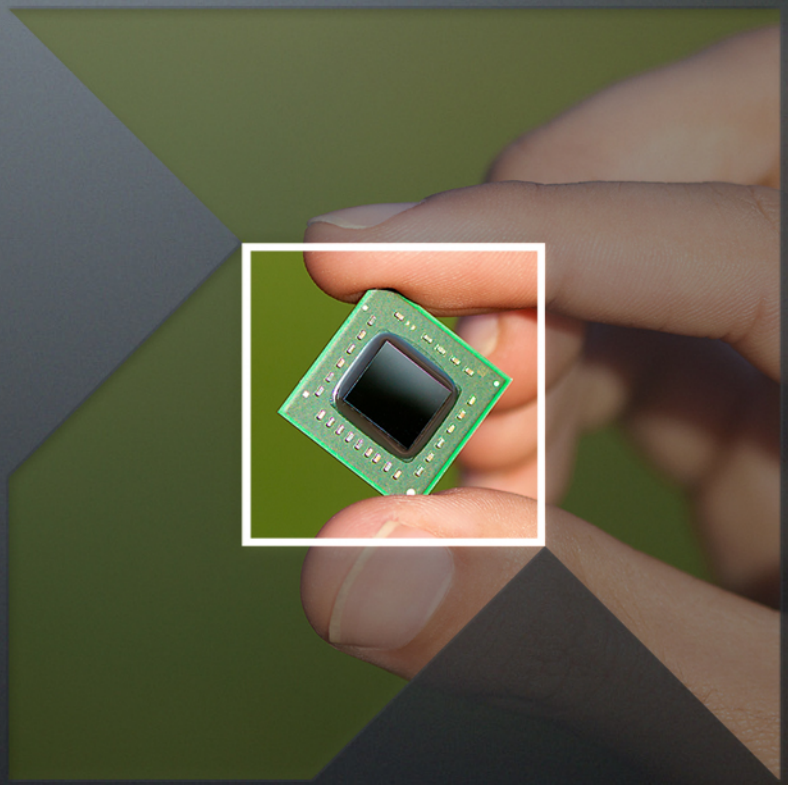


Exhaustive Latch Flow - Through Verification with Formal Methods

Baosheng Wang
Sean Ater
Baris Piyade
Brian McMinn
Borhan Roohipour

Antonio Celso Caldeira, Jr.
Bill Au
Rajeev Ranjan

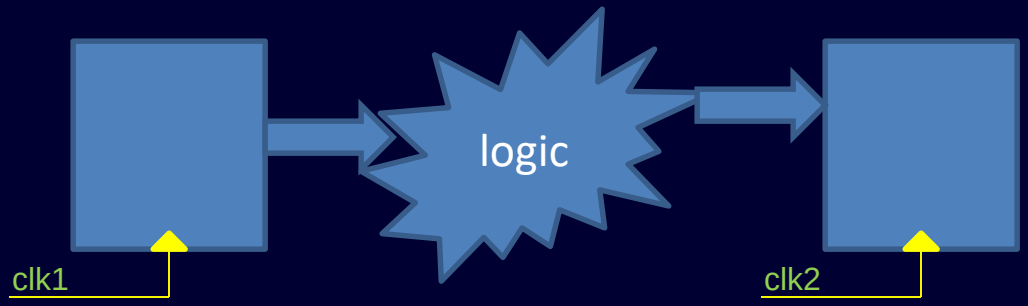
JASPER design automation

AMD

Talk Outline

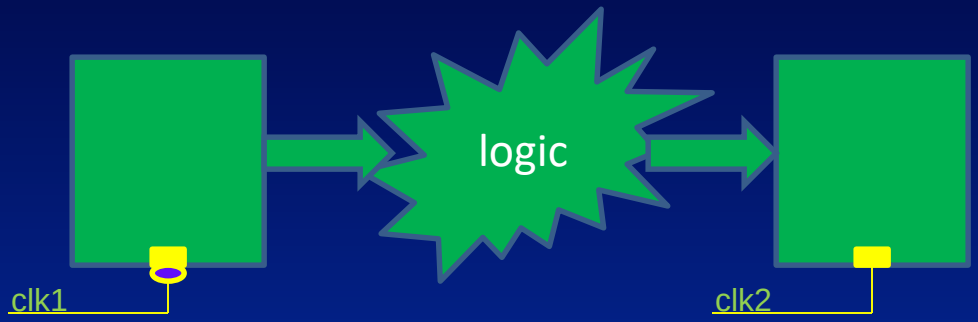
- Introduction
- Verification Challenges
- The Proposal
- Questions

