CSCI 2150 -- "Decoders & Demultiplexers"

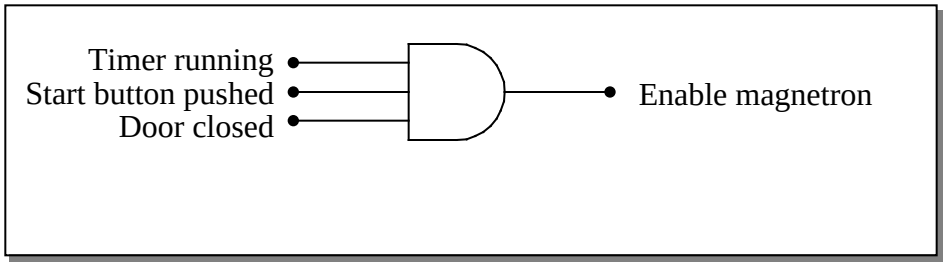
Reading: *Digital Fundamentals* section 6.8

The Need for Decoders

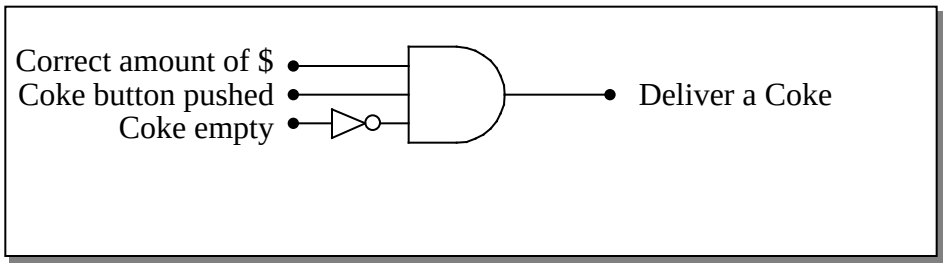
Digital signals are often used to enable something. For example, we've spoken in class of the basic design of a microwave oven. The magnetron (the mysterious device producing the electromagnetic waves) is "enabled" when the timer is running *and* the start button has been pushed *and* the oven door is closed.

This concept of enabling a device or operation based on the condition of a number of inputs is common among digital devices. Even computer memory uses this principle to determine where information is to be stored or retrieved.

In the example above, the sentence used to describe the enabling of the magnetron joined each of the inputs with the word AND. Therefore, the enabling circuit for the magnetron can be realized with an AND gate.



There are many other types of digital systems that also enable a process based on satisfying a single combination of levels from multiple inputs. For example, an automobile with a manual transmission enables the starter when the clutch is pushed in *and* the key is turned in the ignition. A Coke drops from a vending machine when a proper amount of money is inserted *and* a button is pushed *and* the machine is not out of Coke.



This works because of the definition of the AND gate. It outputs a 1 if and only if all of its inputs equal 1. If an input is inverted, the output is 1 if and only if all of the inputs without inverters equal 1 and all of the inputs with inverters equal 0.

All of these enable-type circuits follow this rule in that their truth table will have only one row with a 1. All of the other rows will have zeros.

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

When we discussed the Sum-of-Products expressions, we found that each row of a truth table that has an output of 1 has a unique product that satisfies that output. Therefore, the boolean expression that satisfies an enable circuit should consist of a single product that uses all of the inputs. For the sample truth table above, the boolean expression would be:

$$EN = \overline{A} \cdot B \cdot \overline{C}$$

Decoder circuits are simply a group of enable circuits that have an individual output that satisfies each row of the truth table. For example, for a 2-input decoder circuit with inputs A and B, there is an output that is 1 only when A=0 and B=0, an output that is 1 only when A=0 and B=1, an output that is 1 only when A=1 and B=0, and an output that is 1 only when A=1 and B=1. The boolean expressions that satisfy this decoder circuit are:

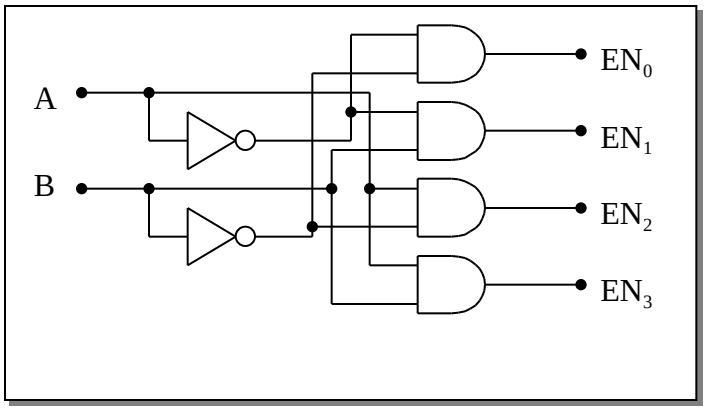
$$EN_0 = \overline{A} \cdot \overline{B}$$

$$EN_1 = \overline{A} \cdot B$$

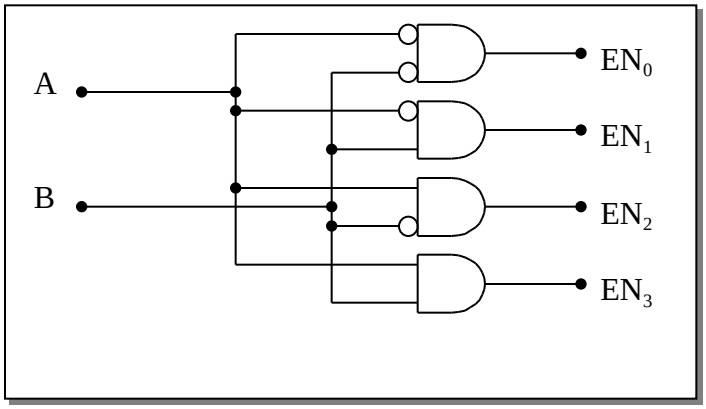
$$EN_2 = A \cdot \overline{B}$$

$$EN_3 = A \cdot B$$

As for the logic circuit, it looks something like the figure on the following page.



