Design and Implementation of Low Power LFSR for Fast ATPG Process

Shahataj Tabassum[1], Shilpa B.Darvesh[2]

[1]PG Scholar, Department of ECE, MVSR Engineering College, Hyderabad, India

[2]Asst. Professor, Department of ECE, MVSR Engineering College, Hyderabad, India

Abstract – In this paper, a new test pattern generation methodology is proposed to check the devices or circuits after manufacturing that takes advantage of shared memory multi-core systems which consumes less power and delay. Unlike the other test generation patterns this method generates the test patterns by Low Power LFSR using Gray code generator which consumes less power due to its less switching activity. This methodology is implemented in Half Adder and FIR filter circuits to check their functionality. This method is synthesized and simulated using Xilinx 14.7. The experimental results demonstrate that the proposed methodology achieves less power consumption and delay.

Index Terms – ATPG, Low Power LFSR (Linear Feedback Shift Register), Gray Code generator, Memory multi-core, Half Adder and FIR filter.

1. INTRODUCTION

ATPG is an electronic design automation method/technology used to recognize whether the circuit is fault free or faulty. To verify the functionality of the circuit/device after manufacturing some test patterns are applied to the CUT. As the technology is shrinking in the IC manufacturing process multiple processing units (cores) are placed on a single chip as well as large amounts of on chip memory but these developments offer high processing power. While previously proposed procedures are very effective [2]-[4], they are inherently non-parallel and thus, cannot perform automatic parallelization using sophisticated compilers. In [5]-[7] the fault portioning is focused to increase the fault coverage. After that parallel ATPG has been proposed which uses on-chip multi-core era, but the ATPG components among multiple processing units are placed on the same chip [8,9]. Next various parallel ATPG methods such as circular pipeline parallel ATPG[10] and GPU based ATPGs[15-18] are came into existence but they are having some disadvantages like same test sets are re-generating each time and limits the speed-up scalability.

In this work we propose a parallel test pattern generation methodology which uses the shared-memory multi-core systems geared towards high speed. To generate the pseudo random test patterns Low Power LFSR is used which consumes less power and delay. In Low Power LFSR the patterns are in Gray code by which switching activity less compared to LFSR.

The rest of the paper is arranged as follows. Section 2 & 3 focuses on existing and proposed methods respectively. In section 4 and 5 proposed method is implemented on Half Adder and FIR filter circuits. Section 6 shows simulation and synthesis results while section 7 concludes the paper.

2. EXISTING METHODOLOGY

In existing method to check whether the device or circuit is working properly or not some of the test patterns are applied to the CUT (Circuit Under Test) which are generated by LFSR which uses XOR operation. The block diagram of this methodology is shown below.
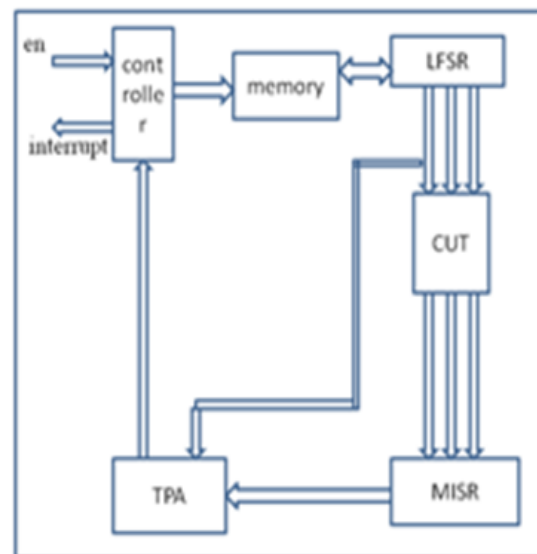

Figure 1: ATPG system on-chip circuit

The block diagram consists of following blocks such as LFSR, CUT, MISR, TPA, Controller and Memory.

