

Verification Leadership Seminar Session: Improving Debug

Sagi meller [Saifun Semi.]



Freddy Bembaron [TI]



Agenda

We will discuss debug improvements options within the following areas:

- What exactly do we have to debug ?
- Design Phase - DFV
- Verification Team Ramp up
- Verification Environment
- Verification phase
- Regression/RPG phase
- Debugging FW Code & Debugging post silicon

What exactly do we have to debug ?

- **RTL** – Simulators, assertions, logs, Verdi.
- **Verification Code (e/sv)** – Specman / Simulator, logs.
- **FW code** – Seamless co-simulation, FPGA.
- **Post Silicon Chip** – DFT, Debug ports, oscilloscope module, synthesable assertions.

Design Phase

Design For Verification

- In concept review Estimate also verification impact and drive to the simplest implementation.
- Using buffer cells for important signal in order to preserve them after the synthesis phase.
- Propagate important signals to interface
- Use Assertion in design
- Use System-Verilog Interfaces.

Verification Team Ramp up

- Who should do the rtl debug ?
The designer ? The VE ? Maybe Both ?
- Getting familiar with the design:
 - How well does the VE know the design?
 - Perform dedicated Design ramp up sessions at the beginning of the project with Q&A
 - Design reviews are not the place to learn

Verification Environment

- How much time do we spend debugging the verification environment?
 - Checkers/monitors :
 - Usually Huge amount of Dirty Code
 - Need to insert modularity, sharing of resources, coding style
 - Insert Assertions to design to indicate design Vs Monitor Error
 - Insert Assertions in critical sequences for basic non-functionality early indicators
 - Perform design like Reviews before coding
 - Monitor extra printing for debug phase (rather than regression)
- Block level verification
 - Integration level debug is usually more complicated
->reduce it with block level early debug
- Reuse of mature VIP's and verification code can save a lot of time while debugging.

Verification phase

● Debug Methods Training

- Create training on Debug methods:
 - When to use breakpoints
 - When to use post processing features
 - When to immediately call designers
 - What to validate before passing to designer debug

● Debug Experts:

- DO we need to allocate in XX% debug expert to go through novice VE's?

● Tools:

- Trace driver/x in circuit/code emulation.
- VNC
- Debussy/Verdi.
- SPECMAN a/o Simulator.

Regression/RPG phase

- Regression phase:
 - Release monitor to regression only after passes basic predefined tests (not after compilation)
- RPG (Random pattern generator):
 - Usually more complicated debug
- Use of good reporting tool s.a. Vmanager which can help you focus on your most failing checkers. Focusing on a specific checker helps us identifying the rtl which cause the failure.

Debugging FW Code & Debugging post silicon

- Debugging FW Code
 - Using FPGA.
 - Seamless simulation.
 - Using debug messages and logs.
- Debugging Post Silicon
 - DFT
 - Debug ports
 - Oscilloscope module
 - Synthesable assertions.