Volume 4, Issue 12, December 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Design and VHDL Implementation of UART with Error Status Register

Ketan Mishra
M.Tech Scholar
Lakshmi Narain College of Technology
Bhopal, India

Soheb Munir
Assistant Professor
Lakshmi Narain College of Technology
Bhopal, India

Abstract— Serial communication having the advantages over parallel communication in terms of system complexity and cost. Signal distortion rate is higher in parallel communication because data bits are transmitted at the same time. Serial communication having simple structure reduces the signal distortion and applied over long distance communication. A Serial communication protocol Universal Asynchronous Receiver Transmitter (UART), widely used for low speed, short distance, low cost data exchange between processor and another peripheral devices. UART provides full duplex serial communication link, also used in control system and data communication so that UART function must be realized over single chip. The paper represents the architecture of UART with error status register to check out overrun error, parity error, framing error and break error. Complete design is synthesized and verified by Xilinx ISE Simulator.

Keywords— Universal Asynchronous Receiver Transmitter, VHDL implementation; Xilinx ISE Simulator, Error Status register, Asynchronous serial communication.

I. INTRODUCTION

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 protocol (where data is sent one bit at a time). The serial port is usually connected to UART, an integrated circuit which handles the conversion between serial and parallel data [1] [4] [6] [7].

Asynchronous serial Communication has advantages of less transmission lines, long transmission distance and high reliability. UART allows full-duplex communication in serial link, thus has been widely used in the data communications and control system UART needs only two signal lines called Receive and Transmit to complete full-duplex data communication [2] including three modules, the baud rate generator, receiver and transmitter. The baud rate generator produces a local clock signal which is much higher than the baud rate to control the UART receive and transmit; UART is used for asynchronous serial data communication by converting data from parallel to serial at transmitter and transmit to receiver with some extra overhead bits using shift register. Than receiver get the original data by converting them serial to parallel.

When the transmitter is idle, the data line is in the high logic state. Otherwise when a word is given to the UART for asynchronous transmissions, "Start Bit" (logic low) is added to the beginning of each word that is to be transmitted. The Start Bit is used to prepared the peripheral receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. After the Start Bit, the unit data bits of the word are sent, with the Least Significant Bit (LSB) being sent first. Each bit is transmitted for exactly the same amount of time as all of the other bits, and the receiver samples at the wire at about halfway through the period assigned to each bit to conclude if the bit is a 1 or a 0. When the whole 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 at least one Stop Bit is sent by the transmitter.

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 new word can be sent as soon as the Stop bit for the previous word has been sent. Asynchronous data are "self-synchronizing" if there are no data to transmit, the transmission line is held idle.

In field of electronic design, System on Chip technology requires to realize the complete system function in a single chip. Integration of only core functions over FPGA chip to achieve compact, stable and reliable data transmission avoids waste of resources and decrease cost.

The paper is prepared into 5 sections.

Section-I gives the Introduction. VHDL Implementation explains in section II. Section-III describes the Proposed UART architecture. Section-IV shows the Simulation Results and Synthesis Report. Finally the paper is concluded in section V.

II. VHDL IMPLEMENTATION

Minimum cost and time, better design and increased flexibility are the advantages of VHDL implementation [1],[3]. All these facts have motivated usage of hardware description language in the design process of digital system.

VHDL is used to illustrate and simulate the operation of variety of digital system, ranging in complexity from a few gates to an interconnection of many complex integrated circuits.

III. PROPOSED UART ARCHITECTURE

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.

UART supports asynchronous communication in which clock information is not shared between transmitter and receiver; several overhead bits are sent along with data bits for synchronization purpose. This indicates that data bits are transmitted in the form of frame. This frame is received at the receiver input where de-framing is done and only the data bits are available in parallel form at the receiver output. The design of UART, shown in Fig. 1, has LCR, Baud Rate Generator (BRG), Transmitter and Receiver as its functional units.

