Comparison between Virtex-6 Technologies on FPGA for UART Transmission

Ramandeep Singh, Kiranpreet Singh, Sandeep Singh

***Abstract*— This paper presents the hardware implementation of high speed Universal Asynchronous Receiver/ Transmitter) UART using (Field Programmable Gate Array) FPGA. UART is an integrated circuit containing a transmitter (parallel to serial converter) and a receiver (serial to parallel converter) each clocked separately. It transmit 9600 to 34800 bps for transmitting data bit. A high speed UART using 90 nm and 40nm technologies has been designed by using FPGA's Virtex 4 and Virtex 6 synthesized on the xilinx ISE 12.4i in VHDL language. By comparing these two technologies on the basis of number of slices, look up tables, GCLK's, slice flip-flops and maximum frequency, it was seen that 40nm technology consumes less area and is 3 times faster than 90nm.**

***Index Terms*—UART, FPGA, VHDL, GCLK's**

I. INTRODUCTION

An UART is a silicon microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232C DTE interface so that it can talk to and exchange data with modems and other serial devices (Conway and Dehaan, 2011). The UART consists of a transmitter, a baud rate generator and a receiver (Sharma and Gupta, 2009).

The rate at which the data is transmitted is known as Baud rate (Joshi et al, 2011). UART is mainly used because of its high speed, lesser cost and more fidelity as compared to ASICs. There are two primary forms of serial communication: synchronous and asynchronous. Synchronous serial transmission requires that the sender and receiver share a clock with one another, or that the sender provide a strobe or other timing signal so that the receiver knows when to "read" the next bit of the data. In most form of serial synchronous communication, if there is no data available at a given instant to transmit, a fill character must be sent instead so that data is always being transmitted. Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver (Kaur and Kaur, 2012). Asynchronous serial communication because of less transmission line, high reliability, and long transmission distance is usually implemented by UART (Yuan and Xue, 2011). VHDL has been used to implement core functions of UART and integrate them into a FPGA chip. As we are using Finite State Machine for transmitter due to which our design has become less complex and the proposed UART becomes more stable, reliable and compact for serial data transmission (Wakhle, 2012). Due to which, the consumption of LUT's, flip-flops or in short the area consumption of the chip becomes less. We have also tested our design for the errors which arises during transmission of data to analyze that our output is free from the errors.

II. METHODS

VHDL code for UART transmitter is designed. VHDL programs have been implemented and tested on XILINX project Navigator Release 12.4i. Device utilization of higher (40nm) and lower (90nm) technology is taken. Timing summary of both technologies is taken. The performance of modified UART transmitter is evaluated by giving the tight timing constraints which provide better results then corresponding section at lower technology. Figure 1 shows the flow chart of UART transmitter and explains the functionality of transmitter that how data has been transmitted. Transmitter section of UART is working upon transmitter clock "SYSCLK". The VHDL code is divided into two main blocks i.e transmit control block and transmit update block. The transmit control block has a state machine with three states i.e idly, synchronous and transmit data. The state machine enters the IDLE state when reset is activated. It checks the external TDRE signal, when it is low it loads the TSR signal. After loading the machine enters the synchronous state, at rising edge the machine goes to transmit state. In this state the zeroth bit represents the start bit, then the shifting starts till the value of BCT reaches 9. After this the machine goes to IDLE state and the process continues.

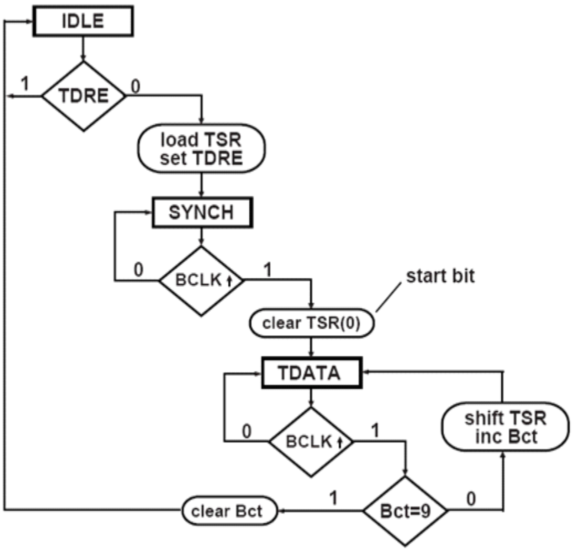


Figure. 1: Flow Chart of UART Transmitter

Manuscript received April 06, 2014

Ramadeep Singh, Electronics and communication department, Guru Nanak Dev Engineering college, Ludhiana, India, +9914700358

Kiranpreet Singh, Electronics and communication department, Guru Nanak Dev Engineering college, Ludhiana, India, +9855100093

