International Journal of Emerging Engineering Research and Technology
Volume 3, Issue 12, December 2015, PP 179-184
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Design of UART with BIST Capability using Cellular Automata

Akki. Siva Gopi[1], C.H Pushpalatha[2]

[1]Department of ECE, Gonna Institute Of Technology, Vishakhapatnam, India (PG Scholar)
[2]Department of ECE, Gonna Institute Of Technology, Vishakhapatnam, India (Associate Professor)

ABSTRACT

In today's life the most manufacturing process are extremely complex, including manufactures to consider testability as a requirement to assure the reliability and functionality of each of their designed circuits. One of the most popular test technique is called built-in –self-test (BIST). BIST is a design technique that allows a system to test automatically itself with slightly larger system size. a universal asynchronous receiver and transmitter(UART) with enabled BIST capability has the objective of testing the UART on chip itself and no external devices are required to perform the test. This paper focuses on the TPG (test pattern generator) circuit of BIST, 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 the 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.

Keywords: *VLSI, BIST, UART, VHDL, Cellular automata.*

INTRODUCTION

Universal Asynchronous Receiver and Transmitter (UART) is a kind of serial communication protocol; mostly used for short-distance, low speed, low-cost, data exchange between computer and peripherals. UARTs are used for asynchronous serial data communication by converting data from parallel to serial at transmitter with some extra overhead bits using shift register and vice versa at receiver. It is generally connected between a processor and a peripheral, to the processor the UART appears as an 8-bit read/write parallel port.

This paper focuses on the design of a UART chip with Enabled BIST architecture using CA (cellular automata) LFSR with the help of VHDL language. Built-In-Self-Test or BIST, is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation using their own circuits, thereby reducing dependence on external Automated Test Equipment (ATE) [6]. BIST is a Design-For-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to.

The rest of paper is as follows. Related Work in section II. Section-III describes the proposed TPG, Section-IV describes the UART, Section-V describes the architecture of UART with BIST, and Section-VI describes the simulation results and conclusion.

RELATED WORK

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.

Address for correspondence:

Asivagopi489@gmail.com

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/ORA): 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.

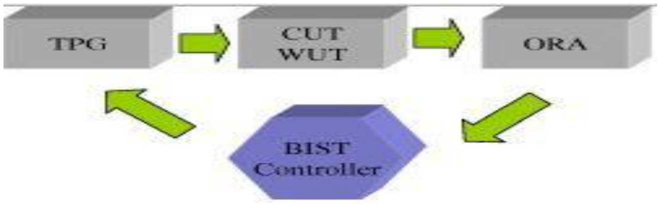


Fig. *Generic BIST Architecture*

PROPOSED TPG

In previous design normal LFSR design used in TPG circuit, But proposed design consist the **Cellular Automata** is used. Test pattern generation is the most critical operations of BIST. Test patterns used in most implementations are pseudo-random in nature i.e. the random numbers are generated algorithmically and are repeat-able [4]. This is a desired characteristic, as truly random test patterns will lead to different fault coverage in every execution [2]. LFSRs are most commonly used to build TPGs [6] but recently there has been interest in CA for test pattern generation. CA generates test vectors which are more random in nature. Highly random vectors help in detection of faults such as the stuck-open faults, delay faults etc. which cannot be easily detected by vectors generated by LFSR.

CA:

Cellular Automata (CA) consists of a collection of cells/nodes formed by flip-flops which are logically related to their nearest neighbors using XOR gates [2] [4]. When the value of a node is determined only by two neighboring cells the CA is known as one-dimensional linear CA (for the rest of the text one-dimensional linear CA is referred as a CA). The logical relations which relate a node to its neighbors are known as *rules* and they de-fine the characteristics of a CA. There are many rules which can be used to construct a CA register, the most popular being rules 90 illustrated in Figure.

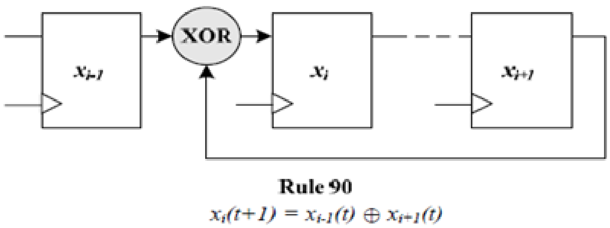


Fig2. *CA 90 Rule*

The CA pattern generator is designed by using CA rule 90, which generate the random values in TPG for UART; the CA is shown in figure 3.

