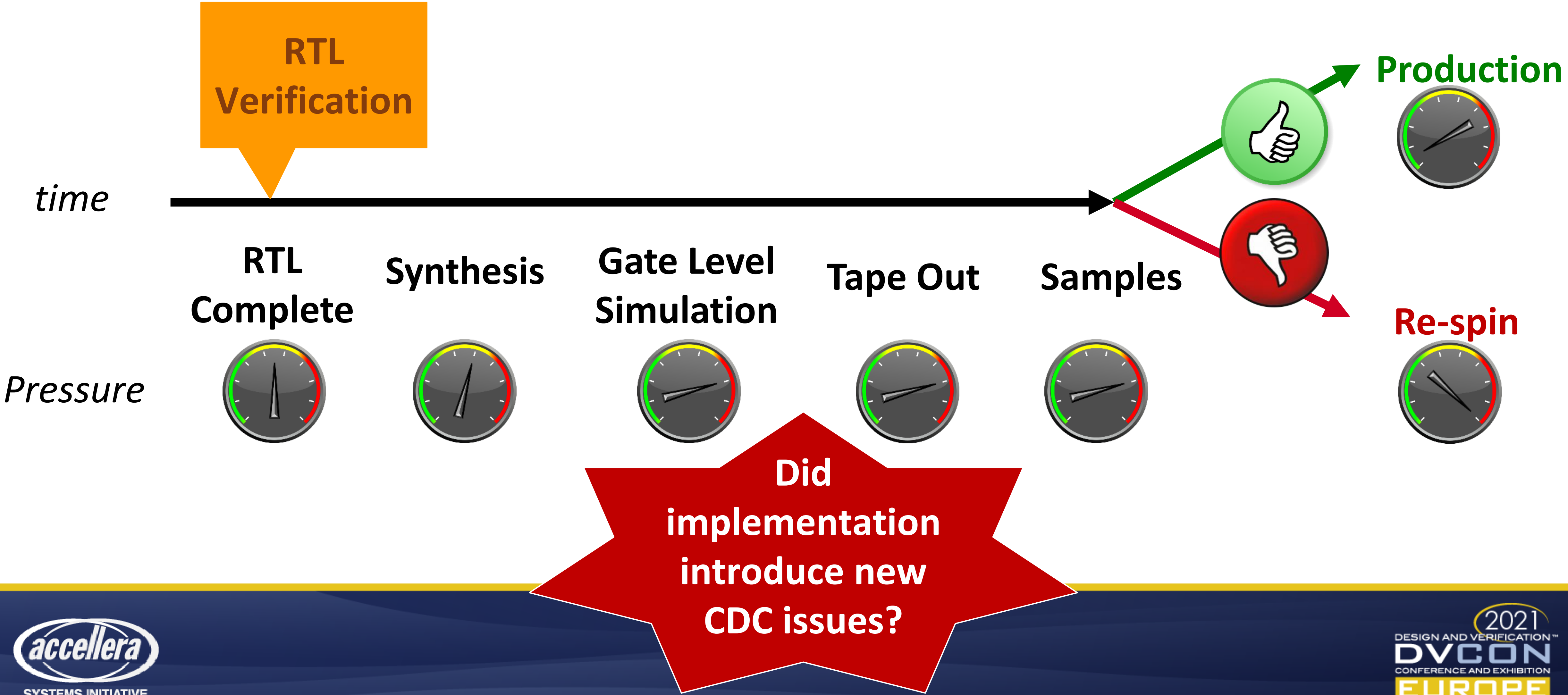
2021
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
OCTOBER 26-27, 2021

Detection of glitch-prone clock and reset propagation with automated formal analysis

Kaushal Shah (Siemens EDA)
Sulabh Kumar Khare (Siemens EDA)

