# Encoders

- An *encoder is a multiplexer without its single output line*

- It is a combinational logic function that has $2^n$ (or fewer) input lines and n output lines,
- The n output lines generate the binary code for the possibl$2^n$ input lines.
- Let us take the case of an octal-to-binary encoder. Such an encoder would have eight input lines,

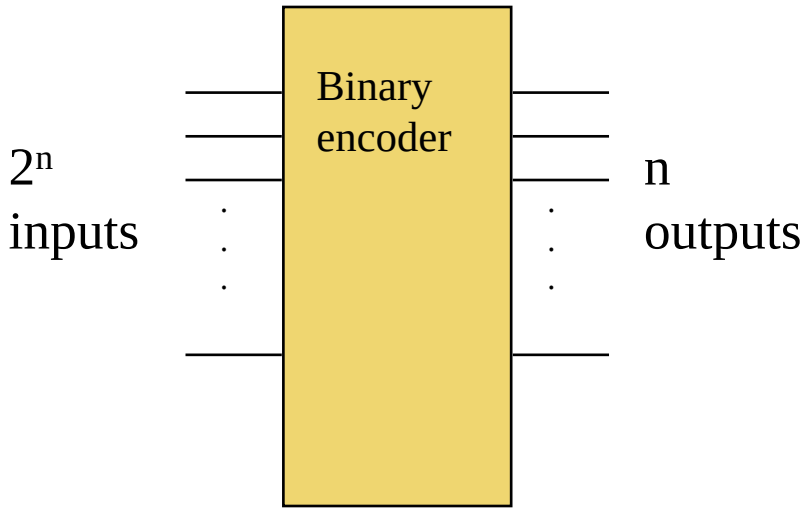
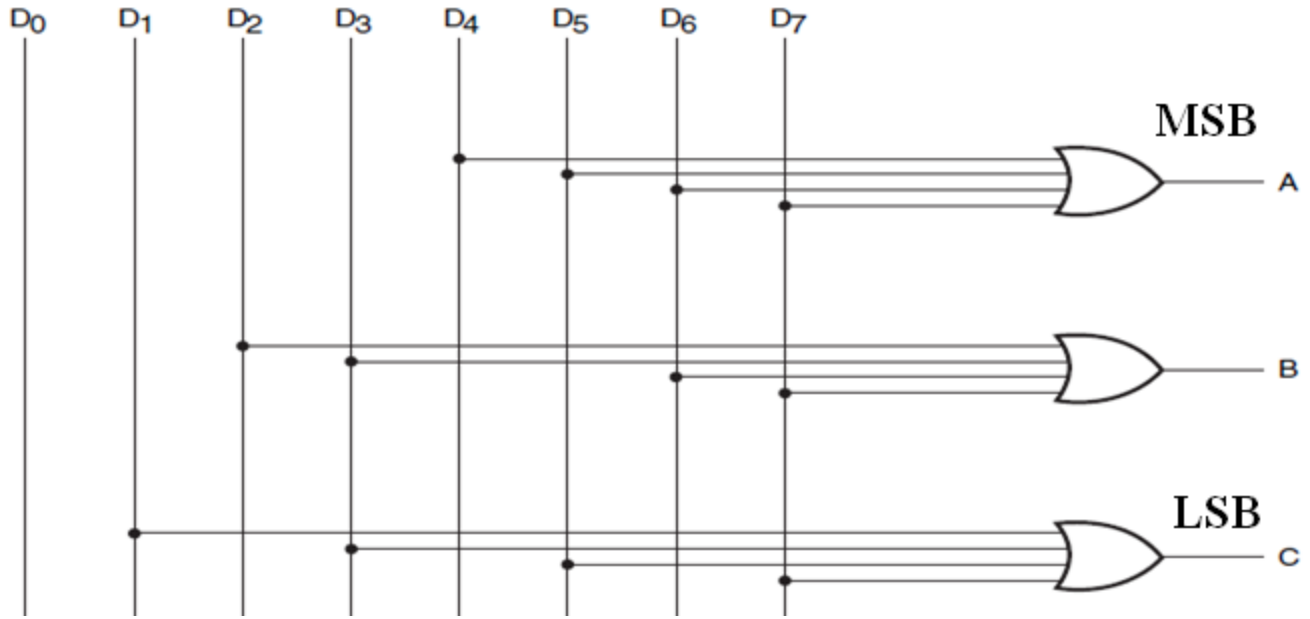


