# Synthesis & FPGA Implementation of UART IP Soft Core

**Sohini Chatterjee[1] Sona Gava[2] Shreya Sharan[3] Sonam Sinha[4] Subir Maity[5]**
[1,2,3,4] B. Tech Student [5]Assistant Professor
[1,2,3,4]Department of Electronics & Telecommunication [5]School of Electronics Engg.
[1,2,3,4,5]KIIT University, Bhubaneswar-751024, Odisha, India

*Abstract*—this paper presents synthesis and hardware implementation of fully functional Universal Asynchronous Receiver Transmitter Intellectual Property core using XILINX SPARTAN-3 XC3S400 series FPGA. The UART soft core module consists of a transmitter along with baud rate generator and a receiver module with false start bit detection features. This has been implemented using VERILOG hardware description language and synthesized using XILINX ISE development tools. All behavioral simulation of UART module performed using MODELSIM simulator. After successful FPGA implementation transmitter and receiver module was tested by connecting FPGA board with Hyper Terminal software via RS232 interface at a data speed of 9.6 kbps.

*Key words:* VERILOG, UART, XILINX, MODELSIM, Baud rate, Soft core.

## I. INTRODUCTION

An IP (intellectual property) core is a block of logic or data that is used in making a field programmable gate array ( FPGA ) or application-specific integrated circuit ( ASIC ) for a product. As essential elements of design reuse, IP cores are part of the growing electronic design automation (EDA) industry trend towards repeated use of previously designed components. Ideally, an IP core should be entirely portable - that is, must be easy to insert into any vendor technology or design methodology. Universal Asynchronous Receiver /Transmitter (UART s), central processing units (CPU s), Ethernet controllers, and PCI interfaces are all examples of IP cores.

IP cores fall into one of three categories: hard cores, firm cores, or soft cores [4]. Hard cores are physical manifestations of the IP design. These are best for plug-and-play applications, and are less portable and flexible than the other two types of cores. Like the hard cores, firm (sometimes called semi-hard) cores also carry placement data but are configurable to various applications. The most flexible of the three, soft cores exist either as a net-list or hardware description language (HDL) code.

A UART is a Universal Asynchronous Receiver-Transmitter, which is used to communicate between two devices [1, 2, 4]. Most computers and microcontrollers include one or more serial data ports to communicate with other serial I/O devices, such as keyboards and serial printers. Serial ports are also used to communicate between two computers using a UART in each computer and a crossover cable, which connects the transmitter (TxD) of one UART to the receiver (RxD) of the other, and vice versa. A common ground (GND) wire connects both computers to a common negative voltage source [8].

UART is to sample the incoming serial data at the right time to precisely capture the binary stream. UART serves as a medium to send information using a minimum number of wires. The data is sent bitwise in a serial fashion, without a clock signal [8]. The main function of a UART is the conversion of parallel-to-serial when transmitting and serial to-parallel when receiving. The fact that a clock signal is not sent with the data complicates the design of a UART.

The following are the components required for this serial to parallel and vice versa Communication:

1) BAUD RATE GENERATOR: A fast clock is used to sample the binary stream multiple times for each data bit. It is basically a clock divider.
2) TRANSMITTER: When transmitting, the UART receives the data in parallel from the application, and sends it serially on the TxD pin.
3) RECEIVER: When receiving, the UART receives the data serially on the RxD pin, and provides the parallel data to the application.

## II. BAUD RATE GENERATOR

Basically it is a clock divider. The main clock frequency is divided to generate specific data transfer rate. In this design-

Main clock: 50 MHz, Required Baud rate: 9.6kbps

Divisor: (50,000/9.6) = 5208

It uses a counter & when count=5208, a pulse is generated at output. Verilog HDL is used to design Baud rate generator. MODELSIM [6] simulation result shown in Fig.1.

