DESIGN & SYNTHESIS OF COMBINATIONAL CIRCUITS

LOGIC DESIGN :-

→ The main application of switching theory is in the design of digital circuits. It is called logic design.

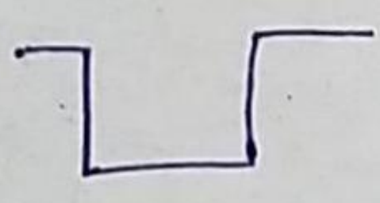
→ These circuits are designed using basic elements called gates.

→ Two things to consider for designing circuit :-

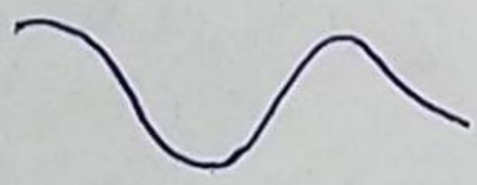
a) How fast is it ?

b) How costly is it ?

→ Digital signal → Two value (0, 1)



Analog signal → Any value



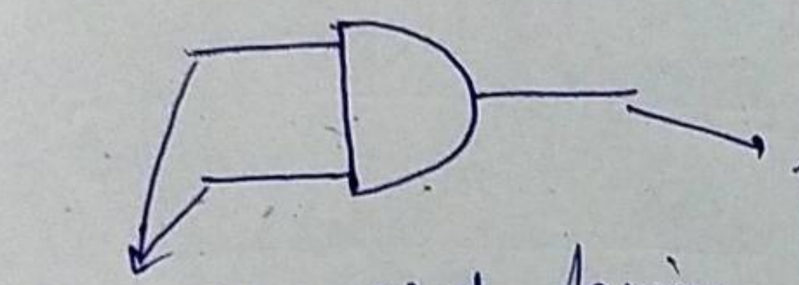
No. of gates	Name
< 10	Small scale Integration (SSI.)
10 - 100	Medium " " (MSI)
100 - 1000	Large " " (LSI)
> 1000	VLSI

AND-OR OR-AND REALIZATION

⇒ SOP can be represented using AND-OR

so it is called AND-OR realization. (NAND-NAND Realization)

