CMSC B240 Computer Organization - Spring 2025
Lab Activity #4 Solutions

Question #1

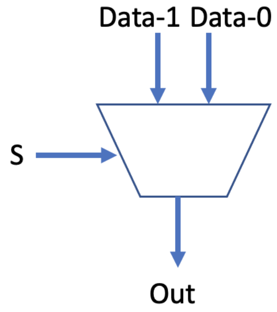
Recall that a **multiplexer,** or “mux”, is a circuit that has 2^N data lines, and uses the value represented by the N select lines to choose which data line to produce as its single output.

Assume you have a 2-way mux (i.e., a mux with two data lines, one select line, and one output) as an individual component. First, complete the table with the output for each input combination:

Data-1	Data-0	Select	Out
0	0	0	**0**
0	0	1	**0**
0	1	0	**1**
0	1	1	**0**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

How could you combine some number of 2-way muxes to create a **4-way mux**, i.e. one that has four data lines, two select lines, and one output?

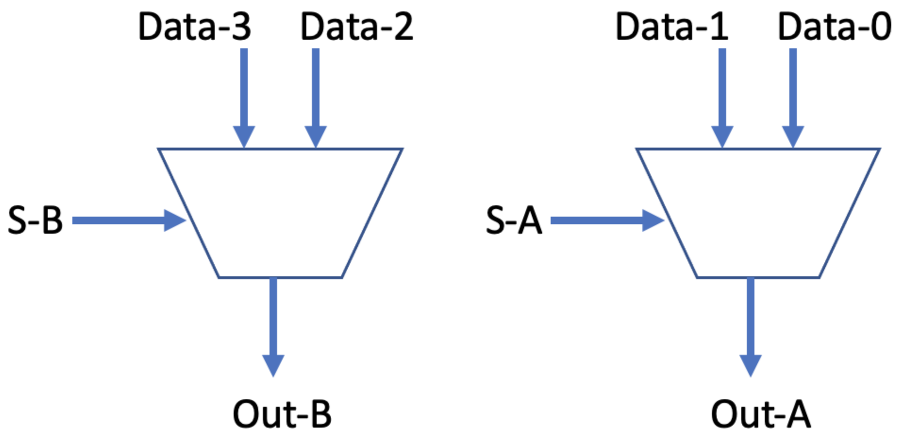
Using a representation like this for your 2-way mux:



draw the circuit diagram for a 4-way mux. You may use NOT, OR, and AND gates as needed, but should attempt to design this circuit using only a combination of 2-way muxes.

Clearly we’ll need at least two 2-way muxes, since we know we’ll have four data lines to choose from. For the first, we’ll call the inputs Data-1 and Data-0, the select line S-A, and its output Out-

A; for the second, we'll call the inputs Data-3 and Data-2, the select line S-B, and its output Out-B:



Now let's say that our 4-to-1 mux will have two select lines S-1 and S-0, and we want the output of the 4-to-1 mux (which we'll call Out) to be as follows according to this truth table:

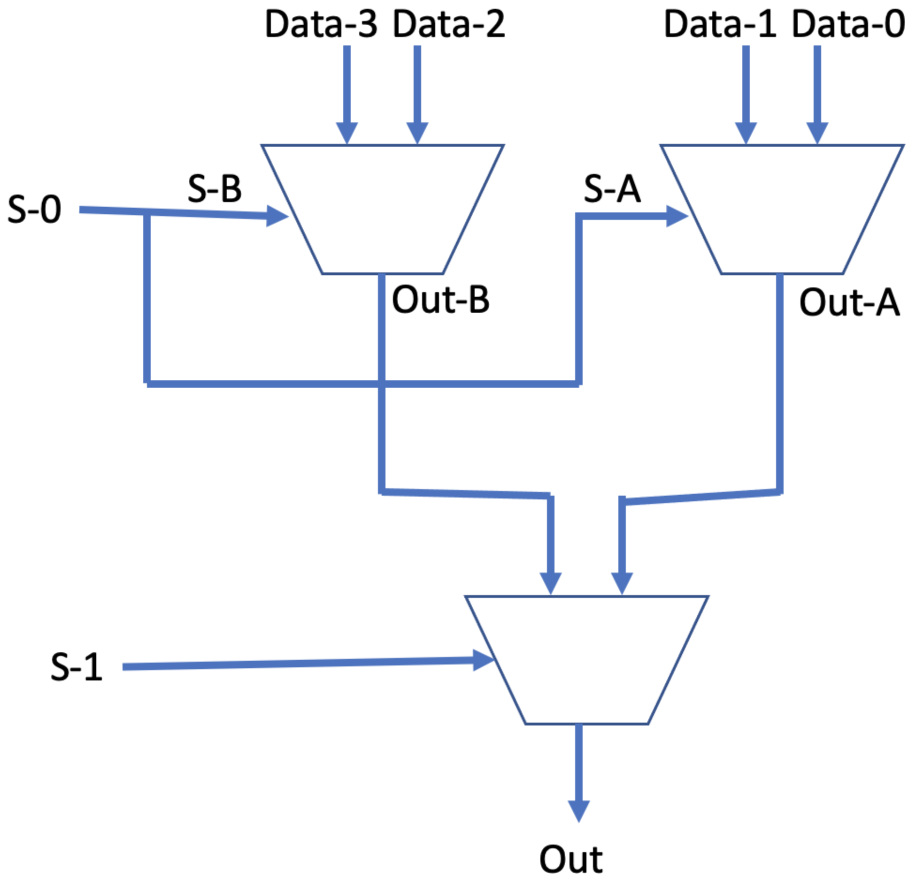
S-1	**S-0**	**Out**
0	0	Value of Data-0
0	1	Value of Data-1
1	0	Value of Data-2
1	1	Value of Data-3

Note that S-1 controls which mux produces the output: when S-1 is 0, the output of the 4-way mux comes from the first 2-way mux (and is equal to Out-A); when S-1 is 1, the output of the 4-way mux comes from the second 2-way mux (and is equal to Out-B).

Note also that S-0 chooses which line of the 2-way mux is used as the output. Therefore, we can use S-0 as the select line for **both** 2-way muxes.

What do we do with their outputs? When S-1 is 0, we use Out-A; when S-1 is 1, we use Out-B. So S-1 is being used to choose between Out-A and Out-B. What circuit can we use to do that? Another 2-way mux! That is, we can use Out-A and Out-B as inputs to a third 2-way mux, with S-1 as the select line, and the output of that mux as the output of our 4-way mux overall.

So the final diagram would look like this:


Data-3
Data-2
Data-1
Data-0
S-0
S-B
S-A
Out-B
Out-A
S-1
Out

Question #2

A **demultiplexer**, or “demux”, is similar to a mux but works in the opposite direction: it has one input line, N select lines, and 2^N output lines: the selected output line equals the input line, and the rest are 0.

Complete the truth table for a 2-way demux, i.e. a demux with N = 2.

