IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 5, Ver. II (Sep. - Oct. 2016), PP 22-27
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Transient Fault Injection in 4 bit Ripple Carry Adder using Random Sequence Generator

Prof. Vaijayanti H Panse
Dept. of EC, Ramdeobaba College of Engineering and Management, Nagpur, India

Abstract: *Fault injection is mainly used to insert a fault in the fault-tolerance based systems. A fault injection technique that enables designers to insert a fault on any signal within the block of a VHDL code is presented. VHDL signals provide the capability to access the values in the code which makes it more testable and observable. The proposed design gives a fault injection system that injects the transient faults in the inputs of a 4 bit ripple carry adder at specific locations. Depending on the control signal generated by a random sequence generator, particular fault is generated in the circuit. The user has a control over generating a fault in the system under test at desired interval and location, so the proposed fault injection system is simple and user friendly. The vhdl code is developed and simulated in Xilinx 9.1i.*

Keywords: *Fault injection, PN sequence generator, VHDL, Xilinx*

I. Introduction

Before actual implementation of any digital system, the design needs to be evaluated based on various criteria. Digital systems are usually represented in a high level language such as VHDL. A fault injection system provides the capability of introducing a fault at any desired location into the VHDL module of a circuit [1]. The faults are divided into two categories: permanent and transient. Permanent faults that exist in logic circuits are normally identified during offline testing by the manufacturer of the IC and the transient faults are the soft faults generated at the consumer side. The occurrence of transient faults in VHDL description of the circuit is important if the circuit has on line fault detection capability. Internal signals can be accessed at the VHDL level for injecting faults in the module, thus it gives greater controllability and observability of the system. Faults in an online testable system are considered mainly as single bit faults where a single bit is flipped from logic 1 to a 0 or vice-versa. It is always better to know when the fault is inserted in the system rather than randomly inserting a fault. The fault injection system proposed in this paper gives the module for multiple fault injection in 4 bit ripple carry adder using random sequence generator. This module allows the user to insert the faults at a certain time and indicates whether the fault is inserted or not. Also, it gives the location of fault at a particular condition. The module is simulated in Xilinx 9.1i.

The organization of the paper is follows. Section 2 discusses the block diagram and basic idea of the proposed fault injection system. Section 3 gives the constituent blocks in the fault injection system. Section 4 shows implementation of fault injection system in VHDL language. Section 5 shows the simulation results, and section 6 is the conclusion.

II. Fault Injection System

The increased dependence on the systems generates a need to focus on their reliability and safety. A faulty system can cause great losses in various industrial applications. Moreover, the systems are often expected to function correctly for a number of years, possibly without maintenance. Fault tolerance is fundamental requirement to assure that those systems are reliable.

