Dr. T.V.S.P.Gupta, Y. Kumari, M. Ashok Kumar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 1, January -February 2013, pp.636-640

UART realization with BIST architecture using VHDL

Dr. T.V.S.P.Gupta[1], Y. Kumari[2], M. Ashok Kumar[3]
1 – Department of ECE, K L University, Guntur, Andhra Pradesh, India
2 – Department of ECE, PSCMR College of Engineering and Technology, Vijayawada-1
3- M.S student of VLSI and ESD, Department of ECE, JNTU Hyderabad

ABSTRACT

The increasing growth of sub-micron technology has resulted in the difficulty of testing. Design and test engineers have left no choice but to accept new responsibilities that had been performed by group of technicians in the previous years. Design engineers who do not design systems with full testability had increased the possibility of product failures and missed market opportunities. BIST is a design technique that allows a circuit to test itself. In this approach the test performance achieved with the implementation of BIST is proven to be adequate to offset the disincentive of the hardware overhead produced by the additional BIST circuit. The technique can provide shorter test time compared to an externally applied test and allows the use of low-cost test equipment during all stages of production.

Keywords – ATPG, BILBO, BIST, LFSR, UART.

1. INTRODUCTION

Manufacturing processes are extremely complex, inducing manufacturers 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 used is Built-in-Self-Test (BIST). A BIST universal asynchronous receive transmit (UART) has two main objectives firstly to satisfy the specified testability requirements and secondly to generate the lowest cost with the high performance implementation. UART has been one of the most important input/output tools for decades and is still widely used.

Serial data is transmitted via its serial port. A serial port is one of the most universal parts of a computer. It is a connector where serial line is attached and connected to peripheral devices such as mouse, modem, printer and even to another computer. In contrast to parallel communication, these peripheral devices communicate using a serial bit stream. This approach mainly focuses on the design of UART with embedded BIST capability and on the problems of Very Large Scale Integrated (VLSI) testing followed by the behavior of UART circuit using Very High Speed Integrated Hardware Description Language (VHDL).

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. In the implementation phase, the BIST technique will be incorporated into the UART design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements.

1.1 Need for using BIST technique

To identify reliable testing methods which will reduce the cost of test equipment, a research to verify each VLSI testing problems has been conducted. The major problems detected so far are as follows:

- Test generation problems
- The input combinatorial problems and
- Gate to I/O pin ratio

TEST GENERATION PROBLEMS

The large number of gates in VLSI circuits has pushed computer automatic-test-generation times to weeks or months of computation. The numbers of test patterns are becoming too large to be handled by an external tester and this has resulted in high computation costs and has outstripped reasonable available time for production testing. Another test generation problem is that computer algorithms providing Automatic Test Pattern Generation (ATPG) work well for combinational logic but rather poorly for sequential logic circuits. Sequential circuits demand too much computer memory and computation since many more time states must be evaluated.

THE INPUT COMBINATORIAL PROBLEM

A combinatorial logic circuit with N primary input nodes has a total set of 2^N possible input vectors. This is the number of test vectors required to exhaustively test a circuit for those functions that a customer might use. In contrast to MSI (Medium-Scale-Integrated) circuits, the number of test vectors needed to exhaustively examine a VLSI circuit such as 32-bits microprocessor is prohibitive. A finite number of test vectors can still be applied to an IC and follow the economic rules of production. The finite number of test vectors is much lesser than the full exhaustive test set of a VLSI circuit.

The Gate to I/O Pin Ratio Problem

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 internal nodes more difficult as they could neither no longer be easily controlled by signal from an input pin (controllability) nor easily observed at an output pin (observability). Pin counts go at a much slower rate than gate counts, which worsens the controllability and observe ability of internal gate nodes.

The VLSI testing problems described above have motivated designers to identify reliable test methods in solving these difficulties. An insertion of special test circuitry on the VLSI circuit that allows efficient test coverage is the answer to the matter. The need for the insertion has been addressed by the need for design for testability and hence BIST technology is used.

