Amanpreet Kaur, Amandeep Kaur / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 3, May-Jun 2012, pp.2305-2311

An Approach For Designing A Universal Asynchronous Receiver Transmitter (UART)

Amanpreet Kaur[#1], Amandeep Kaur[*2]

[#]Pursuing M.Tech, ECE Dept. ,University College of Engineering,Punjabi University Patiala, Punjab, India

[*]Assistant Professor, ECE Dept., University College of Engineering,Punjabi University Patiala, Punjab, India

Abstract—This paper describes the design of Universal Asynchronous Receiver and Transmitter (UART) using VHDL. A UART is a full duplex receiver/transmitter. It is the microchip with programming that controls a computer's interface to its attached serial devices. It handles the conversion between serial and parallel data. It is the most widely used serial data communication circuit ever. Whole process of serial transmission is based upon the principle of shift register.

Keywords— UART, VHDL;

I. INTRODUCTION

Universal Asynchronous Receiver Transmitter (UART) is generally used for better transmission of serial data that is it either transmit or receives data serially. It is a popular and widely used device for data communication in the field of telecommunication. There are different versions of UARTs in the industry. UART is an integrated circuit containing a transmitter(parallel to serial converter) and a receiver (serial to parallel converter) each clocked separately. It transmit 9600 to 38400 bps for transmitting data bit . Whole process of serial transmission is based upon the principle of shift register[14]. There are two primary forms of serial transmission: Synchronous and Asynchronous.

Synchronous serial transmission requires that the sender and receiver share a clock with one another, or that the sender provide a strobe or other timing signal so that the receiver knows when to "read" the next bit of the data. In most form of Serial Synchronous Communication, if there is no data available at a given instant to transmit, a fill character must be sent instead so that data is always being transmitted. Asynchronous transmission allows data to be transmitted without the sender having to send a clock signal to the receiver. Instead, the sender and receiver must agree on timing parameters in advance and special bits are added to each word which are used to synchronize the sending and receiving units[16].

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. This Asynchronous serial communication is usually implemented by UART[5].

UART transmission protocol

It usually include start bit, data bit, parity bit, stop bit and idle state[5] as shown in fig.(a)

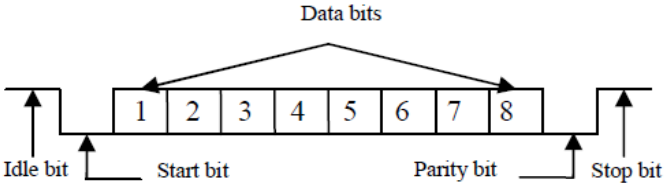


Fig.(a) UART Data Frame Format

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 After the Start Bit, the individual bits of the word of data are sent, with the Least Significant Bit (LSB) being sent first into synchronization with the clock in the transmitter. 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. Then at least one Stop Bit is sent by the transmitter[5]. If incorrectly formatted data is received, the UART may signal a framing error. If another byte is received before the previous one is read, the UART will signal an overrun error[16].

II. ROLE OF VHDL IN DESIGNING DIGITAL CIRCUITS

The design of the system at the gate level is more time consuming since the integrated circuit technology is more complex. Therefore the use of VHDL (Very High Speed Integrated circuit hardware Description Language) is preferred. VHDL can be used to describe and simulate the operation of digital circuits ranging from few gate to more complex gates. VHDL can be used for the behavioral level design implementation of a UART and it offers several advantages. These are the advantages of using VHDL to implement UART:

1. VHDL allows us to describe the function of the transmitter in a more behavioral manner rather than focus on its actual implementation at the gate level .
2. VHDL makes the design implementation easier to read and understand.