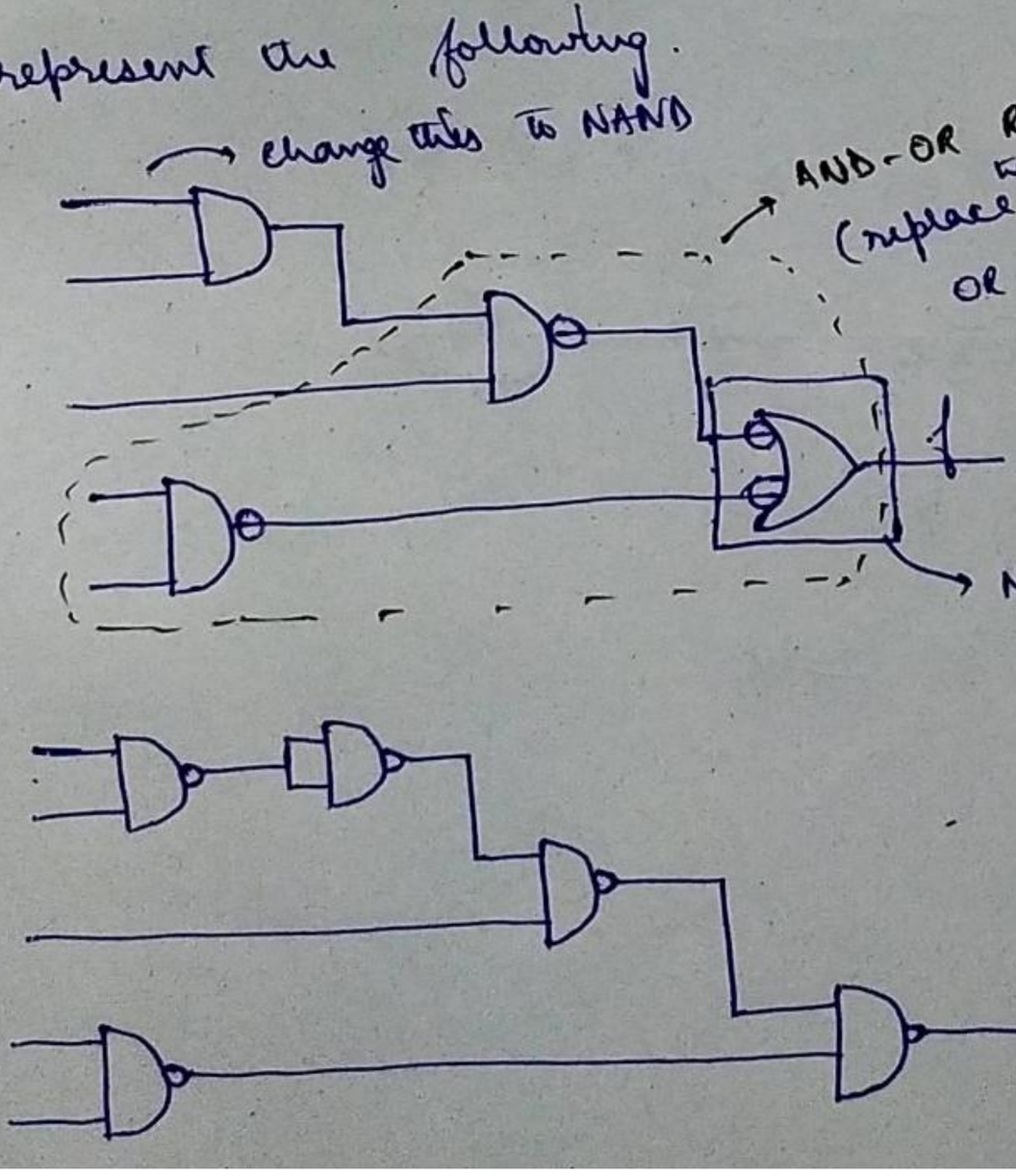
⇒ Similarly POS is called OR-AND realization.
(NOR-NOR Realization)



(*) NAND is the combination of (NOT, AND). And has
switching theory (·, −) is a functionally complete
set. Hence any thing can be designed using NAND.
Similar for NOR gate as well. So they are universal gates.

MINIMUM NO. OF NAND GATES.

● Q. Identify min. no. of two input NAND gate required to represent the following.

