Verification Plan

Verification Plan

- ✓ Plan helps verification engineer to understand how the verification should be done.
- ✓ Plan illustrate road map for how do achieve the goal, it is a living document.
- ✓ Plan includes, introduction, assumptions, list of test cases, list of features to be tested, approach, deliverables, resources, risks and scheduling, entry and exit criteria.
- ✓ Plan could be a spreadsheet, a document or a simple text file.
- ✓ Sometimes, plan simply reside in the engineer's head which is dangerous in which the process cannot be properly measured and controlled.
- ✓ Plan also contains the descriptions of TestBench architecture and description of each component and its functionality.

Verification Plan

Generally Verification plan development is divided in two steps:
What to verify and How to verify?

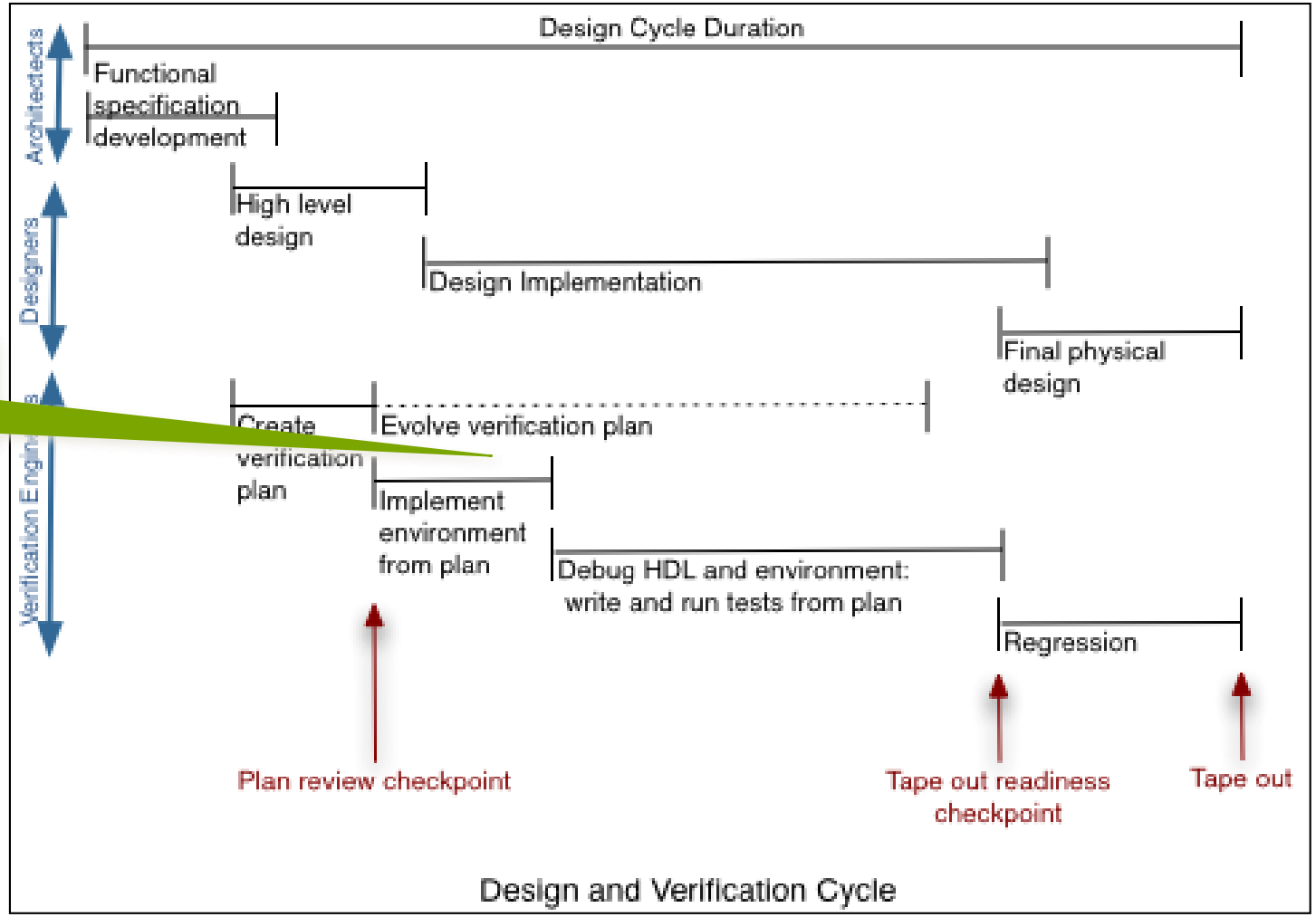
Step one: What to Verify?