2.1 Linear Feedback Shift Register

LFSR is used for generating the test patterns by which the functionality of the circuit is verified. It is a shift register whose input bits are operated by the XOR of some bits of the overall shift register value. The initial value which is given to the LFSR is called the seed.
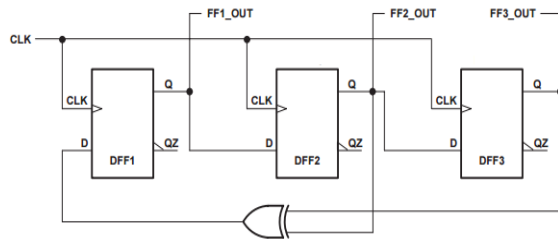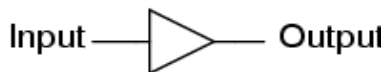
Figure 2: LFSR Circuit

Figure 3: XOR Gate

2.2 Circuit under Test

The Circuit under Test is circuit or device which has to be tested. Buffer circuit is used in this existing method. The output of the Buffer circuit is same as the input.

Figure 4: Buffer circuit

2.3 Multiple Input Shift Register

The Multiple Input Shift Register is used in an output section; the existing output of the circuit/device is to be compared with fault free output for detection of error. It contains all D flip-flops whose output is feed backed to the input and then bits will be shifted. This shifted output patterns are compared with the output patterns of LFSR to check whether circuit is faulty or not.

If CUT has any type of error the MISR patterns does not match with the LFSR patterns. This is because if CUT does not work properly it does not give correct output to the MISR, by that the MISR patterns also have some errors.
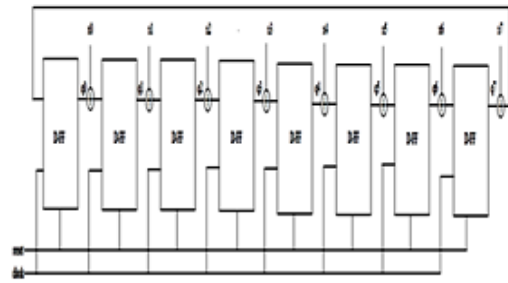

Figure 5: Multiple Input Signature Register

2.4 Test Pattern Analyzer

The Test Pattern Analyzer compares the output patterns of LFSR and MISR and gives the output to controller.

2.5 Controller

The controller has one input as enable and one output as interrupt. When enable is high then only the whole process of the ATPG is started. If the output patterns of LFSR and output patterns of MISR are not same then the interrupt will be 1 otherwise 0.

2.6 Memory

The ATPG system uses the concept of memory multi-core which is used to store the pseudo random test patterns by the speed of the process increases. The memory multi-core is divided into multi-cores i.e., two multi-cores in which the LFSR test patterns are stored in memory. As the number of cores increases the speed also increases.

3. PROPOSED METHODOLOGY

Now-a-days the desire for the portable devices is increasing enormously. So the Power reduction is necessary to satisfy the demands of the consumer. The test patterns generated using LFSR finds complexity in Power. For the purpose of power reduction the patterns ere generated using Low Power LFSR which uses Gray code rather than binary code. The purpose of choosing gray code is to reduce the power requirements.

In proposed methodology Low Power LFSR is used for the generation of the test patterns using Gray code Generator which consumes less power and delay. The Low power LFSR is similar to the LFSR but the initial seed is converted into Gray code using Gray code generator. After that, rest of the patterns is generated using XOR operation. This method reduces power consumption and delay up to 55% and 6% respectively.

3.1 Low Power LFSR

In ATPG to verify the functionality of the circuit some of the test patterns are applied to the CUT (Circuit Under Test). This Low Power Linear Feedback Shift Register is used to generate the test patterns with less power consumption and delay. Normally LFSR is used to generate the test patterns but they

consume high power. So the Low Power LFSR method is proposed to reduce the power consumption and delay.

The LFSR is a shift register whose input bit is a linear function of its previous state. Thus, an LFSR is a shift register whose input bit is operated by the XOR gate and after that the output is feedback to the input. The initial value of the LFSR is called the seed. This LFSR consumes high power.

