ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION

2017 Pacific Southwest Section
Meeting: Tempe, Arizona Apr 20 Paper ID #20694

Empirical Learning of Digital Systems Testing and Testable Design Using Industry-Verified Electronics Design Automation Tools in Classroom

Dr. Reza Raeisi, California State University, Fresno

DR REZA RAEISI a Professor of Electrical and Computer Engineering Department at California State University, Fresno. He is also Chair of he ECE department. His research interests include integrated circuits, embedded systems, and VLSI-CAD technology. He serves as Pacific Southwest regional director of American Society of Engineering Education. He is an entrepreneur with over 20 years of domestic an international experience and professional skills in both industry and academia. Dr. Raeisi may be reached at rraeisi@csufresno.edu

Mr. Vidya sagar reddy Gopala P.E., California State University, Fresno

Vidya sagar reddy Gopala received the B.E. in Electronics and Communication from Visvesvaraya Technological University of India (2015). He is currently perusing M.S. in Computer Engineering at California State University,Fresno. He works as teaching and Graduate Assistant in the Department of Electrical and Computer Engineering at California State University, Fresno. His research interests include NOC, VLSI design, system testing, testable design and verification.

Empirical Learning of Digital Systems Testing and Testable Design Using Industry-Verified Electronics Design Automation Tools in Classroom

Dr. Reza Raeisi, Vidya Sagar Reddy Gopala, Electrical and Computer Engineering Department, California State University, Fresno, CA 93740, USA.

Abstract

The intention of this paper is to introduce and share classroom empirical knowledge on Synopsys TetraMax, an Automatic Test Pattern Generation (ATPG) for design verification and testing of digital logic circuits. TetraMax is an ATPG tool used by the largest innovative silicon companies globally to generate test vectors automatically for design verification of Application-Specific Integrated Circuits (ASIC). TetraMax is the leading tool for generating minimum test patterns possible that covers maximum test coverage for a wide range of designs. The unparalleled ease-of-use and high performance provided by TetraMax allows designers to create efficient, compact test for even the most complex designs in minimal time. Normally, Computer Engineering curriculum does not include courses beyond their fundamental digital logic courses. We have developed a course "Digital Systems Testing and Testable Design"; for students of Computer Engineering who want to be specialized in the design, verification and testing side of VLSI circuits. We will share our knowledge gained through building and configuring Synopsys tools and their application for the design, verification and testing of VLSI circuits in the course. The career field of VLSI verification and test offers excellent opportunities for fresh engineering graduates. Training students to apply theoretical concepts with verified industry tools allows them to gain a deeper level of knowledge of VLSI design, verification and testing. Therefore, enabling them to become career ready upon graduation. This pedagogical experience of course covering the fundamentals of VLSI test process and automatic test equipment (ATE), test economics, faults, fault modeling and fault simulation in conjunction with the empirical learning of Synopsys tools for ATPG will be discussed in the body of the paper along with a results and analysis of a basic example.

I. Introduction

Today's revolutionized modern Field Programmable Gate Array (FPGA), Hardware Descriptive Language (HDL) along with Electronic Design Automation (EDA) tools has made it possible to design, verify, test and implement digital logic circuits in a classroom environment. Design Verification and Testing of design logic circuits or integrated circuits (IC) are done at various levels to ensure the functionality of the design doesn't get to modify or changed while the design flow. Synopsys VCS functional verification, Design Compiler synthesis and TetraMax Automatic Test Pattern Generation solution (ATPG) tools have made it possible for students to boost their digital circuit design skills in a classroom environment. Testing is a huge expense for the manufacturers in the field of integrated circuit design and process. An example of the

Automatic Test Equipment (ATE) purchase price alone is $4.272M [2]. Becoming familiar with VLSI testing and experiencing industry-verified EDA tools will enable graduates to be a career ready for a high-paid VLSI test engineering position.

Synopsys VCS-Verilog Compiler Simulator, Design compiler, and TetraMax have been integrated with a newly developed course Digital Systems Testing and Testable Design in the Electrical and Computer Engineering Department. This is a dual level course to be taken for both undergraduate and graduate students. The course description includes an introduction to VLSI testing, VLSI test process and automatic test equipment, test economic, faults and fault modeling, logic and fault simulation, testability measures, delay test, design for testability, built-in self-test, boundary scan, and JTAG.

