

TetraMAX and Simulation

Posted: 01/30/2018 – Due: 02/01/2017 via Canvas

In this section, the steps of starting a project and its simulation have been declared.

Start a Project in TetraMAX:

1. Start the Synopsys TetraMAX GUI in a directory that you want to do your project by typing: `tmax &`

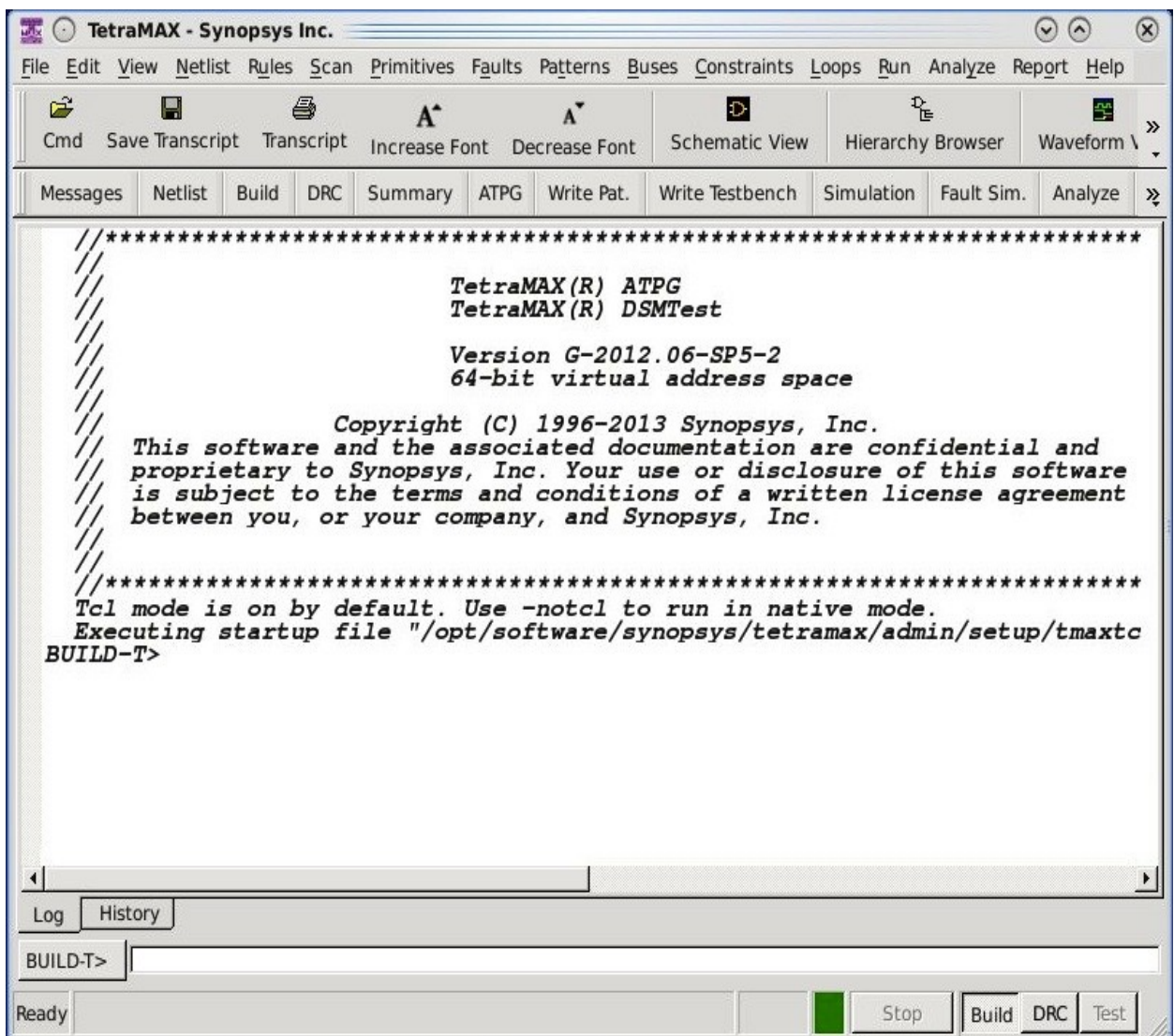


Figure 1

2. Once the GUI opens, notice that you are in the “BUILD” mode.
TetraMAX has 3 modes:

BUILD, DRC, and TEST

Note: You cannot go from BUILD to TEST. You must go in the BUILD -> DRC -> TEST order.

