IMPLEMENTATION OF STATUS REGISTER WITH UART BY VHDL

Ku. Neeta Choubey
PG Student-VLSI Design
Oriental University, Indore, M.P.

MR. H.R. Singh
Principal Of Oriental University, Indore

Abstract

In parallel communication the cost as well as complexity of the system increases due to simultaneous transmission of data bits on multiple wires. Serial communication alleviates this drawback and emerges as effective candidate in many applications for long distance communication as it reduces the signal distortion because of its simple structure. This paper focuses on the VHDL implementation of UART with status register which supports asynchronous serial communication. The paper presents the architecture of UART which indicates, during reception of data, parity error, framing error, overrun error and break error using status register. The whole design is functionally verified using Xilinx ISE Simulator.

Keywords- Universal Asynchronous Receiver Transmitter; status register; VHDL implementation; ISE simulator; asynchronous serial communication

I. INTRODUCTION

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. UARTs are used for asynchronous serial data communication by converting data from parallel to serial at transmitter with some extra overhead bits using shift register and vice versa at receiver. It is generally connected between a processor and a peripheral, to the processor the UART appears as an 8-bit read/write parallel port.

The UART implemented with VHDL language can be integrated into the FPGA to achieve compact, stable and reliable data transmission[5]. Various designs are found in literatures for UART as different systems have different requirements and attributes which require data communication between its functional units. In recent years the researchers have proposed various UART designs like automatic baud rate synchronizing capability[2], predictable timing behavior to allow the integration of nodes with imprecise clocks in time-triggered real-time systems[1], recursive running sum filter to remove noisy samples[4], integration of only core functions into a FPGA chip to achieve compact, stable and reliable data transmission to avoid waste of resources and decrease cost[5], programmable logic to enable interfacing between asynchronous communications protocols and DSP having
synchronous serial ports[6].

In this paper, we present UART which includes three modules which are the baud rate generator, receiver and transmitter. The proposed design of UART satisfies the

system requirements of high integration, stabilization, low bit error rate, and low cost. It also supports configurable baud rate generator and variable data length from 5-8 bits per frame.

The rest of paper is as follow: The advantages of VHSIC hardware description Language (VHDL) are highlighted in section II. Section-III describes the proposed architecture along with algorithms. Simulation results of each module using Xilinx ISE Simulator are illustrated in section IV.

Finally the paper is concluded in section V.

II. VHDL IMPLEMENTATION

The detail design of systems at the gate and flip-flop level has become time consuming as the integrated circuit technology has become very complex [7]. In recent years this fact has motivated usage of hardware description language in the design process of digital system. VHDL is used to describe and simulate the operation of variety of digital system which is ranging in complexity from a few gates to an interconnection of many complex integrated circuits. Advantages of VHDL implementation includes minimum cost and time, better design, faster time to market and increased flexibility [7].

III. PROPOSED UART ARCHITECTURE

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 frame format is shown in fig.1.

Figure 1. Frame Format For UART [5]

The proposed design of UART, shown in Fig. 2, has LCR, Baud Rate Generator (BRG), Transmitter and Receiver as its functional units. All these blocks are explained in brief as course of rest of this section.

Figure 2. UART Architecture

A. Line Control Register (LCR)

The line control register (LCR) is a byte register. It is used for precise 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, format of which is shown in Fig. 4 and Table 1 indicates exact voltage levels of selection lines for selecting different baud rates, odd/even parity and data word length.

Figure 3. LCR Format [5]

B. Baud Rate Genesrator

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. For the selective baud rate the divisor can be obtained by dividing the system clock by desired baud rate as shown in Table 2 which shows divisors for various baud rates using 10 MHz system clock.

C. Transmitter

The transmitter section accepts parallel data, makes the frame of the data and transmits the data in serial form on the Transmitter Output (TXOUT) terminal. Data is loaded from the inputs TXIN0-TXIN7 into the Transmitter FIFO by applying logic high on the WR (Write) input. 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. In frame start bit, stop bit and parity bit will be added. Now data is transmitted from TSR to TXOUT serially. Fig. 4 and fig. 5 shows the entire working in form of flowcharts.

