International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 5, No. 3, November 2016, pp. 170~176
ISSN: 2089-4864

An FPGA Implementation of On Chip UART Testing with BIST Techniques

P Bala Gopal, K Hari Kishore
Department of ECE, K L University, India

Article Info

Article history:

Received Apr 24, 2016
Revised Aug 3, 2016
Accepted Aug 18, 2016

Keyword:

BIST
Error Check
LFSR
UART

ABSTRACT

A Universal Asynchronous Receiver Transmitter (UART) is usually implemented for asynchronous serial communication, mostly used for short distance communications. It allows full duplex serial communication link and is used in data communication and control system. Nowadays there is a requirement for on chip testing to overcome the product failures. This paper targets the introduction of Built-in self test (BIST) for UART to overcome the above two constraints of testability and data integrity. The 8-bit UART with BIST module is coded in Verilog HDL and synthesized and simulated using Xilinx XST and implemented on SPARTAN 3E FPGA. Results indicate that this model eliminates the need for expensive testers and thereby it can reduce the development time and cost.

Corresponding Author:

P Bala Gopal,
Student of VLSI Design, Department of ECE, K L University, India
Email: balagopal460@gmail.com

1. INTRODUCTION

Asynchronous serial communication has advantages of high reliability and long transmission distance. A UART allows full-duplex communication in serial transmissions, thus widely used in the data communications and control systems. So it is widely used in data exchange between Processor and peripherals. The UART converts data from parallel to serial at the transmitter side with some extra control bits using shift register and vice versa at receiver side. At the other end of the UART, it appears as an 8-bit read/write parallel port.

Basic UART communication needs only two signal lines (one is Receive, and other is Transmit) to complete full-duplex data communication. UART includes receiver and transmitter. A 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 block is used to receive the serial signals at RXD and convert them into parallel data. The UART transmitter block converts the bytes into serial bits according to the basic frame format and transmits those bits through TXD line.

When the transmitter is idle the data line is in the high logic state. If the UART is enabled to for transmission, a "Start Bit" (logic low) is added to the beginning of each word that is to be transmitted. Start Bit is used to alert the peripheral receiver that a word of data is about to be sent and to force the clock in the receiver into synchronization with the clock in the transmitter. Once the Start Bit is transmitted, the individual data bits of the word are sent. Each bit is transmitted for exactly the same amount of time as all of the other bits and the receiver samples at the wire at approximately halfway through the period assigned to each bit to determine if the bit is a ‘1’ or a ‘0’. When the entire data word has been sent, then transmitter adds a Parity Bit which has been generated in the transmitter module. The Parity Bit may be used by the receiver to perform simple error checking. After parity bit at least one Stop Bit is sent by the transmitter.

While the receiver receives all of the bits in the frame and it automatically discards the Start and Stop bits. Again if transmitter sends another frame, the Start bit for the new word can be sent as soon as the Stop bit for the previous word has been sent.

In actual applications, only a few key features of UART are needed. A specific interface chip will cause waste of resources and increased cost. Mostly in the field of electronic design, an SOC technology is becoming increasingly mature. Thus, this situation results in the requirement of realizing the whole system function in a single or a very few chips. In order to overcome waste of resources and to decrease the cost, integration of only core functions into a FPGA chip to achieve compact.

The manufacturing processes are extremely complex, so inducing manufacturers to consider testability as a requirement to assure the reliability and the functionality. Nowadays testing of integrated circuits (ICs) is important to ensure a high level of quality in product functionality in both commercially and privately produced products. Modern System-on-a-Chip (SoC) design integrates many cores into a single chip. And some of them are embedded and cannot be accessed directly from the outside of the chip. Thus testing these embedded cores is a great challenge in SOC designs.

In this paper, internal diagnostic capabilities are built into UART by the introduction of Built-In-Self-Test (BIST). The UART with status register and BIST module is coded in Verilog HDL and simulated using ModelSim and synthesized using Xilinx XST tool. The complete implementation and validation is done on Spartan 3E FPGA.

