PRACTICAL 1A: LOGIC GATES AND BOOLEAN LAWS

Ronak Mehta (MHTRON001)

Plagiarism Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.

2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this practical report from the work(s) of other people has been attributed and has been cited and referenced.

3. This practical report (including circuit diagrams and code) is my own work.

4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work.

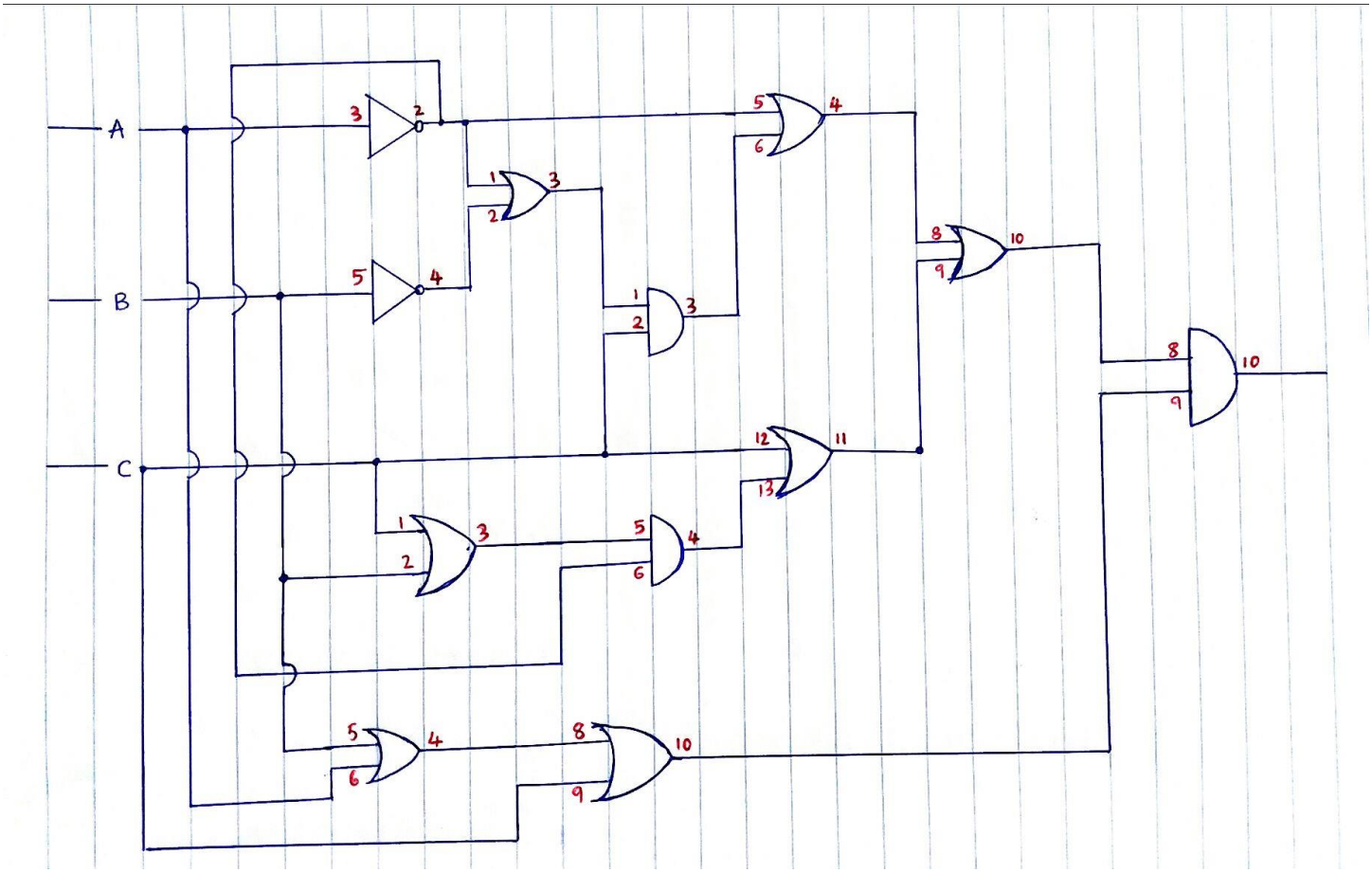
5. I acknowledge that copying someone else's code, schematics or report, or part of it, is wrong, and declare that this is my own work.

16-03-2018

Signature Date

Solutions

a) Y = (A + B + C) . ($\bar{A}$ + C + $\bar{A}$.(B+C) + ($\bar{A}$ + !B).C)



b) Total of 14 Logic Gates are required.

