Security Issues Arising from Hardware Design

Joerg Bormann, Siemens EDA
joerg.bormann@siemens.com

SIEMENS

Motivation

CWE calls it a weakness, if code does does not comply to good coding practice:

- CWE-1099: Inconsistent Naming Conventions for Identifiers
- CWE-1109: Use of Same Variable for Multiple Purposes
- CWE-1113: Inappropriate Comment Style
- CWE-1114: Inappropriate Whitespace Style
- CWE-1116: Inaccurate Comments

Proposal to initiate new CWE entries about the security impact of bad coding practice in HW designs

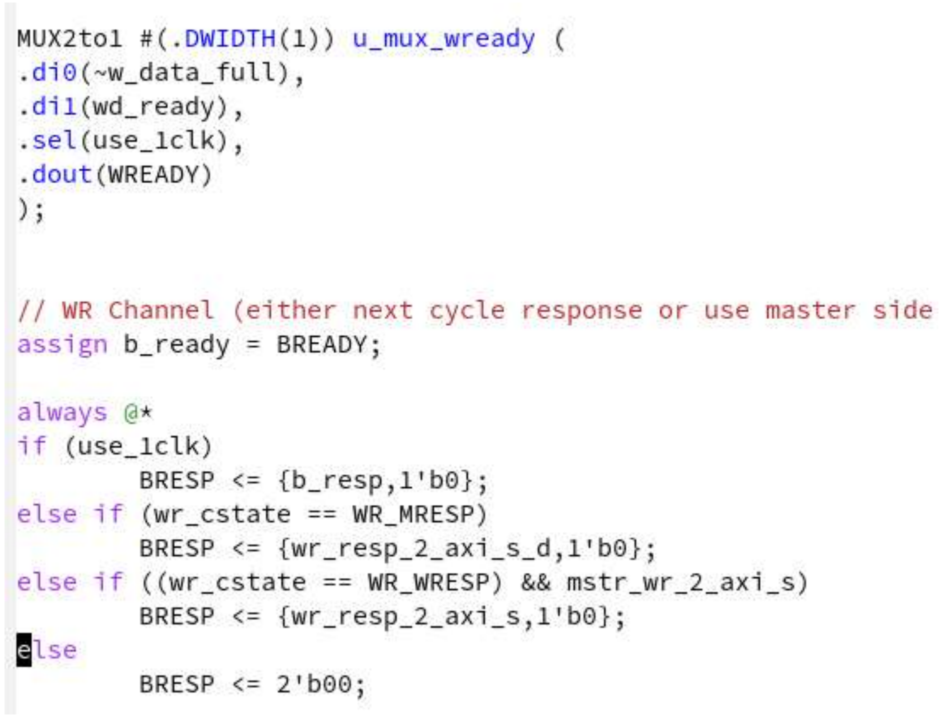
- Simulation / Synthesis Mismatches
- Unused logic
- Inappropriately verified circuits with Rams, Latches, or Registers without Reset
- Combinatorial logic with too high propagation delay
- Combinatorial loops in HW
- Inappropriate Clock Domain Crossings

These weaknesses can make HW behave other than verified. The different behavior can violate security.

