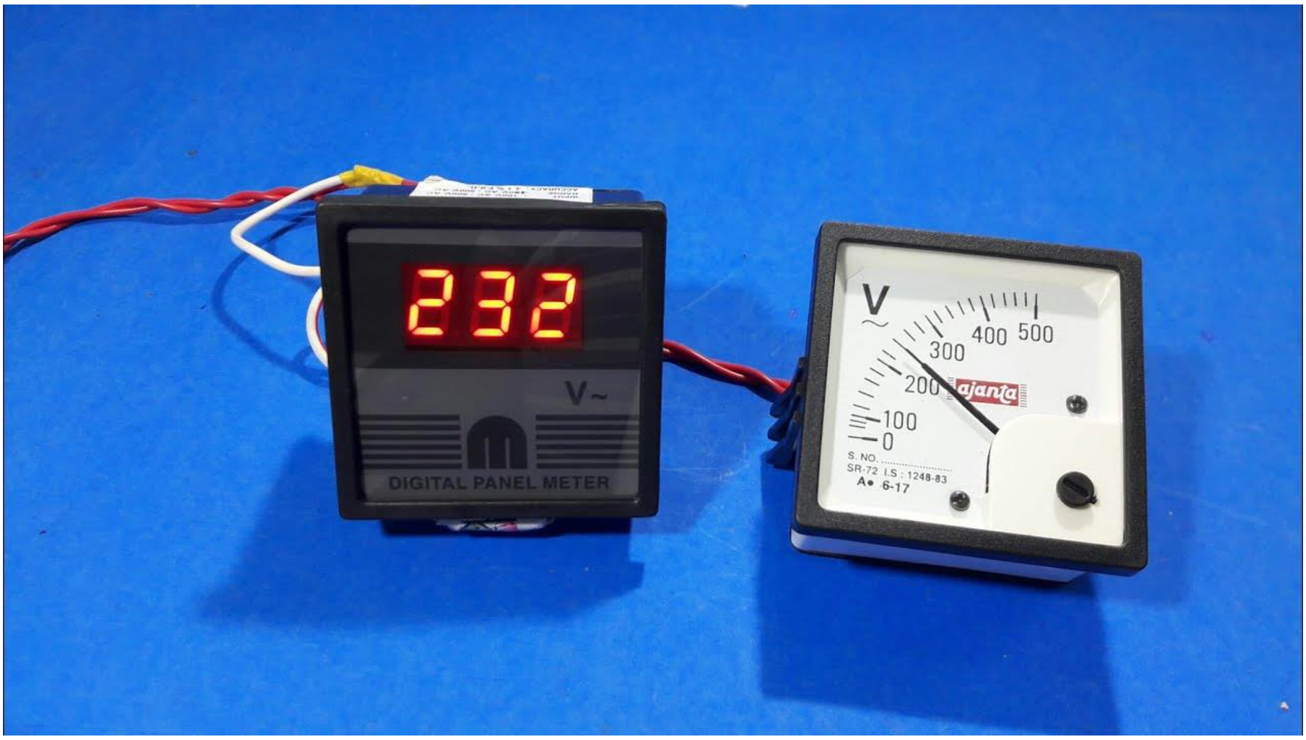
Computer Logic

Digital vs Analogue Computers

During and after the second world war, when the first electronic computers were being built, some engineers opted for digital computers while others worked on analogue computers.

In analogue computers a value is represented by an amount of some physical feature e.g. voltage, so if 1 volt represents the number 3 then 2 volts represent the number 6, 2.5 volts represent 7.5 etc.

In digital computers a value is represented by symbols.



The picture above shows two voltmeters. The one on the left is digital (the voltage is shown with symbols) and the one on the right is analogue (the voltage is shown by means of the amount the hand turns clockwise).

Engineers finally found out that digital computers are better than analogue ones. A digital computer is more reliable i.e. it makes less errors.

The Bit

A **bit** is short for binary digit. It's a single unit of information with a value of either 0 or 1 (off or on, false or true, low or high). By using sequences of bits, we can represent any possible value.

