International Journal of Research in Engineering, Science and Management
Volume-2, Issue-2, February-2019
www.ijresm.com | ISSN (Online): 2581-5792

Implementation of UART Technique in FPGA

Yogivani Malleboina[1], Manisha Erigineni[2], G. Krishnaveni[3]
[1,2,3] *Department of ECE, Bapatla Women's Engineering College, Bapatla, India*

***Abstract*: Asynchronous serial communication is usually implemented by Universal Asynchronous Receiver Transmitter (UART), mostly used for short distance, low speed, low cost data exchange between processor and peripherals. UART allows full duplex serial communication link, and is used in data communication and control system. There is a need for realizing the UART function in a single or a very few chips. Further, design systems without full testability are open to the increased possibility of product failures and missed market opportunities. Also, there is a need to ensure the data transfer is error proof. Status register to UART, to overcome the above two constraints of testability and data integrity. The 8-bit UART with status register is coded in VHDL and synthesized and simulated using Xilinx XST and MODELSIM version 14.2 and realized on FPGA. The results indicate that this model eliminates the need for higher end, expensive testers and thereby it can reduce the development time and cost.**

***Keywords*: UART Technique, FPGA**

1. Introduction

Asynchronous serial Communication has advantages of less transmission lines, high reliability and long transmission distance. UART allows full–duplex communication in serial link, thus has been widely used in the data communications and control system. It is widely used in data exchange between Processor and Peripherals. UART converts data from parallel to serial at transmitter with some extra overhead bits using shift register and vice versa at receiver. To the processor the UART appears as an 8-bit read/write parallel port [1], [2]. Basic UART communication needs only two signal lines (receive, transmit) to complete full-duplex data communication [2], UART includes three modules namely, the baud rate generator, receiver and transmitter. The baud rate generator is used to produce a local clock signal which is much higher than the baud rate to control the UART receive and transmit; 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 bytes into serial bits according to the basic frame format and transmits those bits through TXD.

In actual applications, usually only a few key features of UART are needed. Specific interface chip will cause waste of resources and increased cost [2]. Particularly in the field of electronic design, SOC technology is recently becoming increasingly mature. This situation results in the requirement of realizing the whole system function in a single or a very few chips. Integration of only core functions into a FPGA chip to achieve compact, stable and reliable data transmission avoids waste of resources and decrease cost.

Manufacturing processes are extremely complex, inducing manufacturers to consider testability as a requirement to assure the reliability and the functionality of each of their designed circuits [3],[7]. Testing of integrated circuits (ICs) is important to ensure a high level of quality in product functionality in both commercially and privately produced products. In the modern system-on a-chip (SOC) design, many cores are integrated into a single chip. Some of them are embedded, and cannot be accessed directly from the outside of the chip. Such SOC designs make the test of these embedded cores a great challenge. As ICs grow in gate counts, it is no longer true that most gate nodes are directly accessible by one of the pins on the package. This makes testing of internal nodes more difficult as they could neither no longer be easily controlled by the signal from an input pin (controllability) nor easily observed at an output pin (observe ability). Hence Internal Diagnostic Capabilities are to be introduced to test the embedded cores. In this paper, internal diagnostic capabilities are build in UART by the introduction and error simulation of data at receiver for any data corruption and thereby setting status flags [1]. The UART with status register is coded in VHDL and simulated using Xilinx tool MODELSIM. The complete implementation and validation is done on Spartan 3E FPGA

A. UART transmitter

The transmitter accepts parallel data from peripherals/processor, makes the frame of the data and transmits the data in serial form on the transmitter output (TOUT) terminal fig. 1. The baud rate generator output will be the clock for UART transmitter

