REAL INTENT

Accelerating RTL Sign-off

A Paradigm Shift in Verification Methodology

Pranav Ashar

Real Intent, Inc.

FMCAD, October 2016

The New Paradigm

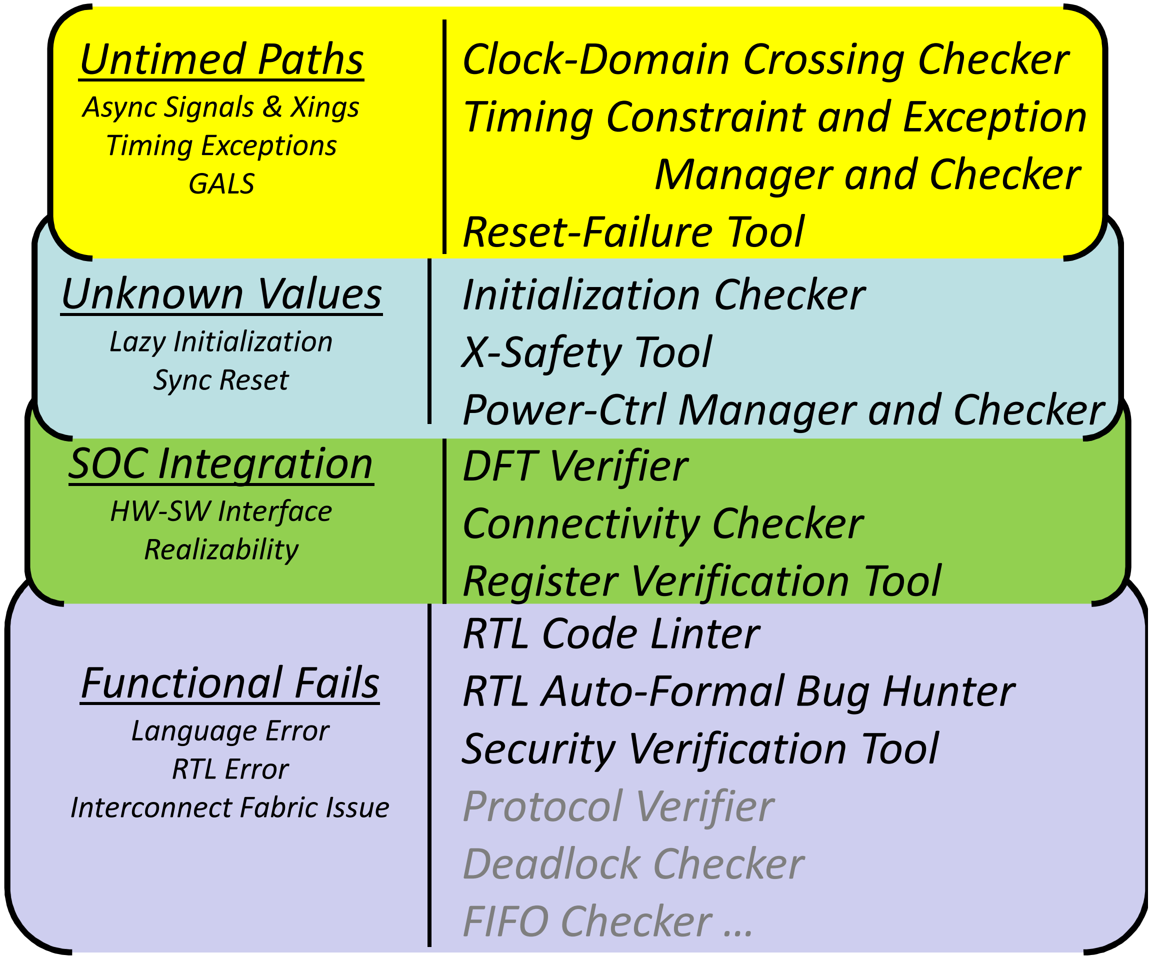
REAL INTENT
Accelerating RTL Sign-off

Generic Tools

RTL & Netlist Simulators
Formal Equivalence Checker
Assertion-based Formal Tool
Static Timing Analyzer

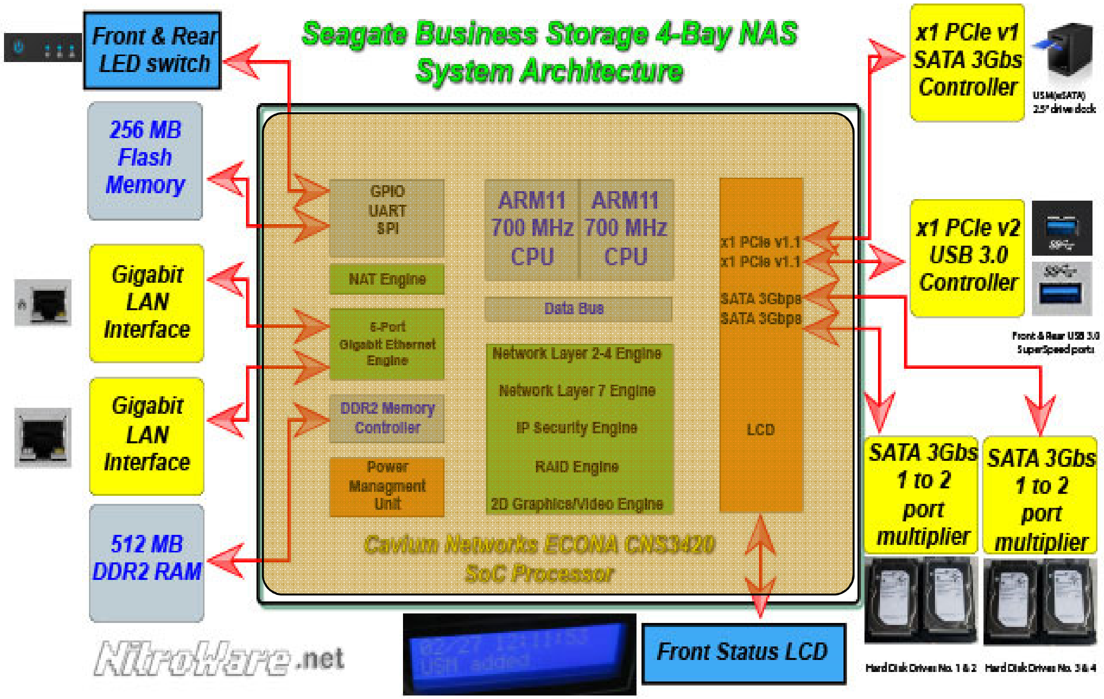
→

Targeted Solutions



Untimed Paths *Async Signals & Xings* *Timing Exceptions* *GALS*	*Clock-Domain Crossing Checker* *Timing Constraint and Exception Manager and Checker* *Reset-Failure Tool*
Unknown Values *Lazy Initialization* *Sync Reset*	*Initialization Checker* *X-Safety Tool* *Power-Ctrl Manager and Checker*
SOC Integration *HW-SW Interface* *Realizability*	*DFT Verifier* *Connectivity Checker* *Register Verification Tool*
Functional Fails *Language Error* *RTL Error* *Interconnect Fabric Issue*	*RTL Code Linter* *RTL Auto-Formal Bug Hunter* *Security Verification Tool* *Protocol Verifier* *Deadlock Checker* *FIFO Checker …*

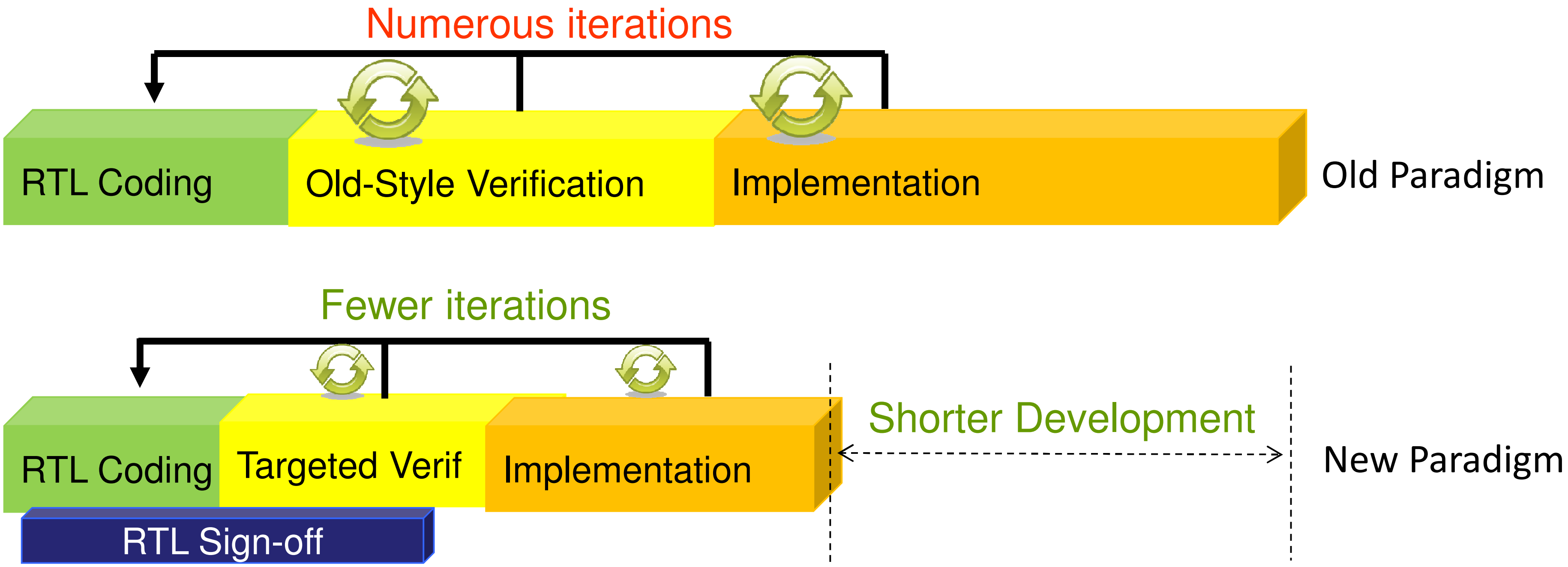
New Failure Modes are Very Real



Cavium CNS3420 SoC Processor , designed specifically for Networking Devices with full offloading and hardware support for network stack, IP/SSL Security, RAID and NAT.

- **High performance**
  - Includes high speed ARM11 cores & caches and over 10 application acceleration engines
  - Needs thorough analysis to ensure correct functionality
- **Complex**
  - Complex clock domain interactions
  - Many reset domains
  - Several asynchronous interfaces: Processor, caches, application engines, low-latency integrated memory, system and networking interfaces
- **Large: >250M gates**
  - Needs massive capacity for the design analysis
- **High risk for silicon failure**
  - Insidious bugs found late in the process

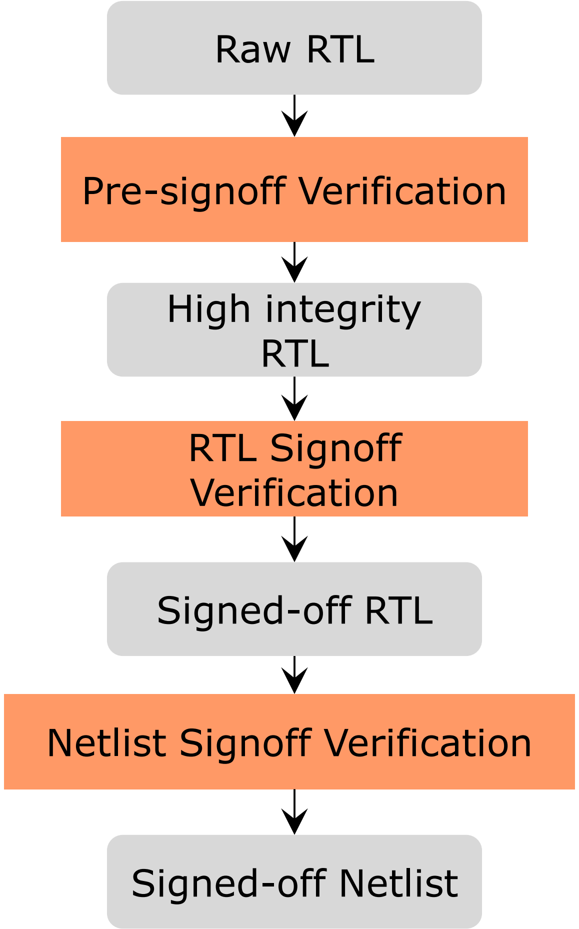
The Left Shift



- **Start verification earlier**
- **Compress the development cycle**
- **Sign-off level confidence**
- **Lower Cost**

Cost of a bug increases exponentially with each stage of the design process

A Manifestation of the New Paradigm



Raw RTL

Pre-signoff Verification

High integrity RTL

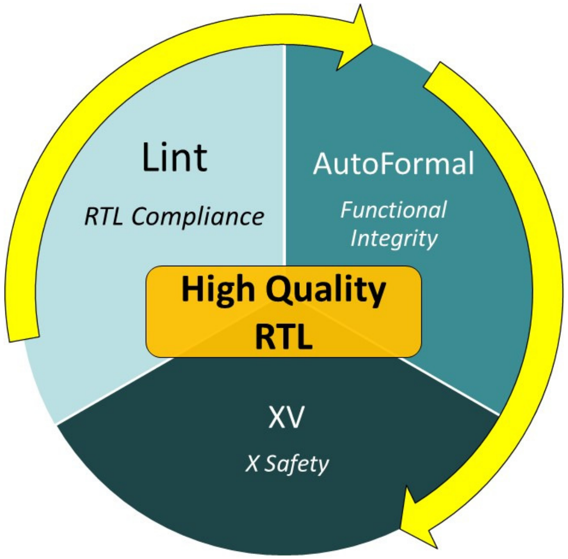
RTL Signoff Verification

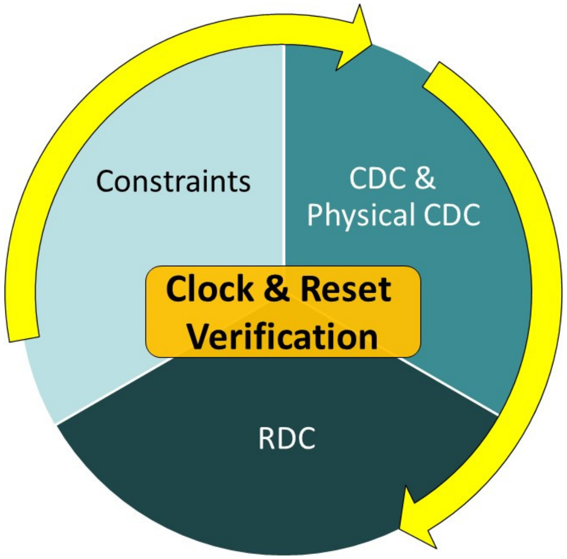
Signed-off RTL

Netlist Signoff Verification

Signed-off Netlist

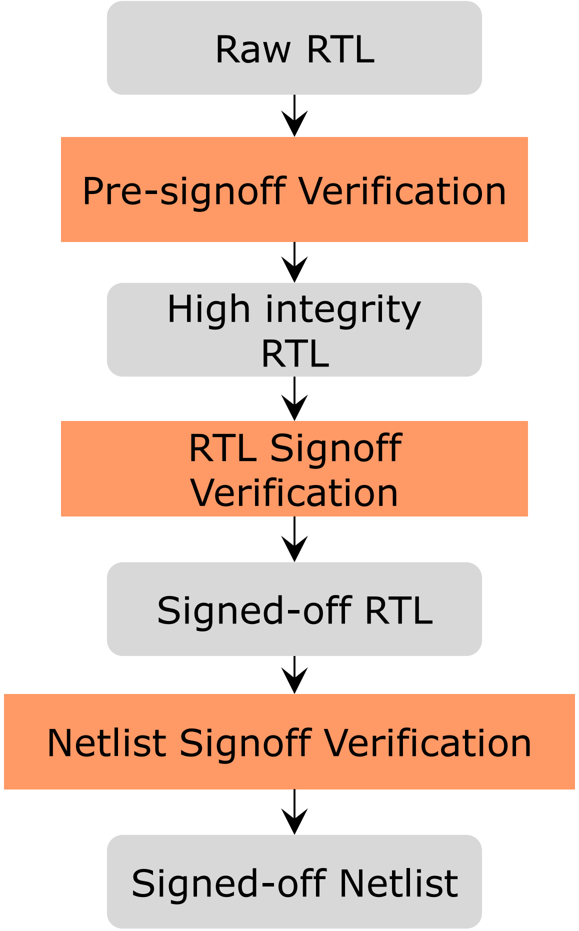
- **High-Value** verification targets
  - CDC, Reset, Constraints, Exceptions, X-safety etc
  - Beyond & complement existing flows (Simulation + STA)
- **Systematic** convergence
  - Setup + Semantic Analysis + Formal Analysis
  - Execute -> Review -> Iterate
- **Use Model**
  - Accuracy, Capacity, Debug, Data Mgmt





