2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
DECEMBER 6 - 7, 2022

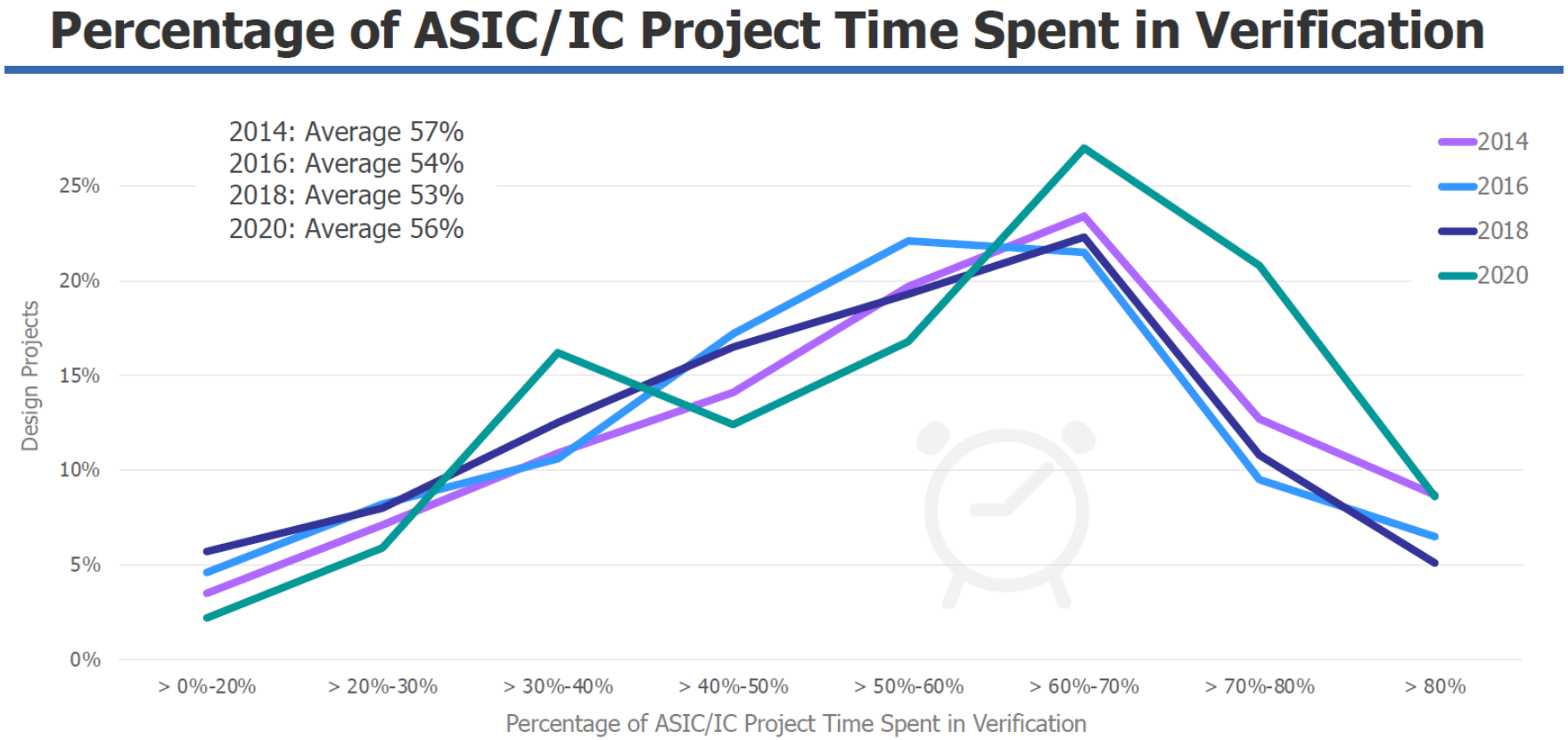
Static Sign-Off Best Practices
Learnings and Experiences from Industry Use Cases

Vikas Sachdeva, Senior Director of Product Strategy and Worldwide Business Development

REAL INTENT

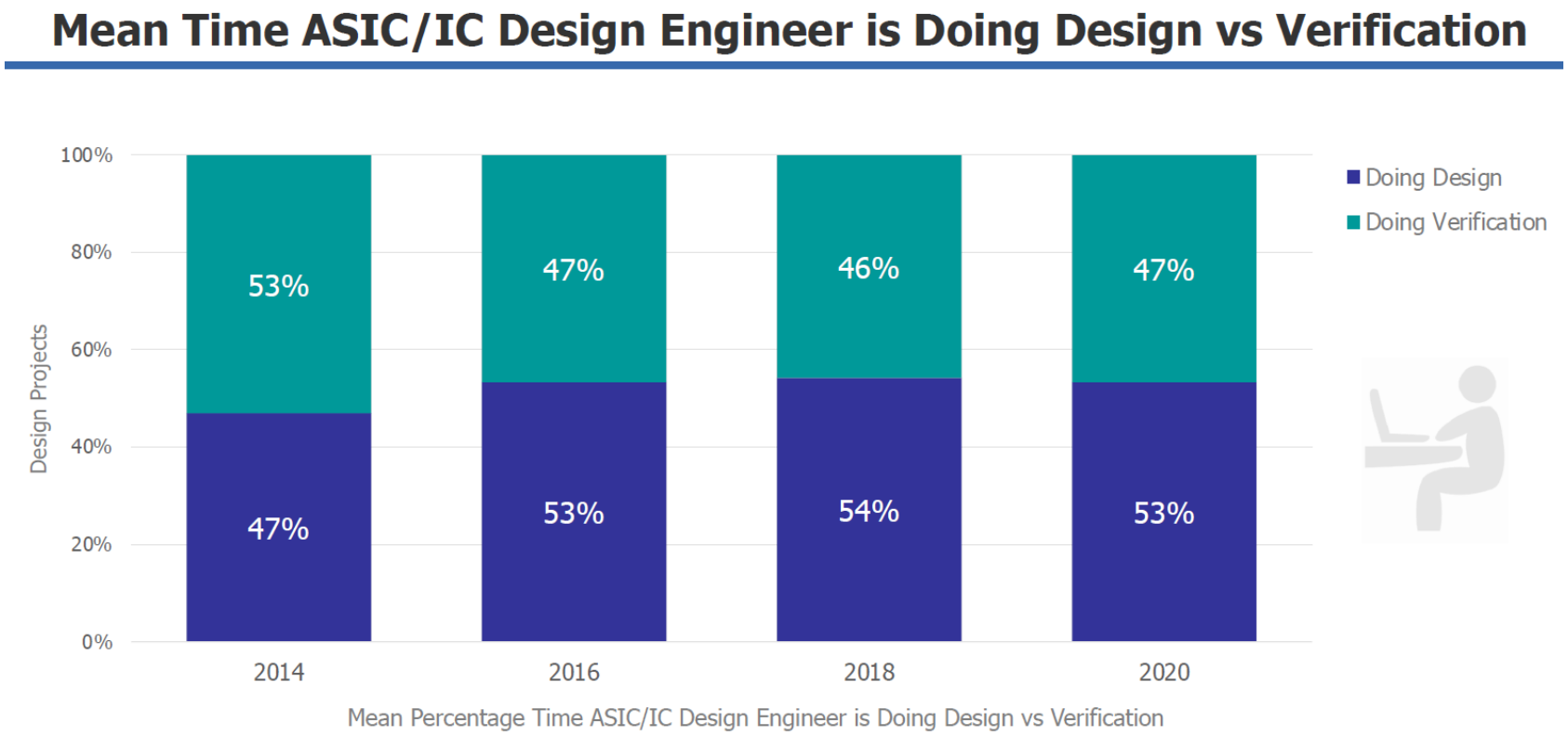
accellera SYSTEMS INITIATIVE

The Verification Challenge



Source: Wilson Research Group and Mentor, A Siemens Business, 2020 Functional Verification Study

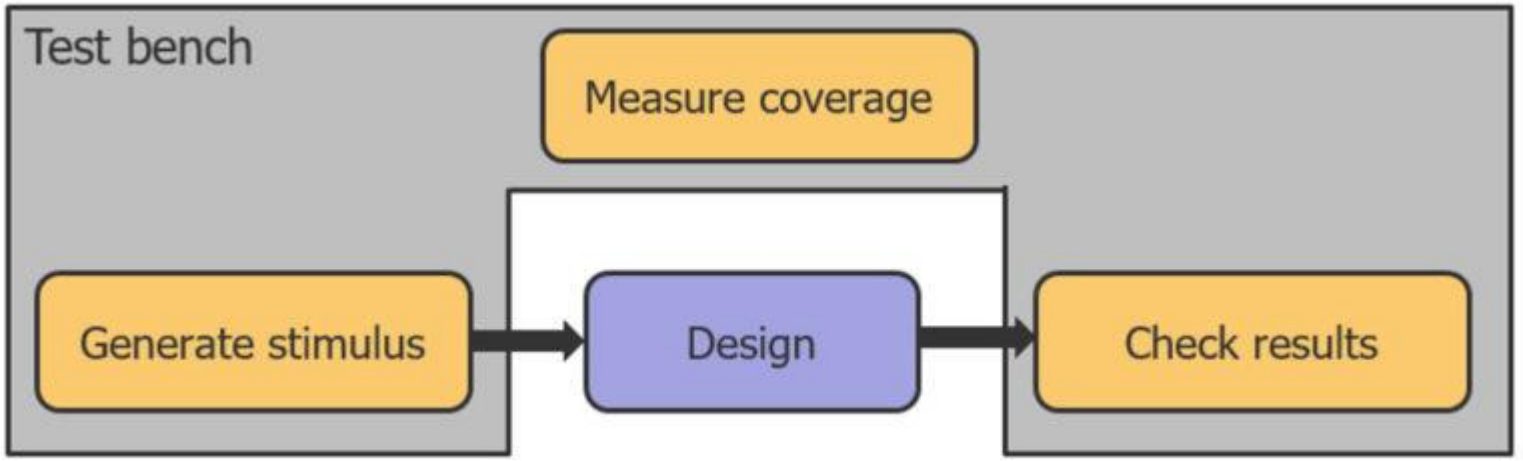
The Verification Challenge



Source: Wilson Research Group and Mentor, A Siemens Business, 2020 Functional Verification Study

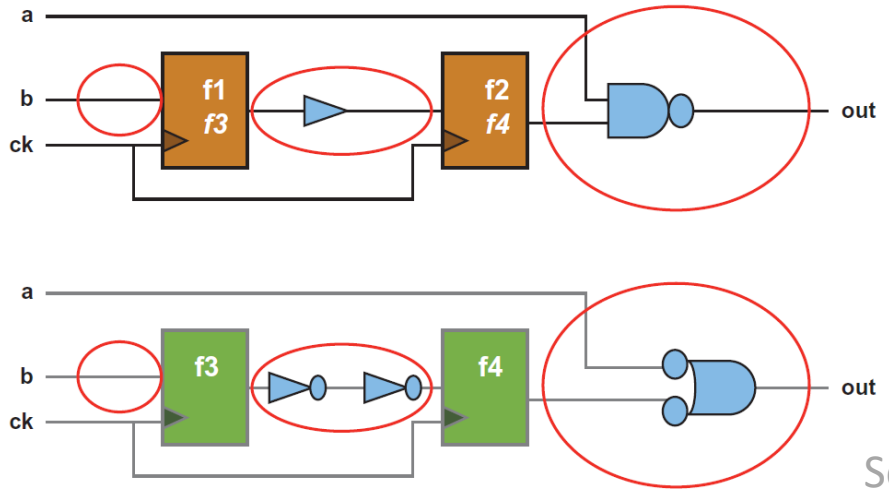
Verification Methodologies – Dynamic Verification

- Compute design behavior for user specified testcases
- Check the computed behavior for failures
- Examples: Simulation and Emulation



Verification Methodologies – Formal Verification

- Uses the tools to mathematically analyze the space of possible behavior of a design
- Example: Equivalence checking, Formal Property Verification

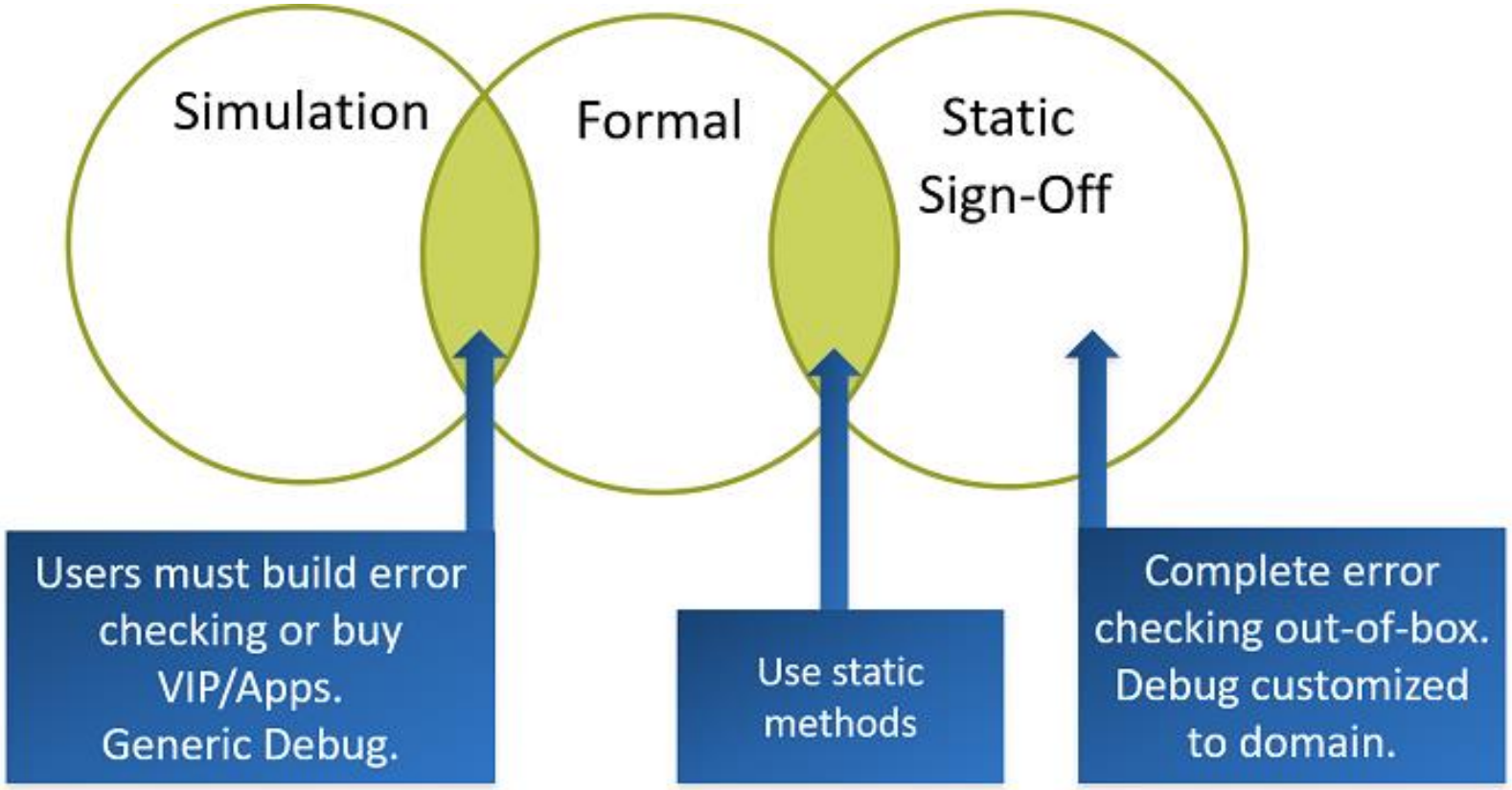


Source: Formal Verification – An essential Toolkit

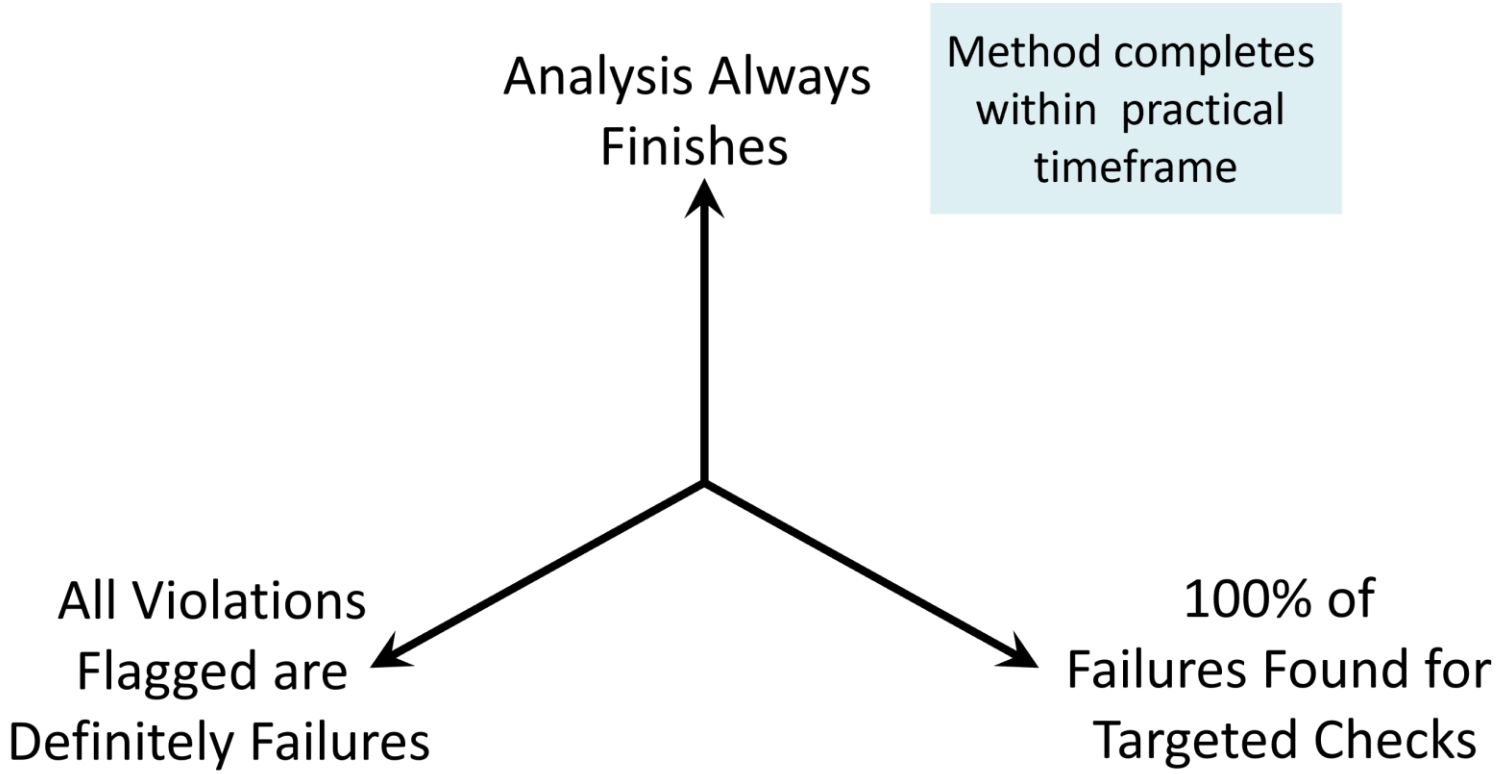
Verification Methodologies – Static Verification

- Utilizes search and analysis techniques
- Checks for design failures under all possible testcases
- Examples: STA, Lint, CDC etc.

