Lab 3 Lecture Notes - Introduction to Arithmetic circuits

Part 1. Introduction to Multiplexers

The multiplexer is one of the basic building blocks of any digital design system. What it does is it takes a number of inputs and multiplexes them onto a single output line. That is, it selects one of the input lines, and passes its state to the output line.

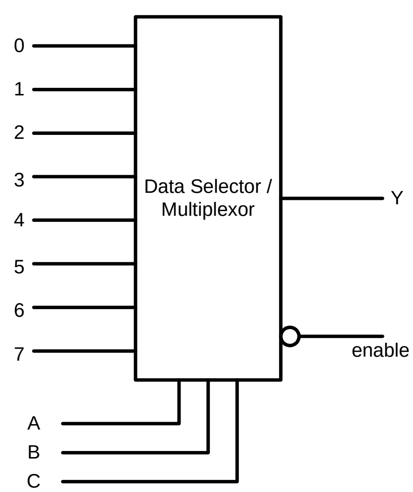

Figure 1: A Multiplexer

The input on the A, B, and C lines tells the multiplexer which one of the inputs 0-7 to pass to the output. In the image above A=0, B=1, and C=0, so the input line selected to be passed to the output would be line 2.

This property of multiplexers is very useful. As you will see later on, we can use a multiplexer to implement complex logic functions. In terms of practical lab experience, it allows one chip, the multiplexer, to do the job of several simple logic gates. Later on, you will see how to use the multiplexer to implement binary addition and subtraction. However, before you can do this, you need to know something about binary arithmetic.

Part 2. Digital Arithmetic - Adition

This section discusses the basics of digital addition. Here is a block diagram of a binary adder:

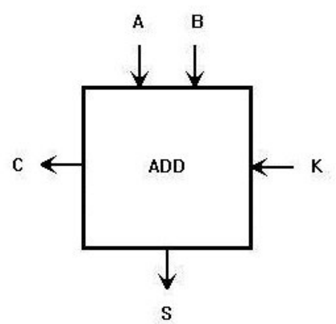

Figure 2. Adder block diagram

From this diagram, you can see that the adder has three inputs and two outputs. This will mean that one must generate two different K-maps, one for each output. Here are the functions each line performs:

- A,B - The two numbers to be added
- K- Carry-in
- C- Carry-out
- S- Sum

A B K	S	C
0 0 0	0	0
0 0 1	1	0
0 1 0	1	0
0 1 1	0	1
1 0 0	1	0
1 0 1	0	1
1 1 0	0	1
1 1 1	1	1

Figure 3. Full adder truth table

Using Multiplexers to Design a Binary Adder

Instead of using the 8-to-1 MUX introduced in Part 1, the 4-to-1 MUX will be used to implement the binary adder. The following is the logic diagram of the 4-to-1 MUX.

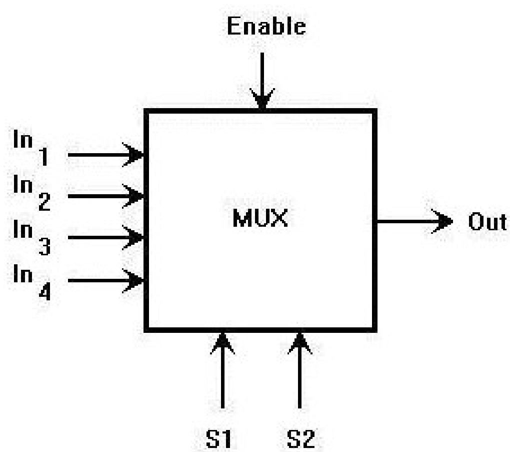

Figure 4. Adder implemented using a mux

Design process

Using the truth table shown in figure 3, generate two K-maps; one K-map for the Sum, and one for the Carry Out.

		B K			
		00	01	11	10
A	0	0	1	0	1
	1	1	0	1	0

Figure 5. K-map for the Sum

Let's take a few moments to inspect this Karnaugh map. Note that the Sum is A when B and K are both true and false. The Sum is the complement of A when B or K is true, but not both.

Now, how can we use this knowledge to create the Sum from a 4-to-1 MUX? Recall that the output is either A or the complement of A. This makes things easier. Also, notice how the 4-to-1 MUX has two control lines, S1, and S2. Recall that the value on the control lines determines which input line is passed to the output.

