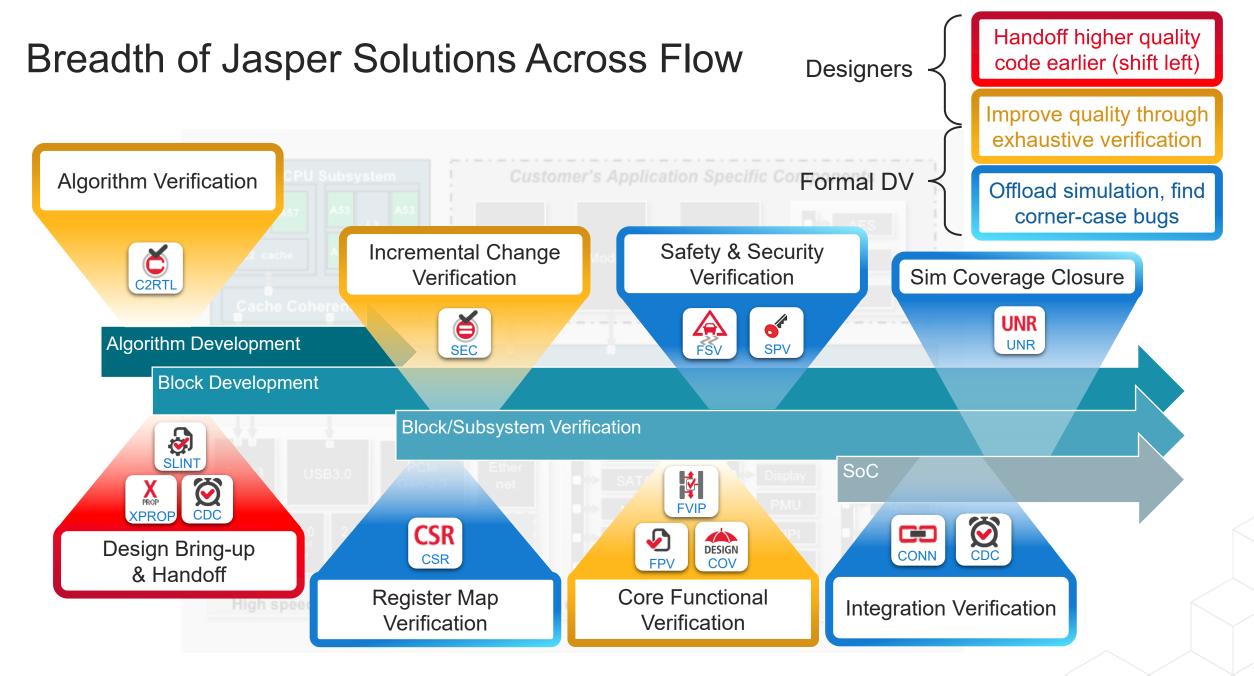


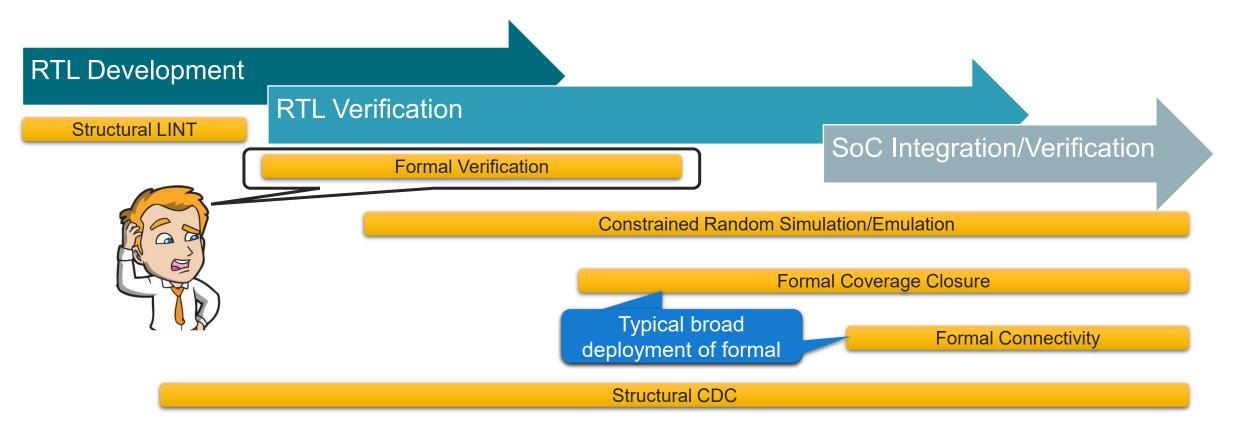
Easy Formal With Jasper Formal for Designers and Early Formal Verification

