Logic Design

1- Introduction

It is nearly impossible to find a part of our society that has not been touched by digital electronics such as computers, televisions, digital video recorders and countless other consumer electronics would not be possible without them. The Internet is run on a system of computers and routing equipment built with digital electronics. Yet even outside of some of these obvious applications we find that our cars and utilitarian home appliances such as microwaves, washers, dryers, coffee makers and even refrigerators are all increasingly being designed with digital electronic controls. You likely carry some sort of device designed with them with you nearly all your waking hours whether it is a watch, cell phone, MP3 player or PDA. Indeed, digital electronics provide the foundation upon which we build the infrastructure of modern society.

You no doubt have heard stories about some of the first computers. Machines built with mechanical relays and vacuum tubes that filled entire rooms. In the 1940s John Bardeen, Walter Brattain and William Shockley developed the first transistor; it allowed computers to be built cheaper, smaller and more reliable than ever before. The integrated circuit, a single package with several transistors along with other circuit components, was developed in the late 1950s by Jack Kilby at Texas Instruments. This helped to further advance the digital revolution. Advances then became so common that in the 1960s Gordon Moore, a founder of Intel, proposed his famous law stating that the capacity of computers we use would double every two years. This observation has held up since then, even being amended to a doubling every eighteen months.

The quad core microprocessors of today contain millions of components, but the basic building blocks are digital logic functions combined with memory. Despite the fact that many of these devices are tremendously complex and require vast amounts of engineering in their design, they all share the ubiquitous bit as their fundamental unit of data. In essence it all starts with TRUE and FALSE or 0 and 1. And so the next chapter starts with the simplest of logic devices, the inverter, built with a single transistor. You then continue your journey into the world of digital electronics by examining the NAND and NOR gates. Remember, the digital revolution would not be possible without these simple devices.

2- Digital Systems

Digital means electronic technology that generates, stores, and processes data in terms of two states: positive and non-positive. Positive is expressed or represented by the number 1 and non-positive by the number 0.

A „**digital system**" is a data technology that uses discrete (discontinuous) values represented by high and low states known as bits. By contrast, non-digital (or analog) systems use a continuous range of values to represent information. Although digital representations are discrete, the information represented can be either discrete, such as numbers, letters or icons, or continuous, such as sounds, images, and other measurements of continuous systems.

Digital computer is a part of digital system, it based on binary system. A block diagram of digital computer is shown in figure (1):

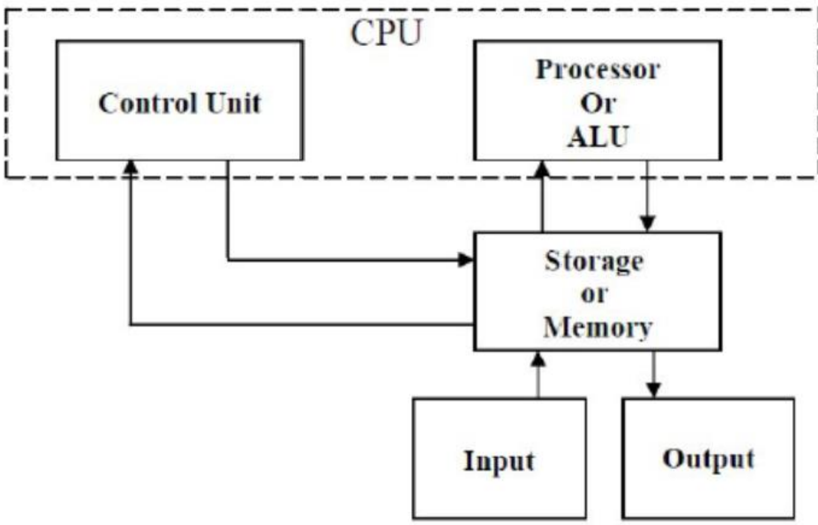


Figure 1- Digital Computer

Where

- CPU is the Central Processing Unit. CU is the Control Unit.
- ALU is the Arithmetic Logic Unit.
- The processor when combined with the control unit form a component referred to as CPU.
- Storage unit stores programs as well as input, output and intermediate data.
- The processor unit performs arithmetic and other data processing tasks as specified by the program.
- The control unit supervised the flow of information between various units.
- The program and data prepared by the user are transformed into the memory unit by means of input devices such as: punch-card reader, keyboard and scanner …etc.
- The output unit presents the results of the computation to the user in a form that the user understands it (compatible with the user) such as: card punching, printer and magnetic tape …etc.

