JES
Journal of Engineering Sciences

Vol 13, Issue 05, May /2022
ISSN NO:0377-9254

Identification of Stuck-at-faults of full adder using BIST as testing device

K. Anuradha[1], S.V. Charani Reddy[1], P. Aparna[1], B. Hruthik Reddy[1], M. Akshara[1]
[1]Department of Electronics and Communication Engineering, Malla Reddy Engineering College (A), Secunderabad, Telangana, India

ABSTRACT

Advanced CMOS devices are a critical issue due to the shrinking process of the size of the transistor. The number of transistors packed in an IC increases with the latest technology. Manufacturing defects in the chip happen due to the interconnection of wires. This leads to unexpected outputs. The process of testing confirms that the chip is fault free. This work discussesthe number of test vectors needed to find the faults present in a full adder. An adder circuit is realized in this project. Test vectors are also implemented in LFSR which is fed as inputs to the full adder. Thus generated outputs are tested with the expected outputs to identify the faulty position(s) in the full adder. Spartan3e xc3s500e-5fg320 chip was used to realize the testing.

1. INTRODUCTION

Now a days verification very important for VLSI designs. Manufacturing defects in the chip happen due to the interconnection of wires. This leads to unexpected outputs.The process of testing confirms that the chip is fault free. before manufacturing we need to test the design so we need to apply test patterns for circuit under test. Basically, LCG and LFSR (linear feedback shift register) are used for test pattern generation, in this project we are selected LCG (Linear congruential generator)In computing, a linear-feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state. The most commonly used linear function of single bits is exclusive-or (XOR). Thus, an LFSR is most often a shift register whose input bit is driven bythe XOR of some bits of the overall shift register value. The initial value of the LFSR is called the seed, and because the operation of the register is deterministic, the stream of values produced by the register is completely determined by its current (or previous) state. Likewise, because the register has a finite number of possible states, it must eventually enter a repeating cycle. However, an LFSR with a well-chosen feedback function can produce a sequence of bits that appears random and has a very long cycle.Applications of LFSRs include generating pseudo-random numbers, pseudo-noise sequences, fast digital counters, and whitening sequences. Both hardware and software implementationsof LFSRs are common. The mathematics of a cyclic redundancy check, used to provide a quick check against transmission errors, are closely related to those of an LFSR LFSRs can be implemented in hardware, and this makes them useful in applications that require very fast generation of a pseudo-random sequence, such as direct-sequence spread spectrum radio. LFSRs have also been used for generating an approximation of white noise in various programmable sound generators.The repeating sequence of states of an LFSR allows it to be used as a clock divider or as a counter when a non-binary sequence is acceptable, as is often the case where computer indexor framing locations need to be machine-readable.[7] LFSR counters have simpler feedback logic than natural binary counters or Gray-code counters, and therefore can operate at higher clock rates. However, it is necessary to ensure that the LFSR never enters an all-zeros state, for example by presetting it at start-up to any other state in the sequence. The table of primitive polynomials show how LFSRs can be arranged in Fibonacci or Galois form to give maximal periods. One can obtain any other period by adding to an LFSR that has a longer period some logic that shortens the sequence by skipping some states.LFSRs have long been used as pseudo-random number generators for use in stream ciphers (especially in military cryptography), due to the ease of construction from simple electromechanical or electronic circuits, long periods, and veryuniformly distributed output streams. However, an LFSR is a linear system, leading to fairly easy cryptanalysis. For example, given

