

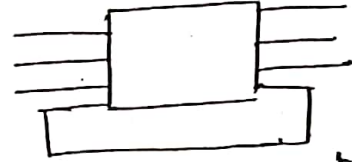
Digital Circuits

Combinational Circuits



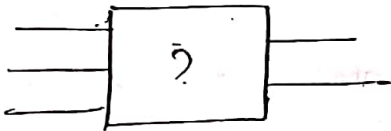
- present output depends on only present input
- No feedback
- No memory
- ~~HA, HS, MUX~~
- HA, HS, MUX

Sequential Circuits



- present o/p $\left\{ \begin{array}{l} \text{present i/p} \\ \text{previous or past o/p} \end{array} \right.$
- feedback
- Memory
- F/P, Register, Counter.

Combinational Circuits

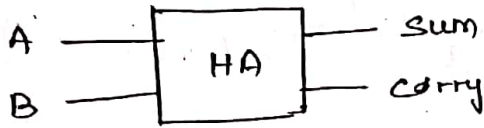


- 1> Identify number of i/p & o/p's ✓
- 2> Construct truth table ✓
- 3> Write logical exp in SOP / POS
- 4> Minimize logical exp
- 5> Implement logic ckt.

Arithmetic Circuits

HALF ADDER

0	0	0	1	1
0	1	0	1	1
0	1	1	0	0
			↑	↑
			carry	sum



Truth table

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

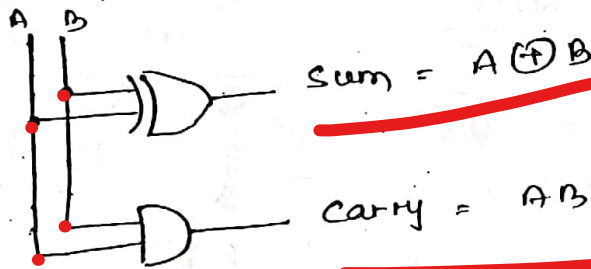
Logical Expression,

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

$$= A \oplus B$$

$$\text{CARRY} = AB$$

Logic Circuit



HA

1> logical exp sum - $A \oplus B$

2> logical exp carry - AB

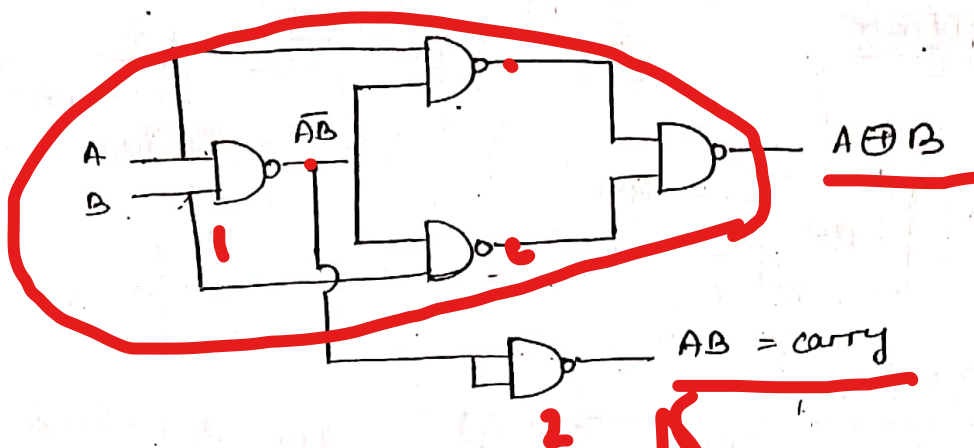
3> Min. no of NAND - 5

4> " " " NOR - 5

5> No of MUX - $3 \rightarrow (2 \times 1), 2 \rightarrow (4 \times 1)$

6> No of decoder - 1

Half adder using NAND



So no. of NAND gate for sum = 4

no of NAND gate for carry = 2

(4, 2)

