Radiation Environment Emulation for VLSI Designs: A Low Cost Platform based on Xilinx FPGA's

J. Nápoles, H.Guzmán, M. Aguirre[1], ,J.N. Tombs, F. Muñoz, V. Baena, A. Torralba, L.G.Franquelo

Escuela Superior de Ingenieros Universidad de Sevilla. Camino de los Descubrimientos s/n 41092 Sevilla (SPAIN){aguirre,jon,fmunoz,baena,torralba}@gte.esi.us.es

***Abstract*—As technology shrinks, critical industral applications have to be designed with special care. VLSI circuits become more sensitive to ambient radiation: it affects to the internal structures, combinational or sequential elements. The effects, known as Single Event Effects (SEEs), are modeled as spontaneous logical changes in a running netlist. They can be mitigated at netlist design level by means of inserting massive redundancy logic in the IC memory elements, as well as designing robust deadlock-free state machines. Current techniques for the analysis and verification of the protection logic for VLSI are inefficient and expensive, lacking either speed or analysis. This paper presents the FT-UNSHADES system. This system is a low cost emulator focused on bit-flip insertion and SEE analysis at hardware speed, based on a Xilinx Virtex-II. Radiation tests are emulated in a highly controlled process, using a non-intrusive method. As a result the system can insert and analyse at least 80K faults per hour in a system with 2 million test vectors.**

***Index Terms*— Fault Tolerant, FPGA, Radiation, Partial Reconfiguration, Reliability, VLSI design.**

I. Introduction

VLSI designs for critical industry applications like automotive, health support, aeronautics, and others have to consider new effects when they are implemented using nanometric technologies, like 90nm, 65nm or less. Ambient neutrons have enough energy to change the logic state of a logic gate or the state of a flip-flop . Although the probability for the phenomena is extremely low, the above applications are managed using large electronic silicon areas and widespread implemented into many applications.

The impact of radiation particles can force transient changes in electronic structures that can modify their electrical states. One consequence is that internal flip-flops can spontaneously change their state (bit-flip). These errors are known as Single Event Upsets (SEUs) and they do not represent any physical damage to the circuit but produce an abnormal functioning (they a re also called *soft errors*). The typical techniques for improving circuit reliability are based on redundancy insertion. For example, Triple Modular Redundancy (TMR), triplicates every flip-flop and inserts a majority voter to resolve the actual state of the Flip-flop. Another example is Error Detection and Correction (EDAC) subcircuit for memories. VLSI designs consequently grow in size with a factor of 3.2, and power consumption. Costs are increased design time and Non Recurring Engineering.

One solution reduce the impact of redundancy insertion is to search the hierarchical modules of the circuit that are critical for the global system and insert protections only to those sections. This technique is known as selective protection. A deep analysis of the circuit is then needed.

Several problems have been addressed in the introduction of redundancies [2]. The main problem comes from the essence of the synthesis tool, where the netlist is optimized for redundancy elimination, second, because the TMR is inserted manually without any restriction and verification. The tools for this latter purpose are too slow.

In space applications, tests are made by means of reproducing the out space environment using radiation chambers, testing the circuit "in system". Radiation effects are measured using a fully functional hardware system and faults are detected by I/O comparison cycle-by-cycle between the tested system and a non ex-posed twin system. These techniques are expensive and non affordable for industry applications. Radiation effects analysis has been traditionally performed using simulators that work using a model of the effect, called bit-flip. The radiation environments are reproduced using fault-injection techniques. They are slow and need many trials to detect a weak FF [3][4]. Prototyping is a technique for netlist validation at design level. Some simulation techniques have been proposed and, but in practice they are very slow.

Hardware approaches are attractive due to its significant speed-up of the injection rate. The

A new prototyping platform focused on the detection and analysis of fault tolerance in designs is being developed under the name FT-UNSHADES. FT-UNSHADES (Fault Tolerant – UNiversity of Sevilla HArdware DEbugging System) is a hardware/software platform that takes advantage of the configuration circuitry present in all of Xilinx Virtex technology. It has been specialized on producing functional testing of a design, using a test vector database contained in on board memory banks, but with a careful control of the clock signal. Time is represented as an ordered set of stimuli that are injected into the FPGA, SEUs are inserted using a read-modify-write strategy of the

configuration memory of the FPGA. This scheme is an application of the previous experience UNSHADES-1 and UNSHADES-2 [6-7].

