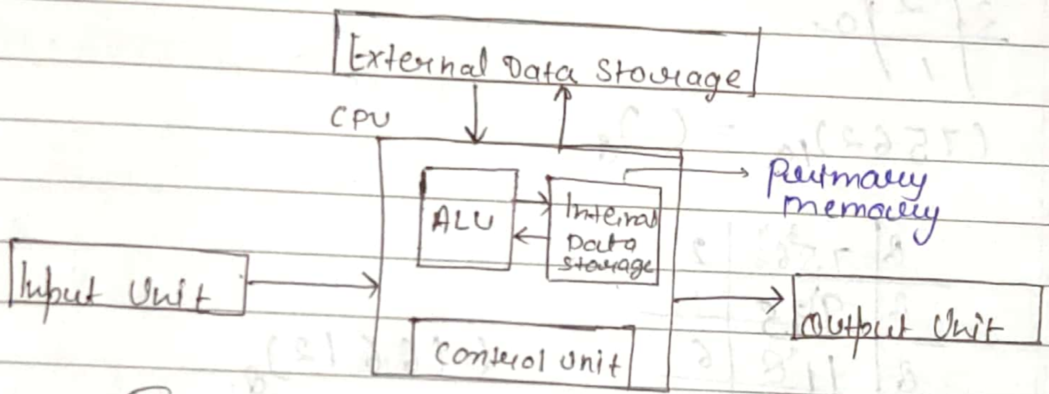
Computer Organisation (BCSC 0005)

Date / /
Page

Objective :-

→ How Computer Works
→ What are different components in Computer.
→ And how all those components are connected together to form a complete computer System.

Secondary Memory



• FUNCTIONAL BLOCKS OF COMPUTER

→ Number Representation

• Decimal
• Binary
• Octal
• Hexadecimal.

(i) $(100)_{10} = (\ )_2$

$= (1100100)_2$.

Page

(i) $(175)_{10} = (\ )_2$.

2	175	1
2	87	1
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
	1	

$(10101111)_2$.

(ii) $(7562)_{10} = (\ )_8$.

8	7562	2
8	945	1
8	118	6
8	14	6
	1	

$(16612)_8$.

(iii) $(1938)_{10} = (\ )_{16}$

16	1938	2
16	121	9
	7	

$(792)_{16}$.

(iv) $(50)_{10} = (\ )_3$

3	50	2
3	16	1
3	5	2
	1	

$(1212)_3$

Date __/__/__
Page ____

$\Rightarrow (1212)_3 = (\ )_{10}$

$\cancel{2 \times 10^0 + 1 \times 10^1 + 2 \times 10^2 + 1 \times 10^3}$

$\cancel{2 + 10 + 200 + 1000}$

$\cancel{1212}$.

$2 \times 3^0 + 1 \times 3^1 + 2 \times 3^2 + 1 \times 3^3$

$2 + 3 + 18 + 27$

50.

(v) $(41.6875)_{10} = (\ )_2$.

2	41	1
2	20	0
2	10	0
2	5	1
2	2	0
	1	

$(101001)_2$.

$0.6875 \times 2 = 1$

$0.375 \times 2 = 0$

$0.75 \times 2 = 1$

$0.5 \times 2 = 1$

$(101001.1011)_2$.

8 4 2 1

(vi) $(121)_8 = (\ )_2$.

$(001010001)_2$.

(vii) $(A251)_{16} = (\ )_2$.

$(1010\ 0010\ 0101\ 0001)_2$.

(viii) $(121)_8 = (\ )_{16}$

$8 \longrightarrow 2 \longrightarrow 16$.

Date __/__/__
Page ____

Questions:-

1). Convert:

a). $(101110)_2 \rightarrow (\ )_{10}$

$0 + 1\times2^1 + 1\times2^2 + 1\times2^3 + 0 + 1\times2^5$

$= 2 + 4 + 8 + 32$

$= (46)_{10}$.

b). $(110110100)_2 \rightarrow (\ )_{10}$

$0 + 0 + 1\times2^2 + 0 + 1\times2^4 + 1\times2^5 + 0 + 1\times2^7 + 1\times2^8$

$4 + 16 + 32 + 128 + 256$

$\Rightarrow (436)_{10}$.

c). $(12121)_3 \rightarrow (\ )_{10}$

$1\times3^0 + 2\times3^1 + 1\times3^2 + 2\times3^3 + 1\times3^4$

$1 + 6 + 9 + 54 + 81$

$(151)_{10}$.

d). $(4310)_5 \rightarrow (\ )_{10}$

$0 + 1\times5^1 + 3\times5^2 + 4\times5^3$

$\Rightarrow 5 + 75 + 500$

$\Rightarrow (580)_{10}$.

e). $(50)_7 \rightarrow (\ )_{10}$

$0 + 5\times7^1$

$(35)_{10}$.

Date / /
Page

(f). $(198)_{12} \rightarrow (\ )_{10}$

$8 \times 12^0 + 9 \times 12^1 + 1 \times 12^2$

$8 + 108 + 144$

$(260)_{10}$.

(g). $(7562)_{10} \rightarrow (\ )_8$.

8	7562	2
8	945	1
8	118	6
8	14	6
	1	

$(16612)_8$.

(h). $(1938)_{10} \rightarrow (\ )_{16}$

16	1938	2
16	121	9
	7	

$(792)_{16}$.

(i) $(175)_{10} \rightarrow (\ )_2$.

2	175	1
2	87	1
2	43	1
2	21	1
2	10	0
2	5	1
2	2	0
	1	0

$(10101111)_2$.

(j). $(FBA7C2)_{16} \rightarrow (?)_2 \rightarrow (?)_8$

$(111110111010011111000010)_2$.

$(76723702)_8$.

2).

a). $(10110.0101)_2 \rightarrow (\ )_{10}$.

$0+1\times2^1+1\times2^2+0+1\times2^4 . 0\times2^{-1}+1\times2^{-2}+0\times2^{-3}+1\times2^{-4}$

$2+4+16 . 0.25+0.0625$

$(22.3125)_{10}$

b). $(16.5)_{16} \rightarrow (\ )_{10}$.

$1\times16^1+6\times16^0 . 5\times16^{-1}$

$6+16 . 0.3125$

$(22.3125)_{10}$

c). $(26.24)_{16} \rightarrow (\ )_{10}$

$6\times16^0+2\times16^1 . 2\times16^{-1}+4\times16^{-2}$

$6+32 . 0.125+0.015625$

$(38.140625)_{10}$.

