DIGITAL ORGANISATION
UNIT-I

DATA TYPES:- Digital Computers में Binary Information को Registers में Store किया जाता हैं। Registers में data या Control Information को Store किया जाता हैं। Control Information bit व bits का ग्रुप हैं जिसे अन्य Registers पर डाटा Manipulation के लिये Command Signals के रूप में प्रदर्शित किया जाता हैं। Data Binary Number या Numbers हैं जिसे Results के लिये उपयोग में लिया जाता हैं। digital Computers में Register Data Types को निम्न श्रेणी में विभक्त किया गया हैः–

(1) Numbers जिन्हें Arithmetic Operation के लिये उपयोग में लिया जाता हैं।
(2) Data Processing में उपयोग में लिये जाने वाले Letters
(3) अन्य Special Symbols

Binary Numbers के अलावा सभी प्रकार के डाटा को Binary code form में Register में प्रदर्शित किया जाता हैं

Number System:- हर Number System को उसके आधार के अनुसार पहचाना जाता हैं जिसे Radix कहते हैं। जैसे decimal Numbers System हमेशा Base 10 को उपयोग में लेता हैं। जैसे यदि (24.5) को Represent करना हैं तो निम्न प्रकार किया जा सकता हैंः–

$7*10^2+2*10^1+4*10^0+5*10^{-1}$

Binary:- इसी प्रकार Binary Number System Base 2 को उपयोग में लेता हैं। इन दो digits को 0 या 1 द्वारा प्रदर्शित किया जाता हैं। इन digits को प्रदर्शित करने के लिये निम्न नियम उपयोग में लिये जाते हैंः– $(101101)_2$

$1*2^5+0*2^4+1*2^3+1*2^2+0*2^1+1*2^0 = 45$

Octal:- इसी प्रकार Octal Number को प्रदर्शित करने के लिये Base 8 को उपयोग में लिया जाता हैं। जैसे यदि $(736.4)_8$ को Decimal Numbers में प्रदर्शित करना हैंः–

$(736.4)_8 = 7*8^2+3*8^1+6*8^0+4*8^{-1}$

Hexadecimal:- Hexadecimal Numbers को प्रदर्शित करने के लिये Base 16 को उपयोग में लिया जाता हैं। जैसे यदि $(F3)_{16}$ को decimal में प्रदर्शित करना हो तोः–

$(F3)_{16} = F*16+3 = 15*16+3 = (243)_{10}$

Number System में मुख्य Systems इस प्रकार हैंः–

(1) Decimal to Binary:- किसी भी Decimal Numbers की Binary (0,1) में परिवर्तित करने के लिये उसमें 2 का भाग दिया जाता हैं जब तक कि शेषफल 1 न रह जाएः–

जैसे $(13)_{10}$ को यदि Binary में परिवर्तित करना होः–

2	13	1
2	6	0
2	3	1
	1	

$(1101)_2$

अतः $(13)_{10} = (1101)_2$ हैं।

यदि $(0.65625)_{10}$ को Binary में बदलना हैं तो

0.65625	1.31250	0.62500	0.25000	0.50000
x2	x2	x2	x2	x2
1.31250	0.62500	1.25000	0.50000	1.00000
↓	↓	↓	↓	↓
1	**0**	**1**	**0**	**1**

अतः $(0.65625)_{10} = (0.10101)_2$

Binary To Decimal:- यदि किसी Binary Number को Decimal में परिवर्तित करना हो, तो निम्न प्रकार किया जाता हैंः–

(i) $(11111)_2 = (?)_{10}$

$1*2^4+1*2^3+1*2^2+1*2^1+1*2^0$

$= 16+8+4+2+1 = (31)_{10}$

(ii) $(110101)_2 = (?)_{10}$

$1*2^5+1*2^4+0*2^3+1*2^2+0*2^1+1*2^0$

$= 32+16+0+4+01 = (53)_{10}$

(iii) $(1100.1011)_2 = (?)_{10}$

$1*2^3+1*2^2+0*2^1+0*2^0+0*2^{-1}+0*2^{-2}+1*2^{-3}+1*2^{-4}$

$= 8+4+0+0+1/2+1/8+1/16$

$= 8+4+.5+.125+.0625$

$= (12.6875)_{10}$

Decimal to Octal:- Decimal Number की Octal में बदलने के लिये 8 से divide किया जाता हैं व शेषफल (0-7) के मध्य आता हैं।

जैसेः– यदि $(247)_{10}$ को Octal में बदलना सेः–

8	247	7
8	30	6
	3	

$(247)_{10} = (367)_8$

यदि $(0.6875)_{10}$ को Octal में परिवर्तित करना होः–

0.6875 *8 = 5.5000 → **5**

0.5000 *8 = 4.0000 → **4**

$(0.6875)_{10} = (0.54)_8$

Binary to Octal:- Binary Number को Octal को बदलने के लिये LSB (Least Significant Bit) (Right Side) से MSB (Most Significant Bit) Left Side की तरफ तीन Bits का Group बनाया जाता हैं व उसे Binary form में परिवर्तित कर दिया जाता हैं।

(i) $(1001110)_2 = (001\ 001\ 110)_2$

$= (1\ 1\ 6)$

$= (116)_8$

(ii) $(0.10100110)_2 = (0.\ 101\ 001\ 110)_2$

$= (5\ 1\ 6)$

$= (0.516)_2$

Octal to Binary:- Octal को Binary में परिवर्तित करने के लिये हर Octal digit को उसके 3 Bit Binary form में परिवर्तित कर दिया जाता हैं।

$(736)_8 = (111\ 011\ 110)_2$

Hexadecimal to Decimal Coversion:- Hexadecimal Number System में Base 16 को उपयोग में लिया जाता हैं। hexadecimal No. को प्रदर्शित करने के लिये Numeric (0-9) व Alphabet (AF) दोनो उपयोग में लिये जाते हैं।

जैसे यदि $(3A.2F)_{16}$ को decimal में परिवर्तित करना हैः–

$(3A.2F)_{16} = 3*16^1+10*16^0+2*16^{-1}+15*16^{-2}$

$= 48+10+2/16+15/16^{-2}$

$= (58.1836)_{10}$

Decimal to Hexadecimal:-

(1) $(95.5)_{10} = (?)_{16}$

16	95	15
	5	

$(5F)_{16}$

.5 *16 = 8.0 → **8**

$(95.5)_{10} = (5F.8)_{16}$

Binary to Hexadecimal:- किसी Binary No. को Hexadecimal में परिवर्तित करने के लिये राईट से 4-Bit के Pair बनाए जाते हैं व उनकी Binary Equivalent Value लिख दी जाती हैं।

जैसेः– $(10100110101111)_2$ = 0010 1001 1010 1111

2 9 A F

$(29AF)_{16}$

Hexadecimal to Octal:- यदि Hexadecimal Number को Octal में बदलना हो तोः–

(i) Hexadecimal Number को Binary में बदलना

(ii) 3 Bit के Group बनाना

(iii) Binary Equivalent Value लिखना

जैसेः– (i) $(A73E)_{16} = (1010\ 0111\ 0010\ 1110)_2$

001 010 011 100 101 110

1 2 3 4 5 6

$= (123456)_8$

Octal to Hexadecimal:-

(i) Binary में परिवर्तित करना

(ii) 4 Bit का Group (Right Left)

(iii) Binary Equivalent में लिखना

जैसेः– $(24736)_8$

$= (010\ 100\ 111\ 011\ 110)_2$

1010 0111 0111 1000

A 7 7 8

$= (A7.78)_{16}$

दशमलव के बाद की संख्या में Group Left से Right (बांयी से दायी) तरफ बनते हैं।

One's Compliment (1's Complement):- किसी भी Binary Number को उसके is Comp. में बदलने के लिए 1 के स्थान पर 0 व 0 के स्थान पर 1 कर दिया जाता हैं। परन्तु यदि 0 Number $(+5) – (1010)_2$ हैं तो I's Comp. निकालने पर वह $(-5) – (1010)_2$ हो जाएगा। इस प्रकार के नम्बर इनके लिये MSB (Most Significant Bit) Left से 1'st Bit जब 0 हो तो वह No. Positive व जब वहीं Bit 1 हो तो No. Negative होता हैं जैसेः–

+7 – 0111 / +8 = $(01000)_2$

-7 = 1000 / -8 = $(10111)_2$

2's Complement:- जब किसी Binary Number के 1's Complement में 1 जोड़ा है तो 2's Compliment प्राप्त होता हैं। जैसेः–

+5 – 0101

-5 – 1010 1's Comp.

+1

-5 – 1011 – 2's Comp.

9's Compliment:- किसी भी Value का 9's Compliment निकालने के लिये कोई नम्बर N का Base r व उसमें N Digits हो तो जैसे decimal Numbers के लियेः–

r=10 तो r=10-1 = 9 अतः किसी नम्बर N का 9's Comp. (10^n-1)-N होगा। जैसे यदि n=4 हो तो

$(10^4-1) = 10000-1 = 9999$

अतः यदि किसी दिये गये नम्बर के हर digit से 9 को घटा दिया जाए तो 9's Comp. प्राप्त होता हैं जैसेः–

546700 –

999999

546700

453299 – 9's Compliment

10's Compliment:- 9's Comp. में 1 जोड़ने से 10's Comp. प्राप्त होता हैं। जैसे 2389 को 10's Comp.

9999

2389

7610 – 9's Comp.
+1 – 10's Comp.
7611

Arithmetic Operation On Branch Numbers:-

Addition:- + 6 00000110
+13 +00001101
19 – 00010011

यदि 1+0 = 1
1+1 = 10 या –> 0 लिखा जाएगा व 1 Carry के रूप में रहेगा।
जब दोनो Value Positive हो तो उन्हें आसानी से Add कर दिया जाता हैं। पर यदि एक Value Positive व एक Value Negative हो तो इसके लिये Negative (-) Value का 2's Compliment निकाल कर उसे Add किया जाता हैं। जैसेः–

(+6)+(-13) = -7

इसके लिये पहले 13 का 2's Compliment निकाला जाएगाः–

13 – 00001101
1's Comp. – 11110010
+1
11110011

अब 13^{th} के 2's Comp. को 6 में Add कर दिया जाएगा।

+ 6 00000110
-13 11110011
- 7 11111001

Subtraction:- जब दो decimal या Binary Numbers की Subtraction करनी हैं तो दो मुख्य बातों का ध्यान रखना आवश्यक हैंः– (i) जब M>N ही
(ii) जब M<N हो।

यदि दो नम्बर M व N हैं व यदि N, M से छाटा हैं अर्थात् M>N हैं तोः–

(i) N का 10's Comp. निकालना
(ii) M में N के 10's Comp. को Add करना।
(iii) End Carry को Discard करने के लिये आने वाले Sun की उतनी ही 10 की घात से लेस करना।

जैसेः–दो Number –

72532-13250 = 59282
M N

यहाँ M>N हैं।

(i) अतः M का 10's Compliment - 99999
13250
9's Comp. 86749
+1
86750
10's Comp.

(ii) M में N को 10's Comp. को Add करनाः–

M – 72532
N's 10's Comp. - 86750
159282

(iii) Carry को discard करने के लियेः–

159282
10^5 - 100000
Answer 59282

जैसे 13250-72532 = -59282
M N

(i) M<N हैंः–
(i) N का 10's Compliment
99999

- 72532
9's Comp. 27467
+1
10's Comp. 27468

(ii) N के 10's Comp. को M में Add करना

13250
+ 27468
Sum. 40718

(iii) Sum का 10's Comp. निकालनाः–

99999
- 40718
59281 Answer

इसी प्रकार यदि Binary Numbers का Subtraction करना हो जब (i) M>N हो (ii) M<N हो

(i) जब M>N होः–

(i) N का 2's Comp.

(ii) M में N के 2's Comp. को Add करना।

(iii) Carry discard करने के लिये 2 की Power को Less करना।

M = 1010100
N = 011110
M>N

(i) N का 2's Comp.

0111101
1's Comp. – 1000010
+1
2's Comp. 1000011

(ii) M में N के 2's Comp. को Add करनाः–

1010100
+1000011
1001011

(iii) Carry discard करने के लिये

10010111
2^7 10000000
10000111 Answer

(iv) जब M<N होः– (i) N का 2's Comp.
(ii) M+N's Comp.
(iii) Result का 2's Comp.

N = 1010100 M = 1000011

Overflow:- जब दो n digits को Add करने पर Result n+1 आता हैं तो इस प्रकार की Condition Overflow Condition कहलाती हैं। जब यही कार्य Paper, Pencil पर किया जाता हैं तो Overflow Condition Occur नहीं होती परन्तु यदि digital Computers में यह कार्य किया जाता हैं तो Problem Occur होती हैं क्योंकि रजीस्टर की Fix Size होती हैं। Overflow तभी Occur होता हैं जब दो Numbers को Add किया जाए। यदि एक Number Positive व अन्य Negative हो तो Overflow Occur नहीं होता।

CODES

Computer तथा अन्य digital Circuit Data को Binary Format में Process करते हैं Numeric Alphabets या अन्य प्रकार के Data को प्रदर्शित करने के लिये विभिन्न Binary Code को उपयोग में लिया जाता हैं।

BCD Code:- इस प्रकार के Code में 0-9 decimal Number को उसकी Binary form में प्रदर्शित किया जाता हैं। यह Binary form 4 Bits में प्रदर्शित की जाती हैं तथा हर decimal Number के decimal digit को उसकी binary form में प्रदर्शित किया जाता हैं। जैसे यदि Binary Number – 1000001 को decimal में = 65 व BCD में = 01000001 = 41 लिखा जायेगा।

(2 3) को BCD में 00100011 लिखा जाएगा। ना कि $(10111)_2$ उसमें Bit by Bit Values को Store किया जाता हैं जिसे बड़ी Values के लिये Registers भी अधिक चाहिये होते हैं इस disadvantage के होते हुए भी digital Systems के लिये Input/Output Oper Valions को इसी प्रकार Represent किया जाता हैं। इस Code को 3-4-2 Code भी कहा जाता हैं।

Decimal	Binary	BCD	Excess3	GRAY
Number	B_3 B_2 B_1 B_0	D C B A	E_3 E_2 E_1 E_0	G_3 G_2 G_1 G_0
0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0
1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1
2	0 0 1 0	0 0 1 0	0 1 0 1	0 0 1 1
3	0 0 1 1	0 0 1 1	0 1 1 0	0 0 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 1 0
5	0 1 0 1	0 1 0 1	1 0 0 0	0 1 1 1
6	0 1 1 0	0 1 1 0	1 0 0 1	0 1 0 1
7	0 1 1 1	0 1 1 1	1 0 1 0	0 1 0 0
8	1 1 1 0	1 1 1 0	1 0 1 1	1 1 0 0
9	1 0 0 1	1 0 0 1	1 1 0 0	1 1 0 1
10	1 0 1 0			1 1 1 1
11	1 0 1 1			1 1 1 0
12	1 1 0 0			1 0 1 0
13	1 1 0 1			1 0 1 1
14	1 1 1 0			1 0 0 1
15	1 1 1 1			1 0 0 0

Excess 3 Code:- यह BCD Code की ही अन्य Form हैं जिसमें यह हर Decimal Digit करे उसके 4 bit Binary Code में प्रदर्शित किया जाता हैं। यह Code वास्तविक BCD Code में 3(0011) को Add कने से बनता हैं जैसे 1 के लिये – 0001+001

0001
+ 0011
0100 – Excess 3 Code

इस Code को Weighted Code कहा जाता हैं यह Code Sly Complimenting Code कहलाता हैं। अर्थात् 2 का Excess 3 Code 0101 हैं तथा इसका 1's Compliment 1010 – का Excess 3 Code हैं। जो कि 2 का 9's Compliment हैं।

Gray Code:- 1 bit के लिये निम्न Gray Code हैं:–

(i)

Decimal Number	Gray Code
0	0
1	1

(ii) 2 bit Gray Code

Decimal Number	Gray Code
0	00
1	01
2	11
3	10

(iii) 3 bit के लिये Gray Code:-

Decimal Number	Gray Code
0	000
1	001
2	011
3	010
4	110
5	111
6	101

7 100

EBCDIC – Extended BCD Interchange Code.

ASCII – American Standard Code for Information Interchange.

