IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

DESIGN AND IMPLEMENTATION OF BIST TECHNIQUE IN UART SERIAL COMMUNICATION

M.Hari Krishna*, P.Pavan Kumar

* Electronics and Communication Engineering Dept. GOKUL College of Engineering Vizianagaram, India.

Electronics and Communication Engineering Dept. GOKUL College of Engineering Vizianagaram, India.

ABSTRACT

This paper describes a design and implementation of BIST technique in UART serial communication. Asynchronous serial communication is usually implemented by uart which is mostly used for less distance, low speed, low cost data to exchange between processor and peripherals. But due to the errors produced in the output of the data received the circuits are being not performed well in the functions In order to reduce the possibility of product failures and missed market opportunities by providing the need to ensure the data to be transferred in error proof. So with the proposed architecture of bist in uart we can reduce expensive tester requirements and testing procedures in circuit are minimized and it eliminates the need to acquire high-end testers. The implementation of BIST technique in uart serial communication is simulated and synthesized using Xilinx and model-sim 12.3 versions.

KEYWORDS: UART, BIST, Error check, Status register, LFSR.

INTRODUCTION

Asynchronous serial communication has advantages of high reliability, less transmission lines, long transmission distance. Uart allows the bi-directional way which is a full duplex communication in serial link, thus has been largely used in control systems and data communications. 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. In processor the uart appears as an 8-bit read/write parallel port basic uart needs only two signal lines to complete full duplex data communication. TXD is the transmitter side which is, the output of uart and RXD is the receiver side, which is the input of UART.

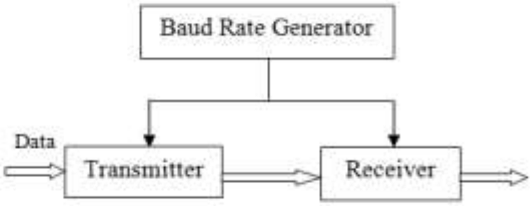


Figure 1: UART MODULE

UART includes three modules namely transmitter, baud rate generator and receiver shown in Fig.1.The UART receiver module is used to receive the serial signals at Receiver, and convert them into parallel data. 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 transmit module converts the bytes into serial bits according to the basic frame format.

IMPLEMENTATION OF UART

When the transmitter is idle, the data line is in high logic state. If not when a word is given to the uart for asynchronous transmissions, "start bit" is added to the beginning of each word that is to be transmitted .The start bit is used to known the peripherals receiver that a word of data is about to be sent, and synchronization is to force the clock in the receiver to synchronize with the clock in the transmitter. After the start bit, the data bits of the word are sent, with the least significant bit (LSB) being sent first. Each bit is transmitted at exactly same time as all of the other bits, and the receiver samples at the wire at 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, the transmitter adds a parity bit that the transmitter generates. The parity bit may be used by the receiver to perform simple error checking then one stop bit is sent by the transmitter is shown in Fig. 2.

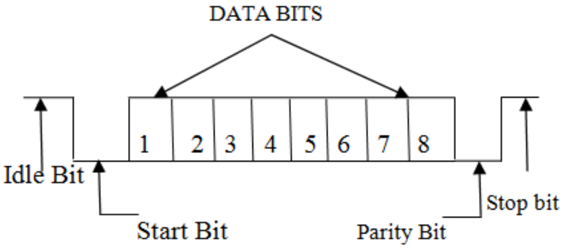


Figure 2: FRAME FORMAT

When the receiver has received all of the bits in the frame, it automatically discards the start, parity and stop bits. If another word is ready for transmission, the start bit for the next word will be sent when the stop bit for the before transmitted bits or data is received**.**

In actual applications, usually a few features in uart are required. Specific interface chip causes increase in cost and wastage of resources in electronic design, SOC technology is being used widely now-a days. This shows the requirement of realizing the whole system function in a single or a few chips.