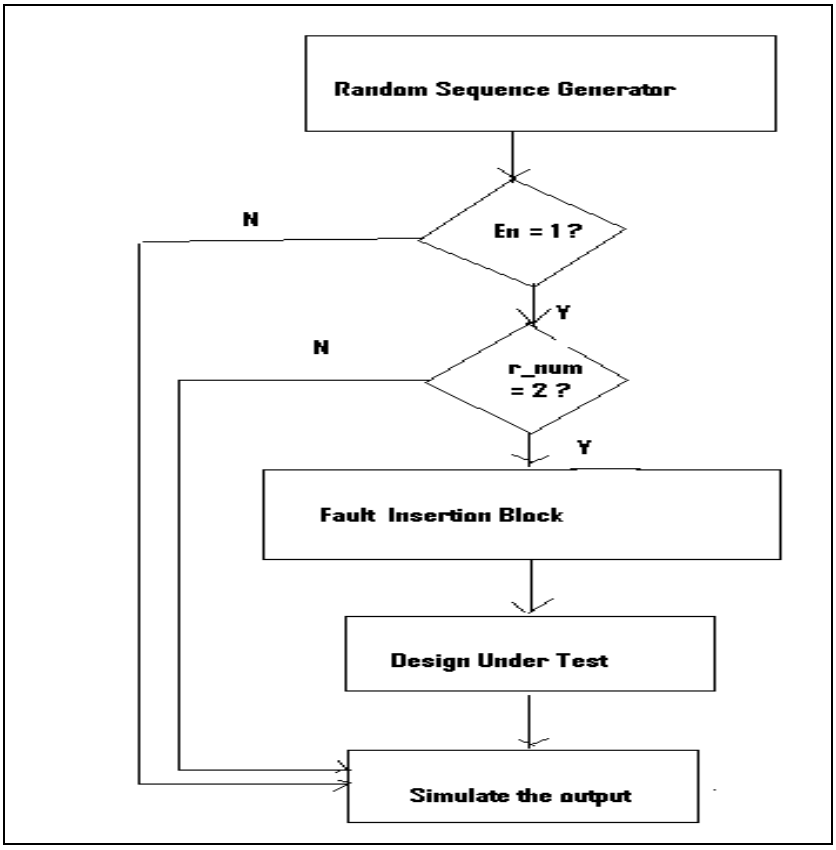


Fig 1. Flow chart for Fault Injection system

Digital systems are usually represented in a high level language such as VHDL. The actual implementation of the system is then performed using their specification in VHDL. Several important criteria of a system need to be evaluated. The capability to check the testability of a system at the VHDL level before it is implemented, allows design to be more fault tolerant and reliable. A fault injection system provides the capability of introducing a fault at any desired location into the VHDL model of a circuit. The injection technique allows the user to insert faults at varying levels of VHDL hierarchy and hence help in evaluating the performance of a testable system.

Fig.1 shows the flow chart for the proposed fault injection system. An 8 bit random sequence generator was implemented using LFSRs. The sequences generated are checked for a desired condition if the system is enabled. When the sequence gets the desired value, a fault is injected in the design under test. Here, the DUT is 4 bit ripple carry adder. After the injection of fault, two signals are generated, one for the fault inserted and the other for giving the location of fault. The circuit is simulated on Xilinx and the output of the design under test is checked for the injected errors. If the fault injection block is disabled, the system will not operate and the design under test gives its estimated output.

III. Constituent Blocks In Fault Injection System

3.1 Random Sequence Generator

The random sequence generator is shown in fig.2. This is an eight-stage shift register with a data input that is xor function of two most significant bits. An all-zero state is not allowed. It produces a sequence of fifteen numbers that repeat every fifteen clock cycles.

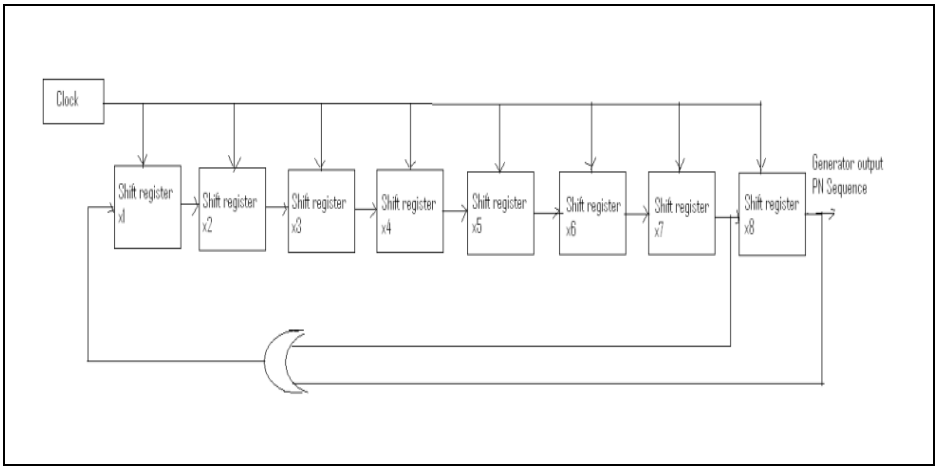


Fig2. Random Sequence Generator

3.2 Dut – 4 Bit Ripple Carry Adder

A full adder is a combinational circuit that performs the arithmetic sum of three input bits: Ai,Bi, and carry in Ci from the previous adder. And give the result containing the sum Si and the carry out, Co to the next stage. Thus, to design a 4-bit adder circuit, firstly designing the 1 – bit full adder then connecting the four 1- bit full adders to get the 4- bit adder as shown in fig.3. For the 1-bit full adder, the Truth Table for the three input and two outputs SUM and CARRY is prepared. The binary full adder along with the truth table is given in fig.4. The Boolean expression for Sum and Carry out is given as:

$$S_{out} = A \oplus B \oplus C$$
$$C_{out} = AB + AC + BC \qquad (1)$$

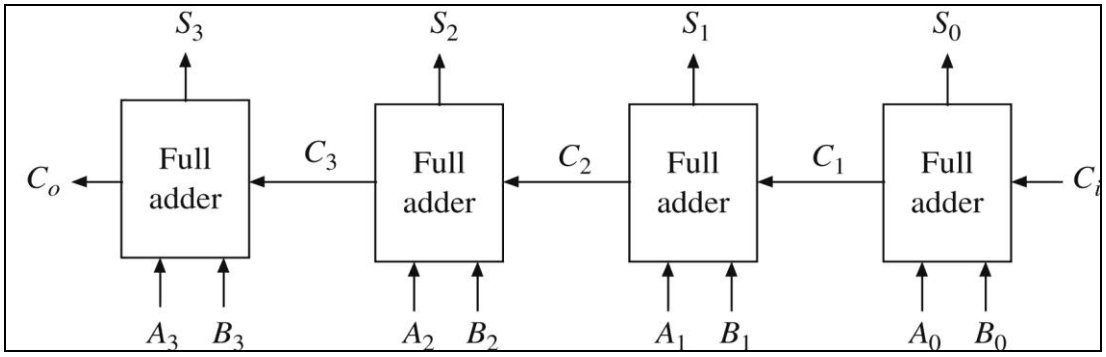
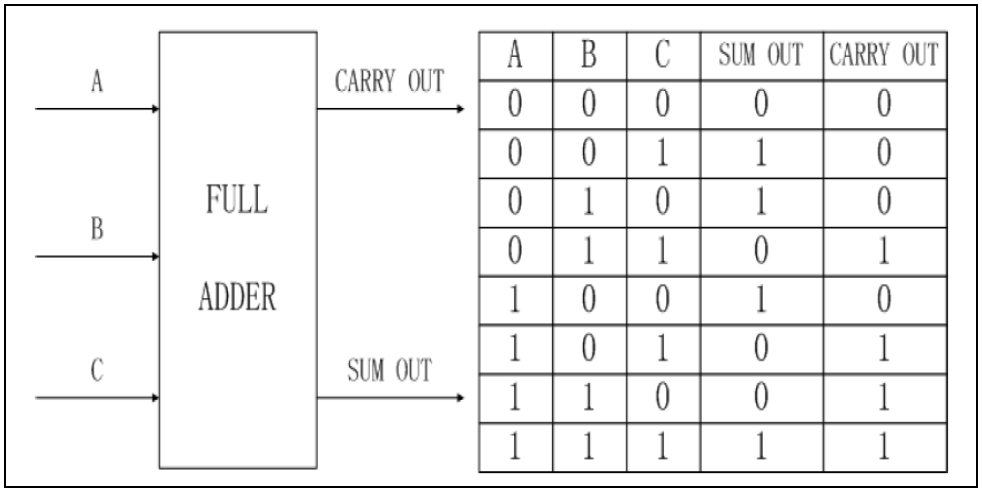


Fig 3. 4 Bit Ripple Carry Adder



A	B	C	SUM OUT	CARRY OUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Fig 4. Block diagram and truth table for full adder

3.3 Fault Injection Block

The fault injection block has three inputs, namely, Enable, random number generated and the DUT inputs. The DUT inputs are created from the signals in VHDL as dummy inputs. The output is calculated depending on the values of inputs. The block diagram of proposed fault injection block is as shown in fig.5. The block is implemented such that when the random number reaches to a particular value, the fault is injected in the design under test.

