INTRODUCTION TO BOOLEAN ALGEBRA AND LOGIC GATES

* Binary logic is used in all digital computers and devices

* Finding simpler and cheaper but equivalent realizations of a circuit is important to reduce the overall cost of design.

* Mathematical methods that simplify circuits rely primarily on Boolean Algebra.

Definition of Boolean Algebra:

• "Boolean Algebra is a mathematical system that is defined with a set of elements, operators and a number of postulates/axioms."

* Postulates form the basic assumptions from which it is possible to deduce rules and theorems.

Boolean Logic Operations :

The basic logic operations are

1. Logical AND operation ($Y = A \cdot B$ Logical multiplication)
2. Logical OR operation ($Y = A + B$ Logical Addition)
3. Logical complement / Inverse operation ($Y = \bar{A}$)

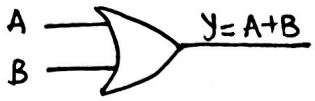
LOGIC GATES :

* Logic gate is an electronic circuit which makes logical decisions by taking one or more input and produce an output.

* Commonly used logic gates are

1. OR
2. AND
3. NOT
4. NAND } universal gates
5. NOR } universal gates
6. EXOR - Inequality detector
7. EX-NOR - Equality detector.

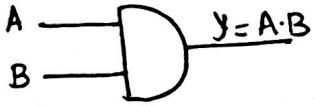
① OR GATE (IC7432)



Truth table for OR

A	B	y = A+B
0	0	0
0	1	1
1	0	1
1	1	1

② AND GATE (IC7408)



Truth table for AND

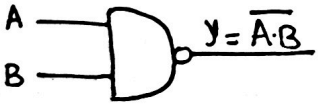
A	B	y = A·B
0	0	0
0	1	0
1	0	0
1	1	1

③ NOT GATE (IC7404)

Truth table for NOT

A	y = Ā
0	1
1	0

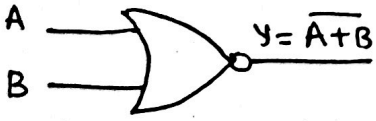
④ NAND GATE (IC7400)



Truth table for NAND

A	B	y = $\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

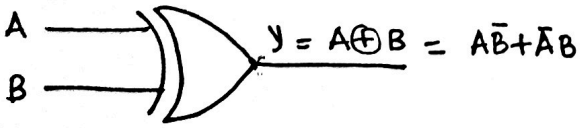
⑤ NOR GATE (IC7402)



Truth table for NOR

A	B	y = $\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

⑥ EX-OR GATE (Inequality Detector)



Truth table for EX-OR

A	B	y = A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

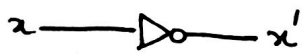
NAND AND NOR IMPLEMENTATIONS

* Digital circuits are frequently constructed with NAND or NOR gates rather than with AND and OR gates.
* NAND and NOR gates are easier to fabricate.
* Because of the prominence of NAND and NOR gates in the design of digital circuits, some rules and procedures have been developed for the conversion from AND, OR, NOT logic diagrams into equivalent NAND and NOR logic diagrams.

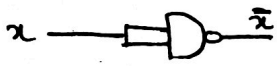
NAND CIRCUITS IMPLEMENTATION

* The NAND gate is said to be a 'universal gate' because any logic circuit can be implemented with it.
* The implementation of AND, OR, NOT operations with NAND gate are as follows:

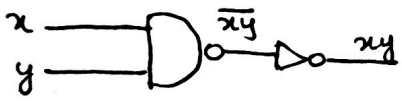
Inverter



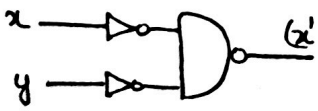
(or)



AND



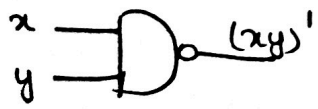
OR



$(x' \cdot y')' = (x')' + (y')' = x + y$

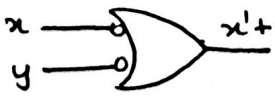
• Graphic symbols for NAND Gate.

