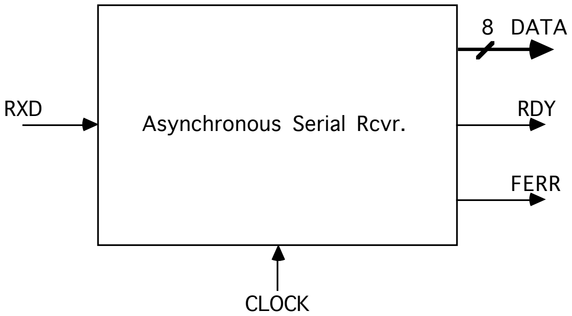
ECE 491 – Senior Design 1

Laboratory 5 – Asynchronous Serial Receiver

October 1, 2007

Goals

1. To develop an FPGA design of an Asynchronous Serial Receiver that converts serial data from an RS-232 interface to parallel data one byte at a time.
2. To gain additional skill designing and debugging complex digital circuits using Verilog and FPGAs.



Requirements

1. The receiver will include a clock input, a 1-bit serial data input **rxd**, an 8-bit **data** output, a 1-bit **rdy** output, and a 1-bit **ferr** ("framing error") output. As described earlier, data will appear on the **rxd** input beginning with a single start bit and ending with one or more stop bits.
2. The basic receiver circuit will operate at a clock rate that is 16 times the baud rate. It will operate by sampling the **rxd** input and waiting for a falling edge that signifies the beginning of a start bit. It will then wait until the middle of the start bit and re-sample **rxd** to verify that the input is still low. If not, it must ignore this "spurious start" and return to waiting for a falling edge. Otherwise, the circuit must assert a low on the **rdy** output and wait until the middle of each of the following successive bits and sample these inputs to assemble a parallel data value. Finally, the circuit must sample the **rxd** input during the middle of the stop bit to verify that the input is high. If not, a "framing error" has occurred and the circuit will assert the **ferr** output until the next time a start bit is detected. The **rdy** output will be asserted high when a full byte and stop bit have been successfully received and held high until a new start bit has been received.
3. The receiver design must be parameterized so that it can be configured to operate at different Baud rates (9600 baud default) given a 50Mhz clock input from the Spartan 3 Starter Kit board. Instructions for configuring your circuit at different baud rates must be included in header comment of your top-level Verilog file.
4. The receiver must be able to successfully receive two characters that have been transmitted with only a single stop bit between the last bit of the first character and the start bit of the second character.

5. To demonstrate its proper operation, you will interface your design to the 7-segment display on the Spartan 3 Starter Kit board to display a received byte as two hexadecimal digits. The **rdy** and **ferr** outputs will be displayed using LEDs.
6. The design must be verified using a self-checking testbench. This testbench will generate serial test data that simulates the transfer several different byte values, including `8'b01010101`, `8'b00110011`, and `8'b00001111`. It must check that successive bytes data are received correctly when separated by both a single stop bit and a longer “idle” period. It must also check that the **rdy** output is asserted properly as each byte is received and that the **ferr** output is asserted when the sampled stop bit is not a logic high.
7. A second testbench must be constructed to verify that the Asynchronous Serial Transmitter designed in Lab 4 can successfully transmit data to your new receiver circuit.
8. All Verilog code must follow the Coding Guidelines discussed in class.

Deliverables

1. Demonstration to Lab Instructor of successfully operating design as it receives characters from a PC running HyperTerminal. This demonstration must show both normal circuit operation and proper reaction to framing errors.
2. A short technical memorandum which describes (a) what was done, (b) what you learned, and (c) what difficulties you encountered. Include a block diagram of your design showing its major components.
3. Verilog listings of all files, including transmitter design and testbench.
4. Timing diagram printouts showing correct simulation of your receiver design, including test cases for spurious start bits, successful data reception, and framing errors.
5. Entries in your Lab Notebook documenting design ideas, the steps taken to create and debug your design, any pitfalls you encountered, and recorded data (if any). Each Lab Notebook entry should be dated and signed by all students in the lab group.

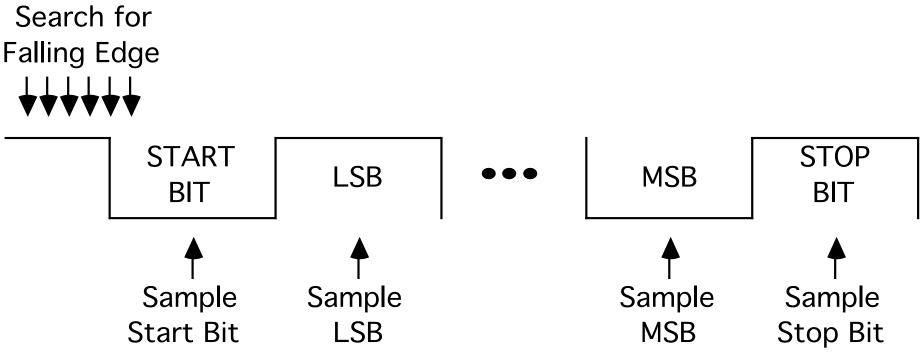
Background

In order to receive asynchronous serial data, a receiver circuit must somehow synchronize the incoming data. This is a form of *clock recovery*, where the receiver circuit attempts to synchronize itself with the clock of the transmitter circuit.

One way to accomplish clock recovery is to clock the receiver circuit at a known multiple of the transmission rate (typically 16X). When this is done, a receiver can be implemented that does the following:

1. Wait for the falling edge of the START bit.

2. Delay to the center of the START bit and sample the current data value. If the data value is not still asserted low, ignore the START bit and return to step 1.
3. Delay to the center of the next data bit. Sample and store the data value.
4. Repeat step 3 seven more times.
5. Delay to the center of the stop bit and sample. If the value is not correct, indicate a framing error condition.
6. Indicate that valid data is available.
7. Go back to step 1.


Search for
Falling Edge
START
BIT
LSB
MSB
STOP
BIT
Sample
Start Bit
Sample
LSB
Sample
MSB
Sample
Stop Bit