Lab 2 – Design and Implementation of Decoders and Multiplexers

Part 1: Introduction

In Lab 2, we are going to look at decoders and multiplexers. Decoders are a type of combinational circuit that converts n lines of input into 2^n lines of output. At any given time only one of the outputs can be high (1). Multiplexers are a type of combinational circuit which have 2^n number of input lines with n number of select lines to select a single output. Figure 1 showcases a 2x4 decoder and Figure 2 showcases a 4x1 multiplexer.

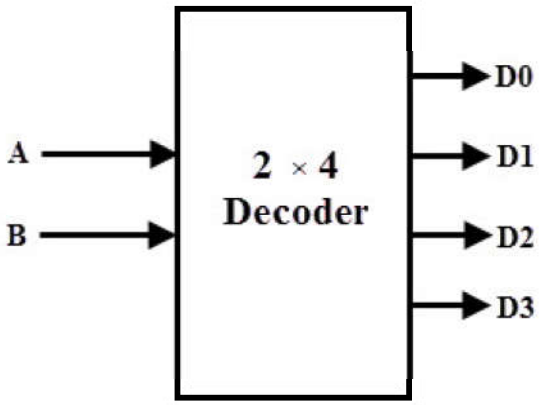

Figure 1: 2x4 decoder

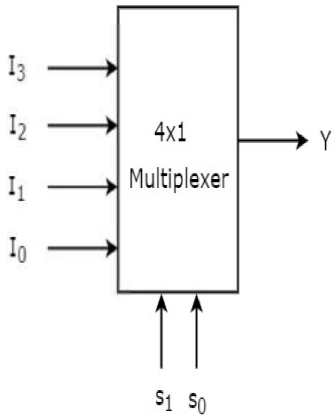

Figure 2: 4x1 multiplexer

Part 2: Design and Implementation of 2 to 4 Decoder using Logisim

For the first task, we are going to implement a 2x4 decoder in Logisim and test its functionality. Start a new project and save it as Lab 2. Before starting, ensure you are on selection mode ( ).

Step 1: In the object browser, expand "Plexers". Click on "Decoder" and place it into the workspace.

Step 2: Before working with the decoder, we are going to modify some attributes in the property window. First, click on the decoder you have created.

Step 3: In the property window, Change the "Select Bits" parameter to have the value '2'.

Step 4: Still in the property window, Change "Include Enable?" parameter to "No".

Step 5: Add an input pin ( ) to the workspace a few spaces under the decoder.

Step 6: Click on the input pin to access its property window. Change the “Data Bits” parameter to “2”.

Step 7: Add four output pins ( ) to the workspace towards the right side of the decoder.

Step 8: Connect the input pin to the select node which is on the bottom side of the decoder.

Step 9: Connect each of the output nodes of the decoder to an output pin in order of top to bottom.

If you have followed all the steps correctly, you should have something like the design shown in Figure 3.

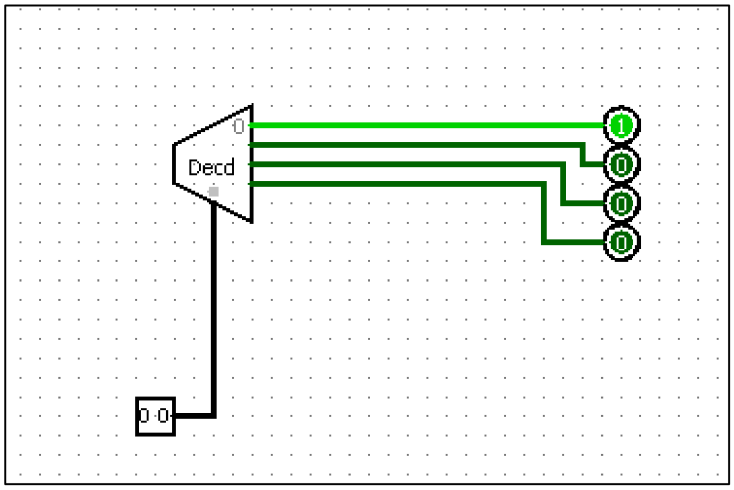

Figure 3: 2x4 decoder design using Logisim

Now, you should test the design to ensure it is working as expected. Switch to the probe using the probe button ( ) and change the inputs to test all the input combinations. Fill out your observations in the table below.