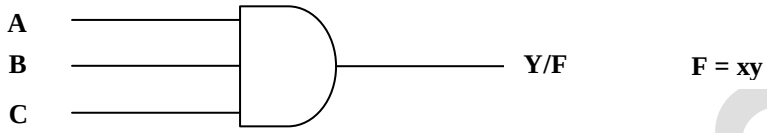
UNIT – II

Boolean Function को AND, Or तथा Not Operation के रूप में परिभाषित किया जाता हैं। इसके अलावा अन्य Gates बनाते समय निम्न चार बातों को ध्यान रखना आवश्यक हैः–

(1) Data को बनाने के Physical Components को ध्यान में रखना
(2) Gate में दो से अधिक Inputs की संभावना को बढ़ाना।
(3) Binary Operators की मुख्य Properties को ध्यान में रखना।
(4) Gate का अन्य Gates के साथ भी Boolean Operation को Perform करना।

AND Gate:- वह सर्किट जिसके द्वारा AND Operation Perform किया जाता हैं। इसे AND Gate कहा जाता हैं। इसे (N>2) को Input के रूप में लिया जाता हैं व एक Output प्राप्त होता हैं।

And Operation को निम्न प्रकार define किया जाता हैः– Output तभी 1 होगा जब डाले गए सभी Inputs 1 हो।



AND Gate

डाला गया सभी Input Binary Form में होता हैं अतः Input या तो 0 या तो 1 ही डाला जाएगा। इन Binary Variable को Logical Variables भी कहा जाता हैं। दो Variables के मध्य जब AND Operation को Perform किया जाता हैं तो इसे (.) से प्रदर्शित करते हैं।

Truth Table:- Logical Variables को दो Values ही होती हैं (0 या 1) किसी भी Logical Operation को Table की Form में परिभाषित किया जाता हैं। जिसमें सभी Possible Input को व उनके Outputs को प्रदर्शित किया जाता हैं। इस प्रकार की Table को Truth Table कहा जाता हैं।

Truth Table को निम्न प्रकार से प्रदर्शित किया जा सकता हैः–

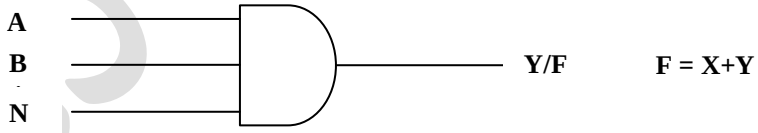
Input		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

A.B = Y

OR-Operation:- Or gate में भी N Inputs (N>2) दिये जा सकते हैं परन्तु आउटपुट हमेशा 1 होता हैं OR Gate को निम्न प्रकार define किया जाता हैं।

Or Gate के द्वारा Output 1 होगा जब दो या एक Input 1 हो।

Y = A or B or C …….. OR….N
= A+B+C …… +N



Or Gate के लिये 2 Inputs की Truth Table निम्न हैः–

Truth Table

Input		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

(3) NOT Operation:- Not Gate को Inverter भी कहा जाता हैं। इसमें 1 Input व एक Output होता हैं। इसे निम्न दो प्रकार से प्रदर्शित किया जा सकता हैं:–

(A) $F=x^1$ **(B) $F=x^1$**

NOT Gate

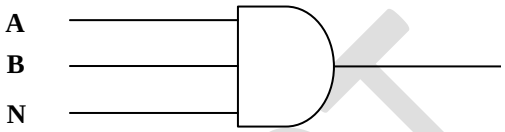
Y = NOT A
= $\overline{A}$ (A Compliment)

TRUTH TABLE

Input	Output
A	B
0	1
1	0

(N>2) gate

NAND Gate:- Mand Function and gate का Compliment होता हैं। इसमें जब दोनो Input 0 व दोनो में से कोई एक Input 1 हो तो Output 1 होता पर यदि दो Input 1 हो तो Output 0 होता हैं।

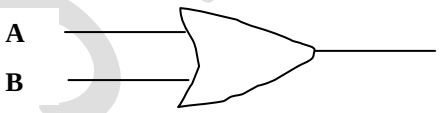


$F = (xy)^1$

TRUTH TABLE

Input		Output
X	Y	F/Y
0	0	1
0	1	1
1	0	1
1	1	0

Nor Gate:- Nor Gate को OR का Compliment कहा जाता हैं। इसमें जब दोनो Input 0 हो तो Result 1 होगा पर यदि दोनो Input 1 या एक भी Input 1 हो तो Result 0 होगा।

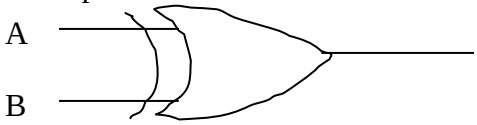


$F = (A+B)^1$

TRUTH TABLE

Input		Output
A	B	Y/F
0	0	1
0	1	0
1	0	0
1	1	0

Exclusive-OR (X-OR):- इसमें यदि दोनो Input 0 या दोनो Input 1 हो तो Output 0 व यदि दोनो Input में से कोई एक Input 1 हो तो Output 1 होगा।



$F = xy^1+x^1y$
$= x+y$

TRUTH TABLE

Input		Output
A	B	Y/F
0	0	0
0	1	1
1	0	1
1	1	0

$F = xy^1+x^1y$
$= x+y$

BOOLEAN ALGEBRA

Digital Signals सीर्फ 0 या। Values को ही समझते हैं। अतः इन पर एक Number System बना हैं जिसे Binary Number System कहा जाता हैं। 19 वीं शताब्दी के मध्य में एक अंग्रेज गणितज्ञ George Boolean द्वारा कुछ Rules बनाए गए (Binary Variables पर) जिन्हें Boolean Algebra कहा जाता हैं।

Boolean Algebra Theorem:-

A+0 = A	$A(\overline{A}+B) = (AB)$
A+1 = A	$AB+A\overline{B} = A$
A+1 = 1	$(A+B).(A+\overline{B}) = A$
A+0 = 0	$AB+\overline{A}C = (A+C)(\overline{A}+B)$
A+A = A	A.B.C = A . B. C
$A+A = A$	(Break the line & Change the Sign.)
$A+\overline{A} = 1$	
$A+\overline{A} = 0$	
A.(B+C) = AB+AC	
A+BC = (A+B) (A+C)	
A+AB = A	
A (A+B) = A	
$A+\overline{A}B = (A+B)$	

इन Theorems को Demorgans Theorem कहा जाता हैं।

MAP Simplification:- Logical Functions को Logical Variables के रूप में प्रदर्शित किया जाता हैं। Logical Function या Logical Variables की Value को Binary form में प्रदर्शित किया जाता हैं। किसी भी Logical Function को निम्न प्रकार से परिभाषित किया जा सकता हैः–

(1) Sum of Products (SOP). and
(2) Product of Sum (POS).

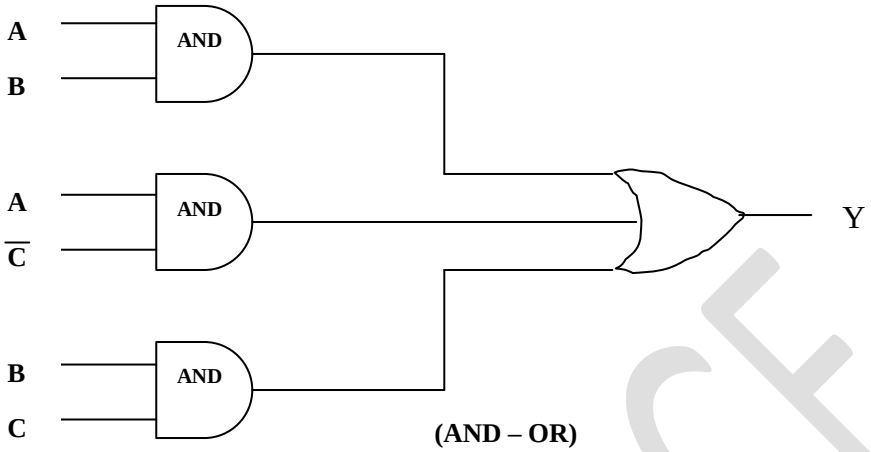
जैसे एक Logical Equation दी गई हैः–
$Y = (A+BC)(B+\overline{C}A)$ को SOP form में परिवर्तित करनाः–
$Y = (A+BC)(B+\overline{C}A)$
पहले Bracket की Values को दूसरे से And Operation Perform करवानाः–

$A(B+\overline{C}A) + BC(B+\overline{C}A)$
$= AB+A\overline{C}A+BCB+BC\overline{C}A$
$= A.B+A.A\overline{C}+B.BC+BA\,C\overline{C}$
$= AB+A\overline{C}+BC+0$

SOP $= AB+A\overline{C}+BC$

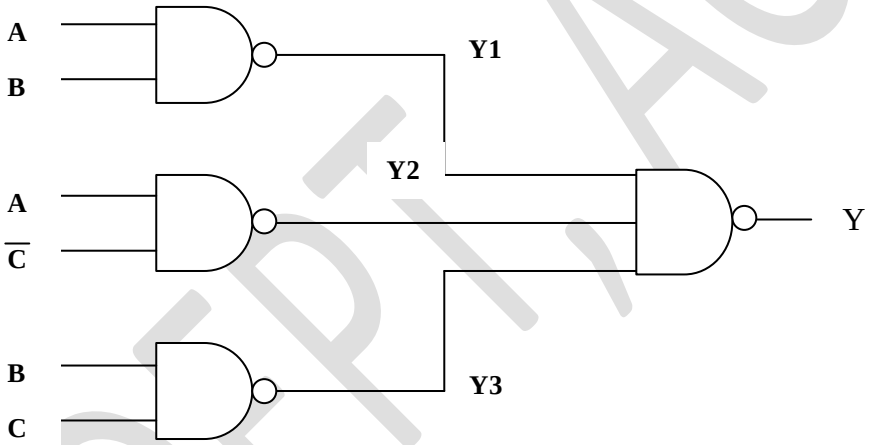
Because:- A.A=A, B.B=A But $C.\overline{C}=0$

Diagram:-



(AND – OR)

इसी Diagram को निम्न प्रकार प्रदर्शित किया जा सकता हैः–



यदि इसे ही POS में Convert करना हो तोः–

y = (A+BC) (B+CA)

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

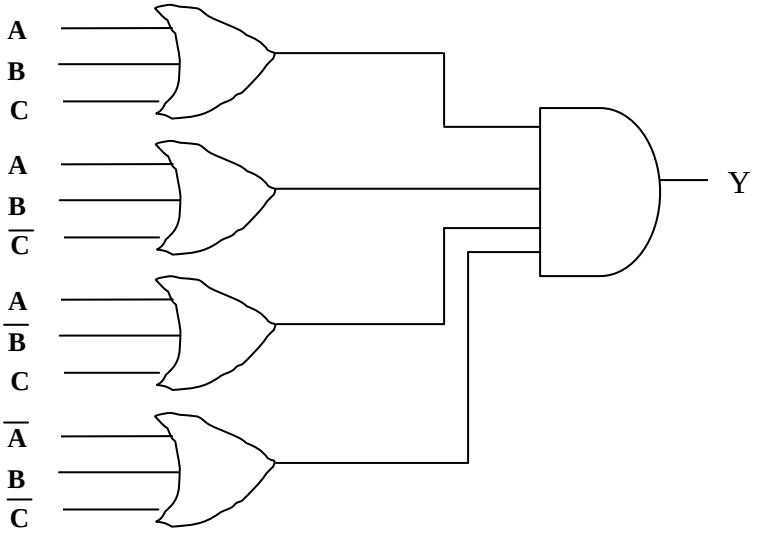
= (A+B) (A+C) (B+$\overline{C}$)

यदि जो Value उपस्थित नहीं उसे एक बार 1 व एक e बार C बना कर लिखा जाए जैसे (A+B) में C उपस्थित नहीं हैं तो उसे CC के रूप में लिखा जा सकता हैः–

(A+B+C$\overline{C}$) (A+C+B$\overline{B}$) (A$\overline{A}$+B+$\overline{C}$)

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

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



K-Maps:- Logical Function को Short hand में लिखने या उनके Minimization के लिये Minterm या Maxtem को उपयोग में लिया जाता हैं। जब Logical Variables के मध्य AND Operation Perform किया जाता हैं तो वे Minters वे जब उन्हीं Logical Variables के मध्य Operation Perform किया जाता हैं तो वे Maxterm कहलाती हैं जब X=0 हो तो वह X^1/X व यदि X=1 हो तो वह X& लिखा जाता हैं। इस प्रकार 2^4=16 के लिये Minterms व Maxterms निम्न हैं:–

No.	Variable	Minterm	Maxterms
	A B C D	Mi	Mi
0	0 0 0 0	$\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} = M_0$	$A+B+C+D = M_0$
1	0 0 0 1	$\overline{A}\,\overline{B}\,\overline{C}\,D = M_1$	$A+B+C+\overline{D} = M_1$
2	0 0 1 0	$\overline{A}\,\overline{B}\,C\,\overline{D} = M_2$	$A+B+\overline{C}+D = M_2$
3	0 0 1 1	$\overline{A}\,\overline{B}\,C\,D = M_3$	$A+\overline{B}+\overline{C}+D = M_3$
4	0 1 0 0	$\overline{A}\,B\,\overline{C}\,\overline{D} = M_4$	$A+\overline{B}+C+D = M_4$
5	0 1 0 1	$\overline{A}\,B\,\overline{C}\,D = M_5$	$A+\overline{B}+\overline{C}+\overline{D} = M_5$
6	0 1 1 0	$\overline{A}\,B\,C\,\overline{D} = M_6$	$A+\overline{B}+\overline{C}+D = M_6$
7	0 1 1 1	$\overline{A}\,B\,C\,D = M_7$	$A+\overline{B}+\overline{C}+\overline{D} = M_7$
8	1 0 0 0	$A\,\overline{B}\,\overline{C}\,\overline{D} = M_8$	$\overline{A}+B+C+\overline{D} = M_8$
9	1 0 0 1	$A\,\overline{B}\,\overline{C}\,D = M_9$	$\overline{A}+B+\overline{C}+\overline{D} = M_9$
10	1 0 1 0	$A\,\overline{B}\,C\,\overline{D} = M_{10}$	$\overline{A}+B+\overline{C}+D = M_{10}$
11	1 0 1 1	$A\,\overline{B}\,C\,D = M_{11}$	$\overline{A}+B+\overline{C}+\overline{D} = M_{11}$
12	1 1 0 0	$A\,B\,\overline{C}\,\overline{D} = M_{12}$	$\overline{A}+\overline{B}+C+\overline{D} = M_{12}$
13	1 1 0 1	$A\,B\,\overline{C}\,D = M_{13}$	$\overline{A}+\overline{B}+C+\overline{D} = M_{13}$
14	1 1 1 0	$A\,B\,C\,\overline{D} = M_{14}$	$\overline{A}+\overline{B}+\overline{C}+D = M_{14}$
15	1 1 1 1	$A\,B\,C\,D = M_{15}$	$\overline{A}+\overline{B}+\overline{C}+D = M_{15}$

K Map को Squares से मिलाकर बनाया जाता हैं हर Square एक Minterm को प्रदर्शित करता हैं। अतः किसी भी Boolean Function को Minterms के Sum में प्रदर्शित किया जाता हैं।

2 or 3 Variable के Map:-2 Variable के Map में 4 Minterm को प्रदर्शित किया जाता हैं। अतः K-Map में चार Squares बनेगें जिसमें हर एक Minterm को प्रदर्शित किया जाएगा।

(1) यदि x=0 = x^1 द्वारा प्रदर्शित करेगें।

यदि x=1 हैं तो उसे x द्वारा प्रदर्शित किया जाएगा।

इसी प्रकार y के लिये भी यही Terms Use में ली जायेगी।

		x/y	**y^1{0**	**y{1**
M_0	M_1	**x^1{0**	**x^1y^1**	**x^1y**
M_2	M_3	**x{1**	**xy^1**	**xy**

