International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN : 2278-800X, www.ijerd.com
Volume 5, Issue 1 (November 2012), PP. 14-17

VHDL Implementation of ALU with Built In Self Test Technique

Navdeep Kaur[1], Neeru Malhotra[2],
[1, 2] DAV Institute of Engineering and Technology, Jalandhar, Punjab

Abstract:- In today's Integrated Circuits (ICs), Built-In-Self Test (BIST) is becoming increasingly important as designs become more complicated. BIST is a design technique that allows a circuit to test itself Test pattern generator (TPG) using Linear Feedback Shift Resister (LFSR) is proposed which is more suitable for BIST architecture.In this paper we have design ALU (arithmetic and logic unit) in VHDL with BIST capability and compared the area overhead of with and without BIST.

Keywords:- Built in Self Test, VLSI Testing, Test Pattern Generation, Arithmetic and Logic Unit, LFSR

I. INTRODUCTION

Testing of integrated circuits (ICs) is of crucial importance to ensure a high level of quality in product functionality in both commercially and privately produced products. The impact of testing affects areas of manufacturing as well as those involved in design. This desire to attain a high quality level must be tempered with the cost and time involved in this process. These two design considerations are at constant odds. It is with both goals in mind (effectiveness vs. cost/time) that Built-In-Self Test (BIST) has become a major design consideration in Design-For-Testability (DFT) methods. As digital systems become more complex, they become much harder and more expensive to test. One solution to this problem is to add logic to the IC so that it can test itself. This is referred to as "Built in self Test" (BIST). BIST approach is beneficial in many ways. First, it can reduce dependency on external Automatic Test Equipment (ATE). This aspect impacts the cost/time constraint because the ATE will be utilized less by the current design and can be used elsewhere or on the other Devices [1].

An arithmetic and logic unit (ALU) is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers. The processors found inside modern CPUs and graphics processing units (GPUs) accommodate very powerful and very complex ALUs; a single component may contain a number of ALUs. Most of a processor's operations are performed by one or more ALUs. An ALU loads data from input registers, an external Control Unit then tells the ALU what operation to perform on that data, and then the ALU stores its result into an output register. The Control Unit is responsible for moving the processed data between these registers, ALU and memory. Most ALUs can perform the following operations:

Bitwise logic operations (AND, NOT, OR, XOR)

Integer arithmetic operations (addition, subtraction, and sometimes multiplication and division, though this is more expensive)

Bit-shifting operations (shifting or rotating a word by a specified number of bits to the left or right, with or without sign extension). Shifts can be seen as multiplications and divisions by a power of two.

This paper focuses on implementation of ALU with BIST capability using LFSR techniques on Field Programmable Gate Array (FPGA) technology.

Bist architecture

BIST is an on-chip test logic that is utilized to test the functional logic of a chip. A generic approach to BIST is shown in Figure 1. BIST solution consists of a Test Pattern Generator (TPG), a circuit to be tested, a way to analyze the results, and a way to compress those results for simplicity and handling. With the rapid increase in the design complexity, BIST has become a major design consideration in Design-For-Testability (DFT) methods and is becoming increasingly important in today's state of the art SoCs. Achieving high fault coverage while maintaining an acceptable design overhead and keeping the test time within limits is of utmost importance. BIST help to meet the desired goals. The brief introductions of BIST architecture component are given below.

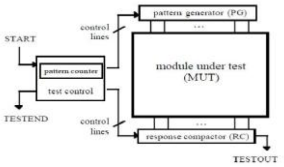

Fig. 1: A generic approach to BIST

Circuit Under Test (CUT): It is the portion of the circuit tested in BIST mode. It can be sequential, combinational or a memory. It is delimited by their Primary Input (PI) and Primary Output (PO).
Test Pattern Generator (TPG): It generates the test patterns for the CUT. It is a dedicated circuit or a microprocessor. The patterns may be generated in pseudorandom or deterministically.

Test Response Analysis (TRA): It analyses the value sequence on PO and compares it with the expected output.
BIST Controller Unit (BCU): It controls the test execution; it manages the TPG, TRA and reconfigures the CUT and the multiplexer. It is activated by the Normal/Test signal and generates a Go/No go.

II. RELATED WORK

Many researchers have been working on the BIST
implementation and GCD related research some of them are given below:

Crouch et al. [7] has discussed the main point is Built-In-Self Test (BIST) architecture for sequential circuits based on cellular Automata (CA).Yamani et al. [2][11], the BIST technique incorporated into the UART design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements. Devi et al.[9], emphasized on the implementation of GCD (greatest common divider) Processor with Built- in-Self- Test feature. In this paper, designing GCD (greatest common divider) processors in VHDL with BIST capability and compared the area overhead off with and without BIST . Hegde Suma T.et.al [4] emphasis on Design and Implementation of ALU using Redundant Binary Signed.

