ISSN: 2321-7782 (Online)
Volume 3, Issue 10, October 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

UART Design Using VHDL

Kavyashree S
1st semester, M.Tech VLSI and Embedded Systems
R.V College of Engineering
Bengaluru, India

Abstract: ***This paper presents the design of UART, a serial interface. The UART is designed using VHDL. The UART designed can be configured to work at different baud rates. It has FIFO storage, programmable serial interface characteristics, complete status reporting capabilities and error detection. The design is simulated on ModelSim using VHDL description.***

Keywords: UART; baud rate; registers; FIFO; interrupts

I. INTRODUCTION

Universal asynchronous Receiver Transmitter (UART) is serial bidirectional full duplex communication interface. It translates data from parallel forms to serial. It uses Asynchronous transmission and hence there no need for clock. Transmitter and Receiver agree upon particular format to synchronize the transmission and reception.

II. DESIGN DETAILS

Wherever Times is specified, Times Roman or Times New Roman may be used. If neither is available on your word processor, please use the font closest in appearance to Times. Avoid using bit-mapped fonts if possible. True-Type 1 or Open Type fonts are preferred. Please embed symbol fonts, as well, for math, etc. The design involves several individual components. The main components of the design are parallel interface, configuration module, FIFO (TxFIFO, RxFIFO), baud rate generator, shift registers, Interrupt controller. The UART design is structural and each component is individually designed and tested and then interfaced.

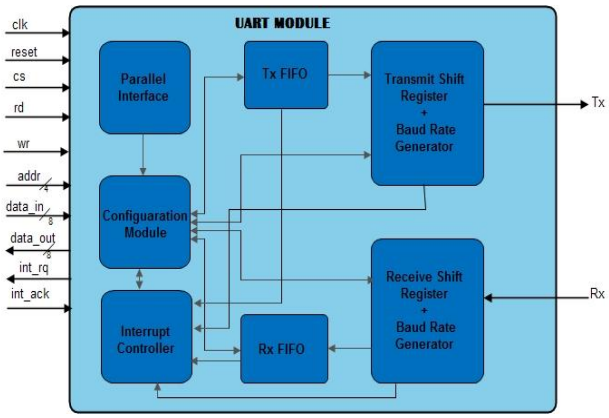

Fig. 1 UART Module Design Details

a) *Parallel Interface:* The interface is multiline channel which synchronizes the control inputs . It converts the asynchronous control (cs,wr,rd)from the device to synchronous controls .

b) *Configuration module:* Configuration module is a set of controlling, monitoring and data transfer registers. Configuration module enables the UART to be suited for several applications. It allows the UART to become programmable to different configurations. The registers may be read only, write only or read write. This module consists of 9 registers and each is accessed using 3 bit address lines to this module. All the registers are of 8 bit.

TABLE I

MEMORY MAP

Register	Address	R/W
Transmit Holding Register (THR)	0x00	W
Receive Holding Register (RHR)	0x01	R
FIFO Control Register (FCR)	0x02	W
Line Control Register (LCR)	0x03	R/W
Line Status Register (LSR)	0x04	R
Interrupt Enable Resister (IER)	0x05	R/W
Interrupt Status Register (ISR)	0x06	R
Divide Latch Low Byte (DLL)	0x07	W
Divide Latch High Byte (DLH)	0x08	W

c) *FIFO:* It is used to buffer the data to be transmitted or received data when two subsystems are operating at different rates. In this design the FIFO stores the data up to 16 bytes. The design contains two FIFOs one for transmission TxFIFO and one for reception RxFIFO. During transmission the data is read out of TxFIFO and during reception the data is written onto RxFIFO.

d) *Baud Rate Generator(BRG):* BRG specifies the rate at which transmission or reception should happen. During transmission the data is shifted out to Tx line at Baud Rate and in reception data on Rx line is sampled at baud rate. It is configurable with 8 bit latch registers (DLL and DLH) to different baud rates.

Divisor (decimal) = (clock frequency) / (baud rate x clock sampling rate)

e) *Shift Registers:* Shift register is the heart of UART module. It is used to translate between serial and parallel forms. The design involves two shift registers. Transmit shift register (TSR) shifts out the data at baud rate. Framing is done to synchronize the communication. A typical frame is as shown below.

