NATIONAL INSTITUTE OF TECHNOLOGY

AGARTALA, TRIPURA(WEST)

NAME – Arghajit Bhomik

ENROLLMENT NO - 20UCS134

REGISTRATION NO - 2012724

SUBJECT – DCLD LAB

BRANCH – Computer Science and Engineering

SECTION - B

1 EXPT NO. 4:

1.1 STUDY OF BOOLEAN EXPRESSION SIMPLIFICATION

Objective:

To study the Boolean rules & Boolean expression simplification Equipments: Logic circuit simulator pro application

Theory

Here is the list of rules used for the Boolean expression simplifications.

• ***The Idempotent Laws:***

Idempotent Law states that combining a quantity with itself either by logical addition or logical multiplication will result in a logical sum or product that is t he equivalent of the quantity.

A.A = A A + A = A

• ***The Associative Laws:***

This law allows the removal of brackets from an expression and regrouping of the variables

(A.B).C = A.(B.C) (A + B) + C = A + (B + C)

• ***The Commutative Laws:***

The order of application of two separate terms is not important.

A.B = B.A A + B = B + A

• ***The Distributive Laws:***

This law permits the multiplying or factoring out of an expression.

A.(B + C) = A.B + A.C A + B.C = (A + B).(A + C)

• ***The Identity Laws:***

A term OR’ed with a “0” or AND’ed with a “1” will always equal that term.

A.F = F A.T = A A + F = A A + T = T

• ***The Complement Laws:***

A term AND’ed with its complement equals “0” and a term OR’ed with its complement equals “1”.

A.A' = F Q.A + A' = 1

• ***The Involution Law:***

A = A

De-Morgan’s Law:

1. Two separate terms NOR’ed together is the same as the two terms inverted (Complement) and AND’ed for example (AB)’ = A’ + B’

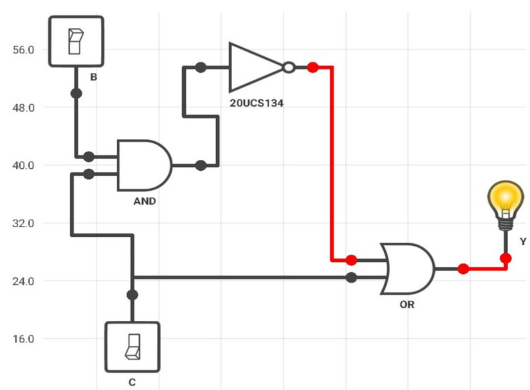
2. Two separate terms NAND’ed together is the same as the two terms I nverted (Complement) and OR’ed for example (A + B)’ = A’B’

(T – True – 1 and F - False – 0)

2 Simplify: C + (B.C)’:

Expression	Rule(s) Used
C + (B.C)'	Original Expression.
C+ (B' + C')	DeMorgan's Law.
(C + C') + B'	Commutative, Associative Laws.
T + B'	Complement Law.
T	Identity Law.

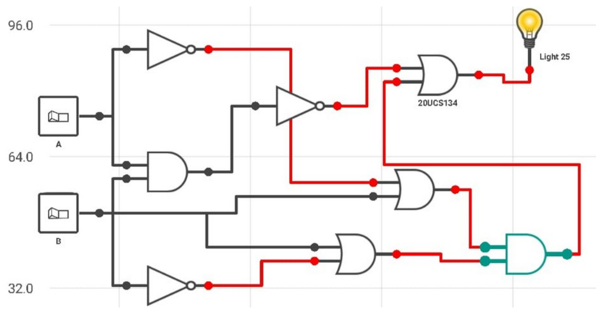
Fig-



3 Simplify (A.B)’ + (A’ + B).(B’ + B):

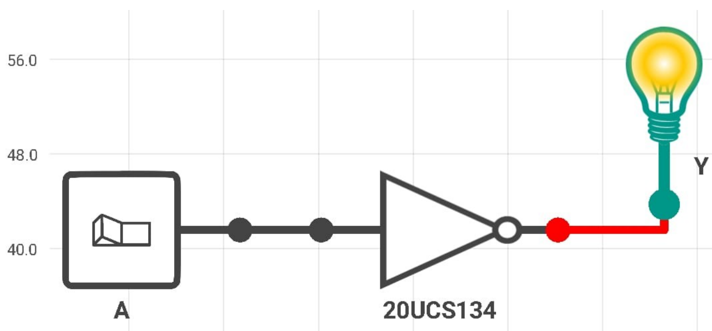
Expression	Rule(s) Used
(A.B)' + (A' + B).(B' + B)	Original Expression

Fig-



(A.B)'.(A' + B)	Complement law, Identity law
(A' + B').(A' + B)	DeMorgan's Law
A' + B'.B	Distributive law.
A'	Complement. Identity.

