

Digital Systems

Prepared by Dr. Alaa Eldeen Sayed Ahmed

IT131

Combinational Circuits – Part II

Lecture 6



Combinational Circuits Part II

- Contents

- Combinational circuits
- Decoder
- Encoder
- Multiplexer

Combinational Circuits Part II

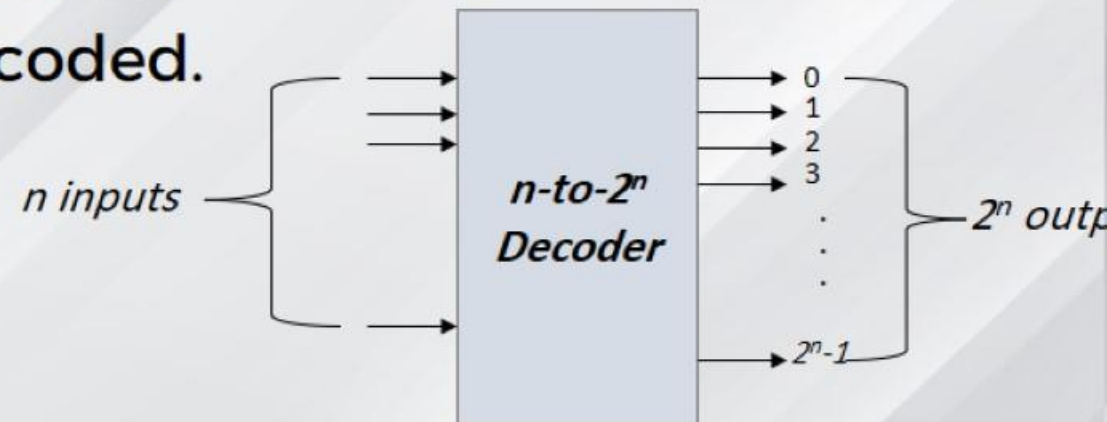
Decoder Definition

- A **decoder** is a combinational circuit that converts binary information from n input lines to a maximum of 2^n unique output lines.
- A decoder generally decodes a binary value into a non-binary one by setting exactly one of its 2^n outputs to logic "1".
- When a decoder receives n inputs, it activates one and only one of its 2^n outputs based on that input with all other outputs deactivated.

Combinational Circuits Part II

Decoding

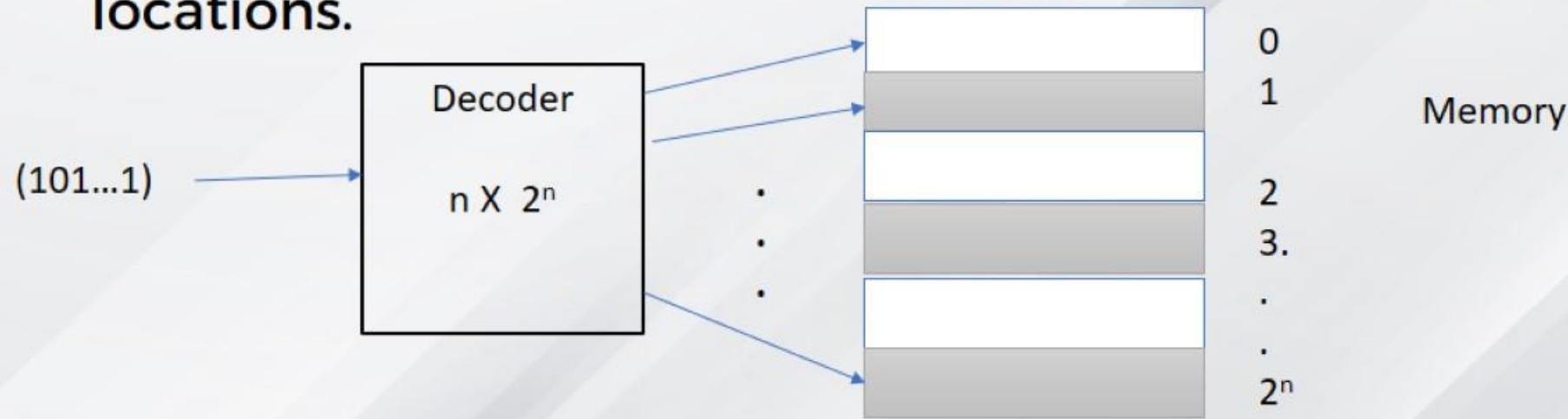
- Known as n -to- m -line decoder, or simply $n:m$ or $n \times m$ ($m = 2^n$).
- It generate 2^n (or fewer) minterms for the n input variables
- When $n = 2$, 4 outputs can be decoded.
- When $n = 3$, 8 outputs can be decoded.



Combinational Circuits Part II

Decoders

- Decoders are important combinational circuits.
- Among other things, they are useful in selecting a memory location according a binary value placed on the address lines of a memory bus.
- Address decoders with n inputs can select any of 2^n locations.



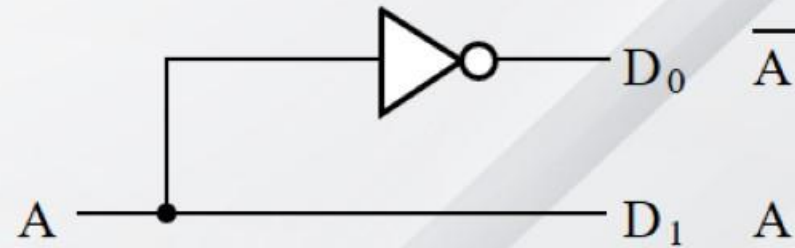
Combinational Circuits Part II

Decoders

➤ 1-to-2-Line Decoder

A	D ₀	D ₁
0	1	0
1	0	1

(a)



(b)

Combinational Circuits Part II

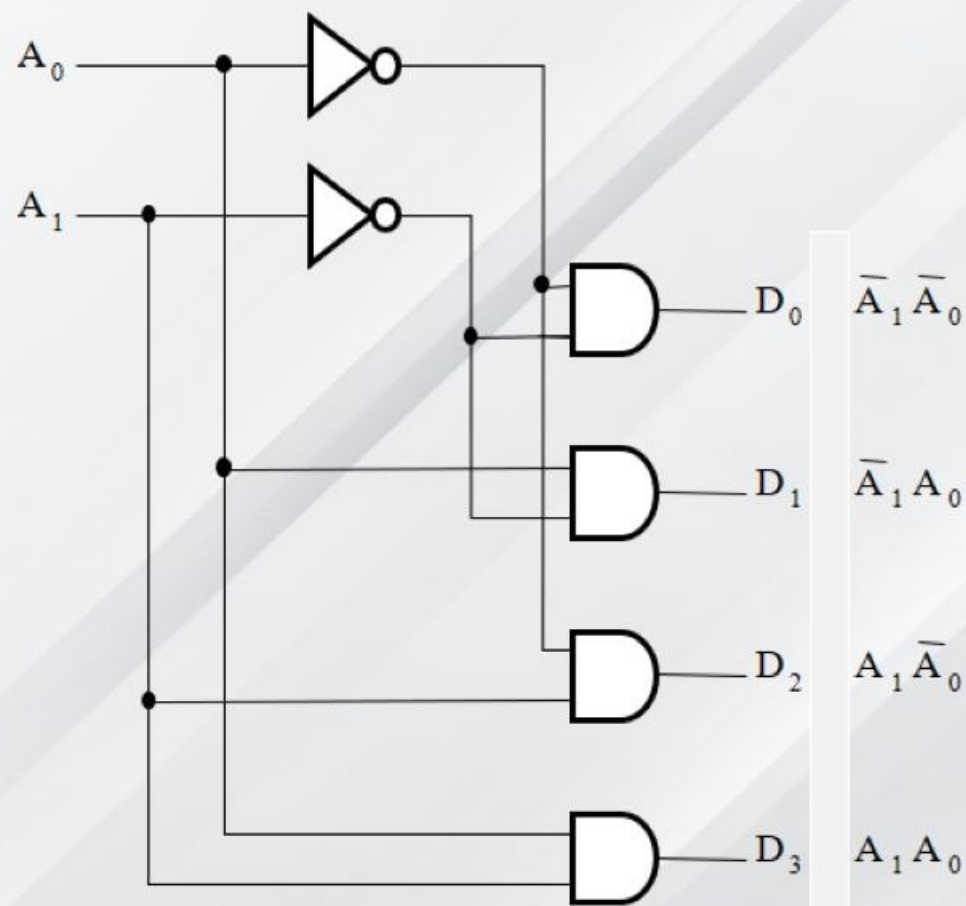
2-to-4-line Decoder

➤ 2-to-4-Line Decoder

- Note that the 2-to-4-line made up of 2 1-to-2-line decoders and 4 AND gates.

