International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

# FPGA Realization of Fault Diagnostic and Fault Tolerant Scheme for Digital Circuits

**John Kalloor, B. Baskaran**
Department of Electrical and Electronics Engineering, Annamalai University, Chidambaram, Tamil Nadu, India

## ABSTRACT

The paper echoes to formulate a sequence for injecting, detecting and healing the random occurrence of stuck at faults in combinational circuits. The philosophy involves the immaculate use of an LFSR to generate interconnect fault patterns in the passage of primary inputs of the circuit on their way to the destination. The theory extends to enjoy the benefits of a checker circuit to identify and ensure the presence of faults in an attempt to transcend the corrective action. It engages the artefacts of digital logic principles to evolve a fault tolerant status for the methodology and facilitates to arrive at the fault free output in the presence of faults. The exercise augurs to annihilate the common types of stuck at faults to enhance the reliability in the use of such circuits. The Modelsim platform espouses to pronounce the reality in realizing the nuances of the design in the procedure and avail the artefacts of an FPGA to demonstrate its practical significance.

**Keywords:** Fault Injection, Fault Tolerance, FPGA, LFSR, Stuck at faults, VHDL.

## I. INTRODUCTION

The impediments in the functioning of the logic circuits of a digital system manifest as deviations of logic variables from the values specified in design. The uncertainties in the power supply system along with extraneous disturbances in the inputs to the system endeavour to produce unexpected outputs and result in the occurrence of errors [6]. It usually projects itself in the form of a fault in the circuit and change the value of a signal in a circuit from 0 (correct) to 1 (erroneous) or vice versa. The phenomenal features of a fault characterized by its nature, value, extent, and duration augur to subservient the basic operation of the system.

The nature of a fault travels to its logical or nonlogical pattern and diversifies further the theory of unpredictability in its operation. While the logical fault causes the logic value at a point in a circuit to become opposite to the specified value, the nonlogical faults include the malfunction of the clock signal, power failure and similar related occurrences. The value of a logical fault at a point in the circuit indicates whether the fault creates fixed or varying erroneous logical values. The extent of a fault specifies the effect of the fault to be either localized or distributed. A logical fault appears to be a local fault, whereas the malfunction of the clock enters the category of a distributed fault. The duration of a fault refers to whether the fault is permanent or temporary.

Permanent, transient and intermittent faults turn out to be the main sources of errors in digital circuits. Permanent faults occur due to irreversible physical changes, typical examples being shorts and opens in gate level or flow level logic. The transient nature of faults erupts due to external environmental conditions like cosmic rays and electromagnetic interference whereas intermittent faults however arise in light of unstable or marginal hardware and manufacturing residues.

The steady reduction in feature size increases the logic to pin ratio of the integrated chips to reach a new high which in turn makes the signal flow analysis difficult [4,12]. The potential rise in testing time and memory requirements of external testers lead to the emergence of Built In Self Test [BIST] as an integral part of VLSI circuits. BIST is an on-chip testing system that generates test vectors with a purpose to detect faults thereby enables the system to enjoy the benefit of being Totally Self Checking [TSC] [13] through online diagnosis.

The theory of fault injection engraves a crucial role to provide a platform for verifying the functionality of the test circuit and the checker within the system to formulate the fault tolerant strategy. The ability of the fault injection module to access any internal signal at the VHDL level [17] promises substantial controllability and observability of the system. The faults in the testable system connive to be single bit faults in which a bit flips from logic 1 to a 0 and vice-versa and can be of any type in nature.

The process of injecting faults traces over hard and soft options [1, 7, 19] in being able to realize the varied classes at the different levels in a system. Though each claims their own merits, it depends to a large extent on the suitability and degree of precision required in the chosen application. The prominent use of VLSI circuits favours the location of the fault injection block as part of the VHDL subsystem to constitute the fault tolerant prodigy and provide a sense of platform independence. It offers significant advantages that include portability among design packages, easy accessibility to any signal within the VHDL description and little knowledge over simulation procedure.

The philosophy of self checking circuits proceed to detect and indicate the occurrence of faults thereby preventing their propagation throughout the system [18,20]. Thus a checker circuit turns out to be a key component in the design of fault tolerant systems which offers its true services even in turbulent situations. Fault tolerant architecture engages the art and science of building computing systems that continue to operate satisfactorily in the presence of faults [9,10].

