CS 211, SUMMER 2017

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

Objective:

This introductory lab allows you to apply everything you learned about Quartus II in the tutorials. You will build a circuit, verify correctness using waveform simulation, and load your circuit onto the FPGA board for testing. You will also be introduced to Quartus Library of Parameterized Modules to speed up your designs.

Review all 4 parts of the previous:
Part 1: you will create a simple digital circuit from AND, OR and NOT gates.
Part 2: you will build a 2-to-1 multiplexer using gates.
Part 3: you will build the same multiplexer using the Library of Parameterized Modules (LPM) available within Quartus to create circuits using the MegaWizard Plug-In Manager.
Part 4: Lab report – You may need to use this format as a general guide to complete subsequent lab reports.

Part I. Guided Design

- ## PART A: REVIEW 2:1 MUX

Exercise 1, Simple 2:1 Multiplexer

Figure 1. Shows a digital circuit that implements a 2-to-1 *multiplexer* with a select input ***s*.**

If ***s* =0** the multiplexer's output ***m*** is equal to the input ***x*,** and if ***s* = 1** the output is equal to ***y*.** Figure 4. gives the truth table for this multiplexer, and *Figure 5.* shows its circuit symbol.

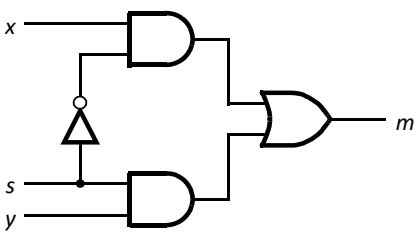


Figure 1. Digital circuit of 2:1 Multiplexer.

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

s	*m*
0	*x*
1	*y*

Figure 2. Truth Table of 2:1 MUX.

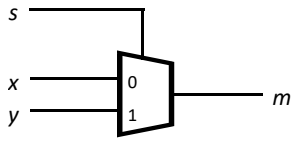


Figure 3. 2:1 Multiplexer Symbol.

What to do you have to do:

- Write a Boolean function for 2:1 multiplexer.
- Create a QUARTUS project for this circuit
- Simulate and verify the correctness of this digital circuit
- Create input and output waveform for 2:1 multiplexer. (Use the same steps as in Exercise 1). Follow all the steps from the tutorial and take a screen shot of the block diagram circuit and the vector waveform output file.
- Program the FPGA board
- Verify correctness of your design on the FPGA board

CS 211, SUMMER 2017

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

PART B

Design 4: 1 MUX Multiplexer

This a New lab based on your previous knowledge

4:1 MUX has 4 input wires and 1 output wire. Each wire carries 1 bit.

Two selector lines connect one out of 4 input wires to the output wire.

What to do:

- Write a Boolean function for 2:1 multiplexer, shown in Figure 1, using NAND gates only
- Simulate and verify correctness of NAND based 2:1 MUX, and compare to the original design
- Write a Boolean function for 4:1 multiplexer
- Draw a block diagram of 4:1 Multiplexer using **NAND gates only**. NAND gate is an AND gate followed by a NOT (inverter gate) gate.
- Draw a TRUTH table for 4:1 Multiplexer
- Create a QUARTUS project for this 4: 1 MUX digital circuit based on NAND gates only
- Simulate and verify the correctness of this digital circuit
- Create input and output waveform for 4:1 multiplexer. (Use the same steps as in Exercise 1). Follow all the steps from the tutorial and take a screen shot of the block diagram circuit and the vector waveform output file.
- Program the FPGA board
- Verify correctness of your design on the DE board

CS 211, SUMMER 2017

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

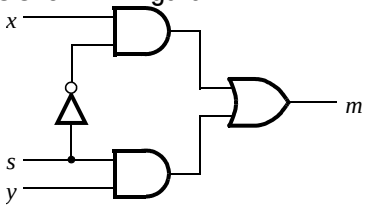
Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

Part C. DESIGN 4: 1 MUX USING *YOUR 2:1 MUX DESIGN FROM PART A AS A SYMBOL*

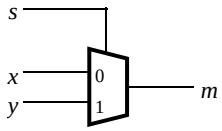
Review of 2:1 MUX from part A is shown in Figure 4.



a) Circuit

s	m
0	x
1	y

b) Truth table



c) Symbol

Figure 4. 2:1 MUX.

