IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

VHDL Implementation of BIST Enabled UART

Manisha Golia
M. Tech. Scholar
manisha_golia@rediffmail.com
Oriental University, Indore
(India)

Sunil Shukla
Assistance Professor
sunilshukla.oriental@gmail.com
Oriental University, Indore
(India)

H. R. Singh
Professor
hrsingh@oriental.ac.in
Oriental University, Indore
(India)

Abstract – **This paper introduces a novel approach of Linear Feedback Shift Resister (LFSR) for Built-in-self-test (BIST) along with 8 bit UART, to overcome testability and data integrity. The complete design is implemented in VHDL and simulated throughout in Modelsim, synthesis is carried out in Xilinx 14.1. The IP core is implemented on FPGA device 3s500efg320-4. The results indicate that this model eliminates the need of higher end, expensive testers and thereby it can reduce the development time and cost.**

Keywords – **BIST, FPGA, LFSR, Modelsim, UART, VHDL, Xilinx.**

I. Introduction

Asynchronous serial Communication has advantages of less transmission lines, high reliability and long transmission distance. Universal Asynchronous Receiver Transmitter (UART) allows full-duplex communication in serial link, thus has been widely used in the data communications and control system. It is widely used in data exchange between Processor and peripherals. UART converts data from parallel to serial at transmitter with some extra overhead bits using shift register and vice-versa at receiver. To the processor the UART appears as an 8-bit read/write parallel port.

Basic UART communication needs only two signal lines (Receive, Transmit) to complete full-duplex data communication. UART includes three modules namely, the baud rate generator, receiver and transmitter. The baud rate generator is used to produce a local clock signal which is much higher than the baud rate to control the UART receive and transmit; The UART receiver module is used to receive the serial signals at RXD, and convert them into parallel data; The UART transmit module converts the bytes into serial bits according to the basic frame format and transmits those bits through TXD.

With the increasing growth of sub-micron technology has resulted in the difficulty of testing. Manufacturing processes are extremely complex, making the manufacturers to consider testability as a requirement to assure the reliability and the functionality of each of their designed circuits. Built-In-Self-Test (BIST) is one of the most popular test technique used. For design and test development, BIST significantly reduces the costs of automatic test-pattern generation (ATPG) and also reduces the likelihood of disastrous product introduction delays because of a fully designed system cannot be tested.

A Universal Asynchronous Receive / Transmit with BIST capability has the objectives of firstly to satisfy specified testability requirements, and secondly to generate the lowest-cost with the highest performance implementation.

Although BIST slightly increases the cost because of the BIST hardware overhead in the design and test development, due to added time required to design and added pattern- generators, response compactors, and testability hardware. However, it is normally less costly than test development with ATPG. With the reasons discussed above, this project focuses on the design of the embedded BIST architecture for an IOP. The designs are implemented using Very High Speed Integrated Circuit Hardware Description Language (VHDL) at the Register Transfer Level (RTL) abstraction level. BIST technique is cooperated into the IOP design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements. This paper focuses on the design of a UART chip with embedded BIST architecture using Field Programmable Gate Array (FPGA) technology.

II. Proposed Method

UART Architecture with BIST

The architecture proposes an 8-bit UART which operates at a baud rate of 9600 bps with a status register to monitor the correctness of every received data byte and enhance the testability of circuit by the introduction of BIST module. The hardware architecture of the 8-bit UART with Status register, incorporated with BIST module is explained in the following sections. The proposed model has two

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

major modules viz. UART and BIST. Further in the UART, we have transmitter, receiver, and baud rate generator. Baud rate generator works at 50 MHz and further reduced as required for the operations in transmitter and receiver to achieve baud rate of 9600 bps. BIST has a control register, pattern generator and a comparator, as shown in Figure 1.