accellera
SYSTEMS INITIATIVE

RTL CDC Verification Is Not Enough



RTL Verification

time

RTL Complete

Synthesis

Gate Level Simulation

Tape Out

Samples

Production

Re-spin

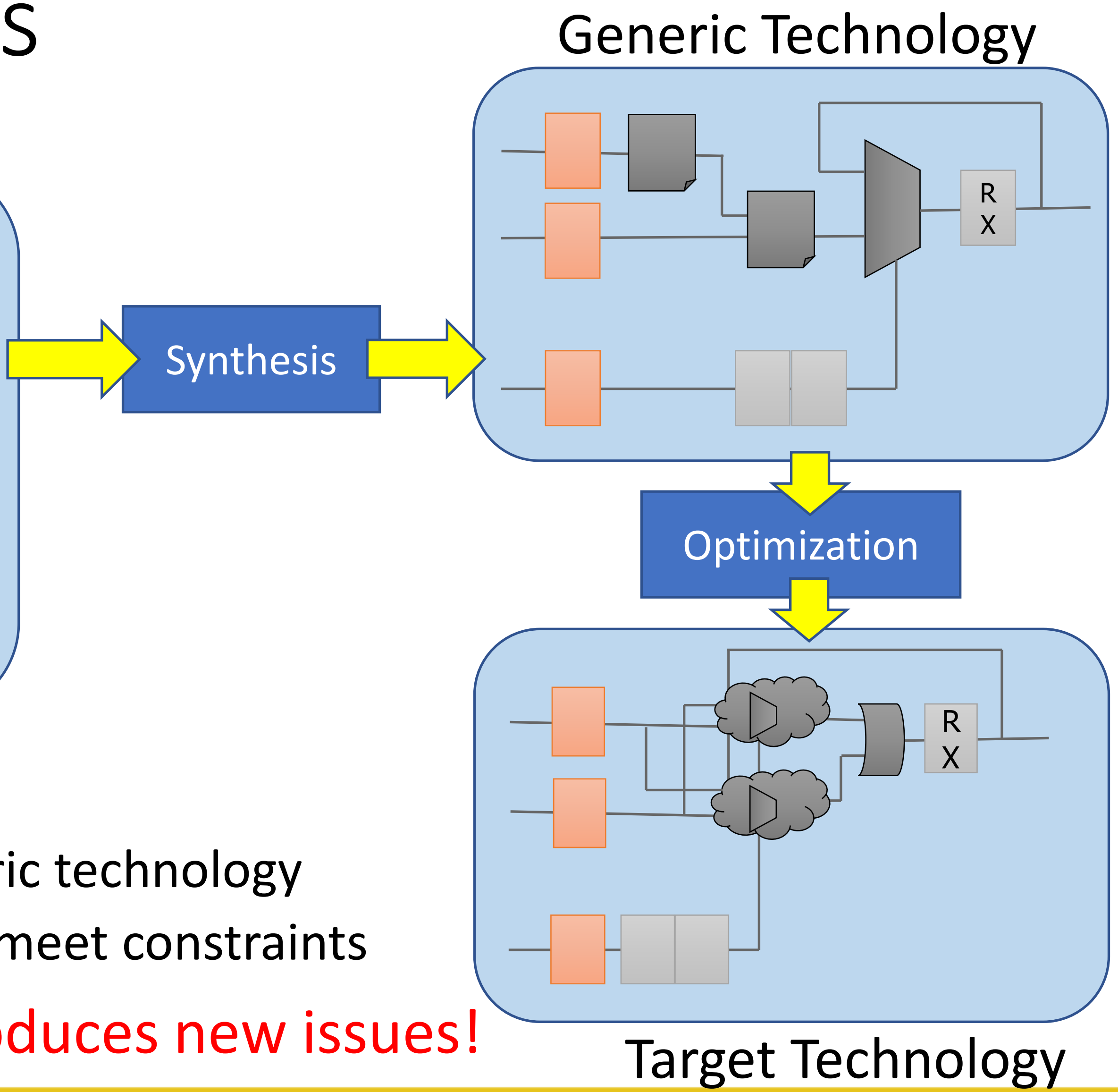
Pressure

Did implementation introduce new CDC issues?

RTL Synthesizes to Gates

RTL

```
always @(posedge rx_clk)
 begin: DMUX
  s1 <= tx_sel;
  rx_sel <= s1;
  if (rx_sel)
   rx_data <= tx_data1 && tx_data2;
  else
   rx_data <= rx_data ^ 4'b1111;
end
```

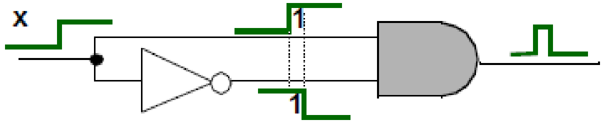


- Designer verifies RTL, but:
  - RTL mapped to logical equivalents in generic technology
  - Further optimized to target technology to meet constraints
- What if the synthesis optimization introduces new issues!

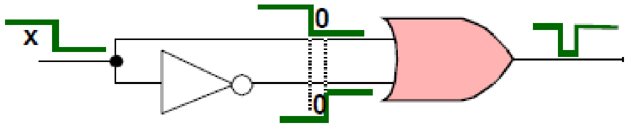
Understanding Structural Glitches

- Static-0 glitch
  - Signal static at 0, rises for glitch
  - Two parallel paths; one inverted
  - Reconvergence at AND gate
- Static-1 glitch
  - Signal static at 1, rises for glitch
  - Two parallel paths; one inverted
  - Reconvergence at OR gate
- Dynamic glitch
  - Signal changes to 0 or 1 after glitch
  - Three parallel paths
  - Contains static glitch

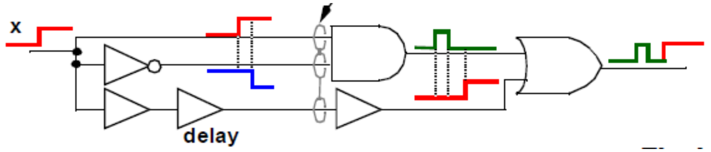
Any combinational logic that can be reduced to any one of the above can cause glitch



