CASE WESTERN RESERVE UNIVERSITY
Case School of Engineering
Department of Electrical, Computer and Systems Engineering
ECSE 281. Logic Design and Computer Organization (4)

Assignment #9 **Due: March 26, 2022**

PLEASE SUBMIT YOUR WORK AS A SINGLE pdf DOCUMENT ON CANVAS.

Laboratory (20 pts)

Decoders and Encoders

This week, we will be writing Verilog functions that implement a basic decoder and encoder using behavioral code. In particular, we will implement the 74x138 3-to-8 decoder and the 74x148 8-input priority encoder as described in the textbook. Both of these devices can be readily implemented using case statements.

Step 1: Write the 3-to-8 Decoder Function

Create a new Modelsim project and call it abc123_eecs281_lab6 where abc123 is your Case ID. In it, create a new SystemVerilog file and name it “lab6.sv”. All of the code from this lab assignment will be going into this file. Inside the lab6.sv file, create a module called “lab6” which will have two functions (one for the decoder and one for the encoder) as well as the testbench code. The module does not need any ports or parameters.

In this step, we will implement the decoder function. Your function should return an 8-bit logic vector which holds the output of the decoder. Name your function “decoder3to8”. As inputs, your function should take a 1-bit logic vector for the enable signal (“en_b”) and a 3-bit logic vector for the inputs (“in”) to the decoder. Unlike the official specifications for the 74x138, our decoder will just have a single active-low enable instead of the G1, G2A_L and G2B_L signals. Other than that your decoder function should exactly implement the truth table for the 74x138. You should implement the behavior of the function using a case (or casex) statement. Remember that the outputs of the 74x138 are active low!

Step 2: Write the 8-Input Priority Encoder Function

In this step we will implement the encoder function which will mimic the behavior of the 74x148. Your function should return a 3-bit logic vector holding the active-low encoded output. We will not implement the group select (GS_L) or enable output (EO_L) outputs of the 74x148. Name your function “encoder8to3” and have it take a 1-bit logic vector for the enable signal (“en_b”) and an 8-bit logic vector for the inputs (“in_b”) to the encoder. The body of the function should implement the truth table. As with the decoder, you should implement the behavior of the function using a case (or casex) statement.

Step 3: Test the Functions

To test our functions we will generate a 3-bit vector that counts up by one every cycle which we will use as the inputs to the decoder. The decoder output will then be fed into the inputs of the encoder to produce an inverted version of the original 3-bit test vector. For every input vector you should toggle the enable signal to both the encoder and decoder function which can be easily done by using the clock waveform used to determine when to increment the input vector as the enable signal as well.

Step 4: Deliverables

To turn in your lab code, submit all your code as part of your homework pdf file. You can copy and paste the code into a word processor and then add it to your homework pdf. You also need to submit the waveform. You can take a screenshot and add it to your homework file.