Big Data Hiding in the Human Debug Process

Eric N. Harris

Arm

Eric N. Harris

Arm Principal Engineer

- B.S, M.S University of Wisconsin-Madison
- 7 Years Samsung Exynos CPU Level Team
  - Debugged and filed 500+ RTL Design Bugs
  - Developed recurring nightmares about the ARM Architecture Reference Manual
  - Triage tool advocate
- 3 Years Arm Big-A CPU Team
  - Fetch Unit Verification Lead (Branch Predictor + ICache)
  - Cortex-X and Neoverse cores

The Realities Facing Debug

- Rarely one definitive design microarchitecture specification
  - Not often machine readable
- Signal names are interesting
  - Many signal names imply their function, except when they don't
  - Humans know broccoli_i1 is likely related to carrot_i1
- Testbenches
  - For every example of well defined and "proper" UVM-style use there is an example of a kludge
- Bugs
  - Rarely filed during early development phase
  - Rarely detailed
- "Debug intuition" difficult to codify, teach or even define.
- Current Narrow AI solutions not well suited to debug
  - No good training data

Debug is a big data problem!

- In the last several months across all projects at Arm
  - 293 Million Tests
  - 6 Million Failures
  - 65K Signatures
- Known issues tend to exist for some time
  - Hopefully known issues have a bug filed on them
- Fails tend to have correlation to
  - Recently changed code (design or testbench)
  - Similar root cause to previous time signature was seen
  - Other fails
    (example: tests targeting parity errors find a parity bug with several failure modes)
- If you can capture all the test run data and most of the “triage result” data you can apply ML and other data science to help human debug

Eric's Verification Utopia

- An overnight "bake" of 500K tests reveled 100 failures across 10 signatures
  - 6 signatures were automatically matched to known issues by the triage tool
    - via engineer entered match criteria when bug was filed
  - 1 signature was seen a month ago and has now returned
    - Last time it was seen, a bug was filed on alu.sv
    - Recently alu.sv had a change in the version control system
    - An automated process attempted a bisect around this change and determined it to be the change that made the test transition from passing to failing
    - User given: the possible offending commit, along with the test log and waveform are generated
  - 1 signature was on the prior project, and an AI process predicted the engineer would want to debug
    - The bug details last time are presented to the debugger, along with verbose test log and waveform generated
  - 1 never before seen signature with several failing tests
    - Big Data classification determined tests with "parity_enable=1" were correlated with a much higher fail rate.
  - 1 single failure with a signature never seen before
    - AI Automated test generation algorithm attempted to and was able to generate 3 additional failing test cases

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה

Debug Workflow

- Before the failure: Gather background knowledge on the design and expected behavior
- View regression results to find failures
- Identify which failures need to be debugged
- Understand the history behind the failure
  - Occurrence of failures
  - Presence of a known issue
  - Types of tests failing
  - The root of the failure last time it was seen
- Gather more information on the failure: verbose logs, waveforms
- Debug and find the bug
- Propose fix
- Test fix on 1 test
- Test fix on all failing tests
  - Failures may be on different versions on the design, and fix needs to be tested on each in a way to avoid changing test randomization
- Regress and Release fix

Debug Workflow

- Before the failure: Gather background knowle
- View regression results to find failures
- Identify which failures need to be debugged
- Understand the history behind the failure
  - Occurrence of failures
  - Presence of a known issue
  - Types of tests failing
  - The root of the failure last time it was seen
- Gather more information on the failure: verbo
- Debug and find the bug
- Propose fix
- Test fix on 1 test
- Test fix on all failing tests
  - Failures may be on different versions on the des y to avoid changing test randomization
- Regress and Release fix