American-Eurasian Journal of Scientific Research 12 (3): 135-144, 2017
ISSN 1818-6785

DOI: 10.5829/idosi.aejsr.2017.135.144

FPGA Implementation of Low Power Self Testable MIPS Processor

[1]S. Ravi and [2]M. Joseph

[1]The Kavery Engineering College, Salem, Tamilnadu, India
[2]St. Joseph's College of Engineering and Technology, Thanjavur, Tamilnadu, India

Abstract: This paper presents a self testable 32- bit MIPS processor that uses a hybrid approach for low power consumption. The components used for self testability include Linear Feedback Shift Register (LFSR), Built-In Logic Block Observer (BILBO) and Concurrent BILBO (CBILBO). While the BILBO and CBILBO use gated clock technique, the LFSR uses Bit Swapping technique for low power consumption. The results in terms of hardware area and power consumption of the testable MIPS processor are obtained targeting Xilinx Virtex-IV FPGA.

Key words: BILBO · BIST · BS-LFSR · CBILBO · Logic synthesis · LFSR · MIPS · Testability

INTRODUCTION

MIPS is a RISC (Reduced Instruction Set Computer) based Instruction Set Architecture (ISA) [1]. The ISA acts as an interface between top level software like, compiler, Operating System (OS), applications software, etc. and low level hardware that executes the actual instructions. The RISC architecture in general has more number of general purpose registers in the processor to store the operands and results in the registers itself. Hence the MIPS architecture is also called as register based architecture. MIPS processors can be found in applications like, Canon Digital Cameras, Sony Play station2 game consoles, CISCO Routers and many more commercial products in use today [2]. There are two main varieties of MIPS architecture available namely, MIPS32 and MIPS64. This paper considers the basic MIPS32 architecture with five stage pipelining for the implementation of the proposed approach. Testing is an important step in the design process that ensures the reliability of the chip. It is the process in which known set of input patterns are applied to the system under test and the result of the system is compared with the known good responses. This ensures that the manufactured chip is free from defects and only good chips are sold to the customers. Testing also helps to improve the yield of Integrated Circuit (IC) manufacturing process, by identifying manufacturing process stage due to which the defect has occurred. There are several types of testing approaches based on way of generating test patterns (external testing and Built-in self test), timing of the test process (off line testing and online testing), test pattern type (random testing, exhaustive testing and pseudo random testing), etc [3]. The performance of testing approach is evaluated based on the test time, fault coverage, cost of the test equipment, the amount of additional hardware resources introduced in the original circuitry for making the testing process simpler, etc.

The remaining part of the paper is organized as follows: Section 2 presents the motivation behind the proposed testable MIPS architecture. Section 3 presents the work carried out by other researchers related to the proposed work. Section 4 discusses the general 32- bit MIPS processor architecture. The testable MIPS processor with low power test structures considered in the proposed work is presented in Section 5. The simulation and synthesis results obtained using Xilinx ISE tool are presented in Section 6. Finally, Section 7 presents the conclusions and future work of the proposed work.

Motivation: Power consumption and silicon area are the two important factors to be considered for portable devices like Mobile phones, Ipod, Personal Digital Assistants (PDAs), Laptop computers, etc. The lifetime of the battery in these devices depends on the power consumption in the circuit. The microprocessors consume electrical energy, in which some energy is dissipated for switching while the remaining energy is lost in the form of thermal radiation due to random movement of charge carriers in the device. The major sources of power consumption in a processor include memory, instruction and data caches, registers, clock networks and the

Corresponding Author: S. Ravi, The Kavery Engineering College, Salem, Tamilnadu, India.

remaining combinational cells [4]. The overall power consumption of microprocessor chip increases by 22 % every year [5]. This leads to increased heat generation in the chip. Hence, power optimization in datapath involves, reducing power consumption of register files, functional units like adder, ALU and instruction and data caches. Power consumption during test mode is twice that of the normal mode of operation of a circuit. This is due to reduced correlations between the successive test patterns applied to the circuit under test and the power consumption due to added test circuitry. Also, during test mode most of the modules in the system are activated, where as the system mode that operates on power saving mode activates only few modules in the system [6]. Conversely, reducing power consumption also leads to reduced heat generated in the chip that in turn lowers the cost of system maintenance in terms of packaging and cooling systems. Hence low power testing approaches are the need of the hour and this paper brings one such solution to this problem by introducing low power test structures in the MIPS architecture.

