International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.7, pp : 737-740 1 July 2013

Design and Implementation of UART using FIFO for Serial Communication

Sanjeev Kumar

Department of Electronics and Communication, RKDF Institute of Science and Technology, Bhopal

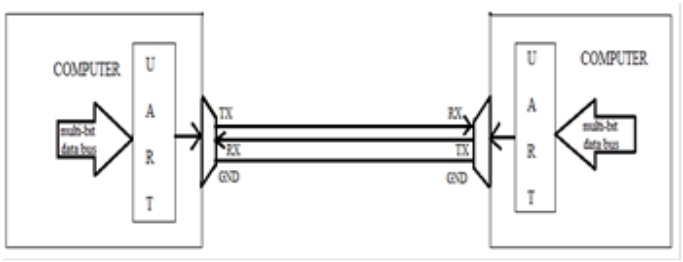
Email id-sanjeev9015@gmail.com

ABSTRACT- This paper presents the design and implementation method of a Universal Asynchronous Receiver Transmitter (UART) as a widely used serial communication protocol using Verilog Hardware Descriptional Language (HDL). In order to achieve the needs of latest complex communication system demands, a UART controller has been designed using FIFO (First In First Out) buffer technique for asynchronous serial data transmission between systems. Also, this reduces the delay, synchronisation error between sub-systems and provides a reliable, high performance logic solution for complex systems. The simulation and synthesis has been carried out using Modelsim DE 6.5 (inc. Mentor Graphics).

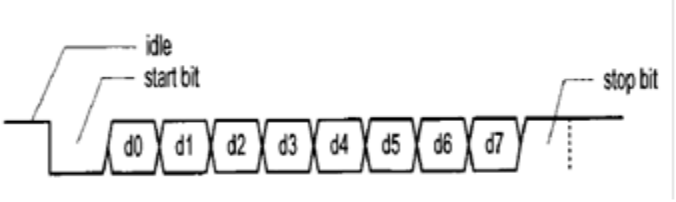
Keywords:- UART, Verilog HDL, FIFO, synchronisation error, baud rate**.**

1. INTRODUCTION

A UART is an integrated circuit that plays the most important role in serial communication. A UART is a circuit that sends parallel data through serial lines[1]. UARTs are used in association with the serial communication EIA standard RS-232. The main function of a UART is parallel-serial conversion during transmission and serial-parallel conversion during reception (for example the communication between a DSP and a PC). In contrast, parallel communication needs a multi-bit address bus and is convenient only for short distance transmission. Serial communication is another means used widely because of its simple design and long transmission distance. Sometimes it is not possible to meet the desired requirements with different baud rate equipments. The communication parameters such as baud rate and synchronisation error also have great impact on system performance. In order to overcome these difficulties, a UART controller can be designed which can transmit and receive data between equipments with different baud rates. The fact that a clock signal is not sent with the data complicates the design of a UART. Fig.1. Communication using UART Figure 1 shows the block diagram of serial communication between two computers using UART. Basic UART communication needs only two signal lines (TxD, RxD) to achieve full-duplex serial data transmission. When transmitting, the UART converts the incoming multi-bit data stream into serial data stream and sends it serially via the TxD pin of RS232 [3].



When receiving, the UART receives the data serially on the RxD pin and provides the parallel data to the application. Fig.2.



Communication using UART There are two states in the signal line: logic '1' and logic '0'. When no data is transmitted, the signal line is in 'idle' state which is generally represented by logic '1' (mark). When a data word is given to UART, the asynchronous data transmission starts with a 'start bit' (logic '0') as shown in the figure 2, which is added at the beginning of each word to be transmitted. The start bit is used to alert the receiver that a data word is about to be sent. The start bit follows the data word to be transmitted. Each bit is transmitted for exactly the same amount of time as all other bits and the receiver ascertains at approximately half-way

through the period assigned to each bit to determine if the bit is a '1' or'0'.

The UART serial communication module is divided into 5 sub-modules: 1. Baud Rate Generator 2.Transmitting Sub-system 3.Receiving Sub-system 4.Transmitter FIFO 5.Receiver FIFO. Designing of these modules will be explained in this paper

2. DESIGN METHODOLOGY

Design and implementation of the above mentioned modules shown in the figure 3 will be discussed in this section.According to the figure, UART transmitter logic receive parallel signal and converted into serial data, the UART receiver logic receive serial signal and converted into parallel signal and FIFO are used to avoid the loss of data. Fig.3. UART with FIFO.

2.1 Baud Rate Generator

The baud rate generator generates sampling ticks whose frequency is exactly 16 times the UART's designated baud rate. To avoid creating a new clock domain and violating the synchronous design principle, the sampling signal should function as enable ticks rather than the clock signal to the UART receiver. To generate 19,200 baud rate, the sampling rate has to be 16 times of specified baud rate which is 307,200 (i.e., 19,200* 16) ticks per second as shown in the figure 3. Therefore each data width received by UART is 16 times the receive clock cycle. Similarly, different baud rates can be generated according to the requirement.