X.~X



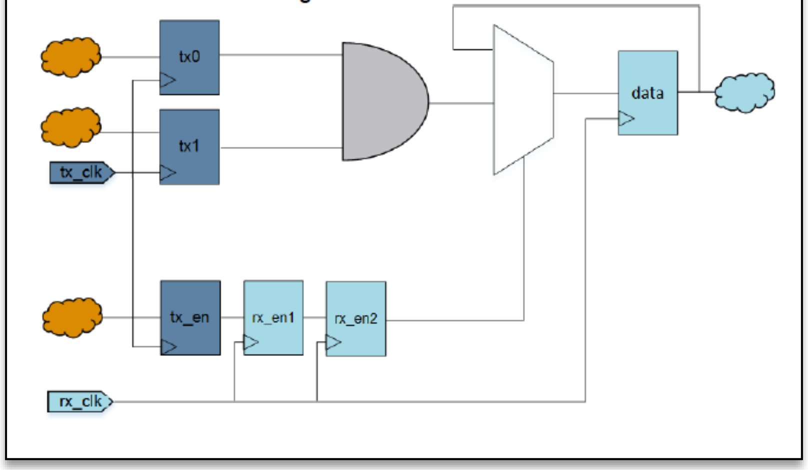
X+~X



(X+~X)X
or
X.~X + X

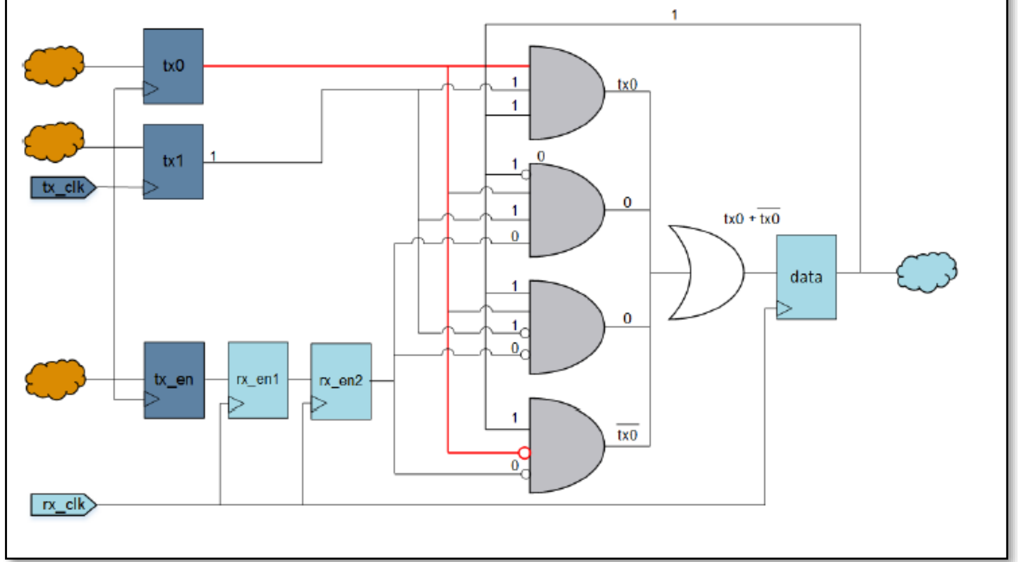
Glitches Can be Introduced on CDC Paths That Were Safe at RTL

Figure 1 Schematic for RTL



Synthesis

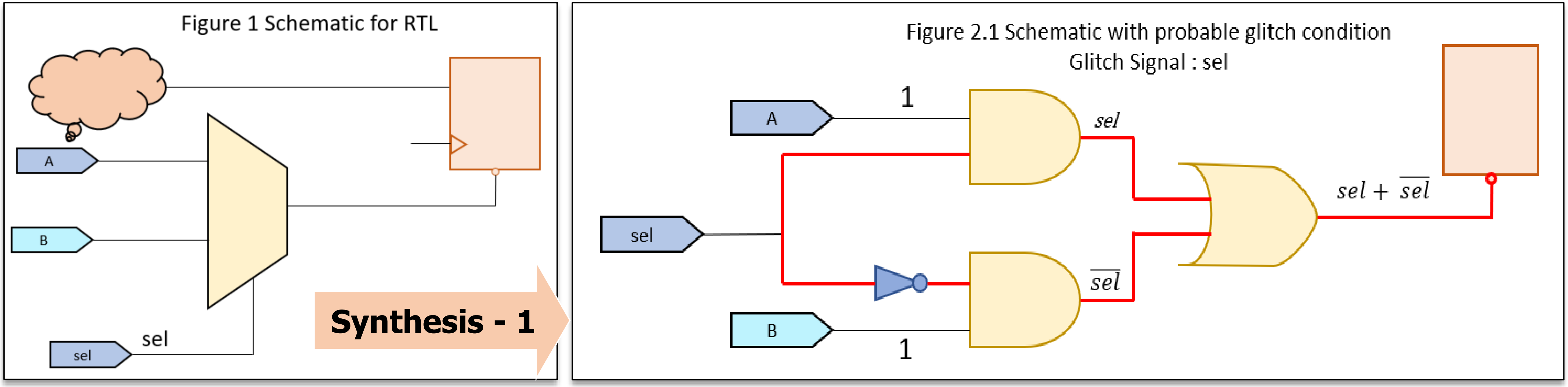
Figure 2 Schematic with Probable Glitch Condition



Synthesis can break corrective circuitry or add surprise paths

- MUX used at RTL for CDC crossing
- Synthesis tool may implement combinational logic which produces glitch
  - X+~X or X.~X
- Potential chip failure issue if glitch is caught by the receiving flip-flop!