Fig-

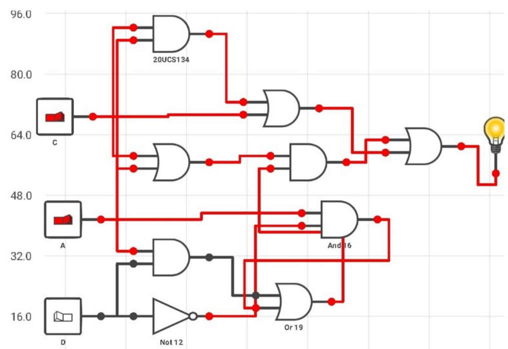


4 Simplify: (A + C).(A.D + A.D’) + A.C + C:

1 Expression   Rule(s) Used

(A + C).(A.D + A.D') + A.C + C   Original Expression

Fig-



(A + C).A.(D + D') + A.C + C   Distributive.

(A + C).A + A.C + C   Complement, Identity.

A((A + C) + C) + C   Commutative. Distributive.

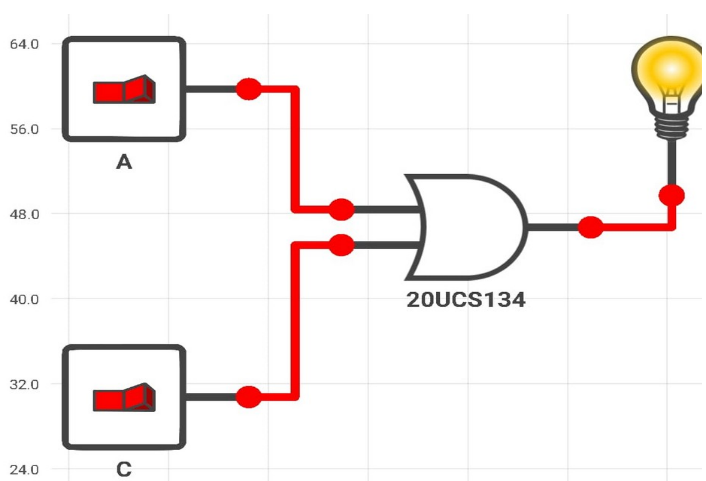
A.(A + C) + C   Associative. Idempotent

A.A + A.C + C   Distributive

A + (A + T).C   Idempotent, Identity, Distributive.

A + C   Identity, twice.

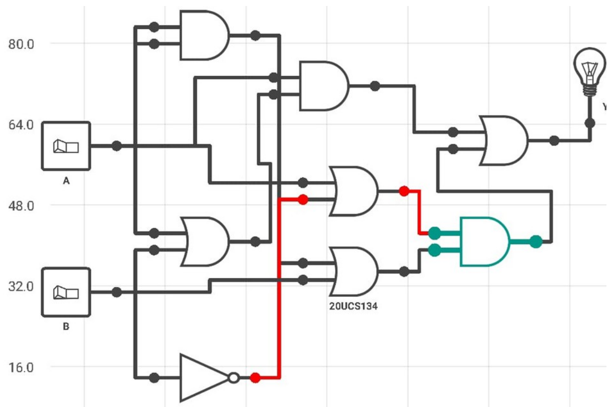
Fig-



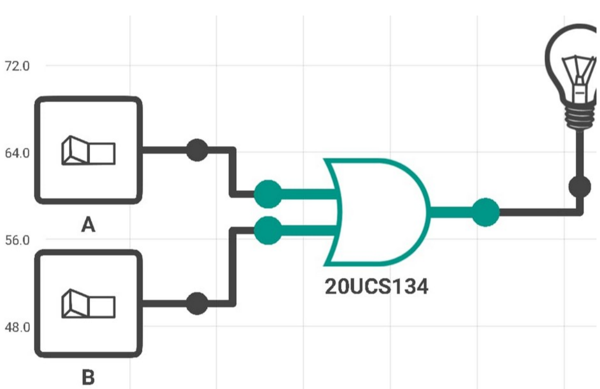
5 Simplify: A.(A + B) + (B + A.A).(A + B’):