Students will learn to use the Synopsys VCS tool to design and simulate some fundamental digital circuits. They experience representing the digital logic circuit in Verilog domain with using VCS to verify the functionality of their design by viewing, analyzing and debugging the generated waveforms. VCS will compile their Verilog design source file into an executable file for simulation and verification of their design while the Design Compiler will generate extra files for analyses and elaborating the design for synthesis. These files then can be inserted into TetraMax for Automatic Test Pattern Generation and fault modeling of the design.

Figure 1 shows a simple FPGA Digital Design flow as students will experience in Digital Systems Testing and Testable Design course. Students are expected to design and develop digital circuits or a system in Verilog HDL language, and a corresponding test bench to test the same. Synopsys VCS tool is used to simulate the design and verify the pre-synthesis design functionality. Then the functionally verified design is fed to the Design compiler tool, which analyzes and elaborates the design to produce three major files: Gate-level netlist file, Standard delay format, and design constraints file. The synthesized gate-level netlist file is again simulated using Synopsys VCS in order to verify the post-synthesis functionality. Design Vision, a GUI of Design compiler can also be used for easy practice. Along with the Gate-level netlist file Schematic view and Symbol view is also generated. TetraMax is then fed with the netlist file and Critical paths generated by Synopsys Primetime along with the Standard library files. As shown below in figure 1, TetraMax has three modes or phases in generating test patterns BUILD, DRC and TEST modes. Thus generated test patterns are again fed to ATE or VCS to verify the functionality of the design. The highlighted path in the flowchart highlights the testing process.

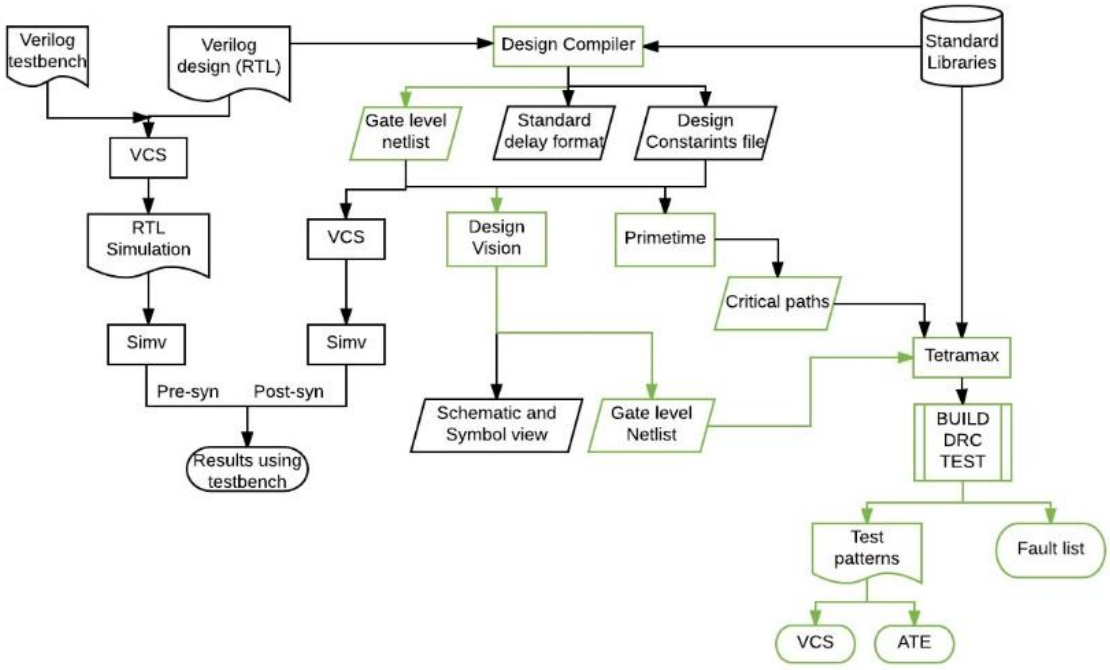

Figure.1. Overview of ASIC design flow highlighting the testing process using TetraMax

Testing is a process to identify the faults in the digital logic circuits. There are few strategic approaches for testing digital logic circuits in case of failure. A failure of a digital logic circuit is when a defect is present in the circuit. Faults can occur in the design process, device defects, and manufacturing process. Faults seen at Design process are cheaper as they are rectified sooner, faults seen after production of IC are more expensive as the process of manufacturing and packaging is expensive. Faults are classified broadly into two types: Dynamic faults which are related to timing delay and static faults which are related to shorts, open, wiring or close running wire. In this course first, the students will familiarize with the basic concepts of modeling faults and testing of digital circuits. Testing is the process of generating a sequence of test vectors as input values for a circuit and observing output value to identify an error. Simply, testing of a circuit consists of the sequence of one or more test steps, where each test step corresponds to a sequence of one or more test vectors. Secondly, students will familiarize with the dominant industry tools in order to gain hands-on knowledge. We discuss a leading tool called TetraMax, an ATPG tool by Synopsys. The course focus is on static faults that are generated during design and fabrication process using relevant ATPG test algorithms.