UART with FIFO. 2.1 Baud Rate Generator The baud rate generator generates sampling ticks whose frequency is exactly 16 times the UART's designated baud rate. To avoid creating a new clock domain and violating the synchronous design principle, the sampling signal should function as enable ticks rather than the clock signal to the UART receiver. To generate 19,200 baud rate, the sampling rate has to be 16 times of specified baud rate which is 307,200 (i.e., 19,200* 16) ticks per second as shown in the figure 3. Therefore each data width received by UART is 16 times the receive clock cycle. Similarly, different baud rates can be generated according to the requirement.

2.2 FIFO The FIFO buffer provides more buffering space and also reduces the probability of data overrun [2]. The main function of FIFO buffer is to send and receive data, and reduce the communication time between the serial port and the CPU as shown in figure 3 . Various signals defined to perform the operation of FIFO. A data-overrun error occurs when a new data word arrives and the FIFO is full. We can adjust the desired number of words in FIFO to accommodate the processing need of the main system.

2.3 Transmitting Sub-System The main function of transmitter logic is to read data from the transmitter FIFO, convert parallel data into serial data and send to peripherals. It is implemented using a shift register that shifts out data bits at a specific rate. The rate can be controlled by ticks generated by the baud rate generator module, the frequency of the ticks is 16 times slower than that of the UART receiver. The symbol logic for transmitter is in figure 5. Here, the baud rate of transmitter is a same as UART receiver. A bit is shifted out every 16 enable ticks as shown in the figure 5.

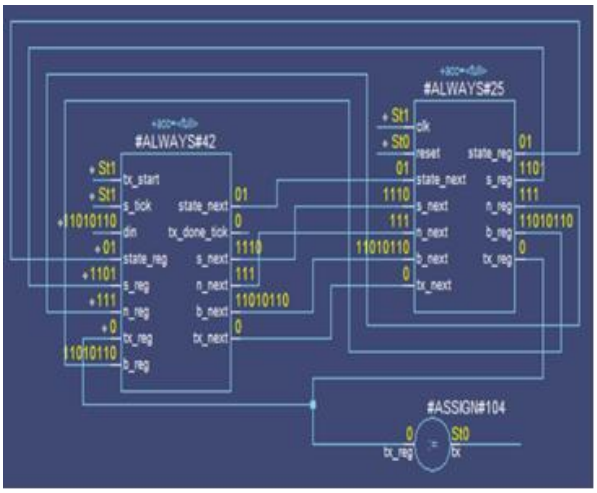
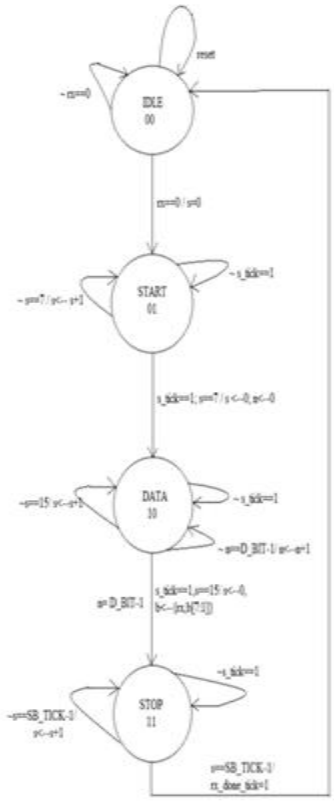


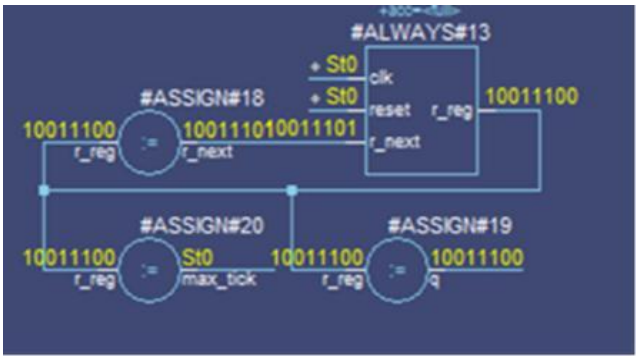
Fig.5. Symbol logic structure of UART Transmitter Logic 2.4 Receiving Sub-System This circuit is implemented to receive the data word using oversampling procedure. The oversampling scheme basically performs the function of a clock signal. Instead of using the rising edge to indicate when the input signal is valid, it utilizes sampling ticks to estimate the middle point of each bit. While the receiver has no information about the exact onset time of the start bit, the estimation can be off by at most 1/16. The ASMD chart using finite state machine (figure 6) and symbol logic (figure 7) structure are shown below. 3. SIMULATION RESULTS All five modules of UART are simulated using Model Sim and synthesised using Xilinx software. 3.1 Baud Rate Generator The waveform of generation of 19,200 baud rate is shown

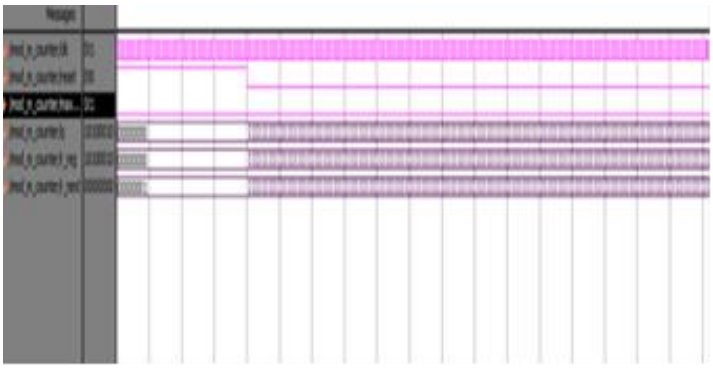
below in the figure 8. Different baud rates can be generated in same way.