2. UART DESCRIPTION

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. The transmitter sends a byte of data bit by bit serially out from UART while UART receiver receives the serial in data bit by bit and converts them into a byte of data. UART starts the data transmission by asserting a bit called the "Start Bit" to the beginning of each data that is to be transmitted. The Start Bit is also used to inform the receiver that a byte of data is about to be sent. After the Start Bit, the individual bits of the "byte" of data are sent, with the Least Significant Bit (LSB) being sent first. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits. On the other, UART the receiver will need to sample the logic value that being received at approximately halfway through the period assigned to each bit to determine if it is logic 1 or logic 0.

When a byte of data has been sent, the transmitter may add a "Parity Bit". The receiver to perform simple error checking may use the Parity Bit. In this project, parity bit is not being implemented. After this, a "Stop Bit" is sent by the transmitter to indicate the transmitter has completed the data transmission. If another byte of data is to be transmitted, the Start Bit for the new data can be sent as soon as the Stop Bit for the previous word has been sent.

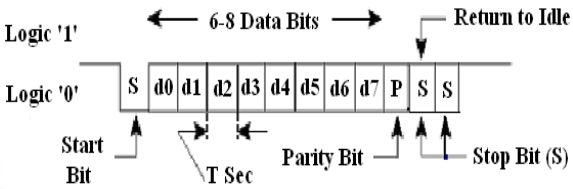

Fig. 1 UART data frame format

2.1 UART character transmission

Below is a timing diagram for the transmission of a single byte, Uses a single wire for transmission and each block represents a bit that can be a mark (logic '1') or space (logic '0')

The following figures are the UART frame formats for transmitting single start bit, 7 data bits, one parity bit and one stop bit.

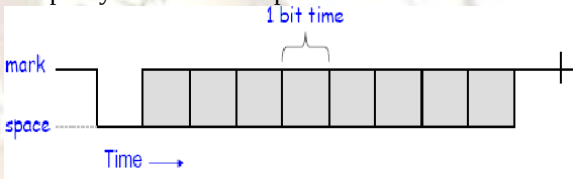

Fig. 2 each bit with fixed time duration

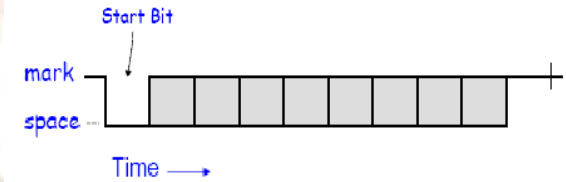

Fig. 3 start bit marks the beginning of a new word

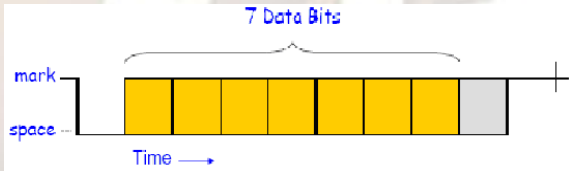

Fig. 4 least significant bit is sent first

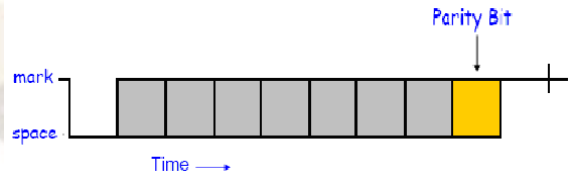

Fig. 5 parity bit is added

2.2 Uses of UART

- A UART may be used when

– High speed is not required

– An inexpensive communication link between two devices is required

- UART communication is very cheap

– Single wire for each direction (plus ground wire)

- Asynchronous because no clock signal is transmitted

– Relatively simple hardware

- PC devices such as mice and modems are used for

Communication of UART with PC

- PC serial port is a UART

– Serializes data to be sent over serial cable

– De-serializes received data.

3. BIST TECHNOLOGY

A digital system is tested and diagnosed during its lifetime on numerous occasions. It is very critical to have quick and very high fault coverage testing. One common and widely used in semiconductor industry for IC chip testing is to ensure this is to specify test as one of the system functions and thus becomes self-test. A system designed without an integrated test strategy which covering all levels from the entire system to components is being described as chip-wise and system-foolish. A proper designed Built-In-Self-Test (BIST) is able to offset the cost of added test hardware while at the same time ensuring the reliability, testability and reduces maintenance cost.

