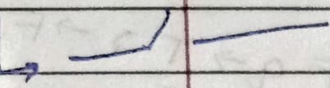
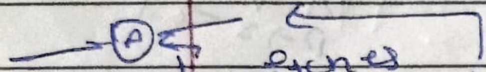
Applications: Switching Networks: → playing with switches and switches connected to two terminals.

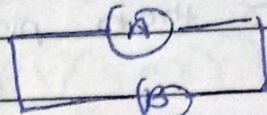
① switches in series → $A \wedge B$ or $A \cdot B$

open → no current flow
close → current flows





closed / open switches represent as

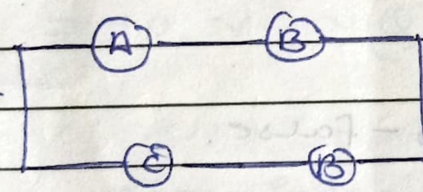


② switches connected in parallel → $A \vee B$ ~~or A+B~~

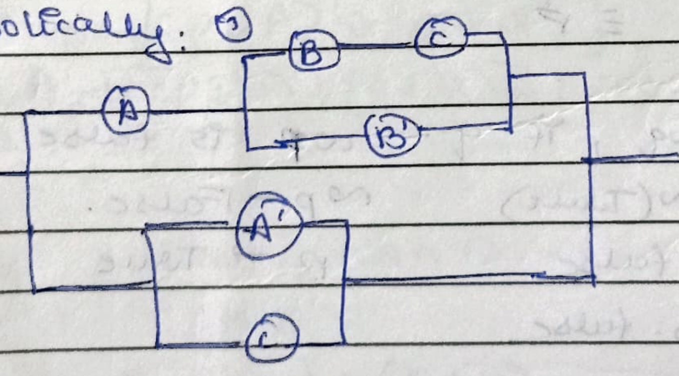
'¬', '∧', '∨' → logical
''', '·', '+' → in electricals and circuits
(not) (and) (or)

series - AND
parallel - OR

$(AB) + (CB)$ → symbolic representation of circuit

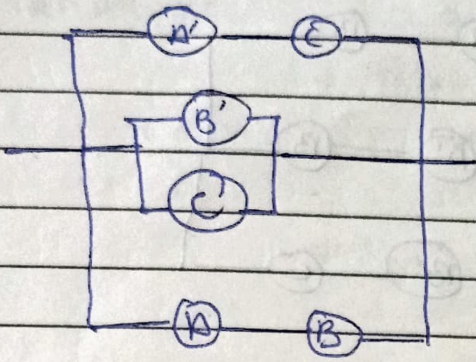
≡  → circuit.

Write symbolically: ①



$\Rightarrow [A \cdot (B \cdot C + B')] + (A' + C) \equiv \{A \wedge (B \wedge C \vee B')\} \vee [A' \vee C]$

(2)



$\Rightarrow A' \cdot C + (B' + C') + A \cdot B$

$\Rightarrow A' \wedge C \vee (B' \vee C') \vee A \wedge B$

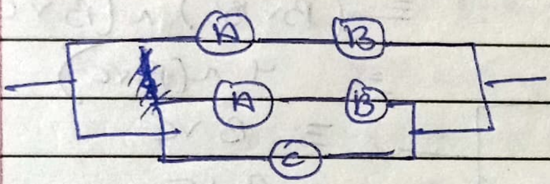
Simplify the above equation.

$(A' \wedge C) \vee (B' \vee C') \vee (A \wedge B)$

cannot be simplified.

(3) Draw a switching network by an equivalent simpler network.

(i)

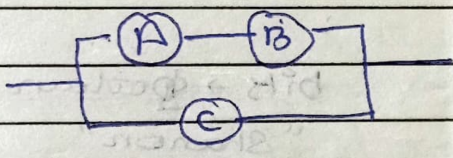


$(A \wedge B) \vee ((A \wedge B) \vee C)$

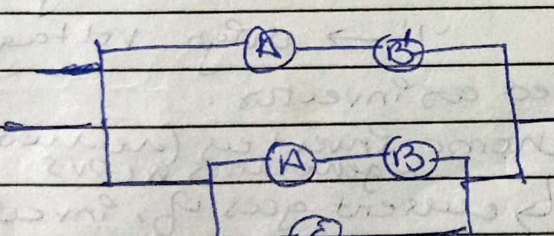
$\equiv [(A \wedge B) \vee (A \wedge B)] \vee C$

