ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9 S(1), DOI: 10.17485/ijst/2016/v9iS(1)/108455 December 2016* ISSN (Online) : 0974-5645

Improving Testability of Design in FPGA using Raspberry Pi

S. Karthik[1*], N. Srividya[1], L. Swetha[1] and Jean Shilpa[2]

[1]Electronics and Communication Engineering, SRM University, Vadapalani, Chennai - 600026, Tamil Nadu, India; skarthikvit@gmail.com, srividya2910@yahoo.co.in, lswetha75@gmail.com
[2]Electronics and Communication Engineering, BSAU University, Chennai - 600048, Tamil Nadu, India; jeanshilpa@gmail.com

Abstract

Objective: To implement a logical approach for providing better ability to observe and control FPGA pins through Raspberry Pi for functional verification of FPGA based system design. **Method/Analysis:** It mainly involves improving the testability of design in FPGA using Raspberry Pi. **Findings:** The use of Raspberry Pi makes the controllability and observability of FPGA pins easy and the inputs, outputs of FPGA can be remotely controlled via Webiopi which is a web interface for Raspberry Pi. **Novelty/Improvement:** The web interface is also used to start the BIST and to test the FPGA with test vectors any times.

Keywords: ATE, BIST, FPGA, Raspberry Pi, Webiopi

1. Introduction

Reconfigurable computing[1–4] establishes itself as a major discipline combining some of the litheness of software and high performance hardware also including various subjects of learning, that include both electronic engineering and computer science engineering. Reconfigurable computing makes use of Field Programmable Gate Arrays (FPGAs) which can be used as reconfigurable device. FPGAs are a well-liked way to implement hardware in many products. In ASIC designing, the logic design, simulation/ verification, static timing analyses are done widely in electronic design automation software before the design is manufactured in silicon wafer. The software based design verification is slow and scrupulous and the design engineer has to put more effort to ensure that the design is bug free, will work with respect to specifications and operate at the preferred quickness when manufactured in silicon. The next lengthy step is that post verification which is called testing done after manufacturing of IC is a lengthy process and also costly. When FPGA is considered the hardware is manufactured and it is ready from outset. So implementing a design in the hardware and testing the design is much faster so verification of a design in the hardware is very much sooner than software and bugs and be easily found and corrected if controllability and observability of the FPGA pins are provided. Reprogrammability in FPGA avoids the broad fabrication step connected with ASICs since designers implement applications directly onto the FPGA device. Even though ASICs are more preferred than FPGAs for tremendously huge and compound designs, such as a high performance processor, where FPGAs just cannot present the needed rate and logic necessary for the design, FPGA based design system offer irreplaceable solutions to an extensive variety of circuit designs. To utilize the fullest advantage of FPGA electronic design automation tool have to make available same level of controllability and observability as compared to software simulator to facilitate design engineers to speed up simulation/ verification of the design. This includes controlling and observing FPGA signals and also to run BIST which can be done remotely.

1.1 Role of Raspberry Pi

The Raspberry Pi[5] is a diminutive sized CPU industrialized with a purpose to encourage elementary teachings in the areas of computer, robotics for schools. The resourceful

* *Author for correspondence*

Raspberry Pi and Pi 2 are industrialized and manufactured in many configurations. In this work we use Raspberry Pi GPIO to control the FPGA pins and also test the design configured in FPGA any time by running the BIST. All this are done remotely using Webiopi which is shown in Figure 1.

Figure 1. Raspberry Pi.

1.2 Role of Webiopi

The Webiopi[5] is a free web interface used for Raspberry Pi which a web based application is allowing you to control the GPIOs of Raspberry Pi. We took the advantage of this in controlling the FPGA pins. Figure 2 shows how Webiopi can be used in controlling the GPIOs of Raspberry Pi.

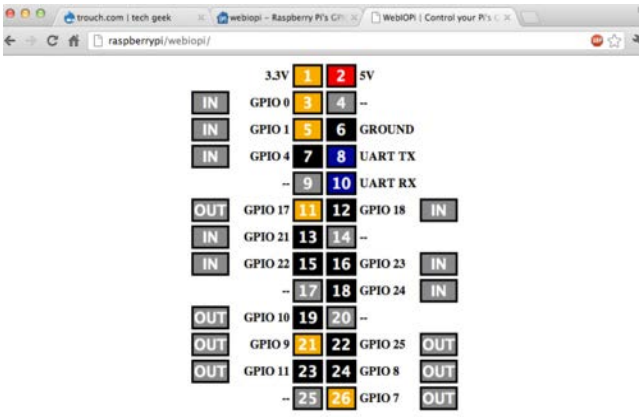

Figure 2. GPIOs of Raspberry Pi shown in Webipoi.

2. BIST

Built in Self-Test are essentially a DFT (Design-For-Testability) method similar to testing done using Automatic Test Equipment. The TPG (test-pattern-generator) and RA (Response-Analyzer) are manufactured along with the design on chip (instead of equipments). Since equipments are not needed, we need to design the compacted version of test pattern generator and response analyzer. The basic BIST block is shown in Figure 3.