BIST system hierarchy for the 3 level of packaging which is the system level, board level and chip level. The system consists of several PCB's (or boards). Each of the PCB has multiple chips. The system Test Controller can activate self-test simultaneously on all PCB's. Each Test Controller on each PCB board can activate self-test on all the chips on the board. The chip Test Controller runs the self-test on the chip and transmits the result out to the board Test Controller. The board Test Controller accumulates test results from all chips on the PCB and sends the results to the system Test Controller. The system Test Controller uses all of these results to determine if the chips and board are faulty.

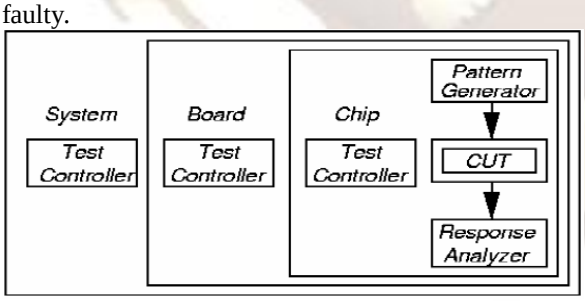

Fig. 6 BIST hierarchy

3.1 BIST Implementation

Basically, a design with embedded BIST architecture consists of a test controller, hardware pattern generator, input multiplexer, circuit under test (CUT). Optionally, a design with BIST capability may includes also the comparator and Read-Only-Memory (ROM). As shown in Figure 3.2, the test controller is used to control the test pattern and test generation during BIST mode. Hardware pattern generator functions to generate the input pattern to the CUT. Normally, the pattern generator generates exhaustive input test patterns to the CUT to ensure the high fault coverage. For example, a CUT with 10 inputs will require 1024 test patterns. Primary Inputs are the input for CUT during the non-BIST mode or in other word, functional mode. Input multiplexer is used to select correct inputs for the CUT for different mode. During BIST mode, it selects input from the hardware pattern generator while during functional mode, selects primary inputs. Output response compactor acts as compactor to reduce the number of circuit responses to manageable size that can be used as the signature and stored on the ROM.

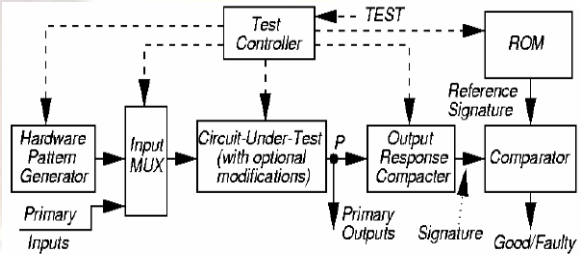

Fig. 7 BIST Architecture

3.2 BIST Pattern Generation

There are various methods and approaches have been used to generate test patterns during BIST. This can be described in brief below:

Linear Feedback Shift Register is used to generate pseudo-random test patterns. This normally requires a sequence of one million or more tests pattern in order to achieve high fault coverage. One of the advantages of LFSR is it uses very little hardware and thus is currently the preferred BIST pattern generation method. In this project, LFSR is being chosen as the test pattern generation method.

A binary counter can generate an exhaustive but not randomized test sequences. Draw back of binary counters as the pattern generator is, it requires more hardware than typical LFSR pattern generator.

Modified counters also have been successfully as test-pattern generators. However, they also require long test sequences.

This method stores a good test pattern set from an ATPG program in a ROM on the chip. However, drawback of this approach is relatively expensive in chip area.

3.3 Standard LFSR

The standard LFSR method has been used in this project as the test pattern generator for the BIST. In this section, the implementation of LFSR will be discussed in detail. A LFSR is a shift register where the input is a linear function of two or more bits (taps). It consists of D flip-flops and linear exclusive-OR (XOR) gates. It is considered an external exclusive-OR LFSR as the feedback

network of the XOR gates feeds externally from X_0 to X_{n-1}.

