BUDDHA SERIES

(Unit Wise Solved Question & Answers)

Course – B.Tech (ECE)

College – Buddha Institute of Technology

(AKTU CODE-525)

Department: Electronics and Communication Engineering

Subject: Digital System Design

(BEC-303)

Faculty Name: Anil Singh

Unit – 1

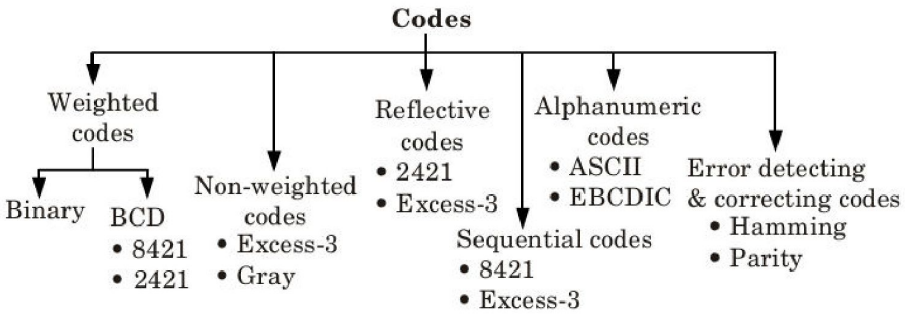
Logic simplification and combinational logic design:

Que 1.1. Describe the binary codes. Show the classification of binary codes in tabular format.

Answer

1. Code is the representation of group of symbols, words, or letters. As the digital data is used as group of binary numbers, so, we call it as the binary codes.
2. These binary codes are used for the designing and analysis of digital circuit, computer applications, in digital communication. The codes are classified into certain following categories:
    i. Weighted codes
    ii. Non-weighted codes Ref
    iii. lective codes
    iv. Sequential codes v. Alphanumeric codes
    v. Error detecting and correcting codes.

3. Since, all these codes use only 0 and 1, so it is easier to implement. The binary codes can also be used for representing the numbers as well as the alphanumeric letters.
4. The classification of codes can be composed in tabular form which is as follows:

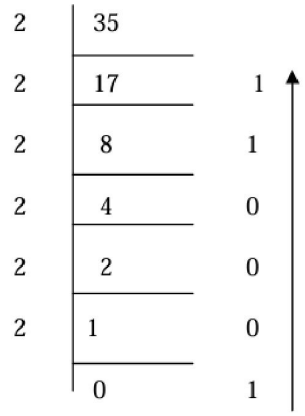


5. Weighted binary codes are those which obey the positional weight for the number to represent.
6. In non-weighted codes, the positional weights are not assigned.
7. In reflective code, the reflectivity is desirable. For example, in 9's complement subtraction, i.e., code for 9 is the complement for 0, code for
8. is complement of 1, 7 for 2, 6 for 3 and 5 for 4. 8. In sequential code, each succeeding code is one binary number greater than the preceding code.
9. The alphanumeric codes are designed to represent numbers as well as characters.
10. The error detecting and correcting codes are used to detect and correct the error like 0 may change to 1 or vice-versa by using some special codes which possess the capacity to detect and correct the error.

Que 1.2. Convert the following decimal numbers into binary.

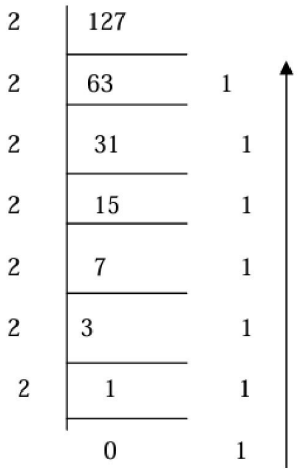
(i). 35 (ii). 127

Solution: (i)



So $(35)_{10} = (10011)_2$

(ii)

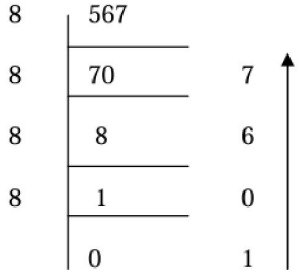


$(127)_{10} = (1111111)_2$

Que 1.3. Convert the following decimal numbers into octal.

(i) 567 (ii) 1276

Solution: (i)



So $(567)_{10} = (1067)_8$

(ii)

8	1276	
8	159	4
8	19	7
8	2	3
	0	2

So $(1276)_{10} = (2374)_8$

Que 1.4. Convert the following decimal numbers into hexadecimal.

(i) 8537 (ii) 98765

Solution: (i)

16	8537	
16	533	9
16	33	5
16	2	1
	0	2

So $(8537)_{10} = (2159)_{16}$

(ii)

16	98765	
16	6172	D
16	385	C
16	24	1
16	1	8
	0	1

So $(98765)_{10} = (181CD)_{16}$

Que 1.5. Convert the following decimal numbers into binary.

(i) 0.625 (ii) 0.6

Solution: (i)

Decimal	Product	Integer part
.625 *x* 2	1.25	1
.250 *x* 2	0.5	0
.500 *x* 2	1.0	1
0		
Stop		

$(0.625)_{10} = (0.101)_2$

(ii)

Decimal	Product	Integer part
0.600 *x* 2	1.200	1
0.200 *x* 2	0.400	0
0.400 *x* 2	0.800	0
0.800 *x* 2	1.600	1

$(0.6)_{10} = (0.1001(1001) \ldots)_2$

Que 1.6. Do the following conversions:

(i). (965.125)10 to octal

(ii). (8765.025)10 to hexadecimal

(iii) (6754.05)8 to decimal.

Solution: (i)

Integer part

8	965	
8	120	5
8	15	0
8	1	7
	0	1

Fractional part

Decimal	Product	Integer part
0.125 *x* 8	1. 00	1

So $(965.125)_{10} = (1705.1)_8$

(ii)

Integer part

16	8765	
16	547	D
16	34	3
16	2	2
	0	2

Fractional part

Decimal	Product	Integer part
0.025 *x* 16	0. 4	0
0.4 x 16	6.4	6
0.4 x 16	6.4	6

repeated value

So $(8765.025)_{10} = (223D.0666\ldots)_{16}$

(iii). $(6754.05)_8$ to decimal

$$6754.05 = 6x8^3 + 7x8^2 + 5x8^1 + 4x8^0 + 0x8^{-1} + 5x8^{-2}$$
$$= 3072 + 448 + 40 + 4 + 0 + .071285$$
$$= 3564.078125$$

So $(6754.05)_8 = (3564.078125)_{10}$

Que 1.7. Do the following conversions:

(i) $(1100110111110.1011)_2$ to octal

(ii) $(2467.534)_8$ to binary

Solution: (i)

Binary number : 001 100 110 111 110 . 101 100
Octal Equivalent: 1 4 6 7 6 3 4

$(1100110111110.1011)_2 = (14676.34)_8$

(ii)

Octal number : 2 4 6 7 . 5 3 4
Binary Equivalent: 010 100 110 111 . 101 011 100

$(2467.534)_8 = (10100110111.1010111)_2$

Que 1.8. Do the following conversions:

(i) (1100110101101111110.101)2 to hexadecimal

(ii) (2AB6E7.5D4)16 to binary

Solution:

(i)

Binary 0011 0011 0101 1011 1110 . 1010
Hexadecimal 3 3 5 B E A

$(1100110101101111110.101)_2 = (335BE.A)_{16}$

(ii)

Hexadecimal: 2 A B 6 E
Binary 0010 1010 1011 0110 1110

7 . 5 D 4
0111 . 0101 1101 0100

$(2AB6E7.5D4)_{16} = (10101010110110111001 11.0101110101)_2$

Que 1.9. Convert the hexadecimal number **4AC7.4B** in to its equivalent octal number.

Solution: The given hexadecimal number is first converted to binary number and then converted to octal number.

Hexadecimal:	4	A	C	7	.	4	B	
Binary:	0100	1010	1100	0111	.	0100	1011	

Now $(100101011000111.01001011)_2$ to octal

	100	101	011	000	111	. 010	010	110
Octal :	4	5	3	0	7	2	2	6

$$(4AC7.4B)_{16} = (45307.226)_8$$

Que 1.10. Perform the following binary additions.

(i) 110111 + 11010

(ii) 101101.101 + 101011.011

(iii) 1101101.101 + 110110.01

Solution:

(i)

Carry → 1 1 1 1 1 0
1 1 0 1 1 1
0 1 1 0 1 0
1 0 1 0 0 0 1

(ii)

Carry → 1 0 1 1 1 1 1 1 1
1 0 1 1 0 1 . 1 0 1
1 0 1 0 1 1 . 0 1 1
1 0 1 1 0 0 1 . 0 0 0

(iii)

Carry → 1 1 1 1 1 0 0 0 0 0
1 1 0 1 1 0 1 . 1 0 1
0 1 1 0 1 1 0 . 0 1 0
1 0 1 0 0 0 1 1 . 1 1 1

Binary Subtraction:

a	*b*	difference	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Que 1.11. Perform the following binary operations.

(i) 1101101 – 1100111

(ii) 11011.01 – 10101.11

(iii) 1101.101 – 1001.011

Solution:

(i)

Borrow →

```
  0 0 0 0 1 1 0
  1 1 0 1 1 0 1
  1 1 0 0 1 1 1
  0 0 0 0 1 1 0
```

(ii)

Borrow →

```
  0 1 0 1 1  0
  1 1 0 1 1.0 1
  1 0 1 0 1.1 1
  0 0 1 0 1.1 0
```

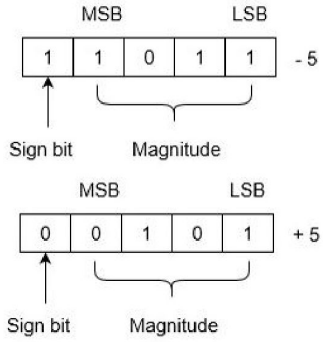
(iii)

Borrow →

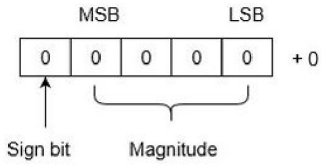
```
  0 0 0 0  1 0
  1 1 0 1 .1 0 1
  1 0 0 1 .0 1 1
  0 1 0 0 .0 1 0
```

Que 1.12. The representation of -5 and +5 in 2's complement form.