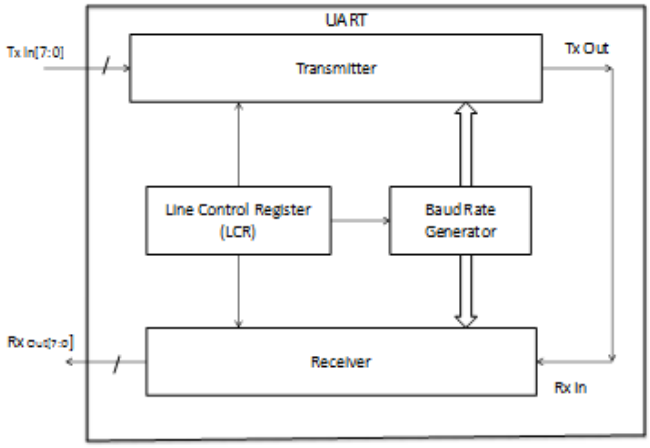


Fig. 1: Proposed UART Architecture

A. Line Control Register consists Baud Rate Generator

The line control register (LCR) is a byte register, used for specification of frame format and desired baud rate. The parity bits, stop bits, baud rate selection and word length can be changed by writing the appropriate bits in LCR.The baud rate generator is programmable by three control bits Bit 0, Bit 1, Bit 2 in LCR as shown in Table 1. 8 different baud rates can be selected by different combinations of Bit 0, Bit 1 and Bit 2. In frame, start bit, parity bit and one stop bit will be added as shown in fig. 2. Now data is transmitted from TSR to TXOUT serially.

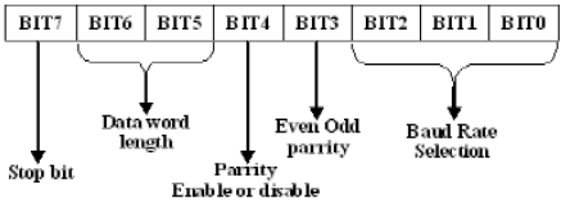


Fig. 2: UART Frame Format

TABLE 1. LCR BIT DESCRIPTION

Bit	Value			Description
	Bit2	*Bit1*	*Bit0*	*BR Sel.*
0,1,2	0	0	0	57600
	0	0	1	38400
	0	1	0	19200
	0	1	1	9600
	1	0	0	4800
	1	0	1	2400
	1	1	0	1200
	1	1	1	600
3	0			Even Parity
	1			Odd Parity
4	0			Parity Disable
	1			Parity Enable
5,6	Bit4		Bit3	DW Length
	0		0	5
	0		1	6
	1		0	7
	1		1	8
7	0			1 Stop Bit
	1			2 Stop Bits

B. UART Transmitter

The transmitter accepts parallel data from peripheral/processor, makes the frame of the data and transmits the data in serial form on the Transmitter Output (TXOUT) terminal (fig. 3). The baud rate generator output will be the clock for UART transmitter.

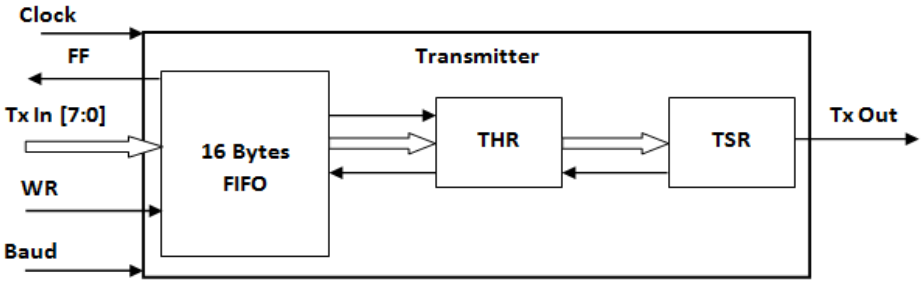


Fig. 3: UART Transmitter

Data is loaded from the inputs TXIN0-TXIN7 into the Transmitter FIFO by applying logic high on the WR input, as shown in fig. 4.If words less than 8 bits are used, only the least significant bits are transmitted. FIFO is 16-byte register. When FIFO contains some data, it will send the signal to Transmitter Hold Register (THR), which is an 8-bit register. At a same time, if THR is empty it will send the signal to FIFO, 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 a 12-bit register in which framing process occurs. Fig.4 & fig.5 show the flowchart explaining transmission of serial data from FIFO to transmitter output.

