Digital System Design

Lecture 14

Combinational Logic Design

Multiplexers, Demultiplexers and Encoders

Objectives:

1. **Multiplexers:**
   a. **4-to-1 Multiplexers.**
   b. **Design of 8:1 Multiplexers.**
2. **Demultiplexers.**
3. **Encoders.**
4. **Examples.**

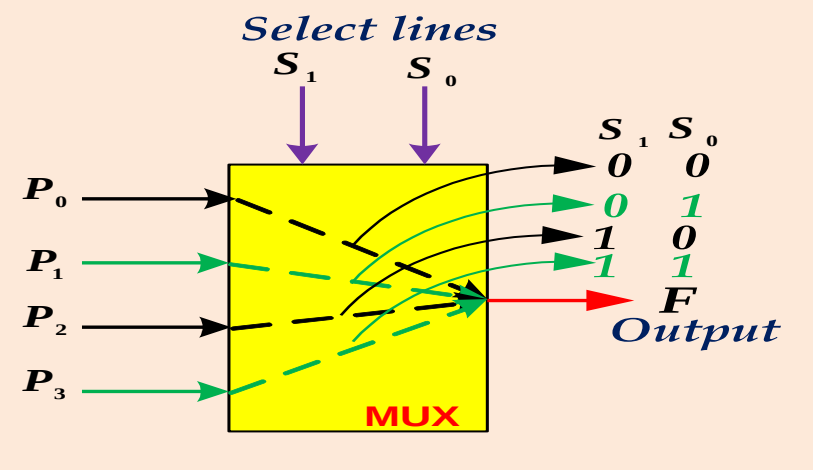
1. Multiplexers

- A Multiplexers (**MUX**) is a combinational logic component that has several inputs and only one output.
- **MUX** directs one of the inputs to its output line by using a control bit word (***selection line***) to its select lines.
- Multiplexer contains the followings:
  - 2^n data inputs
  - n selection inputs
  - a ***single*** output
  - Selection input determines the input that should be connected to the output.
- The multiplexer sometime is called data selector.
- The multiplexer acts like an electronic switch that selects one from different.
- A multiplexer may have an enable input to control the operation of the unit.

a) 4-to-1 Multiplexers

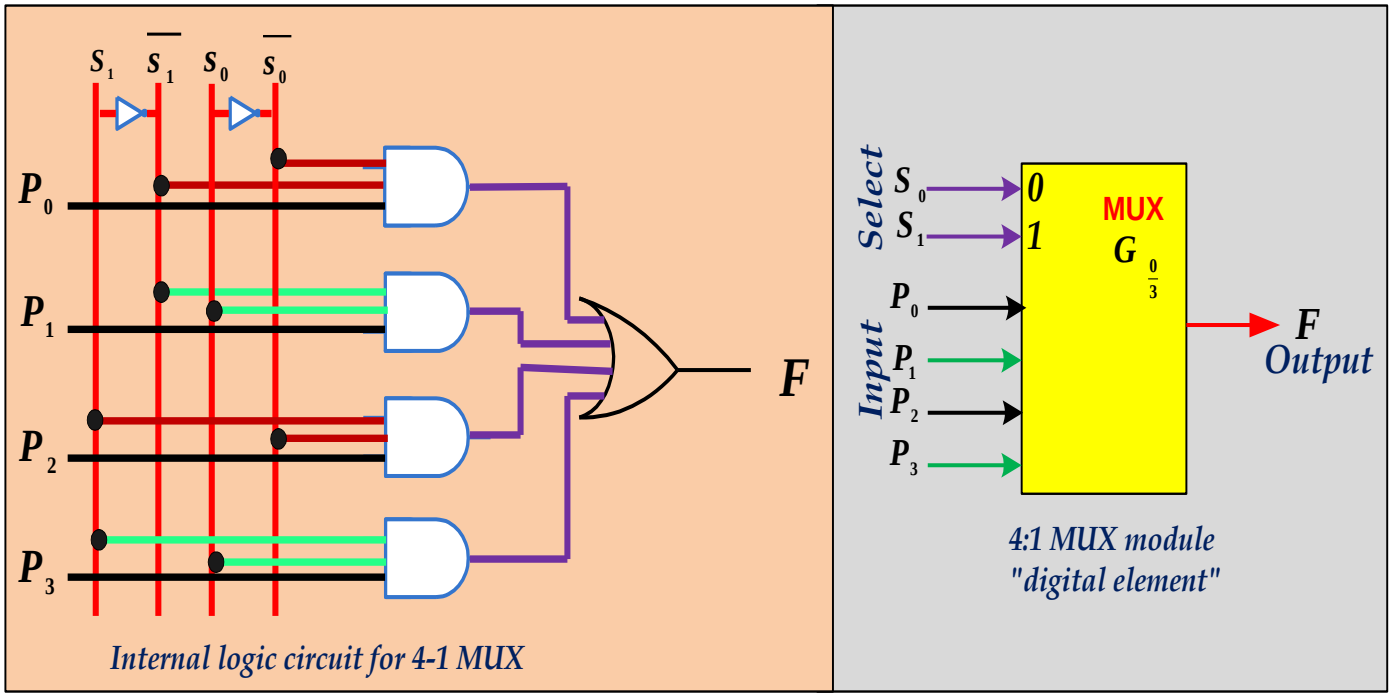
- ✓ 4-data input MUX
- ✓ s_1, s_0 - Select lines.
- ✓ p_0, p_2, p_3, p_1 - Input lines.
- ✓ F- Single output line.