a stretch of known plaintext and corresponding cipher text, an attacker can intercept and recover a stretch of LFSR output stream used in the system described, and from that stretch of the output stream can construct an LFSR of minimal sizethat simulates the intended receiver by using the Bredekamp-Massey algorithm. This LFSR can then be fed the intercepted stretch of output stream to recover the remaining plaintext. LFSRs are used in circuit testing for test-pattern generation (for exhaustive testing, pseudo- random testing or pseudo-exhaustive testing) and for signature analysis.Complete LFSR are commonly used as pattern generators for exhaustive testing, since they cover all possible inputs for an n-input circuit. Maximal-length LFSRs and weighted LFSRsare widely used as pseudo-random test-pattern generators for pseudo-random test applications. Signature analysis In built-in self-test (BIST) techniques, storing all the circuit outputs on chip is not possible, but the circuit output can be compressed to form a signature that will later be compared to the golden signature (of the good circuit) to detect faults. Since this compression is lossy, there is always a possibility that a faulty output also generates the same signature as the golden signature and the faults cannot be detected. This condition is called error masking or aliasing. BIST is accomplished with a multiple-input signature register (MISR or MSR), which is a type of LFSR. A standard LFSR has a single XOR or XNOR gate, where the input of the gate is connected to several "taps" and the output is connected to the input of the first flip-flop. A MISR has the same structure, but the input to every flip-flop is fed through an XOR/XNOR gate. For example, a 4-bit MISR has a 4-bit parallel output and a 4-bit parallel input. The input of the first flip-flop is XOR/XNORd with parallel input bit zero and the "taps". Every other flip-flop input is XOR/XNORd with the preceding flip-flop output and the corresponding parallel input bit. Consequently, the next state of the MISR depends on the last several states opposed to just the current state. Therefore, a MISR will always generate the same golden signature given that the input sequence is the same every time.To prevent short repeating sequences (e.g., runs of 0s or 1s) from forming spectral lines that may complicate symbol tracking at the receiver or interfere with other transmissions, the databit sequence is combined with the output of a linear-feedback register before modulation and transmission. This scrambling is removed at the receiver after demodulation. When the LFSR runs at the same bit rate as the transmitted symbol stream, this technique is referred to as scrambling. When the LFSR runs considerably faster than the symbol stream, the LFSR- generated bit sequence is called chipping code. The chipping code is combined with the data using exclusive or before transmitting using binary phase-shift keying or a similar modulation method. The resulting signal has a higher bandwidth than the data, and therefore this is a method of spread-spectrum communication. When used only for the spread-spectrum property, this technique is called direct-sequence spread spectrum; when used to distinguish several signals transmitted in the same channel at the same time and frequency, it is called code divisionmultiple access.

2. RESULTS

RTL SCHEMATIC: - The RTL schematic is abbreviated as the register transfer level it denotes the blue print of the architecture and is used to verify the designed architecture to the ideal architecture that we are in need of development. The hell language is used to convert the description or summery of the architecture to the working summery by use of the coding language i.e., Verilog, VHDL. The RTL schematic even specifies the internal connection blocks for better analyzing. The figure represented below shows the RTL schematic diagram of the designed architecture.

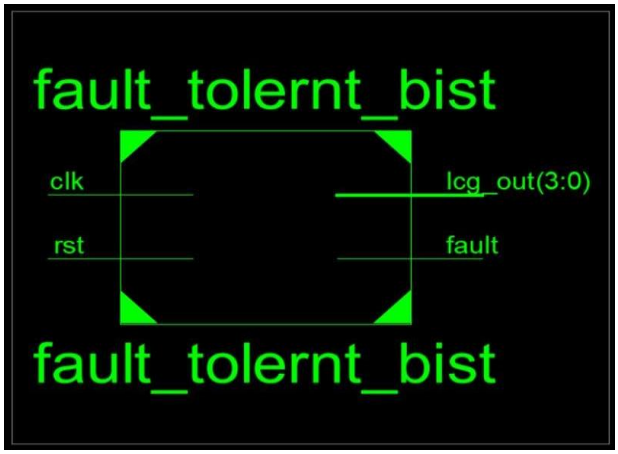

Fig. 2.1 RTL Schematic of Full adder with Fault tolerant best.

