TASK-TABLE OS))))
  'NIL)))))))))
((ENABLE OS-WAITING-OUTPUT-HANDLER-PATH2 TM-CPU TM-R0 TM-R1 TM-R2 TM-R3 TM-R4
  TM-R5 TM-R6 TM-R7 TM-UNPACK-CC TM-UNPACK-ERROR TM-UNPACK-SVCFLAG
  TM-UNPACK-SVCID)
  (ENABLE-THEORY TM-SHELLS GROUND-ZERO)
  (DISABLE-THEORY T))

```

EVENT: Disable tm-cpu-os-waiting-output-handler-path2-crock.

THEOREM: tm-cpu-os-waiting-output-handler-path2

(good-os (*os*)
 \wedge (tm-waiting (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*))))

$$\begin{aligned} &\rightarrow (\text{tm-cpu}(\text{os-waiting-output-handler-path2}(os)) \\ &= \text{getnth}(\text{os-current-taskid}(\text{os-waiting-output-handler-path2}(os)), \\ &\quad \text{table}('9, \\ &\quad \quad \text{os-task-table}(\text{os-waiting-output-handler-path2}(os)))))) \end{aligned}$$

EVENT: Disable tm-cpu-os-waiting-output-handler-path2.

THEOREM: mapup-cpu-os-waiting-output-handler-path2

$$\begin{aligned} &(\text{good-os}(os) \\ &\quad \wedge (\text{tm-waiting}(os) \\ &\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ &\quad \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-cpu}(\text{taskid}, \text{os-waiting-output-handler-path2}(os)) \\ &= \text{mapup-cpu}(\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-cpu-os-waiting-output-handler-path2.

THEOREM: mapup-regs-os-waiting-output-handler-path2

$$\begin{aligned} &(\text{good-os}(os) \\ &\quad \wedge (\text{tm-waiting}(os) \\ &\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ &\quad \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-regs}(\text{taskid}, \text{os-waiting-output-handler-path2}(os)) \\ &= \text{mapup-regs}(\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-regs-os-waiting-output-handler-path2.

THEOREM: mapup-cc-os-waiting-output-handler-path2

$$\begin{aligned} &(\text{good-os}(os) \\ &\quad \wedge (\text{tm-waiting}(os) \\ &\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ &\quad \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-cc}(\text{taskid}, \text{os-waiting-output-handler-path2}(os)) \\ &= \text{mapup-cc}(\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-cc-os-waiting-output-handler-path2.

THEOREM: mapup-error-os-waiting-output-handler-path2

$$\begin{aligned} &(\text{good-os}(os) \\ &\quad \wedge (\text{tm-waiting}(os) \\ &\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ &\quad \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ &\rightarrow (\text{mapup-error}(\text{taskid}, \text{os-waiting-output-handler-path2}(os)) \\ &= \text{mapup-error}(\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-error-os-waiting-output-handler-path2.

THEOREM: mapup-svcflag-os-waiting-output-handler-path2
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcflag (*taskid*, os-waiting-output-handler-path2 (*os*)
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path2.

THEOREM: mapup-svcid-os-waiting-output-handler-path2
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-svcid (*taskid*, os-waiting-output-handler-path2 (*os*)
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-output-handler-path2.

THEOREM: mapup-task-os-waiting-output-handler-path2
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-task (*taskid*, os-waiting-output-handler-path2 (*os*)
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-output-handler-path2.

THEOREM: mapup-tasks-os-waiting-output-handler-path2
(good-os (*os*)
 \wedge (tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*) \wedge (*taskid* \in \mathbf{N}))))))
→ (mapup-tasks (*taskid*, os-waiting-output-handler-path2 (*os*)
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-output-handler-path2.

THEOREM: mapup-os-tasks-os-waiting-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-waiting } (os) \wedge \text{tm-some-output-interruptp } (\text{tm-oports } (os))) \\
\rightarrow & (\text{mapup-os-tasks } (\text{os-waiting-output-handler-path2 } (os))) \\
& = \text{mapup-os-tasks } (os)
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path2.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path2

$$\begin{aligned}
& \text{good-os } (os) \\
\rightarrow & (\text{mapup-os-ibuffers } (\text{os-waiting-output-handler-path2 } (os))) \\
& = \text{mapup-os-ibuffers } (os)
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path2.

THEOREM: mapup-os-obuffers-os-waiting-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad os)))))) \\
\rightarrow & (\text{mapup-os-obuffers } (\text{os-waiting-output-handler-path2 } (os))) \\
& = \text{deq-ith-buffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{mapup-os-obuffers } (os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path2.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path2

$$\begin{aligned}
& \text{good-os } (os) \\
\rightarrow & (\text{mapup-os-mbuffers } (\text{os-waiting-output-handler-path2 } (os))) \\
& = \text{mapup-os-mbuffers } (os)
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path2.

THEOREM: mapup-queue-os-readyq-os-waiting-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os) \wedge \text{tm-waiting } (os)) \\
\rightarrow & (\text{mapup-queue } (\text{os-readyq } (\text{os-waiting-output-handler-path2 } (os)))) \\
& = \text{enq } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{mapup-queue } (\text{os-readyq } (os)))
\end{aligned}$$

EVENT: Disable mapup-queue-os-readyq-os-waiting-output-handler-path2.

THEOREM: mapup-os-status-table-os-waiting-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os) \wedge \text{tm-some-output-interruptp } (\text{tm-oports } (os)))
\end{aligned}$$

\rightarrow (table ('2, os-status-table (os-waiting-output-handler-path2 (os)))
 $=$ putnth ('0 0),
 tm-interrupting-output-port (tm-oports (os)),
 table ('2, os-status-table (os)))

EVENT: Disable mapup-os-status-table-os-waiting-output-handler-path2.

THEOREM: tm-rwstate-os-waiting-output-handler-path2
 tm-rwstate (os-waiting-output-handler-path2 (os)) = '0

EVENT: Disable tm-rwstate-os-waiting-output-handler-path2.

THEOREM: tm-clock-os-waiting-output-handler-path2
 tm-clock (os-waiting-output-handler-path2 (os)) = '1000

EVENT: Disable tm-clock-os-waiting-output-handler-path2.

THEOREM: tm-iports-os-waiting-output-handler-path2
 tm-iports (os-waiting-output-handler-path2 (os)) = tm-iports (os)

EVENT: Disable tm-iports-os-waiting-output-handler-path2.

THEOREM: tm-oports-os-waiting-output-handler-path2
 (good-os (os)
 \wedge (tm-some-output-interruptp (tm-oports (os))
 \wedge (\neg array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
 os))))))
 \rightarrow (tm-oports (os-waiting-output-handler-path2 (os))
 $=$ tm-start-output (qfirst (getnth (tm-interrupting-output-port (tm-oports (os)),
 mapup-os-obuffers (os)),
 tm-interrupting-output-port (tm-oports (os)),
 tm-oports (os)))

EVENT: Disable tm-oports-os-waiting-output-handler-path2.

THEOREM: correctness-of-os-waiting-output-handler-path2
 (good-os (os)
 \wedge (tm-some-output-interruptp (tm-oports (os))
 \wedge (tm-waiting (os)
 \wedge ((getnth ('2
 * tm-interrupting-output-port (tm-oports (os)),
 os-status-table (os))

$$\begin{aligned}
&= '4) \\
&\wedge (\neg \text{array-qemptyp}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oport}(\text{os})), \\
&\hspace{15em} \text{os})))))) \\
\rightarrow &(\text{mapup-os}(\text{os-waiting-output-handler-path2}(\text{os})) \\
&= \text{ak-waiting-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{tm-oport}(\text{os})), \\
&\hspace{10em} \text{mapup-os}(\text{os}))
\end{aligned}$$

EVENT: Disable correctness-of-os-waiting-output-handler-path2.

THEOREM: mapup-base-os-waiting-output-handler-path3

$$\begin{aligned}
&\text{good-os}(\text{os}) \\
\rightarrow &(\text{mapup-base}(\text{taskid}, \text{os-waiting-output-handler-path3}(\text{os})) \\
&= \text{mapup-base}(\text{taskid}, \text{os}))
\end{aligned}$$

EVENT: Disable mapup-base-os-waiting-output-handler-path3.

THEOREM: mapup-limit-os-waiting-output-handler-path3

$$\begin{aligned}
&\text{good-os}(\text{os}) \\
\rightarrow &(\text{mapup-limit}(\text{taskid}, \text{os-waiting-output-handler-path3}(\text{os})) \\
&= \text{mapup-limit}(\text{taskid}, \text{os}))
\end{aligned}$$

EVENT: Disable mapup-limit-os-waiting-output-handler-path3.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path3

$$\begin{aligned}
&(\text{good-os}(\text{os}) \wedge (\text{taskid} < '16)) \\
\rightarrow &(\text{getseg}(\text{mapup-base}(\text{taskid}, \text{os}), \\
&\hspace{10em} \text{mapup-limit}(\text{taskid}, \text{os}), \\
&\hspace{10em} \text{tm-memory}(\text{os-waiting-output-handler-path3}(\text{os}))) \\
&= \text{getseg}(\text{mapup-base}(\text{taskid}, \text{os}), \\
&\hspace{10em} \text{mapup-limit}(\text{taskid}, \text{os}), \\
&\hspace{10em} \text{tm-memory}(\text{os}))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-waiting-output-handler-path3.

THEOREM: mapup-cpu-os-waiting-output-handler-path3

$$\begin{aligned}
&(\text{good-os}(\text{os}) \wedge (\text{tm-waiting}(\text{os}) \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))) \\
\rightarrow &(\text{mapup-cpu}(\text{taskid}, \text{os-waiting-output-handler-path3}(\text{os})) \\
&= \text{mapup-cpu}(\text{taskid}, \text{os}))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-waiting-output-handler-path3.

THEOREM: mapup-regs-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-regs}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-regs}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-waiting-output-handler-path3.

THEOREM: mapup-cc-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-waiting-output-handler-path3.

THEOREM: mapup-error-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-waiting-output-handler-path3.

THEOREM: mapup-svcflag-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path3.

THEOREM: mapup-svcid-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-waiting-output-handler-path3.

THEOREM: mapup-task-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-waiting-output-handler-path3.

THEOREM: mapup-tasks-os-waiting-output-handler-path3

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge (\text{taskid} \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(\text{taskid}, \text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-tasks}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-waiting-output-handler-path3} \\
& (\text{good-os}(os) \wedge \text{tm-waiting}(os)) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path3} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-ibuffers}(\text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-obuffers-os-waiting-output-handler-path3} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-obuffers}(\text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-os-obuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path3} \\
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-os-mbuffers}(\text{os-waiting-output-handler-path3}(os)) \\
& = \text{mapup-os-mbuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: tm-rwstate-os-waiting-output-handler-path3} \\
& \text{tm-rwstate}(\text{os-waiting-output-handler-path3}(os)) = '1
\end{aligned}$$

EVENT: Disable tm-rwstate-os-waiting-output-handler-path3.

$$\begin{aligned}
\text{THEOREM: tm-clock-os-waiting-output-handler-path3} \\
& \text{tm-clock}(\text{os-waiting-output-handler-path3}(os)) = \text{tm-clock}(os)
\end{aligned}$$

EVENT: Disable tm-clock-os-waiting-output-handler-path3.

THEOREM: tm-iports-os-waiting-output-handler-path3
 $\text{tm-iports}(\text{os-waiting-output-handler-path3}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-waiting-output-handler-path3.

THEOREM: tm-oports-os-waiting-output-handler-path3
 $\text{tm-oports}(\text{os-waiting-output-handler-path3}(os))$
 $= \text{tm-clear-output-interrupt}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{tm-oports}(os))$

EVENT: Disable tm-oports-os-waiting-output-handler-path3.

THEOREM: not-ak-waiting-to-outputp-mapup-os
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\wedge (\text{getnth}('2 * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{os-status-table}(os))$
 $\neq '4)))$
 $\rightarrow (\neg \text{ak-waiting-to-outputp}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{mapup-os}(os)))$

EVENT: Disable not-ak-waiting-to-outputp-mapup-os.

THEOREM: correctness-of-os-waiting-output-handler-path3
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\wedge (\text{tm-waiting}(os)$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad \neq '4)$
 $\wedge \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os}))))))$
 $\rightarrow (\text{mapup-os}(\text{os-waiting-output-handler-path3}(os))$
 $= \text{ak-waiting-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-waiting-output-handler-path3.

THEOREM: mapup-base-os-waiting-output-handler-path4
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(\text{taskid}, \text{os-waiting-output-handler-path4}(os))$
 $= \text{mapup-base}(\text{taskid}, os))$

EVENT: Disable mapup-base-os-waiting-output-handler-path4.

THEOREM: mapup-limit-os-waiting-output-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-output-handler-path4.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-output-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-output-handler-path4.

THEOREM: mapup-cpu-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cpu (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-output-handler-path4.

THEOREM: mapup-regs-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-output-handler-path4.

THEOREM: mapup-cc-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-output-handler-path4.

THEOREM: mapup-error-os-waiting-output-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-error}(taskid, \text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-error}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-error-os-waiting-output-handler-path4.

$$\begin{aligned}
& \text{THEOREM: mapup-svcflag-os-waiting-output-handler-path4} \\
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path4.

$$\begin{aligned}
& \text{THEOREM: mapup-svcid-os-waiting-output-handler-path4} \\
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-waiting-output-handler-path4.

$$\begin{aligned}
& \text{THEOREM: mapup-task-os-waiting-output-handler-path4} \\
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-waiting-output-handler-path4.

$$\begin{aligned}
& \text{THEOREM: mapup-tasks-os-waiting-output-handler-path4} \\
& (\text{good-os}(os) \wedge (\text{tm-waiting}(os) \wedge (taskid \in \mathbf{N}))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-waiting-output-handler-path4.

$$\begin{aligned}
& \text{THEOREM: mapup-os-tasks-os-waiting-output-handler-path4} \\
& (\text{good-os}(os) \wedge \text{tm-waiting}(os)) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-waiting-output-handler-path4}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path4.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path4

$\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-ibuffers } (\text{os-waiting-output-handler-path4 } (os)))$
 $= \text{mapup-os-ibuffers } (os)$

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path4.

THEOREM: mapup-os-obuffers-os-waiting-output-handler-path4
 $(\text{good-os } (os)$
 $\wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))$
 $\wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os))),$
 $os))))))$
 $\rightarrow (\text{mapup-os-obuffers } (\text{os-waiting-output-handler-path4 } (os)))$
 $= \text{deq-ith-buffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)),$
 $\text{mapup-os-obuffers } (os))$

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path4.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path4
 $\text{good-os } (os)$
 $\rightarrow (\text{mapup-os-mbuffers } (\text{os-waiting-output-handler-path4 } (os)))$
 $= \text{mapup-os-mbuffers } (os)$

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path4.

THEOREM: tm-rwstate-os-waiting-output-handler-path4
 $\text{tm-rwstate } (\text{os-waiting-output-handler-path4 } (os)) = '1$

EVENT: Disable tm-rwstate-os-waiting-output-handler-path4.

THEOREM: tm-clock-os-waiting-output-handler-path4
 $\text{tm-clock } (\text{os-waiting-output-handler-path4 } (os)) = \text{tm-clock } (os)$

EVENT: Disable tm-clock-os-waiting-output-handler-path4.

THEOREM: tm-iports-os-waiting-output-handler-path4
 $\text{tm-iports } (\text{os-waiting-output-handler-path4 } (os)) = \text{tm-iports } (os)$

EVENT: Disable tm-iports-os-waiting-output-handler-path4.

THEOREM: tm-oports-os-waiting-output-handler-path4
 $(\text{good-os } (os)$
 $\wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))$

$$\begin{aligned}
& \wedge (\neg \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} os)))))) \\
\rightarrow & (\text{tm-oports}(\text{os-waiting-output-handler-path4}(os)) \\
& = \text{tm-start-output}(\text{qfirst}(\text{getnth}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} \text{mapup-os-obuffers}(os))), \\
& \hspace{10em} \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} \text{tm-oports}(os)))
\end{aligned}$$

EVENT: Disable tm-oports-os-waiting-output-handler-path4.

THEOREM: correctness-of-os-waiting-output-handler-path4

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \wedge (\text{tm-waiting}(os) \\
& \wedge ((\text{getnth}('2 \\
& \hspace{10em} * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} \text{os-status-table}(os)) \\
& \neq '4) \\
& \wedge (\neg \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} os))))))))) \\
\rightarrow & (\text{mapup-os}(\text{os-waiting-output-handler-path4}(os)) \\
& = \text{ak-waiting-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-waiting-output-handler-path4.

THEOREM: correctness-of-os-waiting-output-handler

$$\begin{aligned}
& (\text{good-os}(os) \\
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \wedge \text{tm-waiting}(os))) \\
\rightarrow & (\text{mapup-os}(\text{os-waiting-output-handler}(os)) \\
& = \text{ak-waiting-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \hspace{10em} \text{mapup-os}(os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-waiting-output-handler.

THEOREM: mapup-base-os-running-output-handler-path1

$$\begin{aligned}
& \text{good-os}(os) \\
\rightarrow & (\text{mapup-base}(\text{taskid}, \text{os-running-output-handler-path1}(os)) \\
& = \text{mapup-base}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-base-os-running-output-handler-path1.

THEOREM: mapup-limit-os-running-output-handler-path1

$\text{good-os } (os)$
 $\rightarrow (\text{mapup-limit } (taskid, \text{os-running-output-handler-path1 } (os)))$
 $= \text{mapup-limit } (taskid, os)$

EVENT: Disable mapup-limit-os-running-output-handler-path1.

THEOREM: $\text{getseg-tm-memory-os-running-output-handler-path1}$
 $(\text{good-os } (os) \wedge (taskid < '16))$
 $\rightarrow (\text{getseg } (\text{mapup-base } (taskid, os),$
 $\quad \text{mapup-limit } (taskid, os),$
 $\quad \text{tm-memory } (\text{os-running-output-handler-path1 } (os))))$
 $= \text{getseg } (\text{mapup-base } (taskid, os),$
 $\quad \text{mapup-limit } (taskid, os),$
 $\quad \text{tm-memory } (os))$

EVENT: Disable getseg-tm-memory-os-running-output-handler-path1.

THEOREM: $\text{tm-cpu-os-running-output-handler-path1-crock}$
 $\text{tm-cpu } (\text{os-running-output-handler-path1 } (os)) = \text{tm-cpu } (os)$

EVENT: Disable tm-cpu-os-running-output-handler-path1-crock.

THEOREM: $\text{tm-cpu-os-running-output-handler-path1}$
 $(\text{good-os } (os)$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))$
 $\quad \quad \wedge (\text{getnth } ('2$
 $\quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)),$
 $\quad \quad \quad \text{os-status-table } (os))$
 $\quad \quad = '4))))$
 $\rightarrow (\text{tm-cpu } (\text{os-running-output-handler-path1 } (os))$
 $= \text{getnth } (\text{os-current-taskid } (\text{os-running-output-handler-path1 } (os)),$
 $\quad \text{table } ('9,$
 $\quad \quad \text{os-task-table } (\text{os-running-output-handler-path1 } (os))))$

EVENT: Disable tm-cpu-os-running-output-handler-path1.

THEOREM: $\text{mapup-cpu-os-running-output-handler-path1}$
 $(\text{good-os } (os)$
 $\wedge ((\neg \text{tm-waiting } (os))$
 $\quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os))$
 $\quad \quad \wedge ((\text{getnth } ('2$
 $\quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)),$

$$\begin{aligned}
& \text{os-status-table}(os) \\
& = '4) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-cpu}(taskid, \text{os-running-output-handler-path1}(os)) \\
& = \text{mapup-cpu}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-regs-os-running-output-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '4) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-regs}(taskid, \text{os-running-output-handler-path1}(os)) \\
& = \text{mapup-regs}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-regs-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-cc-os-running-output-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \text{os-status-table}(os)) \\
& = '4) \\
& \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-cc}(taskid, \text{os-running-output-handler-path1}(os)) \\
& = \text{mapup-cc}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cc-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-error-os-running-output-handler-path1} \\
& (\text{good-os}(os) \\
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \wedge ((\text{getnth}('2 \\
& \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \text{os-status-table}(os))
\end{aligned}$$

$$\begin{aligned}
&= '4) \\
&\quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-error}(taskid, \text{os-running-output-handler-path1}(os)) \\
&= \text{mapup-error}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-error-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-svcflag-os-running-output-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '4) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcflag}(taskid, \text{os-running-output-handler-path1}(os)) \\
&= \text{mapup-svcflag}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-svcid-os-running-output-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '4) \\
&\quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & \text{mapup-svcid}(taskid, \text{os-running-output-handler-path1}(os)) \\
&= \text{mapup-svcid}(taskid, os)
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-output-handler-path1.

$$\begin{aligned}
\text{THEOREM: mapup-task-os-running-output-handler-path1} \\
&(\text{good-os}(os) \\
&\quad \wedge ((\neg \text{tm-waiting}(os) \\
&\quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
&\quad \quad \quad \wedge ((\text{getnth}('2 \\
&\quad \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
&\quad \quad \quad \quad \text{os-status-table}(os)) \\
&\quad \quad \quad = '4)
\end{aligned}$$

$$\begin{aligned} & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\ \rightarrow & \text{mapup-task}(taskid, \text{os-running-output-handler-path1}(os)) \\ & = \text{mapup-task}(taskid, os) \end{aligned}$$

EVENT: Disable mapup-task-os-running-output-handler-path1.

$$\begin{aligned} & \text{THEOREM: mapup-tasks-os-running-output-handler-path1} \\ & \text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \quad \quad \wedge ((\text{getnth}('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ & \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad = '4) \\ & \quad \quad \wedge (taskid \in \mathbf{N})))))) \\ \rightarrow & \text{mapup-tasks}(taskid, \text{os-running-output-handler-path1}(os)) \\ & = \text{mapup-tasks}(taskid, os) \end{aligned}$$

EVENT: Disable mapup-tasks-os-running-output-handler-path1.

$$\begin{aligned} & \text{THEOREM: mapup-os-tasks-os-running-output-handler-path1} \\ & \text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \quad \quad \wedge (\text{getnth}('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\ & \quad \quad \quad \text{os-status-table}(os)) \\ & \quad \quad = '4)))) \\ \rightarrow & \text{mapup-os-tasks}(\text{os-running-output-handler-path1}(os)) \\ & = \text{mapup-os-tasks}(os) \end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-output-handler-path1.

$$\begin{aligned} & \text{THEOREM: mapup-os-ibuffers-os-running-output-handler-path1} \\ & \text{good-os}(os) \\ \rightarrow & \text{mapup-os-ibuffers}(\text{os-running-output-handler-path1}(os)) \\ & = \text{mapup-os-ibuffers}(os) \end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path1.

$$\begin{aligned} & \text{THEOREM: mapup-os-obuffers-os-running-output-handler-path1} \\ & \text{good-os}(os) \\ \rightarrow & \text{mapup-os-obuffers}(\text{os-running-output-handler-path1}(os)) \\ & = \text{mapup-os-obuffers}(os) \end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-output-handler-path1.

THEOREM: mapup-os-mbuffers-os-running-output-handler-path1
good-os (*os*)
→ (mapup-os-mbuffers (os-running-output-handler-path1 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-output-handler-path1.

THEOREM: mapup-queue-os-readyq-os-running-output-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
= '4))))))
→ (mapup-queue (os-readyq (os-running-output-handler-path1 (*os*)))
= enq (tm-interrupting-output-port (tm-oports (*os*)),
mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-running-output-handler-path1.

THEOREM: mapup-os-status-table-os-running-output-handler-path1
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (table ('2, os-status-table (os-running-output-handler-path1 (*os*)))
= putnth ('(0 0),
tm-interrupting-output-port (tm-oports (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-running-output-handler-path1.

THEOREM: tm-rwstate-os-running-output-handler-path1
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-output-handler-path1 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-output-handler-path1.

THEOREM: tm-clock-os-running-output-handler-path1
tm-clock (os-running-output-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-output-handler-path1.

THEOREM: tm-iports-os-running-output-handler-path1
 $\text{tm-iports}(\text{os-running-output-handler-path1}(os)) = \text{tm-iports}(os)$

EVENT: Disable tm-iports-os-running-output-handler-path1.

THEOREM: tm-oports-os-running-output-handler-path1
 $\text{tm-oports}(\text{os-running-output-handler-path1}(os))$
 $= \text{tm-clear-output-interrupt}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\text{tm-oports}(os))$

EVENT: Disable tm-oports-os-running-output-handler-path1.

THEOREM: correctness-of-os-running-output-handler-path1
 $(\text{good-os}(os)$
 $\wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os))$
 $\wedge ((\neg \text{tm-waiting}(os))$
 $\wedge ((\text{getnth}('2$
 $\quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os-status-table}(os))$
 $\quad = '4)$
 $\wedge \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{os))))))$
 $\rightarrow (\text{mapup-os}(\text{os-running-output-handler-path1}(os))$
 $= \text{ak-running-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)),$
 $\quad \text{mapup-os}(os)))$

EVENT: Disable correctness-of-os-running-output-handler-path1.

THEOREM: mapup-base-os-running-output-handler-path2
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-base}(taskid, \text{os-running-output-handler-path2}(os))$
 $= \text{mapup-base}(taskid, os))$

EVENT: Disable mapup-base-os-running-output-handler-path2.

THEOREM: mapup-limit-os-running-output-handler-path2
 $\text{good-os}(os)$
 $\rightarrow (\text{mapup-limit}(taskid, \text{os-running-output-handler-path2}(os))$
 $= \text{mapup-limit}(taskid, os))$

EVENT: Disable mapup-limit-os-running-output-handler-path2.

THEOREM: getseg-tm-memory-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os) \wedge (\text{taskid} < '16)) \\
\rightarrow & (\text{getseg } (\text{mapup-base } (taskid, os), \\
& \quad \text{mapup-limit } (taskid, os), \\
& \quad \text{tm-memory } (\text{os-running-output-handler-path2 } (os))) \\
= & \text{getseg } (\text{mapup-base } (taskid, os), \\
& \quad \text{mapup-limit } (taskid, os), \\
& \quad \text{tm-memory } (os))
\end{aligned}$$

EVENT: Disable getseg-tm-memory-os-running-output-handler-path2.

THEOREM: tm-cpu-os-running-output-handler-path2-crock
 $\text{tm-cpu } (\text{os-running-output-handler-path2 } (os)) = \text{tm-cpu } (os)$

EVENT: Disable tm-cpu-os-running-output-handler-path2-crock.

$$\begin{aligned}
& \text{THEOREM: tm-cpu-os-running-output-handler-path2} \\
& (\text{good-os } (os) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \quad \wedge (\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad = '4)))) \\
\rightarrow & (\text{tm-cpu } (\text{os-running-output-handler-path2 } (os)) \\
= & \text{getnth } (\text{os-current-taskid } (\text{os-running-output-handler-path2 } (os)), \\
& \quad \text{table } ('9, \\
& \quad \quad \text{os-task-table } (\text{os-running-output-handler-path2 } (os))))))
\end{aligned}$$

EVENT: Disable tm-cpu-os-running-output-handler-path2.

$$\begin{aligned}
& \text{THEOREM: mapup-cpu-os-running-output-handler-path2} \\
& (\text{good-os } (os) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad = '4) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
\rightarrow & (\text{mapup-cpu } (taskid, \text{os-running-output-handler-path2 } (os)) \\
= & \text{mapup-cpu } (taskid, os))
\end{aligned}$$

EVENT: Disable mapup-cpu-os-running-output-handler-path2.

THEOREM: mapup-regs-os-running-output-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad \quad = '4) \\ & \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-regs } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\ & = \text{mapup-regs } (\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-regs-os-running-output-handler-path2.

THEOREM: mapup-cc-os-running-output-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad \quad = '4) \\ & \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-cc } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\ & = \text{mapup-cc } (\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-cc-os-running-output-handler-path2.

THEOREM: mapup-error-os-running-output-handler-path2

$$\begin{aligned} & (\text{good-os } (os)) \\ & \wedge ((\neg \text{tm-waiting } (os)) \\ & \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\ & \quad \quad \wedge ((\text{getnth } ('2 \\ & \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\ & \quad \quad \quad \text{os-status-table } (os)) \\ & \quad \quad \quad = '4) \\ & \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\ \rightarrow & (\text{mapup-error } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\ & = \text{mapup-error } (\text{taskid}, os)) \end{aligned}$$

EVENT: Disable mapup-error-os-running-output-handler-path2.

THEOREM: mapup-svcflag-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '4) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
& \rightarrow (\text{mapup-svcflag } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\
& \quad = \text{mapup-svcflag } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-output-handler-path2.

THEOREM: mapup-svcid-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '4) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
& \rightarrow (\text{mapup-svcid } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\
& \quad = \text{mapup-svcid } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-output-handler-path2.

THEOREM: mapup-task-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad = '4) \\
& \quad \quad \wedge ((\text{taskid} \in \mathbf{N}) \wedge (\text{taskid} < '16)))))) \\
& \rightarrow (\text{mapup-task } (\text{taskid}, \text{os-running-output-handler-path2 } (os)) \\
& \quad = \text{mapup-task } (\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-running-output-handler-path2.

THEOREM: mapup-tasks-os-running-output-handler-path2

$$(\text{good-os } (os))$$

$$\begin{aligned}
& \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \wedge ((\text{getnth}('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad = '4) \\
& \quad \quad \wedge (\text{taskid} \in \mathbf{N})))))) \\
& \rightarrow (\text{mapup-tasks}(\text{taskid}, \text{os-running-output-handler-path2}(os)) \\
& \quad = \text{mapup-tasks}(\text{taskid}, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-running-output-handler-path2.

THEOREM: mapup-os-tasks-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \quad \wedge (\text{getnth}('2 \\
& \quad \quad \quad \quad * \text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \quad \quad \quad \text{os-status-table}(os)) \\
& \quad \quad \quad \quad = '4)))))) \\
& \rightarrow (\text{mapup-os-tasks}(\text{os-running-output-handler-path2}(os)) \\
& \quad = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-output-handler-path2.

THEOREM: mapup-os-ibuffers-os-running-output-handler-path2

$$\begin{aligned}
& \text{good-os}(os) \\
& \rightarrow (\text{mapup-os-ibuffers}(\text{os-running-output-handler-path2}(os)) \\
& \quad = \text{mapup-os-ibuffers}(os))
\end{aligned}$$

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path2.

THEOREM: mapup-os-obuffers-os-running-output-handler-path2

$$\begin{aligned}
& (\text{good-os}(os) \\
& \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \wedge (\neg \text{array-qempty}(\text{os-current-obuffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \quad \quad \text{os})))))) \\
& \rightarrow (\text{mapup-os-obuffers}(\text{os-running-output-handler-path2}(os)) \\
& \quad = \text{deq-ith-buffer}(\text{tm-interrupting-output-port}(\text{tm-oports}(os)), \\
& \quad \quad \text{mapup-os-obuffers}(os)))
\end{aligned}$$

EVENT: Disable mapup-os-obuffers-os-running-output-handler-path2.

THEOREM: mapup-os-mbuffers-os-running-output-handler-path2
 good-os (*os*)
 \rightarrow (mapup-os-mbuffers (os-running-output-handler-path2 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-output-handler-path2.

THEOREM: mapup-queue-os-readyq-os-running-output-handler-path2
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge (getnth ('2
 \quad * tm-interrupting-output-port (tm-oports (*os*)),
 \quad os-status-table (*os*))
 \quad = '4))))))
 \rightarrow (mapup-queue (os-readyq (os-running-output-handler-path2 (*os*)))
 = enq (tm-interrupting-output-port (tm-oports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-running-output-handler-path2.

THEOREM: mapup-os-status-table-os-running-output-handler-path2
 (good-os (*os*) \wedge tm-some-output-interruptp (tm-oports (*os*)))
 \rightarrow (table ('2, os-status-table (os-running-output-handler-path2 (*os*)))
 = putnth ('(0 0),
 tm-interrupting-output-port (tm-oports (*os*)),
 table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-running-output-handler-path2.

THEOREM: tm-rwstate-os-running-output-handler-path2
 (good-tm (*os*) \wedge (\neg tm-waiting (*os*)))
 \rightarrow (tm-rwstate (os-running-output-handler-path2 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-output-handler-path2.

THEOREM: tm-clock-os-running-output-handler-path2
 tm-clock (os-running-output-handler-path2 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-output-handler-path2.

THEOREM: tm-iports-os-running-output-handler-path2
 tm-iports (os-running-output-handler-path2 (*os*)) = tm-iports (*os*)

EVENT: Disable mapup-limit-os-running-output-handler-path3.

THEOREM: getseg-tm-memory-os-running-output-handler-path3
(good-os(*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-output-handler-path3 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-output-handler-path3.

THEOREM: tm-cpu-os-running-output-handler-path3-crock
tm-cpu (os-running-output-handler-path3 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-output-handler-path3-crock.

THEOREM: tm-cpu-os-running-output-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ tm-some-output-interruptp (tm-oports (*os*))))
→ (tm-cpu (os-running-output-handler-path3 (*os*))
= getnth (os-current-taskid (os-running-output-handler-path3 (*os*)),
table ('9,
os-task-table (os-running-output-handler-path3 (*os*))))))

EVENT: Disable tm-cpu-os-running-output-handler-path3.

THEOREM: mapup-cpu-os-running-output-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cpu (*taskid*, os-running-output-handler-path3 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-output-handler-path3.

THEOREM: mapup-regs-os-running-output-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-output-interruptp (tm-oports (*os*))

$$\begin{aligned} & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\ \rightarrow & (\text{mapup-regs}(taskid, \text{os-running-output-handler-path3}(os)) \\ & = \text{mapup-regs}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-regs-os-running-output-handler-path3.

$$\begin{aligned} \text{THEOREM: mapup-cc-os-running-output-handler-path3} \\ & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\ \rightarrow & (\text{mapup-cc}(taskid, \text{os-running-output-handler-path3}(os)) \\ & = \text{mapup-cc}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-cc-os-running-output-handler-path3.

$$\begin{aligned} \text{THEOREM: mapup-error-os-running-output-handler-path3} \\ & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\ \rightarrow & (\text{mapup-error}(taskid, \text{os-running-output-handler-path3}(os)) \\ & = \text{mapup-error}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-error-os-running-output-handler-path3.

$$\begin{aligned} \text{THEOREM: mapup-svcflag-os-running-output-handler-path3} \\ & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\ \rightarrow & (\text{mapup-svcflag}(taskid, \text{os-running-output-handler-path3}(os)) \\ & = \text{mapup-svcflag}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-output-handler-path3.

$$\begin{aligned} \text{THEOREM: mapup-svcid-os-running-output-handler-path3} \\ & (\text{good-os}(os) \\ & \wedge ((\neg \text{tm-waiting}(os)) \\ & \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\ & \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\ \rightarrow & (\text{mapup-svcid}(taskid, \text{os-running-output-handler-path3}(os)) \\ & = \text{mapup-svcid}(taskid, os)) \end{aligned}$$

EVENT: Disable mapup-svcid-os-running-output-handler-path3.

THEOREM: mapup-task-os-running-output-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
→ (mapup-task (*taskid*, os-running-output-handler-path3 (*os*)
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-running-output-handler-path3.

THEOREM: mapup-tasks-os-running-output-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)
 \wedge (tm-some-output-interruptp (tm-oports (*os*)) \wedge (*taskid* \in \mathbf{N}))))
→ (mapup-tasks (*taskid*, os-running-output-handler-path3 (*os*)
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-running-output-handler-path3.

THEOREM: mapup-os-tasks-os-running-output-handler-path3
(good-os (*os*)
 \wedge ((\neg tm-waiting (*os*)) \wedge tm-some-output-interruptp (tm-oports (*os*))))
→ (mapup-os-tasks (os-running-output-handler-path3 (*os*)
 = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-running-output-handler-path3.

THEOREM: mapup-os-ibuffers-os-running-output-handler-path3
good-os (*os*)
→ (mapup-os-ibuffers (os-running-output-handler-path3 (*os*)
 = mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path3.

THEOREM: mapup-os-obuffers-os-running-output-handler-path3
good-os (*os*)
→ (mapup-os-obuffers (os-running-output-handler-path3 (*os*)
 = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-running-output-handler-path3.

THEOREM: mapup-os-mbuffers-os-running-output-handler-path3
good-os (*os*)
→ (mapup-os-mbuffers (os-running-output-handler-path3 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-output-handler-path3.

THEOREM: tm-rwstate-os-running-output-handler-path3
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-output-handler-path3 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-output-handler-path3.

THEOREM: tm-clock-os-running-output-handler-path3
tm-clock (os-running-output-handler-path3 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-output-handler-path3.

THEOREM: tm-iports-os-running-output-handler-path3
tm-iports (os-running-output-handler-path3 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-running-output-handler-path3.

THEOREM: tm-oports-os-running-output-handler-path3
tm-oports (os-running-output-handler-path3 (*os*))
= tm-clear-output-interrupt (tm-interrupting-output-port (tm-oports (*os*)),
tm-oports (*os*))

EVENT: Disable tm-oports-os-running-output-handler-path3.

