Enhancing Quality and Coverage of CDC Closure in Intel's SoC Design

Rohit Kumar Sinha

accellera SYSTEMS INITIATIVE

Overview: CDC Quality and Coverage

Power Aware CDC Analysis & Metasim

Analysis needs to be done post instrumentation of low power cells

Handling parameters flavors

Different flavors of parameters- Width and Modal

Multi-Mode Analysis

Multiple functional, DFT and DFD modes of operation Analysis

Design Independent Quality

Ensure CDC collateral is in correct and consistent

Waiver to Constraints

How to capture design intent with correct constraints instead of waivers

Constraints Validation

SVA Protocol Validation

accellera SYSTEMS INITIATIVE

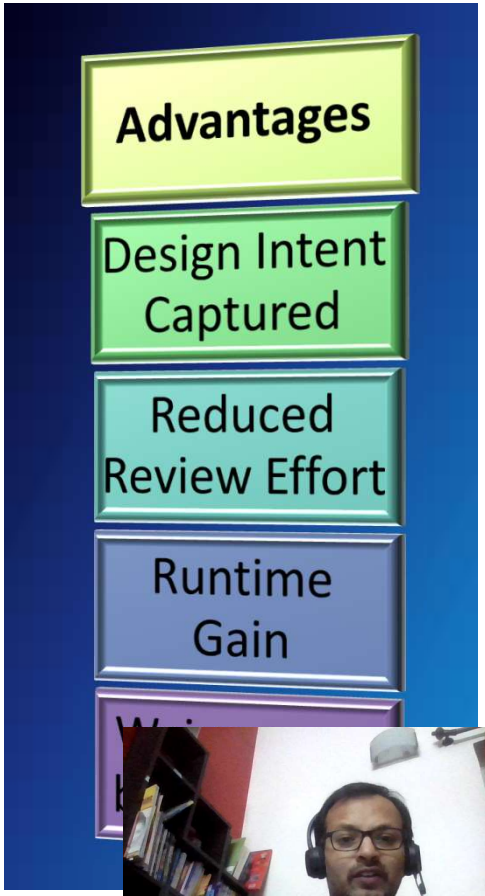
Design Independent Quality Check

- SS/SoC consume abstract for IP/SS
- Late Quality Issues finding post integration cause long iteration
- Debugging at the consumer end creates longer loop
- DIQC ensures that incorrect collateral

```
abstract_block_violation -name SGDC_fifo12 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_fifo12 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_fifo13 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_fifo13 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_fifo14 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_clockreset03 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_clockreset03 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name Param_clockreset06 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name Param_clockreset08 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_cdc_define_transition01 -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name SGDC_cdc_glitch_end_point -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name SGDC_cdc_glitch_end_point -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name SGDC_meta_design_hier01 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_meta_monitor_attributes01 -sev WARNING -count 0 -waived_count 0 -i
```

CONVERSION: Waiver to CDC Constraints

Type of Waivers	Translation Into Constraints
Stable and non-glitch prone signals	Quasi_static
Pulse extender in the crossing path	Clock_relation
MetaFlop in the crossing path - enable_multiflop_sync = yes (sync_cell)	- enable_multiflop_sync = no, add synchronize_cell “instance_name”
Debug modules (VISA & IDV) network signals not Impact on CDC(crossings between test clock & functional clock are waived) - set_clock_groups	Set_clock_groups
Xover in the crossing path - ???? Clock_relation (posedge/negedge)	Clock_relation
Signal going to Power control unit -	Quasi_static
clkack/clkreq are safe - ????	Handshake protocol; qualifier -enable
PwrGood signal is stable.	Quasi_static
Rx samples the signal once Tx settles down - qualifier can be used	Data_hold_check
There is no activity/transactions happening during the time of reset - After the reset deassertion, clock is cut off because of the gating logic.	Reset desertion; reset_filter_path
Registers in bypass mode	Quasi_static
Both TX and RX clocks are aligned.	Clock_relation
Going to config register that is polled by SW	Quasi_static
Mutually exclusive clocks	Set_clock_group
Enable signal asserts long before the valid data is accumulated	qualifier
initial stage mux clocks won't be running or gated during reset de-assertion	Quasi_static_rdc
As per usecase, the d input of the flops will be stable during reset deassertion and will have the value same as reset value	qualifier –src_stable



Handling Reset and Reset Sequences

Two Main Issues

- Issues related to the reset distribution tree
- Issues related to the reset usages