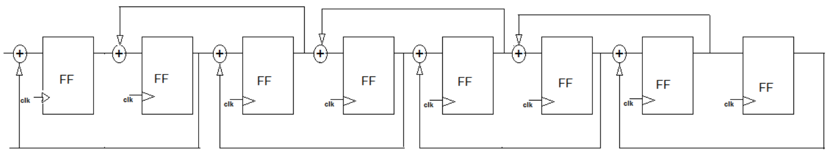


Fig. *CA*

When the trg signal goes high, a new data is obtained from the CA which is fed in parallel to the input of the transmitter, and the ca output is given to PISO (parallel in serial out) block in TPG, it generates the serial data to UART receiver input. The complete TPG block is shows in fig.

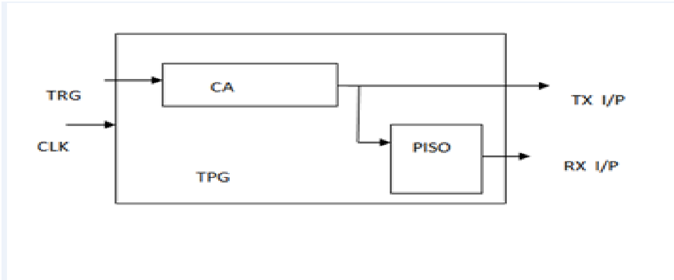


Fig. *TPG block*

UART

Universal asynchronous receive transmit (UART) is an asynchronous serial receiver/transmitter. It is a piece of computer hardware that commonly used in PC serial port to translate data between parallel and serial interfaces. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the receiving point, UART re-assembles the bits into complete bytes. Asynchronous transmission allows data to be transmitted without having to send a clock signal to the receiver. Thus, the sender and receiver must agree on timing parameters in advance and special bits are added to each word, which is used to synchronize the sending and receiving units. In general, UART contains of two main block, the transmitter and receiver block.

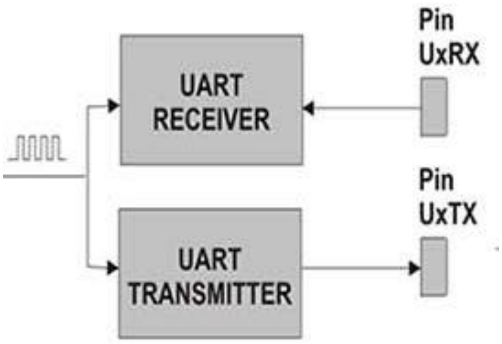


Fig. *UART Blocks*

UART Transmitter Section

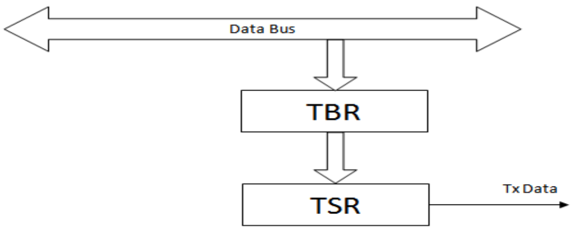


Fig. *UART Transmitter Section*

The Block diagram of UART Transmitter is as shown in figure. The data is loaded from Data Bus into TBR (Transmit Buffer Register) and from TBR to TSR (Transmit Shift Register), based on the control and status signals produced by the Control unit. The Size of TSR is taken in such a way that, it should accommodate the START and STOP bits along with the Data bits which are loaded from the Data Bus.

UART Receiver Section

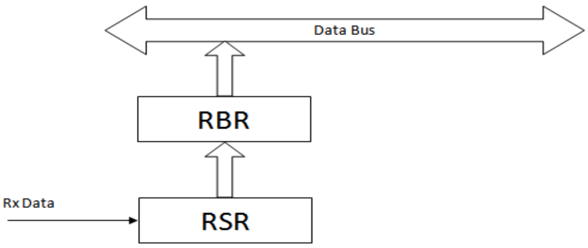
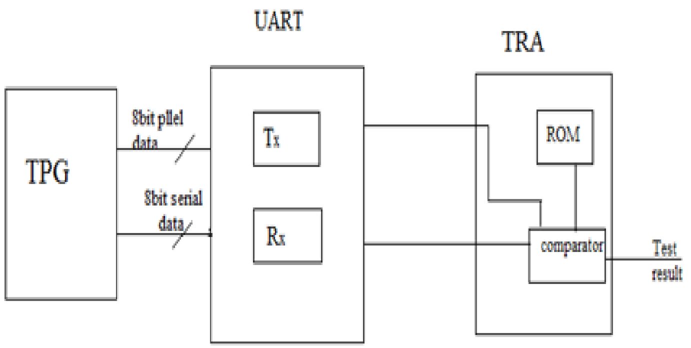


Fig. *UART receiver section architecture*

The Block diagram of UART Receiver is as shown in figure. The data receiving will be captured using receiving baud clock and then loaded into RSR (Receive Shift Register) and from RSR to RBR (Receive Buffer Register), and then to Data Bus, based on the control and status signals produced by the Control unit. The Size of RSR is taken in such a way that, it should accommodate the START and STOP bits along with the Data bits which are loaded from the Data Bus.