THEOREM: correctness-of-os-running-output-handler-path3
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((¬ tm-waiting (*os*))
∧ ((getnth ('2
* tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*)
≠ '4)
∧ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))))))
→ (mapup-os (os-running-output-handler-path3 (*os*))
= ak-running-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-output-handler-path3.

THEOREM: mapup-base-os-running-output-handler-path4
good-os (*os*)
→ (mapup-base (*taskid*, os-running-output-handler-path4 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-output-handler-path4.

THEOREM: mapup-limit-os-running-output-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-output-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-output-handler-path4.

THEOREM: getseg-tm-memory-os-running-output-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-output-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-output-handler-path4.

THEOREM: tm-cpu-os-running-output-handler-path4-crock
tm-cpu (os-running-output-handler-path4 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-output-handler-path4-crock.

THEOREM: tm-cpu-os-running-output-handler-path4
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ tm-some-output-interruptp (tm-oports (*os*))))
→ (tm-cpu (os-running-output-handler-path4 (*os*))
= getnth (os-current-taskid (os-running-output-handler-path4 (*os*)),
table ('9,
os-task-table (os-running-output-handler-path4 (*os*))))))

EVENT: Disable tm-cpu-os-running-output-handler-path4.

THEOREM: mapup-cpu-os-running-output-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cpu (*taskid*, os-running-output-handler-path4 (*os*))
 = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-output-handler-path4.

THEOREM: mapup-regs-os-running-output-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-regs (*taskid*, os-running-output-handler-path4 (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-output-handler-path4.

THEOREM: mapup-cc-os-running-output-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-cc (*taskid*, os-running-output-handler-path4 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-output-handler-path4.

THEOREM: mapup-error-os-running-output-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))
 \wedge (tm-some-output-interruptp (tm-oports (*os*))
 \wedge ((*taskid* \in \mathbf{N}) \wedge (*taskid* < '16))))))
 \rightarrow (mapup-error (*taskid*, os-running-output-handler-path4 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-output-handler-path4.

THEOREM: mapup-svcflag-os-running-output-handler-path4
 (good-os (*os*)
 \wedge ((\neg tm-waiting (*os*))

$$\begin{aligned}
& \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))) \\
\rightarrow & (\text{mapup-svcflag}(taskid, \text{os-running-output-handler-path4}(os)) \\
& = \text{mapup-svcflag}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcflag-os-running-output-handler-path4.

$$\begin{aligned}
\text{THEOREM: mapup-svcid-os-running-output-handler-path4} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-svcid}(taskid, \text{os-running-output-handler-path4}(os)) \\
& = \text{mapup-svcid}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-svcid-os-running-output-handler-path4.

$$\begin{aligned}
\text{THEOREM: mapup-task-os-running-output-handler-path4} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \\
& \quad \quad \quad \wedge ((taskid \in \mathbf{N}) \wedge (taskid < '16)))))) \\
\rightarrow & (\text{mapup-task}(taskid, \text{os-running-output-handler-path4}(os)) \\
& = \text{mapup-task}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-task-os-running-output-handler-path4.

$$\begin{aligned}
\text{THEOREM: mapup-tasks-os-running-output-handler-path4} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \\
& \quad \quad \wedge (\text{tm-some-output-interruptp}(\text{tm-oports}(os)) \wedge (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}(taskid, \text{os-running-output-handler-path4}(os)) \\
& = \text{mapup-tasks}(taskid, os))
\end{aligned}$$

EVENT: Disable mapup-tasks-os-running-output-handler-path4.

$$\begin{aligned}
\text{THEOREM: mapup-os-tasks-os-running-output-handler-path4} \\
& (\text{good-os}(os) \\
& \quad \wedge ((\neg \text{tm-waiting}(os)) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))) \\
\rightarrow & (\text{mapup-os-tasks}(\text{os-running-output-handler-path4}(os)) \\
& = \text{mapup-os-tasks}(os))
\end{aligned}$$

EVENT: Disable mapup-os-tasks-os-running-output-handler-path4.

THEOREM: mapup-os-ibuffers-os-running-output-handler-path4
good-os (*os*)
→ (mapup-os-ibuffers (os-running-output-handler-path4 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path4.

THEOREM: mapup-os-obuffers-os-running-output-handler-path4
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*)
∧ (¬ array-qempty (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))
→ (mapup-os-obuffers (os-running-output-handler-path4 (*os*))
= deq-ith-buffer (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-running-output-handler-path4.

THEOREM: mapup-os-mbuffers-os-running-output-handler-path4
good-os (*os*)
→ (mapup-os-mbuffers (os-running-output-handler-path4 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-output-handler-path4.

THEOREM: tm-rwstate-os-running-output-handler-path4
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-output-handler-path4 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-output-handler-path4.

THEOREM: tm-clock-os-running-output-handler-path4
tm-clock (os-running-output-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-output-handler-path4.

THEOREM: tm-iports-os-running-output-handler-path4
tm-iports (os-running-output-handler-path4 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-running-output-handler-path4.

THEOREM: tm-oports-os-running-output-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad os)))))) \\
\rightarrow & (\text{tm-oports } (\text{os-running-output-handler-path4 } (os)) \\
& = \text{tm-start-output } (\text{qfirst } (\text{getnth } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{mapup-os-obuffers } (os))), \\
& \quad \quad \quad \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{tm-oports } (os)))
\end{aligned}$$

EVENT: Disable tm-oports-os-running-output-handler-path4.

THEOREM: correctness-of-os-running-output-handler-path4

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \\
& \quad \wedge ((\neg \text{tm-waiting } (os)) \\
& \quad \quad \wedge ((\text{getnth } ('2 \\
& \quad \quad \quad * \text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{os-status-table } (os)) \\
& \quad \quad \quad \neq '4) \\
& \quad \quad \wedge (\neg \text{array-qempty } (\text{os-current-obuffer } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad os)))))))))) \\
\rightarrow & (\text{mapup-os } (\text{os-running-output-handler-path4 } (os)) \\
& = \text{ak-running-output-interrupt-handler } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{mapup-os } (os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-running-output-handler-path4.

THEOREM: correctness-of-os-running-output-handler

$$\begin{aligned}
& (\text{good-os } (os)) \\
& \wedge (\text{tm-some-output-interruptp } (\text{tm-oports } (os)) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{mapup-os } (\text{os-running-output-handler } (os)) \\
& = \text{ak-running-output-interrupt-handler } (\text{tm-interrupting-output-port } (\text{tm-oports } (os)), \\
& \quad \quad \quad \text{mapup-os } (os)))
\end{aligned}$$

EVENT: Disable correctness-of-os-running-output-handler.

THEOREM: tm-svcid-mapup-task

$$\begin{aligned}
& (\text{good-os } (os) \wedge (\neg \text{tm-waiting } (os))) \\
\rightarrow & (\text{tm-svcid } (\text{mapup-task } (\text{qfirst } (\text{mapup-queue } (\text{os-readyq } (os))), os)) \\
& = \text{tm-svcid } (os))
\end{aligned}$$

EVENT: Disable tm-svcid-mapup-task.

THEOREM: getnth-mapup-tasks-corollary
 $((i < \text{'16}) \wedge (i \in \mathbf{N}))$
 $\rightarrow (\text{getnth}(i, \text{mapup-tasks}(\text{'0}, os)) = \text{mapup-task}(i, os))$

EVENT: Disable getnth-mapup-tasks-corollary.

THEOREM: tm-svcid-mapup-os-tasks
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-svcid}(\text{getnth}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))),$
 $\text{mapup-os-tasks}(os)))$
 $= \text{tm-svcid}(os)$

EVENT: Disable tm-svcid-mapup-os-tasks.

THEOREM: tm-svcid-getnth-ak-pstates
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-svcid}(\text{getnth}(\text{qfirst}(\text{ak-readyq}(\text{mapup-os}(os))),$
 $\text{ak-pstates}(\text{mapup-os}(os))))$
 $= \text{tm-svcid}(os)$

EVENT: Disable tm-svcid-getnth-ak-pstates.

THEOREM: tm-regs-mapup-task
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-regs}(\text{mapup-task}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), os))$
 $= \text{tm-regs}(os)$

EVENT: Disable tm-regs-mapup-task.

THEOREM: tm-regs-mapup-os-tasks
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow (\text{tm-regs}(\text{getnth}(\text{qfirst}(\text{mapup-queue}(\text{os-readyq}(os))), \text{mapup-os-tasks}(os)))$
 $= \text{tm-regs}(os)$

EVENT: Disable tm-regs-mapup-os-tasks.

THEOREM: tm-regs-getnth-ak-pstates
 $(\text{good-os}(os) \wedge (\neg \text{tm-waiting}(os)))$
 $\rightarrow ((\text{tm-r2}(\text{getnth}(\text{qfirst}(\text{ak-readyq}(\text{mapup-os}(os))),$
 $\text{ak-pstates}(\text{mapup-os}(os))))$
 $= \text{tm-r2}(os)$
 $\wedge (\text{tm-r3}(\text{getnth}(\text{qfirst}(\text{ak-readyq}(\text{mapup-os}(os))),$
 $\text{ak-pstates}(\text{mapup-os}(os))))$
 $= \text{tm-r3}(os))$

EVENT: Disable tm-regs-getnth-ak-pstates.

THEOREM: ak-readyq-mapup-os
 $\text{good-os}(os) \rightarrow (\text{mapup-queue}(\text{os-readyq}(os)) = \text{ak-readyq}(\text{mapup-os}(os)))$

EVENT: Disable ak-readyq-mapup-os.

THEOREM: correctness-of-svc-handler
 $(\text{good-os}(os) \wedge ((\neg \text{tm-waiting}(os)) \wedge (\neg \text{tm-errorp}(os))))$
 $\rightarrow (\text{mapup-os}(\text{os-svc-handler}(os)) = \text{ak-svc-handler}(\text{mapup-os}(os)))$

EVENT: Disable correctness-of-svc-handler.

THEOREM: ak-waiting-mapup-os
 $\text{good-os}(os) \rightarrow (\text{ak-waiting}(\text{mapup-os}(os)) = \text{tm-waiting}(os))$

EVENT: Disable ak-waiting-mapup-os.

THEOREM: ak-iports-mapup-os
 $\text{good-os}(os) \rightarrow (\text{ak-iports}(\text{mapup-os}(os)) = \text{tm-iports}(os))$

EVENT: Disable ak-iports-mapup-os.

THEOREM: correctness-of-input-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-input-interruptp}(\text{tm-iports}(os)))$
 $\rightarrow (\text{mapup-os}(\text{os-input-interrupt-handler}(os))$
 $\quad = \text{ak-input-interrupt-handler}(\text{tm-interrupting-input-port}(\text{ak-iports}(\text{mapup-os}(os))),$
 $\quad \quad \quad \text{mapup-os}(os))$

EVENT: Disable correctness-of-input-interrupt-handler.

THEOREM: ak-oports-mapup-os
 $\text{good-os}(os) \rightarrow (\text{ak-oports}(\text{mapup-os}(os)) = \text{tm-oports}(os))$

EVENT: Disable ak-oports-mapup-os.

THEOREM: correctness-of-output-interrupt-handler
 $(\text{good-os}(os) \wedge \text{tm-some-output-interruptp}(\text{tm-oports}(os)))$
 $\rightarrow (\text{mapup-os}(\text{os-output-interrupt-handler}(os))$
 $\quad = \text{ak-output-interrupt-handler}(\text{tm-interrupting-output-port}(\text{ak-oports}(\text{mapup-os}(os))),$
 $\quad \quad \quad \text{mapup-os}(os))$

EVENT: Disable correctness-of-output-interrupt-handler.

THEOREM: tm-error-mapup-task
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (tm-error (mapup-task (qfirst (mapup-queue (os-readyq(*os*))), *os*))
= tm-error(*os*))

EVENT: Disable tm-error-mapup-task.

THEOREM: tm-error-mapup-os-tasks
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (tm-error (getnth (qfirst (mapup-queue (os-readyq(*os*))),
mapup-os-tasks(*os*)))
= tm-error(*os*))

EVENT: Disable tm-error-mapup-os-tasks.

THEOREM: tm-svcflag-mapup-task
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (tm-svcflag (mapup-task (qfirst (mapup-queue (os-readyq(*os*))), *os*))
= tm-svcflag(*os*))

EVENT: Disable tm-svcflag-mapup-task.

THEOREM: tm-svcflag-mapup-os-tasks
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (tm-svcflag (getnth (qfirst (mapup-queue (os-readyq(*os*))),
mapup-os-tasks(*os*)))
= tm-svcflag(*os*))

EVENT: Disable tm-svcflag-mapup-os-tasks.

THEOREM: not-tm-in-supervisor-mode-mapup-task
 \neg tm-in-supervisor-mode (mapup-task (*taskid*, *os*))

EVENT: Disable not-tm-in-supervisor-mode-mapup-task.

THEOREM: not-tm-in-supervisor-mode-getnth-mapup-os-tasks
(good-os(*os*) \wedge (\neg tm-waiting(*os*)))
 \rightarrow (\neg tm-in-supervisor-mode (getnth (qfirst (mapup-queue (os-readyq(*os*))),
mapup-os-tasks(*os*))))

EVENT: Disable not-tm-in-supervisor-mode-getnth-mapup-os-tasks.

THEOREM: ak-input-interruptp-mapup-os
good-os (*os*)
→ (ak-input-interruptp (mapup-os (*os*)) = tm-input-interruptp (*os*))

EVENT: Disable ak-input-interruptp-mapup-os.

THEOREM: ak-output-interruptp-mapup-os
good-os (*os*)
→ (ak-output-interruptp (mapup-os (*os*)) = tm-output-interruptp (*os*))

EVENT: Disable ak-output-interruptp-mapup-os.

THEOREM: ak-errorp-mapup-os
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (ak-errorp (mapup-os (*os*)) = tm-errorp (*os*))

EVENT: Disable ak-errorp-mapup-os.

THEOREM: ak-svc-interruptp-mapup-os
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (ak-svc-interruptp (mapup-os (*os*)) = tm-svc-interruptp (*os*))

EVENT: Disable ak-svc-interruptp-mapup-os.

THEOREM: ak-clock-interruptp-mapup-os
good-os (*os*)
→ (ak-clock-interruptp (mapup-os (*os*)) = tm-clock-interruptp (*os*))

EVENT: Disable ak-clock-interruptp-mapup-os.

THEOREM: os-step-implements-ak-step
good-os (*os*) → (mapup-os (os-step (*os*)) = ak-step (mapup-os (*os*)))

EVENT: Disable os-step-implements-ak-step.

THEOREM: mapup-cpu-tm-post-interrupt
mapup-cpu (*taskid*, tm-post-interrupt (*interrupt*, *os*)) = mapup-cpu (*taskid*, *os*)

EVENT: Disable mapup-cpu-tm-post-interrupt.

THEOREM: mapup-task-tm-post-interrupt
mapup-task (*taskid*, tm-post-interrupt (*interrupt*, *os*))
= mapup-task (*taskid*, *os*)

EVENT: Disable mapup-task-tm-post-interrupt.

THEOREM: mapup-tasks-tm-post-interrupt
mapup-tasks (*taskid*, tm-post-interrupt (*interrupt*, *os*))
= mapup-tasks (*taskid*, *os*)

EVENT: Disable mapup-tasks-tm-post-interrupt.

THEOREM: mapup-os-tm-post-interrupt
mapup-os (tm-post-interrupt (*interrupt*, *os*))
= ak-post-interrupt (*interrupt*, mapup-os (*os*))

EVENT: Disable mapup-os-tm-post-interrupt.

DEFINITION:
bak-processor (*ak*, *oracle*)
= **if** listp (*oracle*)
 then ak-step (ak-post-interrupt (car (*oracle*),
 bak-processor (*ak*, cdr (*oracle*))))
 else *ak* **endif**

THEOREM: bak-implements-ak
bak-processor (*ak*, *oracle*) = ak-processor (*ak*, reverse (*oracle*))

EVENT: Disable bak-implements-ak.

THEOREM: bos-implements-bak
good-os (*os*)
→ (mapup-os (bos-processor (*os*, *oracle*))
 = bak-processor (mapup-os (*os*), *oracle*))

EVENT: Disable bos-implements-bak.

THEOREM: os-implements-ak
(good-os (*os*) ∧ plistp (*oracle*))
→ (mapup-os (os-processor (*os*, *oracle*))
 = ak-processor (mapup-os (*os*), *oracle*))

EVENT: Disable os-implements-ak.

THEOREM: correctness-of-operating-system
 (good-os (*os*) ∧ plistp (*oracle*))
 → (mapup-os (tm-processor (*os*, os-oracle (*os*, *oracle*)))
 = ak-processor (mapup-os (*os*), *oracle*))

EVENT: Disable correctness-of-operating-system.

DEFINITION: project-ith-task (*i*, *os*) = project (*i*, mapup-os (*os*))

THEOREM: os-implements-parallel-tasks
 (good-os (*os*)
 ∧ (plistp (*oracle*)
 ∧ finite-numberp (*i*, length (ak-pstates (mapup-os (*os*))))))
 → (project-ith-task (*i*, tm-processor (*os*, os-oracle (*os*, *oracle*)))
 = task-processor (project-ith-task (*i*, *os*),
 i,
 control-oracle (*i*, mapup-os (*os*), *oracle*)))

Index

- a-finite-number-is-a-number, 49
- a-finite-number-is-lessp-than-it
s-lub, 49
- a-finite-number-list-is-a-numbe
r-list, 50
- a-finite-number-table-is-a-table, 61
- access-ak-error-handler, 314
- access-ak-input-interrupt-handle
r, 313
- access-ak-output-interrupt-hand
ler, 313
- access-ak-post-interrupt, 313
- access-current-ibuffer, 422
- access-current-ibuffer-qhead, 422
- access-current-mbuffer, 413
- access-current-mbuffer-qhead, 414
- access-current-obuffer, 429
- access-current-obuffer-qhead, 431
- access-current-os-mbuffer-field
s, 410
- s-for-receive, 412
- access-os-memory, 736
- access-os-segment-table, 404
- access-os-status-table, 406
- access-os-status-table-crock, 405
- address-space-good-memory-tm-st
ore, 209
- address-space-good-regs-tm-store, 209
- address-space-memlength, 205
- ak, 279, 282–290, 372
- ak-block-input, 285, 286, 305, 311
- ak-block-output, 285, 304, 311
- ak-block-receive, 284, 304, 310
- ak-block-send, 283, 284, 303, 310
- ak-channels, 312, 313, 315, 316
- ak-clear-input-interrupt, 280, 287, 288
- ak-clear-output-interrupt, 280, 288,
289
- ak-clock, 282–289, 297
- ak-clock-interrupt-handler, 282, 289,
303, 310, 313, 892
- ak-clock-interruptp, 289, 313, 1103
- ak-clock-interruptp-mapup-os, 1103
- ak-communicationp, 280
- ak-destid, 280, 286, 306, 315
- ak-dispatcher, 282–285, 287, 288
- ak-error-handler, 282, 289, 302, 310,
313, 314, 896, 902
- ak-error-status, 281
- ak-errorp, 280, 289, 313, 1103
- ak-errorp-mapup-os, 1103
- ak-execute-input, 286, 305, 311, 315,
316, 958, 964, 971
- ak-execute-input-from-buffer, 286, 305,
311
- ak-execute-input-implements-tas
k-execute-input, 315
- ak-execute-output, 285, 286, 305, 311,
316, 975, 981, 987, 993
- ak-execute-output-implements-ta
sk-execute-output, 316
- ak-execute-output-to-buffer, 285, 305,
311
- ak-execute-receive, 284, 286, 304, 311,
315, 931, 938, 946, 954
- ak-execute-receive-from-buffer, 284,
304, 311
- ak-execute-receive-implements-t
ask-execute-receive, 315
- ak-execute-send, 283, 286, 303, 310,
315, 906, 913, 920, 927
- ak-execute-send-implements-task
-execute-send, 315
- ak-execute-send-to-buffer, 283, 284,
303, 310
- ak-fetch-execute, 280, 286
- ak-ibuffers, 282–289, 294, 305, 312–
314
- ak-idevid, 280
- ak-implements-active-task-step, 317
- ak-implements-non-active-task-step,
314

- ak-implements-parallel-tasks, 317
- ak-input-interrupt-handler, 288, 289, 308, 312, 313, 1101
- ak-input-interruptp, 281, 289, 308, 312, 1103
- ak-input-interruptp-mapup-os, 1103
- ak-input-status, 281
- ak-interrupting-input-port, 281, 289, 308
- ak-interrupting-output-port, 281, 289, 309
- ak-iptos, 281–289, 297, 307, 308, 312, 1101
- ak-iptos-mapup-os, 1101
- ak-mbuffers, 282–289, 295, 303, 304, 312–314
- ak-message, 280, 286, 306, 315, 316
- ak-obuffers, 282–289, 294, 295, 304, 305, 312–314
- ak-odevid, 280
- ak-oport-idlep, 280, 285
- ak-oports, 281–290, 297, 309, 312, 1101
- ak-oports-mapup-os, 1101
- ak-output-interrupt-handler, 289, 309, 312, 313, 1101
- ak-output-interruptp, 281, 289, 309, 312, 1103
- ak-output-interruptp-mapup-os, 1103
- ak-output-status, 281
- ak-post-interrupt, 289, 290, 309, 310, 313, 314, 317, 1104
- ak-post-output-interrupt, 280, 285
- ak-private-step, 286, 289, 306, 312–314, 316, 392
- ak-private-step-implements-task-fetch-execute, 316
- ak-private-step-non-interference, 314
- ak-processor, 290, 310, 317, 1104, 1105
- ak-pstates, 279, 280, 282–289, 294, 310–317, 1100, 1105
- ak-pstates-ak-block-input, 311
- ak-pstates-ak-block-output, 311
- ak-pstates-ak-block-receive, 310
- ak-pstates-ak-block-send, 310
- ak-pstates-ak-clock-interrupt-handler, 310
- ak-pstates-ak-error-handler, 310
- ak-pstates-ak-input-interrupt-handler, 312
- ak-pstates-ak-output-interrupt-handler, 312
- ak-ready-set, 292, 298, 302, 377
- ak-ready-set-mapup-os, 377
- ak-ready-status, 281
- ak-ready-statusp, 281
- ak-readyq, 279, 282–289, 296, 298, 302, 1100, 1101
- ak-readyq-mapup-os, 1101
- ak-receive-instructionp, 279, 286
- ak-receive-status, 281
- ak-run-state, 282
- ak-running-input-interrupt-handler, 287, 288, 307, 1031, 1037, 1043, 1047, 1052, 1057
- ak-running-output-interrupt-handler, 288, 289, 308, 309, 1084, 1090, 1094, 1099
- ak-rwstate, 282–289, 296
- ak-send-instructionp, 279, 286
- ak-send-status, 281
- ak-shell, 279
- ak-shellp, 293
- ak-shellp-ak, 293
- ak-srcid, 280, 286, 306, 315
- ak-start-output, 281, 288, 289
- ak-status, 281–289, 292, 296, 302–304, 306
- ak-status-flag-field, 281
- ak-status-length, 281
- ak-status-lub, 281
- ak-status-taskid-field, 281
- ak-step, 289, 290, 309, 312–314, 317, 1103, 1104
- ak-store-message, 280, 284, 286
- ak-svc-handler, 286, 289, 306, 312–314, 316, 1101

- ak-svc-handler-implements-task-communication-step, 316
- ak-svc-handler-non-interference, 314
- ak-svc-interruptp, 280, 289, 313, 1103
- ak-svc-interruptp-mapup-os, 1103
- ak-taskid, 279, 280, 282, 283, 286, 295, 303–305, 313–316
- ak-taskidlub, 279
- ak-time-slice, 279
- ak-tyi-instructionp, 279
- ak-tyo-instructionp, 279, 286
- ak-update-control, 279, 283–286
- ak-update-ibuffer, 287
- ak-wait-state, 282
- ak-waiting, 282, 288, 289, 295, 298, 302–309, 313–316, 1101
- ak-waiting-iff-qemptyp-ak-readyq, 298
- ak-waiting-input-interrupt-handler, 287, 288, 307, 1000, 1007, 1014, 1017, 1021, 1025
- ak-waiting-mapup-os, 1101
- ak-waiting-output-interrupt-handler, 288, 289, 308, 1064, 1071, 1074, 1078
- ak-waiting-to-inputp, 282, 287, 302, 306, 307, 1000, 1017
- ak-waiting-to-inputp-implies-not-status-flag-readyp, 306
- ak-waiting-to-inputp-implies-readyq-not-full, 302
- ak-waiting-to-inputp-mapup-os, 999
- ak-waiting-to-outputp, 282, 288, 302, 306, 308, 1063, 1074
- ak-waiting-to-outputp-implies-not-status-flag-readyp, 306
- ak-waiting-to-outputp-implies-readyq-not-full, 302
- ak-waiting-to-outputp-mapup-os, 1063
- ak-waiting-to-receivep, 281, 283, 302, 303, 920, 926
- ak-waiting-to-receivep-implies-not-status-flag-readyp, 303
- ak-waiting-to-receivep-implies-readyq-not-full, 302
- ak-waiting-to-receivep-mapup-os, 919
- ak-waiting-to-sendp, 281, 284, 302, 304, 946, 953
- ak-waiting-to-sendp-implies-not-status-flag-readyp, 304
- ak-waiting-to-sendp-implies-readyq-not-full, 302
- ak-waiting-to-sendp-mapup-os, 945
- alu-carry, 85, 98
- alu-result, 85, 86
- alu-value, 85–90, 98, 103–106
- append-cdr, 33
- append-crock, 33
- append-firstn-nthcdr, 54
- append-fixlength, 33
- append-n-times, 318, 319
- append-nil-on-right, 33
- append-non-list-left, 33
- append-non-list-left-corollary, 33
- array-deq, 70–72, 81, 82, 354, 366, 368, 392, 415, 417, 423, 426, 431, 444, 454, 455, 457, 458, 460, 470, 472, 488, 490, 497, 505, 507, 523, 524, 532, 541, 543, 549, 550, 552, 562, 564, 572, 573, 577, 740, 742–747, 749–754, 763–766, 768–770, 779–785, 789, 791, 792, 794, 795, 896, 898–902, 906, 908–911, 913, 931, 933–936, 938, 958, 960–964, 975, 977–981
- array-enq, 70–72, 76, 80, 84, 354, 366, 368, 392, 415, 417, 424–426, 431, 444, 756, 757, 772, 773, 802, 804, 806, 809, 810, 813, 815, 826, 827, 829, 832, 834, 837, 838, 851, 852, 854, 856, 858, 865, 867, 868, 871, 873
- array-nonlast, 73, 74, 76, 77, 83
- array-qarray-array-deq, 72
- array-qarray-array-enq, 71
- array-qarray-array-nonlast, 74
- array-qarray-array-qreplace, 77
- array-qcurrenlength-array-deq, 72

array-qcurrenlength-array-enq, 71
 array-qcurrenlength-array-nonlast, 74
 array-qcurrenlength-array-qreplace, 77
 array-qcurrenlength-in-a-full-queue, 72
 array-qcurrenlength-in-a-non-empty-queue, 73
 array-qcurrenlength-in-a-non-full-queue, 73
 array-qcurrenlength-in-an-empty-queue, 73
 array-qempty, 70, 72, 73, 76–79, 81–84, 354, 364, 366, 368, 370, 392, 393, 417, 418, 426, 427, 454, 455, 457, 458, 460, 470, 472, 488, 490, 497, 503, 505, 507, 510, 523, 524, 526, 529, 531, 532, 539, 541, 543, 546, 549, 550, 552, 562, 564, 572, 573, 577, 680, 682, 691, 694, 699, 701–704, 711, 714, 723, 725, 730, 732, 734–736, 738, 740, 743, 744, 747, 750, 751, 754, 763, 766, 770, 772–775, 778, 779, 782, 785, 787, 789, 792, 795, 851, 855, 857, 859, 862, 864–866, 869, 870, 872, 874, 877, 879, 881, 896, 898–902, 906, 909–911, 913, 931, 934–936, 938, 944, 946, 952, 954, 958, 960–964, 969, 971, 975, 977–981, 1064, 1069–1071, 1074, 1077, 1078, 1084, 1088, 1090, 1094, 1098, 1099
 array-qempty-implies-not-array-qfull, 84
 array-qempty-array-deq-when-de-lta-is-1, 81
 lta-is-1-alternative, 81
 array-qfirst, 70, 82, 84, 354, 357, 365, 366, 368, 379, 380, 392, 403, 407, 408, 411, 413, 415, 417, 426, 427, 432–436, 444, 568, 570, 575–577, 739, 742, 743, 745, 746, 749, 750, 753, 765, 768, 769, 781, 784, 785, 791, 794–796, 799, 802, 889, 895, 898, 899, 902, 905, 908, 909, 912, 914–918, 921–925, 930, 933, 934, 937, 939–944, 947–952, 957, 960, 961, 964–969, 974, 977, 981–985, 987–991, 993
 array-qfirst-array-enq, 84
 array-qfirst-array-enq1, 84
 array-qfirst-finite-number-queue, 354
 -corollary, 366
 -corollary-for-ibuffers, 417
 -corollary-for-mbuffers, 368
 -corollary-for-obuffers, 426
 array-qfirst-os-readyq-os-svc-recv-handler-path3, 939
 array-qfirst-os-readyq-os-svc-snd-handler-path3, 914
 array-qfull, 70–73, 77, 78, 80–82, 84, 354, 366, 368–370, 415, 417, 426, 468, 475, 488, 490, 492, 494, 496, 497, 560, 566, 572, 573, 575–577, 585, 588, 591, 600, 602, 603, 608, 611, 613, 621, 623–625, 633, 636, 639, 647, 648, 650, 655, 658, 661, 668, 670–672, 747, 750, 751, 754, 757–759, 762, 789, 792, 795–797, 799, 802–804, 807, 809, 811, 812, 814, 816, 818, 819, 821, 822, 824–826, 828, 830, 832, 835, 837, 839–841, 843, 845–847, 849, 850, 906, 913–918, 920, 925, 927, 939–943, 975, 981, 986, 987, 991, 993, 998, 1000, 1005, 1007, 1012, 1013, 1016, 1017, 1020, 1021, 1023, 1025, 1029, 1031, 1035, 1037, 1041, 1043, 1046, 1047, 1051, 1052, 1055, 1056
 array-qfull-implies-not-array-qe

- mptyp, 78
- array-qhead-array-deq, 71
- array-qhead-array-enq, 70
- array-qhead-array-nonlast, 73
- array-qhead-array-qreplace, 77
- array-qhead-equals-array-qtail-i
 - ff-qfull-or-qempty, 78
 - n-an-empty-queue, 78
- array-qindex-relation, 67, 68
- array-qmaxlength-array-deq, 72
- array-qmaxlength-array-enq, 71
- array-qmaxlength-array-nonlast, 74
- array-qmaxlength-array-qreplace, 77
- array-qmaxlength-must-be-1, 81
- array-qreplace, 76–78, 84, 370, 393, 418, 424
- array-qtail-array-deq, 72
- array-qtail-array-enq, 70
- array-qtail-array-nonlast, 74
- array-qtail-array-qreplace, 77
- array-queue-facts, 78
- array-queuep, 68–74, 76–84, 352, 353, 365, 367, 392, 393, 415, 416, 425, 975, 1000
- array-queuep-array-deq, 72
- array-queuep-array-enq, 71
- array-queuep-array-nonlast, 76
- array-queuep-array-qreplace, 78
- array-queuep-finite-number-queue, 353
 - corollary, 365
 - corollary-for-ibuffers, 416
 - corollary-for-mbuffers, 367
 - corollary-for-obuffers, 425
- asm-good-arg, 321, 324
- asm-good-instruction, 321, 325
- asm-instruction-arg1, 320, 324, 325
- asm-instruction-arg2, 321, 324, 325
- asm-instruction-index1, 320, 324
- asm-instruction-index2, 321, 324
- asm-instruction-mode1, 320, 324
- asm-instruction-mode2, 320, 324
- asm-instruction-opcode, 319, 324
- asm-opcode-alist, 317
- asm-pack-instruction, 324, 325
- asm-pack-word1, 324, 325
- asm-reverse-opcode-alist, 329
- asm-symbolic-opcode, 330
- assemble, 329
- assemble1, 329
- assemble2, 328, 329
- assembler, 329
- assembly-errorp, 328
- associativity-of-append, 33
- associativity-of-plus, 2
- associativity-of-plus-crock, 414
- associativity-of-plus-inverse, 23
- associativity-of-times, 4
- bak-implements-ak, 1104
- bak-processor, 1104
- base, 194, 195, 363, 371, 374, 379, 380, 386, 387, 403, 404, 740, 747, 754, 758, 762, 769, 774, 778, 785, 788, 795, 798, 801, 806, 811, 815, 816, 830, 834, 839, 842, 846, 850, 854, 858, 859, 869, 874, 877, 880
- base-limit, 194
- bos-implements-bak, 1104
- bos-implements-os, 886
- bos-processor, 886, 1104
- boundp, 318–322, 326
- btimed-tm-implements-bos, 886
- btimed-tm-processor, 885, 886
- build-symtab, 327–329
- cadr-firstn, 61
- canonicalize-plus-terms1, 13
- canonicalize-plus-terms2, 13
- canonicalize-plus-terms3, 13
- car-append, 30
- car-append-casesplit, 30
- car-getseg, 30
- car-nthcdr, 53
- car-putnth-zero, 30
- car-putseg-zero, 31
- cdr-append, 31
- cdr-getseg, 31

- cdr-is-nil-when-length-is-1, 31
- cdr-nthcdr, 53
- cl-arithmetic, 12
- clean-assembly, 342
- commutativity-of-permutation, 26
- commutativity-of-plus, 2
- commutativity-of-times, 4
- commutativity2-of-plus, 2
- commutativity2-of-times, 4
- contract-array-deq, 392
- contract-array-enq, 392
- contract-array-enq-instance, 392
- contract-array-qfirst, 392
- contract-array-qreplace, 393
- contract-getnth-from-current-ib
offer, 422
