Built in Self Test (BIST) Implementation for Booth Multiplier

Parminder Kaur[1]
Guru Nanak Dev College of Engineering, Ludhiana
param.jas@gmail.com

Dr. S.S. Gill[2]
Guru Nanak Dev College of Engineering, Ludhiana
ssg@gndec.ac.in

Abstract: **With the development in the fabrication techniques the numbers of devices on controllers and Processor ICs are increasing on the SOC according to the Moore's Law. Multiplication is the toughest job in the arithmetic operations in the processor. It is the task whose instruction in the assembly language takes maximum machine cycles for the execution, which required too much power during the testing of these circuits. In this paper Booth Multiplier is proposed with on chip testing features. In this fast Booth Multiplier and then BIST circuit for Multiplier is implemented. Then area utilization and power consumption is analysed for both cases.**

Keywords: **BIST, Booth multiplier, Faults, Test patterns.**

I. INTRODUCTION

Very Large Scale Integration (VLSI) has a dramatic impact on the growth of digital technology. VLSI has not only reduced the size and the cost, but has also increased the complexity of the circuits. This brought significant improvements in performance. These welcomed improvements have resulted in significant performance/cost advantages in VLSI-implemented systems. Because of the high device counts and limited input/output access that characterize VLSI circuits, conventional testing approaches are often ineffective and insufficient for VLSI circuits. Automatic test pattern generation for sequential circuits is not feasible even for many LSI circuits. Thus, design for testability techniques such as serial scan must be employed, but for VLSI circuits, such techniques still involve large amounts of test pattern generation and simulation efforts, huge volumes of test input/output data, and excessive testing times. Therefore, Testing has become a more difficult problem over the past two to three decades. While the number of I/O pins for most VLSI devices has increased by an order of magnitude, the number of transistors contained in many VLSI devices has increased by four orders of magnitude [1]. Surface mounted components have become the norm, facilitating mounting components on both sides of PCBs as well as allowing micro via's and floating via's on the PCBs. This makes in-circuit testing (where a "bed- of-nails"4bn c c test fixture provides access to the I/O pins of each VLSI device on the PCB) no longer feasible for many PCBs [2].

Device sizes have become larger, creating a larger area to sustain manufacturing defects. Feature sizes (the sizes of transistors and wires) have become smaller, increasing the number and types of defects that can occur during the fabrication process. For example, thinner wire segments lead to more opens in those segments while wire segments that are closer together lead to more shorts between those segments.

There are several important points to be noticed about the partial product array. First, in the most basic formulation (PPA bits generated via logical AND), all the bits are created in parallel; that is; the static delay of each of the bits is equal. Second, the dimensions of the array are functions of the size of the multiplier and multiplicand: the height of the array is proportional to the size of the multiplier, and the width is proportional to the size of the multiplicand. Conventionally ANDing of multiplier and multiplicand is used to generate the partial products. The number of partial products to be generated can be reduced at this stage by using booth encoding schemes. Recoding arranges multiplier bits in groups that select multiples of multiplicand. In this multiplier is partitioned in groups of 3 bits.

II. PREVIOUS WORK

Stroele and Wunderlich [2] has proposed a high- speed low-power multiplier adopting the new SPST implementing approach. In this a Spurious Power Suppression Technique (SPST) signal is applied on a modified Booth encoder which is controlled by a detection unit using an AND gate. *Hasan et al.* [3] has presented the study of multipliers which is done on the basis of their speeds. Multipliers are designed using VHDL and incorporating two adders in the designs *i.e.,* carry look ahead adders and ripple carry adders. On the basis of performance we studied that multiplier with carry look ahead adder is better than the other one with ripple carry. *Cheng et al.* [4] has presented that for designing delay-area efficient adders, heterogeneous adder architecture is proposed which consists of sub-adders of various sizes and different carry propagation schemes. The proposed decomposition into heterogeneous sub-adders allows more design tradeoffs in delay-area optimization. *Fagot et al.* [5] emphasized on the new effective Built-In-Test (BIST) scheme that achieves 100% fault coverage with low area overhead, and without any modification of the circuit under test (CUT), *i.e.,* no test point insertion and the set of patterns generated by a pseudo- random pattern generator. To transform these patterns, a ring architecture composed by a set of masks is used. During on-chip test pattern generation, each mask is successively selected to map the original pattern sequence into a new test sequence. *Huang et.al* [6] proposed a static test approach suitable for built-in-self-test(BIST) of Analog-to- Digital converter Intellectual ProperSQty(IP). Static parameters (INL, DNL, Gain, and Offset) are tested without using test equipment. The proposed BIST structure is applicable for testing models of analog-to-digital converters up to 12- bits of resolution. Comparison results with dynamic test equipment validated the proposed static test approach.