This is how the numbers from zero to ten are represented by means of bits.

Zero	0	one bit is enough
One	1	one bit is enough
Two	10	two bits are needed
Three	11	two bits are needed
Four	100	three bits are needed
Five	101	three bits are needed
Six	110	three bits are needed
Seven	111	three bits are needed
Eight	1000	four bits are needed
Nine	1001	four bits are needed
Ten	1010	four bits are needed

Basic Logic Operators

The basic logical operators are AND, OR and NOT.

X	Y	X AND Y
T	T	T
T	F	F
F	T	F
F	F	F

In the above table T stands for True and F for False. It shows the output for all possible inputs. In the computer environment T and F are changed respectively to 1 and 0 as shown below. This table is called the **truth table**.

X	Y	X AND Y
1	1	1
1	0	0
0	1	0
0	0	0

The **AND** operator is also represented by a dot and this dot is sometimes left out too for example X AND Y can be represented by **X.Y** or even **XY**.

X	Y	X OR Y
1	1	1
1	0	1
0	1	1
0	0	0

The **OR** operator is also represented by a + for example X OR Y can be represented by **X + Y**.

X	NOT X
1	0
0	1

NOT X can be written as $\overline{\mathbf{X}}$ or **X’**.

Another truth table which may become handy at times is the XOR table. XOR stands for Exclusive OR

X	Y	X XOR Y
1	1	0
1	0	1
0	1	1
0	0	0

The logical operators AND, OR, NOT and XOR (and even others that are not covered here) are also called **Boolean** operators. This is because the person who invented these tables was George Boole, an English school teacher who was born in 1815.

Logical (Boolean) Expressions

The Boolean operators can be joined together to form logical expressions, very much like arithmetic expressions e.g. 3+4-6.3*5. Below, we have four Boolean expressions.

$$\mathrm{A} + \overline{\mathrm{B}}$$

$$\overline{\mathrm{AB}} + \mathrm{C}$$

$$\mathrm{A}(\mathrm{B} + \overline{C})$$

$$(\overline{\mathrm{A}} + \mathrm{B})(\mathrm{A} + \overline{\mathrm{C}})$$

A, B and C can only have values equal to 0 or 1 and the expression will give a result of either 0 or 1.

Example: Let A=0 and B=1. What is the value of $\bar{A} + B$? This is equal to NOT 0 OR 1 which is equal to 1 OR 1 which is equal to 1.

Precedence of Operators

When more than one logical operator is used in a statement, NOT is evaluated first, then AND, and finally OR. This is similar to the BIDMAS rule in arithmetic. The brackets, as in the BIDMAS rule are worked out first. Example: Let A=1, B=0 and C=1.

$$A + \overline{B}$$

evaluates to A OR NOT B =

1 OR NOT 0 =

1 OR 1 =

1

$$\overline{AB} + C$$

evaluates to NOT (A AND B) OR C =

NOT (1 AND 0) OR 1 =

NOT (0) OR 1 =

1 OR 1 =

1

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

evaluates to A AND (B OR NOT C) =

1 AND (0 OR NOT 1) =

1 AND (0 OR 0) =

1 AND 0 =

0

<u>Question 1</u>: Evaluate the following Boolean expressions given that A=0, B=1 and C=1.

a) $\overline{A} + C$

b) $A + \overline{B}.\overline{C}$

c) $A(\overline{B} + \overline{AC})$

d) $(A + \overline{B})(A + \overline{C})$

Truth Table of a Boolean Expression

Let us write the truth table of $A + \overline{B}.\overline{C}$. It is the following:

INPUT						OUTPUT
A	B	C	B'	C'	B'C'	A+B'C'
0	0	0	1	1	1	1
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	1	0	0	1
1	1	0	0	1	0	1
1	1	1	0	0	0	1

