JULY 9–13, 2023

MOSCONE WEST CENTER
SAN FRANCISCO, CA, USA

Divide, Conquer, Upscale!

Conquering the challenges in Formal for SoCs

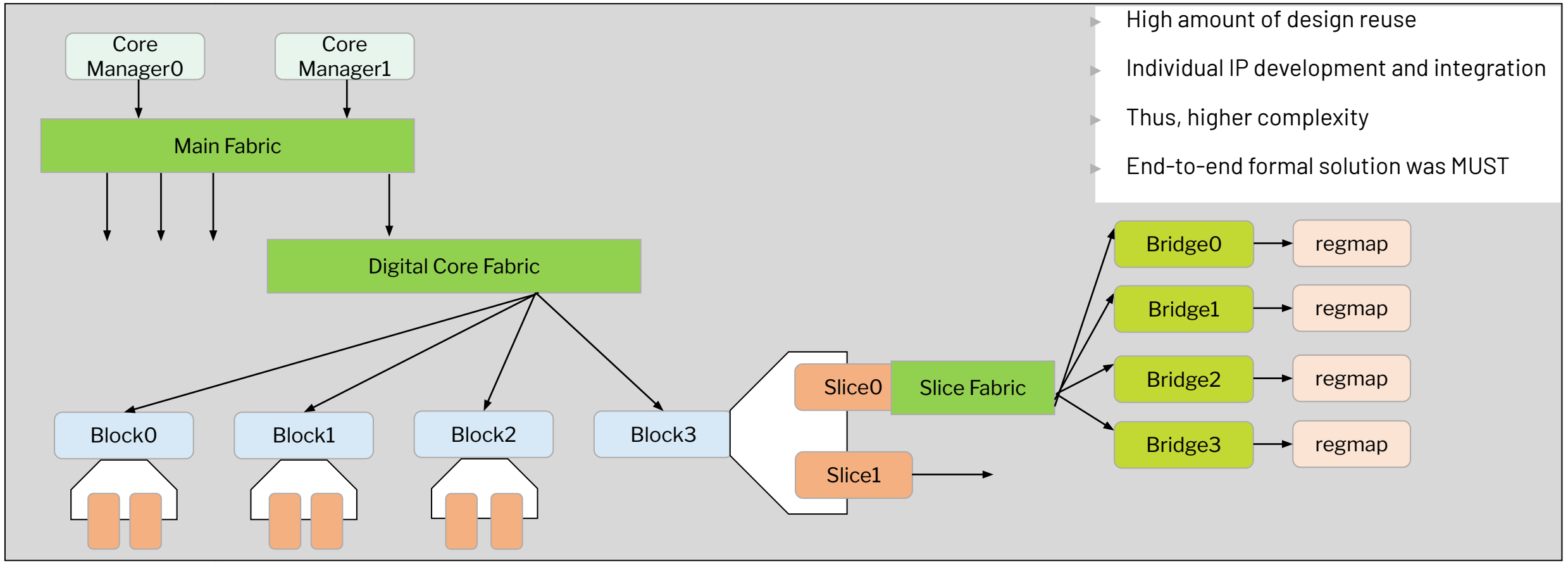
ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

Agenda

- Project Background
- Challenges
- Need for optimizations at SoC level
- Methodology Development
- Results & Conclusion

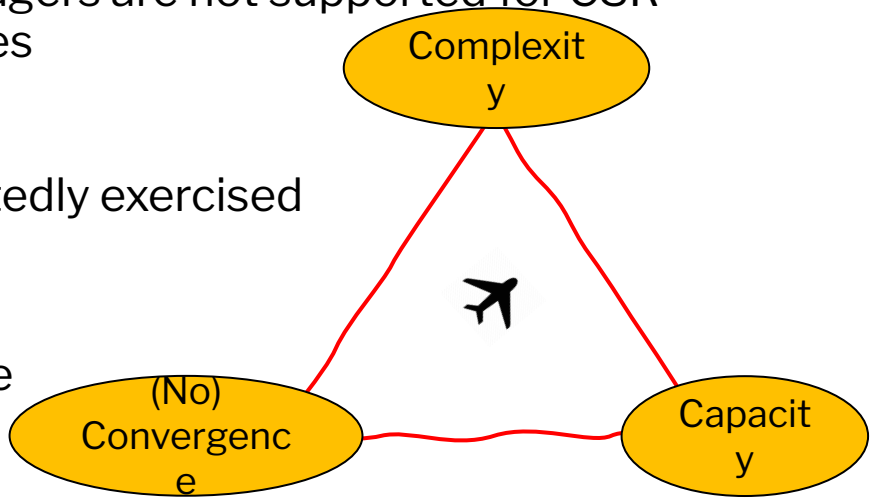
Project Background



- High amount of design reuse
- Individual IP development and integration
- Thus, higher complexity
- End-to-end formal solution was MUST