The influence of the type of Pseudo Random Pattern Generator [PRPG] for stuck at faults has been discussed with the regard to coverage of faults in [16]. The appropriate choice of a generating polynomial and the use of LFSR seed have been shown using mixed mode BIST for ISCAS benchmarks. A circuit specific design methodology has been outlined for generating on-chip BIST patterns through the use of synthesis techniques in [14]. The experimental results of benchmark netlist have been projected to demonstrate the higher fault coverage and uniqueness of that approach.

The differences between permanent, intermittent and transient faults have been discussed and the results of different error patterns provided in [3]. The study has been directed to establish the role of hardware in the implementation of enhanced error correcting codes. Some techniques have been suggested to reduce the time required for simulation based fault injection campaigns in [15]. Static and dynamic methodologies have been proposed for the analysis of the list of faults to be injected as well as for their removal. A fault model has been analyzed in [21] to derive the optimum faulty period in a digital circuit. The associated distributed functions have been derived and compared with the simulation results of the program representing the model.

A fault tolerant architecture based on human immune system suitable for VLSI based digital systems has been proposed in [11]. The error in the digital circuit has been treated as an antigen by the system and a distributed defence mechanism has been evolved to heal itself from the effect of the error. A new approach for fault tolerant property of digital systems has been suggested in [5]. The fault tolerant time characteristics have been analyzed and the possible way of estimation of time to self heal a transient fault has been considered.

Despite the fact that a wide variety of approaches do exist, still the exquisite needs of a reliable digital system calls for resurgence to explore new trends and arrive at innovative use of a fault tolerant concept for a combinational circuit.

The fundamental theory owes to evolve a fault tolerant strategy to survive stuck at intermittent faults occurring at the interconnect lines of any combinational digital circuit. The procedure reiterates its promise to create a thoroughly reliable system with a view to provide fault free output even on the occurrence of faults. It involves the creation of test patterns to inject faults and lay down measures to identify its occurrence in order to formulate the healing sequence. The travel envisages using the portals of Modelsim platform to realize the functional status of the algorithm and ensure its practical suitability with the help of Xilinx FPGA.

## II. DESIGN METHODOLOGY

The primary idea echoes to utilize the innate ability of an LFSR to generate test patterns with a directive to inject faults at the interconnect levels of the digital circuit. The inherent trait of the LFSR to produce a variety of patterns brings the randomness into the system and allows relating to the real world system dynamics in the sense that the occurrence of the faults is highly unpredictable. The specified number of ones and zeros in the pattern determines the frequency of occurrence of faults.

The scheme requires the presence of a checker circuit to produce equal outputs to ensure the fault free status and augurs to mandate a corrective provision when its outputs differ in their logic state. The defence mechanism treats the digital circuit as a black box and follows the desired outputs based on the relationship that exists between the inputs and outputs and succeeds in its endeavour to provide the true values at the primary output lines, irrespective of the status of the interconnect lines.

The block diagram shown in Figure 1 primarily consists of the Circuit Under Test [CUT] and the fault injection module which act as the sources of input to the checker circuit in the fault detection pursuit. The fault injection module generates fault patterns at the interconnect levels of the system and engages the checker in the process of recognizing the fault. The functional behaviour of 1-out-of-checker shown in Figure 2 essays to extricate its pivotal place in the system. The expressions given below lay down the formation of the checker for accomplishing the task with the inherent self checking ability.

$$z(0) \;<=\; (op(0)\ \boldsymbol{xnor}\ op(2))\ \boldsymbol{or}\ (op(1)\ \boldsymbol{or}\ op(3))$$
$$z(1) \;<=\; (op(0)\ \boldsymbol{or}\ op(2))\ \boldsymbol{or}\ (op(1)\ \boldsymbol{xnor}\ op(3))$$