Select lines		Output
S_1	S_0	F
0	0	P_0
0	1	P_1
1	0	P_2
1	1	P_3



$$F = \overline{S_1}\,\overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$

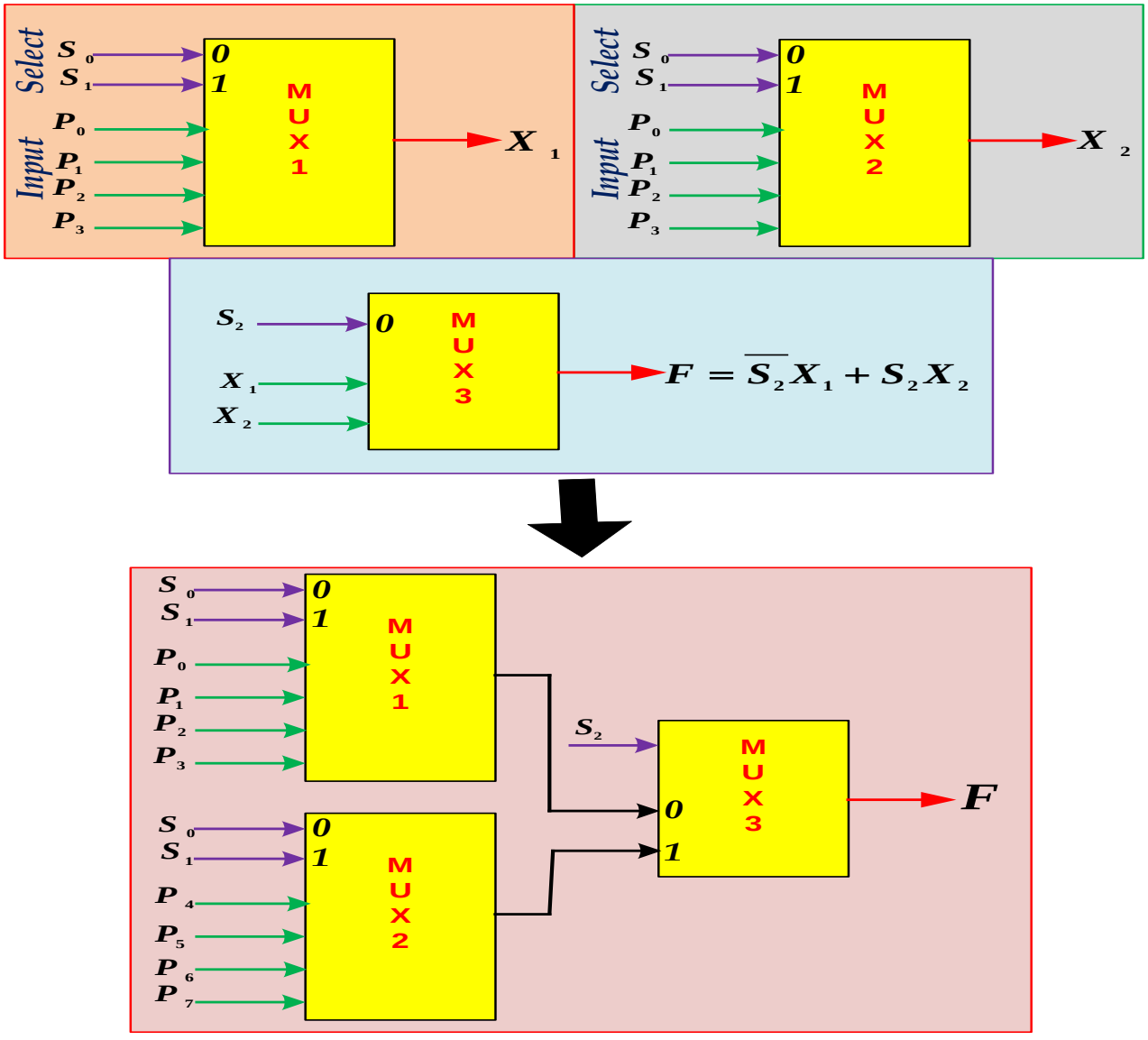
MUX implementation