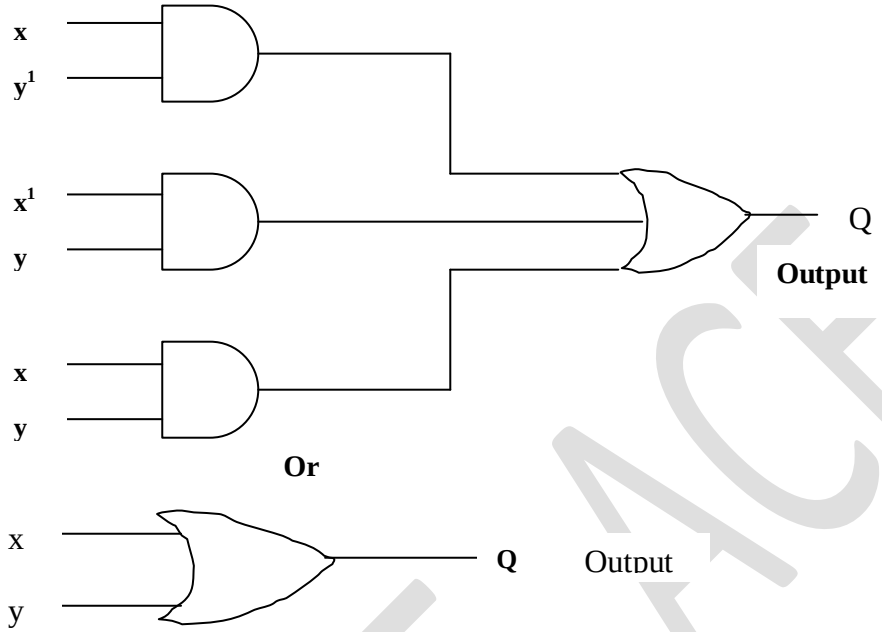
यदि $M_1+m_2+m_3$ के लिये K-map बनाना हैं, तो M_1, M_2 व M_3 के ब्लॉक में 1 Values को रख दिया जाएगा।

x/y	**0**	**1**
0	**0**	**1** 1
1	**1** 2	**1** 3

इसके लिये Minimization Function निम्न हैं:–

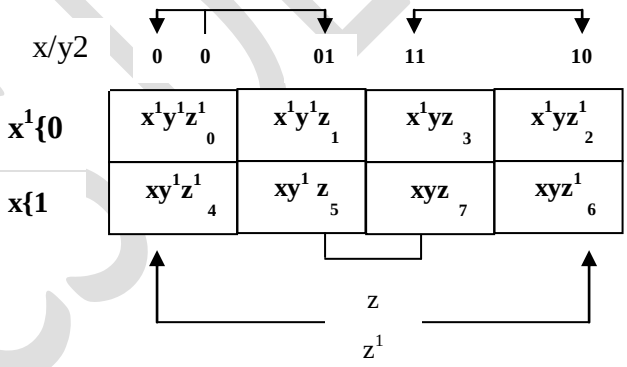
$xy^1+x^1y+xy = x+y$

Diagram:-



K-Map Diagram for 2 Variables

K-Map for Three Variables:- तीन Variable के K-Map के लिये 8 Minterms को उपयोग में लिया जाता हैं। अतः Map में भी 8 Squares को बनाया जाता हैं। परन्तु ये Minterms Binary Table की Form में नहीं बनायी जाती। क्योंकि जब मेप को फोल्ड किया जाए तो एक दूसरे पर आने वाली Bits की Mirror Images Same नहीं होनी चाहिए। अतः K-map Square को निम्न प्रकार से बनाया जा सकता हैं।

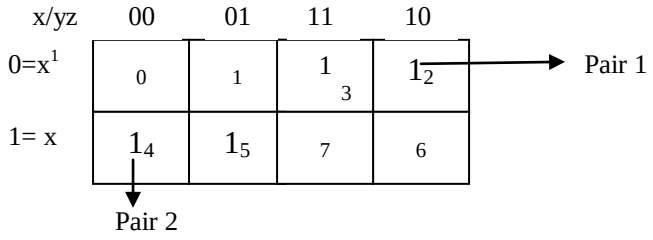


	00	01	11	10
0	M_0	M_1	M_3	M_2
1	M_4	M_5	M_6	M_7

K-map को Solve करने के लिये 2 के गुणांको में Pairs बनाये जाते हैं। व इनमें यह चैक किया जाता हैं तीनो Variable में से कौनसी दो Values Same हैं उन्हें ही Function बनाने के लिये उपयोग में लिया जाता हैं। जैसे एक Function दिया गया हैं।

$F = x^1yz+x^1yz^1+x^1yz^1+xy^1z^1+xy^1z$

सभी Variables की Value रख दी जाएगी।



Part 1 के लिये समीकरणः–

(i) दोनों में x^1 की Value Same हैं।

(ii) दोनों में y की Value Same हैं।

(iii) परन्तु z की Value को एक बार। व दूसरी बाद 0 है अतः इस Value को Function बनाने के लिये उपयोग में नहीं लिया जाएगा।

$Pair1 = x^1y$

Part 2 के लिये समीकरणः–

(iv) दोनों में x की Value Same हैं।

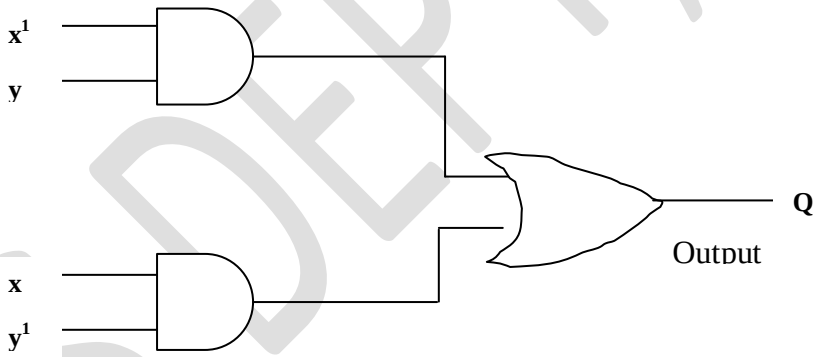
(v) दोनों में y^1 की Value Same हैं।

(vi) परन्तु z की Value बदल रही हैं अतः इसे Function के दौरान उपयोग में नहीं लिया जाएगा।

$Pair2 = xy^1$

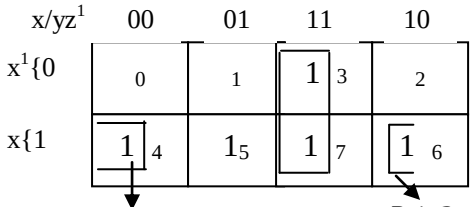
अतः पूरी समीकरण $Pair1 + Pari2$

$= x^1y+xy^1$



3-Variable K-map diagram

$F= x^1yz+xy^1z^1+xyz+xyz^1$



इस समय K-Map को यदि फोल्ड किय Pair 2 ´4 Pair 1 Pair 2 ास आ जाएगें अतः इसे दूसरी Pair बनाया जा सकता हैं।

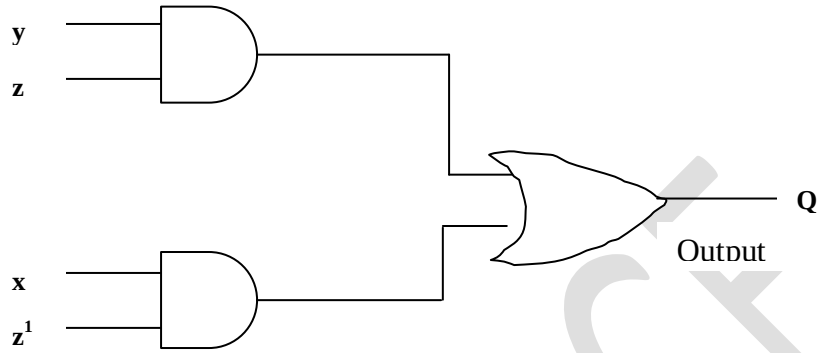
Equation for Pair:- (i) x की Value बदल रही हैं।

(ii) y की Value Same हैं।

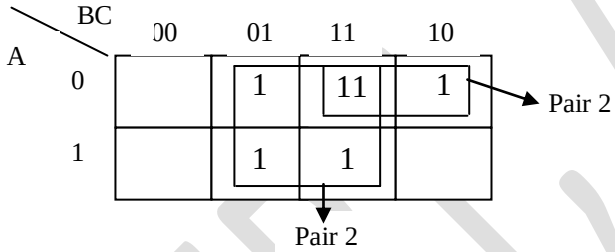
(iii) Z की Value Same हैं।

अतः Equation = yz

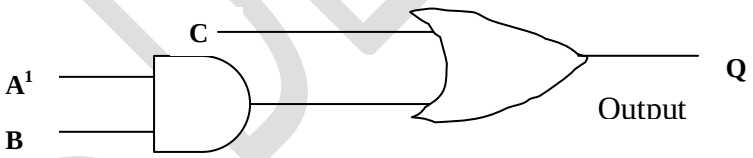
Pair2:- (i) x की Value Same हैं।
(ii) y की Value बदल रही हैं।
(iii) z^1 की Value Same हैं।
अतः समीकरण $F = xz^1$
Full Equation:- Pair1+Pair2
$F = yz+xz^1$



$F = A^1C+A^1B+AB^1C+BC$ के लिये Kmap



A^1C = दो जगह हैं :- (1) 001 (2) 011 अतः दोनों में Value रख दी जाएगी।
A^1B = दो जगह हैं :- (1) 011 (2) 010 अतः इसमें भी दोनो स्थानो पर 1 रखा जाएगा।
BC = BC भी दो स्थानो पर हैं:– (1) 011 व (2) 110 अत+ दोनो स्थानो पर 1 रखा जाएगा।
$F = C+A^1B$

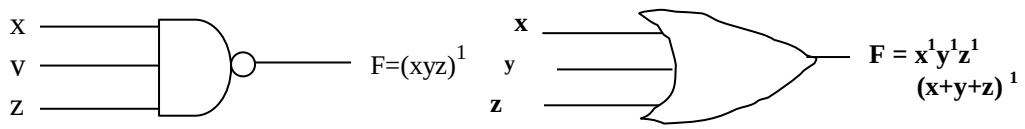


4 Variable K-Map

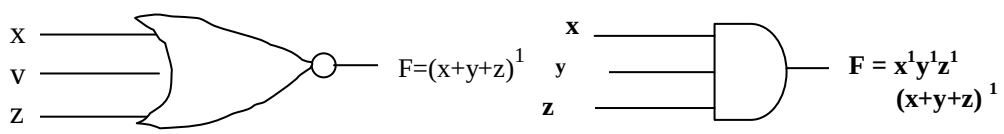
00	$w^1x^1y^1z^1$	$w^1x^1y^1z$	w^1x^1yz	$w^1x^1yz^1$
01	$w^1xy^1z^1$	w^1xy^1z	w^1xyz	w^1xyz^1
11	wxy^1z^1	wxy^1z	$wxyz$	$wxyz^1$
10	$wx^1y^1z^1$	wx^1y^1z	wx^1yz	w^1xyz^1

00	**0**	**1**	**3**	**2**
01	**4**	**5**	**7**	**6**
11	**12**	**13**	**15**	**14**
10	**8**	**9**	**11**	**10**

Diagram Simplification By NAND Or NOR Gates



NAND GATE



Digital Systems के लिये Logical Circuits दो प्रकार के हो सकते हैः–

(1) Combinational (कॉम्बीनेश्नल)

(2) Sequential (सीक्वेंशियल)

Combinational Circuits वे Circuits हैं जो Logical Gates के बने होते हैं, व जिनका Output उनके Input के Combination पर निर्भर करता हैं जो उसे Currently मिला हैं। वह किसी भी Previous (पिछले) Input पर निर्भर नहीं करता। अतः Combination Circuit सिर्फ Boolean Function पर ही Information Processor Perform करता हैं। अतः Combinational Circuits सिफ। Logic Gates के बने होते हैं व अपने Current Input के Combination पर depend करते हैं।

परन्तु Flip Flop एक ऐसा System हैं जो Sequential हैं अर्थात् क्रमवार हैं। Flip-flop व Combinational Circuit को उपयोग में लेकर Sequential Logical Circuit को बनाया जाता हैं। सबसे महत्वपूर्ण Sequential Circuit जिन्हें सबसे ज्यादा Digital System में उपयोग लिया जाता हैं वे Registers व Counters Combinational Circuit को बनाने के लिये निम्न Steps को Follow किया जाता हैः–

(1) सबसे पहले समस्या को समझना।

(2) अब Input Variables के आधार पर Output Variables का अनुमा लगाना।

(3) Input व Output Variables को Letters के रूप में प्रदर्शित किया जाता हैं।

(4) Input व Output को परिभाषित करने के लिये उसकी Truth Table बनाना।

(5) ळर Output के लिये Boolean Function को प्रदर्शित करना व उसे K-Map द्वारा सरल करना।

(6) Logical Diagram को प्रदर्शित करना।

Adders:- Digital Computers बहुत सी Information Processing कार्यों को Peform करते हैं। जिनमें मुख्य Arithmeted Operations हैं। सबसे मुख्य Operation दो Binary digits का Function हैं। इसके लिये 4 Possible Operations किये जाते हैंः–

0+0 = 0, 0+1 = 1, 1+0 = 1, 1+1 = 10

जब 9+1 को Add किया जाता हैं तो Result 10 आता हैं इसमें 0 को Sum के रूप में व 1 को Carry के रूप में प्रदर्शित किया जाता हैं। अतः वह Combinational Circuit जिसके द्वारा दो Bits की Addition को Perform किया जाता हैं तो वह Full Adder कहलाता हैं।

Half-Adder:- इस प्रकार के Circuit में 2 Input व 2 Output को Consider किया जाता हैं। Input को x व y के रूप में प्रदर्शित किया जाता हैं। इसके लिये Truth Table निम्न हैः–

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

अब हर Output (Carry व Sum) के लिये K-Map बनाया जाएगाः–

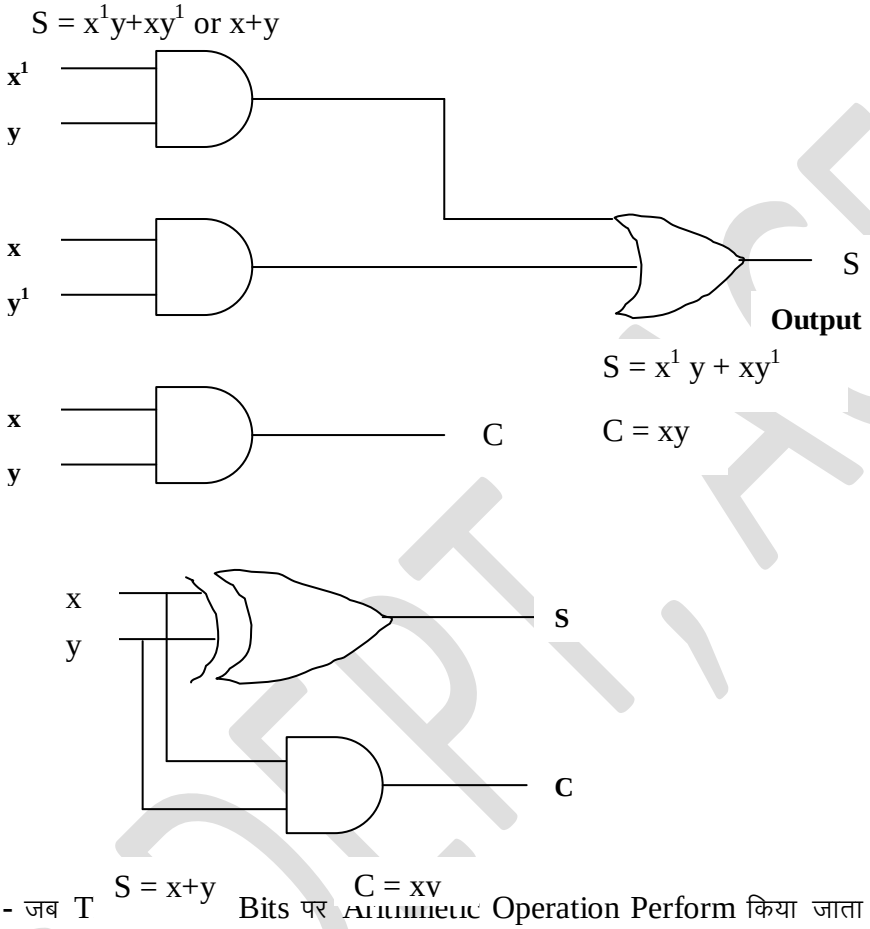
K-Map For Carry:-

x \ y	0	1
0	0	1
1	2	13

C = xy

K-Map For Sum

x \ y	0	1
0	0	1 1
1	1 2	1 3

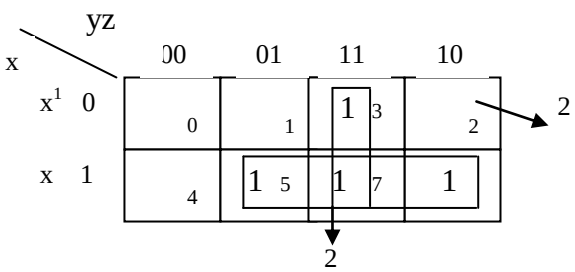


Full Adder:- जब T Bits पर Arithmetic Operation Perform किया जाता हैं तो उसे Full Adder कहते हैं।

