International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 3, March - 2014

Design and Implementation of Low Transition LFSR for Efficient BIST Architectures

M. Suriya
II-ME (Applied electronics)
Jayaram College of Engineering and Technology
Trichy, India

R. Mohandas
Assistant professor (ECE)
Jayaram College of Engineering and Technology
Trichy, India

***Abstract*— In this paper we present a new low transition LFSR for efficient BIST architecture that produce the output for many clock cycles at once. In this paper, we introduce a novel architecture to perform high speed multiplication using ancient Vedic maths techniques. A new high speed approach utilizing 7:2 compressors for addition has been incorporated in the same. The compressor based multiplier is introduced in this paper, is two times faster than the popular methods. This proposed multiplier shows the reduction in power consumption and area. The proposed multiplier has shown the static, dynamic and total power consumption during the testing operation up-to 46.15%, 23.58% and 64.51% respectively for the Quartus II 9.1.**

Keywords- Vedic multiplier; 7:2 Compressor; Urdhva Triyagbhyam; Built-In Self-Test; LFSR technique; low-power pattern generation.

I. INTRODUCTION

Built-in Self Test, or BIST, is the technique of designing additional hardware and software feature into integrated circuits to allow them to perform self-testing. The main drivers for the widespread development of BIST technique are the fast-rising costs of ATE testing and the growing complexity of circuits. Such complex devices require mixed signal testers that posses special digital and analog testing.

Logical Built-in Self Test or LBIST, which is designed for testing random logic, typically employs a PRPG to generate input patterns that are applied to the device's internal scan chain, and MISR for obtaining the response of the device to these test input patterns. An incorrect MISR output indicates the defect is present in the device.

BIST is fast becoming an alternative solution to the rising costs of external electrical testing and increasing intricacy of devices. This approach will find the greater use in a wider variety of circumstances as more and better BIST techniques are developed. Still, BIST proponents are positive that BIST will someday be the chosen mode of testing, instead of being merely an alternating to external ATE testing as it is today.

II. BIST ARCHITECTURE

Stored-pattern BIST may use programs or micro-programs, typically stored in ROM, to perform functional tests of the hardware. Successful applications of such techniques exist, but they are not our focus here. In alternative techniques, we use traditional automatic test pattern generation (ATPG) and fault simulation to generate the test patterns. We store the patterns on the chip or board, apply them to the CUT when BIST is activated, and compare the CUT responses with the corresponding stored responses. Because of the stored data's magnitude, this method is attractive only in limited cases. These include testing structured logic and detecting a small number of faults not handled by other BlST techniques. Overall, although stored pattern BIST can provide excellent fault coverage, it has limited applicability due to its high area overhead. The BIST architecture is shown in figure 2.1.

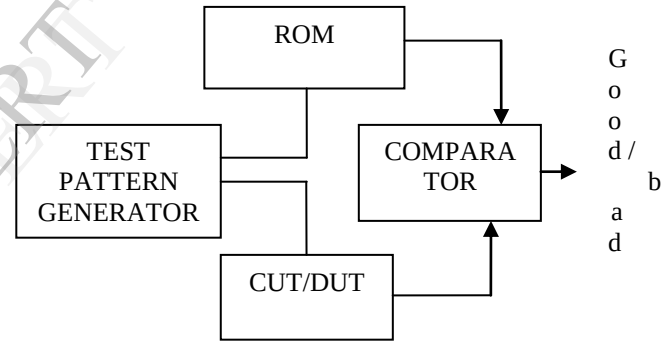

Figure 2.1 Bist block

III. PROGRAMMABLE READ-ONLY MEMORY

A programmable read-only memory (PROM) or field programmable read-only memory (FPROM) or one-time programmable non-volatile memory (OTP NVM) is a form of digital memory where the setting of each bit is locked by a fuse. Such PROMs are used to store up programs enduringly. The key distinction from a severe ROM is that the programming is applied after the device is constructed. A typical PROM comes with all bits reading as "1". Burning a fuse bit during programming causes the bit to read as "0". The memory can be programmed just once after developed by "blowing" the fuses, which is a permanent process. Blowing a fuse opens a connection as programming an antifuse closes a connection. While it is impossible to "unblow" the fuses, it is often to modify the contents of the memory after initial programming by blowing supplementary fuses, changing some remaining "1" bits in the memory to "0"s.

IV. COMPARATOR

A comparator is a device that compares two values and it delivers an output good or bad. Comparator is shown in Figure 2.2.