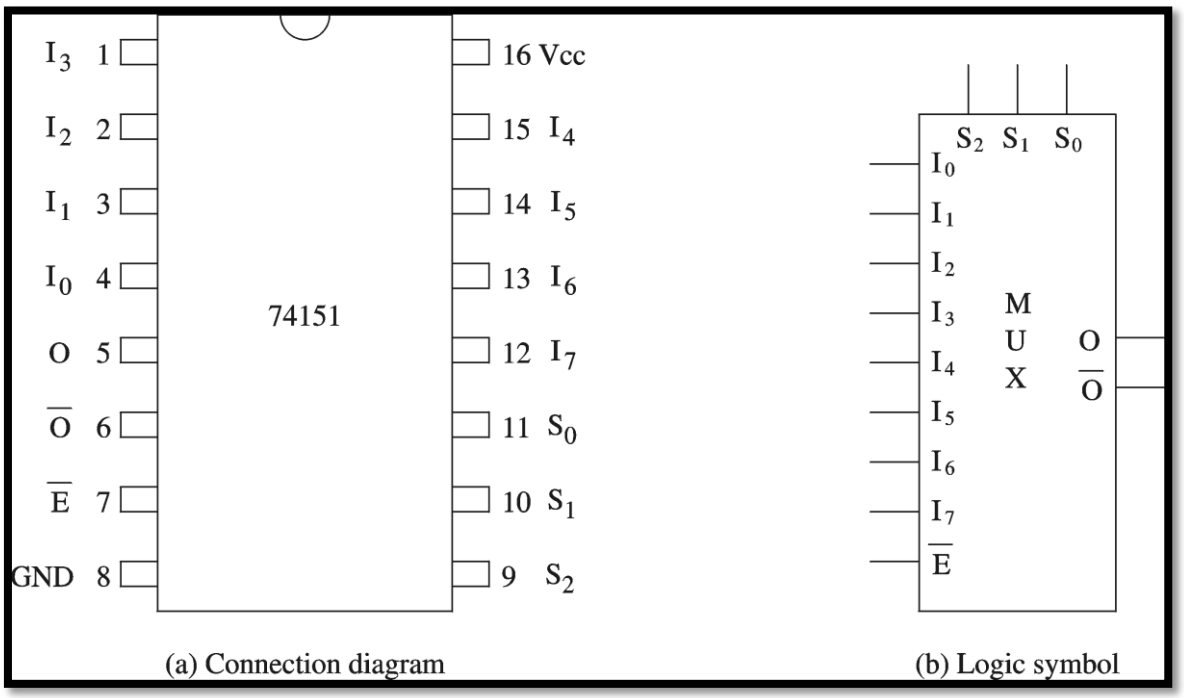
b) Design of a 8:1 multiplexer

- ➢ How to construct a ***8:1 MUX*** from two ***4:1 MUX.***
  - ✓ X_1 and X_2 are the two output lines of two ***4:1 MUX***

