

Logic Gates with Boolean Functions

In this chapter you will learn about,

- signals used in electronic science
- basic logic gates and combinational logic gates
- representing Boolean expressions using truth tables
- creating combinational logic gates based on basic logic gates
- drawing digital circuits for Boolean expressions
- Constructing truth tables after writing Boolean expressions for digital circuits
- integrated Circuits
- practical usage of logic gates

4.1 Introduction

Communication happens in various ways among living beings in day to day life. Drumming was used in the past, as a method of communication. Further, a railway guard in a station blows a horn waving a green flag to signal that a train is going to leave. A green light should be there for the train to start and if it is a red light, the signal is to stop.

Usually, two key are used to open a safe and both keys are needed to open it. Before you start a journey in a car, all its doors should be closed properly. If at least one door is not closed properly, a light will be on or there will be a sound for the driver to signal it. When all the doors are closed, this signal will be stopped. Further, you should wear seat belts if you are seated in the front seats of a car. Otherwise there will be a signal to indicate this. Thus, as we use signals in our day to day life to make decisions, the computer also uses signals.

4.2 Logic Gates

Circuits which enable building of certain logical conditions using binary values and which enable making certain decisions, are called Logic Circuits. The Computer is made of a large number of complex digital circuits. These electronic circuits are designed as required connecting a large number of basic logical circuits called logic gates.

Central Processing Unit is made up of a collection of a large number of logic gates.

The following Figure 4.1 shows a circuit made up of basic logic gates; AND, OR, and NOT.

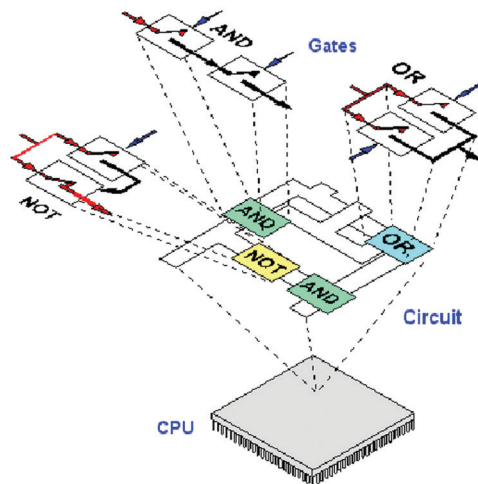


Figure 4.1 – Circuit with Basic Logic Gates

The function carried out by a logic gate is giving an output considering an input or several inputs.

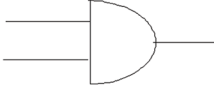

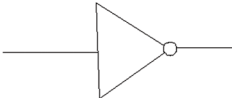
There are numerous of technical methods to produce logic gates and its internal circuit consists of devices such as transistors, diodes and resistors.

According to the way that the circuits are used, logic gates can be classified, into two types.

1. Basic Logic Gates
2. Combinational Logic Gates

4.3 Basic Logic Gates

There are three types of basic logic gates. Those are,

1. AND gate 
2. OR gate 
3. NOT gate 

4.3.1 AND Gate

Let us consider the example given below to understand AND logic.

- If the door of your computer lab is locked with a key and padlock, both door key and padlock key are needed to open that door. If both keys are there, you can open the door. If either door key or padlock key is used, you cannot open the door. Further, if keys are not there, you cannot open the door.
- Let us consider the simple series electronic circuit given below to understand AND operation.

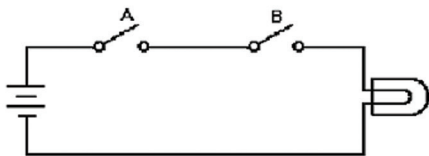


Figure 4.2 – Electronic circuit when input A and B value is 0.

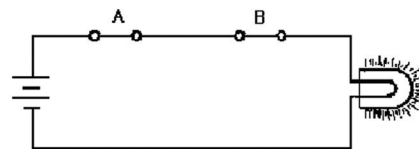


Figure 4.3 – Electronic circuit when input A and B value is 1.

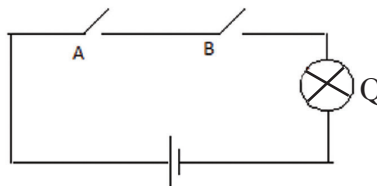


Figure 4.4 - Electronic circuit for AND logic gate.

Here, the two switches A and B, bulb Q and two batteries are connected in series. Let us consider the two switches as input and the bulb as output.