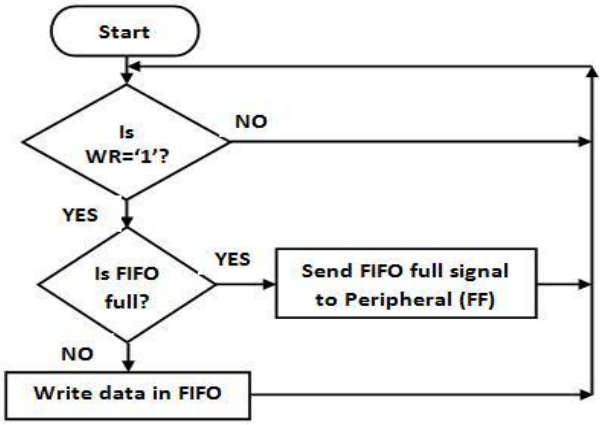


Fig. 4: Transmitter flowchart – Input to FIFO

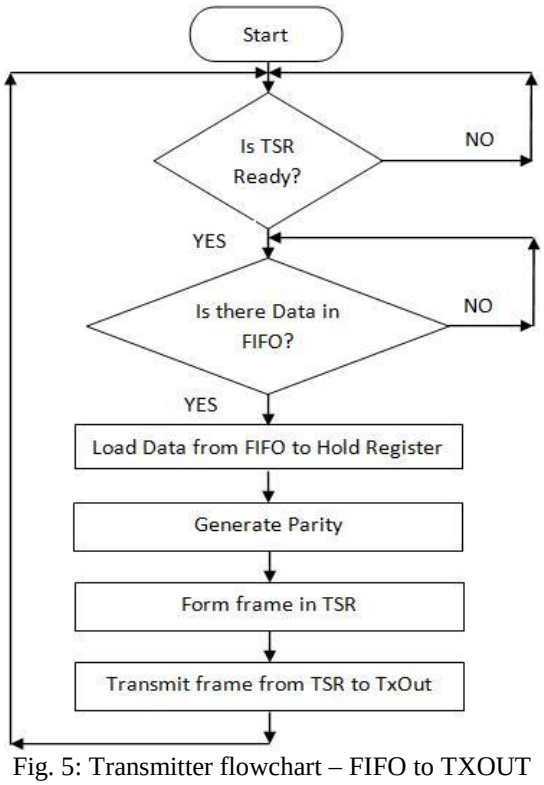


Fig. 5: Transmitter flowchart – FIFO to TXOUT

C. 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.

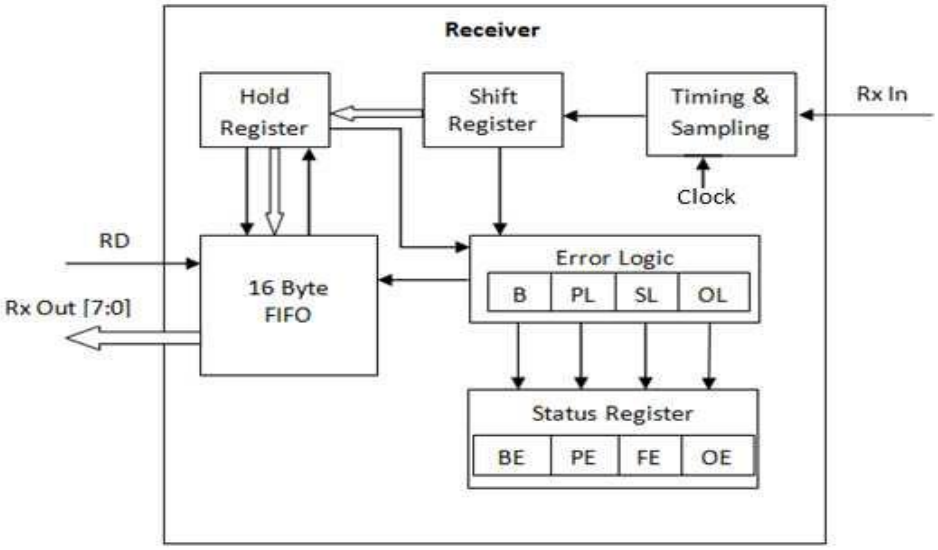


Fig. 6: UART Receiver

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 all the bits are send to Receiver Shift Register (RSR) one by one where the entire frame is stored. RSR is a 12 bit shift register. 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. The remaining bits in the RSR are used by the error logic block. Then, if receiver FIFO is empty it 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. Receiver's flow chart is shown by fig.7.