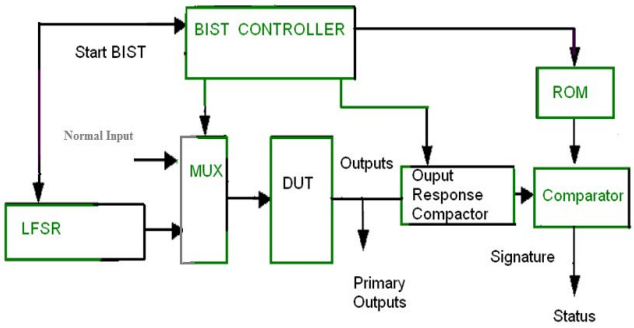

Figure 3. Basic BIST architecture.

2.1 Linear Feed Back Shift Register

A LFSR produce test patterns that are required to activate and sensitize the faults to output of a CUT. The pattern generator used for testing is hardware and not equipment. Linear feedback shift register is designed using series of flip-flops acting as a shift register and are linearly connected using Exclusive OR gates with feedback loops. An LFSR will act as pattern generator, response compaction and also to perform polynomial division etc. The design of LFSRs can be done using two types of feedback: Internal and External. These implementations vary in the feedback position is applied. The flip-flop outputs are connected to the XOR gate which is known as 'taps'. Taps are responsible and used to decide the binary states produced by the shift register and hence defines the characteristic polynomial. If the feedback is external LFSR type the XOR gates lies in the feedback path and XOR of all the taps forms the input to the FF. In case of an internal feedback LFSR type, the xor gate lies between the flipflops. The primary state of the FF is called the seed. The characteristic polynomial and the seed decide the binary states produced by the LFSR.

Generating test patterns using ATPG, storing it in ROM and applying to the circuit under test is not feasible. Reading test patterns and applying it on the CUT takes more time since access time is high. Instead the feasible solution is to use the test pattern generator as a counter which generates test patterns which are random in nature. The main prominence of the register design is to use less area but still produce as many different test patterns as possible.

2.2 Input Multiplexer

The multiplexer takes normal inputs coming from the outside world and also test inputs produced from the linear feedback shift register and depending upon selection line BIST mode or Normal mode can be. The control input to the multiplexer is fed by a central test controller.

2.3 Response Compactor

LFSR are used as response compactor. They are used to perform lossy compression on the outputs of the CUT. A golden signature is obtained from the compactor which is considered to be the good response of the circuit. If the output of the circuit under test does not coincide with the signature, a fault has occurred. Use of compactor will reduce the size of ROM. If the CUT as many outputs then we can use Multiple Input Signature Analyzer (MISR) and generate a single signature.

2.4 Read Only Memory

It saves the correct signature which needs to be matched with the response from CUT.

2.5 Comparator

The Comparator is the hardware used to compare the response from the CUT with the golden signature stored in ROM.

2.6 Test Controller

It is the circuit which controls and generates signals which activate a particular data path in the BIST architecture. Whenever the supply to the IC is given the test controller will issue signals to activate the IC in BIST mode, where the test vectors are applied to the CUT. Once the signatures are verified and if they did not match an output line is made high to indicate the fault. If the signature matches, the controller comes out from the test mode and normal inputs will be applied to the DUT from the output of the multiplexer. Now the IC can be used in normal operation.

3. Spartan FPGA

The Spartan 3E family[9,10] Field Programmable Gate Array (FPGAs) provides a commanding and highly superior self-reliant development platform for designs envisioned to meet the enormous requirements. Because of its extraordinarily low cost per logic cells, the Spartan 3E FPGAs are suited for a extensive choice of applications. They have 36 embedded 18*18 multipliers for high performance DSP applications. The other features are they have 100 k to 1.6 M system gates, 376 I/Os and up to 648 Kbits of block RAM and up to 231 Kbits of distributed RAM. The Spartan-3E FPGAs are configured with the bin or hex file by loading it into FPGA which has static CMOS configurable latches that jointly control all the logical elements and routing resources. The configuration bits .hex or .bin is stored in a Programmable Read Only Memory either off or on the board.

4. Controllability and Observability

Controllability and observability are SCOPE (Sandia Controllability and Observability Analysis Program) testability measures. Controllability measure will tell us the difficulty in setting a node to 0 or 1 from the primary input and observability measure will tell us the difficulty in observing the node at the closest primary output. Here the Raspberry Pi is used to control the inputs of FPGA and is also used for observing the outputs of FPGA. In case of observing and controlling any intermediate signal, the signal can be routed to any pins of FPGA so that it can be viewed or controlled.

5. Proposed Architecture

The proposed design is shown in Figure 4. Here the input and output are controlled/viewed using Raspberry Pi. The inputs to the FPGA and Raspberry Pi's GPIOs are multiplexed and are fed to the FPGA, similarly the outputs of the FPGA are connected to the Raspberry Pi's GPIOs so that it can be viewed.