Truth Table For Full Adder

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

K-Map For Carry:-



K-Map For Sum:-

x \ yz	00	01	11	10
0	0	1 1	3	1 2
1	1 4	5	1 7	6

$S = x^1y^1z+x^1yz^1+xy^1z^1+xyz$
$C = xy+xz+yz$

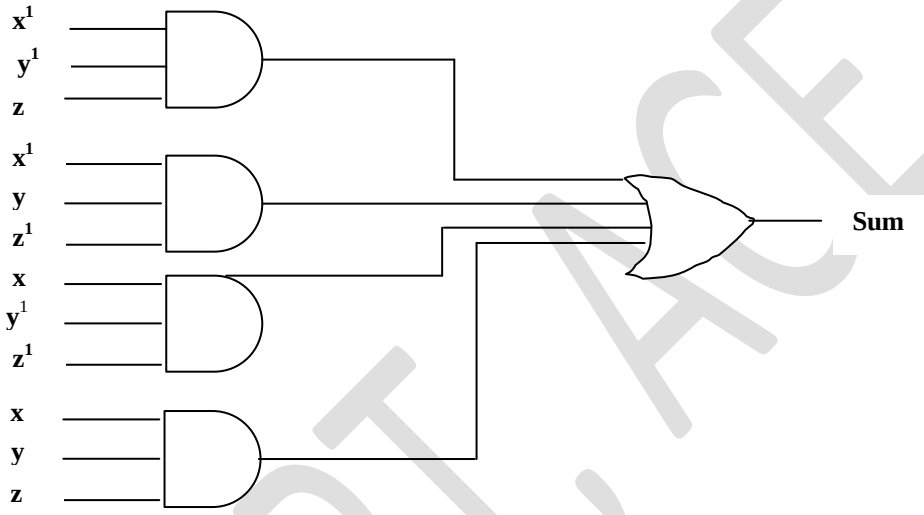
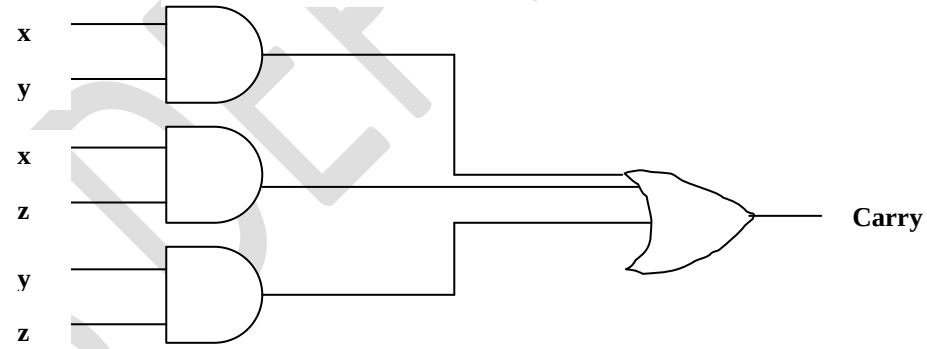


Diagram For Sum



Sub tractor:- दो Bits को Subtraction को Half Adder द्वारा Perform किया जाता हैं। इसमें भी उन्हीं Steps को Follow किया जाता हैं जिनके द्वारा Adder बनते हैं। परन्तु यहाँ आने वाले Output को B (Borrow/उधार) व D (difference से प्रदर्शित किया जाता हैं:–

Truth Table

x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

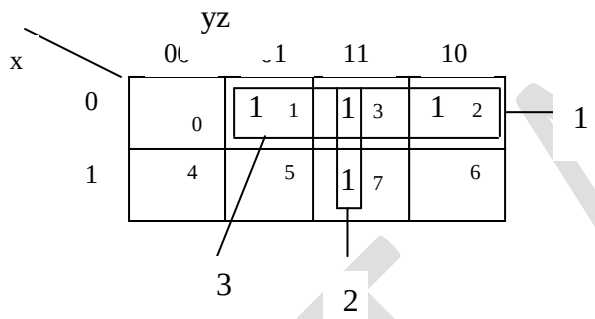
	0	1
0	0	1 1
1	1 2	3

D = x^1y+xy^1

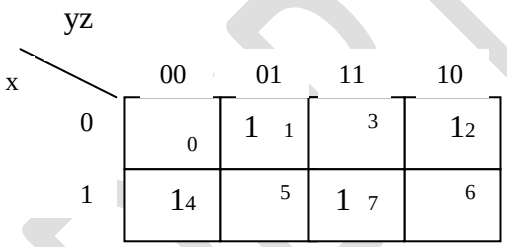
Full Sub tractor:-

x	y	z	C	S
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

K-Map For B



K-Map For D:-



$D = x^1y^1z+x^1yz+xy^1z^1+xyz$

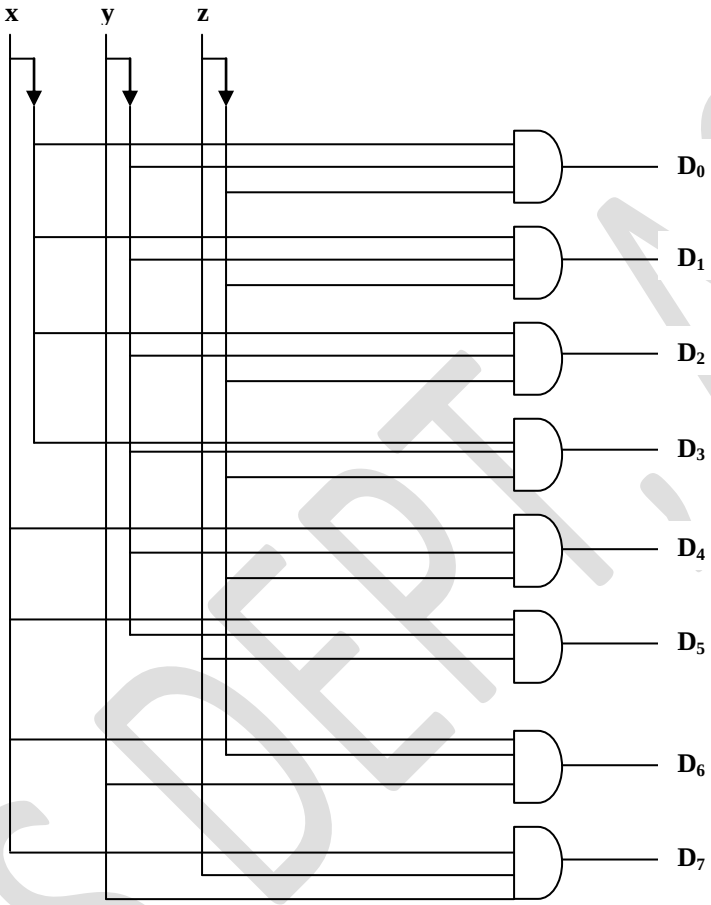
Decoders:- Decode एक ऐसा Combinational Circuit हैं जिसके द्वारा Input से 2^n Outputs प्राप्त किये जा सकते हैं। जैसे 3 Inputs देकर 2^3 Outputs = 8 Outputs प्राप्त किये जा सकते हैं।

Example:- यदि 3x8 Line Decode का Circuit बनाना हैं तो तीन Input को decode करके 8 Outputs decode किये जाएगें। हर Output 3 में से किसी एक Minterm को प्रदर्शित करेगा। इसके लिये Truth Table निम्न प्रकार हैः–

3x8 line Decode

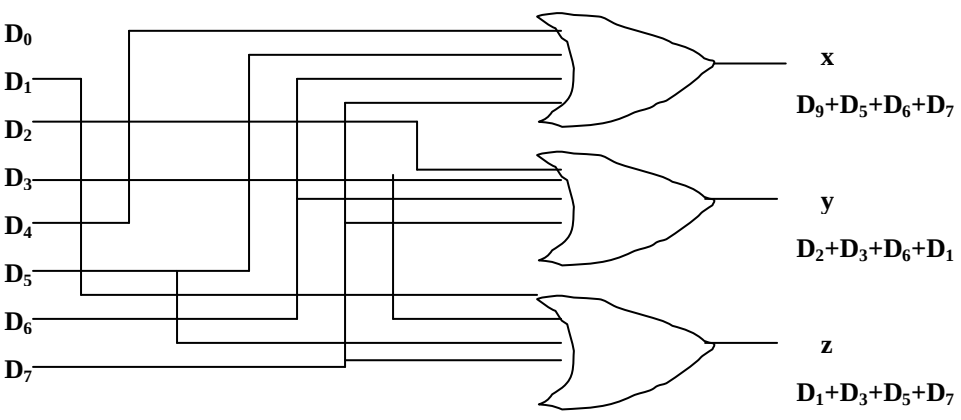
Inputs			Outputs							
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

One Decoder



Encoders:- Encoder ऐसा digital Function हैं जिसके द्वारा decoder का उल्टा Operation Perform किया जाता हैं। अतः Encoder 2^n Inputs लेता हैं व n outputs को प्रदर्शित करता हैं। इन्हें OR Gates द्वारा बनाया जाता हैं।

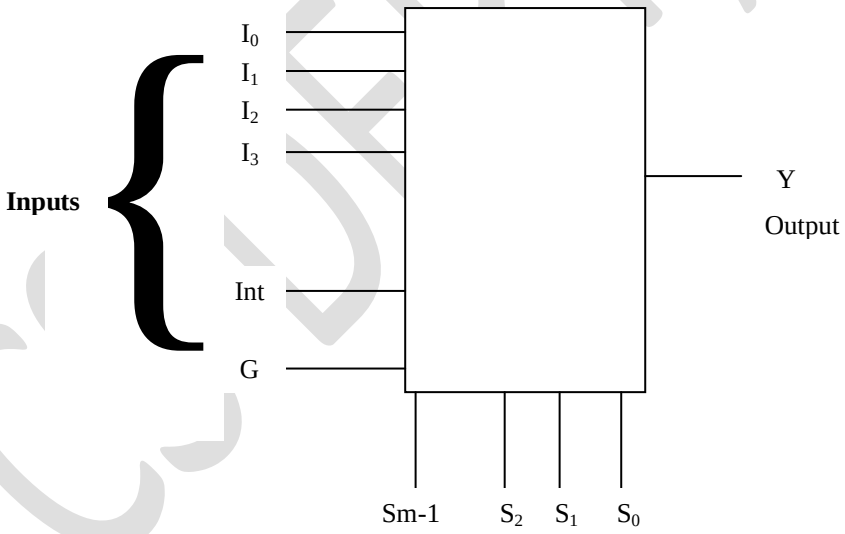
किसी भी समय Encoder केवल एक Input Line को On कर सकता हैं। क्योंकि यदि 8 Inputs डाले जाएं तो Outputs $2^8 = 256$ होते हैं। परन्तु इनमें से सिर्फ 8 Combinations ही Meaning Full होते हैं। व अन्य Inputs को Don't Care Conditions कहा जाता हैं।



TRUTH TABLE

D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Multiplexers:- Multiplexer एक Special Combinational Circuit हैं जिसे Digital Design के लिये उपयोग में लिया जाता हैं। Multiplexer एक ऐसा Logic Circuit हैं जो विभिन्न Input में से किसी एक Output को सलेक्ट करता हैं। Input Selection को Set of Select Inputs के द्वारा Control किया जाता हैं।

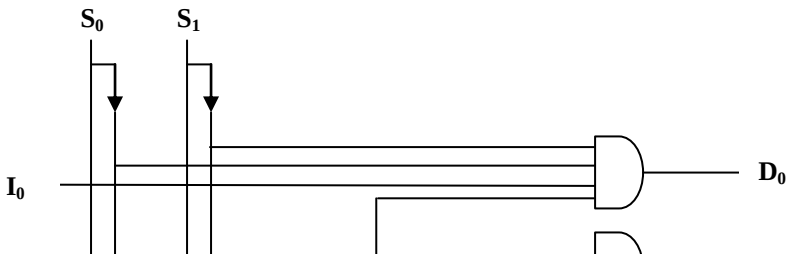


Truthe Table of 4:1 Multiplexer

Select	Inputs	Outputs
S_1	S_0	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$Y = S_1 S_0 I_0 + S_1 S_0 I_1 + S_1 S_0 I_2 + S_1 S_0 I_3$

CIRCUIT DIAGRAM



Multiplexer की विशेषताएँ:-

(1) Logical Expnession को Simplify करने की आवश्यकता नहीं होती।

(2) Design (Gate की) को भी Simple बना दिया जाता हैं।

Flip-Flop

Asked Sequence Circuit में जिन Storage Alement का प्रयोग किया जाता हैं। उन्हें Flip Flop कहते हैं। Flip Flop तक Binary Cell हैं जो एक Bit के Information को Install कर सकते हैं।

Types of Flip-Flop

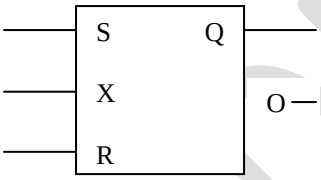
विभिन्न प्रकार Flip Flop इस प्रकार हैं:–

(1) SR Flip-Flop (2) D Flip-Flop

(3) JK Flip-Flop (4) T Flip-Flop

(Set) (Reset)

(1) SR Flip-Flop:- SR F.F Graphics Symbol इस प्रकार का होता हैं:–



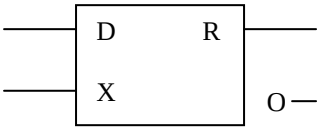
S, R – Output
Q – Output
C – Check Pulse
O – Combliment Output

SR	Q (11)
00	Q (t) No Change
01	O Clear to O
10	1 Sets1
11	? Indentaimiate

Characterisitc Table of SR Data

(2) D Flip-Flop:- SR F.F का Modified रूप हैं। S और R Input के आगे Inverter Gate लगाकर हमने उसे एक Singal Input D Symbol बना दिया।

D-Input, C- Clock Pulse, R- Output

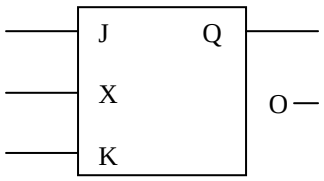


Graphics Symbol

D	R (t+1)
0	0 Clear to O
1	1 Set to 1

Characterisite_table of 1

(3) JK Flip-Flop:- SRF.F. का Refindement रूप हैं। SR ऐसा F.F Indertermative State को इस F.F. के द्वारा difined किया गया हैं।



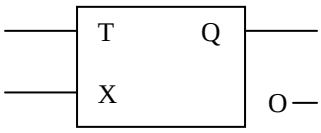
J, K – Input
C – Clock Pulse
Q – Output

Graphics Symbol

J K	Q (t+1)
0 0	Q (t) No Change
0 1	0 ouarts 0
1 0	1 Set to 1
1 1	Q (t) Complement

Characteriste Table of JK

(4) Flip-Flop:- T एक Toggle Flip-Flop हैं। यह F.F J.K. Flip Flop से प्राप्त किया गया हैं। J औस K Input बना दिया हैं।



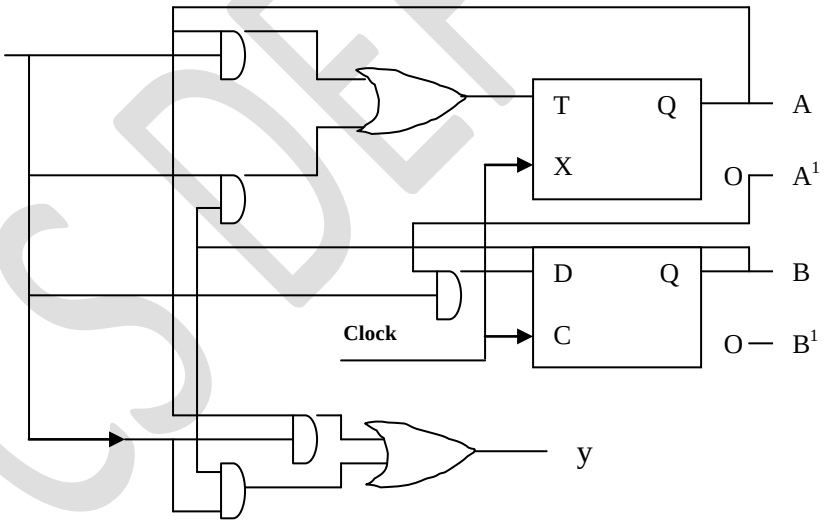
T	Q (t+1)
0	Q (E) N Change
1	Q (t) Compliment

C.T.