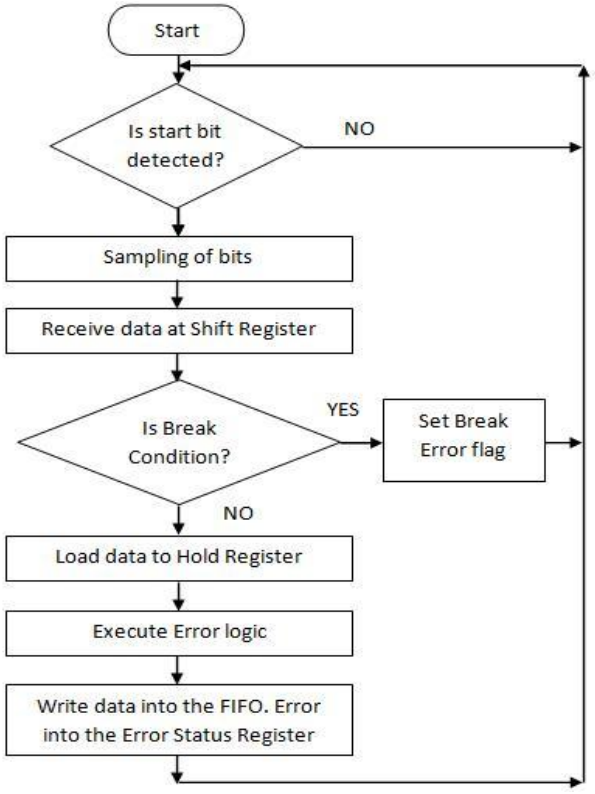


Fig. 7: Receiver flowchart (Input to FIFO)

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. Flow chart shown by fig.8 (FIFO to Output).

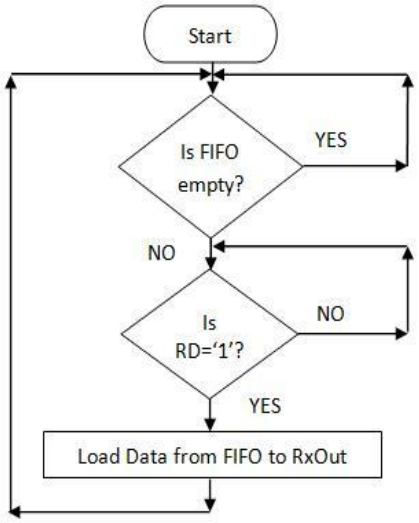
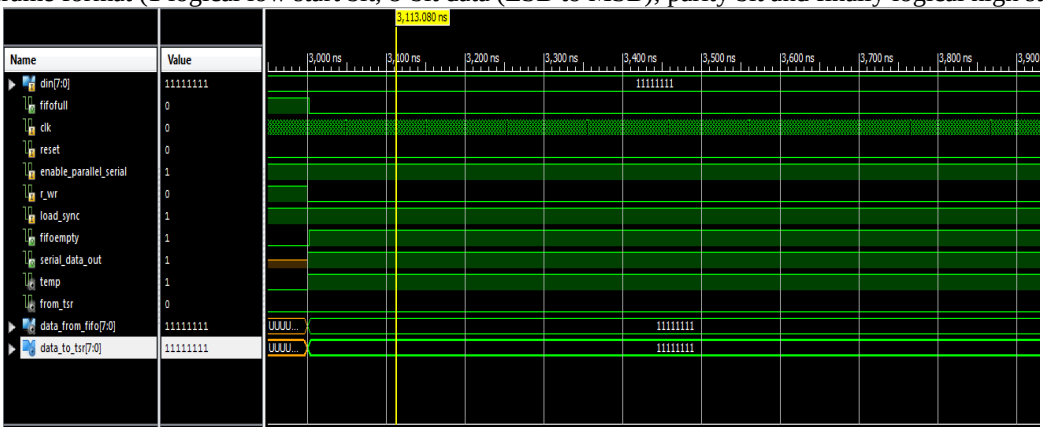