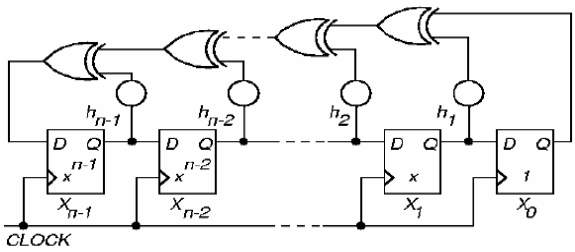

Fig. 8 Standard n-stage LFSR circuit

One of the two main parts of an LFSR is the shift register. A shift register is used to shift its contents into adjacent positions within the register or, in the case of the position on the end, output of the register. The position on the other end is left empty unless some new content is shifted into the register. The contents of a shift register are usually thought of as being binary, that is, ones and zeroes. If a shift register contains the bit pattern 1101, a shift (to the right in this case) would result in the contents being 0110; another shift yields 0011. In an LFSR, the bits contained in selected positions in the shift register are combined in some sort of function and the result is fed back into the register's input bit. By definition, 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, filling the position that is emptied as a result of the shift.

The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is n bits long is the nth bit; the input bit of an LFSR is bit 1. The state of an LFSR that is n bits long can be any one of 2 different values. The largest state space possible for such an LFSR will be 2^n-1, all possible values except the zero state. All zero is not allow in LFSR, as it will always produce 0 in spite of how many clock iteration. Because each state can have only once succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

4. SYNTHESIS RESULTS

4.1 Device utilization summary:

Selected Device: 2s50tq144-6

Number of Slices:	38 out of	768	4%
Number of Slice Flip Flops:	49 out of	1536	3%
Number of 4 input LUTs:	54 out of	1536	3%
Number of bonded IOBs:	19 out of	96	19%
Number of GCLKs:	1 out of	4	25%

4.2 Timing Summary:

Speed Grade: -6
Minimum period: 8.712ns (Maximum Frequency: 114.784MHz)
Minimum input arrival time before clock: 2.599ns
Maximum output required time after clock: 9.668ns
Maximum combinational path delay: No path found
Total memory usage is 96912 kilobytes
Number of errors : 0 (0 filtered)
Number of warnings: 61 (0 filtered)
Number of info's : 0 (0 filtered)

5. SIMULATION RESULTS

5.1 UART in Normal Mode
When Inputs: Clock – 1 Reset- 0 CS – 0
If txd is '1' then rxd is '1'
If txd is '0' then rxd is '0'

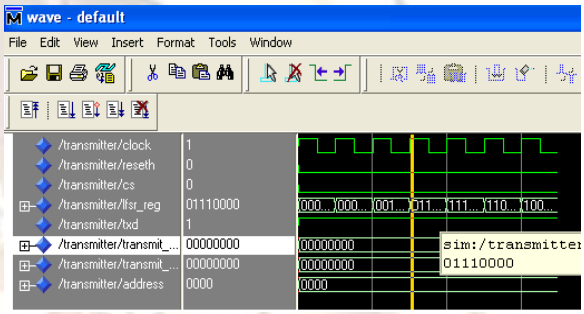

Fig. 9 Simulation result for UART in normal mode.

5.2 UART in Testing Mode
When Inputs: Clock – 1 Reset – 0 CS – 0
When BILBO Mode is "00" Shift register mode

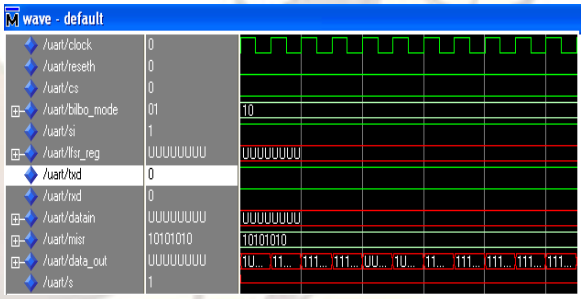

Fig. 10 Simulation result for UART in testing mode.

