Embracing Formal Verification: An Intel® Graphics Experience

M, Achutha KiranKumar V
VPG-Gfx
Agenda

- History of Formal @ intel
- Formal POC @ Gfx
- Our journey in brief
- Rich Experiences
- What worked for us
- What can be better
- Conclusion
FV @ Intel: When did it start Memories of FDIV

- **June 1994**: Oops!
  \[(4195835*3145727)/3145727 = 4195579\]

- **October**: Error posted in public, mass panic

- **December**: **Recall!** Intel agrees to replace any Pentiums with the bug
  - Pentium FDIV escape (0.7 Billion in 2010 $)
What was the problem?

- Graph from Larry Hoyle, U. Kansas: x, y, x/y in small region
- In tiny portions of design space, wrong answers possible
FV Ultimate Reality: i7 EXE Execution

- Complete Datapath Verification for all FPU and data processing units
- Complete Control path verification of control paths in the EXE cluster
- Verification of all Assumptions in the Datapath through control invariants
- Plugged in all these properties at the FC level

Ref: Replacing Testing with Formal Verification in Intel® Core™ i7 Processor Execution Engine Validation - CAV 2009
Story of Formal in Gfx

- Started with Datapath formal through STE, on a standard FPU design
- ~286 bugs found on the next generation design
- Strong and main tool for finding bugs on all next generation designs
- Increased the gamut of formal offerings on simple control path designs
- Started on simple problems and gained the confidence of designers and management
• Deployed various standard flows for control path formal activities.
• RTL2RTL Formal Applications re-invented and drove the deployment.
  - Simpler investment
  - Faster and guaranteed results.
  - Reduced Barrier for design/DV Engineers
• Explored property generation tools and associated benefits of using those flows
Story Continues.....

- Work closely with a central body of formal methodology deployments @ intel to expand the utilization in various use cases
- Sharing our experience and expertise through various technical forums within intel and across the industry through paper submissions and technical discussions
Successes and Failures of FV

- Better Reachability of FV for “non-phD” enthusiasts
- Good “mentoring” by the experts to naïve users
- More areas into Formal and better convergence

- Still the barrier is high – need “right” education
- Challenges in pockets to convince the teams that FV is no “second fiddle”
- Need to penetrate/convince Management/u-arch/Senior Designers
Future Expectations from FV

- Easier acceptance of the tool and better debug features
- Existing “Test bench” to “Over Constrained” Conversion for a kick start.
- Faster convergence and shareable “property modules”
- Tools getting more user friendly
- FV recipe preparation from specification
- Designs are churning faster.....hence faster proving methodologies
What could be done better?

- ABV should be the religion for every design
  - “An Assertion for every bug fix”
- Expert Rotations
- Seamless integration of formal and dynamic methodologies
- Better interaction/partnering with Academia
- Designers to become “partner – in – crime” from inception
Product Lifecycle

Astronomers
- Create upfront abstract hypothesis of how universe works.
- Reverse engineering is often required.

Wine makers
- Pruning, grafting, and genetic engineering (of RTL)

Quilt makers
- Stitch together many small specifications

Environmentalist
- Reduce, Reuse, Recycle

FPV Roles

Dentists
- Regular checkups (regression)
- "proof-decay", Filling cavities during regression
- Maintain records for large population.

Inventors
- Need flexible proof systems that support building solutions
- Hone checking recipe

Bungee jumpers
- Development cycle too long
- Need interactive, responsive proof systems

Judge
- Act from design impartial view
- Take a broader perspective
- Affirm correctness of design
Some BKMs from our FV Experience

- Start simple and be nimble
- Results speak more about the efficiency of formal
- Quality of the bug rather than the quantity of bugs
- FV is time taking but a sure solution for most of the problems
- If we can’t achieve all at one time, better divide and conquer
- FV completion is not a linear function over time. Most of the times, it is a step function. Be patient.
- Regular updates to management in a perceivable format helps all through the chain understand the progress.
- Last but not the least, You need strong management “Will”