Introduction



flop-based design

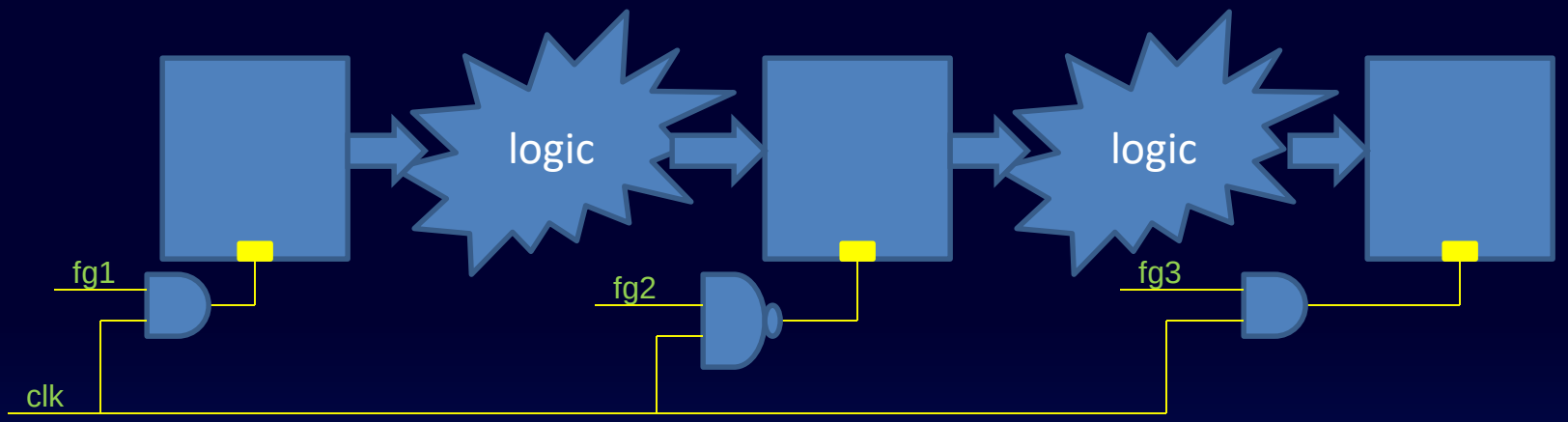
- Easy to implement
- Robust and Stable
- Wide support with EDA industry



latch-based design

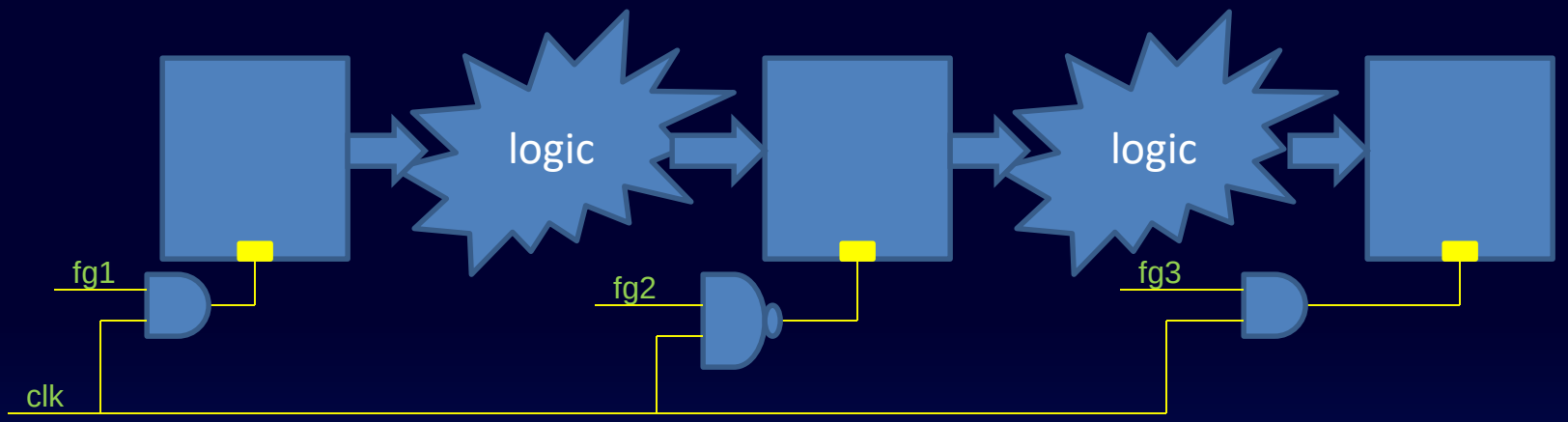
- Smaller
- Less power
- Can implement time-borrowing technique