Solution:



This system's benefit is that 0 only has one representation for -0 and +0. In the 2's complement form, zero (0) is seen as always positive (sign bit is 0). It is therefore a singular portrayal.



Que 1.13. Multiply 10101 by 10011.

Solution:

Multiplicand	1 0 1 0 1
Multiplier	1 0 0 1 1
Partial products	1 0 1 0 1
	1 0 1 0 1
	0 0 0 0 0
	0 0 0 0 0
	1 0 1 0 1
Product	1 1 0 0 0 1 1 1 1

Que 1.14. Divide 1101101 by 101.
Solution:

Divisor) *Quotient* / *Divident*

```
      10101 .11..
101 ) 1101101
      101
      00111
      00101
      0001001
      0000101
      00001000
      00000101
      000000110
      000000101
      000000001
```

So, quotient is $(10101.11)_2$ and remainder $(.01)_2$

Que 1.14. Perform decimal additions in 8421 code
(a). 25+13
(b). 679.6 + 536.8

Solution:
(a)

In BCD	25=	0010 0101
In BCD	+13 =	+0001 0011
	38	0011 1000

No carry, no illegal code. This is the corrected sum

(b)

679.6 = 0110 0111 1001 .0110 in BCD
+536.8 = +0101 0011 0010 .1000 in BCD

1216.4 1011 1010 0110 . 1110 illegal codes
+0110 + 0011+0110 +. 0110 add 0110 to each

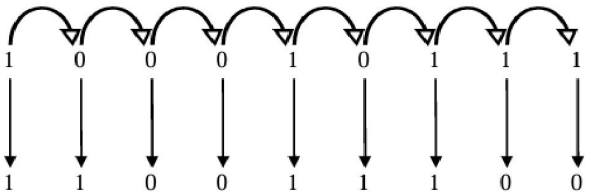
(1)0001 (1)0000 (1)0101. (1)0100 propagate carry
/ / / /
+1 +1 +1 +1

0001 0010 0001 0110 . 0100

1 2 1 6 . 4

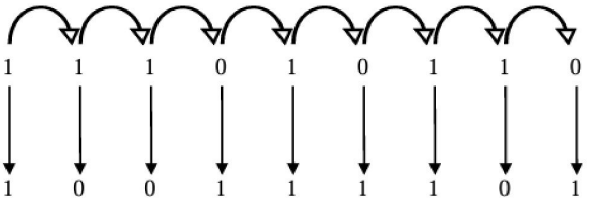
Que 1.15. Find the gray equivalent of the following binary numbers:
(i). 100010111
(ii). 111010110
(iii). 10000101011

Solution: (i)



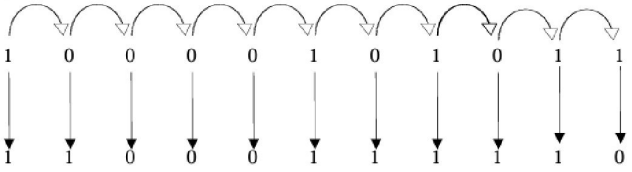
$(100010111)_2 = (110011100)_g$

(ii)



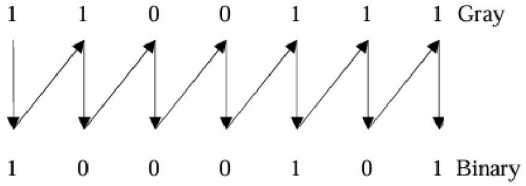
$(111010110)_2 = (100111101)_g$

(iii)



$(10000101011)_2 = (11000111110)_g$

- **Gray Code to Binary:**



$(1100111)_g = (1000101)_2$

Que 1.16. Represent the decimal number 6 in
(i) excess-3 code,
(ii) BCD code,
(iii) Gray code,
(iv) 8421 code and
(v) 2421 codes.

Solution: (i) Excess-3 code:

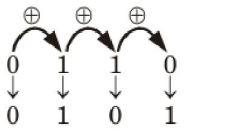
6 (in BCD)	= 0110
+ 3	= 0011
9	= 1001

(iii) BCD code:

$(6)_{10}$ = 0110 (in BCD)

(iv) Gray code:

$(6)_2$ =0110



Gray code = 0101

(iv) 8 4 $\bar{2}$ $\bar{1}$ code:

$(6)_2 = 0110$

$(6)_{8,4,-2,-1} = 1010$

(v) 2421 code:

$(6)_{2,4,2,1} = 1100$

Que 1.17. Using the theorems of Boolean algebra, prove the following identities:

(i) $(A+B)\cdot(A+\overline{A}.\overline{B})\cdot C+\overline{A}\cdot(B+\overline{C})+\overline{A}\cdot B + A\cdot B\cdot C = A+B+C$

(ii) $(A+A\cdot\overline{B})\cdot(A\cdot C+A\cdot\overline{C}\cdot(\overline{A}+B))\cdot(B+C) = A\cdot B+A\cdot C$

(iii) $A\cdot B+B\cdot C+A\cdot C = \overline{A}\cdot B+\overline{B}\cdot C+A\cdot\overline{C}$

Solution: (i) L.H.S.

$$= (A+B)\cdot(A+\overline{A}\cdot\overline{B})\cdot C+\overline{\overline{A}\cdot(B+\overline{C})}+\overline{A}\cdot B+A\cdot B\cdot C$$
$$= (A+B)\cdot(A+\overline{A}\cdot\overline{B})\cdot C+\overline{\overline{A}}+\overline{(B+\overline{C})}+\overline{A}\cdot B+A\cdot B\cdot C$$
$$= (A+B)\cdot(A+\overline{B})\cdot C+\overline{\overline{A}}+\overline{(B+\overline{C})}+$$
$$+\overline{A}\cdot B+A\cdot B\cdot C$$
$$= A\cdot C+A+\overline{B}\cdot C+B\cdot(\overline{A}+C)$$
$$= A+A.C+\overline{B}\cdot C+B\cdot\overline{A}+C$$
$$= A+C+B\cdot\overline{A}$$
$$= A+B+C$$
$$= \text{R.H.S}$$

(ii)

$$= (A+A\cdot\overline{B})\cdot(A\cdot C+A\cdot\overline{C}(\overline{A}+B))(B+C)$$
$$= A.A(C+\overline{C}(\overline{A}+B))(B+C)$$
$$= A(C+\overline{A}+B)(B+C)$$
$$= A(C+B)(B+C)$$
$$= A.(B+C)$$
$$= A.B+A.C$$
$$= \text{R.H.S.}$$

(iii)

$$= A\cdot\overline{B}+B\cdot\overline{C}+\overline{A}\cdot C$$
$$= A\cdot\overline{B}\cdot 1+1\cdot B\cdot\overline{C}+\overline{A}\cdot 1\cdot C$$
$$= A\cdot\overline{B}\cdot(C+\overline{C})+(A+\overline{A})\cdot B\cdot\overline{C}+\overline{A}\cdot(B+\overline{B})\cdot C$$
$$= A\cdot\overline{B}\cdot C+A\cdot\overline{B}\cdot\overline{C}+A\cdot B\cdot\overline{C}+\overline{A}\cdot B\cdot\overline{C}$$
$$+\overline{A}\cdot B\cdot C+\overline{A}\cdot\overline{B}\cdot C$$
$$= A\cdot\overline{B}\cdot C+\overline{A}\cdot\overline{B}\cdot C+A\cdot\overline{B}\cdot\overline{C}+A\cdot B\cdot\overline{C}$$
$$+\overline{A}\cdot B\cdot\overline{C}+\overline{A}\cdot B\cdot C$$
$$= \overline{B}\cdot C\cdot(A+\overline{A})+A\cdot\overline{C}\cdot(\overline{B}+B)+\overline{A}\cdot B\cdot(\overline{C}+C)$$
$$= \overline{B}\cdot C+A\cdot\overline{C}+\overline{A}\cdot B$$
$$= \text{R.H.S.}$$

Que 1.18. Simplify the following Boolean expression to a minimum number of literals.

i. $\overline{A}\overline{C}+ABC+A\overline{C}+A\overline{B}$

ii. $(\overline{x}\,\overline{y}+z)+z+xy+wz$

Solution: i.

$\overline{A}\overline{C}+ABC+A\overline{C}+A\overline{B}$:

Let

$$Y = \overline{A}\overline{C}+A\overline{C}+A\overline{B}+ABC$$
$$= \overline{C}(\overline{A}+A)+A(\overline{B}+BC)$$
$$= \overline{C}+A(\overline{B}+C)$$
$$= \overline{C}+AC+A\overline{B}$$
$$= \overline{C}+A+A\overline{B}$$
$$= \overline{C}+A(1+\overline{B})$$
$$= A+\overline{C}$$

ii.

$$(\bar{x}\,\bar{y} + z) + z + xy + wz$$

Let

$$\begin{aligned} Y &= (\bar{x}\,\bar{y} + z) + z + xy + wz \\ &= \bar{x}\,\bar{y} + z + xy + wz \\ &= \bar{x}\,\bar{y} + xy + z\,(1 + w) \\ &= \bar{x}\,\bar{y} + xy + z = x \odot y + z \end{aligned}$$

Que 1.19. Simplify the following expressions

$$\mathbf{Y = BC + B\bar{C} + BA}$$

Solution:

$$\begin{aligned} &Y = BC + B\bar{C} + BA \\ &Y = B(C + \bar{C}) + BA \\ &Y = B + BA \qquad (\because C + \bar{C} = 1) \\ &Y = B(1 + A) \qquad (\because 1 + A = 1) \\ &Y = B.1 \\ &\mathbf{Y = B} \end{aligned}$$

