IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

High Speed UART Design Using Verilog

Ambika[1], Prof. Anuradha S[2]

M. Tech Student, VLSI Design & Embedded Systems, APPA IET, Kalaburagi, Karnataka, India[1]

Associate Professor, Dept. of Electronics & Communication Engineering, APPA IET, Kalaburagi, Karnataka, India[1]

Abstract: Universal Asynchronous Receiver Transmitter (UART) is usually an individual integrated circuit used for serial communication over a computer or peripheral device serial port. This UART is designed for make an interface between RS232 line and a microcontroller or an IP core. It works fine connected to a serial port of a PC for data exchange with custom electronic. UARTs are now commonly included in microcontrollers. A UART is a full duplex transmitter or receiver. UART is the kind of serial communication protocol. In parallel communication the cost as well as complexity of the system increases due to simultaneous transmission of data bits on multiple serial communications alleviates this drawback of parallel communication and emerges the effectively in many applications for long distance communication as it reduces the signal distortion because of its simple structure. The high speed UART is designed using verilog and delay is minimized (5.582ns). The synthesis of UART is done by using Xilinx 13.2 and simulation is done by ModelSim10.1.

Keywords: UART, Verilog, Xilinx 13.2, ModelSim10.1, High speed.

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. Asynchronous serial communication is usually implemented by Universal Asynchronous Receiver Transmitter (UART). UART allows full-duplex communication in serial link, thus has been widely used in the data communications and control system. In actual applications, usually only a few key features of UART are needed. Specific interface chip will cause waste of resources and increased cost. 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.

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. UART's basic features are:

There are two states in the signal line, using logic 1 (high) and logic 0 (low) to distinguish respectively. For example, 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, 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 be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word. After the Start Bit, the individual data bits of the word are sent, with the Least Significant Bit (LSB) being sent first. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits, and the receiver "looks" at the wire at approximately halfway through the period assigned to each bit to determine if the bit is a 1 or a 0. For example, if it takes two seconds to send each bit the receiver will examine the signal to determine if it is a 1 or a 0 after one second has passed, then it will wait two seconds and then examine the value of the next bit, and so on. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. When the receiver has received all of the bits in the data word, it may check for the Parity Bits (both sender and receiver must agree on whether a Parity Bit is to be used), and then the receiver looks for a Stop Bit. If the Stop Bit does not appear when it is supposed to, the UART considers the entire word to be garbled and will report a Framing Error to the host processor when the data word is read.

The usual cause of a Framing Error is that the sender and receiver clocks were not running at the same speed, or that the signal was interrupted. Regardless of whether the data was received correctly or not, the UART automatically discards the Start, Parity and Stop bits. If the sender and receiver are configured identically, these bits are not passed to the host. If 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. Because asynchronous data are "self-synchronizing", if there are no data to transmit, the transmission line can be idle. The UART frame format is shown in Figure. 1.

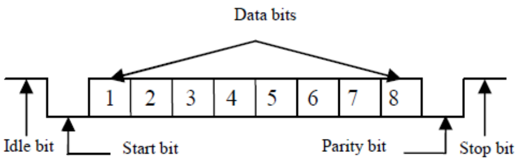


Figure 1: UART frame format

II. PROPOSED WORK

The top to bottom (Top to Down) design method is used. The UART serial communication module is divided into three sub-modules: the baud rate generator, receiver module and transmitter module, shown in Figure 2. Therefore, the implementation of the UART communication module is actually the realization of the three sub-modules. 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.

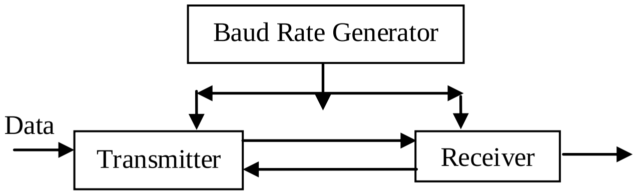


Figure 2: UART Module

Transmitter State Machine

When Transmitter Hold Register (THR) is empty Line Status Register (LSR) logic (1) data transfer from CPU to THR. Now LSR becomes ‘0’.Then data shifted out of Transmitter Shift Register (TSR).Finally stop bit (logic1) is generated to indicate the end of the frame.

After a frame is fully transmitted, another frame will be transmitted immediately. If THR is not empty. Start bit is high when no transmission is taking place.