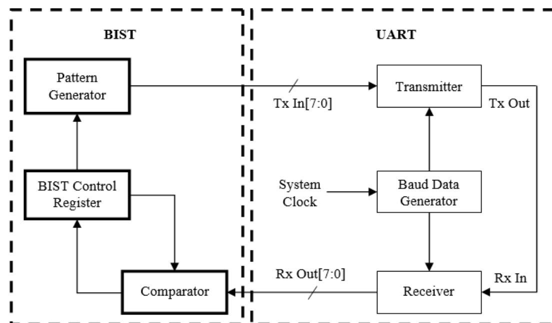

Figure 1: UART with BIST architecture [1]

Serial data is transmitted via its serial port. A serial port is one of the most universal parts of a computer. It is a connector where serial line is attached and connected to peripheral devices such as mouse, modem, printer and even to another computer. In contrast to parallel communication, these peripheral devices communicate using a serial bit stream. A VHDL Implementation of UART Design with BIST Capability protocol (where data is sent one bit at a time). The serial port is usually connected to UART, an integrated circuit which handles the conversion between serial and parallel data [11] [12].

UART receives a byte of parallel data and converts it to a sequence of voltage to represent 0s and 1s on a single wire (serial). To transfer data on a telephone line, the data must be converted from 0s and 1s to audio tones or sounds (the audio tones are sinusoidal shaped signals). This conversion is performed by a peripheral device called a modem (modulator/demodulator). The modem takes the signal on the single wire and converts it to sounds. At the other end, the modem converts the sound back to voltages, and another UART converts the stream of 0s and 1s back to bytes of parallel data.

UART Transmitter

The transmitter accepts parallel data from peripheral/processor, makes the frame of the data and transmits the data in serial form on the Transmitter Output (TXOUT) terminal (Figure 1). The baud rate generator output will be the clock for UART transmitter.

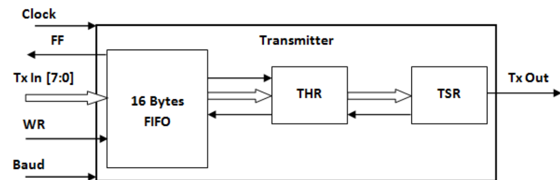

Figure 2: UART transmitter

Data is loaded from the parallel inputs TXIN0-TXIN7 into the Transmitter FIFO by applying logic high on the WR (Write) input. FIFO is 16-byte register. If FIFO is full, it sends FIFO Full (FF) signal to peripheral as shown in Figure 3.

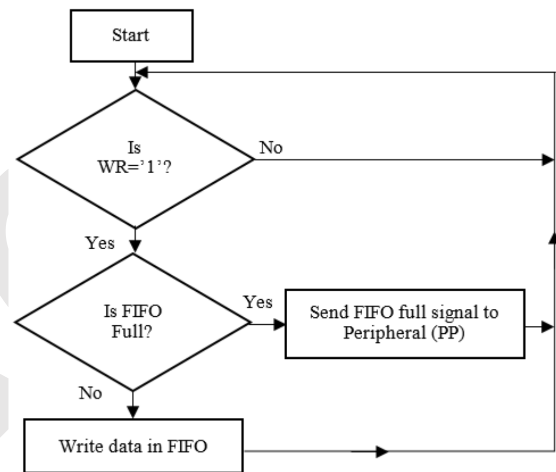

Figure 3: Transmitter flowchart – Input to FIFO

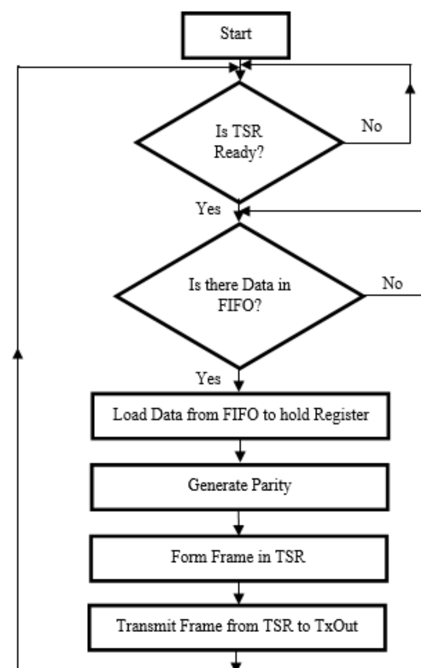

Figure 4: Transmitter flowchart – FIFO to TXOUT

IJDACR
ISSN: 2319-4863
IJDACR
International Journal Of Digital Application & Contemporary Research
International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