Manufacturing process is extremely complex, for the manufacturers even to consider testability requirement to have the reliability and the functionality of each of their designed circuits. Testing of integrated circuits to ensure high level of quality in product functionality in products. In the new system-on-a-chip design, many cores are integrated into a single chip. As ICs grow in gate counts, it is no longer true that most gate nodes are directly accessible by one of the pins on the package. This makes testing of the nodes more difficult as they could be no longer be easily observed by signal from an input pin nor easily observed at an output pin.

In this paper internal diagnostic capabilities are built into Uart by the introduction of built-in-self-test and error simulation of data at receiver for any data corruption. The uart with status register and BIST module is coded in verilog HDL and simulated using Xilinx modelsim tool of 12.3 versions.
The paper is divided into 5 sections. They are: Uart, BIST technique and its implementation, Architecture of UART, Results, Conclusion of work.

BIST TECHNIQUE

In VLSI we have testing problems like input combinatorial problems, gate to I/O pin ratio problems, Test generation problems, led the designer to identify reliable test methods and solve this problems.teh insertion of special test circuitry on the VLSI circuits that allows efficient test methods. This has been addressed by the need for design for testability (DFT) and hence the need for Bist. It tests the circuit or system function itself hence it is named as “self-test".BIST is an on-chip test logic that is utilized to test the functional logic of a chip, by itself. Due to the rapid increase in the design complexity, BIST has become a major design consideration in DFT methods and is becoming increasingly important in today’s state of the art SoCs.

A properly designed BIST is able to offset the cost of added test hardware while at the same time ensuring the reliability, reduces maintenance cost and testability.

BIST solution consists of a Test Pattern Generator (TPG), the circuit to be tested, a way to analyze the results, and a way to compress those results for simplicity and handling as shown in Fig-3.Generic BIST architecture components are;

Circuit under Test (CUT): This 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).

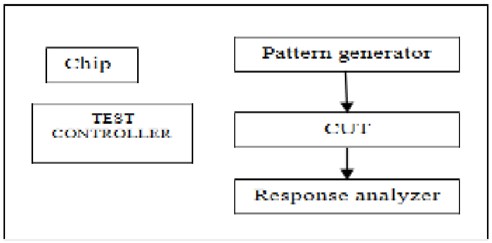


Figure 3: A Generic Bist Module

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.in TRA we have -One's count, Transition count, Parity checking,Syndrome checking, Signature analysis operations.

BIST Controller Unit (BCU): It controls the test and Execution; it manages the TPG, TRA and reconfigures the CUT and the multiplexer. During BIST mode, it selects i/p from the pattern generator to cut while during functional mode, selects primary i/p.

There are different approaches used to generate test patterns for BIST are LFSR, Binary counters, modified counters, Cellular automation and ROM. LFSR is used to generate pseudorandom test patterns. This normally requires a sequence of one million or more tests pattern in order to achieve high fault coverage. A binary counter can generate an exhaustive but not randomized test sequences. Drawback of binary counters as the pattern generator is, it requires more hardware than typical Linear Feedback Shift Register (LFSR) pattern generator. Modified counters also have been successfully as test-pattern generators. However, they also require long test sequences. ROM method stores a good test pattern set on the chip but relatively it is expensive in chip area.so including all these LFSR has advantages of compact and simple design and is preferred to use in bist as test pattern generator.

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 proposed model has two major modules like UART and BIST. Further in the UART, we have transmitter, Receiver, and baud rate generator as discussed before. 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 Fig. 4.

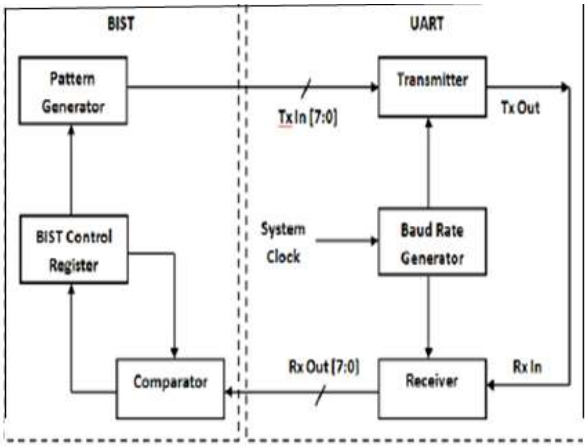


Figure 4: UART WITH BIST ARCHITECTURE

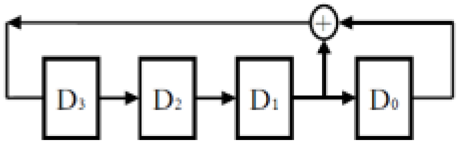
Bist Pattern Generator:

In bist session where the pattern generator is one of the module in it which it produces pseudo–random test patterns by using LFSR as the test pattern generator. A LFSR is a shift generator where its input is a linear function of two or more bits which are called as taps. It consists of the D-flip-flop and the ex-or gates in its function. It performs its operation by shifting the bits and ex-or operation,

In LFSR we have two types they are internal LFSR and external LFSR. Here we are using external LFSR the bits contained in selected positions in the result is fed back into the register’s input bit. The bit positions selected for use in the feedback function are called “taps”. The largest state space possible for such an LFSR will be 2^n-1, all possible values except zero state are shown as an example.

A sequence of binary numbers can be represented using a generation function(polynomials).The behavior of an LFSR is determined by its initial “seed” and its feedback coefficients, both can be represented by polynomials. All zero values is not allowed in LFSR, as it always produce 0 inspite of how many clock iteration. Because each state can have only one succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before again repeating another state.

In LFSR the test random test pattern generators are generated by the ex-or and D-flip flop operation and the generated patterns are sent as input to the transmitter section in the UART module.
LFSR example:



Here 2^4-1=15 near test patterns are generated with the D-flip flop and ex-or gates.

Uart Transmitter:

The transmitter accepts parallel data from the test pattern generator which generates pseudo random test patterns as input, and then it makes the frame of the data and transmits the data in serial form on the Transmitter Output (TXOUT) terminal shown in Fig.5. The baud rate generator output will be the clock for uart transmitter.

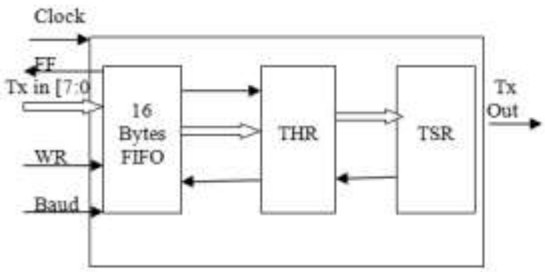


Figure 5: UART TRANSMITTER

Data is loaded from the parallel inputs TXIN0-TXIN7 into the Transmitter FIFO by applying logic high(1) 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 Fig. 5. When FIFO contains some data, it will send the signal to Transmitter Hold Register (THR), which is an 8-bit register. At the same time, if THR is empty it will send the signal to FIFO, back 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-bit register in which framing process occurs. In frame, start bit, parity bit and one stop bit will be added . Now data is transmitted from TSR to TXOUT serially.

From the transmitter the data is sent to the uart receiver as the loopback function where the baud rate acts as a clock for the synchronization between the transmitter and the receiver to receive data correctly.

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 Fig.6 initially the logic line (RxIn) is high.

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 if all the bits are sent to Receiver Shift Register (RSR) one by one where the entire frame is stored.RSR is a 12 bit shift register with 8 bit data and extra 4bits error logic.