Figure 4. Transmitter flowchart (Input to FIFO)

D. Receiver

The transmitted data from the TXOUT pin 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.

Initially the logic line is high 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. Now 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 p arallel form on the RXOUT0-RXOUT7 pins.

Figure 5. Transmitter flowchart (FIFO to Output)

The error logic block handles 4 types of errors: Parity error, Frame error, Overrun error, break error. If the received parity does not match with the parity generated from data bits PL 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 SL bit is set. If the receiver FIFO is full and other data arrives at the RHR overrun error occurs and OL 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 BL bit is set. Working of receiver is shown in terms of flowcharts in fig. 6 and fig. 7.

Figure 6. : Receiver flowchart (Input to FIFO)

Figure 7. Receiver flowchart (FIFO to Output)

IV. SIMULATION

A. Hardware Utilization

The designed UART is synthesized using Xilinx project navigator for device XC3S400. The resource utilization is detailed in Table 3.

TABLE III. RESOURCE UTILIZATION

B. RTL Schematic of top level entity

The RTL Schematic generated by Xilinx for UART is shown in fig.8.

Figure8.Block diagram of the synthesized UART

C. Simulation result of transmitter.

The fig. 9 shows the serial transmission of data. Baud rate selected here is 5.81 MHz using 500 MHz system clock. Data word length is set as 8-bits and odd parity is selected. Data to be transmitted are ABH, 76H, 41H and they are applied at tx_in[7:0]. The serial transmission is observed at ser_out pin along with frame format (1 logical low start bit, 8-bit data (LSB to MSB), odd parity bit and finally 2 logical high stop bits).

Figure 9. Transmitter Simulation

D. Simulation Result of Receiver.

The fig. 10 shows the reception of serial data. Serial data is received through rx_in pin. Once the complete frame of a data is received, framing bits are discarded and data is converted into parallel form internally and it is stored at internal FIFO register. By asserting rdy_read active high signal the parallel data is made available at rx_data[7:0].

Figure 10. Receiver Simulation

E. Simulation Result of combined UART.

Fig. 11 shows simulation of UART as a single entity. It shows independent functioning of transmitter and receiver sections. While the transmitter section of UART converts the parallel data into serial form before its transmission through tx_out pin at the same time receiver section can receive the data serially through rx_in pin, converts it into parallel form and makes it available at rx_out[7:0].

Figure 11. Simulation Result of UART

V. CONCLUSION

This paper describes the architecture of UART that support various data word length, parity selection and different baud rates for serial transmission of data. Working principle of this UART has been tested using ISE simulator, which can be implemented on FPGA. Additionally we can detect the different types of errors occurred during communication and hence correct them.

REFERENCES

Elmenreich, W.; Delvai, M.; , "Time-triggered communication with UARTs," Factory Communication Systems, 2002. 4th IEEE Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

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..

Norhuzaimin, J.; Maimun, H.H.; , "The design of high speed UART," Applied Electromagnetics, 2005. APACE 2005. Asia-Pacific Conference on , vol., no., pp.5 pp., 20-21 Dec. 2005.

Himanshu Patel; Sanjay Trivedi; R. Neelkanthan; V. R. Gujraty; , "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.

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

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. Conference on Convergent Technologies for Asia-Pacific Region , vol.4, no., pp. 1450- 1454 Vol.4, 15-17 Oct. 2003.

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.

Prof. Rami Abielmona, "Project: UART Design", for CEG 3150: Digital Systems-II, Fall 2004, November 24, 2004.

