INTERNATIONAL JOURNAL · ENGINEERING RESEARCH ·

International Journal for Innovative Engineering and Management Research

A Peer Revieved Open Access International Journal

www.ijiemr.org

COPY RIGHT

IJIEMR Transactions, online available on 4th January 2018. Link :

http://www.ijiemr.org/downloads.php?vol=Volume-7&issue=ISSUE-01

Title: A novel approach to realize Built-in-self-test(BIST) enabled UART using VHDL.

Volume 07, Issue 01, Page No: 6 - 12.

Paper Authors

*** MR. NARSAPPA REDDY, MR. KASHETTI SHARANYA, MR. MIRZA SAJID ALI BAIG , MRS. SHAHEENA KAMRANI.**

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

A NOVEL APPROACH TO REALIZE BUILT-IN-SELF-TEST(BIST) ENABLED UART USING VHDL

[1]MR. NARSAPPA REDDY, [2]MS. KASHETTI SHARANYA, [3]MR. MIRZA SAJID ALI BAIG, [4]MRS. SHAHEENA KAMRANI,

[1]Associate Professor, ECM Department, JBIET (T.S), INDIA
[2]Dept of ECE (VLSI and Embedded systems), P.Indra Reddy Memorial Engineering College, (T.S),INDIA, ,
[3]Associate Professor. Dept of ECE , Patnam Rajender Reddy Memorial Engineering College (T.S), INDIA.
[4]Associate Professor. Dept of ECE, Dr VRK women's college of Engineering and technology, aziznagar, (T.S), INDIA.
narsapa@gmail.com sharanya.kashetti011@gmail.com

ABSTRACT:

Testing of VLSI chips are becoming very much complex day by day due to increasing exponential advancement of nano technology. So both front-end and back-end engineers are trying to evolve a system with full testability keeping in mind the possibility of reduced product failures and missed market opportunities. BIST is a design technique that allows a system to test automatically itself with slightly larger system size. In this paper, the simulation result performance achieved by BIST enabled UART architecture through verilog programming is enough to compensate the extra hardware needed in BIST architecture. This technique generate random test pattern automatically, so it can provide less test time compared to an externally applied test pattern and helps to achieve much more productivity at the end

I INTRODUCTION

The processing steps of VLSI chips are extremely complex, and costly inducing vendors to stress on more and more testability as a requirement tool to assure the reliability and the functionality of each of their designed circuits. BIST technique has become as a boon to them, which helps to test a system automatically. Universal Asynchronous Receive/Transmit (UART) has the objectives of firstly to satisfy specified testability requirements, and secondly to generate the lowest-cost with the highest performance implementation. UART has been an important input/output tool for decades and is still widely used. The additional BIST circuit that increases the hardware overhead increases design time and size of the chip, which may degrade the performance.

UART (Universal Asynchronous Receiver Transmitter) is a kind of serial communication protocol; mostly used for short-distance, low speed, low-cost data exchange between computer and peripherals. During the actual industrial production, sometimes we do not need the full functionality of UART, but simply integrate its core part. UART includes three kernel modules which are the baud rate generator, receiver and transmitter. The UART implemented with verilog language can be integrated into the FPGA to achieve compact, stable and reliable data transmission. It's significant for the design of SOC. The simulation results are completely consistent with the UART protocol.

A UART is usually an individual (or part of an) used for over a computer or peripheral device. UARTs are now commonly included in microcontrollers. A UART (Universal Asynchronous Receiver/Transmitter) is the microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C Data Terminal Equipment (DTE) interface so that it can "talk" to and exchange data with modems and other serial devices. Serial transmission is commonly used with modems and for non-networked communication between computers, terminals and other devices.

The Universal Asynchronous Receiver/Transmitter (UART) takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. Each UART contains which is the fundamental method of conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is much more cost effective than parallel transmission.

Asynchronous serial communication has advantages of less transmission line, high reliability, and long transmission distance, therefore is widely used in data exchange between computer and peripherals. Asynchronous serial communications usually implemented by Universal Asynchronous Receiver Transmitter (UART) [1]. UART allows full-duplex communication in serial ink, thus has been widely used in the data communications and control system. In actual applications, usually only a few key features of UART are needed. Specific interface chip will cause waste of resources and increased cost. Particularly in the field of electronic design,SOC technology is recently becoming increasingly mature. This situation results in the requirement of realizing the whole system function in a single or a very few chips. Designers must integrate the similar function module into FPGA. This paper uses VHDL to implement the UART core functions and integrate them into a FPGA chip to achieve compact, stable and reliable data transmission, which effectively solves the above problem [2] [3].

