FPGA IMPLEMENTATION OF UART CONTROLLER WITH AUTOMATIC BAUD RATE GENERATOR

Miss.Poonam R.Kedia*

Prof.N.N.Mandaogade**

Miss.Sneha R.Gade*

Abstract

Universal Asynchronous Receiver and Transmitter (UART) allows full-duplex communication in serial link and has been widely used in the data communications. It provides a way of serial communication, between two devices but whilst there is a need to provide communication between two devices that are operating at different baud rates, it is intricate to provide communication with an UART. It is a kind of serial communication protocol, mostly used for short-distance, low speed, low-cost data exchange between computer and peripherals. Specific interface chip will cause waste of resources and increased cost. In this, we present a design method of asynchronous FIFO and structure of the controller with automatic baud rate detection. This controller is designed with FIFO circuit block and UART circuit block within FPGA to implement communication in modern complex systems quickly and effectively. Here we use VHDL to implement the UART core functions and integrate them into a Sparten 3E FPGA chip to achieve compact, stable and reliable data transmission. In the result and simulation part, we will focus on baud rate generation at different frequencies and check the receive data with error free. The Baud Rate Generator is incorporated into the UART design. This frequency divider will automatically adjust according to requirements.

Keywords: *UART, Baud rate generator, Shift Register, FIFO*

* Student of M.E Electronics & Telecommunication, Department of Electronics & Telecommunication Engg, G.H.Raisoni College of Engineering, Amravati –India

** Faculty of Electronics & Telecommunication, Department of Electronics & Telecommunication Engg., G.H.Raisoni College of Engineering, Amravati-India

1. Introduction

Asynchronous serial communication has advantages of high reliability, less transmission line and long transmission distance, therefore is widely used to exchange data between a computer and external devices.

Asynchronous serial communication is implemented by UART. It provides full-duplex communication in serial link; this has been widely used in the data communications. UART includes a transmitter and a receiver. Transmitter controls transmission by taking a data word in parallel format and directing the UART to transmit it in a serially. Likewise, the Receiver must detect transmission, receive the data in serially, and store the data word in a parallel format. The conversion of serial to parallel data is handled by UART. Serial communication reduces the distortion of a signal; therefore data transfer is possible between two systems separated by great distance. The UART serial module is divided into three sub-modules: the baud rate generator, receiver module and transmitter module. The baud rate generator is used to produce a local clock signal. In data transmission through the UART, once the baud-rate has been established, both the transmitter and the receiver's internal clock are set to the same frequency. TXD is the transmit side, i.e. the output of the UART; RXD is the receiver, i.e. the input of the UART. The UART receiver module is used to receive the serial signals at RXD and convert them into parallel data. The UART transmit module converts the data bytes into serial bits according to the frame format and transmits those bits through TXD.

2. Literature review

FANG Yi-Yuan CHEN XUE- Jun has presented a paper on "Design and simulation Of UART serial communication Module Based on VHDL". In this paper they presented that the UART is the microchip with programming that controls a computer's interface to its peripherals. It is the most widely used serial data communication circuit ever. The whole process of serial transmission is based upon the principle of the shift register. There are two primary forms of serial transmission that are Synchronous & Asynchronous. In Synchronous serial communication requires that the sender and receiver should work on the same clock with one another. Asynchronous transmission allows data to be transmitted without sending a clock signal to the receiver. This design uses VHDL as the design language to achieve the modules of the UART.

The results are reliable & stable. The design has high integration, great flexibility with some reference value [1].

Shouqian Yu, Lili Yi, Weihai Chen, Zhao Jin Wen presented a paper on "Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA". In this paper they have presented that in several systems such as high data collection system, a high speed control system based on PCI and multi-DSP signal processing system, FIFO is used for complete communication between high speed device and low speed device or to complete communication between the same sub controllers. FIFO is the most important part of these systems and it works as a bridge between different devices. At the same, in our controller, asynchronous FIFO based on FPGA is also the most important part. So the features and capabilities of the asynchronous FIFO determine the features of our controller. The FIFO can be used to complete communication in parallel or serial port [2].