Sandeep Singh Gill, Head of Electronics and communication department, Guru Nanak Dev Engineering college, Ludhiana, India, +9814801718

III. RESULTS

All the VHDL programs have been implemented and tested on Xilinx Project Navigator Release ISE 12.4i. The detailed summaries of outputs are given in coming section. The VHDL code is written in Xilinx ISE software by selecting a new project whose name is UART_Tx. The behavioural model of UART Transmitter is implemented using VHDL. After implementation of VHDL code of UART Transmitter, RTL and Technology Schematic is generated by the tool. After implementation of VHDL code, figure 2 shows the Block diagram of UART Transmitter. Figure 3 and 4 show the RTL and Technology Schematic generated by the tool.RTL Schematic shows basic gates (like and, or, nand, nor, not etc), combinational circuits (like mux etc) and sequential circuits (flip-flops). Technology Schematic shows basic blocks i.e. configurable logic blocks (CLBs of FPGA). The CLBs of FPGA consist of look up tables LUT, buffer, mux and flip-flop. CLBs are the programmable logic components of FPGA. A logic block consists of a few logical cells called slices. CLBs are surrounded by a powerful hierarchy of versatile routing resources. FPGA manufacturers try to provide just enough tracks so that most designs that will fit in terms of LUTs and IOs can be routed. Table 1 shows the comparison of two different technologies i.e. lower technology (90nm-virtex4) and higher technology (40nm-virtex6) on the basis of number of slices, look up tables, GCLK’s, slice flip-flops and maximum frequency. While doing comparison it is found that virtex technology shows great significance as all the parameters are consuming less area and giving results at faster rate.

The VHDL programs have been implemented and tested on Xilinx Project Navigator Release ISE 12.4i. After implementation of VHDl code of UART Transmitter, RTL and technology schematic is generated by the tool as shown in Figure 2. It converts verilog code into RTL schematic. RTL schematic of UART Transmitter is made up of basic gates (like and, or, nand, nor, not etc), combinational circuits (like mux etc) and Sequential circuits (Flip-Flops). The RTL schematic is generated by execution of RTL synthesizer which converts VHDL code into RTL schematic made up of basis gates, combinational circuits and sequential circuits as shown in Figure 3. The Technology Schematic is generated by execution of technology Synthesizer. It converts basic verilog code into technology schematic. It implements the UART Transmitter into respective technology which is selected at the formation of project. Technology schematic of UART Transmitter is made up of basic blocks (CLBs) of FPGA. The CLBs of FPGA consist of LUT, buffer, mux and flip flop. Figure 4 shows the technology schematic as generated by execution of technology synthesizer which converts basic VHDL into technology schematic. It is made up of basic blocks of FPGA such as LUT, buffer, mux and flip flops.