Technical Background

SIEMENS

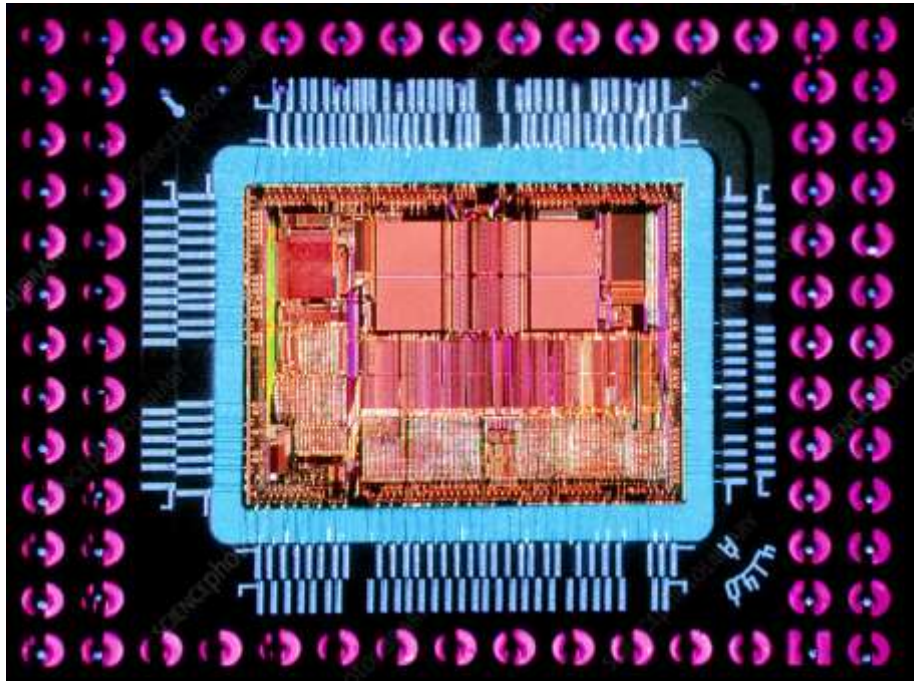
RTL and how it is implemented



```
MUX2to1 #(.DWIDTH(1)) u_mux_wready (
.di0(~w_data_full),
.di1(wd_ready),
.sel(use_1clk),
.dout(WREADY)
);


// WR Channel (either next cycle response or use master side
assign b_ready = BREADY;

always @*
if (use_1clk)
        BRESP <= {b_resp,1'b0};
else if (wr_cstate == WR_MRESP)
        BRESP <= {wr_resp_2_axi_s_d,1'b0};
else if ((wr_cstate == WR_WRESP) && mstr_wr_2_axi_s)
        BRESP <= {wr_resp_2_axi_s,1'b0};
else
        BRESP <= 2'b00;
```

