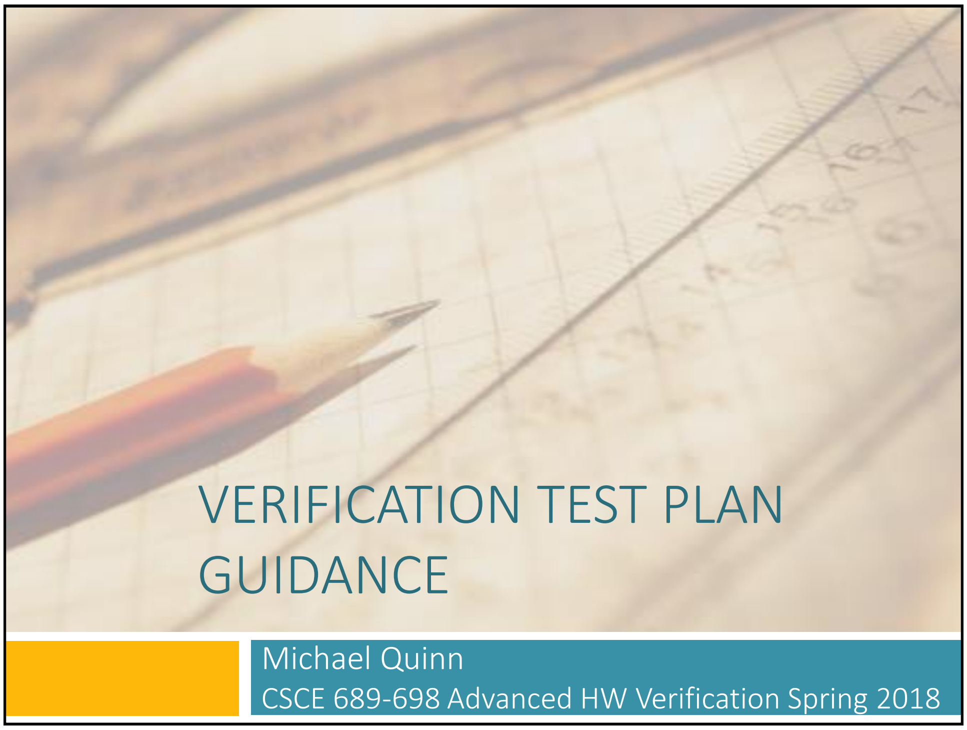
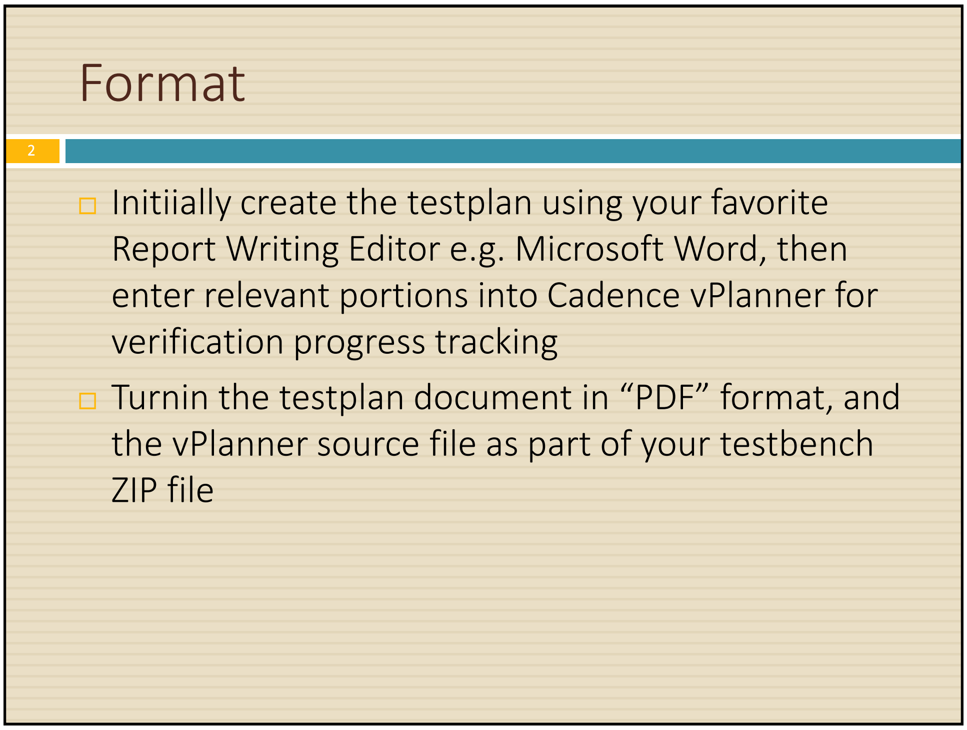


