Built in Self Test Architecture using Concurrent Approach

M. Dodiya Chandni and V. Ravi*
School of Electronics Engineering, VIT University, Chennai-600127, Tamil Nadu, India; dchandni.mahendrasinh2014@vit.ac.in, ravi.v@vit.ac.in

Abstract

Background/objectives: Built in Self Test Architectures are used for the online or offline testing of the digital circuits and can be operated both in normal as well as test mode. So the objective is to test the circuit under test in online mode with less concurrent test latency and less area overhead. Methods/Statistical Analysis: In the case of normal mode the time required for testing becomes undesirable parameter so here we prefer offline testing method with concurrent approach which is also monitoring the window at the input by applying input vectors considering circuit under test as most important part of the processor which is arithmetic logic unit. Findings: The particular locations of the input vectors are stored in the latches which worked as the memory elements and this proposed scheme becomes more efficient by using cellular automata as test pattern generation and response analyzer using rule 90. Application/Improvement: The proposed scheme is comparable with the same architecture, considering TPG as LFSR (Linear Feedback Shift Register) and counter.

Keywords: Arithmetic Logic Unit, Built in Self Test, Cellular Automata, Concurrent Test Latency, LFSR, Memory Elements, TPG, Windowing

1. Introduction

Built in Self Test testing scheme is used to test the circuit itself and it is a most desirable technique nowadays as it reduces the expensive test equipment and also reduce the area overhead and testing time. The BIST scheme can be categorized between online mode and offline mode. In online mode the circuit under test is kept in working condition and the BIST scheme performs the testing as well. While in offline mode the circuit under test is kept disable so it indicates that offline testing may take more testing time so here we prefer offline mode testing. Here there are two modes, called normal and test mode. During the operation of the test mode the inputs are applied to the CUT via test pattern generator and the outputs are captured by the response verifier so the normal operation of the CUT can be affected by the stated process and the performance of the system is disgraced while during normal mode the CUT also works simultaneously with the BIST operation.

Built in Self Test architecture with Concurrent approach proposes a scheme of observing or monitoring a set of vectors coming from the test pattern generation which are applied to the CUT inputs during normal mode operation. If the incoming test vectors are matched with the already active test set, then a vector has performed hit operation and the RVE signal has

*Author for correspondence
Figure 1. Basic BIST block operation.

Figure 2. Proposed BIST scheme.
enabled the response verifier and it observes the output of the particular CUT and similarly when it completes for all vectors or when it performs HIT for all vectors, then the bits of the response verifier are examined. while in test mode the inputs are given from the test pattern generator and then verified by the response verifier. The performance of the architecture can be measured by the hardware complexity which should be less and concurrent test latency which we can call as testing time, which can be also measured by the number of clock cycles or time units.

2. Proposed Block Diagram of BIST Scheme

The proposed BIST architecture is shown in Figure 2. Now suppose the CUT has n inputs, then the possible input vectors are $2^n$. There is very difficult to check all the vectors and verifying the outputs for more inputs so there is a test patterns are generated according to the particular CUT. Here we are dealing with online BIST with a concurrent approach so we are observing the coming input vectors considering a window from the input vector. We are dividing the input vectors between two parts by considering the window size as $W$ and also it should be $W=2^w$ where $w$ is just an integer which should be less than the number of inputs. so at every time the vectors are observed from considered window and if all vectors perform HIT then the response verifier is enabled and it examined all the contents of it.

The input vectors are divided into two sets of bits which should be like $w+h=n$ where h bits are the higher order bits of the window and it shows whether the input vectors are matched with the current window which has considered and it shows the relative location of the vectors and are stored in the memory cells which can be designed in the BIST controller logic block. If the current incoming vector belongs to the current window, then it has removed from the active test sets and if all vector has performed HIT then the response verifier will capture the response to that vector and then next window will be considered and similarly for the other windows the same operation is occurring.

Figure 3 shows the decoding structure which is not a normal decoder, but it has two external signals cmp and tge. According to these signals decoder can be operated and gives a respective output bits.

Figure 3. Decoder.

