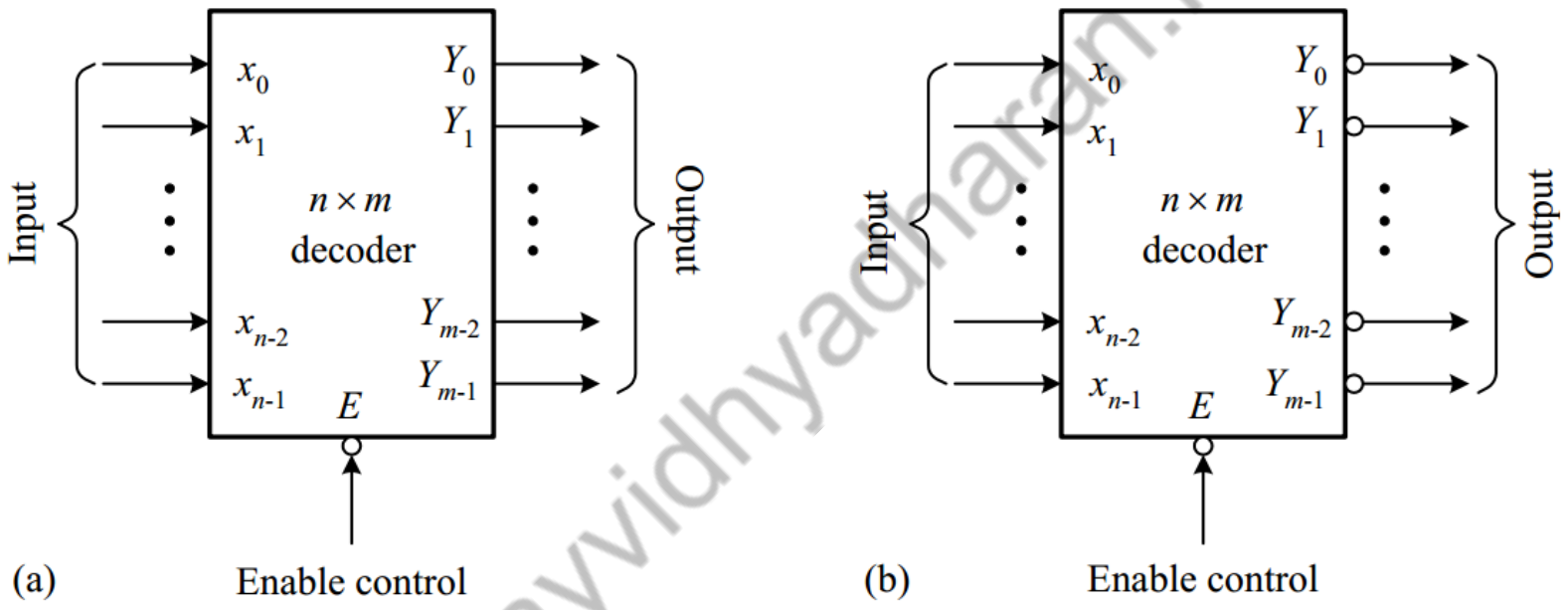
Advanced VLSI Design
Lecture 14
Combinational Components

By Prof. Sanjay Vidhyadharan

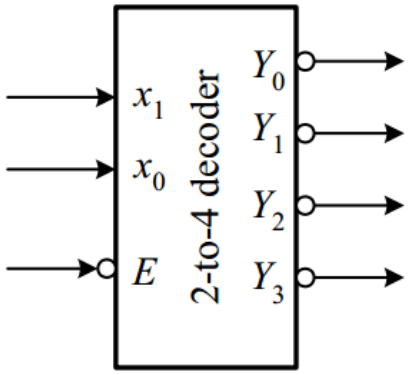
Decoders



(a) active-high output;

(b) active-low output.

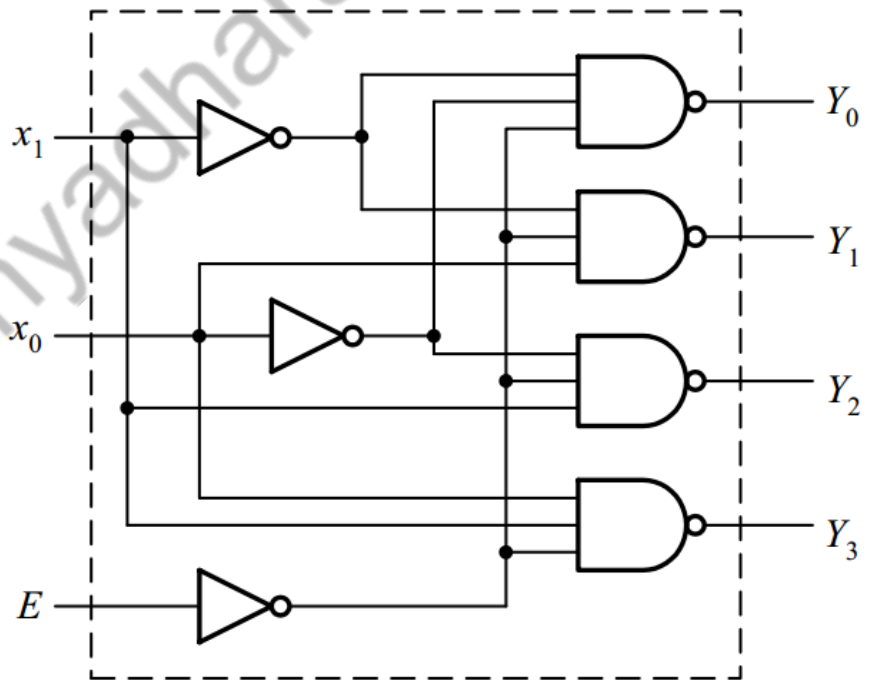
Decoders



(a)

E	x_1	x_0	Y_3	Y_2	Y_1	Y_0
1	ϕ	ϕ	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

(b)



(c)

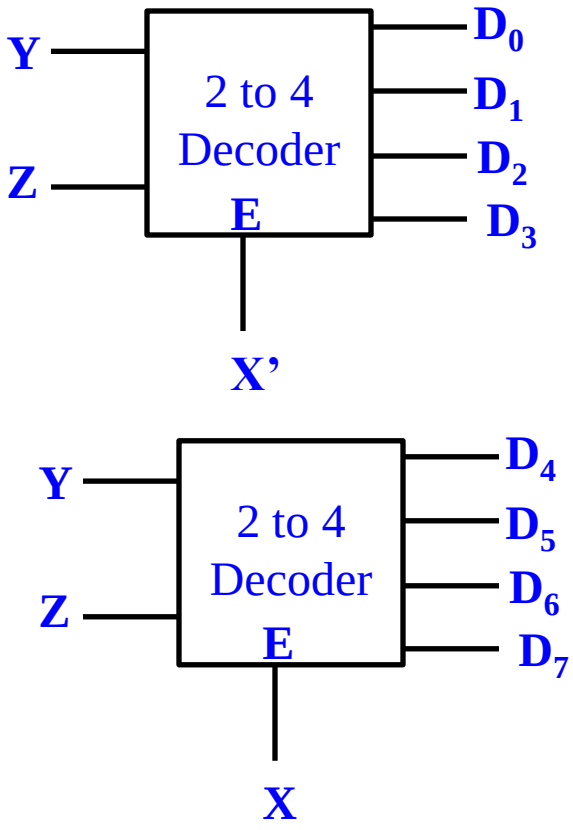
For most practical decoders, one or more enable controls may also be embedded into decoders to allow expanding capability

Decoders

Implementing a 3 to 8 decoder using two 2 to 4 decoders with enable pin

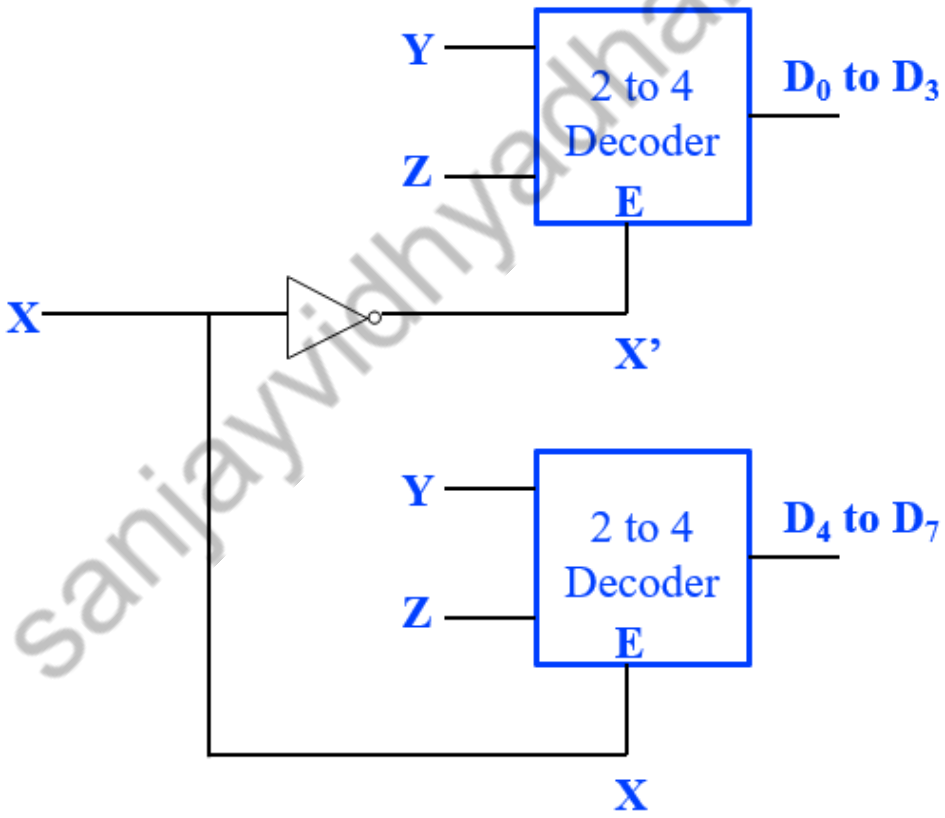
X	Y	Z
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1