* Please follow the following steps in Command-Line Text Field:

Note: You are able to do all of these steps in GUI mode.

3. Load the Library (Note: We will use “tcbn45gsbwphvt.v” library in this project).

```
Command: read_netlist  
/opt/software/cadence/library/tcbn45nm/verilog/HVT/tcbn  
45gsbwphvt.v -library
```

4. Load your synthesized circuit (Note: the “circuit-1.v” file).

```
Command: read_netlist /*** (your directory)/circuit-1.v
```

5. BUILD: It sets the relevant parameters and builds the in-memory simulation model from the design modules that have been read in.

```
Command: run_build_model circuit1
```

*Note: “circuit1” is the name of the top module

6. DRC (Design Rule Check): It is checking to see if your design is in fact valid “Testable” code.

```
Command: run_drc
```

7. You should specify the faults by the following commands:

Commands:

```
remove_faults -all
```

`add_faults -all`

8. ATPG (Automatic Test Pattern Generation): It is used to set up and generate ATPG patterns for a current set of faults using a specified pattern source set.

Command: `run_atpg`

9. Starting Simulation:

Command: `run_simulation`

10. Set up Pin Data Type:

Command: `set_pindata -good_sim_results 0`

11. Finally, 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”. Now, you should be able to see the circuit and its patterns. You can click on “Zoom Full” to see the circuit completely.

Note 1: You will see a schematic of your circuit with Pattern number 0 (of all the patterns), showing you what input is at the input PINS and what “good - non - faulted” data should be at the output PINS. You can change the “Setup” to show the other patterns and their expected data.

Note 2: You will see each input has repetitive value, this is because of this type of pattern generation (Good Sim Results) and you can change it in the “Setup” to whichever type you need.

- Please refer to figure 2.