When both switches A and B are closed only, the bulb will be switched on. The bulb will not be switched on when either switch A or B is closed. When both switches are open, the bulb will not be switched on.

If Logic 1 indicates both switches are closed and the bulb is switched on (Figure 4.2) and if Logic 0 indicates when one switch is open and the bulb is switched off (Figure 4.3), the relationships between AND logic gate inputs and outputs are given below. (Table 4.1)

Table 4.1 – Nature of the bulb in the circuit for AND

| A | B | Q |
|--------|--------|--------------|
| Open | Open | Switched off |
| Open | Closed | Switched off |
| Closed | Open | Switched off |
| Closed | Closed | Switched on |

Table 4.2 – Truth Table for AND Logic Gate

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Here, $Q=1$ when both A and B input are in "1" state only. When either inputs are in "0" state $Q = 0$. This table is called the truth table for AND logic gate.

Observation



If at least one input is in "0" state, the output of AND gate is "0".

Four states ($2^2 = 4$) are shown in the table as this gate consists of two inputs.

Logic of AND gate is "A" AND "B". According to Boolean expressions, it is represented as $A.B$

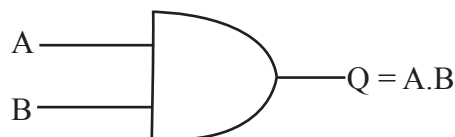


Figure 4.5 - The Boolean algebra expression and symbol for AND logic gate.

This AND logic gate consists of a minimum of two inputs and there are AND logic gates with more than two inputs.

Activity



1. Given below is an AND logic gate with three inputs; A, B and C. (Figure 4.6)

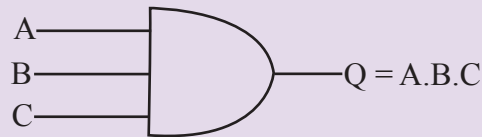


Figure 4.6 - AND logic gate with three inputs.

The above AND logic gate is equivalent to the circuit given below. (Figure 4.7)

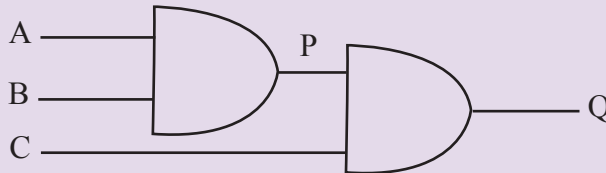


Figure 4.7 - Electronic circuit for AND logic gate with three inputs

- I. Write down P which is the output of the AND logic gate for A and B inputs.
 - II. Write down Q which is the output of AND logic gate for P and C inputs.
 - III. The value of Q is equivalent to the output of AND logic gate with three inputs A, B and C.
Hence, the Boolean expression for this circuit is,
$$Q = A.B.C$$

There are 8 states in the Truth Table related to the above circuit. As there are 3 inputs, there are 8 states in the Truth Table. ($2^3 = 8$)
2. Fill in the table given below. Fill in column A.B column using the truth table for AND logic gate with 2 inputs. (Table 4.3) Fill in A.B.C subsequently.

Table 4.3 - Truth Table for AND logic gate with 3 inputs.

| A | B | C | A.B | Q=A.B.C |
|---|---|---|-------|---------|
| 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | | 0 |
| 0 | 1 | 0 | 0 | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | 0 |
| 1 | 1 | 1 | | |

4.3.2 OR Gate

Let us consider the following example to understand OR logic.

- A bus passenger who is in a two door bus can get down from the front or back door.
- If there are several routes to reach your home, you can use any of these routes,

Let us consider the simple electronic circuit given below. (Figure 4.8 and 4.9)

Here A and B two switches and Q bulb are connected to two batteries in parallel. Let us consider the two switches A and B as input and the bulb as output.

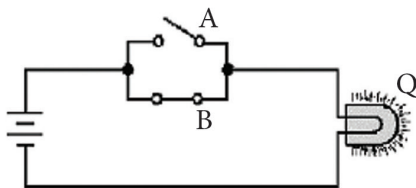


Figure 4.8 - Electronic circuit when one logical state of the two inputs is 1

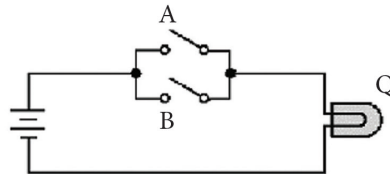


Figure 4.9 – Electronic circuit when the logical state of the two inputs is 0.