$\equiv (A \wedge B) \vee C$

$\equiv$



(ii)



$(A \wedge B') \vee ((A \wedge B) \vee C)$

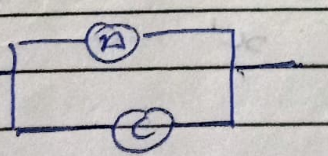
$\equiv (A \wedge B') \vee (A \wedge B) \vee C$

$\equiv A \wedge (B' \vee B) \vee C$ ∵ distributive law

$\equiv (A \wedge T) \vee C$ ∵ $p' \vee p = t$ Inverse law

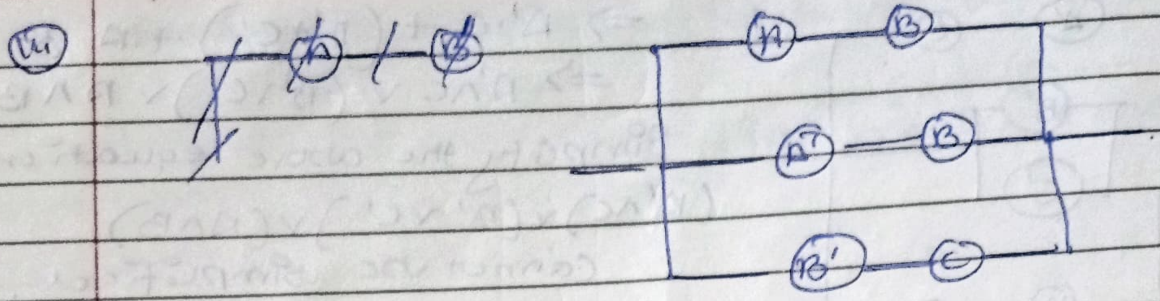
$\equiv A \vee C$

$\equiv$



"When same signs associative law holds good"

(iii)



$(A \wedge B) \vee (A' \wedge B) \vee (B' \wedge C)$

$B \wedge (A \vee A') \vee (B' \wedge C)$ ∵ Distributive law

$B \wedge T \vee (B' \wedge C)$ ∵ Inverse law

$= B \vee (B' \wedge C)$ ∵ distributive law.

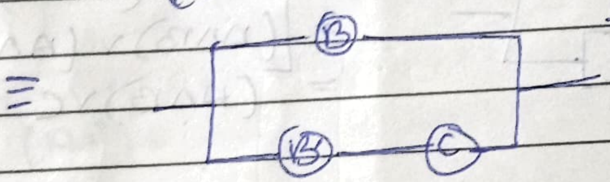
$\equiv (B \vee B') \wedge (B \vee C)$

$\equiv 1 \wedge (B \vee C)$

$\equiv B \vee C$

$= \underline{\underline{B + C}}$

$\equiv$



<u>Logic "gates"</u> :

one or many inputs but only one output.

bits → boolean algebra "Shanon"

'0' → low voltage → false

'1' → high voltage → true.

(i) NOT gate : Also called as inverter.

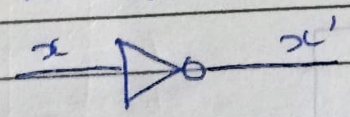
eg: home inverters (real world) or generators in UPS
↳ current goes off, inverter on & vice versa
or streetlights.

Logic table:

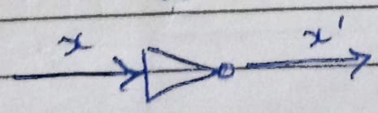
bits

x	x'
0	1
1	0.

NOT Gate :



or



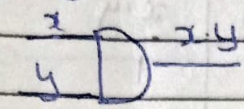
$$x' = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x = 1 \end{cases}$$

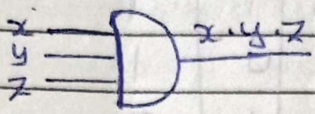
② AND gate: x and y are input signals in terms of bits.

