DESIGN AND SIMULATION OF UART PROTOCOL BASED ON VERILOG

[1]B.JEEVAN & [2]M.NEERAJA

[1,2]Dept of E.I.E, KITS,Warangal, Andhra Pradesh, India

Abstract—UART (Universal Asynchronous receiver Transmitter) is a kind of serial communication protocol; mostly used for short-distance, low speed, low-cost data exchange between computer and peripheral. The UART takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. The UART allows the devices to communicate without the need to be synchronized. UART includes three kernel modules which are generator, receiver and transmitter .The UART implemented with VERILOG language can be integrated into FPGA. The simulation results with Xilinx are completely consistent with the UART protocol.

Keywords- UART, asynchronous serial communication, Xilinx

I. INTRODUCTION

Asynchronous serial communication has advantages of less transmission line, high reliability, and long transmission distance, therefore is widely used in data exchange between computer and peripherals. Specific interface chip will cause waste of resources and increased cost. This situation results in the requirement of realizing the whole system function in a single or a very few chips. Integrate the similar function module into FPGA. This paper uses VERILOG to implement the UART core functions and integrate them into a FPGA chip .

Basic UART communication needs only two signal lines(RXD,TXD) to complete full-duplex data communication.TXD is the transmit side, the output of UART; RXD is the receiver, the input of UART. Logic 1(high) logic 0 (low) are two basic features of UART. When the transmitter is idle, the data line is in the high logic state otherwise in low logic state. When a word is given to the UART for asynchronous transmission, a bit called the “Start Bit" is added to the beginning of each word that is to be transmitted. The start bit is used to alert the receiver that a word of data is about to sent.

After the Star 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 same amount of time, and the receiver looks at approximately halfway through the period assigned to each bit to determine if the bit is 1 or 0.

The sender does not know when the receiver has “looked” at the value of the bit. The sender only knows when the clock says to begin transmitting the next bit of the word. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking (both the sender and receiver must agree on whether a Parity Bit may use or not) and when the receiver has received all of the bits in the data word then receiver looks for stop bit.

If the Stop Bit does not appear, the UART considers the entire word be garbled and will report Framing Error. When the another word is ready for transmission, the start bit for the new word can be sent as soon as the stop bit for the previous word has been sent. The UART frame format is show in fig. 1.

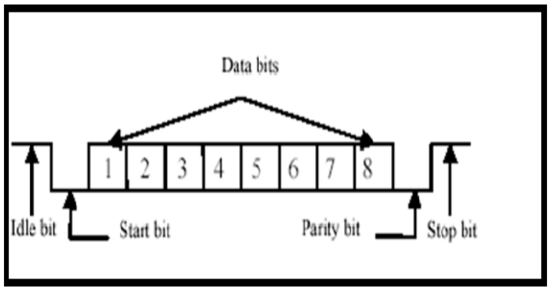

Fig. 1 UART Frame Format

II. IMPLEMENTATION OF UART

In this paper Top to Down design method is used. The UART module is divided into three sub-modules: the transmitter module, receiver module and baud rate generator, shown in fig. 3

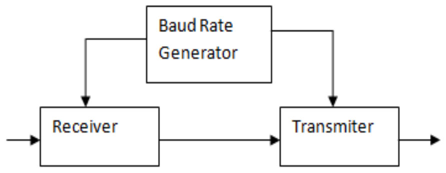

Fig. 3 UART Module

A.BAUD RATE GENERATOR

THE Baud Rate Generator is used to produce used to produce a local clock signal which is much higher than the baud rate to control the UART receiver and transmit. The baud rate generator is actually a frequency divider. The frequency factor is calculated according to the given system clock frequency and requested baud rate. Assume that the system clock is 50MHZ, baud rate is 9600bps.Therefore the frequency coefficient (M) of baud rate generator is:

M=50*10^6/9600Hz=5200

B. TRANSMIT MODULE

The UART transmit module converts the bytes into serial bits according to the basic frame format and transmits those bits through TXD. The 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. The schematic of a transmitter module is as shown in the figure below.The 8-bit ASCII data is supplied in to the module via. Din[7:0]. The ‘ready’line is HIGH whenever the module is ready to transmit data, ‘trde’ line is HIGH when the transmission of data is complete i.e. the stop bit is transmitted. ’CLR’ is used to reset all the parameter, the ‘clk’ provides timing information regarding the flow of data bits. Serial output data is transmitted via. Dout line.

