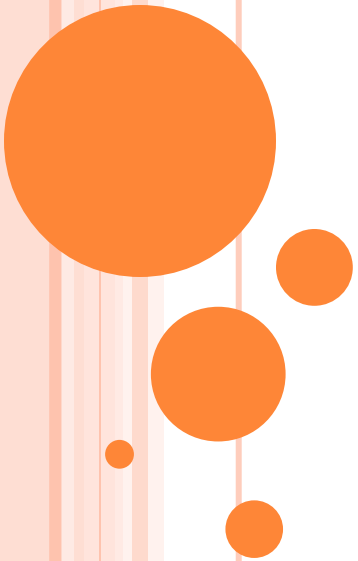
DECODER, ENCODER, PRIORITY ENCODER, DEMULTIPLEXER

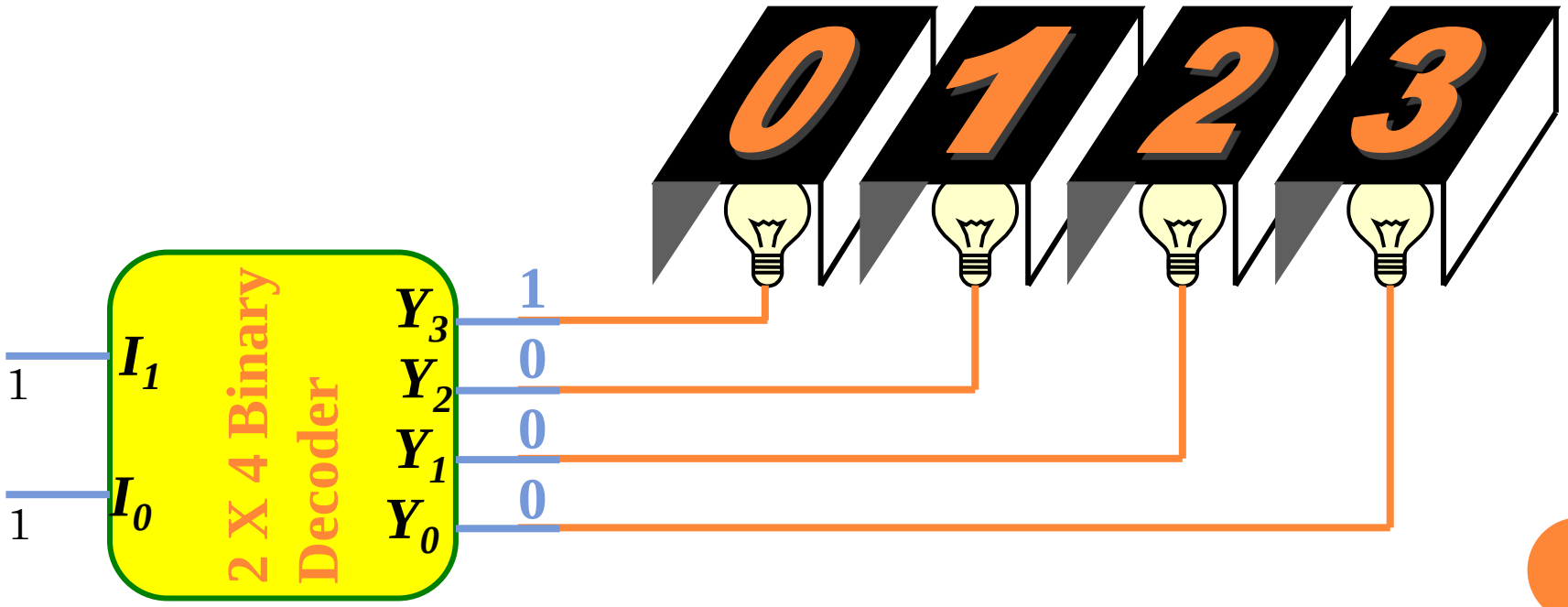


Decoders

- A decoder is a combinational circuit that converts binary information from *n input lines to a maximum of* 2^n *unique output lines.*

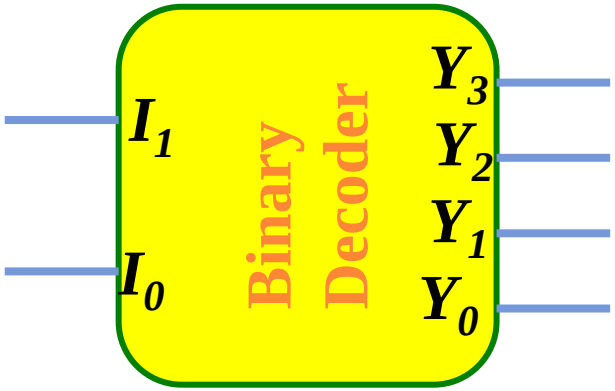
Decoders

- Extract "*Information*" from the code
- Binary Decoder
- Example: 2-to-4 Line Decoder

