International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Design of Inter-Integrated Circuit with BIST Method

Priyanka Lade[1], Sanjay Tembhurne[2]

[1]Student M-Tech GHRAET, RTMNU, Nagpur, India

[2]Assistant Professor, GHRAET, RTMNU, Nagpur India

Abstract: ***The I2C (Inter-Integrated Circuit) Bus is a two-wire, low to medium speed, communication bus (a path for electronic signals) developed by Philips Semiconductors in the early 1980s. I2C was created to reduce the manufacturing costs of electronic products. The Inter-Integrated circuit protocol is used to attach two devices for communicating with each other in fast way excluding data losses. The I2C Bus is a time-proven, industry standard, communication protocol used in a wide variety of electronic products. I2C is found in products we use every day, like cellular and conventional telephones, computers, and ATMs (automatic teller machines). It slow cost and powerful features make I2C ideal for low to medium speed chip-to-chip communications. With the fast development of Integrated circuits (ICs) technology, the complication of the circuits has also increase day by day. To save time and money the circuit requires self -testability in hardware to palliate the product failure. BIST is system develop to performing functional testing at different speed. Built-in self-test (BIST) is a design technique that allows a circuit to test itself it is a set of structured test techniques for combinational and sequential logic, memories, multipliers and other embedded logic blocks. Built-in-self-test (BIST) is such a technique which can meet the necessity of self-testability with an effective solution over pricy circuit testing system. This synopsis proposed designing of Inter-Integrated Circuit (I2C) protocol with self-testing ability. In order to attain compact, stable and reliable data transmission, the I2C is designed with self-testability is needed.***

Keyword: Inter-integrated Circuit, Built-in Self-test Architecture, Verilog HDL.

1. Introduction

The bus has two lines that carry its signals-one has for clock and one is for bidirectional bus. There is standard protocol for I2C bus. I2C interface with different Devices processors, EEPROMs, sensors, real-time clocks.I2C used as a control interface I2C devices can also have separate data interface (digital TV tuners, video decoders, audio processors ...)
I2C operate at three speeds-

- Slow (under 100 Kbps)
- Fast (400 Kbps)
- High-speed (3.4 Mbps) – I2C v.2.0

I2C Bus length is vary from inside the equipment, <1m, maximum few meters. I2C was created to reduce the manufacturing costs of electronic products.I2C is a two wire, bidirectional serial bus that provides effective data communication between two devices. The physical I2C bus consists of just two wires, called SCL and SDA. SCL is the clock line; it is used to synchronize all data transfers over the I2C bus. SDA is the data line; the SCL and SDA lines are connected to all devices on the I2C bus. Fig.1 shows the Format of I2C bus protocol. [3].At the initialization of I2Cprotocol shows Start condition. It also shows the 7 bit slave address send from master and one1bit for reading for writing operation .Next bit shows acknowledgement bit after receiving the data by receiver.1byte data is of word address and 1 byte shows the address of data. At the completion of data its gives stop condition to stop theprocess of data transmission.

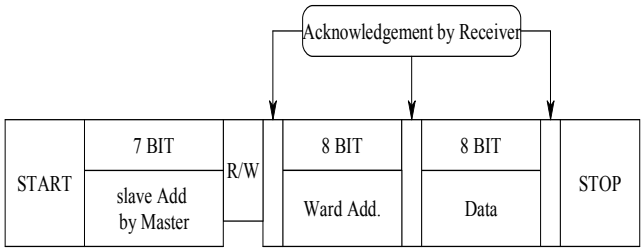


Figure 1: Format of I2C Protocol Bus

I2C-bus compatible ICs don't only assist designers; they also give a wide range of benefits to equipment manufacturers because:

- The simple 2-wire serial I2C-bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result — smaller and less expensive PCBs
- The completely integrated I2C-bus protocol eliminates the need for address decoders and other 'glue logic'
- The multi-master capability of the I2C-bus allows rapid testing and alignment of end-user equipment via external connections to an assembly-line computer
- The availability of I2C-bus compatible ICs in SO (small outline), VSO (very small outline) as well as DIL packages reduces space requirements even more.