Start Bit(0)	Data Bits (5/6/7/8)	Optional parity bit	Stop Bits (1/2)

The programmable data bits, stop bits and optional parity are specified by Line Control Register (LCR).The TSR is a state machine having five states- idle, start, data, parity (optional) and stop.

idle: The Tx line has a value '1' when there is no transmission.

start: Appends a bit '0' at the start of data transmission.

data: Shifts all data bits (specified by FCR) at baud rate(specified by DLL and DLH).

parity: An optional parity bit is appended as specified by LCR. The parity type may be odd or even.

stop: Specify the end of one frame transmission(specified by LCR).

RSR shifts the serial data on Rx line at baud rate. It removes the start, parity and stop bits and stores the actual data. The Rx line is sampled at the middle of the clock cycle to avoid false detection.

f) *Interrupt Controller:* During transmission and reception there may be many undesired events. These may lead to improper functioning of the UART. Interrupt controller handles such situations and takes a course of action to come of situation to proper functioning. The user can configure the interrupts via the Interrupt Enable Register (IER). When the

processor is interrupted, it can find out more details regarding the interrupt by reading the Interrupt Identity Register (IIR). The design is capable of handling 4 interrupts.

Timeout interrupt: Interrupt arises when a particular data remains in the RxFIFO for a very long time

Trigger Level: When RxFIFO reaches a certain level configured by the userthis interrupt is raised.This is used to avoid the reading one byte at a time. When the FIFO reaches the level all the data are read at once saving the system's resources.The FIFO trigger level can be set to 1, 4, 8 and 14 bytes. If the received data does not reach the level then the timeout interrupt is used to read out the data.

Tx empty: It is raised when TxFIFO is empty and transmission is attempted.

Receive data error: This interrupt is raised when there are errors in the received data. The errors may be parity or framing errors.

III. Results and Discussion

The UART designed is tested and verified. Each component is individually tested and the integrated module is tested for the overall function. The simulated results are shown below:

a) Transmission

The data sent is transmitted on the Tx line at baud rate. The baud rate is configured to 57.6kbps. The word length is 8 bits. The data 10010111 is transmitted on Tx line with start bit '0', stop bit '1'.

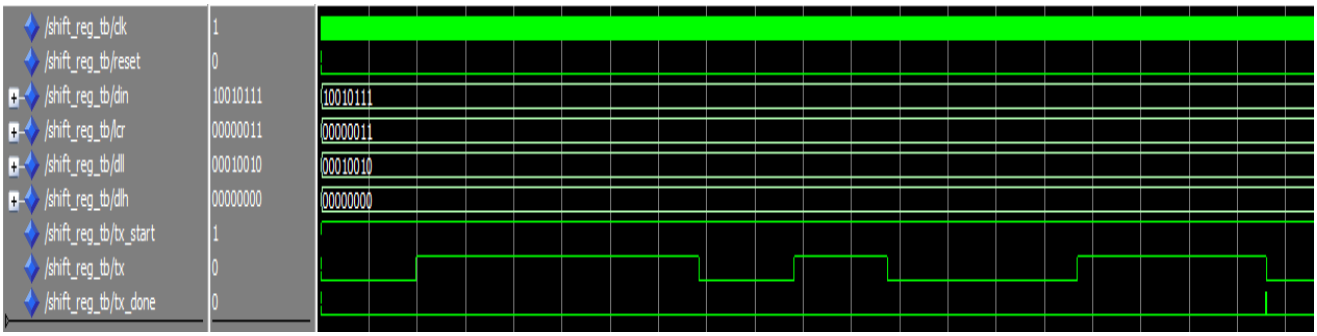

Fig. 1 Simulation Waveform of Transmission

b) Receiver

The receiver is sampled at 115.2kbps.The LCR (Line control register is configured for a word length of 8 bits. The data 1101000is received.

