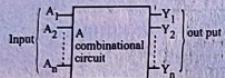


## COMBINATIONAL LOGIC

### Combinational Circuit

A combinational circuit may be defined as a logic circuit the output of which depends only upon the combination of the present inputs. The output does not depend on the past value of inputs or outputs. Therefore, a combinational circuit does not need any memory. Combinational circuits are faster in operation than sequential circuits.



*Fig.: Block diagram of a combinational circuit.*

Examples of combinational circuit are half adder, full adder, half subtractor, full subtractor, encoder, decoder, multiplexer, demultiplexer, etc.

### 5.1 Design Procedure

The steps followed in designing a combinational circuit are as follows:

1. We will be given a problem.
2. Then, we determine the number of inputs and output and assign letter symbol to input and outputs.
3. After that we write a truth table relating the input and output.
4. Then, we write K-map for each output and obtain the simplified Boolean expression for each output.
5. Lastly, we draw the logic diagram of combinational circuit.

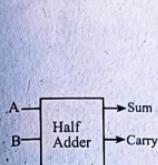
### 5.2 Adders and Subtractors

#### 1. Half Adder (HA)

Half adder is a combinational logic circuit with two inputs and two outputs. It performs the addition of two 1-bit numbers and produces the output sum and carry.

Two inputs = A, B

Two outputs = Sum, Carry



*Fig.: Block diagram*

For sum,	
B	0 1
A	0 0 1 1 1 0

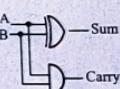
$$\text{Sum} = \overline{A}\overline{B} + A\overline{B} \\ = A \oplus B$$

For carry,	
B	0 1
A	0 0 0 1 0 1

$$\text{Carry} = AB$$

*Truth table*

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

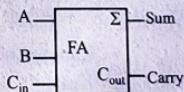


#### Drawback of half adder:

To perform the addition of three bits is not possible by using a half adder circuit.

#### 2. Full Adder (FA)

Full adder circuit overcomes the drawback of a half adder circuit. A combinational circuit that performs the addition of three bits (two significant bits and a previous carry) is called a full adder. The name of the circuit stems from the fact that two half adders can be employed to implement a full adder.



*Fig.: Block diagram of a full adder*

Truth table

Input			Output	
A	B	C <sub>in</sub>	Sum	Carry
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

For sum,

A	BC <sub>in</sub>			
	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$\begin{aligned}
 \text{Sum} &= \bar{A} \bar{B} C_{in} + \bar{A} B \bar{C}_{in} + A \bar{B} C_{in} + A B C_{in} \\
 &= \bar{A} (\bar{B} C_{in} + B \bar{C}_{in}) + A (\bar{B} C_{in} + B C_{in}) \\
 &= \bar{A} (B \oplus C_{in}) + A (B \oplus \bar{C}_{in}) \\
 &= \bar{A} X + A \bar{X} \quad \text{where } X = B \oplus C_{in} \\
 &= A \oplus X \\
 &= A \oplus (B \oplus C_{in}) \\
 &= A \oplus B \oplus C_{in}
 \end{aligned}$$

For carry,

A	BC <sub>in</sub>			
	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$\text{Carry} = BC_{in} + AC_{in} + AB$$

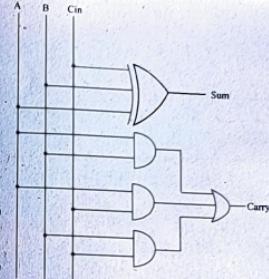


Fig.: Logic diagram of a full adder.

## Construction of Full Adder using Half Adder

Half Adder: Sum = A  $\oplus$  B, Carry = ABFull Adder: Sum = A  $\oplus$  B  $\oplus$  C<sub>in</sub>, Carry = AB + AC<sub>in</sub> + BC<sub>in</sub>

From K-map,

$$\text{Carry} = \bar{A}BC_{in} + \bar{A}\bar{B}C_{in} + ABC_{in} + AB\bar{C}_{in}$$

$$= C_{in}(\bar{A}B + AB) + AB(C_{in} + \bar{C}_{in})$$

$$= C_{in} (A \oplus B) + AB$$

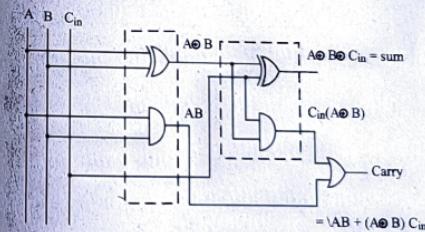


Fig.: Logic diagram of a full adder using half adder.

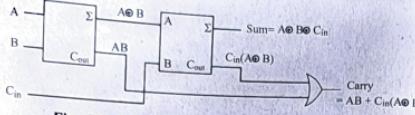


Fig.: Block diagram of a full adder using half adder.

### 3. Half Subtractor (HS)

A half subtractor is a combinational circuit that subtracts two bits and produce their difference. It also has an output to specify if a 1 has been borrowed.

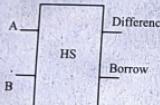


Fig.: Block diagram of a half subtractor

Input		Output	
A	B	Difference (D)	Borrow (B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

For Difference,

B	0	1
A	0 0	1
	1 1	0

$$\text{Diff. } (D) = \bar{A}B + \bar{B}A \\ = A \oplus B$$

For Borrow,

B	0	1
A	0 0	1
	1 0	0

$$\text{Borrow} = \bar{A}B$$

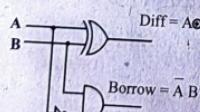


Fig.: Logic diagram of a half subtractor.

