ECE 491 – Senior Design 1

Laboratory 7 – Manchester Code Receiver

October 25, 2007

Introduction

In this lab you will design a circuit that will receive data that has been transmitted using a Manchester Code. The receiver will decode *frames* that contain an arbitrary number of bytes of data. As you learned in the previous lab, the Manchester code transmits data serially over a single signal, which in the receiver is called `rxd`.

Figure 1 shows the block diagram of the Manchester decoder interface. It has a three inputs: `rxd` (serial data in), `reset`, and `clk`. We will assume that `clk` has a frequency that is 16 times the incoming data rate. The interface has four outputs: `cardet`, which is true whenever data is being received; `data`, which is used to output decoded data in 8-bit groups; `write`, which is used to write each byte data into a buffer (not included in your design) as it arrives; and `error`, which is true when an incorrect input is received (as described below).

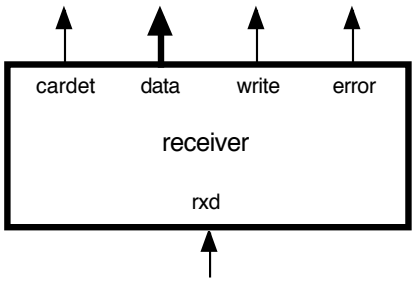

Figure 1 – Manchester Receiver Interface

When no data is being transmitted, the `rxd` input is a logic high and the receiver is *idle*. When a data frame arrives, it always begins with a *preamble* and *Start Frame Delimiter* (SFD) that is used to synchronize the receiver. The SFD is followed in the frame by multiple data bits. The frame ends when the `rxd` input returns to the idle state.

The SFD consists of a sequence of alternating 1's and 0's and ends with with two 1's. We will work with a preamble/SFD that includes three pairs of 1's and 0's followed by two 1's, as shown in Figure 2. However, your design must also support longer preambles as long as the SFD ends with two consecutive “1” bits.

After the SFD, the frame contains data bits transmitted byte by byte with the least significant bit transmitted first. Your receiver will decode the incoming bits as they arrive and convert the incoming bits into parallel bytes. Each time eight bits have been successfully decoded, they are transferred to an external buffer that is not part of your design. This is accomplished by asserting the converted byte on the DATA output while asserting the WRITE output high. Figure 2 shows the timing for this the first byte of a frame.

