

UART_RS232 PROTOCOL USING FPGA

Vikas Pratap Singh and Sanjay Kumar Singh. , 1' Dr Deepak kourav

Research Scholar, IIMT College of Engineering, Greater Noida, UP

Associate Professor, IIMT College of Engineering, Greater Noida, UP

ABSTRACT

The implementation of RS232 through the Field Programmable Gate Array (FPGA) establish Communication Protocol is presented in this paper. For this Project we have used an average of 115200 bauds (maximum speed you can go) at high speed. Binary information is converted to ASCII "American Standard Code for Information Exchange" and sent to FPGA from PC using the protocol communication. TERATERM software used. The ASCII character is displayed in the TERATERM terminal.

INDEX TERMS: FPGA, TERATERM, RS232, Verilog, System Verilog. Etc.

1. INTRODUCTION

FPGAs are built with a system of logical blocks such as check tables, gates, flip-flops, and memory blocks connected to a large communication network. FPGA can be rearranged in a logical circuit using hardware definition language such as VHDL or Verilog. The FPGA design allows for many types of logic designs for realtime application.

Binary data is converted to the ASCII character and the RS232 interface is used for data sent to the board and the results are displayed in the forum. This paper provides a better idea of how to use the RS232 Protocol on the FPGA board. As data is in ASCII format. The hardware required is small and the processing is fast. Due to binary data and ASCII does not need to use a square root algorithm in Verilog or VHDL language.

2. NECESSARY NEEDS AND YOUR USE

2.1. RS232 ACT

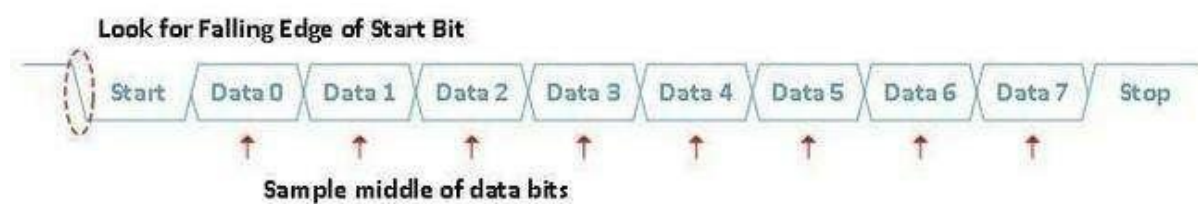
The RS-232 serial protocol is a standard protocol used in asynchronous serial communication. For any unusual encounter, the first thing you need to know is that in time you have to sample (look) data. If you do not sample data in a timely manner, you may see incorrect data. To get your data right, the sender and recipient must agree on a baud rate. The baud rate is the rate at which data is transmitted. 115200 baud mean 115200 bits per second. The code below uses a parameter in Verilog to determine how many clock cycles in each item. This is how the baud rate is determined.

FPGA remains a queue line. As soon as it sees a line change from top to bottom, it knows that UART data name is coming. This first change shows a small start. As soon as the first slow start is detected, the FPGA waits half

the time. This ensures that within the data it receives samples. From there on, FPGA needs to wait one minute (as specified in baud rate) and sample all data. The figure below shows how the UART receiver works within FPGA. The first is found on the falling edge of the serial data line. This indicates a small start. The FPGA then waits until the middle of the initial data slowly and sample the data. It does this on all eight bits of data.

In this project we used one Start Bit, one Stop Bit, eight Data Bits, and no rating. Note that the transfer modules below both have a TxD_start signal. This is used to install a tri-state buffer for half-duplex connections. It is for your specific project needs if you want to create a half-duplex UART or a complete UART. The code below will work for both!

RS232 is a roaming form for different communication data



RS232 asynchronous communication data wave form

3. BAUD RATE CALCULATION

Baud rate is the rate of transmission speed on a unique connection representing the number of bits actually sent to the serial link.

Initial Spartan 3E Kit Operating Frequency = 12 MHz

Baud rate = 9.6 khz = 9000 Hz

Calculate $12 \text{ MHz} / 9000 = 625$

At 115200 bauds, each item stays $(1/115200) = 8.7\text{s}$. If you transfer 8-bits data, that stays $8 \times 8.7\mu\text{s} = 69\text{s}$. But each byte requires extra start and minimal stops, so you need $10 \times 8.7\mu\text{s} = 87\mu\text{s}$. That translates to a top speed of 11.5KBytes per second.

