

**Faculty of Engineering and Technology**

**Department of Electrical and Computer Engineering**

**ENCS 2110**

**EXP 4 Post-Lab: Digital Circuits Implementation using Breadboard**

**Student’s name:** Anas Al Sayed

**Student’s No.:** 1221020

**Section:** 10

**Instructor:** Hanya Radwan

* **How do you go about adding an Enable (E) signal to the decoder in Figure 4.7? Modify the implementation to show that.**

*Assuming* ***Figure 4.7 is a 2x4 decoder*** *with inputs A1 and A0 and 4 outputs (Y0–Y3), we can add an* ***Enable (E)*** *signal as follows:*

***Design Modification:***

* *Pass* ***E as an AND gate input*** *to each output.*
* *If E = 0, all outputs = 0 (disabled).*
* *If E = 1, the decoder functions normally.*
* *Y0 = E AND (NOT A1 AND NOT A0)*

*Y1 = E AND (NOT A1 AND A0)*

*Y2 = E AND (A1 AND NOT A0)*

*Y3 = E AND (A1 AND A0)*

* **How to use that to implement a 3x8 decoder using chips in Figure 1.**

***using two 2x4 decoders and one NOT gate****:*

* *Use* ***one bit as Enable*** *for two* ***2x4 decoders****.*
* *Let’s say inputs are A2, A1, A0.*

***Design:***

* *Use A2 as the* ***selector*** *between the two decoders.*
* *Connect A1 and A0 to both decoders.*
* *Connect:*
  + *First decoder's Enable to* ***NOT A2***
  + *Second decoder's Enable to* ***A2***

*This way:*

* *When A2 = 0 → Enable lower decoder* ***→*** *outputs Y0–Y3*
* *When A2 = 1 → Enable upper decoder* ***→*** *outputs Y4–Y7*
* **Use the just constructed 4x1 multiplexer to design a three-input network that gives 1 if the majority of its inputs are 1 and outputs a zero otherwise.**

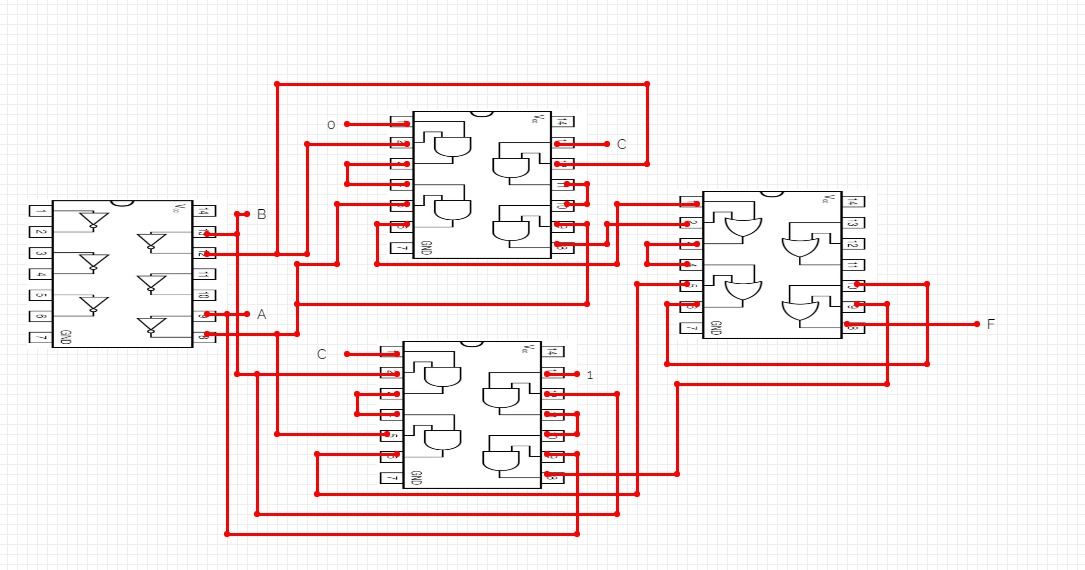


Figure 1:4x1 multiplexer

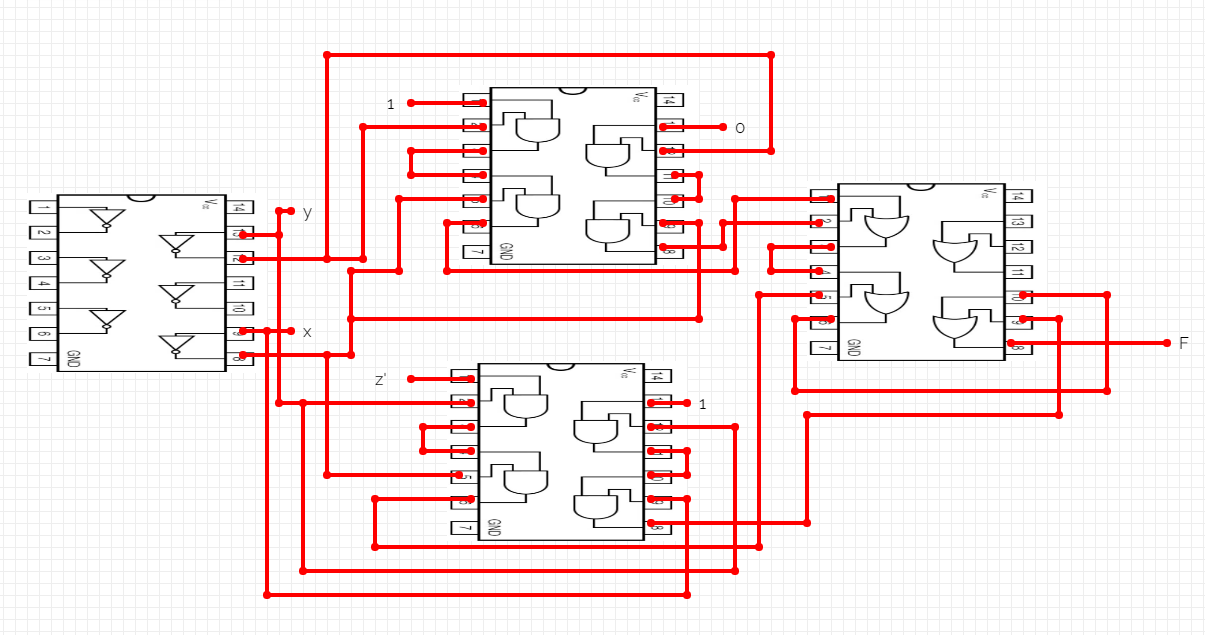
* **Implement f(x, y, z) = m(0, 1, 4, 6, 7), using 4x1 MUX using chips in Figure 1.**

Figure 2:4x1 MUX