



**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA





JUNE 23-27, 2024

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA



MARVELL™



REAL INTENT

# FORMAL CDC GLITCH CHECK ADVANCED SIGNOFF SOLUTION

Noam Eshel-Goldman

Liad Nehama

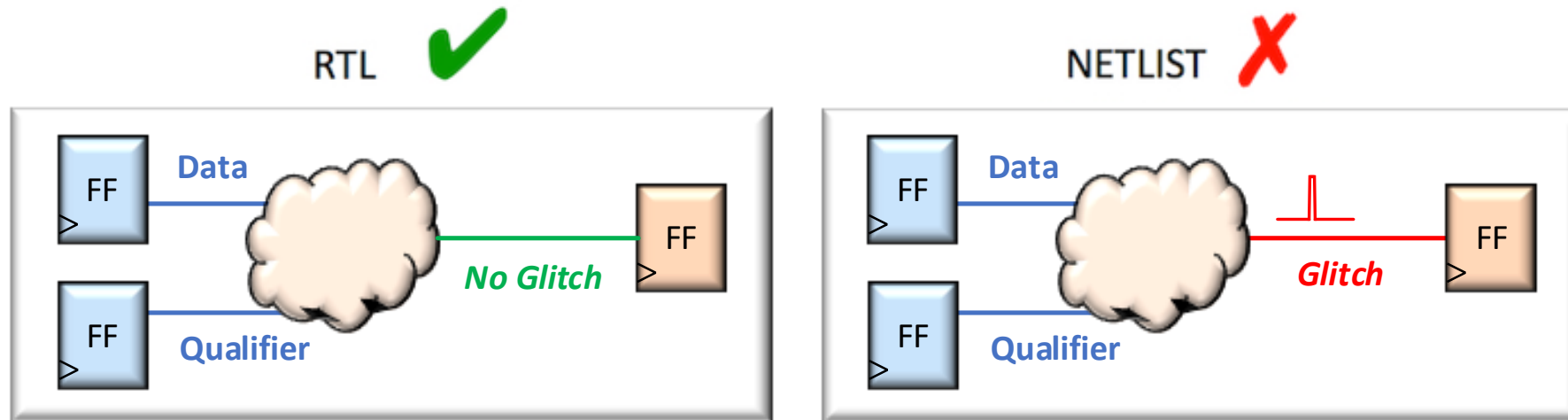
Polina Pashayev

Vikas Sachdeva



# Motivation

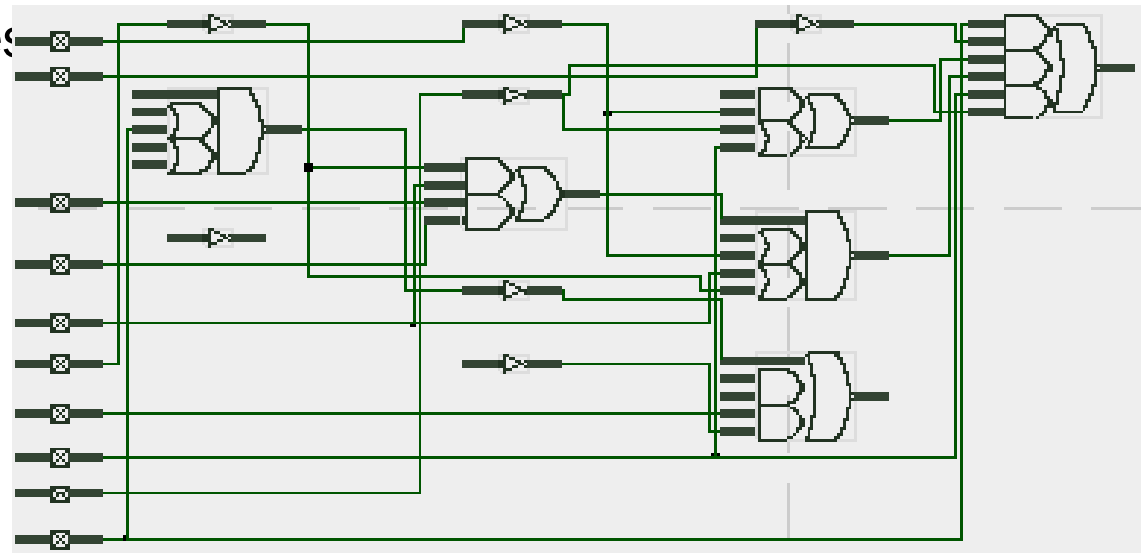
- Contemporary SoCs feature numerous clocks and CDC paths  
One aspect of CDC is glitch propagation
- Glitches cause metastability, leading to unpredictable behavior & data corruption
- Glitch elimination is highly important to ensure robust systems
- A combination of process technology trends and increased intervention by synthesis tools in logic generation, can lead to cases in which a design that is CDC-clean at RTL fails in post synthesis GL netlist



