LAB ASSIGNMENT 9

Title : Asynchronous Serial Transmitter Learning

Objective: Learn how to transmit serial information without a shared clock.

Specification: Design asynchronous serial transmitter matching the receiver designed in the previous assignment.

Inputs:

- rx_in: Taken serially using gtkterm
- clk: On-board clock
- reset: Taken from a push button

Outputs:

- rx_reg: To show the received (and transmitted) data on the LEDs
- tx_out: Serially transmitted data

Implementation Logic:

Hand-Shake Implementation : In this assignment we implemented a handshake cycle between the receiver and the transmitter in which the data is given serially to the receiver whose parallel output was given as parallel input to the transmitter that transmitted the data serially back to the PC through which we had

received the data earlier hence giving a handshake cycle .

Logic: We created a transmitter with baud rate and 9600 bps having parallel data input (as a signal tx_in) and serial output that is the output of the circuit i.e. tx_out.
The serially coming input was stored in an internal register(signal) rx_reg which is seen as the parallel output of the receiver and data of this register was given parallelly to the transmitter input i.e. another internal register(signal) tx_in.
Data through the tx_in register was transmitted serially to the tx_out output with a clock of frequency 9600 Hz to maintain the baud rate 9600.

Error in previous submission: Transmission was repeating again and again and the code needed some correction.

Error Correction: This was caused because the transmitter output was not set to 1 when transmitter was not in transmitting state. Now this was handled by adding another state tx1 in the receiver which remained for 18 clock cycles (of the clock of frequency 9600*16Hz) so that at least 1 rising edge of the clock of frequency 9600Hz comes in the this period and the

transmitter goes to transmitting state then the receiver again goes back to the idle state.