UART BIST DESIGN



When the trigger signal goes high then only a new value is generated in the CA. This CA will generate (2^8-1) different pseudorandom values,. The parallel output "TX i/p" is fed to the transmitter and the serial output is fed to the receiver section of the UART.

When the trg signal goes high, a new data is obtained from the CA which is fed in parallel to the input of the transmitter. After a certain number of clock cycles, the same data is obtained at the output of the transmitter as serial data. This serial data is shifted in the SIPO of the comparator and this is the sipo_op data. This sipo_op is compared with the ROM data (romd). The shaded regions in the figure shows the time of comparison. Depending on the comparison, the result (i.e. rslt) is generated. If the both sipo_op and romd are same then rslt=1, else rslt=0.

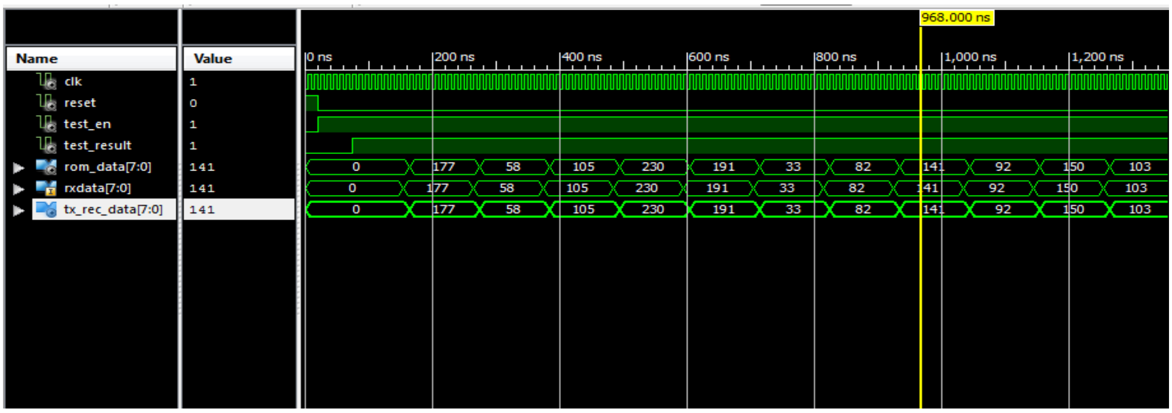
As the receiver provides 8-bit parallel output, a SIPO is not required in this case. The output from the receiver "rop" is directly compared with romd. The shaded regions in the figure too show the comparison time. In all the shaded regions, except the last one romd and rop are same and so the rslt is 1 in all those cases. But in the last comparison, due to the intentionally stored incorrect data in the ROM the comparison results in rslt=0. Henceforth comparison procedure is stopped and the CUT is announced to be faulty.

ROM

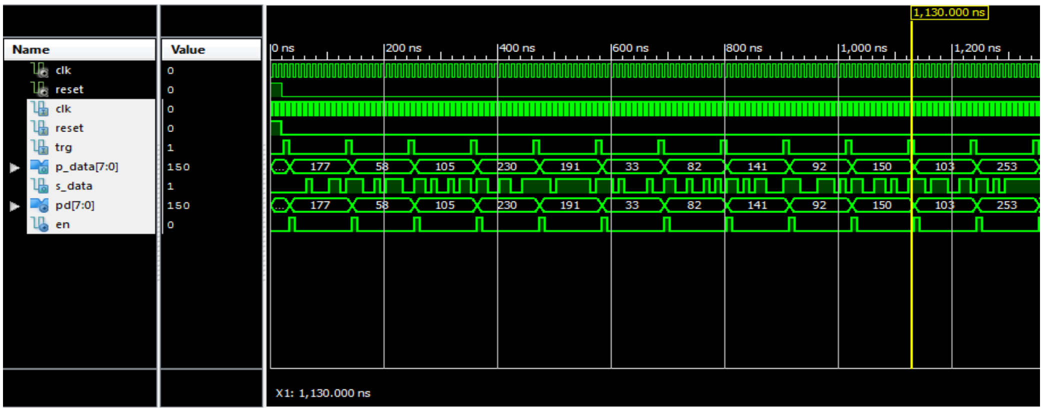
ROM is used to store the 8-bit pseudo random patterns, in order, that will be obtained as the output from the Transmitter and Receiver sections of the developed UART. The data that are obtained as the outputs of the receiver and the transmitter are compared with the data stored at the corresponding addresses of the ROM by the comparator which verifies whether the CUT is working properly or not.

