Unit-III

Combinational logic Circuits

Half Adder:

→ Half adder

→ full adder

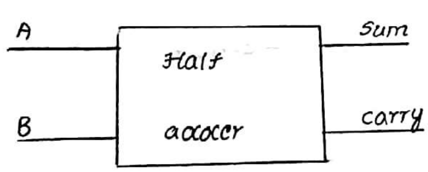
→ Ripple adder / parallel adder

→ carry look ahead adder

Half adder

used to add 2 bits

Block diagram:



Truth table

Inputs		Outpus	
A	B	Sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

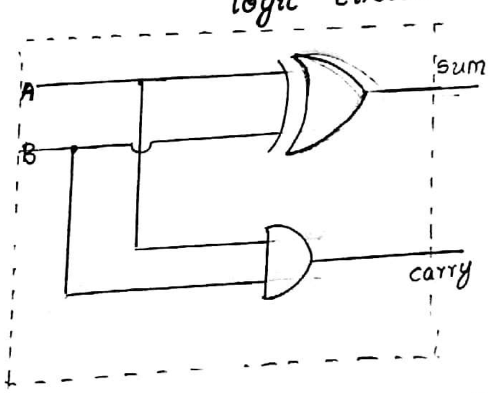
logic expressions:

Sum = $\bar{A}B + A\bar{B}$

$= A \oplus B$

Carry = $A \cdot B$

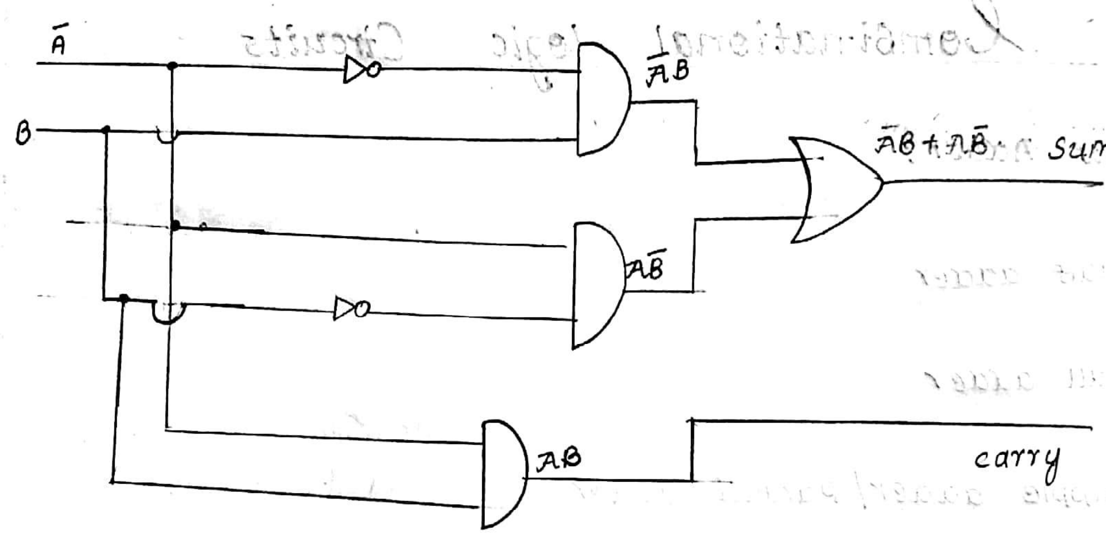
logic circuit:



Half Adder using Basic gates

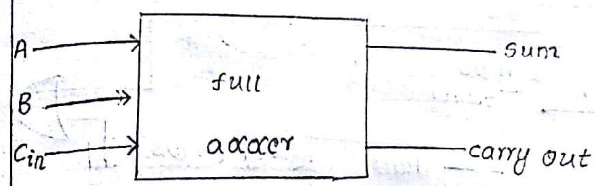
$Sum = \bar{A}B + A\bar{B}$

$carry = AB$



Half adder using ~~N gate~~ Nand gate

Full adder: used to add 2 input bits along with carry input



Truth Table

inputs			outputs	
A	B	C_{in}	Sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$sum = \Sigma m(1,2,4,7)$

$carry = \Sigma m(3,5,6,7)$

Sum :-

$= \bar A \bar B C_{in} + \bar A B \bar C_{in} + A \bar B \bar C_{in} + ABC_{in}$

