

VHDL Implementation of Universal Asynchronous Transmitter

Sakshi S. Kedar^{#1}

Electronics (Communication)
Suresh Deshmukh College Of Engineering
Wardha, India

Dr. S. D. Chede^{*2}

Electronics & Communication
OM College Of Engineering
Wardha, India

Abstract— The paper highlights the detail design of UART's transmitter module which has asynchronous Last In Last Out as buffer for achieving the better speed synchronization between the processor and the UART interface. The paper starts by describing the behavior of transmitter module of Universal Asynchronous Receiver Transmitter using VHDL. The complete design of transmitter block is functionally verified using ModelSim SE 13.1. In the results the simulation waveforms are described.

Keywords— *Universal Asynchronous Receiver Transmitter; asynchronous LIFO; status registers; parallel to serial converter, ISE Simulator; VHDL Implementation.*

I. INTRODUCTION

The fact that serial communication uses a single data line instead of the 8 bit data line of parallel communication not only makes it much cheaper but also enables two computer located in to different cities to communicate over the telephone. Universal Asynchronous Receiver Transmitter (UART) is a kind of serial communication protocol; mostly used for short-distance, low speed, low-cost data exchange between computer and peripherals [1]. Serial data communication uses two methods, asynchronous and synchronous. The synchronous method transfers a block of data (character) at a time, while the asynchronous method transfers a single byte at a time. It is possible to write software to use either of these methods, but the programs can be tedious and long. For this reason, there are special IC chips made by many manufacturers for serial data communications. These chips are commonly referred to as UART i.e Universal Asynchronous Receiver Transmitter. Asynchronous serial data communication is widely used for character-oriented transmissions, while block oriented data transfer use the synchronous method. In asynchronous method, each character is placed between start and stop bits. This is called framing. In data framing for asynchronous communications, the data such as ASCII characters, are packed between start and stop bits. The start bit is always one bit, but the stop bit can be one or two bits. For asynchronous serial communication, peripherals chips and modems can be programmed for data that is 7 or 8 bits wide. The UART implemented with VHDL language can be integrated into the FPGA to achieve compact, stable and

reliable data transmission [2]. In recent years the researchers have proposed various UART designs like automatic baud rate synchronizing capability[3], recursive running sum filter to remove noisy samples[4], predictable timing behavior to allow the integration of nodes with imprecise clocks in time-triggered real-time systems[5]. The UART chip can also be designed with embedded Built-In-Self- Test (BIST) architecture using FPGA technology. UART allows full-duplex communication in serial link, thus has been widely used in the data communications and control system [2]. UART is an asynchronous serial communication device. The transmitter and receiver communicate each other by separate clocks [6].

II. VHDL IMPLEMENTATION

Complex nature of the integrated circuit technology has made the designing of the system, time consuming at the gate and the flip-flop levels. This fact motivated the designers to make use of hardware description language in the design process of digital system. VHDL is a hardware description language that can be used to model a digital system. It contains elements that can be used to describe the behavior or structure of the digital system, with the provision for specifying its timing explicitly. VHDL is a large and verbose language with many complex constructs that have complex semantics meaning and is initially difficult to understand. The language supports for modeling the system hierarchically and also supports top-down and bottom-up design methodologies. VHDL is extensively used today for design of VLSI based digital systems. VHDL provides easier way to test the Universal Asynchronous Receiver Transmitter and helps to find out discrepancy if occurs. It also permits to describe the function of the module in a more behavioral manner. VHDL makes the design implementation easier to read and understand, they also provide the ability to easily describe dependencies between various processes that usually occur in such complex event-driven systems. [8].