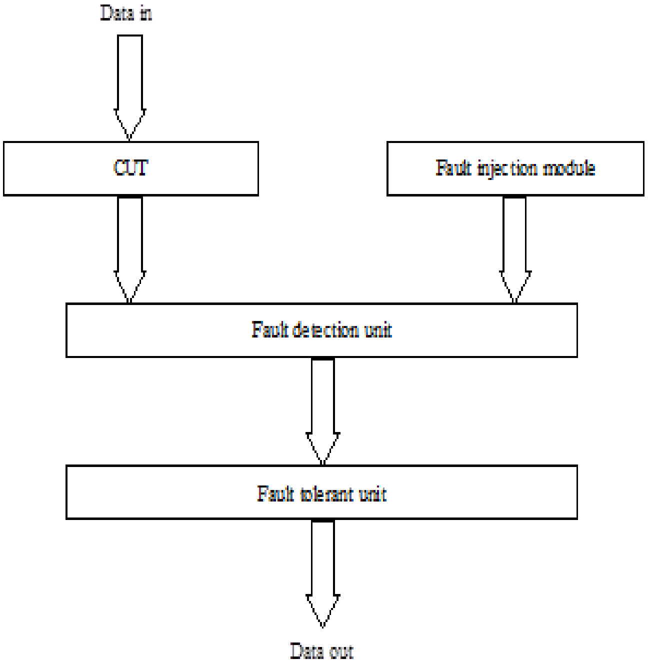

**Figure 1.** Block Diagram of the proposed fault tolerant system

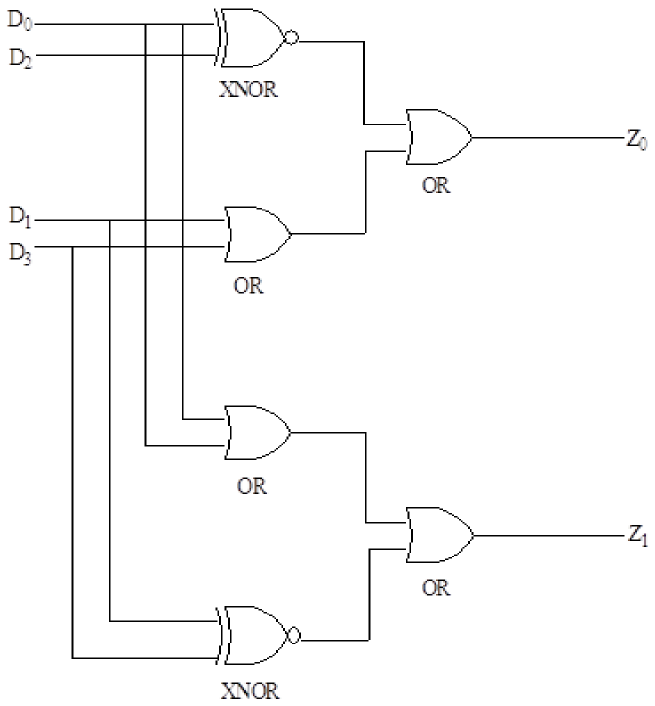

**Figure 2.** 1-out-of-4-checker circuit

The healing strategy forays to retain the primary outputs at the desired logic state even in the presence of error. It comprises of as many ex-or gates as the number of primary outputs of the circuit and the methodology relates to the inherent operating features to track the correct output. The approach extends to follow the desired output and detect the fault when the actual output differs from the desired one. It negates the operation on the faulty interconnect line and traverses to revert it back to its fault free state. The flow diagram seen in Fig. 3 along with the algorithmic steps presented below enumerates the steps that proceed to harmonize the healing part of the mechanism.

**Algorithm**

1. Determine the primary outputs for the given set of primary inputs
2. Check the status of the control signal
3. **If** "control" is not enabled **then**
4. Get the fault free outputs of the system without fault injection
5. **Else**
6. Let the LFSR generate fault patterns
7. **End if**
8. Randomly select certain patterns to be with specified number of ones and zeros
9. **If** the pattern does not contain the specified number of ones and zeros **then**
10. No fault injection
11. **Else**
12. Inject stuck at fault on any of the interconnect line
13. **End if**
14. Monitor the outputs of the checker circuit
15. **If** the outputs are complemented with each other **then**
16. No fault is in the system and get the fault free output
17. **Elsif** the outputs are at the same logic level **then**
18. Fault is detected and heal the system with the built in fault tolerant facility
19. Obtain the true output
20. **End if**