Glitches on Clock and Reset Paths



- Problematic for both synchronous and asynchronous paths
- Glitches will cause unexpected resets or unexpected clock edges

Glitches on Clock and Reset Paths

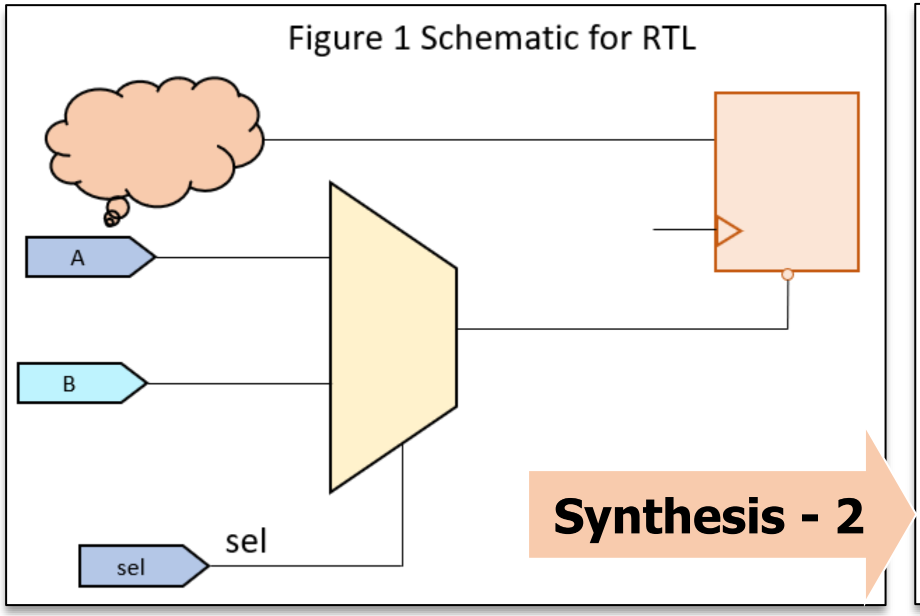


Figure 1 Schematic for RTL

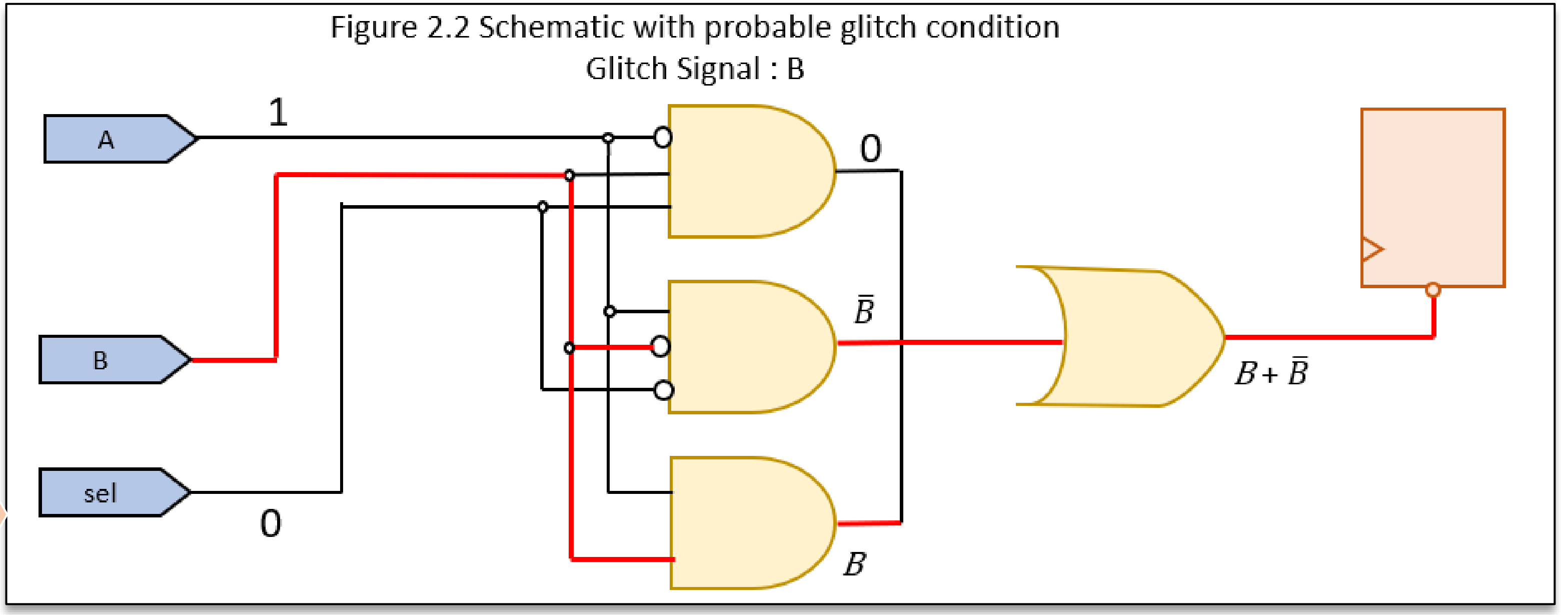


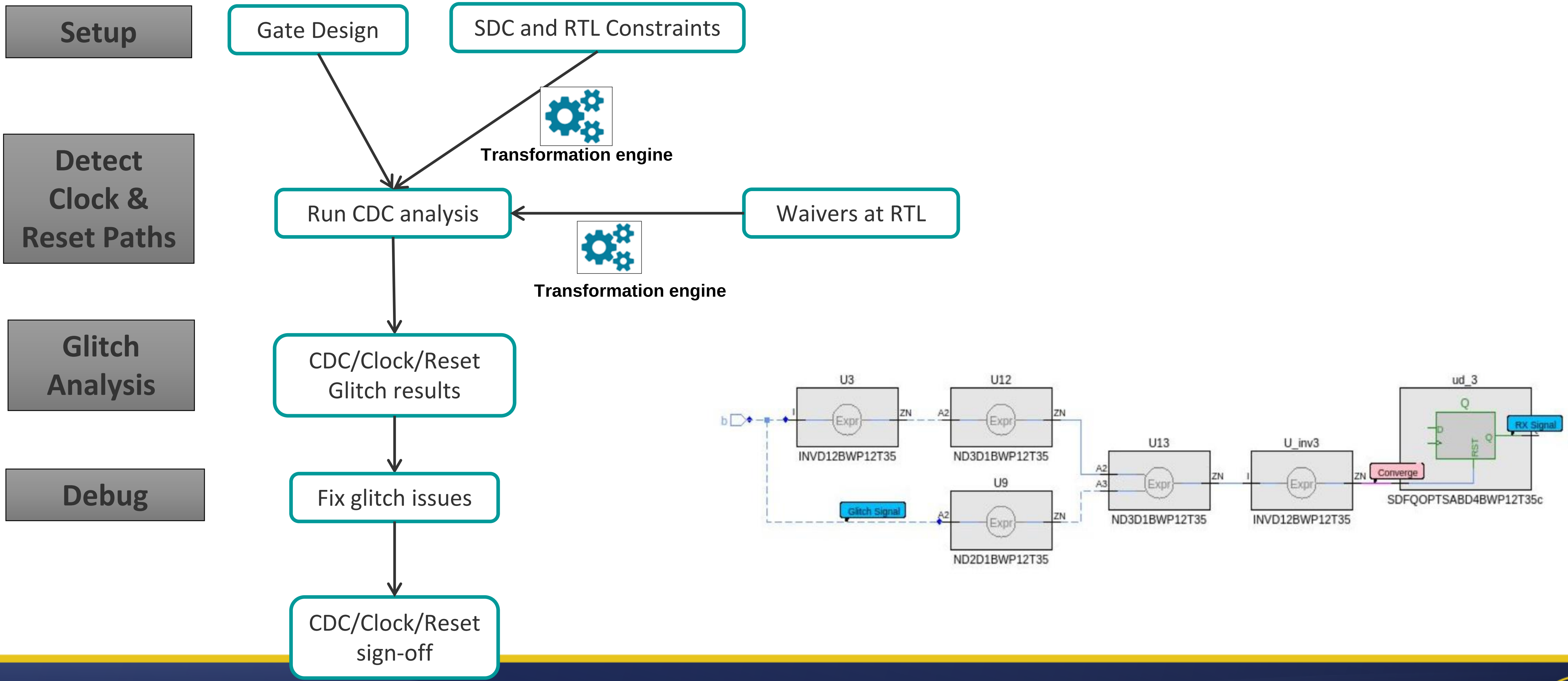
Figure 2.2 Schematic with probable glitch condition
Glitch Signal : B

- Problematic for both synchronous and asynchronous paths
- Glitches will cause unexpected resets or unexpected clock edges

Glitch Detection Challenges

- *Gate-level simulation*
  - Simulation is not exhaustive and time consuming