Kanwar Pal Singh **Product Engineering Group Director**





Design Verification Cycle



Challenge: Believe in the benefits of formal and want to deploy more but have limited or no expertise in house



Design Verification Cycle

Easy Formal w/ Jasper

R) Development

Structural LINT

Implementation Checks

Code Reachability
Overflow/Out-of-Bounds
FSM Deadlock
X Prop

RTL Bringup

"Heartbeat" waveforms
Basic asserts

RTL Verification

Formal Verification

alion

Constrained Random Simulation/Emulation

Formal Coverage Closure

Typical broad deployment of formal

Formal Connectivity

Broad set of high ROI use cases

SoC Integration/Verification

Highly automated → minimal expertise

Early in DV cycle → maximum impact

Structural CDC

Functional CDC

Protocol correctness

RTL Functional Equivalence (RTL == RTL'?)

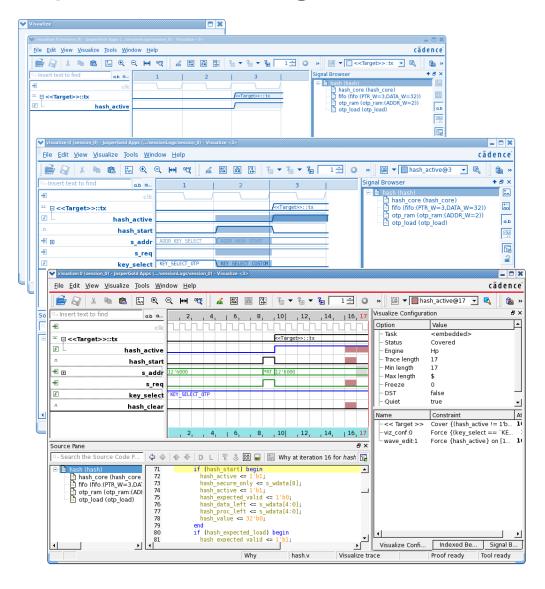
Clock gating
Code refactoring
PPA optimizations

cādence



Start With Formal: Design Exploration

Design Exploration using the Visualize™ Environment



Create waveform:

- Specify target expression
- ✓ Formal engines create waveform

Understand waveform:

- ✓ Full signal visibility
- Integrated source code with value annotation
- ✓ Why feature takes you directly to relevant signals and code
- ✓ Highlight differences between waveforms

Change waveform:

- ✓ Change signal behavior with WaveEdit feature
- ✓ Extend waveform length
- Add constraints on-the-fly for What-If analysis
- Quiet down stimulus with QuietTrace feature





Low-Hanging Fruit: Auto-Formal Checks

Auto-Formal Checks Set 1

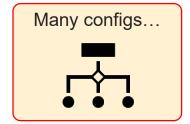
- RESET/INITIALIZATION CHECKS
 - All Flops initialized to a determined value
- AUTO_FORMAL_DEAD_CODE
 - RTL code block reachability
- AUTO_FORMAL_FSM_REACHABILITY
 - FSM state/transition reachability
- AUTO FORMAL SIGNALS
 - Signal Stuck-at
- AUTO_FORMAL_X_ASSIGNMENT
 - A reachable x-assignment was found
- AUTO_FORMAL_CASE
 - Default case reachability

Enabled with <u>Jasper Superlint</u>



- Properties automatically generated by the tool
- Need no constraints
- Violation message view/property view for debug
- Full featured GUI and Resilient waivers
- * Multi-Mode analysis

Multimode Analysis





Run <u>multiple configs</u> and disposition violations in one <u>comprehensive view of all modes</u>.

Compatible with all others next gen features.

Auto-Formal Checks Set 2 and 3

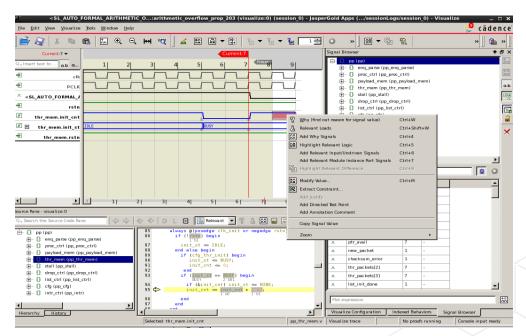
- Stage 1 Auto-Formal Checks +
- AUTO_FORMAL_CASE
 - Priority/Unique violation
- AUTO_FORMAL_OUT_OF_BOUND_INDEXING
 - Index out of range
- AUTO_FORMAL_OVERFLOW
 - Overflow because of arithmetic operation, shift, Divide-by-zero, non-arithmetic
- AUTO_FORMAL_BUS
 - Bus Multiply driven, contention and floating
- AUTO_FORMAL_COMBO_LOOP
 - Combinational loop detection involving listed signals/wires/expressions
- AUTO_FORMAL_DEADLOCK_LIVELOCK
 - FSM state Deadlock/Livelock detection

