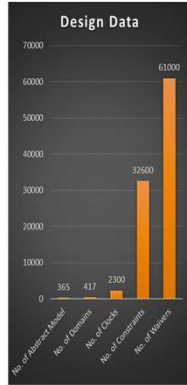
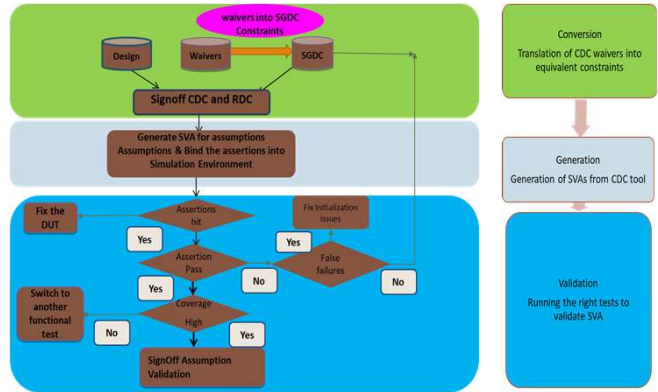


MOTIVATION

- 01 SOC COMPLEXITY** SoC Integration with 1000+ IPs
Multiple Stakeholders such as DFT, MBIST, TAP, DEBUG
CDC Signoff—a colossal effort
- 02 VULNERABILITY** CDC-RDC Closure error prone because of handling waivers, constraints mismatches
Constraints Interpretations 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	<ul style="list-style-type: none"> 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 and to_rst and clocks -from_rst/-to_rst -clock	RDC Verifies the sequencing between from_rst and to_rst and clocks

Functional Test Summary Report

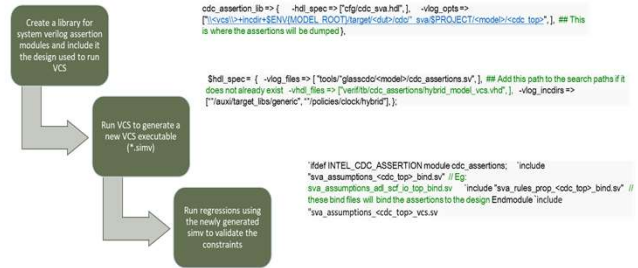
"cpuss_tb.Assumption_mod_cpuss.ADVDCD_quasi_static_9.ADVDCD_DETECT_TOGGLE", 885 attempts, 881 successes, 0 failures, 1 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVDCD_reset_filter_path_0.ADVDCD_DETECT_RFP", 1 attempts, 1 successes, 0 failures, 0 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVDCD_set_case_analysis_0.ADVDCD_WRONG_VALUE_INIT.unnamed\$5_0", 1 attempts, 1 successes, 0 failures, 0 incompletes

CDC Assumption Constraints and its purpose of SVA validation

IMPLEMENTATION



```
cdc_assertion_lib => { -hdl_spec => ["cpg/cdc_sva.hdl"], -vlog_opts => ["-svlog2=mode=SENVIMODEL_ROOT/target/sub/ods" -sva$PROJECT=cmodel>cdc_top*"], ## This is where the assertions will be dumped,

shdl_spec = { -vlog_files => ["tools/glassco/cmodel/ods_assertions.v"], ## Add this path to the search paths if it does not already exist -hdl_files => ["verifib/cdc_assertions/hybrid_model_vcs.vhdl"], -vlog_includes => ["$aux/target/lib/generc", "$policies/clock/hybrid"],

`def INTEL_CDC_ASSERTION module cdc_assertions; include "sva_assumptions_cdc_top_bind.sv" // Eg: sva_assumptions_ad_scf_io_top_bind.sv include "sva_rules_prop_cdc_top_bind.sv" // these bind files will bind the assertions to the design Endmodule include "sva_assumptions_cdc_top_vcs.sv
```

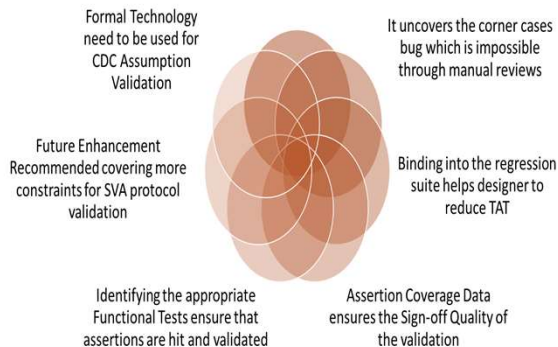
"cpuss_tb.Assumption_mod_cpuss.ADVDCD_quasi_static_9.ADVDCD_DETECT_TOGGLE", 885 attempts, 881 successes, 0 failures, 1 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVDCD_reset_filter_path_0.ADVDCD_DETECT_RFP", 1 attempts, 1 successes, 0 failures, 0 incompletes

"cpuss_tb.Assumption_mod_cpuss.ADVDCD_set_case_analysis_0.ADVDCD_WRONG_VALUE_INIT.unnamed\$5_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