- contract-getnth-from-current-mb
offer-for-receive, 414
- contract-getnth-from-current-ob
offer, 429
offer1, 430
- contract-list-of-getnths8, 398
- contract-os-new-base-expression, 404
- contract-os-new-limit-expression,
404
- contract-putnth-into-current-ib
offer, 422
offer1, 423
offer2, 423
- contract-putnth-into-current-mb
offer, 413
offer-for-receive1, 414
- contract-putnth-into-current-ob
offer, 430
offer1, 430
- contract-putnths-into-status-table,
405
- contract-queues, 393
- contract-sequence-of-putnths2, 404
- contract-sequence-of-putnths8, 399
- control-oracle, 313, 317, 1105
- control-oracle-step, 312–314, 317
- correctness-of-array-deq, 82
- correctness-of-array-deq-case-1, 82
- correctness-of-array-deq-case-3, 81
- correctness-of-array-enq, 80
- correctness-of-array-nonlast, 83
- correctness-of-array-nonlast-ca
se1, 83
se2, 83
- correctness-of-array-qempty, 82
- correctness-of-array-qfirst, 82
- correctness-of-array-qfullp, 82
- correctness-of-array-qfullp-ibuffers-
instance, 1000
- correctness-of-array-qfullp-mbuffers-
instance, 415
- correctness-of-array-qfullp-obuffers-
instance, 975
- correctness-of-array-qreplace, 84
- correctness-of-clock-interrupt-
handler, 892
- correctness-of-input-interrupt-
handler, 1101
- correctness-of-operating-system, 1105
- correctness-of-os-error-handler, 902
-path1, 896
-path2, 902
- correctness-of-os-running-input-
handler, 1057
-handler-path1, 1031
-handler-path2, 1037
-handler-path3, 1043
-handler-path4, 1047
-handler-path5, 1052
-handler-path6, 1056
- correctness-of-os-running-output-
handler, 1099
-handler-path1, 1084
-handler-path2, 1090
-handler-path3, 1094
-handler-path4, 1099
- correctness-of-os-svc-receive-
handler, 954
handler-path1, 931
handler-path2, 937
handler-path3, 946
handler-path4, 954

- correctness-of-os-svc-send-hand
 - ler, 927
 - ler-path1, 906
 - ler-path2, 912
 - ler-path3, 920
 - ler-path4, 927
- correctness-of-os-svc-tyi-handle
 - r, 971
 - r-path1, 958
 - r-path2, 964
 - r-path3, 971
- correctness-of-os-svc-tyo-handle
 - r, 993
 - r-path1, 975
 - r-path2, 981
 - r-path3, 987
 - r-path4, 993
- correctness-of-os-waiting-input
 - handler, 1025
 - handler-path1, 1000
 - handler-path2, 1007
 - handler-path3, 1013
 - handler-path4, 1017
 - handler-path5, 1021
 - handler-path6, 1024
- correctness-of-os-waiting-output
 - handler, 1078
 - handler-path1, 1063
 - handler-path2, 1070
 - handler-path3, 1074
 - handler-path4, 1078
- correctness-of-output-interrupt
 - handler, 1101
- correctness-of-svc-handler, 1101
- correctness-of-tm-fetch-execute, 392
- cpu-regs-segment, 130
- dc-formp, 318, 319
- decr-mod, 63, 64, 73, 74, 77, 82, 83,
 - 85, 87, 370, 393, 421, 423
- decr-mod-size, 63
- delta, 67, 74, 79, 81
- delta-a-a, 67
- delta-equals-length-delta-segme
 - nt, 79
- delta-equals-max-iff-a-equals-b, 67
- delta-incr-mod-a, 67
- delta-incr-mod-b, 67
- delta-segment, 78–83
- delta-segment-decr-mod-arg2, 83
- delta-segment-incr-mod-a, 80
- delta-segment-incr-mod-b, 80
- delta-segment-putnth-non-interfe
 - rence, 80
- delta-sub1-b-case1, 74
- delta-sub1-b-case2, 74
- deq, 64–66, 82, 84, 272, 282–286,
 - 300, 892, 895, 901, 905, 912,
 - 930, 937, 957, 963, 974, 980
- deq-is-remove, 66
- deq-ith-buffer, 272, 275, 288, 969,
 - 1069, 1077, 1088, 1098
- deq2, 84, 274, 284, 944, 952
- difference-0, 3
- difference-1, 10
- difference-2, 10
- difference-add1-arg1, 14
- difference-add1-arg2, 15
- difference-add1-cancellation, 7
- difference-by-larger-number, 14
- difference-cancellation-0, 3
- difference-cancellation-1, 3
- difference-difference, 15
- difference-elim, 5
- difference-equals-0, 16
- difference-is-zero, 14
- difference-plus, 3
- difference-plus-cancelation, 6
- difference-plus-cancellation2-i
 - nstance, 15
- difference-plus1, 6
- difference-plus2, 6
- difference-plus3, 7
- difference-plus4, 22
- difference-sub1-arg1, 15
- difference-sub1-arg2, 15
- difference-sub1-plus-crock, 63
- difference-x-x, 3

- difference-zero-arg2, 391
- disjoint, 194–200, 386
- disjoint-everywhere, 194, 195, 363, 379, 380
- disjoint-everywhere-os-segment-table, 363
- disjoint-getnth, 195
- disjoint-segments-induction, 385, 386
- distributivity-of-divides, 9
- distributivity-of-times-over-difference, 16
- distributivity-of-times-over-plus, 4
 - us-backwards, 14
- divides, 6, 8, 9, 21, 23, 357
- divides-implies-times, 10
- divides-not-lessp-instance, 357
- divides-plus-rewrite, 7
- divides-plus-rewrite-commuted, 7
- divides-plus-rewrite1, 7
- divides-plus-rewrite2, 7
- divides-times, 6
- divides-times1, 8
- divisor-not-lessp, 357
- double-number-double-list-induction, 12, 13
- double-number-induction, 12
- element-of-non-zero-list-is-a-number, 50
- element-of-non-zero-list-is-non-zero, 50
- enq, 64–66, 80, 84, 272, 282–284, 287, 288, 302, 892, 919, 945, 999, 1006, 1012, 1030, 1036, 1042, 1062, 1069, 1083, 1089
- enq-ith-buffer, 272, 275, 285, 287, 986, 991, 1005, 1012, 1020, 1023, 1035, 1041, 1051, 1055
- enq2, 84, 274, 283, 918, 925
- equal-difference-0, 3
- equal-length-status-sum-of-read-y-and-unready-tasks, 300
- equal-lessp, 5
- equal-times-0, 4
- equality-of-getnth-backchain-collary, 379
- equality-of-getnth-backchain-proof, 44
- equality-of-getseg-backchain, 47
- equality-of-getseg-backchain-base-case, 45
- equality-of-getseg-backchain-collary, 379
- equality-of-getseg-backchain-induction-step, 46
 - duction-step-case1, 45
 - duction-step-case2, 46
- equality-of-getseg-backchain-proof, 47
- equivalence-of-clock-interrupt-expressions, 439
- equivalence-of-error-interrupt-expressions, 448
- equivalence-of-input-interrupt-expressions, 578
- equivalence-of-output-interrupt-expressions, 674
- equivalence-of-svc-interrupt-expressions, 882
- equivalence-of-svc-receive-interrupt-expressions, 498
- equivalence-of-svc-send-interrupt-expressions, 461
- equivalence-of-svc-tyi-interrupt-expressions, 533
- equivalence-of-svc-tyo-interrupt-expressions, 553
- euclid, 8
- executable-fns, 438
- executable-theory, 743
- exp, 4, 10, 11, 17
- exp-2-never-0, 17
- exp-2-never-0-linear, 17
- exp-by-0, 11
- exp-exp, 11
- exp-of-0, 10
- exp-of-1, 10

- exp-plus, 4
- exp-times, 11
- find-duplicate-symbol, 328, 329
- find-non-finite-number, 328, 329
- finite-number-listp, 49–52, 61–63, 65, 66, 79, 120, 121, 128, 129, 133, 135, 206, 209, 276–278, 296, 352, 353, 373, 374, 377
- finite-number-listp-address-space-memory, 206
- finite-number-listp-address-space-regs, 206
- finite-number-listp-ak-readyq, 296
- finite-number-listp-append, 51
- finite-number-listp-delta-segment, 79
- finite-number-listp-deq, 65
- finite-number-listp-enq, 65
- finite-number-listp-fixlength, 52
- finite-number-listp-flatten, 61
- finite-number-listp-for-zero-list, 52
- finite-number-listp-from-smaller-lub, 52
- finite-number-listp-getseg, 51
- finite-number-listp-list, 51
- finite-number-listp-mapup-cpu, 373
- finite-number-listp-mapup-finite-number-queue, 377
- finite-number-listp-mapup-finite-number-queue-corollary, 377
- finite-number-listp-mapup-queue, 79
- finite-number-listp-mapup-regs, 374
- finite-number-listp-nil, 61
- finite-number-listp-nonlast, 63
- finite-number-listp-on-non-list, 129
- finite-number-listp-permutation, 51
- finite-number-listp-putnth, 51
- finite-number-listp-putseg, 51
- finite-number-listp-qarray, 353
- finite-number-listp-qreplace, 66
- finite-number-listp-remove, 51
- finite-number-listp-tm-cpu, 129
- finite-number-listp-tm-memory, 120
- finite-number-listp-tm-regs, 121
- finite-number-queue-corollaries, 367
 - for-ibuffers, 418
 - for-ibuffers1, 418
 - for-mbuffers, 406
 - for-mbuffers1, 406
 - for-obuffers, 427
 - for-obuffers1, 427
 - 1, 395
- finite-number-queue-listp, 355–357, 364, 368, 369, 372, 375, 757, 773, 787, 798, 805, 809, 814, 818, 821, 824, 828, 832, 837, 841, 845, 849, 857, 864, 872, 879
- finite-number-queue-listp-first-n, 356
- finite-number-queue-listp-getnth, 356
- finite-number-queue-listp-getnth-corollary, 368
- finite-number-queue-listp-getseg, 357
- finite-number-queue-listp-nthcdr, 356
- finite-number-queue-listp-os-ibuffers, 364
 - offers-os-running-input-handler-path1, 828
 - offers-os-running-input-handler-path2, 832
 - offers-os-running-input-handler-path3, 837
 - offers-os-running-input-handler-path4, 841
 - offers-os-running-input-handler-path5, 845
 - offers-os-running-input-handler-path6, 848
 - offers-os-waiting-input-handler-path1, 804
 - offers-os-waiting-input-handler-path2, 809
 - offers-os-waiting-input-handler-path3, 813

- uffers-os-waiting-input-handler-path4, 818
 - uffers-os-waiting-input-handler-path5, 821
 - uffers-os-waiting-input-handler-path6, 824
 - uffers-with-dequeued-character, 787
- finite-number-queue-listp-os-mb
 - uffers, 364
 - uffers-with-dequeued-message, 772
 - uffers-with-enqueued-message, 756
- finite-number-queue-listp-os-ob
 - uffers, 364
 - uffers-os-running-output-handler-path2, 872
 - uffers-os-running-output-handler-path4, 879
 - uffers-os-waiting-output-handler-path2, 857
 - uffers-os-waiting-output-handler-path4, 864
 - uffers-with-enqueued-character, 797
- finite-number-queue-listp-putnt
 - h, 355
- finite-number-queue-qarray-after-deq, 354
- finite-number-queue-qarray-after-enq, 354
- finite-number-queue-qarray-after-replace, 370
- finite-number-queue-qcurrentlength
 - size, 353
 - size-corollary, 366
 - size-corollary-for-ibuffers, 417
 - size-corollary-for-mbuffers, 367
 - size-corollary-for-obuffers, 426
- finite-number-queue-qhead-size, 353
- finite-number-queue-qhead-size-corollary, 365
- finite-number-queue-qhead-size-corollary-for-ibuffers, 416
- finite-number-queue-qhead-size-corollary-for-mbuffers, 367
- finite-number-queue-qhead-size-corollary-for-obuffers, 426
- finite-number-queue-qmaxlength
 - field, 353
- finite-number-queue-qmaxlength-value, 354
- finite-number-queue-qmaxlength-value-corollary, 366
- finite-number-queue-qmaxlength-value-corollary-for-ibuffers, 417
- finite-number-queue-qmaxlength-value-corollary-for-mbuffers, 367
- finite-number-queue-qmaxlength-value-corollary-for-obuffers, 426
- finite-number-queue-qtail-size, 353
- finite-number-queue-qtail-size-corollary, 365
- finite-number-queue-qtail-size-corollary-for-ibuffers, 416
- finite-number-queue-qtail-size-corollary-for-mbuffers, 367
- finite-number-queue-qtail-size-corollary-for-obuffers, 426
- finite-number-queue-table, 355–357, 364, 368, 369
- finite-number-queue-table-corollaries, 369
- finite-number-queue-table-os-mbuffers, 364
- finite-number-queue-table-putnt
 - h, 356
 - h-corollary, 369
 - h2, 356
 - h2-corollary, 369
- finite-number-queue-table-table, 357
- finite-number-queuep, 352–356, 363, 365–370, 375, 377, 396, 407, 411, 413, 416–418, 420, 425–427, 738, 802, 826, 851, 865
- finite-number-queuep-array-deq, 354
- finite-number-queuep-array-deq-
 - finite-number-queue-corollary, 366
 - finite-number-queue-corollary-for-ibuffers, 417
 - finite-number-queue-corollary-for-mbuffers, 368
 - finite-number-queue-corollary-for-obuffers, 426
- finite-number-queuep-array-enq, 354
- finite-number-queuep-array-enq-
 - finite-number-queue-corollary, 366
 - finite-number-queue-corollary-for-ibuffers, 417
 - finite-number-queue-corollary-for-mbuffers, 368

- finite-number-queue-corollary-for-obuffers, 426
- finite-number-queuep-array-enq-interrupting-input-port, 802
- nterrupting-input-port2, 825
 - nterrupting-output-port, 850
 - nterrupting-output-port2, 865
- finite-number-queuep-array-qrep-lace, 370
- lace-corollary-for-ibuffers, 418
- finite-number-queuep-current-ibuffer, 418
- finite-number-queuep-current-mbuffer, 407
- uffer-for-receive, 407
- finite-number-queuep-current-obuffer, 427
- finite-number-queuep-facts, 355
- finite-number-queuep-getnth, 355
- finite-number-queuep-getnth2, 356
- finite-number-queuep-getnth2-corollary, 368
- finite-number-queuep-os-clock-new-readyq, 738
- finite-number-queuep-os-readyq, 363
- finite-number-sub1, 50
- finite-number-tablep, 60, 61, 328
- finite-number-tablep-append, 61
- finite-number-tablep-list, 61
- finite-numberp, 49–52, 65, 66, 86–89, 91, 98, 99, 102, 103, 116–118, 121–127, 129, 131–133, 136, 137, 139–141, 143, 144, 146–148, 152, 154, 156, 158, 160, 163, 195, 206, 207, 209, 210, 213, 276, 290, 291, 294–297, 302–309, 314, 316, 317, 321, 324, 328, 354, 365, 369, 370, 374, 375, 378, 380, 389, 390, 405, 410, 418, 758, 774, 1105
- finite-numberp-add1-address-space-sp, 213
- finite-numberp-add1-tm-sp, 158
- finite-numberp-address-space-cc, 206
- finite-numberp-address-space-error, 207
- finite-numberp-address-space-pc, 207
- finite-numberp-address-space-sp, 207
- finite-numberp-address-space-svcflag, 207
- cid, 207
- finite-numberp-ak-clock, 297
- finite-numberp-ak-destid, 306
- finite-numberp-ak-message, 306
- finite-numberp-ak-rwstate, 296
- finite-numberp-ak-srcid, 306
- finite-numberp-ak-taskid, 295
- finite-numberp-alu-value-tm-alu
 - decr, 86
 - decr-mod, 86
 - incr, 87
 - incr-mod, 87
 - mod, 88
 - mult, 89
 - plus, 89
- finite-numberp-from-smaller-lub, 49
- finite-numberp-getnth, 50
- finite-numberp-getnth-from-current-mbuffer, 410
- nt-mbuffer1, 410
- finite-numberp-getnth-os-status-table, 405
- finite-numberp-interrupting-ak-output-port, 309
- finite-numberp-interrupting-ak-input-port, 307
- finite-numberp-mapup-cc, 374
- finite-numberp-mapup-error, 374
- finite-numberp-mapup-svcflag, 374
- finite-numberp-mapup-svcid, 374
- finite-numberp-member, 49
- finite-numberp-min-difference-mlength-base-limit, 131
- finite-numberp-os-destid, 758
- finite-numberp-os-message, 276
- finite-numberp-os-srcid, 774
- finite-numberp-os-taskid, 365

finite-numberp-qfirst, 66
 finite-numberp-remainder, 50
 finite-numberp-tm-base, 121
 finite-numberp-tm-cc, 121
 finite-numberp-tm-cc-value, 116
 finite-numberp-tm-clock, 122
 finite-numberp-tm-decr, 91
 finite-numberp-tm-error, 121
 finite-numberp-tm-fetch, 132
 finite-numberp-tm-fetch-from-address-space, 209
 finite-numberp-tm-fetch-from-memory-in-supervisor-mode, 131
 finite-numberp-tm-fetch-in-supervisor-mode, 156
 finite-numberp-tm-good-address, 160
 finite-numberp-tm-good-address-for-address-space, 209
 finite-numberp-tm-ichar-getnth-crock, 127
 finite-numberp-tm-incr, 91
 finite-numberp-tm-incr-memlength, 213
 finite-numberp-tm-incrn, 91
 finite-numberp-tm-interrupting-output-port, 125
 output-port-crock, 127
 output-port-tm-wordlub, 126
 finite-numberp-tm-interrupting-input-port, 125
 nput-port-crock, 127
 nput-port-tm-wordlub, 125
 finite-numberp-tm-limit, 121
 finite-numberp-tm-overflow-char, 127
 -tm-wordlub, 127
 finite-numberp-tm-pack-psw, 122
 finite-numberp-tm-pack-psw1, 98
 finite-numberp-tm-pc, 122
 finite-numberp-tm-rwstate, 123
 finite-numberp-tm-slimit, 121
 finite-numberp-tm-sp, 122
 finite-numberp-tm-svcflag, 122
 finite-numberp-tm-svcid, 122
 finite-numberp-tm-svmode, 122
 finite-numberp-tm-unpack-cc, 98
 finite-numberp-tm-unpack-error, 98
 finite-numberp-tm-unpack-svcflag, 99
 finite-numberp-tm-unpack-svcid, 99
 finite-segment-tablep, 194, 364, 374
 finite-segment-tablep-os-segment-table, 364
 firstn, 29, 37, 53, 54, 56, 61, 62, 346, 356, 376
 firstn-firstn, 54
 firstn-fixlength, 62
 firstn-for-length, 54
 firstn-n-times, 346
 firstn-nthcdr, 54
 firstn-with-small-recursion, 346, 347
 firstn-zero, 54
 fixlength, 29, 32, 33, 48, 52, 53, 62, 393
 fixlength-idempotence, 48
 fixlength-noop, 48
 fixlength-of-length-0, 48
 flatten, 55, 57, 61
 getnth, 28, 30, 33–36, 38, 40, 41, 45, 47, 50, 53, 56, 58–62, 67–74, 77–84, 97, 100–102, 114, 120, 121, 123–130, 195, 272, 275, 277–282, 285–289, 291, 292, 294–297, 299–306, 312, 314–316, 352–357, 362, 363, 365–372, 374, 379, 380, 386, 387, 391–394, 396–399, 403–405, 407, 408, 410–418, 420–427, 429–436, 477, 480, 482, 485, 492, 494, 496, 497, 512, 514, 517, 519, 526, 529, 531, 532, 582, 600, 602, 603, 605, 621, 623–625, 630, 641, 643, 645, 647, 648, 650, 652, 668, 670–672, 677, 689, 699, 701–704, 709, 716, 718, 720, 730, 732, 734–737, 740, 741, 744, 747, 748, 751, 754, 755,

757–760, 762–764, 767, 769–
 771, 773–776, 778–780, 782,
 783, 785, 786, 788, 790, 793,
 795, 796, 798–803, 806–808,
 810–813, 815–817, 819, 820,
 822, 823, 825–827, 829–831,
 834–836, 838–840, 842–844,
 846–848, 850–852, 854–856,
 858–860, 862, 863, 865, 866,
 868–871, 873–875, 877, 878,
 880, 881, 889, 898, 899, 909,
 917, 919, 920, 924, 926, 927,
 934, 943–946, 951, 953, 954,
 960, 961, 968, 969, 975, 977,
 978, 985, 990, 991, 996, 998,
 1000, 1003, 1005, 1007, 1010,
 1012, 1013, 1016, 1017, 1020,
 1021, 1023, 1024, 1026–1044,
 1046–1048, 1051–1053, 1055,
 1056, 1060, 1063, 1064, 1067,
 1070, 1074, 1078–1091, 1094,
 1095, 1099, 1100, 1102
 getnth-0-2, 33
 getnth-1-2, 34
 getnth-after-putseg, 36
 getnth-append-casesplit, 34
 getnth-append-for-length-left, 34
 getnth-append-left, 35
 getnth-append-right, 35
 getnth-before-putseg, 36
 getnth-getseg, 36
 getnth-getseg-induction, 35
 getnth-getseg-inverse, 36
 getnth-getseg-zero-index, 35
 getnth-mapup-queue-list, 372
 getnth-mapup-tasks, 391
 getnth-mapup-tasks-corollary, 1100
 getnth-mapup-tasks-corollary2, 391
 getnth-nthcdr, 58
 getnth-os-code, 393
 getnth-plus-1-is-cadr-getseg, 62
 getnth-plus-1-is-cadr-getseg-inve
 rse, 62
 getnth-plus-1-is-cadr-nthcdr, 61
 getnth-putnth-casesplit, 35
 getnth-putnth-coincidence, 35
 getnth-putnth-coincidence1, 47
 getnth-putnth-non-interference, 35
 getnth-putseg-coincidence, 36
 getnth-table, 58
 getnth-table-inverse, 58
 getnth-tm-reglength-tm-cpu, 387
 getnth-with-non-number-index, 34
 getnth2, 47, 48, 84, 278, 295, 356,
 368, 416
 getnth2-putnth2-coincidence, 47
 getnth2-putnth2-non-interference
 1, 47
 2, 48
 getseg, 28, 30–32, 35, 36, 40–47, 51,
 52, 55, 56, 58, 62, 69, 71,
 72, 74, 77, 79, 81–83, 130,
 195–200, 216, 348, 352–354,
 357, 370, 376, 379, 399, 403,
 432, 434, 435, 887, 893, 897,
 903, 907, 913, 921, 928, 932,
 938, 947, 955, 959, 965, 971,
 976, 982, 988, 994, 1001,
 1007, 1014, 1018, 1022, 1025,
 1031, 1037, 1038, 1044, 1048,
 1053, 1057, 1064, 1071, 1075,
 1079, 1085, 1091, 1095
 getseg-across-append, 41
 getseg-add1-cons, 40
 getseg-add1-length, 40
 getseg-add1-length2, 41
 getseg-after-putnth, 41
 getseg-after-putseg, 42
 getseg-append-left, 41
 getseg-append-right, 41
 getseg-around-putseg, 44
 getseg-around-putseg-proof, 43
 getseg-before-putnth, 42
 getseg-before-putseg, 42
 getseg-equals-firstn-nthcdr, 56
 getseg-fixlength, 62
 getseg-for-length, 56
 getseg-getseg, 42

- getseg-length-1, 40
- getseg-length-1-rule, 41
- getseg-must-be-nil, 45
- getseg-nthcdr, 58
- getseg-nthcdr-proof, 58
- getseg-of-length-zero, 40
- getseg-putnth-coincidence, 42
- getseg-putseg-induction, 43
- getseg-tm-memory-os-clock-inter
 - rupt-handler, 887
- getseg-tm-memory-os-error-handle
 - r-path1, 893
 - r-path2, 897
- getseg-tm-memory-os-running-inp
 - ut-handler-path1, 1025
 - ut-handler-path2, 1031
 - ut-handler-path3, 1037
 - ut-handler-path4, 1043
 - ut-handler-path5, 1048
 - ut-handler-path6, 1052
- getseg-tm-memory-os-running-outp
 - ut-handler-path1, 1079
 - ut-handler-path2, 1085
 - ut-handler-path3, 1091
 - ut-handler-path4, 1095
- getseg-tm-memory-os-svc-receive
 - handler-path1, 928
 - handler-path2, 932
 - handler-path3, 938
 - handler-path4, 947
- getseg-tm-memory-os-svc-send-hand
 - ndler-path1, 903
 - ndler-path2, 907
 - ndler-path3, 913
 - ndler-path4, 921
- getseg-tm-memory-os-svc-tyi-hand
 - ndler-path1, 955
 - ndler-path2, 959
 - ndler-path3, 965
- getseg-tm-memory-os-svc-tyo-hand
 - ndler-path1, 971
 - ndler-path2, 976
 - ndler-path3, 982
 - ndler-path4, 988
- getseg-tm-memory-os-waiting-inp
 - ut-handler-path1, 993
 - ut-handler-path2, 1001
 - ut-handler-path3, 1007
 - ut-handler-path4, 1014
 - ut-handler-path5, 1018
 - ut-handler-path6, 1021
- getseg-tm-memory-os-waiting-outp
 - ut-handler-path1, 1057
 - ut-handler-path2, 1064
 - ut-handler-path3, 1071
 - ut-handler-path4, 1075
- getseg-with-non-number-index, 40
- getseg-with-small-recursion, 347
- getseg-within-putseg, 42
- good-address-space, 205–216, 276, 277,
 - 290, 291, 294, 375
- good-address-space-base-limit-p
 - roperty, 208
- good-address-space-getnth, 291
- good-address-space-getnth-from-
 - ak, 294
- good-address-space-list, 290, 291, 294,
 - 375
- good-address-space-list-ak-pstate
 - s, 294
- good-address-space-list-mapup-o
 - s-tasks, 375
- good-address-space-list-mapup-t
 - asks, 375
- good-address-space-list-putnth, 291
- good-address-space-mapup-task, 375
- good-address-space-os-store-mes
 - sage, 276
 - sage1, 276
- good-address-space-properties, 208
- good-address-space-tm-decrement
 - clock, 216
- good-address-space-tm-execute, 216
- good-address-space-tm-execute-a
 - dd, 210
- good-address-space-tm-execute-b
 - ranch, 211
 - ranch-not-zero, 211

- ranch-on-zero, 211
- good-address-space-tm-execute-binary, 216
- good-address-space-tm-execute-call, 211
 - compare, 211
- good-address-space-tm-execute-decr, 211
 - cr-mod, 212
- good-address-space-tm-execute-incr, 212
 - ncr-mod, 212
 - nstruction, 216
- good-address-space-tm-execute-load-base, 212
 - oad-limit, 212
 - oad-psw, 212
- good-address-space-tm-execute-mod, 213
 - ove, 213
 - ultiply, 213
- good-address-space-tm-execute-nullary, 215
- good-address-space-tm-execute-post-output-interrupt, 215
- good-address-space-tm-execute-run, 214
- good-address-space-tm-execute-return, 214
- good-address-space-tm-execute-set-clock, 214
- good-address-space-tm-execute-start-output, 215
- good-address-space-tm-execute-svc, 215
 - c-return, 215
- good-address-space-tm-execute-test-iport, 214
 - st-oport, 214
- good-address-space-tm-execute-unary, 215
- good-address-space-tm-execute-wait, 214
- good-address-space-tm-fetch-execute, 216
- good-address-space-tm-set-cc, 210
- good-address-space-tm-set-error, 210
- good-address-space-tm-set-pc, 210
- good-address-space-tm-set-sp, 210
- good-address-space-tm-set-svcflag, 210
- good-address-space-tm-set-svcid, 210
- good-address-space-tm-store, 209
- good-ak, 293–298, 302–310, 314–317, 377
 - good-ak-ak-block-input, 305
 - good-ak-ak-block-output, 304
 - good-ak-ak-block-receive, 303
 - good-ak-ak-block-send, 303
 - good-ak-ak-clock-interrupt-handler, 303
 - good-ak-ak-error-handler, 302
 - good-ak-ak-execute-input, 305
 - good-ak-ak-execute-input-from-buffer, 305
 - good-ak-ak-execute-output, 305
 - good-ak-ak-execute-output-to-buffer, 305
 - good-ak-ak-execute-receive, 304
 - good-ak-ak-execute-receive-from-buffer, 304
 - good-ak-ak-execute-send, 303
 - good-ak-ak-execute-send-to-buffer, 303
 - good-ak-ak-input-interrupt-handler, 308
 - good-ak-ak-output-interrupt-handler, 309
 - good-ak-ak-post-input-interrupt, 309
 - good-ak-ak-post-interrupt, 310
 - good-ak-ak-post-interrupt-noop, 309
 - good-ak-ak-post-output-interrupt, 309
 - good-ak-ak-private-step, 306
 - good-ak-ak-processor, 310
 - good-ak-ak-running-input-interrupt-handler, 307
 - upt-handler-case1, 307
 - upt-handler-case2, 307

good-ak-ak-running-output-interrupt-handler, 309
 rupt-handler-case1, 308
 rupt-handler-case2, 308
 good-ak-ak-step, 309
 good-ak-ak-svc-handler, 306
 good-ak-ak-waiting-input-interrupt-handler, 307
 upt-handler-case1, 307
 upt-handler-case2, 307
 good-ak-ak-waiting-output-interrupt-handler, 308
 rupt-handler-case1, 308
 rupt-handler-case2, 308
 good-ak-construction, 292
 good-ak-ibuffers-properties, 305
 good-ak-mapup-os, 377
 good-ak-obuffers-properties, 304
 good-ak-properties, 298
 good-cpu, 128, 129
 good-cpu-getnth, 129
 good-cpu-list, 128, 129, 363, 404, 772, 787
 good-cpu-list-os-new-task-table, 404
 good-cpu-list-os-task-table, 363
 good-cpu-list-os-task-table-with-delivered-character, 787
 h-delivered-message, 771
 good-cpu-list-putnth, 129
 good-cpu-tm-cpu, 129
 good-memory-address-implies-non-zero-limit-for-address-space, 208
 good-os, 360, 362–365, 369, 370, 373–378, 380–388, 390–393, 404, 407, 410–412, 418, 420, 421, 425, 427, 429, 432–436, 439, 440, 442, 444, 446–448, 450, 452, 454, 455, 457, 458, 460, 461, 463, 465, 467, 470, 472, 475, 477, 480, 482, 484, 486–489, 492, 494, 496, 498, 500, 503, 505, 507, 509, 512, 514, 517, 519, 521, 522, 524, 526, 528, 531–533, 535, 537, 539, 541, 543, 546, 548–550, 552, 553, 555, 557, 560, 562, 564, 566, 568, 570, 572, 573, 575–578, 580, 582, 585, 588, 590, 593, 596, 598, 600, 601, 603, 605, 608, 611, 613, 616, 618, 620, 621, 623–625, 628, 630, 633, 636, 638, 641, 643, 645, 647, 648, 650, 652, 655, 658, 661, 663, 665, 666, 668, 670, 671, 673–675, 677, 680, 682, 685, 687, 689, 691, 694, 695, 697, 699, 700, 702–704, 706, 709, 711, 714, 716, 718, 720, 722, 725, 727, 729, 730, 732, 733, 735–740, 743, 744, 746, 747, 750, 753, 754, 756–759, 761–763, 765, 766, 768–775, 777–779, 781, 782, 784, 785, 787–789, 791, 792, 794–799, 801–807, 809–816, 818, 819, 821, 822, 824–826, 828–830, 832–835, 837–843, 845–848, 850, 851, 853–855, 857–859, 862, 864, 865, 868–870, 872–874, 877, 879–884, 886, 887, 889–993, 996–1001, 1003–1007, 1009–1057, 1060–1064, 1066–1105
 good-os-bos-processor, 886
 good-os-btimed-tm-processor, 886
 good-os-os-clock-interrupt-handler, 740
 good-os-os-error-handler, 747
 good-os-os-error-handler-path1, 743
 good-os-os-error-handler-path2, 747
 good-os-os-input-interrupt-handler, 883
 good-os-os-output-interrupt-handler, 883
 good-os-os-processor, 884
 good-os-os-running-input-handler, 850
 r-path1, 830

- r-path2, 835
- r-path3, 839
- r-path4, 843
- r-path5, 846
- r-path6, 850
- good-os-os-running-output-handle
 - r, 881
 - r-path1, 869
 - r-path2, 874
 - r-path3, 877
 - r-path4, 880
- good-os-os-step, 884
- good-os-os-svc-handler, 882
- good-os-os-svc-receive-handler, 779
- good-os-os-svc-receive-handler-p
 - ath1, 766
 - ath2, 769
 - ath3, 774
 - ath4, 778
- good-os-os-svc-send-handler, 763
- good-os-os-svc-send-handler-pat
 - h1, 750
 - h2, 754
 - h3, 758
 - h4, 762
- good-os-os-svc-tyi-handler, 789
- good-os-os-svc-tyi-handler-path
 - 1, 782
 - 2, 785
 - 3, 789
- good-os-os-svc-tyo-handler, 802
- good-os-os-svc-tyo-handler-path
 - 1, 792
 - 2, 795
 - 3, 799
 - 4, 802
- good-os-os-waiting-input-handle
 - r, 825
 - r-path1, 807
 - r-path2, 811
 - r-path3, 816
 - r-path4, 819
 - r-path5, 822
 - r-path6, 825
- good-os-os-waiting-output-handle
 - r, 865
 - r-path1, 855
 - r-path2, 859
 - r-path3, 862
 - r-path4, 865
- good-os-properties, 365
- good-os-properties-crock, 388
- good-os-properties1, 395
- good-os-timed-tm-step, 886
- good-os-tm-decrement-clock, 385
- good-os-tm-execute, 385
- good-os-tm-execute-add, 381
- good-os-tm-execute-binary, 384
- good-os-tm-execute-branch, 381
- good-os-tm-execute-branch-not-zero, 381
- good-os-tm-execute-branch-on-zero, 381
- good-os-tm-execute-call, 381
- good-os-tm-execute-compare, 381
- good-os-tm-execute-decr, 381
- good-os-tm-execute-decr-mod, 382
- good-os-tm-execute-incr, 382
- good-os-tm-execute-incr-mod, 382
- good-os-tm-execute-instruction, 384
- good-os-tm-execute-load-base, 382
- good-os-tm-execute-load-limit, 382
- good-os-tm-execute-load-psw, 382
- good-os-tm-execute-mod, 382
- good-os-tm-execute-move, 383
- good-os-tm-execute-multiply, 383
- good-os-tm-execute-nullary, 384
- good-os-tm-execute-post-output-interrupt, 383
- good-os-tm-execute-return, 383
- good-os-tm-execute-run, 383
- good-os-tm-execute-set-clock, 383
- good-os-tm-execute-start-output, 383
- good-os-tm-execute-svc, 383
- good-os-tm-execute-svc-return, 384
- good-os-tm-execute-test-iport, 384
- good-os-tm-execute-test-oport, 384
- good-os-tm-execute-unary, 384

- good-os-tm-execute-wait, 384
- good-os-tm-fetch-execute, 385
- good-os-tm-post-interrupt, 385
- good-os-tm-set-cc, 378
- good-os-tm-set-error, 378
- good-os-tm-set-iptos, 385
- good-os-tm-set-oports, 385
- good-os-tm-set-pc, 378
- good-os-tm-set-sp, 378
- good-os-tm-set-svcflag, 378
- good-os-tm-set-svcid, 378
- good-os-tm-store, 380
- good-os-tm-store-in-memory, 380
- good-os-tm-store-in-regmem, 378
- good-status, 291, 296, 304
- good-status-getnth, 291
- good-status-getnth-from-ak-status, 296
- good-status-list, 291, 296, 363, 743, 746, 805, 810, 814, 829, 833, 838, 853, 858, 868, 872
- good-status-list-ak-status, 296
- good-status-list-os-status-table, 363
 - os-error-handler-path1, 743
 - os-error-handler-path2, 746
 - os-running-input-handler-path1, 829
 - os-running-input-handler-path2, 833
 - os-running-input-handler-path3, 838
 - os-running-output-handler-path1, 868
 - os-running-output-handler-path2, 872
 - os-waiting-input-handler-path1, 805
 - os-waiting-input-handler-path2, 810
 - os-waiting-input-handler-path3, 814
 - os-waiting-output-handler-path1, 853
 - os-waiting-output-handler-path2, 858
- good-status-list-putnth, 291
