A Formal Model of Clock Domain Crossing and Automated Verification of Time-Triggered Hardware

Julien Schmaltz*

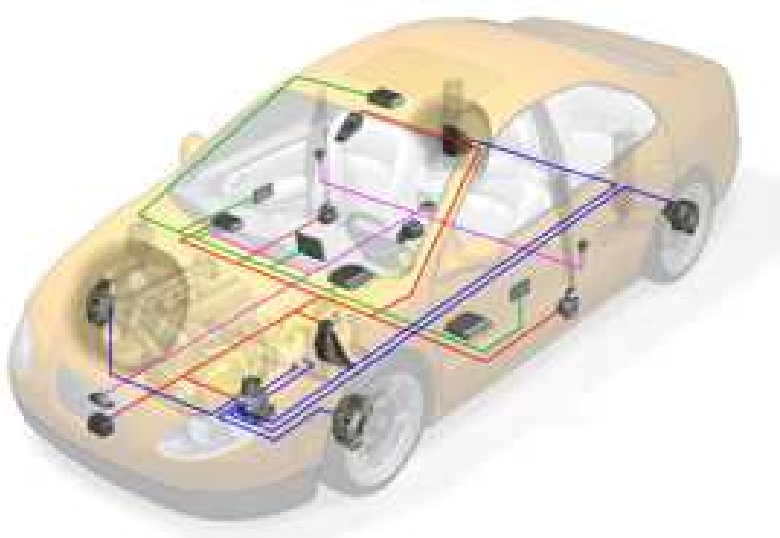
Institute for Computing and Information Sciences
Radboud University Nijmegen
The Netherlands
julien@cs.ru.nl

* Part of this work funded by the Verisoft Project, Uni. Saarbrücken, Germany
and the Marie Curie project TAROT

FMCAD 2007, Nov. 11–14

eCall: Safety-Critical Automotive Application

- Automatic emergency call system
- A phone call is automatically emitted when car sensors detect an accident

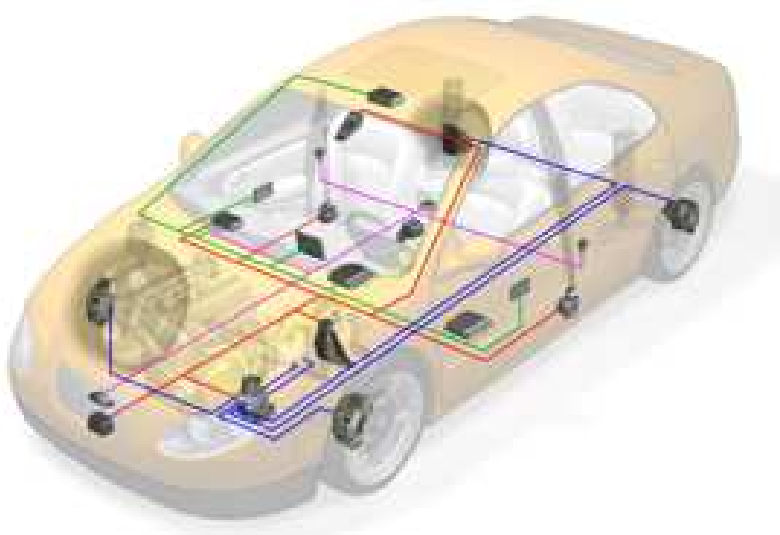


4 distributed components

- Sensors: severity
- Navigation System: position
- Mobile Phone: send information
- eCall: central application

The Verisoft Project

- CLI: original work on stack proof (Moore *et al.*)
- Verisoft: Pervasive verification of distributed systems



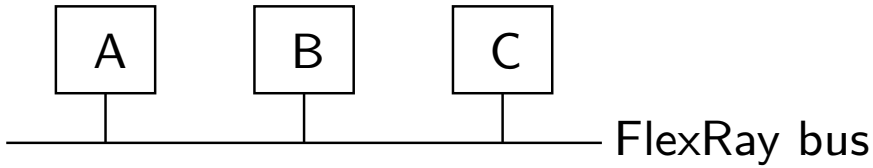
Formal Proofs of

- Applications
- Operating systems
- Compilers
- Processors
- FlexRay bus
  - Asynchronous communications

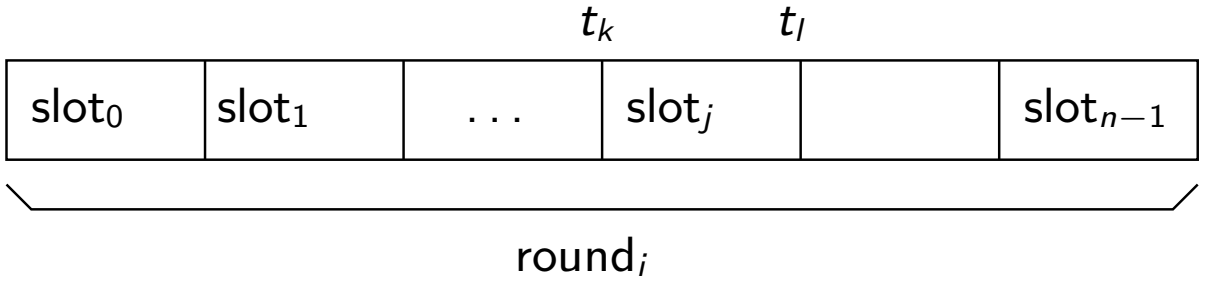
Asynchronous Communications

- Clock imperfections
  - drift: different clocks with different rates
  - jitter: clocks without constant rates
- Metastability
  - Metastable states: register output undefined
  - Resolution: output stabilized non-deterministically to 1 or 0

FlexRay Architecture: Schedule Overview

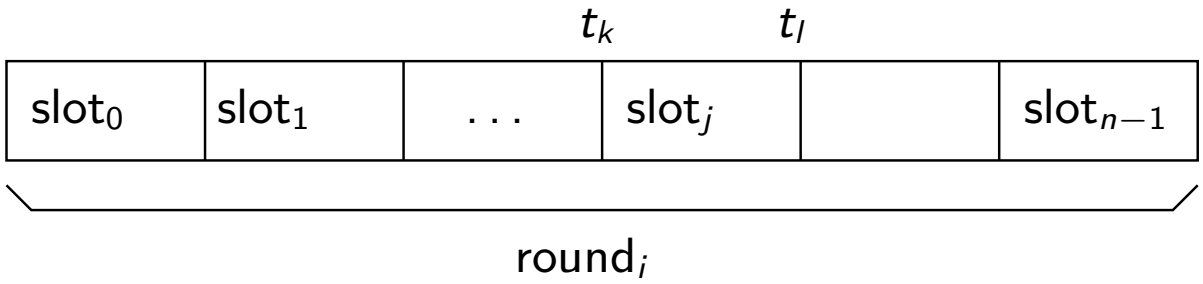


- Time divided into rounds
- Each round divided into slots



- Every unit owns one slot
  - $\text{slot}_0 \longrightarrow A$
  - $\text{slot}_1 \longrightarrow B$
  - $\text{slot}_j \longrightarrow C$
- Clock synchronization algorithm

FlexRay Verification: Overview



- Clock synchronization correctness
  - All units agree on global timing
- Schedule correctness
  - Unit C starts sending m at time t_k at the earliest
  - Unit C stops sending at time t_l at the latest
- Transmission correctness
  - At time t_l, all units have received m
  - Functional correctness + timing analysis

Related Work

Physical layer protocol analysis

- First work by Moore (1993)
  - Biphase mark protocol
  - Theorem proving (Nqthm)
- Contemporary work by Bosscher, Polak and Vaandrager (1994)
  - Philips audio control protocol
- Recent work by Brown and Pike (2006)
  - Biphase mark and 8N1 protocols
  - k-induction (SAL)
- Recent work by Vaandrager and de Groot (2007)
  - Biphase mark protocol
  - Real-time model checking (Uppaal)

All works on abstract models, no real hardware

Contribution

- General formal model of clock domain crossing
  - Metastability
  - Clock drift/jitter
  - Detailed timing parameters
  - Realization in Isabelle/HOL
- Mixed with gate-level hardware designs
  - Combination of theorem proving with automatic tools
- Proof of a FlexRay-like hardware interface
  - Basis theorem for pervasive verification of distributed systems
  - Functional correctness and timing analysis
  - Bounds on crucial parameter of the bit clock synchronization algorithm