Upscaling the formal usage to SoC level

- **Assumptions**
  - Proofs will have reasonable depths
  - ‘Scaling’ of the formal efforts will be proportional to design size from blocks to SoC
  - It will work ‘right out of the box’!
- **Challenges**
  - Common to have multiple managers at SoC level - Multiple managers are not supported for CSR
  - Thousands of registers to be tested - ran into tool capacity issues
  - Convergence issues due to sequential depths and FIFOs
  - Too large COI for top-to-leaf cell
  - Inefficient formal use as access paths through design are repeatedly exercised
- **Results post first attempt**
  - Proofs did NOT have reasonable depths
  - ‘Scaling’ of the formal efforts is NOT proportionate to design size
  - Complexity and capacity issues
  - It did NOT work ‘right out of the box’!



Need a methodology to reasonably scale formal efforts to bigger designs

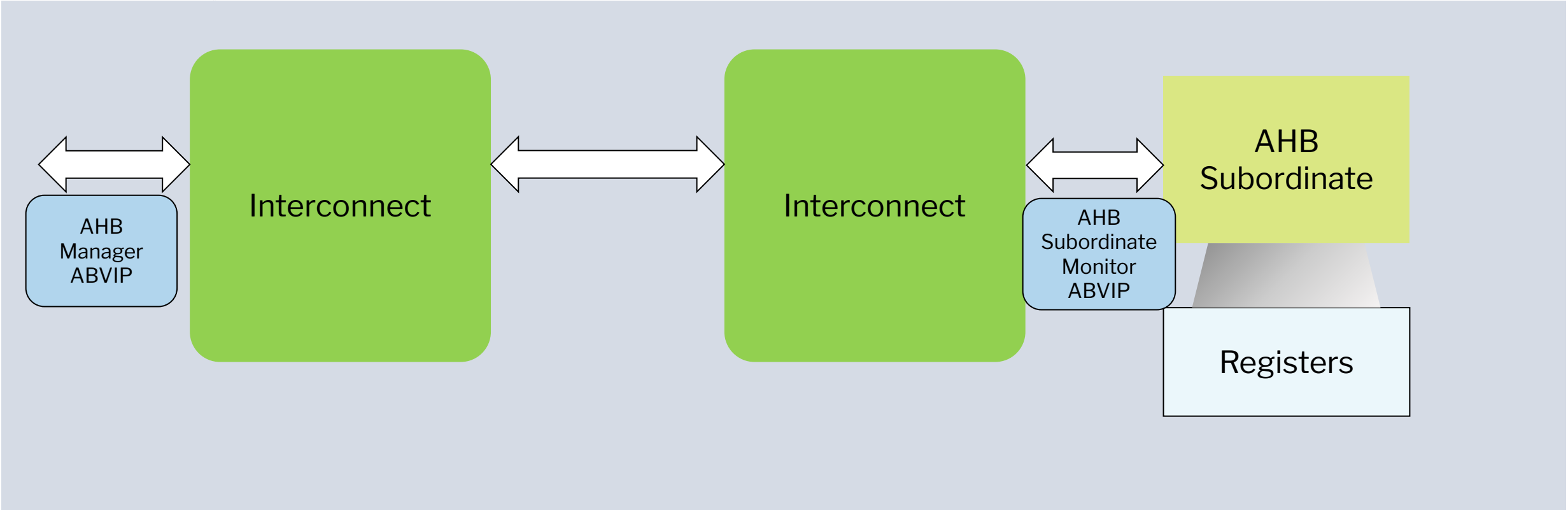
Divide and Conquer

Action	Rationale
• **CSR at the leaf cell level** • Verify register access and interaction	• Full proofs obtained at leaf level are valid checks for accesses • Full convergence achievable

• **Interconnect scoreboards**
  • ABVIPs for protocol checking
  • Custom assertions for arbitration
  • Scoreboards for data connection and integrity
	• Ensures that read and write requests reach target module and read data is returned
	• Efficient to do access checks only once in dedicated manner

