



# Combinational Logic

*Rab Nawaz Khan Jadoon*

Lecturer

**COMSATS Lahore**

**Pakistan**

Department of Computer Science

**DCS**

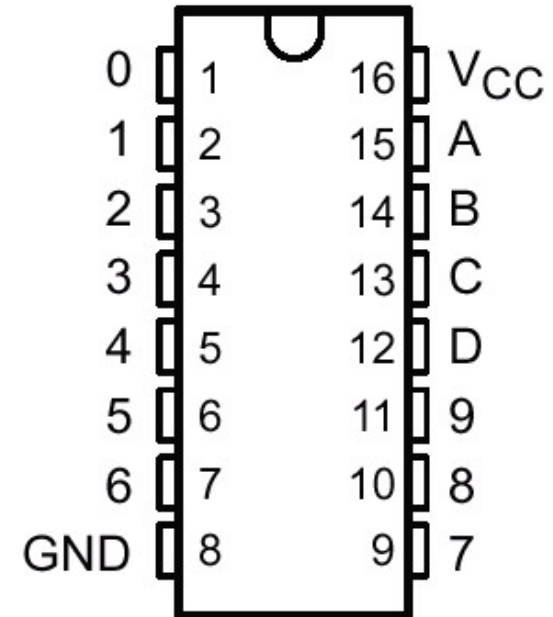
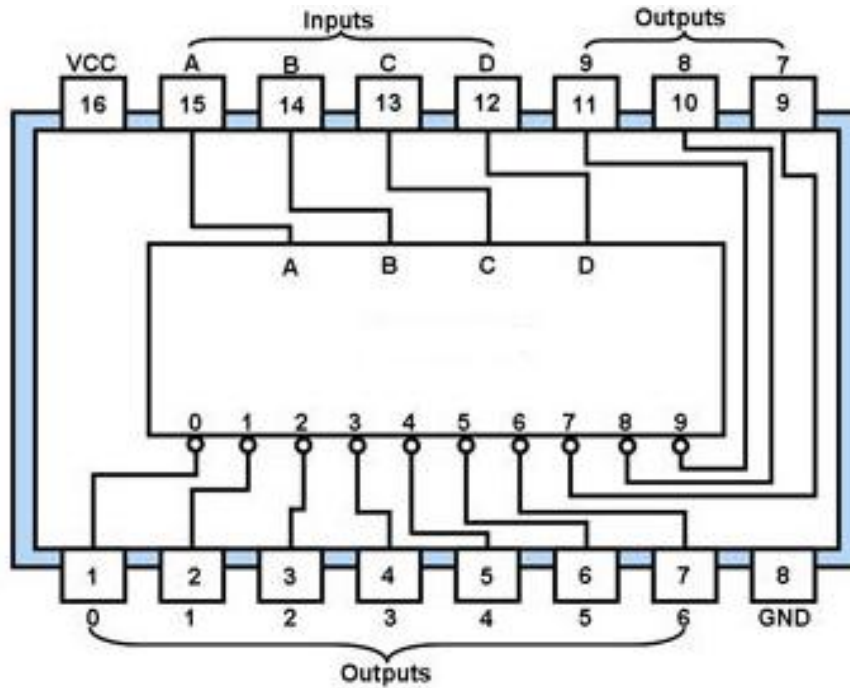
COMSATS Institute of  
Information Technology

**Digital Logic and Computer Design**

## ■ Design Procedures

- Starts from the verbal outline of the problem and ends in a logic circuit diagram.
- The procedure involves the following step,
  - The problem is stated.
  - Input and required output variables are determined.
  - Assigned the variables letter symbols.
  - Make the truth table.
  - The simplified Boolean functions for each output is obtained.
  - The logic diagram is drawn.