This paper is organized as follows: first, we introduce to the SEU measurement problem. Next we describe the internals of the FT-UNSHADES solution. The fourth section describes the tools for producing testers and fifth section shows results of the FT-UNSHADES behavior.

II. SEU AS FAULT INSERTION PROBLEM

The fault insertion problem is exposed in this section in order to achieve a solution for the implementation strategy. It is accepted that when energetic particles hit to sensitive areas of a digital circuit, it produces soft errors, that are equivalent to one or several bit-flips in the set of FFs (flip-flops), changing the currently stored value at the same clock cycle of the impact. The state is corrupted and can be propagated to primary outputs, if the sequence of inputs drives the circuit to an unexpected behavior of its I/Os (primary input/outputs), this fault can cause a damage to the system. Another possibility is that the fault remains latent in the circuit without any effect to the system. The fault activity should be detected if the complete circuit state is compared with the theoretical state, at the end of the test cycle.

Reliability against radiation of a circuit depends on circuit architecture and the functionality which the circuit is designed for. Designers can protect every FF using redundancy techniques. Circuit protections increase the area and power consumption. It is desirable to select the critical FFs as candidates for being protected or guarantee that the complete circuit is reliable. In both cases the fault injection study has to be performed.

Several techniques have been proposed in literature for design dependability evaluation. Software based techniques [4] are based on HDL simulators. Tests are performed recording the state at every clock cycle and making comparisons with the state in gold simulations contained in a previously recorded database. Simulations are too slow when compared against the huge number of test cycles that are needed to produce a large enough test and to extract conclusions.

Opposed to software approaches, tests performed by hardware emulators (eg. using FPGAs) are an attractive solution that allows speeding-up the tests. The main problem is that additional circuits have to be inserted during synthesis for hardware access to the FF contents. Additionally a poor analysis can be performed because the observability is oriented to external pin observation and little internal information is obtained.

The present approach is based on Xilinx Virtex technology. It has two unique features that can be exploited intensively for solving the problem. The first feature is called the *capture and readback* mechanism, described in [xx], and provides a non intrusive way to observe the entire internal state without any design modification and overhead in time or resources. The second is that the configuration can be partially read and written. Using an adequate approach, it is possible to force the desired values into selected FFs whilst the rest of the system state remains constant.

A. FPGAs emulating a radiation test.

Our study is made on a post-synthesis description of the design, being valid because an incremental synthesis tool is used for the design for test to. Other radiation upset effect such as latch-up is not covered as they must be protected by means of technological solutions and are out of the scope of this paper. The study is concentrated over the flip-flops, so the results can be referred to the VLSI circuit itself.

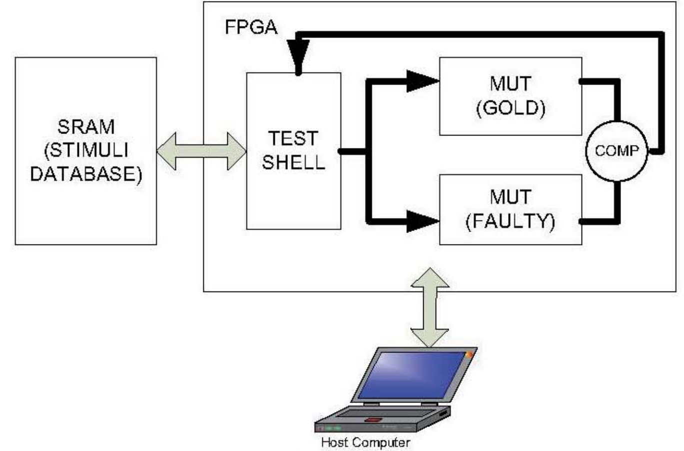

Figure 1 Testing approach

Figure 1 shows the proposed testing scheme. The module under test (MUT) is a post synthesis description of the design. Two copies of the MUT are forms the test system. One is dedicated to produce the right outputs (GOLD) and other will be the candidate to receive the SEUs (FAULTY). If s is a feasible (a reachable configuration of the design state with a set of vectors) state of all the FF for a test, that is at a particular moment represented by the set of stimuli T, and G(s,T) is the set of responses theoretically given by the system without any perturbation. If s_i is the state s with the FF i modified (a bit-flip), this modification can be produced at different moments, so if Tj is the time to inject the SEU. F(s_i, Tj) is the set of responses of the system when the perturbation is inserted.