- good-status-properties, 304
- good-task, 277
- good-task-buffer, 276–278, 294, 295, 375
- good-task-buffer-getnth, 277
- good-task-buffer-getnth-from-ak-ibuffers, 294
- good-task-buffer-getnth-from-ak-obuffers, 295
- good-task-buffer-getnth2, 278
- good-task-buffer-getnth2-from-ak-mbuffers, 295
- good-task-buffer-list, 276–278, 294, 295, 375–377
- good-task-buffer-list-ak-ibuffers, 294
- good-task-buffer-list-ak-obuffers, 295
- good-task-buffer-list-firstn, 376
- good-task-buffer-list-getnth, 278
- good-task-buffer-list-getseg, 376
- good-task-buffer-list-mapup-os-obuffers, 376
- good-task-buffer-list-mapup-os-ibuffers, 376
- good-task-buffer-list-mapup-queue-list, 375
- good-task-buffer-list-nthcdr, 376
- good-task-buffer-list-putnth, 278
- good-task-buffer-mapup-queue, 375
- good-task-buffer-properties, 277
- good-task-buffer-table, 277–279, 295, 377
- good-task-buffer-table-ak-mbuffers, 295
- good-task-buffer-table-mapup-os-mbuffers, 377
- good-task-buffer-table-putnth, 278
- good-task-buffer-table-putnth2, 278
- good-task-buffer-table-table, 377
- good-tm, 120–127, 129–133, 135–144, 146–148, 150–152, 154–160,

162, 163, 217–220, 225–246,
 248–254, 256–260, 262–271,
 360, 378, 379, 405, 410, 420,
 429, 736–742, 744–749, 751–
 753, 755–757, 759–761, 763–
 773, 775–777, 779–805, 807–
 809, 812–814, 816–828, 831–
 833, 835–837, 840–845, 847–
 853, 855–857, 859–864, 866–
 868, 870–872, 874–880, 919,
 926, 945, 953, 970, 987, 992,
 1030, 1036, 1042, 1047, 1051,
 1056, 1083, 1089, 1094, 1098
 good-tm-iport, 118, 123, 124, 297,
 306
 good-tm-iport-array, 118, 123, 124,
 126, 150, 163, 290, 297, 385
 good-tm-iport-array-ak-ports, 297
 good-tm-iport-array-putnth, 124
 good-tm-iport-array-putnth-into
 -tm-ports, 124
 good-tm-iport-array-putnth1, 290
 good-tm-iport-array-tm-ports, 123
 good-tm-iport-getnth, 123
 good-tm-iport-getnth-from-ak-ip
 orts, 297
 good-tm-iport-getnth-from-tm-ip
 orts, 124
 good-tm-iport-getnth-properties, 125
 good-tm-iport-properties, 306
 good-tm-ports-tm-clear-input-i
 nterrupt, 126
 good-tm-ports-tm-post-input-inte
 rrupt, 163
 good-tm-oport, 118, 119, 124, 297,
 308
 good-tm-oport-array, 119, 123, 124,
 126, 151, 159, 290, 297, 385
 good-tm-oport-array-ak-ports, 297
 good-tm-oport-array-putnth, 124
 good-tm-oport-array-putnth-into
 -tm-ports, 124
 good-tm-oport-array-putnth1, 290
 good-tm-oport-array-tm-ports, 123
 good-tm-oport-getnth, 124
 good-tm-oport-getnth-from-ak-op
 orts, 297
 good-tm-oport-getnth-from-tm-op
 orts, 124
 good-tm-oport-getnth-properties, 126
 good-tm-oport-properties, 308
 good-tm-ports-tm-clear-output-i
 nterrupt, 126
 good-tm-ports-tm-post-output-i
 nterrupt, 159
 good-tm-ports-tm-start-output, 159
 good-tm-os, 360
 good-tm-os-clock-interrupt-hand
 ler, 737
 good-tm-os-error-handler-path1, 740
 good-tm-os-error-handler-path2, 744
 good-tm-os-intended-clock-inter
 rupt, 736
 good-tm-os-intended-error-inter
 rupt, 740
 good-tm-os-intended-input-inter
 rupt, 802
 good-tm-os-intended-output-inte
 rrupt, 850
 good-tm-os-intended-svc-receive
 -interrupt, 763
 good-tm-os-intended-svc-send-inte
 rrupt, 747
 good-tm-os-intended-svc-tyi-inte
 rrupt, 779
 good-tm-os-intended-svc-tyo-inte
 rrupt, 789
 good-tm-os-running-input-handle
 r-path1, 826
 r-path2, 830
 r-path3, 835
 r-path4, 840
 r-path5, 843
 r-path6, 847
 good-tm-os-running-output-handle
 r-path1, 865
 r-path2, 870
 r-path3, 874

- r-path4, 877
- good-tm-os-svc-recv-handler-p
 - ath1, 763
 - ath2, 766
 - ath3, 770
 - ath4, 775
- good-tm-os-svc-send-handler-pat
 - h1, 747
 - h2, 750
 - h3, 754
 - h4, 759
- good-tm-os-svc-tyi-handler-path
 - 1, 779
 - 2, 782
 - 3, 785
- good-tm-os-svc-tyo-handler-path
 - 1, 789
 - 2, 792
 - 3, 796
 - 4, 799
- good-tm-os-waiting-input-handle
 - r-path1, 803
 - r-path2, 807
 - r-path3, 812
 - r-path4, 816
 - r-path5, 819
 - r-path6, 822
- good-tm-os-waiting-output-handle
 - r-path1, 851
 - r-path2, 855
 - r-path3, 859
 - r-path4, 862
- good-tm-properties, 130
- good-tm-tm-decrement-clock, 160
- good-tm-tm-execute, 160
- good-tm-tm-execute-add, 155
- good-tm-tm-execute-binary, 160
- good-tm-tm-execute-branch, 155
- good-tm-tm-execute-branch-not-ze
 - ro, 155
- good-tm-tm-execute-branch-on-ze
 - ro, 155
- good-tm-tm-execute-call, 155
- good-tm-tm-execute-clock-interr
 - upt, 162
- good-tm-tm-execute-compare, 155
- good-tm-tm-execute-decr, 155
- good-tm-tm-execute-decr-mod, 156
- good-tm-tm-execute-error-interr
 - upt, 162
- good-tm-tm-execute-incr, 156
- good-tm-tm-execute-incr-mod, 156
- good-tm-tm-execute-input-interr
 - upt, 162
- good-tm-tm-execute-instruction, 160
- good-tm-tm-execute-load-base, 156
- good-tm-tm-execute-load-limit, 156
- good-tm-tm-execute-load-psw, 157
- good-tm-tm-execute-mod, 157
- good-tm-tm-execute-move, 157
- good-tm-tm-execute-multiply, 157
- good-tm-tm-execute-nullary, 159
- good-tm-tm-execute-output-inter
 - rupt, 163
- good-tm-tm-execute-post-output-i
 - nterrupt, 159
- good-tm-tm-execute-return, 158
- good-tm-tm-execute-run, 158
- good-tm-tm-execute-set-clock, 158
- good-tm-tm-execute-start-output, 159
- good-tm-tm-execute-svc, 159
- good-tm-tm-execute-svc-interrupt, 162
- good-tm-tm-execute-svc-return, 159
- good-tm-tm-execute-test-iport, 158
- good-tm-tm-execute-test-oport, 158
- good-tm-tm-execute-unary, 160
- good-tm-tm-execute-wait, 158
- good-tm-tm-fetch-execute, 162
- good-tm-tm-post-interrupt, 163
- good-tm-tm-processor, 163
- good-tm-tm-set-base, 139
- good-tm-tm-set-cc, 136
- good-tm-tm-set-clock, 148
- good-tm-tm-set-error, 137
- good-tm-tm-set-iports, 150
- good-tm-tm-set-limit, 140
- good-tm-tm-set-memory, 135
- good-tm-tm-set-oports, 151

good-tm-tm-set-pc, 152
 good-tm-tm-set-regs, 133
 good-tm-tm-set-rwstate, 147
 good-tm-tm-set-slimit, 141
 good-tm-tm-set-sp, 154
 good-tm-tm-set-svcflag, 144
 good-tm-tm-set-svcid, 146
 good-tm-tm-set-svmode, 143
 good-tm-tm-step, 163
 good-tm-tm-store, 133

half-plus, 10

if-times-then-divides, 9
 incr-dc-position, 325, 328
 incr-instruction-position, 325, 328
 incr-mod, 63, 64, 67, 70, 71, 80, 81, 85, 88, 392
 incr-mod-decr-mod, 64
 incr-mod-equals-remainder-add1, 63
 index-ready-set, 292, 298–302, 357, 370, 743, 746, 750, 753, 757, 765, 768, 773, 781, 784, 791, 794, 805, 810, 815, 829, 833, 838, 853, 858, 868, 873
 index-ready-set-putnth-ready-status, 301
 index-ready-set-putnth-unready-status, 299
 index-within-length-of-ibuffer, 420
 index-within-length-of-mbuffer, 411
 index-within-length-of-mbuffer-for-recv, 413
 insert, 23–25, 27, 28, 301
 insert-remove, 24
 integer-equality-crock, 19
 integer-induction-instance, 42
 integer-inequality-casesplit, 19
 integer-nthcdr-induction, 57, 58

length, 23, 24, 29, 31–44, 47, 48, 50–66, 68–71, 73, 77, 79–84, 111, 120, 121, 123–126, 128–130, 133, 135, 150, 151, 159, 163, 195, 200–204, 206, 209, 264, 265, 273, 274, 276–278, 290, 291, 294–297, 299–302, 304, 305, 310–312, 314, 316, 317, 322, 324, 325, 328, 348–350, 352, 353, 355–357, 365, 367–369, 372–377, 379, 380, 385, 386, 397–399, 404, 415, 416, 424, 425, 431, 433, 435, 1105

length-address-space-memory, 206
 length-address-space-regs, 206
 length-ak-ibuffers, 294
 length-ak-iports, 297
 length-ak-mbuffers, 295
 length-ak-obuffers, 294
 length-ak-oports, 297
 length-ak-pstates, 294
 length-ak-pstates-ak-execute-input, 311
 ut-from-buffer, 311
 length-ak-pstates-ak-execute-output, 311
 ut-to-buffer, 311
 length-ak-pstates-ak-execute-receive, 311
 ceive-from-buffer, 311
 length-ak-pstates-ak-execute-send, 310
 nd-to-buffer, 310
 length-ak-pstates-ak-private-step, 312
 length-ak-pstates-ak-step, 312
 length-ak-pstates-ak-svc-handle-r, 312
 length-ak-readyq, 296
 length-ak-status, 296
 length-append, 32
 length-array-deq, 71
 length-array-enq, 70
 length-array-nonlast, 73
 length-array-qarray, 69
 length-array-qreplace, 77
 length-array-queue, 68
 length-delta-segment, 79

- length-delta-segment-a-a, 81
- length-delta-segment-decr-mod-a
rg1, 82
- length-deq, 65
- length-enq, 64
- length-finite-number-queue, 353
- length-finite-number-queue-coro
llary, 365
 - llary-for-ibuffers, 416
 - llary-for-mbuffers, 367
 - llary-for-obuffers, 425
- length-firstn, 53
- length-fixlength, 33
- length-flatten, 57
- length-getnth-from-table, 56
- length-getnth-table, 56
- length-getseg, 32
- length-insert, 24
- length-mapup-cpu, 373
- length-mapup-os-ibuffers, 375
- length-mapup-os-mbuffers, 376
- length-mapup-os-obuffers, 376
- length-mapup-os-tasks, 373
- length-mapup-queue, 80
- length-mapup-queue-is-qcurren
gth, 84
- length-mapup-queue-list, 372
- length-mapup-regs, 373
- length-mapup-tasks, 373
- length-non-empty-queue, 64
- length-nonlast, 62
- length-nthcdr, 53
- length-of-nthcdr-goes-down, 55
- length-os-code, 352
- length-os-ibuffers, 349
- length-os-ibuffers-with-dequeue
d-character, 424
- length-os-ibuffers-with-enqueue
d-character, 425
 - d-overflow-character, 424
- length-os-ibuffers-with-qreplace
d-overflow-character, 424
- length-os-mbuffers, 350
- length-os-mbuffers-with-dequeue
d-message, 415
- length-os-mbuffers-with-enqueue
d-message, 415
- length-os-obuffers, 350
- length-os-obuffers-with-dequeue
d-character, 431
- length-os-obuffers-with-enqueue
d-character, 431
- length-os-readyq, 349
- length-os-segment-table, 349
- length-os-status-table, 349
- length-os-task-table, 348
- length-os-task-table-with-delive
red-character, 433
 - red-message, 435
- length-permutation, 24
- length-putnth, 32
- length-putnth2, 47
- length-putseg, 32
- length-qreplace, 65
- length-remove, 24
- length-table, 56
- length-tm-cpu, 129
- length-tm-ports, 123
- length-tm-memory, 120
- length-tm-memory-mapup-task, 375
- length-tm-memory-os-store-messa
ge, 276
- length-tm-memory-tm-execute, 204
- length-tm-memory-tm-execute-add,
200
- length-tm-memory-tm-execute-bin
ary, 204
- length-tm-memory-tm-execute-bra
nch, 200
 - nch-not-zero, 201
 - nch-on-zero, 201
- length-tm-memory-tm-execute-cal
l, 201
- length-tm-memory-tm-execute-comp
are, 201
- length-tm-memory-tm-execute-dec
r, 201
 - r-mod, 201

- length-tm-memory-tm-execute-incr, 201
- r-mod, 202
- length-tm-memory-tm-execute-instruction, 204
- length-tm-memory-tm-execute-load-base, 202
- d-limit, 202
- d-psw, 202
- length-tm-memory-tm-execute-mod, 202
- length-tm-memory-tm-execute-move, 202
- length-tm-memory-tm-execute-multiply, 202
- length-tm-memory-tm-execute-nullary, 204
- length-tm-memory-tm-execute-post-output-interrupt, 203
- length-tm-memory-tm-execute-return, 202
- length-tm-memory-tm-execute-run, 203
- length-tm-memory-tm-execute-set-clock, 203
- length-tm-memory-tm-execute-start-output, 203
- length-tm-memory-tm-execute-svc, 203
- return, 204
- length-tm-memory-tm-execute-test-iport, 203
- oport, 203
- length-tm-memory-tm-execute-unary, 204
- length-tm-memory-tm-execute-wait, 203
- length-tm-memory-tm-fetch-execute, 204
- length-tm-memory-tm-store, 200
- length-tm-oports, 123
- length-tm-regs, 121
- leq-casesplit, 19
- leq-delta, 67
- leq-number-of-ready-tasks-length-status, 300
- leq-plus, 18
- leq-plus-base-limit-getnth, 374
- leq-plus-base-limit-getnth-corollary, 374
- leq-plus-mapup-base-mapup-limit, 374
- leq-plus-times, 19
- leq-plus1, 18
- leq-plus2, 19
- leq-times, 18
- lessp-4-crock, 882
- lessp-alu-value-tm-alu-decr-mod-wordsizes, 86
- wordsizes-linear, 86
- lessp-alu-value-tm-alu-decr-wordsizes-linear, 86
- lessp-alu-value-tm-alu-incr-mod-wordsizes, 88
- wordsizes-linear, 88
- lessp-alu-value-tm-alu-incr-wordsizes-linear, 87
- lessp-alu-value-tm-alu-mod-wordsizes-linear, 88
- lessp-alu-value-tm-alu-mult-wordsizes-linear, 89
- lessp-alu-value-tm-alu-plus-wordsizes-linear, 89
- lessp-car-index-ready-set, 301
- lessp-car-index-ready-set-instance, 301
- lessp-crock-for-good-cpu-list-ostask-table-with-delivered-message, 771
- lessp-difference, 19
- lessp-getnth, 50
- lessp-getnth-linear, 50
- lessp-good-memory-address-limit, 218
- lessp-incr-mod, 64
- lessp-number-of-ready-tasks-length-status, 301

- lessp-plus-base-addr-memlength, 132
- lessp-plus-cancellation, 7
- lessp-plus-cancellation3, 8
- lessp-plus-fact-for-mbuffers-fo
 - r-receive1, 412
 - r-receive2, 412
- lessp-plus-fact-for-mbuffers1, 411
- lessp-plus-fact-for-mbuffers2, 411
- lessp-plus-facts-for-ibuffers, 420
- lessp-plus-facts-for-obuffers, 429
- lessp-qcurrlength-add1-qmaxlength, 69
- lessp-qfirst-mapup-queue-os-readyq, 377
- lessp-qhead-qmaxlength, 69
- lessp-qmaxlength-field-os-current-ibuffer-tm-wordlub, 420
- lessp-qmaxlength-field-os-current-obuffer-tm-wordlub, 429
- lessp-qtail-qmaxlength, 69
- lessp-quotient, 21
- lessp-quotient-induction, 20
- lessp-quotient-quotient, 21
- lessp-quotient-quotient-induction, 21
- lessp-quotient-remainder, 20
- lessp-quotient-remainder-crock, 20
- lessp-quotient-times, 20
- lessp-quotient1, 6
- lessp-remainder-divisor, 12
- lessp-remainder-remainder, 19
- lessp-remainder-sub1-crock, 18
- lessp-remainder1, 6
- lessp-remainder2, 5
- lessp-times, 19
- lessp-times-1, 5
- lessp-times-2, 6
- lessp-times-cancellation, 8
- lessp-times-quotient, 21
- lessp-tm-cc-value-tm-cclub, 116
- lessp-tm-cc-value-tm-cclub-linear, 116
- lessp-transitivity, 20
- limit, 194, 195, 363, 371, 374, 379, 380, 386, 387, 403, 404, 740, 747, 754, 758, 762, 769, 774, 778, 785, 788, 795, 798, 802, 806, 807, 811, 815, 816, 830, 834, 839, 843, 846, 850, 854, 859, 869, 874, 877, 880
- limited-associativity-of-plus, 406
- limited-associativity-of-plus2, 416
- list-getnth-0, 34
- list-of-getnths8, 130
- list-of-getnths9, 397
- list-of-putnths8, 398
- listp-append, 29
- listp-append-left, 29
- listp-getseg, 30
- listp-implies-non-zero-length, 29
- listp-nthcdr, 53
- listp-putnth, 30
- listp-putseg, 30
- listp-table, 57
- lookup, 318–323, 327, 331
- mapup-address-space, 216, 218–247, 249–272, 371, 387
- mapup-base, 371, 373, 374, 887, 893, 896, 897, 903, 907, 913, 920, 921, 928, 931, 932, 938, 947, 954, 955, 958, 959, 965, 971, 975, 976, 981, 982, 987, 988, 993, 994, 1000, 1001, 1007, 1014, 1018, 1021, 1022, 1025, 1031, 1037, 1043, 1044, 1048, 1052, 1053, 1057, 1064, 1071, 1074, 1075, 1078, 1079, 1084, 1085, 1090, 1091, 1095
- mapup-base-os-clock-interrupt-handler, 887
- mapup-base-os-error-handler-path1, 893
- mapup-base-os-error-handler-path2, 896
- mapup-base-os-running-input-handler-path1, 1025
- mapup-base-os-running-input-handler-path2, 1031
- mapup-base-os-running-input-handler-path3, 1037
- mapup-base-os-running-input-handler-path4, 1043

dler-path5, 1048
 dler-path6, 1052
 mapup-base-os-running-output-handler-path1, 1078
 ndler-path2, 1084
 ndler-path3, 1090
 ndler-path4, 1095
 mapup-base-os-svc-recv-handle-r-path1, 928
 r-path2, 931
 r-path3, 938
 r-path4, 946
 mapup-base-os-svc-send-handler-path1, 903
 ath2, 906
 ath3, 913
 ath4, 920
 mapup-base-os-svc-tyi-handler-path1, 954
 ath2, 958
 ath3, 965
 mapup-base-os-svc-tyo-handler-path1, 971
 ath2, 975
 ath3, 981
 ath4, 987
 mapup-base-os-waiting-input-handler-path1, 993
 dler-path2, 1000
 dler-path3, 1007
 dler-path4, 1014
 dler-path5, 1018
 dler-path6, 1021
 mapup-base-os-waiting-output-handler-path1, 1057
 ndler-path2, 1064
 ndler-path3, 1071
 ndler-path4, 1074
 mapup-cc, 371, 374, 890, 894, 899, 904, 910, 914, 915, 921, 922, 929, 935, 939–941, 948, 949, 955, 961, 966, 967, 972, 978, 982, 983, 988, 989, 996, 1004, 1010, 1015, 1018, 1022, 1027, 1033, 1039, 1045, 1049, 1054, 1060, 1067, 1072, 1075, 1080, 1086, 1092, 1096
 mapup-cc-os-clock-interrupt-handler, 890
 mapup-cc-os-error-handler-path1, 894
 mapup-cc-os-error-handler-path2, 899
 mapup-cc-os-running-input-handle-r-path1, 1027
 r-path2, 1033
 r-path3, 1039
 r-path4, 1045
 r-path5, 1049
 r-path6, 1054
 mapup-cc-os-running-output-handler-path1, 1080
 ler-path2, 1086
 ler-path3, 1092
 ler-path4, 1096
 mapup-cc-os-svc-recv-handler-path1, 928
 -path2, 935
 -path3, 941
 -path4, 949
 mapup-cc-os-svc-send-handler-path1, 904
 h2, 910
 h3, 915
 h4, 922
 mapup-cc-os-svc-tyi-handler-path1, 955
 h2, 961
 h3, 967
 mapup-cc-os-svc-tyo-handler-path1, 972
 h2, 978
 h3, 983
 h4, 989
 mapup-cc-os-waiting-input-handle-r-path1, 996
 r-path2, 1003
 r-path3, 1010
 r-path4, 1015
 r-path5, 1018

- r-path6, 1022
- mapup-cc-os-waiting-output-hand
 - ler-path1, 1060
 - ler-path2, 1067
 - ler-path3, 1072
 - ler-path4, 1075
- mapup-cpu, 370, 371, 373, 889, 893,
 - 899, 903, 909, 914, 915, 921,
 - 922, 928, 934, 939, 940, 947,
 - 948, 955, 961, 965, 966, 972,
 - 978, 982, 983, 988, 989, 996,
 - 1003, 1010, 1014, 1018, 1022,
 - 1026, 1032, 1038, 1044, 1049,
 - 1053, 1060, 1067, 1071, 1075,
 - 1080, 1085, 1091, 1096, 1103
- mapup-cpu-os-clock-interrupt-ha
 - ndler, 889
- mapup-cpu-os-error-handler-path
 - 1, 893
 - 2, 899
- mapup-cpu-os-running-input-hand
 - ler-path1, 1026
 - ler-path2, 1032
 - ler-path3, 1038
 - ler-path4, 1044
 - ler-path5, 1049
 - ler-path6, 1053
- mapup-cpu-os-running-output-han
 - dl-path1, 1079
 - dl-path2, 1085
 - dl-path3, 1091
 - dl-path4, 1096
- mapup-cpu-os-svc-recv-handle
 - r-path1, 928
 - r-path2, 934
 - r-path3, 940
 - r-path3-case1, 939
 - r-path3-case2, 939
 - r-path4, 948
 - r-path4-case1, 947
 - r-path4-case2, 948
- mapup-cpu-os-svc-send-handler-p
 - ath1, 903
 - ath2, 909
 - ath3, 914
 - ath3-case1, 914
 - ath3-case2, 914
 - ath4, 922
 - ath4-case1, 921
 - ath4-case2, 922
- mapup-cpu-os-svc-tyi-handler-pat
 - h1, 955
 - h2, 961
 - h3, 966
 - h3-case1, 965
 - h3-case2, 966
- mapup-cpu-os-svc-tyo-handler-pat
 - h1, 972
 - h2, 978
 - h3, 983
 - h3-case1, 982
 - h3-case2, 983
 - h4, 989
 - h4-case1, 988
 - h4-case2, 988
- mapup-cpu-os-waiting-input-hand
 - ler-path1, 996
 - ler-path2, 1003
 - ler-path3, 1010
 - ler-path4, 1014
 - ler-path5, 1018
 - ler-path6, 1022
- mapup-cpu-os-waiting-output-han
 - dl-path1, 1060
 - dl-path2, 1067
 - dl-path3, 1071
 - dl-path4, 1075
- mapup-cpu-tm-post-interrupt, 1103
- mapup-current-task-tm-fetch-exe
 - cute, 390
- mapup-error, 371, 374, 890, 894, 900,
 - 904, 910, 914, 915, 921–923,
 - 929, 935, 939–941, 948, 949,
 - 956, 962, 966, 967, 972, 979,
 - 982–984, 988, 989, 997, 1004,
 - 1011, 1015, 1019, 1022, 1027,
 - 1033, 1039, 1045, 1049, 1054,

1061, 1067, 1072, 1076, 1081,
 1086, 1092, 1096
 mapup-error-os-clock-interrupt-
 handler, 890
 mapup-error-os-error-handler-pat
 h1, 894
 h2, 900
 mapup-error-os-running-input-ha
 ndler-path1, 1027
 ndler-path2, 1033
 ndler-path3, 1039
 ndler-path4, 1045
 ndler-path5, 1049
 ndler-path6, 1054
 mapup-error-os-running-output-h
 andler-path1, 1080
 andler-path2, 1086
 andler-path3, 1092
 andler-path4, 1096
 mapup-error-os-svc-recv-hand
 ler-path1, 929
 ler-path2, 935
 ler-path3, 941
 ler-path4, 949
 mapup-error-os-svc-send-handler
 -path1, 904
 -path2, 910
 -path3, 915
 -path4, 922
 mapup-error-os-svc-tyi-handler-p
 ath1, 956
 ath2, 962
 ath3, 967
 mapup-error-os-svc-tyo-handler-p
 ath1, 972
 ath2, 978
 ath3, 983
 ath4, 989
 mapup-error-os-waiting-input-ha
 ndler-path1, 997
 ndler-path2, 1004
 ndler-path3, 1010
 ndler-path4, 1015
 ndler-path5, 1019
 ndler-path6, 1022
 mapup-error-os-waiting-output-h
 andler-path1, 1060
 andler-path2, 1067
 andler-path3, 1072
 andler-path4, 1076
 mapup-limit, 371, 373–375, 887, 893,
 896, 897, 903, 907, 913, 921,
 928, 932, 938, 947, 955, 958,
 959, 965, 971, 975, 976, 982,
 987, 988, 993, 994, 1000,
 1001, 1007, 1014, 1018, 1021,
 1022, 1025, 1031, 1037, 1038,
 1043, 1044, 1048, 1052, 1053,
 1057, 1064, 1071, 1075, 1079,
 1084, 1085, 1090, 1091, 1095
 mapup-limit-os-clock-interrupt-
 handler, 887
 mapup-limit-os-error-handler-pat
 h1, 893
 h2, 896
 mapup-limit-os-running-input-ha
 ndler-path1, 1025
 ndler-path2, 1031
 ndler-path3, 1037
 ndler-path4, 1043
 ndler-path5, 1048
 ndler-path6, 1052
 mapup-limit-os-running-output-h
 andler-path1, 1079
 andler-path2, 1084
 andler-path3, 1090
 andler-path4, 1095
 mapup-limit-os-svc-recv-hand
 ler-path1, 928
 ler-path2, 932
 ler-path3, 938
 ler-path4, 947
 mapup-limit-os-svc-send-handler
 -path1, 903
 -path2, 907
 -path3, 913
 -path4, 921
 mapup-limit-os-svc-tyi-handler-p

- ath1, 955
- ath2, 958
- ath3, 965
- mapup-limit-os-svc-tyo-handler-p
 - ath1, 971
 - ath2, 975
 - ath3, 982
 - ath4, 987
- mapup-limit-os-waiting-input-ha
 - ndler-path1, 993
 - ndler-path2, 1000
 - ndler-path3, 1007
 - ndler-path4, 1014
 - ndler-path5, 1018
 - ndler-path6, 1021
- mapup-limit-os-waiting-output-h
 - andler-path1, 1057
 - andler-path2, 1064
 - andler-path3, 1071
 - andler-path4, 1075
- mapup-os, 372, 377, 392, 892, 896,
 - 902, 906, 913, 920, 926, 927,
 - 931, 938, 946, 953, 954, 958,
 - 964, 971, 975, 981, 987, 993,
 - 1000, 1007, 1014, 1017, 1021,
 - 1025, 1031, 1037, 1043, 1047,
 - 1052, 1057, 1063, 1064, 1071,
 - 1074, 1078, 1084, 1090, 1094,
 - 1099–1101, 1103–1105
- mapup-os-ibuffers, 372, 375, 376, 425,
 - 891, 895, 901, 905, 911, 918,
 - 924, 930, 936, 944, 951, 957,
 - 963, 969, 973, 980, 985, 991,
 - 998, 1005, 1012, 1016, 1020,
 - 1023, 1029, 1035, 1041, 1046,
 - 1051, 1055, 1062, 1069, 1073,
 - 1077, 1082, 1088, 1093, 1098
- mapup-os-ibuffers-os-clock-inte
 - rrupt-handler, 891
- mapup-os-ibuffers-os-error-hand
 - ler-path1, 895
 - ler-path2, 901
- mapup-os-ibuffers-os-running-inp
 - ut-handler-path1, 1029
 - ut-handler-path2, 1035
 - ut-handler-path3, 1041
 - ut-handler-path4, 1046
 - ut-handler-path5, 1051
 - ut-handler-path6, 1055
- mapup-os-ibuffers-os-running-outp
 - ut-handler-path1, 1082
 - ut-handler-path2, 1088
 - ut-handler-path3, 1093
 - ut-handler-path4, 1098
- mapup-os-ibuffers-os-svc-receive
 - handler-path1, 930
 - handler-path2, 936
 - handler-path3, 944
 - handler-path4, 951
- mapup-os-ibuffers-os-svc-send-h
 - andler-path1, 905
 - andler-path2, 911
 - andler-path3, 918
 - andler-path4, 924
- mapup-os-ibuffers-os-svc-tyi-ha
 - ndler-path1, 957
 - ndler-path2, 963
 - ndler-path3, 969
- mapup-os-ibuffers-os-svc-tyo-ha
 - ndler-path1, 973
 - ndler-path2, 980
 - ndler-path3, 985
 - ndler-path4, 991
- mapup-os-ibuffers-os-waiting-inp
 - ut-handler-path1, 998
 - ut-handler-path2, 1005
 - ut-handler-path3, 1012
 - ut-handler-path4, 1016
 - ut-handler-path5, 1020
 - ut-handler-path6, 1023
- mapup-os-ibuffers-os-waiting-outp
 - ut-handler-path1, 1062
 - ut-handler-path2, 1069
 - ut-handler-path3, 1073
 - ut-handler-path4, 1077
- mapup-os-mbuffers, 372, 376, 377,
 - 416, 891, 895, 901, 905, 911,
 - 918, 925, 930, 936, 944, 952,

957, 963, 970, 974, 980, 986,
 992, 998, 1005, 1012, 1016,
 1020, 1024, 1029, 1035, 1042,
 1047, 1051, 1056, 1062, 1069,
 1073, 1077, 1083, 1089, 1094,
 1098
 mapup-os-mbuffers-os-clock-inte
 rrupt-handler, 891
 mapup-os-mbuffers-os-error-hand
 ler-path1, 895
 ler-path2, 901
 mapup-os-mbuffers-os-running-inp
 ut-handler-path1, 1029
 ut-handler-path2, 1035
 ut-handler-path3, 1041
 ut-handler-path4, 1047
 ut-handler-path5, 1051
 ut-handler-path6, 1056
 mapup-os-mbuffers-os-running-outp
 ut-handler-path1, 1083
 ut-handler-path2, 1089
 ut-handler-path3, 1094
 ut-handler-path4, 1098
 mapup-os-mbuffers-os-svc-receive
 -handler-path1, 930
 -handler-path2, 936
 -handler-path3, 944
 -handler-path3-support, 944
 -handler-path4, 952
 -handler-path4-support, 952
 mapup-os-mbuffers-os-svc-send-h
 andler-path1, 905
 andler-path2, 911
 andler-path3, 918
 andler-path3-support, 918
 andler-path4, 925
 andler-path4-support, 925
 mapup-os-mbuffers-os-svc-tyi-ha
 ndler-path1, 957
 ndler-path2, 963
 ndler-path3, 970
 mapup-os-mbuffers-os-svc-tyo-ha
 ndler-path1, 974
 ndler-path2, 980
 ndler-path3, 986
 ndler-path4, 992
 mapup-os-mbuffers-os-waiting-inp
 ut-handler-path1, 998
 ut-handler-path2, 1005
 ut-handler-path3, 1012
 ut-handler-path4, 1016
 ut-handler-path5, 1020
 ut-handler-path6, 1024
 mapup-os-mbuffers-os-waiting-outp
 ut-handler-path1, 1062
 ut-handler-path2, 1069
 ut-handler-path3, 1073
 ut-handler-path4, 1077
 mapup-os-obuffers, 372, 376, 891, 895,
 901, 905, 911, 918, 925, 930,
 936, 944, 952, 957, 963, 969,
 973, 975, 980, 986, 991, 998,
 1005, 1012, 1016, 1020, 1024,
 1029, 1035, 1041, 1046, 1051,
 1055, 1062, 1069, 1070, 1073,
 1077, 1078, 1082, 1088, 1090,
 1093, 1098, 1099
 mapup-os-obuffers-os-clock-inte
 rrupt-handler, 891
 mapup-os-obuffers-os-error-hand
 ler-path1, 895
 ler-path2, 901
 mapup-os-obuffers-os-running-inp
 ut-handler-path1, 1029
 ut-handler-path2, 1035
 ut-handler-path3, 1041
 ut-handler-path4, 1046
 ut-handler-path5, 1051
 ut-handler-path6, 1055
 mapup-os-obuffers-os-running-outp
 ut-handler-path1, 1082
 ut-handler-path2, 1088
 ut-handler-path3, 1093
 ut-handler-path4, 1098
 mapup-os-obuffers-os-svc-receive
 -handler-path1, 930
 -handler-path2, 936
 -handler-path3, 944

- handler-path4, 952
- mapup-os-obuffers-os-svc-send-h
 - andler-path1, 905
 - andler-path2, 911
 - andler-path3, 918
 - andler-path4, 925
- mapup-os-obuffers-os-svc-tyi-ha
 - ndler-path1, 957
 - ndler-path2, 963
 - ndler-path3, 969
- mapup-os-obuffers-os-svc-tyo-ha
 - ndler-path1, 973
 - ndler-path2, 980
 - ndler-path3, 986
 - ndler-path4, 991
- mapup-os-obuffers-os-waiting-inp
 - ut-handler-path1, 998
 - ut-handler-path2, 1005
 - ut-handler-path3, 1012
 - ut-handler-path4, 1016
 - ut-handler-path5, 1020
 - ut-handler-path6, 1024
- mapup-os-obuffers-os-waiting-outp
 - ut-handler-path1, 1062
 - ut-handler-path2, 1069
 - ut-handler-path3, 1073
 - ut-handler-path4, 1077
- mapup-os-status-table-os-clock-i
 - nterrupt-handler, 892
- mapup-os-status-table-os-error-
 - handler-path1, 895
 - handler-path2, 902
- mapup-os-status-table-os-runnin
 - g-input-handler-path1, 1030
 - g-input-handler-path2, 1036
 - g-input-handler-path3, 1042
 - g-output-handler-path1, 1083
 - g-output-handler-path2, 1089
- mapup-os-status-table-os-svc-re
 - ceive-handler-path1, 930
 - ceive-handler-path2, 937
 - ceive-handler-path3, 945
 - ceive-handler-path4, 953
- mapup-os-status-table-os-svc-se
 - nd-handler-path1, 905
 - nd-handler-path2, 912
 - nd-handler-path3, 919
 - nd-handler-path4, 926
- mapup-os-status-table-os-svc-ty
 - o-handler-path1, 974
 - o-handler-path2, 980
 - o-handler-path3, 986
 - o-handler-path4, 992
- mapup-os-status-table-os-svc-tyi
 - handler-path1, 957
 - handler-path2, 964
 - handler-path3, 970
- mapup-os-status-table-os-waitin
 - g-input-handler-path1, 999
 - g-input-handler-path2, 1006
 - g-input-handler-path3, 1013
 - g-output-handler-path1, 1062
 - g-output-handler-path2, 1069
- mapup-os-tasks, 371–373, 375, 391,
 - 891, 895, 901, 905, 911, 917,
 - 924, 930, 936, 943, 951, 957,
 - 963, 969, 973, 980, 985, 991,
 - 998, 1005, 1012, 1016, 1019,
 - 1023, 1029, 1035, 1041, 1046,
 - 1050, 1055, 1062, 1069, 1073,
 - 1076, 1082, 1088, 1093, 1097,
 - 1100, 1102
- mapup-os-tasks-os-clock-interrupt
 - handler, 891
- mapup-os-tasks-os-error-handler
 - path1, 895
 - path2, 901
- mapup-os-tasks-os-running-input
 - handler-path1, 1029
 - handler-path2, 1035
 - handler-path3, 1041
 - handler-path4, 1046
 - handler-path5, 1050
 - handler-path6, 1055
- mapup-os-tasks-os-running-output
 - handler-path1, 1082
 - handler-path2, 1088
 - handler-path3, 1093

- handler-path4, 1097
- mapup-os-tasks-os-svc-receive-h
 - andler-path1, 930
 - andler-path2, 936
 - andler-path3, 943
 - andler-path3-support, 943
 - andler-path4, 951
 - andler-path4-support, 951
- mapup-os-tasks-os-svc-send-hand
 - ler-path1, 905
 - ler-path2, 911
 - ler-path3, 917
 - ler-path3-support, 917
 - ler-path4, 924
 - ler-path4-support, 924
- mapup-os-tasks-os-svc-tyi-handle
 - r-path1, 957
 - r-path2, 963
 - r-path3, 969
 - r-path3-support, 969
- mapup-os-tasks-os-svc-tyo-handle
 - r-path1, 973
 - r-path2, 980
 - r-path3, 985
 - r-path3-support, 985
 - r-path4, 991
 - r-path4-support, 991
- mapup-os-tasks-os-waiting-input
 - handler-path1, 998
 - handler-path2, 1005
 - handler-path3, 1012
 - handler-path4, 1016
 - handler-path5, 1019
 - handler-path6, 1023
- mapup-os-tasks-os-waiting-output
 - handler-path1, 1062
 - handler-path2, 1069
 - handler-path3, 1073
 - handler-path4, 1076
- mapup-os-tasks-tm-fetch-execute, 391
- mapup-os-tm-post-interrupt, 1104
- mapup-queue, 79, 80, 82–84, 364, 369–372, 375, 377, 391, 415, 416, 425, 739, 743, 746, 750, 753, 757, 765, 769, 773, 781, 784, 792, 794, 806, 810, 815, 829, 834, 838, 854, 858, 868, 873, 892, 895, 901, 905, 912, 917–919, 924, 925, 930, 937, 943–945, 951, 952, 957, 963, 969, 970, 974, 975, 980, 985, 986, 991, 992, 999, 1000, 1006, 1012, 1030, 1036, 1042, 1062, 1069, 1083, 1089, 1099–1102
- mapup-queue-getnth, 372
- mapup-queue-getnth-for-ibuffers, 425
- mapup-queue-getnth-for-obuffers, 975
- mapup-queue-getnth2-os-mbuffers, 416
- mapup-queue-list, 371, 372, 375
- mapup-queue-list-putnth, 372
- mapup-queue-os-readyq-os-clock-i
 - nterrupt-handler, 892
- mapup-queue-os-readyq-os-error-
 - handler-path1, 895
