```
#|
 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
$(y \not\in \mathbf{N}) \rightarrow ((x + y) = \mathrm{fix}\,(x))$

EVENT: Disable plus-right-id2.

THEOREM: plus-add1
$(x + (1 + y))$
$=$ **if** $y \in \mathbf{N}$ **then** $1 + (x + y)$
else $1 + x$ **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
$((a + b) = \texttt{'0}) = ((a \simeq \texttt{0}) \wedge (b \simeq \texttt{0}))$

EVENT: Disable plus-equal-0.

THEOREM: difference-x-x
$(x - x) = \texttt{'0}$

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
$(y \not< x) \rightarrow ((x - y) = \texttt{'0})$

EVENT: Disable difference-0.

THEOREM: equal-difference-0
$(\texttt{'0} = (x - y)) = (y \not< x)$

EVENT: Disable equal-difference-0.

THEOREM: difference-cancellation-0
$(x = (x - y)) = ((x \in \mathbf{N}) \wedge ((x = \texttt{'0}) \vee (y \simeq \texttt{0})))$

EVENT: Disable difference-cancellation-0.

THEOREM: difference-cancellation-1
$((x - y) = (z - y))$
$=$ **if** $x < y$ **then** $y \not< z$
elseif $z < y$ **then** $y \not< x$
else $\text{fix}(x) = \text{fix}(z)$ **endif**

EVENT: Disable difference-cancellation-1.

THEOREM: times-zero2
$(y \notin \mathbf{N}) \rightarrow ((x * y) = \texttt{'0})$

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
$(x * (1 + y))$
$=$ **if** $y \in \mathbf{N}$ **then** $x + (x * y)$
else $\mathrm{fix}\,(x)$ **endif**

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

EVENT: Disable equal-times-0.

DEFINITION:
$\exp(i, j)$
$=$ **if** $j \simeq \texttt{0}$ **then** $\texttt{'1}$
else $i * \exp(i, j - 1)$ **endif**

THEOREM: exp-plus
$\exp(i, j + k) = (\exp(i, j) * \exp(i, k))$

EVENT: Disable exp-plus.

THEOREM: equal-lessp
$((x < y) = z)$
$=$ **if** $x < y$ **then** `'*1*true` $= z$
else `'*1*false` $= z$ **endif**

EVENT: Disable equal-lessp.

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

THEOREM: remainder-quotient
$((x \textbf{ mod } y) + (y * (x \div y))) = \text{fix}\,(x)$

EVENT: Disable remainder-quotient.

THEOREM: remainder-wrt-1
$(y \textbf{ mod } \texttt{'1}) = \texttt{'0}$

EVENT: Disable remainder-wrt-1.

THEOREM: remainder-wrt-12
$(x \notin \mathbf{N}) \rightarrow ((y \textbf{ mod } x) = \text{fix}\,(y))$

EVENT: Disable remainder-wrt-12.

THEOREM: lessp-remainder2
$((x \textbf{ mod } y) < y) = (y \not\simeq \texttt{0})$

EVENT: Disable lessp-remainder2.

THEOREM: remainder-x-x
$(x \textbf{ mod } x) = \texttt{'0}$

EVENT: Disable remainder-x-x.

THEOREM: remainder-quotient-elim
$((y \not\simeq \texttt{0}) \wedge (x \in \mathbf{N})) \rightarrow (((x \textbf{ mod } y) + (y * (x \div y))) = x)$

THEOREM: lessp-times-1
$(i \not\simeq \texttt{0}) \rightarrow ((i * j) \not< j)$

EVENT: Disable lessp-times-1.

THEOREM: lessp-times-2
$(i \not\simeq \mathtt{0}) \rightarrow ((j * i) \not< j)$

EVENT: Disable lessp-times-2.

THEOREM: lessp-quotient1
$((i \div j) < i) = ((i \not\simeq \mathtt{0}) \wedge ((j \simeq \mathtt{0}) \vee (j \neq \mathtt{'1})))$

EVENT: Disable lessp-quotient1.

THEOREM: lessp-remainder1
$((x \textbf{ mod } y) < x) = ((y \not\simeq \mathtt{0}) \wedge ((x \not\simeq \mathtt{0}) \wedge (x \not< y)))$

EVENT: Disable lessp-remainder1.

THEOREM: difference-plus1
$((x + y) - x) = \mathrm{fix}(y)$

EVENT: Disable difference-plus1.

THEOREM: difference-plus2
$((y + x) - x) = \mathrm{fix}(y)$

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: $\mathrm{divides}(x, y) = ((y \textbf{ mod } x) \simeq \mathtt{0})$

THEOREM: divides-times
$((x * z) \textbf{ mod } z) = \mathtt{'0}$

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
$((y \not\simeq \texttt{0}) \wedge (y \neq \texttt{'1})) \rightarrow (((1 + (x * y)) \textbf{ mod } y) \neq \texttt{'0})$

EVENT: Disable remainder-add1.

THEOREM: divides-plus-rewrite1
$(((x \textbf{ mod } z) = \texttt{'0}) \wedge ((y \textbf{ mod } z) = \texttt{'0}))$
$\rightarrow \quad (((x + y) \textbf{ mod } z) = \texttt{'0})$

EVENT: Disable divides-plus-rewrite1.

THEOREM: divides-plus-rewrite2
$(((x \textbf{ mod } z) = \texttt{'0}) \wedge ((y \textbf{ mod } z) \neq \texttt{'0}))$
$\rightarrow \quad (((x + y) \textbf{ mod } z) \neq \texttt{'0})$

EVENT: Disable divides-plus-rewrite2.

THEOREM: divides-plus-rewrite
$((x \textbf{ mod } z) = \texttt{'0})$
$\rightarrow \quad ((((x + y) \textbf{ mod } z) = \texttt{'0}) = ((y \textbf{ mod } z) = \texttt{'0}))$

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
$((x \textbf{ mod } z) = \texttt{'0})$
$\rightarrow \quad ((((y + x) \textbf{ mod } z) = \texttt{'0}) = ((y \textbf{ mod } z) = \texttt{'0}))$

EVENT: Disable divides-plus-rewrite-commuted.

THEOREM: euclid
$((x \textbf{ mod } z) = \texttt{'0})$
$\rightarrow \quad ((((y - x) \textbf{ mod } z) = \texttt{'0})$
$\quad = \quad \textbf{if } x < y \textbf{ then } (y \textbf{ mod } z) = \texttt{'0}$
$\quad\quad \textbf{else } \texttt{'*1*true } \textbf{endif})$

EVENT: Disable euclid.

THEOREM: lessp-times-cancellation
$((x * z) < (y * z)) = ((z \not\simeq \texttt{0}) \wedge (x < y))$

EVENT: Disable lessp-times-cancellation.

THEOREM: lessp-plus-cancellation3
$(y < (x + y)) = (x \not\simeq \texttt{0})$

EVENT: Disable lessp-plus-cancellation3.

THEOREM: quotient-times1
$((y \in \mathbf{N}) \wedge ((x \in \mathbf{N}) \wedge ((x \neq \texttt{'0}) \wedge \text{divides}\,(x,\, y))))$
$\rightarrow \quad ((x * (y \div x)) = y)$

EVENT: Disable quotient-times1.

THEOREM: quotient-lessp
$((x \not\simeq \texttt{0}) \wedge (x < y)) \rightarrow ((y \div x) \neq \texttt{'0})$

EVENT: Disable quotient-lessp.

THEOREM: times-id-iff-1
$(z = (w * z)) = ((z \in \mathbf{N}) \wedge ((z = \texttt{'0}) \vee (w = \texttt{'1})))$

EVENT: Disable times-id-iff-1.

THEOREM: divides-times1
$(a = (z * y)) \rightarrow ((a \textbf{ mod } z) = \texttt{'0})$

EVENT: Disable divides-times1.

THEOREM: times-identity1
$((y \in \mathbf{N}) \wedge ((y \neq \texttt{'1}) \wedge ((y \neq \texttt{'0}) \wedge (x \neq \texttt{'0}))))$
$\rightarrow \quad (x \neq (x * y))$

EVENT: Disable times-identity1.

THEOREM: times-identity
$(x = (x * y)) = ((x = \texttt{'0}) \vee ((x \in \mathbf{N}) \wedge (y = \texttt{'1})))$

EVENT: Disable times-identity.

THEOREM: quotient-divides
$((y \in \mathbf{N}) \wedge ((x * (y \div x)) \neq y)) \rightarrow ((y \textbf{ mod } x) \neq \texttt{'0})$

EVENT: Disable quotient-divides.

THEOREM: remainder-times
$((y * x) \textbf{ mod } y) = \texttt{'0}$

EVENT: Disable remainder-times.

THEOREM: quotient-times
$((y * x) \div y)$
$= \quad \textbf{if } y \simeq \texttt{0} \textbf{ then } \texttt{'0}$
$\quad\quad \textbf{else } \mathrm{fix}\,(x) \textbf{ endif}$

EVENT: Disable quotient-times.

THEOREM: distributivity-of-divides
$((a \not\simeq \texttt{0}) \wedge \mathrm{divides}\,(a,\, w)) \rightarrow ((c * (w \div a)) = ((c * w) \div a))$

EVENT: Disable distributivity-of-divides.

THEOREM: if-times-then-divides
$((c \not\simeq \texttt{0}) \wedge (\neg\, \mathrm{divides}\,(c,\, x))) \rightarrow ((c * y) \neq x)$

EVENT: Disable if-times-then-divides.

THEOREM: times-equal-1
$((a * b) = \texttt{'1})$
$= \quad ((a \neq \texttt{'0})$
$\quad\quad \wedge \quad ((b \neq \texttt{'0})$

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

EVENT: Disable times-equal-1.

THEOREM: divides-implies-times
$((a \not\simeq \texttt{0}) \wedge ((c \in \mathbf{N}) \wedge ((a * c) = b)))$
$\rightarrow \quad ((c = (b \div a)) = \texttt{'*1*true})$

EVENT: Disable divides-implies-times.

THEOREM: difference-1
$(x - \texttt{'1}) = (x - 1)$

EVENT: Disable difference-1.

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

EVENT: Disable difference-2.

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

EVENT: Disable half-plus.

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

EVENT: Disable times-1.

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

EVENT: Disable exp-of-0.

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

EVENT: Disable exp-of-1.

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

EVENT: Disable exp-by-0.

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

EVENT: Disable exp-times.

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

EVENT: Disable exp-exp.

THEOREM: remainder-plus-times-1
$((x + (i * j)) \textbf{ mod } j) = (x \textbf{ mod } j)$

EVENT: Disable remainder-plus-times-1.

THEOREM: remainder-plus-times-2
$((x + (j * i)) \textbf{ mod } j) = (x \textbf{ mod } j)$

EVENT: Disable remainder-plus-times-2.

THEOREM: remainder-times-1
$((b * (a * c)) \textbf{ mod } a) = \texttt{'0}$

EVENT: Disable remainder-times-1.

THEOREM: remainder-of-1
$(\texttt{'1} \textbf{ mod } x)$
$=$ **if** $x = \texttt{'1}$ **then** `'0`
else `'1` **endif**

EVENT: Disable remainder-of-1.

THEOREM: remainder-difference-times
$(((p * x) - (p * y)) \textbf{ mod } p) = \texttt{'0}$

EVENT: Disable remainder-difference-times.

THEOREM: lessp-remainder-divisor
$(y \not\simeq \mathtt{0}) \rightarrow ((x \textbf{ mod } y) < y)$

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-cancelation, divides-plus-rewrite, divides-plus-rewrite2, divides-plus-rewrite1, remainder-add1, difference-add1-cancellation, difference-plus3, divides-times, divides, times-difference, difference-plus-cancelation, 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** $n \simeq \mathtt{0}$ **then** `'0`
elseif listp (l) **then** number-and-list-induction $(n - 1,$ cdr $(l))$
else `'0` **endif**

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

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

$j - 1$,
cdr ($l1$),
cdr ($l2$)) **endif**

else '0 **endif**

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

EVENT: Disable plus-0.

THEOREM: plus-1
$(\texttt{'1} + x) = (1 + x)$

EVENT: Disable plus-1.

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

EVENT: Disable plus-0-arg1.

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

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

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
$(b \not\simeq \texttt{0}) \rightarrow ((a + ((b - 1) * a)) = (b * a))$

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
if $a < b$ **then** `'*1*false`
else `'*1*true` **endif**
$\rightarrow \quad (((1 + a) - b) = (1 + (a - b)))$

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
$((b \not\simeq 0)$
$\wedge$ **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
$((a \in \mathbf{N}) \wedge (a < b)) \rightarrow ((b - a) \neq \texttt{'0})$

EVENT: Disable non-zero-difference.

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

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

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

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**$))$
$\rightarrow$ $(((a - b) = \texttt{'0}) = (a = b))$

EVENT: Disable difference-equals-0.

THEOREM: times-zero3
$(x * \texttt{'0}) = \texttt{'0}$

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
$('0 < \exp('2, i)) = \text{'*1*true}$

EVENT: Disable exp-2-never-0.

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

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

THEOREM: remainder-noop
$((a \in \mathbf{N}) \wedge ((a < b) \wedge (b \not\simeq 0))) \rightarrow ((a \bmod b) = a)$

EVENT: Disable remainder-noop.

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

EVENT: Disable remainder-add1-casesplit.

THEOREM: zerop-remainder-difference
$((a \bmod b) = '0) \rightarrow (((a - b) \bmod b) = '0)$

EVENT: Disable zerop-remainder-difference.

THEOREM: quotient-x-x
$(x \not\simeq 0) \rightarrow ((x \div x) = '1)$

EVENT: Disable quotient-x-x.

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

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
$(i \not< j) \rightarrow (((i - j) \div j) = ((i \div j) - 1))$

EVENT: Disable quotient-difference.

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

EVENT: Disable quotient-plus.

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

EVENT: Disable quotient-plus-times.

THEOREM: leq-times
$(($**if** $a \not\simeq$ `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**
$\rightarrow$ $((c < a) =$ `'*1*false`$)$

THEOREM: leq-casesplit
$((a \in \mathbf{N}) \wedge (b \in \mathbf{N}))$
$\rightarrow$ (**if** $b < a$ **then** `'*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
$((c \not\simeq$ `0`$) \wedge ((b$ **mod** $c) \not\simeq$ `0`$))$
$\rightarrow$ $(((a$ **mod** $(b$ **mod** $c)) < c) =$ `'*1*true`$)$

EVENT: Disable lessp-remainder-remainder.

THEOREM: lessp-times
$((a \not\simeq$ `0`$) \wedge (b < c)) \rightarrow ((a * b) < (a * c))$

EVENT: Disable lessp-times.

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

THEOREM: lessp-difference
$((a \not\simeq$ `0`$) \wedge (a < b)) \rightarrow ((b - a) < b)$

THEOREM: remainder-difference-non-zero
$((a \not\simeq$ `0`$) \wedge ((b \in \mathbf{N}) \wedge (a < b))) \rightarrow (((b - a)$ **mod** $b) \not\simeq$ `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
$(((a \textbf{ mod } b) = \texttt{'0}) \wedge (a \not\simeq \texttt{0})) \rightarrow ((a < b) = \texttt{'*1*false})$

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

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

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

EVENT: Disable lessp-quotient-remainder-crock.

THEOREM: lessp-quotient-remainder
$((b \not\simeq \texttt{0}) \wedge (c \not\simeq \texttt{0})) \rightarrow (((a \textbf{ mod } (b * c)) \div c) < \text{fix}\,(b))$

EVENT: Disable lessp-quotient-remainder.

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

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
$((a + n) \textbf{ mod } a) = (n \textbf{ mod } a)$

EVENT: Disable remainder-plus.

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

EVENT: Disable remainder-divides.

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

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
$((n \not\simeq \texttt{0}) \wedge (((b\ \mathbf{mod}\ n) = \texttt{'0}) \wedge (a < b)))$
$\rightarrow\ (((a \div n) < (b \div n)) = \texttt{'*1*true})$

EVENT: Disable lessp-quotient-quotient.

THEOREM: lessp-times-quotient
$((n \not\simeq \texttt{0}) \wedge ((i \in \mathbf{N}) \wedge ((i < j) \wedge ((j\ \mathbf{mod}\ n) = \texttt{'0}))))$
$\rightarrow\ (((n * (i \div n)) < j) = \texttt{'*1*true})$

EVENT: Disable lessp-times-quotient.

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

THEOREM: not-lessp-times-plus-instance
$$\begin{array}{l}((x \in \mathbf{N}) \\ \wedge \quad ((v \in \mathbf{N}) \\ \qquad \wedge \quad ((v < n) \\ \qquad\quad \wedge \quad ((z \in \mathbf{N}) \\ \qquad\qquad \wedge \quad ((n \neq \texttt{'0}) \\ \qquad\qquad\quad \wedge \quad ((n \in \mathbf{N}) \\ \qquad\qquad\qquad \wedge \quad ((n * z) \\ \qquad\qquad\qquad\quad < \quad (n + (n * x)))))))))) \\ \rightarrow \quad (((v + (n * x)) < (n * z)) = \texttt{'*1*false})\end{array}$$

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

THEOREM: not-lessp-plus-times-quotient
$$((n \not\simeq \texttt{0}) \wedge ((i \in \mathbf{N}) \wedge ((i < j) \wedge ((j \ \mathbf{mod}\ n) = \texttt{'0})))) \\ \rightarrow \quad ((j < ((n * (i \div n)) + n)) = \texttt{'*1*false})$$

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

THEOREM: quotient-plus-times-remainder
$$(n \not\simeq \texttt{0}) \rightarrow ((((n * a) + (b \ \mathbf{mod}\ n)) \div n) = \mathrm{fix}\,(a))$$

EVENT: Disable quotient-plus-times-remainder.

THEOREM: remainder-plus-times-remainder
$$(((n * a) + (b \ \mathbf{mod}\ n)) \ \mathbf{mod}\ n) = (b \ \mathbf{mod}\ n)$$

EVENT: Disable remainder-plus-times-remainder.

THEOREM: remainder-non-zero
$$((n \not\simeq \texttt{0}) \wedge (n < m)) \rightarrow (\texttt{'0} < (n \ \mathbf{mod}\ m))$$

THEOREM: plus-not-equal
$$(\mathrm{fix}\,(b) \neq \mathrm{fix}\,(c)) \rightarrow ((a + b) \neq (a + c))$$

EVENT: Disable plus-not-equal.

THEOREM: difference-plus4
$$((a + (b + c)) - (b + c)) = \mathrm{fix}\,(a)$$

EVENT: Disable difference-plus4.

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

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

DEFINITION:
length (l)
$=$ **if** listp (l) **then** $1 +$ length (cdr (l))
else `'0` **endif**

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

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

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

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

DEFINITION:
plistp (l)
$=$ **if** listp (l) **then** plistp (cdr (l))
else $l =$ `'nil` **endif**

DEFINITION:
setp (l)
$=$ **if** listp (l) **then** $(\text{car}\ (l) \notin \text{cdr}\ (l)) \wedge$ setp (cdr (l))
else $l =$ `'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
$(\neg$ listp $(l)) \rightarrow (\neg$ listp (remove $(x,$ cons $(x,\ l))))$

EVENT: Disable not-listp-remove.

THEOREM: length-insert
length (insert $(i,\ l)) = (1 +$ length $(l))$

EVENT: Disable length-insert.

THEOREM: length-remove
length (remove $(x,\ l)$)
$=$ **if** $x \in l$ **then** length $(l) - 1$
else length (l) **endif**

EVENT: Disable length-remove.

THEOREM: length-permutation
permutation $(a,\ b) \rightarrow$ (length $(a) =$ length $(b))$

THEOREM: number-of-occurrences-of-set-member-is-1
(setp $(l) \wedge (x \in l)) \rightarrow$ (occurrences $(x,\ l) =$ `'1`)

THEOREM: plistp-remove
plistp (remove $(x,\ l)) =$ plistp (l)

EVENT: Disable plistp-remove.

THEOREM: plistp-set
setp $(l) \rightarrow$ plistp (l)

EVENT: Disable plistp-set.

THEOREM: insert-remove
(sorted $(l) \wedge (x \neq y)$)
$\rightarrow$ (insert $(x,$ remove $(y,\ l)) =$ remove $(y,$ 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
sorted $(l) \rightarrow$ sorted (remove $(x,\ l)$)

EVENT: Disable sorted-remove.

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

EVENT: Disable not-permutation.

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

EVENT: Disable permutation-right-cons1.

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

EVENT: Disable permutation-right-cons.

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

EVENT: Disable commutativity-of-permutation.

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

THEOREM: permutation-reflexivity
permutation $(l,\ l)$

EVENT: Disable permutation-reflexivity.

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

EVENT: Disable permutation-append-car.

THEOREM: permutation-append-insert
(permutation (a, b) ∧ (sorted (b) ∧ ((x ∉ a) ∧ (x ∉ b))))
→ permutation (append (a, cons (x, 'nil)), insert (x, b))

EVENT: Disable permutation-append-insert.

THEOREM: permutation-remove-remove
(permutation (a, b) ∧ ((x ∈ a) ∧ (x ∈ b)))
→ 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 ≃ **nil**) ∧ (permutation (a, b) ∧ permutation (b, c))) → permutation (a, c)

THEOREM: transitivity-of-permutation-induction-step
(listp (a)
∧ (((permutation (cdr (a), remove (car (a), b))
∧ permutation (remove (car (a), b), remove (car (a), c)))
→ permutation (cdr (a), remove (car (a), c)))
∧ (permutation (a, b) ∧ permutation (b, c))))
→ permutation (a, c)

THEOREM: transitivity-of-permutation
(permutation (a, b) ∧ permutation (b, c)) → permutation (a, c)

THEOREM: setp-remove
setp (a) → 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
$$
\begin{array}{l}
(\text{listp}\,(a) \\
\quad \wedge \quad (\text{setp}\,(b) \\
\qquad \wedge \quad (\text{plistp}\,(a) \\
\qquad\quad \wedge \quad (\text{permutation}\,(a,\ b) \\
\qquad\qquad \wedge \quad ((\text{setp}\,(\text{remove}\,(\text{car}\,(a),\ b)) \\
\qquad\qquad\quad \wedge \quad (\text{plistp}\,(\text{cdr}\,(a)) \\
\qquad\qquad\qquad \wedge \quad \text{permutation}\,(\text{cdr}\,(a), \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{remove}\,(\text{car}\,(a),\ b)))) \\
\qquad\qquad\quad \rightarrow \quad \text{setp}\,(\text{cdr}\,(a)))))))) \\
\rightarrow \quad \text{setp}\,(a)
\end{array}
$$

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 $n \simeq$ `0` **then** car (l)
else getnth $(n - 1,\ \text{cdr}\,(l))$ **endif**
else `'0` **endif**

DEFINITION:
putnth $(v,\ n,\ l)$
= **if** listp (l)
then if $n \simeq$ `0` **then** cons $(v,\ \text{cdr}\,(l))$
else cons $(\text{car}\,(l),\ \text{putnth}\,(v,\ n - 1,\ \text{cdr}\,(l)))$ **endif**
else l **endif**

DEFINITION:
getseg $(n,\ k,\ l)$
= **if** $k \simeq$ `0` **then** `'nil`
else cons $(\text{getnth}\,(n,\ l),\ \text{getseg}\,(1 + n,\ k - 1,\ l))$ **endif**

DEFINITION:
putseg $(s,\ n,\ l)$
= **if** listp (s) **then** putnth $(\text{car}\,(s),\ n,\ \text{putseg}\,(\text{cdr}\,(s),\ 1 + n,\ l))$
else l **endif**

DEFINITION:
fixlength $(n,\ l,\ \mathit{default})$
$=$ **if** listp (l)
then if $n \simeq$ `0` **then** `'nil`
else cons $(\text{car}\,(l),\ \text{fixlength}\,(n-1,\ \text{cdr}\,(l),\ \mathit{default}))$ **endif**
elseif $n \simeq$ `0` **then** `'nil`
else cons $(\mathit{default},\ \text{fixlength}\,(n-1,\ l,\ \mathit{default}))$ **endif**

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

DEFINITION:
nthcdr $(n,\ l)$
$=$ **if** listp (l)
then if $n \simeq$ `0` **then** l
else nthcdr $(n-1,\ \text{cdr}\,(l))$ **endif**
else l **endif**

THEOREM: non-zero-length-implies-listp
$(\text{length}\,(l) \not\simeq$ `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
$(\text{plistp}\,(a) \wedge (\text{length}\,(a) \neq$ `'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
$(k \not\simeq \mathtt{0}) \rightarrow \mathrm{listp}\,(\mathrm{getseg}\,(n,\ k,\ l))$

EVENT: Disable listp-getseg.

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

EVENT: Disable listp-putnth.

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

EVENT: Disable listp-putseg.

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

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

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

EVENT: Disable car-append-casesplit.

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

EVENT: Disable car-append.

THEOREM: car-getseg
$(k \not\simeq \mathtt{0}) \rightarrow (\mathrm{car}\,(\mathrm{getseg}\,(n,\ k,\ l)) = \mathrm{getnth}\,(n,\ l))$

EVENT: Disable car-getseg.

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

EVENT: Disable car-putnth-zero.

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

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
$(k \not\simeq \texttt{0}) \rightarrow (\text{cdr}\,(\text{getseg}\,(n,\ k,\ l)) = \text{getseg}\,(1 + n,\ k - 1,\ l))$

EVENT: Disable cdr-getseg.

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

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

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

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
$\text{plistp}\,(\text{cons}\,(x,\ \texttt{'nil}))$

EVENT: Disable plistp-list1.

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

EVENT: Disable plistp-list2.

THEOREM: plistp-putnth
plistp (*l*) → plistp (putnth (*v*, *i*, *l*))

EVENT: Disable plistp-putnth.

THEOREM: plistp-getseg
plistp (getseg (*n*, *k*, *l*))

EVENT: Disable plistp-getseg.

THEOREM: plistp-putseg
plistp (*l*) → plistp (putseg (*s*, *n*, *l*))

EVENT: Disable plistp-putseg.

THEOREM: plistp-fixlength
plistp (fixlength (*n*, *l*, *v*))

EVENT: Disable plistp-fixlength.

THEOREM: length-append
length (append (*a*, *b*)) = (length (*a*) + length (*b*))

EVENT: Disable length-append.

THEOREM: length-putnth
length (putnth (*n*, *v*, *l*)) = length (*l*)

EVENT: Disable length-putnth.

THEOREM: length-getseg
length (getseg (*n*, *k*, *l*)) = fix (*k*)

EVENT: Disable length-getseg.

THEOREM: length-putseg
length (putseg (*s*, *n*, *l*)) = length (*l*)

EVENT: Disable length-putseg.

THEOREM: length-fixlength
length (fixlength (*n*, *l*, *default*)) = fix (*n*)

EVENT: Disable length-fixlength.

THEOREM: append-non-list-left
(¬ listp (*a*)) → (append (*a*, *b*) = *b*)

EVENT: Disable append-non-list-left.

THEOREM: append-non-list-left-corollary
(length (*a*) = '0) → (append (*a*, *b*) = *b*)

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

THEOREM: append-nil-on-right
plistp (*a*) → (append (*a*, 'nil) = *a*)

EVENT: Disable append-nil-on-right.

THEOREM: append-cdr
listp (*a*) → (append (cdr (*a*), *b*) = cdr (append (*a*, *b*)))

EVENT: Disable append-cdr.

THEOREM: append-crock
append (cons (*a*, 'nil), append (*b*, *c*)) = append (cons (*a*, *b*), *c*)

EVENT: Disable append-crock.

THEOREM: associativity-of-append
append (append (*a*, *b*), *c*) = append (*a*, append (*b*, *c*))

EVENT: Disable associativity-of-append.

THEOREM: append-fixlength
append (fixlength (*a*, 'nil, *default*), fixlength (*b*, 'nil, *default*))
= fixlength (*a* + *b*, 'nil, *default*)

EVENT: Disable append-fixlength.

THEOREM: getnth-0-2
getnth ('0, cons (*a*, cons (*b*, 'nil))) = *a*

EVENT: Disable getnth-0-2.

THEOREM: getnth-1-2
getnth ('1, cons (a, cons (b, 'nil))) = b

EVENT: Disable getnth-1-2.

THEOREM: list-getnth-0
(plistp (l) ∧ (length (l) = '1)) → (cons (getnth ('0, l), 'nil) = l)

EVENT: Disable list-getnth-0.

THEOREM: open-up-getnth
getnth (1 + n, cons (a, b)) = getnth (n, b)

EVENT: Disable open-up-getnth.

THEOREM: open-up-getnth1
($n \not\simeq$ 0) → (getnth (n, cons (a, b)) = getnth (n − 1, b))

EVENT: Disable open-up-getnth1.

THEOREM: open-up-getnth-on-zero
getnth ('0, cons (a, b)) = a

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

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

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

THEOREM: getnth-append-for-length-left
getnth (length (a), append (a, b))
= **if** listp (b) **then** car (b)
else '0 **endif**

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

THEOREM: getnth-append-casesplit
getnth (n, append (a, b))
= **if** $n <$ length (a) **then** getnth (n, a)
else getnth (n − length (a), b) **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
$((k \not\simeq \texttt{0}) \wedge (i \simeq \texttt{0})) \rightarrow (\text{getnth}(i, \text{getseg}(n, k, l)) = \text{getnth}(n, l))$

EVENT: Disable getnth-getseg-zero-index.

DEFINITION:
$\text{getnth-getseg-induction}(i, n, k)$
$=$ **if** $(i \simeq \texttt{0}) \vee (k \simeq \texttt{0})$ **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** $n \simeq \texttt{0}$ **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
$(n \not\simeq \texttt{0}) \rightarrow (\text{putnth}\,(v,\ n,\ \text{cons}\,(a,\ b)) = \text{cons}\,(a,\ \text{putnth}\,(v,\ n - 1,\ b)))$

EVENT: Disable open-up-putnth1.

THEOREM: putnth-into-nil
putnth (*v*, *i*, 'nil) = 'nil

EVENT: Disable putnth-into-nil.

THEOREM: open-up-putnth-on-zero
putnth (*val*, '0, cons (*a*, *b*)) = cons (*val*, *b*)

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

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

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

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

EVENT: Disable putnth-has-no-effect.

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

EVENT: Disable putnth-cons1.

THEOREM: putnth-cons2
$(n \not\simeq 0) \rightarrow$ (putnth (*v*, *n*, cons (*a*, *b*)) = cons (*a*, putnth (*v*, *n* − 1, *b*)))

EVENT: Disable putnth-cons2.

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

EVENT: Disable putnth-shift.

THEOREM: putnth-append-casesplit
putnth (*v*, *n*, append (*a*, *b*))
= **if** $n <$ length (*a*) **then** append (putnth (*v*, *n*, *a*), *b*)
else append (*a*, putnth (*v*, *n* − length (*a*), *b*)) **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
$((i \in \mathbf{N})$
$\wedge$ $((j \in \mathbf{N})$
$\wedge$ **if** $j < i$ **then** `'*1*false`
else `'*1*true` **endif**$))$
$\rightarrow$ $(\text{putnth}(b,\ j,\ \text{putnth}(a,\ i,\ l))$
$=$ **if** $i = j$ **then** $\text{putnth}(b,\ j,\ l)$
else $\text{putnth}(a,\ i,\ \text{putnth}(b,\ j,\ l))$ **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
$(\text{fix}\,(i) < (1 + j))$
$\rightarrow \quad (\text{putnth}\,(b,\ 1 + j,\ \text{putnth}\,(a,\ i,\ l)) = \text{putnth}\,(a,\ i,\ \text{putnth}\,(b,\ 1 + j,\ l)))$

EVENT: Disable putnth-non-interference-corollary1.

THEOREM: putnth-non-interference-corollary2
$(\text{fix}\,(i) < (j + k))$
$\rightarrow \quad (\text{putnth}\,(b,\ j + k,\ \text{putnth}\,(a,\ i,\ l)) = \text{putnth}\,(a,\ i,\ \text{putnth}\,(b,\ j + k,\ l)))$

EVENT: Disable putnth-non-interference-corollary2.

THEOREM: putnth-putseg-coincidence-proof
$((n \in \mathbf{N})$
$\wedge \quad ((i \in \mathbf{N})$
$\wedge \quad (\textbf{if}\ i < n\ \textbf{then}\ \texttt{'*1*false}$
$\textbf{else}\ \texttt{'*1*true}\ \textbf{endif}$
$\wedge \quad ((i - n) < \text{length}\,(s)))))$
$\rightarrow \quad (\text{putnth}\,(v,\ i,\ \text{putseg}\,(s,\ n,\ l)) = \text{putseg}\,(\text{putnth}\,(v,\ i - n,\ s),\ n,\ l))$

EVENT: Disable putnth-putseg-coincidence-proof.

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

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

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

EVENT: Disable putnth-putseg-coincidence.

THEOREM: putnth-putseg-coincidence-inverse
$((i < \text{length}\,(s)) \wedge (i \in \mathbf{N}))$
$\rightarrow \quad (\text{putseg}\,(\text{putnth}\,(v,\ i,\ s),\ n,\ l) = \text{putnth}\,(v,\ i + n,\ \text{putseg}\,(s,\ n,\ l)))$

EVENT: Disable putnth-putseg-coincidence-inverse.

THEOREM: putnth-after-putseg-proof

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

EVENT: Disable putnth-after-putseg-proof.

THEOREM: putnth-after-putseg
$(\textbf{if } i < n \textbf{ then } \texttt{'*1*false}$
$\textbf{else } \texttt{'*1*true } \textbf{endif}$
$\wedge \quad \textbf{if } (i - n) < \text{length}\,(s) \textbf{ then } \texttt{'*1*false}$
$\quad \textbf{else } \texttt{'*1*true } \textbf{endif})$
$\rightarrow \quad (\text{putnth}\,(v,\ i,\ \text{putseg}\,(s,\ n,\ l)) = \text{putseg}\,(s,\ n,\ \text{putnth}\,(v,\ i,\ l)))$

EVENT: Disable putnth-after-putseg.

THEOREM: getseg-of-length-zero
$(k \simeq \texttt{0}) \rightarrow (\text{getseg}\,(n,\ k,\ l) = \texttt{'nil})$

EVENT: Disable getseg-of-length-zero.

THEOREM: getseg-add1-length
$\text{getseg}\,(n,\ 1 + k,\ l) = \text{append}\,(\text{getseg}\,(n,\ k,\ l),\ \text{cons}\,(\text{getnth}\,(n + k,\ l),\ \texttt{'nil}))$

EVENT: Disable getseg-add1-length.

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

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

THEOREM: getseg-add1-cons
$\text{getseg}\,(1 + n,\ k,\ \text{cons}\,(a,\ b)) = \text{getseg}\,(n,\ k,\ b)$

EVENT: Disable getseg-add1-cons.

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

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

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**))
$\rightarrow$ $(\text{getseg}\,(n,\ k,\ \text{append}\,(a,\ b))$
$=$ $\text{append}\,(\text{getseg}\,(n,\ \text{length}\,(a) - n,\ a),$
$\text{getseg}\,(\texttt{'0},\ k - (\text{length}\,(a) - n),\ b)))$

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**
$\rightarrow$ (getseg $(i,\ k,$ 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**
$\wedge$ $(((n - i) < k) \wedge (n <$ length $(l))))$
$\rightarrow$ (getseg $(i,\ k,$ putnth $(v,\ n,\ l)) =$ putnth $(v,\ n - i,$ 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**
$\rightarrow$ (getseg $(i,\ j,$ getseg $(n,\ k,\ l)) =$ getseg $(n + i,\ j,\ l)$)

EVENT: Disable getseg-getseg.

THEOREM: getseg-after-putseg
if $i <$ (length $(s) + n)$ **then** `'*1*false`
else `'*1*true` **endif**
$\rightarrow$ (getseg $(i,\ k,$ 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**
$\rightarrow$ (getseg $(i,\ k,$ 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**
∧ (**if** length (*l*) < (*n* + length (*s*)) **then** `'*1*false`
else `'*1*true` **endif**
∧ **if** (*n* + length (*s*)) < (*i* + *k*) **then** `'*1*false`
else `'*1*true` **endif**))
→ (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* ≃ `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* ≮ length (*l*)) → (putseg (*s*, *n*, *l*) = *l*)

EVENT: Disable putseg-has-no-effect.

THEOREM: putseg-cons2
(*n* ≄ `0`) → (putseg (*s*, *n*, cons (*a*, *b*)) = cons (*a*, putseg (*s*, *n* − 1, *b*)))

EVENT: Disable putseg-cons2.

THEOREM: getseg-around-putseg-proof
((*i* ∈ **N**)
∧ ((*n* ∈ **N**)
∧ (**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-proof.

THEOREM: numberp-casesplit
(*a* ∈ **N**) ∨ (*a* ∉ **N**)

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

EVENT: Disable getseg-around-putseg.

THEOREM: putseg-nil-segment
$\text{putseg}(\texttt{'nil}, n, l) = l$

EVENT: Disable putseg-nil-segment.

THEOREM: putseg-non-list-segment
$(\neg \text{listp}(s)) \rightarrow (\text{putseg}(s, n, l) = l)$

EVENT: Disable putseg-non-list-segment.

THEOREM: open-up-putseg-on-segment-cons
($n \in \mathbf{N}$)
$\wedge$ **if** $\text{length}(l) < (n + \text{length}(\text{cons}(a, b)))$ **then** `'*1*false`
else `'*1*true` **endif**)
$\rightarrow$ $(\text{putseg}(\text{cons}(a, b), n, l) = \text{putnth}(a, n, \text{putseg}(b, \texttt{'1} + n, l)))$

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

THEOREM: putseg-cons1
$((n \simeq \texttt{0}) \wedge \text{listp}(s))$
$\rightarrow$ $(\text{putseg}(s, n, \text{cons}(a, b)) = \text{cons}(\text{car}(s), \text{putseg}(\text{cdr}(s), n, b)))$

EVENT: Disable putseg-cons1.

THEOREM: putseg-getseg-coincidence
$(i = n) \rightarrow (\text{putseg}(\text{getseg}(n, k, l), i, l) = l)$

EVENT: Disable putseg-getseg-coincidence.

THEOREM: equality-of-getnth-backchain-proof
(**if** $i < n$ **then** `'*1*false`

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

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** $(a \simeq \texttt{0}) \vee (b \simeq \texttt{0})$ **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**)
$\rightarrow$ $(j \simeq \texttt{0})$

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**)
$\rightarrow$ $(\text{getseg}\,(i,\ j,\ l) = \texttt{'nil})$

EVENT: Disable getseg-must-be-nil.

THEOREM: equality-of-getseg-backchain-base-case
$(((j \simeq \texttt{0}) \vee (k \simeq \texttt{0}))$
$\wedge$ $((i \in \mathbf{N})$
$\wedge$ $((n \in \mathbf{N})$
$\wedge$ $(\text{nested-intervals}\,(n,\ k,\ i,\ j)$
$\wedge$ $(\text{getseg}\,(n,\ k,\ l1) = \text{getseg}\,(n,\ k,\ l2))))))$
$\rightarrow$ $(\text{getseg}\,(i,\ j,\ l1) = \text{getseg}\,(i,\ j,\ l2))$

THEOREM: equality-of-getseg-backchain-induction-step-case1
$((i \in \mathbf{N})$

$$
\begin{array}{ll}
\wedge & ((n \in \mathbf{N}) \\
 & \quad \wedge \quad ((j \not\simeq 0) \\
 & \qquad \wedge \quad ((k \not\simeq 0) \\
 & \qquad\quad \wedge \quad (((\text{nested-intervals}\,(1 + n,\, k - 1,\, 1 + i,\, j - 1) \\
 & \qquad\qquad \wedge \quad (\text{getseg}\,(1 + n,\, k - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + n,\, k - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \rightarrow \quad (\text{getseg}\,(1 + i,\, j - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + i,\, j - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \wedge \quad (\text{nested-intervals}\,(n,\, k,\, i,\, j) \\
 & \qquad\qquad\quad \wedge \quad ((\text{getseg}\,(n,\, k,\, \mathit{l1}) \\
 & \qquad\qquad\qquad\quad = \quad \text{getseg}\,(n,\, k,\, \mathit{l2})) \\
 & \qquad\qquad\qquad\quad \wedge \quad (i = n)))))))) \\
\rightarrow & \quad (\text{getseg}\,(i,\, j,\, \mathit{l1}) = \text{getseg}\,(i,\, j,\, \mathit{l2}))
\end{array}
$$

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

$$
\begin{array}{ll}
 & (i \in \mathbf{N}) \\
\wedge & ((n \in \mathbf{N}) \\
 & \quad \wedge \quad ((j \not\simeq 0) \\
 & \qquad \wedge \quad ((k \not\simeq 0) \\
 & \qquad\quad \wedge \quad (((\text{nested-intervals}\,(1 + n,\, k - 1,\, 1 + i,\, j - 1) \\
 & \qquad\qquad \wedge \quad (\text{getseg}\,(1 + n,\, k - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + n,\, k - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \rightarrow \quad (\text{getseg}\,(1 + i,\, j - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + i,\, j - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \wedge \quad (\text{nested-intervals}\,(n,\, k,\, i,\, j) \\
 & \qquad\qquad\quad \wedge \quad ((\text{getseg}\,(n,\, k,\, \mathit{l1}) \\
 & \qquad\qquad\qquad\quad = \quad \text{getseg}\,(n,\, k,\, \mathit{l2})) \\
 & \qquad\qquad\qquad\quad \wedge \quad (n < i)))))))) \\
\rightarrow & \quad (\text{getseg}\,(i,\, j,\, \mathit{l1}) = \text{getseg}\,(i,\, j,\, \mathit{l2}))
\end{array}
$$

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

$$
\begin{array}{ll}
 & (i \in \mathbf{N}) \\
\wedge & ((n \in \mathbf{N}) \\
 & \quad \wedge \quad ((j \not\simeq 0) \\
 & \qquad \wedge \quad ((k \not\simeq 0) \\
 & \qquad\quad \wedge \quad (((\text{nested-intervals}\,(1 + n,\, k - 1,\, 1 + i,\, j - 1) \\
 & \qquad\qquad \wedge \quad (\text{getseg}\,(1 + n,\, k - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + n,\, k - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \rightarrow \quad (\text{getseg}\,(1 + i,\, j - 1,\, \mathit{l1}) \\
 & \qquad\qquad\qquad = \quad \text{getseg}\,(1 + i,\, j - 1,\, \mathit{l2}))) \\
 & \qquad\qquad \wedge \quad (\text{nested-intervals}\,(n,\, k,\, i,\, j) \\
 & \qquad\qquad\quad \wedge \quad (\text{getseg}\,(n,\, k,\, \mathit{l1}) \\
 & \qquad\qquad\qquad\quad = \quad \text{getseg}\,(n,\, k,\, \mathit{l2})))))))) \\
\rightarrow & \quad (\text{getseg}\,(i,\, j,\, \mathit{l1}) = \text{getseg}\,(i,\, j,\, \mathit{l2}))
\end{array}
$$

THEOREM: equality-of-getseg-backchain-proof
$((i \in \mathbf{N})$
$\wedge \quad ((n \in \mathbf{N})$
$\quad \wedge \quad (\text{nested-intervals}\,(n,\, k,\, i,\, j)$
$\quad\quad \wedge \quad (\text{getseg}\,(n,\, k,\, l1) = \text{getseg}\,(n,\, k,\, l2)))))$
$\rightarrow \quad (\text{getseg}\,(i,\, j,\, l1) = \text{getseg}\,(i,\, j,\, l2))$

THEOREM: equality-of-getseg-backchain
$(\text{nested-intervals}\,(n,\, k,\, i,\, j) \wedge (\text{getseg}\,(n,\, k,\, l1) = \text{getseg}\,(n,\, k,\, l2)))$
$\rightarrow \quad (\text{getseg}\,(i,\, j,\, l1) = \text{getseg}\,(i,\, j,\, l2))$

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

EVENT: Disable getnth-putnth-coincidence1.

THEOREM: getnth2-putnth2-coincidence
$((i < \text{length}\,(l)) \wedge (j < \text{length}\,(\text{getnth}\,(i,\, l))))$
$\rightarrow \quad (\text{getnth2}\,(i,\, j,\, \text{putnth2}\,(v,\, i,\, j,\, l)) = v)$

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

EVENT: Disable getnth2-putnth2-non-interference2.

THEOREM: fixlength-of-length-0
$(n \simeq \texttt{0}) \rightarrow (\text{fixlength}\,(n,\ l,\ x) = \texttt{'nil})$

EVENT: Disable fixlength-of-length-0.

THEOREM: fixlength-idempotence
$\text{fixlength}\,(n,\ \text{fixlength}\,(n,\ l,\ a),\ b) = \text{fixlength}\,(n,\ l,\ a)$

EVENT: Disable fixlength-idempotence.

THEOREM: fixlength-noop
$(\text{plistp}\,(l) \wedge (n = \text{length}\,(l))) \rightarrow (\text{fixlength}\,(n,\ l,\ \textit{default}) = l)$

EVENT: Disable fixlength-noop.

DEFINITION:
$\text{reverse}\,(l)$
$= \quad$ **if** $\text{listp}\,(l)$ **then** $\text{append}\,(\text{reverse}\,(\text{cdr}\,(l)),\ \text{cons}\,(\text{car}\,(l),\ \texttt{'nil}))$
else `'nil` **endif**

THEOREM: plistp-reverse
$\text{plistp}\,(\text{reverse}\,(a))$

EVENT: Disable plistp-reverse.

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

EVENT: Disable reverse-append.

THEOREM: reverse-list1
$\text{reverse}\,(\text{cons}\,(a,\ \texttt{'nil})) = \text{cons}\,(a,\ \texttt{'nil})$

EVENT: Disable reverse-list1.

THEOREM: reverse-list2
$\text{reverse}\,(\text{cons}\,(a,\ \text{cons}\,(a,\ \texttt{'nil}))) = \text{cons}\,(a,\ \text{cons}\,(a,\ \texttt{'nil}))$

EVENT: Disable reverse-list2.

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

EVENT: Disable reverse-reverse.

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

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

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

DEFINITION:
non-zero-listp (l)
= **if** listp (l) **then** $(\text{car}(l) \not\simeq$ `0`$) \wedge$ 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
finite-numberp $(a,\ b) \rightarrow$ finite-numberp $(a - 1,\ b)$

EVENT: Disable finite-number-sub1.

THEOREM: finite-numberp-remainder
$(b \not\simeq \texttt{0}) \rightarrow$ finite-numberp $(a\ \textbf{mod}\ b,\ b)$

EVENT: Disable finite-numberp-remainder.

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

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

EVENT: Disable finite-numberp-getnth.

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

EVENT: Disable numberp-getnth.

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

EVENT: Disable lessp-getnth.

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

EVENT: Disable lessp-getnth-linear.

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

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

THEOREM: finite-number-listp-list
finite-numberp (*n*, *lub*) → finite-number-listp (cons (*n*, `'nil`), *lub*)

EVENT: Disable finite-number-listp-list.

THEOREM: finite-number-listp-append
(finite-number-listp (*a*, *lub*) ∧ finite-number-listp (*b*, *lub*))
→ finite-number-listp (append (*a*, *b*), *lub*)

EVENT: Disable finite-number-listp-append.

THEOREM: finite-number-listp-remove
finite-number-listp (*l*, *lub*) → finite-number-listp (remove (*x*, *l*), *lub*)

EVENT: Disable finite-number-listp-remove.

THEOREM: finite-number-listp-permutation
(finite-number-listp (*b*, *lub*) ∧ permutation (*a*, *b*))
→ finite-number-listp (*a*, *lub*)

EVENT: Disable finite-number-listp-permutation.

THEOREM: finite-number-listp-putnth
(finite-number-listp (*l*, *lub*) ∧ finite-numberp (*v*, *lub*))
→ finite-number-listp (putnth (*v*, *n*, *l*), *lub*)

EVENT: Disable finite-number-listp-putnth.

THEOREM: finite-number-listp-getseg
(finite-number-listp (*l*, *lub*)
∧ **if** length (*l*) < (*n* + *k*) **then** `'*1*false`
else `'*1*true` **endif**)
→ finite-number-listp (getseg (*n*, *k*, *l*), *lub*)

EVENT: Disable finite-number-listp-getseg.

THEOREM: finite-number-listp-putseg
(finite-number-listp (*l*, *lub*) ∧ finite-number-listp (*s*, *lub*))
→ finite-number-listp (putseg (*s*, *n*, *l*), *lub*)

EVENT: Disable finite-number-listp-putseg.

THEOREM: finite-number-listp-fixlength
(finite-number-listp (*l*, *lub*) ∧ finite-numberp (*default*, *lub*))
→ finite-number-listp (fixlength (*n*, *l*, *default*), *lub*)

EVENT: Disable finite-number-listp-fixlength.

THEOREM: finite-number-listp-for-zero-list
(*lub* ≠ 0) → 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*) ∧ (*i* < *j*)) → finite-number-listp (*l*, *j*)

THEOREM: non-zero-listp-list
(*a* ≠ 0) → non-zero-listp (cons (*a*, 'nil))

EVENT: Disable non-zero-listp-list.

THEOREM: non-zero-listp-append
(non-zero-listp (*a*) ∧ non-zero-listp (*b*)) → non-zero-listp (append (*a*, *b*))

EVENT: Disable non-zero-listp-append.

THEOREM: non-zero-listp-putnth
(non-zero-listp (*l*) ∧ (*v* ≠ 0)) → non-zero-listp (putnth (*v*, *n*, *l*))

EVENT: Disable non-zero-listp-putnth.

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

EVENT: Disable non-zero-listp-getseg.

THEOREM: non-zero-listp-putseg
(non-zero-listp (*l*) ∧ non-zero-listp (*s*)) → non-zero-listp (putseg (*s*, *n*, *l*))

EVENT: Disable non-zero-listp-putseg.

THEOREM: non-zero-listp-fixlength
(non-zero-listp (*l*) ∧ (*default* ≠ 0))
→ non-zero-listp (fixlength (*n*, *l*, *default*))

EVENT: Disable non-zero-listp-fixlength.

THEOREM: car-nthcdr
(*n* < length (*l*)) → (car (nthcdr (*n*, *l*)) = getnth (*n*, *l*))

EVENT: Disable car-nthcdr.

THEOREM: cdr-nthcdr
(*n* < length (*l*)) → (cdr (nthcdr (*n*, *l*)) = nthcdr (1 + *n*, *l*))

EVENT: Disable cdr-nthcdr.

THEOREM: listp-nthcdr
(*n* < length (*l*)) → listp (nthcdr (*n*, *l*))

EVENT: Disable listp-nthcdr.

THEOREM: plistp-firstn
plistp (firstn (*n*, *l*))

EVENT: Disable plistp-firstn.

THEOREM: plistp-nthcdr
plistp (nthcdr (*n*, *l*)) = plistp (*l*)

EVENT: Disable plistp-nthcdr.

THEOREM: length-firstn
if length (*l*) < *n* **then** `'*1*false`
else `'*1*true` **endif**
→ (length (firstn (*n*, *l*)) = fix (*n*))

EVENT: Disable length-firstn.

THEOREM: length-nthcdr
length (nthcdr (*n*, *l*)) = (length (*l*) − *n*)

EVENT: Disable length-nthcdr.

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

EVENT: Disable firstn-zero.

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

EVENT: Disable nthcdr-zero.

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

EVENT: Disable nthcdr-with-large-n.

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

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

EVENT: Disable firstn-nthcdr.

THEOREM: nthcdr-append-right

if $n <$ length (a) **then** `'*1*false`
else `'*1*true` **endif**
$\rightarrow$ (nthcdr $(n,$ append $(a,\ b)) =$ nthcdr $(n -$ length $(a),\ b))$

EVENT: Disable nthcdr-append-right.

THEOREM: nthcdr-append-left
if length $(a) < n$ **then** `'*1*false`
else `'*1*true` **endif**
$\rightarrow$ (nthcdr $(n,$ append $(a,\ b)) =$ append (nthcdr $(n,\ a),\ b))$

EVENT: Disable nthcdr-append-left.

DEFINITION:
tablep $(n,\ l)$
$=$ **if** listp (l)
then plistp (car (l))
$\wedge$ ((length (car $(l)) = n) \wedge$ tablep $(n,$ cdr $(l)))$
else $l =$ `'nil` **endif**

THEOREM: length-of-nthcdr-goes-down
(listp $(l) \wedge (n \not\simeq$ `0`$))$
$\rightarrow$ ((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 $(n,$ nthcdr $(n,\ l)))$
else `'nil` **endif**

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

DEFINITION:
flatten $(table)$
$=$ **if** listp $(table)$ **then** append (car $(table)$, flatten (cdr $(table)))$
else `'nil` **endif**

THEOREM: plistp-table
plistp $(l) \rightarrow$ plistp (table $(n,\ l))$

EVENT: Disable plistp-table.

THEOREM: length-table
$((n \not\simeq \texttt{0}) \wedge ((\text{length}\,(l)\ \textbf{mod}\ n) = \texttt{'0}))$
$\rightarrow\quad (\text{length}\,(\text{table}\,(n,\ l)) = (\text{length}\,(l) \div n))$

EVENT: Disable length-table.

THEOREM: tablep-table
$(n \not\simeq \texttt{0}) \rightarrow \text{tablep}\,(n,\ \text{table}\,(n,\ l))$

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
$((n \not\simeq \texttt{0}) \wedge (((\text{length}\,(l)\ \textbf{mod}\ n) = \texttt{'0}) \wedge (i < (\text{length}\,(l) \div n))))$
$\rightarrow\quad (\text{length}\,(\text{getnth}\,(i,\ \text{table}\,(n,\ l))) = n)$

EVENT: Disable length-getnth-table.

THEOREM: plistp-getnth-table
$((n \not\simeq \texttt{0}) \wedge (((\text{length}\,(l)\ \textbf{mod}\ n) = \texttt{'0}) \wedge (i < (\text{length}\,(l) \div n))))$
$\rightarrow\quad \text{plistp}\,(\text{getnth}\,(i,\ \text{table}\,(n,\ l)))$

EVENT: Disable plistp-getnth-table.

THEOREM: getseg-equals-firstn-nthcdr
(**if** length $(l) < n$ **then** `'*1*false`
else `'*1*true` **endif**
$\wedge$ **if** $(\text{length}\,(l) - n) < k$ **then** `'*1*false`
else `'*1*true` **endif**)
$\rightarrow\quad (\text{getseg}\,(n,\ k,\ l) = \text{firstn}\,(k,\ \text{nthcdr}\,(n,\ l)))$

EVENT: Disable getseg-equals-firstn-nthcdr.

THEOREM: getseg-for-length
$(\text{plistp}\,(l) \wedge (k = \text{length}\,(l))) \rightarrow (\text{getseg}\,(\texttt{'0},\ k,\ l) = l)$

EVENT: Disable getseg-for-length.

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

EVENT: Disable plistp-flatten.

THEOREM: length-flatten
((*n* ≉ 0) ∧ tablep (*n*, *l*)) → (length (flatten (*l*)) = (length (*l*) ∗ *n*))

EVENT: Disable length-flatten.

THEOREM: table-append
((*n* ≉ 0) ∧ ((length (*a*) **mod** *n*) = '0))
→ (table (*n*, append (*a*, *b*)) = append (table (*n*, *a*), table (*n*, *b*)))

EVENT: Disable table-append.

THEOREM: nthcdr-for-length
plistp (*l*) → (nthcdr (length (*l*), *l*) = 'nil)

EVENT: Disable nthcdr-for-length.

THEOREM: table-for-length
(plistp (*l*) ∧ (length (*l*) ≉ 0)) → (table (length (*l*), *l*) = cons (*l*, 'nil))

EVENT: Disable table-for-length.

THEOREM: table-flatten
((*n* ≉ 0) ∧ tablep (*n*, *table*)) → (table (*n*, flatten (*table*)) = *table*)

EVENT: Disable table-flatten.

THEOREM: listp-table
(*n* ≉ 0) → (listp (table (*n*, *l*)) = listp (*l*))

EVENT: Disable listp-table.

DEFINITION:
integer-nthcdr-induction (*i*, *n*, *l*)
= **if** *n* ≃ 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
getnth $(i,$ nthcdr $(n,\ l)) =$ getnth $(n + i,\ l)$

EVENT: Disable getnth-nthcdr.

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

EVENT: Disable getseg-nthcdr-proof.

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

EVENT: Disable getseg-nthcdr.

THEOREM: getnth-table
$(((i * n) <$ length $(l)) \wedge (n \not\simeq$ `0`$))$
$\rightarrow$ (getnth $(i,$ table $(n,\ l)) =$ getseg $(i * n,\ n,\ l)$)

EVENT: Disable getnth-table.

THEOREM: getnth-table-inverse
$(((i * n) <$ length $(l)) \wedge (n \not\simeq$ `0`$))$
$\rightarrow$ (getseg $(i * n,\ n,\ l) =$ getnth $(i,$ table $(n,\ l)))$

EVENT: Disable getnth-table-inverse.

THEOREM: rewrite-getnth-from-table
$(n \not\simeq$ `0`$)$
$\wedge$ $((i < n)$
$\wedge$ **if** length $(l) < (n + (n * j))$ **then** `'*1*false`
else `'*1*true` **endif**$)$
$\rightarrow$ (getnth $(i + (n * j),\ l) =$ getnth $(i,$ getnth $(j,$ table $(n,\ l))))$

EVENT: Disable rewrite-getnth-from-table.

THEOREM: rewrite-getnth-from-table0

$((n \not\simeq \mathtt{0})$
$\wedge$ **if** $\text{length}(l) < (n + (n * j))$ **then** `'*1*false`
else `'*1*true` **endif**)
$\rightarrow$ $(\text{getnth}(n * j, l) = \text{getnth}(\mathtt{'0}, \text{getnth}(j, \text{table}(n, l))))$

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**
$\rightarrow$ $(\text{nthcdr}(i, \text{putseg}(s, n, l)) = \text{nthcdr}(i, l))$

EVENT: Disable nthcdr-putseg.

THEOREM: table-putseg-subcase
$(\text{plistp}(s)$
$\wedge$ $((\text{length}(s) \not\simeq \mathtt{0})$

$\wedge$ **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 $(s) = n$)
$\wedge$ (($n \not\simeq$ `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 $(s) = n$)
$\wedge$ (($n \not\simeq$ `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
($(n \not\simeq$ `0`)
$\wedge$ (($i \in \mathbf{N}$) $\wedge$ (($i <$ 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))

$\wedge$ (finite-number-listp (car (l), lub)
$\wedge$ ((length (car (l)) = n)
$\wedge$ finite-number-tablep (n, cdr (l), lub)))
else l = `'nil` **endif**

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

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

THEOREM: finite-number-listp-nil
finite-number-listp (`'nil`, lub)

EVENT: Disable finite-number-listp-nil.

THEOREM: finite-number-listp-flatten
(($n \not\simeq$ `0`) $\wedge$ finite-number-tablep (n, l, lub))
$\rightarrow$ finite-number-listp (flatten (l), lub)

EVENT: Disable finite-number-listp-flatten.

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

EVENT: Disable finite-number-tablep-list.

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

EVENT: Disable finite-number-tablep-append.

THEOREM: getnth-plus-1-is-cadr-nthcdr
getnth (`'1` + n, l) = car (cdr (nthcdr (n, l)))

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

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

EVENT: Disable cadr-firstn.

THEOREM: getnth-plus-1-is-cadr-getseg
((’1 < k)
 ∧ (**if** length (l) < n **then** ’*1*false
 else ’*1*true **endif**
 ∧ **if** (length (l) − n) < k **then** ’*1*false
 else ’*1*true **endif**))
→ (getnth (’1 + n, l) = car (cdr (getseg (n, k, l))))

THEOREM: getnth-plus-1-is-cadr-getseg-inverse
((’1 < k)
 ∧ (**if** length (l) < n **then** ’*1*false
 else ’*1*true **endif**
 ∧ **if** (length (l) − n) < k **then** ’*1*false
 else ’*1*true **endif**))
→ (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
 else ’*1*true **endif**
 ∧ **if** length (l) < i **then** ’*1*false
 else ’*1*true **endif**)
→ (firstn (i, fixlength (n, l, *default*)) = firstn (i, l))

EVENT: Disable firstn-fixlength.

THEOREM: getseg-fixlength
(plistp (l) ∧ ((n ≠ 0) ∧ (length (l) < n)))
→ (getseg (’0, length (l), fixlength (n, l, *default*))
 = getseg (’0, length (l), l))

EVENT: Disable getseg-fixlength.

THEOREM: number-listp-finite-number-listp
finite-number-listp (l, *lub*) → 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*) → finite-number-listp (nonlast (*l*), *lub*)

EVENT: Disable finite-number-listp-nonlast.

THEOREM: difference-sub1-plus-crock
((*a* ≉ 0) ∧ (*b* ≉ 0)) → ((((*a* + *b*) − 1) − *a*) = (*b* − 1))

EVENT: Disable difference-sub1-plus-crock.

THEOREM: nonlast-append
plistp (*a*)
→ (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* ≃ 0 **then** *lub* − 1
else *n* − 1 **endif**

THEOREM: decr-mod-size
((*n* ≉ 0) ∧ (*a* < *n*)) → (decr-mod (*a*, *n*) < *n*)

EVENT: Disable decr-mod-size.

THEOREM: incr-mod-equals-remainder-add1
(*n* < *lub*) → (incr-mod (*n*, *lub*) = ((1 + *n*) **mod** *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: $\text{enq}\,(item,\ table) = \text{append}\,(table,\ \text{cons}\,(item,\ \texttt{'nil}))$

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

DEFINITION: $\text{qemptyp}\,(table) = (\text{length}\,(table) = \texttt{'0})$

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

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
$(\text{plistp}\,(queue) \wedge (\neg\ \text{qemptyp}\,(queue))) \rightarrow \text{plistp}\,(\text{deq}\,(queue))$

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
$(\neg\ \text{qemptyp}\,(queue)) \rightarrow (\text{length}\,(queue) \neq \texttt{'0})$

EVENT: Disable length-non-empty-queue.

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

EVENT: Disable length-deq.

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

EVENT: Disable length-qreplace.

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

EVENT: Disable not-qemptyp-enq.

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

EVENT: Disable not-qfullp-deq.

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

EVENT: Disable qfirst-enq.

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

EVENT: Disable finite-number-listp-enq.

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

EVENT: Disable finite-number-listp-deq.

THEOREM: finite-number-listp-qreplace
(finite-number-listp (*queue*, *lub*) ∧ finite-numberp (*item*, *lub*))
→ finite-number-listp (qreplace (*item*, *queue*), *lub*)

EVENT: Disable finite-number-listp-qreplace.

THEOREM: finite-numberp-qfirst
(finite-number-listp (*queue*, *lub*) ∧ (¬ qemptyp (*queue*)))
→ finite-numberp (qfirst (*queue*), *lub*)

EVENT: Disable finite-numberp-qfirst.

THEOREM: numberp-qfirst
((¬ qemptyp (*queue*)) ∧ number-listp (*queue*)) → (qfirst (*queue*) ∈ **N**)

EVENT: Disable numberp-qfirst.

THEOREM: deq-is-remove
(setp (*l*) ∧ ('0 < length (*l*))) → (deq (*l*) = remove (qfirst (*l*), *l*))

EVENT: Disable deq-is-remove.

THEOREM: member-qfirst-queue
(¬ qemptyp (*queue*)) → (qfirst (*queue*) ∈ *queue*)

EVENT: Disable member-qfirst-queue.

THEOREM: permutation-enq-deq
(¬ qemptyp (*queue*)) → 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:
delta$(a,\ b,\ max)$
$=$ **if if** $a < b$ **then** `'*1*false`
else `'*1*true` **endif then** $(max - a) + b$
else $b - a$ **endif**

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

EVENT: Disable delta-a-a.

THEOREM: delta-incr-mod-a
$((a < max) \wedge (b < max))$
$\rightarrow$ (delta(incr-mod$(a,\ max),\ b,\ max)$
$=$ **if** delta$(a,\ b,\ max) =$ `'1` **then** max
else delta$(a,\ b,\ max) - 1$ **endif**)

EVENT: Disable delta-incr-mod-a.

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

EVENT: Disable delta-incr-mod-b.

THEOREM: delta-equals-max-iff-a-equals-b
$((a \in \mathbf{N}) \wedge ((b \in \mathbf{N}) \wedge ((a < max) \wedge (b < max))))$
$\rightarrow$ $((\text{delta}(a,\ b,\ max) = max) = (a = b))$

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

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

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

DEFINITION:
array-queuep (*queue*)
= (plistp (*queue*)
∧ ((length (*queue*) = ('4 + getnth ('3, *queue*)))
∧ ((getnth ('0, *queue*) ∈ **N**)
∧ ((getnth ('1, *queue*) ∈ **N**)
∧ ((getnth ('2, *queue*) ∈ **N**)
∧ ((getnth ('3, *queue*) ≠ 0)
∧ ((getnth ('0, *queue*)
< getnth ('3,
queue))
∧ ((getnth ('1,
queue)
< getnth ('3,
queue))
∧ ((getnth ('2,
queue)
< (1 + getnth ('3,
queue)))
∧ array-qindex-relation (*queue*)))))))))

THEOREM: plistp-array-queue
array-queuep (*queue*) → plistp (*queue*)

EVENT: Disable plistp-array-queue.

THEOREM: length-array-queue
array-queuep (*queue*) → (length (*queue*) = ('4 + getnth ('3, *queue*)))

EVENT: Disable length-array-queue.

THEOREM: numberp-qhead
array-queuep (*queue*) → (getnth ('0, *queue*) ∈ **N**)

EVENT: Disable numberp-qhead.

THEOREM: numberp-qtail
array-queuep (*queue*) → (getnth ('1, *queue*) ∈ **N**)

EVENT: Disable numberp-qtail.

THEOREM: numberp-qcurrlength
array-queuep (*queue*) → (getnth ('2, *queue*) ∈ **N**)

EVENT: Disable numberp-qcurrlength.

THEOREM: numberp-qmaxlength
array-queuep (*queue*) → (getnth ('3, *queue*) ∈ **N**)

EVENT: Disable numberp-qmaxlength.

THEOREM: qmaxlength-non-zero
array-queuep (*queue*) → (getnth ('3, *queue*) ≠ '0)

EVENT: Disable qmaxlength-non-zero.

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

EVENT: Disable qmaxlength-greater-than-zero.

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

EVENT: Disable lessp-qhead-qmaxlength.

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

EVENT: Disable lessp-qtail-qmaxlength.

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

EVENT: Disable lessp-qcurrlength-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*)
→ (length (getseg ('4, length (*queue*) − '4, *queue*)) = getnth ('3, *queue*))

EVENT: Disable length-array-qarray.

DEFINITION:
array-enq (*item*, *queue*)
= putnth (incr-mod (getnth ('1, *queue*), getnth ('3, *queue*)),
'1,
putnth (1 + getnth ('2, *queue*),
'2,
putnth (*item*, '4 + getnth ('1, *queue*), *queue*)))

DEFINITION:
array-deq (*queue*)
= putnth (incr-mod (getnth ('0, *queue*), getnth ('3, *queue*)),
'0,
putnth (getnth ('2, *queue*) − 1, '2, *queue*))

DEFINITION:
array-qfirst (*queue*) = getnth ('4 + getnth ('0, *queue*), *queue*)

DEFINITION:
array-qfullp (*queue*) = (getnth ('2, *queue*) = getnth ('3, *queue*))

DEFINITION: array-qemptyp (*queue*) = (getnth ('2, *queue*) ≃ 0)

THEOREM: length-array-enq
length (array-enq (*item*, *queue*)) = length (*queue*)

EVENT: Disable length-array-enq.

THEOREM: plistp-array-enq
plistp (*queue*) → plistp (array-enq (*item*, *queue*))

EVENT: Disable plistp-array-enq.

THEOREM: array-qhead-array-enq
getnth ('0, array-enq (*item*, *queue*)) = getnth ('0, *queue*)

EVENT: Disable array-qhead-array-enq.

THEOREM: array-qtail-array-enq
array-queuep (*queue*)
→ (getnth ('1, array-enq (*item*, *queue*))
= incr-mod (getnth ('1, *queue*), getnth ('3, *queue*)))

EVENT: Disable array-qtail-array-enq.

THEOREM: array-qcurrlength-array-enq
array-queuep (*queue*)
→ (getnth ('2, array-enq (*item*, *queue*)) = (1 + getnth ('2, *queue*)))

EVENT: Disable array-qcurrlength-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*)
→ (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*) ∧ (¬ array-qfullp (*queue*)))
→ 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*) → plistp (array-deq (*queue*))

EVENT: Disable plistp-array-deq.

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

EVENT: Disable array-qhead-array-deq.

THEOREM: array-qtail-array-deq
getnth ('1, array-deq (*queue*)) = getnth ('1, *queue*)

EVENT: Disable array-qtail-array-deq.

THEOREM: array-qcurrlength-array-deq
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ (getnth ('2, array-deq (*queue*)) = (getnth ('2, *queue*) − 1))

EVENT: Disable array-qcurrlength-array-deq.

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

EVENT: Disable array-qmaxlength-array-deq.

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

EVENT: Disable array-qarray-array-deq.

THEOREM: array-queuep-array-deq
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ array-queuep (array-deq (*queue*))

EVENT: Disable array-queuep-array-deq.

THEOREM: not-array-qemptyp-array-enq
array-queuep (*queue*) → (¬ array-qemptyp (array-enq (*item*, *queue*)))

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

THEOREM: not-array-qfullp-array-deq
array-queuep (*queue*) → (¬ array-qfullp (array-deq (*queue*)))

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

THEOREM: array-qcurrlength-in-a-full-queue
(array-queuep (*queue*) ∧ array-qfullp (*queue*))
→ (getnth ('2, *queue*) = getnth ('3, *queue*))

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

THEOREM: array-qcurrlength-in-a-non-full-queue
(array-queuep (*queue*) ∧ (¬ array-qfullp (*queue*)))
→ (getnth ('2, *queue*) < getnth ('3, *queue*))

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

THEOREM: array-qcurrlength-in-an-empty-queue
(array-queuep (*queue*) ∧ array-qemptyp (*queue*))
→ (getnth ('2, *queue*) = '0)

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

THEOREM: array-qcurrlength-in-a-non-empty-queue
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ (getnth ('2, *queue*) ≠ '0)

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

DEFINITION:
array-nonlast (*queue*)
= putnth (decr-mod (getnth ('1, *queue*), getnth ('3, *queue*)),
'1,
putnth (getnth ('2, *queue*) − 1, '2, *queue*))

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

EVENT: Disable length-array-nonlast.

THEOREM: plistp-array-nonlast
plistp (*queue*) → plistp (array-nonlast (*queue*))

EVENT: Disable plistp-array-nonlast.

THEOREM: array-qhead-array-nonlast
getnth ('0, array-nonlast (*queue*)) = 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-qcurrlength-array-nonlast
array-queuep (*queue*)
→ (getnth ('2, array-nonlast (*queue*)) = (getnth ('2, *queue*) − 1))

EVENT: Disable array-qcurrlength-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
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ array-queuep (array-nonlast (*queue*))

EVENT: Disable array-queuep-array-nonlast.

DEFINITION:
array-qreplace (*item*, *queue*) = array-enq (*item*, array-nonlast (*queue*))

THEOREM: length-array-qreplace
length (array-qreplace (*item*, *queue*)) = length (*queue*)

EVENT: Disable length-array-qreplace.

THEOREM: plistp-array-qreplace
plistp (*queue*) → plistp (array-qreplace (*item*, *queue*))

EVENT: Disable plistp-array-qreplace.

THEOREM: array-qhead-array-qreplace
getnth ('0, array-qreplace (*item*, *queue*)) = getnth ('0, *queue*)

EVENT: Disable array-qhead-array-qreplace.

THEOREM: array-qtail-array-qreplace
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ (getnth ('1, array-qreplace (*item*, *queue*)) = getnth ('1, *queue*))

EVENT: Disable array-qtail-array-qreplace.

THEOREM: array-qcurrlength-array-qreplace
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ (getnth ('2, array-qreplace (*item*, *queue*)) = getnth ('2, *queue*))

EVENT: Disable array-qcurrlength-array-qreplace.

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

EVENT: Disable array-qmaxlength-array-qreplace.

THEOREM: array-qarray-array-qreplace
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ (getseg ('4, getnth ('3, *queue*), array-qreplace (*item*, *queue*))
= putnth (*item*,
decr-mod (getnth ('1, *queue*), getnth ('3, *queue*)),
getseg ('4, getnth ('3, *queue*), *queue*)))

EVENT: Disable array-qarray-array-qreplace.

THEOREM: not-array-qfullp-array-nonlast
array-queuep (*queue*) → (¬ array-qfullp (array-nonlast (*queue*)))

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

THEOREM: array-queuep-array-qreplace
(array-queuep (*queue*) ∧ (¬ array-qemptyp (*queue*)))
→ 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-qemptyp-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-qemptyp
(array-queuep (*queue*) ∧ array-qfullp (*queue*))
→ (¬ array-qemptyp (*queue*))

THEOREM: array-qhead-equals-array-qtail-in-an-empty-queue
(array-queuep (*queue*) ∧ array-qemptyp (*queue*))
→ (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*)
→ ((getnth (`'0`, *queue*) = getnth (`'1`, *queue*))
= (array-qfullp (*queue*) ∨ array-qemptyp (*queue*)))

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

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

= **if 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 < length (s)) ∧ (b < length (s)))
→ (length (delta-segment (a, b, s)) = delta (a, b, 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*)
∧ (($b \in \mathbf{N}$)
∧ (**if** length (s) < b **then** `'*1*false`
else `'*1*true` **endif**
∧ (($a \in \mathbf{N}$)
∧ **if** length (s) < a **then** `'*1*false`
else `'*1*true` **endif**))))
→ finite-number-listp (delta-segment (a, b, s), *lub*)

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

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

DEFINITION:
mapup-queue (*queue*)
= **if** array-qemptyp (*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*)
∧ finite-number-listp (getseg (`'4`, getnth (`'3`, *queue*), *queue*), *lub*))
→ 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**)

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

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

THEOREM: array-qemtpyp-array-deq-when-delta-is-1
$(\text{array-queuep}(queue)$
$\wedge \quad ((\neg \text{array-qemptyp}(queue))$
$\wedge \quad (\text{delta}(\text{getnth}(\texttt{'0}, queue), \text{getnth}(\texttt{'1}, queue), \text{getnth}(\texttt{'3}, queue))$
$= \quad \texttt{'1})))$
$\rightarrow \quad \text{array-qemptyp}(\text{array-deq}(queue))$

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

THEOREM: array-qemtpyp-array-deq-when-delta-is-1-alternative
$(\text{array-queuep}(queue)$
$\wedge \quad ((\neg \text{array-qemptyp}(queue)) \wedge \text{array-qemptyp}(\text{array-deq}(queue))))$
$\rightarrow \quad (\text{delta}(\text{getnth}(\texttt{'0}, queue), \text{getnth}(\texttt{'1}, queue), \text{getnth}(\texttt{'3}, queue)) = \texttt{'1})$

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

THEOREM: correctness-of-array-deq-case-3
$(\text{array-queuep}(queue)$
$\wedge \quad ((\neg \text{array-qemptyp}(queue)) \wedge \text{array-qemptyp}(\text{array-deq}(queue))))$
$\rightarrow \quad (\text{cdr}(\text{delta-segment}(\text{getnth}(\texttt{'0}, queue),$
$\text{getnth}(\texttt{'1}, queue),$
$\text{getseg}(\texttt{'4}, \text{getnth}(\texttt{'3}, queue), queue)))$
$= \quad \texttt{'nil})$

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

THEOREM: array-qmaxlength-must-be-1
$(\text{array-queuep}(queue)$
$\wedge \quad (\text{array-qfullp}(queue) \wedge \text{array-qemptyp}(\text{array-deq}(queue))))$
$\rightarrow \quad (\text{getnth}(\texttt{'3}, queue) = \texttt{'1})$

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

THEOREM: length-delta-segment-a-a
$(\text{array-queuep}(queue)$

$\wedge$ (array-qfullp (*queue*) $\wedge$ array-qemptyp (array-deq (*queue*))))
$\rightarrow$ (length (delta-segment (getnth (`'0`, *queue*),
getnth (`'0`, *queue*),
getseg (`'4`, getnth (`'3`, *queue*), *queue*)))
= `'1`)

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

THEOREM: correctness-of-array-deq-case-1
(array-queuep (*queue*)
$\wedge$ (array-qfullp (*queue*) $\wedge$ array-qemptyp (array-deq (*queue*))))
$\rightarrow$ (cdr (delta-segment (getnth (`'0`, *queue*),
getnth (`'0`, *queue*),
getseg (`'4`, getnth (`'3`, *queue*), *queue*)))
= `'nil`)

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

THEOREM: correctness-of-array-deq
(array-queuep (*queue*) $\wedge$ ($\neg$ array-qemptyp (*queue*)))
$\rightarrow$ (mapup-queue (array-deq (*queue*)) = deq (mapup-queue (*queue*)))

EVENT: Disable correctness-of-array-deq.

THEOREM: correctness-of-array-qfirst
(array-queuep (*queue*) $\wedge$ ($\neg$ array-qemptyp (*queue*)))
$\rightarrow$ (array-qfirst (*queue*) = qfirst (mapup-queue (*queue*)))

EVENT: Disable correctness-of-array-qfirst.

THEOREM: correctness-of-array-qemptyp
array-queuep (*queue*)
$\rightarrow$ (array-qemptyp (*queue*) = qemptyp (mapup-queue (*queue*)))

EVENT: Disable correctness-of-array-qemptyp.

THEOREM: correctness-of-array-qfullp
(array-queuep (*queue*) $\wedge$ (*max* = getnth (`'3`, *queue*)))
$\rightarrow$ (array-qfullp (*queue*) = qfullp (mapup-queue (*queue*), *max*))

THEOREM: length-delta-segment-decr-mod-arg1
(($a \in \mathbf{N}$) $\wedge$ ((b = length (s)) $\wedge$ ($a <$ length (s))))
$\rightarrow$ (length (delta-segment (decr-mod (a, b), a, s)) = `'1`)

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

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

EVENT: Disable nonlast-of-length-1.

THEOREM: correctness-of-array-nonlast-case1
$$\begin{array}{ll}
 & (\text{array-queuep}\,(\mathit{queue}) \\
\wedge & ((\neg\ \text{array-qemptyp}\,(\mathit{queue})) \wedge \text{array-qemptyp}\,(\text{array-nonlast}\,(\mathit{queue})))) \\
\rightarrow & (\text{nonlast}\,(\text{delta-segment}\,(\text{getnth}\,(\texttt{'0},\ \mathit{queue}), \\
 & \qquad\qquad\qquad\qquad \text{getnth}\,(\texttt{'1},\ \mathit{queue}), \\
 & \qquad\qquad\qquad\qquad \text{getseg}\,(\texttt{'4},\ \text{getnth}\,(\texttt{'3},\ \mathit{queue}),\ \mathit{queue}))) \\
 & \quad = \quad \texttt{'nil})
\end{array}$$

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

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

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

THEOREM: correctness-of-array-nonlast-case2
$$\begin{array}{ll}
 & (\text{array-queuep}\,(\mathit{queue}) \\
\wedge & ((\neg\ \text{array-qemptyp}\,(\mathit{queue})) \\
 & \quad \wedge \quad (\neg\ \text{array-qemptyp}\,(\text{array-nonlast}\,(\mathit{queue}))))) \\
\rightarrow & (\text{delta-segment}\,(\text{getnth}\,(\texttt{'0},\ \mathit{queue}), \\
 & \qquad\qquad \text{decr-mod}\,(\text{getnth}\,(\texttt{'1},\ \mathit{queue}),\ \text{getnth}\,(\texttt{'3},\ \mathit{queue})), \\
 & \qquad\qquad \text{getseg}\,(\texttt{'4},\ \text{getnth}\,(\texttt{'3},\ \mathit{queue}),\ \mathit{queue})) \\
 & \quad = \quad \text{nonlast}\,(\text{delta-segment}\,(\text{getnth}\,(\texttt{'0},\ \mathit{queue}), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{getnth}\,(\texttt{'1},\ \mathit{queue}), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{getseg}\,(\texttt{'4},\ \text{getnth}\,(\texttt{'3},\ \mathit{queue}),\ \mathit{queue}))))
\end{array}$$

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

THEOREM: correctness-of-array-nonlast
$$\begin{array}{ll}
 & (\text{array-queuep}\,(\mathit{queue}) \wedge (\neg\ \text{array-qemptyp}\,(\mathit{queue}))) \\
\rightarrow & (\text{mapup-queue}\,(\text{array-nonlast}\,(\mathit{queue})) = \text{nonlast}\,(\text{mapup-queue}\,(\mathit{queue})))
\end{array}$$

EVENT: Disable correctness-of-array-nonlast.

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

EVENT: Disable correctness-of-array-qreplace.

THEOREM: length-mapup-queue-is-qcurrlength
array-queuep (*queue*) → (length (mapup-queue (*queue*)) = getnth ('2, *queue*))

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

DEFINITION: qfirst2 (*i*, *j*, *qtable*) = qfirst (getnth2 (*i*, *j*, *qtable*))

DEFINITION:
enq2 (*msg*, *i*, *j*, *qtable*) = putnth2 (enq (*msg*, getnth2 (*i*, *j*, *qtable*)), *i*, *j*, *qtable*)

DEFINITION:
deq2 (*i*, *j*, *qtable*) = putnth2 (deq (getnth2 (*i*, *j*, *qtable*)), *i*, *j*, *qtable*)

DEFINITION:
qfullp2 (*i*, *j*, *qtable*, *max*) = qfullp (getnth2 (*i*, *j*, *qtable*), *max*)

DEFINITION: qemptyp2 (*i*, *j*, *qtable*) = qemptyp (getnth2 (*i*, *j*, *qtable*))

THEOREM: array-qemptyp-implies-not-array-qfullp
(array-queuep (*queue*) ∧ array-qemptyp (*queue*))
→ (¬ array-qfullp (*queue*))

THEOREM: array-qfirst-array-enq
(array-queuep (*queue*)
∧ ((¬ array-qemptyp (*queue*)) ∧ (¬ array-qfullp (*queue*))))
→ (array-qfirst (array-enq (*item*, *queue*)) = array-qfirst (*queue*))

EVENT: Disable array-qfirst-array-enq.

THEOREM: array-qfirst-array-enq1
(array-queuep (*queue*) ∧ array-qemptyp (*queue*))
→ (array-qfirst (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* ≃ 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) ≃ 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 $((a + b)$ **mod** `'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-wordsize
(alu-value (tm-alu-decr (a)) $<$ `'65536`) $=$ `'*1*true`

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

THEOREM: lessp-alu-value-tm-alu-decr-wordsize-linear
alu-value (tm-alu-decr (a)) $<$ `'65536`

EVENT: Disable lessp-alu-value-tm-alu-decr-wordsize-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
$((a \not\simeq$ `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-wordsize
(alu-value (tm-alu-decr-mod $(a,\ b)$) $<$ `'65536`) $=$ `'*1*true`

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

THEOREM: lessp-alu-value-tm-alu-decr-mod-wordsize-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
alu-value (tm-alu-decr-mod (a, b)) $\in$ **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))
$\rightarrow$ (alu-value (tm-alu-decr-mod (a, b)) = decr-mod (a, b))

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

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

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

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

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

THEOREM: lessp-alu-value-tm-alu-incr-wordsize-linear
alu-value (tm-alu-incr (a)) < '65536

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

THEOREM: numberp-alu-value-tm-alu-incr
alu-value (tm-alu-incr (a)) $\in$ **N**

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

THEOREM: tm-alu-incr-value-identical-to-ideal-result
((1 + a) < '65536) $\rightarrow$ (alu-value (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
finite-numberp (alu-value (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
(alu-value (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
alu-value (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
alu-value (tm-alu-incr-mod (a, b)) $\in$ **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))
$\rightarrow$ (alu-value (tm-alu-incr-mod (a, b)) = incr-mod (a, b))

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

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

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

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

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

THEOREM: numberp-alu-value-tm-alu-mod
alu-value (tm-alu-mod (a, b)) $\in$ **N**

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

THEOREM: tm-alu-mod-value-identical-to-ideal-result
$((b \not\simeq \texttt{0}) \wedge (b < \texttt{'65536}))$
$\rightarrow \quad (\text{alu-value}\,(\text{tm-alu-mod}\,(a,\ b)) = (a\ \textbf{mod}\ b))$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

THEOREM: numberp-alu-value-tm-alu-plus
alu-value (tm-alu-plus (a, b)) $\in$ **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-wordsize-linear, lessp-alu-value-tm-alu-plus-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-mult-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-mod-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-incr-mod-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-incr-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-decr-mod-wordsize, 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-wordsize-linear, lessp-alu-value-tm-alu-decr-wordsize, 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-rwstate*, 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-REGLENGTH = '8

DEFINITION: TM-MEMLENGTH = '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*),

tm-clock (*tm*),
tm-iports (*tm*),
tm-oports (*tm*))

DEFINITION:
tm-set-cc (*cc*, *tm*)
= tm (tm-memory (*tm*),
tm-regs (*tm*),
cc,
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-error (*error*, *tm*)
= tm (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
error,
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-svcflag (*svcflag*, *tm*)
= tm (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
svcflag,

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-svcid (*svcid*, *tm*)
= tm (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
svcid,
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-base (*base*, *tm*)
= tm (tm-memory (*tm*),
tm-regs (*tm*),
tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
base,
tm-limit (*tm*),
tm-slimit (*tm*),
tm-svmode (*tm*),
tm-rwstate (*tm*),
tm-clock (*tm*),
tm-iports (*tm*),
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,
tm-rwstate (*tm*),
tm-clock (*tm*),
tm-iports (*tm*),
tm-oports (*tm*))

DEFINITION:
tm-set-rwstate (*rwstate*, *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*),
rwstate,
tm-clock (*tm*),
tm-iports (*tm*),
tm-oports (*tm*))

DEFINITION:
tm-set-clock (*clock*, *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*),
clock,
tm-iports (*tm*),
tm-oports (*tm*))

DEFINITION:
tm-set-iports (*iports*, *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*),
iports,
tm-oports (*tm*))

DEFINITION:
tm-set-oports (*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*) ≃ 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)
∧ (finite-numberp (*error*, '64)
∧ (finite-numberp (*svcflag*, '2) ∧ finite-numberp (*svcid*, '128))))
→ 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) ÷ '4)

DEFINITION:
tm-unpack-svcflag (*psw*) = ((*psw* **mod** '512) ÷ '256)

DEFINITION: tm-unpack-svcid (*psw*) = (*psw* ÷ '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) → 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*) ≃ 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*) ≃ 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*) ≃ 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* ≃ 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*) ≃ `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*) ≃ `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*) ≃ `0`)

DEFINITION:
tm-stack-underflowp (*tm*)
= **if** tm-in-supervisor-mode (*tm*) **then** (1 + tm-sp (*tm*)) ≮ tm-slimit (*tm*)
else (1 + tm-sp (*tm*))
≮ min (`'65536` − tm-base (*tm*), tm-limit (*tm*)) **endif**

THEOREM: tm-stack-underflowp-guards-good-address
((¬ tm-stack-underflowp (*tm*)) ∧ finite-numberp (tm-slimit (*tm*), `'65536`))
→ 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-ac
tm)),
tm-set-svcflag (tm-unpack-svcflag (tm-f
tm-set-svcid (tm-unpacl
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-ad
tm)),
tm-set-svcflag (’0,
tm-set-svcid (tm-unpacl
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*) ≃ `0` **then** *tm*
elseif ¬ 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-svcflag (*tm*),
tm-svcid (*tm*)),
tm-incrn-address (`'2`, real-addr (`'2`, *addr*)),
tm)))

DEFINITION:
tm-clock-interruptp (*tm*)
= ((¬ tm-in-supervisor-mode (*tm*)) ∧ (tm-clock (*tm*) ≃ `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*) ≠ `'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-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-interrup
tm-iports (*t*
tm-store-interrupting-input-device (t
t

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-interru
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)) $\wedge$ 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-IPORTS TM))
                   (AND
                    (EQUAL (LENGTH (TM-IPORTS TM)) '16)
                    (AND
                     (GOOD-TM-IPORT-ARRAY (TM-IPORTS TM))
                     (AND
                      (PLISTP (TM-OPORTS TM))
                      (AND
                       (EQUAL (LENGTH (TM-OPORTS TM)) '16)
                       (GOOD-TM-OPORT-ARRAY
                                    (TM-OPORTS TM)))))))))))
 NIL)
```