# Decoder

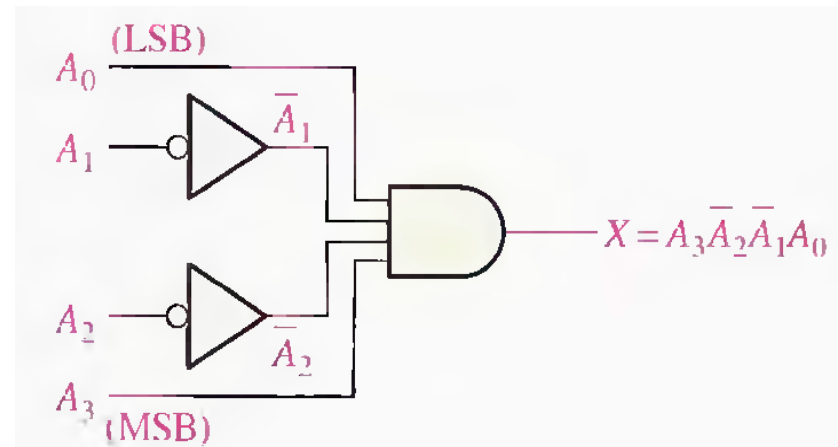
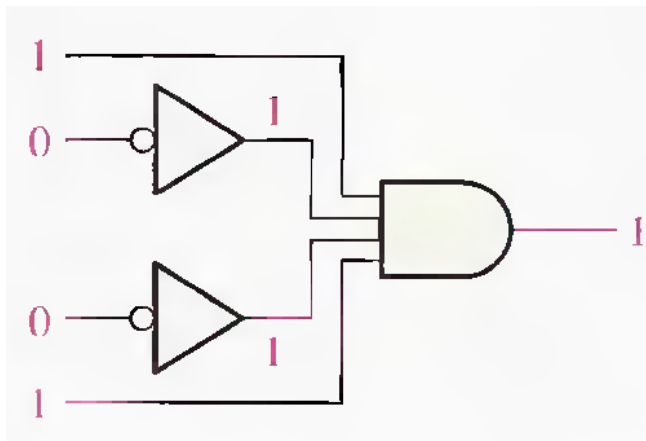


- A decoder is a digital circuit that detects the presence of a specified combination of bits (code) on its inputs and indicates the presence of that code by a specified output level.
  - A **decoder** has  $n$  input lines to handle  $n$  bits and from one to  $2^n$  output lines to indicate the presence of one or more  $n$ -bit combinations. Or
  - It is a combinational circuit that converts binary information from  $n$  input lines to a maximum of  $2^n$  unique output lines
  - Suppose you need to determine when a binary 1001 occurs on the inputs of a digital circuit.

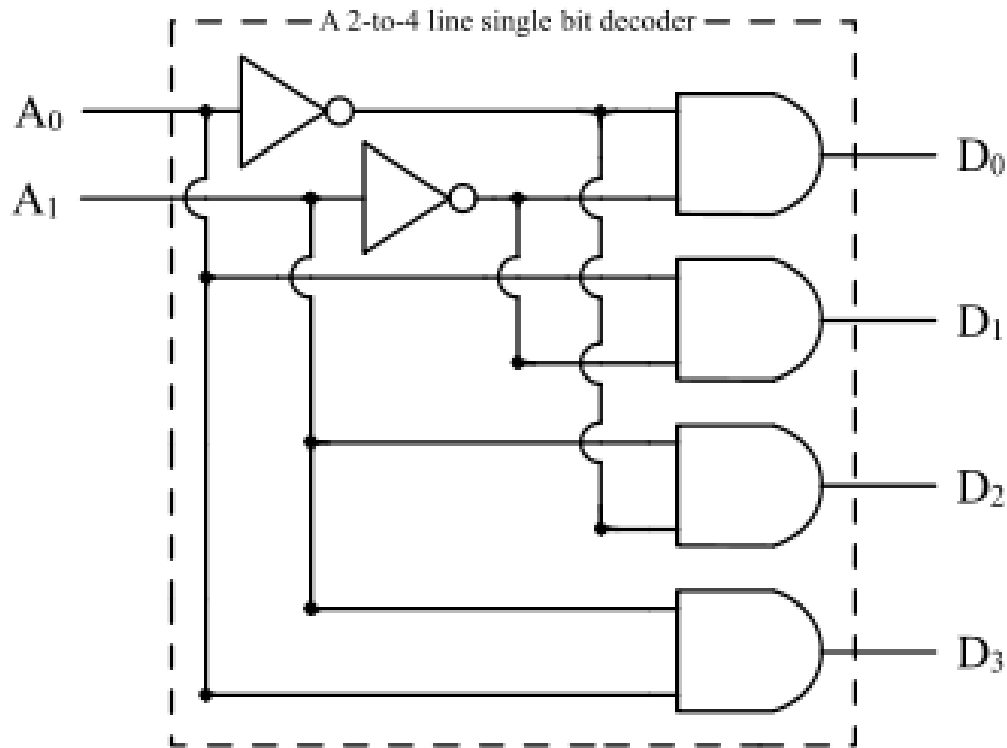
- An AND gate can be used as the basic decoding element because it produces a HIGH output only when all of its inputs are HIGH.
- **Decoding** is necessary in applications such as data multiplexing, 7 segment display and memory address decoding.
- A **decoder** that contains enable inputs is also known as a decoder-demultiplexer.
- A **decoder** is a device which does the reverse of an encoder, undoing the encoding so that the original information can be retrieved.

