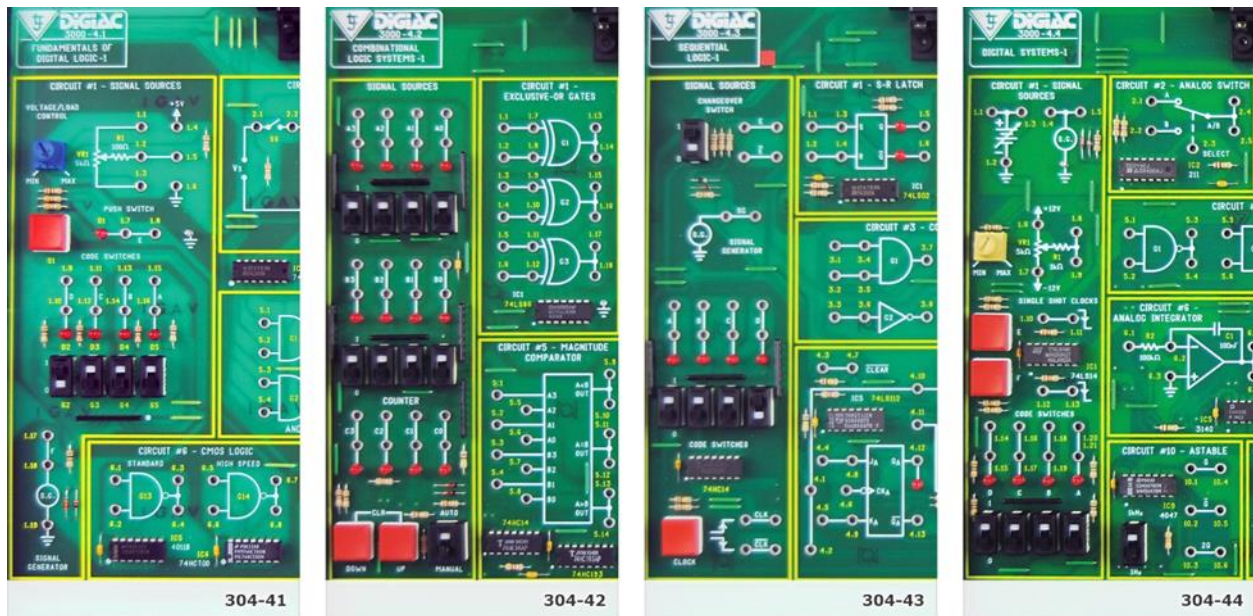


Biomedical Equipment Servicing

Level – II

Based on September 2021, curriculum Version-II



MODULE TITLE: Testing Basic Digital Electronic Components

MODULE CODE: HLT BES2 M08 0822

NOMINAL DURATION: 85Hours

August 2022

Addis Ababa, Ethiopia

Table of Contents

Table of Contents	I
List of Figures and Table	II
Acknowledgement	IV
Acronyms	V
Introduction	1
Unit one: Binary Number System	3
1. 1 Principle and use of binary system	3
1.1.1 Binary Number System	3
1.2 Mathematical operations of binary system	4
1.1.2 Arithmetic Operations of Binary Numbers	4
1.3 Binary number system conversion	8
Self-check -1	10
Unit two: working principle of logic gates	12
2.1 Types of Logic gates and Truth Tables	12
2.1.1 Positive and Negative Logic	12
2.1.2 Truth Table	13
2.1.3 Logic Gates	14
2.1.4 Universal Gate	24
2.2 Constructing logic gate on bread board	26
2.2.1 Constructing AND Gate using Transistors	26
2.3 Testing logic gates	36
Transistor – Basics, and Working	38
Self-check 2	40
Operation sheet-1	41
LAP Test	42
Unit Three: Basic Digital Electronic Circuits	44
3.1 Principles of Combinational logic gate	45
3.2 Basic digital electronic circuits	54
3.2.1 Basics of Flip Flop	55
3.2.2 Counters	58
3.3.3 Registers	62

3.2.4 Transistor technology	64
3.3 Applications of digital electronic circuits.	65
3.4 Testing digital electronic circuit	68
Self-check 3	Error! Bookmark not defined.
Operation sheet-2	78
LAP Test	79
Reference	79
Developers Profile	80

List of Figures and Table

Table 2.1: Two-input logic system.....	14
Table 2.2: Truth table of a three-input logic system.....	14
Figure 2.1:OR Gate	15
Figure 2.2: Two-input OR gate	16
Figure 2.4: NAND Gate.....	17
Figure 2.5: Two-input AND gate.....	17
Figure 2.6 NOT Gate.....	19
Figure 2.11 NAND Gate	22
Figure 2.14 NANG Gate	25
Figure 2.15 NOR Gate.....	25
Figure 2.16 AND gate with transistor	Error! Bookmark not defined.
Figure 2.17 transistor	28
Figure 2.18 AND gate construction	29
Figure 2.19 AND gate on breadboard	30
Figure 2.20 working of and Gate using Transistor	31
Figure 2.21 logic gate IC.....	36
Figure 3.1 combinational circuit	45
Figure 3.2 half adder.....	46
Figure 3.3 Full Adder	47
Figure 3.4 4 Bit Parallel Adder	48
Figure 3.5 4 Bit Parallel Subtractor	49
Figure 3.6 Half Subtractors.....	50
Figure 3.7 Full Subtractors	51
Figure 3.8 Multiplexers	53
Figure 3.9 Demultiplexers	54
Figure 3.10 Decoder.....	54

Figure 3.11 SR Flip Flop	56
Table 3.12 truth table of S-R flip.....	56
Figure 3.12 J-K Flip-flop.....	56
Table 3.13 truth table of J-K Flip-flop	57
Figure 3.14 D-flip flop	57
Figure 3.15 T-flip flop	58
Table 3.4 truth table of T-flip-flop	58
Table 3.5 counter truth table.....	59
Figure 3.16 Signal Diagram.....	60
Figure 3.17 Logical Diagram.....	61
Figure 3.18 Signal Diagram.....	62
Figure 3.19 Types register	63
Figure 3.22. An ac signal being converted to digital pulses.	65
Figure 3.23. Digital discs	66
Figure 3.24 basic logic probe.....	69
Figure 3.25 Output stuck at '0' (for a single gate)	71
Figure 3.26 Common in a digital IC.....	73

Acknowledgement

Ministry of Labor and Skills, Ministry of Health, regional health Science College and polytechnic colleges wish to extend thanks and appreciation to the many representatives of TVET instructors and respective industry experts who donated their time and expertise to the development of this Teaching, Training and Learning Materials (TTLM).

IV	<u>Author/Copyright</u> Ministry of Labor and Skills	Module title: Test Basic Digital Electronic Components	Training module Version -1 August , 2022
----	---	--	--

Acronyms

AC	Alternate current
BJT	Bipolar junction transistor
DC	Direct current
DMM	Digital Multimeter
IC	Integrated circuit
LED	Light emitted diode
MoH	Ministry of health
MoLS	Ministry of labor and skill
MOSFET	Metal oxide semiconductor field effect transistor
NPN	Negative positive negative
PNP	Positive negative positive

Introduction digital electronics

Digital electronics is a field of electronics involving the study of digital signals and the engineering of devices that use or produce them. This is in contrast to analog electronics and analog signals. Digital electronic circuits are usually made from large assemblies of logic gates, often packaged in integrated circuits.

Digital electronics is the branch of electronics that deals with the study of digital signals and the components that use or create them.

Digital electronics, digital circuits, and digital technology are electronics that are operated on digital signals. Digital techniques are much easier for getting the electronic device. These devices are used to switch into one of the known states apart from reproducing a continuous range of values. Digital circuits are made from a large collection of logic gates and a simple electronic representation of the Boolean logic function.

1	<u>Author/Copyright</u> Ministry of Labor and Skills	Module title: Test Basic Digital Electronic Components	Training module Version -1 August , 2022
---	---	--	--

Module units:

- Binary Number System
- Logic Gates
- Applying logic gates to demonstrate basic digital electronic circuits

Learning objectives of the Module

At the end of this session, the students will be able to:

- Demonstrate the principle of binary number system
- Demonstrate the working principle of logic gates
- Apply logic gates to demonstrate basic digital electronic circuits

Module Learning Instructions:

- Read the specific objectives of this Learning Guide.
- Follow the instructions described below.
- Read the information written in the information Sheets
- Accomplish the Self-checks
- Perform Operation Sheets
- Do the “LAP test”

2	<u>Author/Copyright</u> Ministry of Labor and Skills	Module title: Test Basic Digital Electronic Components	Training module Version -1 August , 2022
---	---	--	--

Unit one: Binary Number System

This learning unit is developed to provide the trainees the necessary information regarding the following content coverage and topics:

- Principles and use of binary system
- Mathematical operations of binary system
- Binary number system conversion

This unit will also assist you to attain the unit stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Demonstrate Principles and use of binary system
- Demonstrate Mathematical operations of binary system
- Convert Binary number system

1. 1 Principle and use of binary system

1.1.1 Binary Number System:

According to digital electronics and mathematics, a binary number is defined as a number that is expressed in the binary system or base 2 numeral system. It describes numeric values by two separate symbols; 1 (one) and 0 (zero). The base-2 system is the positional notation with 2 as a radix.

The binary system is applied internally by almost all latest computers and computer-based devices because of its direct implementation in electronic circuits using logic gates. Every digit is referred to as a bit.

Data in digital circuits and computers is stored and transmitted as a series of zeros and ones and so various numbering systems are used to represent the data. Conveniently, binary numbers have only two digits that are 0 and 1, so every piece of data (number) can be represented using a binary numbering system.

Unlike a linear or analogue circuit such as AC amplifiers which process signals that are constantly changing from one value to another, for example amplitude or frequency. Digital circuits process signals that contain just two voltage levels or states, labelled, Logic “0” and Logic “1”.

Generally, a logic “1” represents a higher voltage, such as 5 volts, which is commonly referred to as a HIGH value, while a logic “0” represents a low voltage, such as 0 volts or ground, and

is commonly referred to as a LOW value. These two discrete voltage levels representing the digital values of “1’s” (one’s) and “0’s” (zeros) are commonly called: Binary Digit and in digital and computational circuits and applications they are normally referred to as binary BITS

1.2 Mathematical operations of binary system

The study of number systems is important from the viewpoint of understanding how data are represented before they can be processed by any digital system including a digital computer. It is one of the most basic topics in digital electronics.

Different characteristics that define a number system include the number of independent digits used in the number system, the place values of the different digits constituting the number and the maximum numbers that can be written with the given number of digits. Among the three characteristic parameters, the most fundamental is the number of independent digits or symbols used in the number system. It is known as the radix or base of the number system. The decimal number system with which we are all so familiar can be said to have a radix of 10 as it has 10 independent digits, i.e. 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. Similarly, the binary number system with only two independent digits, 0 and 1, is a radix-2 number system. The octal and hexadecimal number systems have a radix (or base) of 8 and 16 respectively. The place values of different digits in the integer part of the number are given by r^{-1} , r^{-2} , r^{-3} and so on, starting with the digit adjacent to the radix point. For the fractional part, these are r^{-1} , r^{-2} , r^{-3} and so on, again starting with the digit next to the radix point. Here, r is the radix of the number system. Also, maximum numbers that can be written with n digits in a given number system are equal to r^n .

1.1.2 Arithmetic Operations of Binary Numbers

Binary Arithmetic:

Binary arithmetic includes the basic arithmetic operations of **addition, subtraction, multiplication and division.**

The following sections present the rules that apply to these operations when they are performed on binary numbers.

- **Binary Addition:** Binary addition is performed in the same way as addition in the decimal system and is, in fact, much easier to master. Binary addition obeys the following four basic rules:

$$\begin{array}{r} 0 \\ +0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ +1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +1 \\ \hline 10 \end{array}$$

The results of the last rule may seem somewhat strange, remember that these are binary numbers. Put into words, the last rule states that binary one + binary one = binary two = binary "one zero" When adding more than single-digit binary number, carry into, higher order columns as is done when adding decimal numbers.

For example 11 and 10 are added as follows:

$$\begin{array}{r} 11 \\ + 10 \\ \hline 101 \end{array}$$

In the first column (L S C or 2^0) '1 plus 0 equal 1. In the second column (2^1) 1 plus 1 equals 0 with a carry of 1 into the third column (2^2). When we add 1 + 1. + 1 (carry) produces 11, recorded as 1 with a carry to the next column.

Example 12: Add (a) 111 and 101 (b) 1010, 1001 and 1101.

(a)

$$\begin{array}{r} (1)(1) \\ 111 \\ 101 \\ \hline 1100 \end{array}$$

(B)

$$\begin{array}{r} (2)(1)(1)(1) \\ 1010 \\ 1001 \\ 1101 \\ \hline 10000 \end{array}$$

- **Binary Subtraction:** Binary subtraction is just as simple as addition subtraction of one bit from another.
- Obey the following four basic rules $0 - 0 = 0$ $1 - 1 = 0$ $1 - 0 = 1$ $10 - 1 = 1$ with a transfer (borrow) of 1. When doing subtracting, it is sometimes necessary to borrow from the next higher-order column. The only it will be necessary to borrow is when we try to subtract a 1 from a 0. In this case a 1 is borrowed from the next higher-order column, which leaves a 0 in that column and creates a 10 i.e., 2 in the column being subtracted. The following examples illustrate binary subtraction.

Example 13: Perform the following subtractions. (a) $11 - 01$, (b) $11-10$ (c) $100 - 011$

Solution:

$$\begin{array}{r} 11 \\ -01 \\ \hline \end{array} \quad \begin{array}{r} 11 \\ -10 \\ \hline \end{array} \quad \begin{array}{r} 100 \\ -011 \\ \hline \end{array}$$

(a) $\frac{10}{10}$ (b) $\frac{01}{01}$ (c) $\frac{001}{001}$

Part (c) involves to borrows, which handled as follows. Since a 1 is to be subtracted from a 0 in the first column, a borrow is required from the next higher order column. However, it also contains a 0; therefore, the second column must borrow the 1 in the third column. This leaves a 0 in the third column and place a 10 in the second column. Borrowing a 1 from 10 leaves a 1 in the second column and places a 10 i.e, 2 in the first column: When subtracting a larger number from a smaller number, the results will be negative. To perform this subtraction, one must subtract the smaller number from the larger and prefix the results with the sign of the larger number.

- **Binary multiplication:** Binary multiplication is performed in the same manner as decimal multiplication. It is much easier, since there are only two possible results of multiplying two bits.

The Binary multiplication obeys the four basic rules.

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Example: Multiply the following binary numbers.

(a) 101×11 (b) 1101×10 (c) 1010×101 (d) 1011×1010

Solution

<p>(a)</p> $\begin{array}{r} 101 \\ \times 11 \\ \hline 101 \\ 101 \\ \hline 1111 \end{array}$	<p>(b)</p> $\begin{array}{r} 11101 \\ \times 10 \\ \hline 0000 \\ 1101 \\ \hline 11010 \end{array}$
<p>(c)</p> $\begin{array}{r} 1010 \\ \times 101 \\ \hline 1010 \\ 0000 \\ 1010 \\ \hline 110010 \end{array}$	<p>(d)</p> $\begin{array}{r} 1011 \\ \times 1010 \\ \hline 0000 \\ 1011 \\ 0000 \\ 1011 \\ \hline 1101110 \end{array}$

- **Binary Division:** Division in the binary number system employs the same procedure as division in the decimal system, as will be seen in the following examples.

Example: Perform the following binary division.

(a) $110 \div 11$

(b) $1100 \div 11$

Solution:

<p>(a)</p> $\begin{array}{r} 10 \\ 11 \overline{)110} \\ \underline{11} \\ 00 \\ \underline{00} \\ 00 \\ \underline{00} \end{array}$	<p>(b)</p> $\begin{array}{r} 100 \\ 11 \overline{)11000} \\ \underline{11} \\ 00 \\ \underline{00} \\ 00 \\ \underline{00} \\ 00 \end{array}$
--	---

Binary division problems with remainders are also treated the same as in the decimal system, as illustrates the following example.

Example 18: Perform the following binary division:

(a) $1111 \div 110$ (b) $1100 \div 101$

Solution: (a)
$$\begin{array}{r} 10.1 \\ 110 \overline{) 1111.00} \\ \underline{110} \\ 110 \\ \underline{110} \\ 000 \end{array}$$

(b)
$$\begin{array}{r} 10.011 \\ 110 \overline{) 1100.00} \\ \underline{101} \\ 100 \\ \underline{000} \\ 1000 \\ \underline{101} \\ 110 \\ \underline{101} \\ 1 \\ \text{(remainder)} \end{array}$$

1.3 Binary number system conversion

The decimal number system is a radix-10 number system and therefore has 10 different digits or symbols. These are 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9. All higher numbers after ‘9’ are represented in terms of these 10 digits only. The process of writing higher-order numbers after ‘9’ consists in writing the second digit (i.e. ‘1’) first, followed by the other digits, one by one, to obtain the next 10 numbers from ‘10’ to ‘19’. The next 10 numbers from ‘20’ to ‘29’ are obtained by writing the third digit (i.e. ‘2’) first, followed by digits ‘0’ to ‘9’, one by one. The process continues until we have exhausted all possible two-digit combinations and reached ‘99’. Then we begin with three-digit combinations. The first three-digit number consists of the lowest two-digit number followed by ‘0’ (i.e. 100), and the process goes on endlessly. The place values of different digits in a mixed decimal number, starting from the decimal point, are 100, 101, 102 and so on (for the integer part) and 10⁻¹, 10⁻², 10⁻³ and so on (for the fractional part)

The value or magnitude of a given decimal number can be expressed as the sum of the various digits multiplied by their place values or weights. As an illustration, in the case of the decimal number 3586.265, the integer part (i.e. 3586) can be expressed as

$$3586 = 6 \times 100 + 8 \times 101 + 5 \times 102 + 3 \times 103 = 6 + 80 + 500 + 3000 = 3586$$

And the fractional part can be expressed as

$$265 = 2 \times 10^{-1} + 6 \times 10^{-2} + 5 \times 10^{-3} = 0.2 + 0.06 + 0.005 = 0.265$$

We have seen that the place values are a function of the radix of the concerned number system and the position of the digits. We will also discover in subsequent sections that the concept of each digit having a place value depending upon the position of the digit and the radix of the number system is equally valid for the other more relevant number systems.

