VHDL Implementation of FPGA Based High Speed Fault Tolerance Tool

1.R.SIDHARDHA , *MTECH(VLSI)* ,*JBREC* , rathnamsiddharth@gmail.com
2. P.NAVM SASTRY , *Professor* , *JBREC* , shastrypnvm@gmail.com
3. Dr.D.N RAO, *Principal* , *JBREC*, principal_jbr@yahoo.com
1,2,3.Joginapally BR Engineering College ,Yenkapally, Moinabad Mandal , R.R. District, Hyderabad Telangana State.

Abstract- **Fault injection is mainly used to test and evaluate the fault-tolerance based designs. In current VLSI technology fault injection has become a popular technique for experimentally determining dependability parameters of a system, such as fault latency, fault propagation and fault coverage. There are fundamentally two types of fault injection methods; they are hardware-based fault injection and software-based fault injection. Both have their own limitations and advantages. The FPGA synthesizable fault injection model can give reasonable solution with high speed testing platform and also allows good controllability and observability. Even though a considerable progress has been made in research part of the fault injection algorithms, there is a little implementation is done which could be of great interest to VLSI industry.**

In this project an FPGA-based fault injection tool (FITO) that supports several synthesizable fault models for dependability analysis of digital systems modeled by Verilog HDL. Aim is to build real time fault injection mechanism with good controllability and observability. Fault injection will be done by applying some extra gates and wires to the original design description and modifying the target Verilog model of the target system. The design will be validated with state machine based example and applying different types of faults. Analysis will be carried out studying the controllability and observability of the proposed scheme. Comparison will be carried out to estimate the speed wise improvement with respect to software simulation based fault injection method.

Keywords— **Fault injection***; Fault tolerance; FPGA implementation;* **hardware description language**

I.INTRODUCTION

In this project we implement an FPGA-based fault injection tool, called FITO that supports several synthesizable fault models of digital systems.By using this fault injection experiments can be performed in real time with good controllability and observability. Different Fault injection methods and advantages of FPGA based fault injection technique over other fault injection methods

In the simulation based fault-injection, faults are injected using hardware description languages (HDL) and simulation software. The main advantage of such approaches with respect to other types of fault injection is the larger observability and controllability. However their main drawback is their big time consumption. On the other hand, FPGA-based fault-injection allows faster emulation experiments as well as a good controllability and observability. the VHDL code may be available. FPGA-based fault injection methods became largely used for the error-rate estimation of integrated circuits, even before their fabrication, exploring thus the efficiency of implemented rad-hard designs.

In this work a new fault-injection method/tool called FITO (Fault Injection Tool) is presented. It injects faults in any HDL model, VHDL,Verilog etc. In addition to its speed and automation, the main contribution of this method is that it can target the BRAMs (block rams) of a given VHDL circuit.

Automating the steps from synthesis to fault injection was also a goal of this work so that for any circuit or system described in VHDL that work properly on the FPGA, fault-injection can be performed with minimum time and effort. Finally, it is important to mention that at the opposite of many state-of-the-art FPGA-based fault-injection methods, the size of the FPGA is not a limitation for FITO.

The fault injection is a technique of Fault Tolerant Systems (FTSs) validation which is being increasingly consolidated and applied in a wide range of fields. Fault injection is the validation technique of the Dependability of Systems which consists in the accomplishment of controlled experiments where the observation of the system's behavior in presence of faults is induced explicitly by the voluntary introduction (injection) of faults to the system.

The fault injection methods are classified as shown below.

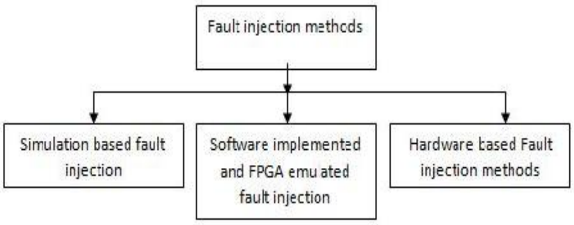

Fault injection methods
Simulation based fault injection
Software implemented and FPGA emulated fault injection
Hardware based Fault injection methods

Fig : Types of fault injection methods

Simulation based fault injection

In this technique, the system under test is simulated in other computer system. The faults are induced altering the logical values during the simulation.