The bulb of this circuit will be 'on' when either A or B switch or both A and B switches are closed. This can be tabulated as follows. (Table 4.4)

Table 4.4 - Nature of the bulb of the electronic circuit for OR

| A | B | Q |
|--------|--------|--------------|
| Open | Open | Switched off |
| Open | Closed | Switched on |
| Closed | Open | Switched on |
| Closed | Closed | Switched on |

When either of the switches A or B is "on", it is represented as state "1" (Figure 4.7). The state when both switches are open or the bulb is switched off (Figure 4.8) is indicated as logic "0". This table is called the Truth Table for OR logic gate. (Table 4.5)

Table 4.5 - Truth Table for OR logic gate.

| A | B | Q |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Observation



At least one input should be 1 for the output of OR gate to be 1.

When both input of the above OR gate is 0, output will always be 0. Further, when all inputs are 0 of an OR logic gate with more than 2 inputs, output will always be 0.

In the above table (Table 4.5), when A=1 or B=1 or both A and B are in state "1", Q=1. This table is called the Truth Table related to the OR logic gate.

In Boolean algebra, this operation is symbolically represented as "A + B". Note that this is not the addition operation you use in Mathematics to add numbers. Further, it is not the plus mark to indicate a plus number. Hence, it represents "A OR B".

Given below in Figure 4.9 is the Boolean algebra expression related to the OR gate and the circuit symbols.

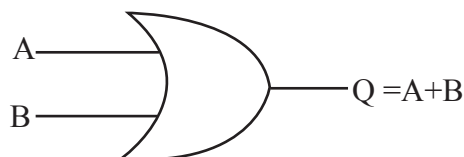


Figure 4.10 - The Boolean algebra expression and the symbol for OR logic gate.

Activity



Following is the logical circuit for AND to get the Q output from A, B and C inputs. (Figure 4.11)

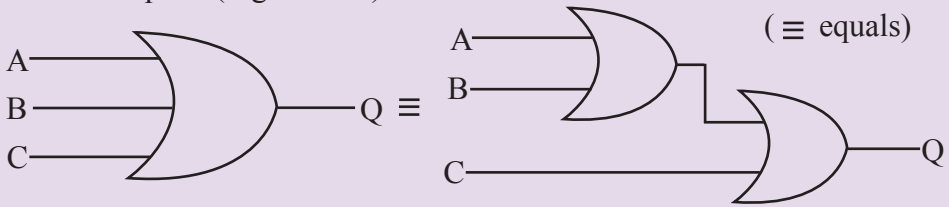


Figure 4.11 – OR logic gate with three inputs.

1. Write the Boolean expression for this.
2. Construct the Truth Table to obtain the above output.

4.3.3 NOT Gate

Let us learn about the complement symbol to understand the NOT gate. The task of the NOT gate is to provide the complement as the output. Thus, complement of “0” is “1” and the complement of “1” is “0”.

The NOT gate is the electronic circuit that provides the complement as the output of the input. Let us consider the circuit given below. (Figure 4.12)

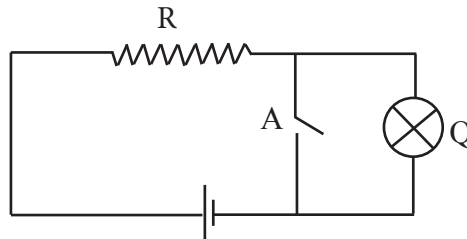


Figure 4.12 – Electronic circuit for NOT logic gate.

When the A switch is closed, the bulb is not switched on as no electricity runs through. However, when the A switch is open, the bulb will be switched on. When the input is A here, the complement of A is the output. In Figure 4.13 is the Boolean algebra expression related to the NOT operation and its circuit symbols.

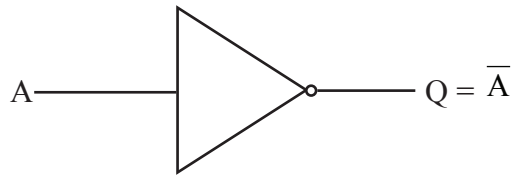


Figure 4.13 - Boolean algebra expression related to NOT logic gate and symbols.

Following is the truth table related to the NOT gate. (Table 4.6)

Table 4.6 - Truth Table related to the NOT logic gate.

| A | Q |
|---|---|
| 0 | 1 |
| 1 | 0 |

4.4 Combinational Logic Gates

The function of devices such as the computer, calculator, washing machine, microwave oven, mobile phone, modern televisions, digital clock, air condition etc is based on the function of logic gates. There are circuits which are designed with various logic gates to get the required output. Such combinational logic circuit can be designed using basic logic gates.