Outline

Overall Verification Approach

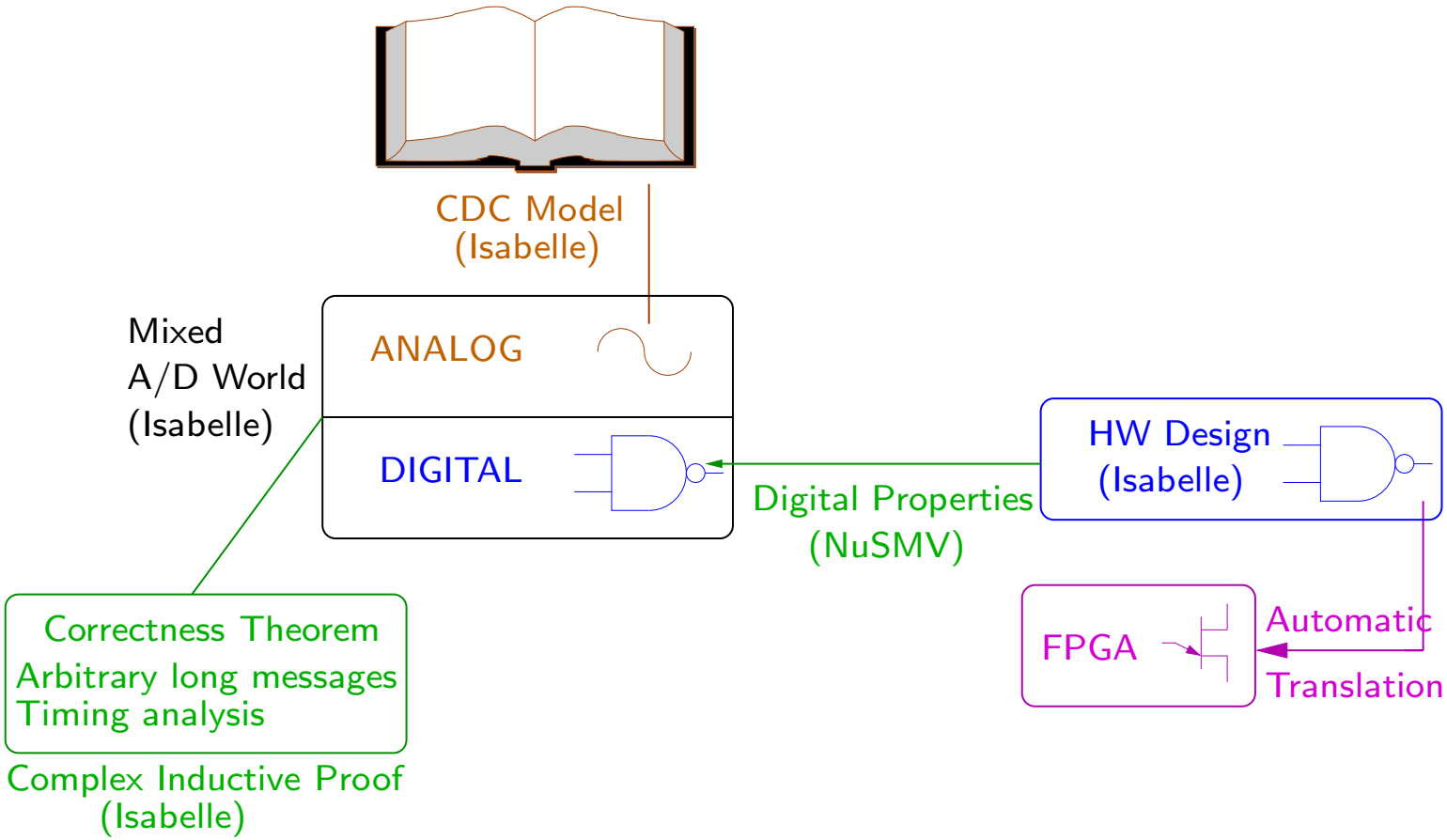
FlexRay Hardware Interface

Clock Domain Crossing Model

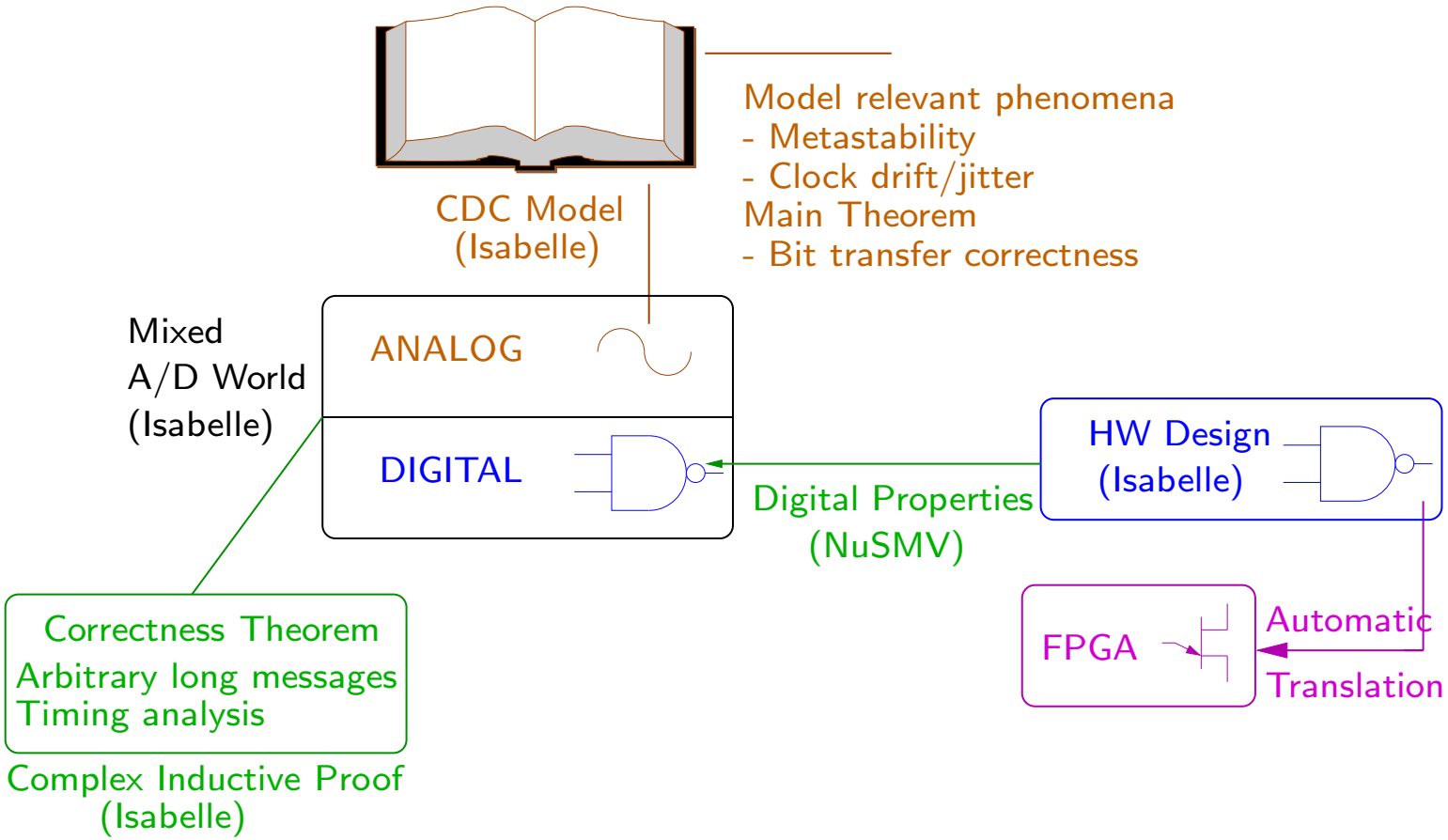
Mixing Digital and Analog

Final Correctness Proof

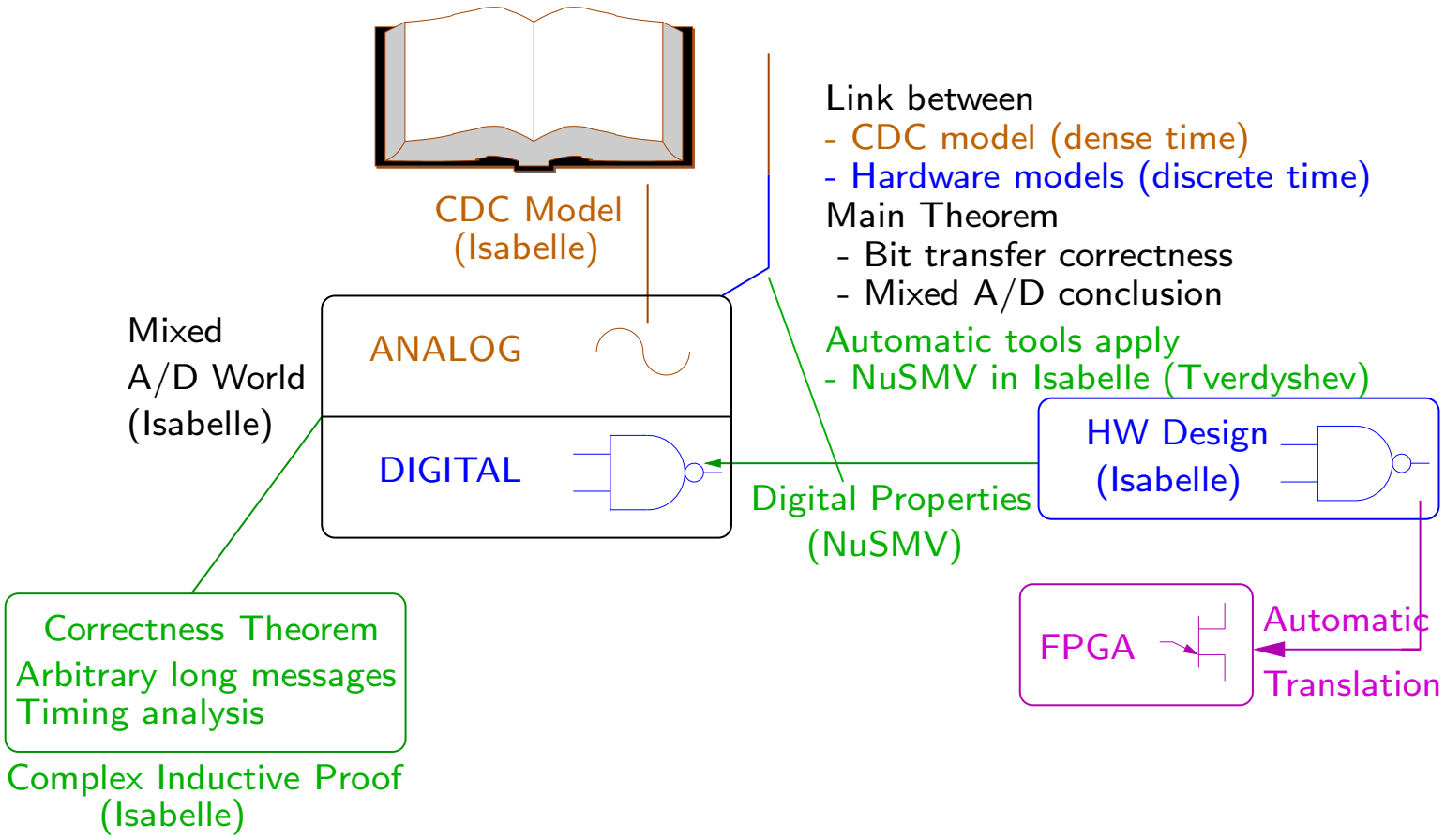
Verification Method



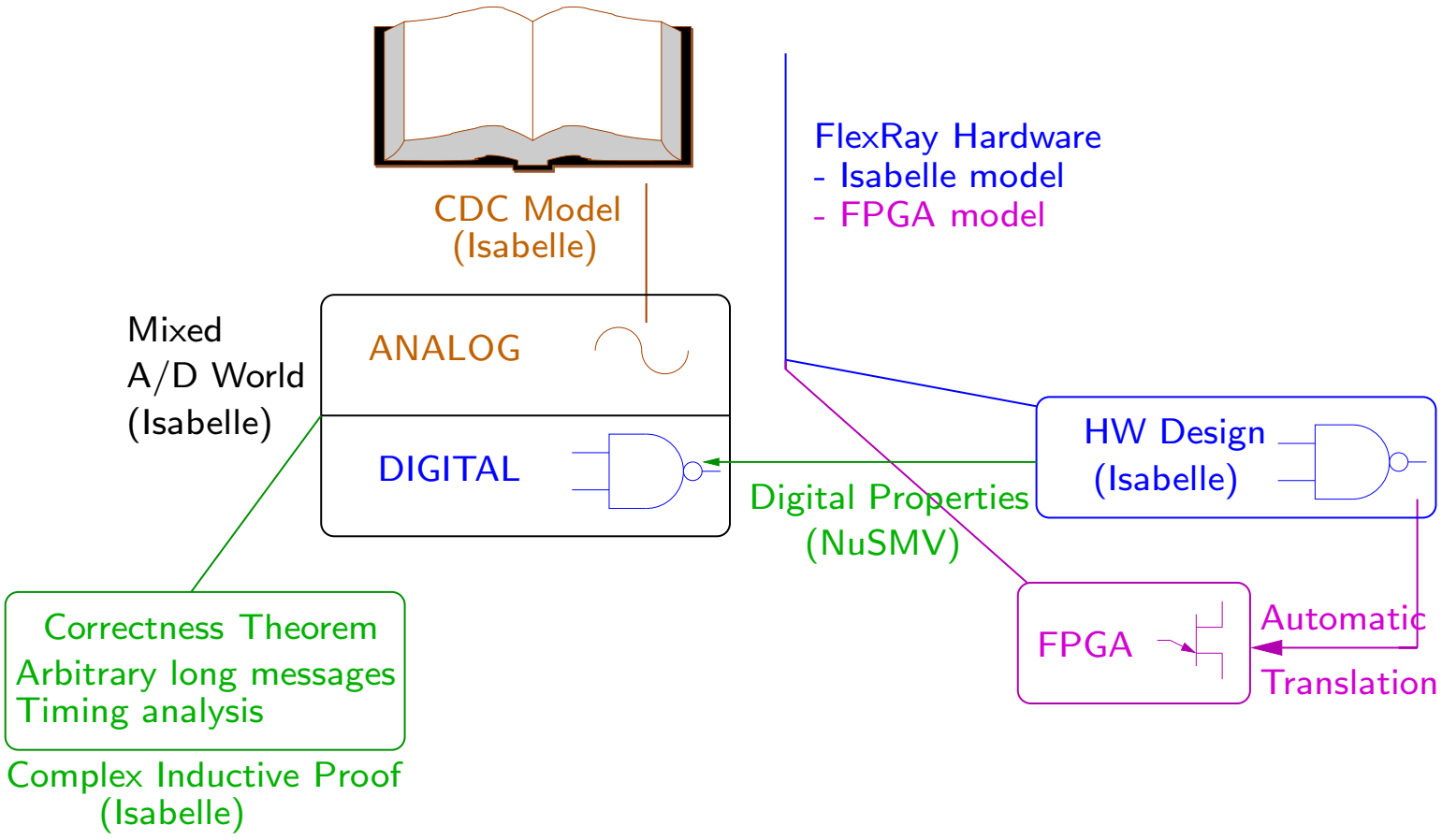
Verification Method: CDC Model



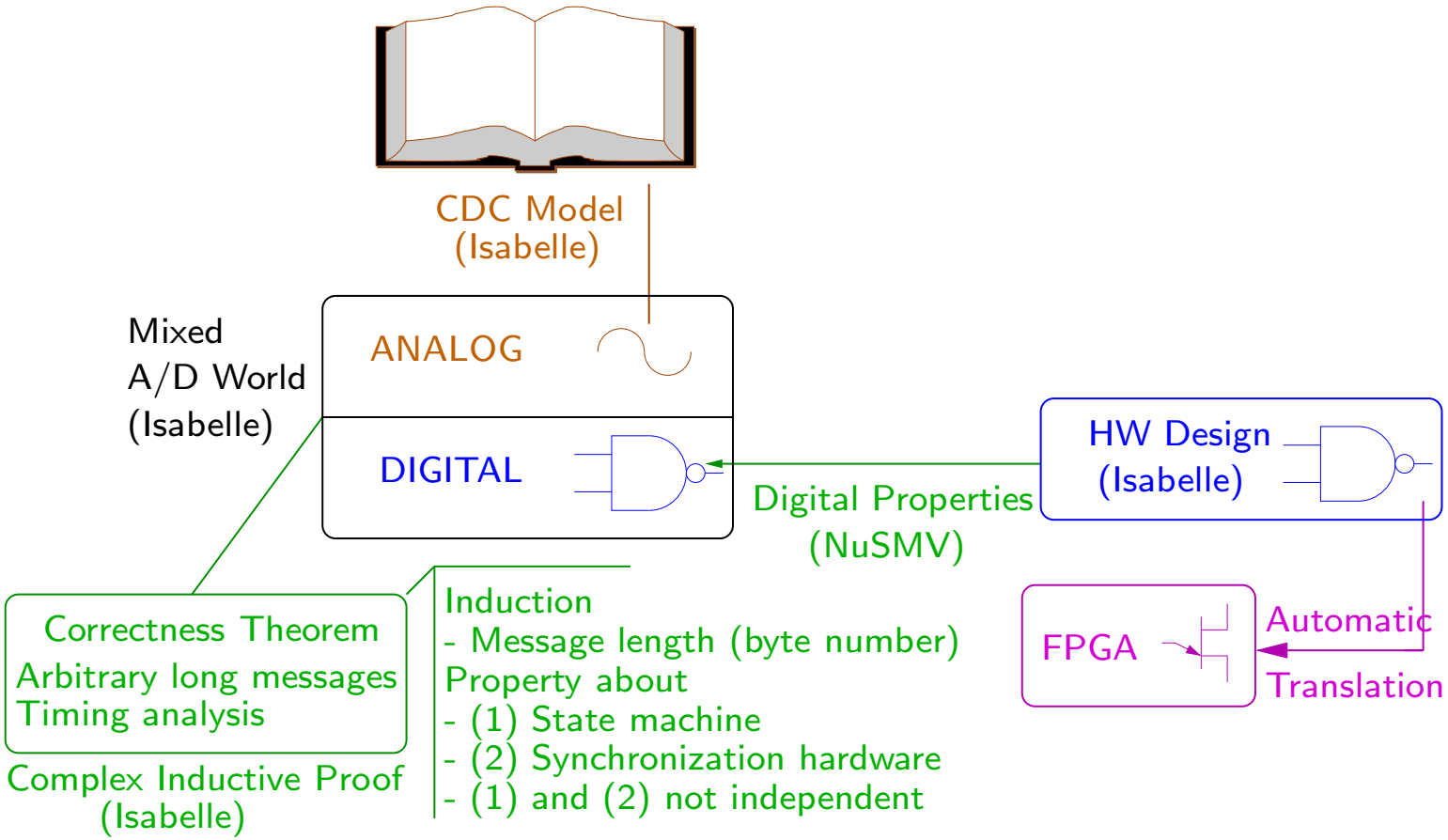
Verification Method: Mixed A/D World



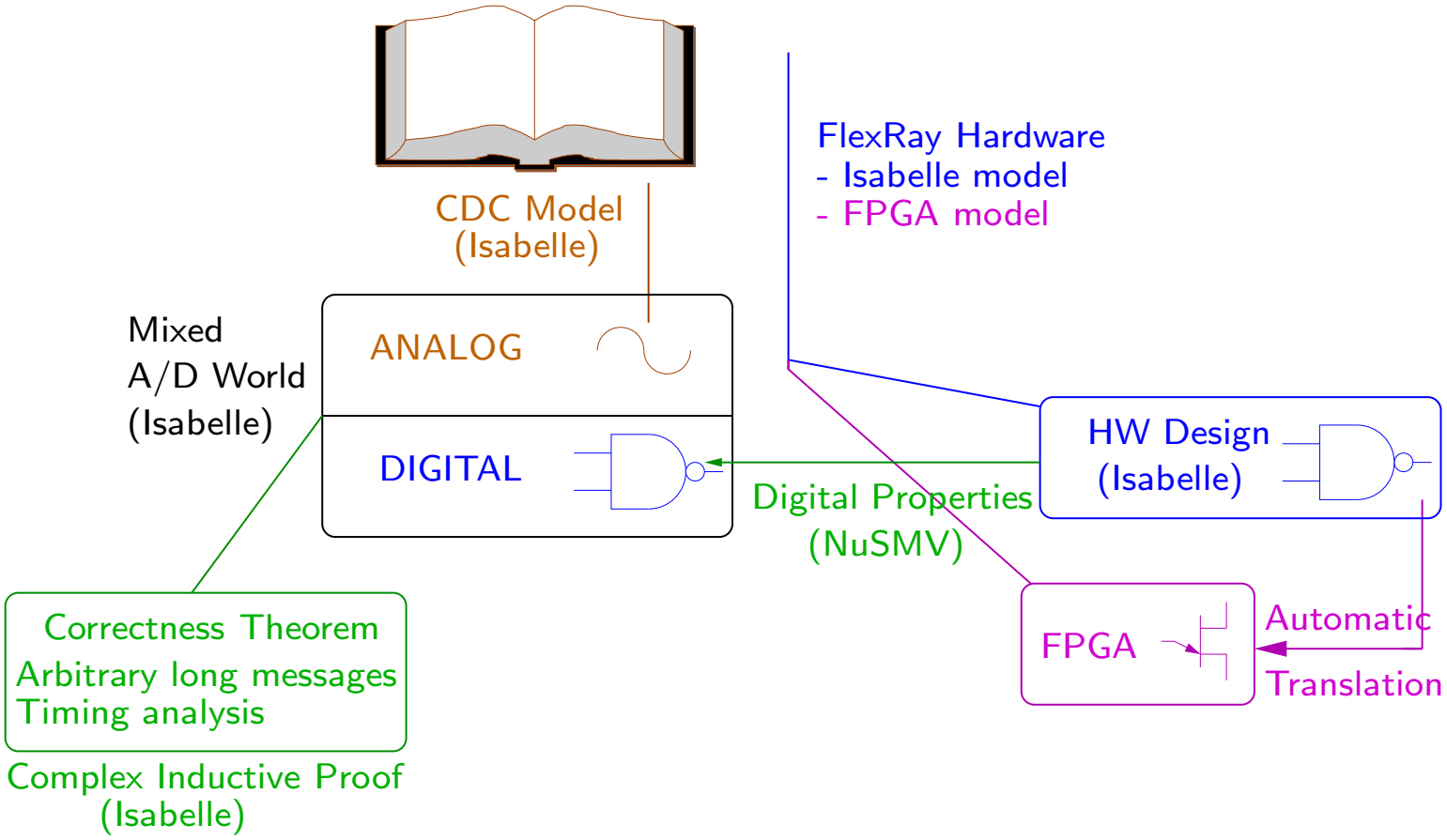
Verification Method: HW Design



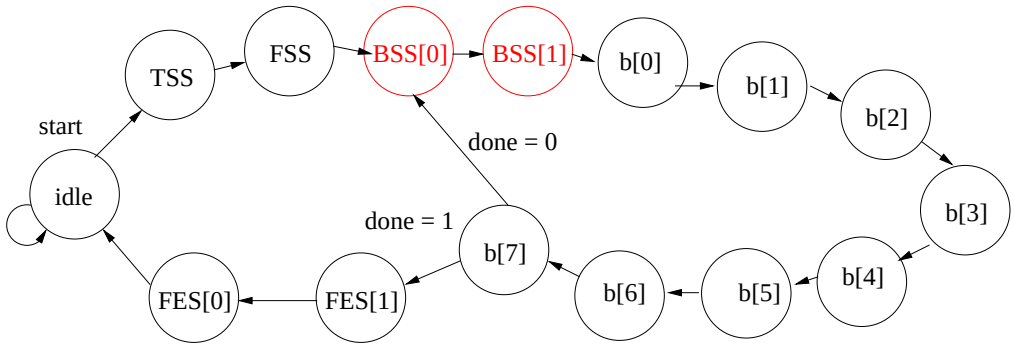
Verification Method: Final Inductive Proof



Outline: HW Design



FlexRay Architecture: Protocol Overview

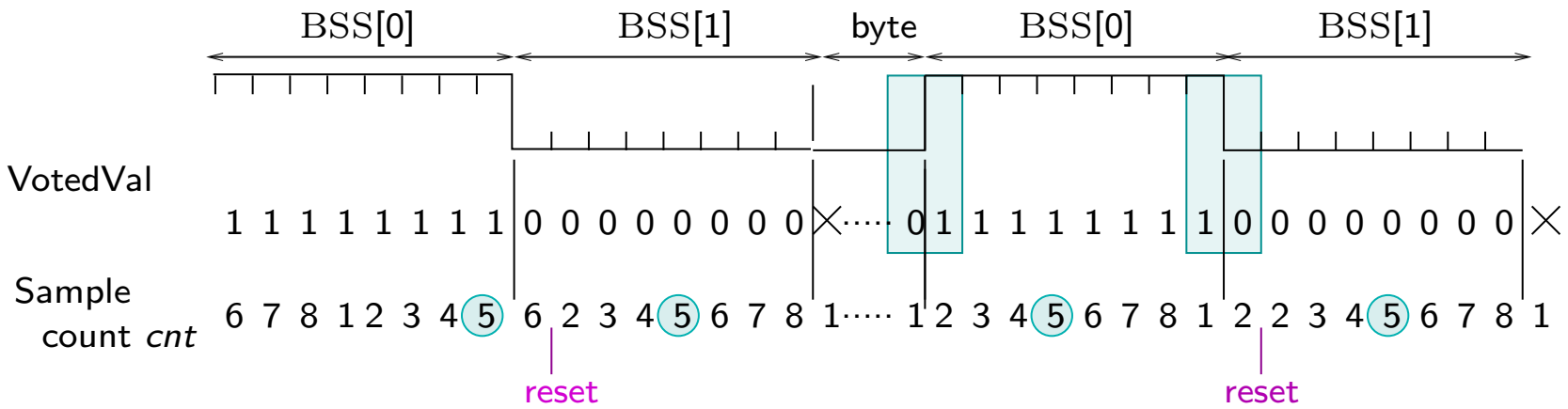