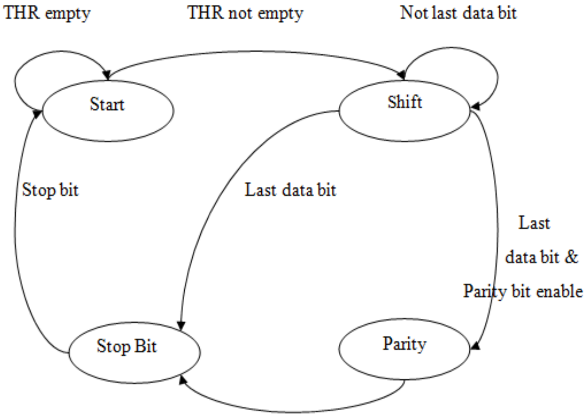


Figure 3: Transmitter State Machine

Receiver State Machine

The status of Receiver Shift Register (RSR) can be indicated by receiver frame synchronization requires 8valid clock start bit is received. The data bit and parity bit will be sampled every 16 clock (receiving baud rate).LSR will be up dated to show the received frame status. Whenever framing error is detected, the UART assumes that the error was due to the start bit.

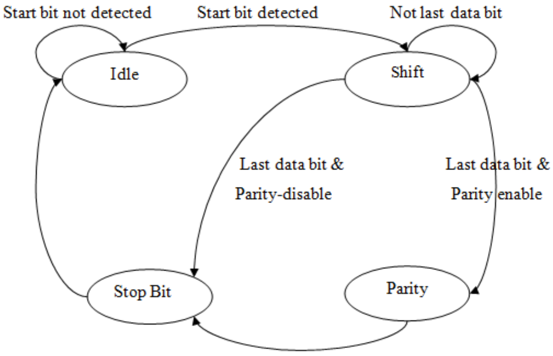


Figure 4: Receiver state machine

III. SYNTHESIS & SIMULATION RESULTS

HDL Synthesis Report

Macro Statistics

# Adders/Subtractors	: 3
4-bit adder	: 3
# Registers	: 19
1-bit register	: 14
4-bit register	: 3
8-bit register	: 2
# Comparators	: 6
4-bit comparator greatequal	: 2
4-bit comparator greater	: 1
4-bit comparator less	: 1
4-bit comparator lessequal	: 2

Final Register Report

Macro Statistics

# Registers	: 42
Flip-Flops	: 42

Device utilization summary:

Selected Device : 3s500efg320-4

Number of Slices:	43 out of	4656	0%
Number of Slice Flip Flops:	42 out of	9312	0%
Number of 4 input LUTs:	64 out of	9312	0%
Number of IOs:	27		
Number of bonded IOBs:	27 out of	232	11%
Number of GCLKs:	2 out of	24	8%

Timing Summary:

Speed Grade: -4

Minimum period: 5.582ns (Maximum Frequency: 179.147MHz)

Minimum input arrival time before clock: 5.482ns

Maximum output required time after clock: 4.532ns

Maximum combinational path delay: No path found

Delay Analysis

Delay	:	5.582ns (Levels of Logic = 3)
Source	:	rx_sample_cnt_0 (FF)
Destination	:	rx_cnt_0 (FF)
Source Clock	:	rxclk rising
Destination Clock	:	rxclk rising

Data Path: rx_sample_cnt_0 to rx_cnt_0

Gate Net

DOI 10.17148/IJARCCE.2016.5229

```
Cell:in->out        fanout   Delay   Delay  Logical Name
------------------------------------------------
FDCE:C->Q                       5     0.591      0.637
rx_sample_cnt_0
LUT4_D:I3->O                  12    0.704       0.965
rx_empty_cmp_eq00011
LUT4_L:I3->LO                 1     0.704       0.135
rx_cnt_not000117
LUT4:I2->O                      4     0.704      0.587
rx_cnt_not000130
 FDCE:CE                0.555             rx_cnt_0
 ------------------------------------------------
```

Total Delay 5.582ns (3.258ns logic, 2.324ns route) (58.4% logic, 41.6% route)

Memory Usage
Total memory usage is 238624 kilobytes.

