UNIT-2

- Karnaugh Map (K-Map) -

2 - variable K-Map -

$F = \sum(0,1,3)$

$F = \overline{A} + B$.

A \ B	$(\overline{B})$ 0	(B) 1
$(\overline{A})$ 0	1	1
(A) 1		1

3 - variable K-Map -

$f = \sum(0,1,5)$

$= \overline{A}\,\overline{B} + \overline{B}C$

A \ BC	$(\overline{B}\,\overline{C})$ 00	$(\overline{B}C)$ 01	(BC) 11	$(B\overline{C})$ 10
$(\overline{A})$ 0	1	1		
(A) 1		1		

4 - variable K-Map -

$F = \sum(3,4,5,7,9,13,14,15)$

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

AB \ CD	00	01	11	10
00			1	
01	1	1	1	
11		1	1	1
10		1		

Don't Care Conditions -

Don't care combination -

Minimal expression in SOP form:
$X \to 1$

Minimal expression in POS form:
$X \to 0$.

Ex - $F_2(A,B,C,D) = \sum(5,6,9,13,15) + \underbrace{d(1,7,14)}_{\text{Don't Cares}}$

AB \ CD	00	01	11	10
00	(0)	X (1)	(3)	(2)
01	(4)	1 (5)	X (7)	1 (6)
11	(12)	1 (13)	1 (15)	X (14)
10	(8)	1 (9)	(11)	(10)

$F_2(A,B,C,D) = BC + \bar{C}D$

Don't care condition helps to minimise the no. of groups, hence, simplifying the expression.

The same above function can be written in POS form as

$F_2(A,B,C,D) = \pi(0,2,3,4,8,10,11,12) \cdot d(1,7,14)$

AB \ CD	00	01	11	10
00	0	X	0	0
01	0		X	
11	0			X
10	0		0	0

$F_2' = \bar{B}C + \bar{C}\bar{D}$

By De-Morgan's law

$F_2 = (B+\bar{C}) \cdot (C+D)$

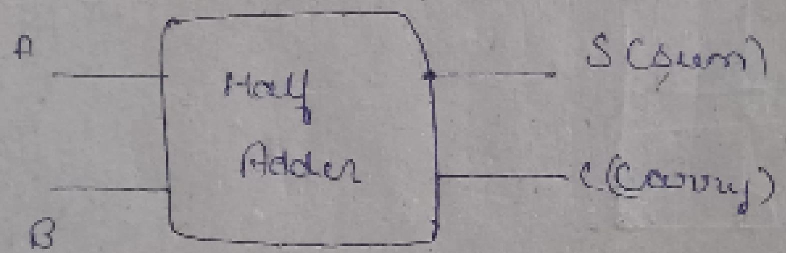
It is not important to include all the don't cares while forming equation but can be used if required.

• Half Adder -

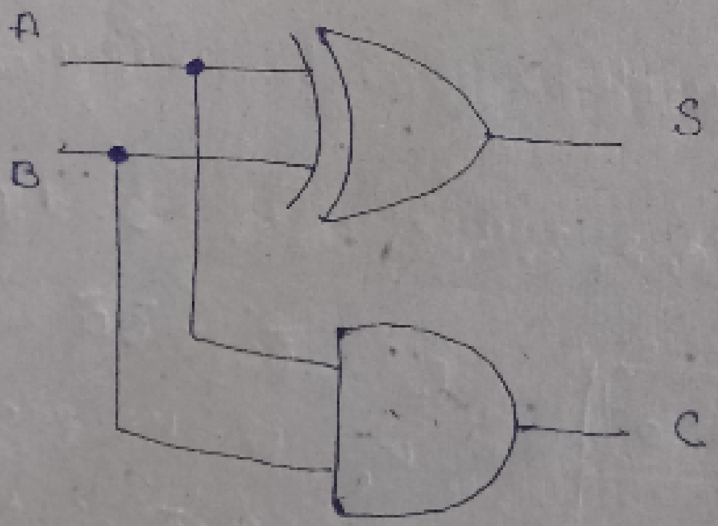
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

$C = AB$

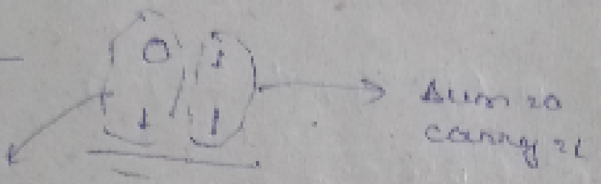


Logic circuit -



Using the half adder we cannot generate the result if there is a carry from the previous operation.

