



## Department of Electrical and Computer Engineering Utah State University

### ECE 5930/6930: VLSI Testing and Verification Preparer: Shayan (Sean) Taheri

#### Lab 05

Assigned Date: Month/Day/Year

Due Date: Month/Day/Year

---

This lab consists of two sections and the goal is verifying three designs using both hand calculation (combinational and sequential algorithms) and TetraMAX tool.

Before starting the lab, you need to be familiarized with some commands for this lab.

#### Effort Level in Fast Sequential Only

Fast Sequential Only is for designs that consist of sequential elements. Since these elements are triggered by clock signal, we may have different values for some wires and different states for some sequential elements after each clock cycle. Effort Level of Clock Signal is defined as the parameter in ATPG tools that makes different levels for analyzing circuits (based on values of wires and states of sequential elements). In the following, the command that specifies the effort level of clock signal has been provided:

#### Command:

```
set_atpg -capture_cycles N
```

// N = Number of Clock Cycles that are applied to the design.

#### Fault List

Before generating test patterns, you must initialize the fault list of the tool. TetraMAX attempts to generate a test pattern to test all the faults contained in the fault list.

Some key commands related to fault list have been specified in the following.

1) To add or remove specific blocks, instances, gates, or pins to/from the fault list:

```
add_faults pin_path -stuck <0|1|01>          // For Adding Fault!
```

```
remove_faults pin_path -stuck <0|1|01>       // For Removing Fault!
```

Important: A faulty wire affects some components that have the faulty wire in their inputs. So, for adding a faulty wire in the fault list, you can find the pin's path that is under the effect of the faulty wire.

2) For writing "Collapsed" faults:

```
write_faults file_name -all -replace -collapsed
```

## Verifying Sequential Circuits

To verify sequential circuits, you should use "Fast Sequential Only" and "Full Sequential Only" modes of automatic test pattern generation. Meanwhile, you should enable the clock signal along the procedure that you followed in previous labs. For enabling clock signal, the following command should be used before running Design Rule Checking:

```
add_clocks <0 | 1> {pin_names_list} [-timing {period LE TE measure_time}] [-unit <ps | ns>]
```

This command is used to specify and define clocks. Top-level ports that can change the stored state of sequential devices are defined as clocks. This includes not only the traditional active-high and active-low clocks, but also the asynchronous sets/resets and RAM write lines. For ATPG purposes, a "clock" is a pulsed input port that can change the state of a flip-flop, latch, or RAM. Explanation of each option can be found in here:

**<0 | 1>:** It specifies the off state of the clock. The off state is the logic level (either 0 or 1) for which the clock port is inactive. It results in holding the state of sequential devices such as flip-flops, latches, or RAMs. An active-high clock has an off state of **0** and an active-low clock has an off state of **1**.

**{pin\_names\_list}:** It specifies the port name that should be treated as a clock. You can specify multiple port names using a single command, but they must all share the same off state.

**[-timing {period LE TE measure\_time}]:** It specifies the configuration and specification of a clock signal. Here, the options of this command are explained:

**period** = It defines the test cycle period associated with the clock.

**LE** = It is the time of the leading edge of the clock.

**TE** = It is the time of the trailing edge of the clock.

**measure\_time** = It specifies the time within the test cycle at which the design's output pins is measured.

Notice: If measure\_time is less than LE, then a pre-clock measure protocol is implied. If measure\_time is greater than TE, then an end-of-cycle measure protocol is implied.

**unit <ps | ns>:** It specifies the time units used with the -timing option and should only be specified if -timing is used. By default, the time unit is ns. All units must be the same for all add clock commands.

Important: If you want to use the default values for options, you can follow the GUI procedure:

- 1) Click on DRC tab and go to Quick STIL.
- 2) Make sure the View has been set on Clocks. Then, select your desired clock signal in Port Name list.
- 3) Select **0** for list of “Off” option and after adding the configured clock signal, click on OK.

Note: Full Sequential Only mode of ATPG can be highly useful in validating “Full Scan Equipped Circuits”. For more information about these circuits, you can read about “Reconfigurable Scan Networks”.

## Section A

For completing this section, you should do seven tasks according to the following:

- 1) Figure out the schematic of circuit 1 using its net-list.
- 2) Derive a combinational circuit by replacing each flip-flop with a buffer. This is known as the pseudo-combinational transformation.

Note: BUFFD1BWPHT (I, Z) // High Threshold Voltage Buffer!

- 3) Derive test vectors for the fault **n2** (stuck-at-0) and the fault **NO2** (stuck-at-1) of the obtained circuit using PODEM algorithm.
- 4) Repeat task 3 using Basic Scan Only mode in TetraMAX tool.
- 5) Verify the detection of the faults in original circuit and obtain required test sequences using Time-Frame Expansion method.
- 6) Repeat task 5 using Basic Scan Only, Fast Sequential Only and Full Sequential Only modes of TetraMAX in sequence.
- 7) If there are differences between the results of Combinational ATPG for Pseudo-Combinational Circuit and the results of Sequential ATPG for Original-Sequential Circuit, what can be the reason for that?

## Section B

This section consists of the following tasks:

- 1) Figure out the schematics of Circuit 2 and Circuit 3.

- 2) Write the collapsed fault list of Circuits 1, 2, and 3.
- 3) Load the written fault list of each design and run Basic Scan Only, Fast Sequential Only and Full Sequential Only modes to generate test vectors for three circuits.
- 4) What are differences between three circuits based on their schematics?
- 5) What can be understood from analyzing all fault coverage reports for each circuit after applying three ATPG modes?
- 6) If there are differences between the final fault coverage reports of three circuits, what are the reasons for that?

### **What to deliver for this lab:**

- 1) Provide schematics of three circuits.
- 2) Derived test vectors for Circuit 1 after applying PODEM algorithm.
- 3) Derived test vectors and fault coverage report for Circuit 1 after applying Basic Scan Only mode.
- 4) Obtained test sequences for Circuit 1 after using Time-Frame Expansion method.
- 5) Obtained test sequences and all fault coverage reports for Circuit 1 after applying three ATPG modes.
- 6) Explanation for task 7 of section A.
- 7) The collapsed fault lists of three circuits.
- 8) Generated test vectors and all fault coverage reports for three circuits after applying three ATPG modes on collapsed fault lists.
- 9) Explanation for tasks 4, 5, and 6 of section B.

**Good Luck**