Memory Capacity Measurement

The value of the computer is not measured by its large size or its small size, but its value is measured by its memory capacity gradually. It starts with the following:

- Bit = 0 or 1.
- Nibble = 4 bits.
- Byte = 8 bits.
- Word = 2 bytes = 16 bits.
- 1 K (Kilo) = 2^{10} = 1024.
- 1 M (Mega) = 2^{20} = 1024 * 1024.
- 1 G (Giga) = 2^{30} = 1024 * 1024 * 1024.
- 1 T (Tera) = 2^{40} = 1024 * 1024 * 1024 * 1024.

Ex:- Find the exact number of bits of 64 Mega Bytes?

$= 64 * 2^{20} * 8$

$= 64 * 1024 * 1024 * 8$

$= 536870912$

3- Data Representation and Number system

Numeric Systems (Decimal, Binary, Octal, Hexadecimal)

A number system is nothing more than a code representing quantity. We have learned and use the decimal numbering system simply because humans are born with ten fingers! The decimal system has served us well. But with digital systems, we need a 2-value system (binary). We could attribute this to the fact that computers only have open or closed switches (or one finger, if you prefer). This means, we have to learn the binary system in addition to the decimal system. We also will discuss the octal and hexadecimal systems because conversion to/from binary is easy and numbers in these systems are easier to read than binary numbers for humans.

System Basic	**Base (Radix) Number**	**Digits**
Binary	2	0,1
Octal	8	0, 1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

- ❖ **Decimal Number (base or radix 10)**
  - Humans use the decimal numbering system as a default, so when you see a number 56 your assumption is that its base is 10 or $(56)_{10}$ which is "56 base 10".
  - Each digit is weighted based on its position in the sequence (power of 10) from the Least Significant Digit (LSD, power of 0) to the Most Significant Digit (MSD, highest power).
  - Each digit must be less than 10 (0 to 9).
  - The weight structure of the decimal number is:

$$10^{n-1} \ldots.. \; 10^4 \;\; 10^3 \;\; 10^2 \;\; 10^1 \;\; 10^0 \;\; . \;\; 10^{-1} \;\; 10^{-2} \;\; 10^{-3} \;\; 10^{-4} \;\; \ldots\ldots.. \; 10^{-n}$$

For example $(2375.46)_{10}$ is evaluated as:

$$(2375.46)_{10} = (2 \text{ x } 10^3) + (3 \text{ x } 10^2) + (7 \text{ x } 10^1) + (5 \text{ x } 10^0) + (4 \text{ x } 10^{-1}) + (6 \text{ x } 10^{-2})$$
$$= 2000 + 300 + 70 + 5 + 0.4 + 0.06$$
$$= 2375.46$$

- ❖ **Binary Number (base or radix 2)**
  - Digital and computer technology is based on the binary number system, since the foundation is based on a transistor, which only has two states: **on** or **off**.
  - Each digit of the number is called a **bit** or which is a short for **binary digits.**

- Each bit is weighted based on its position in the sequence (powers of 2) from the Least Significant Bit (LSB) to the Most Significant Bit (MSB).
- Each bit must be less than 2 which mean it has to be either 0 or 1.
- The weight structure of the decimal number is:

$$2^{n-1} \ldots\ldots 2^4\ 2^3\ 2^2\ 2^1\ 2^0\ .\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ \ldots\ldots\ 2^{-n}$$

For example $(1010.11)_2$ is evaluated as:

$$\begin{aligned}(1010.11)_2 &= (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) + (1 \times 2^{-1}) + (1 \times 2^{-2}) \\ &= 8 + 0 + 2 + 0 + 0.5 + 0.25 \\ &= (10.75)_{10}\end{aligned}$$

More examples:-

	32	**16**	**8**	**4**	**2**	**1**
	2^5	**2^4**	**2^3**	**2^2**	**2^1**	**2^0**
4	0	0	0	1	0	0
12	0	0	1	1	0	0
22	0	1	0	1	1	0

❖ **Octal (base 8) and Hexadecimal (base 16)**

These number systems are used by humans as a representation of long strings of bits since they are:

- Easier to read and write, for example $(347)_8$ is easier to read and write than $(011100111)_2$.
- Easy to convert (Groups of 3 or 4)
- Today, the most common way is to use Hex to write the binary equivalent; two hexadecimal digits make a Byte (groups of 8-bit), which are basic blocks of data in Computers.
- The hexadecimal system is base 16, so the digits range in value from 0 to 15. How do you represent Hexadecimal digits above 9?
  Use A for 10, B for 11, C for 12, D for 13, E for 14 and F for 15. So (CAB)16 or (CAB)HEX is a valid hexadecimal number.