Eg —

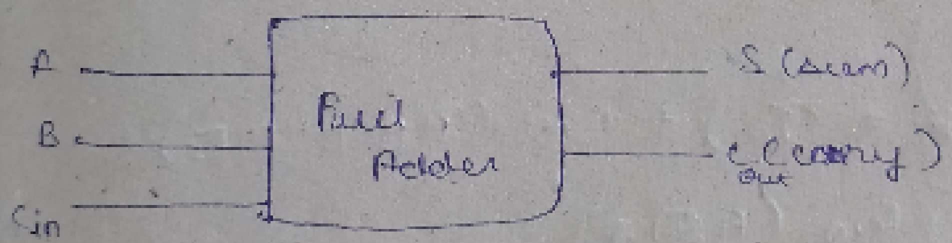


this cannot be calculated using half adder because there is a carry 1.

For this we use Full adder.

• Full Adder -

Is used if you want to add 2 bits alongwith the carry bit.



A	B	Cin	sum	Cout
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

Boolean expression of sum and carry –

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

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

$$= \bar{A}\underbrace{(B \oplus C_{in})}_{x} + A\underbrace{(\overline{B \oplus C_{in}})}_{\bar{x}}$$

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

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

Using K-Map –

A \ BC_{in}				
			1	
		1	1	1

$$\boxed{C_{out} = AB + BC_{in} + AC_{in}}$$

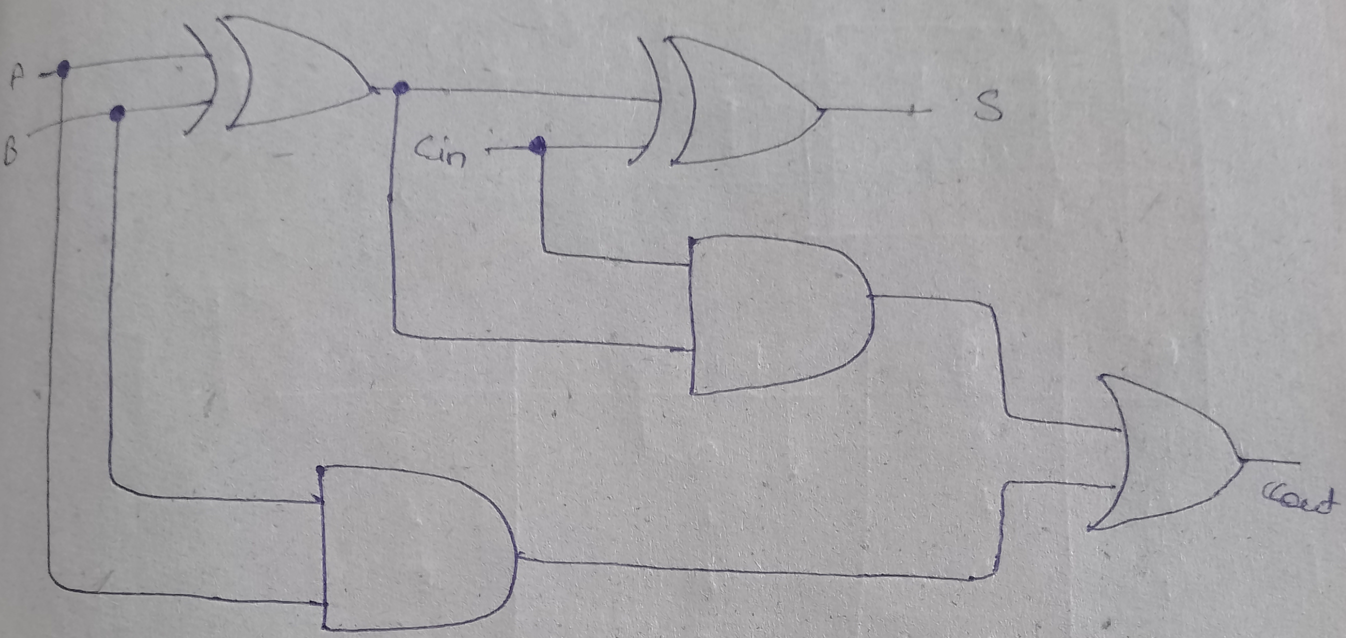
Or

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

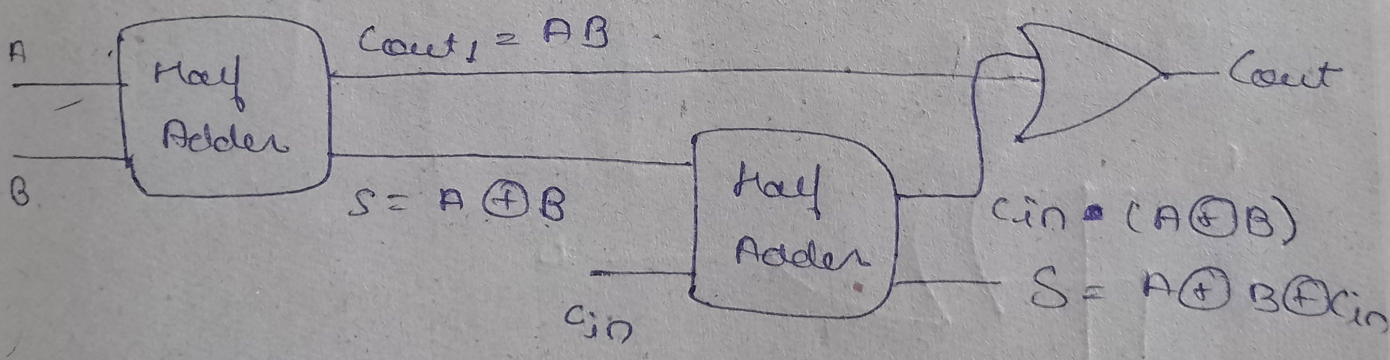
$$= AB + C_{in}(A \oplus B)$$

$$\boxed{C_{out} = AB + BC_{in} + AC_{in} = AB + C_{in}(A \oplus B)}$$

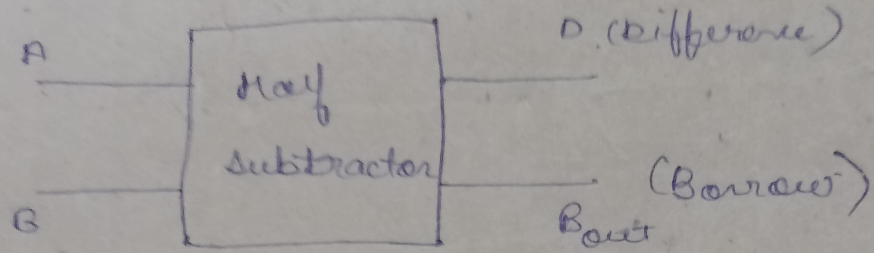
Logic circuit -



Full adder using half adders -



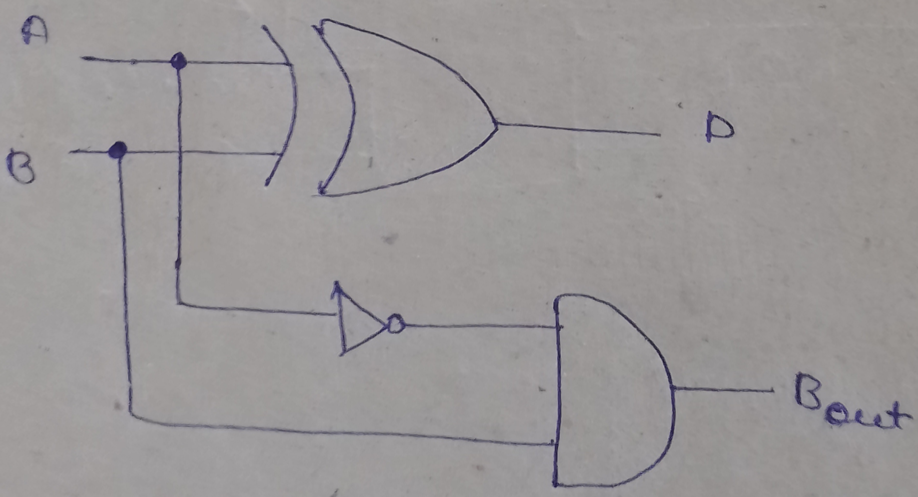
• Half Subtractor –



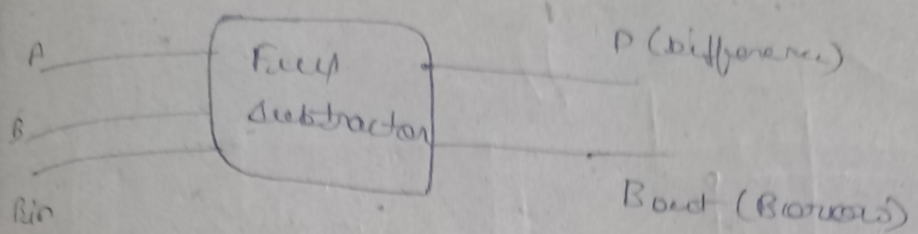
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$D = \overline{A}B + A\overline{B} = A \oplus B$.

