ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Instrumentation Oriented Fault Injection Tool Based on FPGA

P. Sampath Kumar
Associate professor & HOD,
Department of ECE,
Malla Reddy Institute of Technology, Hyderabad.

Rama Gopal T S V
M.Tech (VLSI &ES),
Department of ECE,
Malla Reddy Institute of Technology, Hyderabad.

Abstract:

The fault emulation technique is used to evaluate dependability parameters of computer based embedded systems or safety critical systems, by injecting faults in a system. An observation on the behavior or of the response for both -fault free and faulty systems- is made and the results are matched. In the instrumentation based technique, faults are injected by adding some additional circuits in the target system known as fault models. Different fault models are explained in this paper, e.g. Stuck at 0/1, Bit flip, and SEU. A variety of FPGA based tools are discussed and a new FPGA based fault injection technique is introduced. The tool that has been developed according to our methodology can work on any simple or complex digital circuits or SoC realized on FPGA. This can be exercised to evaluate the dependability parameters for the target systems under test e.g. fault coverage, fault latency and reliability

In this project an FPGA-based fault injection tool that supports several synthesizable fault models for dependability analysis of digital systems modeled by VHDL. Aim is to build real time fault emulation mechanism with good controllability and observability. Fault emulation will be done by applying some extra gates and wires to the original design description and modifying the target VHDL of the target system. The design will be validated with ALU based addrer 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 area overhead needed for instrumented logic and speed wise improvement with respect to software simulation based fault injection method.

Keywords:

Fault emulation; Fault tolerance; FPGA implementation; Hardware Description Language.

I.INTRODUCTION:

Recently the principle of fault injection has become a widely used technique in order to calculate dependability parameters of embedded and safety critical systems. These systems are based on computers or microprocessor systems which are usedmostly in areas where failure leads to a huge problem, such as railway traffic control, aircraft flight schedules and space craft etc. Due to the condensed size of components in electronic systems, it is difficult to guarantee an acceptable degree ofoperational reliability of a system, hence testing and verification are more important. The dependability parameters for embedded systems and computer based systems are verified before they are handed over to the end user. Four main methods exist in which fault injection testing is possible namely:

- Hardware based fault injection
- Software based fault injection
- Simulation based fault injection
- Emulation based fault injection

There are certain merits and demerits of every method stated above. Detailed information about this, is presented by H Ziade et al [3]. Recent development in the field of programmable logic devices i.e. FPGA (Field Programmable Gate Array) has allowed safety critical systems and computer based systems to be realized on it.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

It provides many properties like reconfigurability higher speed, higher performance andcost effectiveness [4] but FPGA devices are also required testing. For that purpose various FPGA based fault injection tools and techniques are presented in the literature.The FPGA based fault injection method is the most widely used nowadays because of the FPGA's features mentioned above. The FPGA based tools cover all benefits of simulation based tools. Furthermore, they are able to overcome the majordisadvantages of time consumption and hence increase the speed of the fault injection process. FPGA based fault injection tools can be categorized into two main groups [2].

1)Reconfiguration based technique
2)Instrumentation based technique

The first group takes advantage of the features provided by FPGA, i.e. reconfiguration, partial reconfiguration. In this technique the faults are injected by changing bits in different copies of the bit stream. The major downside of this method is the reconfiguration of the FPGA every time the fault injection experiment is performed, which is the source of time overhead. While in the technique of the second group additional circuits are added or a modification of the original code is carried out, which contributes in area overhead. This paper is focused on the second approach and presents the latest techniques and tools for FPGA fault injection. The Section 2 describes background and proposed new methodology. Section 3 illustrates the basic building block and functioning of any fault injection tools. Section 4 presents the results of the experiments, tools used. Finally, the conclusion of this paper is presented

II . Fault Injection Environment:

From literature survey, it is beheld that fault injectiontools are designed with the use of a variety ofapproaches and methods. Basically two devices are needed to perform this whole operation, a host PCand a target board (FPGA board for emulation).Mostly both devices communicate on serial links fordebugging. The host PC contains the software partwhich is comprised of the fault list manager, the faultlist and the result analyser while the target boardholds the fault injection manager or the fault injectionunit. The fault injection system can be divided intothree fundamental modules which are outlined below.

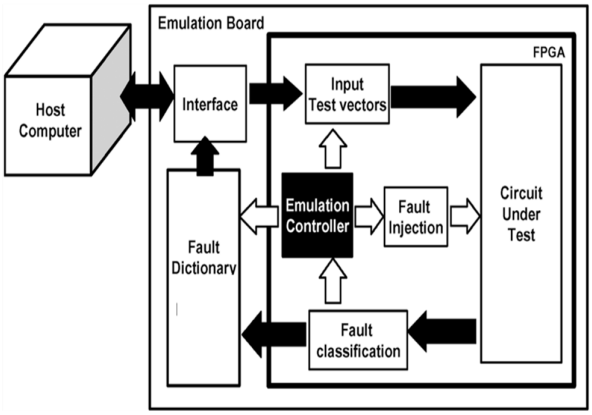

Fig: Fault Injection Environment

Environment consists of three parts:

1- Fault List
2- Fault Injection Manager
3- Result Analyzer

Fault List:

The Fault list manager is in charge of generating the list of faults which will be injected in thetarget system (FPGA board). This handles the types of faults injected in the system. The faults are injected by user selection at RTL level.

