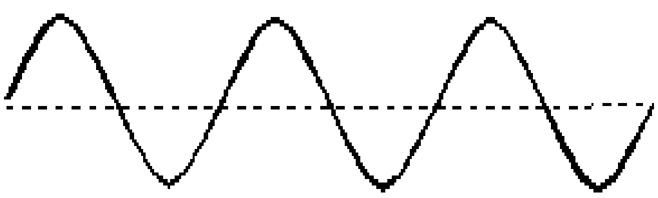
Applied Logic

Chapter One
Digital Computers & Digital Systems

1.1 **Logic Terminologies**

- Analog Signal : A waveform is a representation of how alternating current (AC) varies with time. The most familiar AC waveform is the *sine wave*, which derives its name from the fact that the current or voltage varies with the sine of the elapsed time. Broadcast and telephone transmission have conventionally used analog technology.

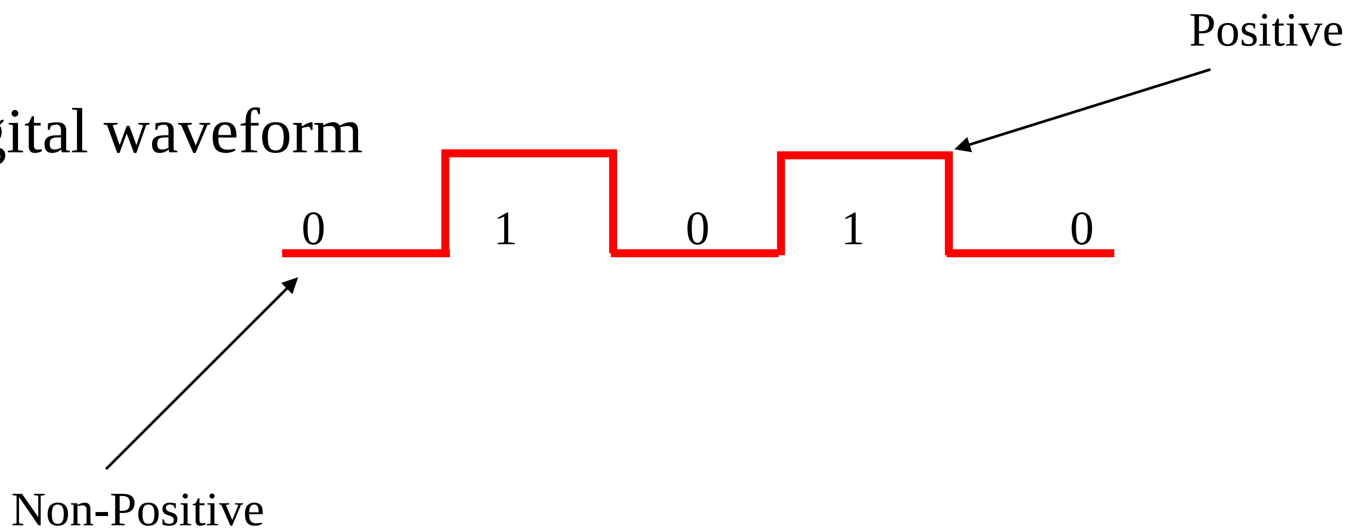
- Analog signal waveform



- Wireless signal has a sine waveform, with a frequency usually measured in megahertz (MHz) or gigahertz (GHz). Household utility current has a sine waveform with a frequency of 60 Hz in most countries including the United States, although in some countries it is 50 Hz. Such as $V(t)=220 \sin(1/50)$

- Digital describes 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. Thus, data transmitted or stored with digital technology is expressed as a string of 0's and 1's. Each of these state digits is referred to as a bit (and a string of bits that a computer can address individually as a group is a byte). 1byte=8bit

- A bit (short for *binary digit*) is the smallest unit of data in a computer. A bit has a single binary value, either 0 or 1.
- Digital waveform



1.2 **Introduction**

- Digital design is concerned with the design of digital electronic circuits. The subject is also known by other names such as logic design, digital logic, switching circuits, and digital systems. Digital circuits are employed in the design of systems such as digital computers, control systems, data communications, and many other applications that require electronic digital hardware.