### 4) Full Subtractor

A full subtractor is a combinational circuit that performs a subtraction between two bits, taking into account that a 1 may have been borrowed by a lower significant stage. This circuit has three inputs and two outputs.

$$\begin{aligned} \text{Diff} &= A - B - C = A - (B + C) \\ &= A + \bar{B} + \bar{C} + 1 + 1 \end{aligned}$$

If  $A < (B + C)$ , Borrow = 1

If  $A > (B + C)$ , Borrow = 0

Truth table

Input			Output	
A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

For difference,

BC	00	01	11	10
A	0 0	1	0 1	1
	1 1	0	1 0	0

$$\text{Difference} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$= \bar{A}(\bar{B}C + BC) + A(\bar{B}\bar{C} + BC)$$

$$= \bar{A}(B \oplus C) + A(\bar{B} \oplus C)$$

$$= A \oplus (B \oplus C)$$

$$= A \oplus B \oplus C$$

For borrow,

	BC	00	01	11	10
A	0	0	1	1	1
0	0	0	1	1	0
1	0	0	1	0	0

$$\text{Borrow} = \bar{A}C + \bar{A}\bar{B} + BC$$

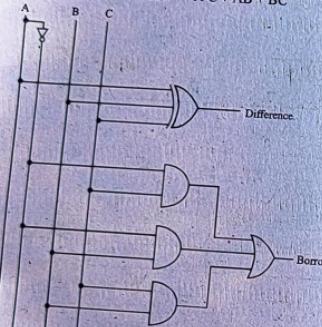


Fig.: Logic diagram of a full subtractor

#### Construction of Full Subtractor using Half Subtractor

**Half subtractor:** Difference =  $A \oplus B$ , Borrow =  $\bar{A}B$

**Full subtractor:** Difference =  $A \oplus B \oplus C$ , Borrow =  $\bar{A}C + \bar{A}\bar{B} + BC$

From K-map,

$$\text{Borrow} = \bar{A}BC + \bar{A}BC + \bar{A}BC + ABC$$

$$= \bar{A}B(C + \bar{C}) + C(\bar{A}B + AB)$$

$$= \bar{A}B + C(A \oplus B)$$

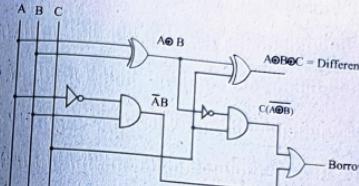


Fig.: Logic diagram of a full subtractor using half subtractor.

#### Design of Half Adder and Half Subtractor in a Single Circuit

When  $C = 0$ , circuit performs addition

When  $C = 1$ , circuit performs subtraction

Inputs	Output1 ( $Y_{out1}$ )		Output2 ( $Y_{out2}$ )	
	A	B	sum/ diff	carry/ borrow
0 0	0	0	0	0
0 1	1	0	1	0
1 0	0	1	1	0
1 1	1	1	0	1

For  $Y_{out1}$ ,

	AB	00	01	11	10
C	0	0	1	0	1
1	0	1	1	0	1

$$Y_{out1} = AB + A\bar{B}$$

$$= A \oplus B$$

	AB	00	01	11	10
C	0	0	0	1	0
1	0	1	0	0	0

$$Y_{out2} = \bar{C}AB + C\bar{A}B$$

$$= B(CA + CA)$$

$$= B(C \oplus A)$$

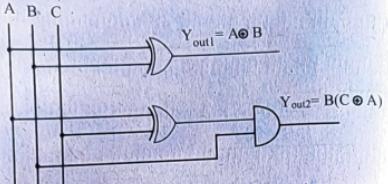


Fig.: Half adder/half subtractor in a single circuit.

#### Design of Full Adder/Full Subtractor in a Single Circuit

Select (S) = Control input

When S = 0, circuit performs addition

When S = 1, circuit performs subtraction

Input				Output	
S	A	B	C	$Y_{sum/diff}$	$Y_{carry/borrow}$
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	1
1	1	1	1	1	1

For  $Y_{sum/diff}$ ,

SA	BC		00		01		11		10	
	0	1	0	1	0	1	0	1	0	1
00	0	1	1	0	1	0	1	0	1	0
01	1	0	0	1	0	1	0	1	0	0
11	1	0	0	1	1	0	1	0	0	0
10	0	1	1	0	0	1	0	1	1	0

$$Y_{sum/diff} = A \cdot B \cdot C + \bar{A} \cdot BC + ABC + \bar{A} \cdot \bar{B} \cdot C$$

$$= \bar{B} \cdot (\bar{A} \cdot \bar{C} + AC) + B(AC + \bar{A} \cdot C)$$

$$= \bar{B} \cdot (A \oplus C) + B \cdot (\bar{A} \oplus C)$$

$$= B \oplus (A \oplus C)$$

$$= A \oplus B \oplus C$$

For  $Y_{carry/borrow}$ ,

SA	BC		00		01		11		10	
	0	1	0	1	0	1	1	0	0	1
00	0	0	0	0	1	0	1	0	1	0
01	0	1	1	1	0	1	1	1	1	1
11	0	0	1	0	1	0	0	1	0	0
10	0	1	1	0	1	1	0	1	1	1

$$Y_{carry/borrow} = BC + SAC + SAB + SAC + SAB$$

$$= BC + SAC + SAC + SAB + SAB$$

$$= BC + C(SA + \bar{S}A) + B(SA + \bar{S}A)$$

$$= BC + C(A \oplus S) + B(S \oplus A)$$

$$= BC + (A \oplus S)(B + C)$$

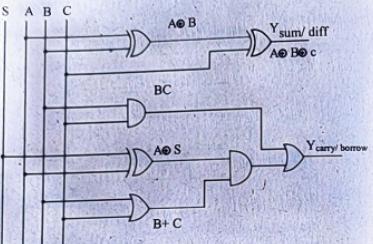


Fig.: Full adder/full subtractor in a single circuit.

### 5.3 Code Conversion

#### 1) 3-bit Binary to Gray Code Converter

No. of inputs = 3 ( $B_2, B_1, B_0$ )

No. of outputs = 3 ( $G_2, G_1, G_0$ )

Inputs			Outputs			
$B_2$	$B_1$	$B_0$	$G_2$	$G_1$	$G_0$	
0	0	0	0	0	0	0
0	0	1	0	0	1	1
0	1	0	0	1	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	0	1	1	1	1	1
1	1	0	1	0	1	1
1	1	1	1	0	0	0

For  $G_2$

$B_2$	00	01	11	10
0	0	0	0	0
1	1	1	1	1

$$G_2 = B_2$$

For  $G_1$

$B_2$	00	01	11	10
0	0	0	(1)	0
1	(1)	0	0	0

$$G_1 = B_2 B_1 + B_2 \bar{B}_1 = B_2 \oplus B_1$$

For  $G_0$

$B_2$	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$G_0 = B_1 B_0 + B_2 \bar{B}_1 = B_1 \oplus B_0$$

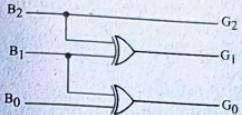


Fig.: Circuit diagram of 3-bit binary to gray code converter.

