Deepthi.P, Vinay.P / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1107-1110

Pseudo Exhustive Test Patterns For Online And Offline Bist On Fpga

Deepthi.P, Vinay.P
Department Of Ece, Jntuh, Hyderabad

Abstract

In this paper we discuss the built-in self-test (BIST) or built-in test (BIT) techniques forreconfigurable FPGAs.The FPGA is configured with the BIST logic during offline testing. Once the test is completed the BIST configuration is erased and the PGA is reconfigured to the original operation. In offline the BIST logic for the programmable logic blocks and interconnections are mentioned.

In online, the original operation of FPGA can be configured as a processor ALU with built-in-self test (BIST) feature.

Key features of the design including FPGA array structure, it's configuration for BIST, ALU, application of design with RISC architecture, data path, and simulation results are presented. The design is implemented using VHDL and verified on Xilinx ISE simulator

Keywords-RISC, BIST, VHDL, FPGA

I.INTRODUCTION

Field ProgrammableGate Arrays (FPGAs) feature their ability to be configured in the field to implement an arbitrary desired function according to the real-time demands. This ability of FPGAs can help people to achieve a faster design cycle, lower development costs and a reduced time-to market .Therefore, are widely used in many applications such as networking, storage systems, communication, and adaptive computing.

Field programmable Gate Arrays (FPGA) are devices consisting of an array of proposed in [5]. These methods are applicable only to reconfigurable FPGAs.In today's Integrated Circuits (ICs), Built-In- Self Test (BIST) is becoming increasingly important as designs become more and more complicated. Keeping the structural fault coverage high along with maintaining an acceptable design overhead is critical. It is important to achieve a high level of reliability with minimum cost and time. It is with this goal in mind that BIST has become a major design consideration in Design-For-Testability (DFT) methods. BIST is beneficial in many ways: First, it can reduce dependency on external Automatic Test Equipment (ATE). In addition, BIST can provide at speed, in system testing of the Circuit-Under-Test (CUT). This is crucial to the quality component of testing. BIST can overcome pin limitations due to packaging, make efficient use of available extra chip area, and provide more detailed information about the faults present. A generic approach to BIST is shown in Fig 1. On a very basic level, BIST needs a stimulus (the Test Pattern Generator - TPG), a circuit to be tested (CUT) and a way to analyze the results (ORA). Additionally, there may be compression schemes for the TPG and the ORA [2-3].

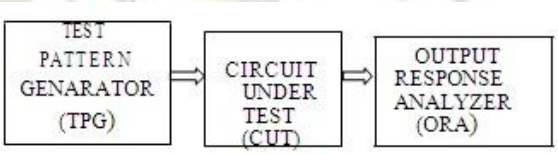

Fig. 1: Schematic of BIST

programmable logic blocks, interconnects and I/O blocks used to implement both combinational and sequential logic by programming these circuit elements. There are several different FPGA testing strategies. The main test technique that we are going to address is built in self test (BIST) method Reduced Instruction Set Computer (RISC) focuses on reducing the number and complexity of instructions in the machine. An arithmetic logic unit 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. 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 it result into an output register.

The organization of the paper is as follows: In Section 2, we present the FPGA array structure, BIST for CLB and interconnections. In section 3, we represent BIST feature ALU, RISC processor architecture, and data path of the proposed design. Simulation results are also provided in this section. The paper concludes in section 4.

II.BUILT IN SELF TEST (BIST)

The Xilinx FPGA uses symmetrical array structure and the logic blocks are called Configurable Logic blocks (CLB). Figure 2 shows the basic array structure of a Xilinx FPGA [1].

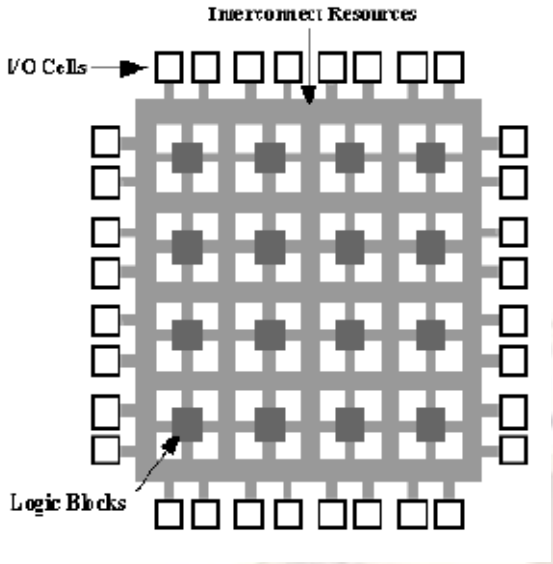

Fig.2.FPGA ARRAY STRUCTURE

A.TESTING THE CLB

The CLB consists of 1) Multiplexers and 2) LUT which together form the combinational part and 3) Flip flops which form the sequential module.

