ISSN : 2348 - 9065 (Online)
ISSN : 2349 - 3143 (Print)

BIST Enabled UART for Real Time Interface Applications using FPGA

[I]M. Mamatha, [II]Vijaykumar R. Urkude
[I]M. Tech in VLSI at Vignan Institute of Engineering and Technology
[II]Associate Prof., Dept. of Electronics and Comm. Engg. in Vignan Institute of Engg. and Technology

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 VHDL 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[1].
In this project, the protocol of BIST Enabled UART is studied. The BIST Enabled UART architecture will be designed. Various blocks of BIST Enabled UART are modeled in VHDL. The design is functionally verified by simulating the code in Modelsim from Mentor Graphics. The FPGA synthesis is done using Xilinx ISE tool. The synthesis results of ISE are analyzed for timing and area. Various applications of designed BIST Enabled UART are studied.

Keywords
BIST, UART, VHDL, protocol, ATE.

I. Introduction

Manufacturing processes are extremely complex, inducing manufacturers to consider testability as a requirement to assure the reliability and the functionality of each of their designed circuits. One of the most popular test techniques is called Built-In-Self-Test (BIST). A BIST 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. Although BIST techniques are becoming more common in industry, the additional BIST circuit that increases the hardware overhead increases design time and performance degradation is often cited as the reason for the limited use of BIST [2].
BIST technique has become as a boon, 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 is an important input/ output tool for decades. The additional BIST circuit that increases the hardware overhead increases designs time and size of the chip, which may degrade the performance. This paper focuses on the design of a UART chip with embedded BIST architecture using VHDL language [3].
The paper describes the problems of (VLSI) testing followed by the behavior of UART that includes both transmitter and Receiver section. Serial data is transmitted via its serial port. A serial port is most universal parts of a computer. It is a connector where serial line is attached and connected to peripheral devices such as mouse, modem, and printer and even to another computer. In contrast to parallel communication, these peripheral devices communicate using a serial bit stream protocol (where data is sent one bit at a time). The serial port is connected to UART, an integrated circuit which handles the conversion between serial and parallel data.

II. UART

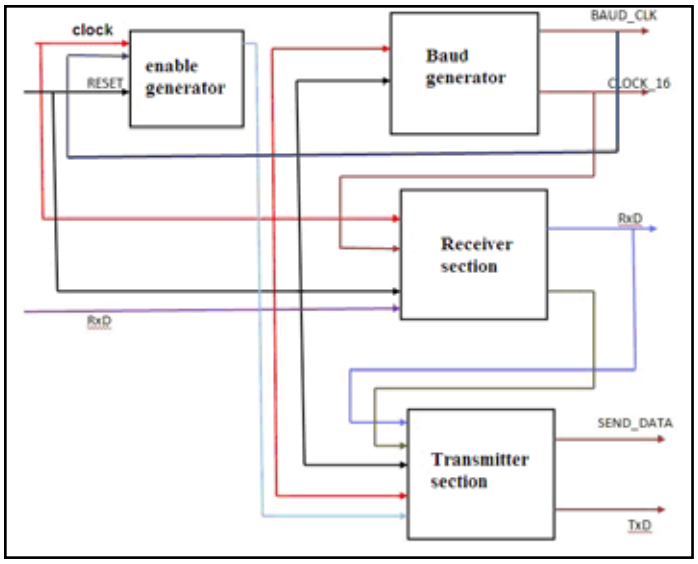

Fig. 1 : UART block diagram

The UART block diagram is shown in Figure 1. It consists of 5 blocks namely transmitter, receiver, register block, parity generator and clock divider. The UART has serial interface to the RS232 driver. The operation of UART is controlled by an external host processor. There is an 8-bit data interface to host along with read and write control signals. Clock is fed from external crystal [4].

A. Baud Generator

Baud generation section is a clock divider circuit, FPGA board clock runs at 50MHz, but UART transfer data at predefined standards that had to be maintained, in present system is designed for a rate of 9600/sec(i.e $50x10^6$ is scaled down for 9600). generates a 9600 pluses for a sec, this implies the speed of UART is 9600 bits per sec. another clock with a 16 times faster is required to the receiver section so that the data will not be corrupted, baud out is given to the enable generator section.