#### Binary to Excess-3 Code Converter

No. of outputs = 4 ( $E_3, E_2, E_1, E_0$ )

No. of inputs = 4 ( $B_3, B_2, B_1, B_0$ )

$B_3$	$B_2$	$B_1$	$B_0$	$E_3$	$E_2$	$E_1$	$E_0$
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	1	0	1	0
0	0	1	1	1	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	1	0	1
1	0	0	0	1	1	1	0
1	0	0	1	1	1	1	1
1	0	1	0	x	x	x	0
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

For  $E_3$

$B_3$	$B_2$	$B_1$	$B_0$
00	0	0	0
01	0	1	1
11	x	x	x
10	1	1	x

$$E_3 = B_3 + B_2 B_0 + B_2 B_1 = B_3 + B_2 (B_0 + B_1)$$

For $E_2$	
$B_3B_2$	$00 \quad 01 \quad 11 \quad 10$
00	0 1 1 1
01	1 0 0 0
11	x x x x
10	0 x x x

For $E_1$	
$B_3B_2$	$00 \quad 01 \quad 11 \quad 10$
00	1 0 1 0
01	1 0 1 0
11	x x x x
10	1 0 x x

For $E_0$	
$B_3B_2$	$00 \quad 01 \quad 11 \quad 10$
00	1 0 0 1
01	0 0 0 1
11	x x x x
10	1 0 x x

$$\begin{aligned} E_2 &= B_2 \bar{B}_1 \bar{B}_0 + \bar{B}_2 B_0 + \bar{B}_2 B_1 \\ &= B_2 \bar{B}_1 \bar{B}_0 + B_2 (B_0 + B_1) \\ &= B_2 \bar{B}_1 \bar{B}_0 + B_2 (B_0 + B_1) \end{aligned}$$

$$E_1 = \bar{B}_1 \bar{B}_0 + B_1 B_0$$

$$E_0 = \bar{B}_0$$

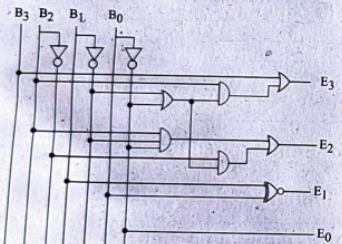
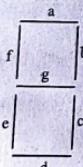


Fig.: Circuit diagram of binary to excess-3 code converter

### Seven Segment Decoder



A LED emits radiation when forward biased, free electrons recombine with holes near the junction. As the free electrons fall from higher energy level to lower one, they give up energy in the form of energy and light.

forward biasing different LEDs, we can display the digits 0 through 9. A seven segment decoder driver is an IC decoder that can be used to drive a seven segment indicator.

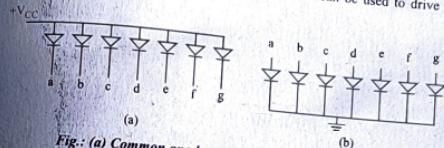


Fig.: (a) Common anode type (b) Common cathode type  
Truth table for common cathode type

Inputs	Outputs										
	A	B	C	D	a	b	c	d	e	f	g
0 0	0	0	0	0	1	1	1	1	1	1	0
1 0	0	0	0	1	0	1	1	0	0	0	0
2 0	0	0	1	0	1	1	0	1	1	0	1
3 0	0	0	1	1	1	1	0	1	1	0	1
4 0	1	0	0	0	1	1	1	1	0	0	1
5 0	1	0	0	1	1	0	1	0	0	1	1
6 0	1	1	1	0	1	0	1	1	0	1	1
7 0	1	1	1	1	1	0	1	1	1	1	1
8 1	0	0	0	1	1	1	1	0	0	0	0
9 1	1	0	0	1	1	1	1	1	1	1	1

For a,

CD	AB	00	01	11	10
00	1	0	1	1	
01	0	1	1	1	
11	x	x	x	x	
10	1	x	x	x	

$$\begin{aligned} a &= A + C + BD + \bar{B}D \\ &= A + C + B\bar{D} \end{aligned}$$

For b,

CD	AB	00	01	11	10
00	1	1	1	1	
01	1	0	1	0	
11	x	x	x	x	
10	1	1	x	x	

$$\begin{aligned} b &= \bar{B} + \bar{C}D + CD \\ &= \bar{B} + C\bar{D} \end{aligned}$$

For c,

	CD	00	01	11	10
AB	00	1	1	1	0
	01	1	1	1	x
	11	x	x	x	x
	10	1	1	x	x

$$c = \bar{C}D + D\bar{B}$$
$$= B + \bar{C}D$$

For d,

	CD	00	01	11	10
AB	00	1	0	1	(1)
	01	0	1	0	1
	11	x	x	x	x
	10	1	1	x	(c)

$$d = \bar{C}\bar{D} + \bar{B}\bar{D} + \bar{B}C + \bar{B}CD + A$$
$$= A + \bar{B}\bar{D} + \bar{B}C + \bar{C}\bar{D} + B\bar{C}D$$