II. Fault models

Fault models are used to analyze various faults. Simplest and popular ones are stuck at faults. Stuck at fault model(SAF) is still a prevalent model in use today [3]. Defects are seen at random locations on a wafer. Therefore, it is assumed that SAF can occur anywhere on the chip. Any pin on a gate can be the site of a potential SA0 or SA1. If a wire is stuck at ‘0' where it is expected to

be logic ‘1' is called as stuck at ‘0' (SA0) and if a wire is stuck at ‘1' where it is expected to be logic ‘0' is called as stuck at ‘1'(SA1). The complexity of a chip prevents considering stuck at faults at the transistor level. Many other fault models exist, but SAF is commonly considered for its simplicity. Certain vectors are generated to test and detect such faults: these vectors are called at Test patterns. Pattern generation of the combinational circuit is defined as, for given set of faults(F) and set of test vectors (T) identify the smallest possible subset of test vectors (V) which covers either all the faults in F or says a predetermined fraction of faults. Later, these techniques are extended for learning of sequential circuit testing. For given test vectors, by simulating the circuit with faults, identify all faults covered by the test vectors is called as Fault Simulation. Given a fault, identification of all such test vectors that cover the fault is known as test generation or test pattern generation.

III. D Algorithm

There are many models and algorithms for detecting stuck-at faults. However, the classic model which is required for understanding the advanced models and for detecting any stuck at faults, is known as D-algorithm. D algorithm detects faults (SAF) anywhere in the combinational circuit without the requirement of any internal probing. Though the first paper on it was published in 1966, even now D algorithm is essential to understand most of the Automatic test pattern generation (ATPG) tools. The letter ‘D' in D algorithm stands for the discrepancy. D algorithm detects one stuck at a time, and it makes a discrepancy between fault-free and fault evaluation. The D algorithm can be actually divided into four steps of action. Firstly, specify a stuck-at fault at a particular node or pin of a combinational circuit. Then, drive the fault to opposite value and propagate the fault to the primary output, which is called as the sensitization of the path. Then, Justification is required. When a fault is propagated to the primary output, a reverse approach is followed to force the inputs in such a way to generate that fault at the output. The consistency check for the path is done and then the pattern that is required to identify the fault is back tracked. Thus, the input pattern justified to obtain the fault is the test pattern generated for that particular fault. Figure 2 explains the D-algorithm in the simplest way possible. The test pattern can be defined as a sequence of one or more vectors that applies stimulus and checks for an expected response to detecting a target fault. Students will familiarize with the techniques of testing sequential circuits. In a sequential circuit test, regular flip-flops are replaced with testable flip-flops which known as a scan replacement. Testable flip-flops are known to have two modes of operation. Firstly, normal mode- while the regular operation is seen as it was designed. Then, test mode- when the IC chip is undergoing the manufacturing tests. When undergoing manufacturing test, all the flip-flops are connected like a shift register. This configuration is called as a scan chain. Most of the time, multiplexed flip-flops are used as scan-replacement for normal flip-flops. All the above-mentioned algorithms can be used to generate test patterns and validate it on a small circuit manually. However, circuits or the designs have millions of logics in a single chip, thus we use Electronic Design Automation (EDA) tools to generate and validate such patterns. Students will apply the theoretical testing concepts learned in the course to experience the use of a leading tool called TetraMax, which is an ATPG tool developed by Synopsys for testing of VLSI circuits.