Overall = 5

W.B

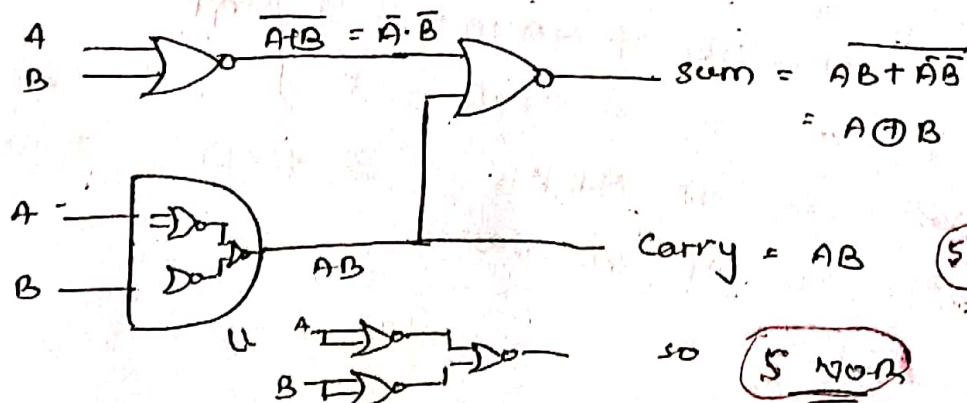
11-01 > (d)

Half adder using NOR

$$\text{EXOR} \rightarrow A \oplus B = \overline{A \odot B} = \overline{AB + \overline{A}\overline{B}} \quad \textcircled{1}$$

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

EXOR-gate can be implemented using two NOR gate & 1-AND gate.



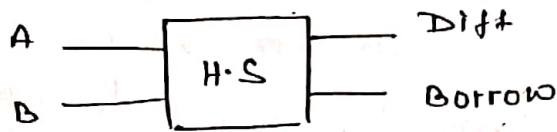
so

(5 NOR)

carry = 3
sum = 2

Replace Add.

Half Subtractor :



A-B

Truth Table

A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

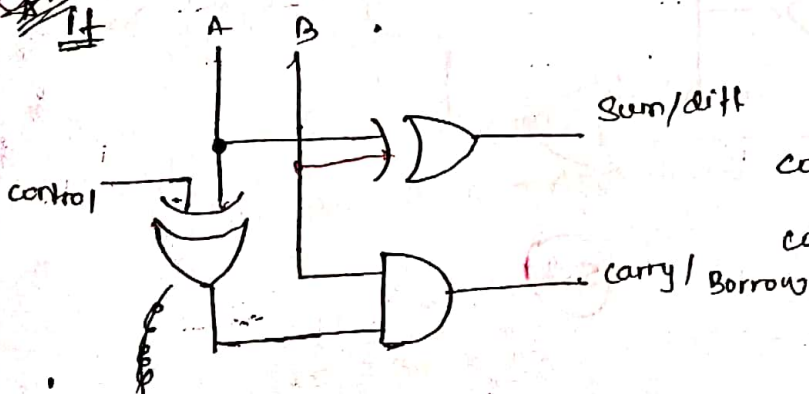
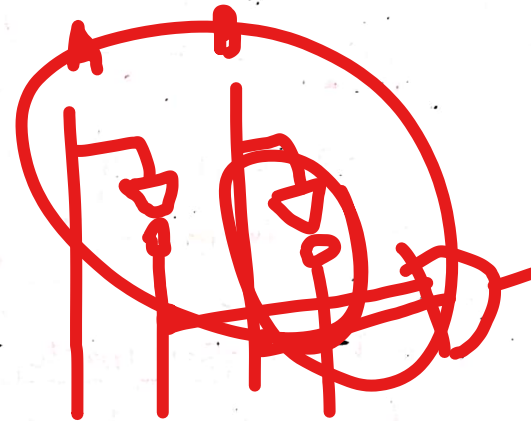
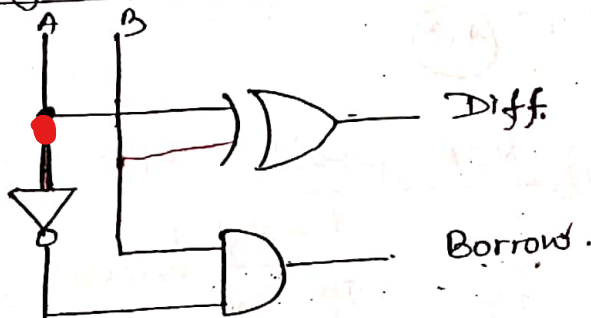
Diff.