A Manifestation of the New Paradigm

REAL INTENT
Accelerating RTL Sign-off



Raw RTL

Pre-signoff Verification

High integrity RTL

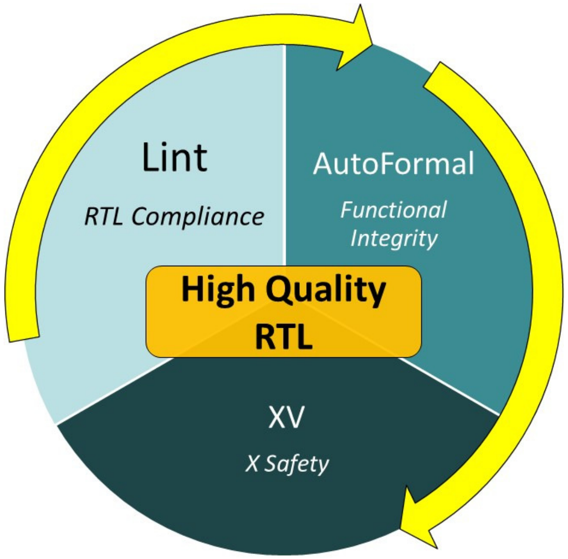
RTL Signoff Verification

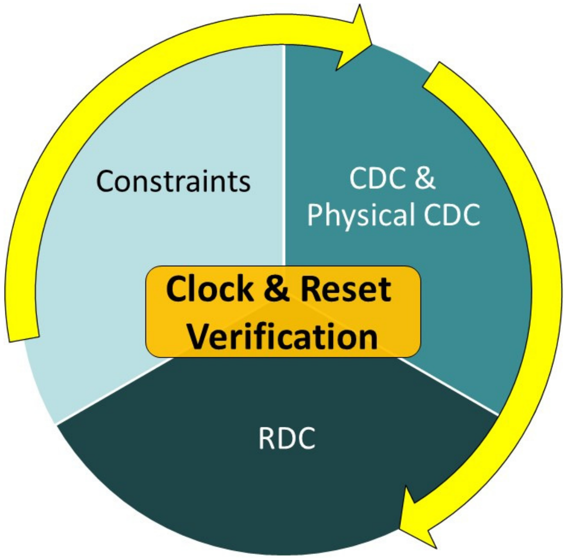
Signed-off RTL

Netlist Signoff Verification

Signed-off Netlist

- **<u>High-Value</u>** verification targets
  - CDC, Reset, Constraints, Exceptions, X-safety etc
  - Beyond & complement existing flows (Simulation + STA)
- **<u>Systematic</u>** convergence
  - Framing + Scoping + Sign-off
  - Execute -> Review -> Iterate
- **<u>Use Model</u>**
  - Accuracy, Capacity, Debug, Data Mgmt





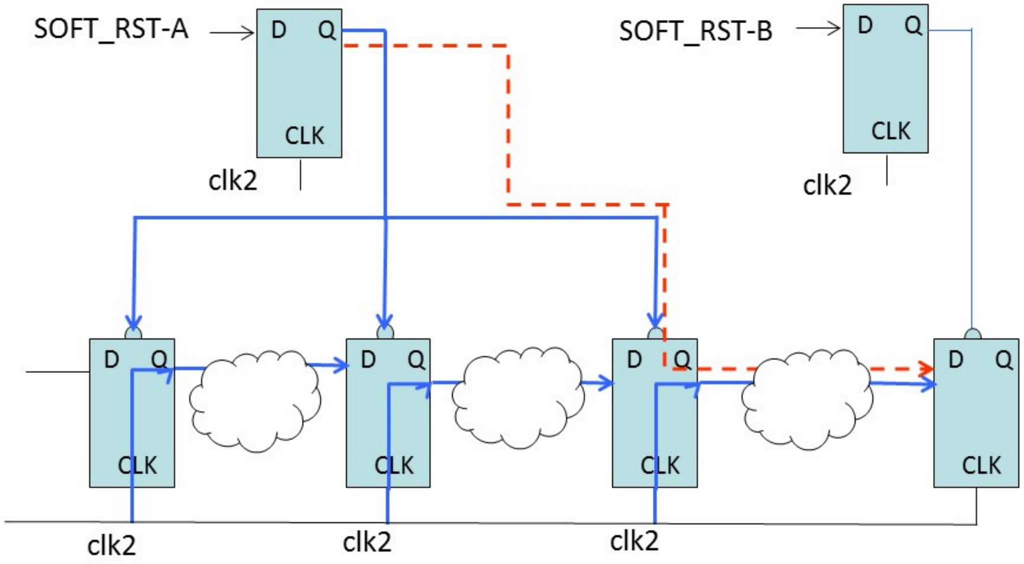
Hidden Cost Without the New Tools: Over-design

REAL INTENT
Accelerating RTL Sign-off

- Many examples:
  - Extra latency on async crossings
  - Paths that could be exceptions are timed in STA
  - Explicitly reset every FF
  - Synchronous reset where Async reset could've worked

Hidden Cost Without the New Tools: Over-design

- Many examples:
  - Extra latency on async crossings
  - Paths that could be exceptions are timed in STA
  - Explicitly reset every FF
  - Synchronous reset where Async reset could've worked

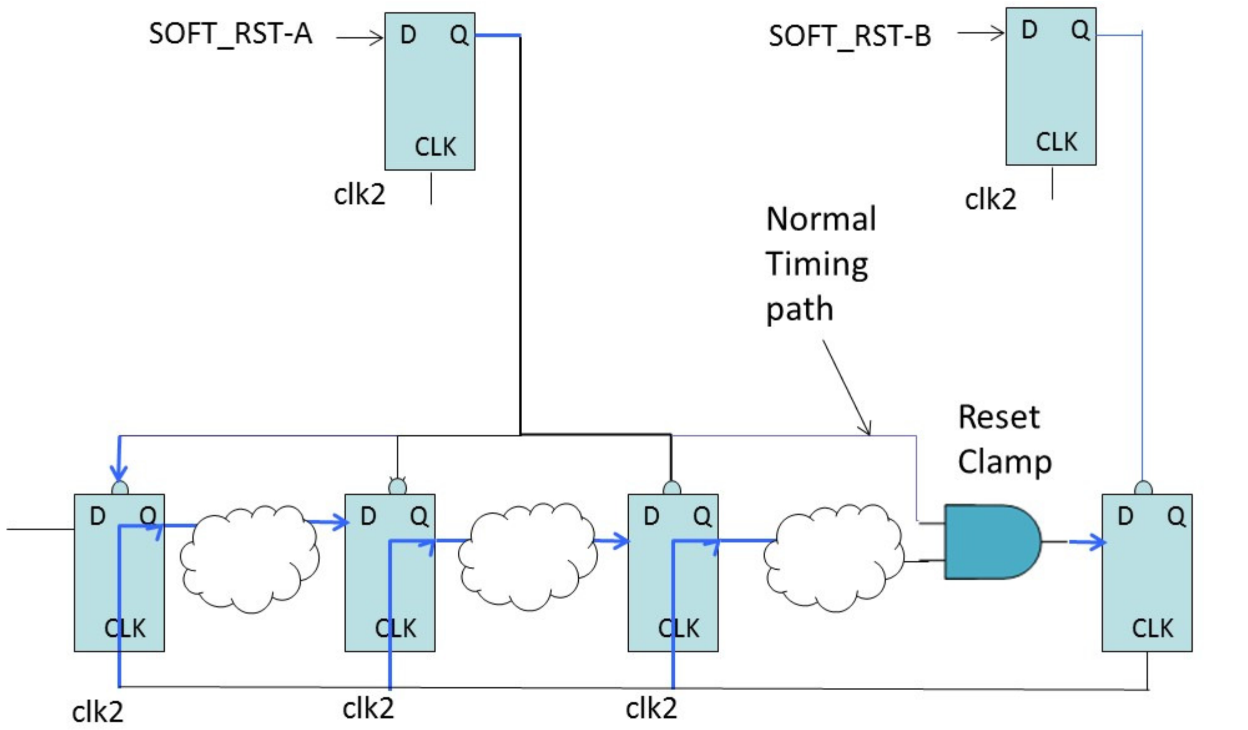


STA constrains the normal timing paths →

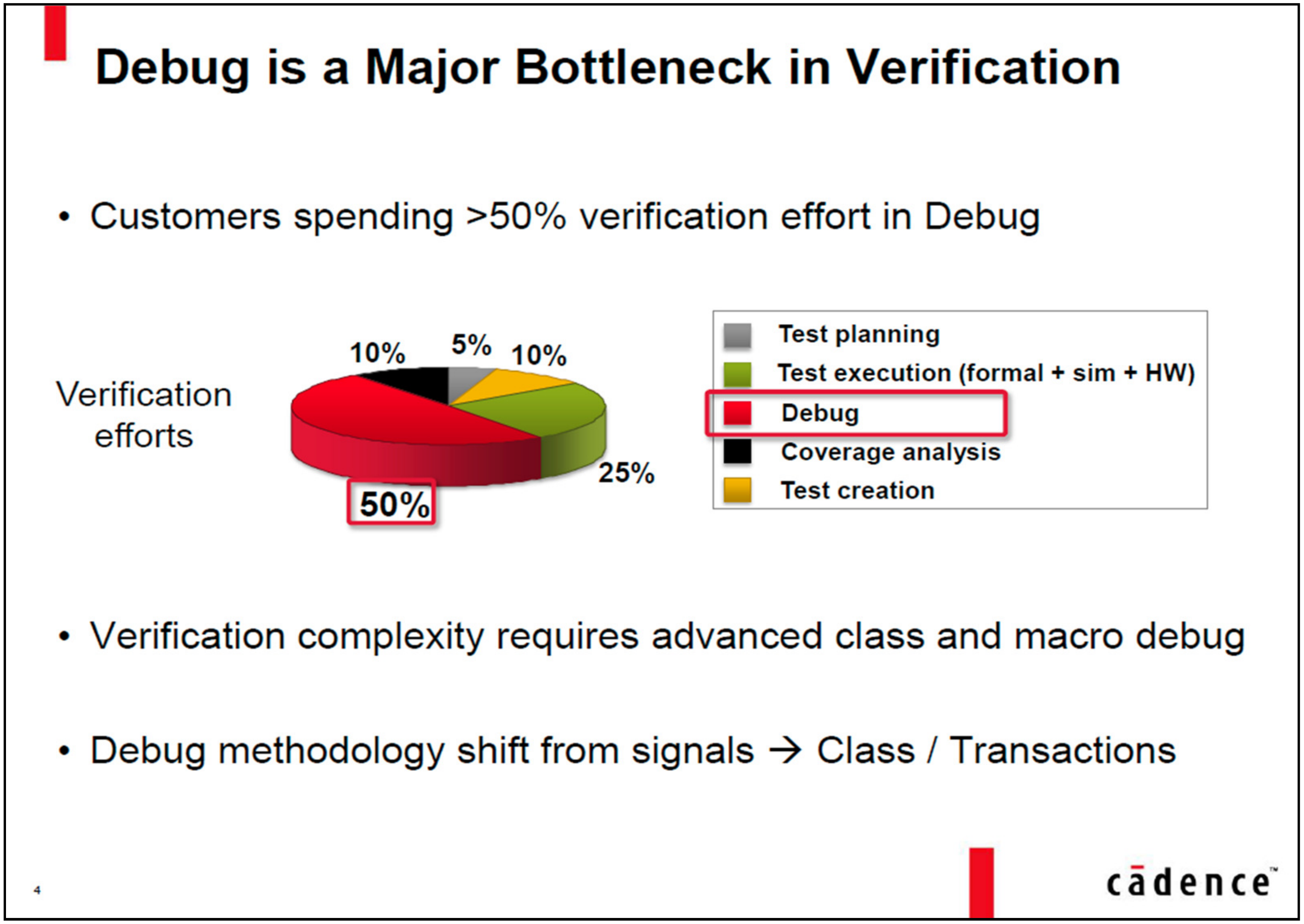
Problem: Assertion of SOFT_RST_A creates an untimed path - - - →

Hidden Cost Without the New Tools: Over-design

- Many examples:
  - Extra latency on async crossings
  - Paths that could be exceptions are unnecessarily timed in STA
  - Explicitly reset every FF
  - Synchronous reset where Async reset could've worked



Debug in the New Paradigm



Debug is a Major Bottleneck in Verification

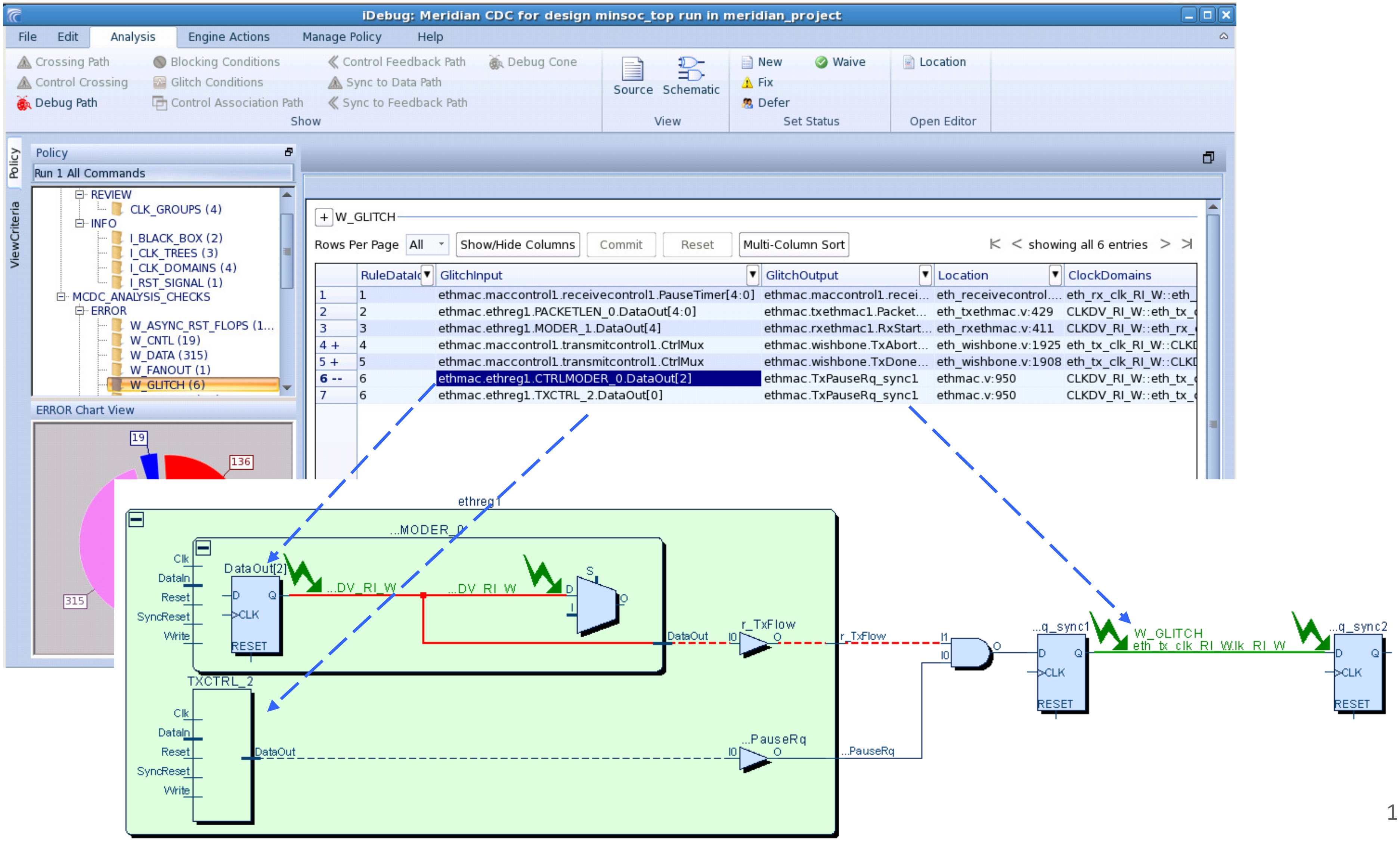
- Customers spending >50% verification effort in Debug

- Verification complexity requires advanced class and macro debug
- Debug methodology shift from signals → Class / Transactions