- Computers are used in scientific calculations, commercial and business data processing, air traffic control, space guidance, the educational field, and many other areas. The most striking property of a digital computer is its generality.

- Computer can follow a sequence of instructions, called a *program*, that operates on given data.

- The general-purpose digital computer is the best-known example of a digital system.

- Characteristic of a digital system is its manipulation of discrete elements of information. Such discrete elements may be electric impulses, the decimal digits, the letters of an alphabet, arithmetic operations.

- In this case, the discrete elements used are the digits. From this application, the term *digital computer* has emerged. A more appropriate name for a digital computer would be a "*discrete information-processing system*."

- Discrete elements of information are represented in a digital system by physical quantities called *signals*. Electrical signals such as voltages and currents are the most common. The signals in all present-day electronic digital systems have only two discrete values and are said to be *binary (1,0)or (ON, Off).*

- The digital-system designer is restricted to the use of binary signals because of the lower reliability of many-valued electronic circuits.

- The term *analog signal is* sometimes substituted for *continuous signal* .

- To simulate a physical process in a digital computer, the quantities must be quantized. When the variables of the process are presented by real-time continuous signals, the latter are quantized by an analog-to-digital conversion device.

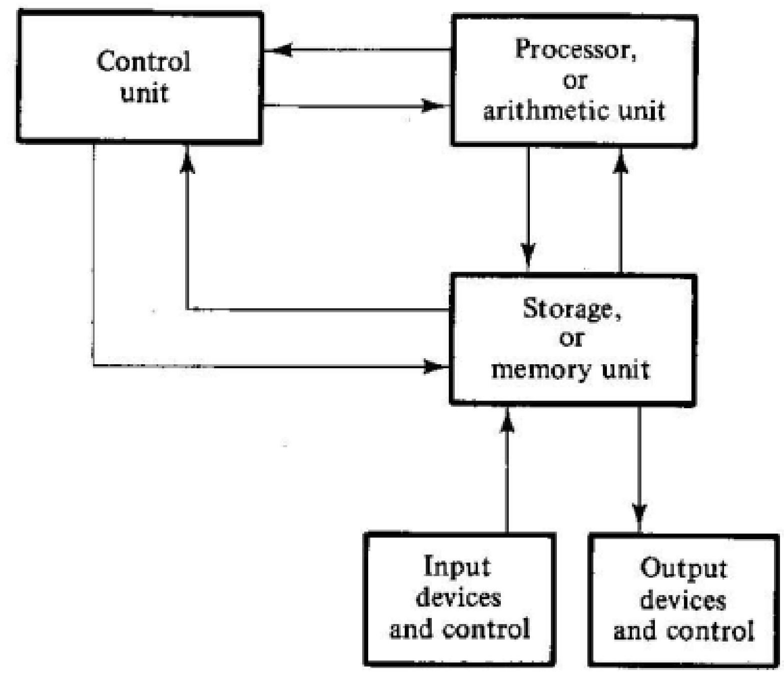


Figure 1-1 Block diagram of a digital computer

- Memory unit stores programs.

- The processor unit per-forms arithmetic and other data-processing tasks as specified by a program.

- The control unit supervises the flow of information between the various units. The control unit retrieves the instructions, one by one, from the program that is stored in memory. For each instruction, the control unit informs the processor to execute the operation specified by the instruction.

- The program and data prepared by the user are transferred into the memory unit by means of an input device such as a keyboard.

- An output device, such as a printer

Example on digital computer

- An electronic calculator is a digital system similar to a digital computer, with the input device being a keyboard and the output device a numerical display. Instructions are entered in the calculator by means of the function keys, such as plus and minus. Data are entered through the numeric keys. Results are displayed directly in numeric form.