II.LITERATURE SURVEY

Digital systems are composed of data paths, control paths and memories. The low cost of memory and high memory demand of high-speed DSP circuits and generic microprocessors have made the memory subsystem an important focus of the design. Defects in memory arrays are generally due to shorts and opens in memory cells, address decoder and read/write logic. These defects can be modeled as single and multicell memory faults [6, 10]. Memories are more likely to fail than random logic and therefore three classes of memory tests have been proposed to detect the memory faults [10, 1, 2]. Application of test sequences to embedded memories using on-chip testers results in a high test time and test cost due to the large size of embedded memories. To overcome this problem, the computed test sequences are generated on-chip using a memory Built-In Self Test (BIST) unit.

A memory BIST unit consists of a controller to control the ow of test sequences and other components to generate the necessary test control and data. A memory BIST controller could be designed as a Finite State Machine (FSM)-based or microcode-based controller. The FSM-based controllers are the hardware realization of a selected memory test algorithm. This type of memory BIST architecture has optimum logic overhead, however, lacks the edibility to accommodate any changes in the selected memory test algorithm. This results in re-design and re-implementation of the FSM-based controller for any minor changes in the selected memory test algorithm. In microcode-based controllers, a selected memory test algorithm is written in terms of a set of supported instructions and loaded in the memory BIST controller. This type of memory BIST architecture allows changes in the selected test algorithm with no impact on the hardware of the controller.

This edibility might result in higher logic overhead for the memory BIST controller [8]. However, the additional logic overhead, if kept in reasonable levels, could be justified by using the memory BIST to reduce the cost of diagnostics. Memories undergo different type of testing during the course of their design and fabrication.

Tests necessary for embedded memories and diagnostics have a set of requirements on the memory BIST controller. To satisfy these requirements additional hardware might have to be added to the memory BIST unit.

Comparison of logic overhead of non-programmable versus programmable memory BIST architectures considering only the logic overhead of only one type of memory test or diagnostics might not truly reveal the overhead of one architecture over another. Therefore, to compare and evaluate deferent memory BIST architectures, the overall testing and diagnostics requirements of memories and the exibilities of each memory BIST architecture must be considered.

Programmable memory BIST controllers that could accommodate testing requirements of embedded memories were proposed in [4, 3, 9]. The drawback of the method described in [4] is that only memory tests where each component does not have more than two instructions could be coded efficiently.

Furthermore, it is not clear if their architecture has the flexibility to accommodate the test requirements of the memories in different stages of their fabrication . The programmable architecture described in [3] considers dividing a test algorithm into smaller sub-tests and loading the necessary microcode's through multiple loads. This is time consuming and might not always be feasible. Also, this architecture does not support optimum representation of the symmetric test algorithms and test algorithms that consist of several smaller loops such as march test algorithms [5]. The architecture described in [9] is mainly designed for diagnostics and process monitoring. In this approach, the test algorithm is loaded in a 3214 SRAM using an initialization sequence. This SRAM has 3 to 4 times larger size than the previous programmable methods. Also, testing the SRAM itself adds to the complexity of the overall testing of the design.

III. METHODOLOGY

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.

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 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.

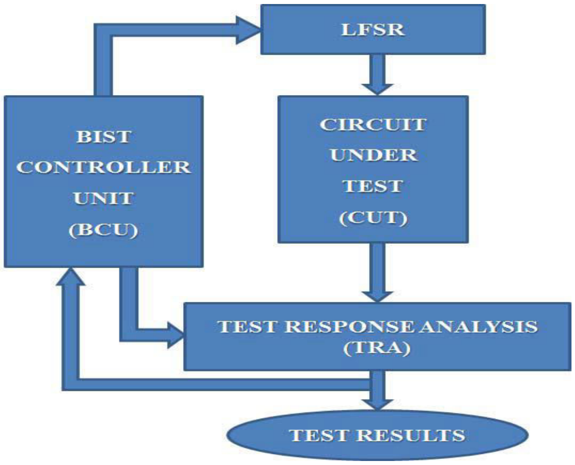


Figure 1. BIST architecture

LFSR (BIST TEST PATTERN GENERATION):

LFSR of polynomial X8+X6+X5+1 is made. A PISO with one 3 bit down counter & one 1 clk delay component is connected. O/P from all DFF comes to the PISO , PISO has another two I/P sel & Reg_load & one PS O/P which serially fed to the receiver.3 bit down counter with 1 CLK delay is connected to the Reg_load of PISO. Now whensel is zero O/P from DFF is fed to the receiver serially with every count of the down counter (counter CLK is the trigger signal from comparator). Now when counter value goes to zero i.e 8 bit data has been send to the receiver then PISO will send all 8 bit data parallel to the transmitter in one clock period.