  - Glitch detection is difficult
- *Assertions to identify glitch*
  - Glitches may occur where assertions are not
- *Static checks at the netlist level*
  - Detection is noisy
  - Manual review is error-prone and time-consuming

Questa Gate-level CDC : Overview



Setup

Detect Clock & Reset Paths

Glitch Analysis

Debug

Gate Design

SDC and RTL Constraints

Transformation engine

Run CDC analysis

Waivers at RTL

Transformation engine

CDC/Clock/Reset Glitch results

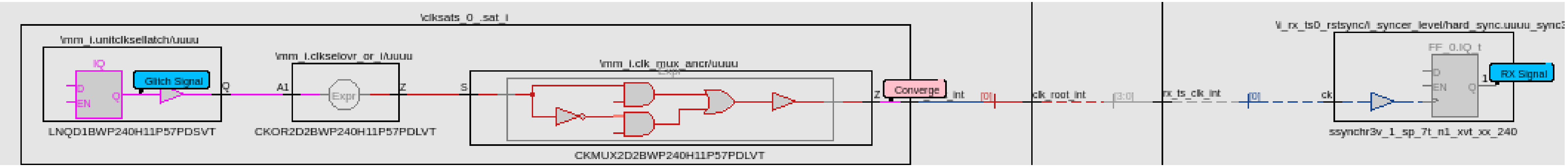
Fix glitch issues

CDC/Clock/Reset sign-off

Clock Glitch Results

- Design A (10M gates)
- Design B (2M gates)