The system is robust against that fault if F(s_i, Tj) is identical to G(s, T). Any discrepancy should lead to an abnormal behavior, and the fault is classified as damage. S_∞[F(s_i, Tj)] and S_∞[G(s, T)], are each module state at the end of every test. If fault is not damage and any discrepancy is found between both configurations, the fault is classified as latent.

The goal of the tester is to generate a fault dictionary, where every pair (fault, instant) is classified into three categories obtaining the following information:

- Sensitive FF
- Time of fault insertion
- Outputs modified
- Time of output discrepancy

This information characterizes every fault for a more detailed analysis.

B. FTUNSHADES hardware framework

The framework has been designed in order to achieve a fully controlled test conditions. Because of the necessary

readings and writings in the configuration memory, a Xilinx VirtexII called the System FPGA (S-FPGA) has been selected to do all the workload. Within it, two versions of the MUT will play the role of the design ex-posed to radiation and the shielded version of the circuit. Outputs are compared cycle by cycle, to detect damage faults.

One important issue inside this scheme is time. Time is controlled in terms of clock cycles applied to both, faulty and gold emulations, which is obviously represented by a counter. In the same way, time is the way to address vectors stored in SRAM memory banks. When Tj is achieved for fault injection or a fault is detected, the circuit has to be frozen in order to perform the necessary internal manipulations in the configuration of the S-FPGA. In other words, clock has to be carefully stopped at a precise clock cycle and continued when the accesses are completed. A second FPGA (called C-FPGA) acts as a high performance link between system and computer. Both FPGAs are connected using the SelectMap port and receive from the PC data and commands through parallel or USB port.

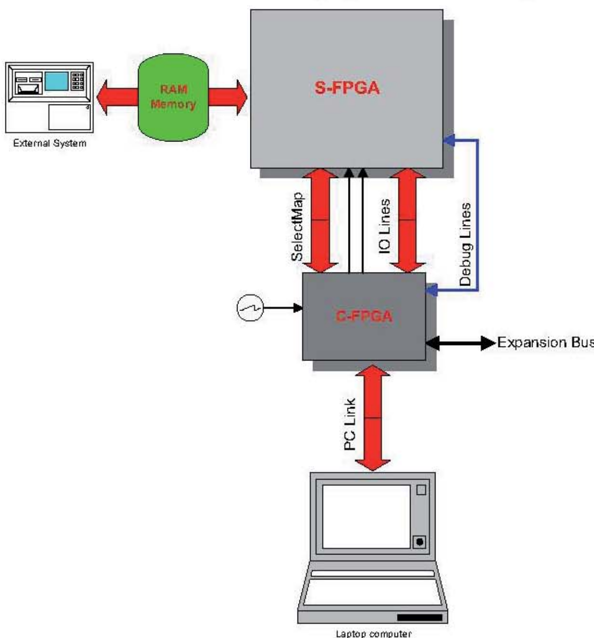

Figure 2. FT-UNSHADES Framework

This scenario needs a highly controlled data transfer scheme between a host computer and the emulated system. A software tool is dedicated to decide which FF is candidate to receive a bit-flip, and the insertion time and fault effects.

C. *Testing Software framework*

Analysis over S-FPGA is controlled by the host computer. Software tools have to provide all the necessary services to control the testing board. These services are:

- Board performance control: Select current Computer-bord link, system clock rate, detect other boards in multiboard link, etc
- Vector download: Program on board SRAM memories. The S-FPGA is configured using a dedicated vector loader bitstream and the computer transfers the vector file.
- Handle debug lines for detecting fault events [4-5].
- Generate automatically the ready-for-synthesize netlist
- Define testing strategy: Test is defined by two parameters: WHERE the fault will be inserted inside the complete netlist and WHEN it will be inserted during the test cycle. HOW these parameters are controlled by the computer is the basis of FT-UNSHADES software tools.
- Elaborate a fault dictionary.
- Provide analysis tools.

1) WHERE, WHEN and HOW

Three problems have to be solved in the SEU testing campaigns. Fault locations are called the WHERE problem. In terms of an RTL netlist the problem consists on deciding which FFs are candidates for being modified. Using the information contained in the bit allocation file (generated by Xilinx Design Flow), a relationship between the FFs logical name joined to its hierarchical path can be established with the layout location in the S-FPGA configuration. The knowledge of the hierarchical path allows the designer to concentrate the testing effort selectively into a subset of the FFs in the netlist.

