

LAB ASSIGNMENT 8

Title: Asynchronous Serial Receiver

Learning Objective:

Learn how to receive serial information without a shared clock.

Specification:

Design asynchronous serial receiver with baud rate = 9600, 8 data bits and no parity bits. Connect this to the micro USB port of the BASYS 3 board. Use gtkterm on PC to test and demonstrate.

Design:

Inputs :

- rx_in : input given to the receiver by the keyboard (Serial Input)
- clk : On-Board clock given by the BASYS3 board

Output :

- rx_reg : register that stores the serially read data from the rx_in input (Parallel Output)

Implementation:

The ASM for a serial receiver discussed in the class was used to implement the serial receiver in VHDL.

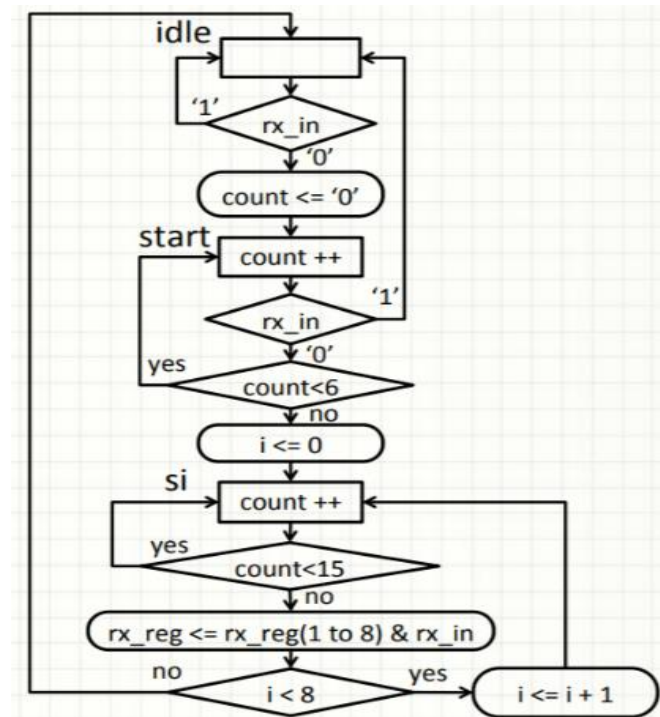
Baud Rate: 9600 bits per second

Receiver clock frequency: 9600×16 Hz

The receiver clock was made by dividing the on-board clock by an appropriate value to get the clock of the required frequency.

States Used: Three states were used in the implementation as shown in the ASM diagram.

1. Idle : when the rx_in input is set to 1 and the receiver is not reading.
2. Start : When rx_in input is set to 0 for the first time after 1. This state remains for at most 8 clock cycles only if the rx_in input was invariantly 0 in all the clock cycles otherwise the state goes back to idle
3. Read : After reading a valid start bit the state changes from start to read. In this state we will be reading the sequentially given input till 8 bits and then assign it to the rx_out output so that it becomes a serial in parallel out register.



State Transitions involved :

1. When rx_in input is 1 invariantly then the receiver is not reading and it is in IDLE state.
2. When a rx_in changes to 0 after being 1 this indicates that the transmitter will be sending data in the next 8 bits. This intermediate state that indicates that the data transfer will be starting is the START state.
3. To make sure that the start state comes only when the transmitter gives the start bit and not when 0 comes due to some noise the start bit 0 is read for 8 consecutive cycles and only then the START state changes to the READ state and the reading of the data for the rx_in input starts.
4. When in READ state the receiver reads the rx_in input and stores it sequentially in the register one by one after 16 clock cycles each so that bit reading takes place in between the edges on the rx_in input.

Giving output to LEDs:

The rx_out output is given to 8 LEDs on the BASYS3 board which display whether the i^{th} read bit is set or not. If the read bit is set then the LED glows otherwise it remains off.