Computer Number Systems

Computer number systems are consisting of:-

1. Decimal System

•Base 10
•Valid digits (0,1,2,3,4,5,6,7,8,9)
•Each coefficient is multiplied by 10
Humans are familiar with this system.

Example:
7392 should be written as: $7x10^3 + 3x10^2 + 9x10^1 + 2x10^0$

$(7392)_{10}$

2. Binary System

- Base 2
- Valid digits (0,1)
- Each coefficient is multiplied by 2
- 0 → Low 0Volt
- 1→High 5Volt
- MSD (Most Significant Digit)
- LSD (Least Significant Digit)
- Binary Fraction: $2^n.....2^3 2^2\ 2^1\ 2^0.2^{-1}\ 2^{-2}\ 2^{-3}.......2^{-n}$

Example: Conversion from binary to decimal 1101?

Solution: Decimal equivalent= $1x2^3+1x2^2+0\ X\ 2^1+1x\ 2^\circ$

$= 8+4+0+1=13$

Example: Convert $(11010.11)_2$ to decimal?

Solution: $1\ x2^4+1\ x2^3+0x2^2+1\ X\ 2^1+0x\ 2^\circ+1\ x\ 2^{-1}+1\ x2^{-2}$

$16+\ 8+2+½+¼=26.75$

Example: Convert $(110111)_2$ to decimal$(\ )_{10}$?

3. Octal System

- Base 8
- Valid digits (0,1,2,3,4,5,6,7,8)

- There are eight bits in a byte which is used very often in the computer field. Since $2^3=8$, each octal digit corresponds to three binary digits.

- The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right.

Example: Conversion from binary to octal $(010110001101011.111100000110)_2$?

Solution: By partitioning binary number into three digits.

$$(\underline{010}\ \ \underline{110}\ \ \underline{001}\ \ \underline{101}\ \ \underline{011}\ .\ \underline{111}\ \ \underline{100}\ \ \underline{000}\ \ \underline{110})_2 = (26153.7406)_8$$

$$2 \quad 6 \quad 1 \quad 5 \quad 3 \qquad 7 \quad 4 \quad 0 \quad 6$$

4. Hexadecimal System

- Base 16
- Valid digits (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
- Five new numbers had to be created. Those numbers are A, B, C, D, E, F. “A” has a value of 10, “B” is 11, and so on. Since $2^4=16$, each hexadecimal digit corresponds to four binary digits.

Example: Conversion from binary to hexadecimal $(0010110001101011.11110010)_2$?

Solution: By partitioning binary number into four digits.

$(\underline{0010}\ \ \underline{1100}\ \ \underline{0110}\ \ \underline{1011}.\underline{1111}\ \underline{0010})_2 = (2C6B.F2)_{16}$

2 C 6 B F 2

Number Base Conversions

- $(1010.011)_2 = 2^3 + 2^1 + 2^{-2} + 2^{-3} = (10.375)_{10}$
- $(630.4)_8 = 6 \times 8^2 + 3 \times 8 + 4 \times 8^{-1} = (408.5)_{10}$
- $(4021.2)_5 = 4x5^3 + 0\ x5^2 + 2x5^1 + 1x5^0 + 2x5^{-1} = (511.4)_{10}$
- $(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \text{ X } 16^1 + 15 \times 16^0 = (46687)10$

Example: Convert decimal 41 to binary?

Solution: First, 41 is divided by 2 to give an integer quotient of 20 and a remainder of 1/2. The quotient is again divided by 2 to give a new quotient and remainder. This process is continued until the integer quotient becomes 0. The *coefficients* of the desired binary number are obtained from the *remainders* as follows:

Integer quotient		Remainder	Coefficient
41/2=20	+	½	A0=1
20/2=10	+	0	A1=0
10/2=5	+	0	A2=0
5/2=2	+	½	A3=1
2/2=1	+	0	A4=0
1/2=0	+	½	A5=1

Answer: $(41)_{10} = (A5\,A4\,A3\,A2\,A1\,A0)_2 = (101001)_2$

Example: Convert decimal 153 to octal?