Nick Heaton, Senior Solution Architect, Verification Futures

Tool Guides Debug Example: CDC-Glitch

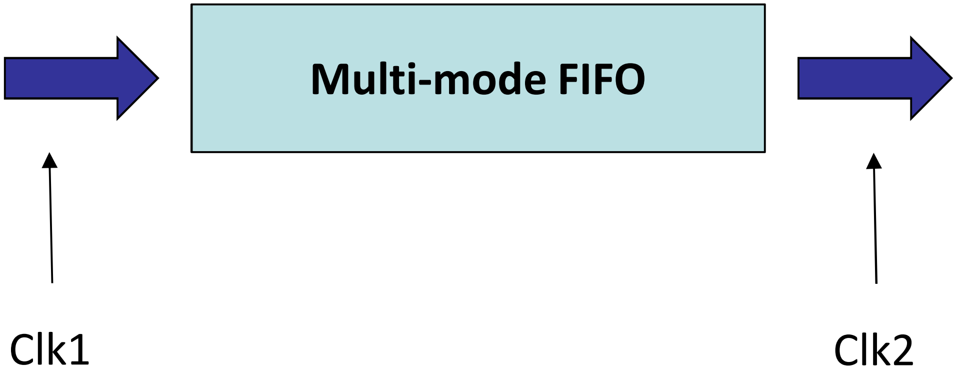
REAL INTENT
Accelerating RTL Sign-off



	RuleDataId	GlitchInput	GlitchOutput	Location	ClockDomains
1	1	ethmac.maccontrol1.receivecontrol1.PauseTimer[4:0]	ethmac.maccontrol1.recei...	eth_receivecontrol....	eth_rx_clk_RI_W::eth_
2	2	ethmac.ethreg1.PACKETLEN_0.DataOut[4:0]	ethmac.txethmac1.Packet...	eth_txethmac.v:429	CLKDV_RI_W::eth_tx_
3	3	ethmac.ethreg1.MODER_1.DataOut[4]	ethmac.rxethmac1.RxStart...	eth_rxethmac.v:411	CLKDV_RI_W::eth_rx_
4 +	4	ethmac.maccontrol1.transmitcontrol1.CtrlMux	ethmac.wishbone.TxAbort...	eth_wishbone.v:1925	eth_tx_clk_RI_W::CLKD
5 +	5	ethmac.maccontrol1.transmitcontrol1.CtrlMux	ethmac.wishbone.TxDone...	eth_wishbone.v:1908	eth_tx_clk_RI_W::CLKD
6 --	6	ethmac.ethreg1.CTRLMODER_0.DataOut[2]	ethmac.TxPauseRq_sync1	ethmac.v:950	CLKDV_RI_W::eth_tx_
7	6	ethmac.ethreg1.TXCTRL_2.DataOut[0]	ethmac.TxPauseRq_sync1	ethmac.v:950	CLKDV_RI_W::eth_tx_

Moral Hazard (1)

- Better tools => Designers take more risks

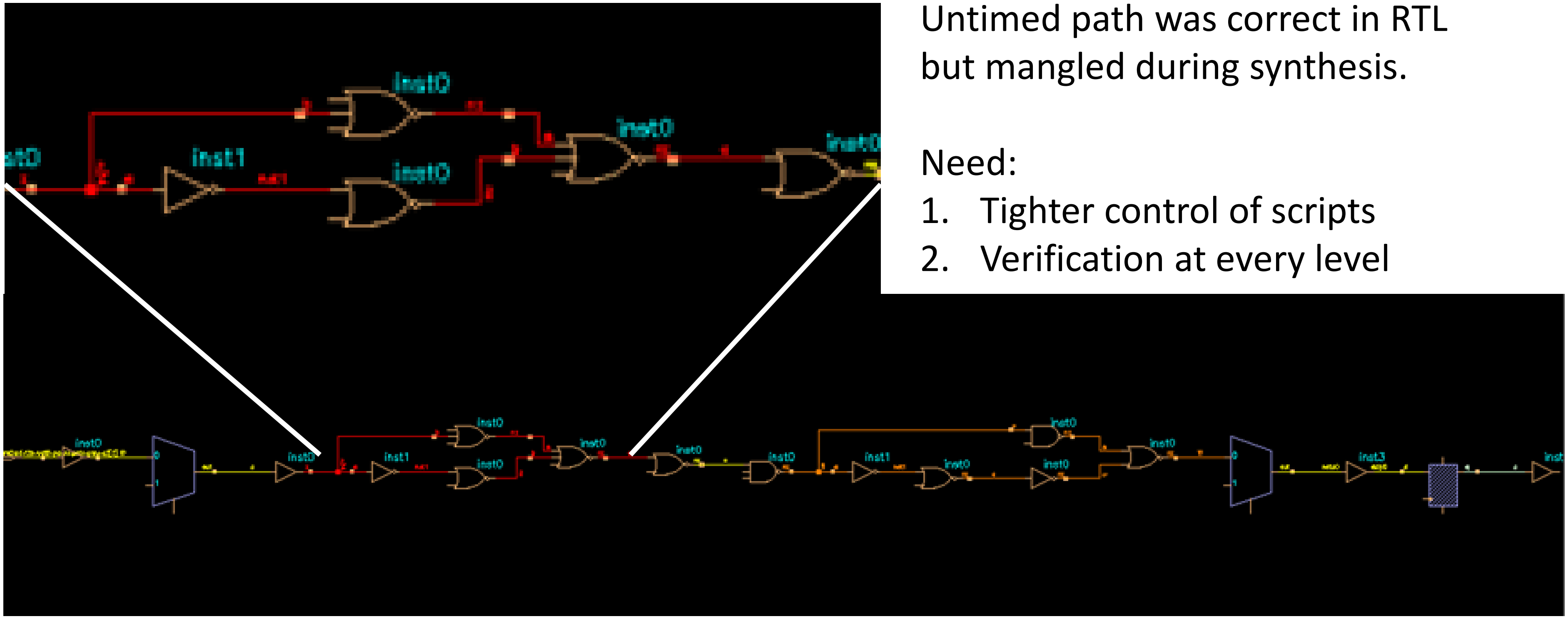


Mode 1: Clk1 and Clk2 are synchronous
Mode 2: Clk1 and Clk2 are asynchronous

Sneaky path causes a glitch

Moral Hazard (2)

- Better tools => Methodology is irrelevant

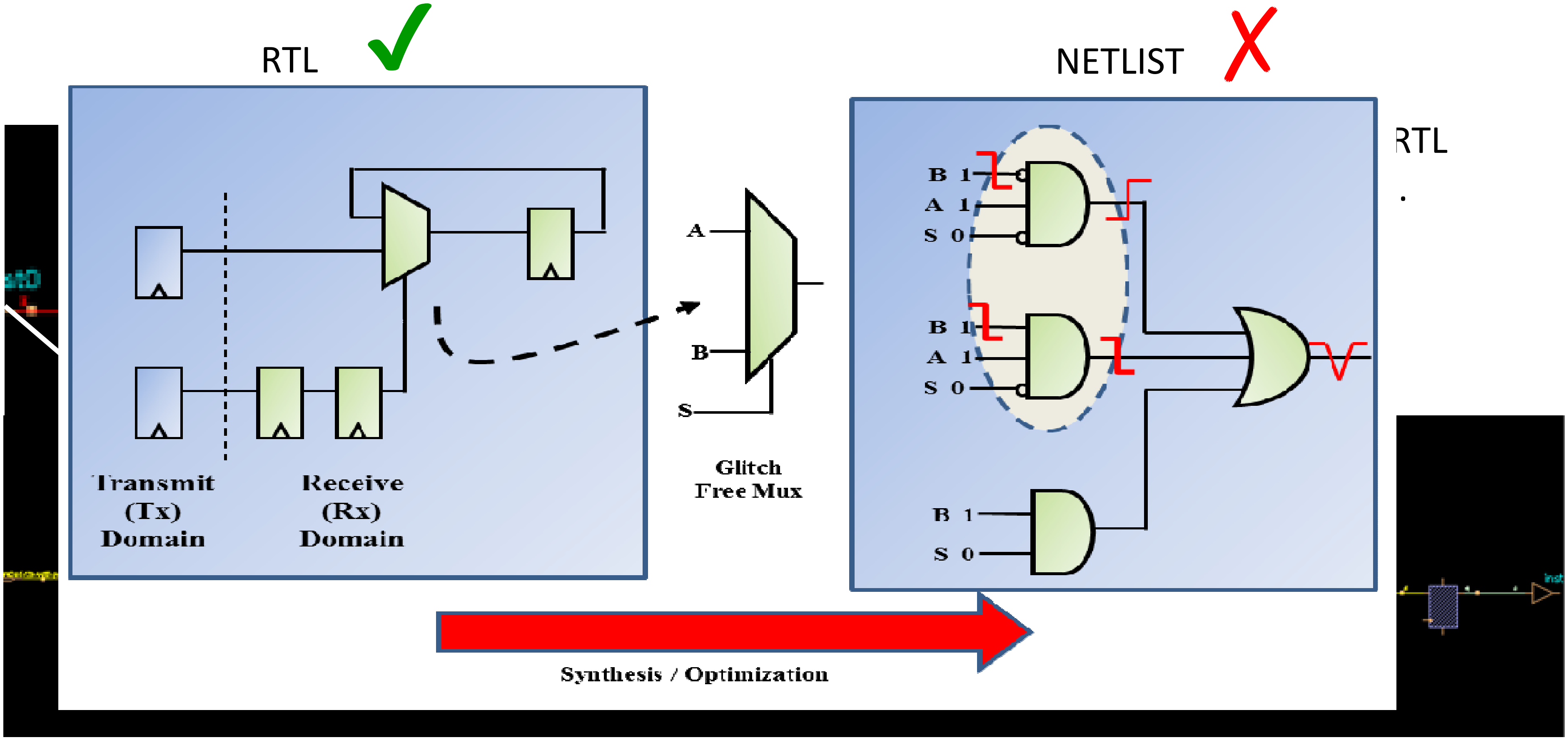


Untimed path was correct in RTL but mangled during synthesis.

Need:

1. Tighter control of scripts
2. Verification at every level

Moral Hazard (2)



Overall Impact of the New Paradigm is Salutary

REAL INTENT

Accelerating RTL Sign-off

- ✓ Exhaustive - No test benches
- ✓ Quick start and minimal setup
- ✓ Early detection - Helps prepare the design for simulation
- ✓ Sign-off on failure modes that are hard for simulation
- ✓ Address simulation’s limited semantics e.g. x-prop
- ✓ Parallelizes verification: Reduced simulation
- ✓ Shorter debug cycle time

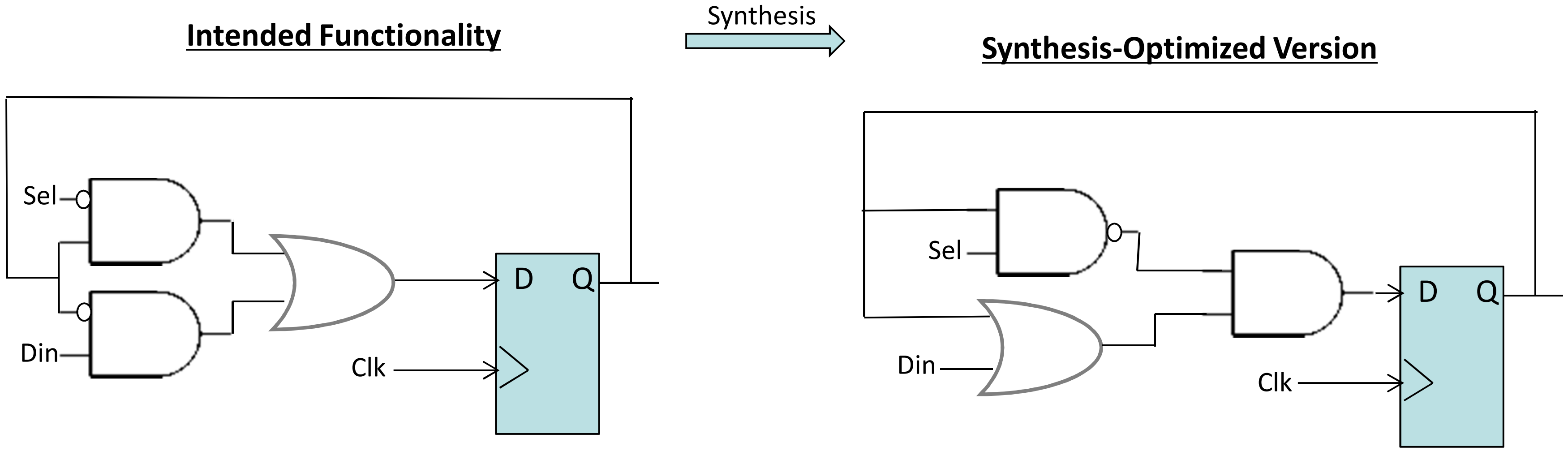
Narrows the Verification Gap

A New-Paradigm Template: X-Safety

REAL INTENT
Accelerating RTL Sign-off

Problem: Synthesis optimizes logic without knowing that X-pessimism is introduced

Observed in actual netlists



Initializes to 0

(Sel == 1 and Din == 0)

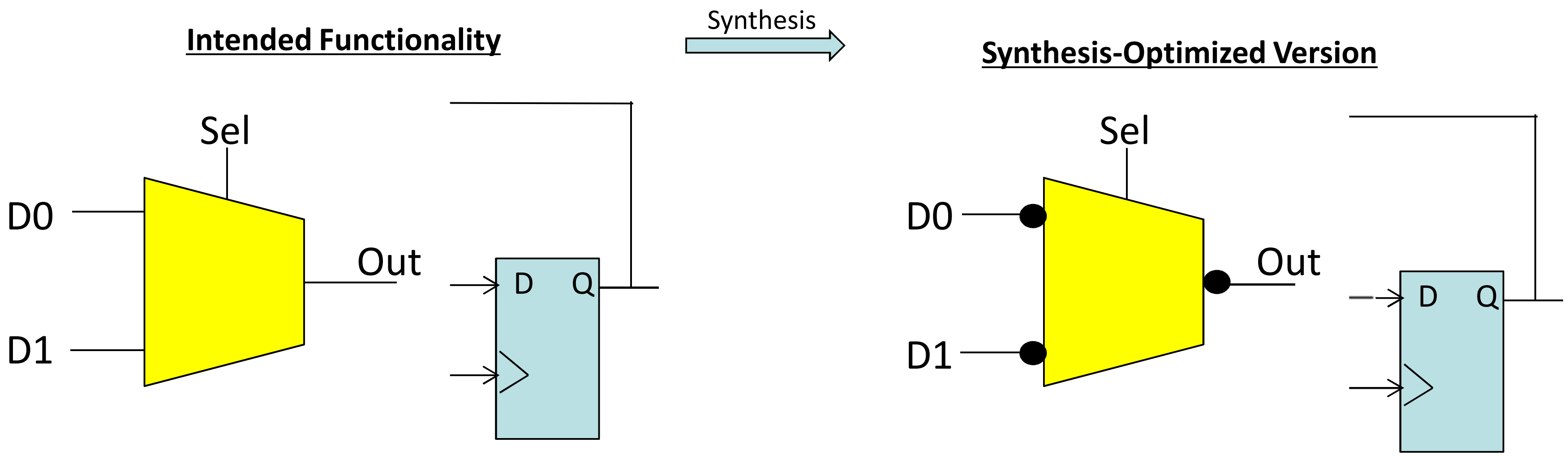
Will not be able to initialize to 0

(Sel == 1 and Din == 0)

A New-Paradigm Template: X-Safety

Problem: Synthesis optimizes logic without knowing that X-pessimism is introduced

Observed in actual netlists



Initializes to 0

(Sel == 1 and Din == 0)

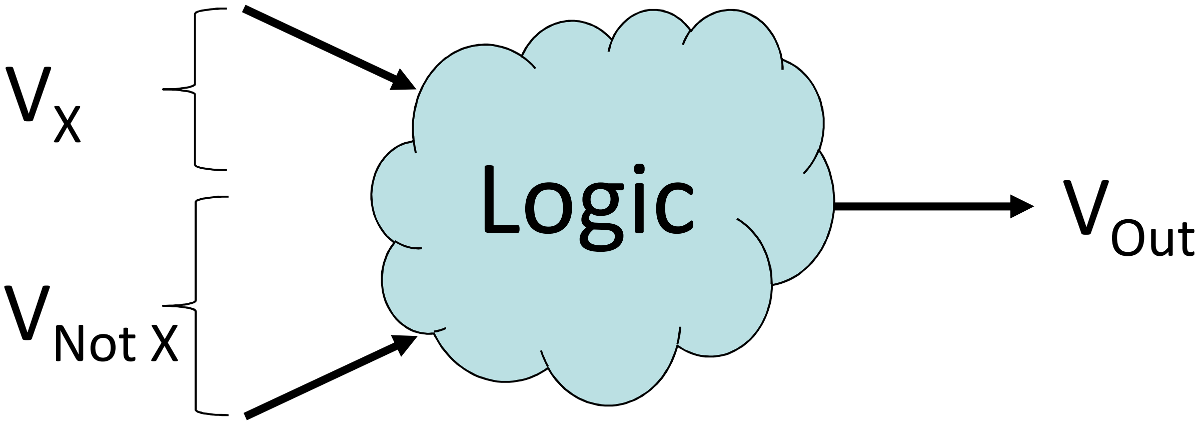
Will not be able to initialize to 0

(Sel == 1 and Din == 0)