- Binary-to-Decimal Conversion** The decimal equivalent of the binary number $(1001.0101)_2$ is determined as follows:
 - ✓ The integer part = 1001
 - ✓ The decimal equivalent = $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = 1 + 0 + 0 + 8 = 9$
 - ✓ The fractional part = .0101
 - ✓ Therefore, the decimal equivalent = $0 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = 0 + 0.25 + 0 + 0.0625 = 0.3125$
 - ✓ Therefore, the decimal equivalent of $(1001.0101)_2 = 9.3125$
- Decimal-to-Binary Conversion:** As outlined earlier, the integer and fractional parts are worked on separately. For the integer part, the binary equivalent can be found by successively dividing the integer part of the number by 2 and recording the remainders until the quotient becomes '0'. The remainders written in reverse order constitute the binary equivalent. For the fractional part, it is found by successively multiplying the fractional part of the decimal number by 2 and recording the carry until the result of multiplication is '0'. The carry sequence written in forward order constitutes the binary equivalent of the fractional part of the decimal number. If the result of multiplication does not seem to be heading towards zero in the case of the fractional part, the process may be continued only until the requisite number of equivalent bits has been obtained. This method of decimal–binary conversion is popularly known as the double-dabble method. The process can be best illustrated with the help of an example

Example 1.3 we will find the binary equivalent of $(13.375)_{10}$

Solution

- The integer part = 13

Divisor	Dividend	Remainder
2	13	—
2	6	1
2	3	0
2	1	1
—	0	1

The binary equivalent of $(13)_{10}$ is therefore $(1101)_2$

- ✓ The fractional part = .375
- ✓ $0.375 \times 2 = 0.75$ with a carry of 0
- ✓ $0.75 \times 2 = 0.5$ with a carry of 1
- ✓ $0.5 \times 2 = 0$ with a carry of 1
- ✓ The binary equivalent of $(0.375)_{10} = (.011)_2$
- ✓ Therefore, the binary equivalent of $(13.375)_{10} = (1101.011)_2$

Self-check -1

Part I True or False

1. Digital techniques are much harder for getting the electronic device
2. Data in digital circuits and computers is stored and transmitted as a series of numeric numbers
3. logic “1” represents a higher voltage, such as 5 volts, which is commonly referred to as a HIGH value
4. The octal and hexadecimal number systems have a radix (or base) of 8 and 16 respectively
5. According to digital electronics and mathematics, a binary number is defined as a number that is expressed in the binary system or base 2 numeral system

Part II multiple choice

1. what is the addition value of 111 and 101
A) 1001 B) 1111 C) 1100 D) 1000

2. $11 - 01 = \text{-----}$

A) 1001 B) 1111 C) 1100 D) 10

3. $101 \times 11 = \text{-----}$

A) 1001 B) 1111 C) 1100 D) 1111

Part III short answer

1. Explain Arithmetic Operations of Binary Numbers

Unit two: working principle of logic gates

This learning unit is developed to provide the trainees the necessary information regarding the following content coverage and topics:

- Types of Logic gates and Truth Tables
- Constructing logic gate on bread board
- Testing logic gates

This unit will also assist you to attain the unit stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Identify Different types of Logic gates and truth tables for each Logic gates
- Construct logic gate on bread board
- Test logic gates

2.1 Types of Logic gates and Truth Tables

Logic gates are electronic circuits that can be used to implement the most elementary logic expressions, also known as Boolean expressions. The logic gate is the most basic building block of combinational logic. There are three basic logic gates, namely

- ✓ OR gate,
- ✓ AND gate
- ✓ NOT gate.

Other logic gates that are derived from these basic gates are the NAND gate, the NOR gate, the EXCLUSIVE -OR gate and the EXCLUSIVE-NOR gate. The treatment of the subject matter is mainly with the help of respective truth tables and Boolean expressions.

2.1.1 Positive and Negative Logic

The binary variables, as we know, can have either of the two states, i.e. the logic '0' state or the logic '1' state. These logic states in digital systems such as computers, for instance, are represented by two different voltage levels or two different current levels. If the more positive of the two voltage or current levels represents a logic '1' and the less positive of the two levels represents a logic '0', then the logic system is referred to as a positive logic system. If the more positive of the two voltage or current levels represents a logic '0' and the less positive of the

two levels represents a logic '1', then the logic system is referred to as a negative logic system. The following examples further illustrate this concept

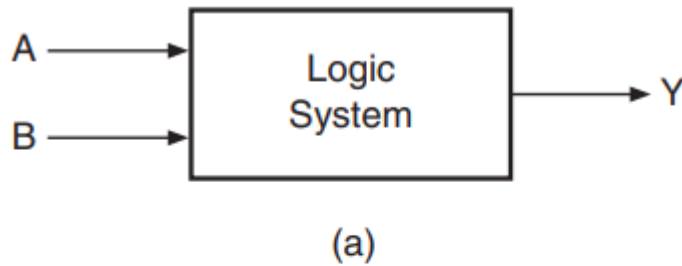
If the two voltage levels are 0 V and +5 V, then in the positive logic system the 0 V represents a logic '0' and the +5 V represents a logic '1'. In the negative logic system, 0 V represents a logic '1' and +5 V represents a logic '0'. If the two voltage levels are 0 V and -5 V, then in the positive logic system the 0 V represents a logic '1' and the -5 V represents a logic '0'. In the negative logic system, 0 V represents a logic '0' and -5 V represents a logic '1'.

2.1.2 Truth Table

A truth table lists all possible combinations of input binary variables and the corresponding outputs of a logic system. The logic system output can be found from the logic expression, often referred to as the Boolean expression that relates the output with the inputs of that very logic system. When the number of input binary variables is only one, then there are only two possible inputs, i.e. '0' and '1'. If the number of inputs is two, there can be four possible input combinations, i.e. 00, 01, 10 and 11

Similarly, for three input binary variables, the number of possible input combinations becomes eight, i.e. 000, 001, 010, 011, 100, 101, 110 and 111. This statement can be generalized to say that, if a logic circuit has n binary inputs, its truth table will have 2^n possible input combinations, or in other words 2^n rows.

Table 2.1: Two-input logic system



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Table 2.2: Truth table of a three-input logic system

2.1.3 Logic Gates

The logic gate is the most basic building block of any digital system, including computers. Each one of the basic logic gates is a piece of hardware or an electronic circuit that can be used to implement some basic logic expression. While laws of Boolean algebra could be used to do manipulation with binary variables and simplify logic expressions, these are actually

implemented in a digital system with the help of electronic circuits called logic gates. The three basic logic gates are the OR gate, the AND gate and the NOT gate

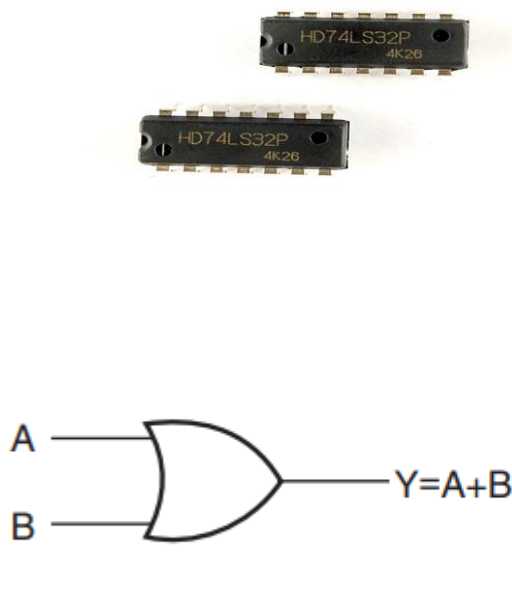
✓ OR Gate

Figure 2.1:OR Gate

An OR gate performs an ORing operation on two or more than two logic variables. The OR operation on two independent logic variables A and B is written as $Y = A+B$ and reads as Y equals A OR B and not as A plus B. An OR gate is a logic circuit with two or more inputs and one output. The output of an OR gate is LOW only when all of its inputs are LOW. For all other possible input combinations, the output is HIGH. This statement when interpreted for a positive logic system means the following. The output of an OR gate is a logic '0' only when all of its inputs are at logic '0'. For all other possible input combinations, the output is a logic '1'. Figure 4.3 shows the circuit symbol and the truth table of a two-input OR gate. The operation of a two-input OR gate is explained by the logic expression

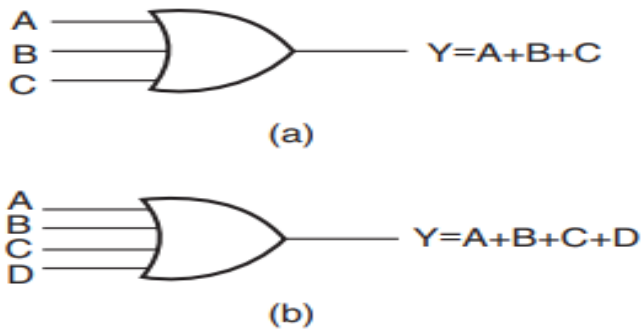
$$Y = A+B$$

As an illustration, if we have four logic variables and we want to know the logical output of $(A + B + C + D)$, then it would be the output of a four-input OR gate with A, B, C and D as its inputs



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Figure 2.2: Two-input OR gate



A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(c)

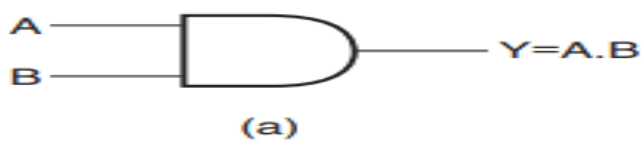
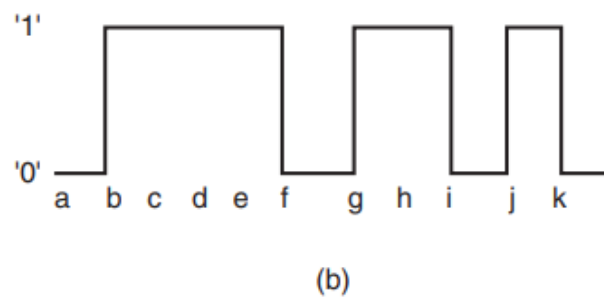
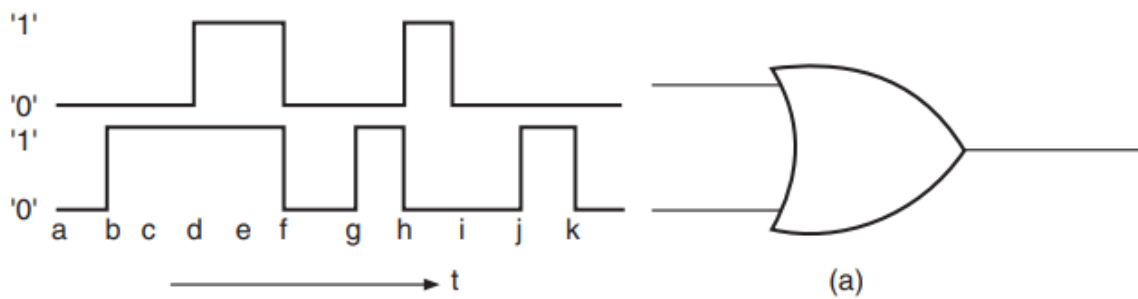
Figure 2.3 (a) Three-input OR gate, (b) four-input OR gate and (c) the truth table of a three-input OR gat

• AND Gate

An AND gate is a logic circuit having two or more inputs and one output. The output of an AND gate is HIGH only when all of its inputs are in the HIGH state. In all other cases, the output is LOW. When interpreted for a positive logic system, this means that the output of the AND gate is a logic '1' only when all of its inputs are in logic '1' state. In all other cases, the output is logic '0'. The AND operation on two independent logic variables A and B is written as $Y = A.B$ and reads as Y equals A AND B and not as A multiplied by B. Here, A and B are input logic variables and Y is the output. An AND gate performs an ANDing operation



Figure 2.4: NAND Gate

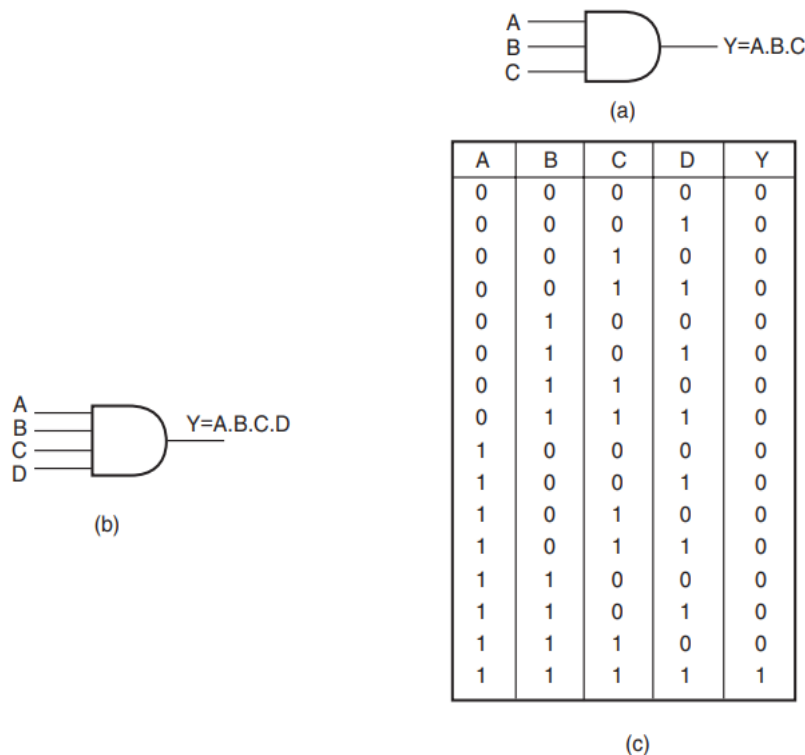


(b)

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Figure 2.5: Two-input AND gate

Table 2.1 truth table of AND gate



- for a two-input AND gate, $Y = A.B$;
- for a three-input AND gate, $Y = A.B.C$;
- For a four-input AND gate, $Y = A.B.C.D$.

If we interpret the basic definition of OR and AND gates for a negative logic system, we have an interesting observation. We find that an OR gate in a positive logic system is an AND gate in a negative logic system. Also, a positive AND is a negative OR

• NOT Gate

A NOT gate is a one-input, one-output logic circuit whose output is always the complement of the input. That is, a LOW input produces a HIGH output, and vice versa. When interpreted for a positive logic system, a logic '0' at the input produces a logic '1' at the output, and vice versa. It is also known as a 'complementing circuit' or an 'inverting circuit'. Figure 4.10 shows the circuit symbol and the truth table. The NOT operation on a logic variable X is denoted as \overline{X} or X' . That is, if X is the input to a NOT circuit, then its output Y is given by $Y = \overline{X}$ or X' and reads as Y equals NOT X . Thus, if $X = 0$, $Y = 1$ and if $X = 1$, $Y = 0$.

Example: For the logic circuit arrangements of Figs 2.9 (a) and (b), draw the output waveform. Solution In the case of the OR gate arrangement of Fig. 4.11(a), the output will be permanently in logic '1' state as the two inputs can never be in logic '0' state together owing to the presence of the inverter. In the case of the AND gate arrangement of Fig. 2.9 (b), the output will be permanently in logic '0' state as the two inputs can never be in logic '1' state together owing to the presence of the inverter

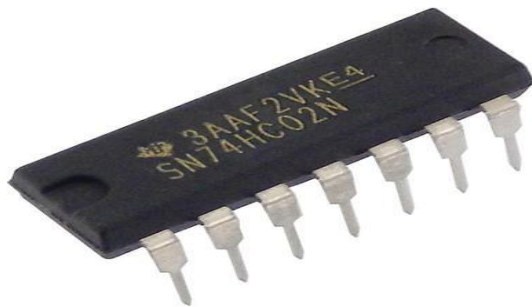


Figure 2.6 NOT Gate

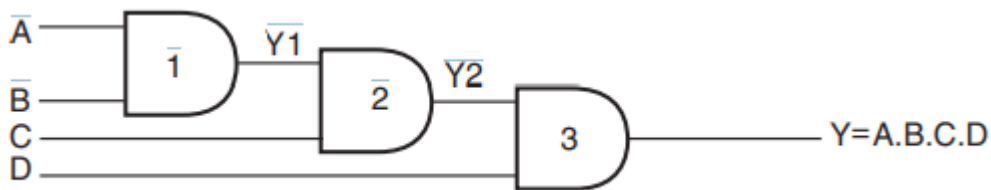


Figure 2.7 Implementation of a four-input AND gate using two-input AND gate

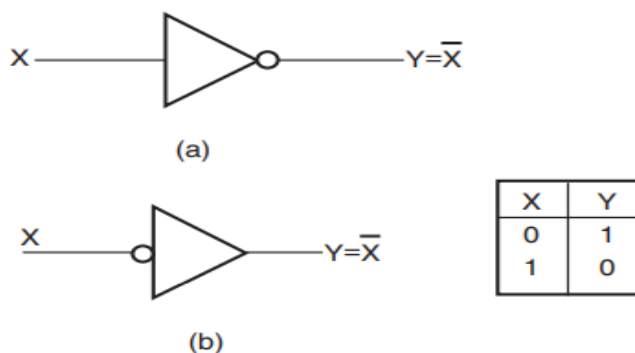


Figure 2.8 (a) Circuit symbol of a NOT circuit and (b) the truth table of a NOT circuit.

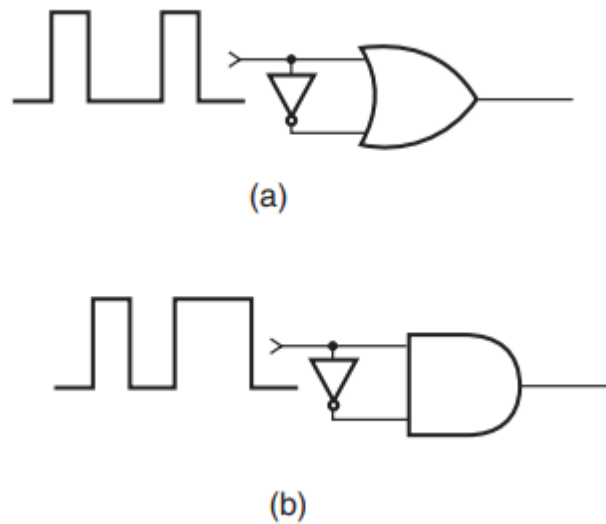


Figure 2.9 Example

• EXCLUSIVE-OR Gate

The EXCLUSIVE-OR gate, commonly written as EX-OR gate, is a two-input, one-output gate. Figures 2.10 (a) and (b) respectively show the logic symbol and truth table of a two-input EX-OR gate. As can be seen from the truth table, the output of an EX-OR gate is a logic '1' when the inputs are unlike and a logic '0' when the inputs are like. Although EX-OR gates are available in integrated circuit form only as two-input gates, unlike other gates which are available in multiple inputs also, multiple-input EX-OR logic functions can be implemented using more than one two-input gates. The truth table of a multiple-input EX-OR function can be expressed as follows. The output of a multiple-input EX-OR logic function is a logic '1' when the number of 1s in the input sequence is odd and a logic '0' when the number of 1s in the input sequence is even, including zero. That is, an all 0s input sequence also produces a logic '0' at the output. Figure 4.12(c) shows the truth table of a four-input EX-OR function. The output of a two-input EX-OR gate is expressed by

$$Y = (A \oplus B) = \bar{A}B + A\bar{B}$$

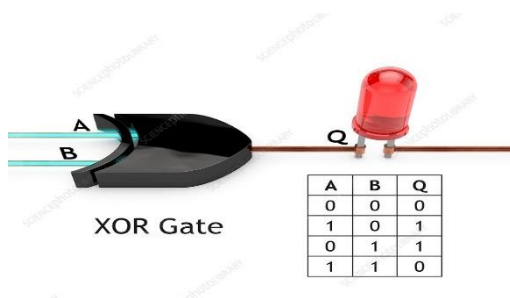


Figure :EXCLUSIVE-OR Gate



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

(b)

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

(c)

Figure 2.10 (a) Circuit symbol of a two-input EXCLUSIVE-OR gate, (b) the truth table of a two-input EXCLUSIVE-OR gate and (c) the truth table of a four-input EXCLUSIVE-OR gate

- **NAND Gate**

NAND stands for NOT AND. An AND gate followed by a NOT circuit makes it a NAND gate. The truth table of a NAND gate is obtained from the truth table of an AND gate by complementing the output entries. The output of a NAND gate is a logic '0' when all its inputs are a logic '1'. For all other input combinations, the output is a logic '1'. NAND gate operation is logically expressed as

$$Y = \overline{A.B}$$

In general, the Boolean expression for a NAND gate with more than two inputs can be written as

$$Y = \overline{(A.B.C.D...)}$$

NOR Gate NOR stands for NOT OR. An OR gate followed by a NOT circuit makes it a NOR gate [Fig. 4.16(a)]. The truth table of a NOR gate is obtained from the truth table of an OR gate by complementing the output entries. The output of a NOR gate is a logic '1' when all its inputs are logic '0'. For all other input combinations, the output is a logic '0'. The output of a two-input NOR gate is logically expressed as