Bhavana Mahure and Rahul Tanwar have presented a paper on "UART with automatic baud rate generator and frequency divider". In this paper they have presented that the most commonly used numbers of data bits of a serial connection are eight, which corresponds to a byte. When a regular ASCII code is used in communication, only seven LSBs are used and the MSB is 0. If the UART is configured as 8 data bits, 1 stop bit, and no parity bit, the received word is in the form of 0-dddd-ddd-0-1, in which d is a data bit and can be 0 or 1. Assume that the UART configuration is 8 data bits, 1 stop bit, and no parity bit, and the baud rate can be 4800, 9600, or 19,200 baud [3].

The circuit produces Frequency Division as it now divides the input frequency by a factor of two. Frequency Divider is dividing the frequency according to system requirement. So we can use this UART with frequency divider, no need to attach another device in that system to divide the frequency.

Nurul Fatihah Jusoh, Azlina Ibrahim, Muhamad Adib Haron and Fuziah Sulaiman presented a paper on "An FPGA Implementation of Shift Converter Block Technique on FIFO for UART" the paper represents the implementation of the bidirectional shift converter technique with FIFO circuit block and UART circuit block through FPGA device using Verilog HDL language to be applied in embedded system converter RS232 to USB (Universal serial bus) [4].

Nennie Farina Mahat presented a paper on " Design of a 9-bit UART Module Based on Verilog HDL." In this paper, a modified UART design is proposed with automatic address indicated, which is called 9-bit UART [5]

"Platform-Independent Customizable UART Soft-Core" Biplab Roy. In this paper, we propose a technique for software implementation of a UART using shift register with the goal of getting a customizable UART-core which can be used as a module in implementing a bigger system irrespective of one's choice of the implementation platform [6].

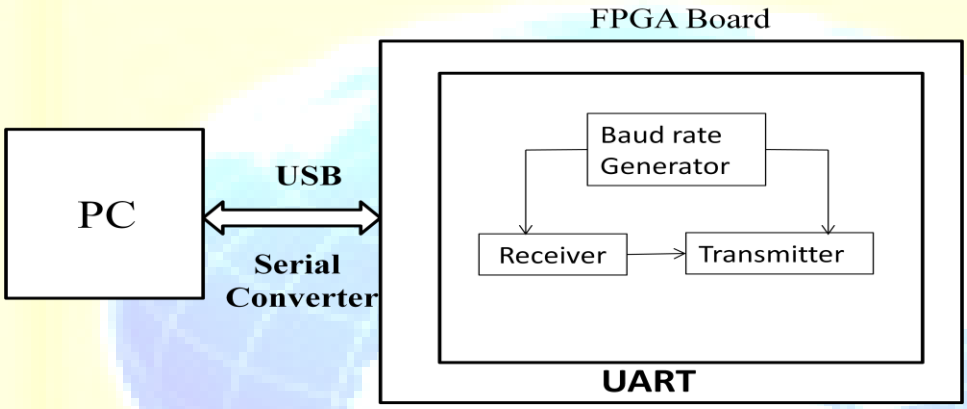

Fig.1. Block diagram of interfacing of PC with FPGA Board.

3. Methodology

There are two primary forms of serial transmission: Synchronous and Asynchronous. They are:

1. UART Universal Asynchronous Receiver/Transmitter

2.USART Universal Synchronous-Asynchronous Receiver

Asynchronous serial communication is usually implemented by UART. The Universal Asynchronous Receiver/Transmitter (UART) controller is the key component of the serial communications of a computer.

In Serial communication, there occurs, reduction of the distortion of a signal; therefore it is possible to make data transfer between two systems at great distance It enables to control the conversion between serial and parallel data. The UART serial communication module is divided into three sub-modules: the baud rate generator, receiver module and transmitter module.

The transmitter is a distinctive shift register that loads data in parallel, then at a specific rate it shifts out bit by bit. On the other hand The receiver shifts the data bit by bit and then rearrange the data. UART transmitter controls communication by getting a data word in parallel format & directing the Universal Asynchronous Receiver Transmitter to transmit it in a serial format.