B. Enable Generator

This section receives baud_clock signal as a enable signal and gives enable out signal to the transmitter section as a enable input signal. This signal is used to synchronize the transmitter section when ever the data is to transferred.

C. Transmitter

The transmitter block is responsible for the transmission of serial data from UART. It takes 8-bit data from the receiver section (in this architecture it takes data after processing image operation block) in parallel and send data in serial form. Data inserted between start and stop bits. An optional parity bit also may be used for error detection. State machine for transmitter is shown in Figure 2.

Transmitter stays in IDLE state unless transmit enable (tx_enable) is made as '1'.The data transmission starts with tx_enable = 1. As mandated by the protocol, a '0' is transmitted to indicate start of transmission or start bit. This is done in START state. Then data bits 0 to 7 are transmitted in states DATA0 to DATA7. If parity is enabled in configuration register, the data is attached with parity in PARITY state. Then transmitter enters STOP state and sends a '1'. This indicates the completion of transmission. Then the transmitter enters the IDLE state and waits for next data transmission [5].

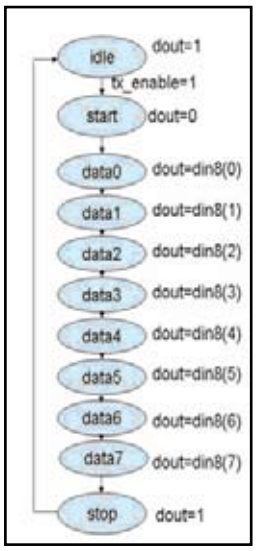

Fig. 2 : UART transmitter state machine

D. Receiver

UART receiver handles reception of data from RS232 port. Main functions of receiver block are to convert the serial data to parallel data, check the correctness of data from parity and store the received data. UART receiver state machine is shown in Figure 3. The receiver is in IDLE state by default. When the serial data pin goes low, indicating the start bit, the state machine enters DATA0 state. The data is received; one bit at a time from LSB to MSB in states DATA0 to DATA7. If parity is enabled, the state machine checks the parity bit received against the parity obtained from received data. If the data received is fine, the data_rx (data_rx_done) bit is set to '1' and the receiver goes back to IDLE state again[6].

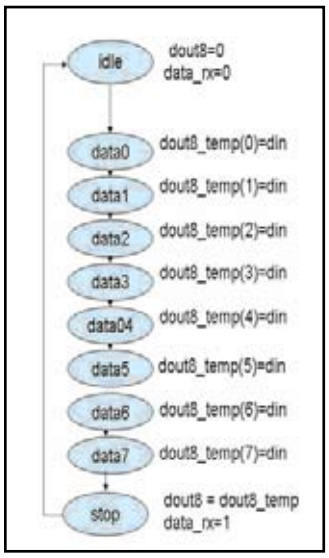

Fig. 3 : UART receiver state machine

III. Bist technique

Built-In Self Test is a technique of integrating the functionality of an automatic test system onto a chip. It is a Design for Test technique in which testing (test generation and test application) is accomplished through built in hardware features. The general BIST architecture has a BIST test controller which controls the BIST circuit, test generator which generates the test address sequence, response verification as a comparator which compares the memory output response with the expected correct data and a circuit under test (CUT) [7].

A Field-Programmable Gate Array (FPGA) is a logic device that can be programmed to implement a variety of digital circuits. FPGA's are widely used both in product prototyping and development because of their ability for configuration and re-configuration. Some of the advantages are reduced design time and implementation cycles, the low non-recurring engineering cost. FPGA consists of an array of configurable logic blocks inter connected by programmable routing resources, and programmable 110 cells.

In contrast, conventional BIST approaches introduce both area overhead (typically between 10 and 30 percent) and delay penalties. Our approach is applicable to any in-circuit reprogrammable FPGA, such as SRAM-based FPGAs. However, with the increase in density, capability and speed, FPGAs have become more vulnerable to faults, as it is the case for all circuits. A percentage of manufactured FPGA chips are determined to be faulty after initial application-independent tests. Faulty FPGAs can also be found after delivery to users, during the system development or operation. They may be still usable for some particular application if only a portion of the circuitry is defective [8].