The paper is organized into 5 sections. In section 2 describes the BIST technique and implementation possibilities. The section 3 describes the architecture of UART with introduction of BIST technique. The section 4 presents the results and section 5 provides the conclusion of work.

2. BIST TECHNIQUE

VLSI testing problems like Test generation problems, gate to I/O pin ratio problems, input combinatorial problems are discussed and this made the designers to identify reliable test methods in solving these difficulties. Introducing of special test circuitry on the VLSI circuit that allows efficient test coverage is the answer to the matter. It has been addressed that there is a need of design for testability (DFT) and hence BIST gives a solution for this problem. A BIST is an on-chip test logic that is used to test the functional logic of a chip. With the rapid increase in the design complexity, a BIST has become a major design component in DFT methods and is becoming increasingly important in today's state of the art SOCs.

BIST is an exemption for an extra hardware which is used to test the chip and simultaneously it ensures the cost effective, reliability and testability. Test Pattern Generator (TPG) is a circuit used in BIST for the generation of test patterns. Figure 1 shows a BIST module composition. A generic BIST architecture component is as follows.

Circuit under Test (CUT): This is the portion of the circuit tested in BIST mode. It can be sequential circuit, or a combinational circuit or a memory. It is delimited by their Primary Input (PI) and Primary Output (PO).

Test Pattern Generator (TPG): The test patterns which are required to test the CUT are generated by TPG. The patterns may be generated in pseudorandom sequence. Normally the pattern generator generates exhaustive input test patterns to the CUT to ensure the high fault coverage.

Test Response Analysis (TRA): It compares the obtained output value with expected output value.

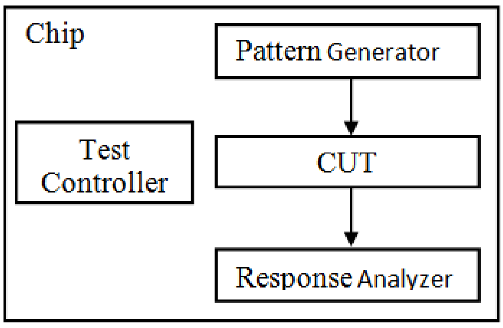


Figure 1. A Generic BIST Module

BIST Controller Unit (BCU): This unit controls all the modules; it manages the TPG and TRA. Also reconfigures the CUT and the multiplexer. When it is activated by the Normal/Test signal and generates a

Go/No-Go. When it is in test mode, test pattern generator output is connected to the CUT while during functional mode, external inputs are connected to CUT.

There are various approaches being used to generate test patterns for BIST such as LFSR, ROM, binary counters, cellular automation. The binary counter can generate an exhaustive but not randomized test sequences. The drawback of binary counters is that it requires more hardware than typical Linear Feedback Shift Register (LFSR) pattern generator. A modified counter also has been successfully worked as test-pattern generators. Thus, they also require long test sequences. While this method stores a good test pattern set in a ROM on the chip. Drawback of this approach is relatively expensive in chip area. A LFSR is used to generate pseudorandom test patterns. Normally it requires a sequence of one million or more tests pattern in order to achieve high fault coverage. The advantage of LFSR is their compact and simple design and thus is currently the preferred BIST pattern generation method. LFSR is preferred as a test pattern generator in this project.

3. PROPOSED UART ARCHITECTURE WITH BIST

The proposed architecture describes the design of an 8-bit UART and enhances the testability of circuit by the introduction of BIST module is explained in the following sections.

The proposed model has two major modules such as UART and BIST. Further in the UART, we have transmitter, receiver, and baud rate generator. BIST has a control register, pattern generator and a comparator, as shown in figure 2.

