



# Test Automation

Jacob Arian – CEVA-DSP

Michael Goldenberg – PMC-sierra

# Agenda

- Test Automation
- Motivation
- Problems
- Examples
- Solutions

# Test Automation

- Building an environment that tests the DUT automatically
  - Instead of checking the DUT by eye, get computers to do the work for us
- Automated tests can create as many problems as it solves but it's worth it

# Motivation

- Improve the testing process
  - Speed up testing to accelerate releases
  - Reduce costs by reducing man power
  - Improve test coverage
- Allow testing to be done by staff with less skill
- Make testing more interesting

# Problems

- Take a lot of time to create the automation
- Need experience and knowledge in E/VERA/SV/SYSTEM C etc.
- Need a good understanding of the Architecture spec

# Problems (Cont.)

- Need Complex and Expensive Tools
- Dual Gain – less design and more verification bugs
- Need Multi Level Review

# Examples

- Scoreboards – forecasting the results.
- Checksum or CRC – calculation check.
- Assertions – check protocol legality.
- Trace files – save registers value during the test from the reference model and from the Verilog and check on test end.
- Use input and output vectors for post silicon automation test (with IXIA or smartbit).

# Solutions

- Use Common Scripts – create.cmd, stub.cmd, compile.cmd, run.cmd and analyze.cmd.
- Use Common Tools
- Use Common Template – environments, TBs, tests and documents.

# Solutions (Cont.)

- Use “creator” Scripts – from XML, Word, EXL to V, VHDL, E, H, DOC, PDF, vPLAN, Coverage and others
- Divide and Rule – build your “automatic” environment from Bottom to Up
- Review “Auto” Code in relevant forum

# Solutions (Cont.)

- What can be improved?
- How reduce penalties?
- Best practices?

**The End**

**CEVA™**

**PMC**  
PMC-SIERRA