2022 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Advanced Functional Verification for Automotive System on a Chip

Jaein Hong, Jieun Jeong, Namyoung Kim, Hongkyu Kim, Sungcheol Park, Samsung Electronics

Sangjun Mun, Cadence Design Systems

accellera SYSTEMS INITIATIVE

Agenda

- Motivation
- Ensuring Verification Completeness on Low-Power Design
- Increasing Verification Efficiency for Self-Diagnosis and Recovery Design
- Conclusion

Motivation

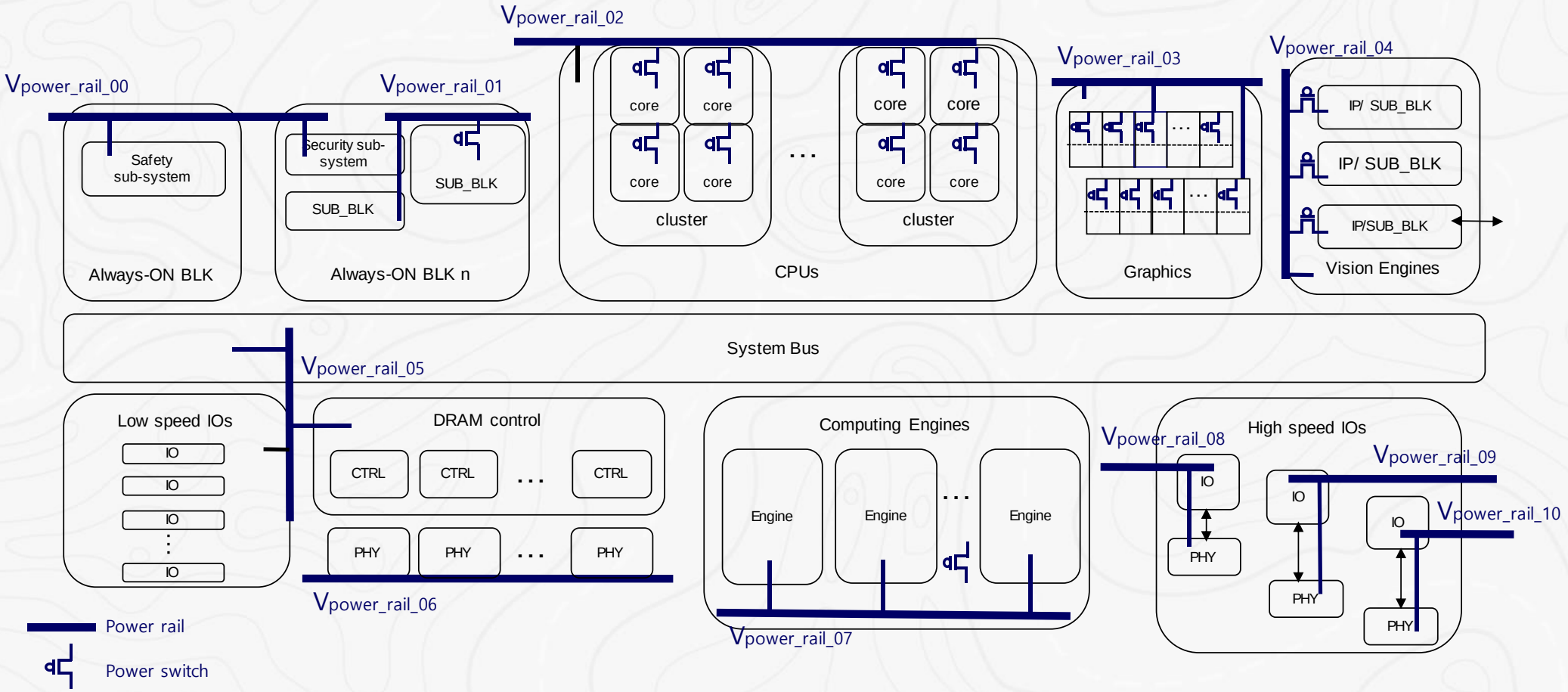
- Automotive applications
  - Increase the number of integrated components
    - High performance computing – machine-learning engines, CPU cores, and GPU cores
  - Increase the requirements of low power design
    - Isolated safety/security islands – multiple voltage islands, complex power modes, sophisticated clock/power gating control
  - Increase the requirements of system reliability
    - Self-diagnosis and recovery scheme, fault-tolerant design