For e,

	CD	00	01	11	10
AB	00	1	0	0	0
	01	0	0	0	1
	11	x	x	x	x
	10	1	0	x	(e)

$$e = \bar{C}D + BD$$

For f,

	CD	00	01	11	10
AB	00	1	0	0	0
	01	1	1	0	1
	11	x	x	x	x
	10	1	1	x	x

$$f = A + \bar{C}\bar{D} + \bar{B}\bar{D} + \bar{B}C$$

For g,

	CD	00	01	11	10
AB	00	0	0	1	1
	01	1	1	0	1
	11	x	x	x	x
	10	1	1	x	x

$$g = \bar{A} + \bar{B}\bar{C} + \bar{C}\bar{D} + \bar{B}C$$

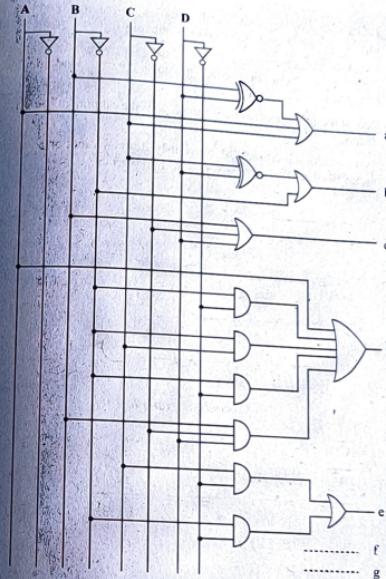


Fig.: Circuit diagram of a seven segment decoder

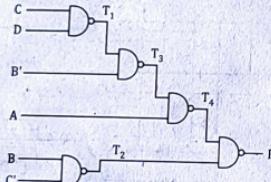
#### 5.4 Analysis Procedure

The design of a combinational circuit starts from the verbal specification of a required function and culminates with a set of output Boolean functions or a logic diagram. The analysis of a combinational circuit is somewhat the reverse process. It starts with a given logic diagram and culminates with a set of Boolean functions, a truth table, or a verbal explanation of the circuit operation.

To obtain the output Boolean functions from a logic diagram, proceed as follows:

- Label with arbitrary symbols all gate outputs that are a function of the input variables. Obtain the Boolean functions for each gate.
- Label with other arbitrary symbols those gates which are a function of input variables and/or previously labeled gates. Find the Boolean function for these gates.
- Repeat the process outlined in step 2 until the outputs of the circuit are obtained.
- By repeated substitution of previously defined functions, obtain the output Boolean functions in terms of input variables only.

**Example:**



*Fig.: Analysis example*

$$T_1 = (\overline{CD}) = \overline{C} + \overline{D}$$

$$T_2 = (\overline{BC}) = \overline{B} + C$$

$$\begin{aligned} T_3 &= (\overline{T_1}, \overline{B}) = (\overline{\overline{C}} + \overline{D}) \overline{B} = \overline{\overline{C}\overline{B}} + \overline{D\overline{B}} = \overline{\overline{C}\overline{B}} \cdot \overline{D\overline{B}} \\ &= (B + C)(B + D) \\ &= B + CD \end{aligned}$$

$$T_4 = (\overline{T_3}, A) = (\overline{(B + CD)}, A) = \overline{AB} + \overline{ACD}$$

$$\begin{aligned} F &= T_4 \cdot T_2 = \overline{AB} + \overline{ACD} \cdot (\overline{B} + C) \\ &= \overline{AB} \cdot \overline{ACD} \cdot \overline{B} + \overline{ACD} \cdot C \\ &= \overline{AB} + \overline{ACD} + \overline{B} \cdot \overline{C} \\ &= AB + ACD + \overline{B} \cdot \overline{C} \\ &= AB + ACD + BC \end{aligned}$$

## 5.5 Multilevel NAND and NOR Circuits

Steps to obtain multilevel NAND diagram from Boolean expression are given below:

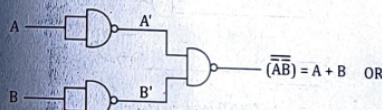
- From the given algebraic expression, draw the logic diagram with AND, OR and NOT gates. Assume that both the normal and complement inputs are available.
- Draw a second logic diagram with the equivalent NAND logic.
- Remove any two cascaded inverters from the diagram, since double inversion does not perform a logic function. Remove inverters connected to single external inputs and complement the corresponding input variable. The new logic diagram obtained is the required NAND gate implementation.



NOT



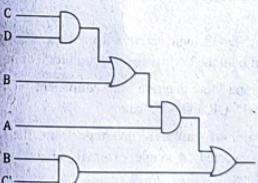
AND



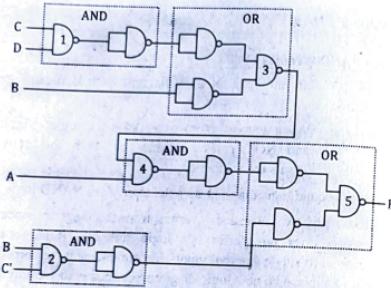
OR

**Example:**

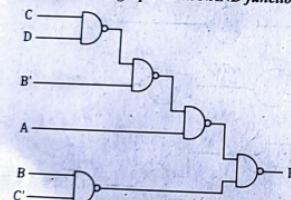
$$F = A(B + CD) + BC'$$



*(a) AND/OR implementation*



(b) Substituting equivalent NAND function



(c) NAND implementation

### Multilevel NOR Circuits

Steps to obtain multilevel NOR diagram from Boolean expression are as follows:

1. Draw the AND-OR logic diagram from the given algebraic expression assume that both the normal and the complement inputs are available.
2. Draw a second logic diagram with equivalent NOR logic substituted for each AND, OR and NOT gate.
3. Remove pairs of cascaded inverters from the diagram. Remove inverters connected to single external inputs and complement the corresponding.

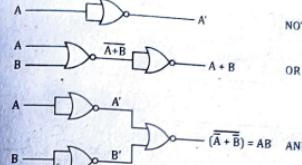
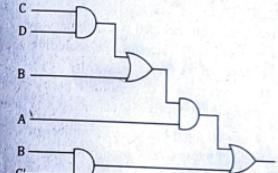


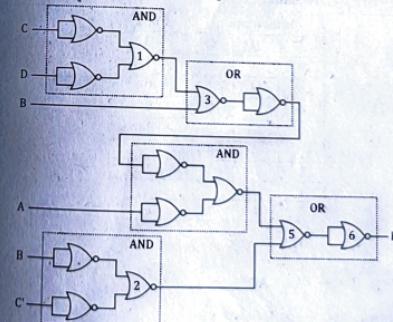
Fig.: Implementation of NOT, OR and AND by NOR gates

Example:

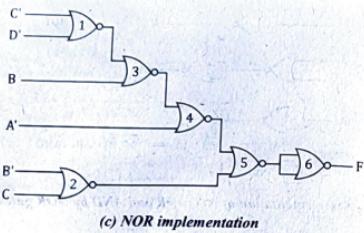
$$F = A(B + CD) + BC'$$



(a) AND/OR implementation



(b) Substituting equivalent NOR functions



### 5.6 Parity Generation and Checking

A parity bit is an extra bit included with a binary message to make a number of 1s either odd or even. The message including the parity bit is transmitted and then checked at the receiving end for errors. The circuit that generates the parity bit in the transmitter is called a parity generator and the circuit that checks the parity in the receiver is called a parity checker.