Logical Expression

$$\text{Difference} = \bar{A}B + A\bar{B} = A \oplus B$$

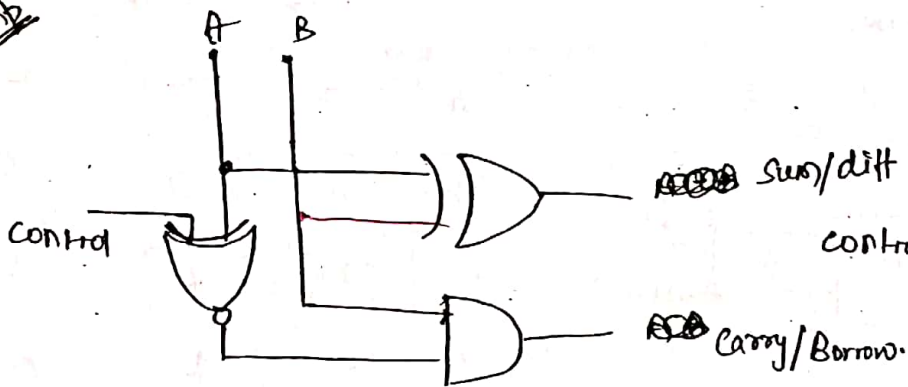
$$\text{Borrow} = \bar{A}B$$

Logical Circuit

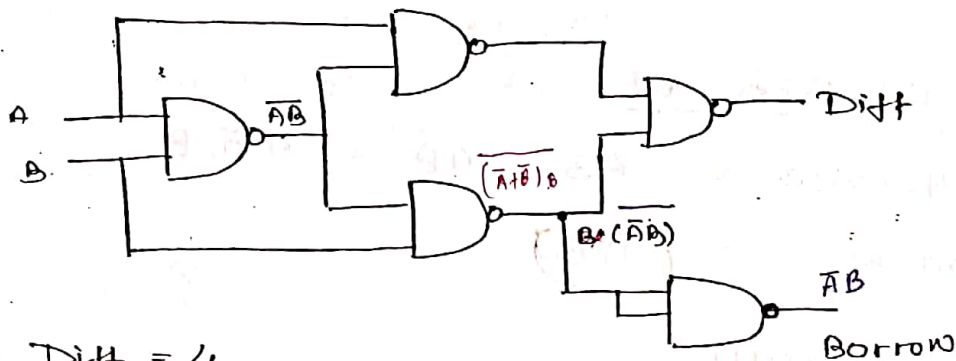


$$\text{control} = 0 \rightarrow \text{H.A}$$

$$\text{control} = 1 \rightarrow \text{H.S}$$



Half subtractor using NAND

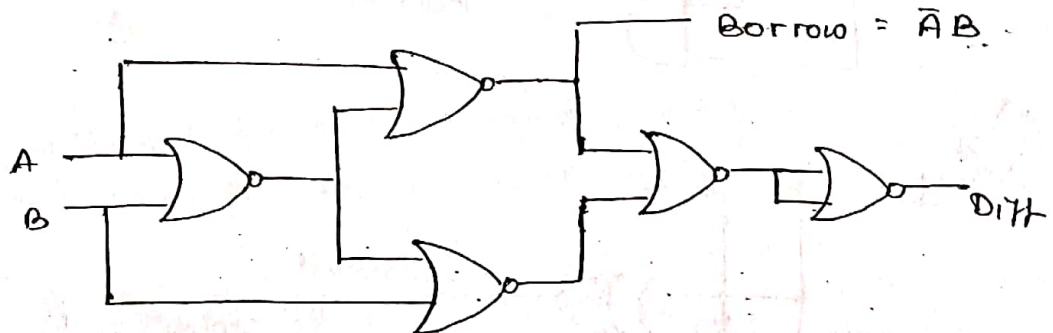


So Diff = 4

Borrow = 3

(4,3)

Half Subtractor using NOR gate



Diff \rightarrow 5

Borrow \rightarrow 2

(5,2)