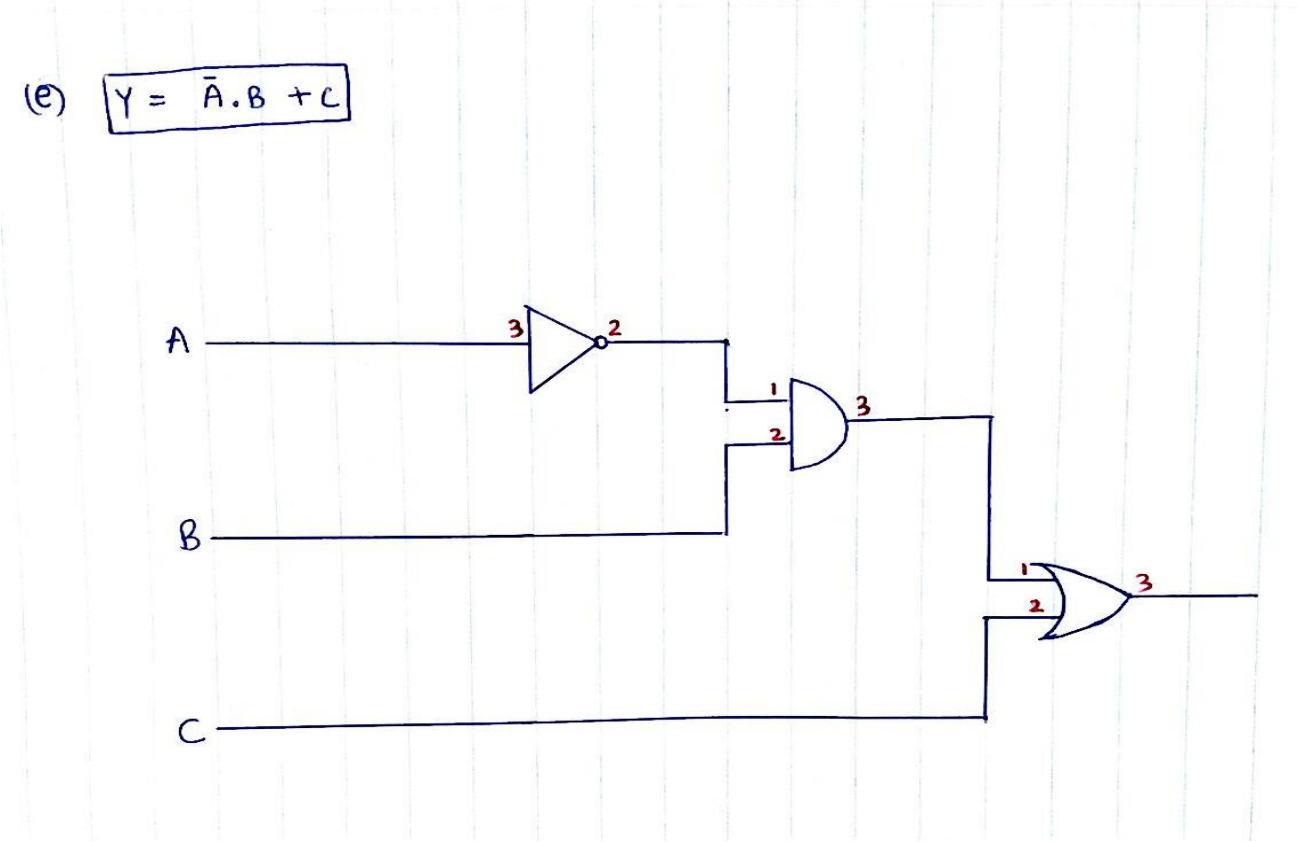
- 7 OR Gates
- 3 AND Gates
- 4 NOT Gates

c) It will need Four 4000 series logic chips.

- 2 Quad 2-input OR-gate (4071)
- 1 Quad 2-input AND-gate (4081)
- 1 Hex Buffer Inverter (4049)

d) $Y = (A + B + C) . (\bar{A} + C + \bar{A}.(B+C) + (\bar{A} + !B).C)$
$= (A + B + C) . (\bar{A} (1 + (B+C)) + C (1 + (\bar{A} + !B)))$ *// Took common factors out ($\bar{A}$ and C)*
$= (A + B + C) . (\bar{A} (1) + C (1))$ *// TRUE value (1) OR-ed (+) with any variable always results in TRUE(1)*
$= (A + B + C) . (\bar{A} + C)$ *// TRUE value AND-ed (.) with a variable always results in that variable*
$= \bar{A}.A + \bar{A}.B + \bar{A}.C + A.C + B.C + C.C$ *//Open Brackets*
$= \bar{A}.A + \bar{A}.B + \bar{A}.C + A.C + B.C + C$ *// C.C = C (I.e.: TRUE.TRUE = TRUE and FALSE.FALSE = FALSE)*
$= 0 + \bar{A}.B + C (\bar{A} + A + B + 1)$ *// Took C as a common factor*
$= 0 + \bar{A}.B + C (1)$ *// TRUE value (1) OR-ed(+) with any variable always results in TRUE(1)*
$Y = \bar{A}.B + C$

e)



g) Truth Table for the Boolean Expression: $Y = \bar{A}.B + C$

A	B	C	$\bar{A}$	$\bar{A} . B$	$Y = \bar{A}.B + C$	From Practical
0	0	0	1	0	0	0
0	0	1	1	0	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	0	0	0
1	1	1	0	0	1	1