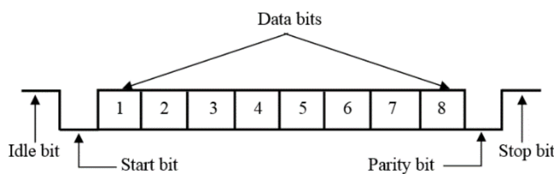

Figure 5: Transmitter flowchart – FIFO to TXOUT

When FIFO contains some data, it will send the signal to Transmitter Hold Register (THR), which is an 8-bitregister. At the same time, if THR is empty it will send the signal to FIFO, which indicates that THR is ready to receive data from FIFO. If Transmitter Shift Register (TSR) is empty, it will send the signal to THR and it indicates that TSR is ready to receive data from THR. TSR is an 11-bitregister in which framing process occurs. In frame, start bit, parity bit and one stop bit will be added as shown in Figure 5. Now data is transmitted from TSR to TXOUT serially. Figure 4 is the flowchart explaining transmission of serial data from FIFO to transmitter output.

UART Receiver

The received serial data is available on the RXIN pin. The received data is applied to the sampling logic block. The receiver timing and control is used for synchronization of clock signal between transmitter and receiver. The receiver sampling is 16 times to that of the transmitter baud rate. In the architecture of UART receiver (Figure 6), initially the logic line (RxIn) is high.

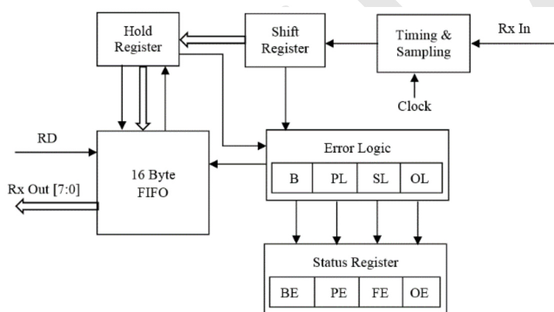

Figure 6: UART Receiver

Whenever it goes low sampling and logic block will take 4 samples of that bit and if all four are same it indicates the start of a frame. After that remaining bits are sampled in the same way and all the bits are send to Receiver Shift Register (RSR) one by one where the entire frame is stored. RSR is a 12 bit shift register. Figure 7 shows the receiver logic.

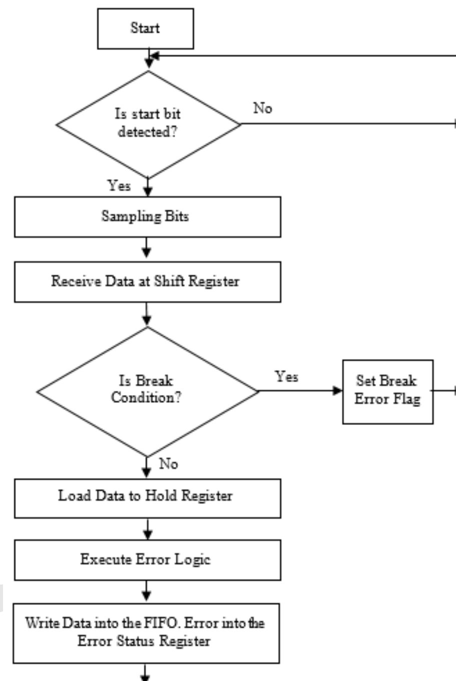

Figure 7: Receiver flowchart (Input to FIFO)

Now if the Receiver Hold Register (RHR) is empty it sends signal to RSR so that only the data bits from RSR goes to RHR which is an 8 bit register. The remaining bits in the RSR are used by the error logic block. Then, if receiver FIFO is empty it send the signal to RHR so that the data bits goes to FIFO. When RD signal is asserted the data is available in parallel form on the RXOUT0-RXOUT7 pins. The status register is implemented with flags for error logic operations performed on the received data. The error logic block handles 4 types of errors: Parity error (PE), Frame error (FE), Over run error (OE), Break error (BE). If the received parity does not match with the parity generated from data bits, PE bit will be set which indicates that parity error occurred. If receiver fails to detect correct stop bit or when 4 samples do not match frame error occurs and FE bit is set. If the receiver FIFO is full and other data arrives at, RHR overrun error occurs and OE bit is set. If the RXIN pin is held low for long time than the frame time then there is a break in received data and break error occurs and BE bit is set. Reading of data from receiver is explained by means of flowchart in Figure 8.