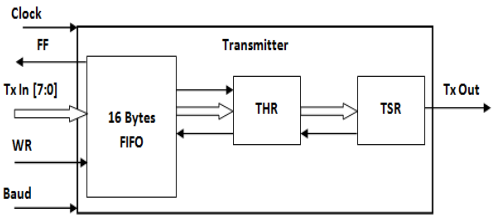

Fig. 1. UART Transmitter

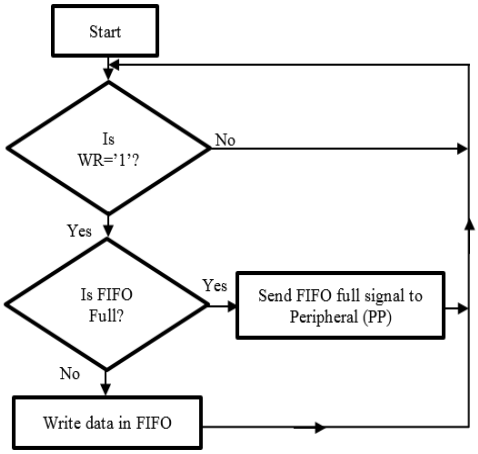

Fig. 2. Fifo

Data is loaded from the parallel inputs TXINO-TXIN7 into the transmitter FIFO by applying logic high on the WR (Write) input. FIFO is 16-byte register. If FIFO is full, it sends FIFO Full(FF) signal to peripheral as shown in fig. 2.

When FIFO contains some data, it will send the signal to Transmitter Hold Register (THR), which is an 8-bit register. At the 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 an 11-bit register in which framing process occurs. In frame, start bit, parity bit and one stop bit will be added as shown in figure 4. Now data is transmitted from TSR to TXOUT serially.

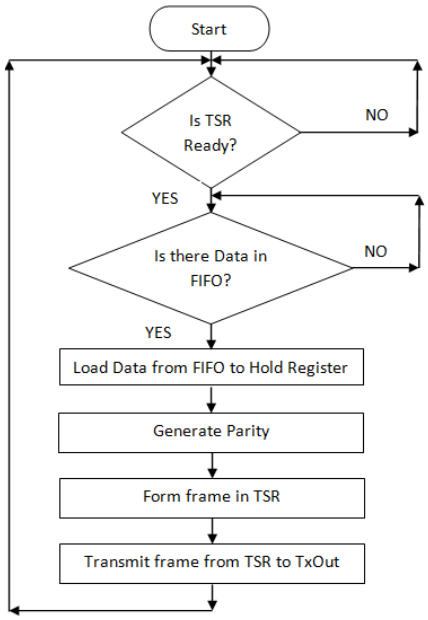

Fig. 3. Transmitter flow chart-FIFO to TXOUT

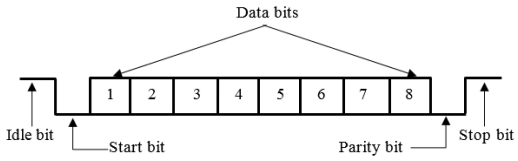

Fig. 4. UART frame format

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 law) is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the peripheral receiver into synchronization with the clock in the transmitter. After the Start Bit, the individual 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 wire at approximately halfway through the period assigned to each bit to determine if the bit is a 1 or a 0. When the entire data word has been sent, 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 previous word has been sent. Asynchronous data are “self-synchronizing” If there are no data to transmit, the transmission line is held idle

B. 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. 5), initially the logic line (Rxln) is high.

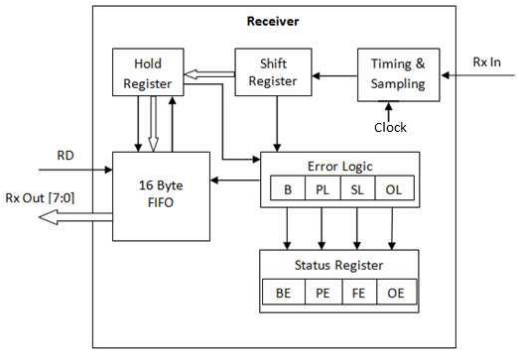

Fig. 5. UART Receiver

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. Fig.8 shows the receiver.

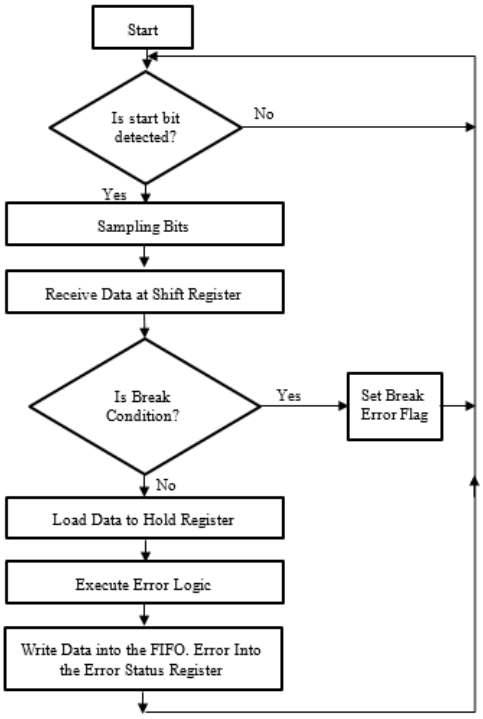

Fig. 6. Receiver flowchart (Input to FIFO)