A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

(a)



(b)

Combinational Circuits Part II

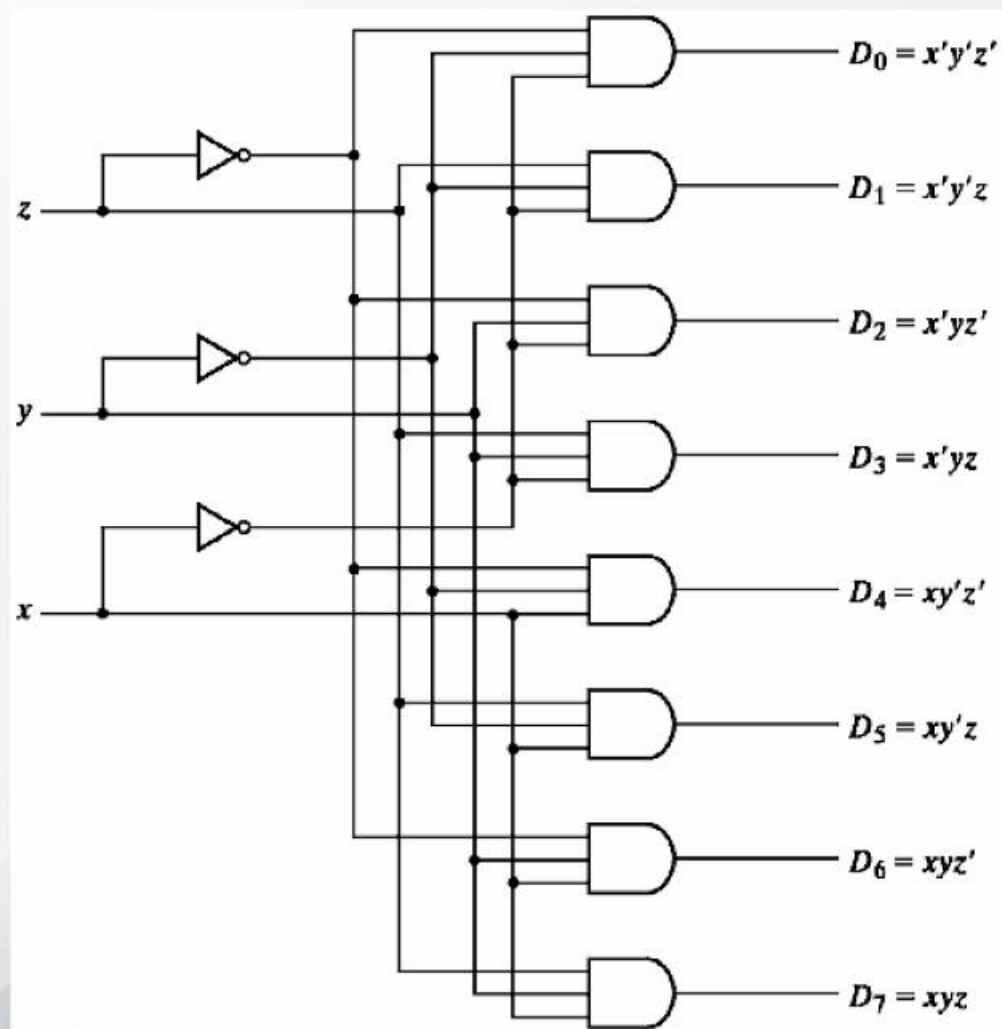
3-to-8-line Decoder

➤ Truth table for a 3X8 decoder

Input			Output							
x	y	z	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Combinational Circuits Part II

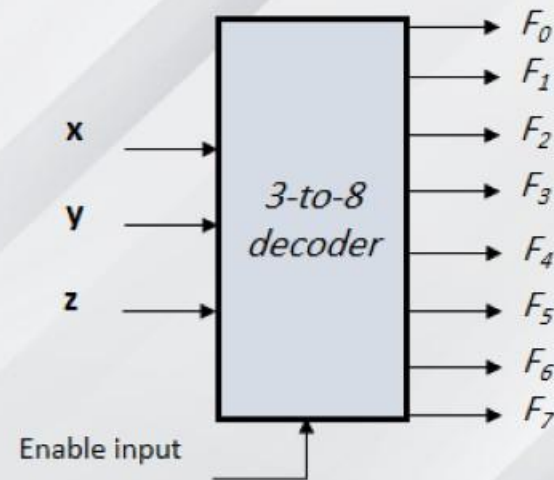
3-to-8-line Decoder



Combinational Circuits Part II

Decoder with Enable input

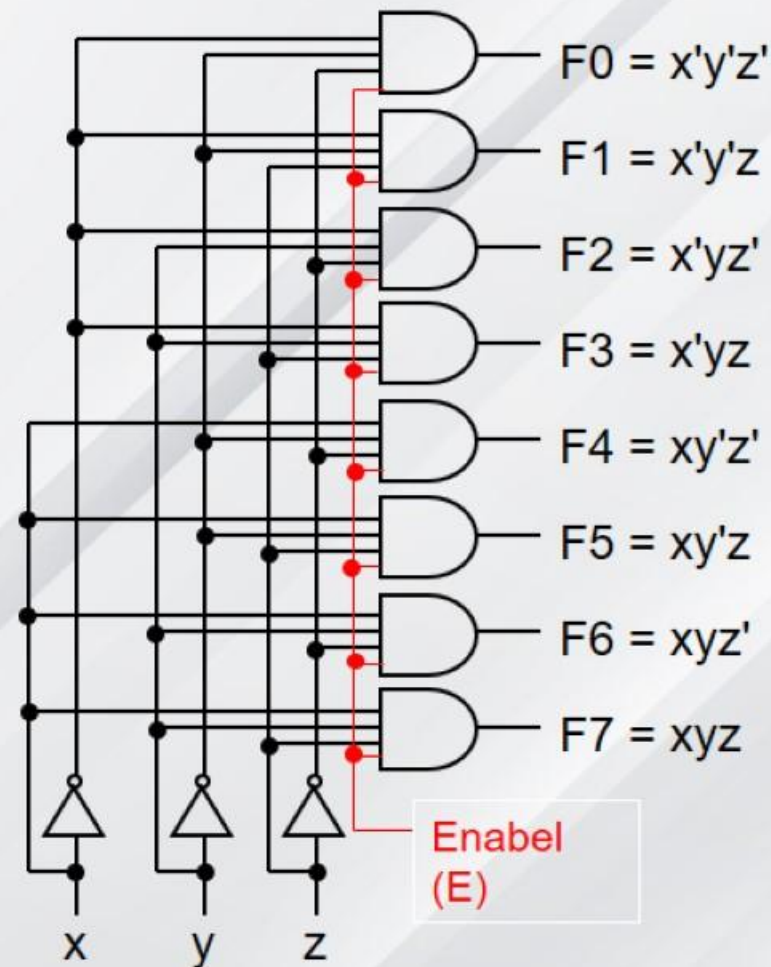
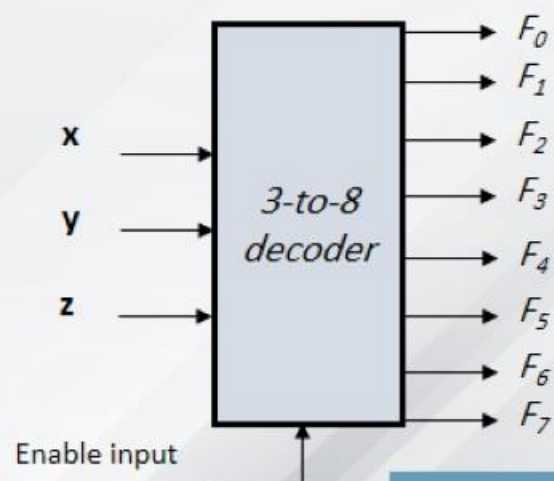
- Decoders may include one or more enable inputs to control the circuit operation.
- The decoder is enabled (or activated) if $E=1$



Combinational Circuits Part II

3x8 decoder with enable (E)

E	x	y	z	F ₀	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇
0	x	x	x	0	0	0	0	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	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1

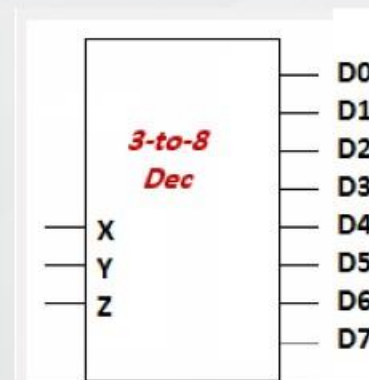


Combinational Circuits Part II

Decoder Expansion

- Another way to design a decoder is to break it into smaller pieces.
- Notice some patterns in the table below:
 - When $X = 0$, outputs D0-D3 are generated as in a 2-to-4 decoder.
 - When $X = 1$, outputs D4-D7 are generated as in a 2-to-4 decoder.