Latch Flow-through (ABA) Problem



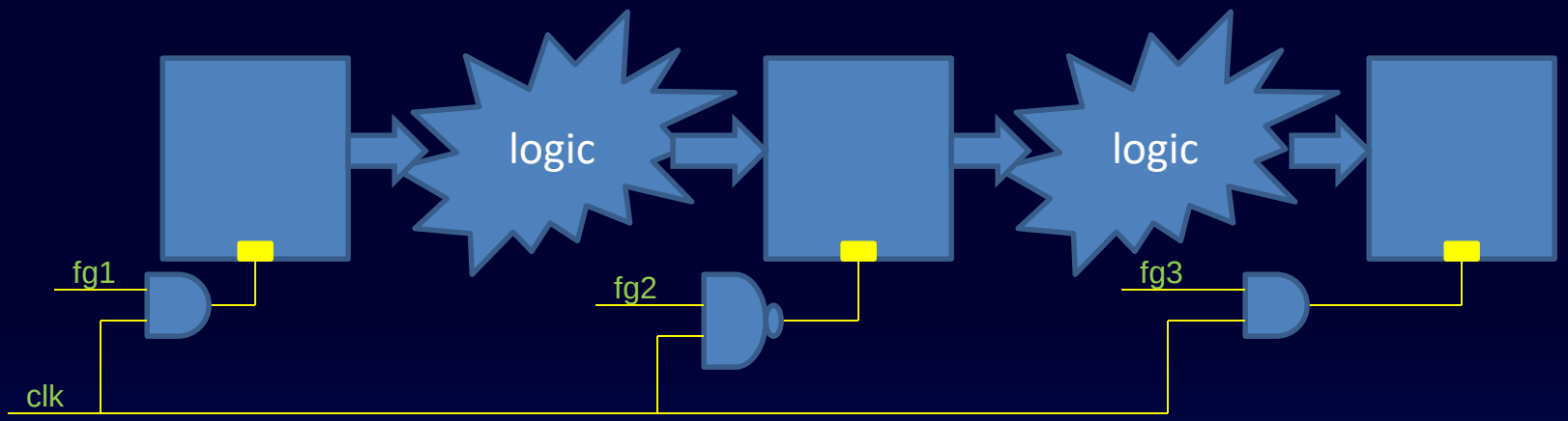
- Latch flows through!
  - When {fg1, fg2, fg3} = {1, 0, 1}
- Causes
  - Excessive clock gating for power
  - Random scan vectors
- Impacts
  - Functional failure due to data racing
  - Yield loss due to overkill