Yamani et al. [7] has focused on the design of a UART chip with embedded Built-In-Self-Test (BIST) architecture using FPGA technology.

III. BIST IMPLEMENTATION FOR BOOTH MULTIPLIER

The paper is divided into five sections: in section I an introduction to multipliers and BIST is discussed. In section II, a short related literature papers are discussed. In section 3, a booth multiplier logic and sub part of this multiplier is designed in HDL. In Sub Section of section 3, BIST implementation in VHDL for the Booth Multiplier is discussed. In section 4 conclusion of the whole work done is discussed.

o CONCEPT OF BOOTH MULTIPLIER

Booth algorithm gives a procedure for multiplying binary integers in signed –2's complement representation.

Step 1: Making the Booth table

I. From the two numbers, pick the number with the smallest difference between a series of consecutive numbers, and make it a multiplier. i.e., 0010 -- From 0 to 0 no change, 0 to 1 one change, 1 to 0 another change, so there are two changes on this one 1100 -- From 1 to 1 no change, 1 to 0 one change, 0 to 0 no change, so there is only one change on this one.
Therefore, multiplication of 2 x (– 4), where 2 10 (0010 2) is the multiplicand and (– 4) 10 (11002) is the multiplier.

II. Let X = 1100 (multiplier) and Let Y = 0010 (multiplicand). Take the 2's complement of Y and call it –Y and –Y = 1110

III. Load the X value in the table.

IV. Load 0 for X-1 value it should be the previous first least significant bit of X

V. Load 0 in U and V rows which will have the product of X and Y at the end of operation.

VI. Make four rows for each cycle; this is because we are multiplying four bits numbers.

Step 2: Booth Algorithm

Booth algorithm requires examination of the multiplier bits, and shifting of the partial product. Prior to the shifting, the multiplicand may be added to partial product, subtracted from the partial product, or left unchanged according to the following rules: Look at the first least significant bits of the multiplier "X", and the previous least significant bits of the multiplier "X - 1".

I.

0	0	Shift only
0	1	Add y to U and shift
1	0	Add –Y to U and Shift
1	1	Shift only

II. Take U & V together and shift arithmetic right shift which preserves the sign bit of 2's complement number. Thus a positive number remains positive, and a negative number remains negative.

III. Shift X circular right shift because this will prevent us from using two registers for the X value. Repeat the same steps until the four cycles are completed. As Shown in table1.

Table1: Complete Shifting for Booth Multiplication

U	V	X	X-1
0000	0000	1100	0
0000	0000	0110	0
0000	0000	0011	0
1111	0000	1001	1
1111	1000	1100	1

IV. VHDL IMPLEMENTATION

○ VHDL Implementation of Booth Multiplier

VHDL implementation of the Booth Multiplier includes the designing of the following component:D-flip flop, Shifter, Booth invertors, Booth coder, Booth Adder which is the combination of the full adders. Complete RTL Schematic of the Booth Multiplier in VHDL is show in the figure 1.

