




NORTHROP GRUMMAN

DEFINING THE FUTURE



Validation and Design
in a Small Team Environment:
*The Development of the High
Performance Embedded Computing
Process at Northrop Grumman*

Matthew Clark, Ph.D

February 19th, 2009

Motivation

Validation is:

- Complex
- Long
- Requires expensive tools
- Requires experienced employees.

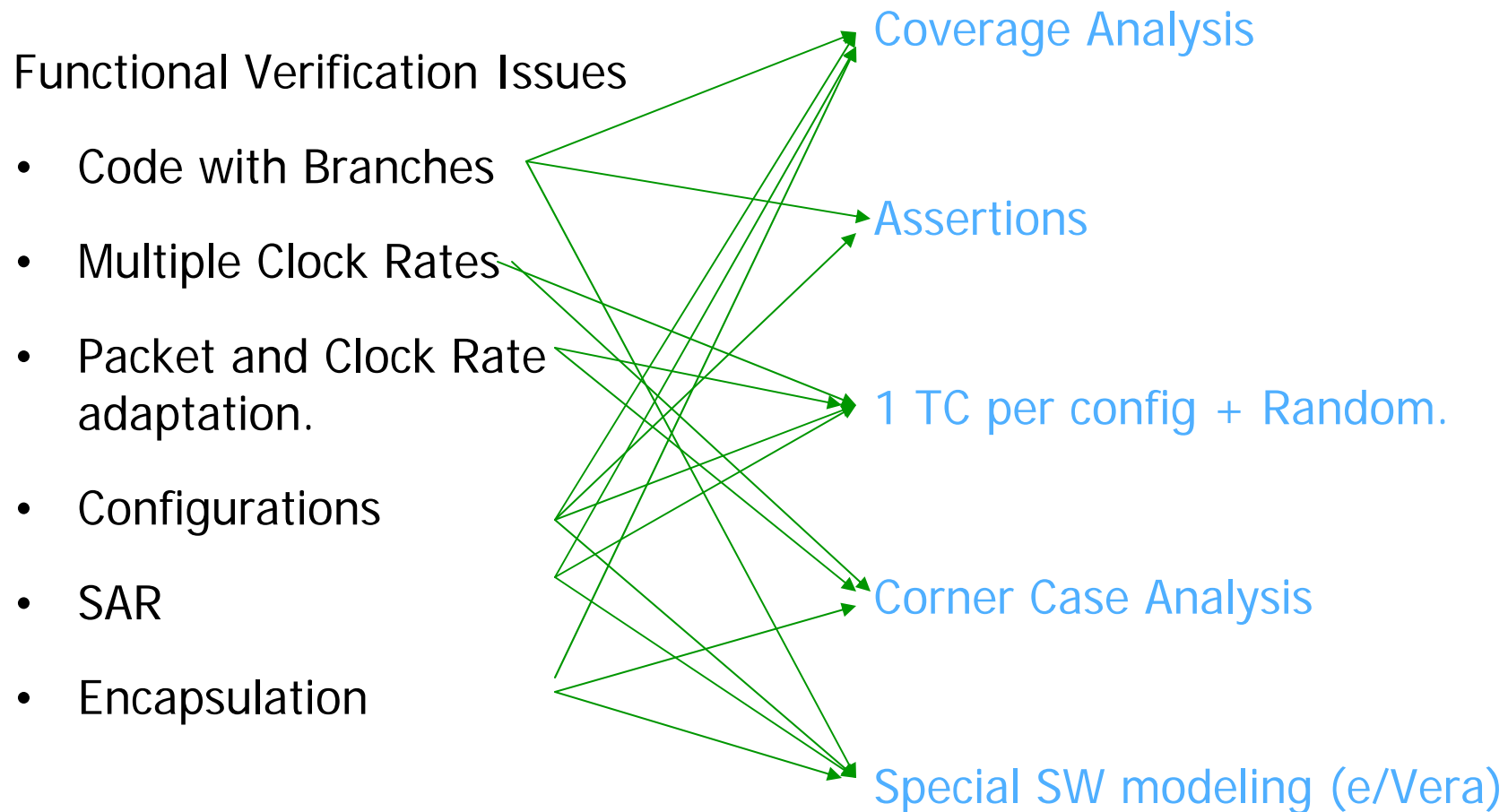
It is both an Art and Science.