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 a 7-bit register. The remaining bits in the RSR are used by the error logic block. Then, if receiver FIFO. When RD signal is asserted the data is available in parallel form on the RXOUTO-RXOUT7 Pins. 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 bits is set. Reading of data from receiver is explained by means of flowchart in fig. 6.

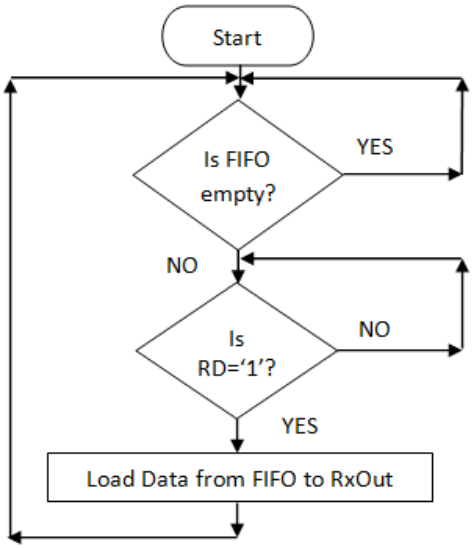

Fig. 7. Receiver flowchart (FIFO to Output)

Simulation Results of Transmitter: The fig. 8 shows the serial transmission of data. Data transmitted is "10101010". This 8-bit data is loaded to transmit shift register and start, stop & parity bits are added to form the frame inside TSR and sent to TXD. When the reset is 0 and transmit is 1, the transmitter starts transmitting the data. i.e. the data starts shifting out from the transmitter shift register. Since the desired baud rate is 9600bps, the bits are shifted out on TXD line at the interval of 50MHZ/9600=5208 clock cycles. Similarly all the bits are sent. The serial transmission is observed at TXD pin along with frame format (1 logic low start bit, 8-bit data (LSB to MSB), parity bit and finally logical high stop bit). Data Simulation Results of Receiver The UART receiver converts the serial.

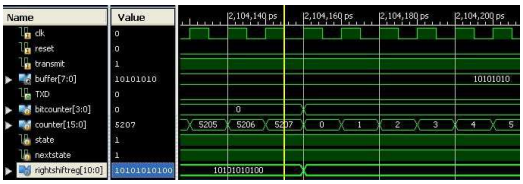

Fig. 8. Simulation Results of transmitter

The UART receiver converts the serial data into parallel form and makes it available at RxData[7:0]. The serial data is received at RXD pin. 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, RxData. The 8-bit data simulated is "11111111". Further received data will be stored in the remaining FIFO locations. Fig. 9 shows the reception of serial data.

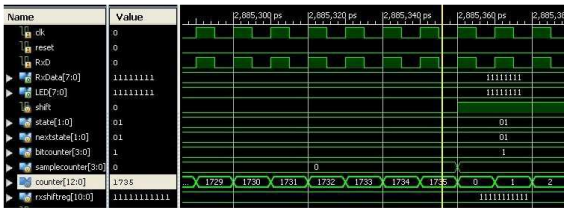

Fig. 9. Simulations Results of Receiver

2. Conclusion

The architecture of UART that support. 8-bit data baud rate for serial transmission of data with the addition of status register for detecting errors in data transfer Working of UART has been tested using Xilinx ISE Simulator, which is implemented on FPGA. With error checking status register, we can detect the different types of errors occurred during communication and hence correct them.

References

[1] Naresh Vatsalkumar and Vikaskumartatel "VHDL" Implementation of UART with statused registers", in the procceedings of international conference on communication system and Technologies, IEEE computer society,11-13th, May 2012.

[2] R. Gallo, M. Delvi, W. Elmenraich, and A. Steingir, "Rrevision and verification of an enchanced UART".