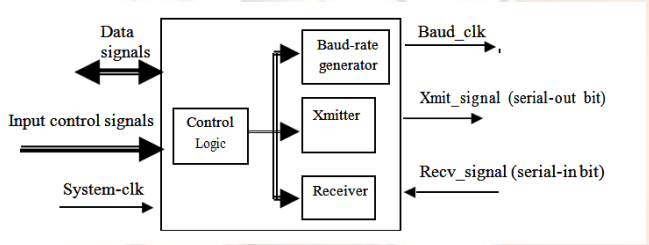
3. It is easier to test the UART by the VHDL simulation and find out if any discrepancy occurs.[5]

III. THEORY OF OPERATION

UART is generally consists of transmitter module, receiver module, baud-rate generator and control circuitAs shown in fig.(b)

System-clock is used to produce baud_clock, which is generally 16 times baud-rate. Xmitter module converts parallel data input into serial data output and receiver module does vice versa. The basic functions of a UART are a microprocessor interface, double buffering of transmitter data, frame generation, parity generation, parallel to serial conversion, double buffering of receiver data, parity checking, serial to parallel conversion.

In data transmission through UART, once the baud-rate has been established (prior to initial communication), both the transmitter and the receiver's internal clock are set to the same frequency.



Fig(b) Block diagram of UART[13]

The serial transmitter section consists of an 8-bit Transmitter Hold Register (THR) and Transmitter Shift Register (TSR). Parallel data is stored in THR which received from the CPU, then it is transferred to the shift register and send out in serial data as shown in fig(c). At the same time parity bit is generated and transmitted by TSR, when the whole character is removed from the TSR, CPU interrupt signal is generated[3].

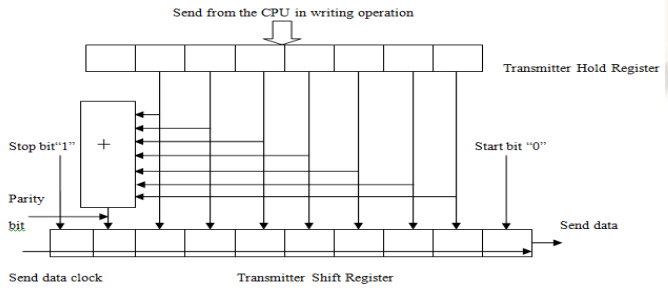


Fig.(c) Transmitter Unit[13]

The serial receiver section also contains an 8-bit Receiver Buffer Register (RBR) and Receiver Shift Register (RSR). Serial data received is stored in the RSR, when the RSR receive the whole character, automatically sent to RBR, status register will be set and generate CPU interrupt signal, parallel data will be transmitted to the CPU in the read command of the CPU, serial-to-parallel conversion is performed as shown in fig(d).

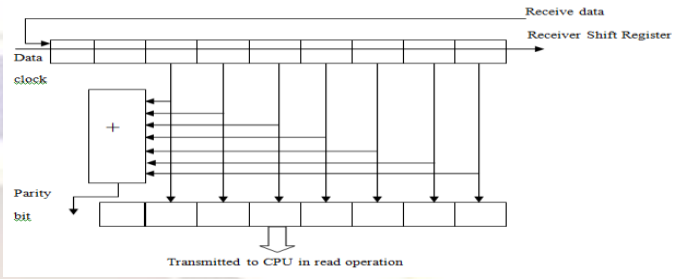


Fig.(e) Receiver Unit

Features of UART

1. Full-duplex asynchronous receiver/transmitter
2. The ability to convert data from serial to parallel, and from parallel to serial, using shift registers.
3. Programmable data format: usually include start bit, data bit, parity bit, stop bit and idle state.
4. Parity, framing, and overrun detection
5. False start bit detection
6. Line break detection and generation

IV UART Transmitter and Receiver design using VHDL

A. UART Transmitter Functional Pins

The signals used by the transmitter are given in the table 1. The transmitter interfaces to the data bus with the transmitter buffer register empty(tbre) and the wrn signals. The controller can generate a wrn strobe if tbre is high. The transmitter is double buffered, allowing data on din[7:0] to be written to the buffer register tbr[7:0] while data is being shifted out of the shift register tsr[7:0]. The transmitter generates a frame which consists of the idle state (high on sdo), low start bit, eight data bits, and a stop bit.

The no_bits_sent controls the word size and sequences the transmitter operations. To change the word size, change the value of no_bits_sent .

