CDA 3201L Thursday Section
Lab number 03 -- Combinational Logic Circuits III
06/07/14 , John Gangemi & Kyle Peck

PURPOSE AND OBJECTIVES

The three part lab focused on a specific combinational logic circuit, called a multiplexer, this electronic device is "designed to switch one of several input lines through to a single common output line by the application of a control signal" (*Basic Electronic Tutorials*).

Part A of the lab request that our group designs and implements the Boolean expression $f(x,y,z) = \Sigma(0,2,5,7)$ using a single 8-to-1 multiplexer and provide the function's truth table with all possible input combinations.

Part B of the lab request that our group designs and implements the Boolean expression $f(x1,x2,x3,x4) = \Sigma(0,1,2,3,4,9,13,14,15)$ using a single 8-to-1 multiplexer and inverters.

Part C of the lab request that our group designs and implements a logic circuit that compares two 3-bit binary numbers and generates different 2-bit outputs dependent upon the values for each binary number. The two 3-bit binary numbers are either equivalent or one is greater than/less than the other.

COMPONENTS USED

- Integrated Circuits For Demonstration
    - . 74LS04 HEX Inverter
    - . 74LS11 3-Input AND gate
    - . 74LS08 2-Input AND gate
    - . 74LS32 2-Input OR gate
    - . 74LS151 8-to-1 Selector/Multiplexer
    - . 74LS266 2-Input Exclusive-NOR gate
- Additional Integrated Circuits In Observations Section
    - . 74LS150 16-to-1 Selector/Multiplexer
    - . 74LS02 2-Input NOR gate
- 5 x 5mm Red LEDs
- 5 x 470 ohm Resistors
- 5 Volt Regulated DC Power Supply
- Assortment of 22 AWG Jumper Wires
- 2250 point Breadboard
- Logisim Program version 2.7.1

DESIGN DESCRIPTION

Part A

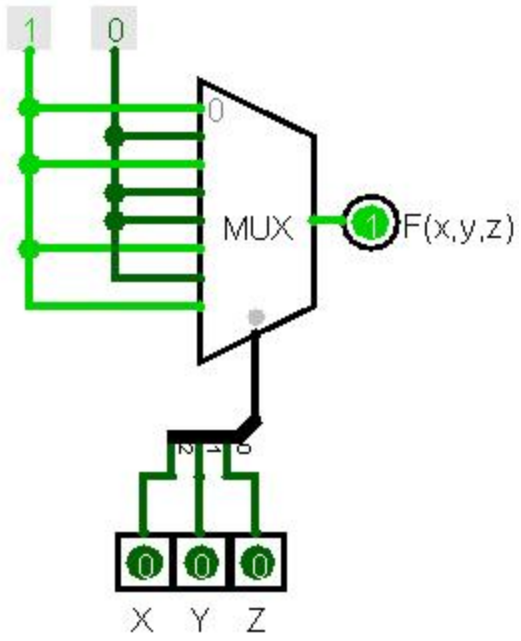
Understanding the Boolean expression, $f(x,y,z) = \Sigma(0,2,5,7)$, is fundamental before designing a circuit, it states that the expected output for the 8-to-1 multiplexer should be "1" when the given minterms $\Sigma(0,2,5,7)$ are selected by three control signals (x,y,z).

Inputs to the 8-to-1 multiplexer are provided by +5V and GRND signals from the power supply (corresponding to "1" and "0") and appropriately connected on the breadboard to voltage rails where there are two such connections, positive and negative voltage.

The truth table for the Boolean expression, $f(x,y,z) = \Sigma(0,2,5,7)$, is shown below…

X	Y	Z	F(x,y,z)
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

The truth table produces the following logic diagram (note the use of constants to define +5V and GRND)…



Part B

For any *M* size multiplexer, *N* control signals are required. Where *N* is derived from the equation $M = 2^N$. Furthermore, any *M* size multiplexer can reproduce the functionality of a 2^{N+1} multiplexer if the additional (*N+1*) control signal (the least significant bit) is represented as an input to the multiplexer.