Reset Definition

- Master Clock Spreadsheet
- MicroArchitecture Specification Document

Reset Order Sequence

CDC owner often handle RDC through reset assertion sequences. There are multiple ways to address this

- Define_reset_order to define the order of reset assertion sequence
- Reset_filter_path –type rdc –from * -to * to filter such reset crossings

accellera SYSTEMS INITIATIVE

Multi-Mode CDC Analysis

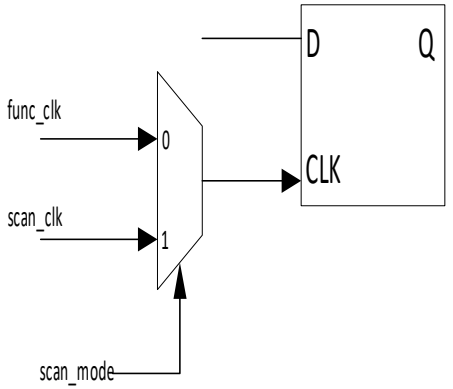
Traditionally CDC analysis is done assuming each sequential receives a single clock/domain

Just like timing analysis is done on MMMC

If there are clock muxes with multiple clocks/domains, user needs to choose 1 mode for analyzing their design. However, silicon is tested and used in many other modes.

This is extremely important for the SOC teams where multiple chips are churned out with different use cases with little/no modifications.

The effort is to bridge the gap between validation and Si usage



- Multi-modal or Modal analysis the methodology though which we analyze the CDC for different Modes, Use cases or various clocks / domains.
- Typical example is for a design, betw and DFT

accellera SYSTEMS INITIATIVE

Handling Different Flavors of Parameters

Parameters categorized into 2 flavors

1. Parameters that impacts CDC analysis
2. Parameters don't impact CDC

Parameters mainly for the BUS width don't impact CDC Analysis

No new abstract model or hierarchical data models required

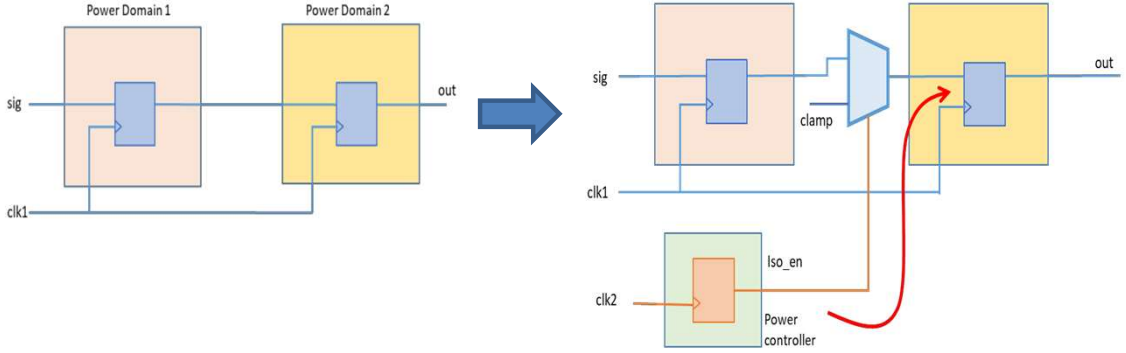
In the SoC run, based on the SoC configuration keep changing. This requires new CDC IP collaterals causing long iteration

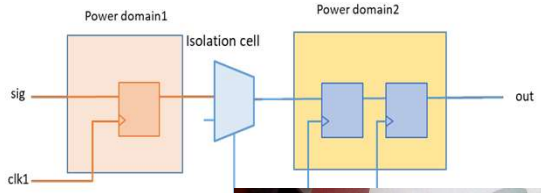
```
Number of DNC RTL parameters                                    : 5
---------------------------------
S. No.      Param Name
---------------------------------
   1.    ABUTWIDTH4
   2.    BROADCAST_EN_WIDTH
   3.    WIDTH2
   4.    WIDTH4
   5.    WIDTH8
---------------------------------


Number of NON-DNC RTL parameters                                : 10
---------------------------------
S. No.      Param Name
---------------------------------
   1.    DP_EN
   2.    FAST_SIMMODE
   3.    GBE_EN
   4.    MIPI_EN
   5.    OUTPUTFLOPPED
   6.    PCIE_EN
   7.    SATA_EN
   8.    SERDES_EN
   9.    TBT_EN
  10.    USB_EN
---------------------------------
```