X AND $Y \equiv xy$

Logic table:

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

gate: 



example: examination with internal and external markings.

$$xy = \begin{cases} 1 & \text{if } x = y = 1 \\ 0 & \text{otherwise.} \end{cases}$$

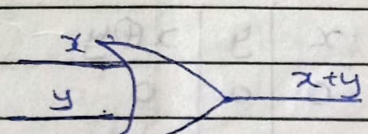
③ OR gate: x and y are input signals in terms of bits

$x+y \equiv X$ OR Y

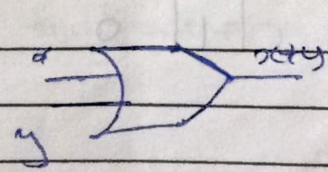
$$x+y = \begin{cases} 1 & \text{if } x = 1 \text{ or } y = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Logic table:

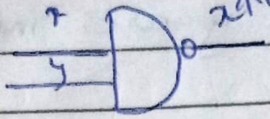
x	y	$x+y$
0	0	0
0	1	1
1	0	1
1	1	1

gate: 

or.



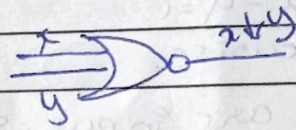
④ NAND :



$x \uparrow y = \begin{cases} 0 & \text{if } x = y = 1 \\ 1 & \text{otherwise} \end{cases}$

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

⑤ NOR

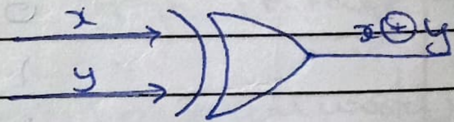


$x \downarrow y \begin{cases} \phi & \text{if } x = y = 0 \\ 0 & \text{otherwise} \end{cases}$

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

⑥ XOR : $x \oplus y$

x-oring some why

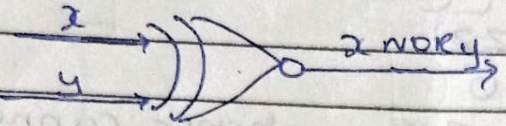


x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

$x \oplus y \begin{cases} 0 & x \text{ and } y \text{ same value} \\ 1 & \text{otherwise} \end{cases}$

(4) XNOR : Exclusive NOR Gate. ≡ Biconditional

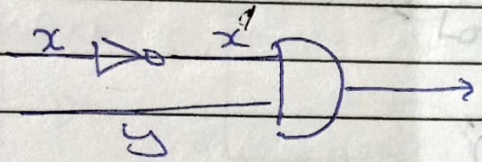
x XNOR y



x	y	x XNOR y
0	0	1
0	1	0
1	0	0
1	1	1

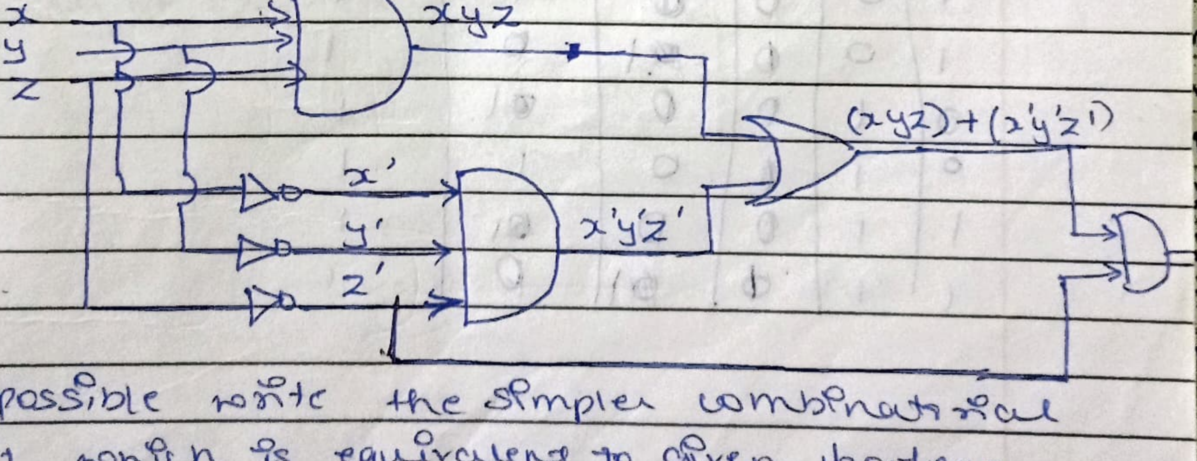
Circuits with logic gates → Combinatorial Circuit
Circuits with switches and wires → Switching circuits

Combinatorial Circuit:



$x'y$ — Boolean expression.

(5) $[(xyz) + (x'y'z')]$ ~~xy~~ z' (Actual question $[(xyz) + (x'y'z)]z'$.)



If possible write the simpler combinatorial circuit which is equivalent to given boolean expression.