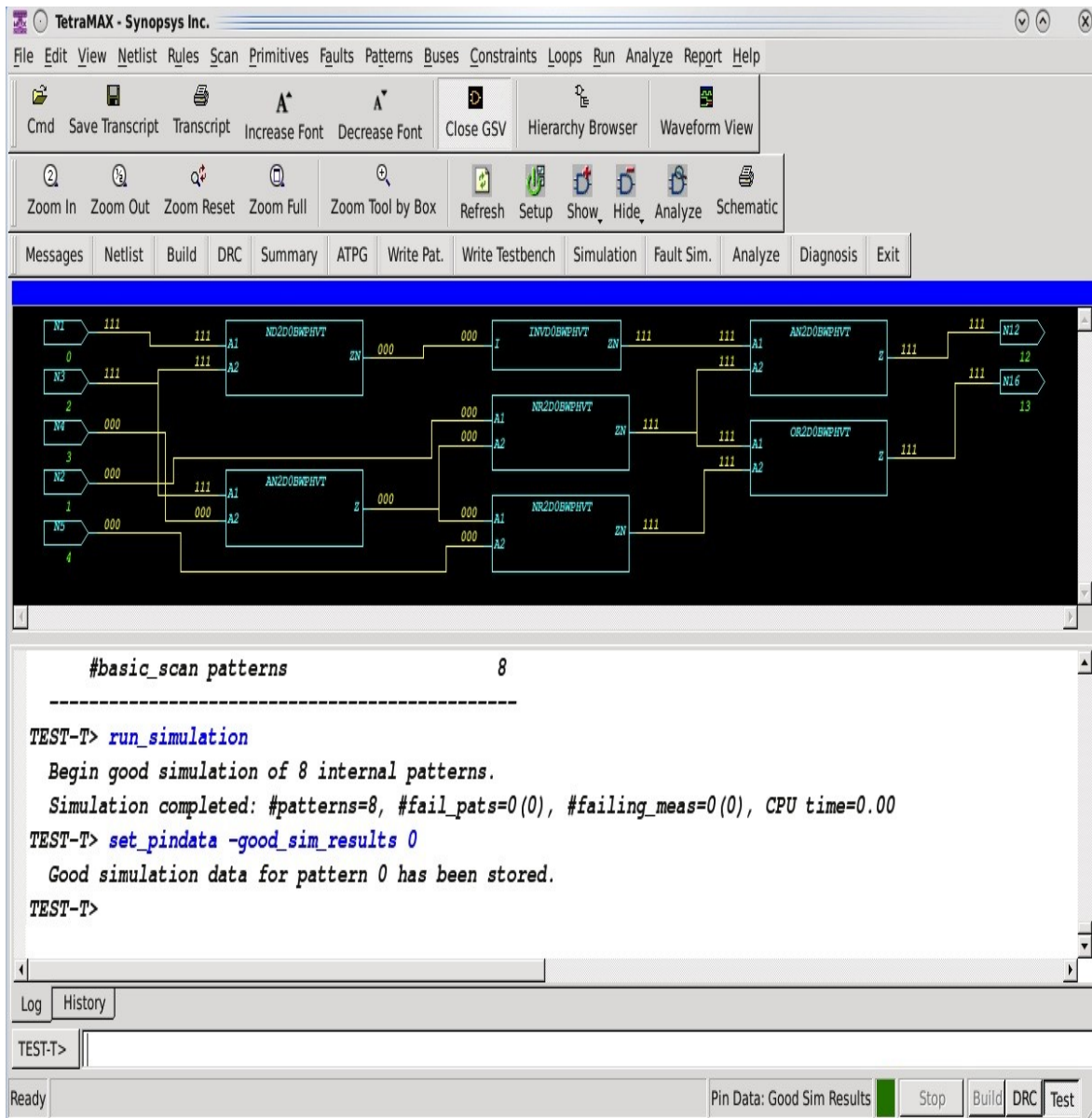


Figure 2

12. Changing Pin Data Type for viewing another mode (Fault Sim Results).

Click on the "Set up" button, then:

- Set the Pin Data Type to: "Fault Sim Results".
- Before pressing OK, click on the "set parameters" button.
- Specify the "Gate id": put 0 (or 1, 2, 3 or etc.) for related inputs.
- Set the "stuck-at" fault to be 0 (or 1).
- Set the "pattern no." to be 0, click OK.

Your schematic will update and show the stuck-at-0 faults captured by pattern number 0.

Note: In “Setup” window, you can see some Pin Data Types which each has its own application.