Impose huge verification challenges
- Completeness (Quality) vs. TAT

Agenda

- Motivation
- **Ensuring Verification Completeness on Low-Power Design**
- Increasing Verification Efficiency for Self-Diagnosis and Recovery Design
- Conclusion

Example of Automotive SoC Power Scheme



Challenges for Low Power Verification

- Explosion of power modes and state transitions

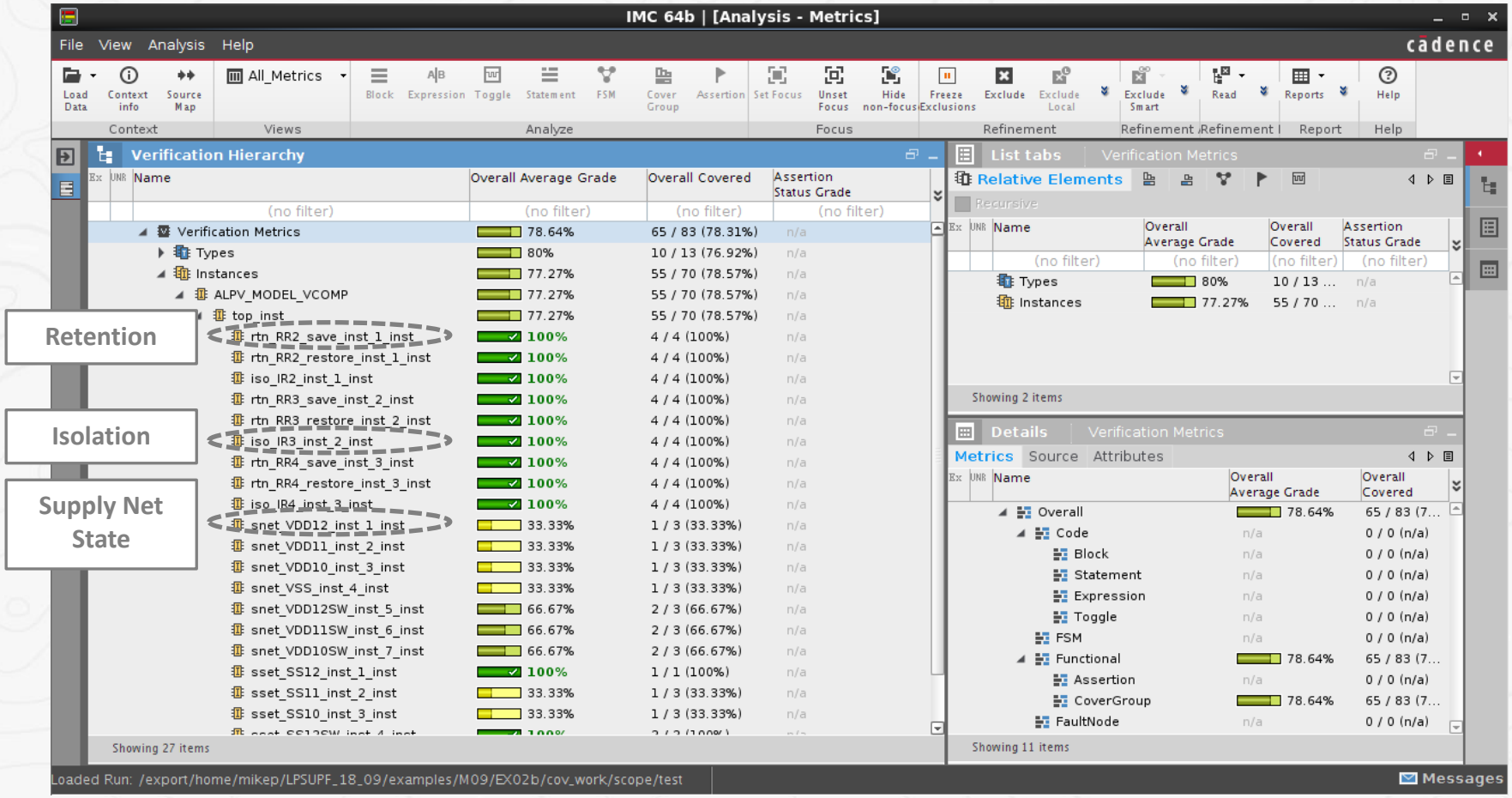
Power modes	BLK_A	BLK_B	BLK_C	CPU	Graphics	BLK_D	BLK_E	...	BLK_XX
AON_0 mode	ON(possibly gating)	OFF	OFF	OFF	OFF	OFF	OFF	...	OFF
AON_1 mode	ON	ON(possibly gating)	OFF	OFF	OFF	OFF	OFF	...	OFF
...	...	...	...	...	...	...	...	...	...
AON_n mode	ON	OFF	ON	OFF	OFF	OFF	OFF	...	OFF
Usage_0 mode	ON	ON(possibly gating)	OFF	ON(possibly gating)	OFF	ON	OFF	...	OFF
Usage_1 mode	ON	ON(possibly gating)	ON	ON(possibly gating)	ON(possibly gating)	OFF	ON(possibly gating)	...	OFF
...	...	...	...	...	...	...	...	...	...
Usage_n mode	ON	ON(possibly gating)	OFF	ON(possibly gating)	ON(possibly gating)	ON	OFF	...	OFF
ALL ON mode	ON	ON	ON	ON(possibly gating)	ON(possibly gating)	ON	ON(possibly gating)	...	ON