What if your project is:

- Complex
- Lasts 3-6 months,
- Has a limited budget,
 - And new grads.
- Is done in a group with 30 years of SW experience, and you represent the sum total of HW development experience.

First Question

What makes verification so expensive and time consuming?



Identifying and reducing the green arrows is the Art.

Problem Statement



How can you reduce the number of green arrows so that designs can be completed quickly, reliably, without having the “normal” 2.5:1 Verification : Designer ratio?

A: Do what Capt. Kirk did, change the problem.

We can't change the verification restrictions, perhaps we can change the design issues so that we don't have some verification problems... by design.

Problem Space

- Offload CPU-intensive portions of numerical algorithms.
- Rapid deployment. Standard project is 3-6 months.
- Must be modular:
 - Single module stand-alone.
 - Combined into algorithm.
 - Multiple algorithms in single container.
- Exact duplicates of existing SW applications.
 - Goal is to reduce execution time, or improve performance (ie replace 64K pt FFT with 512K pt FFT).
 - To SW, there must be no difference between CPU, FPGA, FPGA+DSP, DSP implementation.

Design and Verification constraints.

- Single Clock Rate for every block.
- No Packet and Clock Rate Adaptation
 - Force it to occur at a single ingress/egress point.
- Heavily branched code
 - Algorithms (fortunately) tend to be datapath centric.
- Configurations
 - **Separate algorithm implementation verification from IO verification.**
 - Build bit/cycle accurate model of algorithm, use IO controls to vary data tx/rx.

Design constraints reduce verification space

Design Constraints

- All blocks use small input fifo's, and programmable full/empty indicators.
- All blocks execute at 210 MHz. (Max Register Access speed).
- Top Level register interface can be pipelined for timing closure.
- Coregen modules only (Ref. Designs Need Not Apply)
- Each block uses (1+) async resets, that are externally synch'd.
- Each block will have a testbench that:
 - Uses golden Matlab code to create **all** stimulus/checker files.
 - Scenarios are generated from configure files.
 - Test bench is nothing but interface BFM comparing values against file.
 - Can slew input/output rates independently.
 - Can randomly assert the full/empty flags on interface pins.
 - Interrupt conditions may be checked by hand.
 - Pass fail conditions checked in VHDL at run time.
 - Test bench can be run as regression.

Design constraints reduce verification space

Functional Verification Issues

- Code with Branches
- ~~Multiple Clock Rates~~ IO_clock and funct_clock only.
- ~~Packet and Clock Rate adaptation.~~ All IO is FIFO based. IO ring handles.
- Configurations
- ~~SAR~~ Not an issue.
- ~~Encapsulation~~ Not an issue

Results

- MegaCore Function (FFT, IFFT, Matched Filter, Correlation, Polyphase/WOLA), plus Shim for PCI bridge.
- Designer Tasks: Architect, Document (MAS), Block Test, Synthesize.
 - Forward FFT: BSEE (1 mo Verilog)
 - Inverse FFT : MSEE (6 mo Verilog)
 - Packet Aggregation: Intern (2 mo VHDL).
- Verification:
 - MSEE (no HDL).
 - Create Matlab based Testbench, stimulus checker files.
 - Created Stimulus files for Block Level Test Benches
 - Reused FIFO BFM's from previous project.
- Code + Block Test: 4 Months
- Verification + Synthesis: 2 Months.

Summary

Q: How can you reduce the number of green arrows so that designs can be completed quickly, reliably, without having the “normal” Verification : Designer ratio?

- Use standard, non-optimal, fifo interfaces for every block.
- Separate algorithm verification from implementation verification.
- Repeatable test cases at the cost of text file storage.
- Design each module like it is going into a reuse container:
 - Standard IF
 - Standard Clock and reset.
 - Standard IO and register interfaces.
 - All data widths are 64 bit + Vld @ 210 MHz.

Summary: Trade improved verification schedule and TTM with reduced performance and complexity.