VERIFICATION TEST PLAN GUIDANCE

Michael Quinn
CSCE 689-698 Advanced HW Verification Spring 2018



Format

2

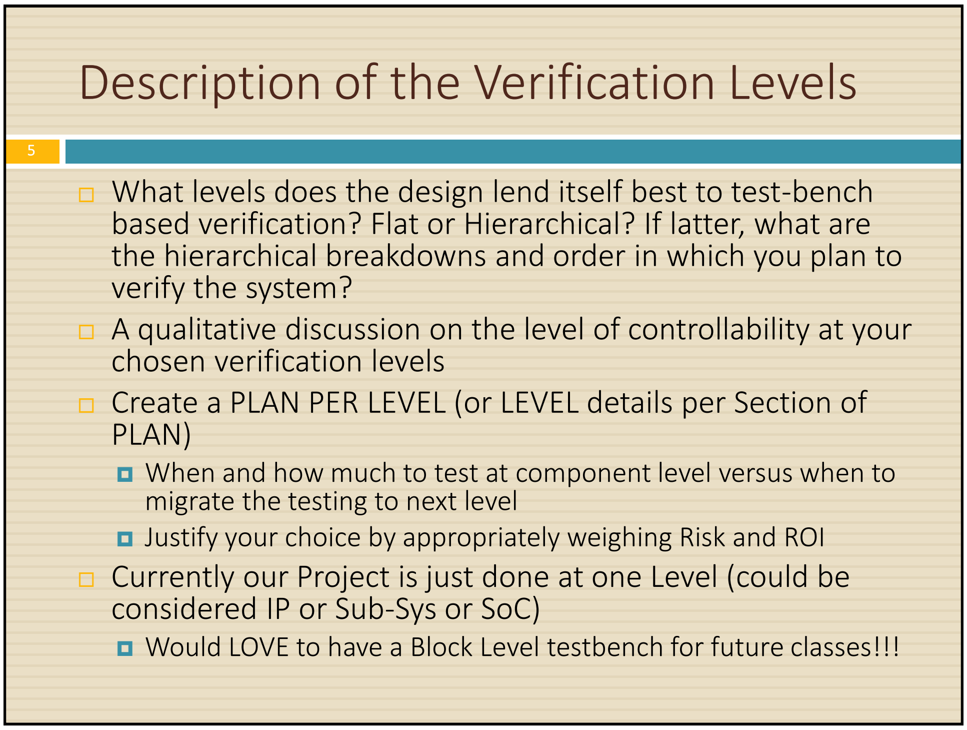
- Initiially create the testplan using your favorite Report Writing Editor e.g. Microsoft Word, then enter relevant portions into Cadence vPlanner for verification progress tracking
- Turnin the testplan document in “PDF” format, and the vPlanner source file as part of your testbench ZIP file

Organization

- Organize your testplan along the following sections
  - Overall Description of the Design
  - Description of the Verification Levels
  - Features to be Verified
  - Test Methods and Scenarios
  - Design Functionality Coverage