4.4.1 NOR Gate

The logic gate which indicates NOT OR (i.e. complement operation of the OR operation) is called the NOR gate. The logic of the NOR gate is the combination of OR and NOT to get the output from NOT after leading the output to NOT from OR gate. This can be shown as follows. (Figure 4.14)

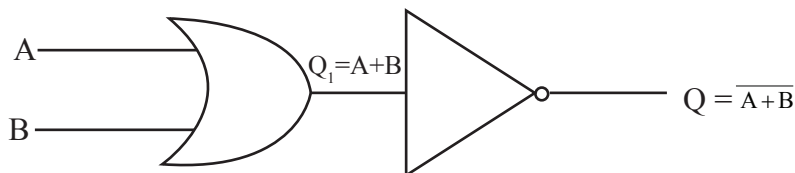


Figure 4.14 – Electronic circuit for the NOR logic gate.

The truth table for this can be built based on basic truth tables. (Table 4.7)

Table 4.7 - Truth Table for NOR logic gate based on basic Truth Tables.

| A | B | $Q_1 = A+B$ | $Q = \overline{A+B}$ |
|---|---|-------------|----------------------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |

Observation



When the both inputs of the NOR logic gate is 0, the output is always "1". Further, in an NOR logic gate with more than 2 inputs, when all those inputs are 0, output is always 1.

Thus, the NOR gate is equivalent to the OR and NOT gates connected in series. Following is the Boolean expression for the NOR gate and the circuit symbols.

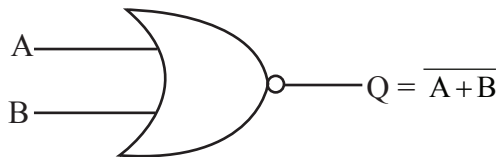


Figure 4.15 - Boolean expression for NOR gate and the circuit symbols

The logic circuit and the Truth Table related to this are given below. (Table 4.8)

Table 4.8 - Truth Table for the NOR logic gate

| A | B | Q |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Activity



The logic circuit with output Q and inputs A, B and C is given below. (Figure 4.16)

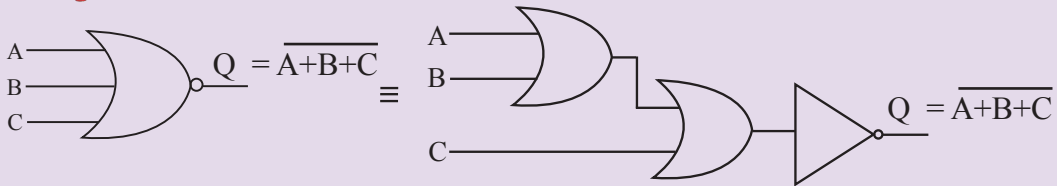


Figure 4.16 – NOR logic gate with three inputs

1. Write the Boolean expression for the above.
2. Construct the truth table.

4.4.2 NAND Gate

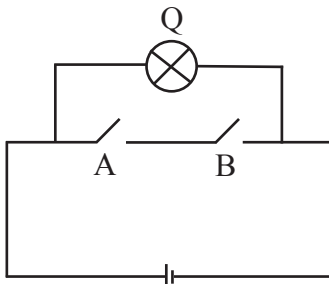


Figure 4.17 – Electronic circuit for the NAND logic gate

• Here, the two switches A and B, and the bulb Q are connected to a battery. Let us consider the two switches as input and the bulb as output. When both A and B switches are closed only the bulb is not switched on. On all the other occasions, the bulb will be switched on.

The logic gate which indicates the complement operation of AND or NOT AND operation is called the NAND gate. This is equivalent to connecting output of AND gate to a NOT gate in series. This can be shown as follows. (Figure 4.18)

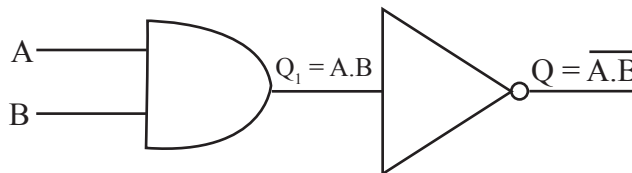


Figure 4.18 - Electronic circuit for NAND logic gate

The Truth Table for this logic based on the Truth Tables related to basic gates. (Table 4.9)

Table 4.9 - The Truth Table for the NAND logic gate based on basic Truth Tables.

| A | B | $Q = A.B$ | $Q = \overline{A.B}$ |
|---|---|-----------|----------------------|
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

The Boolean symbol and expression to represent NAND can be shown as follows. (Figure 4.19)

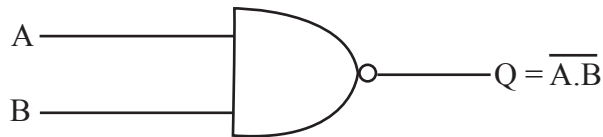


Figure 4.19 – Boolean expression and the symbol for NAND logic gate.