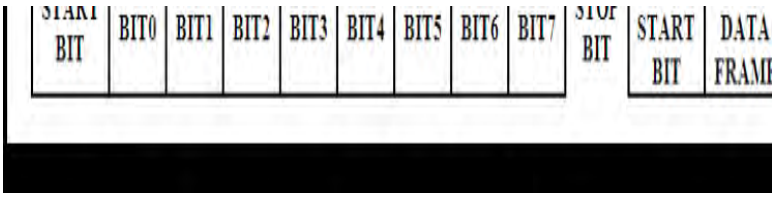

Figure 1. Frame Format For UART [5]

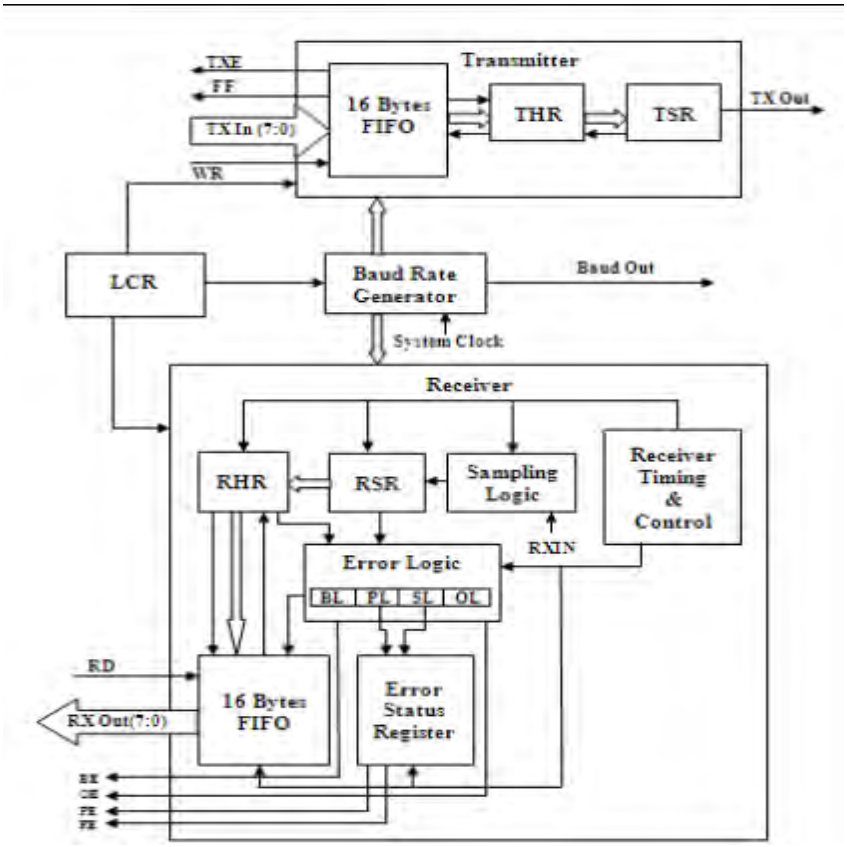

Figure 2. UART Architecture

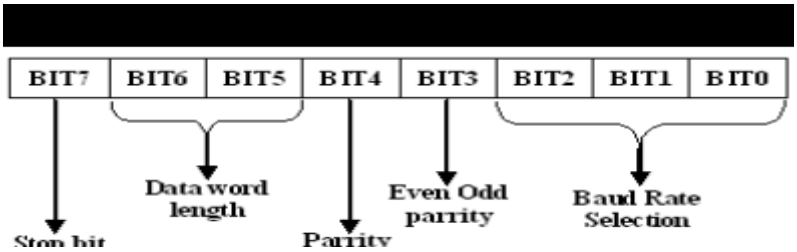

Figure 3. LCR Format [5]

TABLE II DIVISORS FOR VARIOUS BAUD RATES

Ba ud rate	Freq-10MHz	
	Divisor	%Error
600	8333	0.00
1200	4165	0.00
2400	2082	0.00
4800	1040	0.00
9600	520	0.00
19200	259	0.10
38400	129	0.26
57600	86	0.17