The Low Power LFSR is same as the normal LFSR but it uses gray code generator by which the initial seed is converted from binary code to gray code, after that rest of the patterns are generated using XOR operation. The switching activity in gray code is less compared to the binary code. So the Low Power LFSR consumes less power. Low Power LFSR generated patterns are in Gray code which has less switching activity compared to the binary code. As the switching activity is less, the glitches, noise, power consumption and delay are also less compared to the binary code. The Gray code generator is used to convert binary code into gray code.
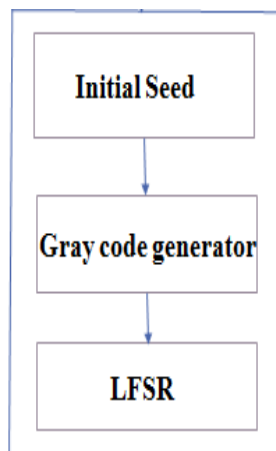
The Low Power LFSR block is shown in fig. 6.

Figure 6: Low Power LFSR

3.2 Binary to Gray Code Conversion

It is a very simple process which includes various steps for the conversions. By observing below steps we can get a clear idea about this conversion.

(1) Firstly the M.S.B. of the gray code should be exactly equal to the 1st bit of that binary number.

(2) Next the 2nd bit of the code is exclusive-or of the 1st and 2nd bits of that binary number, i.e. the result should be 0 when both the bits are same and otherwise the result will be 1.

(3)The 3rd bit of that gray code is equal to the exclusive-or of the 2nd and 3rd bit of that binary number. Like this the process of the Binary to gray code conversion continues. An example of Binary to Gray conversion is shown below.
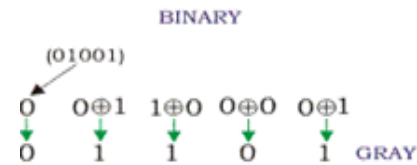

Figure 7: Binary to gray conversion

Hence the equivalent gray code of that binary code is 01101. Now see the example where the M.S.B. bit of the binary is 0 so the M.S.B bit for gray code is also 0. Next, the XOR of the 1st and 2nd bit is 1 because both the bits are different. Again go to the next step where the XOR of 2nd and 3rd bit is again 1 because they are different. Next, XOR of 3rd and 4th bit is 0 because both the bits are equal. Finally the XOR of 4th and 5th bit is 1 because they are different. Like this the gray code conversion of 01001 is 01101.

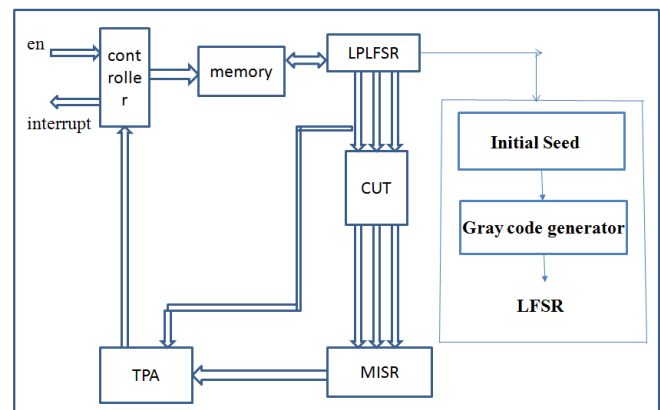The block diagram of proposed methodology is shown in fig.8.

Figure 8: Low power ATPG system

It is similar to fig. 1 except LPLFSR block. Each block rather than low power LFSR of above methodology are explained in section II.

The advantage of Gray code compared to the binary code is it has only one bit which changes for each step that means it has less switching activity so that the power consumption is less. This will be useful in the circuits which are sensitive to the glitches or noise.

This proposed methodology reduces power and delay up to 55% and 6% respectively.