Literature Survey: A survey on low power testing techniques applied at the circuit, RTL and system level is presented in [7]. Since majority of the modern electronic systems are mobile devices and are battery operated, low power test structures are becoming essential modules for both FPGA and ASIC based systems. Several techniques have been proposed for the optimization of MIPS architecture in terms of silicon area and power consumption. Power reduction in pipelined MIPS processor by eliminating / bypassing unwanted pipeline stage process can be found in [8]. Also, clock gating technique is applied in [8] for the unused pipeline stages during instruction execution for reducing power consumption in the processor. A low power BIST technique using modified clock scheme for the Linear Feedback Shift register (LFSR) is presented in [6]. A Low Power LFSR (LP-LFSR) based test pattern generation approach in combination with binary counter and Binary-to Gray code converter is presented in [9]. However, this approach reduces power consumption for test pattern generation at the expense of hardware complexity and silicon area. Power consumption due to clock sources of a system plays a major part and is due to high switching or toggling activity and capacitive loading of the clock networks. Reducing power consumption in master slave flip-flop by deactivating clock signal when there are no data transitions is presented in [10]. The proposed approach uses the same technique on each BILBO and CBILBO flip-flops, forming gated clock flip-flops based CBILBO register. Testing of modern Microprocessors becomes more complex due to increased device density, large amounts of internal cache memory, presence of additional registers for improving throughput, pipelining registers, for example. The testability of the hardware can be improved by providing easier control or observation of internal nodes in the circuit. A feasible solution to this problem is to use Built-in Self Test (BIST) technique. It provides several important advantages, like no need of external test equipment, using internal registers to operate in normal and test modes when testing of internal functional units in the system, etc. In addition, the BIST provides a signature analyzer that compresses the actual output from the system under test and hence it reduces the size of the memory required to store the good responses. Since MIPS processor is a larger design, it is very difficult to test it exhaustively with all possible test patterns. Hence, pseudo random test patterns are generated that are repeatable and tests the design to maximum fault coverage [11].

MIPS Processor Architecture: The proposed testable MIPS processor is implemented by modifying the 32- bit MIPS processor and the simplified architecture of the MIPS processor is as shown in Fig. 1.

The MIPS processor consists of mainly two units namely, datapath and control unit [12]. While the datapath performs the intended operation and storage, the timely behavior of the datapath is controlled by the control unit. Again, the control unit is available as two separate units namely, main control and ALU control. In this figure, the main control unit, which sends the control signals to ALU control and other datapath elements of the MIPS processor is omitted from the diagram to reduce the complexity. The ALU control generates six different control signals based on the input control signal received from the main control unit that decides the operations to be performed by the ALU. The list of operations performed by the ALU and its input control signal values are shown in Table 1. There are four different multiplexers available in the MIPS processor for routing the signal between various units in the datapath and these multiplexers are also not shown in Figure for simplicity.

The datapath comprises of functional units like ALU, adders and sign extend and storage devices such as register file, instruction and data memory and finally a special purpose register called Program Counter (PC). Here, binary value PC points to the address of the current instruction in the instruction memory and is updated

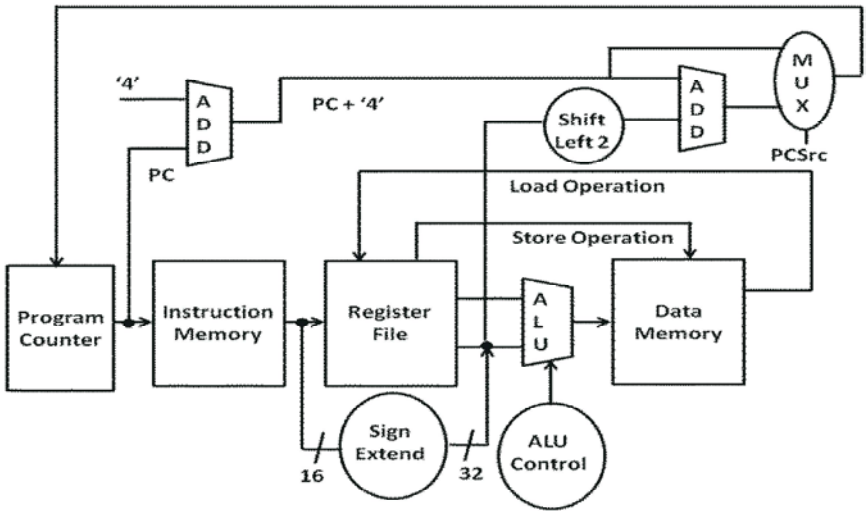

Fig. 1: A simplified MIPS Processor Architecture

Table 1: ALU Function Table

S. No	ALU Control Input	Operation
1	0000	AND
2	0001	OR
3	0010	Add
4	0110	Subtract
5	0111	Set on Less Than
6	1100	NOR

during the program execution for fetching the next instruction to be executed. Instruction memory stores the instructions of the program under execution. Register file contains 32 general purpose registers, each of size 32 bits. Two additional adders are provided to calculate the next instruction address for the PC. Though this addition can be performed by the ALU itself, the additional adders helps to implement the pipelining by performing address calculation in a separate unit.

Data memory stores the operands and the results of the operation. This memory can be accessed using load and store instructions of the MIPS instruction set. Multiplexers in the datapath provide the signal routing path between functional elements in the datapath, depending on the instruction under execution. The sign extend unit converts 16- bit data into 32- bit which acts as one of the source operands for ALU for arithmetic and logical operations, or as an input for the adder unit that calculates address for Jump instructions.