THEOREM: tm-shellp-tm
good-tm (*tm*) → tm-shellp (*tm*)

EVENT: Disable tm-shellp-tm.

THEOREM: plistp-tm-memory
good-tm (*tm*) → plistp (tm-memory (*tm*))

EVENT: Disable plistp-tm-memory.

THEOREM: length-tm-memory
good-tm (*tm*) → (length (tm-memory (*tm*)) = '65536)

EVENT: Disable length-tm-memory.

THEOREM: finite-number-listp-tm-memory
good-tm (*tm*) → finite-number-listp (tm-memory (*tm*), '65536)

EVENT: Disable finite-number-listp-tm-memory.

THEOREM: numberp-getnth-from-tm-memory
(good-tm (*tm*) ∧ (*i* < '65536)) → (getnth (*i*, tm-memory (*tm*)) ∈ **N**)

EVENT: Disable numberp-getnth-from-tm-memory.

THEOREM: plistp-tm-regs
good-tm (*tm*) → plistp (tm-regs (*tm*))

EVENT: Disable plistp-tm-regs.

THEOREM: length-tm-regs
good-tm (*tm*) → (length (tm-regs (*tm*)) = '8)

EVENT: Disable length-tm-regs.

THEOREM: finite-number-listp-tm-regs
good-tm (*tm*) → finite-number-listp (tm-regs (*tm*), '65536)

EVENT: Disable finite-number-listp-tm-regs.

THEOREM: numberp-getnth-from-tm-regs
(good-tm (*tm*) ∧ (*i* < '8)) → (getnth (*i*, tm-regs (*tm*)) ∈ **N**)

EVENT: Disable numberp-getnth-from-tm-regs.

THEOREM: finite-numberp-tm-cc
good-tm (*tm*) → finite-numberp (tm-cc (*tm*), '4)

EVENT: Disable finite-numberp-tm-cc.

THEOREM: finite-numberp-tm-error
good-tm (*tm*) → finite-numberp (tm-error (*tm*), '64)

EVENT: Disable finite-numberp-tm-error.

THEOREM: finite-numberp-tm-base
good-tm (*tm*) → finite-numberp (tm-base (*tm*), '65536)

EVENT: Disable finite-numberp-tm-base.

THEOREM: finite-numberp-tm-limit
good-tm (*tm*) → finite-numberp (tm-limit (*tm*), '65536)

EVENT: Disable finite-numberp-tm-limit.

THEOREM: finite-numberp-tm-slimit
good-tm (*tm*) → finite-numberp (tm-slimit (*tm*), '65536)

EVENT: Disable finite-numberp-tm-slimit.

THEOREM: finite-numberp-tm-svmode
good-tm (*tm*) → finite-numberp (tm-svmode (*tm*), '2)

EVENT: Disable finite-numberp-tm-svmode.

THEOREM: finite-numberp-tm-clock
good-tm (*tm*) → finite-numberp (tm-clock (*tm*), '65536)

EVENT: Disable finite-numberp-tm-clock.

THEOREM: finite-numberp-tm-pc
good-tm (*tm*) → finite-numberp (tm-pc (*tm*), '65536)

EVENT: Disable finite-numberp-tm-pc.

THEOREM: finite-numberp-tm-sp
good-tm (*tm*) → finite-numberp (tm-sp (*tm*), '65536)

EVENT: Disable finite-numberp-tm-sp.

THEOREM: finite-numberp-tm-pack-psw
good-tm (*tm*)
→ finite-numberp (tm-pack-psw (tm-cc (*tm*),
tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*)),
'65536)

EVENT: Disable finite-numberp-tm-pack-psw.

THEOREM: finite-numberp-tm-svcflag
good-tm (*tm*) → finite-numberp (tm-svcflag (*tm*), '2)

EVENT: Disable finite-numberp-tm-svcflag.

THEOREM: finite-numberp-tm-svcid
good-tm (*tm*) → finite-numberp (tm-svcid (*tm*), '128)

EVENT: Disable finite-numberp-tm-svcid.

THEOREM: finite-numberp-tm-rwstate
good-tm (*tm*) → finite-numberp (tm-rwstate (*tm*), '2)

EVENT: Disable finite-numberp-tm-rwstate.

THEOREM: plistp-tm-iports
good-tm (*tm*) → plistp (tm-iports (*tm*))

EVENT: Disable plistp-tm-iports.

THEOREM: length-tm-iports
good-tm (*tm*) → (length (tm-iports (*tm*)) = '16)

EVENT: Disable length-tm-iports.

THEOREM: good-tm-iport-array-tm-iports
good-tm (*tm*) → good-tm-iport-array (tm-iports (*tm*))

EVENT: Disable good-tm-iport-array-tm-iports.

THEOREM: plistp-tm-oports
good-tm (*tm*) → plistp (tm-oports (*tm*))

EVENT: Disable plistp-tm-oports.

THEOREM: length-tm-oports
good-tm (*tm*) → (length (tm-oports (*tm*)) = '16)

EVENT: Disable length-tm-oports.

THEOREM: good-tm-oport-array-tm-oports
good-tm (*tm*) → good-tm-oport-array (tm-oports (*tm*))

EVENT: Disable good-tm-oport-array-tm-oports.

THEOREM: good-tm-iport-getnth
(good-tm-iport-array (*l*) ∧ (*i* < length (*l*)))
→ good-tm-iport (getnth (*i*, *l*))

EVENT: Disable good-tm-iport-getnth.

THEOREM: good-tm-iport-array-putnth
(good-tm-iport-array (*l*) ∧ good-tm-iport (*iport*))
→ 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*) ∧ finite-numberp (*id*, '16))
→ 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*) ∧ good-tm-iport (*iport*))
→ 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*) ∧ (*i* < length (*l*)))
→ good-tm-oport (getnth (*i*, *l*))

EVENT: Disable good-tm-oport-getnth.

THEOREM: good-tm-oport-array-putnth
(good-tm-oport-array (*l*) ∧ good-tm-oport (*oport*))
→ 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*) ∧ finite-numberp (*id*, '16))
→ 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*) ∧ good-tm-oport (*oport*))
→ 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*) ∧ tm-some-output-interruptp (tm-oports (*tm*)))
→ 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*) ∧ finite-numberp (*id*, '16))
→ (tm-oportp (getnth (*id*, tm-oports (*tm*)))
∧ (finite-numberp (tm-ointerrupt-flag (getnth (*id*, tm-oports (*tm*))),
'2)
∧ (finite-numberp (tm-obusy-flag (getnth (*id*, tm-oports (*tm*))),
'2)
∧ 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*) ∧ finite-numberp (*id*, '16))
→ (plistp (tm-clear-input-interrupt (*id*, tm-iports (*tm*)))
∧ ((length (tm-clear-input-interrupt (*id*, tm-iports (*tm*))) = '16)
∧ 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*) ∧ finite-numberp (*id*, '16))
→ (plistp (tm-clear-output-interrupt (*id*, tm-oports (*tm*)))
∧ ((length (tm-clear-output-interrupt (*id*, tm-oports (*tm*)))
= '16)
∧ 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

(good-tm (*tm*) ∧ finite-numberp (*id*, '16))
→ finite-numberp (tm-ichar (getnth (*id*, tm-iports (*tm*))), '65536)

EVENT: Disable finite-numberp-tm-ichar-getnth-crock.

THEOREM: finite-numberp-tm-overflow-char
finite-numberp (*char*, '256)
→ finite-numberp (tm-overflow-char (*char*), '512)

THEOREM: finite-numberp-tm-overflow-char-tm-wordlub
finite-numberp (*char*, '256)
→ finite-numberp (tm-overflow-char (*char*), '65536)

EVENT: Disable finite-numberp-tm-overflow-char-tm-wordlub.

THEOREM: tm-iport-error-flag-is-zero
(good-tm (*tm*)
∧ ((*id* ∈ **N**)
∧ ((*id* < '16) ∧ (¬ tm-iport-errorp (*id*, tm-iports (*tm*))))))
→ (tm-ierror-flag (getnth (*id*, tm-iports (*tm*))) = '0)

EVENT: Disable tm-iport-error-flag-is-zero.

THEOREM: finite-numberp-tm-interrupting-input-port-crock
(good-tm (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ ((tm-interrupting-input-port (tm-iports (*os*)) ∈ **N**)
∧ (tm-interrupting-input-port (tm-iports (*os*)) < '16))

EVENT: Disable finite-numberp-tm-interrupting-input-port-crock.

THEOREM: tm-ichar-facts
(good-tm (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (((('256
+ tm-ichar (getnth (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))))
< '65536)
= '*1*true)

EVENT: Disable tm-ichar-facts.

THEOREM: finite-numberp-tm-interrupting-output-port-crock
(good-tm (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ ((tm-interrupting-output-port (tm-oports (*os*)) ∈ **N**)
∧ (tm-interrupting-output-port (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*) ∧ finite-number-listp (*b*, *lub*))

EVENT: Disable open-up-finite-number-listp.

THEOREM: finite-number-listp-on-non-list
(¬ listp (*x*)) → (finite-number-listp (*x*, *lub*) = `'*1*true`)

EVENT: Disable finite-number-listp-on-non-list.

THEOREM: length-tm-cpu
good-tm (*tm*) → (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*) → finite-number-listp (tm-cpu (*tm*), `'65536`)

EVENT: Disable finite-number-listp-tm-cpu.

THEOREM: good-cpu-tm-cpu
good-tm (*tm*) → good-cpu (tm-cpu (*tm*))

EVENT: Disable good-cpu-tm-cpu.

THEOREM: good-cpu-getnth
(good-cpu-list (*l*) ∧ (*n* < length (*l*))) → good-cpu (getnth (*n*, *l*))

EVENT: Disable good-cpu-getnth.

THEOREM: good-cpu-list-putnth
(good-cpu-list (*l*) ∧ good-cpu (*cpu*)) → good-cpu-list (putnth (*cpu*, *i*, *l*))

EVENT: Disable good-cpu-list-putnth.

THEOREM: list-of-getnths8
(plistp (*l*) ∧ (length (*l*) = '8))
→ (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
good-tm (*tm*) → (getseg ('0, '8, tm-cpu (*tm*)) = 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-mode, 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
(good-tm (*tm*) ∧ (tm-in-supervisor-mode (*tm*) ∧ finite-numberp (*n*, '65536)))
→ finite-numberp (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
good-tm (*tm*)
→ finite-numberp (min ('65536 − tm-base (*tm*), 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-IPORTS (TM-STORE VALUE ADDR TM))
                                            (TM-IPORTS TM))
                                     (EQUAL (TM-OPORTS (TM-STORE VALUE ADDR TM))
                                            (TM-OPORTS 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-OPORTS (TM-SET-REGS REGS TM))
                                            (TM-OPORTS 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-IPORTS (TM-SET-MEMORY MEMORY TM))
                                    (TM-IPORTS TM))
                             (AND (EQUAL (TM-OPORTS (TM-SET-MEMORY MEMORY TM))
                                         (TM-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*cc*, '4)) → good-tm (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-IPORTS (TM-SET-CC CC TM))
                                         (TM-IPORTS TM))
                                  (AND (EQUAL (TM-OPORTS (TM-SET-CC CC TM))
                                              (TM-OPORTS 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*) ∧ 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-IPORTS (TM-SET-ERROR ERROR TM))
                                      (TM-IPORTS TM))
```

```
                                (AND (EQUAL (TM-OPORTS (TM-SET-ERROR ERROR TM))
                                            (TM-OPORTS 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*) ∧ finite-numberp (*base*, '65536))
→ 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))
  (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-IPORTS (TM-SET-BASE BASE TM))
                                  (TM-IPORTS TM))
                           (AND (EQUAL (TM-OPORTS (TM-SET-BASE BASE TM))
                                       (TM-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*limit*, '65536))
→ good-tm (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-IPORTS (TM-SET-LIMIT LIMIT TM))
                                     (TM-IPORTS 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*) ∧ 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-OPORTS (TM-SET-SLIMIT SLIMIT TM))
                                            (TM-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*svmode*, '2))
→ good-tm (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
(good-tm (*tm*) ∧ finite-numberp (*svcflag*, '2))
→ good-tm (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-IPORTS (TM-SET-SVCFLAG SVCFLAG TM))
                                (TM-IPORTS TM))
                         (AND (EQUAL (TM-OPORTS (TM-SET-SVCFLAG SVCFLAG TM))
                                     (TM-OPORTS 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*) ∧ 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-IPORTS (TM-SET-SVCID SVCID TM))
                                      (TM-IPORTS TM))
                               (AND (EQUAL (TM-OPORTS (TM-SET-SVCID SVCID TM))
                                           (TM-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*rwstate*, '2))
→ good-tm (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-IPORTS (TM-SET-RWSTATE RWSTATE TM))
                                   (TM-IPORTS TM))
                            (AND (EQUAL (TM-OPORTS (TM-SET-RWSTATE RWSTATE TM))
                                        (TM-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*clock*, '65536))
→ good-tm (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-IPORTS (TM-SET-CLOCK CLOCK TM))
                                     (TM-IPORTS 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-iports
(good-tm (*tm*)
 ∧ (plistp (*iports*)
  ∧ ((length (*iports*) = '16) ∧ good-tm-iport-array (*iports*))))
→ good-tm (tm-set-iports (*iports*, *tm*))

EVENT: Disable good-tm-tm-set-iports.

```
(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-iports.

THEOREM: good-tm-tm-set-oports
(good-tm (*tm*)
 ∧ (plistp (*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-OPORTS
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-SET-OPORTS OPORTS TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-SET-OPORTS OPORTS TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-SET-OPORTS OPORTS TM))
           (TM-CC TM))
    (AND
     (EQUAL (TM-ERROR (TM-SET-OPORTS OPORTS TM))
            (TM-ERROR TM))
     (AND
```

```
      (EQUAL (TM-BASE (TM-SET-OPORTS OPORTS TM))
             (TM-BASE TM))
      (AND
       (EQUAL (TM-LIMIT (TM-SET-OPORTS OPORTS TM))
              (TM-LIMIT TM))
       (AND
        (EQUAL (TM-SLIMIT (TM-SET-OPORTS OPORTS TM))
               (TM-SLIMIT TM))
        (AND
         (EQUAL (TM-SVMODE (TM-SET-OPORTS OPORTS TM))
                (TM-SVMODE TM))
         (AND
          (EQUAL (TM-SVCFLAG (TM-SET-OPORTS OPORTS TM))
                 (TM-SVCFLAG TM))
          (AND
              (EQUAL (TM-SVCID (TM-SET-OPORTS OPORTS TM))
                     (TM-SVCID TM))
              (AND (EQUAL (TM-RWSTATE (TM-SET-OPORTS OPORTS TM))
                          (TM-RWSTATE TM))
                   (AND (EQUAL (TM-CLOCK (TM-SET-OPORTS OPORTS TM))
                               (TM-CLOCK TM))
                        (AND (EQUAL (TM-IPORTS (TM-SET-OPORTS OPORTS TM))
                                    (TM-IPORTS TM))
                             (AND (EQUAL (TM-OPORTS (TM-SET-OPORTS OPORTS TM))
                                         OPORTS)
                                  (AND (EQUAL (TM-PC (TM-SET-OPORTS OPORTS TM))
                                              (TM-PC TM))
                                       (EQUAL (TM-SP (TM-SET-OPORTS OPORTS TM))
                                              (TM-SP TM)))))))))))))))
 ((ENABLE TM-SET-OPORTS 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
(good-tm (*tm*) ∧ finite-numberp (*pc*, '65536')) → good-tm (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-IPORTS (TM-SET-PC PC TM))
                                            (TM-IPORTS TM))
                                     (AND (EQUAL (TM-OPORTS (TM-SET-PC PC TM))
                                                 (TM-OPORTS 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-IPORTS (TM-SET-SP SP TM))
                                             (TM-IPORTS TM))
                                      (AND (EQUAL (TM-OPORTS (TM-SET-SP SP TM))
                                                  (TM-OPORTS 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
good-tm (*tm*) → good-tm (tm-execute-add (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-add.

THEOREM: good-tm-tm-execute-branch
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-branch (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-branch.

THEOREM: good-tm-tm-execute-branch-on-zero
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-branch-on-zero (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-branch-on-zero.

THEOREM: good-tm-tm-execute-branch-not-zero
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-branch-not-zero (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-branch-not-zero.

THEOREM: good-tm-tm-execute-call
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-call (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-call.

THEOREM: good-tm-tm-execute-compare
good-tm (*tm*) → good-tm (tm-execute-compare (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-compare.

THEOREM: good-tm-tm-execute-decr
good-tm (*tm*) → good-tm (tm-execute-decr (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-decr.

THEOREM: good-tm-tm-execute-decr-mod
(good-tm (*tm*) ∧ (tm-good-address (*addr1*, *tm*) ∧ tm-good-address (*addr2*, *tm*)))
→ 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*) → good-tm (tm-execute-incr (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-incr.

THEOREM: good-tm-tm-execute-incr-mod
(good-tm (*tm*) ∧ (tm-good-address (*addr1*, *tm*) ∧ tm-good-address (*addr2*, *tm*)))
→ 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*) ∧ tm-good-address (*addr*, *tm*))
→ 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*) ∧ tm-good-address (*addr*, *tm*))
→ 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*) ≃ 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*)
∧ (tm-in-supervisor-mode (*tm*) ∧ tm-good-supervisor-address (*addr*)))
→ 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*) ∧ (¬ tm-stack-underflowp (*tm*)))
→ finite-numberp (1 + tm-sp (*tm*), '65536)

EVENT: Disable finite-numberp-add1-tm-sp.

THEOREM: tm-incr-sp-is-add1-sp
(good-tm (*tm*) ∧ (¬ tm-stack-underflowp (*tm*)))
→ (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*) → good-tm (tm-execute-return (*tm*))

EVENT: Disable good-tm-tm-execute-return.

THEOREM: good-tm-tm-execute-test-iport
good-tm (*tm*) → 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*) → 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*) ∧ tm-good-address (*addr*, *tm*))
→ 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*) → good-tm (tm-execute-run (*tm*))

EVENT: Disable good-tm-tm-execute-run.

THEOREM: good-tm-tm-execute-wait
good-tm (*tm*) → good-tm (tm-execute-wait (*tm*))

EVENT: Disable good-tm-tm-execute-wait.

THEOREM: good-tm-oports-tm-post-output-interrupt
(plistp (*l*) ∧ good-tm-oport-array (*l*))
→ (plistp (tm-post-output-interrupt (*id*, *l*))
∧ ((length (tm-post-output-interrupt (*id*, *l*)) = length (*l*))
∧ good-tm-oport-array (tm-post-output-interrupt (*id*, *l*))))

EVENT: Disable good-tm-oports-tm-post-output-interrupt.

THEOREM: good-tm-tm-execute-post-output-interrupt
good-tm (*tm*) → good-tm (tm-execute-post-output-interrupt (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-post-output-interrupt.

THEOREM: good-tm-oports-tm-start-output
(plistp (*l*) ∧ good-tm-oport-array (*l*))
→ (plistp (tm-start-output (*char*, *id*, *l*))
∧ ((length (tm-start-output (*char*, *id*, *l*)) = length (*l*))
∧ good-tm-oport-array (tm-start-output (*char*, *id*, *l*))))

EVENT: Disable good-tm-oports-tm-start-output.

THEOREM: good-tm-tm-execute-start-output
good-tm (*tm*) → good-tm (tm-execute-start-output (*addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-start-output.

THEOREM: good-tm-tm-execute-svc
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-svc (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-svc.

THEOREM: good-tm-tm-execute-svc-return
(good-tm (*tm*) ∧ tm-good-address (*addr*, *tm*))
→ good-tm (tm-execute-svc-return (*addr*, *tm*))

EVENT: Disable good-tm-tm-execute-svc-return.

THEOREM: good-tm-tm-execute-nullary
good-tm (*tm*) → good-tm (tm-execute-nullary (*opcode*, *tm*))

EVENT: Disable good-tm-tm-execute-nullary.

THEOREM: good-tm-tm-execute-unary
good-tm (*tm*) → good-tm (tm-execute-unary (*opcode*, *addr*, *tm*))

EVENT: Disable good-tm-tm-execute-unary.

THEOREM: good-tm-tm-execute-binary
good-tm (*tm*) → good-tm (tm-execute-binary (*opcode*, *addr1*, *addr2*, *tm*))

EVENT: Disable good-tm-tm-execute-binary.

THEOREM: good-tm-tm-execute-instruction
good-tm (*tm*) → good-tm (tm-execute-instruction (*instruction*, *tm*))

EVENT: Disable good-tm-tm-execute-instruction.

THEOREM: plus-tm-base-tm-limit-tm-set-pc
(tm-base (tm-set-pc (*pc*, *tm*)) + tm-limit (tm-set-pc (*pc*, *tm*)))
= (tm-base (*tm*) + tm-limit (*tm*))

EVENT: Disable plus-tm-base-tm-limit-tm-set-pc.

THEOREM: finite-numberp-tm-good-address
(good-tm (*tm*) ∧ tm-good-address (real-addr ('2, *n*), *tm*))
→ finite-numberp (*n*, '65536)

EVENT: Disable finite-numberp-tm-good-address.

THEOREM: good-tm-tm-execute
good-tm (*tm*) → good-tm (tm-execute (*instruction*, *tm*))

EVENT: Disable good-tm-tm-execute.

THEOREM: good-tm-tm-decrement-clock
good-tm (*tm*) → good-tm (tm-decrement-clock (*tm*))

EVENT: Disable good-tm-tm-decrement-clock.

THEOREM: plus-tm-base-tm-limit-tm-decrement-clock
(tm-base (tm-decrement-clock (*tm*)) + tm-limit (tm-decrement-clock (*tm*)))
= (tm-base (*tm*) + 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-IPORTS (TM-DECREMENT-CLOCK TM))
                                      (TM-IPORTS 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
tm-good-address (*addr*, tm-decrement-clock (*tm*)) = tm-good-address (*addr*, *tm*)

EVENT: Disable tm-good-address-tm-decrement-clock.

THEOREM: good-tm-tm-fetch-execute
good-tm (*tm*) → good-tm (tm-fetch-execute (*tm*))

EVENT: Disable good-tm-tm-fetch-execute.

THEOREM: tm-good-address-tm-store
tm-good-address (*addr*, tm-store (*value*, *addr*, *tm*)) = tm-good-address (*addr*, *tm*)

EVENT: Disable tm-good-address-tm-store.

THEOREM: good-tm-tm-execute-clock-interrupt
good-tm (*tm*) → good-tm (tm-execute-clock-interrupt (*tm*))

EVENT: Disable good-tm-tm-execute-clock-interrupt.

THEOREM: good-tm-tm-execute-error-interrupt
good-tm (*tm*) → good-tm (tm-execute-error-interrupt (*tm*))

EVENT: Disable good-tm-tm-execute-error-interrupt.

THEOREM: good-tm-tm-execute-svc-interrupt
good-tm (*tm*) → good-tm (tm-execute-svc-interrupt (*tm*))

EVENT: Disable good-tm-tm-execute-svc-interrupt.

THEOREM: good-tm-tm-execute-input-interrupt
(good-tm (*tm*) ∧ tm-some-input-interruptp (tm-iports (*tm*)))
→ good-tm (tm-execute-input-interrupt (*tm*))

EVENT: Disable good-tm-tm-execute-input-interrupt.

THEOREM: good-tm-tm-execute-output-interrupt
(good-tm (*tm*) ∧ tm-some-output-interruptp (tm-oports (*tm*)))
→ good-tm (tm-execute-output-interrupt (*tm*))

EVENT: Disable good-tm-tm-execute-output-interrupt.

THEOREM: good-tm-tm-step
good-tm (*tm*) → good-tm (tm-step (*tm*))

EVENT: Disable good-tm-tm-step.

THEOREM: good-tm-iports-tm-post-input-interrupt
(plistp (*l*)
∧ (finite-numberp (*id*, length (*l*))
∧ (finite-numberp (*char*, '256) ∧ good-tm-iport-array (*l*))))
→ (plistp (tm-post-input-interrupt (*char*, *id*, *l*))
∧ ((length (tm-post-input-interrupt (*char*, *id*, *l*)) = length (*l*))
∧ good-tm-iport-array (tm-post-input-interrupt (*char*, *id*, *l*))))

EVENT: Disable good-tm-iports-tm-post-input-interrupt.

THEOREM: good-tm-tm-post-interrupt
good-tm (*tm*) → good-tm (tm-post-interrupt (*event*, *tm*))

EVENT: Disable good-tm-tm-post-interrupt.

THEOREM: good-tm-tm-processor
good-tm (*tm*) → good-tm (tm-processor (*tm*, *n*))

EVENT: Disable good-tm-tm-processor.

THEOREM: registers-untouched-by-tm-execute-add
(tm-error (tm-execute-add (*addr1*, *addr2*, *tm*)) = tm-error (*tm*))
∧ ((tm-base (tm-execute-add (*addr1*, *addr2*, *tm*)) = tm-base (*tm*))
∧ ((tm-limit (tm-execute-add (*addr1*, *addr2*, *tm*)) = tm-limit (*tm*))
∧ ((tm-slimit (tm-execute-add (*addr1*, *addr2*, *tm*))
= tm-slimit (*tm*))
∧ ((tm-svmode (tm-execute-add (*addr1*, *addr2*, *tm*))
= tm-svmode (*tm*))
∧ ((tm-svcflag (tm-execute-add (*addr1*,
addr2,
tm))

$$
\begin{array}{l}
= \quad \text{tm-svcflag}\,(tm)) \\
\wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-add}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad = \quad \text{tm-svcid}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-add}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad = \quad \text{tm-rwstate}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-add}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad = \quad \text{tm-clock}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-add}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-add}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(tm)))))))))))
\end{array}
$$

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-IPORTS (TM-EXECUTE-BRANCH ADDR TM))
                                   (TM-IPORTS TM))
                            (AND (EQUAL (TM-OPORTS (TM-EXECUTE-BRANCH ADDR TM))
                                        (TM-OPORTS 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))
     (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-IPORTS (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
                           (TM-IPORTS TM))
                    (AND (EQUAL (TM-OPORTS (TM-EXECUTE-BRANCH-ON-ZERO ADDR TM))
                                (TM-OPORTS 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-IPORTS (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
                         (TM-IPORTS TM))
                  (AND (EQUAL (TM-OPORTS (TM-EXECUTE-BRANCH-NOT-ZERO ADDR TM))
                              (TM-OPORTS 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
$(\text{tm-cc}\,(\text{tm-execute-call}\,(addr,\ tm)) = \text{tm-cc}\,(tm))$
$\wedge \quad ((\text{tm-base}\,(\text{tm-execute-call}\,(addr,\ tm)) = \text{tm-base}\,(tm))$
$\quad \wedge \quad ((\text{tm-limit}\,(\text{tm-execute-call}\,(addr,\ tm)) = \text{tm-limit}\,(tm))$

$$
\begin{array}{l}
\wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-call}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad = \quad \text{tm-slimit}\,(\mathit{tm})) \\
\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-call}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad\qquad = \quad \text{tm-svmode}\,(\mathit{tm})) \\
\qquad\qquad \wedge \quad ((\text{tm-svcflag}\,(\text{tm-execute-call}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad\qquad\qquad = \quad \text{tm-svcflag}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-call}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svcid}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-call}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-rwstate}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-call}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-clock}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-call}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-call}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(\mathit{tm}))))))))))
\end{array}
$$

EVENT: Disable registers-untouched-by-tm-execute-call.

```
(PROVE-LEMMA
 REGISTERS-UNTOUCHED-BY-TM-EXECUTE-COMPARE
 (REWRITE)
 (AND
  (EQUAL (TM-MEMORY (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-ERROR (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
           (TM-ERROR TM))
    (AND
     (EQUAL (TM-BASE (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
            (TM-BASE TM))
     (AND
      (EQUAL (TM-LIMIT (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
             (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-IPORTS (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
                          (TM-IPORTS TM))
                   (AND (EQUAL (TM-OPORTS (TM-EXECUTE-COMPARE ADDR1 ADDR2 TM))
                               (TM-OPORTS 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

$$
\begin{array}{l}
(\text{tm-error}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) = \text{tm-error}\,(\mathit{tm})) \\
\wedge\quad ((\text{tm-base}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) = \text{tm-base}\,(\mathit{tm})) \\
\qquad \wedge\quad ((\text{tm-limit}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) = \text{tm-limit}\,(\mathit{tm})) \\
\qquad\qquad \wedge\quad ((\text{tm-slimit}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad\qquad\qquad =\quad \text{tm-slimit}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge\quad ((\text{tm-svmode}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad =\quad \text{tm-svmode}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-svcflag}\,(\text{tm-execute-decr}\,(\mathit{addr},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-svcflag}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-svcid}\,(\text{tm-execute-decr}\,(\mathit{addr}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm}))
\end{array}
$$

$$
\begin{array}{l}
\quad = \quad \text{tm-svcid}\,(tm)) \\
\wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-decr}\,(addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad = \quad \text{tm-rwstate}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-decr}\,(addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad = \quad \text{tm-clock}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-decr}\,(addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad = \quad \text{tm-iports}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-decr}\,(addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(tm)))))))))))
\end{array}
$$

EVENT: Disable registers-untouched-by-tm-execute-decr.

THEOREM: registers-untouched-by-tm-execute-decr-mod

$$
\begin{array}{l}
(\text{tm-error}\,(\text{tm-execute-decr-mod}\,(addr1,\ addr2,\ tm)) = \text{tm-error}\,(tm)) \\
\wedge \quad ((\text{tm-base}\,(\text{tm-execute-decr-mod}\,(addr1,\ addr2,\ tm)) = \text{tm-base}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-limit}\,(\text{tm-execute-decr-mod}\,(addr1,\ addr2,\ tm)) \\
\qquad\qquad = \quad \text{tm-limit}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-decr-mod}\,(addr1,\ addr2,\ tm)) \\
\qquad\qquad\qquad = \quad \text{tm-slimit}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-decr-mod}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svmode}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcflag}\,(\text{tm-execute-decr-mod}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcflag}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-decr-mod}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcid}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-decr-mod}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-rwstate}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-decr-mod}\,(addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-clock}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-decr-mod}\,(addr1,
\end{array}
$$

$addr2$,
tm))
= tm-iports (tm))
∧ (tm-oports (tm-execute-decr-mod ($addr1$, $addr2$, tm))
= tm-oports (tm)))))))))))

EVENT: Disable registers-untouched-by-tm-execute-decr-mod.

THEOREM: registers-untouched-by-tm-execute-incr
(tm-error (tm-execute-incr ($addr$, tm)) = tm-error (tm))
∧ ((tm-base (tm-execute-incr ($addr$, tm)) = tm-base (tm))
∧ ((tm-limit (tm-execute-incr ($addr$, tm)) = tm-limit (tm))
∧ ((tm-slimit (tm-execute-incr ($addr$, tm))
= tm-slimit (tm))
∧ ((tm-svmode (tm-execute-incr ($addr$, tm))
= tm-svmode (tm))
∧ ((tm-svcflag (tm-execute-incr ($addr$, tm))
= tm-svcflag (tm))
∧ ((tm-svcid (tm-execute-incr ($addr$,
tm))
= tm-svcid (tm))
∧ ((tm-rwstate (tm-execute-incr ($addr$,
tm))
= tm-rwstate (tm))
∧ ((tm-clock (tm-execute-incr ($addr$,
tm))
= tm-clock (tm))
∧ ((tm-iports (tm-execute-incr ($addr$,
tm))
= tm-iports (tm))
∧ (tm-oports (tm-execute-incr ($addr$,
tm))
= tm-oports (tm)))))))))))

EVENT: Disable registers-untouched-by-tm-execute-incr.

THEOREM: registers-untouched-by-tm-execute-incr-mod
(tm-error (tm-execute-incr-mod ($addr1$, $addr2$, tm)) = tm-error (tm))
∧ ((tm-base (tm-execute-incr-mod ($addr1$, $addr2$, tm)) = tm-base (tm))
∧ ((tm-limit (tm-execute-incr-mod ($addr1$, $addr2$, tm))
= tm-limit (tm))

$$
\begin{array}{l}
\wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad = \quad \text{tm-slimit}\,(\mathit{tm})) \\
\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad = \quad \text{tm-svmode}\,(\mathit{tm})) \\
\qquad\qquad \wedge \quad ((\text{tm-svcflag}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad = \quad \text{tm-svcflag}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svcid}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-rwstate}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-clock}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(\mathit{tm}))))))))))
\end{array}
$$

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-IPORTS (TM-EXECUTE-LOAD-BASE ADDR TM))
                               (TM-IPORTS TM))
                        (AND (EQUAL (TM-OPORTS (TM-EXECUTE-LOAD-BASE ADDR TM))
                                    (TM-OPORTS 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-IPORTS (TM-EXECUTE-LOAD-LIMIT ADDR TM))
                              (TM-IPORTS 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-IPORTS (TM-EXECUTE-LOAD-PSW ADDR TM))
                                 (TM-IPORTS TM))
                          (AND (EQUAL (TM-OPORTS (TM-EXECUTE-LOAD-PSW ADDR TM))
                                      (TM-OPORTS 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

$$
\begin{array}{l}
(\text{tm-error}\,(\text{tm-execute-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) = \text{tm-error}\,(\mathit{tm})) \\
\wedge\ ((\text{tm-base}\,(\text{tm-execute-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) = \text{tm-base}\,(\mathit{tm})) \\
\quad \wedge\ ((\text{tm-limit}\,(\text{tm-execute-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) = \text{tm-limit}\,(\mathit{tm})) \\
\qquad \wedge\ ((\text{tm-slimit}\,(\text{tm-execute-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad\quad = \text{tm-slimit}\,(\mathit{tm})) \\
\qquad\quad \wedge\ ((\text{tm-svmode}\,(\text{tm-execute-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad\qquad = \text{tm-svmode}\,(\mathit{tm})) \\
\qquad\qquad \wedge\ ((\text{tm-svcflag}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\quad = \text{tm-svcflag}\,(\mathit{tm})) \\
\qquad\qquad\quad \wedge\ ((\text{tm-svcid}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad = \text{tm-svcid}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge\ ((\text{tm-rwstate}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\quad = \text{tm-rwstate}\,(\mathit{tm})) \\
\qquad\qquad\qquad\quad \wedge\ ((\text{tm-clock}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad = \text{tm-clock}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad \wedge\ ((\text{tm-iports}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad = \text{tm-iports}\,(\mathit{tm}))
\end{array}
$$

$$\wedge \quad (\text{tm-oports}\,(\text{tm-execute-mod}\,(\mathit{addr1}, \mathit{addr2}, \mathit{tm})) = \text{tm-oports}\,(\mathit{tm})))))))))))$$

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-IPORTS (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
                                   (TM-IPORTS TM))
                            (EQUAL (TM-OPORTS (TM-EXECUTE-MOVE ADDR1 ADDR2 TM))
                                   (TM-OPORTS 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*))
$\land$ ((tm-base (tm-execute-multiply (*addr1*, *addr2*, *tm*)) = tm-base (*tm*))
$\land$ ((tm-limit (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-limit (*tm*))
$\land$ ((tm-slimit (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-slimit (*tm*))
$\land$ ((tm-svmode (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-svmode (*tm*))
$\land$ ((tm-svcflag (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-svcflag (*tm*))
$\land$ ((tm-svcid (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-svcid (*tm*))
$\land$ ((tm-rwstate (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-rwstate (*tm*))
$\land$ ((tm-clock (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-clock (*tm*))
$\land$ ((tm-iports (tm-execute-multiply (*addr1*, *addr2*, *tm*))
= tm-iports (*tm*))
$\land$ (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

$$
\begin{array}{l}
(\text{tm-memory}\,(\text{tm-execute-return}\,(tm)) = \text{tm-memory}\,(tm)) \\
\wedge \quad ((\text{tm-cc}\,(\text{tm-execute-return}\,(tm)) = \text{tm-cc}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-base}\,(\text{tm-execute-return}\,(tm)) = \text{tm-base}\,(tm)) \\
\qquad\quad \wedge \quad ((\text{tm-limit}\,(\text{tm-execute-return}\,(tm)) = \text{tm-limit}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad = \quad \text{tm-slimit}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svmode}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcflag}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcflag}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcid}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-rwstate}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-clock}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-return}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(tm))))))))))))
\end{array}
$$

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-IPORTS (TM-EXECUTE-SET-CLOCK ADDR TM))
                              (TM-IPORTS TM))
                       (AND (EQUAL (TM-OPORTS (TM-EXECUTE-SET-CLOCK ADDR TM))
                                   (TM-OPORTS 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-IPORTS (TM-EXECUTE-RUN TM))
                                         (TM-IPORTS TM))
                                  (AND (EQUAL (TM-OPORTS (TM-EXECUTE-RUN TM))
                                              (TM-OPORTS 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-IPORTS (TM-EXECUTE-WAIT TM))
                                        (TM-IPORTS TM))
                                 (AND (EQUAL (TM-OPORTS (TM-EXECUTE-WAIT TM))
                                             (TM-OPORTS 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-IPORT ADDR TM))
         (TM-MEMORY TM))
  (AND
   (EQUAL (TM-REGS (TM-EXECUTE-TEST-IPORT ADDR TM))
          (TM-REGS TM))
   (AND
    (EQUAL (TM-CC (TM-EXECUTE-TEST-IPORT ADDR TM))
           (TM-CC TM))
    (AND
     (EQUAL (TM-BASE (TM-EXECUTE-TEST-IPORT ADDR TM))
            (TM-BASE TM))
     (AND
      (EQUAL (TM-LIMIT (TM-EXECUTE-TEST-IPORT ADDR TM))
             (TM-LIMIT TM))
      (AND
       (EQUAL (TM-SLIMIT (TM-EXECUTE-TEST-IPORT ADDR TM))
              (TM-SLIMIT TM))
       (AND
        (EQUAL (TM-SVMODE (TM-EXECUTE-TEST-IPORT ADDR TM))
               (TM-SVMODE TM))
        (AND
         (EQUAL (TM-SVCFLAG (TM-EXECUTE-TEST-IPORT ADDR TM))
                (TM-SVCFLAG TM))
         (AND
          (EQUAL (TM-SVCID (TM-EXECUTE-TEST-IPORT ADDR TM))
                 (TM-SVCID TM))
          (AND
            (EQUAL (TM-RWSTATE (TM-EXECUTE-TEST-IPORT ADDR TM))
                   (TM-RWSTATE TM))
            (AND (EQUAL (TM-CLOCK (TM-EXECUTE-TEST-IPORT ADDR TM))
                        (TM-CLOCK TM))
                 (AND (EQUAL (TM-IPORTS (TM-EXECUTE-TEST-IPORT ADDR TM))
                             (TM-IPORTS TM))
                      (AND (EQUAL (TM-OPORTS (TM-EXECUTE-TEST-IPORT ADDR TM))
                                  (TM-OPORTS TM))
                           (AND (EQUAL (TM-PC (TM-EXECUTE-TEST-IPORT ADDR TM))
                                       (TM-PC TM))
                                (EQUAL (TM-SP (TM-EXECUTE-TEST-IPORT ADDR TM))
                                       (TM-SP TM)))))))))))))))))
((ENABLE TM-EXECUTE-TEST-IPORT 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-IPORTS (TM-EXECUTE-TEST-OPORT ADDR TM))
                              (TM-IPORTS TM))
```

```
                        (AND (EQUAL (TM-OPORTS (TM-EXECUTE-TEST-OPORT ADDR TM))
                                    (TM-OPORTS TM))
                             (AND (EQUAL (TM-PC (TM-EXECUTE-TEST-OPORT ADDR TM))
                                         (TM-PC TM))
                                  (EQUAL (TM-SP (TM-EXECUTE-TEST-OPORT ADDR TM))
                                         (TM-SP TM))))))))))))))))))
 ((ENABLE TM-EXECUTE-TEST-OPORT 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
```

```
           (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-IPORTS (TM-EXECUTE-START-OUTPUT ADDR1
                                                                    ADDR2 TM))
                                (TM-IPORTS TM))
                         (AND (EQUAL (TM-OPORTS (TM-EXECUTE-START-OUTPUT ADDR1
                                                                         ADDR2
                                                                         TM))
                                     (TM-OPORTS 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-IPORTS (TM-EXECUTE-POST-OUTPUT-INTERRUPT ADDR TM))
                   (TM-IPORTS 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-IPORTS (TM-EXECUTE-SVC ADDR TM))
                                     (TM-IPORTS TM))
                              (AND (EQUAL (TM-OPORTS (TM-EXECUTE-SVC ADDR TM))
                                          (TM-OPORTS 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-IPORTS (TM-EXECUTE-SVC-RETURN ADDR TM))
                              (TM-IPORTS 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))
                                 (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

$$
\begin{array}{l}
(\neg\ \text{tm-in-supervisor-mode}\,(tm)) \\
\rightarrow\quad ((\text{tm-base}\,(\text{tm-execute-nullary}\,(opcode,\ tm)) = \text{tm-base}\,(tm)) \\
\qquad \wedge\quad ((\text{tm-limit}\,(\text{tm-execute-nullary}\,(opcode,\ tm)) = \text{tm-limit}\,(tm)) \\
\qquad\qquad \wedge\quad ((\text{tm-slimit}\,(\text{tm-execute-nullary}\,(opcode,\ tm)) \\
\qquad\qquad\qquad =\quad \text{tm-slimit}\,(tm)) \\
\qquad\qquad\qquad \wedge\quad ((\text{tm-svmode}\,(\text{tm-execute-nullary}\,(opcode,\ tm)) \\
\qquad\qquad\qquad\qquad =\quad \text{tm-svmode}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-svcflag}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-svcflag}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-svcid}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-svcid}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-rwstate}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-rwstate}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-clock}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-clock}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad ((\text{tm-iports}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-iports}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge\quad (\text{tm-oports}\,(\text{tm-execute-nullary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad =\quad \text{tm-oports}\,(tm))))))))))).
\end{array}
$$

EVENT: Disable registers-untouched-by-tm-execute-nullary.

THEOREM: registers-untouched-by-tm-execute-unary

$$
\begin{array}{l}
(\neg\ \text{tm-in-supervisor-mode}\,(tm)) \\
\rightarrow\quad ((\text{tm-base}\,(\text{tm-execute-unary}\,(opcode,\ addr,\ tm)) = \text{tm-base}\,(tm)) \\
\qquad \wedge\quad ((\text{tm-limit}\,(\text{tm-execute-unary}\,(opcode,\ addr,\ tm)) = \text{tm-limit}\,(tm)) \\
\qquad\qquad \wedge\quad ((\text{tm-slimit}\,(\text{tm-execute-unary}\,(opcode,\ addr,\ tm))
\end{array}
$$

$$
\begin{array}{l}
\quad = \quad \text{tm-slimit}\,(tm)) \\
\wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-unary}\,(opcode,\ addr,\ tm)) \\
\qquad = \quad \text{tm-svmode}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-unary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad = \quad \text{tm-rwstate}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-unary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad = \quad \text{tm-clock}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-unary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-unary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(tm)))))))))
\end{array}
$$

EVENT: Disable registers-untouched-by-tm-execute-unary.

THEOREM: registers-untouched-by-tm-execute-binary

$$
\begin{array}{l}
(\neg\ \text{tm-in-supervisor-mode}\,(tm)) \\
\rightarrow \quad ((\text{tm-base}\,(\text{tm-execute-binary}\,(opcode,\ addr1,\ addr2,\ tm)) = \text{tm-base}\,(tm)) \\
\qquad \wedge \quad ((\text{tm-limit}\,(\text{tm-execute-binary}\,(opcode,\ addr1,\ addr2,\ tm)) \\
\qquad\qquad = \quad \text{tm-limit}\,(tm)) \\
\qquad\qquad \wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-binary}\,(opcode,\ addr1,\ addr2,\ tm)) \\
\qquad\qquad\qquad = \quad \text{tm-slimit}\,(tm)) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-binary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svmode}\,(tm)) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcflag}\,(\text{tm-execute-binary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcflag}\,(tm)) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-svcid}\,(\text{tm-execute-binary}\,(opcode, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr1, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad addr2, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad tm)) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-svcid}\,(tm))
\end{array}
$$

$$
\begin{array}{l}
\wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-binary}\,(\mathit{opcode},\ \mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad = \quad \text{tm-rwstate}\,(\mathit{tm})) \\
\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-binary}\,(\mathit{opcode},\ \mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad\qquad = \quad \text{tm-clock}\,(\mathit{tm})) \\
\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-binary}\,(\mathit{opcode},\ \mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad\qquad\qquad = \quad \text{tm-iports}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-binary}\,(\mathit{opcode},\ \mathit{addr1},\ \mathit{addr2},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-oports}\,(\mathit{tm}))))))))))
\end{array}
$$

EVENT: Disable registers-untouched-by-tm-execute-binary.

THEOREM: registers-untouched-by-tm-execute-instruction

$$
\begin{array}{l}
(\neg\ \text{tm-in-supervisor-mode}\,(\mathit{tm})) \\
\rightarrow \quad ((\text{tm-base}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) = \text{tm-base}\,(\mathit{tm})) \\
\qquad \wedge \quad ((\text{tm-limit}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad = \quad \text{tm-limit}\,(\mathit{tm})) \\
\qquad\qquad \wedge \quad ((\text{tm-slimit}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad\qquad = \quad \text{tm-slimit}\,(\mathit{tm})) \\
\qquad\qquad\qquad \wedge \quad ((\text{tm-svmode}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad = \quad \text{tm-svmode}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-rwstate}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-clock}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-clock}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad ((\text{tm-iports}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{tm-iports}\,(\mathit{tm})) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{tm-oports}\,(\text{tm-execute-instruction}\,(\mathit{instruction},\ \mathit{tm}))
\end{array}
$$

$$= \quad \text{tm-oports}\,(tm)))))))))$$

EVENT: Disable registers-untouched-by-tm-execute-instruction.

THEOREM: registers-untouched-by-tm-execute
$$(\neg\ \text{tm-in-supervisor-mode}\,(tm))$$
$$\rightarrow \quad ((\text{tm-base}\,(\text{tm-execute}\,(opcode,\ tm)) = \text{tm-base}\,(tm))$$
$$\wedge \quad ((\text{tm-limit}\,(\text{tm-execute}\,(opcode,\ tm)) = \text{tm-limit}\,(tm))$$
$$\wedge \quad ((\text{tm-slimit}\,(\text{tm-execute}\,(opcode,\ tm)) = \text{tm-slimit}\,(tm))$$
$$\wedge \quad ((\text{tm-svmode}\,(\text{tm-execute}\,(opcode,\ tm))$$
$$= \quad \text{tm-svmode}\,(tm))$$
$$\wedge \quad ((\text{tm-rwstate}\,(\text{tm-execute}\,(opcode,\ tm))$$
$$= \quad \text{tm-rwstate}\,(tm))$$
$$\wedge \quad ((\text{tm-clock}\,(\text{tm-execute}\,(opcode,\ tm))$$
$$= \quad \text{tm-clock}\,(tm))$$
$$\wedge \quad ((\text{tm-iports}\,(\text{tm-execute}\,(opcode,\ tm))$$
$$= \quad \text{tm-iports}\,(tm))$$
$$\wedge \quad (\text{tm-oports}\,(\text{tm-execute}\,(opcode,\ tm))$$
$$= \quad \text{tm-oports}\,(tm)))))))))$$

EVENT: Disable registers-untouched-by-tm-execute.

THEOREM: registers-untouched-by-tm-fetch-execute
$$(\neg\ \text{tm-in-supervisor-mode}\,(tm))$$
$$\rightarrow \quad ((\text{tm-base}\,(\text{tm-fetch-execute}\,(tm)) = \text{tm-base}\,(tm))$$
$$\wedge \quad ((\text{tm-limit}\,(\text{tm-fetch-execute}\,(tm)) = \text{tm-limit}\,(tm))$$
$$\wedge \quad ((\text{tm-slimit}\,(\text{tm-fetch-execute}\,(tm)) = \text{tm-slimit}\,(tm))$$
$$\wedge \quad ((\text{tm-svmode}\,(\text{tm-fetch-execute}\,(tm))$$
$$= \quad \text{tm-svmode}\,(tm))$$
$$\wedge \quad ((\text{tm-rwstate}\,(\text{tm-fetch-execute}\,(tm))$$
$$= \quad \text{tm-rwstate}\,(tm))$$
$$\wedge \quad ((\text{tm-iports}\,(\text{tm-fetch-execute}\,(tm))$$
$$= \quad \text{tm-iports}\,(tm))$$
$$\wedge \quad (\text{tm-oports}\,(\text{tm-fetch-execute}\,(tm))$$
$$= \quad \text{tm-oports}\,(tm)))))))$$

EVENT: Disable registers-untouched-by-tm-fetch-execute.

THEOREM: tm-clock-tm-fetch-execute
$$((\neg\ \text{tm-in-supervisor-mode}\,(tm)) \wedge (\neg\ \text{tm-clock-interruptp}\,(tm)))$$
$$\rightarrow \quad (\text{tm-clock}\,(\text{tm-fetch-execute}\,(tm)) = (\text{tm-clock}\,(tm) - 1))$$

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
(disjoint-everywhere (b, l, *table*) $\wedge$ ($i <$ length (*table*)))
$\rightarrow$ disjoint (b, l, base (getnth (i, *table*)), limit (getnth (i, *table*)))

EVENT: Disable disjoint-getnth.

THEOREM: tm-good-address-tm-set-cc
tm-good-address (*addr*, tm-set-cc (*cc*, *tm*)) = tm-good-address (*addr*, *tm*)

EVENT: Disable tm-good-address-tm-set-cc.

THEOREM: tm-good-address-tm-set-pc
tm-good-address (*addr*, tm-set-pc (*pc*, *tm*)) = tm-good-address (*addr*, *tm*)

EVENT: Disable tm-good-address-tm-set-pc.

THEOREM: tm-good-address-tm-set-sp
tm-good-address (*addr*, tm-set-sp (*sp*, *tm*)) = tm-good-address (*addr*, *tm*)

EVENT: Disable tm-good-address-tm-set-sp.

THEOREM: tm-store-in-memory-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
$\wedge$ (disjoint (b, l, tm-base (*tm*), tm-limit (*tm*))
$\wedge$ finite-numberp (*addr*,
min (`'65536` − tm-base (*tm*), tm-limit (*tm*)))))
$\rightarrow$ (getseg (b, l, tm-memory (tm-store-in-memory (*value*, *addr*, *tm*)))
= getseg (b, l, tm-memory (*tm*)))

EVENT: Disable tm-store-in-memory-protection-theorem.

THEOREM: tm-store-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
$\wedge$ (disjoint (b, l, tm-base (*tm*), tm-limit (*tm*)) $\wedge$ tm-good-address (*addr*, *tm*)))
$\rightarrow$ (getseg (b, l, tm-memory (tm-store (*value*, *addr*, *tm*)))
= getseg (b, l, tm-memory (*tm*)))

EVENT: Disable tm-store-protection-theorem.

THEOREM: tm-execute-add-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
$\wedge$ (disjoint (b, l, tm-base (*tm*), tm-limit (*tm*))

$\wedge$ tm-good-address ($addr1$, tm)))
$\rightarrow$ (getseg (b, l, tm-memory (tm-execute-add ($addr1$, $addr2$, tm)))
= getseg (b, l, tm-memory (tm)))

EVENT: Disable tm-execute-add-protection-theorem.

THEOREM: tm-execute-branch-protection-theorem
tm-memory (tm-execute-branch ($addr$, tm)) = tm-memory (tm)

EVENT: Disable tm-execute-branch-protection-theorem.

THEOREM: tm-execute-branch-on-zero-protection-theorem
tm-memory (tm-execute-branch-on-zero ($addr$, tm)) = tm-memory (tm)

EVENT: Disable tm-execute-branch-on-zero-protection-theorem.

THEOREM: tm-execute-branch-not-zero-protection-theorem
tm-memory (tm-execute-branch-not-zero ($addr$, tm)) = tm-memory (tm)

EVENT: Disable tm-execute-branch-not-zero-protection-theorem.

THEOREM: tm-execute-call-protection-theorem
((¬ tm-in-supervisor-mode (tm))
$\wedge$ (disjoint (b, l, tm-base (tm), tm-limit (tm)) $\wedge$ tm-good-address ($addr$, tm)))
$\rightarrow$ (getseg (b, l, tm-memory (tm-execute-call ($addr$, tm)))
= getseg (b, l, tm-memory (tm)))

EVENT: Disable tm-execute-call-protection-theorem.

THEOREM: tm-execute-compare-protection-theorem
tm-memory (tm-execute-compare ($addr1$, $addr2$, tm)) = tm-memory (tm)

EVENT: Disable tm-execute-compare-protection-theorem.

THEOREM: tm-execute-decr-protection-theorem
((¬ tm-in-supervisor-mode (tm))
$\wedge$ (disjoint (b, l, tm-base (tm), tm-limit (tm)) $\wedge$ tm-good-address ($addr$, tm)))
$\rightarrow$ (getseg (b, l, tm-memory (tm-execute-decr ($addr$, tm)))
= getseg (b, l, tm-memory (tm)))

EVENT: Disable tm-execute-decr-protection-theorem.

THEOREM: tm-execute-decr-mod-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*))
  ∧ tm-good-address (*addr1*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-decr-mod (*addr1*, *addr2*, *tm*)))
  = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-decr-mod-protection-theorem.

THEOREM: tm-execute-incr-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)) ∧ tm-good-address (*addr*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-incr (*addr*, *tm*)))
  = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-incr-protection-theorem.

THEOREM: tm-execute-incr-mod-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*))
  ∧ tm-good-address (*addr1*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-incr-mod (*addr1*, *addr2*, *tm*)))
  = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-incr-mod-protection-theorem.

THEOREM: tm-execute-load-base-protection-theorem
tm-memory (tm-execute-load-base (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-load-base-protection-theorem.

THEOREM: tm-execute-load-limit-protection-theorem
tm-memory (tm-execute-load-limit (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-load-limit-protection-theorem.

THEOREM: tm-execute-load-psw-protection-theorem
tm-memory (tm-execute-load-psw (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-load-psw-protection-theorem.

THEOREM: tm-execute-mod-protection-theorem

((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*))
 ∧ tm-good-address (*addr1*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-mod (*addr1*, *addr2*, *tm*)))
 = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-mod-protection-theorem.

THEOREM: tm-execute-move-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*))
 ∧ tm-good-address (*addr1*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-move (*addr1*, *addr2*, *tm*)))
 = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-move-protection-theorem.

THEOREM: tm-execute-multiply-protection-theorem
((¬ tm-in-supervisor-mode (*tm*))
 ∧ (disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*))
 ∧ tm-good-address (*addr1*, *tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-multiply (*addr1*, *addr2*, *tm*)))
 = getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-multiply-protection-theorem.

THEOREM: tm-execute-return-protection-theorem
tm-memory (tm-execute-return (*tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-return-protection-theorem.

THEOREM: tm-execute-test-iport-protection-theorem
tm-memory (tm-execute-test-iport (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-test-iport-protection-theorem.

THEOREM: tm-execute-test-oport-protection-theorem
tm-memory (tm-execute-test-oport (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-test-oport-protection-theorem.

THEOREM: tm-execute-set-clock-protection-theorem
tm-memory (tm-execute-set-clock (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-set-clock-protection-theorem.

THEOREM: tm-execute-run-protection-theorem
tm-memory (tm-execute-run (*tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-run-protection-theorem.

THEOREM: tm-execute-wait-protection-theorem
tm-memory (tm-execute-wait (*tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-wait-protection-theorem.

THEOREM: tm-execute-start-output-protection-theorem
tm-memory (tm-execute-start-output (*addr1*, *addr2*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-start-output-protection-theorem.

THEOREM: tm-execute-post-output-interrupt-protection-theorem
tm-memory (tm-execute-post-output-interrupt (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-post-output-interrupt-protection-theorem.

THEOREM: tm-execute-svc-protection-theorem
tm-memory (tm-execute-svc (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-svc-protection-theorem.

THEOREM: tm-execute-svc-return-protection-theorem
tm-memory (tm-execute-svc-return (*addr*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-svc-return-protection-theorem.

THEOREM: tm-execute-nullary-protection-theorem
tm-memory (tm-execute-nullary (*opcode*, *tm*)) = tm-memory (*tm*)

EVENT: Disable tm-execute-nullary-protection-theorem.

THEOREM: tm-execute-unary-protection-theorem
((¬ tm-in-supervisor-mode (*tm*)) ∧ disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-unary (*opcode*, *addr*, *tm*)))
= getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-unary-protection-theorem.

THEOREM: tm-execute-binary-protection-theorem
((¬ tm-in-supervisor-mode (*tm*)) ∧ disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-binary (*opcode*, *addr1*, *addr2*, *tm*)))
= getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-binary-protection-theorem.

THEOREM: tm-execute-instruction-protection-theorem
((¬ tm-in-supervisor-mode (*tm*)) ∧ disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute-instruction (*instruction*, *tm*)))
= getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-instruction-protection-theorem.

THEOREM: tm-execute-protection-theorem
((¬ tm-in-supervisor-mode (*tm*)) ∧ disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-execute (*opcode*, *tm*)))
= getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-execute-protection-theorem.

THEOREM: tm-fetch-execute-protection-theorem
((¬ tm-in-supervisor-mode (*tm*)) ∧ disjoint (*b*, *l*, tm-base (*tm*), tm-limit (*tm*)))
→ (getseg (*b*, *l*, tm-memory (tm-fetch-execute (*tm*)))
= getseg (*b*, *l*, tm-memory (*tm*)))

EVENT: Disable tm-fetch-execute-protection-theorem.

THEOREM: length-tm-memory-tm-store
length (tm-memory (tm-store (*value*, *addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-store.

THEOREM: length-tm-memory-tm-execute-add
length (tm-memory (tm-execute-add (*addr1*, *addr2*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-add.

THEOREM: length-tm-memory-tm-execute-branch
length (tm-memory (tm-execute-branch (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-branch.

THEOREM: length-tm-memory-tm-execute-branch-on-zero
length (tm-memory (tm-execute-branch-on-zero (*addr*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-branch-on-zero.

THEOREM: length-tm-memory-tm-execute-branch-not-zero
length (tm-memory (tm-execute-branch-not-zero (*addr*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-branch-not-zero.

THEOREM: length-tm-memory-tm-execute-call
length (tm-memory (tm-execute-call (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-call.

THEOREM: length-tm-memory-tm-execute-compare
length (tm-memory (tm-execute-compare (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-compare.

THEOREM: length-tm-memory-tm-execute-decr
length (tm-memory (tm-execute-decr (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-decr.

THEOREM: length-tm-memory-tm-execute-decr-mod
length (tm-memory (tm-execute-decr-mod (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-decr-mod.

THEOREM: length-tm-memory-tm-execute-incr
length (tm-memory (tm-execute-incr (*addr*, *tm*))) = length (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
length (tm-memory (tm-execute-test-iport (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-test-iport.

THEOREM: length-tm-memory-tm-execute-test-oport
length (tm-memory (tm-execute-test-oport (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-test-oport.

THEOREM: length-tm-memory-tm-execute-set-clock
length (tm-memory (tm-execute-set-clock (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-set-clock.

THEOREM: length-tm-memory-tm-execute-run
length (tm-memory (tm-execute-run (*tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-run.

THEOREM: length-tm-memory-tm-execute-wait
length (tm-memory (tm-execute-wait (*tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-wait.

THEOREM: length-tm-memory-tm-execute-post-output-interrupt
length (tm-memory (tm-execute-post-output-interrupt (*addr*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-post-output-interrupt.

THEOREM: length-tm-memory-tm-execute-start-output
length (tm-memory (tm-execute-start-output (*addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-start-output.

THEOREM: length-tm-memory-tm-execute-svc
length (tm-memory (tm-execute-svc (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-svc.

THEOREM: length-tm-memory-tm-execute-svc-return
length (tm-memory (tm-execute-svc-return (*addr*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-svc-return.

THEOREM: length-tm-memory-tm-execute-nullary
length (tm-memory (tm-execute-nullary (*opcode*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-nullary.

THEOREM: length-tm-memory-tm-execute-unary
length (tm-memory (tm-execute-unary (*opcode*, *addr*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-unary.

THEOREM: length-tm-memory-tm-execute-binary
length (tm-memory (tm-execute-binary (*opcode*, *addr1*, *addr2*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-binary.

THEOREM: length-tm-memory-tm-execute-instruction
length (tm-memory (tm-execute-instruction (*instruction*, *tm*)))
= length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute-instruction.

THEOREM: length-tm-memory-tm-execute
length (tm-memory (tm-execute (*instruction*, *tm*))) = length (tm-memory (*tm*))

EVENT: Disable length-tm-memory-tm-execute.

THEOREM: length-tm-memory-tm-fetch-execute
length (tm-memory (tm-fetch-execute (*tm*))) = length (tm-memory (*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
      (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*) → tm-shellp (*address-space*)

EVENT: Disable tm-shellp-address-space.

THEOREM: non-number-memlength-gives-bad-address-space
(*memlength* ∉ **N**) → (¬ 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*) ∧ finite-numberp (*cc*, '4))
→ 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*) ∧ finite-numberp (*error*, '64))
→ 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*) ∧ finite-numberp (*svcflag*, '2))
→ 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*)
∧ finite-numberp (*svcid*, '128))
→ 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*) ∧ finite-numberp (*pc*, '65536))
→ 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*) ∧ finite-numberp (*sp*, '65536))
→ 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*)
→ 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*)
 ∧ 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*)
 ∧ 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*)
 ∧ 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*)
 ∧ 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*)
→ 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*)
∧ tm-good-address (*addr2*, *address-space*))
→ 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*)
→ 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*)
∧ (¬ tm-stack-underflowp (*address-space*)))
→ 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*)
→ (min ('65536, *memlength*) = *memlength*)

THEOREM: finite-numberp-tm-incr-memlength
((*memlength* ∈ **N**)
∧ ((min ('65536, *memlength*) = *memlength*) ∧ (*b* < (*memlength* − 1))))
→ 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*)
∧ (¬ tm-stack-underflowp (*address-space*)))

→ tm-good-address (real-addr ('2, tm-incr (tm-sp (*address-space*))),
address-space)

EVENT: Disable tm-good-address-tm-incr-tm-sp-for-address-space.

THEOREM: good-address-space-tm-execute-return
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-return (*address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-return.

THEOREM: good-address-space-tm-execute-set-clock
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-set-clock (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-set-clock.

THEOREM: good-address-space-tm-execute-run
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-run (*address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-run.

THEOREM: good-address-space-tm-execute-wait
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-wait (*address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-wait.

THEOREM: good-address-space-tm-execute-test-iport
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-test-iport (*addr*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-test-iport.

THEOREM: good-address-space-tm-execute-test-oport
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-test-oport (*addr*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-test-oport.

THEOREM: good-address-space-tm-execute-start-output
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-start-output (*addr1*, *addr2*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-start-output.

