Lab1: Introduction to TetraMAX

Posted 1/16/2018, Due 1/23/2018 in canvas

TetraMAX is used to check testability design rules and to automatically generate manufacturing test vectors for a logic design. It is included in RTL synthesis and test category of Synopsys tools.

TetraMAX is a high-speed, high-capacity automatic test pattern generation (ATPG) tool. It can generate test patterns that maximize test coverage while using a minimum number of test vectors for a variety of design types and design flows. It is well suited for designs of all sizes up to millions of gates.

TetraMAX can read design netlists in Verilog, VHDL, and EDIF formats; and test protocol information in STIL format. Also, it can write pattern files in a variety of standard and proprietary formats: WGL, STIL, Verilog, VHDL, Fujitsu TDL, TI TDL91, and Toshiba TSTL2. This tool supports the following design-for-test (DFT) styles:

- Various scan flip-flop types (multiplexed flip-lop, master, slave, transparent latch, and so on)
- Internal, non-decoded three-state buses
- Bus keepers
- RAM and ROM models
- Proprietary and standard test controllers (such as IEEE 1149.1-compliant boundry scan)

ATGP Modes

TetraMAX offers three different ATGP modes:

- Basic-Scan ATGP: In this mode, TetraMAX operates as a full-scan, combinational-only ATGP tool. To get high-test coverage, the sequential elements need to be scanning elements.

Fast-Sequential ATGP: This mode provides limited support for partial-scan designs. It allows data to be propagated through non-scan sequential elements in the design. Using the "-capture_cycles" option of the "set_atpg" command can enable it.

Full-Sequential ATGP: It is like Fast-Sequential ATGP with this difference that it can increase test coverage in partial-scan designs. Using the "-full_seq_atgp" option of the "set_atpg" command can enable it.

Supported Fault Models:

TetraMAX supports test pattern generation for five types of fault models:

Stuck-At: This model is the standard model for test pattern generation.

Transition: It is used to generate test patterns to detect single-node slow-to-rise and slow-to-fall faults.

Path Delay: The path delay fault model tests and characterizes critical timing paths in a design.

IDDQ: This model assumes that a circuit defect will cause excessive current drain due to an internal short circuit from a node to ground or to a power supply.

Let's Start TetraMAX

1) Setting up your run directory

Log on to one of the vlsi-xx machines using your account.

2) The LINUX initial file (.bashrc)

Each LINUX user has an initialization file in their home directory

This directory is called ".bashrc" and is hidden.

To open the file, first go to your home directory and then type:

"**gedit ~/.bashrc**"

(Note: gedit is a text editor! You can use any other editor of your choice)

Your .bashrc file will look something like this:

```
# .bashrc
# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
```

Add the following at the bottom of the ".bashrc" file:

```
# User specific aliases and functions
export LM_LICENSE_FILE=/opt/software/elicense.dat:/opt/software/qed license.dat
export PATH=/opt/software/mentor/modeltech/bin:/opt/software/cadenc
e/edi110/bin:/opt/software/synopsys/synthesis/bin:/opt/softw
are/synopsys/coretools/bin:/opt/software/synopsys/hspice/bin :/opt/software/synopsys/
tetramax/bin:$PATH
export MODELSIM=/opt/software/mentor/modeltech/modelsim.ini
```

Save and then exit the file

Make sure there are no line breaks in the LICENSE and PATH commands.

Type the command: "source ~/.bashrc" and then the new ".bashrc" file will take effect.

3) Running TetraMAX

Please type "**tmax &**" in terminal. By default, the command starts TetraMAX in its GUI mode.

Starting and Stopping the TetraMAX GUI from shell mode:

To use TetraMAX in shell mode use the following command**: "tmax -nogui".**

To switch from shell to GUI mode, use command "**gui_start**". The command line should look as follows:

BUILD-T> gui_start

To exit the GUI mode, use command "**gui_stop**".

Note: Please refer to Figure 1!