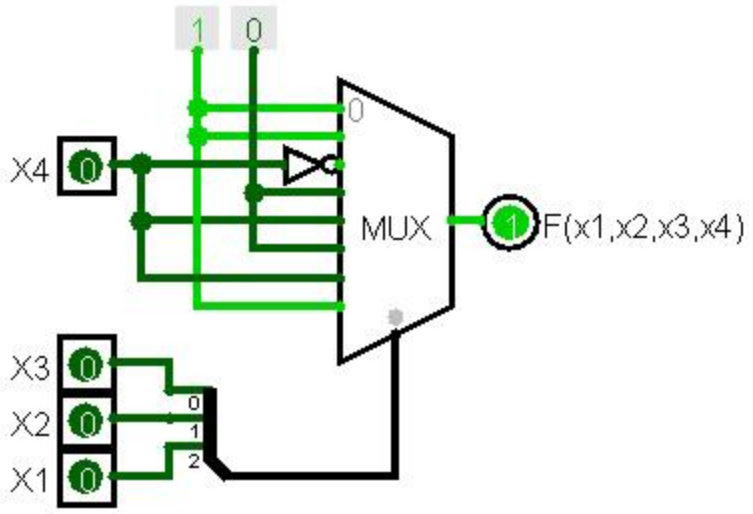
In order to represent 16 possible outcomes ($f(x1, x2, x3, x4)$) for the 8-to-1 multiplexer the least significant bit ($x4$) must be used as an input to the multiplexer to produce the minterms $\Sigma(0, 1, 2, 3, 4, 9, 13, 14, 15)$.

The truth table for the Boolean expression is shown below (note the table is separated by control signals, input, and output)...

x1	x2	x3	x4	F(x1,x2,x3,x4)
0	0	0	0	1
			1	
0	0	1	0	1
			1	
0	1	0	0	x4'
			1	
0	1	1	0	0
			1	
1	0	0	0	x4

			1	
1	0	1	0	0
			1	
1	1	0	0	x4
			1	
1	1	1	0	1
			1	

The truth table produces the following logic diagram…



Part C

Designing a circuit that compares two 3-bit binary numbers and produces a 2-bit output would require a 64 row truth table (2^6), while possible this approach is very tedious and proved to be error prone. The presence of six variables (A_2, A_1, A_0 *and* B_2, B_1, B_0) makes Boolean expressions difficult to simplify if using ordinary Karnaugh map strategy, therefore the "Two-Level Simplification" approach is preferable.

However for the specific problem of bit comparisons only two bits need to be compared at once before continuing the rest of the bit comparisons. The first bit-to-bit comparison should start with the most significant bits (A_2 *and* B_2) and then proceed to the least significant bits (A_0 *and* B_0) if and only if the more significant bits directly preceding are equivalent, else one of the two bits being compared is greater than the other.

Boolean expressions for the comparison of two 1-bit binary numbers is given by the truth table shown below...

A	B	A = B	A < B	A > B
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

$A = B:\ A'B' + AB = A\ XNOR\ B$

$A < B:\ A'B$

$A > B:\ AB'$

Now that the logic for a bit-to-bit comparison has been presented it's possible to define a simplified truth table for the circuit and generate the appropriate Boolean expressions for the 2-bit output.

Note that *Q2* and *Q1* are the bit representations for the 2-bit output.