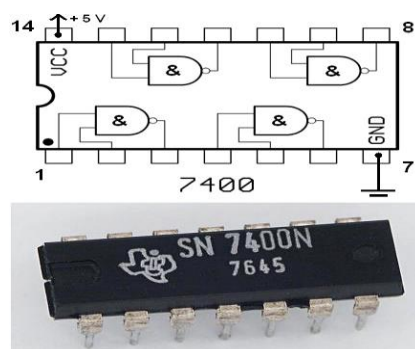


Figure 2.11 NAND Gate

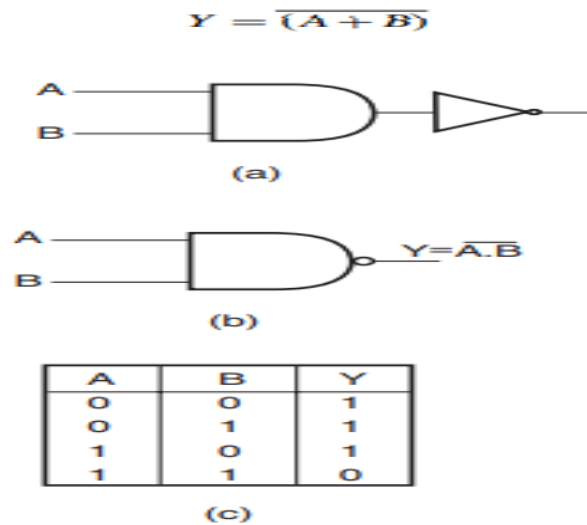


Figure 2.12 (a) Two-input NAND implementation using an AND gate and a NOT circuit, (b) the circuit symbol of a two-input NAND gate and (c) the truth table of a two-input NAND gate

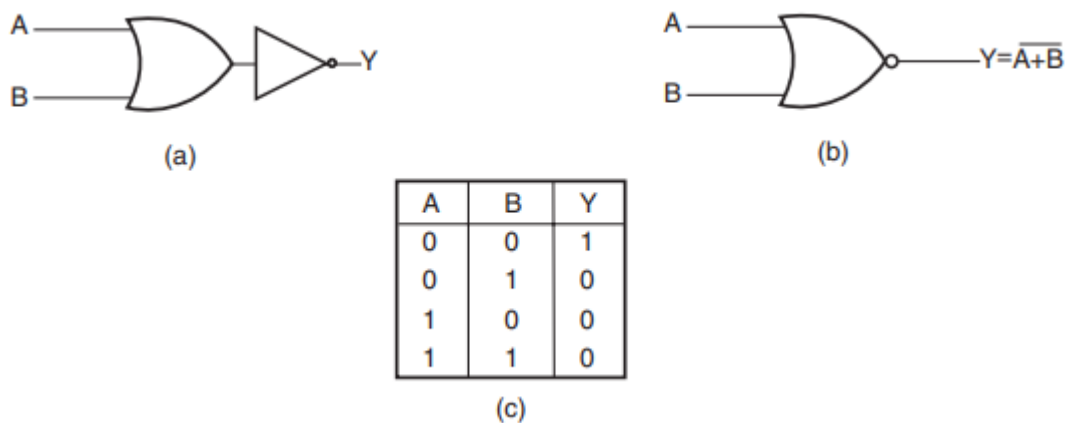


Figure 2.13 (a) Two-input NOR implementation using an OR gate and a NOT circuit, (b) the circuit symbol of a two-input NOR gate and (c) the truth table of a two-input NOR gate

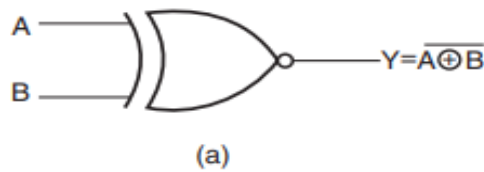
In general, the Boolean expression for a NOR gate with more than two inputs can be written as

$$Y = \overline{(A + B + C + D...)}$$

- EXCLUSIVE-NOR Gate

EXCLUSIVE-NOR (commonly written as EX-NOR) means NOT of EX-OR, i.e. the logic gate that we get by complementing the output of an EX-OR gate. Figure 4.17 shows its circuit symbol along with its truth table. The truth table of an EX-NOR gate is obtained from the truth table of an EX-OR gate by complementing the output entries. Logically,

$$Y = (\overline{A \oplus B}) = (A.B + \overline{A}.\overline{B})$$



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

(b)

Figure 2.13 (a) Circuit symbol of a two-input EXCLUSIVE-NOR gate and (b) the truth table of a two-input EXCLUSIVE-NOR gate

The output of a two-input EX-NOR gate is a logic '1' when the inputs are like and a logic '0' when they are unlike. In general, the output of a multiple-input EX-NOR logic function is a logic '0' when the number of 1s in the input sequence is odd and a logic '1' when the number of 1s in the input sequence is even including zero. That is, an all 0s input sequence also produces a logic '1' at the output

2.1.4 Universal Gate

A universal gate is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates. In practice, this is advantageous since NAND and NOR gates are economical and easier to fabricate and are the basic gates used in all IC digital logic families.

- NAND Gate:

The NAND gate represents the complement of the AND operation. Its name is an abbreviation of NOT AND. The graphic symbol for the NAND gate consists of an AND symbol with a bubble on the output, denoting that a complement operation is performed on the output of the AND gate.

X	Y	NAND
0	0	1
0	1	1
1	0	1
1	1	0

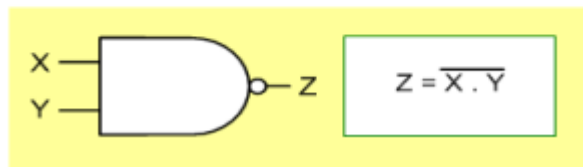


Figure 2.14 NANG Gate

- **NOR Gate**

The NOR gate represents the complement of the OR operation. Its name is an abbreviation of NOT OR. The graphic symbol for the NOR gate consists of an OR symbol with a bubble on the output, denoting that a complement operation is performed on the output of the OR gate.

X	Y	NOR
0	0	1
0	1	0
1	0	0
1	1	0

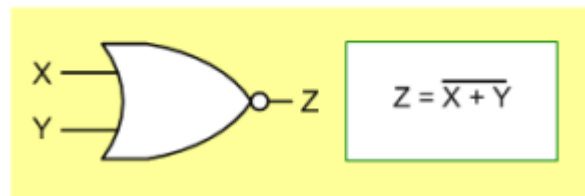


Figure 2.15 NOR Gate

2.2 Constructing logic gate on bread board

2.2.1 Constructing AND Gate using Transistors

Introduction

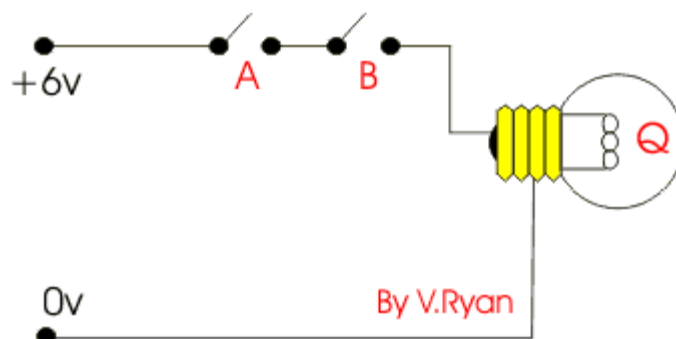
Integrated Circuit or IC is a combination of many small circuits in a small package which together performs a common task. Like an Operational Amplifier or 555 Timer IC is built by combination of many Transistors, Flip-Flops, Logic Gates and other combinational digital circuits. Similarly a Flip-Flop can be built by using a combination of Logic Gates and the Logic Gates itself can be built by using a few Transistors.

Logic Gates are the basics of a many digital electronic circuit. From the basic Flip-Flops to Microcontrollers Logic gates form the underlying principle on how bits are stored and processed. They state the relation between every input and output of a system using an Arithmetic logic. There are many different types of logic gates and each of them has a different logic which be used for different purposes.

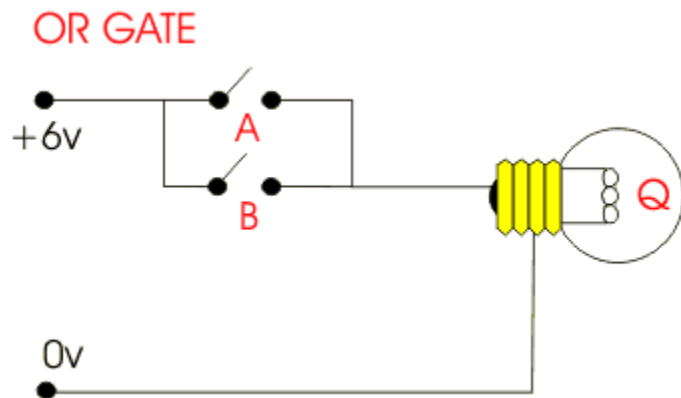
➤ Logic circuit with electrical diagram

LOGIC circuits are normally composed of 'gates'. A combination of gates make up a circuit and some digital circuits can be extremely complex. It is the logic gates that produce pulses of electrical current (1s and 0s). At school level, digital logic circuits are relatively simple. Below are simple drawings that help explain the two most popular logic gates - the AND gate and the OR gate.

AND GATE



The simplified AND gate shown above has two inputs, switch A and switch B. The bulb Q will only light if both switches are closed. This will allow current to flow through the bulb, illuminating the filament.



The simplified OR gate shown above has two inputs, switch A and switch B. The bulb Q will light if either switch A or B are closed. This will allow current to flow through the bulb, illuminating the filament.

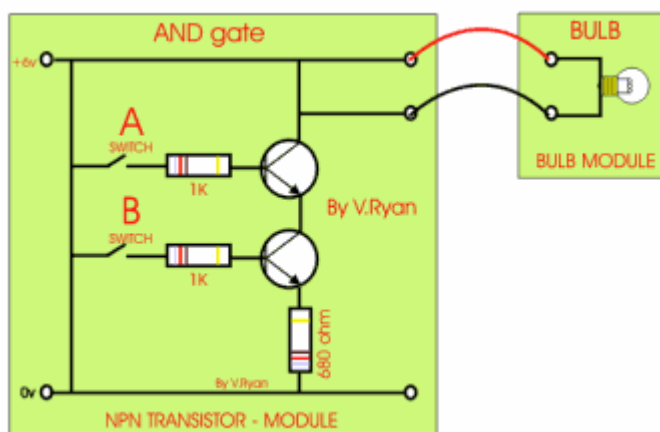
When the bulb lights this represents a '1' as current is running through the filament. If current is not running through the filament the bulb will not light and this represents a '0' (zero).

THE ROLE OF TRANSISTORS

Transistors are vital for digital circuits to work. These components are used as very fast switches in digital logic circuits. Transistors are normally so small that hundreds of thousands fit on one processing chip on a computer motherboard. The types of transistors used in school projects are normally large enough to fit on the end of a small finger. However, the way they switched on and off is the same

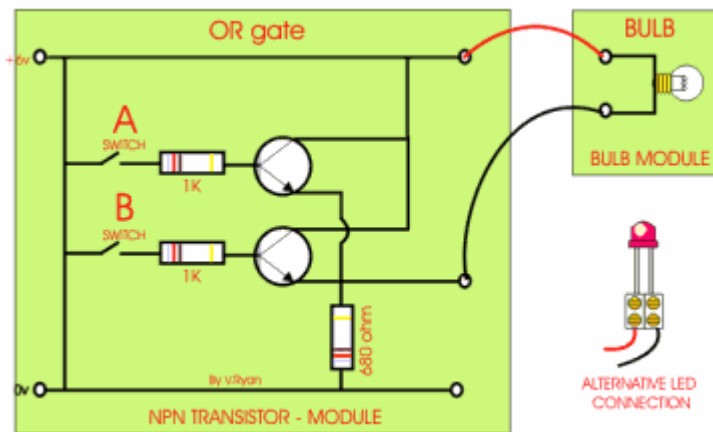
When a transistor is switched on it produces a '1' and when it is switched off it produces a '0'. Transistors in the circuit of a computer microprocessor can switch on and off thousands of times per second. Without the invention of the transistor, computer processing power would be very limited and slow.

Two basic examples of simple transistor driven logic (AND / OR) circuits are shown below.



This is an AND gate circuit and it can be made quite easily. The example shown is built from a modular electronics kit. Both switches 'A' and 'B' must be pressed together for the bulb to light.

If you construct this circuit, you may need to alter the value of the resistors. This will depend on the type of transistors used and whether to bulb or an LED is used.



This is an **OR** gate circuit. Either switch 'A' or 'B' must be pressed for the bulb to light. The switches do not have to be pressed together.

Transistor

A transistor is a semiconductor device with three terminals that can be connected to an external circuit. The device can be used as a switch and also as an amplifier to change the values or control the passing of an electrical signal.

For building an AND logic gate using a transistor we would be using BJT transistors which can be further classified into two types: PNP and NPN – Bipolar Junction Transistors. The circuit symbol for each of them can be seen below.

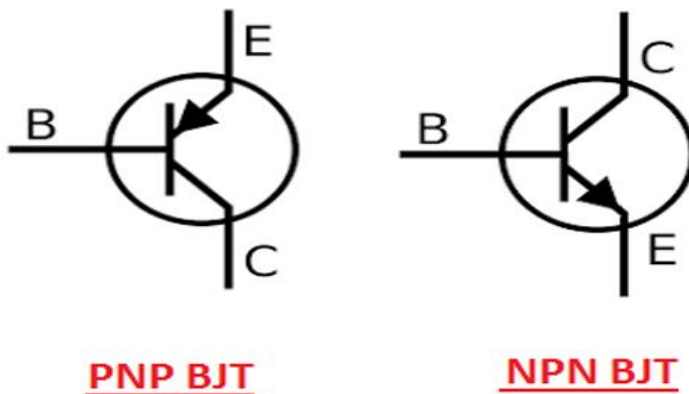


Figure 2.17 transistor

Circuit Diagram and Components Required

The list of components required to **build an AND gate using an NPN transistor** are listed as follows:

1. Two NPN transistors. (You can also use PNP transistor if available)

2. Two 10K Ω resistors & one 4-5K Ω resistor.
3. One LED (Light Emitting Diode) to check the output.
4. A Breadboard.
5. A +5V Power supply.
6. Two PUSH buttons.
7. Connecting Wires.

The circuit represents both the inputs A & B for the AND gate and Output, Q which also has a +5V supply to the collector of the first transistor which is connected in series to the second transistor and an LED is connected to the emitter terminal of the second transistor. The inputs A & B are connected to the base terminal of Transistor 1 and Transistor 2, respectively and the output Q goes to the positive terminal LED. The below diagram represents the above-explained circuit to build an AND gate using NPN Transistor.

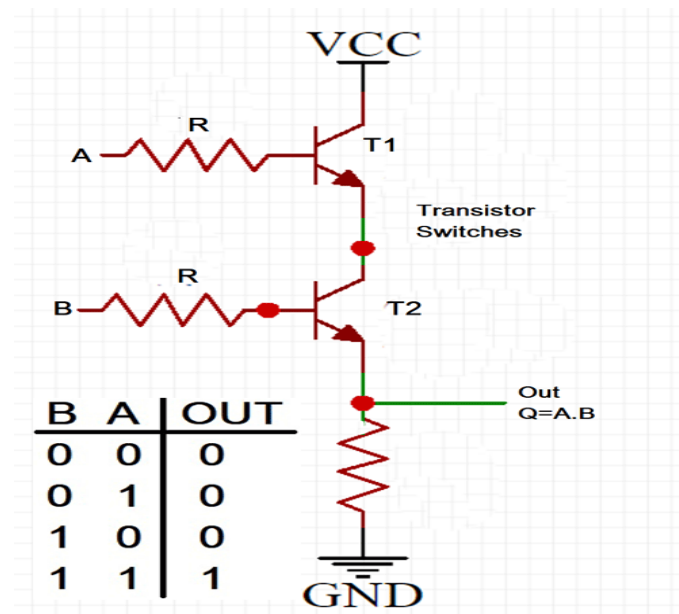


Figure 2.18 AND gate construction

The Transistors used in this tutorial are BC547 NPN Transistor and were added with all the above-mentioned components in the circuit, as shown below.

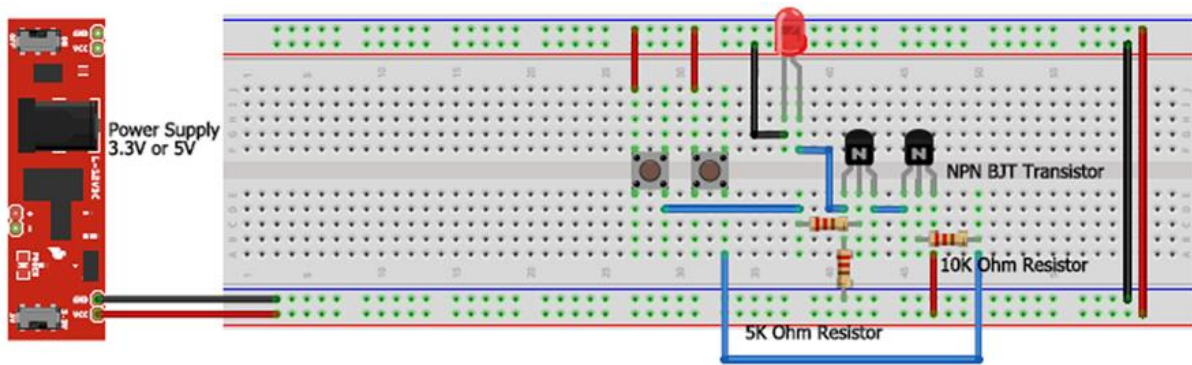


Figure 2.19 AND gate on breadboard

If you do not have the push buttons with you, you can also use wires as a switch by adding or removing them whenever required (instead of pressing the switch). The same could be seen in the video where I would use the wires as a switch connected to the base terminal for both the transistors.

The same circuit when built using the above mentioned hardware components, the circuit would look something like in the image below.

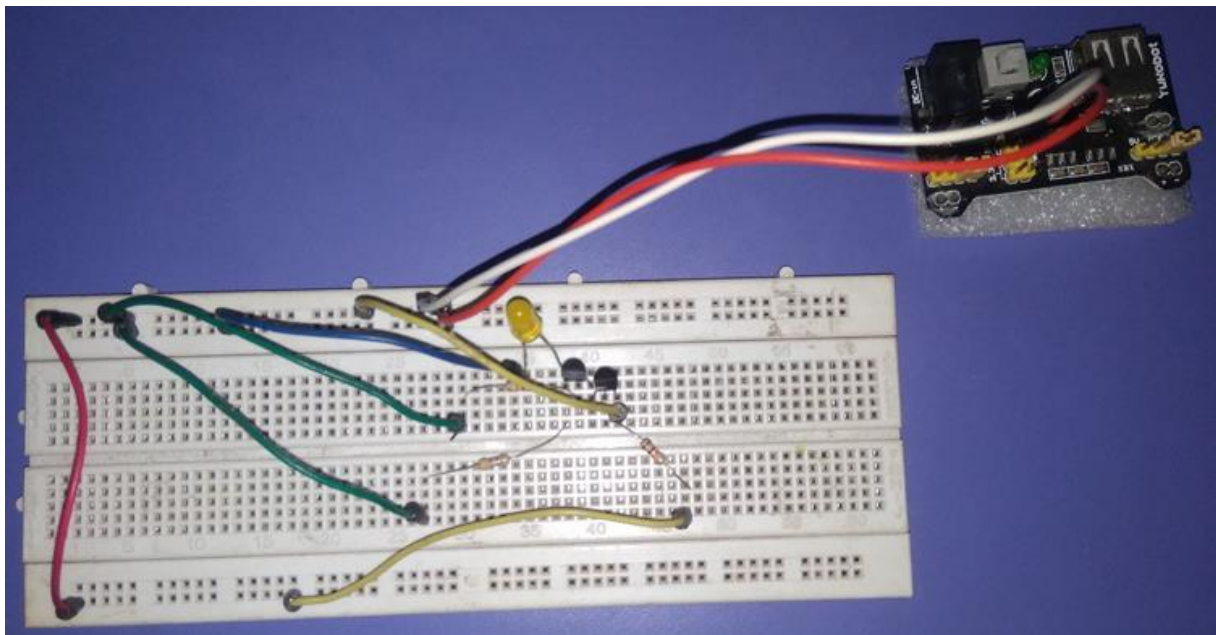


Figure 2.20 working of and Gate using Transistor

Here we will be using the transistor as a switch and so, when a voltage is applied through a Collector terminal of the NPN transistor, the voltage reaches the Emitter Junction only when the Base Junction is having a voltage supply between 0V and Collector Voltage.