$B_{out} = \overline{A}B$.



Full subtractor-



A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

$$D = \bar{A}\bar{B}B_{in} + \bar{A}B\overline{B_{in}} + A\bar{B}\overline{B_{in}} + ABB_{in}$$

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

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

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

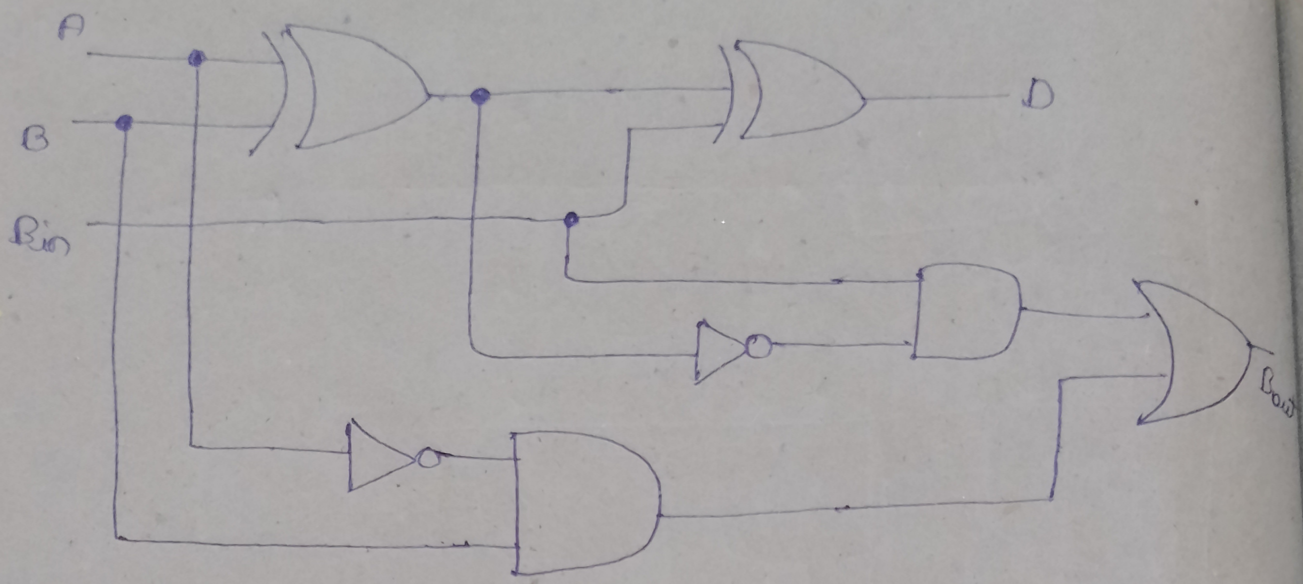
$$B_{out} = \bar{A}\bar{B}B_{in} + \bar{A}B\overline{B_{in}} + \bar{A}BB_{in} + ABB_{in}$$