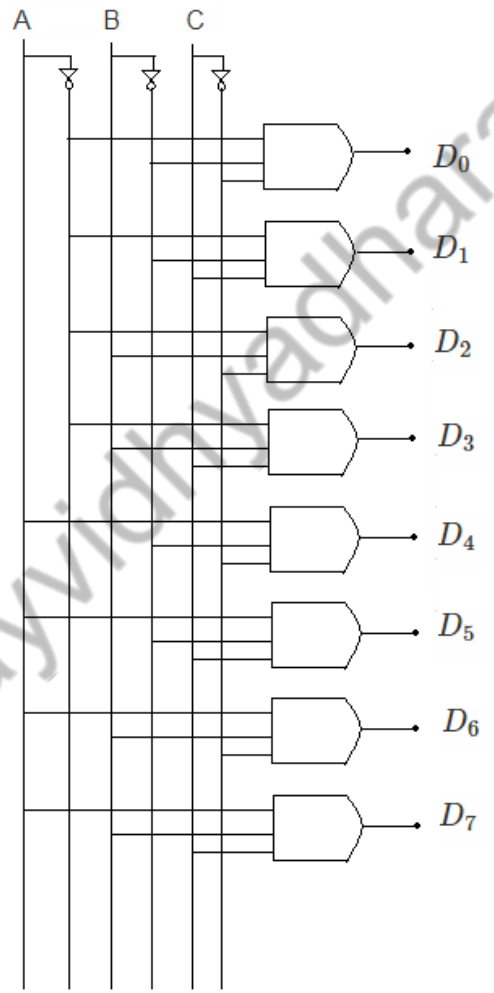
Decoders

Implementing a 3 to 8 decoder using two 2 to 4 decoders with enable pin



Decoders

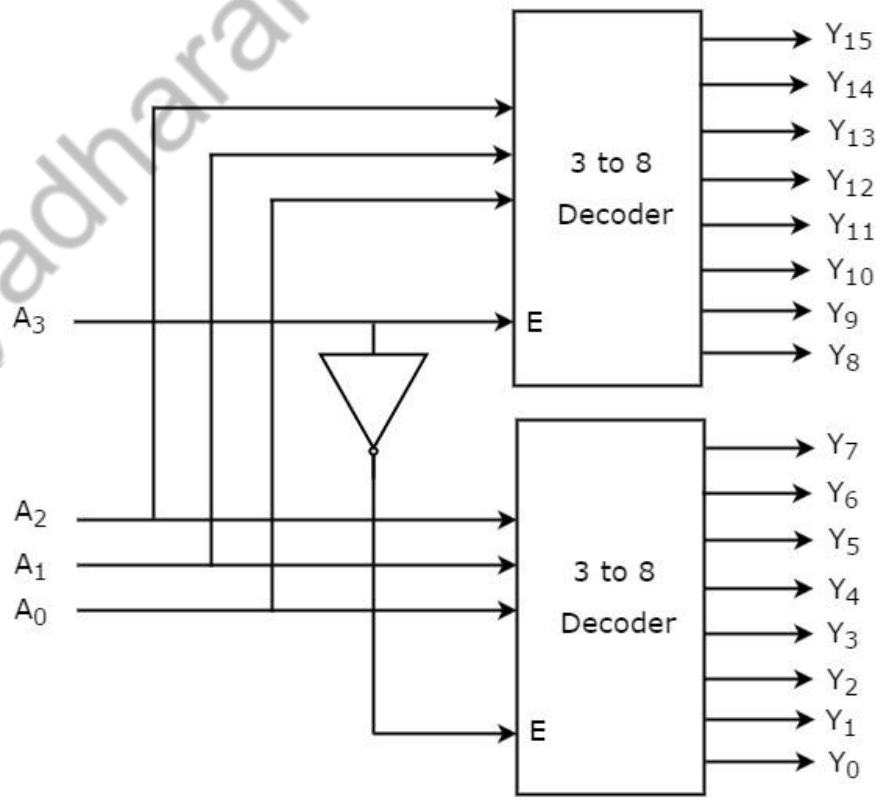
3 to 8 decoder



Decoders

4 to 16 decoder

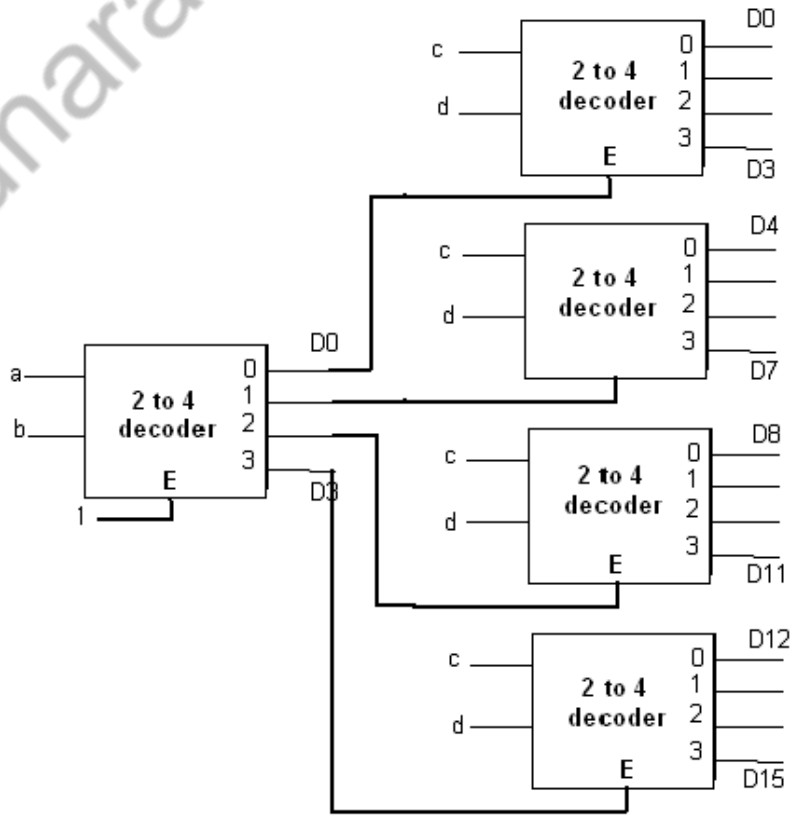
INPUTS				OUTPUTS															
A_3	A_2	A_1	A_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	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	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Decoders

4 to 16 decoder

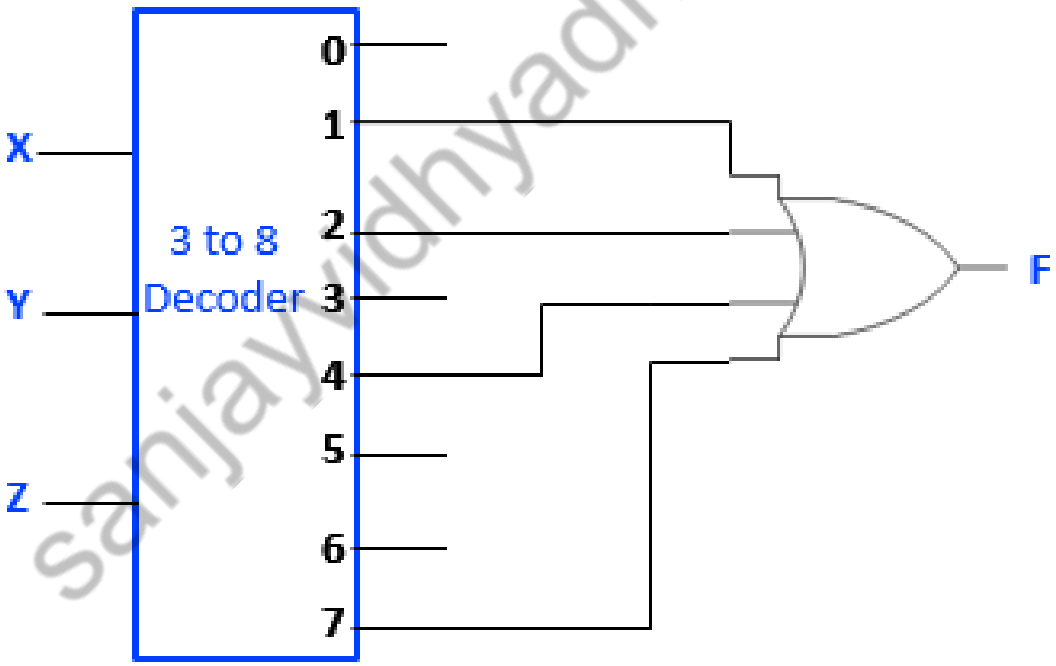
INPUTS				OUTPUTS														
A_3	A_2	A_1	A_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	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	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



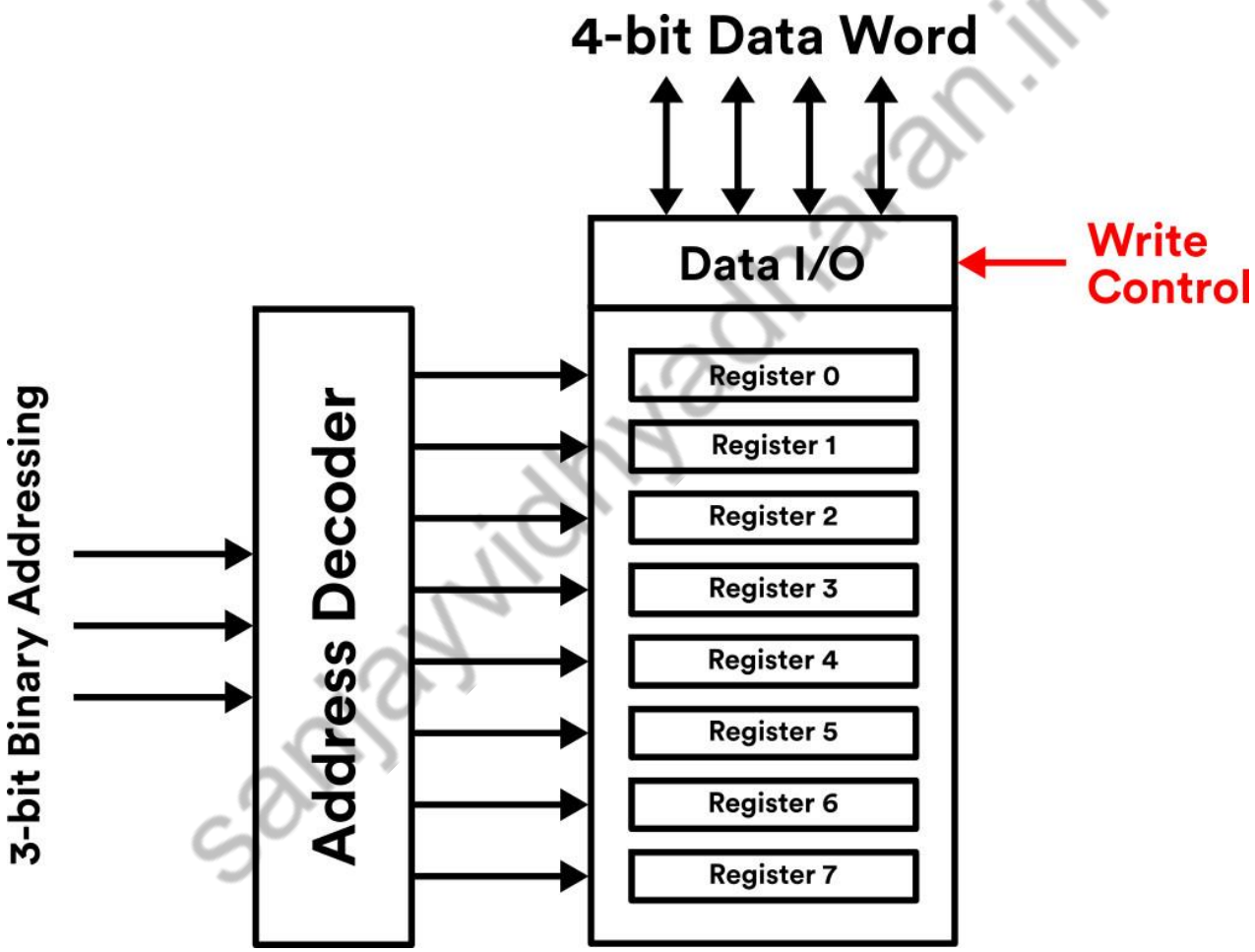
Decoders

Implementing function using Decoders

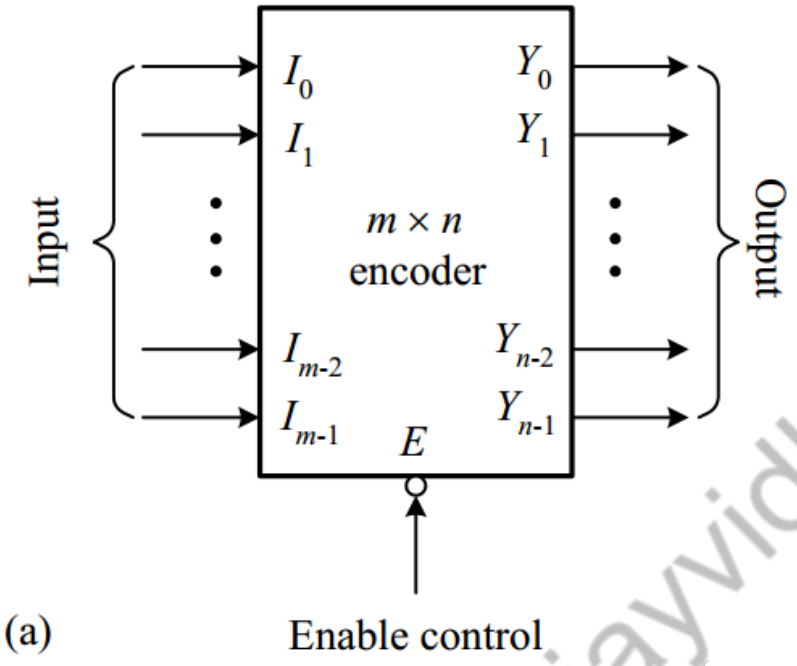
$F = \sum (1, 2, 4, 7) = X'Y'Z + X'YZ' + XY'Z' + XYZ$



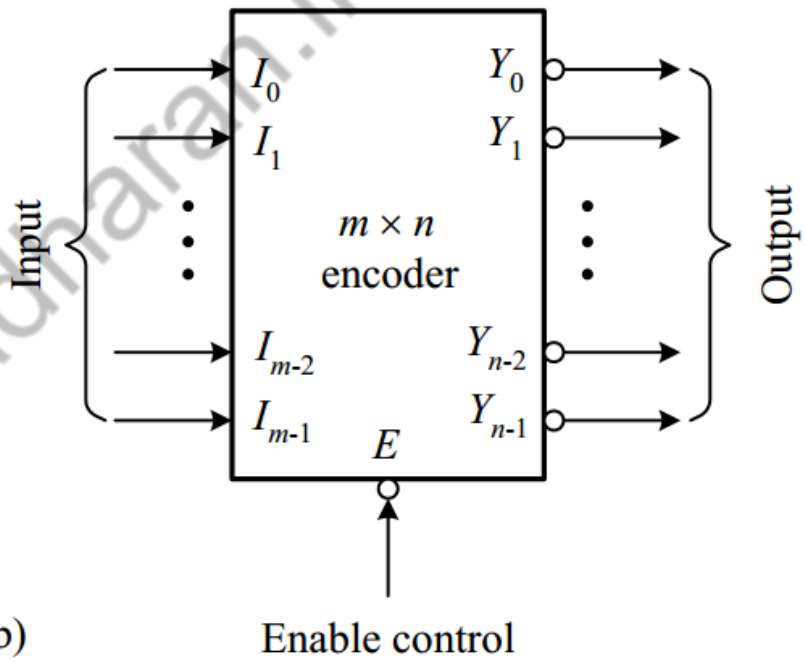
Decoder Applications



Encoders

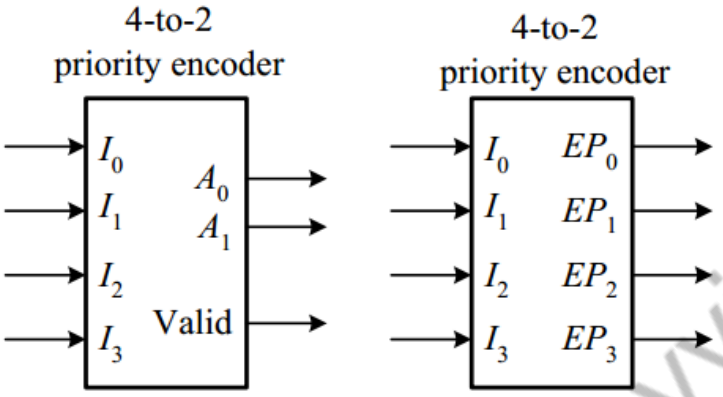


(a) active-high output;



(b) active-low output.