Receiver Module

During the UART reception, the serial data and the receiving clock are asynchronous, so it is very important to correctly determine the start bit of a frame data. The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of a data frame. The start bit is identified by detecting RXD level changes from high to low. The serial receiver module includes receiving, serial and parallel transform, and receive caching, etc . The function of the receiver module is that it will store the tx_out i.e. the output of the transmitter which is of single bit into the intermediate register with the start bit as the least significant bit and collectively provides the serial data of 8. When the load signal is high it will get the start bit from the transmitter which assures that the original data is now being send by the transmitter. Once the shift signal is becomes high with no load signal, the data coming from the transmitter gets shifted into the intermediate register of the receiver and provides the 8 bit serial data which we have given as an input to the transmitter.

Trasmit Module

The function of transmit module is to convert the sending 8-bit parallel data into serial data, adds start bit at the head of the data as well as the parity and stop bits at the end of the data. When the UART transmit module is reset by the reset signal, the transmit module immediately enters the ready state to send. The function of the transmitter module is to convert the 8 bit serial data into the single bit data. In this module, when our load signal is high the data_in is stored into the holding register. The data in the holding register is shifted to the intermediate register with the start bit of zero and this intermediate register is of 9 bits. Once the shift signal is high the least significant bit of the intermediate register i.e. the start bit comes at the output of the transmitter and served as the input to the receiver. When the entire data has been sent, the transmitter provides a parity bit which is served as the input to the receiver.

5. Implementation and Results

The design of automatic baud rate generator is coded using VHDL language. Simulation and synthesis of UART is done on ModelSim software and Xilinx ISE software respectively. We implemented the UART controller with automatic baud rate generator on a Xilinx XC3S500E Spartan-3E FPGA kit.

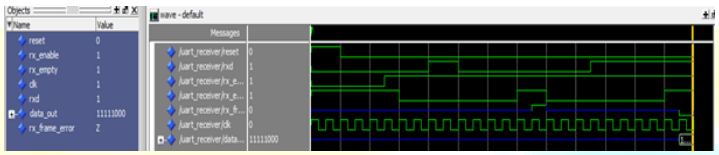

Fig2 : Output of Receiver

If the system has a signal reset active then reception is high impedance while other status signal i.e rx_enable is '1' and rx_empty will be '1'. When system has reset =0 and reception signal is enable (rx) then data reception is started. For every rising edge of clock pulse individual data bit is received by the receiver module as UART protocol states first bit is 0, then remaining 8 bit are data bit last bit is 1 then successful reception of data has been occurred. If start bit is not '0' then all remaining data has been discarded by receiver module by indicating rx_frame error='1'.

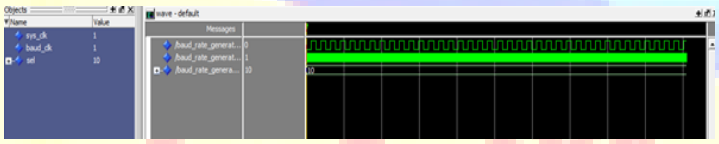

Fig3: Output of Baud Rate Generator

Fig shows the simulated waveform for the baud rate 19200. First we set the system clock at '1'. We have four combinations for select lines that are 00,01,10,11. From these four options we can obtain four different baud rates and default baud rate generated is 9600. In above waveform we have set the system clock at "10", so 19200 baud rate is generated.

+

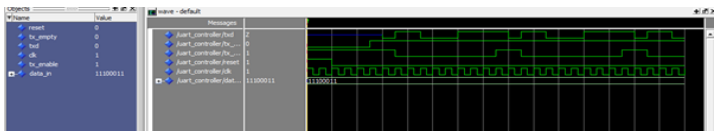

Fig4: Output of Transmitter

If the system has a signal reset active then reception is high impedance then data transmission has been occurred, while other status signal i.e rx_enable is '0 ' and rx_empty is '1'.