RTL Description

- Looks like a program
- Well-defined simulation semantics

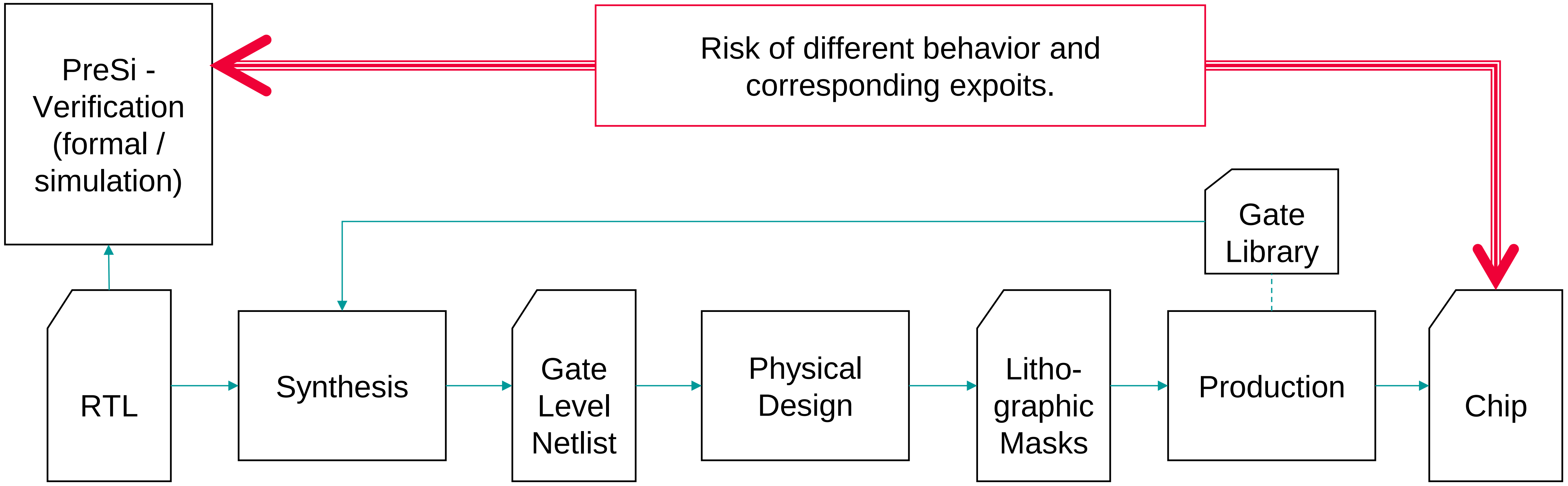


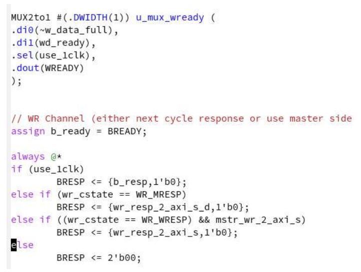
source: sciencephoto.com

Implemented by a phyiscal device

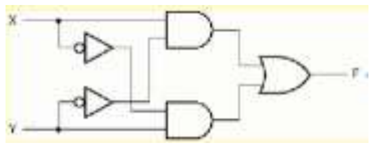
- Transistors, resistors, capacities, wires, Millions of them
- Many influence factors: Temperature, Radiation, ...
- Analog view required to capture all potential behavior

Creation of an Integrated Circuit from RTL

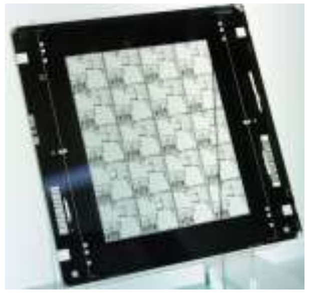

PreSi - Verification (formal / simulation)
Risk of different behavior and corresponding expoits.
Gate Library
RTL
Synthesis
Gate Level Netlist
Physical Design
Litho-graphic Masks
Production
Chip



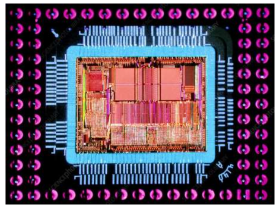
```
MUX2to1 #(.DWIDTH(1)) u_mux_wready (
.di0(~w_data_full),
.di1(wd_ready),
.sel(use_1clk),
.dout(WREADY)
);


// WR Channel (either next cycle response or use master side
assign b_ready = BREADY;

always @*
if (use_1clk)
        BRESP <= {b_resp,1'b0};
else if (wr_cstate == WR_MRESP)
        BRESP <= {wr_resp_2_axi_s_d,1'b0};
else if ((wr_cstate == WR_WRESP) && mstr_wr_2_axi_s)
        BRESP <= {wr_resp_2_axi_s,1'b0};
else
        BRESP <= 2'b00;
```



Source
https://webdocs.cs.ualberta.ca/~amaral/courses/329/webslides/Topic9-3StateBuffers/img034.gif



Source:
https://en.wikipedia.org/wiki/Photomask#/media/File:Semiconductor_photomask.jpg