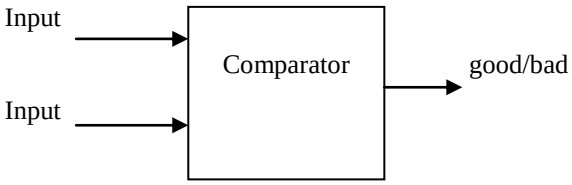

Figure 2.2 Comparator

V. URDHVA TIRYAKBHYAM SUTRA

The proposed "Urdhva Tiryagbhyam" multiplier algorithm is used for the multiplication of two numbers in the decimal number system. In this work, we apply the ideas to the binary number system to make the proposed algorithm attuned with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. The algorithm can be used for n x n bit number. Since the partial products and their sums are calculate in parallel, the multiplier is sovereign of the clock rate of the processor. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases slowly as compared to other conventional multipliers.

.

VI. MULTIPLICATION OF TWO DECIMAL NUMBERS

To illustrate this scheme, let us consider the multiplication of two decimal numbers 123 x 541 by Urdhva-Tiryakbhyam method as shown in Figure.2.3. The digits on the both sides of the line are multiplied and added with the carry from the preceding step. This generates the bits of the result and a carry. This carry is added in the subsequent step and hence the procedure goes on. If more than one line are there in one step, all the results are added to the preceding carry. In each step, LSB acts as the result bit and all other bits act as carry for the subsequent step. In the beginning the carry is assumed as zero.

STEP 1

1 2 3 Result = 3

5 4 1 Pre Carry = 0

3 3

STEP 2

1 2 3 Result = 14

5 4 1 Pre Carry = 0

4 3 14

STEP 3

1 2 3 Result = 24

5 4 1 Pre Carry = 1

5 4 3 25

STEP 4

1 2 3 Result = 14

5 4 1 Pre Carry = 2

6 5 4 3 16

STEP 5

1 2 3 Result = 5

5 4 1 Pre Carry = 1

6 5 6 4 3 6

123 x 541 = 6 5 6 4 3

Figure 2.3 Multiplication of two decimal numbers

As mentioned earlier, the partial products obtained are added with the help of full adders and half adders. This leads to additional hardware and additional stages, since the full adder is capable of adding only 3 bits at a time. In the next section two different types of compressor architectures are explored which assist in adding more that 3 bits at a time, with reduced architecture and increased efficiency in terms of speed.

4:2 COMPRESSOR ADDER

A compressor adder is a logical circuit which is used to improve the computational speed of the addition of 4 or more bits at a time. Compressors can efficiently replace them combination of several half adders and full adders, thereby enabling high speed operation of the processor which incorporate the same. The 4:2 compressor adder is used in this paper. A comparison of the 4:2 compressor with a corresponding circuit, using full adders and half adders has been given. A 4:2 compressor as shown in figure 2.4 is capable of adding 4 bits and one carry, and producing a 3 bit output.

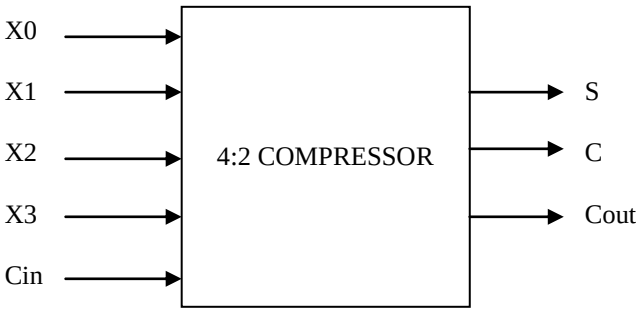

Figure.2.4 4:2 compressor adder

7:2 COMPRESSOR ADDER

Similar to its 4:2 compressor counterpart, the 7:2 compressor as shown in Figure.2.5, is capable of adding 7 bits of input and 2 carry's from the previous stages, at a time. In this work, we have designed 7:2 compressor utilizing two 4:2 compressors, two full adders and one half adder. The architecture for the same has been shown in Figure. 2.6. As mentioned earlier, since the 4:2 compressor shows a significant increase in speed by around 66.6%, utilizing the same in this architecture would improve the efficiency as opposed to a conventional approach of adding nine bits at a time using only full adders and half adders. This leads to a great improvisation in speed of the processor.

