Coverage Driven Signoff with Formal Verification on Power Management IPs

Baosheng Wang
Advanced Micro Devices, Inc. Sunnyvale CA

Xiaolin Chen
Synopsys Inc. Mountain View, CA

Abstract

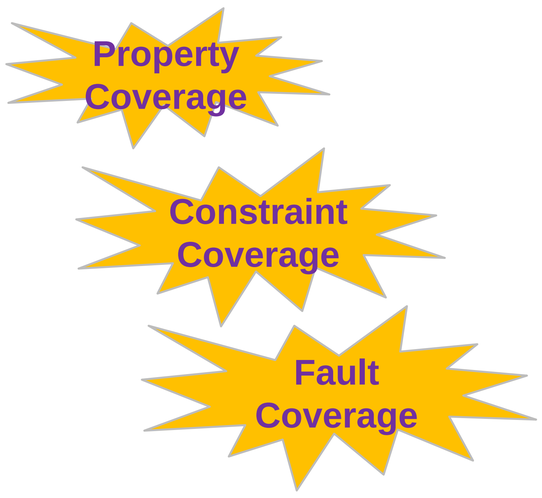
Identifying the right balance between simulation and formal verification has always been a challenge. We have learned that in some cases where formal verification is the only viable approach to verify critical features in SoCs because they require exhaustive coverage. These critical features included reset, clock, fuse, power management controller, and so on. Using a couple of power management IPs as an example, this paper describes our experience working with formal technology from verification planning, scoping, to signing off on multiple power management blocks completely. The tools we used are VC Formal and Certitude from Synopsys. We feel the outcome is very valuable and it leads us to establish the flow in the verification process across projects deploying formal verification to fully validate all possible reset states and their potential transition scenarios. Coverage measurements are used as the guidance throughout the flow from beginning to signoff. We hope you can benefit from what we have learned in our experience.

Power Management IP with Formal Verification

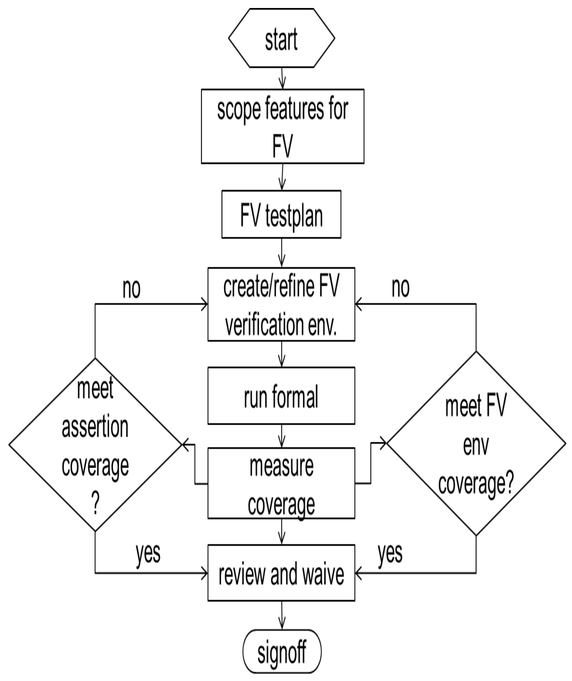
- Design characteristics
  - Control logic centric
  - Complex FSM
  - Smaller size blocks
  - Suitable for formal
- Challenges
  - Require formal expertise
  - Ad hoc methodology
  - Manual process
  - Lack of signoff progress measurement

Coverage Rises Up to the Challenge

- "Have I written enough assertions?"
- "Have I over-constrained my formal environment?"
- "How effective are my checkers at catching design bugs?"



Coverage Driven Formal

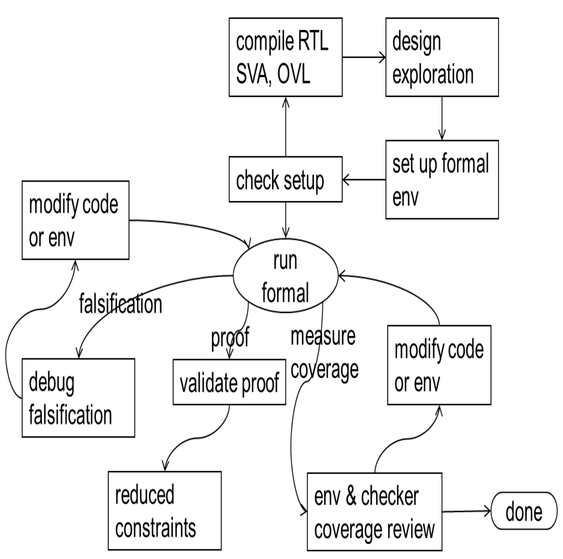


Verification Test Plan Example for a Reset IP

- Verify correct functionality of the FSM
- Verify no deadlock, unexpected uncoverable states, invalid transitions for the FSM
- Verify no X's are on the inputs/outputs
- Verify reset IP stays in the same state once the "GO" flag for that state is asserted until the "DONE" flag is asserted.
- Verify reset IP pauses in the desired state when "pause" signal is asserted during silicon debug
- Verify "Ready" signal is asserted once the reset completes
- Verify outputs from the reset IP is generated properly as expected.