- Receiver and sender implements the same control automaton
- Frames follow the following format

$$f(m) = \langle \mathrm{TSS}, \mathrm{FSS}, \mathrm{BSS}, m[0], \ldots, \mathrm{BSS}, m[l-1], \mathrm{FES} \rangle$$

- Byte synchronization sequence $\mathrm{BSS} = 10$
- Each bit sent 8 times + majority voting

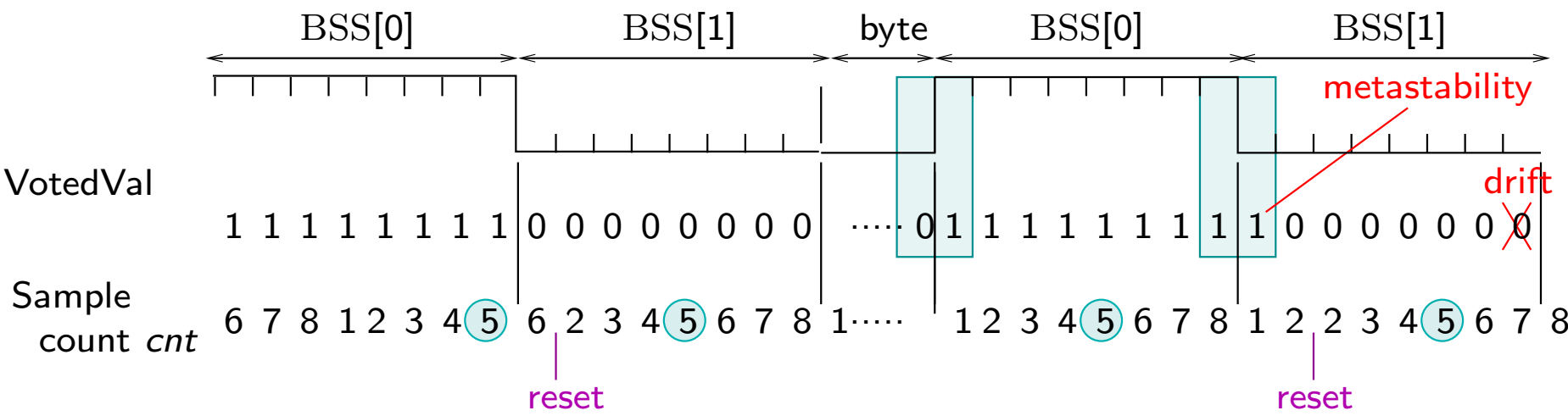
FlexRay Architecture: Bit Clock Synchronization



- Strobe when $cnt = 5$
- cnt reset to 2 at synchronization edges
- Values 5 and 2 fixed by specification document

  (Figure 3-8 page 243 of Protocol Specification v2.1)

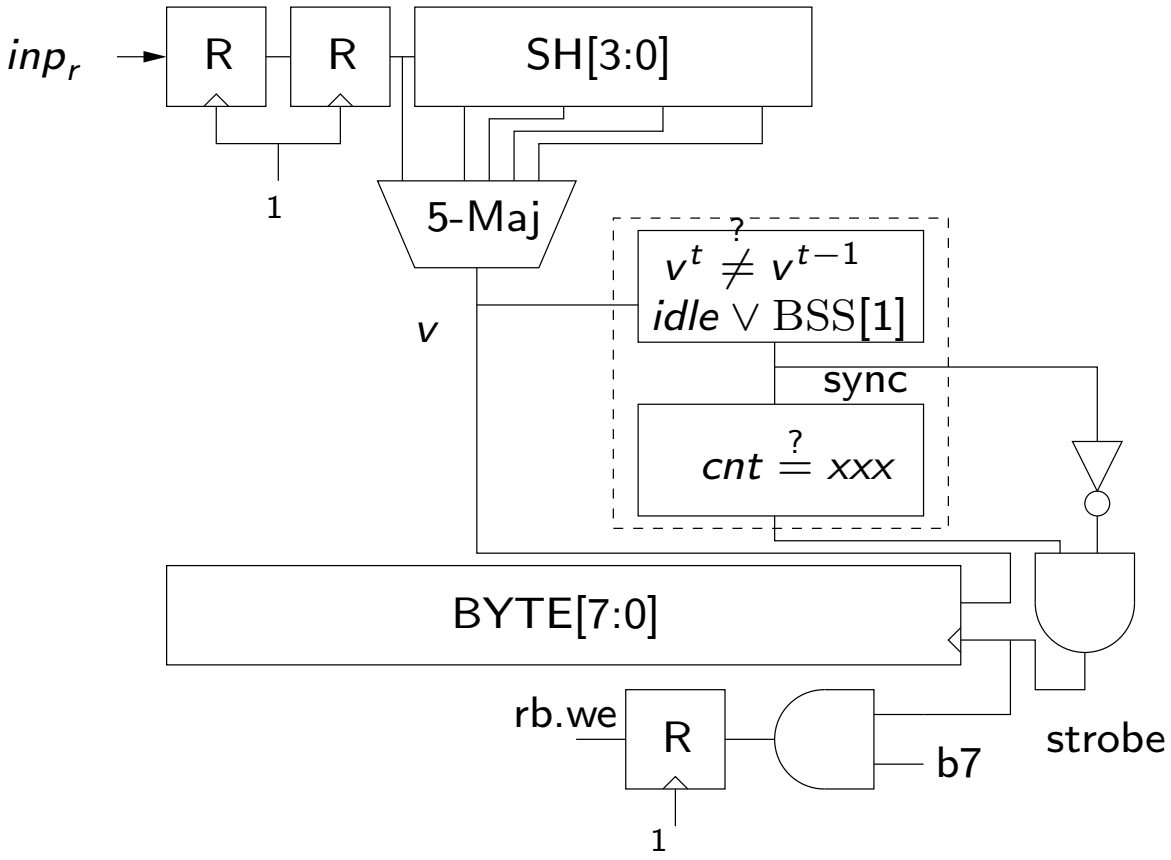
Bit Clock Synchronization and Metastability



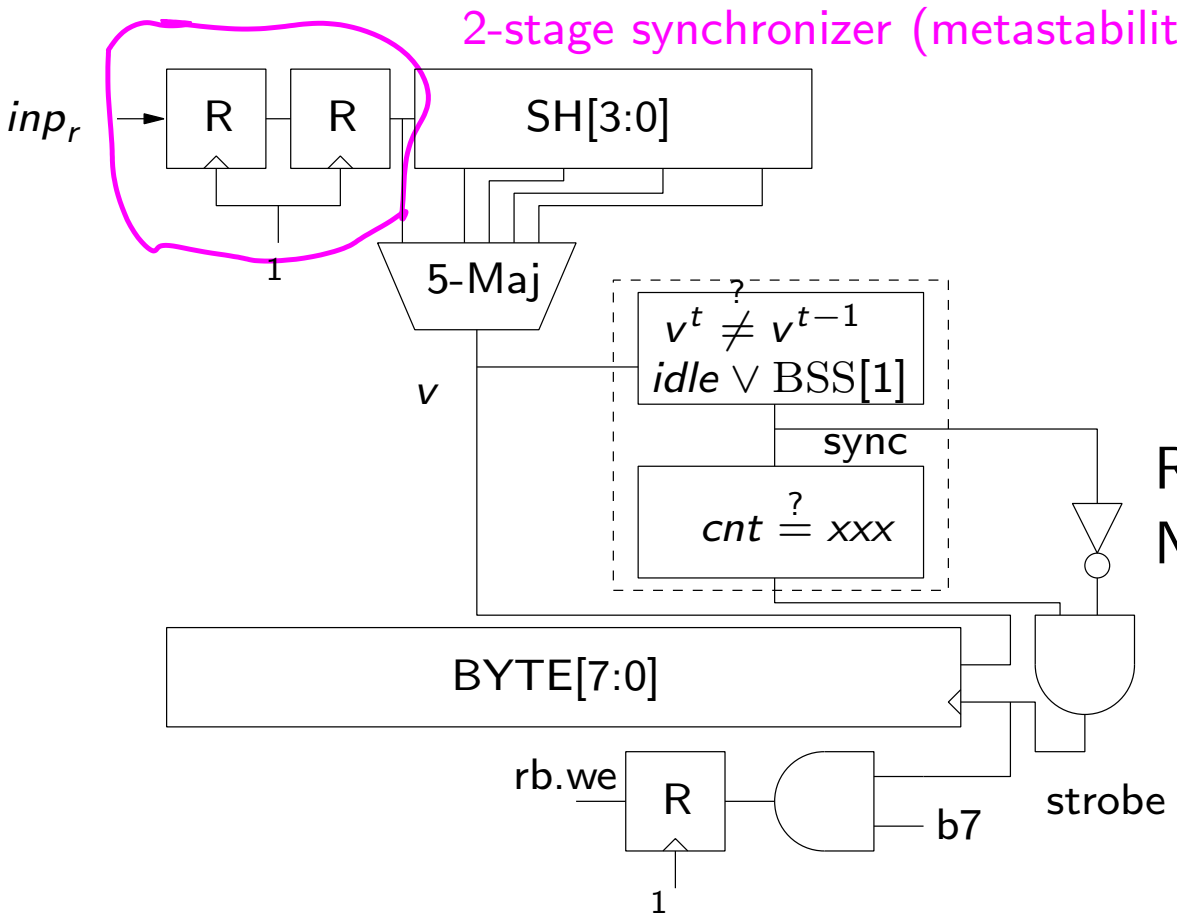
Objective: always sample (roughly) in the middle