Related Weaknesses

SIEMENS

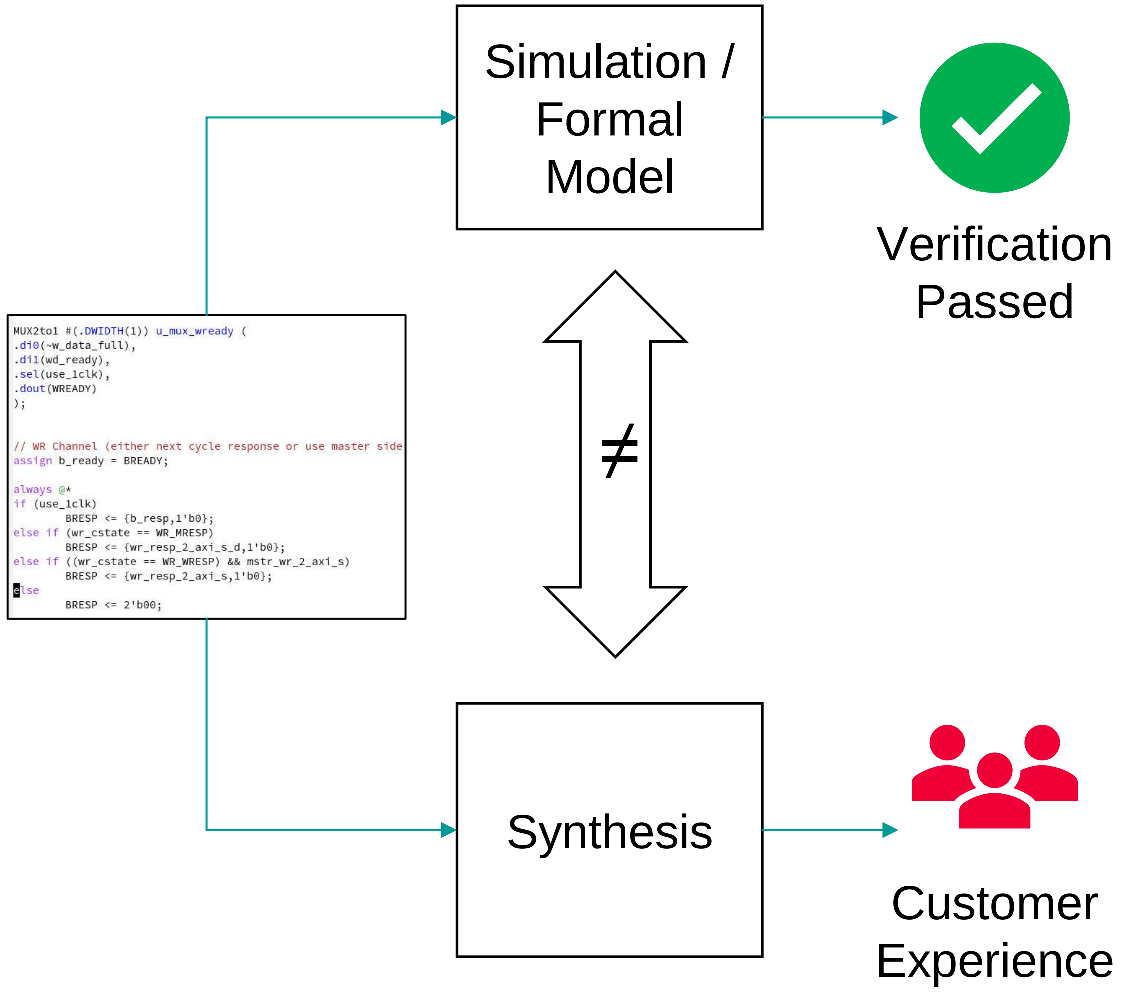
Simulation / Synthesis Mismatches

RTL simulation and synthesis have different semantics.

Main mitigation: Syntactic restriction to „Synthesizable subset“.

Still there are simulation / synthesis mismatches from certain combinations of constructs:

- RTL for combinatorial logic is not ordered according to dependency
- Use of X
- See Mills, Cummings: RTL Coding Styles That Yield Simulation and Synthesis Mismatches
  http://www.sunburst-design.com/papers/CummingsSNUG1999SJ_SynthMismatch.pdf



```
MUX2to1 #(.DWIDTH(1)) u_mux_wready (
.di0(~w_data_full),
.di1(wd_ready),
.sel(use_1clk),
.dout(WREADY)
);


// WR Channel (either next cycle response or use master side
assign b_ready = BREADY;

always @*
if (use_1clk)
        BRESP <= {b_resp,1'b0};
else if (wr_cstate == WR_MRESP)
        BRESP <= {wr_resp_2_axi_s_d,1'b0};
else if ((wr_cstate == WR_WRESP) && mstr_wr_2_axi_s)
        BRESP <= {wr_resp_2_axi_s,1'b0};
else
        BRESP <= 2'b00;
```

SIEMENS

Simulation / Synthesis Mismatches and CWE

Current CWEs:

CWE-1298: Hardware Logic Contains Race Conditions

- Seems not to be related to Simulation / Synthesis Mismatches

Proposed New CWE entry:

Description: RTL code contains parts that have a different behavior in simulation than on the synthesized hardware, so called Simulation-Synthesis Mismatches.

Common Consequences: The RTL verification is usually based on the simulated behavior. If HW behavior is different from the simulated behavior, it is not verified, and may contain security critical behavior.

Potential Mitigation: Use of an appropriate coding standard, and verification tools to detect deviations from this coding standard.