# Decoder

- Therefore, you must make sure that all of the inputs to the AND gate are HIGH when the binary number 1001 occurs.
- this can be done by inverting the two middle bits (the 0s), as shown in Figure,



# 2 to 4 line single bit decoder



Truth Table

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

Minterm Equations

$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

$$D_3 = A_1 \cdot A_0$$

## ■ The 4-Bit Decoder

- In order to decode all possible combinations of four bits, sixteen decoding gates are required ( $2^4 = 16$ ).
- This type of decoder is commonly called either a 4-line-to-16-line decoder because there are four inputs and sixteen outputs.
- **1-of-16 decoder** because for any given code on the inputs, one of the sixteen outputs is activated.
- A list of the sixteen binary codes and their corresponding decoding functions is given in Table on next slide.





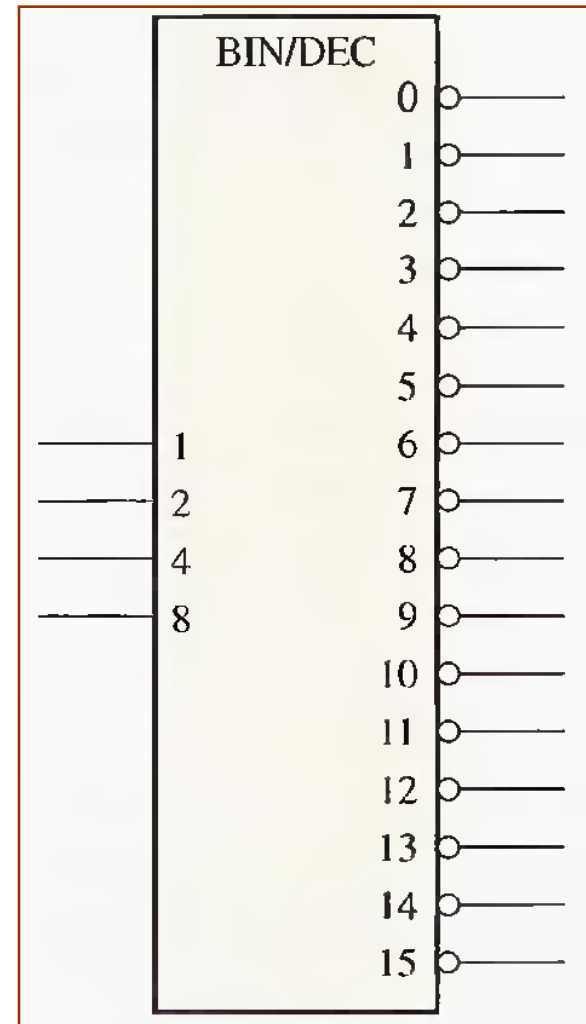
# Decoder

DECIMAL DIGIT	BINARY INPUTS				DECODING FUNCTION	OUTPUTS																
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	0	0	0	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	$\bar{A}_3\bar{A}_2A_1A_0$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	$\bar{A}_3A_2\bar{A}_1A_0$	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	$\bar{A}_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
7	0	1	1	1	$\bar{A}_3A_2A_1A_0$	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
8	1	0	0	0	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
9	1	0	0	1	$A_3\bar{A}_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
10	1	0	1	0	$A_3\bar{A}_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
11	1	0	1	1	$A_3\bar{A}_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
12	1	1	0	0	$A_3A_2\bar{A}_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
13	1	1	0	1	$A_3A_2\bar{A}_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
14	1	1	1	0	$A_3A_2A_1\bar{A}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
15	1	1	1	1	$A_3A_2A_1A_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0