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

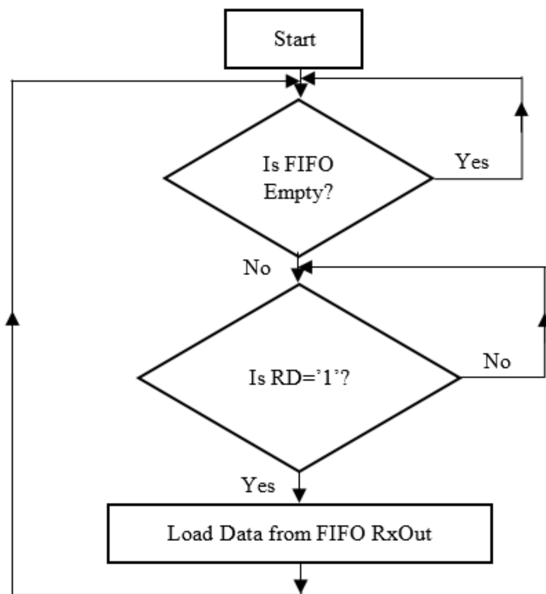

Figure 8: Receiver flowchart (FIFO to Output)

BIST Architecture

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.

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.

BIST Implementation on UART

In implementation part, the BIST techniques with four LFSR based test pattern generation is incorporated into the UART design before the overall design is synthesized by means of recognizing the exiting design to match built in test requirements as we discussed Test pattern generator is the very important part, which have different possible circuits.

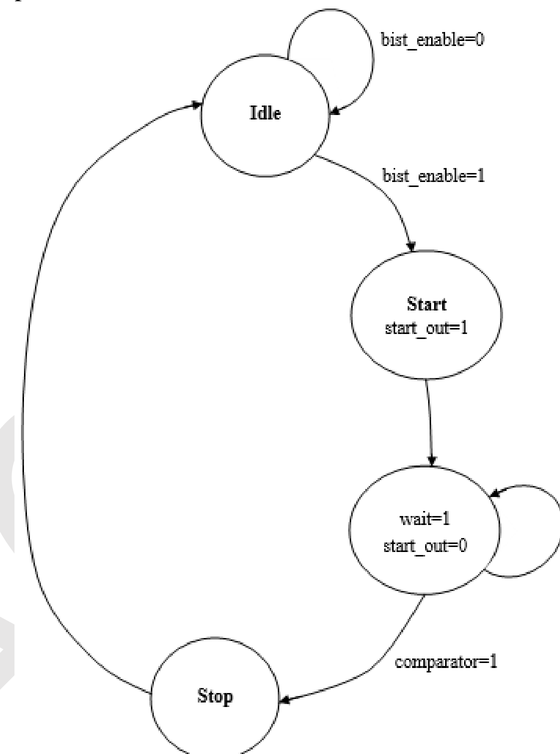

Figure 10: BIST interface

III. Simulation And Results

Proposed architecture is synthesized using Xilinx 13.1 ISE and simulated using Modelsim 6.3.