THEOREM: good-address-space-tm-execute-post-output-interrupt
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-post-output-interrupt (*addr*,
address-space),
memlength)

EVENT: Disable good-address-space-tm-execute-post-output-interrupt.

THEOREM: good-address-space-tm-execute-svc
(good-address-space (*address-space*, *memlength*)
∧ tm-good-address (*addr*, *address-space*))
→ good-address-space (tm-execute-svc (*addr*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-svc.

THEOREM: good-address-space-tm-execute-svc-return
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-svc-return (*addr*, *address-space*),
memlength)

EVENT: Disable good-address-space-tm-execute-svc-return.

THEOREM: good-address-space-tm-execute-nullary
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-nullary (*opcode*, *address-space*), *memlength*)

EVENT: Disable good-address-space-tm-execute-nullary.

THEOREM: good-address-space-tm-execute-unary
good-address-space (*address-space*, *memlength*)
→ good-address-space (tm-execute-unary (*opcode*, *addr*, *address-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*)
 ∧ ((¬ tm-in-supervisor-mode (*tm*)) ∧ (*addr* < tm-limit (*tm*))))
→ (tm-fetch-from-memory (*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-fetch-from-memory (*addr*, *tm*))

EVENT: Disable tm-fetch-from-memory-mapup-address-space.

THEOREM: tm-fetch-from-regmem-mapup-address-space
good-tm (*tm*)
→ (tm-fetch-from-regmem (*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-fetch-from-regmem (*addr*, *tm*))

EVENT: Disable tm-fetch-from-regmem-mapup-address-space.

THEOREM: lessp-good-memory-address-limit
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ ((real-addr-source (*addr*) ≯ 0)
∧ ((real-addr-source (*addr*) ≠ '1)
∧ tm-good-address (*addr*, *tm*)))))
→ (real-addr-num (*addr*) < tm-limit (*tm*))

THEOREM: tm-fetch-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*)) ∧ tm-good-address (*addr*, *tm*)))
→ (tm-fetch (*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-fetch (*addr*, *tm*))

EVENT: Disable tm-fetch-mapup-address-space.

THEOREM: tm-store-in-memory-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*))
then '*1*false
else '*1*true **endif**
∧ (*addr* < tm-limit (*tm*)))))
→ (tm-store-in-memory (*value*,
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-store-in-memory (*value*,
addr,
tm)),
tm-regs (tm-store-in-memory (*value*, *addr*, *tm*)),
tm-cc (tm-store-in-memory (*value*, *addr*, *tm*)),
tm-error (tm-store-in-memory (*value*, *addr*, *tm*)),
tm-svcflag (tm-store-in-memory (*value*,
addr,
tm)),
tm-svcid (tm-store-in-memory (*value*, *addr*, *tm*)),
tm-base (tm-store-in-memory (*value*, *addr*, *tm*)),
tm-limit (tm-store-in-memory (*value*, *addr*, *tm*))))

EVENT: Disable tm-store-in-memory-commutes-with-mapup-address-space.

THEOREM: tm-store-in-regmem-commutes-with-mapup-address-space
good-tm (*tm*)
→ (tm-store-in-regmem (*value*,
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-store-in-regmem (*value*,
addr,
tm)),
tm-regs (tm-store-in-regmem (*value*, *addr*, *tm*)),
tm-cc (tm-store-in-regmem (*value*, *addr*, *tm*)),
tm-error (tm-store-in-regmem (*value*, *addr*, *tm*)),
tm-svcflag (tm-store-in-regmem (*value*,
addr,
tm)),
tm-svcid (tm-store-in-regmem (*value*, *addr*, *tm*)),
tm-base (tm-store-in-regmem (*value*, *addr*, *tm*)),
tm-limit (tm-store-in-regmem (*value*, *addr*, *tm*))))

EVENT: Disable tm-store-in-regmem-commutes-with-mapup-address-space.

THEOREM: tm-store-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 (*addr*, *tm*))))
→ (tm-store (*value*,
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-store (*value*, *addr*, *tm*)),
tm-regs (tm-store (*value*, *addr*, *tm*)),
tm-cc (tm-store (*value*, *addr*, *tm*)),
tm-error (tm-store (*value*, *addr*, *tm*)),
tm-svcflag (tm-store (*value*, *addr*, *tm*)),
tm-svcid (tm-store (*value*, *addr*, *tm*)),

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
tm-set-cc (*cc*,
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-cc (*cc*, *tm*)),
tm-regs (tm-set-cc (*cc*, *tm*)),
tm-cc (tm-set-cc (*cc*, *tm*)),
tm-error (tm-set-cc (*cc*, *tm*)),
tm-svcflag (tm-set-cc (*cc*, *tm*)),
tm-svcid (tm-set-cc (*cc*, *tm*)),
tm-base (tm-set-cc (*cc*, *tm*)),
tm-limit (tm-set-cc (*cc*, *tm*)))

EVENT: Disable tm-set-cc-commutes-with-mapup-address-space.

THEOREM: tm-set-error-commutes-with-mapup-address-space
tm-set-error (*error*,
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-error (*error*, *tm*)),
tm-regs (tm-set-error (*error*, *tm*)),
tm-cc (tm-set-error (*error*, *tm*)),
tm-error (tm-set-error (*error*, *tm*)),
tm-svcflag (tm-set-error (*error*, *tm*)),
tm-svcid (tm-set-error (*error*, *tm*)),
tm-base (tm-set-error (*error*, *tm*)),
tm-limit (tm-set-error (*error*, *tm*)))

EVENT: Disable tm-set-error-commutes-with-mapup-address-space.

THEOREM: tm-set-svcflag-commutes-with-mapup-address-space
tm-set-svcflag (*svcflag*,
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-svcflag (*svcflag*, *tm*)),
tm-regs (tm-set-svcflag (*svcflag*, *tm*)),
tm-cc (tm-set-svcflag (*svcflag*, *tm*)),
tm-error (tm-set-svcflag (*svcflag*, *tm*)),
tm-svcflag (tm-set-svcflag (*svcflag*, *tm*)),
tm-svcid (tm-set-svcflag (*svcflag*, *tm*)),
tm-base (tm-set-svcflag (*svcflag*, *tm*)),
tm-limit (tm-set-svcflag (*svcflag*, *tm*)))

EVENT: Disable tm-set-svcflag-commutes-with-mapup-address-space.

THEOREM: tm-set-svcid-commutes-with-mapup-address-space
tm-set-svcid (*svcid*,
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-svcid (*svcid*, *tm*)),
tm-regs (tm-set-svcid (*svcid*, *tm*)),
tm-cc (tm-set-svcid (*svcid*, *tm*)),
tm-error (tm-set-svcid (*svcid*, *tm*)),
tm-svcflag (tm-set-svcid (*svcid*, *tm*)),
tm-svcid (tm-set-svcid (*svcid*, *tm*)),
tm-base (tm-set-svcid (*svcid*, *tm*)),
tm-limit (tm-set-svcid (*svcid*, *tm*)))

EVENT: Disable tm-set-svcid-commutes-with-mapup-address-space.

THEOREM: tm-in-supervisor-mode-tm-set-cc
tm-in-supervisor-mode (tm-set-cc (*cc*, *tm*)) = tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-set-cc.

THEOREM: tm-in-supervisor-mode-tm-set-svcflag
tm-in-supervisor-mode (tm-set-svcflag (*svcflag*, *tm*))
= tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-set-svcflag.

THEOREM: tm-in-supervisor-mode-tm-set-svcid
tm-in-supervisor-mode (tm-set-svcid (*svcid*, *tm*)) = tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-set-svcid.

THEOREM: tm-in-supervisor-mode-tm-set-sp
tm-in-supervisor-mode (tm-set-sp (*sp*, *tm*)) = tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-set-sp.

THEOREM: tm-in-supervisor-mode-tm-set-pc
tm-in-supervisor-mode (tm-set-pc (*pc*, *tm*)) = tm-in-supervisor-mode (*tm*)

EVENT: Disable tm-in-supervisor-mode-tm-set-pc.

THEOREM: tm-limit-mapup-address-space
tm-limit (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*)))
= fix (tm-limit (*tm*))

EVENT: Disable tm-limit-mapup-address-space.

THEOREM: tm-base-mapup-address-space
tm-base (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*)))
= '0

EVENT: Disable tm-base-mapup-address-space.

THEOREM: plus-tm-base-tm-limit-tm-set-cc
(tm-base (tm-set-cc (*cc*, *tm*)) + tm-limit (tm-set-cc (*cc*, *tm*)))
= (tm-base (*tm*) + tm-limit (*tm*))

EVENT: Disable plus-tm-base-tm-limit-tm-set-cc.

THEOREM: plus-tm-base-tm-limit-tm-set-sp
(tm-base (tm-set-sp (*sp*, *tm*)) + tm-limit (tm-set-sp (*sp*, *tm*)))
= (tm-base (*tm*) + tm-limit (*tm*))

EVENT: Disable plus-tm-base-tm-limit-tm-set-sp.

THEOREM: tm-execute-add-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*))
then '*1*false
else '*1*true **endif**
∧ (tm-good-address (*addr1*, *tm*)
∧ tm-good-address (*addr2*, *tm*)))))
→ (tm-execute-add (*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-add (*addr1*, *addr2*, *tm*)),
tm-regs (tm-execute-add (*addr1*, *addr2*, *tm*)),
tm-cc (tm-execute-add (*addr1*, *addr2*, *tm*)),
tm-error (tm-execute-add (*addr1*, *addr2*, *tm*)),
tm-svcflag (tm-execute-add (*addr1*, *addr2*, *tm*)),
tm-svcid (tm-execute-add (*addr1*, *addr2*, *tm*)),

tm-base (tm-execute-add (*addr1*, *addr2*, *tm*)),
tm-limit (tm-execute-add (*addr1*, *addr2*, *tm*))))

EVENT: Disable tm-execute-add-commutes-with-mapup-address-space.

THEOREM: tm-execute-branch-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*)) ∧ tm-good-address (*addr*, *tm*)))
→ (tm-execute-branch (*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-branch (*addr*, *tm*)),
tm-regs (tm-execute-branch (*addr*, *tm*)),
tm-cc (tm-execute-branch (*addr*, *tm*)),
tm-error (tm-execute-branch (*addr*, *tm*)),
tm-svcflag (tm-execute-branch (*addr*, *tm*)),
tm-svcid (tm-execute-branch (*addr*, *tm*)),
tm-base (tm-execute-branch (*addr*, *tm*)),
tm-limit (tm-execute-branch (*addr*, *tm*))))

EVENT: Disable tm-execute-branch-commutes-with-mapup-address-space.

THEOREM: tm-cc-mapup-address-space
good-tm (*tm*)
→ (tm-cc (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-cc (*tm*))

EVENT: Disable tm-cc-mapup-address-space.

THEOREM: tm-error-mapup-address-space

good-tm (*tm*)
→ (tm-error (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-error (*tm*))

EVENT: Disable tm-error-mapup-address-space.

THEOREM: tm-svcflag-mapup-address-space
good-tm (*tm*)
→ (tm-svcflag (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-svcflag (*tm*))

EVENT: Disable tm-svcflag-mapup-address-space.

THEOREM: tm-svcid-mapup-address-space
good-tm (*tm*)
→ (tm-svcid (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-svcid (*tm*))

EVENT: Disable tm-svcid-mapup-address-space.

THEOREM: tm-pc-mapup-address-space
good-tm (*tm*)

→ (tm-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*)))
= tm-pc (*tm*))

EVENT: Disable tm-pc-mapup-address-space.

THEOREM: tm-sp-mapup-address-space
good-tm (*tm*)
→ (tm-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*)))
= tm-sp (*tm*))

EVENT: Disable tm-sp-mapup-address-space.

THEOREM: tm-execute-branch-on-zero-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*)) ∧ tm-good-address (*addr*, *tm*)))
→ (tm-execute-branch-on-zero (*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-branch-on-zero (*addr*,
tm)),
tm-regs (tm-execute-branch-on-zero (*addr*, *tm*)),
tm-cc (tm-execute-branch-on-zero (*addr*, *tm*)),
tm-error (tm-execute-branch-on-zero (*addr*,

tm)),
tm-svcflag (tm-execute-branch-on-zero ($addr$,
tm)),
tm-svcid (tm-execute-branch-on-zero ($addr$,
tm)),
tm-base (tm-execute-branch-on-zero ($addr$, tm)),
tm-limit (tm-execute-branch-on-zero ($addr$,
tm))))

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)
$\wedge$ ((¬ tm-in-supervisor-mode (tm)) $\wedge$ tm-good-address ($addr$, tm)))
$\rightarrow$ (tm-execute-branch-not-zero ($addr$,
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-branch-not-zero ($addr$,
tm)),
tm-regs (tm-execute-branch-not-zero ($addr$,
tm)),
tm-cc (tm-execute-branch-not-zero ($addr$, tm)),
tm-error (tm-execute-branch-not-zero ($addr$,
tm)),
tm-svcflag (tm-execute-branch-not-zero ($addr$,
tm)),
tm-svcid (tm-execute-branch-not-zero ($addr$,
tm)),
tm-base (tm-execute-branch-not-zero ($addr$,
tm)),
tm-limit (tm-execute-branch-not-zero ($addr$,
tm))))

EVENT: Disable tm-execute-branch-not-zero-commutes-with-mapup-address-space.

THEOREM: tm-stack-overflowp-mapup-address-space
good-tm (tm)

→ (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*)
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
∧ (¬ tm-in-supervisor-mode (*tm*))))
→ (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*)
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ tm-good-address (*addr*, *tm*))))
→ (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*),

tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-call (*addr*, *tm*)),
tm-regs (tm-execute-call (*addr*, *tm*)),
tm-cc (tm-execute-call (*addr*, *tm*)),
tm-error (tm-execute-call (*addr*, *tm*)),
tm-svcflag (tm-execute-call (*addr*, *tm*)),
tm-svcid (tm-execute-call (*addr*, *tm*)),
tm-base (tm-execute-call (*addr*, *tm*)),
tm-limit (tm-execute-call (*addr*, *tm*))))

EVENT: Disable tm-execute-call-commutes-with-mapup-address-space.

THEOREM: tm-execute-compare-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ (tm-good-address (*addr1*, *tm*) ∧ tm-good-address (*addr2*, *tm*))))
→ (tm-execute-compare (*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-compare (*addr1*,
addr2,
tm)),
tm-regs (tm-execute-compare (*addr1*, *addr2*, *tm*)),
tm-cc (tm-execute-compare (*addr1*, *addr2*, *tm*)),
tm-error (tm-execute-compare (*addr1*,
addr2,
tm)),
tm-svcflag (tm-execute-compare (*addr1*,
addr2,
tm)),
tm-svcid (tm-execute-compare (*addr1*,
addr2,
tm)),
tm-base (tm-execute-compare (*addr1*, *addr2*, *tm*)),
tm-limit (tm-execute-compare (*addr1*,
addr2,

tm))))

EVENT: Disable tm-execute-compare-commutes-with-mapup-address-space.

THEOREM: tm-execute-decr-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 ($addr$, 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)
 ∧ (**if** '65536 < (tm-base (tm) + tm-limit (tm)) **then** '*1*false
 else '*1*true **endif**
 ∧ ((¬ tm-in-supervisor-mode (tm))
 ∧ (tm-good-address ($addr1$, tm)
 ∧ 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),

tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-regs (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-cc (tm-execute-decr-mod (*addr1*, *addr2*, *tm*)),
tm-error (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-svcflag (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-svcid (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-base (tm-execute-decr-mod (*addr1*,
addr2,
tm)),
tm-limit (tm-execute-decr-mod (*addr1*,
addr2,
tm))))

EVENT: Disable tm-execute-decr-mod-commutes-with-mapup-address-space.

THEOREM: tm-execute-incr-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 (*addr*, *tm*))))
→ (tm-execute-incr (*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*)))

$$
\begin{array}{lll}
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-regs}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-cc}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-error}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcid}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-base}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm})), \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-limit}\,(\text{tm-execute-incr}\,(\mathit{addr},\ \mathit{tm}))))
\end{array}
$$

EVENT: Disable tm-execute-incr-commutes-with-mapup-address-space.

THEOREM: tm-execute-incr-mod-commutes-with-mapup-address-space

$$
\begin{array}{lll}
 & (\text{good-tm}\,(\mathit{tm}) & \\
\wedge & \quad (\textbf{if}\ \texttt{'65536} < (\text{tm-base}\,(\mathit{tm}) + \text{tm-limit}\,(\mathit{tm}))\ \textbf{then}\ \texttt{'*1*false} & \\
 & \quad\ \textbf{else}\ \texttt{'*1*true}\ \textbf{endif} & \\
\wedge & \quad ((\neg\ \text{tm-in-supervisor-mode}\,(\mathit{tm})) & \\
 & \qquad \wedge\quad (\text{tm-good-address}\,(\mathit{addr1},\ \mathit{tm}) & \\
 & \qquad\qquad \wedge\quad \text{tm-good-address}\,(\mathit{addr2},\ \mathit{tm}))))) & \\
\rightarrow & (\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad \text{mapup-address-space}\,(\text{tm-memory}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-regs}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-cc}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-error}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcid}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-base}\,(\mathit{tm}), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-limit}\,(\mathit{tm}))) & \\
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-regs}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-cc}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1},\ \mathit{addr2},\ \mathit{tm})), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-error}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \mathit{tm})), & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcid}\,(\text{tm-execute-incr-mod}\,(\mathit{addr1}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, & \\
 & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), &
\end{array}
$$

tm-base (tm-execute-incr-mod (*addr1*,
addr2,
tm)),
tm-limit (tm-execute-incr-mod (*addr1*,
addr2,
tm))))

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*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (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*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (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*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (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.

THEOREM: tm-execute-mod-commutes-with-mapup-address-space
(good-tm (*tm*)
 ∧ ((¬ tm-in-supervisor-mode (*tm*))
  ∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*))
  then '*1*false
  else '*1*true **endif**
  ∧ (tm-good-address (*addr1*, *tm*)
  ∧ tm-good-address (*addr2*, *tm*)))))
→ (tm-execute-mod (*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-mod (*addr1*, *addr2*, *tm*)),
  tm-regs (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-cc (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-error (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-svcflag (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-svcid (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-base (tm-execute-mod (*addr1*, *addr2*, *tm*)),
  tm-limit (tm-execute-mod (*addr1*, *addr2*, *tm*))))

EVENT: Disable tm-execute-mod-commutes-with-mapup-address-space.

THEOREM: tm-execute-move-commutes-with-mapup-address-space
(good-tm (*tm*)
 ∧ ((¬ tm-in-supervisor-mode (*tm*))
  ∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*))
  then '*1*false
  else '*1*true **endif**
  ∧ (tm-good-address (*addr1*, *tm*)
  ∧ tm-good-address (*addr2*, *tm*)))))
→ (tm-execute-move (*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-move (*addr1*, *addr2*, *tm*)),
tm-regs (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-cc (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-error (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-svcflag (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-svcid (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-base (tm-execute-move (*addr1*, *addr2*, *tm*)),
tm-limit (tm-execute-move (*addr1*, *addr2*, *tm*))))

EVENT: Disable tm-execute-move-commutes-with-mapup-address-space.

THEOREM: tm-execute-multiply-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*))
then '*1*false
else '*1*true **endif**
∧ (tm-good-address (*addr1*, *tm*)
∧ tm-good-address (*addr2*, *tm*)))))
→ (tm-execute-multiply (*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-multiply (*addr1*,
addr2,
tm)),
tm-regs (tm-execute-multiply (*addr1*,
addr2,
tm)),
tm-cc (tm-execute-multiply (*addr1*, *addr2*, *tm*)),
tm-error (tm-execute-multiply (*addr1*,
addr2,
tm)),
tm-svcflag (tm-execute-multiply (*addr1*,
addr2,

tm)),
tm-svcid (tm-execute-multiply (*addr1*, *addr2*, tm)),
tm-base (tm-execute-multiply (*addr1*, *addr2*, tm)),
tm-limit (tm-execute-multiply (*addr1*, *addr2*, tm))))

EVENT: Disable tm-execute-multiply-commutes-with-mapup-address-space.

THEOREM: tm-stack-underflowp-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-stack-underflowp (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-underflowp (tm))

EVENT: Disable tm-stack-underflowp-mapup-address-space.

THEOREM: tm-execute-return-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-return (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-return (*tm*)),
tm-regs (tm-execute-return (*tm*)),
tm-cc (tm-execute-return (*tm*)),
tm-error (tm-execute-return (*tm*)),
tm-svcflag (tm-execute-return (*tm*)),
tm-svcid (tm-execute-return (*tm*)),
tm-base (tm-execute-return (*tm*)),
tm-limit (tm-execute-return (*tm*))))

EVENT: Disable tm-execute-return-commutes-with-mapup-address-space.

THEOREM: tm-execute-set-clock-commutes-with-mapup-address-space
(good-tm (*tm*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (tm-execute-set-clock (*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-set-clock (*addr*, *tm*)),
tm-regs (tm-execute-set-clock (*addr*, *tm*)),
tm-cc (tm-execute-set-clock (*addr*, *tm*)),
tm-error (tm-execute-set-clock (*addr*, *tm*)),
tm-svcflag (tm-execute-set-clock (*addr*, *tm*)),
tm-svcid (tm-execute-set-clock (*addr*, *tm*)),
tm-base (tm-execute-set-clock (*addr*, *tm*)),
tm-limit (tm-execute-set-clock (*addr*, *tm*))))

EVENT: Disable tm-execute-set-clock-commutes-with-mapup-address-space.

THEOREM: tm-execute-run-commutes-with-mapup-address-space
(good-tm (*tm*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (tm-execute-run (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*)))

$$
\begin{array}{ll}
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-regs}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-cc}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-error}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-svcflag}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-svcid}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-base}\,(\text{tm-execute-run}\,(tm)),\\
& \qquad \text{tm-limit}\,(\text{tm-execute-run}\,(tm))))
\end{array}
$$

EVENT: Disable tm-execute-run-commutes-with-mapup-address-space.

THEOREM: tm-execute-wait-commutes-with-mapup-address-space

$$
\begin{array}{ll}
& (\text{good-tm}\,(tm) \wedge (\neg\ \text{tm-in-supervisor-mode}\,(tm)))\\
\rightarrow & (\text{tm-execute-wait}\,(\text{mapup-address-space}\,(\text{tm-memory}\,(tm),\\
& \qquad \text{tm-regs}\,(tm),\\
& \qquad \text{tm-cc}\,(tm),\\
& \qquad \text{tm-error}\,(tm),\\
& \qquad \text{tm-svcflag}\,(tm),\\
& \qquad \text{tm-svcid}\,(tm),\\
& \qquad \text{tm-base}\,(tm),\\
& \qquad \text{tm-limit}\,(tm)))\\
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-regs}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-cc}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-error}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-svcflag}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-svcid}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-base}\,(\text{tm-execute-wait}\,(tm)),\\
& \qquad \text{tm-limit}\,(\text{tm-execute-wait}\,(tm))))
\end{array}
$$

EVENT: Disable tm-execute-wait-commutes-with-mapup-address-space.

THEOREM: tm-execute-test-iport-commutes-with-mapup-address-space

$$
\begin{array}{ll}
& (\text{good-tm}\,(tm) \wedge (\neg\ \text{tm-in-supervisor-mode}\,(tm)))\\
\rightarrow & (\text{tm-execute-test-iport}\,(addr,\\
& \qquad \text{mapup-address-space}\,(\text{tm-memory}\,(tm),\\
& \qquad\qquad \text{tm-regs}\,(tm),\\
& \qquad\qquad \text{tm-cc}\,(tm),\\
& \qquad\qquad \text{tm-error}\,(tm),\\
& \qquad\qquad \text{tm-svcflag}\,(tm),\\
& \qquad\qquad \text{tm-svcid}\,(tm),\\
& \qquad\qquad \text{tm-base}\,(tm),\\
& \qquad\qquad \text{tm-limit}\,(tm)))
\end{array}
$$

$$
\begin{array}{ll}
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-regs}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-cc}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-error}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-svcflag}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-svcid}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-base}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-limit}\,(\text{tm-execute-test-iport}\,(\mathit{addr},\ \mathit{tm}))))
\end{array}
$$

EVENT: Disable tm-execute-test-iport-commutes-with-mapup-address-space.

THEOREM: tm-execute-test-oport-commutes-with-mapup-address-space

$$
\begin{array}{ll}
& (\text{good-tm}\,(\mathit{tm}) \wedge (\neg\ \text{tm-in-supervisor-mode}\,(\mathit{tm})))\\
\rightarrow & (\text{tm-execute-test-oport}\,(\mathit{addr},\\
& \qquad\qquad \text{mapup-address-space}\,(\text{tm-memory}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-regs}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-cc}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-error}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-svcid}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-base}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-limit}\,(\mathit{tm})))\\
= & \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-regs}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-cc}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-error}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-svcflag}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-svcid}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-base}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm})),\\
& \qquad\qquad \text{tm-limit}\,(\text{tm-execute-test-oport}\,(\mathit{addr},\ \mathit{tm}))))
\end{array}
$$

EVENT: Disable tm-execute-test-oport-commutes-with-mapup-address-space.

THEOREM: tm-execute-start-output-commutes-with-mapup-address-space

$$
\begin{array}{ll}
& (\text{good-tm}\,(\mathit{tm}) \wedge (\neg\ \text{tm-in-supervisor-mode}\,(\mathit{tm})))\\
\rightarrow & (\text{tm-execute-start-output}\,(\mathit{addr1},\\
& \qquad\qquad \mathit{addr2},\\
& \qquad\qquad \text{mapup-address-space}\,(\text{tm-memory}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-regs}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-cc}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-error}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\mathit{tm}),\\
& \qquad\qquad\qquad\qquad \text{tm-svcid}\,(\mathit{tm}),
\end{array}
$$

tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-regs (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-cc (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-error (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-svcflag (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-svcid (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-base (tm-execute-start-output (*addr1*,
addr2,
tm)),
tm-limit (tm-execute-start-output (*addr1*,
addr2,
tm))))

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*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (tm-execute-post-output-interrupt (*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-post-output-interrupt (*addr*,
tm)),
tm-regs (tm-execute-post-output-interrupt (*addr*,
tm)),

tm-cc (tm-execute-post-output-interrupt (*addr*, *tm*)),
tm-error (tm-execute-post-output-interrupt (*addr*, *tm*)),
tm-svcflag (tm-execute-post-output-interrupt (*addr*, *tm*)),
tm-svcid (tm-execute-post-output-interrupt (*addr*, *tm*)),
tm-base (tm-execute-post-output-interrupt (*addr*, *tm*)),
tm-limit (tm-execute-post-output-interrupt (*addr*, *tm*))))

EVENT: Disable tm-execute-post-output-interrupt-commutes-with-mapup-address-space.

THEOREM: tm-execute-svc-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ ((¬ tm-in-supervisor-mode (*tm*)) ∧ tm-good-address (*addr*, *tm*)))
→ (tm-execute-svc (*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-svc (*addr*, *tm*)),
tm-regs (tm-execute-svc (*addr*, *tm*)),
tm-cc (tm-execute-svc (*addr*, *tm*)),
tm-error (tm-execute-svc (*addr*, *tm*)),
tm-svcflag (tm-execute-svc (*addr*, *tm*)),
tm-svcid (tm-execute-svc (*addr*, *tm*)),
tm-base (tm-execute-svc (*addr*, *tm*)),
tm-limit (tm-execute-svc (*addr*, *tm*))))

EVENT: Disable tm-execute-svc-commutes-with-mapup-address-space.

THEOREM: tm-execute-svc-return-commutes-with-mapup-address-space
(good-tm (*tm*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (tm-execute-svc-return (*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-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*),

tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-nullary2 (*opcode*, *tm*)),
tm-regs (tm-execute-nullary2 (*opcode*, *tm*)),
tm-cc (tm-execute-nullary2 (*opcode*, *tm*)),
tm-error (tm-execute-nullary2 (*opcode*, *tm*)),
tm-svcflag (tm-execute-nullary2 (*opcode*, *tm*)),
tm-svcid (tm-execute-nullary2 (*opcode*, *tm*)),
tm-base (tm-execute-nullary2 (*opcode*, *tm*)),
tm-limit (tm-execute-nullary2 (*opcode*, *tm*))))

EVENT: Disable tm-execute-nullary2-commutes-with-mapup-address-space.

THEOREM: tm-execute-nullary1-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-nullary1 (*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-nullary1 (*opcode*, *tm*)),
tm-regs (tm-execute-nullary1 (*opcode*, *tm*)),
tm-cc (tm-execute-nullary1 (*opcode*, *tm*)),
tm-error (tm-execute-nullary1 (*opcode*, *tm*)),
tm-svcflag (tm-execute-nullary1 (*opcode*, *tm*)),
tm-svcid (tm-execute-nullary1 (*opcode*, *tm*)),
tm-base (tm-execute-nullary1 (*opcode*, *tm*)),
tm-limit (tm-execute-nullary1 (*opcode*, *tm*))))

EVENT: Disable tm-execute-nullary1-commutes-with-mapup-address-space.

THEOREM: tm-execute-nullary-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-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** ¬ 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*)
∧ (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-unary8 (*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-unary8 (*opcode*,
addr,
tm)),
tm-regs (tm-execute-unary8 (*opcode*, *addr*, *tm*)),
tm-cc (tm-execute-unary8 (*opcode*, *addr*, *tm*)),
tm-error (tm-execute-unary8 (*opcode*, *addr*, *tm*)),
tm-svcflag (tm-execute-unary8 (*opcode*,
addr,
tm)),
tm-svcid (tm-execute-unary8 (*opcode*, *addr*, *tm*)),
tm-base (tm-execute-unary8 (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary8 (*opcode*, *addr*, *tm*))))

EVENT: Disable tm-execute-unary8-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary7-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-unary7 (*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-unary7 (*opcode*,
addr,
tm)),
tm-regs (tm-execute-unary7 (*opcode*, *addr*, *tm*)),
tm-cc (tm-execute-unary7 (*opcode*, *addr*, *tm*)),
tm-error (tm-execute-unary7 (*opcode*, *addr*, *tm*)),
tm-svcflag (tm-execute-unary7 (*opcode*,
addr,
tm)),
tm-svcid (tm-execute-unary7 (*opcode*, *addr*, *tm*)),

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**

$\wedge$ $(\neg$ tm-in-supervisor-mode $(tm)))))$
$\rightarrow$ (tm-execute-unary5 $(opcode,$
$addr,$
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-unary5 $(opcode,$
$addr,$
$tm)),$
tm-regs (tm-execute-unary5 $(opcode,\ addr,\ tm)),$
tm-cc (tm-execute-unary5 $(opcode,\ addr,\ tm)),$
tm-error (tm-execute-unary5 $(opcode,\ addr,\ tm)),$
tm-svcflag (tm-execute-unary5 $(opcode,$
$addr,$
$tm)),$
tm-svcid (tm-execute-unary5 $(opcode,\ addr,\ tm)),$
tm-base (tm-execute-unary5 $(opcode,\ addr,\ tm)),$
tm-limit (tm-execute-unary5 $(opcode,\ addr,\ tm))))$

EVENT: Disable tm-execute-unary5-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary4-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-unary4 $(opcode,$
$addr,$
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-unary4 $(opcode,$

addr,
tm)),
tm-regs (tm-execute-unary4 (*opcode*, *addr*, *tm*)),
tm-cc (tm-execute-unary4 (*opcode*, *addr*, *tm*)),
tm-error (tm-execute-unary4 (*opcode*, *addr*, *tm*)),
tm-svcflag (tm-execute-unary4 (*opcode*,
addr,
tm)),
tm-svcid (tm-execute-unary4 (*opcode*, *addr*, *tm*)),
tm-base (tm-execute-unary4 (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary4 (*opcode*, *addr*, *tm*))))

EVENT: Disable tm-execute-unary4-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary3-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-unary3 (*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-unary3 (*opcode*,
addr,
tm)),
tm-regs (tm-execute-unary3 (*opcode*, *addr*, *tm*)),
tm-cc (tm-execute-unary3 (*opcode*, *addr*, *tm*)),
tm-error (tm-execute-unary3 (*opcode*, *addr*, *tm*)),
tm-svcflag (tm-execute-unary3 (*opcode*,
addr,
tm)),
tm-svcid (tm-execute-unary3 (*opcode*, *addr*, *tm*)),
tm-base (tm-execute-unary3 (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary3 (*opcode*, *addr*, *tm*))))

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*),

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-unary1 (*opcode*,
addr,
tm)),
tm-regs (tm-execute-unary1 (*opcode*, *addr*, *tm*)),
tm-cc (tm-execute-unary1 (*opcode*, *addr*, *tm*)),
tm-error (tm-execute-unary1 (*opcode*, *addr*, *tm*)),
tm-svcflag (tm-execute-unary1 (*opcode*,
addr,
tm)),
tm-svcid (tm-execute-unary1 (*opcode*, *addr*, *tm*)),
tm-base (tm-execute-unary1 (*opcode*, *addr*, *tm*)),
tm-limit (tm-execute-unary1 (*opcode*, *addr*, *tm*))))

EVENT: Disable tm-execute-unary1-commutes-with-mapup-address-space.

THEOREM: tm-execute-unary-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-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** (¬ tm-good-address (*addr1*, *tm*))
∨ (¬ 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
(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-binary6 (*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-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-regs (tm-execute-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-cc (tm-execute-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-error (tm-execute-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-svcflag (tm-execute-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-svcid (tm-execute-binary6 (*opcode*,
  addr1,
  addr2,
  tm)),
  tm-base (tm-execute-binary6 (*opcode*,

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*)
∧ (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-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)),
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*)))
= mapup-address-space (tm-memory (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*),

tm-error (*tm*),
tm-svcflag (*tm*),
tm-svcid (*tm*),
tm-base (*tm*),
tm-limit (*tm*)))
= mapup-address-space (tm-memory (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-regs (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-cc (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-error (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-svcflag (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-svcid (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-base (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm)),
tm-limit (tm-execute-binary3 (*opcode*,
addr1,
addr2,
tm))))

EVENT: Disable tm-execute-binary3-commutes-with-mapup-address-space.

THEOREM: tm-execute-binary2-commutes-with-mapup-address-space
(good-tm (*tm*)
∧ (tm-good-address (*addr1*, *tm*)
∧ (tm-good-address (*addr2*, *tm*)

$$
\begin{array}{l}
\quad \wedge \quad (\textbf{if } \texttt{'65536} < (\text{tm-base}\,(tm) + \text{tm-limit}\,(tm)) \\
\qquad \textbf{then } \texttt{'*1*false} \\
\qquad \textbf{else } \texttt{'*1*true } \textbf{endif} \\
\qquad \wedge \quad (\neg\ \text{tm-in-supervisor-mode}\,(tm))))))) \\
\rightarrow \quad (\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad \text{mapup-address-space}\,(\text{tm-memory}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-regs}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-cc}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-error}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcflag}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcid}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-base}\,(tm), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-limit}\,(tm))) \\
\qquad = \quad \text{mapup-address-space}\,(\text{tm-memory}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-regs}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-cc}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-error}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcflag}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\ \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-svcid}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-base}\,(\text{tm-execute-binary2}\,(\mathit{opcode}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr1}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{addr2}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathit{tm})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-limit}\,(\text{tm-execute-binary2}\,(\mathit{opcode},
\end{array}
$$

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*)
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
∧ (¬ tm-in-supervisor-mode (*tm*))))
→ (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*)
∧ (**if** `'65536` < (tm-base (*tm*) + tm-limit (*tm*)) **then** `'*1*false`
else `'*1*true` **endif**
∧ (¬ tm-in-supervisor-mode (*tm*))))
→ (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)),
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*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (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*)))
= mapup-address-space (tm-memory (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*)
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
∧ (¬ tm-in-supervisor-mode (*tm*))))
→ (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*)
∧ (**if** '65536 < (tm-base (*tm*) + tm-limit (*tm*)) **then** '*1*false
else '*1*true **endif**
∧ (¬ tm-in-supervisor-mode (*tm*))))

→ (tm-execute-instruction (*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-instruction (*instruction*, *tm*)),
tm-regs (tm-execute-instruction (*instruction*, *tm*)),
tm-cc (tm-execute-instruction (*instruction*, *tm*)),
tm-error (tm-execute-instruction (*instruction*, *tm*)),
tm-svcflag (tm-execute-instruction (*instruction*, *tm*)),
tm-svcid (tm-execute-instruction (*instruction*, *tm*)),
tm-base (tm-execute-instruction (*instruction*, *tm*)),
tm-limit (tm-execute-instruction (*instruction*, *tm*))))

EVENT: Disable tm-execute-instruction-commutes-with-mapup-address-space.

THEOREM: tm-decrement-clock-commutes-with-mapup-address-space
(good-tm (*tm*) ∧ (¬ tm-in-supervisor-mode (*tm*)))
→ (tm-decrement-clock (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-decrement-clock (*tm*)),
tm-regs (tm-decrement-clock (*tm*)),
tm-cc (tm-decrement-clock (*tm*)),
tm-error (tm-decrement-clock (*tm*)),
tm-svcflag (tm-decrement-clock (*tm*)),
tm-svcid (tm-decrement-clock (*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*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ tm-good-address (real-addr ('2, tm-incrn ('1, tm-pc (*tm*))), *tm*)))
→ (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*)
∧ ((¬ tm-in-supervisor-mode (*tm*))
∧ tm-good-address (real-addr ('2, tm-incrn ('2, tm-pc (*tm*))), *tm*)))
→ (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*),

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 (*instruction*, *tm*)),
tm-regs (tm-execute (*instruction*, *tm*)),
tm-cc (tm-execute (*instruction*, *tm*)),
tm-error (tm-execute (*instruction*, *tm*)),
tm-svcflag (tm-execute (*instruction*, *tm*)),
tm-svcid (tm-execute (*instruction*, *tm*)),
tm-base (tm-execute (*instruction*, *tm*)),
tm-limit (tm-execute (*instruction*, *tm*))))

EVENT: Disable tm-execute-commutes-with-mapup-address-space.

THEOREM: tm-fetch-opcode-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-pc (*tm*)), *tm*))))
→ (tm-fetch-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*)))
= tm-fetch-opcode (*tm*))

EVENT: Disable tm-fetch-opcode-mapup-address-space.

THEOREM: tm-fetch-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-fetch-execute (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-fetch-execute (*tm*)),
tm-regs (tm-fetch-execute (*tm*)),
tm-cc (tm-fetch-execute (*tm*)),
tm-error (tm-fetch-execute (*tm*)),
tm-svcflag (tm-fetch-execute (*tm*)),
tm-svcid (tm-fetch-execute (*tm*)),
tm-base (tm-fetch-execute (*tm*)),
tm-limit (tm-fetch-execute (*tm*))))

EVENT: Disable tm-fetch-execute-commutes-with-mapup-address-space.

DEFINITION:
deq-ith-buffer (*id*, *buffers*) = putnth (deq (getnth (*id*, *buffers*)), *id*, *buffers*)

DEFINITION:
enq-ith-buffer (*item*, *id*, *buffers*)
= putnth (enq (*item*, getnth (*id*, *buffers*)), *id*, *buffers*)

DEFINITION:
qreplace-ith-buffer (*item*, *id*, *buffers*)
= putnth (qreplace (*item*, getnth (*id*, *buffers*)), *id*, *buffers*)

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: task-ibuffers (*task*) = car (task-channels (*task*))

DEFINITION: task-obuffers (*task*) = car (cdr (task-channels (*task*)))

DEFINITION:
task-mbuffers (*task*) = car (cdr (cdr (task-channels (*task*))))

DEFINITION: TASK-IBUFFER-CAPACITY = '4

DEFINITION: TASK-OBUFFER-CAPACITY = '4

DEFINITION: TASK-MBUFFER-CAPACITY = '4

DEFINITION: OS-SVC-SEND-ID = '0

DEFINITION: OS-SVC-RECEIVE-ID = '1

DEFINITION: OS-SVC-TYO-ID = '2

DEFINITION: OS-SVC-TYI-ID = '3

DEFINITION:
task-send-instructionp (*task*)
= ((tm-svcid (task-pstate (*task*)) **mod** '4) = '0)

DEFINITION:
task-receive-instructionp (*task*)
= ((tm-svcid (task-pstate (*task*)) **mod** '4) = '1)

DEFINITION:
task-tyo-instructionp (*task*)
= ((tm-svcid (task-pstate (*task*)) **mod** '4) = '2)

DEFINITION:
task-tyi-instructionp (*task*)
= ((tm-svcid (task-pstate (*task*)) **mod** '4) = '3)

DEFINITION:
task-update-control (*pstate*) = tm-set-svcflag ('0, *pstate*)

DEFINITION:
task-communicationp (*task*) = tm-svc-interruptp (task-pstate (*task*))

DEFINITION: os-srcid (*task*, *max*) = (tm-r2 (*task*) **mod** *max*)

DEFINITION: os-destid (*task*, *max*) = (tm-r2 (*task*) **mod** *max*)

DEFINITION: os-idevid (*task*, *max*) = (tm-r2 (*task*) **mod** *max*)

DEFINITION: os-odevid (*task*, *max*) = (tm-r2 (*task*) **mod** *max*)

DEFINITION: os-message (*task*) = tm-r3 (*task*)

DEFINITION:
os-store-message (*msg*, *task*) = tm-store (*msg*, '(1 3), *task*)

DEFINITION:
task-srcid (*task*)
= os-srcid (task-pstate (*task*), length (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-receive (*srcid*, *destid*, *task*)
= **if** qemptyp2 (*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** qemptyp (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))
∧ ((length (car (l)) = *length*)
∧ (good-task-buffer-list (car (l), *capacity*)
∧ good-task-buffer-table (cdr (l),
length,
capacity)))
else l = 'nil **endif**

DEFINITION:
good-task (*task*, *ilength*, *olength*, *mlength*)
= (task-shellp (*task*)
∧ (good-address-space (task-pstate (*task*),
length (tm-memory (task-pstate (*task*))))
∧ ((length (task-ibuffers (*task*)) = *ilength*)
∧ (good-task-buffer-list (task-ibuffers (*task*),
'4)
∧ ((length (task-obuffers (*task*))
= *olength*)
∧ (good-task-buffer-list (task-obuffers (*task*),
'4)
∧ ((length (task-mbuffers (*task*))
= *mlength*)
∧ good-task-buffer-table (task-mbuffers (*task*),
mlength,
'4)))))))

THEOREM: good-task-buffer-properties
good-task-buffer (*buffer*, *capacity*)
→ (plistp (*buffer*)
∧ ((length (*buffer*) < (1 + *capacity*))
∧ finite-number-listp (*buffer*, '65536)))

EVENT: Disable good-task-buffer-properties.

THEOREM: good-task-buffer-getnth
(good-task-buffer-list (l, *capacity*) ∧ (i < length (l)))
→ 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*) → plistp (l)

EVENT: Disable plistp-good-task-buffer-list.

THEOREM: good-task-buffer-list-putnth
(good-task-buffer-list (*l*, *capacity*)
 ∧ (plistp (*buffer*)
  ∧ ((length (*buffer*) < (1 + *capacity*))
   ∧ finite-number-listp (*buffer*, '65536))))
→ 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*) ∧ (*i* < length (*l*)))
→ (plistp (getnth (*i*, *l*))
  ∧ ((length (getnth (*i*, *l*)) = *length*)
   ∧ 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*)
 ∧ ((*i* < length (*l*)) ∧ (*j* < *length*)))
→ 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*)
 ∧ (plistp (*x*)
  ∧ ((length (*x*) = *length*)
   ∧ good-task-buffer-list (*x*, *capacity*))))
→ 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*)
 ∧ ((*i* < length (*l*))
  ∧ (plistp (*buffer*)
   ∧ ((length (*buffer*) < (1 + *capacity*))
    ∧ finite-number-listp (*buffer*, '65536)))))
→ 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*) → 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*))) **mod** '4) = '0)

DEFINITION:
ak-receive-instructionp (*ak*)
= ((tm-svcid (getnth (ak-taskid (*ak*), ak-pstates (*ak*))) **mod** '4) = '1)

DEFINITION:
ak-tyo-instructionp (*ak*)
= ((tm-svcid (getnth (ak-taskid (*ak*), ak-pstates (*ak*))) **mod** '4) = '2)

DEFINITION:
ak-tyi-instructionp (*ak*)
= ((tm-svcid (getnth (ak-taskid (*ak*), ak-pstates (*ak*))) **mod** '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*, *oports*) = tm-oport-idlep (*id*, *oports*)

DEFINITION:
ak-post-output-interrupt (*id*, *oports*) = tm-post-output-interrupt (*id*, *oports*)

DEFINITION:
ak-clear-input-interrupt (*id*, *iports*) = tm-clear-input-interrupt (*id*, *iports*)

DEFINITION:
ak-clear-output-interrupt (*id*, *oports*)
= tm-clear-output-interrupt (*id*, *oports*)

DEFINITION:
ak-start-output (*char*, *id*, *oports*) = tm-start-output (*char*, *id*, *oports*)

DEFINITION:
ak-input-interruptp (*ak*) = tm-some-input-interruptp (ak-iports (*ak*))

DEFINITION:
ak-output-interruptp (*ak*) = tm-some-output-interruptp (ak-oports (*ak*))

DEFINITION:
ak-interrupting-input-port (*iports*) = tm-interrupting-input-port (*iports*)

DEFINITION:
ak-interrupting-output-port (*oports*) = tm-interrupting-output-port (*oports*)

DEFINITION: AK-STATUS-FLAG-FIELD = '0

DEFINITION: AK-STATUS-TASKID-FIELD = '1

DEFINITION: AK-READY-STATUS = '0

DEFINITION: AK-ERROR-STATUS = '1

DEFINITION: AK-SEND-STATUS = '2

DEFINITION: AK-RECEIVE-STATUS = '3

DEFINITION: AK-OUTPUT-STATUS = '4

DEFINITION: AK-INPUT-STATUS = '5

DEFINITION: AK-STATUS-LUB = '6

DEFINITION: AK-STATUS-LENGTH = '2

DEFINITION:
ak-ready-statusp (*taskid*, *ak*)
= (getnth ('0, getnth (*taskid*, ak-status (*ak*))) = '0)

DEFINITION:
ak-waiting-to-sendp (*srcid*, *destid*, *ak*)
= ((getnth ('0, getnth (*srcid*, ak-status (*ak*))) = '2)
∧ (getnth ('1, getnth (*srcid*, ak-status (*ak*))) = *destid*))

DEFINITION:
ak-waiting-to-receivep (*srcid*, *destid*, *ak*)
= ((getnth ('0, getnth (*destid*, ak-status (*ak*))) = '3)
∧ (getnth ('1, getnth (*destid*, 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 qemptyp (ak-readyq (ak)) **then** '1
else '0 **endif**,
if qemptyp (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** qemptyp2 (*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-idlep (*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** qemptyp (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-receive (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 qemptyp (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 qemptyp (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 qemptyp (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 qemptyp (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: ak-clock-interruptp (*ak*) = (ak-clock (*ak*) ≃ 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-oports (*ak*))
else ak-oports (*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*)
∧ (tm-iportp (*iport*)
∧ (finite-numberp (tm-iinterrupt-flag (*iport*), '2)
∧ (finite-numberp (tm-ierror-flag (*iport*), '2)
∧ finite-numberp (tm-ichar (*iport*), '256)))))
→ 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*)
∧ (tm-oportp (*oport*)
∧ (finite-numberp (tm-ointerrupt-flag (*oport*), '2)
∧ (finite-numberp (tm-obusy-flag (*oport*), '2)
∧ finite-numberp (tm-ochar (*oport*), '256)))))
→ 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*))))
∧ good-address-space-list (cdr (*l*))
else *l* = 'nil **endif**

THEOREM: plistp-good-address-space-list
good-address-space-list (*l*) → plistp (*l*)

EVENT: Disable plistp-good-address-space-list.

THEOREM: good-address-space-getnth
(good-address-space-list (l) $\wedge$ (i < 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 $status$ = '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 < 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-IPORTS AK))
          (AND
           (EQUAL (LENGTH (AK-IPORTS AK)) '16)
           (AND
            (GOOD-TM-IPORT-ARRAY (AK-IPORTS AK))
            (AND
             (PLISTP (AK-OPORTS AK))
             (AND
              (EQUAL (LENGTH (AK-OPORTS AK)) '16)
              (AND
               (GOOD-TM-OPORT-ARRAY (AK-OPORTS 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 $(ak) \rightarrow$ ak-shellp (ak)

EVENT: Disable ak-shellp-ak.

THEOREM: length-ak-pstates
good-ak (*ak*) → (length (ak-pstates (*ak*)) = '16)

EVENT: Disable length-ak-pstates.

THEOREM: good-address-space-list-ak-pstates
good-ak (*ak*) → good-address-space-list (ak-pstates (*ak*))

EVENT: Disable good-address-space-list-ak-pstates.

THEOREM: good-address-space-getnth-from-ak
(good-ak (*ak*) ∧ (*i* < '16))
→ good-address-space (getnth (*i*, ak-pstates (*ak*)),
length (tm-memory (getnth (*i*, ak-pstates (*ak*)))))

EVENT: Disable good-address-space-getnth-from-ak.

THEOREM: length-ak-ibuffers
good-ak (*ak*) → (length (ak-ibuffers (*ak*)) = '16)

EVENT: Disable length-ak-ibuffers.

THEOREM: good-task-buffer-list-ak-ibuffers
good-ak (*ak*) → good-task-buffer-list (ak-ibuffers (*ak*), '4)

EVENT: Disable good-task-buffer-list-ak-ibuffers.

THEOREM: plistp-ak-ibuffers
good-ak (*ak*) → plistp (ak-ibuffers (*ak*))

EVENT: Disable plistp-ak-ibuffers.

THEOREM: good-task-buffer-getnth-from-ak-ibuffers
(good-ak (*ak*) ∧ finite-numberp (*id*, '16))
→ good-task-buffer (getnth (*id*, ak-ibuffers (*ak*)), '4)

EVENT: Disable good-task-buffer-getnth-from-ak-ibuffers.

THEOREM: length-ak-obuffers
good-ak (*ak*) → (length (ak-obuffers (*ak*)) = '16)

EVENT: Disable length-ak-obuffers.

THEOREM: good-task-buffer-list-ak-obuffers
good-ak (*ak*) → good-task-buffer-list (ak-obuffers (*ak*), '4)

EVENT: Disable good-task-buffer-list-ak-obuffers.

THEOREM: plistp-ak-obuffers
good-ak (*ak*) → plistp (ak-obuffers (*ak*))

EVENT: Disable plistp-ak-obuffers.

THEOREM: good-task-buffer-getnth-from-ak-obuffers
(good-ak (*ak*) ∧ finite-numberp (*id*, '16))
→ good-task-buffer (getnth (*id*, ak-obuffers (*ak*)), '4)

EVENT: Disable good-task-buffer-getnth-from-ak-obuffers.

THEOREM: length-ak-mbuffers
good-ak (*ak*) → (length (ak-mbuffers (*ak*)) = '16)

EVENT: Disable length-ak-mbuffers.

THEOREM: good-task-buffer-table-ak-mbuffers
good-ak (*ak*) → good-task-buffer-table (ak-mbuffers (*ak*), '16, '4)

EVENT: Disable good-task-buffer-table-ak-mbuffers.

THEOREM: plistp-ak-mbuffers
good-ak (*ak*) → plistp (ak-mbuffers (*ak*))

EVENT: Disable plistp-ak-mbuffers.

THEOREM: good-task-buffer-getnth2-from-ak-mbuffers
(good-ak (*ak*) ∧ ((*i* < '16) ∧ (*j* < '16)))
→ good-task-buffer (getnth2 (*i*, *j*, ak-mbuffers (*ak*)), '4)

EVENT: Disable good-task-buffer-getnth2-from-ak-mbuffers.

THEOREM: finite-numberp-ak-taskid
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ finite-numberp (ak-taskid (*ak*), '16)

EVENT: Disable finite-numberp-ak-taskid.

THEOREM: plistp-ak-readyq
good-ak (*ak*) → plistp (ak-readyq (*ak*))

EVENT: Disable plistp-ak-readyq.

THEOREM: length-ak-readyq
good-ak (*ak*) → (length (ak-readyq (*ak*)) < '17)

EVENT: Disable length-ak-readyq.

THEOREM: finite-number-listp-ak-readyq
good-ak (*ak*) → finite-number-listp (ak-readyq (*ak*), '16)

EVENT: Disable finite-number-listp-ak-readyq.

THEOREM: length-ak-status
good-ak (*ak*) → (length (ak-status (*ak*)) = '16)

EVENT: Disable length-ak-status.

THEOREM: good-status-list-ak-status
good-ak (*ak*) → good-status-list (ak-status (*ak*))

EVENT: Disable good-status-list-ak-status.

THEOREM: plistp-ak-status
good-ak (*ak*) → plistp (ak-status (*ak*))

EVENT: Disable plistp-ak-status.

THEOREM: good-status-getnth-from-ak-status
(good-ak (*ak*) ∧ (*i* < '16)) → good-status (getnth (*i*, ak-status (*ak*)))

EVENT: Disable good-status-getnth-from-ak-status.

THEOREM: finite-numberp-ak-rwstate
good-ak (*ak*) → finite-numberp (ak-rwstate (*ak*), '2)

EVENT: Disable finite-numberp-ak-rwstate.

THEOREM: finite-numberp-ak-clock
good-ak (*ak*) → finite-numberp (ak-clock (*ak*), '65536)

EVENT: Disable finite-numberp-ak-clock.

THEOREM: plistp-ak-iports
good-ak (*ak*) → plistp (ak-iports (*ak*))

EVENT: Disable plistp-ak-iports.

THEOREM: length-ak-iports
good-ak (*ak*) → (length (ak-iports (*ak*)) = '16)

EVENT: Disable length-ak-iports.

THEOREM: good-tm-iport-array-ak-iports
good-ak (*ak*) → good-tm-iport-array (ak-iports (*ak*))

EVENT: Disable good-tm-iport-array-ak-iports.

THEOREM: good-tm-iport-getnth-from-ak-iports
(good-ak (*ak*) ∧ (*i* < '16)) → good-tm-iport (getnth (*i*, ak-iports (*ak*)))

EVENT: Disable good-tm-iport-getnth-from-ak-iports.

THEOREM: plistp-ak-oports
good-ak (*ak*) → plistp (ak-oports (*ak*))

EVENT: Disable plistp-ak-oports.

THEOREM: length-ak-oports
good-ak (*ak*) → (length (ak-oports (*ak*)) = '16)

EVENT: Disable length-ak-oports.

THEOREM: good-tm-oport-array-ak-oports
good-ak (*ak*) → good-tm-oport-array (ak-oports (*ak*))

EVENT: Disable good-tm-oport-array-ak-oports.

THEOREM: good-tm-oport-getnth-from-ak-oports
(good-ak (*ak*) ∧ (*i* < '16)) → good-tm-oport (getnth (*i*, ak-oports (*ak*)))

EVENT: Disable good-tm-oport-getnth-from-ak-oports.

THEOREM: permutation-ak-readyq-ak-ready-set
good-ak (*ak*) → permutation (ak-readyq (*ak*), ak-ready-set (*ak*))

EVENT: Disable permutation-ak-readyq-ak-ready-set.

THEOREM: ak-waiting-iff-qemptyp-ak-readyq
good-ak (*ak*) → (ak-waiting (*ak*) ↔ qemptyp (ak-readyq (*ak*)))

EVENT: Disable ak-waiting-iff-qemptyp-ak-readyq.

EVENT: Let us define the theory *good-ak-properties* to consist of the following events: ak-waiting-iff-qemptyp-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-qemptyp-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*) → (*i* ∉ index-ready-set (*n*, *status*))

EVENT: Disable not-member-index-ready-set.

THEOREM: setp-index-ready-set
setp (index-ready-set (*n*, *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
$((i < \text{length}\,(l))$
$\wedge$ (status-flag-readyp (getnth $(i,\ l)) \wedge (\neg$ status-flag-readyp $(v))))$
$\rightarrow$ (index-ready-set $(n,$ putnth $(v,\ i,\ l))$
$=$ remove $(n + i,$ index-ready-set $(n,\ l)))$

EVENT: Disable index-ready-set-putnth-unready-status.

THEOREM: not-status-flag-readyp-list
$(\mathit{flag} \neq$ `'0`$) \rightarrow (\neg$ status-flag-readyp (cons $(\mathit{flag},$ cons $(\mathit{taskid},$ `'nil`$))))$

EVENT: Disable not-status-flag-readyp-list.

THEOREM: status-flag-readyp-iff-member-index-ready-set1
$(i <$ length $(\mathit{status}))$
$\rightarrow$ (status-flag-readyp (getnth $(i,\ \mathit{status})$)
$=$ $((n + i) \in$ index-ready-set $(n,\ \mathit{status})))$

THEOREM: status-flag-readyp-iff-member-index-ready-set
$((i \in \mathbf{N}) \wedge (i <$ length $(\mathit{status})))$
$\rightarrow$ (status-flag-readyp (getnth $(i,\ \mathit{status})$)
$=$ $(i \in$ index-ready-set (`'0`, $\mathit{status})))$

EVENT: Disable status-flag-readyp-iff-member-index-ready-set.

THEOREM: status-flag-readyp-getnth
(permutation $(\mathit{queue},$ index-ready-set (`'0`, $\mathit{status})$)
$\wedge$ $((\neg$ qemptyp $(\mathit{queue}))$
$\wedge$ (number-listp $(\mathit{queue}) \wedge$ (qfirst $(\mathit{queue}) <$ length $(\mathit{status})))))$
$\rightarrow$ status-flag-readyp (getnth (qfirst $(\mathit{queue}),\ \mathit{status})$)

EVENT: Disable status-flag-readyp-getnth.

THEOREM: permutation-deq-index-ready-set-putnth
(permutation $(\mathit{queue},$ index-ready-set (`'0`, $\mathit{status})$)
$\wedge$ (number-listp (queue)
$\wedge$ (plistp (queue)

$\wedge$ $((\neg$ qemptyp $(queue))$
$\wedge$ $((taskid =$ qfirst $(queue))$
$\wedge$ $((flag \neq$ 'O$)$
$\wedge$ $(taskid <$ length $(status)))))))$
$\rightarrow$ permutation (deq ($queue$),
index-ready-set ('O,
putnth (cons ($flag$, cons (x, 'nil)),
$taskid$,
$status$)))

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 'O **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 'O **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 $(status)) \wedge (\neg$ status-flag-readyp (getnth (i, $status$))))$
$\rightarrow$ ('O < number-of-unready-tasks ($status$))

EVENT: Disable number-of-unready-tasks-non-zero.

THEOREM: not-equal-number-of-ready-tasks-length-status
$((i <$ length $(status)) \wedge (\neg$ status-flag-readyp (getnth (i, $status$))))$
$\rightarrow$ (number-of-ready-tasks ($status$) $\neq$ length ($status$))

THEOREM: lessp-number-of-ready-tasks-length-status
$((i < \text{length}(\mathit{status})) \wedge (\neg\ \text{status-flag-readyp}(\text{getnth}(i,\ \mathit{status}))))$
$\rightarrow\quad (\text{number-of-ready-tasks}(\mathit{status}) < \text{length}(\mathit{status}))$

THEOREM: number-of-ready-tasks-equals-length-index-ready-set
$\text{number-of-ready-tasks}(\mathit{status}) = \text{length}(\text{index-ready-set}(n,\ \mathit{status}))$

THEOREM: status-flag-readyp-list
$\text{status-flag-readyp}(\text{cons}(\texttt{'0},\ \text{cons}(x,\ \texttt{'nil})))$

EVENT: Disable status-flag-readyp-list.

THEOREM: lessp-car-index-ready-set
$((i < n) \wedge \text{listp}(\text{index-ready-set}(n,\ l)))$
$\rightarrow\quad (i < \text{car}(\text{index-ready-set}(n,\ l)))$

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
$\text{listp}(\text{index-ready-set}(1 + n,\ z))$
$\rightarrow\quad (\text{insert}(n,\ \text{index-ready-set}(1 + n,\ z))$
$\qquad =\quad \text{cons}(n,\ \text{index-ready-set}(1 + n,\ z)))$

THEOREM: rewrite-insert-to-cons-crock2
$(\neg\ \text{listp}(\text{index-ready-set}(1 + n,\ z)))$
$\rightarrow\quad (\text{insert}(n,\ \text{index-ready-set}(1 + n,\ z))$
$\qquad =\quad \text{cons}(n,\ \text{index-ready-set}(1 + n,\ z)))$

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
$((i < \text{length}(l))$
$\wedge\quad ((\neg\ \text{status-flag-readyp}(\text{getnth}(i,\ l))) \wedge \text{status-flag-readyp}(v)))$
$\rightarrow\quad (\text{index-ready-set}(n,\ \text{putnth}(v,\ i,\ l))$
$\qquad =\quad \text{insert}(n + i,\ \text{index-ready-set}(n,\ l)))$

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, $status$))
 ∧ ((¬ status-flag-readyp (getnth ($taskid$, $status$)))
  ∧ (plistp ($queue$)
   ∧ (($taskid$ ∈ **N**) ∧ ($taskid$ < length ($status$))))))
→ 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))
→ (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)
 ∧ (ak-waiting-to-sendp ($srcid$, $destid$, ak) ∧ finite-numberp ($srcid$, '16)))
→ (¬ qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-receivep-implies-readyq-not-full
(good-ak (ak)
 ∧ (ak-waiting-to-receivep ($srcid$, $destid$, ak)
  ∧ finite-numberp ($destid$, '16)))
→ (¬ qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-outputp-implies-readyq-not-full
(good-ak (ak) ∧ (ak-waiting-to-outputp (id, ak) ∧ finite-numberp (id, '16)))
→ (¬ qfullp (ak-readyq (ak), '16))

THEOREM: ak-waiting-to-inputp-implies-readyq-not-full
(good-ak (ak) ∧ (ak-waiting-to-inputp (id, ak) ∧ finite-numberp (id, '16)))
→ (¬ qfullp (ak-readyq (ak), '16))

THEOREM: good-ak-ak-error-handler
(good-ak (ak) ∧ (¬ ak-waiting (ak))) → good-ak (ak-error-handler (ak))

EVENT: Disable good-ak-ak-error-handler.

THEOREM: good-ak-ak-clock-interrupt-handler
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ good-ak (ak-clock-interrupt-handler (*ak*))

EVENT: Disable good-ak-ak-clock-interrupt-handler.

THEOREM: good-ak-ak-block-send
(good-ak (*ak*)
∧ ((¬ ak-waiting (*ak*))
∧ ((*srcid* = ak-taskid (*ak*)) ∧ finite-numberp (*destid*, '16'))))
→ 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*)
→ (¬ status-flag-readyp (getnth (*destid*, ak-status (*ak*))))

THEOREM: good-ak-ak-execute-send-to-buffer
(good-ak (*ak*)
∧ ((¬ ak-waiting (*ak*))
∧ ((¬ qfullp2 (*srcid*, *destid*, ak-mbuffers (*ak*), '4'))
∧ (finite-numberp (*srcid*, '16')
∧ (finite-numberp (*destid*, '16')
∧ finite-numberp (*msg*, '65536'))))))
→ 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*)
∧ ((¬ ak-waiting (*ak*))
∧ (finite-numberp (*destid*, '16')
∧ finite-numberp (*msg*, '65536'))))
→ 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*)

$\wedge$ ((¬ ak-waiting (*ak*))
$\wedge$ ((*destid* = ak-taskid (*ak*)) $\wedge$ finite-numberp (*srcid*, '16))))
$\rightarrow$ good-ak (ak-block-receive (*srcid*, *destid*, *ak*))

EVENT: Disable good-ak-ak-block-receive.

THEOREM: ak-waiting-to-sendp-implies-not-status-flag-readyp
ak-waiting-to-sendp (*srcid*, *destid*, *ak*)
$\rightarrow$ (¬ status-flag-readyp (getnth (*srcid*, ak-status (*ak*))))

THEOREM: good-status-properties
good-status (*status*)
$\rightarrow$ (plistp (*status*)
$\wedge$ ((length (*status*) = '2)
$\wedge$ (finite-numberp (getnth ('0, *status*), '6)
$\wedge$ finite-numberp (getnth ('1, *status*), '16))))

EVENT: Disable good-status-properties.

THEOREM: good-ak-ak-execute-receive-from-buffer
(good-ak (*ak*)
$\wedge$ ((¬ ak-waiting (*ak*))
$\wedge$ ((¬ qemptyp2 (*srcid*, *destid*, ak-mbuffers (*ak*)))
$\wedge$ (finite-numberp (*srcid*, '16)
$\wedge$ finite-numberp (*destid*, '16)))))
$\rightarrow$ good-ak (ak-execute-receive-from-buffer (*srcid*, *destid*, *ak*))

EVENT: Disable good-ak-ak-execute-receive-from-buffer.

THEOREM: good-ak-ak-execute-receive
(good-ak (*ak*) $\wedge$ ((¬ ak-waiting (*ak*)) $\wedge$ finite-numberp (*srcid*, '16)))
$\rightarrow$ good-ak (ak-execute-receive (*srcid*, ak-taskid (*ak*), *ak*))

EVENT: Disable good-ak-ak-execute-receive.

THEOREM: good-ak-ak-block-output
(good-ak (*ak*) $\wedge$ ((¬ ak-waiting (*ak*)) $\wedge$ (*id* = ak-taskid (*ak*))))
$\rightarrow$ good-ak (ak-block-output (*id*, *ak*))

EVENT: Disable good-ak-ak-block-output.

THEOREM: good-ak-obuffers-properties
good-ak (*ak*)
$\rightarrow$ (plistp (ak-obuffers (*ak*)) $\wedge$ (length (ak-obuffers (*ak*)) = '16))

EVENT: Disable good-ak-obuffers-properties.

THEOREM: good-ak-ak-execute-output-to-buffer
(good-ak (*ak*)
 ∧ ((¬ ak-waiting (*ak*))
  ∧ ((¬ qfullp (getnth (*id*, ak-obuffers (*ak*)), '4))
   ∧ (finite-numberp (*id*, '16)
    ∧ finite-numberp (*char*, '65536)))))
→ 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*) ∧ ((¬ ak-waiting (*ak*)) ∧ finite-numberp (*char*, '65536)))
→ 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*) ∧ ((¬ ak-waiting (*ak*)) ∧ (*id* = ak-taskid (*ak*))))
→ good-ak (ak-block-input (*id*, *ak*))

EVENT: Disable good-ak-ak-block-input.

THEOREM: good-ak-ibuffers-properties
good-ak (*ak*)
→ (plistp (ak-ibuffers (*ak*)) ∧ (length (ak-ibuffers (*ak*)) = '16))

EVENT: Disable good-ak-ibuffers-properties.

THEOREM: good-ak-ak-execute-input-from-buffer
(good-ak (*ak*)
 ∧ ((¬ ak-waiting (*ak*))
  ∧ (finite-numberp (*id*, '16)
   ∧ (¬ qemptyp (getnth (*id*, ak-ibuffers (*ak*)))))))
→ 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*) ∧ (¬ ak-waiting (*ak*)))
→ good-ak (ak-execute-input (ak-taskid (*ak*), *ak*))

EVENT: Disable good-ak-ak-execute-input.

THEOREM: finite-numberp-ak-srcid
good-ak (*ak*) → finite-numberp (ak-srcid (*ak*), '16)

EVENT: Disable finite-numberp-ak-srcid.

THEOREM: finite-numberp-ak-destid
good-ak (*ak*) → finite-numberp (ak-destid (*ak*), '16)

EVENT: Disable finite-numberp-ak-destid.

THEOREM: finite-numberp-ak-message
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ finite-numberp (ak-message (*ak*), '65536)

EVENT: Disable finite-numberp-ak-message.

THEOREM: good-ak-ak-svc-handler
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*))) → good-ak (ak-svc-handler (*ak*))

EVENT: Disable good-ak-ak-svc-handler.

THEOREM: good-ak-ak-private-step
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*))) → good-ak (ak-private-step (*ak*))

EVENT: Disable good-ak-ak-private-step.

THEOREM: ak-waiting-to-inputp-implies-not-status-flag-readyp
ak-waiting-to-inputp (*id*, *ak*)
→ (¬ status-flag-readyp (getnth (*id*, ak-status (*ak*))))

THEOREM: ak-waiting-to-outputp-implies-not-status-flag-readyp
ak-waiting-to-outputp (*id*, *ak*)
→ (¬ status-flag-readyp (getnth (*id*, ak-status (*ak*))))

THEOREM: good-tm-iport-properties
good-tm-iport (*port*)
→ (tm-iportp (*port*)
∧ (finite-numberp (tm-iinterrupt-flag (*port*), '2)
∧ (finite-numberp (tm-ierror-flag (*port*), '2)
∧ finite-numberp (tm-ichar (*port*), '256))))

EVENT: Disable good-tm-iport-properties.

THEOREM: good-ak-ak-waiting-input-interrupt-handler-case1
(good-ak (*ak*)
 ∧ (finite-numberp (*id*, '16)
  ∧ (ak-waiting (*ak*) ∧ ak-waiting-to-inputp (*id*, *ak*))))
→ good-ak (ak-waiting-input-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-waiting-input-interrupt-handler-case2
(good-ak (*ak*)
 ∧ (finite-numberp (*id*, '16)
  ∧ (ak-waiting (*ak*) ∧ (¬ ak-waiting-to-inputp (*id*, *ak*)))))
→ good-ak (ak-waiting-input-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-waiting-input-interrupt-handler
(good-ak (*ak*) ∧ (finite-numberp (*id*, '16) ∧ ak-waiting (*ak*)))
→ 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*)
 ∧ (finite-numberp (*id*, '16)
  ∧ ((¬ ak-waiting (*ak*)) ∧ ak-waiting-to-inputp (*id*, *ak*))))
→ good-ak (ak-running-input-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-running-input-interrupt-handler-case2
(good-ak (*ak*)
 ∧ (finite-numberp (*id*, '16)
  ∧ ((¬ ak-waiting (*ak*)) ∧ (¬ ak-waiting-to-inputp (*id*, *ak*)))))
→ good-ak (ak-running-input-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-running-input-interrupt-handler
(good-ak (*ak*) ∧ (finite-numberp (*id*, '16) ∧ (¬ ak-waiting (*ak*))))
→ 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*) ∧ tm-some-input-interruptp (ak-iports (*ak*)))
→ finite-numberp (tm-interrupting-input-port (ak-iports (*ak*)), '16)

EVENT: Disable finite-numberp-interrupting-ak-input-port.

THEOREM: good-ak-ak-input-interrupt-handler
(good-ak (*ak*) ∧ ak-input-interruptp (*ak*))
→ good-ak (ak-input-interrupt-handler (ak-interrupting-input-port (ak-iports (*ak*)),
ak))

EVENT: Disable good-ak-ak-input-interrupt-handler.

THEOREM: good-tm-oport-properties
good-tm-oport (*port*)
→ (tm-oportp (*port*)
∧ (finite-numberp (tm-ointerrupt-flag (*port*), '2)
∧ (finite-numberp (tm-obusy-flag (*port*), '2)
∧ finite-numberp (tm-ochar (*port*), '256))))

EVENT: Disable good-tm-oport-properties.

THEOREM: good-ak-ak-waiting-output-interrupt-handler-case1
(good-ak (*ak*)
∧ (finite-numberp (*id*, '16)
∧ (ak-waiting (*ak*) ∧ ak-waiting-to-outputp (*id*, *ak*))))
→ good-ak (ak-waiting-output-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-waiting-output-interrupt-handler-case2
(good-ak (*ak*)
∧ (finite-numberp (*id*, '16)
∧ (ak-waiting (*ak*) ∧ (¬ ak-waiting-to-outputp (*id*, *ak*)))))
→ good-ak (ak-waiting-output-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-waiting-output-interrupt-handler
(good-ak (*ak*) ∧ (finite-numberp (*id*, '16) ∧ ak-waiting (*ak*)))
→ good-ak (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
(good-ak (*ak*)
∧ (finite-numberp (*id*, '16)
∧ ((¬ ak-waiting (*ak*)) ∧ ak-waiting-to-outputp (*id*, *ak*))))
→ good-ak (ak-running-output-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-running-output-interrupt-handler-case2
(good-ak (*ak*)
∧ (finite-numberp (*id*, '16)
∧ ((¬ ak-waiting (*ak*)) ∧ (¬ ak-waiting-to-outputp (*id*, *ak*)))))
→ good-ak (ak-running-output-interrupt-handler (*id*, *ak*))

THEOREM: good-ak-ak-running-output-interrupt-handler
(good-ak (*ak*) ∧ (finite-numberp (*id*, '16) ∧ (¬ ak-waiting (*ak*))))
→ good-ak (ak-running-output-interrupt-handler (*id*, *ak*))

EVENT: Disable good-ak-ak-running-output-interrupt-handler.

THEOREM: finite-numberp-interrupting-ak-output-port
(good-ak (*ak*) ∧ tm-some-output-interruptp (ak-oports (*ak*)))
→ finite-numberp (tm-interrupting-output-port (ak-oports (*ak*)), '16)

EVENT: Disable finite-numberp-interrupting-ak-output-port.

THEOREM: good-ak-ak-output-interrupt-handler
(good-ak (*ak*) ∧ ak-output-interruptp (*ak*))
→ good-ak (ak-output-interrupt-handler (ak-interrupting-output-port (ak-oports (*ak*)),
ak))

EVENT: Disable good-ak-ak-output-interrupt-handler.

THEOREM: good-ak-ak-step
good-ak (*ak*) → good-ak (ak-step (*ak*))

EVENT: Disable good-ak-ak-step.

THEOREM: good-ak-ak-post-input-interrupt
(good-ak (*ak*) ∧ tm-device-input-eventp (*interrupt*))
→ good-ak (ak-post-interrupt (*interrupt*, *ak*))

EVENT: Disable good-ak-ak-post-input-interrupt.

THEOREM: good-ak-ak-post-output-interrupt
(good-ak (*ak*) ∧ tm-device-output-eventp (*interrupt*))
→ good-ak (ak-post-interrupt (*interrupt*, *ak*))

EVENT: Disable good-ak-ak-post-output-interrupt.

THEOREM: good-ak-ak-post-interrupt-noop
(good-ak (*ak*)
∧ ((¬ tm-device-input-eventp (*interrupt*))
∧ (¬ tm-device-output-eventp (*interrupt*))))
→ good-ak (ak-post-interrupt (*interrupt*, *ak*))

EVENT: Disable good-ak-ak-post-interrupt-noop.

THEOREM: good-ak-ak-post-interrupt
good-ak (*ak*) → good-ak (ak-post-interrupt (*interrupt*, *ak*))

EVENT: Disable good-ak-ak-post-interrupt.

THEOREM: good-ak-ak-processor
good-ak (*ak*) → good-ak (ak-processor (*ak*, *oracle*))

EVENT: Disable good-ak-ak-processor.

THEOREM: ak-pstates-ak-clock-interrupt-handler
ak-pstates (ak-clock-interrupt-handler (*ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-clock-interrupt-handler.

THEOREM: ak-pstates-ak-error-handler
ak-pstates (ak-error-handler (*ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-error-handler.

THEOREM: ak-pstates-ak-block-send
ak-pstates (ak-block-send (*srcid*, *destid*, *ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-block-send.

THEOREM: length-ak-pstates-ak-execute-send-to-buffer
length (ak-pstates (ak-execute-send-to-buffer (*msg*, *srcid*, *destid*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-send-to-buffer.

THEOREM: length-ak-pstates-ak-execute-send
length (ak-pstates (ak-execute-send (*msg*, *srcid*, *destid*, *ak*)))
= length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-execute-send.

THEOREM: ak-pstates-ak-block-receive
ak-pstates (ak-block-receive (*srcid*, *destid*, *ak*)) = 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
length (ak-pstates (ak-svc-handler (*ak*))) = length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-svc-handler.