Synthesis

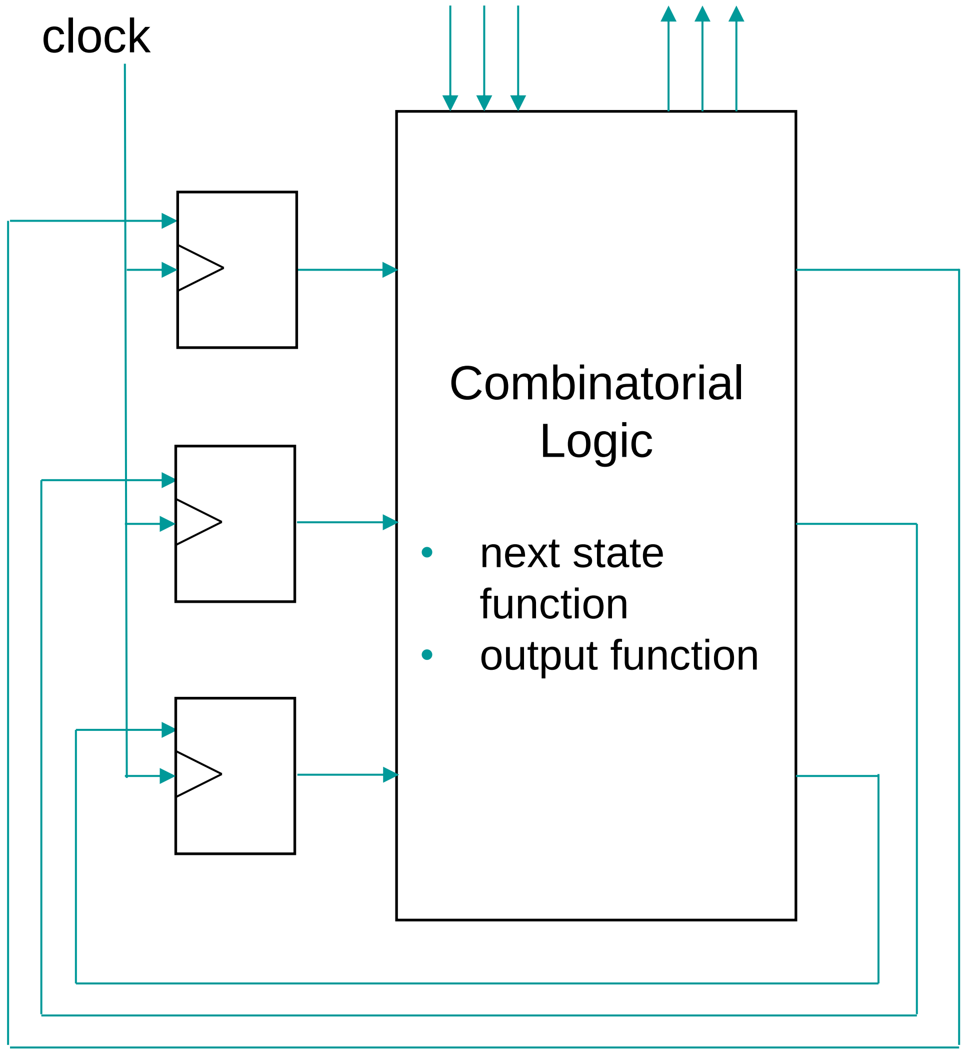
Maps RTL description on a finite state machine

- state bits
- next state function
- output function

Clock initiates state changes.

Functionality of a synthesis tool:

- State Inference: Identifies state bits by syntactic pattern matching. State bits are mapped to register gates, latch gates, or RAMs.
- Logic synthesis: Rest of the RTL is the next state and output function. Logic synthesis turns it into a netlist of combinatorial logic gates: e.g., AND, OR, NOT



HW Specific Weaknesses of the FSM implementation: Unused Logic

Unused logic never influences any output.