Figure 4 shows a sum-of-products circuit that implements a 2-to-1 multiplexer with a select input s. If s =0 the multiplexer's output m is equal to the input x, and if s = 1 the output is equal to y. Part b of the figure gives a truth table for this multiplexer, and part c shows its circuit symbol.

The 2:1 multiplexer can be described with the following sum of products (SOP) Boolean expression:

m <= (NOT (s) AND x) OR (s AND y);

- For part C, Create a QUARTUS project for 4: 1 MUX based on 2: MUX symbol.

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

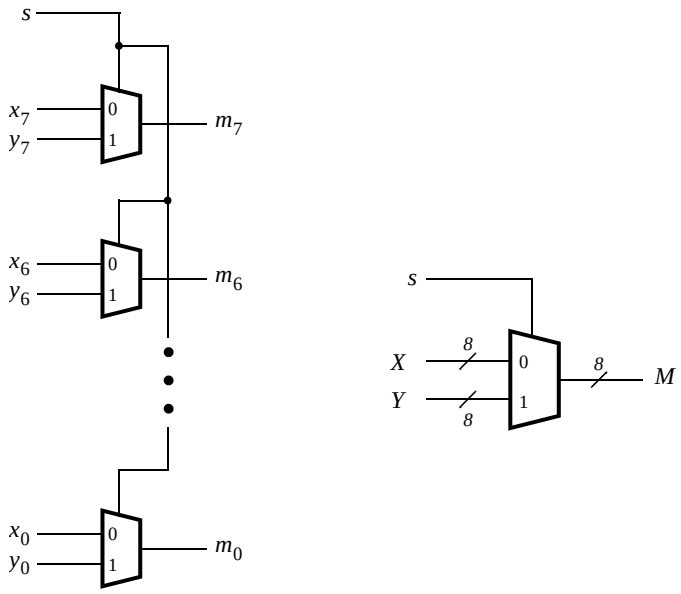
- Simulate and verify the correctness of this digital circuit
- Create input and output waveform for 4:1 multiplexer. (Use the same steps as in Exercise 1). Follow all the steps from the tutorial and take a screen shot of the block diagram circuit and the vector waveform output file.
- Program the FPGA board
- Verify correctness of your design on the FPGA board

• Part 2

- 2.1 DESIGN 2: 1 MUX where each input line is 4 BITS, output line is also 4 BITS, and there is one bit control line

This circuit has two eight-bit inputs, X and Y, and produces the eight-bit output M. If s =0 then M = X, while if s =1 then $M = Y$. We refer to this circuit as an eight-bit wide 2-to-1 multiplexer. It has the circuit symbol shown in Figure 5b, in which X, Y, and M are depicted as eight-bit wires. Perform the steps shown below.

2:1 (4 Bit)MUX



a) Circuit b) Symbol

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

Figure 5. The width of vector input **X is 4 bits.** The width of vector input **Y is 4 bits** The ouput vector M is also 4 bits A 4 bits (NOTE: you can use any number of bits in the wire, however the board we are using now has small numbe of pins).

- For this part , Create a QUARTUS project for 4 bit 2: 1 MUX based on 4 components of 1 bit 2:1 MUX (symbol).
- Simulate and verify the correctness of this digital circuit
- Create input and output waveforms for a 4 bit 2:1 multiplexer. (Use the same steps as in Exercise 1). Follow all the steps from the tutorial and take a screen shot of the block diagram circuit and the vector waveform output file.
- Program the FPGA board
- Verify correctness of your design on the FPGA board
-
- Create a new Quartus II project for your circuit.
- Include your file for the eight-bit wide 2-to-1 multiplexer in your project. Use switch SW_{17} on the DE2-series board as the s input, switches SW_{7-0} as the X input and SW_{15-8} as the Y input. Connect the *SW* switches to the red lights *LEDR* and connect the output M to the green lights $LEDG_{7-0}$.
  NOTE: Fort he board you are using please use 4 switches for each input.
- Include in your project the required pin assignments for the FPGA board you are using. As discussed in Part I, theseassignments ensure that the input ports of your design will use the pins on the FPGA board that are connected to the *SW* switches, and the output ports of your design will use the FPGA pins connected to the *LEDR* and *LEDG* lights.
- Compile and simulate the project.
- Download the compiled circuit into the FPGA chip. Test the functionality of the eight-bit wide 2-to-1 multiplexer by toggling the switches and observing the LEDs.

