Digital Design Lab #6: Multiplexers and Decoders

Charlie Coleman
Lab Partner: Luke Taylor
2016 October 13

Objective: The objective of this lab was to implement functions using decoders and multiplexers and to design more complex multiplexers.

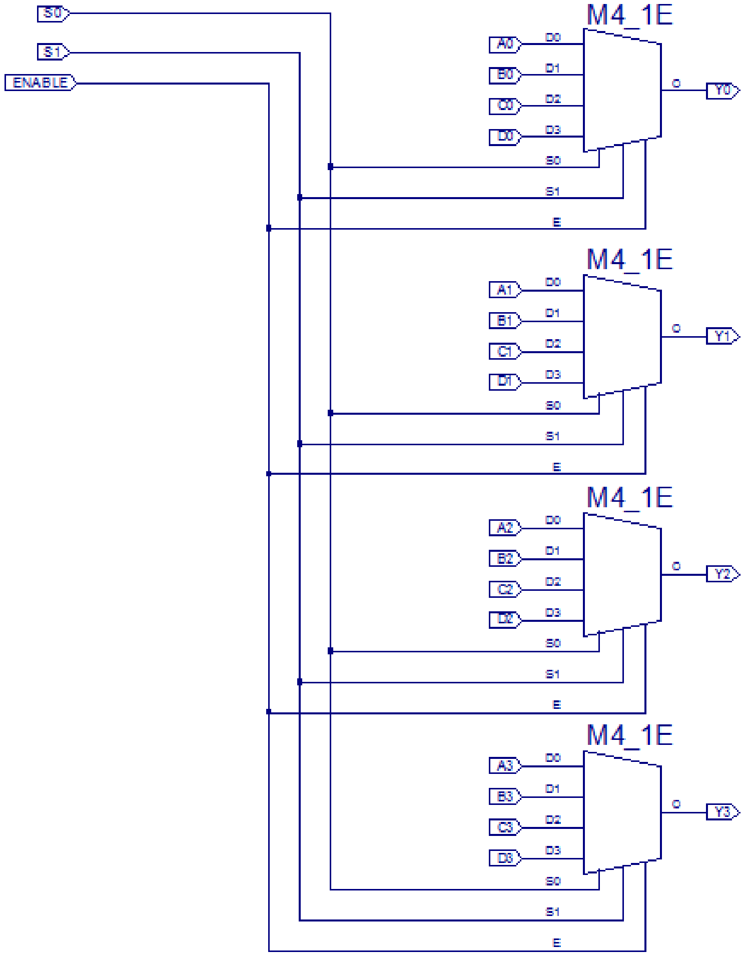
Design:

In part one of the lab, we were to design a quad 4:1 multiplexer using regular 4:1 multiplexers. The quad multiplexer takes four groups of four inputs and outputs one group of four based on the select lines.

In part two, we were to implement a function of four variables with an 8:1 multiplexer and extra logic gates and then implement the same function with a 4:1 multiplexer and extra logic gates.

In part three, we were to implement multiple functions that all had the same inputs using a 4:16 decoder and extra logic gates.

Part 1:



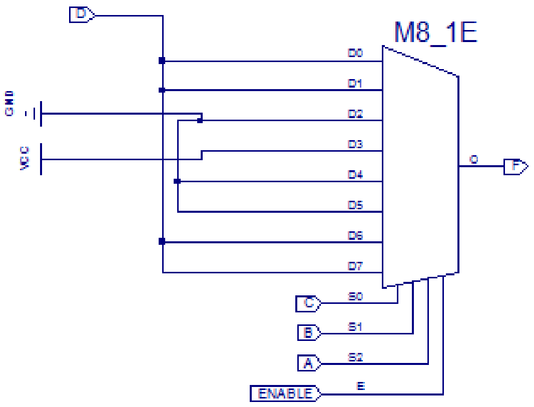
Part 2:

$F(A,B,C,D) = \sum m(1, 3, 6, 7, 13, 15)$

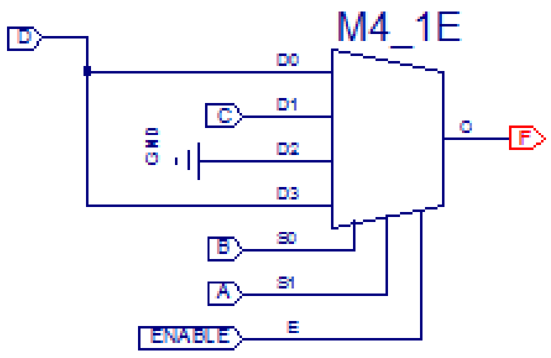
Truth Table 1:

A	B	C	D	F		
0	0	0	0	0	D	D
0	0	0	1	1		
0	0	1	0	0	D	
0	0	1	1	1		
0	1	0	0	0	0	C
0	1	0	1	0		
0	1	1	0	1	1	
0	1	1	1	1		
1	0	0	0	0	0	0
1	0	0	1	0		
1	0	1	0	0	0	
1	0	1	1	0		
1	1	0	0	0	D	D
1	1	0	1	1		
1	1	1	0	0	D	
1	1	1	1	1		

A)



B)



Part 3:

F0 (A, B, C, D) = Σm(0, 14, 15)

F1 (A, B, C, D) = Σm(1, 13, 15)

F2 (A, B, C, D) = Σm(2, 12, 15)