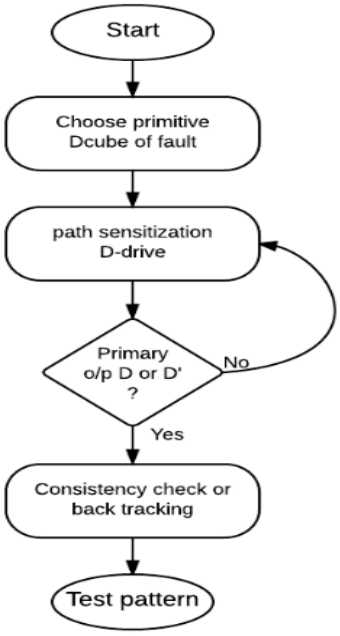

Figure.2. Flow chart representing D-algorithm

IV. TetraMax

Synopsys TetraMax is an ATPG tool used by the largest innovative silicon companies globally to generate test patterns automatically for the developed VLSI circuits. It is the only test solution for a wide range of methodologies. TetraMax tool is known for generating minimum test patterns possible that covers maximum test coverage for a wide range of designs. The unparalleled ease-of-use and high performance provided by TetraMax allows designers to quickly create an efficient, compact test for even the most complex VLSI designs [3]. It has the highest compatibility with the design, synthesis and other tools of Synopsys [5]. It generates test patterns for even most complex and largest design under test. It quickly isolates the faults and enables faster yield. TetraMax can read design in Verilog, VHDL, and EDIF formats and can write test patterns in WGL, STIL, Verilog, VHDL, Fujitsu TDL, TI TDL91, and Toshiba TSTL2 format. Based on the user situation and requirement, TetraMax can be run in three different modes: Basic-scan ATPG which is an efficient combinational only mode for full-scan designs, Fast-sequential ATPG for partial scan design but limited support only and a Full-sequential ATPG for full test coverage in partial scan mode. It provides a Graphical Schematic Viewer (GSV), a graphical interface for analysis of design rule violations. To validate the test pattern, it provides the links to VHDL or the Verilog simulators.

Figure 3 shows TetraMax integrated test flow. TetraMax ATPG automatically generates high-quality manufacturing test patterns. With TetraMax ATPG, designers can generate test patterns without compromising on high-performance design techniques [8]. TetraMax ATPG's design rule checker (DRC) identifies chip-level test issues. The DRC enables the designer to create "test-friendly" RTL that can then be easily synthesized in the DFT synthesis environment.

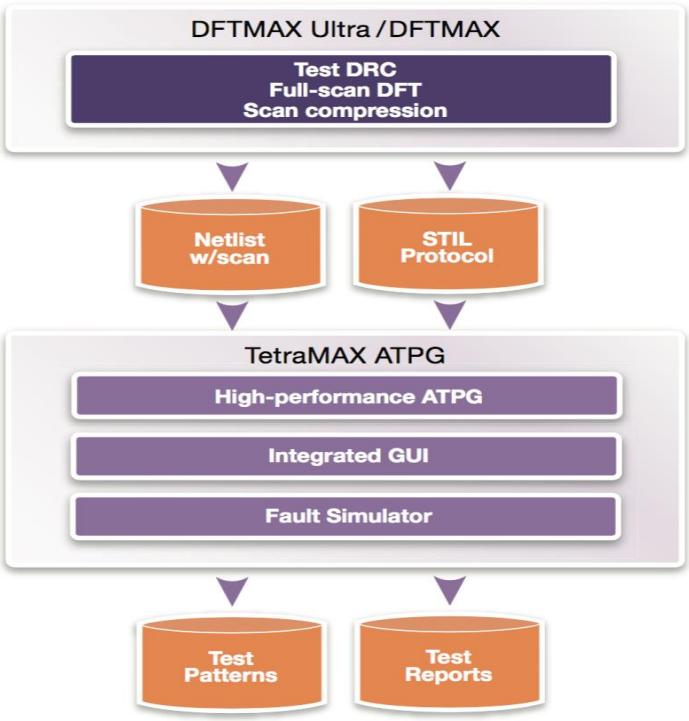

Figure.3. Integrated Test Flow using TetraMax ATPG [8]

The primary function of Test DRC is to provide feedback on the testability of the design during the pre-synthesis stage. The module designer invokes the Test DRC feature on the RTL module prior to synthesis to verify a comprehensive set of pre-scan DRC rules. The designer has the option to x the violations in the RTL source code based on the feedback. This enables the designer to account for RTL testability early in the design process [1]. The feedback on violations can be viewed through a browser in the Design Vision graphical user interface (Figure 1).