Part 2.2,

- ## Design 5:1 MUX

In Figure 1 we showed a 2-to-1 multiplexer that selects between the two inputs *x* and *y*. For this part E , consider a circuit in which the output *m* has to be selected from **five** (5 input lines) inputs u, v, w, x, and y.

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: *Encoders, Decoders, DeMUX*

Instructor: Professor Izidor Gertner

Due June 22, 2017

In other words it is a 5:1 MUX

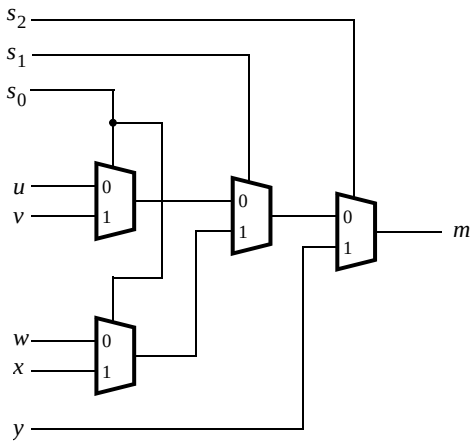
Part *a* of Figure 6 shows how we can build the required 5-to-1 multiplexer by using four 2-to-1 multiplexers as components.

The circuit uses a 3-bit select input $s_2s_1s_0$ and implements the truth table shown in Figure 6b. A circuit symbol for this multiplexer is given in part *c* of the figure.

Self-check question: Can you use 2 selector lines for 5:1 MUX?

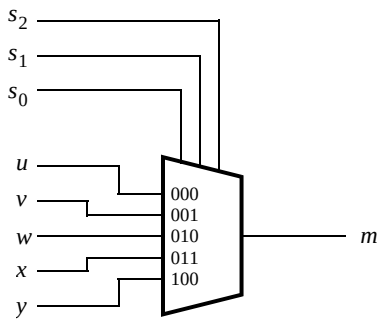
Can you use 4 selector lines for 5:1 MUX?

- What to do For part E, Create a QUARTUS project for 1 bit 5: 1 MUX based on 2:1 MUX components.
- Simulate and verify the correctness of this digital circuit, and program FPGA device on FPGA board.



a) Circuit

s_2 s_1 s_0	m
0 0 0	u
0 0 1	v
0 1 0	w
0 1 1	x
1 0 0	y
1 0 1	y
1 1 0	y
1 1 1	y



b) Truth table c) Symbol

Figure 6. A 5-to-1 multiplexer.

Part 2.3, 2-bit wide 5:1 MUX

Recall from PART D, Figure 5 that an eight-bit wide 2-to-1 multiplexer can be built by using eight instances of a 2-to-1 multiplexer components. Figure 7 applies this concept to define a two-bit wide 5-to-1 multiplexer. It contains three instances of the circuit shown in Figure 6*a*.

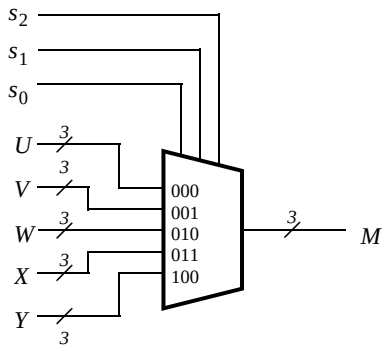


Figure 7. A three-bit wide 5-to-1 multiplexer.

(For this exersize You have to design 2 bit wide 5:1 MUX).

Perform the following steps to implement the two-bit wide 5-to-1 multiplexer.

1. Create a new Quartus II project for your circuit.
2. Create a block entity for the three-bit wide 5-to-1 multiplexer. Connect its select inputs to switches

SW_{17-15}, and use the remaining 15 switches SW_{14-0} to provide the five 3-bit inputs U to Y. Connect the *SW* switches to the red lights *LEDR* and connect the output M to the green lights $LEDG_{2-0}$.

3. Include in your project the required pin assignments for the FPGA- board you are using. Compile the project.
4. Download the compiled circuit into the FPGA chip. Test the functionality of the three-bit wide 5-to-1 multiplexer by toggling the switches and observing the LEDs. Ensure that each of the inputs U to Y can be properly selected as the output M.

PART 3:

2:1 MUX-
2-to-1 Multiplexer using
LPM (Library of Parametrized Modules)

When you design your circuits, you will realize that there are certain circuit blocks that you need to use over and over, such as adders, subtractors, multipliers, decoders, shifters, and counters. To save you time, these blocks are provided by Altera in the form of library modules that can be inserted into your Verilog designs. In Quartus, this library of modules is referred to as LPM (Library of parameterized modules).