Enabled with **Jasper Superlint**



In addition to Stage 1

- Specify clocks/resets and some basic constraints
- FSM browser to debug FSM related failures
- Jasper Visualize to debug using waveforms





CDC Protocol Checks

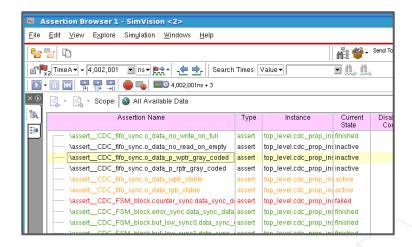
Enabled with Jasper CDC



- CDC protocol checks
 - Auto-generated protocol checks for all identified synchronizers
 - Supports user-defined protocol checks for custom synchronizers
 - Requires very strong formal tool capabilities
 - Best in class Jasper formal engines
 - Export protocol checks to run in simulation

# ig_clock_control1-> jg_clo # ig_clock_control2-> jg_clo # ig_clock_control2-> jg_clo # ig_clock_control3-> ig_clo	🖔 Pairs 📗 🧨 Schemes 🗎 🐉 Conve	e 🚨 Functional 🗎 👭 Meta	astability 🖪 Reset				
# ig_clock_controll -> jg_clo # jg_clock_controll -> jg_clo # ig_clock_controll -> jg_clo		76					
# jg_clock_control1 -> jg_clo # jg_clock_control2 -> jg_clo			Scheme Property	Check	Scheme	Scheme Type	Detection Type
## jg_clock_control1 -> jg_clo ##ck.dataout_src_req_hold		✓	nc.out_sync_data_stable	data_stable	eq_sync.out_sync	NDFF	Automatic
Wind a continuation Wind a continuation	jg_clock_control2 -> jg_clo	✓	nc.out_sync_data_stable	data_stable	ck_sync.out_sync	NDFF	Automatic
✓k.dataout_dest_ack_hold		✓	ck.dataout_src_req_hold	src_req_hold	t.rx_block.dataout	Handshake	Automatic
✓k.dataout_dest_new_ack ✓taout_data_stability_dest ✓under_pulse_data_stable ✓tabet_pulse_data_stable ✓tabet_pulse_data_stab		✓	ck.dataout_src_new_req	src_new_req	t.rx_block.dataout	Handshake	Automatic
 taout_data_stability_dest ounter_pulse_data_stable l_state_pulse_data_stable l_state_pul		✓	k.dataout_dest_ack_hold	dest_ack_hold	t.rx_block.dataout	Handshake	Automatic
 ounter_pulse_data_stable data_stable I_state_pulse_data_stable data_stable I_state_pulse_data_stable I_state_pulse_data_s		✓					Automatic
 ounter_pulse_data_stable data_stable I_state_pulse_data_stable data_stable I_state_pulse_data_stable I_state_pulse_data_s		✓	taout_data_stability_dest	data_stability_dest FL	unctionally checks	s if the e	Automatic
Stable Clock domain remains Automatic		×	ounter_pulse_data_stable				Automatic
		X 1	I_state_pulse_data_stable	data_stable C	clock dornain remains		Automatic
		Total: 53 Filtered: 53			captured properly in the		

Protocol Checks	Synchronization Schemes			
Control Path Stability	NDFF, NDFF_BUS, Pulse, Edge, Glitch Protector			
Data Path Stability	MUX_NDFF, MUX_PULSE, Sync Enabler, Glitch Protector			
Pulse Width Toggle Circuit	Pulse			
Req/Ack Stability Data Stability Handshake Protocol	Handshake Synchronizer			
Rear/Write Pointer Stability Overrun/Underrun Gray Encoding of Pointers	FIFO Synchronizer			