Que 1.20. Simplify the following expressions

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

Solution:

$$\begin{aligned} &Y = A + \bar{A}B + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}E \\ &Y = A + \bar{A}(B + \bar{B}C + \bar{B}\bar{C}D + \bar{B}\bar{C}\bar{D}E) \\ &Y = A + (B + \bar{B}C + \bar{B}\bar{C}D + \bar{B}\bar{C}\bar{D}E) \\ &Y = A + (B + \bar{B}(C + \bar{C}D + \bar{C}\bar{D}E)) \\ &Y = A + (B + (C + \bar{C}D + \bar{C}\bar{D}E)) \\ &Y = A + \left(B + \left(C + \bar{C}(D + \bar{D}E)\right)\right) \\ &Y = A + (B + (C + (D + \bar{D}E))) \\ &\mathbf{Y = A + B + C + D + E} \end{aligned}$$

Que 1.21. Simplify the following expressions

$$\mathbf{Y = C + \overline{B\bar{C}}}$$

Solution:

$$\begin{aligned} &Y = C + \bar{B} + \bar{C} \\ &Y = (C + \bar{C}) + \bar{B} \\ &Y = 1 + \bar{B} \\ &\mathbf{Y = 1} \end{aligned}$$

Que 1.22. Simplify the following expressions

$$\mathbf{Y = \bar{A}\bar{B}(\bar{A} + B)(\bar{B} + B)}$$

Solution:

$$\begin{aligned} &Y = (\bar{A} + \bar{B})(\bar{A} + B)(1) \\ &Y = \bar{A}\bar{A} + \bar{A}B + \bar{B}\bar{A} + \bar{B}B \\ &Y = 0 + \bar{A}(B + \bar{B}) + 0 \\ &Y = \bar{A}(1) \\ &\mathbf{Y = \bar{A}} \end{aligned}$$

Que 1.23. Simplify the following expressions

$$\mathbf{Y = (A + C)(AD + A\bar{D}) + AC + C}$$

Solution:

$$Y = (A + C)(A(D + \overline{D})) + AC + C$$
$$Y = (A + C)(A)(1) + AC + C$$
$$Y = AA + AC + AC + C$$
$$Y = A + AC + C$$
$$Y = A(1 + C) + C$$
$$\mathbf{Y = A + C}$$

Que 1.24. Simplify the following expressions

$$\mathbf{Y = \overline{A}(A + B) + (B + AA)(A + \overline{B})}$$

Solution:

$$Y = \overline{A}A + \overline{A}B + BA + B\overline{B} + AAA + AA\overline{B}$$
$$Y = 0 + \overline{A}B + AB + 0 + A + A\overline{B}$$
$$Y = B(\overline{A} + A) + A + A\overline{B}$$
$$Y = B + A(1 + \overline{B})$$
$$\mathbf{Y = A + B}$$

Que 1.25. Simplify the following expressions

$$\mathbf{Y = \overline{\overline{A + B\overline{C}} + D(\overline{E + \overline{F}})}}$$

Solution:

$$Y = \overline{(A + B\overline{C})}\ \ \overline{(D(\overline{E + \overline{F}}))}$$
$$Y = (A + B\overline{C})(\overline{D} + (\overline{E + \overline{F}})$$
$$\mathbf{Y = (A + B\overline{C})(\overline{D} + E + \overline{F})}$$

Que 1.26. Simplify the following expressions

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

Solution:

$$Y = AB + AB + AC + BB + BC$$
$$Y = AB + AC + B + BC$$
$$Y = AB + AC + B(1 + C)$$
$$Y = AB + AC + B$$
$$Y = B(A + 1) + AC$$
$$\mathbf{Y = B + AC}$$

Que 1.27. Simplify the following expressions

$$\mathbf{Y = A\overline{B} + A(\overline{B + C}) + B(\overline{B + C})}$$

Solution:

$$Y = A\overline{B} + A(\overline{B}\overline{C}) + B\overline{B}\overline{C}$$
$$Y = A\overline{B}(1 + \overline{C})$$
$$\mathbf{Y = A\overline{B}}$$

Que 1.28. Simplify the following expressions

$$\mathbf{Y = \overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + A\overline{B}C + ABC}$$

Solution:

$$Y = \overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + A\overline{B}C + ABC$$
$$Y = \overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + AC(\overline{B} + B)$$
$$Y = \overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + AC$$
$$Y = \overline{A}BC + \overline{B}\overline{C}(A + \overline{A}) + AC$$
$$Y = \overline{A}BC + \overline{B}\overline{C} + AC$$
$$Y = (\overline{A}B + A)C + \overline{B}\overline{C}$$
$$Y = (A + B)C + \overline{B}\overline{C}$$
$$\mathbf{Y = AC + BC + \overline{B}\overline{C}}$$

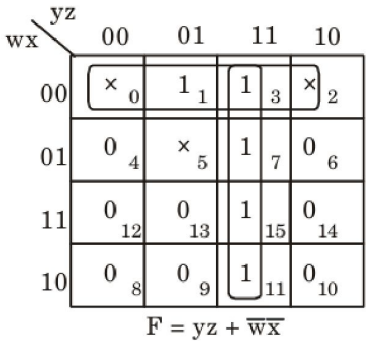
Que 1.29. Simplify the Boolean function.

F (w, x, y, z) =Σm (1, 3, 7, 11, 15)

which has the don't care conditions

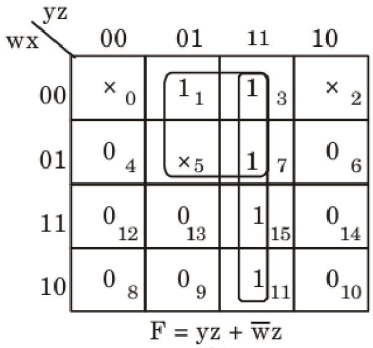
d (w, x, y, z) =Σd (0, 2, 5)

Solution



$F = yz + \overline{wx}$

(a)



$F = yz + \overline{w}z$

(b)

The K-map in Fig. (b) is more feasible because, we have to use the minimum don't care.

Que 1.30. Simplify the following expression into product of sum (POS) form

i. $A\bar{B}\bar{C} + A\bar{B}D + BCD$

ii. $AC\bar{D} + \bar{C}D + A\bar{B} + ABCD$

Solution (i)

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

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

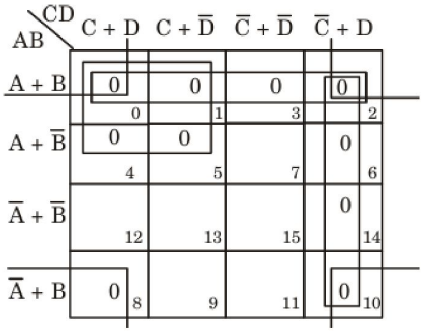
$= A\bar{B}\bar{C}(D + \bar{D}) + A\bar{B}(C + \bar{C})D + (A + \bar{A})BCD$

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

$Y = \Sigma m(7, 9, 11, 12, 13, 15)$

Now for POS form we will take complement function,

$\bar{Y} = \Pi M(0, 1, 2, 3, 4, 5, 6, 8, 10, 14)$



$$\overline{Y} = (A + B)(A + C)(\overline{C} + D)(B + D)$$

(ii)

$$\boldsymbol{ACD + \overline{C}D + A\overline{B} + ABCD}$$

Let, $Y = AC\overline{D} + \overline{C}D + A\overline{B} + ABCD$

$$= A(B+\overline{B})C\overline{D} + (A+\overline{A})(B+\overline{B})\overline{C}D + A\overline{B}(C+\overline{C})(D+\overline{D}) + ABCD$$

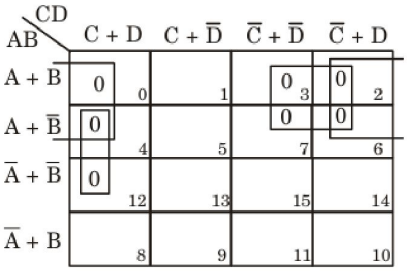
$$= A(B+\overline{B})C\overline{D} + (A+\overline{A})(B+\overline{B})\overline{C}D + A\overline{B}(C+\overline{C})(D+\overline{D}) + ABCD$$

$$= ABC\overline{D} + A\overline{B}C\overline{D} + ABCD + A\overline{B}CD + ABC\overline{D} + \overline{A}\overline{B}\overline{C}D + A\overline{B}CD + A\overline{B}\overline{C}\overline{D} + ABCD$$

$$Y = \Sigma m(1, 5, 8, 9, 10, 11, 13, 14, 15)$$

Now for POS form, we have to take complement function,

$$\overline{Y} = \Pi M(0, 2, 3, 4, 6, 7, 12)$$

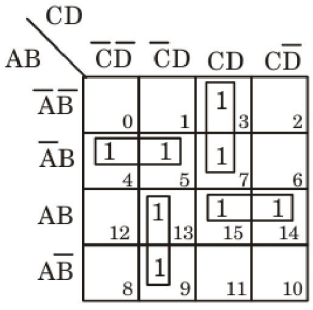


$$\overline{Y} = (A + D)(A + \overline{C})(\overline{B} + C + D)$$

Que 1.31. Minimize the given Boolean function using K-map.

F(A, B, C, D) = Σ m (3, 4, 5, 7, 9, 13, 14, 15)

Solution

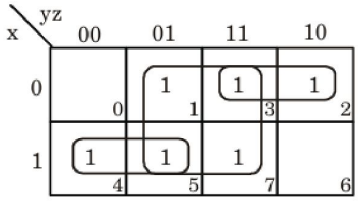


$$F = \bar{A}B\bar{C} + \bar{A}CD + A\bar{C}D + ABC$$

Que 1.32. Implement the following Boolean function with NAND gates.

