DIGITAL LOGIC DESIGEN

TRUTH TABLE FOR ADDITION [BINARY]

A	B	O/P
0	1	1
1	0	1
0	0	0
1	1	0 (carry 1)

augen = jisme plus karna hota hai.

adderd = jisko plus karte hai.

TRUTH TABLE FOR SUBTRACTION -

A	B	O/P
0	0	0
0	1	1 (Borrow of 1)
1	0	1
1	1	0

minuend = jisme se minus karte hai.

subtraherd = jisko minus karte hai.

Binary to decimal

Number . Number

$x(2^0 + \cdots 2^n) \; x(2^{-1} + \cdots 2^{-n})$

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

Decimal to Binary

For Exp

	128	64	32	16	8	4	2	1
18 = Num =	0	0	0	1	0	0	1	0

jinke sum se number bna hota hai unki Bits high kardo !!

Scanned with OKEN Scanner

complements -

- Diminished radix complement
  {(r-1)'s complement}
- Radix complement {(r)'s complement}

For decimal Numbers

9's complement & 10's complement
Diminished Redix Redix

for 9's completment
Subtract the number from 999.... 9

For 10's complement
1. Find the 9's complement and then add 1 on that.

For Binary numbers -

- 1's complement & 2's complement

for 1's complement -
Subtract the number from 111.... 1

trick
Replace all the zeros from 1 & all the ones from 0

For 2's complement -
- Add 1 in 1's complement

Subtraction with complement -

① using 10's complement - [For decimal numbers]

(i) Find the 10 complement of subtrahend
(ii) Add 10's complement of subtrahend with Minuend.
(iii) (1) If M > N carry is discarded.
Minuend
(2) If M < N take 10's complement of sum and place a negetive sign.

② using 2's complement - [For Binary numbers]

(i) Find 2's complement of subtrahend.
(ii) Add 2's complement and Minuend.
(iii) - If sum contain 1 more bit than minuend and subtrahend have discard it
- If not, but MSB is 1, result is -ve and 2's complement of sum

③ 1's complement { Practicall implement 2's complement }

(i) Take 1's complement of subtrahend
(ii) Add minuend & 1's complement of subtrahend

Scanned with OKEN Scanner

(iii) • If Sum contain 1 more bit than minuend and subtrahend Add that in Last significant bit

• If no carry over but sum contain MSB as 1. then Result will be -ve and 1's complement of sum.

Signed Binary Numbers -

Acc to LMB 0 → +ve 1 → -ve

8	4	2	1	
0	0	1	1	→ unsigned = 8x1 + 0x4 + 1x2 + 1x1 = $(11)_{10}$
1	0	1	1	→ signed

Only indicates sign → 1011

↳ 1x1

↳ 1x2

$= (-3)_{10}$

BCD Addition -

- when the Binary sum ≤ 1001, (without carry), the corresponding BCD is correct.
- when the Binary sum is greater than (>) 1010, the Result is invalid. The addition of $(0110)_2 = [6]_{10}$ to the Binary sum converts it to the correct

BCD digit and also produces a carry as Required.

Excess - 3 code

- made up of BCD + 0011.

→ Self Complementry

→ Not weighted code

matlab decimal mein vo number nhi hota.

Grey codes

$n=1$	$n=2$	$n=3$
(2^1 code)	(2^2 code)	(2^3 code)
0	0 0	0 0 0
1	0 1	0 0 1
	1 1	0 1 1
	1 0	0 1 0
		1 1 0
		1 1 1
		1 0 1
		1 0 0

Binary to grey

$(9)_{10} = (1001)_2$

MSB

1 1 0 1

{Carry aya to discard Kardo}

grey to Binary

Code = (1011)

MSB

1 1 0 1

↓

Carry (1) {discard}

Boolean algebra and logic gates

Table - postulate and theorems

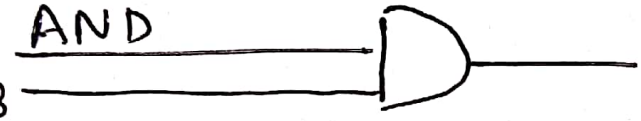
	(a)	(b)
• Postulate 2	$x + 0 = x$	$x \cdot 1 = x$
• Postulate 5	$x + x' = 1$	$x \cdot x' = 0$
• Theorem 1	$x + x = x$	$x \cdot x = x$
• Theorem 2	$x + 1 = 1$	$x \cdot 0 = 0$
• Theorem 3, (involution)	$(x')' = x$	$(x')' = x$
• Postulate 3 (commutative)	$x + y = y + x$	$x \cdot y = y \cdot x$
• Theorem 4, (associative)	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
• Postulate 4, (distributive)	$x \cdot (y + z) = xy + xz$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
• Theorem 5, (De-morgan)	$(x + y)' = x'y'$	$(x \cdot y)' = x' + y'$