$= [C_{in}(\bar A \bar B + AB)] + [\bar C_{in}(\bar A B + A\bar B)]$

$= [C_{in}(\overline{A \oplus B})] + [C_{in}(A \oplus B)]$

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

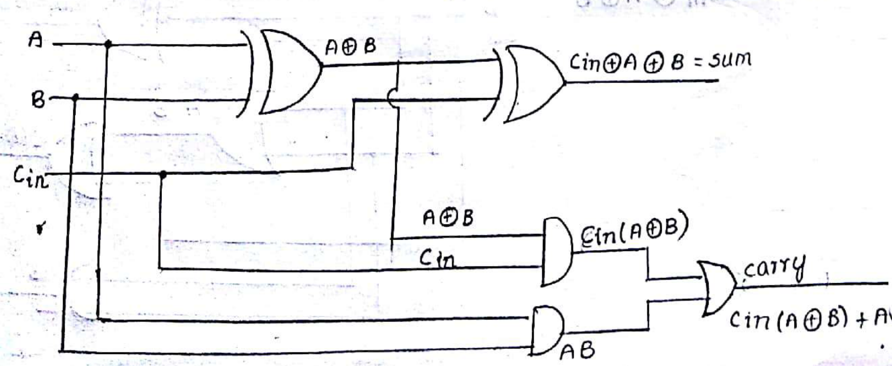
$= C_{in} \oplus A \oplus B$

Carry :- $\bar A B C_{in} + A \bar B C_{in} + AB\bar C_{in} + ABC_{in}$

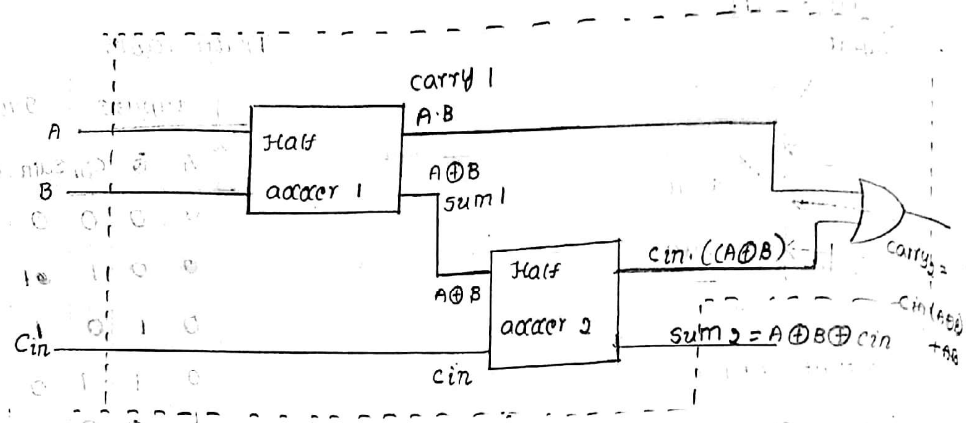
$= C_{in}(\bar A B + A\bar B + AB) + AB\bar C_{in}$

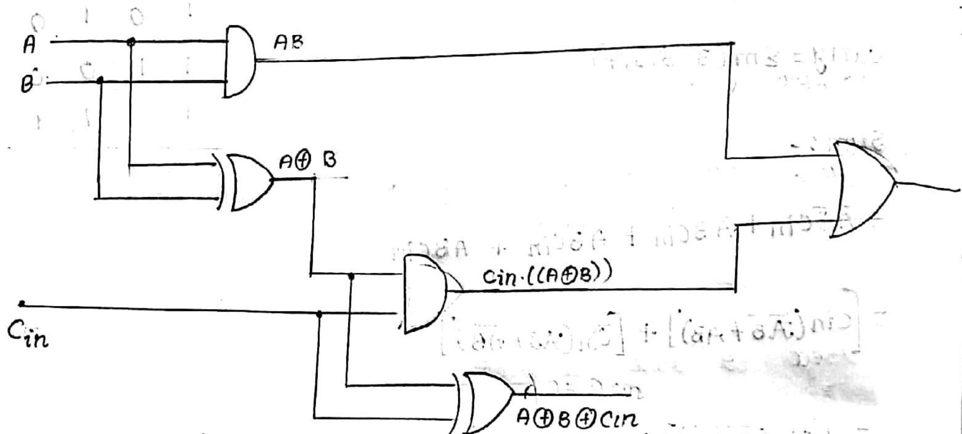
$= C_{in}(\bar A B + A\bar B) + AB(C_{in} + \bar C_{in})$