13. You can refresh the schematic of your circuit after each change by the following command.

Command: refresh_schematic

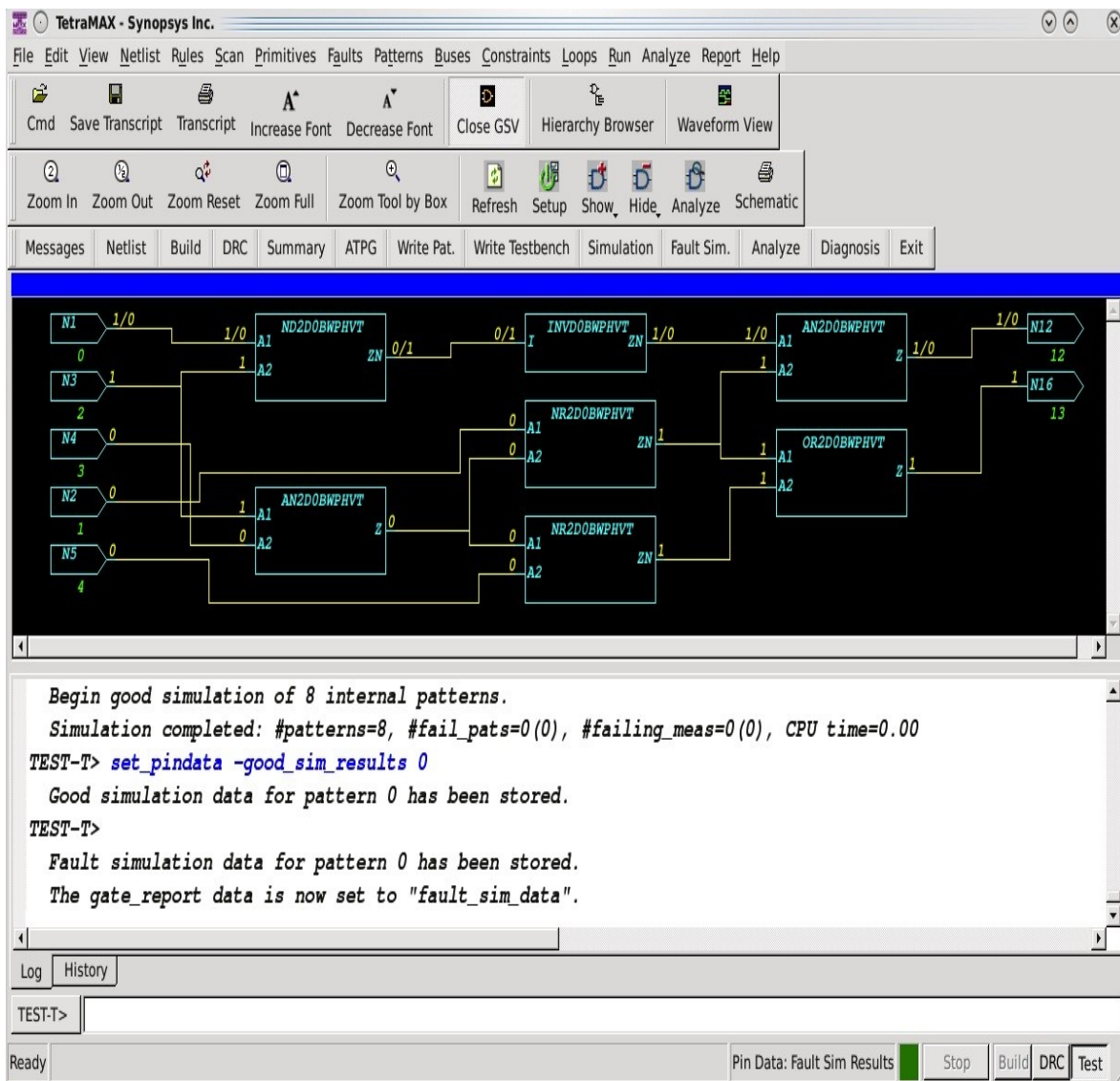


Figure 3

14. You are able to see the specifications of your generated patterns by the following command.

Command: `report_patterns -all`

15. If you want to save the generated patterns, you can use the following command. You can write the generated patterns file in different formats.

Command: `write_patterns name.stil -format stil`

Notice: Don't close TetraMAX otherwise you should do all of these steps again! You don't need to close it for changing the patterns!

Hint: If you want to come back later or you need to close tetramax for some reason, you can write .tcl scripts to run in tetramax. Each line of the script should be a tetramax Command, and you should save the script with a .tcl extension. You can run the script in tetramax by using the Command: `run_commands YOUR_FILE.tcl`.

16. Make your own patterns:

After writing the generated patterns file, please open it with a text editor and then you will find their definitions, specifications, and etc. Next, you should go to the end of the file to see the values of generated patterns. In there, you can change their values to whichever values you want.

Note 1: You shouldn't change the values of “_pi” and “_po” of “precondition all signal” section. For changing each pattern, you can change the value of “_pi” of each pattern (placed in Call “capture”). Then, you should change each bit of “_po” of the changed pattern to ‘X’, because we don't know them in this step.

Notice: For this assignment, you can just work on “pattern 0” and there is no need to change all of them.

Note 2: Each bit of a “_pi” belongs to one of the inputs of the circuit and its value can be ‘0’, ‘1’ or ‘Z’.

- Please refer to figure 4.