# Decoder

- A logic symbol for a 4-line-to-16-line (1-of-16) decoder.
  - The BIN/DEC label indicates that a binary input makes the corresponding decimal output active.
  - The input labels 8, 4, 2, and 1 represent the binary weights of the input bits ( $2^3$   $2^2$   $2^1$   $2^0$ ).
  - Decoder element if AND,
    - Output is Active High output
  - Decoder element, if NAND
    - Output is active low output

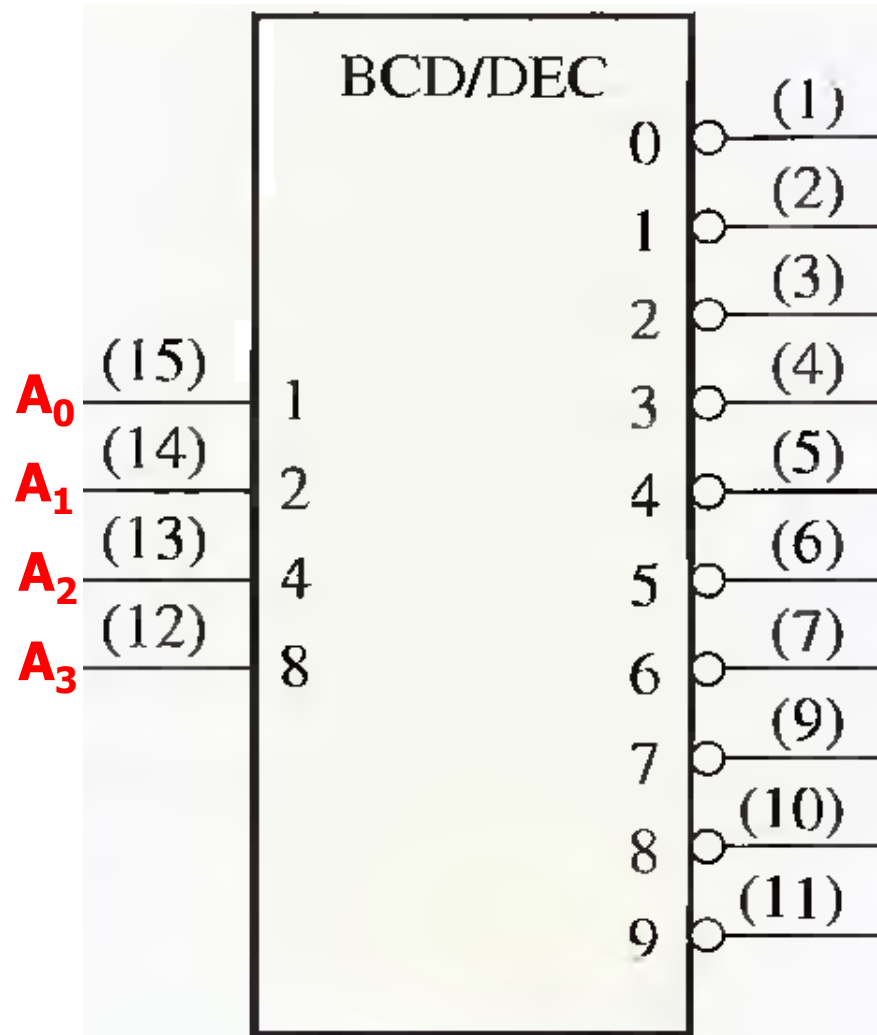


- **The BCD-to-Decimal Decoder**
  - The BCD-to-decimal decoder converts each BCD code (8421 code) into one of ten possible decimal digit indications.
  - Only ten decoding gates are required because the BCD code represents only the ten decimal digits 0 through 9.
  - A list of the ten BCD codes and their corresponding decoding functions is given in Table on the next slide,