Comparing Simulation, Formal & Static Sign-Off



3 Verification Methodology Metrics



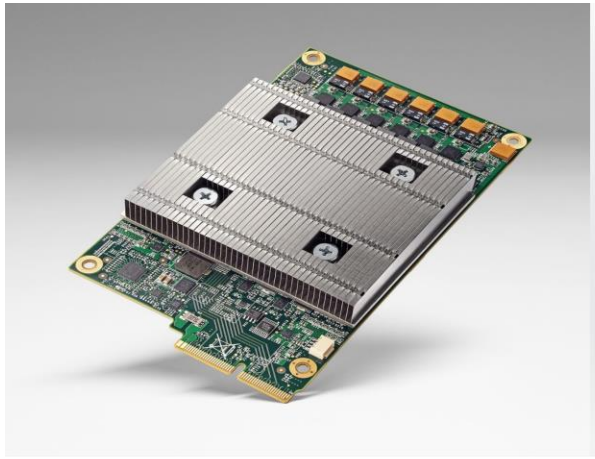
Comparing Simulation, Formal & Static Sign-Off – On verification metrics

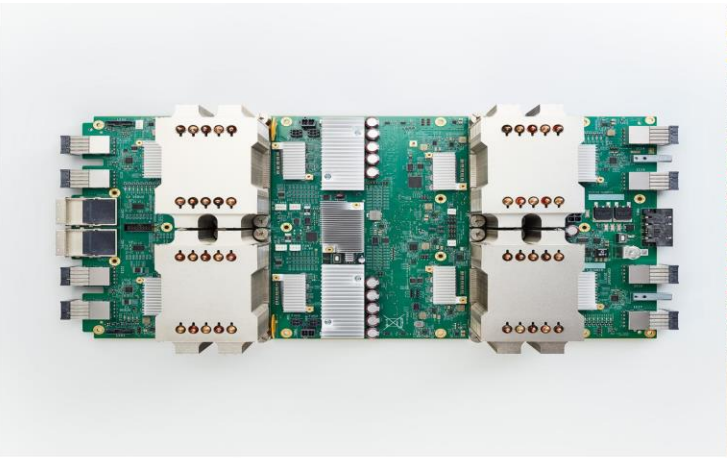
Verification Metric	Simulation	Formal	Static Sign-off
Analysis always finishes	Yes	No	Yes
100% of failures found for target checks	No	Yes	Yes
All violations flagged are definite failures	Yes	Yes	No

Learnings and Experiences from Google's Cloud Based Sign-Off Methodology

TPU – Custom Hardware for Machine Learning

Source: https://www.realintent.com/google-static-sign-off-methodology-results/





Enabling businesses to manage more data,
Even with Moore's Law over

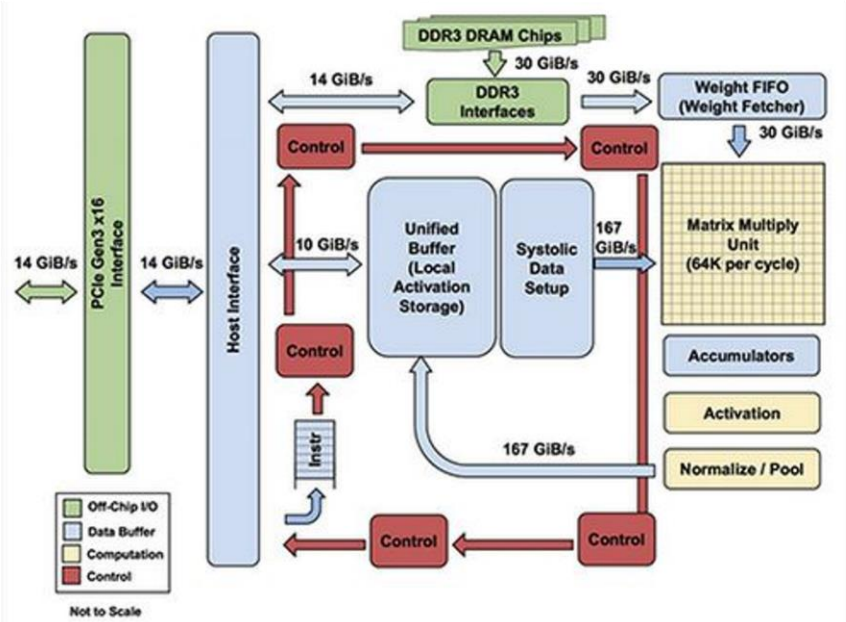
Google ASIC Methodology – Main Challenges

Speed

- Very fast design cycle for ASIC and system
- Rapid bring up and deployment

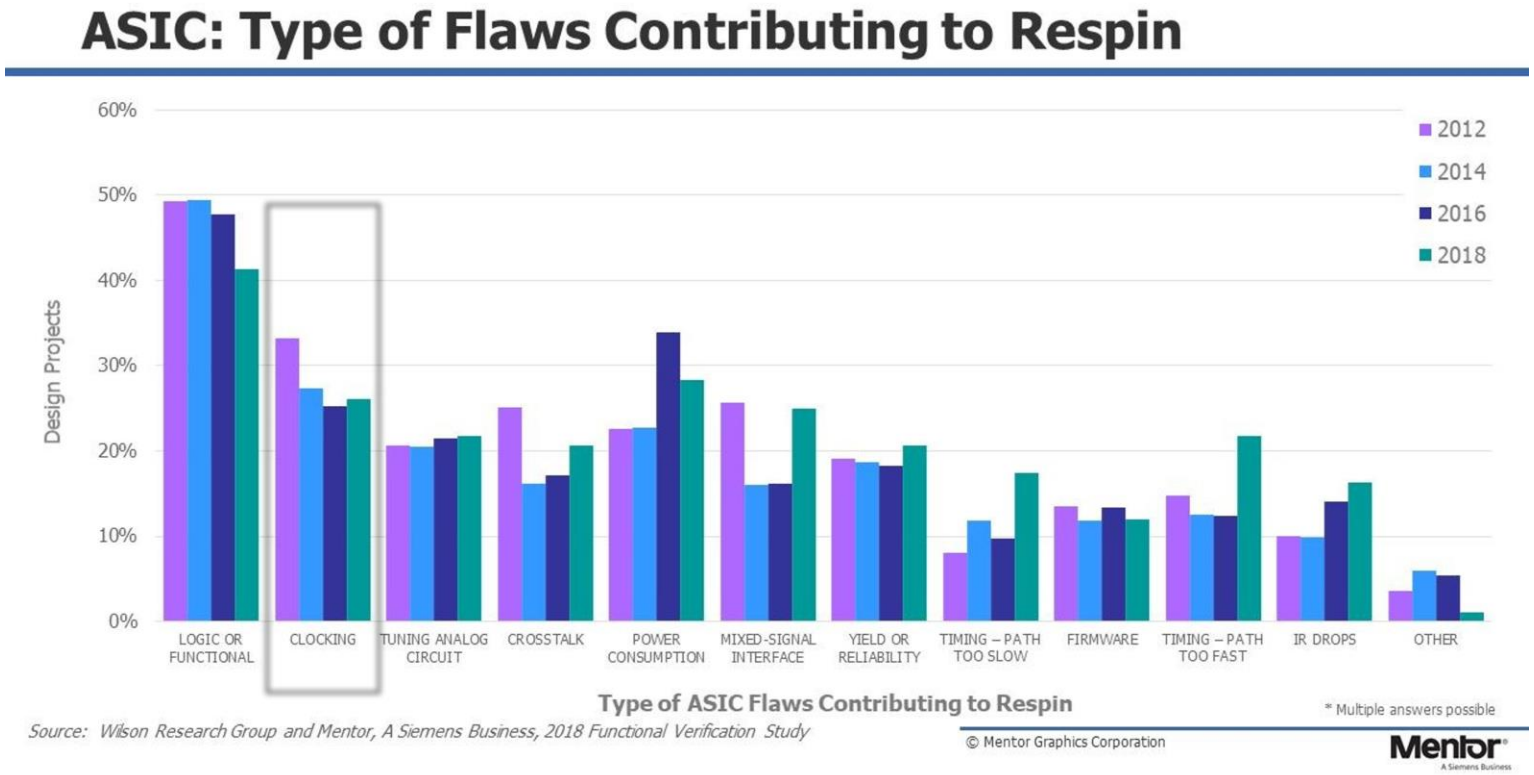
Capacity

- Very complex ASICs
- Power, performance, and other tradeoffs

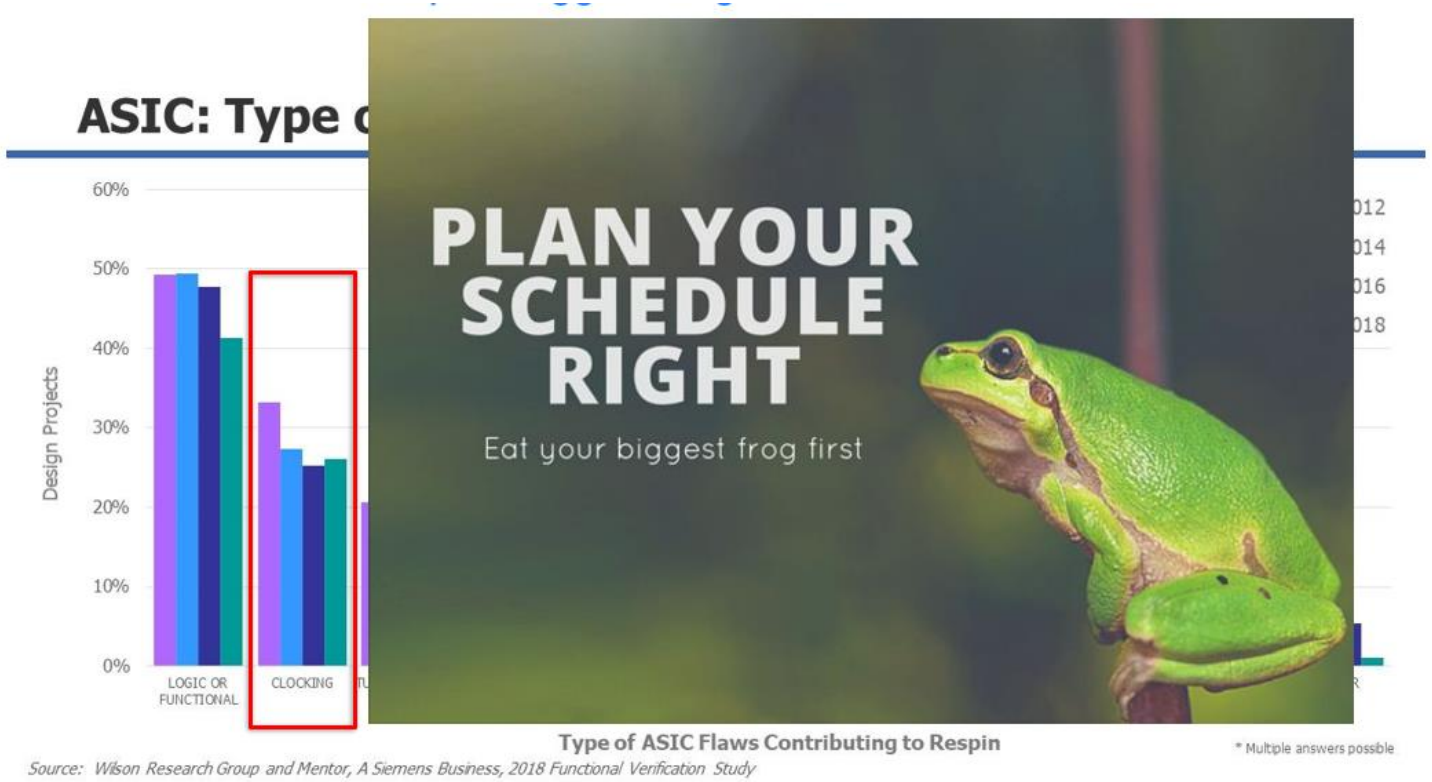


Source: https://www.realintent.com/google-static-sign-off-methodology-results/

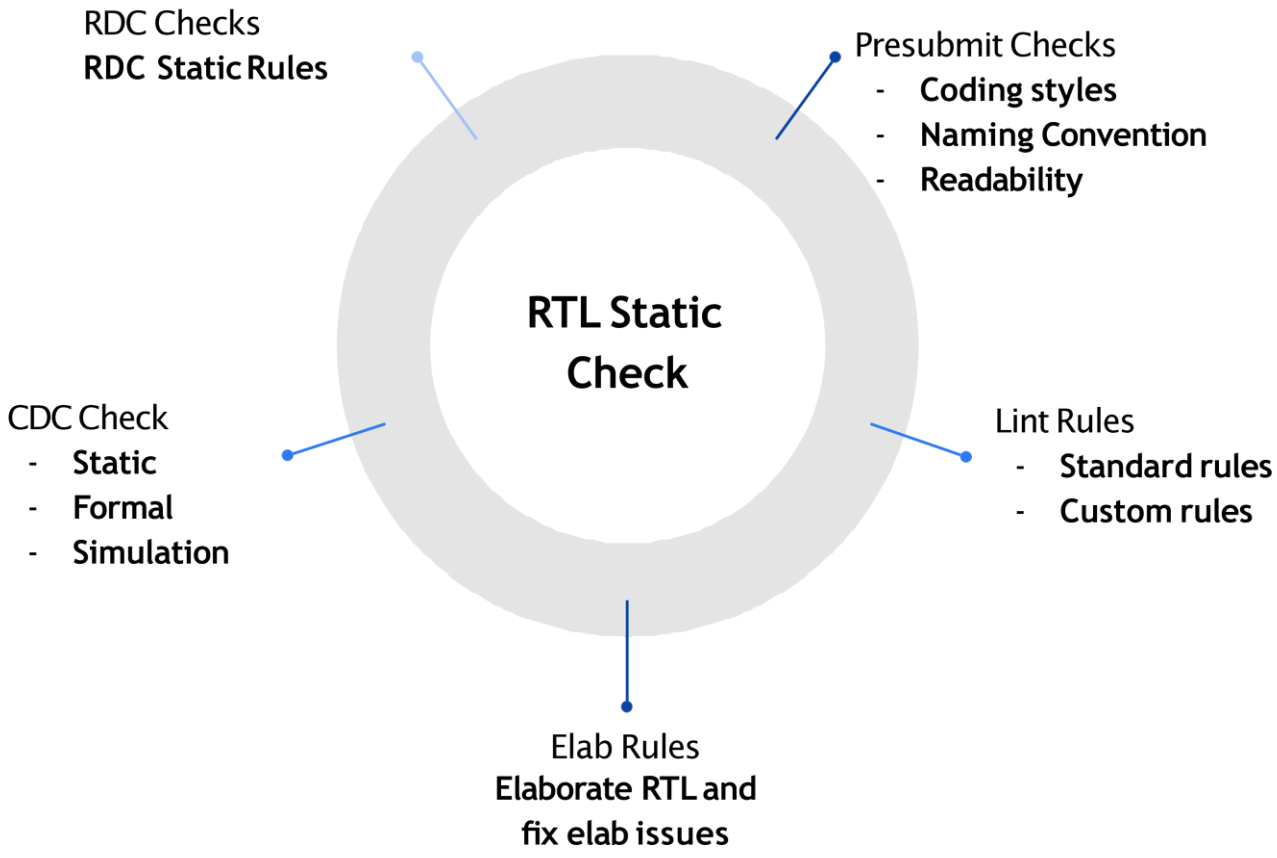
Clocking is the Primary Challenge in ASIC Design