“Trivial” Problem Needs State-of-art Algorithms

REAL INTENT
Accelerating RTL Sign-off

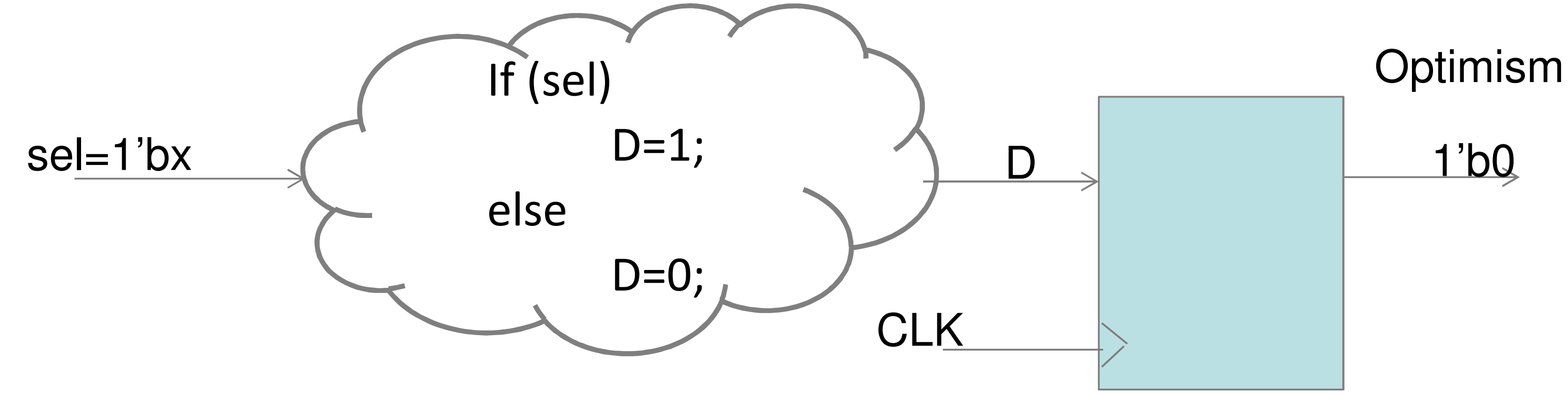
- X-pessimism analysis is conceptually a QBF problem



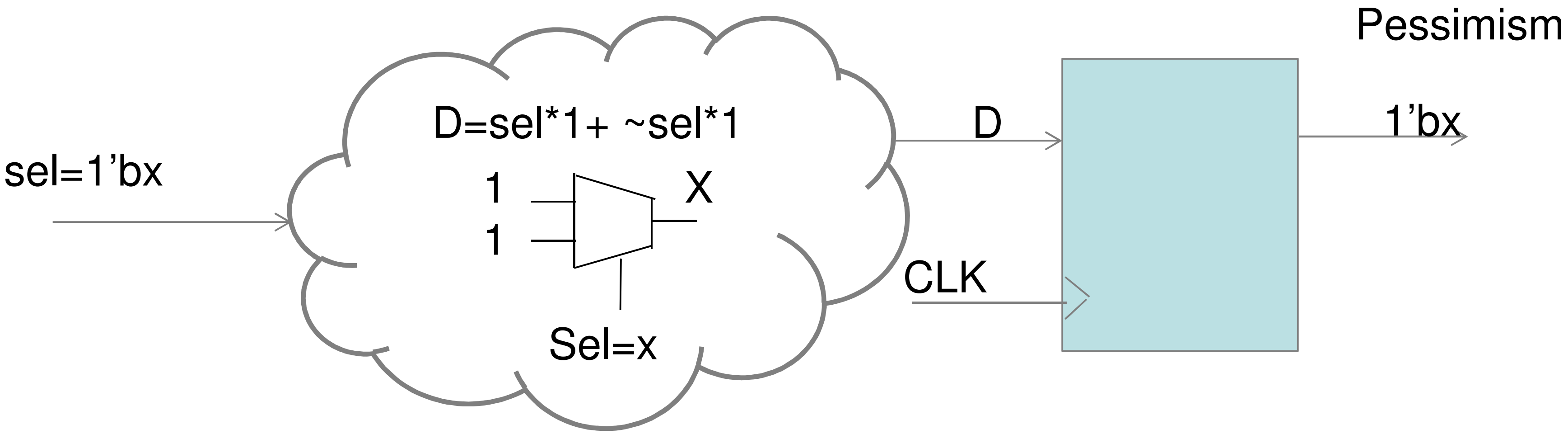
Is there a combination of $V_{Not X}$ such that the value of V_{Out} is the same for all projections of V_X?

$V_{Not X}$ and V_X are dynamic subsets of V_{In}

RTL and Netlist Simulations are Inaccurate in the Presence of X



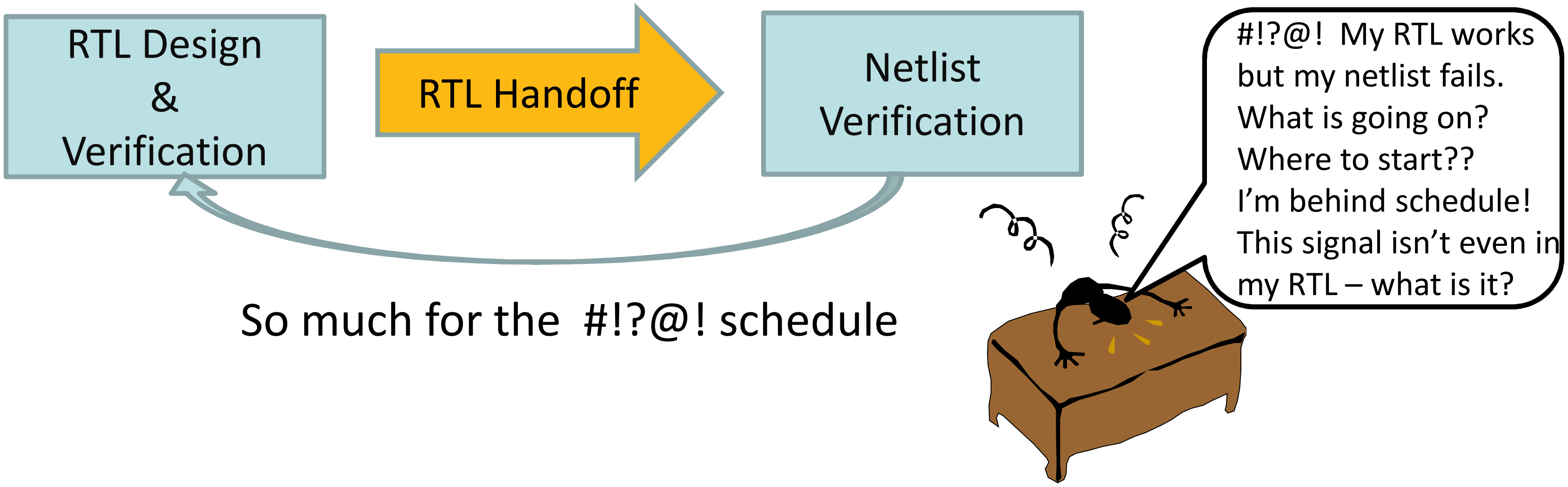
```
If (sel)
        D=1;
else
        D=0;
```



D=sel*1+ ~sel*1

X-Safe Design Is a High-Value Target

- Simulation behavior inaccurate
  - X's cause bugs to be missed at RTL
  - X's cause unnecessary additional X's at netlist
- Difficult to verify initialization in the presence of X's
- Gate level simulation bring up times are impacted by X's
  - Massive productivity loss



RTL Design & Verification

RTL Handoff

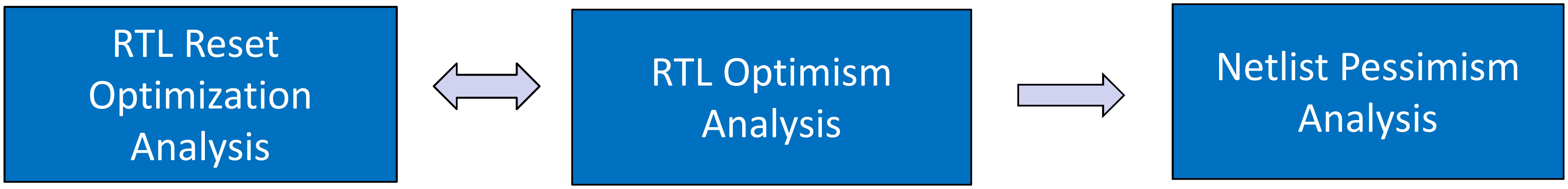
Netlist Verification

So much for the #!?@! schedule

#!?@! My RTL works but my netlist fails. What is going on? Where to start?? I'm behind schedule! This signal isn't even in my RTL – what is it?

Focus in on the Problem and Develop a Complete and Systematic Solution

REAL INTENT
Accelerating RTL Sign-off



Eliminate the X-source

Optimize the Reset Scheme

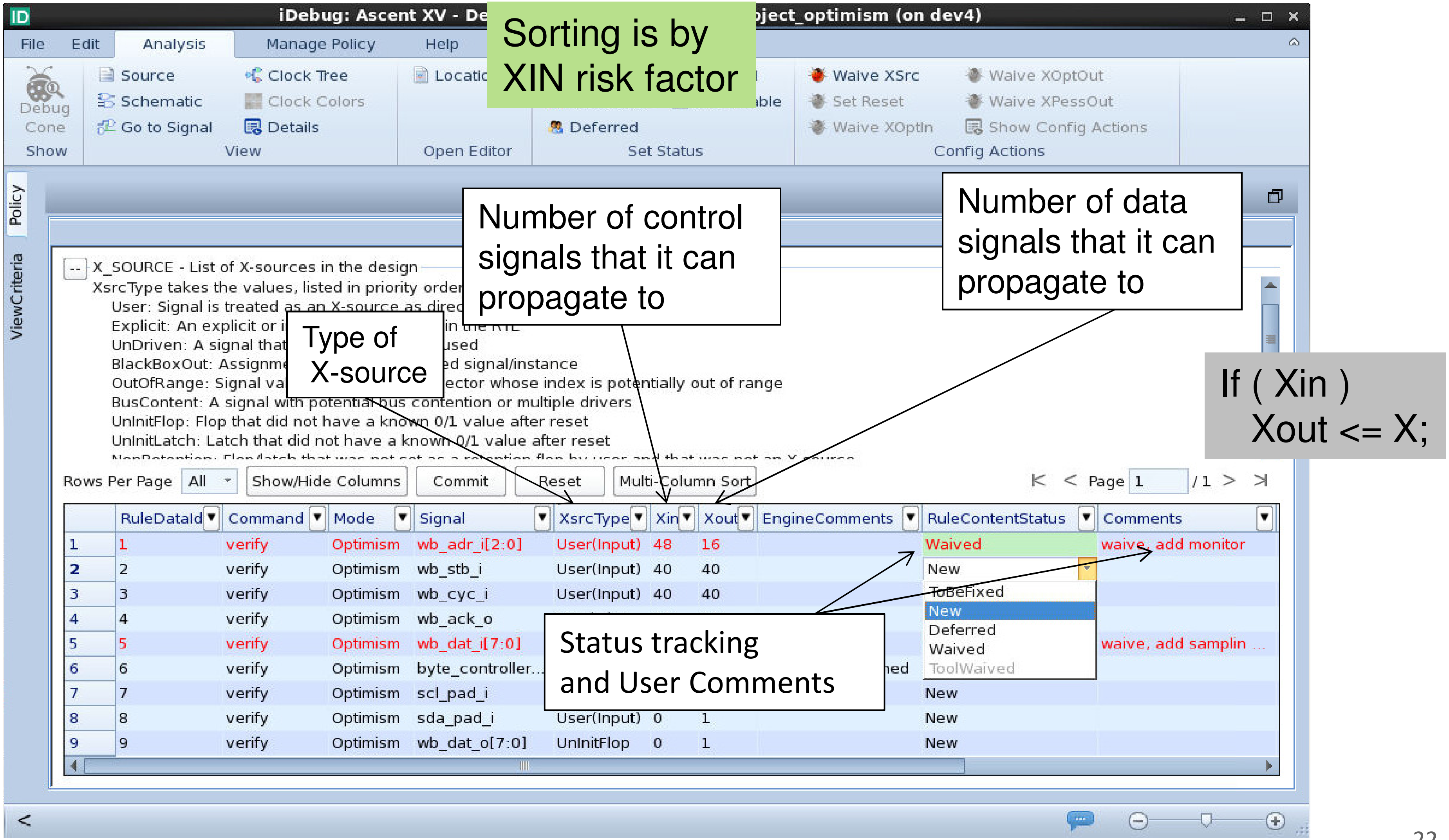
Code for X-accuracy and/or X-accurate Simulation

Pessimism Correction and/or Monitoring In Simulation

- X’s appear in netlist simulations that were not in RTL simulations due to pessimism and due to real X’s that were masked by optimism in RTL
- Must resolve the optimism at RTL and then correct the pessimism in netlist simulations to avoid simulation differences at netlist.

Context-Smart Reporting and Debug

REAL INTENT
Accelerating RTL Sign-off



Sorting is by XIN risk factor

Number of control signals that it can propagate to

Number of data signals that it can propagate to

Type of X-source

If (Xin)
Xout <= X;

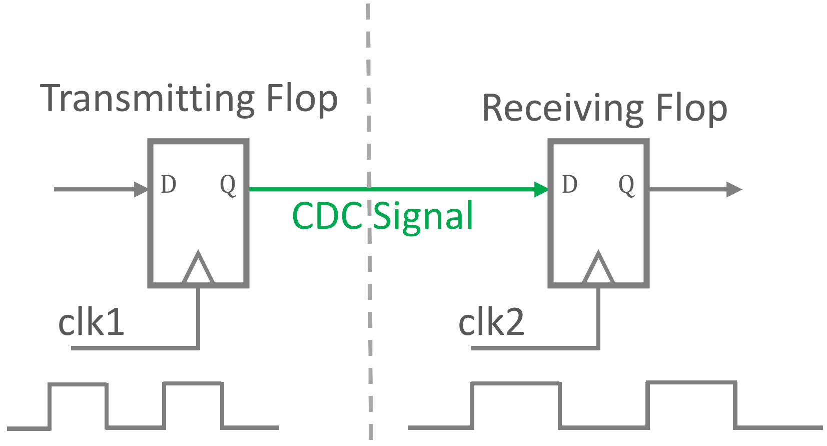
Status tracking and User Comments

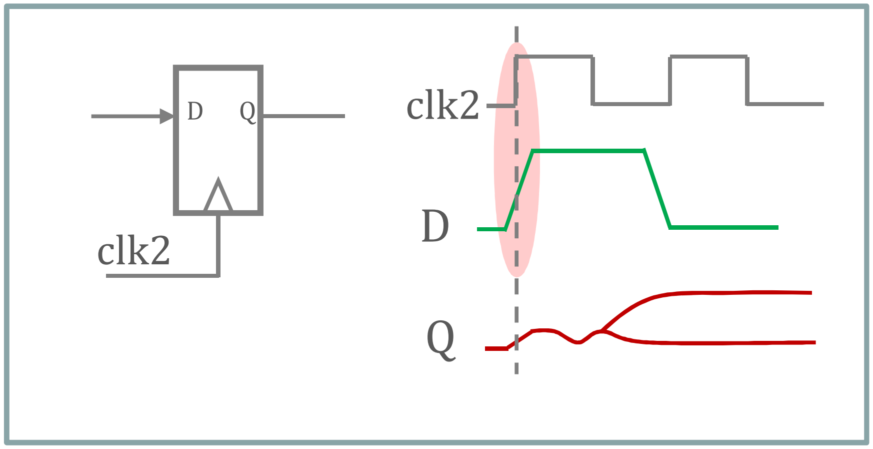
X_SOURCE - List of X-sources in the design
XsrcType takes the values, listed in priority order
User: Signal is treated as an X-source as direc
Explicit: An explicit or i in the RTL
UnDriven: A signal that used
BlackBoxOut: Assignme ed signal/instance
OutOfRange: Signal va ector whose index is potentially out of range
BusContent: A signal with potential bus contention or multiple drivers
UnInitFlop: Flop that did not have a known 0/1 value after reset
UnInitLatch: Latch that did not have a known 0/1 value after reset