Testable MIPS Processor Architecture: The proposed testable MIPS processor is implemented by modifying ALU, register file and memory by their respective testable versions. The register file and memory can be applied with the same procedure for their testing. Figure 2 shows the testable version of ALU and memory modules in the MIPS processor. In the case of testable ALU, one of the registers acts as TPG, while the other act as TPG and LFSR simultaneously using CBILBO register.

Testing of memories in the processor plays a vital role, since it occupies most of the chip area and expected to increase up to 94% by the year 2014 [13]. Also, memory is the most defect sensitive part of the system. The capacity of DRAM chip increases 25 to 40% per year and the capacity of flash memory increases 50 to 60% per year [12]. Hence, it is very much essential to consider testing of memory in a computer. The overall memory test arrangement is shown in Figure 2. It uses up/down LFSR that acts as address generator and a mutual comparator that compares the results from program and data memory. A major advantage of the mutual comparator approach is that it does not require good machine response to be stored in a memory or generated.

Test Structures for MIPS Processor: The MIPS processor becomes testable due to the addition of test hardware in the native MIPS architecture. Three major components used in the proposed testable MIPS processor include, (i) Normal LFSR (ii) Up / Down LFSR (iii) BILBO and (iv) Concurrent BILBO.

The components have two modes of operation namely, normal mode and test mode. During the normal mode they perform normal operations according to the circuit functionality. During test mode, these components used either to route the test data among various FUs in the design, or act as test pattern generator (TPG), test response analyzer (TRA), sometimes both TPG and TRA.

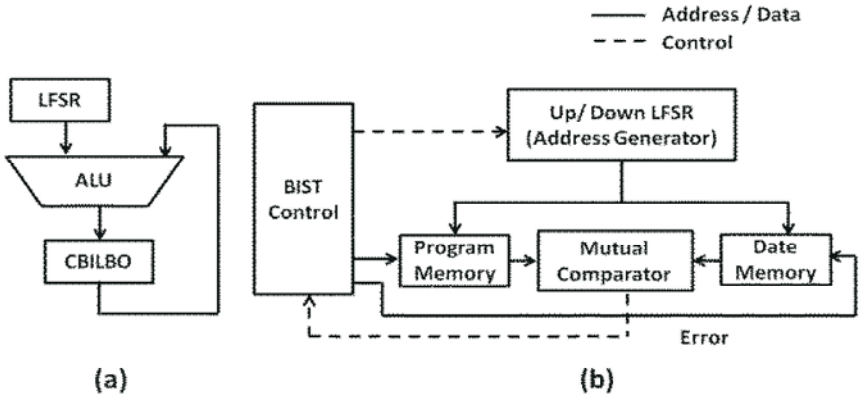

Fig. 2: Testable processor (a) Testable ALU and (b) Testable memory

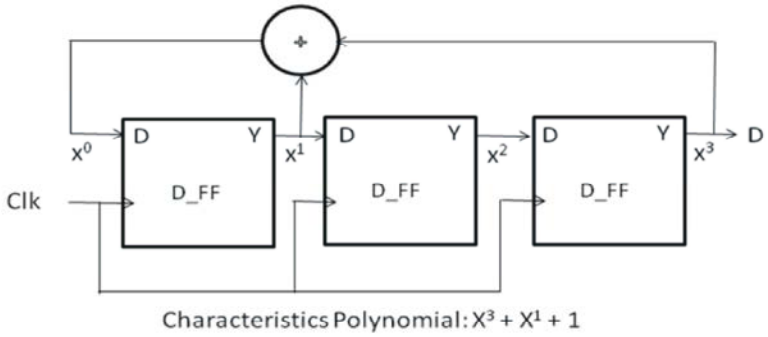

Fig. 3: A three stage LFSR and its characteristics polynomial

Linear Feedback Shift Register: The Linear feedback shift register (LFSR) is the most commonly used circuit for testing a portion of combinational circuitry by producing the input signals as a pseudo random pattern generation (PRPG) and as a test response analyzer (TRA) to observe the output signals. The LFSR is constructed from a set of flip-flops connected in serial fashion as shown in Fig. 3.

The XOR of particular outputs are fed back to the input of the LFSR. An n-bit LFSR will cycle through 2^n-1 states before repeating the sequence. The LFSR does not traverse all 2^n states, since LFSR will be locked in all 0 state as the XOR operation cannot make new state transitions [14]. The feedback connections or tapping points in an LFSR is represented by a polynomial function called characteristics polynomial. The feedback connections decide the number of possible binary combinations in the flip-flops, called length of the sequence [15].

Up / Down LFSR: It is a preferred Memory BIST pattern generator due to test patterns generated can also detect address decoder faults in the memory circuit. In the case of Memory testing, LFSR acts as address generator. This condition satisfies the March test, one of the popular memory testing approaches being used today. It is combined with mutual comparator circuit to test both program and data memory of the MIPS processor simultaneously. The logic diagram of Up / down LFSR and mutual comparator are shown in Figures 4 and 5 respectively.

This structure introduces a NOR gate that takes the output of all stages except the LSB bit (feedback value) and a two input XOR gate that takes the output of NOR gate and LSB bit to make the circuit to generate all 0 pattern. The all 0 state occurs after 0000001 state [16]. This arrangement is called Complete Feedback Shift Register (CFSR), also called Complete LFSR.