Fig. 8: Receiver flowchart (FIFO to Output)

IV. SIMULATION RESULTS

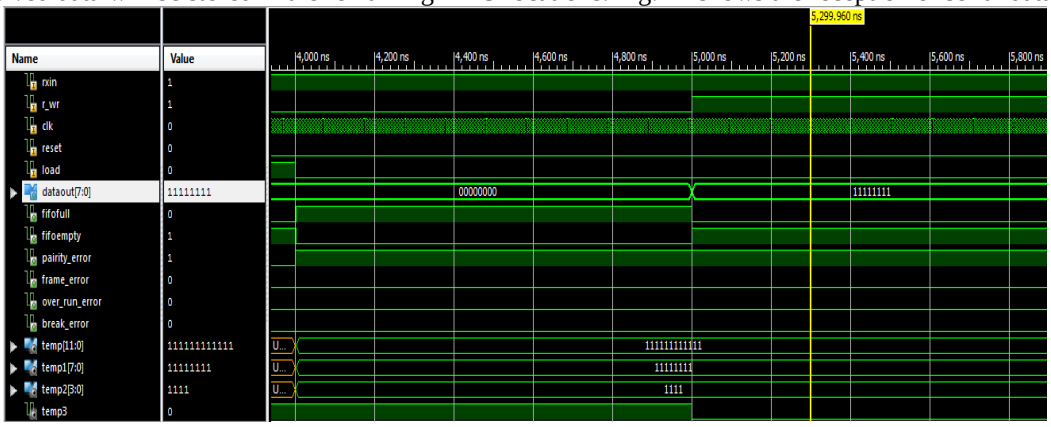
A. Simulation Results of Transmitter

The fig. 10 shows the serial data transmission. This transmit 8-bit data"11111111" is loaded to shift register (data_from_fifo[7:0]) and start, stop and parity bits are added to form the frame inside TSR and sent to data_to_tsr[7:0] by enable_parallel_serial. When the reset is 0, the transmitter starts transmitting the data. Since the desired baud rate is 57600bps, the bits are shifted out using 50MHz system clock. The serial transmission is observed at data_to_tsr[7:0] pin along with frame format (1 logical low start bit, 8-bit data (LSB to MSB), parity bit and finally logical high stop bit).



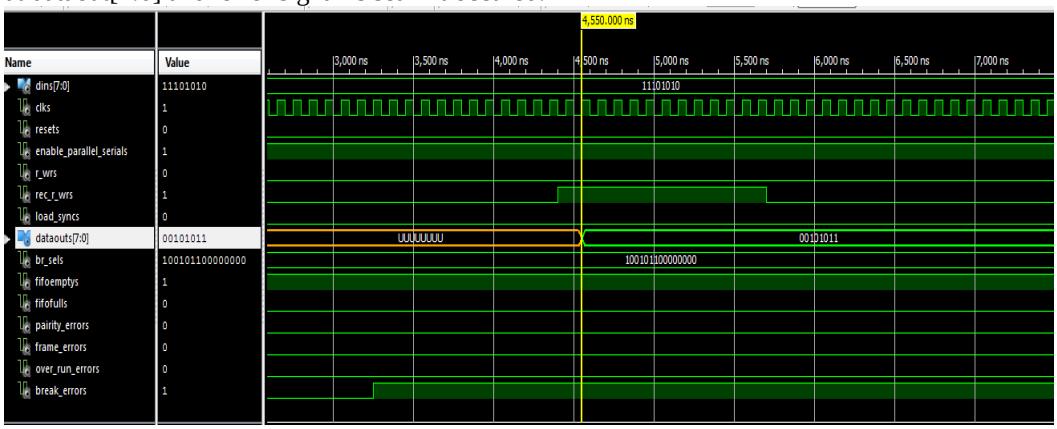
B. Simulation Results of Receiver

By setting rxin '1'Serial data is converted into parallel form by the UART receiver and shows converted data at dataout[7:0] line.. Each bit is sampled and the sampled bit is saved into receive shift register. From this, the frame bits viz. start, parity and stop bits are discarded in RSR and written to receive FIFO. The 8-bit data simulated is "11111111". Further received data will be stored in the remaining FIFO locations. Fig. 11 shows the reception of serial data.



