Half - Subtractor

=> Half Subtractor is a combinational circuit which is used to perform subtraction of two bits. It has two inputs 'A' & 'B' (minuend & subtrahend) and two outputs 'D' (Difference) and 'Bo' (borrow).

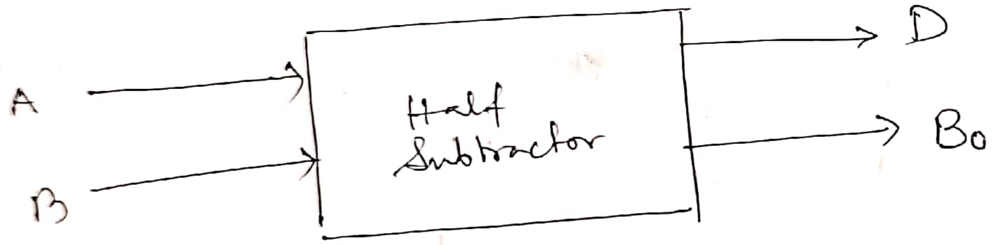


Fig. - Logic Symbol

Inputs		Outputs	
A	B	D	Bo
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

=> From the above truth table, it is visible that the difference (D) is '0' if $A = B$ and $D = 1$, if $A \neq B$.

=> The borrow output 'Bo' is '1', whenever $A = 0$ and $B = 1$ or 'A' is lesser than 'B'.

=> So, the boolean expression for difference and Borrow output can be written as -

$$D = \bar{A}B + A\bar{B} = A \oplus B$$

$$B_0 = \bar{A}B$$

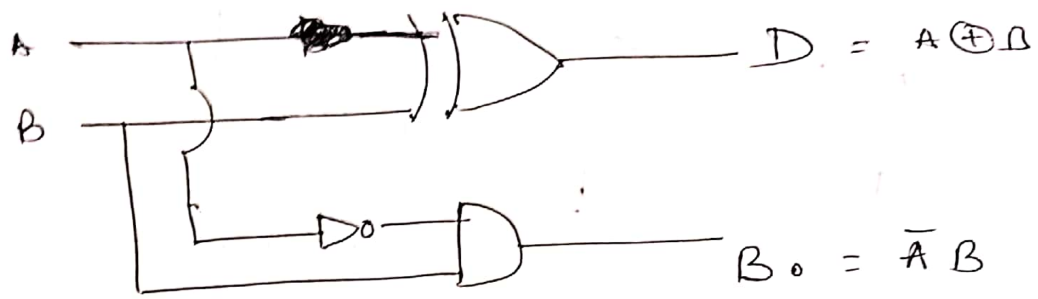


Fig.- Half Subtractor using Logic Gates

Half Subtractor using universal Logic :-

① using NAND Gate :-

$$D = A \oplus B = \bar{A}B + A\bar{B}$$

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

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

$$= \cancel{(\bar{A} + \bar{B})}($$

$$= B \cdot \overline{AB} + A \cdot \overline{AB}$$

$$= \overline{\overline{B \cdot \overline{AB}} \cdot \overline{A \cdot \overline{AB}}}$$ [using De-Morgan's]