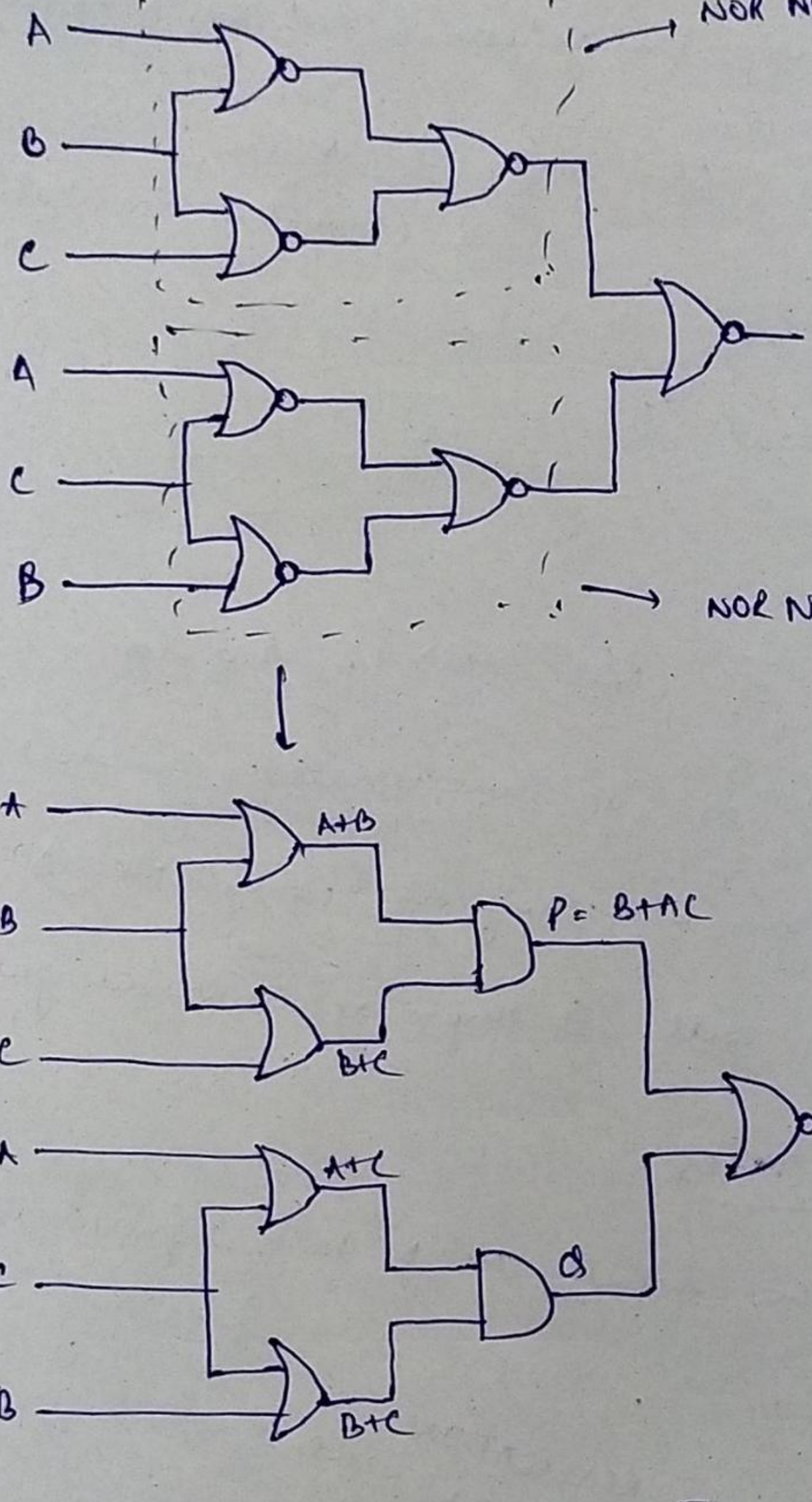


NAND = 5

NOR - NOR Realization

Q. What does the following function represent ?

→ NOR NOR can be converted to OR - AND



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

~~$= B\bar{A}C(\bar{A}+\bar{B}) = B\bar{A}C = \bar{A}BC$~~ Aw