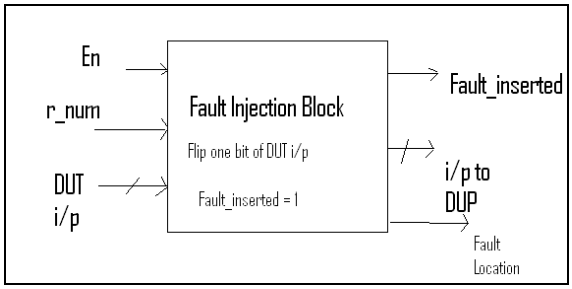


Fig.5 Fault injection block

The fault is implemented by flipping one of the bits of inputs. If the fault is inserted by the block, the fault inserted signal is activated and the location is specified by the fault location signal. The faulty inputs are then given to the design under test to check for the faulty outputs.

IV. Implementation Of Fault Injection System In Vhdl

4.1 Random Sequence Generator

The fault injection system has the ability to insert faults at desired intervals. To accomplish this task the injection system uses pseudo-random sequences. The code is written for 8 bit random sequence generator using simple design approach in VHDL. When the random sequence reaches to a particular value, one of the bits of input is flipped.

```
entity random is
   Port ( clk,rst : in  STD_LOGIC;
   r_num : out  STD_LOGIC_VECTOR (07 downto 0));
end random;
```

4.2 Fault Injection Block

The proposed fault injection system is able to inject faults into VHDL descriptions and into behavioral or structural VHDL coding. The fault injection block is meant to insert faults at many low-level VHDL blocks containing primary inputs and outputs, internal signals as possible. The fault injection block is placed such that it gets inputs as the input of DUT and gives the output as faulty input of DUT, if enabled. The placement of fault injection block is shown in fig.6

```
entity fault_gen is
  Port ( clk,rst, en : in  STD_LOGIC;
  r_num : out  STD_LOGIC_VECTOR (07 downto 0);
  fault_inserted : out  STD_LOGIC;
 in1,in2 : in  STD_LOGIC_VECTOR (03 downto 0);
  cin : in  STD_LOGIC;
  sum : out  STD_LOGIC_VECTOR (03 downto 0);
  carry : out  STD_LOGIC)
end fault_gen;
```

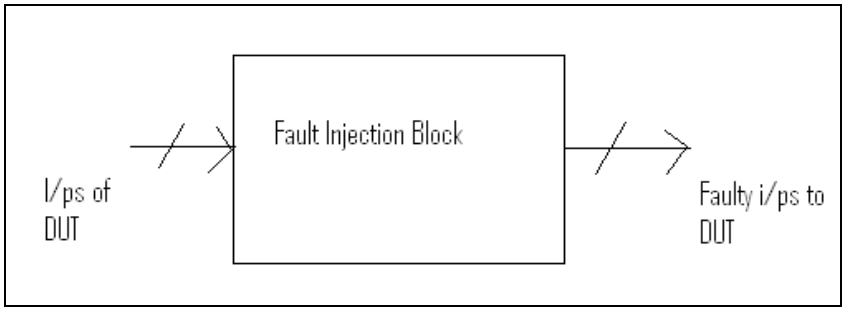


Fig 6. Placement of Fault injection block

4.3 Dut Component Instantiation

To illustrate the use of fault injection system in an on-line testable circuit, the fault injection block must be inserted into a VHDL block that is meant to be self-checking. A self-checking circuit can determine whether a fault has propagated to an output data word or not. For this proposed system, a 4 bit ripple carry adder is used as DUT to check the injected errors. The faults are injected at the inputs of adder using signals in VHDL as dummy inputs to the DUT.

```
component adder4
port (a,b: in std_logic_vector(3 downto 0);
      ci: in std_logic;
     S: out std_logic_vector(3 downto 0);
     Co: out std_logic);
end component;
```

V. Simulation Results

The VHDL coding and simulation of the system was performed on the Xilinx Foundation 9.1i. It is observed that if the enable signal is on, then the one of the bits of input is flipped and fault inserted becomes one. The corresponding change in the output of adder can be observed in the simulation waveform. It can be observed that if the random number is two, the fault is injected in bit 1 of input1; if random number is four, the fault is injected in bit 2 of input 1; if random number is eight, the fault is injected in bit 1 of input 2 and if random number is sixteen, the fault is injected in bit 2 of input 2. The corresponding location of fault is indicated by the fault location signal. If enable is zero, then no fault will be injected in the system and output will be the estimated output of adder.