List of clearly defined features-to-verify. This is called feature extraction phase.

Step two: How to Verify?

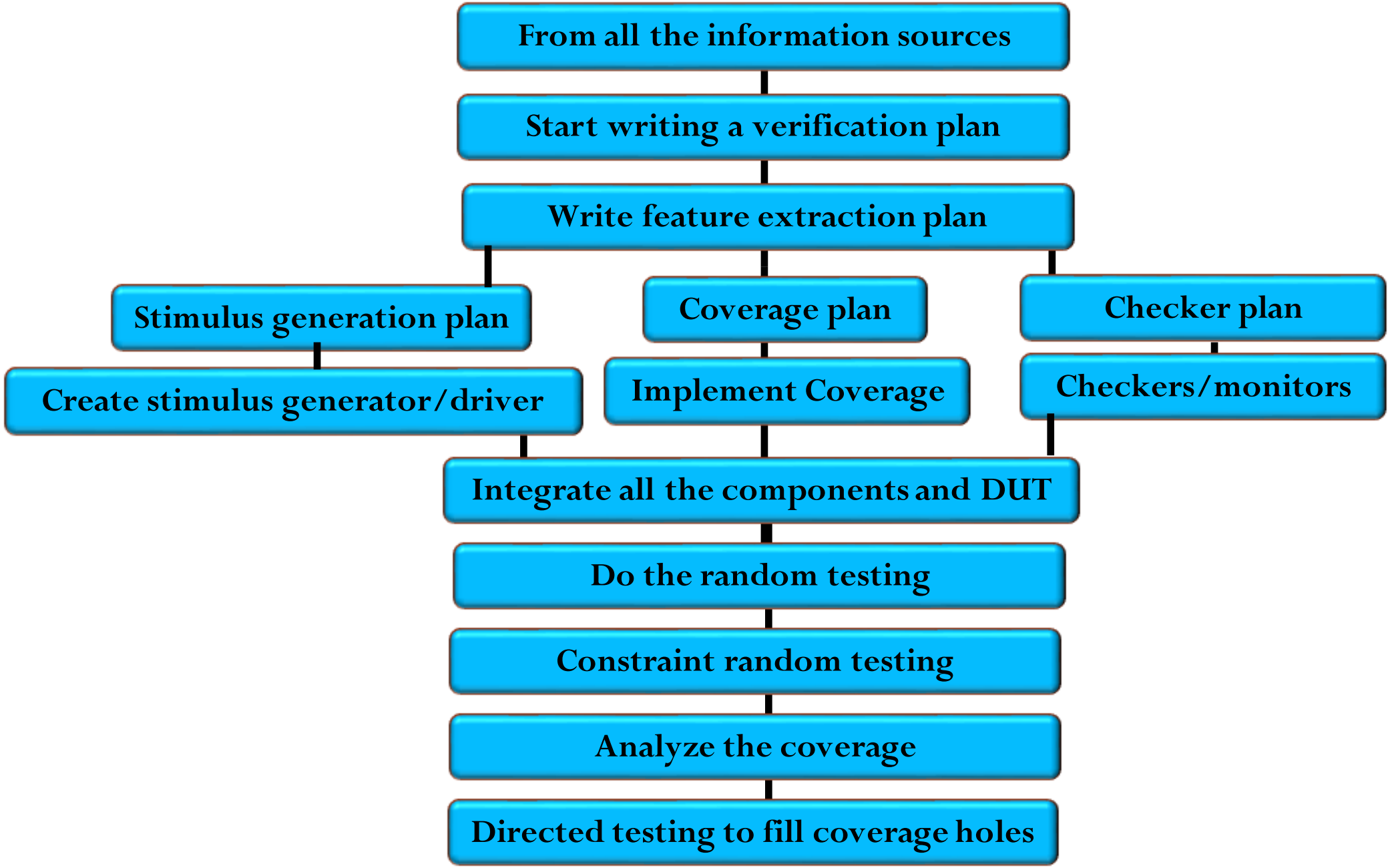
After defining what exactly need to be verified, define how to verify them.