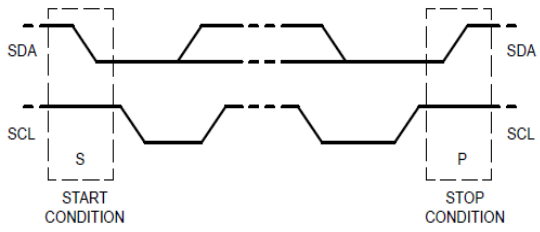


Figure 2: START and STOP Condition

1.1 Data Validity

The data on the SDA line must be stable during the HIGH period of the clock. The HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW.

1.2 START and STOP conditions

Within the procedure of the I2C-bus, unique situations arise which are defined as START and STOP conditions (see Figure 6). A HIGH to LOW transition on the SDA line while SCL is HIGH is one such unique case. This situation indicates a START condition. A LOW to HIGH transition on the SDA line while SCL is HIGH a STOP condition. START and STOP conditions are always s generated by the master. The bus is considered to be busy after the START condition. The bus is considered to be free again a certain time after the STOP condition.

1.3 Byte Format

Every byte put on the SDA line must be 8-bits long. The number of bytes that can be transmitted per transfer is unrestricted. Each byte has to be followed by an acknowledge bit. Data is transferred with the most significant bit (MSB) first . If a receiver can't receive another complete byte of data until it has performed some other function, for example servicing an internal interrupt, it can hold the clock line SCL LOW to force the transmitter into a wait state. Data transfer then continues when the receiver is ready for another byte of data and releases clock line SCL. In some cases, it's permitted to use a different format from the I2C-bus format (for CBUS compatible devices for example). A message which starts with such an address can be terminated by generation of a STOP condition, even during the transmission of a byte. In this case, no acknowledge is generate. Acknowledge Data transfer with acknowledge is obligatory. The acknowledge-related clock pulse is generated by the master. The releases the SDA line (HIGH) during the acknowledge clock pulse.

2. Previous Work

Built-In Self-Test (BIST), as the name suggests is a technique in which the circuit is capable of testing itself. It reduces testing and maintenance cost and also reduces cost of automatic test pattern generation (ATPG).

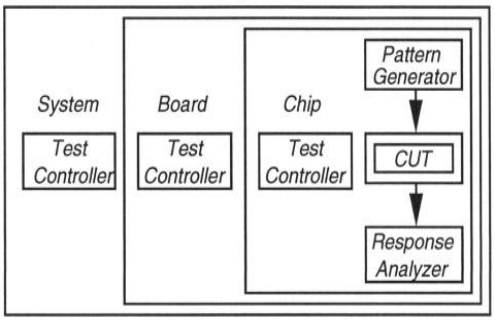


Figure 3: BIST hierarchy

Figure shows the BIST system hierarchy and all three levels of packaging mentioned earlier. The system has several PCBs, each of which, in turn, has multiple chips. The system *Test Controller* can activate self-test simultaneously on all PCBs. Each Test Controller on each PCB can activate self-test on all chips on the PCB. The Test Controller on a chip executes self-test for that chip, and then transmits the result to the PCB Test Controller, which accumulates test results from all chipson the board and sends the results to the system Test Controller. The system Test Controller uses all of these results to isolate faulty chips and boards. System diagnosis is effective only if the self-test procedures are thorough. For BIST, fault coverage is a major issue. Other issues are chip area overhead, its impact on chip yield, the cost of the additional chip pins required for test, the performance Penalty in terms of added circuit delay, and extra power requirements. For BIST, the test engineer frequently, but not always, modifies the chip logic to make alllatches and flip-flops controllable, perhaps by using the scan technique.