The reprogrammable nature of FPGAs can be used to an advantage for BIST. The FPGA is configured with the BIST logic during offline testing. Once the test is completed the BIST configuration is erased and the FPGA is reconfigured to the original operation. This implies that there is no overhead attached to the BIST in terms of additional circuitry or space. The BIST is applicable at all levels (wafer, package, board, system) and is performed at normal operating frequency [4].

A test controller is used to configure the FPGA with the BIST logic, start the testing and read the test results. Additional memory is required to store the BIST configuration as well as the original FPGA configuration that is restored after the FPGA has been tested. This is the main cost of the BIST;but memory is available in the test machine and hence no additional resources are required of the device under test. Also as the testing is at-speed, the testing time is low and so the cost of testing is low. The BIST configuration is independent of the function implemented in the FPGA, and is dependent only on the FPGA architecture so all the FPGAs of the same type can be tested by the same BIST configuration. This reduces cost of testing because of reusability of the test.

The main steps of carrying out BIST on a FPGA are: 1) The test controller machine reconfigures the FPGA with the BIST logic and 2) initiates the test. 3) The BIST logic generates test patterns within the FPGA and 4) analyzes responses and generates the pass/fail output. 5) The test controller reads the test results. In steps 1, 2, 5 the test controller interacts with the FPGA through the I/O pins and steps 3, 4 are internal to the FPGA.

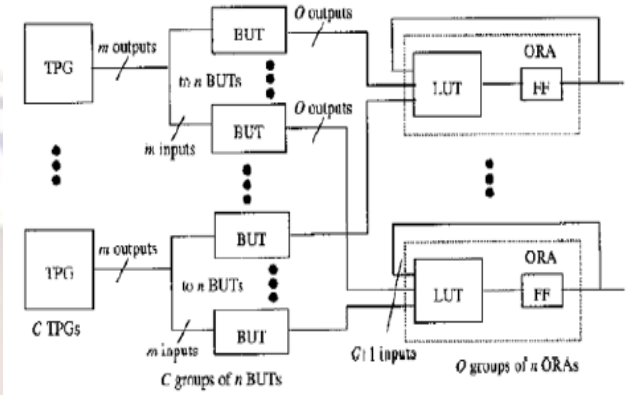

Fig.3.FPGA configuration for BIST

For conducting the self test in steps 3 and 4 some of the CLBs in the FPGA are configured as test pattern generators (TPGs) and output response analyzers (ORAs). The remaining CLBs are tested with pseudo exhaustive test patterns as described in [5]. These CLBs are referred to as the blocks under test (BUTs). As every BUT in an FPGA is identical, the ORA has to compare the outputs of any two BUTs. Only if the two BUTs fail in an identical manner will the fault escape detection which is highly unlikely. Another case when the tests would be incomplete is when the two BUTs are fed by the same faulty TPG. This problem is solved by careful configuration which ensures that an ORA compares BUTs tested by different TPGs and that all the TPGs are synchronized. Figure 3 shows the configuration for testing [6].

The BUTs are completely tested for all modes of operation in one test session. After the BUTs are tested the roles of the CLBs are reversed and the BUTs become TPGs and ORAs and vice versa and the test session is repeated. At least 2 test sessions are required to test the CLBs. In one test session as all the BUTs are tested in parallel the time required for the BIST does not depend on the size of the FPGA.

During most of the test configurations the TPGs behave as m-bit binary counters to supply exhaustive test patterns to the m-input BUTs. As the numbers of inputs are more than number of outputs many CLBs are required to generate the m-bit counter. As different TPGs are required for BUTs compared by the same ORA the number of TPGs required is equal to the number of BUT

outputs that can be compared by one ORA. An ORA is built from an LUT for comparing inputs and a flip-flop that provides feedback to the LUT as shown in figure 3. The flip-flop records the first error detected and also disables further comparisons by the LUT. All the inputs of the LUT except one for the feedback are connected to the BUT outputs and the LUT compares the values and if there is a mismatch it records it on the flip-flop.