3).

1). $(33)_{10} \rightarrow (\ )_2$.

2	33	1
2	16	0
2	8	0
2	4	0
2	2	0
	1	

$(100001)_2$.

My Write is My Future

$(100001)_2 \rightarrow (\ )_8 \Rightarrow (41)_8.$

$(41)_8 \rightarrow (\ )_{16}$

$\Rightarrow (101001)_2$

$\Rightarrow (29)_{16}.$

2	41	1
2	20	0
2	10	0
2	5	1
2	2	0
	1	

2). $(1110101)_2 \rightarrow (\ )_{10}$

$1\times 2^0 + 0 + 1\times 2^2 + 0 + 1\times 2^4 + 1\times 2^5 + 1\times 2^6$

$1 + 4 + 16 + 32 + 64$

$(117)_{10} \Rightarrow (165)_8 \Rightarrow (?)_{16}.$

$(165)_8 \rightarrow (\ )_2 \Rightarrow (001110101)_2. \Rightarrow (75)_{16}.$

Date / /
Page

⇒ Complement of a Number:-

$x = (10)_{10}$

9's complement ⇒ 99 − 10
⇒ 89

10's complement ⇒ 9's complement + 1
⇒ 89 + 1
⇒ 90.

If $x = (1011)_2$.

1's complement ⇒ 1111
1011
0100

2's complement :- 0100
+ 1
0101.

⇒ $x = 12389$

9's complement ⇒ 99999
− 12389
87610

10's complement ⇒ 87611

⇒ $x = 1011001$

1's complement ⇒ 0100110

2's Comp = 0100111

Date / / Page

⇒ Rules for addition:-

0 + 0 = 0

0 + 1 = 1

1 + 1 = 0, carry = 1

1 + 1 + 1 = 1, carry = 1

(i)
```
  11011101
+ 11011011
 110111000
```

Subtract:-

```
  11011101
- 11011011
  00000010
```

(b)
```
      1101
     x0010
      0000
     1101x
    0000xx
    011010
```

(a). 10101.001

1's ⇒ 01010.110 (alag alag karna hai)

2's ⇒ 01011.111

carry will be discarded

```
  111
   +1
 1000
```
discard.

(b) $(1010.110)_{10}$

9's complement ⇒
```
  9999
- 1010
(8989)
    +1
⇒ 8990
```

```
  999
- 110
  889
   +1
  890
```

$(8990.890)_{10}$

(b) 2's complement $(1010.110)_{10}$.

$(1010)_{10} \rightarrow (\ \ )_2$

2	1010	0
2	505	1
2	252	0
2	126	0
2	63	1
2	31	1
2	15	1
2	7	1
2	3	1
	1	

(1111110010)

2	~~110~~	0
2	~~55~~	~~1~~
2	~~27~~	1
2	~~13~~	1
2	~~6~~	~~0~~
2	~~3~~	0
	~~1~~	

~~.110 × 2 =~~

~~.220~~

multiply.

$.110 \Rightarrow (0001)_2$.

~~(1001110)~~.

(1111110010)

1's ⇒ 0000001101

2's ⇒ +1

(0000001110)

~~1's ⇒ 0110001~~

2's = ~~+1~~

~~0110010~~

~~(0000001110.0110010). A~~

$(1110.1111)_2$ Ans.

$.110 \times 2 = 0$

$.220 \times 2 = 0$

$.440 \times 2 = 0$

$.880 \times 2 = 1$

1.660

(0001)

⇒ Integer Reperesntation

10 → Unsigned reporesentation (By default positive)
+10, −(10) → signed representation.
magnitude.

for Binary number:-
0 → represents +ve number
1 → represents −ve number.

ex:- +10   −10 1^{st} Method.

$(01010)_2$ $(11010)_2$.
sign   sign

2^{nd} method for representing a negative binary number:-
→ 1's complement.

−10 ⇒ ~~1110~~   +10 ⇒ 01010
1's complu ⇒ 10101.

→ 3^{rd} method (2's complement).

−10 ⇒ +10 ⇒ 01010
1's ⇒ 10101
2's ⇒ +1
10110.

(1) +32 ⇒

2	32	0
2	16	0
2	8	0
2	4	0
2	2	0

$(100000)_2$.

(0100000). Ans.

$\Rightarrow$ $-32 \Rightarrow +32 \Rightarrow 0100000$

(1100000) $1's \Rightarrow (1011111)$

$+1$

$2's \Rightarrow (1100000)$

$\rightarrow$ Subtraction of two n-digit no. using r's complement Method :-

Ex :- $M - N$

$M \rightarrow$ Minued

$N \rightarrow$ Subtrahand.

$M \rightarrow 72532$

$N \rightarrow 13250$

$M - N = 59282.$

$\rightarrow$ $72532 + (10's$ complement of subtrahand$)$

13250

$$\begin{array}{r} 99999 \\ -\ 13250 \\ \hline 86749 \\ +1 \\ \hline 86750 \end{array} \Rightarrow \begin{array}{r} 72532 \\ +\ 86750 \\ \hline (1)59282 \end{array}$$

Discard.

$\rightarrow$ ~~N − M~~

$$\begin{array}{r} 13250 \\ -\ 72532. \\ \hline -59282 \end{array}$$

$$\begin{array}{r} 13250 \\ +\ 86750 \\ \hline 99900 \end{array}$$

$$\begin{array}{r} 99999 \\ -\ 99900 \\ \hline 00099 \\ +1 \\ \hline 00 \end{array}$$

My Write is My Future

~~13250 + (10's complement of 72532).~~

$$\begin{array}{r} 99999 \\ -\ 72532 \\ \hline 27467 \\ +\ 1 \\ \hline 27468 \end{array} \qquad \begin{array}{r} 13250 \\ +\ 27468 \\ \hline 40718 \end{array}$$

→ N − M

$$\begin{array}{r} 13250 \\ -\ 72532 \\ \hline -59282. \end{array}$$

13250 + (10's complement of 72532).

$$\begin{array}{r} 99999 \\ -\ 72532 \\ \hline 27467 \\ +\ 1 \\ \hline 27468 \end{array} \qquad \begin{array}{r} 13250 \\ +\ 27468 \\ \hline 40718 \end{array}$$

$$\begin{array}{r} 99999 \\ -\ 40718 \\ \hline 59281 \\ +\ 1 \\ \hline -59282 \end{array}$$

Because there is no carry generated so for finding the final value we have to find 10's complement.