III. PROPOSED UART ARCHITECTURE

The architecture of Universal Asynchronous Receiver Transmitter is shown Fig No. 01. The proposed design of UART includes four functional units; they are transmitter, receiver, baud rate generator and line control register. Asynchronous communication is done using UART wherein the clock information is not communal between the two units that are communicating with each other. For synchronization some overhead bits are added to data bits while transmission which specifies that the data bits are transmitted in the form of frame. The process of de-framing is done at the receiver input once the frame is received. Thus only the data bits are presented in parallel form at the receiver port. UART transmits and receives the entire data in the format as shown in Fig. 2. The data is framed between the start bit and a single stop bit. The data format includes idle state start bit, data bit, parity bit, and stop bit. Idle state is at logic 1. When the transmitter is ready to send a data, firstly start bit is sent which is logic 0. The Start Bit is used to alert the 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. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word [6]. Start bit is followed by data bits, which can vary from 5 to 8 bits. In some systems. the parity bit of character byte is included in the data frame in order to maintain data integrity. This implies that for each character (7 or 8bit, depending on the system) we have a single parity bit in addition to start and stop bits. The parity bit can be even or odd or even no parity. In case of an odd parity bit, the number if data bits, including the parity bit, has odd number of 1s. Similarly, in an even parity bit system the total number of bits including the parity bit is even.

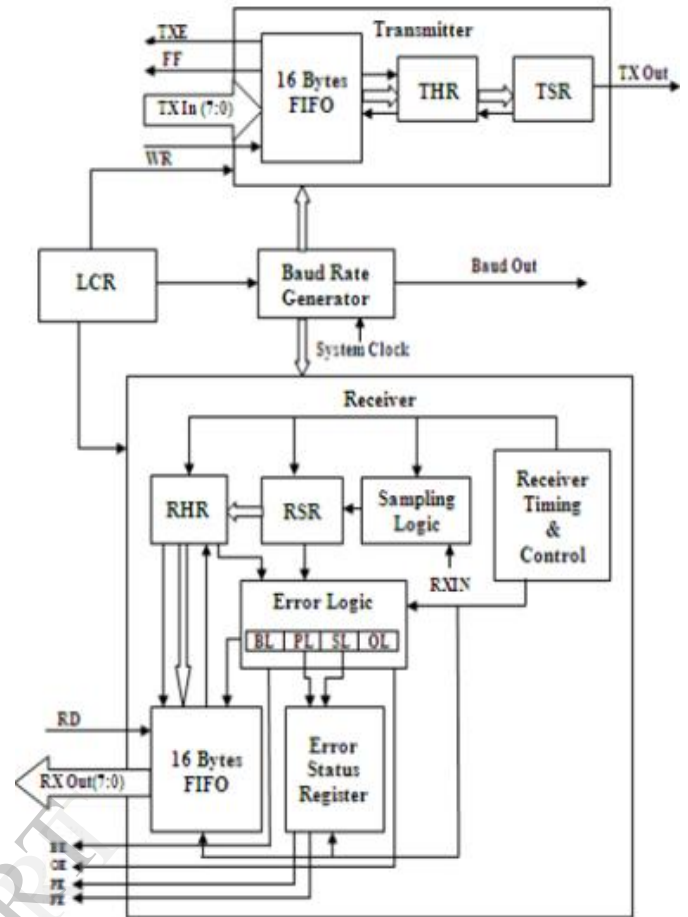


Fig.1 UART Architecture

For example, the ASCII character, “A”, binary 01000001, has 0 for the even parity bit. UART chips allow programming of the parity bit for odd, even and no parity. The parity bit is followed by stop bit which is logic 1. In case if there is no parity, the data bits are followed by stop bit. Thus, start bit is the beginning of the data while the stop bit is the end of the data. The time width is a baud rate clock cycle. The character synchronization can be achieved using this data format. Notice that the LSB is sent out first the rest of the bits until the MSB and finally the one stop bit indicating the end of the character.

