CSCE 230L – Lab 6: Decoders and Multiplexers

February, 2014

Objectives

- Learn what a decoder does and build one with a block diagram
- Implement a multiplexer using your decoder
- Use a block diagram files in another block diagram

Useful References

Appendix A in the textbook (Sections 9 and 10)

Quartus and ModelSim help (the last page of this file)

What to Turn In

- Pre-lab questions (due at the beginning of lab; in paper)
  - All questions total to 20 points
- Files and questions from lab (submit on handin)
  - Project archive file (.qar)
  - Simulation results for each component (.bmp)
    - Decoder
    - One-bit multiplexer
    - Four-bit multiplexer

Answer all the questions in a single text file. For the project files, create a Quartus archive file (.qar). Simulation results can be a bitmap file (.bmp). Remember that the Quartus archive file does not include your simulation results. The .bmp files will be separate from the .qar file.

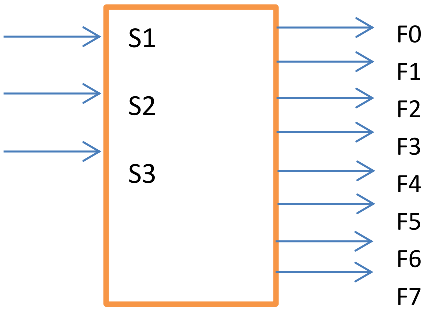
Name (1 point):

Pre-Lab (19 points)

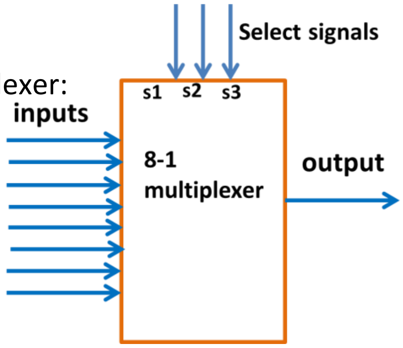
1. What's the difference between a decoder and a multiplexer? (From the aspect of input, output, and functions)

2. Using the decoder in figure A.35 on page 506 as an example, draw a gate diagram for a 3-to-8 decoder (i.e. inputs S_1, S_2, and S_3; outputs f_0, f_1, …, f_7).

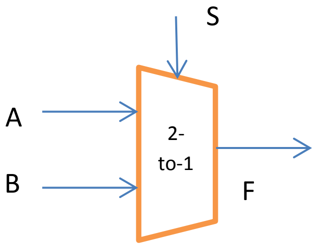
3. Using the decoder you just made above, implement an 8-to-1 multiplexer. You can use the given diagram for a decoder. S1, S2, and S3 are your select signals, and you should have 8 more input signals in addition to the select signals.
   The decoder:



the multiplexer:



4. Using 1-bit 2-to-1 multiplexers (shown below), create a 4-bit 2-to-1 multiplexer, in which the select signal has only one bit, and can handle 4-bit input. Each of the multiplexer works parallel with others. Your inputs will look like A[3..0] and to specify one bit you can use A[0], A[1], etc. You will have a multiplexer for each bit of the inputs (in this case, 4-bits so 4 multiplexers).



Lab

Part 1

Decoders have the function of taking in an n-bit input, and having up to 2^n outputs. The simplest decoder will have one of its outputs be 1 and the rest of them are 0. However, as the book shows in figure A.36, we see that decoders can have a more specific functionality, where outputs can all vary, based on the input.

For this part of the lab, create your decoder with a block diagram file. We want to make the general decoder, where one output will be 1 and the rest of the outputs will be 0. Then create a test script and run a simulation in ModelSim to ensure that it works correctly. Remember in the test file ***to test all possible values of the inputs***. To ***<u>save your test results</u>*** from ModelSim, go to File->Export->Image. Make sure you ***zoom in on your wave some***.

Part 2

It's better to make this part and also part 3 all in the same project as part 1. Now that we made a block diagram for the decoder, we want to be able to use it in another block diagram file to make the 8-to-1 multiplexer. Quartus will let us do that with a few steps.

1. With the decoder block diagram file open, go to "File"->"Create/Update"->"Create Symbol Files for Current File".
2. Now in our block diagram for the multiplexer, we can add the decoder to the diagram just like we were adding any other gate, only it will be in the Project folder instead of the Altera libraries folder.

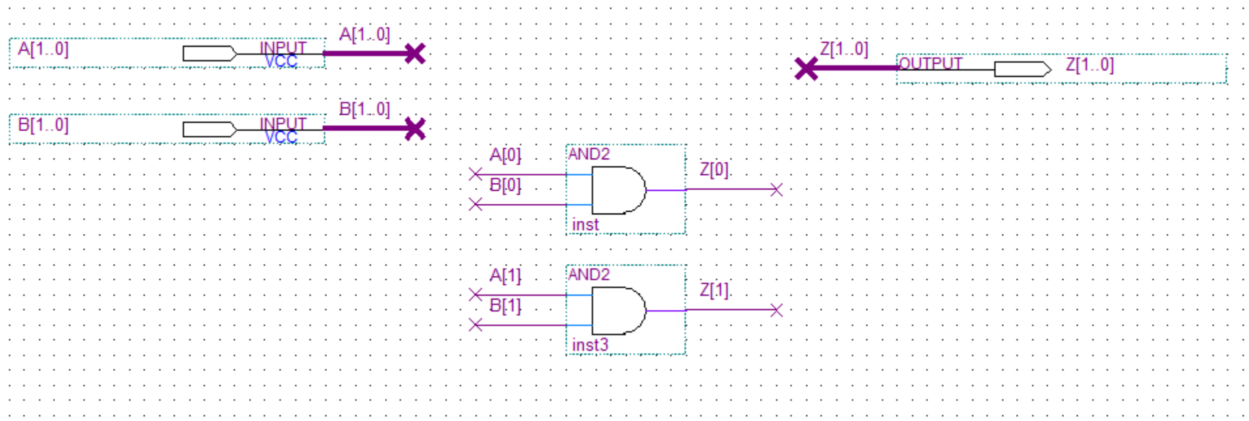
Now being able to use the decoder, go ahead and create the 8-to-1 multiplexer you made in the pre-lab. We can see that using the decoder eliminates some of the logic needed for the selection inputs. Make sure to create test script and run a simulation to test that it works correctly. In the test file, you only need to go through ***all the possible values of the selection inputs*** (S1, S2, and S3). The eight multiplexer inputs can be constant (but make them different from each other). ***<u>Save an image of your test results.</u>***

Part 3

Now we want to take out multiplexer one step further. The 8-to-1 multiplexer we just made only handles 1 bit inputs. We know that computers use variables of several bits, so we need multiplexers that can handle multiple bit inputs. Like we did with the decoder, create a symbol file from your multiplexer. Now build the 4 bit multiplexer in ***similar*** fashion to what you did in pre-lab (note: they are different from the one in pre-lab). When you finished the block diagram, create a test file and perform a simulation. ***<u>Save an image your results</u>***.

Quartus Help

There are a few nuances to have inputs and outputs be 4 bits instead of 1. When naming them, name them in the following manner: A[3..0]. Name the wire connecting to the input the same as well. To handle multiple bits, we want to use the orthogonal bus tool instead of the orthogonal node tool. The inputs to the single bit 8-to-1 multiplexers will still be one bit, so to specify one bit, we use the orthogonal node tool and name the wire something like A[0], A[1], etc. In Quartus, wires don't have to be connected as long as they have the same name. The following picture hopefully puts it all together.



ModelSim Help

How to assign value to a multiple bits? See the ModelSim help file posted in lab5.