→ For Binary Number System

X = 10101000

Y = 1000011

X − Y = X + (2's complement of Y)

```
  1111111
− 1000011
  0111100
       +1
  0111101
```

```
   10101000
+  00111101
 0)00000101
discard.
```

⇒ Y − X

Y − X = Y + (2's complement of X).

```
  1111111
− 1010100
  0101011
       +1
  0101100
```

```
   1000011
+  0101100
   1101111
```

No-carry.

Result is negative

Again 2's complement

```
  1111111
− 1101111
  0010000
       +1
  0010001
```

final Ans

$-20 + (-10)$
-20 -10

Subtrahend will be selected by seeing − sign before a number.
i.e. −20 + 10
Subtrahend.

Date
Page

→ 20 − 10 → Subtraction by using 2's complement.

2	20	0
2	10	0
2	5	1
2	2	0
	1	

$(10100)_2$.

2	10	0
2	5	1
2	2	0
	1	

$(01010)_2$.

(10100) − (01010).

20 + (2's complement of 01010)

```
  10101
   +  1
  -----
  10110
```

```
   10100
 + 10110
 -------
  101010
```

01010 Ans. ⇒ 10.

MyWrite is My Future

(i) $(+6) + (+13)$

$$\begin{array}{r} 6 \\ +13 \\ \hline 19 \end{array} \Rightarrow 19.$$

(ii) $(-6) + (+13)$

13 + (10's complement of 6).

$$\begin{array}{r} 9 \\ -6 \\ \hline 3 \\ +1 \\ \hline 4 \end{array} \qquad \begin{array}{r} 13 \\ -\cancel{0}4 \\ \hline \cancel{0}7 \end{array}$$

(iii) $(+6) + (-13)$

6 + (10's complement of 13).

$$\begin{array}{r} 99 \\ -13 \\ \hline 86 \\ +1 \\ \hline 87 \end{array} \qquad \cancel{\begin{array}{r} 13 \\ +87 \\ \hline 100 \end{array}} \qquad \begin{array}{r} 87 \\ +6 \\ \hline 93 \end{array}$$

Discard.

$$\begin{array}{r} 99 \\ -93 \\ \hline -7 \end{array}$$

(iv) $(-6) + (-13)$

(v) $(-6) - (-13)$

$(-6) + (+13)$.

13 + (10's complement of 6).

$$\begin{array}{r} 9 \\ -6 \\ \hline 3 \\ +1 \\ \hline 4 \\ \hline \end{array} \qquad \begin{array}{r} 13 \\ -4 \\ \hline 7 \\ \hline \end{array}$$

x —— x —— x

Using Binary :- Arithemetic Addition and subtraction always using 2's complement representation.

(i) $(-6) + (+13)$

$-6 \Rightarrow +6 \Rightarrow 00110$

$+13 \Rightarrow 01101$

sign.

$$\begin{array}{rr} & 00110 \\ \text{1's com -} & 11001 \\ & +1 \\ \hline \text{2's complem} & 11010 \\ \hline \end{array} \Rightarrow -6$$

$$\begin{array}{rr} +13 \rightarrow & 01101 \\ & 11010 \\ \hline & 1)00111 \end{array}$$

Discard $\Rightarrow +7$.

$$\begin{array}{rr} & \cancel{01101} \\ \cancel{\text{1's comp}} & \cancel{10010} \\ & \cancel{+1} \\ \hline \cancel{\text{2's complem}} & \cancel{10011} \end{array}$$

Date
Page

(ii) $+6 + (-13)$

$+6 \rightarrow 00110$

$+13 \rightarrow 01101$

1's comp $\rightarrow 10010$

$+1$

$10011 \Rightarrow -13.$

$$\begin{array}{r} 00110 \\ +\ 10011 \\ \hline (1)1001 \end{array}$$

$\rightarrow$ Again 2's comp -

Sign will be decided here.

$$\begin{array}{r} 00110 \\ +1 \\ \hline 00111 \end{array} = -7$$

(iii) $(\cancel{+}6) - (-\cancel{+}13)$

$+6 \rightarrow 00110 \Rightarrow -6 \Rightarrow 11010$

$+13 \rightarrow 01101$

1's com 10010

$+1$

$10011 \Rightarrow -13.$

~~6 + (2's complement of 13).~~

16 8 4 2 1

$$\begin{array}{r} \cancel{11111} \\ -\ \cancel{10011} \\ \hline \cancel{01100} \end{array}$$

$$\begin{array}{r} \cancel{00110} \\ +\ \cancel{01100} \\ \hline \cancel{10010} \end{array}$$

8 4 2 1

$(11010) - (10011).$

$\hookrightarrow$ 2's complement

Date / /
Page

	10011		1
1's	01100		11010
	+1	+	01101
	01101		1) 00111 + (7)

Discard.

→ Overflow

Unsigned Number.

x = 10010101
y = 10101010
1) 00111111

End carry → Overflow is happening.

Signed Number

carry from sign bit → carry on sign bit

(1) x = 01010101
01101010
10111111

Overflow is happening. because carry on sign bit and carry from sign bit are different.

→ No overflow because carry on sign bit and carry from sign bit are same.

(ii) x = 00101011
01000011
01101110

Date / /
Page

→ fixed & floating point Representation of a number:-

• 325.08
↳ $M \times r^e$ — exponent, ↳ Base → scientific representation of a floating point number.
↓
mantissa
0.32508×10^3.

• $(11011.001)_2$.
$\Rightarrow 0.11011001 \times 2^{5}$
$5 = 101$

→ Normalization.
$001001.001 \Rightarrow 0.001001001 \times 2^6$.
$\Rightarrow 000.1001001 \times 2^4$
0100 1001, 0100

(I) 0.0000101101
$\Rightarrow 0.101101 \times 2^{-4}$

(II) 01101110.101
$\Rightarrow 0.1101110101 \times 2^7$

(III) 101.001×2^{-2}
$\Rightarrow 0.101001 \times 2$.