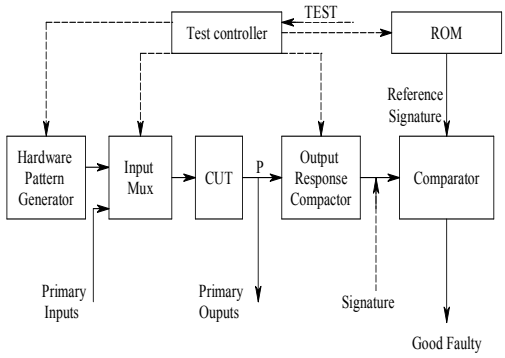


Figure 4: BIST Architecture

Figure shows typical BIST hardware in more detail. Note that the wires from PIs to the Input MUX and the wires from circuit outputs P to primary outputs(POs) cannot be tested by BIST. These wires, instead, require another testing method, such as an external ATE or JTAG Boundary Scan hardware. Figure also shows how a comparator compares the signature produced by the data compacter with a reference signature stored in a ROM during BIST. This comparator and ROM hardware can frequently be implemented with a single logic gate with 32 or fewer inputs. This is acceptable only when the comparison can occur at extremely low rates of circuit operation, since this logic gate is exceedingly slow. Figure shows typical BIST process. The wires from PIs to the Input MUX and the wires from circuit outputs P to primary outputs (POs) cannot be tested by BIST. These wires, instead, require another testing method, such as an external ATE or JTAG Boundary Scan hardware. Figure also shows how a comparator compares the signature produced by the data compacter with a reference signature stored in a ROM during BIST. The wires from PIs to the Input MUX and the wires from circuit outputs P to primary outputs (POs) cannot be tested by BIST. Figure also shows how a comparator compares the signature produced by the data compacter with a reference signature stored in a ROM during BIST. This comparator and ROM hardware can frequently be implemented with a single logic gate with 32 or fewer inputs.

2.1 BIST Pattern Generation:

The following hardware pattern generation approaches have been used.

2.1.1 ROM. One method is to store a good test-pattern set (from an ATPG program) in a ROM on the chip, but this is prohibitively expensive in chip area, and will not be discussed further.

2.2.2 LFSR. Another method is to use a linear feedback shift register (LFSR) to generate pseudo-random tests. This frequently requires a sequence of 1 million or more tests to obtain high fault coverage's, but the method uses very little hardware and is currently the preferred BIST pattern generation method.

2.2.3 Binary Counters. A binary counter can generate an exhaustive test sequence, but this can use too much test time if the number of inputs is huge. For example, with 64 inputs and the test-pattern generator clocked at 100 MHz, this takes 51,240,955.8 hours of test time to generate all 2^{64} patterns, which is impractical. Therefore, this type of pattern generator must be partitioned. Also, the binary counter requires more hardware than the typical LFSR pattern generator.

2.2.4 Modified Counters. Modified counters have also been successful as test-pattern generators, but they also require long test sequences.

2.2.5 LFSR and ROM. One of the most effective approaches is to use an LFSR as the primary test mode, and then generate test-patterns with an ATPG program for the faults that are missed by the LFSR sequence. These few additional test-patterns can either be stored in a small ROM on the chip for a second test epoch, they can be embedded in the output of the LFSR, or they can be embedded in a scan chain in order to augment the stuck-fault coverage to100%.

2.2.6 Cellular Automaton. In this approach, each pattern generator cell has a few logic gates, a flip-flop, and connections only to neighboring gates. The cell is replicated to produce the cellular automaton.