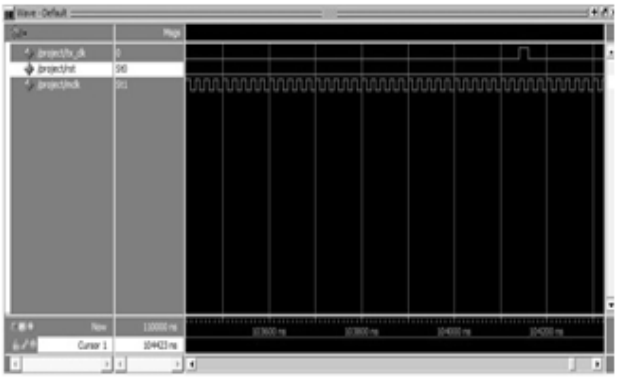

Fig. 1: Simulation Result of Baud Rate Generator

## III. UART TRANSMITTER

The transmitter was designed by five states. Those five states are namely- IDLE, START, SEND, STOP1, and STOP2 as shown in Fig. 2.

All rights reserved by www.ijsrd.com 

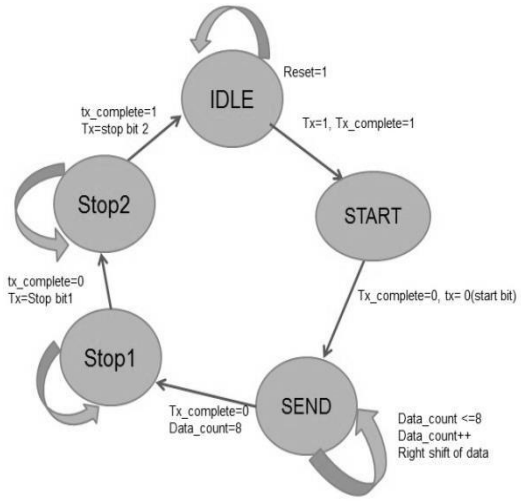


Fig. 2: FSM for UART Transmitter

In this design 1-start bit, 2 stop bit and no parity bits were used to transmit 8 bit data. Transmission started with a START bit which is basically HIGH to LOW transition. Then 8 bit data are transmitted and after sending 8 bits two STOP bits were transmitted via TX line. STOP bits are HIGH. After completion of transmission, TX line becomes HIGH which indicates Transmitter is in IDLE or REST condition. Transition from one state to another state occurs at rising edge of BAUD RATE clock pulse.

## IV. UART RECEIVER

In this design, Transmitted data bits were sampled 16 times by using receiver side BAUD RATE clock which is 16 times faster than Transmitter side clock pulse. Each data bit is detected at its mid position and stored in a temporary register. When all 8 bits are detected, then 8 bit parallel data is available at UART receiver output.

The receiver was designed by five states. Receiver side FSM diagram given in Fig. 3.

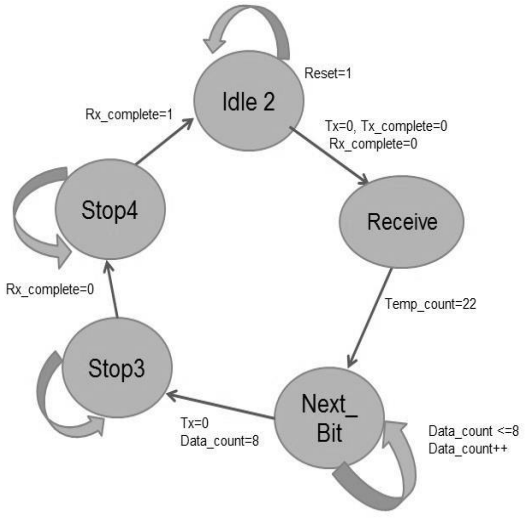


Fig. 3: FSM for UART Receiver

*False Start Bit Detection:* When transmitter is in IDLE or REST condition TX line is HIGH which is connected with RX line of UART receiver. In conventional UART design data bit detection in the receiver side starts with START bit which is HIGH to LOW transition. Whenever there is one HIGH to LOW transition in TX line receiver recognizes it as beginning of a new transmission. But due to environmental noise or man-made noise such as electric sparking etc., there may be false short duration HIGH to LOW transition which is detected by receiver which then treats the next data bits as incoming data. To avoid this problem, one simple false start bit detection scheme is introduced in this design. Detection of START bit started with HIGH to LOW transition and it is already sampled 16 times by UART receiver clock. At $5^{th}$, $8^{th}$ and $11^{th}$ clock pulse, data sample of START bit are collected and those bits are logically 'AND'ed. If output is Logic 1 then there was no false transition. If the result is Logic 0, then receiver detects it as a false transition and ignore it.