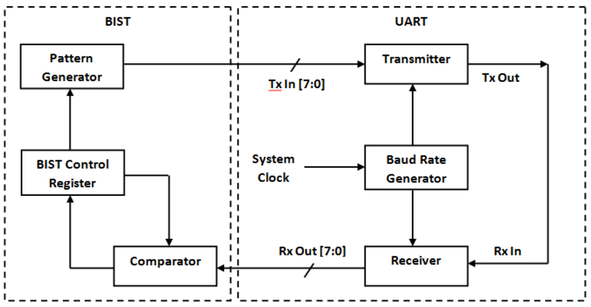


Figure 2. UART with BIST Architecture

A. UART Transmitter

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

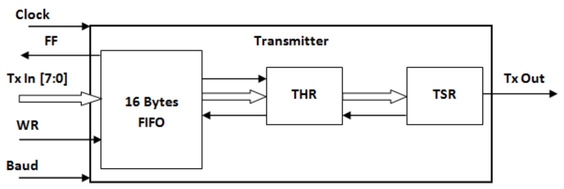


Figure 3. UART Transmitter

A Data is loaded from the parallel inputs TXIN0-TXIN7 into the Transmitter FIFO by applying logic high on the WR (Write) input. A FIFO is 16-byte register module. If FIFO is full, it sends FIFO Full (FF) signal to peripheral as shown in figure 4.

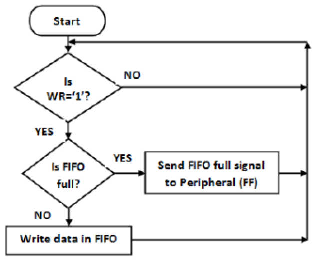


Figure 4. Transmitter Flowchart – Input to FIFO

When FIFO contains some data it will send the signal to Transmitter Hold Register (THR) which is an 8-bit temporary register. Simultaneously if THR is empty it will send the signal to FIFO which indicates that THR is ready to receive data from FIFO. If the 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-bit shift register in which framing process occurs. A frame contains start bit, data bits, parity bit and one stop bit will be added as shown in figure 6. Now the data is transmitted from TSR to TXOUT serially. Figure 5 is the flowchart explaining transmission of serial data from FIFO to transmitter output.

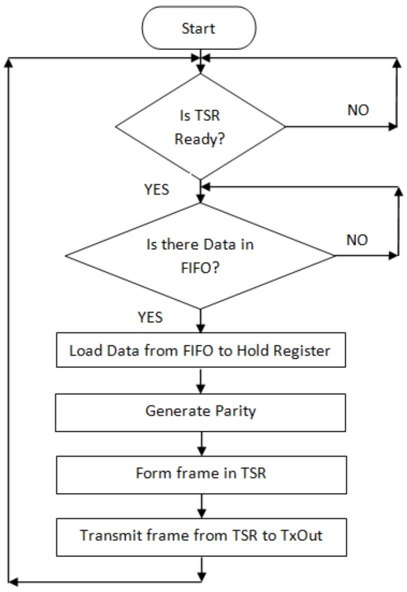


Figure 5. Transmitter Flowchart – FIFO to TXOUT

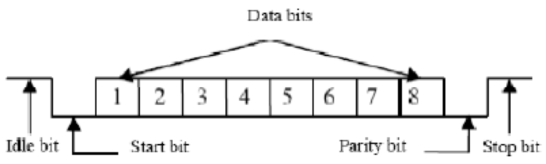


Figure 6. UART Frame Format

B. UART Receiver

The received serial data is available on the RXIN pin. 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. In the architecture of UART receiver initially the logic line (RxIn) is high as shown in figure 7.

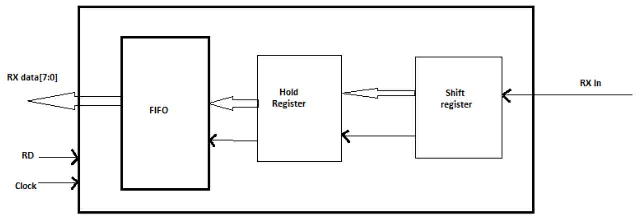
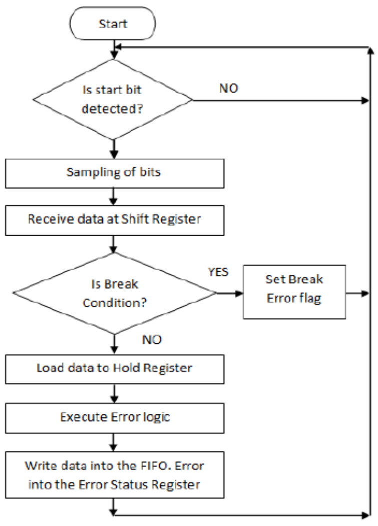


Figure 7. UART Receiver

Once the RxIn line goes low that indicates the start of frame and the receiver will be active now to receive the data transmitted by the transmitter. 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. Figure 8 shows the receiver logic.


