You as the student throughout the last few weeks have learned many logics and designs that will come in handy in the near future. Believe it or not you have started on the road to designing a Single Cycle CPU. With that said, you have coded in VHDL the following designs: **AND, COUNTER, OR, XOR, MUX, DECODE, ADD and DFF** – for the moment keep these entities in mind while I show you how you can use a library in which it allows you to select any of the above entities and customize them for future designs. The library that we will be using is located in Quartus II; it is referred to as a Library of Parameterized Modules (LPM) – For more information go to [http://www.altera.com/products/software/products/legacy/maxplus2/sfw-lpm.html](http://www.altera.com/products/software/products/legacy/maxplus2/sfw-lpm.html). This will give you a brief introduction to LPM and why they were created.

After this tutorial you should be able to do the same steps for other library functions and as well as to be able to simulate them, so that you are able to understand the logic behind the LPM. As a further notice, in the future you will be using LPM for your other projects.

**Begin the tutorial by doing the following steps:**

1. The first step we need to do is open up Quartus II. In the Unix Lab you need to open it by typing `quartus` in the command prompt.

2. In the top-left hand side of the window go through the following procedure:
   a. Click on `file` and go to `New Project Wizard (Click on this)`.

3. Now, create a temporary directory in your workspace. For example, if you are working in the `My Documents` (or your home directory if you are in UNIX) directory create a folder beginning with your initials (`<initials>_tutorial`; where your initials replace the `<initials>` block). You will use this folder as a place to store all of your projects. Within this folder create another folder titled `LPM_Testing`.

4. Making sure that you are in your working directory at the moment, you should get a screen that looks like this:
5. For our example we are going to name the project *LPM_Testing* and the name of the top-level design entity *LPM_Testing*. Note: Quartus already may have done this automatically for you.

6. Click *next* on the current screen and you should get the add files window for our example we will continue because we do not have these files yet (we’re going to create them in a few moments); *in the future this will come in handy when you need to include many files to one project*. Click *next* we do not need to do anything for the moment.

7. Now you should be on the *Family & Device Settings* window. Please do the following:
   a. In the family drop-down menu select *Cyclone*.
   b. In the target device section select the *Specific device selected in ‘Available devices’ list* radio button.
c. In the filters section you want the package to be set to *Any*, the pin count to be set at 240 and you want the speed grade to be set to 8. You may want to have the Show Advanced Devices check on.

d. Finally, from the available devices select EP1C6Q240C8. The whole process should match the following.

e. Click next to proceed.

8. You should now be in the EDA Tool Settings window. You may skip this section, as we need not require specifying a tool at this time.

9. The final screen that you will come across is the Summary window. It should resemble the following screen.
10. Click Finish. We just created our project now we need to create our design.

The Design

1. In the top-left hand side of the window go through the following procedure:
   a. Click on file and go to New (Click on this). You should have a screen that looks as the following.
b. Under the Device Design Files tab highlight Block Diagram/Schematic File by clicking on it. The choose OK. You should now have a new window with grids on them. This is the area where you will place the components required to do this tutorial.

2. Let’s start by placing a 3x1 multiplexer symbol into the area. You can do this clicking the Symbol Tool icon (this is usually located to the left of the area that you place the components in) or by double-click anywhere in the workspace.
3. You should now be in the Symbol window and you’ll get the following window.
a. You want to click on the MegaWizard Plug-In Manager button.

b. You want to Create a new custom megafuction variation, click Next to continue.
1. In the MegaWizard Plug In Manager window, we will expand the *gates* folder. It will look like the following screen:

![MegaWizard Plug In Manager](image)

The *MUX* gate function is located under this folder. You will want to name the output file as *MYMUX3X1.vhd* (you can also name it something else as long as it’s not the same as the file name that is reserved for the software). *Note*: Quartus may complain if you name this file exactly as the one that is reserved for the software.

Click *Next* to continue.

2. You will then get this next window where you will edit the function with other attributes.
Note: You are creating a multiplexer with 3 data inputs and the data inputs are 4 bit wide (for now; in the future you can configure it as you wish). Click Next to continue.

3. Finally, click on Finish to complete the process and you will get the screen below.
CSc 343: LMP Tutorial Part I of II

4. Place one symbol into the area. Right-click the mouse if you need to and select cancel to stop the pasting operation from occurring twice.