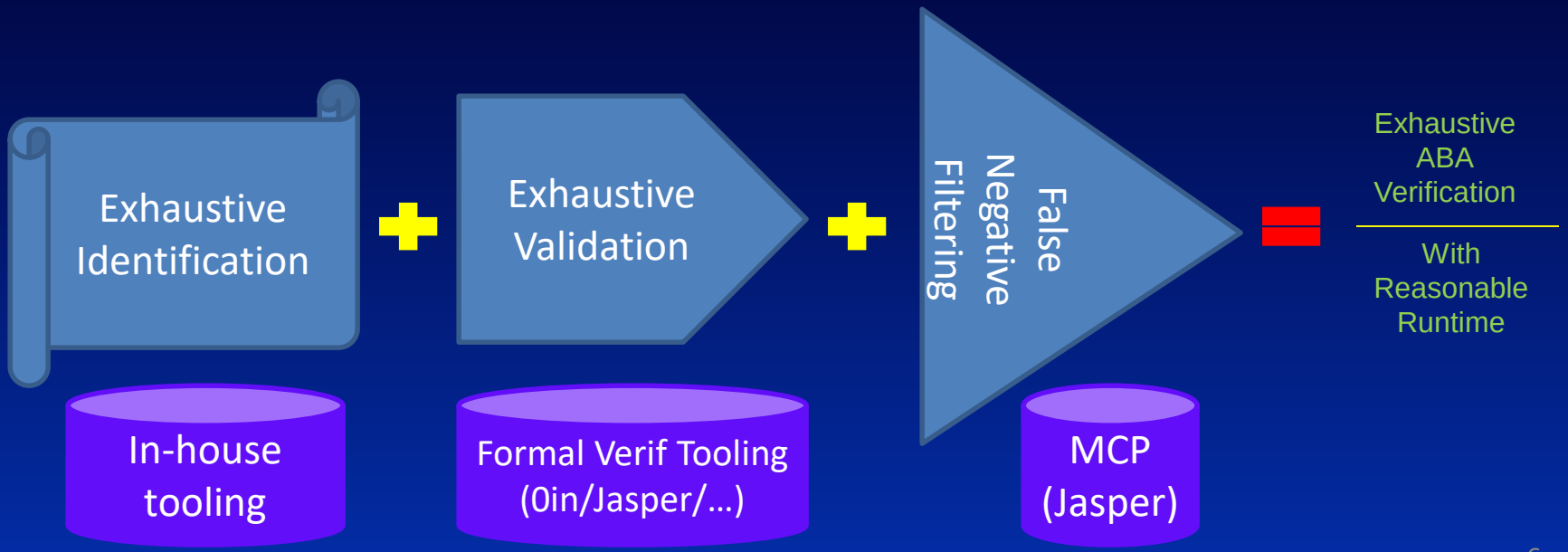
Verifying ABA Problem



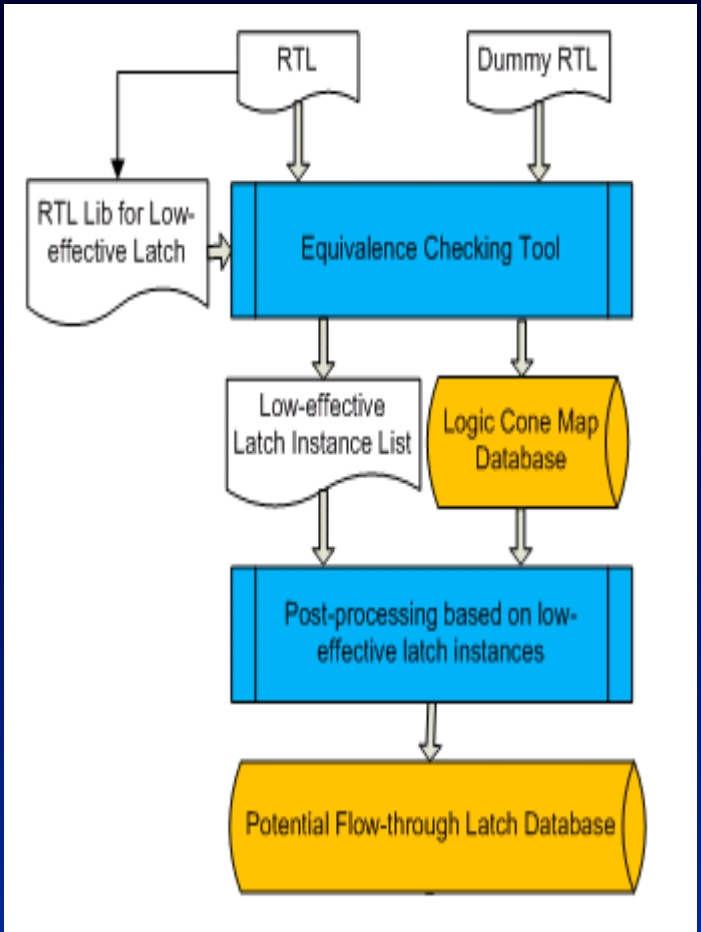
- Challenges
  - Identification
  - Tool capacity issues
    - Large number of instances
    - Large designs
  - Data Blocking filtering (see next several slides)
- Existing solution(s)
  - ATPG vectors + Special ATPG models
    - Not exhaustive

The Exhaustive Verification Solution





ABA Instance Identification



- Starting from RTL
- Using Conformal as synthesis engine
- Post-processing
- Creating assertions