Table 1: 2x4 decoder design results

Select Input		Output			
S0	S1	D0	D1	D2	D3
0	0				
0	1				
1	0				
1	1				

Next, let us implement a 3x8 decoder. Before starting, ensure you are on selection mode ( ).

Step 1: In the object browser, expand “Plexers”. Click on “Decoder” and place it into the workspace.

Step 2: Before working with the decoder, we are going to modify some attributes in the property window. First, click on the decoder you have created.

Step 3: In the property window, Change the “Select Bits” parameter to have the value ‘3’.

Step 4: Still in the property window, Change “Include Enable?” parameter to “No”.

Step 5: Add an input pin ( ) to the workspace a few spaces under the decoder.

Step 6: Click on the input pin to access its property window. Change the “Data Bits” parameter to “3”.

Step 7: Add eight output pins ( ) to the workspace towards the right side of the decoder.

Step 8: Connect the input pin to the select node which is on the bottom side of the decoder.

Step 9: Connect each of the output nodes of the decoder to an output pin in order of top to bottom.

If you have followed all the steps correctly, you should have something like the design shown in Figure 4.

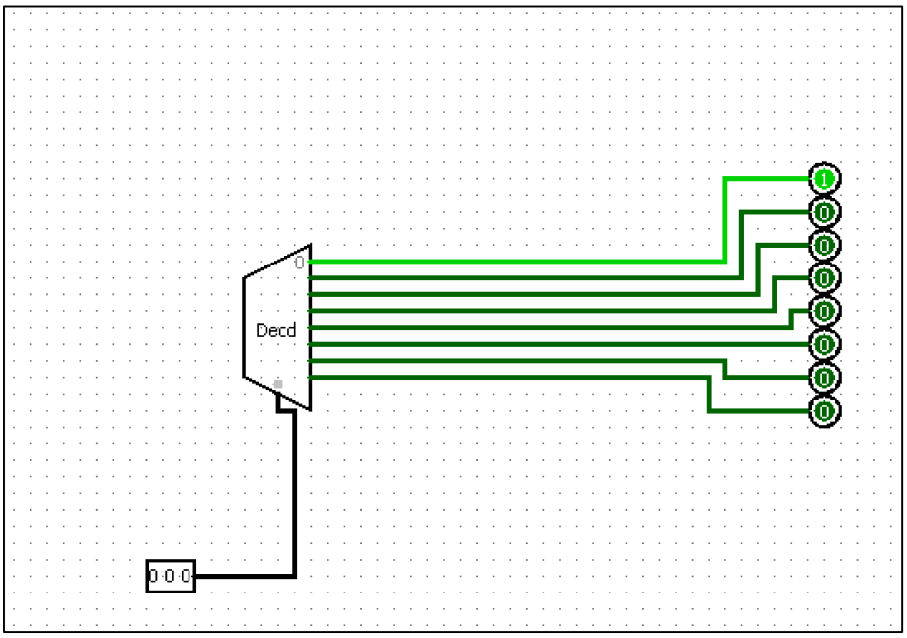

Figure 4: 3x8 decoder design in Logisim

Test the design to ensure it is working as expected. Switch to the probe using the probe button ( ) and change the inputs to test all the input combinations. Fill out your observations in the table below.

Table 2: 3x8 decoder design results

Select Input			Output							
S0	S1	S2	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

Now, let us do a simple exercise for combinational logic implementation using decoders. Any Boolean function can be expressed as a sum of min terms, a decoder along with external OR gates that form their logical sums, can be used to form a circuit of any Boolean function. Implement the following functions using only one decoder and any number of external OR gates. (Label each output pin to show which function it is representing)

Note: The input pins in the order of MSB to LSB can be labeled by ABC. For example, A binary number 101 is A = 1, B = 0, C = 1.