$$X_1 = \overline{S_1}\,\overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$
$$X_2 = \overline{S_1}\,\overline{S_0}P_0 + \overline{S_1}S_0P_1 + S_1\overline{S_0}P_2 + S_1S_2P_3$$

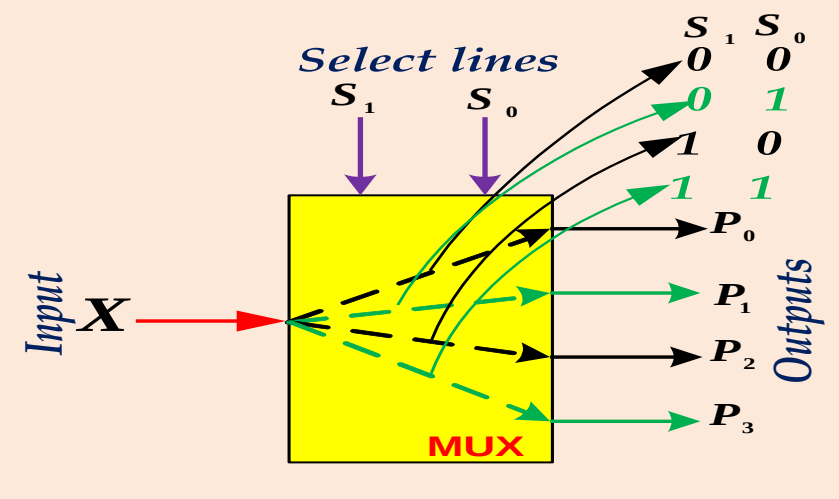


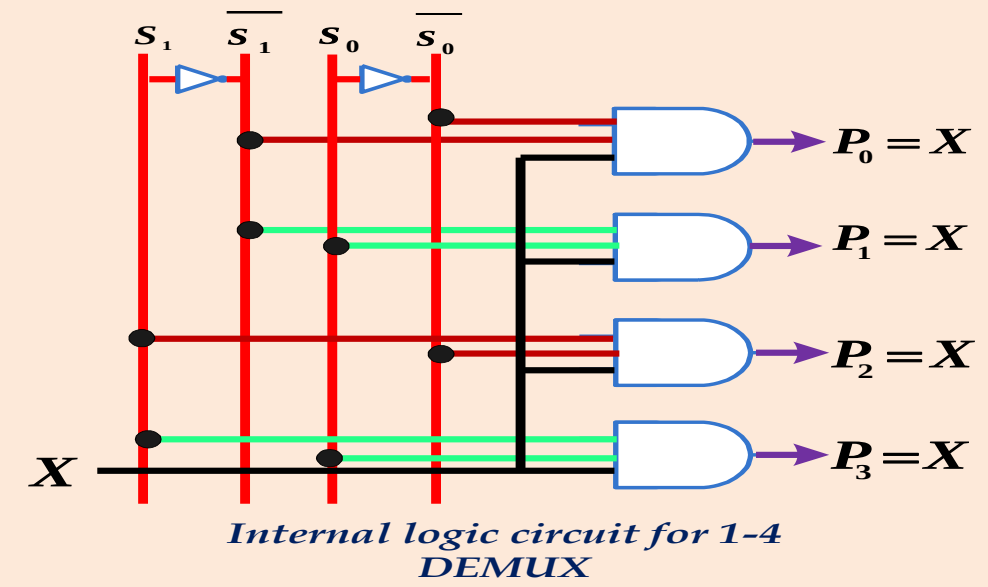
Design for a 8:1 MUX network



2. Demultiplexers

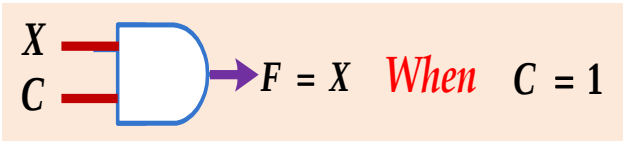
- The demultiplexer is a combinational logic circuit that performs the reverse operation of multiplexer (Several output lines, one input line).