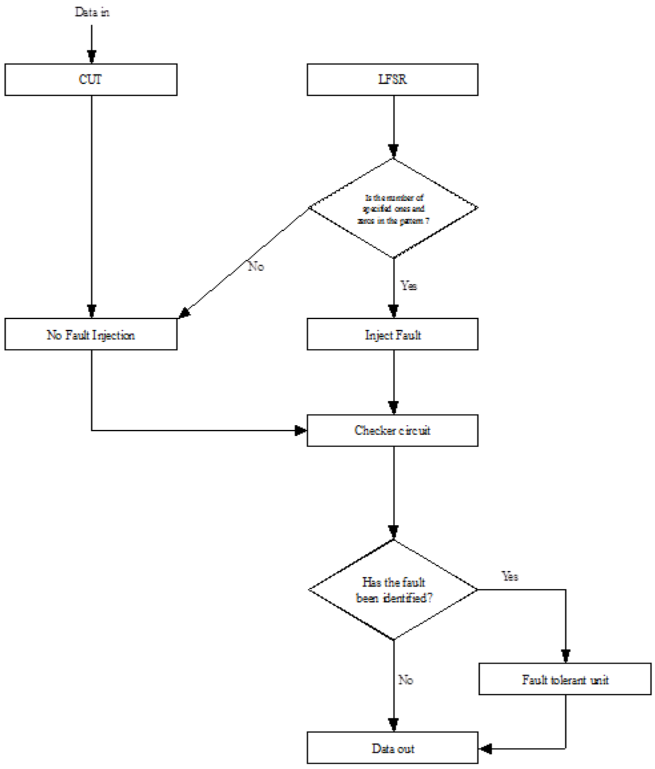

**Figure 3**. Flow diagram of the proposed scheme

## III. SIMULATION RESULTS

The scheme prefers the choice of a 2:4decoder as the CUT and aims to maintain it in its fault free state. The control signal enables the fault injector in the form of 10 bit LFSR to generate 1023 test patterns in an attempt to spread stuck at faults randomly at the interconnect levels of the system. The 1 out of 4 checker circuit senses the occurrence of a fault in the interconnect line based on the idea that only one output of the decoder can be at logic 1 level in its fault free active state. The fault tolerant analogy extends its role to keep the primary outputs of the decoder at the desired logic state even in the presence of error.

The Modelsim based simulation results displayed in Figure 4 exhibits a segment of pseudo random patterns generated by the 10 bit LFSR for the given seed value. The two outputs of the 1-out-of-4 checker keep themselves complemented with each other in the fault free environment, whereas remain at the same logic level to indicate the presence of a fault in the interconnect lines as shown in Figure 5. The identical outputs of the checker archives the inherent fault tolerant circuit associated with the system to initiate the corrective action and derive the true values on the primary output lines. The timing sequence in Figure 6 and 7 elucidate the ability of the proposed strategy to retain the decoder in the fault tolerant state.