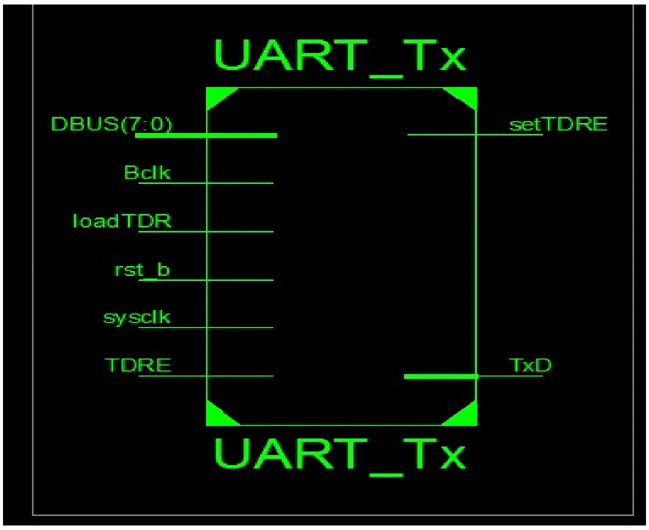


Figure 2. Block diagram of UART transmitter

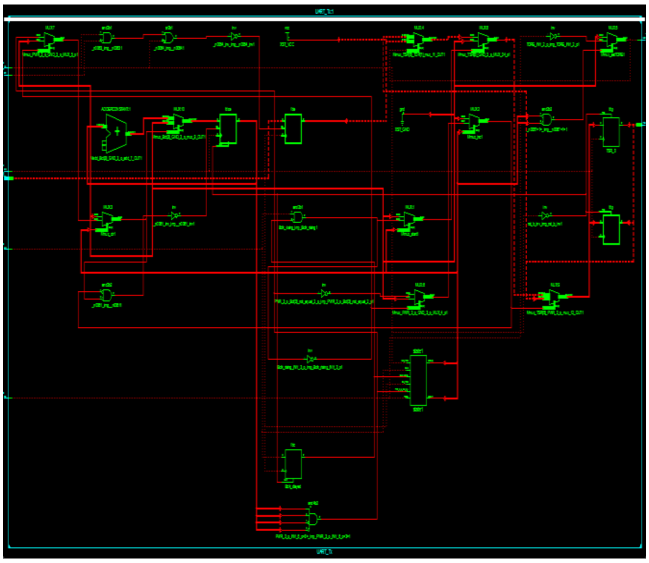


Figure 3. RTL view of UART transmitter

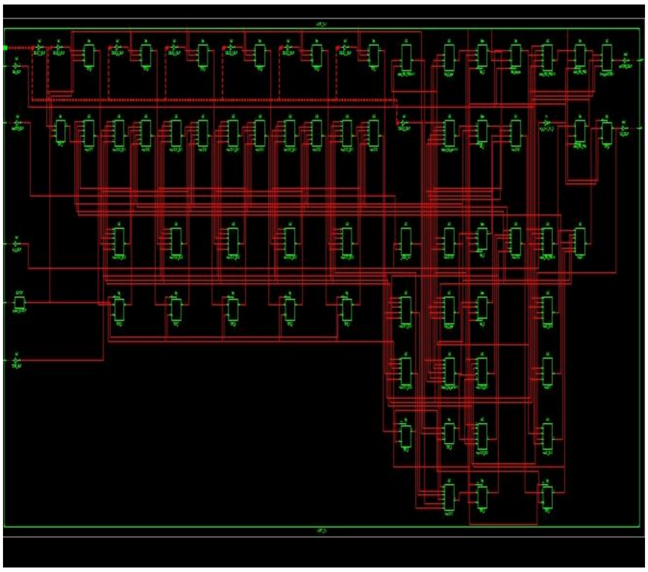
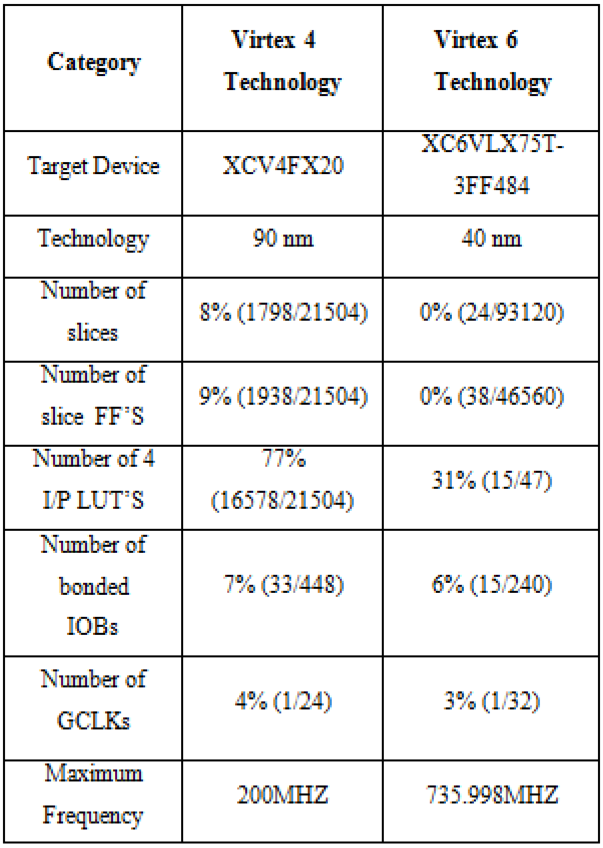


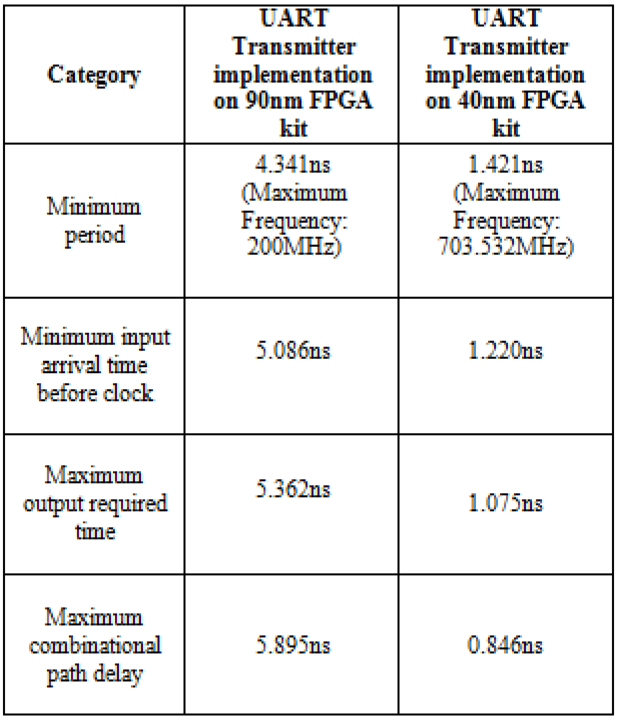
Figure 4. Technology Schematic view of UART transmitter