Auto-generated Power Coverage based on UPF

- Power-aware simulator automatically generates power coverage
  - Supply net states
  - Supply set sim-states
  - Power switch
  - Power state table (PST)
  - Control signals of isolation and retention

Analyze auto-generated Power Coverage

- Example of coverage database



Extending functional coverage bins

- Limitation of auto-generated coverage bins

- Solution: Combining system-level power management events with a set of system operations that access and change the state of power-control signals

Coverage combining system reset with power state

- Automotive SoC has multiple system resets with different purposes and reset regions.

```
 cp_system_reset_power_mode: coverpoint
system_power_state iff($rose(system_reset)){
 bins power_state_b[] = {ALL_ON, Usage_1, Usage_n};
 }
```

Coverage for power state transition during interrupt handling

- Interrupt signals may pass across many power domains.
- Wrong clamp value of isolation can cause unexpected interrupts or a system malfunction

```
cr_intr_during_power_state_transition:
cross cp_interrupt_00,
cp_system_power_state_transition_AON_0;
```

Coverage for LPI handshaking

- LPI handshaking of IPs in a power domain must be completed before the power domain is turned off to ensure that there are no remaining transactions.
- Missing LPI handshaking or wrong order of LPI handshaking can cause system malfunction.

```
 cr_qch_transition_and_power_state: cross
cp_qch_state_transition, cp_system_power_state;
```

Enhancing debug capabilities using assertions

- Identifying X problems in real silicon, necessitates X-propagation verification in power-aware simulation.
- However, power-aware X-propagation simulation is time consuming and tedious for bug identification.
- Solution: Using assertions
  - Helps verification engineer to easily identify an X-problem before running power-aware X-propagation simulation
  - Enables faster power-issue detection without waveform analysis

Example of low-power assertions

- Power switch/isolation/retention control signals must not be X or high-z
- Initial value of power switch/isolation/retention control signals
- Relation between isolation control signal and power domain shutoff
- Retention: Signals for save and restore should not toggle during power off. Retention save process should be done before power off and restore.
- Power on/off sequence: The order of power supply, switch, isolation, retention, and reset controls must follow the specification.

Reference assertion code for isolation signal

- Relation between isolation control signal and power domain shutoff

```
 always @(PD_PLL_A_ISO_IN_signal) begin
 CHK_ISO_TOGGLE_WHILE_SHUTOFF__PD_PLL_A_ISO_IN_signal: assert
(PD_PLL_A_shutoff==0) else $display("Isolation control signal
should not toggle while power domain is shutoff");
 end
 always @(posedge PD_PLL_A_shutoff) begin
 CHK_ISO_ENABLE_BEFORE_SHUTOFF__PD_PLL_A_ISO_IN_signal: assert
(PD_PLL_A_ISO_IN_signal) else $display("Isolation should be
enabled before power domain shutoff");
 end
```

Bugs Found

- Wrong sequence of retention save/restore and power switch signals
- Missing isolation between analog components and digital-logic power domains.
- Wrong isolation values
- Missing LPI handshaking
- Incorrect order of LPI handshaking for IP and asynchronous bridge
- Power Control Module errors – incorrect power control signal sequencing
- Missing test scenarios that combine real IP operations and power mode transitions