F(x, y, z) = Σ m (1, 2, 3, 4, 5, 7)

Solution

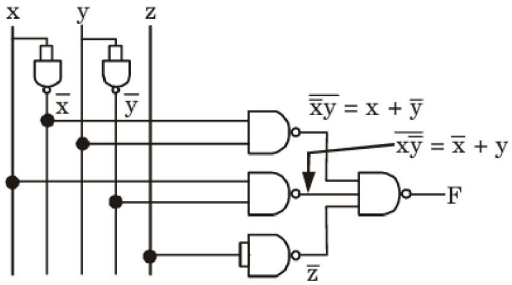


$F = z + \bar{x}y + x\bar{y}$

Implementation using NAND gates

$$F = \overline{(x + \bar{y}).(\bar{x} + y).(\bar{z})}$$

$$= z + x\bar{y} + \bar{x}.y$$



Que 1.33. Simplify the boolean function Y together with don't care condition d using K-map and implement it with two level NAND gate circuit.

$$\boldsymbol{Y = BD + BC\bar{D} + A\bar{B}C\bar{D}}$$

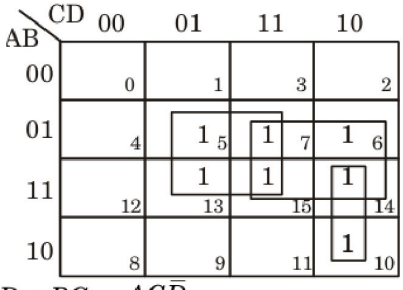
Solution

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

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

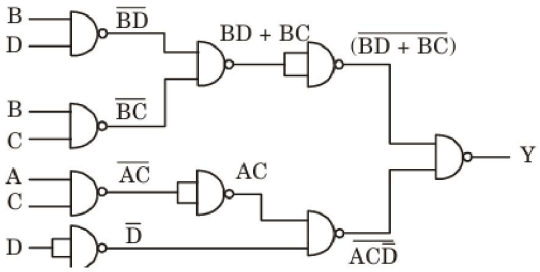
$$= ABCD + \bar{A}BCD + AB\bar{C}D + \bar{A}B\bar{C}D + ABC\bar{D} + \bar{A}BC\bar{D} + A\bar{B}C\bar{D}$$

Y =Σ m(5, 6, 7, 10, 13, 14, 15)



$Y = BD + BC + ACD$

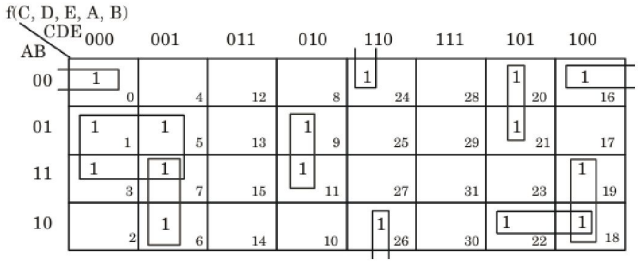
NAND gate implementation:



Que 1.34. Simplify the following Boolean function using K-map

Y = Σm (0, 1, 3, 5, 6, 7, 9, 11, 16, 18, 19, 20, 21, 22, 24, 26)

Solution

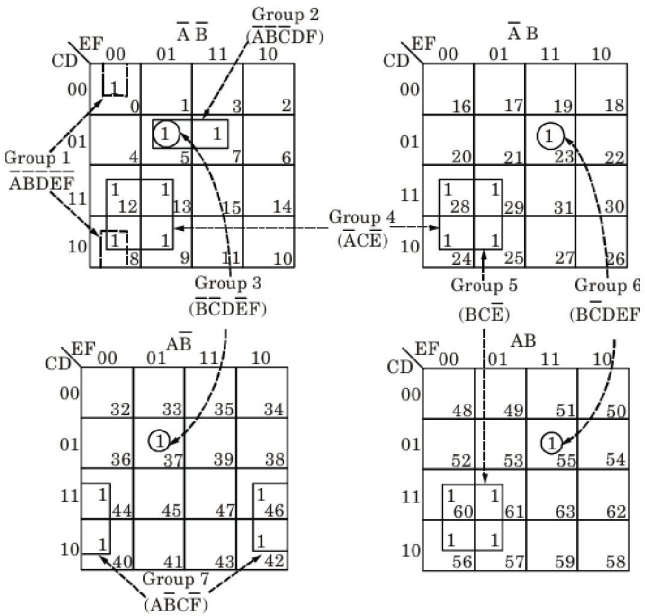


$$Y = \bar{D}\bar{E}\bar{A}\bar{B} + \bar{C}\bar{D}B + \bar{A}\bar{C}\bar{D}E + C\bar{D}A\bar{B} + \bar{C}D\bar{E}\bar{B} + CD\bar{E}\bar{B} + C\bar{D}E\bar{A} + C\bar{D}\bar{E}A$$

Que 1.35. Simplify the Boolean function

F(A, B, C, D, E, F) = Σm (0, 5, 7, 8, 9, 12, 13, 23, 24, 25, 28, 29, 37, 40, 42, 44, 46, 55, 56, 57, 60, 61)

Solution

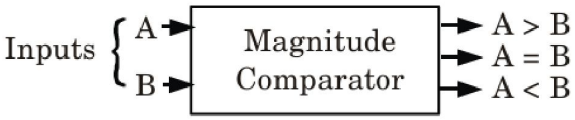


Unit – 2
Combinational circuits:

Que 2.1. What is magnitude comparator? Design a three-bit comparator circuit using logic gates.

Solution: A magnitude comparator is a combinational circuit designed primarily to compare the relative magnitude of the two binary numbers A and B.

- Naturally, the result of this comparison is specified by three binary variables that indicate, whether
  - A > B,
  - A = B
  - A < B.



Truth table:

Inputs		Outputs		
A	***B***	Z_1	Z_2	Z_3
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Boolean Equation

Z_1 is high when A > B,

$Z_1 = A B'$

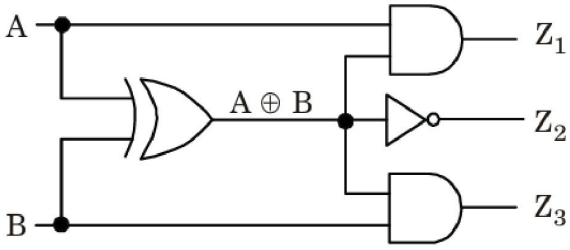
Z_2 is high when A = B,

$Z_2 = AB + A'B' = A \odot B$

Z_3 is high when A < B

$Z_3 = A'B$

Circuit Diagram



<u>3-bit magnitude comparator:</u>

$$A = A_2 A_1 A_0$$
$$B = B_2 B_1 B_0$$

Two number A and B are equal, only if all the pairs of significant digits are equal. i.e.,

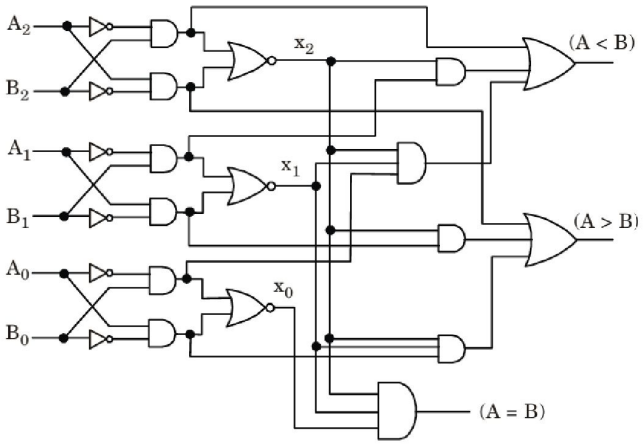
$$A_2 = B_2, A_1 = B_1, A_0 = B_0$$

When numbers are binary, then equality relation of each pair of bits can be expressed by the equivalent function as,

$$x_i = A_i B_i + \bar{A}_i \bar{B}_i \, , \, i = 0, 1, 2$$

Design procedure

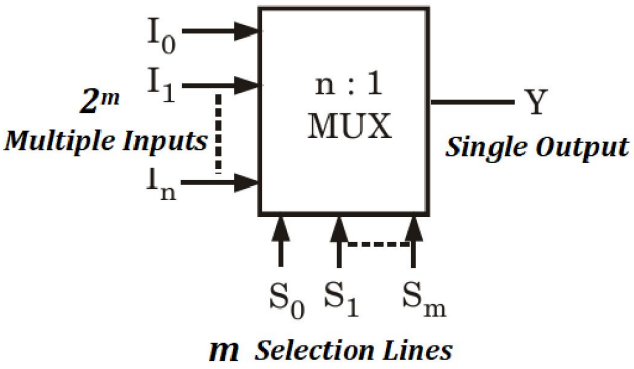
$$(A = B) = x_2 . x_1 . x_0 = (A_2 \odot B_2)(A_1 \odot B_1)(A_0 \odot B_0)$$
$$(A > B) = A_2 \bar{B}_2 + x_2 A_1 \bar{B}_1 + x_2 x_1 A_0 \bar{B}_0$$
$$(A < B) = \bar{A}_2 B_2 + x_2 \bar{A}_1 B_1 + x_2 x_1 \bar{A}_0 B_0$$



Que 2.2. What is the role of multiplexer in the digital electronics? Explain the logic how it selects a one input among several inputs.

Solution: A multiplexer (MUX), also known as a data selector, is a digital circuit that combines multiple data inputs into a single output line. It essentially acts as a switch, selecting one of the input lines to be routed to the output, based on the control signals applied to its selection lines.

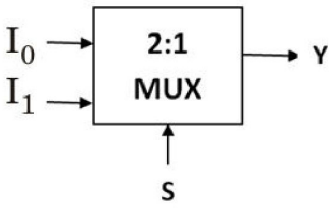
- **Multiple Inputs:** A multiplexer can have many inputs lines **n = 2^m** (e.g., 2, 4, 8, 16).
- **Single Output:** It always produces a single output line.
- **Selection Lines:** These control signals determine which of the input lines is connected to the output **(m)**.
- **Combinational Logic:** Multiplexers are combinational circuits, meaning their output depends only on the current input values.