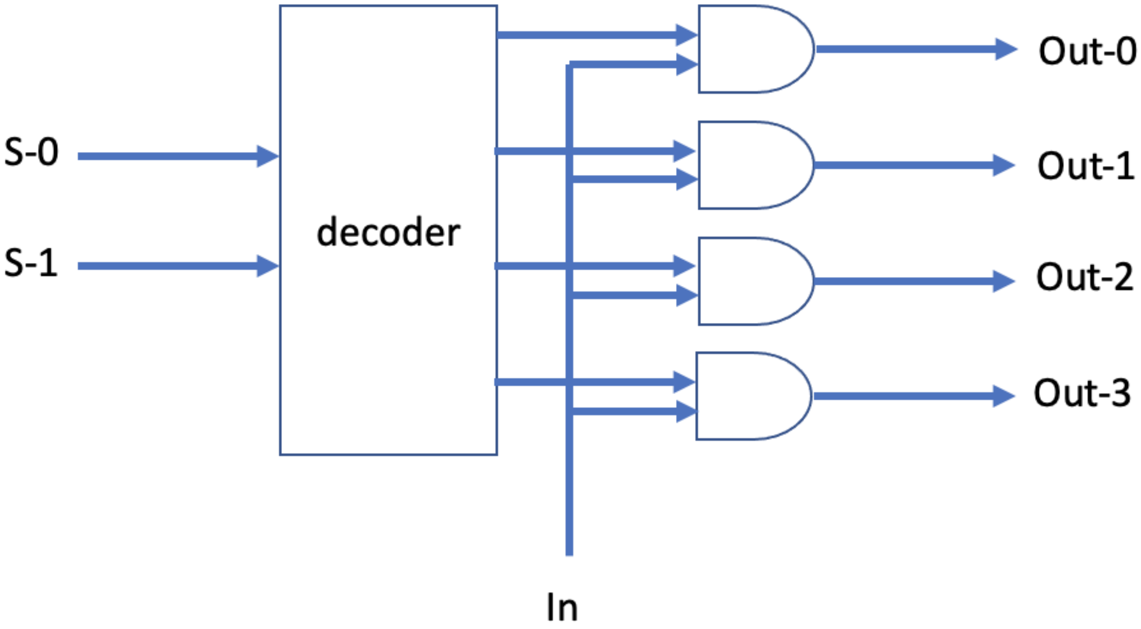
Because there is one input line and two select lines, that means we have three input combinations, so our truth table will have 2^3 = 8 rows. It will have four outputs but those don’t affect the number of rows in the table.

S-1	S-0	Data	Out-0	Out-1	Out-2	Out-3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

Using a 2-way decoder like the one we saw in class, draw the circuit diagram for a 2-way demux (*Hint!* Think about how we used a decoder to draw the circuit diagram for a multiplexer!)

Recall that the 2-way decoder would take two inputs representing a binary number K and have four outputs: the Kth output line would be 1, and the rest are 0.

We can then AND each of the decoder’s output lines with the Input to the demux, and use the results as the demux’s outputs: if the Input is 0, then all the outputs will be 0; if the Input is 1, then the Kth output line will be 1 and the rest will be 0.



Question #3

Consider a computer system that has a 32-bit address space and 2-byte addressability. How much **addressable memory** does this system have?

32-bit address → 2^32 addresses

2-byte addressability → 2 bytes/address

Total = (2^32 addresses) * (2 bytes/address) = 2^33 bytes = **8GB**

Question #4

Consider a computer system that has 8GB of addressable memory and a 32-bit address space, and assume each instruction is 16 bits long (remember, instructions are stored in memory along with data). How many **addresses** are required to hold one instruction in this system?

8GB addressable memory → 2^33 bytes

32-bit address space → 2^32 addresses

addressability = (2^33 bytes) / (2^32 addresses) = 2 bytes/address

an instruction is 16 bits = 2 bytes

need **one address** to hold this

Question #5

Consider a memory system with the following design:

- When data is read from memory, it uses 16 multiplexers to get each of the bits
- Each of the multiplexers has 24 select lines
- The same 24 select lines are used to control each of the 16 multiplexers

How many **bytes** of memory can this system store in total?

Recall that each multiplexer is able to read a single bit from 2^N data lines, one of which is chosen using N select lines.

Since each multiplexer has 24 select lines, that means N = 24 and we have 2^24 data lines into the multiplexer, which means we have 2^24 addresses.

Since we have 16 multiplexers, and each reads one bit, this means we're getting 16 bits, or 2 bytes, of data each time we read from memory.

So if we have 2^24 addresses and each holds 2 bytes, then we have 2^24 * 2 = 2^25 bytes available in memory, which is 32MB.