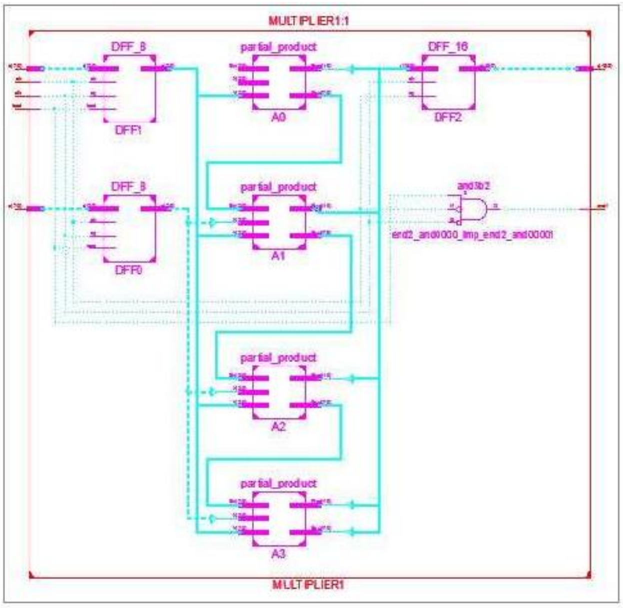

Figure 1: RTL Schematic of the Booth Multiplier.

○ VHDL implementation of Built in self Test for Booth Multipliers

The top Module of Built in self Test (BIST) for the Booth Multiplier includes sub modules:

1. Test Pattern Generator (TPG)
2. Comparator
3. BIST Controller
4. MUX
5. Signal Generator
6. Booth Multiplier

The main part of the BIST is Test Pattern Generator (TPG). In this LFSR Linear Feedback register are mainly used. This is a bank of circuit flip-flops with added testing hardware, which can be configured to make the flip-flops behave like a scan chain, a linear feedback shift register (LFSR) pattern generator, a LFSR-based response compacter, or merely as D flip-flops shown in figure 3.

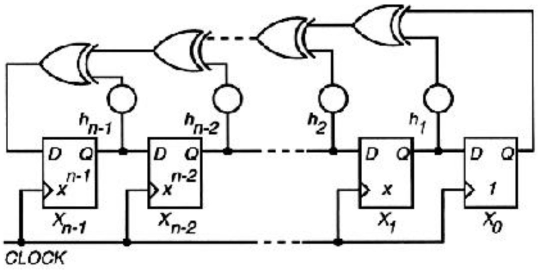

Figure 3: Schematic of LFSR [7]

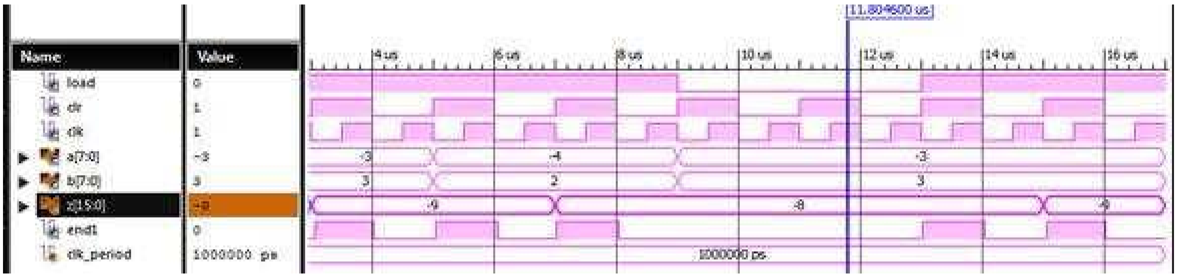

Figure 2: Simulation results of Booth Multiplier

In figure 4 RTL for BIST for Booth Multiplier is given in which a BIST controller is the Mind of the whole design. Which includes 'clk' as the clock input, and a reset switch? Based upon the starting default patterns for every clock tick new test patterns are generated. Comparator is used in the BIST for the comparison of the test patterns with output for the detection of the divergence of the result form the required one. Figure 4 includes the RTL schematic of the same. This includes two inputs 'I1' and 'I2', each of the 17 bits, and an output 'O1'. The "Complete design can work in two modes. One is the Normal Mode, in which Booth Multiplier works as normal and second is the Testing Mode, in which BIST is activated. Selection of these two modes is done with the help of Multiplexer, having two inputs and a selection switch. Here two inputs are; 'inputs' and 'TPG' signals. Signal 'Test' is used as the select pin for the selection of the mode of the operation. Further TPG is the output of the Test Pattern generator. Then embedding the entire component in one design to the BIST implemented with Booth Multipliers.