2x1 MUX

No of input n = 2
No. of output = 1
No of selection line m = 1 ($n = 2^m$)

Block diagram of 2 to 1 MUX



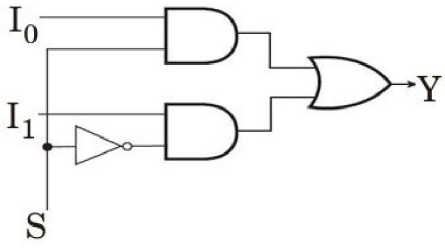
Truth table

S	Y
0	I_0
1	I_1

Boolean Equation

$\mathbf{Y = \bar{S}.I_0 + S.I_1}$

Circuit Diagram



Que 2.3. Implement the function

$F(A\ B\ C) = \sum m\ (2, 4, 7)$

i. Using 8 x 1 MUX
ii. Using 4 x 1 MUX

Solution. (i) Using 8 x 1 MUX
Given No of Variables or selection line m = 3

No of input n = 8 (**n = 2^m**)
No. of output = 1

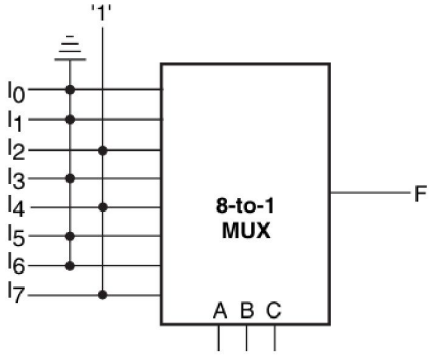
Required MUX

No of input n = 8
No. of output = 1
No of Variables or selection line m = 3
(**n = 2^m**)

Implementation table

Minterm	*A*	*B*	*C*	*f(A,B,C)*
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Logic diagram



(ii) Using 4 x 1 MUX

Given

- No of Variables or selection line m = 3
- No of input n = 8 (**n = 2^m**)
- No. of output = 1
-

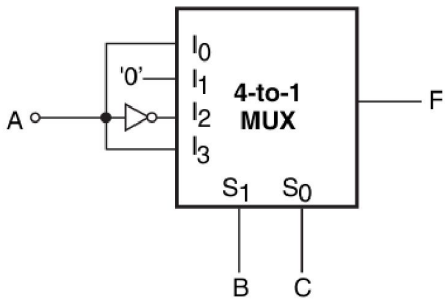
Required MUX

- No of input n = 4
- No. of output = 1
- No of Variables or selection line m = 2
  (**n = 2^m**)
- Let's Suppose '**A**' as an input and **B, C** as a selection line

Implementation table

	I_0	I_1	I_2	I_3
$\overline{A}$	0	1	**2**	3
A	**4**	5	6	**7**
	A	0	$\overline{A}$	A

Logic diagram



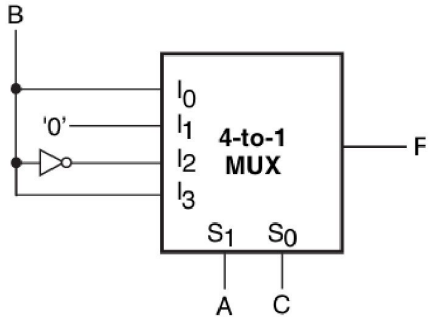
OR

- Let's Suppose '**B**' as an input and **A, C** as a selection line

Implementation table

	I_0	I_1	I_2	I_3
$\overline{B}$	0	1	**4**	5
B	**2**	3	6	**7**
	B	0	$\overline{B}$	B

Logic diagram



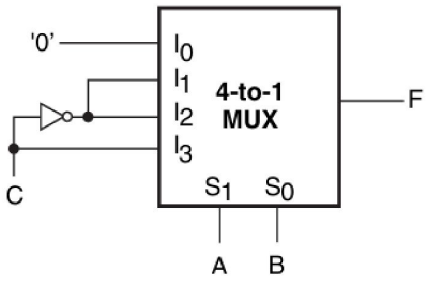
OR

- Let's Suppose '**C**' as an input and **A, B** as a selection line

Implementation table

	I_0	I_1	I_2	I_3
$\overline{C}$	0	**2**	**4**	6
C	1	3	5	**7**
	0	$\overline{C}$	$\overline{C}$	C

Logic diagram



Que 2.4. Implement the product-of-sums Boolean function expressed by

$$\mathbf{F} = \prod \mathbf{M}\,(\mathbf{1}, \mathbf{2}, \mathbf{5})$$

(i) Using 4 x 1 MUX
(ii) Using 8 x 1 MUX

Solution. We know that

$$\mathbf{F} = \prod \mathbf{M}\,(\mathbf{1}, \mathbf{2}, \mathbf{5}) = \sum \mathbf{m}\,(\mathbf{0}, \mathbf{3}, \mathbf{4}, \mathbf{6}, \mathbf{7})$$

$$\mathbf{F(A\,B\,C)} = \sum \mathbf{m}\,(\mathbf{0}, \mathbf{3}, \mathbf{4}, \mathbf{6}, \mathbf{7})$$

(i) Using 8 x 1 MUX

Given

- No of Variables or selection line m = 3
- No of input n = 8 ($\mathbf{n = 2^m}$)
- No. of output = 1

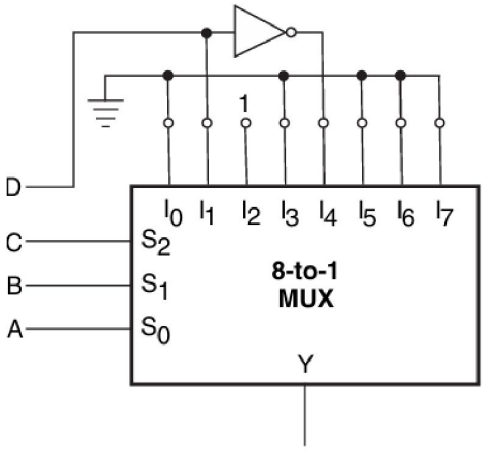
Required MUX

- No of input n = 8
- No. of output = 1
- No of Variables or selection line m = 3 ($\mathbf{n = 2^m}$)

Implementation table

C	*B*	*A*	*f(A,B,C)*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Logic diagram



(ii) Using 4 x 1 MUX

Given

- No of Variables or selection line m = 3
- No of input n = 8 (**$n = 2^m$**)
- No. of output = 1

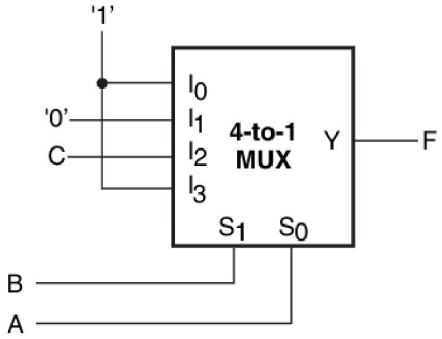
Required MUX

- No of input n = 4
- No. of output = 1
- No of Variables or selection line m = 2
  (**$n = 2^m$**)
- Let's Suppose '**C**' as an input and **B, A** as a selection line

Implementation table

	I_0	I_1	I_2	I_3
$\overline{C}$	**0**	1	2	**3**
C	**4**	5	**6**	**7**
	1	0	C	1

Logic diagram



Que 2.5. Implement the function:

$$F(A, B, C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

Using 4 : 1 multiplexer using B and C variables to the selection lines.

Solution.

$$F(A, B, C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

F(A, B, C) = Σm (1, 2, 4, 7)

Given

- No of Variables or selection line m = 3
- No of input n = 8 (**$n = 2^m$**)
- No. of output = 1

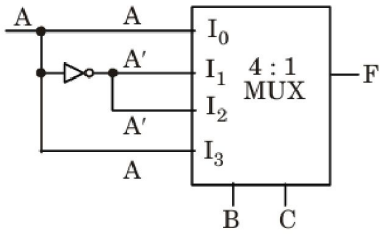
Required MUX

- No of input n = 4
- No. of output = 1
- No of Variables or selection line m = 2 (**$n = 2^m$**)
- Let's Suppose '**A**' as an input and **B, C** as a selection line

Implementation table

Input	I_0	I_1	I_2	I_3
A′	0	①	②	3
A	④	5	6	⑦
Output to multiplexer	A	A′	A′	A

Logic diagram



Que 2.6. Design the following boolean function using 4 × 1 multiplexer.

F(A, B, C, D) =Σ m(0, 1, 3, 4, 8, 9, 15)

Solution.

F (A, B, C, D) = Σ m(0, 1, 3, 4, 8, 9, 15)

Given

- No of Variables or selection line m = 4
- No of input n = 16 (**$n = 2^m$**)
- No. of output = 1

Required MUX

- No of input n = 4
- No. of output = 1
- No of Variables or selection line m = 2

  (**n = 2^m**)
- Let's Suppose **C, D** as an input and **A, B** as a selection line

Implementation table

	I_0	I_1	I_2	I_3
$\overline{C}\,\overline{D}$	(0)	(4)	(8)	12
$\overline{C}D$	(1)	5	(9)	13
$C\overline{D}$	2	6	10	14
CD	(3)	7	11	(15)
	$\overline{C} + D$	$\overline{C}\,\overline{D}$	$\overline{C}$	CD

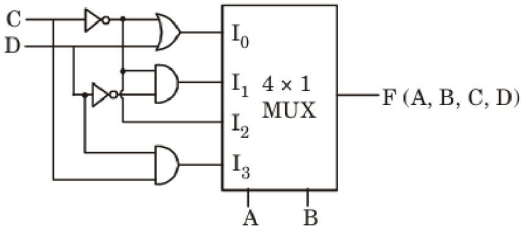
$I_0 = \overline{C}\,\overline{D} + \overline{C}D + CD = \overline{C} + D$

$I_1 = \overline{C}\,\overline{D}$

$I_2 = \overline{C}\,\overline{D} + \overline{C}D = \overline{C}$

$I_3 = CD$

Logic diagram



Que 2.7. Implement the following Boolean function.

