Digital System Design

Slide - 3

Decoder

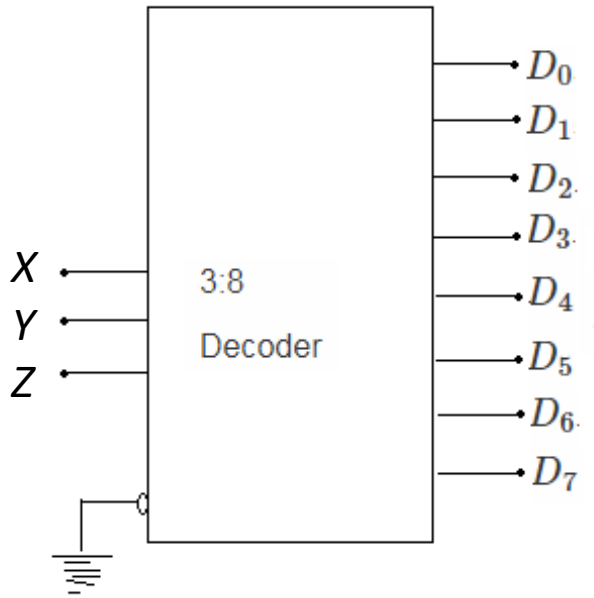
- A combinational circuit that converts binary information from **n** input lines to a maximum of $\mathbf{2^n}$ unique output lines
- Called **n-to-m** line decoders where $\mathbf{m<=2^n}$
- Purpose is to generate the 2^n (or less) minterms of n input variables
- **Application:** Binary to Octal Conversion (Input variables represent a binary number, and output represents eight digits in the octal number system)
- A 3-to-8 line decoder can be used to decode any 3-bit code to provide 8-outputs
- **Example:** 3:8, 2:4, 4:16 Decoder

3-to-8 Decoder

Truth Table:

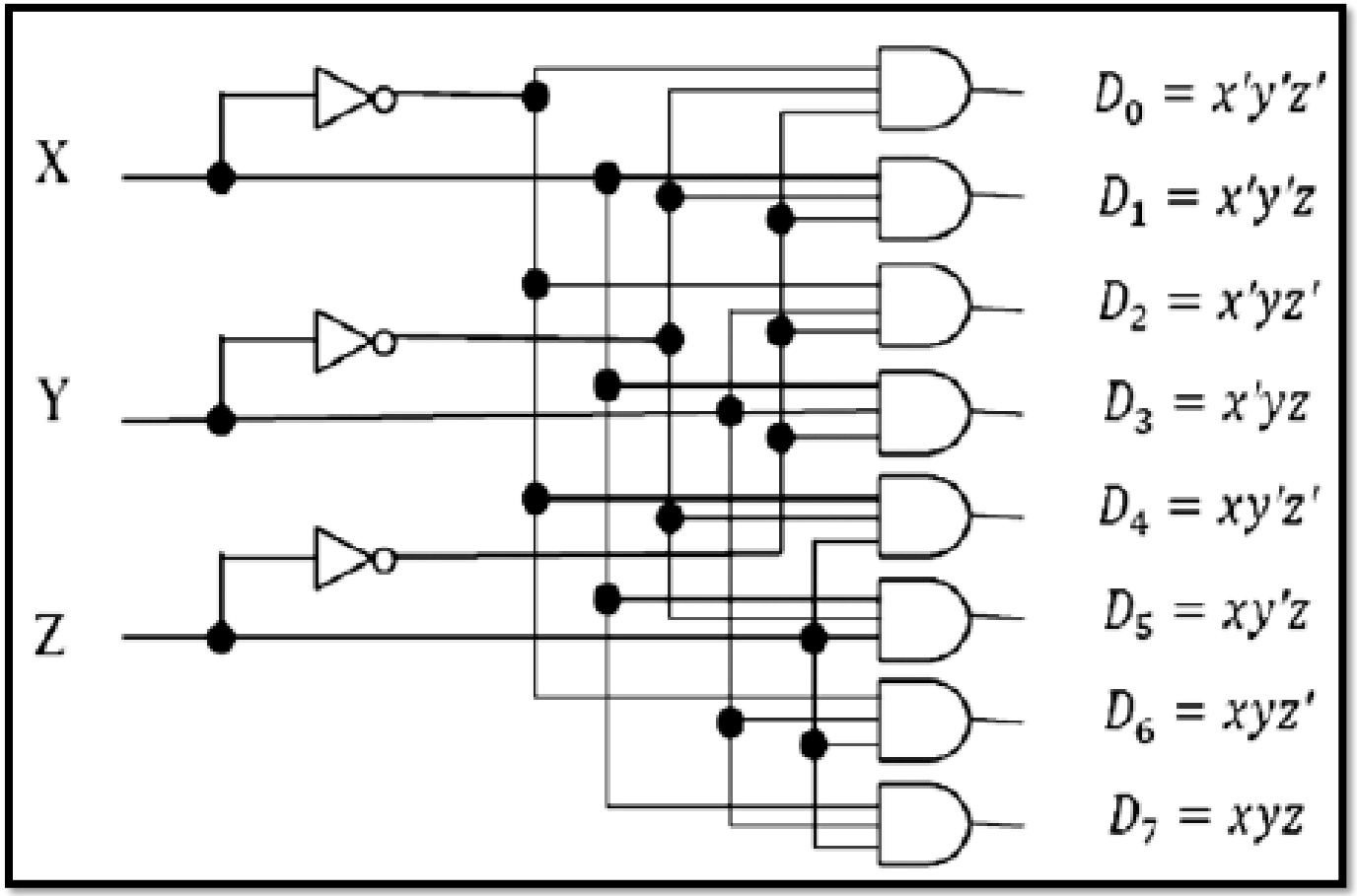
Inputs			Outputs							
X	Y	Z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	**1**	0	0	0	0	0	0	0
0	0	1	0	**1**	0	0	0	0	0	0
0	1	0	0	0	**1**	0	0	0	0	0
0	1	1	0	0	0	**1**	0	0	0	0
1	0	0	0	0	0	0	**1**	0	0	0
1	0	1	0	0	0	0	0	**1**	0	0
1	1	0	0	0	0	0	0	0	**1**	0
1	1	1	0	0	0	0	0	0	0	**1**