G.S

I – Input
C – Clock Pulse
Q - Output

Sequential Circuites:- ऐसे Circuit जो Gates और Flip-Flop से मिलकर बनाते हैं उन्हें Sequential Circuits कहते हैं।

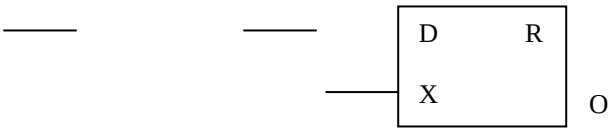


Graphics Symbol

Example of a Sequential Circuits

Decoders:- एक ऐसा Circuit जिसमें N Bits Input के लिए 2^n Output अथवा different Codes होते हैं। उसे Decoders कहते हैं। या

Decoderes वह Combinational Circuit हैं जो N Input की Binary Information को Maximum 2^n Unique में परिवर्तित कर देता हैं।

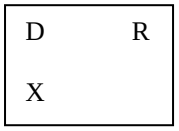


D	R (t+1)
0	0 Clear to O
1	1 Set to 1

Characterisite_table of 1

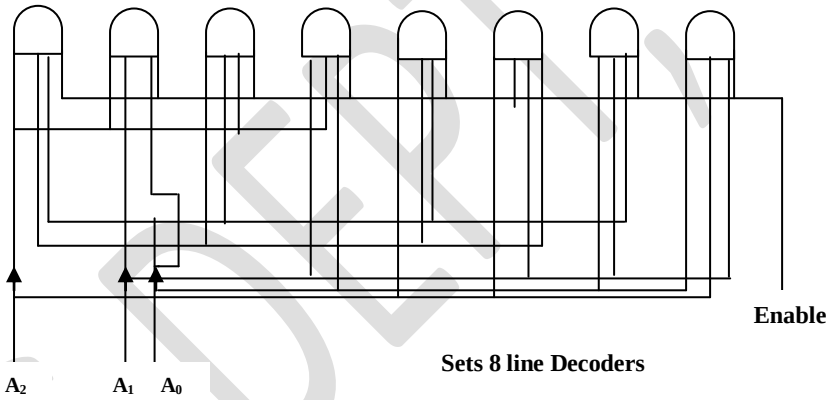
Input				Output							
Enable	A^2	A^1	A^0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	1	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

Graphics Symbol



D	R (t+1)
0	0 Clear to O
1	1 Set to 1

Characterisite_table of 1



Encoders:- Encoders एक Digital Circuit हैं जो Decoders का Inverse Operation Perform करता हैं। एक Encoders में 2^n Input Line और N Output line होती हैं।

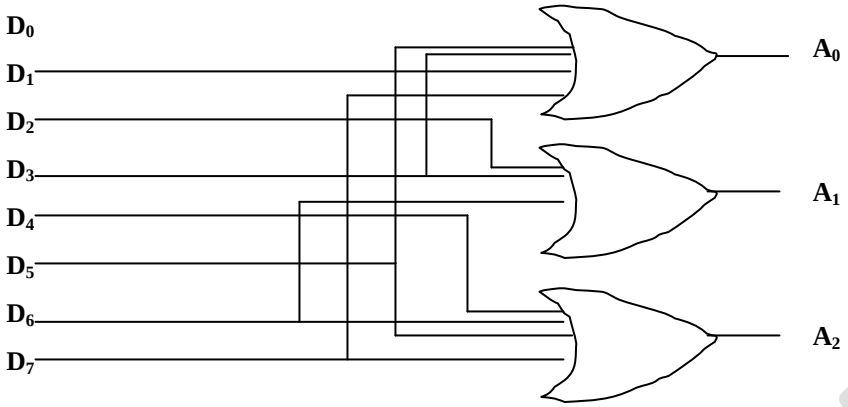
Truth table From Octal to Binary Encoders

Input								Output		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$A_0 = d_1+d_3+d_5+d_7$

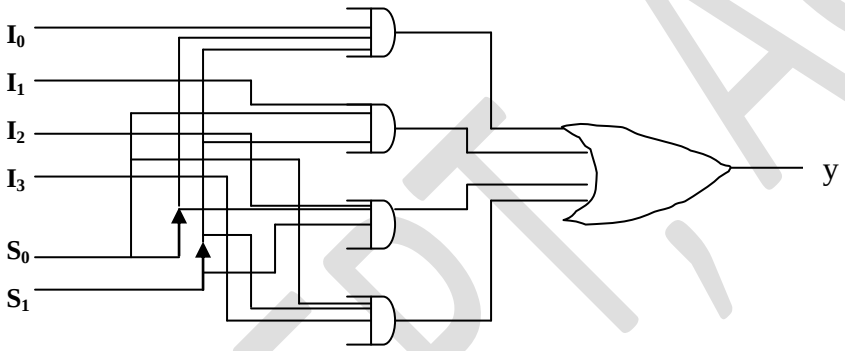
$A_1 = D_2+D_3+D_6$

$A_2 = D_4+D_5+D_6+D_7$



8 To 3 Line Encoders

Multiplexer:- Multiplexer वह Combinational Circuit है जो 2^n-Input Data lines से Binary Line Received करता हैं उसके एक Singal Output Lines पद Direct कर देता हैं। Particular Data line को सलेक्ट करने के लिए हमें Selection Input line की आवश्यकता होता हैं। 2^n-to-1 Multiplexer में 2^n InputLine हैं और n Input Selection Line हैं। इसे Mux भी कहते हैं।



4 to 1 line mwc

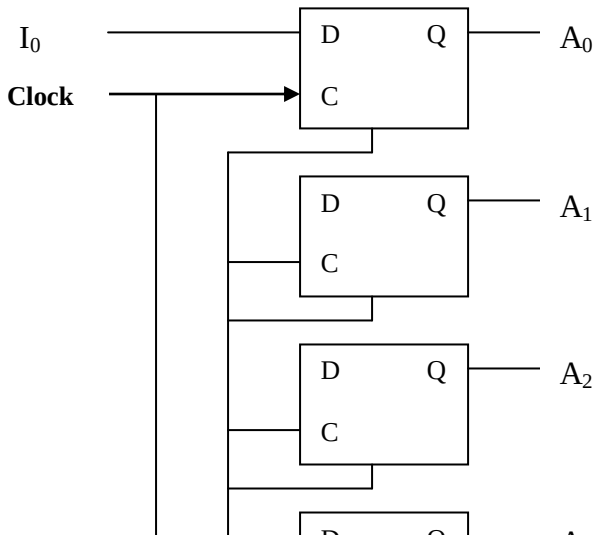
Function Table 4 to 1 line mwc

Input		Out put
S.	So	y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

2^n INPUT Line

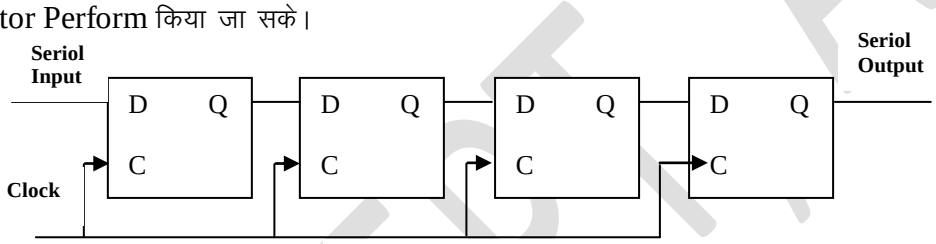
N Selection Line

Registers:- Flip-Flop के समूह को Register कहते हैं। एक N Bit Register में N Flip-Flop होते हैं जो N Bit को Install कर सकते हैं। Flip-Flop के साथ–2 एक Register में Data Processing कार्य करने के लिए Combinational Gates भी हो सकते हैं। इस प्रकार हम यह कह सकते हैं कि Register Flip-Flop व Gates का समूह होता हैं।



Shift-Register:- एक ऐसा Register जो दोनो दिशाओं में Binary Information को Shift कर सकता हैं, उसे Shift Register कहते हैं।

वास्तव में Shift Register में Flip Flop की एक बहुत बड़ी श्रृंखला होती हैं। जिसमें Flip-Flop का Output दूसरे Flip-Flop के लिए Input का कार्य करता हैं। सभी Flip-Flop Comman Clock Pulse REcive करते हैं ताकि Shift Operator Perform किया जा सके।



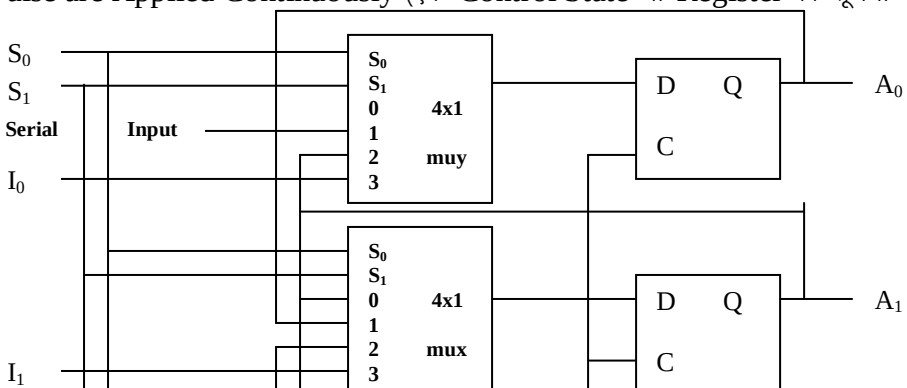
4 Bit Shift Register

एक Flip-Flop का Output Right Side वाले Flip Flop के लिए D Input के साथ Connect किया गया हैं। सभी Flip-Flop में Clock Pulse Common हैं। Serial Input यह Determation करता हैं। कि Left Most Posion से क्या Input आएगा। Serial Output Right Most Flip Flop से प्राप्त होगा।

Bidirection Shift Register With Parallel Load

एक Register जो सिर्फ एक ही दिशा में Shift कर सकता हैं। उसे Unidirectional Shift Register कहते हैं। तथा एक अन्य Register जो दोनो दिशा में Shift कर सकता हैं। उसे Biddirectional Shift Register कहते हैं। कई Register Parallel Transfer के लिए Input एवं Output Terminals प्रदान करते हैं। एक General Shift Register में निम्नलिखित विशेषताएँ होती हैं:–

(1) An Input for Clock Pulse is Synchormise all Operations सभी Operation को प्रारम्भ करने के लिए N Bits होते हैं।
(2) A Shift Right Operation & a serial Input line Associated With the Shift right Connect होते हैं।
(3) A Sift Left Operation & a Serial Input line associated with the Shift Left. (Sift Left के लिए S.L.O. एवं Serial Input lines आपस में Left Connect होते हैं।
(4) A Parallel Load Operation & in input lines Associated with the Parallel Transfer. (Parallel Transfer के लिए Parallel Load और Serial line आपस में Connect होते हैं।
(5) N Parllel Output lines.
(6) A Control State that leaves the in for Matlon in the Register Unchanged even Thogh Clock Pulse are Applied Continuously (एक Control State जो Register की सूचना को नहीं बदलती हैं।)



उपरोक्त चित्र 4-bit Bidirectional Shift Register With Parallel Load का हैं। जिसमें हर एक Stage के लिए एक D Flip Flop हैं और एक 4x1 mux प्रदान किया गया हैं। 2 Selection Input S_1 और S_0 D Flip Flop के लिए Multiplexer में से Data Input Select करते हैं। Selection Line Operation के male को Control करती हैं। जब S, S_0 = 00 होगा तो mux का Data Input 0 Select किया जाएगा। इसी प्रकार जब S, S_0 = 01 होगा तब हर mwc का Input का Input 1 Select होगा। इस प्रकार यह क्रम निरंतर चलता रहेगा।

Table For Register

Model Control		
S_1	S_0	Register Operation
0	0	No Change
0	1	Shift Right (down)
1	0	Shift Left (Up)
1	1	Parellel Load

Diagram चाहे किसी भी प्रकार से बनाया जाए। Shift Right Operation में Register Contents को Down Direction में Shift किया जाता हैं। और Shift Left में Register को Contents को Up Words किया जाता हैं।

Memory Unit

Memory Unit Storage Cells का Collection हैं जिसमें Circuit के साथ Associate करके Information को स्टोर या Retrieve किया जाता हैं। Memory Binary Information को जिन Group of bits में स्टोर करती हैं, उन्हें Memory Words कहते हैं। Words Memory की वह Entity हैं जिसमें Bits को स्टोर किया जाता हैं या Retrieve किया जाता हैं। Memory Words is a Group of 1's and o's होते हैं जो एक Number Instruction Code हैं, एक या एक से अधिक Alpha Numerica Character या अन्य Binary Coded Information को Represent करना हैं। A Group of 8 bits is Called a Byte अधिकतर Computer Memory Bytes के रूप में Words को Represent करती हैं।

एक Memory की आंतरिक संरचना के बारे में जानकारी उसके No. of Words एवं हर Word में No. of Bits के द्वारा प्राप्त की जा सकती हैं। हर एक Word को Memory में एक Indentification Number दिया जाता हैं जिसे Address कहते हैं। यह Address 0 से शुरू होता हैं और 2k1 तक हो सकता हैं। जहाँ k = Number of Address Lines हैं। किसी भी निश्चित Word को Memory में सलेक्ट करने के लिए K Bit Binary Address का प्रयोग Address Line पर किया जाता हैं। Memory के अन्दर एक Decoders होता हैं। जो इस Address को Accept करता हैं उस निश्चित Bit को Select करने के लिए मार्ग प्रशस्त कर देता हैं।

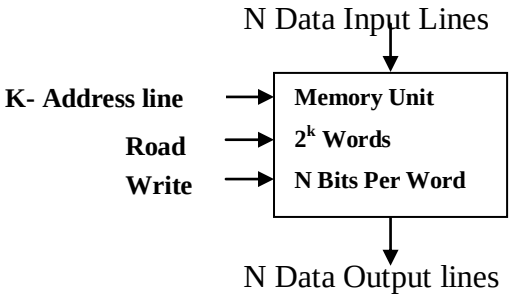
Example:- $1024 = 2^{10} = 1024$ 1 Memory Word के लिए हमें 10 Bit की आवश्यकता होती हैं।

1024 – Memory Word Address – 10 bit Computer System में मुख्यतः दो प्रकार की Memory काम में ली जाती हैं।

(1) RAM (2) ROM

(1) RAM:-Ram में सूचना को Store करने के लिए उसकी किसी भी Location का प्रयोग किया जा सकता हैं। अर्थात् किसी भी Location को ढूढँने का समय Same हैं। Memory और उसके वातावरण के बीच Communication स्थापित करने के लिए हमें निम्न की आवश्यकता होती हैं।

(1) Data Input/Output Lines
(2) Data Address Selection line
(3) Control Lines



Block Diagram of RAM

N Data Input Lines के द्वारा सूचना को Memory में Store किया जाता हैं। N Data, Out Line के द्वारा सूचना को Memory के बाहर संग्रहित किया जाता हैं। K-Address Line हमें K-Bits को Binary No. प्रदान करती हैं जो एक निश्चित Word को Represent करता हैं जिसे Memory में उपलब्ध 2^k Message में Choose किया जाता हैं। 2 Control Input Direction of Transfer की दिशा निर्धारीत करता हैं।

RAM में मुख्य रूप से दो Aperation होते हैं।

(1) Write
(2) Read

Write Signal के द्वारा Memory में डाटा लिखा जाता हैं। Write Signal Transfer In Operation को Input की Specify करता हैं एवं Read Signal Transfer Out Operation Output को Specify करता हैं।

Memory में एक नये Word को Store करने के लिए निम्न Steps इस प्रकार हैः–

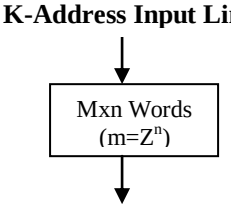
(1) जिस Word को लिखा जाता हैं उस Word का Binary Add; Address line पद डाल देना चाहिए।
(2) Data Bits को Data Input Lines पर डाल देना चहिए।
(3) Write Input को Activited कर देना चाहिए।

किसी Word को Memory से Read करने के लिए निम्न Steps इस प्रकार हैः–

(1) जिस Word को पढ़ना हैं उसके Binary Address को Address Line पर डाल देना चाहिए।
(2) Read Input को Activeted कर देना चाहिए।

ROM:- वह Memory Unit हैं जो सिर्फ Read Operation Perform करती हैं। इसमें डाटा को लिखने की क्षमता नहीं होती हैं अर्थात् ROM में Stored Binary Information Hardware Production के दौरान Permanent बना दी जाती हैं जिसे बदला नहीं जा सकता हैं। ROM में Special Internal Electronic Busses होते हैं जिन्हें Programed किया जा सकता हैं।