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

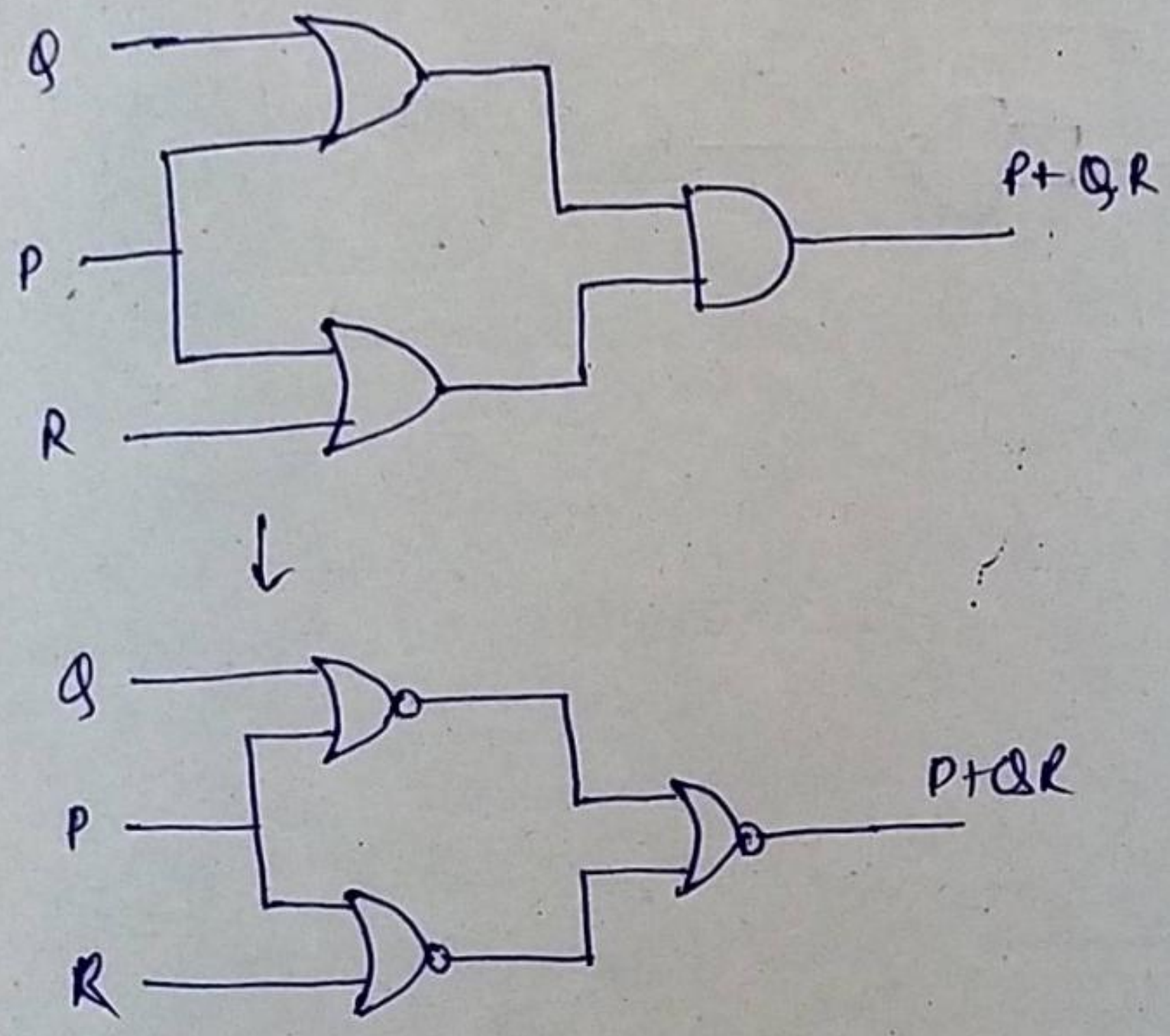
$= \overline{B\bar{C}} + \bar{A}\bar{B}\bar{C} = \bar{B}\bar{C}$ Aw.

MINIMUM NO. OF NOR GATES

Q. Find the number of NOR gates required to represent

$F(P, Q, R) = P + QR$.