Fault injection manager:

The main purpose of the Fault injection manager is to control and activate faults from the fault list manager on the target system. In this circuit instrumentation technique, additional circuits are added for faults whose activation is controlled by this block.

Result Analyzer:

The task of this module is to comprise the gatheringof the results so that a report can be generated during the fault injection experiment. It is beheld that the fundamental operation of any fault injection tool is to inject the fault in the target system and observe the output before and after the activation of faults. Following this, the next both results are compared. Based on that, fault coverage, fault latency and other dependability parameters are examined.In FPGA based fault injection techniques and tools, faults can be injected in many points as shown in Figure 1[14].

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In order to develop the FPGA based tool, there are different abstraction levels to program a FPGA e.g. gate level, data flow and RTL level. The procedure to program a FPGA consists of certain steps of programming from higher abstraction level VHDL or Verilog program to bitstream generation. In this paper RTL abstraction level is picked out. The code can be modified in order to inject faults in the target circuit/system. The contributions in this regard are presented in the background section.

III .VHDL IMPLEMENTATION OF TOOL 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

Fault Injection 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 64 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: Faults 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 tool 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.

Fault injection components:

Fault injection components are gates with FIS (fault injection signal) control to inject the faults when the FIS is active high.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

These components instances are automatically made in the selected module whenever 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:

The tool 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, tool 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.

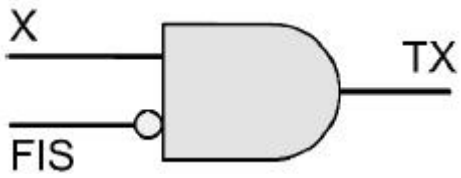

Fig : synthesizable fault model for stuck-at-0

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

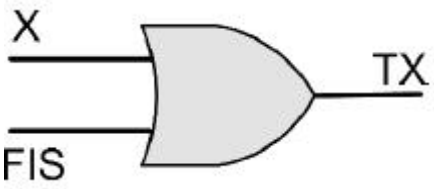

Fig 3.3: synthesizable fault model for stuck-at-1

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.

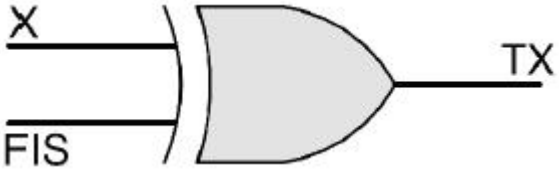

Fig3.4: synthesizable transient fault model

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

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 Tool 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. Tool 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.

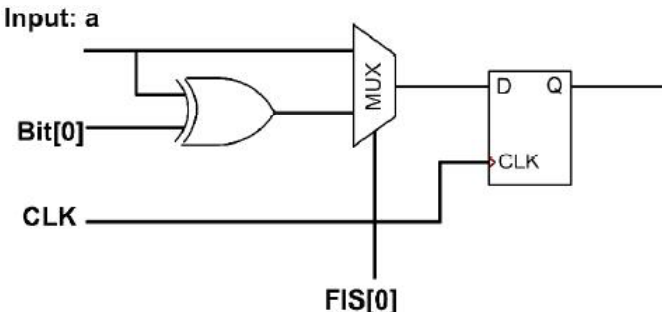

Fig 3.5: synthesizable bit-flip model

For supporting the bit-flip model, tool 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 inverted input will go to the flip-flop for the next clock when the FIS and bit are '1'. The fault injection manager part of tool is responsible for setting and resetting the FIS and bit signals.