Software implemented and FPGA emulated fault injection

In this the fault injection plan will be made in software usually at the RTL level, but the actual application of faults and capturing the fault response happens on FPGA emulation phase.

Hardware based fault injection

It is accomplished at physical level, disturbing the hardware with parameters of the environment (heavy ions radiation, electromagnetic interference, etc.) or modifying the value of the pins of the integrated circuits.

In software based fault injection mainly we rely on simulation of faults and observing the fault responses. Mainly this technique limited by huge simulation time.

Advantages of FPGA based fault injection technique are

(1) High controllability and observability
(2) High speed fault injection experiments with target system running at full speed
(3) Capability to inject permanent and transient faults.
(4) Minimum time and area overhead into a target system.

II . THE BLOCK DIAGRAMOF FITO

FITO environment consists of three parts :

1- Source Code Modifier & Fault List Generator
2- Fault Injection Manager
3- Result Analyzer

Source Code Modifier & Fault List Generator:

In Source Code Modifier we are modifying the VHDL code. In Fault List Generator whatever faults we are injecting it will put it into one log file at the end when u ask to generate the list of all the faults we included it will give summary.

Fault injection manger :

Fault injection manger is responsible for performing the real time fault injection.

Result Analyser:

Whatever coming from the serial port, what is fault inserted and what is the result response is analysed.

.

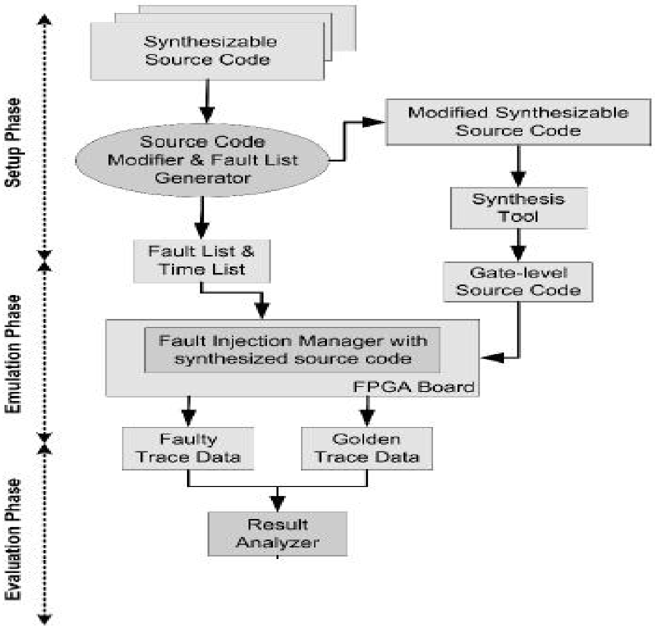
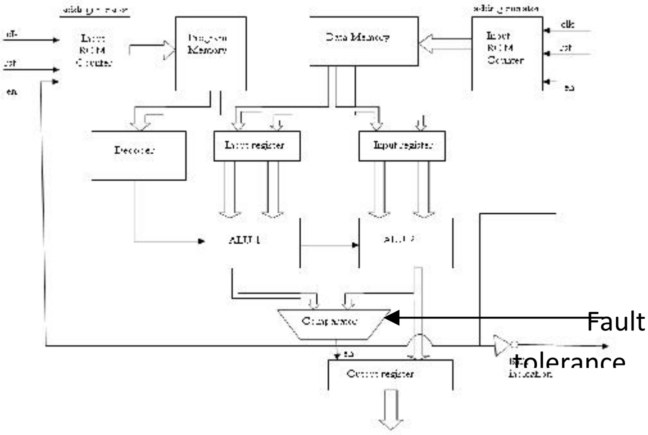


Fig: Types of fault injection methods

This hardware part is implemented on the FPGA board. Result analysis will be carried out with FPGA emulation results and fault list generated by C program. The analysis shall summarize the fault responses for each injected fault. The golden trace data is ideal trace data (results obtained) without any faults.