2.2 Linear Feedback Shift Register

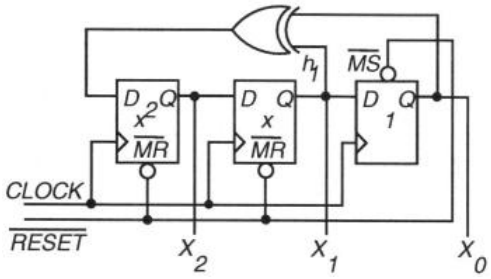


Figure 5: Example external-XOR standard LFSR

Figure shows an example external XOR standard LFSR with characteristic polynomial f(x) =1+x+x^3 The characteristic polynomial f(x) of the external-XOR LFSR is read from right to left, so since the rightmost flip-flop is always tapped, this polynomial has 1(x^0)a and since the middle flip-flop is tapped, it also has an x term, since However, there is no x^2 term, because the leftmost flip-flop is not tapped. There is always an x^n term in the characteristic polynomial for an n-bit standard LFSR, so we include the x^3 term. So, f(x)=1+x+x^3 We see the pattern repeating after the first seven patterns are generated. The student should convince him/herself of the correctness of the pattern sequence by simulating the circuit logic. During BIST, as will be explained below, it is essential that the circuit be excitedonce and only once with a particular pattern. This is because a given pattern causes an error vector to appear at the faulty circuit outputs, which are read by the BISTResponse compacter, and repeating the pattern later causes the same error vector to appear again. Since the response compacter is an XORing system as well, the two erroneous responses from that error vector will cancel and leave the BIST system with only the good-machine response. This will cause the testing hardware to accepta faulty circuit as a good circuit. Therefore, we must avoid repeating any of theLFSR patterns more than once, and we must not initialize the LFSR to all zeros,or it will hang indefinitely in the all zero state

3. Proposed Work

In previous work I2C with BIST can be design .In that method BIST can be used for testing itself. In BIST scheme there is different blocks such as Test controller, pattern generator, CUT and the Response analyzer. In that BIST structure pattern generation will be found by Linear Feedback Shift Register (LFSR) to get the random pattern formation. But the proposed work is that working on that pattern generation by using the Cellular Automata

3.1 Cellular Automata

Mathematical object defined as:n-dimensional homogeneous and infinite cellular space, consisting of cells of equal size (2-D CA = torus, not a plane); Cells in one of a discrete number of states; Cells change state as the result of a transition rule; Transition rule is defined in terms of the states of cells that are part of a neighbourhood; Time progresses in discrete steps. All cells change state simultaneously.

A one-dimensional cellular automaton (CA) consists of two things: a row of "cells" and a set of "rules". Each of the cells can be in one of several "states". The number of possible states depends on the automaton. Think of the states as colors. In a two-state automaton, each of the cells can be either black or white. Over time, the cells can change from state to state. The cellular automaton's rules determine how the states change. It works like this: When the time comes for the cells to change state, each cell looks around and gathers information on its neighbors' states. (Exactly which cells are considered "neighbors" is also something that depends on the particular CA.) Based on its own state, its neighbors' states, and the rules of the CA, the cell decides what its new state should be. All the cells change state at the same time.

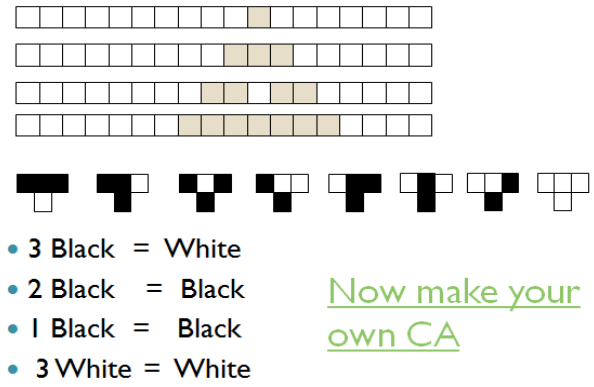
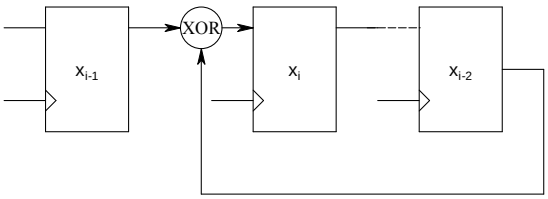
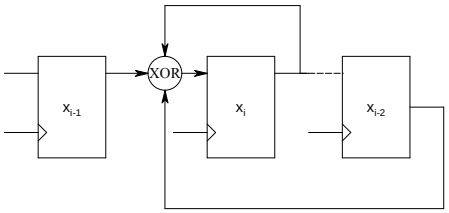


Figure 6: One dimensional CA



Rule 90
$x_i(t+1)=x_{i-1}(t)$ xor $x_{i+1}(t)$



Rule 150
$x_i(t+1)=x_{i-1}(t)$ xor $x_i(t)$ xor $x_{i+1}(t)$

Figure 7: Cellular Automata Implementation

Cellular automata(CA) are excellent for pattern generation, because they have a better randomness distribution than LFSRs.A cellular automaton is a cell collection with regular connections each cell can only connect to its local neighbors. The connections are expressed as rules, which determine theNext state based on the state of the cell's neighbors. If cell *c* can talk only with its neighbors,c-1 and c+1 then the following rule, called *rule 90*, can be established based on the following state transition table:

$x_{c-1}(t)x_c(t)x_{c+1}(t)$	111	110	101	100	011	010	001	000
$x_c(t+1)$	0	1	0	1	1	0	1	0

$$2^6+2^4+2^3+2^1=90$$

The term *rule 90* comes from the decimal equivalent of the binary code for the next state of cell *c*. In this case, : Xc(t+1)= Xc-1(t) xor Xc+1(t) Another relation, *rule 150*, is implemented as asXc(t+1) = Xc-1(t) xorXc(t) xor Xc+1(t). Figure 5shows a hybrid cellular automaton alternately using rules 90 and 150 in its cells. Serra *et al.*demonstrated an isomorphism between a one-dimensional linear hybrid cellular automaton and the LFSR having the same irreducible characteristic polynomial. However, state sequencing may still differ between the CA and the LFSR. CAs has no shift-induced bit value correlation, whereas LFSRs do.

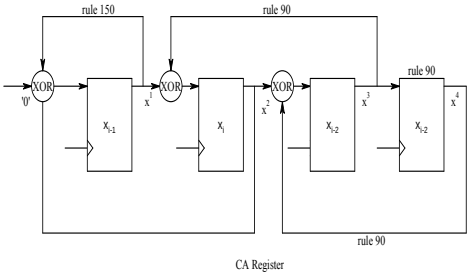


Figure 8: Cellular Automata Implementation

A CA register is also known as Linear Hybrid Cellular Automata (LHCA) or Linear Cellular Automata Register (LCAR) .Similar to LFSRs there are some combination of rules which produce exhaustive pseudo-random patterns. LHCAs are capable of generating patterns which are more random in nature as compared to an LFSR [3]. But LHCAs have larger nodes and require many more XOR gates as compared to LFSRs. Large number of XOR gates results in higher area over-head of LHCAs. In cases where LHCA is used as a TPG, it is desirable to use a CA for Signature analysis instead of a LFSR.A signature analyzer has the same aliasing properties as linear feedback shift registers.

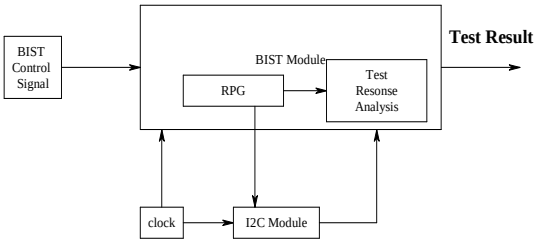


Figure 9: Block Diagram of I2C with BIST

Figure 9. shows the block dig of I2C with BIST.To control the BIST module use the Control signal. There are four sub blocks are added in BIST module. They are three random pattern generators and Test Response analyzer.

1) Random Pattern Generators (RPG): To verification of devices like I2C use Random Pattern Generator (RPG) generates random patterns .In BIST structure RPG most important part for the verification of the circuits. Many methods have been proposed for the BIST equipment design. To produce bytes to test the circuit the method of a random pattern generator (RPG) is used. This RPG consists CA is used to generate the slave address. CA produces word address.CA gives the data. The generated bytes are used directly in theI2C to obtain better fault coverage.2) Test Response analyzer: This is just like comparator which is used to compare the received and transmitted bit pattern. And then it gives the value of error.

