

EER-018 Introduction to Digital Computers

LABORATORY 3

Decoders and Multiplexors

Objective

In this lab you will gain some experience designing decoder and multiplexor circuits and using a multiplexor circuit to implement a function.

Equipment

- Laboratory PC
- Altera Max+Plus II Version 9.0
- 7408 and 7404 TTL parts

Prelab

1. Find expressions for the following functions, F3, F2, F1, F0 and draw a logic schematic that implements this decoder truth table.

A	B	F3	F2	F1	F0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Using the 7408 and 7404 chips, draw a pin diagram that implements F3, F2, F1, F0. You will need to refer to the TTL part handout that was distributed in class.

2. Using only 2 to 1 multiplexors, design an 8 to 1 multiplexor with select inputs S2, S1, S0 and inputs I0, I1, I2, I3, I4, I5, I6, I7. [Hint: The S2, S1, S0 inputs will be connected to the S input of the 2 to 1 multiplexors.] The output is F. Draw a diagram illustrating how the 2 to 1 multiplexors are connected, and give the function table for the resulting 8 to 1 multiplexor. Be sure to label all inputs and outputs.

Procedure

Build and Test Decoder

1. Using your results from part 1. of prelab, construct the circuit for the 2 to 4 decoder. Use toggle switches for the A and B inputs. Use LEDs to indicate the F3, F2, F1 and F0 outputs.
2. Demonstrate to your lab instructor that your circuit implements the truth table in the prelab.

Enter and Simulate Multiplexor Circuit





Construct the Circuit

1. For the multiplexor circuit from part 2 of the prelab, we will be using the 21mux component in the "mf" library. To understand the detailed operation of the 21mux, double-click on the device to see the internal construction. Analyze the 21mux circuit and develop a function table for it. Fill in the function table below:

S	Y
0	
1	

2. Using the steps outlined in [Laboratory 2](#), enter the design from part 2 of the prelab using the 21mux component. Add the components "input" and "output", from the "prim" library, to your circuit. You will need 11 inputs and 1 output components.
3. Label each input and output by double-clicking on the default " pin_name" label that appears on the component. Don't forget to save your design early and often.

Test the Circuit

1. Define the circuit to be your current project. Choose "File - Project - Set Project to Current File" .
2. Compile the circuit and make any corrections needed.
3. Open the waveform editor by choosing "MAX+plus II - Waveform Editor" and choose "Node - Enter Nodes from SNF" to add the inputs and outputs from your circuit to the waveform window.
4. Re-order and group the signals so they are in the following order:
S[2..0]
I0
I1
I2
I3
I4
I5
I6
I7
5. Choose "Options - Snap to Grid" to turn off the snap to grid option.
6. Click on signal group S[2..0] and then click on the Counter tool. (under the Clock tool, looks like this: ). Enter a starting value of 0, a binary count type, and count every 50ns.
7. Set the I7, I6, I5, I4, I3, I2, I1, I0 values to 10011010 using the  and  tools. Save the waveform file.
8. Given the waveform you just produced, predict the output, f, for the first 5 values of S2,S1,S0 (0,1,2,3,4). Review your prediction with the lab instructor before proceeding to simulation.
9. Simulate the circuit (click on the simulator button ). Record the results.

Implement Function with 8 to 1 Multiplexor

Next we will use the 8 to 1 multiplexor previously designed to implement a prime number detector. Let the inputs of the circuit be N3, N2, N1, N0. The output, P, will be true when the binary number that is input is a prime number. Recall the definition of a prime number:

"any integer other than 0 or ± 1 that is not divisible without remainder by any other integers except ± 1 and \pm the integer itself".

N3	N2	N1	N0	P
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Implement this function with the multiplexor that you have designed and additional inverters if necessary. The "gnd" and "vcc" components in the "prim" library can be used to implement constant 0 and 1 value inputs. Rename the file so that you preserve your multiplexor circuit. Remember to set the project to the new file. Verify that your results match the truth table above and demonstrate your working circuit to the lab instructor. Display your inputs in the decimal radix to make verification easier.

Report

In your discussion of your results, address the following questions:

How would the decoder design be different if active low outputs were desired? Show the logic diagram for a decoder with active low outputs.

Suppose the multiplexor inputs, I7, I6, I5, I4, I3, I2, I1, I0 were set to 11010010. What would the resulting simulation waveform be?

How would you implement the prime number detector if you only had a 4 to 1 multiplexor? Would you need additional logic? Show the implementation.