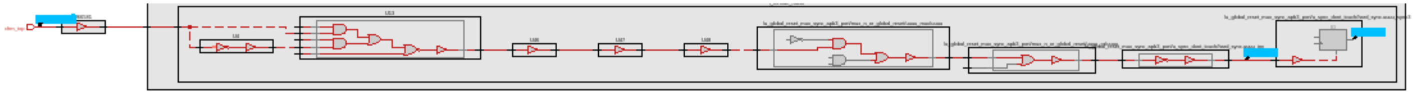
Design name	Number of clock domains	Number of glitch signals	Number of clock paths impacted
Design A	107	68	8437
Design B	22	3	1

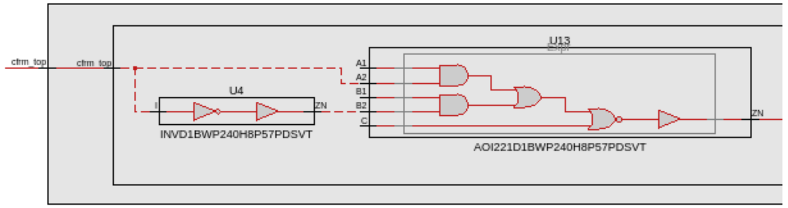


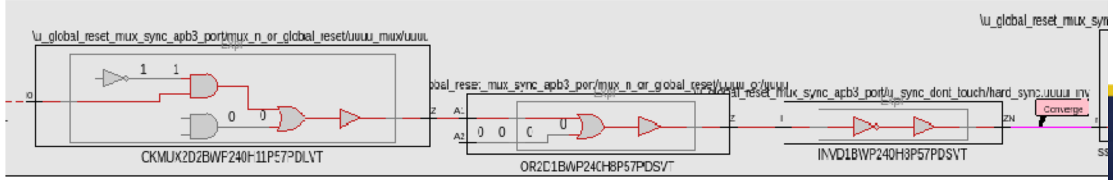
Reset Glitch Results

- Design A (10M gates)
- Design B (2M gates)

Design name	Number of reset domains	Number of glitch signals	Number of reset paths impacted
Design A	33	2	714
Design B	43	6	4