Block Diagram:



3-to-8 Decoder (Contd.)

Circuit Diagram:



Practice Work:

Design a BCD-to-decimal decoder with the help of don't care conditions.

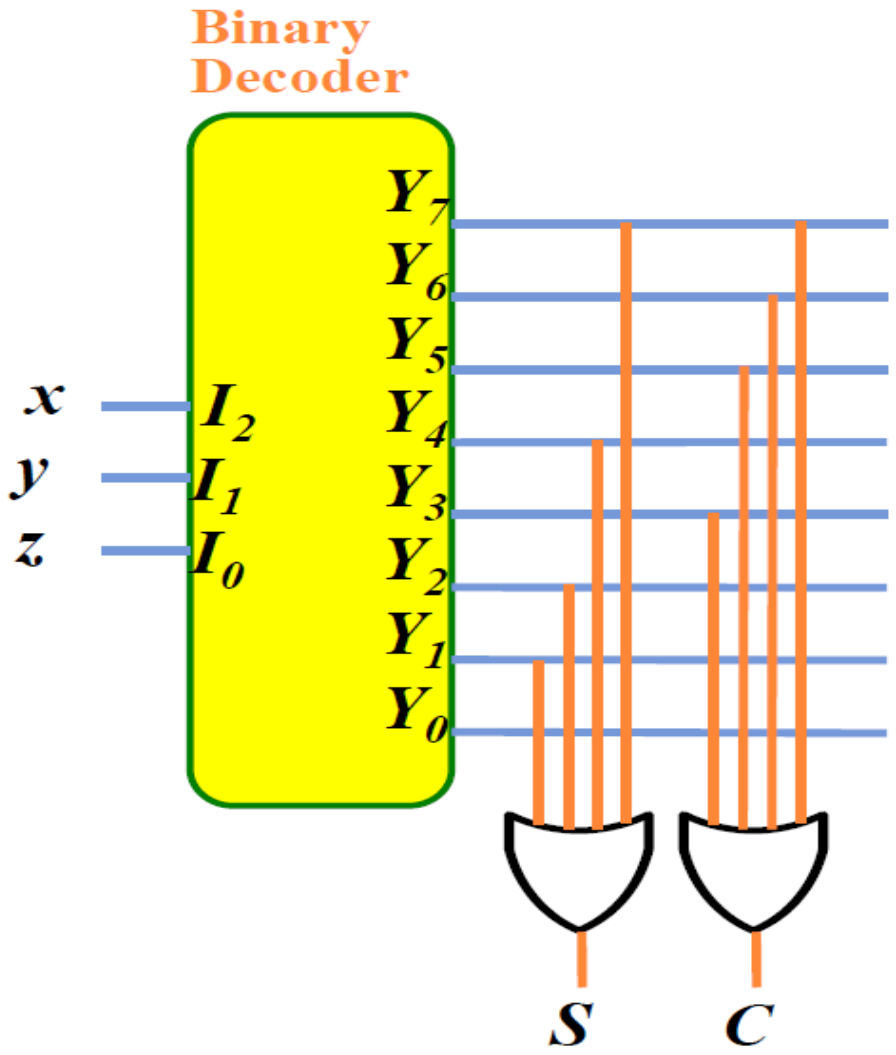
Implementation Using Decoder

Input			Output	
x	y	z	S	C
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Example: Full Adder

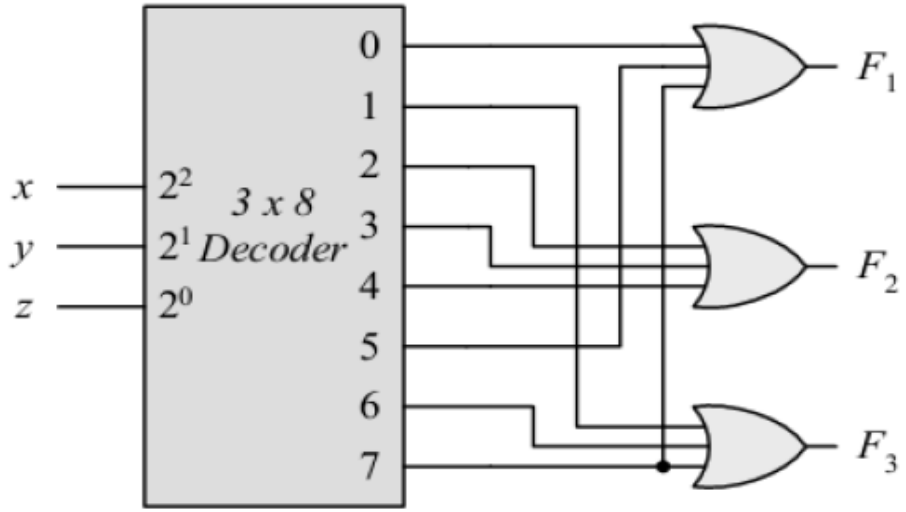
$S(x, y, z) = \sum(1, 2, 4, 7)$

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



Implementation Using Decoder (Contd.)