Table1. Transmitter Signals

Signal	Direction	Function
rst	Input	Resets wrn1,wrn2,no_bits_sent, clkdiv[3:0],tbr[7:0],tsr[7:0]
clk16x	Input	Local reference clock 16X the data rate
wrn	Input	Control signal which strobes data from din[7:0] to tbr[7:0]
din[7:0]	Input	Input data bus
sdo	Output	Serial data output
tbre	Output	Status signal indication that the transmitter buffer register is empty
no_bits_sent	Internal	Controls word_size and sequences transmitter operation
clk1x_enable	Internal	Enables internal clock clk1x.
tbr[7:0]	Internal	Accepts data from din[7:0] and transfers data to tsr[7:0]
tsr[7:0]	Internal	Receives data from tbr[7:0] and shifts to sdo
clkdiv[3:0]	Internal	Used in generation of internal clock

Table2. Receiver Signals

Signal	Direction	Function
rst	Input	Reset
clk16x	Input	16x input clock
rdn	Input	Read strobe
dout[7:0]	Output	Output data bus
framing_error	Output	Framing error status signal
parity_error	Output	Parity error status signal.
rbr[7:0]	Internal	Receiver buffer register - accepts data from data[7:0] and transfers it to rsr[7:0]
rsr[7:0]	Internal	Receiver shift register - accepts data from rbr[7:0] and transfers it to sdo
no_bits_rcvd	Internal	Tracks character size and sequences receiver operation
clk1x_enable	Internal	Enable signal for registers clocked by clk1x
clk1x	Internal	1x clock used for internal operations

B. UART Receiver Functional Pins

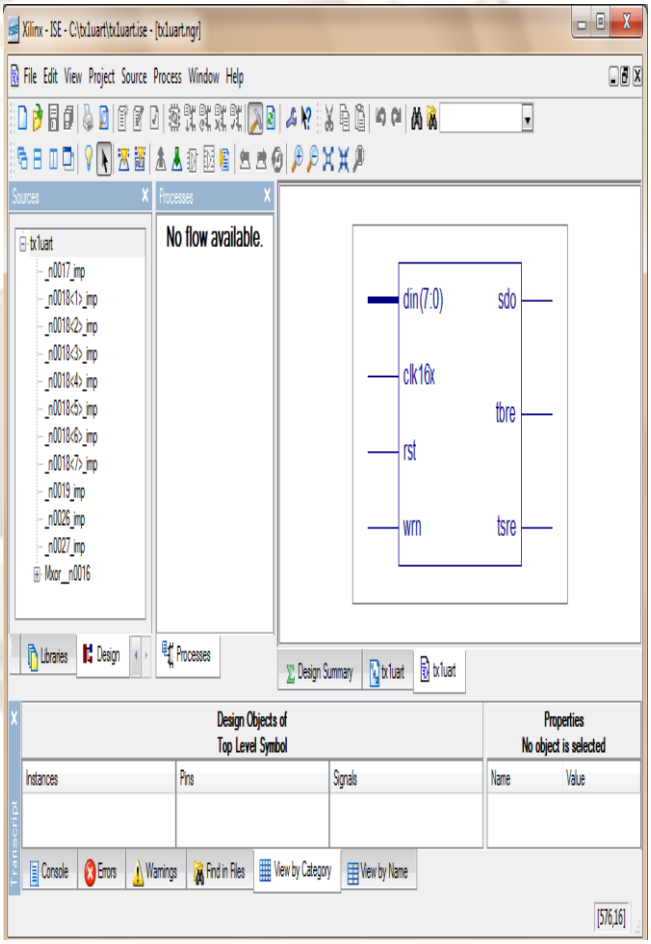
The signals used by the receiver are given in the table 2. The receiver interfaces to the data bus dout[7:0] with the rdn signal. The controller can generate a rdn strobe if data_ready is true. The receiver is double buffered, allowing data to be held in the buffer register rbr[7:0] while data is shifted in serially into the receiver shift register rsr[7:0]. This provides the controller flexibility with bus read operations.
The receiver detects the character frame and strips the start and stop bits. The no_bits_rcvd variable controls the word size. The clkdiv[3:0] register is used to control the time at which the data is decoded. The receiver uses the 16x local clock and decodes the value of start, data, and stop bits in the center of the data cells. To do this, the start bit initializes a count operation using clkdiv[3:0].
After detecting the low going edge on the start bit, the receiver counts the 16x clock to 8 and decodes, or samples the value of the signal. The clkdiv[3:0] register is then reset to 0, and subsequently counts the 16x clock to 16. This provides center sampling for the data and stop bits.