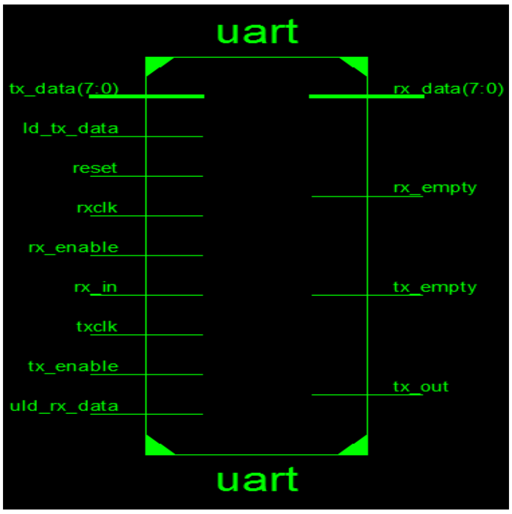


Figure 5: Top module of UART

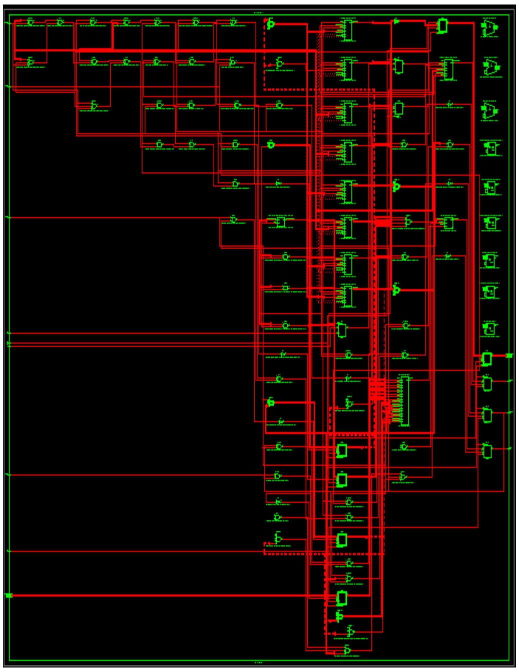
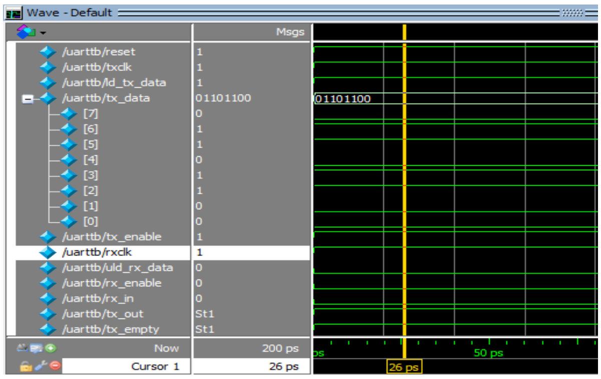


Figure 6: RTL Schematic of UART



	Msgs
/uarttb/reset	1
/uarttb/txclk	1
/uarttb/ld_tx_data	1
/uarttb/tx_data	01101100
[7]	0
[6]	1
[5]	1
[4]	0
[3]	1
[2]	1
[1]	0
[0]	0
/uarttb/tx_enable	1
/uarttb/rxclk	1
/uarttb/uld_rx_data	0
/uarttb/rx_enable	0
/uarttb/rx_in	0
/uarttb/tx_out	St1
/uarttb/tx_empty	St1
Now	200 ps
Cursor 1	26 ps

Figure 7: Simulation result of UART

IV. CONCLUSION

The proposed design of high speed UART is simulated in ModelSim 10.1 and synthesized in Xilinx ISE 13.2 software and the source code is written in Verilog. This proposed design has less delay & memory usage. This proposed high speed UART has delay 5.582ns (Levels of Logic =1) and memory utilized 238624 kilobytes.

REFERENCES

1. FANG Yi-yuan CHEN Xue-jun, “Design and Simulation of UART Serial Communication Module Based on VHDL”, 978-1-4244-9857-4/11/$26.00 ©2011 IEEE.
2. Amanpreet Kaur, Amandeep Kaur, “An Approach for Designing a Universal synchronous Receiver Transmitter (UART)”, International Journal of Engineering Research and Applications. Vol. 2, Issue 3, May-Jun 2012.
3. Ms.Neha R. Laddha and Prof.A.P.Thakare, Implementation of serial communication using UART with configurable baud rate”, International Journal on Recent and Innovation Trends in Computing and Communication ISSN 2321 – 8169 Volume: 1 Issue: 4, IJRITCC | APR 2013.
4. SAAKSHI KEDAR et al, “VHDL IMPLEMENTATION OF UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER”, Proceedings of 11th IRF International Conference, 15th June-2014, Pune, India, ISBN: 978-93-84209-27-8. IEEE2014.
5. ARPITA TIWARI et al, “FPGA IMPLEMENTATION OF ADVANCED UART CONTROLLER USING VHDL”, IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET) ISSN(E): 2321-8843; ISSN(P): 2347-4599 Vol. 2, Issue 8, Aug 2014.
6. Sandeep Singh Rawat, & Prof.Vishal Ramola, “ UART design using FIFO ram and LCR circuit with BIST capability at different baud rate”, International Journal of Engineering Development and Research. 2015 IJEDR | Volume 3, Issue 3 | ISSN: 2321-9939.

DOI 10.17148/IJARCCE.2016.5229