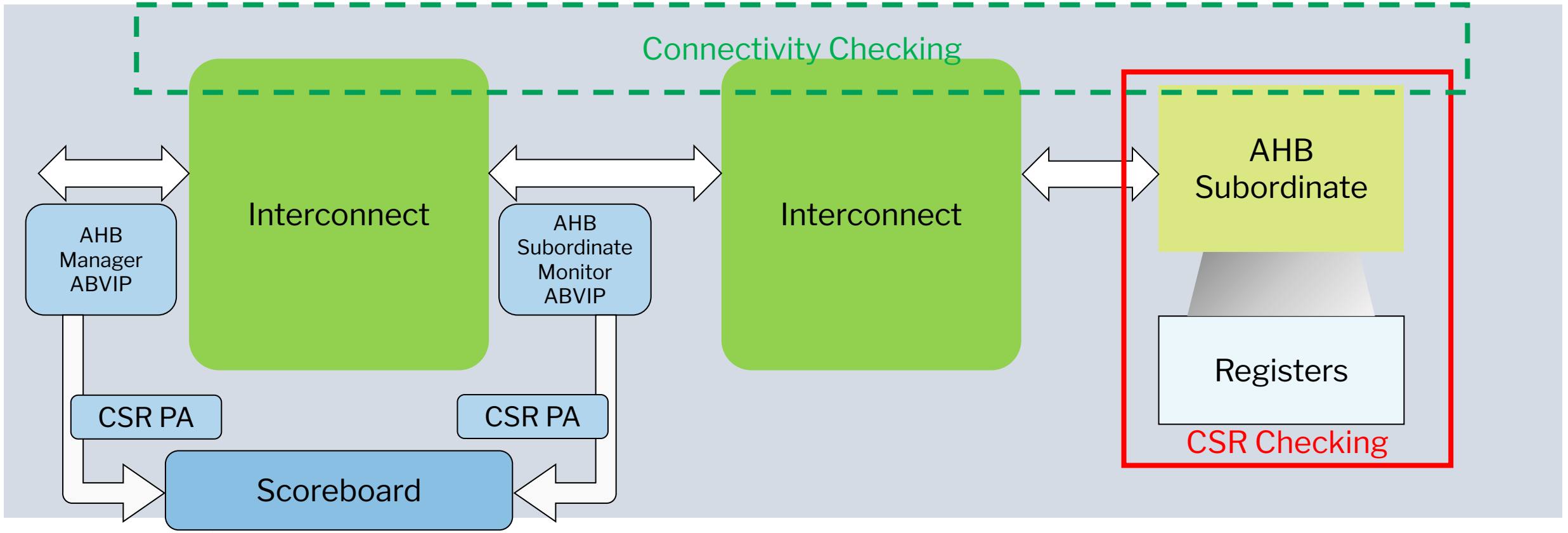
• **Connectivity checking**
  • Checking wiring between fabrics, managers & leaf cells
  • Catching bus width mismatches
	• Routing and connection checks for data path
	• Conditional connectivity is checked

Methodology modifications



Traditional CSR and Interconnect/Bus-matrix Verification Flow

Methodology modifications



Optimized CSR and Interconnect/Bus-matrix Verification

[Repeat for each fabric configuration individually]

Flow

Need for automation

- Tasks for the scoreboard-based checking:
  - Scoreboards should be checking paths from each manager to subordinate
  - Need customization to each interconnect fabric's address map
    - Not every manager is connected to every subordinate
    - Some subordinates respond to multiple address ranges
    - Multiple protocols required custom connections for each client
- Manual instantiation and connections are error prone
  - **Needs an automated solution!**

Automated Solution

Type	Property	SVA	Description
Enabled Pairs	data_integrity	Assert	Data integrity issues between the incoming and the outgoing data
	no_overflow	Assert	The maximum number of pending transactions is respected
	latency_check	Assert	An incoming transition generates an outgoing transaction
	cover[n].data_in	Cover	Witness n incoming transactions
	cover[n].data_out	Cover	Witness n outgoing transactions
Disabled Pairs	disabled_manager_subordinate	Assert	Specified manager and subordinate cannot have valid communication
	disabled_addr_range	Assert	Specified manager cannot send an access request to address of specified subordinate
	single_manager_rd_valid	Assume	Only one manager active while reading
	single_manager_wr_valid	Assume	Only one manager active while writing