THEOREM: ak-pstates-ak-input-interrupt-handler
ak-pstates (ak-input-interrupt-handler (*id*, *ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-input-interrupt-handler.

THEOREM: ak-pstates-ak-output-interrupt-handler
ak-pstates (ak-output-interrupt-handler (*id*, *ak*)) = ak-pstates (*ak*)

EVENT: Disable ak-pstates-ak-output-interrupt-handler.

THEOREM: length-ak-pstates-ak-private-step
length (ak-pstates (ak-private-step (*ak*))) = length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-private-step.

THEOREM: length-ak-pstates-ak-step
length (ak-pstates (ak-step (*ak*))) = length (ak-pstates (*ak*))

EVENT: Disable length-ak-pstates-ak-step.

DEFINITION:
ak-channels (*ak*)
= cons (ak-ibuffers (*ak*),
cons (ak-obuffers (*ak*), cons (ak-mbuffers (*ak*), 'nil)))

DEFINITION:
project (*i*, *ak*) = task (getnth (*i*, ak-pstates (*ak*)), ak-channels (*ak*))

DEFINITION:
control-oracle-step (*i*, *ak*)
= **if** ak-input-interruptp (*ak*)
then ak-channels (ak-input-interrupt-handler (tm-interrupting-input-port (ak-iports (*ak*)),
ak))
elseif ak-output-interruptp (*ak*)
then ak-channels (ak-output-interrupt-handler (tm-interrupting-output-port (ak-oports (*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*))
∧ ((ak-ibuffers (ak-post-interrupt (*interrupt*, *ak*)) = ak-ibuffers (*ak*))
∧ ((ak-obuffers (ak-post-interrupt (*interrupt*, *ak*))
= ak-obuffers (*ak*))
∧ ((ak-mbuffers (ak-post-interrupt (*interrupt*, *ak*))
= ak-mbuffers (*ak*))
∧ (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*))
∧ ((ak-obuffers (ak-input-interrupt-handler (*id*, *ak*)) = ak-obuffers (*ak*))
∧ (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*))
∧ ((ak-ibuffers (ak-output-interrupt-handler (*id*, *ak*)) = ak-ibuffers (*ak*))
∧ (ak-mbuffers (ak-output-interrupt-handler (*id*, *ak*))
= ak-mbuffers (*ak*)))

EVENT: Disable access-ak-output-interrupt-handler.

THEOREM: access-ak-error-handler
(ak-pstates (ak-error-handler (*ak*)) = ak-pstates (*ak*))
∧ ((ak-ibuffers (ak-error-handler (*ak*)) = ak-ibuffers (*ak*))
∧ ((ak-obuffers (ak-error-handler (*ak*)) = ak-obuffers (*ak*))
∧ (ak-mbuffers (ak-error-handler (*ak*))
= ak-mbuffers (*ak*))))

EVENT: Disable access-ak-error-handler.

THEOREM: ak-private-step-non-interference
(good-ak (*ak*)
∧ (finite-numberp (*i*, length (ak-pstates (*ak*)))
∧ ((*i* ≠ ak-taskid (*ak*)) ∧ (¬ ak-waiting (*ak*)))))
→ (getnth (*i*, ak-pstates (ak-private-step (*ak*)))
= getnth (*i*, ak-pstates (*ak*)))

EVENT: Disable ak-private-step-non-interference.

THEOREM: ak-svc-handler-non-interference
(good-ak (*ak*)
∧ (finite-numberp (*i*, length (ak-pstates (*ak*)))
∧ ((*i* ≠ ak-taskid (*ak*)) ∧ (¬ ak-waiting (*ak*)))))
→ (getnth (*i*, ak-pstates (ak-svc-handler (*ak*)))
= getnth (*i*, ak-pstates (*ak*)))

EVENT: Disable ak-svc-handler-non-interference.

THEOREM: ak-implements-non-active-task-step
(good-ak (*ak*)
∧ (finite-numberp (*i*, length (ak-pstates (*ak*)))
∧ (¬ task-activep (control-oracle-step (*i*,
ak-post-interrupt (car (*oracle*),
ak))))))
→ (project (*i*, ak-step (ak-post-interrupt (car (*oracle*), *ak*)))
= task-update-channels (project (*i*, *ak*),
control-oracle-step (*i*,
ak-post-interrupt (car (*oracle*),
ak)))))

EVENT: Disable ak-implements-non-active-task-step.

THEOREM: ak-execute-receive-implements-task-execute-receive
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ (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
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ (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
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ (task (getnth (ak-taskid (*ak*),
ak-pstates (ak-execute-input (ak-taskid (*ak*), *ak*))),

ak-channels (ak-execute-input (ak-taskid (*ak*), *ak*)))
= task-execute-input (ak-taskid (*ak*),
task (getnth (ak-taskid (*ak*), ak-pstates (*ak*)),
ak-channels (*ak*))))

EVENT: Disable ak-execute-input-implements-task-execute-input.

THEOREM: ak-execute-output-implements-task-execute-output
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ (task (getnth (ak-taskid (*ak*),
ak-pstates (ak-execute-output (ak-message (*ak*),
ak-taskid (*ak*),
ak))),
ak-channels (ak-execute-output (ak-message (*ak*), ak-taskid (*ak*), *ak*)))
= task-execute-output (task-message (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-output-implements-task-execute-output.

THEOREM: ak-svc-handler-implements-task-communication-step
(good-ak (*ak*) ∧ (¬ ak-waiting (*ak*)))
→ (task (getnth (ak-taskid (*ak*), ak-pstates (ak-svc-handler (*ak*))),
ak-channels (ak-svc-handler (*ak*)))
= task-communication-step (task (getnth (ak-taskid (*ak*),
ak-pstates (*ak*)),
ak-channels (*ak*)),
ak-taskid (*ak*)))

EVENT: Disable ak-svc-handler-implements-task-communication-step.

THEOREM: ak-private-step-implements-task-fetch-execute
(good-ak (*ak*)
∧ ((¬ ak-waiting (*ak*))
∧ (finite-numberp (*i*, length (ak-pstates (*ak*)))
∧ (*i* = ak-taskid (*ak*)))))
→ (task (getnth (*i*, ak-pstates (ak-private-step (*ak*))),
ak-channels (ak-private-step (*ak*)))
= task (tm-fetch-execute (getnth (*i*, ak-pstates (*ak*))),
ak-channels (*ak*)))

EVENT: Disable ak-private-step-implements-task-fetch-execute.

THEOREM: ak-implements-active-task-step
(good-ak (*ak*)
∧ (finite-numberp (*i*, length (ak-pstates (*ak*)))
∧ task-activep (control-oracle-step (*i*,
ak-post-interrupt (car (*oracle*),
ak)))))
→ (project (*i*, ak-step (ak-post-interrupt (car (*oracle*), *ak*)))
= task-step (project (*i*, *ak*), *i*))

EVENT: Disable ak-implements-active-task-step.

THEOREM: ak-implements-parallel-tasks
(good-ak (*ak*) ∧ finite-numberp (*i*, length (ak-pstates (*ak*))))
→ (project (*i*, ak-processor (*ak*, *oracle*))
= task-processor (project (*i*, *ak*), *i*, control-oracle (*i*, *ak*, *oracle*)))

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 $(\text{car}\ (alist)) \wedge (x = \text{car}\ (\text{car}\ (alist)))$
then cdr $(\text{car}\ (alist))$
else lookup $(x,\ \text{cdr}\ (alist))$ **endif**
else `'nil` **endif**

DEFINITION:
boundp $(x,\ alist)$
$=$ **if** listp $(alist)$
then if listp $(\text{car}\ (alist))$
then if $x = \text{car}\ (\text{car}\ (alist))$ **then** `'*1*true`
else boundp $(x,\ \text{cdr}\ (alist))$ **endif**
else boundp $(x,\ \text{cdr}\ (alist))$ **endif**
else `'*1*false` **endif**

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

DEFINITION:
dc-formp (x)
$=$ $(\text{listp}\ (x)$
$\wedge$ $((\text{car}\ (x) =$ `'dc`$)$
$\wedge$ $((\text{car}\ (\text{cdr}\ (x)) \in \mathbf{N})$
$\wedge$ $((\text{car}\ (\text{cdr}\ (\text{cdr}\ (x))) \in \mathbf{N})$
$\vee$ $(\text{litatom}\ (\text{car}\ (\text{cdr}\ (\text{cdr}\ (x)))))$
$\vee$ $(\text{listp}\ (\text{car}\ (\text{cdr}\ (\text{cdr}\ (x)))))$
$\vee$ number-and-litatom-listp $(\text{car}\ (\text{cdr}\ (\text{cdr}\ (x))))))))))$

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-mode1 (*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 (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)))
```
$\wedge$ ((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**)
∧ (finite-numberp (asm-instruction-mode1 (*instruction*, *symtab*), '4)
∧ (finite-numberp (asm-instruction-mode2 (*instruction*, *symtab*), '4)
∧ (asm-good-arg (asm-instruction-arg1 (*instruction*, *symtab*), asm-instruction-mode1 (*instruction*, *symtab*))
∧ (asm-good-arg (asm-instruction-arg2 (*instruction*, *symtab*), asm-instruction-mode2 (*instruction*, *symtab*))
∧ (finite-numberp (asm-instruction-index1 (*instruction*, *symtab*), '8)
∧ 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)) ∧ (car (car (source)) = 'dcl)
         then translate-source (cdr (source), symtab)
         elseif listp (car (source)) ∧ (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*)) ∧ (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)) ∧ (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)) ∈ 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)
=   (¬ (plistp (assembler-result)
        ∧   ((length (assembler-result) = '2)
             ∧   (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 qemptyp)
(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 qemptyp)
(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 qemptyp)
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 qemptyp)
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 qemptyp)
(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 qemptyp)
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)
qemptyp
(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)
  (qemptyp . 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-iport-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)
(qcurrlength-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:
nthcdr-with-small-recursion $(n,\ l,\ max)$
= nthcdr $(n$ **mod** $max,$ nthcdr-n-times $(n \div max,\ max,\ l))$

DEFINITION:
getseg-with-small-recursion $(n,\ k,\ l,\ max)$
= firstn-with-small-recursion $(k,$
nthcdr-with-small-recursion $(n,\ l,\ max),$
$max)$

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
length (os-readyq (*os*)) = '20

EVENT: Disable length-os-readyq.

THEOREM: plistp-os-readyq
plistp (os-readyq (*os*))

EVENT: Disable plistp-os-readyq.

THEOREM: length-os-segment-table
length (os-segment-table (*os*)) = '32

EVENT: Disable length-os-segment-table.

THEOREM: plistp-os-segment-table
plistp (os-segment-table (*os*))

EVENT: Disable plistp-os-segment-table.

THEOREM: length-os-status-table
length (os-status-table (*os*)) = '32

EVENT: Disable length-os-status-table.

THEOREM: plistp-os-status-table
plistp (os-status-table (*os*))

EVENT: Disable plistp-os-status-table.

THEOREM: length-os-ibuffers
length (os-ibuffers (*os*)) = '128

EVENT: Disable length-os-ibuffers.

THEOREM: plistp-os-ibuffers
plistp (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-QEMPTYP-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*)
∧ ((getnth ('3, *queue*) = *capacity*)
∧ finite-number-listp (getseg ('4, *capacity*, *queue*), *lub*)))

THEOREM: array-queuep-finite-number-queue
finite-number-queuep (*queue*, *capacity*, *lub*) → array-queuep (*queue*)

EVENT: Disable array-queuep-finite-number-queue.

THEOREM: finite-number-queue-qmaxlength-field
finite-number-queuep (*queue*, *capacity*, *lub*)
→ (getnth ('3, *queue*) = *capacity*)

EVENT: Disable finite-number-queue-qmaxlength-field.

THEOREM: finite-number-listp-qarray
finite-number-queuep (*queue*, *capacity*, *lub*)
→ 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*)
→ (length (*queue*) = ('4 + *capacity*))

THEOREM: plistp-finite-number-queue
finite-number-queuep (*queue*, *capacity*, *lub*) → plistp (*queue*)

THEOREM: finite-number-queue-qhead-size
finite-number-queuep (*queue*, *capacity*, *lub*)
→ ((getnth ('0, *queue*) ∈ **N**) ∧ (getnth ('0, *queue*) < *capacity*))

EVENT: Disable finite-number-queue-qhead-size.

THEOREM: finite-number-queue-qtail-size
finite-number-queuep (*queue*, *capacity*, *lub*)
→ ((getnth ('1, *queue*) ∈ **N**) ∧ (getnth ('1, *queue*) < *capacity*))

EVENT: Disable finite-number-queue-qtail-size.

THEOREM: finite-number-queue-qcurrlength-size
finite-number-queuep (*queue*, *capacity*, *lub*)
→ ((getnth ('2, *queue*) ∈ **N**) ∧ (getnth ('2, *queue*) < (1 + *capacity*)))

EVENT: Disable finite-number-queue-qcurrlength-size.

THEOREM: finite-number-queue-qmaxlength-value
finite-number-queuep (*queue*, *capacity*, *lub*)
→ (getnth ('3, *queue*) = *capacity*)

EVENT: Disable finite-number-queue-qmaxlength-value.

THEOREM: finite-number-queue-qarray-after-enq
finite-number-queuep (*queue*, *capacity*, *lub*)
→ (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*)
∧ ((¬ array-qfullp (*queue*)) ∧ finite-numberp (*item*, *lub*)))
→ 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*)
→ (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*) ∧ (¬ array-qemptyp (*queue*)))
→ 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*) ∧ (¬ array-qemptyp (*queue*)))
→ ((array-qfirst (*queue*) ∈ **N**) ∧ (array-qfirst (*queue*) < *lub*))

EVENT: Disable array-qfirst-finite-number-queue.

THEOREM: remainder-array-qfirst-finite-number-queue
(finite-number-queuep (*queue*, *capacity*, *lub*) ∧ (¬ array-qemptyp (*queue*)))
→ ((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*)
∧ finite-number-queue-listp (cdr (*list*), *capacity*, *lub*)
else `'*1*true` **endif**

THEOREM: finite-number-queuep-getnth
(finite-number-queue-listp (*l*, *capacity*, *lub*) ∧ (*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*)
∧ 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*))
∧ ((length (car (*l*)) = *n*)
∧ (finite-number-queue-listp (car (*l*), *capacity*, *lub*)
∧ 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*) $\wedge$ ($i <$ 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*)
$\wedge$ (($i <$ 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, 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*)
$\wedge$ (($i <$ length (l)) $\wedge$ finite-number-queuep (*queue*, *capacity*, *lub*)))
$\rightarrow$ finite-number-queue-tablep (n, 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* ≉ 0) ∧ ((*a* ≉ 0) ∧ divides (*n*, *a*))) → (*a* ≮ *n*)

THEOREM: divides-not-lessp-instance
((*n* ≉ 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* ≉ 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)
                         (ARRAY-QEMPTYP (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
good-os (*os*) → (getnth ('3, tm-memory (*os*)) = '2654)

EVENT: Disable os-clock-new-pc.

THEOREM: os-error-new-pc
good-os (*os*) → (getnth ('4, tm-memory (*os*)) = '2667)

EVENT: Disable os-error-new-pc.

THEOREM: os-svc-new-pc
good-os (*os*) → (getnth ('5, tm-memory (*os*)) = '2690)

EVENT: Disable os-svc-new-pc.

THEOREM: os-input-new-pc
good-os (*os*) → (getnth ('6, tm-memory (*os*)) = '3042)

EVENT: Disable os-input-new-pc.

THEOREM: os-output-new-pc
good-os (*os*) → (getnth ('7, tm-memory (*os*)) = '3135)

EVENT: Disable os-output-new-pc.

THEOREM: os-time-slice
good-os (*os*) → (getnth ('11, tm-memory (*os*)) = '1000)

EVENT: Disable os-time-slice.

THEOREM: os-not-in-supervisor-mode
good-os (*os*) → (¬ tm-in-supervisor-mode (*os*))

EVENT: Disable os-not-in-supervisor-mode.

THEOREM: tm-base-os

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-base (*os*)
= base (getnth (os-current-taskid (*os*),
table ('2, os-segment-table (*os*)))))

EVENT: Disable tm-base-os.

THEOREM: tm-limit-os
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-limit (*os*)
= limit (getnth (os-current-taskid (*os*),
table ('2, os-segment-table (*os*)))))

EVENT: Disable tm-limit-os.

THEOREM: good-cpu-list-os-task-table
good-os (*os*) → good-cpu-list (table ('9, os-task-table (*os*)))

EVENT: Disable good-cpu-list-os-task-table.

THEOREM: good-status-list-os-status-table
good-os (*os*) → good-status-list (table ('2, os-status-table (*os*)))

EVENT: Disable good-status-list-os-status-table.

THEOREM: finite-number-queuep-os-readyq
good-os (*os*) → finite-number-queuep (os-readyq (*os*), '16, '16)

EVENT: Disable finite-number-queuep-os-readyq.

THEOREM: mutually-disjoint-os-segment-table
good-os (*os*) → mutually-disjoint (table ('2, os-segment-table (*os*)))

EVENT: Disable mutually-disjoint-os-segment-table.

THEOREM: disjoint-everywhere-os-segment-table
good-os (*os*)
→ disjoint-everywhere ('0, '3311, table ('2, 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: os-taskid (*os*) = array-qfirst (os-readyq (*os*))

THEOREM: finite-numberp-os-taskid
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ finite-numberp (os-taskid (*os*), '16)

EVENT: Disable finite-numberp-os-taskid.

EVENT: Let us define the theory *good-os-properties* 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-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, os-code-os, tm-slimit-os, good-tm-os.

THEOREM: length-finite-number-queue-corollary
finite-number-queuep (*queue*, '16, '16) → (length (*queue*) = '20)

EVENT: Disable length-finite-number-queue-corollary.

THEOREM: plistp-finite-number-queue-corollary
finite-number-queuep (*queue*, '16, '16) → plistp (*queue*)

EVENT: Disable plistp-finite-number-queue-corollary.

THEOREM: array-queuep-finite-number-queue-corollary
finite-number-queuep (*queue*, '16, '16) → array-queuep (*queue*)

EVENT: Disable array-queuep-finite-number-queue-corollary.

THEOREM: finite-number-queue-qhead-size-corollary
finite-number-queuep (*queue*, '16, '16)
→ ((getnth ('0, *queue*) ∈ **N**) ∧ (getnth ('0, *queue*) < '16))

EVENT: Disable finite-number-queue-qhead-size-corollary.

THEOREM: finite-number-queue-qtail-size-corollary
finite-number-queuep (*queue*, '16, '16)
→ ((getnth ('1, *queue*) ∈ **N**) ∧ (getnth ('1, *queue*) < '16))

EVENT: Disable finite-number-queue-qtail-size-corollary.

THEOREM: finite-number-queue-qcurrlength-size-corollary
finite-number-queuep (*queue*, '16, '16)
→ ((getnth ('2, *queue*) ∈ **N**) ∧ (getnth ('2, *queue*) < '17))

EVENT: Disable finite-number-queue-qcurrlength-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-qemptyp (*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-qemptyp (*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-qemptyp (*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-qcurrlength-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) → (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) → 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) → 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)
→ ((getnth ('0, *queue*) ∈ **N**) ∧ (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)
→ ((getnth ('1, *queue*) ∈ **N**) ∧ (getnth ('1, *queue*) < '4))

EVENT: Disable finite-number-queue-qtail-size-corollary-for-mbuffers.

THEOREM: finite-number-queue-qcurrlength-size-corollary-for-mbuffers
finite-number-queuep (*queue*, '4, '65536)
→ ((getnth ('2, *queue*) ∈ **N**) ∧ (getnth ('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrlength-size-corollary-for-mbuffers.

THEOREM: finite-number-queue-qmaxlength-value-corollary-for-mbuffers
finite-number-queuep (*queue*, '4, '65536) → (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
(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-mbuffers.

THEOREM: finite-number-queuep-array-deq-finite-number-queue-corollary-for-mbuffers
(finite-number-queuep (*queue*, '4, '65536) ∧ (¬ array-qemptyp (*queue*)))
→ finite-number-queuep (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
(finite-number-queuep (*queue*, '4, '65536) ∧ (¬ array-qemptyp (*queue*)))
→ ((array-qfirst (*queue*) ∈ **N**) ∧ (array-qfirst (*queue*) < '65536))

EVENT: Disable array-qfirst-finite-number-queue-corollary-for-mbuffers.

THEOREM: remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers
(finite-number-queuep (*queue*, '4, '65536) ∧ (¬ array-qemptyp (*queue*)))
→ ((array-qfirst (*queue*) **mod** '65536) = array-qfirst (*queue*))

EVENT: Disable remainder-array-qfirst-finite-number-queue-corollary-for-mbuffers.

THEOREM: finite-number-queue-listp-getnth-corollary
(finite-number-queue-tablep ('16, *l*, '4, '65536) ∧ (*i* < length (*l*)))
→ (plistp (getnth (*i*, *l*))
 ∧ ((length (getnth (*i*, *l*)) = '16)
 ∧ finite-number-queue-listp (getnth (*i*, *l*), '4, '65536)))

EVENT: Disable finite-number-queue-listp-getnth-corollary.

THEOREM: finite-number-queuep-getnth2-corollary
(finite-number-queue-tablep ('16, *l*, '4, '65536)
 ∧ ((*i* < length (*l*)) ∧ (*j* < '16)))
→ finite-number-queuep (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-qemptyp (*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:
mapup-os-ibuffers (*os*) = mapup-queue-list (table ('8, os-ibuffers (*os*)))

DEFINITION:
mapup-os-obuffers (*os*) = mapup-queue-list (table ('8, os-obuffers (*os*)))

DEFINITION:
mapup-os-mbuffers (*os*)
= table ('16, mapup-queue-list (table ('8, os-mbuffers (*os*))))

DEFINITION:
mapup-os (*os*)
= ak (mapup-os-tasks (*os*),
mapup-os-ibuffers (*os*),
mapup-os-obuffers (*os*),
mapup-os-mbuffers (*os*),
mapup-queue (os-readyq (*os*)),
table ('2, os-status-table (*os*)),
tm-rwstate (*os*),
tm-clock (*os*),
tm-iports (*os*),
tm-oports (*os*))

THEOREM: length-mapup-queue-list
length (mapup-queue-list (*l*)) = length (*l*)

EVENT: Disable length-mapup-queue-list.

THEOREM: getnth-mapup-queue-list
(i < length (*l*))
→ (getnth (i, mapup-queue-list (*l*)) = mapup-queue (getnth (i, *l*)))

EVENT: Disable getnth-mapup-queue-list.

THEOREM: mapup-queue-list-putnth
mapup-queue-list (putnth (v, i, *l*))
= putnth (mapup-queue (v), i, mapup-queue-list (*l*))

EVENT: Disable mapup-queue-list-putnth.

THEOREM: mapup-queue-getnth
(finite-number-queue-listp (*l*, *capacity*, *lub*) ∧ (id < length (*l*)))
→ (mapup-queue (getnth (id, *l*)) = getnth (id, mapup-queue-list (*l*)))

EVENT: Disable mapup-queue-getnth.

THEOREM: length-mapup-tasks
length (mapup-tasks (*taskid*, *os*)) = ('16 − *taskid*)

EVENT: Disable length-mapup-tasks.

THEOREM: length-mapup-os-tasks
length (mapup-os-tasks (*os*)) = '16

EVENT: Disable length-mapup-os-tasks.

THEOREM: length-mapup-cpu
(good-os (*os*) ∧ (*taskid* < '16))
→ (length (mapup-cpu (*taskid*, *os*)) = '9)

EVENT: Disable length-mapup-cpu.

THEOREM: finite-number-listp-mapup-cpu
(good-os (*os*) ∧ (*taskid* < '16))
→ finite-number-listp (mapup-cpu (*taskid*, *os*), '65536)

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

THEOREM: numberp-mapup-base
(good-os (*os*) ∧ (*taskid* < '16)) → (mapup-base (*taskid*, *os*) ∈ **N**)

EVENT: Disable numberp-mapup-base.

THEOREM: numberp-mapup-limit
(good-os (*os*) ∧ (*taskid* < '16)) → (mapup-limit (*taskid*, *os*) ∈ **N**)

EVENT: Disable numberp-mapup-limit.

THEOREM: plistp-mapup-regs
plistp (mapup-regs (*taskid*, *os*))

EVENT: Disable plistp-mapup-regs.

THEOREM: length-mapup-regs
length (mapup-regs (*taskid*, *os*)) = '8

EVENT: Disable length-mapup-regs.

THEOREM: finite-number-listp-mapup-regs
(good-os (*os*) ∧ (*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*) ∧ (*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*) ∧ (*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) ∧ (*i* < length (*l*)))
→ ('65536 ≮ (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*) ∧ (*taskid* < '16))
→ ('65536 ≮ (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*) ∧ (*taskid* < '16))
→ (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*) ∧ finite-numberp (*taskid*, '16))
→ 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*) ∧ finite-numberp (*taskid*, '16))
→ 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*) → 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)
→ 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-listp (*l*, *capacity*, '65536)
→ 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*)
 ∧ (((length (*l*) **mod** *n*) = '0) ∧ (*n* ≠ 0)))
→ 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*) → 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*)
→ finite-number-listp (mapup-queue (*queue*), *lub*)

THEOREM: finite-number-listp-mapup-finite-number-queue-corollary
good-os (*os*) → 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*) → good-ak (mapup-os (*os*))

EVENT: Disable good-ak-mapup-os.

THEOREM: not-qemptyp-mapup-queue-os-readyq
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (¬ qemptyp (mapup-queue (os-readyq (*os*))))

EVENT: Disable not-qemptyp-mapup-queue-os-readyq.

THEOREM: lessp-qfirst-mapup-queue-os-readyq
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*))

```
      ∧   (tm-good-address (addr, tm)
            ∧   ((real-addr-source (addr) ≯ 0)
                  ∧   (real-addr-source (addr) ≠ '1)))))
→     ((tm-base (tm) + tm-limit (tm)) ≠ '0)
```

THEOREM: equality-of-getseg-backchain-corollary

```
(good-tm (os)
 ∧   ((¬ tm-in-supervisor-mode (os))
      ∧   (nested-intervals ('0, '3311, i, j)
           ∧   (disjoint-everywhere ('0,
                                     '3311,
                                     table ('2, os-segment-table (os)))
                ∧   ((array-qfirst (os-readyq (os))
                      <   length (table ('2,
                                         os-segment-table (os))))
                     ∧   ((tm-base (os)
                           =   base (getnth (array-qfirst (os-readyq (os)),
                                             table ('2,
                                                    os-segment-table (os)))))
                          ∧   ((tm-limit (os)
                                =   limit (getnth (array-qfirst (os-readyq (os)),
                                                   table ('2,
                                                          os-segment-table (os)))))
                               ∧   (tm-good-address (addr,
                                                     os)
                                    ∧   ((real-addr-source (addr) ≯ 0)
                                         ∧   (real-addr-source (addr)
                                              ≠   '1)))))))))
→     (getseg (i,
               j,
               putnth (value,
                       tm-base (os) + real-addr-num (addr),
                       tm-memory (os)))
      =   getseg (i, j, tm-memory (os)))
```

EVENT: Disable equality-of-getseg-backchain-corollary.

THEOREM: equality-of-getnth-backchain-corollary

```
(good-tm (os)
 ∧   ((¬ tm-in-supervisor-mode (os))
      ∧   (if i < '0 then '*1*false
           else '*1*true endif
           ∧   (((i − '0) < '3311)
                ∧   (disjoint-everywhere ('0,
```

```
                                        '3311,
                                        table ('2,
                                               os-segment-table (os)))
                    ∧   ((array-qfirst (os-readyq (os))
                          <   length (table ('2,
                                             os-segment-table (os))))
                        ∧   ((tm-base (os)
                              =   base (getnth (array-qfirst (os-readyq (os)),
                                                table ('2,
                                                       os-segment-table (os)))))
                            ∧   ((tm-limit (os)
                                  =   limit (getnth (array-qfirst (os-readyq (os)),
                                                     table ('2,
                                                            os-segment-table (os)))))
                                ∧   (tm-good-address (addr,
                                                      os)
                                    ∧   ((real-addr-source (addr) ≉ 0)
                                        ∧   (real-addr-source (addr)
                                            ≠   '1)))))))))))
→   (getnth (i,
             putnth (value,
                     tm-base (os) + real-addr-num (addr),
                     tm-memory (os)))
    =   getnth (i, tm-memory (os)))
```

EVENT: Disable equality-of-getnth-backchain-corollary.

```
THEOREM: good-os-tm-store-in-memory
(good-os (os)
 ∧   ((¬ tm-waiting (os))
     ∧   ((real-addr-source (addr) ≉ 0)
         ∧   ((real-addr-source (addr) ≠ '1)
             ∧   (tm-good-address (addr, os)
                 ∧   finite-numberp (value, '65536))))))
→   good-os (tm-store-in-memory (value, real-addr-num (addr), os))
```

EVENT: Disable good-os-tm-store-in-memory.

```
THEOREM: good-os-tm-store
(good-os (os)
 ∧   ((¬ tm-waiting (os))
     ∧   (tm-good-address (addr, os) ∧ finite-numberp (value, '65536))))
→   good-os (tm-store (value, addr, os))
```

EVENT: Disable good-os-tm-store.

THEOREM: good-os-tm-execute-add
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr1*, *os*)))
→ good-os (tm-execute-add (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-add.

THEOREM: good-os-tm-execute-branch
(good-os (*os*) ∧ tm-good-address (*addr*, *os*))
→ 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*) ∧ tm-good-address (*addr*, *os*))
→ 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*) ∧ tm-good-address (*addr*, *os*))
→ 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*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr*, *os*)))
→ good-os (tm-execute-call (*addr*, *os*))

EVENT: Disable good-os-tm-execute-call.

THEOREM: good-os-tm-execute-compare
(good-os (*os*) ∧ tm-good-address (*addr1*, *os*))
→ good-os (tm-execute-compare (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-compare.

THEOREM: good-os-tm-execute-decr
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr*, *os*)))
→ good-os (tm-execute-decr (*addr*, *os*))

EVENT: Disable good-os-tm-execute-decr.

THEOREM: good-os-tm-execute-decr-mod
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr1*, *os*)))
→ good-os (tm-execute-decr-mod (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-decr-mod.

THEOREM: good-os-tm-execute-incr
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr*, *os*)))
→ good-os (tm-execute-incr (*addr*, *os*))

EVENT: Disable good-os-tm-execute-incr.

THEOREM: good-os-tm-execute-incr-mod
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr1*, *os*)))
→ good-os (tm-execute-incr-mod (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-incr-mod.

THEOREM: good-os-tm-execute-load-base
good-os (*os*) → good-os (tm-execute-load-base (*addr*, *os*))

EVENT: Disable good-os-tm-execute-load-base.

THEOREM: good-os-tm-execute-load-limit
good-os (*os*) → good-os (tm-execute-load-limit (*addr*, *os*))

EVENT: Disable good-os-tm-execute-load-limit.

THEOREM: good-os-tm-execute-load-psw
good-os (*os*) → good-os (tm-execute-load-psw (*addr*, *os*))

EVENT: Disable good-os-tm-execute-load-psw.

THEOREM: good-os-tm-execute-mod
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr1*, *os*)))
→ good-os (tm-execute-mod (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-mod.

THEOREM: good-os-tm-execute-move
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-good-address (*addr1*, *os*) ∧ tm-good-address (*addr2*, *os*))))
→ good-os (tm-execute-move (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-move.

THEOREM: good-os-tm-execute-multiply
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ tm-good-address (*addr1*, *os*)))
→ good-os (tm-execute-multiply (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-multiply.

THEOREM: good-os-tm-execute-return
good-os (*os*) → good-os (tm-execute-return (*os*))

EVENT: Disable good-os-tm-execute-return.

THEOREM: good-os-tm-execute-post-output-interrupt
good-os (*os*) → good-os (tm-execute-post-output-interrupt (*addr*, *os*))

EVENT: Disable good-os-tm-execute-post-output-interrupt.

THEOREM: good-os-tm-execute-run
(good-os (*os*) ∧ (¬ tm-waiting (*os*))) → good-os (tm-execute-run (*os*))

EVENT: Disable good-os-tm-execute-run.

THEOREM: good-os-tm-execute-set-clock
good-os (*os*) → good-os (tm-execute-set-clock (*addr*, *os*))

EVENT: Disable good-os-tm-execute-set-clock.

THEOREM: good-os-tm-execute-start-output
good-os (*os*) → good-os (tm-execute-start-output (*addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-start-output.

THEOREM: good-os-tm-execute-svc
(good-os (*os*) ∧ tm-good-address (*addr*, *os*))
→ good-os (tm-execute-svc (*addr*, *os*))

EVENT: Disable good-os-tm-execute-svc.

THEOREM: good-os-tm-execute-svc-return
good-os (*os*) → good-os (tm-execute-svc-return (*addr*, *os*))

EVENT: Disable good-os-tm-execute-svc-return.

THEOREM: good-os-tm-execute-test-iport
good-os (*os*) → good-os (tm-execute-test-iport (*addr*, *os*))

EVENT: Disable good-os-tm-execute-test-iport.

THEOREM: good-os-tm-execute-test-oport
good-os (*os*) → good-os (tm-execute-test-oport (*addr*, *os*))

EVENT: Disable good-os-tm-execute-test-oport.

THEOREM: good-os-tm-execute-wait
(good-os (*os*) ∧ (¬ tm-waiting (*os*))) → good-os (tm-execute-wait (*os*))

EVENT: Disable good-os-tm-execute-wait.

THEOREM: good-os-tm-execute-nullary
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-os (tm-execute-nullary (*opcode*, *os*))

EVENT: Disable good-os-tm-execute-nullary.

THEOREM: good-os-tm-execute-unary
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-os (tm-execute-unary (*opcode*, *addr*, *os*))

EVENT: Disable good-os-tm-execute-unary.

THEOREM: good-os-tm-execute-binary
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-os (tm-execute-binary (*opcode*, *addr1*, *addr2*, *os*))

EVENT: Disable good-os-tm-execute-binary.

THEOREM: good-os-tm-execute-instruction
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-os (tm-execute-instruction (*instruction*, *os*))

EVENT: Disable good-os-tm-execute-instruction.

THEOREM: good-os-tm-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-os (tm-execute (*instruction*, *os*))

EVENT: Disable good-os-tm-execute.

THEOREM: good-os-tm-decrement-clock
(good-os (*os*) ∧ (¬ tm-waiting (*os*))) → good-os (tm-decrement-clock (*os*))

EVENT: Disable good-os-tm-decrement-clock.

THEOREM: good-os-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*))) → good-os (tm-fetch-execute (*os*))

EVENT: Disable good-os-tm-fetch-execute.

THEOREM: good-os-tm-set-iports
(good-os (*os*)
∧ (plistp (*iports*)
∧ ((length (*iports*) = '16) ∧ good-tm-iport-array (*iports*))))
→ good-os (tm-set-iports (*iports*, *os*))

EVENT: Disable good-os-tm-set-iports.

THEOREM: good-os-tm-set-oports
(good-os (*os*)
∧ (plistp (*oports*)
∧ ((length (*oports*) = '16) ∧ good-tm-oport-array (*oports*))))
→ good-os (tm-set-oports (*oports*, *os*))

EVENT: Disable good-os-tm-set-oports.

THEOREM: good-os-tm-post-interrupt
good-os (*os*) → good-os (tm-post-interrupt (*interrupt*, *os*))

EVENT: Disable good-os-tm-post-interrupt.

DEFINITION:
disjoint-segments-induction (*i*, *j*, *l*)
= **if** *i* ≃ 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*)))),

limit (getnth (*taskid*,
table ('2, os-segment-table (*os*)))))
= mapup-address-space (tm-memory (*os*),
regs,
cc,
error,
svcflag,
svcid,
base (getnth (*taskid*,
table ('2,
os-segment-table (*os*)))),
limit (getnth (*taskid*,
table ('2,
os-segment-table (*os*))))))

EVENT: Disable tm-fetch-execute-mapup-address-space-separation.

THEOREM: getnth-tm-reglength-tm-cpu
getnth ('8, tm-cpu (*tm*))
= tm-pack-psw (tm-cc (*tm*), tm-error (*tm*), tm-svcflag (*tm*), tm-svcid (*tm*))

EVENT: Disable getnth-tm-reglength-tm-cpu.

THEOREM: os-ibuffers-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ibuffers (tm-fetch-execute (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-tm-fetch-execute.

THEOREM: os-obuffers-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-obuffers (tm-fetch-execute (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-tm-fetch-execute.

THEOREM: os-mbuffers-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-mbuffers (tm-fetch-execute (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-tm-fetch-execute.

THEOREM: os-readyq-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-readyq (tm-fetch-execute (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-tm-fetch-execute.

THEOREM: os-status-table-tm-fetch-execute
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) → (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
finite-numberp $(cc,$ '4$)$
$\rightarrow$ (tm-unpack-cc (tm-pack-psw $(cc,\ error,\ svcflag,\ svcid)) = cc$)

EVENT: Disable tm-unpack-cc-tm-pack-psw.

THEOREM: remainder-plus-plus-times
$((a + (b + (j * i))))$ **mod** $j) = ((a + b)$ **mod** $j)$

EVENT: Disable remainder-plus-plus-times.

THEOREM: tm-unpack-error-tm-pack-psw
(finite-numberp $(cc,$ '4$) \wedge$ finite-numberp $(error,$ '64$)$)
$\rightarrow$ (tm-unpack-error (tm-pack-psw $(cc,\ error,\ svcflag,\ svcid)) = error$)

EVENT: Disable tm-unpack-error-tm-pack-psw.

THEOREM: remainder-plus-plus-plus-times
$((a + (b + (c + (i * j))))$ **mod** $j) = ((a + (b + c))$ **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
(finite-numberp $(cc,$ '4$)$
$\wedge$ (finite-numberp $(error,$ '64$) \wedge$ finite-numberp $(svcflag,$ '2$)$))
$\rightarrow$ (tm-unpack-svcflag (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)
∧ (finite-numberp (*error*, '64)
∧ (finite-numberp (*svcflag*, '2) ∧ 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*) ∧ (¬ 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*)
∧ ((¬ tm-waiting (*os*))
∧ ((*taskid* ∈ **N**)
∧ ((*taskid* < '16)
∧ (*taskid* ≠ 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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
∧ ((¬ tm-waiting (*os*))
∧ ((*taskid* ∈ **N**) ∧ (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
$((i < \texttt{'16})$
$\wedge \quad (\textbf{if } i < j \textbf{ then } \texttt{'*1*false}$
$\textbf{else } \texttt{'*1*true } \textbf{endif}$
$\wedge \quad ((j \in \mathbf{N}) \wedge (i \in \mathbf{N}))))$
$\rightarrow \quad (\text{getnth}\,(i - j, \text{mapup-tasks}\,(j,\ os)) = \text{mapup-task}\,(i,\ os))$

THEOREM: getnth-mapup-tasks-corollary2
$((i < \texttt{'16}) \wedge (i \in \mathbf{N}))$
$\rightarrow \quad (\text{getnth}\,(\texttt{'0}, \text{mapup-tasks}\,(i,\ os)) = \text{mapup-task}\,(i,\ os))$

EVENT: Disable getnth-mapup-tasks-corollary2.

THEOREM: mapup-tasks-tm-fetch-execute-case2
$(\text{good-os}\,(os)$
$\wedge \quad ((\neg\ \text{tm-waiting}\,(os))$
$\wedge \quad ((taskid \in \mathbf{N})$
$\wedge \quad \textbf{if } \text{os-current-taskid}\,(os) < taskid \textbf{ then } \texttt{'*1*false}$
$\textbf{else } \texttt{'*1*true } \textbf{endif}$)))
$\rightarrow \quad (\text{mapup-tasks}\,(taskid,\ \text{tm-fetch-execute}\,(os))$
$= \quad \text{putnth}\,(\text{tm-fetch-execute}\,(\text{getnth}\,(\text{os-current-taskid}\,(os)$
$- \quad taskid,$
$\text{mapup-tasks}\,(taskid,\ os))),$
$\text{os-current-taskid}\,(os) - taskid,$
$\text{mapup-tasks}\,(taskid,\ os)))$

EVENT: Disable mapup-tasks-tm-fetch-execute-case2.

THEOREM: difference-zero-arg2
$(n - \texttt{'0}) = \text{fix}\,(n)$

EVENT: Disable difference-zero-arg2.

THEOREM: mapup-os-tasks-tm-fetch-execute
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad (\text{mapup-os-tasks}\,(\text{tm-fetch-execute}\,(os))$
$= \quad \text{putnth}\,(\text{tm-fetch-execute}\,(\text{getnth}\,(\text{qfirst}\,(\text{mapup-queue}\,(\text{os-readyq}\,(os))),$
$\text{mapup-os-tasks}\,(os))),$
$\text{qfirst}\,(\text{mapup-queue}\,(\text{os-readyq}\,(os))),$
$\text{mapup-os-tasks}\,(os)))$

EVENT: Disable mapup-os-tasks-tm-fetch-execute.

THEOREM: correctness-of-tm-fetch-execute
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-clock-interruptp (*os*))))
→ (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*))
→ (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*) ∧ array-qemptyp (*queue*))
→ ((*maxlength* = getnth ('3, *queue*))
→ (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*))
→ (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*)
∧ ((¬ array-qemptyp (*queue*)) ∧ (*maxlength* = getnth ('3, *queue*))))
→ (putnth (*item*, '4 + decr-mod (getnth ('1, *queue*), *maxlength*), *queue*)
= 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: tick (*n*) = fixlength (*n*, 'nil, 'tick)

THEOREM: tick-plus
tick (*a* + *b*) = append (tick (*a*), tick (*b*))

THEOREM: getnth-os-code
good-os (*os*)
→ (getnth (*n*, os-code (*os*))
= 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-qcurrlength-in-a-non-full-queue, array-qcurrlength-in-a-full-queue, not-array-qfullp-array-deq, not-array-qemptyp-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-qcurrlength-add1-qmaxlength, lessp-qtail-qmaxlength, lessp-qhead-qmaxlength, qmaxlength-greater-than-zero, qmaxlength-non-zero, numberp-qmaxlength, numberp-qcurrlength, numberp-qtail, numberp-qhead.

DEFINITION: OS-READYQ-QARRAY-ADDRESS = '195

THEOREM: os-readyq-address-facts
finite-number-queuep (*queue*, '16, '16)
→ ((((('195 + getnth ('0, *queue*)) < '65536) = '*1*true)
∧ (((('195 + getnth ('1, *queue*)) < '65536) = '*1*true))

EVENT: Disable os-readyq-address-facts.

THEOREM: tm-pack-psw-tm-unpack-fields1
tm-pack-psw (tm-unpack-cc (*n*),
tm-unpack-error (*n*),
tm-unpack-svcflag (*n*),
tm-unpack-svcid (*n*))
= ((*n* **mod** '256)
+ (('256 ∗ ((*n* **mod** '512) ÷ '256))
+ ('512 ∗ (*n* ÷ '512))))

EVENT: Disable tm-pack-psw-tm-unpack-fields1.

THEOREM: tm-pack-psw-tm-unpack-fields2
((*n* **mod** '256)
+ (('256 ∗ ((*n* **mod** '512) ÷ '256))
+ ('512 ∗ (*n* ÷ '512))))
= ((*n* **mod** '512) + ('512 ∗ (*n* ÷ '512)))

EVENT: Disable tm-pack-psw-tm-unpack-fields2.

THEOREM: tm-pack-psw-tm-unpack-fields
(*n* ∈ **N**)
→ (tm-pack-psw (tm-unpack-cc (*n*),
tm-unpack-error (*n*),
tm-unpack-svcflag (*n*),
tm-unpack-svcid (*n*))
= *n*)

EVENT: Disable tm-pack-psw-tm-unpack-fields.

THEOREM: list-of-getnths9
(plistp (l) $\wedge$ (length (l) = '9))
$\rightarrow$ (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),
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-putnths8

```
(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-putnths8.

THEOREM: contract-list-of-getnths8

```
((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),
```

cons (getnth ('5 + n, l),
cons (getnth ('6 + n, l),
cons (getnth ('7 + n, l), 'nil))))))))
= getseg (n, '8, l))

EVENT: Disable contract-list-of-getnths8.

THEOREM: contract-sequence-of-putnths8
(($n \in \mathbf{N}$)
∧ **if** length (l) < ('9 + n) **then** '*1*false
else '*1*true **endif**)
→ (putnth ($v0$,
n,
putnth ($v1$,
'1 + n,
putnth ($v2$,
'2 + n,
putnth ($v3$,
'3 + n,
putnth ($v4$,
'4 + n,
putnth ($v5$,
'5 + n,
putnth ($v6$,
'6 + n,
putnth ($v7$,
'7 + n,
putnth ($v8$,
'8
+ n,
l))))))))))
= putseg (cons ($v0$,
cons ($v1$,
cons ($v2$,
cons ($v3$,
cons ($v4$,
cons ($v5$,
cons ($v6$,
cons ($v7$,
cons ($v8$, 'nil))))))))),
n,
l))

EVENT: Disable contract-sequence-of-putnths8.

```
(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-putnths-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*) ∧ (¬ tm-waiting (*os*)))
→ 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)
→ ((((('159 + ('2 ∗ *id*)) < '65536) = '*1*true)
∧ ((((('160 + ('2 ∗ *id*)) < '65536) = '*1*true)
∧ ((((('2 ∗ *id*) < '32) = '*1*true)
∧ ((('1 + ('2 ∗ *id*)) < '32)
= '*1*true))))

EVENT: Disable os-segment-table-address-facts.

THEOREM: contract-os-new-base-expression
(('2 ∗ *taskid*) < '32)
→ (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)
→ (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-putnths2
((*n* ∈ **N**)
∧ **if** length (*l*) < ('2 + *n*) **then** '*1*false
else '*1*true **endif**)
→ (putnth (*a*, *n*, putnth (*b*, '1 + *n*, *l*))
= putseg (cons (*a*, cons (*b*, 'nil)), *n*, *l*))

EVENT: Disable contract-sequence-of-putnths2.

THEOREM: contract-putnths-into-status-table
$(i < \texttt{'16})$
$\rightarrow \quad (\text{putnth}\,(a,\ \texttt{'2} * i,\ \text{putnth}\,(b,\ \texttt{'1} + (\texttt{'2} * i),\ \text{os-status-table}\,(os)))$
$\quad = \quad \text{putseg}\,(\text{cons}\,(a,\ \text{cons}\,(b,\ \texttt{'nil})),\ \texttt{'2} * i,\ \text{os-status-table}\,(os)))$

EVENT: Disable contract-putnths-into-status-table.

THEOREM: os-status-table-address-facts
$(id < \texttt{'16})$
$\rightarrow \quad ((((\texttt{'2} * id) < \texttt{'65536}) = \texttt{'*1*true})$
$\quad \wedge \quad ((((\texttt{'211} + (\texttt{'2} * id)) < \texttt{'65536}) = \texttt{'*1*true})$
$\quad\quad \wedge \quad ((((\texttt{'212} + (\texttt{'2} * id)) < \texttt{'65536})$
$\quad\quad\quad = \quad \texttt{'*1*true})$
$\quad\quad\quad \wedge \quad ((((\texttt{'2} * id) < \texttt{'32}) = \texttt{'*1*true})$
$\quad\quad\quad\quad \wedge \quad ((((1 + (\texttt{'2} * id)) < \texttt{'32})$
$\quad\quad\quad\quad\quad = \quad \texttt{'*1*true})$
$\quad\quad\quad\quad\quad \wedge \quad ((((\texttt{'1} + (\texttt{'2} * id))$
$\quad\quad\quad\quad\quad\quad < \quad \texttt{'32})$
$\quad\quad\quad\quad\quad\quad = \quad \texttt{'*1*true})$
$\quad\quad\quad\quad\quad\quad \wedge \quad ((\texttt{'32}$
$\quad\quad\quad\quad\quad\quad\quad < \quad (\texttt{'2}$
$\quad\quad\quad\quad\quad\quad\quad\quad + \quad (\texttt{'2}$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad * \quad id)))$
$\quad\quad\quad\quad\quad\quad\quad = \quad \texttt{'*1*false})))))))$

EVENT: Disable os-status-table-address-facts.

THEOREM: access-os-status-table-crock
$(\text{good-tm}\,(os) \wedge (id < \texttt{'16}))$
$\rightarrow \quad ((\text{getnth}\,(\texttt{'2} * id,\ \text{os-status-table}\,(os)) \in \mathbf{N})$
$\quad \wedge \quad ((\text{getnth}\,(\texttt{'2} * id,\ \text{os-status-table}\,(os)) < \texttt{'65536})$
$\quad\quad \wedge \quad ((\text{getnth}\,(\texttt{'1} + (\texttt{'2} * id),\ \text{os-status-table}\,(os)) \in \mathbf{N})$
$\quad\quad\quad \wedge \quad (\text{getnth}\,(\texttt{'1} + (\texttt{'2} * id),$
$\quad\quad\quad\quad\quad\quad \text{os-status-table}\,(os))$
$\quad\quad\quad\quad < \quad \texttt{'65536}))))$

EVENT: Disable access-os-status-table-crock.

THEOREM: finite-numberp-getnth-os-status-table
$(\text{good-tm}\,(os) \wedge (i < \texttt{'32}))$
$\rightarrow \quad \text{finite-numberp}\,(\text{getnth}\,(i,\ \text{os-status-table}\,(os)),\ \texttt{'65536})$

EVENT: Let us define the theory *access-os-status-table* to consist of the following

events: contract-putnths-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

$$
\begin{array}{l}
((i < \texttt{'16}) \wedge (j < \texttt{'16})) \\
\rightarrow \quad (((((\texttt{'16} * j) + i) < \texttt{'256}) = \texttt{'*1*true}) \\
\quad \wedge \quad ((((\texttt{'8} * (i + (\texttt{'16} * j))) < \texttt{'2048}) = \texttt{'*1*true}) \\
\quad\quad \wedge \quad ((((\texttt{'8} * ((\texttt{'16} * j) + i)) < \texttt{'2048}) \\
\quad\quad\quad = \quad \texttt{'*1*true}) \\
\quad\quad\quad \wedge \quad (((\texttt{'2048} \\
\quad\quad\quad\quad < \quad (\texttt{'8} \\
\quad\quad\quad\quad\quad + \quad (\texttt{'8} \\
\quad\quad\quad\quad\quad\quad * \quad ((\texttt{'16} * j) + i)))) \\
\quad\quad\quad\quad = \quad \texttt{'*1*false})
\end{array}
$$

$$
\begin{array}{l}
\wedge \quad (((((\texttt{'8} * j) + (\texttt{'128} * i)) \\
\qquad < \quad \texttt{'2048}) \\
\qquad = \quad \texttt{'*1*true}) \\
\qquad \wedge \quad (((\texttt{'2048} \\
\qquad\qquad < \quad (\texttt{'8} \\
\qquad\qquad\qquad + \quad (\texttt{'8} \\
\qquad\qquad\qquad\qquad * \quad (i + (\texttt{'16} \\
\qquad\qquad\qquad\qquad\qquad\qquad * \quad j)))))) \\
\qquad\qquad = \quad \texttt{'*1*false}) \\
\qquad\qquad \wedge \quad (((i + (\texttt{'16} * j)) \\
\qquad\qquad\qquad < \quad \texttt{'256}) \\
\qquad\qquad\qquad = \quad \texttt{'*1*true})))))))
\end{array}
$$

EVENT: Disable os-mbuffers-address-facts3.

THEOREM: finite-numberp-getnth-from-current-mbuffer

$$
\begin{array}{l}
(\text{good-tm}\,(os) \wedge ((sid < \texttt{'16}) \wedge ((did < \texttt{'16}) \wedge (i < \texttt{'8})))) \\
\rightarrow \quad \text{finite-numberp}\,(\text{getnth}\,(i, \\
\qquad\qquad\qquad\qquad \text{getnth}\,((\texttt{'16} * sid) + did, \\
\qquad\qquad\qquad\qquad\qquad \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os)))), \\
\qquad\qquad\qquad \texttt{'65536})
\end{array}
$$

THEOREM: finite-numberp-getnth-from-current-mbuffer1

$$
\begin{array}{l}
(\text{good-tm}\,(os) \wedge ((sid < \texttt{'16}) \wedge ((did < \texttt{'16}) \wedge (i < \texttt{'8})))) \\
\rightarrow \quad ((\text{getnth}\,(i, \text{getnth}\,((\texttt{'16} * sid) + did, \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os)))) \in \mathbf{N}) \\
\qquad \wedge \quad (\text{getnth}\,(i, \\
\qquad\qquad\qquad \text{getnth}\,((\texttt{'16} * sid) + did, \\
\qquad\qquad\qquad\qquad \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os)))) \\
\qquad\qquad < \quad \texttt{'65536}))
\end{array}
$$

EVENT: Disable finite-numberp-getnth-from-current-mbuffer1.

THEOREM: access-current-os-mbuffer-fields

$$
\begin{array}{l}
(\text{good-os}\,(os) \wedge ((sid < \texttt{'16}) \wedge (did < \texttt{'16}))) \\
\rightarrow \quad ((\text{getnth}\,(\texttt{'1}, \\
\qquad\qquad\qquad \text{getnth}\,((\texttt{'16} * sid) + did, \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os)))) \\
\qquad\qquad < \quad \texttt{'65535}) \\
\qquad\qquad \wedge \quad ((\text{getnth}\,(\texttt{'2}, \\
\qquad\qquad\qquad\qquad \text{getnth}\,((\texttt{'16} * sid) + did, \\
\qquad\qquad\qquad\qquad\qquad \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os)))) \\
\qquad\qquad\qquad\qquad < \quad \texttt{'65535}) \\
\qquad\qquad\qquad\qquad \wedge \quad (\text{getnth}\,(\texttt{'3}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{getnth}\,((\texttt{'16} * sid) + did, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{table}\,(\texttt{'8}, \text{os-mbuffers}\,(os))))
\end{array}
$$

$<$ '65535)))

EVENT: Disable access-current-os-mbuffer-fields.

THEOREM: lessp-plus-fact-for-mbuffers1
(good-os (os)
 $\wedge$ (($\neg$ tm-waiting (os)) $\wedge$ ((sid $<$ '16) $\wedge$ (did $<$ '16))))
$\rightarrow$ ((('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)
 $\wedge$ (($\neg$ tm-waiting (os)) $\wedge$ ((sid $<$ '16) $\wedge$ (did $<$ '16))))
$\rightarrow$ ((('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**
 $\wedge$ finite-number-queuep (getnth ((('16 $*$ array-qfirst (os-readyq (os)))
 + (getnth ('2, tm-regs (os)) **mod** '16),
 table ('8, os-mbuffers (os))),
 '4,
 '65536))
$\rightarrow$ (((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
('4
 + (('8 * i)
     + (('128 * j)
         + getnth (('8 * i) + ('128 * j), os-mbuffers (os)))))
= (('8 * i)
    + (('128 * j)
        + ('4
            + getnth (('8 * i) + ('128 * j),
                       os-mbuffers (os)))))

EVENT: Disable associativity-of-plus-crock.

THEOREM: contract-getnth-from-current-mbuffer-for-receive
((('4 + getnth ('0, os-current-mbuffer (sid, did, os))) < '8)
 ∧ ((sid < '16) ∧ (did < '16)))
→ (getnth ('4
            + ((('8 * did) + ('128 * sid))
                + getnth ('0,
                           getnth (did + ('16 * sid),
                                  table ('8, os-mbuffers (os)))))),
            os-mbuffers (os))
    = getnth ('4 + getnth ('0, os-current-mbuffer (sid, did, os)),
               os-current-mbuffer (sid, did, os)))

EVENT: Disable contract-getnth-from-current-mbuffer-for-receive.

THEOREM: access-current-mbuffer-qhead
((sid < '16) ∧ (did < '16))
→ (getnth (('8 * did) + ('128 * sid), os-mbuffers (os))
    = getnth ('0, os-current-mbuffer (sid, did, os)))

EVENT: Disable access-current-mbuffer-qhead.

THEOREM: contract-putnth-into-current-mbuffer-for-receive1
((i ∈ **N**) ∧ ((i < '4) ∧ ((sid < '16) ∧ (did < '16))))
→ (putnth ($value$,
            i + (('8 * did) + ('128 * sid)),
            os-mbuffers (os))
    = putseg (putnth ($value$, i, os-current-mbuffer (sid, did, os)),
               ('8 * did) + ('128 * sid),
               os-mbuffers (os)))

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*) ∧ ('4 = getnth ('3, *queue*)))
→ (array-qfullp (*queue*) = qfullp (mapup-queue (*queue*), '4))

EVENT: Disable correctness-of-array-qfullp-mbuffers-instance.

THEOREM: mapup-queue-getnth2-os-mbuffers
$((i < \texttt{'16}) \wedge (j < \texttt{'16}))$
$\rightarrow$ (mapup-queue (getnth2 (i, j, table ('16, table ('8, os-mbuffers (os)))))
$=$ getnth2 (i, j, mapup-os-mbuffers (os)))

EVENT: Disable mapup-queue-getnth2-os-mbuffers.

THEOREM: os-current-mbuffer-rewrite-to-getnth2
$((sid < \texttt{'16}) \wedge (did < \texttt{'16}))$
$\rightarrow$ (getnth (('16 $*$ sid) $+$ did, table ('8, os-mbuffers (os)))
$=$ getnth2 (sid, did, table ('16, table ('8, os-mbuffers (os)))))

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
finite-number-queuep ($queue$, '4, '65536) $\rightarrow$ (length ($queue$) $=$ '8)

EVENT: Disable length-finite-number-queue-corollary-for-ibuffers.

THEOREM: plistp-finite-number-queue-corollary-for-ibuffers
finite-number-queuep ($queue$, '4, '65536) $\rightarrow$ plistp ($queue$)

EVENT: Disable plistp-finite-number-queue-corollary-for-ibuffers.

THEOREM: array-queuep-finite-number-queue-corollary-for-ibuffers
finite-number-queuep ($queue$, '4, '65536) $\rightarrow$ array-queuep ($queue$)

EVENT: Disable array-queuep-finite-number-queue-corollary-for-ibuffers.

THEOREM: finite-number-queue-qhead-size-corollary-for-ibuffers
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-ibuffers.

THEOREM: finite-number-queue-qtail-size-corollary-for-ibuffers
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-ibuffers.

THEOREM: finite-number-queue-qcurrlength-size-corollary-for-ibuffers
finite-number-queuep (*queue*, '4, '65536)
→ ((getnth ('2, *queue*) ∈ **N**) ∧ (getnth ('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrlength-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-qemptyp (*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-qemptyp (*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-qemptyp (*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-QCURRLENGTH-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*) ∧ (*id* < '16))
→ 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)
∧ ((¬ array-qemptyp (*queue*)) ∧ finite-numberp (*item*, '65536)))
→ 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*) ∧ ((¬ tm-waiting (*os*)) ∧ (*id* < '16)))
→ ((((('4
+ (('8 ∗ *id*)
+ getnth ('0, getnth (*id*, table ('8, os-ibuffers (*os*))))))
< '128)
= '*1*true)
∧ (((('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**
∧ finite-number-queuep (getnth (*id*, table ('8, os-ibuffers (*os*))),
'4,
'65536))
→ (((*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*) ∧ (*id* < '16))
→ (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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ ((((('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*)))),

```
                                  '4)))
          <   '65536)
         =   '*1*true)
        ∧   ((((('247
                 +   (('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)))),
                                    '4)))
                  <   '65536)
                 =   '*1*true)
                ∧   (('('4
                         +   (('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)))),
                                            '4)))
                           <   '128)
                          =   '*1*true)))
```

EVENT: Disable qreplace-address-fact.

THEOREM: more-ibuffer-address-facts

```
(good-os (os) ∧ tm-some-input-interruptp (tm-iports (os)))
→   ((((('243
           +   (('8 * tm-interrupting-input-port (tm-iports (os)))
                 +   getnth ('1,
                             getnth (tm-interrupting-input-port (tm-iports (os)),
                                     table ('8, os-ibuffers (os))))))
          <   '65536)
         =   '*1*true)
        ∧   ((((('247
                 +   (('8
                       *   tm-interrupting-input-port (tm-iports (os)))
                      +   getnth ('1,
                                  getnth (tm-interrupting-input-port (tm-iports (os)),
                                          table ('8, os-ibuffers (os))))))
                  <   '65536)
                 =   '*1*true)
                ∧   (('('4
```

```
      +   (('8
             *   tm-interrupting-input-port (tm-iports (os)))
            +   getnth ('1,
                        getnth (tm-interrupting-input-port (tm-iports (os)),
                                table ('8,
                                       os-ibuffers (os))))))
      <   '128)
     =   '*1*true)))
```

EVENT: Disable more-ibuffer-address-facts.

THEOREM: access-current-ibuffer
$((id < \texttt{'16}) \wedge (i < \texttt{'8}))$
$\rightarrow$ (getnth $(i + (\texttt{'8} * id)$, os-ibuffers (os))
= getnth $(i,$ os-current-ibuffer $(id,\ os)))$

EVENT: Disable access-current-ibuffer.

THEOREM: contract-getnth-from-current-ibuffer
$(((\texttt{'4} +$ getnth $(\texttt{'0},$ os-current-ibuffer $(id,\ os))) < \texttt{'8}) \wedge (id < \texttt{'16}))$
$\rightarrow$ (getnth ('4
+ (('8 $*$ id)
+ getnth ('0,
getnth (id, table ('8, os-ibuffers (os))))),
os-ibuffers (os))
= getnth ('4 + getnth ('0, os-current-ibuffer $(id,\ os)$),
os-current-ibuffer $(id,\ os)))$

EVENT: Disable contract-getnth-from-current-ibuffer.

THEOREM: access-current-ibuffer-qhead
$(id < \texttt{'16})$
$\rightarrow$ (getnth $(\texttt{'8} * id,$ os-ibuffers (os))
= getnth $(\texttt{'0},$ os-current-ibuffer $(id,\ os)))$

EVENT: Disable access-current-ibuffer-qhead.

THEOREM: contract-putnth-into-current-ibuffer
$((i \in \mathbf{N}) \wedge ((i < \texttt{'4}) \wedge (id < \texttt{'16})))$
$\rightarrow$ (putnth $(value,\ i + (\texttt{'8} * id)$, os-ibuffers (os))
= putseg (putnth $(value,\ i,$ os-current-ibuffer $(id,\ os)),$
'8 $*$ id,
os-ibuffers $(os)))$

EVENT: Disable contract-putnth-into-current-ibuffer.

THEOREM: contract-putnth-into-current-ibuffer1
```
(((i + decr-mod (getnth ('1, os-current-ibuffer (id, os)), '4)) < '8)
 ∧   (id < '16))
→   (putnth (value,
              i + (('8 * id)
                   +   decr-mod (getnth ('1,
                                         getnth (id,
                                                 table ('8,
                                                        os-ibuffers (os)))),
                                 '4)),
              os-ibuffers (os))
     =   putseg (putnth (value,
                         i + decr-mod (getnth ('1,
                                               os-current-ibuffer (id, os)),
                                       '4),
                         os-current-ibuffer (id, os)),
                 '8 * id,
                 os-ibuffers (os)))
```

EVENT: Disable contract-putnth-into-current-ibuffer1.

THEOREM: contract-putnth-into-current-ibuffer2
```
(((i + getnth ('1, os-current-ibuffer (id, os))) < '8) ∧ (id < '16))
→   (putnth (value,
              i + (('8 * id)
                   +   getnth ('1,
                               getnth (id, table ('8, os-ibuffers (os))))),
              os-ibuffers (os))
     =   putseg (putnth (value,
                         i + getnth ('1, os-current-ibuffer (id, os)),
                         os-current-ibuffer (id, os)),
                 '8 * id,
                 os-ibuffers (os)))
```

EVENT: Disable contract-putnth-into-current-ibuffer2.

DEFINITION:
```
os-ibuffers-with-dequeued-character (os)
=   putseg (array-deq (os-current-ibuffer (os-current-taskid (os), os)),
            '8 * os-current-taskid (os),
            os-ibuffers (os))
```

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*) ∧ (*id* < '16))
→ (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) → (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) → 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) → array-queuep (*queue*)

EVENT: Disable array-queuep-finite-number-queue-corollary-for-obuffers.

THEOREM: finite-number-queue-qhead-size-corollary-for-obuffers
finite-number-queuep (*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-queuep (*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-qcurrlength-size-corollary-for-obuffers
finite-number-queuep (*queue*, '4, '65536)
→ ((getnth ('2, *queue*) ∈ **N**) ∧ (getnth ('2, *queue*) < '5))

EVENT: Disable finite-number-queue-qcurrlength-size-corollary-for-obuffers.

THEOREM: finite-number-queue-qmaxlength-value-corollary-for-obuffers
finite-number-queuep (*queue*, '4, '65536) → (getnth ('3, *queue*) = '4)

EVENT: Disable finite-number-queue-qmaxlength-value-corollary-for-obuffers.

THEOREM: finite-number-queuep-array-enq-finite-number-queue-corollary-for-obuffers
(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-obuffers.

THEOREM: finite-number-queuep-array-deq-finite-number-queue-corollary-for-obuffers
(finite-number-queuep (*queue*, '4, '65536) ∧ (¬ array-qemptyp (*queue*)))
→ finite-number-queuep (array-deq (*queue*), '4, '65536)

EVENT: Disable finite-number-queuep-array-deq-finite-number-queue-corollary-for-obuffers.

THEOREM: array-qfirst-finite-number-queue-corollary-for-obuffers
(finite-number-queuep (*queue*, '4, '65536) ∧ (¬ array-qemptyp (*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) ∧ (¬ array-qemptyp (*queue*)))
→ ((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-QCURRLENGTH-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*) ∧ (*id* < '16))
→ 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
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (*id* < '16)))
→ ((((('4
+ (('8 ∗ *id*)
+ getnth ('1, getnth (*id*, table ('8, os-obuffers (*os*))))))
< '128)
= '*1*true)
∧ (((('375
+ (('8 ∗ *id*)
+ getnth ('1,
getnth (*id*, table ('8, os-obuffers (*os*))))))
< '65536)
= '*1*true))

EVENT: Disable lessp-plus-facts-for-obuffers.

THEOREM: lessp-qmaxlength-field-os-current-obuffer-tm-wordlub
(good-tm (*os*) ∧ (*id* < '16))
→ (getnth ('3, getnth (*id*, table ('8, os-obuffers (*os*)))) < '65536)

EVENT: Disable lessp-qmaxlength-field-os-current-obuffer-tm-wordlub.

THEOREM: obuffer-address-facts
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ ((('375
+ (('8 ∗ tm-interrupting-output-port (tm-oports (*os*)))
+ getnth ('0,
getnth (tm-interrupting-output-port (tm-oports (*os*)),
table ('8, os-obuffers (*os*))))))
< '65536)
= '*1*true)

EVENT: Disable obuffer-address-facts.

THEOREM: access-current-obuffer
((*id* < '16) ∧ (*i* < '8))
→ (getnth (*i* + ('8 ∗ *id*), os-obuffers (*os*))
= getnth (*i*, os-current-obuffer (*id*, *os*)))

EVENT: Disable access-current-obuffer.

THEOREM: contract-getnth-from-current-obuffer
(((('4 + getnth ('1, os-current-obuffer (*id*, *os*))) < '8) ∧ (*id* < '16))

$\rightarrow$ (getnth ('4
+ (('8 $*$ *id*)
+ getnth ('1,
getnth (*id*, table ('8, os-obuffers (*os*)))))),
os-obuffers (*os*))
= getnth ('4 + getnth ('1, os-current-obuffer (*id*, *os*)),
os-current-obuffer (*id*, *os*)))

EVENT: Disable contract-getnth-from-current-obuffer.

THEOREM: contract-putnth-into-current-obuffer
(((*i* + getnth ('1, os-current-obuffer (*id*, *os*))) $<$ '8) $\wedge$ (*id* $<$ '16))
$\rightarrow$ (putnth (*value*,
i + (('8 $*$ *id*)
+ getnth ('1,
getnth (*id*, table ('8, os-obuffers (*os*)))))),
os-obuffers (*os*))
= putseg (putnth (*value*,
i + getnth ('1, os-current-obuffer (*id*, *os*)),
os-current-obuffer (*id*, *os*)),
'8 $*$ *id*,
os-obuffers (*os*)))

EVENT: Disable contract-putnth-into-current-obuffer.

THEOREM: contract-putnth-into-current-obuffer1
((*i* $\in$ **N**) $\wedge$ ((*i* $<$ '4) $\wedge$ (*id* $<$ '16)))
$\rightarrow$ (putnth (*value*, *i* + ('8 $*$ *id*), os-obuffers (*os*))
= putseg (putnth (*value*, *i*, os-current-obuffer (*id*, *os*)),
'8 $*$ *id*,
os-obuffers (*os*)))

EVENT: Disable contract-putnth-into-current-obuffer1.

THEOREM: contract-getnth-from-current-obuffer1
(('4 + getnth ('0, os-current-obuffer (*id*, *os*))) $<$ '8) $\wedge$ (*id* $<$ '16))
$\rightarrow$ (getnth ('4
+ (('8 $*$ *id*)
+ getnth ('0,
getnth (*id*, table ('8, os-obuffers (*os*)))))),
os-obuffers (*os*))
= getnth ('4 + getnth ('0, os-current-obuffer (*id*, *os*)),
os-current-obuffer (*id*, *os*)))

EVENT: Disable contract-getnth-from-current-obuffer1.

THEOREM: access-current-obuffer-qhead
$(id < \texttt{'16})$
$\rightarrow$ (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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
putseg (tm-cpu (*os*),
'9 ∗ array-qfirst (os-readyq (*os*)),
os-task-table (*os*)))

mod '256)
÷ '4)
= tm-error (*os*))

EVENT: Disable restore-error.

THEOREM: restore-svcflag
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
putseg (tm-cpu (*os*),
'9 ∗ array-qfirst (os-readyq (*os*)),
os-task-table (*os*)))
mod '512)
÷ '256)
= tm-svcflag (*os*))

EVENT: Disable restore-svcflag.

THEOREM: restore-svcid
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
putseg (tm-cpu (*os*),
'9 ∗ array-qfirst (os-readyq (*os*)),
os-task-table (*os*)))
÷ '512)
= tm-svcid (*os*))

EVENT: Disable restore-svcid.

DEFINITION:
os-task-table-with-delivered-character (*os*)
= putseg (putnth (array-qfirst (getnth (os-current-taskid (*os*),
table ('8, os-ibuffers (*os*)))),
'3,
tm-cpu (*os*)),
'9 ∗ array-qfirst (os-readyq (*os*)),
os-task-table (*os*))

THEOREM: length-os-task-table-with-delivered-character
length (os-task-table-with-delivered-character (*os*)) = '144

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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ ((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*) ∧ (¬ tm-waiting (*os*)))
→ (((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-character (*os*))
mod '256)
÷ '4)
= tm-error (*os*))

EVENT: Disable restore-error-for-tyi-svc-return.

THEOREM: restore-svcid-for-tyi-svc-return
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-character (*os*))
÷ '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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ ((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-message (*os*))

mod '4)
= tm-cc (*os*))

EVENT: Disable restore-cc-for-receive-svc-return.

THEOREM: restore-error-for-receive-svc-return
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-message (*os*))
mod '256)
÷ '4)
= tm-error (*os*))

EVENT: Disable restore-error-for-receive-svc-return.

THEOREM: restore-svcid-for-receive-svc-return
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((getnth ('8 + ('9 ∗ array-qfirst (os-readyq (*os*))),
os-task-table-with-delivered-message (*os*))
÷ '512)
= tm-svcid (*os*))

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*qemptyp2, *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)
=  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,
                                          putseg (os-task-table (tm),
                                                  '15,
                                                  putseg (os-segment-table (tm),
                                                          '159,
                                                          putseg (os-readyq (tm),
                                                                  '191,
                                                                  putseg (os-code (tm),
                                                                          '2547,
                                                                          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
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (tm-execute-clock-interrupt (*os*) = 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-clock-interrupt-handler-thru-save-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-clock-interrupt (*os*),
os-time-for-clock-interrupt-handler-thru-save-state (*os*))
= 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 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-clock-interrupt-handler-to-dispatcher
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-clock-interrupt-handler-thru-save-state (*os*),
os-time-for-clock-interrupt-handler-to-dispatcher (*os*))
= os-clock-interrupt-handler-to-dispatcher (*os*))

EVENT: Disable trace-clock-interrupt-handler-to-dispatcher.

DEFINITION:
os-time-for-clock-interrupt-handler-to-restore-state (*os*)
= '(tick 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-clock-interrupt-handler-to-restore-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-clock-interrupt-handler-to-dispatcher (*os*),
os-time-for-clock-interrupt-handler-to-restore-state (*os*))
= os-clock-interrupt-handler-to-restore-state (*os*))

EVENT: Disable trace-clock-interrupt-handler-to-restore-state.

DEFINITION:
os-clock-new-readyq (*os*)
= array-enq (array-qfirst (os-readyq (*os*)), array-deq (os-readyq (*os*)))

DEFINITION:
os-time-for-clock-interrupt-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-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-IPORTS OS)
 (TM-OPORTS OS))
NIL)
```

THEOREM: trace-clock-interrupt-handler-to-finish
(good-os (os) $\land$ ($\neg$ tm-waiting (os)))
$\rightarrow$ (tm-processor (os-clock-interrupt-handler-to-restore-state (os),
os-time-for-clock-interrupt-handler-to-finish (os))
= 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 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)
```

THEOREM: rewrite-clock-trace1
tm-processor (os-intended-clock-interrupt (*os*),
os-time-for-clock-interrupt-handler (*os*))
= tm-processor (tm-processor (os-intended-clock-interrupt (*os*),
os-time-for-clock-interrupt-handler-thru-save-state (*os*)),
os-cih-time1 (*os*))

EVENT: Disable rewrite-clock-trace1.

THEOREM: rewrite-clock-trace2
tm-processor (os-clock-interrupt-handler-thru-save-state (*os*), os-cih-time1 (*os*))
= tm-processor (tm-processor (os-clock-interrupt-handler-thru-save-state (*os*),
os-time-for-clock-interrupt-handler-to-dispatcher (*os*)),
os-cih-time2 (*os*))

EVENT: Disable rewrite-clock-trace2.

THEOREM: rewrite-clock-trace3
tm-processor (os-clock-interrupt-handler-to-dispatcher (*os*), os-cih-time2 (*os*))
= tm-processor (tm-processor (os-clock-interrupt-handler-to-dispatcher (*os*),
os-time-for-clock-interrupt-handler-to-restore-state (*os*)),
os-time-for-clock-interrupt-handler-to-finish (*os*))

EVENT: Disable rewrite-clock-trace3.

THEOREM: trace-clock-interrupt-handler
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-clock-interrupt (*os*),
os-time-for-clock-interrupt-handler (*os*))
= os-clock-interrupt-handler (*os*))

EVENT: Disable trace-clock-interrupt-handler.

DEFINITION:
os-intended-error-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,
                                                        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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-execute-error-interrupt (*os*) = 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-error-handler-thru-save-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-error-interrupt (*os*),
os-time-for-error-handler-thru-save-state (*os*))
= 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-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 ’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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-error-handler-to-dispatcher
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-error-handler-thru-save-state (*os*),
os-time-for-error-handler-to-dispatcher (*os*))
= os-error-handler-to-dispatcher (*os*))

EVENT: Disable trace-error-handler-to-dispatcher.

DEFINITION:
os-time-for-error-handler-to-wait-state (*os*)
= `'(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-qemptyp (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-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 '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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-error-handler-to-restore-state

$$
\begin{array}{ll}
 & (\text{good-os}\,(os) \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) \\
 & \quad \wedge \quad (\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))) \\
\rightarrow & (\text{tm-processor}\,(\text{os-error-handler-to-dispatcher}\,(os), \\
 & \qquad\qquad \text{os-time-for-error-handler-to-restore-state}\,(os)) \\
 & \quad = \quad \text{os-error-handler-to-restore-state}\,(os))
\end{array}
$$

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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-error-handler-to-finish
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
→ (tm-processor (os-error-handler-to-restore-state (*os*),
                 os-time-for-error-handler-to-finish (*os*))
  = os-error-handler-path2 (*os*))

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)
  ∧   ((¬ tm-waiting (os)) ∧ array-qemptyp (array-deq (os-readyq (os)))))
 →   (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)
```

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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
→ (tm-processor (os-intended-error-interrupt (*os*),
  os-time-for-error-handler-path2 (*os*))
  = os-error-handler-path2 (*os*))

EVENT: Disable trace-error-handler-path2.