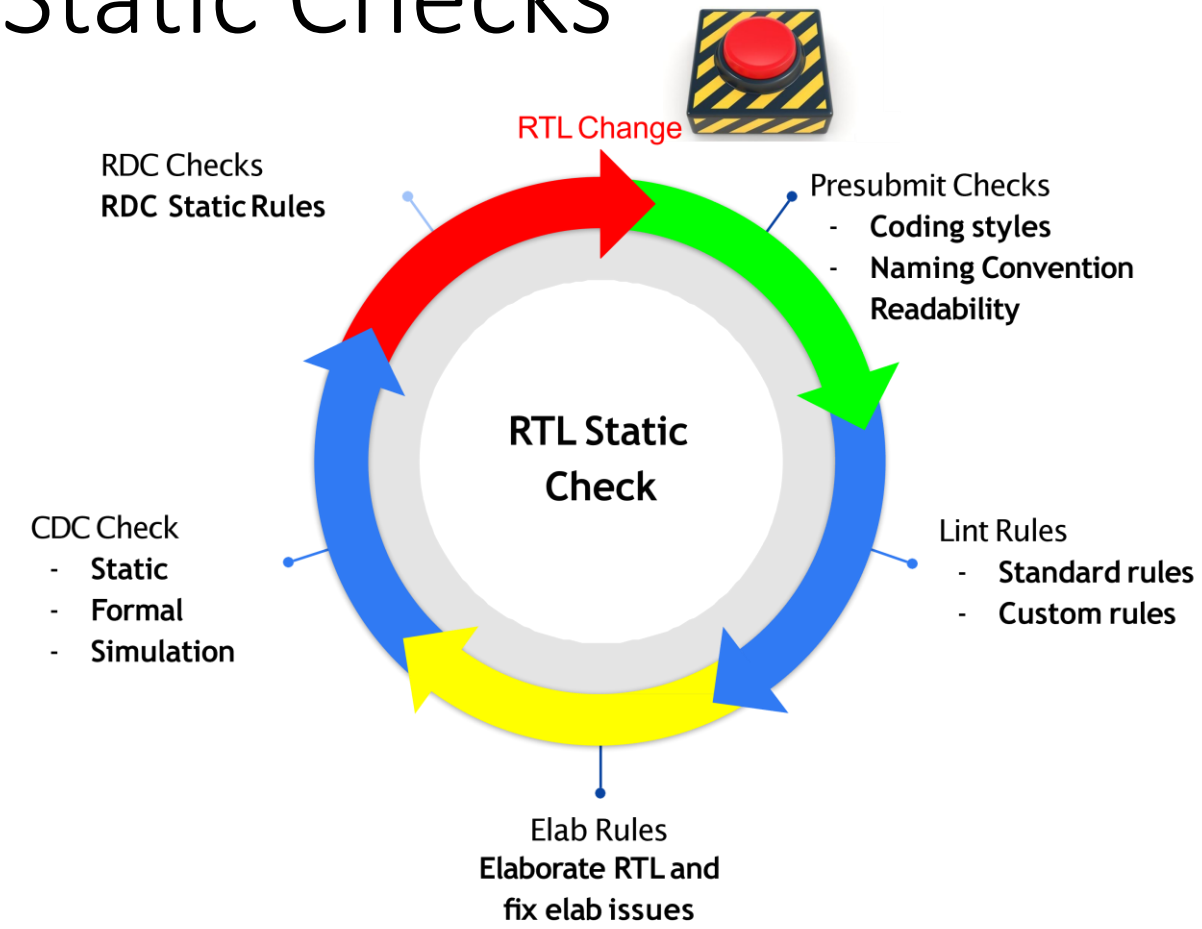
Google Static Sign-Off Best Practices – #1 Static Checks First



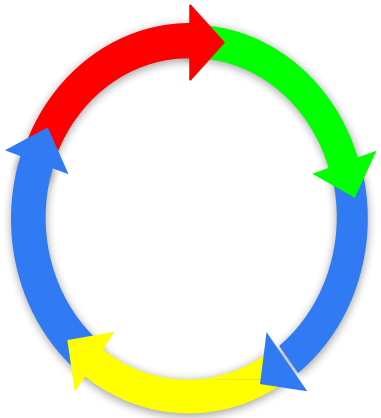
Google Static Sign-Off Best Practices - #2 Breadth of Static Sign-Off Checks



Google Static Sign-Off Best Practices - #3 Continuous Static Checks



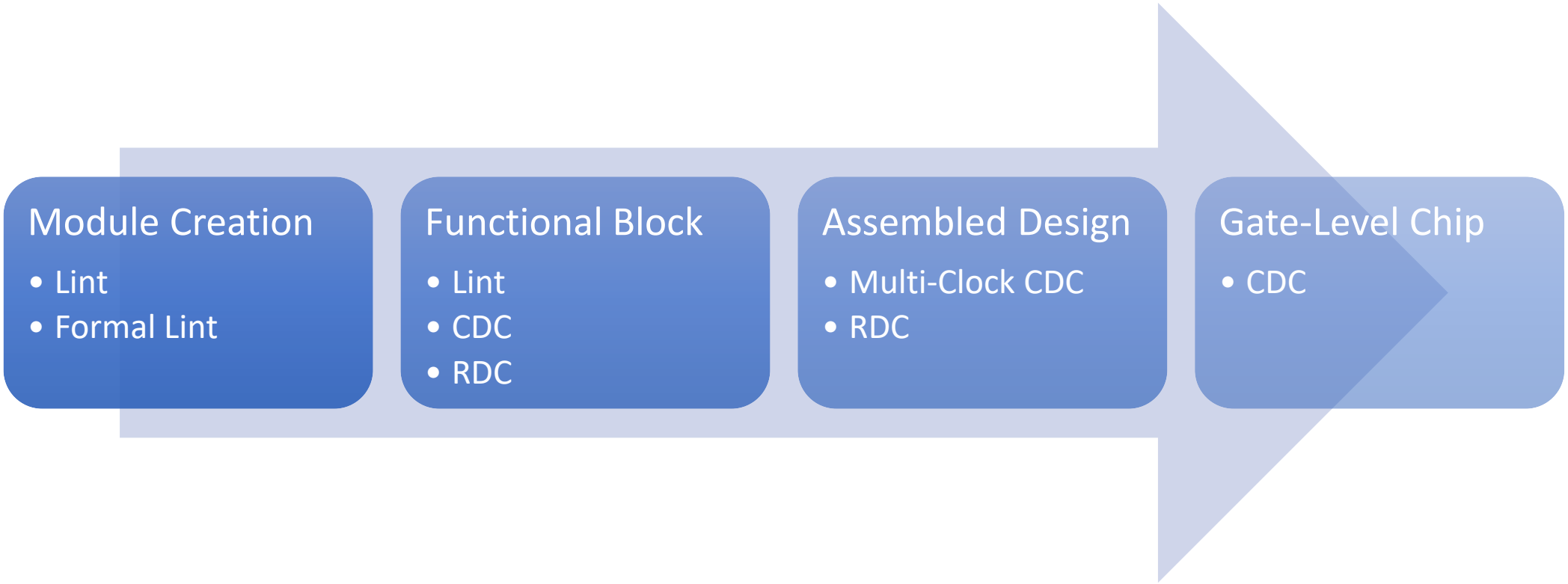
Impact of Google's Static Sign-Off Methodology



- More bugs were found by continuous approach
  - Nightly static runs
  - Automatic dashboard updates
  - Automatic bug filing
- Reduced late-stage RTL changes
  - Higher quality RTL reduces risk of expensive iterations
  - Saves ECO efforts
- Better Schedule Control
  - Reduced violation noise as the most pressing signoff challenge

Learnings and Experiences from Nvidia's Sign-Off Methodology

Static Signoff Tools: When and Where



Problems Solved – Catching Problems Others May Miss, Earlier

Lint	CDC	RDC
• Find RTL problems before simulation or synthesis • Find bugs not found otherwise	• Catch tricky problems that escape best efforts of design guidelines, golden IP, and simulations	• Catch another class of subtle problems that could cause perplexing silicon bugs

Successes – Anecdotes From The Trenches

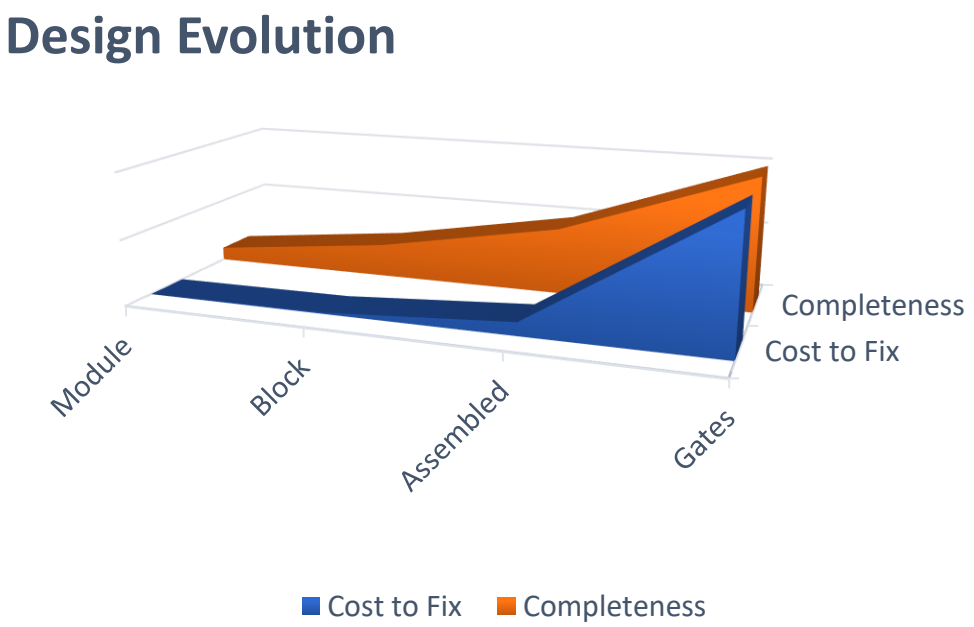
- Using Lint means we don’t find RTL problems at synthesis or equivalence checking.
  - Simple: dangling inputs, multi-driven nets
  - Corner Cases: parallel case, bounds check, arithmetic overflow checks
  - Subtle: Self-Determined Expressions
- CDC
  - Block RTL CDC helps guarantee safe interface design.
  - CDC after Assembly helps catch that inter-block pipelining registers were inserted on the correct clock domain.
  - Full-Chip Gate CDC is a final check, including DFT, ECOs, etc.

Best Practices – Automation and Enforcement

- Static Sign-Off is not optional!
- Provide push-button flows to create environment, run tool, analyze report, apply waivers.
- Automatically determine Pass/Fail status, post to dashboard.
- Run the tools regularly at prescribed points: at check-in, regressions, project milestones.
- Static Sign-Off is NOT a designer running the tool in a GUI and telling the chip manager it passed. Needs rigor in tool application and result tracking.

Best Practices – Very Early and Late

- Finding problems earlier dramatically lowers the cost to fix them.
- But, design completeness and correctness evolve over time.
- Also, the design environment (e.g., constraints) evolves over time.
- There's no one best time to run SSO. Need to run at every stage.



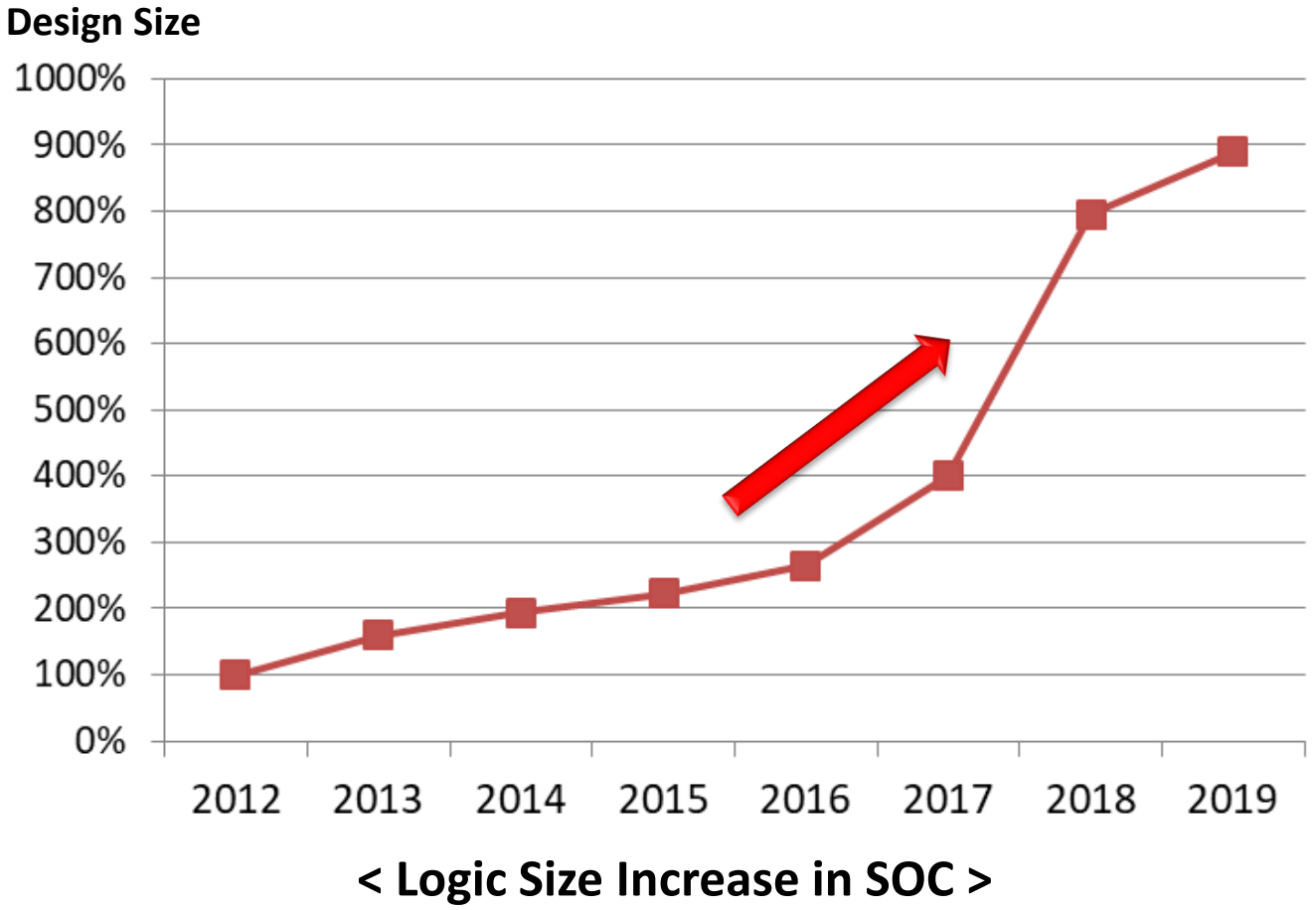
Key Static Signoff Application Capabilities

Coverage and Correctness.	• This is the reason the tool exists. • Does it cover everything that it can? • Is it conservative (not optimistic)?
Very High SNR	• Noise is the bane of static analysis! • Minimal false violations, compact reports, root cause reporting.
Programmability and Debuggability	• Key to “automation and enforcement”. • Implementation tools set a good example: Tcl, design object access, useful attributes, etc.

Learnings and Experiences from Samsung's Sign-Off Methodology

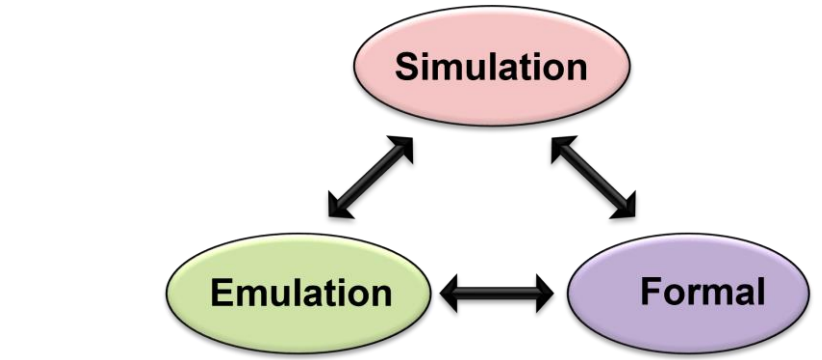
Where are We with Design Complexity

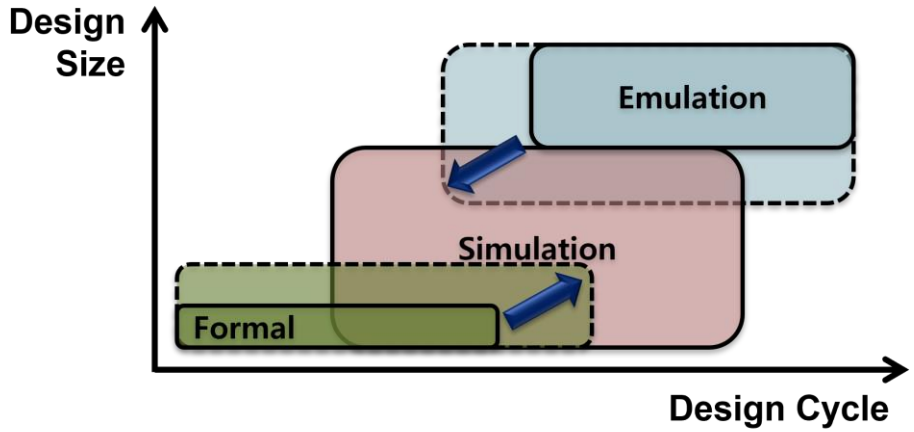
- 30% design size increase on average YoY
- More IPs are integrated in SOC
- Design cycle is shrinking



How are We Doing on Functional Verification

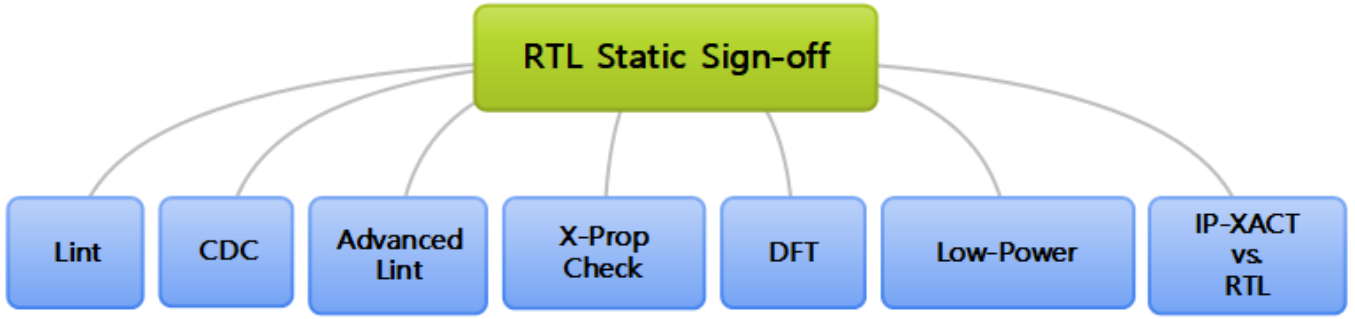
- Deep bug-hunting by strengthening IP level formal verification
- Simulation with multi-cores
- Simulation acceleration using Emulation
- Early SW development using Hybrid Emulation