**Table 8.8** Truth table of an encoder.

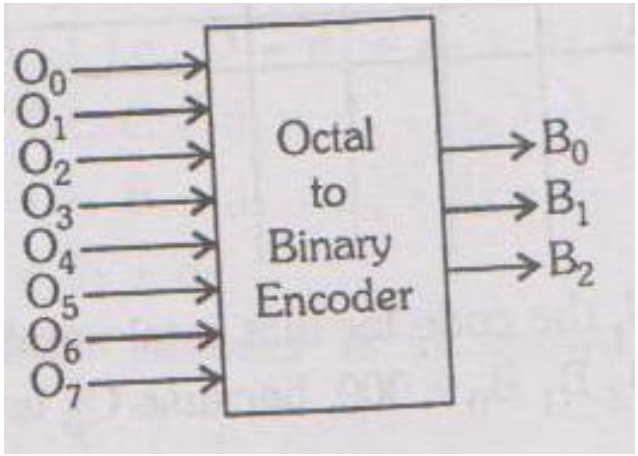
| $D_0$ | $D_1$ | $D_2$ | $D_3$ | $D_4$ | $D_5$ | $D_6$ | $D_7$ | $A$ | $B$ | $C$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 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 |



# Types of Encoder

- Octal to binary Encoder
- Decimal to BCD Encoder
- Priority Encoder

# Octal to binary Encoder




$$B_2 = O_4 + O_5 + O_6 + O_7$$

$$B_1 = O_2 + O_3 + O_6 + O_7$$

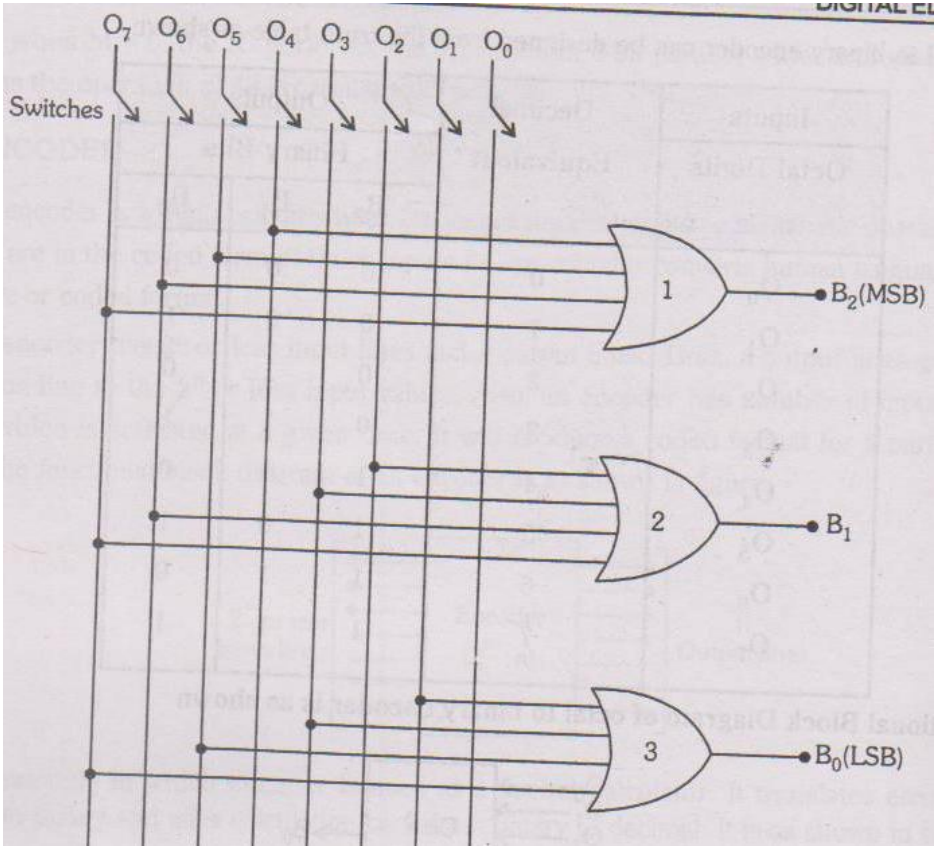
$$B_0 = O_1 + O_3 + O_5 + O_7$$

| Inputs | Decimal Equivalent | Outputs | | |
|---|---|---|---|---|
| Octal Digits | | Binary Bits | | |
| | | $B_2$ | $B_1$ | $B_0$ |
| $O_0$ | 0 | 0 | 0 | 0 |
| $O_1$ | 1 | 0 | 0 | 1 |
| $O_2$ | 2 | 0 | 1 | 0 |
| $O_3$ | 3 | 0 | 1 | 1 |
| $O_4$ | 4 | 1 | 0 | 0 |
| $O_5$ | 5 | 1 | 0 | 1 |
| $O_6$ | 6 | 1 | 1 | 0 |
| $O_7$ | 7 | 1 | 1 | 1 |