Three error detection signals are commonly used in UART:

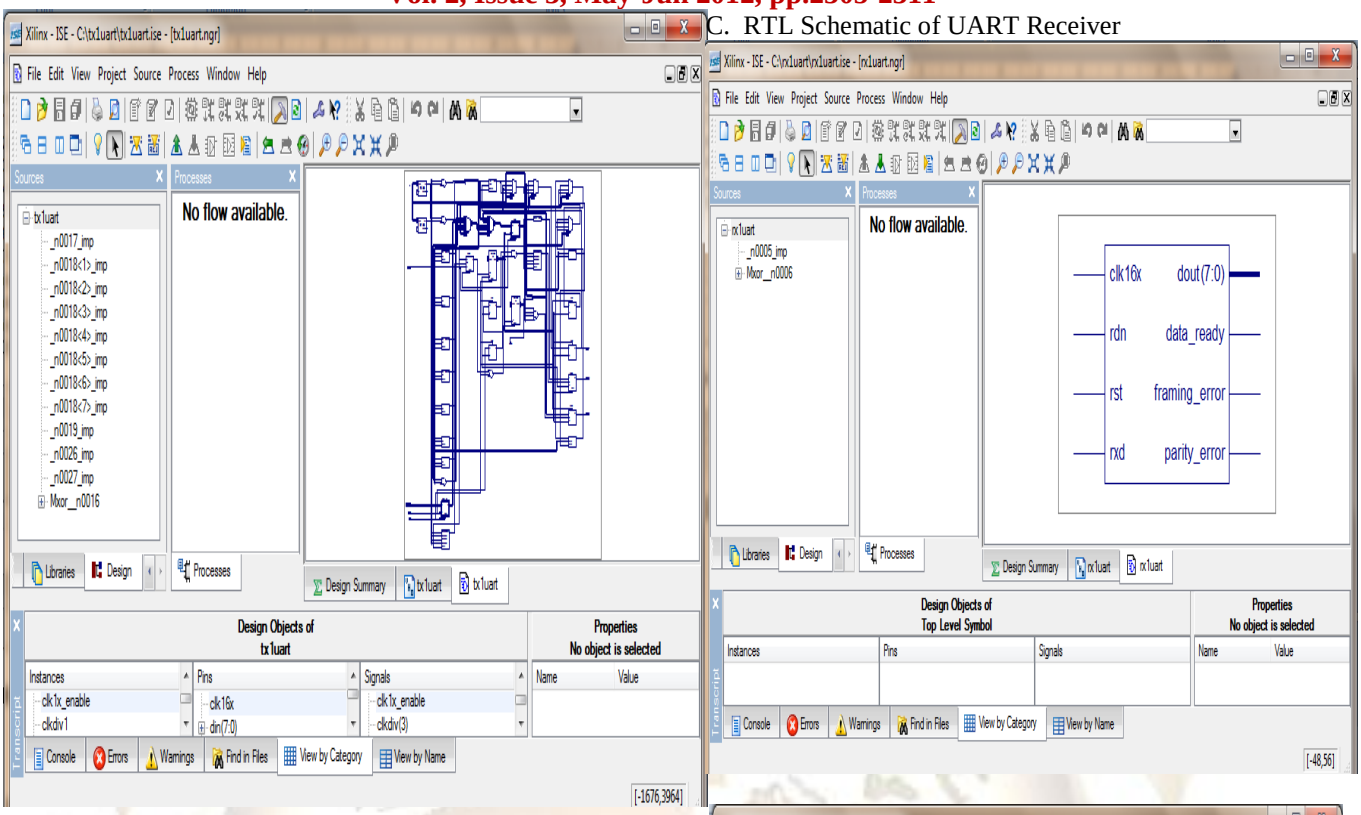
1. Parity Error indicates whether an even or odd number of "1s" are present in a data work.
2. Overrun Error indicates whether the receive buffer register is overwritten by the receive shift register prior to the controller reading the receiver buffer register. Overrun Error is not implemented in the VHDL/Verilog source.
3. A "framing error" occurs when the designated "start" and "stop" bits are not valid. As the "start" bit is used to identify the beginning of an incoming character, it acts as a reference for the remaining bits. Framing Error indicates if the stop bit is not high[16].