The configuration frames that contain the information related to selected FF are read from the S-FPGA, modified to infer the desired state, and transferred into the S-FPGA. To avoid synchronization problems the clock signal must be frozen using a glitch free procedure. This procedure is directly controlled by the time counter, where the WHEN variable is defined. For the sake of clarity a single test runs as the following pseudo code:

```
Initial: Assert MUT Reset
Program time counter with fault insertion cycle;
Release MUT Reset
Start test cycle. Reset Circuit;
if time counter equal to fault insertion cycle
freeze MUT;
read (FF state) /*the state of the desired FF*/
insert not (FF state);
end if;
Resume clock:
if(I/O discrepancy is true)
   read counter;
   read I/O;
   classify fault as damage;
else continue;
end if;
Read STATE if latent;
Goto Start test cycle; /* Next cycle */
```

The third variable is HOW. It represents the fault insertion criteria and represents a model of the radiation reception. The most restrictive should be the bit-flip induction for all FFs and for all clock cycles; on the other hand the user can define periods and time windows over submodules in the design hierarchy for a more efficient insertion strategies.

III. Test Shell

Test shell is a set of hardware resources required for the control of the test procedure. Three blocks are:

- Time counter. This block maintains the number of clock cycles that activate the MUT.

• Clock handler. This block has to stop the MUT, re-launch the test, produces the necessary signals to indicate fault detection, activates the VIRTEX2_CAPTURE circuit and, finally handles the debug signals for single stepping analysis.

• Vector addressing. At every clock cycle an address derived from the Time counter points to the corresponding vector stored in the on-board memories. Vector can easily be compressed by means of simple techniques to increase storing capacity.

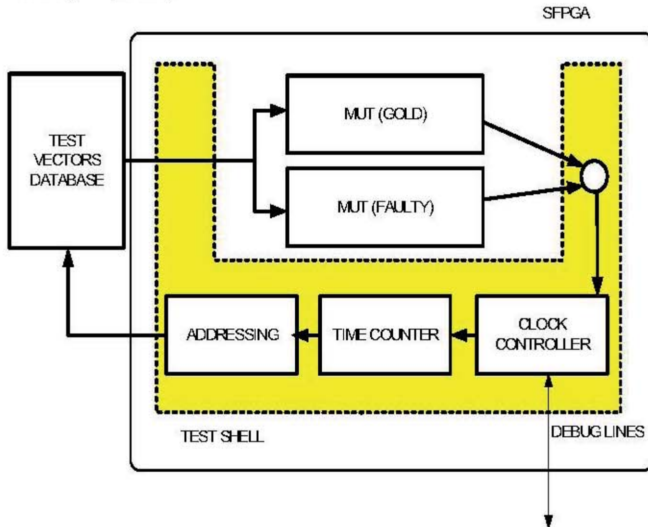

Figure 3. Test shell insertion

The test shell uses very little resources for control, equivalent to around 300 system gates, and as they work over resources that only control the clock no delay penalty over the system behaviour is introduced.

A. Preparation of the Design for Test Emulation

The most important issue for introducing such a system in a design flow is to avoid special requirements in the design. Figure 5 depicts a complete design flow. The test shell is generated automatically from a toolbox when the designer indicates the top level. Software reads the entity for detecting inputs, outputs and bidirectional signals. The test shell adopts its parameters to allow:

• Input signal sharing between Gold and Faulty copies of the MUT

• Outputs comparison

• Bidirectional signals handling

• Adds debugging control and test vector handling

• Creates the corresponding user constraint files to map pins and set clock rates etc.

The tool creates a new top level call *Design for Test Emulation* (DTE) that reproduces the schematic of figure 3. After this, the DTE is synthesised by means of the Xilinx standard design flow (ISE, Alliance,…), and the physical design flow is launched to obtain the bitstream and the *bit allocation* file, which link the FF locations in the configuration of the FPGA with the instance names given in the HDL Hardware Description Language) source. Note that the MUT description can be a postsynthesis description, making the approach fully non intrusive.

B. Bidirectiona I/Os treatment

Inputs are used for vector injection, outputs are used for making the comparison and fault detection, but, bidirectional I/Os have potential contention if the input and output are mapped to the same pin, because there are two possibilities:

- The fault is in the pin definition (eg.. in faulty it is defined as output and gold it is defined as input)
- The fault is in the pin value.

Figure 4 shows a schematic the solution adopted. In order to avoid a fail in the contention the outputs never drives memory pins and values are filtered using this circuit and storing in memory the theoretical values. Three extra pins are used to compare the bidirectional connected to two external resistors as depicted. When both signals are inputs, the comparator receives the same value, but if a fault is detected in the pin definition and/or the output value, then the comparator detects the discrepancy safely for the S-FPGA external pins.