Overall Description of the Design

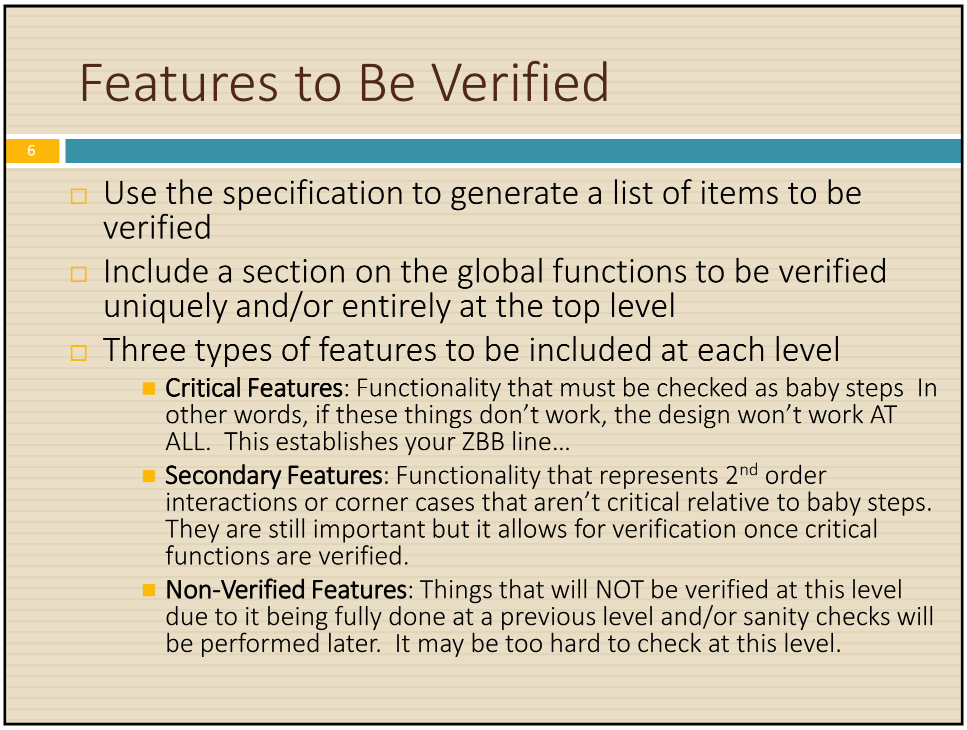
- No more than 2 pages. Is meant to reflect your understanding of the Design from the vantage point of a Verification Engineer. Remember, your role is to "BREAK THE DESIGN"
- DO NOT Recreate the HAS
- But DO cover the following
  - High Level Block Diagram showing significant cross-block interfaces
  - A qualitative assessment of Design Complexity; point out which blocks are straightforward versus the ones that contain complex functional interactions along with some reasoning



Description of the Verification Levels

5

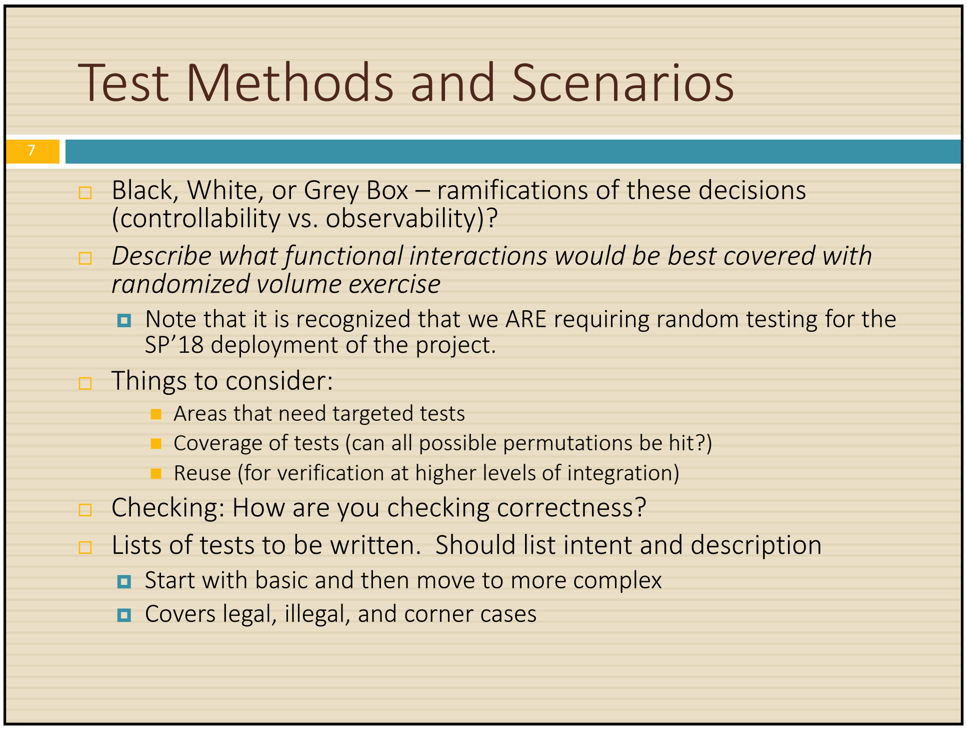
- What levels does the design lend itself best to test-bench based verification? Flat or Hierarchical? If latter, what are the hierarchical breakdowns and order in which you plan to verify the system?
- A qualitative discussion on the level of controllability at your chosen verification levels
- Create a PLAN PER LEVEL (or LEVEL details per Section of PLAN)
  - When and how much to test at component level versus when to migrate the testing to next level
  - Justify your choice by appropriately weighing Risk and ROI
- Currently our Project is just done at one Level (could be considered IP or Sub-Sys or SoC)
  - Would LOVE to have a Block Level testbench for future classes!!!