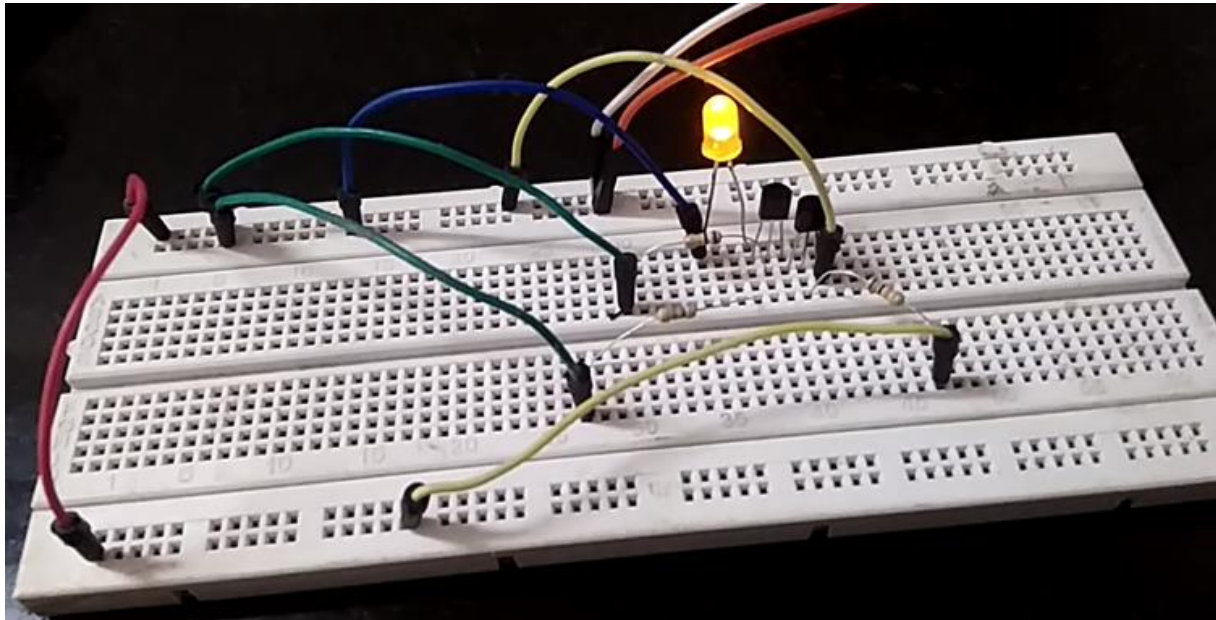


Figure 2.20 working of and Gate using Transistor

Similarly, the circuit above would make the LED glow i.e. the output is 1 (High) only when both the inputs are 1 (High) i.e. when there is a voltage supply at the base terminal of both the transistors. Meaning, there will be a straight line current path from VCC (+5V power supply) to the LED and further to the ground. Rest in all the cases, the output will be 0 (Low) and the LED will be OFF. These all can be explained in more detail by understanding each case one by one.

Case 1: When both inputs are zero – $A = 0$ & $B = 0$.

When both the inputs A & B are 0, you need not press any of the pushbuttons in this case. If you are not using the push buttons then remove the wires connected with, the pushbuttons and the base terminal of both the transistors. So, we got both inputs A & B as 0 and now we need to check for the output, which also should be 0 according to the AND gate truth table.

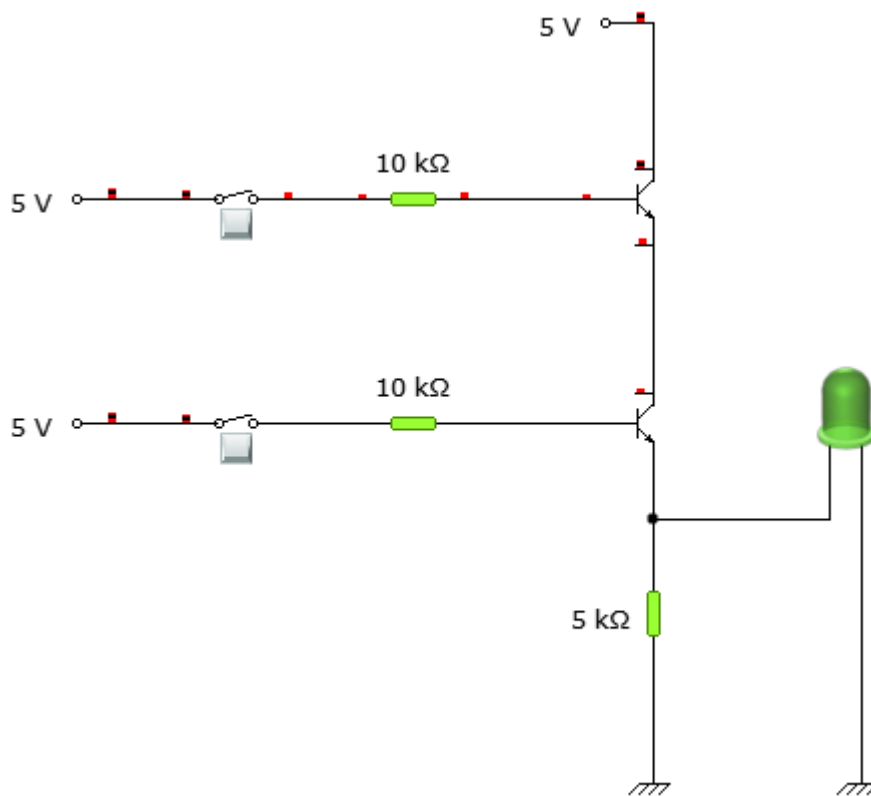


Figure 2.12 Block diagram of AND gate using transistor

Now, when a voltage is supplied through the collector terminal of Transistor 1, the emitter doesn't receive any input because the base terminal value is 0. Similarly, the emitter of transistor 1 which is connected to the collector of Transistor 2, supplies no current or voltage and also the base terminal value of transistor 2 is 0. So, the 2nd transistor's emitter outputs the value 0 and as a result, the LED would be OFF.

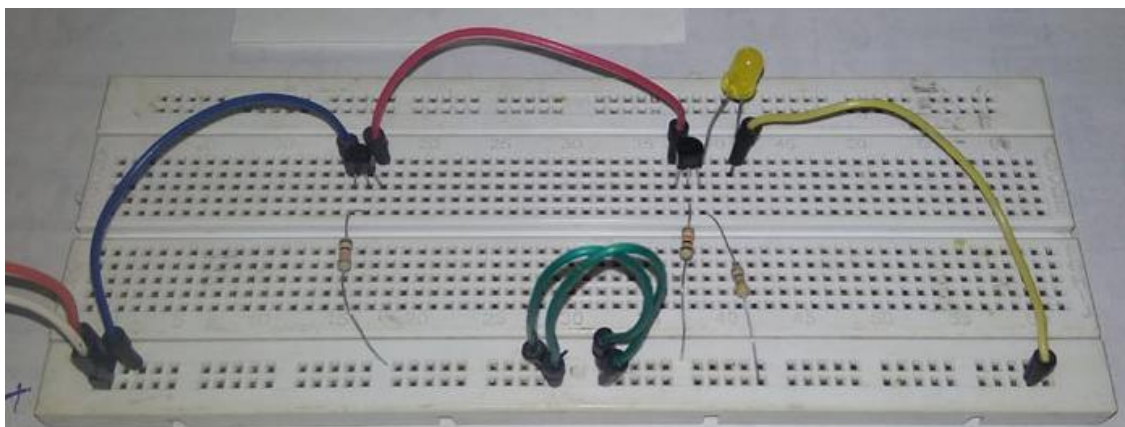


Figure 2.20 working of and Gate using Transistor

Case 2: When the inputs are – A = 0 & B = 1.

In the second case, when the inputs are $A = 0$ & $B = 1$, the circuit has first input as 0 (Low) and the second input as 1 (High) to the base of transistor 1 & 2, respectively. Now, when a 5V supply is passed to the collector of the first transistor, then there is no change in the phase shift of transistor since that the base terminal has 0 input. Which passes 0 value to the emitter and the emitter of first transistor is connected to the collector of the second transistor in series, so 0 value goes into the collector of second transistor.

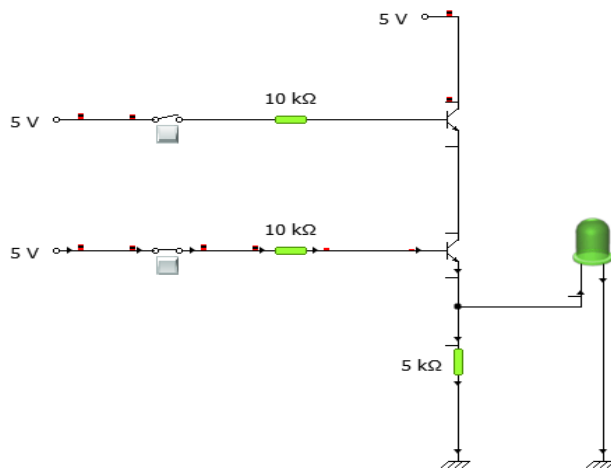


Figure 2.12 Block diagram of AND gate using transistor

Now, the second transistor has a high value in the base, so it would allow the same value received in the collector to pass to the emitter. But since the value is 0 in the collector terminal of the second transistor, that's why the emitter will also be 0 and the LED connected to the emitter would not glow.

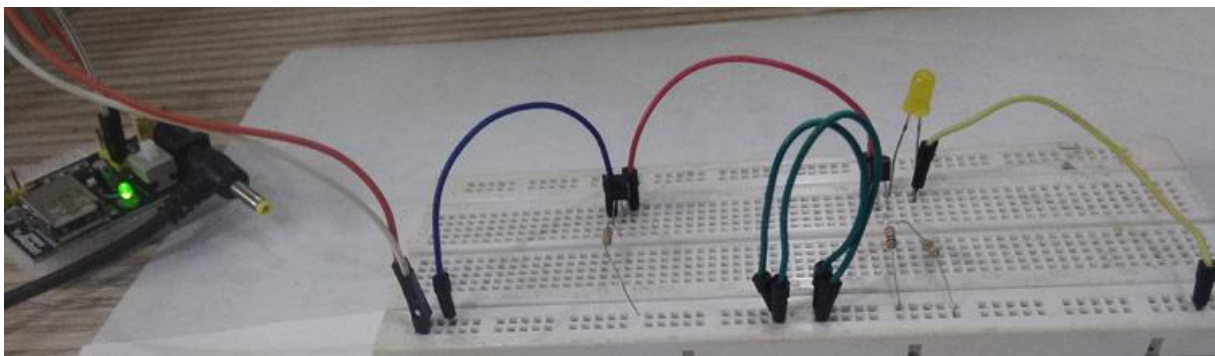


Figure 2.20 working of and Gate using Transistor

Case 3: When the inputs are – $A = 1$ & $B = 0$.

Here, the input is 1 (high) for the first transistor base and low for the second transistor base. So, the current path will start from 5V power supply to the collector of the second transistor passing through the collector and emitter of the first transistor since the base terminal value is high for the first transistor.

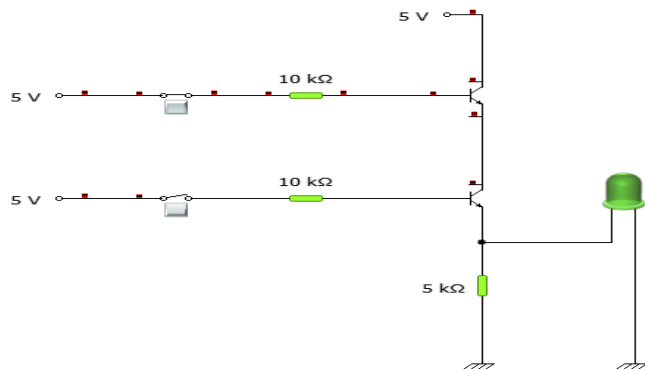


Figure 2.12 Block diagram of AND gate using transistor

But in the second transistor, the base terminal value is 0 and so, no current passes from collector to the emitter of the second transistor and as a result, the led would still be OFF only.

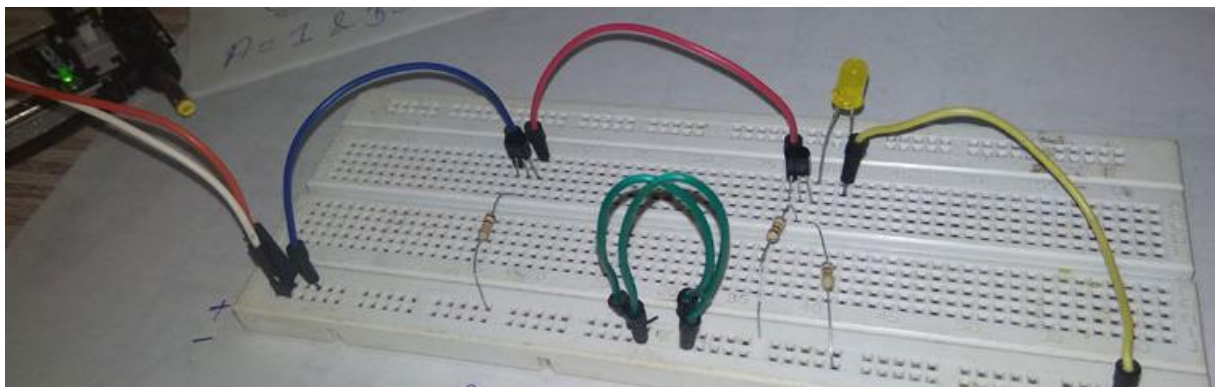


Figure 2.20 working of and Gate using Transistor

Case 4: When both inputs are one – $A = 1$ & $B = 1$.

The last case and here both the inputs are supposed to be high which are connected to the base terminals of both the transistors. This means whenever a current or voltage passes through the collector of both the transistors, the base reaches its saturation and the transistor conducts.

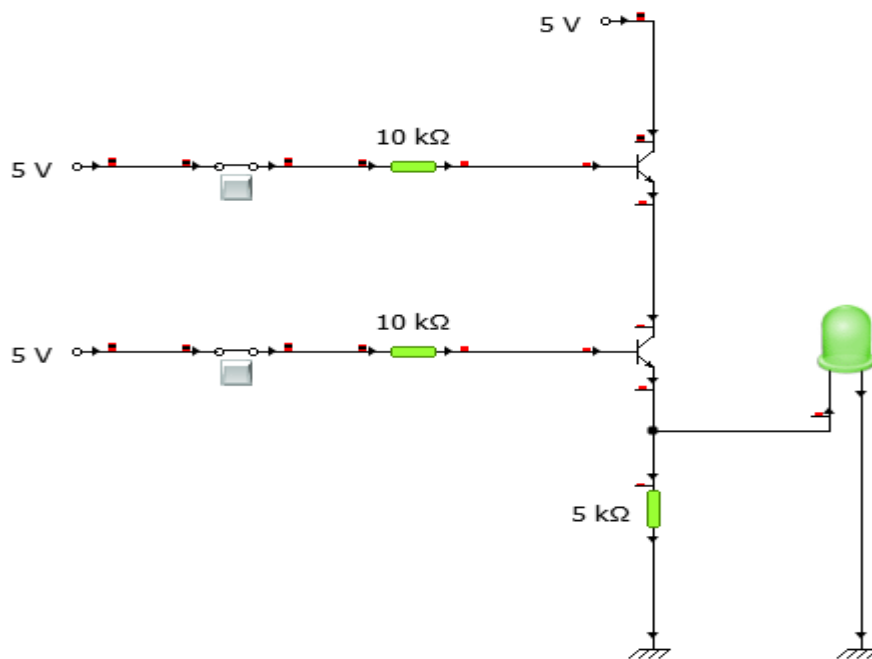


Figure 2.12 Block diagram of AND gate using transistor

Practically explaining, when a +5V supply is provided to the collector terminal of transistor 1 and also the base terminal is saturated then, the emitter terminal would receive a high output since the transistor is forward biased. This high output at the emitter goes directly to the collector of 2nd transistor through a series connection. Now, similarly at the second transistor, the input at the collector is high and in this case, the base terminal is also high, meaning the second transistor is also in a saturated state and the high input would pass from the collector to the emitter. This high output at the emitter goes to the LED which turns the LED ON.

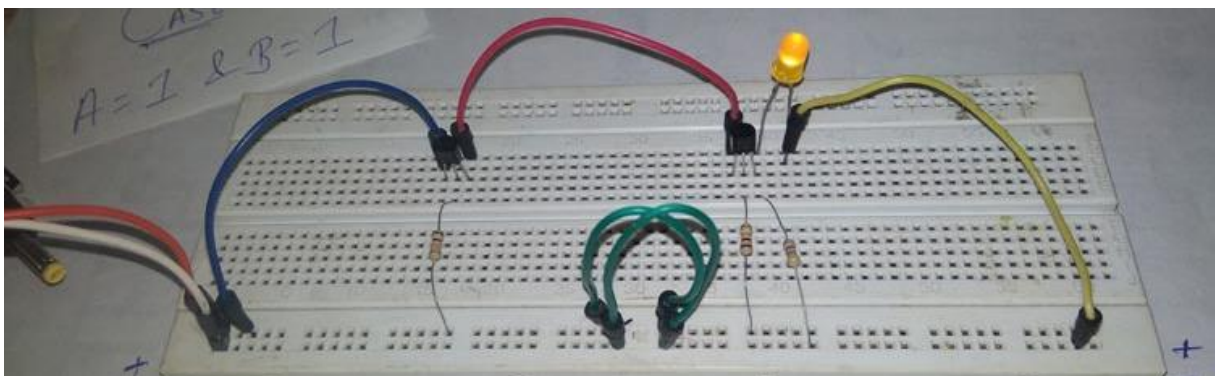


Figure 2.20 working of AND Gate using Transistor

2.3 Testing logic gates

When you make a project with logic circuit IC's it's very likely that you need a lot of them. Imagine a situation where one of them isn't working properly. You solder everything together and power it up but nothing happens. That is very disappointing. It's hard to find the reason for that but it can be even harder to replace that part then with an IC that is working. In the next few steps we will see you how this circuit is tested. We called it simple logic gate IC-tester.

Required materials

- 1x Circuit Board (I've used 5cm x 7cm)
- 2x Buttons
- 1x 28-pin socket
- male header connectors
- 4x 3mm green LED
- 4x 3mm blue LED
- 2x 1k ohm resistor
- 4x 680 ohm resistor
- 4x 470 ohm resistor
- wires
- soldering iron (or access to one)

Step 1: General Principle



Figure 2.21 logic gate IC

Basically, logic gates work by having two inputs and an output based on the state of the two inputs. In the left figure, inputs marked in yellow and the outputs is in blue. In the right figure the inputs yellow and the outputs green. If you start from the top left point of both IC's you see that there are two different arrangements of the pins. On one the output comes first and then the inputs and on the other it is the other way round.

What logic the IC's contain is therefore not so important. You wanted to test NOR-gates with the blue arrangement and AND-gates with the green one but every logic gate IC with the same pinout can be tested with it.

VCC (in our case 5V) is connected to the upper right and GND to the lower left. This is the same for most IC's.