Start
Is start bit detected?
NO
Sampling of bits
Receive data at Shift Register
Is Break Condition?
YES
Set Break Error flag
NO
Load data to Hold Register
Execute Error logic
Write data into the FIFO. Error into the Error Status Register
Figure 8. Receiver flowchart (Input to FIFO)

C. BIST Pattern Generator

Linear Feedback Shift Register is used to generate pseudo-random test patterns for the BIST. LFSR is a shift register where the input is a linear function of two or more bits also called as taps. Circuit consists of D flip-flops and linear exclusive-OR (XOR) gates. A shift registers with initial data given as a feedback which contains a linear logic. The selected bit values are collected before the register is clocked and the result of the feedback function is inserted into the shift register during the shift and filling the position that is emptied as a result of the shift.

Thus the bit positions selected for use in the feedback function are called "taps". A list of the taps is known as the tap sequence. Obtaining all zeros is not allowed in LFSR as it will always produce 0 in spite of how many clock iteration. Each state can have only once succeeding state, so an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

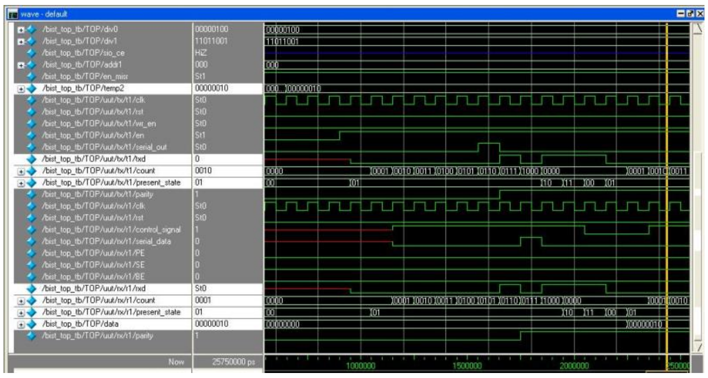
D. BIST Operation

For BIST, a UART is set in an internal loop back mode as shown in figure 2. When it is in test mode the transmitter data is connected to the receiver when it is in loop back mode. The test pattern is generated by LFSR as mentioned in the last section and the pattern is loaded to the FIFO of the UART transmitter when it is in BIST mode. Applied test byte is then padded with start, parity and stop bits and sent from transmitter and is looped back to receiver module. The receiver will extract the data from frames received and loads to receiver FIFO. And then the Tx FIFO is compared with Rx FIFO to verify the transmitted data and received data are equal or not. If FIFOs are with same data then BIST pass else BIST fail.

4. RESULTS

The Verilog HDL coding and simulation of the design are done in ModelSim simulator.