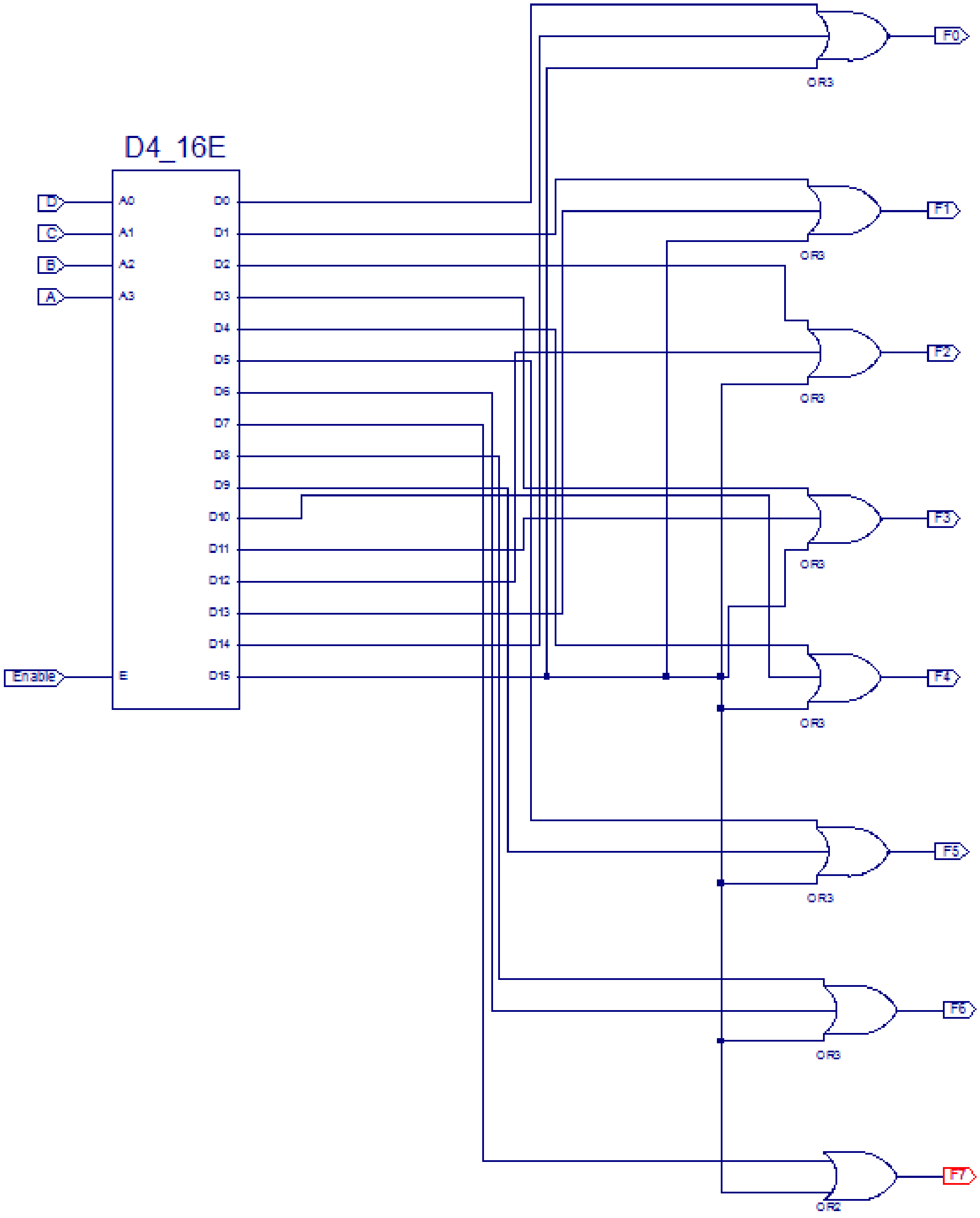
F3 (A, B, C, D) = Σm(3, 11, 15)

F4 (A, B, C, D) = Σm(4, 10, 15)

F5 (A, B, C, D) = Σm(5, 9, 15)

F6 (A, B, C, D) = Σm(6, 8, 15)

F7 (A, B, C, D) = Σm(7, 15)



Procedure:

To begin this week's lab, we designed the multiplexer circuit, 4:1 mux, 8:1 mux, and a 4:16 decoder to the specifications above. For the multiplexer circuit, we simulated the design for S = 0 with A changing from 0 to 6 to 12 and B C and D held constant. We then set S = 1 and let B vary between 5, 13, and 15 with A C and D constant. We then set S = 2 and let C vary between 4, 6, and 9 with A B and D constant. Finally we tested with S = 3 and D varying between 10, 12, and 3 with A B and C constant. We then simulated the 8:1 mux and 4:1

mux for all possible combinations using the clock function. These mux’s were then downloaded to a board and again verified. We then simulated the 4:16 decoder for all possible combinations using the clock function. We downloaded this to the board and again tested.

Data: See Xilinx simulation results

Data Analysis: Based on the simulation results for part one of the lab, the circuit performed as intended. There was no truth table, but through comparing the results to the expected behavior of a quad 4:1 multiplexer. Both circuits in part two performed as intended as their simulation matched Truth Table 1, but the 4:1 multiplexer was more efficient than the 8:1, as it had 7 input lines rather than 11 (not including those in the multiplexers). Part three also worked as intended as the simulation matched the min terms given in the design.

Conclusion: In this lab, we practiced application of multiplexers and decoders in implementing functions and creating more complicated decoders. Creating multi-output multiplexers helps when you need to select and read multiple data values at a time. Implementing functions using multiplexers allows us another way to design schematics and makes the schematic easier to read. The implementation of multiple functions using the same inputs using a decoder is helpful when many functions are defined using the same inputs, as it makes the functions easier to read, and if there are many functions it can reduce total input lines. One challenge faced in the lab was the number of wires in part three made it very difficult to see what inputs were going to what OR gate. It was also difficult trying to set multiplexer inputs to one or zero because we had not used Vcc or GND before.