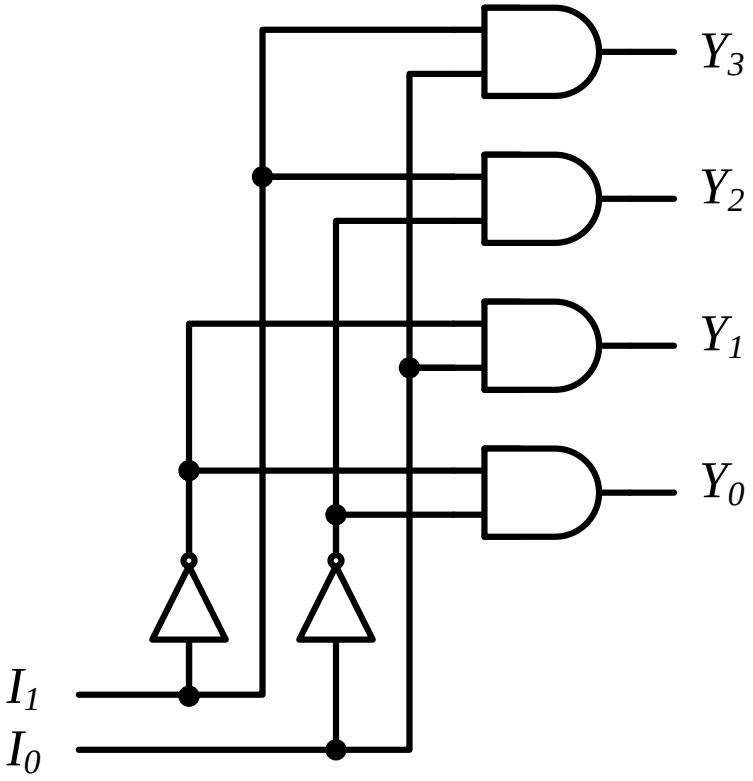


Decoders

- 2-to-4 Line Decoder



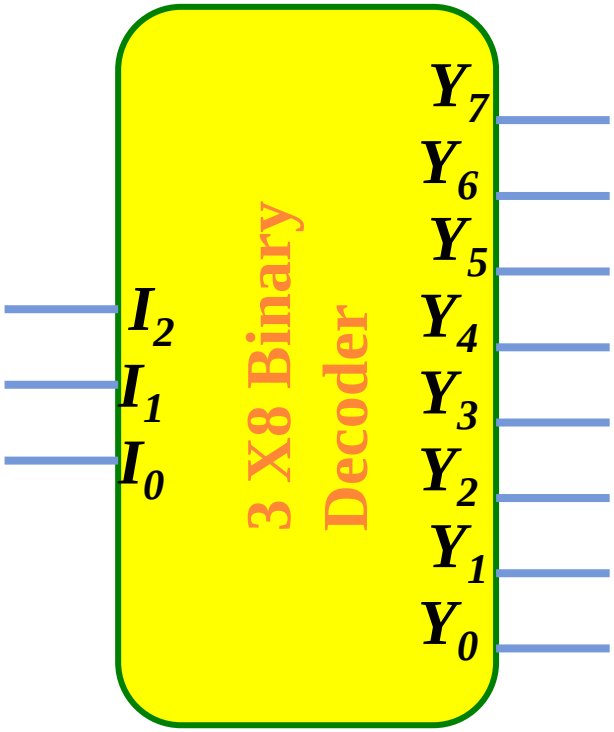
I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0



$$Y_3 = I_1 I_0 \qquad Y_2 = I_1 \bar{I}_0$$

$$Y_1 = \bar{I}_1 I_0 \qquad Y_0 = \bar{I}_1 \bar{I}_0$$

3-to-8 Line Decoder



Truth table:

I_2	I_1	I_0	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	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