Agenda

- Motivation
- Ensuring Verification Completeness on Low-Power Design
- **Increasing Verification Efficiency for Self-Diagnosis and Recovery Design**
- Conclusion

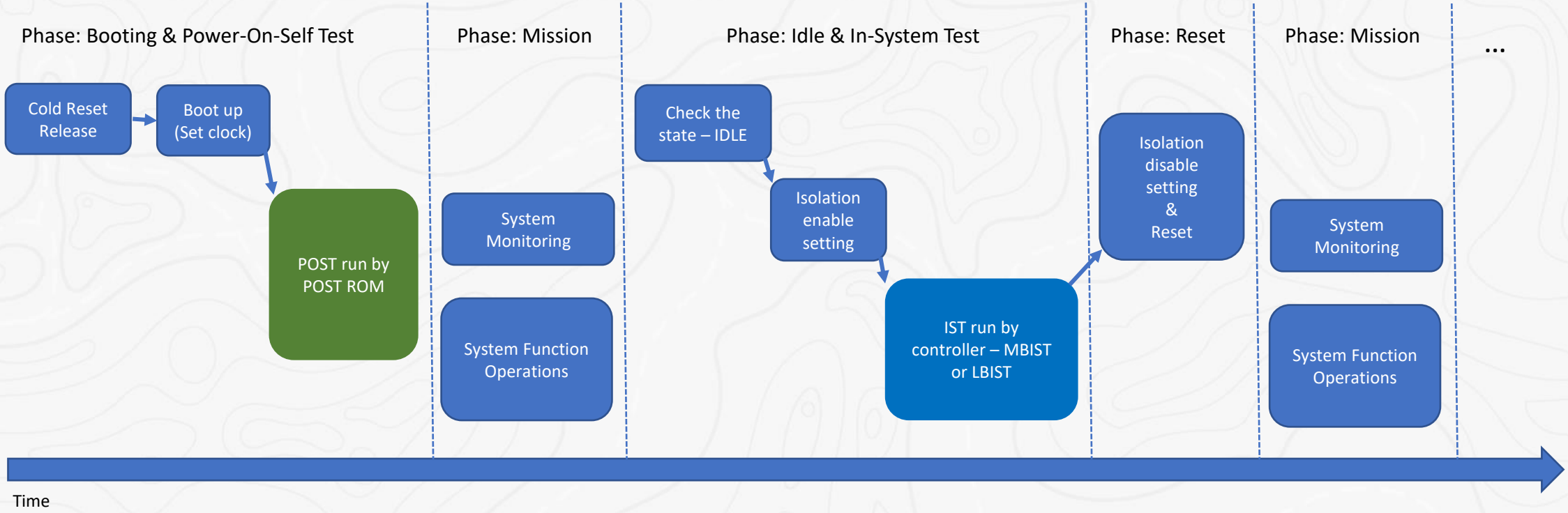
Automotive System Reliability

- Functional safety requirements
  - Whenever system is turned on, system should be checked to ensure its proper and safe → POST (Power-On-Self-Test)
  - While system is running, system should be checked periodically for random or aging related defects → IST (In-System-Test)

- Built-In-Self-Test on Automotive SoC
  - Memory BIST
  - Logic BIST – implemented using SCAN chain