Table 1 shows the comparison between two different technologies on the basis of number of slices, number of flip flops, maximum frequency, input/output blocks, Gclks used, number of bonded blocks. Comparison on the basis of number of slices used states that Virtex 4 uses 1798 slices out of 21504 availabled slices i.e 8% as compared to almost 0% in Virtex 6. Similarly, Virtex 4 uses 9% of flip flops available 77% of 4 I/P LUT'S, 7% I/P blocks, 4% GCLKs and 200MHZ maximum frequency as compared to 0% flip-flops, 31% of 4 I/P LUT's, 6% I/P blocks, 3% GCLKs and 735.998MHZ maximum frequency used respectively.



Category	Virtex 4 Technology	Virtex 6 Technology
Target Device	XCV4FX20	XC6VLX75T-3FF484
Technology	90 nm	40 nm
Number of slices	8% (1798/21504)	0% (24/93120)
Number of slice FF'S	9% (1938/21504)	0% (38/46560)
Number of 4 I/P LUT'S	77% (16578/21504)	31% (15/47)
Number of bonded IOBs	7% (33/448)	6% (15/240)
Number of GCLKs	4% (1/24)	3% (1/32)
Maximum Frequency	200MHZ	735.998MHZ

Table 1: Comparison between two technologies



Category	UART Transmitter implementation on 90nm FPGA kit	UART Transmitter implementation on 40nm FPGA kit
Minimum period	4.341ns (Maximum Frequency: 200MHz)	1.421ns (Maximum Frequency: 703.532MHz)
Minimum input arrival time before clock	5.086ns	1.220ns
Maximum output required time	5.362ns	1.075ns
Maximum combinational path delay	5.895ns	0.846ns

Table 2: Timing summary of outputs

Table 2 shows the timing summary of UART Transmitter implemented on 90nm and 40nm FPGA
kit in which time used by 40nm FPGA kit to produce output is 1.421ns as compared to 4.341ns time taken by 90nm FPGA kit at a frequency of 703.532MHz and 200MHz respectively.

IV. CONCLUSIONS

The purposed modified UART transmitter section using higher technology at 40nm work which is 3 times faster than the transmitter section implemented at lower technology 90nm. The performance is evaluated by giving the tight timing constraints which provide better results than corresponding section at lower technology. The result of present study revealed that by using 40nm technology with improved architecture can enhance the speed of UART by 3 times the 90nm technology. This implementation uses VHDL to get the modules of UART. After studying the comparative analysis it was seen that the results are quiet stable and reliable which provide high bps rate.

V. FUTURE SCOPE

The study can be further extended to the following issues:

- The study only includes transmitter section of UART and further work could be done in receiver section.
- The performance of UART can be analyzed for different architecture so as to get optimum results.
- Moreover, the study could be extended as power analysis in terms of transmitter and receiver.

REFERENCES

[1] Wakhle G.B. and Gaba I. (2012), "Synthesis and Implementation of UART using VHDL codes", Computer, Consumer Control, pp. 1-3.

[2] Kaur A and Kaur A. (2012), "An approach for designing a UART", International Journal of Engineering Research and Applications, vol.2, Issue 3, pp. 2305-2311.

[3] Chakraborty A, Surbhi, Gupta S, Deshkar S and Jaisal P. k. (2012), "Design of Universal Asynchronous Receiver Transmitter Using VHDL", *International Journal of Advances in Computer Science and Technology,* Vol. 3, Issue 1.

[4] Wang Y and Song K. (2011), "A new approach to realize UART", Electronic and Mechanical Engineering and Information Technology, pp. 2749-2752.

[5] Yuan F. and Xue C. (2011), "Design and simulation of UART Serial Communication Module based upon VHDL", Intelligent Systems and Applications ,pp. 1-4.

[6] Spartan-3E Starter kit Board User Guide. UG230-Xilinx, v.1.2, January 11, 2011 available at http://www.xilinx.com/support/documentation/boards_and_kits

[7] Sharma P and Gupta A. (2009), "Design, Implementation and Optimization of Highly Efficient UART" , The IUP Journal of Science and technology, Vol: 5, No. 4, pp. 21-30.

[8] Joshi K., Patel D., Patel C. and Kher R. (2011), "Design and simulation of UART IP core for FPGA implementation", National Conference on Recent Trends in Engineering & Technology.

[9] Kolte V.S (2012), "VLSI Design and Technology", Techmax Publicatios, pune-411009. pp. 48-49.

[10] Norhuzaimin J and Maimun H.H. (2005), "The Design of High Speed UART", Asia-Pacific Conference on applied Electromagnetics.