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

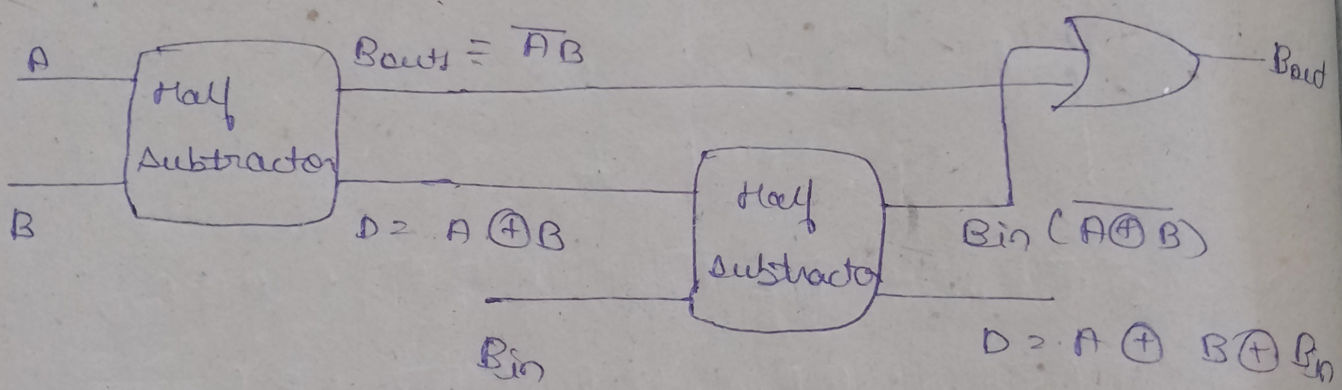
$$= \bar{A}\bar{B} + B_{in}(\overline{A \oplus B})$$