Even parity means n-bit input has an even number of 1s. For example, 110011 has even parity. Odd parity means an n-bit input has an odd number of 1s. For example, 01011 has odd parity.

#### 3-bit Odd Parity Generator

Information			Parity
A	B	C	P
0	0	0	1
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	0

$$\begin{aligned}
 P &= \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + A \bar{B} C + A B \bar{C} \\
 &= \bar{A} (\bar{B} \bar{C} + BC) + A(\bar{B} C + B\bar{C}) \\
 &= \bar{A} (\overline{B \oplus C}) + A (B \oplus C) \\
 &= A \odot (B \oplus C)
 \end{aligned}$$

For P

	BC	00	01	11	10
A	0	1	0	1	0
	1	0	1	0	1

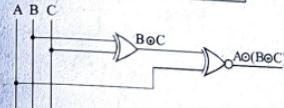


Fig.: Circuit diagram of 3-bit odd parity generator

The three input messages and the parity bit are transmitted to their destinations, where they are applied to a parity checker circuit. An error occurs if the parity of the four bit received is even, since the binary information transmitted was odd. So the output of the parity checker will be '1' when error occurs i.e., when the number of 1s in the four input is even.

#### 4-bit Odd Parity Checker

A	B	C	D	Checker
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

For Checker,

		CD 00	01	11	10
AB		1	0	1	0
00		0	1	0	1
01		1	0	1	0
11		0	1	0	1
10		1	0	0	1

$$\begin{aligned}
 \text{Checker} &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + \\
 &\quad ABCD + A\bar{B}\bar{C}D + A\bar{B}CD \\
 &= \bar{A}\bar{B}(\bar{C}\bar{D} + CD) + \bar{A}B(\bar{C}D + \bar{C}\bar{D}) + AB(\bar{C}\bar{D} + CD) + AB(\bar{C} \\
 &\quad D + \bar{C}\bar{D}) \\
 &= (\bar{A}\bar{B} + AB)(\bar{C}\bar{D} + CD) + (\bar{A}B + A\bar{B})(\bar{C}\bar{D} + C\bar{D}) \\
 &= (A\oplus B)(C\oplus D) + (A\oplus B)(C\oplus D) \\
 &= \bar{X}\bar{Y} + XY \quad [X = A\oplus B, Y = C\oplus D] \\
 &= X\otimes Y
 \end{aligned}$$

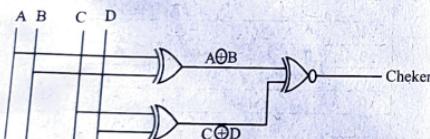


Fig.: Circuit diagram of 4-bit odd parity checker

#### SOLUTION TO IMPORTANT AND EXAM QUESTIONS

1. Design a combinational circuit with three inputs and one output.
  - i. The output is 1 when binary value of the inputs is less than or equal to 3. The output is 0 otherwise.
  - ii. The output is 1 when binary value of the inputs is an odd number.
  - iii. The output is 1 when the binary value of the inputs is an even number. [2021 Fall]

Solution:

Let the 3 inputs of combinational circuit = A, B, C

The output = Y

Now, according to the question, the truth table will be,

Inputs		Output	
A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

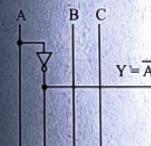
Using K-map

For Y,



∴ The function is  $Y(A, B, C) = \bar{A}$

The circuit diagram is as follows.



2. Design a combinational logic circuit with 3 inputs that provide 1 output when exactly two variables are one. [2021 Fall]

Solution:

Let the 3 inputs of combinational circuit = A, B, C and one output = Y.

Then, according to the question, the truth table will be

Inputs			Output
A	B	C	Y
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	0

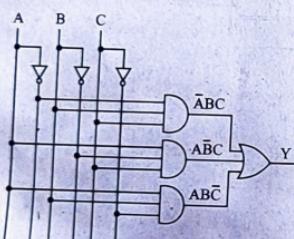
Using K-map,

For Y,

BC		00	01	11	10
A	0	0	0	1	0
	1	0	1	0	1

$$\therefore Y = \bar{A}BC + \bar{A}\bar{B}C + A\bar{B}\bar{C}$$

The circuit diagram is as follows.



### 3. Explain 3-bit even parity generator circuit clearly.

*Solution:*

3-bit even parity generator has 3 inputs A, B, C and parity P. P becomes 1 when there are odd number of 1's i.e., message information along with parity should be even for 3 bit even parity generator.

Information			Parity P
A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Plotting K-map for P,

BC		00	01	11	10
A	0	0	0	1	0
	1	0	1	0	1

$$\begin{aligned} P &= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC \\ &= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC) \\ &= \bar{A}(B \oplus C) + A(\bar{B} \oplus \bar{C}) \\ &= A \oplus B \oplus C \end{aligned}$$

Thus, even parity generator is an X-OR function which returns 1 when total number of 1s in input is odd and hence, generates an even parity bit.