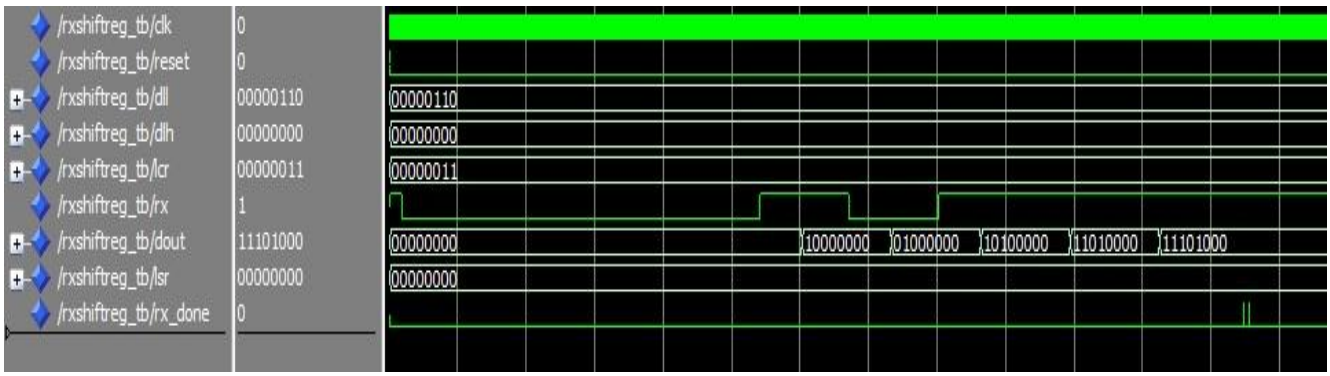

Fig. 3 Simulation Waveform of Receiver

c) *Complete UART Module*

The UART with all the controls is shown. It shows the transmission of data '11100100' on Tx with start and stop bits and also the receiving of '11101000' on Rx line.

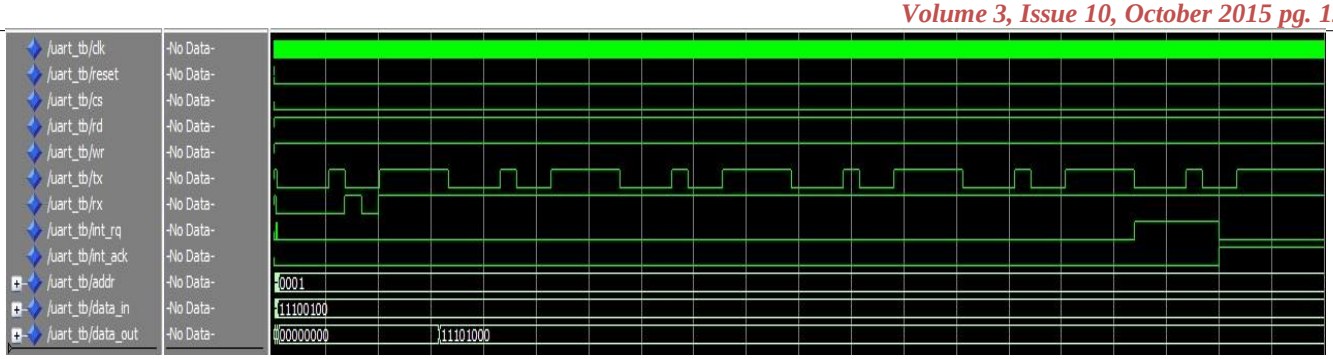

Fig. 4 Simulation Waveform of UART Module

IV. CONCLUSION

In this paper the design of UART is described along with the individual modules contained in it. Each module is described using VHDL and simulated using modelsim. This design provides a programmable UART suitable for variety of subsystems

References

1. Microcomputer Systems- The 8086/8088 Family: Yu-cheng Liu & Glenn A. Gibson.
2. Pong P Chu, RTL hardware design using VHDL, coding for efficiency, portability and scalability, WILEY-INTERSCIENCE.
3. Circuit Design With VHDL By Volnei A. Pedroni,MIT Press.

AUTHOR(S) PROFILE

Kavyashree S received Bachelor of Engineering degree from National Institute of Engineering, Mysore in 2013. She is perceiving her Master of Technology from R.V college of Engineering in VLSI design and Embedded Systems.