How About Static Verification and RTL Sign-Off

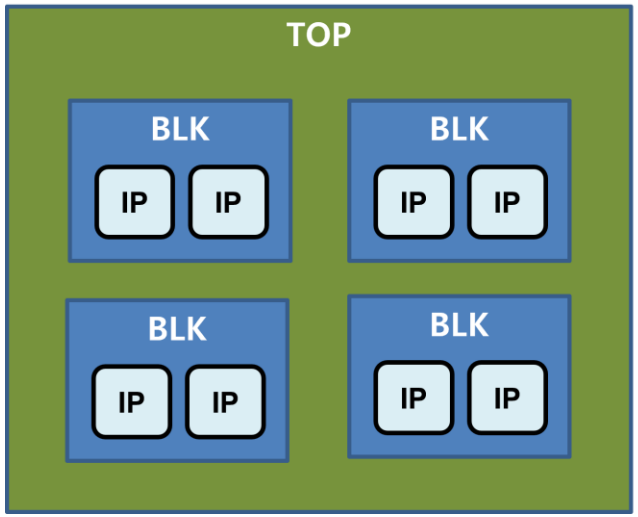
- Non-manageable design size with current static tools
  - 30~40 hours runtime & 1TB memory footprint at SOC level CDC → NOT practical!
- Excessive review & debug time/resources
  - 500k CDC paths to review → 90 man-weeks
  - Wasting effort to review too many false negatives



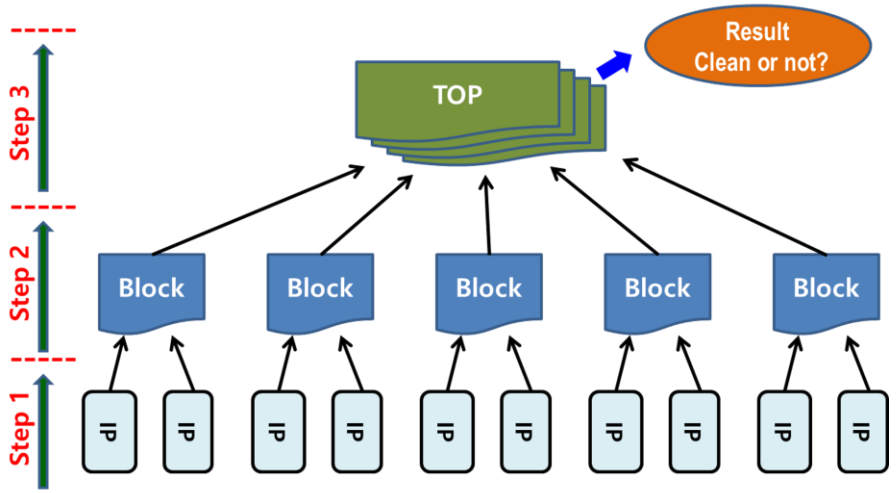
Hierarchical CDC for SOC

- Flat analysis vs Hierarchical analysis
  - Abstracting IP or block level information as “metadata”
  - Using lower level metadata for upper level CDC analysis

< Flat CDC Analysis >

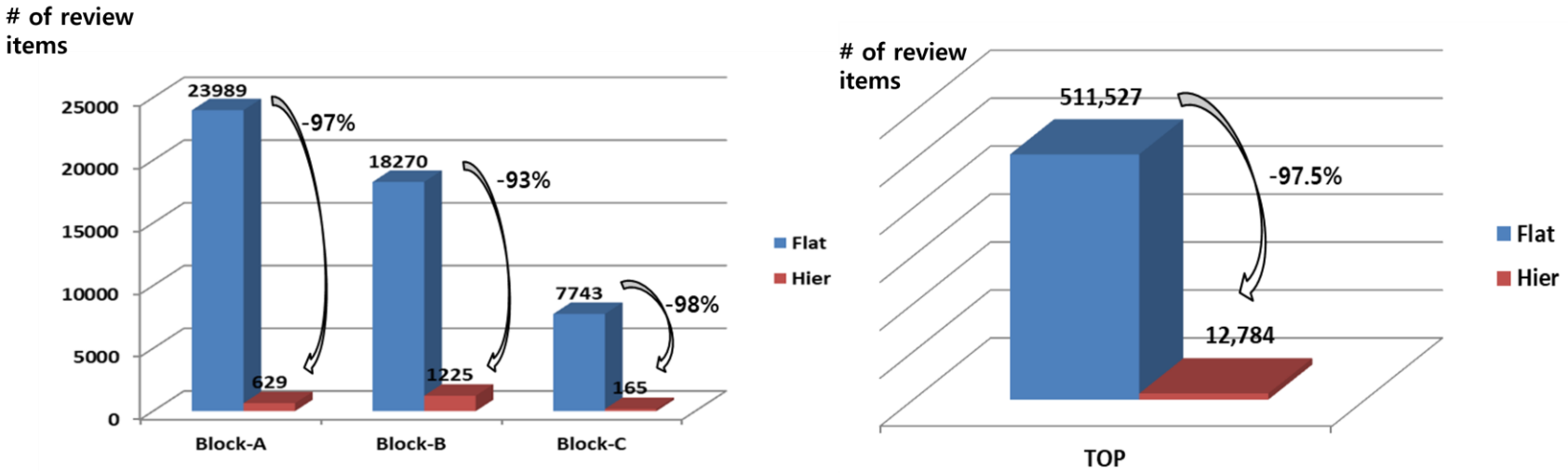


< Hierarchical CDC Analysis >



Hierarchical CDC for SOC

- Abstracting IP information for block level CDC
- Abstracting block information for SOC level CDC
- Reduction for runtime (30h→3h) and review man-hours (100%→30%)



We Still Have Challenges!

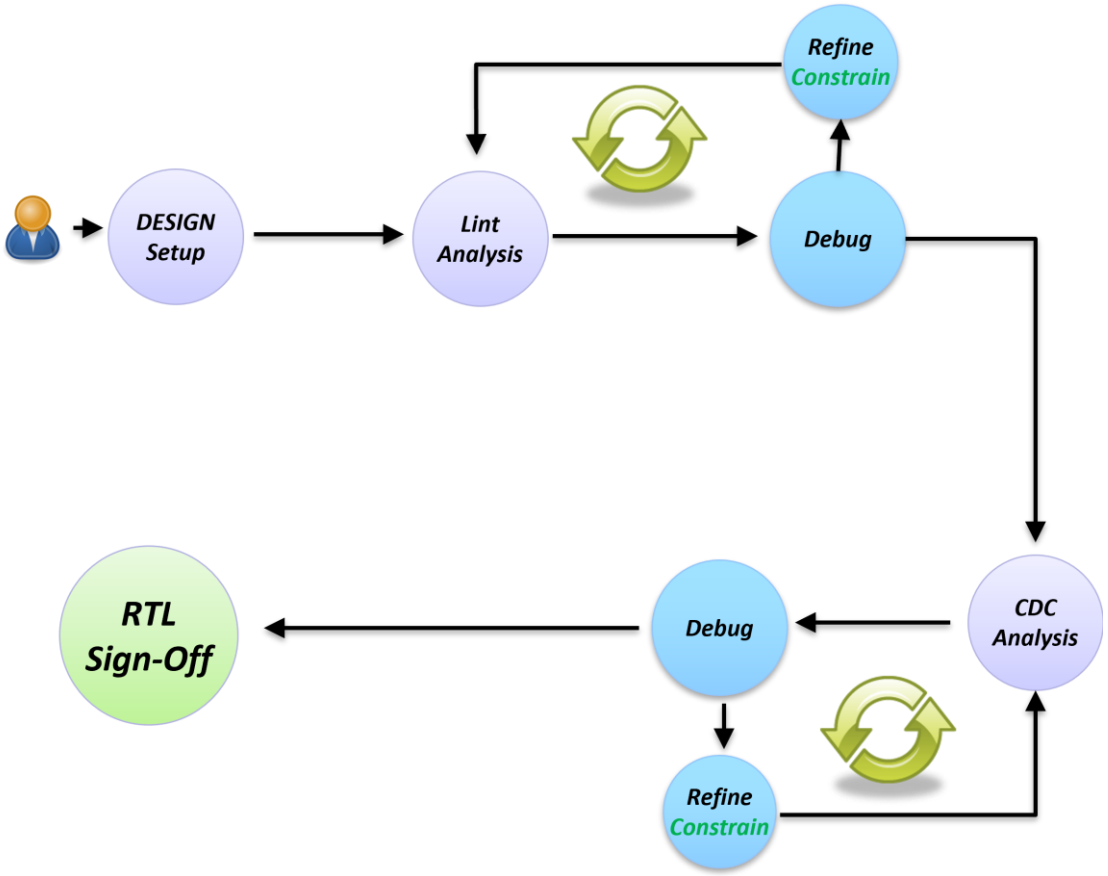
- Stiff learning curve for formal verification
  - Industry Standard Formal Verification Methodology (such as UVM) may be required
  - Better support for resolving inconclusive assertion is wanted
- Still suffering from excessive debug time & effort
  - Smart technology to reduce false negative (99% in CDC review) is wanted
  - Can we leverage Machine Learning?
- Insufficient tool capacity for multi-billion gate SOCs
  - Can we apply Divide-n-Conquer to all static verification?
  - Hierarchical formal or Emulation for formal?

- We believe we have a lot to improve!!!

Hailo’s Static Signoff Methodology for Edge AI Processor

Hailo's RTL Static Sign-Off flow

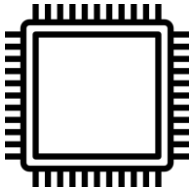
- Bottom-up flow used for full chip runs
- Lint is run at block & full chip level
- CDC static sign-off done at block & full-chip level

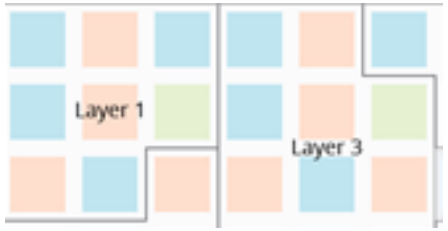


Static Signoff Challenges for Hailo's Edge AI Processor

- Pressured time-frame for RTL freeze
- Had to sign-off in most efficient manner
- Unfamiliarity with the tool at project start
- Complex clock structure & knowledge was scattered

Handle Challenges with Efficient Static Tools





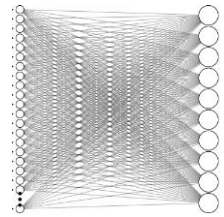
Design Challenge

Huge amount of compute elements

High Locality

Static Sign-Off Tool Impact

Fast performance



Many repeating identical components

More layers => solves complex problems

Eliminates duplicate violations, reduces noise

Highly efficient shelling flow

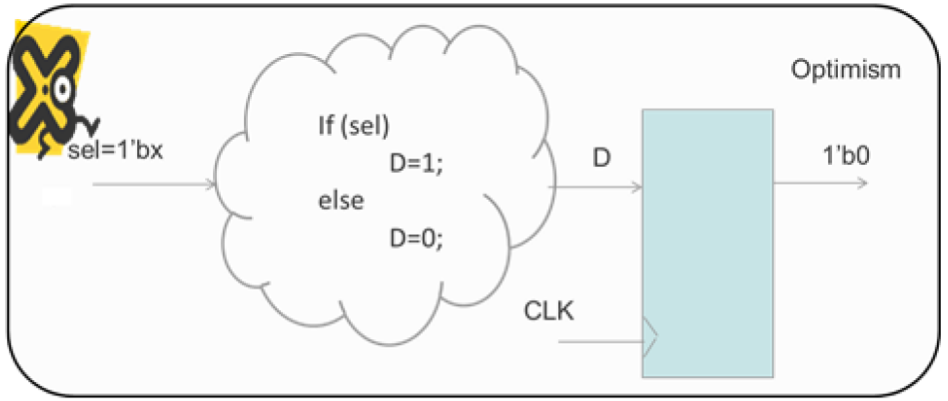
Scales well with complexity

Paulo Alto Networks – Advanced X-Propagation Methodology to identify X-initialization source errors

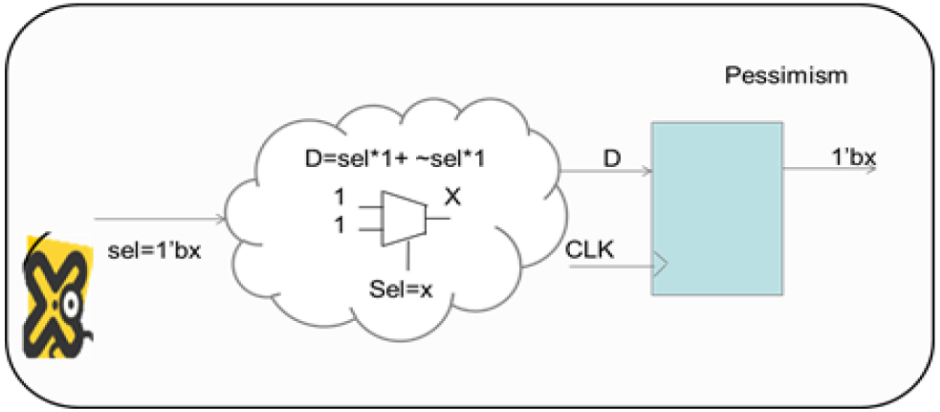
Initial Methodology: Analysis During Simulation Only

- Earlier methodology only involved X-propagation analysis during simulation
- Risked missing issues as dynamic analysis limited by test patterns
- Coverage limited by the test runs

X-Optimistic Behavior



X-Pessimistic Behavior



Methodology Advancement: Adding X-Propagation Static Sign-off

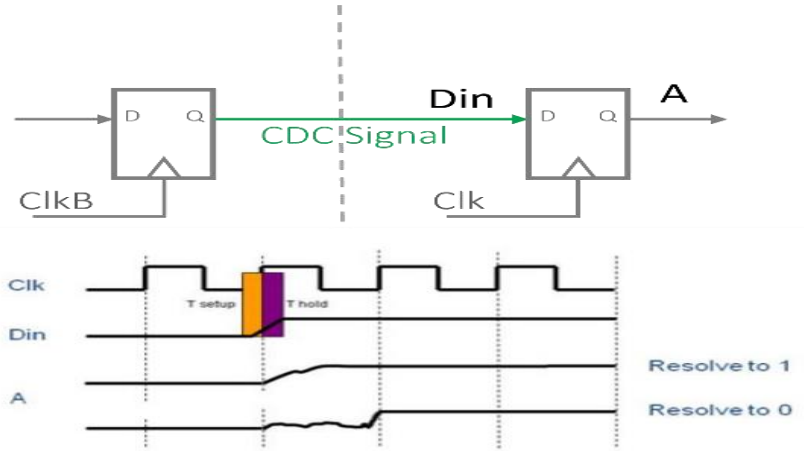
Exhaustive and High Performance

Module	Size (gates)	REAL INTENT Meridian RXV Runtime
module A	5 M	5 min
module B	30 M	60 min
module C	7 M	2 min
module D	5 M	10 min
module E	4 M	8 min
module F	3 M	3 min

Samsung – Using the right mix of static and dynamic verification for CDC Sign-Off

CDC Metastability

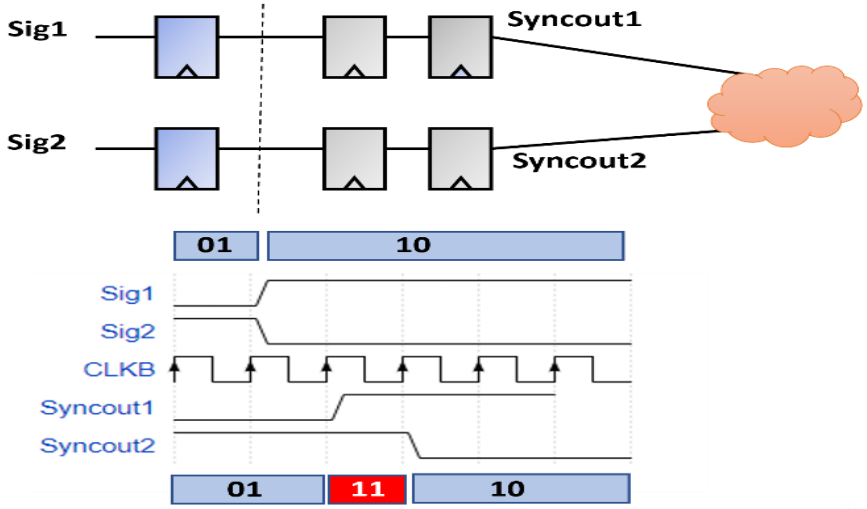
- Metastability on CDC paths can lead to
  - Unpredictable results
  - Unpredictable delays
- Synchronizers are used to squelch metastability, but ...