 - handler-path2, 901
- mapup-queue-os-readyq-os-runnin
 - g-input-handler-path1, 1030
 - g-input-handler-path2, 1036
 - g-input-handler-path3, 1042
 - g-output-handler-path1, 1083
 - g-output-handler-path2, 1089
- mapup-queue-os-readyq-os-svc-re
 - ceive-handler-path1, 930
 - ceive-handler-path2, 937
 - ceive-handler-path3, 944
 - ceive-handler-path4, 952
- mapup-queue-os-readyq-os-svc-se
 - nd-handler-path1, 905
 - nd-handler-path2, 912
 - nd-handler-path3, 919
 - nd-handler-path4, 925
- mapup-queue-os-readyq-os-svc-ty
 - o-handler-path1, 974
 - o-handler-path2, 980
 - o-handler-path3, 986
 - o-handler-path4, 992
- mapup-queue-os-readyq-os-svc-tyi

- handler-path1, 957
- handler-path2, 963
- handler-path3, 970
- mapup-queue-os-readyq-os-waitin
 - g-input-handler-path1, 999
 - g-input-handler-path2, 1006
 - g-input-handler-path3, 1012
 - g-output-handler-path1, 1062
 - g-output-handler-path2, 1069
- mapup-regs, 370, 371, 373, 374, 890, 893, 899, 903, 909, 915, 922, 928, 934, 940, 949, 955, 961, 967, 972, 978, 983, 989, 996, 1003, 1010, 1014, 1018, 1022, 1027, 1033, 1039, 1044, 1049, 1053, 1060, 1067, 1072, 1075, 1080, 1086, 1092, 1096
- mapup-regs-os-clock-interrupt-handler, 890
- mapup-regs-os-error-handler-path1, 893
- h2, 899
- mapup-regs-os-running-input-handler-path1, 1026
- dlr-path2, 1032
- dlr-path3, 1038
- dlr-path4, 1044
- dlr-path5, 1049
- dlr-path6, 1053
- mapup-regs-os-running-output-handler-path1, 1080
- ndler-path2, 1086
- ndler-path3, 1091
- ndler-path4, 1096
- mapup-regs-os-svc-receive-handler-path1, 928
- r-path2, 934
- r-path3, 940
- r-path4, 949
- mapup-regs-os-svc-send-handler-path1, 903
- ath2, 909
- ath3, 915
- ath4, 922
- mapup-regs-os-svc-tyi-handler-path1, 955
- ath2, 961
- ath3, 967
- mapup-regs-os-svc-tyo-handler-path1, 972
- ath2, 978
- ath3, 983
- ath4, 989
- mapup-regs-os-waiting-input-handler-path1, 996
- dlr-path2, 1003
- dlr-path3, 1010
- dlr-path4, 1014
- dlr-path5, 1018
- dlr-path6, 1022
- mapup-regs-os-waiting-output-handler-path1, 1060
- ndler-path2, 1067
- ndler-path3, 1072
- ndler-path4, 1075
- mapup-svcflag, 371, 374, 890, 894, 900, 904, 910, 916, 923, 929, 935, 941, 949, 956, 962, 967, 972, 979, 984, 989, 997, 1004, 1011, 1015, 1019, 1022, 1028, 1034, 1040, 1045, 1050, 1054, 1061, 1068, 1072, 1076, 1081, 1087, 1092, 1097
- mapup-svcflag-os-clock-interrupt-handler, 890
- mapup-svcflag-os-error-handler-path1, 894
- ath2, 900
- mapup-svcflag-os-running-input-handler-path1, 1027
- handler-path2, 1033
- handler-path3, 1039
- handler-path4, 1045
- handler-path5, 1049
- handler-path6, 1054
- mapup-svcflag-os-running-output-handler-path1, 1081
- handler-path2, 1087

- handler-path3, 1092
- handler-path4, 1096
- mapup-svcflag-os-svc-receive-handler-path1, 929
- ndler-path2, 935
- ndler-path3, 941
- ndler-path4, 949
- mapup-svcflag-os-svc-send-handle
 - r-path1, 904
 - r-path2, 910
 - r-path3, 916
 - r-path4, 923
- mapup-svcflag-os-svc-tyi-handle
 - r-path1, 956
 - r-path2, 962
 - r-path3, 967
- mapup-svcflag-os-svc-tyo-handle
 - r-path1, 972
 - r-path2, 979
 - r-path3, 984
 - r-path4, 989
- mapup-svcflag-os-waiting-input-handler-path1, 997
- handler-path2, 1004
- handler-path3, 1011
- handler-path4, 1015
- handler-path5, 1019
- handler-path6, 1022
- mapup-svcflag-os-waiting-output
 - handler-path1, 1061
 - handler-path2, 1068
 - handler-path3, 1072
 - handler-path4, 1076
- mapup-svcid, 371, 374, 890, 894, 900, 904, 910, 914–916, 921–923, 929, 935, 939–941, 948, 950, 956, 962, 966, 967, 973, 979, 982–984, 988–990, 997, 1004, 1011, 1015, 1019, 1023, 1028, 1034, 1040, 1045, 1050, 1054, 1061, 1068, 1072, 1076, 1081, 1087, 1092, 1097
- mapup-svcid-os-clock-interrupt-handler, 890
- mapup-svcid-os-error-handler-path1, 894
- h2, 900
- mapup-svcid-os-running-input-handler-path1, 1028
- ndler-path2, 1034
- ndler-path3, 1040
- ndler-path4, 1045
- ndler-path5, 1050
- ndler-path6, 1054
- mapup-svcid-os-running-output-handler-path1, 1081
- andler-path2, 1087
- andler-path3, 1092
- andler-path4, 1097
- mapup-svcid-os-svc-receive-handler-path1, 929
- ler-path2, 935
- ler-path3, 941
- ler-path4, 950
- mapup-svcid-os-svc-send-handler
 - path1, 904
 - path2, 910
 - path3, 916
 - path4, 923
- mapup-svcid-os-svc-tyi-handler-path1, 956
- ath2, 962
- ath3, 967
- mapup-svcid-os-svc-tyo-handler-path1, 973
- ath2, 979
- ath3, 984
- ath4, 990
- mapup-svcid-os-waiting-input-handler-path1, 997
- ndler-path2, 1004
- ndler-path3, 1011
- ndler-path4, 1015
- ndler-path5, 1019
- ndler-path6, 1023
- mapup-svcid-os-waiting-output-handler-path1, 1061
- andler-path2, 1068

- andler-path3, 1072
- andler-path4, 1076
- mapup-task, 371, 375, 390, 391, 891, 894, 900, 904, 911, 916, 923, 929, 936, 942, 950, 956, 962, 968, 973, 979, 984, 990, 997, 1004, 1011, 1015, 1019, 1023, 1028, 1034, 1040, 1046, 1050, 1055, 1061, 1068, 1072, 1076, 1082, 1087, 1093, 1097, 1099, 1100, 1102, 1104
- mapup-task-os-clock-interrupt-handler-path1, 890
- mapup-task-os-error-handler-path1, 894
- mapup-task-os-error-handler-path2, 900
- mapup-task-os-running-input-handler-path1, 1028
- mapup-task-os-running-input-handler-path2, 1034
- mapup-task-os-running-input-handler-path3, 1040
- mapup-task-os-running-input-handler-path4, 1045
- mapup-task-os-running-input-handler-path5, 1050
- mapup-task-os-running-input-handler-path6, 1054
- mapup-task-os-running-output-handler-path1, 1081
- mapup-task-os-running-output-handler-path2, 1087
- mapup-task-os-running-output-handler-path3, 1093
- mapup-task-os-running-output-handler-path4, 1097
- mapup-task-os-svc-receive-handler-r-path1, 929
- mapup-task-os-svc-receive-handler-r-path2, 935
- mapup-task-os-svc-receive-handler-r-path3, 942
- mapup-task-os-svc-receive-handler-r-path4, 950
- mapup-task-os-svc-send-handler-path1, 904
- mapup-task-os-svc-send-handler-path2, 910
- mapup-task-os-svc-send-handler-path3, 916
- mapup-task-os-svc-send-handler-path4, 923
- mapup-task-os-svc-tyi-handler-path1, 956
- mapup-task-os-svc-tyi-handler-path2, 962
- mapup-task-os-svc-tyi-handler-path3, 968
- mapup-task-os-svc-tyo-handler-path1, 973
- ath2, 979
- ath3, 984
- ath4, 990
- mapup-task-os-waiting-input-handler-path1, 997
- mapup-task-os-waiting-input-handler-path2, 1004
- mapup-task-os-waiting-input-handler-path3, 1011
- mapup-task-os-waiting-input-handler-path4, 1015
- mapup-task-os-waiting-input-handler-path5, 1019
- mapup-task-os-waiting-input-handler-path6, 1023
- mapup-task-os-waiting-output-handler-path1, 1061
- mapup-task-os-waiting-output-handler-path2, 1068
- mapup-task-os-waiting-output-handler-path3, 1072
- mapup-task-os-waiting-output-handler-path4, 1076
- mapup-task-separation, 390
- mapup-task-tm-fetch-execute-case-split, 390
- mapup-task-tm-post-interrupt, 1104
- mapup-tasks, 371, 373, 375, 390, 391, 891, 894, 901, 904, 911, 916, 917, 923, 924, 929, 936, 942, 943, 950, 951, 956, 963, 968, 973, 979, 984, 985, 990, 998, 1005, 1011, 1015, 1019, 1023, 1029, 1035, 1041, 1046, 1050, 1055, 1061, 1068, 1073, 1076, 1082, 1088, 1093, 1097, 1100, 1104
- mapup-tasks-os-clock-interrupt-handler, 891
- mapup-tasks-os-error-handler-path1, 894
- mapup-tasks-os-error-handler-path2, 900
- mapup-tasks-os-running-input-handler-path1, 1028
- mapup-tasks-os-running-input-handler-path2, 1034
- mapup-tasks-os-running-input-handler-path3, 1040
- mapup-tasks-os-running-input-handler-path4, 1046
- mapup-tasks-os-running-input-handler-path5, 1050
- mapup-tasks-os-running-input-handler-path6, 1055
- mapup-tasks-os-running-output-handler-path1, 1082
- mapup-tasks-os-running-output-handler-path2, 1087

- andler-path3, 1093
- andler-path4, 1097
- mapup-tasks-os-svc-recv-handler-path1, 929
- ler-path2, 936
- ler-path3-case1, 942
- ler-path3-case2, 942
- ler-path4-case1, 950
- ler-path4-case2, 950
- mapup-tasks-os-svc-send-handler-path1, 904
- path2, 911
- path3-case1, 916
- path3-case2, 917
- path4-case1, 923
- path4-case2, 924
- mapup-tasks-os-svc-tyi-handler-path1, 956
- ath2, 962
- ath3-case1, 968
- ath3-case2, 968
- mapup-tasks-os-svc-tyo-handler-path1, 973
- ath2, 979
- ath3-case1, 984
- ath3-case2, 985
- ath4-case1, 990
- ath4-case2, 990
- mapup-tasks-os-waiting-input-handler-path1, 998
- ndler-path2, 1005
- ndler-path3, 1011
- ndler-path4, 1015
- ndler-path5, 1019
- ndler-path6, 1023
- mapup-tasks-os-waiting-output-handler-path1, 1061
- andler-path2, 1068
- andler-path3, 1073
- andler-path4, 1076
- mapup-tasks-tm-fetch-execute-case1, 390
- se2, 391
- mapup-tasks-tm-post-interrupt, 1104
- member-insert, 25
- member-permutation, 25
- member-permutation2, 26
- member-qfirst-queue, 66
- member-remove1, 25
- member-remove2, 25
- memory-reference, 395
- min, 20, 102, 103, 131, 195, 208, 213
- min-memlength-tm-memlength-in-good-address-space, 213
- min-tm-memlength-memlength-in-good-address-space, 208
- more-ibuffer-address-facts, 421
- mutually-disjoint, 194, 363, 386
- mutually-disjoint-os-segment-table, 363
- mutually-disjoint-segments-are-disjoint, 386
- disjoint-case1, 386
- nested-intervals, 45–47, 379
- non-1-finite-number-is-zero, 49
- non-number-memlength-gives-bad-address-space, 205
- non-zero-difference, 16
- non-zero-length-implies-listp, 29
- non-zero-listp, 49, 50, 52, 53
- non-zero-listp-append, 52
- non-zero-listp-fixlength, 53
- non-zero-listp-getseg, 52
- non-zero-listp-list, 52
- non-zero-listp-putnth, 52
- non-zero-listp-putseg, 52
- nonlast, 62–64, 83
- nonlast-append, 63
- nonlast-of-length-1, 83
- not-ak-waiting-to-inputp-mapup-os, 1017
- not-ak-waiting-to-outputp-mapup-os, 1074
- not-ak-waiting-to-recvp-mapup-os, 926
- not-ak-waiting-to-sendp-mapup-os, 953

- not-array-qempty-array-enq, 72
- not-array-qempty-os-clock-new-readyq, 738
- not-array-qfullp-array-deq, 72
- not-array-qfullp-array-nonlast, 77
- not-equal-number-of-ready-tasks-length-status, 300
- not-equal-plus-base-limit-0, 378
- not-lessp-plus-times-quotient, 22
- not-lessp-times-plus, 21
- not-lessp-times-plus-instance, 22
- not-listp-remove, 24
- not-member-index-ready-set, 298
- not-member-insert, 25
- not-member-permutation, 25
- not-member-remove1, 25
- not-member-remove2, 25
- not-permutation, 26
- not-qempty-enq, 65
- not-qempty-mapup-queue-os-read-yq, 377
- not-qfullp-deq, 65
- not-status-flag-ready-list, 299
- not-tm-in-supervisor-mode-getnth-mapup-os-tasks, 1102
- not-tm-in-supervisor-mode-mapup-task, 1102
- not-tm-in-supervisor-mode-os-clock-interrupt-handler, 737
- not-tm-in-supervisor-mode-os-error-handler-path1, 741
- not-tm-in-supervisor-mode-os-error-handler-path2, 744
- not-tm-in-supervisor-mode-os-running-input-handler-path1, 827
- not-tm-in-supervisor-mode-os-running-input-handler-path2, 831
- not-tm-in-supervisor-mode-os-running-input-handler-path3, 836
- not-tm-in-supervisor-mode-os-running-input-handler-path4, 841
- not-tm-in-supervisor-mode-os-running-input-handler-path5, 844
- not-tm-in-supervisor-mode-os-running-input-handler-path6, 848
- not-tm-in-supervisor-mode-os-running-output-handler-path1, 867
- not-tm-in-supervisor-mode-os-running-output-handler-path2, 871
- not-tm-in-supervisor-mode-os-running-output-handler-path3, 875
- not-tm-in-supervisor-mode-os-running-output-handler-path4, 878
- not-tm-in-supervisor-mode-os-svc-receive-handler-path1, 764
- not-tm-in-supervisor-mode-os-svc-receive-handler-path2, 767
- not-tm-in-supervisor-mode-os-svc-receive-handler-path3, 771
- not-tm-in-supervisor-mode-os-svc-receive-handler-path4, 776
- not-tm-in-supervisor-mode-os-svc-send-handler-path1, 748
- not-tm-in-supervisor-mode-os-svc-send-handler-path2, 751
- not-tm-in-supervisor-mode-os-svc-send-handler-path3, 755
- not-tm-in-supervisor-mode-os-svc-send-handler-path4, 760
- not-tm-in-supervisor-mode-os-svc-tyi-handler-path1, 780
- not-tm-in-supervisor-mode-os-svc-tyi-handler-path2, 783
- not-tm-in-supervisor-mode-os-svc-tyi-handler-path3, 786
- not-tm-in-supervisor-mode-os-svc-tyo-handler-path1, 790
- not-tm-in-supervisor-mode-os-svc-tyo-handler-path2, 793
- not-tm-in-supervisor-mode-os-svc-tyo-handler-path3, 797
- not-tm-in-supervisor-mode-os-svc-tyo-handler-path4, 800
- not-tm-in-supervisor-mode-os-waiting-input-handler-path1, 804
- not-tm-in-supervisor-mode-os-waiting-input-handler-path2, 808
- not-tm-in-supervisor-mode-os-waiting-input-handler-path3, 813
- not-tm-in-supervisor-mode-os-waiting-input-handler-path4, 817
- not-tm-in-supervisor-mode-os-waiting-input-handler-path5, 820
- not-tm-in-supervisor-mode-os-waiting-input-handler-path6, 823
- not-tm-in-supervisor-mode-os-waiting-output-handler-path1, 852
- not-tm-in-supervisor-mode-os-waiting-output-handler-path2, 856
- not-tm-in-supervisor-mode-os-waiting-output-handler-path3, 860
- not-tm-in-supervisor-mode-os-waiting-output-handler-path4, 863
- not-tm-waiting-os-clock-interrupt-handler, 740
- not-tm-waiting-os-error-handler-path2, 747
- not-tm-waiting-os-running-input-handler-path1, 829
- not-tm-waiting-os-running-input-handler-path2, 834
- not-tm-waiting-os-running-input-handler-path3, 838
- not-tm-waiting-os-running-input-handler-path4, 842
- not-tm-waiting-os-running-input-handler-path5, 846
- not-tm-waiting-os-running-input-handler-path6, 849
- not-tm-waiting-os-running-output-handler-path1, 869
- not-tm-waiting-os-running-output-handler-path2, 873
- not-tm-waiting-os-running-output-handler-path3, 876
- not-tm-waiting-os-running-output-handler-path4, 880
- not-tm-waiting-os-svc-receive-handler-path2, 769

- andler-path3, 773
- andler-path4, 777
- not-tm-waiting-os-svc-send-handler-path2, 754
- andler-path3, 757
- andler-path4, 761
- not-tm-waiting-os-svc-tyi-handle-r-path2, 785
- andler-path3, 788
- not-tm-waiting-os-svc-tyo-handle-r-path2, 795
- andler-path3, 798
- andler-path4, 801
- not-tm-waiting-os-waiting-input-handler-path1, 806
- andler-path2, 810
- andler-path3, 815
- not-tm-waiting-os-waiting-output-handler-path1, 854
- andler-path2, 858
- nthcdr, 29, 37, 53–61, 346, 347, 356, 376
- nthcdr-append-left, 55
- nthcdr-append-right, 55
- nthcdr-for-length, 57
- nthcdr-n-times, 346, 347
- nthcdr-nthcdr, 59
- nthcdr-putnth, 59
- nthcdr-putseg, 59
- nthcdr-with-large-n, 54
- nthcdr-with-small-recursion, 347
- nthcdr-zero, 54
- number-and-list-induction, 12
- number-and-litatom-listp, 318
- number-listp, 49, 50, 62, 66, 299
- number-listp-finite-number-listp, 62
- number-of-occurrences-of-set-member-is-1, 24
- number-of-ready-tasks, 300–302, 369
- number-of-ready-tasks-equals-length-ak-readyq, 302
- andler-index-ready-set, 301
- andler-os-readyq, 369
- number-of-unready-tasks, 300
- number-of-unready-tasks-non-zero, 300
- numberp-alu-value-tm-alu-decr, 86
- numberp-alu-value-tm-alu-decr-mod, 87
- numberp-alu-value-tm-alu-incr, 87
- numberp-alu-value-tm-alu-incr-mod, 88
- numberp-alu-value-tm-alu-mod, 88
- numberp-alu-value-tm-alu-mult, 89
- numberp-alu-value-tm-alu-plus, 90
- numberp-casesplit, 43
- numberp-getnth, 50
- numberp-getnth-from-tm-memory, 120
- numberp-getnth-from-tm-regs, 121
- numberp-mapup-base, 373
- numberp-mapup-limit, 373
- numberp-os-current-taskid, 386
- numberp-qcurrentlength, 68
- numberp-qfirst, 66
- numberp-qhead, 68
- numberp-qmaxlength, 69
- numberp-qtail, 68
- numberp-tm-fix-reg-address, 118
- numberp-tm-fix-word, 117
- obuffer-address-facts, 429
- occurrences, 23, 24
- open-putnth-casesplit, 36
- open-up-finite-number-listp, 129
- open-up-getnth, 34
- open-up-getnth-on-zero, 34
- open-up-getnth1, 34
- open-up-putnth-on-zero, 37
- open-up-putnth1, 36
- open-up-putseg-on-segment-cons, 44
- open-up-tm-processor, 116
- os-cih-time1, 446, 447
- os-cih-time2, 446, 447
- os-clock-interrupt-handler, 446, 447, 737–740, 883, 887, 889–892
- os-clock-interrupt-handler-addresses, 346
- os-clock-interrupt-handler-thru

- save-state, 440, 442, 447
- os-clock-interrupt-handler-to-di
 - spatcher, 442, 444, 447
- os-clock-interrupt-handler-to-re
 - store-state, 444, 446
- os-clock-new-pc, 362
- os-clock-new-readyq, 444, 738, 739, 889
- os-code, 348, 352, 360, 393, 438, 448, 461, 533, 553, 673, 737, 740, 744, 748, 751, 755, 759, 763, 766, 770, 775, 779, 782, 786, 790, 792, 796, 799, 803, 807, 812, 817, 820, 823, 826, 831, 835, 840, 843, 847, 851, 855, 860, 862, 866, 870, 875, 878
- os-code-address, 346
- os-code-length, 346
- os-code-os, 360
- os-code-os-clock-interrupt-hand
 - ler, 737
- os-code-os-error-handler-path1, 740
- os-code-os-error-handler-path2, 744
- os-code-os-running-input-handle
 - r-path1, 826
 - r-path2, 831
 - r-path3, 835
 - r-path4, 840
 - r-path5, 843
 - r-path6, 847
- os-code-os-running-output-handle
 - r-path1, 866
 - r-path2, 870
 - r-path3, 875
 - r-path4, 878
- os-code-os-svc-recv-handler-p
 - ath1, 763
 - ath2, 766
 - ath3, 770
 - ath4, 775
- os-code-os-svc-send-handler-pat
 - h1, 748
 - h2, 751
 - h3, 755
 - h4, 759
- os-code-os-svc-tyi-handler-path
 - 1, 779
 - 2, 782
 - 3, 786
- os-code-os-svc-tyo-handler-path
 - 1, 790
 - 2, 792
 - 3, 796
 - 4, 799
- os-code-os-waiting-input-handle
 - r-path1, 803
 - r-path2, 807
 - r-path3, 812
 - r-path4, 817
 - r-path5, 820
 - r-path6, 823
- os-code-os-waiting-output-handle
 - r-path1, 851
 - r-path2, 855
 - r-path3, 860
 - r-path4, 862
- os-current-ibuffer, 418, 422, 423, 539, 546, 549, 550, 552, 585, 588, 591, 600, 602, 603, 608, 611, 613, 621, 623–625, 633, 636, 639, 647, 648, 650, 655, 658, 661, 668, 670–672, 779, 782, 785, 787, 789, 803, 804, 807, 809, 811, 812, 814, 816, 818, 819, 821, 822, 824–826, 828, 830, 832, 835, 837, 839–841, 843, 845–847, 849, 850, 958, 964–969, 971, 998, 1000, 1005, 1007, 1012, 1013, 1016, 1017, 1020, 1021, 1023, 1025, 1029, 1031, 1035, 1037, 1041, 1043, 1046, 1047, 1051, 1052, 1055, 1056
- os-current-mbuffer, 407, 413–415, 435, 468, 475, 488, 490, 492, 494, 496, 497, 503, 510, 523, 524, 526, 529, 531, 532, 747, 750, 751, 754, 757–759, 762, 763,

766, 770, 772–775, 778, 906,
 913, 918, 920, 925, 927, 931,
 938–940, 942–944, 946–952,
 954
 os-current-mbuffer-rewrite-to-get
 nth2, 416
 os-current-obuffer, 427, 429–431, 560,
 566, 572, 573, 575–577, 680,
 682, 691, 694, 699, 701–704,
 711, 714, 723, 725, 730, 732,
 734–736, 789, 792, 795–797,
 799, 802, 851, 855, 857, 859,
 862, 864–866, 869, 870, 872,
 874, 877, 879, 881, 975, 981,
 986, 987, 991, 993, 1064,
 1069–1071, 1074, 1077, 1078,
 1084, 1088, 1090, 1094, 1098,
 1099
 os-current-taskid, 357, 363, 370, 386,
 388, 390, 391, 415, 423, 431,
 433–435, 468, 475, 477, 483,
 485, 488, 489, 492, 494, 496,
 497, 503, 510, 512, 517, 519,
 522, 524, 526, 527, 529, 531,
 532, 539, 546, 549, 550, 552,
 560, 566, 572, 573, 575–577,
 740, 746, 747, 750, 751, 754,
 755, 757–759, 762, 763, 766,
 769–771, 773–775, 778, 779,
 782, 785, 787–789, 792, 795–
 799, 801, 802, 806, 807, 811,
 816, 830, 834, 839, 842, 843,
 846, 850, 854, 859, 869, 874,
 877, 880, 889, 898, 899, 906,
 909, 913, 918, 920, 925–927,
 931, 933, 934, 938–940, 942–
 944, 946–954, 958, 960, 961,
 964–969, 971, 975, 977, 978,
 981, 986, 987, 991, 993, 996,
 1003, 1010, 1026, 1032, 1038,
 1044, 1048, 1053, 1060, 1067,
 1079, 1085, 1091, 1095
 os-current-taskid-address, 350
 os-current-taskid-tm-fetch-exec
 ute, 388
 os-destid, 273, 274, 280, 468, 475,
 477, 480, 482, 485, 488, 490,
 492, 494, 496, 497, 747, 749–
 751, 753–759, 762, 905, 906,
 912, 913, 917–920, 925–927
 os-dispatcher-address, 352
 os-dispatcher-pc-after-wait, 352
 os-dispatcher-qempty-return-ad
 dress, 352
 os-dispatcher-return-address, 346
 os-dispatcher-trace-label1, 352
 os-dispatcher-trace-label2, 352
 os-error-handler, 460, 747, 883, 902
 os-error-handler-address, 346
 os-error-handler-path1, 454, 458, 460,
 740–743, 893–896
 os-error-handler-path2, 457, 460, 744–
 747, 896–902
 os-error-handler-thru-save-state, 450,
 452, 458, 459
 os-error-handler-to-dispatcher, 452,
 454, 455, 459
 os-error-handler-to-restore-state, 455,
 457
 os-error-new-pc, 362
 os-error-time1, 457, 458
 os-error-time2, 458, 459
 os-error-time3, 459
 os-ibuffer-length, 331
 os-ibuffers, 348, 349, 364, 372, 387,
 418, 420–425, 433, 434, 533,
 738, 742, 745, 749, 752, 756,
 761, 765, 768, 772, 777, 780,
 783, 787, 791, 794, 797, 800,
 804, 805, 809, 813, 814, 818,
 821, 824, 828, 832, 837, 841,
 845, 848, 849, 852, 857, 861,
 864, 867, 871, 876, 879
 os-ibuffers-address, 346
 os-ibuffers-address-plus-qarray
 -field, 418
 os-ibuffers-address-plus-qcurrl
 ength-field, 418

- os-ibuffers-address-plus-qmaxlength-field, 418
- os-ibuffers-address-plus-qtail-field, 418
- os-ibuffers-length, 331
- os-ibuffers-os-clock-interrupt-handler, 738
- os-ibuffers-os-error-handler-path1, 742
- os-ibuffers-os-error-handler-path2, 745
- os-ibuffers-os-running-input-handler-path1, 828
- os-ibuffers-os-running-input-handler-path2, 832
- os-ibuffers-os-running-input-handler-path3, 837
- os-ibuffers-os-running-input-handler-path4, 841
- os-ibuffers-os-running-input-handler-path5, 845
- os-ibuffers-os-running-input-handler-path6, 848
- os-ibuffers-os-running-output-handler-path1, 867
- os-ibuffers-os-running-output-handler-path2, 871
- os-ibuffers-os-running-output-handler-path3, 876
- os-ibuffers-os-running-output-handler-path4, 879
- os-ibuffers-os-svc-receive-handler-path1, 765
- os-ibuffers-os-svc-receive-handler-path2, 768
- os-ibuffers-os-svc-receive-handler-path3, 772
- os-ibuffers-os-svc-receive-handler-path4, 777
- os-ibuffers-os-svc-send-handler-path1, 749
- os-ibuffers-os-svc-send-handler-path2, 752
- os-ibuffers-os-svc-send-handler-path3, 756
- os-ibuffers-os-svc-send-handler-path4, 761
- os-ibuffers-os-svc-tyi-handler-path1, 780
- os-ibuffers-os-svc-tyi-handler-path2, 783
- os-ibuffers-os-svc-tyi-handler-path3, 787
- os-ibuffers-os-svc-tyo-handler-path1, 791
- os-ibuffers-os-svc-tyo-handler-path2, 794
- os-ibuffers-os-svc-tyo-handler-path3, 797
- os-ibuffers-os-svc-tyo-handler-path4, 800
- os-ibuffers-os-waiting-input-handler-path1, 804
- os-ibuffers-os-waiting-input-handler-path2, 809
- os-ibuffers-os-waiting-output-handler-path3, 813
- os-ibuffers-os-waiting-output-handler-path4, 818
- os-ibuffers-os-waiting-output-handler-path5, 821
- os-ibuffers-os-waiting-output-handler-path6, 824
- os-ibuffers-os-waiting-output-handler-path1, 852
- os-ibuffers-os-waiting-output-handler-path2, 857
- os-ibuffers-os-waiting-output-handler-path3, 861
- os-ibuffers-os-waiting-output-handler-path4, 864
- os-ibuffers-tm-fetch-execute, 387
- os-ibuffers-with-dequeued-character, 423, 424, 787
- os-ibuffers-with-enqueued-character, 425, 813, 824, 837, 848
- os-ibuffers-with-enqueued-overflow-character, 424, 425, 809, 821, 832, 845
- os-ibuffers-with-qreplaced-overflow-character, 424, 804, 818, 828, 841
- os-idevid, 273, 274, 280
- os-implements-ak, 1104
- os-implements-parallel-tasks, 1105
- os-input-interrupt-handler, 882, 883, 1101
- os-input-interrupt-handler-address, 346
- os-input-new-pc, 362
- os-intended-clock-interrupt, 438–440, 447, 736
- os-intended-error-interrupt, 447, 448, 450, 458–460, 740
- os-intended-input-interrupt, 578, 580, 599–603, 621–625, 628, 646–650, 667–671, 673, 802, 883
- os-intended-output-interrupt, 673–675, 698–704, 706, 730–736, 850, 883
- os-intended-svc-interrupt, 881, 882
- os-intended-svc-receive-interrupt, 498, 522–525, 527–532, 763, 881
- os-intended-svc-send-interrupt, 460, 461, 463, 487–498, 747, 881

- os-intended-svc-tyi-interrupt, 533, 535, 548–553, 779, 881
- os-intended-svc-tyo-interrupt, 553, 555, 571–577, 789, 881
- os-limit, 348
- os-machine-code, 347
- os-mbuffer-dest-multiplier, 406
- os-mbuffer-length, 331
- os-mbuffer-source-multiplier, 406
- os-mbuffers, 348, 350, 364, 372, 387, 407, 410–416, 435, 461, 739, 742, 745, 749, 752, 756, 761, 765, 768, 772, 777, 781, 784, 788, 791, 794, 798, 801, 805, 809, 814, 818, 821, 825, 828, 833, 837, 842, 845, 849, 853, 857, 861, 864, 867, 872, 876, 880
- os-mbuffers-address, 346
- os-mbuffers-address-facts1, 408
- os-mbuffers-address-facts3, 409
- os-mbuffers-address-plus-qarray-field, 407
- os-mbuffers-address-plus-qcurrle-ngth-field, 407
- os-mbuffers-address-plus-qmaxle-ngth-field, 407
- os-mbuffers-address-plus-qtail-field, 406
- os-mbuffers-length, 331
- os-mbuffers-length-is-divisible-by-4, 331
- os-mbuffers-length-quarter, 331
- os-mbuffers-os-clock-interrupt-handler, 739
- os-mbuffers-os-error-handler-path1, 742
- os-mbuffers-os-error-handler-path2, 745
- os-mbuffers-os-running-input-handler-path1, 828
- os-mbuffers-os-running-input-handler-path2, 833
- os-mbuffers-os-running-input-handler-path3, 837
- os-mbuffers-os-running-input-handler-path4, 842
- os-mbuffers-os-running-input-handler-path5, 845
- os-mbuffers-os-running-input-handler-path6, 849
- os-mbuffers-os-running-output-handler-path1, 867
- os-mbuffers-os-running-output-handler-path2, 872
- os-mbuffers-os-running-output-handler-path3, 876
- os-mbuffers-os-running-output-handler-path4, 880
- os-mbuffers-os-svc-receive-handler-path1, 765
- os-mbuffers-os-svc-receive-handler-path2, 768
- os-mbuffers-os-svc-receive-handler-path3, 772
- os-mbuffers-os-svc-receive-handler-path4, 777
- os-mbuffers-os-svc-send-handler-path1, 749
- os-mbuffers-os-svc-send-handler-path2, 752
- os-mbuffers-os-svc-send-handler-path3, 756
- os-mbuffers-os-svc-send-handler-path4, 761
- os-mbuffers-os-svc-tyi-handler-path1, 781
- os-mbuffers-os-svc-tyi-handler-path2, 784
- os-mbuffers-os-svc-tyi-handler-path3, 788
- os-mbuffers-os-svc-tyo-handler-path1, 791
- os-mbuffers-os-svc-tyo-handler-path2, 794
- os-mbuffers-os-svc-tyo-handler-path3, 798
- os-mbuffers-os-svc-tyo-handler-path4, 801
- os-mbuffers-os-waiting-input-handler-path1, 805
- os-mbuffers-os-waiting-input-handler-path2, 809
- os-mbuffers-os-waiting-input-handler-path3, 814
- os-mbuffers-os-waiting-input-handler-path4, 818
- os-mbuffers-os-waiting-input-handler-path5, 821
- os-mbuffers-os-waiting-input-handler-path6, 825
- os-mbuffers-os-waiting-output-handler-path1, 853
- os-mbuffers-os-waiting-output-handler-path2, 857
- os-mbuffers-os-waiting-output-handler-path3, 861
- os-mbuffers-os-waiting-output-handler-path4, 864
- os-mbuffers-tm-fetch-execute, 387
- os-mbuffers-with-dequeued-message, 415, 772, 773, 777
- os-mbuffers-with-enqueued-message, 415, 756, 757, 761
- os-message, 273, 274, 276, 280, 906, 913, 920, 927, 975, 981, 986,

- 987, 991, 993
- os-new-base, 403, 739, 746, 753, 769, 784, 795
- os-new-cc, 403
- os-new-cpu, 403
- os-new-error, 403
- os-new-flags, 403
- os-new-limit, 403, 739, 746, 753, 769, 785, 795
- os-new-regs, 403
- os-new-svcflag, 403
- os-new-svcid, 403
- os-new-task-table, 403, 404, 738, 741, 745, 748, 752, 756, 760, 764, 767, 780, 783, 790, 793, 797, 800, 827, 832, 836, 841, 844, 848, 867, 871, 875, 878
- os-not-in-supervisor-mode, 362
- os-obuffer-length, 331
- os-obuffers, 348, 350, 364, 372, 387, 427, 429–431, 553, 673, 739, 742, 745, 749, 752, 756, 761, 765, 768, 772, 777, 781, 784, 788, 791, 794, 797, 801, 805, 809, 814, 818, 821, 824, 828, 833, 837, 842, 845, 849, 853, 857, 861, 864, 867, 871, 872, 876, 879, 975
- os-obuffers-address, 346
- os-obuffers-address-plus-qarray-field, 427
- os-obuffers-address-plus-qcurrlength-field, 427
- os-obuffers-address-plus-qmaxlength-field, 427
- os-obuffers-address-plus-qtail-field, 427
- os-obuffers-length, 331
- os-obuffers-os-clock-interrupt-handler, 739
- os-obuffers-os-error-handler-path1, 742
- os-obuffers-os-error-handler-path2, 745
- os-obuffers-os-running-input-handler-path1, 828
- os-obuffers-os-running-input-handler-path2, 833
- os-obuffers-os-running-input-handler-path3, 837
- os-obuffers-os-running-input-handler-path4, 842
- os-obuffers-os-running-input-handler-path5, 845
- os-obuffers-os-running-input-handler-path6, 849
- os-obuffers-os-running-output-handler-path1, 867
- os-obuffers-os-running-output-handler-path2, 871
- os-obuffers-os-running-output-handler-path3, 876
- os-obuffers-os-running-output-handler-path4, 879
- os-obuffers-os-svc-receive-handler-path1, 765
- os-obuffers-os-svc-receive-handler-path2, 768
- os-obuffers-os-svc-receive-handler-path3, 772
- os-obuffers-os-svc-receive-handler-path4, 777
- os-obuffers-os-svc-send-handler-path1, 749
- os-obuffers-os-svc-send-handler-path2, 752
- os-obuffers-os-svc-send-handler-path3, 756
- os-obuffers-os-svc-send-handler-path4, 761
- os-obuffers-os-svc-tyi-handler-path1, 781
- os-obuffers-os-svc-tyi-handler-path2, 784
- os-obuffers-os-svc-tyi-handler-path3, 788
- os-obuffers-os-svc-tyo-handler-path1, 791
- os-obuffers-os-svc-tyo-handler-path2, 794
- os-obuffers-os-svc-tyo-handler-path3, 797
- os-obuffers-os-svc-tyo-handler-path4, 801
- os-obuffers-os-waiting-input-handler-path1, 805
- os-obuffers-os-waiting-input-handler-path2, 809
- os-obuffers-os-waiting-input-handler-path3, 814