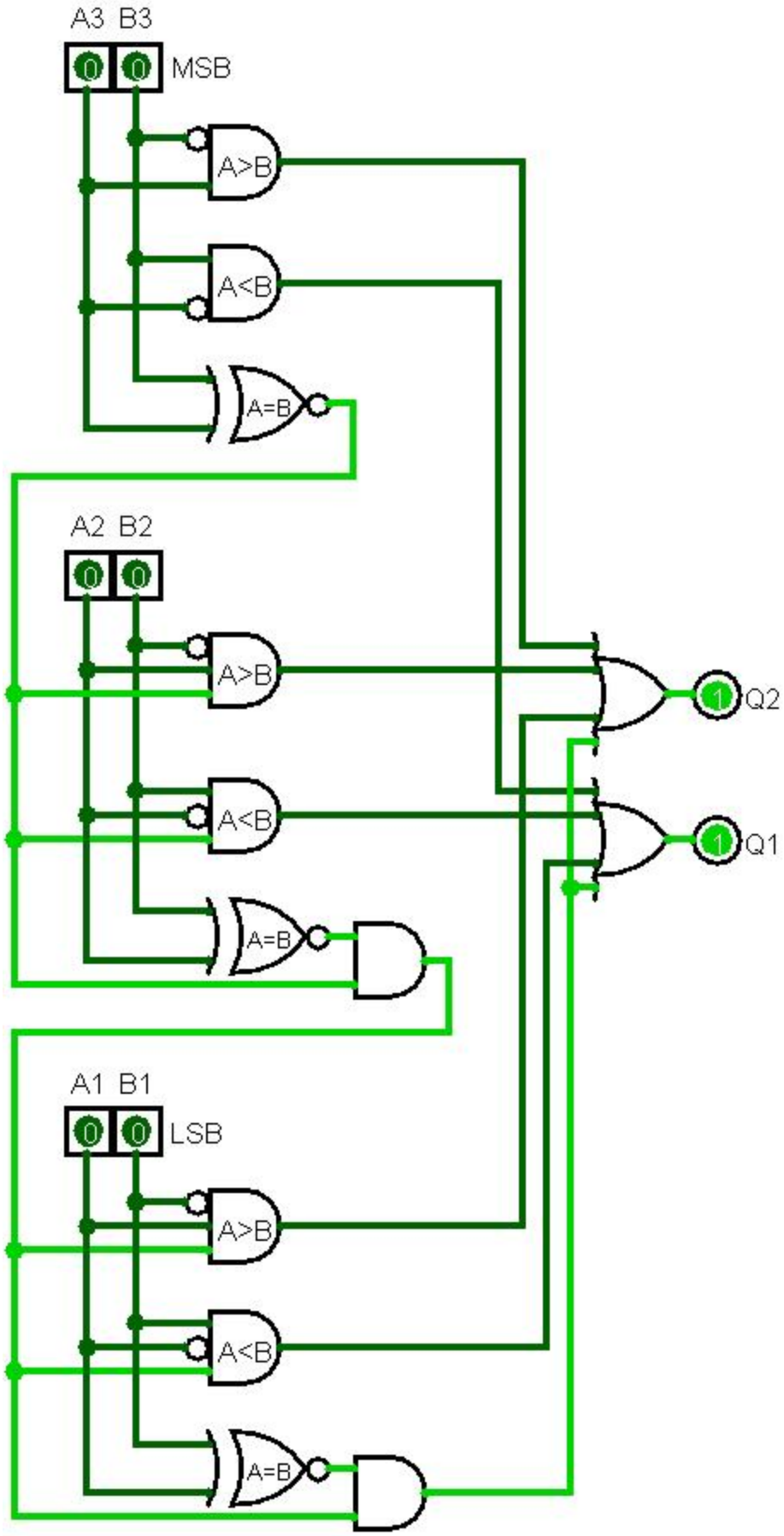
Q2	Q1	Evaluation
0	0	x
0	1	*A < B*
1	0	*A > B*
1	1	*A = B*

A3/B3	A2/B2	A1/B1	Q2	Q1
A3 < B3	x	x	0	1
A3 > B3	x	x	1	0
A3 = B3	A2 < B2	x	0	1
A3 = B3	A2 > B2	x	1	0
A3 = B3	A2 = B2	A1 < B1	0	1
A3 = B3	A2 = B2	A1 > B1	1	0
A3 = B3	A2 = B2	A1 = B1	1	1

$$Q2 = A_3 > B_3 + (A_3 = B_3)(A_2 > B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 > B_1) + (A_3 = B_3)(A_2 = B_2)(A_1 = B_1)$$

$$Q1 = A_3 < B_3 + (A_3 = B_3)(A_2 < B_2) + (A_3 = B_3)(A_2 = B_2)(A_1 < B_1) + (A_3 = B_3)(A_2 = B_2)(A_1 = B_1)$$

The logic diagram depicted below was created from the Boolean expressions above…


A3 B3
0 0
MSB
A>B
A<B
A=B
A2 B2
0 0
A>B
A<B
A=B
1
Q2
1
Q1
A1 B1
0 0
LSB
A>B
A<B
A=B

OBSERVATIONS AND DATA ANALYSIS

Using the knowledge gained from Part A and Part B, our group questioned the possibility of re-creating the logic design for Part C using the fundamental principles of multiplexers.

Given the requirement of a 2-bit output it was concluded that at minimum two multiplexers were necessary to correctly display the differing outcomes for bit comparisons, but how do we know what size to make each multiplexer?

Remembering that any *M* size multiplexer can reproduce the functionality of a 2^{N+1} multiplexer if the least significant bit (*N+1*) is used as an input then the common control signals would be bits A_2, B_2, A_1, B_1 for both multiplexers leading to a size of 16-to-1.

Least significant bit A_0 will drive the first 16-to-1 multiplexer and its output represents the high-order bit, *Q2*, of the 2-bit output. While B_0 will drive the second 16-to-1 multiplexer and its output represents the low-order bit, *Q1*, of the 2-bit output.