$$B_2 = O_4 + O_5 + O_6 + O_7$$

$$B_1 = O_2 + O_3 + O_6 + O_7$$

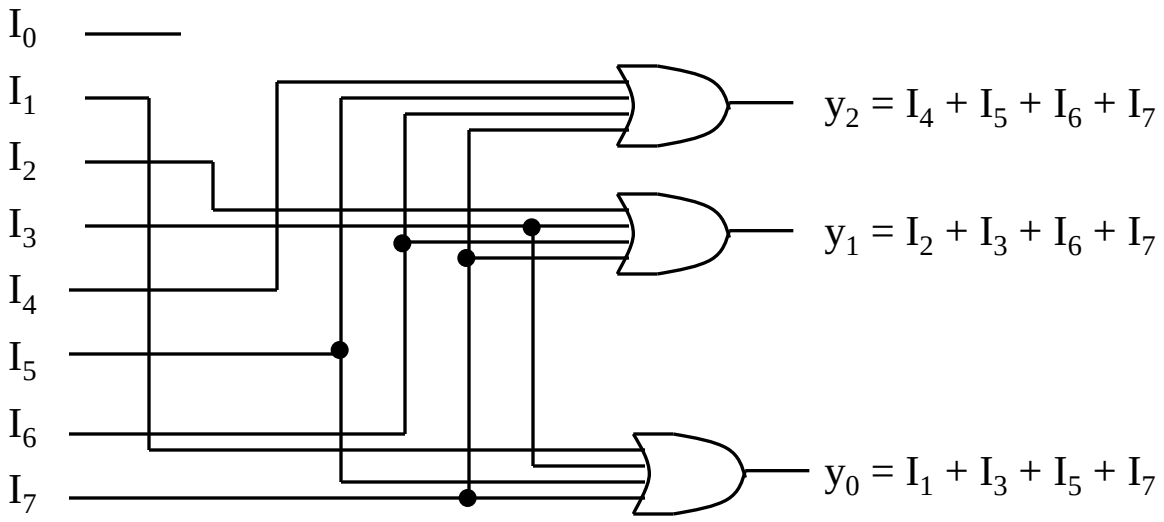
$$B_0 = O_1 + O_3 + O_5 + O_7$$



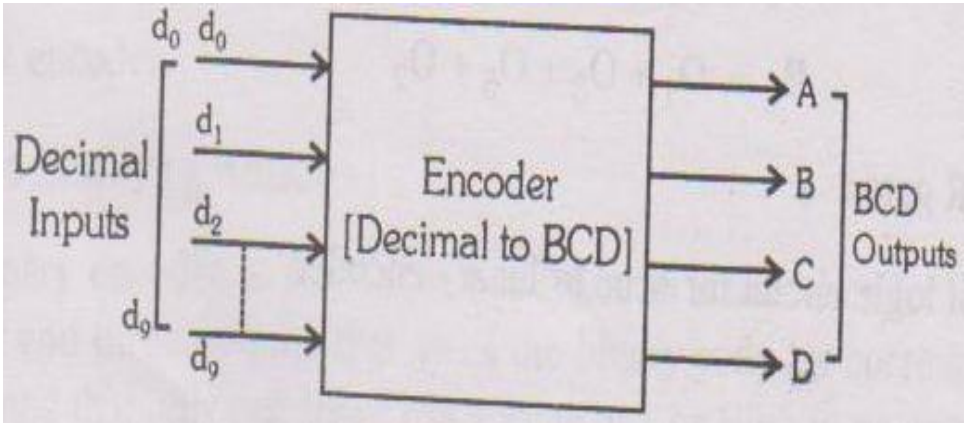
# 8-to-3 Binary Encoder

At any one time, only
one input line has a value of 1.

| Inputs | | | | | | | | Outputs | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $y_2$ | $y_1$ | $y_0$ |
| 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 |



# Decimal to BCD Encoder




| Decimal | Binary | BCD | | | |
|---|---|---|---|---|---|
| | | A | B | C | D |
| 0 | 0000 0000 | 0 | 0 | 0 | 0 |
| 1 | 0000 0001 | 0 | 0 | 0 | 1 |
| 2 | 0000 0010 | 0 | 0 | 1 | 0 |
| 3 | 0000 0011 | 0 | 0 | 1 | 1 |
| 4 | 0000 0100 | 0 | 1 | 0 | 0 |
| 5 | 0000 0101 | 0 | 1 | 0 | 1 |
| 6 | 0000 0110 | 0 | 1 | 1 | 0 |
| 7 | 0000 0111 | 0 | 1 | 1 | 1 |
| 8 | 0000 1000 | 1 | 0 | 0 | 0 |
| 9 | 0000 1001 | 1 | 0 | 0 | 1 |

# 8-to-3 Priority Encoder

- What if more than one input line has a value of 1?