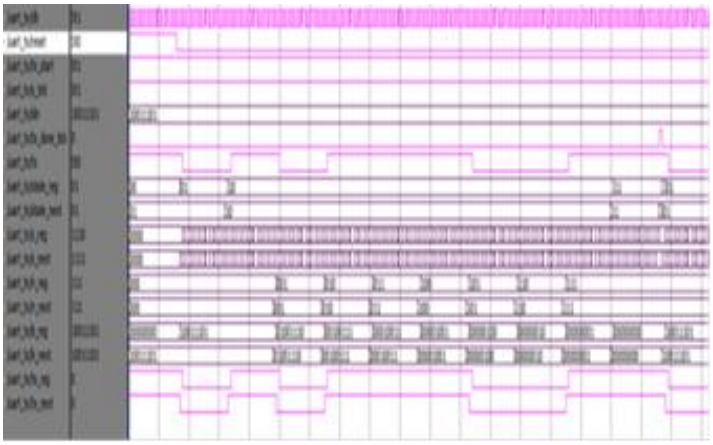
3.2 Transmission Sub system

The baud rate of 19,200 for transmitter sub system is taken for simulation. Waveform for transmitter logic is shown below in the figure 9Fig.6. ASMD chart for receiver logic. Fig.7. Symbol logic structure of UART Receiver Logic. Fig.8. Waveform showing 19,200 baud rate Fig.9. Simulation result of UART Transmitter

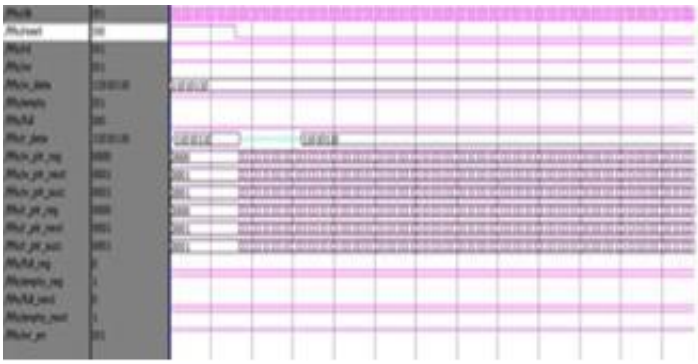








Receiving Sub system During the receiver's simulation process, the system clock frequency is set to 50 MHz, and the baud rate of UARTs is maintained at 19200bps. Simulation result of receiver section is shown below in figure 10 3.3 FIFO at Transmitter and Receiver FIFO is used as interfacing circuit for transmitter and receiving sub system. Simulation results of FIFO are shown



CONCLUSION-UART using FIFO is designed in this paper which is comprised of five modules; baud rate generators, FIFO, transmitters and receivers. The design is successfully simulated using Modelsim software and synthesized using

Xilinx software. The results are depicting the correct behaviour and functionality of the system. The design has great flexibility, high integration as FIFO is used to avoid data loss.

AUTHORS BIOGRAPHY- Sanjeev kumar is currently pursuing M.TECH in Electronics and Communication Engineering at RGPV University, Bhopal and would be postgraduating in summer 2013. He has undertaken various microcontroller and FPGA based projects. His field of interest includes Digital VLSI Design, EDA, RTL simulation and synthesis, Verilog HDL. Prof murli manohar hinnawar is currently my guide in Electronics and Communication Engineering at RGPV University, Bhopal . He possesses excellent knowledge of embedded systems and has interests in fields of VHDL & Verilog programming. He has undertaken various microcontroller and FPGA based projects. His field of interest includes Digital VLSI Design, EDA, RTL simulation and synthesis, Verilog HDL. Prof murli manohar hinnawar is working as Professor with RKDFIST BHOPAL in RGPV University. She received the M.Tech in VLSI Design.i also regardful to Prof Shatrughan Prasad presently worked at M.M College chandi, nalanda (department of physics) magadh university for support in my research area.

REFERENCES-

i. Liu Weifeng, Zhuang Yiqi, Liu Feng, and He Wei (2007), Design of a High performance embedded UART,"Chinese Journal of Electron Devices, Vol. 30,No.4, pp.1275-1278.

ii. Vijay A. Nebhrajan, Asynchronous FIFO Architectures,

iii. Frank Durda, Serial and UART Tutorial.,

iv. M.S. Harvey, Generic UART Manual (1999), Silicon Valley.

v. wikkipedia

google search