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



Encoder

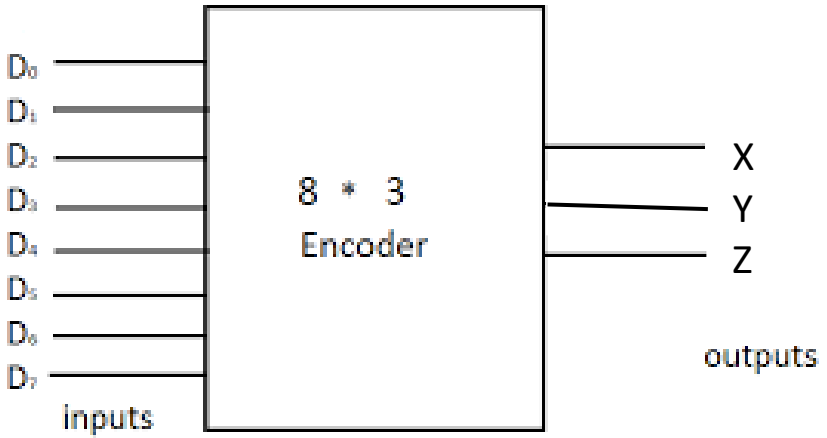
- A combinational circuit that performs the reverse operation of Decoder
- It encodes $\mathbf{2^n}$ input lines to **n** output lines
- Produces a binary code equivalent to the input
- Called **m-to-n** line encoders where $\mathbf{m=2^n}$
- Only one of the inputs become "high" (logic state "1") at a time.
- **Application:** Octal to Binary Conversion
- **Example:** 4:2, 8:3, 16:4 Encoder