→ $F(P, Q, R) = (P+Q)(P+R)$



#NOR = 3

Q. ⓔ $f = x + x\bar{y} + x\bar{y}z$ ⇒ How many NOR gates are required

→ $f = x + x\bar{y} + x\bar{y}z$

$= x(1 + \bar{y} + \bar{y}z) = x$

Only 'x' dont need any NOR gate

#NOR = 0

NOTES

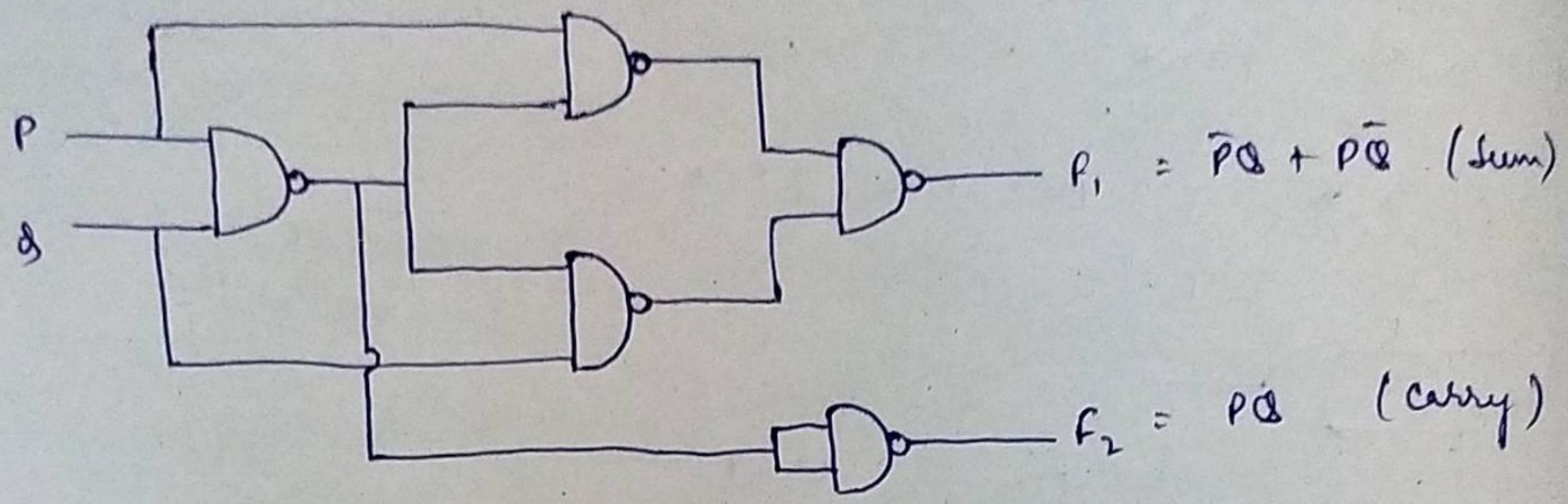
1) # NAND for Ex OR = 4

2) # NOR for Ex NOR = 4

3) # NAND for Ex NOR = 5

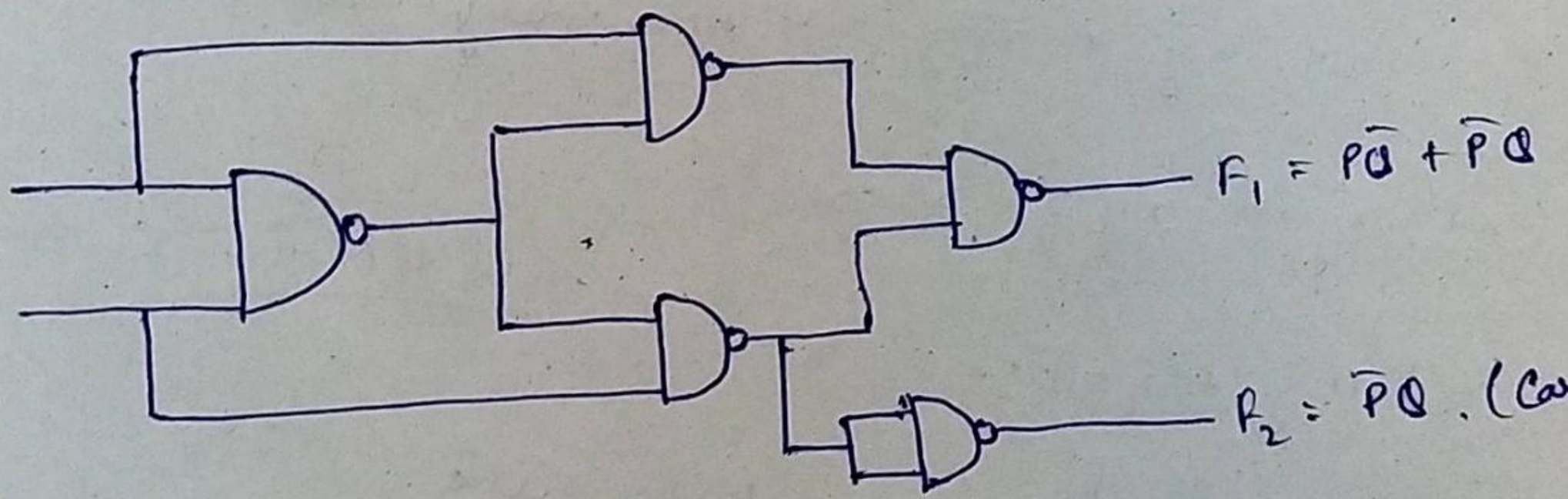
4) # NOR for Ex OR = 5

HALF ADDER



P	Q	F_1	F_2
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

HALF SUBTRACTOR



P	Q	F_1	F_2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0