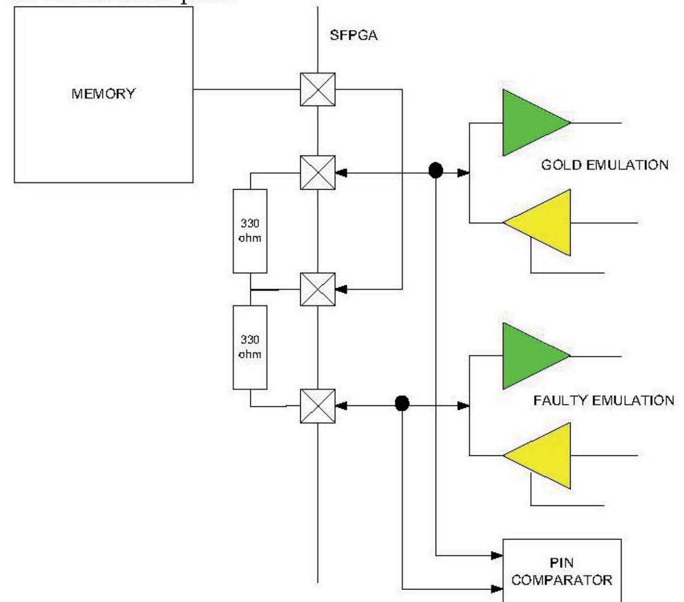

Figure 4. Test circuit for bidirecctional signals

IV. FT-UNSHADES SOFTWARE

The user interface is the host computer, where all tools run and the fault database is created. All tools are classified in four kinds of services:

• Test vector services. Creates the vector database using a test bench that produces the vector file, formatted for the on board memories. A program called *generateTVF.exe* creates a new Test Bench that generates the test vector files containing the stimuli of the DTE. Also a *MUT.pin* and *DTE.ucf* files are generated, containing input pin allocation and design constraints file respectively, at this stage that will be used in the DTE generation phase.

• Test definition services. As explained above, the DTE is automatically generated using a MUT description, *MUT.pin* and *DTE.ucf.*

• Test handling services. The bit allocation file keeps the complete instance path of all FFs. This important feature allows the selection of subsets of FFs inside the whole configuration. The command WHERE is then extended to concentrate the test over certain desired modules. Also the time (WHEN commands) can be han-dled in certain

predefined windows. All possibilities can be mixed producing refined test method known as the HOW method.

• Post testing analysis tools. The fault dictionary has enough information for re-producing the test using a step-by-step method. A detailed analysis is possible, even using a waveform viewer.

A command line environment has been created for software services. Figure shows a scheme of the complete environment.

V. Experimental Results

The actual system uses 80MHz crystal oscillator, and SFPGA is fed at 160MHz. The C-FPGA functions act as interface with PC links. The current version works over either EPP 1.9 (1.6MB/sec) or USB 2.0 high speed (1.5MB/sec). Let us assume that the DTE design works at 50MHz and uses 2 million of compressed test vectors.

A single fault injection needs at least 3 reading and writings of VirtexII frames. For a VirtexII XC2V6000, the size of a single frame is 984 bytes (this number changes if other VirtexII device is soldered on the board). A bit-flip insertion therefore requires 40 microseconds. With these conditions 20.000 faults per second could be injected, obviously it's given on the basis that the circuit is robust for the faults, because in other case system halts when a fault is detected.

The system has also been tested using huge and complex benchmark circuits as Leon2 that can be found in [9]. Using different stimuli database the final fault rate has been that fault rate has been reduced to 200 faults per second. A detailed analysis of the system shows that the bottleneck is located in the communication link between the computer and the board.

The system can perform a detailed analysis of how an injected fault is propagated through the netlist. How it is affected until it reaches a primary output. Both, campaign and single fault analysis are supported in the same framework.

Conclusions

A new framework for fault tolerance measurement has been presented. Xilinx FPGA plays an essential role because of its partial readings and writings of the configuration circuit and the capture and readback scheme accelerates the bit-flip insertion and circuit analysis.

The framework has a software toolbox that allows fault tolerance measurement at any stage of the design. The test is 100% non intrusive, even a post synthesis model of the module under test can be the input of the system. No extra work is needed.