Implement the following Boolean functions using a single 3 to 8 decoder with active high outputs and OR gates.

$$F_1(A,B,C) = \sum(0,3,6)$$

$$F_2(A,B,C) = \prod(1,2,5,7)$$

$$F_3(A,B,C) = A\bar{B} + \bar{A}C + \overline{(BC)}$$

Screenshot of Boolean functions implementation using decoders:

Part 3: Design and Implementation of Multiplexers using Logisim

In this second part of the lab, we are going to look at how to implement multiplexers in Logisim. Start a new project and save it as Lab 2a. First, we are going to implement a 2-bit 4x1 multiplexer. Before starting, ensure you are on selection mode ( ).

Step 1: In the object browser, expand “Plexers”. Click on “Multiplexer” and place it into the workspace.

Step 2: Before working with the multiplexer, we are going to modify some attributes in the property window. First, click on the multiplexer you have created.

Step 3: In the property window, change the “Select Bits” parameter to have the value ‘2’.

Step 4: Still in the property window, change the “Include Enable?” parameter to “No”.

Step 5: Still in the property window, change the “Data Bits” parameter to have the value ‘2’.

Step 6: Add five input pins ( ) to the workspace. One a few spaces under the multiplexer and four to the left side of the multiplexer.

Step 7: Click on an input pin to access its property window. Change the “Data Bits” parameter to “2”. Repeat for all the input pins.

Step 8: Add an output pin ( ) to the workspace towards the right side of the multiplexer.

Step 9: Click on the output pin to access its property window. Change the “Data Bits” parameter to “2”.

Step 10: Connect the input pin on the bottom side of the multiplexer to the select node which is also the bottom side of the multiplexer.

Step 11: Connect the input pins on the left side of the multiplexer to the input nodes on the multiplexer starting from top to bottom, respectively.

Step 12: Connect the output pin to the output node on the right side of the multiplexer.

If you have followed all the steps correctly, you should have something like the design shown in Figure 5.

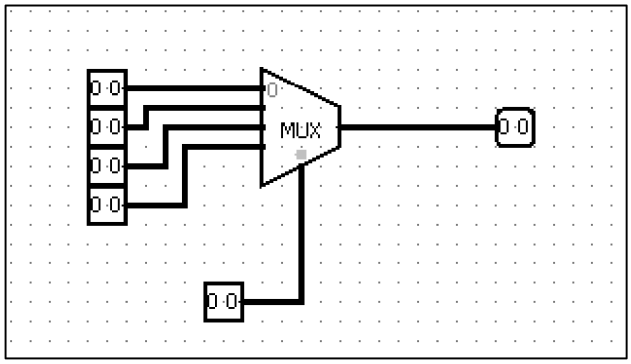

Figure 5: 2-bit 4x1 multiplexer design in Logisim

Now, you should test the design to ensure it is working as expected. Switch to the probe using the probe button ( ) and change the data inputs of the multiplexer you designed. From top to bottom, the inputs are to be labelled ‘i0’, ‘i1’, ‘i2’, ‘i3’. Once you have done this, assign a random 2-bit value to the inputs in the table below. Test all the combinations of the select pin and record the observations of the output you have obtained in the table below.

	i0	i1	i2	i3
2-bit Input Value				

Table 3: 2-bit 4x1 multiplexer design results

Select Input		Output F	
S0	S1		
0	0		
0	1		
1	0		
1	1		

Now, let us look at designing a simple combinational logic circuit using multiplexers and logic gates. For this design, we have two inputs A and B. Let us take the output to be Y.

Implement the following functions using a 1-bit 4x1 multiplexer and any number of logic gates.

1. Y = A AND B when Select = 00
2. Y = A OR B when Select = 01
3. Y = A XOR B when Select = 10
4. Y = /A XNOR B when Select = 11

Screenshot of Implementation of Logical functions using 4 to 1 Multiplexer:

Implement the following Boolean function using a 4 to 1 multiplexer and 8 to 1 multiplexer.

$$F_1(A, B, C) = \sum(0,3,6)$$

Screenshot of Boolean function implementation using 4 to 1 Multiplexer:

Screenshot of Boolean function implementation using 8 to 1 Multiplexer:

Part 4: Conclusion

In this lab, you have learnt how to implement decoders and multiplexers using Logisim. You have also learnt how to combine logic gates with decoders and multiplexers to design combinational circuits.