5. Now, go back to the Symbol Tool icon and under the `altera/quartus50/libraries/pin` (this may be a different directory if in Unix, but the idea is the same) folder we are going to add four input (s) and one output to our design. The end result should look as followed.

6. Now, right-click the input (s) and output, and select Properties and name them as followed:

   a. The input leading to `data0x[3..0]` should be named `a[3..0]` and `data1x[3..0]` should be named `b[3..0]` and `data2x[3..0]` should be named `c[3..0]` and last the `Sel[1..0]` should be named `S[1..0]`. The output pin should be named `Out[3..0]`.

7. Now, we need to connect the pin(s) as followed using the Orthogonal Bus Tool icon.

8. We are now ready to save our design. Click file then save or the save icon to save your design as `LPM_Testing`; Quartus automatically saves this as the
CSc 343: LMP Tutorial Part I of II

same name as the folder. You should get the following screen.

9. You want to have the *Add file to current project* check on, so that you can add this design to the *current project*.

10. We finished our design now it’s time to compile our design and view waveforms.

**Compilation and Waveforms**

1. Start by clicking *Processing* and then choosing *Start Compilation*. It should look as followed.
2. You should see a status screen and a final confirmation that the project compiled successfully.

![Status Screen](image1.png)

![Confirmation Screen](image2.png)

3. Let us now generate test vectors for simulation.

   a. In the top-left hand side of the window go through the following procedure:
   b. Click on file and go to New (Click on this). You should have a screen that looks as the following.

![New Screen](image3.png)
c. Under the Other Files tab highlight Vector Waveform File by clicking on it. Then choose OK. You should now have the following window.

4. Now we need to go to the Node Finder by going to the View tab and then going to Utility Windows. The Node Finder will give us the number of input(s) and output(s). This step is required for us to do our waveform simulation. The step should look as followed.
5. The *Node Finder* window should now be opened. The first step you’ll need to do is to click on *List* (this will list all the input (s) and output (s) of a design). It should look as the following.

![Node Finder window with list of nodes](image)
6. You will need to highlight the following nodes (of the type: Input Group (s) and Output Group); \(a, b, c, \text{Out},\) and \(S\) and then drag them to the \textit{waveform} window. You should end up with the following results.

7. Now, right-click the \(a\) input from the \textit{waveform} window and do the following:
   a. Select \textit{Value} \& \textit{Arbitrary Value} and you should get the following screen.

   ![Arbitrary Value Screen]

   b. Under the \textit{Numeric or named value} tab, type \textit{1011}. For now do not worry about the other parameters (leave it as binary).

   c. Do the same procedure for the \(b\) input but type \textit{1111} as is. And as for the \(c\) input type \textit{0101}. 
8. Now, right-click the S input from the *waveform* window and do the following:
   a. Select *Value* \(\|\) *Count Value* and you should get the following screen.

   ![Count Value Window]

   b. Under the *Timing* tab, change the *End time: 2.0 us* to *1.0 us*. Do not worry about the other parameters.

9. Before we run our simulation, we need to save the waveform file by clicking the save icon (\(\square\)) and you may save it with any name that you wish (for now let's use *LPM_Testing*). Now, in order to run the simulation you need to click on *Processing* and then choose *Start Simulation*.  

-16-
10. It is critical that you understand your vector simulation it can come in handy in the toughest spots. From this simulation you can see the logic that you observed when you did the multiplexer in ModelSim. i.e., when S is 00, the first data is selected as the output (a) and when S is 01, the second data is selected as the output (b), etc... An example of the vector waveform simulation is shown below. *Note:* At first you may not have a perfect view like this so you may need to zoom out (by right-clicking the area and going to *zoom* and choosing *Zoom Out (or holding Ctrl + Shift + Space))*). Similarly, you may zoom in (by choosing *Zoom In (or holding Ctrl + Space)*).

<table>
<thead>
<tr>
<th>Name</th>
<th>Value at 5.0 ns</th>
<th>5.0 ns</th>
<th>20.0 ns</th>
<th>40.0 ns</th>
<th>60.0 ns</th>
<th>80.0 ns</th>
<th>100.0 ns</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>B 1011</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>B 1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>B 0111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Out</td>
<td>B 1011</td>
<td>1111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>B 1011</td>
<td></td>
<td>1111</td>
<td>0101</td>
<td>1011</td>
<td>1111</td>
<td>0101</td>
</tr>
</tbody>
</table>