DEFINITION:
os-time-for-error-handler (*os*)
= **if** array-qemptyp (array-deq (os-readyq (*os*)))
  then os-time-for-error-handler-path1 (*os*)
  else os-time-for-error-handler-path2 (*os*) **endif**

DEFINITION:
os-error-handler (*os*)
= **if** array-qemptyp (array-deq (os-readyq (*os*)))
  then os-error-handler-path1 (*os*)
  else os-error-handler-path2 (*os*) **endif**

THEOREM: trace-os-error-handler
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-error-interrupt (*os*),
  os-time-for-error-handler (*os*))
  = os-error-handler (*os*))

EVENT: Disable trace-os-error-handler.

DEFINITION:
os-intended-svc-send-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-mbuffers (*tm*),
'499,
putseg (os-code (*t*
'2547,
tm-memo
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-send-interrupt-expressions
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (tm-execute-svc-interrupt (*os*) = 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-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-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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-send-handler-thru-save-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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-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 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
                                (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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-send-handler-to-qfullp-test
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '0)))
→ (tm-processor (os-svc-send-handler-thru-save-state (*os*),
os-time-for-svc-send-handler-to-qfullp-test (*os*))
= 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*)
 ∧ ((¬ tm-waiting (*os*))
    ∧ (((tm-svcid (*os*) **mod** '4) = '0)

$$\begin{array}{lll} & \wedge & \text{array-qfullp}\,(\text{os-current-mbuffer}\,(\text{os-current-taskid}\,(os),\\ & & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-destid}\,(os,\ \texttt{'16}),\\ & & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)))))\\ \rightarrow & (\text{tm-processor}\,(\text{os-svc-send-handler-to-qfullp-test}\,(os),\\ & & \qquad\qquad\qquad\qquad\qquad\qquad \text{os-time-for-svc-send-handler-to-dispatcher}\,(os))\\ & = & \text{os-svc-send-handler-to-dispatcher}\,(os)) \end{array}$$

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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-send-handler-to-wait-state
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-send-handler-to-lpsw
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (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-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 '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))))))
      '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-IPORTS OS)
  (TM-OPORTS 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-IPORTS OS)
  (TM-OPORTS 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-IPORTS OS)
  (TM-OPORTS 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-IPORTS OS)
  (TM-OPORTS 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*))

$$\neq \quad \text{os-current-taskid}\,(os)))))$$
$$\rightarrow \quad (\text{tm-processor}\,(\text{os-svc-send-handler-thru-enqueue}\,(os),$$
$$\text{os-time-for-svc-send-handler-to-svc-resume-task3}\,(os))$$
$$= \quad \text{os-svc-send-handler-to-svc-resume-task3}\,(os))$$

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)

```
(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*)

$$
\begin{array}{lll}
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \quad \wedge \quad ((\text{getnth}\,(\texttt{'2} * \text{os-destid}\,(os,\ \texttt{'16}),\ \text{os-status-table}\,(os)) & \\
 & \qquad\quad = \quad \texttt{'3}) & \\
 & \qquad\quad \wedge \quad (\text{getnth}\,(\texttt{'1} + (\texttt{'2} * \text{os-destid}\,(os,\ \texttt{'16})), & \\
 & \qquad\qquad\qquad\quad \text{os-status-table}\,(os)) & \\
 & \qquad\qquad = \quad \text{os-current-taskid}\,(os))))) & \\
\rightarrow & (\text{tm-processor}\,(\text{os-svc-send-handler-to-svc-resume-task1}\,(os), & \\
 & \qquad\quad \text{os-time-for-svc-send-handler-to-svcr}\,(os)) & \\
 & \quad = \quad \text{os-svc-send-handler-path3}\,(os)) &
\end{array}
$$

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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-send-handler-to-svcr2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-svc-send-handler-to-svc-resume-task2 (*os*),
os-time-for-svc-send-handler-to-svcr (*os*))
= os-svc-send-handler-path4 (*os*))

EVENT: Disable trace-svc-send-handler-to-svcr2.

THEOREM: trace-svc-send-handler-to-svcr3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '0)
∧ (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os))
∧ array-qemptyp (array-deq (os-readyq (*os*)))))))
→ (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)

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)

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)

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*),

os-destid (*os*, '16),
os))
∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ (tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-path2 (*os*))
= os-svc-send-handler-path2 (*os*))

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)

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)

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
tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-path3 (*os*))
= tm-processor (tm-processor (os-intended-svc-send-interrupt (*os*),
os-time-for-svc-send-handler-thru-save-state (*os*)),
os-svc-send-time5 (*os*))

EVENT: Disable rewrite-svc-send-trace7.

THEOREM: rewrite-svc-send-trace8
tm-processor (os-svc-send-handler-thru-save-state (*os*), os-svc-send-time5 (*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-time6 (*os*))

EVENT: Disable rewrite-svc-send-trace8.

THEOREM: rewrite-svc-send-trace9
tm-processor (os-svc-send-handler-to-qfullp-test (*os*), os-svc-send-time6 (*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-time7 (*os*))

EVENT: Disable rewrite-svc-send-trace9.

THEOREM: rewrite-svc-send-trace10
tm-processor (os-svc-send-handler-thru-enqueue (*os*), os-svc-send-time7 (*os*))
= tm-processor (tm-processor (os-svc-send-handler-thru-enqueue (*os*),
os-time-for-svc-send-handler-to-svc-resume-task1 (*os*)),
os-time-for-svc-send-handler-to-svcr (*os*))

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 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)

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)

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 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

tm-processor (os-svc-send-handler-thru-enqueue (*os*), os-svc-send-time13 (*os*))
= tm-processor (tm-processor (os-svc-send-handler-thru-enqueue (*os*),
os-time-for-svc-send-handler-to-svc-resume-task3 (*os*)),
os-time-for-svc-send-handler-to-svcr (*os*))

EVENT: Disable rewrite-svc-send-trace18.

THEOREM: trace-svc-send-handler-path4b
(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-path4b (*os*))
= os-svc-send-handler-path4 (*os*))

EVENT: Disable trace-svc-send-handler-path4b.

DEFINITION:
os-time-for-svc-send-handler-path4 (*os*)
= **if** getnth ('2 ∗ os-destid (*os*, '16), os-status-table (*os*)) ≠ '3
then os-time-for-svc-send-handler-path4a (*os*)
else os-time-for-svc-send-handler-path4b (*os*) **endif**

THEOREM: trace-svc-send-handler-path4
(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)))

$$
\begin{array}{lll}
 & \wedge & (\neg\ ((\text{getnth}\ (\texttt{'2} * \text{os-destid}\ (os,\ \texttt{'16}),\\
 & & \qquad \text{os-status-table}\ (os))\\
 & & \quad = \ \texttt{'3})\\
 & & \wedge\ (\text{getnth}\ (\texttt{'1}\\
 & & \qquad + \ (\texttt{'2}\\
 & & \qquad\quad * \ \text{os-destid}\ (os,\\
 & & \qquad\qquad \texttt{'16})),\\
 & & \qquad \text{os-status-table}\ (os))\\
 & & \quad = \ \text{os-current-taskid}\ (os)))))))\\
\rightarrow & (\text{tm-processor}\ (\text{os-intended-svc-send-interrupt}\ (os),\\
 & \qquad \text{os-time-for-svc-send-handler-path4}\ (os))\\
 & = \ \text{os-svc-send-handler-path4}\ (os))
\end{array}
$$

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-qemptyp (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-qemptyp (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*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '0)))
→ (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*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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-qemptyp-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-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
            (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-qemptyp-test
$(\text{good-os}\,(os) \wedge ((\neg\ \text{tm-waiting}\,(os)) \wedge ((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4}) = \texttt{'1})))$

$\rightarrow$ (tm-processor (os-svc-receive-handler-thru-save-state (os),
os-time-for-svc-receive-handler-to-qemptyp-test (os))
= os-svc-receive-handler-to-qemptyp-test (os))

EVENT: Disable trace-svc-receive-handler-to-qemptyp-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-IPORTS 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-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)))))
→ (tm-processor (os-svc-receive-handler-to-qemptyp-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 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-receive-handler-to-wait-state
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-receive-handler-to-lpsw
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (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-qemptyp-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-IPORTS OS)
  (TM-OPORTS 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-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 ’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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (getnth ('2 ∗ os-srcid (*os*, '16), os-status-table (*os*))
   ≠ '2)))
→ (tm-processor (os-svc-receive-handler-thru-delivery (*os*),
  os-time-for-svc-receive-handler-to-svc-resume-task2 (*os*))
  = os-svc-receive-handler-to-svc-resume-task2 (*os*))

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-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 ’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-OPORTS 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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-receive-handler-to-svcr1

$$
\begin{array}{l}
(\text{good-os}\,(os)\\
\quad \wedge \quad ((\neg\ \text{tm-waiting}\,(os))\\
\qquad \wedge \quad ((\text{getnth}\,(\texttt{'2} * \text{os-srcid}\,(os,\ \texttt{'16}),\ \text{os-status-table}\,(os))\\
\qquad\quad = \quad \texttt{'2})\\
\qquad\quad \wedge \quad (\text{getnth}\,(\texttt{'1} + (\texttt{'2} * \text{os-srcid}\,(os,\ \texttt{'16})),\\
\qquad\qquad\qquad\quad \text{os-status-table}\,(os))\\
\qquad\qquad\quad = \quad \text{os-current-taskid}\,(os)))))\\
\rightarrow \quad (\text{tm-processor}\,(\text{os-svc-receive-handler-to-svc-resume-task1}\,(os),\\
\qquad\qquad\quad \text{os-time-for-svc-receive-handler-to-svcr}\,(os))\\
\qquad = \quad \text{os-svc-receive-handler-path3}\,(os))
\end{array}
$$

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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-receive-handler-to-svcr2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-svc-receive-handler-to-svc-resume-task2 (*os*),
os-time-for-svc-receive-handler-to-svcr (*os*))
= os-svc-receive-handler-path4 (*os*))

EVENT: Disable trace-svc-receive-handler-to-svcr2.

THEOREM: trace-svc-receive-handler-to-svcr3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-svc-receive-handler-to-svc-resume-task3 (*os*),
os-time-for-svc-receive-handler-to-svcr (*os*))
= os-svc-receive-handler-path4 (*os*))

EVENT: Disable trace-svc-receive-handler-to-svcr3.

DEFINITION:
os-svc-receive-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)

DEFINITION:
os-svc-receive-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-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)

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-qemptyp-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-qemptyp-test (*os*),
os-svc-receive-time2 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyp-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-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),

os))
$\wedge$ array-qemptyp (array-deq (os-readyq (os)))))))
$\rightarrow$ (tm-processor (os-intended-svc-receive-interrupt (os),
os-time-for-svc-receive-handler-path1 (os))
= os-svc-receive-handler-path1 (os))

EVENT: Disable trace-svc-receive-handler-path1.

DEFINITION:
os-svc-receive-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-receive-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)

DEFINITION:
os-time-for-svc-receive-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)

THEOREM: rewrite-svc-receive-trace4
tm-processor (os-intended-svc-receive-interrupt (os),
os-time-for-svc-receive-handler-path2 (os))
= tm-processor (tm-processor (os-intended-svc-receive-interrupt (os),
os-time-for-svc-receive-handler-thru-save-state (os)),
os-svc-receive-time3 (os))

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-qemptyp-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-qemptyp-test (*os*),
os-svc-receive-time4 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyp-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-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os))
∧ (¬ array-qemptyp (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 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 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)`

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)`

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
tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-svc-receive-time5 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-time-for-svc-receive-handler-to-qemptyp-test (*os*)),
os-svc-receive-time6 (*os*))

EVENT: Disable rewrite-svc-receive-trace8.

THEOREM: rewrite-svc-receive-trace9
tm-processor (os-svc-receive-handler-to-qemptyp-test (*os*),
os-svc-receive-time6 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyp-test (*os*),
os-time-for-svc-receive-handler-thru-delivery (*os*)),
os-svc-receive-time7 (*os*))

EVENT: Disable rewrite-svc-receive-trace9.

THEOREM: rewrite-svc-receive-trace10
tm-processor (os-svc-receive-handler-thru-delivery (*os*),
os-svc-receive-time7 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-delivery (*os*),
os-time-for-svc-receive-handler-to-svc-resume-task1 (*os*)),
os-time-for-svc-receive-handler-to-svcr (*os*))

EVENT: Disable rewrite-svc-receive-trace10.

THEOREM: trace-svc-receive-handler-path3
(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)))
∧ ((getnth ('2 ∗ os-srcid (*os*, '16),
os-status-table (*os*))
= '2)
∧ (getnth ('1
+ ('2
∗ os-srcid (*os*,
'16)),
os-status-table (*os*))

$$
\begin{array}{lll}
 & & = \quad \text{os-current-taskid}\,(os)))))))\\
\rightarrow & (\text{tm-processor}\,(\text{os-intended-svc-receive-interrupt}\,(os), & \\
 & \qquad \text{os-time-for-svc-receive-handler-path3}\,(os)) & \\
 & = \quad \text{os-svc-receive-handler-path3}\,(os)) &
\end{array}
$$

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)
```

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-qemptyp-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-qemptyp-test (*os*),
os-svc-receive-time9 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyp-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)

$$
\begin{array}{lll}
 & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os, \\
 & & \qquad \texttt{'16}), \\
 & & \qquad \text{os-current-taskid}\,(os), \\
 & & \qquad os))) \\
 & \wedge & (\text{getnth}\,(\texttt{'2} * \text{os-srcid}\,(os, \texttt{'16}), \\
 & & \qquad \text{os-status-table}\,(os)) \\
 & & \neq \texttt{'2}))))) \\
\rightarrow & (\text{tm-processor}\,(\text{os-intended-svc-receive-interrupt}\,(os), \\
 & & \text{os-time-for-svc-receive-handler-path4a}\,(os)) \\
 & = & \text{os-svc-receive-handler-path4}\,(os))
\end{array}
$$

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
tick tick tick tick tick tick tick tick tick tick)
```

THEOREM: rewrite-svc-receive-trace15
tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-path4b (*os*))
= tm-processor (tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-thru-save-state (*os*)),
os-svc-receive-time11 (*os*))

EVENT: Disable rewrite-svc-receive-trace15.

THEOREM: rewrite-svc-receive-trace16
tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-svc-receive-time11 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-save-state (*os*),
os-time-for-svc-receive-handler-to-qemptyp-test (*os*)),
os-svc-receive-time12 (*os*))

EVENT: Disable rewrite-svc-receive-trace16.

THEOREM: rewrite-svc-receive-trace17
tm-processor (os-svc-receive-handler-to-qemptyp-test (*os*),
os-svc-receive-time12 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-to-qemptyp-test (*os*),
os-time-for-svc-receive-handler-thru-delivery (*os*)),
os-svc-receive-time13 (*os*))

EVENT: Disable rewrite-svc-receive-trace17.

THEOREM: rewrite-svc-receive-trace18
tm-processor (os-svc-receive-handler-thru-delivery (*os*),
os-svc-receive-time13 (*os*))
= tm-processor (tm-processor (os-svc-receive-handler-thru-delivery (*os*),
os-time-for-svc-receive-handler-to-svc-resume-task3 (*os*)),
os-time-for-svc-receive-handler-to-svcr (*os*))

EVENT: Disable rewrite-svc-receive-trace18.

THEOREM: trace-svc-receive-handler-path4b
(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*)))
∧ ((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-intended-svc-receive-interrupt (*os*), os-time-for-svc-receive-handler-path4b (*os*))
= os-svc-receive-handler-path4 (*os*))

EVENT: Disable trace-svc-receive-handler-path4b.

DEFINITION:
os-time-for-svc-receive-handler-path4 (*os*)
= **if** getnth ('2 ∗ os-srcid (*os*, '16), os-status-table (*os*)) ≠ '2
then os-time-for-svc-receive-handler-path4a (*os*)
else os-time-for-svc-receive-handler-path4b (*os*) **endif**

THEOREM: trace-svc-receive-handler-path4
(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*)))
∧ (¬ ((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-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler-path4 (*os*))
= os-svc-receive-handler-path4 (*os*))

EVENT: Disable trace-svc-receive-handler-path4.

DEFINITION:
os-time-for-svc-receive-handler (*os*)
= **if** array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os))
then if array-qemptyp (array-deq (os-readyq (*os*)))
then os-time-for-svc-receive-handler-path1 (*os*)
else os-time-for-svc-receive-handler-path2 (*os*) **endif**
elseif (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*))
then os-time-for-svc-receive-handler-path3 (*os*)
else os-time-for-svc-receive-handler-path4 (*os*) **endif**

DEFINITION:
os-svc-receive-handler (*os*)
= **if** array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os))
then if array-qemptyp (array-deq (os-readyq (*os*)))
then os-svc-receive-handler-path1 (*os*)
else os-svc-receive-handler-path2 (*os*) **endif**
elseif (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*))
then os-svc-receive-handler-path3 (*os*)
else os-svc-receive-handler-path4 (*os*) **endif**

THEOREM: trace-svc-receive-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '1)))
→ (tm-processor (os-intended-svc-receive-interrupt (*os*),
os-time-for-svc-receive-handler (*os*))
= os-svc-receive-handler (*os*))

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 (*t*
'2547,
tm-memo
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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyi-handler-thru-save-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-svc-tyi-interrupt (*os*),
os-time-for-svc-tyi-handler-thru-save-state (*os*))
= os-svc-tyi-handler-thru-save-state (*os*))

EVENT: Disable trace-svc-tyi-handler-thru-save-state.

DEFINITION:
os-time-for-svc-tyi-handler-to-qemptyp-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-TYI-HANDLER-TO-QEMPTYP-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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyi-handler-to-qemptyp-test
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '3)))
→ (tm-processor (os-svc-tyi-handler-thru-save-state (*os*),
os-time-for-svc-tyi-handler-to-qemptyp-test (*os*))
= os-svc-tyi-handler-to-qemptyp-test (*os*))

EVENT: Disable trace-svc-tyi-handler-to-qemptyp-test.

DEFINITION:
os-time-for-svc-tyi-handler-to-dispatcher (*os*)
=
```
'(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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyi-handler-to-dispatcher

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*), *os*))))
→ (tm-processor (os-svc-tyi-handler-to-qemptyp-test (*os*),
                 os-time-for-svc-tyi-handler-to-dispatcher (*os*))
  = os-svc-tyi-handler-to-dispatcher (*os*))

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-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 ’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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyi-handler-to-wait-state
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (array-deq (os-readyq (*os*)))))
→ (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-qemptyp (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)

```
(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-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                         os)))))
→ (tm-processor (os-svc-tyi-handler-to-qemptyp-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-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-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*) ∧ (¬ tm-waiting (*os*)))
→ (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)

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-qemptyp-test (*os*)),
os-svc-tyi-time2 (*os*))

EVENT: Disable rewrite-svc-tyi-trace2.

THEOREM: rewrite-svc-tyi-trace3
tm-processor (os-svc-tyi-handler-to-qemptyp-test (*os*), os-svc-tyi-time2 (*os*))
= tm-processor (tm-processor (os-svc-tyi-handler-to-qemptyp-test (*os*),
os-time-for-svc-tyi-handler-to-dispatcher (*os*)),
os-time-for-svc-tyi-handler-to-wait-state (*os*))

EVENT: Disable rewrite-svc-tyi-trace3.

THEOREM: trace-svc-tyi-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '3)
∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os))
∧ array-qemptyp (array-deq (os-readyq (*os*)))))))
→ (tm-processor (os-intended-svc-tyi-interrupt (*os*),
os-time-for-svc-tyi-handler-path1 (*os*))
= os-svc-tyi-handler-path1 (*os*))

EVENT: Disable trace-svc-tyi-handler-path1.

DEFINITION:
os-svc-tyi-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-tyi-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)
```

DEFINITION:
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 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-qemptyp-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-qemptyp-test (*os*), os-svc-tyi-time4 (*os*))
= tm-processor (tm-processor (os-svc-tyi-handler-to-qemptyp-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-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os))
∧ (¬ array-qemptyp (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)

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)

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
tm-processor (os-svc-tyi-handler-thru-save-state (*os*), os-svc-tyi-time5 (*os*))
= tm-processor (tm-processor (os-svc-tyi-handler-thru-save-state (*os*),
os-time-for-svc-tyi-handler-to-qemptyp-test (*os*)),
os-svc-tyi-time6 (*os*))

EVENT: Disable rewrite-svc-tyi-trace8.

THEOREM: rewrite-svc-tyi-trace9
tm-processor (os-svc-tyi-handler-to-qemptyp-test (*os*), os-svc-tyi-time6 (*os*))
= tm-processor (tm-processor (os-svc-tyi-handler-to-qemptyp-test (*os*),
os-time-for-svc-tyi-handler-thru-delivery (*os*)),
os-time-for-svc-tyi-handler-to-svcr (*os*))

EVENT: Disable rewrite-svc-tyi-trace9.

THEOREM: trace-svc-tyi-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '3)
∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os))))))
→ (tm-processor (os-intended-svc-tyi-interrupt (*os*),
os-time-for-svc-tyi-handler-path3 (*os*))
= os-svc-tyi-handler-path3 (*os*))

EVENT: Disable trace-svc-tyi-handler-path3.

DEFINITION:
os-time-for-svc-tyi-handler (*os*)
= **if** array-qemptyp (os-current-ibuffer (os-current-taskid (*os*), *os*))
then if array-qemptyp (array-deq (os-readyq (*os*)))
then os-time-for-svc-tyi-handler-path1 (*os*)
else os-time-for-svc-tyi-handler-path2 (*os*) **endif**
else os-time-for-svc-tyi-handler-path3 (*os*) **endif**

DEFINITION:
os-svc-tyi-handler (*os*)
= **if** array-qemptyp (os-current-ibuffer (os-current-taskid (*os*), *os*))
then if array-qemptyp (array-deq (os-readyq (*os*)))
then os-svc-tyi-handler-path1 (*os*)
else os-svc-tyi-handler-path2 (*os*) **endif**
else os-svc-tyi-handler-path3 (*os*) **endif**

THEOREM: trace-svc-tyi-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '3)))
→ (tm-processor (os-intended-svc-tyi-interrupt (*os*),
os-time-for-svc-tyi-handler (*os*))
= os-svc-tyi-handler (*os*))

EVENT: Disable trace-svc-tyi-handler.

DEFINITION:
os-intended-svc-tyo-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-obuffers (*tm*),
'371,
putseg (os-code (*t*
'2547,
tm-memo
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-tyo-interrupt-expressions
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (tm-execute-svc-interrupt (*os*) = os-intended-svc-tyo-interrupt (*os*))

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-IPORTS OS)
 (TM-OPORTS OS))
NIL)
```

THEOREM: trace-svc-tyo-handler-thru-save-state
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-thru-save-state (*os*))
= 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 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

$$
\begin{aligned}
&(\text{good-os}\,(os) \wedge ((\neg\ \text{tm-waiting}\,(os)) \wedge ((\text{tm-svcid}\,(os)\ \textbf{mod}\ \text{’4}) = \text{’2})))\\
\rightarrow\quad &(\text{tm-processor}\,(\text{os-svc-tyo-handler-thru-save-state}\,(os),\\
&\qquad\qquad \text{os-time-for-svc-tyo-handler-to-qfullp-test}\,(os))\\
&\quad = \quad \text{os-svc-tyo-handler-to-qfullp-test}\,(os))
\end{aligned}
$$

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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyo-handler-to-wait-state
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyo-handler-to-lpsw
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (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-IPORTS OS)
 (TM-OPORTS OS))
NIL)
```

THEOREM: trace-svc-tyo-handler-thru-enqueue

(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-thru-enqueue (*os*))
  = os-svc-tyo-handler-thru-enqueue (*os*))

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-IPORTS OS)
  (TM-POST-OUTPUT-INTERRUPT (ARRAY-QFIRST (OS-READYQ OS))
                            (TM-OPORTS OS)))
 NIL)
```

THEOREM: trace-svc-tyo-handler-to-svcr1
(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-svcr1 (*os*))
  = os-svc-tyo-handler-path3 (*os*))

EVENT: Disable trace-svc-tyo-handler-to-svcr1.

DEFINITION:
os-time-for-svc-tyo-handler-to-svcr2 (*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-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-IPORTS OS)
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-svc-tyo-handler-to-svcr2
(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-svcr2 (*os*))
  = os-svc-tyo-handler-path4 (*os*))

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-qemptyp (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)

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 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

tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path2 (*os*))
= tm-processor (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-thru-save-state (*os*)),
os-svc-tyo-time3 (*os*))

EVENT: Disable rewrite-svc-tyo-trace4.

THEOREM: rewrite-svc-tyo-trace5
tm-processor (os-svc-tyo-handler-thru-save-state (*os*), os-svc-tyo-time3 (*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-time4 (*os*))

EVENT: Disable rewrite-svc-tyo-trace5.

THEOREM: rewrite-svc-tyo-trace6
tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*), os-svc-tyo-time4 (*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-lpsw (*os*))

EVENT: Disable rewrite-svc-tyo-trace6.

THEOREM: trace-svc-tyo-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '2)
∧ (array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))
∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path2 (*os*))
= os-svc-tyo-handler-path2 (*os*))

EVENT: Disable trace-svc-tyo-handler-path2.

DEFINITION:
os-svc-tyo-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)
```

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-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*)
∧ ((¬ tm-waiting (*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*))))))
→ (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
tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path4 (*os*))
= tm-processor (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-thru-save-state (*os*)),
os-svc-tyo-time7 (*os*))

EVENT: Disable rewrite-svc-tyo-trace10.

THEOREM: rewrite-svc-tyo-trace11
tm-processor (os-svc-tyo-handler-thru-save-state (*os*), os-svc-tyo-time7 (*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-time8 (*os*))

EVENT: Disable rewrite-svc-tyo-trace11.

THEOREM: rewrite-svc-tyo-trace12
tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*), os-svc-tyo-time8 (*os*))
= tm-processor (tm-processor (os-svc-tyo-handler-to-qfullp-test (*os*),
os-time-for-svc-tyo-handler-thru-enqueue (*os*)),
os-time-for-svc-tyo-handler-to-svcr2 (*os*))

EVENT: Disable rewrite-svc-tyo-trace12.

THEOREM: trace-svc-tyo-handler-path4
(good-os (*os*)
∧ ((¬ tm-waiting (*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*)))))))
→ (tm-processor (os-intended-svc-tyo-interrupt (*os*),
os-time-for-svc-tyo-handler-path4 (*os*))
= os-svc-tyo-handler-path4 (*os*))

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-qemptyp (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-qemptyp (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*) ∧ ((¬ tm-waiting (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '2)))
→ (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-IPORTS TM))
      '8
      (PUTNTH
       (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS TM))
                         (TM-IPORTS 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-IPORTS
          (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS TM))
                                    (TM-IPORTS 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-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 (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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-waiting-input-handler-thru-readyq-empty-test
(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-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)

```
(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-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 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-IPORTS OS))
   '2
   (PUTNTH '191
    '3
    (PUTNTH
           (PLUS '195
                 (GETNTH '1 (OS-READYQ OS)))
           '4
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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-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-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-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
     (PLUS '247
      (PLUS
        (TIMES '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))))
                  '4)))
     '4
     (PUTNTH (PLUS '211
                   (TIMES '2
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
```

```
              '5
              (TM-REGS OS)))))))
  (TM-CC-VALUE
   (TM-ALU-PLUS
    (PLUS '243
     (PLUS
          (TIMES '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))))
                    '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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS))
                           (TM-IPORTS OS))))
   ’2
   (PUTNTH
    (PLUS ’243
          (TIMES ’8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    ’3
    (PUTNTH
      (PLUS ’247
            (PLUS (TIMES ’8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH ’1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE ’8 (OS-IBUFFERS OS))))))
      ’4
      (PUTNTH (PLUS ’211
                    (TIMES ’2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              ’5
              (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH ’1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE ’8 (OS-IBUFFERS OS))))
                  (GETNTH ’3
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-iport-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-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-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-IPORTS OS))
                      (TM-IPORTS OS)))
    '2
    (PUTNTH
     (PLUS '243
           (TIMES '8
                  (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
     '3
     (PUTNTH
       (PLUS '247
             (PLUS (TIMES '8
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                   (GETNTH '1
                           (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                   (TABLE '8 (OS-IBUFFERS OS))))))
       '4
       (PUTNTH (PLUS '211
                     (TIMES '2
                            (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
               '5
               (TM-REGS OS)))))))
  (TM-CC-VALUE
   (TM-ALU-INCR-MOD (GETNTH '1
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                    (TABLE '8 (OS-IBUFFERS OS))))
                    (GETNTH '3
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS)))
           (OS-TASK-TABLE OS))
   '0
   (PUTNTH
    (GETNTH (PLUS '1
                  (TIMES '9
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
            (OS-TASK-TABLE OS))
    '1
    (PUTNTH
     (GETNTH (PLUS '8
                   (TIMES '9
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
             (OS-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 '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-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-IPORTS OS))
                (TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS '8
                         (TIMES '9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
           '4)
(QUOTIENT
 (REMAINDER
           (GETNTH (PLUS '8
```

```
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
            '256)
  '4)
  (QUOTIENT
   (REMAINDER
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
            '512)
   '256)
  (QUOTIENT (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
            '512)
  (BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                 (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (TM-SLIMIT OS)
  '0
  '0
  '1000
  (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS)))
          (OS-TASK-TABLE OS))
  '0
  (PUTNTH
   (GETNTH (PLUS '1
                 (TIMES '9
                        (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
           (OS-TASK-TABLE OS))
   '1
   (PUTNTH
    (GETNTH (PLUS '8
                  (TIMES '9
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
            (OS-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 '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 ’3217
                                        ’3310
                                        (TM-MEMORY OS)))))))))))))))
(GETSEG ’0
        ’8
        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                (TABLE ’9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS ’8
                         (TIMES ’9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
           ’4)
(QUOTIENT
 (REMAINDER
           (GETNTH (PLUS ’8
                         (TIMES ’9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
           ’256)
 ’4)
(QUOTIENT
 (REMAINDER
           (GETNTH (PLUS ’8
                         (TIMES ’9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
           ’512)
 ’256)
(QUOTIENT (GETNTH (PLUS ’8
                        (TIMES ’9
                               (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                  (OS-TASK-TABLE OS))
          ’512)
(BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
              (TABLE ’2 (OS-SEGMENT-TABLE OS))))
(LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
               (TABLE ’2 (OS-SEGMENT-TABLE OS))))
```

```
  (TM-SLIMIT OS)
  '0
  '0
  '1000
  (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS)))
           (OS-TASK-TABLE OS))
   '0
   (PUTNTH
    (GETNTH (PLUS '1
                  (TIMES '9
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
            (OS-TASK-TABLE OS))
    '1
    (PUTNTH
     (GETNTH (PLUS '8
                   (TIMES '9
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
             (OS-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 '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-CHARACTER OS)
                        '243
                        (PUTSEG (OS-CODE OS)
                                '2547
                                (PUTNTH '3217
                                        '3310
                                        (TM-MEMORY OS)))))))))))
(GETSEG '0
        '8
        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                (TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER (GETNTH (PLUS '8
                         (TIMES '9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
           '4)
(QUOTIENT
 (REMAINDER
           (GETNTH (PLUS '8
```

```
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
            '256)
  '4)
 (QUOTIENT
  (REMAINDER
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                   (OS-TASK-TABLE OS))
            '512)
  '256)
 (QUOTIENT (GETNTH (PLUS '8
                         (TIMES '9
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                  (OS-TASK-TABLE OS))
           '512)
 (BASE (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
               (TABLE '2 (OS-SEGMENT-TABLE OS))))
 (LIMIT (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                (TABLE '2 (OS-SEGMENT-TABLE OS))))
 (TM-SLIMIT OS)
 '0
 '0
 '1000
 (TM-CLEAR-INPUT-INTERRUPT (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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 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 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*)),

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*)))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path2 (*os*))
= os-waiting-input-handler-path2 (*os*))

EVENT: Disable trace-waiting-input-handler-path2.

DEFINITION:
os-waiting-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 tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:
os-waiting-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
tick tick)

DEFINITION:
os-time-for-waiting-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)

THEOREM: rewrite-waiting-input-trace7
tm-processor (os-intended-input-interrupt (*os*),

os-time-for-waiting-input-handler-path3 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*)),
os-waiting-input-time5 (*os*))

EVENT: Disable rewrite-waiting-input-trace7.

THEOREM: rewrite-waiting-input-trace8
tm-processor (os-waiting-input-handler-thru-readyq-empty-test (*os*),
os-waiting-input-time5 (*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-time6 (*os*))

EVENT: Disable rewrite-waiting-input-trace8.

THEOREM: rewrite-waiting-input-trace9
tm-processor (os-waiting-input-handler-to-ibuffer-full-test (*os*),
os-waiting-input-time6 (*os*))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test (*os*),
os-time-for-waiting-input-handler-to-dispatcher3 (*os*)),
os-time-for-waiting-input-handler-to-lpsw (*os*))

EVENT: Disable rewrite-waiting-input-trace9.

THEOREM: trace-waiting-input-handler-path3
(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*))))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path3 (*os*))
= os-waiting-input-handler-path3 (*os*))

EVENT: Disable trace-waiting-input-handler-path3.

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-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 (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-IPORTS OS))))
                    '5
                    (TM-REGS OS)))))
 (TM-CC-VALUE
  (TM-ALU-DIFFERENCE
                  (GETNTH (TIMES '2
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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*))

$\neq$ '5))))
$\rightarrow$ (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-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
           (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-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
     (PLUS '247
      (PLUS
        (TIMES '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))))
                  '4)))
     '4
     (PUTNTH (PLUS '211
                   (TIMES '2
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
             '5
             (TM-REGS OS)))))))
```

```
  (TM-CC-VALUE
   (TM-ALU-PLUS
    (PLUS '243
     (PLUS
          (TIMES '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))))
                    '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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-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
              (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-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              '5
              (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))
                  (GETNTH '3
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-waiting-input-handler-to-dispatcher5
(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)))
    $\wedge$ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
                                                         tm-iports (os))))))
$\rightarrow$ (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 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-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 (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-IPORTS OS))
                     (TM-IPORTS OS)))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
```

```
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
       '4
       (PUTNTH (PLUS '211
                     (TIMES '2
                            (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
               '5
               (TM-REGS OS)))))))
  (TM-CC-VALUE
   (TM-ALU-INCR-MOD (GETNTH '1
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                    (TABLE '8 (OS-IBUFFERS OS))))
                    (GETNTH '3
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-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
             (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-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH '191
    '3
    (PUTNTH
     (PLUS '247
      (PLUS
        (TIMES '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))))
                  '4)))
     '4
     (PUTNTH (PLUS '211
                   (TIMES '2
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                          (TM-IPORTS OS))
```

```
  (TM-OPORTS 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-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
            (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-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH '191
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-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 (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-IPORTS OS))
                     (TM-IPORTS OS)))
   '2
   (PUTNTH '191
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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 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 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 tick tick tick tick)

THEOREM: rewrite-waiting-input-trace10
tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path4 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*)),
os-waiting-input-time7 (*os*))

EVENT: Disable rewrite-waiting-input-trace10.

THEOREM: rewrite-waiting-input-trace11
tm-processor (os-waiting-input-handler-thru-readyq-empty-test (*os*),
os-waiting-input-time7 (*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-time8 (*os*))

EVENT: Disable rewrite-waiting-input-trace11.

THEOREM: rewrite-waiting-input-trace12
tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-waiting-input-time8 (*os*))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-time-for-waiting-input-handler-to-dispatcher4 (*os*)),
os-time-for-waiting-input-handler-to-wait (*os*))

EVENT: Disable rewrite-waiting-input-trace12.

THEOREM: trace-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))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path4 (*os*))
= os-waiting-input-handler-path4 (*os*))

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
(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*)))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
   os-time-for-waiting-input-handler-path5 (*os*))
  = os-waiting-input-handler-path5 (*os*))

EVENT: Disable trace-waiting-input-handler-path5.

DEFINITION:
os-waiting-input-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)

DEFINITION:
os-waiting-input-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)

DEFINITION:
os-time-for-waiting-input-handler-path6 (*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-input-trace16
tm-processor (os-intended-input-interrupt (*os*),
  os-time-for-waiting-input-handler-path6 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
   os-time-for-waiting-input-handler-thru-readyq-empty-test (*os*)),
  os-waiting-input-time11 (*os*))

EVENT: Disable rewrite-waiting-input-trace16.

THEOREM: rewrite-waiting-input-trace17
tm-processor (os-waiting-input-handler-thru-readyq-empty-test (*os*),
os-waiting-input-time11 (*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-time12 (*os*))

EVENT: Disable rewrite-waiting-input-trace17.

THEOREM: rewrite-waiting-input-trace18
tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-waiting-input-time12 (*os*))
= tm-processor (tm-processor (os-waiting-input-handler-to-ibuffer-full-test2 (*os*),
os-time-for-waiting-input-handler-to-dispatcher6 (*os*)),
os-time-for-waiting-input-handler-to-wait (*os*))

EVENT: Disable rewrite-waiting-input-trace18.

THEOREM: trace-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-iport-errorp (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-waiting-input-handler-path6 (*os*))
= os-waiting-input-handler-path6 (*os*))

EVENT: Disable trace-waiting-input-handler-path6.

DEFINITION:
os-time-for-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-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-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 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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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-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 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-IPORTS OS))
   '2
   (PUTNTH '191
    '3
    (PUTNTH
           (PLUS '195
                 (GETNTH '1 (OS-READYQ OS)))
           '4
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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)

```
(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-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-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-IPORTS OS))
                           (TM-IPORTS OS))))
   ’2
   (PUTNTH
    (PLUS ’243
          (TIMES ’8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    ’3
    (PUTNTH
     (PLUS ’247
      (PLUS
        (TIMES ’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))))
                  ’4)))
     ’4
     (PUTNTH (PLUS ’211
                   (TIMES ’2
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
             ’5
             (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-PLUS
  (PLUS ’243
   (PLUS
        (TIMES ’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))))
                  ’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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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
   (PLUS '256
         (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              '5
              (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))
                  (GETNTH '3
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
```

```
                                (TM-IPORTS OS))
  (TM-OPORTS OS))
 NIL)
```

THEOREM: trace-running-input-handler-to-resume-task2
(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-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-IPORTS OS))))
     '3
     (PUTNTH
       (PLUS '247
             (PLUS (TIMES '8
                          (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                   (GETNTH '1
                           (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                   (TABLE '8 (OS-IBUFFERS OS))))))
       '4
       (PUTNTH (PLUS '211
                     (TIMES '2
                            (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
               '5
               (TM-REGS OS)))))))
  (TM-CC-VALUE
   (TM-ALU-INCR-MOD (GETNTH '1
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                    (TABLE '8 (OS-IBUFFERS OS))))
                    (GETNTH '3
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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)))

$\wedge$ (¬ tm-iport-errorp (tm-interrupting-input-port (tm-iports (os)),
tm-iports (os)))))))

$\rightarrow$ (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)

```
(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-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-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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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-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 '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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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 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 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 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)

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 tick
  tick tick tick tick tick tick tick tick tick tick)

THEOREM: rewrite-running-input-trace4
tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path2 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-thru-save-state (*os*)),
os-running-input-time3 (*os*))

EVENT: Disable rewrite-running-input-trace4.

THEOREM: rewrite-running-input-trace5
tm-processor (os-running-input-handler-thru-save-state (*os*),
os-running-input-time3 (*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-time4 (*os*))

EVENT: Disable rewrite-running-input-trace5.

THEOREM: rewrite-running-input-trace6
tm-processor (os-running-input-handler-to-ibuffer-full-test (*os*),
os-running-input-time4 (*os*))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test (*os*),
os-time-for-running-input-handler-to-resume-task2 (*os*)),
os-time-for-running-input-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-input-trace6.

THEOREM: trace-running-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*)),
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*)))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path2 (*os*))
= os-running-input-handler-path2 (*os*))

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 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)

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)

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*)
∧ (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*))))))))
→ (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 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-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 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-IPORTS OS))))
                    '5
                    (TM-REGS OS))))))
 (TM-CC-VALUE
  (TM-ALU-DIFFERENCE
                  (GETNTH (TIMES '2
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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-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-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-IPORTS OS))
                           (TM-IPORTS OS))))
   ’2
   (PUTNTH
    (PLUS ’243
          (TIMES ’8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    ’3
    (PUTNTH
     (PLUS ’247
      (PLUS
        (TIMES ’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))))
                  ’4)))
     ’4
```

```
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              '5
              (TM-REGS OS)))))))
 (TM-CC-VALUE
  (TM-ALU-PLUS
   (PLUS '243
    (PLUS
         (TIMES '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))))
                   '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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-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 '3126
                                              '3310
                                              (TM-MEMORY OS)))))))))))))))))
(PUTNTH '3231
 '0
 (PUTNTH '3310
  '1
  (PUTNTH
   (PLUS '256
         (TM-ICHAR (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                           (TM-IPORTS OS))))
   '2
   (PUTNTH
    (PLUS '243
          (TIMES '8
                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              '5
              (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))
                  (GETNTH '3
```

```
                              (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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-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-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-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-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-IPORTS OS))))
    '3
    (PUTNTH
      (PLUS '247
            (PLUS (TIMES '8
                         (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS)))
                  (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))))
      '4
      (PUTNTH (PLUS '211
                    (TIMES '2
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              '5
              (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD (GETNTH '1
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                  (TABLE '8 (OS-IBUFFERS OS))))
                  (GETNTH '3
                          (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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-IPORTS OS))
                          (TM-IPORTS OS))
```

```
  (TM-OPORTS 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-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-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-IPORTS OS))
```

```
                              (TM-IPORTS OS))
  (TM-OPORTS 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-IPORTS OS))
                           (TM-IPORTS OS))
 (TM-OPORTS 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-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-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-IPORTS OS))
                            (TM-IPORTS OS))
  (TM-OPORTS 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*),

os-time-for-running-input-handler-to-lpsw (*os*))
= os-running-input-handler-path6 (*os*))

EVENT: Disable trace-running-input-handler-to-lpsw6.

DEFINITION:
os-running-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 tick 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-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)

DEFINITION:
os-time-for-running-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 tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick tick tick
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-trace10
tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path4 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-thru-save-state (*os*)),
os-running-input-time7 (*os*))

EVENT: Disable rewrite-running-input-trace10.

THEOREM: rewrite-running-input-trace11
tm-processor (os-running-input-handler-thru-save-state (*os*),

os-running-input-time7 (*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-time8 (*os*))

EVENT: Disable rewrite-running-input-trace11.

THEOREM: rewrite-running-input-trace12
tm-processor (os-running-input-handler-to-ibuffer-full-test2 (*os*),
os-running-input-time8 (*os*))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (*os*),
os-time-for-running-input-handler-to-resume-task4 (*os*)),
os-time-for-running-input-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-input-trace12.

THEOREM: trace-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))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path4 (*os*))
= os-running-input-handler-path4 (*os*))

EVENT: Disable trace-running-input-handler-path4.

DEFINITION:
os-running-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 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-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 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)
```

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
(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*)))))))
→ (tm-processor (os-intended-input-interrupt (*os*),
                 os-time-for-running-input-handler-path5 (*os*))
  = os-running-input-handler-path5 (*os*))

EVENT: Disable trace-running-input-handler-path5.

DEFINITION:
os-running-input-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)

DEFINITION:
os-running-input-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)

DEFINITION:
os-time-for-running-input-handler-path6 (*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-trace16
tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-path6 (*os*))
= tm-processor (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-running-input-handler-thru-save-state (*os*)),
os-running-input-time11 (*os*))

EVENT: Disable rewrite-running-input-trace16.

THEOREM: rewrite-running-input-trace17
tm-processor (os-running-input-handler-thru-save-state (*os*),
os-running-input-time11 (*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-time12 (*os*))

EVENT: Disable rewrite-running-input-trace17.

THEOREM: rewrite-running-input-trace18
tm-processor (os-running-input-handler-to-ibuffer-full-test2 (*os*),
os-running-input-time12 (*os*))
= tm-processor (tm-processor (os-running-input-handler-to-ibuffer-full-test2 (*os*),
os-time-for-running-input-handler-to-resume-task6 (*os*)),
os-time-for-running-input-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-input-trace18.

THEOREM: trace-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*))))))))
→ (tm-processor (os-intended-input-interrupt (*os*),

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
(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 (*os*))
    = os-running-input-handler (*os*))

EVENT: Disable trace-running-input-handler.

DEFINITION:
os-intended-output-interrupt (*tm*)
= tm-set-memory (putnth (tm-pc (*tm*),
                         'd0,
                         putnth (tm-sp (*tm*),
                                 'd1,
                                 putnth (tm-pack-psw (tm-cc (*tm*),
                                                      tm-error (*tm*),
                                                      tm-svcflag (*tm*),
                                                      tm-svcid (*tm*)),
                                         'd2,
                                         putnth (tm-interrupting-output-port (tm-oports (*tm*)),
                                                 'd9,
                                                 putseg (os-task-table (*tm*),
                                                         'd15,
                                                         putseg (os-segment-table (*tm*),
                                                                 'd159,
                                                                 putseg (os-readyq (*tm*),
                                                                         'd191,
                                                                         putseg (os-status-table (*tm*),
                                                                                 'd211,
                                                                                 putseg (os-obuffers (*tm*),
                                                                                         'd371,
                                                                                         putseg (os-code (*t
                                                                                                 'd2547,
                                                                                                 tm-memor
                 tm-set-pc ('d3135,
                            tm-set-sp ('d3310,
                                       tm-set-svmode ('d1,
                                                      tm-set-error ('d0,
                                                                    tm-set-rwstate ('d0,
                                                                                    tm-set-oports (tm-clear

                                                                                                   tm))))))

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-OPORTS 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-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS OS)))
 NIL)
```

THEOREM: trace-waiting-output-handler-thru-readyq-empty-test
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ tm-waiting (*os*)))
→ (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 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-OPORTS 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-OPORTS OS))
                      (OS-READYQ OS))
           '191
           (PUTSEG
                 (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
    '2
    (PUTNTH '191
     '3
     (PUTNTH
           (PLUS '195
                 (GETNTH '1 (OS-READYQ OS)))
           '4
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                            (TM-OPORTS 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))))

$\rightarrow$ (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-OPORTS 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-OPORTS OS))
                      (OS-READYQ OS))
```

```
         '191
         (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
   '2
   (PUTNTH
    (PLUS '371
          (TIMES '8
                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
    '3
    (PUTNTH
          (PLUS '195
                (GETNTH '1 (OS-READYQ OS)))
          '4
          (PUTNTH (PLUS '211
                        (TIMES '2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                             (TM-OPORTS 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-OPORTS 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-OPORTS OS))
                    (OS-READYQ OS))
         '191
         (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                         (TABLE '8 (OS-OBUFFERS OS))))
   '2
   (PUTNTH
    (PLUS '371
          (TIMES '8
                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
    '3
    (PUTNTH
          (PLUS '195
                (GETNTH '1 (OS-READYQ OS)))
          '4
```

```
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                   '5
                   (TM-REGS OS)))))))
  (TM-CC-VALUE
   (TM-ALU-INCR-MOD
                  (GETNTH '0
                          (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                  (TABLE '8 (OS-OBUFFERS OS))))
                  (GETNTH '3
                          (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                                   (TABLE '8 (OS-OBUFFERS OS))))
             (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
             (TM-OPORTS 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-OPORTS OS)))
           (OS-TASK-TABLE OS))
   '0
   (PUTNTH
    (GETNTH (PLUS '1
                  (TIMES '9
                         (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
            (OS-TASK-TABLE OS))
    '1
    (PUTNTH
     (GETNTH (PLUS '8
                   (TIMES '9
                          (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
             (OS-TASK-TABLE OS))
     '2
     (PUTNTH
      (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                      (OS-READYQ OS))
```

```
         '191
         (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                (TABLE '9 (OS-TASK-TABLE OS))))
(REMAINDER
          (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '4)
(QUOTIENT
 (REMAINDER
          (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '256)
 '4)
(QUOTIENT
 (REMAINDER
          (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '512)
 '256)
(QUOTIENT (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
```

```
          '512)
  (BASE (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (LIMIT (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                 (TABLE '2 (OS-SEGMENT-TABLE OS))))
  (TM-SLIMIT OS)
  '0
  '0
  '1000
  (TM-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS 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-OPORTS OS)))
           (OS-TASK-TABLE OS))
   '0
   (PUTNTH
    (GETNTH (PLUS '1
                  (TIMES '9
                         (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
            (OS-TASK-TABLE OS))
    '1
    (PUTNTH
     (GETNTH (PLUS '8
                   (TIMES '9
                          (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
```

```
          (OS-TASK-TABLE OS))
   ’2
   (PUTNTH
    (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                    (OS-READYQ OS))
         ’191
         (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                (TABLE ’9 (OS-TASK-TABLE OS))))
(REMAINDER
          (GETNTH (PLUS ’8
                        (TIMES ’9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          ’4)
(QUOTIENT
```

```
 (REMAINDER
          (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '256)
 '4)
(QUOTIENT
 (REMAINDER
          (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '512)
 '256)
(QUOTIENT (GETNTH (PLUS '8
                        (TIMES '9
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  (OS-TASK-TABLE OS))
          '512)
(BASE (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
              (TABLE '2 (OS-SEGMENT-TABLE OS))))
(LIMIT (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
               (TABLE '2 (OS-SEGMENT-TABLE OS))))
(TM-SLIMIT OS)
'0
'0
'1000
(TM-IPORTS OS)
(TM-START-OUTPUT
           (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                 (TABLE '8 (OS-OBUFFERS OS))))
           (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
           (TM-OPORTS 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-OPORTS 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-OPORTS OS))))
                    '5
                    (TM-REGS OS)))))
  (TM-CC-VALUE
   (TM-ALU-DIFFERENCE
                  (GETNTH (TIMES '2
                                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS OS)))
 NIL)
```

THEOREM: trace-waiting-output-handler-to-obuffer-empty-test2

(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))))

$\rightarrow$ (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-OPORTS 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-OPORTS OS))))
           '3
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                            (TM-OPORTS OS)))
NIL)
```

THEOREM: trace-waiting-output-handler-to-dispatcher3
(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-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 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-OPORTS 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-OPORTS OS))
                         (TABLE '8 (OS-OBUFFERS OS))))
   '2
   (PUTNTH
          (PLUS '371
                (TIMES '8
                       (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
          '3
          (PUTNTH (PLUS '211
                        (TIMES '2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  '5
                  (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD
                (GETNTH '0
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                (TABLE '8 (OS-OBUFFERS OS))))
                (GETNTH '3
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                                 (TABLE '8 (OS-OBUFFERS OS))))
           (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
           (TM-OPORTS OS)))
```

```
 NIL)
```

THEOREM: trace-waiting-output-handler-to-dispatcher4
(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-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-OPORTS 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-OPORTS 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-IPORTS OS)
 (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                            (TM-OPORTS OS)))
NIL)
```

THEOREM: trace-waiting-output-handler-to-wait1
(good-os (*os*)

$$
\begin{array}{rl}
\wedge & (\text{tm-some-output-interruptp}\,(\text{tm-oports}\,(os)) \wedge \text{tm-waiting}\,(os))) \\
\rightarrow & (\text{tm-processor}\,(\text{os-waiting-output-handler-to-dispatcher3}\,(os), \\
 & \qquad \text{os-time-for-waiting-output-handler-to-wait}\,(os)) \\
= & \text{os-waiting-output-handler-path3}\,(os))
\end{array}
$$

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-OPORTS 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-OPORTS OS))
                          (TABLE '8 (OS-OBUFFERS OS))))
    '2
    (PUTNTH '191
            '3
            (PUTNTH (PLUS '211
                          (TIMES '2
                                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-IPORTS OS)
 (TM-START-OUTPUT
            (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                  (TABLE '8 (OS-OBUFFERS OS))))
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
            (TM-OPORTS 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)

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)

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)

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
tm-processor (os-waiting-output-handler-to-obuffer-empty-test (*os*),
os-waiting-output-time2 (*os*))
= tm-processor (tm-processor (os-waiting-output-handler-to-obuffer-empty-test (*os*),
os-time-for-waiting-output-handler-to-dispatcher1 (*os*)),
os-time-for-waiting-output-handler-to-lpsw (*os*))

EVENT: Disable rewrite-waiting-output-trace3.

THEOREM: trace-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))))))
→ (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-waiting-output-handler-path1 (*os*))
= os-waiting-output-handler-path1 (*os*))

EVENT: Disable trace-waiting-output-handler-path1.

DEFINITION:
os-waiting-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 tick tick
tick tick tick tick tick tick tick tick tick)

DEFINITION:
os-waiting-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
      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 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*)

```
∧   (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-path2 (os))
     =   os-waiting-output-handler-path2 (os))
```

EVENT: Disable trace-waiting-output-handler-path2.

DEFINITION:
os-waiting-output-time5 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick tick tick tick tick tick)

DEFINITION:
os-waiting-output-time6 (*os*)
= '(tick tick tick tick tick tick tick tick tick tick
tick tick tick)

DEFINITION:
os-time-for-waiting-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)

THEOREM: rewrite-waiting-output-trace7
```
 tm-processor (os-intended-output-interrupt (os),
               os-time-for-waiting-output-handler-path3 (os))
=   tm-processor (tm-processor (os-intended-output-interrupt (os),
                                os-time-for-waiting-output-handler-thru-readyq-empty-test (os)),
                  os-waiting-output-time5 (os))
```

EVENT: Disable rewrite-waiting-output-trace7.

THEOREM: rewrite-waiting-output-trace8
```
 tm-processor (os-waiting-output-handler-thru-readyq-empty-test (os),
               os-waiting-output-time5 (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-time6 (os))
```

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*)
∧ (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-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-qemptyp (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-qemptyp (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-qemptyp (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-qemptyp (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*)
∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ tm-waiting (*os*)))
→ (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 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-OPORTS 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-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS OS)))
 NIL)
```

THEOREM: trace-running-output-handler-thru-save-state
(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-thru-save-state (*os*))
    = os-running-output-handler-thru-save-state (*os*))

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 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-OPORTS 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-OPORTS OS))
                       (OS-READYQ OS))
            '191
            (PUTSEG
                 (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
   ’2
   (PUTNTH ’191
    ’3
    (PUTNTH
          (PLUS ’195
                (GETNTH ’1 (OS-READYQ OS)))
          ’4
          (PUTNTH (PLUS ’211
                        (TIMES ’2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
```

```
                                (TM-OPORTS 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-OPORTS 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-OPORTS OS))
                     (OS-READYQ OS))
          ’191
          (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
   ’2
   (PUTNTH
    (PLUS ’371
          (TIMES ’8
                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
    ’3
    (PUTNTH
          (PLUS ’195
```

```
                (GETNTH '1 (OS-READYQ OS)))
          '4
          (PUTNTH (PLUS '211
                        (TIMES '2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                            (TM-OPORTS OS)))
NIL)
```

THEOREM: trace-running-output-handler-to-resume-task1
(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-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-OPORTS 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-OPORTS OS))
                     (OS-READYQ OS))
          '191
          (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                         (TABLE ’8 (OS-OBUFFERS OS))))
   ’2
   (PUTNTH
    (PLUS ’371
          (TIMES ’8
                 (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
    ’3
    (PUTNTH
          (PLUS ’195
                (GETNTH ’1 (OS-READYQ OS)))
          ’4
          (PUTNTH (PLUS ’211
                        (TIMES ’2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  ’5
                  (TM-REGS OS)))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD
                (GETNTH ’0
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                (TABLE ’8 (OS-OBUFFERS OS))))
                (GETNTH ’3
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                                  (TABLE '8 (OS-OBUFFERS OS))))
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
            (TM-OPORTS 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)

```
(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-OPORTS 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-OPORTS OS))
                     (OS-READYQ OS))
          ’191
          (PUTSEG
               (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS 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-OPORTS 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-OPORTS OS))
                      (OS-READYQ OS))
           '191
           (PUTSEG
                (OS-UPDATE-STATUS (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-IPORTS OS)
 (TM-START-OUTPUT
            (ARRAY-QFIRST (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                  (TABLE '8 (OS-OBUFFERS OS))))
            (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
            (TM-OPORTS 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-OPORTS 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-OPORTS OS))))
                   '5
                   (TM-REGS OS))))))
 (TM-CC-VALUE
  (TM-ALU-DIFFERENCE
                (GETNTH (TIMES '2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-IPORTS OS)
 (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                            (TM-OPORTS 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-OPORTS 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
           (PLUS '371
                 (TIMES '8
                        (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
           '3
           (PUTNTH (PLUS '211
                         (TIMES '2
                                (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                            (TM-OPORTS OS)))
NIL)
```

THEOREM: trace-running-output-handler-to-resume-task3
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
    ∧ ((¬ tm-waiting (*os*))

$\wedge$ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)), os)))))

$\rightarrow$ (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-OPORTS 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-OPORTS OS))
                         (TABLE '8 (OS-OBUFFERS OS))))
   '2
   (PUTNTH
          (PLUS '371
                (TIMES '8
                       (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
          '3
          (PUTNTH (PLUS '211
                        (TIMES '2
                               (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                  '5
                  (TM-REGS OS))))))
(TM-CC-VALUE
 (TM-ALU-INCR-MOD
                 (GETNTH '0
                         (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                 (TABLE '8 (OS-OBUFFERS OS))))
                 (GETNTH '3
                         (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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-OPORTS OS))
                                   (TABLE '8 (OS-OBUFFERS OS))))
             (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
             (TM-OPORTS 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-OPORTS 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-IPORTS OS)
  (TM-CLEAR-OUTPUT-INTERRUPT (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TM-OPORTS 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-OPORTS 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-OPORTS OS))
                                   (TABLE '8 (OS-OBUFFERS OS))))
             (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
             (TM-OPORTS OS)))
 NIL)
```

THEOREM: trace-running-output-handler-to-lpsw4
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ (¬ tm-waiting (*os*))))
→ (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
tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path1 (*os*))
= tm-processor (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-thru-save-state (*os*)),
os-running-output-time1 (*os*))

EVENT: Disable rewrite-running-output-trace1.

THEOREM: rewrite-running-output-trace2
tm-processor (os-running-output-handler-thru-save-state (*os*),
os-running-output-time1 (*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-time2 (*os*))

EVENT: Disable rewrite-running-output-trace2.

THEOREM: rewrite-running-output-trace3
tm-processor (os-running-output-handler-to-obuffer-empty-test (*os*),
os-running-output-time2 (*os*))
= tm-processor (tm-processor (os-running-output-handler-to-obuffer-empty-test (*os*),
os-time-for-running-output-handler-to-resume-task1 (*os*)),
os-time-for-running-output-handler-to-lpsw (*os*))

EVENT: Disable rewrite-running-output-trace3.

THEOREM: trace-running-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))))))
→ (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-running-output-handler-path1 (*os*))
= os-running-output-handler-path1 (*os*))

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 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)

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)

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 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

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad * \quad \text{tm-interrupting-output-port}\,(\text{tm-oports}\,(os)),\\
\qquad\qquad\qquad\qquad \text{os-status-table}\,(os))\\
\qquad\qquad\qquad \neq \quad \texttt{'4})\\
\qquad\qquad\qquad \wedge \quad \text{array-qemptyp}\,(\text{os-current-obuffer}\,(\text{tm-interrupting-output-port}\,(\text{tm-oports}\,(os)),\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os))))))\\
\rightarrow \quad (\text{tm-processor}\,(\text{os-intended-output-interrupt}\,(os),\\
\qquad\qquad\qquad\qquad \text{os-time-for-running-output-handler-path3}\,(os))\\
\qquad = \quad \text{os-running-output-handler-path3}\,(os))
\end{array}
$$

EVENT: Disable trace-running-output-handler-path3.

DEFINITION:
os-running-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`
`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-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)`

DEFINITION:
os-time-for-running-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 tick tick tick`
`tick tick tick tick tick tick tick tick tick tick`
`tick tick tick tick tick tick tick tick tick tick`
`tick tick 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-trace10
$$
\begin{array}{l}
\text{tm-processor}\,(\text{os-intended-output-interrupt}\,(os),\\
\qquad\qquad \text{os-time-for-running-output-handler-path4}\,(os))\\
= \quad \text{tm-processor}\,(\text{tm-processor}\,(\text{os-intended-output-interrupt}\,(os),
\end{array}
$$

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*)
∧ (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-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-qemptyp (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-qemptyp (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-qemptyp (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-qemptyp (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
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-tm (os-clock-interrupt-handler (*os*))

EVENT: Disable good-tm-os-clock-interrupt-handler.

THEOREM: tm-slimit-os-clock-interrupt-handler
tm-slimit (os-clock-interrupt-handler (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-clock-interrupt-handler.

THEOREM: os-code-os-clock-interrupt-handler
good-tm (*os*) → (os-code (os-clock-interrupt-handler (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-clock-interrupt-handler.

THEOREM: tm-interrupt-addresses-os-clock-interrupt-handler
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-clock-interrupt-handler (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-clock-interrupt-handler.

THEOREM: not-tm-in-supervisor-mode-os-clock-interrupt-handler
¬ tm-in-supervisor-mode (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-qemptyp (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
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-error-handler-path1 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-error-handler-path1 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-error-handler-path1 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-error-handler-path1 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-error-handler-path1 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-error-handler-path1 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-error-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-error-handler-path1
¬ tm-in-supervisor-mode (os-error-handler-path1 (*os*))

EVENT: Disable not-tm-in-supervisor-mode-os-error-handler-path1.

THEOREM: os-task-table-os-error-handler-path1
good-tm (*os*)
→ (os-task-table (os-error-handler-path1 (*os*)) = os-new-task-table (*os*))

EVENT: Disable os-task-table-os-error-handler-path1.

THEOREM: os-segment-table-os-error-handler-path1
good-tm (*os*)
→ (os-segment-table (os-error-handler-path1 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-error-handler-path1.

THEOREM: os-readyq-os-error-handler-path1

good-tm (*os*)
→ (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*)
→ (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*)
→ (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*)
→ (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*)
→ (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*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*qemptyp2.

THEOREM: good-status-list-os-status-table-os-error-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ 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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ 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*)
∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (array-deq (os-readyq (*os*)))))
→ 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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
→ 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*) → (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*)
→ ((getnth ('3, tm-memory (os-error-handler-path2 (*os*)))
  = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-error-handler-path2 (*os*)))
  = getnth ('4, tm-memory (*os*)))
 ∧ ((getnth ('5, tm-memory (os-error-handler-path2 (*os*)))
  = getnth ('5, tm-memory (*os*)))
 ∧ ((getnth ('6,
  tm-memory (os-error-handler-path2 (*os*)))
  = getnth ('6, tm-memory (*os*)))
 ∧ ((getnth ('7,
  tm-memory (os-error-handler-path2 (*os*)))
  = getnth ('7, tm-memory (*os*)))
 ∧ (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
¬ 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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ good-status-list (table ('2,
os-status-table (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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-error-handler-path2 (*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-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-error-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (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
tm-base (os-error-handler-path2 (*os*))
= os-new-base (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-base-os-error-handler-path2.

THEOREM: tm-limit-os-error-handler-path2
tm-limit (os-error-handler-path2 (*os*))
= os-new-limit (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-limit-os-error-handler-path2.

THEOREM: tm-base-tm-limit-os-error-handler-path2
good-tm (*os*)
→ ((tm-base (os-error-handler-path2 (*os*))
= base (getnth (os-current-taskid (os-error-handler-path2 (*os*)),

table ('2, os-segment-table (os)))))
∧ (tm-limit (os-error-handler-path2 (os))
= limit (getnth (os-current-taskid (os-error-handler-path2 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-error-handler-path2.

THEOREM: not-tm-waiting-os-error-handler-path2
¬ tm-waiting (os-error-handler-path2 (os))

EVENT: Disable not-tm-waiting-os-error-handler-path2.

THEOREM: good-os-os-error-handler-path2
(good-os (os)
∧ ((¬ tm-waiting (os))
∧ (¬ array-qemptyp (array-deq (os-readyq (os))))))
→ good-os (os-error-handler-path2 (os))

EVENT: Disable good-os-os-error-handler-path2.

THEOREM: good-os-os-error-handler
(good-os (os) ∧ (¬ tm-waiting (os))) → good-os (os-error-handler (os))

EVENT: Disable good-os-os-error-handler.

THEOREM: good-tm-os-intended-svc-send-interrupt
(good-os (os) ∧ ((¬ tm-waiting (os)) ∧ (¬ tm-errorp (os))))
→ 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
(good-os (os)
∧ ((¬ tm-waiting (os))
∧ ((¬ tm-errorp (os))
∧ (((tm-svcid (os) **mod** '4) = '0)
∧ (array-qfullp (os-current-mbuffer (os-current-taskid (os),
os-destid (os,
'16),
os))
∧ array-qemptyp (array-deq (os-readyq (os))))))))
→ 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
tm-slimit (os-svc-send-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-send-handler-path1.

THEOREM: os-code-os-svc-send-handler-path1
good-tm (*os*) → (os-code (os-svc-send-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-send-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path1
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-send-handler-path1 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path1
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-send-handler-path1 (*os*))
= 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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ 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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '0)
∧ (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*,
'16),
os))
∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ 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*)
∧ ((¬ tm-waiting (*os*))

```
∧  ((¬ tm-errorp (os))
    ∧  (((tm-svcid (os) mod '4) = '0)
        ∧  (array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                              os-destid (os,
                                                         '16),
                                              os))
            ∧  (¬ array-qemptyp (array-deq (os-readyq (os)))))))))
→  good-tm (os-svc-send-handler-path2 (os))
```

EVENT: Disable good-tm-os-svc-send-handler-path2.

THEOREM: tm-slimit-os-svc-send-handler-path2
tm-slimit (os-svc-send-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-send-handler-path2.

THEOREM: os-code-os-svc-send-handler-path2
good-tm (*os*) → (os-code (os-svc-send-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-send-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-send-handler-path2

```
good-tm (os)
→  ((getnth ('3, tm-memory (os-svc-send-handler-path2 (os)))
     =  getnth ('3, tm-memory (os)))
    ∧  ((getnth ('4, tm-memory (os-svc-send-handler-path2 (os)))
         =  getnth ('4, tm-memory (os)))
        ∧  ((getnth ('5, tm-memory (os-svc-send-handler-path2 (os)))
             =  getnth ('5, tm-memory (os)))
            ∧  ((getnth ('6,
                         tm-memory (os-svc-send-handler-path2 (os)))
                 =  getnth ('6, tm-memory (os)))
                ∧  ((getnth ('7,
                             tm-memory (os-svc-send-handler-path2 (os)))
                     =  getnth ('7, tm-memory (os)))
                    ∧  (getnth ('11,
                                tm-memory (os-svc-send-handler-path2 (os)))
                        =  getnth ('11,
                                   tm-memory (os)))))))))
```

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path2
¬ tm-in-supervisor-mode (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*)

```
→  ((tm-base (os-svc-send-handler-path2 (os))
     =  base (getnth (os-current-taskid (os-svc-send-handler-path2 (os)),
                      table ('2, os-segment-table (os)))))
   ∧  (tm-limit (os-svc-send-handler-path2 (os))
        =  limit (getnth (os-current-taskid (os-svc-send-handler-path2 (os)),
                          table ('2, os-segment-table (os))))))
```

EVENT: Disable tm-base-tm-limit-os-svc-send-handler-path2.

THEOREM: not-tm-waiting-os-svc-send-handler-path2
¬ tm-waiting (os-svc-send-handler-path2 (*os*))

EVENT: Disable not-tm-waiting-os-svc-send-handler-path2.

```
THEOREM: good-os-os-svc-send-handler-path2
 (good-os (os)
  ∧  ((¬ tm-waiting (os))
      ∧  ((¬ tm-errorp (os))
          ∧  (((tm-svcid (os) mod '4) = '0)
              ∧  (array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                                    os-destid (os,
                                                               '16),
                                                    os))
                  ∧  (¬ array-qemptyp (array-deq (os-readyq (os))))))))))
→  good-os (os-svc-send-handler-path2 (os))
```

EVENT: Disable good-os-os-svc-send-handler-path2.

```
THEOREM: good-tm-os-svc-send-handler-path3
 (good-os (os)
  ∧  ((¬ tm-waiting (os))
      ∧  ((¬ tm-errorp (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))))))))
→    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*)

```
∧   ((¬ tm-waiting (os))
     ∧   (¬ array-qfullp (os-current-mbuffer (os-current-taskid (os),
                                              os-destid (os, '16),
                                              os)))))
→   finite-number-queue-listp (table ('8,
                                      os-mbuffers-with-enqueued-message (os)),
                               '4,
                               '65536)
```

EVENT: Disable finite-number-queue-listp-os-mbuffers-with-enqueued-message.

```
THEOREM: os-status-table-os-svc-send-handler-path3
good-tm (os)
→   (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) ∧ (¬ tm-waiting (os)))
→   (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)
 ∧   ((¬ tm-waiting (os))
      ∧   (getnth ('2 * os-destid (os, '16), os-status-table (os))
           =   '3)))
→   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
¬ 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
tm-base (os-svc-send-handler-path3 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-svc-send-handler-path3.

THEOREM: tm-limit-os-svc-send-handler-path3
tm-limit (os-svc-send-handler-path3 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-svc-send-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (getnth ('2 ∗ os-destid (*os*, '16), os-status-table (*os*))
= '3)))
→ ((tm-base (os-svc-send-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-svc-send-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-svc-send-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-svc-send-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-send-handler-path3.

THEOREM: finite-numberp-os-destid
(os-destid (*os*, '16) ∈ **N**)
∧ ((os-destid (*os*, '16) < '16)
∧ finite-numberp (os-destid (*os*, '16), '16))

EVENT: Disable finite-numberp-os-destid.

THEOREM: good-os-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '0)
∧ ((¬ array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*,
'16),
os)))

$$\begin{array}{l}
\wedge\quad((\text{getnth}\,(\texttt{'2}\ast\text{os-destid}\,(os,\ \texttt{'16}),\\
\qquad\qquad\text{os-status-table}\,(os))\\
\qquad=\quad\texttt{'3})\\
\qquad\wedge\quad(\text{getnth}\,(\texttt{'1}\\
\qquad\qquad\qquad+\quad(\texttt{'2}\\
\qquad\qquad\qquad\qquad\ast\quad\text{os-destid}\,(os,\\
\qquad\qquad\qquad\qquad\qquad\qquad\texttt{'16})),\\
\qquad\qquad\qquad\text{os-status-table}\,(os))\\
\qquad\qquad=\quad\text{os-current-taskid}\,(os)))))))\\
\rightarrow\quad\text{good-os}\,(\text{os-svc-send-handler-path3}\,(os))
\end{array}$$

EVENT: Disable good-os-os-svc-send-handler-path3.

THEOREM: good-tm-os-svc-send-handler-path4

$$\begin{array}{l}
(\text{good-os}\,(os)\\
\wedge\quad((\neg\ \text{tm-waiting}\,(os))\\
\qquad\wedge\quad((\neg\ \text{tm-errorp}\,(os))\\
\qquad\qquad\wedge\quad(((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4})=\texttt{'0})\\
\qquad\qquad\qquad\wedge\quad((\neg\ \text{array-qfullp}\,(\text{os-current-mbuffer}\,(\text{os-current-taskid}\,(os),\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{os-destid}\,(os,\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\texttt{'16}),\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)))\\
\qquad\qquad\qquad\wedge\quad(\neg\ ((\text{getnth}\,(\texttt{'2}\\
\qquad\qquad\qquad\qquad\qquad\qquad\ast\quad\text{os-destid}\,(os,\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\texttt{'16}),\\
\qquad\qquad\qquad\qquad\qquad\qquad\text{os-status-table}\,(os))\\
\qquad\qquad\qquad\qquad\qquad=\quad\texttt{'3})\\
\qquad\qquad\qquad\qquad\wedge\quad(\text{getnth}\,(\texttt{'1}\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad+\quad(\texttt{'2}\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\ast\quad\text{os-destid}\,(os,\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\texttt{'16})),\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{os-status-table}\,(os))\\
\qquad\qquad\qquad\qquad\qquad\qquad=\quad\text{os-current-taskid}\,(os)))))))))\\
\rightarrow\quad\text{good-tm}\,(\text{os-svc-send-handler-path4}\,(os))
\end{array}$$

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
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-send-handler-path4 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-send-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-send-handler-path4
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-send-handler-path4 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-svc-send-handler-path4.

THEOREM: os-segment-table-os-svc-send-handler-path4
good-tm (*os*)
→ (os-segment-table (os-svc-send-handler-path4 (*os*))
= os-segment-table (*os*))

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*) ∧ (¬ tm-waiting (*os*)))
→ ((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*)))))
∧ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*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*)))))))))
→ good-os (os-svc-send-handler-path4 (*os*))

EVENT: Disable good-os-os-svc-send-handler-path4.

THEOREM: good-os-os-svc-send-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '0))))
→ good-os (os-svc-send-handler (*os*))

EVENT: Disable good-os-os-svc-send-handler.

THEOREM: good-tm-os-intended-svc-receive-interrupt
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-tm (os-intended-svc-receive-interrupt (*os*))

EVENT: Disable good-tm-os-intended-svc-receive-interrupt.

THEOREM: good-tm-os-svc-receive-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '1)
    ∧ (array-qemptyp (os-current-mbuffer (os-srcid (*os*), '16), os-current-taskid (*os*), *os*))
    ∧ array-qemptyp (array-deq (os-readyq (*os*)))))))
→ good-tm (os-svc-receive-handler-path1 (*os*))

EVENT: Disable good-tm-os-svc-receive-handler-path1.

THEOREM: tm-slimit-os-svc-receive-handler-path1
tm-slimit (os-svc-receive-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-receive-handler-path1.

THEOREM: os-code-os-svc-receive-handler-path1
good-tm (*os*) → (os-code (os-svc-receive-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-receive-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-receive-handler-path1
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-receive-handler-path1 (*os*)))
  = getnth ('3, tm-memory (*os*)))

```
∧  ((getnth ('4, tm-memory (os-svc-receive-handler-path1 (os)))
     =  getnth ('4, tm-memory (os)))
    ∧  ((getnth ('5,
                tm-memory (os-svc-receive-handler-path1 (os)))
         =  getnth ('5, tm-memory (os)))
        ∧  ((getnth ('6,
                    tm-memory (os-svc-receive-handler-path1 (os)))
             =  getnth ('6, tm-memory (os)))
            ∧  ((getnth ('7,
                        tm-memory (os-svc-receive-handler-path1 (os)))
                 =  getnth ('7, tm-memory (os)))
                ∧  (getnth ('11,
                           tm-memory (os-svc-receive-handler-path1 (os)))
                    =  getnth ('11,
                               tm-memory (os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path1
¬ tm-in-supervisor-mode (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*)
→ (os-task-table (os-svc-receive-handler-path1 (*os*))
    = 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*)
→ (os-segment-table (os-svc-receive-handler-path1 (*os*))
    = 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*)
→ (os-readyq (os-svc-receive-handler-path1 (*os*))
    = array-deq (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '1)
    ∧ (array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16), os-current-taskid (*os*), *os*))
     ∧ array-qemptyp (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '1)
    ∧ (array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16), os-current-taskid (*os*), *os*))
     ∧ (¬ array-qemptyp (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
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-receive-handler-path2 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path2
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-receive-handler-path2 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-svc-receive-handler-path2.