→

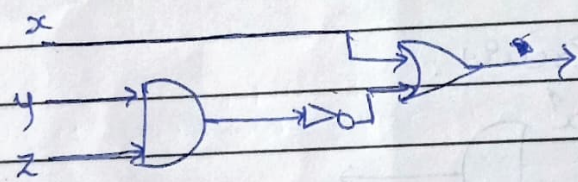
1.1

Boolean function,

$$f(xyz) = [(xyz) + (x'y'z)]z'$$
$$= [z(xy + x'y')]z'$$
$$= zz'$$
$$= 0$$

got contradiction hence cannot simplify and go ahead with the circuit.

Find the output of the combinational circuit, also construct a logic table.

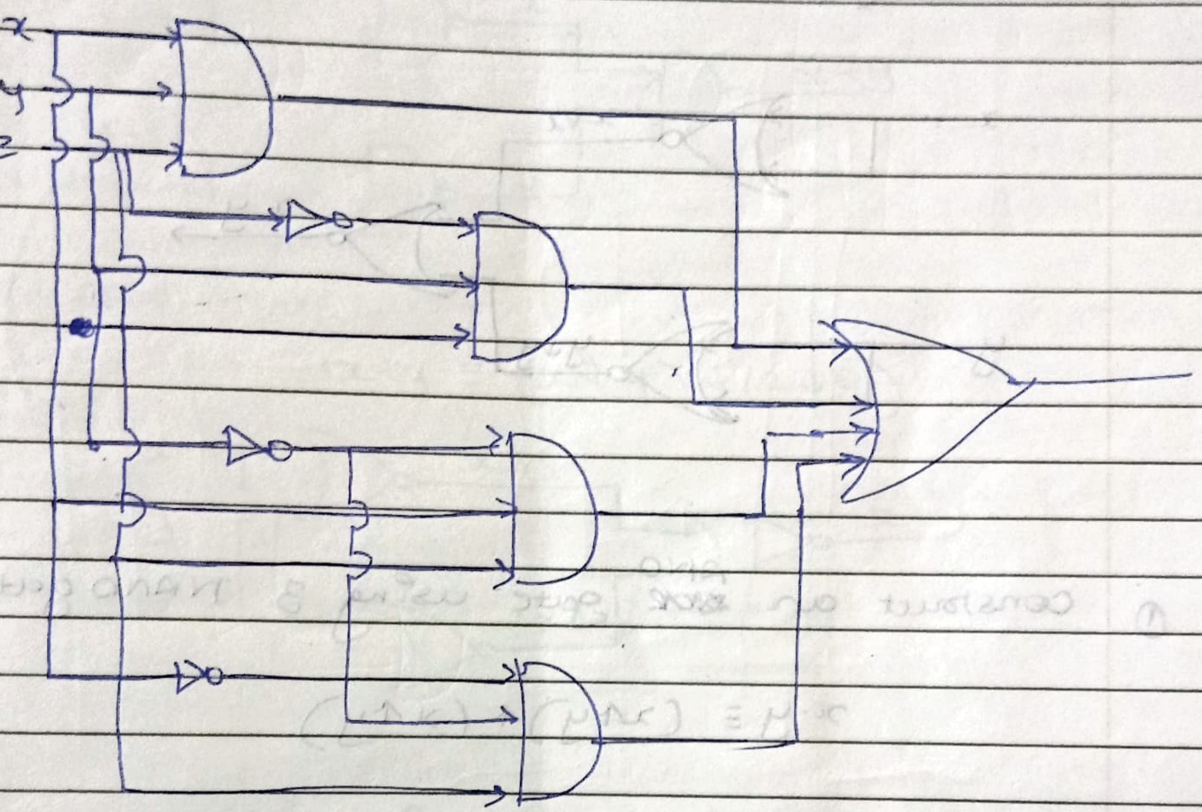


$x + (y \cdot z)'$

y	x	z	yz	(yz)'	x+(yz)'
0	0	0	0	1	1
0	0	1	0	1	1
1	0	0	0	1	1
1	0	1	1	0	1
0	1	0	0	1	1
0	1	1	0	1	1
1	1	0	0	1	1
1	1	1	1	0	1

DOMS Page No. Date / /

Find the output produced by the combinational circuit also find the simple cs equivalent to it