8-to-3 Encoder

Truth Table:

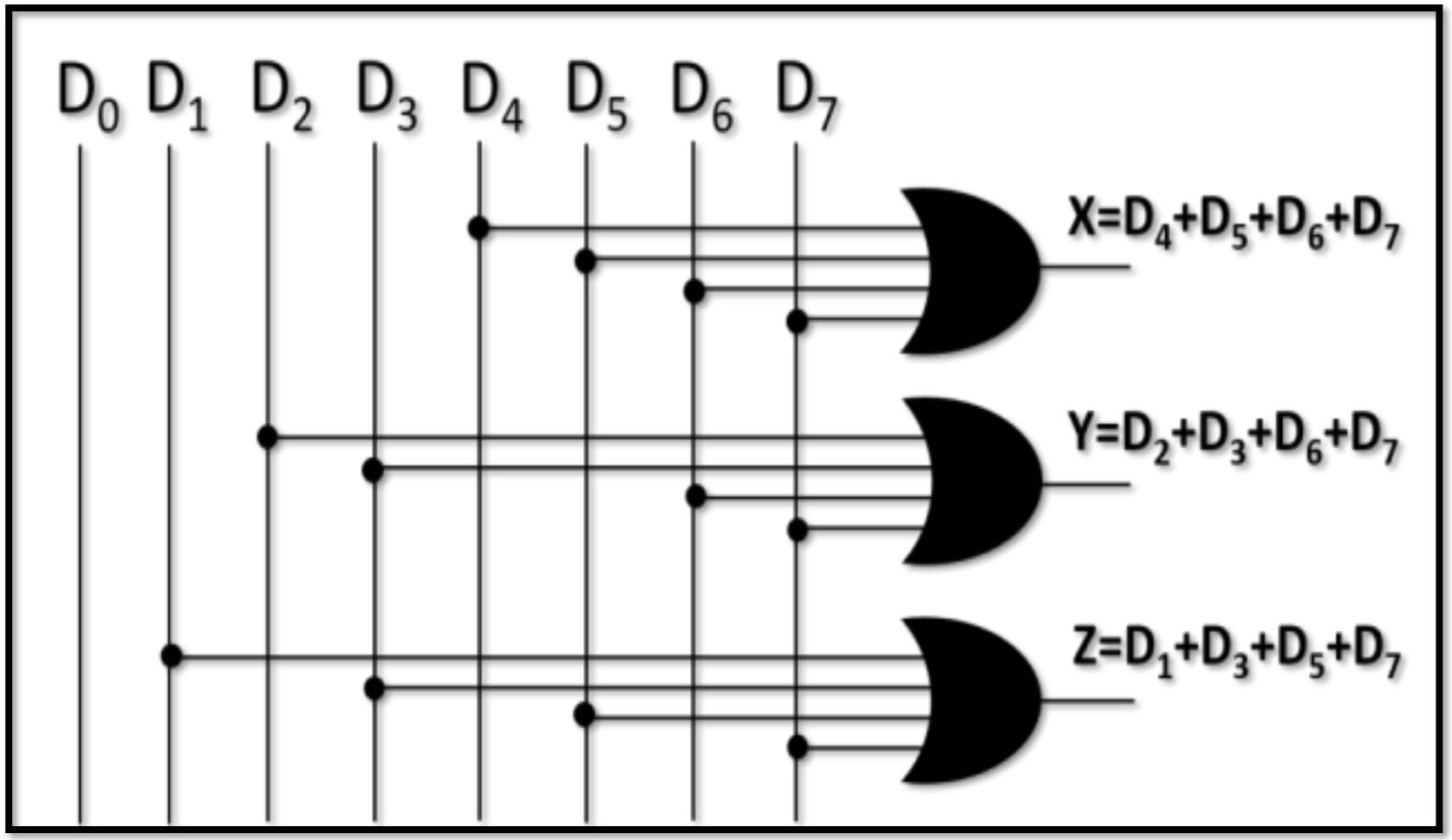
Input								Output		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Block Diagram:



8-to-3 Encoder (Contd.)