The characteristics polynomial of normal LFSR is $G(x) = X^3 + X + 1$ and that of the inverse LFSR is $G(x) = X^3 + X^2 + 1$. Both normal and inverse LFSR combined to act as up down LFSR with an additional control input to select between the mode of operation. The pattern sequence generated by these LFSR is shown in Table 2.

The mutual comparator is used to support multiple memories testing simultaneously [17]. In the proposed work, two memories are available namely program and data memory. Both can be tested simultaneously by applying the same address to both memory units.

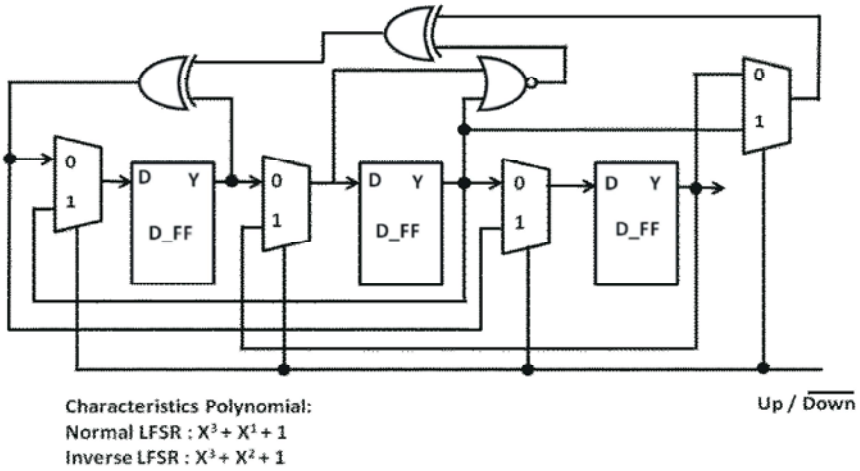

Fig. 4: 3- Bit Up – Down LFSR

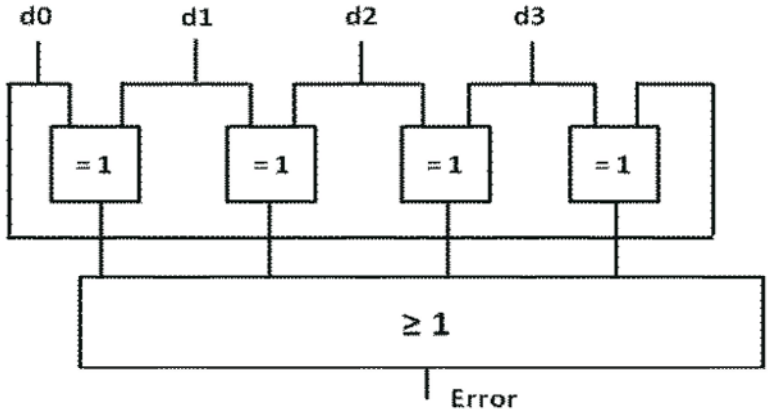

Fig. 5: Mutual Comparator

Table 2: Up / Down LFSR Pattern sequences

Clk	Up counting	Down Counting
Initial State	000	000
1	100	001
2	110	010
3	111	101
4	011	011
5	101	111
6	010	110
7	001	100
8	000 (Starting value)	000 (Starting value)

CBILBO: As the name suggests, the concurrent BILBO (CBILBO) can acts as both PRPG and TRA simultaneously. This component is used to assign hardware for register that has to act as both test pattern generator and signature analyser. These types of registers are formed when there is a self-loop in the scheduled data flow graph (SDFG), which is generated after the scheduling process. Since self-loops increase number of test patterns required to test it, it is very much essential to replace it by CBILBO register. The CBILBO is the extension of BILBO, having additional register that allows PRPG and TRA to operate independently, as shown in Fig. 6.

The mode select signals (B2, B1) decide the operation of CBILBO, as shown in Table 3.

Low Power Test Structures

Bit Swapping LFSR: The Bit-Swapping LFSR (BS-LFSR), is constructed using a normal LFSR and n number of 2-to-1 multiplexers, where n is the size of the LFSR. The multiplexers connected in the output lines of the LFSR that performs the bit swapping operation. A 3- stage Bit swapping LFSR is as shown in Fig. 7.