SIMULATION RESULT

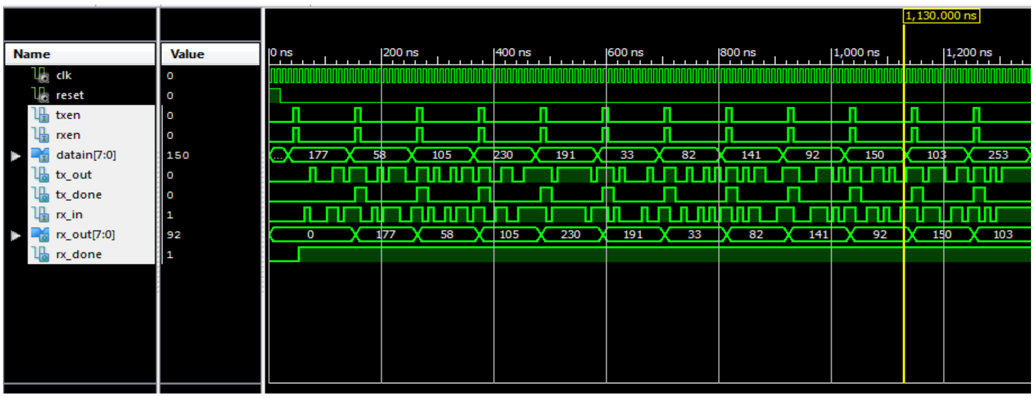
BIST Enabled UART result



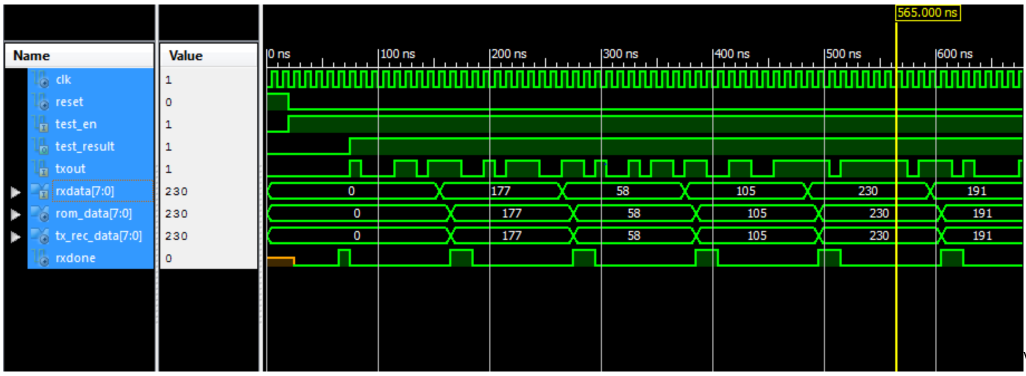
TPG Results



UART



TRA



CONCLUSION

This paper implements the BIST Enabled UART with cellular automata (CA) as TPG and total design Using VHDL language, enabled BIST test the UART transmitter and receiver modules by comparing both outputs at TRA unit.

REFERENCES

[1] S. Zhang, R. Byrne, J.C. Muzio, D.M. Miller, “Why cellular automata are better than LFSRs as built-in self-test generators for sequential-type faults”, IEEE International Symposium on Circuits and Systems, Vol. 1,1994

[2] C. Stroud, *A Designer’s Guide to Built-In Self-Test*, Kluwer Academic Publishers, Bos-ton MA, 2002

[3] M.L. Bushnell, V.D. Agrawal, *Essentials of Electronics Testing for Digital, Memory & Mixed Signal VLSI Circuits*, Kluwer Aca-demic Publishers, Boston MA, 2000

[4] K. Furuya, E.J. McCluskey, “Two-Pattern test capabilities of autonomous TPG circuits,”

[5] *Proc. of International Test Conference*, pp 704 – 711, 1991.

[6] P.H. Bardell, W.H. McAnney, J. Savir, *Built-in test for VLSI: Pseudorandom Techniques,* John Wiley and Sons, New York, 1987

[7] D. Bhavsar and R. Heckelman, "Self Testing by Polynomial Division," *Proc. IEEE International Test Conference*, pp. 208 – 216, 1981.

[8] __, "Linear Feedback Shift Register," en.Wikipedia.org

AUTHOR'S BIOGRAPHY

Akki. Siva Gopi, has received her B.Tech Degree in 2012 Electronics and communication engineering from SANA ENGINEERING College, KODADA and pursuing M.Tech in VLSI DESIGN from GONNA INSTITUTE OF INFORMATION TECHNOLOGY & SCIENCES , Vishakhapatnam

C.H Pushpalatha. has completed her B.Tech in 2011 Electronics and communication engineering from NIE Guntur and completed her M.Tech in AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY Vishakhapatnam. Now working as an associate professor in GONNA INSTITUTE OF INFORMATION TECHNOLOGY & SCIENCES , Vishakhapatnam.

.