Date ___ Page ___

(1) Represent 1101011 in the floating point representation using 32-bit word length (including mantissa and exponent). Assume that 24 bit represent mantissa and 8 bits represents exponent.

1101011.0 $\quad$ making 8 bit

$\Rightarrow 0.1101011 \times 2^7 \longrightarrow 0111 \longrightarrow 00000111$

Mantissa $\rightarrow$ ← 24 bit → ← 8 bit → Exponent

| 0 | 1101011 0000 - - - 000 | 00000111 |

sign $\qquad$ 32-bit

(ii) $-105 \Rightarrow -1101001$

← 24-bit → ← 8-bit →

| 1 | 1101001 000 - - - 00 | 00000111 |

sign

2	105	1
2	22	0
2	11	1
2	5	1
2	2	0
	1	

IEEE Standard for floating point number Representation:-

(1) 32-Bit representation (Single precision)

(2) 64-bit representation (Double precision)

← 8-bit → ← mantissa →

| | E' | 23 bit |

Sign of mantissa ← 1bit → ← 32-bit →

Biased exponent

$E' = E + 127$ (Biased value)

$(2^{e-1} - 1) \Rightarrow (2^{8-1} - 1)$

$\Rightarrow 128 - 1$

$\Rightarrow 127.$

My Write is My Future

→ 64-bit Representation.

sign of mantissa ←	1-bit	← 11-bit → E'	52 bit ← mantissa →

$E' = E + 1023$

$\downarrow$

$(2^{e-1} - 1)$

$\Rightarrow (2^{11-1} - 1)$

$= 2^{10} - 1$

$\Rightarrow \underline{1023}$.

(a) Represent -0.875 decimal in IEEE 754 floating point representation.

$\Rightarrow (-0.875)_{10}$

$\Rightarrow (-0.111)_2$

$0.875 \times 2 = 1$
$0.750 \times 2 = 1$
$0.500 \times 2 = 1$
$.000$

$\boxed{\text{Value represented as} = \pm 1 \cdot M \times r^e}$

$\Rightarrow -1.11 \times 2^{-1}$

$E = -1$ exponent value is always +ve.

$E' = E + 127$

$= -1 + 127$

$= \underline{126}. \rightarrow 01111110$

64 40

32 16 8 4 2 1

1	01111110	110000 - - - 000

← 8-bit →← 23-bit →

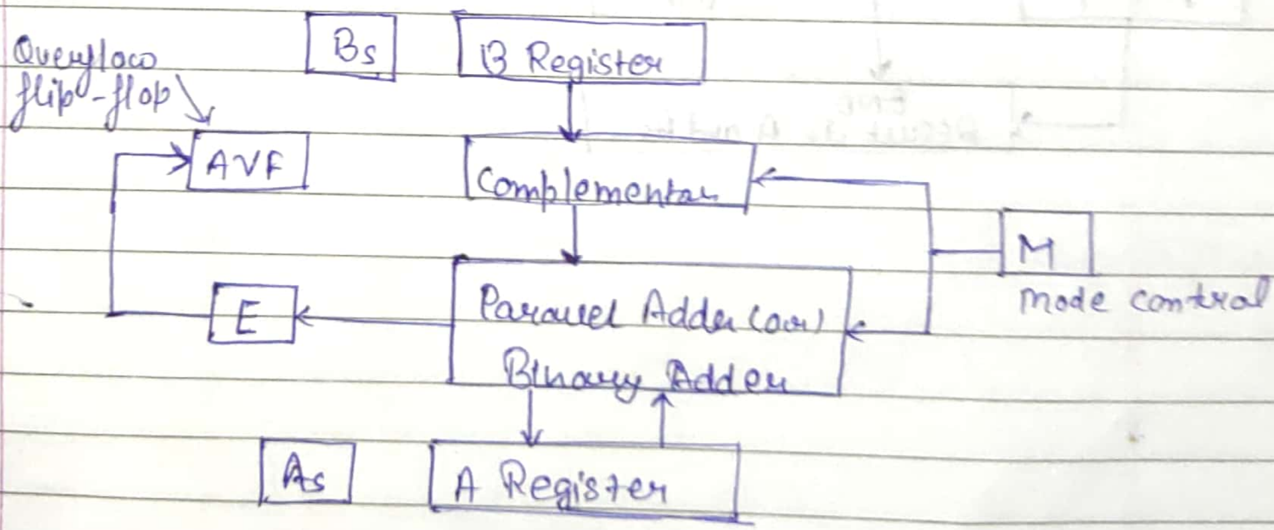
E'

→ Addition & Subtraction

Addition

Operation	Add magnitude	Subtraction magnitude A > B	A < B	A = B
(+A) + (+B)	+(A + B)			
(+A) + (−B)		+(A − B)	−(A − B)	+(A − B) = 0
(−A) + (+B)		−(A − B)	+(A − B)	+(A − B)
(−A) + (−B)	−(A + B)			
(+A) − (+B)		+(A − B)	−(A − B)	+(A − B)
(+A) − (−B)	+(A + B)			
(−A) − (+B)	+(A + B)	−(A − B)	+(A − B)	+(A − B)
(−A) − (−B)				