Power Aware CDC Verification

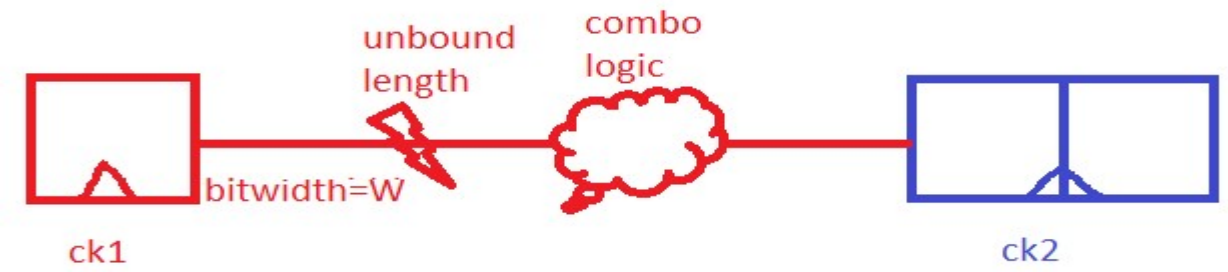
- In absence of UPF during CDC analysis, Power control logic is unconnected in RTL
- CDC analysis on RTL will not verify power control logic
- In presence of UPF with CDC analysis has additional artifacts
  UPF specifies power domain, isolation strategy, retention etc.
- Addition of UPF introduces new CDC paths, or break existing CDC synchronizers
  Data loss or data corruption.
  Intermittent CDC errors



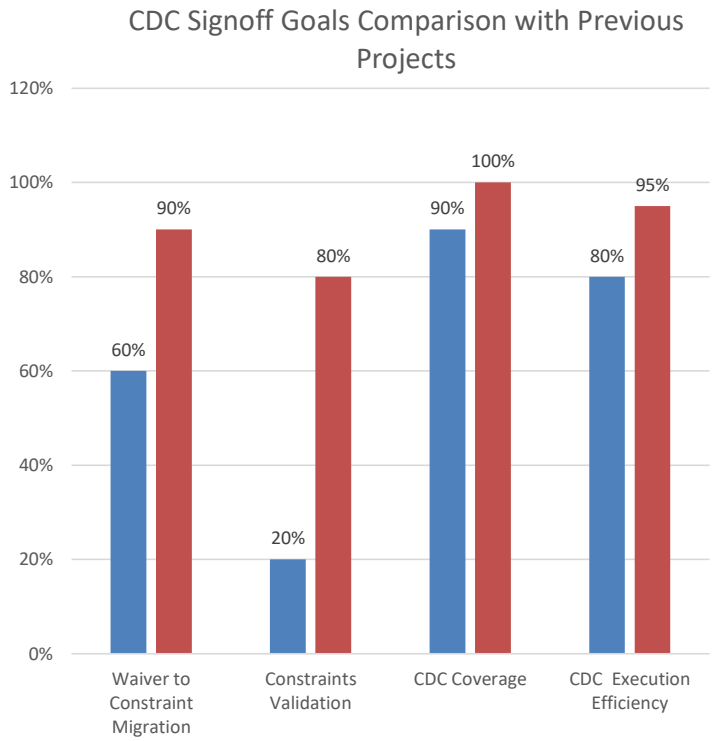


Random Delay Simulation

- Dynamic CDC verification using metastability injection (CDC jitter) mechanism during simulation is mandatory
- Fault injection can be difficult to converage because of debuggability
- A dynamic CDC jitter verification solution generates better control signal jitter injection models for CDC convergence
- Non-deterministic (synchronizer output can occur N+/-3 cycle of ck2 after input)



Conclusion & Summary



Methodology	ROI
Design Independent Quality Check	3-4 weeks of schedule
Handling the waivers through constraints to capture design intent	30% runtime reduction
Developing a mechanism to validate CDC constraints through SVA protocol validation	Quality Enhancement
Validating reset and reset sequences using SVA	1-2 weeks of schedule
Enabling multiple mode CDC flows for better coverage	Quality Enhancement
Handling different flavors for design parameters to avoid reiteration	3-4 weeks of schedule
Power Aware CDC for enhancing quality coverage wit implementation issues	
Random Delay Simulation for Non-Deterministic Path	

Questions

Finalize slide set with questions slide