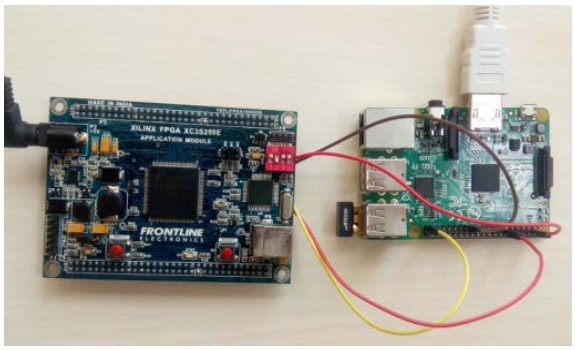

Figure 4. Raspberry Pi connected with FPGA.

6. Experimental Setup

Figure 5 shows the Raspberry Pi's GPIO's are connected to the FPGA .Through the Webiopi the inputs are fed to the FPGA and for running BIST also the control is fed from Webiopi. For demonstration purpose we have taken a sample design form ISCAS bench mark circuit and Verilog code is written for the circuit and BIST for the circuit was designed and configured in FPGA. Then depending on the application, the user can control the FPGA pins or run the BIST for testing the design using Webiopi.

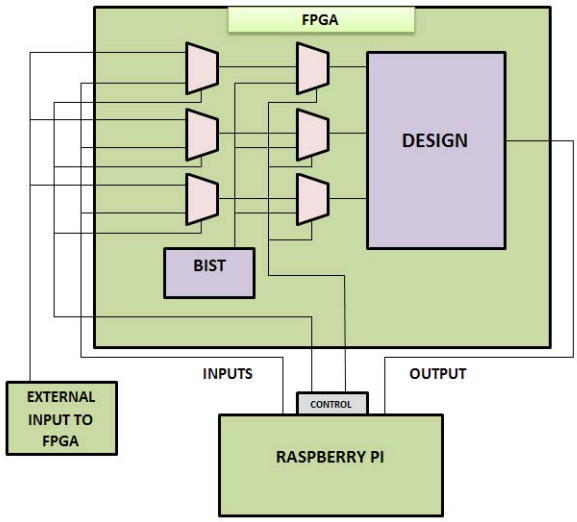

Figure 5. Proposed architecture.

7. Conclusion

Thus the advantage of Webiopi is that it can directly control the GPIOs of Raspberry Pi that are used in controlling and observing FPGA pins and also to run the design in BIST mode. There are plenty of application one can think of when we interface Raspberry Pi and FPGA like fault injection, Ad hoc testing, Remote FPGA labs etc.

8. References

1. Blodget B, James-Roxby P, Keller E, McMillan S, Sundararajan P. A self-reconfiguring platform. Proceedings of the 13th International Conference on Field Programmable Logic and Applications (FPL'03); 2003 Sep. p. 565–74.
2. Fiethe B, Michalik H, Dierker C, Osterloh B, Zhou G. On reconfigurable system-on-chip data processing units for miniaturized space imaging instruments. ACM; 2007. p. 977–82. ISBN 978-3-9810801-2-4.
3. Nathan D, Lim K, Choo K, Wong P, Weisensee A. A high level reconfigurable computing platform software frameworks, arXiv.org cs.AR/0405015; 2004 May.
4. Ullmann M, Huebner M, Grimm B, Becker J. A FPGA runtime system for dynamical on-demand reconfiguration. RAW04; Santa Fe. 2004 Apr.
5. Vidya Sagar D, Vyshnavi Vihitha G. WebIOPi - Remote controlled robot cam using rest framework. International Journal of Emerging Research in Management and Technology. 2015; 4(8):73–7. ISSN: 2278-9359.
6. Wang S, Gupta S. DS-LFSR: A new BIST TPG for low heat dissipation. Proc Int Test Conf. (ITC'97); 1997. p. 848–57.
7. Thameema Begum J, Harshavardhan Naidu S, Vaishnavi N, Sakana G, Prabhakaran N. Design and implementation of reconfigurable ALU for signal processing applications. Indian Journal of Science and Technology. 2016 Jan; 9(2).
8. Hertwig A, Wunderlich HJ. Low power serial built-in self-test. Proc IEEE Eur Test Workshop; 1998 May. p. 49–53.
9. Karthik S, Shreya, Viswanath, Srihari. Remote FPGA Lab. International Journal of Research Engineering and Technology. 2014 Jul. eISSN: 2319-1163 | pISSN: 2321-7308.
10. Karthik S, Prasanna vishal TR, Jayaram SG, Priyadarsini K. GSM based FPGA configuration. IOSR Journal of VLSI and Signal Processing (IOSR-JVSP). 2014 Jul-Aug; 4(4):58–61. e-ISSN: 2319–4200, p-ISSN No: 2319–419.