TetraMax provides three different modes of test pattern generation. Firstly, Basic-Scan ATPG, which operates as a combinational only full-scan. Highest coverage can be obtained if sequential elements are scan elements. Fast sequential ATPG provides limited support for partial scan designs. We can enable Fast-sequential mode and specify its effort level by using `capture_cycles` in `set_atpg` command. Finally, Full-sequential ATPG is similar to fast sequential ATPG and it supports multiple capture cycles between scan loads and scan unloads thus test coverage in partial scan designs increases in this mode. Full-Sequential mode by using the `-full_seq_atpg` option of the `set_atpg` command.

TetraMax supports test pattern generation for five types of fault models. Firstly, stuck-at fault model which is a common or the standard model for test pattern generation. In this model, the circuit behaves as if the node is stuck at either 0 or 1. Then, the fault is propagated towards the primary output and scan cells such that they can be observed at output or scan chains

respectively. The next model is the Transition delay fault model which are used to generate test patterns to detect single-node slow-to-rise and slow-to-fall faults. Logical transition upon completion of a scan load operation and uses a capture clock procedure are launched by TetraMax to observe the transition results. The third model is path delay fault model; it tests to exercise the critical path at-speed to find out whether the path is too slow as the result of manufacturing defects or variation. The last model is the IDDQ fault model; it assumes excessive current drain due to a circuit defect like internal short of ground or power supply.

V. TetraMax working

In Digital Systems Testing and Testable Design, students will gain experience on the leading EDA tool TetraMax. The flow chart or the process flow shown in figure 4 explains clearly the various stages of TetraMax in automatic test pattern generation (ATPG) of digital circuits. It accepts synthesized netlist along with standard technology library and starts the process of ATPG.

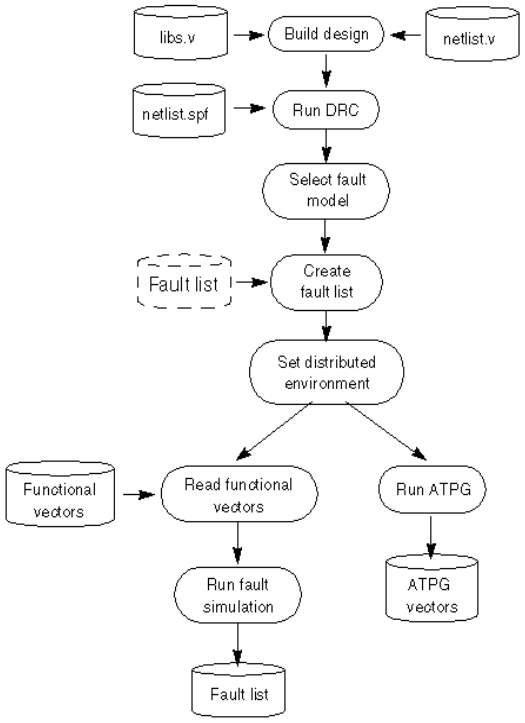

Figure.4. Distributed process flow [6]

i. BUILD mode

As soon as the TetraMax is invoked a startup file called '`tmaxtcl.rc`' is executed automatically. As shown in the flowchart figure 4, TetraMax reads netlist file and the library files or modules. TetraMax supports various file formats like Verilog, EDIF, and VHDL. The netlist can be either hierarchical or flat. The netlist can be a single file or multiple files. In the

case of repetition of netlist, TetraMax will keep the latest module. Building the ATPG design model takes those parts of the design that are to be part of the ATPG process, removes the hierarchy, and puts them into an in-memory image that TetraMax can use [7]. TetraMax creates individual modules for all the gates when accessing gate-level netlist. Building a netlist is nothing but extracting a component that contributes in generating test pattern and mapping them in an in-memory image.

ii. DRC mode

Once the BUILD mode of the TetraMax is successfully executed then the DRC phase begins. Design rule check performs rule checks namely: S rules (Scan chain rules), Z rules (internal tristate busses and bi-directional pins), C rules (Clocks or capture), X rules (combinational feedback loops) and V Rules (Vector statements in the SPF). Rules above mentioned are responsible for following aspects of the design. The user should specify details about clock ports and primary inputs in STIL test protocol. STIL verifies the scan chain outputs and inputs are logically connected and check all the asynchronous set/reset pins and clock pins are in connection with scan through primary ports, verifies if any internal multiple-driver nets can be in conflict and whether the clocks/sets/resets are off when you switch from normal mode to scan shift mode and again when you switch back to normal mode. Using TetraMax command interface or GUI the DRC performed. On successful checking of the Design rules, the mode of TetraMax changes to TEST which signifies that the design has no violation. If in the case of any violation, they can be analyzed using GSV (Graphical Schematic Viewer) and all the errors to be fixed using GSV before proceeding to the further steps. GSV icon in the display is activated only if the netlist BUILD properly. GSV POPS up between transcript window and command tool bar upon clicking schematic view. GSV has its own tool bar using that toolbar user can analyze the design, debug and apply the test patterns to view responses and verify them, trace forward and backward paths of a gate.