For this part, we will implement the same 2-to1 multiplexer you designed in part2; however, this time we will implement it from the MegaWizard Plug-In available in Quartus.

Note: The menu ítems shown here may be slightly different in the Quarstus version you are currently using.

CS 211, SUMMER 2017

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017

YOUR TASK

1. Create a new project. Call it **Your name LPM_MUX**.

2. Go to **Tools > MegaWizard Plug-In Manager**

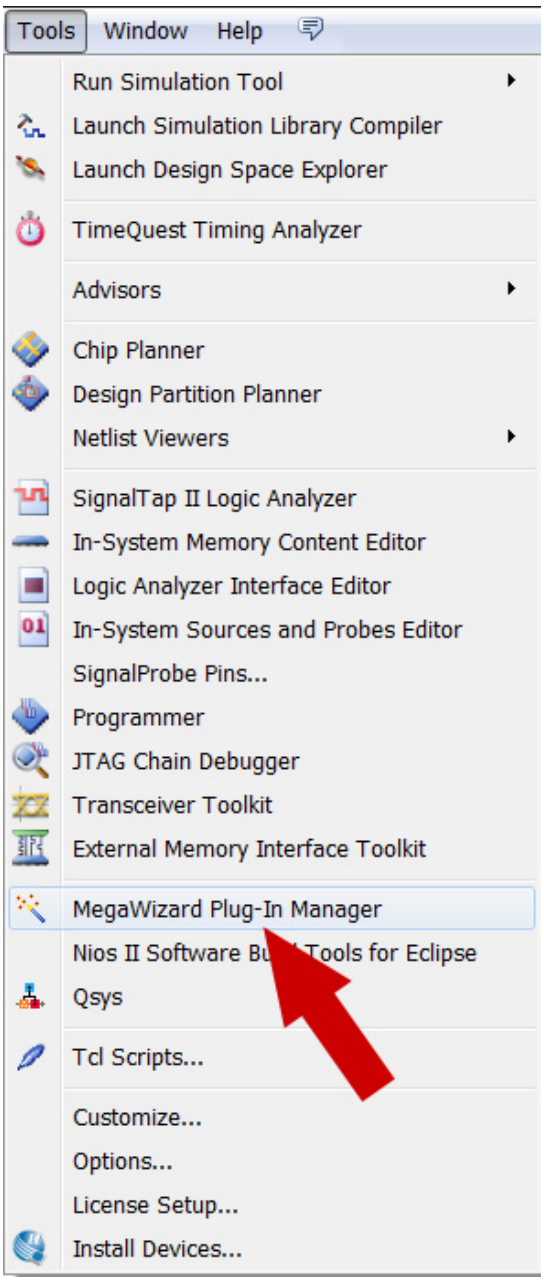
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017



3. The following screen shows up. Click **Next**

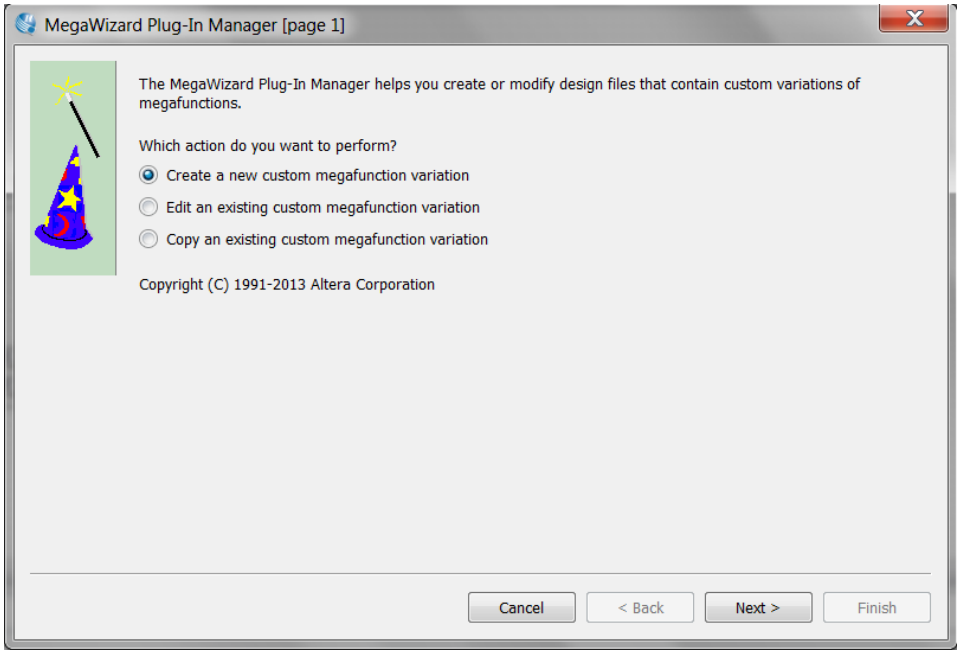
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017