Hardware Required for Signed-Magnitude Addition & Subtraction :-

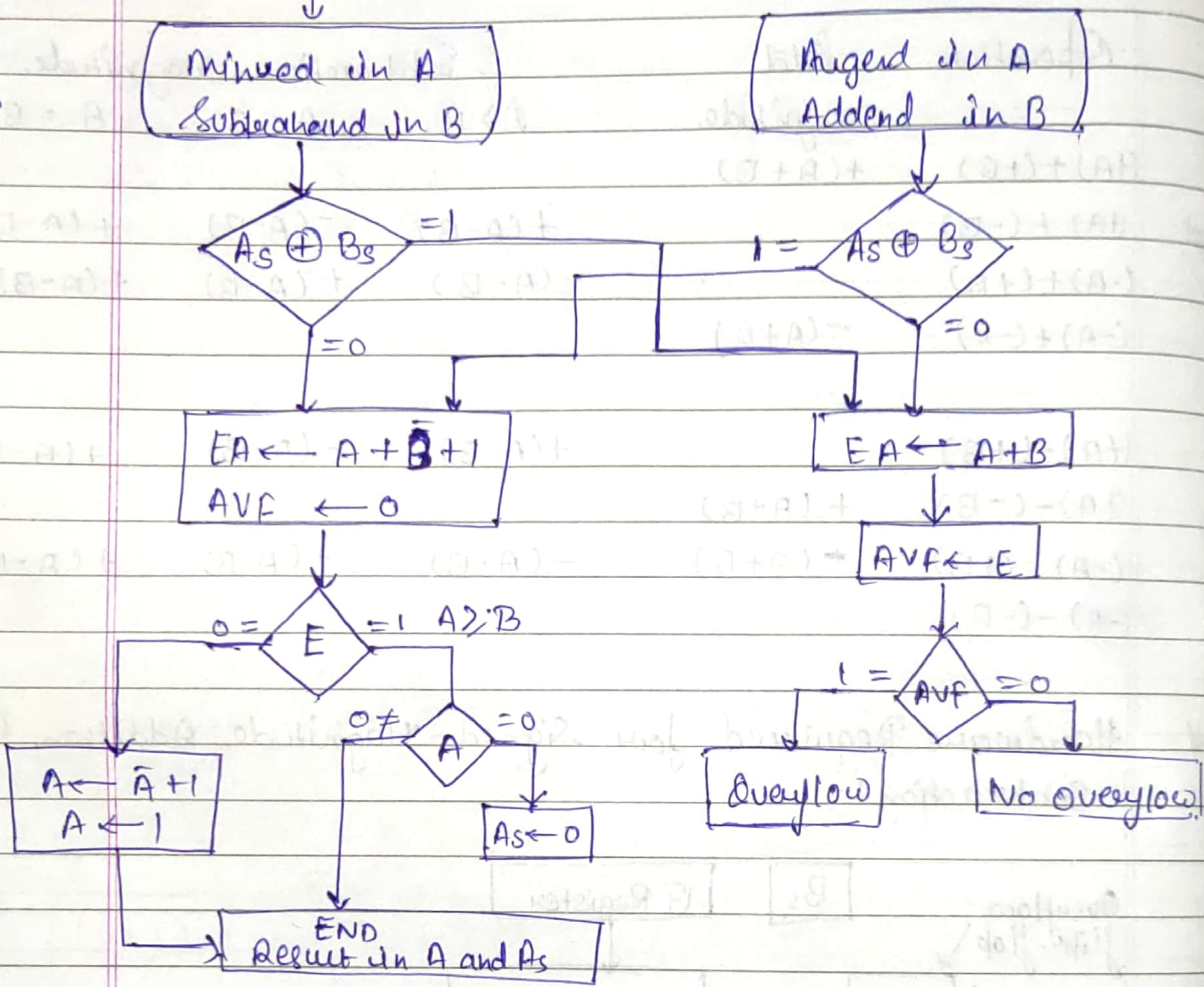


Date / /
Page

Subtraction Operation

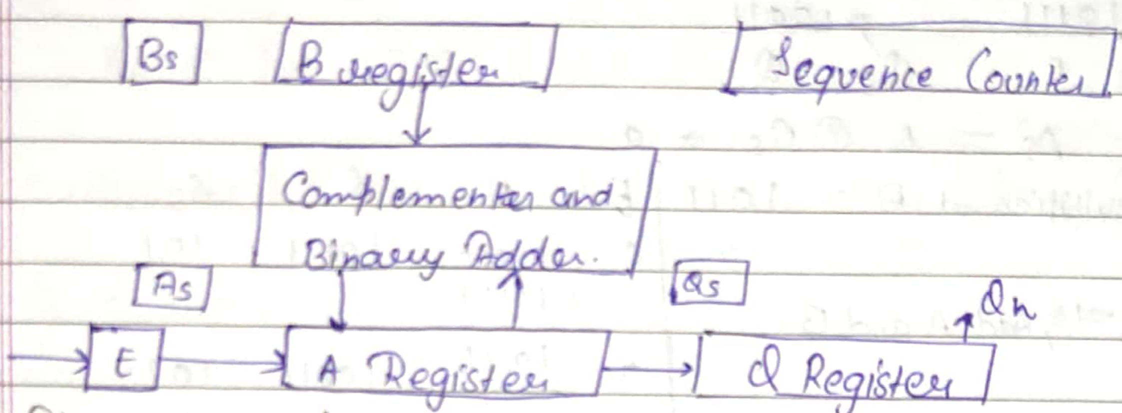
Addition Operation

A_s and B_s - Sign Bits

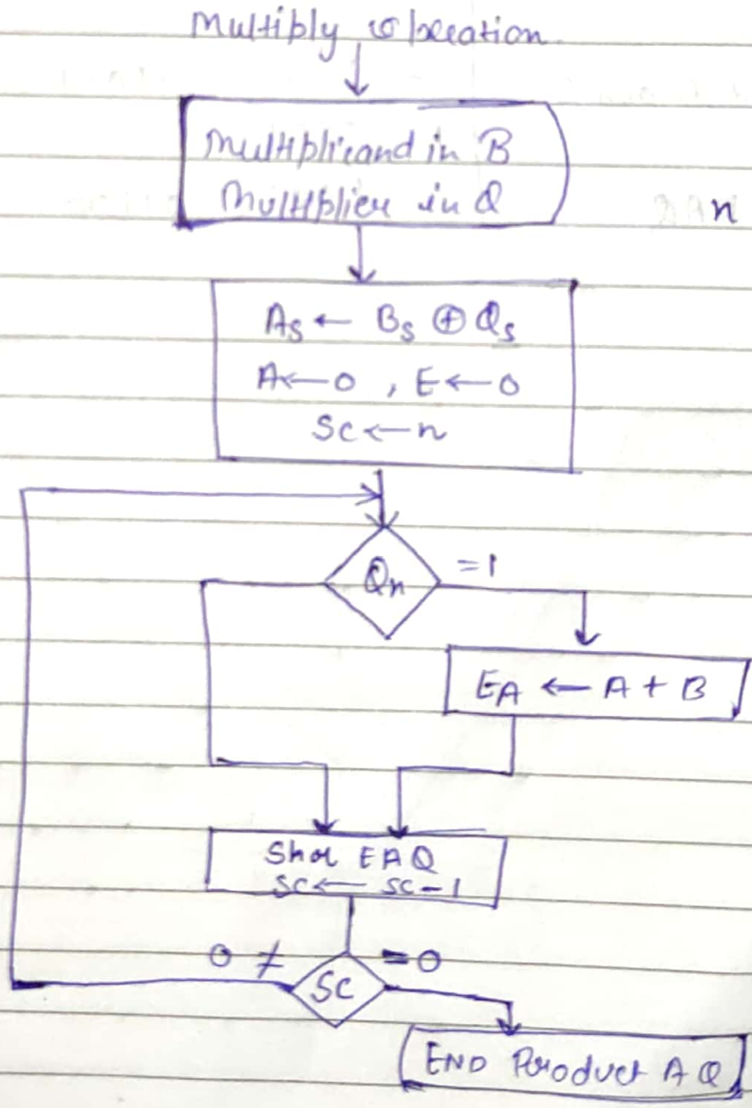


Sign
+ + = 0
− − = 0
+ − = 1
− + = 1

→ Multiplication Algorithm:-
Hardware required.



