2020 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

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

Rohit Kumar Sinha, Intel India, Bangalore, rohit.kumar.sinha@intel.com

***Abstract*— In the non-Intel ARM based architecture SoC design, it is often challenging to ensure that all the asynchronous design challenges are covered with utmost quality while keeping the schedules on track. Since in the non-IA architecture design, all the IPs are mainly sourced from external vendors and hence there is no standardized TFM which ensures the quality of the CDC or RDC closure at the SoC levels.**

As a result of this, late design cycle bugs often occur in the SoC design and at times it costs an entire respin due to meta-stability issues or due to glitches in the clock-reset paths. Therefore, in order to handle challenges due to multiple vendor and multiple TFMs in the SoC design integration, there is an absolute need to revamp the CDC signoff methodologies with series of initiatives which would ensure zero Si escapes in the design.

Keywords—SoC; CDC; Architecture; RDC, Metastability, TFM, ARM, Quality, Efficiency

I. INTRODUCTION

In the SoC Design, while integrating the external IPs, internal IPs, SoC developed IPs, global IPs, there are multiple challenges in terms of the correctness and completeness of CDC closure. In the ongoing ARM based SOC design, IP team often signs off their design using their own TFM and it is nearly impossible for the SoC team to expect from IP vendor- either internal or external to follow the TFMs as per the SoC guidelines. As a result, SoC integration quality sign off becomes very challenging and chances of wrong constraints being used in the CDC closure is quite high. Below is the list of methodology initiatives that were deployed in the ongoing design which ensures high coverage in CDC closure and also the improved CDC closure efficiencies. Also, below table depicts the comparison with the previous projects and the ROI because of each methodology improvements.

Methodology Initiatives	Old Projects	Current Projects	ROI
Design Independent Quality Check	Partial	Enabled	3-4 weeks of schedule
Handling the waivers through constraints to capture design intent	NA	Enabled	30% runtime reduction
Developing a mechanism to validate CDC constraints through SVA protocol validation	NA	Enabled	Quality Enhancement
Revamping reset and reset sequences definition mechanism	Manual	Enabled	1-2 weeks of schedule
Enabling multiple mode CDC flows for better coverage	NA	Enabled	Quality Enhancement
Handling different flavors for design parameters to avoid reiteration	NA	Enabled	3-4 weeks of schedule
Power Aware CDC for enhancing quality coverage with implementation issues	NA	Enabled	Quality Enhancement

II. DETAILS OF THE METHODOLOGY INITIATIVES

A. Design Independent Quality Check

SS/SoC consume abstract models for IPs/SSs, which are generated during IP/SS CDC runs and IP or SS integrators who consume the CDC abstract models often miss checking the quality of the abstract model and which often leads to long iterations and debug time because of the un-necessary violation which are flagged at the SS or SoC Level.

The flow is developed to check the quality of the abstract model upfront even before the CDC setup checks are done during the RTL integration process. Using the flow, SoC designer can upfront reject any IPs which are delivered with the CDC collateral with the incorrect or incomplete CDC closure. Below is the snapshot for the collateral –

```
abstract_block_violation -name SGDC_quasi_static_style01 -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name SignalTypeSetup -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SignalTypeSetup -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_qualifier34 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_qualifier34 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_allow_combo_logic02 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name Param_clockreset07 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name Param_clockreset07 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_generated_clock06 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_generated_clock06 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_clock_sanity01 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_clock_sanity01 -sev WARNING -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name Propagate_Clocks -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name Propagate_Clocks -sev WARNING -count 22 -waived_count 0
abstract_block_violation -name SGDC_virtualclock_validation01 -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name SGDC_virtualclock_validation01 -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name FalsePathSetup -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name SGDC_fifo11 -sev ERROR -count 0 -waived_count 0 -is_builtin
abstract_block_violation -name SGDC_fifo11 -sev WARNING -count 0 -waived_count 0 -is_builtin
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 -is_bui
abstract_block_violation -name SGDC_multiple_virtual_clock01 -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name SGDC_multiple_virtual_clock01 -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name Param_SCAValidation -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_CHECK -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_CHECK -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_READ -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_READ -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_INFO -sev ERROR -count 0 -waived_count 0
abstract_block_violation -name LP_POWERDATA_INFO -sev WARNING -count 0 -waived_count 0
abstract_block_violation -name UPF_lowpower08 -sev ERROR -count 0 -waived_count 0 -is_builtin
```