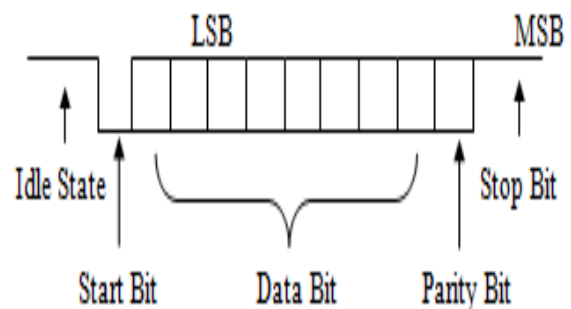


Fig. 1 UART Frame Format

IV. THE TRANSMITTER MODULE

The transmitter module deals with the parallel data at TXIN port which is converted into serial form for transmission purpose on the transmitter output TXOUT port. The RTL view of transmitter module is as shown in Fig. 3. The module comprises of three basic blocks, 16 bytes buffer (LILO), Transmitter Hold Register (THR) and Transmitter Shift Register (TSR). When the write pin is at logic high, data is fed from TXIN0 - TXIN7 into the transmitter LILO. When LILO is loaded with some data, it will send the signal to transmitter hold register (THR). THR is a 8 bit register. If THR is empty and ready to receive the data from LILO it will send a signal to LILO as an acknowledgement. Further if Transmitter Shift Register (TSR) is empty it will send signal to THR indicating that it is ready to receive the data from THR. TSR is 12 bit register. The process of framing is done in TSR where start bit, parity bit and stop bits are added to the frame. Then serially the data is transmitted from TSR to TXOUT port.

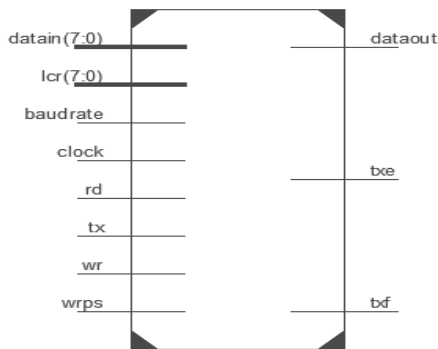


Fig. 3 RTL View Of Transmitter

The body of transmitter module has 16 bytes buffer which is Last In Last Out. The RTL View of LILO buffer is shown in Fig. 4. The inputs to the LILO buffers are datain (0-7), clock, RD and WR while the outputs are dataout (0-7), liloempty and lilo full.

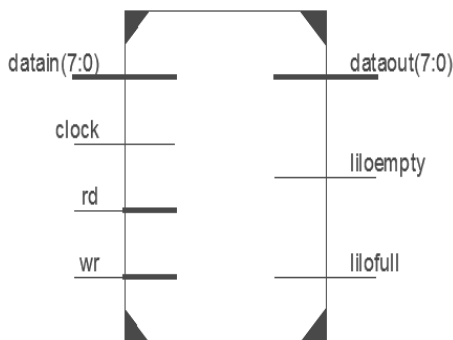


Fig. 4 16 Bytes LILO

The working of the transmitter is shown in terms of flowchart [5]. Fig 5 shows the flowchart which indicates the flow of data from input to LILO. The Fig 6 shows the flowchart which implies the transfer of data from LILO to output. Before transmission framing is done in Transmitter Shift Register where parity bit, start bit and the stop bits are added to the data.

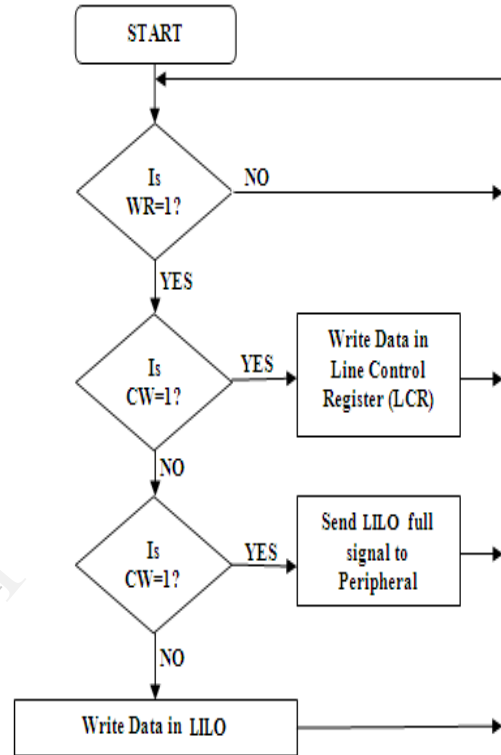


Fig. 5 Transmitter Flowchart (Input to LILO)