The Figure 4 shows the proposed BIST architecture by considering the CUT as one of the most important parts of the microprocessor which is arithmetic unit logic and it has considered here for 8 bit input. The ALU has been implemented for the arithmetic and logical operations based on the input signals. So here among 8 bits of the input vector four least significant bits are considered as window and so there are 16 possible patterns which can be generated by the test pattern generation and compared with the current considered window with the help of comparator. When the they matched the cmp signal get enabled and so as tge signal then the decoder outputs are equal to one. When tge and cmp signals are disabled, then decoder outputs are zero. When tge is disabled and cmp is enabled the module operated as a normal decoder.

The architecture of the proposed BIST scheme with $n=8$, $h=4$ and $w=4$ is shown in Figure 4. The BIST con-
controller consists of a test pattern generator which can be a standard linear feedback register, counter or cellular automata. Here we have generated the patterns using cellular automata. It also contains logic block which consists of logic, memory cells, which are nothing but a latch using nor gates, d-flip-flops with overflow signal and response verifier. The d-flipflop is used for a unit delay, so that overflow signal drives a tge a signal. There are also two external signals reset and clock. The clk and clk’ signals are active high and active low respectively.

The following cases are the operation of the logic block which has been shown in Figure 5: 1) Reset of the block, 2) HIT occurs in the particular vector for a first time, 3) Vector from the same window but not for the first time, 4) When all cells are filled and we can proceed to the next window.

2.1 Reset of the Logic Block
Initially the logic module is reset through the reset signal and when reset signal is enabled, then tge signal is also enabled and all decoder outputs are enabled, and according to the decoder output the memory cell are filled with one value or zero value.
2.2 HIT occurs for the Current Vector and Reaches to the CUT for the First Time

This is the BIST operation during normal mode and during this mode the inputs are driven through the normal inputs as well as CBU and the least significant bits are the w bits are driven to the decoder inputs and the h higher order bits are generated by the cellular automata and given to the comparator input. When vector reaches to the CUT input, then comparator is enabled and it compares the current pattern with the window bits and if they are equal, then one of the decoder output is enabled and according to the one and that particular memory cell becomes enabled. Similarly for all vectors when the HIT occurs, then the counter enables via rev signal.

2.3 Vector from the Same Window but not for the First Time

If the memory cell contains one value for the considered better than the rave signal is not enabled, and similarly w stage counter is also not enabled.

2.4 When All Cells are Filled

When all cells are filled or we can say full then w stage counter becomes 1 it means all the contents of counter are 1 so review signal causes to enable overflow signal after one unit delay through D flip flop. This operation occurs during the 1st half cycle of the clock and after the completion of one window it will start again for the next window in the next clock cycle.

3. Test Pattern Generation

In the BIST architectures test pattern generation is the most important block to consider. Here we have preferred cellular automata for the generation of patterns as it has an advantage to generate the most random pattern compared with linear feedback shift register and counter. How this will affect the BIST scheme that we will check at the end. The cellular automata has generated the patterns using rule 90.

Figure 6 shows the four stage cellular automata which consists of D flip flops and XOR gates. It generates 16 test patterns of four bits. The connection of a cell with neighbouring cell is based on the rules defined by rule 90 and rule 150. Here we have designed using rule 90.

4. Simulation Results

The BIST architecture is implemented in Verilog HDL. Here we have considered a circuit under test as 8 bit arithmetic logic unit which includes addition, subtraction and some logical operation like AND operation and NOT operation.

Figure 7 is the RTL diagram of the 8 bit ALU which we have considered as a circuit under test and here we are implementing BIST architecture to test ALU. So here we are monitoring the incoming input vectors at the inputs of the ALU and verifying the output using response verifier.
Figure 7. 8 bit ALU as CUT.

Figure 8. Output waveforms of the CUT.
Figure 8 shows the output waveforms of 8 bit ALU. As shown in figure add_sub signals indicates arithmetic operation and func signal indicates the logical operation.

Figures 9 and 0 show the most important RTL diagrams of logic block of BIST controller and final BIST architecture with all signals and instantiation. It also includes a global clock signal and reset for the logic module.

5. Evaluation Parameters and Comparison

There are two main parameters we have considered to evaluate our BIST architecture which are area complexity as it is the most important factor to evaluate which we have obtained using Xilinx ISE tool area summary report and another parameter is concurrent test latency which defined by the total testing time in terms of time unit or clock cycles which we have obtained using timing report.

Table 1 shows the comparison of different BIST architecture using different types of TPGs as we have mentioned before. From the table, we can analyze that cellular automata is most suitable for our concurrent testing scheme, though it has more hardware than LFSR but CTL is less and that is what we need in our case.