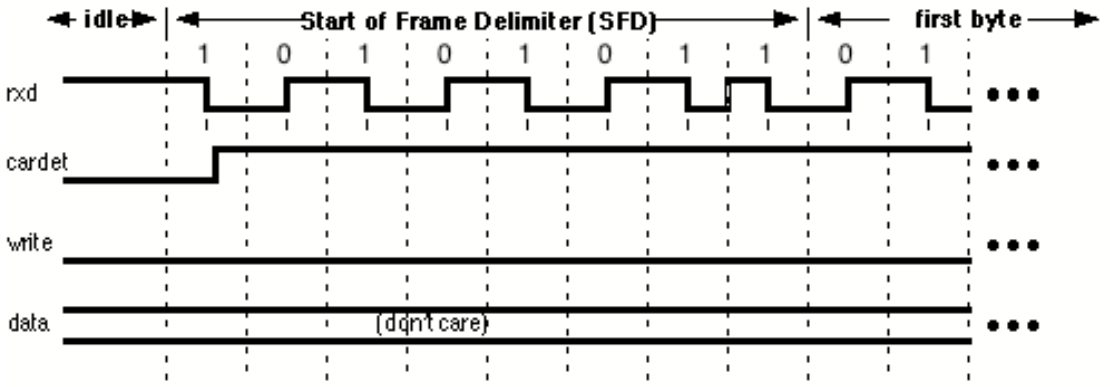

Figure 1 – Beginning of a Data Frame

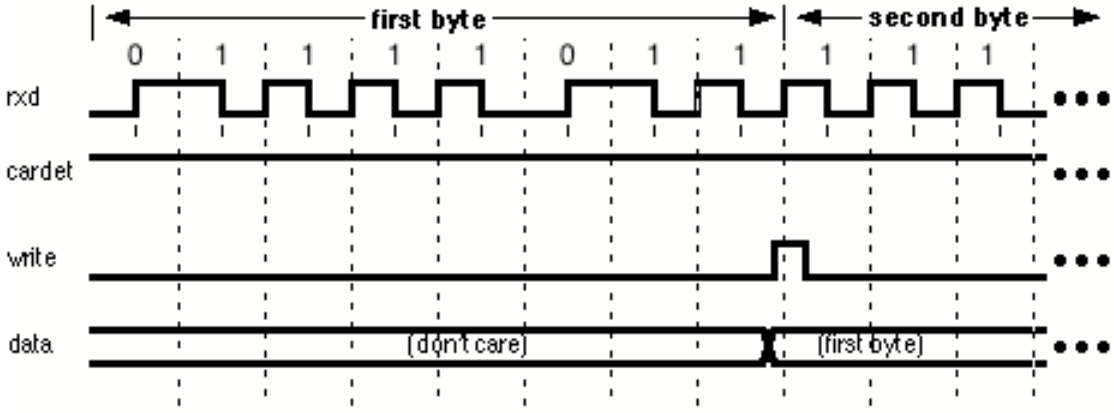

Figure 2 – First Byte of a Data Frame

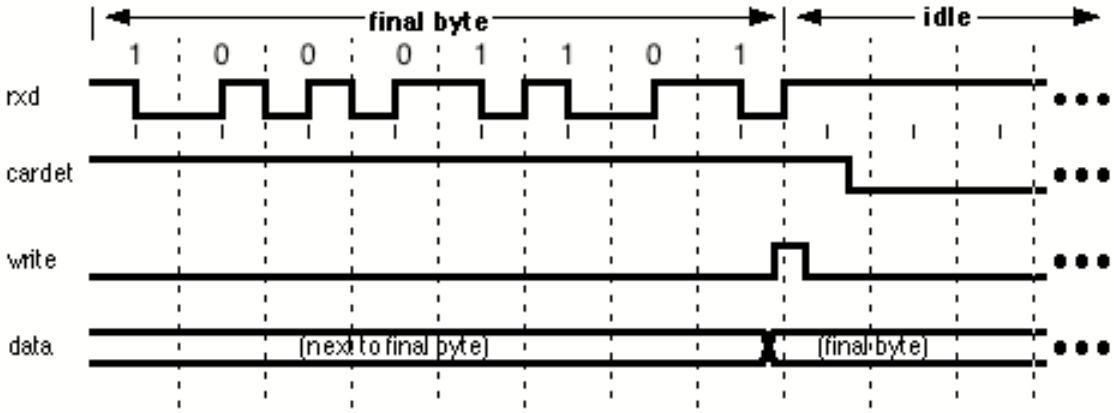

Figure 3 – Final Byte of a Data Frame

When the final byte of the data frame has been received, the `rxd` input returns to a logic high and the receiver asserts the `cardet` output low, as shown in Figure 3.