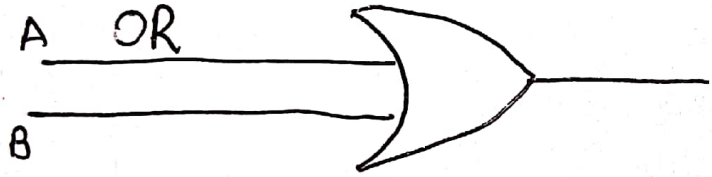
Scanned with OKEN Scanner

- Theorem 6, absorption | $x + xy = x$ | $x \cdot (x+y) = x$

Logic Gates

1) NOT  $A \rightarrow A'$

2) A, B AND  $A \cdot B$

3) A, B OR  $A + B$

Canonical or standard form :

A	B	C	Min terms (m)	Max terms (M)
0	0	0	(m_0) $A'B'C'$	(M_0) $A + B + C$
0	0	1	(m_1) $A'B'C$	(M_1) $A + B + C'$
0	1	0	(m_2) $A'BC'$	(M_2) $A + B' + C$
0	1	1	(m_3) $A'BC$	(M_3) $A + B' + C'$
1	0	0	(m_4) $AB'C'$	(M_4) $A' + B + C$
1	0	1	(m_5) $AB'C$	(M_5) $A' + B + C'$
1	1	0	(m_6) ABC'	(M_6) $A' + B' + C$
1	1	1	(m_7) ABC	(M_7) $A' + B' + C'$

Scanned with OKEN Scanner

Product = 1 (minterm) → and

Sum = 0 (Maxterm) → OR

f → SoP (Sum of Product)

F → Product of sum (POS)

OTHER LOGIC FUNCTIONS –

no. of variables = n

no. of Boolean functions = 2^{2^n}

Scanned with OKEN Scanner

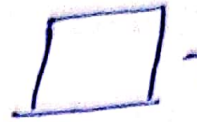
UNIT - 2

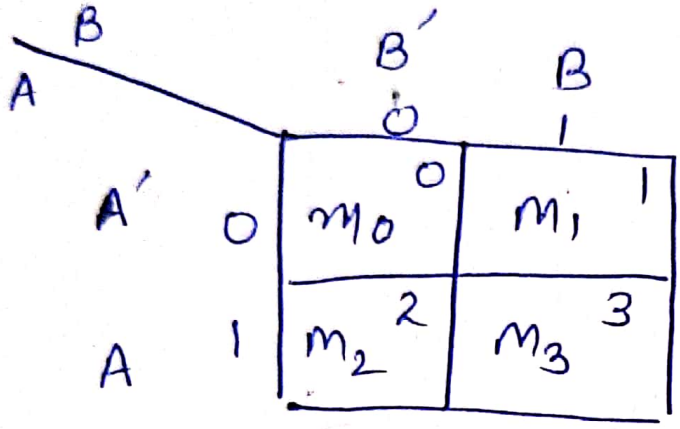
Karnaugh map method

2-variable K-map method

NO. of variables = 2

NO. of cells = $2^2 = 4$

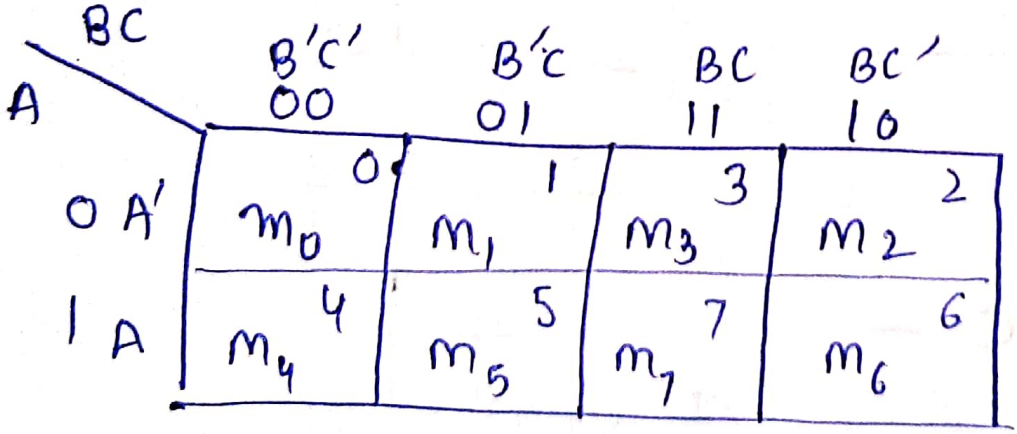




3-variable K-map method

NO. of variable = 3

NO. of cells = $2^3 = 8$



{ ER baar mein ER hi bit change hogi }

Dr. Balram Sharma

4- variable K- map –

CD \ AB	A'B' (00)	A'B (01)	AB (11)	AB' (10)
CD (00)	0 m_0	1 m_1	3 m_3	2 m_2
C'D (01)	4 m_4	5 m_5	7 m_7	6 m_6
CD (11)	12 m_{12}	13 m_{13}	15 m_{15}	14 m_{14}
CD' (10)	8 m_8	9 m_9	11 m_{11}	10 m_{10}