6. Conclusion and Future Scope

As BIST architectures are the best solution to test different combinational and digital logical circuits, we have proposed an efficient testing scheme in online testing, which has used the concept of monitoring and observ-
ing the input vectors for the particular CUT and storing the window locations in the different logic memory cells which we have used NOR based latch are used to observe the input vector has performed a HIT or not. For further modification we can replace the logic memory cells with static or dynamic memory cells so that we can reduce area overhead. The proposed BIST architecture is more efficient using cellular automata as a test pattern generator.

7. References

1. Voyiatzis I, Efstathiou C. Input vector monitoring concurrent BIST architecture using SRAM cells. IEEE Transactions on VLSI Systems. 2014; 22(7):1625–29.
2. Akshay D, Sharma SM, Vyas LA, Sivasankaran K. Design of low power and area efficient 4-bit arithmetic and logic unit using nanoscale FinFET. Indian Journal of Science and Technology. 2015 Jan; 8(S2):1–7. DOI: 10.17485/ijst/2015/v8iS2/70759.
3. Begum JT, Naidu HS, Vaishnavi N, Sakana G, Prabhakaran N. Design and Implementation of Reconfigurable ALU for Signal Processing Applications. Indian Journal of Science and Technology. 2016 Jan; 9(2):1–6. DOI: 10.17485/ijst/2016/v9i2/86343.
4. Nandal A, Vigneswaran T, Rana AT. Booth multiplier using reversible logic with low power and reduced logical complexity. Indian Journal of Science and Technology. 2014 Jan; 7(4):525–29. DOI: 10.17485/ijst/2014/v7i4/48644.
5. Athithan S, Shukla VP, Biradar SR. Voting rule based cellular automata epidemic spread model for leptospirosis. Indian Journal of Science and Technology. 2015 Feb; 8(4):337–41. DOI: 10.17485/ijst/2015/v8i4/60441.
6. Voyiatzis I, Paschalidis A, Gizopoulos D, Kranitis N, Halatsis C. A concurrent BIST architecture based on a self-testing RAM. IEEE Transactions on Reliability. 2005; 54(1):69–78.
7. Kochte MA, Zoellin C, Wunderlich HJ. Concurrent self-test with partially specified patterns for low test latency and overhead. 2009 14th IEEE European Test Symposium; 2009 May. p. 53–58.
8. Saluja KK, Sharma R, Kime CR. A concurrent testing technique for digital circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems. 1988; 7(12):1250–26.
9. Sharma R, Saluja KK. Theory, analysis and implementation of an on-line BIST technique. VLSI Design. 1993; 1(1):9–22.
10. Zorian Y, Ivanov A. An effective BIST scheme for ROMs. IEEE Transactions on Computers. 1992; 41(5):646–53.
11. Voyiatzis I, Haniotakis T, Efstathiou C, Antonopoulou H. A concurrent BIST architecture based on monitoring square windows. 2010 5th International Conference on Design and Technology of Integrated Systems in Nanoscale Era (DTIS), Hammamet; 2010 Mar. p. 1–6.
12. Kochte MA, Zoellin C, Wunderlich HJ. Concurrent self-test with partially specified patterns for low test latency and overhead. in Proc. 14th Eur. Test Symp., Seville ,2009, pp. 53–58.
13. A. Kumaravel, Meetei ON. An application of non-uniform cellular automata for efficient cryptography. Indian Journal of Science and Technology,2013 May, 6(S5), pp. 1-7. DOI: 10.17485/ijst/2013/v6iS5/33355.
14. Kamaraj A, Marichamy P, Devi SK, Subraja MN. Design and implementation of adders using novel reversible gates in quantum cellular automata. Indian Journal of Science and Technology. 2016 Feb; 9(8):1–7. DOI: 10.17485/ijst/2016/v9i8/87929.
15. Sinha N, Ravi V. Implementation of health monitoring system using mixed environment. Indian Journal of Science and Technology. 2015 Aug; 8(20):1–7. DOI:10.17485/ijst/2015/v8i20/77727.
16. Pandharapurkar NG, Ravi V. Design of BIST using self-checking circuits for multipliers. Indian Journal of Science and Technology. 2015 Aug; 8(19):1–7. DOI: 10.17485/ijst/2015/v8i19/77006.