The Xilinx XC4000 series has two 4- input LUTs in a CLB and two flip-flops. So three comparisons (number of inputs/LUT - 1) per LUT can be done and hence the number of TPGs required is three. The number of inputs and outputs per CLB are 12 and 5 respectively. To generate a 12-bit counter 12 flip-flops i.e. 6 CLBs are required for a TPG. All the three parts viz. LUTs, flip flops and multiplexers of the FPGA are tested and the fault coverage is 100%.

B.BIST for interconnect

In [7], the BIST method is extended for testing the interconnect faults in both local (multiplexer PS) and global (switch matrix) routing. The fault model used covers all stuckoff (stuck-on), stuck-open (stuck-short), stuck-at 0/1 in both the wire segments and the PSs. A group of k wires and PSs are combined to form the wires under test (WUT), and exhaustive testing is done by applying all 2k patterns to every group. Such a test that applies 0 and 1 values at one end of the wire and expects the same value at the other end can detect any stuck-open fault on a wire with a closed PS, and also any stuck- at faults. To test the stuck-on fault of a PS, opposite values should be applied to both the wire segments associated with that PS. Multiplexer-based PSs can be tested using functional tests as before, by select all possible data-inputs and with each data being tested for both 0 and 1 values. The BIST configuration is similar to the one used for testing CLBs described in above involving TPGs and ORAs. Test patterns generated by a TPG are applied to two WUTs and the outputs are compared by ORA. The path from the TPG to the ORA consists of the WUTs and intermediate PLBs that are configured to pass data from inputs to outputs. This achieves both local and global interconnect test coverage. The number of configuration phases required is reduced as multiple WUTs are tested in parallel. This method has the same shortcomings as the BIST for CLBs, i.e. identical faults appearing on WUTs compared by the same ORA will escape detection.

IV .BIST feature ALU

The architecture of the ALU consists of two parts, the Operation Architecture, which does the actual operation of the ALU, and the Testing Architecture, which comes into play only during testing. The Operation Architectures consists of five units, 4-bit Carry Look Ahead adder (CLA), and a 4-bit AND, OR,XOR and INVERTER gates. There is a PreCLA to prepare the inputs based on the arithmetic operation to be done. There is a MUX which uses the select pins to select one of the results from the above five units. The Testing Architecture has a ROM which has the discovered test patterns stored in. There is an address decoder to select which of the test patterns will be applied. There is a TestMUX, which depending on the value on the TestMode pin will present the test pattern or the actual inputs to be operated upon, to the Operation Architecture.

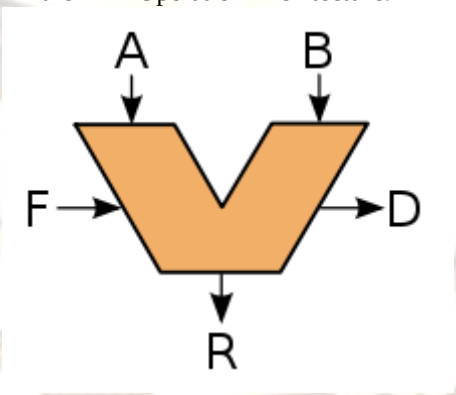

Fig .4. ALU

V.APPLICATION

The RISC processor presented in this paper consists of three components. Those are, the ALU with BIST feature, the Data Path, and the ROM. The Central Processing Unit (CPU) has 33 instructions.