Step 2: Circuit

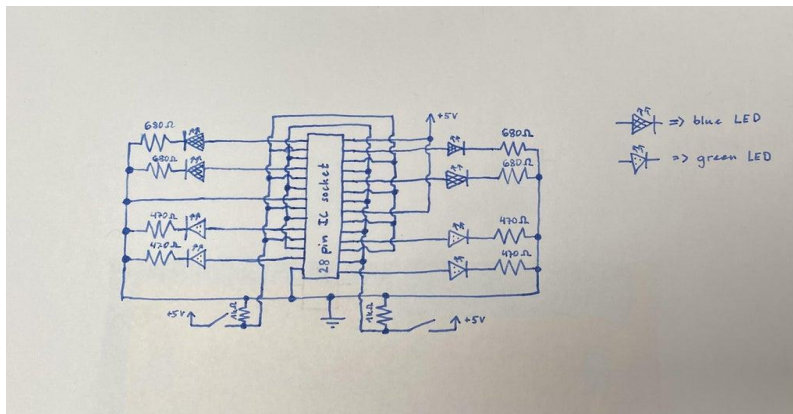


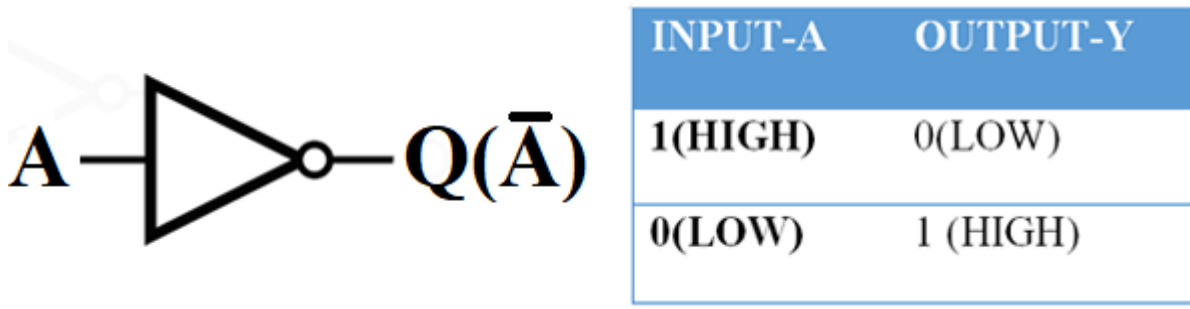
Figure of the circuit

The reasoning behind the principle was that it is easy to say that everything works if the LEDs all show the same behavior. So one button had to be connected to each input A and the other button had to be connected to each input B. The LED's are connected to the output accordingly and so that they are not damaged a series resistor is also necessary. For the blue LED lamps this needs 680 Ohm. With the green ones only one of 470 Ohm is necessary. So that the state of the buttons is clear, the 1k Ohm resistor pulls the inputs of the IC's in digital language to 0. For the current connection there are the two male header pins, to which one can connect jumper wires. Lead them to a breadboard on which is a voltage regulator. So you don't limit the connection only to batteries or a laptop connection.

Now you just have to solder everything together and you can use it. It is important to know that components can be damaged if you insert them incorrectly and it is also a big risk to insert the components during operation, because then you have less time to check if everything is inserted correctly. Also, only one IC can be plugged in at a time, otherwise the cases would overlap.

➤ Constructing electronic circuit using NOT Gate by transistor

NOT gate is the simplest gate when compared to the remaining digital logic gates. The **NOT gate symbol** is shown below, along with the **NOT gate truth table**. It has one input and one output.



The **NOT Gate Boolean equation** can be written as $Y = \bar{A}$, its output will be low when the input is high, and the output will be high when the input is low.

Transistor – Basics, and Working

We are going to learn about transistors as we are going to build a NOT gate using BC547, which is an NPN transistor. A transistor is a back to back connection of a diode. A diode is a semiconductor device, which is doped with impurities to make it either a p-type or n-type depending on the types of impurities used in doping. When these diodes are connected in back to the back connection, they form a transistor. Depending on which both sides are connected, the transistors are of two types namely [NPN Transistor](#) and [PNP Transistor](#).

The difference in circuit wise is that when connecting supply terminals, PNP transistor's emitter terminal is connected with the positive terminal, and for NPN transistor, the positive terminal is given to the collector terminal. From now on, the topic will be discussed based on only the NPN transistor.

Case 1: When base voltage is less than the emitter voltage, the flow of electrons from emitter to collector is blocked by the PN junction (this current is electric current which flows from negative terminal to positive terminal while convention current flows from positive terminal to negative terminal) as it is now acting in reverse bias.

Case 2: When the base voltage is greater than the emitter voltage ($V_b > 0.6v$), the junction gets reduced, and this allows the flow of current from the emitter terminal to the collector terminal. The transistor must work in a saturation region as they provide a low voltage drop in the saturation region.

Circuit Diagram

The circuit for NOT gate using a transistor is given below. The circuit was designed and

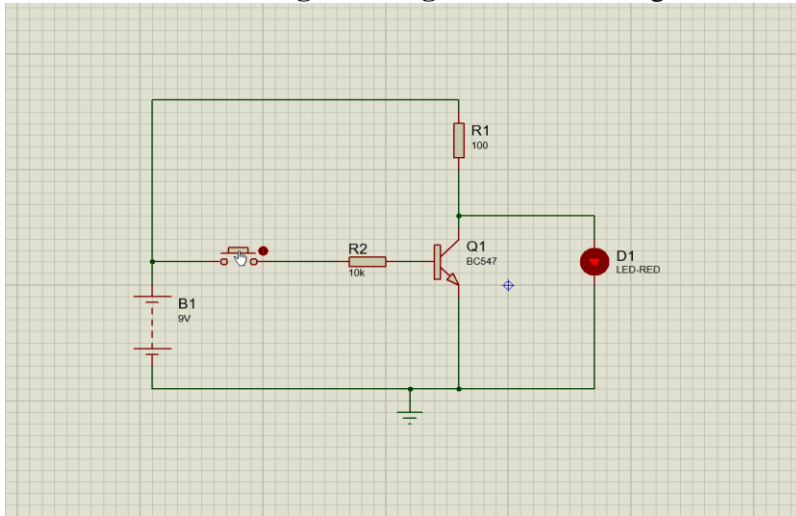


Figure: circuit diagram

Figure: simulated using the Proteus software.

I took supply voltage as 9V, and I want to send 9mA to led, so I used 100 ohms to limit the current. This same current has to flow in the transistor $I_c = 9\text{mA}$. The h_{fe} of the transistor is 100, so I_b value should be 0.09mA. As the I_b is 0.09mA, the base resistor value should be 10k ohms.

Case 1:-

When the switch is in an off state, the current to the base is zero and transistor acts as open circuit due to these current flows in the LED direction and led starts glowing.

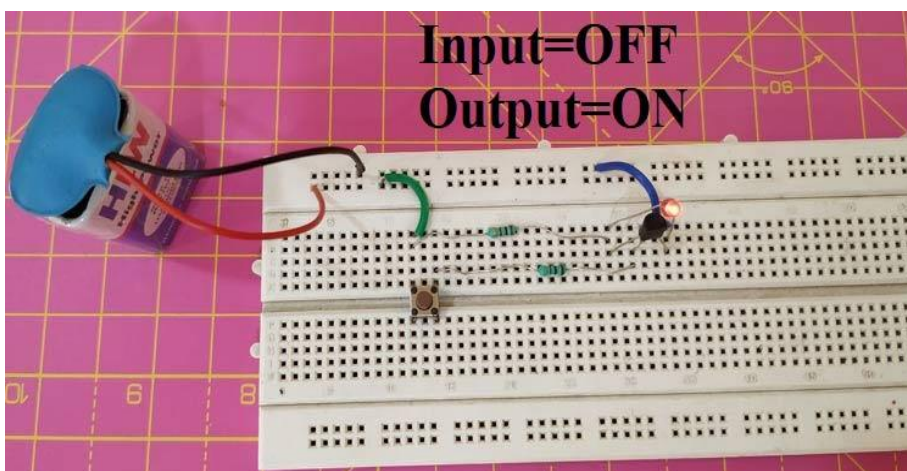


Figure: construction on bread board

Case 2:- When the switch is in the ON state, the current to the base starts flowing, and this makes the transistor act as short circuit, and as the current choose the lowest resistance, which is now provided by the transistor will flow in that path and LED will be switched OFF.

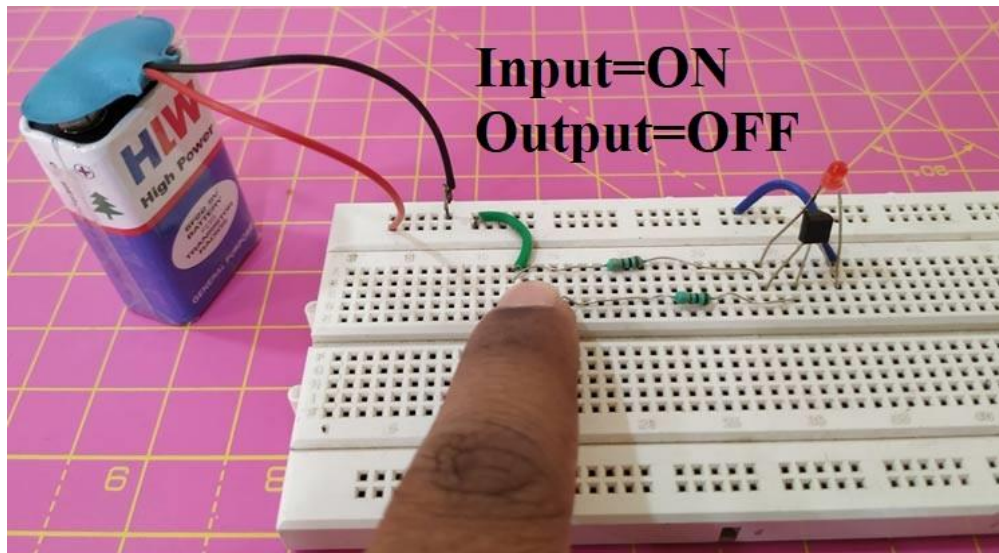


Figure: construction on bread board

Hence, both cases have the same inputs and outputs following the NOT gate truth table. Thus, we have built a **NOT Logic gate using a Transistor**. Hope you understood the tutorial and enjoyed learning something new. The complete working of the set-up can be found in the **video** below. If you have any questions leave them in the comment section below or use our forums for other technical questions.

Self-check 2

Direction I True /False

1. A truth table lists all possible combinations of input binary variables and the corresponding outputs of a logic system
2. The output of an AND gate is HIGH only when all of its inputs are in the HIGH state
3. A universal gate is a gate which can implement any Boolean function with need to use any other gate type
4. The NAND gate represents the complement of the NOR operation
5. A transistor is a semiconductor device with three terminals that can be connected to an external circuit.

Direction II multiple choice

- 1). the logical gates are categorized into _____

- A. One group
 - B. Two groups
 - C. Three groups
 - D. Four groups
2. _____ are universal gates
 - A. NOT
 - B. NAND & NOR
 - C. AND
 - D. NOT, AND, & OR
 3. The base is 16 for _____ number system
 - A. Binary
 - B. Hexadecimal
 - C. Decimal
 - D. Octal
 4. The inverter is
 - A. NOT gate
 - B. OR gate
 - C. AND gate
 - D. None of the above
 5. The inputs of a NAND gate are connected together. The resulting circuit is
 - A. OR gate
 - B. AND gate
 - C. NOT gate
 - D. None of the above

Direction III short answer

1. Explain logic gate with their truth table

Operation sheet-1

Operation Title: construct and test an **AND gate** using an **NPN transistor**

Instruction: In workshop, construct the given digital electronic components on printed circuit.

Purpose: To construct and test given logic gate on bread board

Required tools and equipment:

41	<u>Author/Copyright</u> Ministry of Labor and Skills	Module title: Test Basic Digital Electronic Components	Training module Version -1 August , 2022
----	---	--	--

- Two NPN transistors. (You can also use PNP transistor if available)
- Two 10K Ω resistors & one 4-5K Ω resistor.
- One LED (Light Emitting Diode) to check the output.
- A Breadboard.
- A +5V Power supply.
- Two PUSH buttons.
- Connecting Wires.

Precautions:

- Careful when you touch the component because could be easily fragile
- Hold wires to be heated with tweezers or clamps.
- Always choose the correct terminal of transistor
- Never supply the circuit from main line
- Turn unit off and unplug when not in use.

Procedures:

Step 1: Read the given circuit diagram carefully

Step 2: check the continuity of bread board

Step 3. Place AND gate on the bread board and other listed component

Step 4. check the continuity of the constructed circuit by using digital multimeter

Step 5. supply your circuit with +5V

Step 6. observe the LED by each push button separately

Step 7. Follow the truth table of AND gate and observe and confirm the output when input is changed

Step 8. After finished disassemble the circuit and return the component back to place

Step 9. clean working area

Quality criteria:

- Proper component selection
- correct sequence of the operation
- Proper handling of component

LAP Test	Practical Demonstration
----------	-------------------------

Name: _____

Date: _____

Time started: _____

Time finished: _____

Instruction I: Given necessary templates, tools and materials you are required to perform the following tasks within 2 hours.

Task 1: construct and test given logic gate on bread board

Unit Three: Basic Digital Electronic Circuits

This learning unit is developed to provide the trainees the necessary information regarding the following content coverage and topics:

- Principles of Combinational logic gate
- Basic digital electronic circuits
- Applications of digital electronic circuits.
- Constructing and testing Basic digital circuits

This unit will also assist you to attain the unit stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Demonstrate Principles of Combinational logic gate
- Identify Basic digital electronic circuits
- Demonstrate Applications of digital electronic circuits.
- Construct and test Basic digital circuits are

3.1 Principles of Combinational logic gate

Combinational circuit is a circuit in which we combine the different gates in the circuit, for Example encoder, decoder, multiplexer and DE multiplexer. Some of the characteristics of combinational circuits are following –

- The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- The combinational circuit do not use any memory. The previous state of input does not have any effect on the present state of the circuit.
- A combinational circuit can have an n number of inputs and m number of outputs.

Block diagram



Figure 3.1 combinational circuit

We're going to elaborate few important combinational circuits as follows.

Half Adder

Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary number A and B. It is the basic building block for addition of two **single** bit numbers. This circuit has two outputs **carry** and **sum**.

Block diagram



Truth Table

Inputs		Output	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Circuit Diagram

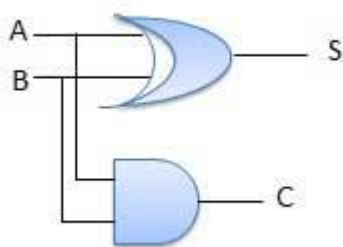
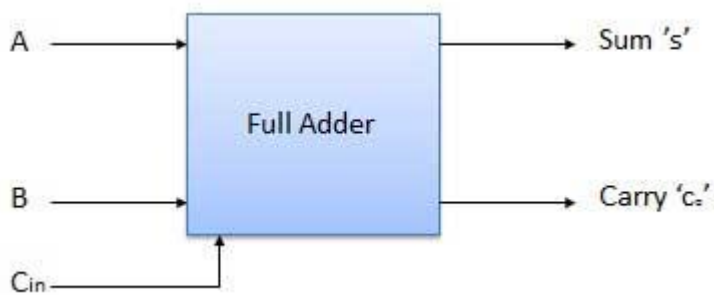


Figure 3.2 half adder

Full Adder

Full adder is developed to overcome the drawback of Half Adder circuit. It can add two one-bit numbers A and B, and carry c. The full adder is a three input and two output combinational circuit.

Block diagram



Truth Table

Inputs			Output	
A	B	C _{in}	S	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Circuit Diagram

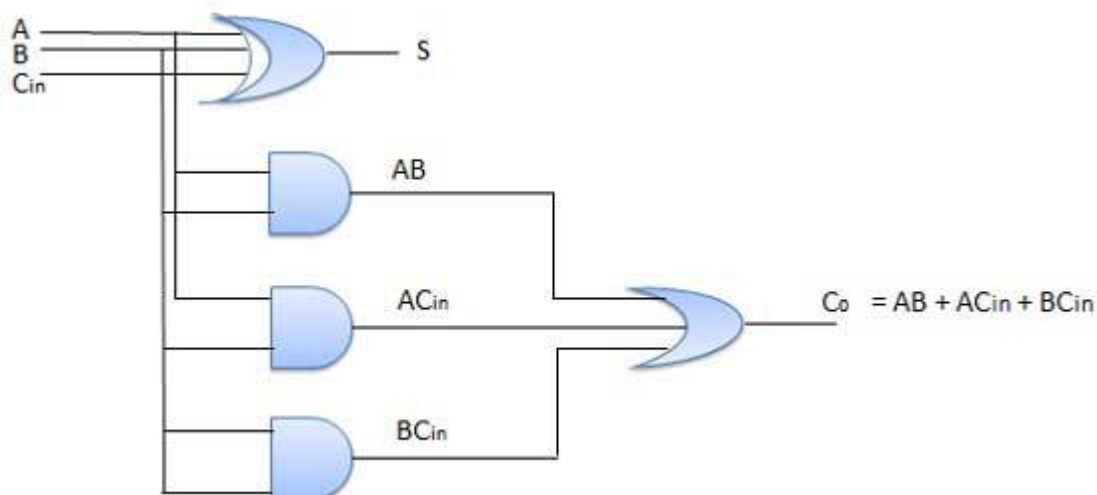


Figure 3.3 Full Adder

N-Bit Parallel Adder

The Full Adder is capable of adding only two single digit binary number along with a carry input. But in practical we need to add binary numbers which are much longer than just one bit. To add two n-bit binary numbers we need to use the n-bit parallel adder. It uses a number of full adders in cascade. The carry output of the previous full adder is connected to carry input of the next full adder.

4 Bit Parallel Adder

In the block diagram, A₀ and B₀ represent the LSB of the four bit words A and B. Hence Full Adder-0 is the lowest stage. Hence its C_{in} has been permanently made 0. The rest of the

connections are exactly same as those of n-bit parallel adder is shown in fig. The four bit parallel adder is a very common logic circuit.

Block diagram

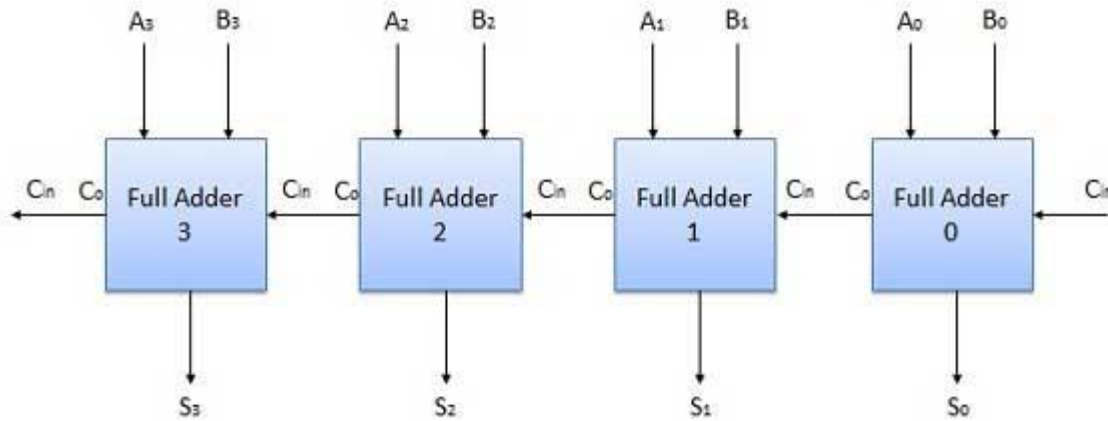


Figure 3.4 4 Bit Parallel Adder

N-Bit Parallel Subtractor

The subtraction can be carried out by taking the 1's or 2's complement of the number to be subtracted. For example we can perform the subtraction (A-B) by adding either 1's or 2's complement of B to A. That means we can use a binary adder to perform the binary subtraction.

4 Bit Parallel Subtractor

The number to be subtracted (B) is first passed through inverters to obtain its 1's complement. The 4-bit adder then adds A and 2's complement of B to produce the subtraction.