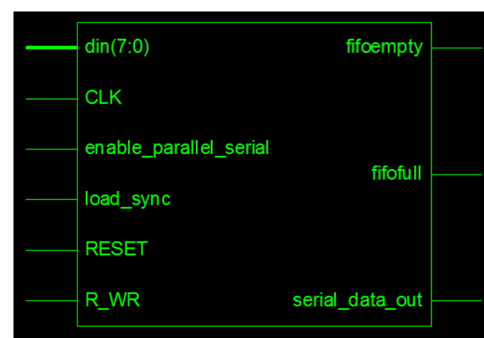

Figure 11: RTL view of UART transmitter

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

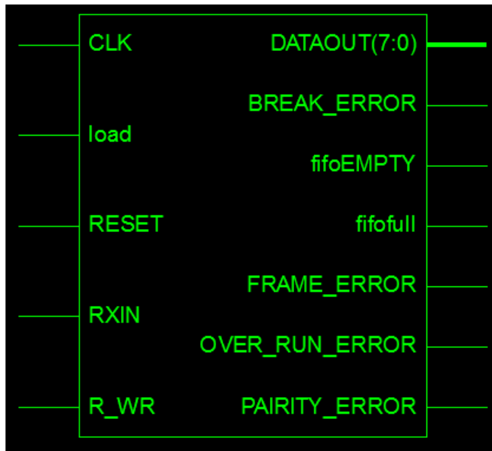

Figure 12: RTL view of UART receiver

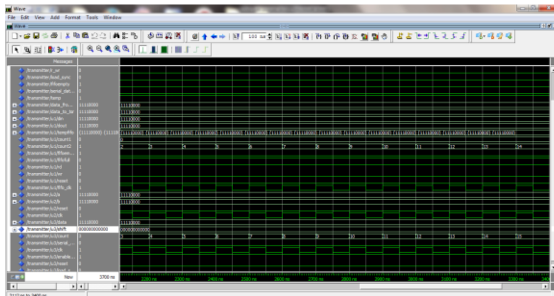

Figure 13: Simulation waveform for UART transmitter

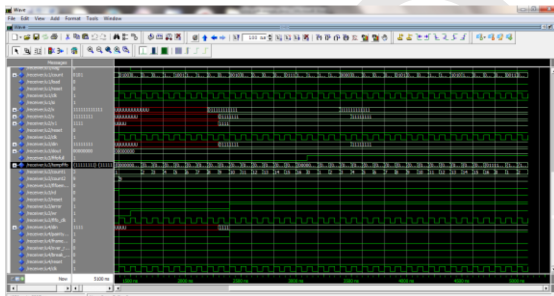

Figure 14: Simulation waveform for UART receiver

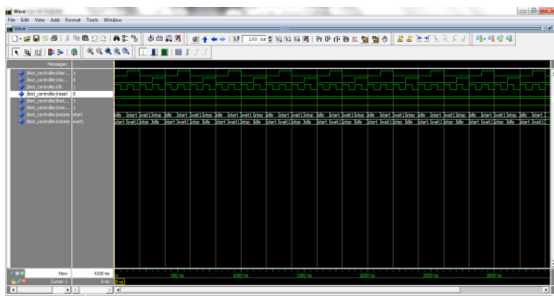

Figure 15: Simulation waveform for BIST Controller

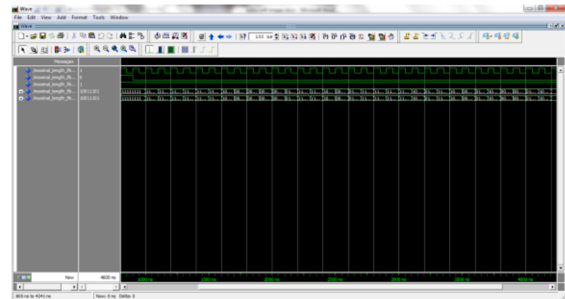

Figure 16: Simulation waveform for LFSR

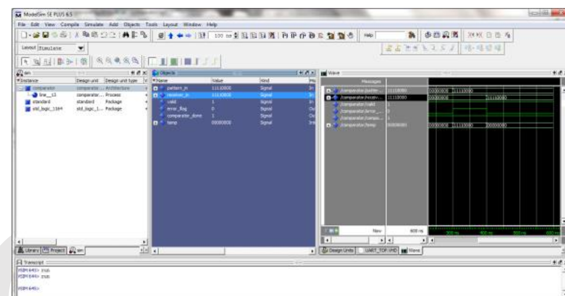

Figure 17: Simulation waveform for comparator

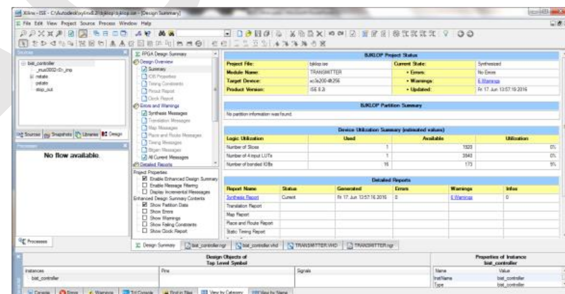

Figure 18: UART transmitter synthesis result

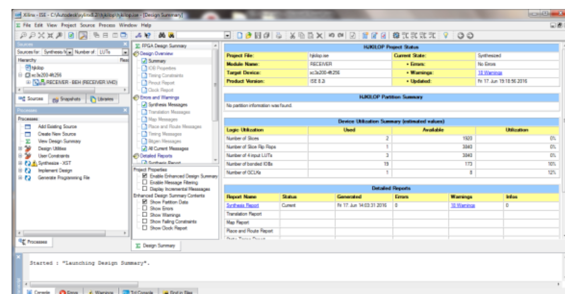

Figure 19: UART receiver synthesis result

IV. Conclusion

With the implementation of BIST, expensive tester requirements and testing procedures for circuit or logic level to field level testing are minimized. The LFSR is taken for test pattern generation that gives good fault coverage to the UART module and reduces the switching activities between the test

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: www.ijdacr.com (Volume 4, Issue 11, June 2016)

patterns. It was found that the proposed architecture out performs than of previous on the basis of different parameter values. Limitation of the this project is the conventional algorithm for the test pattern generation works well for single feedback LFSRs but suffers from major constraints when multiple feedback paths are employed in the LFSR, UART which only good for short distance and low speed communication between computer peripherals.

REFERENCE

[1] Bibin M C, Premananda B S, “Implementation of UART with BIST Technique in FPGA”, International Journal of Inventive Engineering and Sciences (IJIES), ISSN: 2319–9598, Vol. 1, Issue 8, July 2013.