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



Full adder using two half adders and one OR gate





Sum = $\Sigma m(1,2,4,7)$

Carry = $\Sigma m(3,5,6,7)$

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0 (0)	1 (1)	0 (3)	1 (2)
A	1 (4)	0 (5)	1 (7)	0 (6)

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0 (0)	0 (1)	1 (3)	0 (2)
A	0 (4)	1 (5)	1 (7)	1 (6)

$= \bar{A}\bar{B}C_{in} + \bar{A}B\bar{C}_{in} + A\bar{B}\bar{C}_{in} + ABC_{in}$

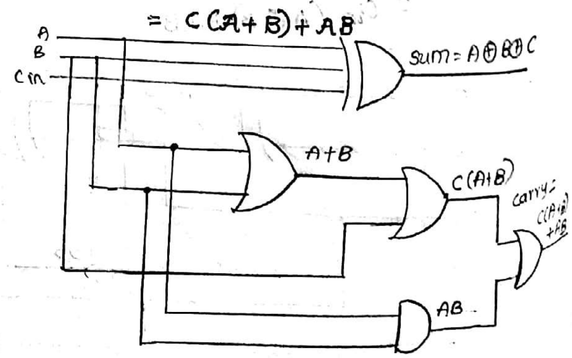
$= C_{in}(\bar{A}\bar{B} + AB) + \bar{C}_{in}(\bar{A}B + A\bar{B})$

$= \cancel{C_{in}(\bar{A}}$ $C_{in} \oplus A \oplus B$

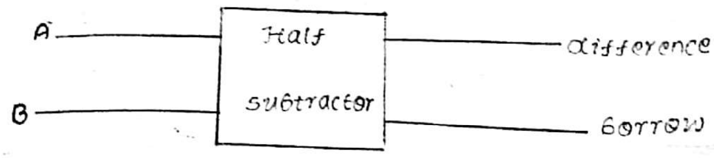
$= (\bar{A} + A)BC + A(\bar{B}C + BC) + A(BC + B\bar{C})$

$= BC + AC + AB$

$= C(A+B) + AB$



Half subtractor : used to subtract 2 bits



Truth Table

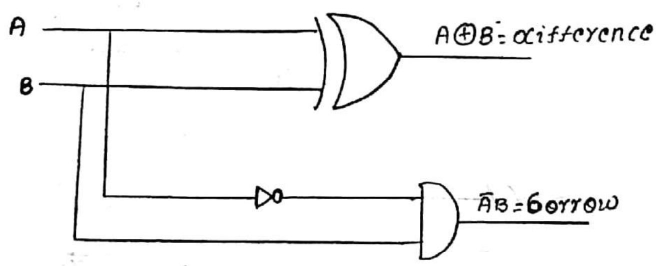
Inputs		Output	
A	B	diff	borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

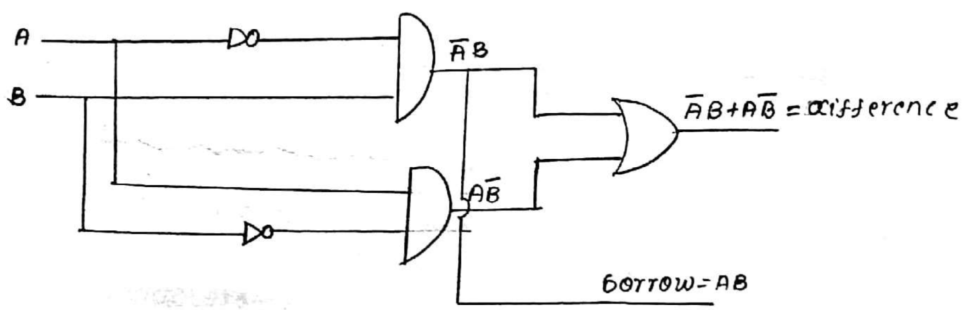
difference $= \bar{A}B + A\bar{B}$

$= A \oplus B$

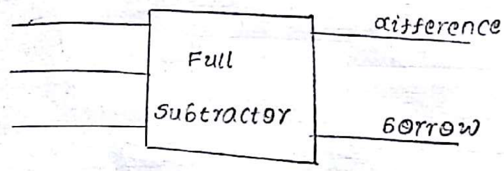
borrow $= \bar{A}B$

Logic circuit:





Full subtractor:



Truth Table

Inputs			Outputs	
A	B	Bin	diff	borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	0 1	1

diff = $\bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + A\bar{B}\bar{B}_{in}$

$+ A B B_{in}$

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

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

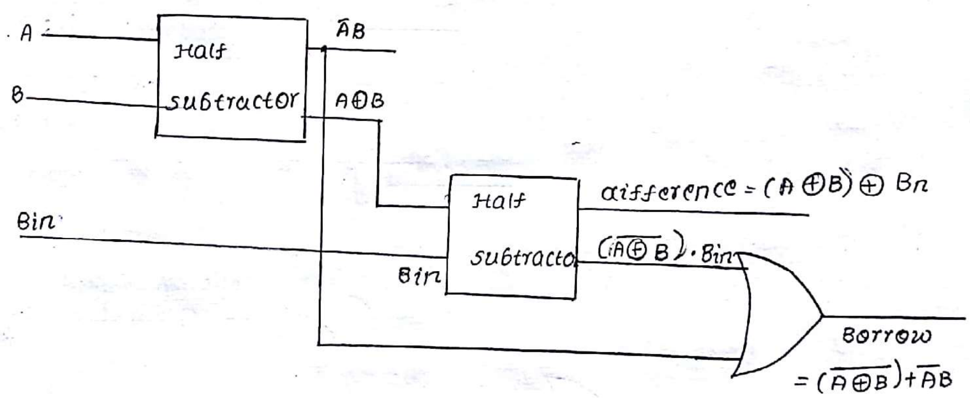
$A\bar{B} + \bar{A}B$

$= B_{in}\oplus A\oplus B$

carry = borrow = $\bar{A}\bar{B}B_{in} + \bar{A}B\bar{B}_{in} + \bar{A}BB_{in} + ABB_{in}$

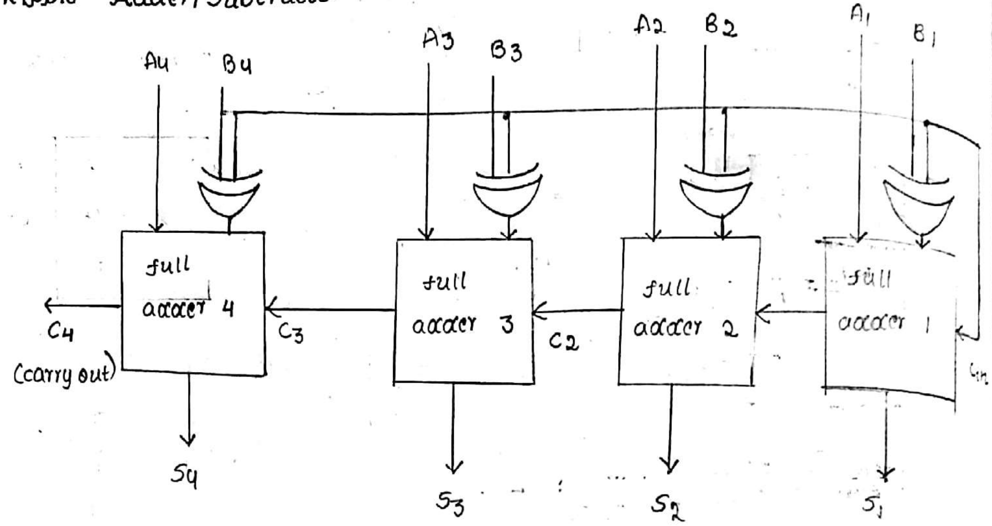
$= B_{in}(\cancel{\bar{A}\bar{B} + \bar{A}B} + AB) + \cancel{\bar{A}B\bar{B}_{in}}$ $\quad B_{in}(\bar{A}\bar{B} + \bar{A}B) + \bar{A}B$

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



Block diagram of full subtractor using 2 half ~~ac~~ subtractors.