X	Y	Z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

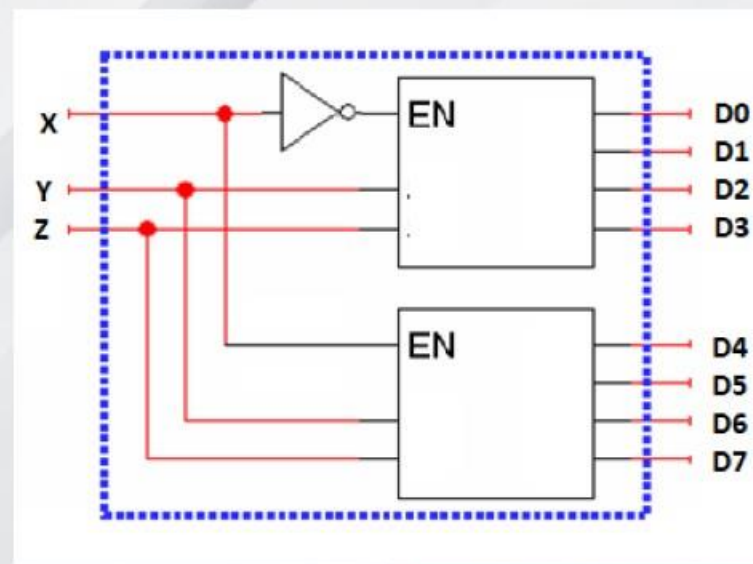


Combinational Circuits Part II

Decoder Expansion

- Here's a 3-to-8 decoder built from two smaller 2-to-4 decoders.
- When $X=0$, the top 2-to-4 decoder is enabled and generates a 1 for one of outputs D0, D1, D2 or D3.
- When $X=1$, the bottom 2-to-4 decoder is enabled instead, and a 1 will be output for either D4, D5, D6 or D7.

X	Y	Z	D0	D1	D2	D3	D4	D5	D6	D7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Combinational Circuits Part II

4-to-16-line Decoder

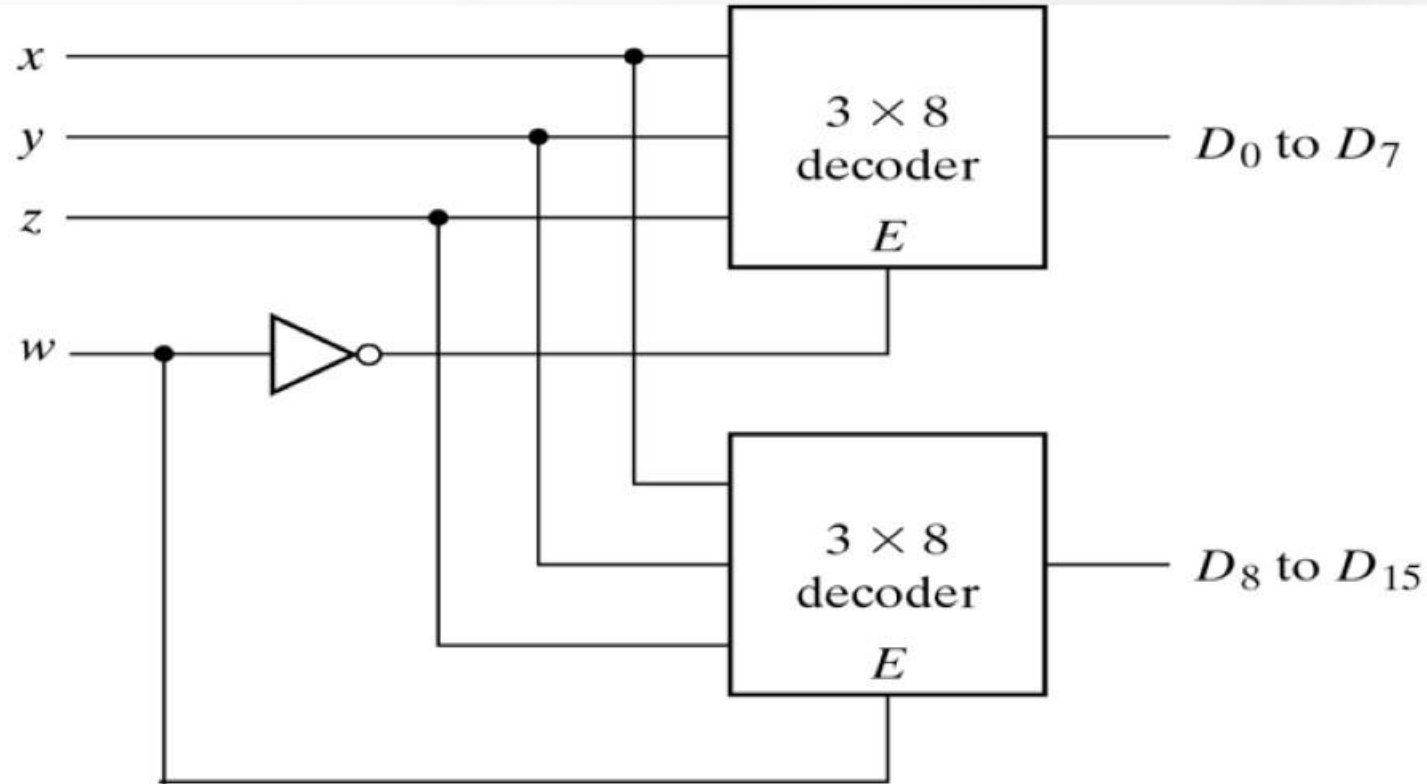


Fig. 4-20 4 $\times$ 16 Decoder Constructed with Two 3 $\times$ 8 Decoders

Combinational Circuits Part II

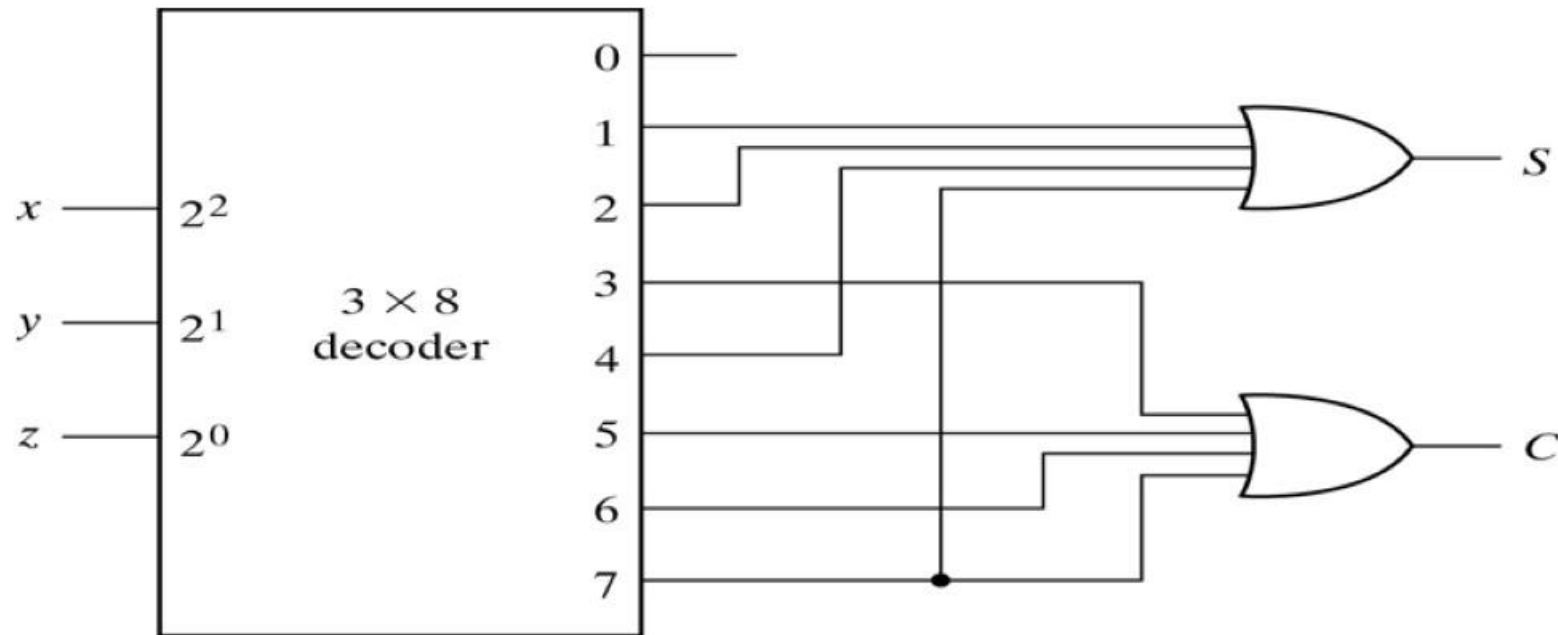
Combinational Logic Implementation

- Since any Boolean function can be expressed in sum-of-minterms form, a decoder that generates the minterms of the function, together with an external OR gate that forms their logical sum, provides a hardware implementation of the function.
- In this way, any combinational circuit with n inputs and m outputs can be implemented with an n -to- 2^n -line decoder and m OR gates.