C. Simulation Results of UART

Fig. 11 shows simulation of UART. It shows independent functioning of transmitter and receiver sections. While the transmitter section of UART converts the parallel data (din[7:0]) into serial form and transmit through its output pin at the same time receiver section can receive the data serially through its input pin,converts it into parallel form and makes it available at dataout[7:0] and error signal is set if it occured.

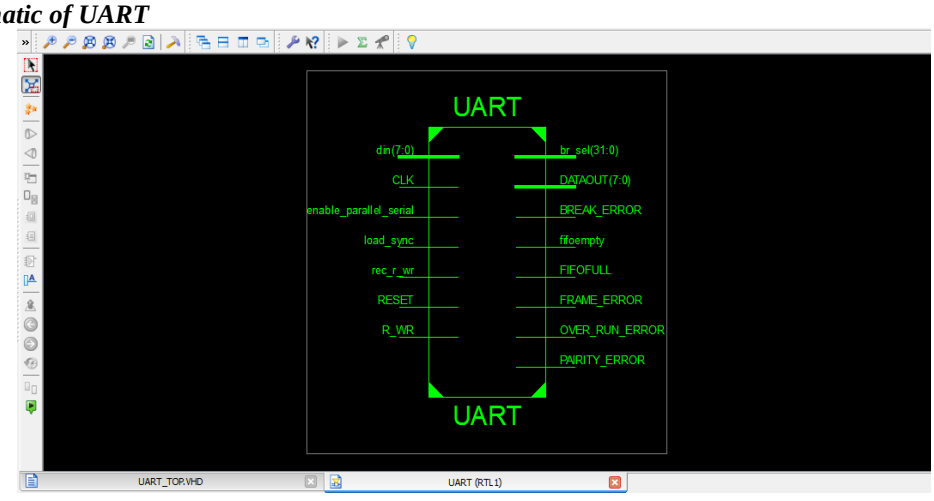


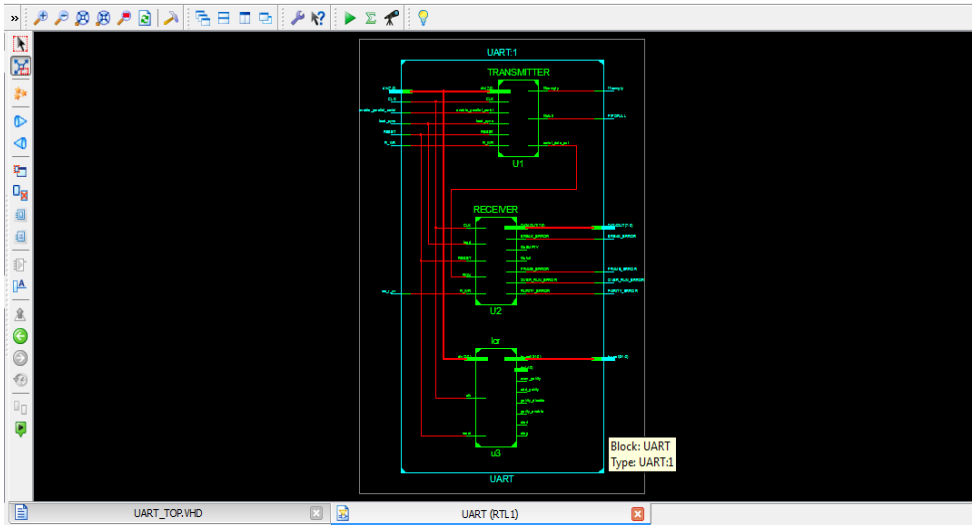
D. Synthesis Report

Table 2. shows the design implementation summary of the proposed architecture.