Here, the LFSR output Q [0] decides the swapping operation. When it is 1, the LFSR output is taken without any swap between adjacent cells. On the other hand, when Q[0] is 0, the two flip-flops in adjacent positions

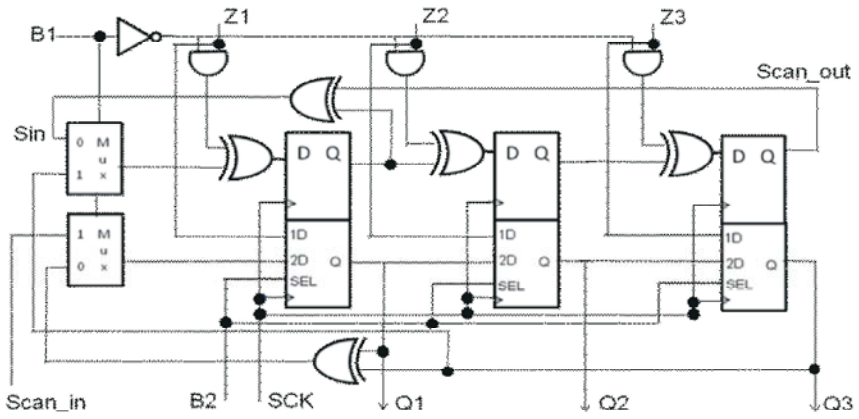

Fig. 6: 3-Bit CBILBO

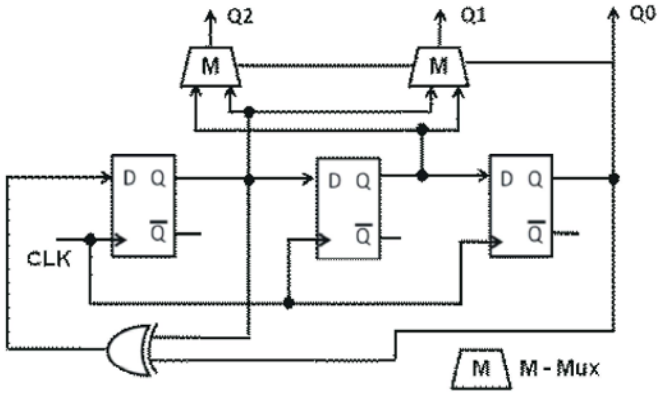

Fig. 7: 3- stage Bit Swapping LFSR

Table 3: Operation modes of CBILBO

Mode Select Bits		Operation Mode
B2	B1	
0	0	Normal
0	1	Scan
1	0	On-line Checking
1	1	Mixed TPG and TRA

(except the flip-flop in the LSB position) interchange their content [18]. For instance, in the above diagram when q[0] = 0 means, the output of the flip-flop in the MSB position is given to its adjacent flip-flop result, while the flip-flop in the adjacent position result is given to the output of the MSB flip-flop. This structure brings out an important property of the BSLFSR. This property states that, if two cells are connected with each other, then the probability that they have the same value at any clock cycle is 0.75. (In a conventional LFSR where the transition probability is 0.5, two adjacent cells will have the same value in 50% of the clocks and different values in 50% of the clocks; for a BS-LFSR that reduces the number of transition of an LFSR by 50%, the transition probability is 0.25 and hence, two adjacent cells will have the same value in 75% of the clock cycles). Thus, for two connected cells (cells j and k), if we apply a sufficient number of test vectors to the CUT, then the values of cells j and k are similar in 75% of the applied vectors.

The number of multiplexers increases as that of the size of the LFSR. However, the important aspect of selection of this approach is that multiplexers are very commonly available resources in all types of FPGA structures. This can be utilized for reducing the number of switching transients in the LFSR output which in turn reduces the power consumption. It is observed that the total number of switching transients for a complete cycle, the normal LFSR takes 12 transients while the BS LFSR takes only 10 transients. As the power consumption in a digital circuit is directly proportional to the number of switching transients [19], the BSLFSR approach reduces the power consumption in the circuit.

Gated clock BILBO: A BILBO flip-flop with gated clock technique applied on its clock input is called gated clock BILBO flip-flop. Figure 8 shows the gated clock BILBO flip-flop with additional logic gates to enable clock signal

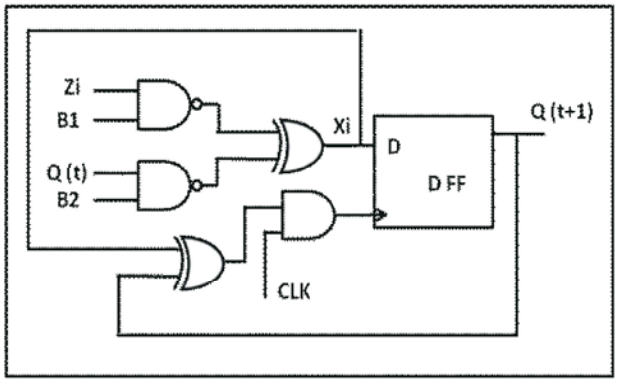

Fig. 8: Gated Clock BILBO Flip-flop

input of the flip-flop. This flip-flop changes its state, only when the present state and next state are different. A group of gated clock BILBO flip-flops are combined together to form gated clock BILBO register.

A similar technique is applied for CBILBO register to save clock power. Since the clock power consumption in a complex system used today is about 25%. Hence, maximum power can be saved by deactivating clock inputs of the flip-flop. The only limitation with this approach is that redundant hardware is introduced with each flip-flop (Two logic gates) to deactivate the clock.

