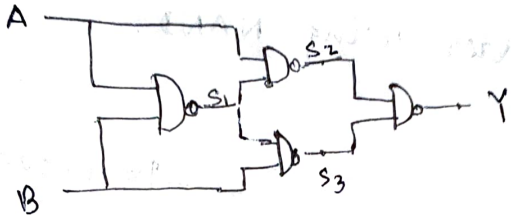
1(a) Design of AND Using NAND.

$S_1 = \overline{(A.B)}$

$Y = \overline{(S_1.S_1)} = \bar{S}_1 = A.B$

1(b) Design of XOR using NAND.



$S_1 = \overline{(A.B)} = \bar{A} + \bar{B}$

$S_2 = \overline{(A.(\bar{A}+\bar{B}))}$

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

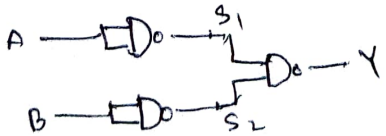
$S_3 = \overline{((\bar{A}+\bar{B}).B)} = \overline{(\bar{A}.B)} = A + \bar{B}$

$Y = \overline{(S_2.S_3)} = \overline{(\bar{A}+B).(A+\bar{B})}$

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

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

1(c) Design of OR gate using NAND.

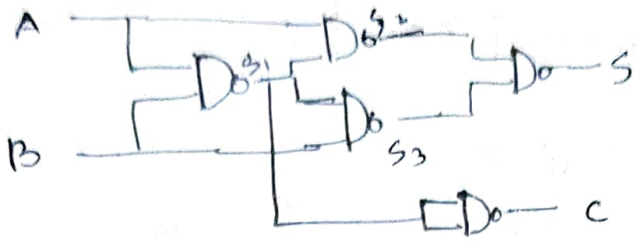


$S_1 = \overline{(A.A)} = \bar{A}$

$S_2 = \overline{(B.B)} = \bar{B}$

$Y = \overline{(S_1.S_2)} = \overline{(\bar{A}.\bar{B})} = A + B$

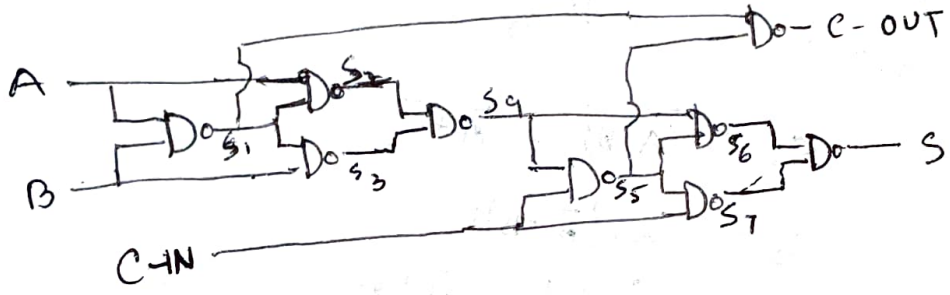
2. (a) Half Adder using NAND.



$S = A \oplus B$ (XOR gate as designed before)

$C = \overline{(\overline{A.B})} = A.B$

(b) Full Adder using NAND



$S = A \oplus B \oplus (C\text{-}IN)$

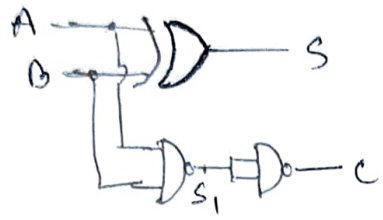
~~$C\text{-}OUT = \overline{(A.B).C + (\overline{A.B}).(C\text{-}IN)}$~~

~~$= (A.B) + C\text{-}IN$~~

$C\text{-}OUT = \overline{\overline{(\overline{A.B})}.\overline{(C\text{-}IN.(A \oplus B))}}$

$= A.B + (C\text{-}IN).(A \oplus B)$

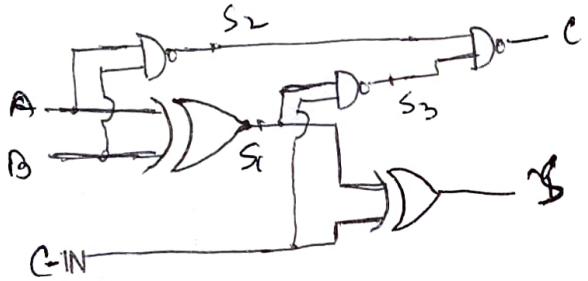
3. (a) Usage of XOR designed § before ~~to use~~ along with NAND for half adder.



$S = A \oplus B$

$C = \overline{(\overline{A \cdot B})} = A \cdot B$

(b) Using XOR designed before ~~to use~~ along with NAND for full adder.



$C = A \cdot B + C\text{-}IN \cdot (A \oplus B)$

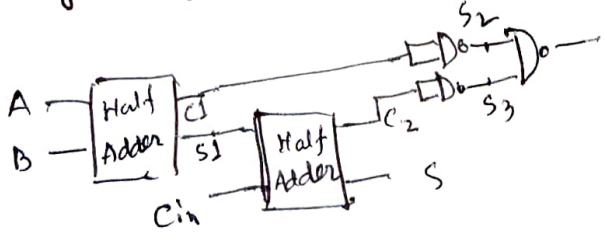
$= \overline{(\overline{A \cdot B}) \cdot (\overline{C\text{-}IN \cdot (A+B)})}$

$S = A \oplus B \oplus C\text{-}IN$

$C = \overline{\left((\overline{A \cdot B}) \cdot (\overline{C\text{-}IN \cdot (A \oplus B)}) \right)}$

$= A \cdot B + C\text{-}IN \cdot (A \oplus B)$

Using half-adder.



NKulkarni
7-8-2025