Top 3 Verification Challenges

Tim Joyce

Verification Manager

STMicroelectronics Unified Products Division, Bristol

life.augmented

Team Background

- SoC products for the Set Top Box and Home Media Gateway markets.
- Design complexity in the range 100-500M transistors.
- One of four SoC teams for UPD worldwide (UK, France, India, USA).
- Bristol SoC team is 20 people – microarchitecture, design, physical implementation, timing closure, DFT and SoC verification.

Challenge #3 – SoC Simulation Complexity

- Increasing IP count of SoCs means slower simulations.
  - Need to attack the problem with multiple measures:
    - Use a model of host CPU that compiles ‘c’ test code direct to host Linux server object code, and issues only bus transactions (~5x speed-up).
    - Turn off functional clocks that you don’t need.
    - Build cut-down SoC snapshots (needs good black-boxing).
    - Use emulation earlier in the project.

Typical project comparison	Project A	Project B
Number of IPs	59	96
Primary functional clocks	71	182
Number of CPUs	4	13
Elapsed simulation time	1	1.7

Challenge #3 – SoC Simulation Complexity

Continued…

- Increasing number of SoC external interfaces means increasing testbench complexity.
  - Testbenches need to include models or VIPs for a wide range of interfaces.
    - DDR, SPI/NAND/NOR flash, SD card, PCIe, USB, SATA, Ethernet, HDMI, JTAG etc.
  - For example: ‘Project A’ has 19 testbench model instances, ‘Project B’ has 33.
- Simulation snapshot must be built and linked with support for VHDL, Verilog, SystemC, SystemVerilog, Specman and instrumented for coverage collection.
  - It is a challenge to get this to link and run successfully.
- …now add in low power verification with CPF/UPF support!

Challenge #2 – SoC Functional Coverage

- What haven’t you covered when you’ve finished running your SoC functional tests?
  - Typically 500 - 1000 tests in SoC regression suite.
    - Tests provided by IP providers promoted to SoC level.
    - ‘SoC’ IP tests re-used and adapted from previous projects.
    - Directed tests for system functionality eg security.
- What’s the functional coverage on ‘integration nets’?
  - Collect functional stimulus (‘toggle’) coverage data for specified levels of hierarchy.
  - Analyse and classify results to identify un-stimulated functional nets.
  - Write extra tests for missing coverage.