iii. TEST mode

Before performing ATPG, the user introduces required fault into the design through TetraMax by choosing a required model on the Run ATPG window. Before generating test patterns fault list must be initialized. Various settings to be done at this stage of pattern generation like setting abort limit, setting coverage percentage, adding fault’s source, selecting fault modeling and type of ATPG (Basic scan / Fast sequential /Full sequential). Therefore, after setting the required configuration the user can run the ATPG and observe the patterns in the transcript window.

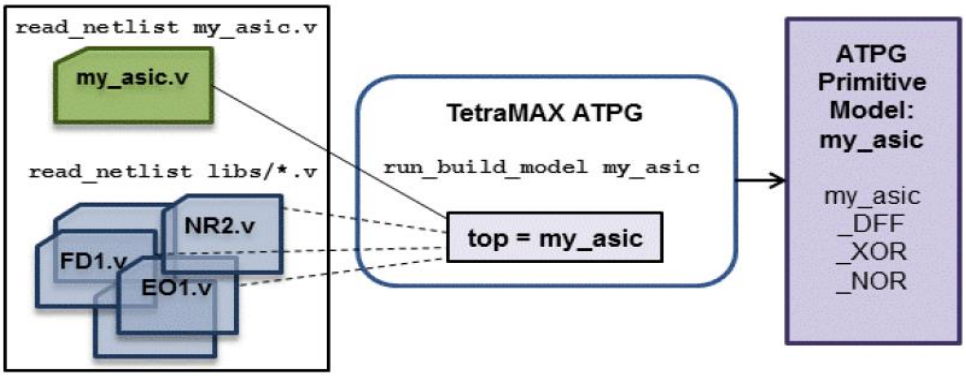

Figure.5. Building ATPG model [7]

Figure 5 above shows the process of building ATPG model. As already mentioned user can choose from available three ATPG engines. The usage of these three engines differs with the type of design used for ATPG. Firstly, Basic scan; ATPG uses basic scan engines for fast and high performance (based on user requirement) and gives good coverage for Full-scan designs. Fast-sequential scan gives higher coverage for a design with memories and for near full scan designs. Lastly, a Full-sequential scan is good for designs with DFT violations. It is the most powerful engine that supports more complex designs. After desired settings are configured ATPG may be performed either in regular `run` mode or by `auto` mode.

iv. Importance of auto mode

In auto mode, ATPG adds faults in the design if not present. ATPG Untestable (AU) faults are detected upon "quick" run. ATPG doesn't store any patterns to detect AU faults at this point of time; the decision is made randomly [7]. After completion of process patterns are stored with high merge efforts whereas in regular run mode user can select this effort. It gives a report of CPU time along with fault and test coverage. While saving patterns run one pass of static compression.

v. Reviewing results

Along with the test patterns, faults, fault coverage, CPU timing, the number of patterns generated are displayed in the script or window. Reviewing results involves in going through various types of faults that are possible in the design. During ATPG, TetraMax classifies faults into 11 fault classes and groups them into five major fault groups as listed in the table 1. Table 1 classifies the 11 faults into five major fault groups.

DETECTED [DT]	POSSIBLY DETECTED [PT]	UNDETECTABLE [UD]	ATPG UNTESTABLE [AU]	NOT DETECTED [ND]
Detected by Simulation [DS]	ATPG untestable possibly detected [AP]	Undetectable unused [UU]	ATPG untestable, not detected [AN]	Not controlled [NC]
Detected by Implication[DI]	Not analyzed, possibly detected [NP]	Undetectably tied [UT]		Not observed [NO]
		Undetectably blocked [UB]		
		undetectable redundant [UR]		

Table.1. Fault classes and their groups

After desired configuration is set up ATPG may be performed either in regular `run` mode or by `auto` mode as discussed above. Transcript window reflects every change we induce in the design process.
In the process of ATPG TetraMax provides two types of pattern compression. Dynamic compression is processed during execution of ATPG phase using `auto` mode. This is performed using high merge effort during ATPG. And Static compression is performed after the patterns are generated; this is done using command `pattern_compression`.