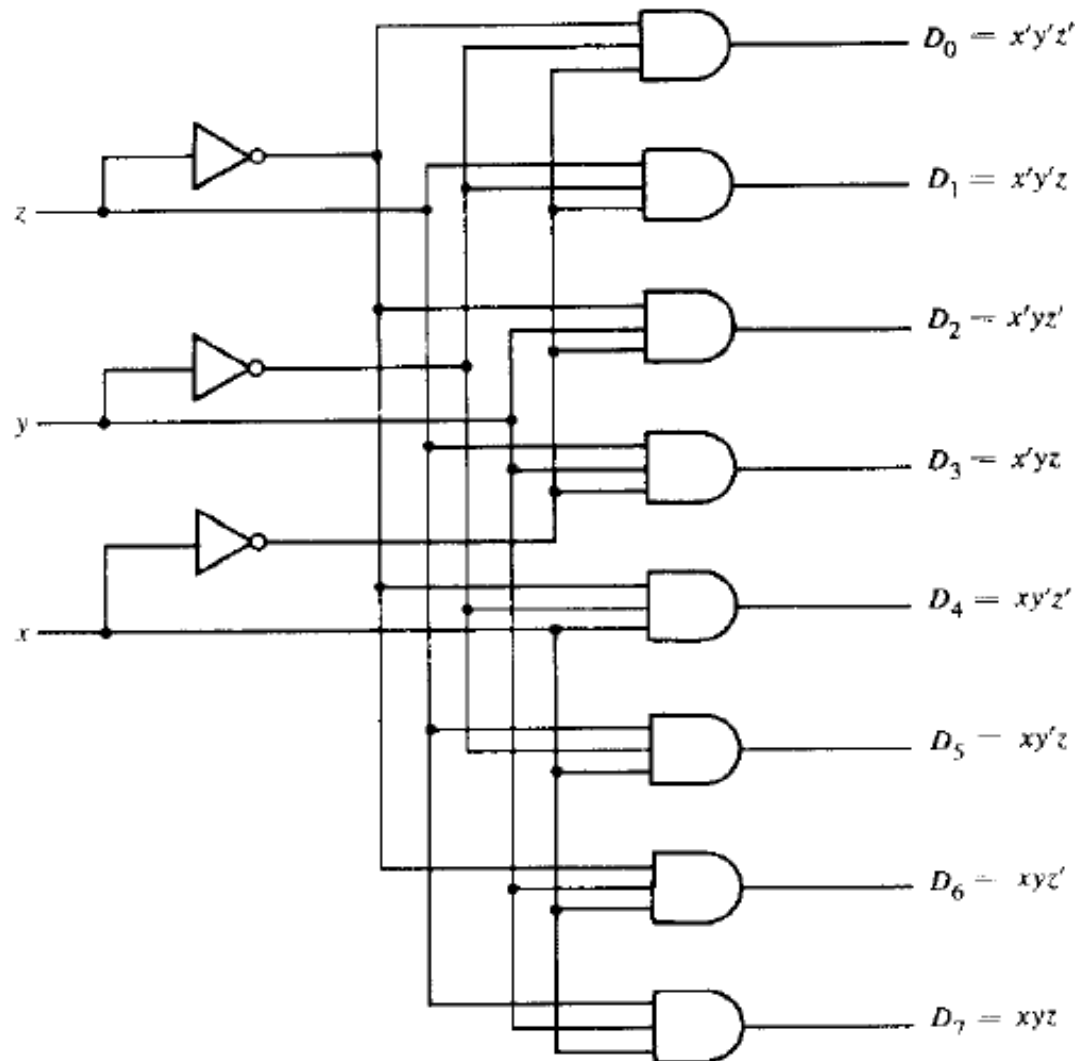
# Decoder

DECIMAL DIGIT	BCD CODE				DECODING FUNCTION
	$A_3$	$A_2$	$A_1$	$A_0$	
0	0	0	0	0	$\bar{A}_3\bar{A}_2\bar{A}_1\bar{A}_0$
1	0	0	0	1	$\bar{A}_3\bar{A}_2\bar{A}_1A_0$
2	0	0	1	0	$\bar{A}_3\bar{A}_2A_1\bar{A}_0$
3	0	0	1	1	$\bar{A}_3\bar{A}_2A_1A_0$
4	0	1	0	0	$\bar{A}_3A_2\bar{A}_1\bar{A}_0$
5	0	1	0	1	$\bar{A}_3A_2\bar{A}_1A_0$
6	0	1	1	0	$\bar{A}_3A_2A_1\bar{A}_0$
7	0	1	1	1	$\bar{A}_3A_2A_1A_0$
8	1	0	0	0	$A_3\bar{A}_2\bar{A}_1\bar{A}_0$
9	1	0	0	1	$A_3\bar{A}_2\bar{A}_1A_0$

