COMS31700 Design Verification:

Verification Cycle, Verification Methodology & Verification Plan

Kerstin Eder

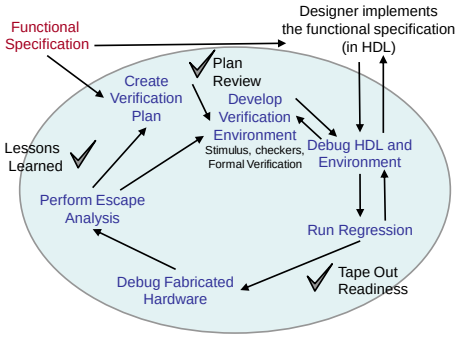
(Acknowledgement: Avi Ziv from the IBM Research Labs in Haifa has kindly permitted the re-use of some of his slides.)

University of BRISTOL

Department of COMPUTER SCIENCE

The Verification Cycle

The Verification Cycle



Common Verification Breakdowns

- Verification based on the design itself instead of the specification
- Underdeveloped verification plans
- Underdeveloped specifications
- Lack of resources
- Tape-out based on schedule instead of pre-defined measures

Summary

- Functional verification is a necessary step in the development of today's complex digital designs
- Verification engineers must understand the specification and internal microarchitecture of the design under verification
  - They couple this knowledge with programming skills, RTL comprehension, and **a detective's ability** to find the scenarios that uncover bugs.
- The two main challenges in the verification process:
  - Creation of a comprehensive set of stimulus
  - Identification of incorrect behavior when encountered
- The foundation for a successful verification is the well-defined verification cycle
  - The process includes creation of test plans, writing and running verification tests, debugging, and analysis of the holes in the verification environments

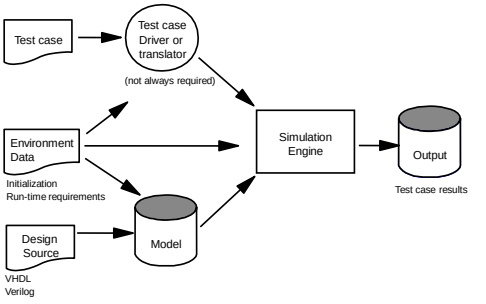
Verification Methodology

Outline

- Verification methodology evolution
- Basic verification environment
- Evolution of the **Verification plan**
- Contents of the **Verification plan**
  - Functions to be verified
  - Specific tests
  - Coverage goals
  - Test case scenarios (Tests list)
- (Calc1 Example)

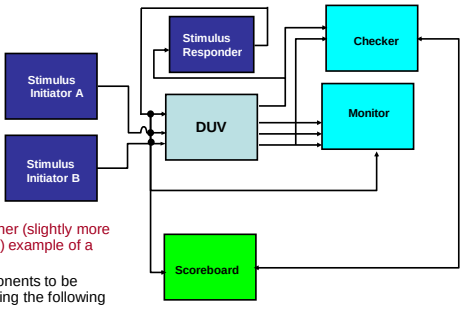
14

Simulation-based Verification Environment Flow



15

Simulation-based Verification Environment Structure

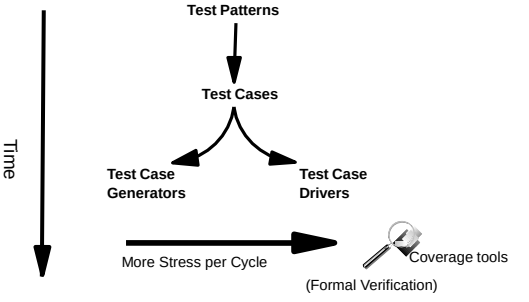


This is another (slightly more sopisticated) example of a testbench.

(New components to be covered during the following lectures.)

16

Verification Methodology Evolution



17

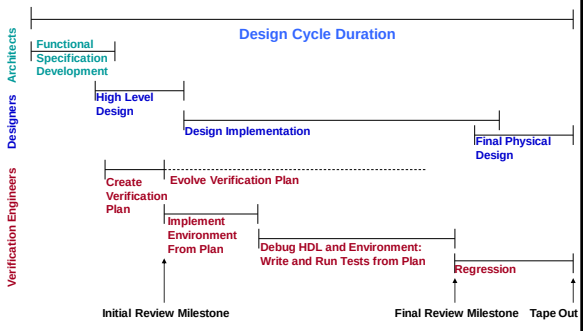
Verification Plan

Evolution of the Verification Plan

- The source of the verification plan is the Functional Spec document
  - Must understand the DUV before determining how to verify it
  - Confront unclear and ambiguous definitions
  - *Incomplete and changing continuously*
- Other factors may affect its content

24

Design and Verification Process Interlock



Calc1 Verification Plan

- The design description from A1 details the intent of the Calc1 design
  - It is the verification engineer's job to prove that the actual design implementation matches the intent.
- Even for a relatively simple design like Calc1, it is still best not to jump into test case writing before thinking through the entire verification plan requirements
- Please note: For the interview on A1 you need a verification plan.