TABLE I. LCR BIT DESCRIPTION [9]

Bit	value			Description
0,1,2	Bit2	Bit1	Bit0	BR Sel.
	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 bit

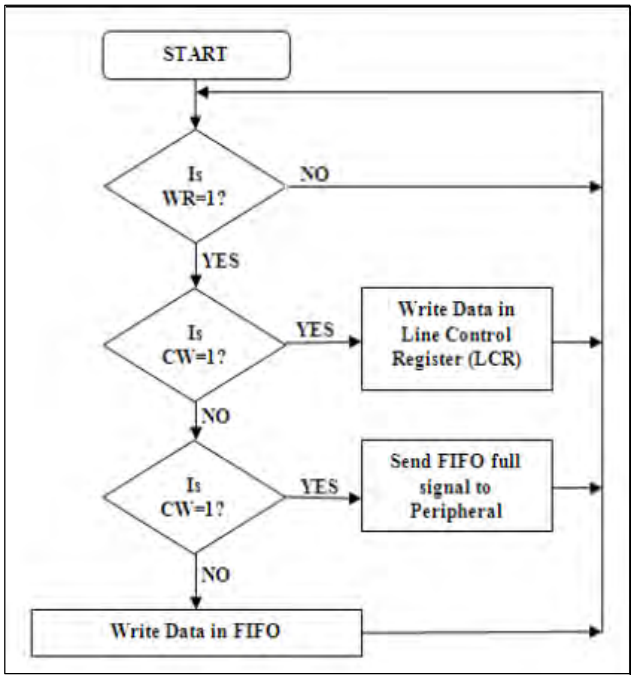

Figure 4. Transmitter flowchart (Input to FIFO)

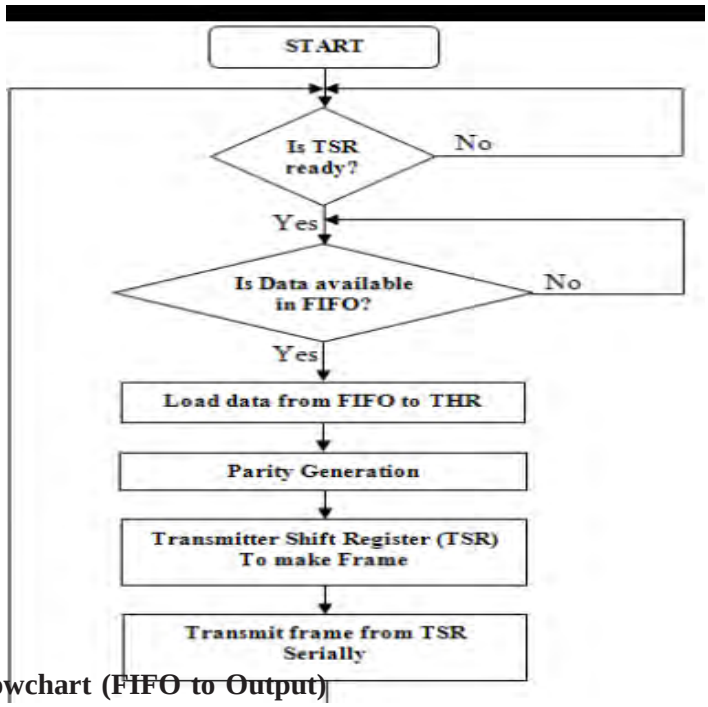

Figure 5. Transmitter flowchart (FIFO to Output)