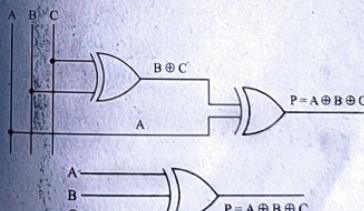


Fig.: Circuit diagram for 3 bit even parity generator

4. Design a combinational circuit using PLD (Programmable logic device) device as ROM which is used to find square of 3 bit numbers.  
[Fall 2020]

*Solution:*

Binary	Decimal	Squares in	
		Decimal	Binary
000	0	0	0
001	1	1	1
010	2	4	100
011	3	9	1001
100	4	16	10000
101	5	25	11001
110	6	36	101100
111	7	49	110001

Here, we require 6 outputs

Truth table

A	B	C	a	b	c	d	e	f	
0	0	0	0	0	0	0	0	0	$0^2 = 0$
0	0	1	0	0	0	0	0	1	$1^2 = 1$
0	1	0	0	0	0	1	0	0	$2^2 = 4$
0	1	1	0	0	1	0	0	1	$3^2 = 9$
1	0	0	0	1	0	0	0	0	$4^2 = 16$
1	0	1	0	1	1	0	0	1	$5^2 = 25$
1	1	0	1	0	1	1	0	0	$6^2 = 36$
1	1	1	1	1	0	0	0	1	$7^2 = 49$

Since inputs = 3, we use  $3 \times 8$  decoder.

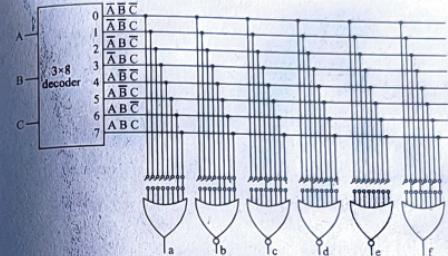


Fig.: Logic diagram to find square of a 3-bit number.

5. Design a combinational circuit that converts decimal digits from 8-4-2-1 to excess-3.  
[Spring 2019]

*Solution:*

No. of inputs = 4(B<sub>3</sub>, B<sub>2</sub>, B<sub>1</sub>, B<sub>0</sub>);

No. of outputs = 4(E<sub>3</sub>, E<sub>2</sub>, E<sub>1</sub>, E<sub>0</sub>)

Decimal	8-4-2-1 Code (BCD code)				Excess-3 Code			
	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	E <sub>3</sub>	E <sub>2</sub>	E <sub>1</sub>	E <sub>0</sub>
0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	1	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	0	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	0	1	0	1
7	0	1	1	1	1	1	0	0
8	1	0	0	0	0	1	0	1
9	1	0	0	1	1	1	0	0

Decimal 10–15 are don't care conditions.

**For E<sub>3</sub>,**

		B <sub>1</sub> B <sub>0</sub>	00	01	11	10
		B <sub>3</sub> B <sub>2</sub>	00	01	11	10
00			0	0	0	0
01			0	1	1	1
11			x	x	x	x
10			1	1	x	x

$$\begin{aligned} \therefore E_3 &= B_3 + B_2B_0 + B_2B_1 \\ &= B_3 + B_2(B_0 + B_1) \end{aligned}$$

**For E<sub>2</sub>,**

		B <sub>1</sub> B <sub>0</sub>	00	01	11	10
		B <sub>3</sub> B <sub>2</sub>	00	01	11	10
00			0	1	1	1
01			1	0	0	0
11			x	x	x	x
10			0	1	x	x

$$\begin{aligned} \therefore E_2 &= B_2\bar{B}_1\bar{B}_0 + \bar{B}_2B_0 + \bar{B}_2B_1 \\ &= B_2\bar{B}_1\bar{B}_0 + \bar{B}_2(B_0 + B_1) \end{aligned}$$

**For E<sub>1</sub>,**

		B <sub>1</sub> B <sub>0</sub>	00	01	11	10
		B <sub>3</sub> B <sub>2</sub>	00	01	11	10
00			1	0	1	0
01			1	0	1	0
11			x	x	x	x
10			1	0	x	x

$$\begin{aligned} \therefore E_1 &= \bar{B}_1\bar{B}_0 + B_1B_0 \\ &= B_1 \otimes B_0 \end{aligned}$$

**For E<sub>0</sub>,**

		B <sub>1</sub> B <sub>0</sub>	00	01	11	10
		B <sub>3</sub> B <sub>2</sub>	00	01	11	10
00			1	0	0	1
01			1	0	0	1
11			x	x	x	x
10			1	0	x	x

$$E_0 = \bar{B}_0$$

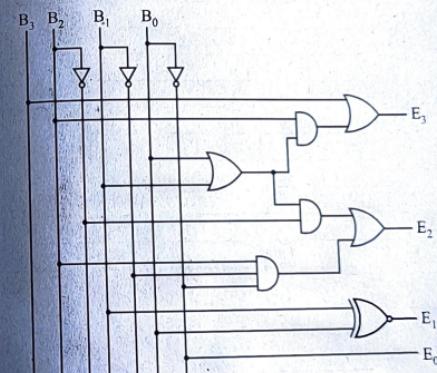


Fig.: Circuit diagram of BCD (8-4-2-1) code to excess-3 code converter

6. Design a code conversion circuit to convert binary code into Gray code.  
[Fall 2019]

*Solution:*

No. of inputs (binary code) = 4(B<sub>3</sub>, B<sub>2</sub>, B<sub>1</sub>, B<sub>0</sub>)

No. of outputs (Gray code) = 4(G<sub>3</sub>, G<sub>2</sub>, G<sub>1</sub>, G<sub>0</sub>)

Inputs (Binary code)				Outputs (Gray code)			
$B_3$	$B_2$	$B_1$	$B_0$	$G_3$	$G_2$	$G_1$	$G_0$
0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	1	0	0	1
0	1	0	0	0	0	1	0
0	1	0	1	1	0	1	1
0	1	1	0	0	1	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