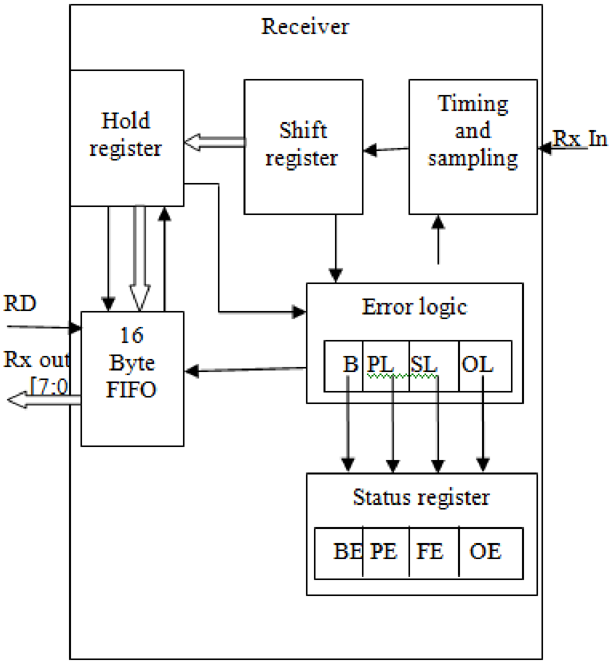


Figure 6: UART RECEIVER

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 with discarding the extra 4 bits which are in error logic. The remaining bits in the RSR are used by the error logic block. Then, if receiver FIFO is empty without any data contained in it, it then 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 (0-7) bits are shown. 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), Overrun 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.

Comparator:
After receiving the data to the receiver state and performing the functions there the data is given to the comparator where it compares the data which is transmitted and the data received is same or not then given to the bist control register.

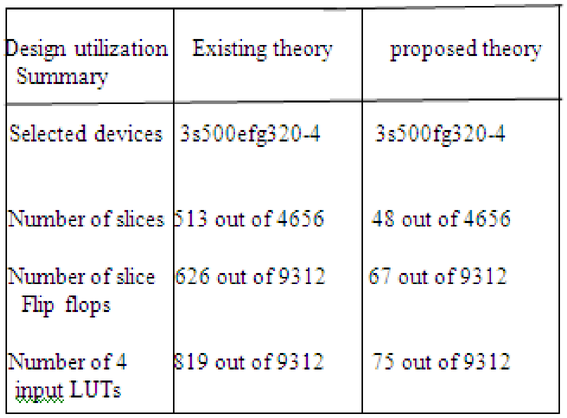
Bist Control Register Operation:

After the comparison if the FIFOS are with the same data then BIST is passed and bit-0 of BIST control register is selected else 0 is selected. The 8-bit BIST control register the bit B7 is set when the BIST starts, bit B6 when the LFSR patterns are generated and the Tx FIFO is loaded,B5 is set when Tx test data is generated ,B4 when the Rx section data is loaded,B3 when the comparison started,B2 is set when there is any error logic obtained in the status register B0 is set as 1 when BIST is passed or else 0 when bit fails.

Simulation Results

The implementation of the function in vhdl coding and verilog coding and simulation of the design are done in Xilinx model sim 12.3.the baud rate set is 9600bps.the word length of the data used here is 8 bit. The comparison is shown here for the already existing theory which is present and the proposed theory.

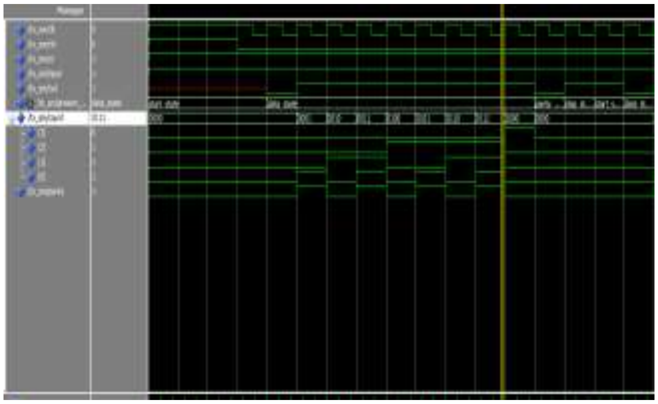
Comparison of the existing theory and proposed theory:



Design utilization Summary	Existing theory	proposed theory
Selected devices	3s500efg320-4	3s500fg320-4
Number of slices	513 out of 4656	48 out of 4656
Number of slice Flip flops	626 out of 9312	67 out of 9312
Number of 4 input LUTs	819 out of 9312	75 out of 9312