Internal logic circuit for 1-4 DEMUX

Explain:



Two select lines four outputs

$\overline{S_1}\,\overline{S_0} = 0\,0 \Rightarrow P_0 = X$

$S_1\,\overline{S_0} = 0\,1 \Rightarrow P_1 = X$

$\overline{S_1}\,S_0 = 1\,0 \Rightarrow P_2 = X$

$S_1\,S_0 = 1\,1 \Rightarrow P_3 = X$

inputs		Output
X	C	F
0	0	0
0	1	0
1	0	0
1	1	1

3. Encoders

- An encoder is a digital circuit that performs the inverse operation of a decoder.
- An encoder has 2^n (or fewer) input lines and n output lines.
- The encoder can be implemented with ***OR gate*** whose inputs are determined directly from the truth table.

Inputs								***Outputs***		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

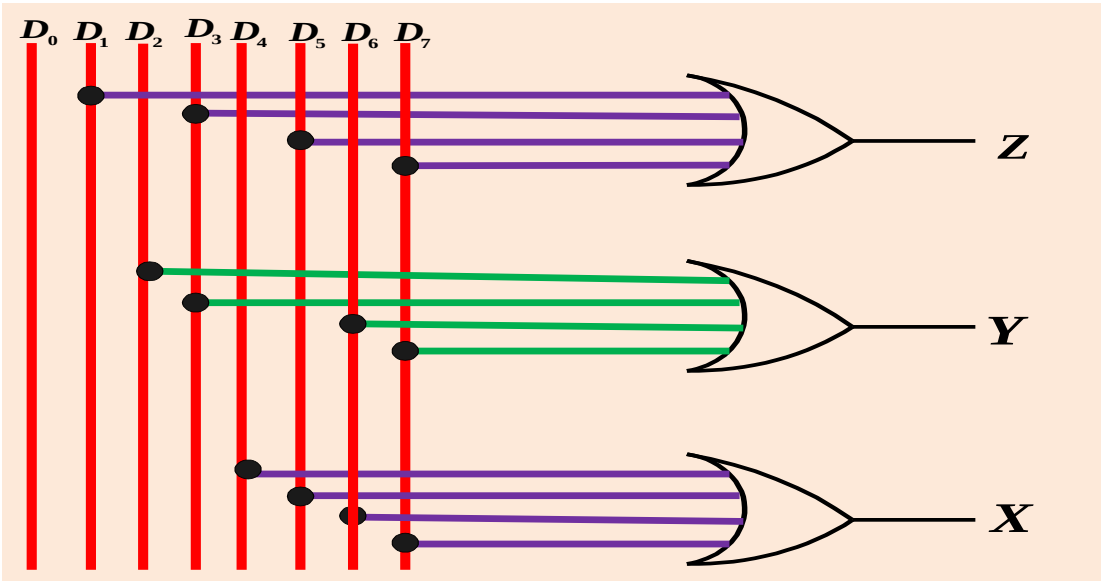
- Output X is equal to 1 when the input digit is $4, 5, 6$ or 7.
- Output Y is equal to 1 when the input digit is $2, 3, 6$ or 7.
- Output Z is equal to 1 when the input digit is $1, 3, 5$ or 7.