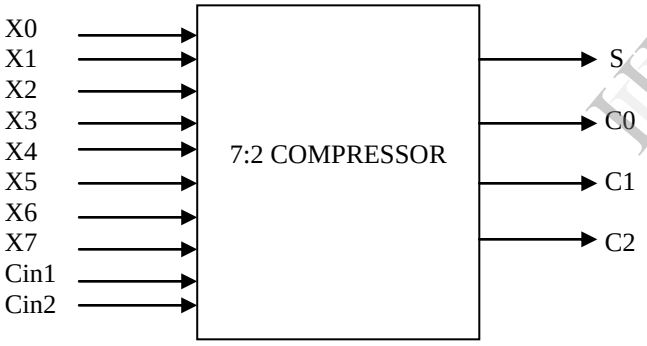

Figure.2.5 7:2 Compressor Adder

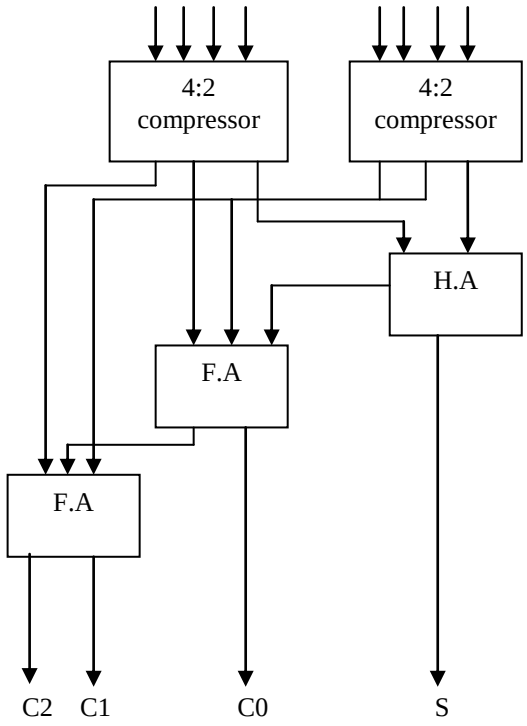

Figure.2.6 7:2 Compressor using 4:2 Compressor adder

VII. PRIOR WORK

XOR-LFSR

An LFSR is a shift register that advances the signal through the register from one bit to the next MSB. A few of the output are combined in XOR to form a feedback method. A LFSR can be formed by performing XOR on the outputs of two or more of the flip-flops together and feeding those outputs back into the input of one of the flip-flops as shown in Figure 3.1.

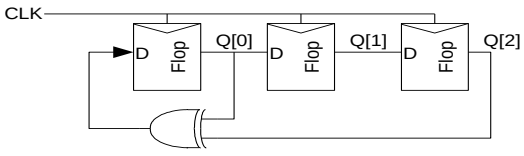

Figure 3.1 Linear Feedback Shift Register

Linear feedback shift registers are widely used in BlST because they are simple and fairly regular in structure, their shift property integrates easily with serial scan, and they can generate exhaustive and/or pseudorandom patterns. The typical components of an LFSR are D flip-flops and XOR gates. Despite their simple appearance, LFSRs are based on a rather complex mathematical theory. Here we present only the aspects of the theory that help explain their behavior as pattern generators and response analyzers pattern generators and response analyzers.

In test pattern generation mode, a pattern generated by an LFSR is the state of all the D flip-flops in the LFSR.

Obviously, we can deduce consecutive patterns generated by an LFSR by simulating it. But by associating polynomials with LFSRs and bit streams or vectors, we can use polynomial algebra to predict LFSR behavior. Throughout this discussion we discuss polynomials with binary coefficients, but almost all the results can be stated in more general terms. We can express a binary vector $R = r_m r_{m-1} \ldots r_o$ as a polynomial $r_m x^m + r_{m-1} x^{m-1} + \ldots + r_0$. The increased power consumption by the device in the manufacturing test environment therefore can in most cases exceed the maximum power consumption specification of the IC resulting in un-repairable device failures begins with a pattern generated using a conventional LFSR causing significant loss of yield. Previous Techniques for falling the power dissipation are

- RSIC test generation which is used to generate low power test pattern. In this method power utilization is reduced.
- Another technique was proposed in that is Low transition LFSR for BIST applications. This will be reduce the average and peak power of the circuit during testing.
- In a Fault model & ATPG algorithm is chosen first, and then test patterns are generated to achieve the desired fault coverage.
- *F.corno et al* proposed a Low power test pattern generation for sequential circuit. In this paper redundancy is introduced during testing. This will reduce the power utilization without affect the fault coverage.