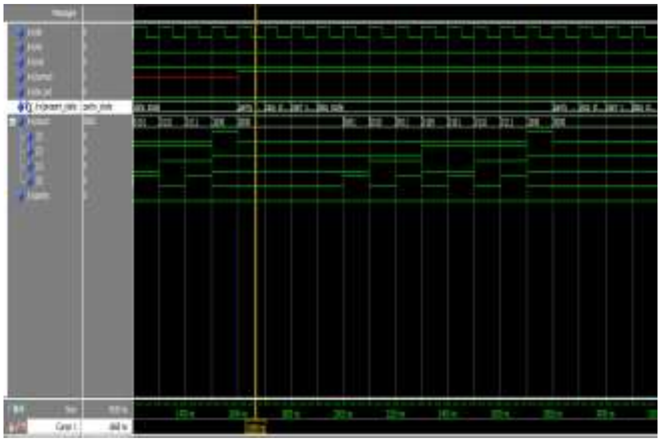
Simulation of UART transmitter in vhdl:

The result is shown when the data is transmitted and is observed how the start bit, parity bit and stop bits are transferred.

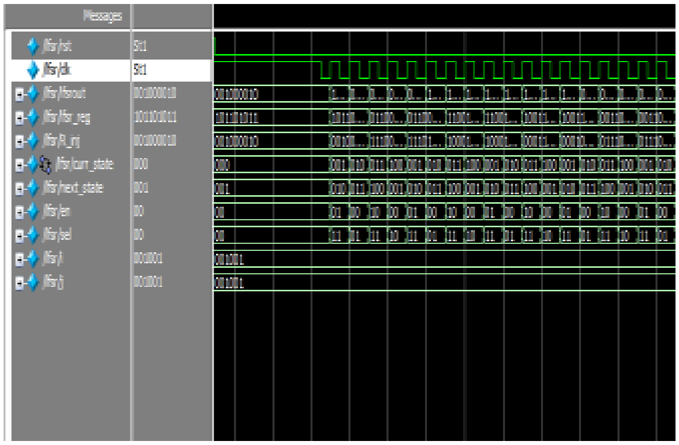


Simulation result of receiver in vhdl:

The result is shown how the data is received at the receiver section with the same start,parity and stop bits.



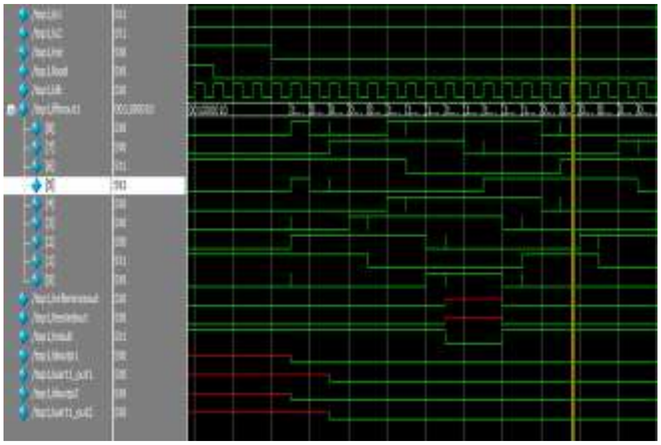
Simulation results of LFSR in verilog:



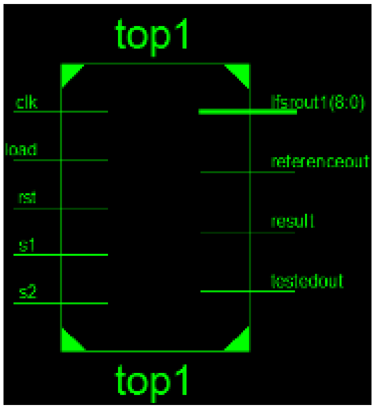
In LFSR the shifting of the bits with D-flip flop and ex-or operation is shown.