① NAND Gate



≡

② Invert-OR Gate



x	y	xy	(xy)'
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

≡

x	y	x'	y'	x'+y'
0	0	1	1	1
0	1	1	0	1
1	0	0	1	1
1	1	0	0	0

∴ Invert-OR Gate is equivalent to NAND Gate.

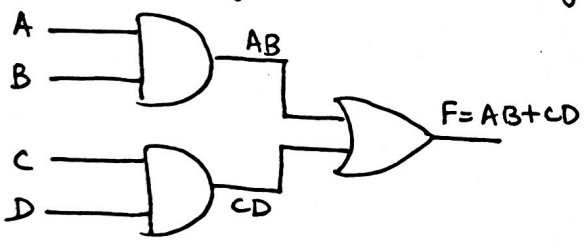
General Procedure for constructing All-NAND diagram.

1. Construct the AND-OR-NOT logic diagram for the given expression
2. Convert all OR gates to Invert-OR &
   all AND gates to NAND gates.
3. check all the bubbles in the diagram. For every bubble that is not compensated with another small circle (inverter), insert an inverter along the same line or complement the input literal
4. Finally, convert Invert-OR to NAND gates &
   all NOT gates to 1-input NAND gate.

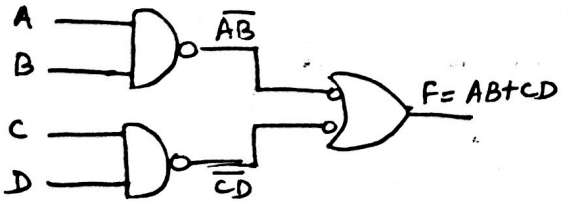
① Realize the function $F = AB + CD$ using AND-OR Gates and NAND Gates.

Solution:

Step 1: Implementing Function F using AND-OR Gates:



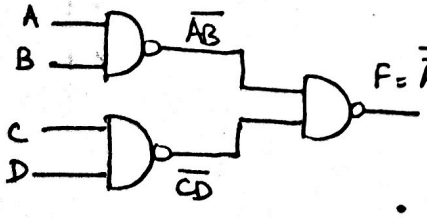
Step 2: Convert all OR gates to Invert-OR &
all AND gates to NAND gates



Step 3:

Check all the bubbles in the diagram. All the bubbles are compensated.

Step 4: Convert all Invert-OR to NAND &
NOT with 1-Input NAND gate.



$F = \overline{\overline{AB} \cdot \overline{CD}} = \overline{\overline{AB}} + \overline{\overline{CD}}$

$F = AB + CD$

$\therefore$ $F = AB + CD$ is implemented with NAND Gates.

② Implement the following Boolean function with NAND gates.
$F(x,y,z) = \Sigma(1,2,3,4,5,7)$

Solution:

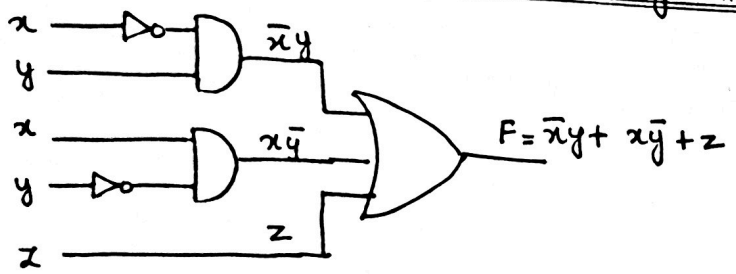
Given the minterms. The function has to be minimized using K-map

$x \backslash yz$	$\bar{y}\bar{z}$ 00	$\bar{y}z$ 01	yz 11	$y\bar{z}$ 10
$\bar{x}$ 0	0	1 (1)	1 (3)	1 (2)
x 1	1 (4)	1 (5)	1 (7)	6