Features to Be Verified

6

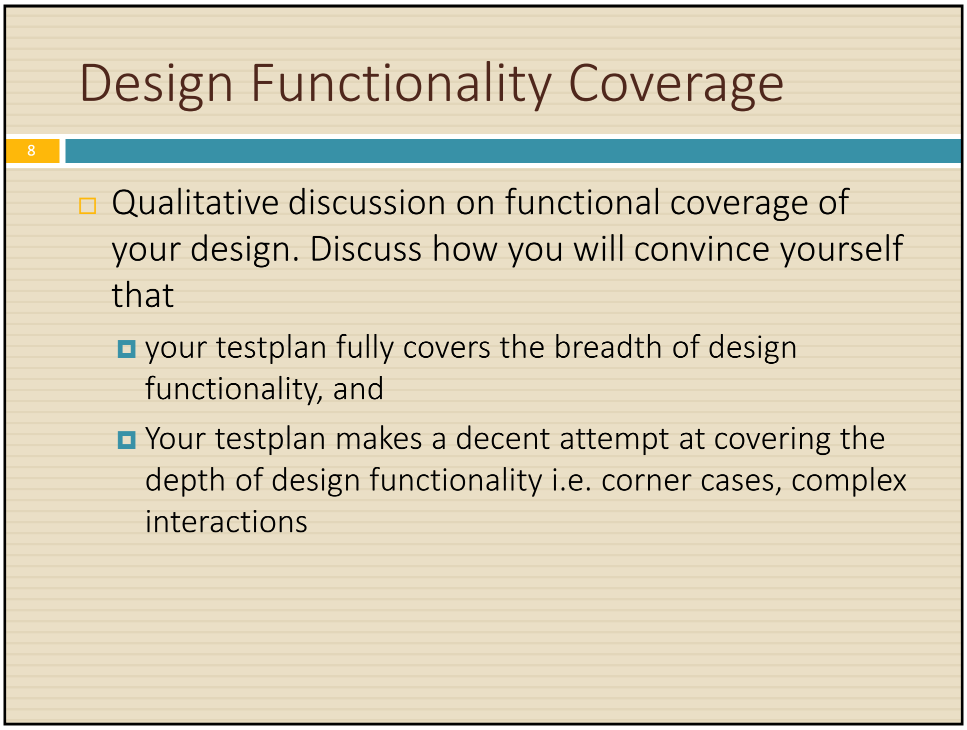
- Use the specification to generate a list of items to be verified
- Include a section on the global functions to be verified uniquely and/or entirely at the top level
- Three types of features to be included at each level
  - **Critical Features**: Functionality that must be checked as baby steps In other words, if these things don't work, the design won't work AT ALL. This establishes your ZBB line…
  - **Secondary Features**: Functionality that represents 2nd order interactions or corner cases that aren't critical relative to baby steps. They are still important but it allows for verification once critical functions are verified.
  - **Non-Verified Features**: Things that will NOT be verified at this level due to it being fully done at a previous level and/or sanity checks will be performed later. It may be too hard to check at this level.



Test Methods and Scenarios

7

- Black, White, or Grey Box – ramifications of these decisions (controllability vs. observability)?
- *Describe what functional interactions would be best covered with randomized volume exercise*
  - Note that it is recognized that we ARE requiring random testing for the SP'18 deployment of the project.
- Things to consider:
    - Areas that need targeted tests
    - Coverage of tests (can all possible permutations be hit?)
    - Reuse (for verification at higher levels of integration)
- Checking: How are you checking correctness?
- Lists of tests to be written. Should list intent and description
  - Start with basic and then move to more complex
  - Covers legal, illegal, and corner cases



Design Functionality Coverage

8

- Qualitative discussion on functional coverage of your design. Discuss how you will convince yourself that
  - your testplan fully covers the breadth of design functionality, and
  - Your testplan makes a decent attempt at covering the depth of design functionality i.e. corner cases, complex interactions