Priority Encoder

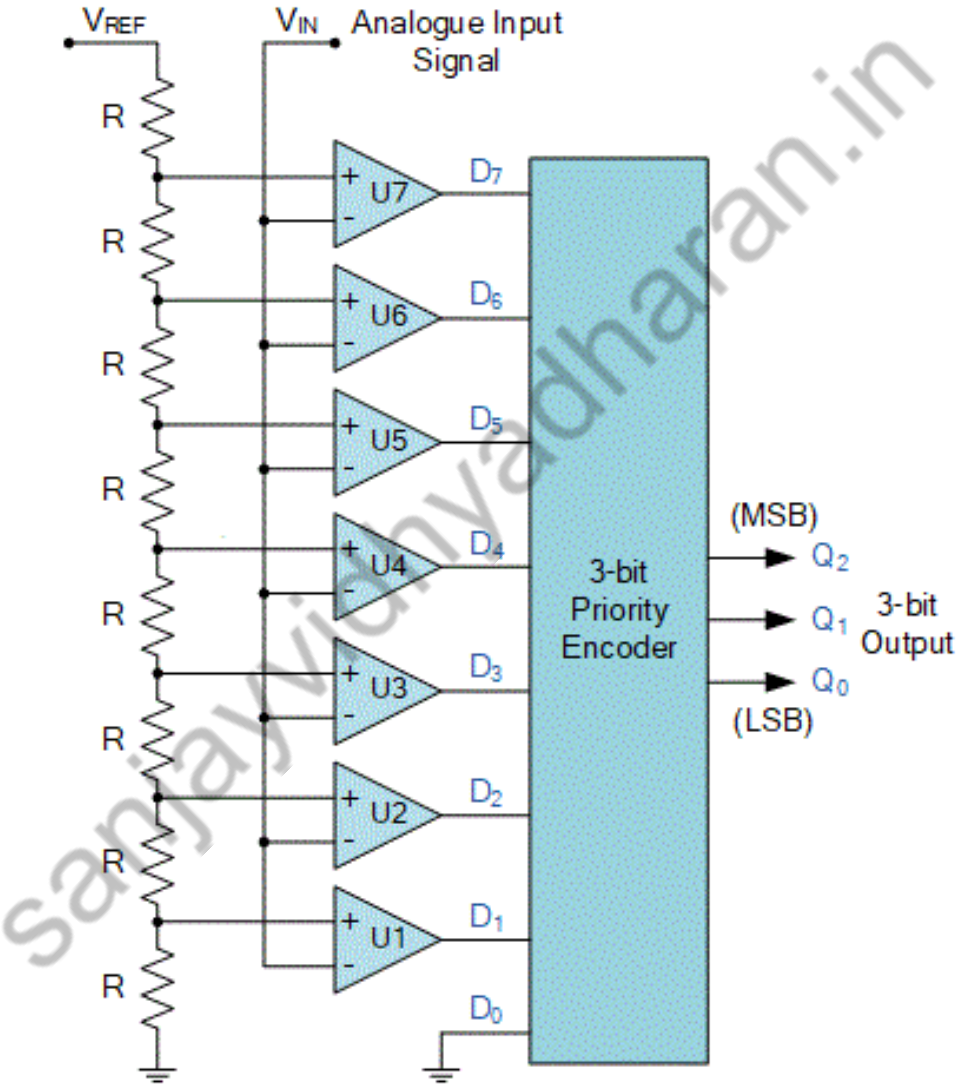


(a)

Input				Binary output			1-out-of-4 output			
I_3	I_2	I_1	I_0	A_1	A_0	*Valid*	EP_3	EP_2	EP_1	EP_0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0	0	0	1
0	0	1	ϕ	0	1	1	0	0	1	0
0	1	ϕ	ϕ	1	0	1	0	1	0	0
1	ϕ	ϕ	ϕ	1	1	1	1	0	0	0

(b)

Priority Encoder



Priority Encoder

D_0	D_1	D_2	D_3
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

X	Y
0	0
0	1
1	0
1	1

Encoder

$X = D_2 + D_3$

$Y = D_1 + D_3$

D_0	D_1	D_2	D_3
1	0	0	0
X	1	0	0
X	X	1	0
X	X	X	1
0	0	0	0

X	Y	V
0	0	1
0	1	1
1	0	1
1	1	1
X	X	0

Priority Encoder

$X = D_2 + D_3$

$Y = D_1D_2' + D_3$

$V = D_0 + D_1 + D_2 + D_3$

Priority Encoder

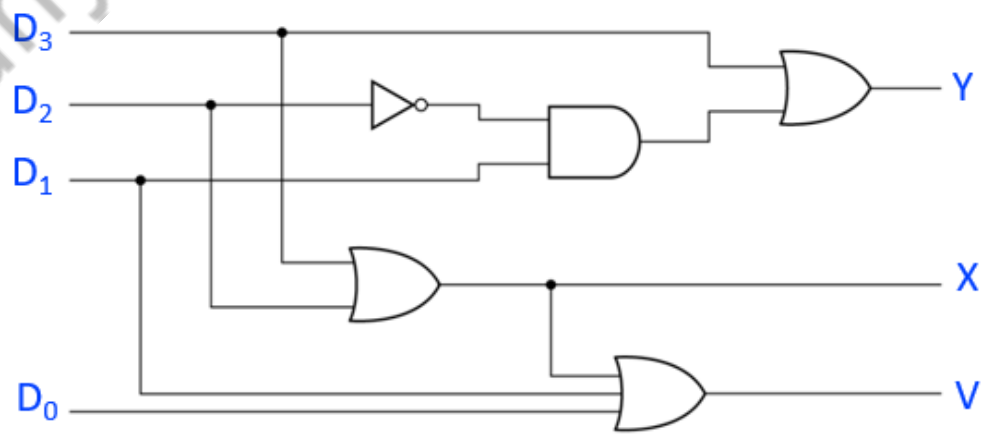
Priority Encoder

$X = D_2 + D_3$

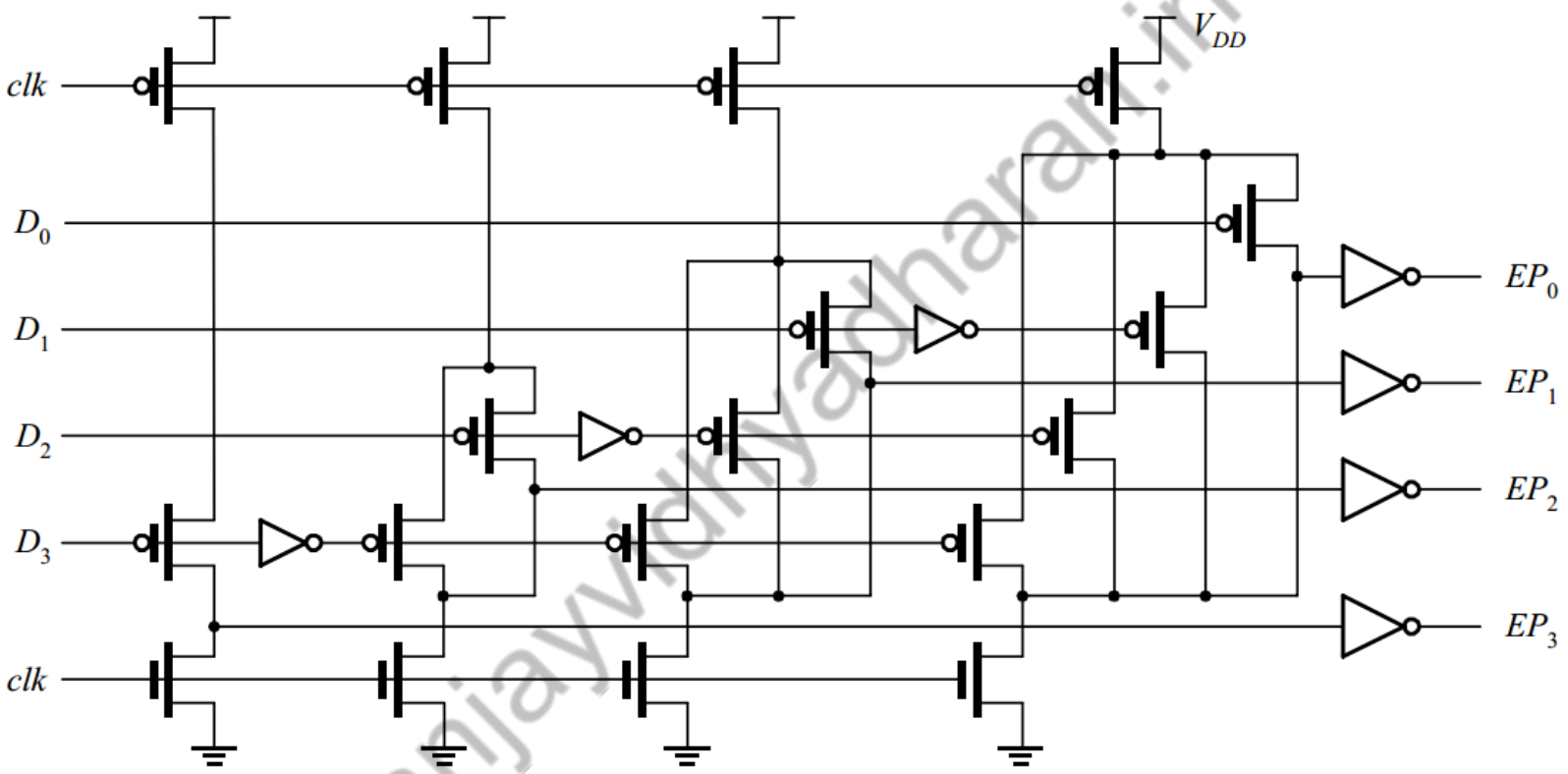
$Y = D_1D_2' + D_3$

$V = D_0 + D_1 + D_2 + D_3$

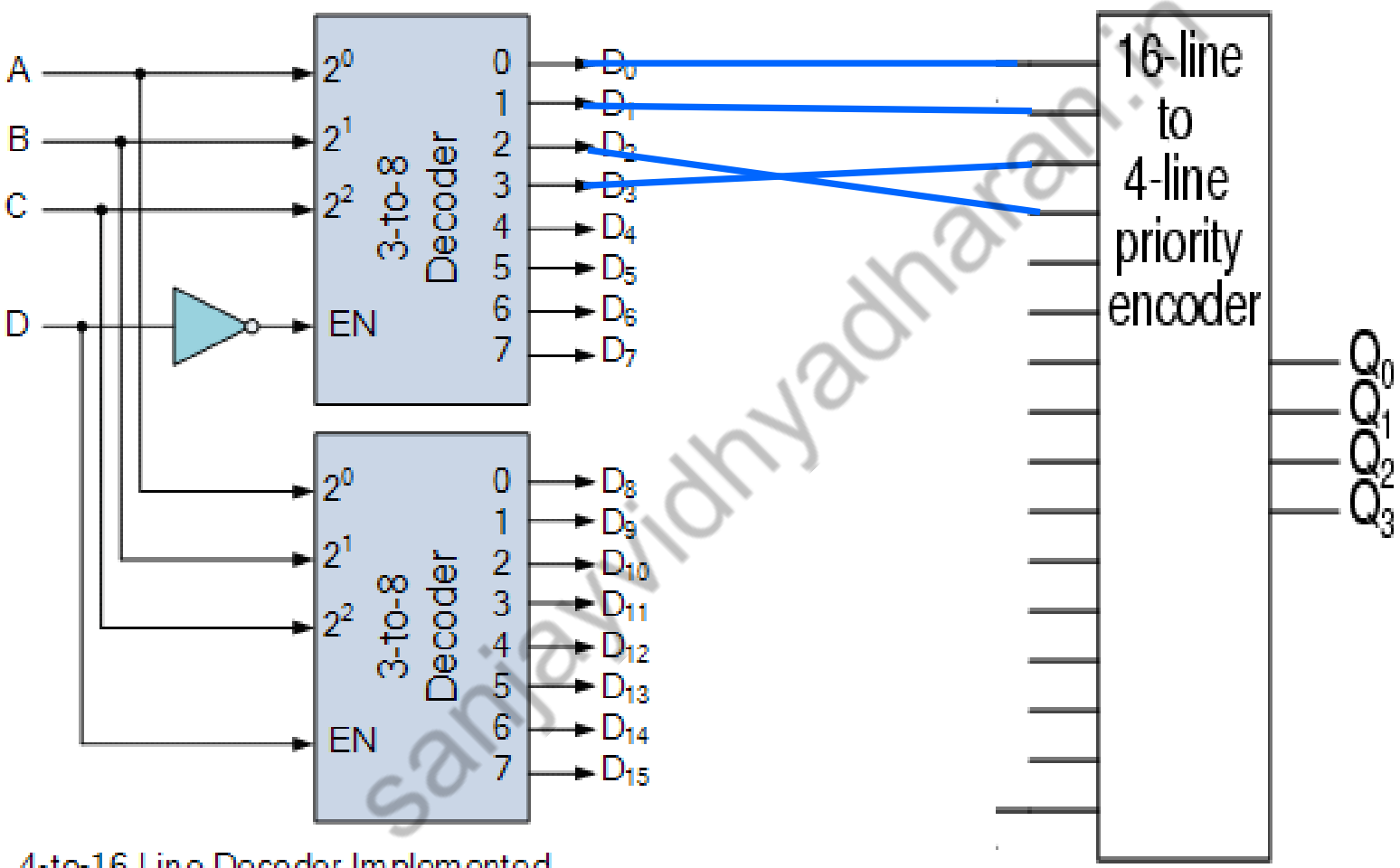
D_0	D_1	D_2	D_3	X	Y	V
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1
0	0	0	0	X	X	0