4. In the screen that appears, browse for LPM_MUX under the Gates folder. Alternatively, you can also type “mux” in the search box on the left, as shown in the figure below. Now click to select LPM_MUX. Then on the right give a name to the output file, you might want to call it muxLPM or something similar. Click **Next**

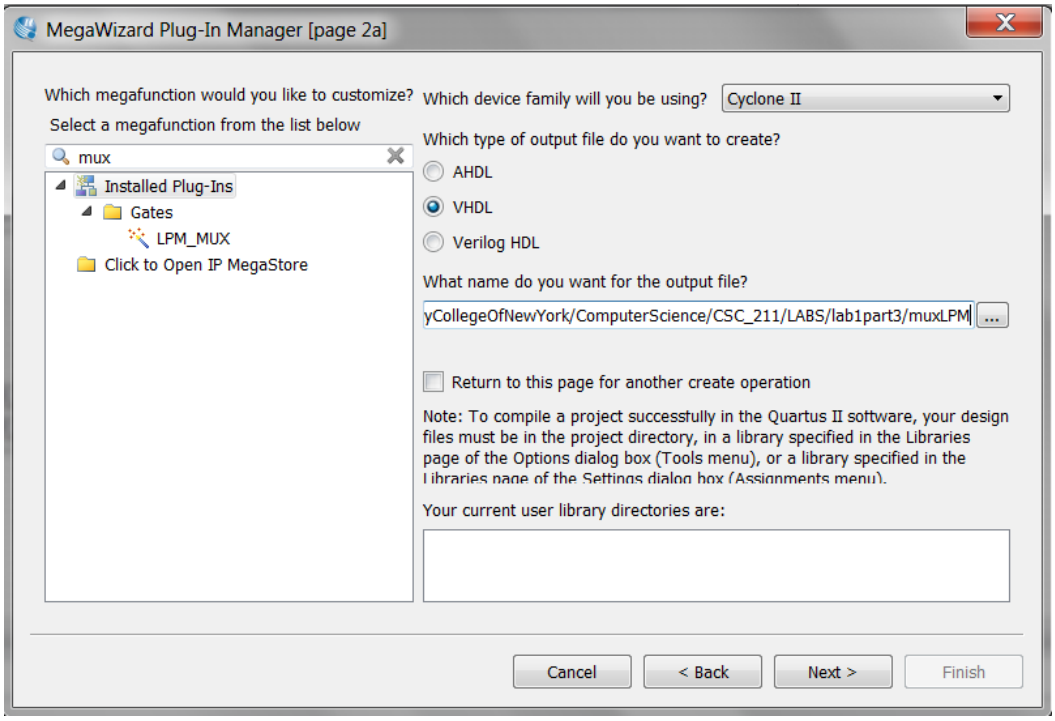
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: ***Professor Izidor Gertner***

Due June 22, 2017



5. In the next screen make sure that you specify 2 inputs and 1 output bit as shown in the figure below.
   Note that the symbol for the resulting LPM is shown in the top left corner and any changes you make through the wizard will be reflected in the top left corner image.
   Click **Next** twice.