$S_3 S_2 S_1 S_0$ represents the result of binary subtraction (A-B) and carry output C_{out} represents the polarity of the result. If $A > B$ then $C_{out} = 0$ and the result of binary form (A-B) then $C_{out} = 1$ and the result is in the 2's complement form.

Block diagram

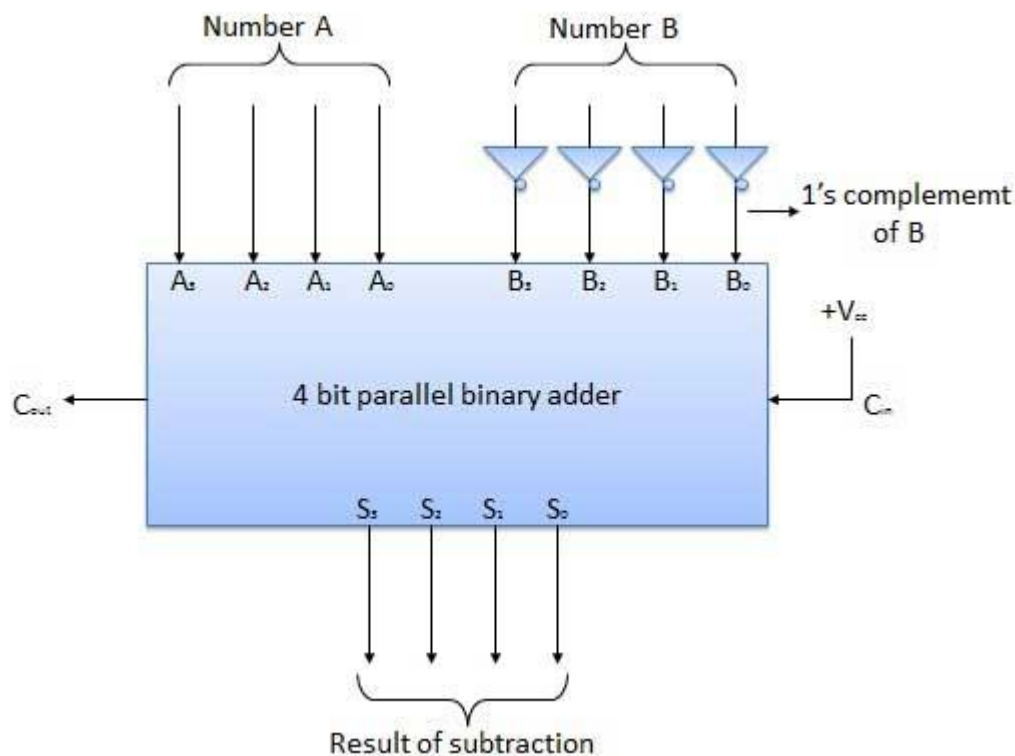


Figure 3.5 4 Bit Parallel Subtractor

Half Subtractors

Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow). It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called as Minuend bit and B is called as Subtrahend bit.

Truth Table

Inputs		Output	
A	B	(A - B)	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Circuit Diagram

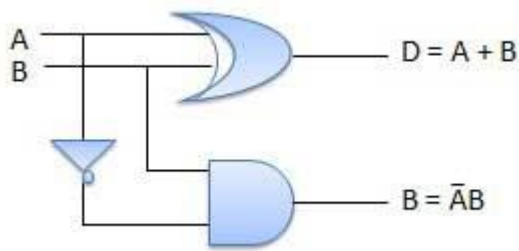


Figure 3.6 Half Subtractors

Full Subtractors

The disadvantage of a half subtractor is overcome by full subtractor. The full subtractor is a combinational circuit with three inputs A,B,C and two output D and C'. A is the 'minuend', B is 'subtrahend', C is the 'borrow' produced by the previous stage, D is the difference output and C' is the borrow output.

Truth Table

Inputs			Output	
A	B	C	(A-B-C)	C'
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Circuit Diagram

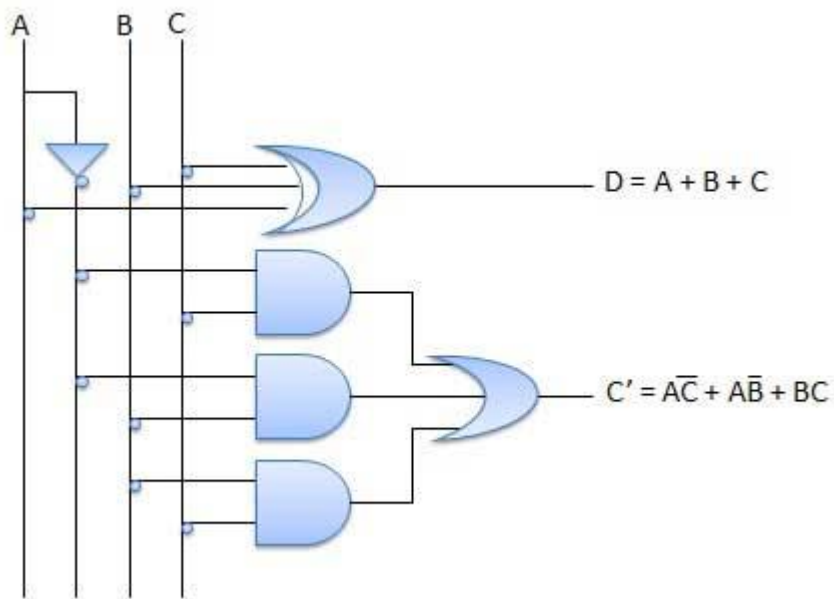
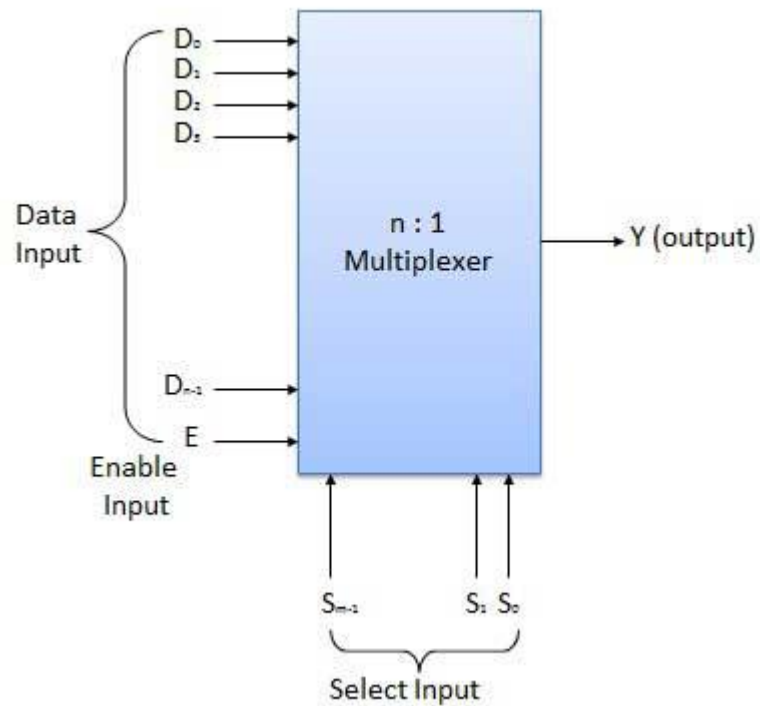


Figure 3.7 Full Subtractors

Multiplexers

Multiplexer is a special type of combinational circuit. There are n-data inputs, one output and m select inputs with $2^m = n$. It is a digital circuit which selects one of the n data inputs and routes it to the output. The selection of one of the n inputs is done by the selected inputs. Depending on the digital code applied at the selected inputs, one out of n data sources is selected and transmitted to the single output Y. E is called the strobe or enable input which is useful for the cascading. It is generally an active low terminal that means it will perform the required operation when it is low.

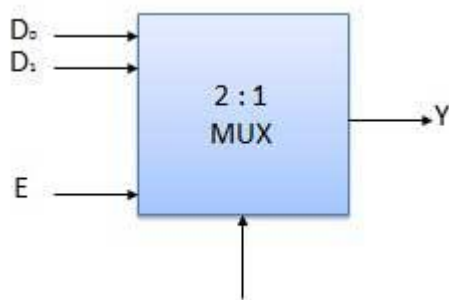
Block diagram



Multiplexers come in multiple variations

- 2 : 1 multiplexer
- 4 : 1 multiplexer
- 16 : 1 multiplexer
- 32 : 1 multiplexer

Block Diagram



Truth Table

Enable	Select	Output
E	S	Y
0	x	0
1	0	D ₀
1	1	D ₁

x = Don't care

Figure 3.8 Multiplexers

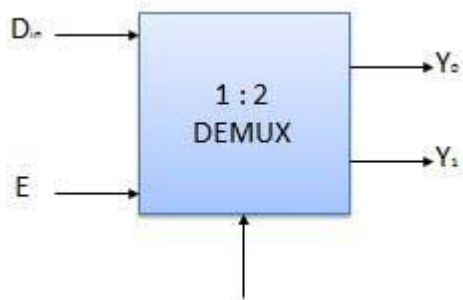
Demultiplexers

A demultiplexer performs the reverse operation of a multiplexer i.e. it receives one input and distributes it over several outputs. It has only one input, n outputs, m select input. At a time only one output line is selected by the select lines and the input is transmitted to the selected output line. A de-multiplexer is equivalent to a single pole multiple way switch as shown in fig.

Demultiplexers comes in multiple variations.

- 1 : 2 demultiplexer
- 1 : 4 demultiplexer
- 1 : 16 demultiplexer
- 1 : 32 demultiplexer

Block diagram



Truth Table

Enable	Select	Output
E	S	Y0 Y1
0	x	0 0
1	0	0 D _{in}
1	1	D _{in} 0

x = Don't care

Figure 3.9 Demultiplexers

Decoder

A decoder is a combinational circuit. It has n input and to a maximum $m = 2^n$ outputs.

Decoder is identical to a demultiplexer without any data input. It performs operations which are exactly opposite to those of an encoder.

Block diagram

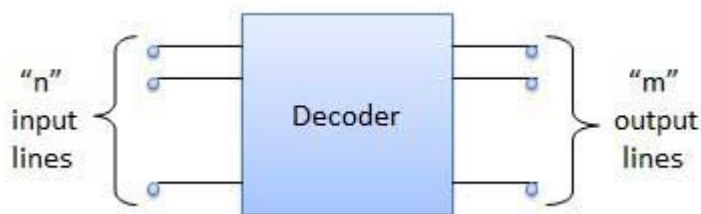


Figure 3.10 Decoder

Examples of Decoders are following.

- Code converters
- BCD to seven segment decoders
- Nixie tube decoders
- Relay actuator

3.2 Basic digital electronic circuits

Digital circuit: A digital circuit is a circuit where the signal must be one of two discrete levels.

Each level is interpreted as one of two different states (for example, on/off, 0/1, true/false).

Digital circuits use transistors to create logic gates in order to perform Boolean logic. This

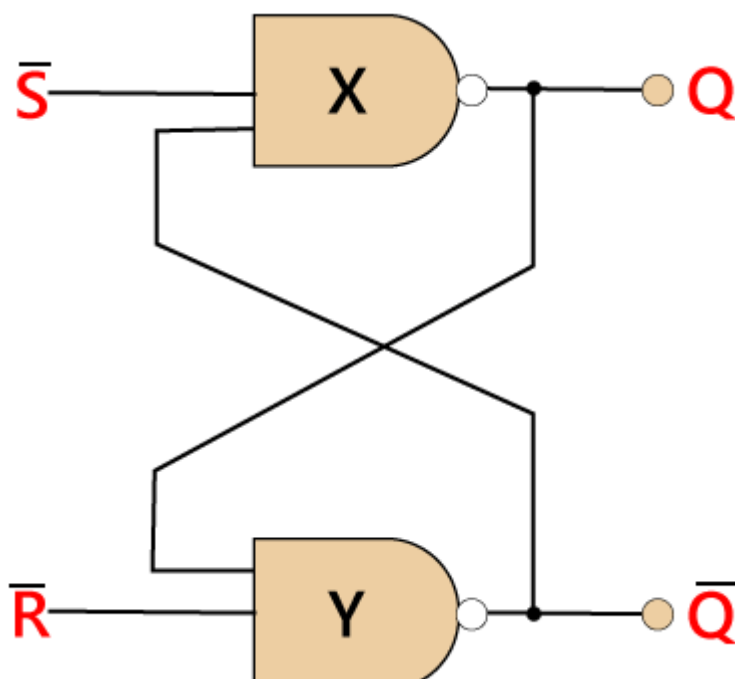
logic is the foundation of digital electronics and computer processing. Digital circuits are less susceptible to noise or degradation in quality than analog circuits. It is also easier to perform error detection and correction with digital signals. To automate the process of designing digital circuits, engineers use electronic design automation (EDA) tools, a type of software that optimizes the logic in a digital circuit.

3.2.1 Basics of Flip Flop

A circuit that has two stable states is treated as a **flip flop**. These stable states are used to store binary data that can be changed by applying varying inputs. The flip flops are the fundamental building blocks of the digital system. Flip flops and latches are examples of data storage elements. In the sequential logical circuit, the flip flop is the basic storage element. The latches and flip flops are the basic storage elements but different in working. There are the following types of flip flops:

- **SR Flip Flop**

It is the most common flip flop used in the digital system. In SR flip flop, when the set input "S" is true, the output Y will be high, and Y' will be low. It is required that the wiring of the circuit is maintained when the outputs are established. We maintain the wiring until set or reset input goes high, or power is shutdown.



J	K	Y	Y'
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
0	0	1	1
0	1	1	0
1	0	1	1
1	1	1	0

Table 3.13 truth table of J-K Flip-flop

- D Flip Flop

D flip flop is a widely used flip flop in digital systems. The D flip flop is mostly used in shift-registers, counters, and input synchronization.

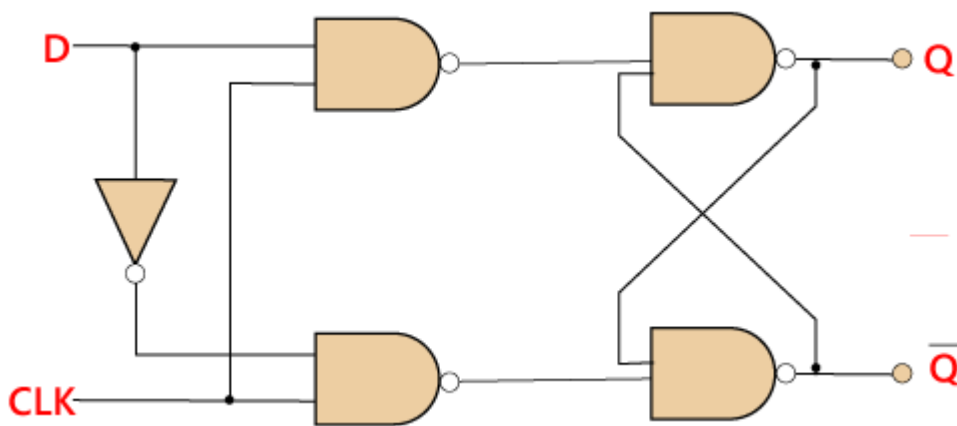


Figure 3.14 D-flip flop

Truth Table:

Clock	D	Y	Y'
↓ » 0	0	0	1
↑ » 1	0	0	1
↓ » 0	1	0	1
↑ » 1	1	1	0

Table 3.3 truth table of D-flip flop

- T Flip Flop

Just like JK flip-flop, T flip flop is used. Unlike JK flip flop, in T flip flop, there is only single input with the clock input. The T flip flop is constructed by connecting both of the inputs of JK flip flop together as a single input.

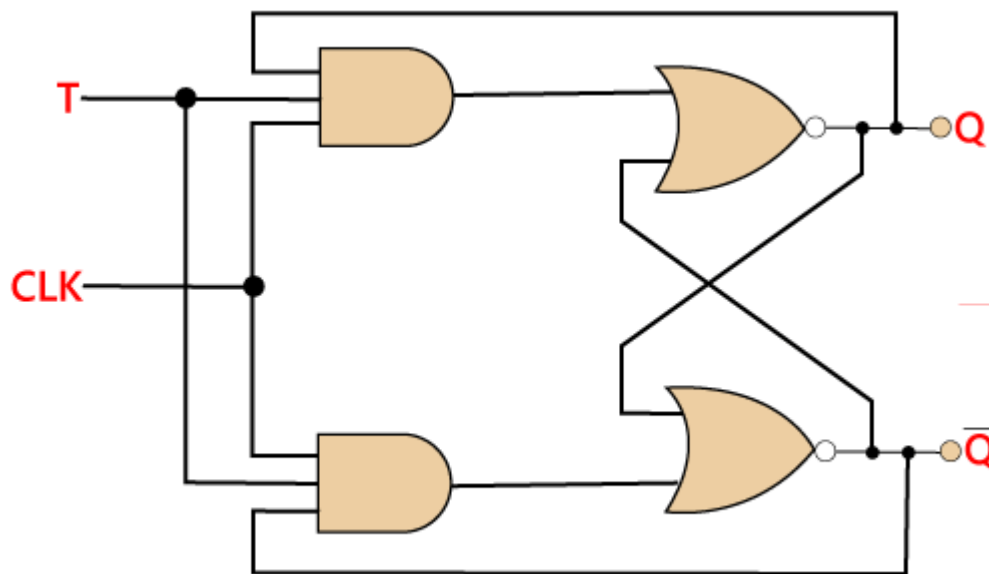


Figure 3.15 T-flip flop

The T flip flop is also known as **Toggle flip-flop**. These T flip-flops are able to find the complement of its state.

Truth Table:

T	Y	Y (t+1)
0	0	0
1	0	1
0	1	1
1	1	0

Table 3.4 truth table of T-flip-flop

3.2.2 Counters

A special type of sequential circuit used to count the pulse is known as a counter, or a collection of flip flops where the clock signal is applied is known as counters.

The counter is one of the widest applications of the flip flop. Based on the clock pulse, the output of the counter contains a predefined state. The number of the pulse can be counted using the output of the counter.

Truth Table

Counters

Clock	Counter output		State number	Decimal counter output
	Q _B	Q _A		
Initially	0	0	-	0
1 st	0	1	1	1
2 nd	1	0	2	2
3 rd	1	1	3	3
4 th	0	0	4	0

Table 3.5 counter truth table

There are the following types of counters:

- Asynchronous Counters
- Synchronous Counters

Asynchronous or ripple counters

The Asynchronous counter is also known as the ripple counter. Below is a diagram of the 2-bit Asynchronous counter in which we used two T flip-flops. Apart from the T flip flop, we can also use the JK flip flop by setting both of the inputs to 1 permanently. The external clock pass to the clock input of the first flip flop, i.e., FF-A and its output, i.e., is passed to clock input of the next flip flop, i.e., FF-B.