- Ignore "lower priority" inputs.
- Idle indicates that no input is a 1.
- Note that polarity of Idle is opposite from Table 4-8 in Mano

| Inputs | | | | | | | | Outputs | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_0$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ | $I_5$ | $I_6$ | $I_7$ | $y_2$ | $y_1$ | $y_0$ | Idle |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | x | x | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| X | X | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| X | X | X | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| X | X | X | X | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| X | X | X | X | X | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| X | X | X | X | X | X | 1 | 0 | 1 | 1 | 0 | 0 |
| X | X | X | X | X | X | X | 1 | 1 | 1 | 1 | 0 |

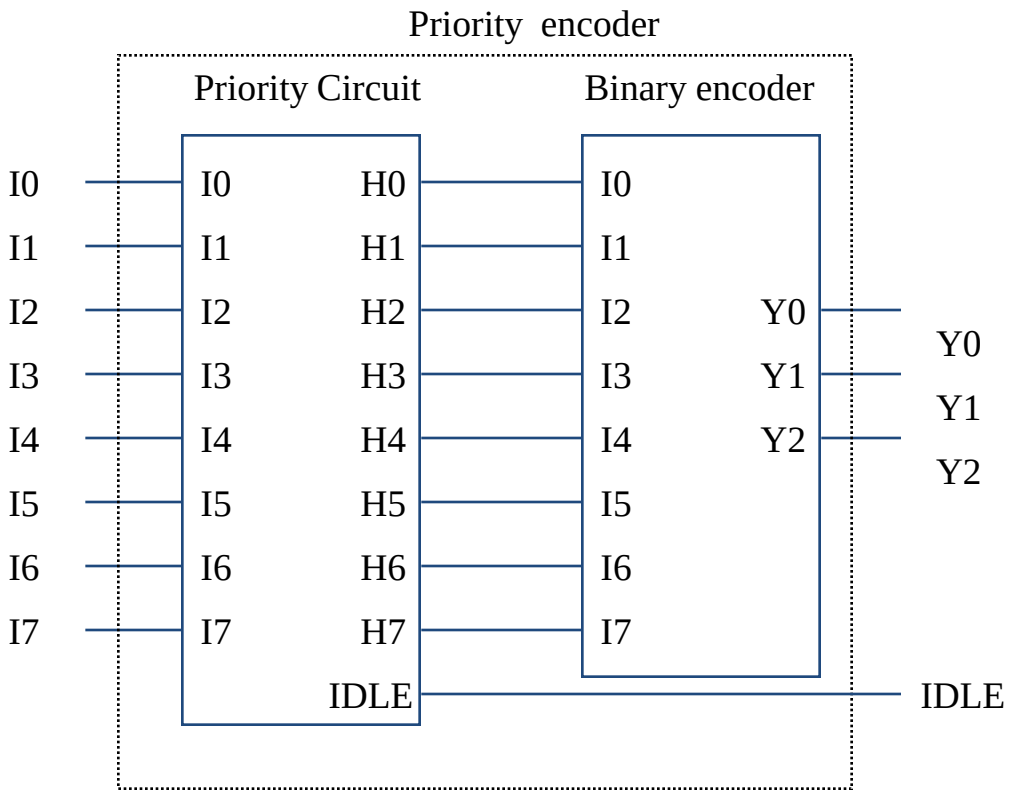
# Priority Encoder (8 to 3 encoder)