	RuleDataId	Command	Mode	Signal	XsrcType	Xin	Xout	EngineComments	RuleContentStatus	Comments
1	1	verify	Optimism	wb_adr_i[2:0]	User(Input)	48	16		Waived	waive, add monitor
2	2	verify	Optimism	wb_stb_i	User(Input)	40	40		New	
3	3	verify	Optimism	wb_cyc_i	User(Input)	40	40			
4	4	verify	Optimism	wb_ack_o						
5	5	verify	Optimism	wb_dat_i[7:0]						waive, add samplin ...
6	6	verify	Optimism	byte_controller...				ned		
7	7	verify	Optimism	scl_pad_i					New	
8	8	verify	Optimism	sda_pad_i	User(Input)	0	1		New	
9	9	verify	Optimism	wb_dat_o[7:0]	UnInitFlop	0	1		New	

Another New-Paradigm Example: CDC

REAL INTENT
Accelerating RTL Sign-off



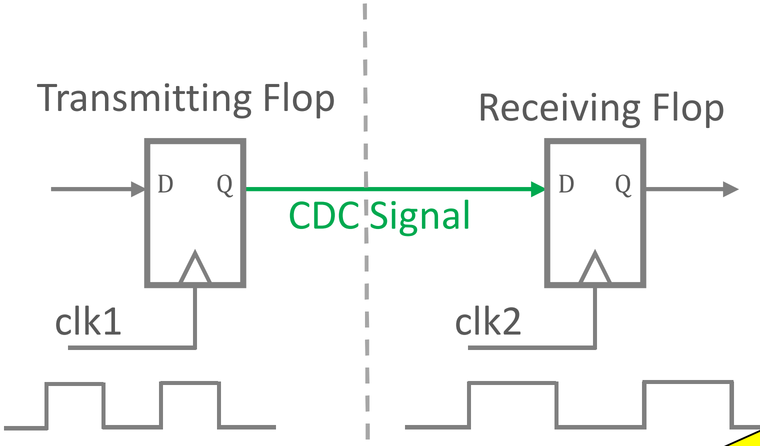


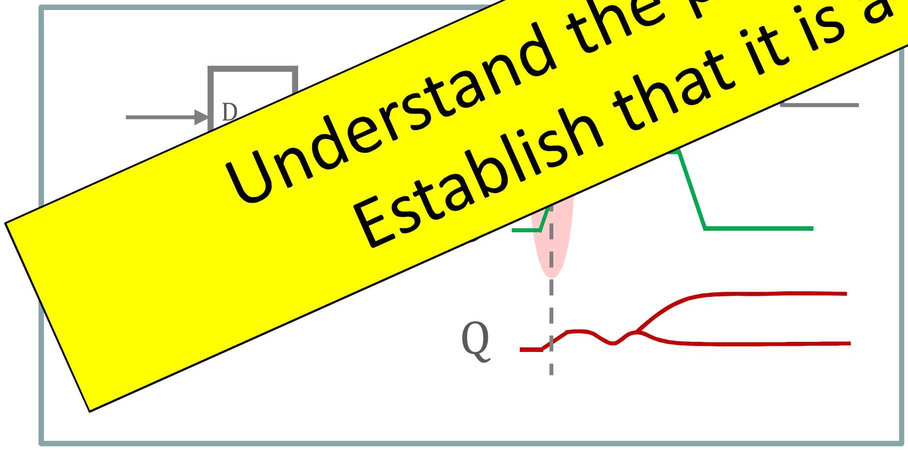
- **The Metastability Problem**
  - When input changes within setup/hold window, the output of the flop becomes metastable, could settle into either 0 or 1

- **The Challenges**
  - Hard to detect and diagnose (with simulation or in the lab)
  - Very high number of CDC crossings
  - Variety of ways of implementing the crossings

- **Impact**
  - Chip failure in the field
  - Expensive to fix

Another New-Paradigm Example: CDC





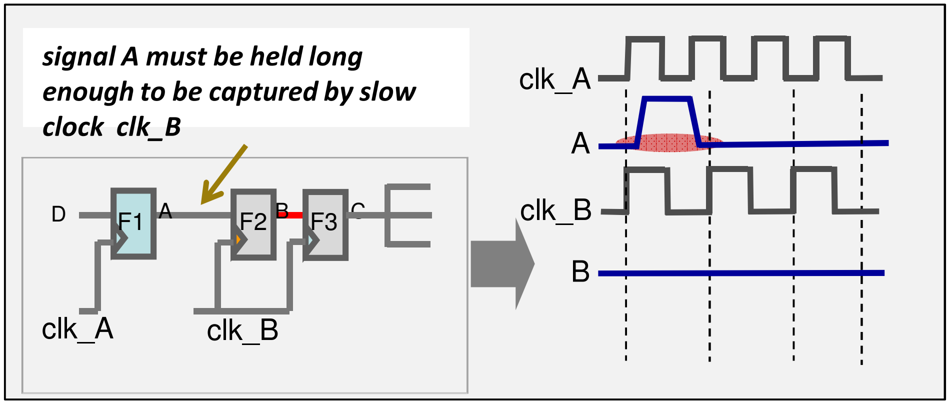
Understand the problem at fundamental level
Establish that it is a high-value problem

- **The Metastability Problem**
  - When input changes w setup/hold wi the flop uld s
- **ges**
  - o detect and diagnose (with simulation or in the lab)
  - Very high number of CDC crossings
  - Variety of ways of implementing the crossings

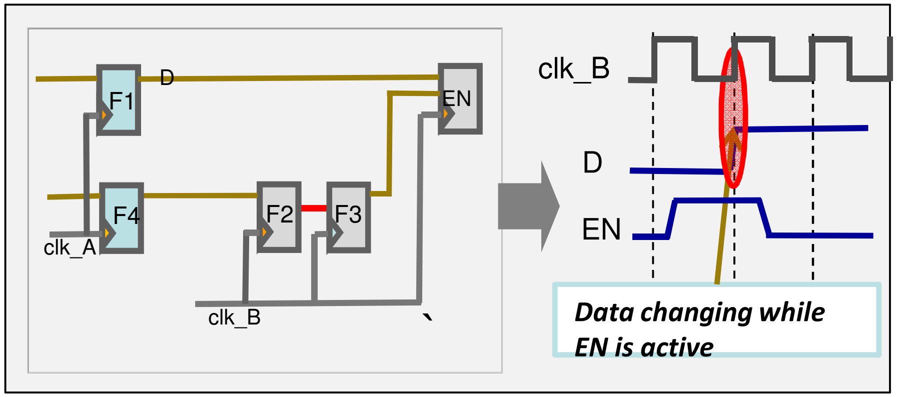
- **Impact**
  - Chip failure in the field
  - Expensive to fix