- **Simulation results of BIST Top module:**

For Normal Mode test='0', load ='1', clr = 2us clock, clk= 1us clk, a="" Testing Mode T=1 sig2comp =crc2comp then bistfail=0 .bistdone=1

No doubt with the BIST development chances for the error for the onsite is reduced. But with this some rise in the total hardware used, that is shown in Table 1. From which it is analysed that BIST architecture is using more hardware in comparison to the simple booth multiplier.

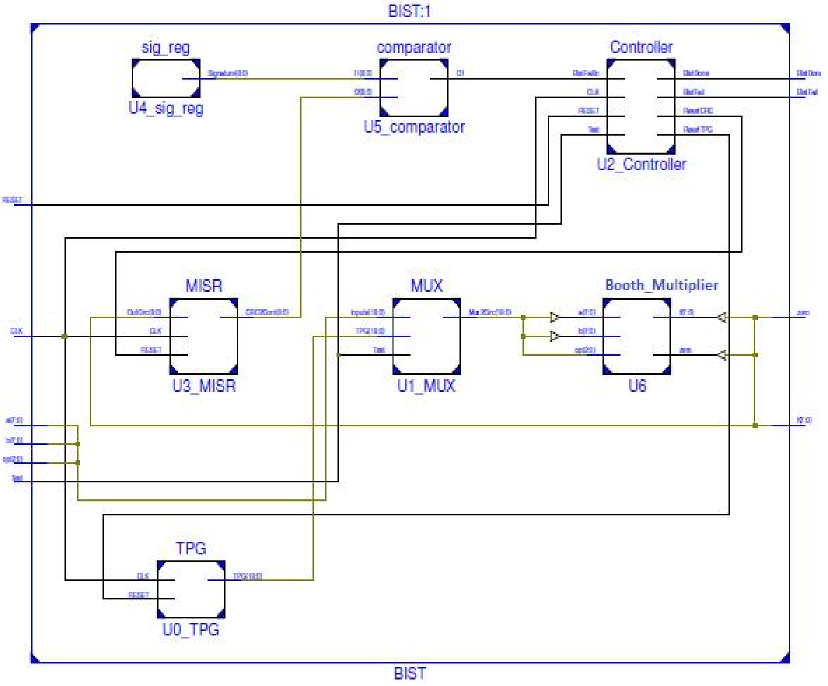