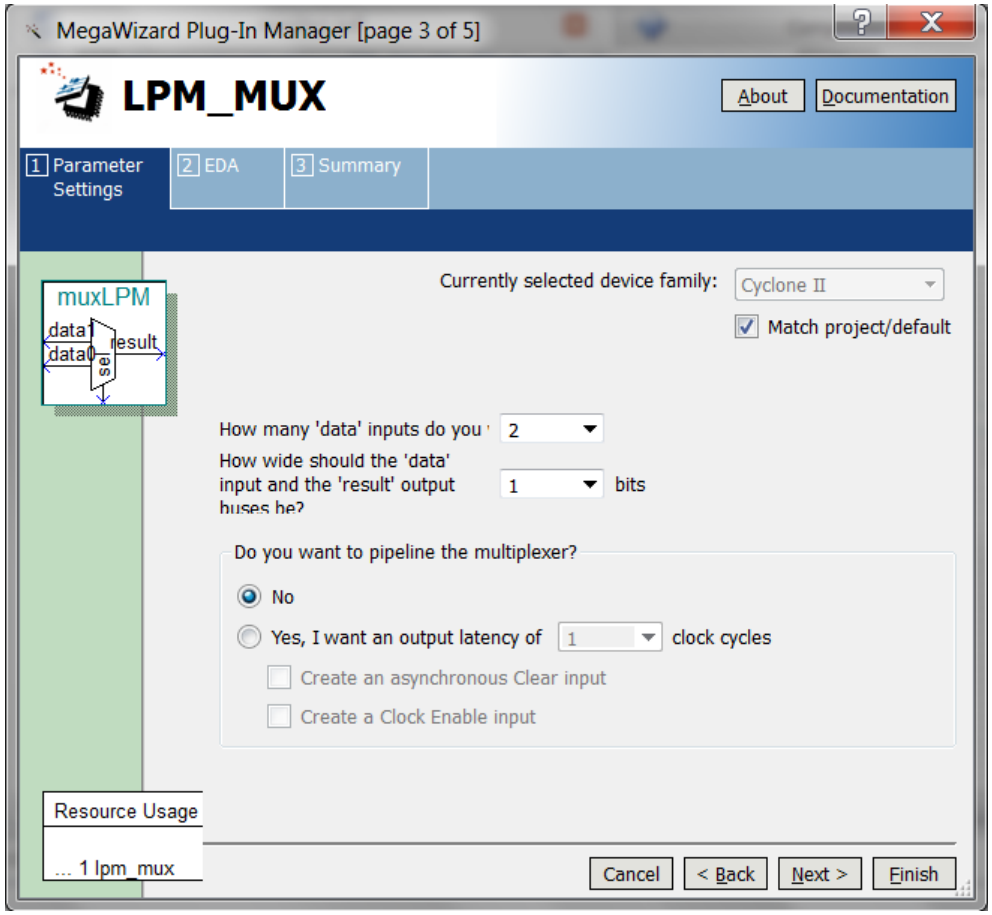
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: ***Professor Izidor Gertner***

Due June 22, 2017



6. This is the final screen, you should simply click **Finish**

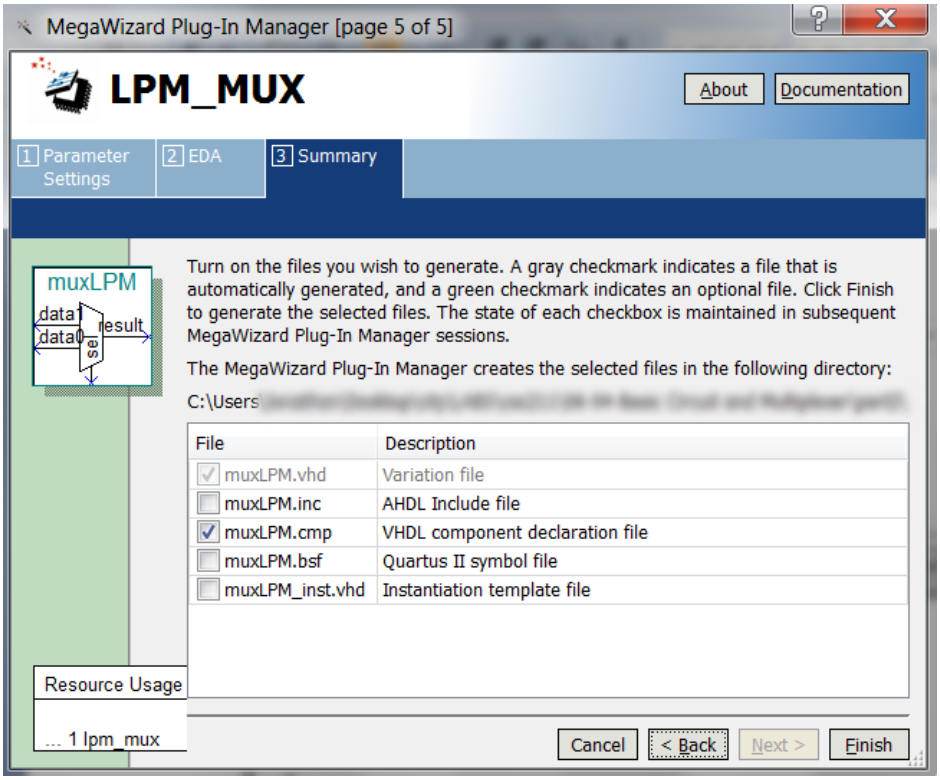
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