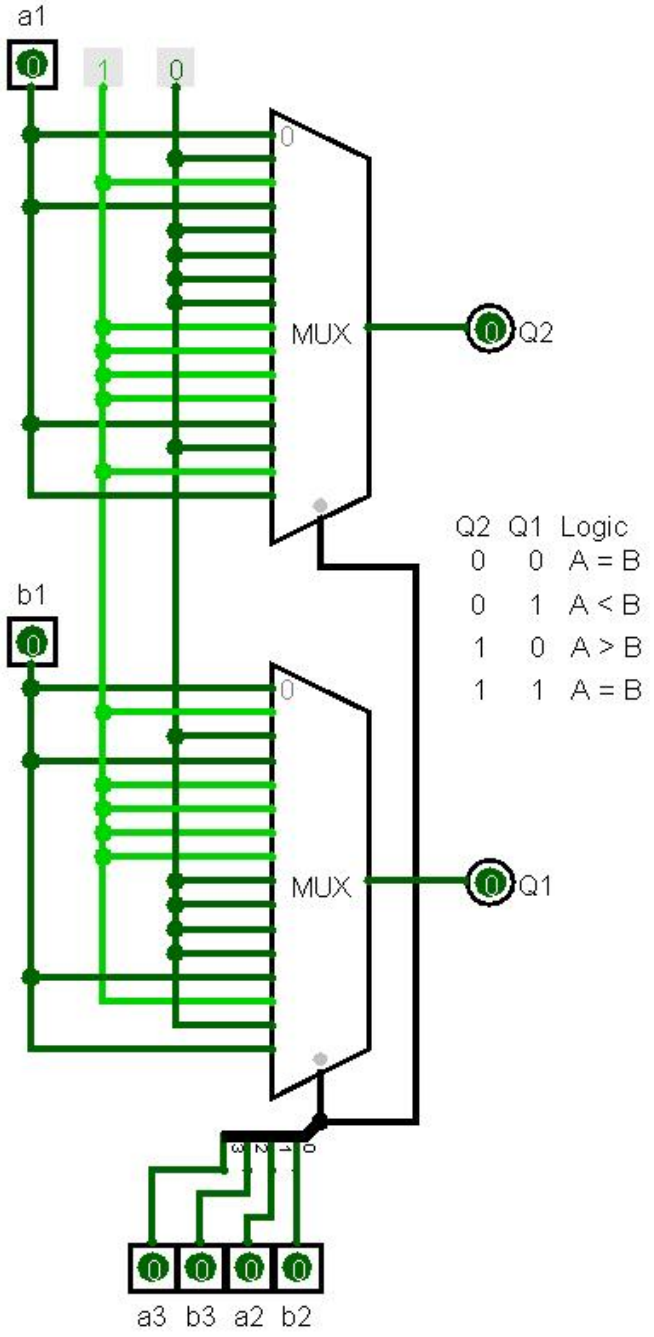
An appropriate truth table to describe the logic is provided below…

A3	B3	A2	B2	A1	B1	Q2	Q1
0	0	0	0	0 / 1	0 / 1	A1	B1
0	0	0	1	0 / 1	0 / 1	0	1
0	0	1	0	0 / 1	0 / 1	1	0
0	0	1	1	0 / 1	0 / 1	A1	B1
0	1	0	0	0 / 1	0 / 1	0	1
0	1	0	1	0 / 1	0 / 1	0	1
0	1	1	0	0 / 1	0 / 1	0	1
0	1	1	1	0 / 1	0 / 1	0	1
1	0	0	0	0 / 1	0 / 1	1	0
1	0	0	1	0 / 1	0 / 1	1	0
1	0	1	0	0 / 1	0 / 1	1	0
1	0	1	1	0 / 1	0 / 1	1	0
1	1	0	0	0 / 1	0 / 1	A1	B1
1	1	0	1	0 / 1	0 / 1	0	1
1	1	1	0	0 / 1	0 / 1	1	0
1	1	1	1	0 / 1	0 / 1	A1	B1

The truth table accounts for all 64 possible configurations of input signals and the bit representation for the 2-bit output is shown below…