When system reset is '0', rx enable is '1' and rx empty is '0'then data bit has been given out at every rising edge of the clock at txd line. First starting bit has been send as '0' on txd line then parallel data has been send at every rising edge of cock. After completion of successive & data bits , stop bit has been send as' 1'.

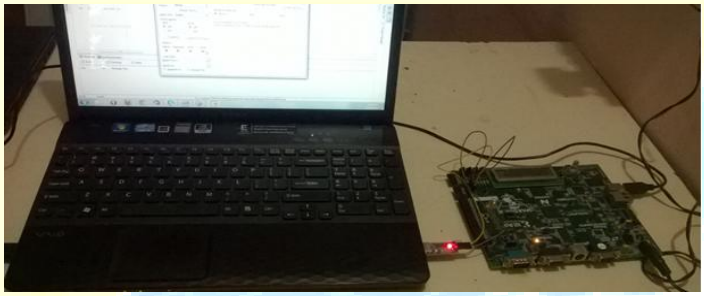

Fig Peripheral connected to FPGA Board

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	89	9,312	1%	
Number of 4 input LUTs	195	9,312	2%	
Number of occupied Slices	134	4,656	2%	
Number of Slices containing only related logic	134	134	100%	
Number of Slices containing unrelated logic	0	134	0%	
Total Number of 4 input LUTs	259	9,312	2%	
Number used as logic	194			
Number used as a route-thru	64			
Number used as Shift registers	1			
Number of bonded IOBs	13	232	5%	

Fig. 4 Synthesis result of UART controller.

Our results for FPGA Altera's Cyclone II FPGA: EP2C5F256C6

our ref clock : 50MHz

	Baud Rate Generator	Transmitter	Receiver
Logic Elements	48	24	39
Registers	33	14	31
I/O pins	2	12	13

Conclusion

Efficient implementation of UART with automatic baud rate generator of Sparten 3E board is presented in this paper. High throughput is achieved in this design In this we had added the concept of automatic baud rate detection, so when the transmitter changes the baud rate the receiver can adjust automatically and it reduces the delay for the reception of data than the fixed baud rate. Especially in the field of electronic design technology has recently become widely used, this design shows great significance.

References

1. FANG Yi-Yuan CHEN XUE- jun , "Design and simulation Of UART serial communication Module Based on VHDL",2011.

2. Shouqian Yu ,Lili Yi, Weihai Chen, Zhao jin Wen," Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA", 2007

3. Bhavna Mahure and Rahul Tanwar (2012) "UART with Automatic Baud Rate Generator and Frequency Divider". Journal of Infor-mation Systems and Communication, ISSN: 0976-8742 & E-ISSN: 0976-8750, Volume 3, Issue 1, pp.- 265-268.

4. Fatihah Jusoh, Azlina Ibrahim, Muhamad Adib Haron and Fuziah Sulaiman "FPGA Implementation of Shift Converter Block Technique on FIFO for UART",2011

Nennie Farina Mahat, " Design of a 9-bit UART Module Based on Verilog HDL.",2012

5. Bipla Roy,"Platform-Independent Customizable UART Soft-Core",2012

6. Jan Henning Mueller, Mojdeh Hamzavi Nejad Moghaddam, Bastian Mohr, Sebastian Strache,Ralf Wunderlich and Stefan Heinen, "An Adaptable UART Based Configuration and Read-out Interface for IC Prototypes",2012

7. Mohd Yamani Idna, ldris Mashkuri Yaacob, "A VHDL Implementation of BIST Technique in UART Design",2003

8. J. Norhuzaimin and H.H Maimun, "The Design of High Speed UART",2005

9. Shihua Tong, "Design and Simulation Analysis of UART IP Core",2011

10. Naresh Patel, Vatsalkumar Patel, Vikaskumar Patel, "VHDL Implementation of UART with Status Register"

11. Dr. Garima Bandhawarkar Wakhle," Synthesis and Implementation of UART using VHDL Codes",2012

12. Ritesh Kumar Agrawal, Vivek Ranjan Mishra," The Design of High Speed UART",2013