RESULTS AND DISCUSSION

This chapter discusses the simulation and synthesis results obtained for the proposed testable MIPS processor. This processor is designed using Verilog HDL and the synthesized netlist file output is generated using Xilinx ISE Design Suite 14.2. This performs the foreground tasks such as program entry, functional simulation and background tasks such as mapping, floor plan, Place and route and to generate bit steam file for FPGA implementation. For synthesis, the target hardware used is in this work is Virtex-IV FPGA from Xilinx Inc. It is an SRAM based FPGA with in-system configuration. This FPGA has some unique features like on-chip precision controlled output impedance, active interconnect architecture, protection of chip designs with bit-stream encryption [20].

Simulation Results: The simulation result of 32-bit LFSR is shown in Figure 9. The LFSR is load with the seed value of 0000000000010100. Now, the LFSR generates various possible binary combinations that acts act test pattern for the module under test.

The simulation result of 32-bit CBILBO register is shown in Figure 10. While BILBO register has a reset mode to initialize the flip-flops in the register, CBILBO does not have reset mode. Instead, reset is done using the reset signal available for flip-flop control input.

The simulation result of 32-bit BS-LFSR is shown in Figure 11. The BS-LFSR is load with the same seed value as that of LFSR. The simulation result of 32-bit Up / Down LFSR is shown in Figure 12.

Synthesis Results: The target device used is XC4VLX15-FF676-12. The area overhead due to additional components for testability enhancement can be calculated using the following equation:

$$AreaOverhead\% = \frac{\text{Circuit with redundant test hardwar} - \text{Circuit without redundant test hardware}}{\text{Circuit without redundant test hardware}} * 100$$

Table 4 presents the hardware utilization of 32 bit LFSR and its comparison with normal register, Bit-swapping LFSR and Up/Down LFSR of 32 bits size. From table it is observed that, BS-LFSR takes 90.90 % area overhead when compared with LFSR, while the LFSR takes area overhead of 3.12 % when compared to normal LFSR.

Table 5 presents the hardware utilization of BILBO register and its comparison with gated clock BILBO and normal register. From the values obtained in Table 5, it is estimated that BILBO register takes 9.37 % hardware overhead than the normal register. The gated clock BILBO takes 88.57 % hardware overhead than BILBO register.

Table 6 presents hardware utilization of CBILBO register and it is estimated that it takes 103% hardware area overhead than traditional register. In addition takes more number of flip-flops than the normal register.