4. System Synthesize and Implementation

A. Circuit Schematic

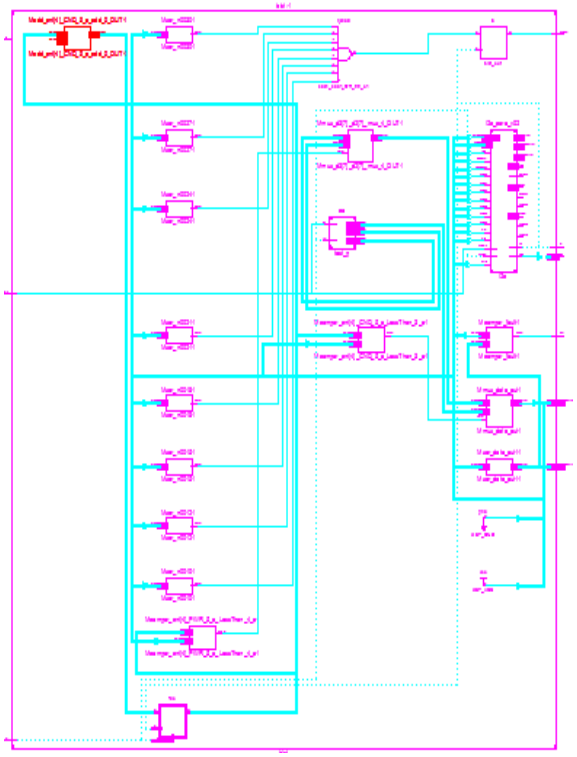


Figure 10: Top level schematics of I2C with BIST capability

Figure 10 shows that schematic of I2C with BIST. There are different blocks in this schematic I2C, BIST, CA, and Comparator.

B. Simulation Results

The timing diagram achieved from Test bencher showed that the received data in the receiver. Then the design is tested in the Xilinx where it also gave the correct output.

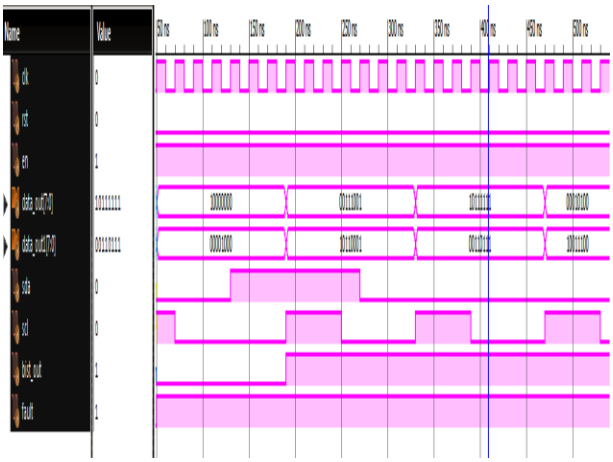


Figure11: I2C with BIST with Fault

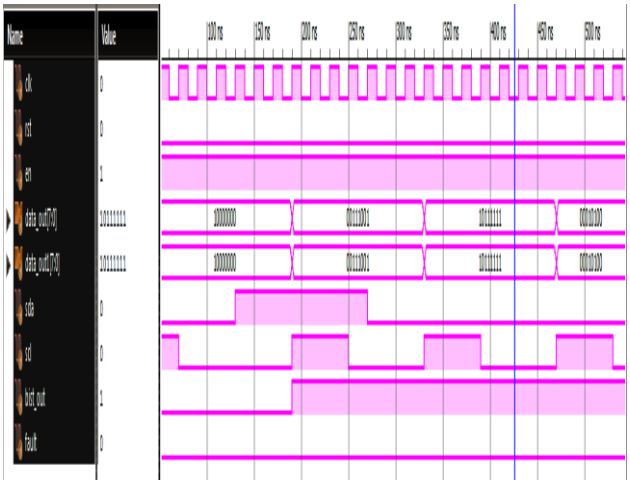


Figure 12: I2C with BIST without Fault

The Device utilization summary and timing summary are given in Table IV and Table V respectively. From the table it is seen that only 25% if the flip flops are used. In case of inputoutput buffers it is 21%. The Number of slice LUTs is only5% and the Number of fully used LUTs FF pair So that, there are many more featherscan be added with the proposed architecture. In the timingsummary, it is seen that the minimum period is 5.03ns whichis much lesser than conventional previous paper.