- Potential metastability when sampling around falling or rising edges
- Misalignment due to clock drift
- Spikes (ignored)
- Roughly in the middle = 8 bits - first - last = 6 bits

Receiver Input Stage

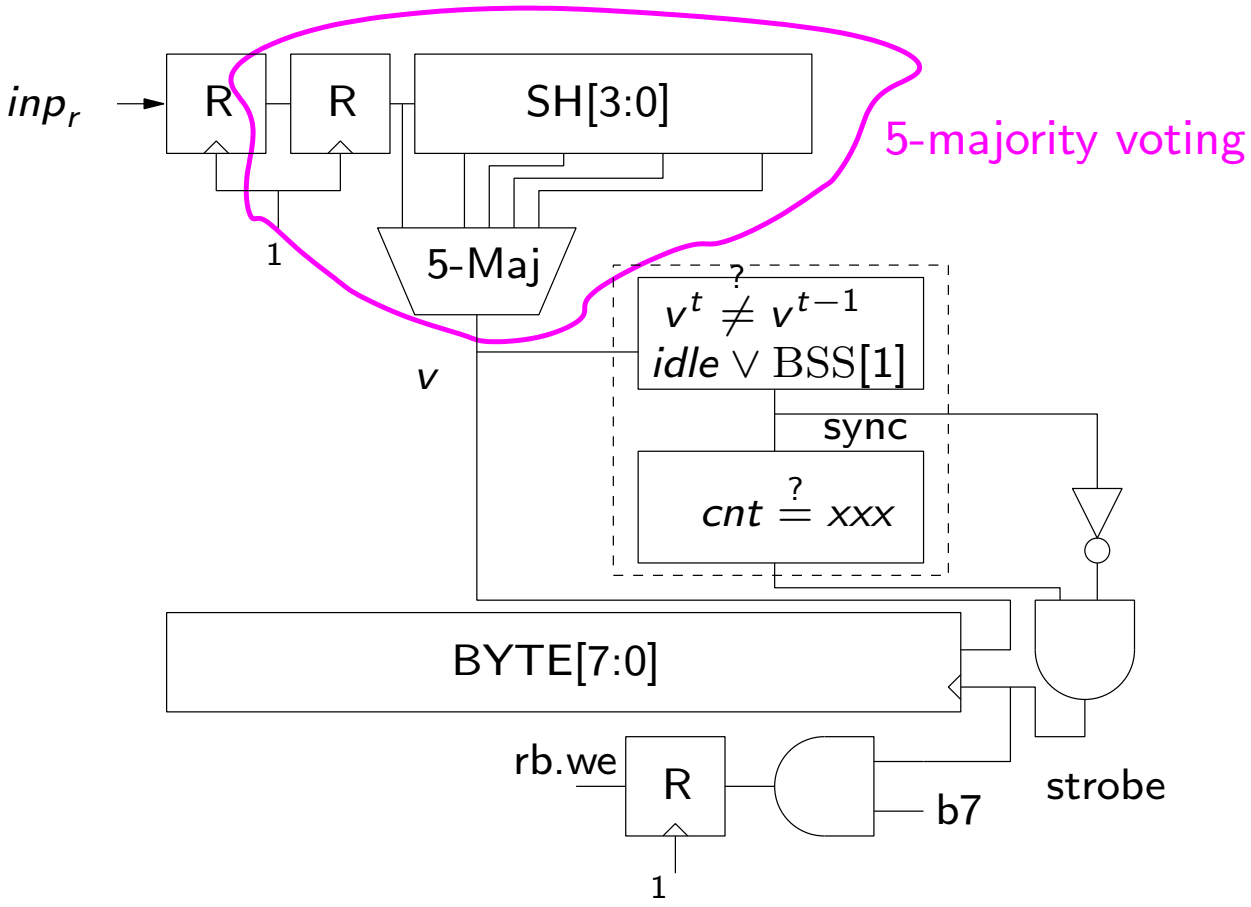


Receiver Input Stage

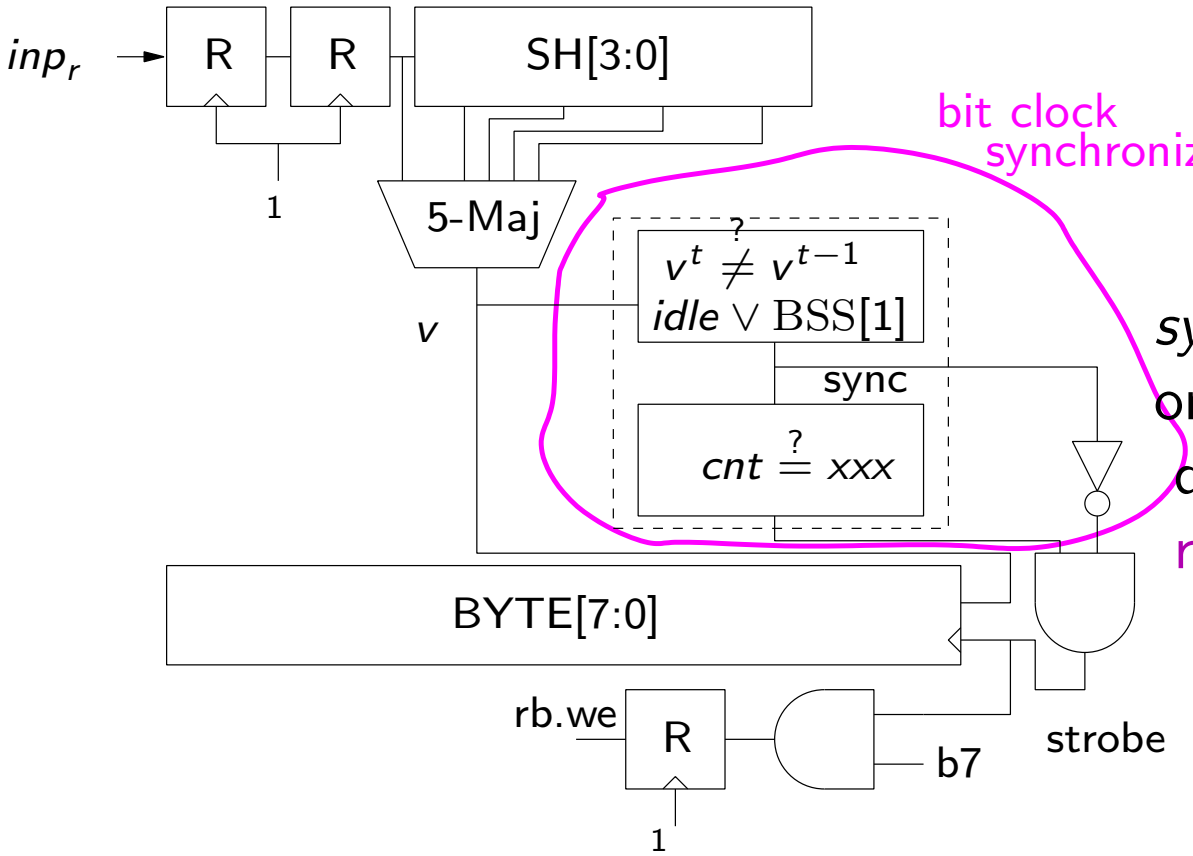


Reg. ♯2 never metastable
Non-det. to 0 or 1

Receiver Input Stage

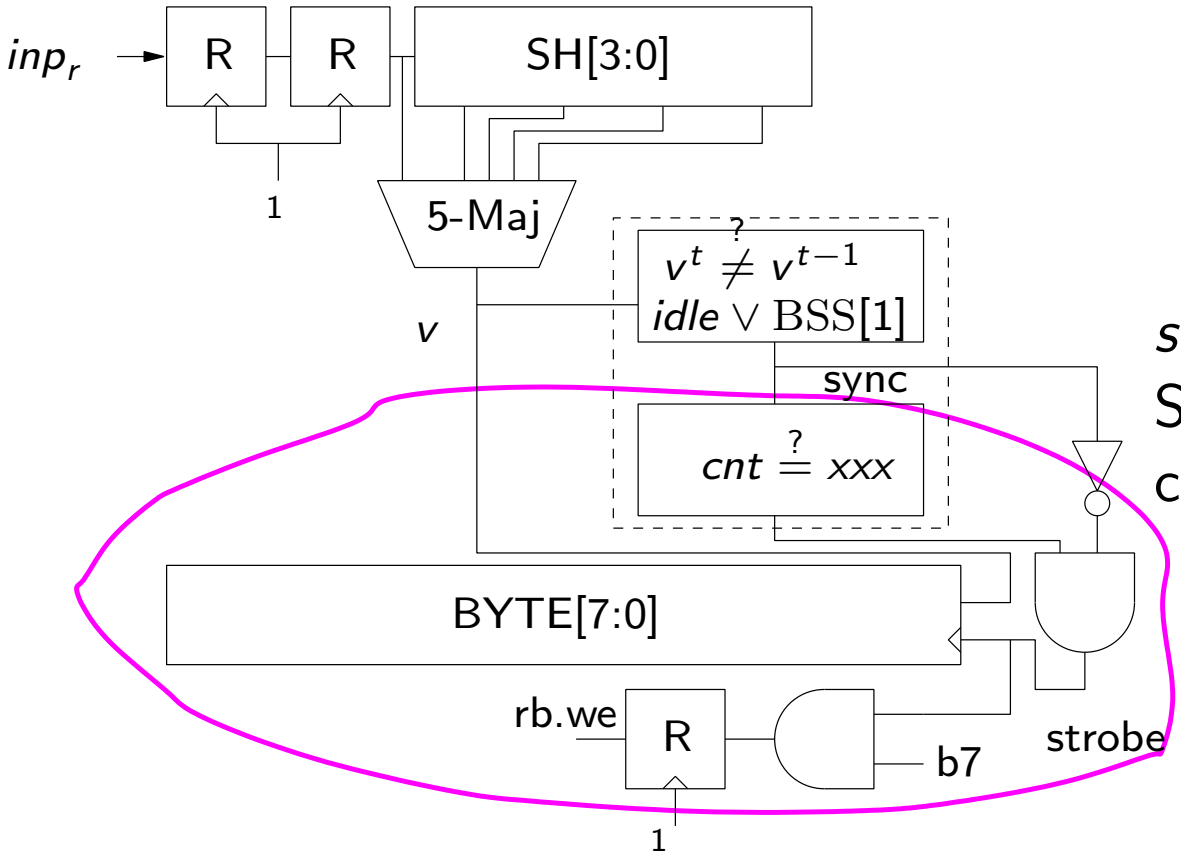


Receiver Input Stage



sync high on falling edges
only if state idle or $\mathrm{BSS}[1]$
disable strobing
reset counter (to yyy)