Truth Table for NAND logic gate is given below. (Table 4.10)

Table 4.10 - Truth Table for the NAND logic gate

| A | B | $Q = \overline{A.B}$ |
|---|---|----------------------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Observation



When the both (or all) inputs of NAND logic gate are 1, the output is always 0.

Activity



The following is the NAND logic circuit with three inputs A, B and C with output Q. (Figure 4.20)

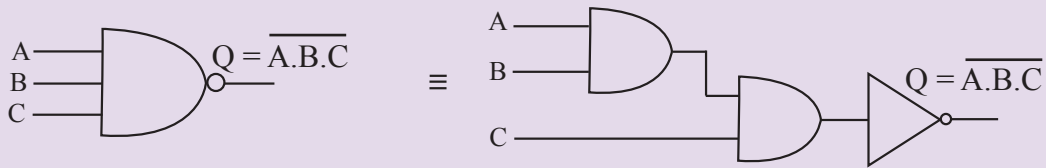


Figure 4.20 – NAND logic gate with three inputs

1. Write the Boolean expression for this.
2. Construct the Truth Table to get the above output.

4.5 Designing Logical Circuits Related to Boolean Expressions

Let us design a circuit with logic gates to get Boolean expression $Q = A.(\overline{A} + B) + \overline{B}$.

There are two inputs in this circuit. These are A and B. The logical circuit for the above Boolean expression is given below. (Figure 4.21)

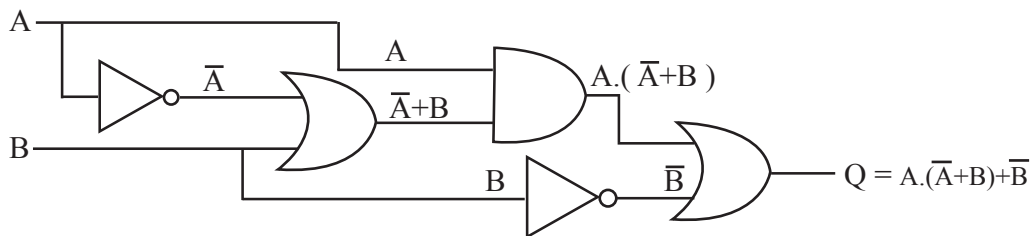


Figure 4.21 – The circuit for the Boolean expression $Q = A.(\overline{A} + B) + \overline{B}$

Activity



1. Draw circuits for the Boolean expressions given below and construct the related truth tables.

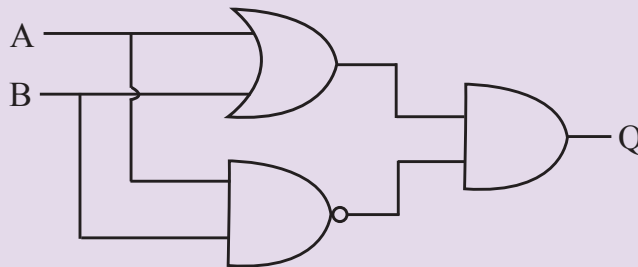
(a) $A+A.B$

(b) $A.(A+B)$

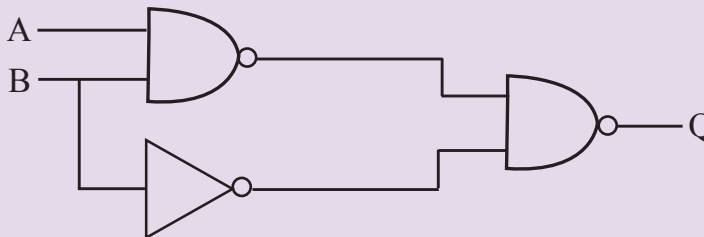
(c) $(A+B).(A.\bar{C})$

2. Write Boolean expressions for the circuits given below and construct Truth Tables.

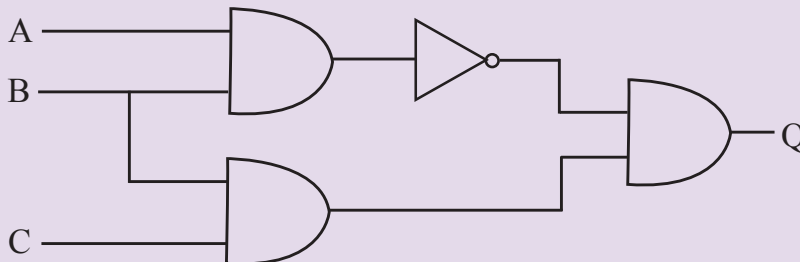
(a).