Architecture of fito

At a hardware level, fault tolerance is achieved by duplexing each hardware component. Disks are mirrored. Multiple processors are "lock-stepped" together and their outputs are compared for correctness. When an anomaly occurs, the faulty component is determined and taken out of service, but the machine continues to function as usual.

The same technique is used in our architecture by duplexing the ALU unit and comparing the results. An enable signal is provided by the comparator which is used to iterate the function and produce fault free results. The address generator part of the block diagram access the data and is stored in two different input registers and are processed separately by different arithmetic and logic units to achieve redundancy. The enable signal also indicates the occurrence of faulty signal. When it is assured that the fault has not occurred, the output is taken from the output register

III . VHDL IMPLEMENTATION OF FITO MODULES

The Fault injection manger is responsible for performing the real time fault injection. The fault injection manager is implemented in VHDL. The fault injection manager

- VHDL package (dynamically updated by C programs)
- Fault scheduler
- Fault injection components

Fito package

The VHDL package is implemented to capture all the constants, type definitions, component declarations and fault injection time for each fault. The package also consists of number of total faults. This VHDL file is automatically updated by C programs every time when a fault is injected in code.

The following section gives the code and explanation of the package used in our design.

The maximum number of faults are taken to be 63 and based on that the other constants are defined. However there is no limitation of the maximum number of faults that can be inserted. Depending on the requirement one has to the constant's values in this package. Another constant is defined to give the number of injected faults. This constant value is updated by C program every time when a new fault is inserted.

FIS vec_type defines a bus of size equal to number of faults. This bus goes through all modules such that any module can use the control lines for fault injection. It may appear that by routing 64 length wider bus to all small and big modules of design under test we are consuming high number of FPGA routing resources. But the synthesis tool can optimize the resources by only routing the lines which are used in this module.

Constant by name Fault injection signal (FIS) high duration indicates the number of clock cycles for which fault will be injected in the design. FIS duration constant tells the time allotted for each fault. That is even after removing the fault we can wait for output to capture before enabling the next fault. This will be useful in cases where the fault propagation time is high. For every fault these two constants are settable in the GUI.

The constant fault type defines the type of fault as per the below table. For each fault that is injected used will choose this option on GUI.

Table 1: constant faults type

Constant value	Fault type
0	Stuck at 0
1	Stuck at 1
2	Transient
3	Bit flip

Package also holds several component declarations. So that in all modules if this package is declared then fault injection only requires giving component instantiation (no need to declare the components)

Fault scheduler

The fault scheduler runs multiple counters to schedule each fault with required fault activation time and fault propagation time as per the constants in FITO package. The fault scheduler produces output fault number which is currently being active. This module generates the parallel fault injection signals for every fault. These signals are routed to all fault sites.

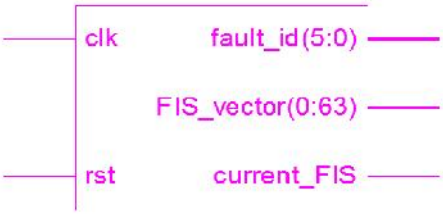


Fig: schematic of fault scheduler

The following code is the VHDL implementation of fault scheduler

3.5.3 Fault injection components

Fault injection components are gates with FIS (fault injection signal) control to inject the faults when the FIS is active high. These components instances are automatically made in the selected module when ever faults are injected. The following section gives the codes for fault injection components.

Since all these components are declared in package fault injection need not add component declaration. Hence the fault insertion becomes easy to implement only the following steps.

(a) Generate code to declare a signal of the same size of the port on which fault need to be injected.

(b) Add the corresponding fault injection component instance connecting the port signal, FIS control line and output signal.

(c) Replace all the port signal instances with the declared new signal.

The following section shows the VHDL code for all the fault injection components. Since the VHDL coding syntax need to change for single bit port and buses for each fault two versions of components are created. One to inject fault on single port signals and other to inject on to vectors.