[2] Shumit Saha, Md. Ashikur Rahman, Amit Thakur, “Design and implementation of a BIST embedded high speed RS-422 utilized UART over FPGA”, Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-5, July 2013.
[3] M. Schrader, R. McConnell, "SoC Design and Test Considerations," Proc. Design, Automation and Test in Europe, pp. 202-207, 2003.
[4] M. Abramovici and C. Stroud, "BIST-Based Test and Diagnosis of FPGA Logic Blocks," IEEE Trans. on VLSI Systems, Vol. 9, No. 1, pp. 159-172, 2001.
[5] C. E. Stroud, K. N. Leach, T. A. Slaughter, “BIST for Xilinx 4000 and Spartan Series FPGAs: A Case Study,” Proc. IEEE International Test Conf., pp. 1258-1267, 2003.
[6] J. Sunwoo, S. Garimella, C. Stroud, “On Embedded Processor Reconfiguration of Logic BIST for FPGA Cores in SoCs,” Proc. IEEE North Atlantic Test Workship, pp. 15-22, 2005.
[7] K. Iijima, A. Akar, C. McDonald, and D. Burek, “Embedded Test Solution as a Breakthrough in Reducing Cost of Test for System on Chips,” Proc. IEEE Asian Test Symp., pp 311 – 316, 2002.
[8] C. E. Stroud, A Designer’s Guide to Built-In Self-Test, Springer-Verlag, New York, 2002.
[9] Mohd Yamani Idna Idris, Mashkuri Yaacob, Zaidi Razak, “A VHDL Implementation of UART Design with BIST Capability”, Malaysian Journal of Computer Science, Vol. 19 (1), 2006.
[10] Shikha Kakar, Balwinder Singh, Arun Khosla, “Implementation of BIST Capability using LFSR Techniques in UART”, International Journal of Recent Trends in Engineering ,Vol 1, No. 3, May 2009.
[11] S. Wang, “Generation of low power dissipation and high fault coverage patterns for scan-based BIST”, in Proceedings of International Test Conference, pp. 834 –843, 2002.
[12] SBS Implementers Forum. System Management Bus (SMBus) Specification, Version 2.0, August 2000.

IJDACR