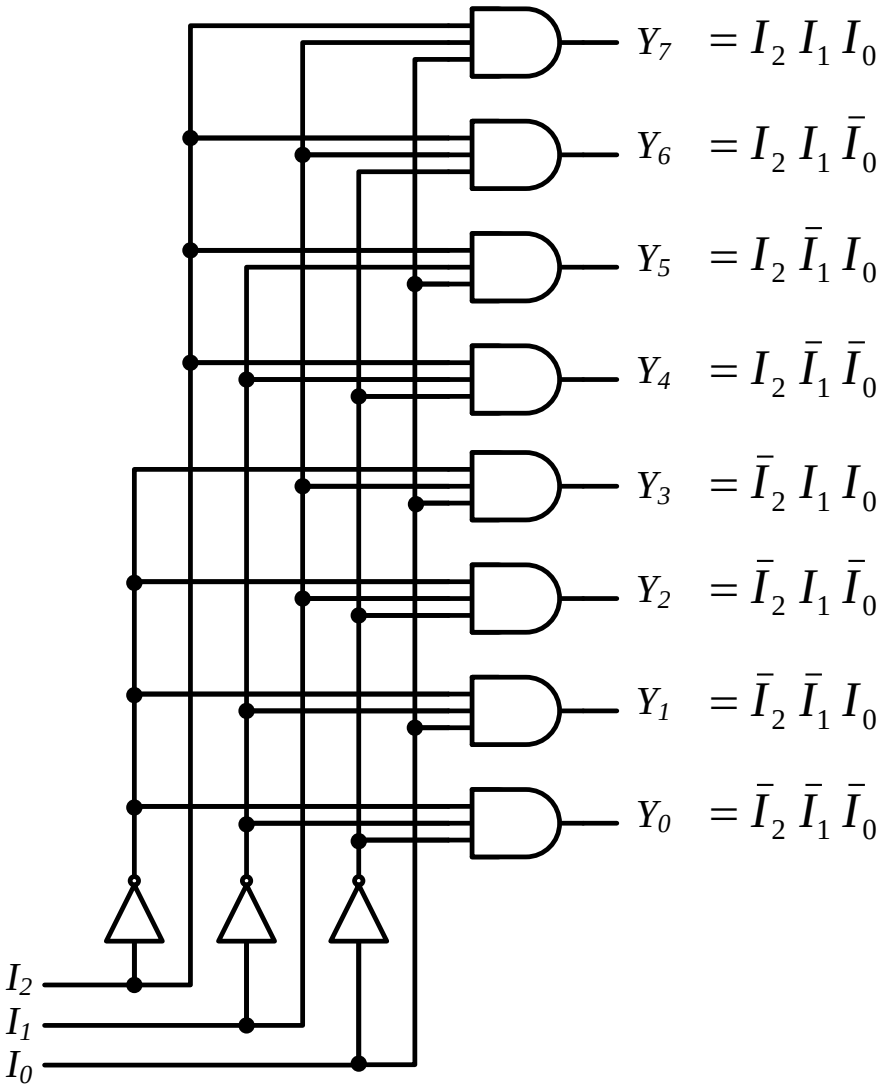
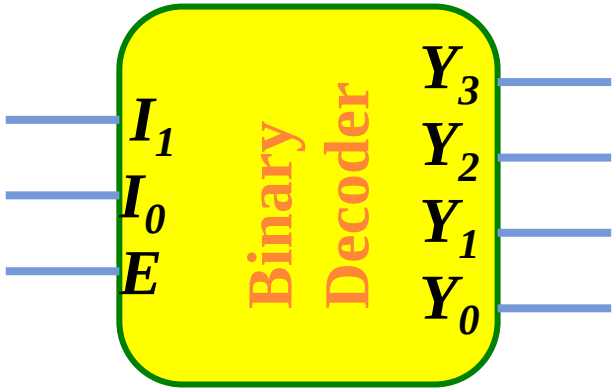