Expression	Rule(s) Used
A'.(A + B) + (B + A.A).(A + B')	Original Expression

Fig-



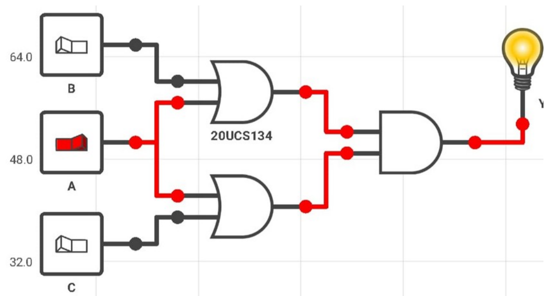
Expression	Rule(s) Used
A'.A + A'B+ (B + A).A + (B + A).B’	Idempotent (A.A to A),
A'.B + (B + A).A + (B + A).B'	Complement, then Identity.
A'.B + B.A + A.A + B.B'+ AB'	Distributive, two places
A'.B + B.A + A + AB'	Idempotent (for the A's)
A'.B + A.B + A.T + A.B'	Commutative, Identity
A'.B + A.(B + T + B')	Distributive.
A'.B + A	Identity
A + A'.B	Commutative.
(A + A').(A + B)	Distributive.
A + B	Complement, Identity



6 Simplify: (A+B).(A+C)

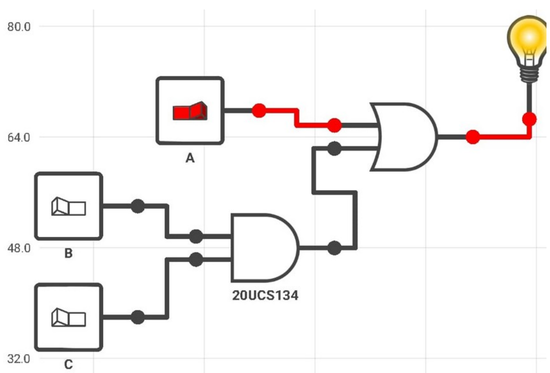
Expression	Rule(s) Used
A.A + A.C + A.B + B.C	Distributive law

Fig-



A + A.C + A.B + B.C	Idempotent AND law (A.A = A)
A.(1 + C) + A.B + B.C	Distributive law
A.1 + A.B + B.C	Identity OR law (1+C = 1)
A.(1 + B) + B.C	Distributive law
A.1 + B.C	Identity OR law (1+ B = 1)
A + (B.C)	Identity AND law (A.1 = A)

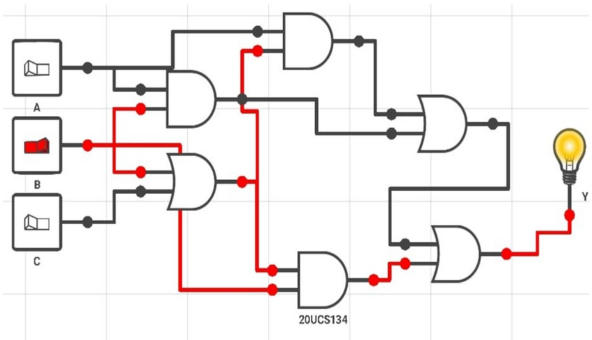
Fig-



7 Simplify: A.B + A.(B + C) + B.(B + C)

Expression	Rule(s) Used
A.B + A.B + A.C + B.B + B.C	Distributive law

Fig-



A.B + A.B + A.C + B + B.C	Idempotent law
A.B + A.C + B + B.C	Idempotent law
A.B + A.C + B	Absorption law
B + A.C	Absorption law

Fig-