A Paper on Highly Efficient UART

Swati Dubey[1] Miss Amrita Singh[2]
[1]M. Tech Scholar [2]Assistant Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Dr. C.V. Raman University

Abstract— Universal Asynchronous Receiver Transmitter (UART) is the use of the serial communication protocol generally converts parallel data to serial data and vice versa, low velocity, short-distance, low-cost data exchange between computer & peripherals. During the genuine industrial production, sometimes we demand to simply integrate core part rather than full functionality of the UART. UART includes three modules which are received, the baud rate generator and transmitter. In the Baud Rate Generator part, before the overall design is synthesized into the UART design the Baud Rate Generator is incorporated. The role of frequency divider here we can use this at those places where we require lower frequent to operate the functionality. These frequency dividers automatically adjust for making it area efficient. All modules will be designed using VHDL and implemented on Xilinx FPGA development board.

Key words: Receiver, Transmitter, FPGA, UART etc.

I. INTRODUCTION

Abbreviation for Universal Asynchronous Receiver-Transmitter, UART is a chip used to manage computer serial ports, disk drive interrupts, screen refresh cycles, and any other device that requires timing. In this proposed work main focused is to reduce the algorithm technique for designing the both part receiver and transmitter With a serial port transmission, the UART converts the bytes into serial bits and transmits those bits through an asynchronous transmission, separate out the start and stop bits for each character. Below is a listing of various UART chips. The 16550 chip series is the most commonly used UART. The Universal Asynchronous Receiver Transmitter (UART) is a popular and widely-used device for data communication in the field of telecommunication. A UART is a microchip with programming that controls a computer's interface to its attached serial devices. Specifically, it provides the computer with the RS-232 Data Terminal Equipment (Data Terminal Equipment) interface so that it can communicate to and exchange data with modems and other serial devices. More advanced UARTs provide some amount of buffer of data so that the computer and serial devices data streams remain coordinated. The most recent UART, the 16550, has a 16-byte buffer that can get filled before the computer's processor needs to handle the data. The original UART was the 8250. If you purchase an internal modem today, it probably includes a 16550 UART (although you should ask when you buy it). According to modem manufacturer robotics, external modems do not include a UART. For an older computer, we as user want to add an internal 16550 to get the most out of the external modem.

A Universal Asynchronous Receiver/Transmitter is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. The universal designation indicates that the data format and transmission speeds are configurable. The electric signaling levels and methods such as differential signaling are handled by a driver circuit external to the UART. A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. UARTs are now commonly Included in microcontrollers

The Universal Asynchronous Receiver/Transmitter (UART) takes bytes of data and pl-the destination, a second UART re-assembles the bits into complete bytes.

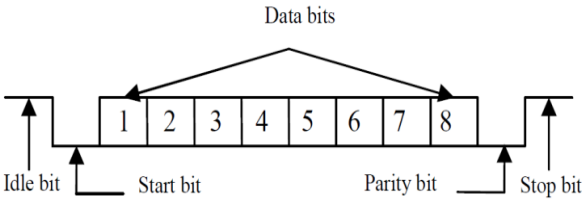

Fig. 1: UART Frame Format

Each UART contains a shift register, which is the fundamental method of Conversion between serial and parallel forms. Serial transmission of digital information (bits) through a single wire or other medium is less costly than parallel transmission through multiple wires. The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels. External signals may be of many different forms. Examples of standards for voltage signaling are RS-232, RS-422 andRS-485 from the EIA. Historically, current (in current loops) was used in telegraph circuits.

II. LITERATURE SURVEY

The UART is the use of the serial communication protocol, which permits the full duplex communication in serial link. They design the hardware implementation of a high speed & competent UART using Field Programmable Gate Array. The UART consists of three components, receiver, transmitter & baud rate generator which is also frequency divider. They simulated on Modelsim SE 10.0a and design by using Verilog description language which has been synthesized on FPGA kits like as Spartan3 & Virtex4. After analyzing the comparative analysis conclude that there is a difference in between the number of slices, LUTs and the maximum frequency. The results are quite stable and reliable and have great flexibility with high integration. If we use FIFO in making the UART our design becomes more flexible, stable and reliable which provides highest bps rate [1]. A UART is a full duplex receiver and transmitter. It is the chip with programming that controls a computer's an interface to its connected serial devices. It manages the transmission between serial and parallel data. The whole process of serial transmission is operating on the principle of the shift register. In data transmission through the UART,