Typical CDC Issues

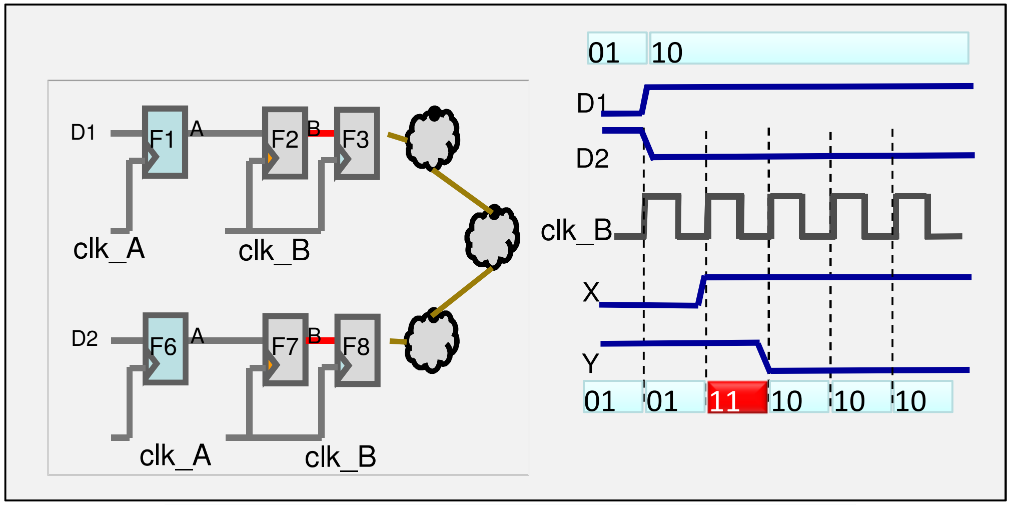
REAL INTENT
Accelerating RTL Sign-off


signal A must be held long enough to be captured by slow clock clk_B

Data loss in fast to slow transfer


Data changing while EN is active

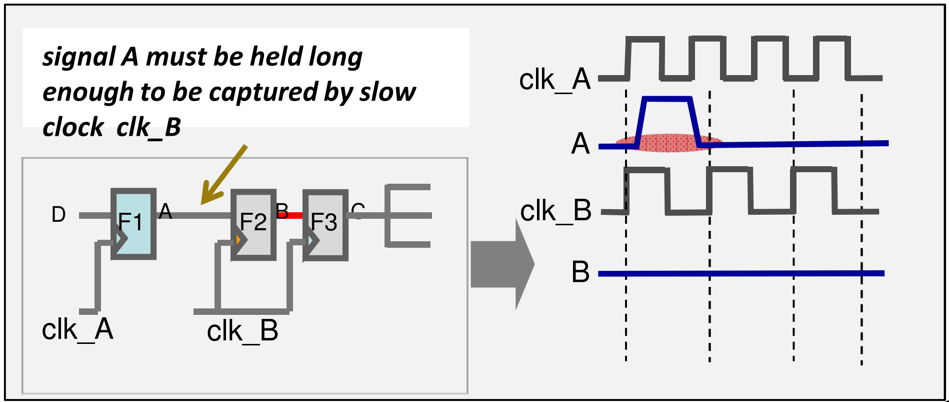
Improper data enable sequence



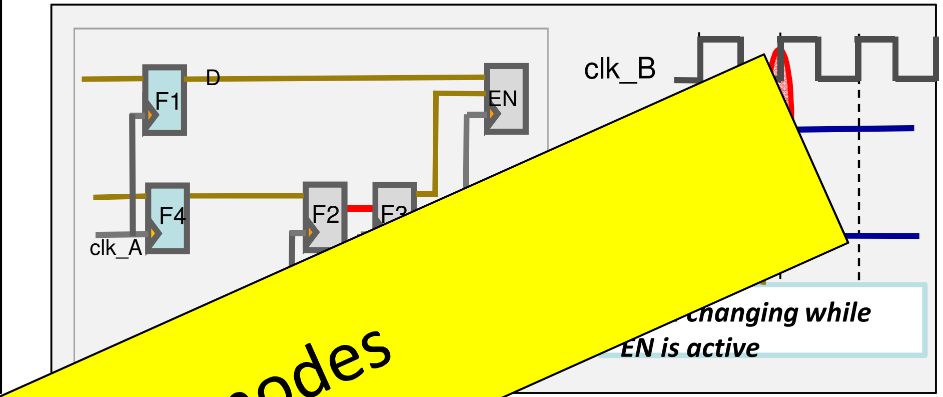
Re-convergence of synced signals

Typical CDC Issues

REAL INTENT
Accelerating RTL Sign-off


signal A must be held long enough to be captured by slow clock clk_B

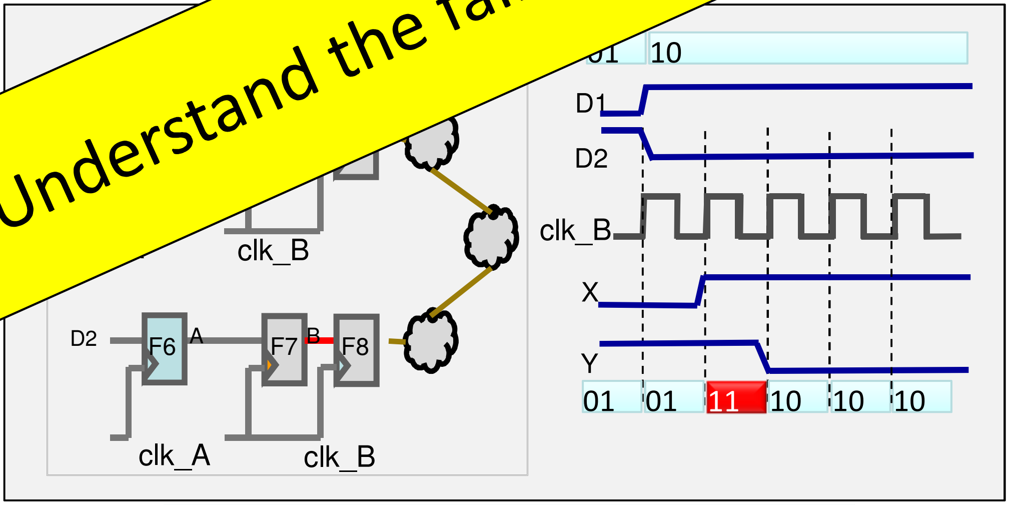
Data loss in fast to slow transfer



changing while EN is active

ata enable sequence

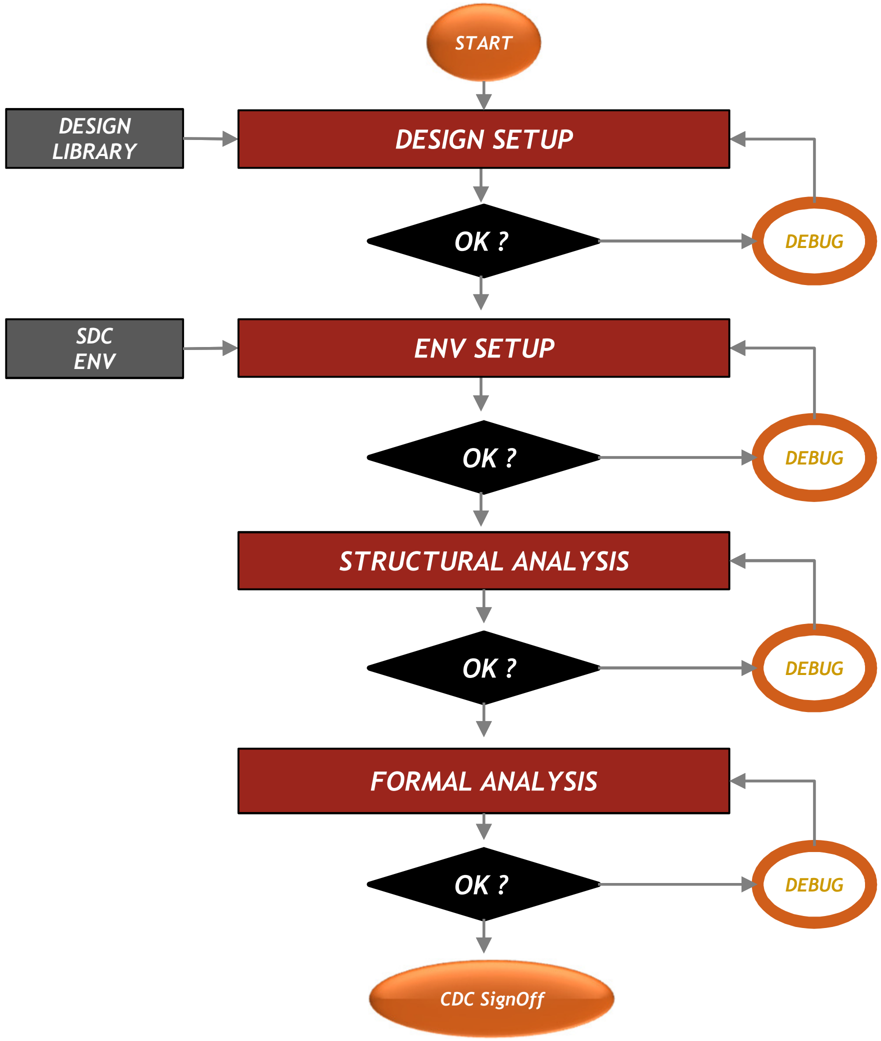
Understand the failure modes



Re-convergence of synced signals

Systematic CDC Methodology

REAL INTENT
Accelerating RTL Sign-off



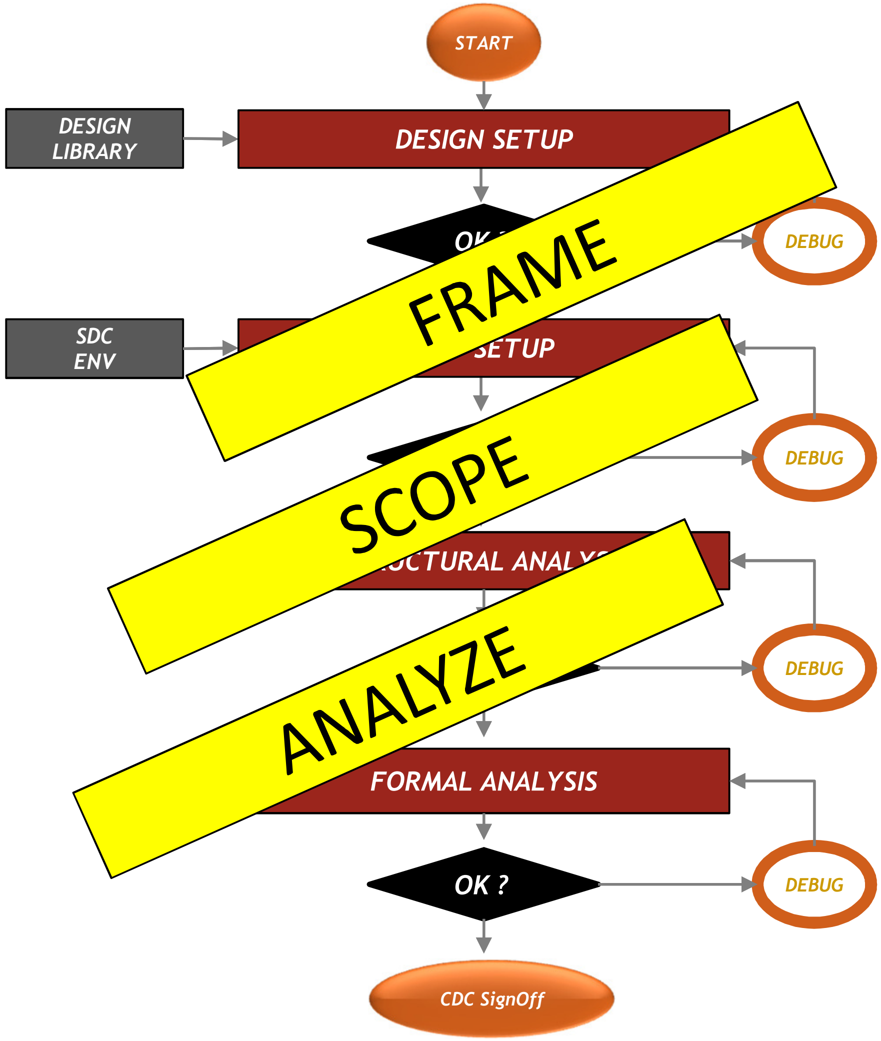
Important checks Setup stage

- Missing clocks and derived clocks
- Missing clock relationships
- Missing boundary conditions
- Missing resets
- Conflicts between env specs and/or design

- **Important Checks Structural analysis**
  - DATA and CNTL
  - Glitch
  - CNTL with multiple fanouts
  - Reconvergence
  - Resets crossing domains

- **Important Checks Formal analysis**
  - Data Stability
  - Pulse Width
  - Glitch Analysis
  - GRAY CODE Checks

Systematic CDC Methodology



START

DESIGN LIBRARY

DESIGN SETUP

OK

DEBUG

SDC ENV

SETUP

DEBUG

UCTURAL ANALY

DEBUG

FORMAL ANALYSIS

OK ?

DEBUG

CDC SignOff

FRAME

SCOPE

ANALYZE

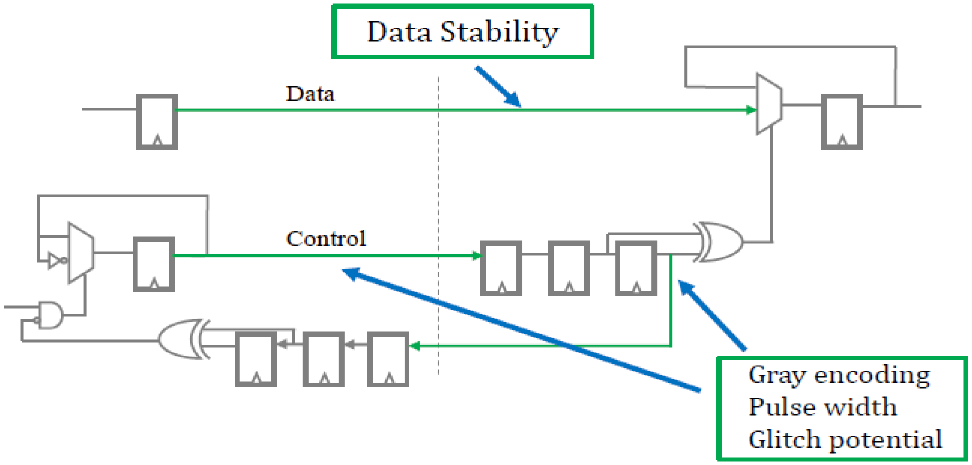
Important checks Setup stage

- Missing clocks and derived clocks
- Missing clock relationships
- Missing boundary conditions
- Missing resets
- Conflicts between env specs and/or design

- **Important Checks Structural analysis**
  - DATA and CNTL
  - Glitch
  - CNTL with multiple fanouts
  - Reconvergence
  - Resets crossing domains

- **Important Checks Formal analysis**
  - Data Stability
  - Pulse Width
  - Glitch Analysis
  - GRAY CODE Checks

Formal CDC Verification



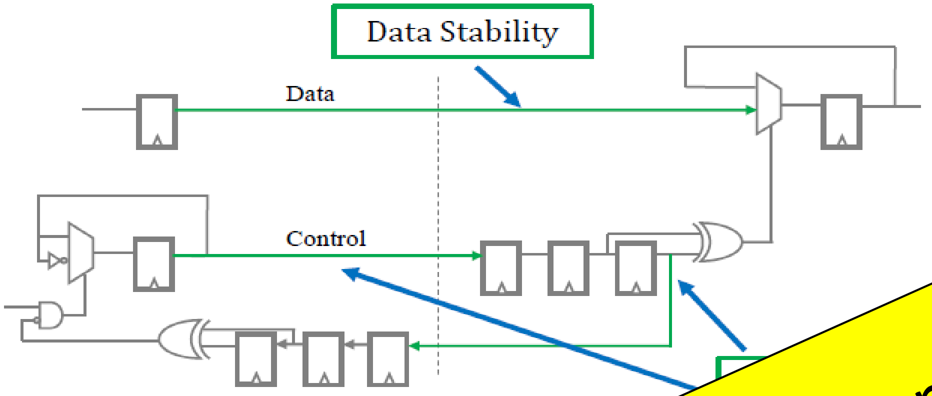
Data Stability

Gray encoding
Pulse width
Glitch potential

Formal Analysis	Description
Data stability	Check for safe data crossings across asynchronous clock domains
Gray code	Check that FIFO-related reconvergent control signals are Gray coded
Glitch analysis	Check that there is no glitch in the combinational circuit that can cause an incorrect value to be captured
Pulse width	Check that control crossings are held long enough to be sampled at the receiving domain

Formal CDC Verification

REAL INTENT
Accelerating RTL Sign-off



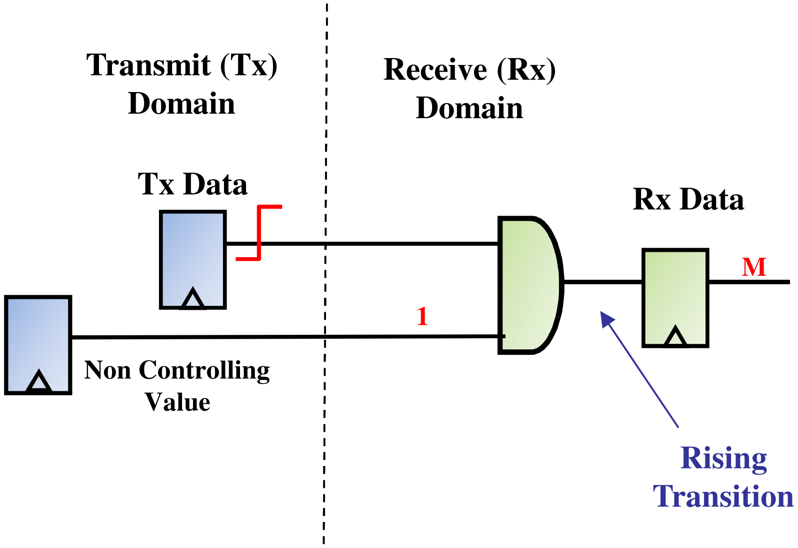
Implicit Generation of Assertions

Formal Analysis	Descripti
Data stability	Ch data crossings across ock domains
Gray co	at FIFO-related reconvergent control nals are Gray coded
alysis	Check that there is no glitch in the combinational circuit that can cause an incorrect value to be captured
Pulse width	Check that control crossings are held long enough to be sampled at the receiving domain

Basic Data Stability Check

REAL INTENT
Accelerating RTL Sign-off

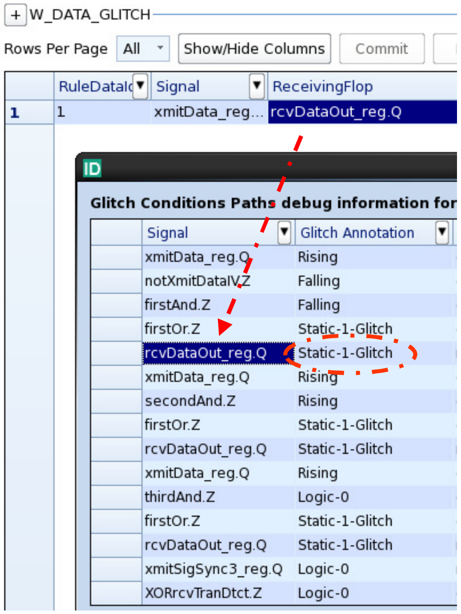
- Rising/Falling transition on Tx Flop lead to Rising/Falling transition on Rx Flop at next edge or Rx Clock.



Glitch-Aware Check

REAL INTENT
Accelerating RTL Sign-off

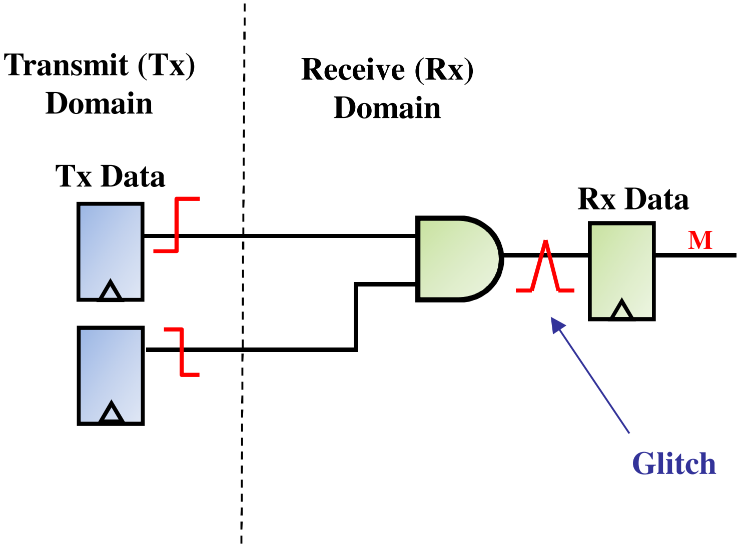
- Opposing transitions on TxFlops lead to a glitch on Rx Flop



	RuleDataId	Signal	ReceivingFlop
1	1	xmitData_reg...	rcvDataOut_reg.Q

Glitch Conditions Paths debug information for

Signal	Glitch Annotation
xmitData_reg.Q	Rising
notXmitDataIV.Z	Falling
firstAnd.Z	Falling
firstOr.Z	Static-1-Glitch
rcvDataOut_reg.Q	Static-1-Glitch
xmitData_reg.Q	Rising
secondAnd.Z	Rising
firstOr.Z	Static-1-Glitch
rcvDataOut_reg.Q	Static-1-Glitch
xmitData_reg.Q	Rising
thirdAnd.Z	Logic-0
firstOr.Z	Static-1-Glitch
rcvDataOut_reg.Q	Static-1-Glitch
xmitSigSync3_reg.Q	Logic-0
XORrcvTranDtct.Z	Logic-0



Formal CDC Verification

REAL INTENT
Accelerating RTL Sign-off

- Parallel Formal for high throughput
  - Almost 100% coverage of failure trace, pass or deep-bounded pass
- Constraints support
  - Enable SVA/PSL constraints on the fly
  - Extract constraint dependence
  - Show in the debug
- Flexible tool control
  - Fast (re)start of formal analysis iterations
  - Inform users on formal run progress and completion status

Formal CDC Verification

REAL INTENT
Accelerating RTL Sign-off

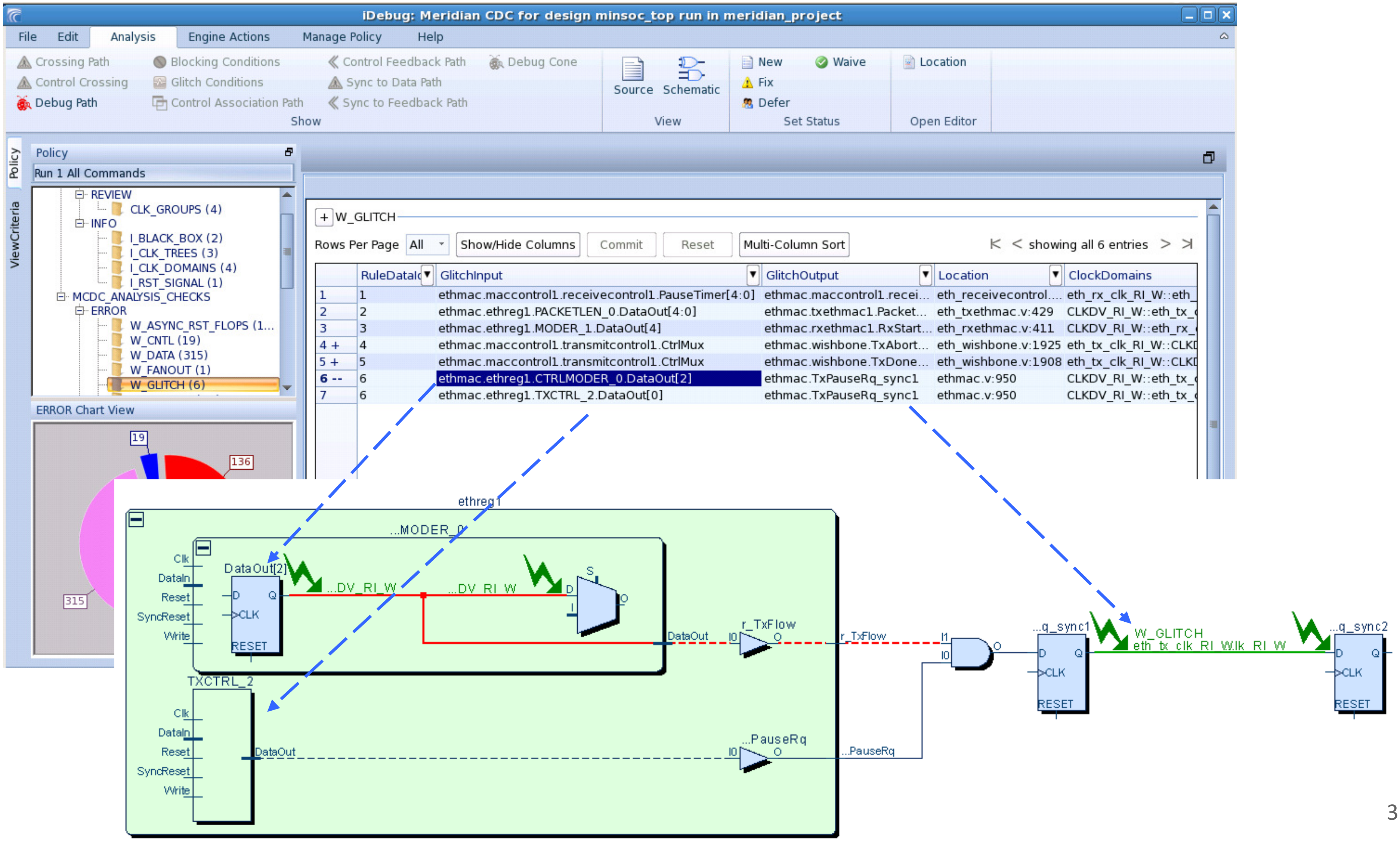
- Parallel Formal for high throughput
  - Almost 100% coverage of failure trace, pass or dee ed pass
- Constraints support
  - Enable SVA/PSL constrai
  - Extract constraint
  - Show in the
- El ntrol
  - tart of formal analysis iterations
  - orm users on formal run progress and completion status