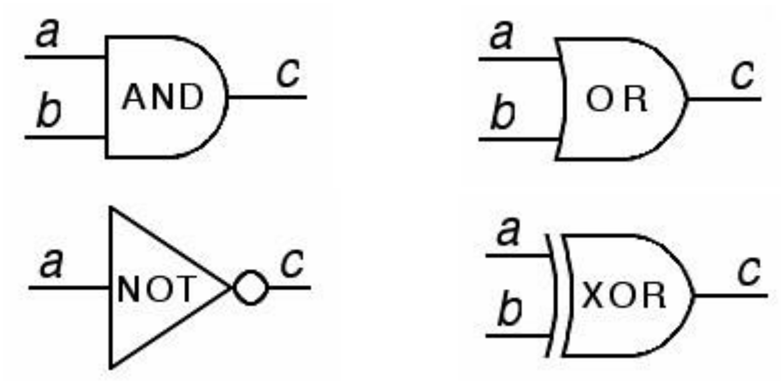
Equivalent Expressions

How can we show that two expressions are equivalent? We do this by constructing the truth tables of both expressions and then check that the output is equivalent.

Logic Gates

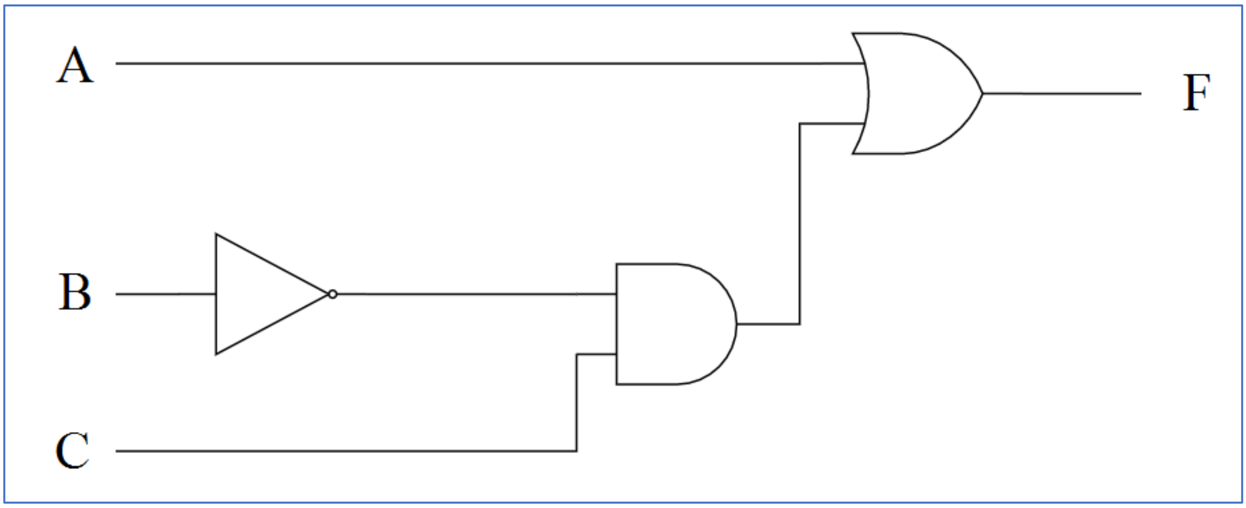
A logic gate is an elementary building block of a digital circuit. Most logic gates have two inputs and one output. At any given moment, every terminal is in one of the two binary conditions low (0) or high (1), implemented by different voltage levels. In most logic gates, the low state is approximately zero volts (0 V), while the high state is approximately five volts positive (+5 V).

The following diagram shows the logic gates of the operators covered above.