# CDC Verification – Standard Approach

- CDC structural checks by EDA tools
  - Usually runs on RTL and proves the RTL correctness
  - Violation reports require reviews to filter false positive reports
  - Can run on GL but results in high number of false positive reports – difficult to filter
- Gate-Level SDF Simulations
  - Requires high effort to ramp-up
  - Only checks the simulated cases - does not cover all possible cases
  - Bad ROI – Sometimes abandoned
- Design reviews
  - Reviews focus on RTL
  - Can miss the fine details

Would you want to review this??

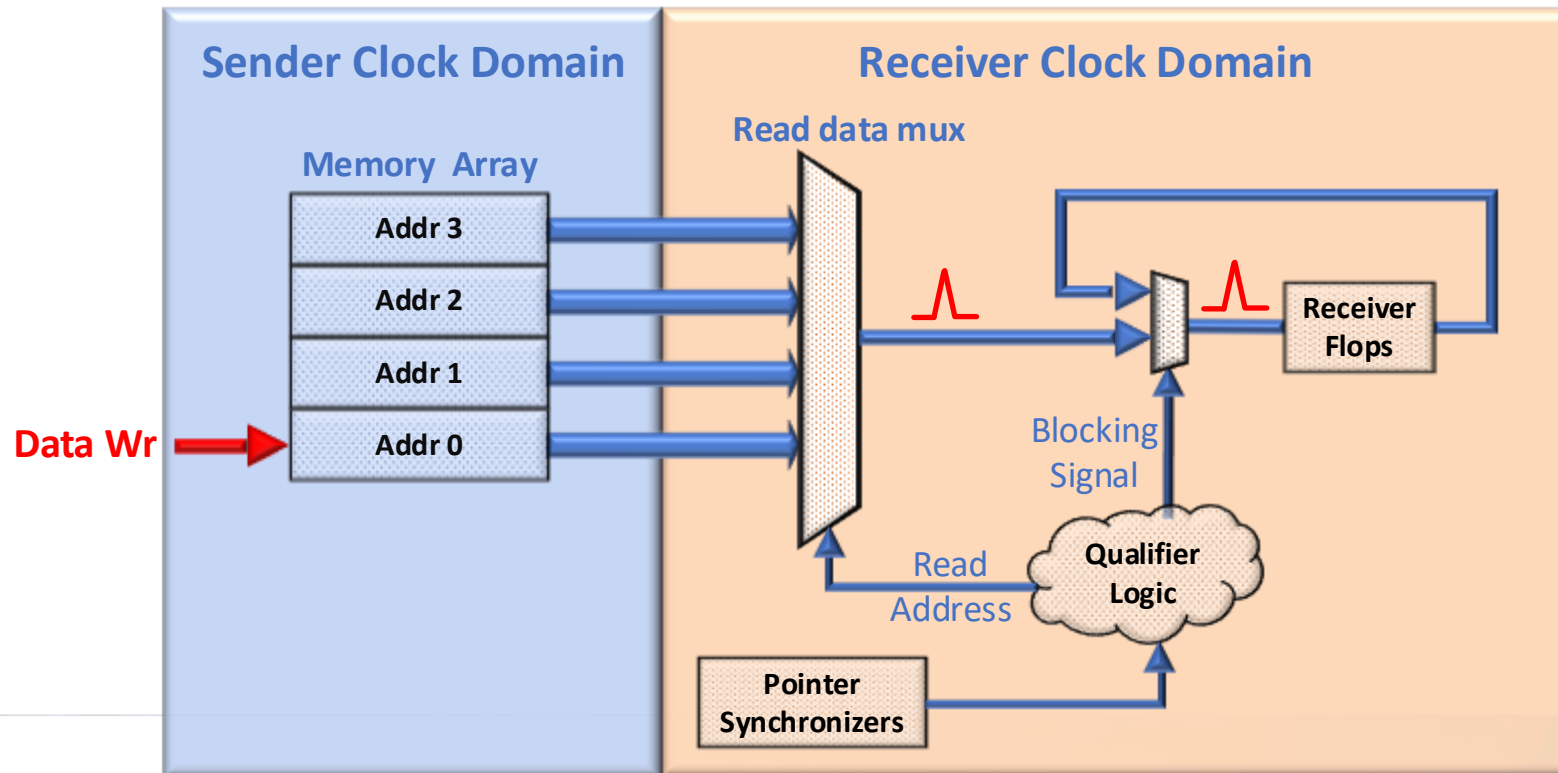


# CDC Verification – Standard Approach

- STA reports
  - Can report CDC violations of GL netlist
  - Often very pessimistic – challenging to filter out all the false positives
- Bottom line:  
Standard CDC verification approaches are focused on RTL.  
Verification of GL netlist is more challenging and often overlooked

