Design and Simulation of Low Power UART with BIST and its application to MIPS Soft-Core Processor

[1]Akshata Surendra Savant, [2]Sandra Benjamin, [3]M Z Kurian,
[1 2 3]Sri Siddhartha Institute of Technology Tumkur, Karnataka.

ABSTRACT- Today, testing of VLSI chips is becoming very much popular in the fields of VLSI Technology. Therefore, there is a lot of work going on in the area of testing to bring in a fault free VLSI chips, as the complexity in integration is increasing drastically. Built-in-self-test is the important on-chip testing method, which has brought tremendous growth in VLSI fields. BIST helps to test a system automatically, by generating a random test patterns, so that it can provide less time compared to external testing and helps to achieve much more productivity at the end. Universal Asynchronous Receiver-Transmitter is widely used in communication field. UART is mostly a kind of serial communication. Therefore there is a necessity of realizing UART in a few chips is needed and it is also important to ensure that the data transfer is error proof and it consumes low power. The low power UART with BIST architecture is coded and simulated using Verilog-HDL using Xilinx product Version ISE 14.7. UART is having a greater application in MIPS soft-core processor. Suppose if there is any change in the assembly code of the implemented processor then it requires re-implement and download the soft-core on FPGA. Therefore, FPGA implementation of a run-time loading technique for a 32-bit MIPS (Microprocessor without Interlocked Pipeline Stages) processor has become significant. The design consists of soft-core processor, a soft-ware tool and an UART. The software tool communicates with the soft-core through UART block. The application of MIPS processor is demonstrated by using UP/DOWN counter assembly code and it is simulated using Xilinx ISE 14.7 and is implemented on Zynq-XC72020CLG484 FPGA kit.

Keywords: UART, BIST, VLSI, LFSR, MIPS, FPGA.

I.INTRODUCTION

Now a day's VLSI chips are becoming extremely complex, and costly. BIST technique has played an important role in testing field and it helps to test a system automatically. Testing of integrated circuits (ICs) is important to bring in a high level of quality in product functionality in both commercially and privately produced products.

Built-in-self-test is an on-chip test logic that is utilized to test functional logic of a chip. BIST is more useful in VLSI testing. BIST is present when the resources required for the test execution are internal to the circuit. BIST it generally consists of a test pattern generator, a circuit which is to be tested, test response analyzer (TRA), BIST controller unit (BCU) and a way to compress the obtained results for simplicity and handling. The circuit under test is a part of the circuit that is to be tested in BIST mode, which can be memory, sequential or combinational. The test pattern generator it generates patterns for the circuit under test (CUT), it can be a microprocessor or a dedicated circuit, the patterns can be generated in a pseudorandom way or deterministically. TRA it analyses the correct value of sequence on primary output and then compares it with the expected one. BIST controller unit it controls the test execution and it handles the TRA, TPG, and reconfigures the CUT and the multiplexer. Achieving high fault coverage while keeping the test time within the limits and maintaining an acceptable design overhead is of more importance. Therefore with the rapid increase in the design complexity, BIST has become important design considerations in Design-For-Testability (DFT) approaches and its demand has increased in today's state of art SOC.

Asynchronous serial communication has advantages of less transmission line, high reliability; therefore it is widely used in data exchange. Asynchronous serial communication is generally implemented by Universal Asynchronous Receiver Transmitter. UART is one of the serial communication protocols which is widely used for short-distance communication. UART includes three modules namely, receiver, transmitter and the baud rate generator. The baud rate generator is normally used to produce a local clock signal, where it is much higher than the baud rate to control the UART transmitter and receiver. The serial signals at RXD are received by the UART receiver module, and then convert them into parallel data. According to the basic frame format the UART transmit module converts the bytes into serial bits and those bits transmitted through TXD.

When the transmitter is in idle state, the data line is logic high state. When a word is given to the UART for asynchronous transmissions, 'Start Bit' (logic low) is added to the starting of each word which is to be transmitted. The Start Bit alerts the peripheral receiver that a word of data is to be sent, and it forces the clock in the transmitter to be in synchronization with the clock in the receiver. After the Start Bit is sent, then the each individual data bits of the word are sent, with the Least Significant Bit (LSB) being sent first. Each bit will be transmitted for exactly the same amount of time as with respect to other bits. After the entire data word has been sent, and then the transmitter will add a Parity Bit that the transmitter generates. For simple error checking the Parity Bit will be used by the receiver. Then at last one Stop Bit will be sent by the transmitter.