Metastability

Correlation Loss

- *Converging synchronizers cause correlation loss*
- CDC Structural verification does report reconvergences and other design problems but …
- *Structural CDC analysis is not enough for:*
  - Identifying functional impact of reconvergences
  - Validating the correctness of synchronizations under metastability stress in synchronizers
  - Detecting problems due to signal skews on synchronizer paths



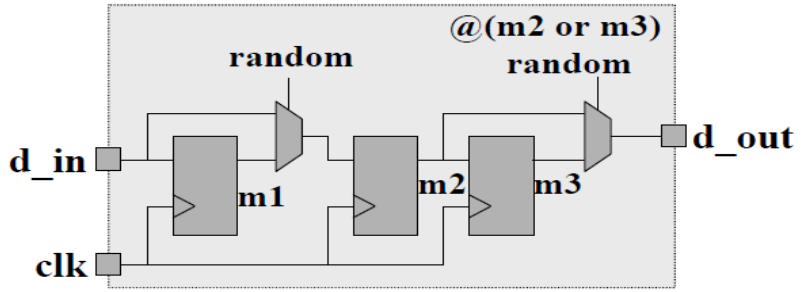
Correlation loss because of reconvergence in CDC

Dynamic CDC & Traditional In-House Jitter Models

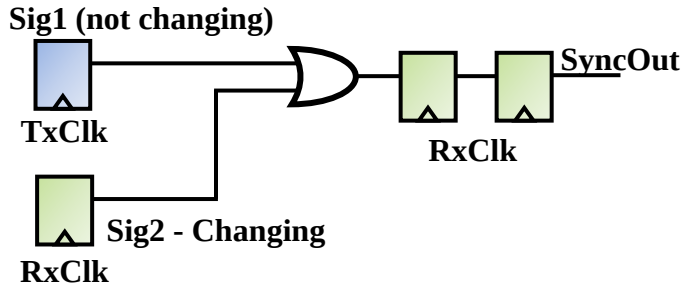
- In Dynamic CDC, reconvergences and other specific CDC problems are checked during functional simulation
- Historically handled by in-house metastability injection model for synchronizers, but …
- *Traditional in-house metastability injection models are not accurate or may cause false injections*

In-House Models Have Shortcomings

- In-house models have several shortcomings
  - Inject metastability even when only sync drivers changed
  - Inject metastability incorrectly even when pulse is wide enough
  - Do not catch metastability due to short-pulse or combo-glitch
  - Handle clock-gating inadequately

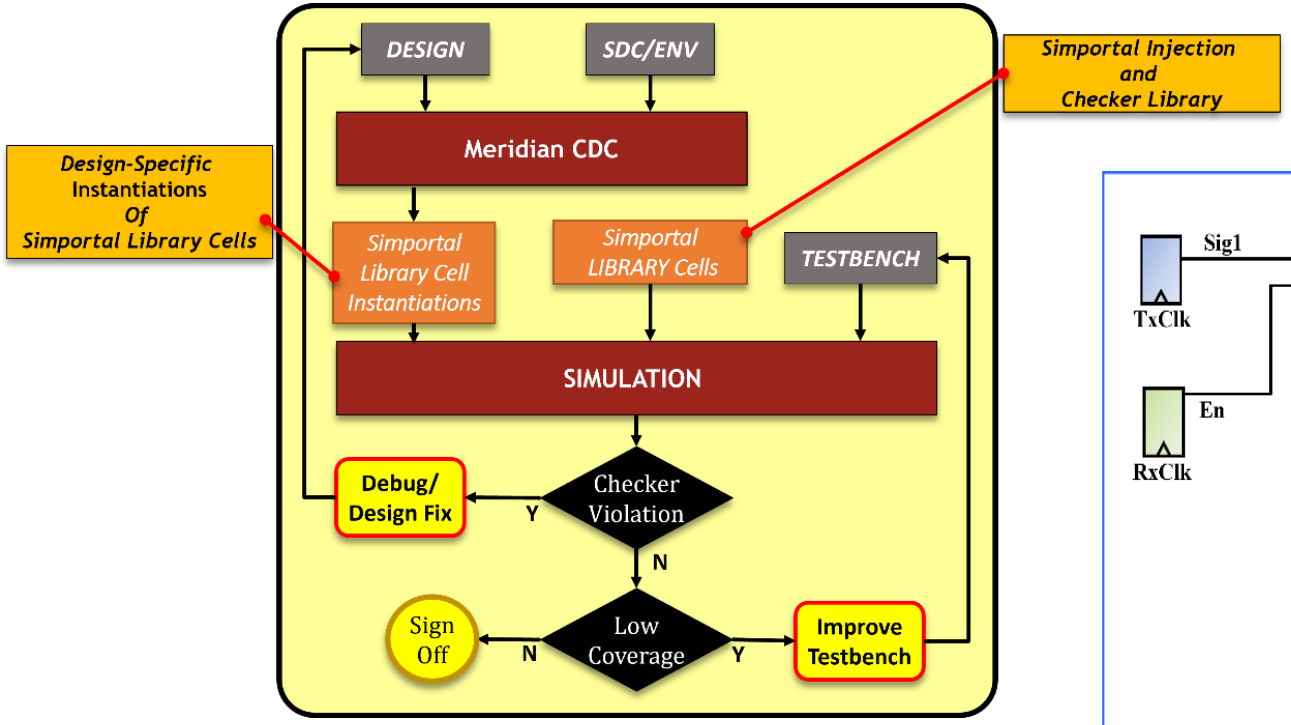


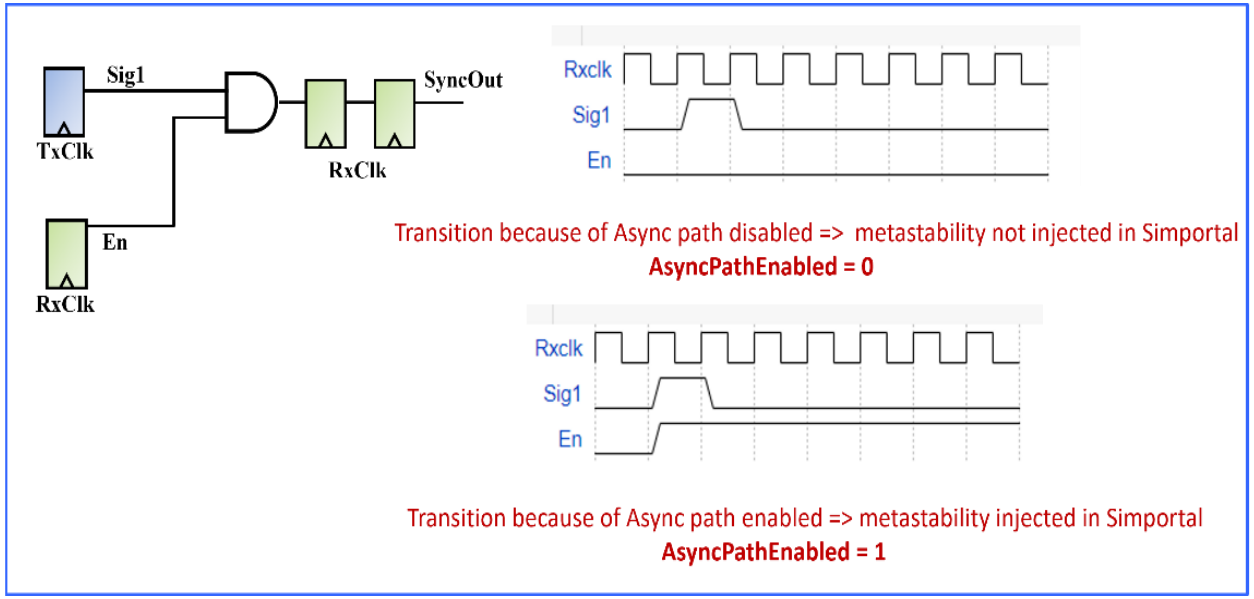
Typical Inhouse Metastability-Injection Model



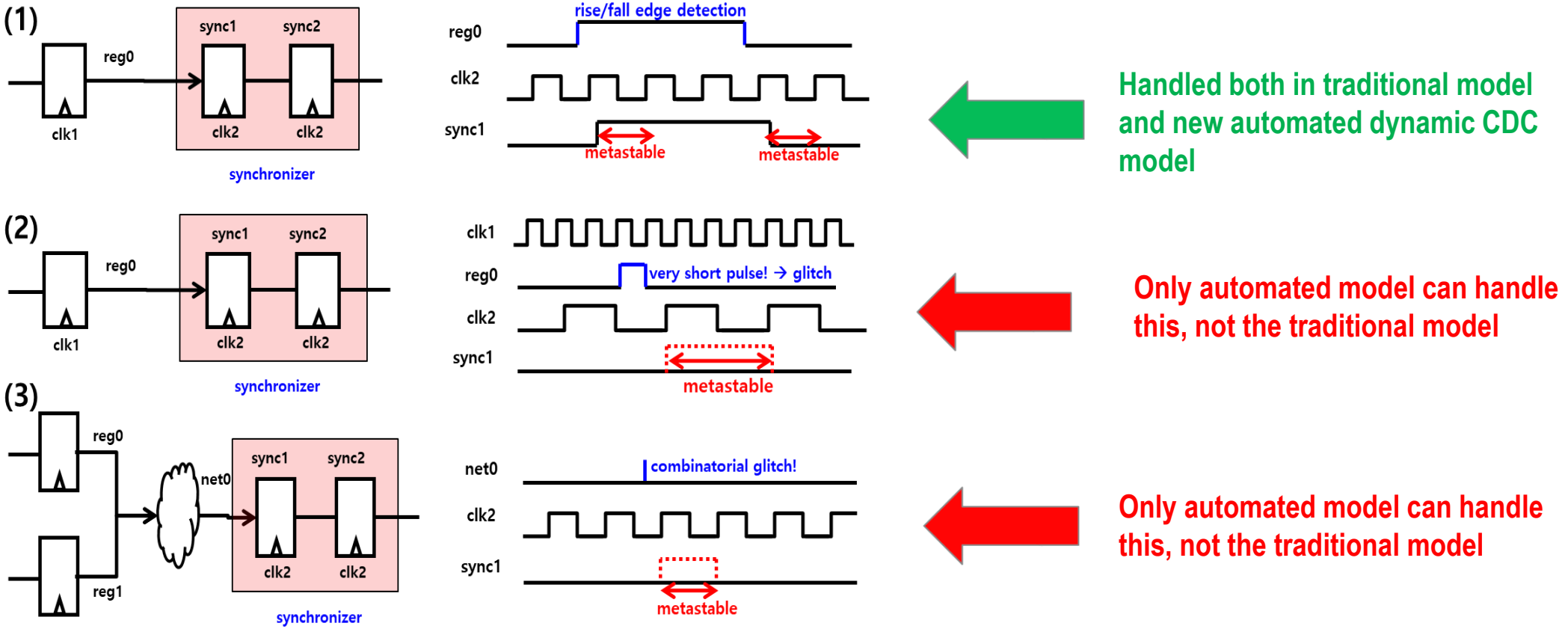
Inhouse Metastability-Injection Models may not be accurate

Dynamic CDC and Automated Models





Dynamic CDC and Automated Models



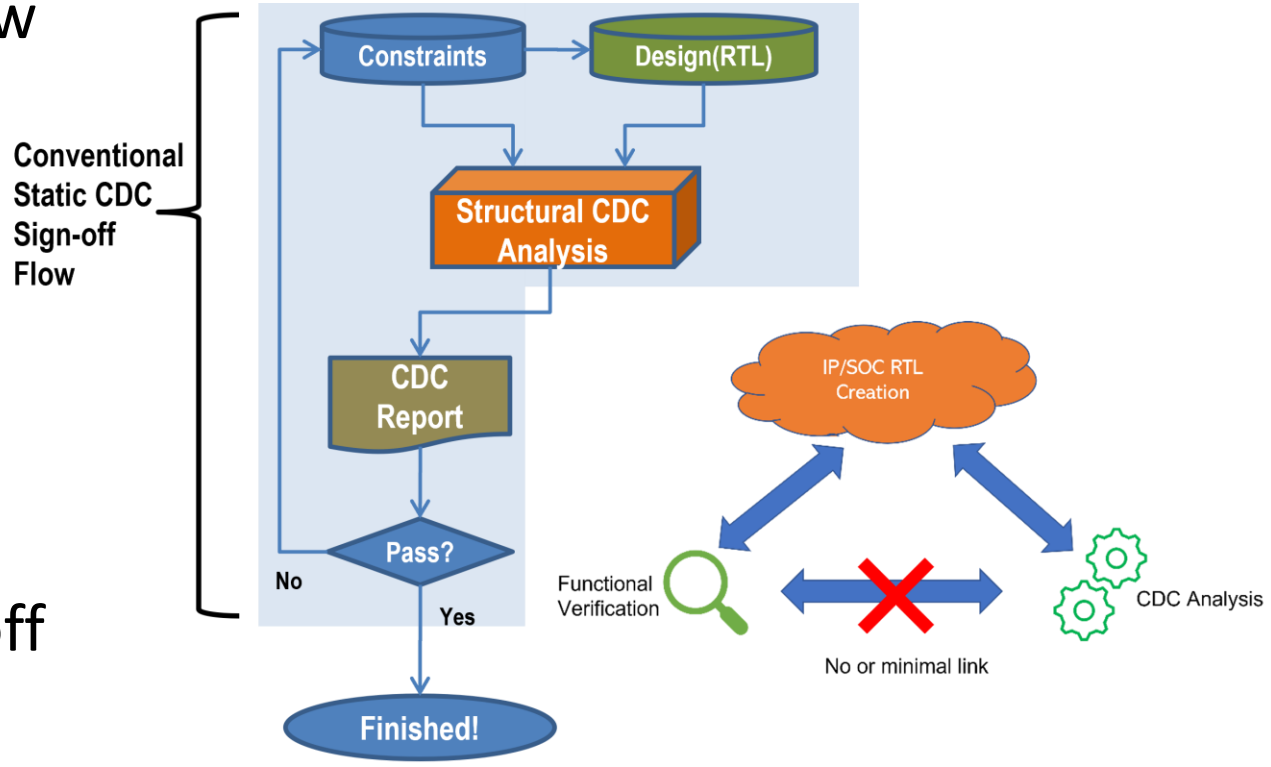
Handled both in traditional model and new automated dynamic CDC model

Only automated model can handle this, not the traditional model

Only automated model can handle this, not the traditional model

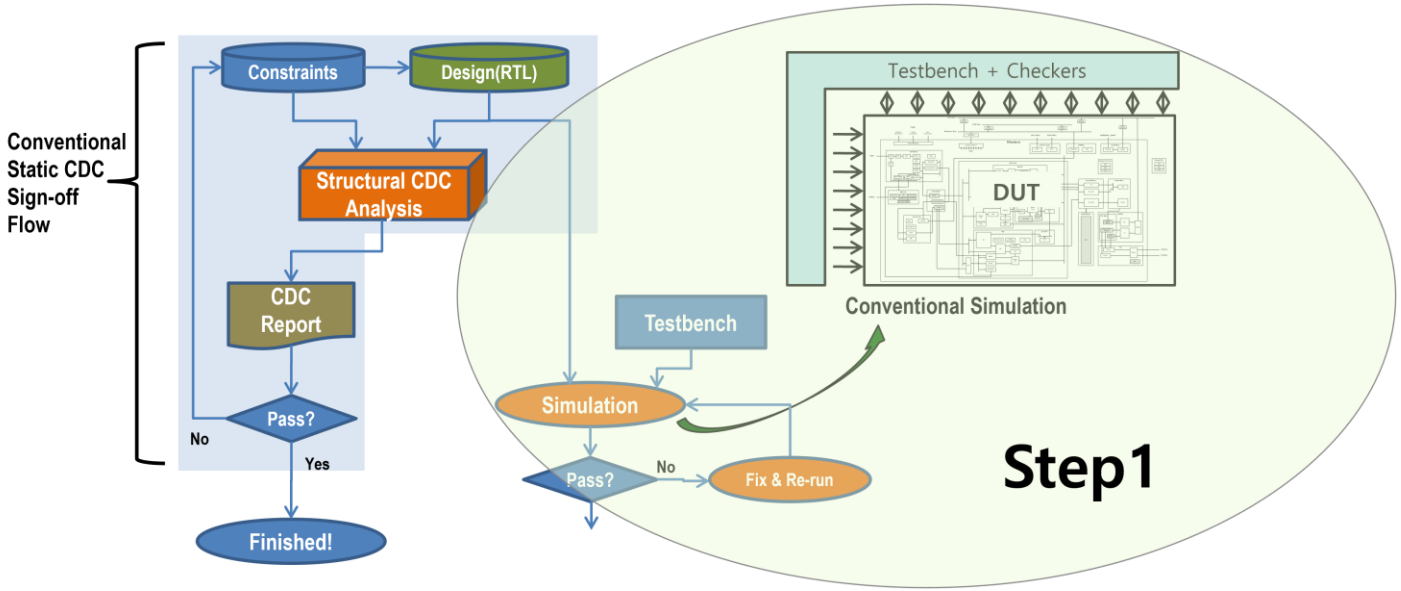
Traditional CDC Flow

- Conventional CDC Static Signoff Flow has no link between Static Signoff and functional verification
- CDC signoff is done independently with certain assumptions
- Functional verification is done independently with certain assumptions
- No minimal link between CDC signoff & functional verification
  - May lead to silicon issues falling through the cracks

