Post Silicon Validation

Introduction

Silicon Validation

Validation is the activity of ensuring

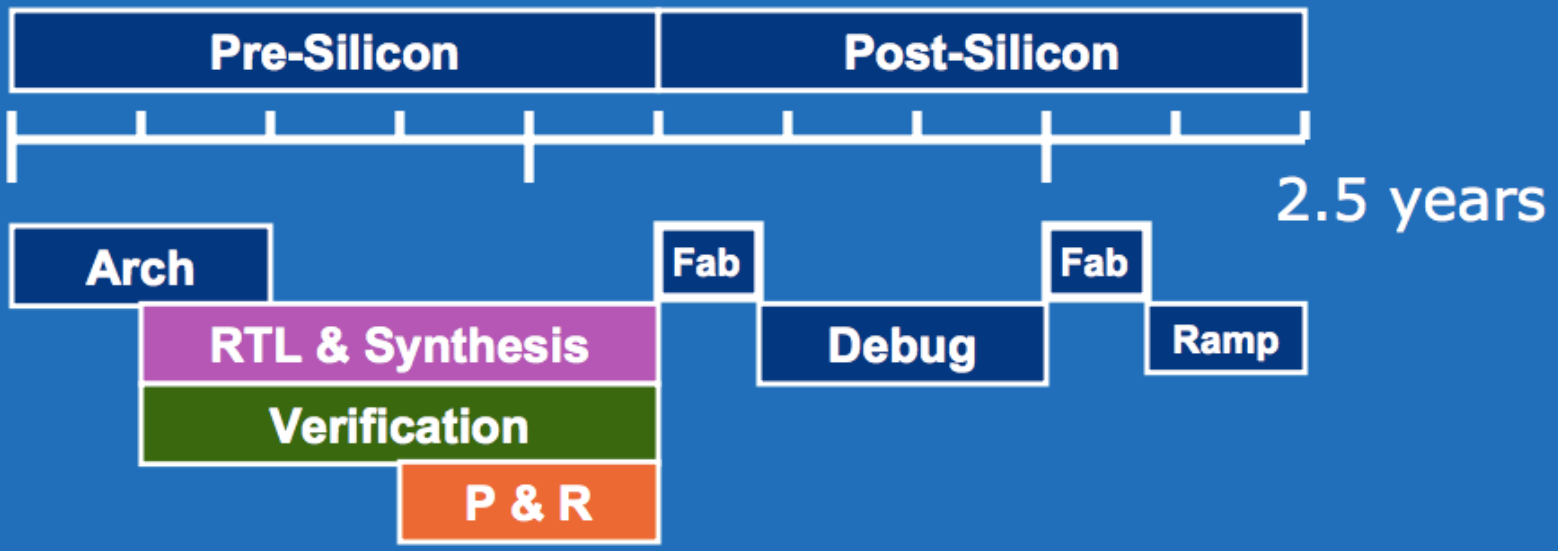
- a product satisfies its specifications
- compatible with related software and hardware
- meets user expectations.

Silicon Validation

Silicon Validation is needed because:

- Numbers of processor bug are growing
- Bugs are becoming more diverse and complex.

ASIC Time-to-Money



CAD: > 50 companies

IP: > 100 companies

Headcount ~ 50

Stanford Courses ~ 20

Test: ~ 10 companies

Debug: ~ 3 companies

Headcount ~ 5

Stanford Courses ~ 3

Pre silicon verification

- Pre Silicon validation starts after planning and architecture build.
- Verify design through simulation
- Methods: software based simulation, formal verification, FPGA based hardware emulation.
- Usually run at low speed (tens of hertz)

Post silicon validation

- Start after the first silicon arrives
- Validate design through silicon testing.
- Very fast (gigahertz)

Pre vs. Post Silicon Verification

Pre Silicon Verification

- Slow speed with limited amount of information
- Easy to control and debug
- Automated
- Full observability

Post Silicon Validation

- Fast execution speed leads to high information coverage.
- Very hard to control and debugging is very expensive
- Less automated, need manual help
- Limitted observability

Why is Post-silicon validation needed

- Because of simulation is much slower than actual system in hardware, not enough cycles can be simulated to gather enough information. Some bug may not be recorded.

Why is Post-silicon validation needed

- Bugs caused by physical effect to the system can't be detected by simulation. Need the actual chip.
- Physical effects includes: crosstalk, thermal, process variation and so on.

Aspect of Post-silicon validation

Debug infrastructure

- Reuse scan chain, trace buffers to output state information.
- On-chip logic control
- Focus on system centric solutions

Aspect of Post-silicon validation

Coverage

- Make sure test cases cover whole area.

Aspect of Post-silicon validation

Coverage

- To reduce overhead created loading software form memory, insert exerciser into the system
- Continuously generate, execute test cases.
- Check the correctness of the result.

Current problems

- **Limited observability**

Only a small amount of signals can be selected to represent the current state

It's very important and difficult to select signals that can reveal essential state of the System.

Use State Restoration Ratio (SRR) as standard

My research area

- Focus on communication centric validation.
- By monitoring signals and translate them into system level flow activities, try to map them into different flows and detect any event that can't be mapped to any existing flow.

Conclusion