Alu implementation with bist feature

A software tool is used which automatically generates built-in self-test blocks into VHDL models of digital circuits by giving the suitable values of initial seed and primitive polynomial in TPG block. The code is generated for BIST, when we insert the code of ALU into CUT, a generated code is synthesized in the Xilinx web pack 12.4 for the Spartan 3e devices. The hardware summary is obtained for each method implementation log file of Xilinx 12.4 project navigator .The RTL view of the ALU with BIST capability is shown in Figure.

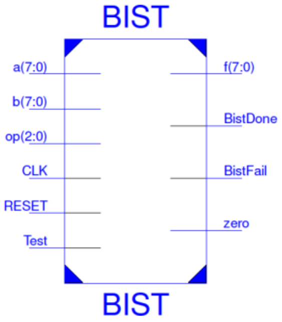

Fig. 2: RTL view of the BIST

It has the following parts Signal Register, Comparator, Controller, MISR, MUX,ALU and Test Pattern Genrator.All these parts forms the BIST. In this MUX is used to select operation, then operation is selected and corresponding operation is performed by the ALU. After this time Test Pattern is also selected by the MUX. Test Pattern output and ALU output is compared by the comparator after passing through the MISR. All the functions are controlled by the controller. Controller will select weather to test the ALU depending upon TEST signal. Controller also provides BIST Fail or BIST Pass depending upon the testing is successful or not.

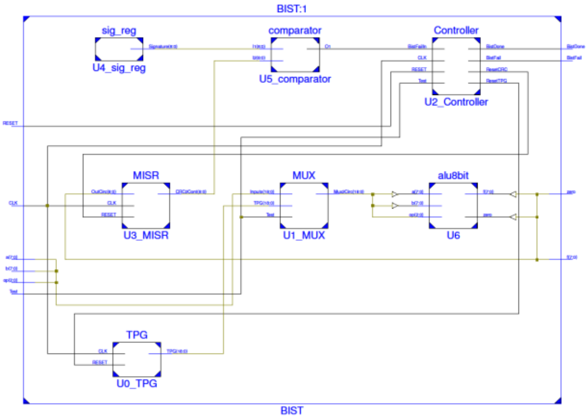

Fig. 3: RTL view of the ALU with BIST

RTL view of the BIST with proper placing and routing. It has following pins: - inputs for input signals, operation input for selecting the oration like AND,OR , addition or subtraction etc. RESET signal is used to reset the bist operations .

TEST signal will start the testing. If TEST =1, then there will be the testing of ALU or if BIST=0 then BIST will not work for testing. IF the testing is done then BistDone=1 otherwise BistFail=1.

Simulation of simple ALU

ALU has two inputs each of 8-bit on which data to be processed is given. Signal "op" is used to select the particular operation by the ALU, that is "op" is used as a select signal here. Signal "f" is the output to hold the final result.Fig.4 shows the simulation of simple alu.

Simulation of Complete BIST for ALU

In figure 5 simulation of complete BIST for ALU are shown, which is the combination all above four sections that are mux, ALU, TPG and controller section."Bist/a", "bist/b" , "op", "bist/test", clk and reset are the inputs. Signal"f","bistdone", "bistfail" , and "zero" are the outputs.

III. RESULTS OF HARDWARE USED

The device utilization Summary shows the total hardware used in the designed circuit. This report gives us information about no. of slices occupied(FF) , no. of 4 input LUTs, no. of bonded IOBs and no. of GCLKs used in the designed circuit . Report is given in the following order:

Table 1: Hardware used for BIST for ALU

Logic utilization	Used	Available	utilization
Number of slices	55	2448	2%
Number of slice flip-flops	43	4896	0%
Number of 4input LUTs	102	4896	2%
Number of bonded IOBs	33	172	19%
Number of GCLKs	1	24	45

IV. CONCLUSION

In the last the result are analyzed and compared for parameters like No. of slices, Flip-Flops used etc. The different methods of the of the test pattern generation effects the cost of the hardware and the speed of the circuits, in these cases some hardware is increased but the overall cost of the ATE is be reduced. The increased in the Area may be compensated with the costing of the ATE.