• Flow chart for Multiplication Operation



n → no. of bit in multiplier.

Shr - Shift right

64 32 16 8 4 2

→ Multiply :-

(1) +23 and +19

0 10111   0 10011

B_s B Q_s Q

$A_s \rightarrow B_s \oplus Q_s \rightarrow 0$

Multiplicand B = 10111	E	A	Q	SC
	0	00000	10011	101
$Q_n = 1$, Add A and B		+10111		
	0	10111	10011	101
Shift Right EAQ	0	01011	11001	100
$Q_n = 1$, Add A and B		+ 10111	11001	
	1	00010		
Shift Right RAQ		10001	01100	011

My Write is My Future

(ii) (+3) and (+4)

$\underset{B_s}{0}\ \underset{B}{011}$ $\underset{Q_s}{0}\ \underset{Q}{100}$

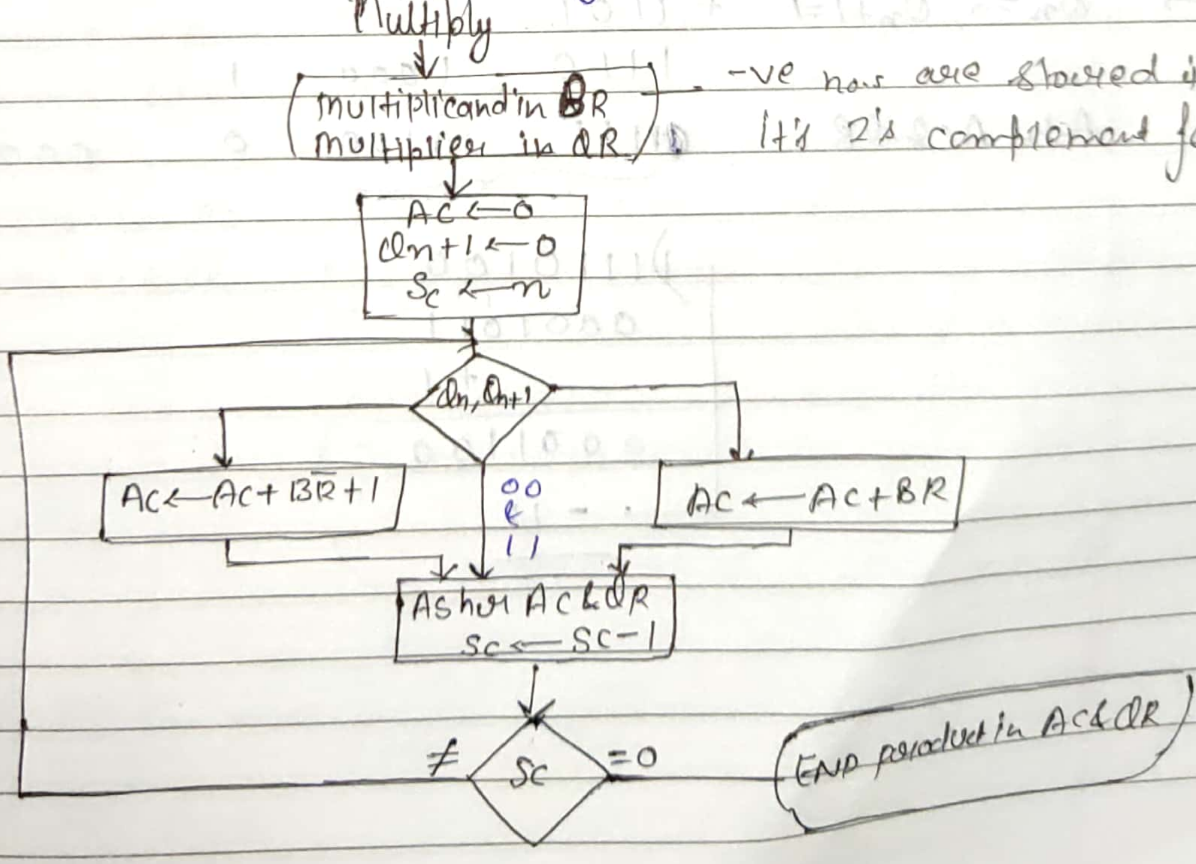
$B_s \oplus Q_s = 0\ (+ve)$

Multiplicand B = 011

	E	A	Q	Sc (3-binary)
	0	000	100	011
$Q_n=0$, shift right EAQ	0	000	010	010
$Q_n=0$, shift right "	0	000	001	001
$Q_n=1$, Add A and B		+011	001	001
	0	011		
Shift right EAQ	0	001	100	000

$\underset{sign}{0}\ \underbrace{001100}_{12} \Rightarrow +12$

Flow Chart For Booth Algorithm :-



-ve nos are stored in it's 2's complement form.

Date / /
Page

(1) $(-3) \times (+4)$

$+4 \rightarrow 0100 \rightarrow QR$

$+3 \rightarrow 0011$

$-3 \rightarrow 1101 \rightarrow BR$

$BR = 1101$

$\overline{BR}+1 = 0011$

	AC	QR	Q_{n+1}	SC
Initialization	0000	0100	0	100
→ $Q_n = 0, Q_{n+1} = 0$ Ashr AC & QR	0000	0010	0	011
→ $Q_n = 0, Q_{n+1} = 0$ Ashr AC & QR	0000	0001	0	010
→ $Q_n = 1, Q_{n+1} = 0$	+ 0011	0001	0	
$\overline{BR}+1$ ⇒ subtract BR in AC	0011			
Ashr AC & QR	0001	1000	1	001
→ $Q_n = 0, Q_{n+1} = 1$	+ 1101			
	1110	1000	1	
Ashr AC & QR	1111	0100	0	000

11110100

00001011

+1

00001100

→ −12

Date __/__/__
Page ____

(17) (+17) x (−6) → Booth Algorithm.