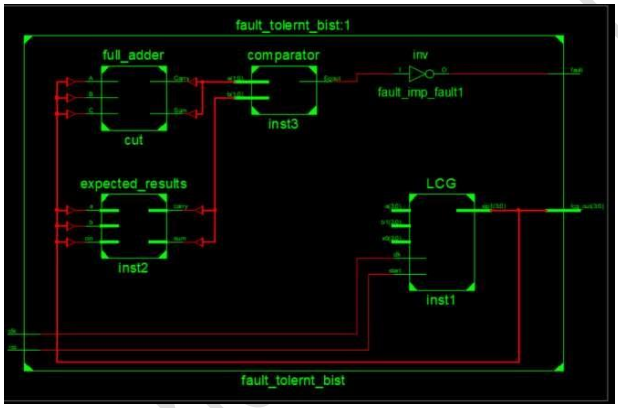

Fig. 2.2 internal structure of RTL Schematic of Full adder with Fault tolerant BIST

TECHNOLOGY SCHEMATIC: - The technology schematic makes the representation of the architecture in the LUT format, where the LUT is consider as the parameter of area that is used in VLSI to estimate the architecture design. the LUT is consider as a square unit the memory allocation of the code is represented in there LUT s in FPGA.

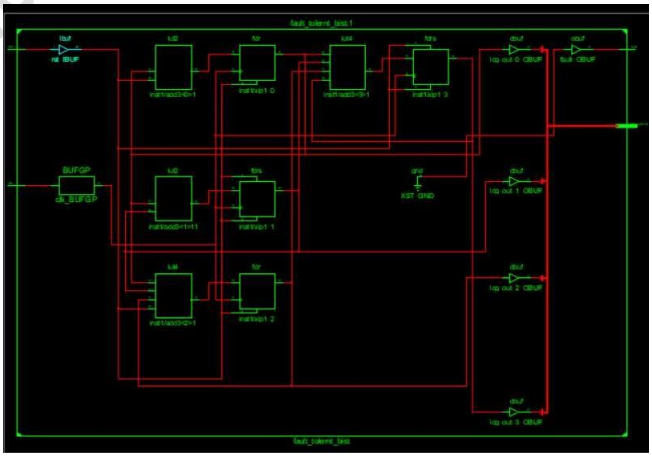

Fig.2.3 View Technology Schematic of Full adder with Fault tolerant bist.

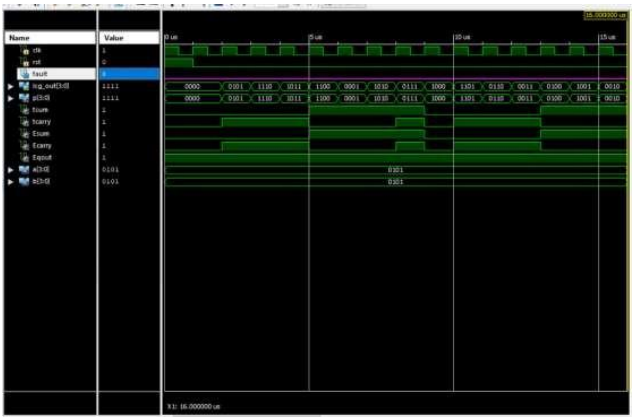

Fig. 2.4 simulated of Full adder with Fault free bist.

Fig. 2.5 simulated of Full adder with Faulty Bist.

The simulation is the process which is termed as the final verification in respect to its working whereas the schematic is the verification of the connections and blocks. The simulationwindow is launched as shifting from implementation to the simulation on the home screen of the tool, and the simulation window confines the output in the form of wave forms output. Here it has the flexibility of providing the different radix number systems.

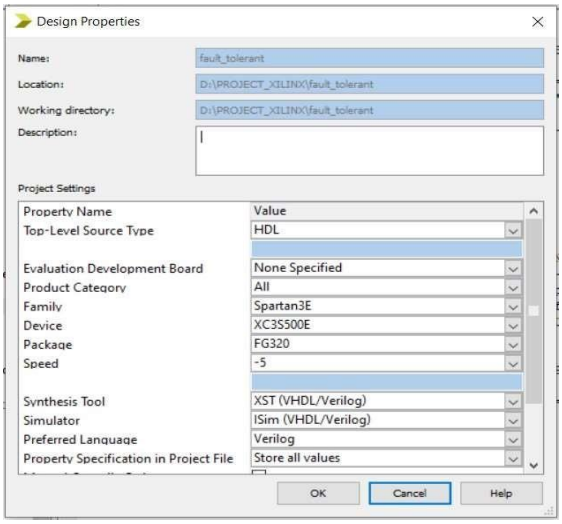

Fig 2.6: family used for synthesis

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	3	4656	0%
Number of Slice Flip Flops	4	9312	0%
Number of 4 input LUTs	5	9312	0%
Number of bonded IOBs	7	232	3%
Number of GCLKs	1	24	4%