Requirements

1. The receiver design must implement the interface described in Figure 1 and in the previous section.
2. The receiver must be able to synchronize to an incoming frame that starts with a preamble and Start Frame Delimiter (SFD) consisting of a minimum of 6 alternating "1" and "0" bits followed two "1" bits. This includes a single-byte SFD consisting of the pattern "110101" transmitted least-significant bit first.
3. Once the end of the SFD is detected, the receiver must then accept all data bits following the SFD and output them as 8-bit bytes until the incoming data returns to the "idle" state. Note that the SFD itself is not considered data and is not passed to the byte output.
4. The receiver design must recognize legal Manchester code bits "1" and "0" as well as the idle input. It must also recognize that an error has occurred when the input is low for both halves off the transmission bit period. When this occurs, the `error` output must be asserted high until either a new frame begins or the circuit is reset.
5. The receiver design must "re-synchronize" on each data bit by waiting for the transition in the middle of the bit.
6. The receiver design must assert the `cardet` output high as soon as the first transition is detected in the preamble/SFD. This output should then remain high until the input returns to the idle state.
7. The receiver design must be parameterized so that can easily be configured to operate at different bit rates (10Kbits per second default) given a 50Mhz clock input from the Spartan 3 Starter Kit board. Instructions for configuring your circuit at different bit rates must be included in header comment of your top-level Verilog file.
8. Your design must not infringe on any active patent which has been brought to your attention (specifically, the Nortel Manchester Decoder patent distributed in class).
9. You must verify your design with one or more self-checking testbenches that simulate (at minimum) the following test cases:
    a. A short data frame consisting of an 8-bit preamble/SFD followed by single 8-bit value.
    b. A longer data frame consisting of an 8-bit preamble/SFD followed by longer stream of bytes.
    c. A "spurious start" – i.e., a brief transition of `rxd` from high to low followed by an immediate return to high. Note that a spurious start should not be considered an error in the sense of Requirement 4.
    d. A deliberately erroneous input in which `rxd` remains low throughout an entire bit period as part of a larger input frame.
    e. Manchester Receiver interfaced to the Manchester Transmitter circuit designed in Lab 6.

10. Your Manchester Receiver design must be tested in hardware using the configuration shown in Figure 4. Specifically, you will connect your Manchester Transmitter design from Lab 6 with the "mxtest" module to generate short frames of data that begin with an SFD. The output of this circuit is connected to the `rxd` input of the Manchester Receiver. Parallel output of the Manchester Receiver is connected to either the 7-segment display or the 8 LEDs on the S3 board. In addition, this output connects to a FIFO circuit which buffers incoming data and sends it to a PC running HyperTerminal for viewing.
11. Your Manchester Receiver should be tested communicating with a Manchester Transmitter design created by another lab group to make sure that your design is interoperable.
12. All Verilog code must follow the Coding Guidelines discussed in class.

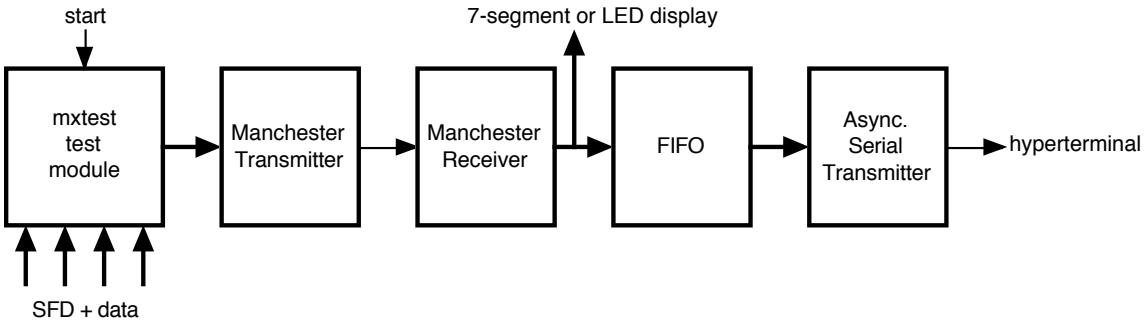

Figure 4 – Hardware Test Configuration

Deliverables

For your lab report hand in the following:

1. Demonstration to Lab Instructor of your successfully operating Manchester Receiver design as it receives characters from a Manchester Transmitter circuit.
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 and state diagrams of any FSMs in your design.
3. Verilog listings of all files, including transmitter design and testbench.
4. Timing diagram printouts showing correct simulation of your receiver design, including all required test cases.
5. Documentation of hardware testing, including the names of the group you tested your receiver with.
6. 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.
7. A short technical memorandum which describes (a) what was done, (b) what you learned, and (c) what difficulties you encountered.