$$X = D_4 + D_5 + D_6 + D_7$$

$$Y = D_2 + D_3 + D_6 + D_7$$

$$Z = D_1 + D_3 + D_5 + D_7$$

- We need ***three OR gates.***



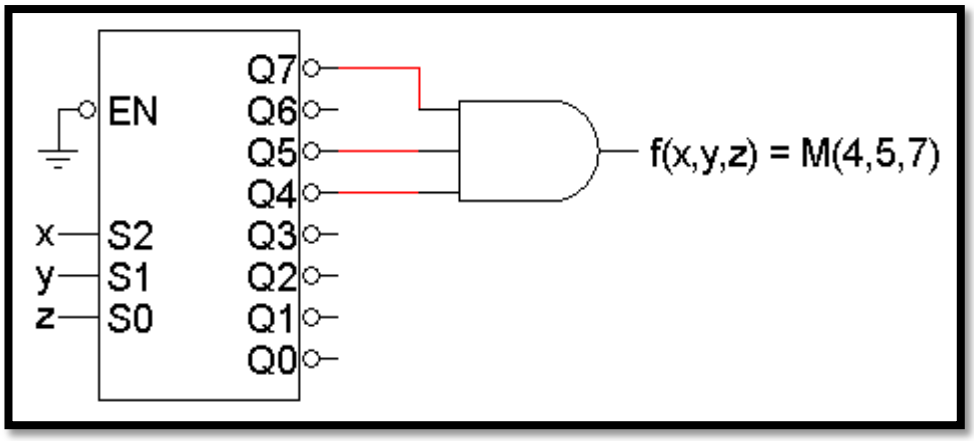
4. Examples

4.1) Active-low decoder:

- So we can use active-low decoders to implement arbitrary functions too, but as a product of maxterms.
- For example, the implementation of the function

$$f(x,y,z) = \prod M(4,5,7)$$

Using an active-low decoder is:

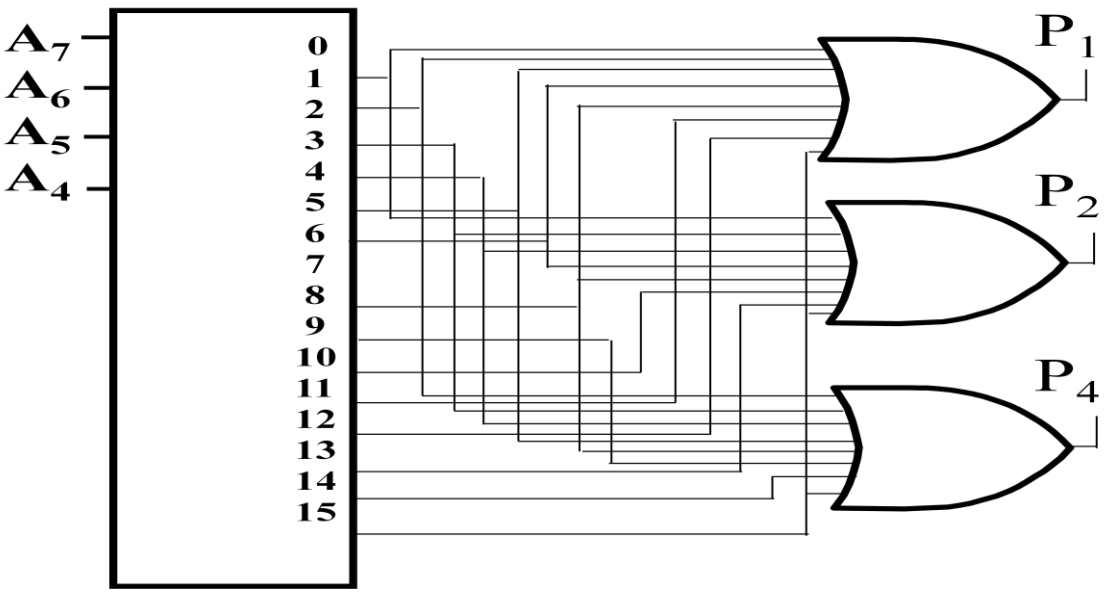


- The *"ground"* symbol connected to ***EN*** represents logical ***0***, so this decoder is always enabled.
- Remember that you need an ***AND gate*** for a product of sums.