- Assign priorities to the inputs
- When more than one input are asserted, the output generates the code of the input with the highest priority
- Priority Encoder :

  H7=I7 (Highest Priority)
  H6=I6.I7'
  H5=I5.I6'.I7'
  H4=I4.I5'.I6'.I7'
  H3=I3.I4'.I5'.I6'.I7'
  H2=I2.I3'.I4'.I5'.I6'.I7'
  H1=I1. I2'.I3'.I4'.I5'.I6'.I7'
  H0=I0.I1'. I2'.I3'.I4'.I5'.I6'.I7'
  IDLE= I0'.I1'. I2'.I3'.I4'.I5'.I6'.I7'

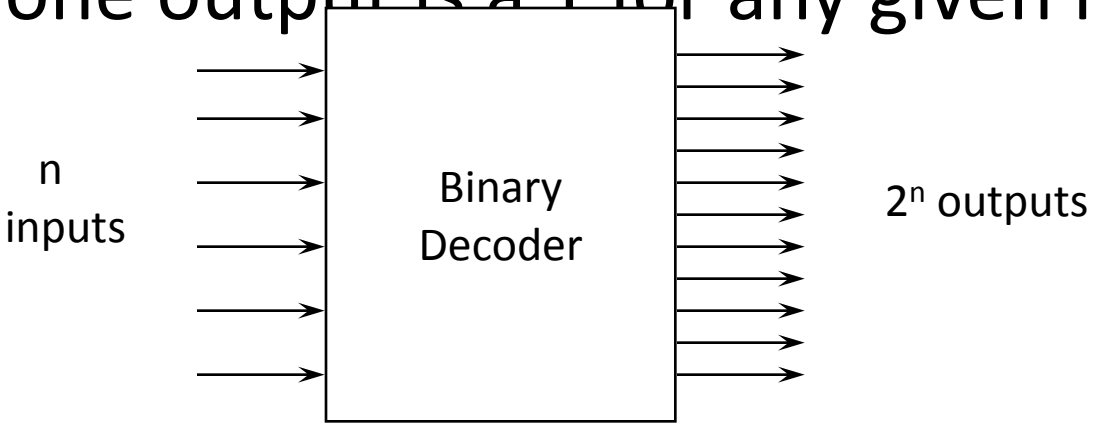
- Encoder

  Y0 = I1 + I3 + I5 + I7
  Y1 = I2 + I3 + I6 + I7
  Y2 = I4 + I5 + I6 + I7



# Binary Decoder

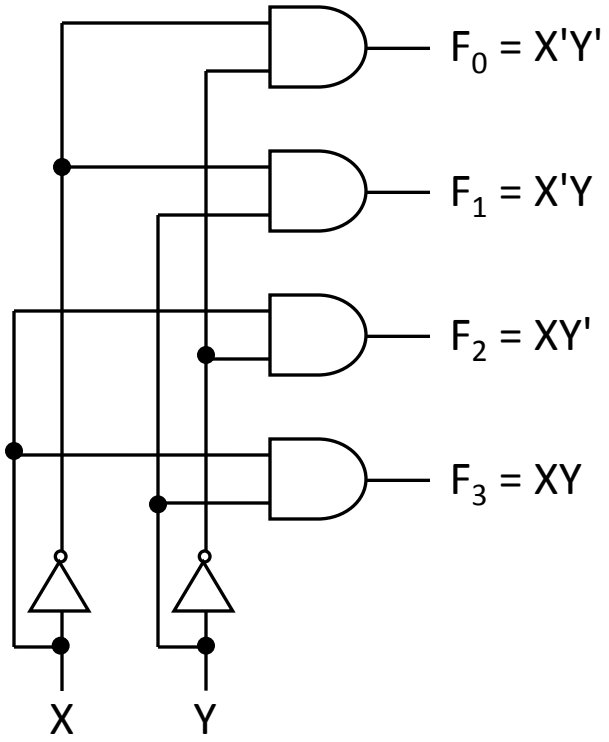
- Black box with n input lines and $2^n$ output lines
- Only one output is a 1 for any given input



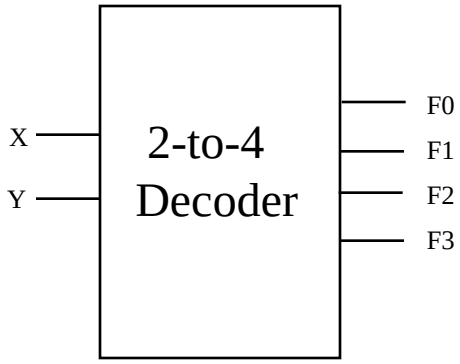
# 2-to-4 Binary Decoder

Truth Table:

| X | Y | $F_0$ | $F_1$ | $F_2$ | $F_3$ |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |



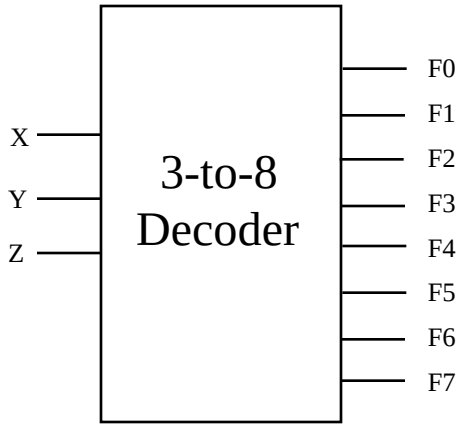

- From truth table, circuit for 2x4 decoder is:
- Note: Each output is a 2-variable minterm (X'Y', X'Y, XY' or XY)