once the baud-rate has been originated, both the transmitter & the receiver's internal clock are set to the identical frequency. [2] Tenure is concerned, developing a serial communication protocol including bit synchronization, automatic baud rate detection with selection and bus, frequency division according to the input clock. All modules are simulated on Xilinx ise [3]. Their work presents design method of asynchronous FIFO and structure of controller. This controller is designed with FIFO circuit block and UART (universal asynchronous receiver transmitter) circuit block with in FPGA to implement communication in modern complex control systems quickly and effectively. This controller can be used to implement communication when master equipment and slaver equipment are set at different baud rate. It can also be used to reduce synchronization error between sub systems in a system with several sub systems. The controller is reconfigurable and scalable. Problem with large area.[4] This work proposes an integrated architecture for a UART module to be used with MIMO-OFDM hardware platform, the purpose of this module is to enable the communication between Matlab and FPGA board. Problem with design complexity [5].

III. DESIGN TECHNIQUE

Figure 2 represents the general architecture of UART. This module is divided in sub modules like receiver, transmitter and baud rate generator etc.

A. Receiver

All operations of the UART hardware are controlled by a clock signal which runs at a multiple of the data rate. For example, each data bit may be as long as 16 clock pulses. The receiver tests the state of the incoming signal on each clock pulse, looking for the beginning of the start bit. If the apparent start bit lasts at least one-half of the bit time, it is valid and signals the start of a new character. If not, the duplicate pulse is ignored. After waiting a further bit time, the state of the line is again sampled and the resulting level clocked into a shift register. After the required number of bit periods for the character length (5 to 8 bits, typically) have elapsed, the contents of the shift register is made available (in parallel fashion) to the receiving system. The UART will set a flag indicating new data is available, and may also generate a processor interrupt to request that the host processor transfers the received data.

```
i/o
-- RX state registers update at each CLK, and RESET
reg_process: process (clk,reset) is
begin
if reset = '1' then
rx_state.fsm_state <= idle;
RX FSM: updates rx_state_next from rx_state and inputs.
rx_process: process (rx_state,sample,rx,rx_state_next.nbits,
rx_state_next.bits ) is
begin
case rx_state.fsm_state is
rx_state_next.counter <= (others => '0');
rx_state_next.bits <= (others => '0');
rx_state_next.nbits <= (others => '0');
rx_state_next.enable <= '0';
if rx = '0' then
rx_state_next.counter <= (others => '0');
rx_state_next.bits <= (others => '0');
rx_state_next.nbits <= (others => '0');
rx_state_next.enable <= '0';
if rx = '0' then
-- start a new byte
 -- start a new byte
rx_state.bits <= (others => '0');
rx_state.nbits <= (others => '0');
rx_state.enable <= '0';
entity basic_uart is
port (
-- Client interface
rx_data: out std_logic_vector(7 downto 0); -- received byte
rx_enable: out std_logic;
-- validates received byte (1 system clock spike)
-- Physical interface
rx,sample, reset    , clk: in std_logic
for the receiver port UART entity when the data is 7down to
0 it received 1 system clock spike architecture Behavioral of
basic_uart is type fsm_state_t is (idle, active); -- common to
both RX and TX FSM
type rx_state_t is  record
fsm_state: fsm_state_t;                -- FSM state
counter: std_logic_vector(3 downto 0); -- tick count
bits: std_logic_vector(7 downto 0);    -- last 8 received
bits……..
When the bits down from 7 to 0 it receive last 8 bit when
down to 3 to 0 it receive single bit with a new byte.
signal rx_state,rx_state_next: rx_state_t;
if reset = '1' then
rx_state.fsm_state <= idle;
rx_state.bits <= (others => '0');
if the state machine is the idle the state come 0.
when idle =>
rx_state_next.counter <= (others => '0');
rx_state_next.bits <= (others => '0');……..
-- keep idle
rx_state_next.fsm_state <= idle;
end if;
if  the state machine is the goes 0 it go to next bit and enable
the next stage data .
if rx_state.counter = 8 then
-- sample next RX bit (at the middle of the counter cycle)
if rx_state.nbits = 9 then
RX output
rx_output: process (rx_state) is
begin
rx_enable <= rx_state.enable;
rx_data <= rx_state.bits;
```

when output process begin the data of n bit is enable means state enable data copied to the state bit.