When all the bits in the frame are received by the receiver, then it will automatically start discarding the Start, Stop and parity bits. If the next word is ready for transmission, then the Start bit for the new word will be sent soon after the Stop bit for the previous word has been sent. Asynchronous data are 'self-synchronizing' if there are no data to transmit, transmission line is held idle.

In actual applications, normally only a few key features of UART are needed. Specific interface chip will cause increased cost and waste of resources. Specifically in the field of electronic design, SOC technology is growing at a faster rate. Hence this situation results in the requirement of realizing the whole system function in a single or a very few chips. The core functions into a FPGA chip are integrated to achieve reliable, stable and compact data transmission and it avoids waste of resources and decrease cost.

UART satisfies the specified testability requirements, and it usually generates the lowest cost with highest performance implementation. UART has arisen as an important input/output tool for decades and till today it is widely used. UART is having greater significance in communication world and it can be used in applications like broadband modem, base station, cell phone, GPS units, Bluetooth modules etc.

The power dissipation is a main concern in VLSI chips. Therefore, power dissipation may bring in adverse affect on performance, reliability, packaging, cost and portability.

The main reasons for power increase during testing are:

- High-switching activity due to nature of test patterns,
- Low correlation among test vectors.
- Parallel activation of internal cores during test.
- Power consumed by extra design-for-test (DFT) circuitry etc.

The three different types of power dissipation are:

- Static power dissipation.
- Short-circuit power dissipation,
- Dynamic power dissipation.

The dynamic power dissipation is more than static and short-circuits power dissipation.

The principle objective of BIST enabled UART is to automatically test the UART blocks and generate test pattern, so it can provide less time compared to an externally applied test pattern. The design will consume reduced timing constraints and less power. The speed of testing will be increased due to BIST compared to externally generated patterns, because the delay caused by external testing is more than the on-chip testing.

A soft-core processor is a microprocessor which is defined in hardware description language (HDL) that can be synthesized in programmable hardware, such as FPGAs. Soft-core processors on Field Programmable Gate Array (FPGA) chips are becoming an increasingly popular solution to support application-specific customization. However, any change in the assembly code of the implemented processor requires re-implement and download the soft-core on FPGA. So therefore the FPGA realization of a run-time loading technique for MIPS processor has become significant.

II. SYSTEM DESIGN

General BIST Architecture:

Built-in-self-test is a design technique in which parts of a circuit are used to test the circuit itself. Main purpose of BIST is to reduce the complexity, and thereby decrease the cost and to reduce the use of external test equipment. BIST consists of elements such as a LFSR (Test Pattern Generator), a circuit to be tested, test response analyzer and a BIST controller unit. BIST architecture it generally consists of a following components:

- **Circuit under Test:** It is a part of the circuit that is to be tested in BIST mode, which can be sequential or combinational.
- **Test Pattern Generator:** The Test Pattern generator is a circuit which is to be tested. It can be a microprocessor or a dedicated circuit. The patterns can be generated in a pseudo random way or deterministically.
- **Test Response Analysis:** It analyses the correct value of sequence on primary output and then compares it with the expected one.
- **BIST Controller Unit (BCU):** BIST controller unit it controls the test execution and it handles the TPG, TRA and reconfigures the CUT and the multiplexer.

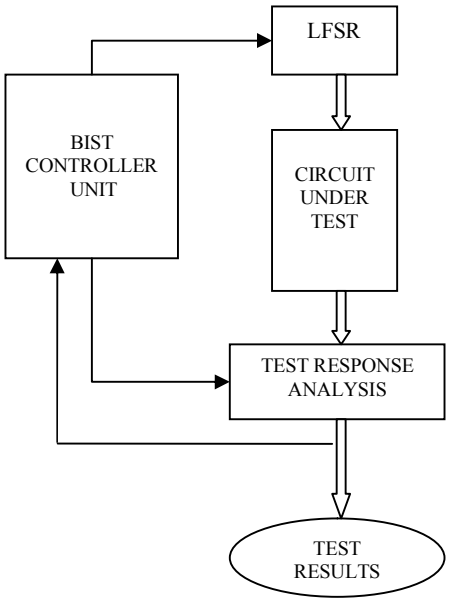


Figure 2.1: BIST Architecture

UART Block Diagram:

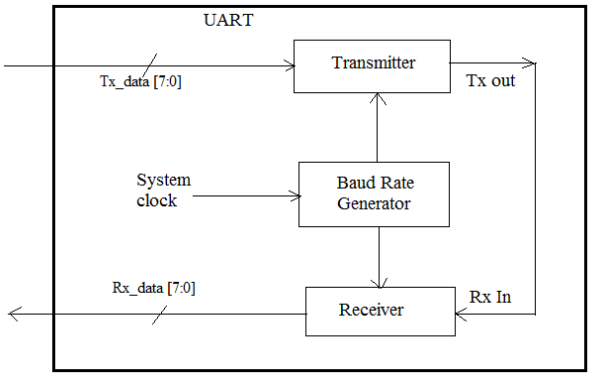
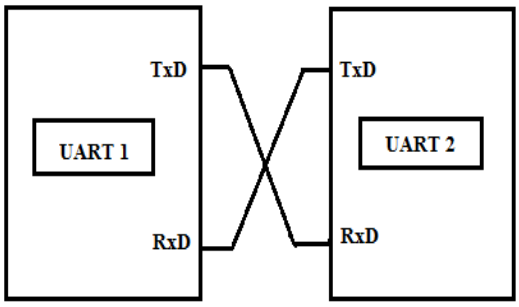


Figure 2.2: UART Block Diagram

Functionally UART consists of following sections

- UART Transmitter
- UART Receiver
- Baud Rate Generator

UART transmitter module consists of an output register, a transmitter register and transmitter control logic where as UART receiver module consists of input register, register buffer register and receiver control logic. Baud rate Generator used to generate the baud rates for both the transmitter and receiver.



TxD: Transmitter Line
RxD: Receiver Line

Figure 2.3: On board communication Interface

Figure 2.3 shows two UART with transmitter and receiver communicating with each other

UART Data Frame Format:

UART transmits and receives the entire data in the frame format as shown in figure 2.4. The data is framed between the start bit and a stop bit. The data frame format includes idle state, start bit, data bit, parity bit, and stop bit. When the transmitter is in idle state, the data line is logic high state. Whenever the transmitter is ready to send a data, firstly start bit is sent which is logic 0 (low). The start bit alerts the receiver that a word of data is ready to be sent, then it forces the clock in the transmitter to be in synchronization with the clock in the receiver. Start bit is followed by data bits, which varies from 5 to 8 bits, with the Least Significant Bit (LSB) being sent first. Each bit will be transmitted for exactly the same amount of time as with respect to other bits. In few cases, the parity bit of character byte is included in the data frame in order to maintain data integrity. It implies that for each character there will be a single parity bit in addition to start and stop bits. The parity bit can be even or odd. In case of an odd parity bit, the number of data bits, including the parity bit, has odd number of 1's. Similarly, in an even parity bit system the total number of bits including the parity bit is even. Then at last one stop bit will be sent by the transmitter.

When all the bits in the frame are received by the receiver, then it will automatically start discarding the start, stop and the parity bits. If the next word is ready for transmission, then the start bit for the new word will be sent soon after the stop bit for the previous word has been sent. The transmission line is held idle if there are no data to transmit.

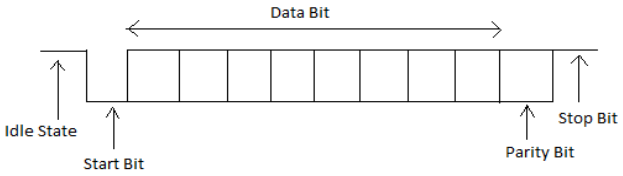


Figure 2.4 UART data Frame Format

Application of UART in Run-Time Loading for MIPS Soft-Core Processor: Soft-core processors on Field Programmable Gate Array (FPGA) chips are becoming an increasingly popular solution to support application-specific customization. Therefore, if there is any change in the assembly codes of the implemented processor requires re-implement and download the soft-core on FPGA. So therefore the FPGA realization of a run-time loading technique for MIPS processor has become significant. The design consists of three main blocks a microprocessor soft-core, a software tool and a UART. The software tools set the content of the instruction memory space of the processor without having to go through the implementation process. The software tool will communicate with soft-core via UART. A UP/DOWN counter assembly code can be used to demonstrate this technique.

The system is divided into three main blocks: a microprocessor (MIPS 32-bit), a UART interface and a software tool. Figure 2.5 illustrates the whole system path from top level view.