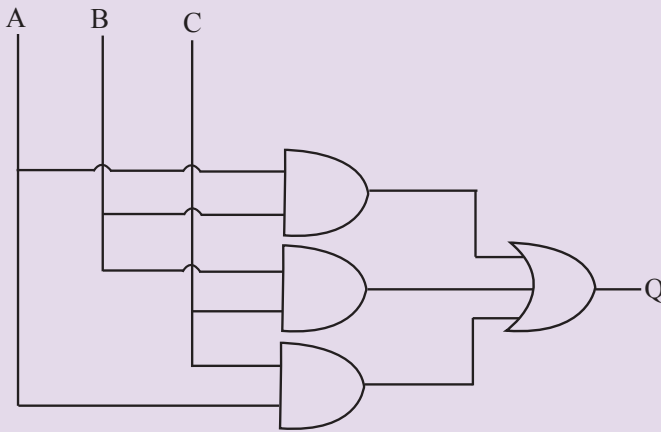
(b).



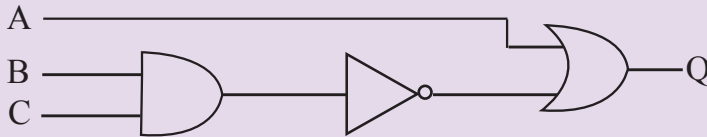
(c).



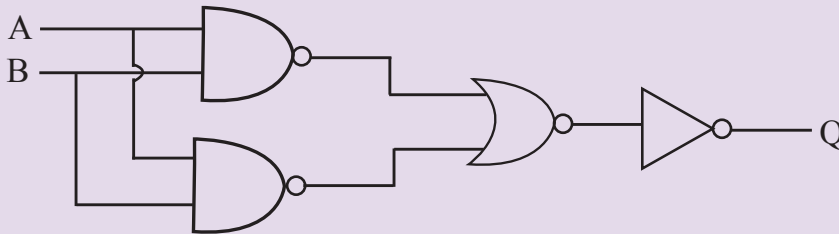
(d).



(e).



(f).



Activity



Use Multimedia Logic (MM Logic) software to draw all the logic circuits you have learnt in this chapter. Observe the function of the circuit. (<http://www.softronix.com/logic.html>)

4.6 Integrated Circuits

Integrated Circuits (ICs) are used to construct electronic circuits. A complex electronic circuit consists of various circuits. For instance, a modern television circuit and a mobile phone consist of many circuits. An integrated circuit is a type of circuit designed for a certain function with devices such as transistors, resistors, capacitors and diodes. Now there are methods to produce circuits in small sizes so that one circuit can be packed in one. Circuits produced in this way are called integrated circuits.

As per Figure 4.23, a micro processor is made of a large number of integrated circuits which use logic gates.

In these integrated circuits, there are logic gates. For instance, Figure 4.22 is a logic circuit which uses an AND gate. Thus, there are integrated circuits designed with the logic gates.

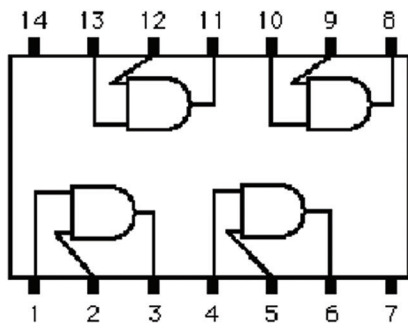


Figure 4.22 – Logic integrated circuit with AND Logic gate

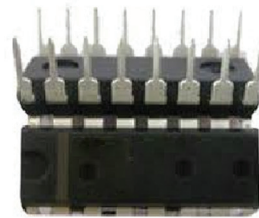


Figure 4.23 – External appearance of an Integrated Circuit.

In this integrated circuit (Figure 4.22), 1, 2, 4, 5, 9, 10, 12 and 13 pins are inputs. 3, 6, 8 and 11 pins are outputs. Figure 4.23 is the external appearance of an integrated circuit which consists of 16 pins.

