ECE 491 – Senior Design 1 Laboratory 5 – RS-232 Serial Receiver

September 28, 2006

Introduction

In this lab you will design and implement a serial receiver. It is more complex that the serial transmitter because it requires that you synchronize with an incoming signal and sample successive bits. Your design will probably consist of a shift register, one or more counters (used to generate delays), and a control unit that ties it all together.

Lab Preparation and Design

We will design the serial receiver assuming a clock rate that is 16 times the baud rate. The receiver circuit must implement the following steps:

- 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 the value of the data value and shift it into a shift register.
- 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 in the shift register.
- 7. Go back to step 1.

Your asynchronous serial receiver should have the following inputs and outputs:



RXD is the serial data input. DATA is the parallel data output. RDY should be asserted when a character has been received, and FERR should be asserted when a character has been received but the value of the stop bit was not correct (FERR and RDY should remain asserted until a new start bit is detected).

- Create a block diagram of your receiver design in terms of primitive building blocks (shift registers, logic blocks, control FSM, etc.) and bring it with you to lab. Include
- 2. Create a state diagram which describes the function of your receiver's control unit.
- 3. Use the block diagram and state diagram to write Verilog code for your receiver design. Follow the coding guidelines distributed in class –use one file for each module, include title blocks, and use the indentation style described in the guidelines. Bring your code with you to Lab.
- 4. Modify the clock divider you used in the previous lab to generate a clock with a 50% duty cycle (approximately) and a clock rate that is 16 times the 9600 baud data rate.
- 5. Modify the multiplexer/decoder circuit developed in Lab 2 to display two digits on the S3 board seven-segment display.
- 6. Modify the "s3board.v" top-level file for the S3 board to implement your receiver module and clock divider. Connect the receiver input to the RxD input from the RS-232 transceiver, and connect the ready and FERR outputs to LEDs. Connect the 8 bits of data to two the seven-segment display.

In the Lab

- 1. Write a testbench that generates the proper sequence of signals to test your receiver design and checks that it operates correctly. It should include test inputs that are
 - a. Generated using Verilog behavioral code.
 - b. Generated using an instantiation of your transmitter design from last week.

In both cases, the testbench should check that the values are correct and otherwise write an error message. Print the timing diagrams for your testbench for inclusion in your report.

2. Synthesize your complete receiver and download it into the S3 board. Use the serial cable and HyperTerminal to send characters to your interface and verify that the corresponding ASCII codes appear on the seven-segment display.

Report

For your lab report hand in the following:

- 1. A short technical memorandum which describes (a) what was done, (b) what you learned, and (c) what difficulties you encountered.
- 2. The block diagram of your serial receiver design.
- 3. The state diagram of your serial receiver control unit.
- 4. Verilog listings for all of your design files and your testbench file.
- 5. Simulator timing diagram printouts that show the proper operation of your design.