Here is the solution:

If we tie B to S1 and K to S2, then we can tie A to input lines 0and 3 (00 and 11). Then we must tie the complement of A to input lines 1 and 2 (01 and 10). This allows us to use a 4-to-1 MUX to produce the Sum. Figure 6 shows a block diagram for the sum implemented using 4x1 Mux.

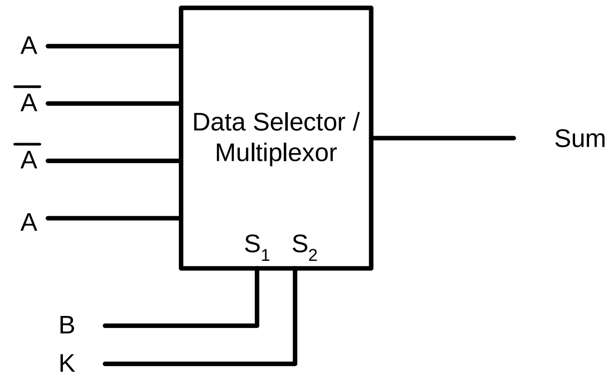

Figure 6. Block diagram for SUM

Now, we have to address the Carry Out output.

		B K			
		00	01	11	10
A	0	0	0	1	0
	1	0	1	1	1

Figure 7. K-map for the Carry-Out.

Now, inspect the K-map shown in figure 7 for a moment. Think of how we could use this K-map to produce the Carry-Out using a 4-to-1 MUX.

Note that when B and K are both false, then the output is also false. Note that when B and K are both true, then the output is also true. Finally, note that when B or K are true, but not both, the output is A.

So, if we again apply B to S1 and K to S2, we can generate the correct Carry-Out by tying input line 0 to ground, input line 3 to Vcc, and both input lines 1 and 2 to A. This will generate the correct output. And, here, you can see that we can build a simple one-bit adder using just two 4-to-1 MUXes. The block diagram for the carry out is shown in figure 8.

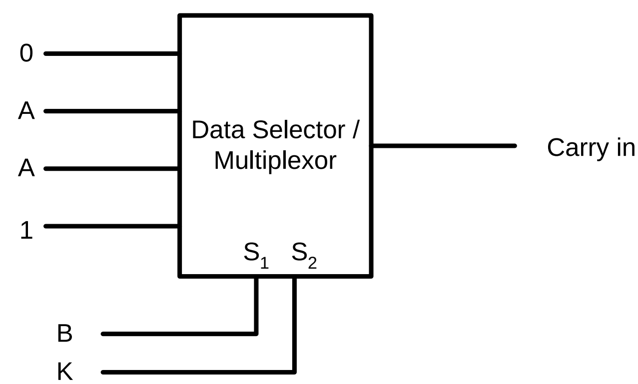

Figure 8. Carry out block diagram

If you want to add bigger numbers, you can cascade several binary adders together to produce the sum.

Part 3. Digital Arithmetic - Substraction

This section discusses the basics of digital subtraction. Here is a block diagram of a binary subtractor.

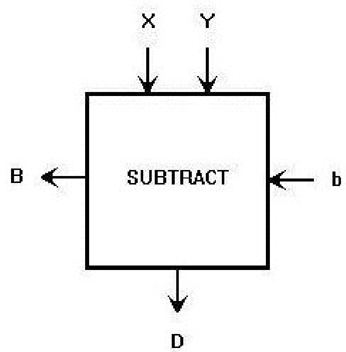

Figure 9. Subtrator block diagram.

From figure 9, you can see that the subtractor has three inputs and two outputs. This will mean that one must generate two different K-maps, one for each output. Here are the functions each line performs.

- X,Y - The two numbers to be subtracted.
- b - Borrow-in.
- B - Borrow-out.
- D – Difference.

This subtractor will work on two one-bit numbers X and Y. Figure 9 show the truth table for the subtractor.

X Y b	D	B
0 0 0	0	0
0 0 1	1	1
0 1 0	1	1
0 1 1	0	1
1 0 0	1	0
1 0 1	0	0
1 1 0	0	0
1 1 1	1	1

Figure 9. Subtractor truth table.