Priority Encoder

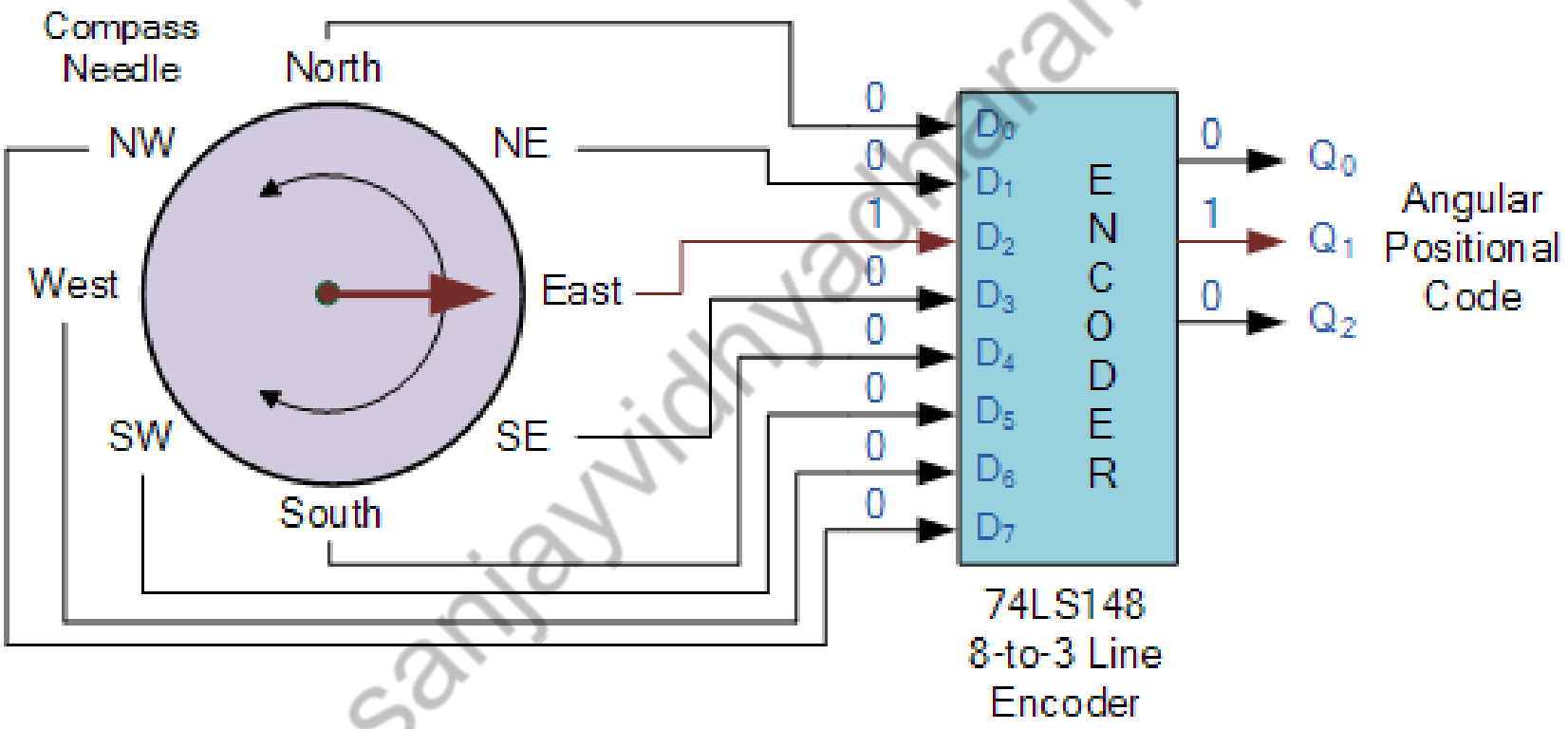


Binary to Gray Using Encoder-Decoder

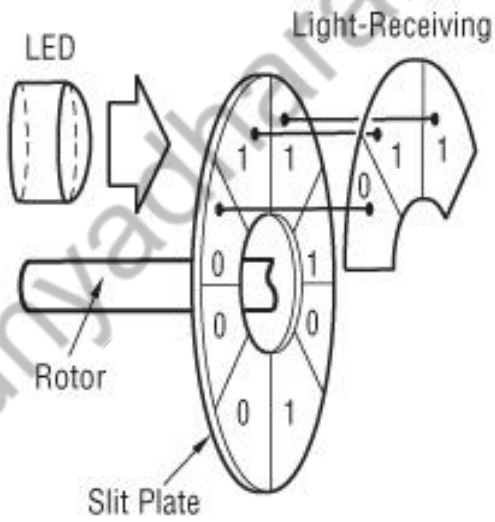


4-to-16 Line Decoder Implemented with two 3-to-8 Decoders

Encoder Applications

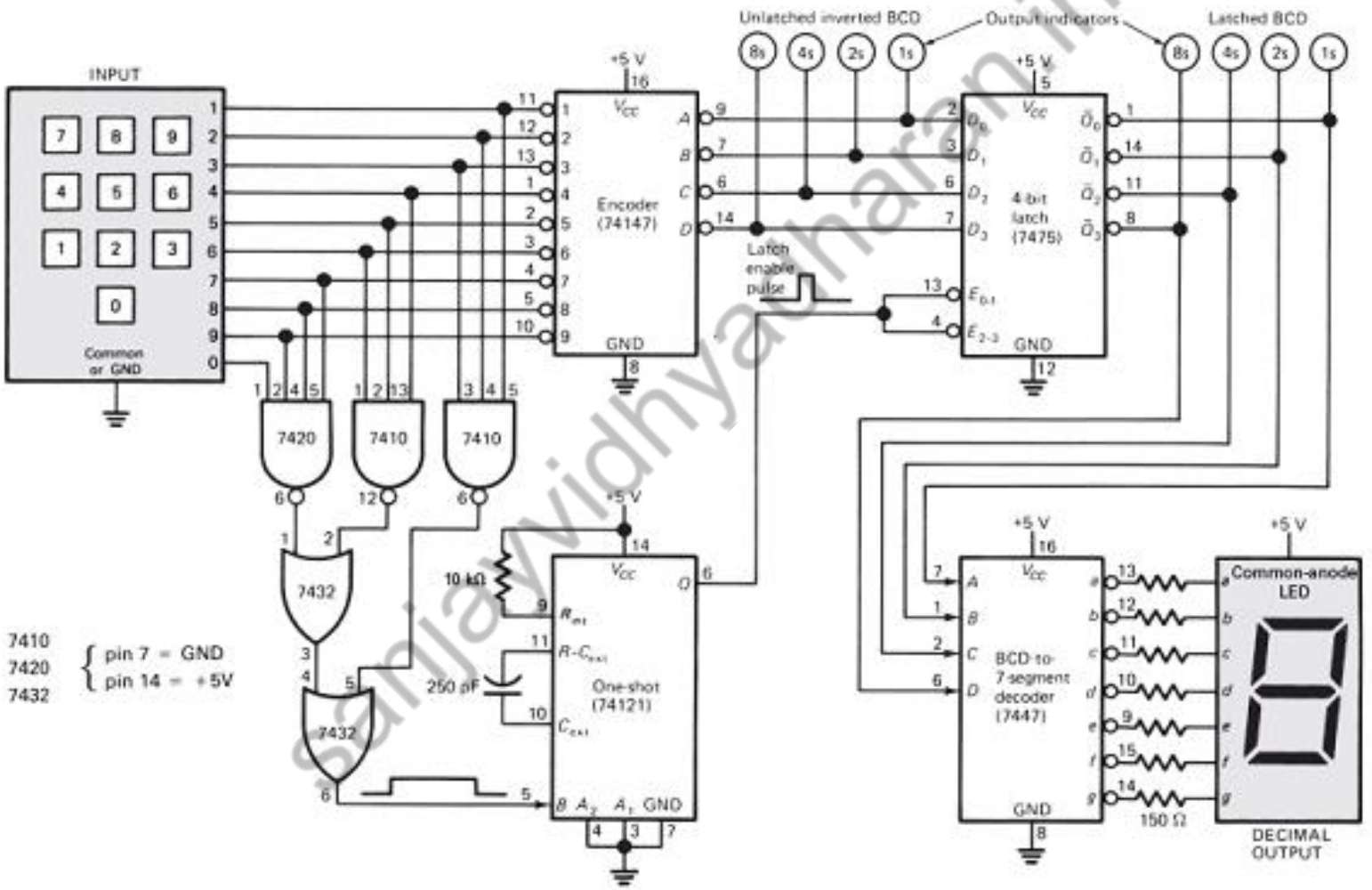


Encoder Applications



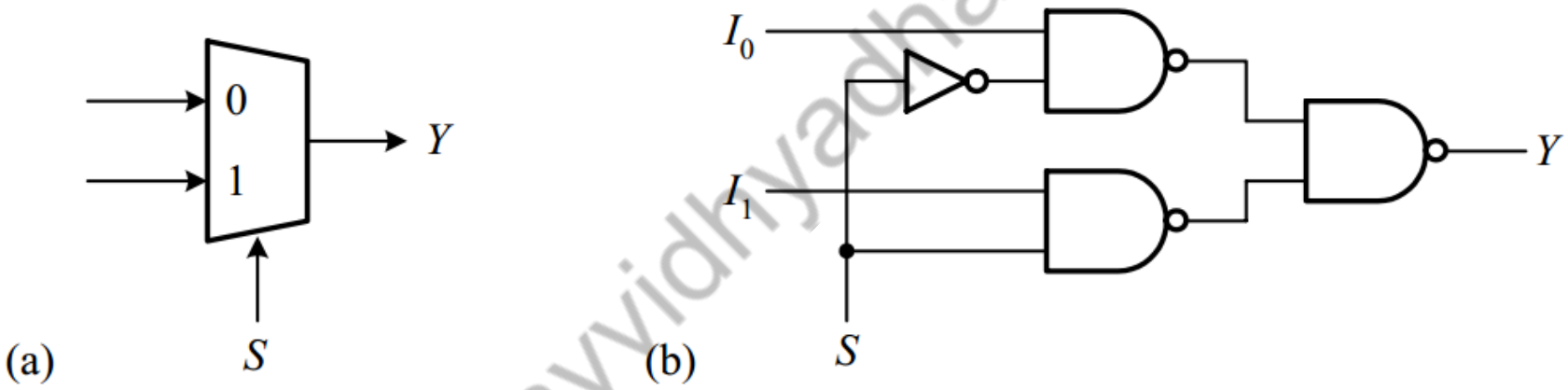
Sensor Output		Angle
011	➡	45°
001	➡	90°
000	➡	135°
100	➡	180°
010	➡	225°
101	➡	270°
110	➡	315°
111	➡	360°

Encoder Applications



Multiplexers

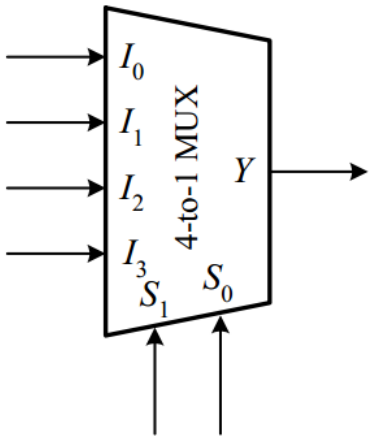
2-to-1 multiplexer



The (a) logic symbol and (b) logic circuit of a gate-based 2-to-1 multiplexer.

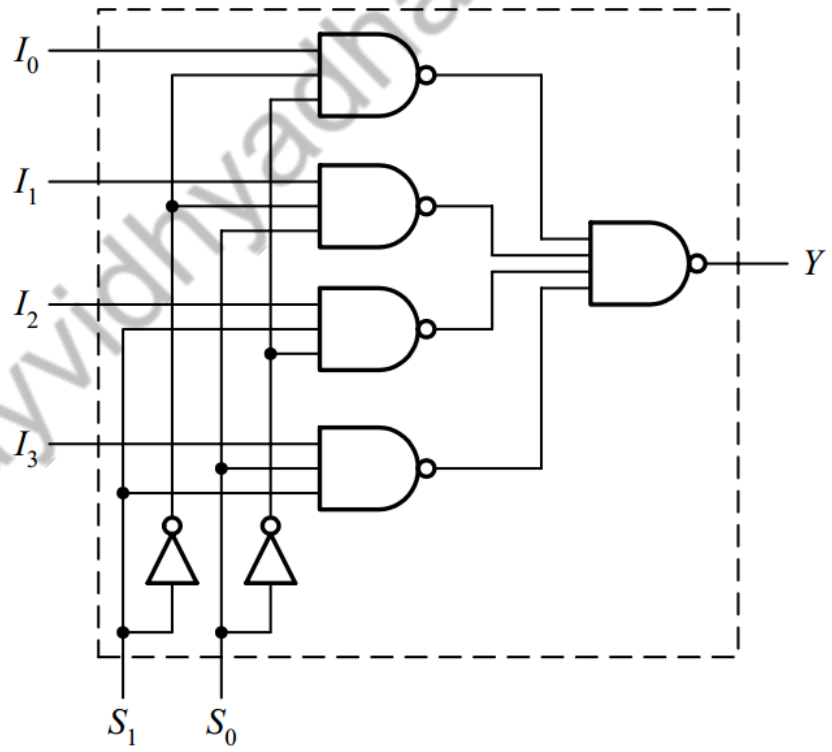
Multiplexers

4-to-1 multiplexer



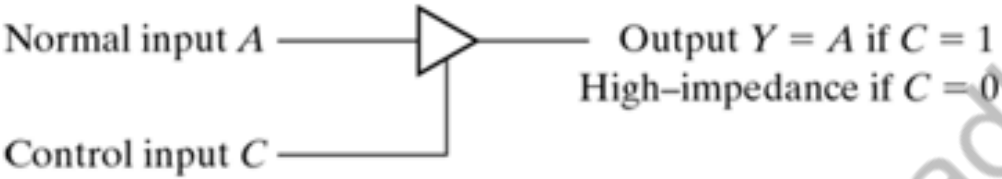
(a)

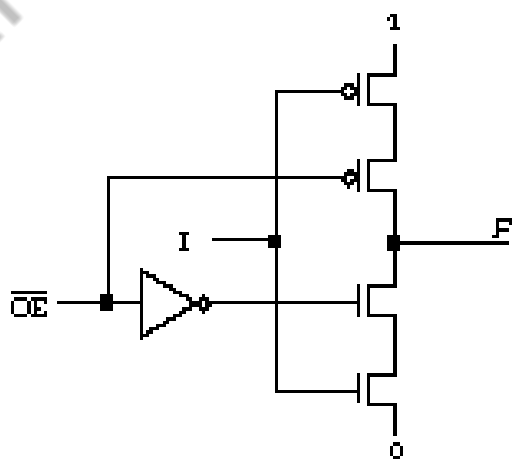
S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