Enable throughput and deep-checking in formal analysis

Context-Smart Debug

REAL INTENT

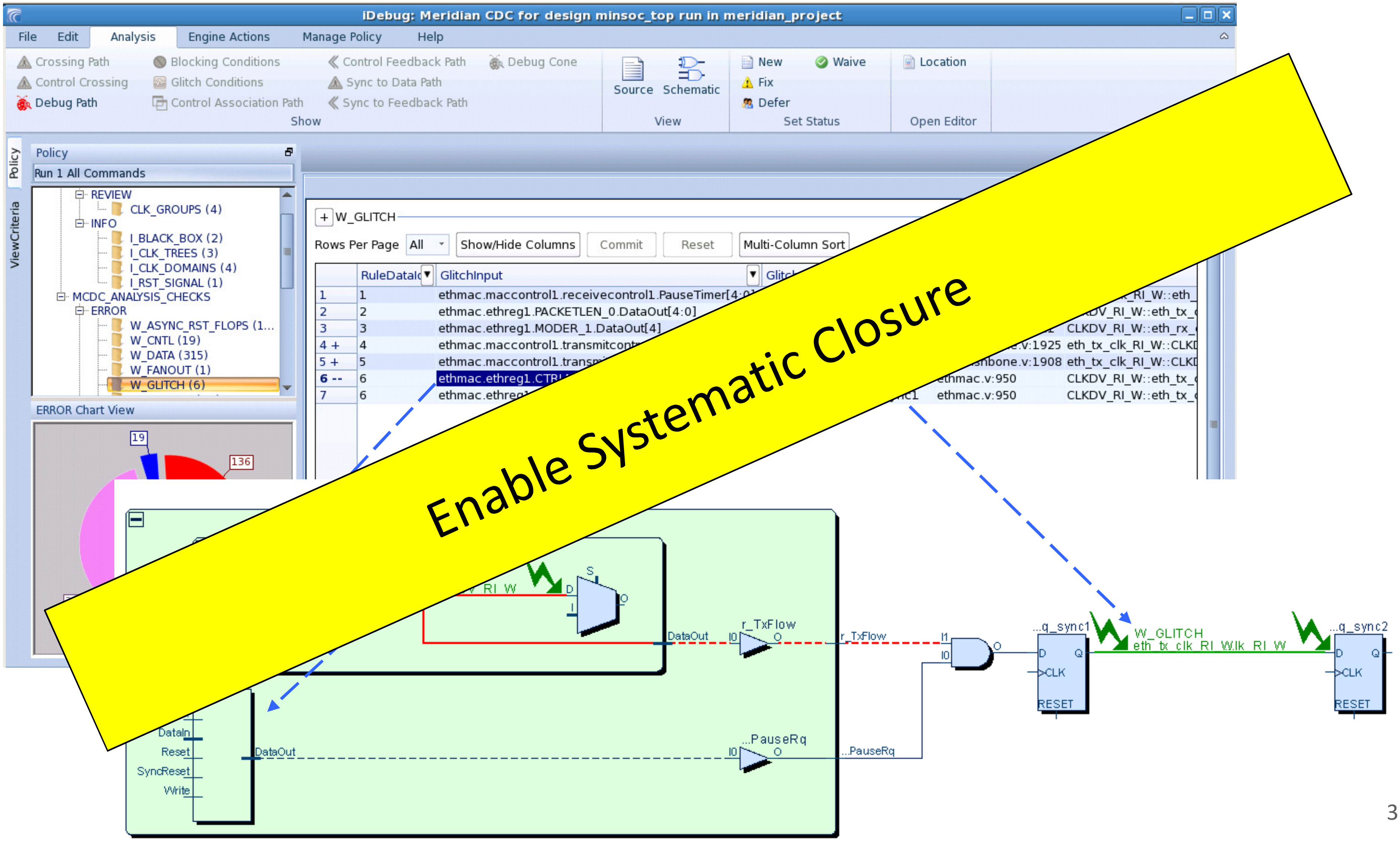
Accelerating RTL Sign-off



iDebug: Meridian CDC for design minsoc_top run in meridian_project

	RuleDataId	GlitchInput	GlitchOutput	Location	ClockDomains
1	1	ethmac.maccontrol1.receivecontrol1.PauseTimer[4:0]	ethmac.maccontrol1.recei...	eth_receivecontrol....	eth_rx_clk_RI_W::eth_
2	2	ethmac.ethreg1.PACKETLEN_0.DataOut[4:0]	ethmac.txethmac1.Packet...	eth_txethmac.v:429	CLKDV_RI_W::eth_tx_
3	3	ethmac.ethreg1.MODER_1.DataOut[4]	ethmac.rxethmac1.RxStart...	eth_rxethmac.v:411	CLKDV_RI_W::eth_rx_
4 +	4	ethmac.maccontrol1.transmitcontrol1.CtrlMux	ethmac.wishbone.TxAbort...	eth_wishbone.v:1925	eth_tx_clk_RI_W::CLKD
5 +	5	ethmac.maccontrol1.transmitcontrol1.CtrlMux	ethmac.wishbone.TxDone...	eth_wishbone.v:1908	eth_tx_clk_RI_W::CLKD
6 --	6	ethmac.ethreg1.CTRLMODER_0.DataOut[2]	ethmac.TxPauseRq_sync1	ethmac.v:950	CLKDV_RI_W::eth_tx_
7	6	ethmac.ethreg1.TXCTRL_2.DataOut[0]	ethmac.TxPauseRq_sync1	ethmac.v:950	CLKDV_RI_W::eth_tx_

Context-Smart Debug



iDebug: Meridian CDC for design minsoc_top run in meridian_project

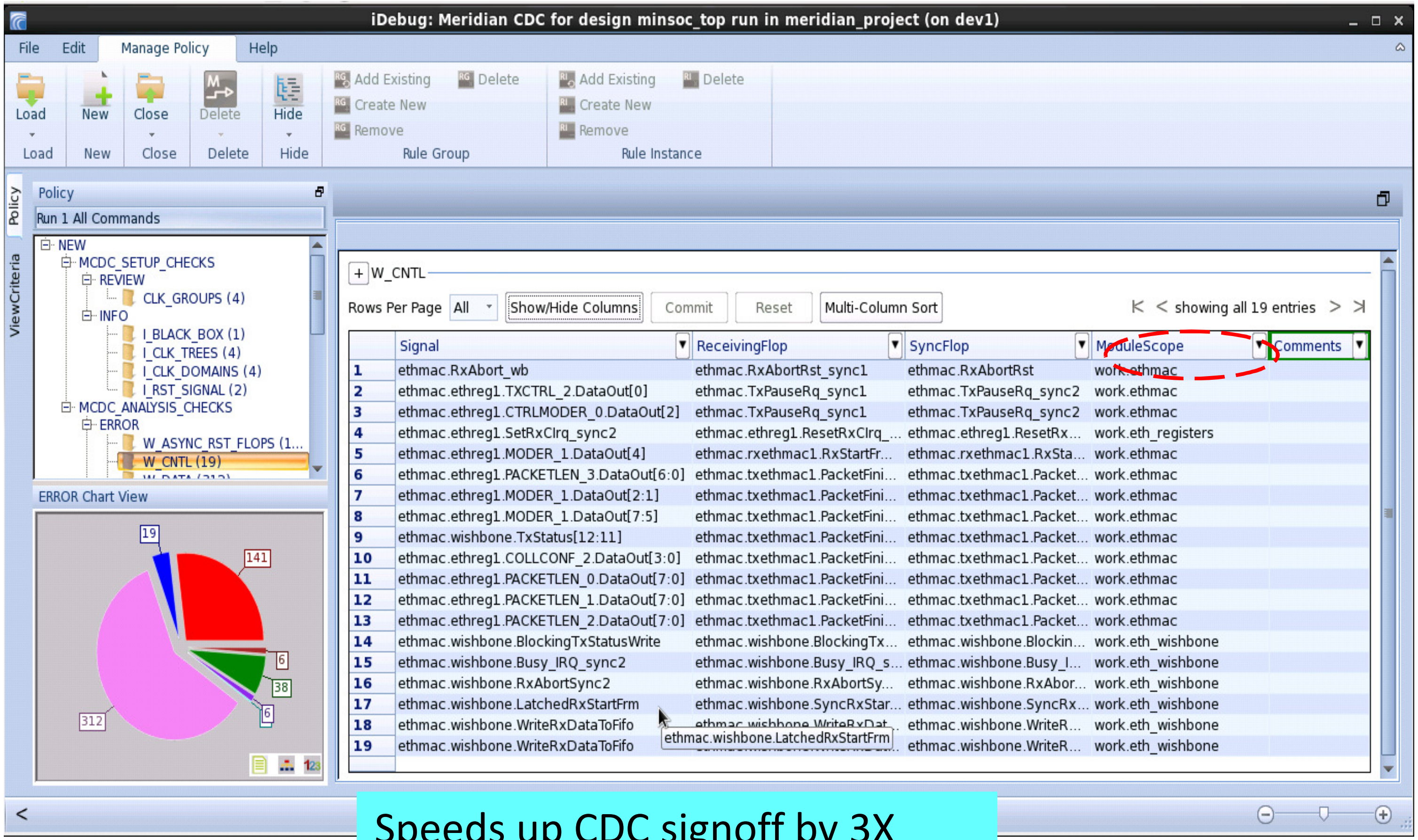
	RuleDataIc	GlitchInput	Glitc		
1	1	ethmac.maccontrol1.receivecontrol1.PauseTimer[4:0]			_RI_W::eth_
2	2	ethmac.ethreg1.PACKETLEN_0.DataOut[4:0]			DV_RI_W::eth_tx_
3	3	ethmac.ethreg1.MODER_1.DataOut[4]			CLKDV_RI_W::eth_rx_
4 +	4	ethmac.maccontrol1.transmitcontr		.v:1925	eth_tx_clk_RI_W::CLKD
5 +	5	ethmac.maccontrol1.transm		shbone.v:1908	eth_tx_clk_RI_W::CLKD
6 --	6	ethmac.ethreg1.CTRL		ethmac.v:950	CLKDV_RI_W::eth_tx_
7	6	ethmac.ethreg1	c1	ethmac.v:950	CLKDV_RI_W::eth_tx_

Enable Systematic Closure

Scope Based Reporting: Simultaneous Chip Level and Block level results

REAL INTENT
Accelerating RTL Sign-off

- ModuleScope - the scope of the design violation is well contained in
- Available for all the rules
- Accessible through GUI or CLI for quick debug/SignOff

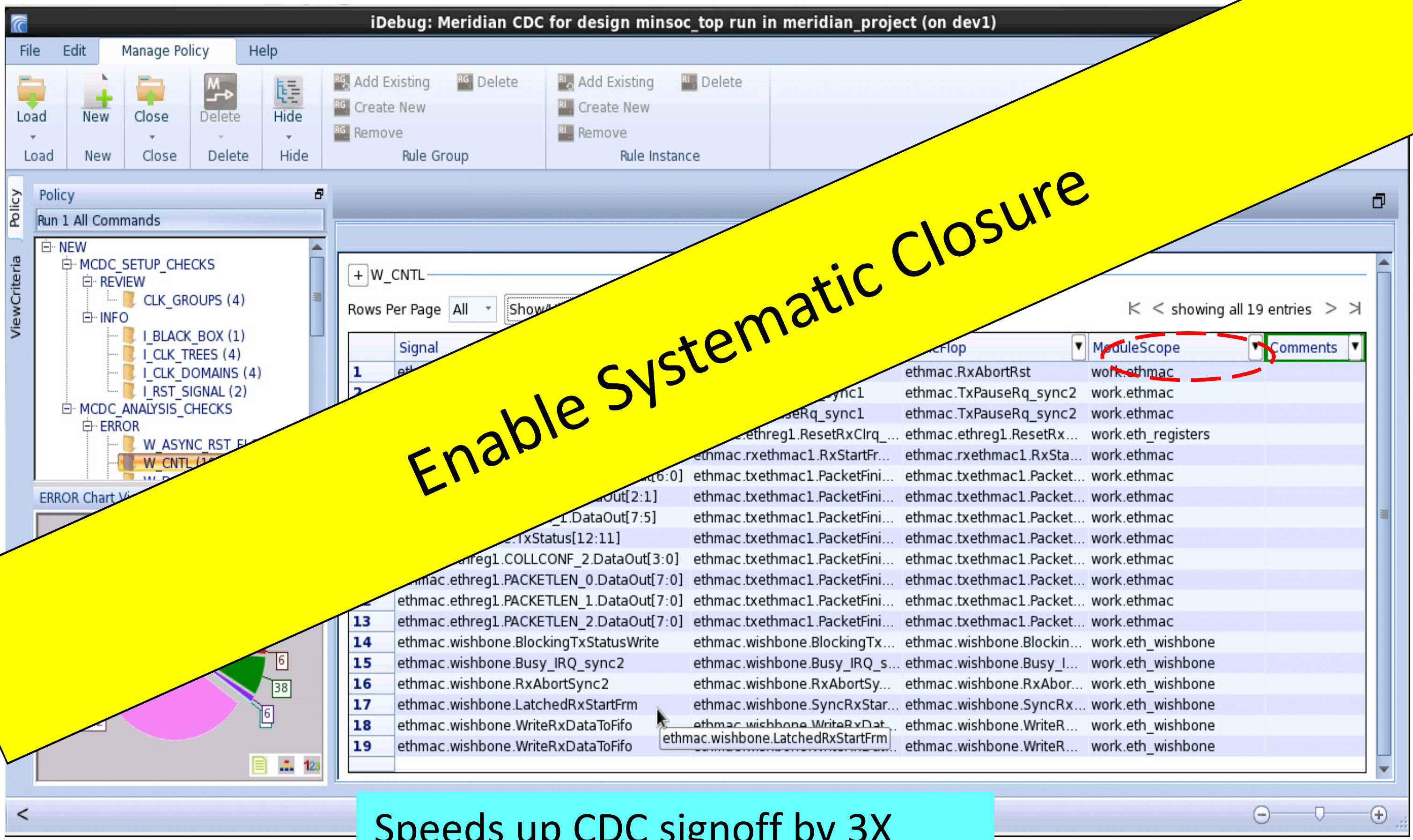


	Signal	ReceivingFlop	SyncFlop	ModuleScope	Comments
1	ethmac.RxAbort_wb	ethmac.RxAbortRst_sync1	ethmac.RxAbortRst	work.ethmac	
2	ethmac.ethreg1.TXCTRL_2.DataOut[0]	ethmac.TxPauseRq_sync1	ethmac.TxPauseRq_sync2	work.ethmac	
3	ethmac.ethreg1.CTRLMODER_0.DataOut[2]	ethmac.TxPauseRq_sync1	ethmac.TxPauseRq_sync2	work.ethmac	
4	ethmac.ethreg1.SetRxClrq_sync2	ethmac.ethreg1.ResetRxClrq_...	ethmac.ethreg1.ResetRx...	work.eth_registers	
5	ethmac.ethreg1.MODER_1.DataOut[4]	ethmac.rxethmac1.RxStartFr...	ethmac.rxethmac1.RxSta...	work.ethmac	
6	ethmac.ethreg1.PACKETLEN_3.DataOut[6:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
7	ethmac.ethreg1.MODER_1.DataOut[2:1]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
8	ethmac.ethreg1.MODER_1.DataOut[7:5]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
9	ethmac.wishbone.TxStatus[12:11]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
10	ethmac.ethreg1.COLLCONF_2.DataOut[3:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
11	ethmac.ethreg1.PACKETLEN_0.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
12	ethmac.ethreg1.PACKETLEN_1.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
13	ethmac.ethreg1.PACKETLEN_2.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
14	ethmac.wishbone.BlockingTxStatusWrite	ethmac.wishbone.BlockingTx...	ethmac.wishbone.Blockin...	work.eth_wishbone	
15	ethmac.wishbone.Busy_IRQ_sync2	ethmac.wishbone.Busy_IRQ_s...	ethmac.wishbone.Busy_I...	work.eth_wishbone	
16	ethmac.wishbone.RxAbortSync2	ethmac.wishbone.RxAbortSy...	ethmac.wishbone.RxAbor...	work.eth_wishbone	
17	ethmac.wishbone.LatchedRxStartFrm	ethmac.wishbone.SyncRxStar...	ethmac.wishbone.SyncRx...	work.eth_wishbone	
18	ethmac.wishbone.WriteRxDataToFifo	ethmac.wishbone.WriteRxDat...	ethmac.wishbone.WriteR...	work.eth_wishbone	
19	ethmac.wishbone.WriteRxDataToFifo	[illegible]	ethmac.wishbone.WriteR...	work.eth_wishbone	