4.2) Implement the following combinational functions using active-high decoder.

$$P_1 = \Sigma_m(1,2,5,6,8,11,12,15)$$
$$P_2 = \Sigma_m(1,3,4,6,8,10,13,15)$$
$$P_4 = \Sigma_m(2,3,4,5,8,9,14,15)$$

Solution:



4.3) A certain logic circuit has four inputs A, B, C, and D. The output X of the circuit is logic 1 if two or more inputs are logic 1.

a) Write the truth table for this circuit.

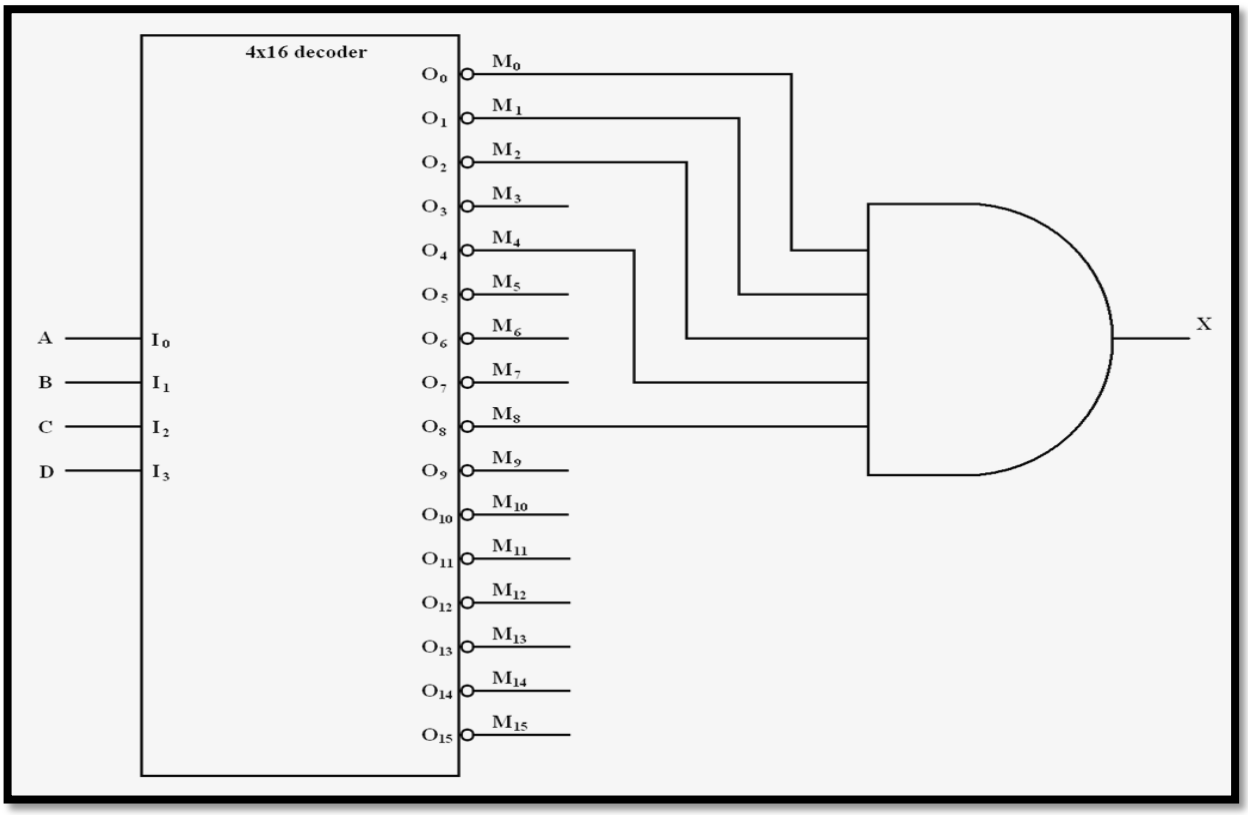
Solution:

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

b) Implement the function using one decoder with active low output.