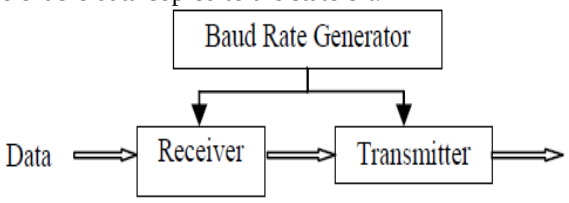

Fig. 2: UART Module

B. Transmitter

Transmission operation is simpler under the control of the transmitting system. As soon as data is deposited in the shift

register after completion of the previous character, the UART hardware generates a start bit, shifts the required number of data bits out to the line, generates and attached the parity bit (if used), and appends the stop bits.

```
entity basic_uart is
port (
-- Client interface
tx_data: in std_logic_vector(7 downto 0);  -- byte to send
tx_enable,sample, clk,reset: in std_logic;                    --
```

validates byte to send if tx_ready is '1' if the logic vector goes down 7 to 0 the clk reset and validates bytes ready is 1.

```
Architecture  Behavioral of basic_uart is type fsm_state_t is
(idle, active); -- common to both RX and TX FSM
type tx_state_t is
bits: std_logic_vector(8 downto 0); -- bits to emit, includes
start bit
TX state registers update at each CLK, and RESET
reg_process: process (clk,reset) is.
if reset = '1' then
tx_state.fsm_state <= idle;
tx_state.bits <= (others => '1');
tx_state.nbits <= (others => '0');
tx_state.ready <= '1';
elsif rising_edge(clk) then
tx_state <= tx_state_next;
end if;
```

if the idle state bit is shifted to 0 and 1 if rising to clk 1 state shift to next level.

Since transmission of a single character may take a long time relative to CPU speeds, the UART will maintain a flag showing busy status so that the host system does not deposit a new character for transmission until the previous one has been completed; this may also be done with an interrupt. Since full-duplex operation requires characters to be sent and received at the same time, UARTs use two different shift registers for transmitted characters and received characters.

C. Baud Rate

Bit rate is a measure of the number of data bit (that"s 0"sand 1"s) transmitted in one second. Baud rate by definition means the number of times a signal in a communications channel changes state. Baud means state changes of the line per second. Baud rate refers to the number of signal or symbols changes that occur per second. A symbol is one of several voltage, frequency or phase changes. NRZ binary has two symbols represent voltage level. Baud Rate represents the number of bits that are actually being sent over the media, not the amount of data that is actually moved from one UART device to the other. The Baud count includes the overhead bits Start, Stop and Parity that are generated by the sending UART and removed by the receiving UART. This means that seven-bit words of data actually take 10 bits to be completely transmitted.

BAUD RATE = NO OF BITS TRANSMITTED/ RECEIVER PER SECOND.

IV. RESULT & SIMULATION

A Universal Asynchronous Receiver/Transmitter is a piece of computer hardware that translates data between parallel and serial forms. UARTs are commonly used in conjunction with communication standards such as EIA, RS-232, RS-422 or RS-485. The universal designation indicates that the data format and transmission speeds are configurable. The electric signaling levels and methods such as differential signaling are handled by a driver circuit external to the UART. A UART is usually an individual (or part of an) integrated circuit used for serial communications over a computer or peripheral device serial port. UARTs are now commonly Included in microcontrollers. A dual UART, or DUART, combines two UARTs into a single chip.

A. UART Reciever Port

For simulation of this port we use **Xilinx ISE 14.7** software. We are giving the input rx , sample , clock , reset after giving all the input we simulate the receiver port the output is shown below.

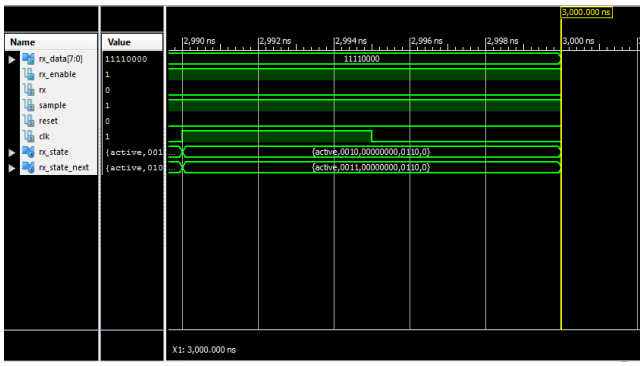

Fig. 3:

Delay-After doing the simulation process we got our synthesis report. There is 0.039ns delay.