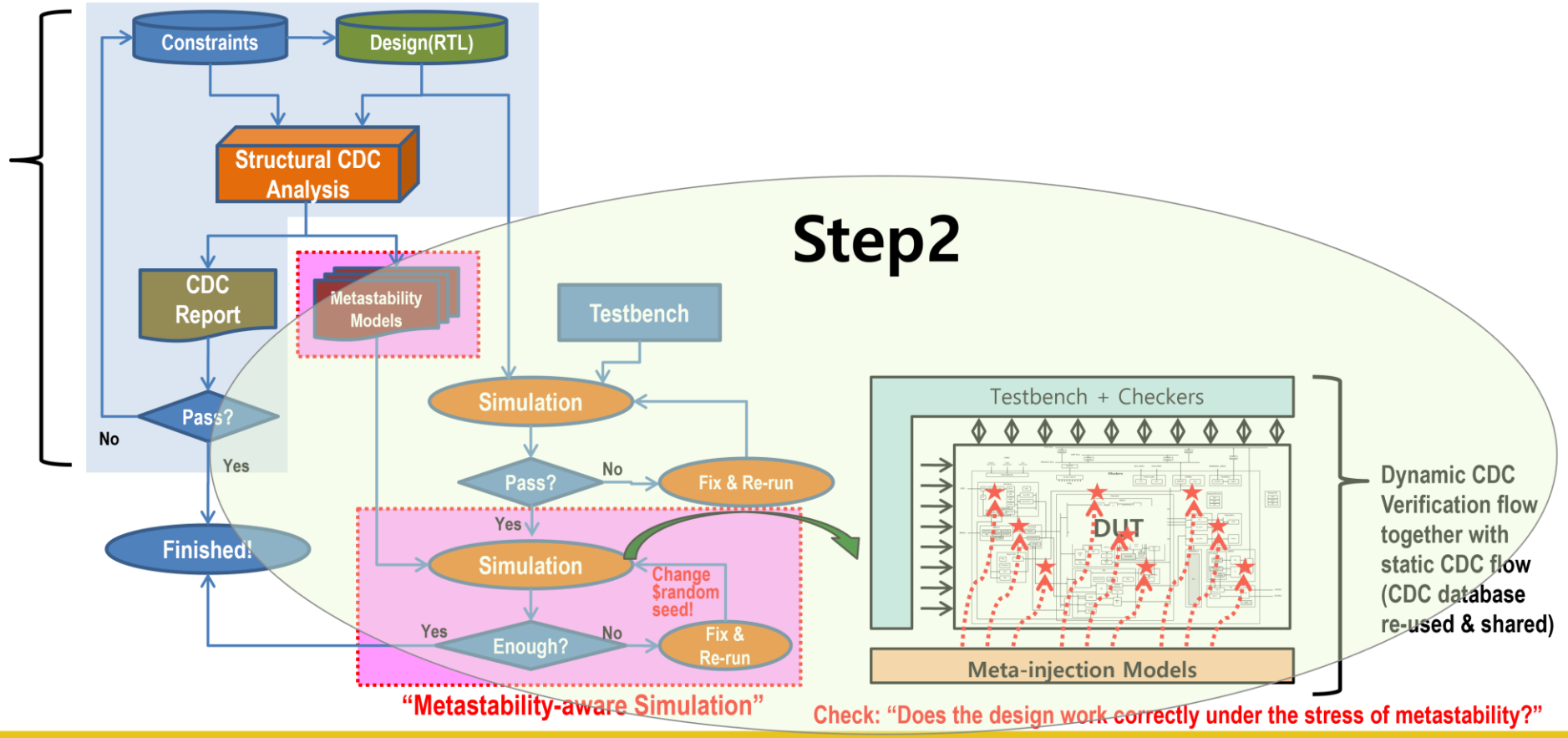


Samsung Dynamic CDC Flow in Conjunction with Static CDC Flow

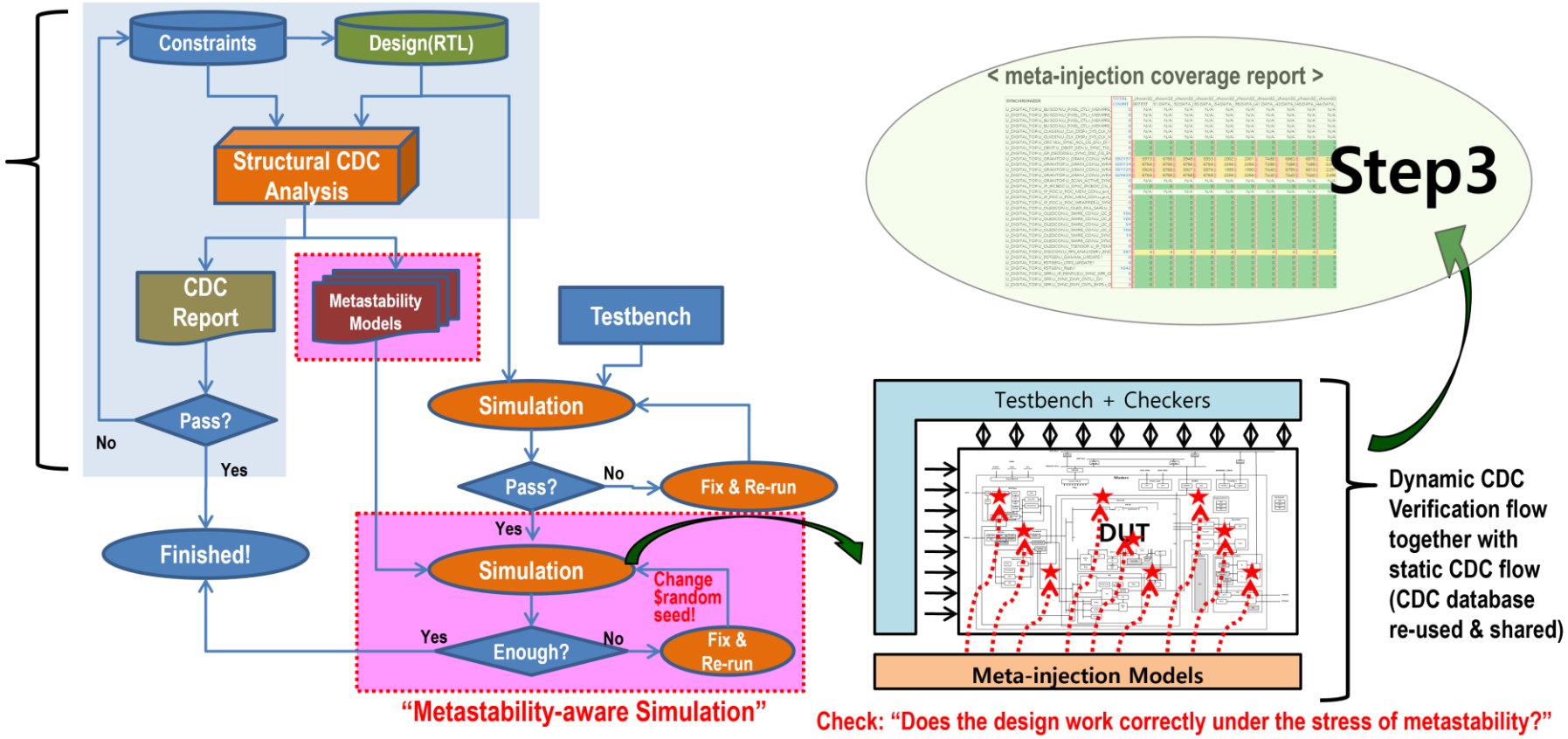
- We run Dynamic CDC verification flow together with Static CDC sign-off flow
- The first step to run Conventional simulation without any link to CDC
- This is to ensure simulation regressions are clean without any metastability effects



Samsung Dynamic CDC Flow in Conjunction with Static CDC Flow

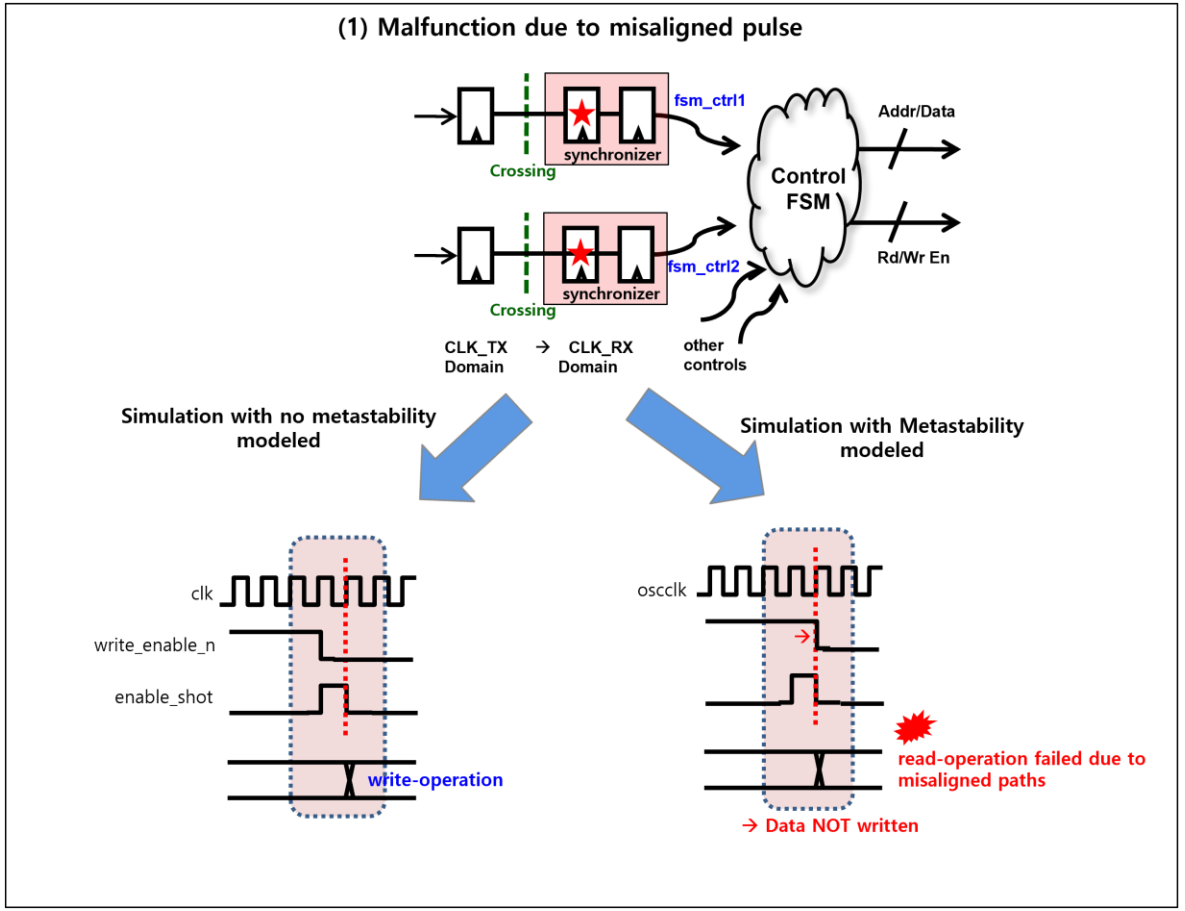


Samsung Dynamic CDC Flow in Conjunction with Static CDC Flow



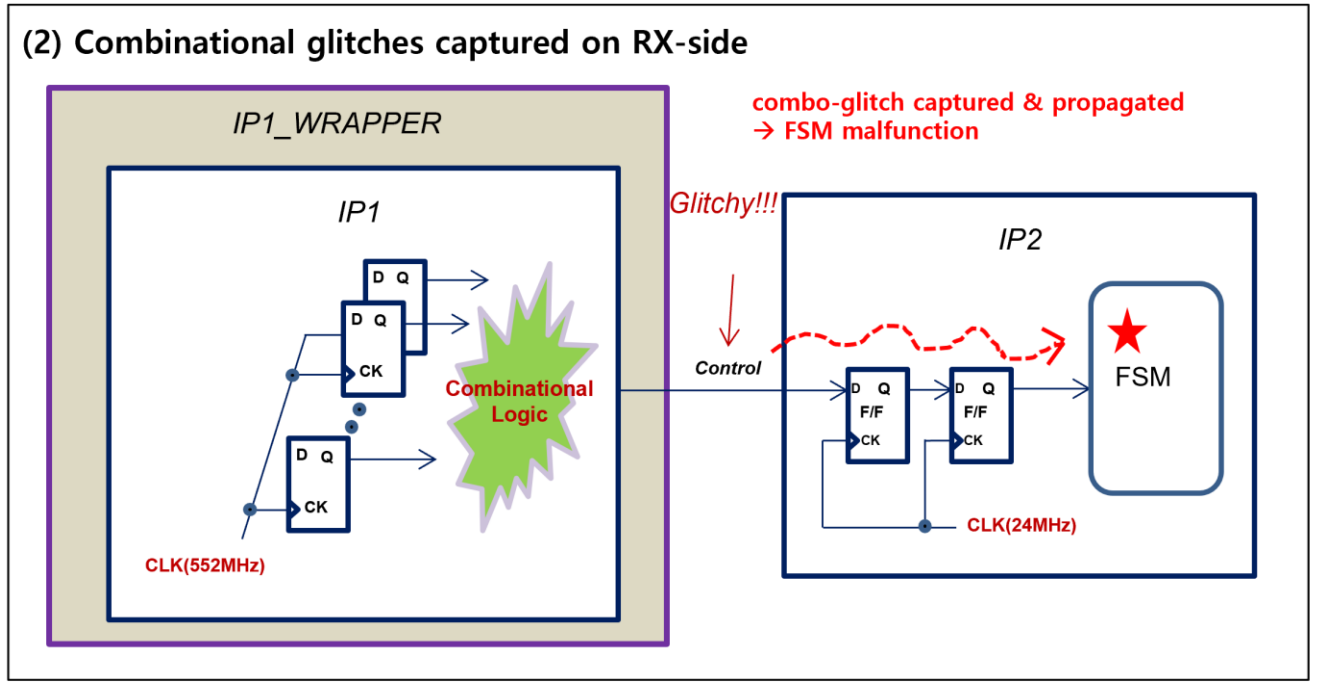
Bugs Revealed in Case Studies

- When metastability is not modeled in simulations
  - Design appears to work correctly
- When metastability is modeled in simulations
  - Read operation failure is observed in simulations



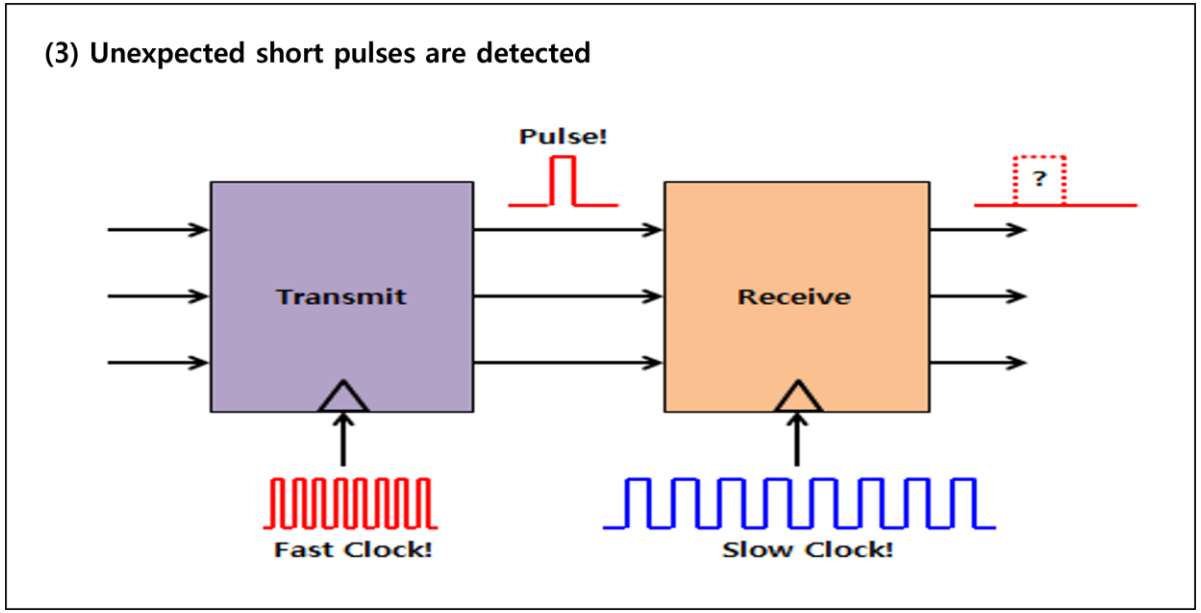
Bugs Revealed in Case Studies

- When design is simulation without metastability models
  - Design appears to work correctly
- When metastability is modeled in simulations
  - Combo glitch is captured and propagates in the design which leads to FSM malfunction



Bugs Revealed in Case Studies

- When metastability is not modeled in simulations
  - Design appears to work correctly
- When metastability is modeled in simulations
  - Unexpected short pulses are detected that are missed leading to design failures



Use Right Mix of Static and Dynamic Verification for CDC Signoff

- Strengthened metastability modeling compared to conventional synchronizer models
- CDC database for static sign-off is re-used for Dynamic CDC verification – No additional effort required
- Problems are detected that pure functional simulation does not reveal
- We recommend running dynamic CDC flow together with static CDC signoff for complete coverage of CDC failures

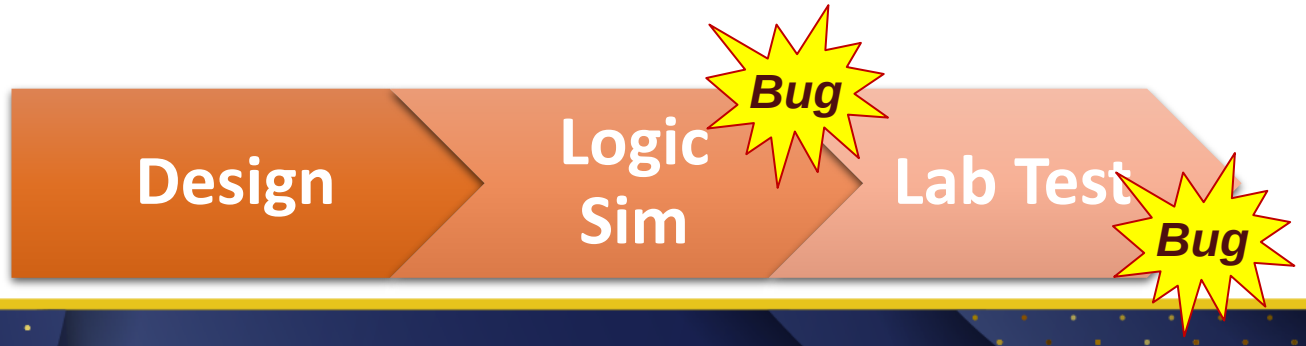
Fujitsu – 30% Reduction In Logic Simulation TAT Using Automatic Formal Techniques

Static Approach is Required for Efficiency

- SOC logic scale has become large and complex
- 100s of IPs in SOCs
- 100s of Clock Domains
- Huge amount of verification is needed