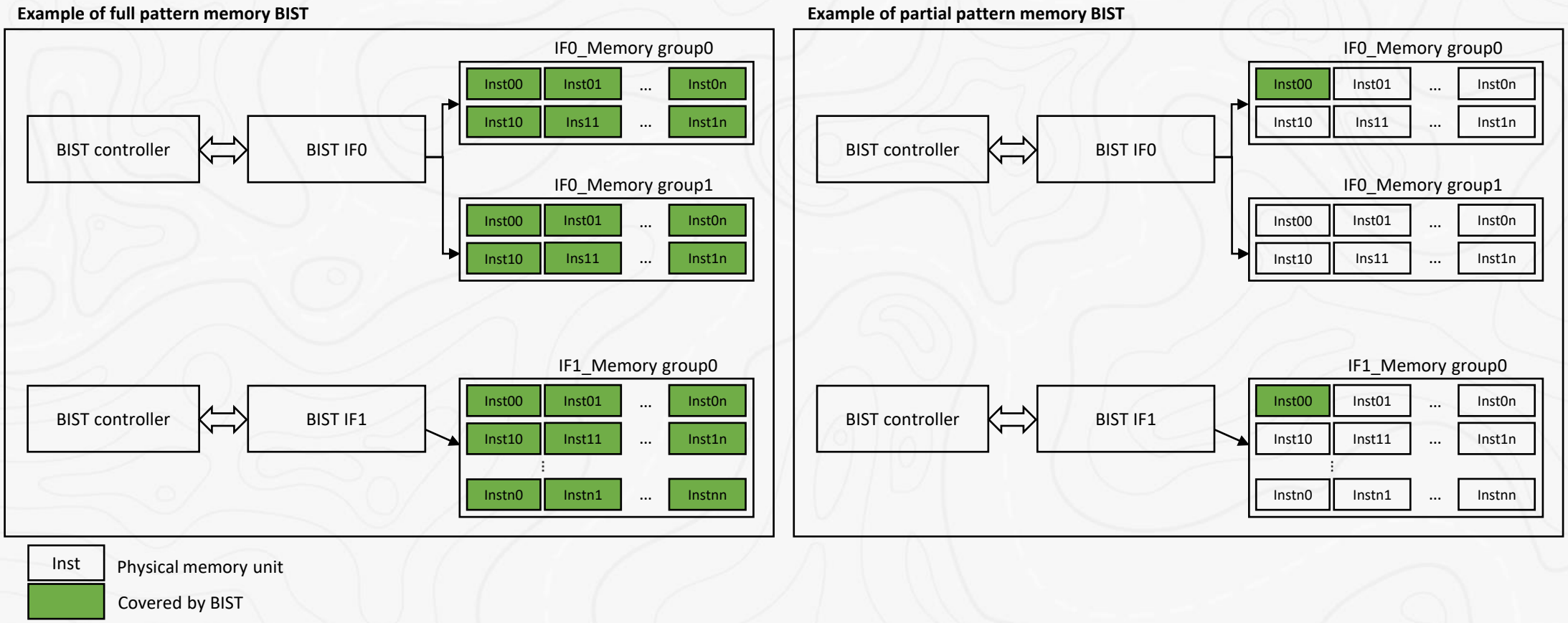
Example of Full BIST Operation Sequence

- System operation phases vs. BIST operation



Conventional Method for Reducing Simulation TAT

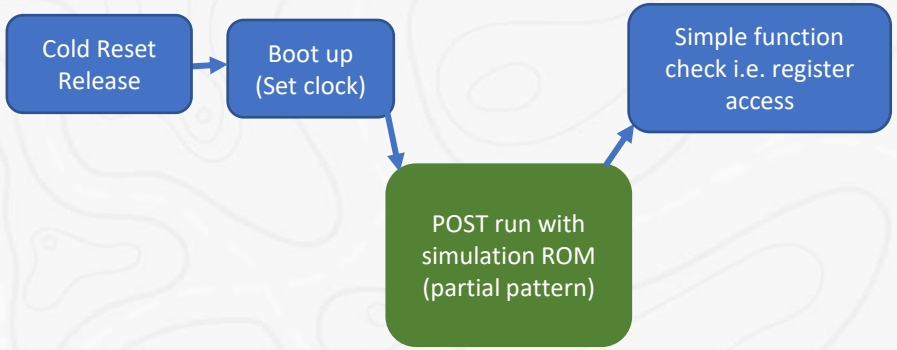
- Cut-down BIST patterns



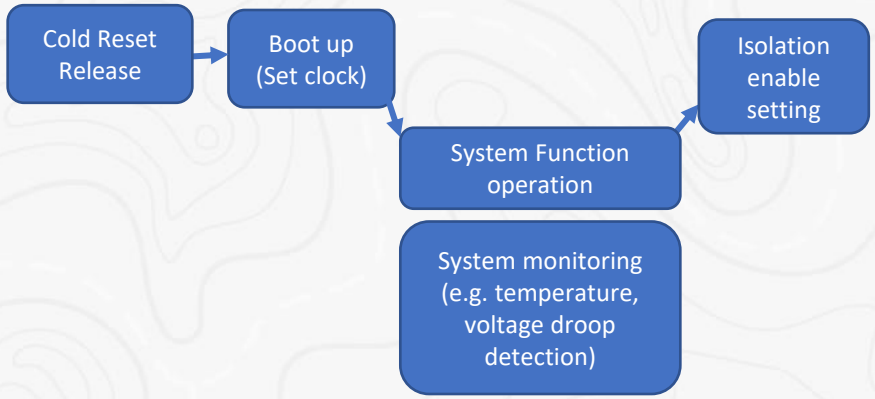
Conventional Method for Reducing Simulation TAT