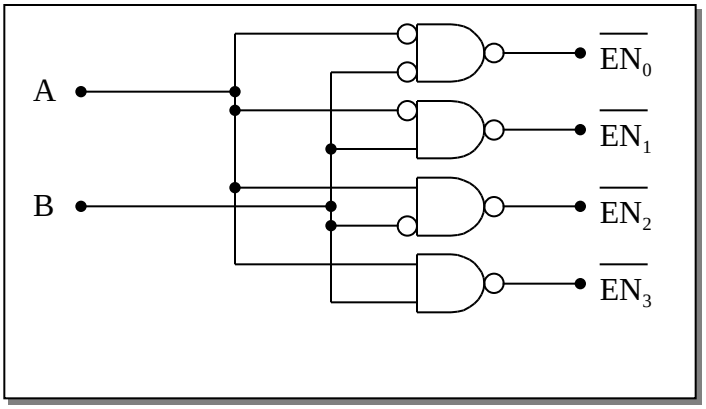
This circuit may be easier to read with the inverters replaced with circles at the inputs of the 2-input AND gates.



Active Low Signals

For a number of reasons, the standard method of enabling a device is NOT to send a logic 1 to it. Rather, the standard practice is to enable devices with a logic 0 and disable them with a logic 1. This is called ***active low*** operation, i.e., the device is active when its enable signal is low or logic 0. The device is inactive when the enable is high or logic 1.

Right away, you should see that this is merely a matter of inverting each of the enable signals. This turns our AND gates into NAND gates. This changes our 1-of-4 decoder from the previous section into the circuit shown on the following page.



Notice the bar over the output names. This bar indicates that the signal is considered active low.

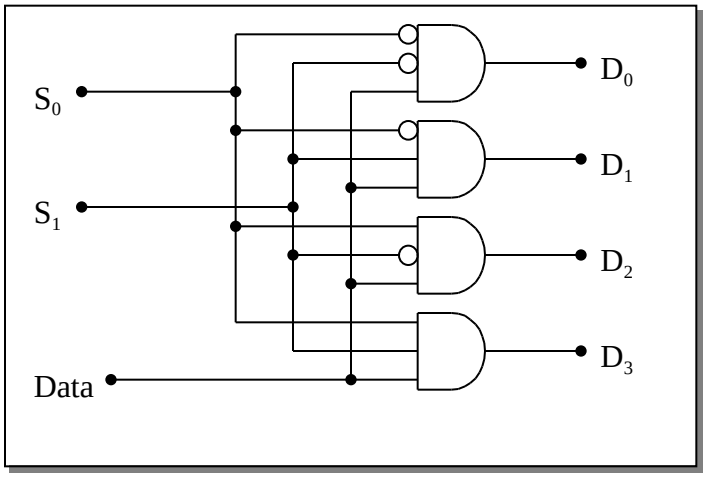
Below is a truth table for an active low 1-of-8 decoder.

A	B	C	EN_0	EN_1	EN_2	EN_3	EN_4	EN_5	EN_6	EN_7
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

Demultiplexers

Perhaps instead of enabling a device, we simply want to route a digital signal to a specific channel. For example, when you make a phone call to another phone, how does the signal manage to go to only one of the billions of phones that make up the telephone network? Demultiplexers!

The operation of a demultiplexer is much like the operation of the non-inverted decoder circuit. This circuit uses select lines (S_0, S_1, … S_n) to enable exactly one AND gate of the decoder circuit. An input is added to each of the AND gates to accept a data input. If this data input is a 1, then the output of the AND gate is a 1. If the data input is a 0, the output of the AND gate is a 0. Meanwhile, all of the other AND gates output a 0, i.e., no data is passed to them.



In effect, the select lines (S_0, S_1, … S_n) "turn on" an AND gate that passes the data through to that AND gate's output. In the figure above, if S_0=0 and S_1=1, then the outputs D_0, D_2, and D_3 are forced to have 0 outputs. D_1 will match the input from the Data line. If S_0=0, S_1=1, and Data=0, then D_1=0. If S_0=0, S_1=1, and Data=1, then D_1=1.

On the next page is the truth table for the 1-line-to-4-line demultiplexer shown in the previous figure.

S_0	S_1	Data	D_0	D_1	D_2	D_3
0	0	0	0	1	1	1
0	0	1	1	1	1	1
0	1	0	1	0	1	1
0	1	1	1	1	1	1
1	0	0	1	1	0	1
1	0	1	1	1	1	1
1	1	0	1	1	1	0
1	1	1	1	1	1	1