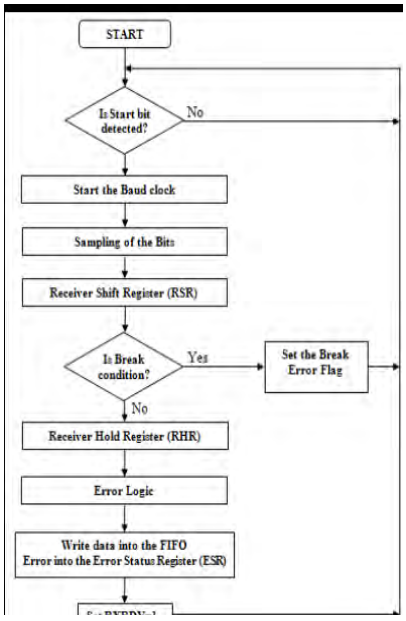

Figure 6. : Receiver flowchart (Input to FIFO)

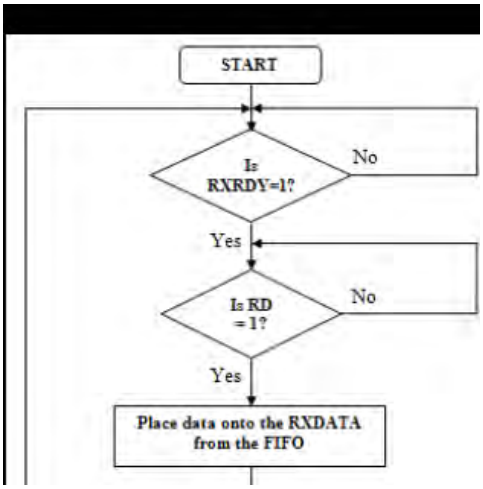

Figure 7. Receiver flowchart (FIFO to Output)

TABLE III. RESOURCE UTILIZATION

Device Utilization Summary (Estimated Values)			
Logic Utilization	***Used***	***Used Available***	***% Utilization***
No. of Slices	**453**	**3584**	**12**
No. of slice FFs	**467**	**7168**	**6**
No. of 4 i/p LUTs	**711**	**7168**	**9**
No. of IOBs	**68**	**141**	**48**
No. of GCLKs	**6**	**8**	**75**