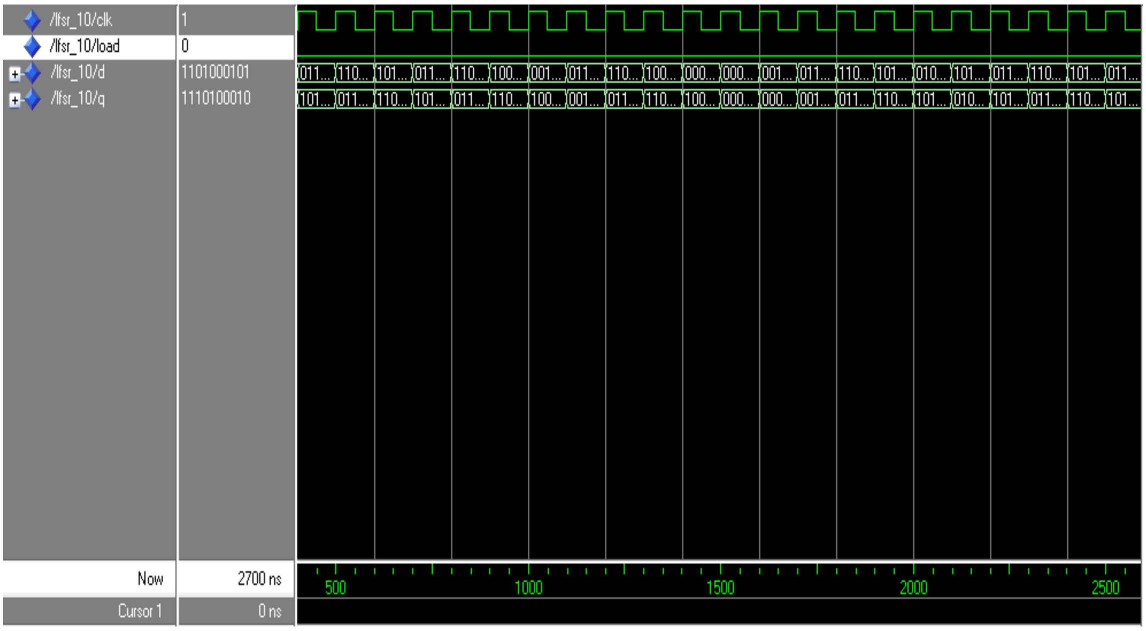

**Figure 4.** Output of 10 bit LFSR

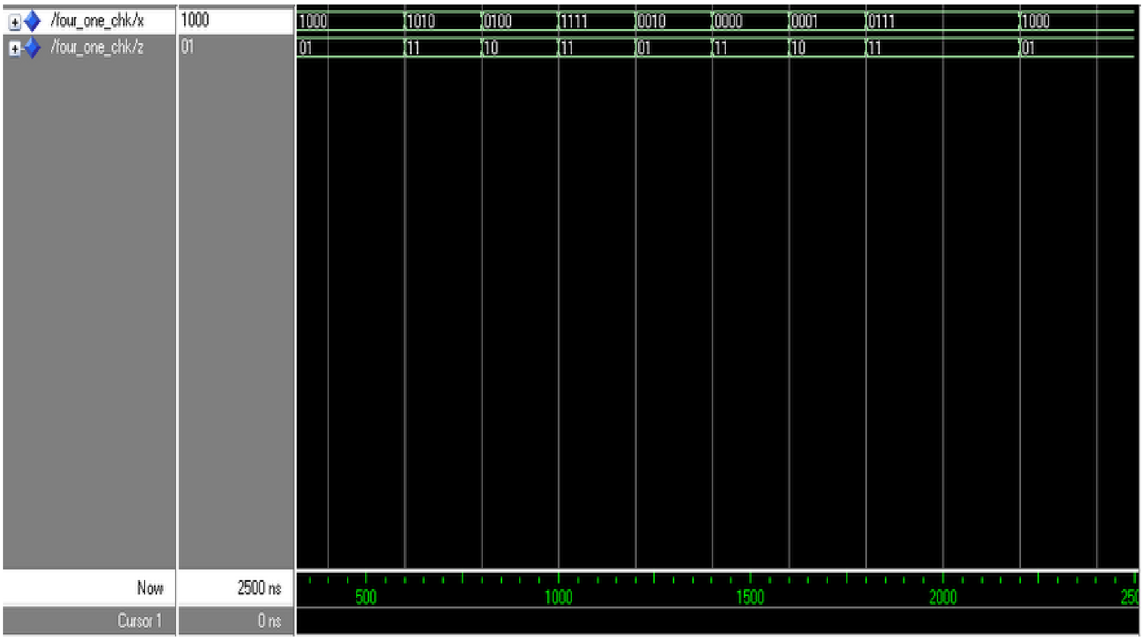

**Figure 5.** 1-out-of-4-checker circuit

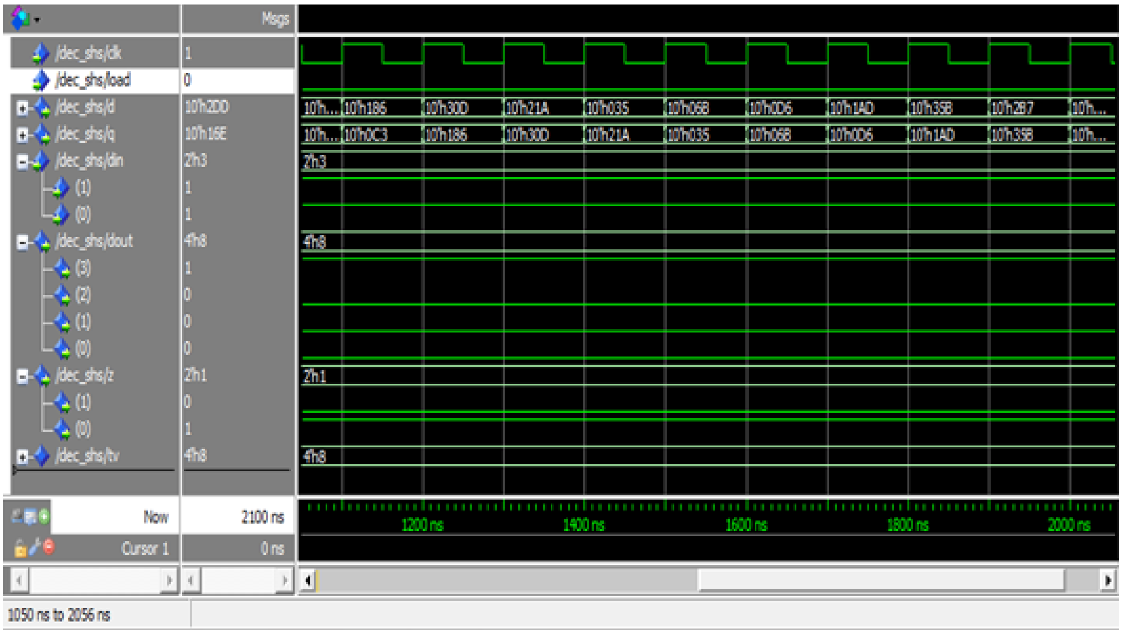

**Figure 6.** Output of the fault tolerant system in the absence of fault

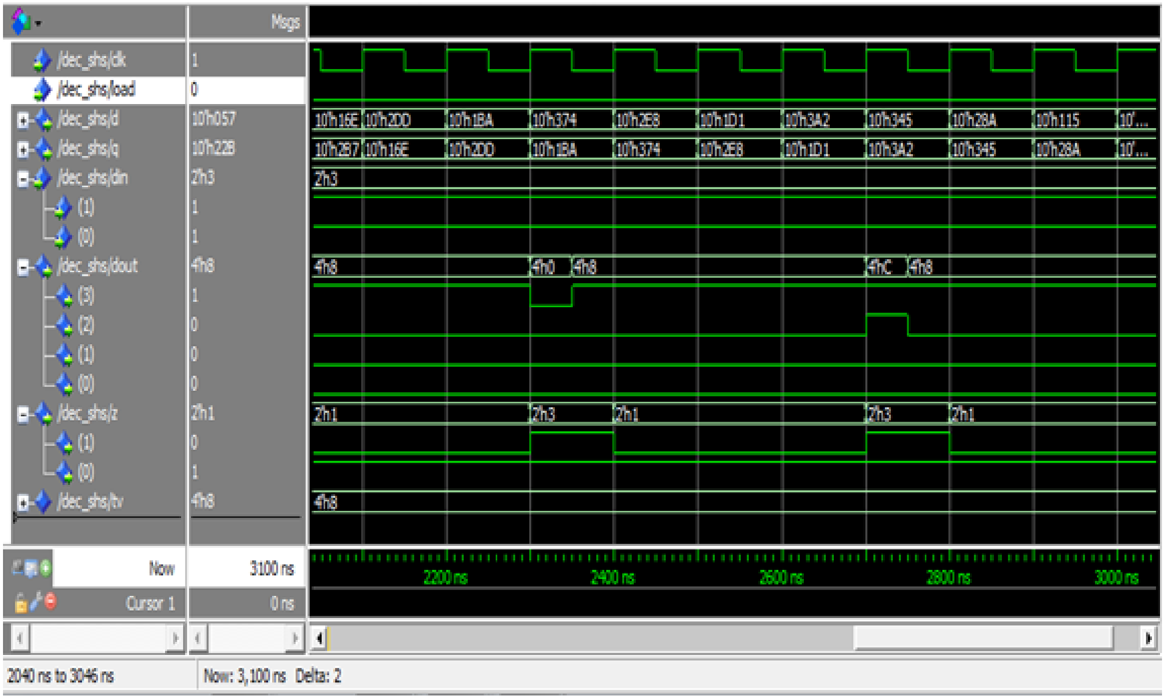

**Figure 7.** Output of the fault tolerant system in the presence of fault

## IV. HARDWARE IMPLEMENTATION

FPGA endeavours to make use of new process technologies and geometries and becomes the preferred choice over today's discrete devices [2, 8]. The inherent ability to re-configure itself to implement logic functions makes it a prominent prospect of digital design implementation. It appears to be tailor made for prototyping on account of its benefits that include low non recurring engineering cost, flexibility to allow design changes even at a later point, short time to market and ability to be reconfigured even at run time. The real time implementation of the proposed fault tolerant system using XC3S500E FPGA serves to validate the simulated performance. The RTL schematic shown in Fig. 8 obtained using Xilinx foundation series ISE 9.2i endorses the applicability of the scheme in real world problems.