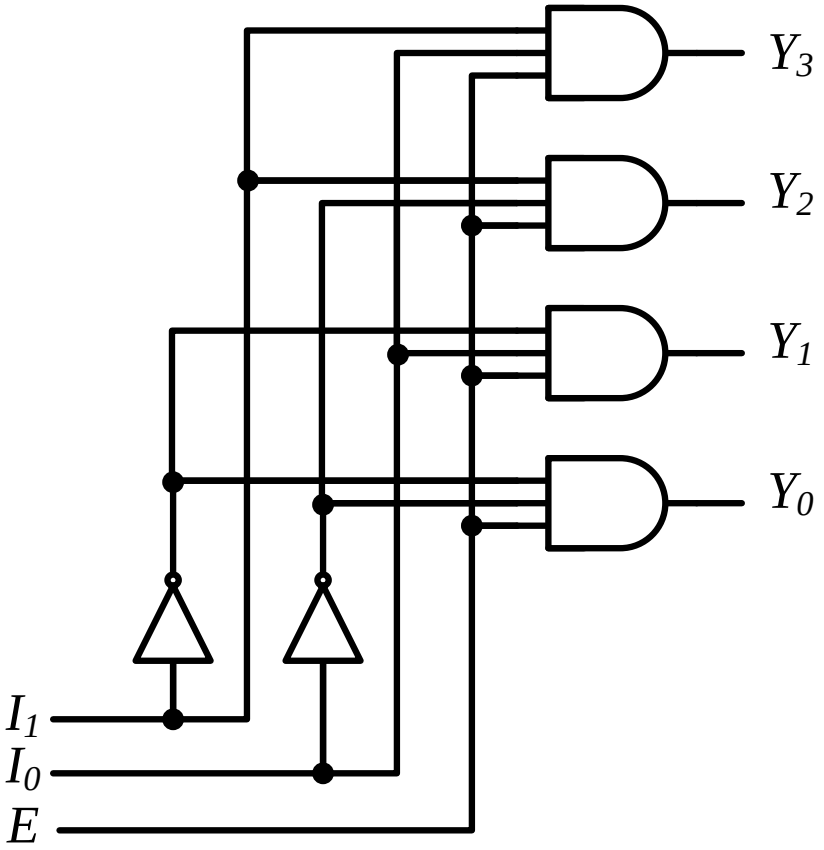
Fig: 3X8 decoder

Decoders

- "*Enable*" Control



E	I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

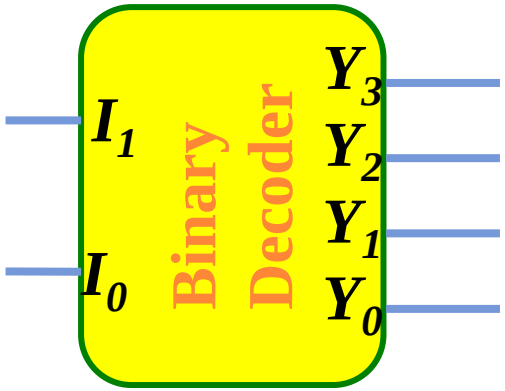


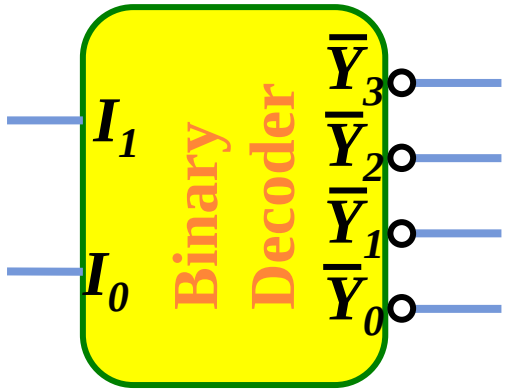
Decoders

- Active-High / Active-Low

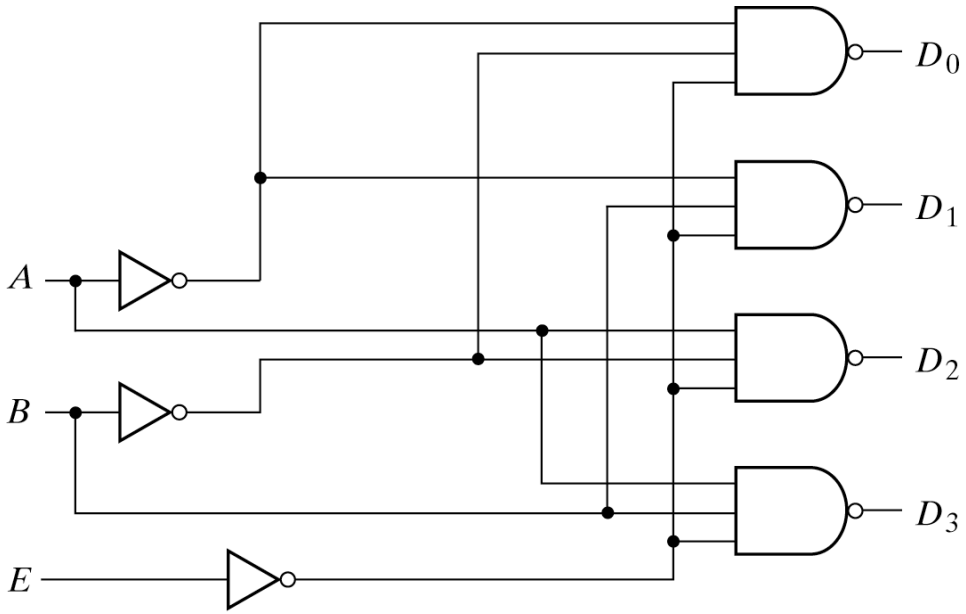
I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1