Combinational Circuits Part II

Combinational Logic Implementation

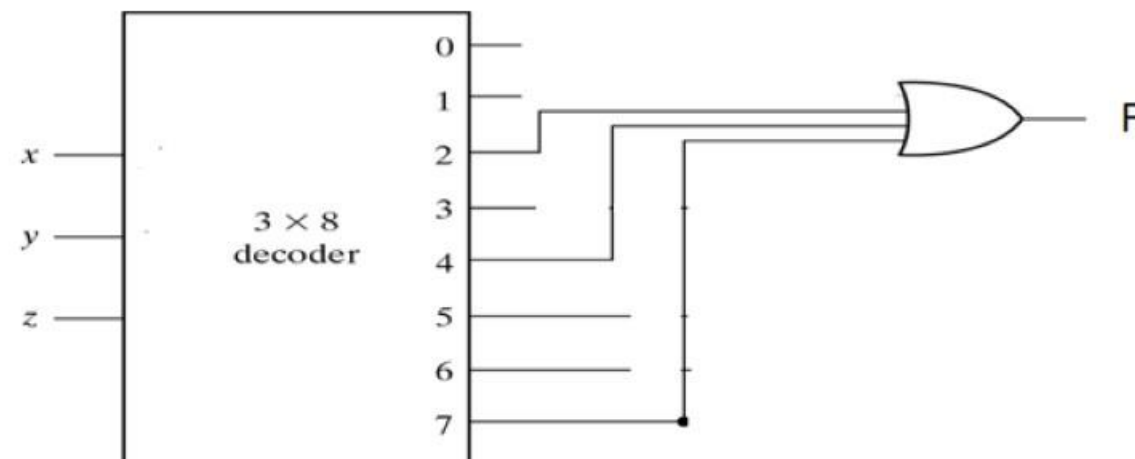
- From truth table of full adder, we have functions
 $S(x, y, z) = \Sigma(1, 2, 4, 7)$ and $C(x, y, z) = \Sigma(3, 5, 6, 7)$



Combinational Circuits Part II

- **Exercise:**
- Implement the following function using 3 to 8 decoders and or gates
- $F(X,Y,Z) = \sum (2,4,7)$

$$F(X,Y,Z) = X'.Y.Z' + X.Y'.Z' + X.Y.Z$$



Combinational Circuits Part II

• Exercise:

- What is the functions implemented by the decoders in the following fig:

Solution:

- $F1 = x \cdot y' \cdot z + x \cdot z'$

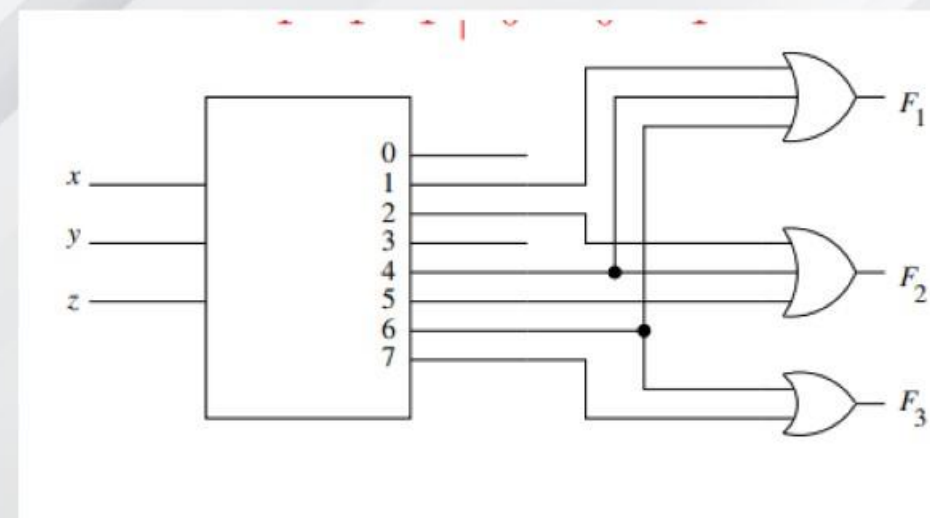
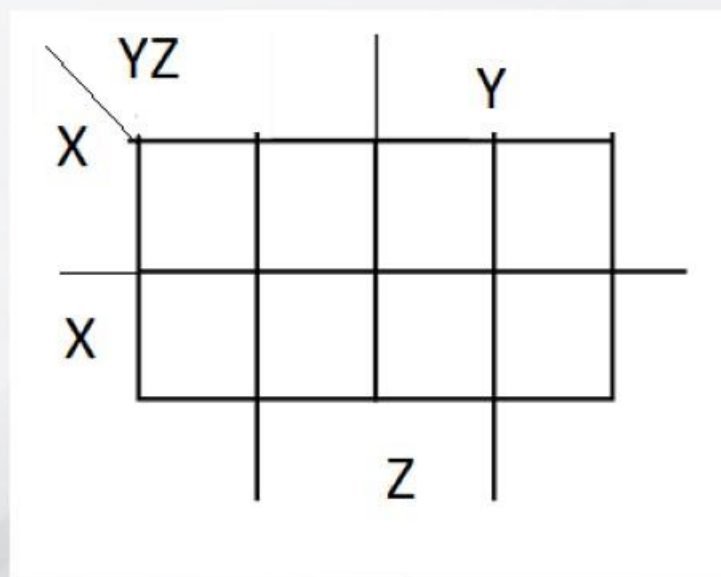
$$F1 = \sum (1, 4, 6)$$

- $F2 = x \cdot y \cdot z' + x \cdot y'$

$$F2 = \sum (2, 4, 5)$$

- $F3 = x \cdot y$

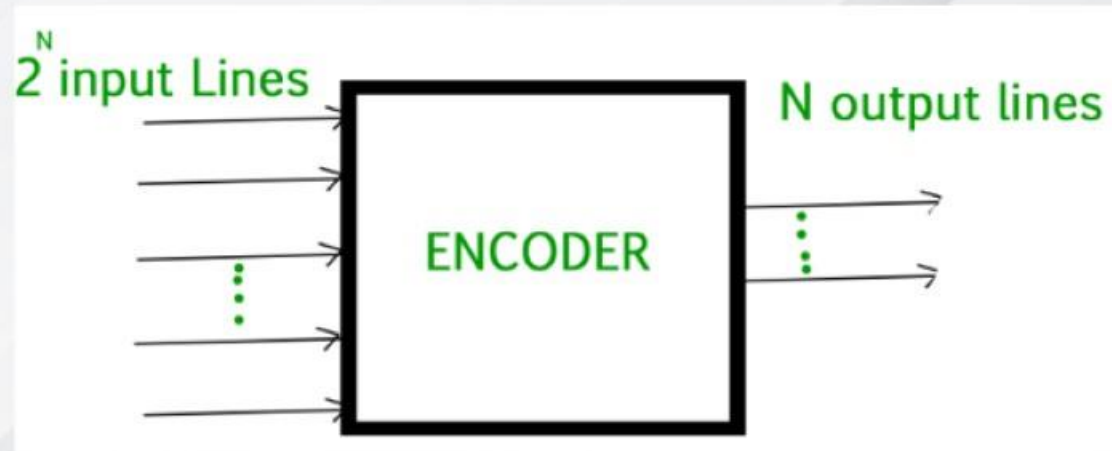
- $= \sum (6, 7)$



Combinational Circuits Part II

Encoding

- **Encoding** is the opposite of decoding –
- It has a maximum of 2^n input lines and n output lines;
- Circuits that perform encoding are called **encoders**



Combinational Circuits Part II

Encoding

- An encoder has 2^n (or fewer) input lines and n output lines which generate the binary code corresponding to the input values
- Typically, an encoder converts a code containing exactly one bit that is 1 to a binary code corresponding to the position in which the 1 appears.