Circuit Diagram:



Practice Work:
Design a Decimal-to-BCD Encoder.

Priority Encoder

- Encoder with priority function
- Multiple inputs may be true simultaneously
- Higher priority input gets the precedence
- Even if more than one input is ‘1’ at the same time, the output will be the (binary) code corresponding to the input, which is having higher priority

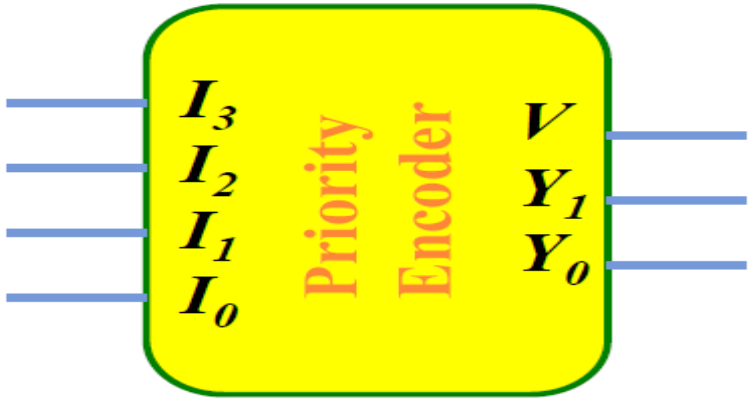
Priority Encoder

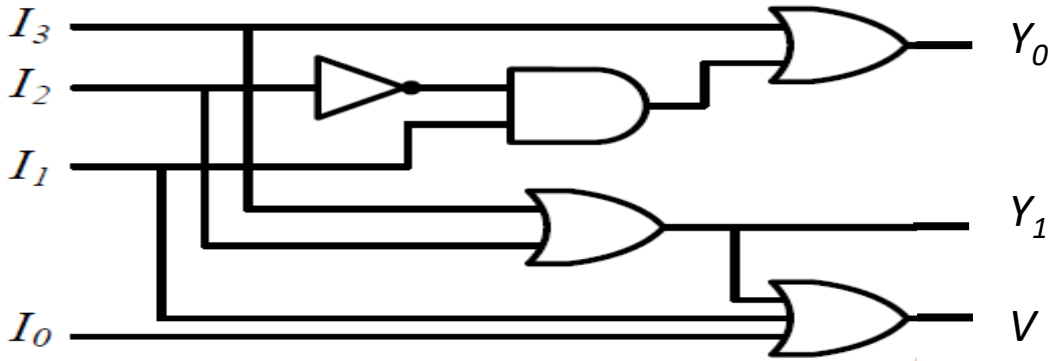
- 4-Input Priority Encoder

Priority (high to low)

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

$$Y_1 = I_3 + I_2$$
$$Y_0 = I_3 + \bar{I}_2 I_1$$
$$V = I_3 + I_2 + I_1 + I_0$$





Practice Problems

I. A combinational circuit is specified by the following three Boolean functions:

$F1(A, B, C) = \Sigma(2, 4, 7)$

$F2(A, B, C) = \Sigma(0, 3)$

$F3(A, B, C) = \Sigma(0, 2, 3, 4, 7)$

Implement the circuit with a decoder constructed with NAND/AND gates. Minimize the number of inputs in the external gates.

II. Implement a full adder with a decoder and NAND gates.

III. Design a combinational circuit that compares two 4-bit numbers to check if they are equal. The circuit output is equal to 1 if two numbers are equal and 0 otherwise. Mention the equation of the circuit.