Block Diagram

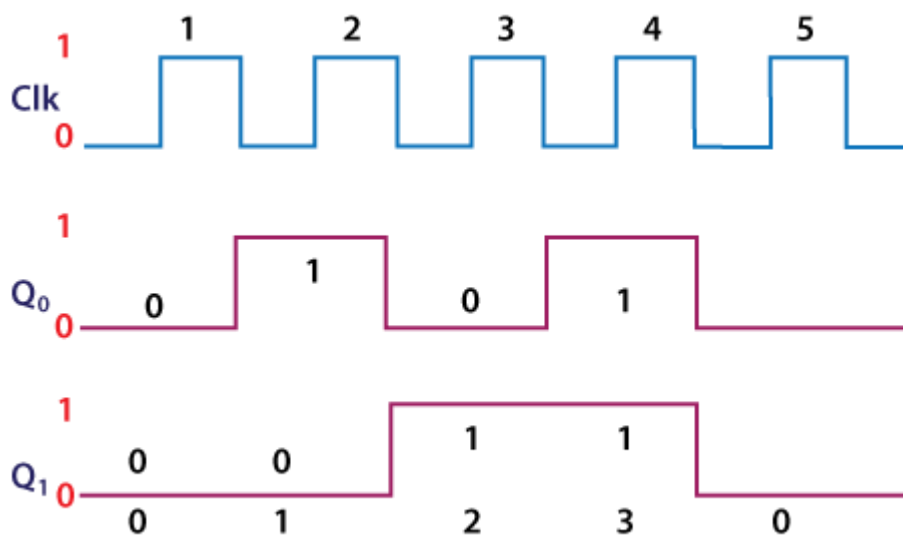
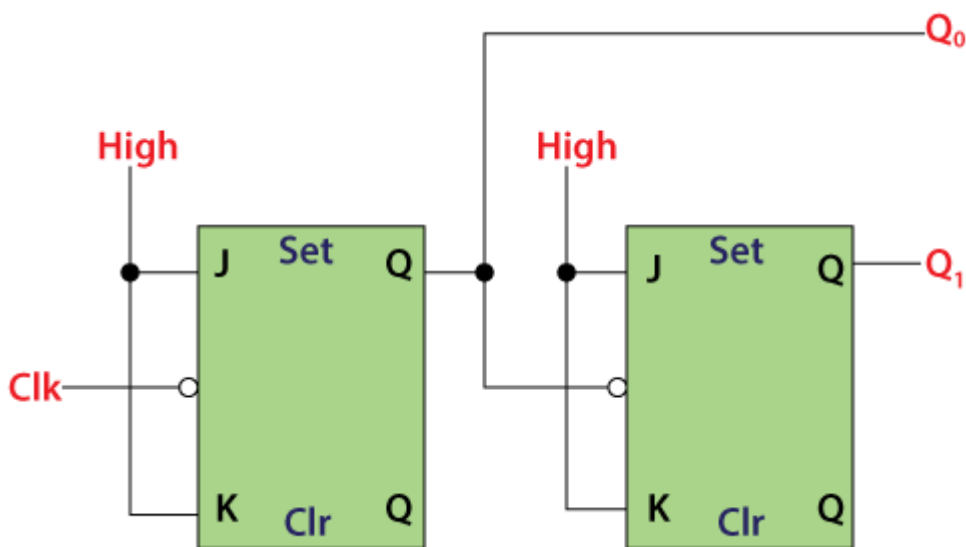
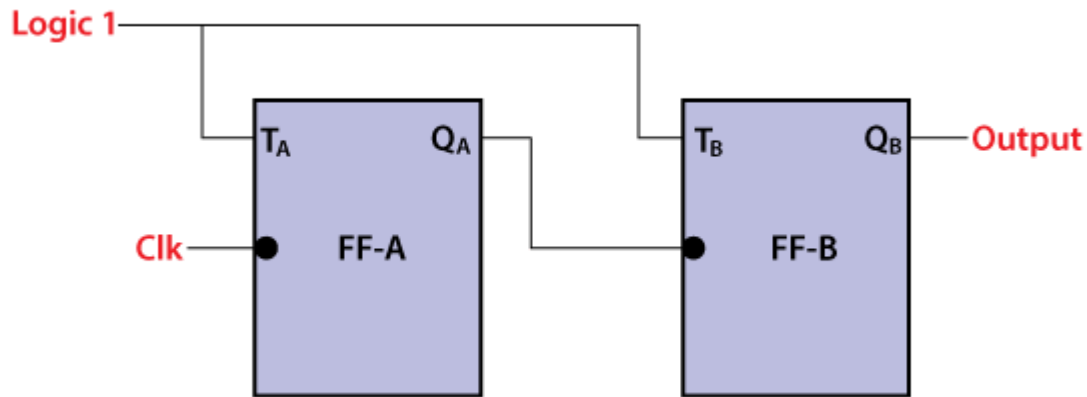


Figure 3.16 Signal Diagram

- **Synchronous counters**

In the **Asynchronous counter**, the present counter's output passes to the input of the next counter. So, the counters are connected like a chain. The drawback of this system is that it creates the counting delay, and the propagation delay also occurs during the counting stage.

The **synchronous counter** is designed to remove this drawback.

In the **synchronous counter**, the same clock pulse is passed to the clock input of all the flip flops. The clock signals produced by all the flip flops are the same as each other. Below is the diagram of a 2-bit synchronous counter in which the inputs of the first flip flop, i.e., FF-A, are set to 1. So, the first flip flop will work as a toggle flip-flop. The output of the first flip flop is passed to both the inputs of the next JK flip flop.

Logical Diagram

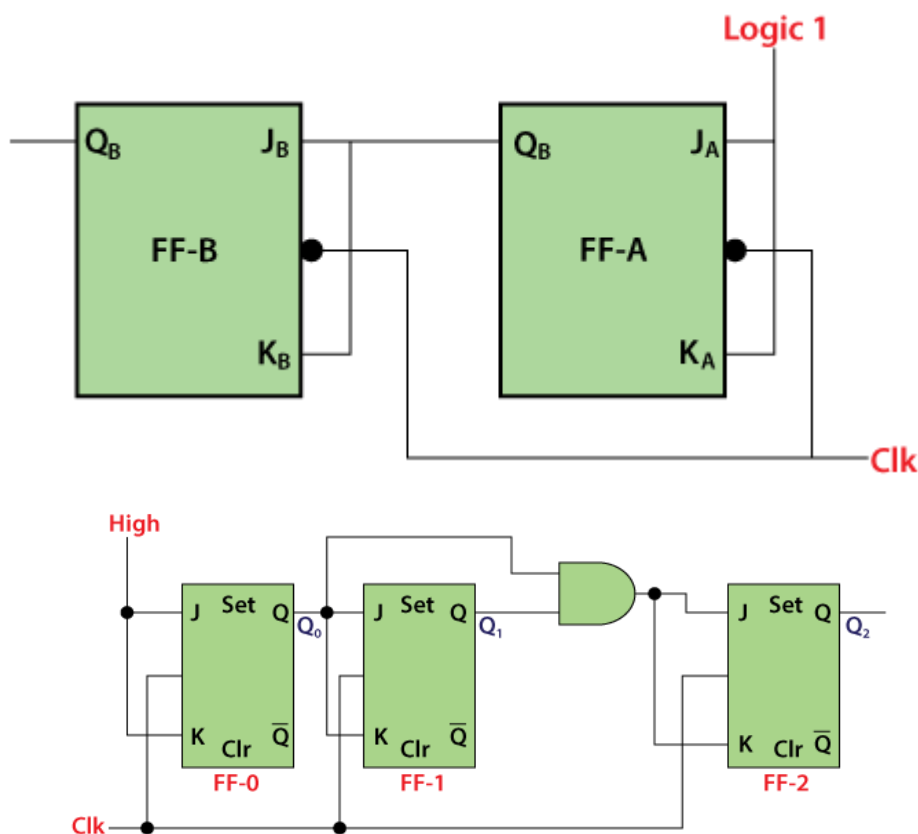


Figure 3.17 Logical Diagram

Signal Diagram

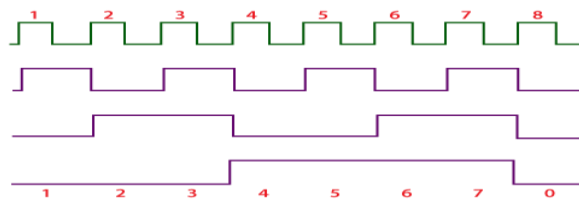


Figure 3.18 Signal Diagram

3.3.3 Registers

A **Register** is a collection of flip flops. A flip flop is used to store single bit digital data. For storing a large number of bits, the storage capacity is increased by grouping more than one flip flops. If we want to store an n-bit word, we have to use an n-bit register containing n number of flip flops.

The register is used to perform different types of operations. For performing the operations, the CPU use these registers. The faded inputs to the system will store into the registers. The result returned by the system will store in the registers. There are the following operations which are performed by the registers:

Fetch:

It is used

To take the instructions given by the users.

To fetch the instruction stored into the main memory.

Decode:

The decode operation is used to interpret the instructions. In decode, the operation performed on the instructions is identified by the CPU. In simple words, the decode operation is used to decode the instructions.

Execute:

The execution operation is used to store the result produced by the CPU into the memory.

After storing this result, it is displayed on the user screen.

- **Types of Registers**

There are various types of registers which are as follows:

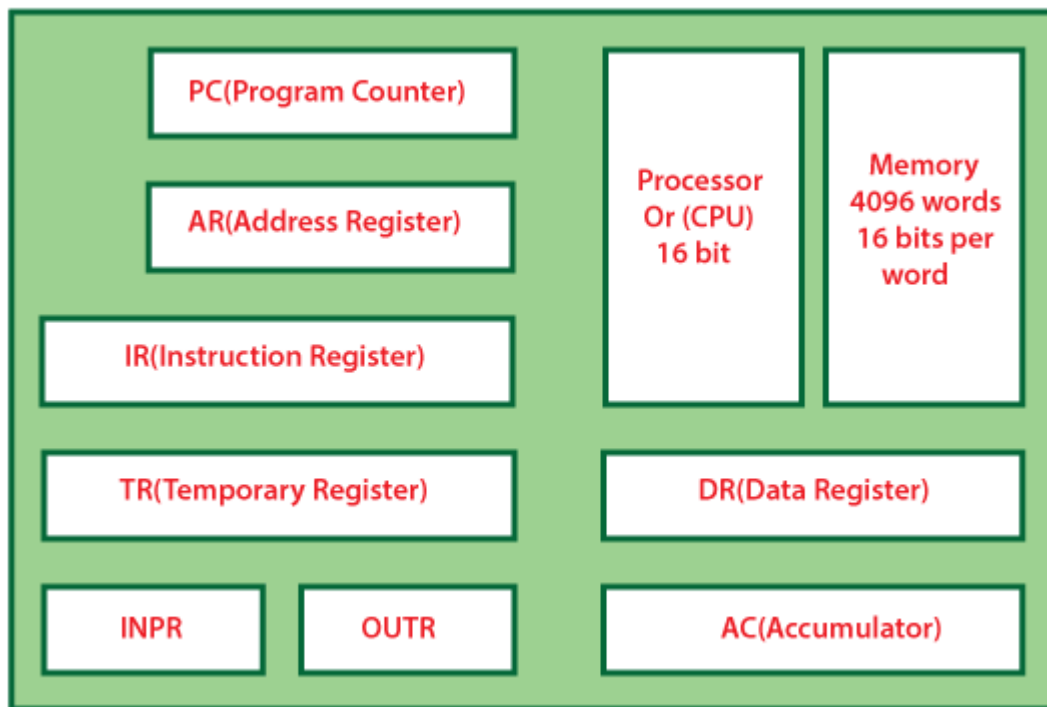


Figure 3.19 Types register

✓ **MAR or Memory Address Register**

The MAR is a special type of register that contains the memory address of the data and instruction. The main task of the MAR is to access instruction and data from memory in the execution phase. The MAR stores the address of the memory location where the data is to be read or to be stored by the CPU.

✓ **Program Counter**

The program counter is also called an instruction address register or instruction pointer. The next memory address of the instruction, which is going to be executed after completing the execution of current instruction is contained in the program counter. In simple words, the program counter contains the memory address of the location of the next instruction.

✓ **Accumulator Register**

The CPU mostly uses an accumulator register. The accumulator register is used to store the system result. All the results will be stored in the accumulator register when the CPU produces some results after processing.

✓ **MDR or Memory Data Register**

Memory Data Register is a part of the computer's control unit. It contains the data that we want to store in the computer storage or the data fetched from the computer storage. The MDR works as a buffer that contains anything for which the processor is ready to use it. The

MDR contains the copied data of the memory for the processor. Firstly the MDR holds the information, and then it goes to the decoder.

The data which is to be read out or written into the address location is contained in the

✓ **Memory Data Register.**

The data is written in one direction when it is fetched from memory and placed into the MDR.

In write instruction, the data place into the MDR from another CPU register. This CPU register writes the data into the memory. Half of the minimal interface between the computer storage and the microprogram is the memory data address register, and the other half is the memory data register.

✓ **Index Register**

The **Index Register** is the hardware element that holds the number. The number adds to the computer instruction's address to create an effective address. In CPU, the index register is a processor register used to modify the operand address during the running program.

✓ **Memory Buffer Register**

Memory Buffer Register is mostly called MBR. The MBR contains the Metadata of the data and instruction written in or read from memory. In simple words, it adds is used to store the upcoming data/instruction from the memory and going to memory.

✓ **Data Register**

The data register is used to temporarily store the data. This data transmits to or from a peripheral device.

3.2.4 Transistor technology

Transistors are semiconductor devices with three (and sometimes more) terminals. The third terminal enables output current to be controlled by a relatively small and low-power input signal. In amplifiers, transistors are used to achieve current gain, voltage gain, or power gain. Most often power gain is the objective in RF and microwave design. Most transistors are fabricated using silicon (Si) or compound semiconductors such as galliumarsenide (GaAs), indium phosphide (InP), or gallium-nitride (GaN). The overwhelming trend is to use silicon technology because of the much higher integration density that is possible, with compound semiconductor technology used only when it provides a unique advantage such as high power, superior noise performance, or high efficiency. Germanium is used as a dopant in silicon and then silicon is referred to as silicon germanium but usually germanium is in a very small proportion to silicon so SiGe as described here is silicon with a dopant. With comparable

concentrations of silicon and germanium SiGe is a compound semiconductor and this is used as a compound semiconductor at times.

There are three fundamental types of microwave transistors

- ✓ Bipolar junction transistors, (BJTs);
- ✓ Junction field effect transistors, (JFETs);
- ✓ Insulated gate FETs, (IGFETs), with the metal-oxide-semiconductor FETs, (MOSFETs), being the most common type of IGFET.

3.3 Applications of digital electronic circuits.

Digital circuits are used for many reasons. For example, digital circuits convert ac sine waves to digital pulses, various points are given digital values. These are later converted to a binary value.

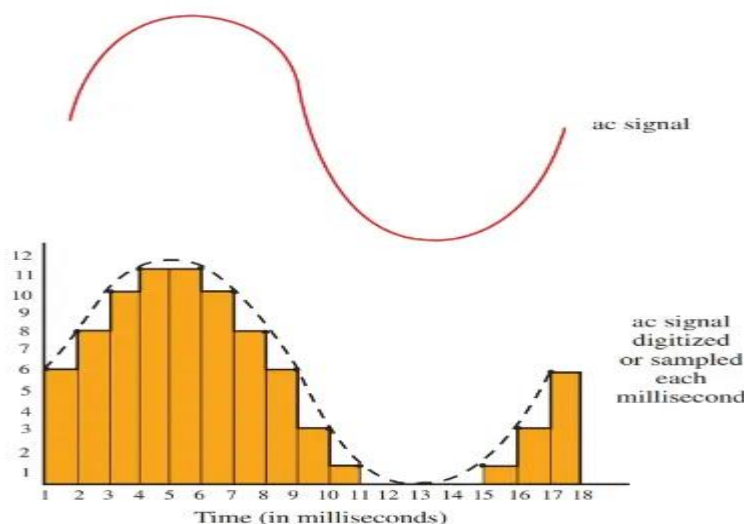


Figure 3.22. An ac signal being converted to digital pulses.

Figure 3.22 shows digital discs that have binary information recorded on their surface. A laser “reads” the information on the recording.

Another example of digital electronics use is the Universal Product Code (UPC). It is found on most items purchased today, **Figure 3**. The UPC is read by a laser (light). The laser improves inventory (item movement) control.

Some laser scanning systems contain a computer-generated voice that tells the operator and the customer what has been purchased and the price of the item.



Figure 3.23. Digital discs

The basic piece of test equipment used for digital circuits is the logic probe. The logic probe indicates either a high or low signal using LEDs. Look at **Figure 4**.

Digital Encoders and Decoders

Most people have only learned the decimal number system. The digital system uses the binary numbering system, which is very different from the decimal system.

To make the binary numbering easily understandable to us, we use an electronic system to translate, or decode, the binary system into the decimal system. A device that makes this translation is called a decoder.

An example of a decoder is the 7448 chip. This chip converts a binary code to its decimal number equivalent on an LED seven-segment display. This instrument allows us to easily read the binary number as a decimal number.

The conversion process also works in reverse. A decimal number can be converted to a binary number through an **encoder**. Once the decimal number has been encoded to a binary number, it can be utilized in a digital system such as a computer.

There are also analog-to-digital and digital-to-analog converters. These devices can change analog information such as temperature to a digital number equivalent.

A digital multimeter is a good example of how analog information can be changed to digital information. When a voltage reading is taken from a circuit using a DMM, the input to the meter is an analog signal.

Inside the meter, the encoder changes the analog signal into a binary code. This code is then turned into the decimal number value that appears on the meter's display.

Some circuits, such as the 7490 counter, are designed to convert electrical pulses into binary numbers. The counter advances by one each time the input receives an electrical pulse.

The counter can be connected to a decoder and an LED driver such as the 7448 to display the pulse count as a decimal number.

Digitized Analog Signals

A sound wave or linear voltage can be represented by a pattern of digital signals. There are many advantages to digitizing a sound wave pattern. A sound wave can be converted into a digital pattern using an analog-to-digital encoder. Each voltage level is represented by a binary code.

3.4 Testing digital electronic circuit

Basic Digital Troubleshooting

- The first step in learning digital troubleshooting is to learn the components that are available, and what is the overall functionality of the component in the circuit.
- The second step is to study basic circuits that are made from the components. The basic logic components in digital circuits are Logic gates (AND, OR, and NOT gates) and basic logic circuits are Flip-flops and so on.
- The third step in learning digital troubleshooting is to study how the basic circuits are combined into a complete system. For example, Counter, can be a digital system.
- Apart from this, one must also know how to conduct tests and measurements, and recognize symptoms.
- A logic circuit can have three basic signals:
 - ✓ Logic 1; it is just a way to refer the higher voltage level
 - ✓ Logic 0; it is the lower voltage level
 - ✓ A pulsating voltage signal that alternates between 0 and 1
- The quickest and most convenient way to measure these signals is to use a logic probe, not a voltmeter. An LED is used as an indicator and a series resistor to limit the current flowing through LED. The following figure shows a drawing of a logic probe:

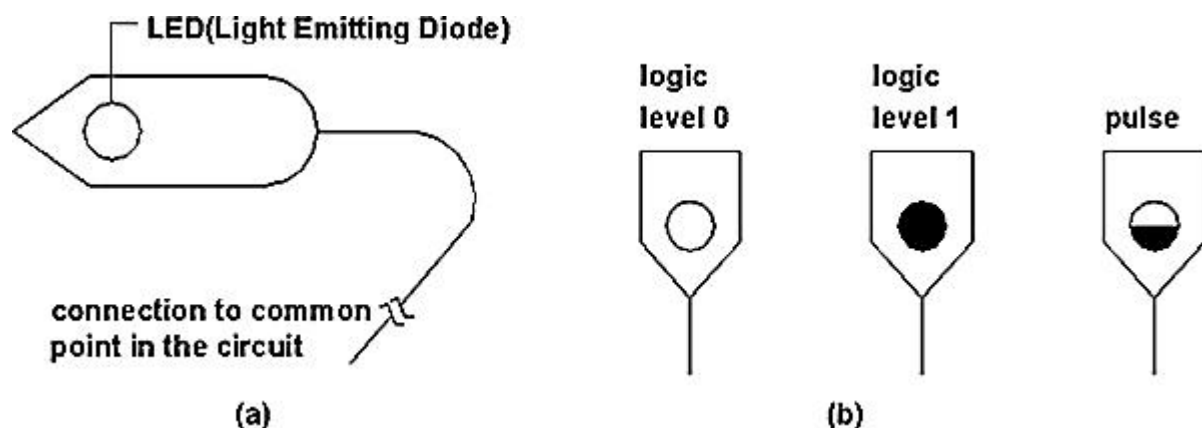


Figure 3.24 basic logic probe

When the probe is touching a 1 signal, the LED is ON, and when it touches a 0 signal, the LED is OFF. When it touches a pulsating signal, the LED is ON at half brightness or it is flashing ON and OFF. The following figure shows symbols for different logic probe indications.

