Address Decoders

We need some way of **enabling** (i.e., selecting) the row and column (i.e., word line and bit line) that we are interested in **reading** from, or **writing** to.

Recall if there are 2^M words in a computer memory, then each **row** can be specified with an ***M*-bit address**.

Likewise, if there are 2^N bits in a computer memory, then each **column** can be specified with an ***N*-bit address**.

Thus, we need some way of constructing an ***M*-bit row decoder**, as well as an ***N*-bit row decoder**.

The logic expression is straightforward—we wish to enable output line Y_n (or Y_m) **if** and **only if** the address bits A_0, A_1, A_2, A_3, ... have the proper value.

Example:

Consider a small amount of RAM memory consisting of just **16 memory words**. Thus, each word can be specified with only an M=**4 bit address** (i.e., 2^4 =16).

Say we wish to build an **address decoder** to select word 9 (i.e., set Y_9 =1) if, and **only** if, the address is a binary 9, i.e.:

$$A_3 = 1, A_2 = 0, A_1 = 0, A_0 = 1$$

Thus, the **Boolean Logic** description of this decoder is:

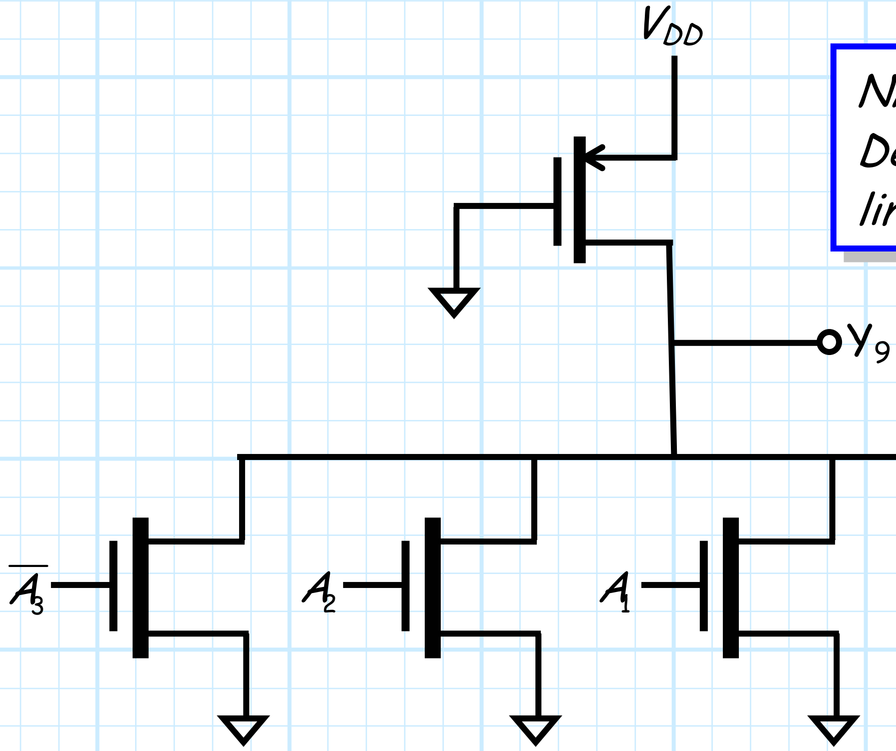
$$Y_9 = A_3 \overline{A_2}\, \overline{A_1}\, A_0$$

Likewise, for **other** word enable lines:

$$Y_0 = \overline{A_3}\, \overline{A_2}\, \overline{A_1}\, \overline{A_0}$$
$$Y_1 = \overline{A_3}\, \overline{A_2}\, \overline{A_1}\, A_0$$
$$Y_2 = \overline{A_3}\, \overline{A_2}\, A_1\, \overline{A_0}$$
$$\vdots$$
$$Y_{15} = A_3\, A_2\, A_1\, A_0$$

Q: *Hey! Don't we **know** how to build logic circuits to realize **these** Boolean expressions?*

A: Yup! We learned how to do this in **section 10.3**. Often, address decoders are complex enough that we choose to use **NMOS** technology to design them.



NMOS Address Decoder for word line 9 (of 16).

We must construct one of these decoders for **each** and **every** word line and bit line (i.e., row and column). In other words, we must construct $\mathbf{2^M}$ row decoders, and $\mathbf{2^N}$ bit column decoders.