4. PROPOSED METHOD APPLIED TO HALF ADDER

This proposed method is implemented in Half Adder to verify its functionality. Firstly the Half Adder is designed using verilog code in Xilinx after that its functionality is checked using proposed method. In Low Power LFSR methodology the Half Adder is taken as CUT.
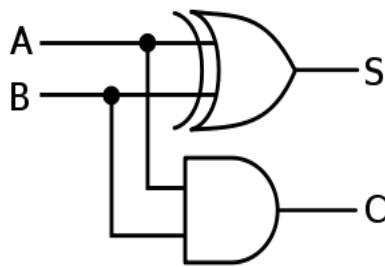
Figure 9: Half Adder Circuit

This Half Adder is instantiated into CUT. The CUT takes the output of LFSR as the input and generates the CUT output. If LFSR has 10 bits then first 5 MSB bits are taken as A input and second 5 LSB bits as B input. After getting CUT output as Sum and Carry they are combined as 10 bits which are inputs to the MISR. Next the output patterns MISR are generated. The TPA compares LFSR & MISR patterns and gives the output as interrupt. If the interrupt is generated as '0' then the CUT is faulty circuit, otherwise CUT is fault free circuit.

The power consumption and delay both are reduced up to 58% and 8% respectively due to less switching activity in Gray code patterns.

5.   PROPOSED METHOD APPLIED TO FIR FILTER

This proposed method is implemented in FIR Filter to verify its functionality. Firstly the Half Adder is designed using verilog code in Xilinx after that its functionality is checked using proposed method. In Low Power LFSR methodology the FIR Filter is taken as CUT.
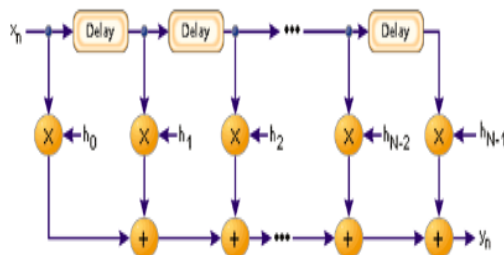

Figure 10: FIR Filter Circuit

This FIR Filter is instantiated into CUT. The CUT takes the output of LFSR as the input and generates the CUT output. The coefficients for FIR filter are generated using Matlab software. In this paper Low pass filter is used whose order is 5, cut-off frequency is 0.5*pi and uses blackman window to generate coefficients.

The coefficients generated are 0.0000, 0.1083, 0.5000, 0.1081 and 0.0000. These coefficients are used to design Low Pass FIR filter. The power consumption & delay both are reduced due to less switching activity in Gray code patterns.
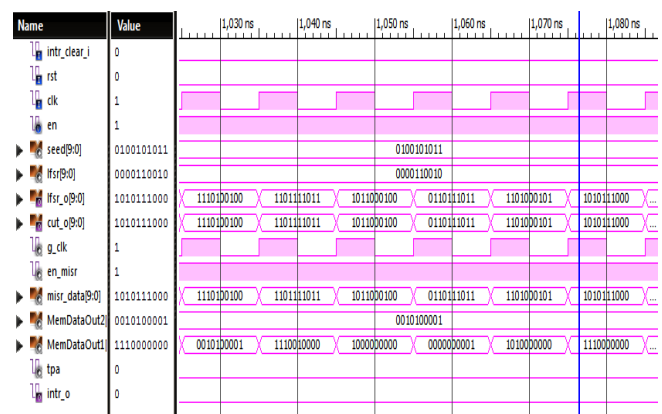
6.   SIMULATION AND SYNTHESIS RESULTS