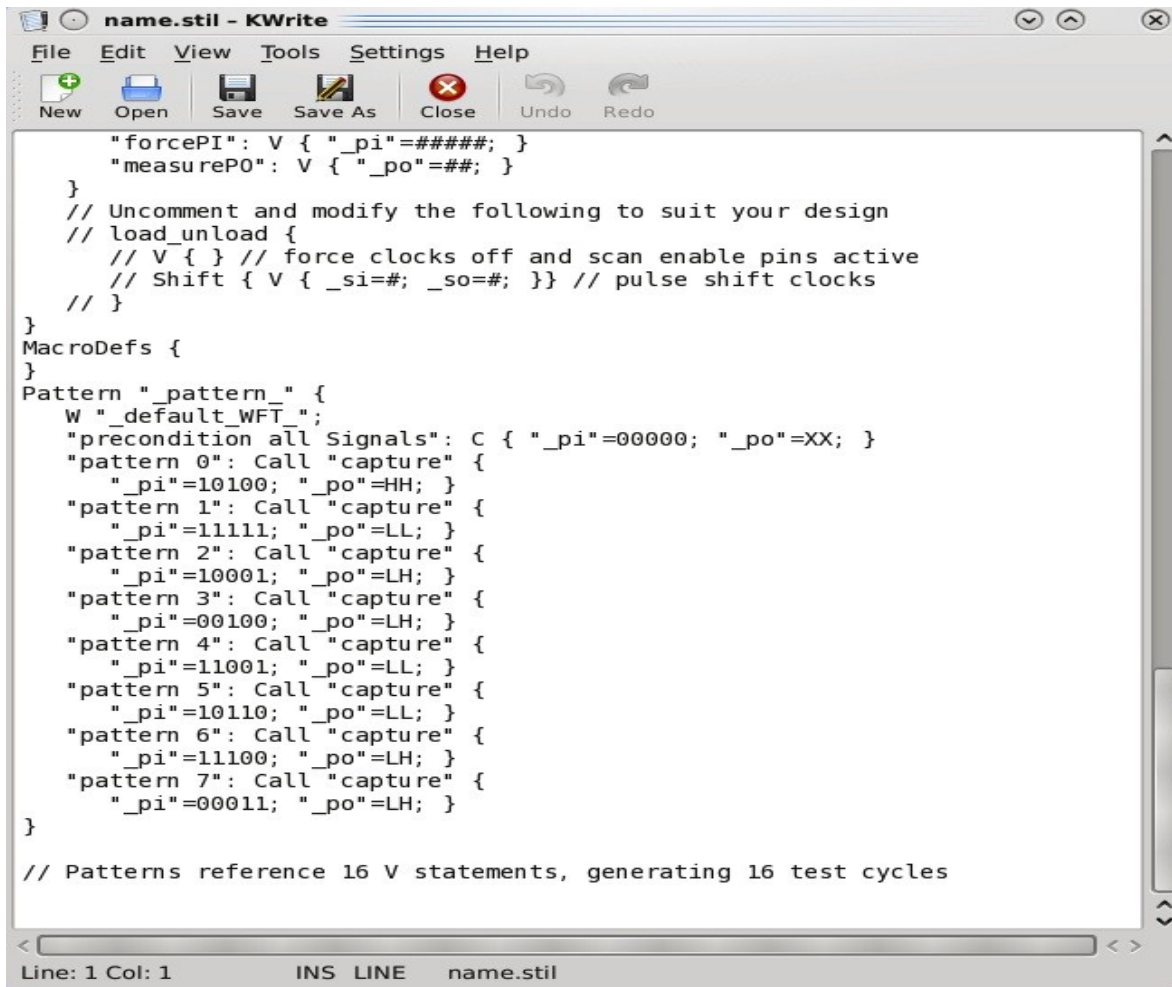


Figure 4

17. You can set the inputs to your own patterns by the following commands.

Commands:

```

set_patterns -delete
set_patterns -external /*** (your directory)/name.stil
-sensitive -append
run_simulation
set_pindata -good_sim_results 0
refresh_schematic

```

Note: The type of Pin Data can be changed as it was mentioned before.

REMINDER (For seeing different patterns):

Click on the "Setup" button, then:

- a. Set the Pin Data Type to: "Fault Sim Results".
- b. Before pressing OK, click on the "set parameters" button.
- c. Specify the "Gate id": put 0 (or 1, 2, 3 or etc.) for related inputs.
- d. Set the "stuck-at" fault to be 0 (or 1).
- e. Set the "pattern no." to be 0 (or 1, 2, 3 or etc.), click OK.

What to turn in for this lab:

Please submit the following via canvas:

- 1) Your name.stil file where you use the patterns "10000" and "10100".
- 2) A snapshot of the schematic showing the good simulation with the above patterns.