With 115200 bauds, some PCs with buggy chips require a "long" stop (1.5 or 2 bits long ...) which slows down to 10.5KBytes per second.

4. IMPLEMENTATION OF THE RS232 PROTOTPE FOR VERILOG USE

We create an "async transmitter" with set parameters: 8 pieces of data, 2 bits, inequality.

It works like this:

The transmitter takes 8-bits data inside FPGA and makes it serial (starting when authorizing "TxD_start" signal).

The "busy" signal is highlighted while the transfer is taking place (signal "TxD_start" is ignored immediately).

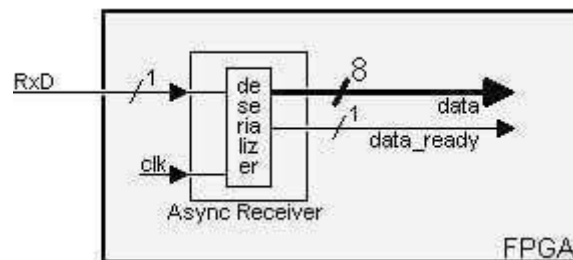
Creating an "async receiver":

Our implementation is as follows:

The module integrates data from the RxD line as it comes.

As the byte is detected, it appears on the "data" bus. Once a complete byte has been received, "data ready" is verified with one clock.

Note that "data" is only valid if authorized "data ready". The rest of the time, don't use it as new exchange data may appear.



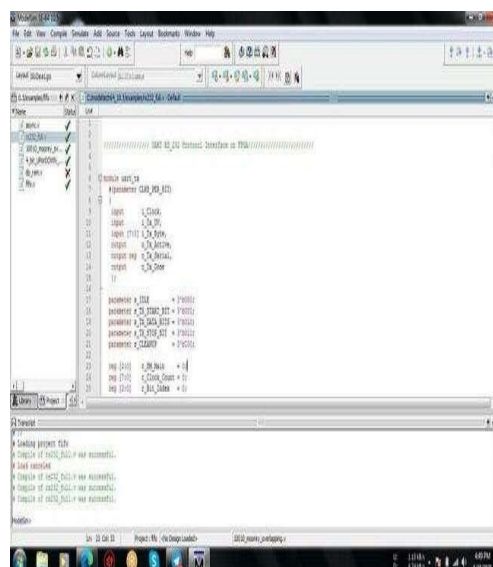
5. USE AND MEASUREMENT RESULTS

Here we have used binary format data. While compiling HDL code, using internal FPGA RAM instead of memory on board, binary data is transferred to the Spartan 3 board via RS232 to FPGA and displays the ASCII character in the terminal.

Hardware Result:



Software Result:



6. CONCLUSION

Converts binary data into ASCII code. With the help of FPGA, we simply change the details and get the details in baht of 115200 (high speed). We have sent binary data that will be converted to ASCII character encoded and the result displayed on a PC. However, the RS232 serial connection is easy to use but the transmission speed is very low compared to other communication methods.

REFERENCE

1. Rashmi, Mukesh Kumar, RohiniSaxena “Algorithm and technique on various edge detection,” An International Journal (SIPIJ) Vol.4, No.3, June 2013.
2. Mr. Manoj K. Vairalkar, Prof. S.U. Nimbhorkar, “Edge detection of images using sobeloperaotr,” International Journal of Emerging Technology and Advance Engineering, Vol.2, January 2012.
3. Han Xiaoru, Gao Yudong, “Design and Implementation of the Universal RS232-GPIBInterface,”Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on.
4. <https://en.wikipedia.org/wiki/Verilog>
5. [https://rashmistudents.yolasite.com/resources/Verilog%20HDL%20-%20Samir %20Palnitkar.pdf](https://rashmistudents.yolasite.com/resources/Verilog%20HDL%20-%20Samir%20Palnitkar.pdf) 6. Verilog – Wikipedia
7. Fpga4fun.com - where FPGAs are fun.
8. FPGA Projects, Verilog Projects, VHDL Projects - FPGA4student.com
9. https://www.xilinx.com/support/documentation/data_sheets/ds312
10. <https://www.instructables.com/Design-of-a-Simple-VGA-Controller>