5-variable K-map

A = 0

DE \ BC	B'C' (00)	B'C (01)	BC (11)	BC' (10)
D'E' 00	0 m_0	1 m_1	3 m_3	2 m_2
D'E 01	4 m_4	5 m_5	7 m_7	6 m_6
DE 11	12 m_{12}	13 m_{13}	15 m_{15}	14 m_{14}
DE' 10	8 m_8	9 m_9	11 m_{11}	10 m_{10}

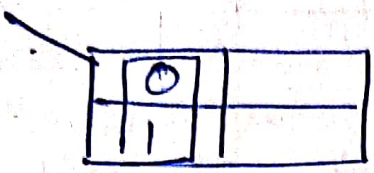
A = 1

DE \ BC	B'C' (00)	B'C (01)	BC (11)	BC' (10)
D'E' 00	16 m_{16}	17 m_{17}	19 m_{19}	18 m_{18}
D'E 01	20 m_{20}	21 m_{21}	23 m_{23}	22 m_{22}
DE 11	28 m_{28}	29 m_{29}	31 m_{31}	30 m_{30}
DE' 10	24 m_{24}	25 m_{25}	27 m_{27}	26 m_{26}

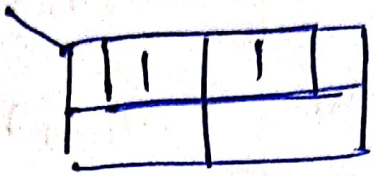
Scanned with OKEN Scanner

Rules for minimization of SOP's

(1) Groups may not include any cell containing a "0"

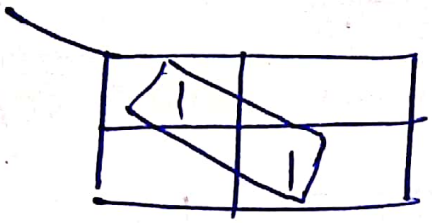


X

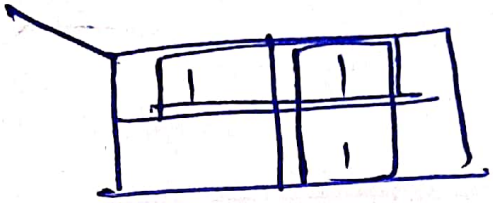


✓

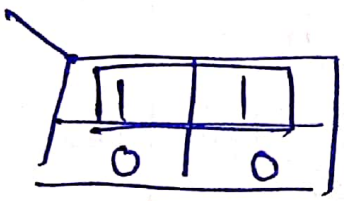
(2) Groups may not be horizontal or vertical but not diagnol



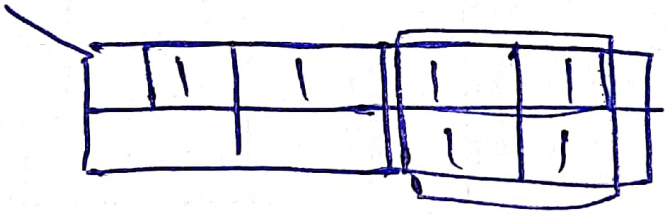
X



(3) groups must contain 1, 2, 4, 8 or in general 2^n cells.