VI. Results and analysis

Now, the results and analysis of results obtained using TetraMax tool are discussed. TetraMax provides an excellent GUI called as Graphical Schematic Viewer(GSV), figure 6 is the screen capture of the Schematic for 3-8 decoder circuit in primitive mode. The Schematic can be displayed in both primitive and design vie. Here in figure 6, we have shown a primitive view of the GSV.

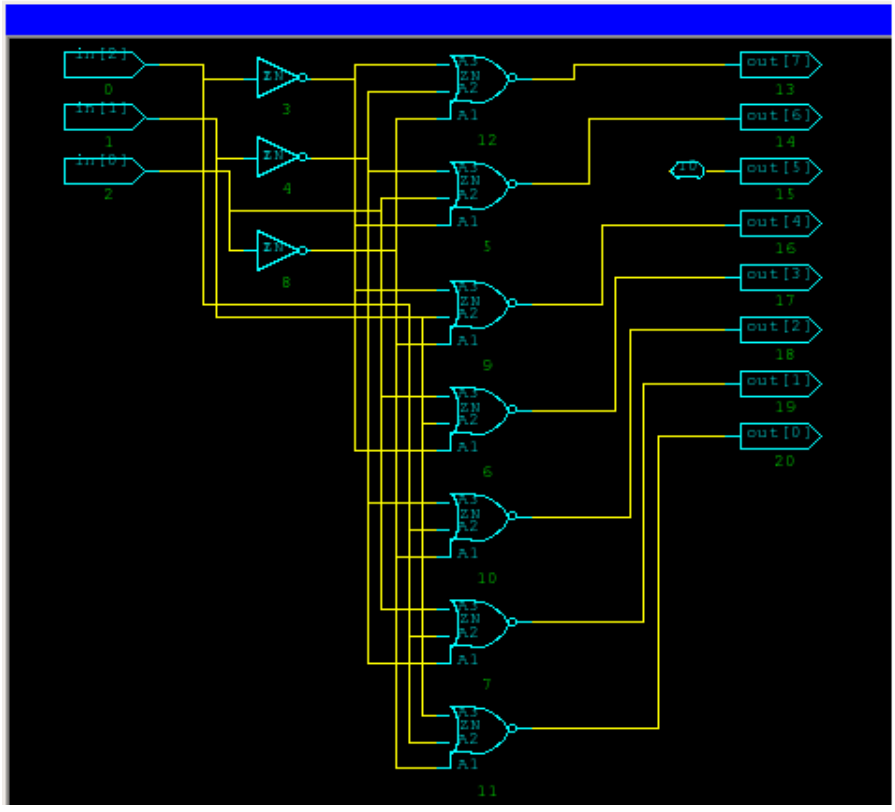

Figure.6. GSV window (3-8 decoder schematic view)

After the patterns are generated and reviewed, they are written as a file for any further use. TetraMax allows users to save output files in various formats according to the requirement as

discussed earlier. Usually, patterns are saved in Verilog-single files. In the Verilog file where the patterns are written, it displays pattern number, pattern and the time at which it was used for simulation.

Running results of the TetraMax ATPG on a 3 to 8 decoder is shown in figure 7 where students will observe a transcript window projecting important test report. As seen in the test report; fault class, total number of faults, test coverage, CPU time and pattern summary are clearly depicted in the included capture of transcript window. For the 3-8 decoder netlist fed as an example, we obtain 14 internal test patterns and totally 90 faults were detected with a test coverage of 100% as shown in the screen shot below.