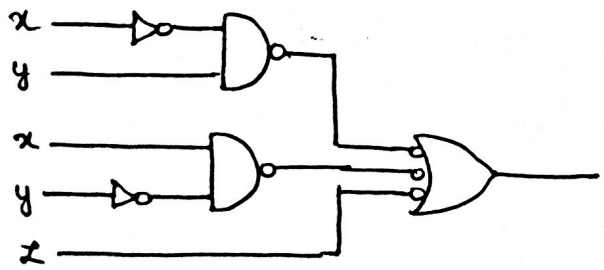
[Group1 (1,3,5,7) = z
Group2 (2,3) = $\bar{x}y$, Group3 (4,5) = $x\bar{y}$]

∴ Simplified $F = z + \bar{x}y + x\bar{y}$

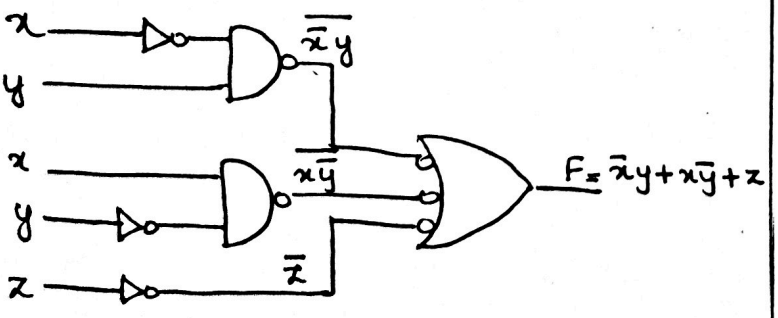
Step 1: Implementing function F using AND-OR Gates



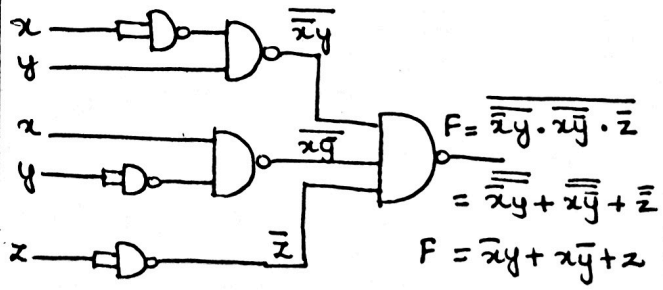
Step 2: Convert all OR gates to Invert-OR & all AND gates to NAND gates.



Step 3: Check all the bubbles in the diagram. If any bubble is not compensated, insert an inverter on the same line.



Step 4: Convert all Invert-OR to NAND & NOT to 1-Input NAND gate

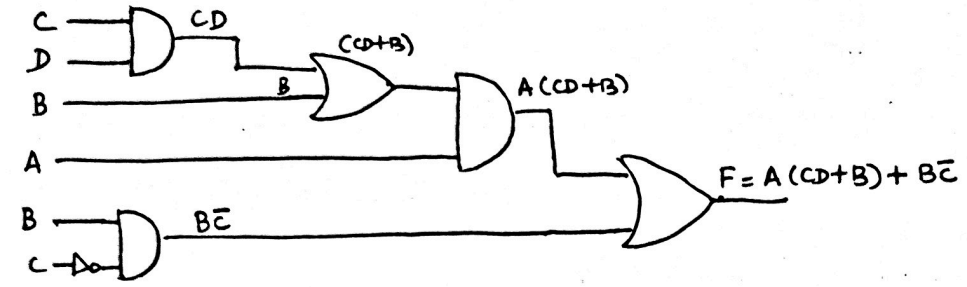


∴ The function $F = z + \bar{x}y + x\bar{y}$ is implemented with NAND gates

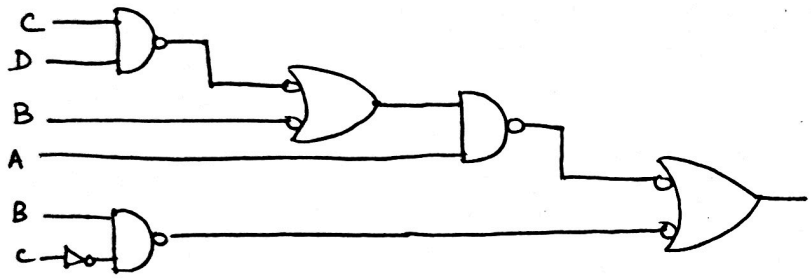
③ Implement $F = A(CD+B) + B\bar{C}$ with NAND Gates