010001 — multiplicand → BR

+6 ⇒ 0110
1's 1001
+1
−6 ⇒ 1010 → Multiplier → QR

AC = 0 Q_{n+1} = 0   SC ⇒ 100.

$\overline{BR}+1$ = 101111

BR = 010001	AC	QR	Q_{n+1}	SC
Initialization	000000	1010	0	100
$Q_n = Q_{n+1} = 0$				
Ashr AC & QR	000000	0101	0	011
$Q_n = 1$ & $Q_{n+1} = 0$	+101111			
Add $\overline{BR}+1$ in AC	101111	0101	0	
Ashr AC & QR	110111	1010	1	010
$Q_n = 0$, $Q_{n+1} = 1$	+010001			
Add BR in AC	001000	1010	1	
Ashr AC & QR	000100	0101	0	001
$Q_n = 1$ & $Q_{n+1} = 0$	+101111			
Add $\overline{BR}+1$ in AC	110011	0101	0.	
Ashr AC & QR	111001	1010	1	000.

111001 1010
1's → 000110 0101
+1
100110 0110 ⇒ −102 Ans.

48
32
80

Qu (−10) x (+20)

Registers

Group of flip-flops is known as Register.

→ Two types :-

1). General Purpose Registers.
2). Special Purpose Registers.

→ Types of Special Purpose Registers.

- MDR. (Memory Data Register)
- MAR (Memory Address ")
- AC (Accumulator)
- IR (Instruction Register)
- PC (Program Counter)
- TR (Temporary Register)
- INPR (Input Register)
- OUTR (Output ")

Register Transfer Language.

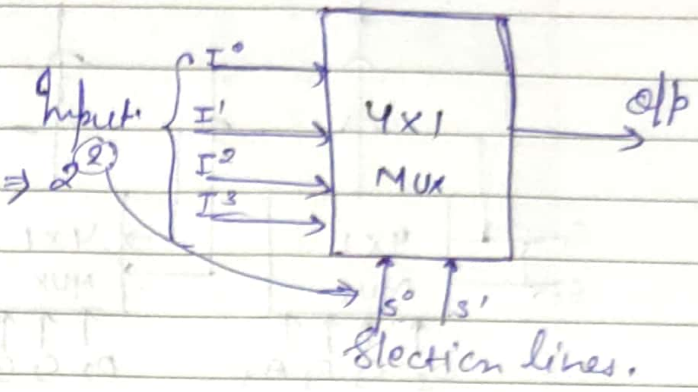
R_1	R_2	R_3
7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0

1). $R_1 \leftarrow R_2$ (copy data of R_2 in R_1)
2). $R_3 \leftarrow R_1 + R_2$ (copy data of $R_1 + R_2$ in R_3)
3). If (P is true)
Condition
$P: R_1 \leftarrow R_2$.

⇒ Multiplexer

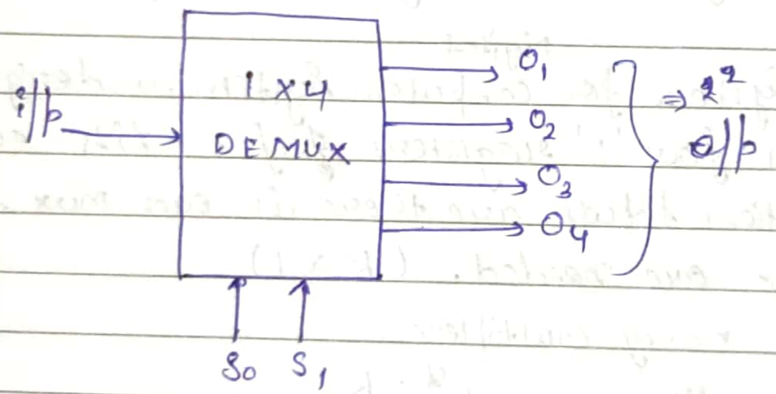
Multiple input and Single Output.

S_0	S_1	
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



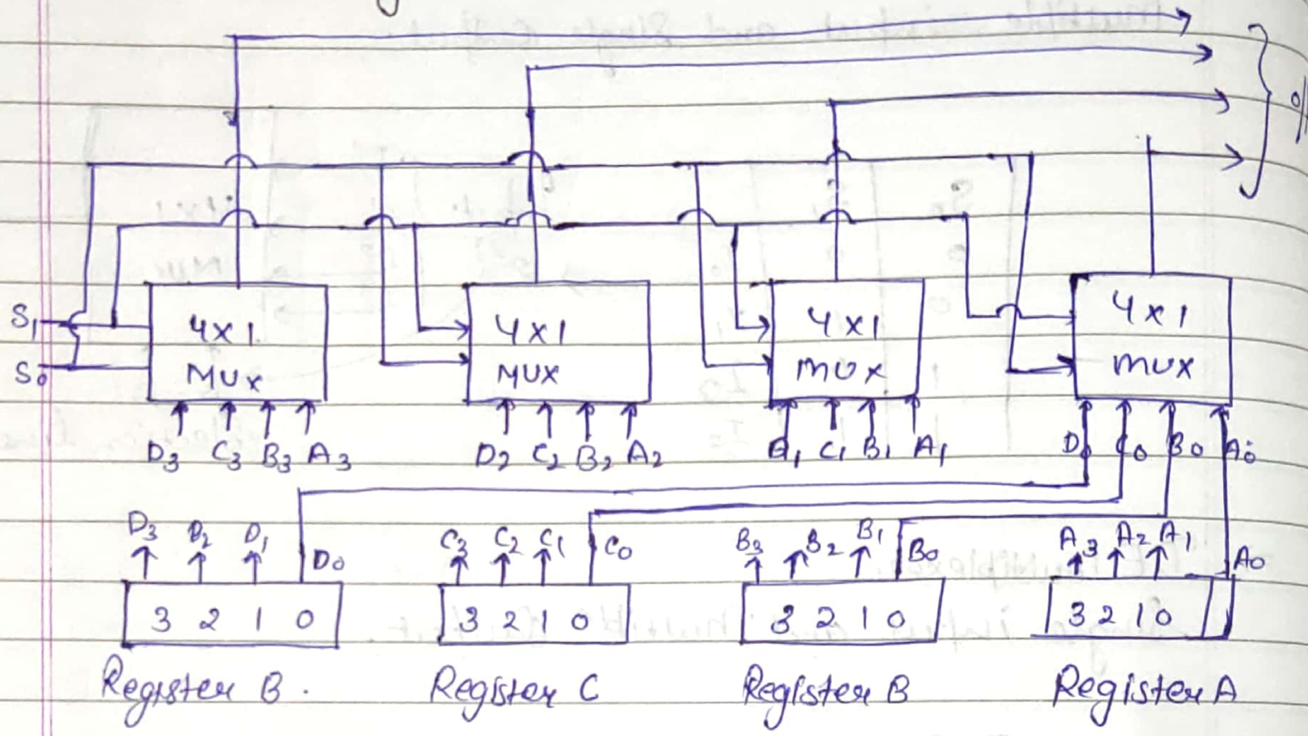
⇒ DE Multiplexer.