Evolution of the Verification Plan

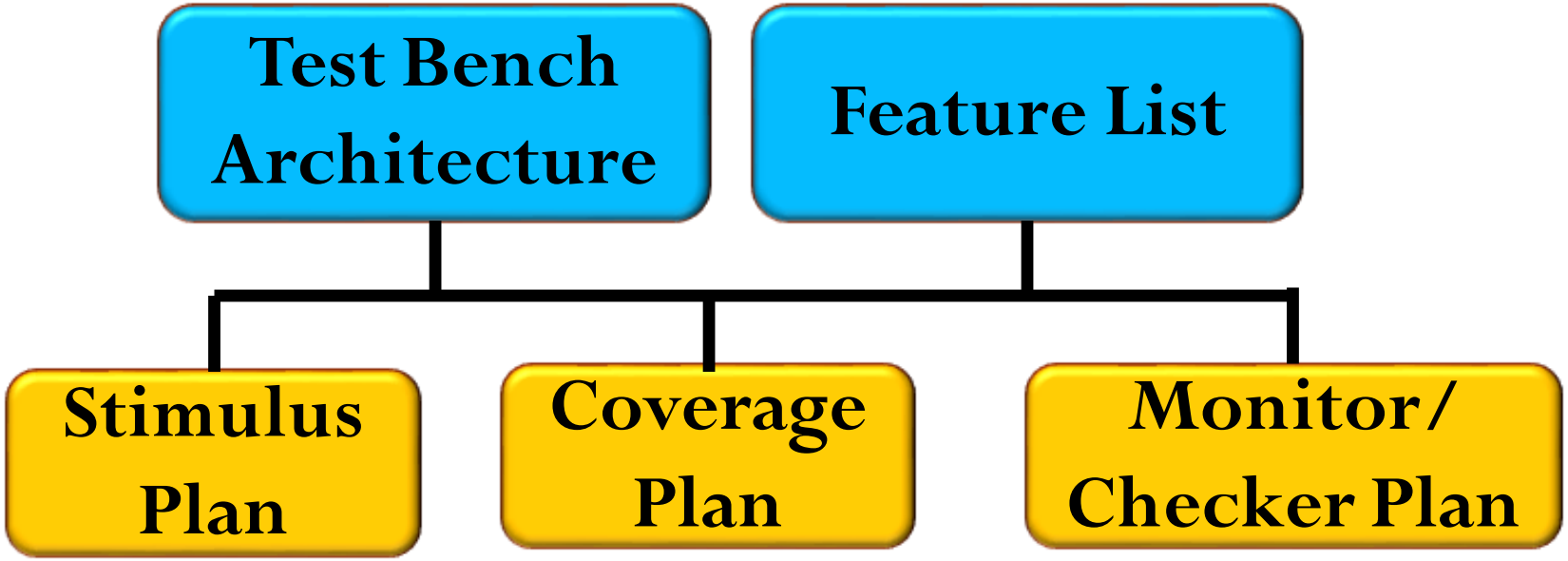


Continuously updated

Design and Verification follow the “Waterfall” flow


From all the information sources
Start writing a verification plan
Write feature extraction plan
Stimulus generation plan
Coverage plan
Checker plan
Create stimulus generator/driver
Implement Coverage
Checkers/monitors
Integrate all the components and DUT
Do the random testing
Constraint random testing
Analyze the coverage
Directed testing to fill coverage holes

Verification Plan Primarily Contains The Following



The Verification Plan

- **Project Functional Overview (**design functionality, defines all external interfaces, and lists the major blocks to be used**)**
- **Verification Approach**
- **Abstraction Levels (**behavioral, functional, gate level, and switch level**)**
- **Verification Technologies**
- **Abstraction Level for Intent Verification (**functional equivalence to the model**)**
- **Test Application Approach (**approach for applying functional tests to the design**)**
- **Results Checking (**selfchecking techniques, golden model (reference model) comparison, or comparing expected results files.**)**
- **Functional Verification Flow**

The Verification Plan

- **Test Definitions**
- **Testbench Requirements**
- **Models (**Model sources, Existing models, Derived models, Authored models**)**
- **Testbench Elements (**Checkers, Transaction verification modules, Stimulus**)**
- **Verification Metrics (**Capacity metrics, Quality metrics**)**
- **Regression Testing (**overnight, continuously, triggered by change levels and so on **)**
- **Issue Tracking and Management**
- **Resource Plan (**human resources, machine resources, and software tool resources.**)**
- **Project Schedule (**key benchmarks and completion dates**)**

The Verification Plan

1. Binds a requirement in the specification to the test(s)

Specification:

The I^2C bus will read/write the config registers

Test Plan:

- Check reset values
- Write a writable config reg
- Write a read-only config reg
- Read a config reg
- Read a memory mapped input
- Stall an I^2C transaction

2. Technique for testing feature

a. Block, hybrid, or System level
b. Directed or random
c. Assertions
d. Emulation (FPGA, accelerator, etc)
e. Verification IP
f. Self checking or visual

Source: Chris Spear: System Verilog for Verification