# Decoder



# 3 to 8 line Decoder



# Decoder

**Truth Table of a 3-to-8-Line Decoder**

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

# Combinational Logic Implementation

- The procedure for implementing a combinational circuit by means of a decoder and OR Gates requires that the boolean functions for the circuit be expressed in SOP.
  - This form can easily be obtained from the truth table.
  - A decoder is then chosen that generates all the minterms of the  $n$  input variables.
  - The inputs to each OR gate are selected from the decoder outputs according to the minterm list in each function.





## ■ **Problem:**

- Implement a full adder circuit with a decoder and two OR gates.

## ■ **Solution:**

- From the truth table of full adder,

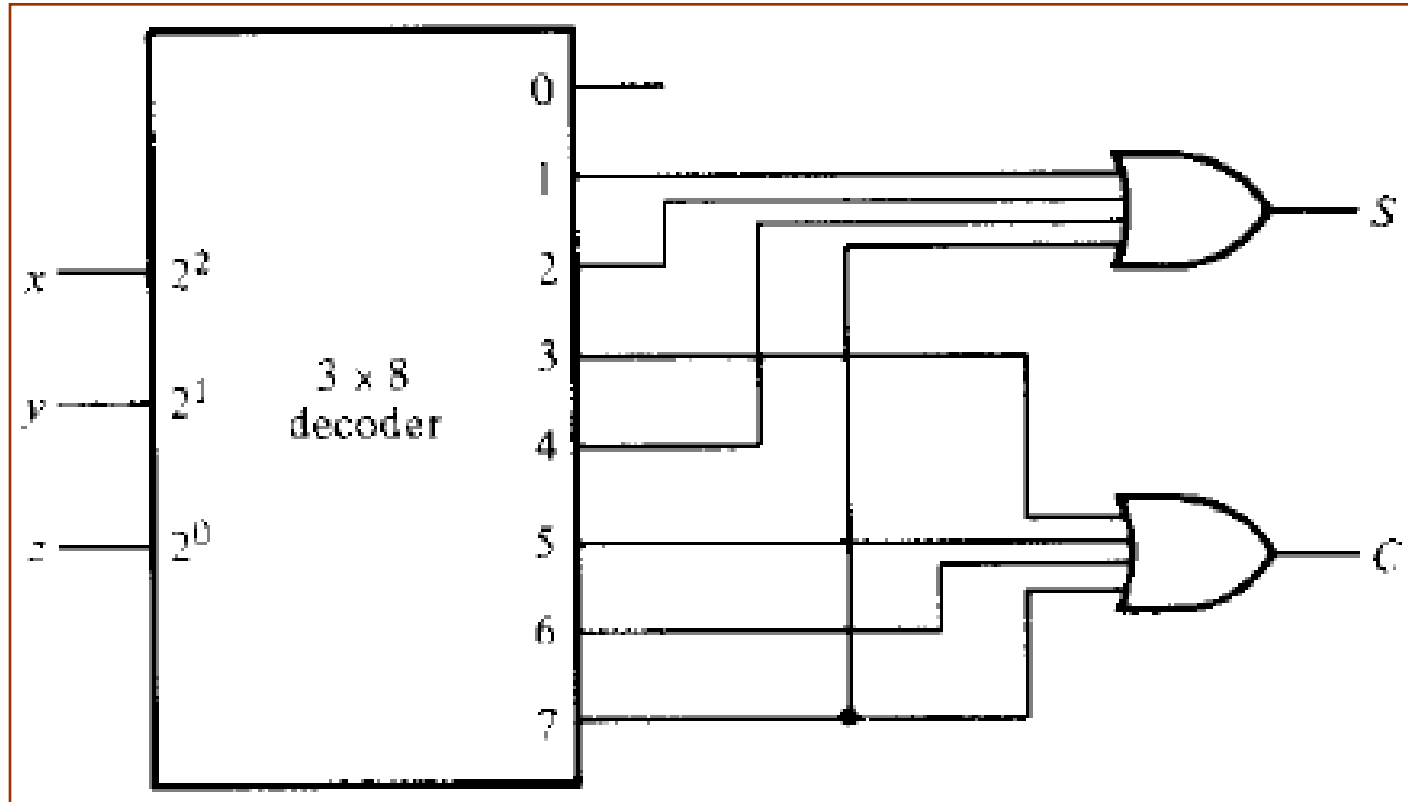
$$S(x, y, z) = \Sigma(1, 2, 4, 7)$$

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

- Since there are three inputs and a total of eight minterms, we need a 3 to 8 line decoder.



# Implementation of a full adder with a decoder

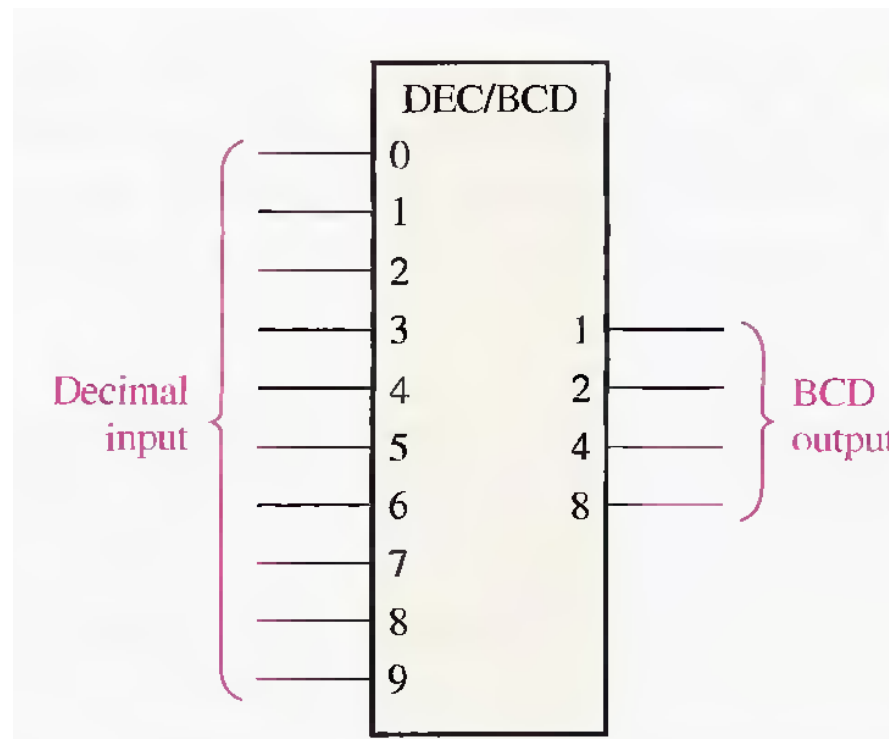


The decoder method can be used to implement any combination Circuit. However its implementation must be compared with all others Possible implementation to determine the best solution.

- An encoder is a combinational logic circuit that essentially performs a "reverse" decoder.
  - An encoder accepts an active level on one of its inputs representing a digit, such as a decimal or octal digit, and converts it to a coded output, such as BCD or binary.

# The Decimal-to-BCD Encoder

- This type of encoder has ten inputs-one for each decimal digit-and four outputs corresponding to the BCD code, as shown in Figure below,



# The BCD (8421) code

DECIMAL DIGIT	BCD CODE			
	$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

# The Decimal-to-BCD Encoder

- From this table you can determine the relationship between each BCD bit and the decimal digits in order to analyze the logic.
  - For instance, the most significant bit of the BCD code,  $A_3$ , is always a 1 for decimal digit 8 or 9.
  - An OR expression for bit  $A_3$  in terms of the decimal digits can therefore be written as,
    - $A_3 = 8 + 9$
  - Bit  $A_2$  is always a 1 for decimal digit 4, 5, 6 or 7 and can be expressed as an OR function as follows:
    - $A_2 = 4 + 5 + 6 + 7$