# Examined CDC Glitch Case

- A typical asynchronous FIFO
- RTL pass verification by all tools including CDC
- Synthesis can generate GL netlist that does not block CDC glitches
  - Asynchronous glitches from sender clock domain could pass to receiver clock domain

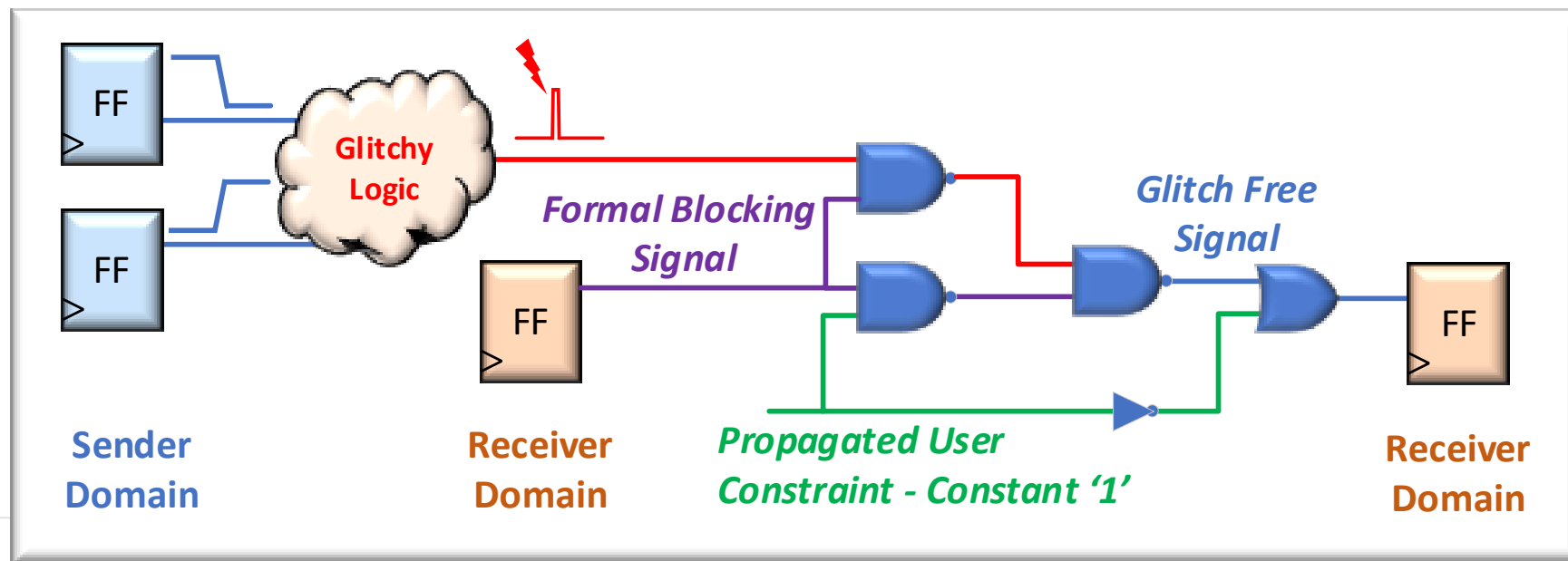


# Glitch Analysis Formal Solution

- Formal glitch analysis checks the GL and provides exact results in minimal run time
- Ability to check a path of interest using user defined constraints considerations

## Formal glitch check example using Meridian CDC:

- The **propagated user constraint** is not blocking the glitch source path
- Formal check considers all values of signals like the **purple signal** for proving if the glitch is blocked
- Value of '0' blocks the CDC path in the first NAND. Value of '1' blocks the CDC path one gate later



# Conclusion & Key Takeaways

- Suggestion for enhanced CDC Verification Methodology:
  - Identify CDC structures (FIFO's / bus qualifiers Etc.)
  - Add Formal CDC checks to prove correct GL functionality
  - SDF simulations can be reduced or eliminated if formal checks are clean
- Design tips for avoiding CDC glitches:
  - Use manually instantiated library gates in RTL for glitch blocking
  - Ensure these gates are maintained in the implementation flow
  - Use proven synchronization building blocks that follow the guidelines above  
And pass all CDC qualification (on RTL & GL)



**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA