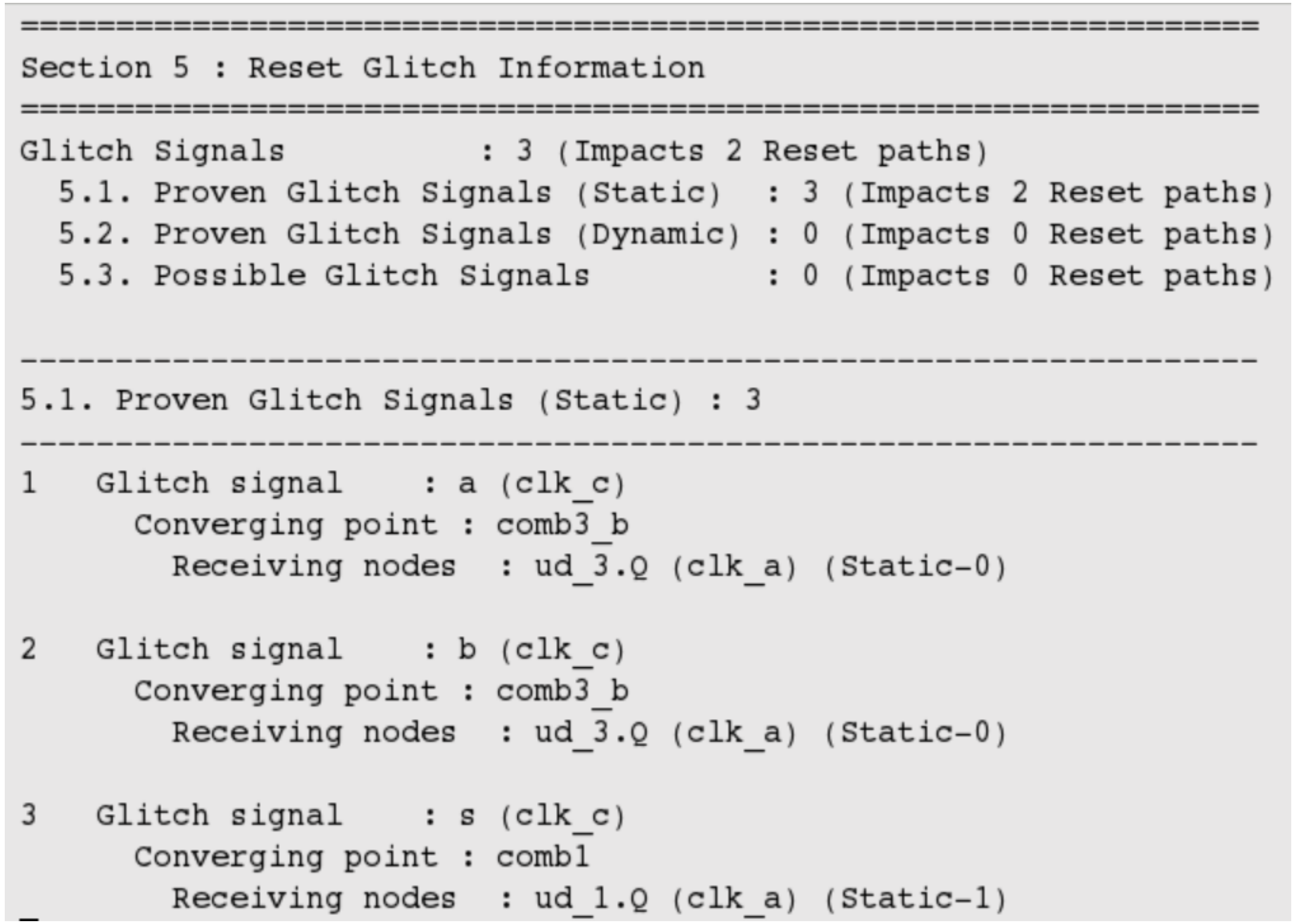





\u_global_reset_mux_sync_apb3_port/mux_n_or_global_reset/uuuu_mux/uuuu
\u_global_reset_mux_syn
1
1
0
0
CKMUX2D2BWF240H11P57PDLVT
0
0
OR2D1BWP240H8P57PDSVT
INVD1BWP240H8P57PDSVT
Converge
ZN

Improved Glitch Debug TAT

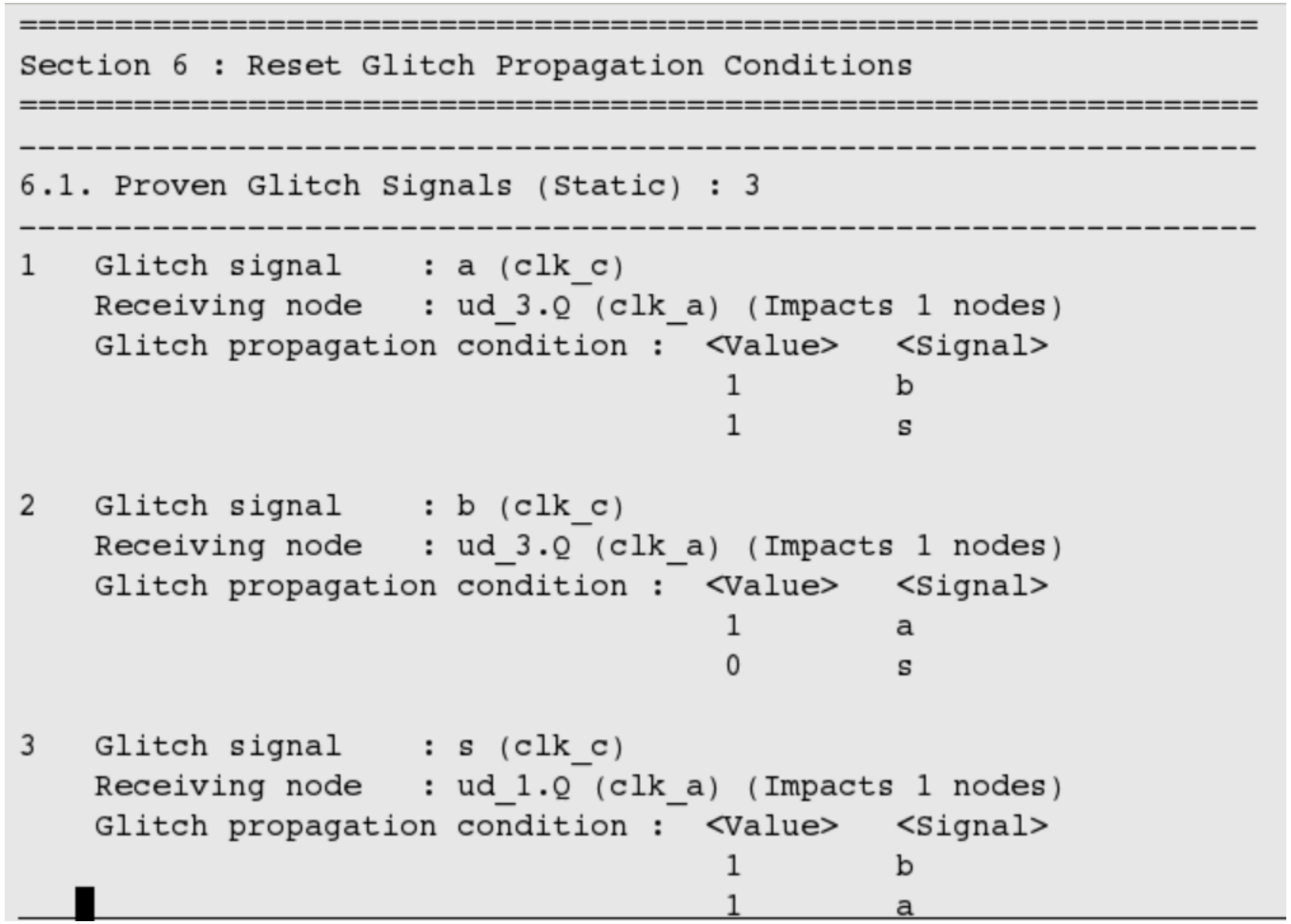
- Reports glitch source, converge point, destination
- Reports paths with possible glitch
  - Could not prove with formal



```
==========================================================================
Section 5 : Reset Glitch Information
==========================================================================
Glitch Signals               : 3 (Impacts 2 Reset paths)
  5.1. Proven Glitch Signals (Static)   : 3 (Impacts 2 Reset paths)
  5.2. Proven Glitch Signals (Dynamic)  : 0 (Impacts 0 Reset paths)
  5.3. Possible Glitch Signals          : 0 (Impacts 0 Reset paths)

--------------------------------------------------------------------------
5.1. Proven Glitch Signals (Static) : 3
--------------------------------------------------------------------------
1   Glitch signal     : a (clk_c)
      Converging point : comb3_b
        Receiving nodes   : ud_3.Q (clk_a) (Static-0)

2   Glitch signal     : b (clk_c)
      Converging point : comb3_b
        Receiving nodes   : ud_3.Q (clk_a) (Static-0)

3   Glitch signal     : s (clk_c)
      Converging point : comb1
        Receiving nodes   : ud_1.Q (clk_a) (Static-1)
```