$$=$$

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

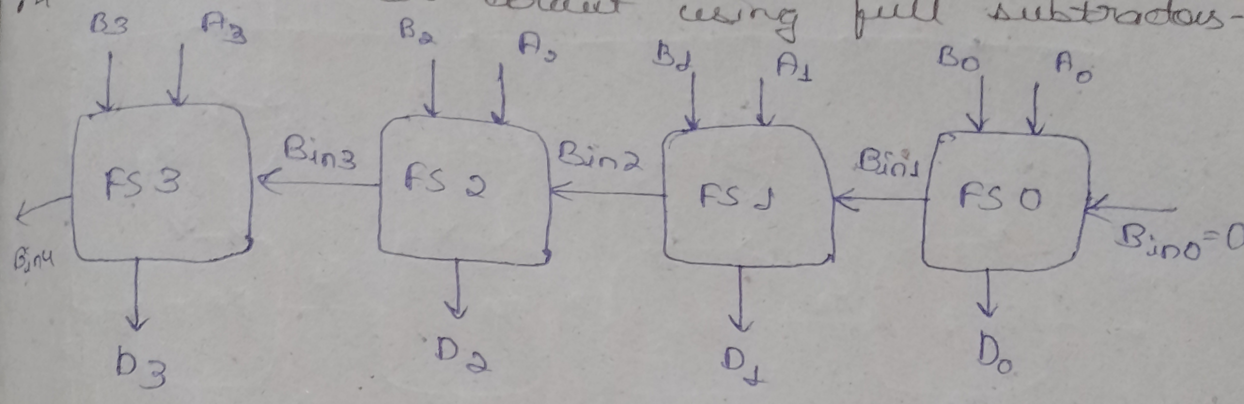


Full subtractor using half subtractor -



4-BIT ADDER AND SUBTRACTOR CIRCUITS

• 4-bit subtractor circuit using full subtractors-



	A_3	A_2	A_1	A_0
$-$	B_3	B_2	B_1	B_0
	D_3	D_2	D_1	D_0

		1	1	1	0
	$-$ 0	1	0	1	
0	1	0	0	1	

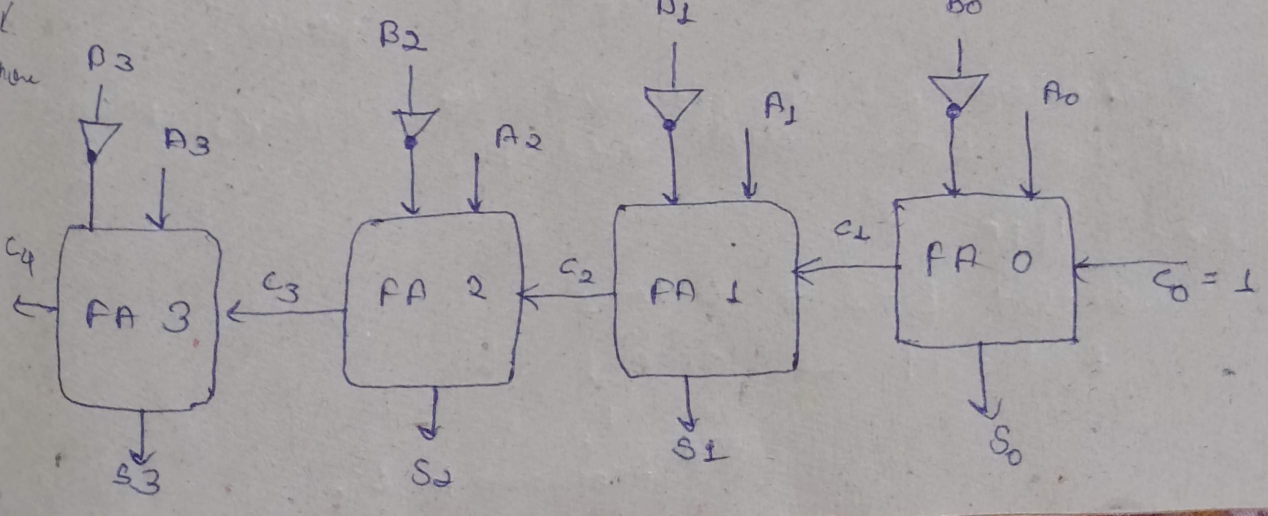
• 4-bit subtractor using adder circuit-

$$\begin{array}{r} 14 \\ -5 \\ \hline \end{array} \Rightarrow 14 + (2\text{'s complement of } 5)$$

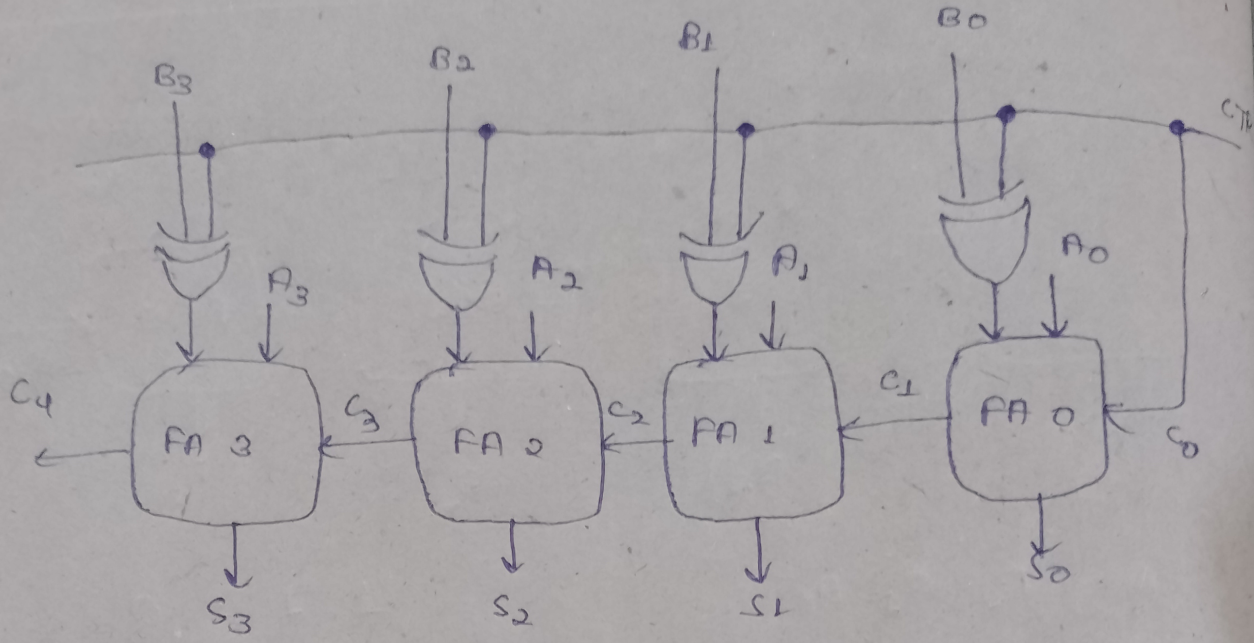
(2's complement of (0101) 5 → 1011)

$$\begin{array}{rl} 1110 & \rightarrow 14 \\ +1011 & \rightarrow \text{2's complement of } 5 \\ \hline (1)\ 1001 & \rightarrow 9 \end{array}$$

ignore



• 4 bit Adder / subtractor -



This circuit can be used for both addition and subtraction.
CTR will be 1 in case of subtraction and 0 in case of addition.

10 (A)
− 12 (B)

when A < B and the answer has no carry then the answer is in two's complement form.

↓

1010 → 10
0100 → 2's complement of 12
1110 → 2's complement form

To convert 2's complement to binary, invert all the bits and add 1.

Ex - 1110

Invert bits - 0001

Add 1 - 0001
+ 1
0010 → Ans.