$$B_0 = \bar{A}B = \bar{A}B + B\bar{B}$$

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

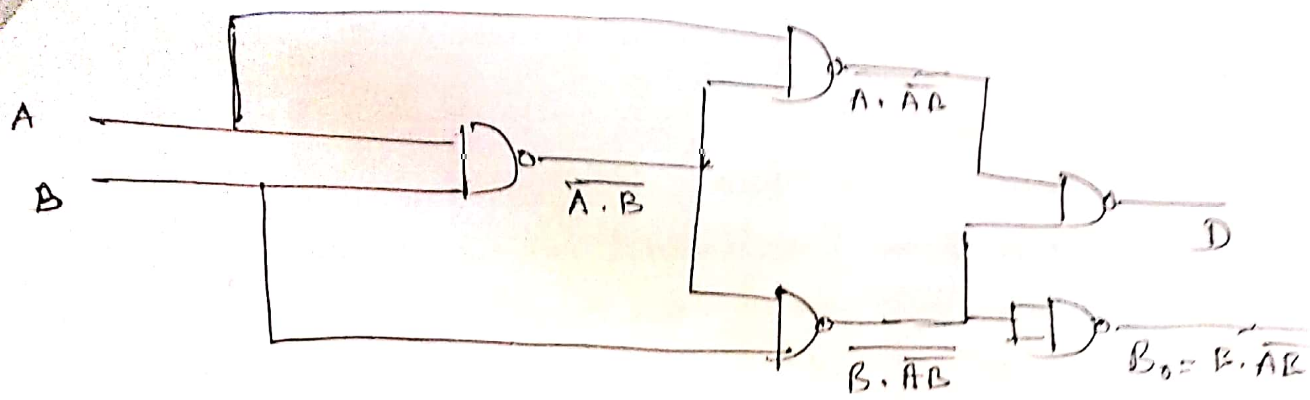


Fig → Half Subtractor using NAND Logic

② using NOR-Logic:-

$$D = A \oplus B = \bar{A}B + A\bar{B}$$

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

$$[\because B\bar{B} = 0 = A\bar{A}]$$

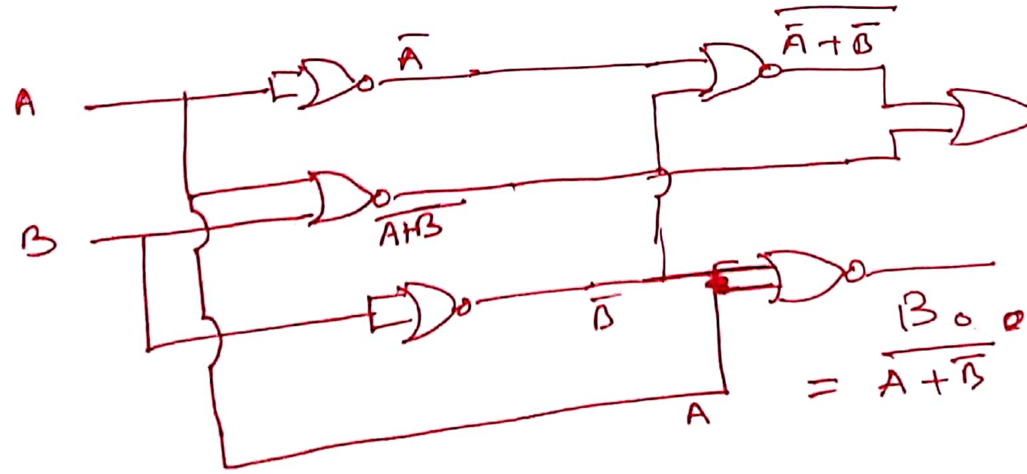
$$= B(\bar{A}+\bar{B}) + A(\bar{A}+\bar{B})$$

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

$$\cancel{B_0 = \bar{A}B = \bar{A}B + B\bar{B} = B(\bar{A}+\bar{B})}$$

$$B_0 = \bar{A}B = \overline{\overline{\bar{A}B}} = \overline{\bar{\bar{A}} + \bar{B}}$$