V RESULTS OBTAINED

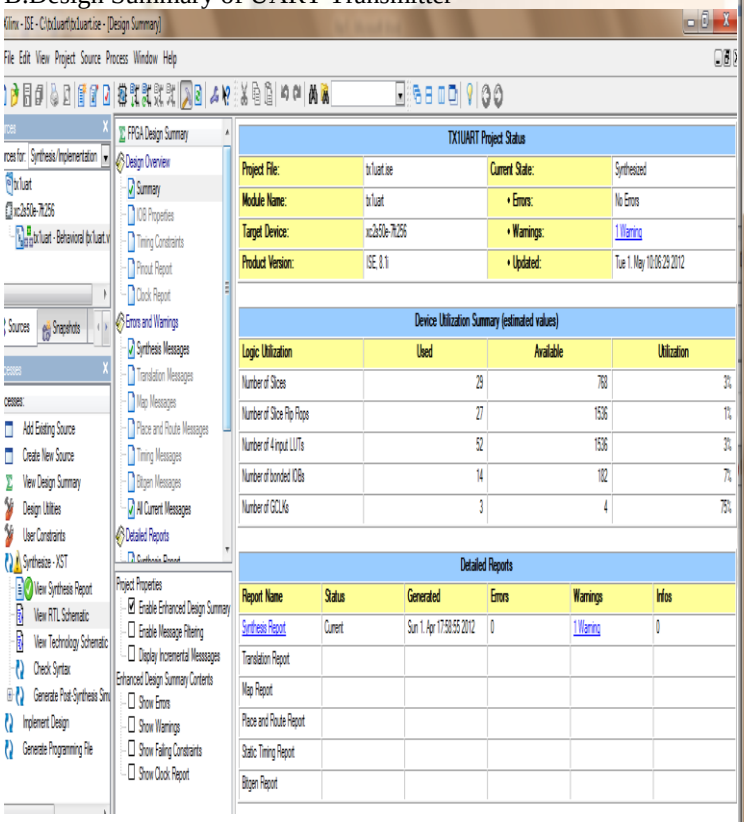
A. RTL Schematic of UART Transmitter

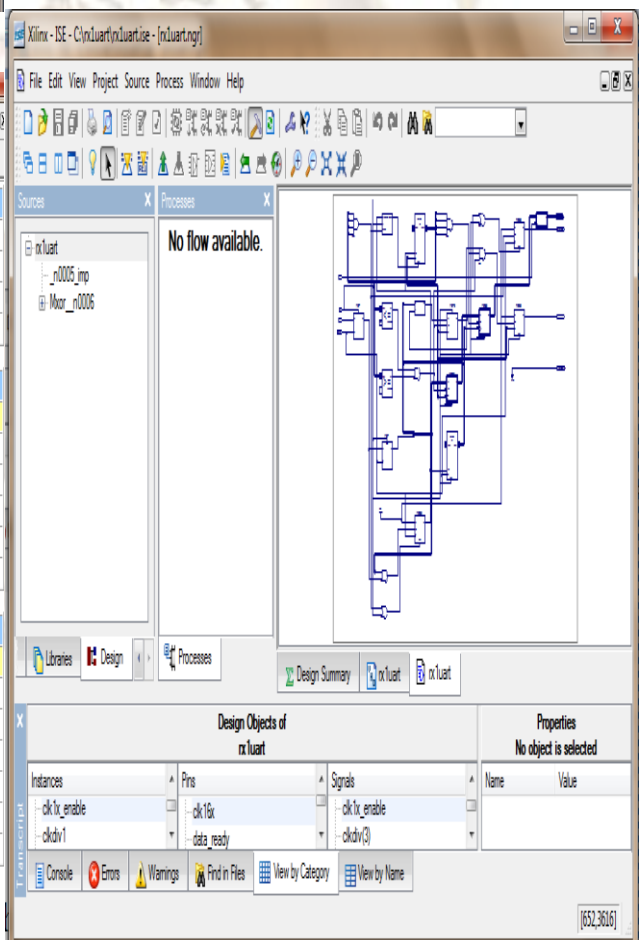


C. RTL Schematic of UART Receiver

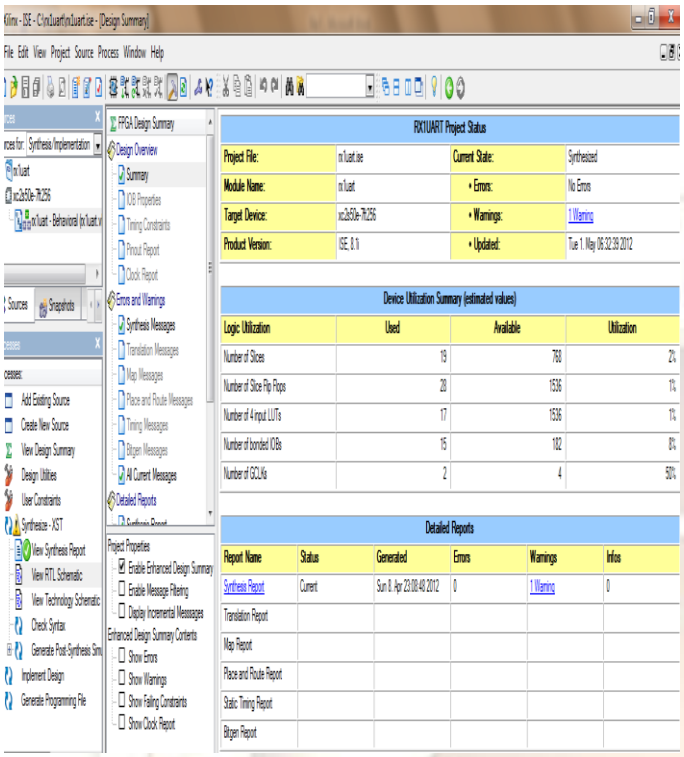


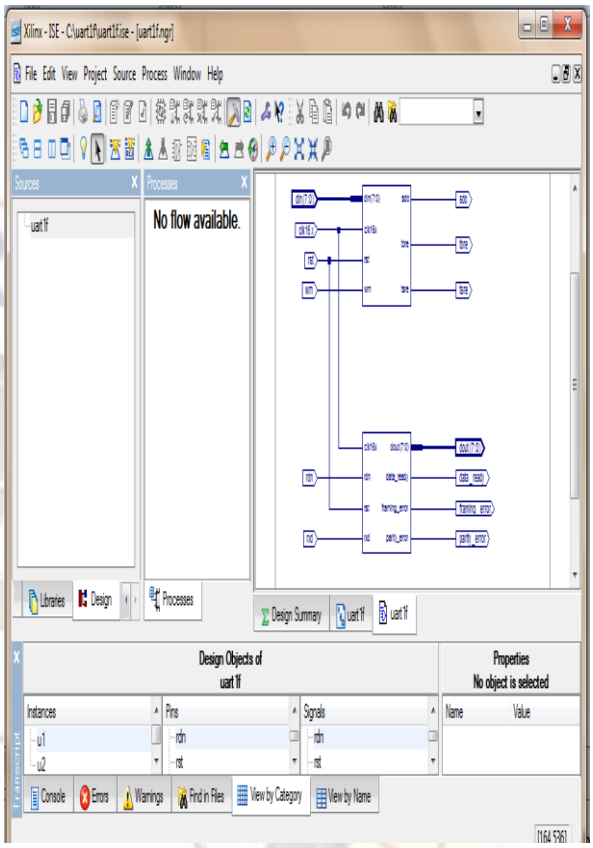
B.Design Summary of UART Transmitter



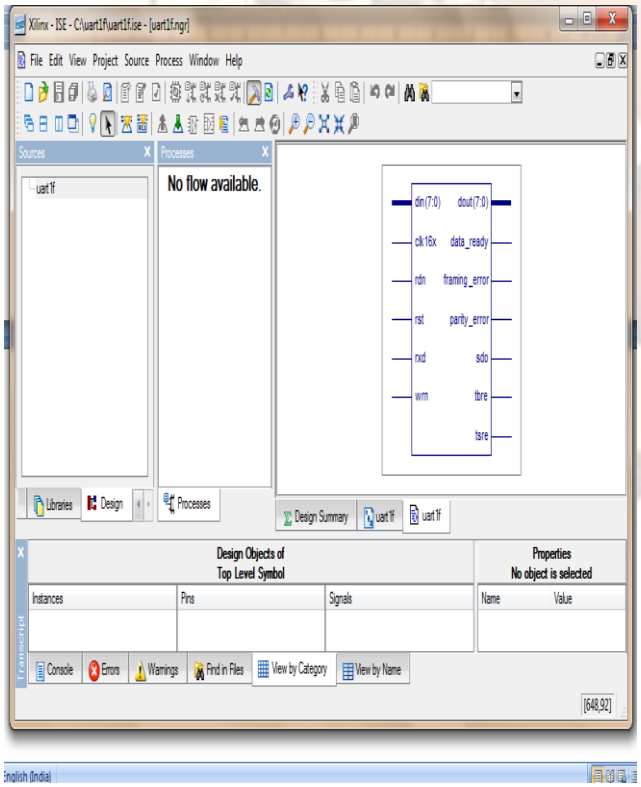


D. Design Summary of UART Receiver





E .RTL Schematic Of complete UART



VI. REFERENCES

[1] J. Norhuzaimin and H.H Maimun , "The design of high speed UART" Asia-Pacific Conference on Applied Electromagnetics(APACE-2005), Dec. 2005

[2] Mohammad-Hamed Razmkhah,Seyed Ghassen Miremadi and Alireza Ejlali, "A Micro- FT-UART for Safety-Critical SoC-Based Applications",2009

[3] Yongcheng Wang,Kefei Song, "A New Approach to Realize UART",2011

[4] Mohd Yamani Idna Idris, Mashkuri Yaacob, "A VHDL implementation of BIST Technique in UART Design",2006