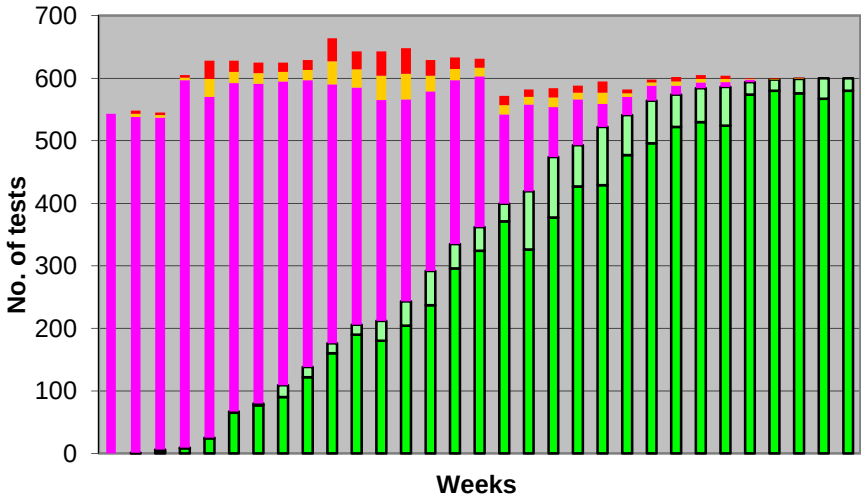
Challenge #2 – SoC Functional Coverage

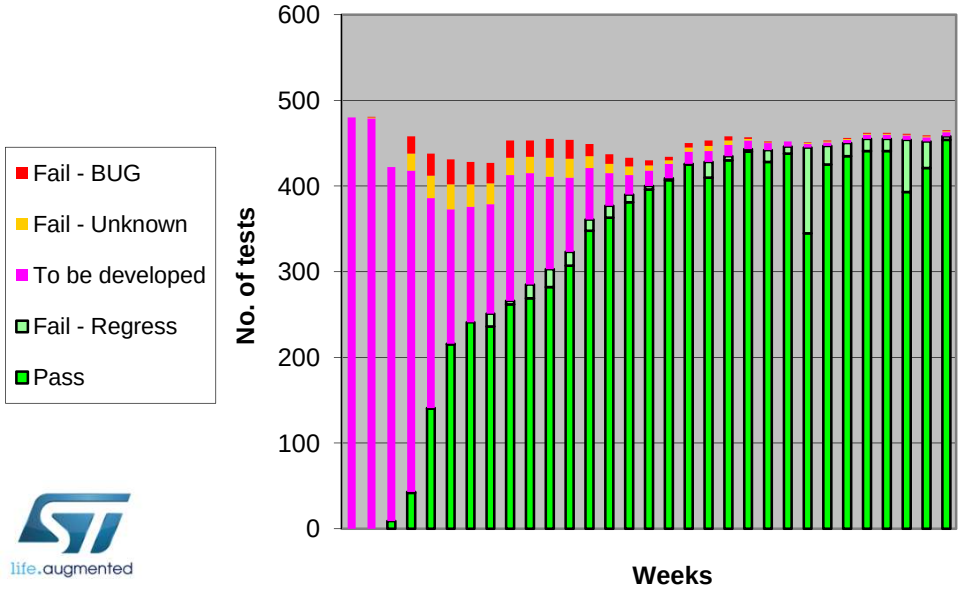
Continued…

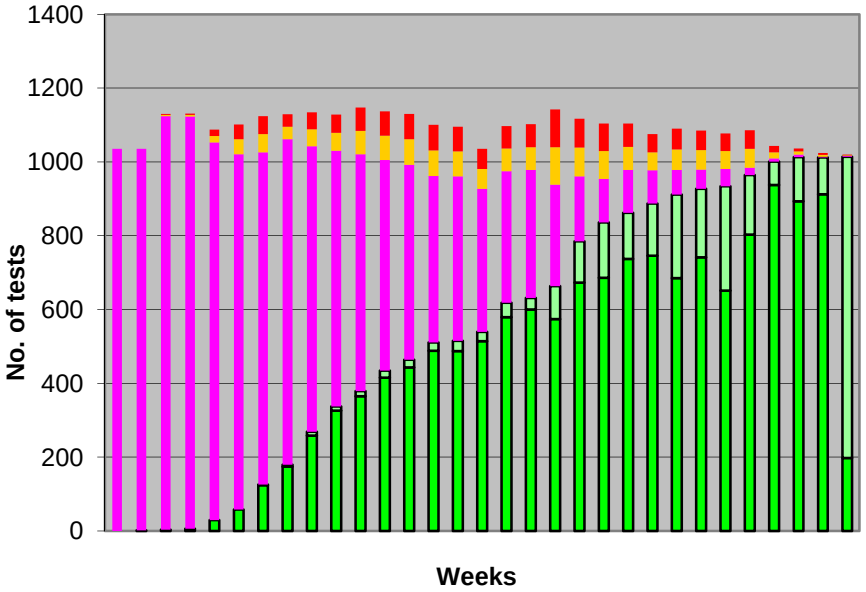
- Analysing and ‘excluding’ irrelevant missing coverage is a key, but very manual, task.
  - In-house flow based on importing toggle coverage data into an SQL database with a Visual Basic Excel front end.
  - Allows ‘exclude’ comments to be annotated and re-used on the next project.
- Ideally, tools need to automatically ‘exclude’ across design hierarchy.
  - Use design database to identify all endpoints of a net.

Challenge #1 - Resourcing

- How do you calculate how much resource you need?
  - Look at previous projects.
  - Average out the progress for one person – it's predictable!
  - Different rate of progress for a new project or a variant.





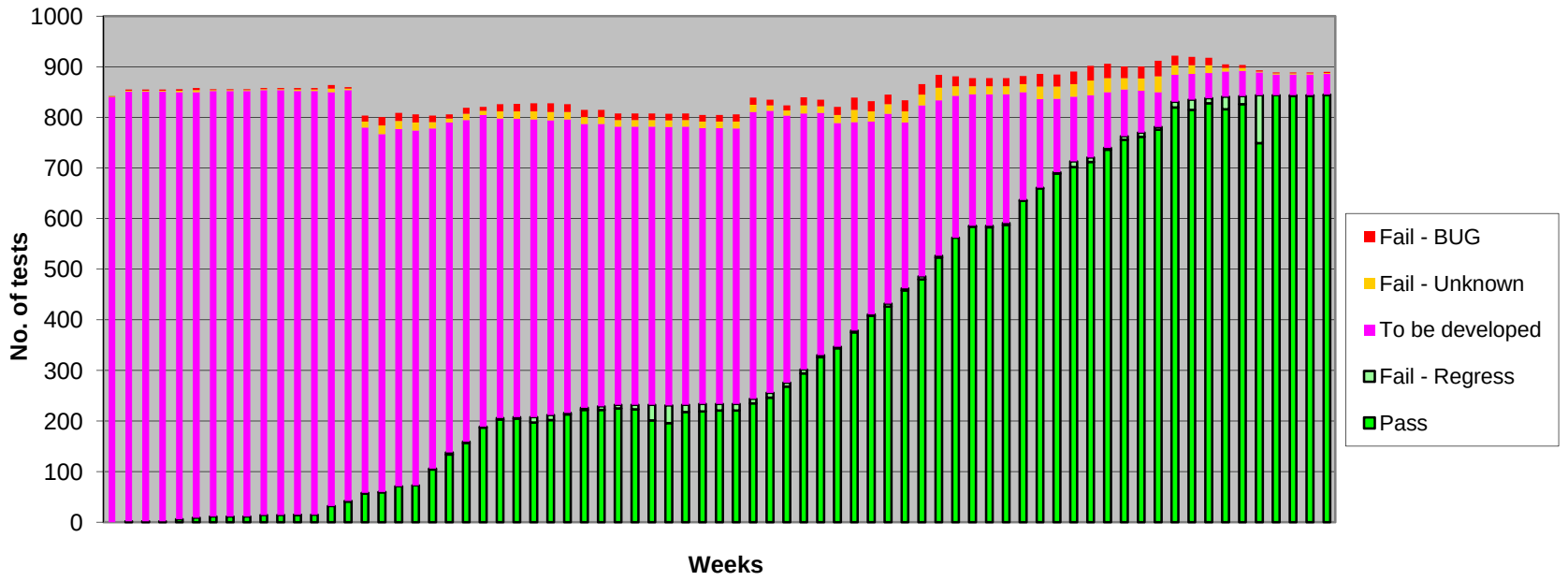


life.augmented

Challenge #1 – Resourcing

Continued…

- When you know how many engineers you need…
  - Work with other SoC Verification teams around the world, if available.
  - Or work with a company who can provide contracting engineers to effectively resource the project.



life.augmented

The Final Challenge…

life.augmented

The Final Challenge…

- ST Bristol site closure in Q1 2014 means a new challenge awaits!