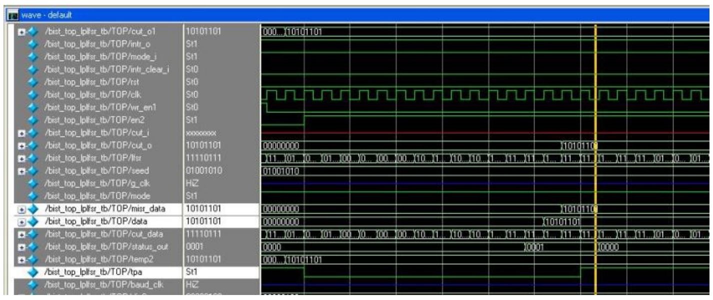
A. Simulation Results of Transmitter and receiver



```
wave - default
/bist_top_tb/TOP/div0                          00000100
/bist_top_tb/TOP/div1                          11011001
/bist_top_tb/TOP/sio_ce                        HiZ
/bist_top_tb/TOP/addr1                         000
/bist_top_tb/TOP/en_misr                       St1
/bist_top_tb/TOP/temp2                         00000010
/bist_top_tb/TOP/uut/tx/t1/clk                 St0
/bist_top_tb/TOP/uut/tx/t1/rst                 St0
/bist_top_tb/TOP/uut/tx/t1/wr_en               St0
/bist_top_tb/TOP/uut/tx/t1/en                  St1
/bist_top_tb/TOP/uut/tx/t1/serial_out          St0
/bist_top_tb/TOP/uut/tx/t1/txd                 0
/bist_top_tb/TOP/uut/tx/t1/count               0010
/bist_top_tb/TOP/uut/tx/t1/present_state       01
/bist_top_tb/TOP/uut/tx/t1/parity              1
/bist_top_tb/TOP/uut/rx/r1/clk                 St0
/bist_top_tb/TOP/uut/rx/r1/rst                 St0
/bist_top_tb/TOP/uut/rx/r1/control_signal      1
/bist_top_tb/TOP/uut/rx/r1/serial_data         0
/bist_top_tb/TOP/uut/rx/r1/PE                  0
/bist_top_tb/TOP/uut/rx/r1/SE                  0
/bist_top_tb/TOP/uut/rx/r1/BE                  0
/bist_top_tb/TOP/uut/rx/r1/rxd                 St0
/bist_top_tb/TOP/uut/rx/r1/count               0001
/bist_top_tb/TOP/uut/rx/r1/present_state       01
/bist_top_tb/TOP/data                          00000010
/bist_top_tb/TOP/uut/rx/r1/parity              1
                                         Now   25750000 ps
```

Figure 9. Simulation Result of UART Transmitter and Receiver

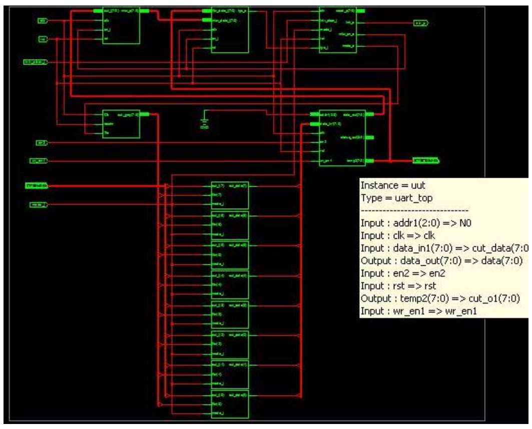
B. Simulation Results of BIST operation



```
wave - default
/bist_top_lplfsr_tb/TOP/cut_o1          10101101
/bist_top_lplfsr_tb/TOP/intr_o          St1
/bist_top_lplfsr_tb/TOP/mode_i          St1
/bist_top_lplfsr_tb/TOP/intr_clear_i    St0
/bist_top_lplfsr_tb/TOP/rst             St0
/bist_top_lplfsr_tb/TOP/clk             St0
/bist_top_lplfsr_tb/TOP/wr_en1          St0
/bist_top_lplfsr_tb/TOP/en2             St1
/bist_top_lplfsr_tb/TOP/cut_i           xxxxxxxx
/bist_top_lplfsr_tb/TOP/cut_o           10101101
/bist_top_lplfsr_tb/TOP/lfsr            11110111
/bist_top_lplfsr_tb/TOP/seed            01001010
/bist_top_lplfsr_tb/TOP/g_clk           HiZ
/bist_top_lplfsr_tb/TOP/mode            St1
/bist_top_lplfsr_tb/TOP/misr_data       10101101
/bist_top_lplfsr_tb/TOP/data            10101101
/bist_top_lplfsr_tb/TOP/cut_data        11110111
/bist_top_lplfsr_tb/TOP/status_out      0001
/bist_top_lplfsr_tb/TOP/temp2           10101101
/bist_top_lplfsr_tb/TOP/tpa             St1
/bist_top_lplfsr_tb/TOP/baud_clk        HiZ
```