Combinational Circuits Part II

Encoders

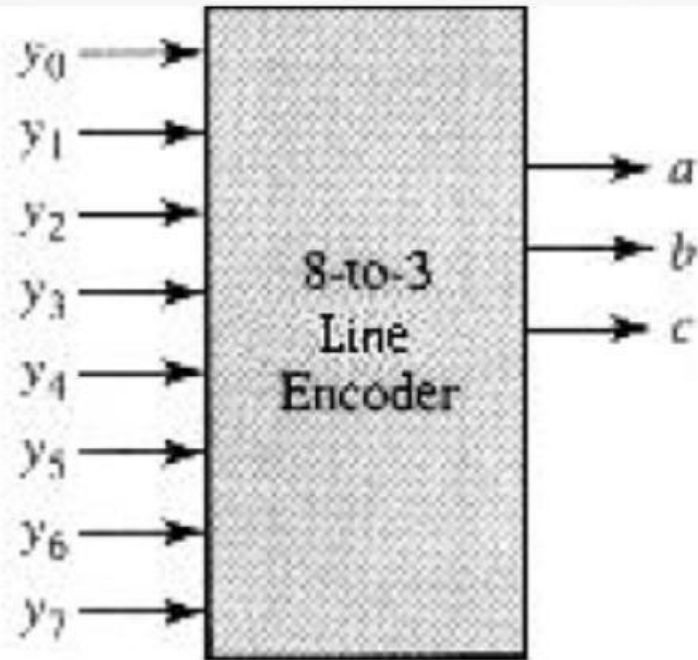
➤ Truth table for a 8X3 encoder

Inputs								Outputs		
Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	A	B	C
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

Combinational Circuits Part II

Encoders

➤ Block diagram for a 8X3 encoder



- An encoder is the inverse operation of a decoder.
- We can derive the Boolean functions from the previous table as follows:
- $C = y_1 + y_3 + y_5 + y_7$
- $B = y_2 + y_3 + y_6 + y_7$
- $A = y_4 + y_5 + y_6 + y_7$

Combinational Circuits Part II

Priority encoder

- If two inputs are active simultaneously, the output produces an undefined combination.
- We can establish an input priority to ensure that only one input is encoded.
- Another ambiguity in the octal-to-binary encoder is that an output with all 0's is generated when all the inputs are 0 and the output will be the same as when D0 is equal to 1.

Combinational Circuits Part II

Priority encoder

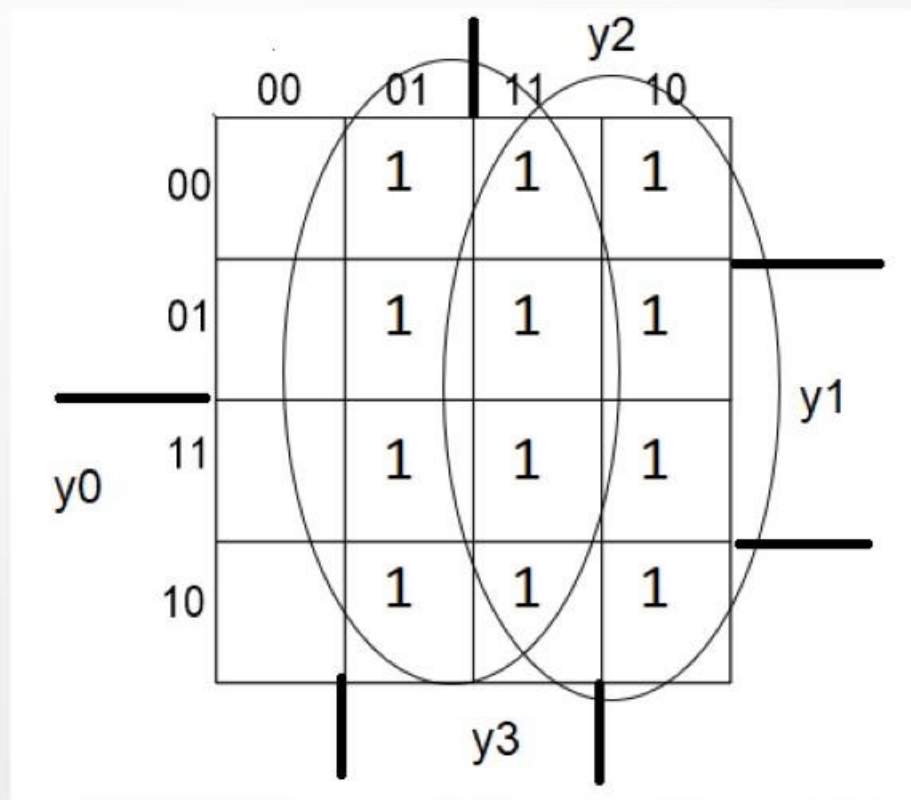
- This can be resolved by providing one more output “V” to indicate that at least one input is equal to 1.
- $V=0$ if no valid inputs
- $V=1$ if valid inputs

Combinational Circuits Part II

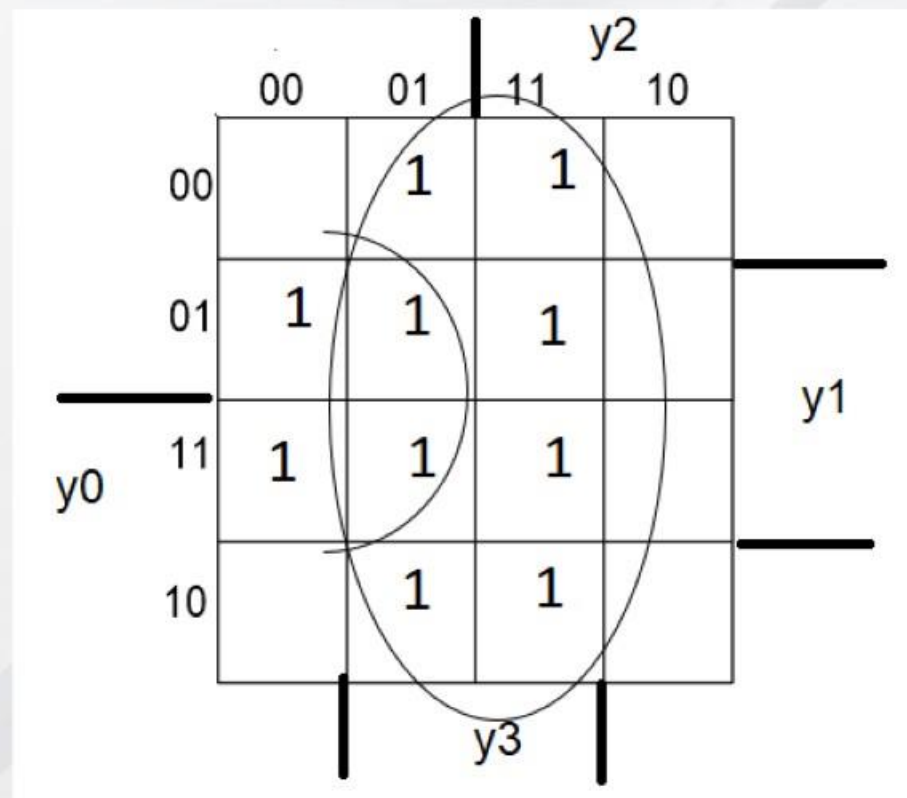
y0	y1	y2	y3	A	B	V
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

y0	y1	y2	y3	A	B	V
0	0	0	0	X	X	0
0	0	0	1	1	1	1
0	0	1	0	1	0	1
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	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	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