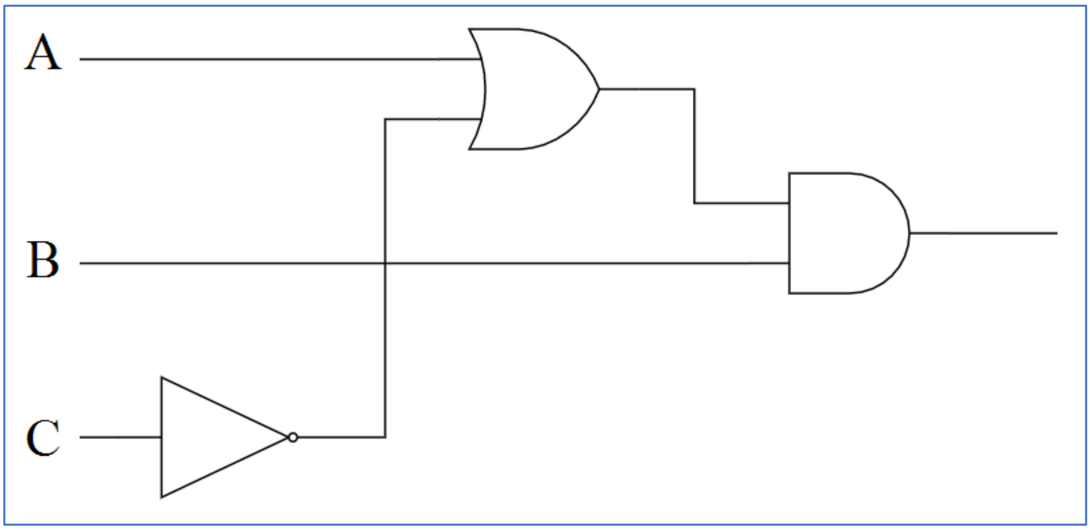


Logic Circuits

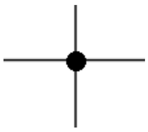
Boolean expressions can be expressed as logic circuits. For example, the following logic circuit implements the expression $A + \bar{B}.C$ (in the diagram the output is called F).

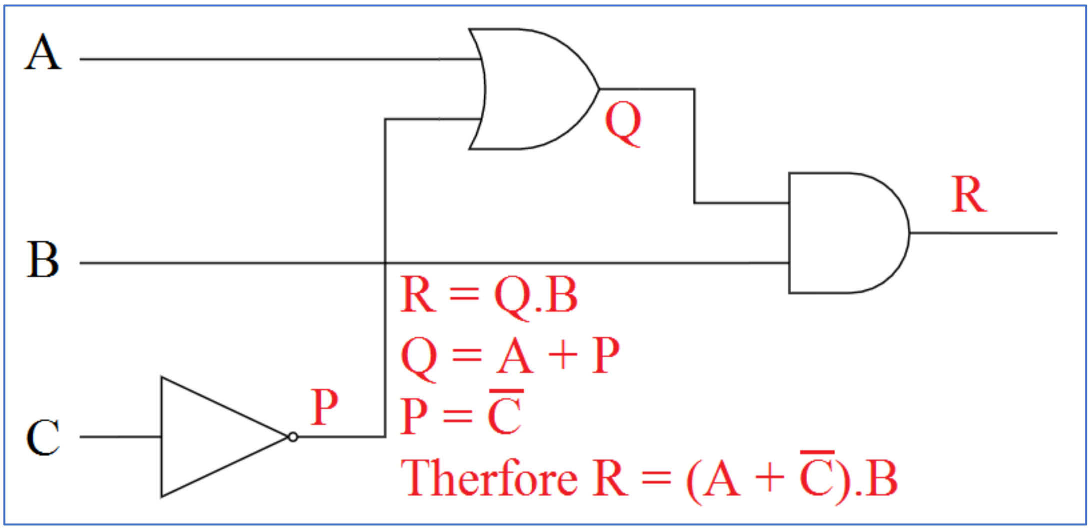


How can we deduce the Boolean expression from a logic circuit? Let us consider the following circuit.



Note that the lines crossing each other in the above diagram are not meant to touch each other. When lines are meant to touch each other, they are drawn as shown below.





Question 2:

a) Draw the logic circuits of the Boolean expressions found in question 1.
b) Draw the truth tables of the same expressions.
c) Prove the following identities by showing that the truth table of the left-hand expression is identical to the truth table of the right-hand expression.
   i $AB + A\overline{B} \equiv A + \overline{B}$
   ii $(P + \overline{Q})(P + Q) \equiv P$
d) What is the Boolean equation of the following logic circuit?