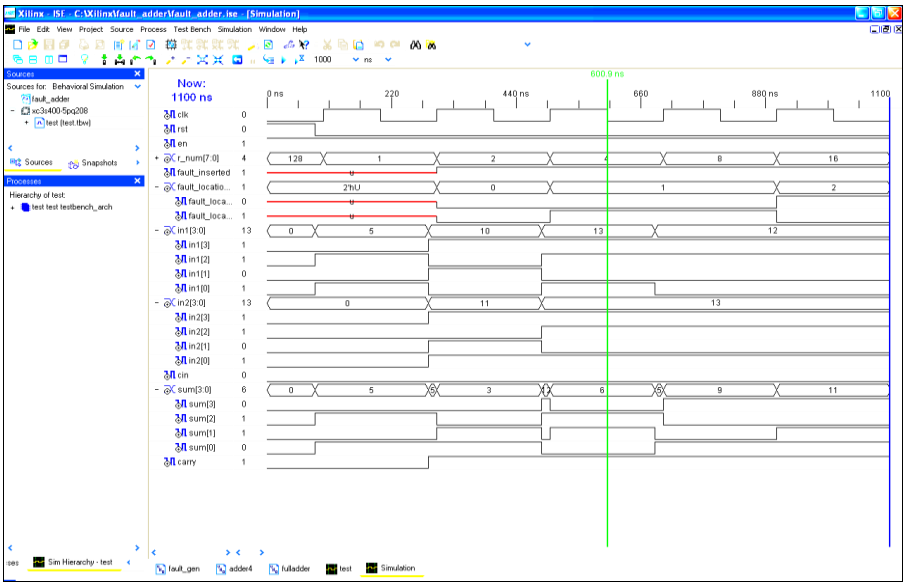


Fig 7. Simulation results for fault injection in 4 bit ripple carry adder

VI. Conclusion

This paper presents a fault injection system that enables designers to insert a fault on any signal within the block of a VHDL code. It allows the injection of transient faults randomly at input of 4 bit ripple carry adder using a random sequence generator. The proposed approach for fault injection is significantly simpler than other currently available approaches. Only a component instantiation and few dummy signals are needed to activate the injection mechanism. The system allows the user to inject faults at desired interval and location which makes it more user friendly. The implementation of the system in Xilinx9.1i is simulated to get the desired results.

References

[1]. Benso A. and Prinetto P, "Fault Injection Techniques and Tools for Embedded Systems Reliability Evaluation," Kluwer Academic Publishers, Holland, 2003

[2]. Lala Parag K , "Transient and Permanent Fault Injection in VHDL Description of Digital Circuits", Circuits and Systems, 2012, 3, 192-199 http://dx.doi.org/10.4236/cs.2012.32026 Published Online April 2012 (http://www.SciRP.org/journal/cs)

[3]. P. K. Lala, "Self-Checking and Fault Tolerant Digital Design," Morgan Kaufmann Publishers, Waltham, 2001.

[4]. Barbosa Raul, Thesis on "Layered Fault Tolerance for Distributed Embedded Systems", ISBN 978-91-7385-209-8
[5]. Sidhardha R., Navm Sastry P. and Rao Dr.D.N, "VHDL Implementation of FPGA Based High Speed Fault Tolerance Tool" International Journal of New Trends in Electronics and Communication (IJNTEC-ISSN: 2347-734) Vol. 2, Issue. 7, Oct. 2014
[6]. Saiz L. J., Gracia J., Baraza J. C., Gil D. and Gil P. J., "Applying Fault Injection to Study the Effects of Inter-mittent Faults," 7th European Dependable Computing Conference, Kaunas, 7-9 May 2008, pp. 67-69.
[7]. Sheng W., Xiao L. and Mao Z. , "An Automated Fault Injection Technique Based on VHDL Syntax Analysis and Stratified Sampling," 4th IEEE International Sympo-sium on Electronic Design, Test and Applications (Delta), Hong Kong, 23-25 January 2008, pp. 587-591.