Single input and Multiple Output.



Date / /
Page

⇒ Common Bus System.



Qus. A common Bus System for Digital computer System is designed using multiplexers four 'k' registers of n-bits each.

(i) How many selection inputs are there in each mux.

(ii) What size of mux are needed. ($k \times 1$)

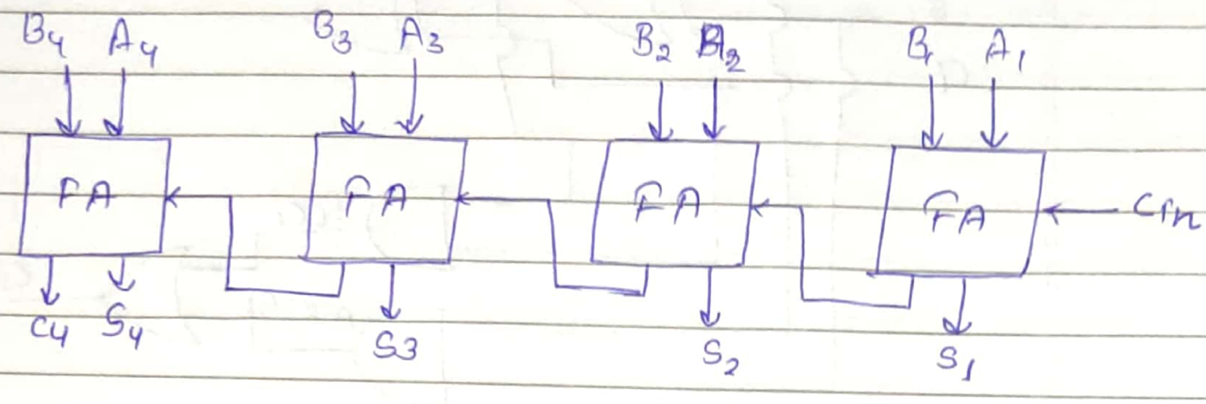
no. of bits ⇒ no. of multiplexer.

Selection lines → $2^x = k$

$x = \log_2 k$

Carry Look Ahead Adder.

4-bit Binary Adder.



The general strategy of designing fast adder or carry look ahead adder is to reduce the time required to form carry signal.

A	B	Ci	Sum	Carry (C_{i+1})	
0	0	0	0	0	No carry
0	0	1	1	0	
0	1	0	1	0	carry propagate C_i = Cout
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	Carry Generate.
1	1	0	0	1	
1	1	1	1	1	

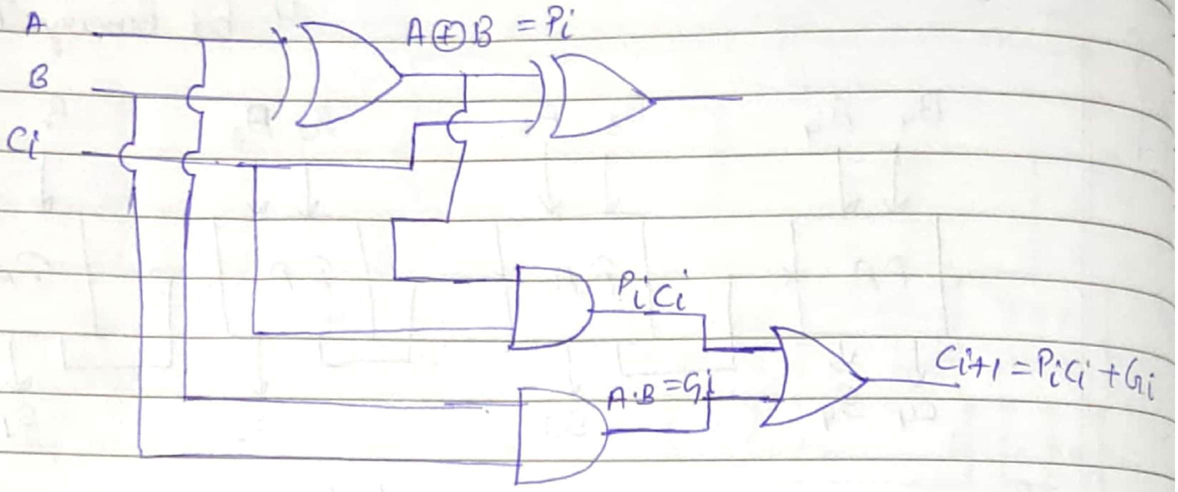
Carry Generate

$$G_i = AB\bar{C} + ABC$$
$$= AB(\bar{C}+C)$$
$$= A_i B_i$$

Carry propagate

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

Carry Look Ahead Adder.



$$\boxed{C_{i+1} = P_i c_i + G_i}$$

$i = 0$

$$C_1 = P_0 C_0 + G_0$$
$$(A_0 \oplus B_0) \cdot C_0 + A_0 B_0$$

$i = 1$

$$C_2 = P_1 C_1 + G_1$$
$$= (A_1 \oplus B_1) \cdot C_1 + A_1 \cdot B_1$$
$$= A_1 \oplus B_1 \, [(A_0 \oplus B_0) \cdot C_0 + A_0 B_0] + A_1 B_1$$