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



$$D = \overline{\overline{A+B} + \overline{\bar{A}+\bar{B}}}$$

$$= \overline{\overline{(A+B)}} \cdot \overline{\overline{\bar{A}+\bar{B}}}$$

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

B_0

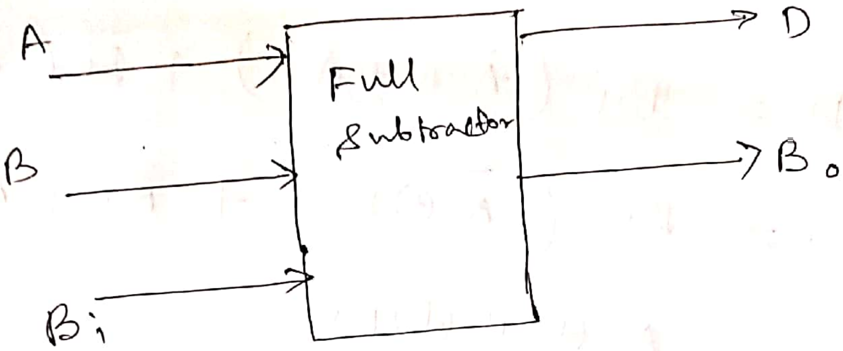
$$= \overline{A+\bar{B}}$$

Scanned with CamScanner

Full Subtractor

=> A full Subtractor is a Combinational Circuit that performs Subtraction involving three bits, namely minuend, Subtrahend & bit and the borrow from the previous stage.

Logic Symbol :-



Bi = Borrow input from previous stage.

Truth Table :-

Inputs			Outputs	
A	B	Bi	D	Bo
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

From the truth table;

$$D = \bar{A}\bar{B}B_i + \bar{A}B\overline{B_i} + A\bar{B}\overline{B_i} + ABB_i$$

Simplifying the above expression →

$$D = B_i(\bar{A}\bar{B} + AB) + \overline{B_i}(\bar{A}B + A\bar{B})$$

$$= B_i(\overline{A \oplus B}) + \overline{B_i}(A \oplus B)$$

$$= A \oplus B \oplus B_i$$

* Similarly for B_o, the SOP expression can be written as →

$$B_o = \bar{A}\bar{B}B_i + \bar{A}B\overline{B_i} + \bar{A}BB_i + ABB_i$$

The expression can be simplified as K-map.

A \ BB_i	00	01	11	10
0	0	1	1	1
1	0	0	1	0

I_1 I_2 I_3

$$\therefore \quad B_o = \bar{A}B + \bar{A}B_i + BB_i$$

Using the above simplified expressions, the full-subtractor can be realized as shown below:-