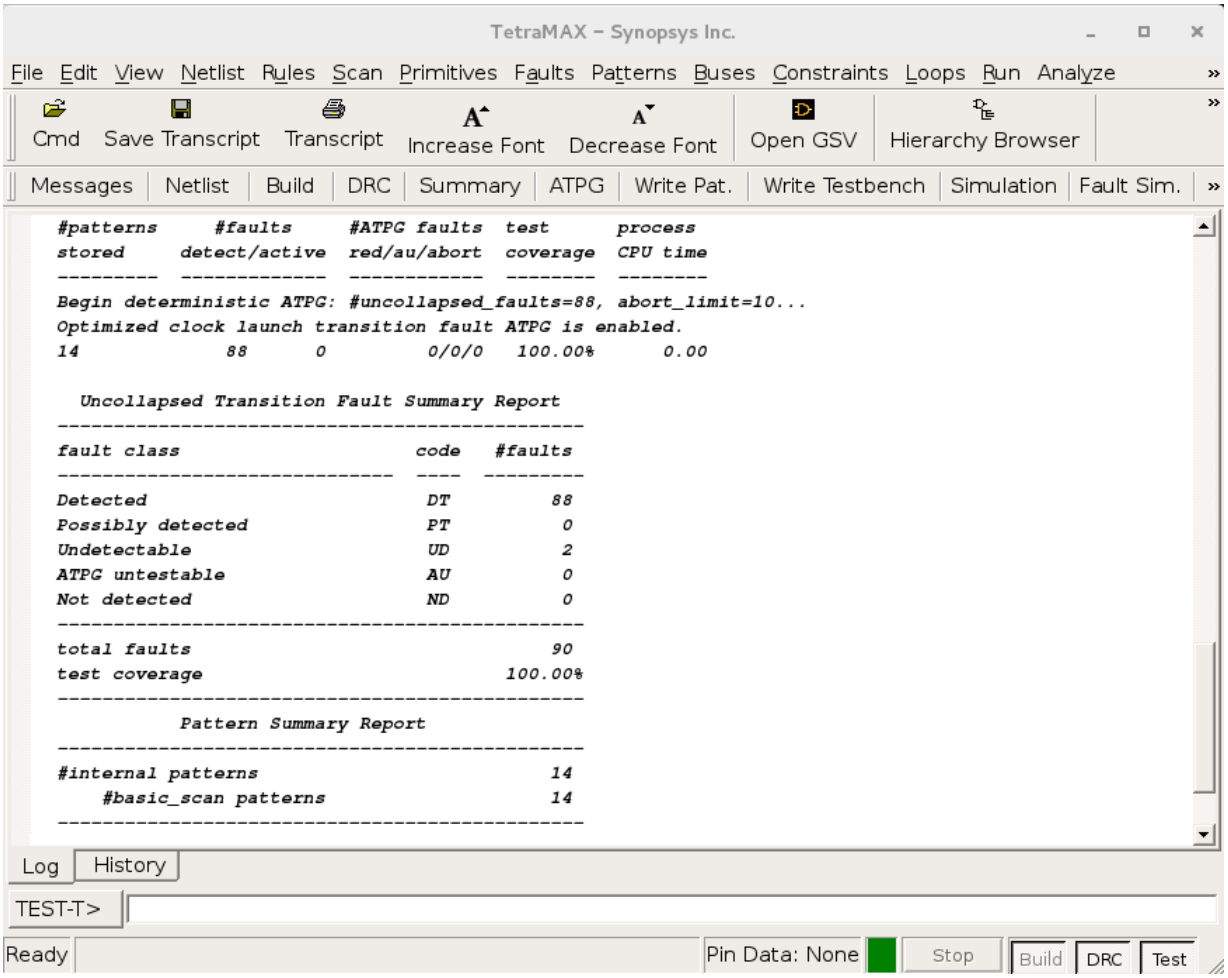

Figure.7. Transcript window displaying report

VII. Conclusion

Objective of this paper was to share the basic experience learned in Digital Systems Testing and Testable Design course using Synopsys EDA tools for design verification and testing of digital logic circuits. Application and characteristics of TetraMax tool have been highlighted. We

examined the benefits of hands-on experience by transferring the advanced VLSI testing concepts using EDA tools to improve students learning. Finally, bridging the gap between educational concepts and industrial application and enabling students career ready upon graduation.

VIII. Bibliography

[1] https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/test-automation/tetramax-atpg.html

[2] http://www.engr.uconn.edu/~tehrani/teaching/test/

[3] Abraham, J. A., & Fuchs, W. K. (1986). Fault and error models for VLSI. *Proceedings of the IEEE*, *74*(5), 639-654.

[4] https://www.ece.ncsu.edu/erl/faculty/paulf.html

[5] https://www.synopsys.com/content/dam/synopsys/implementation&signoff/datasheets/tetramax-ds.pdf

[6] http://www.lirmm.fr/~bosio/tmax_olh/Content/tmax_ug/13.fault_sim/fault_sim_des_flow.htm

[7] Synopsys Inc., "TetraMax ATPG user guide, Version E-2010.12, December 2010.

[8] https://www.synopsys.com/content/dam/synopsys/implementation&signoff/datasheets/tetramax-ds.pdf