Chap 7
Simulation Management

教育部顧問室
「超大型積體電路與系統設計」教育改進計畫
EDA聯盟編製

Outline

- Behavioral models
- Pass or fail?
- Managing simulations
- Regression
- Summary

Outline

- ***Behavioral models***
- Pass or fail?
- Managing simulations
- Regression
- Summary

Behavioral Models

- Demonstrate how behavioral models can benefit a design project.
- Show how to properly model exceptions.
- Explain how to demonstrate a behavioral model to be an RTL model.

Why Behavioral Models?

- Behavioral models are used to debug testbenchs.
- Behavioral models are available earlier than RTL models.
- Behavioral models run faster then RTL models.

Synthesizable vs. Behavioral Models

- Synthesizable model
  - A model that can be automatically translated into a gate-level netlist by a synthesis tool, such as Synopsys' Design Compiler. It may also be called RTL model.
- Behavioral model
  - A model that describe the black-box functionality of a design.

- Using behavioral descriptions for testebcnhs is easily acceptable by most design engineers.
  - The testbench will never be implemented in hardware.
- To write a proper behavioral model, you have to focus on the functionality, not the implementation.

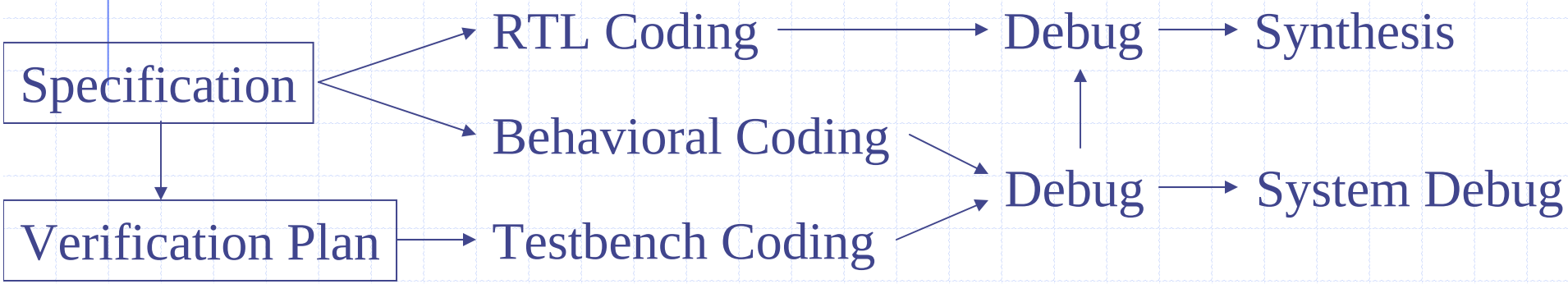
Characteristics of Behavioral Models

- They are partitioned for maintenance.
- They don’t use a clock.
- They don’t use FSMs.
- Data can remain at a high-level of abstraction.
- Data structure are designed for ease-of-use, not implementation.
- Their interfaces are implemented using bus-functional models.

Benefits of Behavioral Models

- Audit of the specification.
- Development and debug of testbench in parallel with the RTL coding.
- System verification can start earlier.

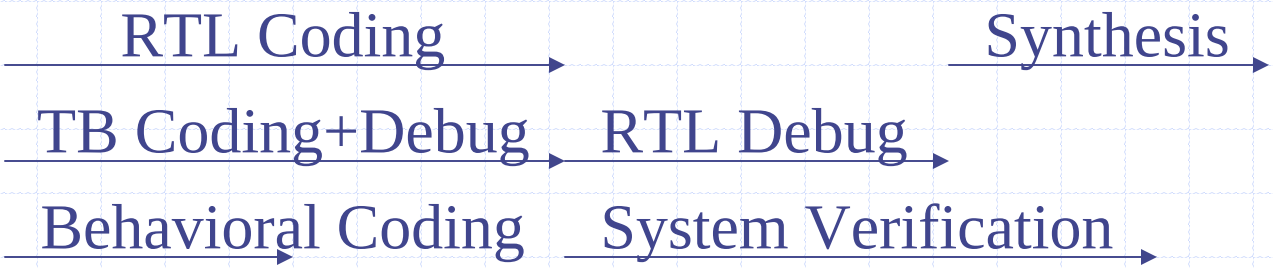
Design Process Including Behavioral Model



The Effect of Behavioral Models on a Project Timeline

Without Behavioral Model:

With Behavioral Model:



Behavioral Models are Faster

- Fast to design.
- Fast to debug.
- Fast to simulate.
- Fast to bring to market.

Modeling Reset

- Disable all the blocks in the model using *disable* statement.
- Replace *initial* blocks with *always* blocks with an infinite *wait* statement at the bottom.
- Encapsulate all *disable* statements into a single task.

Demonstrating Equivalence

- Equivalence checking
  - A RTL model ⟷ A gate-level model
- Equivalence demonstration
  - A behavioral model ⟷ A RTL model
  - The only way is to verify both of them using the same test suite.

Outline

- Behavioral models
- ***Pass or fail?***
- Managing simulations
- Regression
- Summary

Pass or Fail?

- Describe how the ultimate failure or success of a self-checking testbench is determined.

Methodology

- Inject errors deliberately.
- Produce a different label to easily distinguish them from real errors.
- Parse the simulation output log file.
- Compare the number of errors encountered against the expectation.
- To facilitate the implementation of the parser, use a consistent error format.

Outline

- Behavioral models
- Pass or fail?
- ***Managing simulations***
- Regression
- Summary

Managing Simulations

- How do you bring all of components together in a single simulation?
- How can you reproduce a simulation?
- How do you make sure that what you simulate is what will be built?

Configuration Management

- Deal with the particular set of models you decide to use in a particular simulation.
  - Identify the testcase, the test harness, and the model of the design under verification.
  - The most efficient way is to provide a script that expands a testcase name and an abstraction level for the design under verification into their respective simulation components.

Verilog Configuration Management

- Five different ways to include a source file into a Verilog simulation.
  - Specify the filename on the command line.
  - Specify the name of a file containing a list of filenames, using the *–f* option.
  - Specify a directory to search for files likely to contain the definition of missing modules, using the *–y* option.
  - Specify the name of a file that may contain the definition of missing modules, using the *–v* option.
  - Include a source file inside another using the '*include* directive.

SDF Back-Annotation

- SDF file are used to model accurate delays.
- SDF annotation can take a long time.
- Use compile-time back-annotation whenever possible.
- Concatenate testcases to minimize back-annotation.
- Use a single task or procedure to control each testcase.

SDF Back-Annotation (cont.)

Output File Management

- Simulations produce output files.
- Multiple simulations can clobber each other's files.
- Specify output filenames on the command line in your simulation run script.

Output File Management (cont.)

Outline

- Behavioral models
- Pass or fail?
- Managing simulations
- ***Regression***
- Summary

Regression

- A regression suite ensures that modifications to the design remain backward compatible with previously verified functionality.

Running Regressions

- Regressions are run at regular intervals.
- Testbenchs may have a fast mode to speed-up regressions.
- Use a script to run regressions.

Regression Management

- Check out a fresh view with local copies.
- Put a time-bomb in all simulations.
- Using the time-bomb procedure.
- Don’t rely on a time-bomb for normal termination.
- Automatically generate a report after each regression run.

Outline

- Behavioral models
- Pass or fail?
- Managing simulations
- Regression
- ***Summary***

Summary

- Behavioral models can be used to accelerate the verification of a design.
- Behavioral models let system-level verification start sooner.
- From determining success or failure to configuration management to regression simulation.