As future work more effort in characterization of the tool is needed. Secondly a study of where are the weaknesses of the technique should lead to an improvement of the tool performance. Finally a study for determine the size of the MUT in which the tool must be used.

References

[1] M. Whirthlin, E. Johnson and N. Rollins, M. Caffrey and P. Graham. "The reliability of FPGA Circuit Designs in the Presence of Radiation Induced Configuration Upsets" IEEE Symp. On Field Programmable Custom Computing Machines 2003 (FCCM'03)

[2] A. Fernández-Leon "Field Programmable Gates Arrays in Space". IEEE Instrumentation & Measurement Magazine. Dec 2003.pp 42-

[3] H.R. Zarandi, S.G. Miremadi, A. Ejlali, "Dependability Analysis Using a Fault Injection Tools Based on Synthesizability of HDL Models", Proceedings 18th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'03), Boston

[4] J.C. Baraza, J. Gracia, D. Gil, P.J. Gil, "A Prototype of a VHDL-Based Fault Injection Tool. Description and Application". Journal of Systems Architecture, 47(10):847-867. Abril 2002.

[5] L. Berrojo, F. Corno, L. Entrena, I. González, C. López, M. Sonza Reorda, G. Squillero "An Industrial Environment for High Level Fault-Tolerant Structures Insertion and Validation". VTS2002: 20th IEEE VLSI Test Symposium, Monterey, CA (USA), 28 April - 2 May, 2002, pp. 229-236

[6] M.A.Aguirre, J.N. Tombs, V.Baena, J.M. Carrasco, A. Torralba and L.G. Franquelo. "Mi-croprocessors and FPGA interfaces for in-system co-debugging in field programmable hybrid systems" Accepted for Elsevier Microprocessors and Microsystems. Special issue on FPGAs.

[7] M.A. Aguirre, J.N. Tombs, A. Torralba and L.G Franquelo, "UNSHADES-1: An advanced tool for In-System Run-Time Hardware Debugging". Proceedings of the Field Programmable Logic and Applications. Lisbon 2003. pp 1170-1173.

[8] R. Velazco, R. Leveugle and O. Calvo. "Upset-like Fault Injection in VHDL Descriptions: A Method and Preliminary Results". Proceedings on the 2001 IEEE international Symposium on Defect and Fault Tolerance in VLSI Systems (DFT'01)

[9] Leon2 datashhet. GAissler Research.

[10] Xilinx application notes 138 and 151.

[11] P. Civera, L. Macchiarulo, M. Rebaudengo, M. Sonza Reorda, M. Violante. "Exploiting Circuit Emulation for Fast Hardness Evaluation" IEEE Transactions on Nuclear Science, Vol 48, nº 6, 2001

[12] J. W. Wilson, I. W. Jones, D. L. Maiden and P. Goldhagen "Atmospheric Ionizing Radiation (AIR):Analysis, Results, and Lessons Learned.From the June 1997 ER-2 Campaign" NASA/CP-2003-212155. February 2003.

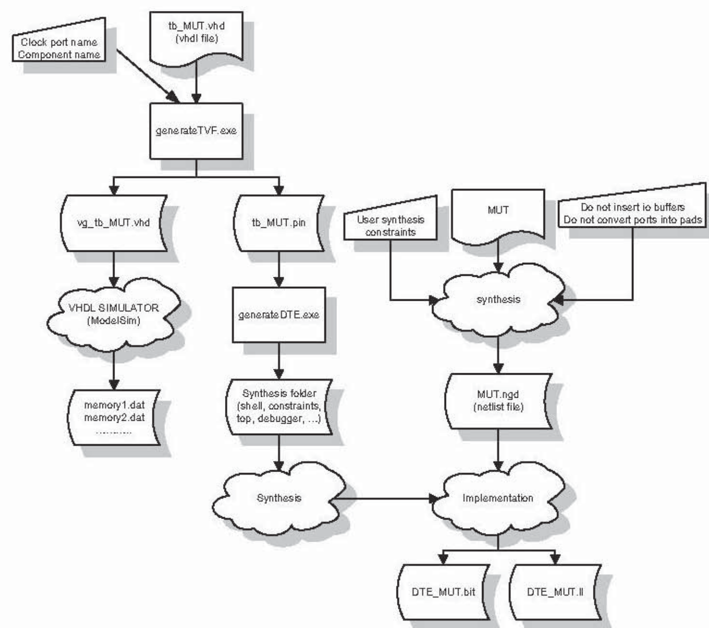

Figure 5. DTE generation flow