B. Handling the waivers through constraints to capture design intent

One of the major initiative to eliminate the waiver mechanism to achieve the following objective

- Capturing the Design Intent
- Eliminate the noise due to handling of huge waiver commands
- Improve the CDC tool performance

In order to migrate all the waivers into the related constraints, we analyzed all the waivers which were used in the previous projects and identified the command waiver usage as below –

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

C. Developing a mechanism to validate CDC constraints through SVA protocol validation

With the complexity of design increasing day by day, there isn an absolute necessity mainly in the SoC design to ensure that the assumption used for signing off CDC or RDC design challenges are validated using an autonomous flow. Following is the approach adopted to ensure that the assumption which were translated into SVA and assertions generated out of CDC tool were validated into simulation environment. Below are essentially 3 steps to bind the SVA into functional simulation tests-

- Create a library for system verilog assertion modules and include it the design used to run Simulation Tool
- Run simulation to generate a new sim executable (*.simv)
- Run regressions using the newly generated simv to validate the constraints

D. Revamping reset and reset sequences definition mechanism

Traditionally reset design and verification are done by design engineers. They are expected to follow some standard reset architecture guidelines to avoid any potential metastability issues. However, with the advent of complex power management design flows and due to the increase in reset signaling complexity with the emergence of multiple reset domains, reset domain crossing verification becomes an absolute need to ensure glitch free reset assertions during various power states. There are essentially two common problems with the resets verification. We will separate it into two main categories:

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

To solve such problems, we defined a methodology to upfront define all the resets, its active value, domain and the reset ordering sequence in the Micro Architecture Specification Document and in Master Clock Spreadsheet

so that there are minimal waivers or post processing required. Below is the constraints that must be defined upfront

Reset Order Sequence – CDC owner often handle RDC challenges by defining the TX flop and RX flop reset assertion sequences. There are multiple ways to address this

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

E. Enabling multiple mode CDC flows for better coverage

Motivation to enable multi-mode CDC analysis is as follows

- Traditionally CDC analysis is done assuming each sequential receives a single clock/domain. 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. Hence there is a big gap and we are seeing Silicon escapes in modes not analyzed.
- Each mode is defined with unique set of constraints.
- Few users run other modes in paranoia for the IPs multiple times but provide only 1 abstract to SOC
- The effort is to bridge the gap between validation and Si usage and enable users to catch issues that are otherwise difficult and costly to find during Post-Si debug

Use Case – There are 4 use case / operational modes in HSIOSS. PCIe / SATA controller will use common PHY for the communication.

<table>
<tr><th>Mode</th><th>Lane0</th><th>Lane1</th><th>Lane2</th><th>Lane3</th></tr>
<tr><td>0 - SATA Only</td><td colspan="4">SATA X4 Controller</td></tr>
<tr><td>1 - PCIe RP + SATA</td><td colspan="2">X2 DM Controller (in X1 or X2 RP Mode)</td><td colspan="2">SATA X4 Controller (using 2 ports)</td></tr>
<tr><td>2 - PCIe RP + PCIe EP</td><td colspan="2">X2 DM Controller (in X1 or X2 RP Mode or X1 or X2 EP Mode)</td><td>X1 RP Controller</td><td>NA</td></tr>
<tr><td>3 - PCIe RP (or) PCIe EP</td><td>X2 DM Controller (in X1 DM mode can be configured as X1 RP or X1 EP)</td><td>NA</td><td>NA</td><td>NA</td></tr>
</table>

Following are the benefits to enable Multimode CDC analysis

- Minimum performance issues with the modal analysis run since the per mode CDC runs are parallelized
- Top-Level can parallelly consume abstract models for different operations for the block
- CDC signoff Coverage can be greatly enhanced
- Cross-functional domain runs can be performed in the single CDC run

F. Handling different flavors for design parameters to avoid reiteration

For parameterized IPs, there may be some RTL parameters, which based on their usage in the design, cannot impact the CDC hierarchical abstract model generation. In such cases, doing the abstraction for multiple values of these parameters will not be required since the generated abstract model would always be the same.