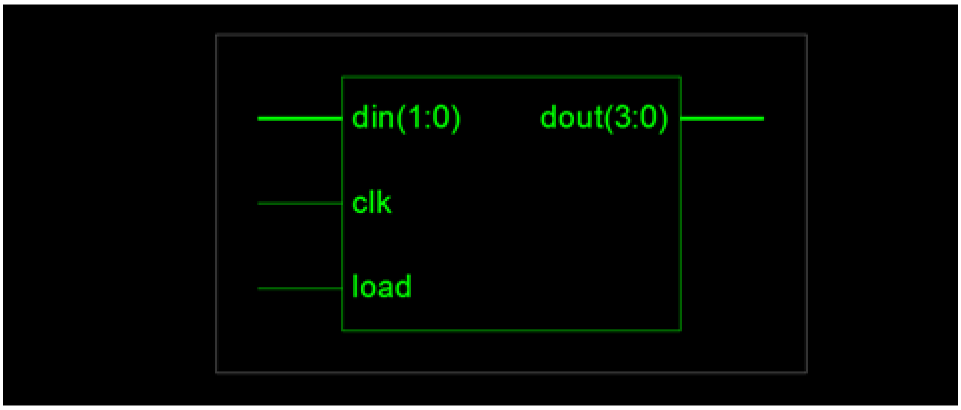

**Figure 8.** RTL schematic of the fault tolerant 2:4 decoder

## V. CONCLUSION

A fault diagnostic and tolerant scheme has been formulated to detect and correct stuck at faults occurring in digital circuits. The intricacies of LFSR have been sought to archive a sense of randomness in the process of generating faults. The reliability in the measurement of digital variables has been epitomized through a checker circuit and endorsed by the fault tolerant nature of the methodology. The Modelsim based simulation results obtained for a decoder circuit to exhibit the simplicity and veracity in the strength of the proposed approach. The features of Xilinx FPGA have been utilized to synthesize the code and establish the applicability of the suggested fault tolerant facility in the real world. The services of the checker together with the ability of the system to suppress the faults enable the system to be both observable and controllable. The ultimate benefit transcends to the drastic reduction in the redundant components needed for its implementation as compared to the traditional TMR based fault tolerant approaches. The relative traits in the fault sensitivities have been castigated to allure the exclusive benefits of the scheme and its inherent ability to correct the interconnect faults in digital circuitry that forms part of a larger periphery acclaims its suitability for practical applications and forges a new dimension for fault tolerance in digital systems.

## VI. REFERENCES

[1]. 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, vol.47, no. 10, pp. 847–867, 2002.

[2]. P. Civera, L. Macchiarulo, M. Rebaudengo, M. S. Reorda and A. Violante, "Exploiting FPGA for Accelerating Fault Injection Experiments," Proceedings of IEEE Online Testing Workshop, Taormina, pp. 9-13, July 2001.

[3]. Cristian Constantinescu, "Intermittent Faults in VLSI Circuits", 2nd Workshop on Silicon Errors in Logic - System Effects (SELSE2), April 2006.