Octal examples:-

	64	**8**	**1**
	8^2	**8^1**	**8^0**
4	0	0	4
12	0	1	4
22	0	2	6

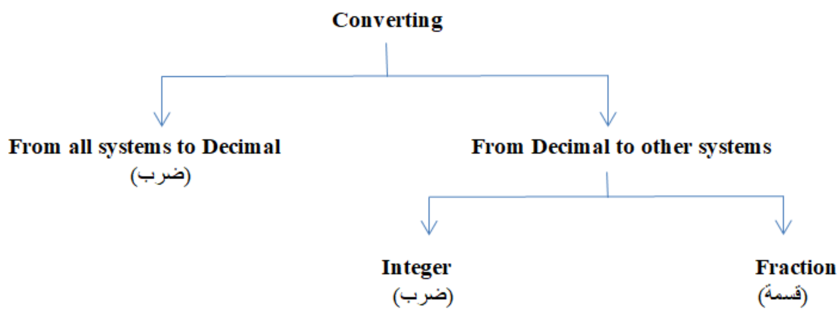
Hexadecimal examples:-

4096	**256**	**16**	**1**
16^3	**16^2**	**16^1**	**16^0**

4	0	0	0	4
25	0	0	1	9
45	0	0	2	D
284	0	1	1	C

4- The Conversion between Numbering Systems

1- Converting from/to Decimal Number:



- From Binary to Decimal Conversion:

To convert any binary number to its decimal equivalent, we must use the positional number law of numbers. This law applies when the binary number is integer or a fraction, bearing in mind that the basis of the counting system here is 2.

$$\mathbf{N = a_nR^n + a_{n-1}R^{n-1} + \ldots + a_0R^0 + a_{-1}R^{-1} + \ldots + a_{-m}R^{-m}}$$

In other words, the decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0.

Ex1:- Convert the binary number $(101101)_2$ to decimal.
$(101101)_2$ ……$(\ )_{10}$
Solution:

Weight: 2^5 2^4 2^3 2^2 2^1 2^0
Binary: 1 0 1 1 0 1

$(101101)_2 = (1 * 2^5) + (0 * 2^4) + (1*2^3) + (1*2^2) + (0*2^1) + (1*2^0)$
$= 2^5 + 2^3 + 2^2 + 2^0$
$= 32 + 8 + 4 + 1$
$= (45)_{10}$

Ex2:- Convert the binary number $(0.1011)_2$ to decimal.
$(0.1011)_2$ ……$(\ )_{10}$
Solution:

Weight: 2^{-1} 2^{-2} 2^{-3} 2^{-4}
Binary: 0 . 1 0 1 1

$(0.1011)_2 = (1 * 2^{-1}) + (0 * 2^{-2}) + (1*2^{-3}) + (1*2^{-4})$

$= 2^{-1} + 2^{-3} + 2^{-4}$
$= 0.5 + 0.125 + 0.0625$
$= (0.6875)_{10}$

- From Octal to Decimal Conversion:

To convert from octal to decimal, the positional number representation law is used, bearing in mind that the basis of the numeral system here is 8.

In other words, the evaluation of an octal number in terms of its decimal equivalent is accomplished by multiplying each digit by its weight and summing the products.

Ex1:- Convert the octal number $(2374)_8$ to decimal.
$(2374)_8$ ……$(\ )_{10}$
Solution:

Weight: 8^3 8^2 8^1 8^0
Octal: 2 3 7 4

$$(2374)_8 = (2 * 8^3) + (3 * 8^2) + (7 * 8^1) + (4 * 8^0)$$
$$= (2 * 512) + (3 * 64) + (7 * 8) + (4 * 1)$$
$$= 1024 + 192 + 56 + 4$$
$$= (1276)_{10}$$

Ex2:- Convert the octal number $(206.75)_8$ to decimal.
$(206.75)_8$ ……$(\ )_{10}$
Solution:

Weight: 8^2 8^1 8^0 8^{-1} 8^{-2}
Octal: 2 0 6 . 7 5

$$(206.75)_8 = (2 * 8^2) + (0 * 8^1) + (6 * 8^0) + (7 * 8^{-1}) + (5 * 8^{-2})$$
$$= (2 * 64) + (6 * 1) + (7 * 1/8) + (5 * 1/64)$$
$$= 128 + 6 + 7/8 + 5/64$$
$$= 134 + 0.875 + 0.078125$$
$$= (134.953125)_{10}$$

- From Hexadecimal to Decimal Conversion:

To convert from hexadecimal to decimal, the positional number representation law is used, bearing in mind that the basis of the numeral system here is 16.

Ex1:- Convert the octal number $(ABC)_{16}$ to decimal.
$(ABC)_{16}$ ……$(\ )_{10}$
Solution:

Weight: 16^2 16^1 16^0
Octal: A B C

$$(ABC)_{16} = (10 * 16^2) + (11 * 16^1) + (12 * 16^0)$$
$$= 2560 + 176 + 12$$
$$= (2748)_{10}$$