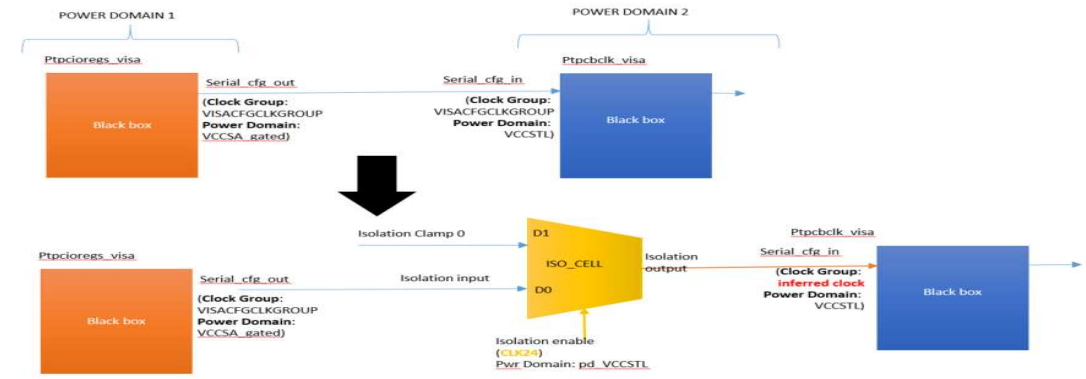
The idea here is to identify such parameters, henceforth referred to as DNC parameters (Don't Care) so that we can optimize our Hierarchical CDC flow so that for every change in the parameter as per the SoC configuration would not mandate IP team to provide the new CDC collateral with the updated parameters.

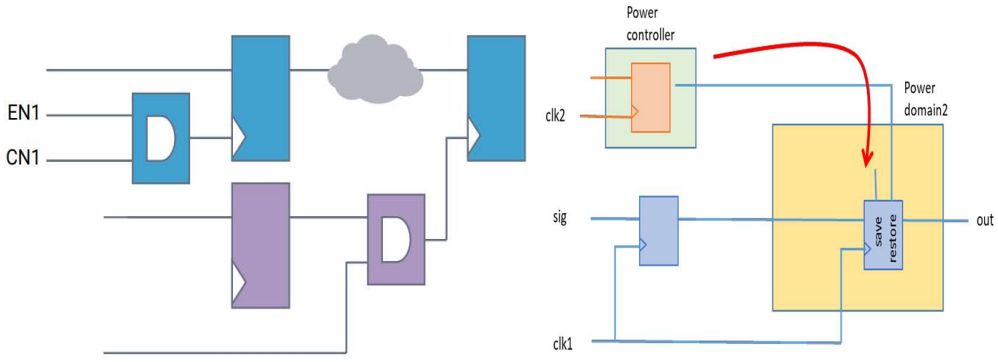
With the DNC flow, CDC tool would generate the list of the parameters that are critical for the CDC analysis and also list of parameters which are don't care. Report is as below -

```
  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
-----------------------------------
```

G. Power Aware CDC for enhancing quality coverage with implementation issues

CDC verification needs to be performed on an integrated design that captures power intent, it leads to creating new CDC paths mainly because of VDCs (voltage domain crossings) for which synchronizers must be added. The presence of power components poses new challenges for CDC verification and therefore, front end CDC Verification must be made power aware to avoid such risks since low power cells are not present at the RTL phase.





Performing power aware CDC verification at an early stage, such as RTL, identifies and enables designers to fix issues that would have been otherwise discovered post synthesis. Our validation on the actual design suggests that power aware verification with advanced technique can help close the CDC verification much faster and save from unexpected chip killing CDC issues.