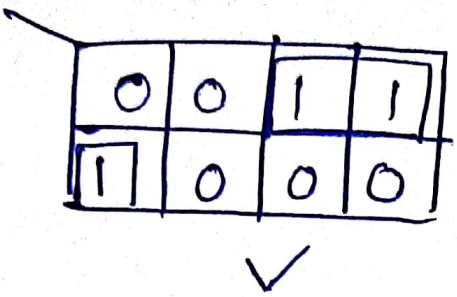


✓

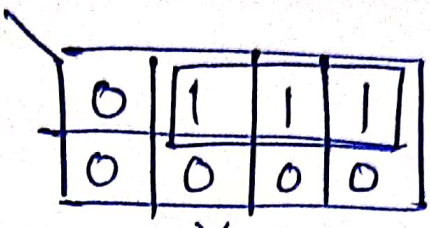


✓

Scanned with OKEN Scanner

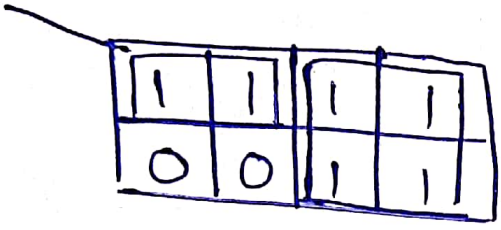


✓

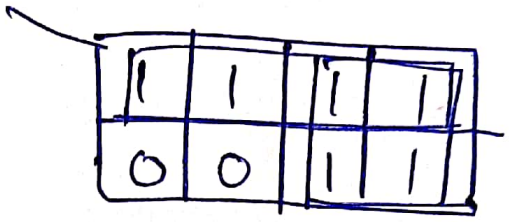


X

(4) Each group should be as large as possible

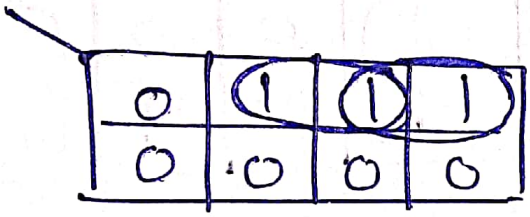


X

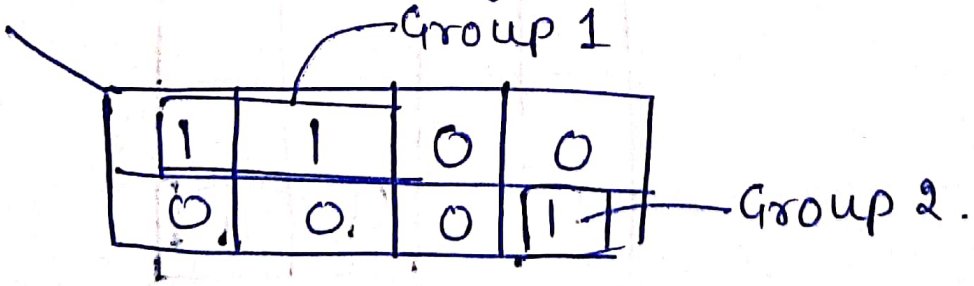


✓

(5) Groups may overlap.

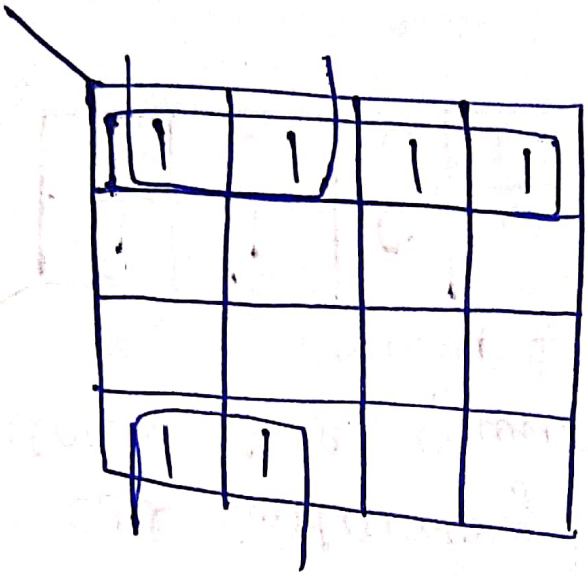


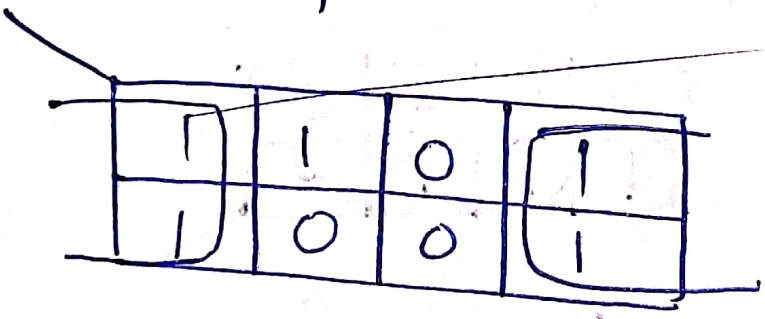
(6) Each cell containing a "1" must be in atleast one group.



Scanned with OKEN Scanner

(7) Groups may wrap around the corners the table





Scanned with OKEN Scanner

Dr. Balram Sharma

Trick For 2 complement

Step 1 : Write down given number

Step 2 : start from LSB, copy all zeros till first 1.

Step 3 : copy the first 1.

Step 4 Complement Remainning bits

Ex 1011000

Complement ↓ ↓

1101000

don't care condition -

Represented by (x or d)

→ inko pairing me le skte ho if Requirement hai pr agar need nhi hai to mat lo !!

Prime Implicants (largest possible groups of 1's)

- Essential [necessary hoti hai, groups bnane ke lie.]
- non Essential [necessary nhi hoti bs aise hi mje mein groups bna dete hai, na bhi bnae to koi fark nhi padega]

Scanned with OKEN Scanner