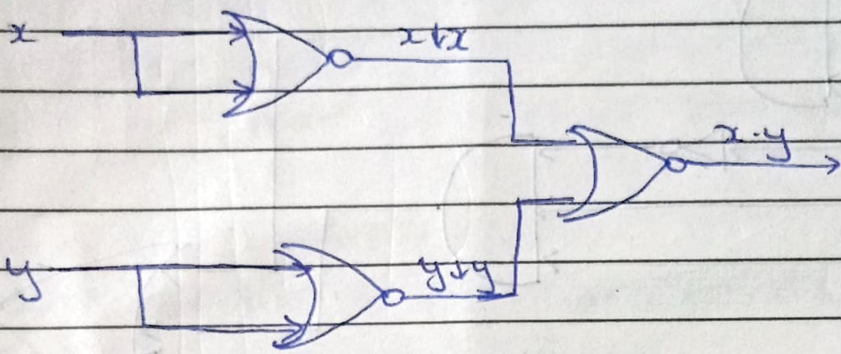
$f(xyz) = (xyz) + (xyz') + (xy'z) + (x'y'z)$ $= xy + y'z$

$= xy(z+z') + (xy' + x'y')z \; (y'z(x+x'))$

$= xy + (xy' + x'y)z \; (\; y'z$

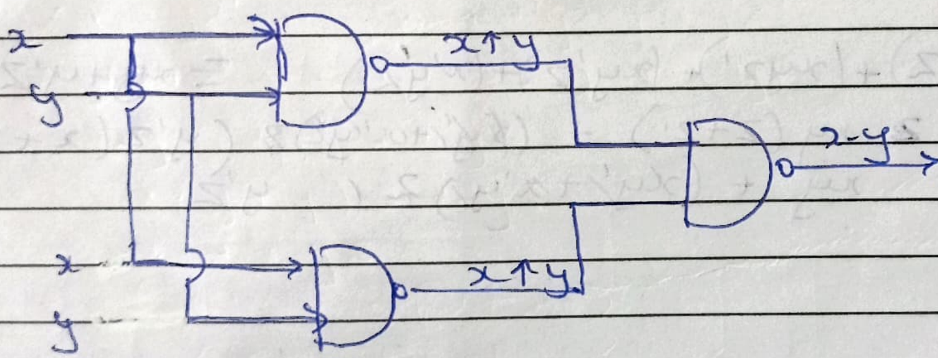
Construct an AND gate using 3 NOR gates.

$x \cdot y \equiv (x \downarrow x) \downarrow (y \downarrow y)$ $\quad xy = \overline{x + y}$