Multiplexers

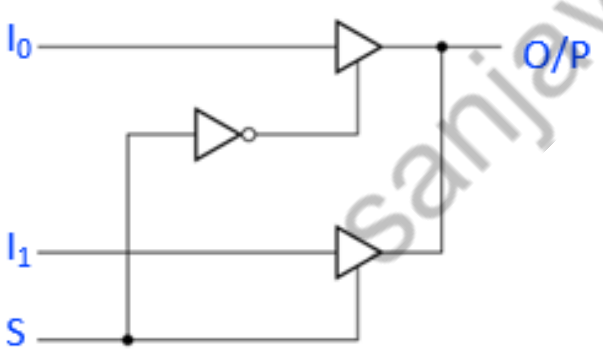
Tri State Buffer





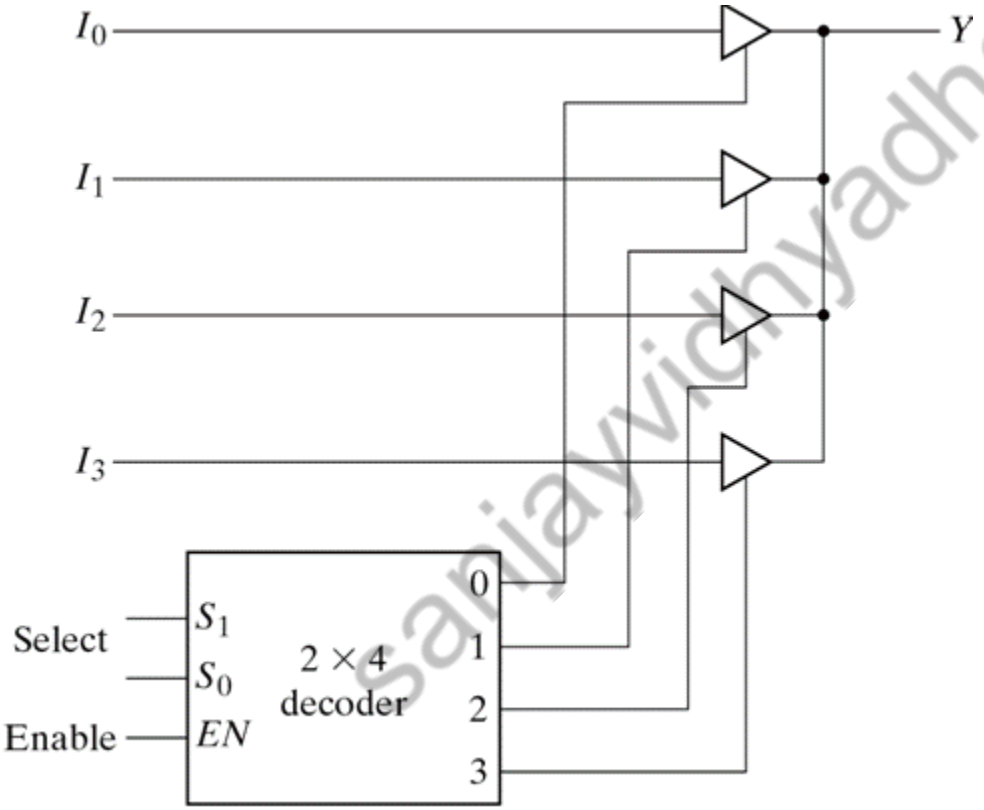
Switch implementation of tri-state gate

2:1 MUX using Tri-state buffer



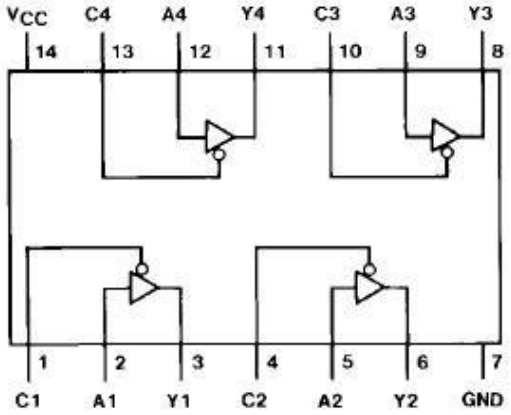
Multiplexers

4:1 MUX using Tri-state buffers and 2 to 4 Decoder

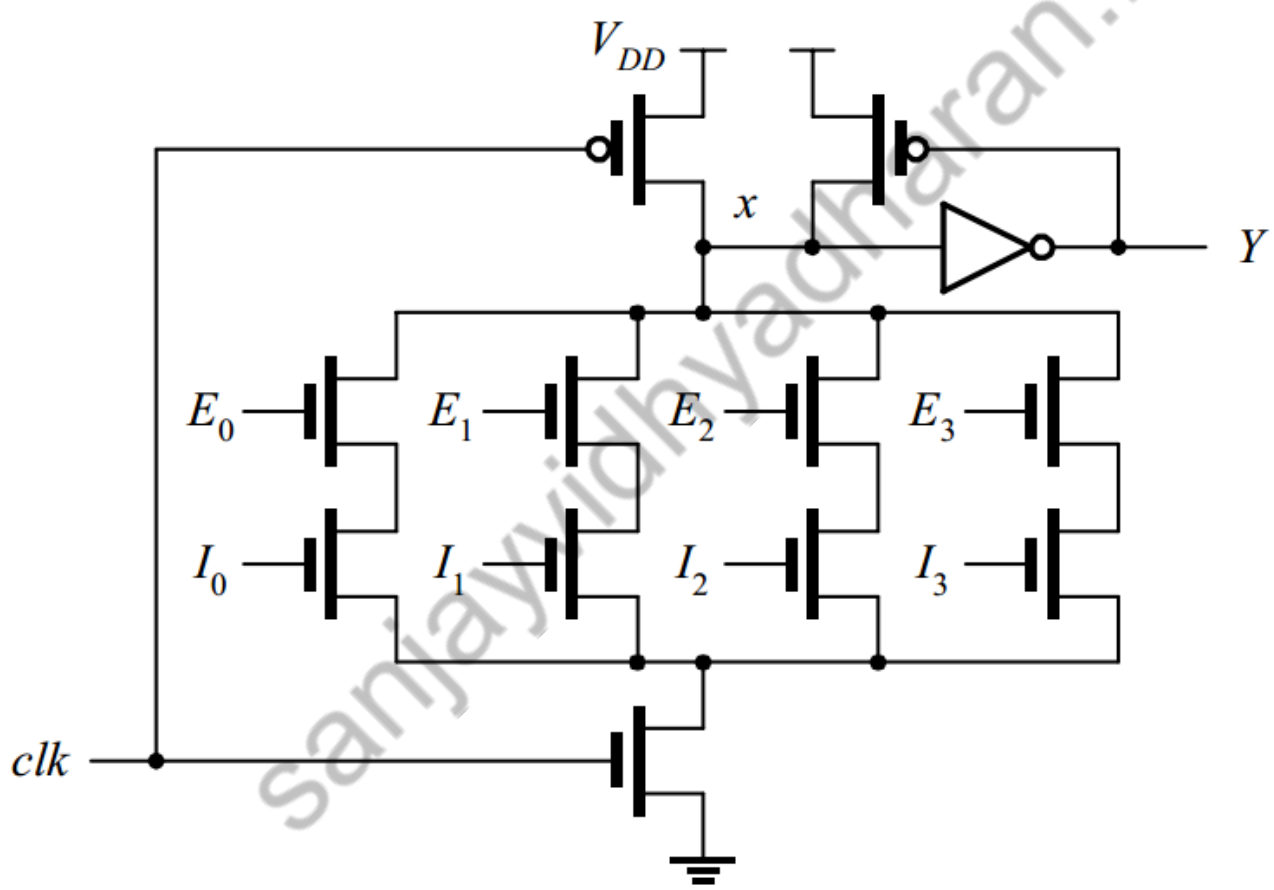


74HCT125 Quad Tri-State Buffer

LOGIC SYMBOL



Multiplexers

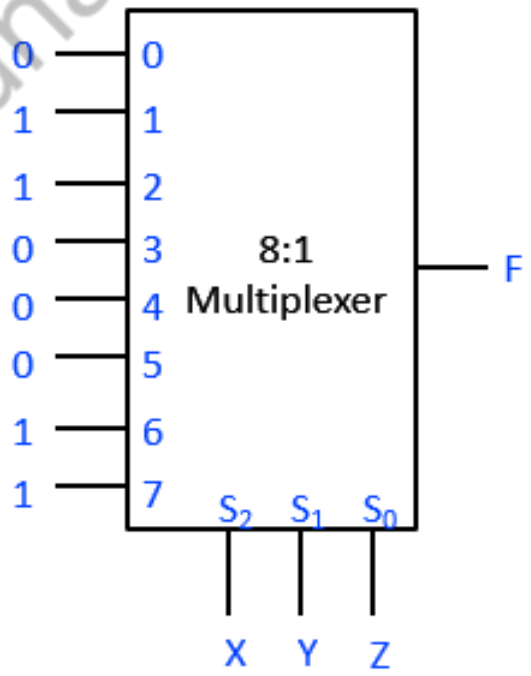


Multiplexers

Boolean Function implementation

$F(X,Y,Z) = \sum (1, 2, 6, 7)$

X	Y	Z	F
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

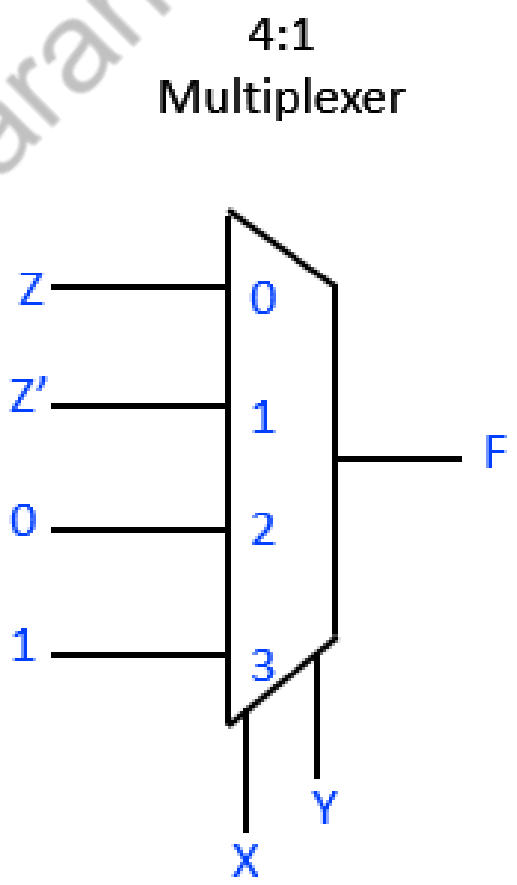