Automated Solution

Set Tcl variables required by the flow

(Investigate in the specification or RTL what information to declare)

```
% set interconnect_check::addr_width    32
% set interconnect_check::data_width    32
% set interconnect_check::bind_target   "wrapper"
% set interconnect_check::cover_count   "4"
% set interconnect_check::bind_clk      "clk"
% set interconnect_check::bind_rstn     "rst_n“
% set interconnect_check::read_order    “subordinate_first“



% set interconnect_check::set_disabled_pairs "manager0
subordinate0"



% dict append interconnect_check::manager_map "manager0"
{instance
"ahb_manager[0].ahb5_lite_master.gen_csr_checker.inst"}

% dict append interconnect_check::subordinate_map
"subordinate0"  {instance "ahb_subordinate
[0].ahb5_lite_slave_monitor.gen_csr_checker.inst" offset
"'h47000000" offset_end "'h470fffff"}
```

Addr and data width of mgr/sub

To what instance should the FE bind?

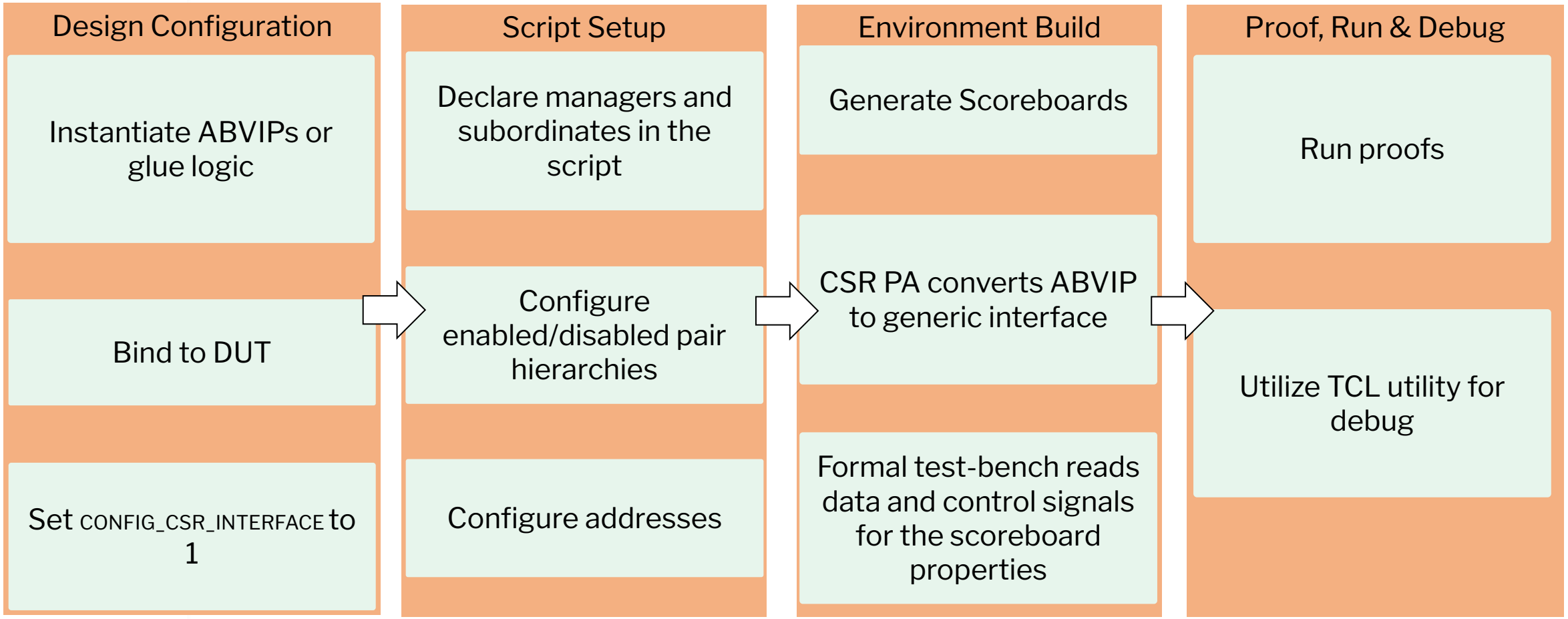
How many times should we witness data in/out?

To what clock and reset should the FE follow?

From whom should the read data come?

Which pairs are disabled?