VIII. PRESENT WORK

A. LOW TRANSITION LFSR

We combine our two proposed techniques of pattern generation for low-power BIST. The new LT-LFSR generates three intermediary patterns (Ti1, Ti2, and Ti3) between Ti and Tiþ1.We embed these two techniques into a bit-sliced LFSR architecture to create LTLFSR, which provides more power reduction compared to having only one of the R-Injection and Bipartite LFSR techniques in an LFSR.

B. ALGORITHM FOR LOW POWER LFSR

Step 1. en1en2 ¼ 10, sel1sel2 ¼ 11. The first half of LFSR is active and the second half is in idle mode. Selecting sel1sel2 ¼ 11, both halves of LFSR are sent to the outputs (O1 to On). In this case, Ti is generated.
Step 2. en1en2 ¼ 00, sel1sel2 ¼ 10. Both halves of LFSR are in inactive mode. The first half is sent to the outputs (O1 to On=2), but the RI injector circuit outputs are sent to the outputs (On2 þ1 to On1). Ti1 is generated.
Step 3. en1en2 ¼ 01, sel1sel2 ¼ 11. The second half of LFSR works and the first half is in inactive mode. Both halves are transferred to the outputs (On1 to On2) and Ti2 is generated.
Step 4. en1en2 ¼ 00, sel1sel2 ¼ 01. Both halves of LFSR are in inactive mode. From the first half, the injector outputs are sent to the outputs of LT-LFSR (On1 to On=2) and the second half sends the exact bits in LFSR to the outputs (On2 þ1 to On1) to generate Ti3.
Step 5. The process continues by going through Step 1 to generate Tiþ1.

LT-LFSR reduces the transitions between consecutive patterns that can be used for test-per-clock architecture. The generated patterns can be used for test-per-scan architecture to feed scan chains with a lower number of transitions. The LT-LFSR is shown in Figure 4.2.

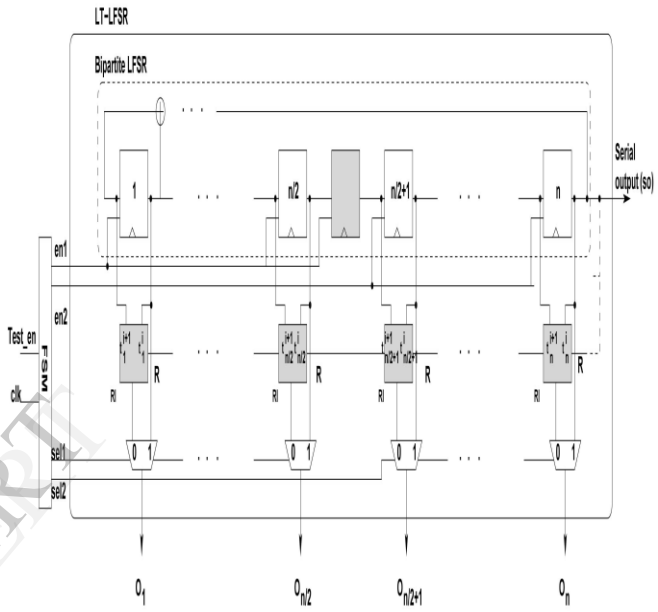

Figure 4.2 LT-LFSR structure.

V. SIMULATION RESULTS

The results obtained from the Quartus II 9.1, the device implemented on Cyclone III.

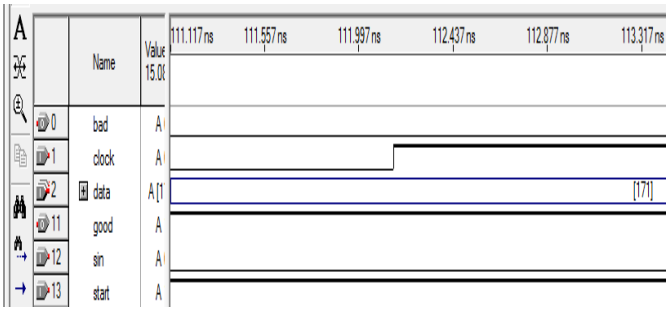

Figure 5.1 Simulation Results

IX. RESULT EVALUVATION

Comparison of Results of results of prior system and Present system is shown in table 5.2

Function	*Prior system*	*Present system*
Total logic element	165	182
Clock setup	130.11 Mhz (7.686 ns)	92.64 Mhz (10.794 ns)
Total thermal power Dissipation	87.53mW	82.32mW
Dynamic thermal Power Dissipation	8.86mW	7.31mW
Static thermal power Dissipation	51.78mW	51.77mW
I/O thermal power Dissipation	26.89mW	23.23mW