A	B	C	D	X	Maxterm
0	0	0	0	0	$M_0 = A + B + C + D$
0	0	0	1	0	$M_1 = A + B + C + \overline{D}$
0	0	1	0	0	$M_2 = A + B + \overline{C} + D$
0	0	1	1	1	
0	1	0	0	0	$M_4 = A + \overline{B} + C + D$
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	1	
1	0	0	0	0	$M_8 = \overline{A} + B + C + D$
1	0	0	1	1	
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	1	

$$X(A,B,C,D)=\prod(M_0,M_1,M_2,M_4,M_8)$$

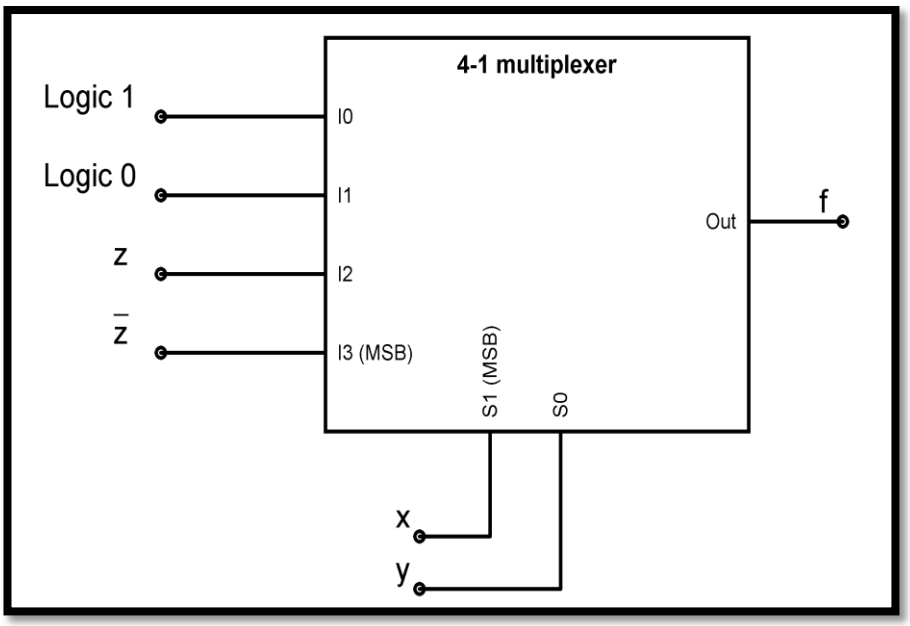


4.4) Implement the logic function f using a single multiplexer; assume that the inputs and their complements are available at the input of the multiplexer.

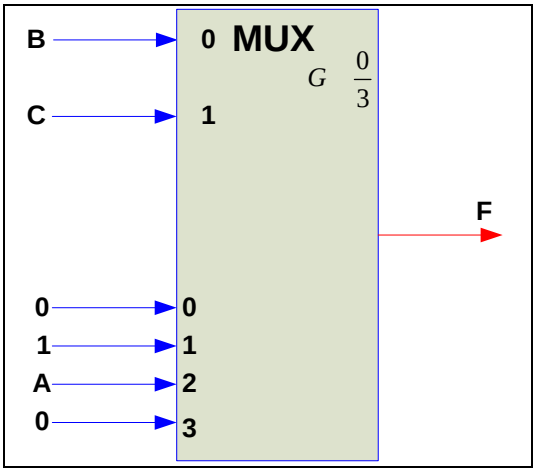
$$f(x,y,z)=\prod(2,3,4,7)$$

Solution:

x	y	z	f	
0	0	0	1	I0
0	0	1	1	
0	1	0	0	I1
0	1	1	0	
1	0	0	0	I2
1	0	1	1	
1	1	0	1	I3
1	1	1	0	



4.5) The following figure shows a 4-input multiplexer connected to implement a function F of three Boolean variables A, B, C.



Complete the following truth table:

Logical Inputs			*Output*
A	***B***	***C***	***F***
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*	*0*