IV. SIMULATION RESULTS:

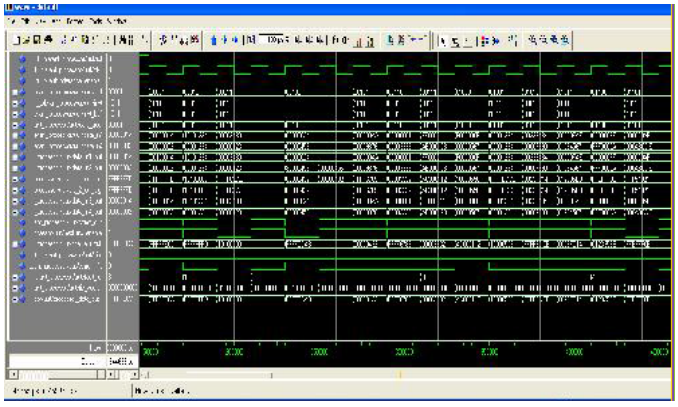

Fig: Simulation results after injection the faults

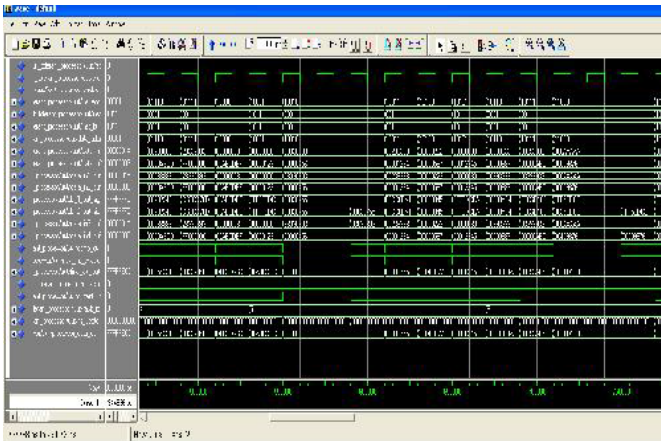

fig: Simulation results after injection the faults

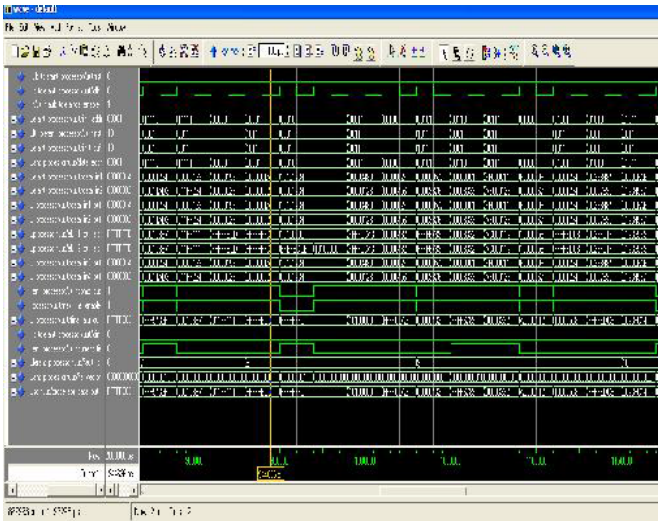

Fig: Simulation results after injection the faults

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

SIGNALTAP RESULTS:

Fig: SignalTap FPGA emulation results without Faults

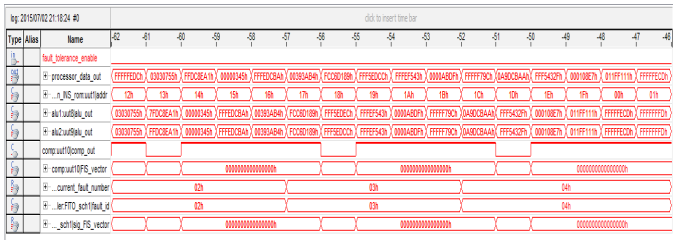

Fig : SignalTap FPGA emulation results with Faults

Fig : SignalTap FPGA emulation results with Faults

These results show that in the fault non-injected version the internal comparator output signal is always high. For the fault injected version the comparator output is low for the fault effected duration.

Analysis of Fault Emulation module resource usage:

Design Module	Resource FF + LUT
Fault Scheduler	21 + 50
SA-0 Fault emulation	0+1
SA-1 Fault emulation	0+1
Bit-Flip(SEU) Fault emulation	7+23
Transient Fault emulation	0+1

Based on the resource analysis to inject twenty faults the approximate increase of resource is negligible. This current version of the tool is capable of upto 64 faults injection at one run.

The same experiment could be repeated in an automated way for complete fault coverage. And, fault latency also can be estimated by using this tool.

Tools Used:

Synthesis, P&R : Altera Quartus II 13.0 sp1
Simulations: ModelSim SE 6.2c
On Chip Debug: Signal TapII Logic Analyzer

HW Platform:

Device: CycloneIV E EP4CE40F23I7
Download cable: USB BlasterII

V. CONCLUSION:

The purpose and objective of fault injection tools is to inject faults and match the faulty response with the fault free. In instrumentation based techniques additional circuits are added in order to inject faults. The most commonly used fault models are presented. Finally, we implemented this technique to automate the fault injection process and it is shown how all fault models can be applied for DUT. Detailed analysis presented with respect to the device utilization for this extra Fault emulation logic modules. The fault models are added to the target system and inject faults, when activated. Numbers of faults for each fault model are injected. The faults are activated one at a time. The fault injection tool needs to be developed for analysing each fault model and calculate different dependability parameters.

REFERENCES:

[1] J. Fan and Z. Zhang, "Speeding up Fault Simulation using Parallel Fault Simulation," Procedia Eng., vol. 15, pp. 1817–1821, Jan. 2011.
[2] M. S. Shirazi, B. Morris, and H. Selvaraj, "Fast PGA-based fault injection tool for embedded rocessors," Int. Symp. Qual. Electron. Des., pp. 476–480, Mar. 2013.
[3] H. Ziade, R. Ayoubi, and R. Velazco, "A Survey on Fault Injection Techniques," vol. 1, no. 2, pp. 171–186, 2004.
[4] 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.
[5] Ref.: M. J. Howes and D. V. Morgan, Reliability and Degradation - Semiconductor Devices and Circuits, Wiley, 1981.
[6] J. Voas, "Fault Injection for the Masses," Computer,vol. 30, pp. 129–130, 1997.