Speeds up CDC signoff by 3X

Scope Based Reporting: Simultaneous Chip Level and Block level results

- ModuleScope - the scope of the design violation is well contained in
- Available for all the rules
- Accessible through GUI or CLI for quick debug/SignOff



	Signal		cFlop	ModuleScope	Comments
1	et		ethmac.RxAbortRst	work.ethmac	
2		ync1	ethmac.TxPauseRq_sync2	work.ethmac	
		seRq_sync1	ethmac.TxPauseRq_sync2	work.ethmac	
		.ethreg1.ResetRxCIrq_...	ethmac.ethreg1.ResetRx...	work.eth_registers	
		thmac.rxethmac1.RxStartFr...	ethmac.rxethmac1.RxSta...	work.ethmac	
	[6:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	Out[2:1]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	1.DataOut[7:5]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	.TxStatus[12:11]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	reg1.COLLCONF_2.DataOut[3:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	mac.ethreg1.PACKETLEN_0.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
	ethmac.ethreg1.PACKETLEN_1.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
13	ethmac.ethreg1.PACKETLEN_2.DataOut[7:0]	ethmac.txethmac1.PacketFini...	ethmac.txethmac1.Packet...	work.ethmac	
14	ethmac.wishbone.BlockingTxStatusWrite	ethmac.wishbone.BlockingTx...	ethmac.wishbone.Blockin...	work.eth_wishbone	
15	ethmac.wishbone.Busy_IRQ_sync2	ethmac.wishbone.Busy_IRQ_s...	ethmac.wishbone.Busy_I...	work.eth_wishbone	
16	ethmac.wishbone.RxAbortSync2	ethmac.wishbone.RxAbortSy...	ethmac.wishbone.RxAbor...	work.eth_wishbone	
17	ethmac.wishbone.LatchedRxStartFrm	ethmac.wishbone.SyncRxStar...	ethmac.wishbone.SyncRx...	work.eth_wishbone	
18	ethmac.wishbone.WriteRxDataToFifo	ethmac.wishbone.WriteRxDat...	ethmac.wishbone.WriteR...	work.eth_wishbone	
19	ethmac.wishbone.WriteRxDataToFifo		ethmac.wishbone.WriteR...	work.eth_wishbone	

ethmac.wishbone.LatchedRxStartFrm

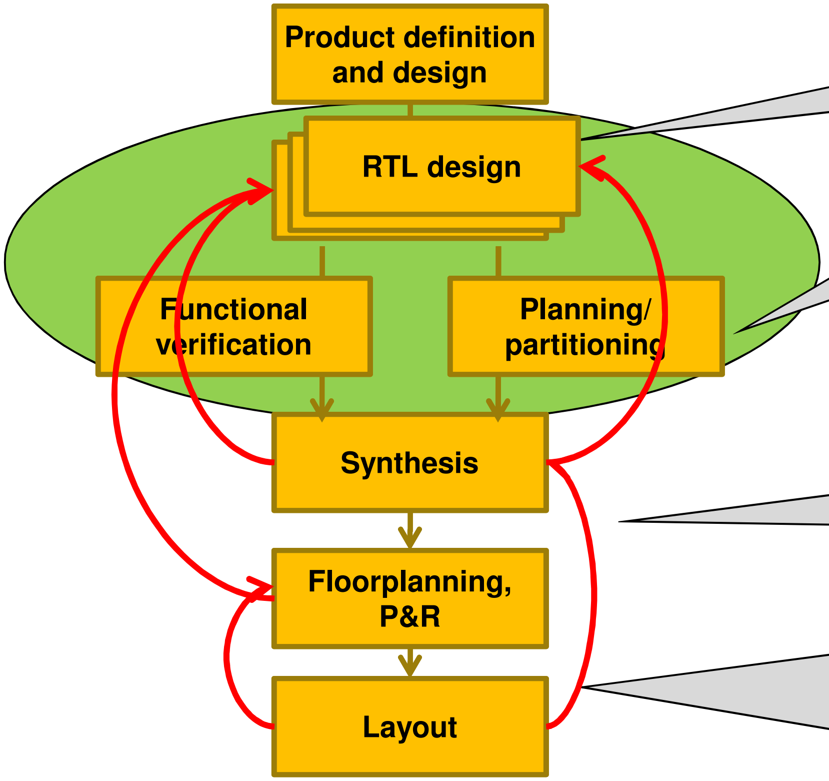
Enable Systematic Closure

Speeds up CDC signoff by 3X

The Constraints Problem

REAL INTENT

Accelerating RTL Sign-off



- Creating synthesis constraints takes a lot of effort

- Constraints need to be managed across design partitions

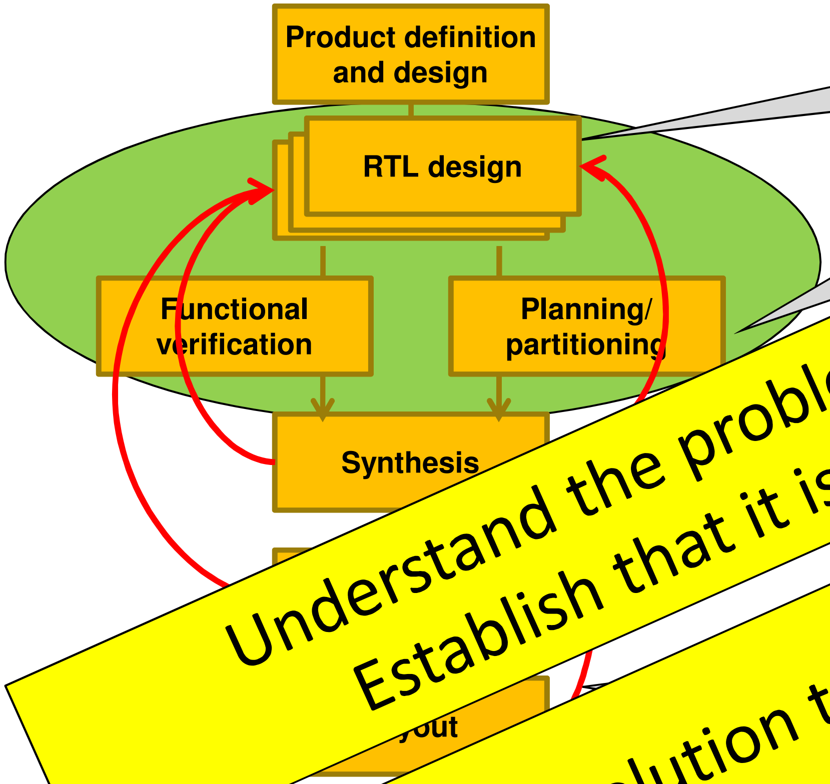
- ***Constraints need to be consistent across design flow***

- Post-budgeting, partitioned SDC misses critical constraints - requires synthesis and P&R iterations
- ***Exceptions must be correct to ensure accurate sign-off analysis***

The result – iterations and timing closure delays

The Constraints Problem

REAL INTENT
Accelerating RTL Sign-off



- Creating synthesis constraints takes a lot of effort

- Constrai
acr

raints
consis

, partitioned SDC
ritical constraints - quires synthesis and P&R iterations
- ***Exceptions must be correct to ensure accurate sign-off analysis***

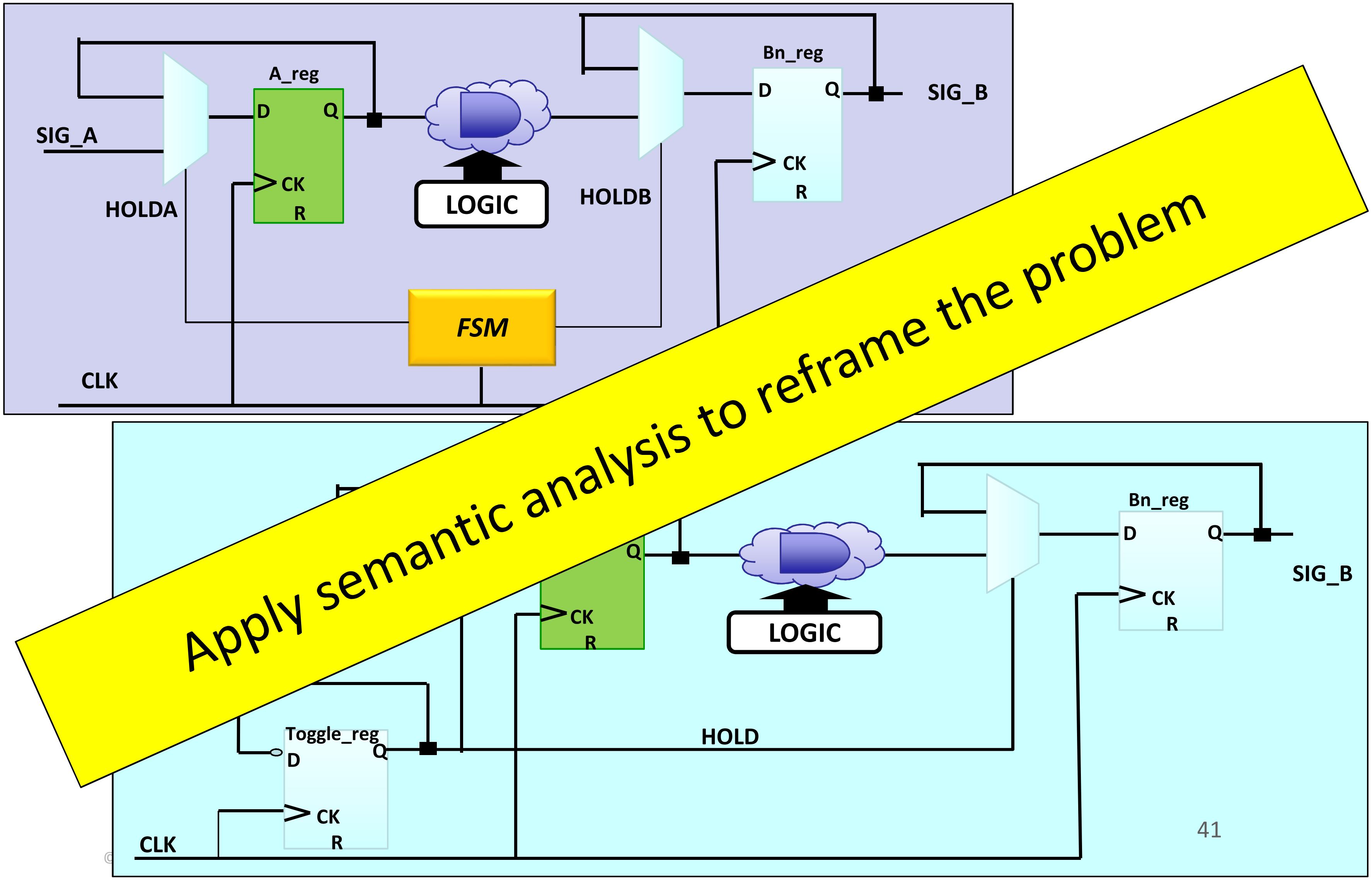
Understand the problem at fundamental level

Establish that it is a high-value problem

Then, set up solution to Frame, Scope and Analyze

The result – iterations and timing closure delays

Functional Analysis of Exceptions

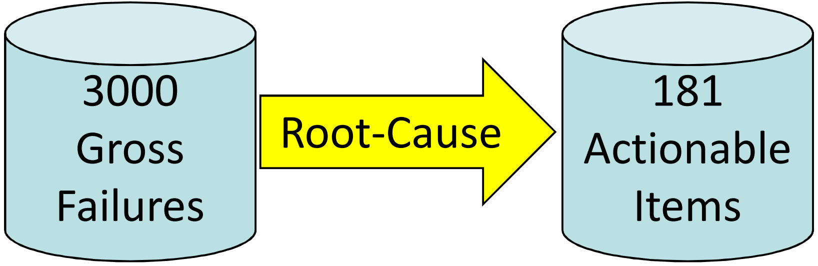


Apply semantic analysis to reframe the problem

Many Other Applications of the New-Paradigm Template

REAL INTENT
Accelerating RTL Sign-off

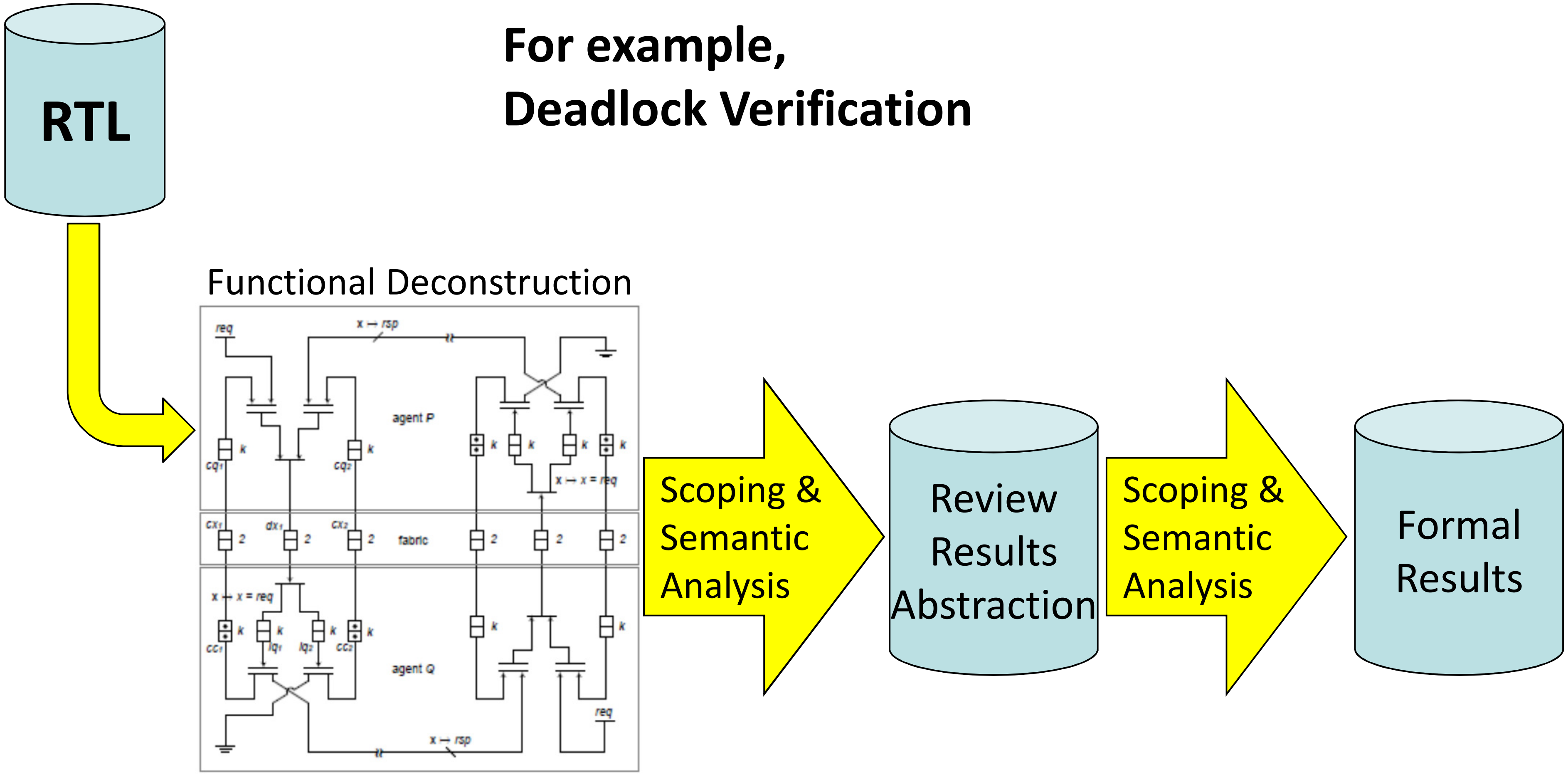
- Reset-Safety
  - Metastability & Correlation-loss based failure modes
- Auto-Formal
  - RTL functional implementation bugs
  - Challenge: Identify actionable failures quickly
  - Very high volume of implicit checks: Throughput vs. Depth
  - Root-cause analysis is key



- Code Quality (Lint)
- A Few More …

Frame, Scope and Analyze Other Problems!

For example, Deadlock Verification


RTL
Functional Deconstruction
req
x → rsp
agent P
k
cq₁
cq₂
x → x = req
cx₁
dx₁
cx₂
2
fabric
x → x = req
cc₁
lq₁
lq₂
cc₂
agent Q
req
x → rsp
Scoping & Semantic Analysis
Review Results Abstraction
Scoping & Semantic Analysis
Formal Results