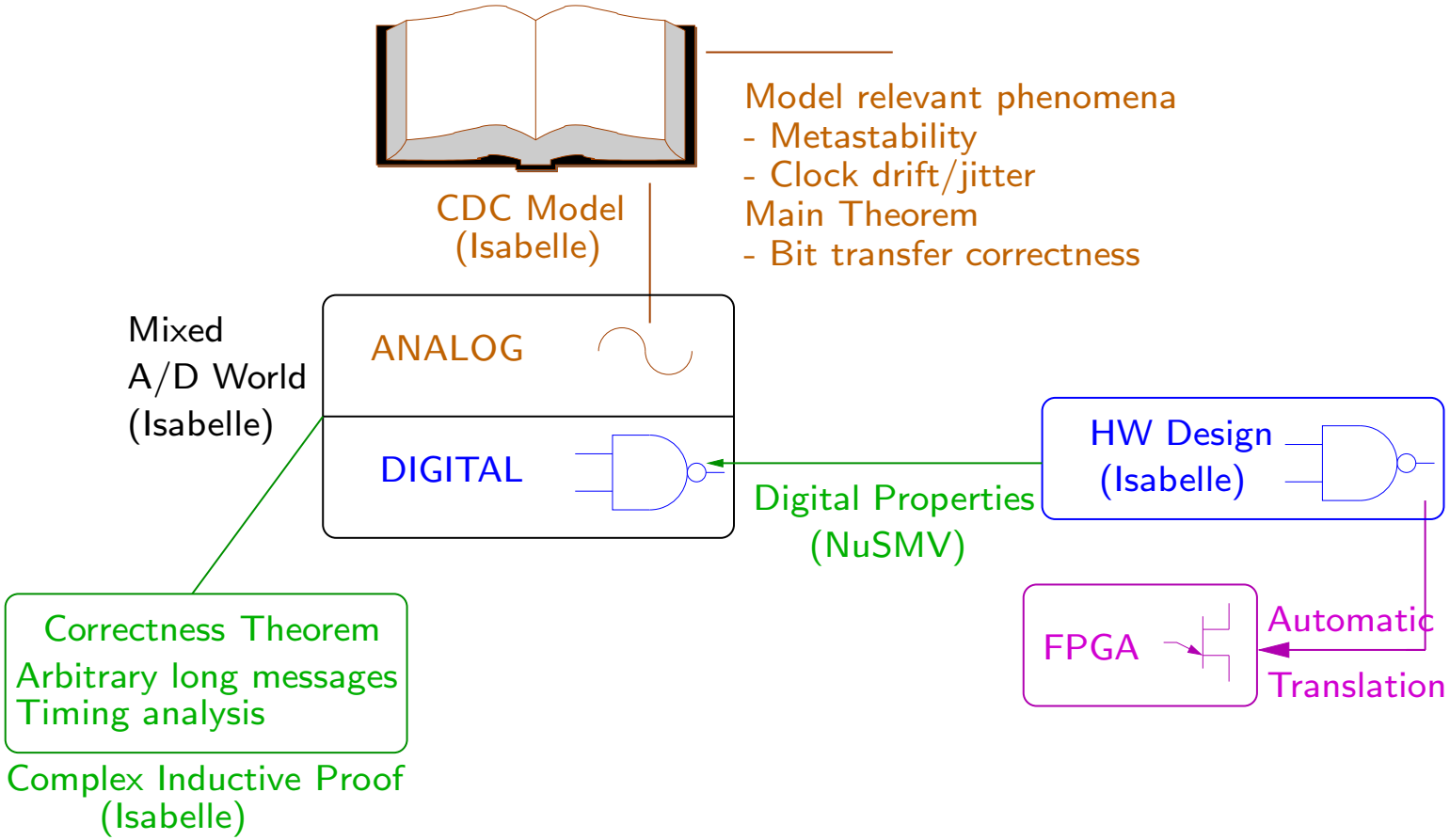
Receiver Input Stage



strobe high when $cnt = xxx$
Store v in BYTE
clock control automaton

strobing mechanism

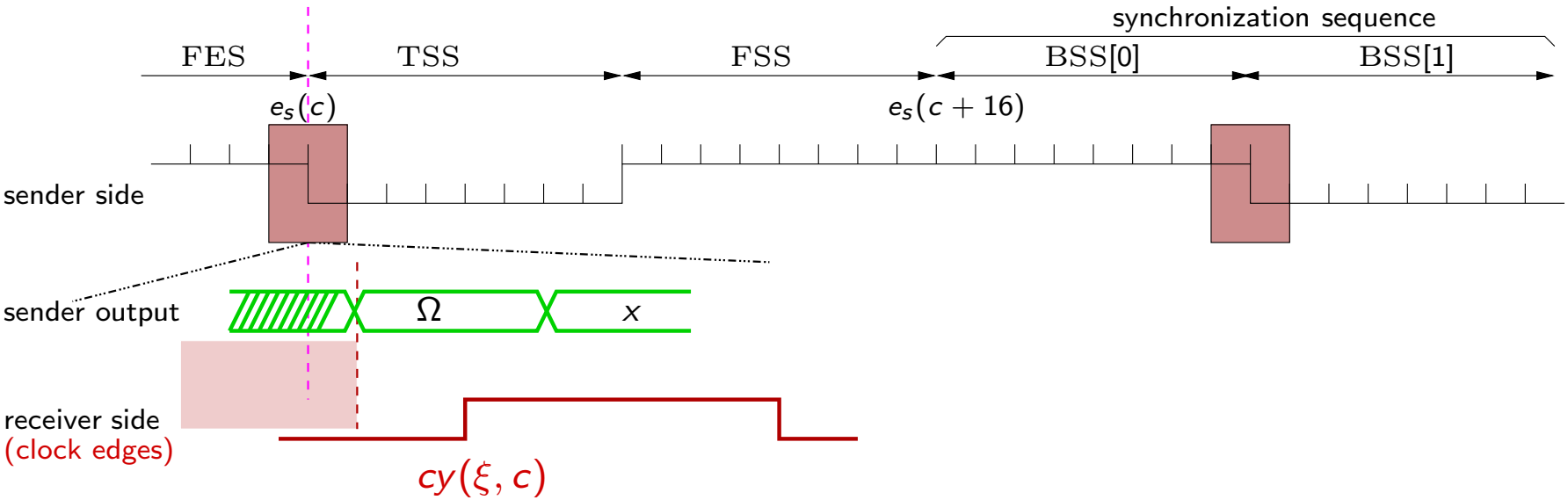
Outline: CDC Model



General Assumptions

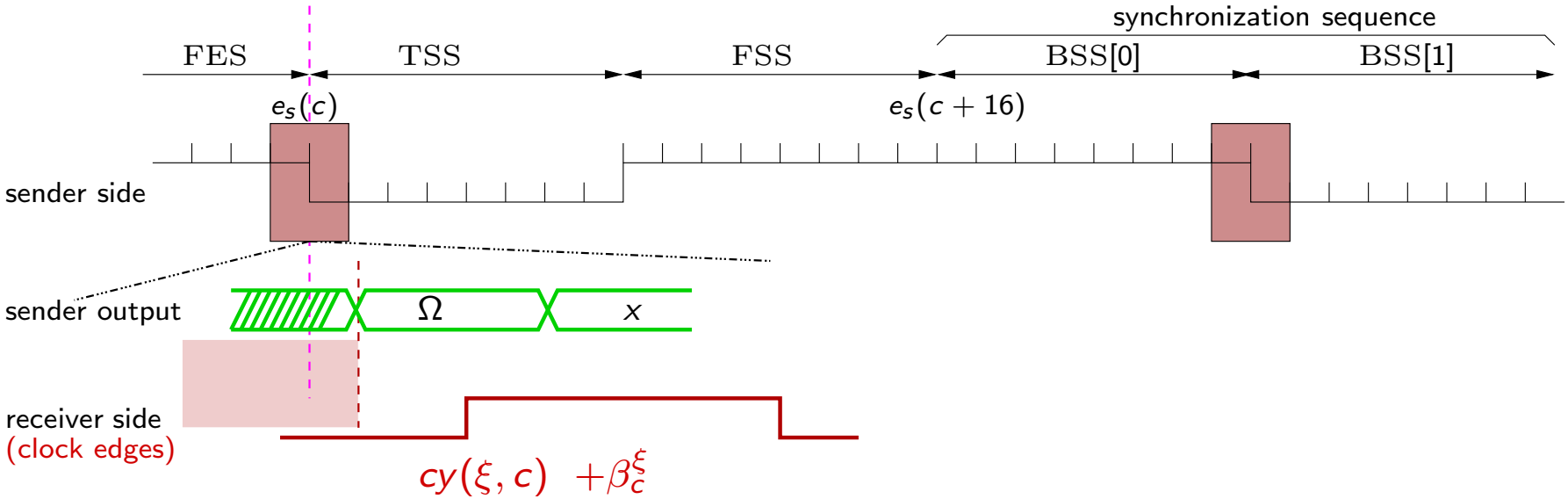
- 3-valued logic:
  - 0, 1 for "low" and "high" voltages
  - Ω for any other voltage
- Time represented by nonnegative reals ($\mathbb{R}_{\geq 0}$)
- Signals are functions from time to $\{0, 1, \Omega\}$
- Transition from low (high) to high (low) via Ω
  - In particular, output signal of registers
  - Consequence: metastable states when sampling Ω
- Clocks represented by their period τ
  - Date of edge $\sharp c$ on unit u noted $e_u(c) = c \cdot \tau_u$
  - Edges have no width

Relating Senders and Receivers



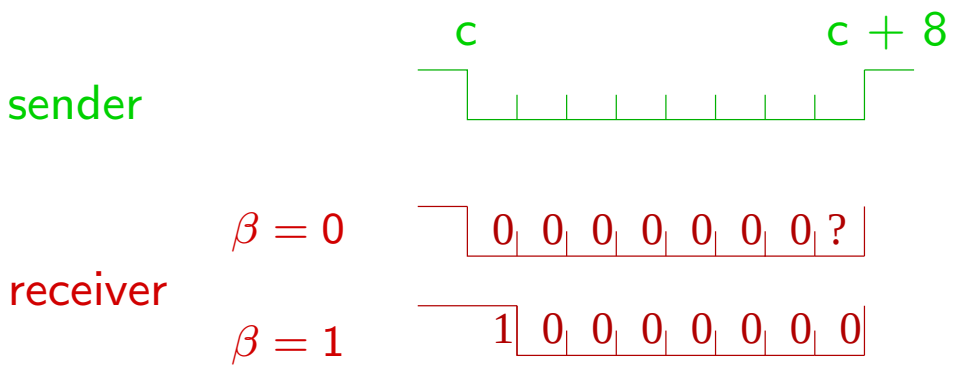
- Sender put x on bus at time $e_s(c)$
- ξ first "affected" (receiver) cycle (to sample x or Ω)

Metastability



- Metastable state when sampling Ω,
- If $cy(\xi, c)$ on Ω, then metastable state
- we may look one cycle later, at $cy(\xi, c) + \beta_c^\xi$:
  - $\beta_c^\xi = 0$ if no metastable state at $cy(\xi, c)$
  - $\beta_c^\xi = 1$ otherwise

Main Analog Theorem: Bit Transfer Correctness



- From sender cycle c
- Bit sent 8 times
- First affected cycle given:
  - $cy(\xi, c)$

Theorem

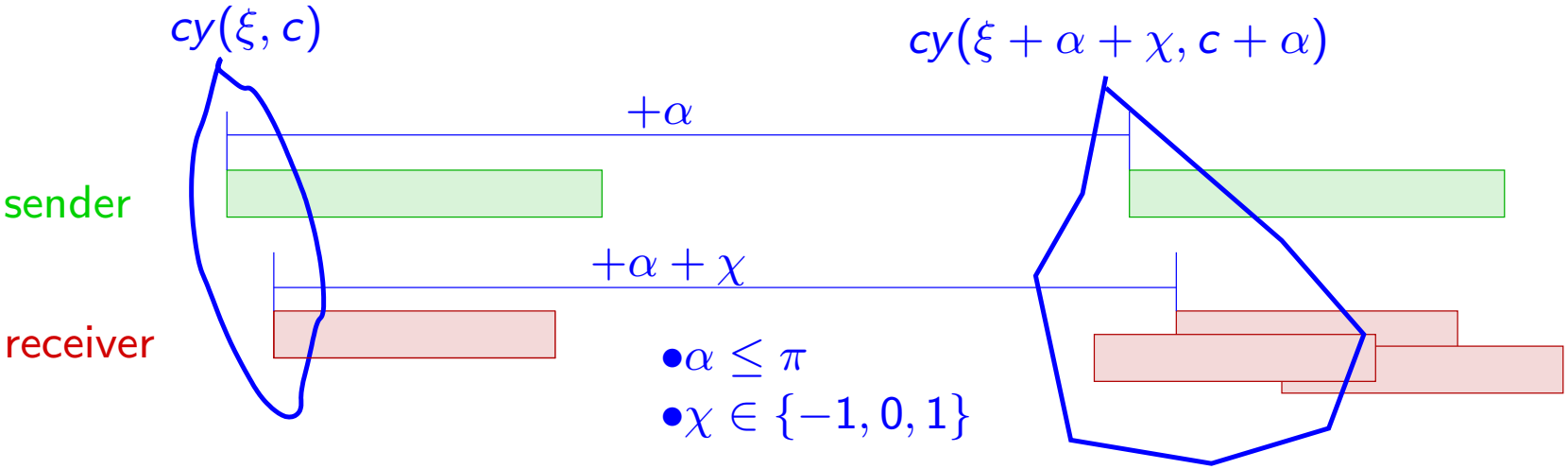
- At least 7 samples on receiver side
- Possible shift of 1 cycle due to metastability

Clock Drift and Jitter

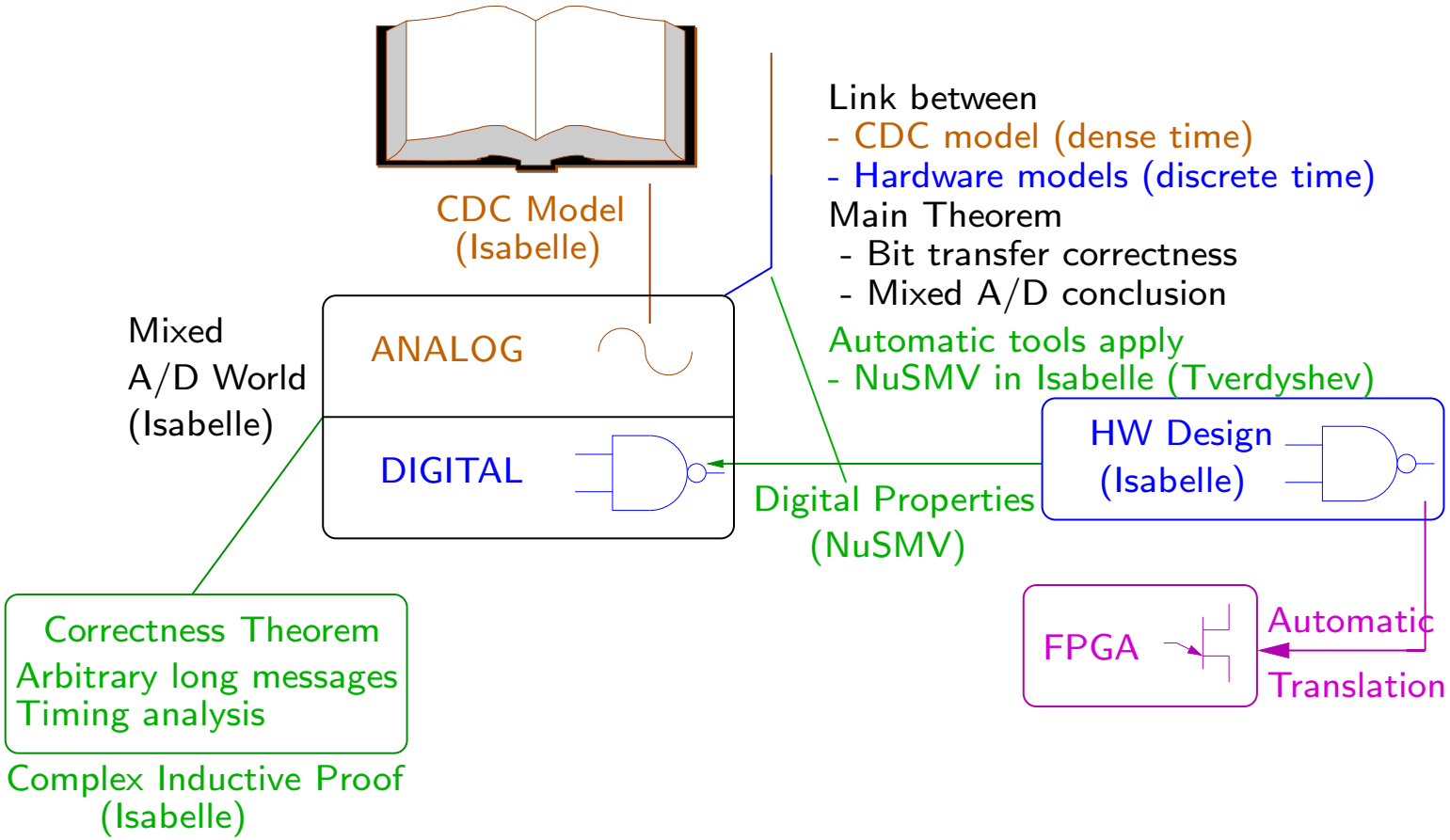
- Clocks not constant over time
  - Drift bounded by percentage δ of reference period