Combinational Circuits Part II



$$A = y_2 + y_3$$



$$B = y_1 y_2' + y_3$$

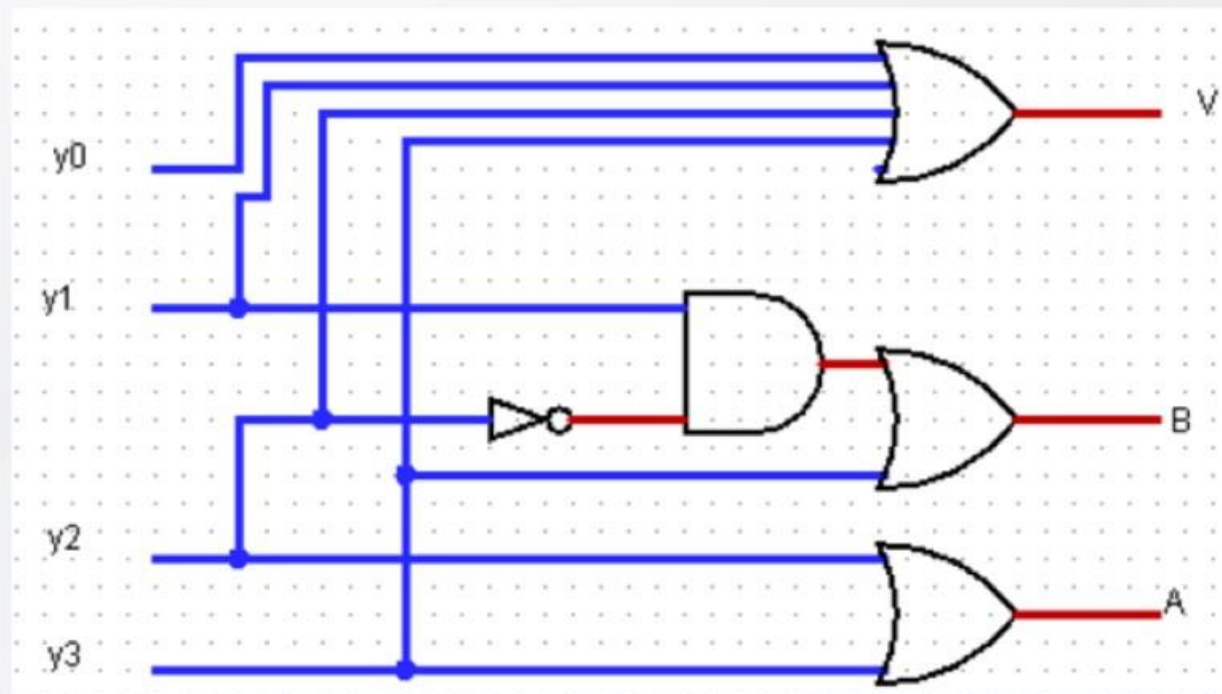
$$V = y_0 + y_1 + y_2 + y_3$$

Combinational Circuits Part II

$$A = y_2 + y_3$$

$$B = y_1 y_2' + y_3$$

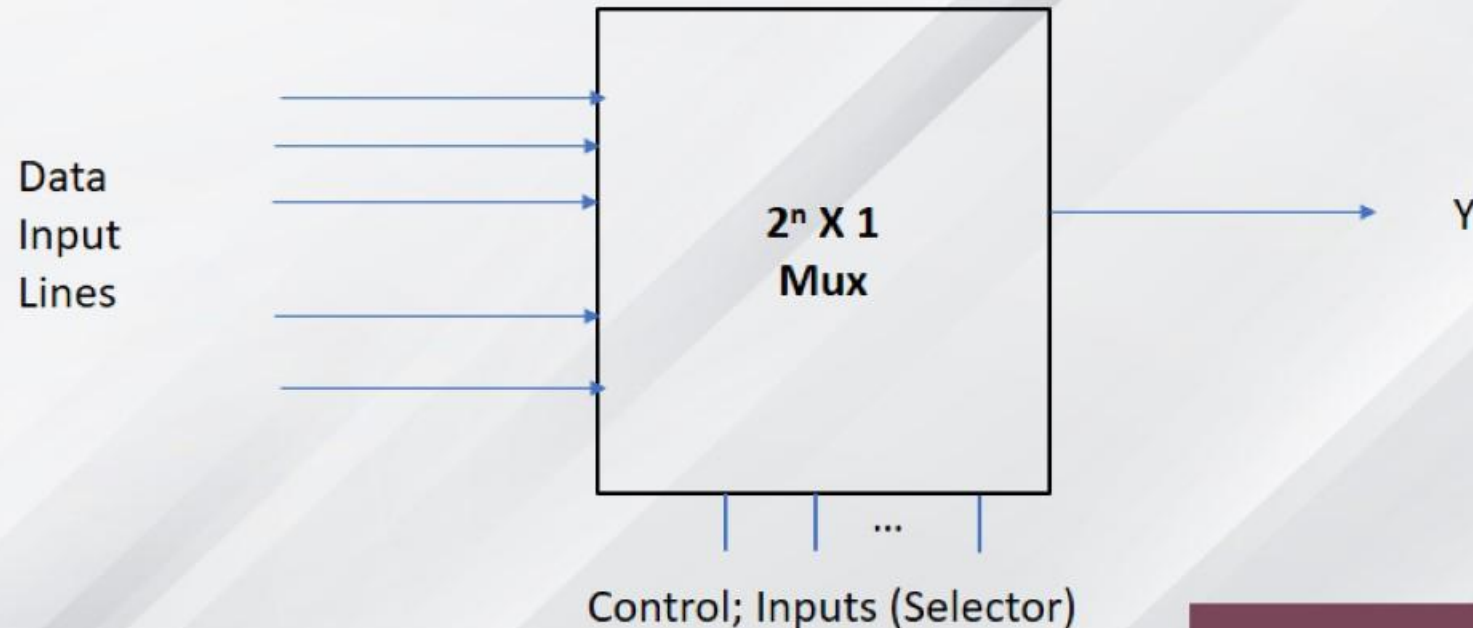
$$V = y_0 + y_1 + y_2 + y_3$$



Combinational Circuits Part II

Multiplexer

- A **multiplexer** or data selector (abbreviated **MUX**) consists of a group of data inputs and a group of control inputs. The control inputs are used to select exactly one data input to be outputted.



Combinational Circuits Part II

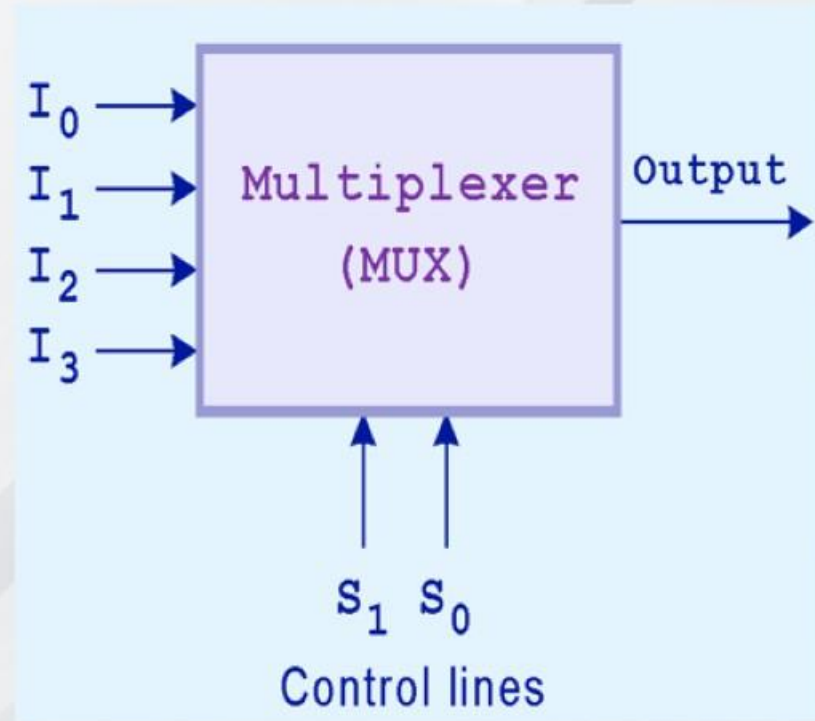
Multiplexer

- A MUX with n control inputs can select from a maximum of 2^n data inputs.
- When $n = 2$, 4 data inputs can be selected.
- When $n = 3$, 8 data inputs can be selected.
- An 8X1 ($n = 3$) MUX uses three control inputs to select exactly one of eight data inputs to be outputted.

Combinational Circuits Part II

Block Diagram for Multiplexer

- It selects a single output from several inputs.
- The particular input chosen for output is determined by the value of the multiplexer's control lines.
- To be able to select among n inputs, $\log_2 n$ control lines are needed.