Design Utilization Summary Selected Device	**Proposed Work** 3s400fg320-4
Number of Slices	299 out of 3584 8%
Number of Slices Flip Flops	389 out of 7168 5%
Number of 4 input LUTs	256 out of 7168 3%
Number of bounded IOBS	60 out of 221 27%
Number of GCLKs	2 out of 8 25%
Timing Summary Speed Grade	-4
Minimum Period	6.022ns(Max.Frequency:166.058 MHz)
Minimum input arrival time before clock	7.803ns
Minimum output arrival time after clock	7.165ns
Maximum combinational path delay	No path found

E. RTL Schematic of UART





V. CONCLUSION

Paper shows the architecture of UART that support 8-bit data word length at different-different baud rates for serial transmission of data with the addition of error status register which can detect the different types of errors occurred during communication and hence correct them in data transfer. Working of UART has been tested using Xilinx ISE simulator, which is implemented on FPGA.

REFERENCES

[1] Naresh, Vatsalkumar and Vikaskumar Patel, "VHDL Implementation of UART with Status Register", in the proceedings ofInternational Conference on Communication Systems and Network Technologies, IEEE Computer Society, 11-13th May 2012, DOI: 10.1109/CSNT.2012.164, pp.750-754.

[2] Shumit Saha, Md. Ashikur Rahman, Amit Thakur; " Design and Implementation of a BIST Embedded High Speed RS-422 Utilized UART over FPGA ,"IEEE 4th ICCCNT July 4 - 6, 2013, Tiruchengode, India.

[3] Idris, M.Y.I.; Yaacob, M.; , "A VHDL implementation of BIST technique in UART design," TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region , vol.4, no., pp. 1450- 1454 Vol.4, 15-17 Oct. 2003.

[4] Fang Yi-yuan; Chen Xue-jun; , "Design and Simulation of UART Serial Communication Module Based on VHDL," Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on , vol., no., pp.1-4, 28-29 May 2011.

[5] Nitin Patel; Naresh Patel; " VHDL Implementation of UART with BIST capability,"IEEE 4th ICCCNT July 4 - 6, 2013, Tiruchengode, India.

[6] Mohd.YamaniIdna Idris,Mashkuri Yaacob,Zaidi Razak,"A VHDL Implementation of UART Design With BIST capability",Faculty of Computer Science and Information Technology,University of Malaya", in the proceedings of Malaysian Journal of Computer Science, June 2006,Vol. 19(1), pp. 73-86,50603 Kuala Lumpur,Malaysia.

[7] Shikha kakar,Balwinder Singh,Arun Khosla, "Implementation of BIST capability using LFSR technique in UART", International Journal of Recent Trends in Engineering ,Vol 1, No. 3, May 2009,pp.301-304.

[8] Bibin M C, Premananda B S, " Implementation of UART with BIST Technique in FPGA," International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-1, Issue-8, July 2013.

[9] Prof Naresh Patel,Vatsal Patel,Vikas Patel,"VHDL Implementation of UART with Status Register",Sarvajanik College of Engg. And Technology.

[10] 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.

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

[12] 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. 633-636, Oct. *2000*.

[13] Dr. T.V.S.P.Gupta, Y. Kumari, M. Ashok Kumar,"UART realization with BIST architecture using VHDL" / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622,Vol. 3, Issue 1, January -February 2013, pp.636- 640.

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

[15] Y. C. Wang and K. Song, "A New Approach to Realize UART," 2011International Conference on Electronic & Mechanical Engineering and Information Technology, 12-14 August, 2011.

ABOUT AUTHOR

Ketan Mishra is M.Tech Scholar (VLSI Design) from Lakshmi Narain College of Technology,Bhopal(M.P.),Under R.G.P.V. Completed his Bachelor of Engineering in Electronics & Tele Communication from Priyadarshini College of Engineering, Nagpur (M.H.), Under Rashtra Santh Tukadogi Maharaj Nagpur University.

Soheb Munir received his Bachelor's and Master's from R.G.P.V. Bhopal and currently working as a Assistant professor in Lakshmi Narain College of Technology, Bhopal(M.P.). His area of interest is Image Processing.