A. Simulation of simple AL

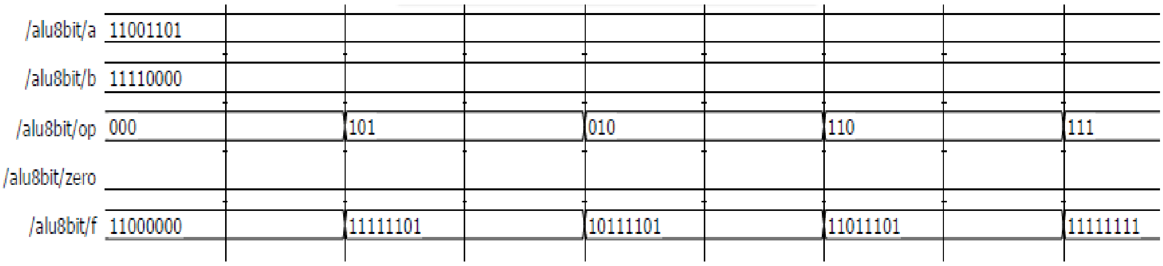

Fig. 4: Simulation of ALU.\

B. Simulation of Complete BIST for ALU

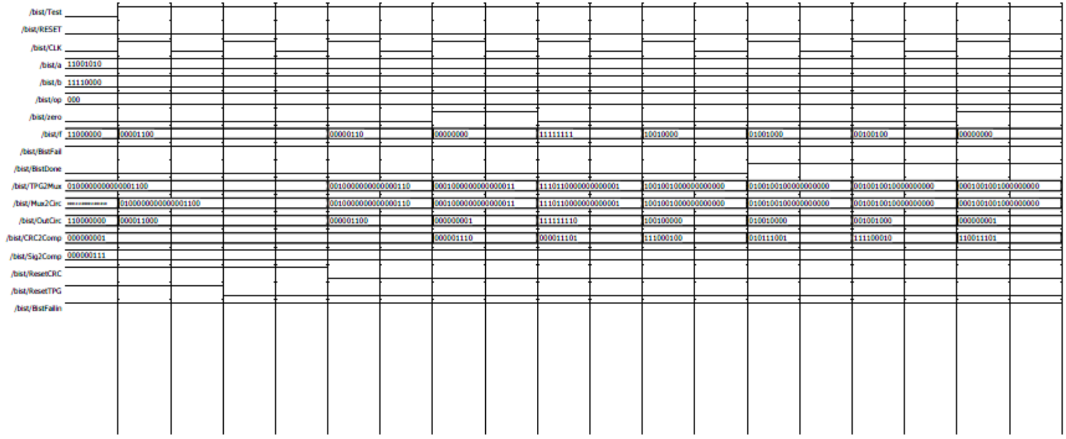

Fig. 5: Simulation of complete BIST for ALU

REFERENCES

[1]. Mohd Yamani Idna Idris, A VHDL Implementation of UART design with BIST capability, Malaysian Journal of Computer Science, Vol. 19 (1), 2006.

[2]. Bushnell M. L. and Agrawal V. D., Essentials of Electronic Testing, Kluwer Academic Publishers. 2000.

[3]. Shikha Kakar, Balwinder Singh, and Arun Khosla, Implementation of BIST Capability using LFSR Techniques in UART, International Journal of Recent Trends in Engineering (IJRTE), Issue on Electrical & Electronics Page(s): 301-304, Volume 1, Number 3, May 2009.

[4]. Suma T.Hegde ,Dr.Siva Yellampalli And Nandeesh R., Design and Implementation of ALU using Redundant Binary Signed Digit, International Conference on VLSI, Communication & Instrumentation (ICVCI), Proceedings published by International Journal of Computer Applications® (IJCA) 30, 2011.

[5]. Z. Navabi, VHDL Analysis and Modeling of Digital Systems, McGraw-Hill Inc., 1991.

[6]. C. H. Roth, Digital System Design Using VHDL, PWS Publishing Company, 1998

[7]. Crouch, A., Design-for-test for Digital IC's and Embedded Core Systems, Prentice Hall, 347 pp, 1999.

[8]. Devi Rehka, Singh Jaget and Singh Mandeep, VHDL implementation of GCD Processor with Build-In-Self-Test Feature, International Journal of Computer Applications (0975 – 8887) Volume 25– No.2, July 2011.

[9]. Singh, B.; Khosla, A.; Bindra, S., Power Optimization of Linear Feedback Shift Register (LFSR) for Low Power BIST, Advance Computing Conference IACC IEEE International., vol., no., pp.311-314, 6-7 March 2009.