Due June 22, 2017



7. If you are asked whether you want to add the circuit you just created to the project, click **Yes**.

8. Quartus automatically generated a VHDL file, you can verify in the navigation panel as shown below:

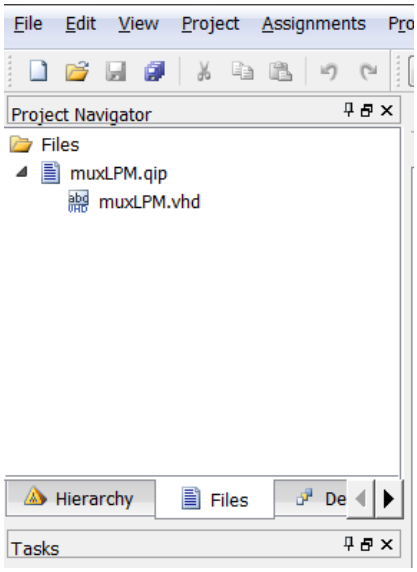
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

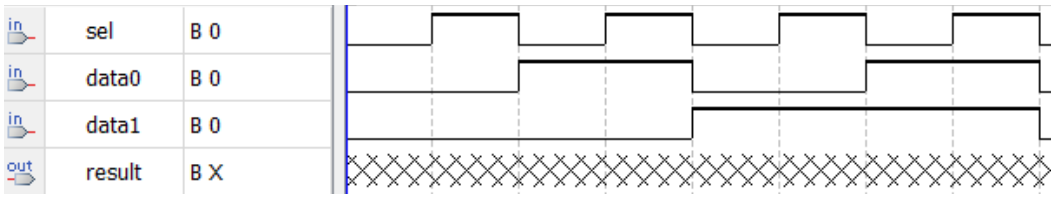
Instructor: Professor Izidor Gertner

Due June 22, 2017



9. You don't need to make any changes to this VHDL file. Simply right click the vhd file and set it as your top level entity, then compile.

10. Create a waveform simulation file. Note that Quartus will give some default names to your inputs and outputs, in this case *sel*, *data0*, *data1* for inputs, and *result* for output.
Your setup before simulation may look something like this:



11. Run your waveform simulation, you should obtain the same output behavior described in part 2 of this lab.

12. Pin Assignment:
[Click here for pin assignment manual]

Create a new text file to add your pin assignments:
data0: Assign to SW[0]
data1: Assign to SW[1]

sel: Assign to SW[2]
result: Assign to LEDR[0]

13. You should be now ready to load and test this circuit onto FPGA board. The behavior should be the same as the multiplexer designed manually in part 2.

PART : Lab Report and DEMO video.

YOUR TASK

After finishing the above steps, write a report following the format below. Be sure to answer all the questions in the report. You should use Microsoft Word to write the report; **no handwritten reports will be accepted**.

Objective:

- What is the goal of this lab?

Functionality and Specifications:

- What are the inputs and outputs for each circuit and what are they assigned to on the DE2 board?
- Include a screenshot of the circuit diagrams you designed (in this case from part 1 and part 2).

Simulation:

- Include a screenshot of the vector waveform from every part of the lab after producing the output.
- Explain the functionality/behavior of the circuits from each part of the lab, referring to the waveforms obtained.

Conclusion:

- What did you learn from this lab?

Appendix:

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

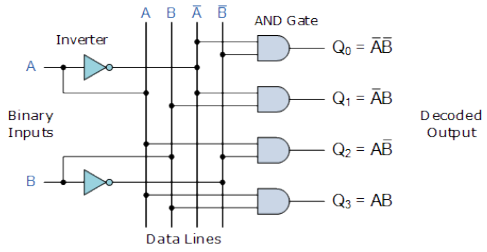
Instructor: Professor Izidor Gertner

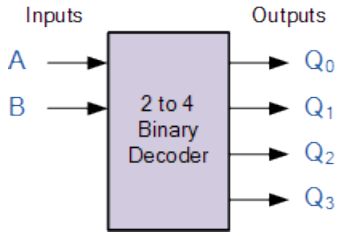
Due June 22, 2017

- Include the text of the pin assignments. Also, whenever you write your own VHDL code in future labs, you should also include your code here.