Figure 4: RTL of the BIST for Booth Multiplier

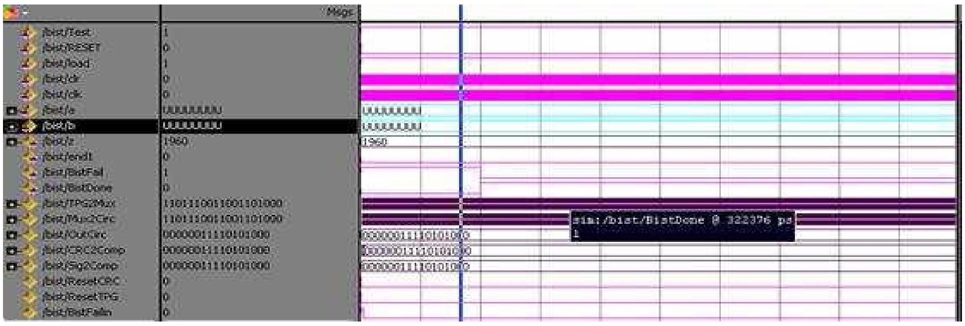

Figure 5: Simulation of the BIST Module in the Normal Mode

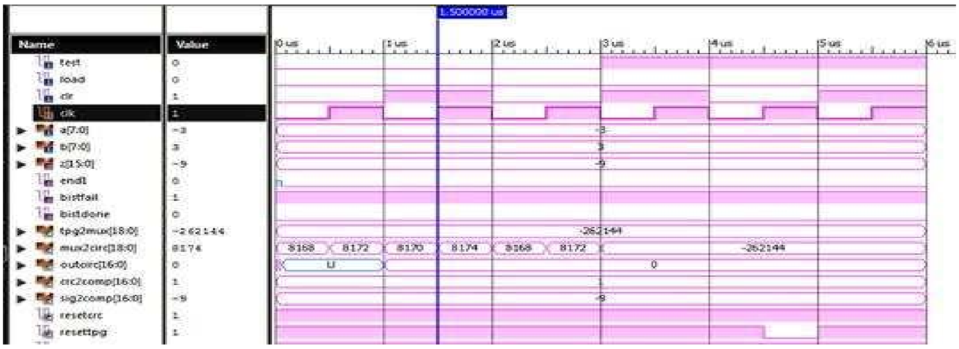

Figure 6: Simulation the BIST Module in the Testing Mode

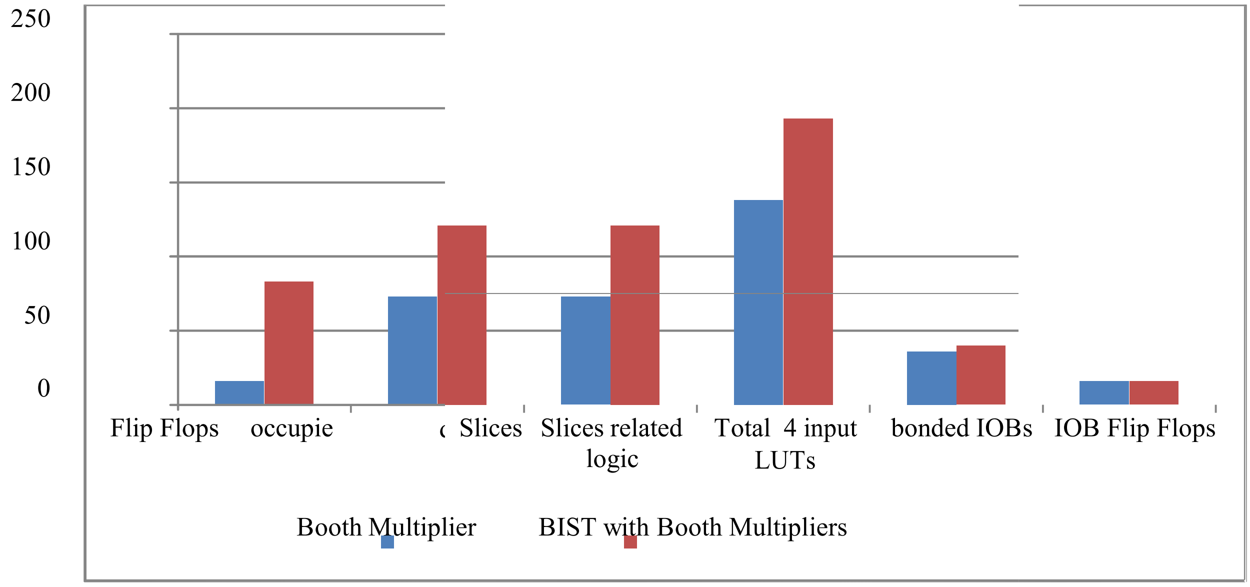

Figure 7: Hardware comparison of the Booth Multiplier and BIST for Booth Multiplier

V. COMPARISON OF PRESENT AND PREVIOUS WORK WITH BIST

Figure 8 shows the graphical representation of Comparison of Present and Previous work With BIST [10].