Power used- After generating the map file by same software. We insert this map file in to the Xilinx Sparten3E Spreadsheet we get 0.032W power consumed in the circuit.

B. UART Transmitter Port

In the Transmitter port we gave eight input data port & enable the transmitter the we give sample, clock, and reset value. After simulation we got desired output that is shown below.

Fig. 4:

After simulation process we get synthesis report the delay is 0.039ns.To find power consumption in the circuit we use Xilinx Sparten3E spreadsheet. The power used by the circuit is 0.032W.

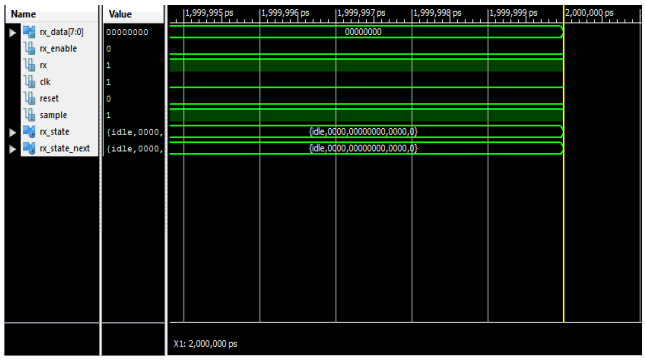

Fig. 5:

In the serial port we gave input values and after simulation process is done. The result is shown below. This simulation is done by using **Xilinx ISE 14.7** software.

Synthesis report shows the delay in the above serial port is 0.037ns.Power consumption in the simulation process is 0.031W.

S.NO	Transmitter Delay/power	Receiver Delay/power	i/o delay/power
1	0.039ns/ 0.032w	0.039ns/ 0.032	0.037ns/ 0.031

Table 1:

V. CONCLUSION

From result of proposed work it is concluded that the design uses VHDL/Verilog language to acquire the modules of universal asynchronous receiver transmitter. By Using the Xilinx software, FPGA board to complete simulation. The results are stable and reliable according to binary information. The outcome is feasible and efficient with power reduction and also area reduction. Especially in the field off electronic design technology has recently become widely used, this design shows great significance and can be used in various applications.

REFERENCES

[1] Miss. Ruchika Longadge1, Prof. Mithilesh Mahindra2, Prof. Vidya Dahake3," Design and Implementation of UART" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 5, May 2015.

[2] Parul Shrivastava1, Prof. Sourabh Sharma2, "Design and simulation of 16 Bit UART Serial Communication Module Based on VHDL" International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 4, April 2014.

[3] Manchineni Vijay Kumar, Suresh Angadi," Study of Uart Transimitter in Microcontroler" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-2, December 2012.

[4] Neha R Laddha, A P Thakare."A Review on Serial Communication by UART" International Journal of Advanced Research in Computer Science and Software Engineering 3(1), January - 2013, pp. 366-369.

[5] Sherif Moussa,, 2Ahmed M.Abdel Razik, 3Adel Omar Dahmane, 4 Habib Hamam," FPGA Implementation Platform for MIMO-OFDM Based on UART" Vol. 4, No. 4 April 2013 ISSN 2079-8407 Journal of Emerging Trends in Computing and Information Sciences.

[6] Mohd Yamani Idna Idris, Mashkuri Yaacob, Zaidi Razak, "A VHDL implementation of UART design with BIST capability" Malaysian Journal of Computer Science, Vol. 19 (1), 2006.

[7] Shouqian Yu,Lili Yi ,Weihai Chen,Zhaojin Wen," Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA" 1-4244-0737-0/07/$20.00 c 2007 IEEE.

[8] Huimei Yuan, Junyou Yang,Peipei Pan," Optimized Design of UART IP Soft Core based on DMA Mode" 978-1-4244-5046-6/10/$26.00 c 2010 IEEE.

[9] Naresh patel, Vatsalkumar Patel and Vikaskumar Patel "VHDL Implementation of UART with Status Register" 2012 International Conference on Communication Systems and Network Technologies (IEEE).

[10] Hazim Kamal Ansari, Asad Suhail Farooqi," Design Of High Speed Uart For Programming FPGA" International Journal Of Engineering And Computer Science Volume1 Issue 1 Oct 2012.

[11] Amit Bhanwal, mayank kumar, Yogendera kumar," FPGA based Design of Low Power Reconfigurable Router for Network on Chip (NoC)", International Conference on Computing, Communication and Automation (ICCCA2015).