F(A, B, C, D) = Σ(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)

(i) 4 : 1 MUX

(ii) 2 : 1 MUX

(iii)

Solution. (i) 4 : 1 MUX

F(A, B, C, D) = Σ(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)

Given

- No of Variables or selection line m = 4
- No of input n = 16 (**n = 2^m**)
- No. of output = 1

Required MUX

- No of input n = 4
- No. of output = 1
- No of Variables or selection line m = 2

  (**n = 2^m**)
- Let's Suppose **C, D** as an input and **A, B** as a

selection line

Implementation table

		$\bar{A}\bar{B}$ I_0	$\bar{A}B$ I_1	$A\bar{B}$ I_2	AB I_3
(00)	$\bar{C}\bar{D}$	(0)	(4)	(8)	12
(01)	$\bar{C}D$	(1)	5	(9)	13
(10)	$C\bar{D}$	2	6	10	(14)
(11)	CD	(3)	(7)	(11)	(15)

First column (I_0)

$$= \bar{C}\,\bar{D} + \bar{C}D + CD$$

$$= \bar{C}\,(\bar{D} + D) + CD \;=\; \bar{C} + CD$$

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

Second column (I_1)

$$= \bar{C}\bar{D} + \bar{C}D + CD$$

Third column (I_2)

$$= \bar{C}\bar{D} + \bar{C}D + CD$$

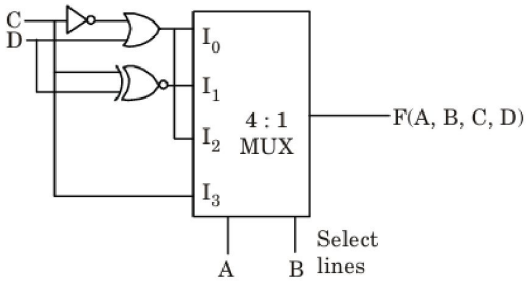
$$= \bar{C}(\bar{D} + D) + CD$$

Fourth column (I_3)

$$= C\bar{D} + CD$$

$$= C(\bar{D} + D) \;= C$$

Logic diagram



(ii) 2 : 1 MUX

F(A, B, C, D) = Σ(0, 1, 3, 4, 7, 8, 9, 11, 14, 15)

Given

- No of Variables or selection line m = 3
- No of input n = 8 (**$n = 2^m$**)
- No. of output = 1

Required MUX

- No of input n = 2
- No. of output = 1
- No of Variables or selection line m = 1

(**$n = 2^m$**)

- Let's Suppose **A, B, C** as an input and **D** as a selection line

Implementation table

	$\overline{D}$ I_0	D I_1
$\overline{A}\overline{B}\overline{C}$	(0)	(1)
$\overline{A}\overline{B}C$	2	(3)
$\overline{A}B\overline{C}$	(4)	5
$\overline{A}BC$	6	(7)
$A\overline{B}\overline{C}$	(8)	(9)
$A\overline{B}C$	10	(11)
$AB\overline{C}$	12	13
ABC	(14)	(15)

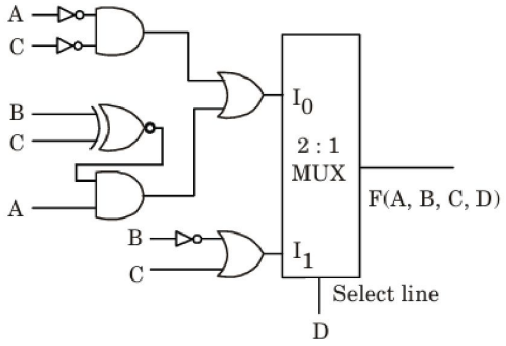
First column (I_0)

$$= \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC$$
$$= \overline{A}\overline{C}(\overline{B} + \overline{B}) + A(\overline{B}\overline{C} + BC)$$
$$= \overline{A}\overline{C} + A(B \odot C)$$

Second column (I_1)

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

Logic diagram



Que 2.8. Design the following boolean function using the multiplexer:

F(A, B, C, D) = Σ m (0, 3, 5, 6, 8, 9, 14, 15).

Solution.

F(A, B, C, D) = Σ m (0, 3, 5, 6, 8, 9, 14, 15)

Given

- No of Variables or selection line m = 4
- No of input n = 16 (**$n = 2^m$**)
- No. of output = 1

Required MUX Lets 8 X 1

- No of input n = 8
- No. of output = 1
- No of Variables or selection line m = 3

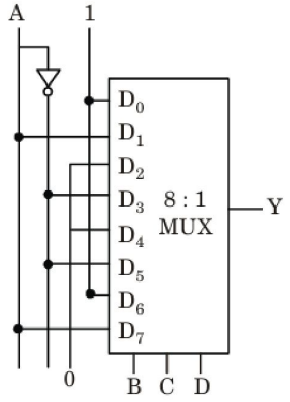
(**$n = 2^m$**)

- Let's Suppose '**A**' as an input and **B, C,** and **D** as a selection line

Implementation table

	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$\overline{A}$	(0)	1	2	(3)	4	(5)	(6)	7
A	(8)	(9)	10	11	12	13	(14)	(15)
	1	A	0	$\overline{A}$	0	$\overline{A}$	1	A

Logic diagram



Que 2.9. Implement the function F = Σm(0, 1, 3, 4, 7, 8, 9, 11, 14, 15) using 8:1 mux.

Solution.

F = Σm (0, 1, 3, 4, 7, 8, 9, 11, 14, 15)

Given

- No of Variables or selection line m = 4
- No of input n = 16 (**$n = 2^m$**)
- No. of output = 1

Required MUX

- No of input n = 8
- No. of output = 1
- No of Variables or selection line m = 3

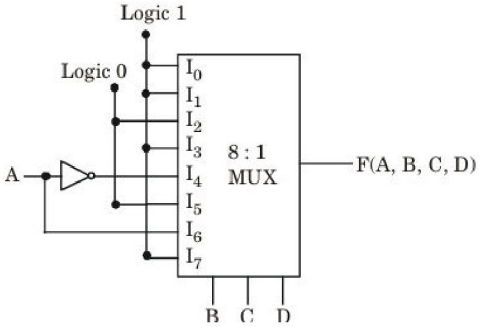
(**$n = 2^m$**)

- Let's Suppose '**A**' as an input and **B, C** and **D** as a selection line

Implementation table

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\overline{A}$	0	1	2	3	4	5	6	7
A	8	9	10	11	12	13	14	15
	1	1	0	1	$\overline{A}$	0	A	1

Logic diagram



Que 2.10. Construct a 16 × 1 multiplexer with two 8 × 1 and one 2 × 1 multiplexer. Use block diagrams.

Solution.

Given

- No of input n = 16
- No. of output = 1
- No of Variables or selection line m = 4

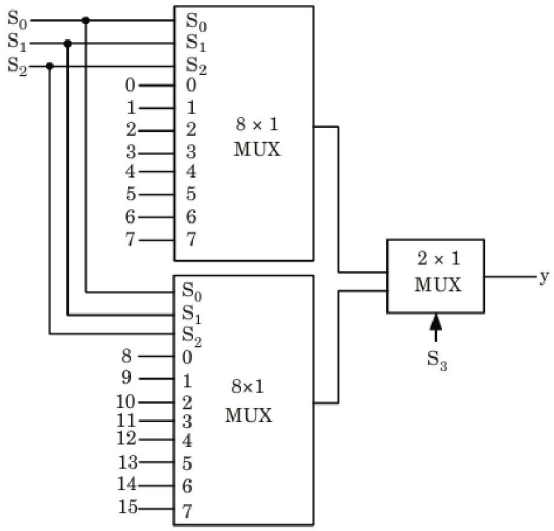
($\mathbf{n = 2^m}$)

Required MUX

- No of input n = 8
- No. of output = 1
- No of Variables or selection line m = 3

($\mathbf{n = 2^m}$)

Logic diagram



Que 2.11. What do you mean by encoder?

Solution.

In digital electronics, an encoder is a combinational logic circuit that converts a set of input signals into a binary code, essentially representing which input is active at a given time. It's a device that transforms information from one format to another, often from a larger number of input lines to a smaller number of output lines.

- **Converts inputs to binary code:**

Encoders take a set of input signals and produce a corresponding binary output code.

- **Combinational logic circuit:**

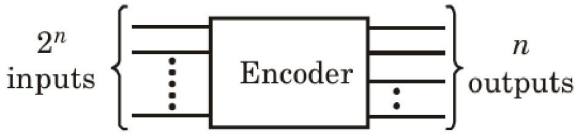
Encoders are built using logic gates (AND, OR, NOT, etc.) and do not require memory elements.

- **Common types:**

Examples include 4-to-2 encoders, 8-to-3 encoders (also known as octal encoders), and decimal-to-BCD encoders.

- **Applications:**

Encoders are used in various digital systems, including data transmission, multiplexing, and controlling.

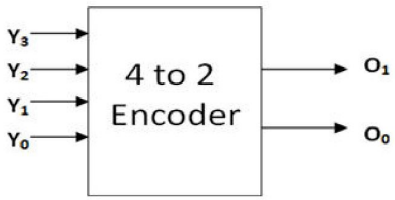


4x2 Encoder

No of input $2^n = 4$

No. of output $n = 2$

Block diagram



Truth table

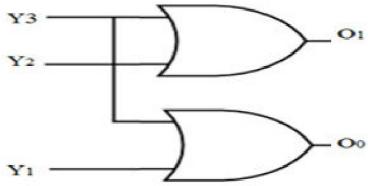
Y_3	Y_2	Y_1	Y_0	O_1	O_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Boolean Equation

$O_1=Y_2+Y_3$,

$O_0=Y_1+Y_3$

Circuit Diagram



Que 2.12. Write a short note on decoder.
Solution.
In digital electronics, a decoder is a combinational logic circuit that converts a binary code (or other coded input) into a specific output signal, activating only one output line for each unique input combination. Essentially, it translates a coded input into a readable, decoded output.

- **Input:**

Decoders receive binary or coded input signals. These inputs represent different combinations of data.

- **Decoding Process:**

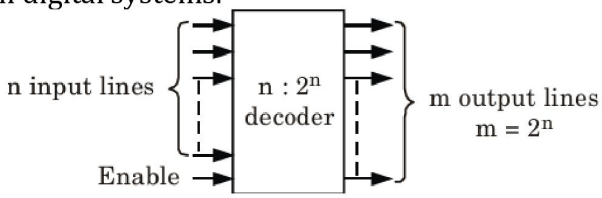
The decoder's logic circuitry is designed to recognize each unique input combination and activate a corresponding output line.

- **Output:**

The decoder produces a unique output signal for each valid input combination, effectively "decoding" the input.

- **Purpose:**

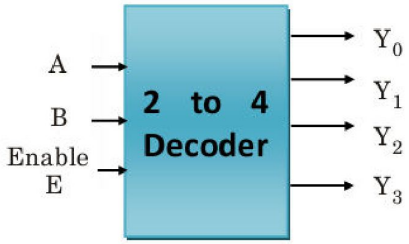
Decoders are used to select specific memory locations, control data routing, and perform various other tasks in digital systems.