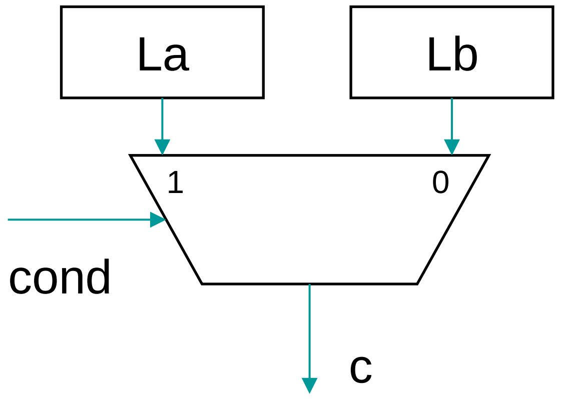
Synthesis tools identify and remove some unused logic, but not all.

Unused logic is active and creates the following attack surfaces:

- Support of power & radiation side channels.
- Create local overheating to insert faults.
- Waste energy to implement an availability attack on battery operated product.

```
if (cond) then
          c = <logic La>;
else
          c = <logic Lb>;
```

becomes



Lb is always active, even if cond is always 1.

Unused Logic and CWE

Current CWEs

CWE-561: Dead code - Unused logic is always active while dead code is never executed.

CWE-563: Assignment to Variable without Use - Description highlights „dead store“, „bad quality“, and the risk of „further bugs and ... weaknesses“, but not the HW specific attack vectors.

CWE-1164: Irrelevant code - Common Consequences mention an impact on „Reliability“ and „Performance“, but not the HW specific attack vectors.

Proposed new CWE: Unused logic

Description: The hardware contains unused logic, i.e. logic that never influences any output.

Extended Description: Although the results produced by unused logic are nowhere used, the logic itself is active.

Common Consequences:

- Support of power and radiation side channels
- Fault insertion attacks by local overheating
- Attacks on the availability of battery operated devices.

HW Specific Weaknesses of the FSM implementation: Storage without reset

Storage elements without Reset are often an economic necessity.

- RAMs are cheaper than registers, but not resettable.
- Cost of reset wiring grows with the number of connected registers.

On purpose, some storage elements have no reset.

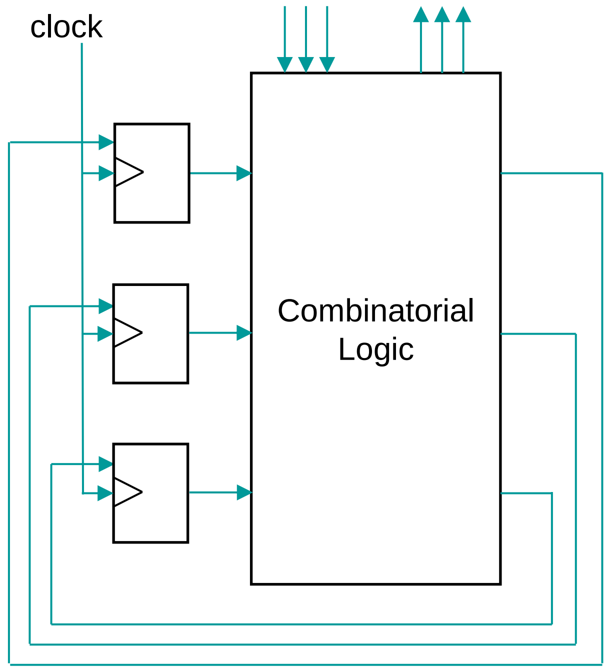
- Then the circuit has multiple reset states.
- Verification must cover all reset states.

Weakness: Failure to cover all reset states in verification may lead to sporadic unverified hardware behavior that starts in one of the uncovered reset states.

Note on initialization by X to represent „arbitrarily either 0 or 1“

- X does not make dependency on missing initialization obvious.
- See Turpin, The Dangers of Living with an X (bugs hidden in your Verilog)

  https://www.researchgate.net/publication/240753489_The_Dangers_of_Living_with_an_X_bugs_hidden_in_your_Verilog



SIEMENS

Storage without Reset & CWE

Related current CWEs:

CWE-1271: Uninitialized Value on Reset for Registers Holding Security Settings – limited to registers that hold security settings. But there are more reasons for different behavior that depends on the initial state.

CWE-1419: Incorrect Initialization of Resource – A resource is initialized, but not as intended. This is different from the danger that arises from different behavior depending on initial state.