Part II . Independent Design

II.1 Design 2:4 Decoder, Your goal is to design





Truth Table

A	B	Q_0	Q_1	Q_2	Q_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Simulate, verify correctness and Program FPGA

You can use any reference on the WWW.

Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: ***Professor Izidor Gertner***

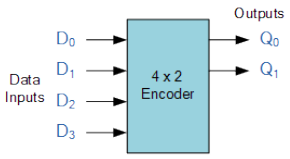
Due June 22, 2017

II.2 Design 3:8 Decoder, Your goal is to Extend your design in II.1 to create a 3: 8 decoder.

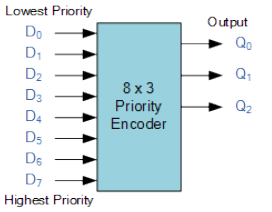
Simulate, verify correctness and Program FPGA

You can use any reference on the WWW.

III.3 Design Octal to Binary Encoder (8:3 Encoder), Your goal is to Simulate, verify correctness and Program FPGA



Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x



Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	x	0	0	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	1	x	x	x	x	1	0	0
0	0	1	x	x	x	x	x	1	0	1
0	1	x	x	x	x	x	x	1	1	0
1	x	x	x	x	x	x	x	1	1	1

X = dont care

Please write all three Boolean functions and simplify to get the following circuit.. You can use any reference on the WWW.

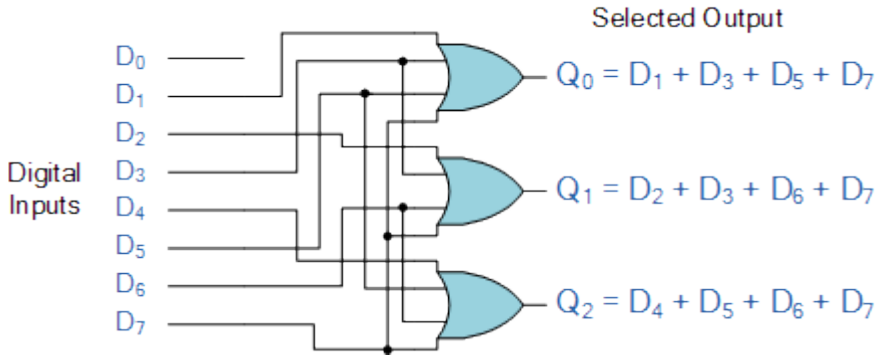
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

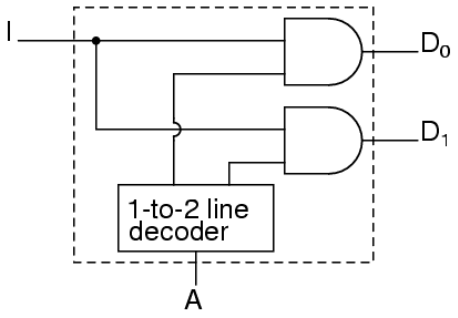
Due June 22, 2017



III.3 Design Demultiplexer- DMUX , Your goal is to Simulate, verify correctness and Program FPGA

A demultiplexer is a circuit that has one input and more than one output.

It is used when a circuit wishes to send a signal to one of many devices.



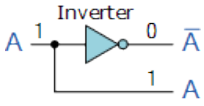
Laboratory Exercise 3 : Advanced Digital Circuits

Using LPM modules, MUXes

Independent Design Part: ***Encoders, Decoders, DeMUX***

Instructor: Professor Izidor Gertner

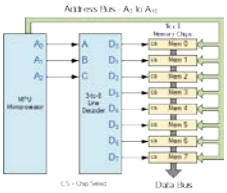
Due June 22, 2017



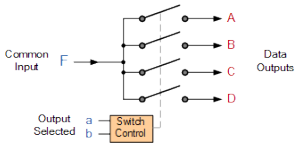
An inverter (*NOT-gate*) can be called as a 1-to-2 binary decoder

This description sounds similar to the description given for a decoder, but

a decoder is used to select among many devices



while a **demultiplexer is used to send** a signal to one out of many devices.



b	a	Data output Select
0	0	A
0	1	B
1	0	C
1	1	D