$$1 - \delta \leq \frac{\tau_u}{\tau_{ref}} \leq 1 + \delta$$

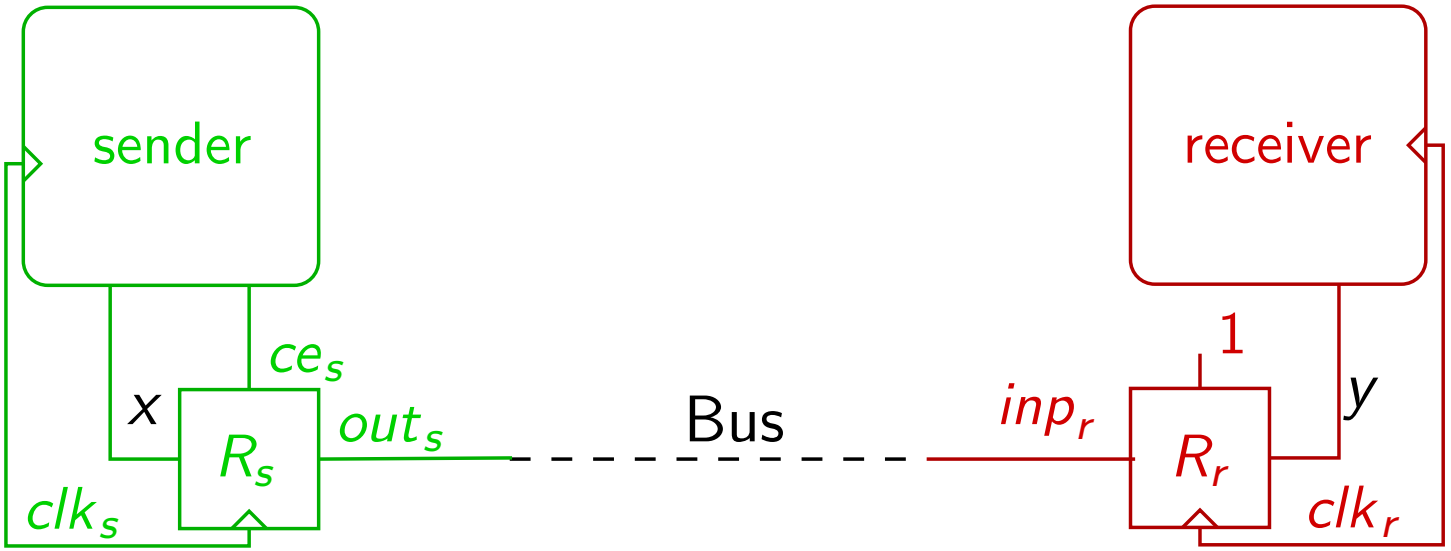
- Lemma
  - Within π cycles, clocks cannot drift by more than 1 cycle
  - From one known mark, next marks have 3 possible positions



Outline: Mixed A/D World

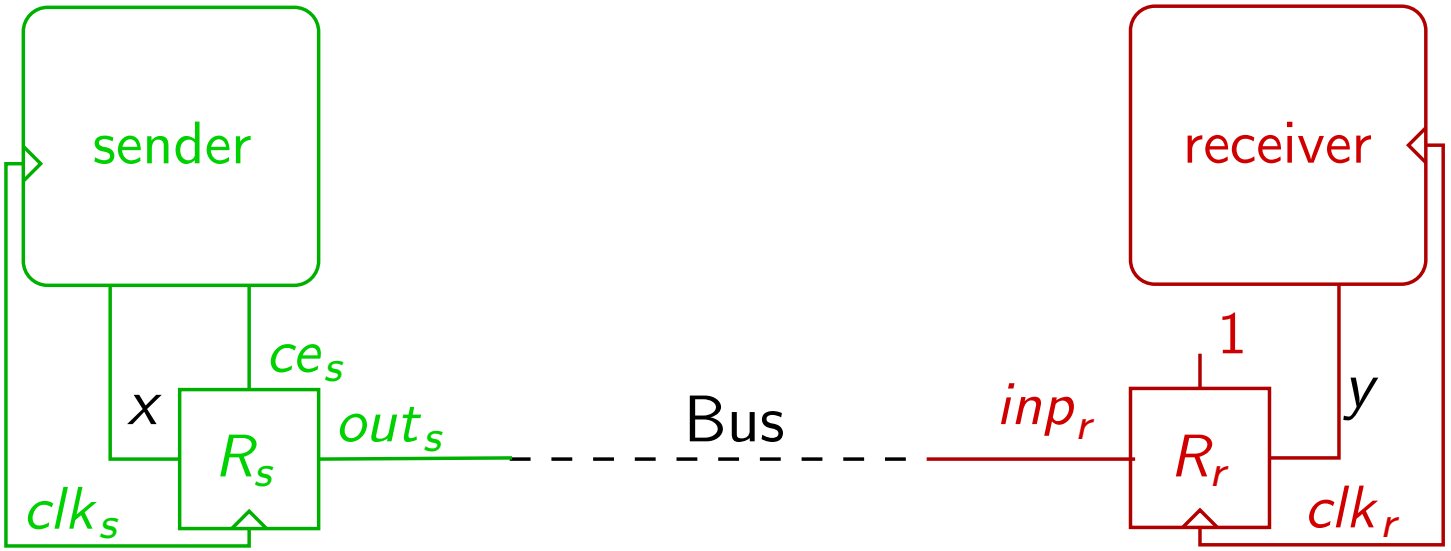


CDC Model and Hardware Designs



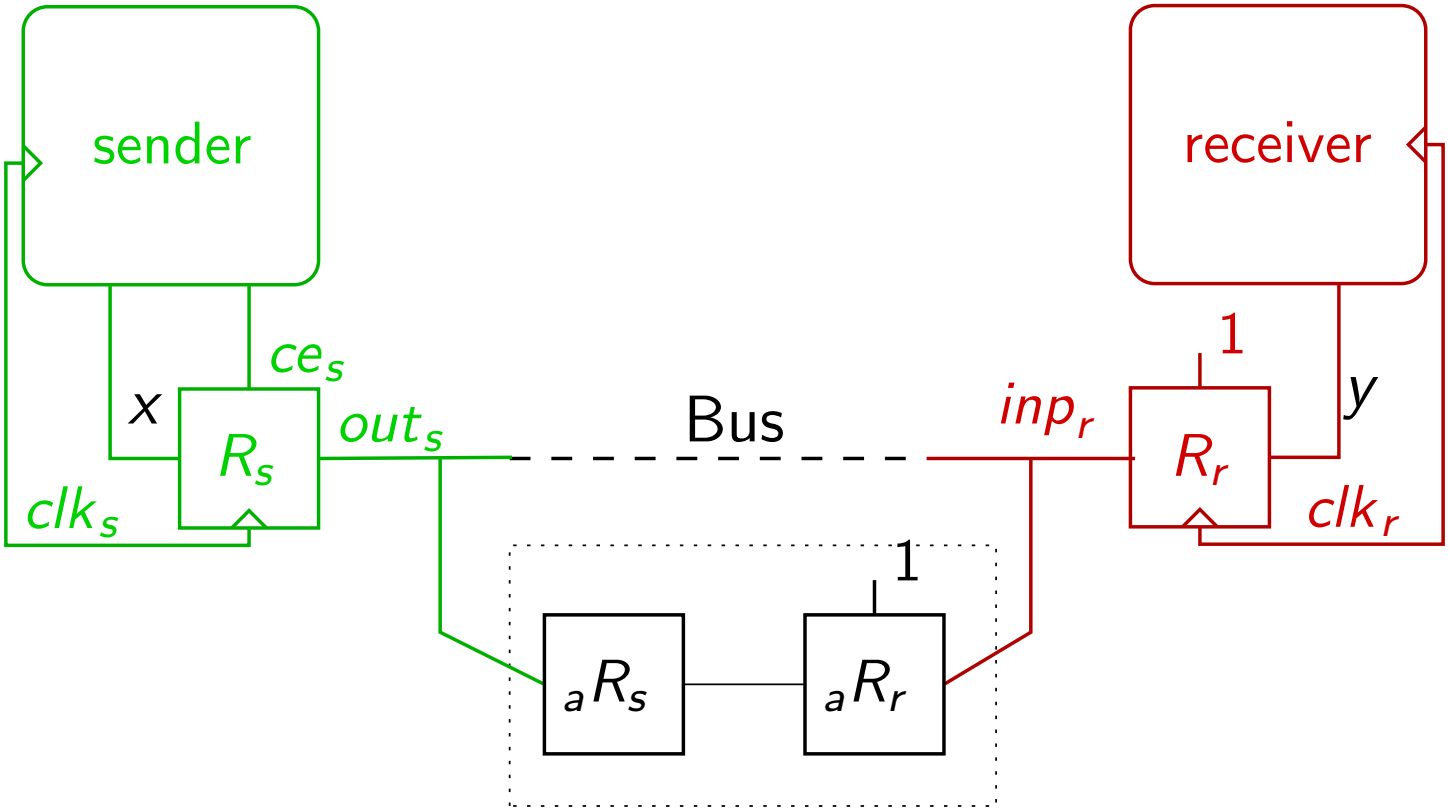
- Goal: insert CDC model without modifying designs

CDC Model and Hardware Designs



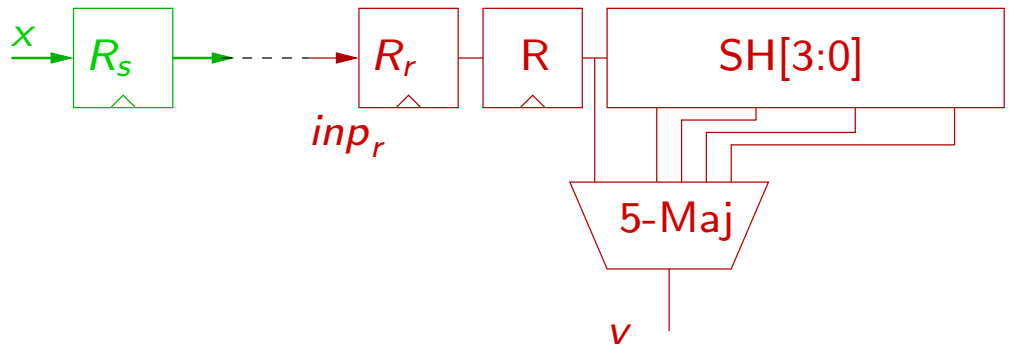
- Goal: insert CDC model without modifying designs
- 2 digital transitions to "move" x to y

CDC Model and Hardware Designs



- 2 digital transitions to "move" x to y
- One analog register function matched to one digital transition
- Designs not modified

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

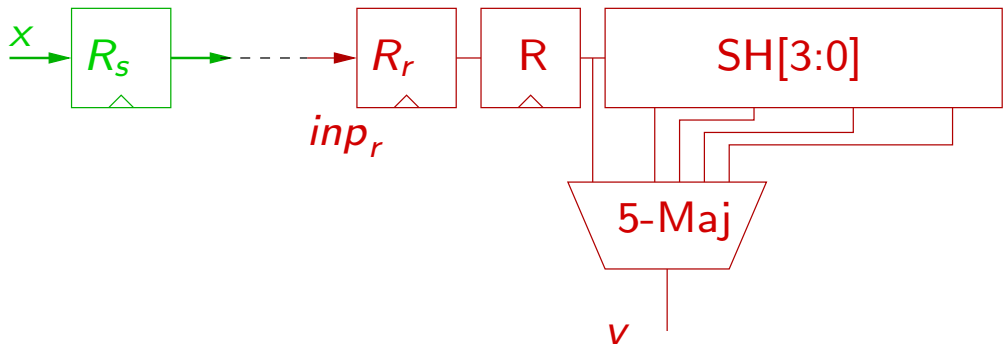
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

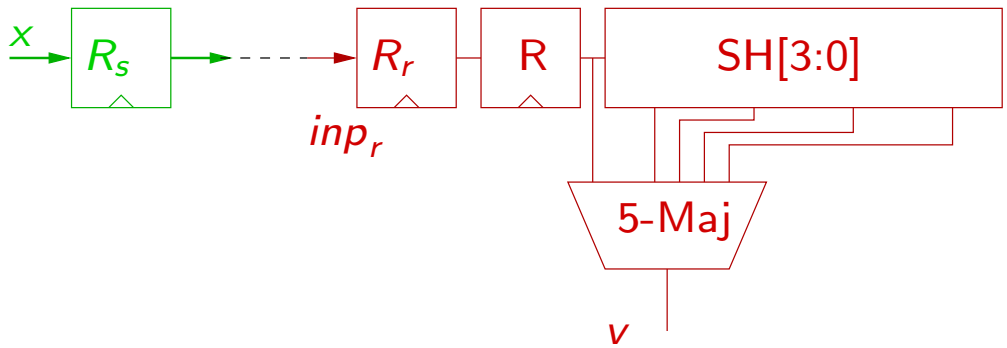
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

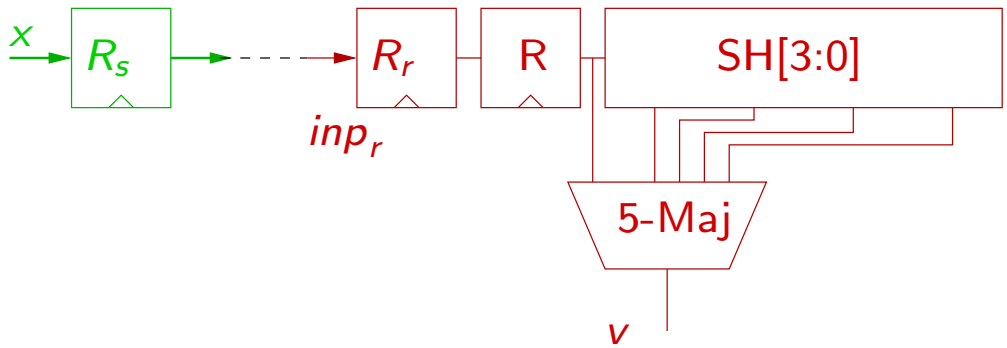
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