Random bit generator for bit flip fault

A random bit generator for bit flip fault is implemented with a Gaussian random variable generated through a Look up table. A Look up table with 127 values is taken and is used to randomly flip the bits in memory when the bit flip fault is activated.

faults modeled in our project

FITO supports the following synthesizable fault models for injecting into any HDL level designs.

- Permanent faults
- Transition faults
- Single event upset faults (or) Bit-flip

Fault injection process can be done by applying some extra gates and wires to the original design description and modifying the target VHDL model of the system. One of these extra wires is the Fault injection system (FIS) which playing the key role in the fault injection experiments. If a FIS takes the value 1, fault would be activated and if it takes the value 0, the fault would become inactive.

For example in the case of Stuck-at-0 fault when the FIS is made 1 then the signal is forced to zero, implementing the fault condition. The below section gives the detailed discussion about injecting the permanent faults.

For supporting the permanent faults in VHDL design, FITO nominates wires for fault injection and apply the FIS signal with one extra gate,. So by selecting the FIS signal high at fault injection time, the permanent fault into the specified wire will be injected.

For example if the signal name in the original code is X then the modified signal TX will be generated as below. In all the places in the code instead of X, the TX will be replaced.

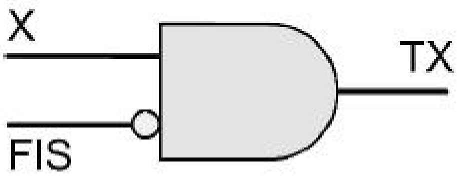


Fig 3.2: synthesizable fault model for stuck-at-0

Similarly the following code shows the required extra gate and control signal FIS for implementing the stuck-at-1 fault.

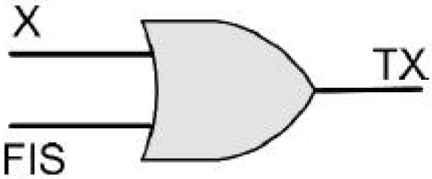


Fig 3.3: synthesizable fault model for stuck-at-1

For each FIS there would be a path through all levels of hierarchy to its modified circuit. After modification, the final synthesizable VHDL description will be produced which is suitable to use in emulators.

For example in the above example if the mux is component in next high level module then the component assignment of mux will be accordingly changed.

Transient faults

The modified circuit that is suitable for transient fault injection is shown in below figure.

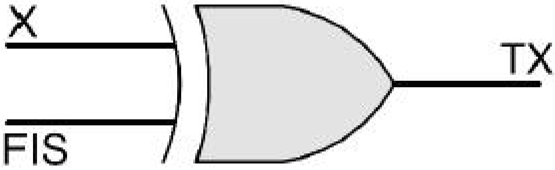


Fig3.4: synthesizable transient fault model

For injecting a transient fault, after reaching the fault injection time, the FIS signal will be made high and the timer, which have been loaded with the duration of the transient fault injection start to count. Therefore, the FIS will be high (at logic 1) for the specified duration of time. As similar to the permanent fault, the additional wire (TX) will be used and each wire, namely X will be replaced with TX.

The fault injection manager is responsible for managing the fault injection experiments, such as loading the timers, setting the FIS for the predetermined time, introducing additional wires and performing the fault injection.

Bit flip or single event upset (SEU)

The fault model that is used by FITO at this level is bit-flip (or single event upset). SEUs are the random events and may flip the content of the memory element at unpredictable times. FITO generate modified circuit for each memory element that is specified for fault injection. The modified circuit for supporting bit flip fault model is shown in below figure.

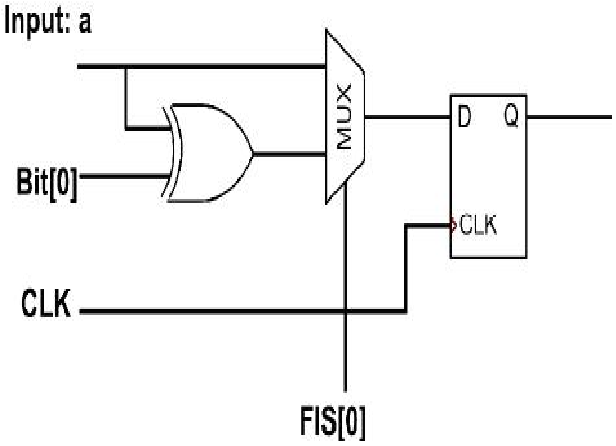


Fig 3.5: synthesizable bit-flip model

For supporting the bit-flip model, FITO produces the additional signals such as Bit and FIS with one multiplexer. The VHDL synthesizable code for supporting this fault model is shown in above figure. The inverterted input will go to the flip-flop for the next clock when the FIS and bit are ‘1’. The fault injection manager part of FITO is responsible for setting and resetting the FIS and bit signals.