Multiplexers

Boolean Function implementation

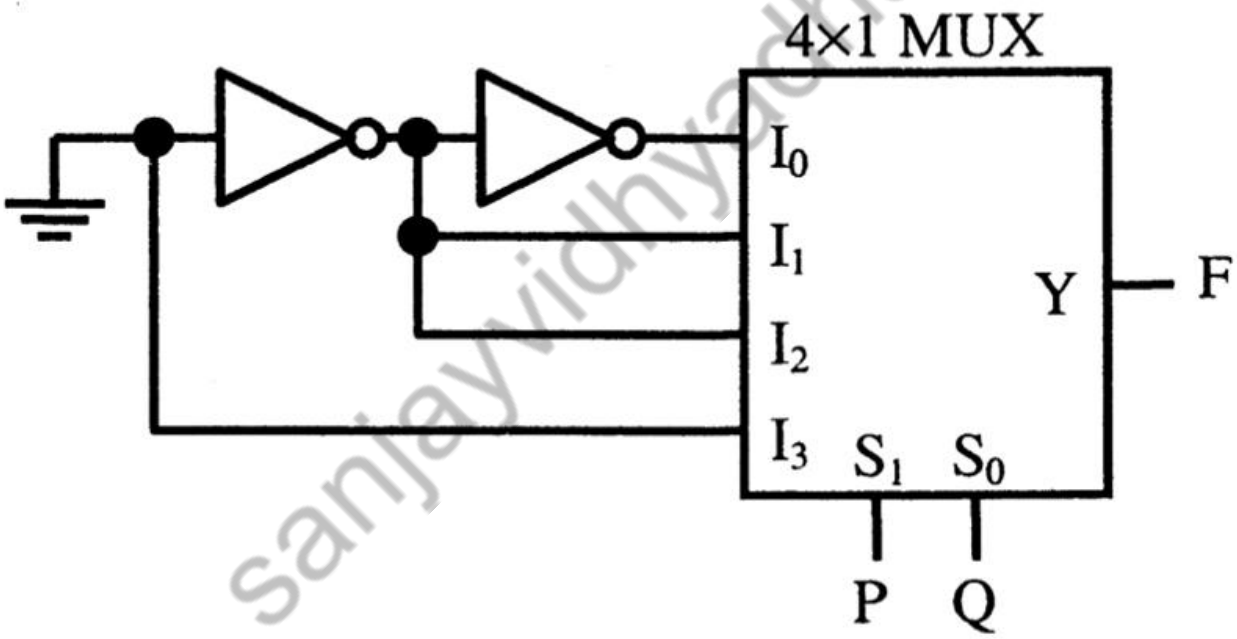
$F(X,Y,Z) = \sum (1, 2, 6, 7)$

X	Y	Z	F	
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	

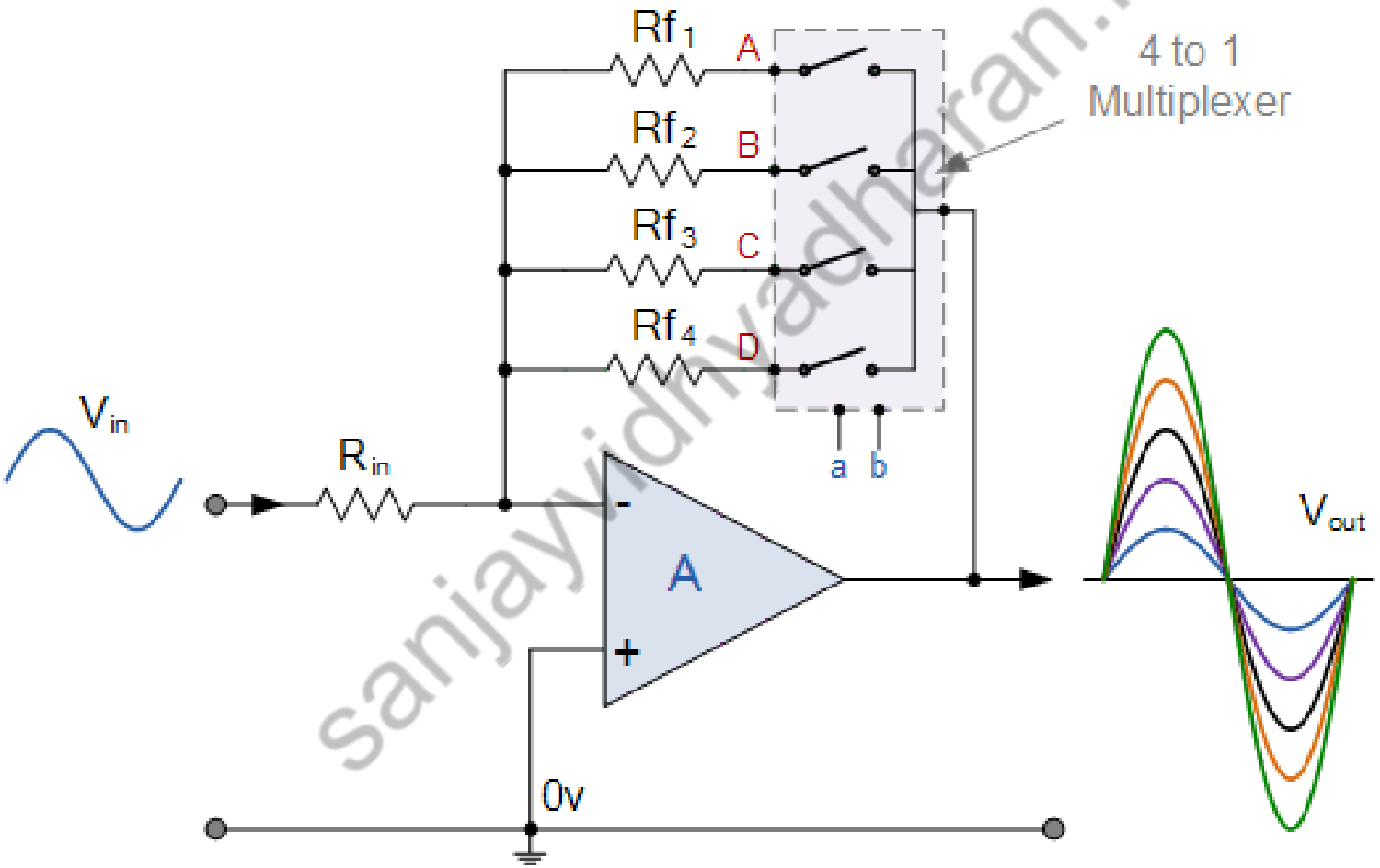


Multiplexers

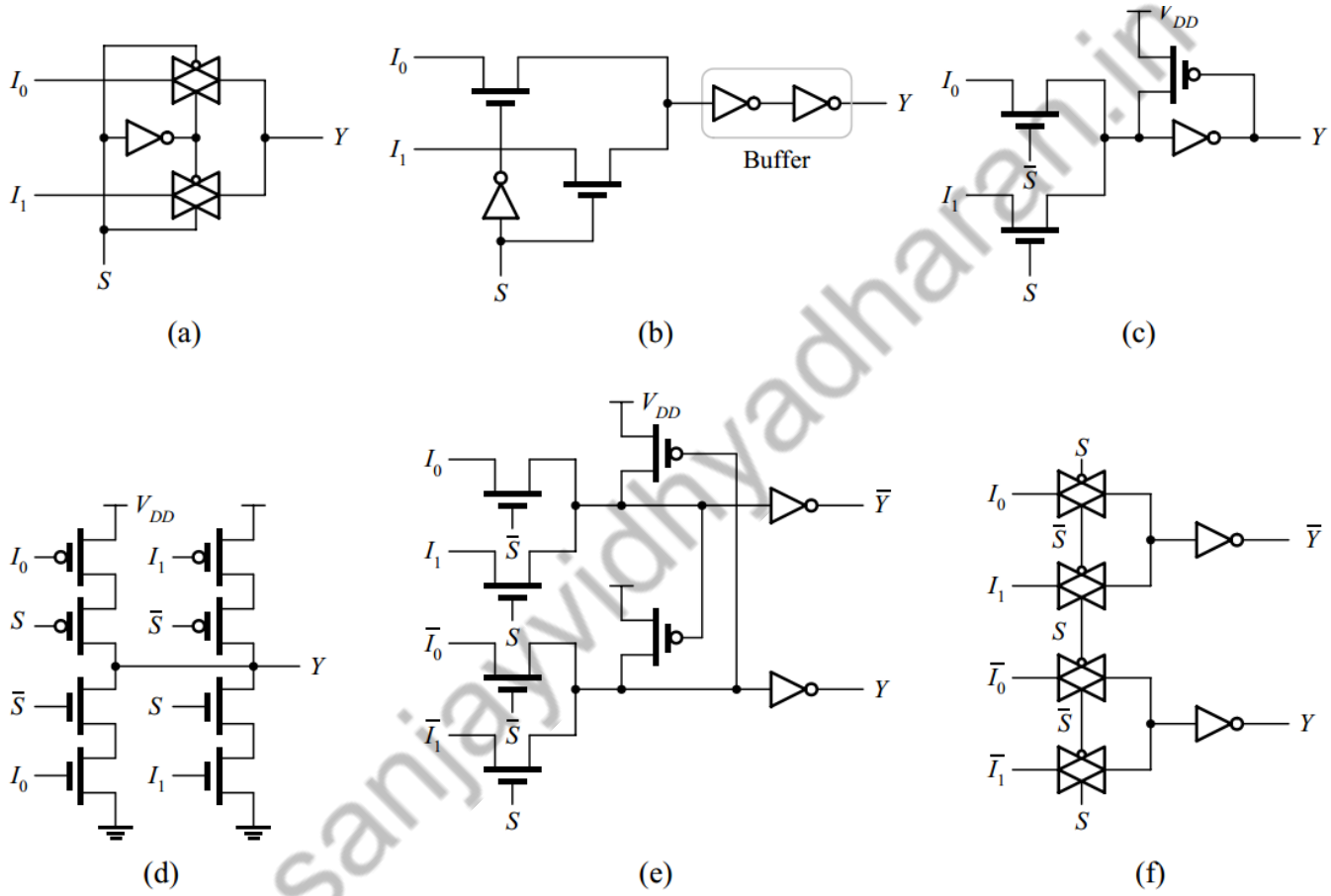
Boolean Function implementation



Multiplexers

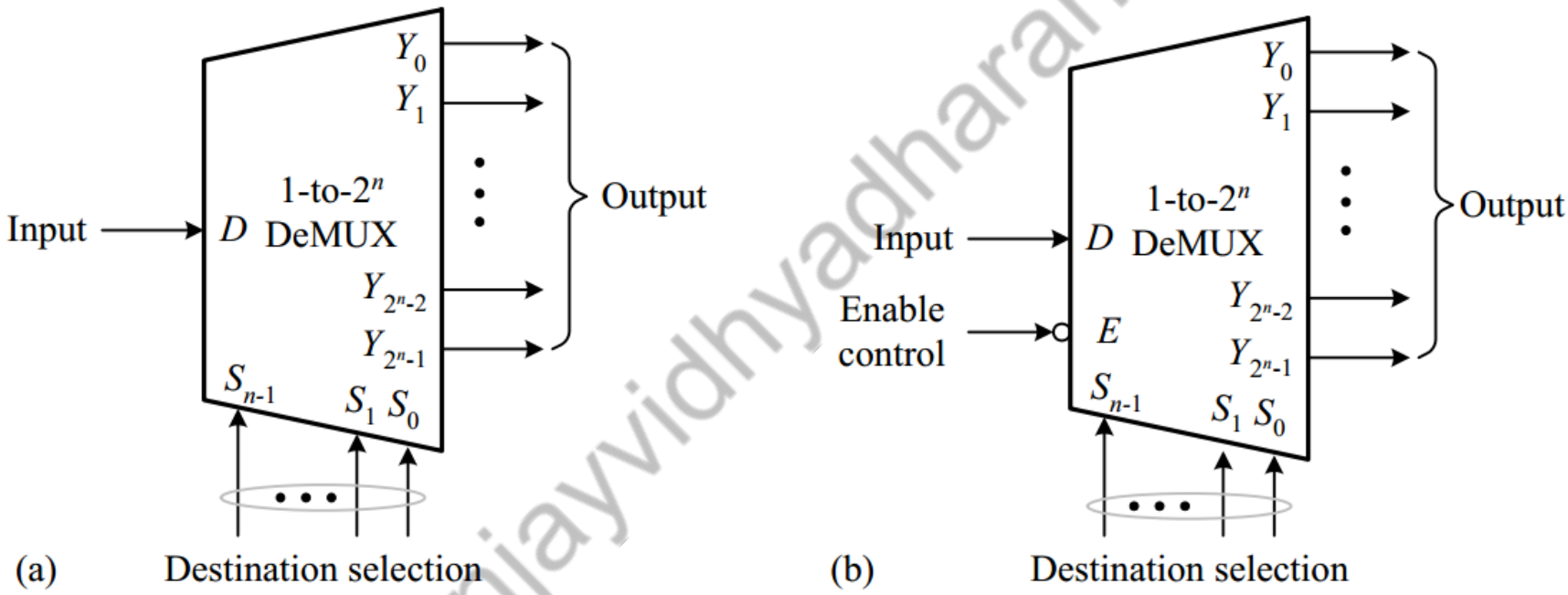


Multiplexers

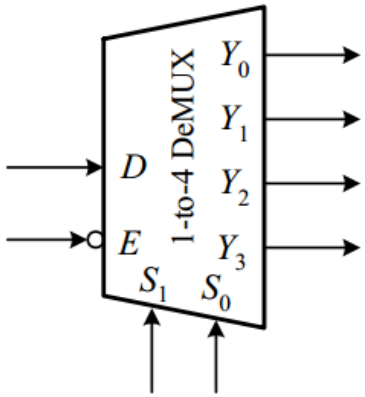


Switch-based 2-to-1 multiplexers: (a) TG-switch circuit; (b) Nmos circuit; (c) Lean Integration with Pass-Transistors (LEAP) circuit; (d) CMOS circuit; (e) CPL circuit; (f) DPL circuit.