III. RESULTS

Below are the impacts of each of the initiative

A. Design Independent Quality Check

- Overall savings of 5 weeks of integration cycles due to reduction in the iterations due to low quality abstract model
- At least 5 abstract models in current SoC were fixed even before CDC_SETUP_CHECK was started SG_CDC_Presetupcheck is independent of the RTL reading
- Abstract_SGDC11 is enhanced to capture all the detailed related to rules, goals and parameters
- Spreadsheet is added to the rules Abstract_SGDC11 to capture each Abstract Model and its ruleset

B. Handling the waivers through constraints to capture design intent

- Overall impact was update 60% reduction of respective constraints
- Design Intent were correctly captured using constraints which makes review much easier and faster
- TAT was reduced by 3 weeks

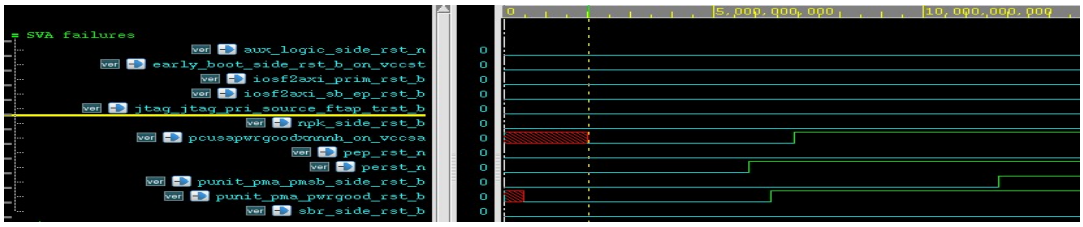
C. Developing a mechanism to validate CDC constraints through SVA protocol validation

- Issues related to reset filter paths were detected which would have slipped through the design reviewed
- Issues related to wrong set_case implementation for the respective modes were caught that could lead to late finding of CDC issues
- Issues related to quasit static signals were detected using SVA protocol validation
  For ex- issues reported because of reset file paths definition

Assumption failure (constraint 'reset_filter_path' at /nfs/sc/disks/tbh_rtl_010/rohitks/cpuss-tbh-a0/verif/tests/static_checks/*_cdc/cpuss/cpuss.sgdc:910): Reset cpuss.par_noc_cpuss.par_noc_north_cpu01.cpuss_cpr_wrap.cpuss_cpr.cpuss_cpr_tap_ovrd.tap_mux_dbg_rst_n.o asserted after cpuss.par_noc_cpuss.par_noc_north_cpu01.cpuss_cpr_wrap.cpuss_cpr.cpuss_cpr_tap_ovrd.tap_mux_a53_ncpuporeset_1.o

D. Revamping reset and reset sequences definition mechanism

Wrong reset ordering detected in the functional simulation leading to RDC issues



Some of the additional issues uncovered

- Wrong Reset Propagation
- Power Control logic interpreted as Reset
- Reset is always active high
- 2nd Flop with no set/reset pin
- Reset wrongly used as reset pin
- Set to Set domain crossing not synchronized
- Set to Reset domain crossing not synchronized

E. Enabling multiple mode CDC flows for better coverage

- Using multimode analysis, all the potential modes were covered in the single run
- Now CDC is being checked with functional mode, misson mode, debug mode, VISA mode etc.
- More of a mindset change, to be taken as seriously as Timing Analysis
- Easy to analyze the design for each of the modes independently
- Easily deployable in SOC projects
- Less chances of silicon escapes
- Modal analysis is parallel in nature, hence no performance hit

F. Handling different flavors for design parameters to avoid reiteration

With the DNC flow, CDC tool reports the list of parameters which will impact or not impact on CDC analysis. For ex-