Combinational Circuits Part II

2-to-1-line Multiplexer

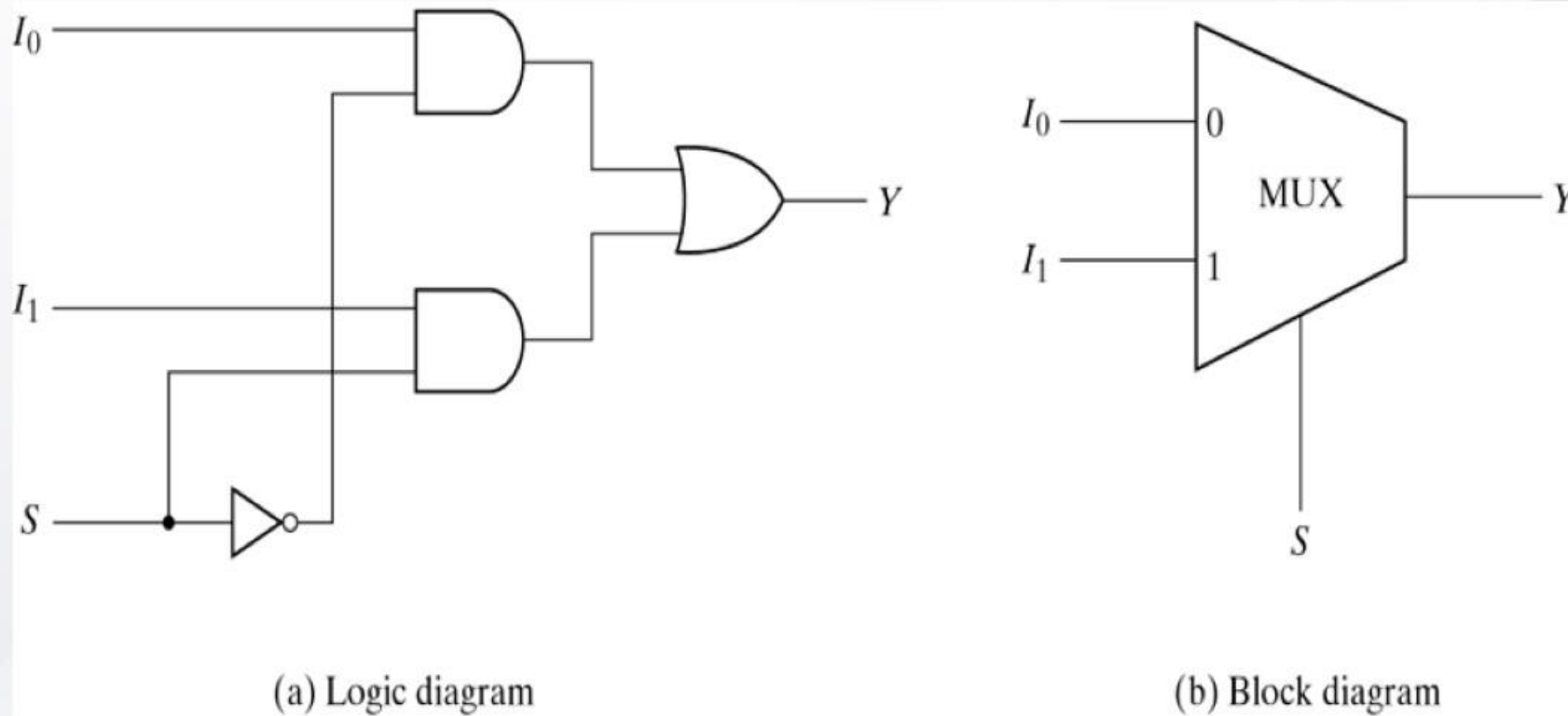
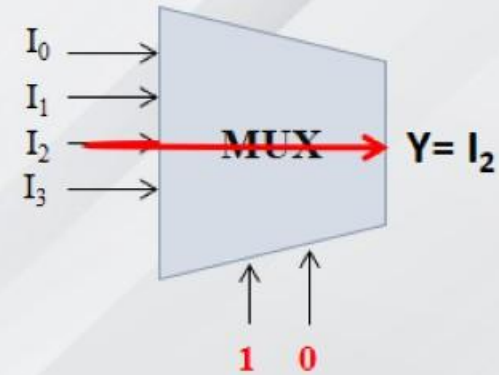
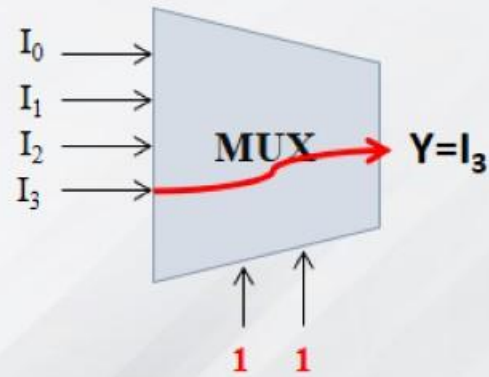
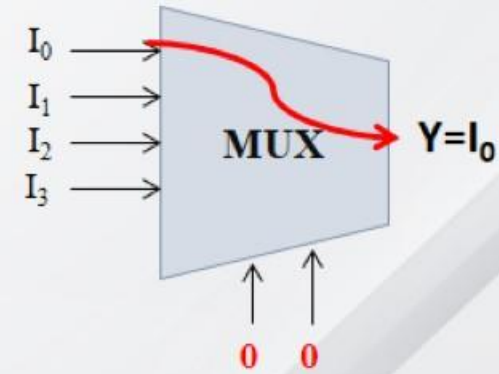
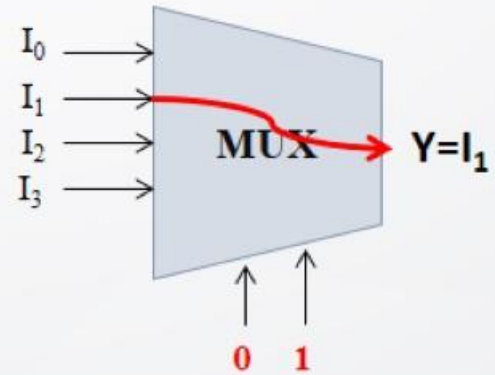


Fig. 4-24 2-to-1-Line Multiplexer

Combinational Circuits Part II

Multiplexer

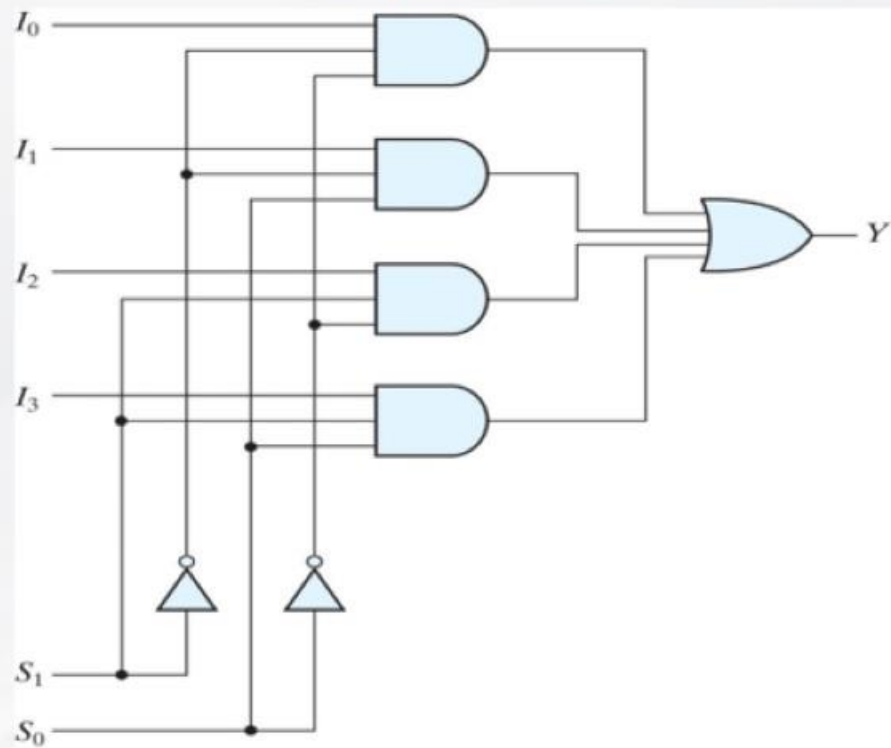


Combinational Circuits Part II

4-to-1-line Multiplexer

Design 4-to1 MUX:

- There are four data inputs $\rightarrow$ two selection inputs S_1, S_0 .
- The input selected to be passed to the output depends on the minterm of the input.