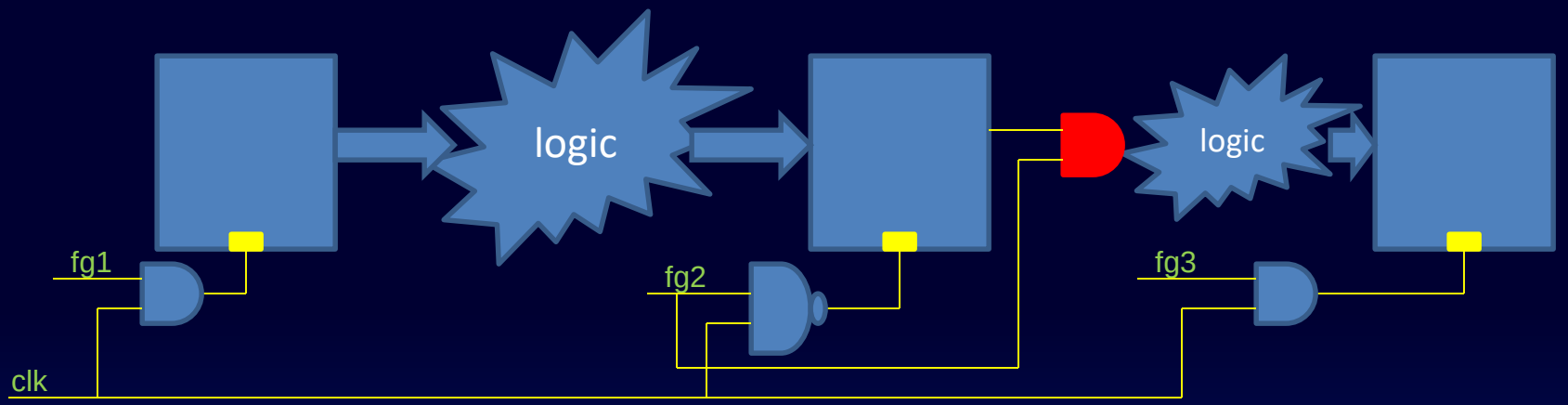
// Source latch gater = /foo/foo2/l_foo3
// Middle latch gater = /boo/boo2/l_boo3
// Destination latch gater = /poo/poo2/l_poo3
… …

assert_never … … .clk (~CLK), … … .test_expr
((*/foo/foo2/l_foo3) &*
~(/boo/boo2/l_boo3) &
(/poo/poo2/l_poo3));

Resolve formal tool capacity Issues

- Tool Capacity Challenges
  - Large number of assertions: 20k (base) x 50 (variance) = 1M
  - Large designs: up to 200k latches
- Resolution with Formal Tools
  - Running on BASE assertions only
  - Group assertions with parallel run
  - Black-box non-target scan-able flops
    - Mimic scan randomness, i.e., no reset
    - Need necessary data propagation on target states
      - Up to 99% of states black boxed after post processing

Data Blocking ABA Case



- Latches transparent but not flow-through
  - Blocking mechanism could be “AND/OR/MUX” gates or assertions
- False negatives with current assertion methods
- Hard-to-write assertions
- Manual review is counter-productive

Data Blocking Filter with MCP Technology

MCP Technology in Jasper

Use an assertion to check that the path requires at least N cycles for value propagation

```
assert    -name … …
          -mcp
          -from
          /foo/foo2/l_src1
          -through
          /boo/boo2/l_mid1
          -to
          /poo/poo2/l_dest1
          -cycle 2
```

Data Blocking Filter

formal tool tries to find a counter example that the destination latch can not be propagated with deterministic value within 2 evaluation cycles

If proved:

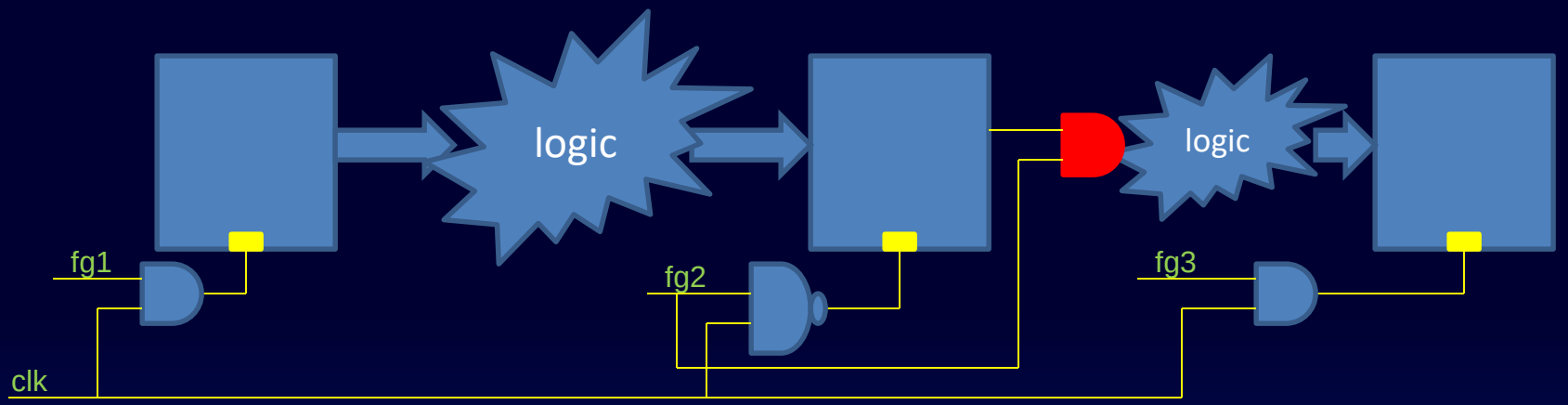
data blocking cases

If fired:

real ABA bug

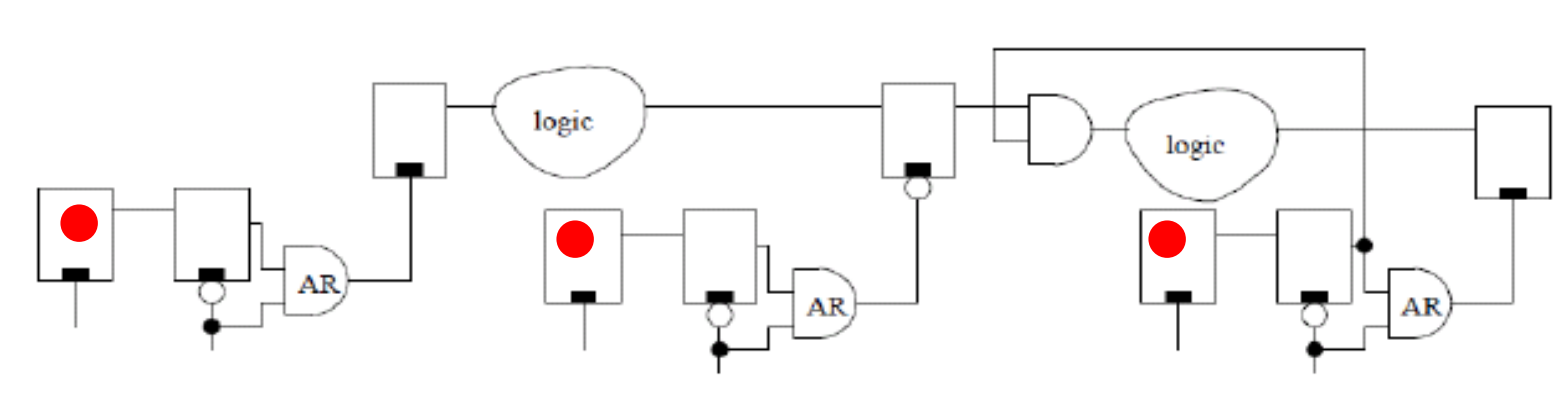
Since there are no clock and/or data correlations within this path

Data Blocking Filter with MCP Technology



- Tool capacity issues
  - Large number of MCP instances
  - Lots of bounded proof
- Resolution
  - Re-grouping bit-blaster instance into bus scenario
  - Start with uninitialized states but stop at depth =2

Experimental Results – Proof of Concept



An ABA Violation Instance

BLMs	# (Latch Sequences)	# (assertions)	# (data blocking cases)	# (real latch flow-through scenarios)
BLM1	588	11	78	0
BLM2	10,730	638	530	16

ABA Verification Results on A Sample of Block-Level-Modules (BLMs)

Experimental Results – Regression Performance

Component	C0	C1	C2
Computing Memory	10G	26G	26G
# Latch Sequences	456k	2.04m	1.893m
# assertions	201	1324	18014
# scannable flops	15k	107k	89k
% scannable flops blackboxed	98.9	95.8	92.1
Assertion generation time (hours)	0.7	3.0	5.7
Assertion+MCP prove time (hours)	3.0	3.3	3.7
Total run time (hours)	3.7	6.3	9.4

Program Deployment Success

- Regression flow in place for multiple programs
  - Running twice a week
  - Gating RTL check in
- Catch many bugs in late phase of the design
  - Screen over 30 pre-silicon bugs
  - Identify one post silicon bug

Summary

- Latch flow-through issue is another concern for latch-based design
  - Unnecessary functional vectors
  - Yield loss due to “overkill”
- AMD provides an exhaustive verification solution consisting of multiple formal techniques
  - In-house ABA identification scripts
  - Formal Tools
  - MCP technology from Jasper Design Automation
  - Push-button-based regression flow in place, ensuring robust AMD designs!

Questions?