

Verification Futures: The Next 5 Years

Bryan Dickman

Director Design Assurance, PD

15/11/11



1. Complexity

Bug Avoidance

Bug Hunting

Bug Absence

Bug Analysis

- We all agree on the importance of verification, but....
 - 50% of the problem is in how to create designs that work?
- Spaghetti code lives forever...
- Avoid bugs at design capture
- “correct-by-construction”
- Apply design practices to DFV and PPA
- *Can design abstraction/ESL help?*
- *Can formal help?*



2. Scalability

- Constrained-random simulation has been proven as a good bug-hunting flow, but...
 - How much simulation will be enough for a 10 GHz CPU?
 - How many cycles to verify 2 weeks at target speed of 1GHz?
 - Answer: 0.6×10^{15}

	Simulation (KHz)	Emulation (1 MHz)	FPGA (10 MHz)	Si (1 GHz)
Target cycles 10^{15}	1,000,000 sim slots	1000 emulation slots	100 FPGA slots	1 chip
Achievable cycles	10^{11}	10^{12}	10^{14}	10^{15}

- *How will we scale simulation, emulation, FPGA to next gen of CPUs?*
- *What are the alternatives?*



3. Completeness

- Coverage is a great tool to demonstrate that we have done what we said we would do, but...
 - How do we know that we are done? (\$64MB question)
 - How many undiscovered bugs are there?
 - (known-knowns, known-unknowns, unknown-unknowns)
 - (coverage hit, coverage holes, coverage model completeness)
 - Does coverage mean correctness?
- *What other measures do we have?*
 - *Cycles, bug curves*
- *Again, does formal help?*