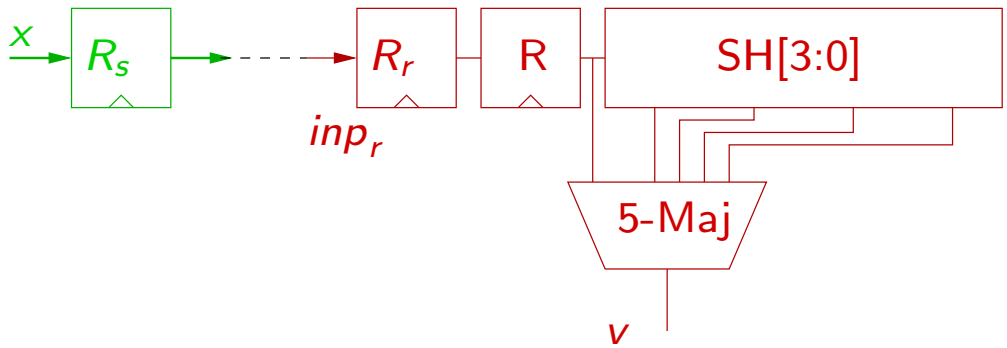
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

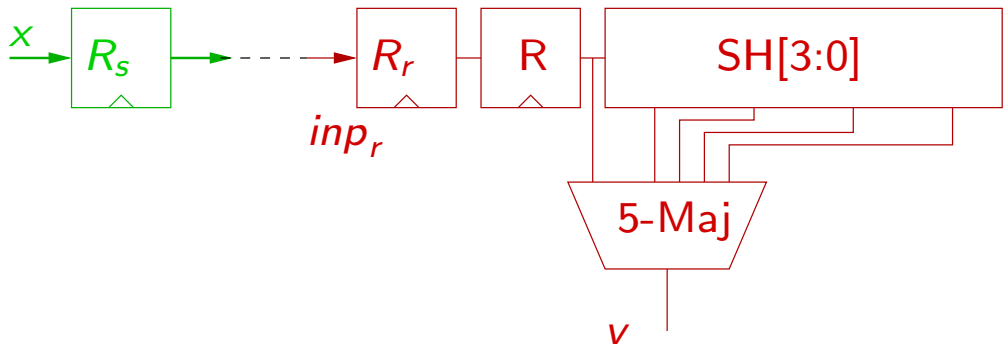
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Example: Majority Voting



- Using NuSMV

$$inp_r^{t+[0:6]} = x \text{ implies } v^{t+[4:10]} = x$$

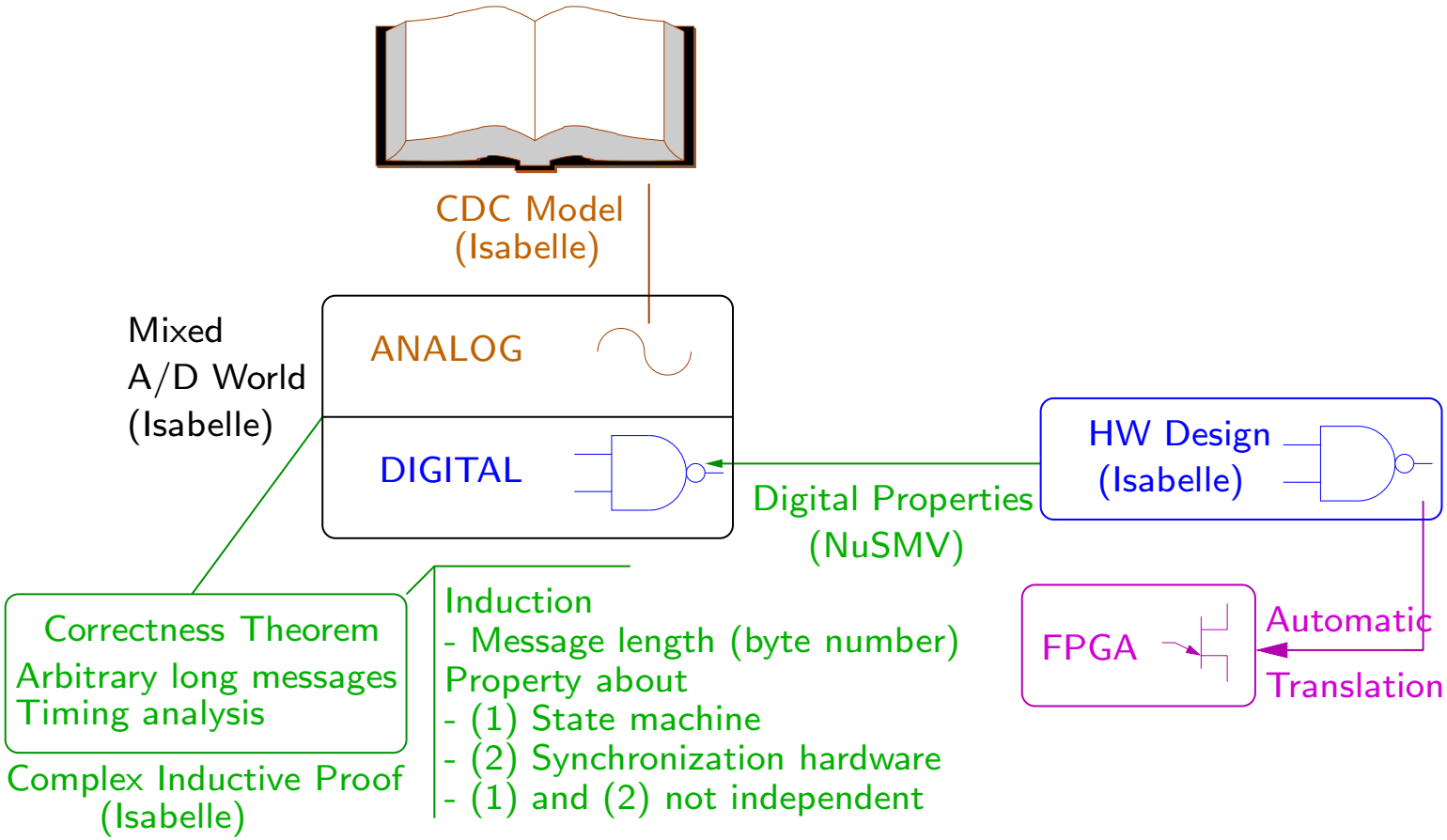
- In Isabelle
  - Insert CDC model for sender cycle c and $cy(\xi, c)$

$$inp_s^{c+[0:7]} = x \text{ implies } inp_r^{\xi+\beta_c^\xi+[0:6]} = x$$

  - then we insert NuSMV result

$$inp_s^{c+[0:7]} = x \text{ implies } v^{\xi+\beta_c^\xi+[4:10]} = x$$

Outline: Final Correctness Proof



Correctness Theorem: Overview

- Functional Correctness
  - For each byte, there exists one receiver cycle from which the byte is correctly sampled
  - This takes 79 to 82 cycles

    Factor $\chi \in \{-1, 0, +1\}$
    Factor $\beta \in \{0, +1\}$
  - Valid counter values: $1 \leq$ (strobe - reset) ≤ 3
- Timing Analysis
  - Derived from functional correctness
    $\longrightarrow$ when receiver affected by first bit of last byte
    $\longrightarrow$ number of cycles to finish transmission
  - Bounded drift used to bound transmission time

Functional Correctness: Proof Overview

- Lemma 1: Traversing synchronization edges
  - Transition from $\mathrm{BSS}[0]$ to end of $\mathrm{BSS}[1]$
  - Synchronization actually takes place
- Lemma 2: Sampling expected values
  - Synchronization is good enough
- Proof Method
  - CDC model: number of unknown inputs (systematic)
  - Unknown inputs are assumptions for NuSMV (automatic)

Conclusion(1)

- General model of clock domain crossing
  - Isabelle/HOL (Isar) theory (1,000 loc)
  - Reusable for other proofs (*e.g.* scheduler)
  - Fully parameterized
- Formal correctness proof of a hardware FlexRay-like interface
  - First detailed gate-level proof: functionality + timing
    - Valid values for crucial parameter
  - Basis theorem for the verification of *distributed stacks*
  - Theorem proving and automatic tools (like model checking)

Conclusion (2)

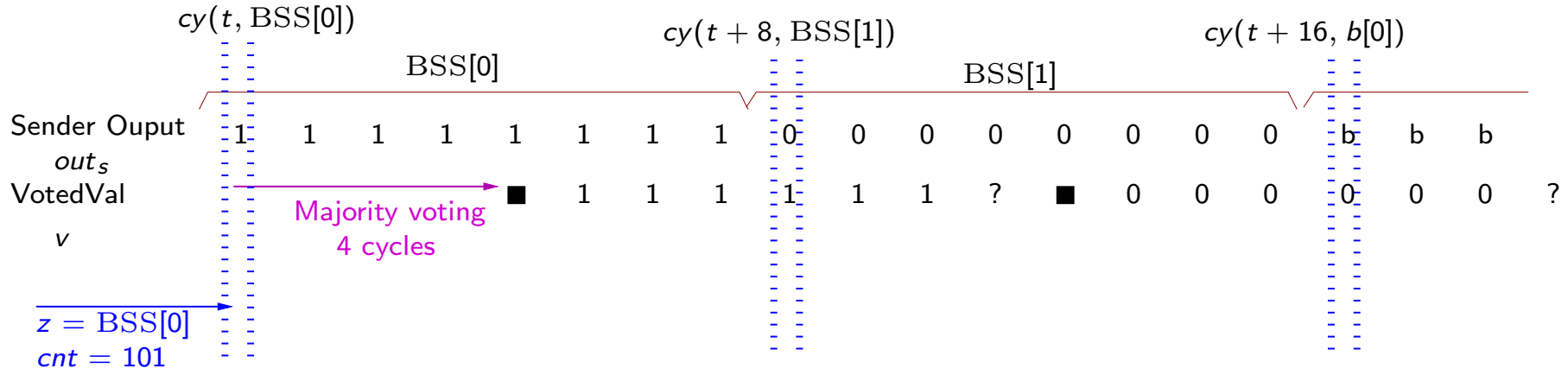
- Practical experience of hybrid verification
  - Automatic tools were crucial
  - Automatic tools must be extremely fast (seconds not minutes)
  - Easy interaction with tactic based theorem prover (Isabelle/Isar)
  - Automatic tools are just new tactics
- Developing the model was the main effort
  - Understanding of the details
  - Deciding between wrong implementation or incomplete model
  - Model can still be improved (spikes, faults)
- From the model the proof of the hardware is systematic
  - General model: exactly where automatic tools apply
  - From first proofs: systematic proof techniques
  - Similar design verification effort would take few weeks
  - ... but tedious: receiver proof $>$ 8,000 loc

THANK YOU !!

Functional Correctness: Proof Overview

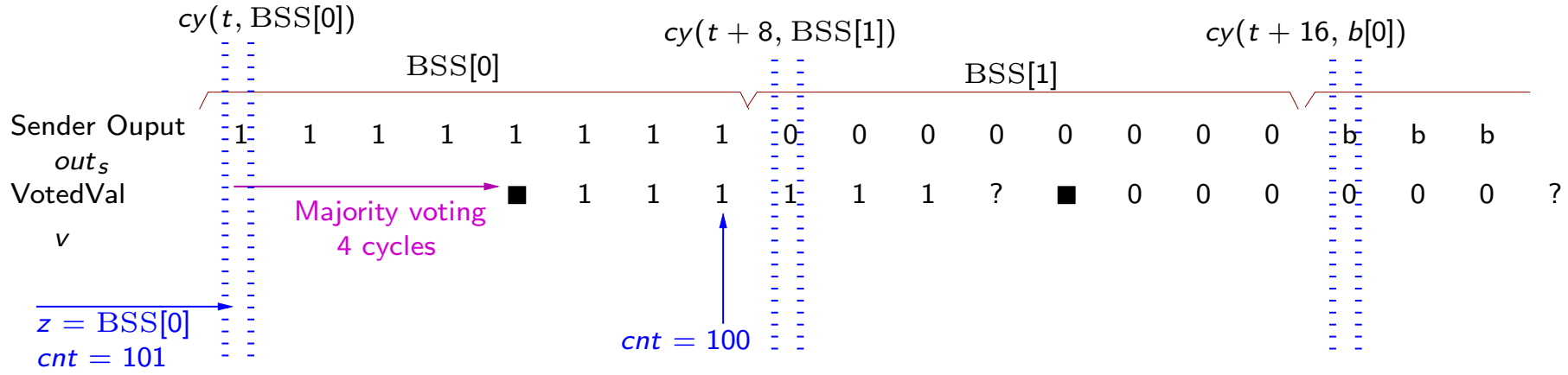
- Show counter-example for the following configuration
  - Counter reset to 000
  - Strobe at 100
  - Strobing distance = 4 - 0 = 4
- FlexRay specifications
  - Counter reset to 010
  - Strobe at 101
  - Strobing distance = 5 - 2 = 3
- Lemma 1: Traversing synchronization edges
  - Transition from $\mathrm{BSS}[0]$ to end of $\mathrm{BSS}[1]$
  - Synchronization actually takes place
- Lemma 2: Sampling expected values
  - Synchronization is good enough