[5] FANG Yi-Yuan CHEN XUE- jun , "Design and simulation Of UART serial communication Module Based on VHDL",2011.

[6] P. Himanshu;T. Sanjay; R. Neelkanthan and V. R. Gujraty, "A Robust UARTArchitecture Based on Recursive Running Sum Filter for Better Noise Performance," *Proc. of the 20th International Conference on VLSI Design*, pp. 819-823, Jan. 2007.

[7] Shouqian Yu ,Lili Yi, Weihai Chen, Zhao jin Wen," Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA", 2007

[8] Brian C. O'Neill, Steve Clark, Kar L. Wong, " Serial Communication Circuit with Optimized Skew Characteristics", 2001.

[9] Tong YAO,Yonghong HU, Lu DING, "FPGA-based for implementation of Multi-Serials to Ethernet Gateway", 2010

.[10] R. Gallo, M. Delvai, W. Elmenreich, and A. Steininger, "Revision and Verification of an Enhanced UART," *IEEE International Workshop on Factory Communication Systems*, pp. 315-318, Sept.2004.

[11] He Chun-zhi , Xia Yin-shui, Wang Lun-yao, "A Universal Asynchronous Receiver Transmitter Design",2011.

[12] W. Elmenreich, and M. Delvai, "Time-triggered Communication With UARTs," *IEEE International Workshop on Factory Communication Systems*, pp. 97-104, 2002.

[13] Liakot Ali, Roslina Sidek ,Ishak Aris, Alauddin Mohd. Ali, Bambang Sunaryo Suparjo," Design of a micro-UART for Soc application" ,2004 ,Elsevier Ltd.

[14] Ananya Chakraborty, Surbhi, Sukanya Gupta, Swati Deshkar, Pradeep Kumar Jaisal,"Design of UART (Universal Asynchronous Receiver Transmitter) using VHDL", IJCST Vol. 3, Issue 1, Jan. - March 2012.

[15]http://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter

[16] Frank Durda Serial and UART Tutorial. uhclem@FreeBSD.org