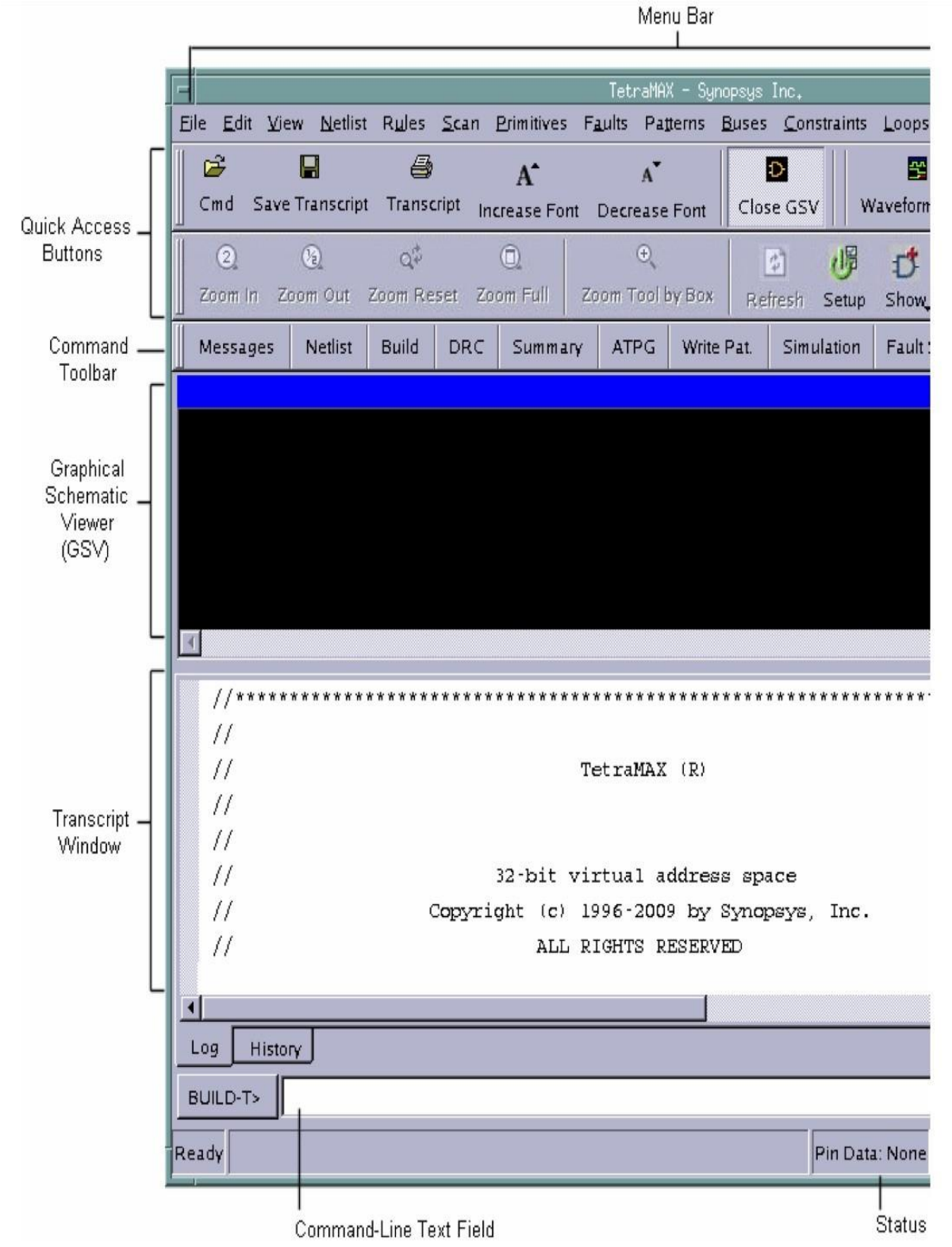


Figure 1: TetraMAX GUI Main Window

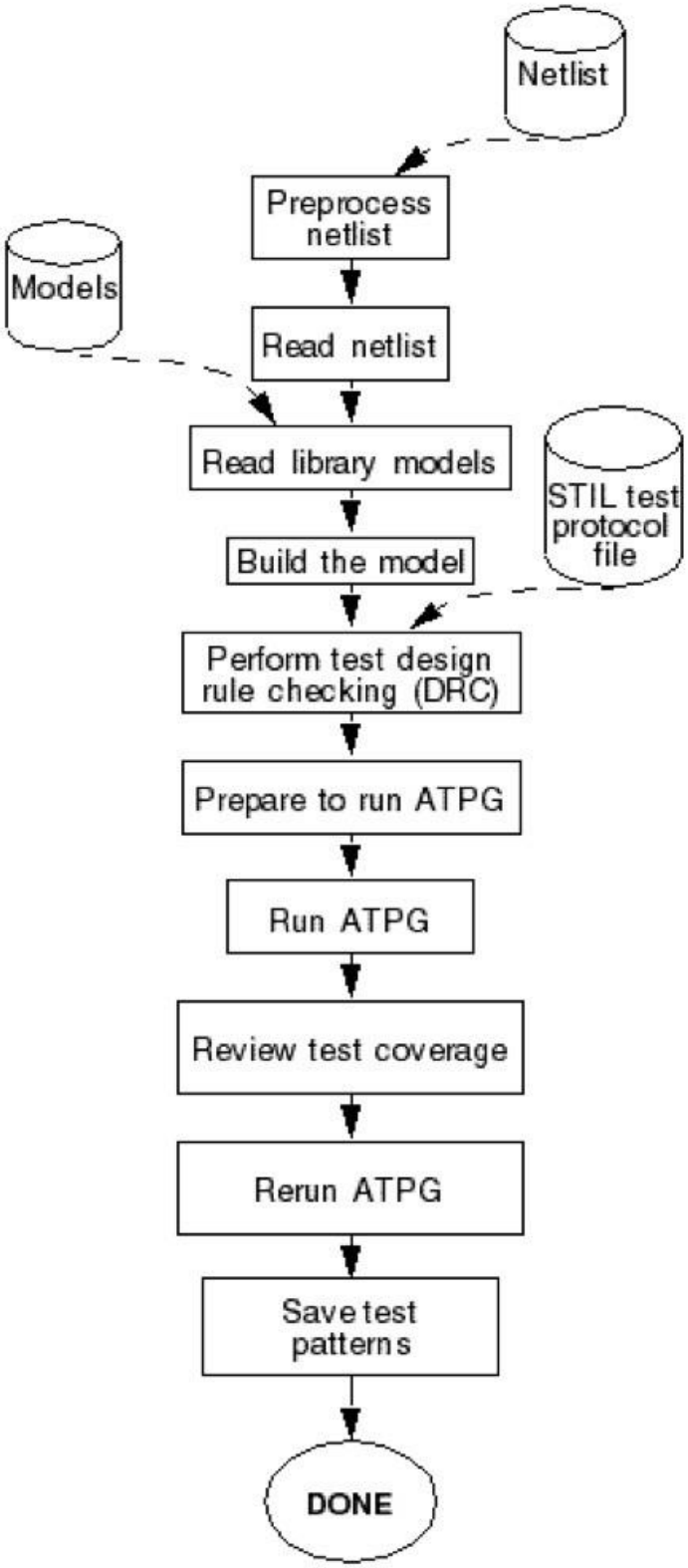
Basic ATPG Design Flow:

The basic ATPG flow applies to relatively straightforward designs. It consists of the following steps:

1) Preprocess your netlist to meet the requirements of TetraMAX (If necessary)
2) Read in the netlist
3) Read in the library models
4) Build the ATPG design model
5) Read in the STIL test protocol file, generated by DFT Compiler or another source
6) To prepare the design for ATGP, set up the fault list, analyze buses for contention, and the ATGP options
7) Run automatic test pattern generation
8) Review the test coverage and rerun ATGP if necessary
9) Rerun ATGP
10) Save the test patterns and fault list

Note: Figure 2 shows the Basic ATGP Design Flow.

Figure 2: Basic ATGP Design Flow


Netlist
Preprocess netlist
Models
Read netlist
Read library models
STIL test protocol file
Build the model
Perform test design rule checking (DRC)
Prepare to run ATPG
Run ATPG
Review test coverage
Rerun ATPG
Save test patterns
DONE

Preparing the Netlist:

Before reading in the netlist or the library models, you should check to see if any of the modules in your netlist have the same names as the library models. In the case of duplicate module definitions, the last one encountered is the one used. Therefore, if you read in your netlist and then read in library models, the modules in your netlist are overwritten by any library models that having the same names.

If you need to start over or you want to clear the in-memory design and switch to a new design, use the following commands:

1) BUILD-T> read_netlist –delete
2) BUILD-T> read_netlist filename –delete

Reading the Netlist:

BUILD-T> read_netlist netlist_directory_name/*.v

Reading Library Models:

BUILD-T> read_netlist library_directory_name/*.v

Specifying a Command File:

You can specify a command file containing TetraMAX commands in a number of ways:

% tmax command_file [other_args] … %
tmax [other_args] … < command_file

```
% tmax [other_args]
    BUILD-T> source command_file
```

What to turn in: Please take a snapshot of the TetraMAX GUI and upload this for the lab credit.