Decoders



(a) Logic diagram

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

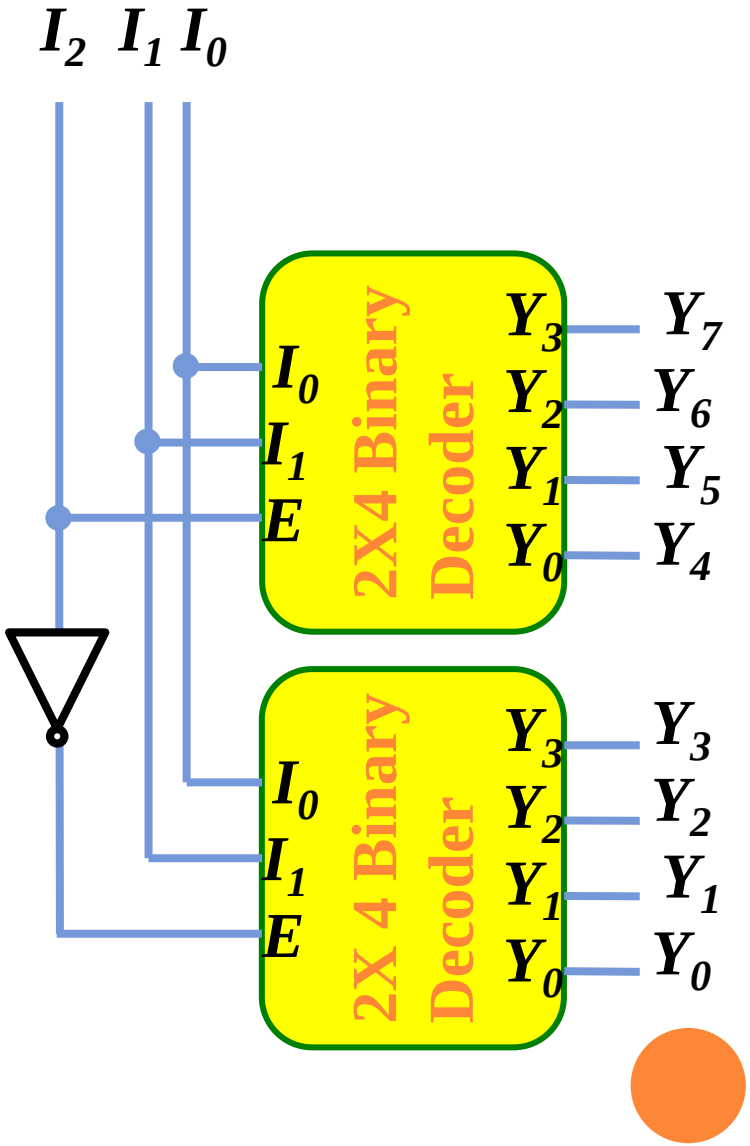
(b) Truth table

Fig. 4-19 2-to-4-Line Decoder with Enable Input

Design a 3x8 line decoder using two 2x4 decoders

- Expansion

I_2	I_1	I_0	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	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

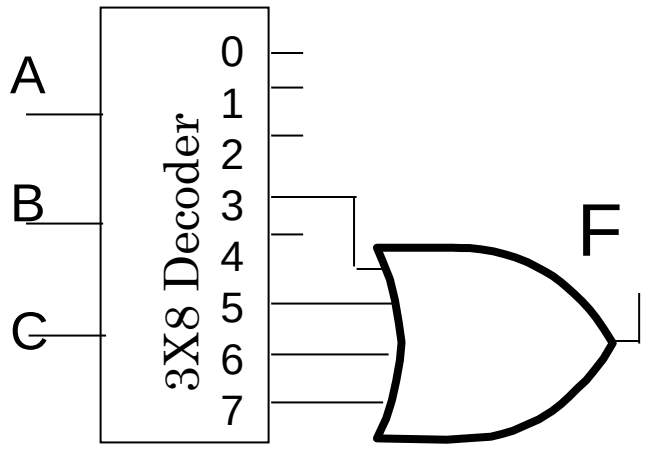


Boolean function Implementation Using Decoders

- Design and implement F= Σ(3,5,6,7) using a 3-to-8 decoder.

Truth Table:

A B C	F
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1



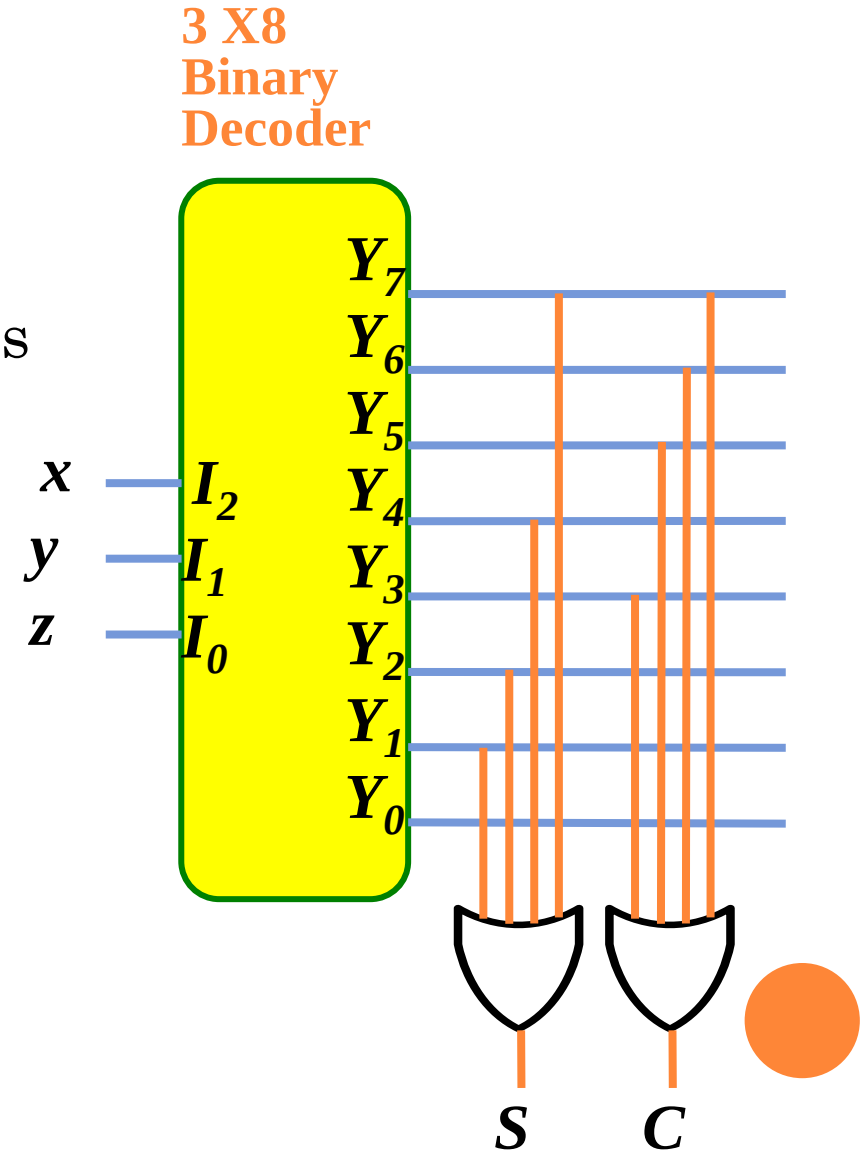
Boolean function Implementation Using Decoders