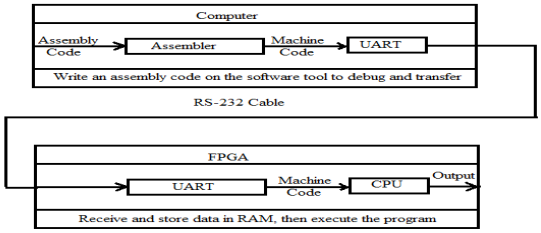


Figure 2.5: Whole system path of Run Time Loading for MIPS Soft Core Processor

III. SYSTEM IMPLEMENTATION

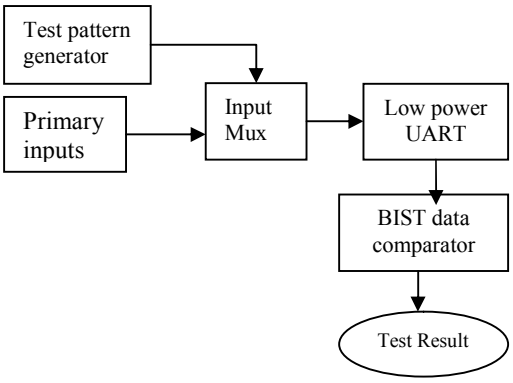


Figure 3.1: UART with BIST Implementation Flow

The mux either selects the test pattern generator inputs or the primary inputs based on the BIST enable signal. The output from the mux is given as an input to the UART. The obtained output of the UART is compared with the actual value of input, if actual value and obtained value matches than it is said to be error free, otherwise there is some error.

IV. SIMULATION RESULTS

The simulation results for low power BIST enabled UART and simulation of UART to MIPS soft-core processor in figure 4.1 and 4.2 respectively.

A. Low power BIST enabled UART Simulation result:

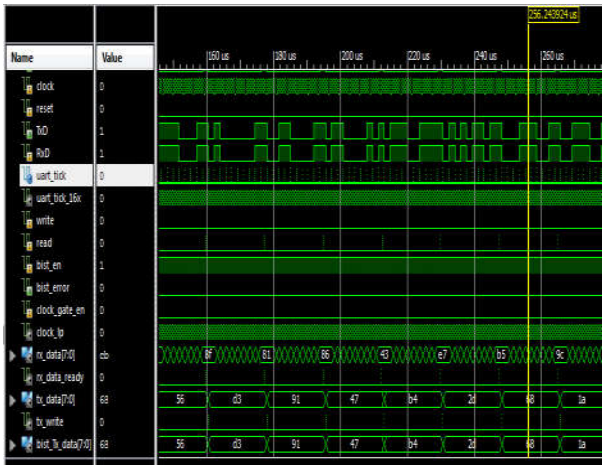


Figure 4.1: Simulated Behavioral Model of low power BIST enabled UART

B. Simulation result for UART with MIPS soft-core processor Application

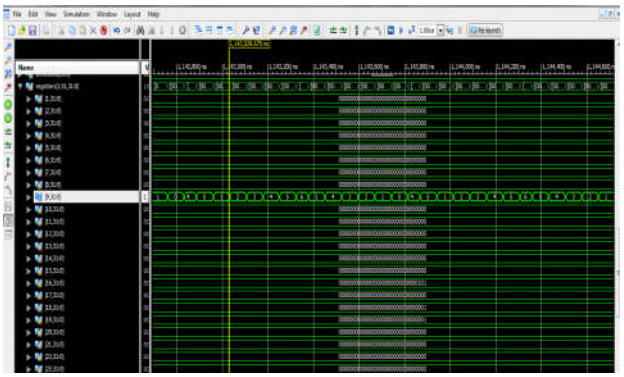


Figure 4.2: Simulated Behavioral Model of UART with MIPS soft-core processor application

V. CONCLUSION

The low power UART with BIST architecture has been designed and simulated in Xilinx 14.7 and the application of UART with MIPS soft-core processor has been implemented on FPGA.

REFERENCE

[1] "VHDL Implementation of UART Design with BIST capability", Mashkuri Yaacob, Mohd Yamani Idna Idris, zaidi Razak.
[2] Implementation of BIST Capability using LFSR Techniques in UART, Balwinder Singh, Shikha Kakar, and Arun Khosla, in the proceedings of International Journal of Recent Trends in Engineering, Vol 1, May 2009, No.3.
[3] "UART realization with BIST architecture using VHDL", Dr.T.V.S.P. Gupta, Y. Kumari, M.Asok Kumar, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 3, Issue 1, pp. 636-640, JANUARY-February 2013.
[4] VHDL Implementation of Universal Asynchronous Receiver Transmitter, Saakshi Kedar, S. D. Chede, S. M. Sakahre.
[5] "Implementation of UART with BIST Technique in FPGA", Premananda B S, Bibin M C.