एक बार Pattern बनाने के बाद वह सदैव उसमें स्थित हो जाती हैं अर्थात् Power बंद हो जाने के बावजूद वह नष्ट नहीं होता हैं।



(Block Diagram of ROM)

एक Mxn ROM में Binary Cells का एक Array हैं जिसमें Memory Words और N Bits होती हैं। उपर दर्शाये गए चित्र में K-All Input Lines हैं जिनका प्रयोग Memory Words को Memory में से किसी एक Word को Select करने के लिए किया जाता हैं और N Output Lines हैं।

Types of ROM

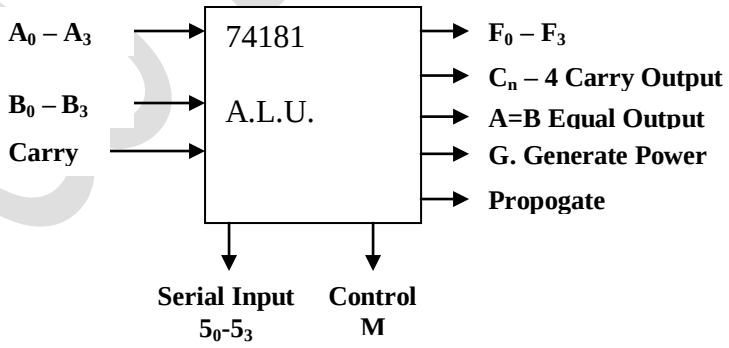
(1) P ROM
(2) E E P ROM
(3) E R ROM

Program Control:- निर्देश सदैव लगातार Memory Location में स्टोर होते हैं। जब भी CPU में Processing की जाती हैं तो निर्देशों को उन लगातार Memory Location से Betch किया जाता हैं एवं उन्हें क्रियान्वित किया जाता हैं। जब भी एक निर्देश को Memory से Fetch किया जाता हैं तो प्रोग्राम काउन्टर को Incriment कर दिया जाता हैं ताकि उसमें अगला Address का Refrence आ जाए। अर्थात् Program Counter में सदैव अगले निर्देश का Address होता हैं। इसी की सहायता से ही हमारा Control Fetch Cycle के अगले Step पर return होता हैं। दूसरी तरफ एक Program Control निर्देश उस प्रकार का निर्देश हैं कि जब इसे क्रियान्वित किया जाता हैं तो यह Program Counter की Address Value को बदल देता हैं और इस प्रकार यह Flow of Control बदल देता हैं।
वह Register जिसमें अगला निर्देश क्रियान्वित किया जाता हैं या हमारे Control Flow को Change कर सकता हैं उसे Program Control निर्देश कहते हैं।

निम्न Program Control Instrantis इस प्रकार हैं:–

Name	**Numeric**
(1) Branch	BR
(2) JUMP	JMP
(3) SKIP	SKP
(4) CALL	CALL
(5) RETURN	RET
(6) Compare (Subtraction)	CMP
(7) Test (By Adding)	TST

Airthmetic Logic Circuit:- अलग–2 Register के द्वारा Micro Operation Perform करने की बजाय Computer में सभी Unit जोड़ दिया जाता हैं जिसे A.L.U. कहते हैं। किसी भी निश्चित Micro Operation को Perform करने के लिए किसी निश्चित Register के Containts, Comman A.L.U. Input के रूप में दिए जाते हैं।



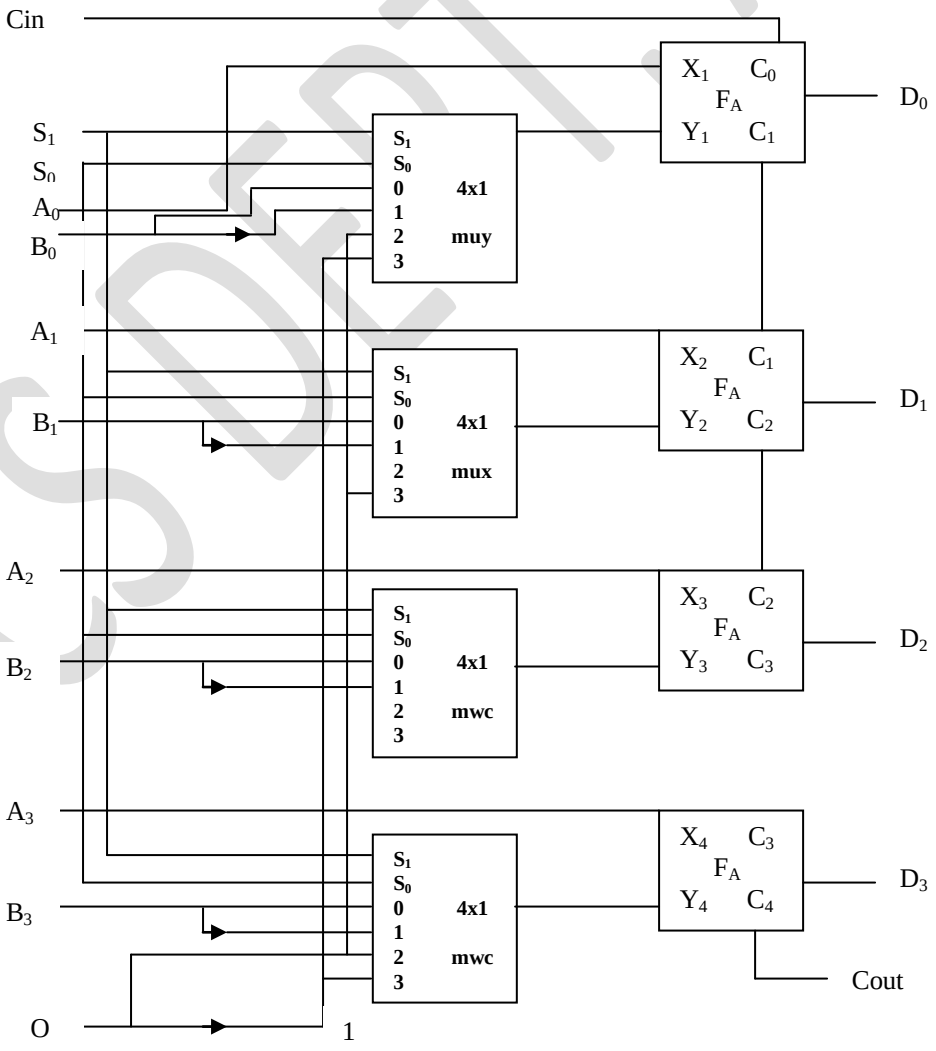
Airthmetical Circuit:- Airthmetic Micro Operation को एक Composite Airthmetic Circuit के द्वारा Execute किया जाता हैं Airthmetic Circuit का प्रमुख Componatit Pareller Adder हैं। Adder के Data Input को Control करके हम विभिन्न प्रकार के Airthmetic Operation Perform कर सकते हैं।

Airthometich Circuit Function Table

Select	Input	Micro Operation	Output

S_1	S_0	Cin	y		D=A+y+Cin
0	0	0	B	Add	D = A+B
0	0	1	B	Add with Carry	D = A+B+1
0	1	0	$\overline{B}$	Sub With Power	D = A+$\overline{B}$
0	1	1	$\overline{B}$	Sub	D = A+$\overline{B}$+1
1	0	0	0	Transfer A	D = A
1	0	1	0	Increment A	D = A+1
1	1	0	1	Decrement A	D = A-1
1	1	1	1	Transfer A	D = A

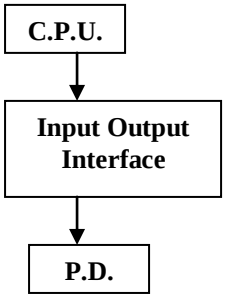
Micro Operation:- Micro Operation व Elementry Operation हैं जो Registers में संग्रहित डाटा पद Perform किया हैं।



4 –Bit Airthmetich Circuit

UNIT – IV

Input Output Interface Memory एवं Input Output Devices के बीच सूचना को Transfer करने का तरीका उपलब्ध करवाता हैं। जो भी Peripheral Devices Computer से Connect होते हैं उन्हें C.P.U. के साथ Interact करने के लिए Special Communication Links की आवश्यकता होती हैं। Communication Links की आवश्यकता होती हैं। Communication Links का मुख्य उद्देश्य कम्पयूटर और विभिन्न Devices के बीच जो असमानताएँ हैं उन्हें हटाना हैं।



असमानताएँ निम्न प्रकार की हो सकती हैः–

(1) Pariphoralls Electro Mechamical या Electro Magnetic Devices हो सकते हैं जिनका कार्य करने का तरीका C.P.U. एवं Memory से भिन्न हो सकता हैं। इसलिए हमें Signals को परिवर्तित करने की आवश्यकता हो सकती हैं।

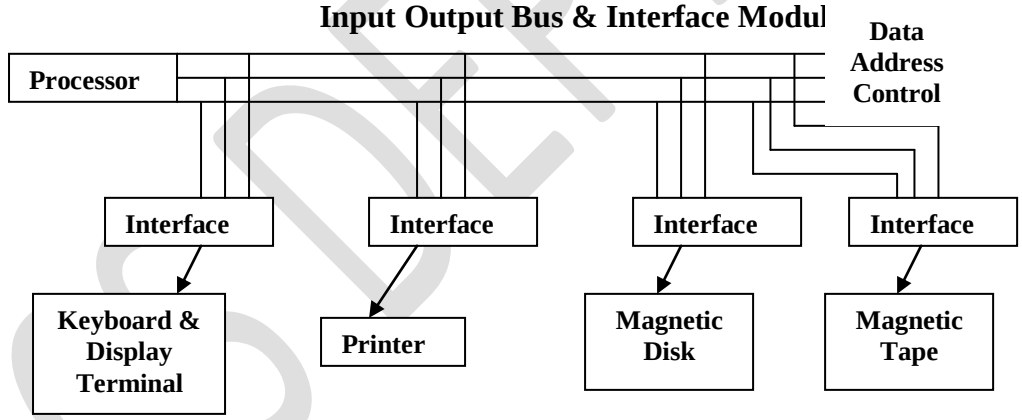
(2) Paripherals की Data Transfer Rate वह C.P.U. की तुलना में बहुत कम होती हैं।

(3) Perpherals के Data Code एवं Format C.P.U. एवं Memory के Word Format से अलग होते हैं।

(4) Peripherals का Operating Mode एक दूसरे से भिन्न होता हैं। इसलिए हमें हर Devices को Control करना पड़ता हैं ताकि एक Devices दूसरे Devices में बाधा नहीं पहुँचा सके।

ठस प्रकार की असमानताओं को हटाने के लिए Computer में Special Hardware Compoments का प्रयोग किया जाता हैं। Hardware Components को Interface कहते हैं।

Interface सदैव C.P.U. और Perphirals के बीच कार्य करते हैं। एवं सभी प्रकार के Input एवं Output Transfer को Supervision करते हैं।



Connection of Input bus to Input-Output एक टिपिकल Communication Link Devices एक Processor व Perpherals के बीच उपरोक्त चित्र में प्रदर्शित किया गया हैं। Input Output Bus में तीन विभिन्न प्रकार की Lines होती हैं।

(1) Data Line
(2) Address Line
(3) Control Line

हर एक Peripheral Devices हर एक Interface से जुड़ा होता हैं। हर Interface Input Output Bus के द्वारा प्राप्त किए गए Control को Decode करता हैं एवं यह पत्रा लगाता हैं कि यह Address किस Device का हैं एवं Perphords और Processor के बीच डाटा Transfer के बीच Signal प्रदान करता है। प्रदान करता हैं हर Peribheral का Controller होता हैं।

Peripherals Devices:- सभी Input एवं Output Devices को Peripherals Devices कहते हैं।

Input Output Command:- जब Interface Selected होता हैं तो उसका एक Function Code होता हैं, जिसे I/O Command कहते हैं।

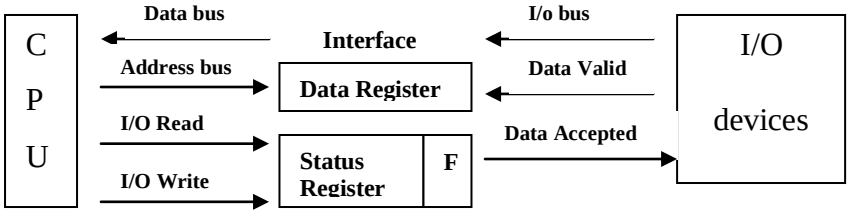
Control Command:- यह Paripherals को बताता हैं कि उसे क्या करना हो।
Status:- यह Peripheral और Interface की Condition को चैक करता हैं।
Output Data:- Output Data Base के द्वारा Data को Register में Transfer किया जाता हैं।
Input Data:- यह Peripheral तथा Interface से Data लेकर buffer Register में भेजता हैं।
Input Output V/s Memory Bus:- Processor Input Output Devices के साथ Communication करने के साथ–2 Memory Unit से भी Communication करता हैं।
Input Output bus की तरह ही एक Memory bus होती हैं जिसमें Data, Address एवं Read & Write Control Lines होती हैं। Memory तथा Input Output के बीच Communication स्थापित करने के लिए Computer Buses को तीन प्रकार से काम में लिया जाता हैं।

(1) दो Sepret Bus रखना एवं Memory के लिए एक I/O
(2) I/O तथा Memory के बीच Common Bus के लेकिन अलग–2 Control Line
(3) Common Control Line के साथ I/O तथा Memory के लिए का ही Common Line.

Input Output Processor:- Input एवं Output Devices में Control करने के लिए जिन Seprate का Processor का प्रयोग किया जाता हैं, उसे Input Output Processor कहते हैं। कई प्रकार के Computer में Data के लिए एक Seprate Interface Logic Combinded कर दिया जाता हैं। Input Output, DMA एवं Interface Facitatus
Isolated Verous Memory Mapped Input Output Insloated Input Output:- इसमं Read एवं Write करने के लिए अलग–2 Lines का उपयोग किया जाता हैं। यह Memory वाले डाटा को Memory Unit पर तथा Interface वाले डाटा को Interface Unit पद भेजता हैं। इसके पास जिसका Address आता हैं उसी को भेज दिया जाता हैं। यह एक समय में या तो Data Read करता हैं या फिर Write करता हैं। यदि Bus Common हैं तो Read एवं Write की Lines अलग–2 होती हैं। Input Output को Memory का Part मानते हैं। इसलिए C.P.U. Memory Word को Change कर सकता हैं।
हर Interface Read एवं Write करने के लिए अलग–2 Register की आवश्यकता होती हैं।
Memory Mapped Input Output:- Inslated Input Output Method Memory or Input Output Address को अलग–2 रखता हैं।
इसका एक अन्य तरीका यह हो सकता हैं। कि दोनो मेमोरी और Input Output के लिए एक ही Address Space प्रयोग किया जाता हैं। इस प्रकार की स्थिति उन Computer में होती हैं जो मेमोरी और Input Output Address के बीच अन्तर नहीं रखते हैं और उनमें रीड और Write Signals पद ही सेट होता हैं। इस प्रकार के Configuration को Memory Mapped Input Output कहते हैं।
Modes of Data Transfer:- बाहय उपकरण से इनपुट लेने के लिए विभिन्न प्रकार के Modes का प्रयोग किया जाता हैं। यह सारा कार्य Memory में होता हैं। Memory से डाटा C.P.U. में जाता हैं। इसके तीन तरीके हैं।

(1) Programed Input Output
(2) Interupt –Initiatied Input Output
(3) Direct Memory Access (DMA)

(1) Programed Input Output:- यह कम्पयूअर प्रोग्राम में लिखे होते हैं। हर डाटा के लिए एक निर्देश होता हैं। पहला Transfer C.P.U. तथा Periphalis के बीच तथा दूसरा Transfer C.P.U. तथा मेमारी के बीच होता हैं।
(2) Interupt-Initiated Input Output:- C.P.U. हमेशा Processing के लिए तैयार रहता हैं। हम जानते हैं कि CPU का टाइम अत्यंत महत्वपूर्ण हैं इसलिए CPU का टाईम बिल्कुल भी खराब नहीं किया जाना चहिए ऐसा करने के लिए Interupt-initiated का उपयोग करते हैं। Intrupt-initiated या तो Input devices से आता हैं या फिर Output devices से आता हैं। जब भी कोई Interupt आता हैं तो C.P.U. उसे प्राथमिकता देता हैं जो कार्य कर रहा हैं उसे बीच में से छोड़कर पहले Interupt को Handle करता हैं। उसके पश्चात् पुनः अपना कार्य शुरू करता हैं।
(3) Direct Memory Access (DMA):- Programed Input Output में C.P.U. और Preripale के बीच Data Transfer होता हैं। DMA में हमें एक Interface प्रदान किया जाता हैं जिसके द्वारा हम मेमोरी में डाटा रीड एवं Write कर सकते हैं। इस प्रकार का Data Transfer Memory Bus के द्वारा किया जाता हैं।
C.P.U. Starting Address Interface को Supply के साथ ही Transfer को Initiate कर देता हैं एवं उसके पश्चात् अपना कार्य करने लग जाता हैं। जब Transfer शुरू होता हैं तो DMA Memory bus से Memory Cycles की मांग करता हैं। जब Memory Contoraler के द्वारा यह मांग पूर्ण कर दी जाती हैं तो DMA Directly Data को Memory में Transfer कर देता हैं।



F = Flag bit

Data Transfer From Input Output devices to C.P.U.