Solution:

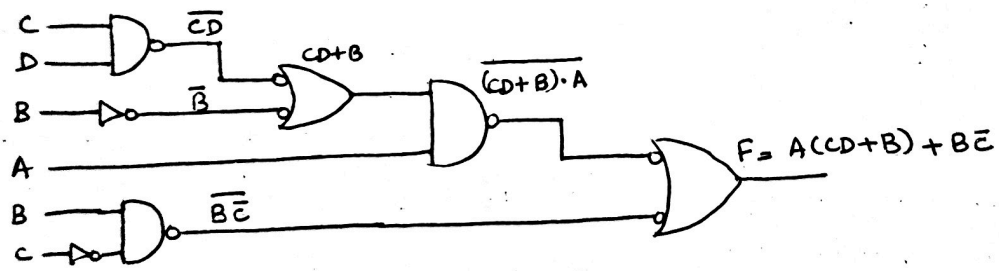
Step 1: Implementing Function F with AND-OR Gates:



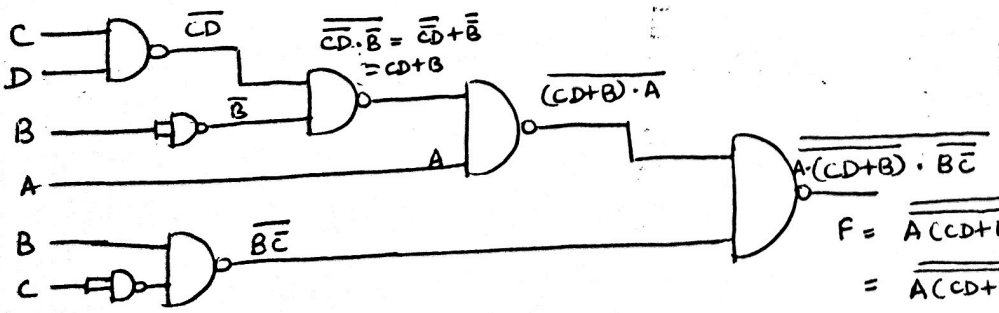
Step 2: Convert all OR gates to Invert-OR && all AND gates to NAND gates



Step 3: Check all the bubbles in the diagram. If any bubble is not compensated, insert an inverter on the same line.



Step 4: Convert all Invert-OR to NAND && NOT to 1-input NAND gate.