Table 5.2 comparison of conventional system and present system

VI. CONCLUSION

This paper presents a new low-power LFSR to reduce the average and peak power of combinational and sequential circuits during the test mode. It is observed that total power consumed in LT-LFSR is less than the normal LFSR. It is concluded that low power LFSR is very much useful for power optimization of BIST. In this paper, we have proposed a novel high speed architecture for multiplication of two 8 bit numbers, combining the advantages of compressor based adders and also the ancient Vedic maths method. A new 7:2 compressor architecture, based on the 4:2 compressor architecture is also discussed in this paper. It is seen that the speed of the proposed multiplier is higher than that of normal array multiplier the delay has been hugely reduced. 8X8 multiplier can be extended to 16 bit and higher order multipliers. This LT-LFSR is suitable for high speed and low speed multipliers.

ACKNOWLEDGEMENT

First and foremost I thank God, the almighty who stands behind and strengthen me to complete the project successfully. I would like to convey my sincere respect and gratitude towards my supervisor Mr.R.Mohandas M.E, His wide knowledge, severe research attitude and enthusiasm in work deeply impressed me and taught what a true scientific research should be. I am very thankful for the support he extended to me and the freedom to express my views. Words are inadequate to express the gratitude to my beloved parents and friends for their excellent and never ending co-operation.

REFERENCES

[1]. R. Huddar and Sudhir Rao, Kalpana M. and Surabhi Mohan "Novel High Speed Vedic Mathematics Multiplier using Compressors" IEEE, 2013.
[2]. Krishnaveni D., Umarani T.G.," vlsi implementation of vedic multip-lier with reduced delay" Volume 2, Issue 4, July 2012.
[3]. Mehrdad Nourani, Mohammad Tehranipoor, Nisar Ahmed," Low-Transition Test Pattern Generation for BIST-Based Applications" IEEE TRANSACTIONS ON COMPUTERS, VOL. 57, NO. 3, MARCH 2008.
[4]. D. Radhakrishnan and A. P. Preethy, "Low power CMOS pass logic 4-2 compressor for high-speed multiplication," Circuits and Systems, Proc. 43rd IEEE Midwest Symp., vol. 3, pp. 1296-1298, 2000.
[5]. Mohammad Tehranipoor, Mehrdad Nourani, Nisar Ahmed, "Low Transition LFSR for BISTBased Applications.
[6]. Fulvio Corno, Paolo Prinettom, Matteo Sonzar Forda "testability Analysis and ATPG on Behavioral RT- Level VHDL.
[7]. F.Corno, M.Rebaudengo, M.Sonza Reorda, "A Test Pattern Generation Methodology for Low Power Consumption", pp.1-5, 2008.
[8]. Tiwari, Honey Durga, et al., "Multiplier design based on ancient Indian Vedic Mathematics," Int. SoC Design Conf., 2008, vol. 2. IEEE Proc., pp. II-65 - II-68.
[9]. N. Ahmed, M. Tehranipoor, and M. Nourani, "Low Power Pattern Generation for BIST Architecture," Proc. Int'l Symp. Circuits and Systems, vol. 2, pp. 689-692, 2004.
[10]. N. Basturkmen, S. Reddy, and I. Pomeranz, "A Low Power Pseudo-Random BIST Technique," Proc. Int'l Conf. Computer Design, pp. 468-473, 2002.
[11]. M. Tehranipoor, M. Nourani, and N. Ahmed, "Low Transition LFSR for BIST-Based Applications," Proc. IEEE 14th Asian Test Symp, 2005.
[12]. Balwinder Singh, Arun Khosla, Sukhleen Bindra, " power optimization of LFSR for Low Power BIST", IEEE International conference 2009, pp. 311- 314.

BIOGRAPHIES

M.SURIYA received the **B.E.** degree in Electrical and Electronics Engineering from the Sri Krishna College of Engineering and Technology, Coimbatore, Anna University, Chennai, India, in 2012, the **M.E** degree in Applied Electronics from the Jayaram College of Engineering & Technology, Trichy, Anna University, Chennai, India, in 2014. Her research interest includes VLSI, Power Electronics.

R.MOHANDAS received the **B.E.** degree in Electronics and Communication Engineering from the Government Engineering College, Salem, Anna University, Chennai, India, in 2006, the **M.E** degree in Applied Electronics from the Paavai Engineering College, Namakkal, Anna University, Trichy, India, in 2011. He had the Industrial experience about four years. Currently he is working as an Assistant professor in Jayaram College of Engineering & Technology, Trichy, India. His research interest includes Digital Communication, High Speed networks, Digital Image Processing.