CWE-457: Use of Uninitialized Variable – All uninitialized variables are considered bugs. All mitigations aim at prevention of these bugs. The entry does not acknoweldge the economic advantage.

Option 1) Extend CWE-457 by the HW specific situation.

Option 2) New CWE: Storage Elements without Reset

Description: The design contains storage elements without reset, but the verification fails to cover all states that the design could be in after reset.

Common Consequences: Depending on the arbitrary choice of the state that the design assumes after reset, the design might exhibit unverified behavior that could have security issues.

Digital Synchronous Abstraction

Digital abstraction:

- 1 = Voltage level above threshold T1
- 0 = Voltage level below threshold T0

Synchronous abstraction

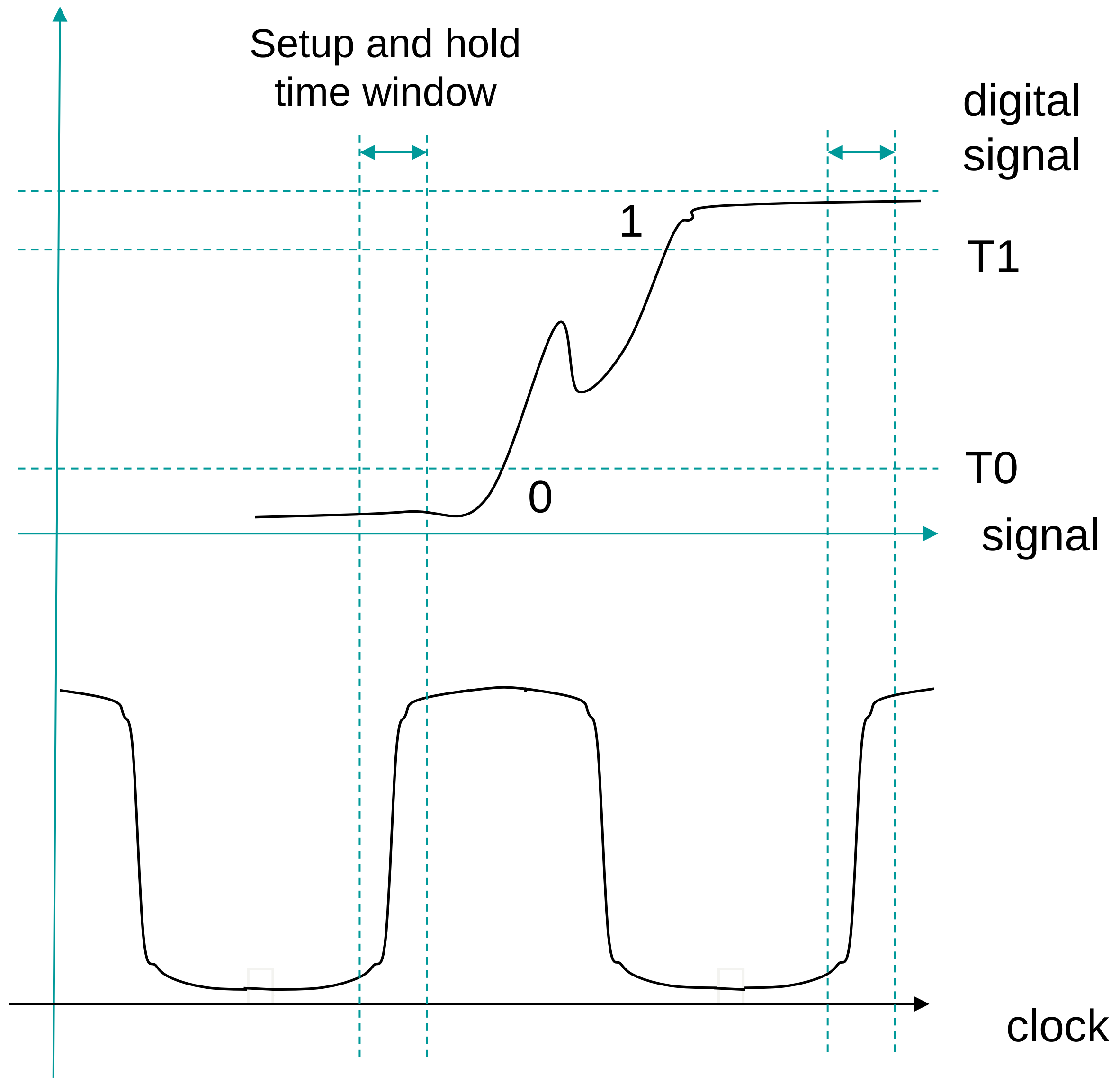
- Voltage of signals only relevant in the setup and hold time window around a rising clock edge.

