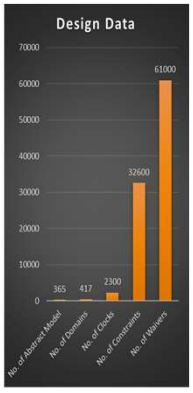
A Step towards Zero Silicon Bugs: SVA Protocol Based Assumption Validation

Rohit Kumar Sinha, Babu Christie

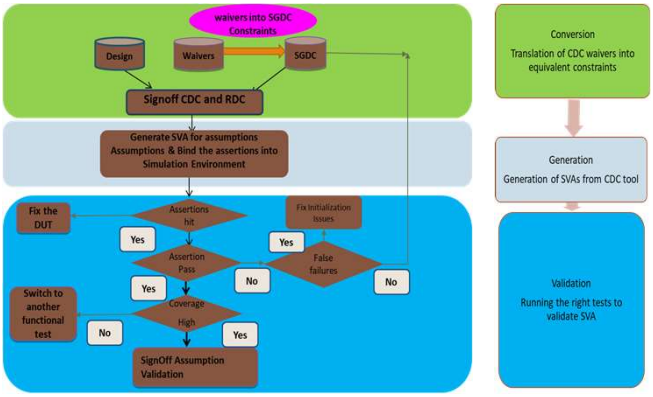
Intel India

MOTIVATION

01	SOC COMPLEXITY	SoC Integration with 1000+ IPs Multiple Stakeholders such as DFT, MBIST, TAP, DEBUG CDC Signoff – a collosal effort
02	VULNERABILITY	CDC-RDC Closure error prone because of handling waivers, constraints mismatches Constraints Intreprations are not understood well because of skill-set gap
03	COSTLY RESPIN	Single CDC or RDC miss is costly Late issues debuggability is complex
04	PROPOSAL	Need an Robus Automated Flow Coverage Analysis for Signoff Quality



CDC Assumption Validation using SVA Protocol



GENERATION: SVA PROTOCOL FROM CDC CONSTRAINTS

Type	CDC Constraint Name	Purpose
Assumption	quasi_static -name	• the value of signals is predominantly static • If a clock on the destination flip-flop is stopped • If a reset is active on a destination flip-flop • If the logic in the clock domain crossing path is not sensitive to any metastability issue
Assumption	set_case_analysis -value	Checks if sca is set to specified value
Assumption	clock_relation –from/-to_clock	Verifies relation between from_clock, to_clock and ref_clock
Assumption	reset_filter_path –type reset_sync02 –from_rst/-to_rst –clock	CDC Verifies the sequencing between from_rst and clocks
Assumption	qualifier –src_stable	Verifies the src_qual != dest_qual
Assumption	quasi_static_rdc	Verifies that whenever reset changes, flop is already at its reset value
Assumption	reset_filter_path –type rdc –from_rst/-to_rst –clock	RDC Verifies the sequencing between from_rst and to_rst and clocks

CDC Assumption Constraints and its purpose of SVA validation

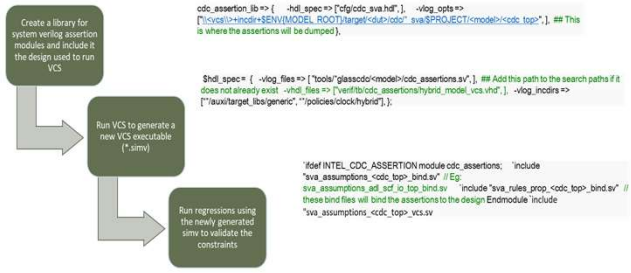
Functional Test Summary Report

"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**quasi_static**_9.ADVCDC_DETECT_TOGGLE", 885 attempts, 881 successes, 0 failures, 1 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**reset_filter_path**_0.ADVCDC_DETECT_RFP", 1 attempts, 1 successes, 0 failures, 0 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**set_case_analysis**_0.ADVCDC_WRONG_VALUE_INIT.unnamed$$_0", 1 attempts, 1 successes, 0 failures, 0 incompletes

IMPLEMENTATION



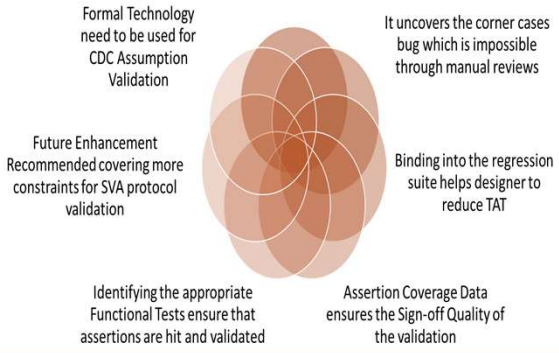
"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**quasi_static**_9.ADVCDC_DETECT_TOGGLE", 885 attempts, 881 successes, 0 failures, 1 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**reset_filter_path**_0.ADVCDC_DETECT_RFP", 1 attempts, 1 successes, 0 failures, 0 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVCDC_**set_case_analysis**_0.ADVCDC_WRONG_VALUE_INIT.unnamed$$_0", 1 attempts, 1 successes, 0 failures, 0 incompletes

CONCLUSIONS

CDC constraint Assumption validation using SVA Protocol is a MUST especially in SoC Design



REFERENCES

DAC-2020- Rohit Kumar Sinha, Babu Christie

Clifford E. Cummings, "Clock Domain Crossing (CDC) Design & Verification Techniques Using SystemVerilog," SNUG 2008, Boston, MA

DVCon-2015 SystemVerilog Assertions for Clock-Domain-Crossing Data Paths

Intel India