THEOREM: os-segment-table-os-svc-receive-handler-path2
good-tm (*os*)
→ (os-segment-table (os-svc-receive-handler-path2 (*os*))
= os-segment-table (*os*))

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*) ∧ (¬ tm-waiting (*os*)))
→ 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*)
→ ((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*)))))
∧ (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
¬ 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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))

$$\begin{array}{l}
\quad \wedge \quad (((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4}) = \texttt{'1}) \\
\qquad \wedge \quad (\text{array-qemptyp}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-current-taskid}\,(os), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad os)) \\
\qquad\qquad\qquad \wedge \quad (\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))))))))) \\
\rightarrow \quad \text{good-os}\,(\text{os-svc-receive-handler-path2}\,(os))
\end{array}$$

EVENT: Disable good-os-os-svc-receive-handler-path2.

THEOREM: good-tm-os-svc-receive-handler-path3

$$\begin{array}{l}
(\text{good-os}\,(os) \\
\wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
\qquad \wedge \quad ((\neg\ \text{tm-errorp}\,(os)) \\
\qquad\quad \wedge \quad (((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4}) = \texttt{'1}) \\
\qquad\qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-current-taskid}\,(os), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad os))) \\
\qquad\qquad\qquad\quad \wedge \quad ((\text{getnth}\,(\texttt{'2} * \text{os-srcid}\,(os,\ \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\quad = \quad \texttt{'2}) \\
\qquad\qquad\qquad\qquad\quad \wedge \quad (\text{getnth}\,(\texttt{'1} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad + \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad * \quad \text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'16})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad = \quad \text{os-current-taskid}\,(os)))))))) \\
\rightarrow \quad \text{good-tm}\,(\text{os-svc-receive-handler-path3}\,(os))
\end{array}$$

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

```
good-tm (os)
→   ((getnth ('3, tm-memory (os-svc-receive-handler-path3 (os)))
      =   getnth ('3, tm-memory (os)))
     ∧   ((getnth ('4, tm-memory (os-svc-receive-handler-path3 (os)))
           =   getnth ('4, tm-memory (os)))
          ∧   ((getnth ('5,
                        tm-memory (os-svc-receive-handler-path3 (os)))
                =   getnth ('5, tm-memory (os)))
               ∧   ((getnth ('6,
                             tm-memory (os-svc-receive-handler-path3 (os)))
                     =   getnth ('6, tm-memory (os)))
                    ∧   ((getnth ('7,
                                  tm-memory (os-svc-receive-handler-path3 (os)))
                          =   getnth ('7, tm-memory (os)))
                         ∧   (getnth ('11,
                                      tm-memory (os-svc-receive-handler-path3 (os)))
                              =   getnth ('11,
                                          tm-memory (os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path3
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-receive-handler-path3 (*os*))
= 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) → (('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
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),

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-qemptyp (os-current-mbuffer (os-srcid (os, '16),

os-current-taskid (*os*),
os)))))
→ finite-number-queue-listp (table ('8,
os-mbuffers-with-dequeued-message (*os*)),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-mbuffers-with-dequeued-message.

THEOREM: os-status-table-os-svc-receive-handler-path3
good-tm (*os*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (getnth ('2 ∗ os-srcid (*os*, '16), os-status-table (*os*))
= '2)))
→ 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
¬ 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
tm-limit (os-svc-receive-handler-path3 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-svc-receive-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-receive-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (getnth ('2 ∗ os-srcid (*os*, '16), os-status-table (*os*))
= '2)))
→ ((tm-base (os-svc-receive-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-svc-receive-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-svc-receive-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-svc-receive-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-receive-handler-path3.

THEOREM: finite-numberp-os-srcid
(os-srcid (*os*, '16) ∈ **N**)
∧ ((os-srcid (*os*, '16) < '16)
∧ finite-numberp (os-srcid (*os*, '16), '16))

EVENT: Disable finite-numberp-os-srcid.

THEOREM: good-os-os-svc-receive-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '1)
∧ ((¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*,
'16),
os-current-taskid (*os*),
os)))
∧ ((getnth ('2 ∗ os-srcid (*os*, '16),
os-status-table (*os*))
= '2)
∧ (getnth ('1
+ ('2

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'16})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad = \quad \text{os-current-taskid}\,(os))))))))) \\
\rightarrow \quad \text{good-os}\,(\text{os-svc-receive-handler-path3}\,(os))
\end{array}
$$

EVENT: Disable good-os-os-svc-receive-handler-path3.

THEOREM: good-tm-os-svc-receive-handler-path4

$$
\begin{array}{l}
(\text{good-os}\,(os) \\
\quad \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
\qquad \wedge \quad ((\neg\ \text{tm-errorp}\,(os)) \\
\qquad\quad \wedge \quad (((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4}) = \texttt{'1}) \\
\qquad\qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-current-taskid}\,(os), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad os))) \\
\qquad\qquad\qquad\qquad \wedge \quad (\neg\ ((\text{getnth}\,(\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \texttt{'2}) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{getnth}\,(\texttt{'1} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \quad (\texttt{'2} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad * \quad \text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \texttt{'16})), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{os-current-taskid}\,(os)))))))))) \\
\rightarrow \quad \text{good-tm}\,(\text{os-svc-receive-handler-path4}\,(os))
\end{array}
$$

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

```
good-tm (os)
→  ((getnth ('3, tm-memory (os-svc-receive-handler-path4 (os)))
     =  getnth ('3, tm-memory (os)))
    ∧  ((getnth ('4, tm-memory (os-svc-receive-handler-path4 (os)))
         =  getnth ('4, tm-memory (os)))
        ∧  ((getnth ('5,
                    tm-memory (os-svc-receive-handler-path4 (os)))
             =  getnth ('5, tm-memory (os)))
            ∧  ((getnth ('6,
                        tm-memory (os-svc-receive-handler-path4 (os)))
                 =  getnth ('6, tm-memory (os)))
                ∧  ((getnth ('7,
                            tm-memory (os-svc-receive-handler-path4 (os)))
                     =  getnth ('7, tm-memory (os)))
                    ∧  (getnth ('11,
                                tm-memory (os-svc-receive-handler-path4 (os)))
                        =  getnth ('11,
                                   tm-memory (os)))))))))
```

EVENT: Disable tm-interrupt-addresses-os-svc-receive-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-receive-handler-path4
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-receive-handler-path4 (*os*))
= 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
good-tm (*os*)
→ (os-segment-table (os-svc-receive-handler-path4 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-svc-receive-handler-path4.

THEOREM: os-readyq-os-svc-receive-handler-path4
good-tm (*os*)
→ (os-readyq (os-svc-receive-handler-path4 (*os*)) = 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
tm-limit (os-svc-receive-handler-path4 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-svc-receive-handler-path4.

THEOREM: tm-base-tm-limit-os-svc-receive-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-svc-receive-handler-path4 (*os*))
= base (getnth (os-current-taskid (os-svc-receive-handler-path4 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-svc-receive-handler-path4 (*os*))
= limit (getnth (os-current-taskid (os-svc-receive-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-receive-handler-path4.

THEOREM: good-os-os-svc-receive-handler-path4
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '1)
∧ ((¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*,
'16),
os-current-taskid (*os*),
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*)))))))))
→ good-os (os-svc-receive-handler-path4 (*os*))

EVENT: Disable good-os-os-svc-receive-handler-path4.

THEOREM: good-os-os-svc-receive-handler

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '1))))
→ good-os (os-svc-receive-handler (*os*))

EVENT: Disable good-os-os-svc-receive-handler.

THEOREM: good-tm-os-intended-svc-tyi-interrupt
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-tm (os-intended-svc-tyi-interrupt (*os*))

EVENT: Disable good-tm-os-intended-svc-tyi-interrupt.

THEOREM: good-tm-os-svc-tyi-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                          os))
     ∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ good-tm (os-svc-tyi-handler-path1 (*os*))

EVENT: Disable good-tm-os-svc-tyi-handler-path1.

THEOREM: tm-slimit-os-svc-tyi-handler-path1
tm-slimit (os-svc-tyi-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyi-handler-path1.

THEOREM: os-code-os-svc-tyi-handler-path1
good-tm (*os*) → (os-code (os-svc-tyi-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyi-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path1
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyi-handler-path1 (*os*)))
  = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-svc-tyi-handler-path1 (*os*)))
  = getnth ('4, tm-memory (*os*)))
 ∧ ((getnth ('5, tm-memory (os-svc-tyi-handler-path1 (*os*)))

$$
\begin{array}{ll}
= & \text{getnth}\,(\texttt{'5},\ \text{tm-memory}\,(os))) \\
\wedge & ((\text{getnth}\,(\texttt{'6}, \\
 & \qquad \text{tm-memory}\,(\text{os-svc-tyi-handler-path1}\,(os))) \\
 & \quad = \text{getnth}\,(\texttt{'6},\ \text{tm-memory}\,(os))) \\
 & \wedge ((\text{getnth}\,(\texttt{'7}, \\
 & \qquad \text{tm-memory}\,(\text{os-svc-tyi-handler-path1}\,(os))) \\
 & \quad = \text{getnth}\,(\texttt{'7},\ \text{tm-memory}\,(os))) \\
 & \wedge (\text{getnth}\,(\texttt{'11}, \\
 & \qquad \text{tm-memory}\,(\text{os-svc-tyi-handler-path1}\,(os))) \\
 & \quad = \text{getnth}\,(\texttt{'11}, \\
 & \qquad \text{tm-memory}\,(os)))))))
\end{array}
$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path1
$\neg$ 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
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                          os))
     ∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ good-os (os-svc-tyi-handler-path1 (*os*))

EVENT: Disable good-os-os-svc-tyi-handler-path1.

THEOREM: good-tm-os-svc-tyi-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                          os))
     ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ good-tm (os-svc-tyi-handler-path2 (*os*))

EVENT: Disable good-tm-os-svc-tyi-handler-path2.

THEOREM: tm-slimit-os-svc-tyi-handler-path2
tm-slimit (os-svc-tyi-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyi-handler-path2.

THEOREM: os-code-os-svc-tyi-handler-path2
good-tm (*os*) → (os-code (os-svc-tyi-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyi-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path2
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyi-handler-path2 (*os*)))
   = getnth ('3, tm-memory (*os*)))
  ∧ ((getnth ('4, tm-memory (os-svc-tyi-handler-path2 (*os*)))
     = getnth ('4, tm-memory (*os*)))
    ∧ ((getnth ('5, tm-memory (os-svc-tyi-handler-path2 (*os*)))
       = getnth ('5, tm-memory (*os*)))
      ∧ ((getnth ('6,

```
                tm-memory (os-svc-tyi-handler-path2 (os)))
  =    getnth ('6, tm-memory (os)))
∧    ((getnth ('7,
                tm-memory (os-svc-tyi-handler-path2 (os)))
       =    getnth ('7, tm-memory (os)))
      ∧    (getnth ('11,
                    tm-memory (os-svc-tyi-handler-path2 (os)))
           =    getnth ('11,
                        tm-memory (os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path2
¬ 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*)
→ (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*)
→ (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*)
→ (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*)
→ (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
good-tm (*os*)
→ (os-obuffers (os-svc-tyi-handler-path2 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-tyi-handler-path2.

THEOREM: os-mbuffers-os-svc-tyi-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-svc-tyi-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyi-handler-path2.

THEOREM: os-status-table-os-svc-tyi-handler-path2
good-tm (*os*)
→ (os-status-table (os-svc-tyi-handler-path2 (*os*))
= os-update-status (array-qfirst (os-readyq (*os*)), '5, '0, *os*))

EVENT: Disable os-status-table-os-svc-tyi-handler-path2.

THEOREM: os-ready-set-os-svc-tyi-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (os-ready-set (os-svc-tyi-handler-path2 (*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-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ permutation (mapup-queue (array-deq (os-readyq (*os*))),
os-ready-set (os-svc-tyi-handler-path2 (*os*)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-svc-tyi-handler-path2.

THEOREM: tm-base-os-svc-tyi-handler-path2
tm-base (os-svc-tyi-handler-path2 (*os*))
= os-new-base (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-base-os-svc-tyi-handler-path2.

THEOREM: tm-limit-os-svc-tyi-handler-path2
tm-limit (os-svc-tyi-handler-path2 (*os*))
= os-new-limit (array-qfirst (array-deq (os-readyq (*os*))), *os*)

EVENT: Disable tm-limit-os-svc-tyi-handler-path2.

THEOREM: tm-base-tm-limit-os-svc-tyi-handler-path2
good-tm (*os*)
→ ((tm-base (os-svc-tyi-handler-path2 (*os*))
= base (getnth (os-current-taskid (os-svc-tyi-handler-path2 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-svc-tyi-handler-path2 (*os*))
= limit (getnth (os-current-taskid (os-svc-tyi-handler-path2 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-tyi-handler-path2.

THEOREM: not-tm-waiting-os-svc-tyi-handler-path2
¬ tm-waiting (os-svc-tyi-handler-path2 (*os*))

EVENT: Disable not-tm-waiting-os-svc-tyi-handler-path2.

THEOREM: good-os-os-svc-tyi-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '3)
∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os))
∧ (¬ array-qemptyp (array-deq (os-readyq (*os*)))))))))
→ good-os (os-svc-tyi-handler-path2 (*os*))

EVENT: Disable good-os-os-svc-tyi-handler-path2.

THEOREM: good-tm-os-svc-tyi-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '3)
∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os)))))))
→ good-tm (os-svc-tyi-handler-path3 (*os*))

EVENT: Disable good-tm-os-svc-tyi-handler-path3.

THEOREM: tm-slimit-os-svc-tyi-handler-path3
tm-slimit (os-svc-tyi-handler-path3 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyi-handler-path3.

THEOREM: os-code-os-svc-tyi-handler-path3
good-tm (*os*) → (os-code (os-svc-tyi-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyi-handler-path3.

THEOREM: tm-interrupt-addresses-os-svc-tyi-handler-path3
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-tyi-handler-path3 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-tyi-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyi-handler-path3
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-tyi-handler-path3 (*os*))
= 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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                         os)))))
→ 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*)
→ (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*) → (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*)
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                         os)))))
→ 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
good-tm (*os*)
→ (os-obuffers (os-svc-tyi-handler-path3 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-svc-tyi-handler-path3.

THEOREM: os-mbuffers-os-svc-tyi-handler-path3
good-tm (*os*)
→ (os-mbuffers (os-svc-tyi-handler-path3 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-svc-tyi-handler-path3.

THEOREM: os-status-table-os-svc-tyi-handler-path3
good-tm (*os*)
→ (os-status-table (os-svc-tyi-handler-path3 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-svc-tyi-handler-path3.

THEOREM: not-tm-waiting-os-svc-tyi-handler-path3
¬ tm-waiting (os-svc-tyi-handler-path3 (*os*))

EVENT: Disable not-tm-waiting-os-svc-tyi-handler-path3.

THEOREM: tm-base-os-svc-tyi-handler-path3
tm-base (os-svc-tyi-handler-path3 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-svc-tyi-handler-path3.

THEOREM: tm-limit-os-svc-tyi-handler-path3
tm-limit (os-svc-tyi-handler-path3 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-svc-tyi-handler-path3.

THEOREM: tm-base-tm-limit-os-svc-tyi-handler-path3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-svc-tyi-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-svc-tyi-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-svc-tyi-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-svc-tyi-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-tyi-handler-path3.

THEOREM: good-os-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                           os)))))))
→ good-os (os-svc-tyi-handler-path3 (*os*))

EVENT: Disable good-os-os-svc-tyi-handler-path3.

THEOREM: good-os-os-svc-tyi-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '3))))
→ good-os (os-svc-tyi-handler (*os*))

EVENT: Disable good-os-os-svc-tyi-handler.

THEOREM: good-tm-os-intended-svc-tyo-interrupt
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-tm (os-intended-svc-tyo-interrupt (*os*))

EVENT: Disable good-tm-os-intended-svc-tyo-interrupt.

THEOREM: good-tm-os-svc-tyo-handler-path1
(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-qemptyp (array-deq (os-readyq (*os*))))))))
→ good-tm (os-svc-tyo-handler-path1 (*os*))

EVENT: Disable good-tm-os-svc-tyo-handler-path1.

THEOREM: tm-slimit-os-svc-tyo-handler-path1
tm-slimit (os-svc-tyo-handler-path1 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyo-handler-path1.

THEOREM: os-code-os-svc-tyo-handler-path1
good-tm (*os*) → (os-code (os-svc-tyo-handler-path1 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyo-handler-path1.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path1
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-tyo-handler-path1 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path1
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-tyo-handler-path1 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-svc-tyo-handler-path1.

THEOREM: os-segment-table-os-svc-tyo-handler-path1
good-tm (*os*)
→ (os-segment-table (os-svc-tyo-handler-path1 (*os*))
= 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
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ 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
tm-waiting (os-svc-tyo-handler-path1 (*os*))

EVENT: Disable tm-waiting-os-svc-tyo-handler-path1.

THEOREM: good-os-os-svc-tyo-handler-path1
(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-qemptyp (array-deq (os-readyq (*os*))))))))
→ 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
(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-qemptyp (array-deq (os-readyq (*os*))))))))
→ 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
tm-slimit (os-svc-tyo-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyo-handler-path2.

THEOREM: os-code-os-svc-tyo-handler-path2
good-tm (*os*) → (os-code (os-svc-tyo-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyo-handler-path2.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path2
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5, tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-svc-tyo-handler-path2 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path2
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-tyo-handler-path2 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-svc-tyo-handler-path2.

THEOREM: os-segment-table-os-svc-tyo-handler-path2
good-tm (*os*)
→ (os-segment-table (os-svc-tyo-handler-path2 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-svc-tyo-handler-path2.

THEOREM: os-readyq-os-svc-tyo-handler-path2

good-tm (*os*)
→ (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*)
→ (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*)
→ (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*)
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ 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-qemptyp (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
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-svc-tyo-handler-path3 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-svc-tyo-handler-path3.

THEOREM: os-segment-table-os-svc-tyo-handler-path3
good-tm (*os*)
→ (os-segment-table (os-svc-tyo-handler-path3 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-svc-tyo-handler-path3.

THEOREM: os-readyq-os-svc-tyo-handler-path3
good-tm (*os*) → (os-readyq (os-svc-tyo-handler-path3 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-svc-tyo-handler-path3.

THEOREM: os-ibuffers-os-svc-tyo-handler-path3
good-tm (*os*)
→ (os-ibuffers (os-svc-tyo-handler-path3 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-svc-tyo-handler-path3.

THEOREM: os-obuffers-os-svc-tyo-handler-path3
good-tm (*os*)
→ (os-obuffers (os-svc-tyo-handler-path3 (*os*))
= 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
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qfullp (os-current-obuffer (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
(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-os (os-svc-tyo-handler-path3 (*os*))

EVENT: Disable good-os-os-svc-tyo-handler-path3.

THEOREM: good-tm-os-svc-tyo-handler-path4
(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-path4 (*os*))

EVENT: Disable good-tm-os-svc-tyo-handler-path4.

THEOREM: tm-slimit-os-svc-tyo-handler-path4
tm-slimit (os-svc-tyo-handler-path4 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-svc-tyo-handler-path4.

THEOREM: os-code-os-svc-tyo-handler-path4
good-tm (*os*) → (os-code (os-svc-tyo-handler-path4 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-svc-tyo-handler-path4.

THEOREM: tm-interrupt-addresses-os-svc-tyo-handler-path4
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-svc-tyo-handler-path4 (*os*)))
  = getnth ('3, tm-memory (*os*)))

$$
\begin{array}{l}
\wedge \quad ((\text{getnth}\,(\texttt{'4},\ \text{tm-memory}\,(\text{os-svc-tyo-handler-path4}\,(os))) \\
\qquad = \quad \text{getnth}\,(\texttt{'4},\ \text{tm-memory}\,(os))) \\
\qquad \wedge \quad ((\text{getnth}\,(\texttt{'5},\ \text{tm-memory}\,(\text{os-svc-tyo-handler-path4}\,(os))) \\
\qquad\qquad = \quad \text{getnth}\,(\texttt{'5},\ \text{tm-memory}\,(os))) \\
\qquad\qquad \wedge \quad ((\text{getnth}\,(\texttt{'6}, \\
\qquad\qquad\qquad\qquad \text{tm-memory}\,(\text{os-svc-tyo-handler-path4}\,(os))) \\
\qquad\qquad\qquad = \quad \text{getnth}\,(\texttt{'6},\ \text{tm-memory}\,(os))) \\
\qquad\qquad\qquad \wedge \quad ((\text{getnth}\,(\texttt{'7}, \\
\qquad\qquad\qquad\qquad\qquad \text{tm-memory}\,(\text{os-svc-tyo-handler-path4}\,(os))) \\
\qquad\qquad\qquad\qquad = \quad \text{getnth}\,(\texttt{'7},\ \text{tm-memory}\,(os))) \\
\qquad\qquad\qquad\qquad \wedge \quad (\text{getnth}\,(\texttt{'11}, \\
\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-memory}\,(\text{os-svc-tyo-handler-path4}\,(os))) \\
\qquad\qquad\qquad\qquad\qquad = \quad \text{getnth}\,(\texttt{'11}, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-memory}\,(os))))))))
\end{array}
$$

EVENT: Disable tm-interrupt-addresses-os-svc-tyo-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-svc-tyo-handler-path4
$\neg$ tm-in-supervisor-mode (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)
$\rightarrow$ (os-task-table (os-svc-tyo-handler-path4 (os))
= 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)
$\rightarrow$ (os-segment-table (os-svc-tyo-handler-path4 (os))
= 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) $\rightarrow$ (os-readyq (os-svc-tyo-handler-path4 (os)) = os-readyq (os))

EVENT: Disable os-readyq-os-svc-tyo-handler-path4.

THEOREM: os-ibuffers-os-svc-tyo-handler-path4
good-tm (os)
$\rightarrow$ (os-ibuffers (os-svc-tyo-handler-path4 (os)) = 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*)))))

$\wedge$ (tm-limit (os-svc-tyo-handler-path4 (*os*))
= limit (getnth (os-current-taskid (os-svc-tyo-handler-path4 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-svc-tyo-handler-path4.

THEOREM: good-os-os-svc-tyo-handler-path4
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ tm-errorp (*os*))
$\wedge$ (((tm-svcid (*os*) **mod** '4) = '2)
$\wedge$ ((¬ array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os)))
$\wedge$ (¬ tm-oport-idlep (array-qfirst (os-readyq (*os*)),
tm-oports (*os*))))))))
$\rightarrow$ good-os (os-svc-tyo-handler-path4 (*os*))

EVENT: Disable good-os-os-svc-tyo-handler-path4.

THEOREM: good-os-os-svc-tyo-handler
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ tm-errorp (*os*)) $\wedge$ ((tm-svcid (*os*) **mod** '4) = '2))))
$\rightarrow$ good-os (os-svc-tyo-handler (*os*))

EVENT: Disable good-os-os-svc-tyo-handler.

THEOREM: good-tm-os-intended-input-interrupt
(good-os (*os*) $\wedge$ tm-some-input-interruptp (tm-iports (*os*)))
$\rightarrow$ good-tm (os-intended-input-interrupt (*os*))

EVENT: Disable good-tm-os-intended-input-interrupt.

THEOREM: finite-number-queuep-array-enq-interrupting-input-port
(good-os (*os*)
$\wedge$ (tm-waiting (*os*) $\wedge$ tm-some-input-interruptp (tm-iports (*os*))))
$\rightarrow$ finite-number-queuep (array-enq (tm-interrupting-input-port (tm-iports (*os*)),
os-readyq (*os*)),
'16,
'16)

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,

os-ibuffers (os-waiting-input-handler-path1 (*os*))),
'4,
'65536)

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path1.

THEOREM: os-obuffers-os-waiting-input-handler-path1
good-tm (*os*)
→ (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*)
→ (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*)
→ (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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ 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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (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*)
 ∧ (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-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
¬ 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*) ∧ tm-waiting (*os*))
→ ((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*)))))
 ∧ (tm-limit (os-waiting-input-handler-path1 (*os*))

$$= \quad \text{limit}\,(\text{getnth}\,(\text{os-current-taskid}\,(\text{os-waiting-input-handler-path1}\,(os)),\ \text{table}\,(\texttt{'2},\ \text{os-segment-table}\,(os))))))$$

EVENT: Disable tm-base-tm-limit-os-waiting-input-handler-path1.

THEOREM: good-os-os-waiting-input-handler-path1
$$\begin{array}{l}(\text{good-os}\,(os)\\ \quad \wedge \quad (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os))\\ \qquad \wedge \quad (\text{tm-waiting}\,(os)\\ \qquad\quad \wedge \quad ((\text{getnth}\,(\texttt{'2}\\ \qquad\qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),\\ \qquad\qquad\qquad \text{os-status-table}\,(os))\\ \qquad\qquad = \quad \texttt{'5})\\ \qquad\qquad \wedge \quad \text{array-qfullp}\,(\text{os-current-ibuffer}\,(\text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)))))\\ \rightarrow \quad \text{good-os}\,(\text{os-waiting-input-handler-path1}\,(os))\end{array}$$

EVENT: Disable good-os-os-waiting-input-handler-path1.

THEOREM: good-tm-os-waiting-input-handler-path2
$$\begin{array}{l}(\text{good-os}\,(os)\\ \quad \wedge \quad (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os))\\ \qquad \wedge \quad (\text{tm-waiting}\,(os)\\ \qquad\quad \wedge \quad ((\text{getnth}\,(\texttt{'2}\\ \qquad\qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),\\ \qquad\qquad\qquad \text{os-status-table}\,(os))\\ \qquad\qquad = \quad \texttt{'5})\\ \qquad\qquad \wedge \quad ((\neg\ \text{array-qfullp}\,(\text{os-current-ibuffer}\,(\text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)))\\ \qquad\qquad\qquad \wedge \quad \text{tm-iport-errorp}\,(\text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),\\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{tm-iports}\,(os)))))))\\ \rightarrow \quad \text{good-tm}\,(\text{os-waiting-input-handler-path2}\,(os))\end{array}$$

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
$$\begin{array}{l}\text{good-tm}\,(os)\\ \rightarrow \quad (\text{os-code}\,(\text{os-waiting-input-handler-path2}\,(os)) = \text{os-code}\,(os))\end{array}$$

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*)

→ (os-status-table (os-waiting-input-handler-path2 (*os*))
= os-update-status (tm-interrupting-input-port (tm-iports (*os*)),
'0,
'0,
os))

EVENT: Disable os-status-table-os-waiting-input-handler-path2.

THEOREM: good-status-list-os-status-table-os-waiting-input-handler-path2
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ 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
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (os-ready-set (os-waiting-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-waiting-input-handler-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-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))))
→ 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
¬ 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*) ∧ tm-waiting (*os*))
→ ((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*)))))
∧ (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*)
∧ (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-os (os-waiting-input-handler-path2 (*os*))

EVENT: Disable good-os-os-waiting-input-handler-path2.

THEOREM: good-tm-os-waiting-input-handler-path3
(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-path3 (*os*))

EVENT: Disable good-tm-os-waiting-input-handler-path3.

THEOREM: tm-slimit-os-waiting-input-handler-path3
tm-slimit (os-waiting-input-handler-path3 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-input-handler-path3.

THEOREM: os-code-os-waiting-input-handler-path3
good-tm (*os*)
→ (os-code (os-waiting-input-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-input-handler-path3.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path3
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,

tm-memory (os-waiting-input-handler-path3 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path3
¬ 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*)
→ (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*)
→ (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*)
→ (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*)
→ (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*)

```
∧   (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-path3 (os))),
                                  '4,
                                  '65536)
```

EVENT: Disable finite-number-queue-listp-os-ibuffers-os-waiting-input-handler-path3.

THEOREM: os-obuffers-os-waiting-input-handler-path3
good-tm (*os*)
→ (os-obuffers (os-waiting-input-handler-path3 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-input-handler-path3.

THEOREM: os-mbuffers-os-waiting-input-handler-path3
good-tm (*os*)
→ (os-mbuffers (os-waiting-input-handler-path3 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-waiting-input-handler-path3.

THEOREM: os-status-table-os-waiting-input-handler-path3
good-tm (*os*)
→ (os-status-table (os-waiting-input-handler-path3 (*os*))
    = os-update-status (tm-interrupting-input-port (tm-iports (*os*)),
                        '0,
                        '0,
                        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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ good-status-list (table ('2,
                           os-status-table (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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))

```
→  (os-ready-set (os-waiting-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-waiting-input-handler-path3.

```
THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-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-waiting-input-handler-path3 (os)))
```

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-input-handler-path3.

```
THEOREM: not-tm-waiting-os-waiting-input-handler-path3
¬ tm-waiting (os-waiting-input-handler-path3 (os))
```

EVENT: Disable not-tm-waiting-os-waiting-input-handler-path3.

```
THEOREM: tm-base-os-waiting-input-handler-path3
tm-base (os-waiting-input-handler-path3 (os))
=  base (getnth (tm-interrupting-input-port (tm-iports (os)),
                 table ('2, os-segment-table (os))))
```

EVENT: Disable tm-base-os-waiting-input-handler-path3.

```
THEOREM: tm-limit-os-waiting-input-handler-path3
tm-limit (os-waiting-input-handler-path3 (os))
=  limit (getnth (tm-interrupting-input-port (tm-iports (os)),
                  table ('2, os-segment-table (os))))
```

EVENT: Disable tm-limit-os-waiting-input-handler-path3.

THEOREM: tm-base-tm-limit-os-waiting-input-handler-path3

(good-os (*os*) ∧ tm-waiting (*os*))
→ ((tm-base (os-waiting-input-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-waiting-input-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-waiting-input-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-waiting-input-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-waiting-input-handler-path3.

THEOREM: good-os-os-waiting-input-handler-path3
(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-os (os-waiting-input-handler-path3 (*os*))

EVENT: Disable good-os-os-waiting-input-handler-path3.

THEOREM: good-tm-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-tm (os-waiting-input-handler-path4 (*os*))

EVENT: Disable good-tm-os-waiting-input-handler-path4.

THEOREM: tm-slimit-os-waiting-input-handler-path4
tm-slimit (os-waiting-input-handler-path4 (*os*)) = 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

good-tm (*os*)
→ (os-segment-table (os-waiting-input-handler-path5 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-waiting-input-handler-path5.

THEOREM: os-readyq-os-waiting-input-handler-path5
good-tm (*os*)
→ (os-readyq (os-waiting-input-handler-path5 (*os*)) = os-readyq (*os*))

EVENT: Disable os-readyq-os-waiting-input-handler-path5.

THEOREM: os-ibuffers-os-waiting-input-handler-path5
good-tm (*os*)
→ (os-ibuffers (os-waiting-input-handler-path5 (*os*))
= 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
(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-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
good-tm (*os*)
→ (os-obuffers (os-waiting-input-handler-path5 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-waiting-input-handler-path5.

THEOREM: os-mbuffers-os-waiting-input-handler-path5
good-tm (*os*)
→ (os-mbuffers (os-waiting-input-handler-path5 (*os*)) = 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-iport-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-iport-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
tm-slimit (os-waiting-input-handler-path6 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-input-handler-path6.

THEOREM: os-code-os-waiting-input-handler-path6
good-tm (*os*)
→ (os-code (os-waiting-input-handler-path6 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-input-handler-path6.

THEOREM: tm-interrupt-addresses-os-waiting-input-handler-path6
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-waiting-input-handler-path6 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-input-handler-path6.

THEOREM: not-tm-in-supervisor-mode-os-waiting-input-handler-path6
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-waiting-input-handler-path6 (*os*))
= 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*)
→ (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*)
→ (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*)
∧ (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-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*)
∧ (tm-some-input-interruptp (tm-iports (*os*)) ∧ tm-waiting (*os*)))
→ 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*)

```
∧   ((¬ tm-waiting (os))
     ∧   (tm-some-input-interruptp (tm-iports (os))
          ∧   (getnth ('2
                       ∗   tm-interrupting-input-port (tm-iports (os)),
                       os-status-table (os))
               =   '5))))
→   finite-number-queuep (array-enq (tm-interrupting-input-port (tm-iports (os)),
                                     os-readyq (os)),
                         '16,
                         '16)
```

EVENT: Disable finite-number-queuep-array-enq-interrupting-input-port2.

THEOREM: good-tm-os-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))))))
→   good-tm (os-running-input-handler-path1 (os))
```

EVENT: Disable good-tm-os-running-input-handler-path1.

THEOREM: tm-slimit-os-running-input-handler-path1

```
tm-slimit (os-running-input-handler-path1 (os)) = tm-slimit (os)
```

EVENT: Disable tm-slimit-os-running-input-handler-path1.

THEOREM: os-code-os-running-input-handler-path1

```
good-tm (os)
→   (os-code (os-running-input-handler-path1 (os)) = os-code (os))
```

EVENT: Disable os-code-os-running-input-handler-path1.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path1

```
good-tm (os)
→   ((getnth ('3, tm-memory (os-running-input-handler-path1 (os)))
      =   getnth ('3, tm-memory (os)))
     ∧   ((getnth ('4, tm-memory (os-running-input-handler-path1 (os)))
```

```
    =   getnth('4, tm-memory(os)))
∧   ((getnth('5,
                tm-memory(os-running-input-handler-path1(os)))
      =   getnth('5, tm-memory(os)))
     ∧   ((getnth('6,
                   tm-memory(os-running-input-handler-path1(os)))
           =   getnth('6, tm-memory(os)))
          ∧   ((getnth('7,
                        tm-memory(os-running-input-handler-path1(os)))
                =   getnth('7, tm-memory(os)))
               ∧   (getnth('11,
                            tm-memory(os-running-input-handler-path1(os)))
                    =   getnth('11,
                               tm-memory(os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path1
¬ tm-in-supervisor-mode(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*)
→ (os-task-table(os-running-input-handler-path1(*os*))
= 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*)
→ (os-segment-table(os-running-input-handler-path1(*os*))
= 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*)
→ (os-readyq(os-running-input-handler-path1(*os*))
= array-enq(tm-interrupting-input-port(tm-iports(*os*)),
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

(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ good-status-list (table ('2,
os-status-table (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
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (os-ready-set (os-running-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-running-input-handler-path1.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-input-handler-path1
(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-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
¬ tm-waiting (os-running-input-handler-path1 (*os*))

EVENT: Disable not-tm-waiting-os-running-input-handler-path1.

THEOREM: tm-base-os-running-input-handler-path1
tm-base (os-running-input-handler-path1 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-input-handler-path1.

THEOREM: tm-limit-os-running-input-handler-path1
tm-limit (os-running-input-handler-path1 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-input-handler-path1.

THEOREM: tm-base-tm-limit-os-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))))
→ ((tm-base (os-running-input-handler-path1 (*os*))
  = base (getnth (os-current-taskid (os-running-input-handler-path1 (*os*)),
    table ('2, os-segment-table (*os*)))))
  ∧ (tm-limit (os-running-input-handler-path1 (*os*))
   = limit (getnth (os-current-taskid (os-running-input-handler-path1 (*os*)),
     table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path1.

THEOREM: good-os-os-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))))))
→ good-os (os-running-input-handler-path1 (*os*))

EVENT: Disable good-os-os-running-input-handler-path1.

THEOREM: good-tm-os-running-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*)),
    os-status-table (*os*))
    = '5)
   ∧ ((¬ 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-tm (os-running-input-handler-path2 (os))

EVENT: Disable good-tm-os-running-input-handler-path2.

THEOREM: tm-slimit-os-running-input-handler-path2
tm-slimit (os-running-input-handler-path2 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-running-input-handler-path2.

THEOREM: os-code-os-running-input-handler-path2
good-tm (os)
$\rightarrow$ (os-code (os-running-input-handler-path2 (os)) = os-code (os))

EVENT: Disable os-code-os-running-input-handler-path2.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path2
good-tm (os)
$\rightarrow$ ((getnth ('3, tm-memory (os-running-input-handler-path2 (os)))
= getnth ('3, tm-memory (os)))
$\wedge$ ((getnth ('4, tm-memory (os-running-input-handler-path2 (os)))
= getnth ('4, tm-memory (os)))
$\wedge$ ((getnth ('5,
tm-memory (os-running-input-handler-path2 (os)))
= getnth ('5, tm-memory (os)))
$\wedge$ ((getnth ('6,
tm-memory (os-running-input-handler-path2 (os)))
= getnth ('6, tm-memory (os)))
$\wedge$ ((getnth ('7,
tm-memory (os-running-input-handler-path2 (os)))
= getnth ('7, tm-memory (os)))
$\wedge$ (getnth ('11,
tm-memory (os-running-input-handler-path2 (os)))
= getnth ('11,
tm-memory (os))))))))

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path2
$\neg$ tm-in-supervisor-mode (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*)
→ (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*)
→ (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*)
→ (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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ 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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))

$$\begin{array}{l}
\quad \wedge \quad (\text{getnth}\,(\texttt{'2} \\
\qquad\qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
\qquad\qquad\qquad \text{os-status-table}\,(os)) \\
\qquad\qquad = \quad \texttt{'5})))) \\
\rightarrow \quad \text{permutation}\,(\text{mapup-queue}\,(\text{array-enq}\,(\text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-readyq}\,(os))), \\
\qquad\qquad\qquad\qquad\qquad \text{os-ready-set}\,(\text{os-running-input-handler-path2}\,(os)))
\end{array}$$

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-running-input-handler-path2.

THEOREM: not-tm-waiting-os-running-input-handler-path2
$\neg\ \text{tm-waiting}\,(\text{os-running-input-handler-path2}\,(os))$

EVENT: Disable not-tm-waiting-os-running-input-handler-path2.

THEOREM: tm-base-os-running-input-handler-path2
$\text{tm-base}\,(\text{os-running-input-handler-path2}\,(os)) = \text{tm-base}\,(os)$

EVENT: Disable tm-base-os-running-input-handler-path2.

THEOREM: tm-limit-os-running-input-handler-path2
$\text{tm-limit}\,(\text{os-running-input-handler-path2}\,(os)) = \text{tm-limit}\,(os)$

EVENT: Disable tm-limit-os-running-input-handler-path2.

THEOREM: tm-base-tm-limit-os-running-input-handler-path2

$$\begin{array}{l}
(\text{good-os}\,(os) \\
\quad \wedge \quad (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os)) \\
\qquad \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
\qquad\qquad \wedge \quad (\text{getnth}\,(\texttt{'2} \\
\qquad\qquad\qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
\qquad\qquad\qquad\qquad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad = \quad \texttt{'5})))) \\
\rightarrow \quad ((\text{tm-base}\,(\text{os-running-input-handler-path2}\,(os)) \\
\qquad\quad = \quad \text{base}\,(\text{getnth}\,(\text{os-current-taskid}\,(\text{os-running-input-handler-path2}\,(os)), \\
\qquad\qquad\qquad\qquad\quad \text{table}\,(\texttt{'2}, \text{os-segment-table}\,(os))))) \\
\qquad \wedge \quad (\text{tm-limit}\,(\text{os-running-input-handler-path2}\,(os)) \\
\qquad\qquad = \quad \text{limit}\,(\text{getnth}\,(\text{os-current-taskid}\,(\text{os-running-input-handler-path2}\,(os)), \\
\qquad\qquad\qquad\qquad\qquad\quad \text{table}\,(\texttt{'2}, \text{os-segment-table}\,(os))))))
\end{array}$$

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path2.

THEOREM: good-os-os-running-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*)),
    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-os (os-running-input-handler-path2 (*os*))

EVENT: Disable good-os-os-running-input-handler-path2.

THEOREM: good-tm-os-running-input-handler-path3
(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-running-input-handler-path3 (*os*))

EVENT: Disable good-tm-os-running-input-handler-path3.

THEOREM: tm-slimit-os-running-input-handler-path3
tm-slimit (os-running-input-handler-path3 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-running-input-handler-path3.

THEOREM: os-code-os-running-input-handler-path3
good-tm (*os*)
→ (os-code (os-running-input-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-input-handler-path3.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path3
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-running-input-handler-path3 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path3
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-running-input-handler-path3 (*os*))
= os-new-task-table (*os*))

EVENT: Disable os-task-table-os-running-input-handler-path3.

THEOREM: os-segment-table-os-running-input-handler-path3
good-tm (*os*)
→ (os-segment-table (os-running-input-handler-path3 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-running-input-handler-path3.

THEOREM: os-readyq-os-running-input-handler-path3
good-tm (*os*)

→ (os-readyq (os-running-input-handler-path3 (*os*))
= array-enq (tm-interrupting-input-port (tm-iports (*os*)),
os-readyq (*os*)))

EVENT: Disable os-readyq-os-running-input-handler-path3.

THEOREM: os-ibuffers-os-running-input-handler-path3
good-tm (*os*)
→ (os-ibuffers (os-running-input-handler-path3 (*os*))
= 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*)
∧ (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-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*)
→ (os-obuffers (os-running-input-handler-path3 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path3.

THEOREM: os-mbuffers-os-running-input-handler-path3
good-tm (*os*)
→ (os-mbuffers (os-running-input-handler-path3 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path3.

THEOREM: os-status-table-os-running-input-handler-path3
good-tm (*os*)
→ (os-status-table (os-running-input-handler-path3 (*os*))
= os-update-status (tm-interrupting-input-port (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
tm-base (os-running-input-handler-path3 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-input-handler-path3.

THEOREM: tm-limit-os-running-input-handler-path3
tm-limit (os-running-input-handler-path3 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-input-handler-path3.

THEOREM: tm-base-tm-limit-os-running-input-handler-path3
(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-base (os-running-input-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-running-input-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-running-input-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-running-input-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path3.

THEOREM: good-os-os-running-input-handler-path3
(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-os (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*))))))

$\wedge$ (tm-limit (os-running-input-handler-path4 (os))
= limit (getnth (os-current-taskid (os-running-input-handler-path4 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path4.

THEOREM: good-os-os-running-input-handler-path4
(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)
$\wedge$ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))))
$\rightarrow$ good-os (os-running-input-handler-path4 (os))

EVENT: Disable good-os-os-running-input-handler-path4.

THEOREM: good-tm-os-running-input-handler-path5
(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)
$\wedge$ ((¬ 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-tm (os-running-input-handler-path5 (os))

EVENT: Disable good-tm-os-running-input-handler-path5.

THEOREM: tm-slimit-os-running-input-handler-path5
tm-slimit (os-running-input-handler-path5 (os)) = tm-slimit (os)

EVENT: Disable tm-slimit-os-running-input-handler-path5.

THEOREM: os-code-os-running-input-handler-path5
good-tm (os)
$\rightarrow$ (os-code (os-running-input-handler-path5 (os)) = os-code (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
¬ tm-waiting (os-running-input-handler-path5 (*os*))

EVENT: Disable not-tm-waiting-os-running-input-handler-path5.

THEOREM: tm-base-os-running-input-handler-path5
tm-base (os-running-input-handler-path5 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-input-handler-path5.

THEOREM: tm-limit-os-running-input-handler-path5
tm-limit (os-running-input-handler-path5 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-input-handler-path5.

THEOREM: tm-base-tm-limit-os-running-input-handler-path5
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-running-input-handler-path5 (*os*))
= base (getnth (os-current-taskid (os-running-input-handler-path5 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-running-input-handler-path5 (*os*))
= limit (getnth (os-current-taskid (os-running-input-handler-path5 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-input-handler-path5.

THEOREM: good-os-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*)))))))
→ good-os (os-running-input-handler-path5 (*os*))

EVENT: Disable good-os-os-running-input-handler-path5.

THEOREM: good-tm-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*))))))))
→ good-tm (os-running-input-handler-path6 (*os*))

EVENT: Disable good-tm-os-running-input-handler-path6.

THEOREM: tm-slimit-os-running-input-handler-path6
tm-slimit (os-running-input-handler-path6 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-running-input-handler-path6.

THEOREM: os-code-os-running-input-handler-path6
good-tm (*os*)
→ (os-code (os-running-input-handler-path6 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-input-handler-path6.

THEOREM: tm-interrupt-addresses-os-running-input-handler-path6
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-input-handler-path6 (*os*)))
   = getnth ('3, tm-memory (*os*)))
  ∧ ((getnth ('4, tm-memory (os-running-input-handler-path6 (*os*)))
     = getnth ('4, tm-memory (*os*)))
    ∧ ((getnth ('5,
                tm-memory (os-running-input-handler-path6 (*os*)))
       = getnth ('5, tm-memory (*os*)))
      ∧ ((getnth ('6,
                  tm-memory (os-running-input-handler-path6 (*os*)))
         = getnth ('6, tm-memory (*os*)))
        ∧ ((getnth ('7,
                    tm-memory (os-running-input-handler-path6 (*os*)))
           = getnth ('7, tm-memory (*os*)))
          ∧ (getnth ('11,

tm-memory (os-running-input-handler-path6 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-running-input-handler-path6.

THEOREM: not-tm-in-supervisor-mode-os-running-input-handler-path6
¬ 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*)
→ (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*)
→ (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*)
→ (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*)
→ (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*)
∧ (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-path6 (*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*)
→ (os-obuffers (os-running-input-handler-path6 (*os*)) = os-obuffers (*os*))

EVENT: Disable os-obuffers-os-running-input-handler-path6.

THEOREM: os-mbuffers-os-running-input-handler-path6
good-tm (*os*)
→ (os-mbuffers (os-running-input-handler-path6 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-input-handler-path6.

THEOREM: os-status-table-os-running-input-handler-path6
good-tm (*os*)
→ (os-status-table (os-running-input-handler-path6 (*os*))
= os-status-table (*os*))

EVENT: Disable os-status-table-os-running-input-handler-path6.

THEOREM: not-tm-waiting-os-running-input-handler-path6
¬ tm-waiting (os-running-input-handler-path6 (*os*))

EVENT: Disable not-tm-waiting-os-running-input-handler-path6.

THEOREM: tm-base-os-running-input-handler-path6
tm-base (os-running-input-handler-path6 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-input-handler-path6.

THEOREM: tm-limit-os-running-input-handler-path6
tm-limit (os-running-input-handler-path6 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-input-handler-path6.

THEOREM: tm-base-tm-limit-os-running-input-handler-path6
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((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*)))))
∧ (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*)
∧ (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-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*)
∧ (tm-some-input-interruptp (tm-iports (*os*)) ∧ (¬ tm-waiting (*os*))))
→ 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*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ 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*)
∧ (tm-waiting (*os*) ∧ 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*)))

```
    =   getnth ('6, tm-memory (os)))
∧   ((getnth ('7,
                tm-memory (os-waiting-output-handler-path1 (os)))
      =   getnth ('7, tm-memory (os)))
     ∧   (getnth ('11,
                   tm-memory (os-waiting-output-handler-path1 (os)))
          =   getnth ('11,
                       tm-memory (os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path1.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path1
¬ tm-in-supervisor-mode (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*)
→ (os-task-table (os-waiting-output-handler-path1 (*os*))
    = 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*)
→ (os-segment-table (os-waiting-output-handler-path1 (*os*))
    = 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*)
→ (os-readyq (os-waiting-output-handler-path1 (*os*))
    = array-enq (tm-interrupting-output-port (tm-oports (*os*)),
                 os-readyq (*os*)))

EVENT: Disable os-readyq-os-waiting-output-handler-path1.

THEOREM: os-ibuffers-os-waiting-output-handler-path1
good-tm (*os*)
→ (os-ibuffers (os-waiting-output-handler-path1 (*os*)) = os-ibuffers (*os*))

EVENT: Disable os-ibuffers-os-waiting-output-handler-path1.

THEOREM: os-obuffers-os-waiting-output-handler-path1
good-tm (*os*)
→ (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*)
→ (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*)
→ (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*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ 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*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (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*)
∧ (tm-waiting (*os*)

$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (getnth ('2
$*$ tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
= '4))))
$\rightarrow$ permutation (mapup-queue (array-enq (tm-interrupting-output-port (tm-oports (os)),
os-readyq (os))),
os-ready-set (os-waiting-output-handler-path1 (os)))

EVENT: Disable permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path1.

THEOREM: not-tm-waiting-os-waiting-output-handler-path1
$\neg$ tm-waiting (os-waiting-output-handler-path1 (os))

EVENT: Disable not-tm-waiting-os-waiting-output-handler-path1.

THEOREM: tm-base-os-waiting-output-handler-path1
tm-base (os-waiting-output-handler-path1 (os))
= base (getnth (tm-interrupting-output-port (tm-oports (os)),
table ('2, os-segment-table (os))))

EVENT: Disable tm-base-os-waiting-output-handler-path1.

THEOREM: tm-limit-os-waiting-output-handler-path1
tm-limit (os-waiting-output-handler-path1 (os))
= limit (getnth (tm-interrupting-output-port (tm-oports (os)),
table ('2, os-segment-table (os))))

EVENT: Disable tm-limit-os-waiting-output-handler-path1.

THEOREM: tm-base-tm-limit-os-waiting-output-handler-path1
(good-os (os) $\wedge$ tm-waiting (os))
$\rightarrow$ ((tm-base (os-waiting-output-handler-path1 (os))
= base (getnth (os-current-taskid (os-waiting-output-handler-path1 (os)),
table ('2, os-segment-table (os)))))
$\wedge$ (tm-limit (os-waiting-output-handler-path1 (os))
= limit (getnth (os-current-taskid (os-waiting-output-handler-path1 (os)),
table ('2, os-segment-table (os))))))

EVENT: Disable tm-base-tm-limit-os-waiting-output-handler-path1.

THEOREM: good-os-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-os (os-waiting-output-handler-path1 (*os*))

EVENT: Disable good-os-os-waiting-output-handler-path1.

THEOREM: good-tm-os-waiting-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)))))))
→ good-tm (os-waiting-output-handler-path2 (*os*))

EVENT: Disable good-tm-os-waiting-output-handler-path2.

THEOREM: tm-slimit-os-waiting-output-handler-path2
tm-slimit (os-waiting-output-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-output-handler-path2.

THEOREM: os-code-os-waiting-output-handler-path2
good-tm (*os*)
→ (os-code (os-waiting-output-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-output-handler-path2.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path2
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-output-handler-path2 (*os*)))
  = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-waiting-output-handler-path2 (*os*)))

```
    =   getnth('4, tm-memory(os)))
∧   ((getnth('5,
              tm-memory(os-waiting-output-handler-path2(os)))
      =   getnth('5, tm-memory(os)))
     ∧   ((getnth('6,
                   tm-memory(os-waiting-output-handler-path2(os)))
           =   getnth('6, tm-memory(os)))
          ∧   ((getnth('7,
                        tm-memory(os-waiting-output-handler-path2(os)))
                =   getnth('7, tm-memory(os)))
               ∧   (getnth('11,
                           tm-memory(os-waiting-output-handler-path2(os)))
                    =   getnth('11,
                               tm-memory(os))))))))
```

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path2
¬ tm-in-supervisor-mode(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*)
→ (os-task-table(os-waiting-output-handler-path2(*os*))
= 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*)
→ (os-segment-table(os-waiting-output-handler-path2(*os*))
= 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*)
→ (os-readyq(os-waiting-output-handler-path2(*os*))
= array-enq(tm-interrupting-output-port(tm-oports(*os*)),
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

(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ good-status-list (table ('2,
os-status-table (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
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (os-ready-set (os-waiting-output-handler-path2 (*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-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-waiting-output-handler-path2
(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-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
¬ tm-waiting (os-waiting-output-handler-path2 (*os*))

EVENT: Disable not-tm-waiting-os-waiting-output-handler-path2.

THEOREM: tm-base-os-waiting-output-handler-path2
tm-base (os-waiting-output-handler-path2 (*os*))
= base (getnth (tm-interrupting-output-port (tm-oports (*os*)),
table ('2, os-segment-table (*os*))))

EVENT: Disable tm-base-os-waiting-output-handler-path2.

THEOREM: tm-limit-os-waiting-output-handler-path2
tm-limit (os-waiting-output-handler-path2 (*os*))
= limit (getnth (tm-interrupting-output-port (tm-oports (*os*)),
table ('2, os-segment-table (*os*))))

EVENT: Disable tm-limit-os-waiting-output-handler-path2.

THEOREM: tm-base-tm-limit-os-waiting-output-handler-path2
(good-os (*os*) ∧ tm-waiting (*os*))
→ ((tm-base (os-waiting-output-handler-path2 (*os*))
= base (getnth (os-current-taskid (os-waiting-output-handler-path2 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-waiting-output-handler-path2 (*os*))
= limit (getnth (os-current-taskid (os-waiting-output-handler-path2 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-waiting-output-handler-path2.

THEOREM: good-os-os-waiting-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)))))))
→ good-os (os-waiting-output-handler-path2 (*os*))

EVENT: Disable good-os-os-waiting-output-handler-path2.

THEOREM: good-tm-os-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))))))
→ good-tm (os-waiting-output-handler-path3 (*os*))

EVENT: Disable good-tm-os-waiting-output-handler-path3.

THEOREM: tm-slimit-os-waiting-output-handler-path3
tm-slimit (os-waiting-output-handler-path3 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-waiting-output-handler-path3.

THEOREM: os-code-os-waiting-output-handler-path3
good-tm (*os*)
→ (os-code (os-waiting-output-handler-path3 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-waiting-output-handler-path3.

THEOREM: tm-interrupt-addresses-os-waiting-output-handler-path3
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-waiting-output-handler-path3 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path3.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path3
¬ tm-in-supervisor-mode (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
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*)
 ∧ (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-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*)
 ∧ (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-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*)
→ (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
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-waiting-output-handler-path4 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-waiting-output-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-waiting-output-handler-path4
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-waiting-output-handler-path4 (*os*))
= os-task-table (*os*))

EVENT: Disable os-task-table-os-waiting-output-handler-path4.

THEOREM: os-segment-table-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-segment-table (os-waiting-output-handler-path4 (*os*))
= os-segment-table (*os*))

EVENT: Disable os-segment-table-os-waiting-output-handler-path4.

THEOREM: os-readyq-os-waiting-output-handler-path4
good-tm (*os*)
→ (os-readyq (os-waiting-output-handler-path4 (*os*)) = os-readyq (*os*))

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-qemptyp (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
(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-os (os-waiting-output-handler-path4 (*os*))

EVENT: Disable good-os-os-waiting-output-handler-path4.

THEOREM: good-os-os-waiting-output-handler
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ tm-waiting (*os*)))
→ good-os (os-waiting-output-handler (*os*))

EVENT: Disable good-os-os-waiting-output-handler.

THEOREM: finite-number-queuep-array-enq-interrupting-output-port2
(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))))
→ 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-port2.

THEOREM: good-tm-os-running-output-handler-path1
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
  ∧ ((¬ tm-waiting (*os*))

$\wedge$ ((getnth ('2
* tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
= '4)
$\wedge$ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
os))))))
$\rightarrow$ 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)
$\rightarrow$ (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)
$\rightarrow$ ((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
$\neg$ 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)
$\rightarrow$ (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)
$\rightarrow$ (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)
$\rightarrow$ (os-readyq (os-running-output-handler-path1 (os))
= array-enq (tm-interrupting-output-port (tm-oports (os)),
os-readyq (os)))

EVENT: Disable os-readyq-os-running-output-handler-path1.

THEOREM: os-ibuffers-os-running-output-handler-path1
good-tm (os)
$\rightarrow$ (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)
$\rightarrow$ (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)
$\rightarrow$ (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
$\neg$ tm-waiting (os-running-output-handler-path1 (os))

EVENT: Disable not-tm-waiting-os-running-output-handler-path1.

THEOREM: tm-base-os-running-output-handler-path1
tm-base (os-running-output-handler-path1 (os)) = tm-base (os)

EVENT: Disable tm-base-os-running-output-handler-path1.

THEOREM: tm-limit-os-running-output-handler-path1
tm-limit (os-running-output-handler-path1 (os)) = tm-limit (os)

EVENT: Disable tm-limit-os-running-output-handler-path1.

THEOREM: tm-base-tm-limit-os-running-output-handler-path1
(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))
   = '4))))
$\rightarrow$ ((tm-base (os-running-output-handler-path1 (os))
  = base (getnth (os-current-taskid (os-running-output-handler-path1 (os)),
   table ('2, os-segment-table (os)))))
  $\wedge$ (tm-limit (os-running-output-handler-path1 (os))
   = limit (getnth (os-current-taskid (os-running-output-handler-path1 (os)),
    table ('2, os-segment-table (os)))))))

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path1.

THEOREM: good-os-os-running-output-handler-path1
(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))
   = '4)
   $\wedge$ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
    os))))))
$\rightarrow$ good-os (os-running-output-handler-path1 (os))

EVENT: Disable good-os-os-running-output-handler-path1.

THEOREM: good-tm-os-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)))))))
→ good-tm (os-running-output-handler-path2 (*os*))

EVENT: Disable good-tm-os-running-output-handler-path2.

THEOREM: tm-slimit-os-running-output-handler-path2
tm-slimit (os-running-output-handler-path2 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-running-output-handler-path2.

THEOREM: os-code-os-running-output-handler-path2
good-tm (*os*)
→ (os-code (os-running-output-handler-path2 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-output-handler-path2.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path2
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-output-handler-path2 (*os*)))
  = getnth ('3, tm-memory (*os*)))
 ∧ ((getnth ('4, tm-memory (os-running-output-handler-path2 (*os*)))
  = getnth ('4, tm-memory (*os*)))
  ∧ ((getnth ('5,
   tm-memory (os-running-output-handler-path2 (*os*)))
   = getnth ('5, tm-memory (*os*)))
   ∧ ((getnth ('6,
    tm-memory (os-running-output-handler-path2 (*os*)))
    = getnth ('6, tm-memory (*os*)))
    ∧ ((getnth ('7,
     tm-memory (os-running-output-handler-path2 (*os*)))
     = getnth ('7, tm-memory (*os*)))

$$\begin{array}{lll} & \wedge & (\text{getnth}\,(\texttt{'11},\\ & & \qquad \text{tm-memory}\,(\text{os-running-output-handler-path2}\,(os)))\\ & & = \quad \text{getnth}\,(\texttt{'11},\\ & & \qquad\quad \text{tm-memory}\,(os))))))))\end{array}$$

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path2.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path2
$\neg$ tm-in-supervisor-mode (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)
$\rightarrow$ (os-task-table (os-running-output-handler-path2 (os))
= 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)
$\rightarrow$ (os-segment-table (os-running-output-handler-path2 (os))
= 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)
$\rightarrow$ (os-readyq (os-running-output-handler-path2 (os))
= array-enq (tm-interrupting-output-port (tm-oports (os)),
os-readyq (os)))

EVENT: Disable os-readyq-os-running-output-handler-path2.

THEOREM: os-ibuffers-os-running-output-handler-path2
good-tm (os)
$\rightarrow$ (os-ibuffers (os-running-output-handler-path2 (os)) = os-ibuffers (os))

EVENT: Disable os-ibuffers-os-running-output-handler-path2.

THEOREM: os-obuffers-os-running-output-handler-path2
good-tm (os)
$\rightarrow$ (os-obuffers (os-running-output-handler-path2 (os))
= 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
(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-running-output-handler-path2 (*os*))),
                                   '4,
                                   '65536)

EVENT: Disable finite-number-queue-listp-os-obuffers-os-running-output-handler-path2.

THEOREM: os-mbuffers-os-running-output-handler-path2
good-tm (*os*)
→ (os-mbuffers (os-running-output-handler-path2 (*os*)) = os-mbuffers (*os*))

EVENT: Disable os-mbuffers-os-running-output-handler-path2.

THEOREM: os-status-table-os-running-output-handler-path2
good-tm (*os*)
→ (os-status-table (os-running-output-handler-path2 (*os*))
  = os-update-status (tm-interrupting-output-port (tm-oports (*os*)),
                      '0,
                      '0,
                      os))

EVENT: Disable os-status-table-os-running-output-handler-path2.

THEOREM: good-status-list-os-status-table-os-running-output-handler-path2
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ good-status-list (table ('2,
                           os-status-table (os-running-output-handler-path2 (*os*))))

EVENT: Disable good-status-list-os-status-table-os-running-output-handler-path2.

THEOREM: os-ready-set-os-running-output-handler-path2
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (os-ready-set (os-running-output-handler-path2 (*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-path2.

THEOREM: permutation-os-readyq-os-ready-set-for-os-running-output-handler-path2
(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-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
¬ tm-waiting (os-running-output-handler-path2 (*os*))

EVENT: Disable not-tm-waiting-os-running-output-handler-path2.

THEOREM: tm-base-os-running-output-handler-path2
tm-base (os-running-output-handler-path2 (*os*)) = tm-base (*os*)

EVENT: Disable tm-base-os-running-output-handler-path2.

THEOREM: tm-limit-os-running-output-handler-path2
tm-limit (os-running-output-handler-path2 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-output-handler-path2.

THEOREM: tm-base-tm-limit-os-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))))
→ ((tm-base (os-running-output-handler-path2 (*os*))
= base (getnth (os-current-taskid (os-running-output-handler-path2 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-running-output-handler-path2 (*os*))
= limit (getnth (os-current-taskid (os-running-output-handler-path2 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path2.

THEOREM: good-os-os-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)))))))
→ good-os (os-running-output-handler-path2 (*os*))

EVENT: Disable good-os-os-running-output-handler-path2.

THEOREM: good-tm-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))))))
→ good-tm (os-running-output-handler-path3 (*os*))

EVENT: Disable good-tm-os-running-output-handler-path3.

THEOREM: tm-slimit-os-running-output-handler-path3
tm-slimit (os-running-output-handler-path3 (*os*)) = 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
tm-limit (os-running-output-handler-path3 (*os*)) = tm-limit (*os*)

EVENT: Disable tm-limit-os-running-output-handler-path3.

THEOREM: tm-base-tm-limit-os-running-output-handler-path3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ ((tm-base (os-running-output-handler-path3 (*os*))
= base (getnth (os-current-taskid (os-running-output-handler-path3 (*os*)),
table ('2, os-segment-table (*os*)))))
∧ (tm-limit (os-running-output-handler-path3 (*os*))
= limit (getnth (os-current-taskid (os-running-output-handler-path3 (*os*)),
table ('2, os-segment-table (*os*))))))

EVENT: Disable tm-base-tm-limit-os-running-output-handler-path3.

THEOREM: good-os-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))))))
→ good-os (os-running-output-handler-path3 (*os*))

EVENT: Disable good-os-os-running-output-handler-path3.

THEOREM: good-tm-os-running-output-handler-path4
(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-running-output-handler-path4 (*os*))

EVENT: Disable good-tm-os-running-output-handler-path4.

THEOREM: tm-slimit-os-running-output-handler-path4
tm-slimit (os-running-output-handler-path4 (*os*)) = tm-slimit (*os*)

EVENT: Disable tm-slimit-os-running-output-handler-path4.

THEOREM: os-code-os-running-output-handler-path4
good-tm (*os*)
→ (os-code (os-running-output-handler-path4 (*os*)) = os-code (*os*))

EVENT: Disable os-code-os-running-output-handler-path4.

THEOREM: tm-interrupt-addresses-os-running-output-handler-path4
good-tm (*os*)
→ ((getnth ('3, tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('3, tm-memory (*os*)))
∧ ((getnth ('4, tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('4, tm-memory (*os*)))
∧ ((getnth ('5,
tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('5, tm-memory (*os*)))
∧ ((getnth ('6,
tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('6, tm-memory (*os*)))
∧ ((getnth ('7,
tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('7, tm-memory (*os*)))
∧ (getnth ('11,
tm-memory (os-running-output-handler-path4 (*os*)))
= getnth ('11,
tm-memory (*os*))))))))

EVENT: Disable tm-interrupt-addresses-os-running-output-handler-path4.

THEOREM: not-tm-in-supervisor-mode-os-running-output-handler-path4
¬ tm-in-supervisor-mode (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
good-tm (*os*)
→ (os-task-table (os-running-output-handler-path4 (*os*))
= 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-qemptyp (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*)
→ (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*)
→ (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
¬ 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*) ∧ (¬ tm-waiting (*os*)))
→ ((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*)))))
∧ (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*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((¬ 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-os (os-running-output-handler-path4 (os))

EVENT: Disable good-os-os-running-output-handler-path4.

THEOREM: good-os-os-running-output-handler
(good-os (os)
$\wedge$ (tm-some-output-interruptp (tm-oports (os)) $\wedge$ ($\neg$ tm-waiting (os))))
$\rightarrow$ good-os (os-running-output-handler (os))

EVENT: Disable good-os-os-running-output-handler.

DEFINITION:
os-svc-handler (os)
= **if** (tm-svcid (os) **mod** '4) = '0 **then** os-svc-send-handler (os)
elseif (tm-svcid (os) **mod** '4) = '1 **then** os-svc-receive-handler (os)
elseif (tm-svcid (os) **mod** '4) = '2 **then** os-svc-tyo-handler (os)
else os-svc-tyi-handler (os) **endif**

DEFINITION:
os-time-for-svc-handler (os)
= **if** (tm-svcid (os) **mod** '4) = '0
then os-time-for-svc-send-handler (os)
elseif (tm-svcid (os) **mod** '4) = '1
then os-time-for-svc-receive-handler (os)
elseif (tm-svcid (os) **mod** '4) = '2
then os-time-for-svc-tyo-handler (os)
else os-time-for-svc-tyi-handler (os) **endif**

DEFINITION:
os-intended-svc-interrupt (os)
= **if** (tm-svcid (os) **mod** '4) = '0
then os-intended-svc-send-interrupt (os)
elseif (tm-svcid (os) **mod** '4) = '1
then os-intended-svc-receive-interrupt (os)
elseif (tm-svcid (os) **mod** '4) = '2
then os-intended-svc-tyo-interrupt (os)
else os-intended-svc-tyi-interrupt (os) **endif**

THEOREM: equivalence-of-svc-interrupt-expressions
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (tm-execute-svc-interrupt (*os*) = os-intended-svc-interrupt (*os*))

EVENT: Disable equivalence-of-svc-interrupt-expressions.

THEOREM: lessp-4-crock
((*a* ∈ **N**)
∧ ((*a* < '4) ∧ ((*a* ≠ '0) ∧ ((*a* ≠ '1) ∧ (*a* ≠ '2)))))
→ (*a* = '3)

THEOREM: remainder-4-equals-3
(((*n* **mod** '4) ≠ '0) ∧ (((*n* **mod** '4) ≠ '1) ∧ ((*n* **mod** '4) ≠ '2)))
→ ((*n* **mod** '4) = '3)

EVENT: Disable remainder-4-equals-3.

THEOREM: trace-svc-interrupt-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (tm-processor (os-intended-svc-interrupt (*os*),
os-time-for-svc-handler (*os*))
= os-svc-handler (*os*))

EVENT: Disable trace-svc-interrupt-handler.

THEOREM: good-os-os-svc-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ good-os (os-svc-handler (*os*))

EVENT: Disable good-os-os-svc-handler.

DEFINITION:
os-input-interrupt-handler (*os*)
= **if** tm-waiting (*os*) **then** os-waiting-input-handler (*os*)
else os-running-input-handler (*os*) **endif**

DEFINITION:
os-time-for-input-interrupt-handler (*os*)
= **if** tm-waiting (*os*) **then** os-time-for-waiting-input-handler (*os*)
else os-time-for-running-input-handler (*os*) **endif**

THEOREM: trace-input-interrupt-handler
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))

→ (tm-processor (os-intended-input-interrupt (*os*),
os-time-for-input-interrupt-handler (*os*))
= os-input-interrupt-handler (*os*))

EVENT: Disable trace-input-interrupt-handler.

THEOREM: good-os-os-input-interrupt-handler
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ good-os (os-input-interrupt-handler (*os*))

EVENT: Disable good-os-os-input-interrupt-handler.

DEFINITION:
os-output-interrupt-handler (*os*)
= **if** tm-waiting (*os*) **then** os-waiting-output-handler (*os*)
else os-running-output-handler (*os*) **endif**

DEFINITION:
os-time-for-output-interrupt-handler (*os*)
= **if** tm-waiting (*os*) **then** os-time-for-waiting-output-handler (*os*)
else os-time-for-running-output-handler (*os*) **endif**

THEOREM: trace-output-interrupt-handler
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (tm-processor (os-intended-output-interrupt (*os*),
os-time-for-output-interrupt-handler (*os*))
= os-output-interrupt-handler (*os*))

EVENT: Disable trace-output-interrupt-handler.

THEOREM: good-os-os-output-interrupt-handler
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ good-os (os-output-interrupt-handler (*os*))

EVENT: Disable good-os-os-output-interrupt-handler.

DEFINITION:
os-step (*os*)
= **if** tm-input-interruptp (*os*) **then** os-input-interrupt-handler (*os*)
elseif tm-output-interruptp (*os*) **then** os-output-interrupt-handler (*os*)
elseif tm-waiting (*os*) **then** *os*
elseif tm-errorp (*os*) **then** os-error-handler (*os*)
elseif tm-clock-interruptp (*os*) **then** os-clock-interrupt-handler (*os*)
elseif tm-svc-interruptp (*os*) **then** os-svc-handler (*os*)
else tm-fetch-execute (*os*) **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
good-os (*os*) → good-os (os-step (*os*))

EVENT: Disable good-os-os-step.

THEOREM: good-os-os-processor
good-os (*os*) → good-os (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*),

btimed-tm-processor (*tm*,
cdr (*oracle*))))
else *tm* **endif**

THEOREM: timed-tm-implements-btimed-tm
timed-tm-processor (*tm*, *oracle*) = btimed-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 (btimed-tm-processor (*os*, *oracle*))

EVENT: Disable good-os-btimed-tm-processor.

THEOREM: btimed-tm-implements-bos
good-os (*os*)
→ (btimed-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* ∈ **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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-clock-interrupt-handler (*os*))
 = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-clock-interrupt-handler.

THEOREM: mapup-cc-os-clock-interrupt-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-clock-interrupt-handler (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-clock-interrupt-handler.

THEOREM: mapup-error-os-clock-interrupt-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error (*taskid*, os-clock-interrupt-handler (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-clock-interrupt-handler.

THEOREM: mapup-svcflag-os-clock-interrupt-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag (*taskid*, os-clock-interrupt-handler (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-clock-interrupt-handler.

THEOREM: mapup-svcid-os-clock-interrupt-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-clock-interrupt-handler (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-clock-interrupt-handler.

THEOREM: mapup-task-os-clock-interrupt-handler
(good-os (*os*)

$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16'$)))$
$\rightarrow$ (mapup-task $(taskid,$ os-clock-interrupt-handler $(os))$
$=$ mapup-task $(taskid,\ os))$

EVENT: Disable mapup-task-os-clock-interrupt-handler.

THEOREM: mapup-tasks-os-clock-interrupt-handler
(good-os $(os) \wedge ((\neg$ tm-waiting $(os)) \wedge (taskid \in \mathbf{N})))$
$\rightarrow$ (mapup-tasks $(taskid,$ os-clock-interrupt-handler $(os))$
$=$ mapup-tasks $(taskid,\ os))$

EVENT: Disable mapup-tasks-os-clock-interrupt-handler.

THEOREM: mapup-os-tasks-os-clock-interrupt-handler
(good-os $(os) \wedge (\neg$ tm-waiting $(os)))$
$\rightarrow$ (mapup-os-tasks (os-clock-interrupt-handler $(os))$
$=$ mapup-os-tasks $(os))$

EVENT: Disable mapup-os-tasks-os-clock-interrupt-handler.

THEOREM: mapup-os-ibuffers-os-clock-interrupt-handler
good-os (os)
$\rightarrow$ (mapup-os-ibuffers (os-clock-interrupt-handler $(os))$
$=$ mapup-os-ibuffers $(os))$

EVENT: Disable mapup-os-ibuffers-os-clock-interrupt-handler.

THEOREM: mapup-os-obuffers-os-clock-interrupt-handler
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-clock-interrupt-handler $(os))$
$=$ mapup-os-obuffers $(os))$

EVENT: Disable mapup-os-obuffers-os-clock-interrupt-handler.

THEOREM: mapup-os-mbuffers-os-clock-interrupt-handler
good-os (os)
$\rightarrow$ (mapup-os-mbuffers (os-clock-interrupt-handler $(os))$
$=$ mapup-os-mbuffers $(os))$

EVENT: Disable mapup-os-mbuffers-os-clock-interrupt-handler.

THEOREM: mapup-queue-os-readyq-os-clock-interrupt-handler

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-queue (os-readyq (os-clock-interrupt-handler (*os*)))
= enq (qfirst (mapup-queue (os-readyq (*os*))),
deq (mapup-queue (os-readyq (*os*)))))

EVENT: Disable mapup-queue-os-readyq-os-clock-interrupt-handler.

THEOREM: mapup-os-status-table-os-clock-interrupt-handler
good-os (*os*)
→ (table ('2, os-status-table (os-clock-interrupt-handler (*os*)))
= table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-clock-interrupt-handler.

THEOREM: tm-rwstate-os-clock-interrupt-handler
tm-rwstate (os-clock-interrupt-handler (*os*)) = '0

EVENT: Disable tm-rwstate-os-clock-interrupt-handler.

THEOREM: tm-clock-os-clock-interrupt-handler
tm-clock (os-clock-interrupt-handler (*os*)) = '1000

EVENT: Disable tm-clock-os-clock-interrupt-handler.

THEOREM: tm-iports-os-clock-interrupt-handler
tm-iports (os-clock-interrupt-handler (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-clock-interrupt-handler.

THEOREM: tm-oports-os-clock-interrupt-handler
tm-oports (os-clock-interrupt-handler (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-clock-interrupt-handler.

THEOREM: correctness-of-clock-interrupt-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (mapup-os (os-clock-interrupt-handler (*os*))
= ak-clock-interrupt-handler (mapup-os (*os*)))

EVENT: Disable correctness-of-clock-interrupt-handler.

THEOREM: mapup-base-os-error-handler-path1
good-os (*os*)
→ (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*)
→ (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*) ∧ (*taskid* < '16))
→ (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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-cc (*taskid*, os-error-handler-path1 (*os*)) = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-error-handler-path1.

THEOREM: mapup-error-os-error-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-error (*taskid*, os-error-handler-path1 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-error-handler-path1.

THEOREM: mapup-svcflag-os-error-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcflag (*taskid*, os-error-handler-path1 (*os*))
 = mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-error-handler-path1.

THEOREM: mapup-svcid-os-error-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcid (*taskid*, os-error-handler-path1 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-error-handler-path1.

THEOREM: mapup-task-os-error-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-task (*taskid*, os-error-handler-path1 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-error-handler-path1.

THEOREM: mapup-tasks-os-error-handler-path1
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-error-handler-path1 (*os*))
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-error-handler-path1.

THEOREM: mapup-os-tasks-os-error-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
→ (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*)
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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
tm-rwstate (os-error-handler-path1 (*os*)) = '1

EVENT: Disable tm-rwstate-os-error-handler-path1.

THEOREM: tm-clock-os-error-handler-path1
tm-clock (os-error-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-error-handler-path1.

THEOREM: tm-iports-os-error-handler-path1
tm-iports (os-error-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-error-handler-path1.

THEOREM: tm-oports-os-error-handler-path1
tm-oports (os-error-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-error-handler-path1.

THEOREM: correctness-of-os-error-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ array-qemptyp (array-deq (os-readyq (*os*)))))
→ (mapup-os (os-error-handler-path1 (*os*))
 = ak-error-handler (mapup-os (*os*)))

EVENT: Disable correctness-of-os-error-handler-path1.

THEOREM: mapup-base-os-error-handler-path2
good-os (*os*)
→ (mapup-base (*taskid*, os-error-handler-path2 (*os*))
 = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-error-handler-path2.

THEOREM: mapup-limit-os-error-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-error-handler-path2 (*os*))
 = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-error-handler-path2.