- Each output is a minterm
- All minterms are produced
- Sum the required minterms

Example: Full Adder

$S(x, y, z) = \sum(1, 2, 4, 7)$

$C(x, y, z) = \sum(3, 5, 6, 7)$



Encoders

- Does reverse operation to decoder
- An encoder has 2^n **(or fewer)** input lines and n output lines
- Constraint – only one input is active at a time
- Example: 4X2, 8X3,16X4, 512X9 encoder etc

Encoders

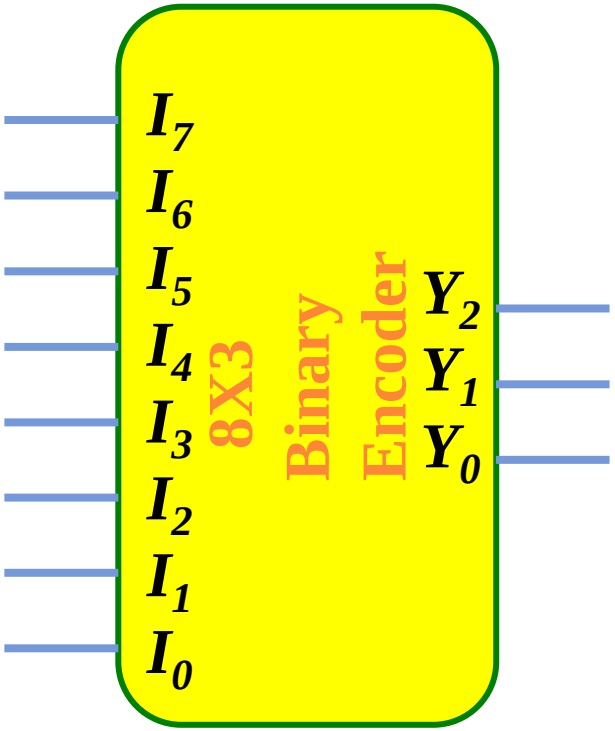
- Octal-to-Binary Encoder (8-to-3)

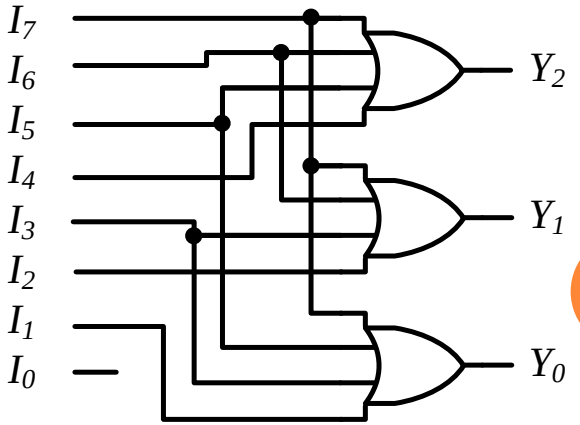
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$Y_2 = I_7 + I_6 + I_5 + I_4$$

$$Y_1 = I_7 + I_6 + I_3 + I_2$$

$$Y_0 = I_7 + I_5 + I_3 + I_1$$





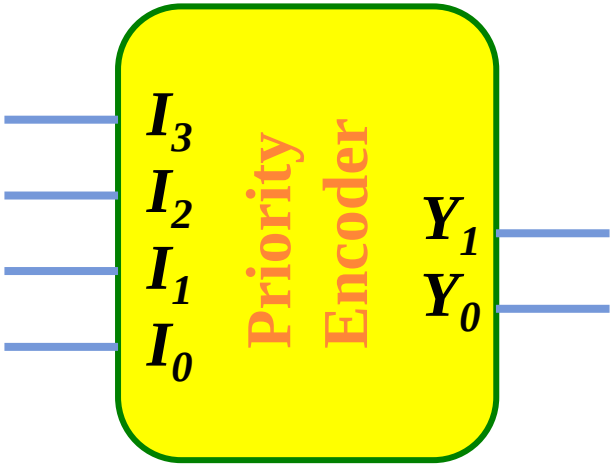
Priority Encoders

- Encoder with priority function
  - Multiple inputs may be true simultaneously
  - Higher priority input gets the precedence