Table 1: Device Utilization Summary

Name	Used Blocks	Percentages (%)
Number of Slice Flip Flops	39out of 384	0%
Number of slice LUTs	126 out of 2400	5%
Number of fully used LUTs FF pair	34 out of 131	25%
Number of bonded IOBs	22 out of 102	21%
Number of GCLKs	2 out of 16	12%

Table 2: Timing Summary

Parameters	Seconds
Minimum period	5.03ns
Minimum input arrival time before clock	2.972ns
Maximum output required time after	7.302ns

5. Conclusion

In this paper, a Xilinx based implementation of I2C with BIST capability is presented. Here all the modules are designed and simulated with Verilog HDL. Then the system isdownloaded in the Xilinx Spartan-6 FPGA (XC6SLX4). The simple 2-wire serial I2C-bus minimizes interconnections so ICs have fewer pins and there are not so many PCB tracks; result — smaller and less expensive PCBs. Inter integrated circuit is more speedy stable and much more flexible. Built-In Self-Test (BIST), as the name suggests is a technique in which the circuit is capable of testing itself .I2C with BIST capability can gives more proper result at the simulation .It also save valuable time and cost of testing circuits significantly.

References

[1] ShumitSaha,Md.AshikurRahman,AmitThakur,"Designa nd implementation of a BIST Embedded Inter

IntegratedCircuit Bus Protocol over FPGA" international conferenceon electrical info. And communication tech.,2013.
[2] ShumitSaha, Md. AshikurRahman, AmitThakur ,"Designand Implementation of a BIST Embedded High Speed RS-422 Utilized UART over FPGA ",2012 Department ofElectronics &Communication Engineering IEEE – 31661
[3] LixinGao, Yongliang Zhang, Jinhong Zhao"BIST usingCellular Automata as Test PatternGenerator and ResponseCompaction," School of Information EngineeringGuangdong JidianPolytechnicGuangzhou, China978-1-4577-1415-3/2012 IEEE
[4] "ESSENTIALS OF ELECTRONIC TESTINGFORDIGITAL, MEMORY ANDMIXED-SIGNAL VLSI CIRCUITS"Michael L. BushnellRutgers University,Vishwani D. AgrawalBell Labs,Lucent Technologies
[5] M. Serra, K. Cattell, S. Zhang, J.C. Muzio, D.M. Miller"One-Dimensional Linear Hybrid Cellular Automata:Their Synthesis, Properties and Applications to DigitalCircuits Testing" Dept. of Computer Science, January27, 2009
[6] Manish PatelNehalParmar,VishwasChaudhari " Designand Implementation OF Logic-BIST Architecture for I2CSlave VLSI ASIC Design Using Verilog" journal ofinformation, knowledge and research in electronics andCommunication engineering, Issn: 0975 – 6779| Nov 12 toOct 13 volume – 02, issue - 02 page 552
[7] I2C Bus Specification, Version 2.1,PhilipsSemiconductors,2000.
[8] NazaninMoarefi, Ali Yarahmadi "The Implementation of Cellular Automata with Non-identical Rule on Serial Base" The Journal of Mathematics and Computer Science Vol. 4 No.2 (2012)
[9] J. Bhasker, Verilog® HDL synthesis: a practical primer, Star Galaxy Publishing, 1998.
[10] M.Y.I. Idris, M. Yaacob, "A VHDL implementation of BIST techniquein UART design," in Proc. Conference on Convergent Technologies forthe Asia-Pacific Region (TENCON), Oct. 2003, pp.1450-1454.
[11] S. Jamuna and Dr. V.K. Agrawal,"Implementation of BIST structureusing VHDL for VLSI circuits," International Journal of EngineeringScience and Technology, vol. 3, no. 6, pp. 5041-5048, June 2011.
[12] Idris, M.Y.I.,Yaacob, M.; , "A VHDL implementation of BISTtechnique in UART design," TENCON 2003. Conference onConvergent Technologies for Asia-Pacific Region , vol.4, no., pp.1450- 1454 Vol.4, 15-17 Oct. 2003.Chun-zhi, He; Yin-shui, Xia; Lun-yao, Wang.
[13] Xin-She Yang andY. YoungCellular Automata,PDEs, and Pattern FormationCHAPMAN: "C4754_C018" — 2005/5/5
[14] SachinDhingra,"Comparison of LFSR and CA for BIST" Student memberIEEE Dept. of Electrical and Computer Engineering 200 Broun Hall, Auburn University.