Table 2.1 device utilization summery

Minimum period: 2.084ns (Maximum Frequency: 479.835MHz)minimum input arrival time before clock: 2.781ns

Maximum output required time after clock: 5.342nsDelay 5.342ns

Power :0. 0434m.Watt

3. CONCLUSION

The method for reducing test vectors and performing testing at speed testing is discussed in this paper. A circuit can be simulated in the presence of faults. The fault may be stuck-at-0 or stuck at 1 simulates at a functional level. The simulator is used for design verification and verifying its timing analysis. The process to find test vectors for single stuck at fault using path sensitization method and for multiple stuck-at faults using the Boolean difference method through forming a fault table. The number of test vectors can be minimized to find each line of fault in the circuit. Test vector becomes a major issue for the power consumption of the circuit. The methods for reducing test vector are designedand implemented by Hardware Description Language (HDL limitation tools).

REFERENCES

[1] L.Bushnell, and Vishwani D.Agrawal, "Essentials of Electronic Testing for Digital, Memory & Mixed–Signal VLSI Circuits",:Springer, 2002

[2] MironAbramovici, Melvin A.Breuer, and Arthur D.Friedman. " Digital Testing and Testable Design", Piscataway-New Jersey: IEEE Press,1994.

[3] N. Weste and K. Eshragian, “Principles of CMOS VLSI Design”, A Systems Perspective, second ed. Addison-Wesley, 1994.

[4] Vinod Kumar Khera , R.K. Sharma, and A.K. Gupta " A heuristic fault based optimization approach to reduce test vectors count in VLSI testing", Journal of King Saud University – Computer and Information Sciences (2019)
[5] K.L.V.Ramana Kumari, M.Asha Rani and N. Balaji "Design Verification And Test Vector Minimization Using Heuristic Method Of A Ripple Carry Adder", International Journal on Cybernetics & Informatics (IJCI) vol. 5, No. 4, pp. 307-313, August 2016.
[6] Dhanabalan G and Tamil Selvi S, "Design of parallel conversion multichannel analogto digital converter for scan time reduction of programmable logic controller using FPGA", Computer standards and interfaces, vol. 39, pp. 12 – 21, 2015.
[7] Mary D. Pulukuri and Charles E. Stroud " On Built-In Self-Test for Adders", J Electron Test (2009) 25:343–346 DOI 10.1007/s10836- 009-5114-6.
[8] Ashok Kumar, Rahul Raj Choudhary, and Pooja Bhardwaj, "Universal Pattern Set for Arithmetic Circuits", International Journal of Computer Applications (0975 – 8887) vol. 40– No.15, February 2012.
[9] S Jayanthy, and MC Bhuvaneswari, "Delay Fault Testing of VLSI Circuits", – Springer 2019, Test Generation of Crosstalk Delay Faults, chapter-2, pp 15-35.
[10] Matthias Sauer, Jie Jiang, Sven Reimer, Kohei Miyase, Xiaoqing Wen, Bernd Becker and Ilia Polian, "On Optimal Power-aware Path Sensitization", IEEE 25th Asian Test Symposium, 2016, pp. 179-184.
[11] Nelson, Nagle, Carroll, and Irwin",Digital Logic Circuit Analysis & Design" Prentice- Hall,1995, Chapter 12, pp. 739 to 757.
[12] Mojtaba Valinataj, Abbas Mohammadnezhad, and Jari Nurmi, "A Low-Cost High- Speed Self-Checking Carry Select Adder with Multiple Fault Detection",
ELSEVIER Microelectronics Journal, Volume 81,November 2018, pp. 16-27
[13] Henley Court Pullman,"Genesys™ FPGA Board Reference Manual", Revised April 11, 2016.This manual applies to the Genesys rev.