IV. SIMULATION RESULTS

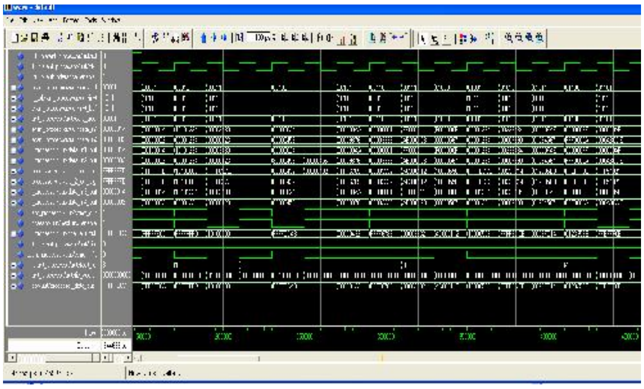


Fig: FITO results after injection the faults

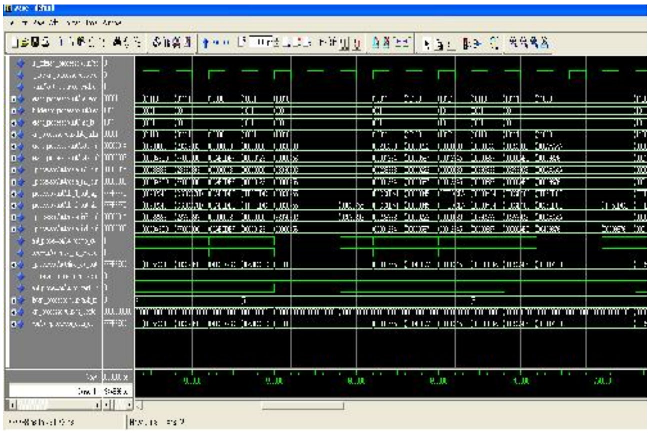


fig: FITO results after injection the faults

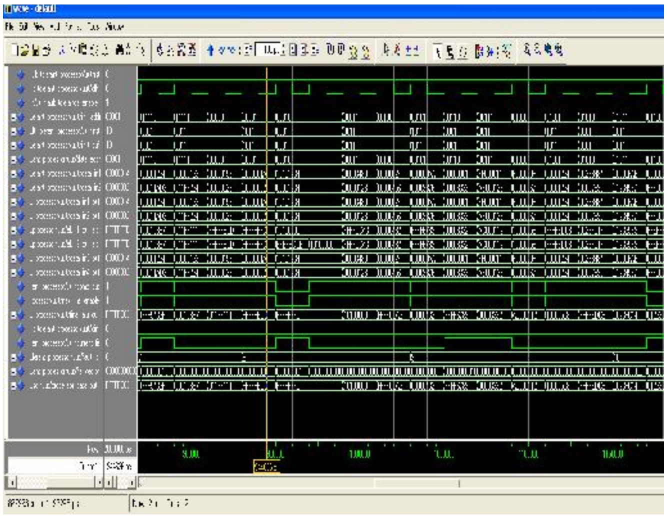


Fig: FITO results after injection the faults

CHIP SCOPE RESULTS

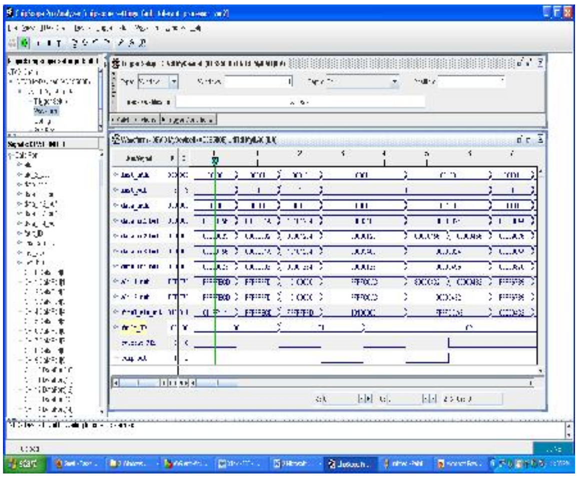


Fig: Chipscope FPGA emulation results

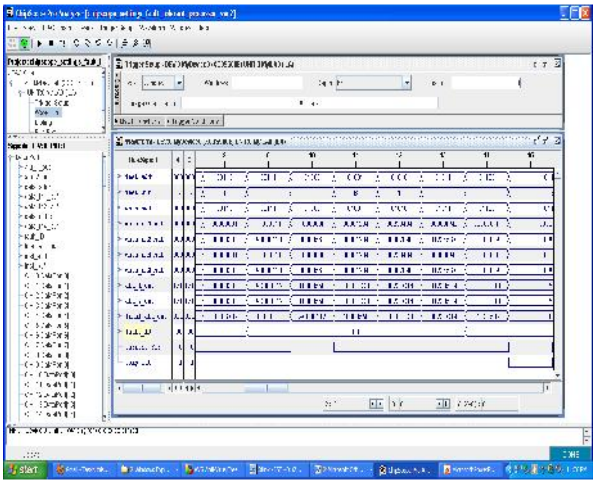


Fig: Chipscope FPGA emulation

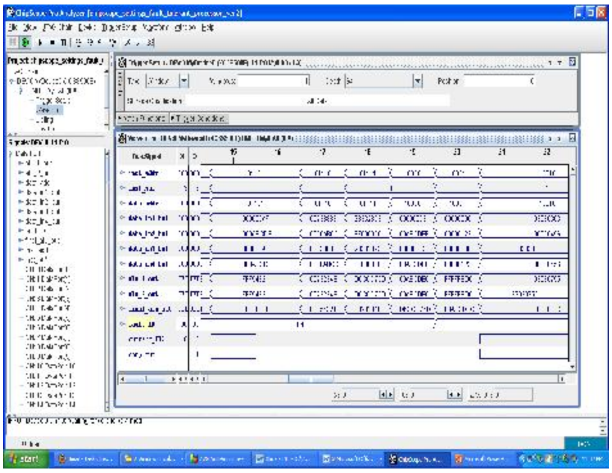


Fig : Chipscope FPGA emulation results

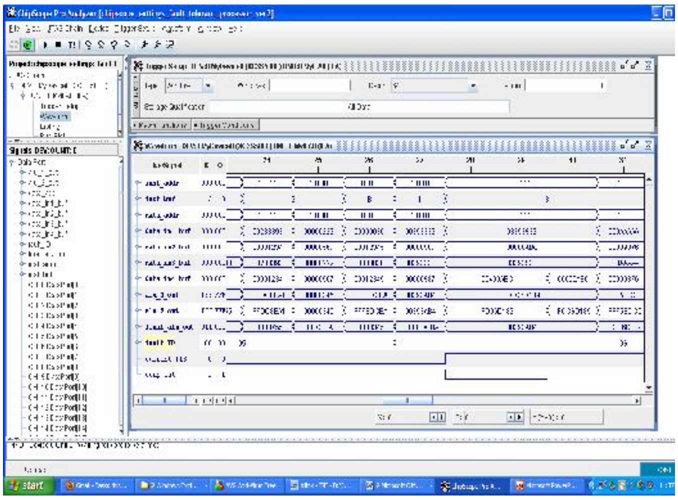


Fig : Chipscope FPGA emulation results

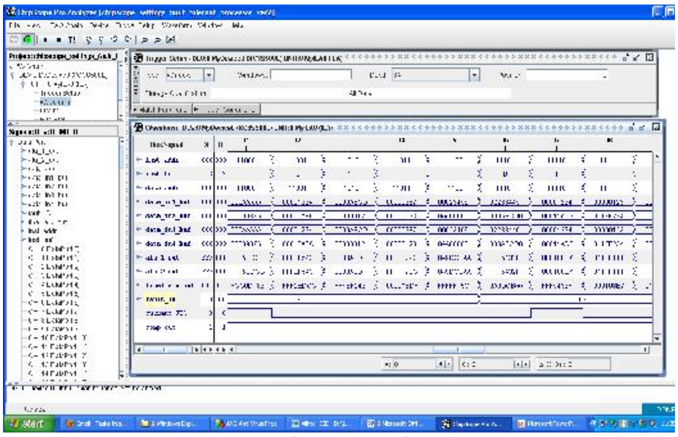


Fig: Chipscope FPGA emulation results

V. CONCLUSION

This paper described the FPGA based fault injection tool called FITO for evaluating the digital systems modeled by VHDL Fault injection with FITO is done by applying some Extra gates and wires to the original design description and modifying the target Vhdl model of the target system. FITO supports some properties such as high speed good controllability, good obsevability and low area overhead

Advantages

High speed as the circuit runs on the hardware with fault injection model (in comparison with simulation based model) We can achieve high controllability and Observability The ASIC design can be first tested in FPGA with fault injection overhead design The ASIC release of the design need not have this. The area overhead on the final design is negligible.