➢ *Bugs are missed in the design process*

- Static approach is essential in early debug and for quality improvement



30% Reduction in Simulation TAT Using Automated Formal Techniques

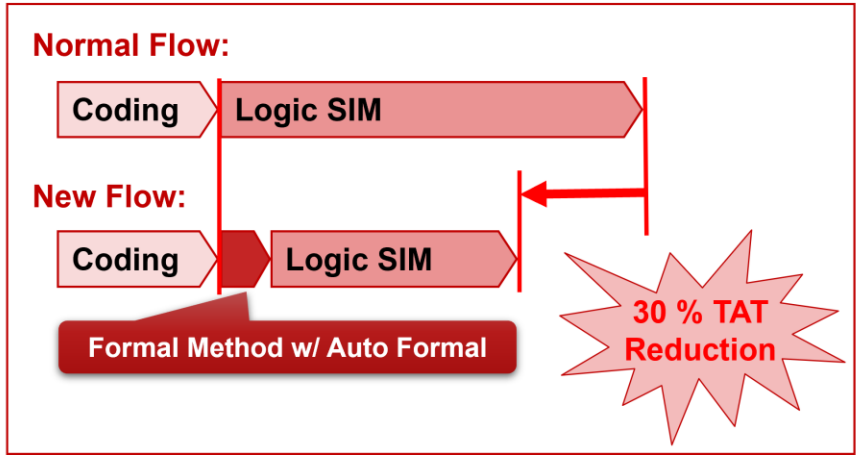
Efficient Early RTL Sign-off

- **Whole-chip analysis is achieved**
- **Found critical deadlock in FSMs in 2 projects**
  - In 3rd-party RTL => revelation to Fujitsu designers!
- **A 30% reduction in logic simulation TAT**
  - Primary-Secondary listing saved design iterations
  - Huge compression of items to review
- **Performed focused checks on RTL patterns**
  - Behavioral control
  - FSMs
  - Tristate drivers

List1: Simple Example of Failure Result for 106,356 Logic-gate SOC

	ERROR (Primary)	WARNING (Secondary)	INFO
DESIGN CHECKS	6	0	98
FSM CHECKS	1	9	348
LANGUAGE	0	0	31
COVERAGE	397	7214	92674

Designer could solve FSM issue by solve only one error debug
Other tool detected these as 10 errors (Not 1 error and 9 warnings)



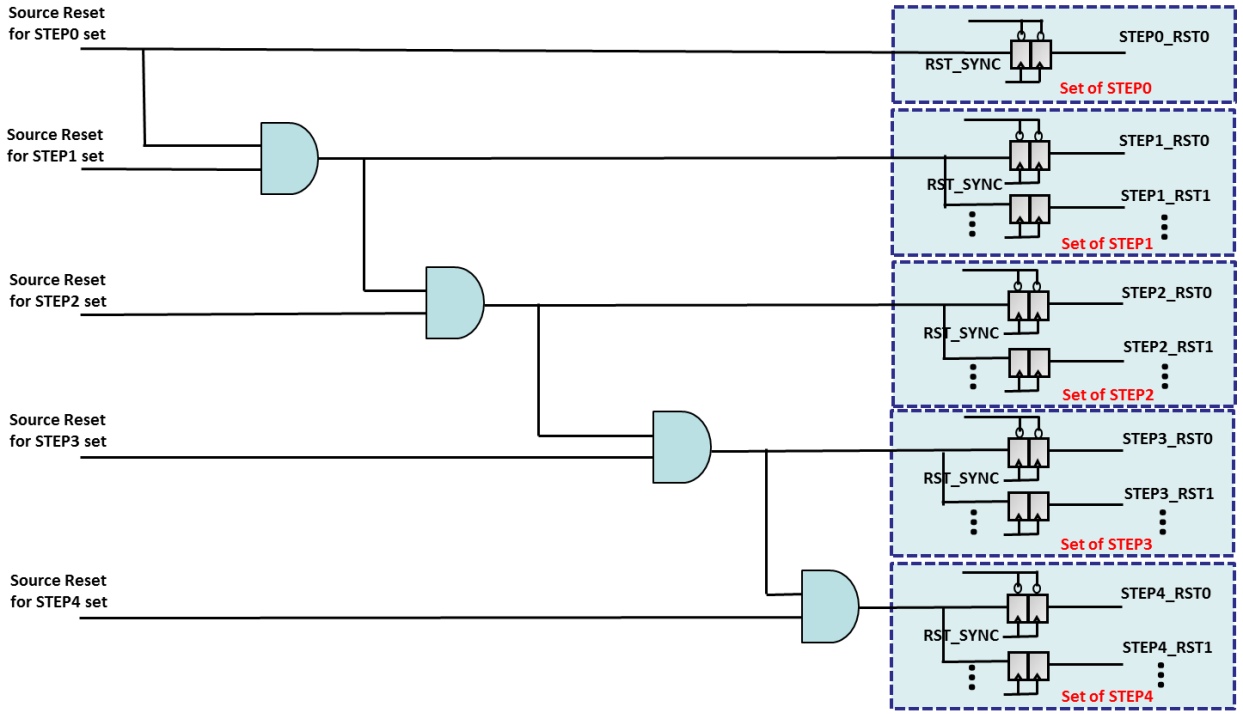
Use Effective Static Tools For Efficiency

- Static approach is essential for robust sign-off
  - Early RTL sign-off and CDC sign-off are iconic examples
  - Complex High-end Computer and Networking SOCs require systematic static sign-off
- Effective static tool introduction itself became systematic sign-off methodology

SK Hynix – Advanced Reset Design and Verification Methodology

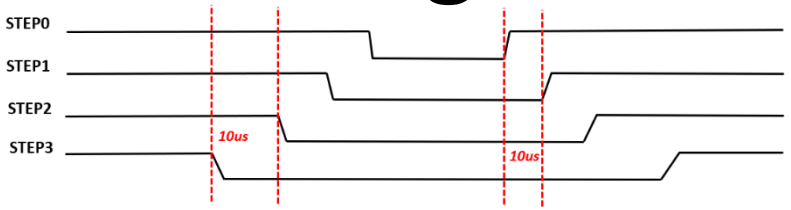
Our Reset Design Challenge

- Multiple primary resets
- Primary resets feed large number of synchronizers (secondary resets)
- Numerous combinations and interactions of primary and secondary resets and their clamping logic

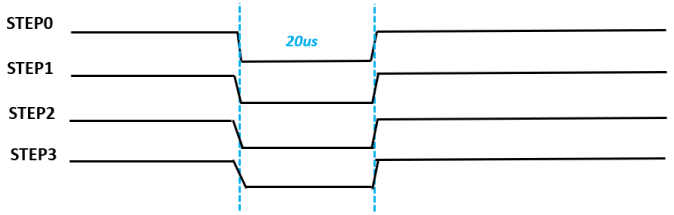


Our Reset Verification Challenge

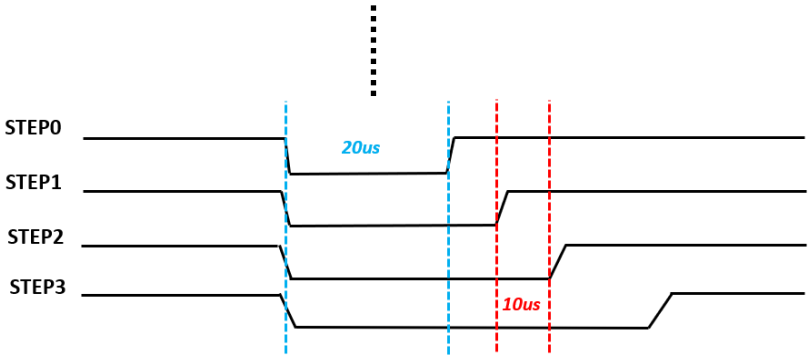
- Verification of multiple interactions of primary and secondary resets and their clamping logic
- Primary resets used in multiple complex waveform scenarios



Reset Sequence Case1



Reset Sequence Case2

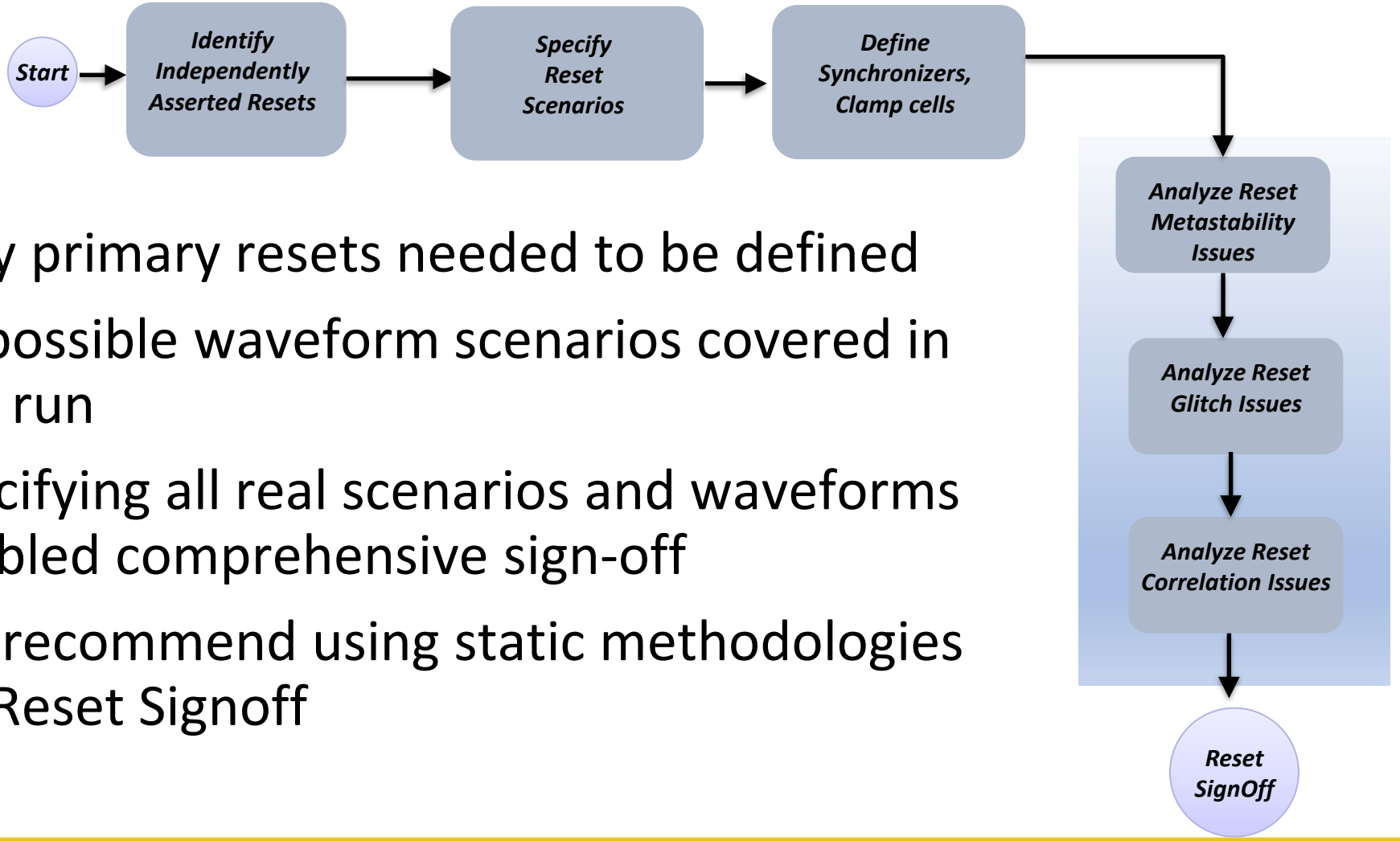


Reset Sequence Case N

Traditional Approaches and Limitations

- Dynamic verification using simulation
- Verification by running regressions tests on FPGA
- Limitations
  - Requires a large number of test benches to over all our reset cases
  - Requires large number of CPU and Human resources

Static Analysis Based Reset Sign-Off Methodology

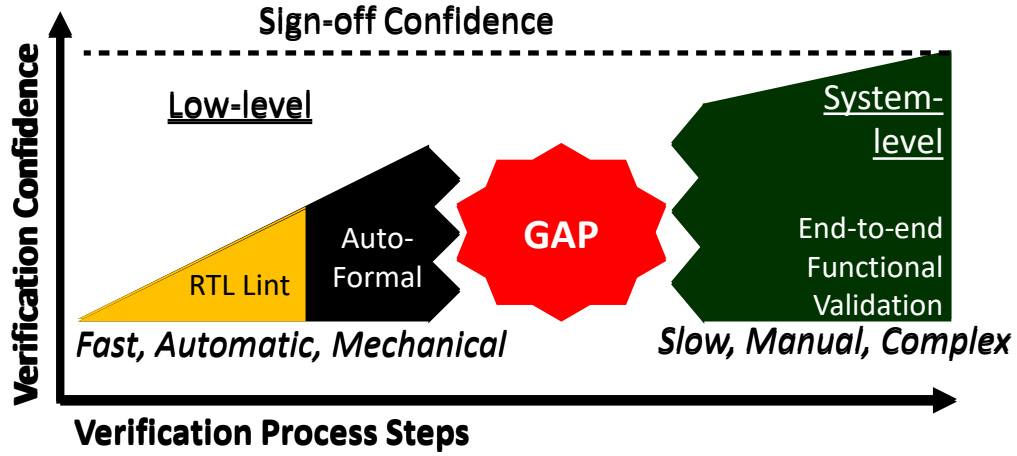


- Only primary resets needed to be defined
- All possible waveform scenarios covered in one run
- Specifying all real scenarios and waveforms enabled comprehensive sign-off
- We recommend using static methodologies for Reset Signoff

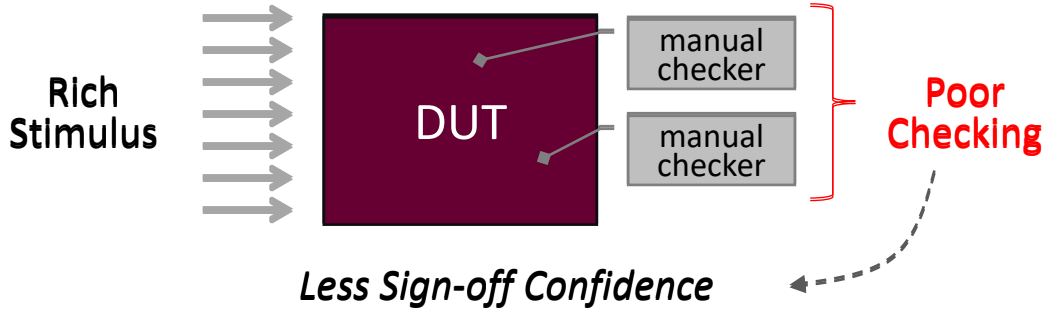
Renesas – Efficient functional sign-off by automatic assertion generation for RTL building blocks

The Verification Challenge

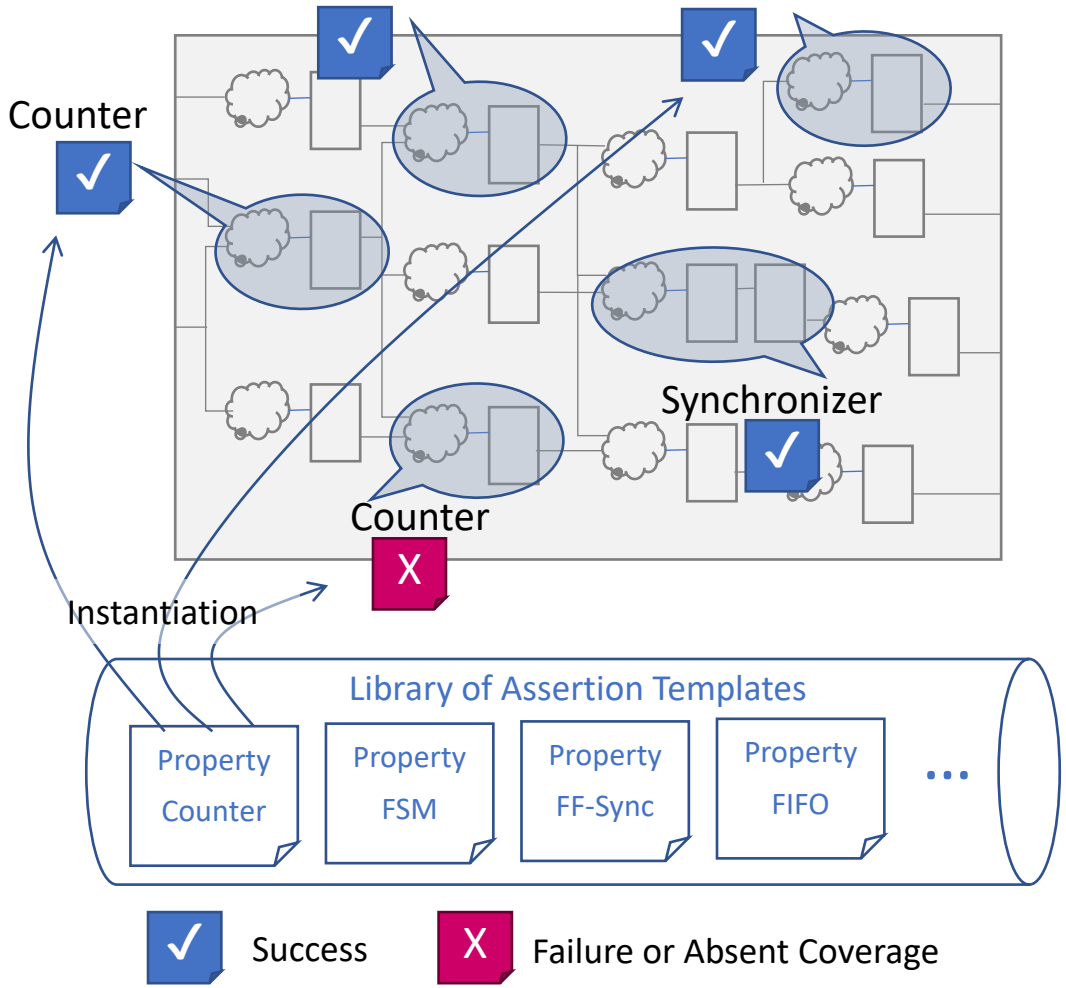
- System-level validation is complex, slow, and incomplete, pushing up HW design cost
- Systematic functional sign-off is an underserved imperative
- Vast gap between low-level RTL checks and system-level functional RTL sign-off
- *Must Bridge the Gap!*



- System-level validation is very hard due to
  - 3rd-party IP, Distributed design team, Legacy RTL in SOC assembly
- Stimulus automation has been the focus so far
  - Constraint random, Formal, PSS, UVM..
- *But, Manually-Guided Checkers are Slow, Unstable, and Insufficient*
  - Researching, planning, coding, reviewing, debugging..
- Need automation in checker generation also!