Using K-map,

For  $G_3$ ,

$B_3B_0$	00	01	11	10
$B_3B_2$	00	0	0	0
	01	0	0	0
	11	1	1	1
	10	1	1	1

$$\therefore G_3 = B_3$$

For  $G_2$ ,

$B_3B_0$	00	01	11	10
$B_3B_2$	00	0	0	0
	01	1	1	1
	11	0	0	0
	10	1	1	1

$$G_2 = \bar{B}_3B_2 + B_3\bar{B}_2 + B_2 \oplus B_3$$

For  $G_1$ ,

$B_3B_0$	00	01	11	10
$B_3B_2$	00	0	0	1
	01	1	1	0
	11	1	1	0
	10	0	1	1

$$G_1 = B_2\bar{B}_1 + \bar{B}_2B_1 = B_1 \oplus B_2$$

For  $G_0$ ,

$B_3B_0$	00	01	11	10
$B_3B_2$	00	0	1	0
	01	0	1	1
	11	0	1	0
	10	0	1	1

$$G_0 = \bar{B}_3B_0 + \bar{B}_1B_0 = B_0 \oplus B_1$$

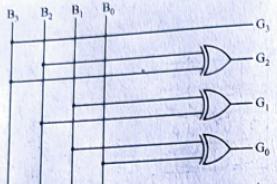


Fig.: Circuit diagram for binary to gray code converter

7. Design a combinational circuit that has four inputs and two outputs, one of the outputs high when majority of inputs are high. The second output is high only when all inputs are of same type.  
[Spring 2018, Fall 2018, Spring 2017, Fall 2017]

**Solution:**

Let A, B, C, D are four inputs and X and Y are two outputs.

Now, according to the question, the truth table of the combinational circuit will be.

**Note:**

X is high when 3 or more inputs are high

Y is high when A = B = C = D (all inputs are same)

A	B	C	D	X	Y
0	0	0	0	0	1
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	1	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	1	0
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	1

Using K-map,

For X,

CD	00	01	11	10
AB	00	0	0	0
	01	0	0	1
	11	0	1	1
	10	0	0	1

$$X = BCD + ABD + ABC + ACD$$

For Y,

CD	00	01	11	10
AB	00	1	0	0
	01	0	0	0
	11	0	0	1
	10	0	0	0

$$Y = \bar{A} \bar{B} \bar{C} \bar{D} + ABCD$$

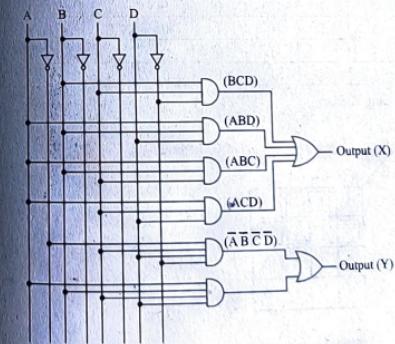


Fig.: Logic diagram of combinational circuit

8. Design a combinational circuit that accepts a 4 bit number as input and generates the output binary number equal to the  $2^3$  complement of input number.  
 [Spring 2016]

*Solution:*

Let the 4-bit input = A, B, C, D and output = W, X, Y, Z which is  $2^3$  complement of input number.

The truth table is as follows:

Inputs				Outputs			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	0	1	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	1	0	0	1

Using K-map,

**For W,**

CD		00	01	11	10
AB		0	1	1	1
00	0	1	1	1	
01	1	1	1	1	
11	0	0	0	0	
10	0	0	0	0	

$$\therefore W = \bar{A}D + \bar{A}B$$

**For X,**

CD		00	01	11	10
AB		0	1	1	1
00	0	1	0	0	0
01	1	0	0	0	0
11	1	0	0	0	0
10	0	1	1	1	1

$$\bar{B}\bar{C}\bar{D} + \bar{B}D + \bar{B}C$$

**For Y,**

CD		00	01	11	10
AB		0	1	0	1
00	0	1	0	1	
01	0	1	0	1	
11	0	1	0	1	
10	0	1	0	1	

$$Y = \bar{C}D$$

**For Z,**

CD		00	01	11	10
AB		0	1	1	0
00	0	1	1	0	
01	0	1	1	0	
11	0	1	1	0	
10	0	1	1	0	

$$Z = D$$

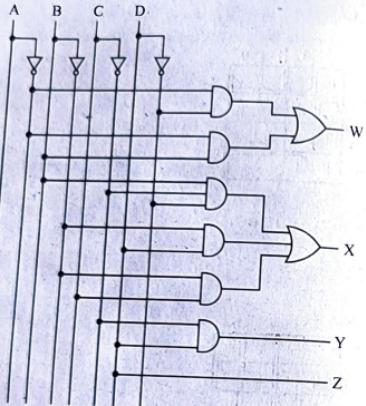


Fig.: Circuit diagram of combinational circuit

9. Design a combinational circuit that converts a decimal digit from the 2421 code to BCD. [Fall 2016]

*Solution:*

Let A, B, C, D be the inputs and  $B_3, B_2, B_1, B_0$  be the outputs.  
According to the question, truth table will be

Decimal digit	2421 code				BCD code			
	A	B	C	D	$-B_3$	$-B_2$	$-B_1$	$B_0$
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	1	0	1	1	0	1	0	0
6	1	1	0	0	0	1	1	0
7	1	1	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0
9	1	1	1	1	1	0	0	1

Using K-map,

For  $B_3$ ,

CD	00	01	11	10
AB	00	0	0	0
	01	0	x	x
	11	0	1	1
	10	x	x	0

$$\therefore B_3 = BC$$

For  $B_2$ ,

CD	00	01	11	10
AB	00	0	0	0
	01	1	x	x
	11	1	1	0
	10	x	x	1

$$\therefore B_2 = B\bar{C} + A\bar{B}$$

For  $B_1$ ,

CD	00	01	11	10
AB	00	0	0	1
	01	0	x	x
	11	1	1	0
	10	x	x	0

$$\therefore B_1 = \bar{A}C + A\bar{C} = A \oplus C$$

For  $B_0$ ,

CD	00	01	11	10
AB	00	0	1	0
	01	0	x	x
	11	0	1	0
	10	x	x	1

$$\therefore B_0 = \bar{C}D + C\bar{D} + A\bar{B} = (C \oplus D) + A\bar{B}$$

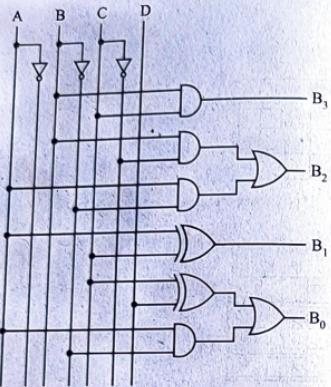


Fig.: Circuit diagram of combinational circuit

10. Design a combinational circuit that converts a decimal digit from the 2421 code to 84-2-1 code to binary. [Spring 2015]