- Cut-down use case scenario

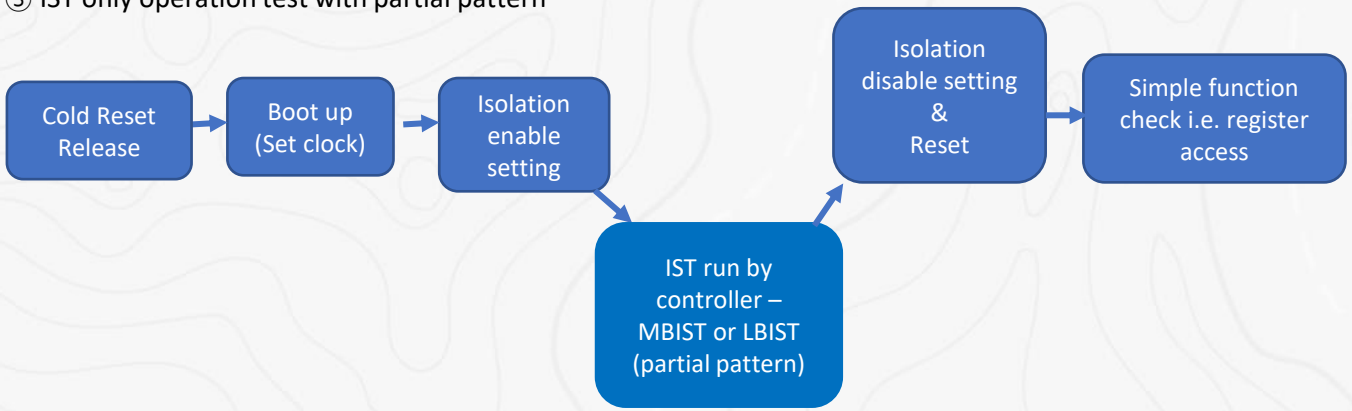
① POST only operation test with simulation ROM



② Mission mode operation for BIST target



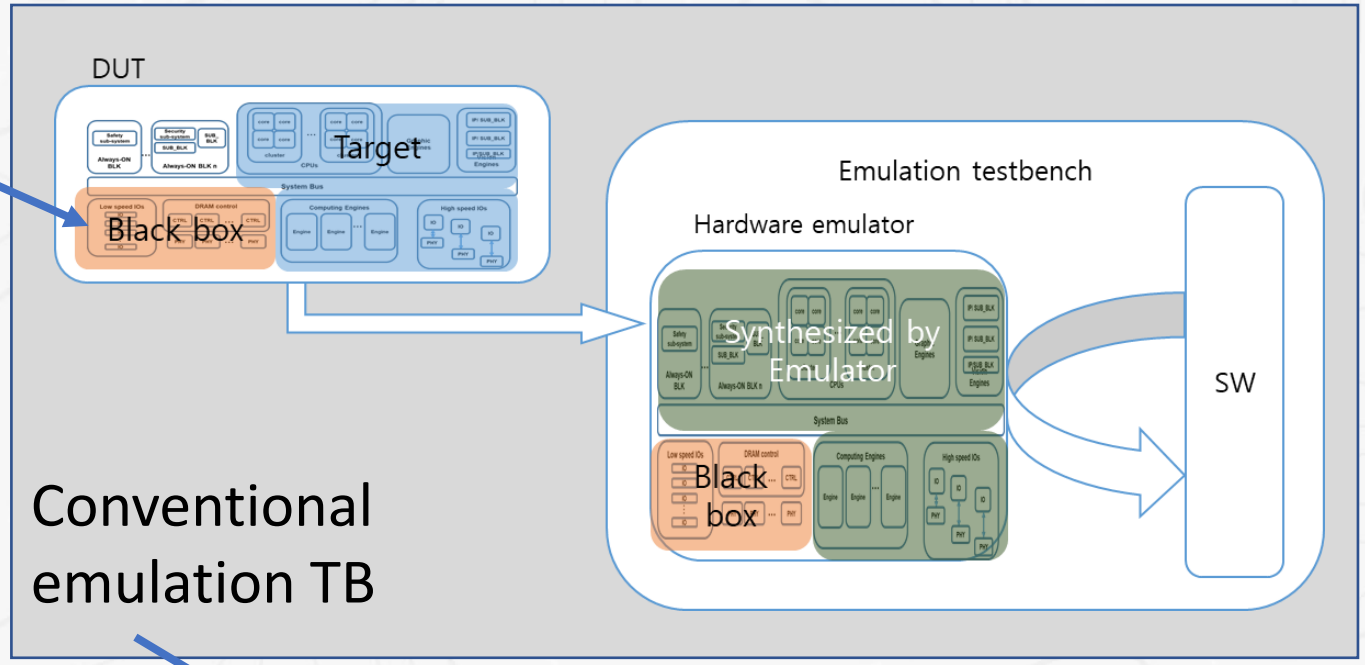
③ IST only operation test with partial pattern