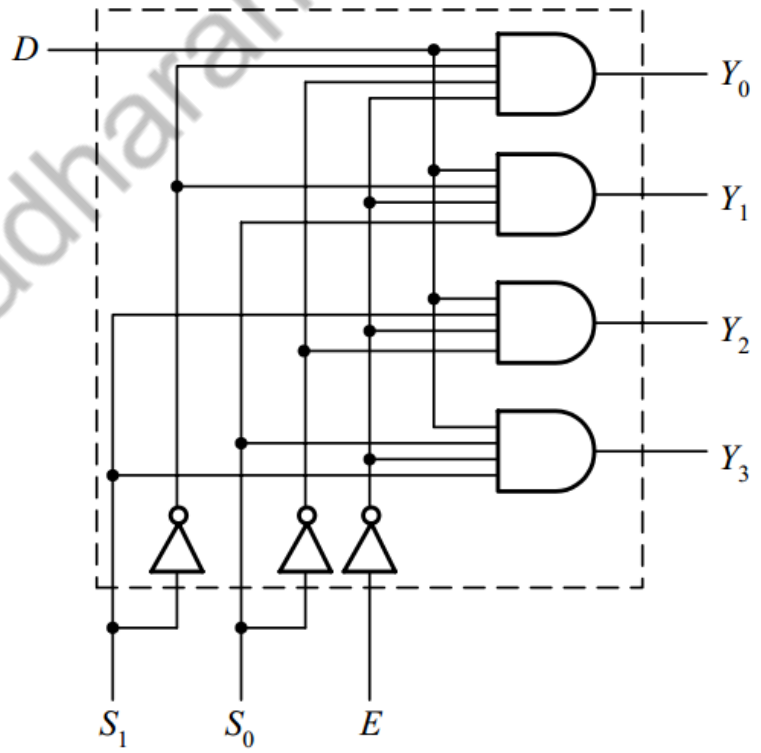
Demultiplexers



Demultiplexers



E	S_1	S_0	Y_0	Y_1	Y_2	Y_3
1	ϕ	ϕ	0	0	0	0
0	0	0	D	0	0	0
0	0	1	0	D	0	0
0	1	0	0	0	D	0
0	1	1	0	0	0	D



Magnitude Comparators

Magnitude Comparator

1	0	1
1	1	0

Magnitude Comparator

1	0	1
1	1	0

Magnitude Comparator

1-Bit Comparator

A_0

B_0

Let G_0 indicate greater, L_0 indicate Lesser and E_0 indicate equal

A_0	B_0	G_0	L_0	E_0
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

$G_0 = A_0B_0'$

$L_0 = A_0'B_0$

$E_0 = (A_0 \oplus B_0)' = (A_0 \odot B_0) = (G_0 + L_0)'$

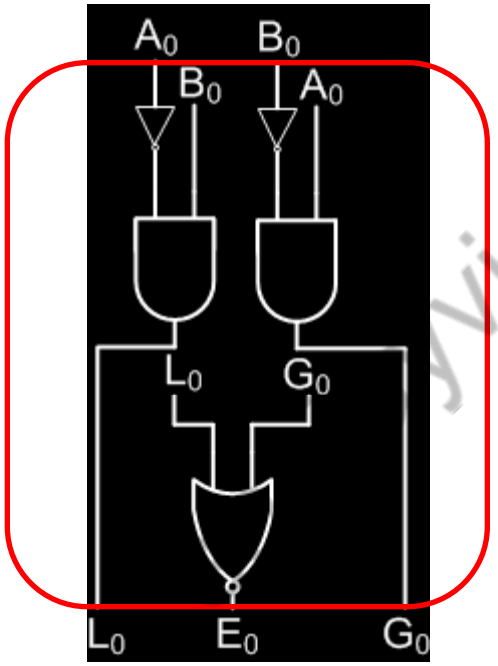
Magnitude Comparator

1-Bit Comparator

A_0

B

0



$G_0 = A_0B_0'$

$L_0 = A_0'B_0$

$E_0 = (G_0 + L_0)' = (A_0 \oplus B_0)'$

B_0 A_0

1-Bit Comp

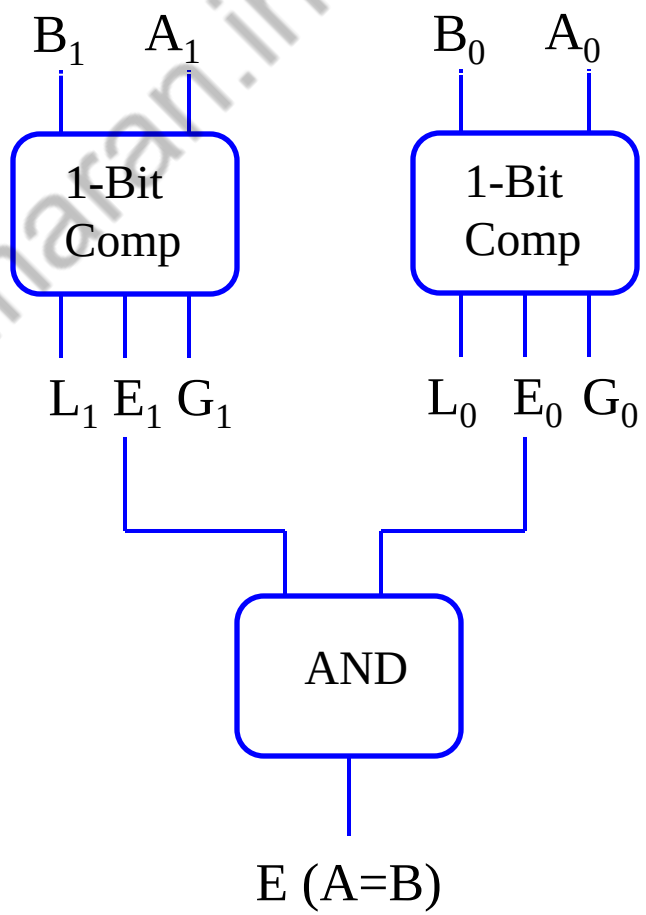
L_0 E_0 G_0

Magnitude Comparator

2-Bit Comparator

A A_1A_0

B $B_1\ B_0$



A = B When $A_0 = B_0$ and $A_1 = B_1$

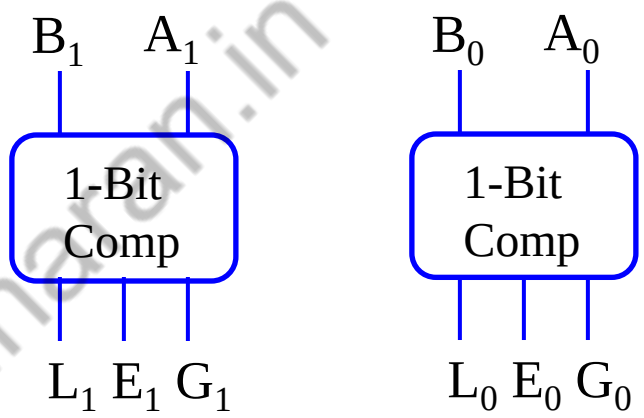
$E = E_0 . E_1$

Magnitude Comparator

2-Bit Comparator

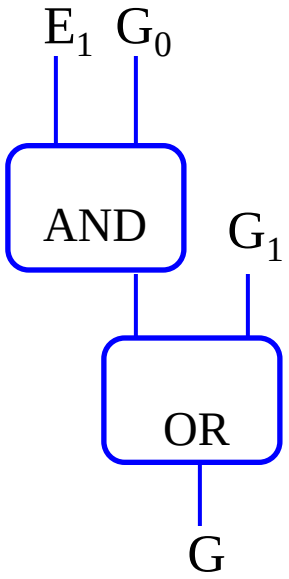
A A_1A_0

B B_1B_0



A > B When $A_1 > B_1$ OR
When $A_1 = B_1$ and $A_0 > B_0$