## V. FPGA IMPLEMENTATION

After successful behavioral simulation of UART Module using MODELSIM simulator, it was synthesized and mapped into XILINX SPARTAN-3 XC3S400 series FPGA using XILINX ISE [5] design suite. Transmitter module is connected with computer's serial port via RS232 interface and incoming transmission was successfully tested using popular communication program named 'HYPER TERMINAL' with data transfer speed of 9600 bps, 2 stop bit, no parity, no flow control mechanism. To test receiver module, ASCII character was transmitted from HYPER TERMINAL and displayed on the LEDs connected at 8 bit data output pin of UART receiver and data bits were received correctly.

## VI. RESULTS & DISCUSSION

Fig. 4 shows MODELSIM simulation result of complete UART transmitter and receiver for the transmission of 8bit digital data- '10110010'. In SPARTAN-3 XC3S400 series FPGA. Total 63 slices and 113 LUTs were utilized to implement complete UART IP soft core.

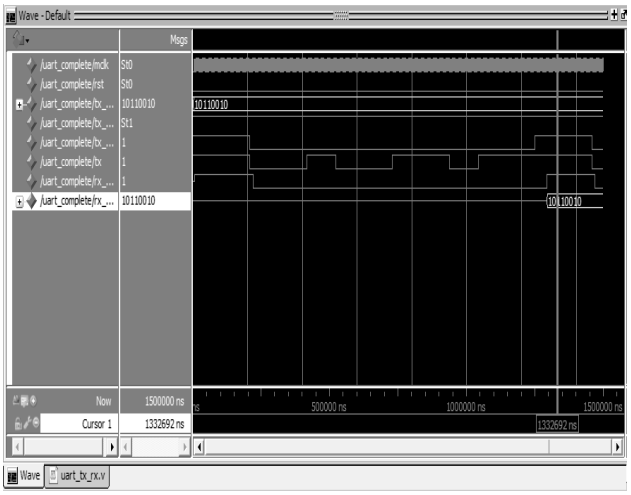


Fig. 4: Simulation of UART Transmitter with Receiver

## VII. CONCLUSION

One complete UART module was successfully implemented in XILINX SPARTAN-3 XC3S400 series FPGA and tested by connecting board with PC Hyper Terminal. Receiver module includes false start bit detection features. In future FIFO can be introduced to enhance data processing

All rights reserved by *www.ijsrd.com* 

capability and to introduce parallel processing concept in UART module. Also parity bit may be included to detect error present in receiving bits.

REFERENCES

[1] Yu, Chen, "Implementation of a Multi-channel UART Controller Based on FIFO Technique and FPGA" Industrial Electronics and Applications, 2007. ICIEA 2007. 2nd IEEE Conference, Page(s): 2633 – 2638
[2] Fang Yi-yuan , Chen Xue-jun " Design and Simulation of UART Serial Communication Module Based on VHDL ", 3rd International Workshop on Intelligent Systems and Applications (ISA), 2011, Page(s): 1 - 4
[3] M.Kaur, R.Mittal," Fpga Implementation & Design of Micro UART with Different Baud Rates" Journal of Information Systems and Communication, Volume 3, Issue 1, 2012, pp.-41-44
[4] 16750 UART core, http:// opencores.org/ project/ Uart 16750"
[5] XILINX ISE Tutorial "download.xilinx.com/ …ect/ ise9_tutorials/ ise9tut.pdf"
[6] Modelsim Tutorial, "www.mentor.com"
[7] PC16550D tutorial," http://www.ti.com/general /docs/lit getliterature.tsp? literatureNumber = snls378b"
[8] Wikipedia, The Free Encyclopedia, Universal asynchronous receiver/transmitter, "http://en.wikipedia.org/wiki/Universal_asynchronous_ receiver/transmitter"

All rights reserved by *www.ijsrd.com* 