- os-obuffers-os-waiting-input-handler-path4, 818
- os-obuffers-os-waiting-input-handler-path5, 821
- os-obuffers-os-waiting-input-handler-path6, 824
- os-obuffers-os-waiting-output-handler-path1, 853
- os-obuffers-os-waiting-output-handler-path2, 857
- os-obuffers-os-waiting-output-handler-path3, 861
- os-obuffers-os-waiting-output-handler-path4, 864
- os-obuffers-tm-fetch-execute, 387
- os-obuffers-with-dequeued-character, 387

- cter, 431, 857, 864, 871, 879
- os-obuffers-with-enqueued-chara-cter, 431, 797, 798, 801
- os-odevid, 273, 274, 280
- os-oracle, 885, 887, 1105
- os-oracle-step, 885
- os-output-interrupt-handler, 883, 1101
- os-output-interrupt-handler-address, 346
- os-output-new-pc, 362
- os-pc-after-wait, 352
- os-processor, 884, 886, 887, 1104
- os-ready-set, 357, 364, 369, 377, 739, 743, 746, 750, 753, 757, 761, 765, 768, 769, 773, 777, 781, 784, 791, 792, 794, 805, 806, 810, 815, 829, 833, 834, 838, 853, 854, 858, 868, 872, 873
- os-ready-set-os-clock-interrupt-handler, 739
- os-ready-set-os-error-handler-path1, 743
- os-ready-set-os-error-handler-path2, 746
- os-ready-set-os-running-input-handler-path1, 829
- os-ready-set-os-running-input-handler-path2, 833
- os-ready-set-os-running-input-handler-path3, 838
- os-ready-set-os-running-output-handler-path1, 868
- os-ready-set-os-running-output-handler-path2, 872
- os-ready-set-os-svc-receive-handler-path1, 765
- os-ready-set-os-svc-receive-handler-path2, 768
- os-ready-set-os-svc-receive-handler-path3, 773
- os-ready-set-os-svc-receive-handler-path4, 777
- os-ready-set-os-svc-send-handler-path1, 750
- os-ready-set-os-svc-send-handler-path2, 753
- os-ready-set-os-svc-send-handler-path3, 757
- os-ready-set-os-svc-send-handler-path4, 761
- os-ready-set-os-svc-tyi-handler-path1, 781
- os-ready-set-os-svc-tyi-handler-path2, 784
- os-ready-set-os-svc-tyo-handler-path1, 791
- os-ready-set-os-svc-tyo-handler-path2, 794
- os-ready-set-os-waiting-input-handler-path1, 805
- os-ready-set-os-waiting-input-handler-path2, 810
- os-ready-set-os-waiting-input-handler-path3, 814
- os-ready-set-os-waiting-output-handler-path1, 853
- os-ready-set-os-waiting-output-handler-path2, 858
- os-readyq, 348, 349, 357, 363–365, 369, 370, 372, 377, 379, 380, 387, 391, 403, 407, 408, 411, 413, 415, 432–436, 438, 444, 448, 454, 455, 457, 458, 460, 461, 470, 472, 488, 490, 497, 505, 507, 523, 524, 532, 533, 541, 543, 549, 550, 552, 553, 562, 564, 568, 570, 572, 573, 575–577, 673, 738, 740, 742–747, 749–754, 756, 757, 761, 763–766, 768–770, 772, 773, 776, 779–785, 787, 789, 791, 792, 794–797, 799, 800, 802, 804, 806, 809, 810, 813, 815, 818, 821, 824, 826, 827, 829, 832, 834, 837, 838, 841, 845, 848, 851, 852, 854, 856, 858, 861, 863, 865, 867, 868, 871, 873, 876, 879, 889, 892, 895, 896, 898–902, 905, 906, 908–919, 921–925, 930, 931, 933–945, 947–952, 957, 958, 960–970, 974, 975, 977–993, 999, 1006, 1012, 1030, 1036, 1042, 1062, 1069, 1083, 1089, 1099–1102
- os-readyq-address, 346
- os-readyq-address-facts, 396
- os-readyq-length, 331
- os-readyq-os-clock-interrupt-handler, 738
- os-readyq-os-error-handler-path1, 742
- os-readyq-os-error-handler-path2, 745

- os-readyq-os-running-input-hand
 - ler-path1, 827
 - ler-path2, 832
 - ler-path3, 836
 - ler-path4, 841
 - ler-path5, 845
 - ler-path6, 848
- os-readyq-os-running-output-han
 - dl-path1, 867
 - dl-path2, 871
 - dl-path3, 876
 - dl-path4, 879
- os-readyq-os-svc-receive-handle
 - r-path1, 764
 - r-path2, 768
 - r-path3, 772
 - r-path4, 776
- os-readyq-os-svc-send-handler-p
 - ath1, 749
 - ath2, 752
 - ath3, 756
 - ath4, 761
- os-readyq-os-svc-tyi-handler-pat
 - h1, 780
 - h2, 783
 - h3, 787
- os-readyq-os-svc-tyo-handler-pat
 - h1, 791
 - h2, 794
 - h3, 797
 - h4, 800
- os-readyq-os-waiting-input-hand
 - ler-path1, 804
 - ler-path2, 809
 - ler-path3, 813
 - ler-path4, 818
 - ler-path5, 821
 - ler-path6, 824
- os-readyq-os-waiting-output-han
 - dl-path1, 852
 - dl-path2, 856
 - dl-path3, 861
 - dl-path4, 863
- os-readyq-qarray, 357
- os-readyq-qarray-address, 396
- os-readyq-tm-fetch-execute, 387
- os-restore-state-address, 352
- os-resume-task-address, 352
- os-resume-task-trace-label1, 352
- os-running-input-handler, 672, 673, 850, 882, 1057
- os-running-input-handler-path1, 641, 647, 672, 826–830, 1025–1031
- os-running-input-handler-path2, 643, 648, 672, 831–835, 1031–1037
- os-running-input-handler-path3, 645, 650, 672, 835–839, 1037–1043
- os-running-input-handler-path4, 663, 668, 672, 840–843, 1043–1047
- os-running-input-handler-path5, 665, 670, 672, 843–846, 1048–1052
- os-running-input-handler-path6, 667, 672, 847–850, 1052–1057
- os-running-input-handler-thru-s
 - ave-state, 628, 630, 646, 648, 649, 653, 667–669, 671
- os-running-input-handler-to-ibu
 - ffer-full-test, 630, 633, 636, 639, 646, 648, 650
 - ffer-full-test2, 653, 655, 658, 661, 668, 669, 671
- os-running-input-handler-to-res
 - ume-task1, 633, 641
 - ume-task2, 636, 643
 - ume-task3, 639, 645
 - ume-task4, 655, 663
 - ume-task5, 658, 665
 - ume-task6, 661, 666
- os-running-input-time1, 645, 646
- os-running-input-time10, 668, 669
- os-running-input-time11, 670, 671
- os-running-input-time12, 670, 671
- os-running-input-time2, 645, 646

- os-running-input-time3, 647, 648
- os-running-input-time4, 647, 648
- os-running-input-time5, 649
- os-running-input-time6, 649, 650
- os-running-input-time7, 667, 668
- os-running-input-time8, 667, 668
- os-running-input-time9, 668, 669
- os-running-output-handler, 736, 881, 883, 1099
- os-running-output-handler-path1, 716, 730, 736, 866–869, 1078–1084
- os-running-output-handler-path2, 718, 732, 736, 870–874, 1084–1090
- os-running-output-handler-path3, 727, 734, 736, 874–877, 1090–1094
- os-running-output-handler-path4, 729, 735, 736, 877–881, 1095–1099
- os-running-output-handler-thru-save-state, 706, 709, 720, 730, 731, 733, 735
- os-running-output-handler-to-offer-empty-test, 709, 711, 714, 730, 732
- os-running-output-handler-to-offer-empty-test2, 720, 723, 725, 733, 735
- os-running-output-handler-to-resume-task1, 711, 716
- os-running-output-handler-to-resume-task2, 714, 718
- os-running-output-handler-to-resume-task3, 723, 727
- os-running-output-handler-to-resume-task4, 725, 729
- os-running-output-time1, 729, 730
- os-running-output-time2, 729, 730
- os-running-output-time3, 731
- os-running-output-time4, 731, 732
- os-running-output-time5, 732, 733
- os-running-output-time6, 732, 733
- os-running-output-time7, 734, 735
- os-running-output-time8, 734, 735
- os-save-state-return-address, 346
- os-segment-lengths, 395
- os-segment-table, 348, 349, 363, 364, 371, 379, 380, 386–388, 403, 404, 438, 448, 461, 533, 553, 673, 738, 740, 741, 745, 747, 749, 752, 754, 756, 758, 760, 762, 764, 767, 769, 772, 774, 776, 778, 780, 783, 785, 787, 788, 790, 793, 795, 797, 798, 800–802, 804, 806–808, 811, 813, 815–817, 821, 824, 827, 830, 832, 834, 836, 839, 841–844, 846, 848, 850, 852, 854, 856, 858, 859, 861, 863, 867, 869, 871, 874, 875, 877, 879, 880
- os-segment-table-address, 346
- os-segment-table-address-facts, 404
- os-segment-table-length, 331
- os-segment-table-os-clock-interrupt-handler, 738
- os-segment-table-os-error-handler-path1, 741
- os-segment-table-os-error-handler-path2, 745
- os-segment-table-os-running-input-handler-path1, 827
- os-segment-table-os-running-input-handler-path2, 832
- os-segment-table-os-running-input-handler-path3, 836
- os-segment-table-os-running-input-handler-path4, 841
- os-segment-table-os-running-input-handler-path5, 844
- os-segment-table-os-running-input-handler-path6, 848
- os-segment-table-os-running-output-handler-path1, 867
- os-segment-table-os-running-output-handler-path2, 871
- os-segment-table-os-running-output-handler-path3, 875
- os-segment-table-os-running-output-handler-path4, 879
- os-segment-table-os-svc-receive-handler-path1, 764
- os-segment-table-os-svc-receive-handler-path2, 767
- os-segment-table-os-svc-receive-handler-path3, 772
- os-segment-table-os-svc-receive-handler-path4, 776
- os-segment-table-os-svc-send-handler-path1, 749
- os-segment-table-os-svc-send-handler-path2, 752
- os-segment-table-os-svc-send-handler-path3, 756

- ndler-path4, 760
- os-segment-table-os-svc-tyi-handler-path1, 780
- dler-path2, 783
- dler-path3, 787
- os-segment-table-os-svc-tyo-handler-path1, 790
- dler-path2, 793
- dler-path3, 797
- dler-path4, 800
- os-segment-table-os-waiting-inp-ut-handler-path1, 804
- ut-handler-path2, 808
- ut-handler-path3, 813
- ut-handler-path4, 817
- ut-handler-path5, 821
- ut-handler-path6, 824
- os-segment-table-os-waiting-outp-ut-handler-path1, 852
- ut-handler-path2, 856
- ut-handler-path3, 861
- ut-handler-path4, 863
- os-segment-table-tm-fetch-execute, 388
- os-source, 331
- os-srcid, 273, 280, 415, 435, 503, 510, 512, 514, 517, 519, 522, 524, 526, 529, 531, 532, 763, 765, 766, 768, 770–775, 778, 930, 931, 937–940, 942–954
- os-status-table, 348, 349, 357, 363, 369, 370, 372, 388, 405, 406, 448, 461, 477, 480, 482, 485, 492, 494, 496, 497, 512, 514, 517, 519, 526, 529, 531–533, 553, 582, 600, 602, 603, 605, 621, 623–625, 630, 641, 643, 645, 647, 648, 650, 652, 668, 670–673, 677, 689, 699, 701–704, 709, 716, 718, 720, 730, 732, 734–736, 739, 742, 743, 745, 746, 749, 750, 753–755, 757–759, 761, 762, 765, 768, 770, 773–775, 777, 778, 781, 784, 788, 791, 794, 798, 801, 803, 805–807, 810–812, 814–816, 819, 822, 825, 826, 828–830, 833–835, 837–840, 842, 843, 845–847, 849–851, 853–855, 857–859, 861, 862, 864–866, 868–870, 872–874, 876, 877, 880, 881, 892, 895, 902, 905, 912, 917, 919, 920, 926, 927, 930, 937, 943–946, 953, 954, 957, 964, 970, 974, 980, 981, 986, 992, 999, 1000, 1006, 1007, 1013, 1017, 1021, 1024, 1026–1043, 1047, 1052, 1056, 1062–1064, 1070, 1074, 1078–1090, 1094, 1099
- os-status-table-address, 346
- os-status-table-address-facts, 405
- os-status-table-length, 331
- os-status-table-os-clock-interr-upt-handler, 739
- os-status-table-os-error-handle-r-path1, 742
- r-path2, 745
- os-status-table-os-running-input-handler-path1, 828
- handler-path2, 833
- handler-path3, 837
- handler-path4, 842
- handler-path5, 845
- handler-path6, 849
- os-status-table-os-running-outp-ut-handler-path1, 868
- ut-handler-path2, 872
- ut-handler-path3, 876
- ut-handler-path4, 880
- os-status-table-os-svc-receive-handler-path1, 765
- handler-path2, 768
- handler-path3, 773
- handler-path4, 777
- os-status-table-os-svc-send-handler-path1, 749
- dler-path2, 753
- dler-path3, 757

- dler-path4, 761
- os-status-table-os-svc-tyi-hand
 - ler-path1, 781
 - ler-path2, 784
 - ler-path3, 788
- os-status-table-os-svc-tyo-hand
 - ler-path1, 791
 - ler-path2, 794
 - ler-path3, 798
 - ler-path4, 801
- os-status-table-os-waiting-input
 - handler-path1, 805
 - handler-path2, 809
 - handler-path3, 814
 - handler-path4, 818
 - handler-path5, 822
 - handler-path6, 825
- os-status-table-os-waiting-outp
 - ut-handler-path1, 853
 - ut-handler-path2, 857
 - ut-handler-path3, 861
 - ut-handler-path4, 864
- os-status-table-tm-fetch-execute, 388
- os-step, 883, 884, 886, 1103
- os-step-implements-ak-step, 1103
- os-store-message, 273, 274, 276, 280,
 - 942, 943, 946, 950, 951, 968, 969
- os-svc-handler, 881–883, 1101
- os-svc-handler-address, 346
- os-svc-new-pc, 362
- os-svc-receive-handler, 532, 779, 881,
 - 954
- os-svc-receive-handler-address, 350
- os-svc-receive-handler-path1, 505, 523,
 - 532, 763–766, 928–931
- os-svc-receive-handler-path2, 507, 524,
 - 532, 766–770, 931–938
- os-svc-receive-handler-path3, 519, 527,
 - 532, 770–775, 938–946
- os-svc-receive-handler-path4, 521, 529,
 - 531, 532, 775–778, 947–954
- os-svc-receive-handler-thru-delive
 - ry, 510, 512, 514, 517, 526, 528, 530
- os-svc-receive-handler-thru-save
 - state, 498, 501, 522, 524, 526, 528, 530
- os-svc-receive-handler-to-dispat
 - cher, 503, 505, 507
- os-svc-receive-handler-to-qempt
 - yp-test, 501, 503, 510, 522, 524, 526, 528, 530
- os-svc-receive-handler-to-svc-re
 - sume-task1, 512, 519
 - sume-task2, 514, 521
 - sume-task3, 517, 521
- os-svc-receive-id, 273
- os-svc-receive-time1, 521, 522
- os-svc-receive-time10, 527, 528
- os-svc-receive-time11, 529, 530
- os-svc-receive-time12, 529, 530
- os-svc-receive-time13, 529, 530
- os-svc-receive-time2, 521, 522
- os-svc-receive-time3, 523, 524
- os-svc-receive-time4, 523, 524
- os-svc-receive-time5, 524–526
- os-svc-receive-time6, 525, 526
- os-svc-receive-time7, 525, 526
- os-svc-receive-time8, 527, 528
- os-svc-receive-time9, 527, 528
- os-svc-resume-task-address, 352
- os-svc-resume-task-trace-label1, 352
- os-svc-send-handler, 497, 498, 763,
 - 881, 927
- os-svc-send-handler-path1, 470, 488,
 - 497, 747–750, 903–906
- os-svc-send-handler-path2, 472, 490,
 - 497, 751–754, 907, 909–913
- os-svc-send-handler-path3, 485, 492,
 - 498, 755–759, 913–920
- os-svc-send-handler-path4, 486, 487,
 - 494, 496–498, 759–762, 920–927
- os-svc-send-handler-thru-enqueue, 475,
 - 477, 480, 483, 491, 493, 494, 496

os-svc-send-handler-thru-save-st
 ate, 463, 465, 487, 489, 491, 493,
 495, 498
 os-svc-send-handler-to-dispatche
 r, 468, 470, 472
 os-svc-send-handler-to-qfullp-te
 st, 465, 468, 475, 488, 489, 491,
 493, 495
 os-svc-send-handler-to-svc-resu
 me-task1, 477, 485
 me-task2, 480, 486
 me-task3, 483, 487
 os-svc-send-id, 273
 os-svc-send-time1, 487
 os-svc-send-time10, 492, 493
 os-svc-send-time11, 494, 495
 os-svc-send-time12, 494, 495
 os-svc-send-time13, 494–496
 os-svc-send-time2, 487, 488
 os-svc-send-time3, 488, 489
 os-svc-send-time4, 488, 489
 os-svc-send-time5, 490, 491
 os-svc-send-time6, 490, 491
 os-svc-send-time7, 490, 491
 os-svc-send-time8, 492, 493
 os-svc-send-time9, 492, 493
 os-svc-tyi-handler, 552, 553, 789, 881,
 971
 os-svc-tyi-handler-address, 351
 os-svc-tyi-handler-path1, 541, 549,
 552, 779–782, 954–958
 os-svc-tyi-handler-path2, 544, 551,
 552, 782–785, 958–964
 os-svc-tyi-handler-path3, 548, 552,
 785–789, 965–971
 os-svc-tyi-handler-thru-deliver
 y, 546, 548
 os-svc-tyi-handler-thru-save-st
 ate, 535, 537, 548, 550, 552
 os-svc-tyi-handler-to-dispatche
 r, 539, 541, 543
 os-svc-tyi-handler-to-qemptyt-te
 st, 537, 539, 546, 549, 550, 552
 os-svc-tyi-id, 273
 os-svc-tyi-time1, 548
 os-svc-tyi-time2, 548, 549
 os-svc-tyi-time3, 549, 550
 os-svc-tyi-time4, 549, 550
 os-svc-tyi-time5, 551, 552
 os-svc-tyi-time6, 551, 552
 os-svc-tyo-handler, 577, 802, 881, 993
 os-svc-tyo-handler-address, 351
 os-svc-tyo-handler-path1, 562, 572,
 577, 789–792, 971–975
 os-svc-tyo-handler-path2, 564, 573,
 577, 792–795, 975–981
 os-svc-tyo-handler-path3, 568, 575,
 577, 796–799, 981–987
 os-svc-tyo-handler-path4, 570, 576,
 577, 799–802, 987–993
 os-svc-tyo-handler-thru-enqueue, 566,
 568, 570
 os-svc-tyo-handler-thru-save-st
 ate, 555, 557, 571, 573, 574, 576
 os-svc-tyo-handler-to-dispatche
 r, 560, 562, 564
 os-svc-tyo-handler-to-qfullp-te
 st, 557, 560, 566, 571, 573, 574,
 576
 os-svc-tyo-id, 273
 os-svc-tyo-time1, 570, 571
 os-svc-tyo-time2, 571
 os-svc-tyo-time3, 572, 573
 os-svc-tyo-time4, 572, 573
 os-svc-tyo-time5, 573, 574
 os-svc-tyo-time6, 574
 os-svc-tyo-time7, 575, 576
 os-svc-tyo-time8, 575, 576
 os-syntab, 342
 os-task-table, 348, 363, 370, 388, 403,
 432, 433, 435, 438, 448, 460,
 533, 553, 673, 738, 741, 745,
 748, 752, 756, 760, 764, 767,
 771, 776, 780, 783, 786, 790,
 793, 797, 800, 804, 808, 813,
 817, 820, 823, 827, 832, 836,
 841, 844, 848, 852, 856, 861,
 863, 867, 871, 875, 878, 889,

898, 899, 909, 934, 960, 961,
 977, 978, 996, 1003, 1010,
 1026, 1032, 1038, 1044, 1048,
 1053, 1060, 1067, 1079, 1085,
 1091, 1095
 os-task-table-address, 346
 os-task-table-length, 331
 os-task-table-os-clock-interrupt
 -handler, 738
 os-task-table-os-error-handler-p
 ath1, 741
 ath2, 745
 os-task-table-os-running-input-
 handler-path1, 827
 handler-path2, 832
 handler-path3, 836
 handler-path4, 841
 handler-path5, 844
 handler-path6, 848
 os-task-table-os-running-output
 -handler-path1, 867
 -handler-path2, 871
 -handler-path3, 875
 -handler-path4, 878
 os-task-table-os-svc-recv-handle
 ndler-path1, 764
 ndler-path2, 767
 ndler-path3, 771
 ndler-path4, 776
 os-task-table-os-svc-send-handle
 r-path1, 748
 r-path2, 752
 r-path3, 756
 r-path4, 760
 os-task-table-os-svc-tyi-handle
 r-path1, 780
 r-path2, 783
 r-path3, 786
 os-task-table-os-svc-tyo-handle
 r-path1, 790
 r-path2, 793
 r-path3, 797
 r-path4, 800
 os-task-table-os-waiting-input-
 handler-path1, 804
 handler-path2, 808
 handler-path3, 813
 handler-path4, 817
 handler-path5, 820
 handler-path6, 823
 os-task-table-os-waiting-output
 -handler-path1, 852
 -handler-path2, 856
 -handler-path3, 860
 -handler-path4, 863
 os-task-table-tm-fetch-execute, 388
 os-task-table-with-delivered-ch
 aracter, 433, 434, 786, 787
 os-task-table-with-delivered-me
 ssage, 435, 436, 771, 772, 776
 os-taskid, 365
 os-temp-r2-address, 352
 os-temp-r3-address, 352
 os-time-for-clock-interrupt-han
 dler, 446, 447, 884, 885
 dler-thru-save-state, 439, 440,
 447
 dler-to-dispatcher, 441, 442, 447
 dler-to-finish, 444, 446, 447
 dler-to-restore-state, 442, 444,
 447
 os-time-for-error-handler, 460, 884,
 885
 os-time-for-error-handler-path1, 458,
 460
 os-time-for-error-handler-path2, 459,
 460
 os-time-for-error-handler-thru-
 save-state, 448, 450, 458, 459
 os-time-for-error-handler-to-di
 spatcher, 450, 452, 458, 459
 os-time-for-error-handler-to-fi
 nish, 456, 457, 459
 os-time-for-error-handler-to-re
 store-state, 454, 455, 459
 os-time-for-error-handler-to-wait
 -state, 452, 454, 458
 os-time-for-input-interrupt-han

- dler, 882–885
 - os-time-for-output-interrupt-handler, 883–885
 - os-time-for-running-input-handler, 672, 673, 882
 - r-path1, 646, 647, 672
 - r-path2, 647, 648, 672
 - r-path3, 649, 650, 672
 - r-path4, 667, 668, 672
 - r-path5, 669, 670, 672
 - r-path6, 670–672
 - r-thru-save-state, 626, 628, 646, 648, 649, 667, 669, 671
 - r-to-ibuffer-full-test, 628, 630, 646, 648, 650
 - r-to-ibuffer-full-test2, 650, 653, 668, 669, 671
 - r-to-lpsw, 639, 641, 643, 645, 646, 648, 650, 663, 665, 667–669, 671
 - r-to-resume-task1, 630, 633, 646
 - r-to-resume-task2, 633, 636, 648
 - r-to-resume-task3, 636, 639, 650
 - r-to-resume-task4, 653, 655, 668
 - r-to-resume-task5, 655, 658, 669
 - r-to-resume-task6, 658, 661, 671
 - os-time-for-running-output-handler, 735, 736, 883
 - ler-path1, 729, 730, 736
 - ler-path2, 731, 732, 736
 - ler-path3, 733, 734, 736
 - ler-path4, 734–736
 - ler-thru-save-state, 704, 706, 730, 731, 733, 735
 - ler-to-lpsw, 714, 716, 718, 727, 729, 730, 732, 733, 735
 - ler-to-obuffer-empty-test, 706, 707, 709, 730, 732
 - ler-to-obuffer-empty-test2, 718, 720, 733, 735
 - ler-to-resume-task1, 709, 711, 730
 - ler-to-resume-task2, 711, 714, 732
 - ler-to-resume-task3, 720, 721, 723, 733
 - ler-to-resume-task4, 723, 725, 735
 - os-time-for-svc-handler, 881, 882, 884, 885
 - os-time-for-svc-receive-handler, 532, 881
 - path1, 522, 523, 532
 - path2, 523, 524, 532
 - path3, 525, 527, 532
 - path4, 531, 532
 - path4a, 527–529, 531
 - path4b, 529–531
 - thru-delivery, 507, 510, 526, 528, 530
 - thru-save-state, 498, 522, 523, 525, 528, 530
 - to-dispatcher, 501, 503, 522, 524
 - to-lpsw, 505, 507, 524
 - to-qempty-test, 498, 501, 522, 524, 526, 528, 530
 - to-svc-resume-task1, 510, 512, 526
 - to-svc-resume-task2, 512, 514, 528
 - to-svc-resume-task3, 514, 517, 530
 - to-svcr, 517, 519, 521, 526, 528, 530
 - to-wait-state, 503, 505, 522
 - os-time-for-svc-send-handler, 497, 498, 881
 - os-time-for-svc-send-handler-path
 - h1, 487, 488, 497
 - h2, 489, 490, 497
 - h3, 490–492, 497
 - h4, 496, 497
 - h4a, 493, 494, 496
 - h4b, 495, 496
 - os-time-for-svc-send-handler-thru-enqueue, 472, 475, 491, 493, 495
 - ru-save-state, 461, 463, 487, 489, 491, 493, 495, 498
 - os-time-for-svc-send-handler-to-dispatcher, 465, 468, 488, 489

- lpsw, 470, 472, 489
- qfullp-test, 463, 465, 488, 489, 491, 493, 495
- svc-resume-task1, 475, 477, 491
- svc-resume-task2, 477, 480, 494
- svc-resume-task3, 480, 483, 496
- svcr, 483, 485–487, 491, 494, 496
- wait-state, 468, 470, 488
- os-time-for-svc-tyi-handler, 552, 553, 881
- os-time-for-svc-tyi-handler-pat
 - h1, 548, 549, 552
 - h2, 549–552
 - h3, 551, 552
- os-time-for-svc-tyi-handler-thr
 - u-delivery, 544, 546, 552
 - u-save-state, 533, 535, 548, 550, 551
- os-time-for-svc-tyi-handler-to-dispatcher, 537, 539, 549, 550
 - lpsw, 541, 544, 550
 - svcr, 546, 548, 552
 - wait-state, 539–541, 549
- os-time-for-svc-tyi-handler-to-qe
 - mptyp-test, 535, 537, 548, 550, 552
- os-time-for-svc-tyo-handler, 577, 881
- os-time-for-svc-tyo-handler-pat
 - h1, 571, 572, 577
 - h2, 572, 573, 577
 - h3, 574, 575, 577
 - h4, 575–577
- os-time-for-svc-tyo-handler-thr
 - u-enqueue, 564, 566, 575, 576
 - u-save-state, 554, 555, 571, 573, 574, 576
- os-time-for-svc-tyo-handler-to-dispatcher, 558, 560, 571, 573
 - lpsw, 562, 564, 573
 - svcr1, 566, 568, 575
 - svcr2, 568, 570, 576
 - wait-state, 560, 562, 571
- os-time-for-svc-tyo-handler-to-q
 - fullp-test, 556, 557, 571, 573, 574, 576
- os-time-for-waiting-input-handle
 - r, 624, 625, 882
 - r-path1, 599, 600, 625
 - r-path2, 600–602, 625
 - r-path3, 602, 603, 625
 - r-path4, 620, 621, 625
 - r-path5, 622, 623, 625
 - r-path6, 623–625
 - r-thru-readyq-empty-test, 578, 580, 599, 601, 603, 621–623
 - r-to-dispatcher1, 583, 585, 600
 - r-to-dispatcher2, 585, 588, 601
 - r-to-dispatcher3, 588, 591, 603
 - r-to-dispatcher4, 606, 608, 621
 - r-to-dispatcher5, 608, 611, 622
 - r-to-dispatcher6, 611, 613, 624
 - r-to-ibuffer-full-test, 580, 582, 599, 601, 603
 - r-to-ibuffer-full-test2, 604, 606, 621, 622, 624
 - r-to-lpsw, 591, 593, 596, 598, 600, 601, 603
 - r-to-wait, 613, 614, 616, 618, 620–622, 624
- os-time-for-waiting-output-hand
 - ler, 704, 883
 - ler-path1, 698, 699, 704
 - ler-path2, 700, 701, 704
 - ler-path3, 701, 702, 704
 - ler-path4, 702–704
 - ler-thru-readyq-empty-test, 674, 675, 698, 700, 701, 703
 - ler-to-dispatcher1, 678, 680, 699
 - ler-to-dispatcher2, 680, 682, 700
 - ler-to-dispatcher3, 690, 691, 702
 - ler-to-dispatcher4, 692, 694, 703
 - ler-to-lpsw, 682, 685, 687, 699, 700
 - ler-to-obuffer-empty-test, 675, 678, 698, 700
 - ler-to-obuffer-empty-test2, 688, 689, 701, 703

- ler-to-wait, 694, 696, 697, 702, 703
- os-time-slice, 362
- os-time-slice-address, 346
- os-trace-label1, 350
- os-trace-label10, 350
- os-trace-label11, 350
- os-trace-label12, 351
- os-trace-label13, 351
- os-trace-label14, 351
- os-trace-label15, 351
- os-trace-label16, 351
- os-trace-label17, 351
- os-trace-label18, 351
- os-trace-label19, 351
- os-trace-label2, 350
- os-trace-label20, 351
- os-trace-label21, 351
- os-trace-label22, 351
- os-trace-label23, 351
- os-trace-label24, 351
- os-trace-label25, 351
- os-trace-label26, 351
- os-trace-label27, 351
- os-trace-label28, 351
- os-trace-label29, 351
- os-trace-label3, 350
- os-trace-label30, 351
- os-trace-label31, 351
- os-trace-label32, 351
- os-trace-label33, 351
- os-trace-label34, 351
- os-trace-label35, 352
- os-trace-label36, 352
- os-trace-label4, 350
- os-trace-label5, 350
- os-trace-label6, 350
- os-trace-label7, 350
- os-trace-label8, 350
- os-trace-label9, 350
- os-update-status, 406, 742, 743, 745, 746, 749, 753, 765, 768, 773, 781, 784, 791, 794, 805, 810, 814, 828, 833, 838, 853, 857, 868, 872
- os-waiting-input-handler, 625, 825, 882, 1025
- os-waiting-input-handler-path1, 593, 600, 625, 803–807, 993, 994, 996–1000
- os-waiting-input-handler-path2, 596, 602, 625, 807–811, 1000, 1001, 1003–1007
- os-waiting-input-handler-path3, 598, 603, 625, 812–816, 1007, 1010–1014
- os-waiting-input-handler-path4, 616, 621, 625, 816–819, 1014–1017
- os-waiting-input-handler-path5, 618, 623, 625, 819–822, 1018–1021
- os-waiting-input-handler-path6, 620, 624, 625, 822–825, 1021–1025
- os-waiting-input-handler-thru-re
 - adyq-empty-test, 580, 582, 599, 601, 603, 606, 621, 622, 624
- os-waiting-input-handler-to-disp
 - atcher1, 585, 593
 - atcher2, 588, 596
 - atcher3, 591, 598
 - atcher4, 608, 616
 - atcher5, 611, 618
 - atcher6, 613, 620
- os-waiting-input-handler-to-ibu
 - ffer-full-test, 583, 585, 588, 591, 600, 601, 603
 - ffer-full-test2, 606, 608, 611, 613, 621, 622, 624
- os-waiting-input-time1, 598, 599
- os-waiting-input-time10, 622
- os-waiting-input-time11, 623, 624
- os-waiting-input-time12, 623, 624
- os-waiting-input-time2, 599, 600
- os-waiting-input-time3, 600, 601
- os-waiting-input-time4, 600, 601

- os-waiting-input-time5, 602, 603
- os-waiting-input-time6, 602, 603
- os-waiting-input-time7, 620, 621
- os-waiting-input-time8, 620, 621
- os-waiting-input-time9, 622
- os-waiting-output-handler, 704, 865, 883, 1078
- os-waiting-output-handler-path1, 685, 699, 704, 851–855, 1057, 1060–1064
- os-waiting-output-handler-path2, 687, 701, 704, 855–859, 1064, 1067–1071
- os-waiting-output-handler-path3, 696, 702, 704, 859–862, 1071–1074
- os-waiting-output-handler-path4, 698, 703, 704, 862–865, 1074–1078
- os-waiting-output-handler-thru-readyq-empty-test, 675, 678, 689, 698, 700, 701, 703
- os-waiting-output-handler-to-di-spatcher1, 680, 685
- os-waiting-output-handler-to-di-spatcher2, 682, 687
- os-waiting-output-handler-to-di-spatcher3, 691, 696
- os-waiting-output-handler-to-di-spatcher4, 694, 697
- os-waiting-output-handler-to-offer-empty-test, 678, 680, 682, 699, 700
- os-waiting-output-handler-to-offer-empty-test2, 689, 691, 694, 702, 703
- os-waiting-output-time1, 698
- os-waiting-output-time2, 698, 699
- os-waiting-output-time3, 699, 700
- os-waiting-output-time4, 699, 700
- os-waiting-output-time5, 701
- os-waiting-output-time6, 701, 702
- os-waiting-output-time7, 702, 703
- os-waiting-output-time8, 702, 703
- os-waiting-to-inputp-implies-readyq-not-full, 370
- os-waiting-to-outputp-implies-readyq-not-full, 370
- os-waiting-to-receivep-implies-readyq-not-full, 369
- os-waiting-to-sendp-implies-readyq-not-full, 369
- permutation, 23–28, 51, 66, 298–300, 302, 364, 369, 370, 739, 743, 746, 750, 753, 757, 765, 769, 773, 781, 784, 792, 794, 806, 810, 815, 829, 834, 838, 854, 858, 868, 873
- permutation-ak-readyq-ak-ready-set, 298
- permutation-append-car, 26
- permutation-append-insert, 27
- permutation-deq-index-ready-set-putnth, 299
- permutation-enq-deq, 66
- permutation-enq-index-ready-set-putnth, 302
- permutation-os-clock-new-readyq-os-ready-set, 739
- permutation-os-readyq-os-ready-set-for-os-error-handler-path1, 743
- permutation-os-readyq-os-ready-set-for-os-error-handler-path2, 746
- permutation-os-readyq-os-ready-set-for-os-running-input-handler-path1, 829
- permutation-os-readyq-os-ready-set-for-os-running-input-handler-path2, 833
- permutation-os-readyq-os-ready-set-for-os-running-input-handler-path3, 838
- permutation-os-readyq-os-ready-set-for-os-running-output-handler-path1, 868
- permutation-os-readyq-os-ready-set-for-os-running-output-handler-path2, 873
- permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path1, 765
- permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path2, 769
- permutation-os-readyq-os-ready-set-for-os-svc-receive-handler-path3, 773
- permutation-os-readyq-os-ready-set-for-os-svc-send-handler-path1, 750

- set-for-os-svc-send-handler-path2, 753
- set-for-os-svc-send-handler-path3, 757
- set-for-os-svc-tyi-handler-path1, 781
- set-for-os-svc-tyi-handler-path2, 784
- set-for-os-svc-tyo-handler-path1, 792
- set-for-os-svc-tyo-handler-path2, 794
- set-for-os-waiting-input-handler-path1, 806
- set-for-os-waiting-input-handler-path2, 810
- set-for-os-waiting-input-handler-path3, 815
- set-for-os-waiting-output-handler-path1, 853
- set-for-os-waiting-output-handler-path2, 858
- permutation-os-readyq-qarray-os-ready-set, 364
- permutation-os-readyq-qarray-os-ready-set-corollary, 370
- permutation-reflexivity, 26
- permutation-remove-remove, 27
- permutation-right-cons, 26
- permutation-right-cons1, 26
- plist-of-length-0, 31
- plist-of-non-zero-length-is-a-list, 29
- plistp, 23, 24, 28, 29, 31–34, 47–49, 53–57, 59–64, 68–71, 73, 77, 79, 80, 120, 121, 123, 126, 128–130, 133, 135, 150, 151, 159, 163, 206, 209, 276–279, 290, 291, 294–297, 299, 302, 304, 305, 328, 348–350, 352, 353, 355, 356, 365, 367–369, 373, 385, 397, 398, 415, 416, 424, 425, 431, 434, 435, 1104, 1105
- plistp-address-space-memory, 206
- plistp-address-space-regs, 206
- plistp-ak-ibuffers, 294
- plistp-ak-ports, 297
- plistp-ak-mbuffers, 295
- plistp-ak-obuffers, 295
- plistp-ak-oports, 297
- plistp-ak-readyq, 296
- plistp-ak-status, 296
- plistp-append, 31
- plistp-array-deq, 71
- plistp-array-enq, 70
- plistp-array-nonlast, 73
- plistp-array-qarray, 69
- plistp-array-qreplace, 77