2 to 4 binary decoder:

No of input n = 2
No. of output m = $2^n = 4$

Block diagram



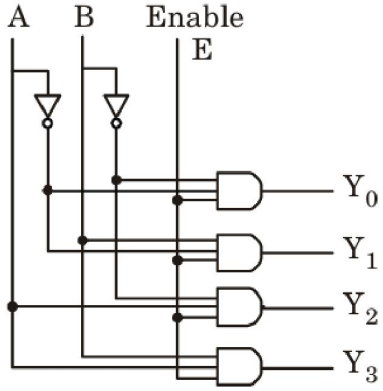
Truth table

Inputs			Outputs			
E	A	B	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

Boolean Equation

Y_0 = E A' B'
Y_1 = E A' B
Y_2 = E A B'
Y_3 = E A B

Circuit Diagram



Que 2.13. Using a decoder and external gates, design the combinational circuit defined by the following three Boolean functions:

$$F_1 = x'yz' + xz,$$

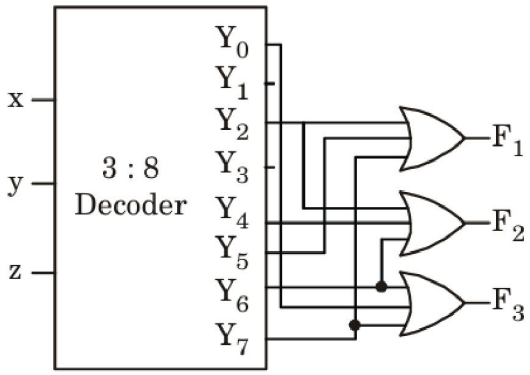
$$F_2 = xy'z' + yz',$$

$$F_3 = x'y'z' + xy$$

Solution.

No of input n = 3
No. of output m = $2^n = 8$

$$F_1 = x'yz' + xz = x'yz' + xz(y + y')$$
$$= x'yz' + xyz + xy'z = \Sigma m(2, 7, 5)$$
$$F_2 = xy'z' + yz' = xy'z' + (x + x')yz'$$
$$= xy'z' + xyz' + x'yz' = \Sigma m(2, 4, 6)$$
$$F_3 = x'y'z' + xy = x'y'z' + xy(z + z')$$
$$= x'y'z' + xyz + xyz' = \Sigma m(0, 6, 7)$$



Que 2.14. Draw the logic diagram of a two-to-four-line decoder using NOR gates only.
Solution.
Truth table:

Enable	Input		Output			
E	*A*	*B*	Y_3	Y_2	Y_1	Y_0
0	×	×	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

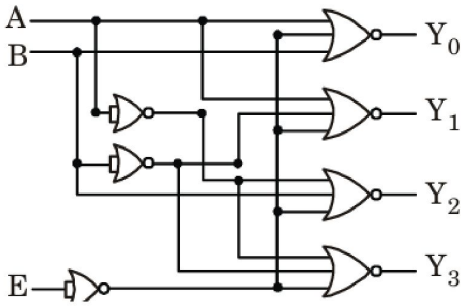
$$Y_0 = \overline{(A + B + \overline{E})} = \overline{A}\,\overline{B}\,E$$

$$Y_1 = \overline{(A + \overline{B} + \overline{E})} = \overline{A}\,B\,E$$

$$Y_2 = \overline{(\overline{A} + B + \overline{E})} = A\,\overline{B}\,E$$

$$Y_3 = \overline{(\overline{A} + \overline{B} + \overline{E})} = A\,B\,E$$

Circuit using NOR gate:



Que 2.15. Design a full subtractor circuit with a decoder and two OR gates.
Solution.
Full subtractor using decoder
Truth table:

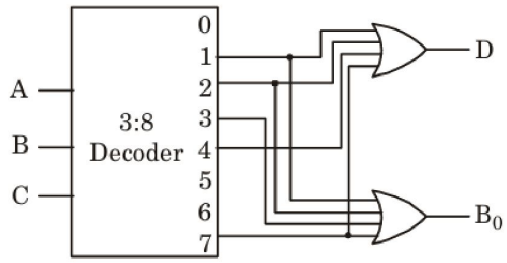
Input			Output	
A	*B*	*C*	*D*	B_0
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Boolean Equation

Difference: $D = \Sigma\, m\,(1, 2, 4, 7)$

Borrow: $B_0 = \Sigma\, m\,(1, 2, 3, 7)$

Decoder using OR gates



Que 2.16. Describe half adder and full adder in brief. Implement the circuit using logic gates.

OR

Design a full adder using two half adders.

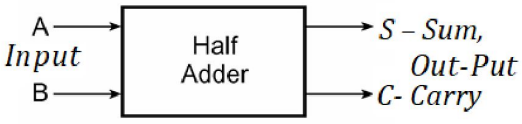
Solution.

ADDERS,

- Half Adder
- Full Adder

Half Adder

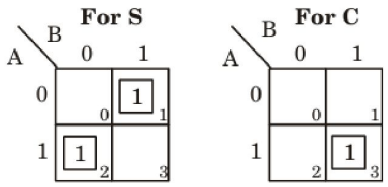
A half adder is a digital logic circuit that adds two single-bit binary numbers, producing a sum and a carry output. It has two input bit (A B) and two output (S – Sum, C- Carry)



If there are two input variables the combinations are four ($2^2 = 4$). Now form the truth table of the half adder.

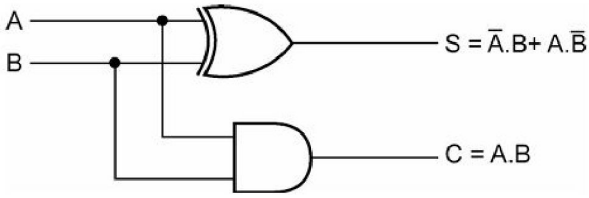
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Using two-variable K-map

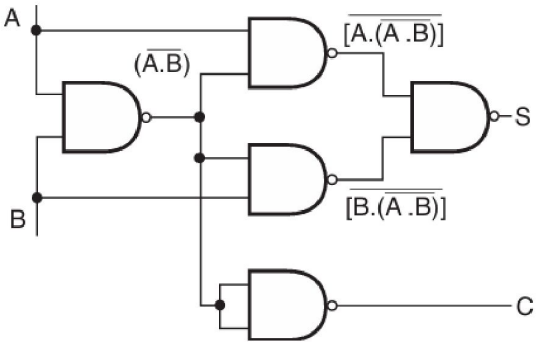


$$S = A\overline{B} + \overline{A}B = A \oplus B$$
$$C = AB$$

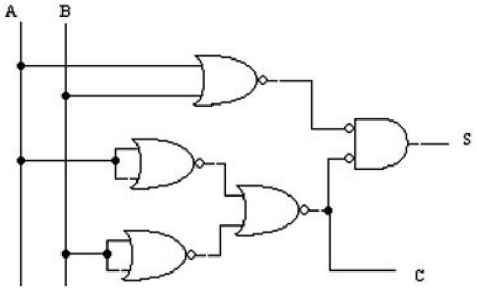
Logic implementation of a half-adder.



Half-adder implementation using NAND gates.



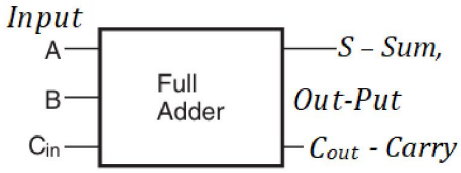
Half-adder implementation using NOR gates.



Full Adder

A full adder is a combinational logic circuit that adds three one-bit binary numbers, producing a sum and a carry-out bit.

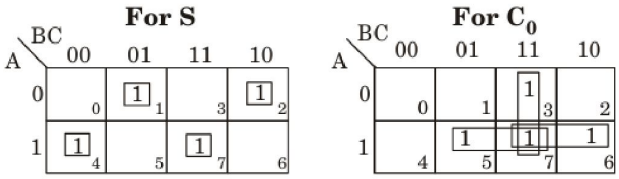
- It takes two input bits (A and B) and a carry-in bit (C_{in}) from a previous stage,
- Outputs a sum bit (S) and a carry-out bit (Cout).
- Full adders are essential components in digital circuits, particularly for multi-bit binary addition.



If there are three input variables the combinations are eight ($2^3 = 8$). Now form the truth table of the full adder.

A	B	C_{in}	SUM (S)	C_{out}
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

Using three-variable K-map



$$S = ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$$
$$C_o = AB + AC + BC$$

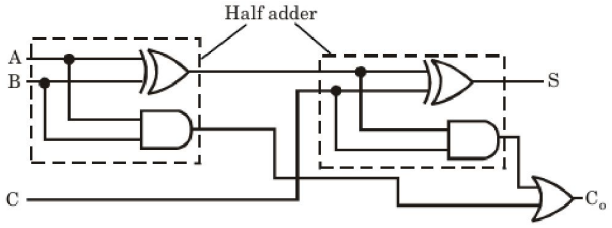
A full adder can be implemented using two half adders and one OR gate.
Sum:

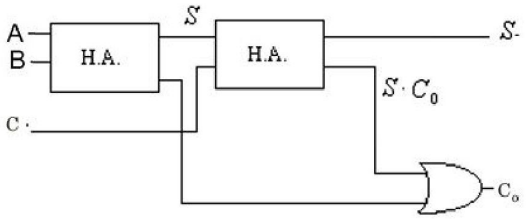
$$S = ABC + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$$
$$= ABC + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C}$$
$$= C(AB + \bar{A}\bar{B}) + \bar{C}(A\bar{B} + \bar{A}B)$$
$$= C\overline{(A\bar{B} + \bar{A}B)} + \bar{C}(A\bar{B} + \bar{A}B)$$
$$= (A \oplus B) \oplus C$$

Carry:

$$
\begin{aligned}
C_o &= AB + AC + BC \\
&= AB + C(A + B) \\
&= AB + C(A + B)(A + \bar{A})(B + \bar{B}) \\
&= AB + C[AB + A\bar{B} + \bar{A}B] \\
&= AB + ABC + C(A\bar{B} + \bar{A}B) \\
&= AB(1 + C) + C(A \oplus B) \\
&= AB + C(A \oplus B)
\end{aligned}
$$

Logic implementation of a half-adder.