Activity

1. Consider the Integrated Circuit given below. (Figure 4.24)

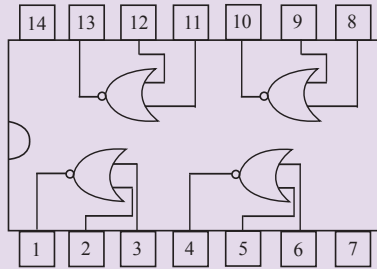


Figure 4.24 – NOR Logic Integrated Circuit

Consider the pins 1, 2 and 3 in the above circuit. If $2 = 0$ and $3 = 0$, what is the value of pin 1?

2. In the Integrated Circuit given below (Figure 4.34), if pins $1 = 1$ and $2 = 1$, what is the value of 3?

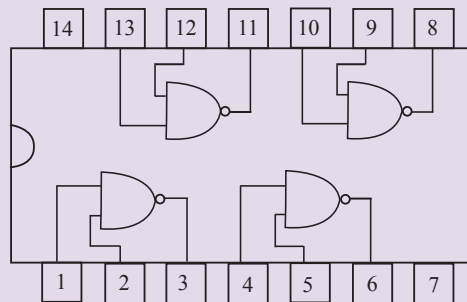


Figure 4.25 – NAND Logic Integrated Circuit

4.7 Practical Usage of Logic Gates

Example 1

Home Alarm System

The circuit given below is an alarm system which informs the house owner of a theft at home. This is designed using OR gates. This circuit protects two windows, the front and back door of the house. When any window or door is opened, an alarm system will be on. The logic gates which is connected to windows and doors are connected to sensors. Input is "1" when windows or doors are open; input is 0 when windows or door are closed. As shown in Figure 4.27, when all the inputs are 0, there will be no alarm.

This means all the windows are closed at this moment. However, there will be an alarm when one input value is 1 or several input values are 1 or all input values are 1. This means when a thief opens one window or door or several of these, there will be an alarm. For instance, Figure 4.28 shows the alarm which warns the house owner when the first window is opened by somebody. The moments the alarm is on like this are shown in Table 4.11.

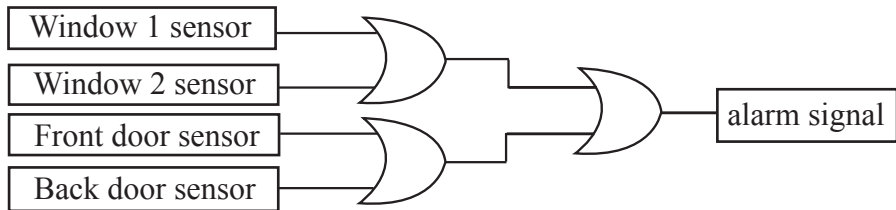


Figure 4. 26 - Home alarm system

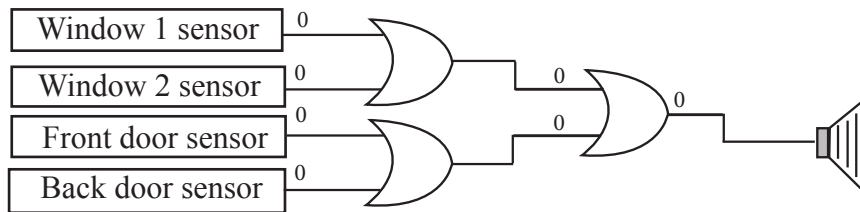


Figure 4.27 - No alarm when doors and windows are closed

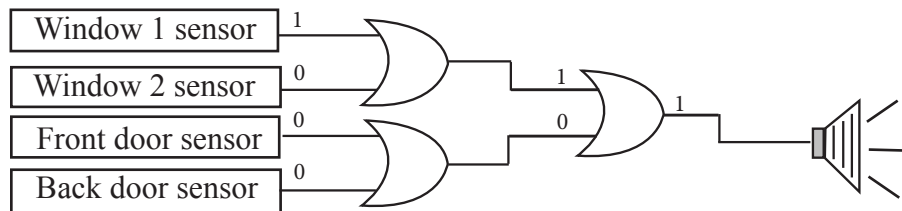


Figure 4.28 – Alarm on when at least one door or window is open

| Window 1 | Window 2 | Front Door | Back Door | Occasion |
|----------|----------|------------|-----------|----------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Status

open = 0
 closed = 1
 warning state = 1
 no warning state = 0

Example 2

Circuits designed to control street lights.