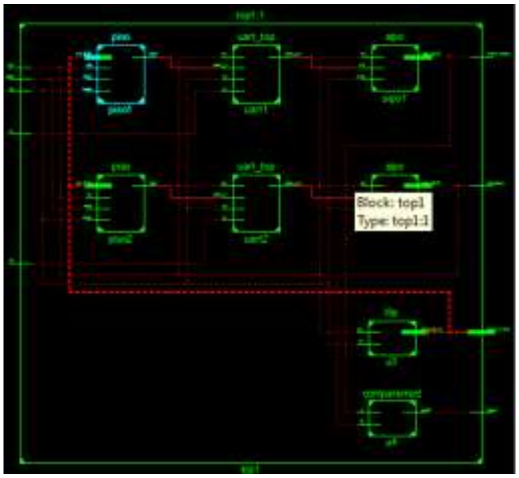
Simulation result of top module:

The result shows the data transmitted is received without any error corrections is checked in the form of reference output and tested output.

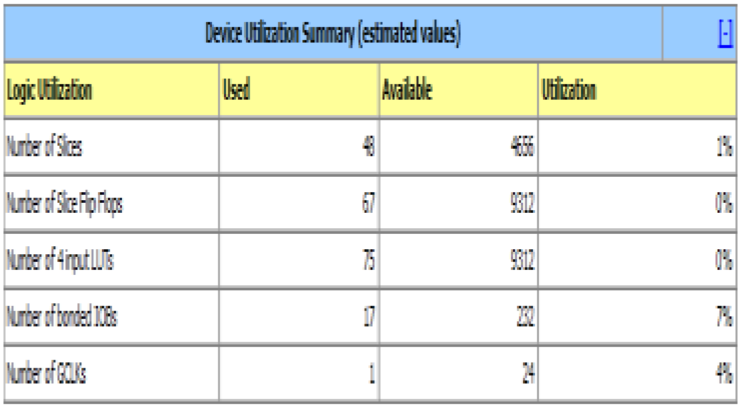


Schematic diagram of top module:



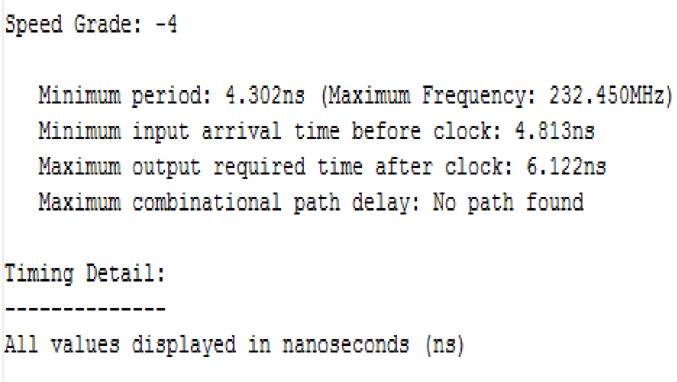


Design summary:



Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slices	48	4656	1%
Number of Slice Flip Flops	67	9312	0%
Number of 4 input LUTs	75	9312	0%
Number of bonded IOBs	17	232	7%
Number of GCLKs	1	24	4%

Timing summary:



```
Speed Grade: -4

   Minimum period: 4.302ns (Maximum Frequency: 232.450MHz)
   Minimum input arrival time before clock: 4.813ns
   Maximum output required time after clock: 6.122ns
   Maximum combinational path delay: No path found

Timing Detail:
--------------
All values displayed in nanoseconds (ns)
```

CONCLUSION OF WORK

The architecture of UART with BIST technique that supports the 8-bit data word length for the serial communication of data with the status register to detect the errors produced and gives the correct transmission of data at input and receiving at the output.

By using the BIST technique in UART serial communication we can reduce the requirements of the tester functions and the steps in it. The LFSR which produces the pseudo random test patterns as input to the transmitter to give better fault coverage to the UART module. Due to the extension of the circuit by adding BIST concept additionally we had increase in design time and hardware utilization but by the reduction of the cost and good market opportunities we can cover the problem.

REFERNCES

[1] Chakrabartyl,“Deterministic Built-in Test Pattern Generation for High-Performance Circuits Using Twisted-Ring Counters,” IEEE Trans. of VLSI Systems, Vol. 8, No. 5, pp. 633636, Oct. 2000
[2] 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.
[3] 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.
[4] S. R. Das, “Built-in self-testing of VLSI circuits,” IEEE Potentials, vol.10, pp. 23–26, Oct. 1991.
[5] 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.