Dis advantages

The technique can't work for big designs due the FPGA capacity limitations However the module wise testing is still possible The clock frequencies at which the faults need to be tested could be different from FPGA version to ASIC version For most of the designs this is still not a problem.

REFERENCES

[1] Course slides from "SoC testing and verification" prof Virendra singh, IISc, Bangalore

[2] Fault Injection into VHDL Models: Experimental Validation of a Fault Tolerant Microcomputer SystemD. Gil, R. Martínez, J. C. Baraza1, J. V. Busquets, P. J. Gil

[3] Ref.: M. J. Howes and D. V. Morgan, Reliability and Degradation - Semiconductor Devices and Circuits, Wiley, 1981.

[4] J. Voas, "Fault Injection for the Masses," Computer, vol. 30, pp. 129–130, 1997.

[5] J. V. Carreira, D. Costa, and S. J. G, "Fault Injection Spot-Checks Computer System Dependability," IEEE Spectrum, pp. 50–55, 1999.

[6] www.wikipedia.com

R.SIDHARDHA , Pursuing M.TECH in VLSI from JBREC college Affiliated from JNTUH. Moinabad, Rangareddy Dist, Hyderabad, Telangana..He got 72% in her Mtech 1^{st} year. he has received the Btech degee in Krishna Murthy Institute of Technology college in 2012