THEOREM: getseg-tm-memory-os-error-handler-path2
(good-os (*os*) ∧ (*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

$$
\begin{array}{l}
(\text{good-os}\,(os)\\
\quad \wedge \quad ((\neg\ \text{tm-waiting}\,(os))\\
\qquad \wedge \quad (\text{array-qfirst}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))\\
\qquad\quad = \quad \text{array-qfirst}\,(\text{os-readyq}\,(os)))))\\
\rightarrow \quad (\text{tm-cpu}\,(\text{os-error-handler-path2}\,(os))\\
\qquad = \quad \text{getnth}\,(\text{os-current-taskid}\,(\text{os-error-handler-path2}\,(os)),\\
\qquad\qquad\quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-error-handler-path2}\,(os)))))
\end{array}
$$

THEOREM: tm-cpu-os-error-handler-path2-case2

$$
\begin{array}{l}
(\text{good-os}\,(os)\\
\quad \wedge \quad ((\neg\ \text{tm-waiting}\,(os))\\
\qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\
\qquad\quad \wedge \quad (\text{array-qfirst}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))
\end{array}
$$

$$\begin{array}{lll} & \neq & \text{array-qfirst}\,(\text{os-readyq}\,(os))))))\\ \rightarrow & (\text{tm-cpu}\,(\text{os-error-handler-path2}\,(os))\\ & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-error-handler-path2}\,(os)),\\ & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-error-handler-path2}\,(os)))))\end{array}$$

THEOREM: tm-cpu-os-error-handler-path2

$$\begin{array}{ll} & (\text{good-os}\,(os)\\ \wedge & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge\ (\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))))\\ \rightarrow & (\text{tm-cpu}\,(\text{os-error-handler-path2}\,(os))\\ & = \quad \text{getnth}\,(\text{os-current-taskid}\,(\text{os-error-handler-path2}\,(os)),\\ & \qquad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-error-handler-path2}\,(os)))))\end{array}$$

EVENT: Disable tm-cpu-os-error-handler-path2.

THEOREM: mapup-cpu-os-error-handler-path2

$$\begin{array}{ll} & (\text{good-os}\,(os)\\ \wedge & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge\ ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & (\text{mapup-cpu}\,(taskid,\ \text{os-error-handler-path2}\,(os))\\ & = \quad \text{mapup-cpu}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-cpu-os-error-handler-path2.

THEOREM: mapup-regs-os-error-handler-path2

$$\begin{array}{ll} & (\text{good-os}\,(os)\\ \wedge & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge\ ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & (\text{mapup-regs}\,(taskid,\ \text{os-error-handler-path2}\,(os))\\ & = \quad \text{mapup-regs}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-regs-os-error-handler-path2.

THEOREM: mapup-cc-os-error-handler-path2

$$\begin{array}{ll} & (\text{good-os}\,(os)\\ \wedge & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge\ ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & (\text{mapup-cc}\,(taskid,\ \text{os-error-handler-path2}\,(os)) = \text{mapup-cc}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-cc-os-error-handler-path2.

THEOREM: mapup-error-os-error-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))

$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (os))))
$\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-error-handler-path2 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-error-handler-path2.

THEOREM: mapup-os-tasks-os-error-handler-path2
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (¬ array-qemptyp (array-deq (os-readyq (os))))))
$\rightarrow$ (mapup-os-tasks (os-error-handler-path2 (os)) = mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-error-handler-path2.

THEOREM: mapup-os-ibuffers-os-error-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-ibuffers (os-error-handler-path2 (os))
= mapup-os-ibuffers (os))

EVENT: Disable mapup-os-ibuffers-os-error-handler-path2.

THEOREM: mapup-os-obuffers-os-error-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-error-handler-path2 (os))
= mapup-os-obuffers (os))

EVENT: Disable mapup-os-obuffers-os-error-handler-path2.

THEOREM: mapup-os-mbuffers-os-error-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-mbuffers (os-error-handler-path2 (os))
= mapup-os-mbuffers (os))

EVENT: Disable mapup-os-mbuffers-os-error-handler-path2.

THEOREM: mapup-queue-os-readyq-os-error-handler-path2
(good-os (os) $\wedge$ (¬ tm-waiting (os)))
$\rightarrow$ (mapup-queue (os-readyq (os-error-handler-path2 (os)))
= deq (mapup-queue (os-readyq (os))))

EVENT: Disable mapup-queue-os-readyq-os-error-handler-path2.

THEOREM: mapup-os-status-table-os-error-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table ('2, os-status-table (os-error-handler-path2 (*os*)))
= putnth ('(1 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-error-handler-path2.

THEOREM: tm-clock-os-error-handler-path2
tm-clock (os-error-handler-path2 (*os*)) = '1000

EVENT: Disable tm-clock-os-error-handler-path2.

THEOREM: tm-iports-os-error-handler-path2
tm-iports (os-error-handler-path2 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-error-handler-path2.

THEOREM: tm-oports-os-error-handler-path2
tm-oports (os-error-handler-path2 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-error-handler-path2.

THEOREM: tm-rwstate-os-error-handler-path2
tm-rwstate (os-error-handler-path2 (*os*)) = '0

EVENT: Disable tm-rwstate-os-error-handler-path2.

THEOREM: correctness-of-os-error-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
→ (mapup-os (os-error-handler-path2 (*os*))
= ak-error-handler (mapup-os (*os*)))

EVENT: Disable correctness-of-os-error-handler-path2.

THEOREM: correctness-of-os-error-handler
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-os (os-error-handler (*os*)) = ak-error-handler (mapup-os (*os*)))

EVENT: Disable correctness-of-os-error-handler.

THEOREM: mapup-base-os-svc-send-handler-path1
good-os (*os*)
→ (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*)
→ (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*) ∧ (*taskid* < '16))
→ (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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-cc (*taskid*, *os*))

Event: Disable mapup-cc-os-svc-send-handler-path1.

Theorem: mapup-error-os-svc-send-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-error (*taskid*, *os*))

Event: Disable mapup-error-os-svc-send-handler-path1.

Theorem: mapup-svcflag-os-svc-send-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-svcflag (*taskid*, *os*))

Event: Disable mapup-svcflag-os-svc-send-handler-path1.

Theorem: mapup-svcid-os-svc-send-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-svcid (*taskid*, *os*))

Event: Disable mapup-svcid-os-svc-send-handler-path1.

Theorem: mapup-task-os-svc-send-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-task (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-task (*taskid*, *os*))

Event: Disable mapup-task-os-svc-send-handler-path1.

Theorem: mapup-tasks-os-svc-send-handler-path1
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-svc-send-handler-path1 (*os*))
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-svc-send-handler-path1.

THEOREM: mapup-os-tasks-os-svc-send-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
→ (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*)
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*))))

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path1.

THEOREM: tm-rwstate-os-svc-send-handler-path1
tm-rwstate (os-svc-send-handler-path1 (*os*)) = '1

EVENT: Disable tm-rwstate-os-svc-send-handler-path1.

THEOREM: tm-clock-os-svc-send-handler-path1
tm-clock (os-svc-send-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-send-handler-path1.

THEOREM: tm-iports-os-svc-send-handler-path1
tm-iports (os-svc-send-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-send-handler-path1.

THEOREM: tm-oports-os-svc-send-handler-path1
tm-oports (os-svc-send-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-send-handler-path1.

THEOREM: correctness-of-os-svc-send-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '0)
    ∧ (array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
                                         os-destid (*os*,
                                                    '16),
                                         os))
     ∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ (mapup-os (os-svc-send-handler-path1 (*os*))
  = ak-execute-send (os-message (*os*),
                     os-current-taskid (*os*),
                     os-destid (*os*, '16),
                     mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-send-handler-path1.

THEOREM: mapup-base-os-svc-send-handler-path2
good-os (*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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (array-qfirst (array-deq (os-readyq (*os*)))

$$
\begin{array}{lll}
 & = & \text{array-qfirst}\,(\text{os-readyq}\,(os))))) \\
\rightarrow & (\text{tm-cpu}\,(\text{os-svc-send-handler-path2}\,(os)) \\
 & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-send-handler-path2}\,(os)), \\
 & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-send-handler-path2}\,(os)))))
\end{array}
$$

THEOREM: tm-cpu-os-svc-send-handler-path2-case2

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
 & & \wedge \quad (\text{array-qfirst}\,(\text{array-deq}\,(\text{os-readyq}\,(os))) \\
 & & \quad \neq \quad \text{array-qfirst}\,(\text{os-readyq}\,(os)))))) \\
\rightarrow & (\text{tm-cpu}\,(\text{os-svc-send-handler-path2}\,(os)) & \\
 & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-send-handler-path2}\,(os)), \\
 & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-send-handler-path2}\,(os)))))
\end{array}
$$

THEOREM: tm-cpu-os-svc-send-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & (\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))) \\
\rightarrow & (\text{tm-cpu}\,(\text{os-svc-send-handler-path2}\,(os)) & \\
 & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-send-handler-path2}\,(os)), \\
 & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-send-handler-path2}\,(os)))))
\end{array}
$$

EVENT: Disable tm-cpu-os-svc-send-handler-path2.

THEOREM: mapup-cpu-os-svc-send-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
 & & \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-cpu}\,(taskid,\ \text{os-svc-send-handler-path2}\,(os)) & \\
 & = & \text{mapup-cpu}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path2.

THEOREM: mapup-regs-os-svc-send-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
 & & \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-regs}\,(taskid,\ \text{os-svc-send-handler-path2}\,(os)) & \\
 & = & \text{mapup-regs}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-regs-os-svc-send-handler-path2.

THEOREM: mapup-cc-os-svc-send-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (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-qemptyp (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-qemptyp (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-qemptyp (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*))

$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-task (*taskid*, os-svc-send-handler-path2 (*os*))
= mapup-task (*taskid*, *os*))

Event: Disable mapup-task-os-svc-send-handler-path2.

Theorem: mapup-tasks-os-svc-send-handler-path2
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
$\wedge$ (*taskid* $\in$ **N**))))
$\rightarrow$ (mapup-tasks (*taskid*, os-svc-send-handler-path2 (*os*))
= mapup-tasks (*taskid*, *os*))

Event: Disable mapup-tasks-os-svc-send-handler-path2.

Theorem: mapup-os-tasks-os-svc-send-handler-path2
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
$\rightarrow$ (mapup-os-tasks (os-svc-send-handler-path2 (*os*)) = mapup-os-tasks (*os*))

Event: Disable mapup-os-tasks-os-svc-send-handler-path2.

Theorem: mapup-os-ibuffers-os-svc-send-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-ibuffers (os-svc-send-handler-path2 (*os*))
= mapup-os-ibuffers (*os*))

Event: Disable mapup-os-ibuffers-os-svc-send-handler-path2.

Theorem: mapup-os-obuffers-os-svc-send-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-obuffers (os-svc-send-handler-path2 (*os*))
= mapup-os-obuffers (*os*))

Event: Disable mapup-os-obuffers-os-svc-send-handler-path2.

Theorem: mapup-os-mbuffers-os-svc-send-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-mbuffers (os-svc-send-handler-path2 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-send-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-send-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ 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))
$\wedge$ ($\neg$ array-qemptyp (array-deq (os-readyq (*os*)))))))))
$\rightarrow$ (mapup-os (os-svc-send-handler-path2 (*os*))
= ak-execute-send (os-message (*os*),
os-current-taskid (*os*),
os-destid (*os*, '16),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-send-handler-path2.

THEOREM: mapup-base-os-svc-send-handler-path3
good-os (*os*)
$\rightarrow$ (mapup-base (*taskid*, os-svc-send-handler-path3 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-send-handler-path3.

THEOREM: mapup-limit-os-svc-send-handler-path3
good-os (*os*)
$\rightarrow$ (mapup-limit (*taskid*, os-svc-send-handler-path3 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-send-handler-path3.

THEOREM: getseg-tm-memory-os-svc-send-handler-path3
(good-os (*os*) $\wedge$ (*taskid* < '16))
$\rightarrow$ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-send-handler-path3 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-send-handler-path3.

THEOREM: tm-cpu-os-svc-send-handler-path3
(good-os (*os*) $\wedge$ ($\neg$ tm-waiting (*os*)))
$\rightarrow$ (tm-cpu (os-svc-send-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-send-handler-path3.

THEOREM: array-qfirst-os-readyq-os-svc-send-handler-path3

$$
\begin{array}{ll}
 & (\text{good-os}\,(os) \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) \wedge (\neg\ \text{array-qfullp}\,(\text{os-readyq}\,(os)))) \\
\rightarrow & (\text{array-qfirst}\,(\text{os-readyq}\,(\text{os-svc-send-handler-path3}\,(os))) \\
 & \quad = \quad \text{array-qfirst}\,(\text{os-readyq}\,(os)))
\end{array}
$$

EVENT: Disable array-qfirst-os-readyq-os-svc-send-handler-path3.

THEOREM: mapup-cpu-os-svc-send-handler-path3-case1

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ (¬ array-qfullp (os-readyq (os))))
 →   (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

$$
\begin{array}{ll}
 & (\text{good-os}\,(os) \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) \\
 & \quad \wedge \quad ((\neg\ \text{array-qfullp}\,(\text{os-readyq}\,(os))) \\
 & \qquad \wedge \quad ((taskid \in \mathbf{N}) \\
 & \qquad\quad \wedge \quad ((taskid < \texttt{'16}) \\
 & \qquad\qquad \wedge \quad (taskid \neq \text{array-qfirst}\,(\text{os-readyq}\,(os)))))))) \\
\rightarrow & (\text{mapup-cpu}\,(taskid, \text{os-svc-send-handler-path3}\,(os)) \\
 & \quad = \quad \text{mapup-cpu}\,(taskid, os))
\end{array}
$$

EVENT: Disable mapup-cpu-os-svc-send-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-send-handler-path3

$$
\begin{array}{ll}
 & (\text{good-os}\,(os) \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) \\
 & \quad \wedge \quad ((\neg\ \text{array-qfullp}\,(\text{os-readyq}\,(os))) \\
 & \qquad \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))
\end{array}
$$

→ (mapup-cpu (*taskid*, os-svc-send-handler-path3 (*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-send-handler-path3.

THEOREM: mapup-regs-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qfullp (os-readyq (*os*)))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-regs (*taskid*, os-svc-send-handler-path3 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-send-handler-path3.

THEOREM: mapup-cc-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qfullp (os-readyq (*os*)))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cc (*taskid*, os-svc-send-handler-path3 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-send-handler-path3.

THEOREM: mapup-error-os-svc-send-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qfullp (os-readyq (*os*)))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-error (*taskid*, os-svc-send-handler-path3 (*os*))
= mapup-error (*taskid*, *os*))

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
(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-send-handler-path3 (*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-path3-case2.

THEOREM: mapup-os-tasks-os-svc-send-handler-path3-support
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ (¬ array-qfullp (os-readyq (*os*)))))
→ (mapup-os-tasks (os-svc-send-handler-path3 (*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-path3-support.

THEOREM: mapup-os-tasks-os-svc-send-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (getnth (`'2` ∗ os-destid (*os*, `'16`), os-status-table (*os*))
    = `'3`)))
→ (mapup-os-tasks (os-svc-send-handler-path3 (*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-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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (getnth ('2 ∗ os-destid (*os*, '16), os-status-table (*os*))
   = '3)))
→ (mapup-queue (os-readyq (os-svc-send-handler-path3 (*os*)))
  = enq (os-destid (*os*, '16), mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-send-handler-path3.

THEOREM: mapup-os-status-table-os-svc-send-handler-path3
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table ('2, os-status-table (os-svc-send-handler-path3 (*os*)))
  = putnth ('(0 0),
    os-destid (*os*, '16),
    table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path3.

THEOREM: tm-clock-os-svc-send-handler-path3
tm-clock (os-svc-send-handler-path3 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-send-handler-path3.

THEOREM: tm-iports-os-svc-send-handler-path3
tm-iports (os-svc-send-handler-path3 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-send-handler-path3.

THEOREM: tm-oports-os-svc-send-handler-path3
tm-oports (os-svc-send-handler-path3 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-send-handler-path3.

THEOREM: tm-rwstate-os-svc-send-handler-path3
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-svc-send-handler-path3 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-send-handler-path3.

THEOREM: ak-waiting-to-receivep-mapup-os
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))

```
    ∧   ((¬ tm-errorp (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))))))
→   ak-waiting-to-receivep (os-current-taskid (os),
                            os-destid (os, '16),
                            mapup-os (os))
```

EVENT: Disable ak-waiting-to-receivep-mapup-os.

THEOREM: correctness-of-os-svc-send-handler-path3

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   ((¬ tm-errorp (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)))))))))
→   (mapup-os (os-svc-send-handler-path3 (os))
     =   ak-execute-send (os-message (os),
                          os-current-taskid (os),
                          os-destid (os, '16),
                          mapup-os (os)))
```

EVENT: Disable correctness-of-os-svc-send-handler-path3.

THEOREM: mapup-base-os-svc-send-handler-path4

```
good-os (os)
→   (mapup-base (taskid, os-svc-send-handler-path4 (os))
     =   mapup-base (taskid, os))
```

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

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   ((taskid ∈ N)
           ∧   ((taskid < '16)
                ∧   (taskid ≠ array-qfirst (os-readyq (os)))))))
→   (mapup-cpu (taskid, os-svc-send-handler-path4 (os))
      =   mapup-cpu (taskid, os))
```

EVENT: Disable mapup-cpu-os-svc-send-handler-path4-case2.

THEOREM: mapup-cpu-os-svc-send-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ ((taskid ∈ N) ∧ (taskid < '16))))
→   (mapup-cpu (taskid, os-svc-send-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-send-handler-path4.

THEOREM: mapup-regs-os-svc-send-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ ((taskid ∈ N) ∧ (taskid < '16))))
→   (mapup-regs (taskid, os-svc-send-handler-path4 (os))
      =   mapup-regs (taskid, os))
```

EVENT: Disable mapup-regs-os-svc-send-handler-path4.

THEOREM: mapup-cc-os-svc-send-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ ((taskid ∈ N) ∧ (taskid < '16))))
→   (mapup-cc (taskid, os-svc-send-handler-path4 (os))
      =   mapup-cc (taskid, os))
```

EVENT: Disable mapup-cc-os-svc-send-handler-path4.

THEOREM: mapup-error-os-svc-send-handler-path4

```
(good-os (os)
```

$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-error $(taskid,$ os-svc-send-handler-path4 $(os))$
$=$ mapup-error $(taskid,\ os))$

EVENT: Disable mapup-error-os-svc-send-handler-path4.

THEOREM: mapup-svcflag-os-svc-send-handler-path4
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-svcflag $(taskid,$ os-svc-send-handler-path4 $(os))$
$=$ **if** $taskid =$ array-qfirst (os-readyq $(os))$ **then** '0
else mapup-svcflag $(taskid,\ os)$ **endif**)

EVENT: Disable mapup-svcflag-os-svc-send-handler-path4.

THEOREM: mapup-svcid-os-svc-send-handler-path4
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-svcid $(taskid,$ os-svc-send-handler-path4 $(os))$
$=$ mapup-svcid $(taskid,\ os))$

EVENT: Disable mapup-svcid-os-svc-send-handler-path4.

THEOREM: mapup-task-os-svc-send-handler-path4
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-task $(taskid,$ os-svc-send-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-send-handler-path4.

THEOREM: mapup-tasks-os-svc-send-handler-path4-case1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os))$
$\wedge$ $((taskid \in \mathbf{N}) \wedge (\text{array-qfirst (os-readyq } (os)) < taskid))))$
$\rightarrow$ (mapup-tasks $(taskid,$ os-svc-send-handler-path4 $(os))$
$=$ mapup-tasks $(taskid,\ os))$

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*)
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qfullp (os-current-mbuffer (os-current-taskid (*os*),
os-destid (*os*, '16),
os)))))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table ('2, os-status-table (os-svc-send-handler-path4 (*os*)))
= table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-svc-send-handler-path4.

THEOREM: tm-clock-os-svc-send-handler-path4
tm-clock (os-svc-send-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-send-handler-path4.

THEOREM: tm-iports-os-svc-send-handler-path4
tm-iports (os-svc-send-handler-path4 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-send-handler-path4.

THEOREM: tm-oports-os-svc-send-handler-path4
tm-oports (os-svc-send-handler-path4 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-send-handler-path4.

THEOREM: tm-rwstate-os-svc-send-handler-path4
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-svc-send-handler-path4 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-send-handler-path4.

THEOREM: not-ak-waiting-to-receivep-mapup-os
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*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*)))))))
→ (¬ ak-waiting-to-receivep (os-current-taskid (*os*),
os-destid (*os*, '16),
mapup-os (*os*)))

EVENT: Disable not-ak-waiting-to-receivep-mapup-os.

THEOREM: correctness-of-os-svc-send-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*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*)))))))))
→ (mapup-os (os-svc-send-handler-path4 (*os*))
  = ak-execute-send (os-message (*os*),
                     os-current-taskid (*os*),
                     os-destid (*os*, '16),
                     mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-send-handler-path4.

THEOREM: correctness-of-os-svc-send-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '0))))
→ (mapup-os (os-svc-send-handler (*os*))
  = ak-execute-send (os-message (*os*),
                     os-current-taskid (*os*),
                     os-destid (*os*, '16),
                     mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-send-handler.

THEOREM: mapup-base-os-svc-receive-handler-path1

good-os (*os*)
→ (mapup-base (*taskid*, os-svc-receive-handler-path1 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-receive-handler-path1.

THEOREM: mapup-limit-os-svc-receive-handler-path1
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-receive-handler-path1 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-receive-handler-path1.

THEOREM: getseg-tm-memory-os-svc-receive-handler-path1
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-receive-handler-path1 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-receive-handler-path1.

THEOREM: mapup-cpu-os-svc-receive-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cpu (*taskid*, os-svc-receive-handler-path1 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-receive-handler-path1.

THEOREM: mapup-regs-os-svc-receive-handler-path1
(good-os (*os*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-svc-receive-handler-path1 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-receive-handler-path1.

THEOREM: mapup-cc-os-svc-receive-handler-path1
(good-os (*os*)

$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow$ (mapup-cc $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-cc $(taskid,\ os))$

Event: Disable mapup-cc-os-svc-receive-handler-path1.

Theorem: mapup-error-os-svc-receive-handler-path1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow$ (mapup-error $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-error $(taskid,\ os))$

Event: Disable mapup-error-os-svc-receive-handler-path1.

Theorem: mapup-svcflag-os-svc-receive-handler-path1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow$ (mapup-svcflag $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-svcflag $(taskid,\ os))$

Event: Disable mapup-svcflag-os-svc-receive-handler-path1.

Theorem: mapup-svcid-os-svc-receive-handler-path1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow$ (mapup-svcid $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-svcid $(taskid,\ os))$

Event: Disable mapup-svcid-os-svc-receive-handler-path1.

Theorem: mapup-task-os-svc-receive-handler-path1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow$ (mapup-task $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-task $(taskid,\ os))$

Event: Disable mapup-task-os-svc-receive-handler-path1.

Theorem: mapup-tasks-os-svc-receive-handler-path1
(good-os $(os) \wedge ((\neg$ tm-waiting $(os)) \wedge (taskid \in \mathbf{N})))$
$\rightarrow$ (mapup-tasks $(taskid,$ os-svc-receive-handler-path1 $(os))$
$=$ mapup-tasks $(taskid,\ os))$

EVENT: Disable mapup-tasks-os-svc-receive-handler-path1.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
→ (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*)
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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-os-status-table-os-svc-receive-handler-path1.

THEOREM: tm-rwstate-os-svc-receive-handler-path1
tm-rwstate (os-svc-receive-handler-path1 (*os*)) = '1

EVENT: Disable tm-rwstate-os-svc-receive-handler-path1.

THEOREM: tm-clock-os-svc-receive-handler-path1
tm-clock (os-svc-receive-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-receive-handler-path1.

THEOREM: tm-iports-os-svc-receive-handler-path1
tm-iports (os-svc-receive-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-receive-handler-path1.

THEOREM: tm-oports-os-svc-receive-handler-path1
tm-oports (os-svc-receive-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-receive-handler-path1.

THEOREM: correctness-of-os-svc-receive-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '1)
    ∧ (array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16), os-current-taskid (*os*), *os*))
     ∧ array-qemptyp (array-deq (os-readyq (*os*)))))))))
→ (mapup-os (os-svc-receive-handler-path1 (*os*))
  = ak-execute-receive (os-srcid (*os*, '16), os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-receive-handler-path1.

THEOREM: mapup-base-os-svc-receive-handler-path2
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-receive-handler-path2 (*os*))
  = mapup-base (*taskid*, *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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*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
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (array-deq (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*)))))

EVENT: Disable tm-cpu-os-svc-receive-handler-path2.

THEOREM: mapup-cpu-os-svc-receive-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))

$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-task (*taskid*, os-svc-receive-handler-path2 (*os*))
= mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-svc-receive-handler-path2.

THEOREM: mapup-tasks-os-svc-receive-handler-path2
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
$\wedge$ (*taskid* $\in$ **N**))))
$\rightarrow$ (mapup-tasks (*taskid*, os-svc-receive-handler-path2 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-svc-receive-handler-path2.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path2
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
$\rightarrow$ (mapup-os-tasks (os-svc-receive-handler-path2 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path2.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-ibuffers (os-svc-receive-handler-path2 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path2.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-obuffers (os-svc-receive-handler-path2 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path2.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-mbuffers (os-svc-receive-handler-path2 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))

$\wedge$ (((tm-svcid (*os*) **mod** '4) = '1)
$\wedge$ (array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os))
$\wedge$ ($\neg$ array-qemptyp (array-deq (os-readyq (*os*)))))))))
$\rightarrow$ (mapup-os (os-svc-receive-handler-path2 (*os*))
= ak-execute-receive (os-srcid (*os*, '16),
os-current-taskid (*os*),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-receive-handler-path2.

THEOREM: mapup-base-os-svc-receive-handler-path3
good-os (*os*)
$\rightarrow$ (mapup-base (*taskid*, os-svc-receive-handler-path3 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-receive-handler-path3.

THEOREM: mapup-limit-os-svc-receive-handler-path3
good-os (*os*)
$\rightarrow$ (mapup-limit (*taskid*, os-svc-receive-handler-path3 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-receive-handler-path3.

THEOREM: getseg-tm-memory-os-svc-receive-handler-path3
(good-os (*os*) $\wedge$ (*taskid* < '16))
$\rightarrow$ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-receive-handler-path3 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-receive-handler-path3.

THEOREM: tm-cpu-os-svc-receive-handler-path3
(good-os (*os*) $\wedge$ ($\neg$ tm-waiting (*os*)))
$\rightarrow$ (tm-cpu (os-svc-receive-handler-path3 (*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-path3.

THEOREM: array-qfirst-os-readyq-os-svc-receive-handler-path3
(good-os (os)
∧ ((¬ tm-waiting (os)) ∧ (¬ array-qfullp (os-readyq (os)))))
→ (array-qfirst (os-readyq (os-svc-receive-handler-path3 (os)))
= array-qfirst (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)
∧ ((¬ tm-waiting (os)) ∧ (¬ array-qfullp (os-readyq (os)))))
→ (mapup-cpu (array-qfirst (os-readyq (os)),
os-svc-receive-handler-path3 (os))
= putnth (array-qfirst (os-current-mbuffer (os-srcid (os, '16),
os-current-taskid (os),
os)),
'3,
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-receive-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-receive-handler-path3-case2
(good-os (os)
∧ ((¬ tm-waiting (os))

$\wedge$ ((¬ array-qfullp (os-readyq (*os*)))
$\wedge$ ((*taskid* $\in$ **N**)
$\wedge$ ((*taskid* $<$ '16)
$\wedge$ (*taskid* $\neq$ array-qfirst (os-readyq (*os*)))))))
$\rightarrow$ (mapup-cpu (*taskid*, os-svc-receive-handler-path3 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-receive-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-receive-handler-path3
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ array-qfullp (os-readyq (*os*)))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-cpu (*taskid*, os-svc-receive-handler-path3 (*os*))
= **if** *taskid* = array-qfirst (os-readyq (*os*))
then putnth (array-qfirst (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)),
'3,
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-receive-handler-path3.

THEOREM: mapup-regs-os-svc-receive-handler-path3
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ ((¬ array-qfullp (os-readyq (*os*)))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-regs (*taskid*, os-svc-receive-handler-path3 (*os*))
= **if** *taskid* = array-qfirst (os-readyq (*os*))
then putnth (array-qfirst (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)),
'3,
mapup-regs (*taskid*, *os*))
else mapup-regs (*taskid*, *os*) **endif**)

EVENT: Disable mapup-regs-os-svc-receive-handler-path3.

THEOREM: mapup-cc-os-svc-receive-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qfullp (os-readyq (*os*)))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qfullp (os-readyq (*os*)))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qfullp (os-readyq (*os*)))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qfullp (os-readyq (*os*)))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*))

$-$ *taskid*,
mapup-tasks (*taskid*,
os)))),
array-qfirst (os-readyq (*os*)) $-$ *taskid*,
mapup-tasks (*taskid*, *os*)))

EVENT: Disable mapup-tasks-os-svc-receive-handler-path3-case2.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path3-support
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*)) $\wedge$ (¬ array-qfullp (os-readyq (*os*)))))
$\rightarrow$ (mapup-os-tasks (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*)),
mapup-os-tasks (*os*)))),
array-qfirst (os-readyq (*os*)),
mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path3-support.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path3
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ (getnth (`'2` $*$ os-srcid (*os*, `'16`), os-status-table (*os*))
= `'2`)))
$\rightarrow$ (mapup-os-tasks (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 (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-receive-handler-path3.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path3

good-os (*os*)
→ (mapup-os-ibuffers (os-svc-receive-handler-path3 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path3.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path3
good-os (*os*)
→ (mapup-os-obuffers (os-svc-receive-handler-path3 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path3.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path3-support
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)))))
→ (mapup-os-mbuffers (os-svc-receive-handler-path3 (*os*))
= deq2 (os-srcid (*os*, '16),
array-qfirst (os-readyq (*os*)),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path3-support.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16),
os-current-taskid (*os*),
os)))))
→ (mapup-os-mbuffers (os-svc-receive-handler-path3 (*os*))
= deq2 (os-srcid (*os*, '16),
qfirst (mapup-queue (os-readyq (*os*))),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path3.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path3
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (getnth ('2 ∗ os-srcid (*os*, '16), os-status-table (*os*))

$= \quad$ '2)))
$\rightarrow$ (mapup-queue (os-readyq (os-svc-receive-handler-path3 (os)))
$=$ enq (os-srcid (os, '16), mapup-queue (os-readyq (os))))

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path3.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path3
(good-os (os) $\wedge$ ($\neg$ tm-waiting (os)))
$\rightarrow$ (table ('2, os-status-table (os-svc-receive-handler-path3 (os)))
$=$ putnth ('(0 0),
os-srcid (os, '16),
table ('2, os-status-table (os))))

EVENT: Disable mapup-os-status-table-os-svc-receive-handler-path3.

THEOREM: tm-clock-os-svc-receive-handler-path3
tm-clock (os-svc-receive-handler-path3 (os)) = tm-clock (os)

EVENT: Disable tm-clock-os-svc-receive-handler-path3.

THEOREM: tm-iports-os-svc-receive-handler-path3
tm-iports (os-svc-receive-handler-path3 (os)) = tm-iports (os)

EVENT: Disable tm-iports-os-svc-receive-handler-path3.

THEOREM: tm-oports-os-svc-receive-handler-path3
tm-oports (os-svc-receive-handler-path3 (os)) = tm-oports (os)

EVENT: Disable tm-oports-os-svc-receive-handler-path3.

THEOREM: tm-rwstate-os-svc-receive-handler-path3
(good-tm (os) $\wedge$ ($\neg$ tm-waiting (os)))
$\rightarrow$ (tm-rwstate (os-svc-receive-handler-path3 (os)) = tm-rwstate (os))

EVENT: Disable tm-rwstate-os-svc-receive-handler-path3.

THEOREM: ak-waiting-to-sendp-mapup-os
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os))
$\wedge$ (($\neg$ tm-errorp (os))
$\wedge$ ((getnth ('2 $*$ os-srcid (os, '16),
os-status-table (os))

$$
\begin{array}{l}
\quad = \quad \texttt{'2}) \\
\quad \wedge \quad (\text{getnth}\,(\texttt{'1} + (\texttt{'2} * \text{os-srcid}\,(os,\ \texttt{'16})), \\
\qquad\qquad \text{os-status-table}\,(os)) \\
\qquad\quad = \quad \text{os-current-taskid}\,(os)))))) \\
\rightarrow \quad \text{ak-waiting-to-sendp}\,(\text{os-srcid}\,(os,\ \texttt{'16}), \\
\qquad\qquad\qquad\qquad \text{os-current-taskid}\,(os), \\
\qquad\qquad\qquad\qquad \text{mapup-os}\,(os))
\end{array}
$$

EVENT: Disable ak-waiting-to-sendp-mapup-os.

THEOREM: tm-set-svcflag-commutes-with-os-store-message

$$
\begin{array}{l}
\text{os-store-message}\,(msg,\ \text{tm-set-svcflag}\,(flag,\ tm)) \\
= \quad \text{tm-set-svcflag}\,(flag,\ \text{os-store-message}\,(msg,\ tm))
\end{array}
$$

EVENT: Disable tm-set-svcflag-commutes-with-os-store-message.

THEOREM: correctness-of-os-svc-receive-handler-path3

$$
\begin{array}{l}
(\text{good-os}\,(os) \\
\quad \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
\qquad \wedge \quad ((\neg\ \text{tm-errorp}\,(os)) \\
\qquad\quad \wedge \quad (((\text{tm-svcid}\,(os)\ \textbf{mod}\ \texttt{'4}) = \texttt{'1}) \\
\qquad\qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os, \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-current-taskid}\,(os), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os))) \\
\qquad\qquad\qquad\qquad \wedge \quad ((\text{getnth}\,(\texttt{'2} * \text{os-srcid}\,(os,\ \texttt{'16}), \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \texttt{'2}) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \quad (\text{getnth}\,(\texttt{'1} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \quad (\texttt{'2} \\
\qquad * \quad \text{os-srcid}\,(os, \\
\qquad \texttt{'16})), \\
\qquad \text{os-status-table}\,(os)) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad = \quad \text{os-current-taskid}\,(os))))))))) \\
\rightarrow \quad (\text{mapup-os}\,(\text{os-svc-receive-handler-path3}\,(os)) \\
\qquad = \quad \text{ak-execute-receive}\,(\text{os-srcid}\,(os,\ \texttt{'16}), \\
\qquad \text{os-current-taskid}\,(os), \\
\qquad \text{mapup-os}\,(os)))
\end{array}
$$

EVENT: Disable correctness-of-os-svc-receive-handler-path3.

THEOREM: mapup-base-os-svc-receive-handler-path4

$$
\text{good-os}\,(os)
$$

→ (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*))

$$
\begin{array}{ll}
= & \text{putnth}\,(\text{array-qfirst}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os,\ \texttt{'16}),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-current-taskid}\,(os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)),\\
& \qquad\qquad\qquad\qquad \texttt{'3},\\
& \qquad\qquad\qquad\qquad \text{putnth}\,(\text{tm-pack-psw}\,(\text{mapup-cc}\,(\text{array-qfirst}\,(\text{os-readyq}\,(os)),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-error}\,(\text{array-qfirst}\,(\text{os-readyq}\,(os)),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'0},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-svcid}\,(\text{array-qfirst}\,(\text{os-readyq}\,(os)),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)),\\
& \qquad\qquad\qquad\qquad\qquad\qquad \texttt{'8},\\
& \qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-cpu}\,(\text{array-qfirst}\,(\text{os-readyq}\,(os)),\ os))))
\end{array}
$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path4-case1.

THEOREM: mapup-cpu-os-svc-receive-handler-path4-case2

$$
\begin{array}{ll}
& (\text{good-os}\,(os)\\
\wedge & ((\neg\ \text{tm-waiting}\,(os))\\
\wedge & \quad ((taskid \in \mathbf{N})\\
\wedge & \quad\quad ((taskid < \texttt{'16})\\
\wedge & \quad\quad\quad (taskid \neq \text{array-qfirst}\,(\text{os-readyq}\,(os)))))))\\
\rightarrow & (\text{mapup-cpu}\,(taskid,\ \text{os-svc-receive-handler-path4}\,(os))\\
= & \quad \text{mapup-cpu}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path4-case2.

THEOREM: mapup-cpu-os-svc-receive-handler-path4

$$
\begin{array}{ll}
& (\text{good-os}\,(os)\\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\
\rightarrow & (\text{mapup-cpu}\,(taskid,\ \text{os-svc-receive-handler-path4}\,(os))\\
= & \quad \textbf{if}\ taskid = \text{array-qfirst}\,(\text{os-readyq}\,(os))\\
& \quad \textbf{then}\ \text{putnth}\,(\text{array-qfirst}\,(\text{os-current-mbuffer}\,(\text{os-srcid}\,(os,\ \texttt{'16}),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{os-current-taskid}\,(os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad os)),\\
& \qquad\qquad\qquad\qquad\qquad \texttt{'3},\\
& \qquad\qquad\qquad\qquad\qquad \text{putnth}\,(\text{tm-pack-psw}\,(\text{mapup-cc}\,(taskid,\ os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-error}\,(taskid,\ os),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'0},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-svcid}\,(taskid,\ os)),\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \texttt{'8},\\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{mapup-cpu}\,(taskid,\ os)))\\
& \quad \textbf{else}\ \text{mapup-cpu}\,(taskid,\ os)\ \textbf{endif})
\end{array}
$$

EVENT: Disable mapup-cpu-os-svc-receive-handler-path4.

THEOREM: mapup-regs-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-regs (*taskid*, os-svc-receive-handler-path4 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*))
 then putnth (array-qfirst (os-current-mbuffer (os-srcid (*os*, '16'),
 os-current-taskid (*os*),
 os)),
 '3',
 mapup-regs (*taskid*, *os*))
 else mapup-regs (*taskid*, *os*) **endif**)

EVENT: Disable mapup-regs-os-svc-receive-handler-path4.

THEOREM: mapup-cc-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-cc (*taskid*, os-svc-receive-handler-path4 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-receive-handler-path4.

THEOREM: mapup-error-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-error (*taskid*, os-svc-receive-handler-path4 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-receive-handler-path4.

THEOREM: mapup-svcflag-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcflag (*taskid*, os-svc-receive-handler-path4 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*)) **then** '0
 else mapup-svcflag (*taskid*, *os*) **endif**)

EVENT: Disable mapup-svcflag-os-svc-receive-handler-path4.

THEOREM: mapup-svcid-os-svc-receive-handler-path4

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-svc-receive-handler-path4 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-receive-handler-path4.

THEOREM: mapup-task-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-task (*taskid*, os-svc-receive-handler-path4 (*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-path4.

THEOREM: mapup-tasks-os-svc-receive-handler-path4-case1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (array-qfirst (os-readyq (*os*)) < *taskid*))))
→ (mapup-tasks (*taskid*, os-svc-receive-handler-path4 (*os*))
 = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-svc-receive-handler-path4-case1.

THEOREM: mapup-tasks-os-svc-receive-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-receive-handler-path4 (*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*))
− *taskid*,
mapup-tasks (*taskid*,
os)))),
array-qfirst (os-readyq (*os*)) − *taskid*,
mapup-tasks (*taskid*, *os*)))

EVENT: Disable mapup-tasks-os-svc-receive-handler-path4-case2.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path4-support
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-os-tasks (os-svc-receive-handler-path4 (*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*)),
mapup-os-tasks (*os*)))),
array-qfirst (os-readyq (*os*)),
mapup-os-tasks (*os*)))

EVENT: Disable mapup-os-tasks-os-svc-receive-handler-path4-support.

THEOREM: mapup-os-tasks-os-svc-receive-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-os-tasks (os-svc-receive-handler-path4 (*os*))
= putnth (os-store-message (array-qfirst (os-current-mbuffer (os-srcid (*os*,
'16),
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-receive-handler-path4.

THEOREM: mapup-os-ibuffers-os-svc-receive-handler-path4
good-os (*os*)
→ (mapup-os-ibuffers (os-svc-receive-handler-path4 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-receive-handler-path4.

THEOREM: mapup-os-obuffers-os-svc-receive-handler-path4
good-os (*os*)
→ (mapup-os-obuffers (os-svc-receive-handler-path4 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-receive-handler-path4.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path4-support
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, 'tm), os-current-taskid (*os*), *os*)))))
→ (mapup-os-mbuffers (os-svc-receive-handler-path4 (*os*))
= deq2 (os-srcid (*os*, 'tm),
array-qfirst (os-readyq (*os*)),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path4-support.

THEOREM: mapup-os-mbuffers-os-svc-receive-handler-path4
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, 'tm), os-current-taskid (*os*), *os*)))))
→ (mapup-os-mbuffers (os-svc-receive-handler-path4 (*os*))
= deq2 (os-srcid (*os*, 'tm),
qfirst (mapup-queue (os-readyq (*os*))),
mapup-os-mbuffers (*os*)))

EVENT: Disable mapup-os-mbuffers-os-svc-receive-handler-path4.

THEOREM: mapup-queue-os-readyq-os-svc-receive-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-queue (os-readyq (os-svc-receive-handler-path4 (*os*)))
= mapup-queue (os-readyq (*os*)))

EVENT: Disable mapup-queue-os-readyq-os-svc-receive-handler-path4.

THEOREM: mapup-os-status-table-os-svc-receive-handler-path4

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table ('2, os-status-table (os-svc-receive-handler-path4 (*os*)))
= table ('2, os-status-table (*os*)))

EVENT: Disable mapup-os-status-table-os-svc-receive-handler-path4.

THEOREM: tm-clock-os-svc-receive-handler-path4
tm-clock (os-svc-receive-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-receive-handler-path4.

THEOREM: tm-iports-os-svc-receive-handler-path4
tm-iports (os-svc-receive-handler-path4 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-receive-handler-path4.

THEOREM: tm-oports-os-svc-receive-handler-path4
tm-oports (os-svc-receive-handler-path4 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-receive-handler-path4.

THEOREM: tm-rwstate-os-svc-receive-handler-path4
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-svc-receive-handler-path4 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-receive-handler-path4.

THEOREM: not-ak-waiting-to-sendp-mapup-os
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*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*)))))))
→ (¬ ak-waiting-to-sendp (os-srcid (*os*, '16),
os-current-taskid (*os*),
mapup-os (*os*)))

EVENT: Disable not-ak-waiting-to-sendp-mapup-os.

THEOREM: correctness-of-os-svc-receive-handler-path4
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '1)
    ∧ ((¬ array-qemptyp (os-current-mbuffer (os-srcid (*os*, '16), os-current-taskid (*os*), *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*)))))))))
→ (mapup-os (os-svc-receive-handler-path4 (*os*))
  = ak-execute-receive (os-srcid (*os*, '16), os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-receive-handler-path4.

THEOREM: correctness-of-os-svc-receive-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '1))))
→ (mapup-os (os-svc-receive-handler (*os*))
  = ak-execute-receive (os-srcid (*os*, '16), os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-receive-handler.

THEOREM: mapup-base-os-svc-tyi-handler-path1
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-tyi-handler-path1 (*os*))

$=$ mapup-base ($taskid$, os))

EVENT: Disable mapup-base-os-svc-tyi-handler-path1.

THEOREM: mapup-limit-os-svc-tyi-handler-path1
good-os (os)
$\rightarrow$ (mapup-limit ($taskid$, os-svc-tyi-handler-path1 (os))
$=$ mapup-limit ($taskid$, os))

EVENT: Disable mapup-limit-os-svc-tyi-handler-path1.

THEOREM: getseg-tm-memory-os-svc-tyi-handler-path1
(good-os (os) $\wedge$ ($taskid$ $<$ '16))
$\rightarrow$ (getseg (mapup-base ($taskid$, os),
mapup-limit ($taskid$, os),
tm-memory (os-svc-tyi-handler-path1 (os)))
$=$ getseg (mapup-base ($taskid$, os),
mapup-limit ($taskid$, os),
tm-memory (os)))

EVENT: Disable getseg-tm-memory-os-svc-tyi-handler-path1.

THEOREM: mapup-cpu-os-svc-tyi-handler-path1
(good-os (os)
$\wedge$ ($\neg$ tm-waiting (os)) $\wedge$ (($taskid$ $\in$ **N**) $\wedge$ ($taskid$ $<$ '16))))
$\rightarrow$ (mapup-cpu ($taskid$, os-svc-tyi-handler-path1 (os))
$=$ mapup-cpu ($taskid$, os))

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path1.

THEOREM: mapup-regs-os-svc-tyi-handler-path1
(good-os (os)
$\wedge$ ($\neg$ tm-waiting (os)) $\wedge$ (($taskid$ $\in$ **N**) $\wedge$ ($taskid$ $<$ '16))))
$\rightarrow$ (mapup-regs ($taskid$, os-svc-tyi-handler-path1 (os))
$=$ mapup-regs ($taskid$, os))

EVENT: Disable mapup-regs-os-svc-tyi-handler-path1.

THEOREM: mapup-cc-os-svc-tyi-handler-path1
(good-os (os)
$\wedge$ ($\neg$ tm-waiting (os)) $\wedge$ (($taskid$ $\in$ **N**) $\wedge$ ($taskid$ $<$ '16))))
$\rightarrow$ (mapup-cc ($taskid$, os-svc-tyi-handler-path1 (os))
$=$ mapup-cc ($taskid$, os))

EVENT: Disable mapup-cc-os-svc-tyi-handler-path1.

THEOREM: mapup-error-os-svc-tyi-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*) ∧ ((¬ tm-waiting (*os*)) ∧ (*taskid* ∈ **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

(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-os-tasks (os-svc-tyi-handler-path1 (*os*)) = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path1.

THEOREM: mapup-os-ibuffers-os-svc-tyi-handler-path1
good-os (*os*)
→ (mapup-os-ibuffers (os-svc-tyi-handler-path1 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-os-obuffers-os-svc-tyi-handler-path1
good-os (*os*)
→ (mapup-os-obuffers (os-svc-tyi-handler-path1 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-os-mbuffers-os-svc-tyi-handler-path1
good-os (*os*)
→ (mapup-os-mbuffers (os-svc-tyi-handler-path1 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-svc-tyi-handler-path1.

THEOREM: mapup-queue-os-readyq-os-svc-tyi-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-queue (os-readyq (os-svc-tyi-handler-path1 (*os*)))
= deq (mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyi-handler-path1.

THEOREM: mapup-os-status-table-os-svc-tyi-handler-path1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table (`'2`, os-status-table (os-svc-tyi-handler-path1 (*os*)))
= putnth (`'(5 0)`,
array-qfirst (os-readyq (*os*)),
table (`'2`, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-svc-tyi-handler-path1.

THEOREM: tm-rwstate-os-svc-tyi-handler-path1
tm-rwstate (os-svc-tyi-handler-path1 (*os*)) = '1

EVENT: Disable tm-rwstate-os-svc-tyi-handler-path1.

THEOREM: tm-clock-os-svc-tyi-handler-path1
tm-clock (os-svc-tyi-handler-path1 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-svc-tyi-handler-path1.

THEOREM: tm-iports-os-svc-tyi-handler-path1
tm-iports (os-svc-tyi-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyi-handler-path1.

THEOREM: tm-oports-os-svc-tyi-handler-path1
tm-oports (os-svc-tyi-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-tyi-handler-path1.

THEOREM: correctness-of-os-svc-tyi-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                          os))
     ∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ (mapup-os (os-svc-tyi-handler-path1 (*os*))
  = ak-execute-input (os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyi-handler-path1.

THEOREM: mapup-base-os-svc-tyi-handler-path2
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-tyi-handler-path2 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-tyi-handler-path2.

THEOREM: mapup-limit-os-svc-tyi-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-tyi-handler-path2 (*os*))
  = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyi-handler-path2.

THEOREM: getseg-tm-memory-os-svc-tyi-handler-path2
(good-os (*os*) ∧ (*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-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-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*)))

$$\begin{array}{lll} & \neq & \text{array-qfirst}\,(\text{os-readyq}\,(os))))))\\ \rightarrow & (\text{tm-cpu}\,(\text{os-svc-tyi-handler-path2}\,(os))\\ & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-tyi-handler-path2}\,(os)),\\ & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-tyi-handler-path2}\,(os)))))\end{array}$$

THEOREM: tm-cpu-os-svc-tyi-handler-path2

$$\begin{array}{lll} & & (\text{good-os}\,(os)\\ \wedge & & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge & (\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))))\\ \rightarrow & & (\text{tm-cpu}\,(\text{os-svc-tyi-handler-path2}\,(os))\\ & = & \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-tyi-handler-path2}\,(os)),\\ & & \quad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-tyi-handler-path2}\,(os)))))\end{array}$$

EVENT: Disable tm-cpu-os-svc-tyi-handler-path2.

THEOREM: mapup-cpu-os-svc-tyi-handler-path2

$$\begin{array}{lll} & & (\text{good-os}\,(os)\\ \wedge & & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & & (\text{mapup-cpu}\,(taskid,\ \text{os-svc-tyi-handler-path2}\,(os))\\ & = & \text{mapup-cpu}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path2.

THEOREM: mapup-regs-os-svc-tyi-handler-path2

$$\begin{array}{lll} & & (\text{good-os}\,(os)\\ \wedge & & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & & (\text{mapup-regs}\,(taskid,\ \text{os-svc-tyi-handler-path2}\,(os))\\ & = & \text{mapup-regs}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-regs-os-svc-tyi-handler-path2.

THEOREM: mapup-cc-os-svc-tyi-handler-path2

$$\begin{array}{lll} & & (\text{good-os}\,(os)\\ \wedge & & ((\neg\ \text{tm-waiting}\,(os))\\ & \wedge & ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\ & & \quad \wedge\ ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))\\ \rightarrow & & (\text{mapup-cc}\,(taskid,\ \text{os-svc-tyi-handler-path2}\,(os))\\ & = & \text{mapup-cc}\,(taskid,\ os))\end{array}$$

EVENT: Disable mapup-cc-os-svc-tyi-handler-path2.

THEOREM: mapup-error-os-svc-tyi-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))

$\wedge$ ((¬ array-qemptyp (array-deq (os-readyq (os))))
$\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-svc-tyi-handler-path2 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-svc-tyi-handler-path2.

THEOREM: mapup-os-tasks-os-svc-tyi-handler-path2
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (¬ array-qemptyp (array-deq (os-readyq (os))))))
$\rightarrow$ (mapup-os-tasks (os-svc-tyi-handler-path2 (os)) = mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-svc-tyi-handler-path2.

THEOREM: mapup-os-ibuffers-os-svc-tyi-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-ibuffers (os-svc-tyi-handler-path2 (os))
= mapup-os-ibuffers (os))

EVENT: Disable mapup-os-ibuffers-os-svc-tyi-handler-path2.

THEOREM: mapup-os-obuffers-os-svc-tyi-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-svc-tyi-handler-path2 (os))
= mapup-os-obuffers (os))

EVENT: Disable mapup-os-obuffers-os-svc-tyi-handler-path2.

THEOREM: mapup-os-mbuffers-os-svc-tyi-handler-path2
good-os (os)
$\rightarrow$ (mapup-os-mbuffers (os-svc-tyi-handler-path2 (os))
= mapup-os-mbuffers (os))

EVENT: Disable mapup-os-mbuffers-os-svc-tyi-handler-path2.

THEOREM: mapup-queue-os-readyq-os-svc-tyi-handler-path2
(good-os (os) $\wedge$ (¬ tm-waiting (os)))
$\rightarrow$ (mapup-queue (os-readyq (os-svc-tyi-handler-path2 (os)))
= deq (mapup-queue (os-readyq (os))))

EVENT: Disable mapup-queue-os-readyq-os-svc-tyi-handler-path2.

THEOREM: mapup-os-status-table-os-svc-tyi-handler-path2
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (table ('2, os-status-table (os-svc-tyi-handler-path2 (*os*)))
= putnth ('(5 0),
array-qfirst (os-readyq (*os*)),
table ('2, os-status-table (*os*))))

EVENT: Disable mapup-os-status-table-os-svc-tyi-handler-path2.

THEOREM: tm-clock-os-svc-tyi-handler-path2
tm-clock (os-svc-tyi-handler-path2 (*os*)) = '1000

EVENT: Disable tm-clock-os-svc-tyi-handler-path2.

THEOREM: tm-iports-os-svc-tyi-handler-path2
tm-iports (os-svc-tyi-handler-path2 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-svc-tyi-handler-path2.

THEOREM: tm-oports-os-svc-tyi-handler-path2
tm-oports (os-svc-tyi-handler-path2 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-svc-tyi-handler-path2.

THEOREM: tm-rwstate-os-svc-tyi-handler-path2
tm-rwstate (os-svc-tyi-handler-path2 (*os*)) = '0

EVENT: Disable tm-rwstate-os-svc-tyi-handler-path2.

THEOREM: correctness-of-os-svc-tyi-handler-path2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '3)
∧ (array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os))
∧ (¬ array-qemptyp (array-deq (os-readyq (*os*)))))))))
→ (mapup-os (os-svc-tyi-handler-path2 (*os*))
= ak-execute-input (os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-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*))

```
    =   putnth (array-qfirst (os-current-ibuffer (os-current-taskid (os),
                                                  os)),
                '3,
                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-tyi-handler-path3-case1.

THEOREM: mapup-cpu-os-svc-tyi-handler-path3-case2

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   ((taskid ∈ N)
           ∧   ((taskid < '16)
                ∧   (taskid ≠ array-qfirst (os-readyq (os)))))))
→   (mapup-cpu (taskid, os-svc-tyi-handler-path3 (os))
      =   mapup-cpu (taskid, os))
```

EVENT: Disable mapup-cpu-os-svc-tyi-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-tyi-handler-path3

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ ((taskid ∈ N) ∧ (taskid < '16))))
→   (mapup-cpu (taskid, os-svc-tyi-handler-path3 (os))
      =   if taskid = array-qfirst (os-readyq (os))
          then putnth (array-qfirst (os-current-ibuffer (os-current-taskid (os),
                                                         os)),
                       '3,
                       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-tyi-handler-path3.

THEOREM: mapup-regs-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-svc-tyi-handler-path3 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*))
 then putnth (array-qfirst (os-current-ibuffer (os-current-taskid (*os*),
 os)),
 '3,
 mapup-regs (*taskid*, *os*))
 else mapup-regs (*taskid*, *os*) **endif**)

EVENT: Disable mapup-regs-os-svc-tyi-handler-path3.

THEOREM: mapup-cc-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-svc-tyi-handler-path3 (*os*))
 = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-tyi-handler-path3.

THEOREM: mapup-error-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error (*taskid*, os-svc-tyi-handler-path3 (*os*))
 = mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-svc-tyi-handler-path3.

THEOREM: mapup-svcflag-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag (*taskid*, os-svc-tyi-handler-path3 (*os*))
 = **if** *taskid* = array-qfirst (os-readyq (*os*)) **then** '0
 else mapup-svcflag (*taskid*, *os*) **endif**)

EVENT: Disable mapup-svcflag-os-svc-tyi-handler-path3.

THEOREM: mapup-svcid-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-svc-tyi-handler-path3 (*os*))
 = mapup-svcid (*taskid*, *os*))

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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
os)))))
→ (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*)
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-svc-tyi-handler-path3 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-tyi-handler-path3.

THEOREM: correctness-of-os-svc-tyi-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '3)
    ∧ (¬ array-qemptyp (os-current-ibuffer (os-current-taskid (*os*),
                                            os)))))))
→ (mapup-os (os-svc-tyi-handler-path3 (*os*))
  = ak-execute-input (os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyi-handler-path3.

THEOREM: correctness-of-os-svc-tyi-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '3))))
→ (mapup-os (os-svc-tyi-handler (*os*))
  = ak-execute-input (os-current-taskid (*os*), mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyi-handler.

THEOREM: mapup-base-os-svc-tyo-handler-path1
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-tyo-handler-path1 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-tyo-handler-path1.

THEOREM: mapup-limit-os-svc-tyo-handler-path1
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-tyo-handler-path1 (*os*))
  = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyo-handler-path1.

THEOREM: getseg-tm-memory-os-svc-tyo-handler-path1
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
          mapup-limit (*taskid*, *os*),
          tm-memory (os-svc-tyo-handler-path1 (*os*)))
  = getseg (mapup-base (*taskid*, *os*),
            mapup-limit (*taskid*, *os*),
            tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path1.

THEOREM: mapup-cpu-os-svc-tyo-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcid (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path1.

THEOREM: mapup-task-os-svc-tyo-handler-path1
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-task (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-svc-tyo-handler-path1.

THEOREM: mapup-tasks-os-svc-tyo-handler-path1
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-svc-tyo-handler-path1 (*os*))
 = 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*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-os-tasks (os-svc-tyo-handler-path1 (*os*)) = 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*)
→ (mapup-os-ibuffers (os-svc-tyo-handler-path1 (*os*))
 = 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*)
→ (mapup-os-obuffers (os-svc-tyo-handler-path1 (*os*))
 = 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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (*id* < '16))
→ (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*) ∧ ('4 = getnth ('3, *queue*)))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((¬ tm-errorp (*os*))
∧ (((tm-svcid (*os*) **mod** '4) = '2)
∧ (array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os))
∧ array-qemptyp (array-deq (os-readyq (*os*))))))))
→ (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*)
→ (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*)
→ (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

$$
\begin{array}{l}
(\text{good-os}\,(os) \wedge (taskid < \texttt{'16})) \\
\rightarrow \quad (\text{getseg}\,(\text{mapup-base}\,(taskid,\ os), \\
\qquad\qquad \text{mapup-limit}\,(taskid,\ os), \\
\qquad\qquad \text{tm-memory}\,(\text{os-svc-tyo-handler-path2}\,(os))) \\
\qquad = \quad \text{getseg}\,(\text{mapup-base}\,(taskid,\ os), \\
\qquad\qquad\qquad \text{mapup-limit}\,(taskid,\ os), \\
\qquad\qquad\qquad \text{tm-memory}\,(os)))
\end{array}
$$

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-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-tyo-handler-path2-crock.

THEOREM: tm-cpu-os-svc-tyo-handler-path2-case1

$$
\begin{array}{ll}
 & (\text{good-os}\,(os)\\
\wedge & ((\neg\ \text{tm-waiting}\,(os))\\
 & \quad \wedge \quad (\text{array-qfirst}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))\\
 & \qquad = \quad \text{array-qfirst}\,(\text{os-readyq}\,(os)))))\\
\rightarrow & (\text{tm-cpu}\,(\text{os-svc-tyo-handler-path2}\,(os))\\
 & \quad = \quad \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-tyo-handler-path2}\,(os)),\\
 & \qquad\qquad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-tyo-handler-path2}\,(os)))))
\end{array}
$$

THEOREM: tm-cpu-os-svc-tyo-handler-path2-case2

$$
\begin{array}{ll}
 & (\text{good-os}\,(os)\\
\wedge & ((\neg\ \text{tm-waiting}\,(os))\\
 & \quad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os))))\\
 & \qquad \wedge \quad (\text{array-qfirst}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))\\
 & \qquad\quad \neq \quad \text{array-qfirst}\,(\text{os-readyq}\,(os))))))\\
\rightarrow & (\text{tm-cpu}\,(\text{os-svc-tyo-handler-path2}\,(os))\\
 & \quad = \quad \text{getnth}\,(\text{os-current-taskid}\,(\text{os-svc-tyo-handler-path2}\,(os)),\\
 & \qquad\qquad \text{table}\,(\texttt{'9},\ \text{os-task-table}\,(\text{os-svc-tyo-handler-path2}\,(os)))))
\end{array}
$$

THEOREM: tm-cpu-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*))))))
→ (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*)))))

EVENT: Disable tm-cpu-os-svc-tyo-handler-path2.

THEOREM: mapup-cpu-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
  ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cpu (*taskid*, os-svc-tyo-handler-path2 (*os*))
  = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path2.

THEOREM: mapup-regs-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
  ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-regs (*taskid*, os-svc-tyo-handler-path2 (*os*))
  = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-tyo-handler-path2.

THEOREM: mapup-cc-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ array-qemptyp (array-deq (os-readyq (*os*))))
  ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cc (*taskid*, os-svc-tyo-handler-path2 (*os*))
  = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-tyo-handler-path2.

THEOREM: mapup-error-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))

$$
\begin{array}{ll}
& \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
& \qquad \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-error}\,(taskid,\ \text{os-svc-tyo-handler-path2}\,(os)) \\
& \quad = \quad \text{mapup-error}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-error-os-svc-tyo-handler-path2.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path2

$$
\begin{array}{ll}
& (\text{good-os}\,(os) \\
& \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
& \qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
& \qquad\quad \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-svcflag}\,(taskid,\ \text{os-svc-tyo-handler-path2}\,(os)) \\
& \quad = \quad \text{mapup-svcflag}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path2.

THEOREM: mapup-svcid-os-svc-tyo-handler-path2

$$
\begin{array}{ll}
& (\text{good-os}\,(os) \\
& \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
& \qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
& \qquad\quad \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-svcid}\,(taskid,\ \text{os-svc-tyo-handler-path2}\,(os)) \\
& \quad = \quad \text{mapup-svcid}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path2.

THEOREM: mapup-task-os-svc-tyo-handler-path2

$$
\begin{array}{ll}
& (\text{good-os}\,(os) \\
& \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
& \qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
& \qquad\quad \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))) \\
\rightarrow & (\text{mapup-task}\,(taskid,\ \text{os-svc-tyo-handler-path2}\,(os)) \\
& \quad = \quad \text{mapup-task}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-task-os-svc-tyo-handler-path2.

THEOREM: mapup-tasks-os-svc-tyo-handler-path2

$$
\begin{array}{ll}
& (\text{good-os}\,(os) \\
& \wedge \quad ((\neg\ \text{tm-waiting}\,(os)) \\
& \qquad \wedge \quad ((\neg\ \text{array-qemptyp}\,(\text{array-deq}\,(\text{os-readyq}\,(os)))) \\
& \qquad\quad \wedge \quad (taskid \in \mathbf{N})))) \\
\rightarrow & (\text{mapup-tasks}\,(taskid,\ \text{os-svc-tyo-handler-path2}\,(os)) \\
& \quad = \quad \text{mapup-tasks}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path2.

THEOREM: mapup-os-tasks-os-svc-tyo-handler-path2
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qemptyp (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*) ∧ (¬ 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*) ∧ (¬ 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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*))
   ∧ (((tm-svcid (*os*) **mod** '4) = '2)
    ∧ (array-qfullp (os-current-obuffer (os-current-taskid (*os*),
                                         os))
     ∧ (¬ array-qemptyp (array-deq (os-readyq (*os*)))))))))
→ (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*)
→ (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

(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((*taskid* ∈ **N**)
   ∧ ((*taskid* < '16)
    ∧ (*taskid* ≠ array-qfirst (os-readyq (*os*)))))))
→ (mapup-cpu (*taskid*, os-svc-tyo-handler-path3 (*os*))
  = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-svc-tyo-handler-path3-case2.

THEOREM: mapup-cpu-os-svc-tyo-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cpu (*taskid*, os-svc-tyo-handler-path3 (*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-path3.

THEOREM: mapup-regs-os-svc-tyo-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-svc-tyo-handler-path3 (*os*))
  = mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-svc-tyo-handler-path3.

THEOREM: mapup-cc-os-svc-tyo-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-svc-tyo-handler-path3 (*os*))
  = mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-svc-tyo-handler-path3.

THEOREM: mapup-error-os-svc-tyo-handler-path3
(good-os (*os*)

$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-error $(taskid,$ os-svc-tyo-handler-path3 $(os))$
$=$ mapup-error $(taskid,\ os))$

EVENT: Disable mapup-error-os-svc-tyo-handler-path3.

THEOREM: mapup-svcflag-os-svc-tyo-handler-path3
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-svcflag $(taskid,$ os-svc-tyo-handler-path3 $(os))$
$=$ **if** $taskid =$ array-qfirst (os-readyq $(os))$ **then** '0
else mapup-svcflag $(taskid,\ os)$ **endif**)

EVENT: Disable mapup-svcflag-os-svc-tyo-handler-path3.

THEOREM: mapup-svcid-os-svc-tyo-handler-path3
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-svcid $(taskid,$ os-svc-tyo-handler-path3 $(os))$
$=$ mapup-svcid $(taskid,\ os))$

EVENT: Disable mapup-svcid-os-svc-tyo-handler-path3.

THEOREM: mapup-task-os-svc-tyo-handler-path3
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os)) \wedge ((taskid \in \mathbf{N}) \wedge (taskid <$ '16$))))$
$\rightarrow$ (mapup-task $(taskid,$ os-svc-tyo-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-tyo-handler-path3.

THEOREM: mapup-tasks-os-svc-tyo-handler-path3-case1
(good-os (os)
$\wedge$ $((\neg$ tm-waiting $(os))$
$\wedge$ $((taskid \in \mathbf{N}) \wedge (\text{array-qfirst (os-readyq } (os)) < taskid))))$
$\rightarrow$ (mapup-tasks $(taskid,$ os-svc-tyo-handler-path3 $(os))$
$=$ mapup-tasks $(taskid,\ os))$

EVENT: Disable mapup-tasks-os-svc-tyo-handler-path3-case1.

THEOREM: mapup-tasks-os-svc-tyo-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-tyo-handler-path3 (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-path3-case2.

```
THEOREM: mapup-os-tasks-os-svc-tyo-handler-path3-support
 (good-os (os) ∧ (¬ tm-waiting (os)))
→   (mapup-os-tasks (os-svc-tyo-handler-path3 (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-path3-support.

```
THEOREM: mapup-os-tasks-os-svc-tyo-handler-path3
 (good-os (os) ∧ (¬ tm-waiting (os)))
→   (mapup-os-tasks (os-svc-tyo-handler-path3 (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-path3.

```
THEOREM: mapup-os-ibuffers-os-svc-tyo-handler-path3
 good-os (os)
→   (mapup-os-ibuffers (os-svc-tyo-handler-path3 (os))
     =   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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (¬ array-qfullp (os-current-obuffer (os-current-taskid (*os*),
                                         os)))))
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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
tm-oports (os-svc-tyo-handler-path3 (*os*))
= 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
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-svc-tyo-handler-path3 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-svc-tyo-handler-path3.

THEOREM: correctness-of-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*)))))))
→ (mapup-os (os-svc-tyo-handler-path3 (*os*))
= ak-execute-output (os-message (*os*),
os-current-taskid (*os*),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyo-handler-path3.

THEOREM: mapup-base-os-svc-tyo-handler-path4
good-os (*os*)
→ (mapup-base (*taskid*, os-svc-tyo-handler-path4 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-svc-tyo-handler-path4.

THEOREM: mapup-limit-os-svc-tyo-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-svc-tyo-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-svc-tyo-handler-path4.

THEOREM: getseg-tm-memory-os-svc-tyo-handler-path4

(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-svc-tyo-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-svc-tyo-handler-path4.

THEOREM: tm-cpu-os-svc-tyo-handler-path4
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-cpu (os-svc-tyo-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-tyo-handler-path4.

THEOREM: mapup-cpu-os-svc-tyo-handler-path4-case1
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (mapup-cpu (array-qfirst (os-readyq (*os*)), os-svc-tyo-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-tyo-handler-path4-case1.

THEOREM: mapup-cpu-os-svc-tyo-handler-path4-case2
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ ((*taskid* ∈ **N**)
∧ ((*taskid* < '16)
∧ (*taskid* ≠ array-qfirst (os-readyq (*os*)))))))
→ (mapup-cpu (*taskid*, os-svc-tyo-handler-path4 (*os*))
= mapup-cpu (*taskid*, *os*))

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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
∧ ((¬ tm-waiting (*os*)) ∧ ((*taskid* ∈ **N**) ∧ (*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*)
∧ ((¬ tm-waiting (*os*))
∧ ((*taskid* ∈ **N**) ∧ (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*)
∧ ((¬ tm-waiting (*os*))
∧ ((*taskid* ∈ **N**)
∧ **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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*)
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (¬ array-qfullp (os-current-obuffer (os-current-taskid (*os*),
os)))))
→ (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*)
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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
(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*))))))))
→ (mapup-os (os-svc-tyo-handler-path4 (*os*))
  = ak-execute-output (os-message (*os*),
                       os-current-taskid (*os*),
                       mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyo-handler-path4.

THEOREM: correctness-of-os-svc-tyo-handler
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ ((¬ tm-errorp (*os*)) ∧ ((tm-svcid (*os*) **mod** '4) = '2))))
→ (mapup-os (os-svc-tyo-handler (*os*))
  = ak-execute-output (os-message (*os*),
                       os-current-taskid (*os*),
                       mapup-os (*os*)))

EVENT: Disable correctness-of-os-svc-tyo-handler.

THEOREM: mapup-base-os-waiting-input-handler-path1
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-input-handler-path1 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path1.

THEOREM: mapup-limit-os-waiting-input-handler-path1
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-input-handler-path1 (*os*))
  = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-input-handler-path1.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path1
(good-os (*os*) ∧ (*taskid* < '16))

$\rightarrow$ (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-IPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (GETNTH '1
            (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
    (CONS
     (GETNTH '2
             (GETSEG '0
                     '8
                     (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                             (TABLE '9 (OS-TASK-TABLE OS)))))
     (CONS
      (GETNTH '3
              (GETSEG '0
                      '8
                      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                              (TABLE '9 (OS-TASK-TABLE OS)))))
      (CONS
       (GETNTH '4
               (GETSEG '0
                       '8
```

```
                    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
          (CONS
           (GETNTH '5
                   (GETSEG '0
                           '8
                           (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                   (TABLE '9 (OS-TASK-TABLE OS)))))
           (CONS
            (GETNTH '6
                    (GETSEG '0
                            '8
                            (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                    (TABLE '9 (OS-TASK-TABLE OS)))))
            (CONS
             (GETNTH '7
                     (GETSEG '0
                             '8
                             (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                     (TABLE '9 (OS-TASK-TABLE OS)))))
             (CONS
              (TM-PACK-PSW
               (TM-UNPACK-CC
                (GETNTH (PLUS '8
                              (TIMES '9
                                     (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                        (OS-TASK-TABLE OS)))
               (TM-UNPACK-ERROR
                (GETNTH (PLUS '8
                              (TIMES '9
                                     (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                        (OS-TASK-TABLE OS)))
               (TM-UNPACK-SVCFLAG
                (GETNTH (PLUS '8
                              (TIMES '9
                                     (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                        (OS-TASK-TABLE OS)))
               (TM-UNPACK-SVCID
                (GETNTH (PLUS '8
                              (TIMES '9
                                     (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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*)
 ∧ (tm-waiting (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (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*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ (tm-waiting (*os*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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

(good-os (*os*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*)) ∧ (*taskid* ∈ **N**))))
→ (mapup-tasks (*taskid*, os-waiting-input-handler-path1 (*os*))
  = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-input-handler-path1.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path1
(good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (mapup-os-tasks (os-waiting-input-handler-path1 (*os*))
  = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path1.

THEOREM: mapup-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))))
→ (mapup-os-ibuffers (os-waiting-input-handler-path1 (*os*))
  = qreplace-ith-buffer (tm-overflow-char (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-path1.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path1
good-os (*os*)
→ (mapup-os-obuffers (os-waiting-input-handler-path1 (*os*))
  = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path1.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path1
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-input-handler-path1 (*os*))
  = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path1.

THEOREM: mapup-queue-os-readyq-os-waiting-input-handler-path1
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-queue (os-readyq (os-waiting-input-handler-path1 (*os*)))
= enq (tm-interrupting-input-port (tm-iports (*os*)),
mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-waiting-input-handler-path1.

THEOREM: mapup-os-status-table-os-waiting-input-handler-path1
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (table ('2, os-status-table (os-waiting-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-waiting-input-handler-path1.

THEOREM: tm-rwstate-os-waiting-input-handler-path1
tm-rwstate (os-waiting-input-handler-path1 (*os*)) = '0

EVENT: Disable tm-rwstate-os-waiting-input-handler-path1.

THEOREM: tm-clock-os-waiting-input-handler-path1
tm-clock (os-waiting-input-handler-path1 (*os*)) = '1000

EVENT: Disable tm-clock-os-waiting-input-handler-path1.

THEOREM: tm-iports-os-waiting-input-handler-path1
tm-iports (os-waiting-input-handler-path1 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))

EVENT: Disable tm-iports-os-waiting-input-handler-path1.

THEOREM: tm-oports-os-waiting-input-handler-path1
tm-oports (os-waiting-input-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-waiting-input-handler-path1.

THEOREM: ak-waiting-to-inputp-mapup-os
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))

$\wedge$ (getnth ('2 $*$ tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
= '5)))
$\rightarrow$ ak-waiting-to-inputp (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*))

EVENT: Disable ak-waiting-to-inputp-mapup-os.

THEOREM: correctness-of-array-qfullp-ibuffers-instance
(array-queuep (*queue*) $\wedge$ ('4 = getnth ('3, *queue*)))
$\rightarrow$ (array-qfullp (*queue*) = qfullp (mapup-queue (*queue*), '4))

EVENT: Disable correctness-of-array-qfullp-ibuffers-instance.

THEOREM: correctness-of-os-waiting-input-handler-path1
(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)
$\wedge$ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os))))))
$\rightarrow$ (mapup-os (os-waiting-input-handler-path1 (*os*))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler-path1.

THEOREM: mapup-base-os-waiting-input-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-base (*taskid*, os-waiting-input-handler-path2 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path2.

THEOREM: mapup-limit-os-waiting-input-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-limit (*taskid*, os-waiting-input-handler-path2 (*os*))
= 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*) ∧ (*taskid* < '16))
→ (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-IPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (GETNTH '1
            (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
    (CONS
     (GETNTH '2
             (GETSEG '0
                     '8
                     (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                             (TABLE '9 (OS-TASK-TABLE OS)))))
     (CONS
      (GETNTH '3
              (GETSEG '0
                      '8
                      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                              (TABLE '9 (OS-TASK-TABLE OS)))))
      (CONS
       (GETNTH '4
```

```
       (GETSEG '0
               '8
               (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                       (TABLE '9 (OS-TASK-TABLE OS)))))
(CONS
 (GETNTH '5
         (GETSEG '0
                 '8
                 (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                         (TABLE '9 (OS-TASK-TABLE OS)))))
 (CONS
  (GETNTH '6
          (GETSEG '0
                  '8
                  (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                          (TABLE '9 (OS-TASK-TABLE OS)))))
  (CONS
   (GETNTH '7
           (GETSEG '0
                   '8
                   (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (TM-PACK-PSW
     (TM-UNPACK-CC
      (GETNTH (PLUS '8
                    (TIMES '9
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              (OS-TASK-TABLE OS)))
     (TM-UNPACK-ERROR
      (GETNTH (PLUS '8
                    (TIMES '9
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              (OS-TASK-TABLE OS)))
     (TM-UNPACK-SVCFLAG
      (GETNTH (PLUS '8
                    (TIMES '9
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
              (OS-TASK-TABLE OS)))
     (TM-UNPACK-SVCID
      (GETNTH (PLUS '8
                    (TIMES '9
                           (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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
(good-os (*os*)
∧ (tm-waiting (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (tm-cpu (os-waiting-input-handler-path2 (*os*))
= getnth (os-current-taskid (os-waiting-input-handler-path2 (*os*)),
table ('9,
os-task-table (os-waiting-input-handler-path2 (*os*)))))

EVENT: Disable tm-cpu-os-waiting-input-handler-path2.

THEOREM: mapup-cpu-os-waiting-input-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cpu (*taskid*, os-waiting-input-handler-path2 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path2.

THEOREM: mapup-regs-os-waiting-input-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-regs (*taskid*, os-waiting-input-handler-path2 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-input-handler-path2.

THEOREM: mapup-cc-os-waiting-input-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))

$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad$ (mapup-cc ($taskid$, os-waiting-input-handler-path2 (os))
$= \quad$ mapup-cc ($taskid$, os))

EVENT: Disable mapup-cc-os-waiting-input-handler-path2.

THEOREM: mapup-error-os-waiting-input-handler-path2
(good-os (os)
$\wedge \quad$ (tm-waiting (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad$ (mapup-error ($taskid$, os-waiting-input-handler-path2 (os))
$= \quad$ mapup-error ($taskid$, os))

EVENT: Disable mapup-error-os-waiting-input-handler-path2.

THEOREM: mapup-svcflag-os-waiting-input-handler-path2
(good-os (os)
$\wedge \quad$ (tm-waiting (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad$ (mapup-svcflag ($taskid$, os-waiting-input-handler-path2 (os))
$= \quad$ mapup-svcflag ($taskid$, os))

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path2.

THEOREM: mapup-svcid-os-waiting-input-handler-path2
(good-os (os)
$\wedge \quad$ (tm-waiting (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad$ (mapup-svcid ($taskid$, os-waiting-input-handler-path2 (os))
$= \quad$ mapup-svcid ($taskid$, os))

EVENT: Disable mapup-svcid-os-waiting-input-handler-path2.

THEOREM: mapup-task-os-waiting-input-handler-path2
(good-os (os)
$\wedge \quad$ (tm-waiting (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad$ (mapup-task ($taskid$, os-waiting-input-handler-path2 (os))
$= \quad$ mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-waiting-input-handler-path2.

THEOREM: mapup-tasks-os-waiting-input-handler-path2
(good-os (*os*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-input-interruptp (tm-iports (*os*)) ∧ (*taskid* ∈ **N**))))
→ (mapup-tasks (*taskid*, os-waiting-input-handler-path2 (*os*))
  = mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-input-handler-path2.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path2
(good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (mapup-os-tasks (os-waiting-input-handler-path2 (*os*))
  = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path2.

THEOREM: mapup-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)))))
→ (mapup-os-ibuffers (os-waiting-input-handler-path2 (*os*))
  = enq-ith-buffer (tm-overflow-char (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-path2.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path2
good-os (*os*)
→ (mapup-os-obuffers (os-waiting-input-handler-path2 (*os*))
  = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path2.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path2
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-input-handler-path2 (*os*))
  = mapup-os-mbuffers (*os*))

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
(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-waiting-input-handler-path2 (*os*))
  = ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
    mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler-path2.

THEOREM: mapup-base-os-waiting-input-handler-path3
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-input-handler-path3 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path3.

THEOREM: mapup-limit-os-waiting-input-handler-path3
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-input-handler-path3 (*os*))
  = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-input-handler-path3.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path3
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
   mapup-limit (*taskid*, *os*),
   tm-memory (os-waiting-input-handler-path3 (*os*)))
  = getseg (mapup-base (*taskid*, *os*),
    mapup-limit (*taskid*, *os*),
    tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path3.