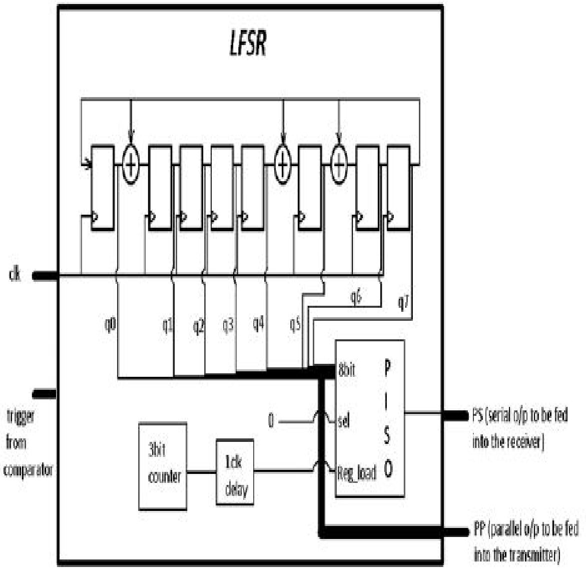
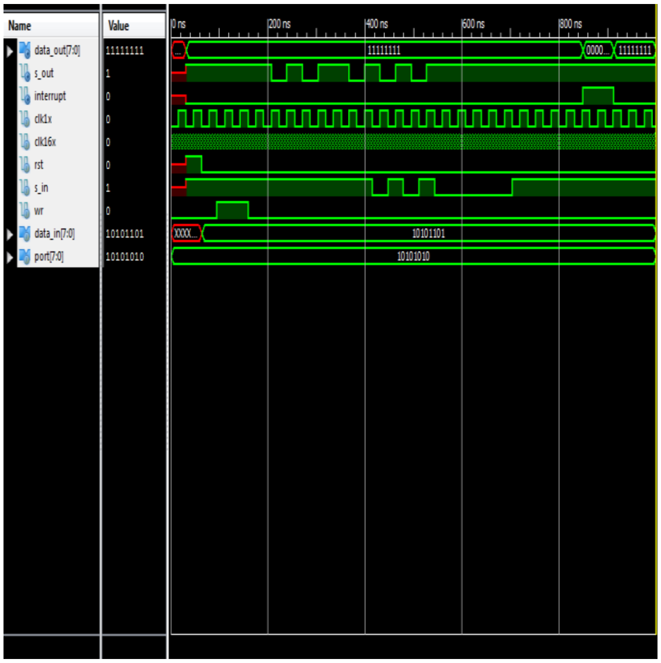


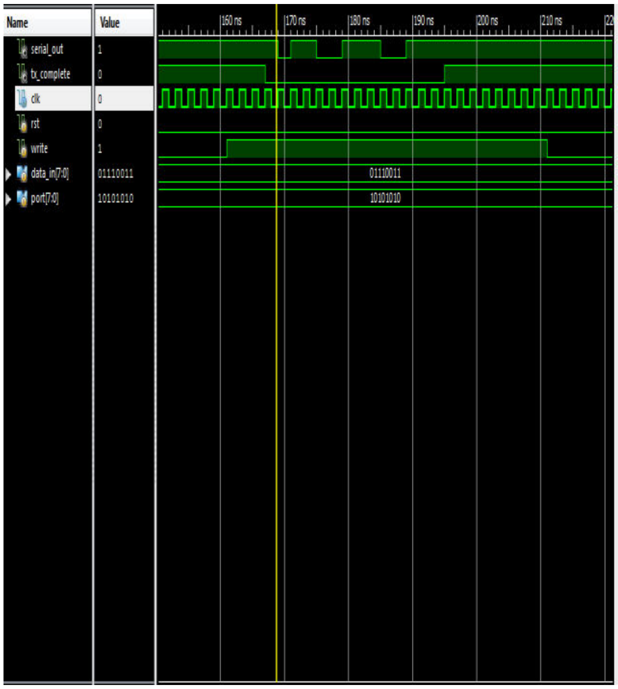
Figure 2: Port mapping and the components of LFSR