- plistp-array-queue, 68
- plistp-delta-segment, 79
- plistp-deq, 64
- plistp-enq, 64
- plistp-finite-number-queue, 353
- plistp-finite-number-queue-corollary, 365
- plistp-for-ibuffers, 416
- plistp-for-mbuffers, 367
- plistp-for-obuffers, 425
- plistp-firstn, 53
- plistp-fixlength, 32
- plistp-flatten, 57
- plistp-getnth-from-table, 56
- plistp-getnth-table, 56
- plistp-getseg, 32
- plistp-good-address-space-list, 290
- plistp-good-status-list, 291
- plistp-good-task-buffer-list, 277
- plistp-good-task-buffer-table, 279
- plistp-list1, 31
- plistp-list2, 31
- plistp-mapup-queue, 80
- plistp-mapup-regs, 373
- plistp-nonlast, 63
- plistp-nthcdr, 53
- plistp-os-code, 352
- plistp-os-ibuffers, 349
- plistp-os-ibuffers-with-dequeue-d-character, 424

- plistp-os-ibuffers-with-enqueue
 - d-character, 425
 - d-overflow-character, 425
- plistp-os-ibuffers-with-qreplace
 - d-overflow-character, 424
- plistp-os-mbuffers, 350
- plistp-os-mbuffers-with-dequeue
 - d-message, 415
- plistp-os-mbuffers-with-enqueue
 - d-message, 415
- plistp-os-obuffers, 350
- plistp-os-obuffers-with-dequeue
 - d-character, 431
- plistp-os-obuffers-with-enqueue
 - d-character, 431
- plistp-os-readyq, 349
- plistp-os-segment-table, 349
- plistp-os-status-table, 349
- plistp-os-task-table, 348
- plistp-os-task-table-with-delive
 - red-character, 434
 - red-message, 435
- plistp-putnth, 32
- plistp-putnth2, 47
- plistp-putseg, 32
- plistp-qreplace, 64
- plistp-remove, 24
- plistp-reverse, 48
- plistp-set, 24
- plistp-table, 55
- plistp-tm-cpu, 129
- plistp-tm-ports, 123
- plistp-tm-memory, 120
- plistp-tm-ports, 123
- plistp-tm-regs, 121
- plus-0, 13
- plus-0-arg1, 13
- plus-1, 13
- plus-add1, 2
- plus-arg2-must-be-zero, 45
- plus-cancellation, 3
- plus-commutativity-associativit
 - y-crock, 14
- plus-difference-cancellation, 13
- plus-equal-0, 2
- plus-not-equal, 22
- plus-plus-difference, 14
- plus-right-id2, 2
- plus-times-reduction, 14
- plus-times-sub1-reduction, 14
- plus-tm-base-tm-limit-tm-decreme
 - nt-clock, 160
- plus-tm-base-tm-limit-tm-set-cc, 225
- plus-tm-base-tm-limit-tm-set-pc, 160
- plus-tm-base-tm-limit-tm-set-sp, 225
- project, 312, 314, 317, 1105
- project-ith-task, 1105
- putnth, 28, 30, 32, 35–42, 44, 47, 51,
 - 52, 59, 60, 70, 71, 73, 77,
 - 80, 97, 100–102, 124, 129,
 - 272, 278–280, 283–288, 290,
 - 291, 299–302, 354–356, 369,
 - 370, 372, 379, 380, 391–393,
 - 398, 399, 404, 405, 413, 414,
 - 422, 423, 430, 433–435, 438,
 - 448, 461, 533, 553, 673, 750,
 - 753, 757, 765, 768, 773, 781,
 - 784, 791, 794, 805, 810, 815,
 - 829, 833, 838, 853, 858, 868,
 - 873, 895, 902, 905, 912, 914,
 - 915, 917, 919, 921, 922, 924,
 - 930, 937, 939, 940, 943, 945,
 - 947–949, 951, 957, 964–969,
 - 974, 981–983, 985, 988–991,
 - 999, 1006, 1013, 1030, 1036,
 - 1042, 1063, 1070, 1083, 1089
- putnth-after-putseg, 40
- putnth-after-putseg-proof, 40
- putnth-append-casesplit, 37
- putnth-append-left, 38
- putnth-append-right, 38
- putnth-coincidence, 38
- putnth-cons1, 37
- putnth-cons2, 37
- putnth-getnth-coincidence, 38
- putnth-getnth-coincidence1, 38
- putnth-has-no-effect, 37
- putnth-into-nil, 37

- putnth-non-interference, 38
- putnth-non-interference-corolla
 - ry1, 39
 - ry2, 39
- putnth-nthcdr, 59
- putnth-putnth-casesplit, 38
- putnth-putseg-coincidence, 39
- putnth-putseg-coincidence-inverse, 39
- putnth-putseg-coincidence-proof, 39
- putnth-shift, 37
- putnth-with-non-number-index, 37
- putnth2, 47, 48, 84, 278, 356, 369
- putseg, 28, 30–32, 36, 39, 40, 42–44, 51, 52, 59, 60, 399, 403–406, 413–415, 422–425, 430–433, 435, 438, 448, 461, 533, 553, 673, 757
- putseg-cons1, 44
- putseg-cons2, 43
- putseg-getseg-coincidence, 44
- putseg-has-no-effect, 43
- putseg-nil-segment, 44
- putseg-non-list-segment, 44
- putseg-nthcdr, 59
- putseg-nthcdr-proof, 59
- putseg-with-non-number-index, 39
- qarray-field, 66
- qcurlength-field, 66
- qempty, 64–66, 82, 84, 275, 282, 286, 288, 289, 298–300, 305, 377
- qempty2, 84, 274, 284, 304
- qfirst, 64–66, 82, 84, 275, 279, 286, 288, 289, 299, 300, 377, 391, 892, 917, 918, 924, 925, 943, 944, 951, 952, 969, 985, 991, 1070, 1078, 1090, 1099, 1100, 1102
- qfirst-enq, 65
- qfirst2, 84, 274, 284
- qfullp, 64, 65, 82, 84, 275, 285, 287, 302, 305, 415, 975, 1000
- qfullp2, 84, 274, 283, 303
- qhead-field, 66
- qmaxlength-field, 66
- qmaxlength-greater-than-zero, 69
- qmaxlength-non-zero, 69
- qreplace, 64–66, 84, 272
- qreplace-address-fact, 420
- qreplace-ith-buffer, 272, 287, 998, 1016, 1029, 1046
- qtail-field, 66
- queue-facts, 298
- quotient-difference, 18
- quotient-divides, 9
- quotient-lessp, 8
- quotient-plus, 18
- quotient-plus-plus-plus-times, 389
- quotient-plus-plus-times, 389
- quotient-plus-times, 18
- quotient-plus-times-remainder, 22
- quotient-times, 9
- quotient-times-other-way, 18
- quotient-times1, 8
- quotient-x-x, 17
- re-associate-times, 406
- re-associate-times1, 407
- real-addr, 101–104, 106, 111, 112, 160, 208, 209, 214, 268, 269, 271
- real-addr-num, 101–103, 132, 156, 218, 379, 380
- real-addr-source, 101–103, 132, 156, 218, 379, 380
- registers-untouched-by-tm-execute, 193
 - add, 163
 - binary, 191
 - call, 167
 - decr, 169
 - decr-mod, 170
 - incr, 171
 - incr-mod, 171
 - instruction, 192
 - mod, 176
 - multiply, 178

- nullary, 190
- return, 179
- unary, 190
- registers-untouched-by-tm-fetch
 - execute, 193
- regroup-fact, 16
- remainder-4-equals-3, 882
- remainder-add1, 7
- remainder-add1-casesplit, 17
- remainder-array-qfirst-finite-n
 - umber-queue, 354
 - umber-queue-corollary, 366
 - umber-queue-corollary-for-ibuffers, 417
 - umber-queue-corollary-for-mbuffers, 368
 - umber-queue-corollary-for-obuffers, 427
- remainder-difference-non-zero, 19
- remainder-difference-times, 11
- remainder-divides, 21
- remainder-non-zero, 22
- remainder-noop, 17
- remainder-of-1, 11
- remainder-plus, 20
- remainder-plus-plus-plus-times, 389
- remainder-plus-plus-times, 389
- remainder-plus-times-1, 11
- remainder-plus-times-2, 11
- remainder-plus-times-remainder, 22
- remainder-quotient, 5
- remainder-quotient-elim, 5
- remainder-remainder, 21
- remainder-remainder-inverse, 23
- remainder-times, 9
- remainder-times-1, 11
- remainder-times-other-way, 17
- remainder-wrt-1, 5
- remainder-wrt-12, 5
- remainder-x-x, 5
- remainder-zero-implies-leq-fact, 20
- remove, 23–28, 51, 66, 299
- remove-remove, 25
- restore-cc, 432
 - restore-cc-for-receive-svc-return, 435
 - restore-cc-for-tyi-svc-return, 434
 - restore-error, 432
 - restore-error-for-receive-svc-return, 436
 - restore-error-for-tyi-svc-return, 434
 - restore-regs, 432
 - restore-regs-for-receive-svc-return, 435
 - restore-regs-for-tyi-svc-return, 434
 - restore-regs1, 432
 - restore-svcflag, 433
 - restore-svcid, 433
 - restore-svcid-for-receive-svc-return, 436
 - restore-svcid-for-tyi-svc-return, 434
- reverse, 48, 49, 886, 1104
- reverse-alist, 329
- reverse-append, 48
- reverse-list1, 48
- reverse-list2, 48
- reverse-reverse, 49
- rewrite-car-to-getnth-0, 30
- rewrite-clock-trace1, 447
- rewrite-clock-trace2, 447
- rewrite-clock-trace3, 447
- rewrite-error-trace1, 458
- rewrite-error-trace2, 458
- rewrite-error-trace3, 459
- rewrite-error-trace4, 459
- rewrite-error-trace5, 459
- rewrite-getnth-from-table, 58
- rewrite-getnth-from-table0, 59
- rewrite-insert-to-cons-crock, 301
- rewrite-insert-to-cons-crock1, 301
- rewrite-insert-to-cons-crock2, 301
- rewrite-non-zero-difference-as-lessp, 16
- rewrite-os-mbuffer-address, 407
- rewrite-os-mbuffer-address1, 408
- rewrite-running-input-trace1, 646

rewrite-running-input-trace10, 667
rewrite-running-input-trace11, 667
rewrite-running-input-trace12, 668
rewrite-running-input-trace13, 669
rewrite-running-input-trace14, 669
rewrite-running-input-trace15, 669
rewrite-running-input-trace16, 671
rewrite-running-input-trace17, 671
rewrite-running-input-trace18, 671
rewrite-running-input-trace2, 646
rewrite-running-input-trace3, 646
rewrite-running-input-trace4, 648
rewrite-running-input-trace5, 648
rewrite-running-input-trace6, 648
rewrite-running-input-trace7, 649
rewrite-running-input-trace8, 649
rewrite-running-input-trace9, 650
rewrite-running-output-trace1, 730
rewrite-running-output-trace10, 734
rewrite-running-output-trace11, 735
rewrite-running-output-trace12, 735
rewrite-running-output-trace2, 730
rewrite-running-output-trace3, 730
rewrite-running-output-trace4, 731
rewrite-running-output-trace5, 731
rewrite-running-output-trace6, 732
rewrite-running-output-trace7, 733
rewrite-running-output-trace8, 733
rewrite-running-output-trace9, 733
rewrite-svc-receive-trace1, 522
rewrite-svc-receive-trace10, 526
rewrite-svc-receive-trace11, 528
rewrite-svc-receive-trace12, 528
rewrite-svc-receive-trace13, 528
rewrite-svc-receive-trace14, 528
rewrite-svc-receive-trace15, 530
rewrite-svc-receive-trace16, 530
rewrite-svc-receive-trace17, 530
rewrite-svc-receive-trace18, 530
rewrite-svc-receive-trace2, 522
rewrite-svc-receive-trace3, 522
rewrite-svc-receive-trace4, 523
rewrite-svc-receive-trace5, 524
rewrite-svc-receive-trace6, 524
rewrite-svc-receive-trace7, 525
rewrite-svc-receive-trace8, 526
rewrite-svc-receive-trace9, 526
rewrite-svc-send-trace1, 487
rewrite-svc-send-trace10, 491
rewrite-svc-send-trace11, 493
rewrite-svc-send-trace12, 493
rewrite-svc-send-trace13, 493
rewrite-svc-send-trace14, 493
rewrite-svc-send-trace15, 495
rewrite-svc-send-trace16, 495
rewrite-svc-send-trace17, 495
rewrite-svc-send-trace18, 496
rewrite-svc-send-trace2, 487
rewrite-svc-send-trace3, 488
rewrite-svc-send-trace4, 489
rewrite-svc-send-trace5, 489
rewrite-svc-send-trace6, 489
rewrite-svc-send-trace7, 491
rewrite-svc-send-trace8, 491
rewrite-svc-send-trace9, 491
rewrite-svc-tyi-trace1, 548
rewrite-svc-tyi-trace2, 548
rewrite-svc-tyi-trace3, 549
rewrite-svc-tyi-trace4, 550
rewrite-svc-tyi-trace5, 550
rewrite-svc-tyi-trace6, 550
rewrite-svc-tyi-trace7, 551
rewrite-svc-tyi-trace8, 552
rewrite-svc-tyi-trace9, 552
rewrite-svc-tyo-trace1, 571
rewrite-svc-tyo-trace10, 576
rewrite-svc-tyo-trace11, 576
rewrite-svc-tyo-trace12, 576
rewrite-svc-tyo-trace2, 571
rewrite-svc-tyo-trace3, 571
rewrite-svc-tyo-trace4, 573
rewrite-svc-tyo-trace5, 573
rewrite-svc-tyo-trace6, 573
rewrite-svc-tyo-trace7, 574
rewrite-svc-tyo-trace8, 574
rewrite-svc-tyo-trace9, 574
rewrite-waiting-input-trace1, 599
rewrite-waiting-input-trace10, 621

rewrite-waiting-input-trace11, 621
 rewrite-waiting-input-trace12, 621
 rewrite-waiting-input-trace13, 622
 rewrite-waiting-input-trace14, 622
 rewrite-waiting-input-trace15, 622
 rewrite-waiting-input-trace16, 623
 rewrite-waiting-input-trace17, 624
 rewrite-waiting-input-trace18, 624
 rewrite-waiting-input-trace2, 599
 rewrite-waiting-input-trace3, 600
 rewrite-waiting-input-trace4, 601
 rewrite-waiting-input-trace5, 601
 rewrite-waiting-input-trace6, 601
 rewrite-waiting-input-trace7, 602
 rewrite-waiting-input-trace8, 603
 rewrite-waiting-input-trace9, 603
 rewrite-waiting-output-trace1, 698
 rewrite-waiting-output-trace10, 703
 rewrite-waiting-output-trace11, 703
 rewrite-waiting-output-trace12, 703
 rewrite-waiting-output-trace2, 698
 rewrite-waiting-output-trace3, 699
 rewrite-waiting-output-trace4, 700
 rewrite-waiting-output-trace5, 700
 rewrite-waiting-output-trace6, 700
 rewrite-waiting-output-trace7, 701
 rewrite-waiting-output-trace8, 701
 rewrite-waiting-output-trace9, 702
 rewrite-zero-difference-as-equality, 16

 set-status-induction, 299
 setp, 23–25, 27, 28, 66, 298
 setp-index-ready-set, 298
 setp-insert, 28
 setp-permutation, 28
 setp-permutation-base-case, 28
 setp-permutation-induction, 27
 setp-permutation-induction-step, 28
 setp-remove, 27
 size-of-tm-interrupting-input-port, 125
 size-of-tm-interrupting-output-port, 125

 sorted, 24–27, 302
 sorted-index-ready-set, 302
 sorted-insert, 25
 sorted-remove, 26
 status-flag-readyp, 292, 299–304, 306
 status-flag-readyp-getnth, 299
 status-flag-readyp-iff-member-index-ready-set, 299
 index-ready-set1, 299
 status-flag-readyp-list, 301
 strange-induction, 45
 sub1-difference, 15
 sum-zero-implies-addends-zero, 19
 symmetry-of-disjoint, 194

 table, 55–60, 328, 357, 363, 364, 369–372, 377, 379, 380, 386, 387, 403, 404, 407, 410–414, 416, 418, 420–425, 427, 429, 430, 432–435, 740, 743, 746, 747, 750, 753, 754, 757, 758, 762, 765, 768, 769, 772–774, 778, 781, 784, 785, 787, 788, 791, 794, 795, 798, 801, 802, 805–807, 809–811, 814–816, 818, 821, 824, 828–830, 832–834, 837–839, 841–843, 845, 846, 849, 850, 853, 854, 857–859, 864, 868, 869, 872–874, 877, 879, 880, 889, 892, 895, 898, 899, 902, 905, 909, 912, 919, 926, 930, 934, 937, 945, 953, 957, 960, 961, 964, 970, 974, 975, 977, 978, 980, 981, 986, 992, 996, 999, 1003, 1006, 1010, 1013, 1026, 1030, 1032, 1036, 1038, 1042, 1044, 1048, 1053, 1060, 1062, 1063, 1067, 1070, 1079, 1083, 1085, 1089, 1091, 1095
 table-append, 57
 table-flatten, 57
 table-for-length, 57
 table-putnth, 60

table-putseg, 60
 table-putseg-inverse, 60
 table-putseg-subcase, 59
 tablep, 55–57, 61
 tablep-table, 56
 task, 272, 274, 275, 312, 315, 316
 task-activep, 275, 314, 317
 task-channels, 272, 275
 task-communication-step, 275, 316
 task-communicationp, 273, 275
 task-destid, 274, 275, 315
 task-execute-input, 275, 316
 task-execute-output, 274, 275, 316
 task-execute-receive, 274, 275, 315
 task-execute-send, 274, 275, 315
 task-fetch-execute, 274, 275
 task-ibuffer-capacity, 272
 task-ibuffers, 272, 274, 275, 277
 task-idevid, 274
 task-mbuffer-capacity, 273
 task-mbuffers, 272–275, 277
 task-message, 274, 275, 315, 316
 task-obuffer-capacity, 272
 task-obuffers, 272, 274, 275, 277
 task-odevid, 274
 task-private-step, 275
 task-processor, 275, 276, 317, 1105
 task-pstate, 273–275, 277
 task-receive-instructionp, 273, 275
 task-send-instructionp, 273, 275
 task-shell, 272
 task-shellp, 277
 task-srcid, 273, 275, 315
 task-step, 275, 276, 317
 task-store-message, 274, 275
 task-tyi-instructionp, 273
 task-tyo-instructionp, 273, 275
 task-update-channels, 275, 276, 314
 task-update-control, 273–275
 tick, 393
 tick-plus, 393
 timed-tm-implements-btimed-tm, 886
 timed-tm-processor, 884–886
 timed-tm-step, 884–886
 timed-tm-step-implements-os-step, 886
 times-1, 10
 times-add1, 4
 times-commutativity-associativit
 y-crock, 389
 times-difference, 6
 times-equal-1, 9
 times-id-iff-1, 8
 times-identity, 9
 times-identity1, 9
 times-zero2, 3
 times-zero3, 16
 tm, 91–97, 217
 tm-add-opcode, 110
 tm-address-error, 99
 tm-alu-arithmetic, 90
 tm-alu-decr, 85, 86, 104
 tm-alu-decr-mod, 85–87, 104
 tm-alu-decr-mod-value-identical
 -to-ideal-result, 87
 tm-alu-decr-value-identical-to-i
 deal-result, 86
 tm-alu-difference, 85, 104, 116, 117
 tm-alu-incr, 85, 87, 104
 tm-alu-incr-mod, 85, 87, 88, 105
 tm-alu-incr-mod-value-identical
 -to-ideal-result, 88
 tm-alu-incr-value-identical-to-i
 deal-result, 87
 tm-alu-mod, 85, 88, 89, 105
 tm-alu-mod-value-identical-to-i
 deal-result, 89
 tm-alu-mult, 85, 89, 106
 tm-alu-mult-value-identical-to-i
 deal-result, 89
 tm-alu-plus, 86, 89, 90, 103
 tm-alu-plus-value-identical-to-i
 deal-result, 90
 tm-arg-datum, 101
 tm-arg-index, 101
 tm-arg-mode, 101
 tm-arg1, 108, 111, 264, 265
 tm-arg2, 109, 111, 264

- tm-base, 92–97, 102, 103, 121, 131, 132, 160, 163, 167, 169–171, 176, 178, 179, 190–193, 195–200, 207, 208, 218–272, 363, 379, 380, 739, 740, 746, 753, 754, 758, 762, 769, 773, 774, 777, 778, 784, 785, 788, 795, 798, 801, 806, 811, 815, 816, 829, 830, 834, 839, 842, 846, 849, 850, 854, 858, 859, 869, 873, 874, 876, 877, 880
- tm-base-address-space, 207
- tm-base-mapup-address-space, 224
- tm-base-os, 363
- tm-base-os-clock-interrupt-handler, 739
- tm-base-os-error-handler-path2, 746
- tm-base-os-running-input-handle
 - r-path1, 829
 - r-path2, 834
 - r-path3, 839
 - r-path4, 842
 - r-path5, 846
 - r-path6, 849
- tm-base-os-running-output-handle
 - r-path1, 869
 - r-path2, 873
 - r-path3, 876
 - r-path4, 880
- tm-base-os-svc-receive-handler-path2, 769
 - ath3, 773
 - ath4, 777
- tm-base-os-svc-send-handler-path2, 753
 - h3, 758
 - h4, 762
- tm-base-os-svc-tyi-handler-path2, 784
 - 3, 788
- tm-base-os-svc-tyo-handler-path2, 795
 - 3, 798
 - 4, 801
- tm-base-os-waiting-input-handle
 - r-path1, 806
 - r-path2, 811
 - r-path3, 815
- tm-base-os-waiting-output-handle
 - r-path1, 854
 - r-path2, 858
- tm-base-tm-limit-os-clock-interrupt-handler, 740
- tm-base-tm-limit-os-error-handle
 - r-path2, 746
- tm-base-tm-limit-os-running-input-handler-path1, 830
 - ut-handler-path2, 834
 - ut-handler-path3, 839
 - ut-handler-path4, 842
 - ut-handler-path5, 846
 - ut-handler-path6, 850
- tm-base-tm-limit-os-running-output-handler-path1, 869
 - ut-handler-path2, 873
 - ut-handler-path3, 877
 - ut-handler-path4, 880
- tm-base-tm-limit-os-svc-receive-handler-path2, 769
 - handler-path3, 774
 - handler-path4, 778
- tm-base-tm-limit-os-svc-send-handler-path2, 753
 - ndler-path3, 758
 - ndler-path4, 762
- tm-base-tm-limit-os-svc-tyi-handler-path2, 785
 - dler-path3, 788
- tm-base-tm-limit-os-svc-tyo-handler-path2, 795
 - dler-path3, 798
 - dler-path4, 801
- tm-base-tm-limit-os-waiting-input-handler-path1, 806
 - ut-handler-path2, 811
 - ut-handler-path3, 816
- tm-base-tm-limit-os-waiting-output-handler-path1, 854

- ut-handler-path2, 859
- tm-br-opcode, 109
- tm-brnz-opcode, 109
- tm-brz-opcode, 109
- tm-call-opcode, 109
- tm-cc, 92–97, 103, 104, 112, 121, 122, 128, 167, 179, 206, 218–254, 256–272, 387, 432, 434, 436, 438, 448, 460, 533, 553, 673, 913, 921, 939, 947, 965, 982, 988
- tm-cc-divisor, 98
- tm-cc-mapup-address-space, 226
- tm-cc-value, 98, 103–106, 116, 117
- tm-cc-value-on-compare-equal, 116
- tm-cc-value-on-compare-greaterp, 117
- tm-cc-value-on-compare-lessp, 117
- tm-cc-value-on-compare-not-equal, 117
 - l-to-zero, 117
- tm-cclub, 91
- tm-ccsize, 91
- tm-charlub, 92
- tm-charsize, 91
- tm-clear-input-interrupt, 100, 114, 126, 280, 999, 1006, 1013, 1017, 1020, 1024, 1030, 1036, 1042, 1047, 1051, 1056
- tm-clear-output-interrupt, 101, 115, 126, 280, 673, 1063, 1074, 1084, 1094
- tm-cleared-svcflag, 100
- tm-clock, 92–97, 112, 122, 164, 168, 170–172, 176, 178, 179, 190–193, 372, 892, 896, 902, 906, 912, 919, 926, 931, 937, 945, 953, 958, 964, 970, 974, 981, 986, 992, 999, 1006, 1013, 1016, 1020, 1024, 1030, 1036, 1042, 1047, 1051, 1056, 1063, 1070, 1073, 1077, 1083, 1089, 1094, 1098
- tm-clock-interruptp, 112, 115, 193, 392, 883–885, 1103
- tm-clock-new-pc-addr, 99
- tm-clock-os-clock-interrupt-handler, 892
- tm-clock-os-error-handler-path1, 896
- tm-clock-os-error-handler-path2, 902
- tm-clock-os-running-input-handle
 - r-path1, 1030
 - r-path2, 1036
 - r-path3, 1042
 - r-path4, 1047
 - r-path5, 1051
 - r-path6, 1056
- tm-clock-os-running-output-hand
 - ler-path1, 1083
 - ler-path2, 1089
 - ler-path3, 1094
 - ler-path4, 1098
- tm-clock-os-svc-receive-handler
 - path1, 931
 - path2, 937
 - path3, 945
 - path4, 953
- tm-clock-os-svc-send-handler-pat
 - h1, 906
 - h2, 912
 - h3, 919
 - h4, 926
- tm-clock-os-svc-tyi-handler-pat
 - h1, 958
 - h2, 964
 - h3, 970
- tm-clock-os-svc-tyo-handler-pat
 - h1, 974
 - h2, 981
 - h3, 986
 - h4, 992
- tm-clock-os-waiting-input-handle
 - r-path1, 999
 - r-path2, 1006
 - r-path3, 1013
 - r-path4, 1016
 - r-path5, 1020
 - r-path6, 1024
- tm-clock-os-waiting-output-hand

- ler-path1, 1063
- ler-path2, 1070
- ler-path3, 1073
- ler-path4, 1077
- tm-clock-tm-fetch-execute, 193
- tm-compare-opcode, 110
- tm-compute-address, 101, 111, 264, 265
- tm-compute-address-mapup-address-space, 264
- tm-cpu, 128–130, 370, 387, 403, 432, 433, 435, 889, 898, 899, 909, 913, 914, 921, 933, 934, 938, 939, 947, 960, 961, 965, 977, 978, 982, 988, 996, 1003, 1010, 1026, 1032, 1038, 1044, 1048, 1053, 1060, 1067, 1079, 1085, 1091, 1095
- tm-cpu-length, 128
- tm-cpu-os-clock-interrupt-handle
 - r, 889
 - r-case1, 889
 - r-case2, 889
- tm-cpu-os-error-handler-path2, 899
- tm-cpu-os-error-handler-path2-case1, 898
- tm-cpu-os-error-handler-path2-case2, 898
- tm-cpu-os-running-input-handler
 - path1, 1026
 - path1-crock, 1026
 - path2, 1032
 - path2-crock, 1032
 - path3, 1038
 - path3-crock, 1038
 - path4, 1044
 - path4-crock, 1044
 - path5, 1048
 - path5-crock, 1048
 - path6, 1053
 - path6-crock, 1053
- tm-cpu-os-running-output-handle
 - r-path1, 1079
 - r-path1-crock, 1079
 - r-path2, 1085
 - r-path2-crock, 1085
 - r-path3, 1091
 - r-path3-crock, 1091
 - r-path4, 1095
 - r-path4-crock, 1095
- tm-cpu-os-svc-recv-handler-path
 - ath2, 934
 - ath2-case1, 933
 - ath2-case2, 934
 - ath3, 938
 - ath4, 947
- tm-cpu-os-svc-send-handler-path
 - 2-case1, 908
 - 2-case2, 909
 - 2, 909
 - 3, 913
 - 4, 921
- tm-cpu-os-svc-tyi-handler-path2, 961
 - case1, 960
 - case2, 960
- tm-cpu-os-svc-tyi-handler-path3, 965
- tm-cpu-os-svc-tyo-handler-path2, 978
 - case1, 977
 - case2, 977
- tm-cpu-os-svc-tyo-handler-path3, 982
- tm-cpu-os-svc-tyo-handler-path4, 988
- tm-cpu-os-waiting-input-handler
 - path1, 996
 - path2, 1003
 - path3, 1009
- tm-cpu-os-waiting-output-handle
 - r-path1, 1060
 - r-path2, 1066
- tm-datum1, 108
- tm-datum2, 108, 109
- tm-decr, 90, 91, 104, 112–114
- tm-decr-mod-opcode, 110
- tm-decr-opcode, 109
- tm-decrement-clock, 112, 160, 162, 216, 267, 268, 385
- tm-decrement-clock-commutes-with-mapup-address-space, 267
- tm-definition, 131
- tm-device-input-event, 115

tm-device-input-eventp, 115, 289, 309
 tm-device-output-event, 115
 tm-device-output-eventp, 115, 290, 309
 tm-error, 92–97, 112, 121, 122, 128, 163, 169–171, 176, 178, 207, 218–254, 256–272, 387, 433, 434, 436, 438, 448, 460, 533, 553, 673, 913, 921, 939, 947, 965, 982, 988, 1102
 tm-error-divisor, 98
 tm-error-mapup-address-space, 227
 tm-error-mapup-os-tasks, 1102
 tm-error-mapup-task, 1102
 tm-error-new-pc-addr, 99
 tm-errorlub, 91
 tm-errorp, 112, 115, 280, 439, 461, 498, 533, 553, 736, 737, 740, 747, 750, 751, 754, 758, 759, 762, 763, 766, 769, 770, 774, 775, 778, 779, 782, 785, 789, 792, 795, 796, 799, 802, 882–885, 892, 906, 912, 920, 926, 927, 931, 937, 945, 946, 953, 954, 958, 964, 971, 975, 981, 987, 993, 1101, 1103
 tm-errorsize, 91
 tm-execute, 111, 112, 160, 193, 200, 204, 216, 269, 271, 385
 tm-execute-add, 103, 110, 155, 163, 164, 196, 200, 210, 225, 226, 255, 381
 tm-execute-add-commutes-with-mapup-address-space, 225
 tm-execute-add-protection-theorem, 195
 tm-execute-binary, 110, 111, 160, 191, 192, 200, 204, 216, 255, 263, 264, 384
 tm-execute-binary-commutes-with-mapup-address-space, 263
 tm-execute-binary-protection-theorem, 200
 tm-execute-binary1, 255, 262, 263
 tm-execute-binary1-commutes-with-mapup-address-space, 262
 tm-execute-binary1-equals-tm-execute-binary, 255
 tm-execute-binary2, 255, 261, 262
 tm-execute-binary2-commutes-with-mapup-address-space, 260
 tm-execute-binary3, 255, 260
 tm-execute-binary3-commutes-with-mapup-address-space, 259
 tm-execute-binary4, 255, 258, 259
 tm-execute-binary4-commutes-with-mapup-address-space, 258
 tm-execute-binary5, 255, 257, 258
 tm-execute-binary5-commutes-with-mapup-address-space, 257
 tm-execute-binary6, 255–257
 tm-execute-binary6-commutes-with-mapup-address-space, 256
 tm-execute-branch, 103, 110, 155, 196, 200, 211, 226, 248, 381
 tm-execute-branch-commutes-with-mapup-address-space, 226
 tm-execute-branch-not-zero, 104, 110, 155, 196, 201, 211, 229, 248, 381
 tm-execute-branch-not-zero-commutes-with-mapup-address-space, 229
 tm-execute-branch-not-zero-protection-theorem, 196
 tm-execute-branch-on-zero, 103, 110, 155, 196, 201, 211, 228, 229, 248, 381
 tm-execute-branch-on-zero-commutes-with-mapup-address-space, 228
 tm-execute-branch-on-zero-protection-theorem, 196
 tm-execute-branch-protection-theorem, 196
 tm-execute-call, 104, 110, 155, 167, 168, 196, 201, 211, 231, 248, 381
 tm-execute-call-commutes-with-m

- apup-address-space, 230
- tm-execute-call-protection-theorem, 196
- tm-execute-clock-interrupt, 112, 115, 162, 439, 884
- tm-execute-commutes-with-mapup-address-space, 270
- tm-execute-compare, 104, 110, 155, 196, 201, 211, 231, 232, 255, 381
- tm-execute-compare-commutes-with-mapup-address-space, 231
- tm-execute-compare-protection-theorem, 196
- tm-execute-decr, 104, 110, 155, 169, 170, 196, 201, 211, 232, 248, 381
- tm-execute-decr-commutes-with-mapup-address-space, 232
- tm-execute-decr-mod, 104, 110, 156, 170, 171, 197, 201, 212, 233, 255, 382
- tm-execute-decr-mod-commutes-with-mapup-address-space, 232
- tm-execute-decr-mod-protection-theorem, 197
- tm-execute-decr-protection-theorem, 196
- tm-execute-error-interrupt, 113, 115, 162, 448, 884
- tm-execute-incr, 104, 110, 156, 171, 197, 201, 212, 233, 234, 248, 382
- tm-execute-incr-commutes-with-mapup-address-space, 233
- tm-execute-incr-mod, 104, 110, 156, 171, 172, 197, 202, 212, 234, 235, 255, 382
- tm-execute-incr-mod-commutes-with-mapup-address-space, 234
- tm-execute-incr-mod-protection-theorem, 197
- tm-execute-incr-protection-theorem, 197
- tm-execute-input-interrupt, 114, 115, 162, 578, 884
- tm-execute-instruction, 111, 160, 192, 200, 204, 216, 265, 267, 269, 384
- tm-execute-instruction-commutes-with-mapup-address-space, 266
- tm-execute-instruction-protection-theorem, 200
- tm-execute-instruction1, 265, 266
- tm-execute-instruction1-commutes-with-mapup-address-space, 266
- tm-execute-instruction1-equals-tm-execute-instruction, 265
- tm-execute-instruction2, 264–266
- tm-execute-instruction2-commutes-with-mapup-address-space, 265
- tm-execute-load-base, 105, 110, 156, 197, 202, 212, 235, 248, 382
- tm-execute-load-base-commutes-with-mapup-address-space, 235
- tm-execute-load-base-protection-theorem, 197
- tm-execute-load-limit, 105, 110, 156, 197, 202, 212, 236, 247, 382
- tm-execute-load-limit-commutes-with-mapup-address-space, 236
- tm-execute-load-limit-protection-theorem, 197
- tm-execute-load-psw, 105, 110, 157, 197, 202, 212, 236, 247, 382
- tm-execute-load-psw-commutes-with-mapup-address-space, 236
- tm-execute-load-psw-protection-theorem, 197
- tm-execute-mod, 105, 110, 157, 176, 177, 198, 202, 213, 237, 255, 382
- tm-execute-mod-commutes-with-mapup-address-space, 237
- tm-execute-mod-protection-theorem, 198
- tm-execute-move, 106, 110, 157, 198, 202, 213, 238, 255, 383

tm-execute-move-commutes-with-mapup-address-space, 237
 tm-execute-move-protection-theorem, 198
 tm-execute-multiply, 106, 111, 157, 178, 198, 202, 213, 238, 239, 255, 383
 tm-execute-multiply-commutes-with-mapup-address-space, 238
 tm-execute-multiply-protection-theorem, 198
 tm-execute-nullary, 109, 111, 159, 190, 199, 204, 215, 245, 247, 265, 384
 tm-execute-nullary-commutes-with-mapup-address-space, 246
 tm-execute-nullary-protection-theorem, 199
 tm-execute-nullary1, 245, 246
 tm-execute-nullary1-commutes-with-mapup-address-space, 246
 tm-execute-nullary1-equals-tm-execute-nullary, 245
 tm-execute-nullary2, 245, 246
 tm-execute-nullary2-commutes-with-mapup-address-space, 245
 tm-execute-output-interrupt, 114, 115, 163, 674, 884
 tm-execute-post-output-interrupt, 106, 110, 159, 199, 203, 215, 243, 244, 247, 383
 -commutes-with-mapup-address-space, 243
 -protection-theorem, 199
 tm-execute-protection-theorem, 200
 tm-execute-return, 106, 109, 158, 179, 198, 202, 214, 239, 240, 245, 383
 tm-execute-return-commutes-with-mapup-address-space, 239
 tm-execute-return-protection-theorem, 198
 tm-execute-run, 106, 109, 158, 199, 203, 214, 240, 241, 245, 383
 tm-execute-run-commutes-with-mapup-address-space, 240
 tm-execute-run-protection-theorem, 199
 tm-execute-set-clock, 106, 110, 158, 198, 203, 214, 240, 247, 383
 tm-execute-set-clock-commutes-with-mapup-address-space, 240
 tm-execute-set-clock-protection-theorem, 198
 tm-execute-start-output, 106, 111, 159, 199, 203, 215, 243, 255, 383
 tm-execute-start-output-commutes-with-mapup-address-space, 242
 tm-execute-start-output-protection-theorem, 199
 tm-execute-svc, 106, 107, 110, 159, 199, 203, 215, 244, 247, 383
 tm-execute-svc-commutes-with-mapup-address-space, 244
 tm-execute-svc-interrupt, 113, 115, 162, 461, 498, 533, 553, 882, 884
 tm-execute-svc-protection-theorem, 199
 tm-execute-svc-return, 107, 110, 159, 199, 204, 215, 245, 247, 384
 tm-execute-svc-return-commutes-with-mapup-address-space, 244
 tm-execute-svc-return-protection-theorem, 199
 tm-execute-test-iport, 107, 110, 158, 198, 203, 214, 241, 242, 247, 384
 tm-execute-test-iport-commutes-with-mapup-address-space, 241
 tm-execute-test-iport-protection-theorem, 198
 tm-execute-test-oport, 107, 110, 158, 198, 203, 214, 242, 247, 384
 tm-execute-test-oport-commutes-with-mapup-address-space, 242
 tm-execute-test-oport-protection-theorem, 198

- n-theorem, 198