DNC will be printed in the below format,

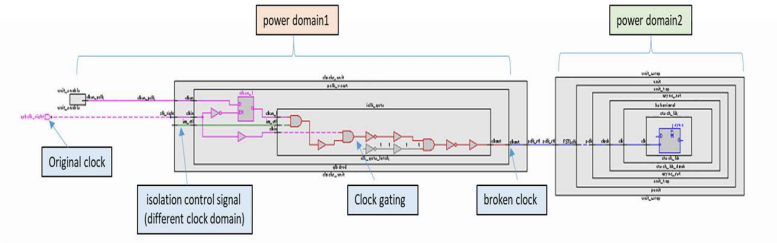
```
if { $::sg_use_cdc_abstract_view == 1 }  {
 abstract_file -version 5.1.0 -scope cdc

 current_design "block" -param { N=_dnc_ }
```

CDC-detailed-report.rpt:

```
Number of DNC RTL parameters                    : 1
--------------------------------
 S. No.      Param Name
--------------------------------
 1.    N
--------------------------------
```

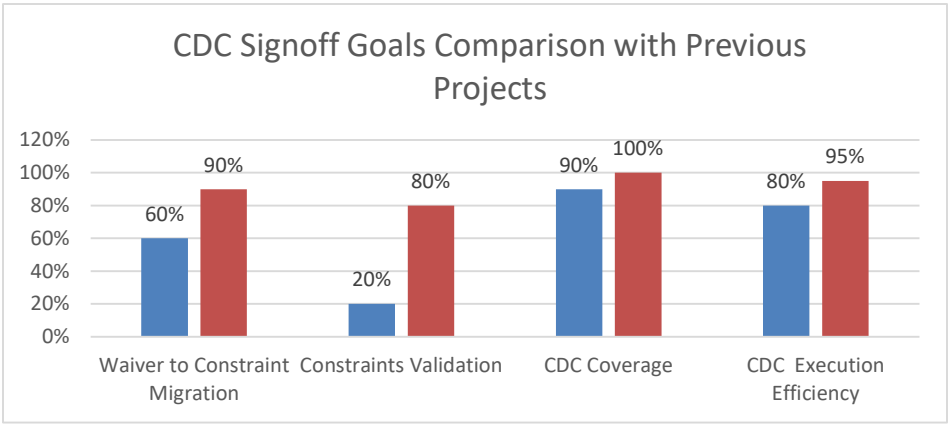
G. Power Aware CDC for enhancing quality coverage with implementation issues

Figure below shows the example of a broken clock issue found in the design. A clock was originally gated with latch-based clock gating. When the clock was crossing the power domain, an isolation control signal was added. Due to the isolation control signal, an additional gating signal was created in the clock path when it reaches to the second power domain, and became prone to glitch.



IV. SUMMARY

The mentioned methodology initiatives were developed and deployed in the Intel's latest SoC ARM based design. The SoC consists of 10 SubSystems with more 200 internal and external IPs. There were many benefits in term of enhancing CDC quality signoff as well as improving the TAT. In addition to these, we were able to achieve higher coverage in terms of left shifting the asynchronous design issues which were otherwise only would get caught in the functional verification or implementation domain. Below fig depicts the overall benefit over the previous projects in terms of Quality improvement and execution efficiencies.



ACKNOWLEDGMENT (STYLE: HEADING 5)

Thanks to my mentors, peers, CAD team as well as vendors for helping in the whole process

REFERENCES

[1] Rohit Kumar Sinha, Ashish Hari, Sulabh Kumar Khare, "AdvanceTechnique to Accompolish Power Aware CDC Verification", DVCon Europe 2018
[2] A. Chen, et. al., "Power Aware Clock Domain Crossing Verification", DAC 2014.
[3] P. Yeung, "Multi-Domain Verification: When Clock, Power and Reset Domains Collide", DVCon, March 2015.
[4] White Paper POWER AWARE CDC VERIFICATION OF DYNAMIC FREQUENCY AND VOLTAGE SCALING (DVFS) ARTIFACTS by MARK HANDOVER, JONATHAN LOVET T, KURT TAKARA
[5] P. Kothanath, "Clock Domain Crossing Verification for an SoC : Beyond The Usual Suspects", User2User Conference, October 2014.
[6] C.E.Cummings, "Clock Domain Crossing (CDC) Design and Verification Techniques using System Verilog," SNUG 2008