Priority Encoders

- 4-Input Priority Encoder

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	0	x	x
0	0	0	1	0	0
0	0	1	x	0	1
0	1	x	x	1	0
1	x	x	x	1	1

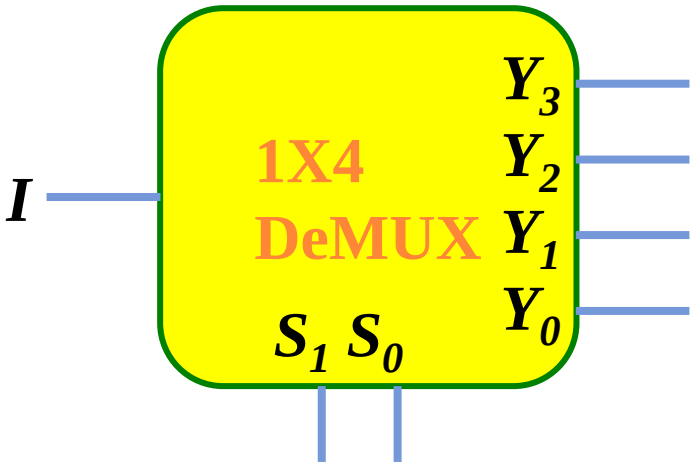


$$Y_1 = I_3 + \bar{I}_3 I_2 = I_3 + I_2$$

$$Y_0 = I_3 + \bar{I}_3 \bar{I}_2 I_1 = I_3 + \bar{I}_2 I_1$$

DeMultiplexers

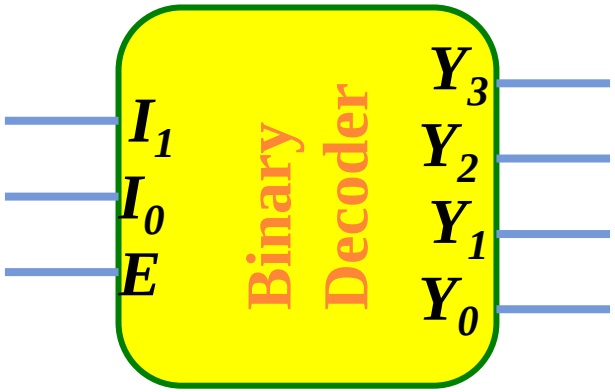
- A circuit receives information from a single line and directs it to one of 2^n possible output lines.
- The selection of a specific output line is controlled by the bit values of n selection lines.



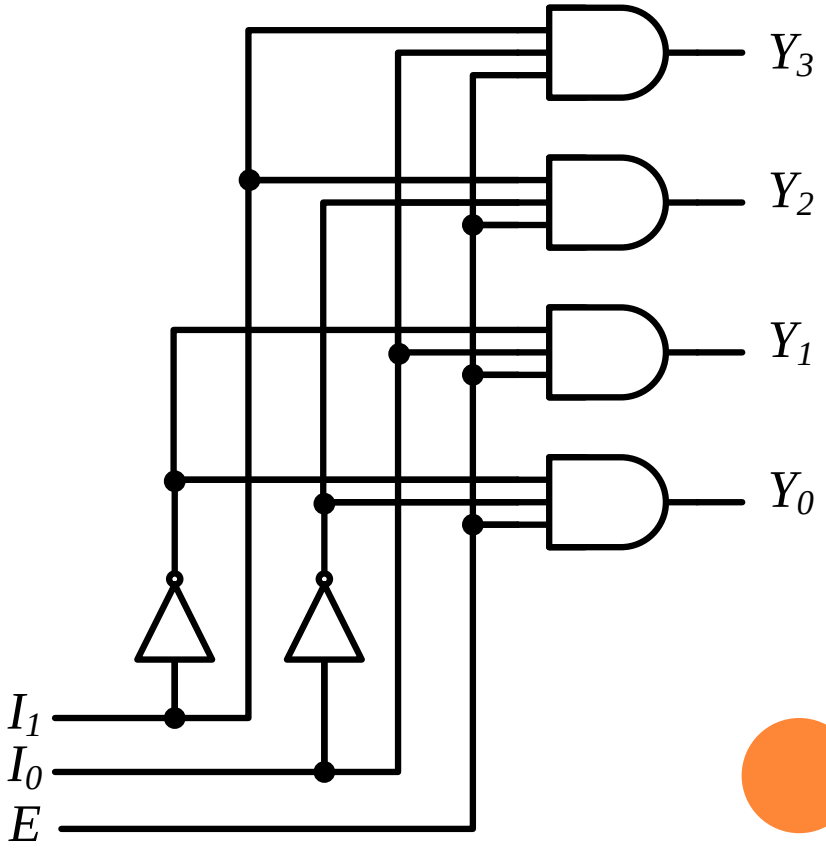
S_1	S_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	I
0	1	0	0	I	0
1	0	0	I	0	0
1	1	I	0	0	0