.

Formal Work Flow



VC Formal Property Verification Work Flow

FV Environment Development

SVA Checker Example

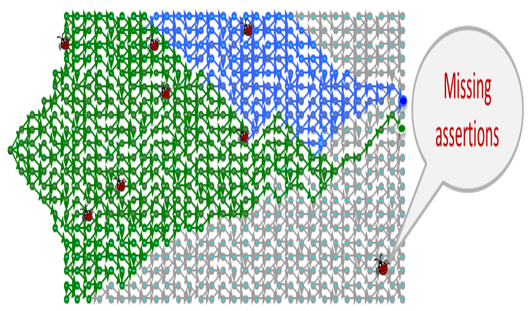
```
//FuseSel high when enter fuse
statefusesel_chk:
assert property (
@(posedge CLK) disable iff (disable_chk)
((!BypassFuse && !DftWarmReset) ##1
(!$stable(IP_RstState) && (IP_RstState ==
`IP_FUSE) && !BypassFuse && !DftWarmReset
&& !WarmResetFlag)) |=> (IP_FuseSel));
```

SVA Constraint Example

```
//Bist one done once
bist_done_assume3:
assume property (
@(posedge CCLK) disable iff
(disable_chk) ((IP_RstState == `IP_BIST)
&& BistDone)) |=> (BistDone));
```

Measure Property Completeness

- "Have I written enough assertions?"
  - Coverage to check for property completeness
  - Identify register elements outside of cone of influence of any properties



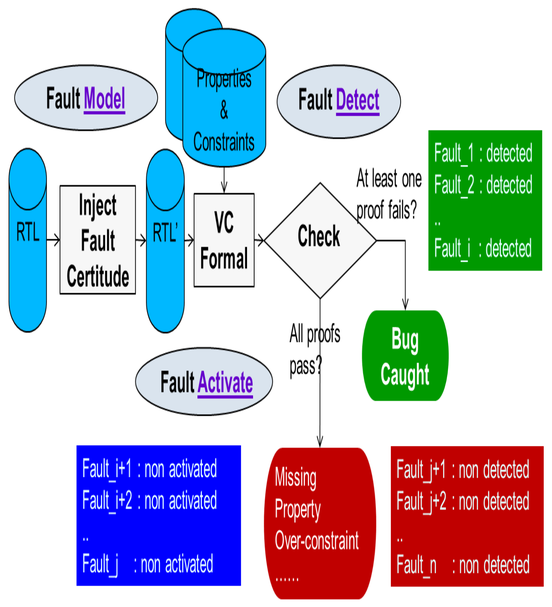
  - Lightweight, fast run time
  - Good for initial assertion coverage measurement

Measure Environment Coverage

- "Have I over-constrained my formal testbench?"
  - Code coverage
    - Deadcode
    - Uncoverable targets due to constraints

toggle		q[0]	0->1		srdff._GRS_LongReset
toggle		srdff_q[0]	0->1		srdff._GRS_LongReset
toggle		q[0]	1->0		srdff._GlbRstStateDone_d2
toggle		srdff_q[0]	1->0		srdff._GlbRstStateDone_d2
line	IF			UNINT_UNR_INT_RCH	srdff._GlbRstState_d1[0]
line	IF			UNINT_UNR_INT_RCH	srdff._GlbRstState_d1[0]
line	IF			UNINT_UNR_INT_RCH	srdff._GlbRstState_d1[1]
line	IF			UNINT_UNR_INT_RCH	srdff._GlbRstState_d1[1]

Fault Injection + Formal



Fault Analysis Results

- "How effective are my assertions?"

Class Name	Faults In Design	Non-Activated	Non-Propogated	Detected	Non-Detected	Disabled By Certitude	Disabled By User	Dropped	Not Yet Qualified
TopOutputsConnectivity	24	0	0	12	6	0	0	6	0
ResetConditionTrue	0	0	0	0	0	0	0	0	0
InternalConnectivity	0	0	0	0	0	0	0	0	0
SynchronousControlFlow	0	0	0	0	0	0	0	0	0
SynchronousDeadAssign	0	0	0	0	0	0	0	0	0
ComboLogicControlFlow	60	0	0	7	3	10	0	40	0
SynchronousLogic	0	0	0	0	0	0	0	0	0
ComboLogic	384	0	0	43	18	28	0	295	0
OtherFaults	0	0	0	0	0	0	0	0	0
All Fault Classes (9)	468	0	0	62	27	38	0	341	0

Total number of effective faults injected in the design

Faults detected by at least one property

Faults not detected by any of the properties

Faults cannot affect result

Fix the more important weaknesses first

  - Not enough checkers
    - Faults not activated
  - Checkers not good enough
  - Missing checkers
  - Over-constraining
    - Faults not detected

What We Learned

Summary

- **50** RTL bugs found in **12** power management IPs
- Junior engineers completed formal tasks from start to finish
- Coverage closure deployed in formal verification environment
- Created automated regression flow for all blocks

Conclusion

- Coverage flow provided guidance and measurable progress
- High confidence in IP quality
- Reduced verification cycle
- Efficiency in resource planning