Traversing Synchronization Edges



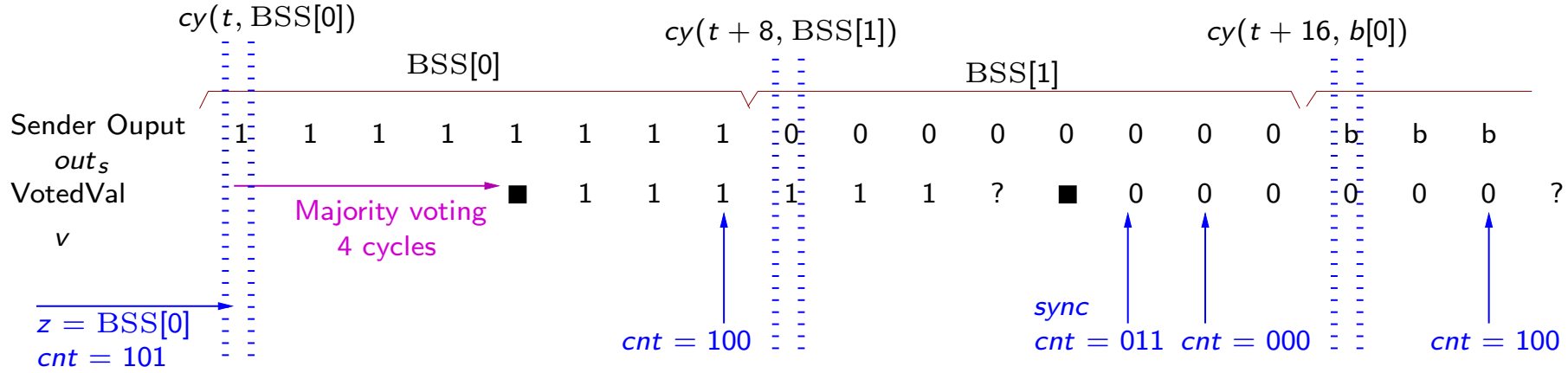
- out_s = sender output, v = voted bit, z = receiver state
- Delay of 4 cycles from majority voting

Traversing Synchronization Edges



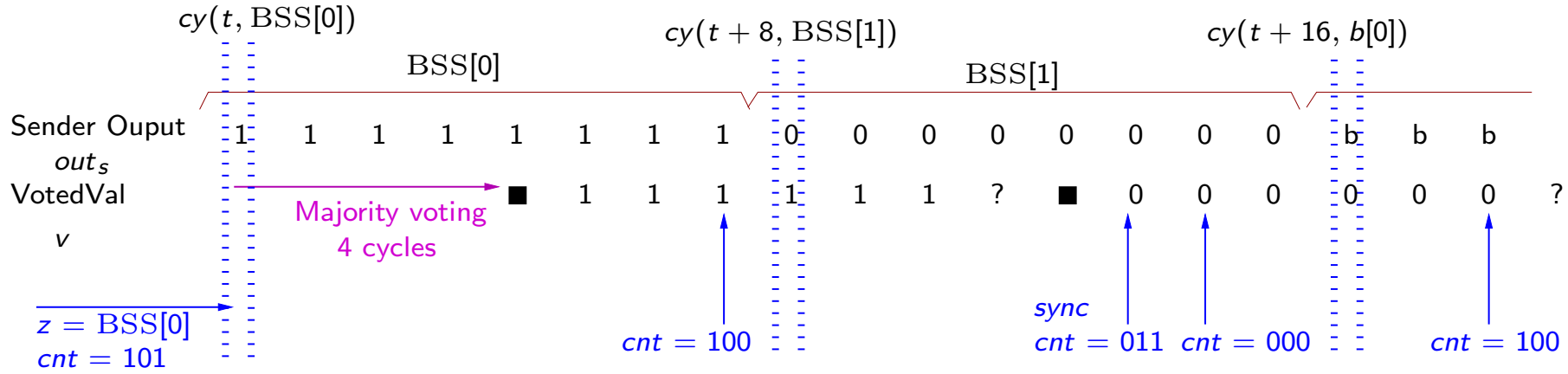
- out_s = sender output, v = voted bit, z = receiver state
- Delay of 4 cycles from majority voting
- Strobe at 100

Traversing Synchronization Edges



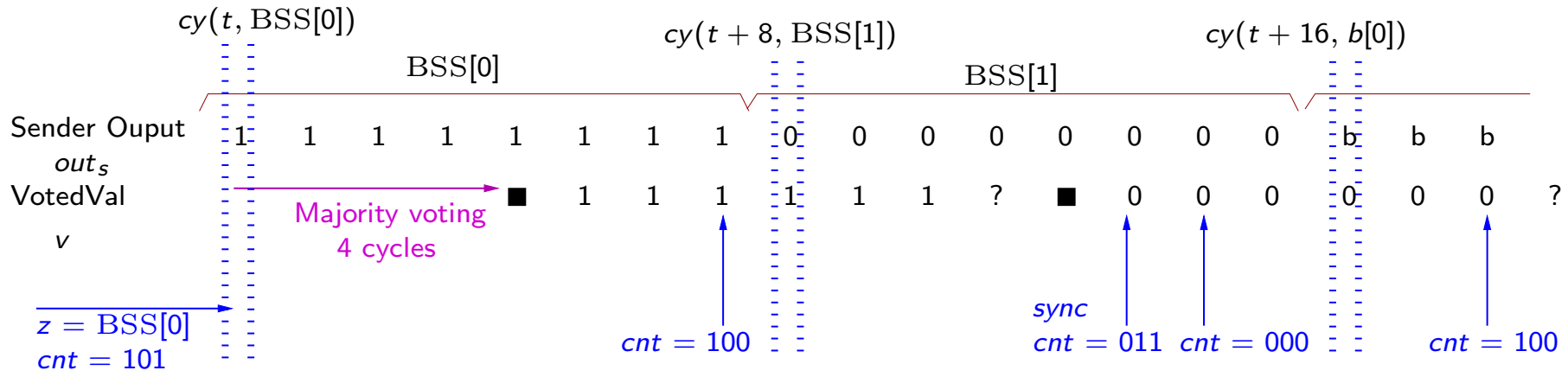
- At $t + 13$, *sync* is high (falling edge detected)
- Counter *cnt* reset to 000
- Strobe at $t + 18$

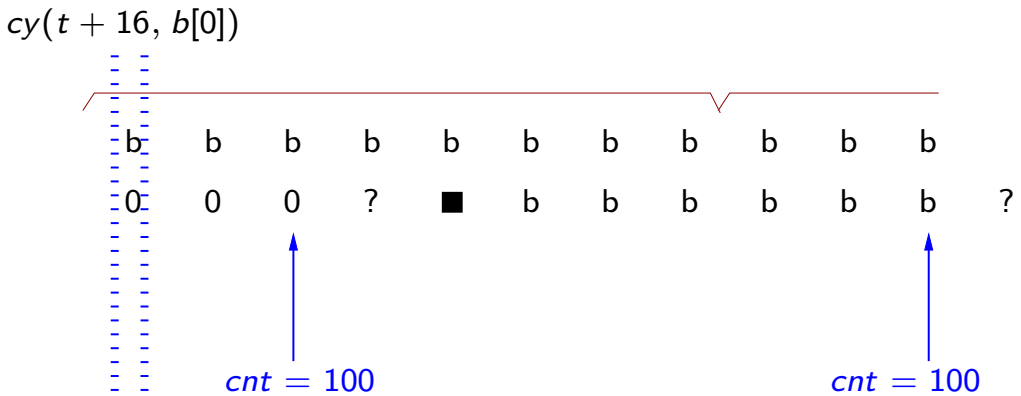
Traversing Synchronization Edges



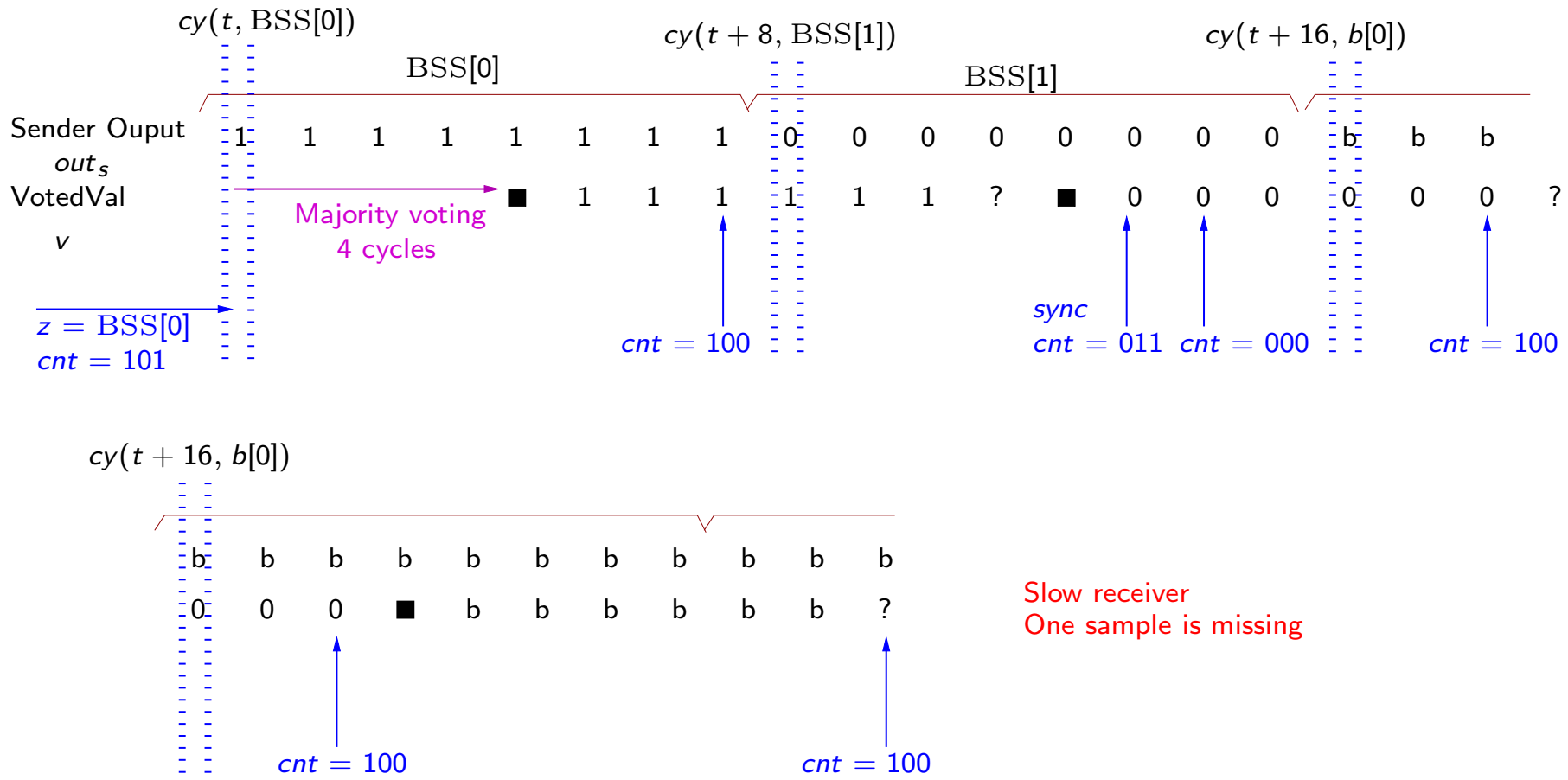
- At $t + 13$, *sync* is high (falling edge detected)
- Counter *cnt* reset to 000
- Strobe at $t + 18$
- Lemma 1:
  - 15 to 18 cycles from t to second strobing point
  - Assuming drift, jitter and metastability
- Proof by NuSMV and Isabelle/HOL
  - CDC model: 1 or 2 unknown inputs (systematic)
  - Unknown inputs are assumptions for NuSMV proof (automatic)

Sampling Good Values: Counter-Example



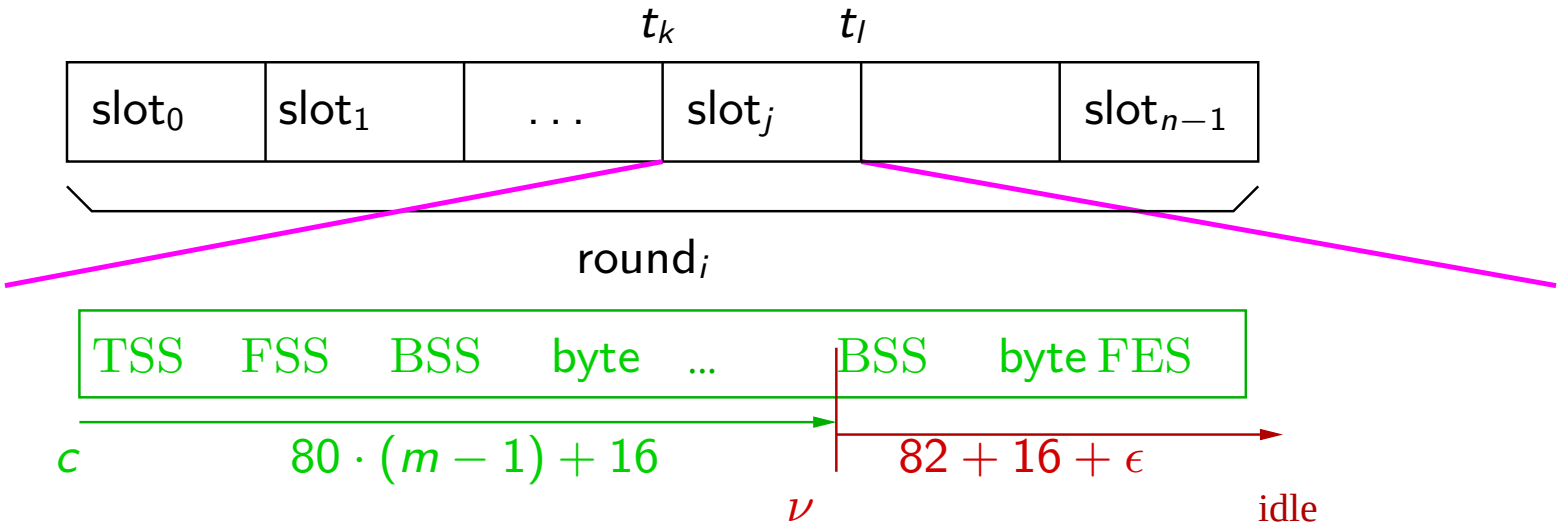


Sampling Good Values: Counter-Example



Slow receiver
One sample is missing

Timing Correctness



- Transmission correctness theorem:
  - For all bytes b, there exists a receiver cycle ν from which b is correctly sampled after 79 to 82 (receiver) cycles.
  - Note: we have $cy(\nu, c + 80 \cdot (m-1) + 16)$
- Timing theorem easily follows:
  - Number of transmission cycles $t = 32 + 80 \cdot (m-1) + 82 + \epsilon$
  - Bound on maximum length of clock periods $\tau_{max} = (1 + \delta) \cdot \tau_{ref}$
  - Transmission time bounded by the following:

$$(32 + 80 \cdot m + 2 + \epsilon) \cdot (1 + \delta) \cdot \tau_{ref}$$