Ripple Adder/Subtractor



$B \oplus 0 = B$

$B \oplus 1 = \overline{B}$

$B = 0$ or 1

$C_{in} = 0 \rightarrow$ addition

$= 1 \rightarrow$ subtraction in 2's complement form

XS-3 ~~Addition~~: Adder:

1. To perform XS-3 addition we have to add 2 XS-3 code groups.
2. If carry = 1 add 0011 (3) to the sum of those group
3. If carry = 0 add 1101 (13) to the sum of those groups

Eg 1:- 7+5=12

```
    1010
    1000
00010010
0011+0011
0100 0101
  1    2
  XS-3
```

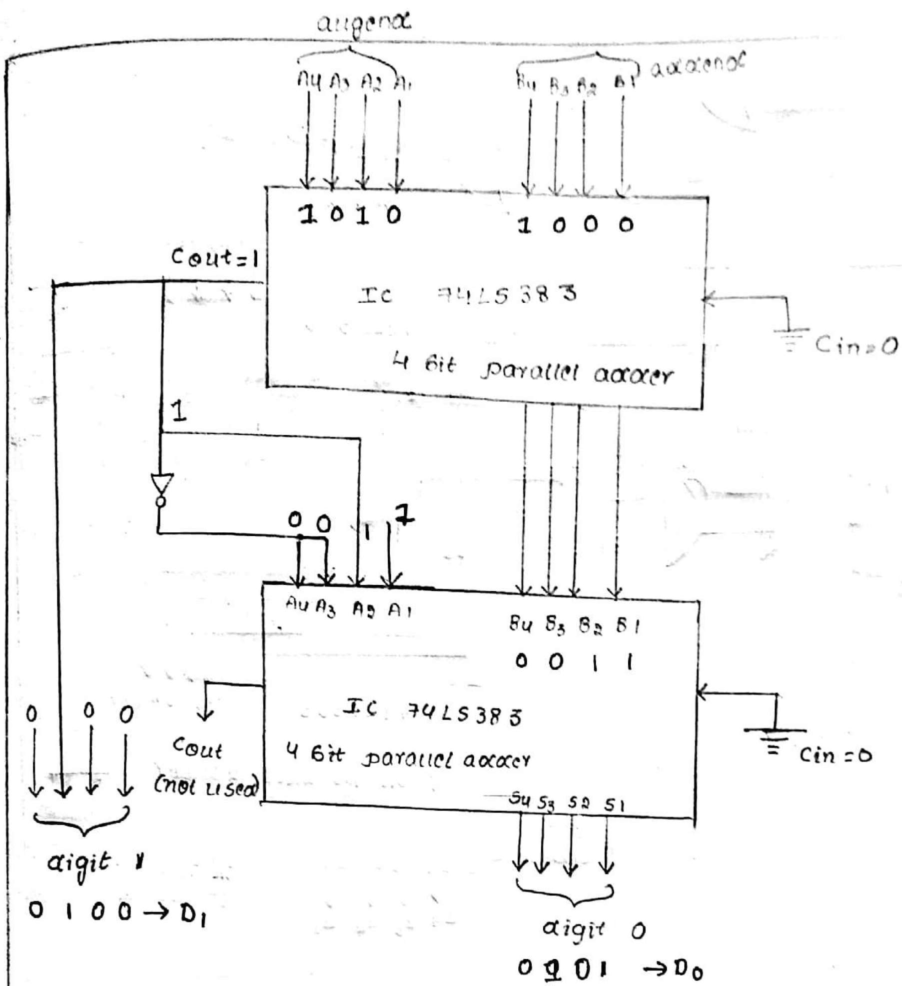
Eg 2:- 4+3=7

$4 \xrightarrow{XS-3} 0111$

$3 \xrightarrow{XS-3} 0110$

```
 1101
 1101 (13 in binary)
11010
```

discard 7 is XS-3



BCD Adder (4 bit);

Eg1:-

$$\begin{array}{rcr} 5 & \rightarrow & 0101 \\ 6 & \rightarrow & 0110 \\ \hline 11 & & 1011 \\ & & +\ 0110 \\ \hline & & \underbrace{0001}_{\text{digit 1}}\ \underbrace{0001}_{\text{digit 0}} \end{array}$$