	S_1	S_0	Y
m_0	0	0	I_0
m_1	0	1	I_1
m_2	1	0	I_2
m_3	1	1	I_3

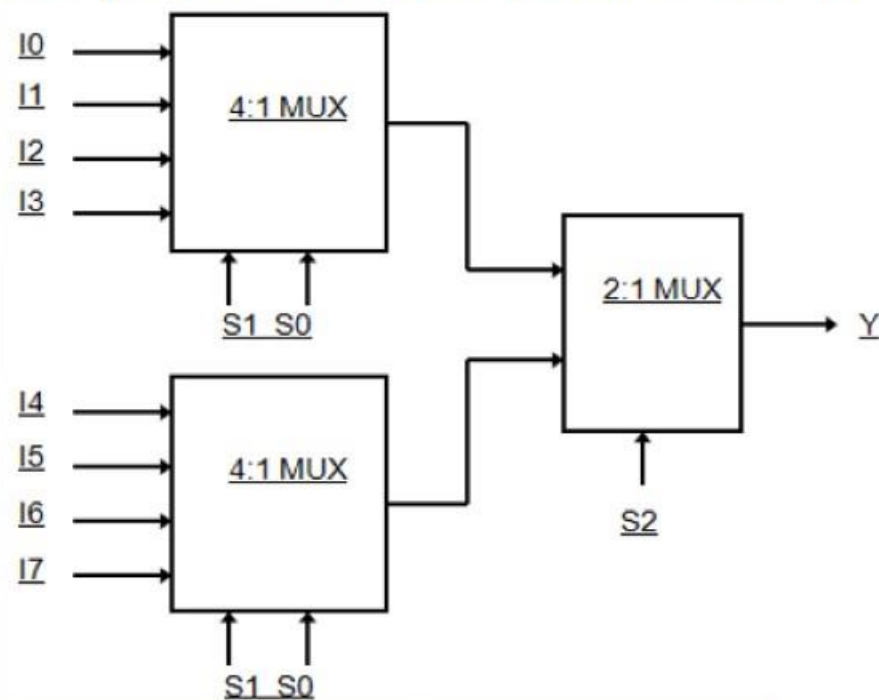
$$Y = \boxed{S_1' S_0'} I_0 + \boxed{S_1' S_0} I_1 + \boxed{S_1 S_0'} I_2 + \boxed{S_1 S_0} I_3$$

$\downarrow$ $\downarrow$ $\downarrow$ $\downarrow$
 m_0 m_1 m_2 m_3

Combinational Circuits Part II

Multiplexer Expansion

- Larger multiplexer can be constructed from smaller ones.
- An 8-to-1 multiplexer can be constructed from smaller multiplexers like this (note placement of selector lines):



S_2	S_1	S_0	Y
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7

4-to-1 has 4 data input, 2 selection input, and 1 output.
2-to-1 has 2 data input, 1 selection input, and 1 output.

Combinational Circuits Part II

CC Implementation with MUX

- Given a function of n -variables MUXex can be used to implement this function.
- This can be accomplished in one of 2 ways:
 - **Using a Mux with n -select inputs**
 - n variables need to be connected to n select inputs.
 - Minterms of a function are generated according to select inputs.
 - Individual minterm can be selected by the data inputs $\rightarrow$ proper assignment of the data inputs ($D_i \in \{0, 1\}$).

Combinational Circuits Part II

CC Implementation with MUX

- **Using a Mux with $n-1$ select inputs**

- Find truth table.
- The first $n-1$ variables in table are connected to selection inputs of MUX.
- For each combination of selection variables, evaluate output as function of the remaining variable (d)
- This remaining variable (d) is then used for data inputs which can be 0,1, d , d' .

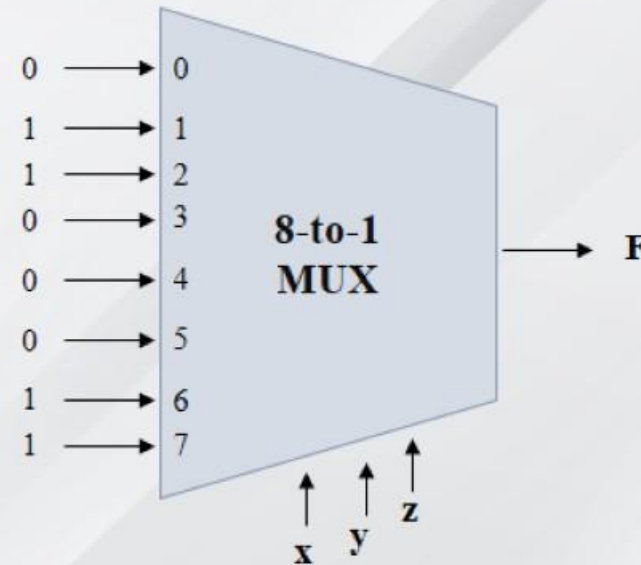
Combinational Circuits Part II

Example 1

Implement the function $F(x,y,z) = \Sigma(1,2,6,7)$ using a multiplexer with (n-select inputs)

- The function has 3 variables $\rightarrow$ using 3-select inputs, we need a 8-to-1 MUX.

<i>x</i>	<i>y</i>	<i>z</i>	<i>F</i>
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



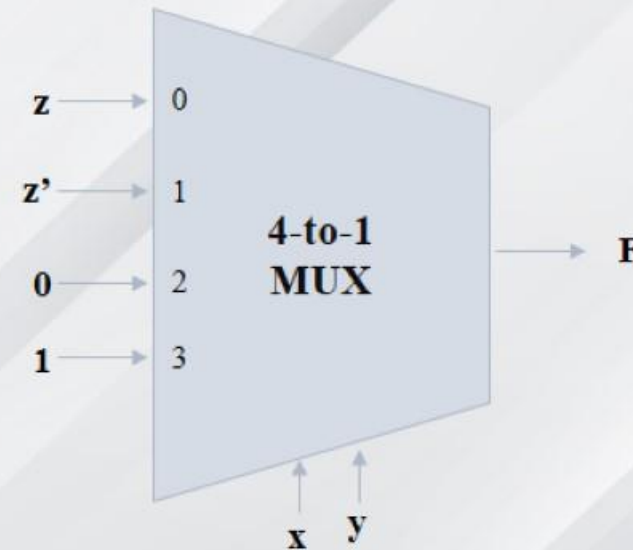
Combinational Circuits Part II

Example 2

Implement the function $F(x,y,z) = \Sigma(1,2,6,7)$ using a multiplexer with (n-1 select inputs).

- The function has 3 variables $\rightarrow$ using 2-select inputs, (2^{n-1} data inputs) we need a 4-to-1 MUX.

<i>x</i>	<i>y</i>	<i>z</i>	<i>F</i>	
0	0	0	0	$F = z$
0	0	1	1	
0	1	0	1	$F = z'$
0	1	1	0	
1	0	0	0	$F = 0$
1	0	1	0	
1	1	0	1	$F = 1$
1	1	1	1	



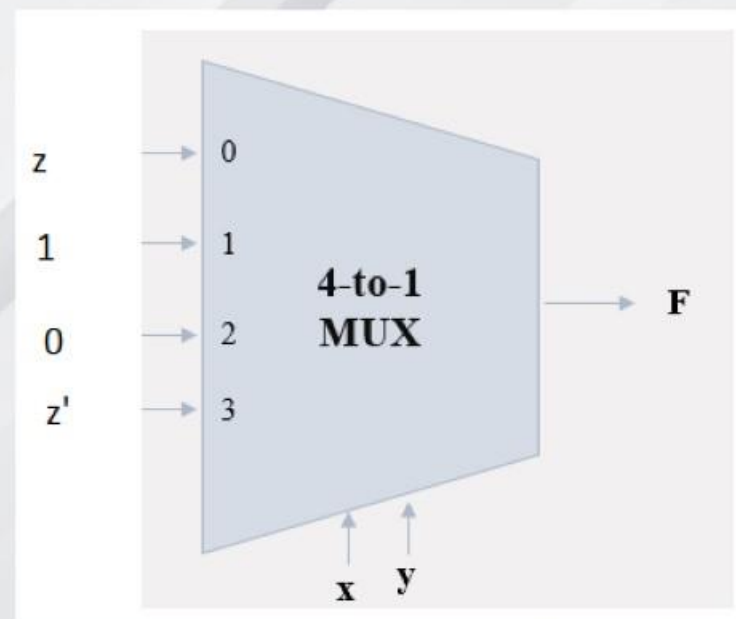
Combinational Circuits Part II

- **Exercise:**
- Deduce the function implemented by the following circuit?

Solution

$$F(x,y,z) = \sum (1,2,3,6)$$

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0





Thank You