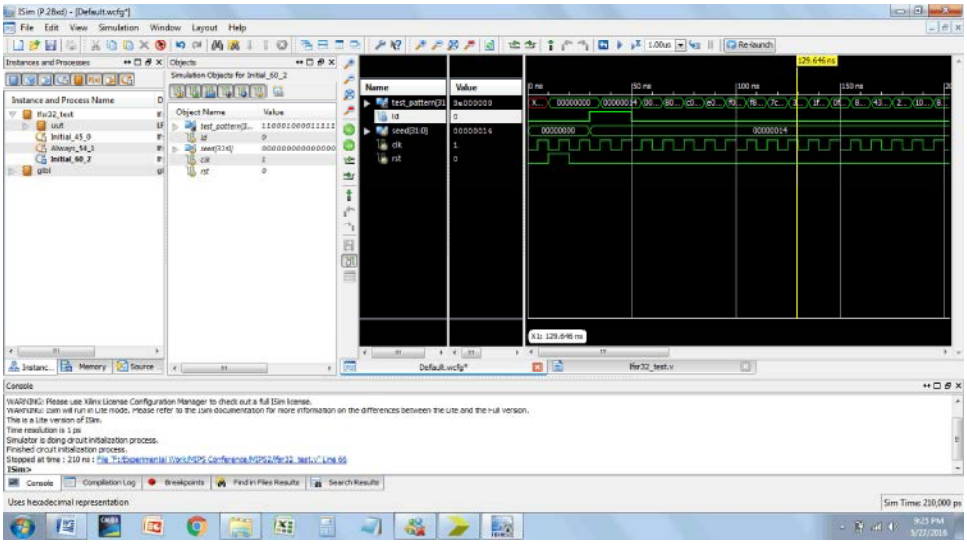

Fig. 9: Simulation Result of 32-bit LFSR

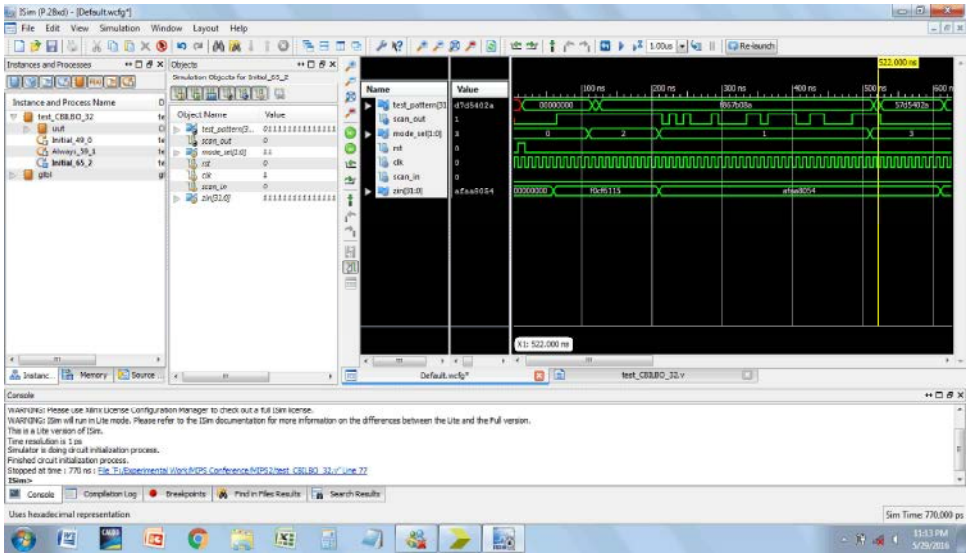

Fig. 10: Simulation Result of 32-bit CBILBO

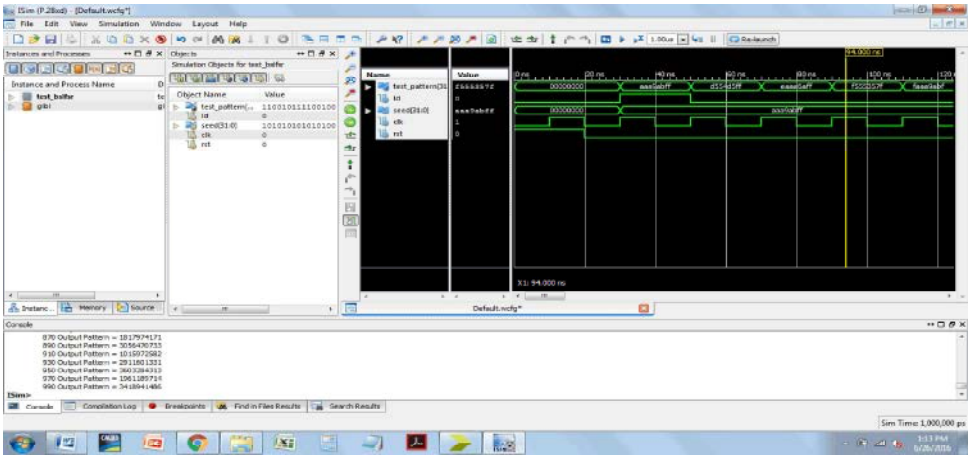

Fig. 11: Simulation Result of 32-bit BS-LFSR

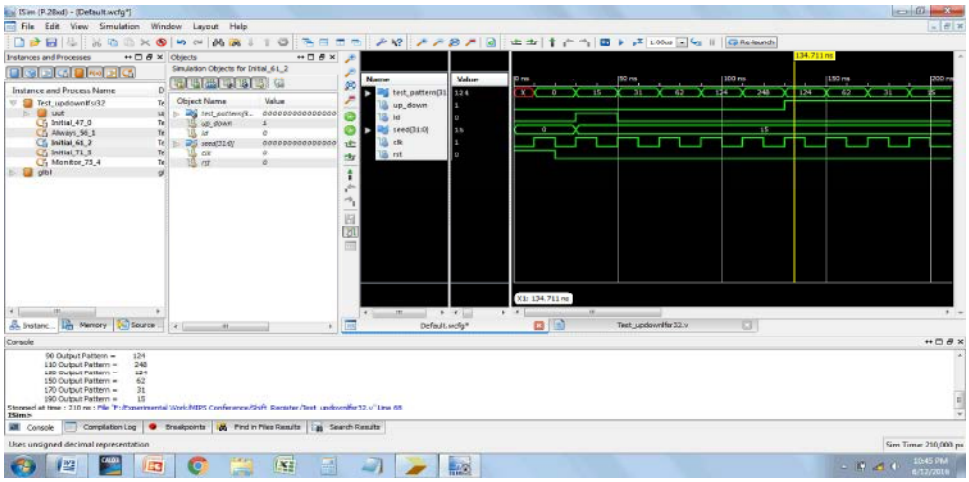

Fig. 12: Simulation Result of 32-bit Up / Down LFSR

Table 4: Device Utilization for 32-bit LFSR

S. No	Metric	Normal Shift Register	LFSR	BS LFSR	Up / Down LFSR
1	Slices	18	19	36	41
2	Slice Flip-Flops	32	32	32	32
3	4 I/p LUTs	32	33	63	81
4	No. of IOs	36	67	67	68

Table 5: Device Utilization for 32-bit BILBO

S. No	Metric	Normal Shift Register	BILBO	Gated Clock BILBO
1	Slices	18	20	38
2	IOB Flip-Flops	32	32	32
3	4 I/p LUTs	32	35	66
4	No. of IOs	36	70	70

Table 6: Device Utilization for 32-bit CBILBO

S. No	Metric	Normal Shift Register	CBILBO	Gated Clock BILBO
1	Slices	18	35	53
2	IOB Flip-Flops	32	37	37
3	4 I/p LUTs	32	65	96
4	No. of IOs	36	70	70

Table 7: Device Utilization for 32-bit CPU

S. No	Metric	Normal CPU	Testable CPU	Low power Testable CPU
1	Slices	1258	1307	1342
2	IOB Flip-Flops	494	500	500
3	4 I/p LUTs	2384	2483	2544
4	No. of IOs	1	35	35

Table 7 presents the hardware utilization of CPU and the same for testable version of MIPS processor and its low power testable version. When compared to normal CPU, testable version of CPU takes 4.15% hardware overhead. Also the low power testable MIPS processor takes only 6.71% hardware overhead than the normal CPU.