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

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

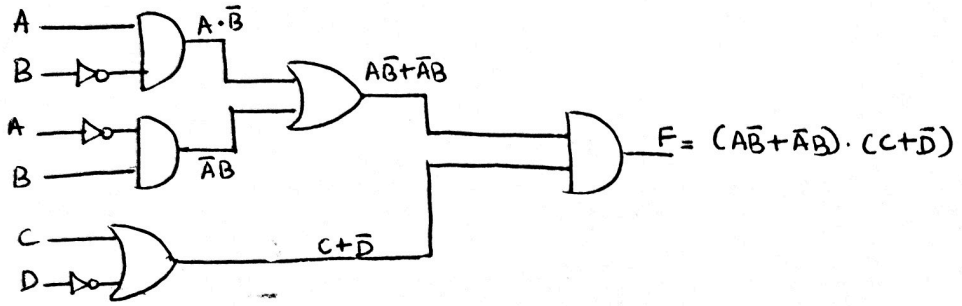
$$F = A(CD+B) + B\bar{C}$$

$\therefore$ $F = A(CD+B) + B\bar{C}$ is implement with NAND Gates

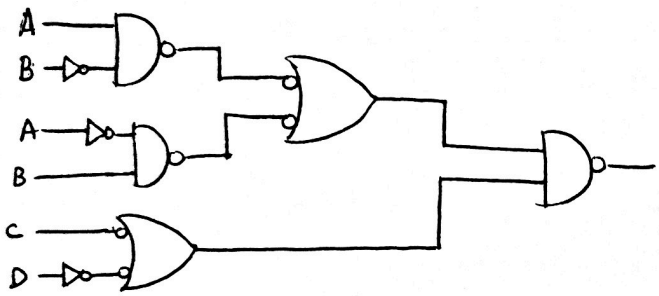
④ Implement $F = (AB' + A'B)(C+D')$ using NAND gates.

Solution:

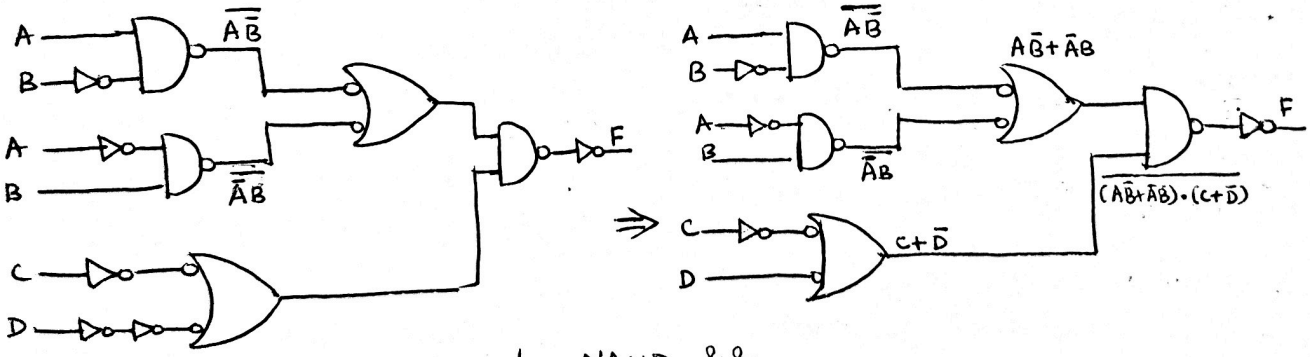
Step 1 : Implementing Function with AND-OR Gates:



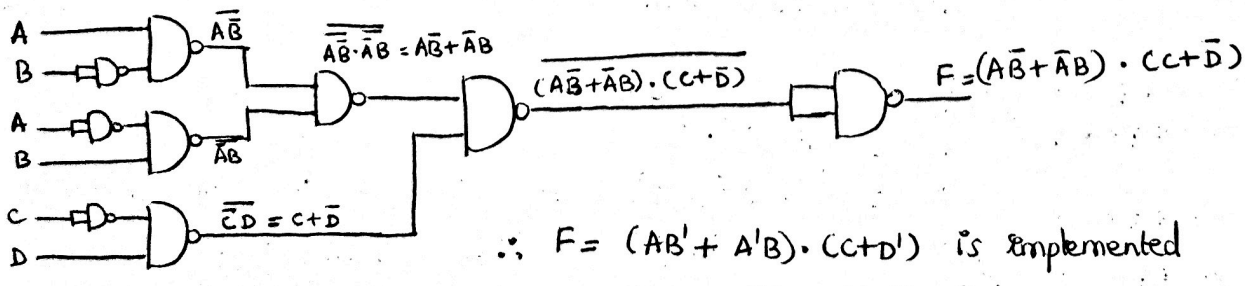
Step 2 : Convert all OR gates to Invert-OR & all AND gates to NAND gates.



Step 3 : Check all the bubbles in the diagram. If any bubble is not compensated, insert an inverter on the same line.



Step 4 : Convert all Invert-OR to NAND & NOT to 1-input NAND gate



$\therefore$ $F = (AB' + A'B)\cdot(C+D')$ is implemented with NAND Gates.