[4]. Debanjan Ghosh, Raj Sharman, H. Raghav Rao, Shambhu Upadhyaya, "Self-healing systems — Survey and Synthesis", Decision Support Systems, vol. 42, no. 4, pp. 2164-2185, 2006.

[5]. S. L. Frenkel and A. V. Pechinkin, "Estimation of Self-Healing Time for Digital Systems under Transient Faults", Informatics and its Applications, vol. 4, no. 3, pp 2-8, 2010.

[6]. Goran Lj. Djordjevec, Mile K. Stojcev, Tatjana R. Stankovic, "Approach to Partially Self-Checking Combinational Circuits Design", Microelectronics journal, vol. 35, no. 12, pp. 945-952, December 2004.

[7]. Haissam Ziade, Rafic Ayoubi, and Raoul Velazco, "A Survey on Fault Injection

Techniques", The International Arab Journal of Information Technology, vol. 1, no. 2, July 2004.

[8]. R.V. KShirsagar, R.M. Patrikar, "A New Design Approach For Tolerating Interconnect Faults in Digital Circuits", Proceedings of SPIT-IEEE colloquium and International conference, India, vol. 2, pp. 117-120.

[9]. R.V. KShirsagar, R.M. Patrikar, "A Novel Fault Tolerant Design and an Algorithm for tolerating Faults in Digital Circuits", 3rd International Design and Test Workshop, Tunisia, pp. 148-153, December 2008.

[10]. P. K. Lala, "Self-Checking and Fault Tolerant Digital Design", Morgan Kaufmann publishers, 2001.

[11]. P.K.Lala and B. K.Kumar, "An Architecture for Self-Healing Digital Systems," Journal of Electronic Testing:Theory and Applications, vol. 19, pp. 523-535,2003.

[12]. P.K. Lala, B. Kiran Kumar, J.P. Parkerson, "On Self-Healing Digital System Design", Microelectronics Journal, vol. 37, no. 4, pp. 353-362, 2006.

[13]. Marc Hunger, Sybille Hellebrand, Alejandro Czutro, Ilia Polian and Bernd Becker, "ATPG-Based Grading of Strong Fault-Secureness", Proceedings of 15th IEEE International On-Line Testing Symposium, Portugal, pp. 269-274, June 2009.

[14]. Mitrajit Chatterjee and Dhiraj K. Pradhan, "A BIST Pattern Generator Design for Near-Perfect Fault Coverage", IEEE Transactions on Computers, vol. 52, no. 12, pp. 1543-1548, December 2003.

[15]. B. Parrotta, M. Rebaudengo, M. Sonza Reorda, M. Violante, "New techniques for accelerating Fault Injection in VHDL descriptions", Proceedings of 6th IEEE Online Testing Workshop, Palma de Mallorca, pp. 61-66, July 2000.

[16]. Petr Fiser, Hana Kubatova, "Pseudorandom Testing – A Study of the Effect of the Generator Type", Acta Polytechnica, vol. 45, no. 2, 2005.

[17]. S. R. Seward, P. K. Lala, "Fault Injection in Digital Logic Circuits at the VHDL Level", Proceedings of the 9th IEEE International Conference on On-Line Testing Symposium (IOLTS'03), July 2003.

[18]. M.K. Stojcev, G.Lj. Djordjevic, T.R.Stankovic, "Implementation of self- checking two-level combinational logic on FPGA and CPLD circuits", Journal of Microelectronics Reliability, vol. 44, no.1, pp. 173-178, 2004.

[19]. Swathi Rudrakshi, Vasujadevi Midasala and S.NagaKishore Bhavanam, "Implementation of FPGA Based Fault Injection Tool (FITO) for Testing Fault Tolerant Designs", IACSIT International Journal of Engineering and Technology, vol.4, no.5, pp. 522-526, October 2012.

[20]. Tatjana R. Stankovic, Mile K. Stojcev, and Goran Lj. Djordjevic, "On VHDL Synthesis Of Self Checking Two Level Combinational Circuits", Electrical Energy, vol.17, no. 1, pp. 69-79, April 2004.

[21]. Trailokya Nath Sasamal and Anand Mohan, "A Specially Designed Transient Faults Injection Technique at the VHDL Level and Modelling", IJRRAS, vol. 9, no. 2, November 2011.