Simulation Acceleration using Emulator

- Mapping an entire design into a hardware platform and enable to run the test at its full speed

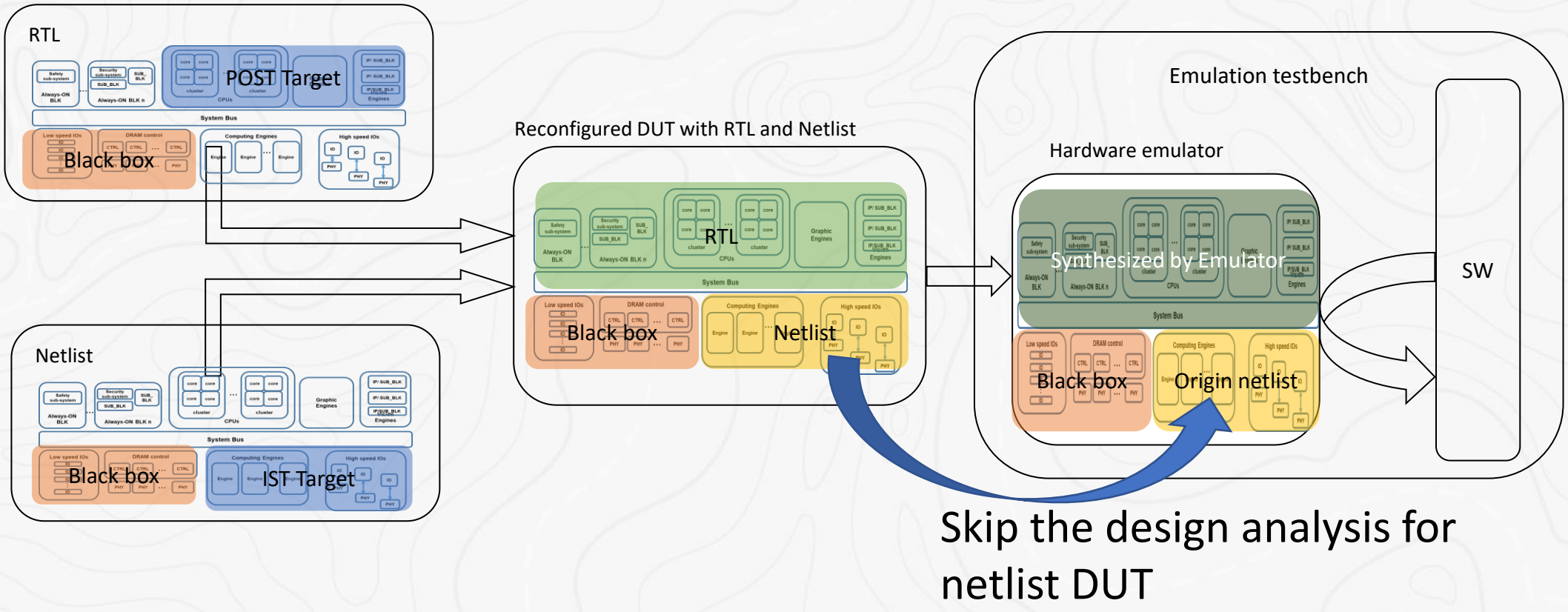
- Hurdles to use emulator
  - Limitation of HW capacity
    - Use black-box
  - Redundant design synthesis
  - Low debugging capability