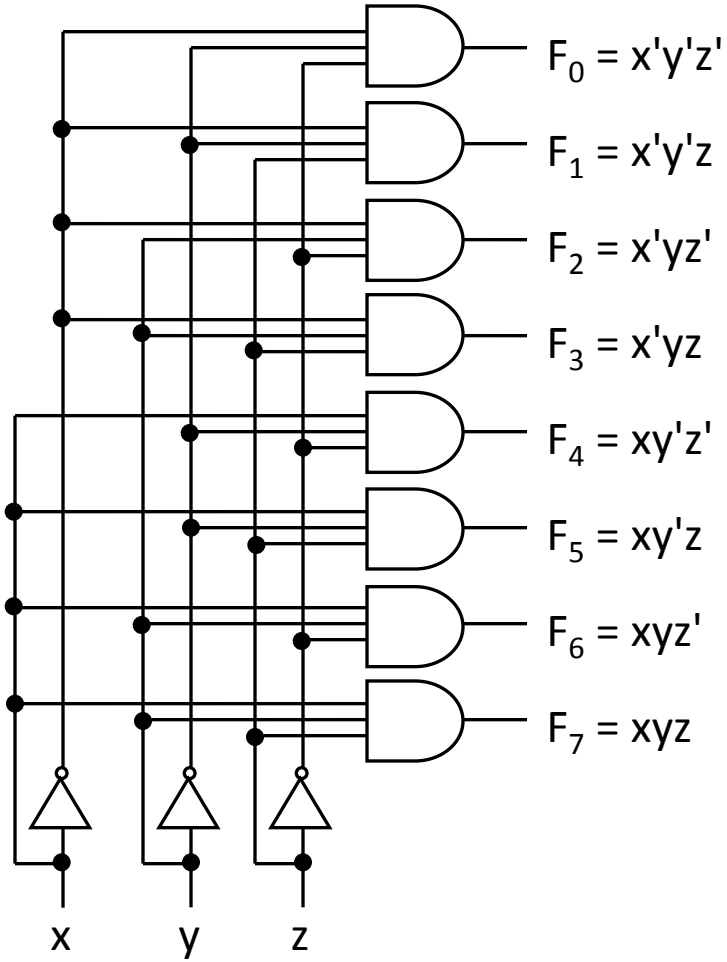


# 3-to-8 Binary Decoder

Truth Table:

| x | y | z | $F_0$ | $F_1$ | $F_2$ | $F_3$ | $F_4$ | $F_5$ | $F_6$ | $F_7$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 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 |



# Implementing Functions Using Decoders

- Any n-variable logic function can be implemented using a single n-to-$2^n$ decoder to generate the minterms
  - OR gate forms the sum.
  - The output lines of the decoder corresponding to the minterms of the function are used as inputs to the or gate.

- Any combinational circuit with *n* inputs and *m* outputs can be implemented with an n-to-$2^n$ decoder with *m* OR gates.
- Suitable when a circuit has many outputs, and each output function is expressed with few minterms.