Synchoronous & Asynchoronous Data Transfer

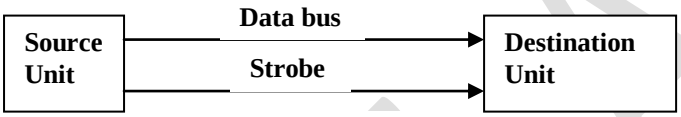
Synchoronous:- जब C.P.U. or Input Output Interface के Register के बीच Comman Clock होता हैं तो उसे Synchoronous कहते हैं।

Asynchoronous:- जब C.P.U. or Input Output Interface के Register के बीच अलग–2 Clock होती हैं तो उसे Asynchoronous कहते हैं।

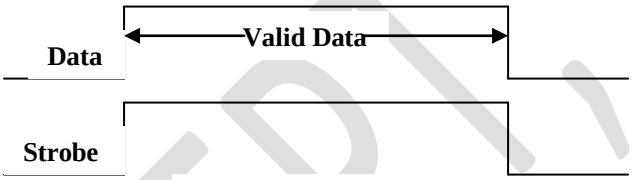
Asynchoronous Data Transfer दो तरीकों से किया जाता हैं।

(1) Strobe:- यह एक Asynchoronous Data Transfer का तरीका हैं इसमें Strobe (जो कि एक Pulse हैं।) के द्वारा Data Transfer किया जाता हैं। इसका नुकसान यही हैं कि हमें यह पता नहीं चलता हैं कि डाटा सही जगह पर सही रूप से पहुँचा हैं अथवा नहीं। इसलिए यह तरीका अविश्वसनीय हैं।

(2) Handshaking:- Asynchronous Data Transfer होकर सही जगह पर पहुँचता हैं या फिर डाटा पहुँच गया हैं। इस प्रकार के Method को Handshaking कहते हैं। इसमें हमें यह ज्ञात हो जाता हैं कि डाटा पहुँचा हैं अथवा नहीं। Hand shaking एक लचीला एवं विश्वसनीय System हैं इसमें किसी भी Error का आसानी से पता लगाया जा सकता हैं।

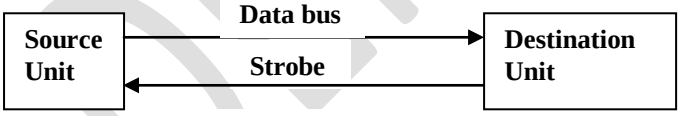


(a) Block Diagrame

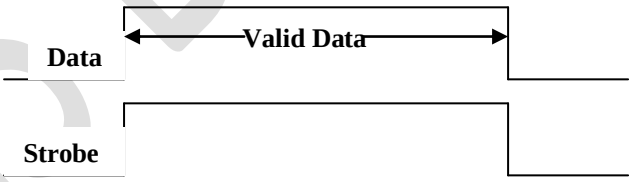


(b) Timing Diagrame

Source-Initiated Strobe for Data Transfer

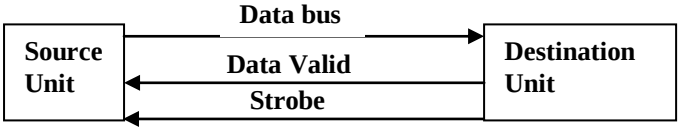


(a) Block Diagrame

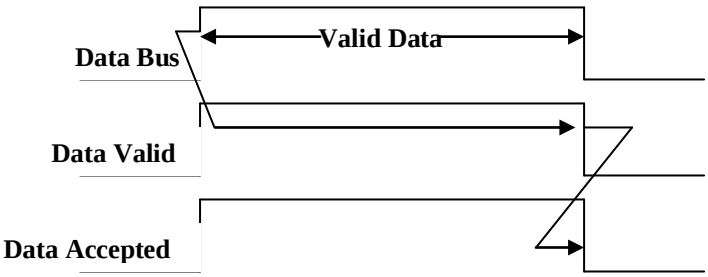


(b) Timing Diagrame

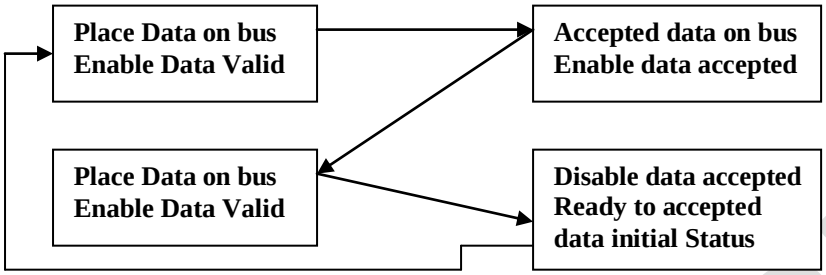
Destination-intiated Strobe for data Transfer



(a) Block Diagrame



(b) Timing Diagrame



(c) Sequence of Event

Source initited Transfer Using hand shaking

(3) Asynchronous Serial data Transfer:- इसमें Data Serial bit by bit Transfer होता हैं। यह एक समय में एक Bit Transfer करता हैं। इसकी Speed काफी कम होती हैं। यह कम खर्चीला होता हैं तथा अधिक दूरी के लिए काम में आता हैं।

समान्तर में एक साथ 4 bit का Data Transfer होता हैं। समान्तर Data Transfer में हर Bit का अलग–2 Batch होता हैं। इस प्रकार का Transfer महँगा होता हैं। तथा Speed Fast होती हैं

Serial Transfer में कार्य को Fast करने के लिए समान्तर Transfer का उपयोग किया जाता हैं।

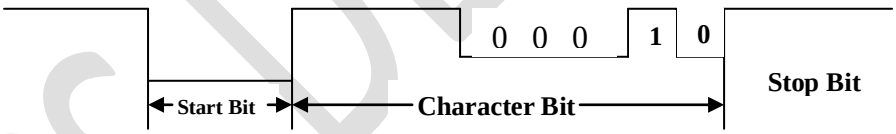
Asynchronous Serial Transfer:- इस प्रकार के Transfer में तीन

(1) Start Bit
(2) Character Bit
(3) Stop Bit

(1) Start Bit:- पहले Bit को Start Bit कहते हैं। यदि पूरी Line में 1 हैं इसका अर्थ यह हैं कि Data Transfer नहीं हो रहा हैं और हमारी Communication Line idle पड़ी हैं।

(2) Characterbits:- Character Bit हमेशा Start Bit के पीछे होती हैं। अर्थात् यहाँ हमारा डाटा शुरू हो जाता हैं।

(3) Stop bit:- Stop bit हमेशा 1 होती हैं और यह हमेशा Last में होती हैं।

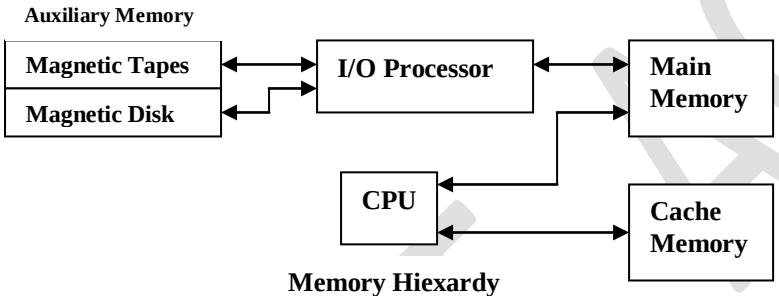


Baud Rate:- एक Second में गिनती Bits Transfer होती हैं उनको Baud Rate कहते हैं।

UNIT-5

Memory (Chain) Heirarchy:- Digital Computer में Memory Unit एक महत्वपूर्ण तत्व हैं। जिसके द्वारा डाटा एवं Programes को Store किया जाता हैं। एक छोटे Computer में Limited Application के कार्य पूर्ण किए जा सकते हैं अर्थात् हमें अतिरिक्त Storage क्षमता की आवश्यकता नहीं होती हैं। यदि General Computers में Main Memory के साथ–2 कुछ अतिरिक्त Storage Unit लगा दी जाए तो वह अपना कार्य और प्रभावशाली तरीके से कर सकता हैं क्योंकि Main Memory काफी महंगी होती हैं इसलिए Back-up data के लिए कम खर्चीले Storage devices का प्रयोग किया जाता हैं। वह Memory Unit जो C.P.U. से Directly Communication करती हैं उसे Main Memory कहते हैं। वे devices जो Back-up Storage की सुविधा प्रदान करते हैं उन्हें Aqciliary Memory Devices जिनका Computer System में उपयोग किया जाता हैं Disk व Tape हैं। इनका प्रयोग System Programe, Large Data Files और अन्य Backup सूचना को Store करने के लिए किया जाता हैं।

एक अन्य प्रकार की Memory जिसका Computer System में उपयोग किया जाता हैं एवं जिसकी Speed बहुत अधिक होती हैं उसे Cache Memory कहते हैं। इस प्रकार की Memory का उपयोग Computer की Processing की Speed को बढ़ाने के लिए किया जाता हैं।



Memory Hiexardy

Auxilary Memory:- Computer Systems में जो Common Auxiliary Memory Device काम में लिए जाते हैं वे हैं Disk व Tape अन्य और कई Componenets हैं जैसे – Magnetic Durm, Bubble Memory जिनका प्रयोग काम में लिया जाता हैं। इन सभी Auxiliary Machin समझने के लिए हमें Magnetices, Electronics एवं Electon Magnetic का ज्ञान होना आवश्यक हैं। हालांकि इन सभी devices की Physical Properties काफी Comples होती हैं लेकिन सभी Logical Properties को परिभाषित करके कुछ Parameters द्वारा उन्हें Compare किया जा सकता हैं।

इन Devices की प्रमुख विशेषताएँ इस प्रकार हैं:–

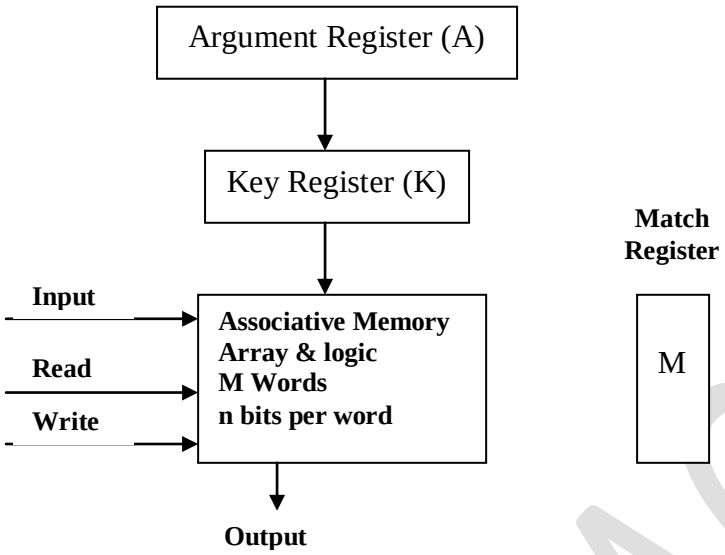
	कम दूरी	अधिक दूरी
(1) Access Mode	(Random)	(Serial)
(2) Acces devices	(कम)	
(3) Transfer Rate	(ज्यादा)	
(4) Capicity	(ज्यादा)	
(5) Cost	(कम)	

(1) Magnetic Drum:- Magnetic Drum, Magnetic Disk की तरह ही हैं। इसमें एक High Speed Rolating Surface होता हैं जिस पद Magnetic पदार्थ की परत होती हैं। Drum का Rolating Surface एक बेलनाकार होता हैं। Recording Surface एक निश्चित गति से Rotated होती हैं। और Access Operation के दौरान उसे Start एवं बंद नहीं किया जा सकता हैं। Bits को Magnetic Spotes के रूप में Surface पर Record किया जाता हैं। इस प्रक्रिया को Write head कहते हैं। Sotred bit को Magnetic Field के बदलाव के कारण पहचान किया जाता हैं। इस प्रक्रिया को Read Head कहते हैं। Disk की तुलना में Drum की Size (Capacity) कम होती हैं। इसलिए Drum की बजाय disk का उपयोग कम्पयूटर में अधिक किया जाता हैं।

<u>Associative Memory</u>:- एक ऐसी Memory जिसे उसके डाटा द्वारा Access किया जाता हैं उसे Associative या Contente Addressable Memory कहते हैं। (AM)

जब भी कोई Word Type करते हैं और उसका Address नहीं लिखते हैं तो उसे Associative Memory के द्वारा खोजा जाता हैं। इसके पहले यह बताना पड़ता हैं कि इसे क्या खोजना हैं क्योंकि इसमें Address नहीं होता हैं। कार्य को Fast करने

के लिए Associative Memory का उपयोग किया जाता हैं। यह बहुत मंहगी होती हैं। यह स्वयं के Data के द्वारा ही चलती हैं। इसका प्रत्येक Call Data को Store करने की क्षमता रखता हैं।

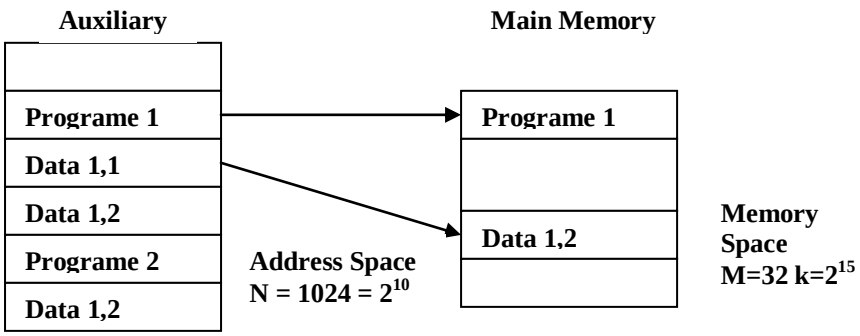


Block Diagrame of Associative Memory:- Memory में जो Word होते हैं उन्हें Argument Register से Compare करते हैं। Key Register Particular Key को Choose करने की Facitaly देता हैं जिससे पूरा Argument Memory से Compare किया जाता हैं। जब Key Register के सभी Contente 1 होते हैं तब Argument के वे Bits जो कि 1 हैं सिर्फ वे ही Compare किए जाते हैं। वह यह बताती हैं कि Memory Contentes को किस प्रकार Reference किया जाए।

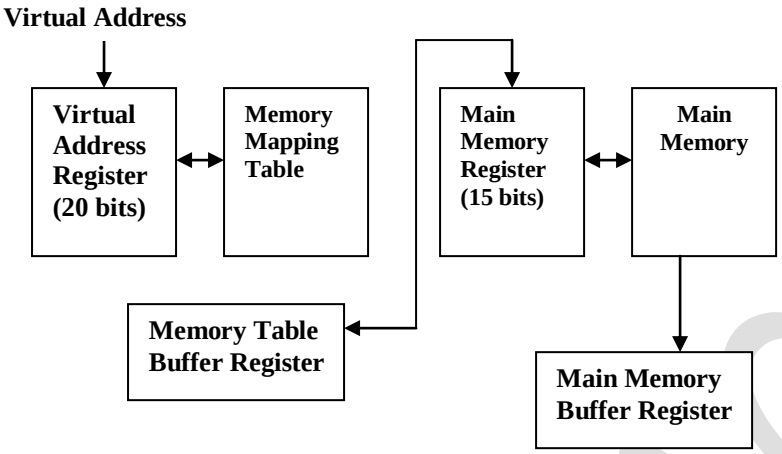
Virtual Memory:- Memory Chain System में प्रोग्राम व डाटा को पहले Awiliry Memory में स्टोर किया जाता हैं। CPU की आवश्यकता के अनुसार। प्रोग्राम या डाटा का कुछ हिस्सा Main Memory में लाया जाता हैं। Virtual Memory एक Concept हैं लाया जाता हैं। प्रयोग बड़े Computer System में किया जाता हैं जो User को बड़े प्रोग्राम बनाने की सुविधा प्रदान करते हैं। जैसे उनके पास बहुत अधिक मात्रा में Memory उपलब्ध हैं जितनी कि Auxiliary Memory हैं। हर एक Address जिसे C.P.U. के द्वारा Refer किया जाता हैं। उसे एक Address Mapping की प्रक्रिया से गुजरना पड़ता हैं जो Virtual Address को Main Memory के Physical Address में परिवर्तित कर देता हैं। Virtual Memory Programers को ऐसा आभास कराती हैं कि उनके पास बहुत अधिक मात्रा में Main Memory उपलब्ध हैं। परन्तु वास्तव में Main Memory बहुत कम मात्रा में होती हैं।

वह Address जो Programers के द्वारा प्रयोग में लिया जाता हैं। उसे Virtual Address कहते हैं।

वह Address जो Main Memory में होता हैं उसे Physical Address कहते हैं। Physical Address के समूह को Memory Space कहते हैं।

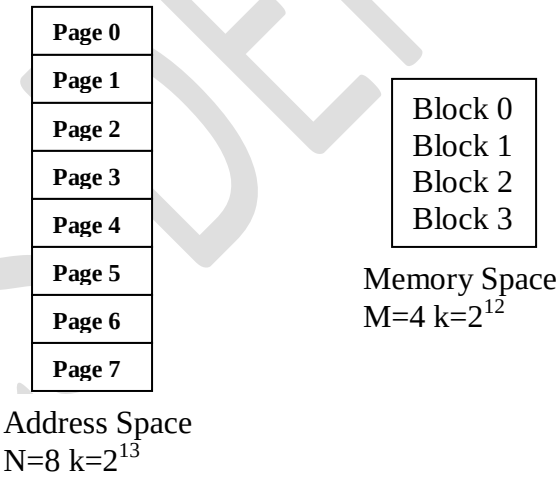


Relation Between Address & Memory Space in a Virtual Memory System.