The simulation waveform of LILO buffer is as shown in Fig. 7. The data coming at the input terminal TXIN (0-7) is fed to the buffer serially when the WR is high. The data that is fed to the buffer lastly will be out at the end of the data. The term last come last serve can be used for the same. The data at the output port Data out (7-0) is then to be forwarded to the Transmitter Hold Register.

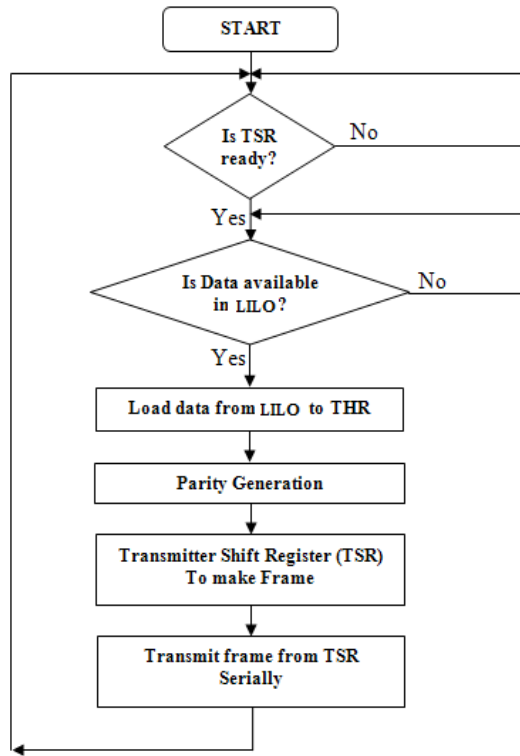


Fig. 6 Transmitter Flowchart (LIFO to Output)

Next to the 16 byte buffer we have Transmitter Hold Register (THR). The data coming out from buffer is shifted to Transmitter Shift Register through this THR. The RTL view of parallel to serial converter is as shown in Fig. 8 while the Fig 9 explains the Simulation waveform for the same.

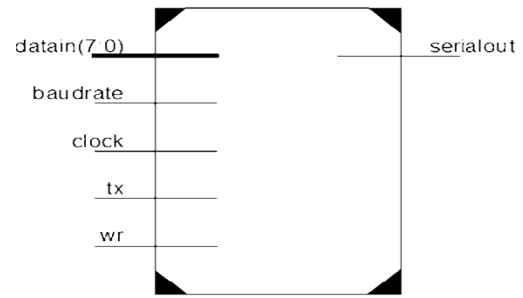


Fig. 8 RTL view of parallel to serial converter

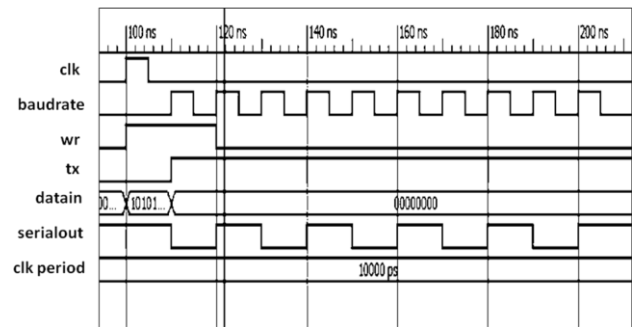


Fig 9 Simulation Result of Parallel to Serial Converter

V. SIMULATION

A. Hardware Utilization

The designed UART is synthesized using Xilinx project navigator. The resource utilization is detailed in Table 1. The table explains the percentage utilization depending upon various parameters such as number of slices register, number of fully used LUT-FF pairs, number of slice LUT's and number of IOB's required to design the entire transmitter module.