Execution Flow



Limitations in current flow

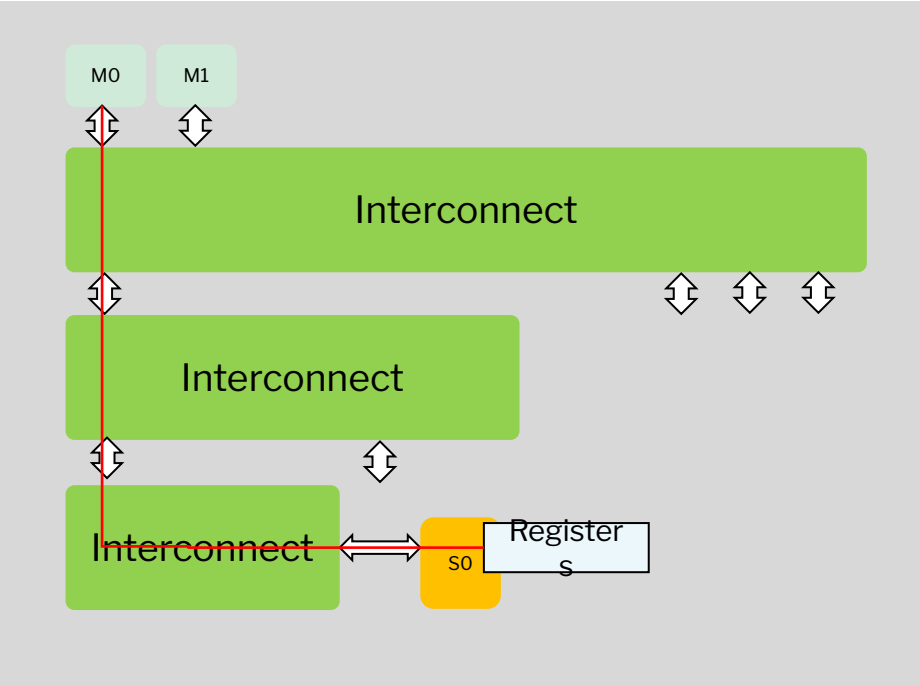
- Managers and subordinates must have CSR interface so currently supported out-of-the-box for AMBA ABVIPS (Formal VIPs)
- jasper_csr_extended_checker and custom glue logic required for other protocols
- Data and address size must be consistent in the manager-subordinate pairs
- Custom properties required when transactions travel through protocol translation bridges

Results

Project A

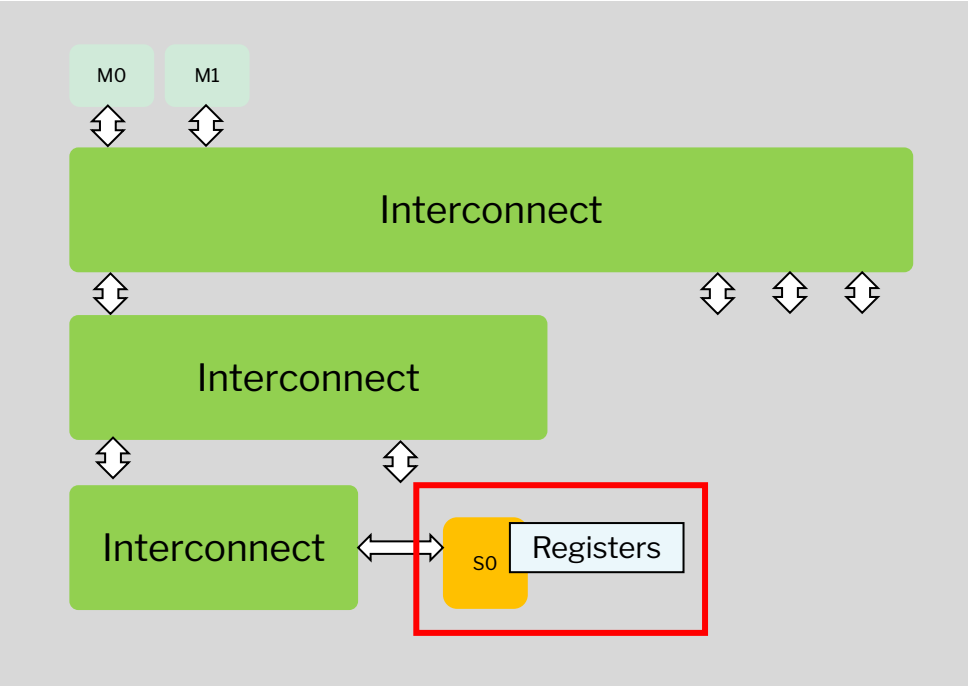
End-to-end CSR:

- 200k+ total CSR properties
- 100+ cycle deep for single write
- No convergence
- 30+ cycles through each interconnect and FIFOs



Leaf level CSR:

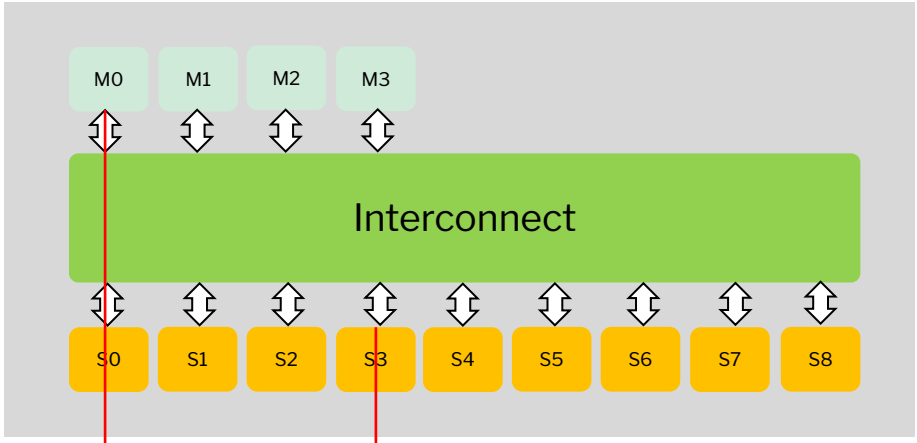
- ~1500 properties per leaf node
- 1 minute total proof time
- Read followed by write was 4 cycles deep



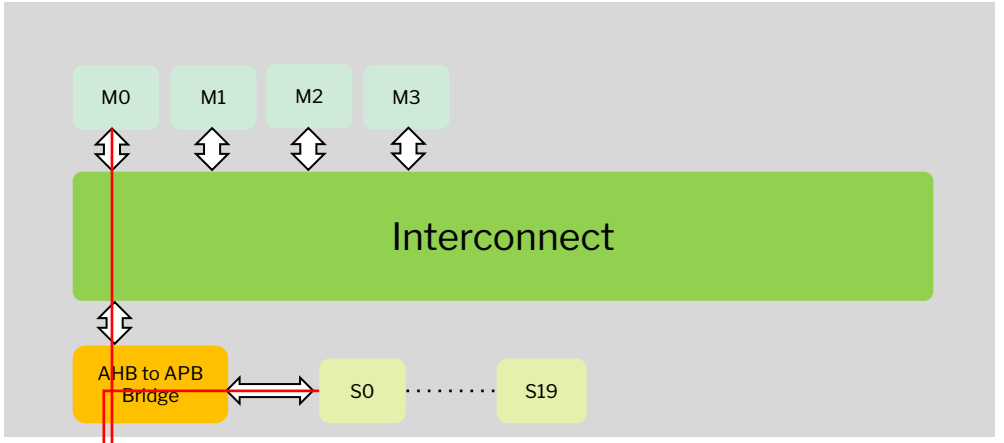
Results

Project B

AHB Interconnect:
- 4 Managers 9 Subordinates
- 864 total properties
- 1.5hrs to prove



AHB-Bridge-APB Properties:
- 4 Managers 20 Subordinates
- 3840 total properties
- 13hrs to prove



- Bug Found:
  - Mis-aligned address ranges for the subordinates
  - Design not in sync with specifications

Conclusion & Next Steps

- Prototyped on simple interconnect block
- Experimented on a section of the complex SoC design
- Verified end-to-end transaction integrity through SoC interconnect in automated way
- Propagated and applied learnings on 2 new project designs
- Significant **reduction in sequential depths** for CSR properties
- Adoption of the interconnect scoreboard flow and automation is helping in **exhaustive interconnect verification**
- **Reduction in manual efforts** to create signoff properties for exclusive accesses
- Next Steps:
  - Enable coverage collection on the enhanced flow
  - Enable connectivity checks to improve robustness and earn toggle coverage

Acknowledgements

- Tom Weiss, Cadence Design Systems
- Murilo da Silva Santos, Cadence Design Systems

- Sivasubrahmanya Evani, Analog Devices Inc
- Amey Chindarkar, Analog Devices Inc
- Rufa Leninkumar, Analog Devices Inc
- Nimay Shah, Analog Devices Inc
- Lee Anthony Grajo, Analog Devices Inc
- Ponnambalam Lakshmanan, Analog Devices Inc

Thank You!

Questions?

Title Here

- Edit Master text styles
  - Second level
    - Third level
      - Fourth level
        - dfgdfFifth level