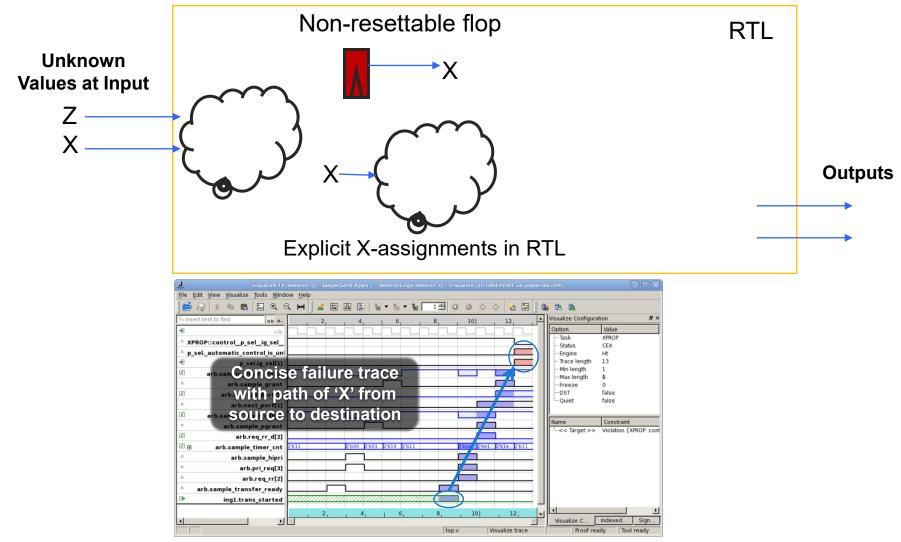


Worried About Xs?

Auto-X Checks

Enabled with Jasper XPROP





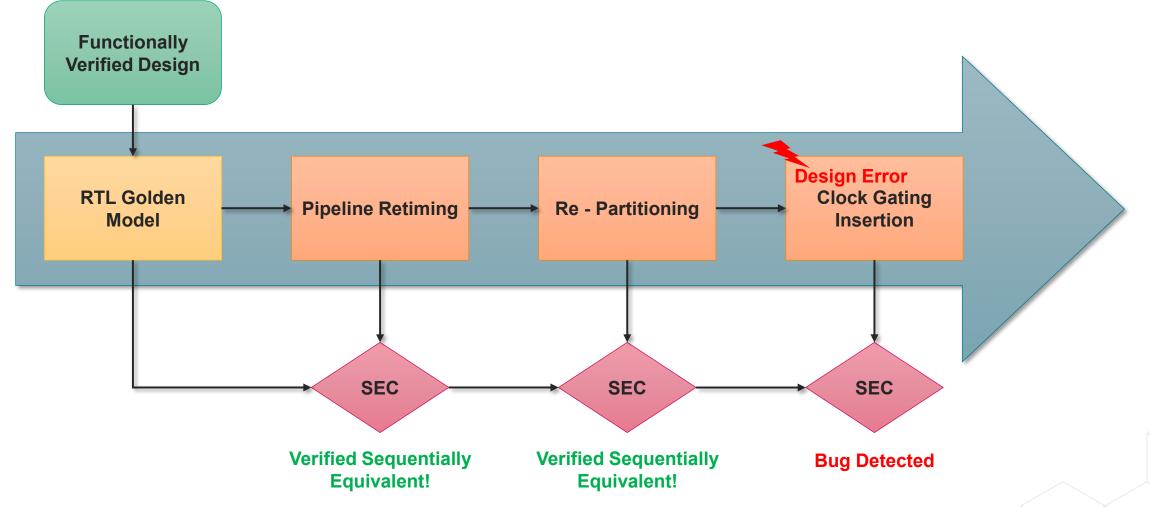


Changing RTL For PPA Optimizations?

Check Functional Equivalence After Each Change

Enabled with <u>Jasper SEC</u>







Bringing it All Together

The Intersection: Designer/Early DV Tasks in Context of Timeframes

Check-In (Frequent) Regression **Key Milestones/Signoff** (Nightly/Need basis) **RTL** Regression Tasks + Check-In Tasks + Compile + LINT Code Sim: Gate level Sim: Sanity Test Suite Sim: Full Test Suite **Synthesis** Trial ATPG Synthesis compile + DFT **DFT LINT** CDC/RDC Analysis Trial P&R **Timing** PPA Analysis Power **Asserts**

Augmented Task Map

Formal Enabled

RTL Code **Synthesis** + DFT **Timing** Power **Asserts**

Check-In (Frequent) Compile + LINT Sim: Sanity Test Suite SLINT Auto Formal Set 1

Regression (Nightly/Need basis) Check-In Tasks + Sim: Full Test Suite Auto Formal Set 2, 3 Synthesis compile **DFT LINT** CDC/RDC Analysis **CDC** CDC Auto-Checks

Auto-X Checks XPROP

Key Milestones/Signoff

Regression Tasks +

Sim: Gate level

Trial P&R

Trial ATPG

PPA Analysis
Clock Gating

cādence®

© 2025 Cadence Design Systems, Inc. All rights reserved worldwide. Cadence, the Cadence logo, and the other Cadence marks found at https://www.cadence.com/go/trademarks are trademarks or registered trademarks or logo. All other trademarks or trademarks or service marks owned by MIPI Alliance. All PCI-SIG specifications are registered trademarks or trademarks or trademarks are the property of their respective owners.