Auto-Inferred Building-Block Property Checking (AIPC)



- Most designs have primitive building-blocks
  - Counter, FSM, FIFO, Stack, FF-Sync, RAM, Shift-Reg etc.
- *Advanced Functional Static Analysis* successfully *automatically* infers such building-blocks in RTL
- Generate white-box assertions based on *Simple Assertion Template* for each building-block type
- Bind these assertions to RTL using co-generated bind files without user effort
- AIPC method allows uniform safety and coverage criteria to be created across a variety of implementations

AIPC Assertion Library

COUNTER

- Initialization
- Load data
- Hold data
- Count
- Loaded value

FSM

- State initialization
- State transition

FIFO

- Initialization
- Overflow
- Underflow
- Full
- Empty
- Empty pointer
- Full pointer
- Data integrity

STACK

- Initialization
- Push
- Pop
- Empty
- Full
- Data Integrity
- Empty pop
- Full push

RAM DP

- Initialization
- Data integrity
- Unknown value

RAM SP

- Initialization
- Data integrity
- Unknown value

SISO SHIFTREG

- Initialization
- Shift check

PIPO SHIFTREG

- Initialization
- Shift check

FF SYNC

- Initialization
- Data stability
- Synchronization

GRAYCODE SYNC

- Initialization
- Data stability

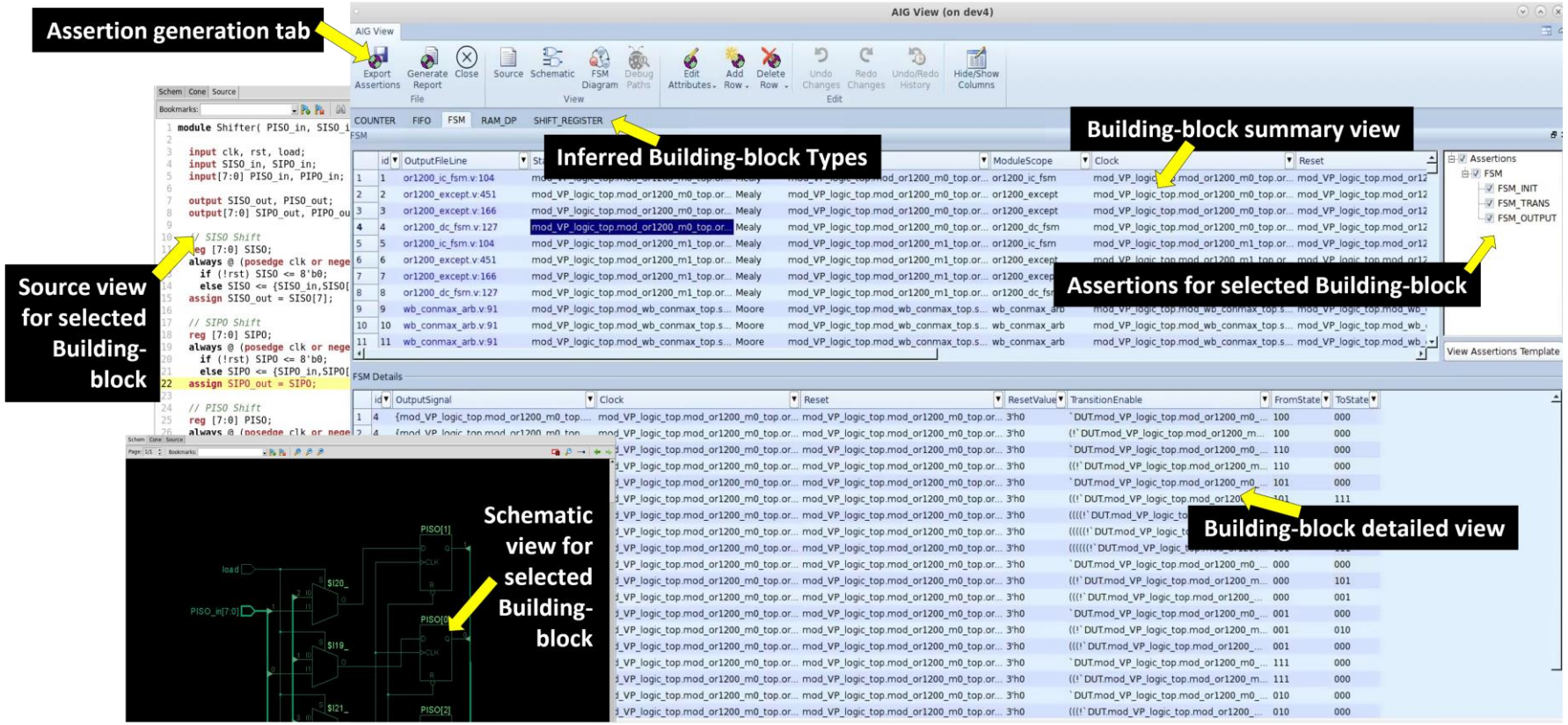
HANDSHAKE SYNC

- Data stability
- Complete cycle
- No ack without request
- Req not asserted until ack deasserted

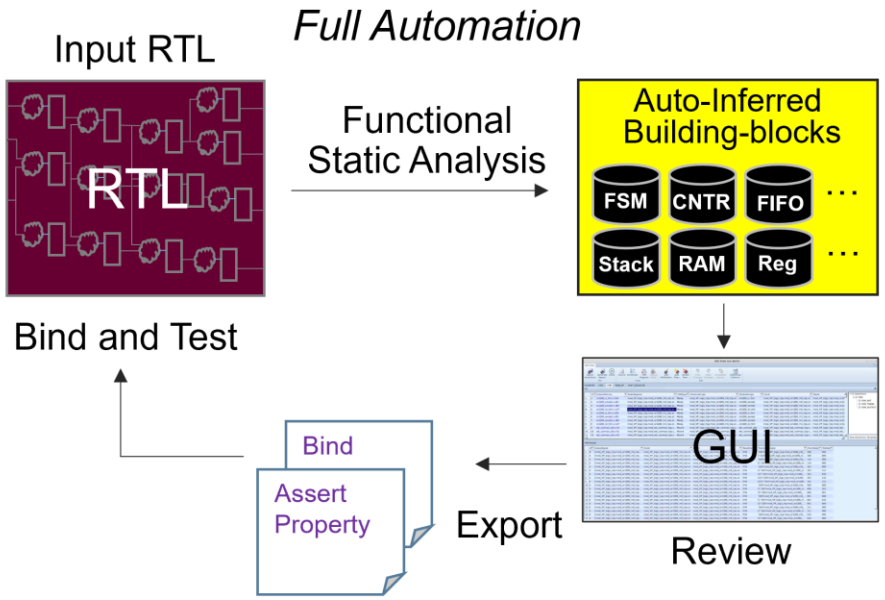
MUX SYNC

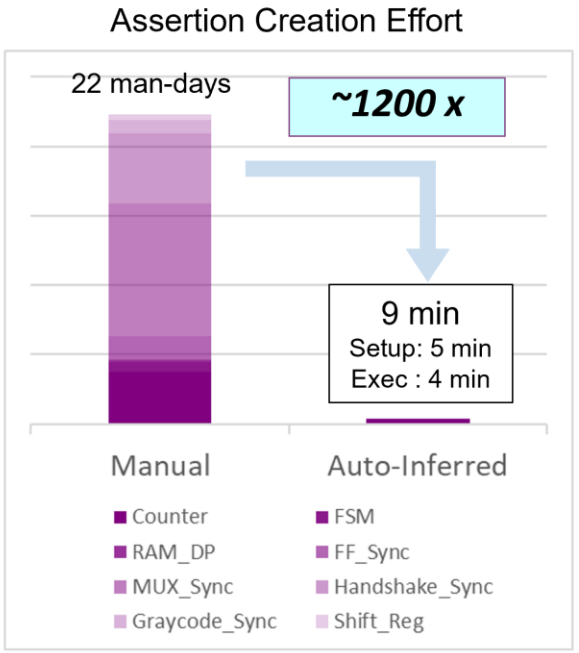
- Control signal stability
- Data stability

GUI Snapshots



Full and Instant Automation





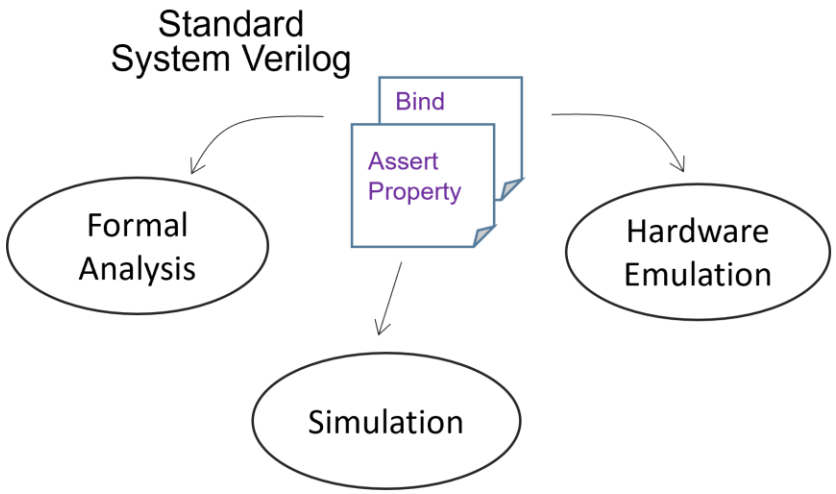
Communication Design IP Core
Design size : 800k Gates

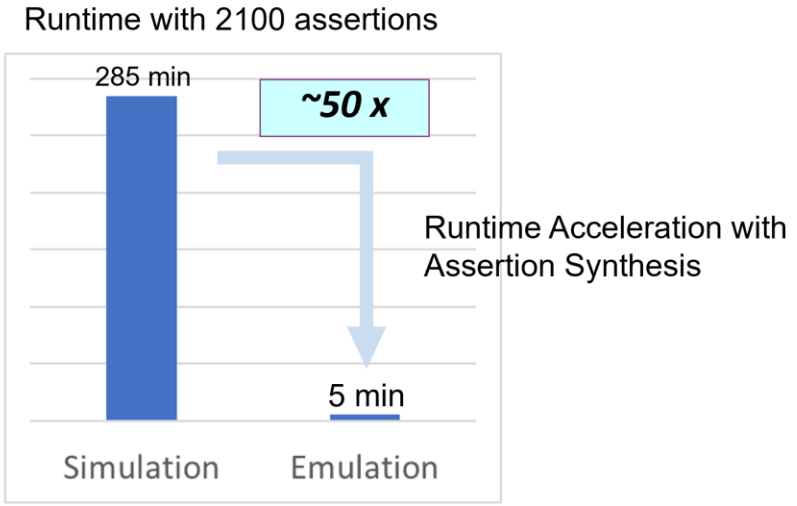
Number of Building-blocks	
Counter	299
FSM	59
RAM_DP	12
FF_Sync	200
MUX_Sync	1152
Handshake_Sync	608
Graycode_Sync	111
Shift_Reg	35

- Tool can automatically generate ready-to-use assertion files *Without Any User Hints*
- The generated assertions are applicable to every design with uniform quality

- *Extremely Fast Generation*
- Vast enhancement in productivity

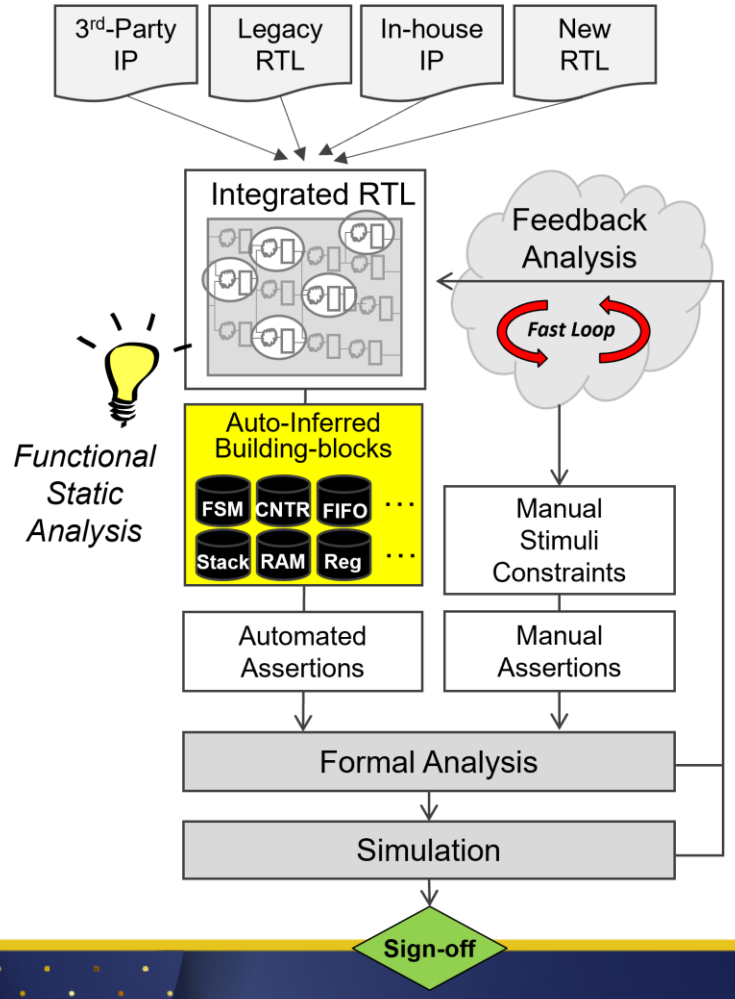
Multi-Purpose Use for RTL Verification





- Thousands of assertions cause a performance overhead in simulation
- In HW Emulation, *Assertion Synthesis* can accelerate runtime dramatically

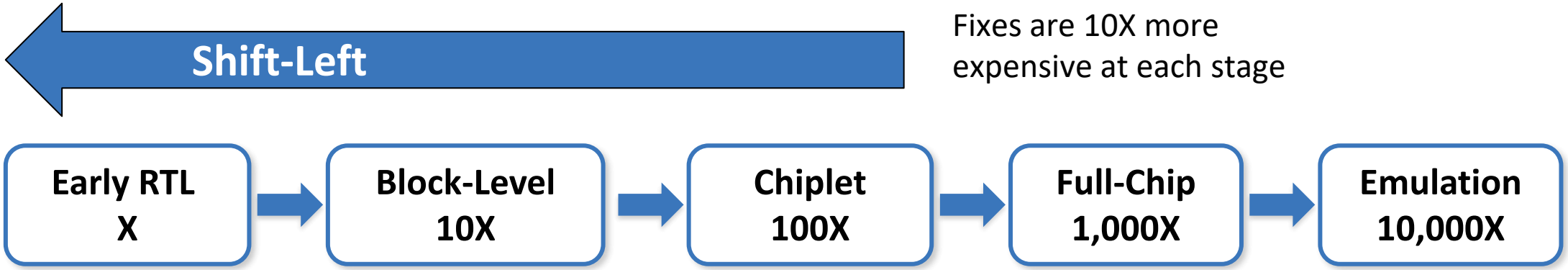
Verification Flow with AIPC



- IP-level to system-level validation becomes more efficient with AIPC method
- AIPC provides seeds that introduce *the initial review points* in a black-box RTL
- Analyzing *absent coverage and inconsistency* helps understand module-level behavior
- Leads to creation of manual assertions, stimuli, and constraints from *a bottom-up view* in addition to the existing top-down approach
- AIPC enabled methods are notably beneficial *for 3rd-party engineers or for engineers dealing with 3rd-party IP or Legacy RTL*

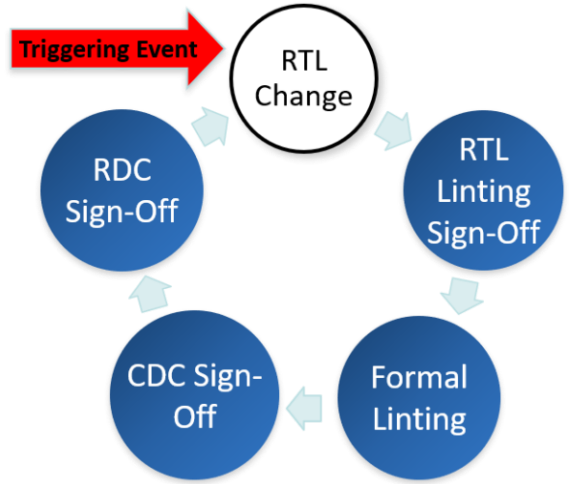
Summary: Static Signoff Best Practices

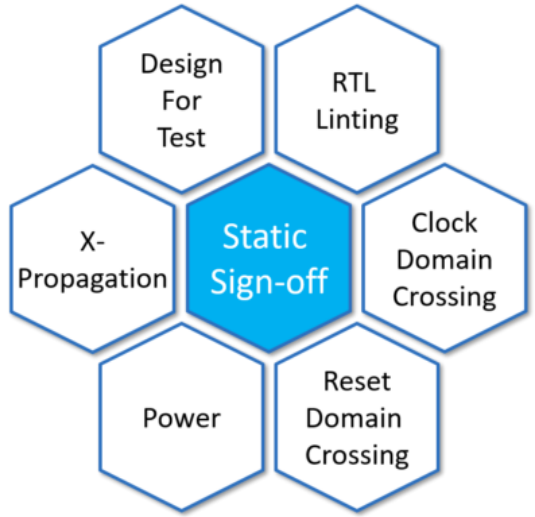
Static Sign-Off Key to Shifting Left



Static Sign-Off Best Practices

1 Start Early	• Bugs found early are less costly to fix
2 Include in Continuous Regression	• Detecting new issues immediately, before check-ins
3 Keep it Hierarchical	• Distribution of engineering effort
4 Ensure It's Complete	• Avoids missed errors with Multi-Mode CDC
5 Deploy Beyond CDC, STA, Lint	• Complete Static signoff includes RDC, X-Verification & DFT





Questions?

Source: istockphoto