- tm-execute-unary, 110, 111, 160, 190, 191, 199, 204, 215, 248, 254, 255, 265, 384
- tm-execute-unary-commutes-with-mapup-address-space, 254
- tm-execute-unary-protection-theorem, 199
- tm-execute-unary1, 248, 254
- tm-execute-unary1-commutes-with-mapup-address-space, 253
- tm-execute-unary1-equals-tm-execute-unary, 248
- tm-execute-unary2, 248, 253
- tm-execute-unary2-commutes-with-mapup-address-space, 253
- tm-execute-unary3, 248, 252
- tm-execute-unary3-commutes-with-mapup-address-space, 252
- tm-execute-unary4, 248, 251, 252
- tm-execute-unary4-commutes-with-mapup-address-space, 251
- tm-execute-unary5, 247, 248, 251
- tm-execute-unary5-commutes-with-mapup-address-space, 250
- tm-execute-unary6, 247, 250
- tm-execute-unary6-commutes-with-mapup-address-space, 250
- tm-execute-unary7, 247, 249, 250
- tm-execute-unary7-commutes-with-mapup-address-space, 249
- tm-execute-unary8, 247, 249
- tm-execute-unary8-commutes-with-mapup-address-space, 248
- tm-execute-wait, 107, 109, 158, 199, 203, 214, 241, 245, 384
- tm-execute-wait-commutes-with-mapup-address-space, 241
- tm-execute-wait-protection-theorem, 199
- tm-execute1, 269, 270
- tm-execute1-commutes-with-mapup-address-space, 270
- tm-execute1-equals-tm-execute, 269
- tm-execute2, 268–270
- tm-execute2-commutes-with-mapup-address-space, 269
- tm-fetch, 103–107, 111, 132, 156, 209, 219
- tm-fetch-arg1, 111, 268, 269
- tm-fetch-arg1-mapup-address-space, 268
- tm-fetch-arg2, 111, 268, 269
- tm-fetch-arg2-mapup-address-space, 268
- tm-fetch-execute, 112, 115, 162, 193, 200, 204, 216, 272, 274, 280, 316, 385–388, 390–392, 883, 884
- tm-fetch-execute-commutes-with-mapup-address-space, 271
- tm-fetch-execute-mapup-address-space-separation, 386
- tm-fetch-execute-protection-theorem, 200
- tm-fetch-from-memory, 102, 103, 112, 131, 218
- tm-fetch-from-memory-mapup-address-space, 217
- tm-fetch-from-regmem, 102, 103, 218
- tm-fetch-from-regmem-mapup-address-space, 218
- tm-fetch-mapup-address-space, 218
- tm-fetch-new-pc-on-interrupt, 112–115
- tm-fetch-opcode, 111, 112, 271
- tm-fetch-opcode-mapup-address-space, 271
- tm-fix-reg-address, 101, 118
- tm-fix-reg-address-noop, 118
- tm-fix-reg-address-size, 118
- tm-fix-reg-address-size-linear, 118
- tm-fix-word, 101, 117
- tm-fix-word-noop, 117
- tm-fix-word-size, 117
- tm-fix-word-size-linear, 117
- tm-good-address, 102–104, 110–112, 132, 155–160, 162, 195–198,

208, 209, 211, 213–215, 218,
 220, 225, 226, 228–234, 237,
 238, 244, 248–253, 255–260,
 268, 269, 271, 379–383
 tm-good-address-mapup-address-sp
 ace, 230
 tm-good-address-tm-decrement-cl
 ock, 162
 tm-good-address-tm-incr-tm-sp-f
 or-address-space, 213
 tm-good-address-tm-set-cc, 195
 tm-good-address-tm-set-pc, 195
 tm-good-address-tm-set-sp, 195
 tm-good-address-tm-store, 162
 tm-good-memory-tm-store, 133
 tm-good-pc-address, 112
 tm-good-regs-tm-store, 133
 tm-good-supervisor-address, 156, 157
 tm-good-supervisor-address-tm-g
 ood-address, 157
 tm-good-supervisor-address-tm-i
 ncr-address, 157
 ncrn-address, 157
 tm-halt-status, 99
 tm-ichar, 100, 114, 118, 125, 127,
 287, 290, 306, 424, 425, 998,
 1005, 1012, 1016, 1020, 1023,
 1029, 1035, 1041, 1046, 1051,
 1055
 tm-ichar-facts, 127
 tm-idatum, 115, 289
 tm-idevid, 115, 289
 tm-ierror-flag, 100, 118, 125, 127,
 290, 306
 tm-iinterrupt-flag, 100, 113, 118, 125,
 290, 306
 tm-implements-os, 887
 tm-implements-timed-tm, 885
 tm-in-supervisor-mode, 100, 102, 103,
 105–108, 112–114, 131, 132,
 156, 190–193, 195–200, 208,
 217–220, 223–226, 228–254,
 256–259, 261–271, 362, 378,
 379, 737, 741, 744, 748, 751,
 755, 760, 764, 767, 771, 776,
 780, 783, 786, 790, 793, 797,
 800, 804, 808, 813, 817, 820,
 823, 827, 831, 836, 841, 844,
 848, 852, 856, 860, 863, 867,
 871, 875, 878, 1102
 tm-in-supervisor-mode-good-addre
 ss-space, 208
 tm-in-supervisor-mode-mapup-add
 ress-space, 235
 tm-in-supervisor-mode-tm-decreme
 nt-clock, 268
 tm-in-supervisor-mode-tm-set-cc, 223
 tm-in-supervisor-mode-tm-set-pc, 224
 tm-in-supervisor-mode-tm-set-sp, 224
 tm-in-supervisor-mode-tm-set-sv
 cflag, 224
 cid, 224
 tm-incr, 90, 91, 102, 106, 158, 213,
 214
 tm-incr-address, 102, 157
 tm-incr-mod-opcode, 110
 tm-incr-opcode, 109
 tm-incr-sp-is-add1-sp, 158
 tm-incrn, 90, 91, 111, 268, 269
 tm-incrn-address, 102, 105, 107, 112,
 157
 tm-index-size, 108
 tm-index1, 108
 tm-index1-divisor, 108
 tm-index2, 108, 109
 tm-input-char-addr, 99
 tm-input-devid-addr, 99
 tm-input-interruptp, 113, 115, 883–
 885, 1103
 tm-input-new-pc-addr, 99
 tm-interrupt-addresses-os-clock
 -interrupt-handler, 737
 tm-interrupt-addresses-os-error
 -handler-path1, 741
 -handler-path2, 744
 tm-interrupt-addresses-os-runni
 ng-input-handler-path1, 826
 ng-input-handler-path2, 831

ng-input-handler-path3, 836
 ng-input-handler-path4, 840
 ng-input-handler-path5, 844
 ng-input-handler-path6, 847
 ng-output-handler-path1, 866
 ng-output-handler-path2, 870
 ng-output-handler-path3, 875
 ng-output-handler-path4, 878
 tm-interrupt-addresses-os-svc-re
 ceive-handler-path1, 763
 ceive-handler-path2, 767
 ceive-handler-path3, 771
 ceive-handler-path4, 776
 tm-interrupt-addresses-os-svc-se
 nd-handler-path1, 748
 nd-handler-path2, 751
 nd-handler-path3, 755
 nd-handler-path4, 760
 tm-interrupt-addresses-os-svc-t
 yi-handler-path1, 779
 yi-handler-path2, 782
 yi-handler-path3, 786
 yo-handler-path1, 790
 yo-handler-path2, 793
 yo-handler-path3, 796
 yo-handler-path4, 799
 tm-interrupt-addresses-os-waiti
 ng-input-handler-path1, 803
 ng-input-handler-path2, 808
 ng-input-handler-path3, 812
 ng-input-handler-path4, 817
 ng-input-handler-path5, 820
 ng-input-handler-path6, 823
 ng-output-handler-path1, 851
 ng-output-handler-path2, 855
 ng-output-handler-path3, 860
 ng-output-handler-path4, 863
 tm-interrupting-input-port, 113, 114,
 125, 127, 281, 307, 312, 420–
 422, 424, 425, 582, 585, 588,
 591, 600–603, 605, 608, 611,
 613, 621, 623–625, 630, 633,
 636, 638, 639, 641, 643, 645,
 647, 648, 650, 652, 655, 658,
 661, 668, 670–672, 802–807,
 809–816, 818, 819, 821, 822,
 824–835, 837–841, 843, 845–
 848, 850, 998–1000, 1005–
 1007, 1012–1014, 1016, 1017,
 1020, 1021, 1023–1043, 1046,
 1047, 1051, 1052, 1055–1057,
 1101
 tm-interrupting-output-port, 114, 115,
 125–127, 281, 309, 312, 429,
 431, 673, 677, 680, 682, 689,
 691, 694, 699, 701–704, 709,
 711, 714, 716, 718, 720, 723,
 725, 730, 732, 734–736, 851–
 859, 862, 864–874, 877, 879,
 881, 1062–1064, 1069–1071,
 1074, 1077–1090, 1094, 1098,
 1099, 1101
 tm-iport, 100
 tm-iport-error-flag-is-zero, 127
 tm-iport-errorrp, 100, 107, 127, 287,
 588, 591, 602, 603, 611, 613,
 623–625, 636, 639, 648, 650,
 658, 661, 670–672, 807, 811,
 812, 816, 819, 822, 825, 831,
 835, 839, 843, 846, 847, 850,
 1007, 1014, 1021, 1025, 1037,
 1043, 1052, 1056
 tm-iportp, 118, 125, 290, 306
 tm-iports, 92–97, 107, 113–115, 123–
 127, 162, 164, 168, 170–172,
 176, 178, 179, 190–193, 372,
 420–422, 424, 425, 580, 582,
 585, 588, 591, 593, 596, 598,
 600–603, 605, 608, 611, 613,
 616, 618, 620, 621, 623–625,
 628, 630, 633, 636, 638, 639,
 641, 643, 645, 647, 648, 650,
 652, 655, 658, 661, 663, 665,
 666, 668, 670–673, 802–807,
 809–816, 818, 819, 821, 822,
 824–835, 837–841, 843, 845–
 848, 850, 882, 883, 892, 896,
 902, 906, 912, 919, 926, 931,

- 937, 945, 953, 958, 964, 970,
974, 981, 986, 992, 996–1000,
1003–1007, 1010–1014, 1016,
1017, 1020, 1021, 1023–1057,
1063, 1070, 1074, 1077, 1084,
1089, 1094, 1098, 1101
- tm-iports-os-clock-interrupt-handler, 892
- tm-iports-os-error-handler-path
 - 1, 896
 - 2, 902
- tm-iports-os-running-input-handler-path1, 1030
- tm-iports-os-running-input-handler-path2, 1036
- tm-iports-os-running-input-handler-path3, 1042
- tm-iports-os-running-input-handler-path4, 1047
- tm-iports-os-running-input-handler-path5, 1051
- tm-iports-os-running-input-handler-path6, 1056
- tm-iports-os-running-output-handler-path1, 1084
- tm-iports-os-running-output-handler-path2, 1089
- tm-iports-os-running-output-handler-path3, 1094
- tm-iports-os-running-output-handler-path4, 1098
- tm-iports-os-svc-receive-handler-path1, 931
- tm-iports-os-svc-receive-handler-path2, 937
- tm-iports-os-svc-receive-handler-path3, 945
- tm-iports-os-svc-receive-handler-path4, 953
- tm-iports-os-svc-send-handler-path1, 906
- tm-iports-os-svc-send-handler-path2, 912
- tm-iports-os-svc-send-handler-path3, 919
- tm-iports-os-svc-send-handler-path4, 926
- tm-iports-os-svc-tyi-handler-path1, 958
- tm-iports-os-svc-tyi-handler-path2, 964
- tm-iports-os-svc-tyi-handler-path3, 970
- tm-iports-os-svc-tyo-handler-path1, 974
- tm-iports-os-svc-tyo-handler-path2, 981
- tm-iports-os-svc-tyo-handler-path3, 986
- tm-iports-os-svc-tyo-handler-path4, 992
- tm-iports-os-waiting-input-handler-path1, 999
- tm-iports-os-waiting-output-handler-path1, 1063
- tm-iports-os-waiting-output-handler-path2, 1070
- tm-iports-os-waiting-output-handler-path3, 1074
- tm-iports-os-waiting-output-handler-path4, 1077
- tm-lbase-opcode, 109
- tm-limit, 92–97, 102, 103, 121, 131,
160, 163, 167, 169–171, 176,
178, 179, 190–193, 195–200,
207, 208, 217–272, 363, 379,
380, 739, 740, 746, 747, 753,
754, 758, 762, 769, 774, 778,
785, 788, 795, 798, 801, 802,
806, 811, 815, 816, 829, 830,
834, 839, 842, 843, 846, 849,
850, 854, 859, 869, 873, 874,
877, 880
- tm-limit-address-space, 207
- tm-limit-mapup-address-space, 224
- tm-limit-os, 363
- tm-limit-os-clock-interrupt-handler, 739
- tm-limit-os-error-handler-path2, 746
- tm-limit-os-running-input-handler-path1, 829
- tm-limit-os-running-input-handler-path2, 834
- tm-limit-os-running-input-handler-path3, 839
- tm-limit-os-running-input-handler-path4, 842
- tm-limit-os-running-input-handler-path5, 846
- tm-limit-os-running-input-handler-path6, 849
- tm-limit-os-running-output-handler-path1, 869
- tm-limit-os-running-output-handler-path2, 873
- tm-limit-os-running-output-handler-path3, 877
- tm-limit-os-running-output-handler-path4, 880
- tm-limit-os-svc-receive-handler-path2, 769
- tm-limit-os-svc-receive-handler-path3, 774
- tm-limit-os-svc-receive-handler-path4, 778
- tm-limit-os-svc-receive-handler-path5, 1006
- tm-limit-os-svc-receive-handler-path6, 1013
- tm-limit-os-svc-receive-handler-path7, 1017
- tm-limit-os-svc-receive-handler-path8, 1020
- tm-limit-os-svc-receive-handler-path9, 1024

tm-limit-os-svc-send-handler-pat
 h2, 753
 h3, 758
 h4, 762
 tm-limit-os-svc-tyi-handler-pat
 h2, 785
 h3, 788
 tm-limit-os-svc-tyo-handler-pat
 h2, 795
 h3, 798
 h4, 801
 tm-limit-os-waiting-input-handle
 r-path1, 806
 r-path2, 811
 r-path3, 815
 tm-limit-os-waiting-output-hand
 ler-path1, 854
 ler-path2, 859
 tm-llimit-opcode, 109
 tm-lpsw-opcode, 109
 tm-memlength, 92
 tm-memory, 92–97, 102, 120, 133,
 179, 195–204, 206, 209, 217–
 254, 256–272, 276, 277, 290,
 291, 294, 348, 362, 371, 375,
 379, 380, 386, 387, 438, 448,
 461, 533, 553, 673, 737, 741,
 744, 748, 751, 755, 760, 763,
 764, 767, 771, 776, 779, 780,
 782, 783, 786, 790, 793, 796,
 799, 800, 803, 808, 812, 813,
 817, 820, 823, 826, 827, 831,
 836, 840, 844, 847, 848, 851,
 852, 855, 856, 860, 863, 866,
 870, 871, 875, 878, 887, 893,
 897, 903, 907, 913, 921, 928,
 932, 938, 947, 955, 959, 965,
 971, 976, 982, 988, 994, 1001,
 1007, 1014, 1018, 1022, 1025,
 1031, 1037, 1038, 1044, 1048,
 1053, 1057, 1064, 1071, 1075,
 1079, 1085, 1091, 1095
 tm-mod-opcode, 110
 tm-mode-size, 108
 tm-mode1, 108
 tm-mode1-divisor, 108
 tm-mode2, 108, 109
 tm-mode2-divisor, 108
 tm-move-opcode, 110
 tm-mult-opcode, 110
 tm-no-error, 99
 tm-no-svc, 100
 tm-non-zero-carry-condition, 97
 tm-non-zero-no-carry-condition, 97
 tm-nullary-instructionp, 108, 111, 269,
 322, 327
 tm-obusy-flag, 100, 101, 118, 126,
 290, 308
 tm-ochar, 101, 118, 126, 290, 308
 tm-odevid, 115, 290
 tm-ointerrupt-flag, 114, 118, 126, 290,
 308
 tm-opcode, 108, 111, 264, 265, 330
 tm-opcode-divisor, 108
 tm-opcode-error, 99
 tm-opcode-size, 108
 tm-oport, 100, 101
 tm-oport-idlep, 100, 107, 280, 568,
 570, 575–577, 796, 799, 802,
 987, 993
 tm-oportp, 118, 126, 290, 308
 tm-oports, 92–97, 106, 107, 114, 115,
 123–127, 163, 164, 168, 170–
 172, 177–179, 190–193, 372,
 429, 431, 568, 570, 575–577,
 673, 675, 677, 680, 682, 685,
 687, 689, 691, 694, 696, 697,
 699, 701–704, 706, 709, 711,
 714, 716, 718, 720, 722, 723,
 725, 727, 729, 730, 732–736,
 796, 799, 802, 850–859, 862,
 864–874, 877, 879–881, 883,
 892, 896, 902, 906, 912, 919,
 926, 931, 937, 945, 953, 958,
 964, 970, 974, 981, 987, 992,
 993, 999, 1006, 1013, 1017,
 1021, 1024, 1031, 1037, 1043,
 1047, 1052, 1056, 1060–1064,

1066–1071, 1074, 1077–1099,
 1101
 tm-oports-os-clock-interrupt-handler, 892
 tm-oports-os-error-handler-path
 1, 896
 2, 902
 tm-oports-os-running-input-handler-path1, 1031
 ler-path2, 1037
 ler-path3, 1043
 ler-path4, 1047
 ler-path5, 1052
 ler-path6, 1056
 tm-oports-os-running-output-handler-path1, 1084
 dler-path2, 1090
 dler-path3, 1094
 dler-path4, 1099
 tm-oports-os-svc-recv-handle
 r-path1, 931
 r-path2, 937
 r-path3, 945
 r-path4, 953
 tm-oports-os-svc-send-handler-path1, 906
 ath2, 912
 ath3, 919
 ath4, 926
 tm-oports-os-svc-tyi-handler-path1, 958
 h2, 964
 h3, 970
 tm-oports-os-svc-tyo-handler-path1, 974
 h2, 981
 h3, 987
 h4, 992
 tm-oports-os-waiting-input-handler-path1, 999
 ler-path2, 1006
 ler-path3, 1013
 ler-path4, 1017
 ler-path5, 1021
 ler-path6, 1024
 tm-oports-os-waiting-output-handler-path1, 1063
 dler-path2, 1070
 dler-path3, 1074
 dler-path4, 1077
 tm-output-devid-addr, 99
 tm-output-interruptp, 114, 115, 883–885, 1103
 tm-output-new-pc-addr, 99
 tm-overflow-char, 113, 127, 287, 424, 998, 1005, 1016, 1020, 1029, 1035, 1046, 1051
 tm-pack-psw, 98, 112, 122, 128, 387, 389, 390, 396, 438, 448, 460, 533, 553, 673, 913–915, 921, 922, 939, 940, 947, 948, 965, 966, 982, 983, 988, 989
 tm-pack-psw-tm-unpack-fields, 396
 tm-pack-psw-tm-unpack-fields1, 396
 tm-pack-psw-tm-unpack-fields2, 396
 tm-pc, 97, 104, 111, 112, 122, 207, 228, 268, 269, 271, 438, 447, 460, 533, 553, 673
 tm-pc-address, 92
 tm-pc-address-error, 99
 tm-pc-mapup-address-space, 227
 tm-port-length, 92
 tm-post-input-interrupt, 100, 115, 163, 289
 tm-post-interrupt, 115, 116, 163, 385, 884–886, 1103, 1104
 tm-post-opcode, 109
 tm-post-output-interrupt, 101, 106, 115, 159, 280, 290, 987
 tm-privilege-error, 99
 tm-processor, 115, 116, 163, 440, 442, 444, 446, 447, 450, 452, 454, 455, 457–460, 463, 465, 468, 470, 472, 475, 477, 480, 483, 485–498, 501, 503, 505, 507, 510, 512, 514, 517, 519, 521–532, 535, 537, 539, 541, 544, 546, 548–553, 555, 557, 560,

562, 564, 566, 568, 570–577,
 580, 582, 585, 588, 591, 593,
 596, 598–603, 606, 608, 611,
 613, 616, 618, 620–625, 628,
 630, 633, 636, 639, 641, 643,
 645–650, 653, 655, 658, 661,
 663, 665, 667–673, 675, 678,
 680, 682, 685, 687, 689, 691,
 694, 696–704, 706, 709, 711,
 714, 716, 718, 720, 723, 725,
 727, 729–736, 882–885, 887,
 1105
 tm-processor-append-oracle, 116
 tm-processor-implements-timed-t
 m-step, 885
 tm-processor-oracle-distributes, 885
 tm-processor-with-null-oracle, 116
 tm-r0, 128
 tm-r1, 128
 tm-r2, 128, 273, 1100
 tm-r3, 128, 273, 918, 925, 1100
 tm-r4, 128
 tm-r5, 128
 tm-r6, 128
 tm-r7, 128
 tm-register-save-area-addr, 99
 tm-reglength, 92
 tm-regs, 92–97, 101, 102, 121, 128,
 130, 133, 206, 209, 218–254,
 256–272, 407, 408, 411, 413,
 415, 431, 432, 434, 435, 1100
 tm-regs-getnth-ak-pstates, 1100
 tm-regs-mapup-os-tasks, 1100
 tm-regs-mapup-task, 1100
 tm-regs-size, 91
 tm-return-opcode, 109
 tm-run-opcode, 109
 tm-run-state, 100
 tm-rwstate, 92–97, 100, 123, 164,
 168, 170–172, 176, 178, 179,
 190–193, 372, 892, 896, 902,
 906, 912, 919, 926, 931, 937,
 945, 953, 958, 964, 970, 974,
 981, 987, 992, 999, 1006,
 1013, 1016, 1020, 1024, 1030,
 1036, 1042, 1047, 1051, 1056,
 1063, 1070, 1073, 1077, 1083,
 1089, 1094, 1098
 tm-rwstate-os-clock-interrupt-h
 andler, 892
 tm-rwstate-os-error-handler-pat
 h1, 896
 h2, 902
 tm-rwstate-os-running-input-han
 dler-path1, 1030
 dler-path2, 1036
 dler-path3, 1042
 dler-path4, 1047
 dler-path5, 1051
 dler-path6, 1056
 tm-rwstate-os-running-output-ha
 ndler-path1, 1083
 ndler-path2, 1089
 ndler-path3, 1094
 ndler-path4, 1098
 tm-rwstate-os-svc-recv-handle
 r-path1, 931
 r-path2, 937
 r-path3, 945
 r-path4, 953
 tm-rwstate-os-svc-send-handler-p
 ath1, 906
 ath2, 912
 ath3, 919
 ath4, 926
 tm-rwstate-os-svc-tyi-handler-p
 ath1, 958
 ath2, 964
 ath3, 970
 tm-rwstate-os-svc-tyo-handler-p
 ath1, 974
 ath2, 981
 ath3, 987
 ath4, 992
 tm-rwstate-os-waiting-input-han
 dler-path1, 999
 dler-path2, 1006
 dler-path3, 1013

- dler-path4, 1016
 - dler-path5, 1020
 - dler-path6, 1024
- tm-rwstate-os-waiting-output-ha
 - ndler-path1, 1063
 - ndler-path2, 1070
 - ndler-path3, 1073
 - ndler-path4, 1077
- tm-set-base, 94, 105, 139
- tm-set-cc, 93, 103–107, 136, 195, 210, 222, 223, 225, 378
- tm-set-cc-commutes-with-mapup-address-space, 222
- tm-set-clock, 96, 106, 112, 148
- tm-set-error, 93, 104–115, 138, 210, 222, 245, 247, 248, 255, 264, 269, 378, 439, 448, 461, 533, 553, 673
- tm-set-error-commutes-with-mapup-address-space, 222
- tm-set-iptables, 96, 114, 115, 150, 385
- tm-set-limit, 94, 95, 105, 140
- tm-set-memory, 92, 102, 135, 439, 448, 461, 533, 553, 673
- tm-set-ports, 97, 106, 115, 151, 385, 673
- tm-set-pc, 97, 103–107, 111, 112, 152, 160, 195, 210, 221, 224, 269, 378, 439, 448, 461, 533, 553, 673
- tm-set-pc-commutes-with-mapup-address-space, 221
- tm-set-regs, 92, 97, 102, 133
- tm-set-rwstate, 96, 106, 108, 114, 115, 147, 439, 448, 461, 533, 553, 673
- tm-set-slimit, 95, 142
- tm-set-sp, 97, 104–107, 112–115, 154, 195, 210, 221, 224, 225, 378, 439, 448, 461, 533, 553, 673
- tm-set-sp-commutes-with-mapup-address-space, 221
- tm-set-svcflag, 93, 105, 107, 144, 210, 223, 224, 273, 279, 378, 916, 917, 923, 924, 942, 943, 946, 950, 951, 968, 969, 984, 985, 990, 991
- tm-set-svcflag-commutes-with-mapup-address-space, 223
- tm-set-svcflag-commutes-with-os-store-message, 946
- tm-set-svcid, 94, 105, 107, 146, 210, 223, 224, 378
- tm-set-svcid-commutes-with-mapup-address-space, 223
- tm-set-svmode, 95, 105, 107, 108, 112–115, 143, 439, 448, 461, 533, 553, 673
- tm-shellp, 120, 205
- tm-shellp-address-space, 205
- tm-shellp-tm, 120
- tm-shells, 126
- tm-slimit, 92–97, 103, 112–114, 121, 163, 168–172, 176, 178, 179, 190–193, 360, 737, 740, 744, 748, 751, 755, 759, 763, 766, 770, 775, 779, 782, 786, 789, 792, 796, 799, 803, 807, 812, 816, 819, 823, 826, 831, 835, 840, 843, 847, 851, 855, 860, 862, 866, 870, 874, 878
- tm-slimit-os, 360
- tm-slimit-os-clock-interrupt-handler, 737
- tm-slimit-os-error-handler-path
 - 1, 740
 - 2, 744
- tm-slimit-os-running-input-handler-path
 - 1, 826
 - 2, 831
 - 3, 835
 - 4, 840
 - 5, 843
 - 6, 847
- tm-slimit-os-running-output-handler-path
 - 1, 866
 - 2, 870
 - 3, 874

tm-store-in-memory-protection-theorem, 195
 tm-store-in-regmem, 102, 103, 220, 378
 tm-store-in-regmem-commutes-with-mapup-address-space, 219
 tm-store-interrupting-input-device, 114
 tm-store-interrupting-output-device, 114, 115
 tm-store-old-psw-on-interrupt, 112–115
 tm-store-protection-theorem, 195
 tm-store-svcid-on-interrupt, 113
 tm-stout-opcode, 110
 tm-supervisor-mode, 99
 tm-svc, 100
 tm-svc-interruptp, 113, 115, 273, 280, 883–885, 1103
 tm-svc-new-pc-addr, 99
 tm-svc-opcode, 109
 tm-svc-return-opcode, 109
 tm-svcflag, 92–97, 112, 113, 122, 128, 163, 164, 168–172, 176, 178, 179, 190, 191, 207, 218–254, 256–272, 387, 433, 438, 448, 460, 533, 553, 673, 1102
 tm-svcflag-divisor, 98
 tm-svcflag-mapup-address-space, 227
 tm-svcflag-mapup-os-tasks, 1102
 tm-svcflag-mapup-task, 1102
 tm-svcflaglub, 92
 tm-svcflagsize, 91
 tm-svcid, 92–97, 112, 113, 122, 128, 164, 168–172, 176, 178, 179, 190, 191, 207, 218–254, 256–273, 279, 387, 433, 434, 436, 438, 448, 460, 465, 467, 475, 488, 489, 492, 494, 496, 498, 500, 503, 509, 522, 524, 526, 528, 531–533, 537, 549, 550, 552, 553, 557, 560, 566, 572, 573, 575–577, 673, 747, 750, 751, 754, 758, 759, 762, 763, 766, 770, 774, 775, 778, 779, 782, 785, 789, 792, 795, 796, 799, 802, 881, 906, 913, 920, 921, 927, 931, 938, 939, 946, 947, 954, 958, 964, 965, 971, 975, 981, 982, 987, 988, 993, 1099, 1100
 tm-svcid-addr, 99
 tm-svcid-getnth-ak-pstates, 1100
 tm-svcid-mapup-address-space, 227
 tm-svcid-mapup-os-tasks, 1100
 tm-svcid-mapup-task, 1099
 tm-svcidlub, 92
 tm-svcidsize, 91
 tm-svmode, 92–97, 100, 122, 163, 168–172, 176, 178, 179, 190–193, 208
 tm-svmode-address-space, 208
 tm-testi-opcode, 109
 tm-testo-opcode, 109
 tm-time-opcode, 109
 tm-unary-instructionp, 108, 111, 269, 323, 327
 tm-unpack-cc, 98, 105, 107, 371, 389, 396, 403
 tm-unpack-cc-tm-pack-psw, 389
 tm-unpack-error, 98, 105, 107, 371, 389, 396, 403
 tm-unpack-error-tm-pack-psw, 389
 tm-unpack-svcflag, 98, 99, 105, 371, 389, 396, 403
 tm-unpack-svcflag-tm-pack-psw, 389
 tm-unpack-svcid, 98, 99, 105, 107, 371, 390, 396, 403
 tm-unpack-svcid-tm-pack-psw, 390
 tm-user-mode, 100
 tm-wait-opcode, 109
 tm-wait-state, 100
 tm-waiting, 100, 115, 363–365, 370, 377, 380–388, 390–392, 404, 411, 412, 420, 429, 432–436, 439, 440, 442, 444, 446–448, 450, 452, 454, 455, 457, 458, 460, 461, 463, 465, 467, 470,

472, 475, 477, 480, 482, 485–489, 492, 494, 496, 498, 500, 503, 505, 507, 509, 512, 514, 517, 519, 521, 522, 524, 526, 528, 531–533, 535, 537, 539, 541, 543, 546, 548–550, 552, 553, 555, 557, 560, 562, 564, 566, 568, 570, 572, 573, 575–577, 580, 582, 585, 588, 591, 593, 596, 598, 600, 601, 603, 605, 608, 611, 613, 616, 618, 620, 621, 623–625, 628, 630, 633, 636, 638, 641, 643, 645, 647, 648, 650, 652, 655, 658, 661, 663, 665, 666, 668, 670, 671, 673, 675, 677, 680, 682, 685, 687, 689, 691, 694, 696, 697, 699, 701–704, 706, 709, 711, 714, 716, 718, 720, 722, 725, 727, 729, 730, 732, 733, 735–740, 743, 744, 746, 747, 750, 753, 754, 757–759, 761–763, 765, 766, 768–775, 777–779, 781, 782, 784, 785, 787–789, 791, 792, 794–799, 801–803, 806, 807, 810–812, 815, 816, 819, 822, 825, 826, 829, 830, 833–835, 838–840, 842, 843, 846, 847, 849–851, 853–855, 858, 859, 862, 864, 865, 868–870, 873, 874, 876, 877, 880–885, 889–896, 898–906, 908–931, 933–958, 960–975, 977–993, 996–1000, 1003–1007, 1010–1019, 1021–1057, 1060–1062, 1064, 1066–1076, 1078–1103
 tm-waiting-iff-array-qempty-os-readyq, 364
 tm-waiting-os-error-handler-path1, 743
 tm-waiting-os-svc-receive-handler-path1, 766
 tm-waiting-os-svc-send-handler-path1, 750
 tm-waiting-os-svc-tyi-handler-path1, 781
 tm-waiting-os-svc-tyo-handler-path1, 792
 tm-waiting-os-waiting-input-handler-path4, 819
 tm-waiting-os-waiting-input-handler-path5, 822
 tm-waiting-os-waiting-input-handler-path6, 825
 tm-waiting-os-waiting-output-handler-path3, 862
 tm-waiting-os-waiting-output-handler-path4, 864
 tm-waiting-tm-fetch-execute, 388
 tm-wordlub, 85
 tm-wordsz, 85
 tm-zero-carry-condition, 97
 tm-zero-no-carry-condition, 97
 trace-arithmetic, 395
 trace-array-queue-facts, 396
 trace-clock-interrupt-handler, 447
 trace-clock-interrupt-handler-thru-save-state, 440
 o-dispatcher, 442
 o-finish, 446
 o-restore-state, 444
 trace-error-handler-path1, 458
 trace-error-handler-path2, 460
 trace-error-handler-thru-save-state, 450
 trace-error-handler-to-dispatcher, 452
 trace-error-handler-to-finish, 457
 trace-error-handler-to-restore-state, 455
 trace-error-handler-to-wait-state, 454
 trace-input-interrupt-handler, 882
 trace-os-error-handler, 460
 trace-output-interrupt-handler, 883
 trace-running-input-handler, 673
 trace-running-input-handler-path1, 647
 h2, 648
 h3, 650
 h4, 668

- h5, 670
- h6, 671
- trace-running-input-handler-thru-save-state, 628
- trace-running-input-handler-to-lpsw1, 641
- lpsw2, 643
- lpsw3, 645
- lpsw4, 663
- lpsw5, 665
- lpsw6, 666
- resume-task1, 633
- resume-task2, 636
- resume-task3, 638
- resume-task4, 655
- resume-task5, 658
- resume-task6, 661
- trace-running-input-handler-to-i-buffer-full-test, 630
- buffer-full-test2, 652
- trace-running-output-handler, 736
- trace-running-output-handler-path1, 730
- h2, 732
- h3, 733
- h4, 735
- trace-running-output-handler-thru-save-state, 706
- trace-running-output-handler-to-lpsw1, 716
- lpsw2, 718
- lpsw3, 727
- lpsw4, 729
- obuffer-empty-test, 709
- obuffer-empty-test2, 720
- resume-task1, 711
- resume-task2, 714
- resume-task3, 722
- resume-task4, 725
- trace-svc-interrupt-handler, 882
- trace-svc-recv-handler, 532
- trace-svc-recv-handler-path1, 522
- trace-svc-recv-handler-path2, 524
- trace-svc-recv-handler-path3, 526
- trace-svc-recv-handler-path4, 531
 - a, 528
 - b, 531
- trace-svc-recv-handler-thru-delivery, 509
- save-state, 498
- trace-svc-recv-handler-to-dispatcher, 503
- trace-svc-recv-handler-to-lpsw, 507
- trace-svc-recv-handler-to-qempty-test, 500
- trace-svc-recv-handler-to-svc-c-resume-task1, 512
- c-resume-task2, 514
- c-resume-task3, 517
- cr1, 519
- cr2, 521
- cr3, 521
- trace-svc-recv-handler-to-wait-state, 505
- trace-svc-send-handler, 498
- trace-svc-send-handler-path1, 488
- trace-svc-send-handler-path2, 489
- trace-svc-send-handler-path3, 492
- trace-svc-send-handler-path4, 496
- trace-svc-send-handler-path4a, 494
- trace-svc-send-handler-path4b, 496
- trace-svc-send-handler-thru-enqueue, 475
- trace-svc-send-handler-thru-save-state, 463
- trace-svc-send-handler-to-dispatcher, 467
- trace-svc-send-handler-to-lpsw, 472
- trace-svc-send-handler-to-qfullp-test, 465
- trace-svc-send-handler-to-svc-resume-task1, 477
- sume-task2, 480
- sume-task3, 482
- trace-svc-send-handler-to-svcr1, 484
- trace-svc-send-handler-to-svcr2, 486
- trace-svc-send-handler-to-svcr3, 487

trace-svc-send-handler-to-wait-state, 470
 trace-svc-tyi-handler, 553
 trace-svc-tyi-handler-path1, 549
 trace-svc-tyi-handler-path2, 550
 trace-svc-tyi-handler-path3, 552
 trace-svc-tyi-handler-thru-delivery, 546
 trace-svc-tyi-handler-thru-save-state, 535
 trace-svc-tyi-handler-to-dispatcher, 539
 trace-svc-tyi-handler-to-lpsw, 543
 trace-svc-tyi-handler-to-qempty-test, 537
 trace-svc-tyi-handler-to-svcr, 548
 trace-svc-tyi-handler-to-wait-state, 541
 trace-svc-tyo-handler, 577
 trace-svc-tyo-handler-path1, 572
 trace-svc-tyo-handler-path2, 573
 trace-svc-tyo-handler-path3, 575
 trace-svc-tyo-handler-path4, 576
 trace-svc-tyo-handler-thru-enqueue, 566
 trace-svc-tyo-handler-thru-save-state, 555
 trace-svc-tyo-handler-to-dispatcher, 560
 trace-svc-tyo-handler-to-lpsw, 564
 trace-svc-tyo-handler-to-qfullp-test, 557
 trace-svc-tyo-handler-to-svcr1, 568
 trace-svc-tyo-handler-to-svcr2, 570
 trace-svc-tyo-handler-to-wait-state, 562
 trace-waiting-input-handler, 625
 trace-waiting-input-handler-path1, 600
 trace-waiting-input-handler-path2, 601
 trace-waiting-input-handler-path3, 603
 trace-waiting-input-handler-path4, 621
 trace-waiting-input-handler-path5, 623
 trace-waiting-input-handler-path6, 624
 trace-waiting-input-handler-thru-readyq-empty-test, 580
 trace-waiting-input-handler-to-dispatcher1, 585
 trace-waiting-input-handler-to-dispatcher2, 588
 trace-waiting-input-handler-to-dispatcher3, 590
 trace-waiting-input-handler-to-dispatcher4, 608
 trace-waiting-input-handler-to-dispatcher5, 611
 trace-waiting-input-handler-to-dispatcher6, 613
 trace-waiting-input-handler-to-lpsw1, 593
 trace-waiting-input-handler-to-lpsw2, 596
 trace-waiting-input-handler-to-lpsw3, 598
 trace-waiting-input-handler-to-wait1, 616
 trace-waiting-input-handler-to-wait2, 618
 trace-waiting-input-handler-to-wait3, 620
 trace-waiting-input-handler-to-i-buffer-full-test, 582
 trace-waiting-input-handler-to-i-buffer-full-test2, 605
 trace-waiting-output-handler, 704
 trace-waiting-output-handler-path1, 699
 trace-waiting-output-handler-path2, 700
 trace-waiting-output-handler-path3, 702
 trace-waiting-output-handler-path4, 703
 trace-waiting-output-handler-thru-readyq-empty-test, 675
 trace-waiting-output-handler-to-dispatcher1, 680
 trace-waiting-output-handler-to-dispatcher2, 682
 trace-waiting-output-handler-to-dispatcher3, 691
 trace-waiting-output-handler-to-dispatcher4, 694
 trace-waiting-output-handler-to-lpsw1, 685
 trace-waiting-output-handler-to-lpsw2, 687
 trace-waiting-output-handler-to-obuffer-empty-test, 677
 trace-waiting-output-handler-to-obuffer-empty-test2, 689
 trace-waiting-output-handler-to-wait1, 695
 trace-waiting-output-handler-to-wait2, 697
 tracing-lemmas, 395
 transitivity-of-permutation, 27
 transitivity-of-permutation-base-case, 27
 transitivity-of-permutation-induction, 27
 transitivity-of-permutation-induction-step, 27

translate-dc, 319, 325
translate-dc-arg, 319
translate-dc-arglist, 319
translate-instruction, 325
translate-source, 325, 329

yet-another-difference-plus-cro
ck, 15

zerop-remainder-difference, 17