NOR CIRCUITS IMPLEMENTATION

* The NOR gate is another universal gate that can be used to implement any Boolean function.

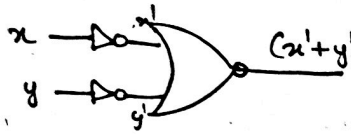
* The NOR operation is the dual of the NAND operation.

* The implementation of AND, OR and NOT operations with NOR gates are as follows.

Inverter

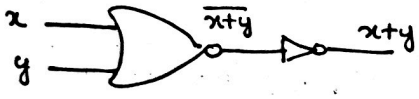
(or)

AND



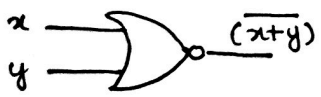
$(x'+y')' = (x')' \cdot (y')' = xy$

OR



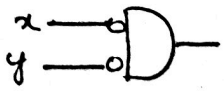
• Graphic symbols for NOR Gate •

① NOR Gate



≡

② Invert-AND gate



$\bar{x} \cdot \bar{y} = (\overline{x+y})$

x	y	$x+y$	$\overline{x+y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

≡

x	y	$\bar{x}$	$\bar{y}$	$\bar{x} \cdot \bar{y}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

∴ Invert-AND is equivalent to a NOR Gate.

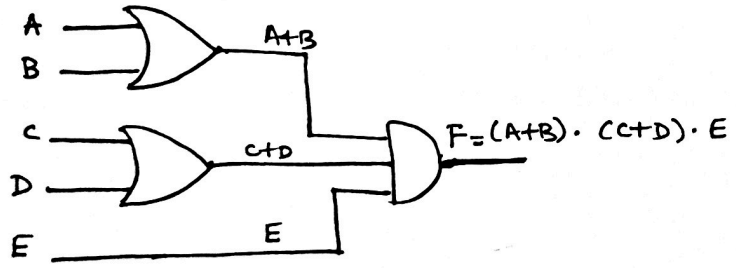
General Procedure for constructing All-NOR circuit diagram.

1. Construct the AND-OR-NOT logic diagram for the given expression
2. Convert all OR gates to NOR gates &&
   all AND gates to Invert-AND gates
3. Check all the bubbles in the diagram. For every bubble that is not compensated with another small circle (inverter), insert an inverter along the same line or complement the input literal.
4. Finally, Convert Invert-AND to NOR gates &&
   all NOT gates to 1-input NOR gate

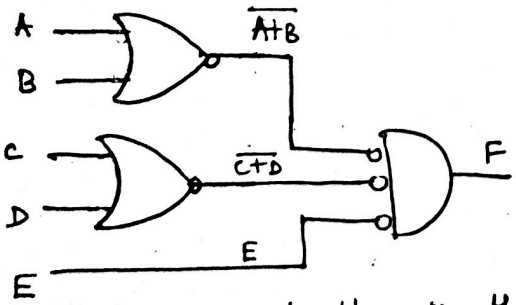
① Implement the following $F = (A+B)\cdot(C+D)\cdot E$ using NOR gates

Solution :

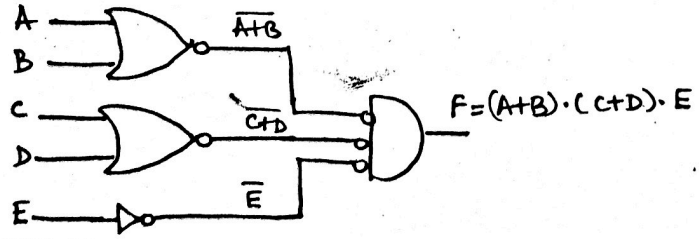
Step1: Implementing F using AND-OR Logic