The proposed method is simulated and synthesized using verilog HDL on Xilinx 14.7.The simulation results are shown below and the synthesis reports are explained in below tables.
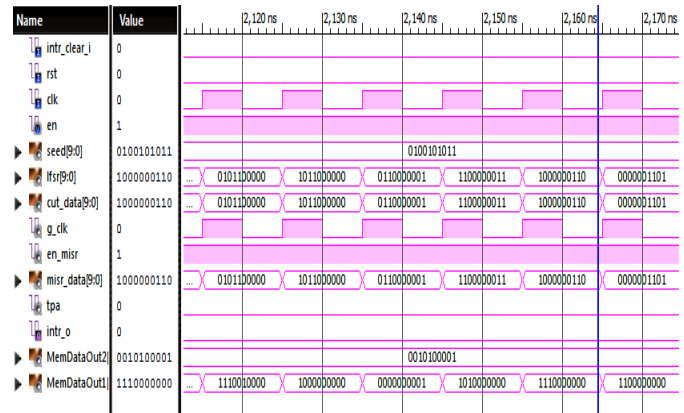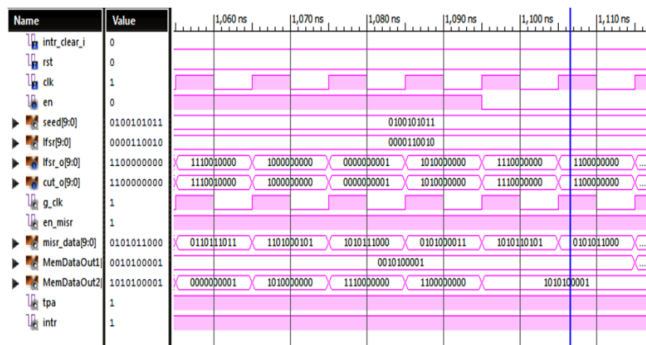

(a) Top module output of Existing method which shows circuit is fault free.
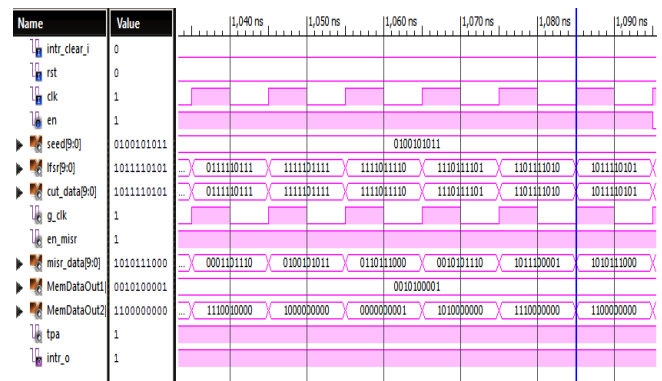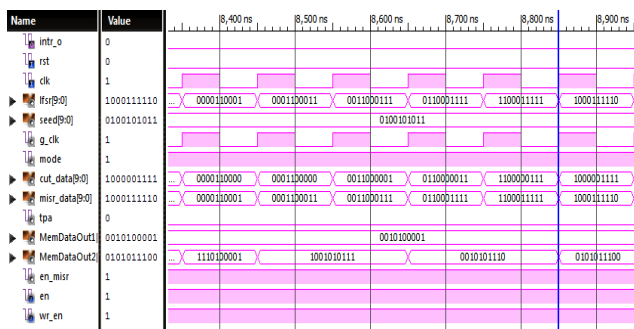

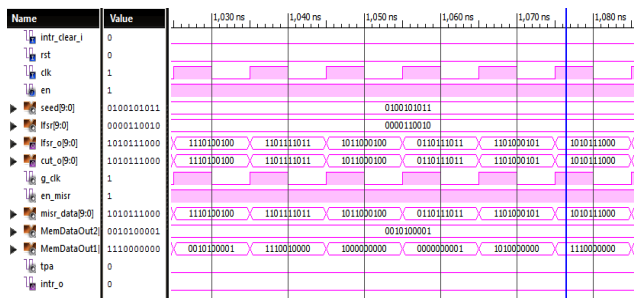(b) Top module output of Existing method which shows circuit is faulty.