ISSN : 2348 - 9065 (Online)
ISSN : 2349 - 3143 (Print)

IV. Implementation

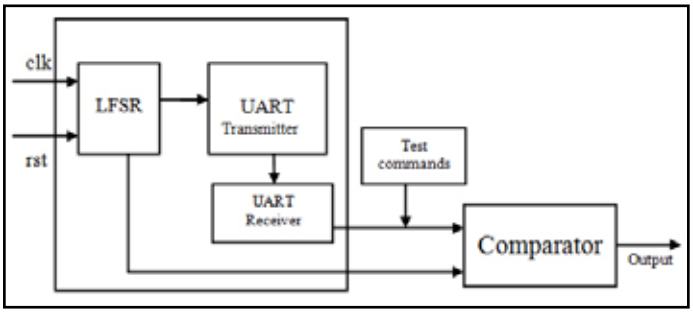

Fig. 4 : Implementation of BIST enabled UART block diagram

The above figure 4 shows the block diagram of UART with a BIST capability. The block diagram of BIST enabled UART consists of LFSR, UART transmitter, UART receiver, comparator.
LFSR (Linear Feedback Shift Register) generates 8-bit random data. This serial data coming from LFSR is fed to transmitter which converts serial data to parallel. This 8-bit parallel data coming from transmitter is given to receiver and this data is used when the test enable is in high condition which means circuit is in testing mode. The receiver takes 8-bit parallel data and it gives serial data.
This 8-bit serial data coming from receiver is given to comparator. Comparator one input is coming from receiver output and the other input from LFSR. If the data coming from receiver is same as LFSR data, then the UART is working properly, and it is indicated by BIST fail flag as zero. In case, the data is not same, then UART is misbehaving, BIST fail becomes one.
The UART developed here as the capability of working with the testing mode that means at any time if circuit misbehaves then BIST fail flag indicates the same. In this design test enable is meant foe only providing the all test cases for the design in real time use this will not be there. Whenever test enable is low the UART is going into normal working mode. The design results in effective working UART, because whenever any module of UART fails working, BIST will automatically generates a flag indicating that UART is misbehaving.

V. Simulation Results

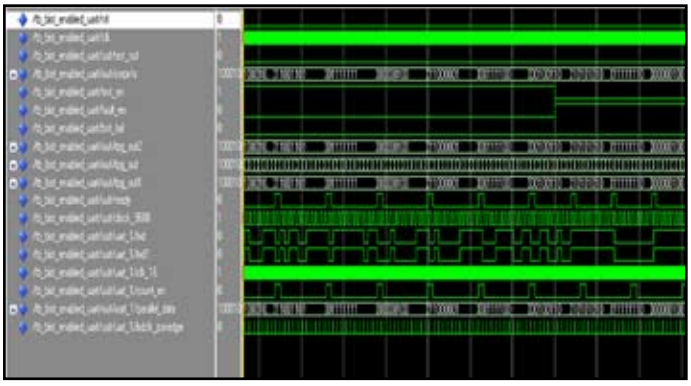

Fig. 5 : Simulation results

VI. Conclusion

In this paper, the architecture of BIST enabled UART was designed and various blocks of bist and uart are modeled in VHDL. The design is functionally verified by simulating the code in ModelSim from Mentor Graphics. The FPGA synthesis is done using Xilinx ISE tool and the design is implemented on SPARTAN 3E FPGA, which describes the characteristics and the architecture of the designed UART with embedded BIST.

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, 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 Exhibition 1999. Proceedings, 1999, pp. 708 -713.*
[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.*

Authors Profile

M. Mamatha is presently pursuing final semester, M. Tech in VLSI at vignan institute of engineering and technology.

Vijaykumar R. Urkude is presently working as Associate Professor in the department of Electronics and Communication Engineering in vignan institute of engineering and technology.