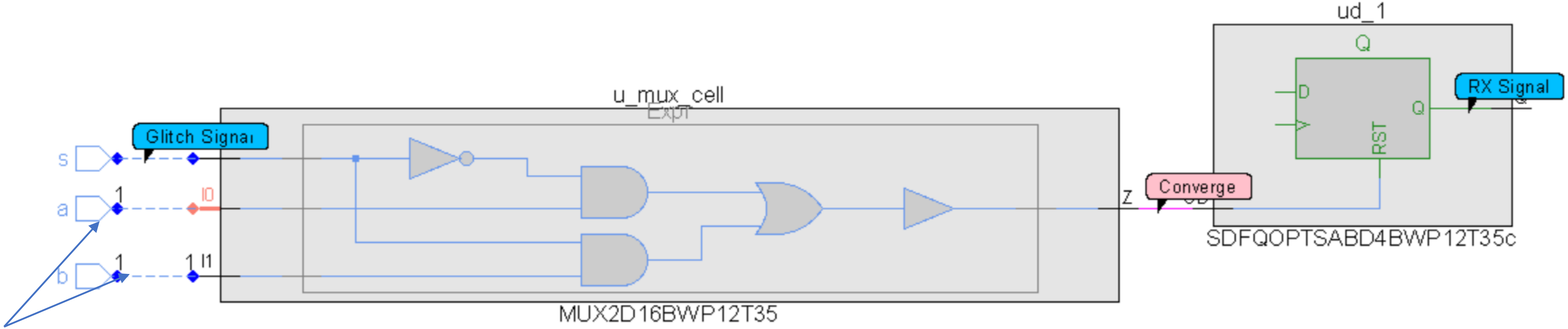
Improved Glitch Debug TAT

- Reports glitch source, converge point, destination
- Reports paths with possible glitch
  - Could not prove with formal
- Reports glitch propagation conditions
  - Proven with formal analysis



```
=============================================================================
Section 6 : Reset Glitch Propagation Conditions
=============================================================================
-----------------------------------------------------------------------------
6.1. Proven Glitch Signals (Static) : 3
-----------------------------------------------------------------------------
1    Glitch signal     : a (clk_c)
     Receiving node    : ud_3.Q (clk_a) (Impacts 1 nodes)
     Glitch propagation condition :  <Value>   <Signal>
                                       1          b
                                       1          s

2    Glitch signal     : b (clk_c)
     Receiving node    : ud_3.Q (clk_a) (Impacts 1 nodes)
     Glitch propagation condition :  <Value>   <Signal>
                                       1          a
                                       0          s

3    Glitch signal     : s (clk_c)
     Receiving node    : ud_1.Q (clk_a) (Impacts 1 nodes)
     Glitch propagation condition :  <Value>   <Signal>
                                       1          b
                                       1          a
```

Improved Glitch Debug TAT



Glitch Propagation Conditions

Summary

- Glitch methodology identifies clock and reset glitches
- Formal-based glitch analysis improves quality of results (QoR)
    - Reduces violation noise
    - Provides debug triage

2021
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
EUROPE
OCTOBER 26-27, 2021

Thank you!

accellera
SYSTEMS INITIATIVE