(c) Top module output of proposed method which shows circuit is fault free.
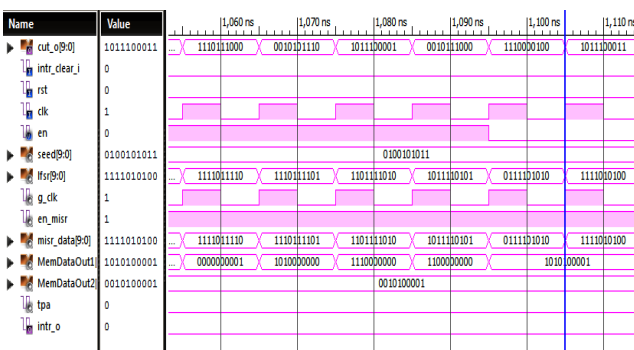
(d) Top module output of proposed method which shows circuit is faulty.
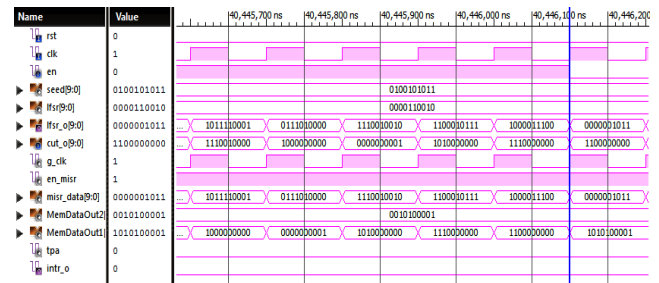

(e) Top module output of Half Adder using existing method.

(f) Top module output of Half Adder using proposed method.

(g) Top module output of FIR filter using existing method.

(h) Top module output of FIR Filter using proposed method.

Figure 11: HDL Simulation results

The synthesized report of proposed method for logic utilization is summarized in Table I.

TABLE I

LOGIC UTILIZATION OF PROPOSED METHOD

Logic	Utilization
Number of slice registers	47
Number of slice Flip Flops	42
Number of 4 input LUTs	29
Number of bonded IOBs	25
Average fan-out of non-clock nets	2.76

The comparison between existing and proposed methods is shown in Table II.

TABLE II

POWER, DELAY AND POWER DELAY PRODUCT COMPARISON

Method	Power (W)	Delay (ns)	PDP (pJ)
ATPG using LFSR	0.034W	6.469ns	219.9pJ
ATPG using low power LFSR	0.015W	6.097ns	91.4pJ

The comparison of Half Adder using existing and proposed methods is shown in Table III.

TABLE III

Method	Power (W)	Delay (ns)	PDP (pJ)
Half Adder using LFSR	0.082	7.051	578.1
Half Adder using low power LFSR	0.034	6.469	219.9

The comparison of FIR filter using existing and proposed methods is shown in Table IV.

TABLE IV

Method	Power (W)	Delay (ns)	PDP (pJ)
FIR filter using LFSR	0.082	7.051	578.1
FIR filter using low power LFSR	0.034	6.469	219.9

7. CONCLUSION

In this paper a parallel test pattern methodology for shared memory multi-core environments is proposed using low power LFSR and it is implemented on Half Adder and FIR filter circuits to verify their functionality. This method enhances effective test patterns which are used for testing a device or circuit with less power consumption and delay. The proposed method is simulated and synthesized by using Xilinx 14.7. From the above values we can say that power consumption and delay are successfully reduced by 55% and 6% respectively. The comparison results shows that the proposed design with less power and delay can be good candidate and suitable for design automations with effective testing analysis.

The future scope of this Low Power ATPG methodology is to reduce the delay by which the time for generating the pseudo random test sequences can be increased.

REFERENCES

[1] Stavros Hadjitheophanous, Stelios N. Neophytou and Maria K. Michael, " Utilizing shared memory multi-cores to speed up the ATPG process," in Proc. of ETS , pp. 1-6 , 2016.