Contents of the Verification Plan

- Description of the verification levels
- Functions to be verified
- Resource requirements
- Required tools
- Schedule
- Specific tests and methods
- Coverage requirements
- Completion criteria
- Test scenarios (Matrix)
- Risks and dependencies

Description of Verification Levels

- The first step in building the verification plan is to decide on which levels to perform the verification
- The decision is based on many factors, such as
  - The complexity of each level
  - Resources
  - Risk
  - Existence of a clean interface and specification
- The decision should include which functions are verified at lower levels and which at the current level
- Each level and piece selected need to have its own verification plan

Verification Levels for Calc1

- Calc1 is simple enough to be verified only at the top level
  - In addition we do not have enough details on the internal components
- In more realistic world we may decide to verify the ALU and shifter alone
  - For example, using formal verification

Functions to be verified

- This section lists the specific functions of the DUV that the verification team will exercise
  - Omitted functions may slip away and not be verified
- Assign Priority for each function
  - Critical functions
  - Secondary functions
- Functions not verified at this level
  - Fully verified at a lower level
  - Not applicable to this level

Required Tools

- Specification and list of the verification toolset
  - Simulation engines
  - Debuggers
  - Verification environment authoring tools
  - Formal verification tools
  - … and more
- For **Calc1 A1**
  - Simulation engine
  - Waveform viewer
  - Verification environment authoring tool
    - Or use the HDL to provide the environment

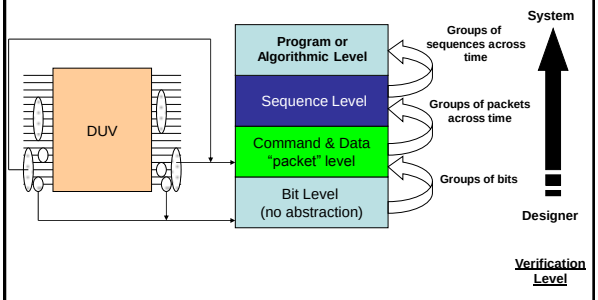
31

Specific Tests and Methods

- What type of Verification?
  - Black box
  - White box
  - Grey box
- Verification Strategy
  - Formal Verification
  - Deterministic
  - Random based
- Abstraction level
  - Transactions (packets)
- Checking
  - Simple I/O checking for data correctness
  - Behavioral rules for timing

32

Abstraction Levels



33

Coverage Requirements

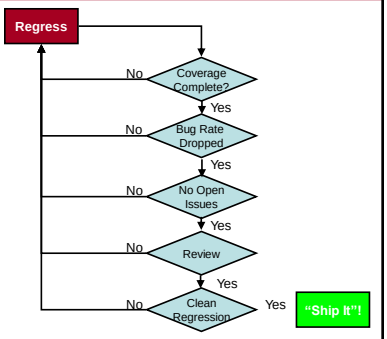
- Traditionally, **coverage is the feedback mechanism that evaluates the quality of the stimuli**
  - Required in all random-based verification environments
  - Some aspects of coverage are directly achieved in deterministic testing
- Coverage is defined as events (or scenarios) or families of events that span the functionality and code of the DUV
  - The environment has exercised all types of commands and transactions
  - The stimulus has created a specific or varying range of data types
  - The environment has driven varying degrees of legal concurrent stimulus
- Soon: **Coverage metrics**

34

Completion Criteria

These might include:

- Coverage targets
- Target metrics, e.g. bug rate drop
- Resolution of open issues
- Review
- Regression results



35

Test Scenarios (Matrix)

- Specifies test scenarios that will be used throughout the verification process
  - deterministic or random
  - Scenarios are connected to items in the coverage requirements
- Start with a basic set for the basic functionality
  - Add more tests to plug holes in coverage, reach corner cases, etc.
- Examples for calc1 design

36

Risks and Risk Management

- Complexity of design project
- Architecture and microarchitecture closure
- Resources
  - Not just verification
- New tools
- Deliveries
  - Internal
  - External
- Dependencies
  - Design availability
  - Quality of lower levels verification
  - Tools and verification IP

39

Summary

- Verification Cycle
  - Foundation for successful verification
- Verification Methodology
  - Evolution of:
    - Test patters
    - Test cases
    - Test case generators/drivers
- Verification Plan
  - The specification for the verification process.

40