```
(PROVE-LEMMA
 TM-CPU-OS-WAITING-INPUT-HANDLER-PATH3-CROCK
 (REWRITE)
 (EQUAL
  (TM-CPU (OS-WAITING-INPUT-HANDLER-PATH3 OS))
  (CONS
   (GETNTH '0
           (GETSEG '0
                   '8
                   (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (GETNTH '1
            (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
    (CONS
     (GETNTH '2
             (GETSEG '0
                     '8
                     (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                             (TABLE '9 (OS-TASK-TABLE OS)))))
     (CONS
      (GETNTH '3
              (GETSEG '0
                      '8
                      (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                              (TABLE '9 (OS-TASK-TABLE OS)))))
      (CONS
       (GETNTH '4
               (GETSEG '0
                       '8
                       (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                               (TABLE '9 (OS-TASK-TABLE OS)))))
       (CONS
        (GETNTH '5
                (GETSEG '0
                        '8
                        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                (TABLE '9 (OS-TASK-TABLE OS)))))
        (CONS
```

```
        (GETNTH '6
                (GETSEG '0
                        '8
                        (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                (TABLE '9 (OS-TASK-TABLE OS)))))
        (CONS
         (GETNTH '7
                 (GETSEG '0
                         '8
                         (GETNTH (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))
                                 (TABLE '9 (OS-TASK-TABLE OS)))))
         (CONS
          (TM-PACK-PSW
           (TM-UNPACK-CC
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                    (OS-TASK-TABLE OS)))
           (TM-UNPACK-ERROR
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                    (OS-TASK-TABLE OS)))
           (TM-UNPACK-SVCFLAG
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS OS))))
                    (OS-TASK-TABLE OS)))
           (TM-UNPACK-SVCID
            (GETNTH (PLUS '8
                          (TIMES '9
                                 (TM-INTERRUPTING-INPUT-PORT (TM-IPORTS 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*)

$\wedge$ (tm-waiting (*os*) $\wedge$ tm-some-input-interruptp (tm-iports (*os*))))
$\rightarrow$ (tm-cpu (os-waiting-input-handler-path3 (*os*))
= getnth (os-current-taskid (os-waiting-input-handler-path3 (*os*)),
table ('9,
os-task-table (os-waiting-input-handler-path3 (*os*)))))

EVENT: Disable tm-cpu-os-waiting-input-handler-path3.

THEOREM: mapup-cpu-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-cpu (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path3.

THEOREM: mapup-regs-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-regs (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-input-handler-path3.

THEOREM: mapup-cc-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-cc (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-input-handler-path3.

THEOREM: mapup-error-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))

$\rightarrow$ (mapup-error (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-input-handler-path3.

THEOREM: mapup-svcflag-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-svcflag (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path3.

THEOREM: mapup-svcid-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-svcid (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-input-handler-path3.

THEOREM: mapup-task-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*))
$\wedge$ ((*taskid* $\in$ **N**) $\wedge$ (*taskid* $<$ '16)))))
$\rightarrow$ (mapup-task (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-input-handler-path3.

THEOREM: mapup-tasks-os-waiting-input-handler-path3
(good-os (*os*)
$\wedge$ (tm-waiting (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*)) $\wedge$ (*taskid* $\in$ **N**))))
$\rightarrow$ (mapup-tasks (*taskid*, os-waiting-input-handler-path3 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-input-handler-path3.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path3
(good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*))))
→ (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*)
 ∧ (tm-some-input-interruptp (tm-iports (*os*))
 ∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
 os)))))
→ (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*)
→ (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*)
→ (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*) ∧ tm-waiting (*os*))
→ (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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (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*)
∧ (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-waiting-input-handler-path3 (*os*))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler-path3.

THEOREM: mapup-base-os-waiting-input-handler-path4
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-input-handler-path4 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path4.

THEOREM: mapup-limit-os-waiting-input-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-input-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-input-handler-path4.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-input-handler-path4 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path4.

THEOREM: mapup-cpu-os-waiting-input-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cpu (*taskid*, os-waiting-input-handler-path4 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path4.

THEOREM: mapup-regs-os-waiting-input-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-waiting-input-handler-path4 (*os*))
= 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
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-os-tasks (os-waiting-input-handler-path4 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path4.

THEOREM: mapup-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))))
→ (mapup-os-ibuffers (os-waiting-input-handler-path4 (*os*))
= qreplace-ith-buffer (tm-overflow-char (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-path4.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path4
good-os (*os*)
→ (mapup-os-obuffers (os-waiting-input-handler-path4 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path4.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path4
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-input-handler-path4 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path4.

THEOREM: tm-rwstate-os-waiting-input-handler-path4
tm-rwstate (os-waiting-input-handler-path4 (*os*)) = '1

EVENT: Disable tm-rwstate-os-waiting-input-handler-path4.

THEOREM: tm-clock-os-waiting-input-handler-path4
tm-clock (os-waiting-input-handler-path4 (*os*)) = 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

good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-input-handler-path5 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path5.

THEOREM: mapup-limit-os-waiting-input-handler-path5
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-input-handler-path5 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-input-handler-path5.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path5
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-input-handler-path5 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path5.

THEOREM: mapup-cpu-os-waiting-input-handler-path5
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cpu (*taskid*, os-waiting-input-handler-path5 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path5.

THEOREM: mapup-regs-os-waiting-input-handler-path5
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-waiting-input-handler-path5 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-input-handler-path5.

THEOREM: mapup-cc-os-waiting-input-handler-path5
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-waiting-input-handler-path5 (*os*))
= 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 mapup-os-tasks-os-waiting-input-handler-path5.

THEOREM: mapup-os-ibuffers-os-waiting-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)))))
→ (mapup-os-ibuffers (os-waiting-input-handler-path5 (*os*))
  = enq-ith-buffer (tm-overflow-char (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-path5.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path5
good-os (*os*)
→ (mapup-os-obuffers (os-waiting-input-handler-path5 (*os*))
  = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path5.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path5
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-input-handler-path5 (*os*))
  = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path5.

THEOREM: tm-rwstate-os-waiting-input-handler-path5
tm-rwstate (os-waiting-input-handler-path5 (*os*)) = '1

EVENT: Disable tm-rwstate-os-waiting-input-handler-path5.

THEOREM: tm-clock-os-waiting-input-handler-path5
tm-clock (os-waiting-input-handler-path5 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-waiting-input-handler-path5.

THEOREM: tm-iports-os-waiting-input-handler-path5
tm-iports (os-waiting-input-handler-path5 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
  tm-iports (*os*))

EVENT: Disable tm-iports-os-waiting-input-handler-path5.

THEOREM: tm-oports-os-waiting-input-handler-path5
tm-oports (os-waiting-input-handler-path5 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-waiting-input-handler-path5.

THEOREM: correctness-of-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*)))))))
→ (mapup-os (os-waiting-input-handler-path5 (*os*))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler-path5.

THEOREM: mapup-base-os-waiting-input-handler-path6
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-input-handler-path6.

THEOREM: mapup-limit-os-waiting-input-handler-path6
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-input-handler-path6.

THEOREM: getseg-tm-memory-os-waiting-input-handler-path6
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),

mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-input-handler-path6 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-input-handler-path6.

THEOREM: mapup-cpu-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-cpu (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-input-handler-path6.

THEOREM: mapup-regs-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-regs (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-input-handler-path6.

THEOREM: mapup-cc-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-cc (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-input-handler-path6.

THEOREM: mapup-error-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-error (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-input-handler-path6.

THEOREM: mapup-svcflag-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcflag (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-input-handler-path6.

THEOREM: mapup-svcid-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcid (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-input-handler-path6.

THEOREM: mapup-task-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-task (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-input-handler-path6.

THEOREM: mapup-tasks-os-waiting-input-handler-path6
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-waiting-input-handler-path6 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-input-handler-path6.

THEOREM: mapup-os-tasks-os-waiting-input-handler-path6
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-os-tasks (os-waiting-input-handler-path6 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-input-handler-path6.

THEOREM: mapup-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)))))
→ (mapup-os-ibuffers (os-waiting-input-handler-path6 (*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-path6.

THEOREM: mapup-os-obuffers-os-waiting-input-handler-path6

good-os (*os*)
→ (mapup-os-obuffers (os-waiting-input-handler-path6 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-input-handler-path6.

THEOREM: mapup-os-mbuffers-os-waiting-input-handler-path6
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-input-handler-path6 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-input-handler-path6.

THEOREM: tm-rwstate-os-waiting-input-handler-path6
tm-rwstate (os-waiting-input-handler-path6 (*os*)) = '1

EVENT: Disable tm-rwstate-os-waiting-input-handler-path6.

THEOREM: tm-clock-os-waiting-input-handler-path6
tm-clock (os-waiting-input-handler-path6 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-waiting-input-handler-path6.

THEOREM: tm-iports-os-waiting-input-handler-path6
tm-iports (os-waiting-input-handler-path6 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
tm-iports (*os*))

EVENT: Disable tm-iports-os-waiting-input-handler-path6.

THEOREM: tm-oports-os-waiting-input-handler-path6
tm-oports (os-waiting-input-handler-path6 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-waiting-input-handler-path6.

THEOREM: correctness-of-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*))

$\neq$ '5)
$\wedge$ ((¬ 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*)))))))
$\rightarrow$ (mapup-os (os-waiting-input-handler-path6 (*os*))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)), mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler-path6.

THEOREM: correctness-of-os-waiting-input-handler
(good-os (*os*)
$\wedge$ (tm-some-input-interruptp (tm-iports (*os*)) $\wedge$ tm-waiting (*os*)))
$\rightarrow$ (mapup-os (os-waiting-input-handler (*os*))
= ak-waiting-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)), mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-input-handler.

THEOREM: mapup-base-os-running-input-handler-path1
good-os (*os*)
$\rightarrow$ (mapup-base (*taskid*, os-running-input-handler-path1 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-input-handler-path1.

THEOREM: mapup-limit-os-running-input-handler-path1
good-os (*os*)
$\rightarrow$ (mapup-limit (*taskid*, os-running-input-handler-path1 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-input-handler-path1.

THEOREM: getseg-tm-memory-os-running-input-handler-path1
(good-os (*os*) $\wedge$ (*taskid* < '16))
$\rightarrow$ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-input-handler-path1 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-input-handler-path1.

THEOREM: tm-cpu-os-running-input-handler-path1-crock
tm-cpu (os-running-input-handler-path1 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path1-crock.

THEOREM: tm-cpu-os-running-input-handler-path1
(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))))
→ (tm-cpu (os-running-input-handler-path1 (*os*))
  = getnth (os-current-taskid (os-running-input-handler-path1 (*os*)),
   table ('9,
    os-task-table (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((getnth ('2
    ∗ tm-interrupting-input-port (tm-iports (*os*)),
    os-status-table (*os*))
   = '5)
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cpu (*taskid*, os-running-input-handler-path1 (*os*))
  = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-input-handler-path1.

THEOREM: mapup-regs-os-running-input-handler-path1
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-regs (*taskid*, os-running-input-handler-path1 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-input-handler-path1.

THEOREM: mapup-cc-os-running-input-handler-path1
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cc (*taskid*, os-running-input-handler-path1 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-input-handler-path1.

THEOREM: mapup-error-os-running-input-handler-path1
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-error (*taskid*, os-running-input-handler-path1 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-input-handler-path1.

THEOREM: mapup-svcflag-os-running-input-handler-path1
(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*))

$$
\begin{array}{lll}
 & = & \texttt{'5}) \\
 & \wedge & ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-svcflag}\,(\mathit{taskid},\ \text{os-running-input-handler-path1}\,(\mathit{os})) \\
 & = & \text{mapup-svcflag}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path1.

THEOREM: mapup-svcid-os-running-input-handler-path1

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \quad = \quad \texttt{'5}) \\
 & & \quad \wedge \quad ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-svcid}\,(\mathit{taskid},\ \text{os-running-input-handler-path1}\,(\mathit{os})) \\
 & = & \text{mapup-svcid}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-svcid-os-running-input-handler-path1.

THEOREM: mapup-task-os-running-input-handler-path1

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \quad = \quad \texttt{'5}) \\
 & & \quad \wedge \quad ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-task}\,(\mathit{taskid},\ \text{os-running-input-handler-path1}\,(\mathit{os})) \\
 & = & \text{mapup-task}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-task-os-running-input-handler-path1.

THEOREM: mapup-tasks-os-running-input-handler-path1

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \quad = \quad \texttt{'5})
\end{array}
$$

$\wedge \quad (taskid \in \mathbf{N})))))$
$\rightarrow \quad$ (mapup-tasks ($taskid$, os-running-input-handler-path1 (os))
$= \quad$ mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-running-input-handler-path1.

THEOREM: mapup-os-tasks-os-running-input-handler-path1
(good-os (os)
$\wedge \quad$ ((¬ tm-waiting (os))
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad$ (getnth ('2
$*$ tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
$= \quad$ '5))))
$\rightarrow \quad$ (mapup-os-tasks (os-running-input-handler-path1 (os))
$= \quad$ mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-running-input-handler-path1.

THEOREM: mapup-os-ibuffers-os-running-input-handler-path1
(good-os (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad$ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))
$\rightarrow \quad$ (mapup-os-ibuffers (os-running-input-handler-path1 (os))
$= \quad$ qreplace-ith-buffer (tm-overflow-char (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-running-input-handler-path1.

THEOREM: mapup-os-obuffers-os-running-input-handler-path1
good-os (os)
$\rightarrow \quad$ (mapup-os-obuffers (os-running-input-handler-path1 (os))
$= \quad$ mapup-os-obuffers (os))

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path1.

THEOREM: mapup-os-mbuffers-os-running-input-handler-path1
good-os (os)
$\rightarrow \quad$ (mapup-os-mbuffers (os-running-input-handler-path1 (os))
$= \quad$ mapup-os-mbuffers (os))

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path1.

THEOREM: mapup-queue-os-readyq-os-running-input-handler-path1
(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-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*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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
tm-oports (os-running-input-handler-path1 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-running-input-handler-path1.

THEOREM: correctness-of-os-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))))))
→ (mapup-os (os-running-input-handler-path1 (*os*))
  = ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
    mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler-path1.

THEOREM: mapup-base-os-running-input-handler-path2
good-os (*os*)
→ (mapup-base (*taskid*, os-running-input-handler-path2 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-input-handler-path2.

THEOREM: mapup-limit-os-running-input-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-input-handler-path2 (*os*))
  = mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-input-handler-path2.

THEOREM: getseg-tm-memory-os-running-input-handler-path2
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
    mapup-limit (*taskid*, *os*),
    tm-memory (os-running-input-handler-path2 (*os*)))
  = getseg (mapup-base (*taskid*, *os*),
    mapup-limit (*taskid*, *os*),
    tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-input-handler-path2.

THEOREM: tm-cpu-os-running-input-handler-path2-crock
tm-cpu (os-running-input-handler-path2 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path2-crock.

THEOREM: tm-cpu-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))))
→ (tm-cpu (os-running-input-handler-path2 (*os*))
  = getnth (os-current-taskid (os-running-input-handler-path2 (*os*)),
   table ('9,
    os-task-table (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-input-interruptp (tm-iports (*os*))
   ∧ ((getnth ('2
    ∗ tm-interrupting-input-port (tm-iports (*os*)),
    os-status-table (*os*))
   = '5)
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cpu (*taskid*, os-running-input-handler-path2 (*os*))
  = mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-input-handler-path2.

THEOREM: mapup-regs-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-regs (*taskid*, os-running-input-handler-path2 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-input-handler-path2.

THEOREM: mapup-cc-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cc (*taskid*, os-running-input-handler-path2 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-input-handler-path2.

THEOREM: mapup-error-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-error (*taskid*, os-running-input-handler-path2 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-input-handler-path2.

THEOREM: mapup-svcflag-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*))

$$
\begin{array}{lll}
 & = & \texttt{'5}) \\
 & \wedge & ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-svcflag}\,(\mathit{taskid},\ \text{os-running-input-handler-path2}\,(\mathit{os})) \\
 & = & \text{mapup-svcflag}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path2.

THEOREM: mapup-svcid-os-running-input-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad\qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \qquad = \quad \texttt{'5}) \\
 & & \qquad \wedge \quad ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-svcid}\,(\mathit{taskid},\ \text{os-running-input-handler-path2}\,(\mathit{os})) \\
 & = & \text{mapup-svcid}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-svcid-os-running-input-handler-path2.

THEOREM: mapup-task-os-running-input-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad\qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \qquad = \quad \texttt{'5}) \\
 & & \qquad \wedge \quad ((\mathit{taskid} \in \mathbf{N}) \wedge (\mathit{taskid} < \texttt{'16}))))))) \\
\rightarrow & (\text{mapup-task}\,(\mathit{taskid},\ \text{os-running-input-handler-path2}\,(\mathit{os})) \\
 & = & \text{mapup-task}\,(\mathit{taskid},\ \mathit{os}))
\end{array}
$$

EVENT: Disable mapup-task-os-running-input-handler-path2.

THEOREM: mapup-tasks-os-running-input-handler-path2

$$
\begin{array}{lll}
(\text{good-os}\,(\mathit{os}) \\
\wedge & ((\neg\ \text{tm-waiting}\,(\mathit{os})) \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(\mathit{os})) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & \qquad\qquad * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(\mathit{os})), \\
 & & \qquad\qquad \text{os-status-table}\,(\mathit{os})) \\
 & & \qquad = \quad \texttt{'5})
\end{array}
$$

$\wedge \quad (taskid \in \mathbf{N})))))$
$\rightarrow \quad$ (mapup-tasks $(taskid,$ os-running-input-handler-path2 $(os))$
$= \quad$ mapup-tasks $(taskid,\ os))$

EVENT: Disable mapup-tasks-os-running-input-handler-path2.

THEOREM: mapup-os-tasks-os-running-input-handler-path2
(good-os (os)
$\wedge \quad ((\neg$ tm-waiting $(os))$
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad$ (getnth ('2
$*$ tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
$= \quad$ '5))))
$\rightarrow \quad$ (mapup-os-tasks (os-running-input-handler-path2 (os))
$= \quad$ mapup-os-tasks $(os))$

EVENT: Disable mapup-os-tasks-os-running-input-handler-path2.

THEOREM: mapup-os-ibuffers-os-running-input-handler-path2
(good-os (os)
$\wedge \quad$ (tm-some-input-interruptp (tm-iports (os))
$\wedge \quad (\neg$ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
$os)))))$
$\rightarrow \quad$ (mapup-os-ibuffers (os-running-input-handler-path2 (os))
$= \quad$ enq-ith-buffer (tm-overflow-char (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-running-input-handler-path2.

THEOREM: mapup-os-obuffers-os-running-input-handler-path2
good-os (os)
$\rightarrow \quad$ (mapup-os-obuffers (os-running-input-handler-path2 (os))
$= \quad$ mapup-os-obuffers $(os))$

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path2.

THEOREM: mapup-os-mbuffers-os-running-input-handler-path2
good-os (os)
$\rightarrow \quad$ (mapup-os-mbuffers (os-running-input-handler-path2 (os))
$= \quad$ mapup-os-mbuffers $(os))$

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
tm-oports (os-running-input-handler-path2 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-running-input-handler-path2.

THEOREM: correctness-of-os-running-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*)),
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-path2 (*os*))
= ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler-path2.

THEOREM: mapup-base-os-running-input-handler-path3
good-os (*os*)
→ (mapup-base (*taskid*, os-running-input-handler-path3 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-input-handler-path3.

THEOREM: mapup-limit-os-running-input-handler-path3
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-input-handler-path3 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-input-handler-path3.

THEOREM: getseg-tm-memory-os-running-input-handler-path3
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-input-handler-path3 (*os*)))
= getseg (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (getnth ('2
∗ tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
= '5))))
→ (tm-cpu (os-running-input-handler-path3 (*os*))
= getnth (os-current-taskid (os-running-input-handler-path3 (*os*)),
table ('9,
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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((getnth ('2
∗ tm-interrupting-input-port (tm-iports (*os*)),
os-status-table (*os*))
= '5)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))

$\wedge$ ((getnth ('2
$*$ tm-interrupting-input-port (tm-iports (os)),
os-status-table (os))
$=$ '5)
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16))))))
$\rightarrow$ (mapup-regs ($taskid$, os-running-input-handler-path3 (os))
$=$ mapup-regs ($taskid$, os))

EVENT: Disable mapup-regs-os-running-input-handler-path3.

THEOREM: mapup-cc-os-running-input-handler-path3
(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)
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16))))))
$\rightarrow$ (mapup-cc ($taskid$, os-running-input-handler-path3 (os))
$=$ mapup-cc ($taskid$, os))

EVENT: Disable mapup-cc-os-running-input-handler-path3.

THEOREM: mapup-error-os-running-input-handler-path3
(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)
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16))))))
$\rightarrow$ (mapup-error ($taskid$, os-running-input-handler-path3 (os))
$=$ mapup-error ($taskid$, os))

EVENT: Disable mapup-error-os-running-input-handler-path3.

THEOREM: mapup-svcflag-os-running-input-handler-path3
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os))
$\wedge$ (tm-some-input-interruptp (tm-iports (os))
$\wedge$ ((getnth ('2

$$
\begin{array}{lll}
 & & * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
 & & \text{os-status-table}\,(os)) \\
 & = & \texttt{'5}) \\
 & \wedge & ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16})))))) \\
\rightarrow & (\text{mapup-svcflag}\,(taskid, \text{os-running-input-handler-path3}\,(os)) & \\
 & = & \text{mapup-svcflag}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-svcflag-os-running-input-handler-path3.

THEOREM: mapup-svcid-os-running-input-handler-path3

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os)) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
 & & \text{os-status-table}\,(os)) \\
 & = & \texttt{'5}) \\
 & \wedge & ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16})))))) \\
\rightarrow & (\text{mapup-svcid}\,(taskid, \text{os-running-input-handler-path3}\,(os)) & \\
 & = & \text{mapup-svcid}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-svcid-os-running-input-handler-path3.

THEOREM: mapup-task-os-running-input-handler-path3

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os)) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)), \\
 & & \text{os-status-table}\,(os)) \\
 & = & \texttt{'5}) \\
 & \wedge & ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16})))))) \\
\rightarrow & (\text{mapup-task}\,(taskid, \text{os-running-input-handler-path3}\,(os)) & \\
 & = & \text{mapup-task}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-task-os-running-input-handler-path3.

THEOREM: mapup-tasks-os-running-input-handler-path3

$$
\begin{array}{lll}
(\text{good-os}\,(os) & & \\
\wedge & ((\neg\ \text{tm-waiting}\,(os)) & \\
 & \wedge & (\text{tm-some-input-interruptp}\,(\text{tm-iports}\,(os)) \\
 & & \wedge \quad ((\text{getnth}\,(\texttt{'2} \\
 & & * \quad \text{tm-interrupting-input-port}\,(\text{tm-iports}\,(os)),
\end{array}
$$

os-status-table (*os*))
= '5)
∧ (*taskid* ∈ **N**)))))
→ (mapup-tasks (*taskid*, os-running-input-handler-path3 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-running-input-handler-path3.

THEOREM: mapup-os-tasks-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-os-tasks (os-running-input-handler-path3 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-running-input-handler-path3.

THEOREM: mapup-os-ibuffers-os-running-input-handler-path3
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ (¬ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (*os*)),
os)))))
→ (mapup-os-ibuffers (os-running-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-running-input-handler-path3.

THEOREM: mapup-os-obuffers-os-running-input-handler-path3
good-os (*os*)
→ (mapup-os-obuffers (os-running-input-handler-path3 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path3.

THEOREM: mapup-os-mbuffers-os-running-input-handler-path3
good-os (*os*)

→ (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*))

EVENT: Disable tm-iports-os-running-input-handler-path3.

THEOREM: tm-oports-os-running-input-handler-path3
tm-oports (os-running-input-handler-path3 (*os*)) = tm-oports (*os*)

EVENT: Disable tm-oports-os-running-input-handler-path3.

THEOREM: correctness-of-os-running-input-handler-path3
(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-path3 (*os*))
= ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler-path3.

THEOREM: mapup-base-os-running-input-handler-path4
good-os (*os*)
→ (mapup-base (*taskid*, os-running-input-handler-path4 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-input-handler-path4.

THEOREM: mapup-limit-os-running-input-handler-path4
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-input-handler-path4 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-input-handler-path4.

THEOREM: getseg-tm-memory-os-running-input-handler-path4
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),

```
            mapup-limit (taskid, os),
            tm-memory (os-running-input-handler-path4 (os)))
    =   getseg (mapup-base (taskid, os),
                mapup-limit (taskid, os),
                tm-memory (os)))
```

EVENT: Disable getseg-tm-memory-os-running-input-handler-path4.

THEOREM: tm-cpu-os-running-input-handler-path4-crock
tm-cpu (os-running-input-handler-path4 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path4-crock.

THEOREM: tm-cpu-os-running-input-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ tm-some-input-interruptp (tm-iports (os))))
→   (tm-cpu (os-running-input-handler-path4 (os))
     =   getnth (os-current-taskid (os-running-input-handler-path4 (os)),
                 table ('9,
                        os-task-table (os-running-input-handler-path4 (os)))))
```

EVENT: Disable tm-cpu-os-running-input-handler-path4.

THEOREM: mapup-cpu-os-running-input-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   (tm-some-input-interruptp (tm-iports (os))
           ∧   ((taskid ∈ N) ∧ (taskid < '16)))))
→   (mapup-cpu (taskid, os-running-input-handler-path4 (os))
     =   mapup-cpu (taskid, os))
```

EVENT: Disable mapup-cpu-os-running-input-handler-path4.

THEOREM: mapup-regs-os-running-input-handler-path4

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   (tm-some-input-interruptp (tm-iports (os))
           ∧   ((taskid ∈ N) ∧ (taskid < '16)))))
→   (mapup-regs (taskid, os-running-input-handler-path4 (os))
     =   mapup-regs (taskid, os))
```

EVENT: Disable mapup-regs-os-running-input-handler-path4.

THEOREM: mapup-cc-os-running-input-handler-path4
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))

$\wedge$ (tm-some-input-interruptp (tm-iports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-task ($taskid$, os-running-input-handler-path4 (os))
= mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-running-input-handler-path4.

THEOREM: mapup-tasks-os-running-input-handler-path4
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os))
$\wedge$ (tm-some-input-interruptp (tm-iports (os)) $\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-running-input-handler-path4 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-running-input-handler-path4.

THEOREM: mapup-os-tasks-os-running-input-handler-path4
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os)) $\wedge$ tm-some-input-interruptp (tm-iports (os))))
$\rightarrow$ (mapup-os-tasks (os-running-input-handler-path4 (os))
= mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-running-input-handler-path4.

THEOREM: mapup-os-ibuffers-os-running-input-handler-path4
(good-os (os)
$\wedge$ (tm-some-input-interruptp (tm-iports (os))
$\wedge$ array-qfullp (os-current-ibuffer (tm-interrupting-input-port (tm-iports (os)),
os))))
$\rightarrow$ (mapup-os-ibuffers (os-running-input-handler-path4 (os))
= qreplace-ith-buffer (tm-overflow-char (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-running-input-handler-path4.

THEOREM: mapup-os-obuffers-os-running-input-handler-path4
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-running-input-handler-path4 (os))
= mapup-os-obuffers (os))

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path4.

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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ 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-path5 (os))
= mapup-svcflag ($taskid$, os))

EVENT: Disable mapup-svcflag-os-running-input-handler-path5.

THEOREM: mapup-svcid-os-running-input-handler-path5
(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-svcid ($taskid$, os-running-input-handler-path5 (os))
= mapup-svcid ($taskid$, os))

EVENT: Disable mapup-svcid-os-running-input-handler-path5.

THEOREM: mapup-task-os-running-input-handler-path5
(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-task ($taskid$, os-running-input-handler-path5 (os))
= mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-running-input-handler-path5.

THEOREM: mapup-tasks-os-running-input-handler-path5
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-input-interruptp (tm-iports (os)) $\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-running-input-handler-path5 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-running-input-handler-path5.

THEOREM: mapup-os-tasks-os-running-input-handler-path5
(good-os (os)
$\wedge$ ((¬ tm-waiting (os)) $\wedge$ tm-some-input-interruptp (tm-iports (os))))
$\rightarrow$ (mapup-os-tasks (os-running-input-handler-path5 (os))
= mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-running-input-handler-path5.

THEOREM: mapup-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)))))
→ (mapup-os-ibuffers (os-running-input-handler-path5 (*os*))
  = enq-ith-buffer (tm-overflow-char (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-running-input-handler-path5.

THEOREM: mapup-os-obuffers-os-running-input-handler-path5
good-os (*os*)
→ (mapup-os-obuffers (os-running-input-handler-path5 (*os*))
  = mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path5.

THEOREM: mapup-os-mbuffers-os-running-input-handler-path5
good-os (*os*)
→ (mapup-os-mbuffers (os-running-input-handler-path5 (*os*))
  = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-running-input-handler-path5.

THEOREM: tm-rwstate-os-running-input-handler-path5
(good-tm (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-rwstate (os-running-input-handler-path5 (*os*)) = tm-rwstate (*os*))

EVENT: Disable tm-rwstate-os-running-input-handler-path5.

THEOREM: tm-clock-os-running-input-handler-path5
tm-clock (os-running-input-handler-path5 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-running-input-handler-path5.

THEOREM: tm-iports-os-running-input-handler-path5
tm-iports (os-running-input-handler-path5 (*os*))
= tm-clear-input-interrupt (tm-interrupting-input-port (tm-iports (*os*)),
  tm-iports (*os*))

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*),

```
            mapup-limit (taskid, os),
            tm-memory (os-running-input-handler-path6 (os)))
    =   getseg (mapup-base (taskid, os),
                mapup-limit (taskid, os),
                tm-memory (os)))
```

EVENT: Disable getseg-tm-memory-os-running-input-handler-path6.

THEOREM: tm-cpu-os-running-input-handler-path6-crock
tm-cpu (os-running-input-handler-path6 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-input-handler-path6-crock.

THEOREM: tm-cpu-os-running-input-handler-path6

```
(good-os (os)
 ∧   ((¬ tm-waiting (os)) ∧ tm-some-input-interruptp (tm-iports (os))))
→   (tm-cpu (os-running-input-handler-path6 (os))
     =   getnth (os-current-taskid (os-running-input-handler-path6 (os)),
                 table ('9,
                        os-task-table (os-running-input-handler-path6 (os)))))
```

EVENT: Disable tm-cpu-os-running-input-handler-path6.

THEOREM: mapup-cpu-os-running-input-handler-path6

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   (tm-some-input-interruptp (tm-iports (os))
           ∧   ((taskid ∈ N) ∧ (taskid < '16)))))
→   (mapup-cpu (taskid, os-running-input-handler-path6 (os))
     =   mapup-cpu (taskid, os))
```

EVENT: Disable mapup-cpu-os-running-input-handler-path6.

THEOREM: mapup-regs-os-running-input-handler-path6

```
(good-os (os)
 ∧   ((¬ tm-waiting (os))
      ∧   (tm-some-input-interruptp (tm-iports (os))
           ∧   ((taskid ∈ N) ∧ (taskid < '16)))))
→   (mapup-regs (taskid, os-running-input-handler-path6 (os))
     =   mapup-regs (taskid, os))
```

EVENT: Disable mapup-regs-os-running-input-handler-path6.

THEOREM: mapup-cc-os-running-input-handler-path6
(good-os (*os*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))
∧ (tm-some-input-interruptp (tm-iports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
∧ ((¬ tm-waiting (*os*))

$\wedge$ (tm-some-input-interruptp (tm-iports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-task ($taskid$, os-running-input-handler-path6 (os))
= mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-running-input-handler-path6.

THEOREM: mapup-tasks-os-running-input-handler-path6
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os))
$\wedge$ (tm-some-input-interruptp (tm-iports (os)) $\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-running-input-handler-path6 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-running-input-handler-path6.

THEOREM: mapup-os-tasks-os-running-input-handler-path6
(good-os (os)
$\wedge$ (($\neg$ tm-waiting (os)) $\wedge$ tm-some-input-interruptp (tm-iports (os))))
$\rightarrow$ (mapup-os-tasks (os-running-input-handler-path6 (os))
= mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-running-input-handler-path6.

THEOREM: mapup-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)),
os)))))
$\rightarrow$ (mapup-os-ibuffers (os-running-input-handler-path6 (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-running-input-handler-path6.

THEOREM: mapup-os-obuffers-os-running-input-handler-path6
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-running-input-handler-path6 (os))
= mapup-os-obuffers (os))

EVENT: Disable mapup-os-obuffers-os-running-input-handler-path6.

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*))))))))

→ (mapup-os (os-running-input-handler-path6 (*os*))
= ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler-path6.

THEOREM: correctness-of-os-running-input-handler
(good-os (*os*)
∧ (tm-some-input-interruptp (tm-iports (*os*)) ∧ (¬ tm-waiting (*os*))))
→ (mapup-os (os-running-input-handler (*os*))
= ak-running-input-interrupt-handler (tm-interrupting-input-port (tm-iports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-input-handler.

THEOREM: mapup-base-os-waiting-output-handler-path1
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-output-handler-path1 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-output-handler-path1.

THEOREM: mapup-limit-os-waiting-output-handler-path1
good-os (*os*)
→ (mapup-limit (*taskid*, os-waiting-output-handler-path1 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-output-handler-path1.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path1
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-output-handler-path1 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

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-OPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (GETNTH '1
            (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
    (CONS
     (GETNTH '2
             (GETSEG '0
                     '8
                     (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TABLE '9 (OS-TASK-TABLE OS)))))
     (CONS
      (GETNTH '3
              (GETSEG '0
                      '8
                      (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                              (TABLE '9 (OS-TASK-TABLE OS)))))
      (CONS
       (GETNTH '4
               (GETSEG '0
                       '8
                       (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                               (TABLE '9 (OS-TASK-TABLE OS)))))
       (CONS
        (GETNTH '5
                (GETSEG '0
                        '8
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                (TABLE '9 (OS-TASK-TABLE OS)))))
        (CONS
         (GETNTH '6
                 (GETSEG '0
```

```
                    '8
                    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
          (CONS
           (GETNTH '7
                   (GETSEG '0
                           '8
                           (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                   (TABLE '9 (OS-TASK-TABLE OS)))))
           (CONS
            (TM-PACK-PSW
             (TM-UNPACK-CC
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-ERROR
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-SVCFLAG
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-SVCID
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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*)

$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-error ($taskid$, os-waiting-output-handler-path1 (os))
= mapup-error ($taskid$, os))

EVENT: Disable mapup-error-os-waiting-output-handler-path1.

THEOREM: mapup-svcflag-os-waiting-output-handler-path1
(good-os (os)
$\wedge$ (tm-waiting (os)
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-svcflag ($taskid$, os-waiting-output-handler-path1 (os))
= mapup-svcflag ($taskid$, os))

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path1.

THEOREM: mapup-svcid-os-waiting-output-handler-path1
(good-os (os)
$\wedge$ (tm-waiting (os)
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-svcid ($taskid$, os-waiting-output-handler-path1 (os))
= mapup-svcid ($taskid$, os))

EVENT: Disable mapup-svcid-os-waiting-output-handler-path1.

THEOREM: mapup-task-os-waiting-output-handler-path1
(good-os (os)
$\wedge$ (tm-waiting (os)
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-task ($taskid$, os-waiting-output-handler-path1 (os))
= mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-waiting-output-handler-path1.

THEOREM: mapup-tasks-os-waiting-output-handler-path1
(good-os (os)
$\wedge$ (tm-waiting (os)
$\wedge$ (tm-some-output-interruptp (tm-oports (os)) $\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-waiting-output-handler-path1 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-waiting-output-handler-path1.

THEOREM: mapup-os-tasks-os-waiting-output-handler-path1
(good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*))))
→ (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*)
→ (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*)
→ (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*)
→ (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*) ∧ tm-waiting (*os*))
→ (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*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (table ('2, os-status-table (os-waiting-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-waiting-output-handler-path1.

THEOREM: tm-rwstate-os-waiting-output-handler-path1
tm-rwstate (os-waiting-output-handler-path1 (*os*)) = '0

EVENT: Disable tm-rwstate-os-waiting-output-handler-path1.

THEOREM: tm-clock-os-waiting-output-handler-path1
tm-clock (os-waiting-output-handler-path1 (*os*)) = '1000

EVENT: Disable tm-clock-os-waiting-output-handler-path1.

THEOREM: tm-iports-os-waiting-output-handler-path1
tm-iports (os-waiting-output-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-waiting-output-handler-path1.

THEOREM: tm-oports-os-waiting-output-handler-path1
tm-oports (os-waiting-output-handler-path1 (*os*))
= tm-clear-output-interrupt (tm-interrupting-output-port (tm-oports (*os*)),
tm-oports (*os*))

EVENT: Disable tm-oports-os-waiting-output-handler-path1.

THEOREM: ak-waiting-to-outputp-mapup-os
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (getnth ('2 ∗ tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
= '4)))
→ ak-waiting-to-outputp (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*))

EVENT: Disable ak-waiting-to-outputp-mapup-os.

THEOREM: correctness-of-os-waiting-output-handler-path1
(good-os (*os*)
∧ (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*))
= '4)
$\wedge$ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os))))))
$\rightarrow$ (mapup-os (os-waiting-output-handler-path1 (*os*))
= ak-waiting-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-output-handler-path1.

THEOREM: mapup-base-os-waiting-output-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-base (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-waiting-output-handler-path2.

THEOREM: mapup-limit-os-waiting-output-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-limit (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-waiting-output-handler-path2.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path2
(good-os (*os*) $\wedge$ (*taskid* $<$ '16))
$\rightarrow$ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-waiting-output-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-waiting-output-handler-path2.

```
(PROVE-LEMMA
 TM-CPU-OS-WAITING-OUTPUT-HANDLER-PATH2-CROCK
 (REWRITE)
```

```
(EQUAL
 (TM-CPU (OS-WAITING-OUTPUT-HANDLER-PATH2 OS))
 (CONS
  (GETNTH '0
          (GETSEG '0
                  '8
                  (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                          (TABLE '9 (OS-TASK-TABLE OS)))))
  (CONS
   (GETNTH '1
           (GETSEG '0
                   '8
                   (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                           (TABLE '9 (OS-TASK-TABLE OS)))))
   (CONS
    (GETNTH '2
            (GETSEG '0
                    '8
                    (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                            (TABLE '9 (OS-TASK-TABLE OS)))))
    (CONS
     (GETNTH '3
             (GETSEG '0
                     '8
                     (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                             (TABLE '9 (OS-TASK-TABLE OS)))))
     (CONS
      (GETNTH '4
              (GETSEG '0
                      '8
                      (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                              (TABLE '9 (OS-TASK-TABLE OS)))))
      (CONS
       (GETNTH '5
               (GETSEG '0
                       '8
                       (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                               (TABLE '9 (OS-TASK-TABLE OS)))))
       (CONS
        (GETNTH '6
                (GETSEG '0
                        '8
                        (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                (TABLE '9 (OS-TASK-TABLE OS)))))
```

```
          (CONS
           (GETNTH '7
                   (GETSEG '0
                           '8
                           (GETNTH (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))
                                   (TABLE '9 (OS-TASK-TABLE OS)))))
           (CONS
            (TM-PACK-PSW
             (TM-UNPACK-CC
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-ERROR
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-SVCFLAG
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS OS))))
                     (OS-TASK-TABLE OS)))
             (TM-UNPACK-SVCID
              (GETNTH
                     (PLUS '8
                           (TIMES '9
                                  (TM-INTERRUPTING-OUTPUT-PORT (TM-OPORTS 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*)
 ∧ (tm-waiting (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*))))

→ (tm-cpu (os-waiting-output-handler-path2 (*os*))
= getnth (os-current-taskid (os-waiting-output-handler-path2 (*os*)),
table ('9,
os-task-table (os-waiting-output-handler-path2 (*os*)))))

EVENT: Disable tm-cpu-os-waiting-output-handler-path2.

THEOREM: mapup-cpu-os-waiting-output-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cpu (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-waiting-output-handler-path2.

THEOREM: mapup-regs-os-waiting-output-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-regs (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-output-handler-path2.

THEOREM: mapup-cc-os-waiting-output-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-cc (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-output-handler-path2.

THEOREM: mapup-error-os-waiting-output-handler-path2
(good-os (*os*)
∧ (tm-waiting (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (mapup-error (*taskid*, os-waiting-output-handler-path2 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-output-handler-path2.

THEOREM: mapup-svcflag-os-waiting-output-handler-path2
(good-os (*os*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ (tm-waiting (*os*)
  ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ (*taskid* ∈ **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

(good-os (*os*)
 ∧ (tm-waiting (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*))))
→ (mapup-os-tasks (os-waiting-output-handler-path2 (*os*))
 = mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path2.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path2
good-os (*os*)
→ (mapup-os-ibuffers (os-waiting-output-handler-path2 (*os*))
 = mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path2.

THEOREM: mapup-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)))))
→ (mapup-os-obuffers (os-waiting-output-handler-path2 (*os*))
 = deq-ith-buffer (tm-interrupting-output-port (tm-oports (*os*)),
 mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path2.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path2
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-output-handler-path2 (*os*))
 = mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path2.

THEOREM: mapup-queue-os-readyq-os-waiting-output-handler-path2
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-queue (os-readyq (os-waiting-output-handler-path2 (*os*)))
 = enq (tm-interrupting-output-port (tm-oports (*os*)),
 mapup-queue (os-readyq (*os*))))

EVENT: Disable mapup-queue-os-readyq-os-waiting-output-handler-path2.

THEOREM: mapup-os-status-table-os-waiting-output-handler-path2
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))

→ (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*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))
→ (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*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (tm-waiting (*os*)
∧ ((getnth ('2
∗ tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))

$$
\begin{array}{lll}
 & = & \texttt{'4}) \\
 & \wedge & (\neg\ \text{array-qemptyp}\,(\text{os-current-obuffer}\,(\text{tm-interrupting-output-port}\,(\text{tm-oports}\,(os)), \\
 & & \qquad os))))))) \\
\rightarrow & (\text{mapup-os}\,(\text{os-waiting-output-handler-path2}\,(os)) \\
 & = & \text{ak-waiting-output-interrupt-handler}\,(\text{tm-interrupting-output-port}\,(\text{tm-oports}\,(os)), \\
 & & \qquad \text{mapup-os}\,(os)))
\end{array}
$$

EVENT: Disable correctness-of-os-waiting-output-handler-path2.

THEOREM: mapup-base-os-waiting-output-handler-path3

$$
\begin{array}{lll}
 & \text{good-os}\,(os) & \\
\rightarrow & (\text{mapup-base}\,(taskid,\ \text{os-waiting-output-handler-path3}\,(os)) & \\
 & = & \text{mapup-base}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-base-os-waiting-output-handler-path3.

THEOREM: mapup-limit-os-waiting-output-handler-path3

$$
\begin{array}{lll}
 & \text{good-os}\,(os) & \\
\rightarrow & (\text{mapup-limit}\,(taskid,\ \text{os-waiting-output-handler-path3}\,(os)) & \\
 & = & \text{mapup-limit}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-limit-os-waiting-output-handler-path3.

THEOREM: getseg-tm-memory-os-waiting-output-handler-path3

$$
\begin{array}{lll}
 & (\text{good-os}\,(os) \wedge (taskid < \texttt{'16})) & \\
\rightarrow & (\text{getseg}\,(\text{mapup-base}\,(taskid,\ os), & \\
 & \qquad \text{mapup-limit}\,(taskid,\ os), & \\
 & \qquad \text{tm-memory}\,(\text{os-waiting-output-handler-path3}\,(os))) & \\
 & = & \text{getseg}\,(\text{mapup-base}\,(taskid,\ os), \\
 & & \qquad \text{mapup-limit}\,(taskid,\ os), \\
 & & \qquad \text{tm-memory}\,(os)))
\end{array}
$$

EVENT: Disable getseg-tm-memory-os-waiting-output-handler-path3.

THEOREM: mapup-cpu-os-waiting-output-handler-path3

$$
\begin{array}{lll}
 & (\text{good-os}\,(os) \wedge (\text{tm-waiting}\,(os) \wedge ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16})))) & \\
\rightarrow & (\text{mapup-cpu}\,(taskid,\ \text{os-waiting-output-handler-path3}\,(os)) & \\
 & = & \text{mapup-cpu}\,(taskid,\ os))
\end{array}
$$

EVENT: Disable mapup-cpu-os-waiting-output-handler-path3.

THEOREM: mapup-regs-os-waiting-output-handler-path3

(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-regs (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-waiting-output-handler-path3.

THEOREM: mapup-cc-os-waiting-output-handler-path3
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-cc (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-waiting-output-handler-path3.

THEOREM: mapup-error-os-waiting-output-handler-path3
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-error (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-output-handler-path3.

THEOREM: mapup-svcflag-os-waiting-output-handler-path3
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcflag (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path3.

THEOREM: mapup-svcid-os-waiting-output-handler-path3
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-svcid (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-output-handler-path3.

THEOREM: mapup-task-os-waiting-output-handler-path3
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))
→ (mapup-task (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-output-handler-path3.

THEOREM: mapup-tasks-os-waiting-output-handler-path3

(good-os (*os*) ∧ (tm-waiting (*os*) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-waiting-output-handler-path3 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-output-handler-path3.

THEOREM: mapup-os-tasks-os-waiting-output-handler-path3
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-os-tasks (os-waiting-output-handler-path3 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path3.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path3
good-os (*os*)
→ (mapup-os-ibuffers (os-waiting-output-handler-path3 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path3.

THEOREM: mapup-os-obuffers-os-waiting-output-handler-path3
good-os (*os*)
→ (mapup-os-obuffers (os-waiting-output-handler-path3 (*os*))
= mapup-os-obuffers (*os*))

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path3.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path3
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-output-handler-path3 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path3.

THEOREM: tm-rwstate-os-waiting-output-handler-path3
tm-rwstate (os-waiting-output-handler-path3 (*os*)) = '1

EVENT: Disable tm-rwstate-os-waiting-output-handler-path3.

THEOREM: tm-clock-os-waiting-output-handler-path3
tm-clock (os-waiting-output-handler-path3 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-waiting-output-handler-path3.

THEOREM: tm-iports-os-waiting-output-handler-path3
tm-iports (os-waiting-output-handler-path3 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-waiting-output-handler-path3.

THEOREM: tm-oports-os-waiting-output-handler-path3
tm-oports (os-waiting-output-handler-path3 (*os*))
= tm-clear-output-interrupt (tm-interrupting-output-port (tm-oports (*os*)),
tm-oports (*os*))

EVENT: Disable tm-oports-os-waiting-output-handler-path3.

THEOREM: not-ak-waiting-to-outputp-mapup-os
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (getnth ('2 ∗ tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
≠ '4)))
→ (¬ ak-waiting-to-outputp (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

EVENT: Disable not-ak-waiting-to-outputp-mapup-os.

THEOREM: correctness-of-os-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))))))
→ (mapup-os (os-waiting-output-handler-path3 (*os*))
= ak-waiting-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-waiting-output-handler-path3.

THEOREM: mapup-base-os-waiting-output-handler-path4
good-os (*os*)
→ (mapup-base (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-base (*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

(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-error (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-waiting-output-handler-path4.

THEOREM: mapup-svcflag-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcflag (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-svcflag (*taskid*, *os*))

EVENT: Disable mapup-svcflag-os-waiting-output-handler-path4.

THEOREM: mapup-svcid-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-svcid (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-svcid (*taskid*, *os*))

EVENT: Disable mapup-svcid-os-waiting-output-handler-path4.

THEOREM: mapup-task-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16'))))
→ (mapup-task (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-task (*taskid*, *os*))

EVENT: Disable mapup-task-os-waiting-output-handler-path4.

THEOREM: mapup-tasks-os-waiting-output-handler-path4
(good-os (*os*) ∧ (tm-waiting (*os*) ∧ (*taskid* ∈ **N**)))
→ (mapup-tasks (*taskid*, os-waiting-output-handler-path4 (*os*))
= mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-waiting-output-handler-path4.

THEOREM: mapup-os-tasks-os-waiting-output-handler-path4
(good-os (*os*) ∧ tm-waiting (*os*))
→ (mapup-os-tasks (os-waiting-output-handler-path4 (*os*))
= mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-waiting-output-handler-path4.

THEOREM: mapup-os-ibuffers-os-waiting-output-handler-path4

good-os (*os*)
→ (mapup-os-ibuffers (os-waiting-output-handler-path4 (*os*))
= mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-waiting-output-handler-path4.

THEOREM: mapup-os-obuffers-os-waiting-output-handler-path4
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))
∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))
→ (mapup-os-obuffers (os-waiting-output-handler-path4 (*os*))
= deq-ith-buffer (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-waiting-output-handler-path4.

THEOREM: mapup-os-mbuffers-os-waiting-output-handler-path4
good-os (*os*)
→ (mapup-os-mbuffers (os-waiting-output-handler-path4 (*os*))
= mapup-os-mbuffers (*os*))

EVENT: Disable mapup-os-mbuffers-os-waiting-output-handler-path4.

THEOREM: tm-rwstate-os-waiting-output-handler-path4
tm-rwstate (os-waiting-output-handler-path4 (*os*)) = '1

EVENT: Disable tm-rwstate-os-waiting-output-handler-path4.

THEOREM: tm-clock-os-waiting-output-handler-path4
tm-clock (os-waiting-output-handler-path4 (*os*)) = tm-clock (*os*)

EVENT: Disable tm-clock-os-waiting-output-handler-path4.

THEOREM: tm-iports-os-waiting-output-handler-path4
tm-iports (os-waiting-output-handler-path4 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-waiting-output-handler-path4.

THEOREM: tm-oports-os-waiting-output-handler-path4
(good-os (*os*)
∧ (tm-some-output-interruptp (tm-oports (*os*))

```
      ∧   (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (os)),
                                                os)))))
→     (tm-oports (os-waiting-output-handler-path4 (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-path4.

```
THEOREM: correctness-of-os-waiting-output-handler-path4
 (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-waiting-output-handler-path4 (os))
      =   ak-waiting-output-interrupt-handler (tm-interrupting-output-port (tm-oports (os)),
                                               mapup-os (os)))
```

EVENT: Disable correctness-of-os-waiting-output-handler-path4.

```
THEOREM: correctness-of-os-waiting-output-handler
 (good-os (os)
  ∧   (tm-some-output-interruptp (tm-oports (os)) ∧ tm-waiting (os)))
→     (mapup-os (os-waiting-output-handler (os))
      =   ak-waiting-output-interrupt-handler (tm-interrupting-output-port (tm-oports (os)),
                                               mapup-os (os)))
```

EVENT: Disable correctness-of-os-waiting-output-handler.

```
THEOREM: mapup-base-os-running-output-handler-path1
 good-os (os)
→     (mapup-base (taskid, os-running-output-handler-path1 (os))
      =   mapup-base (taskid, os))
```

EVENT: Disable mapup-base-os-running-output-handler-path1.

THEOREM: mapup-limit-os-running-output-handler-path1

good-os (*os*)
→ (mapup-limit (*taskid*, os-running-output-handler-path1 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-output-handler-path1.

THEOREM: getseg-tm-memory-os-running-output-handler-path1
(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-output-handler-path1 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-output-handler-path1.

THEOREM: tm-cpu-os-running-output-handler-path1-crock
tm-cpu (os-running-output-handler-path1 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-output-handler-path1-crock.

THEOREM: tm-cpu-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))))
→ (tm-cpu (os-running-output-handler-path1 (*os*))
= getnth (os-current-taskid (os-running-output-handler-path1 (*os*)),
table ('9,
os-task-table (os-running-output-handler-path1 (*os*)))))

EVENT: Disable tm-cpu-os-running-output-handler-path1.

THEOREM: mapup-cpu-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cpu (*taskid*, os-running-output-handler-path1 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-output-handler-path1.

THEOREM: mapup-regs-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-regs (*taskid*, os-running-output-handler-path1 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-output-handler-path1.

THEOREM: mapup-cc-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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cc (*taskid*, os-running-output-handler-path1 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-output-handler-path1.

THEOREM: mapup-error-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*))

$$
\begin{array}{lll}
 & = & \texttt{'4}) \\
 & \wedge & ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))))) \\
\rightarrow & \text{(mapup-error (}taskid\text{, os-running-output-handler-path1 (}os\text{))} & \\
 & = & \text{mapup-error (}taskid\text{, }os\text{))}
\end{array}
$$

EVENT: Disable mapup-error-os-running-output-handler-path1.

THEOREM: mapup-svcflag-os-running-output-handler-path1

$$
\begin{array}{lll}
\text{(good-os (}os\text{)} & & \\
\wedge & \text{((}\neg\text{ tm-waiting (}os\text{))} & \\
 & \wedge & \text{(tm-some-output-interruptp (tm-oports (}os\text{))} \\
 & & \wedge \quad \text{((getnth (}\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-output-port (tm-oports (}os\text{)),} \\
 & & \qquad \text{os-status-table (}os\text{))} \\
 & & = \quad \texttt{'4}) \\
 & & \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))))) \\
\rightarrow & \text{(mapup-svcflag (}taskid\text{, os-running-output-handler-path1 (}os\text{))} & \\
 & = & \text{mapup-svcflag (}taskid\text{, }os\text{))}
\end{array}
$$

EVENT: Disable mapup-svcflag-os-running-output-handler-path1.

THEOREM: mapup-svcid-os-running-output-handler-path1

$$
\begin{array}{lll}
\text{(good-os (}os\text{)} & & \\
\wedge & \text{((}\neg\text{ tm-waiting (}os\text{))} & \\
 & \wedge & \text{(tm-some-output-interruptp (tm-oports (}os\text{))} \\
 & & \wedge \quad \text{((getnth (}\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-output-port (tm-oports (}os\text{)),} \\
 & & \qquad \text{os-status-table (}os\text{))} \\
 & & = \quad \texttt{'4}) \\
 & & \wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))))) \\
\rightarrow & \text{(mapup-svcid (}taskid\text{, os-running-output-handler-path1 (}os\text{))} & \\
 & = & \text{mapup-svcid (}taskid\text{, }os\text{))}
\end{array}
$$

EVENT: Disable mapup-svcid-os-running-output-handler-path1.

THEOREM: mapup-task-os-running-output-handler-path1

$$
\begin{array}{lll}
\text{(good-os (}os\text{)} & & \\
\wedge & \text{((}\neg\text{ tm-waiting (}os\text{))} & \\
 & \wedge & \text{(tm-some-output-interruptp (tm-oports (}os\text{))} \\
 & & \wedge \quad \text{((getnth (}\texttt{'2} \\
 & & \qquad * \quad \text{tm-interrupting-output-port (tm-oports (}os\text{)),} \\
 & & \qquad \text{os-status-table (}os\text{))} \\
 & & = \quad \texttt{'4})
\end{array}
$$

$\wedge$ $((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))))$
$\rightarrow$ (mapup-task ($taskid$, os-running-output-handler-path1 (os))
= mapup-task ($taskid$, os))

EVENT: Disable mapup-task-os-running-output-handler-path1.

THEOREM: mapup-tasks-os-running-output-handler-path1
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ ((getnth ('2
$*$ tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
= '4)
$\wedge$ ($taskid \in \mathbf{N}$)))))
$\rightarrow$ (mapup-tasks ($taskid$, os-running-output-handler-path1 (os))
= mapup-tasks ($taskid$, os))

EVENT: Disable mapup-tasks-os-running-output-handler-path1.

THEOREM: mapup-os-tasks-os-running-output-handler-path1
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (getnth ('2
$*$ tm-interrupting-output-port (tm-oports (os)),
os-status-table (os))
= '4))))
$\rightarrow$ (mapup-os-tasks (os-running-output-handler-path1 (os))
= mapup-os-tasks (os))

EVENT: Disable mapup-os-tasks-os-running-output-handler-path1.

THEOREM: mapup-os-ibuffers-os-running-output-handler-path1
good-os (os)
$\rightarrow$ (mapup-os-ibuffers (os-running-output-handler-path1 (os))
= mapup-os-ibuffers (os))

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path1.

THEOREM: mapup-os-obuffers-os-running-output-handler-path1
good-os (os)
$\rightarrow$ (mapup-os-obuffers (os-running-output-handler-path1 (os))
= mapup-os-obuffers (os))

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
tm-iports (os-running-output-handler-path1 (*os*)) = tm-iports (*os*)

EVENT: Disable tm-iports-os-running-output-handler-path1.

THEOREM: tm-oports-os-running-output-handler-path1
tm-oports (os-running-output-handler-path1 (*os*))
= tm-clear-output-interrupt (tm-interrupting-output-port (tm-oports (*os*)),
tm-oports (*os*))

EVENT: Disable tm-oports-os-running-output-handler-path1.

THEOREM: correctness-of-os-running-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))))))
→ (mapup-os (os-running-output-handler-path1 (*os*))
= ak-running-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-output-handler-path1.

THEOREM: mapup-base-os-running-output-handler-path2
good-os (*os*)
→ (mapup-base (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-output-handler-path2.

THEOREM: mapup-limit-os-running-output-handler-path2
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-limit (*taskid*, *os*))

EVENT: Disable mapup-limit-os-running-output-handler-path2.

THEOREM: getseg-tm-memory-os-running-output-handler-path2

(good-os (*os*) ∧ (*taskid* < '16))
→ (getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (os-running-output-handler-path2 (*os*)))
= getseg (mapup-base (*taskid*, *os*),
mapup-limit (*taskid*, *os*),
tm-memory (*os*)))

EVENT: Disable getseg-tm-memory-os-running-output-handler-path2.

THEOREM: tm-cpu-os-running-output-handler-path2-crock
tm-cpu (os-running-output-handler-path2 (*os*)) = tm-cpu (*os*)

EVENT: Disable tm-cpu-os-running-output-handler-path2-crock.

THEOREM: tm-cpu-os-running-output-handler-path2
(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))))
→ (tm-cpu (os-running-output-handler-path2 (*os*))
= getnth (os-current-taskid (os-running-output-handler-path2 (*os*)),
table ('9,
os-task-table (os-running-output-handler-path2 (*os*)))))

EVENT: Disable tm-cpu-os-running-output-handler-path2.

THEOREM: mapup-cpu-os-running-output-handler-path2
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cpu (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-cpu (*taskid*, *os*))

EVENT: Disable mapup-cpu-os-running-output-handler-path2.

THEOREM: mapup-regs-os-running-output-handler-path2
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-regs (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-regs (*taskid*, *os*))

EVENT: Disable mapup-regs-os-running-output-handler-path2.

THEOREM: mapup-cc-os-running-output-handler-path2
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-cc (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-cc (*taskid*, *os*))

EVENT: Disable mapup-cc-os-running-output-handler-path2.

THEOREM: mapup-error-os-running-output-handler-path2
(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)
∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-error (*taskid*, os-running-output-handler-path2 (*os*))
= mapup-error (*taskid*, *os*))

EVENT: Disable mapup-error-os-running-output-handler-path2.

THEOREM: mapup-svcflag-os-running-output-handler-path2

(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)
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-svcflag (*taskid*, os-running-output-handler-path2 (*os*))
  = mapup-svcflag (*taskid*, *os*))

Event: Disable mapup-svcflag-os-running-output-handler-path2.

Theorem: mapup-svcid-os-running-output-handler-path2
(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)
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-svcid (*taskid*, os-running-output-handler-path2 (*os*))
  = mapup-svcid (*taskid*, *os*))

Event: Disable mapup-svcid-os-running-output-handler-path2.

Theorem: mapup-task-os-running-output-handler-path2
(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)
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16))))))
→ (mapup-task (*taskid*, os-running-output-handler-path2 (*os*))
  = mapup-task (*taskid*, *os*))

Event: Disable mapup-task-os-running-output-handler-path2.

Theorem: mapup-tasks-os-running-output-handler-path2
(good-os (*os*)

$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ (tm-some-output-interruptp (tm-oports (*os*))
$\wedge$ ((getnth ('2
$*$ tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
$=$ '4)
$\wedge$ (*taskid* $\in$ **N**)))))
$\rightarrow$ (mapup-tasks (*taskid*, os-running-output-handler-path2 (*os*))
$=$ mapup-tasks (*taskid*, *os*))

EVENT: Disable mapup-tasks-os-running-output-handler-path2.

THEOREM: mapup-os-tasks-os-running-output-handler-path2
(good-os (*os*)
$\wedge$ ((¬ tm-waiting (*os*))
$\wedge$ (tm-some-output-interruptp (tm-oports (*os*))
$\wedge$ (getnth ('2
$*$ tm-interrupting-output-port (tm-oports (*os*)),
os-status-table (*os*))
$=$ '4))))
$\rightarrow$ (mapup-os-tasks (os-running-output-handler-path2 (*os*))
$=$ mapup-os-tasks (*os*))

EVENT: Disable mapup-os-tasks-os-running-output-handler-path2.

THEOREM: mapup-os-ibuffers-os-running-output-handler-path2
good-os (*os*)
$\rightarrow$ (mapup-os-ibuffers (os-running-output-handler-path2 (*os*))
$=$ mapup-os-ibuffers (*os*))

EVENT: Disable mapup-os-ibuffers-os-running-output-handler-path2.

THEOREM: mapup-os-obuffers-os-running-output-handler-path2
(good-os (*os*)
$\wedge$ (tm-some-output-interruptp (tm-oports (*os*))
$\wedge$ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))
$\rightarrow$ (mapup-os-obuffers (os-running-output-handler-path2 (*os*))
$=$ deq-ith-buffer (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os-obuffers (*os*)))

EVENT: Disable mapup-os-obuffers-os-running-output-handler-path2.

THEOREM: mapup-os-mbuffers-os-running-output-handler-path2
good-os (*os*)
→ (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*)
∧ ((¬ 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-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*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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 tm-iports-os-running-output-handler-path2.

THEOREM: tm-oports-os-running-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)))))
→ (tm-oports (os-running-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-running-output-handler-path2.

THEOREM: correctness-of-os-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)))))))
→ (mapup-os (os-running-output-handler-path2 (*os*))
  = ak-running-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
  mapup-os (*os*)))

EVENT: Disable correctness-of-os-running-output-handler-path2.

THEOREM: mapup-base-os-running-output-handler-path3
good-os (*os*)
→ (mapup-base (*taskid*, os-running-output-handler-path3 (*os*))
  = mapup-base (*taskid*, *os*))

EVENT: Disable mapup-base-os-running-output-handler-path3.

THEOREM: mapup-limit-os-running-output-handler-path3
good-os (*os*)
→ (mapup-limit (*taskid*, os-running-output-handler-path3 (*os*))
  = mapup-limit (*taskid*, *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*))

$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad (\text{mapup-regs}\,(taskid,\ \text{os-running-output-handler-path3}\,(os))$
$= \quad \text{mapup-regs}\,(taskid,\ os))$

EVENT: Disable mapup-regs-os-running-output-handler-path3.

THEOREM: mapup-cc-os-running-output-handler-path3
$(\text{good-os}\,(os)$
$\wedge \quad ((\neg\ \text{tm-waiting}\,(os))$
$\wedge \quad (\text{tm-some-output-interruptp}\,(\text{tm-oports}\,(os))$
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad (\text{mapup-cc}\,(taskid,\ \text{os-running-output-handler-path3}\,(os))$
$= \quad \text{mapup-cc}\,(taskid,\ os))$

EVENT: Disable mapup-cc-os-running-output-handler-path3.

THEOREM: mapup-error-os-running-output-handler-path3
$(\text{good-os}\,(os)$
$\wedge \quad ((\neg\ \text{tm-waiting}\,(os))$
$\wedge \quad (\text{tm-some-output-interruptp}\,(\text{tm-oports}\,(os))$
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad (\text{mapup-error}\,(taskid,\ \text{os-running-output-handler-path3}\,(os))$
$= \quad \text{mapup-error}\,(taskid,\ os))$

EVENT: Disable mapup-error-os-running-output-handler-path3.

THEOREM: mapup-svcflag-os-running-output-handler-path3
$(\text{good-os}\,(os)$
$\wedge \quad ((\neg\ \text{tm-waiting}\,(os))$
$\wedge \quad (\text{tm-some-output-interruptp}\,(\text{tm-oports}\,(os))$
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad (\text{mapup-svcflag}\,(taskid,\ \text{os-running-output-handler-path3}\,(os))$
$= \quad \text{mapup-svcflag}\,(taskid,\ os))$

EVENT: Disable mapup-svcflag-os-running-output-handler-path3.

THEOREM: mapup-svcid-os-running-output-handler-path3
$(\text{good-os}\,(os)$
$\wedge \quad ((\neg\ \text{tm-waiting}\,(os))$
$\wedge \quad (\text{tm-some-output-interruptp}\,(\text{tm-oports}\,(os))$
$\wedge \quad ((taskid \in \mathbf{N}) \wedge (taskid < \texttt{'16}))))$
$\rightarrow \quad (\text{mapup-svcid}\,(taskid,\ \text{os-running-output-handler-path3}\,(os))$
$= \quad \text{mapup-svcid}\,(taskid,\ os))$

EVENT: Disable mapup-svcid-os-running-output-handler-path3.

THEOREM: mapup-task-os-running-output-handler-path3
(good-os (*os*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ (*taskid* ∈ **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*)
 ∧ ((¬ tm-waiting (*os*)) ∧ 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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))
  ∧ (tm-some-output-interruptp (tm-oports (*os*))
   ∧ ((*taskid* ∈ **N**) ∧ (*taskid* < '16)))))
→ (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*)
 ∧ ((¬ tm-waiting (*os*))

$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-svcflag ($taskid$, os-running-output-handler-path4 (os))
= mapup-svcflag ($taskid$, os))

Event: Disable mapup-svcflag-os-running-output-handler-path4.

Theorem: mapup-svcid-os-running-output-handler-path4
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-svcid ($taskid$, os-running-output-handler-path4 (os))
= mapup-svcid ($taskid$, os))

Event: Disable mapup-svcid-os-running-output-handler-path4.

Theorem: mapup-task-os-running-output-handler-path4
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-output-interruptp (tm-oports (os))
$\wedge$ (($taskid \in \mathbf{N}$) $\wedge$ ($taskid <$ '16)))))
$\rightarrow$ (mapup-task ($taskid$, os-running-output-handler-path4 (os))
= mapup-task ($taskid$, os))

Event: Disable mapup-task-os-running-output-handler-path4.

Theorem: mapup-tasks-os-running-output-handler-path4
(good-os (os)
$\wedge$ ((¬ tm-waiting (os))
$\wedge$ (tm-some-output-interruptp (tm-oports (os)) $\wedge$ ($taskid \in \mathbf{N}$))))
$\rightarrow$ (mapup-tasks ($taskid$, os-running-output-handler-path4 (os))
= mapup-tasks ($taskid$, os))

Event: Disable mapup-tasks-os-running-output-handler-path4.

Theorem: mapup-os-tasks-os-running-output-handler-path4
(good-os (os)
$\wedge$ ((¬ tm-waiting (os)) $\wedge$ tm-some-output-interruptp (tm-oports (os))))
$\rightarrow$ (mapup-os-tasks (os-running-output-handler-path4 (os))
= mapup-os-tasks (os))

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-qemptyp (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

(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*))
 ∧ (¬ array-qemptyp (os-current-obuffer (tm-interrupting-output-port (tm-oports (*os*)),
os)))))
→ (tm-oports (os-running-output-handler-path4 (*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-running-output-handler-path4.

Theorem: correctness-of-os-running-output-handler-path4
(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-path4 (*os*))
 = ak-running-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

Event: Disable correctness-of-os-running-output-handler-path4.

Theorem: correctness-of-os-running-output-handler
(good-os (*os*)
 ∧ (tm-some-output-interruptp (tm-oports (*os*)) ∧ (¬ tm-waiting (*os*))))
→ (mapup-os (os-running-output-handler (*os*))
 = ak-running-output-interrupt-handler (tm-interrupting-output-port (tm-oports (*os*)),
mapup-os (*os*)))

Event: Disable correctness-of-os-running-output-handler.

Theorem: tm-svcid-mapup-task
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (tm-svcid (mapup-task (qfirst (mapup-queue (os-readyq (*os*))), *os*))
 = tm-svcid (*os*))

Event: Disable tm-svcid-mapup-task.

THEOREM: getnth-mapup-tasks-corollary
$((i < \texttt{'16}) \wedge (i \in \mathbf{N}))$
$\rightarrow \quad (\text{getnth}\,(i,\ \text{mapup-tasks}\,(\texttt{'0},\ os)) = \text{mapup-task}\,(i,\ os))$

EVENT: Disable getnth-mapup-tasks-corollary.

THEOREM: tm-svcid-mapup-os-tasks
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad (\text{tm-svcid}\,(\text{getnth}\,(\text{qfirst}\,(\text{mapup-queue}\,(\text{os-readyq}\,(os))),$
$\text{mapup-os-tasks}\,(os)))$
$= \quad \text{tm-svcid}\,(os))$

EVENT: Disable tm-svcid-mapup-os-tasks.

THEOREM: tm-svcid-getnth-ak-pstates
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad (\text{tm-svcid}\,(\text{getnth}\,(\text{qfirst}\,(\text{ak-readyq}\,(\text{mapup-os}\,(os))),$
$\text{ak-pstates}\,(\text{mapup-os}\,(os))))$
$= \quad \text{tm-svcid}\,(os))$

EVENT: Disable tm-svcid-getnth-ak-pstates.

THEOREM: tm-regs-mapup-task
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad (\text{tm-regs}\,(\text{mapup-task}\,(\text{qfirst}\,(\text{mapup-queue}\,(\text{os-readyq}\,(os))),\ os))$
$= \quad \text{tm-regs}\,(os))$

EVENT: Disable tm-regs-mapup-task.

THEOREM: tm-regs-mapup-os-tasks
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad (\text{tm-regs}\,(\text{getnth}\,(\text{qfirst}\,(\text{mapup-queue}\,(\text{os-readyq}\,(os))),\ \text{mapup-os-tasks}\,(os)))$
$= \quad \text{tm-regs}\,(os))$

EVENT: Disable tm-regs-mapup-os-tasks.

THEOREM: tm-regs-getnth-ak-pstates
$(\text{good-os}\,(os) \wedge (\neg\ \text{tm-waiting}\,(os)))$
$\rightarrow \quad ((\text{tm-r2}\,(\text{getnth}\,(\text{qfirst}\,(\text{ak-readyq}\,(\text{mapup-os}\,(os))),$
$\text{ak-pstates}\,(\text{mapup-os}\,(os))))$
$= \quad \text{tm-r2}\,(os))$
$\wedge \quad (\text{tm-r3}\,(\text{getnth}\,(\text{qfirst}\,(\text{ak-readyq}\,(\text{mapup-os}\,(os))),$
$\text{ak-pstates}\,(\text{mapup-os}\,(os))))$
$= \quad \text{tm-r3}\,(os)))$

EVENT: Disable tm-regs-getnth-ak-pstates.

THEOREM: ak-readyq-mapup-os
good-os (*os*) → (mapup-queue (os-readyq (*os*)) = ak-readyq (mapup-os (*os*)))

EVENT: Disable ak-readyq-mapup-os.

THEOREM: correctness-of-svc-handler
(good-os (*os*) ∧ ((¬ tm-waiting (*os*)) ∧ (¬ tm-errorp (*os*))))
→ (mapup-os (os-svc-handler (*os*)) = ak-svc-handler (mapup-os (*os*)))

EVENT: Disable correctness-of-svc-handler.

THEOREM: ak-waiting-mapup-os
good-os (*os*) → (ak-waiting (mapup-os (*os*)) = tm-waiting (*os*))

EVENT: Disable ak-waiting-mapup-os.

THEOREM: ak-iports-mapup-os
good-os (*os*) → (ak-iports (mapup-os (*os*)) = tm-iports (*os*))

EVENT: Disable ak-iports-mapup-os.

THEOREM: correctness-of-input-interrupt-handler
(good-os (*os*) ∧ tm-some-input-interruptp (tm-iports (*os*)))
→ (mapup-os (os-input-interrupt-handler (*os*))
= ak-input-interrupt-handler (tm-interrupting-input-port (ak-iports (mapup-os (*os*))),
mapup-os (*os*)))

EVENT: Disable correctness-of-input-interrupt-handler.

THEOREM: ak-oports-mapup-os
good-os (*os*) → (ak-oports (mapup-os (*os*)) = tm-oports (*os*))

EVENT: Disable ak-oports-mapup-os.

THEOREM: correctness-of-output-interrupt-handler
(good-os (*os*) ∧ tm-some-output-interruptp (tm-oports (*os*)))
→ (mapup-os (os-output-interrupt-handler (*os*))
= ak-output-interrupt-handler (tm-interrupting-output-port (ak-oports (mapup-os (*os*))),
mapup-os (*os*)))

EVENT: Disable correctness-of-output-interrupt-handler.

THEOREM: tm-error-mapup-task
(good-os (*os*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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*) ∧ (¬ tm-waiting (*os*)))
→ (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
¬ 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*) ∧ (¬ tm-waiting (*os*)))
→ (¬ 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