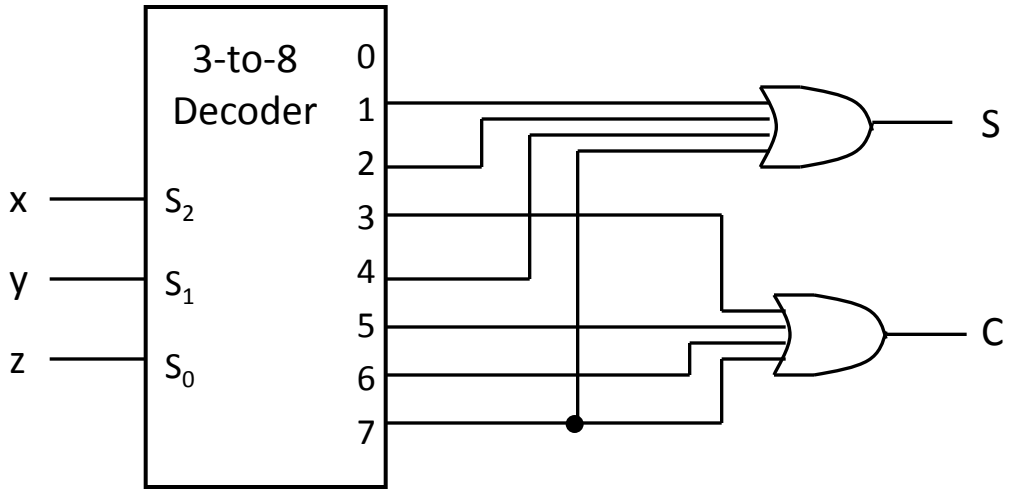
# Implementing Functions Using Decoders

- Example: Full adder

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

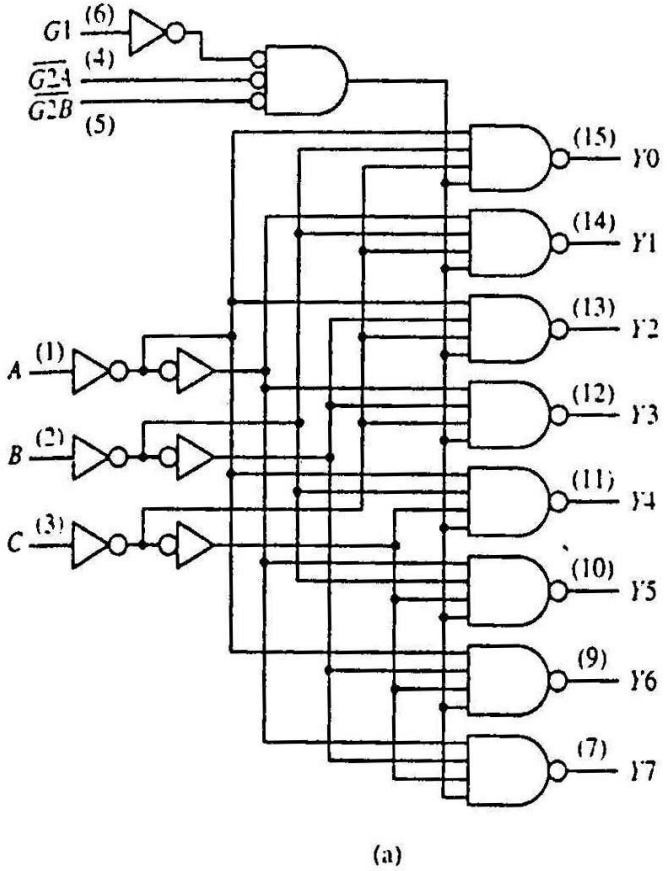
  $C(x, y, z) = \Sigma\,(3,5,6,7)$

| x | y | z | C | S |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

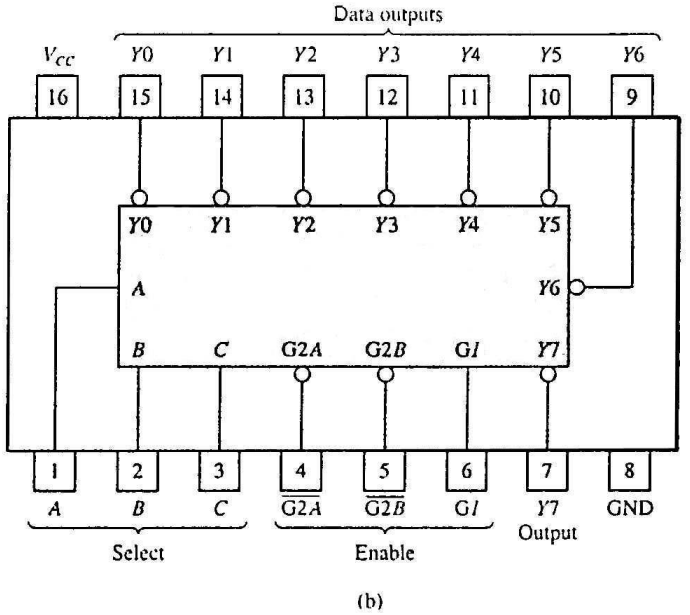


# Standard MSI Binary Decoders Example

## 74138 (3-to-8 decoder)




(a)




(b)

| Inputs | | | | | Outputs | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Enable | | Select | | | | | | | | | | |
| G1 | $\overline{G2^*}$ | C | B | A | Y0 | Y1 | Y2 | Y3 | Y4 | Y5 | Y6 | Y7 |
| H | L | L | L | L | L | H | H | H | H | H | H | H |
| H | L | L | L | H | H | L | H | H | H | H | H | H |
| H | L | L | H | L | H | H | L | H | H | H | H | H |
| H | L | L | H | H | H | H | H | L | H | H | H | H |
| H | L | H | L | L | H | H | H | H | L | H | H | H |
| H | L | H | L | H | H | H | H | H | H | L | H | H |
| H | L | H | H | L | H | H | H | H | H | H | L | H |
| H | L | H | H | H | H | H | H | H | H | H | H | L |
| × | H | × | × | × | H | H | H | H | H | H | H | H |
| L | × | × | × | × | H | H | H | H | H | H | H | H |

$\overline{G2^*} = \overline{G2A} + \overline{G2B}$

(c)

(a) Logic circuit.
(b) Package pin configuration.
(c) Function table.