Figure 10. Simulation result of BIST operation

C. Synthesis Report



```
Instance = uut
Type = uart_top
-------------------------------
Input : addr1(2:0) => N0
Input : clk => clk
Input : data_in1(7:0) => cut_data(7:0
Output : data_out(7:0) => data(7:0)
Input : en2 => en2
Input : rst => rst
Output : temp2(7:0) => cut_o1(7:0)
Input : wr_en1 => wr_en1
```

Figure 11. RTL Schematic of UART with BIST

5. CONCLUSION

The architecture of UART with BIST have been implemented by using Verilog HDL and observed simulation results using ModelSim simulator. The design architecture have been synthesized and implemented on SPARTAN 3E FPGA device and also observed hardware results. And observed that with the implementation of BIST expensive tester requirements and testing procedures starting from circuit or logic level to field level testing are minimized. LFSR replaces the function of the external testing features such as a test pattern generator by automatically generating pseudo random patterns to give good fault coverage to the UART module. The additional BIST circuit increases the hardware overhead and design time, and it eliminates the need to acquire high-end testers. Hence the reduction of the test cost helps in the reduction of overall production cost.

REFERENCES

[1] Mahat N.F, "Design of a 9-bit UART module based on Verilog HDL", *in the proceedings of 10th IEEE International Conference on Semiconductor Electronics (ICSE)*, 19-21st Sept. 2012, DOI: 10.1109/SMElec.2012.6417210, pp. 570-573.

[2] Fang Yi-yuan and Chen Xue-jun, *"Design and Simulation of UART Serial Communication Module Based on VHDL"*, in the proceedings of 3rd International Workshop on Intelligent Systems and Applications (ISA), IEEE, May 2011, DOI: 10.1109/ISA.2011.5873448, pp.1-4.

[3] Mohd Yamani Idna Idris, Mashkuri Yaacob and Zaidi Razak, *"A VHDL Implementation of UART Design with BIST Capability"*, in the proceedings of Malaysian Journal of Computer Science, June 2006, Vol. 19(1), pp. 73-86.

[4] Dr. Garima Bandhawarkar Wakhle, Iti Aggarwal and Shweta Gaba, *"Synthesis and Implementation of UART using VHDL Codes"*, in the proceedings of International Symposium on Computer, Consumer and Control, IEEE June 2012, DOI: 10.1109/IS3C.2012.10.

[5] Norhuzaimin J and Maimun H.H, *"The design of high speed UART"*, in the proceedings of Asia-Pacific Conference on Applied Electromagnetics, APACE 05, IEEE, 20-21st Dec. 2005, DOI: 10.1109/APACE.2005.1607831, pp. 5-8.

[6] Dr. T.V.S.P.Gupta, Y. Kumari and M. Ashok Kumar*, "UART realization with BIST architecture using VHDL*", in the proceedings of International Journal of Engineering Research and Applications, February 2013, Vol. 3, Issue 1, ISSN: 2248-9622, pp.636-640.

[7] Shikha Kakar, Balwinder Singh and Arun Khosla, *"Implementation of BIST Capability using LFSR Techniques in UART"*, in the proceedings of International Journal of Recent Trends in Engineering, May 2009, Vol 1, No. 3.

[8] Sybille Hellebrand, Birgit Reeb and Steffen Tarnick, *"Pattern Generation for a Deterministic BIST Scheme"*, in the proceedings of IEEE/ACM International Conference on Computer-Aided Design, ICCAD-95, Digest of Technical Papers, November 1995, DOI: 10.1109/ICCAD.1995.479997, pp. 88-94.

[9] Naresh, Vatsalkumar and Vikaskumar Patel, *"VHDL Implementation of UART with Status Register"*, in the proceedings of International Conference on Communication Systems and Network Technologies, IEEE Computer Society, 11-13th May 2012, DOI: 10.1109/CSNT.2012.164, pp.750-754.