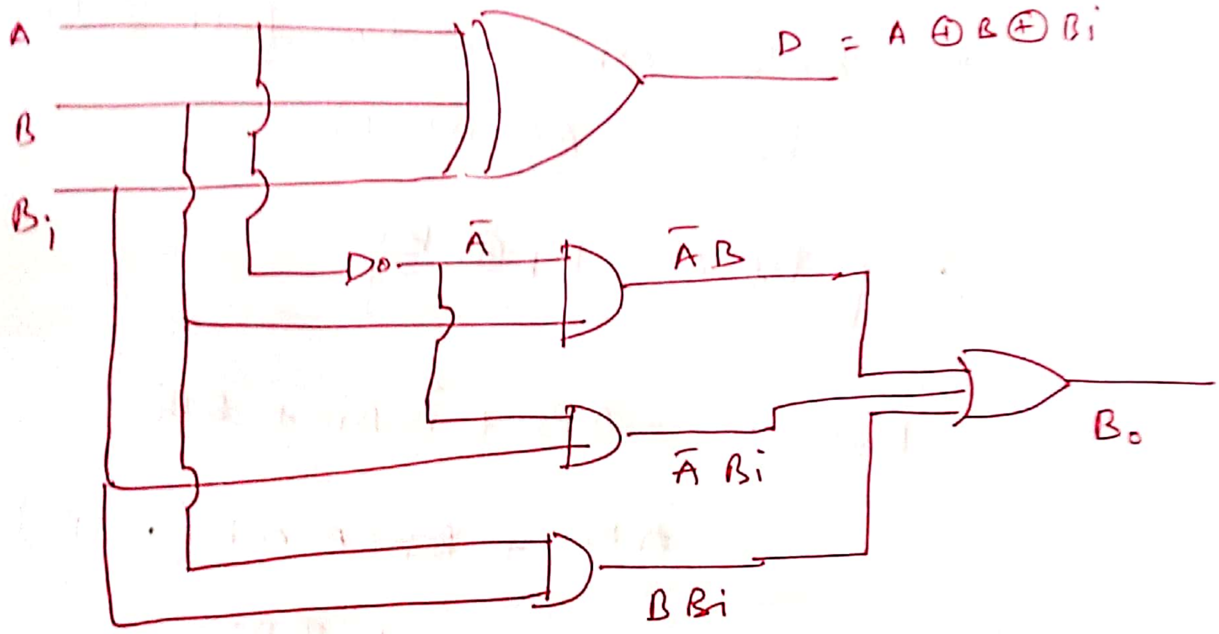
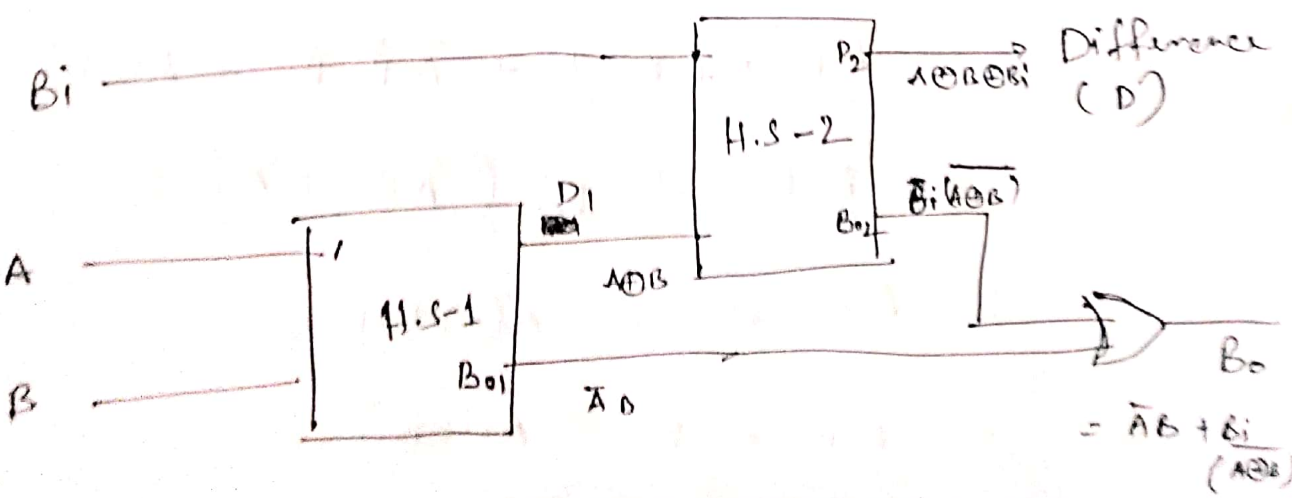


Fig. Full Adder Circuit using Logic Gates

---

* The full-subtractor can be implemented using two half-subtractors and an-OR Gate as shown below.



The boolean expression for half subtractor →

$$D_h = A \oplus B$$

$$B_{oh} = \bar{A} B$$

The boolean expression for full subtractor →

$$D_f = \underline{A \oplus B} \oplus B_i$$

$$\therefore \boxed{D_f = D_h \oplus B_i}$$

$$B_{of} = \bar{A} B + \bar{A} B_i + B B_i$$

$$= \bar{A} B + \bar{A} B_i (B + \bar{B}) \quad [\because B + \bar{B} = 1]$$

$$+ B B_i$$

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

$$= \bar{A} B (1 + B_i) + \bar{A} \bar{B} B_i + B B_i$$

$$= \bar{A} B + \bar{A} \bar{B} B_i + B B_i \quad [\because 1 + B_i = 1]$$

$$= \bar{A} B + \bar{A} \bar{B} B_i + B B_i (A + \bar{A}) \quad [\because A + \bar{A} = 1]$$

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

$$= \bar{A} B (1 + B_i) + \bar{A} \bar{B} B_i + A B B_i$$

$$= \bar{A} B + B_i (A B + \bar{A} \bar{B})$$

$$= \bar{A} B + B_i (A \odot B)$$

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

Scanned with CamScanner