When Inputs: Clock – 1 Reset- 0 CS – 0
When BILBO Mode is "11" Multiple Input Signature register (MISR)

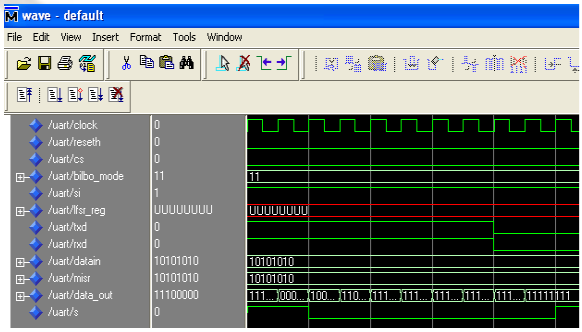

Fig. 11 Simulation result for BILBO mode.

6. CONCLUSION

The simulated waveforms presented in this paper have proven the reliability of the VHDL implementation to describe the characteristics and the architecture of the designed UART with embedded BIST. The simulated waveforms also have shown the observer how long the test result can be achieved by using the BIST technique. With the implementation of BIST, expensive tester requirements and testing procedures starting from circuit or logic level to field level testing are minimized.

The LFSR replaces the function of the external tester features such as a test pattern generator by automatically generating pseudo random patterns to give 100% fault coverage to the UART module. The MISR acts as a compression tool, compressing the output result when automatic pseudo random pattern is fed to the UART. The shift register minimized the input/output overhead by shifting the parallel signature produced by MISR into serial signature. The reduction of the test cost will lead to the reduction of overall production cost.

ACKNOWLEDGEMENTS

Authors like to express their gratitude to words the management of PSCMR College of Engineering and Technology, KL University, and JNTU., Hyderabad for their continuous support and encouragement during this work. Further authors like to express their sincere thanks to Dr. Madhavi Latha, HOD, ECE, JNTUCE, Hyderabad, Dr. Habibulla Khan, HOD, ECE, and Prof N. Venkatram, A/Dean Electrical Sciences, KL University for providing excellent R&D facilities at ECE dept of KL University to carry out this work.

REFERENCES

[1] A.P. Stroele, and H. J. Wunderlich, "Hardware - Optimal Test Register Insertion", in IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, June 1998, Vol. 17, No. 6, pp. 531-539.

[2] J. Turino, "RTL DFT Rule Checking - The Circuit Designer's Secret Weapon", Integrated System Design Magazine, 2000.

[3] M. Ibrahim Abu-Bakr, "A Built in Self Testable Bit-Slice Processor", Faculty of Computer Science & Information Technology, University of Malaya, May 1995.

[4] M. S. Michael, "A Comparison of the INS8250, NS16450 and NS16550AF Series of UARTs", National Semiconductor Application Note 493, April 1989.

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

[6] O. A. Petlin, and S. B. Furber, "Built-In-Self-Testing of Micropipelines" in Advanced Research in Asynchronous Circuits and Systems, IEEE, 1997, pp 22-29.

[7] "PC16550D Universal Asynchronous Receiver / Transmitter with FIFOs", National Semiconductor Application Note, June 1995.

[8] S. Wang**,** "Generation of low power dissipation and high fault coverage patterns for scan-based BIST", in Proceedings of International Test Conference, 2002, pp. 834 -843.

[9] Z. Navabi, "VHDL Analysis and Modeling of Digital Systems", McGraw-Hill Inc., 1991.

Authors Profile:

Dr. T.V.S.P. Gupta received his B.Tech. in E.C.E from S.V. University, Tirupathi, ,M.Tech (D.S.C.E) from J.N.T.University, Hyderabad and Ph.D. from Magadh University. Currently I am working as Professor in the department of ECE, K L University, Vaddeswaram, Guntur (DT), A.P., India. His field of interest is Adhoc Wireless Networks and Low Power VLSI.

Mrs**. Y. Kumari,** received her B.Tech. in E.C.E from Bapatla Enginnering College, M.Tech (D.S.C.E) from J.N.T. University, Hyderabad and pursuing Ph.D. in Andhra University. Currently she is working as Professor in the department of ECE, PSCMR College of Engineering and Technology, Vijayawada-1. A.P., India. Her field of interest is Image processing and VLSI.

M.Ashok kumar, received his B.Tech in ECE from Prasad Institute of Technology and Science, Jaggayapet, Presently pursuing M.S. in JNTU Hyderabad in the field of VLSI & ESD.