Following Figure 4.29 is a circuit designed using logic gates to control street lights. This is designed using a dark / light sensor, timer and a manual switch.

There are several occasions when the lights are on. These are,

- When only the manual switch is closed
- When the input of the timer is 1 and when the environment is dark
- When only the manual switch is closed

When the manual switch is closed, its input is 1. Hence, the light is on.

When the input of the timer is 1 and when the environment is dark

If the time is in the range of two pre-arranged ranges the input is 1 and otherwise 0. This means, the input is 1 when time is from 6 pm and 6 am, and the input is 0 when from 6 am to 6 pm. Further, the light sensor will be 1 when there is light above pre arranged value and it will be 0 when it is dark. Here, the street light will be on when the timer's input is 1 and light sensor's value is 0. When the environment is gloomy with a rain cloud, the street light will not be on though the light sensor's value is 0 if the time is not between 6 am and 6 pm.

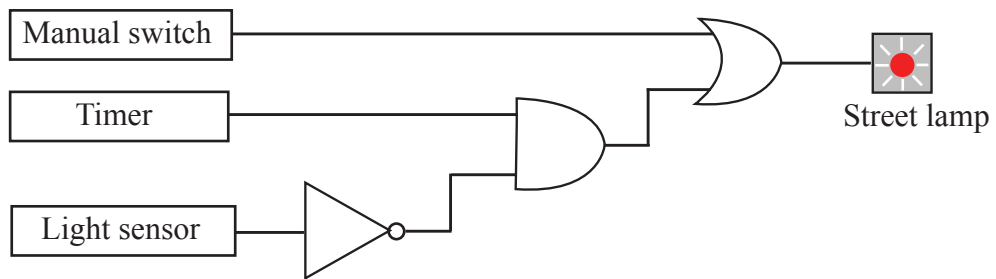


Figure 4.29 - The circuit to control street lights.

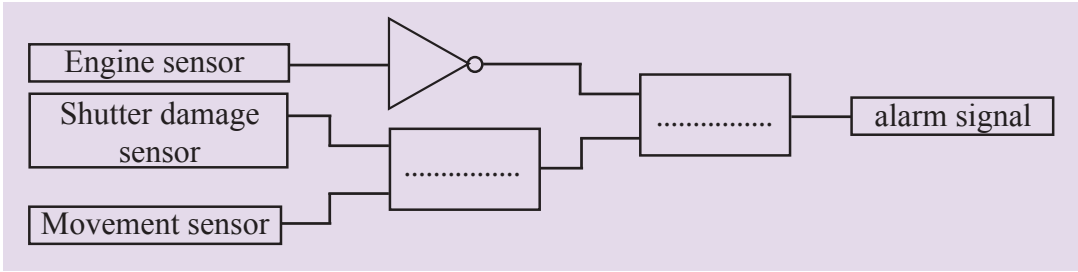
Activity



A motor car manufacturing company has manufactured a motor car protecting circuit to warn if there is a movement in the car when the engine is not on or when there is damage to a shutter. This has three sensors to detect whether the engine is on and another sensor to detect whether there is a damage to a shutter. Another sensor is there to detect whether the car is moving.

This circuit is designed with three basic circuits. One of these is a NOT gate and the other two gates should be included in the empty boxes. What are the logic gates suitable for these?

Draw the circuit.



Summary

| Logic Gate | Symbol | Boolean Expression | Truth Table | | | | | | | | | | | | | | | |
|------------|--------|----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| OR | | $Q = A+B$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | A | B | Q | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| A | B | Q | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | |
| AND | | $Q = A.B$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> | A | B | Q | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| A | B | Q | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | 1 | | | | | | | | | | | | | | | | |
| NOT | | $Q = \bar{A}$ | <table border="1"> <thead> <tr> <th>A</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </tbody> </table> | A | Q | 0 | 1 | 1 | 0 | | | | | | | | | |
| A | Q | | | | | | | | | | | | | | | | | |
| 0 | 1 | | | | | | | | | | | | | | | | | |
| 1 | 0 | | | | | | | | | | | | | | | | | |
| NOR | | $Q = \overline{A+B}$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> | A | B | Q | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |
| A | B | Q | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | |
| 0 | 1 | 0 | | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | |
| NAND | | $Q = \overline{A.B}$ | <table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Q</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table> | A | B | Q | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| A | B | Q | | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | | | | | | | | | | | | | | | | |
| 0 | 1 | 1 | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | | | | | | | | | | | | | | | | |
| 1 | 1 | 0 | | | | | | | | | | | | | | | | |