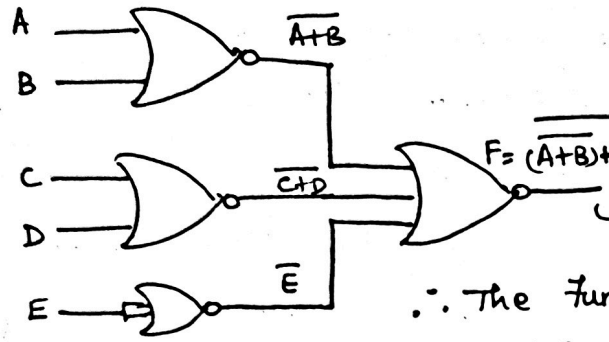
Step2 : Convert all OR gates to NOR &&
all AND to Invert-AND gates



Step3: Check all the bubbles in the diagram. If any bubble is not compensated add a inverter on the same line



Step 4 : Convert all Invert-AND to NOR & all NOT to 1-input NOR gate.



$F = \overline{\overline{(A+B)} + \overline{(C+D)} + \overline{E}} = \overline{\overline{(A+B)}} \cdot \overline{\overline{(C+D)}} \cdot \overline{\overline{E}}$

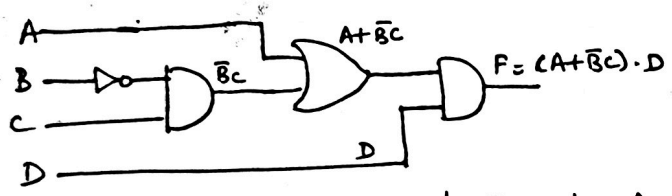
$\hookrightarrow F = (A+B) \cdot (C+D) \cdot E$

$\therefore$ The function $F = (A+B) \cdot (C+D) \cdot E$ is implemented using NOR circuit.

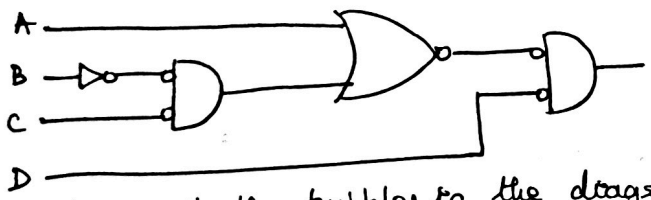
② Realize the function $F = (A+\overline{B}C) \cdot D$ using NOR Logic

Solution:

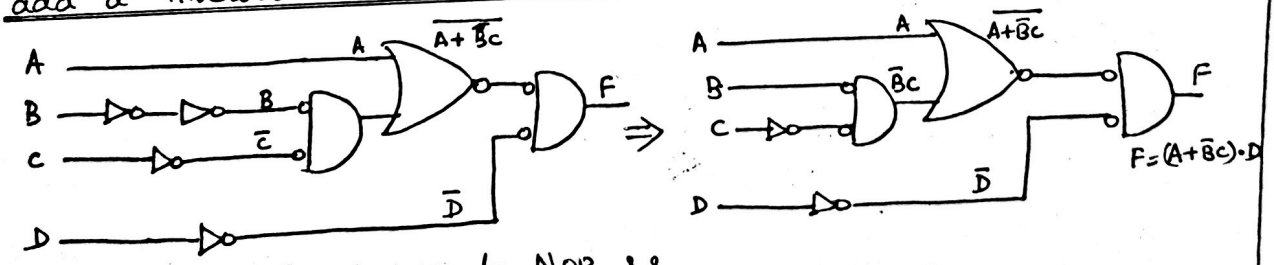
Step 1: Implement F using AND-OR-NOT Logic



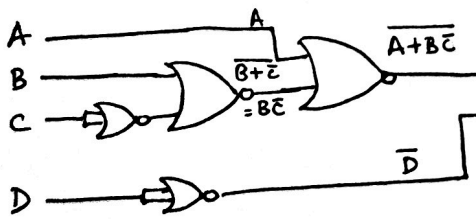
Step 2: Convert all AND gates to Invert-AND & all OR gates to NOR



Step 3: Check all the bubbles in the diagram. If any bubble is not compensated add a inverter on the same line.



Step 4: Convert all Invert-AND to NOR & all NOT to 1-input NOR gate.



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

$\therefore$ The function is implemented using NOR circuits.