Device Utilization Summary (Estimated Values)			
Logic Utilization	Used	Available	% Utilization
No. of Slices Register	11	4800	0%
No. of fully used LUT-FF pairs	11	15	73%
No. of Slice LUTs	15	2400	0%
No. of bonded IOBs	25	102	24%

TABLE 1 RESOURCE UTILISATION

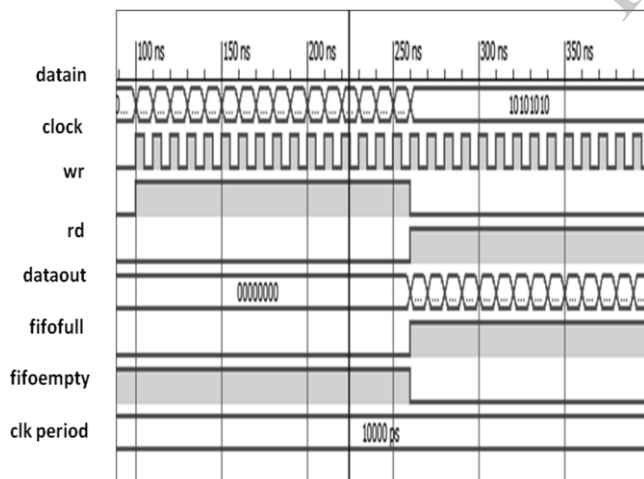


Fig. 7 Simulation Results Of LIFO

B. RTL Schematic Of Top Level Entity

The RTL Schematic generated by Xilinx for Transmitter is shown in fig.10 The logical diagram consist of various gates such as AND, XOR, NOT. The Multiplexers are also used with all these gates in various combinations. At the end of the diagram are LIFO buffer and Parallel to serial converter.

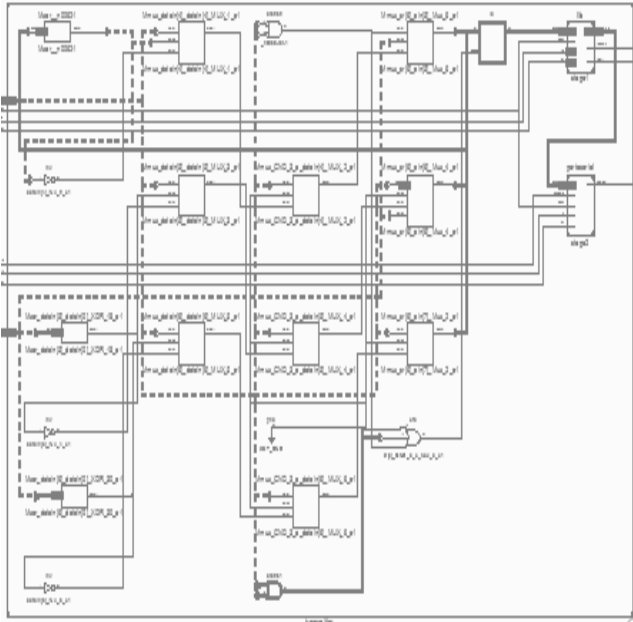


Fig. 10 Block diagram of the synthesized UART

C. Simulation Results Of Transmitter

The simulation results of VHDL implementation of transmitter module of UART is as shown in Fig. 11. The simulation shows the transmission of an 8-bit UART frame format with 1 stop bit and with a parity bit.

