



Lab3: Analyzing Faulty Signals in Digital Circuits using TetraMAX

Posted: 02/09/2017 – Due: 02/14/2017 via Canvas

In this lab, your task is to apply a set of test vectors which are generated by the TetraMAX tool to a given circuit. You will then collect information about the faults that can be detected by these test vectors.

Before starting this lab, you should be familiar with the following TETRAMAX commands.

According to the TetraMAX user guide, these two commands assist in analyzing the faults of a given circuit:

1) **report_faults instance_name|pin_pathname -stuck <0|1|01>|-slow<r|f|rf>**

instance_name: Reports all faults associated with the selected instance name.

pin_pathname: Reports faults associated with the selected pin pathname.

-stuck <0|1|01>: Reports only the stuck-at-1 or stuck-at-0 faults, or both stuck-at-faults associated with the specified `pin_pathname`. The default is 01 (both).

-slow <r|f|rf>: Reports slow-to-rise, slow-to-fall, or both transition delay faults for the specified `pin_pathname`.

2) **report_faults -pattern_id n // n = 0, 1, 2, ...**

This command reports all faults that can be detected by the test pattern with `pattern_id n`. However, this command excludes any faults which are detected by patterns with a `pattern_id` less than `n`. In order for this command to report correct results, the `run_fault_sim -detected_pattern_storage` command must first be issued. The `run_fault_sim` command must be re-run if the patterns or fault configuration are modified.

Starting the Lab

First, create a folder to store the lab results. Copy `circuit-1.v` from the course website into this folder, then open `tetramax`.

1. `tmax &`

Then execute the following commands in the Command-Line Text Field to correctly configure the environment:

2. `read_netlist
/opt/software/cadence/library/tc45nm/verilog/HVT/tc45gsbwphvt.v -`

```

library
3. read_netlist /*** (your directory)/circuit-1.v
4. run_build_model circuit1
5. run_drc
6. remove_faults -all
7. add_faults -all
8. run_atpg
9. run_simulation
10. Click on the "Schematic View" icon at the top of the screen. Then, click on the "Show" icon that
    appears in the schematic view window, scroll down and select: "ALL"

```

The following commands show an example of how to view all faults which are detected by pattern 1, but not pattern 0. It also shows how to examine a particular fault to see if it is detected by the given patterns included in the .stil file.

```

11. write_patterns name.stil -format stil
12. reset_state
13. set_patterns -delete
14. set_patterns -external name.stil
15. run_simulation
16. run_fault_sim -detected_pattern_storage
17. report_faults -pattern_id 1
18. report_faults N3 -stuck 1

```

Notice 1: Command number "17" reports the faults detected by the selected pattern (= pattern 1). In this circuit, we have 8 generated patterns (0-7) and we can analyze each of them by this command. IMPORTANT: the fault report for faults detected by pattern 7 will exclude any fault detected by patterns 0-6.

Notice 2: Command number "18" reports the faults associated with the selected pin pathname (pin = N3). The inputs in this circuit are from N1 to N5 and the associated faults of any of them can be achieved by this command. This report will show whether the fault is detected by the given input patterns. If it is detected, it will report the pattern_id for the first pattern which detects the fault.

Notice 3: In order to identify ALL faults detected by a specific pattern, that pattern must be stored in pattern_id 0. You can manually modify the name.stil file for each test_vector in order to generate these reports..

DELIVERABLES:

Please submit a text file including the following via canvas:

1. The output generated by : report_faults -pattern_id X for all the patterns generated by the run_atpg command. (don't re-arrange the .stil file to identify all faults from every pattern)
2. The output generated by: report_faults -NETID -stuck 0/1 for both sa0 and sa1 faults for all inputs in circuit1