[2] K. Scheibler, D. Erb and B. Becker, "Improving test pattern generation in presence of unknown values beyond restricted symbolic logic," in Proc. of ETS, pp. 1-6, 2015

[3] S. Eggersglub, K. Schmitz, R. Krenz-Baath and R. Drechsler, "Optimization-based multiple target test generation for highly compacted test sets," in Proc. of ETS, pp. 1-6, 2014.

[4] I.Pomeranz, "Generation of compact multi-cycle diagnostic test sets," in Proc. of ETS, pp. 1-1, 2013.

[5] S. Patil and P. Banerjee, "Fault partitioning issues in an integrated parallel test generation/fault simulation environment," in Proc. of ITC, pp. 718–726, 1989.

[6] J. Wolf, L. Kaufman, R. Klenke, J. H. Aylor, and R. Waxman, "An analysis of fault partitioned parallel test generation," IEEE Trans. on CAD, vol. 15, no. 5, pp. 517–534, 1996.

[7] A. Czutro, I. Polian, M. Lewis, P. Engelke, S. M. Reddy and B. Becker, "Thread-parallel integrated test pattern generator utilizing satisfiability analysis," International Journal of Parallel Programming, vol. 38, pp. 185-202, 2010.

[8] K-Y. Liao, C-Y. Chang and JC-M Li "A parallel test pattern generation algorithm to meet multiple quality objectives," IEEE Trans. on CAD, vol.30, no. 11, pp. 1767-1772, 2011.

[9] K.-W. Yeh, M.-F. Wu and J.-L. Huang, "A low communication overhead and load balanced parallel ATPG with improved static fault partition method," in Proc. of Intl. Conf. on Algorithms and Architectures for Parallel Processing, pp. 362-371, 2009.

[10] K-W. Yeh, J-L. Huang, H-J. Chao and L-T. Wang "A circular pipeline processing based deterministic parallel test pattern generator," in Proc. of ITC, pp. 1-8, 2013.

[11] JC-Y. Ku, RH-M. Huang, LY-Z. Lin and CH-P. Wen,"Suppressing test inflation in shared-memory parallel Automatic Test Pattern Generation," in Proc. of ASP-DAC, pp. 664-669, 2014.

[12] X. Cai, P. Wohl, J.A. Waicukauski and P. Notiyath "Highly efficient parallel ATPG based on shared memory," in Proc. of ITC, pp. 1-7, 2010.

[13] X. Cai and P. Wohl, "A distributed-multicore hybrid ATPG system," in Proc. of ITC, pp. 1-7, 2013.

[14] X. Cai, P. Wohl and D. Martin "Fault sharing in a copy-on-write based ATPG system," in Proc. of ITC, pp. 1-8, 2014.

[15] E. Schneider, S. Holst, M. A. Kochte, X. Wen, and H.-J. Wunderlich "GPU-accelerated small delay fault simulation," in Proc. of DATE, pp. 1174–1179, 2015.

[16] M. Li and M. Hsiao,"3D parallel fault simulation with GPGPU," IEEE Trans. on CAD, vol. 30, no. 10, pp. 1545–1555, 2011

[17] K-Y. Liao, S-C. Hsu and JC-M. Li "GPU-based N-detect transition fault ATPG," in Proc. of DAC, pp. 28, 2013.

[18] JC-Y. Ku, RH-M. Huang, LY-Z. Lin and CH-P. Wen,"Suppressing test inflation in shared-memory parallel Automatic Test Pattern Generation," in Proc. of ASP-DAC, pp. 664-669, 2014.

Authors

Shahataj Tabassum completed her B.Tech in 2015 from Sri Indu College of Engg & Tech. Affiliated to JNTUH and pursuing M.E (2015-2017) in Embedded System & VLSI Design from MVSR Engineering College, Affiliated to Osmania University, Hyderabad, India. Her area of interest includes Low Power VLSI.

Shilpa B. Darvesh is currently working as Assistant Professor in Department of Electronics & Communication Engineering since 2006 in MVSR Engineering College with a teaching experience of 11 years. She received her Master's degree from JNTUH University. Her area of interest includes VLSI.