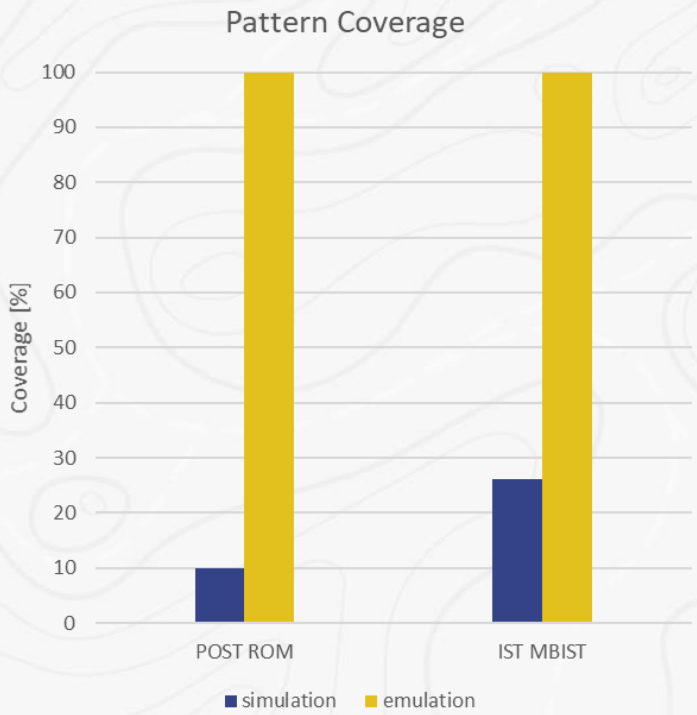
Not suitable for netlist DUT

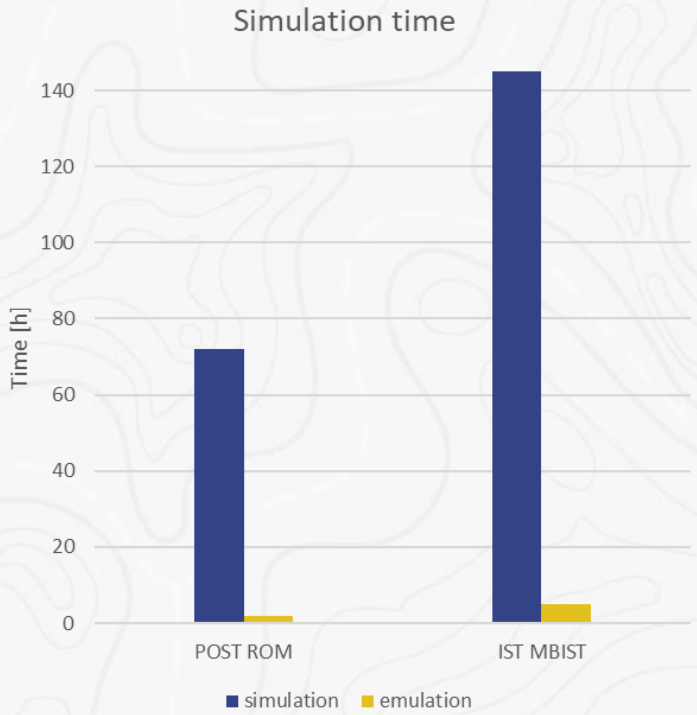
Proposed Emulation Testbench

- Reconfigure DUT with RTL, netlist, and black-box



Results – Simulation vs. Emulation





	Simulation		Emulation	
	Pattern coverage	Run-time [hrs]	Pattern coverage	Run-time[hrs]
POST ROM	10%	72	100%	2
IST MBIST	26%	145	100%	5
IST LBIST	1-pattern	NA	16384-pattern	48

Agenda

- Motivation
- Ensuring Verification Completeness on Low-Power Design
- Increasing Verification Efficiency for Self-Diagnosis and Recovery Design
- **Conclusion**

Conclusion

- Improve low power verification quality by auto-generated coverage metrics and assertions from specifications and combining them with the system operating modes
- Propose new emulation testbench that configured with netlist and RTL for efficient use of emulator
- Enable identifying corner cases and design issues that were undetectable during pre-silicon verification phase
- Achieving a one-time correct silicon development for a billion-gate automotive SoC

Questions ?