IV.RESULTS



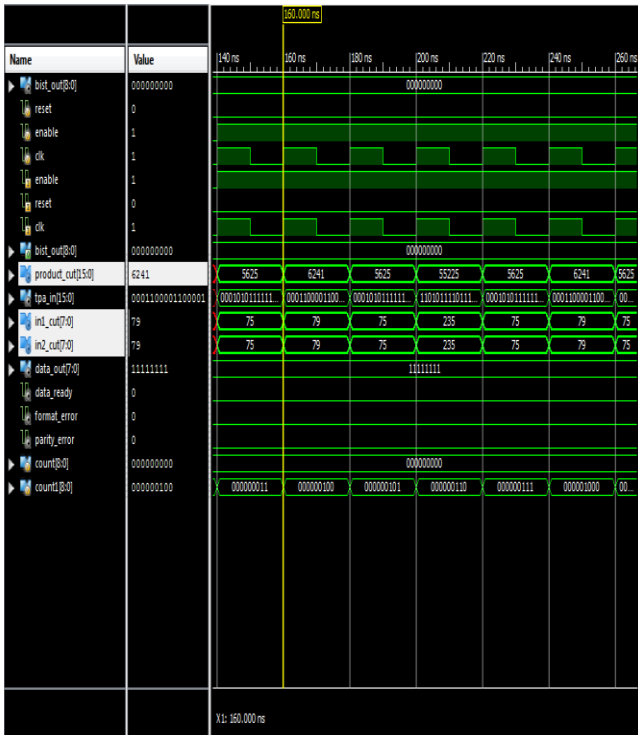
Name	Value
data_out[7:0]	11111111
s_out	1
interrupt	0
clk1x	0
clk16x	0
rst	0
s_in	1
wr	0
data_in[7:0]	10101101
port[7:0]	10101010

Figure 3: Output of UART



Name	Value
serial_out	1
tx_complete	0
clk	0
rst	0
write	1
data_in[7:0]	01110011
port[7:0]	10101010

Figure 4: UART output 2



Name	Value
bist_out[8:0]	000000000
reset	0
enable	1
clk	1
enable	1
reset	0
clk	1
bist_out[8:0]	000000000
product_cut[15:0]	6241
tpa_in[15:0]	0001100001100001
in1_cut[7:0]	79
in2_cut[7:0]	79
data_out[7:0]	11111111
data_ready	0
format_error	0
parity_error	0
count[8:0]	000000000
count1[8:0]	000000100

Figure 5: BIST interfaced with UART output

V.CONCLUSION

This paper implements the verilog HDL language based coding for BIST technique on UART. UART transmitter and receiver sections are differently tested by BIST technique. The transmitter is given a parallel 8 bit input and the serial output of the transmitter is checked with the respective ideal output stored in the ROM. The receiver is given a serial 8 bit input and the parallel output of the receiver is checked with the respective ideal output stored in the ROM. The checking part is done by comparator section. It compares the receiver or transmitter output with the corresponding data stored in the ROM. Using memory BIST has various advantages such as no external test equipment, reduced development efforts, at-speed tests. However, there are many challenges associated with it such as silicon area overhead, extra pins and routing.

In addition, the testability of the test hardware itself is another difficult task. An improvement can be made in the field of implementation of the Test Response Analyzer (TRA). Rather than using ROM as TRA, Multiple Input Shift Register (MISR) can be used for the same purpose. All the simulation result presented in our paper shows that the BIST enabled UART is working well. Hardware test by FPGA Tool(Spartan2E) showed the same result as in the simulation result. The synthesis report that we got from the XILINX tool shows that it provides excellent result compared to the previous work.

VI. REFERENCES

[1] Mohd Yamani Idna Idris, Mashkuri Yaacob, Zaidi Razak, “A VHDL Implementation Of UART Design with BIST capability”

[2] Dr. T.V.S.P. Gupta, Y. Kumari,M.Asok Kumar”UART realization with BIST architecture using VHDL” International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, January -February 2013, pp.636-640

[3] M.S. Harvey, Generic UART Manual, Silicon Valley,December 1999

[4] P. J. Anderson , “The designer’s guide to VHDL” , Morgan Kaufman , [5] 2nd edition, 2002.

[5] Neil H.E. Weste, Kim Haase, David Harris , A. Banerjee , “CMOS VLSI Design: A circuits and Systems Perspective”, Pearson Education.

[6] K. Zarrineh, and S. J. Upadhyaya, “On programmable memory built-in self test architectures”, Design, Automation and Test in Europe Conference and Exhibition1999.Proceedings 1999, pp. 708 - 713.

AUTHORS

Mr.Narsappa reddy, M.Tech(Ph.D), Associate Professor, ECM Department JBIET (T.S), INDIA. He has total Teaching Experience (UG and PG) of 17 years. He has guided and co-guided 57 P.G students. His Research areas included VLSI system Design, Digital signal Processing, Embedded system.

Ms. Kashetti sharanya, PG Scholar, Dept of ECE (VLSI and Embedded systems), P.Indra Reddy Memorial Engineering College, (T.S),INDIA,

Mr. Mirza Sajid Ali baig , Associate Professor. Dept of ECE , Patnam Rajender Reddy Memorial Engineering College (T.S), INDIA.

Mrs. Shaheena kamrani, Associate Professor. Dept of ECE, Dr VRK women's college of Engineering and technology, aziznagar, (T.S), INDIA.