DeMultiplexers / Decoders

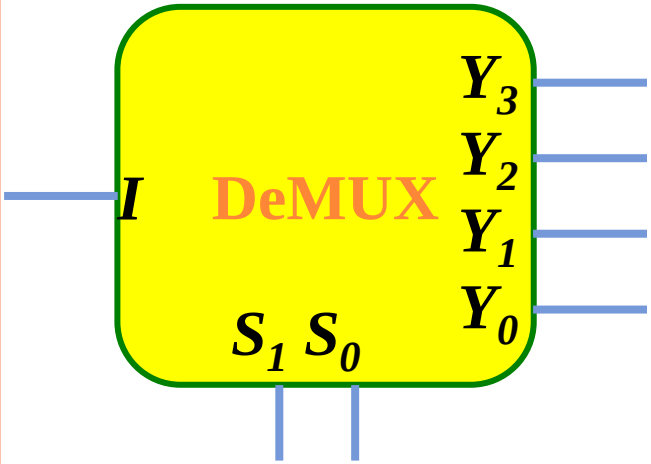
- A decoder with enable input can function as a demultiplexer



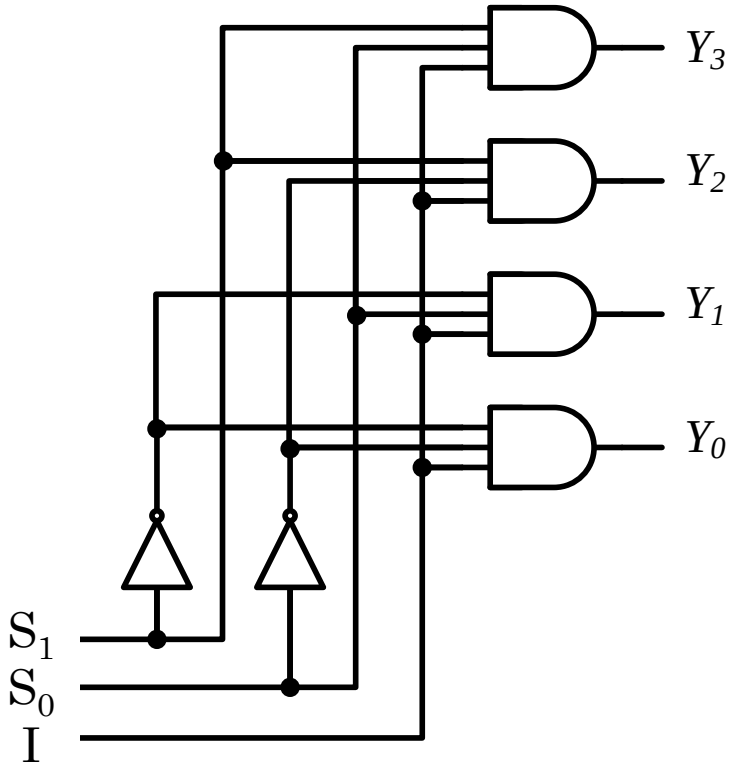
E	I_1	I_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0



DEMULTIPLEXERS



I	S_1 S_0	Y_3 Y_2 Y_1 Y_0
0	x x	0 0 0 0
1	0 0	0 0 0 1
1	0 1	0 0 1 0
1	1 0	0 1 0 0
1	1 1	1 0 0 0



Practice

- Using a decoder and external gates, design the combinational circuit defined by the following three boolean functions:
- $F_1 = x'y'z' + xz = \sum (0, 5, 7)$
- $F_2 = xy'z' + x'y = \sum (2, 3, 4)$
- $F_3 = x'y'z + xy = \sum (1, 6, 7)$

Practice

- Implement the following Boolean function with a 4 X 1 multiplexer and external gates.

  $F(A, B, C, D) = \sum (1, 3, 4, 11, 12, 1\ 3, 14, 15)$

Practice

- Implement the following Boolean function with a 4 X 1 multiplexer and external gates.

  $F(A, B, C, D) = \sum (1, 2, 4, 7, 8, 9, 10, 11, 13, 15)$

Practice

- Construct a 16 X 1 multiplexer with two 8 X 1 and one 2 X 1 multiplexers. Use block diagrams