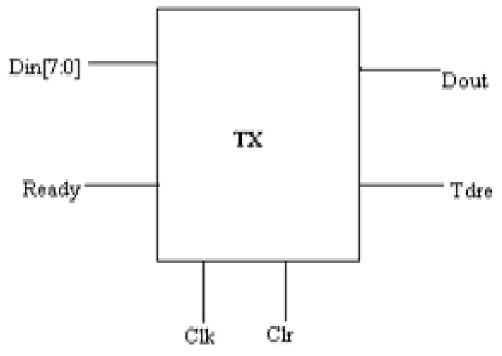

Fig.4 Transmitter module

The transmitter only needs to output 1 bit every clock cycles(the transmitting clock frequency generated by baud rate generator). The order follows 1 start bit , 8 data bits ,1 parity bit and 1 stop bit. The parity bits is determined according to the number of logic 1 in 8 data bits. Then the parity bits is output. Finally 1 is output as the stop bit. Fig. 5 shows the transmit module state diagram

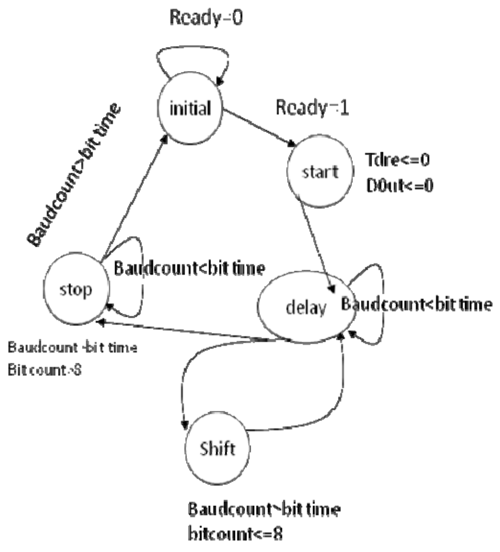

Fig. 5 UART Transmiter State Machine
This state machine has 5 states : MARK(idle), START(start bit), DELAY, SHIFT(shift), STOP(stop bit).

- *MARK Status* **:** When the UART is reset, the state machine will be in this state, the UART transmitter has been waiting a data frame sending command READY. When READY=1 , the state machine transferred to START , get ready to send a start bit.

- *START Status* **:** In this state, sends a logic 0 signal to the TXD for one bit time width, the start bit. Then the state machine transferred to DELAY state.

- *DELAY Status :* when the state machine is in this state, waiting for counting baud rate clock then entering into the SAMPLE to sample the data bits .at the same time determining whether the collected data bit length has reached the data frame length. If reaches, it mens the stop bits arrives.inthis design it is 8, which corresponds to the 8-bit data formate of UART

- *SHIFT Status :* In this state , the state machine realizes the parallel to serial converision of outgoing data. Then immediately return to DELAY state

- *STOP Status :* when the data frame transmit is completed, the state machine transferred to this state,and sends 5200 baud clock cycle logic 1 signal,that is 1 stop bit.then the state turns to mark state after sending the stop bitand waits for another data frame transmit

C.RECEIVER MODULE

The UART receiver module is used to receiver the serial signals at RXD, and convert them into parallel data. The receiver module design is largely complementary to that of the transmitter design. The schematic of a receiver module is as shown in the figure below.The 8-bit ASCII data is revceived by the module via. Din.The 'rdrf'line is HIGH whenever the module has received the stop bit, 'fe' line is HIGH when there is an error in reception of the frame.However for simplcity purpose in this design the 'fe' line is HIGH only when there is an error in reception of the STOP bit. 'CLR' is used to reset all the parameters , the 'clk' provides timing information regarding the flow of data bits.

Serial output data is transmitted via. Dout[7:0] line. The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of the data frame . The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of the data frame. The receiver module receives data from RXD pin.