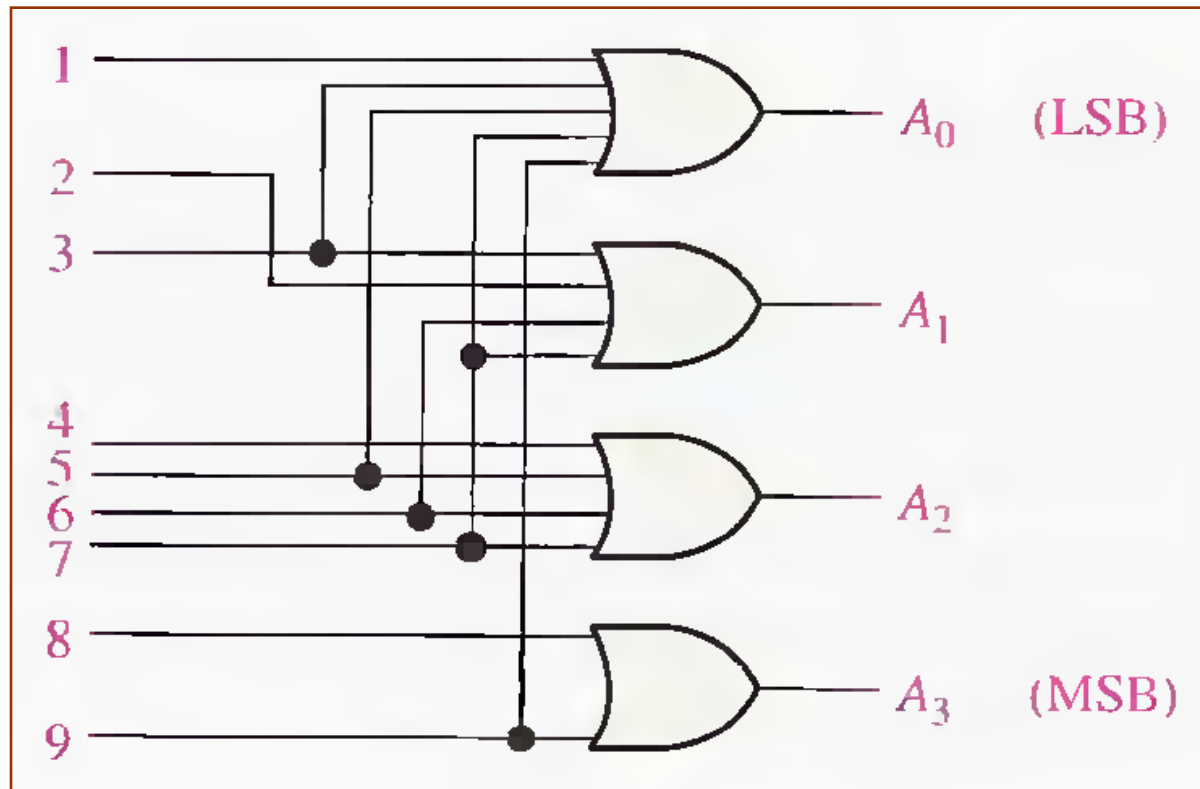


# The Decimal-to-BCD Encoder

- Bit  $A_1$  is always a 1 for decimal digit 2, 3, 6, or 7 and can be expressed as
- $A_1 = 2 + 3 + 6 + 7$
- Finally,  $A_0$  is always a 1 for decimal digit 1, 3, 5, 7, or 9. The expression for  $A_0$  is
- $A_0 = 1 + 3 + 5 + 7 + 9$
- Now let's implement the logic circuitry required for encoding each decimal digit to a BCD code by using the logic expressions just developed.
- It is simply a matter of ORing the appropriate decimal digit input lines to form each BCD output.



# The Decimal-to-BCD Encoder





# The Decimal-to-BCD Encoder

- When a HIGH appears on one of the decimal digit input lines, the appropriate levels occur on the four BCD output lines.
  - For instance, if input line 9 is HIGH (assuming all other input lines are LOW), this condition will produce a HIGH on outputs  $A_0$  and  $A_3$  and LOWs on outputs  $A_1$  and  $A_2$ , which is the BCD code (1001) for decimal 9.