- A logic probe is better than a voltmeter in digital troubleshooting. It is not required to take a reading each time when the probe is set. For example, suppose the voltmeter is in a circuit where 4.5 V is a logic 1 and 1 V is a logic 0 signal. If a voltmeter indicates 2.5 V, it does not mean that you are looking at low 1 signal or high 0 signal or a pulse. But with a good logic probe this guesswork is eliminated.

Typical Faults in Digital Systems

These can be some of the reasons for the faults to get induced in any digital circuit:

- ✓ Breaks in PCB tracks
- ✓ Shorts between tracks on the PCB
- ✓ Failure of discrete components external to the IC
- ✓ Dry joints

The typical failures include inputs or outputs shorted to ground, pins shorted to Vcc supply, pins shorted together, open pins and connections with intermittent defects.

There are hundreds and thousands of semiconductor devices assembled on one small chip. The critical problem for the chip manufacturers can be to get the voltages and signals in and out of such a tiny chip.

The wires which are used as inputs and outputs to the chips are very thin. Any type of thermal stress can affect these tiny wires and the bond may break away from the pad to the chip, causing an open connection.

The faults in a gate output manifest themselves as follows:

- **Stuck at '1'**, i.e. always above 2 V irrespective of input state (for TTL):

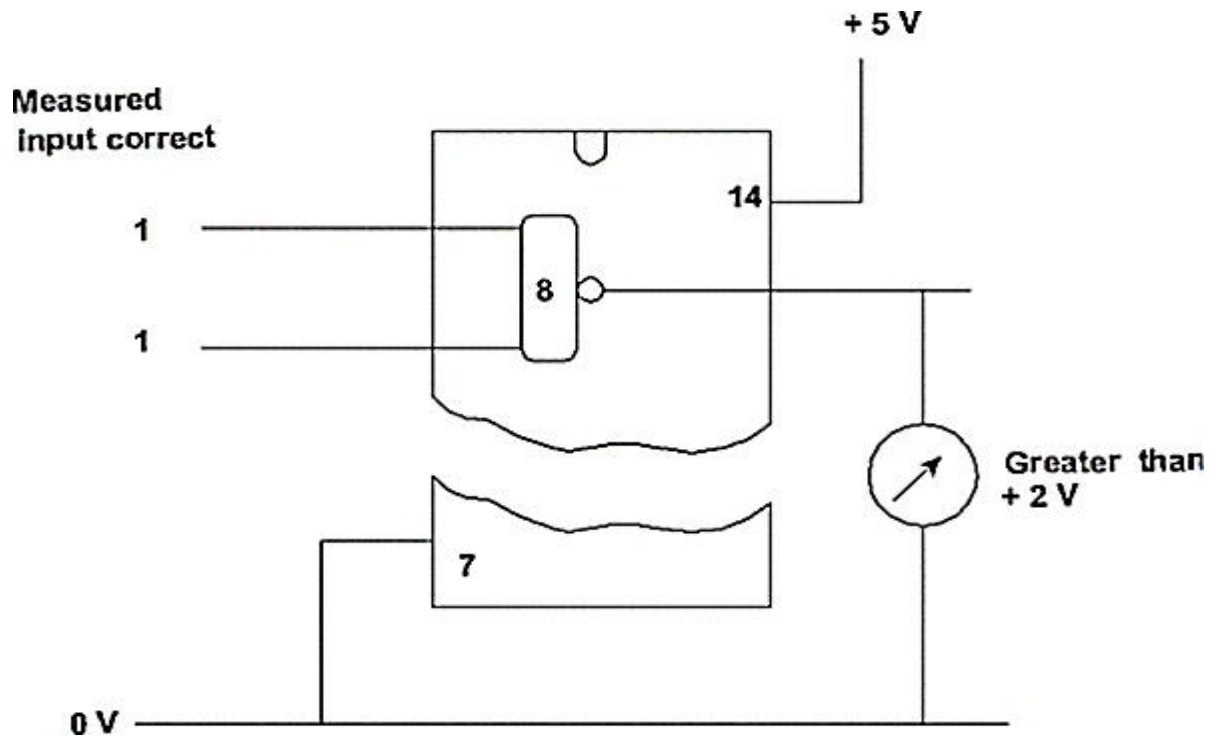


Figure 2.24 Output stuck at '1' (for a single gate)

In the above figure the output is stuck at '1' whereas with logic '1' on inputs the output should be less than 0.8 V. The possible faults in this circuit could be internal transistor open circuit or 0 V line open circuits internally.

- **Stuck at '0'**, i.e. always less than 0.8 V irrespective of input state (for TTL):

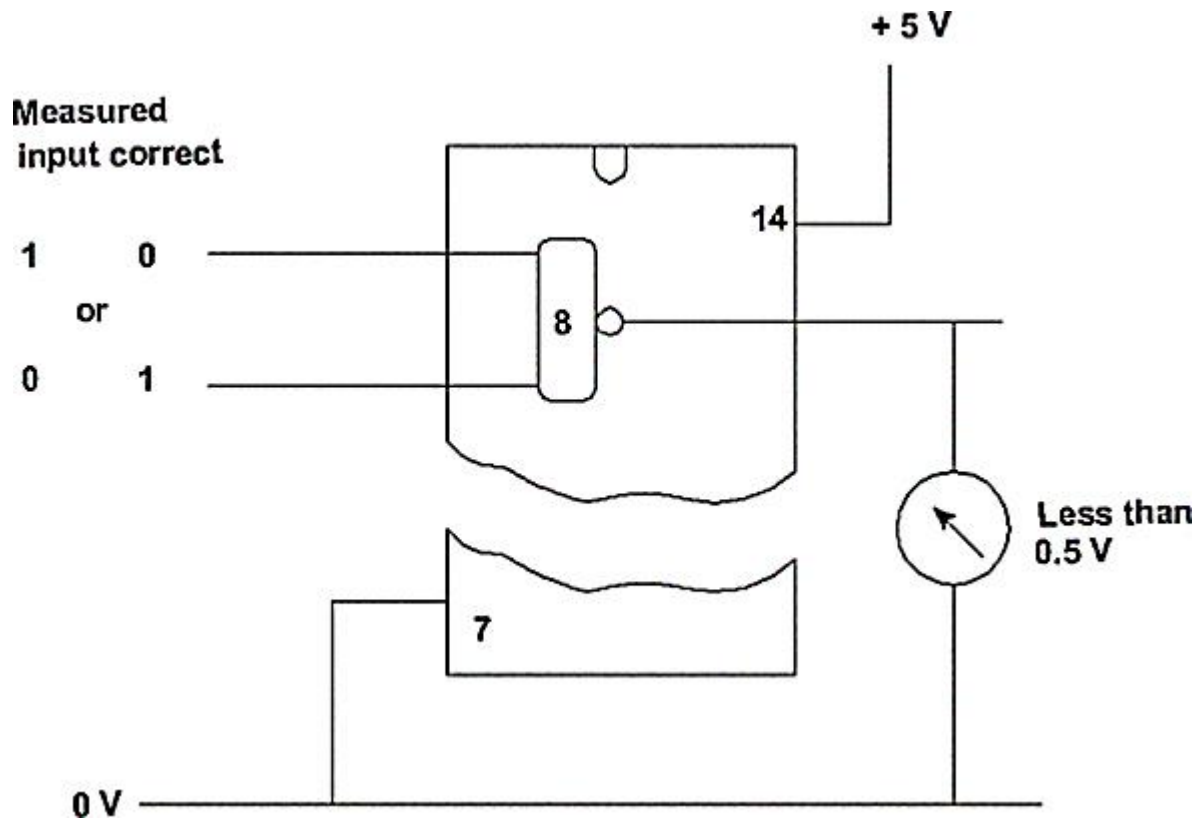


Figure 3.25 Output stuck at '0' (for a single gate)

Here the output is stuck at '0' whereas it should be logic '1'. The possible faults in this gate could be a short-circuited internal transistor, or V_{cc} line internally open.

- **High impedance** state output, i.e. output not 0 or 1.

The causes of the above faults are possibly due to internal IC failure. The internal failures can be of the following type:

- There can be a short circuit between an input terminal or an output terminal to V_{cc} or to the ground.
- There can be a short circuit between two pins, neither of which is at V_{cc} or ground potential.
- There can be an open lead which could be either an input or an output pin.

Common Faults in IC

The following types of common faults are observed in an IC:

- **Open Input/output Bond:**

If the output of an IC is open, it is floating. Let us consider that this output is supposed to be given as an input to another IC. In TTL circuits a floating input rises to approximately 1.5V and usually has the same effect on the circuit operation as a high logic level. This means that an open output bond in an IC will cause all inputs driven by that output to an incorrect level which are usually treated as logic high level by the inputs.

If there is an open input bond inside an IC, the digital signal that drives the signal will be unaffected and will be detectable at input pin. It will be as though the input were at the static high level.

- **Short Circuit between input/output and VCC or ground:**

When there is a short circuit between input/output and VCC or ground, all signal lines connected to that input/output are held either high (in case of short to VCC) or low (in case of short to ground). Thus, a fault usually causes normal signal activity at points beyond the short circuit to disappear and can be detected easily.

The following figure shows some common faults in a digital IC:

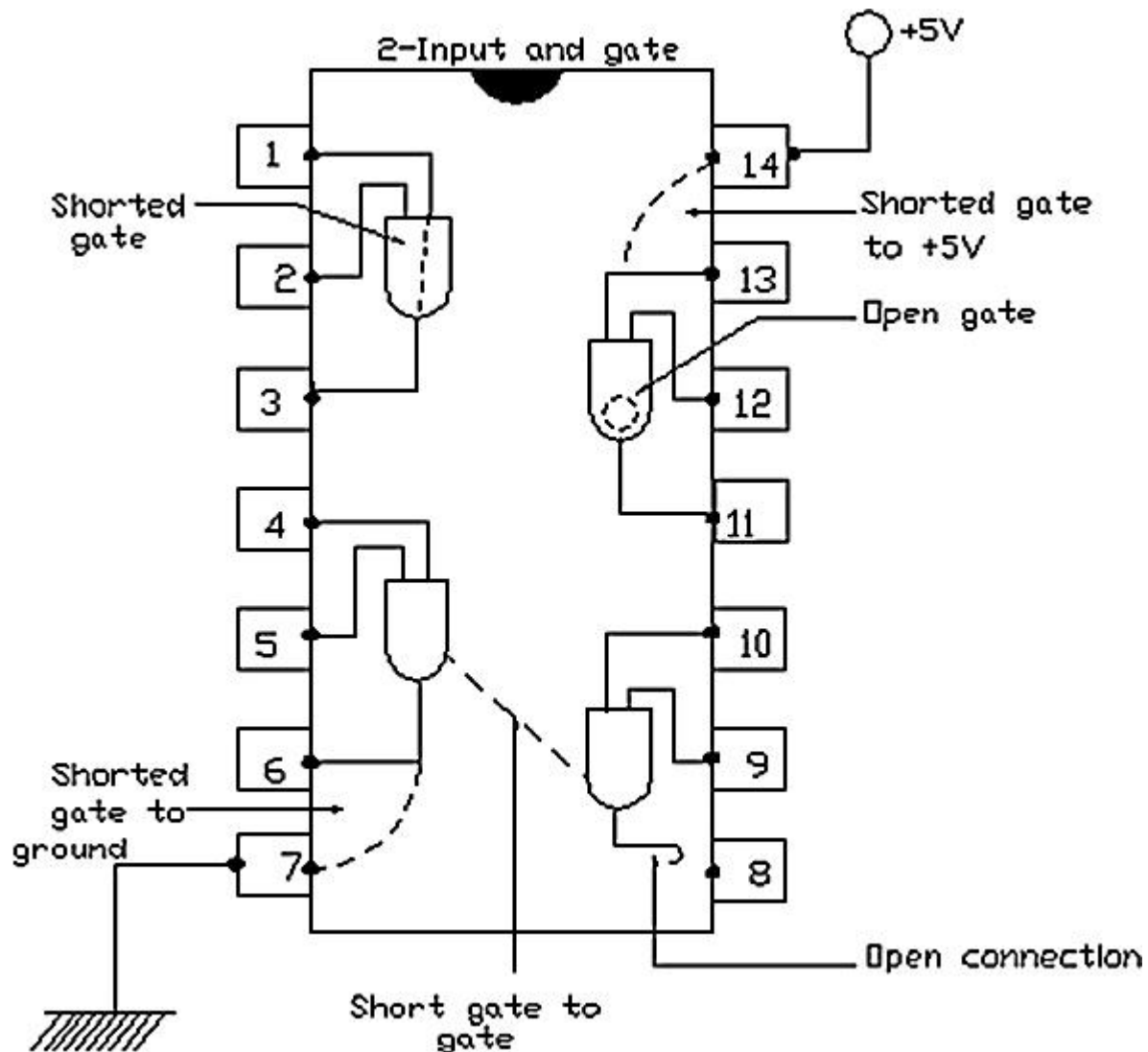


Figure 3.26 Common in a digital IC

The faults are listed as follows:

Shorted gate to + VCC

- Open gate inside
- Shorted gate inside
- Open connection to the pin
- Short between gate to gate and
- Short between gate and ground

Troubleshooting of Integrated Circuits

Troubleshooting of the logic ICs can be performed in three different types of tests. Functional test, AC test and DC test. All these tests can be made using a pulse generator with DC offset facility.

Functional Test

- It is the most common test carried out in servicing. This test simply determines if the IC performs to its truth table definitions. It is adequate for most combinational logic ICs, i.e. gates, inverters, multiplexers, decoders, encoders, etc.
- It is carried out by employing a series of pulses from pulse generators to step an IC through its various states. The states are monitored with the help of an oscilloscope.
- The output of the oscilloscope is so set that the base line of the generator should be at zero volts and the pulse amplitude should be set well above the minimum for logic 1.

DC Test

- If the functional test gives positive results, i.e. if the IC under test is following the truth table, but still the problem is undetectable, the DC test is carried out. In this test the DC parameters of the IC are tested. Information on DC parameter measurement can usually be found in the manufacturer's IC data book. The commonly specified DC voltage parameters of a logic IC are as follows:

- ✓ The maximum DC voltage accepted by an input as a logic 0
- ✓ The maximum DC voltage accepted by an input as a logic 1
- ✓ The maximum output at logic 0 into full load, and the minimum logic 1 output voltage level under full load
- ✓ Maximum power supply current
- ✓ Maximum output current supplied by the IC into a short at logic 1

AC Test

- When an IC fails to operate under high frequency conditions, at or near its maximum rated frequency, it is required to check the AC parameters of the IC. The method and the specifications of the AC parameter measurement are usually provided in the manufacturer's data book. Some of the important AC parameters are:

- ✓ Minimum and maximum input rise times
 - ✓ Minimum and maximum output rise times and fall time
 - ✓ Propagation delay through the IC
 - ✓ Maximum repetition rates
- With the pulse generator, a dual-trace oscilloscope is used.

Precautions

- Handle IC packages without touching the pins.
- Do not use tools which generate a static charge.
- Avoid placing tools or electrical parts on insulators, such as books, paper, rubber pads, plastic bags, etc.
- The grounded equipment should have rubber feet or other means of insulation.
- The work surface (bench top) must be conductive and reliably connected to earth ground through a safety resistance of 250 K Ω and 500 M Ω .
- When testing static charge-sensitive devices, DC power should be on before, during and after application of test signals.
- When boards or components (whether hard-wired or plug-in) are removed or inserted, ensure that all voltages are switched off.
- Enclose IC packages or circuit boards in conductive envelopes during transportation.

Self-check 3

Direction I multiple choice question

1. The gates required to build a half adder are _____
 - a. EX-OR gate and NOR gate
 - b. EX-OR gate and OR gate
 - c. EX-OR gate and AND gate
 - d. EX-NOR gate and AND gate
2. The gates required to build a half adder are _____

- A) EX-OR gate and NOR gate
 - B) EX-OR gate and OR gate
 - C) EX-OR gate and AND gate
 - D) EX-NOR gate and AND gate
3. The gates required to build a half adder are _____
- A) A). EX-OR gate and NOR gate
 - B) EX-OR gate and OR gate
 - C) EX-OR gate and AND gate
 - D) EX-NOR gate and AND gate
4. The gates required to build a half adder are _____
- A) EX-OR gate and NOR gate
 - B) EX-OR gate and OR gate
 - C) EX-OR gate and AND gate
 - D) EX-NOR gate and AND gate
5. The CPU mostly uses-----
- A) Program Counter
 - B) Accumulator Register
 - C) Memory Data Register.
 - D) all
6. Which of the following is faults fault of digital IC
- A) Shorted gate to + VCC
 - B) Open gate inside
 - C) Shorted gate inside
 - D) All

Direction III short answer: - Explain Applications of digital electronic circuits.

Direction II True /False

1. The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
2. The combinational circuit use memory for data backup
3. Half adder is a combinational logic circuit with two inputs and two outputs

4. Half subtractor is a combination circuit with two inputs and two outputs (difference and borrow).
5. The synchronous counter is also known as the ripple counter

Operation sheet-2

Operation Title: digital electronic circuit functional test

Instruction: In workshop, test the given digital electronic circuit.

Purpose: To test functionality of digital electronic circuit

Required tools and equipment: digital multimeter, oscilloscope, DC and AC power supply, wire, IC

Precautions:

- Handle IC packages without touching the pins.
- Do not use tools which generate a static charge.
- Avoid placing tools or electrical parts on insulators, such as books, paper, rubber pads, plastic bags, etc.
- The grounded equipment should have rubber feet or other means of insulation.
- The work surface (bench top) must be conductive and reliably connected to earth ground through a safety resistance of 250 K Ω and 500 M Ω .
- When testing static charge-sensitive devices, DC power should be on before, during and after application of test signals.
- When boards or components (whether hard-wired or plug-in) are removed or inserted, ensure that all voltages are switched off.
- Enclose IC packages or circuit boards in conductive envelopes during transportation

Procedures:

Step 1 let your multimeter to the continuity mode.

Step 2: Connect all of the pins altogether from one of the side in IC to the multimeter cable.

Step 3: Take the terminal-cable from multimeter and connect it one by one to each of the pins of another side separately.

Step 4: If the beep sound occurs of there is a continuity in more than 50% of the combinations then there is a good chance of the IC to be shorted from inside.

Step 5 clean working area and return back the tools to their place

Quality criteria:

- Proper component selection
- correct sequence of the operation

LAP Test	Practical Demonstration
----------	-------------------------

- Proper Handling of IC

Name: _____

Date: _____

Time started: _____

Time finished: _____

Instruction I: Given necessary templates, tools and materials you are required to perform the following tasks within 3 hours.

Task 1: test functionality of digital electronic circuit

Reference

1. Anil K. , M. (2007). *Digital Electronics*.
2. Dr. Debashis, C. (2017). *ELECTRICAL ENGINEERING*.
3. Michael L. , B. (1995). *ESSENTIALS OF ELECTRONIC TESTING*.
4. <https://www.electronics-notes.com/articles/test-methods/meters/transistor-circuit-fault-finding.php>
5. <https://www.techtarget.com/whatis/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR>
6. https://www.electronics-tutorials.ws/combinational/comb_1.html#:~:text=Combinational%20Logic%20Circuits%20are%20made,blocks%20of%20combinational%20logic%20circuits.

Developers Profile

No	Name	Qualification (Level)	Field of Study	Organization/ Institution	Mobile number	E-mail
1	Natnael Seyoum	B	Biomedical Engineer	Kombolcha PTC	0910019968	natybme93@gmail.com
2	Wondimu Zeyede	B	Biomedical Engineer	AATPTC	0961250550	Wondimzeyu336@gmail.com
3	Obsera Bulto	B	Biomedical Engineer	AATPTC	0906610215	obseee57@gmail.com
4	Leta Mengistu	C	Biomedical Technician	JUMC for biomedical engineer	0917926218	boriifanleta@gmail.com
5	Tariku Bedasa	B	Biomedical Engineer	AATPTC	09177708660	Tarikubedasa41@gmail.com