*Solution:*

Let, inputs (2421 code) = A, B, C, D

Output (84-2-1 code) = W, X, Y, Z

Output (Binary) = B<sub>3</sub>, B<sub>2</sub>, B<sub>1</sub>, B<sub>0</sub>

Now, truth table of required combination circuit is

Decimal digit	2421 code				84-2-1 code				Binary code			
	A	B	C	D	W	X	Y	Z	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	1	0	0	0	1
2	0	0	1	0	0	1	1	0	0	0	1	0
3	0	0	1	1	0	1	0	1	0	0	1	1
4	0	1	0	0	0	1	0	1	0	0	0	0
5	1	0	1	1	1	0	1	1	0	1	0	1
6	1	1	0	0	1	0	1	0	0	1	1	0
7	1	1	0	1	1	0	0	1	0	1	1	1
8	1	1	1	0	1	0	0	0	1	0	0	0
9	1	1	1	1	1	1	1	1	0	0	0	1

Using K-map for W, X, Y, Z with inputs A, B, C, D

For W,

		CD	00	01	11	10
		AB	00	0	0	0
			01	0	x	x
			11	1	1	1
			10	x	x	1

$$\therefore W = A$$

For X,

		CD	00	01	11	10
		AB	00	0	1	1
			01	1	x	x
			11	0	0	0
			10	x	x	0

$$\therefore X = \bar{A}B + C\bar{A} + \bar{B}\bar{C}D + BCD$$

$$= \bar{A}(B + C) + D(\bar{B}\bar{C} + BC)$$

$$= \bar{A}(B + C) + D(B \odot C)$$

For Y,

		CD	00	01	11	10
		AB	00	0	1	1
			01	0	x	x
			11	1	0	0
			10	x	x	1

$$Y = AB + AD + CD + \bar{A}\bar{C}D$$

For Z,

		CD				
		AB	00	01	11	10
AB	00	0	1	0	1	
	01	0	x	x	x	
	11	0	1	0	1	
	10	x	x	1	x	

$$\therefore Z = \bar{C}D + \bar{C}\bar{D} + A\bar{B}$$

Using K-map for  $B_3, B_2, B_1, B_0$  using inputs W, X, Y, Z

For  $B_3$ ,

		YZ				
		WX	00	01	11	10
WX	00	0	x	x	x	
	01	0	0	0	0	
	11	x	x	1	x	
	10	1	0	0	0	

$$\therefore B_3 = WX + W\bar{Y}\bar{Z}$$

For  $B_2$ ,

		YZ				
		WX	00	01	11	10
WX	00	0	x	x	x	
	01	1	0	0	0	
	11	x	x	0	x	
	10	0	1	1	1	

$$\therefore B_2 = \bar{X}Y + X\bar{Y}\bar{Z} + W\bar{X}Z$$

For  $B_1$ ,

		YZ				
		WX	00	01	11	10
WX	00	0	x	x	x	
	01	0	1	0	1	
	11	x	x	0	x	
	10	0	1	0	1	

$$B_1 = \bar{Y}Z + Y\bar{Z} = Y \oplus Z$$

For  $B_0$ ,

		YZ				
		WX	00	01	11	10
WX	00	0	x	x	x	
	01	0	1	1	0	
	11	x	x	1	x	
	10	0	1	1	0	

$$B_0 = Z$$

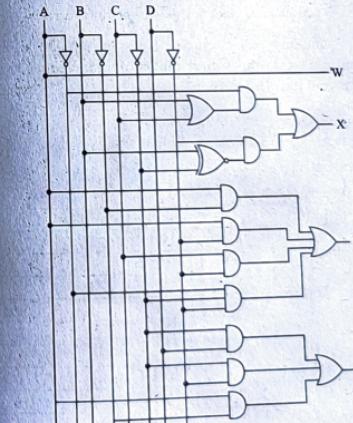


Fig.: Combinational circuit for 2421 to 84-2-1 converter

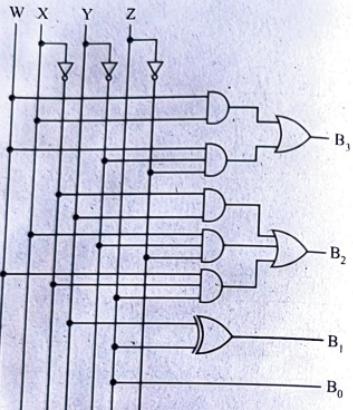


Fig.: Combinational circuit for 84-2-1 to binary converter

11. Design a combinational circuit that accepts a 3-bit number as input and generates the output binary number equal to the 2's complement of input number.  
[Spring 2014]

*Solution:*

Let, 3 bit inputs = ABC and

2's complement outputs = XYZ

The truth table is as follows:

Inputs			Outputs		
A	B	C	X	Y	Z
0'	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	0
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	0	1	1
1	1	0	0	1	0
1	1	1	0	0	1

Using K-map,

For X,

BC		00	01	11	10
A		0	0	1	1
		1	0	0	0

$$X = \bar{A}C + \bar{A}\bar{B} + A\bar{B}\bar{C}$$

For Y,

BC		00	01	11	10
A		0	0	1	0
		1	0	1	0

$$Y = \bar{B}C + \bar{B}\bar{C} = B \oplus C$$

For Z,

BC		00	01	11	10
A		0	0	1	1
		1	0	1	0

$$Z = C$$

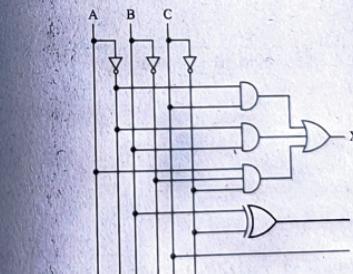


Fig.: Combinational circuit diagram