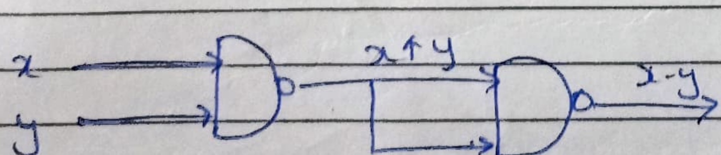


① Construct an ~~NOR~~ AND gate using 3 NAND gate

$x \cdot y \equiv (x \uparrow y) \uparrow (x \uparrow y)$

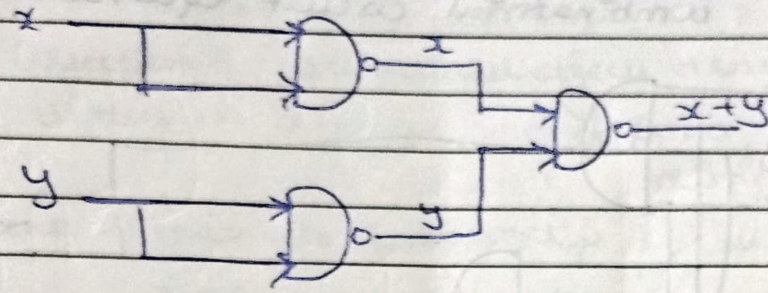


④ AND by only 2 NAND gates

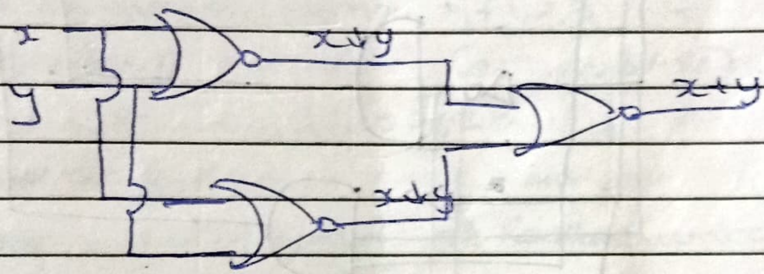


Construct OR gate using 3 NAND gate

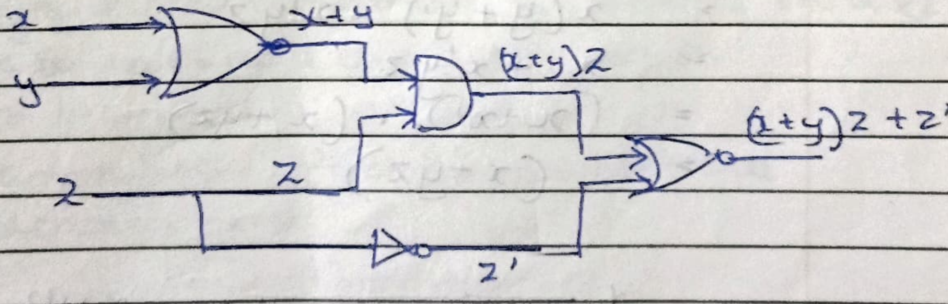
i) $x+y \equiv (x\uparrow x)\uparrow(y\uparrow y)$



ii) $x+y \equiv (x\downarrow y)\downarrow(x\downarrow y)$

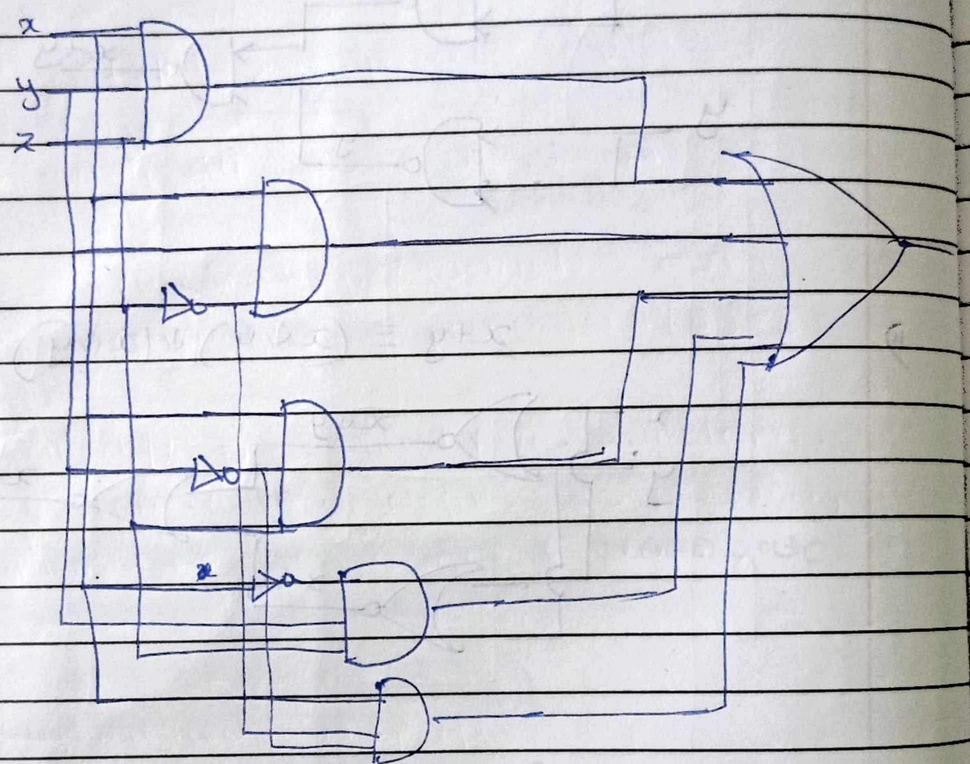


make a combinational circuit that yields the boolean expression $(x+y)z+z'$ as its output



Construct a combinatorial circuit for the boolean expression $xyz + xyz' + xy'z + x'yz + xy'z'$. Find a simpler combinatorial circuit equivalent to it.

Soln:



$$f(x,y,z) = xyz + xyz' + xy'z + x'yz + xy'z'$$
$$= xy(z+z') + xy'(z+z') + x'yz$$
$$= xy + xy' + x'yz$$
$$= x(y+y') + x'yz$$
$$= x + x'yz$$
$$= (x+x')\cdot(x+yz)$$
$$= (x+yz)$$

$\therefore$ equivalent combinatorial circuit is