CONCLUSION

This paper presented a testable MIPS processor implementation on Virtex-IV FPGA from Xilinx Inc. The total hardware overhead due to low power testability is 6.17%. The flip-flops in the BILBO and CBILBO register are incorporated with clock-gating technique for low

power consumption. The testability is incorporated using LFSR, BILBO and CBILBO. Also, various other low power techniques like weighted LFSR, dual speed LFSR structures can be developed in order to reduce the power consumption in the testable MIPS processor [18]. In addition, this paper has presented the results only for the testable datapath and the future work is to implement testable control unit which completes the implementation of the testable MIPS processor.

REFERENCES

1. Mohit N. Topiwala and N. Saraswathi. 2014. Implementation of a 32-bit MIPS Based RISC Processor using Cadence. In Proceedings of IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), pp: 979-983.
2. Marri Mounika and Aleti Shankar, 2013. Design & Implementation of 32-bit RISC (MIPS) Processor. International Journal of Engineering Trends and Technology, 4, 10 (Oct. 2013), 4466-4474.
3. Miron Abromovici, Melvin A. Breuer and Arthur D. Friedman, 2001. "Digital Systems Testing and testable VLSI Design", Jaico Publishers, First Edition, 2001.
4. Alex Voica, 2014. How to reduce dynamic power by 50 (Sept. 2014). Retrieved June 30, 2016 from http://blog.imgtec.com/mips-processors/how-to-reduce-dynamic-power-for-a-mips-cpu.
5. wikipedia, 2016. Low-power electronics. (April 2016). Retrieved June 30, 2016 from https://en.wikipedia.org/wiki/Low-power electronics
6. Girard, P., L. Guiller, C. Landrault, S. Pravossoudovitch and H.J. Wunderlich, 2001. "A modified Clock Scheme for a Low Power BIST Test Pattern Generator, " in Proc. VLSI Test Symp. (VTS'01), pp: 306-311.
7. Girard, P., 2002. "Survey of Low power testing of VLSI circuits", IEEE Design and Test of Computers, 19(3): 80-90.
8. Manan Parikh, Mayuresh Dawoo, Pallavi Manjunath, Prashant Awasthi and Fahad Usmani, 2014. "Design of Power efficient MIPS Processor, " International Journal of research in Engineering and Applied Sciences (IJREAS), 2(1): 19-22.
9. Supriya, K. and B. Rekha, 2013. "Implementation of Low power Test Pattern Generator using LFSR", International Journal of Science and Research (IJSR), 2(8): 165-170.
10. Strollo, A.G.M. and D. De Caro, 2000. "Low Power flip-flop with clock gating on master and slave latches", Electronic Letters, 17th February 2000, 36(4): 294-295.
11. Jessica Hui-Chun Tseng, 1999. Energy-Efficient Register File Design. Master's thesis. Massachusetts Institute of Technology.
12. John L. Hennessy and David A. Patterson, 2012. Computer Architecture A Quantitative Approach (5th ed.). Morgan Kauffmann / Elsevier Publishers Inc.
13. ITRS, 2003. International technology Roadmap for Semiconductors. (Dec. 2003).
14. Michael L. Bushnell and Vishwani D. Agrawal, 2002. Essentials of Electronic Testing: For Digital, Memory and Mixed-signal VLSI Circuits. Kluwer Academic Publishers.
15. Coppersmith, D., Z. Barzilai and A.L. Rosenberg, 1983. Exhaustive Generation of Bit Patterns with Applications to VLSI Self-Testing. IEEE Transactions on Comput., 32, 2 (Feb. 1983), 190-194. DOI:http://dx.doi.org/10.1109/TC.1983.1676202
16. Charles E. Stroud, 2002. "A Designer's Guide to Built-in Self test", Kluwer Academic Publishers", 2002, ISBN: 1-4020-7050-0.
17. Van de Goor, A.J., 1991. Testing Semiconductor Memories: Theory and Practice. Wiley Sons, Inc.
18. Nithya, R., M. Paviya, A. Poornima Devi and Selvaraj Ravi, 2015. "A performance comparison of Low Power LFSR Structures", International Journal of Recent Research in Science, Engineering and Technology, 1(1): 23-35.
19. Neil, H.E. Weste and K. Eshraghian, 1994. Principles of CMOS VLSI Design- A System Perspective (2nd ed.). Addison Wesley.
20. xilinx.com. 2010. Virtex-IV Family Overview. (Aug. 2010). Retrieved June 30, 2016 from http://www.xilinx.com/support/documentation/data sheets/ds112.pdf.