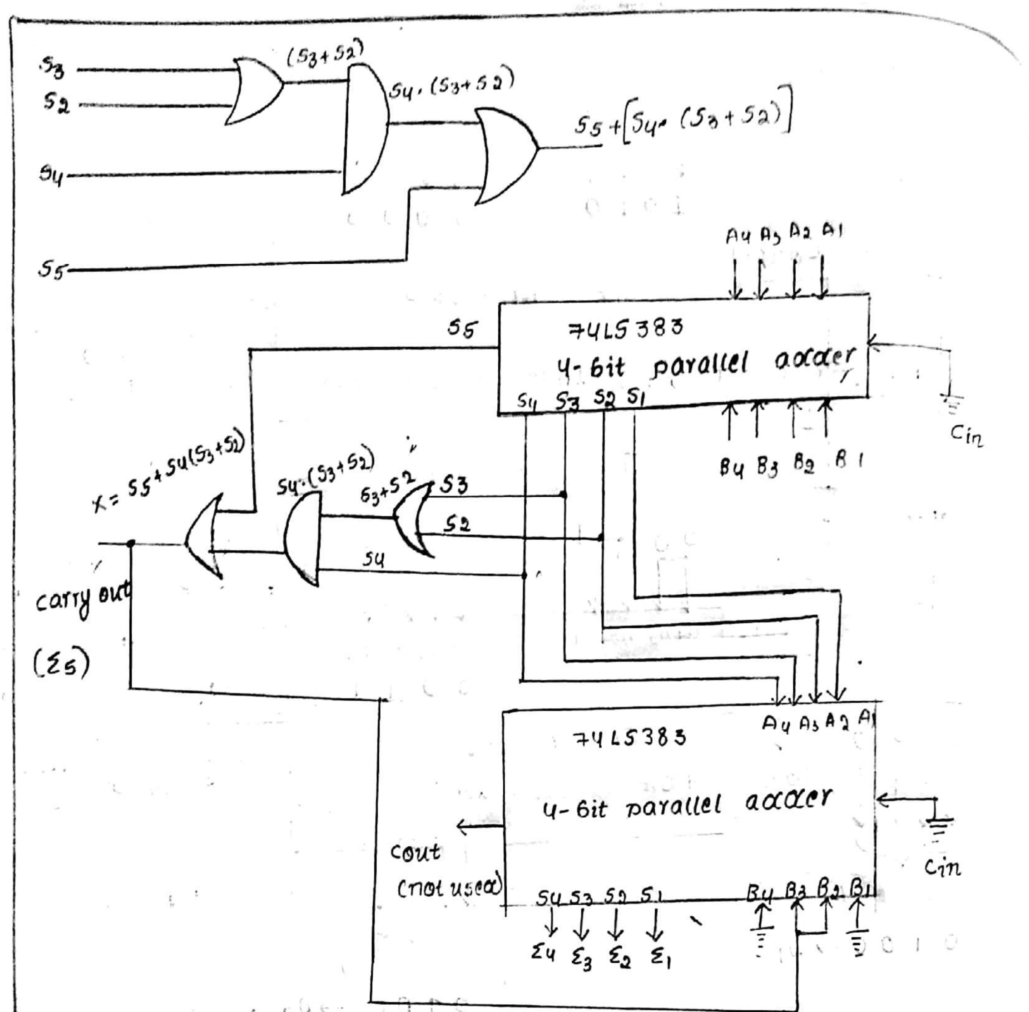
$$\begin{array}{rcl} 4 & \rightarrow & 0100 \\ 3 & \rightarrow & 0011 \\ \hline 7 & & 0111 \rightarrow <9 \\ & & 0000 \\ \hline & & 0111 \rightarrow 7 \text{ in BCD} \end{array}$$

C_3	C_2	C_1	C_{in}
A_4	A_3	A_2	A_1
B_4	B_3	B_2	B_1
$S_5\ S_4$	S_3	S_2	S_1

↑ carry out

$X = S_5 + S_4(S_3 + S_2)$

Sum	S_5	S_4	S_3	S_2	S_1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
1					


S3
S2
(S3+S2)
S4.(S3+S2)
S4
S5
S5+[S4.(S3+S2)]
A4 A3 A2 A1
S5
74LS383
4-bit parallel adder
S4 S3 S2 S1
B4 B3 B2 B1
Cin
x = S5+S4(S3+S2)
S4.(S3+S2)
S3+S2
S3
S2
S4
carry out
(Σ5)
A4 A3 A2 A1
74LS383
4-bit parallel adder
Cout
(not used)
S4 S3 S2 S1
B4 B3 B2 B1
Σ4 Σ3 Σ2 Σ1
Cin