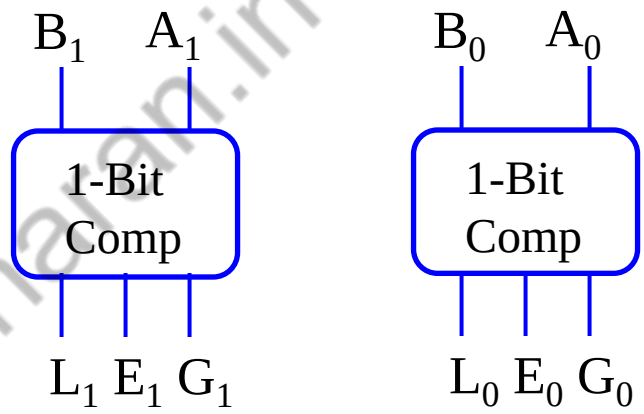
$G = G_1 + E_1 . G_0$



Magnitude Comparator

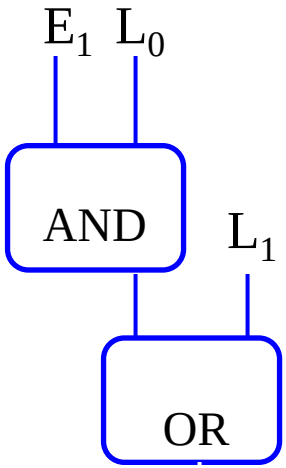
2-Bit Comparator

A	A_1A_0
B	B B 1 0



$A < B$ When $A_1 < B_1$ OR
When $A_1 = B_1$ and $A_0 < B_0$

$$L = L_1 + E_1 . L_0$$



Magnitude Comparator

1-Bit Comparator

$G = G_0 \quad L = L_0 \quad E = E_0$

2-Bit Comparator

$G = G_1 + E_1 . G_0 \qquad L = L_1 + E_1 . L_0 \qquad E = E_1 . E_0$

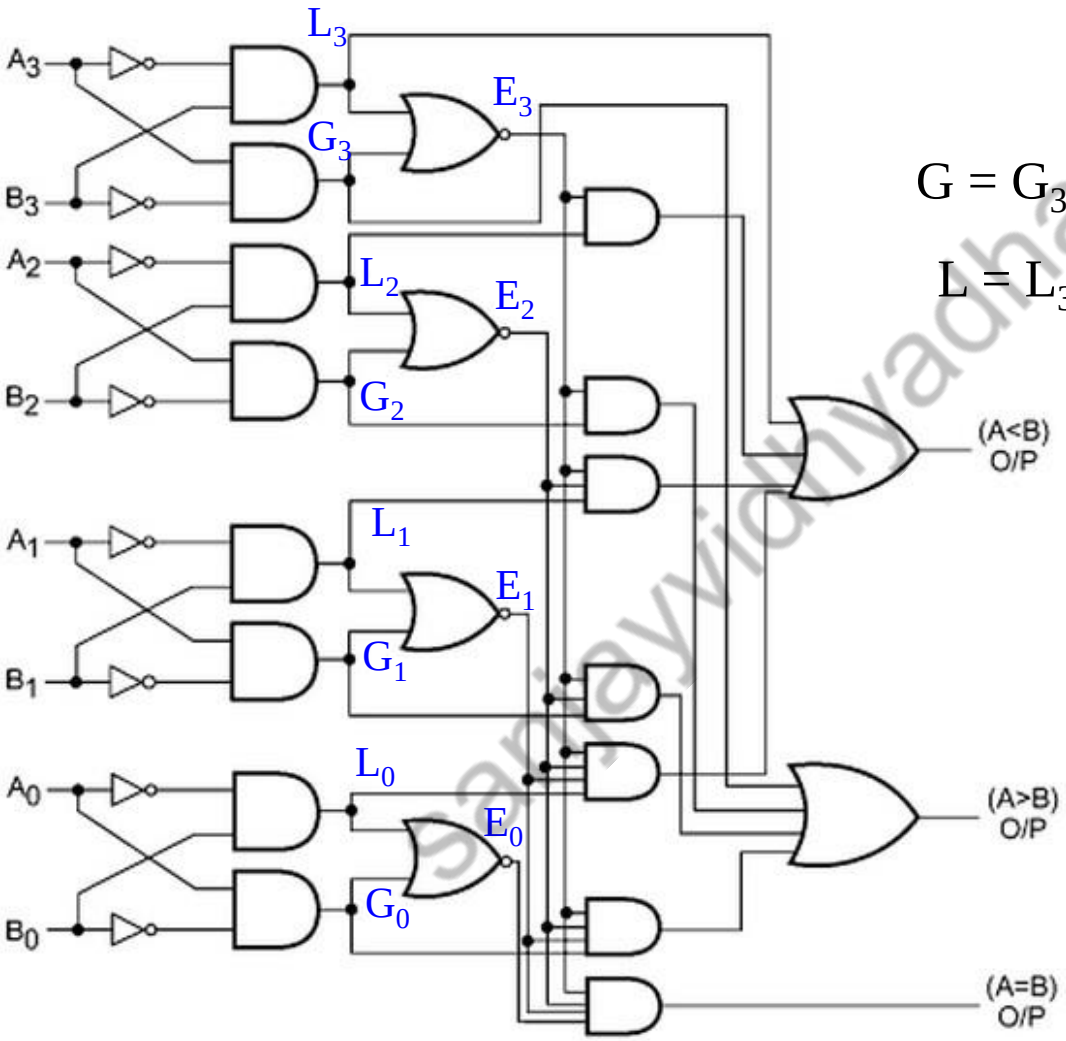
3-Bit Comparator

$G = G_2 + E_2 . G_1 + E_2 . E_1 . G_0$

$L = L_2 + E_2 . L_1 + E_2 . E_1 . L_0$

$E = E_2 . E_1 . E_0$

Magnitude Comparator



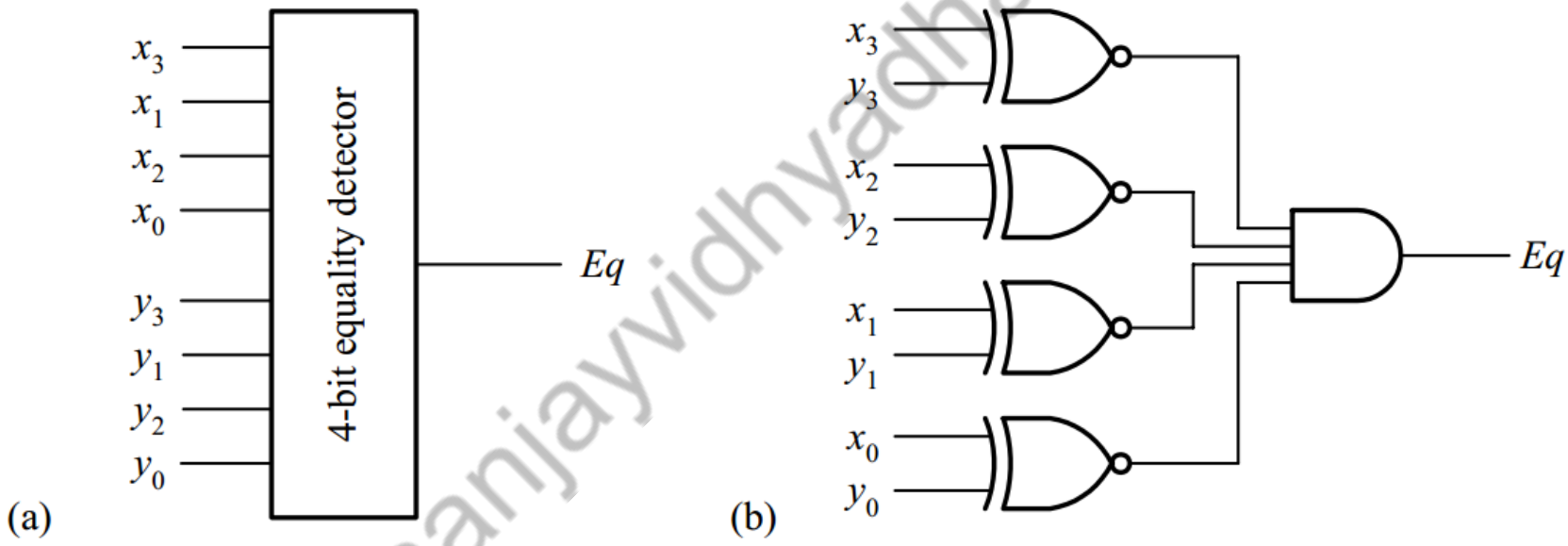
4-Bit Comparator

$$E = E_3.E_2.E_1.E_0$$

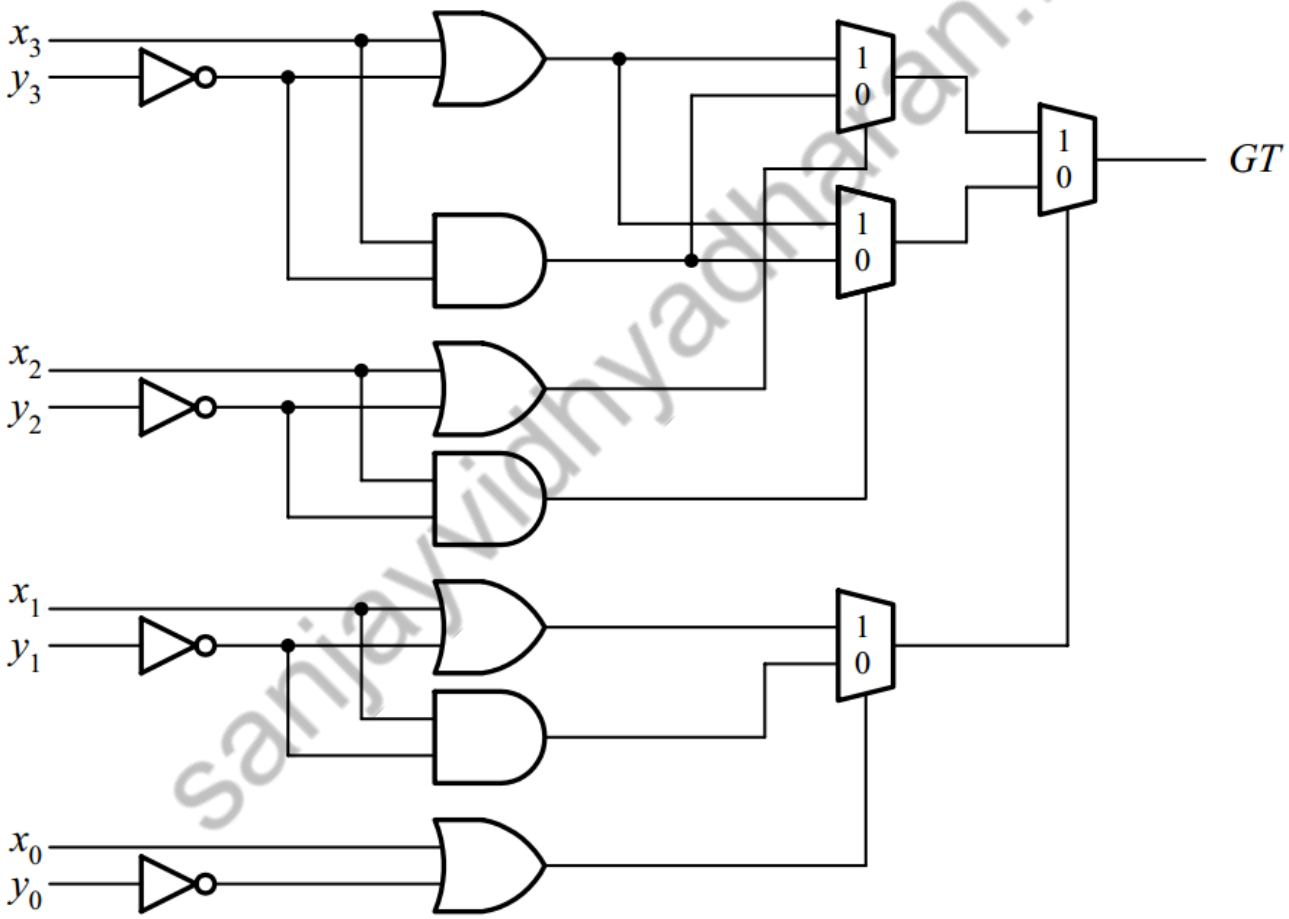
$$G = G_3 + E_3.G_2 + E_3.E_2.G_1 + E_3.E_2.E_1.G_0$$

$$L = L_3 + E_3.L_2 + E_3.E_2.L_1 + E_3.E_2.E_1.L_0$$

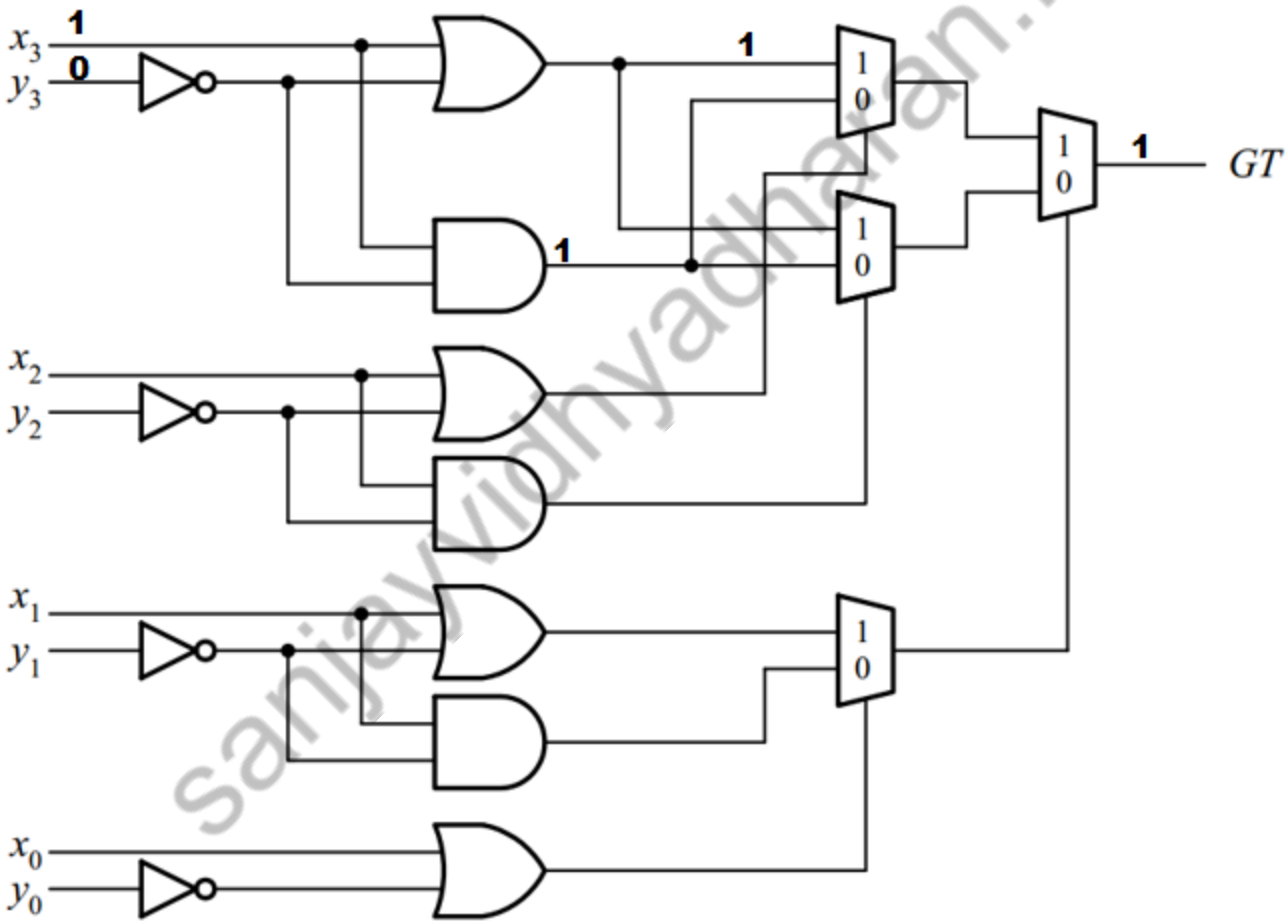
Magnitude Comparator



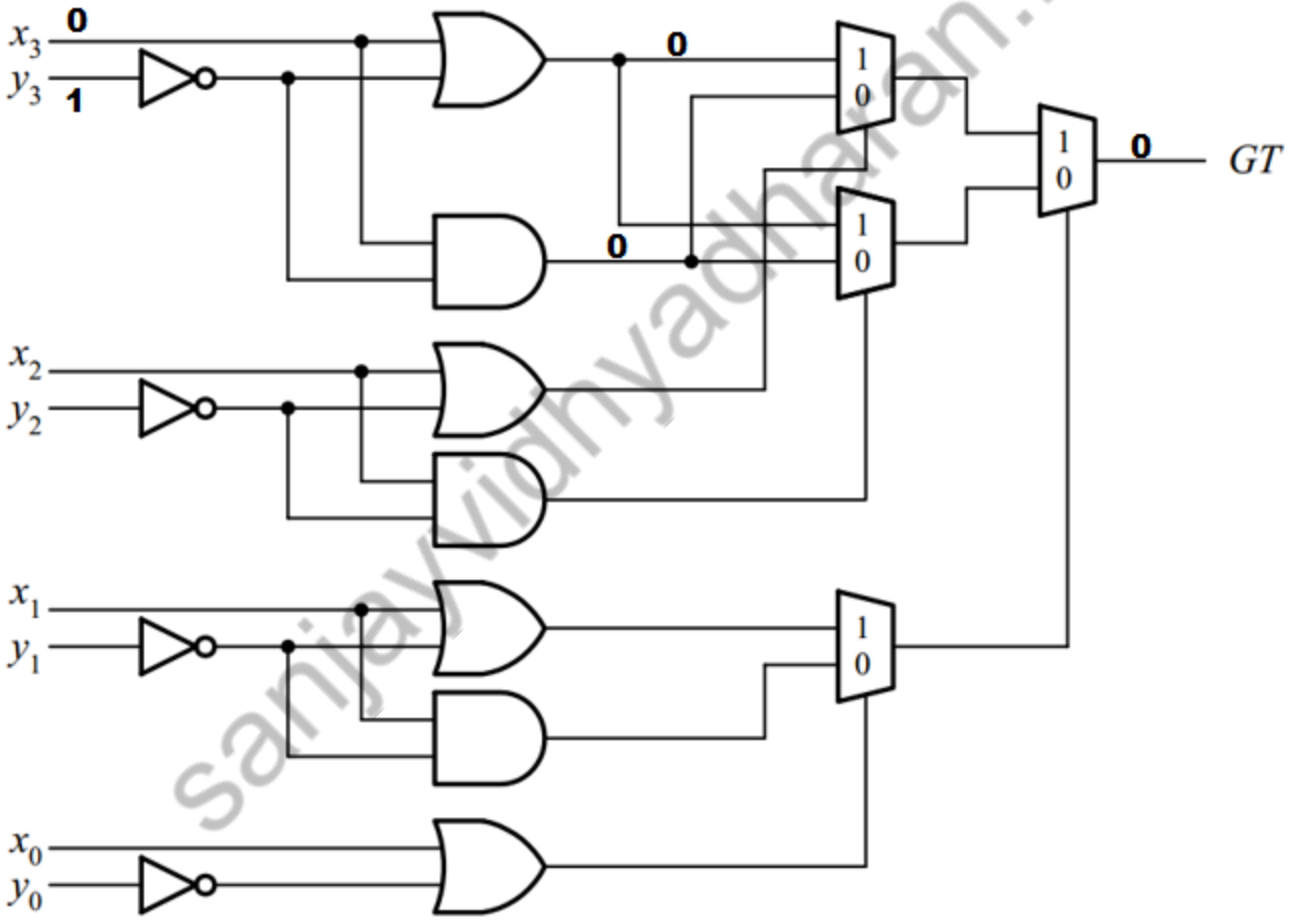
Magnitude Comparator



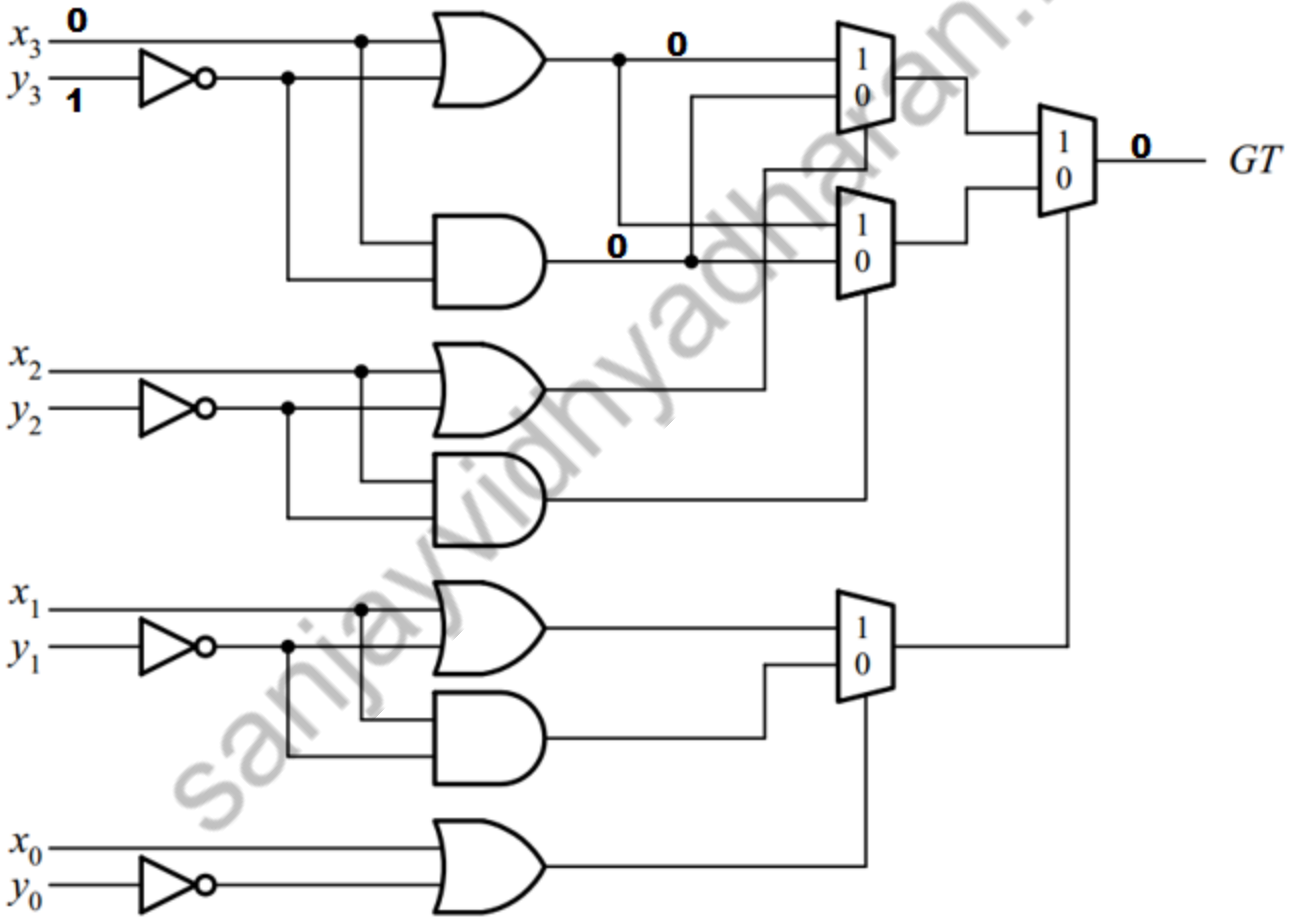
Magnitude Comparator



Magnitude Comparator



Magnitude Comparator



Thank you