Soundness of the abstraction:

- In setup and hold time windows, signals == 0 or 1.
- Combinatorial logic must be fast enough to complete calculations before the next time window. (= Propagation Delay Requirement)

Propagation delay requirement hides all influence onto the physical process (see next slide).

Failure to satisfy the propagation delay requirement is a weakness.



Influence Factors for the Violation Propagation Delay Requirement

Influence Factor	Related Weakness
RTL design	• complex combinatorial logic. • combinatorial feedback loops. • inappropriate clock domain crossing. } see next slides
Clock Distortions	Clock signals have spikes that trigger register writes too early. Captured by CWE-1247: Improper Protection Against Voltage and Clock Glitches.
Placement	Unsuitable placement leads to long wires that make propagation delay too high.
Supply Power	Globally: Less supply voltage => higher propagation delay. Locally: Sudden local high power consumption decreases local supply voltage. Partly capt‘d by CWE-1233: Security Sensitive HW Controls with missing ... protection – but any supply power insufficiency is a problem.
Temperature	Globally: Circuit operates in hot environment => higher propagation delay. Locally: Busy part of the circuit becomes hot.
Cross Talk	Electrical fields from one signal disturb other signals in the neighbourhood.
Radiation	Induces voltage changes in signals.

Combinatorial Logic with a Propagation Delay beyond clock cycle time

No related CWE issue found.

Proposed new CWE entry:

Description: The circuit contains combinatorial logic with a propagation delay beyond the clock cycle time.

Extended Description: Around a rising clock edge the logic may create voltage levels between 0 and 1 at the input of some registers. The registers and hence the whole circuit will then show unpredictable behavior that might involve security issues.

Combinatorial Feedback Loops

No related CWE issue found.

Proposed new CWE entry:

Description: The circuit has combinatorial feedback loops.

Extended Description: An activated combinatorial feedback loop may create voltage levels between 0 and 1 oat the input of some registers around the rising edge of their clock. The registers and hence the whole circuit will then show unpredictable behavior that might involve security issues.

Inappropriate Clock Domain Crossings

Inappropriate Clock Domain crossings can cause unpredictable behavior and losses of data integrity, allowing exploits that violate security.

No related CWE issue found.

Proposed first new CWE entry: Inappropriate handling of a control signal in a clock domain crossing

A control signal is passed from Clock Domain A to Clock Domain B without double synchronizer or protective protocol. This will lead to unspecified circuit behavior, that can be security relevant.

Mitigation: Use of special gates to protect clock domain crossings (so-called double synchronizers).

Proposed second new CWE entry: Inappropriate handling of a data signal in a clock domain crossing

Data is passed from clock domain A to clock domain B an via parallel double synchronizers for each bit.

Extended Description: For certain timings between the rising clocks of clock domain A and B, simultaneous changes of the bits of the data signal in A are not recognized simultaneously in B. This impacts data integrity.

Mitigation: Use of specialized clock domain crossing protocols.

Summary

Proposal to initiate new CWE entries about the security impact of bad coding practice in HW designs

- Simulation / Synthesis Mismatches
- Unused logic
- Inappropriately verified circuits with Rams, Latches, or Registers without Reset
- Combinatorial logic with too high propagation delay
- Combinatorial loops in HW
- Inappropriate Clock Domain Crossings

These weaknesses can make HW behave other than verified. The different behavior can violate security.

Contact

Published by Siemens EDA

Joerg Bormann
PM Advanced Verification
DI SW EDA DVT CSF DV FS
Nymphenburger Straße 20a
80335 Muenchen
Germany

Phone +49 1577 356 4108

E-mail joerg.bormann@siemens.com