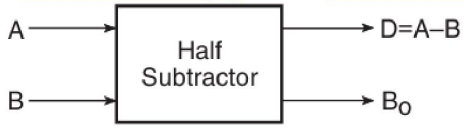




Que 2.17. Describe a half subtractor with its logic diagram.

Solution. Half Subtractor

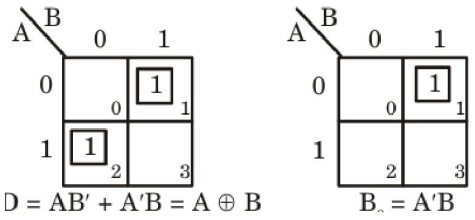
A half subtractor is a combinational logic circuit that performs the subtraction of two binary digits (bits). It takes two input bits, a minuend (A) and a subtrahend (B), and produces two output bits: a difference bit (D) and a borrow bit (Bout).



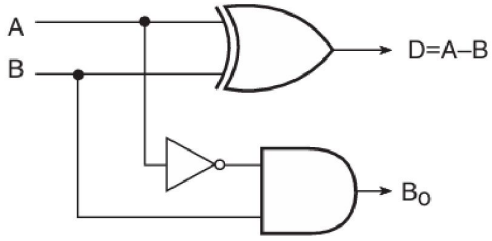
If there are two input variables the combinations are four ($2^2 = 4$). Now form the truth table of the half subtractor.

Inputs		Outputs	
A	***B***	***D***	B_0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

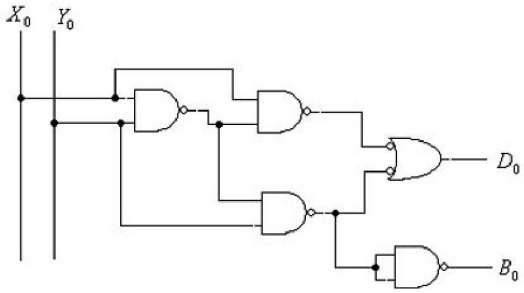
Using three-variable K-map



The logical implementation using basic logic gates and XOR gate:



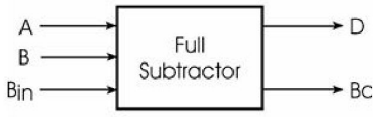
Half Subtractor using NAND gates



Que 2.18. Design a full subtractor circuit with three inputs x, y, Bin and two outputs Diff and B_{out} . The circuit subtracts x – y – B_{in} , where, Bin is the input borrow, Bout is the output borrow and Diff is the difference.

Solution.

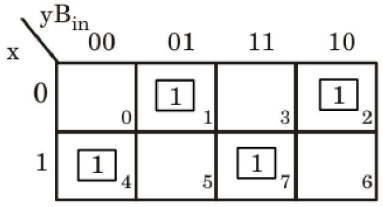
A full subtractor is a combinational logic circuit that performs subtraction of two binary digits, considering a borrow input from the previous stage. It takes three inputs: the minuend bit (A), the subtrahend bit (B), and a borrow-in bit (Bin) from the preceding stage. It produces two outputs: a difference bit (D) and a borrow-out bit (Bout).



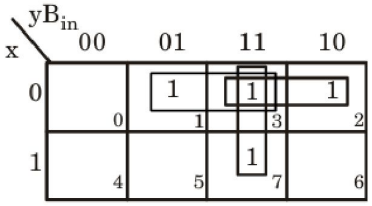
If there are three input variables the combinations are eight ($2^3 = 8$). Now form the truth table of the full adder.

Inputs			Outputs	
x	y	B_{in}	Diff	B_{out}
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

Using three-variable K-map



$$\text{Diff} = \bar{x}\,\bar{y}\,B_{in} + \bar{x}\,y\,\bar{B}_{in} + xyB_{in} + x\bar{y}\bar{B}_{in}$$

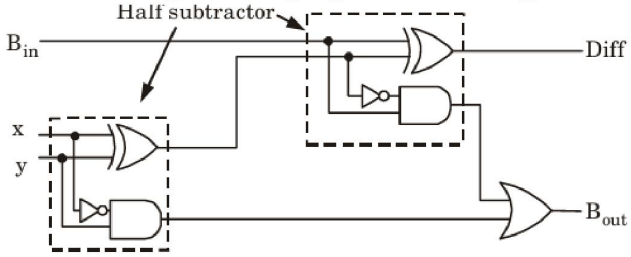


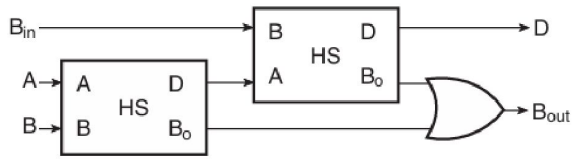
$$B_{out} = \bar{x}\,B_{in} + yB_{in} + \bar{x}y$$

A full subtractor can also be implemented using two half subtractors and an OR gate.

$$\begin{aligned}
\text{Diff} &= xyB_{in} + x\bar{y}\,\bar{B}_{in} + \bar{x}y\,\bar{B}_{in} + \bar{x}\,\bar{y}B_{in} \\
&= B_{in}(xy + \bar{x}\,\bar{y}) + \bar{B}_{in}(x\,\bar{y} + \bar{x}\,y) \\
&= B_{in}\overline{(x \oplus y)} + \bar{B}_{in}(x \oplus y) = (x \oplus y) \oplus B_{in} \\
B_{out} &= \bar{x}\,y + \bar{x}\,B_{in} + yB_{in} = \bar{x}\,y + B_{in}(\bar{x} + y) \\
&= \bar{x}\,y + B_{in}(\bar{x} + y)(x + \bar{x})(y + \bar{y}) \\
&= \bar{x}y + B_{in}(\bar{x}y + xy + \bar{x}\,\bar{y}) = \bar{x}y + \bar{x}yB_{in} + B_{in}(xy + \bar{x}\,\bar{y}) \\
&= \bar{x}\,y(B_{in} + 1) + B_{in}\overline{(x \oplus y)} = \bar{x}\,y + B_{in}\,\overline{(x \oplus y)}
\end{aligned}$$

The logical implementation using basic logic gates and XOR gate:





Full subtractor circuit using NAND gate.

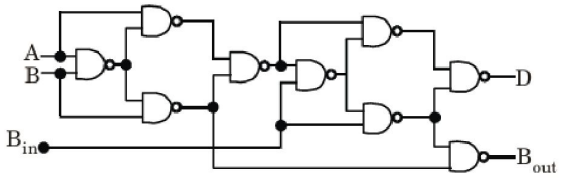
$$= A \oplus B \oplus B_{in} = \overline{\overline{(A \oplus B)(\overline{A \oplus B})B_{in}}\ \overline{B_{in}(\overline{A \oplus B})B_{in}}}$$

$$B_{out} = \bar{A}B + B_{in}\overline{(A \oplus B)} = \overline{\overline{\bar{A}B + B_{in}(\overline{A \oplus B})}}$$

$$= \overline{\overline{\bar{A}B}\,.\,\overline{B_{in}(\overline{A \oplus B})}} = \overline{\overline{B(\bar{A} + \bar{B})}\ \overline{B_{in}\left[\overline{\bar{B}_{in} + (\overline{A \oplus B})}\right]}}$$

$$B_{out} = \overline{\overline{B\,\bar{A}\bar{B}}\ \overline{B_{in}\left[B_{in}\,(\overline{A \oplus B})\right]}}$$

By using the above expressions for D and B_{out}, the full subtractor is implemented using only NAND gates



Que 2.19. Design a 4-bit binary parallel adder.

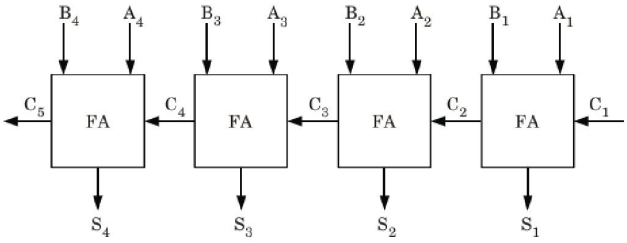
Solution. 4-BIT BINARY PARALLEL ADDER

A 4-bit parallel adder is a digital circuit that adds two 4-bit binary numbers simultaneously. It uses four full adders connected in series, where each full adder processes a pair of bits from the input numbers along with the carry-in from the previous stage. The output is a 4-bit sum and a carry-out.

C_3	C_2	C_1	C_0	
	A_3	**A_2**	**A_1**	**A_0**
	B_3	**B_2**	**B_1**	**B_0**
	S_3	S_2	S_1	S_0

- The sum of two n-bit binary numbers, A and B, can be generated in two ways: either in a serial fashion or in parallel.
- The series addition method uses only one full-adder circuit and a storage device to hold the generated output carry.
- The pair of bits in A and B are transferred serially, one at a time, through the single full-adder to produce a string of output bits for the sum.
- The stored output carry from one pair of bits is used as an input carry for the next pair of bits.
- The parallel method uses n full-adder circuits, and all bits of A and B are applied simultaneously.
- A binary parallel adder is a digital function that produces the arithmetic sum of two binary numbers in parallel.

- It consists of full-adders connected in cascade, with the output carry from one full-adder connected to the input carry of the next full-adder.



Que 2.19. Draw a BCD adder circuit and explain its working.

OR

Draw a decimal adder to add BCD number.

Solution. BCD ADDER

BCD adder is circuit that adds two BCD digits in parallel and produces a sum digit which is also BCD. BCD numbers use 10 symbols (group of 4 bits 0000 to 1001). BCD adder circuit must be able to do the following and it is shown in

- Add two 4-bit BCD numbers using straight binary addition.
- If 4-bit sum is equal to or less than 9, the sum is a valid BCD number and no correction is needed.
- If the 4-bit sum is greater than 9 or if a carry is generated from the sum, the sum is invalid BCD number. Then the digit 6 $(0110)_2$ should be added to the sum to produce the valid BCD symbols.