Q2	Q1	Evaluation
0	0	$A = B$
0	1	$A < B$
1	0	$A > B$
1	1	$A = B$

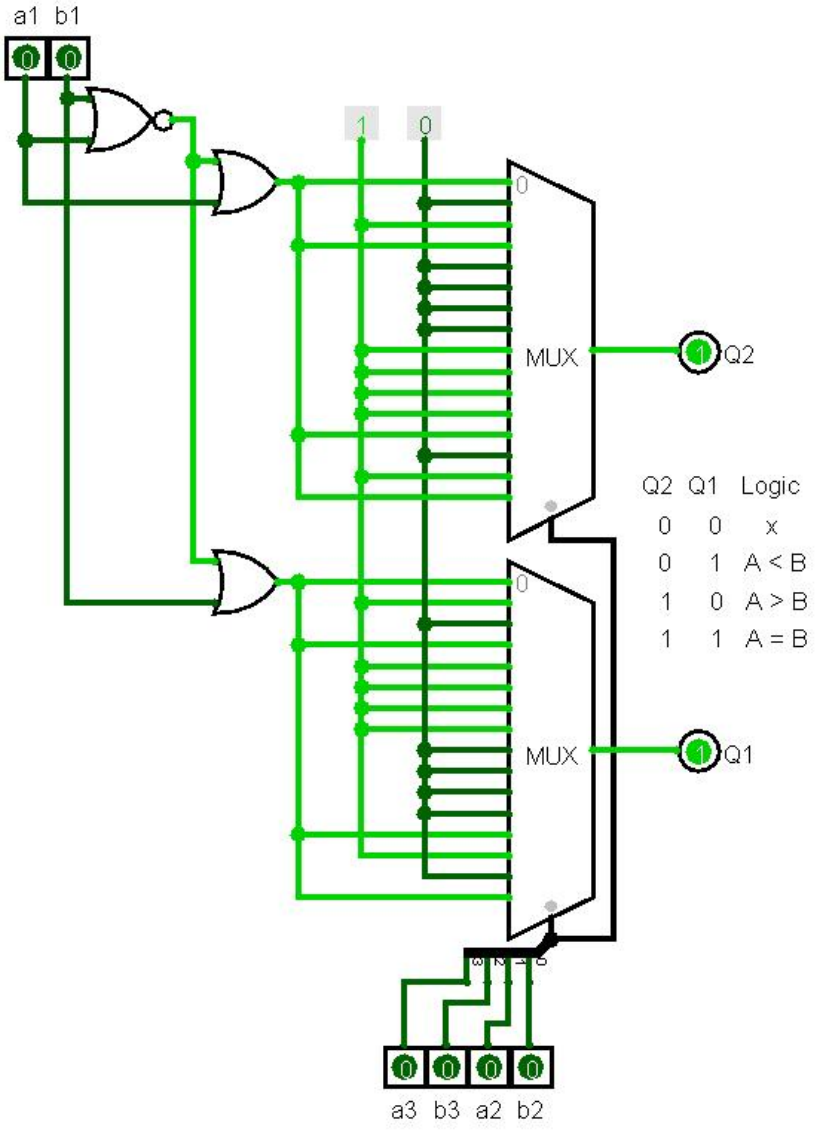
Accordingly a logic diagram for the circuit is depicted below…



While the circuit correctly displays *Q2* and *Q1* their bit representations differ from the original configuration of…

Q2	Q1	Evaluation
0	0	x
0	1	*A < B*
1	0	*A > B*
1	1	*A = B*

To correctly display *Q2* and *Q1* a slight modification was necessary, such that when both least significant bits A_0 and B_0 are "0" their input to the multiplexers should reflect a "1". This logic is comparable to a NOR gate where the Boolean expression is A'B'. The following logic diagram depicting this modification is shown below…



DISCUSSION AND CONCLUSION

Lab number 3 was a great introduction into the applications and properties of multiplexers. While Part C of the lab could have been accomplished using a single integrated circuit, namely the 74LS85 TTL, the process of deriving our own logic was very beneficial to improving the problem solving strategy. This was the first lab were starting with a truth table and simplifying using Karnaugh maps would not have been the easiest method, instead a fundamental understanding of the problem was necessary.

It was also proven that traditional logic structures can be realized using multiplexers, as shown for Part C in the Observations and Analysis section. Given the small number of bit comparisons the use of multiplexers was appropriate however if the number of bit comparisons were larger than 3-bits then the size of the multiplexers would increase and eventually (due to size constraints for multiplexer ICs) the number of multiplexers would increase drastically. On the other hand a 4-bit comparator like the 74LS85 can be combined with any number of 74LS85 chips to provide bit comparisons for larger words.

REFERENCES

"The Multiplexer (MUX)." *Basic Electronics Tutorials*. N.p., n.d. Web. 5 June 2014. <http://www.electronics-tutorials.ws/combination/comb_2.html>.