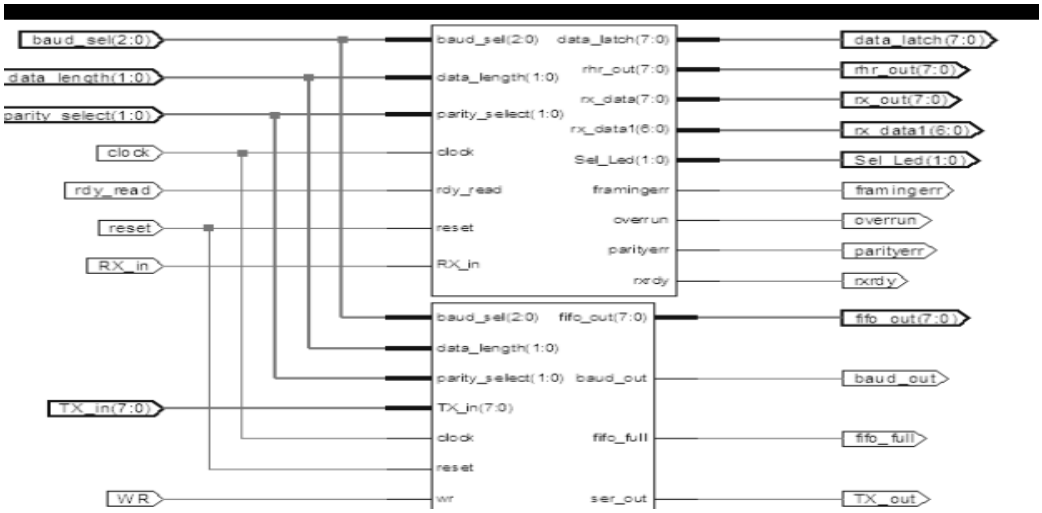

Figure 8. Block diagram of the synthesized UART

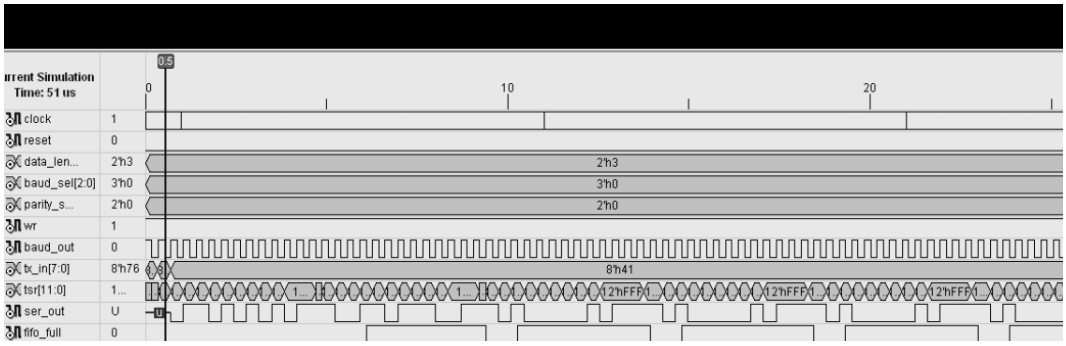

Figure 9. Transmitter Simulation

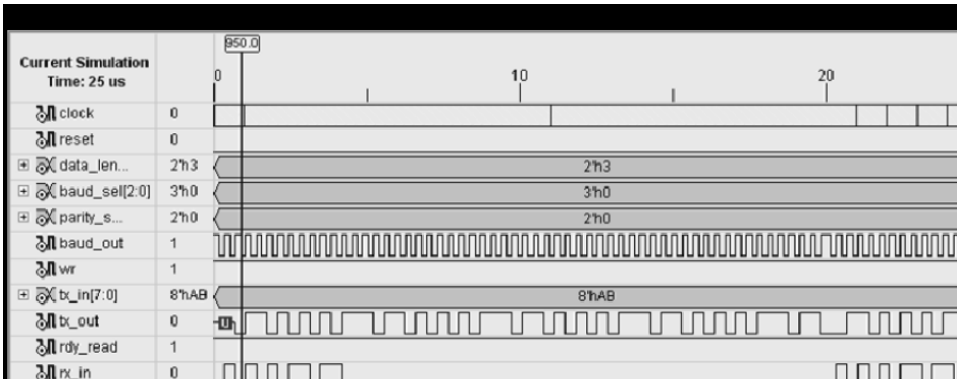

Figure 10. Receiver Simulation

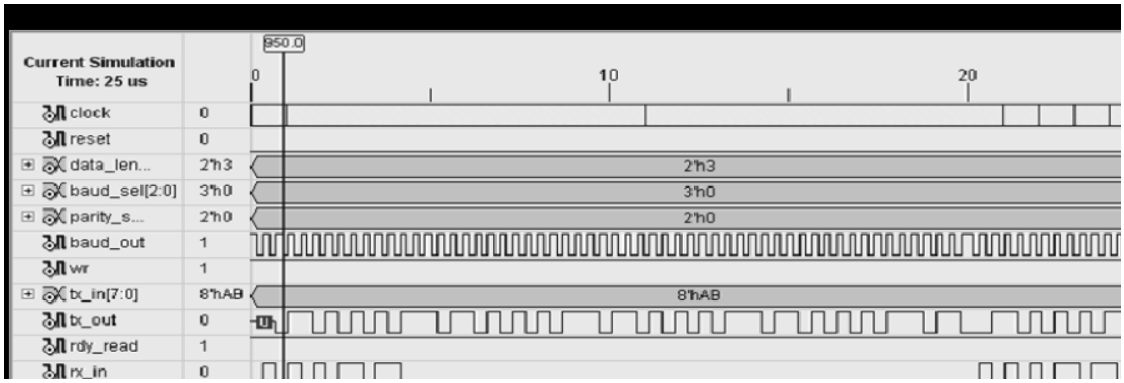

Figure 11. Simulation Result of UART