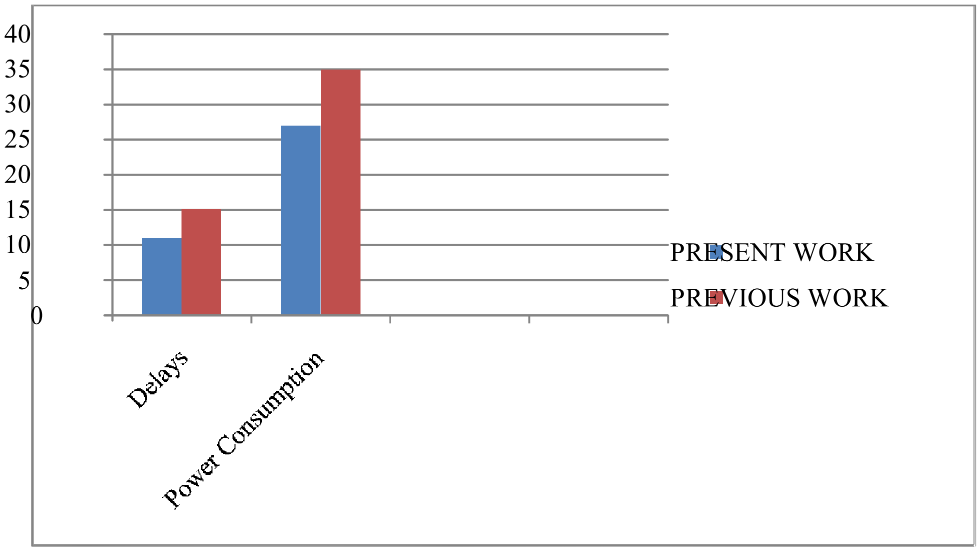

Figure 8: Comparison of Present and Previous work (With BIST)

VI. CONCLUSION

As Built in Self Test is the powerful technique for the testing of the circuits even when CUT (Circuit under test) is in the working. With the help of the Booth Multiplier the target of having a low power and fast multiplier is achieved. But it is not necessary that the designed Multiplier will be work as it was supposed after the fabrication. So to overcome this problem advantage of the BIST is taken in this design. From all the simulation results it is clear that the designed Booth Multiplier by embedding BIST in it, gives the more efficient design than the simple Booth Multiplier, with the extra features of testing in it. But in this design hardware is increased and due to extra activity of testing that dynamic power increased.

Table 2: Comparison of Booth Multiplier with and without BIST

Area utilization	Booth Multiplier	Booth Multiplier with BIST
Number used as Flip Flops	16	83
Number of 4 input LUTs	138	192
Number of occupied Slices	73	121
Number of Slices related logic	73	121
Total Number of 4 input LUTs	138	193
Number of bonded IOBs	36	40
IOB Flip Flops	16	16
Delays (ns)	8.222	10.998
Quiescent power(uw)	23.3	28.4
Dynamic power (uw)	14	43.3
Total power(uw)	37.3	71.7

III. REFERENCES

1. Roth, (1998)," Digital System Design Using VHDL", PWS Publishing Company.
2. Stroele, Wunderlich, (1998). "Hardware- Optimal Test Register `Insertion", in IEEE Transactions On Computer- Aided Design of Integrated Circuits and Systems, Vol.No.6, pp.531-539.
3. Hasan and Aws (2008)," VHDL implementation of the Multipliers based on based on different adders", Faculty of Computer Science and Information Technology, University Of Malaya.
4. Cheng, et al. (2004)," Compactor Independent Direct Diagnosis," Proc. Asian Test Symposium,pp. 204-209.
5. Fagot et al. (1998)" A Ring Architecture Strategry for BIST Test Pattern Generation ," IEEE Asian Test Symposium,pp.418-423.
6. Huang, et al. (2000), "A BIST Scheme for On- Chip ADC and DAC Testing ," Proceeding of Design , Automation & Test in Europe, 216-220.
7. Yamani et al. (2003, 2006) "A VHDL Implementation of UART design with BIST capability", Faculty of Computer Science and Information Technology, University Of Malaya Vol. 19(1), pp.73-86.
8. Bushnell and Agrawal (2000), "Essentials of Electronic Testing for Digital Memory & Mixed Signal VLSI Circuits", Kluwer Academic Publishers, 712 p.
9. Bardell et al. (1987)" Built-In-Test for VLSI: Pseudo Random Techniques", Wiley Interscience, New York.
10. Paruthi P. et al. (2012), "Simulation of IEEE 754 Standard Double Precision Multiplier using Booth Techniques" ISSN: 2248-9622, Vol. 2, Issue 5, September- October 2012, pp.1761-1766.