Prof.P.N.V.M Sastry Currently is working with a Capacity of **Dean- IT EDA Software – R&D CELL & ECE** DEPARTMENT, He Did Master Degree In Science- M.Sc Electronics, AU -1998.Did PG Diploma In VLSI Design From V3 Logic Pvt Ltd B'Lore-2001, Did M.Tech (ECE) From IASE Deemed University-2005. Currently Pursuing (PhD)-ECE , JNTU Hyderabad -2012 , Over **Past 16 years of Rich Professional** Experience with Reputed Software MNC's, Corporates – INFOTECH, ISiTECH as a **Team Leader(Level 6) Eng-Eng-HCM Electronics Vertical, Program Lead – Embedded & VLSI& Delivery Manager – IT Semiconductor Software Engineering Vertical** , also worked with **Govt R&D, Industrial Organizations, Academic Institutions** of Comparative Designations

Dr. D.N Rao B.Tech,M.E,Ph.D, principal of JBREC, Hyderabad. His carrier spans nearly three decades in the field of teaching, administration, R&D, and other diversified in-depth experience in academics and administration. He has actively involved in organizing various conferences and workshops. He has published over 11 international journal papers out of his research work. He presented more than 15 research papers at various national and international conferences. He is Currently approved reviewer of IASTED International journals and conferences from the year 2006. He is also guiding the projects of PG/Ph.D students of various universities