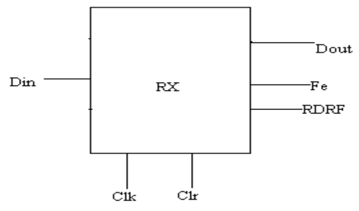

Fig 6.Receiver Module

The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of the data frame . As stated earlier , the parameters used in design of an Rx module remains the same. A new parameter namely 'half_bit_time' which represents half of the bit time. Due to the delay in transmission of data , the start bit appears only after half the bit time has elapsed. Once the start bit been identified, from the next bit,being to count the rising edge of the baud clock, and sample RXD when counting. Each sampled value of the logic level is deposited in the register rbuf[7,0] by order.when the count equals 8, all the data bits are surely received, also the 8 serial bits are converted into parallel data. The receiver finite state manchine diagram is show in fig. 7

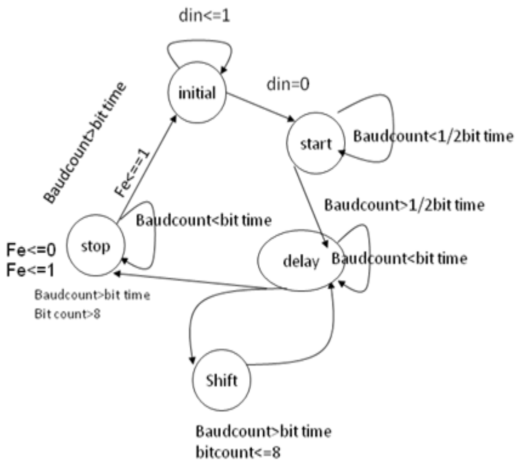

Fig.7 UART Receiver State Machine

The state machine includes five states: MARK(wating for startbit),START(findmidpoint),DELAY,SHIFT(shift) ,STOP(receving stop bit)
.

- *MARK status:* when the UART receiver is reset , the receiver state machine will be in this state.
- *START Status:* For asynchronous serial signal, in the order to detect the correct signal each time , and minimize the total error in the later data bits detection. In this state, the task is to find midpoint of each bit through the start bit.
- *DELAY Status:*Similar with the Transmit _DELAY of UART transmit state machine .
- *SHIFT Status:* After sampling the state machine transfer to DELAY state unconditionally, waits for the arrival of next start bit.
- *STOP Status:* Stop bit is either 1 or 1.5,or 2. State machine doesn't detect RXD in STOP, but output frame receiving done signal. After the stop bit, state machine turns back to START state, waiting for the next frame start.

III. SIMULATION OF MODULES

The simulation software is Xilinx. During simulation, the system clock is set to 50MHZ, and baud rate is set to 9600bps. And the selected device is Sparton 3E FPGA.

A. *Transmitter Simulation*

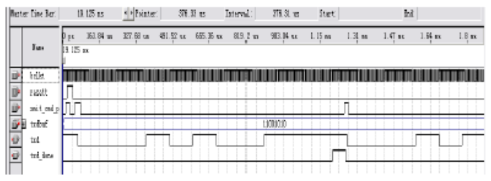

Fig.8 Simulation Results of Transmitter

Fig.8 shows the Transmitter simulation. The present_state indicates the current state of the state machine. It traverses from IDLE to STOP state. The data input can be seen on din and corresponding serial output is given on dout. Since parity_en = 1, parity bit is appended to data.

B. Receiver Simulation

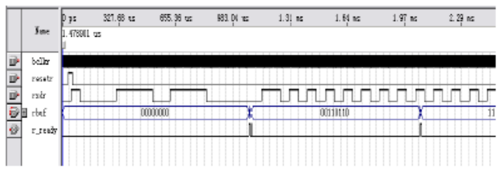

Fig.9 Simulation Results of Receiver

Fig. 9 shows the Receiver simulation. The state transitions are similar to transmitter. The serial data input comes on sin and output data is dout. The data is sampled when data_rx_done = ‘1’.

IV. CONCLUSION

This design uses VERILOG as design language to achieve the modules of UART. Using XILINX software, Saptron 3E FPGA to complete simulation and test . The results are stable and reliable data transmission with some reference value and great flexibility, high integration.

REFERENCES

[1]. Zou,Jie Yang,Jianning Design and Realization of UART Controller Based on FPGA

[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]. HU Hua, BAI Feng-e. Design and Simulation of UART Serial Communication Module Based on Verilog -HDL[J].
J Isuanj I Yu Xianda Ihua 2008 Vol. 8

[4]. Bhaskar, J. A Verilog HDL Primer -Pearson Education. 2001

[5] Palnitkar, Samir. Verilog HDL - A Guide to Digital Design and Synthesis-.

[6] Smith, J. Douglas. HDL design