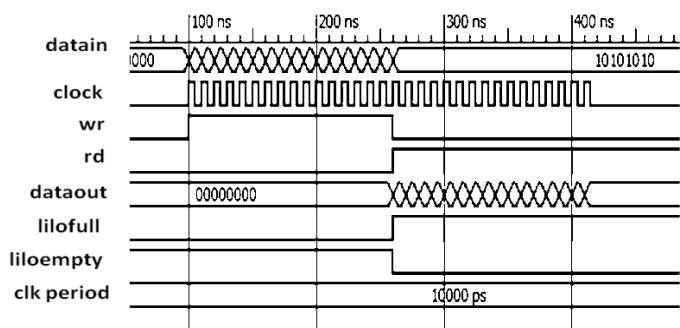


Fig.11 Simulation Result

VI. ACKNOWLEDGMENT

I wish to thank Dr. S. D. Chede and Prof. S. M. Sakhare for their guidance and valuable support.

VII. CONCLUSION

Thus, the design uses VHDL as design language to achieve the transmitter module. Firstly the working principle of transmitter is explained. Then all blocks of transmitter are designed and simulated using VHDL. The design has great flexibility, stability and low cost.

VIII. REFERENCES

- [1] Naresh Patel; Vatsalkumar Patel; Vikaskumar Patel, "VHDL Implementation of UART with Status Register", Sarvajnik College of Engg & Tech Surat, India.
- [2] Liakot Ali , Roslina Sidek , Ishak Aris , Alauddin Mohd. Ali , Bambang Sunaryo Suparjo. Design of a micro - UART for soc application [J].In: Computers and Electrical Engineering 30 (2004) 257– 268.
- [3] Gallo, R.; Delvai, M.; Elmenreich, W.; Steininger, A.; , "Revision and verification of an enhanced UART," Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on , vol., no., pp. 315- 318, 22-24 Sept. 2004.
- [4] Himanshu Patel; Sanjay Trivedi; R. Neelkhanth; V. R. Gujrati; "A Robust UART Architecture Based on Recursive Running Sum Filter for Better Noise Performance," VLSI Design, 2007. Held jointly with 6th International Conference on Embedded Systems., 20th International Conference on , vol., no., pp.819-:823, Jan. 2007.
- [5] Elmenreich, W.; Delvai, M.; , "Time-triggered communication with uarts," Factory Communication Systems, 2002. 4th IEEE International Workshop on , vol., no., pp. 97- 104, 2002J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
- [6] He Chun-zhi Xia Yin-shui Wang Lun-yao, " A Universal Asynchronous Receiver Transmitter Design " (Institute of Circuits and Systems, Ningbo University, Ningbo 315211, China).
- [7] Frank Durda Serial and UART Tutorial. Uhclem@freebsd.org.
- [8] Norhuzaimin, J.; Maimun, H.H.; , "The design of high speed UART," Applied Electromagnetics, 2005. APACE 2005.
- [9] Yongcheng Wang; Kefei Song; , "A new approach to realize UART," Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on , vol.5, no., pp.2749- 2752, 12-14 Aug. 2011.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989. Idris, M.Y.I.; Yaacob, M.; , "A VHDL implementation of BIST technique in UART design," TENCON 2003.
- [10] Convergent Technologies for Asia-Pacific Region , vol.4, no., pp. 1450- 1454 Vol.4, 15-17 Oct. 2003.
- [11] E.E. Reber, R.L. Michell, and C.J. Carter, "Oxygen Absorption in the Earth's Atmosphere," Technical Report TR-0200 (420-46)-3, Aerospace Corp., Los Angeles, Calif., Nov. 1988. (Technical report with report number) .
- [12] Chun-zhi, He; Yin-shui, Xia; Lun-yao, Wang; , "A universal asynchronous receiver transmitter design," Electronics, Communications and Control (ICECC), 2011 International Conference on , vol., no., pp.691-694, 9-11 Sept. 2011.L. Hubert and P. Arabie, "Comparing Partitions," J. Classification, vol. 2, no. 4, pp. 193-218, Apr. 1985. (Journal or magazine citation).
- [14] Prof. Rami Abielmona, "Project: UART Design", for CEG 3150: Digital Systems-II, Fall 2004, November 24, 2004.
- [15] Zou,Jie Yang, Jianning. Design and Realization of UART Controller Based on FPGA.