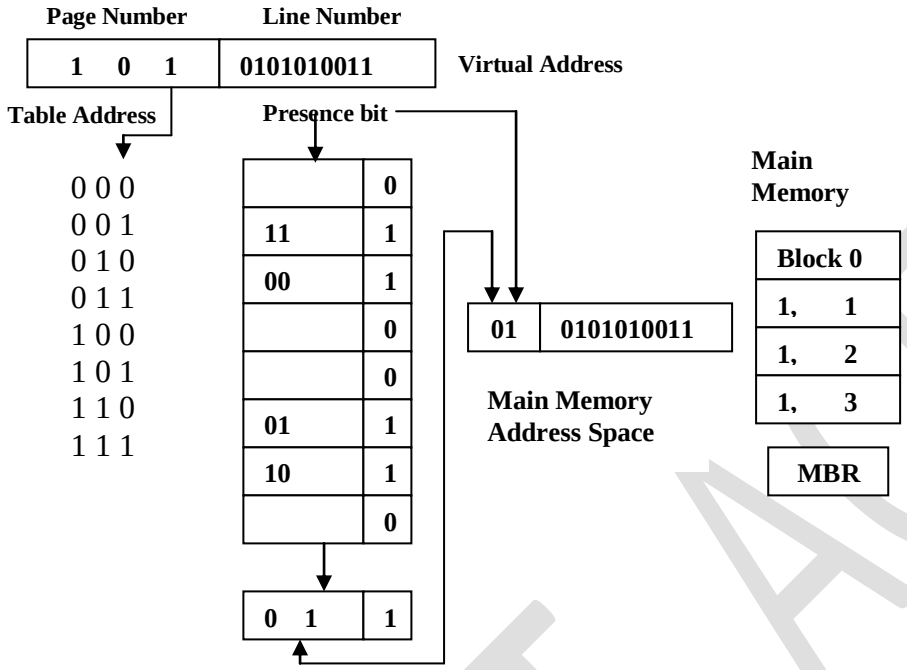


Memory Table For Mapping A Virtual Address

Address Mapping Using Pages:- Address Mapping के Table Implimention को हम सरल कर सकते हैं। यदि Address Space और Memory Space की सूचा Fixed Size के Group में divided हो या विभाजित हैं। Physical Memory को Equal Size में Fixed Blocks में divided किया जाता हैं जिनकी क्षमता 64 से 4096 Words हो सकती हैं। एक ही Size के Address Spaces के समूह को Pages कहते हैं। Programe का Part Awuiliary Memory से Main Memory में Equal Sized Pages के Record के रूप में Transfer होते हैं। इन Blocks को कई प्रकार Page Frame भी कहते हैं।



Address Space & Memory Space Split into Groups of 2k Words.

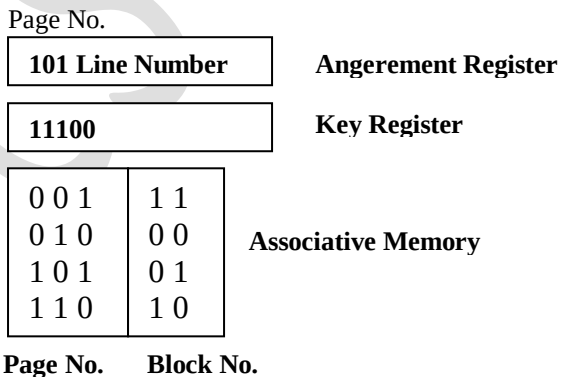


Memory Page

Memory Table In a Paged System

Associative Memory Page Table:- एक Ram Page Table में Storage Utiliagtion का उपयोग प्रभावशाली तरीके से नहीं किया जाता हैं। वास्तव में एक System जिसमें n Pages एवं m block होते हैं। उसमें n location जो m Block को Locate कर सके उतनी Memory Page Table की आवश्यकता होती हैं। इस विधि में ऐसा हो सकता हैं बाकि Location वाली रह जाए अर्थात् उनका उपयोग नहीं किया जा रहा हैं।

इसे Page Table को प्रभावशाली ढंग से बनाया जा सकता हैं। एक Page Table उतने Words की बनाई जए जितने Words Main Memory में Block हैं। इस प्रकार हम Memory की साईज कम कर सकते हैं और सभी Locations का पूर्णतः उपयोग कर सकते हैं। इस प्रकार की विधि में Associtive Memory का उपयोग किया जाता हैं जिसमें हर एक मेमोरी वर्ड के साथ एक पेज नम्बर और उसका Block No. होता हैं। हर एक Word की Page Field को Virtual Address के Page No. के साथ तुलना की जाती हैं। यदि हमें Match मिल जाता हैं जो उस Word को Memory से पढ़ लिया जाता हैं।



An Associative Memory Page Table

Page Replacement:- Virtual Memory System Hardware or Software Technology का Combination हैं। Memory Management Software System Memory Space के Utilization से संबंधीत सभी Software Operation को Handle करता हैं।

It Must Decided:-

(1) किसी पेज को Main Memory से Remove किया जाए ताकि उसकी जगह किसी नए पेज को लाया जा सके।
(2) किस वक्त नये पेज को Auxiliary Memory से Main Memory में Transfer किया जाए।
(3) Main Memory में उस पेज को कहाँ रखा जाए।

इस प्रकार Vritual Memory से Architeds में Hardware Mapping Machigm और Memory Management Software होते हैं।

जब प्रोग्राम Execution होना प्राप्त कर देता हैं तो एक या एक से अधिक Pages को Main Memory में Transfer कर दिया जाता हैं एवं Page Table को Set कर दिया जाता हैं जो इनकी स्थिति के बारे में हमें बताता हैं। एक प्रोग्राम मेन मेमोरी से तब तक चलता रहता हैं। जब तक प्रोग्राम उस पेज के लिए Reference नहीं भेजे जो कि Auxiliary Memory में हैं। इस प्रकार कि स्थिति को Page Fault कहते हैं। जब Page Fault होता हैं तो प्रोग्राम का Exectuion कुछ देर के लिए रूक जाता हैं जब तक की वह Required Page Main Memory में हीं आ जाता हैं। इस स्थिति में एक नया Page Auxiliary Memory से Main Memory में लोड किया जाता हैं। यदि Main Memory Full हैं तो किसी एक पेज को हटाया जाता हैं। किस पेज को Main Memory से हटाया जाए इसके लिए निम्न Algorithems हैं।

(First in First out)

(1) FIFO:- जो पेज पहले मेमोरी में आया हैं उसे पहले Main Memory से हटाया जाएगा।
(2) LRU (Least Recently Used:- जो पेज अभी सबसे कम काम में आ रहा हैं उसे सबसे पहले Memory से हटाया जाएगा।

Cache Memory:- Large Number of Progammer एक निश्चित समय में मेमोरी को Refer करते हैं। यह Reference Memory के एक निश्चित Area में होते हैं। इस प्रकार की Property Localty of Reference कहते हैं।

प्रेग्राम के कुछ Active हिस्से एवं डाटा को एक Fast Memory में रख दिया जाता हैं जिससे Average Access Memory time कम हो जाता हैं। इसे प्रोग्राम का Total Executaion time कम हो जाता हैं। इस Fast Memory को Cach Memory कहते हैं। इसे सदैव C.P.U. एवं Main Memory के बीच रखा जाता हैं।

Performance of Cache Memory:- Performance of Cache Memory bit ratis के द्वारा मापा जाता हैं। यदि हमें Data Cache Memory में मिल जाता हैं तो उसे hit कहते हैं अन्यथा उसे Miss कहते हैं।

$$\text{Hitratis} = \frac{\text{Total No. of Hits}}{\text{Total No. of hits+Miss}}$$

$$y- = \frac{10}{10+40}$$

$$= \frac{10}{50}$$

$$= 0.25 \text{ Meno Seconds}$$

Mapping of Cache Memory:- The Transformation of data from Main Memory to Cache Memory is refered as Mapping Processor. There are three types Mapping:-

(1) Associative Mapping
(2) Direct Mapping
(3) Set Associative Mapping

(1) Associative Mapping:- Associative Memory Address or Data को एक साथ Memory Word में स्टोर करती हैं। Address Value 15 Bit की होती हैं। 5 digit Octal No. में होती हैं। 12 Bit का Word और 4 digit Octal No. में होती हैं।

15 Bit का CPU Address Argument Register में रखा जाता हैं और उसे Associative Memory में Search किया जाता हैं। यदि हमें वह Address मिल जाता हैं तो उसका 12 bit का डाटा पढ़ लिया जाता हैं और उसे CPU के पास भेज दिया जाता हैं यदि हमें यह Match नहीं मिलता हैं तो Word को Main Memory से Access किया जाता हैं। इस प्रकार Address Data Paer को Associative Cache Memory में Transfer किया जाता हैं।

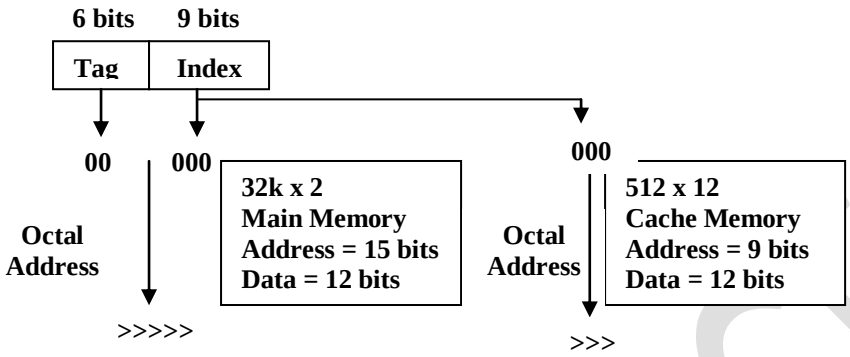
C.P.U. Address (15 Bits)

Argument Register

Address →	**Data** →
01000	**3450**
02777	**6710**
22345	**1234**

Associative Mapping Cache Memory
(All Number in Octal)

(2) Direct Mapping:- RAM की तुलना में Associate Memory काफी मंहगी होती हैं।

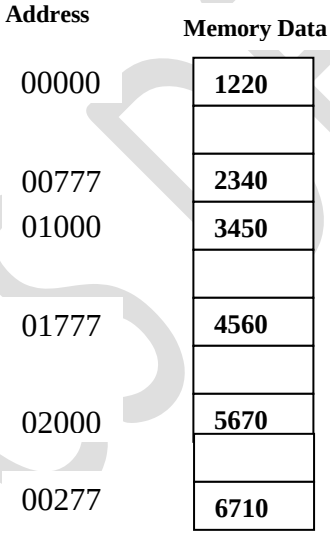


Address Relationship Between Main & Cache Memory

C.P.U का Address 15 Bits का होता हैं। जिसे दो भागों में विभाजित कर दिया जाता हैं। 9 Bits का Index तथा 16 Bits का Tag Field होता हैं। Index Field की संख्या Cache Memory के Address Bits की संख्या के बराबर होती हैं। जब भी C.P.U. कोई Memory Request Genrate करता हैं तो Cache Memory से डाटा को Access करने के लिए Index Field का प्रयोग किया जाता हैं।

C.P.U. Address के Tag Field को Cache Memory में से रीड किए गए Word के साथ Compare किया जाता हैं। यदि दोनों Tag Match हो जाते हैं तो Bit होता हैं अर्थात् वह Data Cache Memroy में ही हैं अन्यथा वह Miss हो जाता हैं अर्थात् उस Word को Main Memory से पढ़ लिया जाता हैं। Direct Mapping का सबसे बड़ नुकसान यही हैं कि यदि हमें दो या दो से अधिक Words जिनके Address का एक ही Index No. हैं तो उस परिस्थिति में Hit Ratio काफि कम हो जाता हैं।

Main Memory



(a) Main Memory

Index Address	Tag	Data
0 0 0	**0 0**	**1220**
777	**02**	**6710**

Cache Memory

Direct Mapping Cache organization

(3) Set Associative Mapping:- Set Associative Mapping, direct Mapping Orinzation का ही Improvement हैं। इसमें Cache Memory का एक Word ही Index Address में एक से अधिक Memory Word को स्टोर कर सकता हैं। हर एक Data Word के साथ उसका Tag भी Stored किया जाता हैं और Tag Cache Memory में Tag Data items को set कहते हैं।

Index	Tag	Data		Tag	Data
000	01	3450		02	5670

777	02	6710		00	2340

Two-Way Set Associative Mapping Cache

जब भी C.P.U. एक Memory Request करता हैं तो Index Value का प्रयोग Cache को Access करने के लिए किया जाता हैं। C.P.U. Address के Tag Field को दोनो Cache के दोनो Tag के साथ Compare किया जाता हैं। यदि वे दोनों Match हो जाते हैं तो Hit होता हैं। अन्यथा Miss होता हैं। जैसे–2 Set की साईज बढ़ती जाती हैं। Hit Ratio भी बढ़ता जाता हैं और जब Set Full हो जाता हैं या वह Data Cache में नहीं होता हैं तो Miss होता हैं।

Writing into cache:- Cache में लिखने के दो तरीके हैं:–

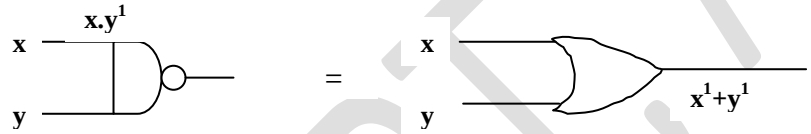
(1) Write Through
(2) Write back

(1) Write Through:- सबसे आसान व सबसे Common तरीका यह हैं कि जब भी Memory में Write Operation किया जाता हैं अर्थात् जब भी Data Cache Memory में लिखा जाता हैं तो साथ–2 Main Memory में भी Updation करना चाहिए। इस विधि को Write Through विधि कहते हैं। इसका फायदा यह हैं कि हमारी Main Memory सदैव Update रहेगी अर्थात् जो Data Cache Memory में हैं Main Memory में भी होगा।

(2) Write Back:- एक अन्य तरीका Write Back हैं। इस Method में Write Operation के दौरान सिर्फ Cache में ही Write किया जाता हैं। Cache में जिस Location में डाटा राईट किया जाता हैं उसे एक Flag के द्वारा Mark कर दिया जाता हैं। ताकि जब भी वह Word Cache से हटाया जाएगा तो उसे Main Memory में कॉपी कर दिया जाएगा।

De Morgan's Therom

(1) $(x.y)^1 = x^1+y^1$



(2) $(x+y)^1 = x^1.y^1$

(3) $(x+y+z)^1 = x^1.y^1.z^1$

(4) $(x.y.z)^1 = x^1+y^1+z^1$