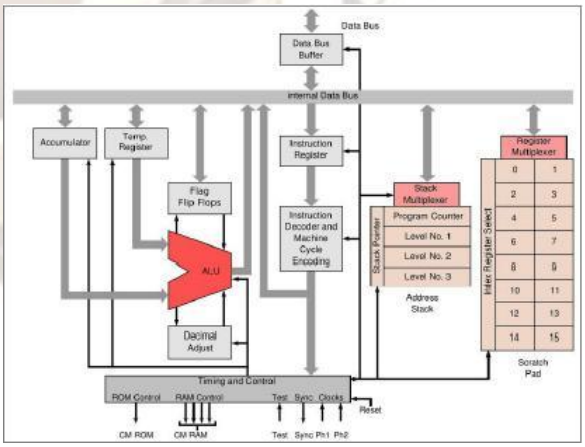

Fig.5.RISC processor architecture

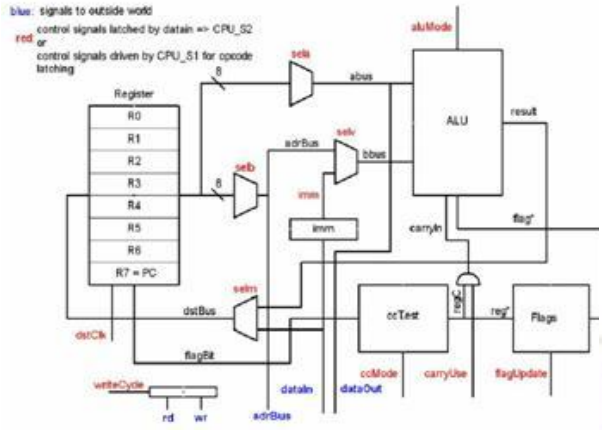

Fig.6. Data path for RISC Processor

The data to the processor is obtained from the external world. The data is processed by the processor by apply control signals. The control signal is designed in architecture of processor. After processing the data its again sent along the data path of processor to external world.

V.SIMULATION RESULTS OF RISC

The design is implemented using VHDL and verified on Xilinx ISE simulator.

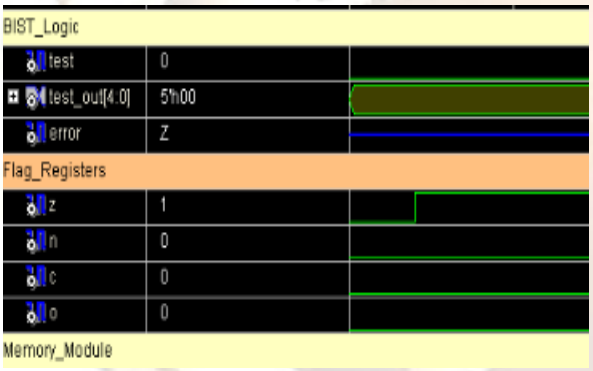

Fig.7. simulation results

IV. CONCLUSION

The most important requirement of a good FPGA test is that it should be independent of the end-application and the array size. BIST uses pseudo exhaustive test pattern that provides maximum fault coverage without assuming a fault model. Two test configurations are required for complete testing and testing is conducted at-speed. Other advantages that BIST offers is that it does not require additional storage space for the test vectors, since the CLBs are configured as TPGs to generate the test vectors. The BIST approach generates the test vectors internally. The comparator based BIST provides complete test coverage for global and local routing but fails to detect identical faults occurring in two WUTs compared by the same ORA. RISC processor with 33 instruction set and operation and testing ALU has been designed. Every instruction is executed in two clock cycles and corresponding result is verified with test vectors in ROM. The design is verified through exhaustive simulations.

REFERENCES

[1] Stephen Brown and Jonathan Rose, Architecture of FPGAs and CPLDs: A Tutorial, University of Toronto.

[2]. R. N. Noyce and M. E. Hoff, "A History of Microprocessor Development at Intel," IEEE Micro, vol. 1, no. 1, 1981, pp. 8-21.

[3]. J.L. Hennessy, "VLSI Processor Architecture," IEEE Trans.Computers, vol. C-33, no. 12, Dec. 1984, pp. 1221-1246.

[4] C. Stroud, S. Konala, P. Chen, M. Abramovici, Built-In Self-Test of Logic Blocks in FPGAs (Finally, a Free Lunch: BIST Without Overhead!), Proc. 14th VLSI Test Symposium, pp. 387-392, 1996.

[5] E. McCluskey, Verification Testing - A Pseudoexhaustive Test Technique, IEEE Transactions on Computers, Vol. C-33, No. 6, pp.541-546, June 1984.

[6] R.Sharama,v.k.sehgal,nitin.pranav bhasker,ishita verma- Design and Implementation of a 64-bit RISC Processor using VHDL, UKSim 2009: 11th International Conference on Computer Modelling and Simulation.

[7] C. Stroud, S. Wijesuriya, C. Hamilton, M. Abramovici,Built-in Self Test of FPGA Interconnect,Proc. IEEE International Test Conference., pp.404-411, 1998.