Solution: The required base r is 8. First, 153 is divided by 8 to give an integer quotient of 19 and a remainder of 1. Then 19 is divided by 8 to give an integer quotient of 2 and a remainder of 3. Finally, 2 is divided by 8 to give a quotient of 0 and a remainder of 2. This process can be conveniently manipulated as follows:

Integer	Remainder
153	
19	1
2	3
0	2

$= (231)_8$

Example: Convert $(0.6875)_{10}$ to binary?

Solution: First, 0.6875 is multiplied by 2 to give an integer and a fraction. The new fraction is multiplied by 2 to give a new integer and a new fraction. This process is continued until the fraction becomes 0 or until the numbers of digits have sufficient accuracy. The coefficients of the binary number are obtained from the integers as follows:

	Integer		Fraction	Coefficient	
0.6875 x 2 =	1	+	0.3750	a_{-1} =	1
0.3750 x2=	0	+	0.7500	a_{-2} =	0
0.7500 x2=	1	+	0.5000	a_{-3} =	1
0.5000 x2=	1	+	0.0000	a_{-4} =	1

Answer: $(0.6875)_{10} = (0.a_{-1}, a_{-2,} a_{-3}, a_{-4}, a_{-5})_2 = (0.1011)_2$

Example: Convert $(0.513)_{10}$ to octal?

Solution:

0.513 x 8 =	4.104	4
0.104 x 8 =	0.832	0
0.832 x 8 =	6.656	6
0.656 x 8 =	5.248	5
0.248 x 8 =	1.984	1
0.984 x 8 =	7.872	7

The answer, to seven significant figures, is obtained from the integer part of the products:

$$(0.513)_{10} = (0.406517\ \ldots)_8$$

Example: Convert $(673.124)_8$ and $(306.D)_{16}$ to binary?

Solution:

$(673.124)_8 = (110\ 111\ 011\ .\ 001\ 010\ 100)_2$

$(306.D)_{16} = (0011\ 0000\ 0110\ .\ 1101)_2$

Binary Arithmetic

1. Addition: The rules of addition are:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 10$$
$$1 + 1 + 1 = 11$$

Example: Perform the following additions?

1. $(101)_2 + (111)_2$
2. $(11.1)_2 + (10.01)_2$

Solution: 1.

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

2.

$$\begin{array}{r} 11.1 \\ 10.01 \\ \hline 101.11 \end{array}$$

2. Subtraction: The rules of subtractions are:

$$0 - 0 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1$$
$$1 - 1 = 0$$

Example: Subtract the following numbers?

1. $(101)_2 - (11)_2$
2. $(1001)_2 - (10)_2$

Solution:

1.
$$\begin{array}{r} 101 \\ 11 \\ \hline 10 \end{array}$$

2.
$$\begin{array}{r} 1001 \\ 10 \\ \hline 111 \end{array}$$

3. Multiplication: The rules of multiplication are: 0 x 0 = 0

0 x 1 = 0

1 x 0 = 0

1 x 1 = 1

Example: Perform the following operates?

1. $(11)_2$ x $(101)_2$
2. $(11.11)_2$ x $(10.1)_2$

Solution: 1.

```
   101
    11
  ----
   101
  101
  ----
  1111
```

2.

```
     11.11
      10.1
    ------
      1111
     0000
    1111
    ------
  1001.011
```

Example: addition, subtraction, and multiplication of two binary numbers are as follows:

Solution:

augend:	101101	minuend:	101101	multiplicand:	1011
addend:	+100111	subtrahend:	-100111	multiplier:	x 101
sum:	1010100	difference:	000110		1011
					0000
					1011
				product:	110111

H.W1:

Convert the following binary numbers to decimal?

$(1101)_2 = (\quad)_{10}$?

$(110111)_2 = (\quad)_{10}$?

$(0.101)_2 = (\quad)_{10}$?

$(11.1011)_2 = (\quad)_{10}$?

H.W2:

Convert the following numbers?

$(69)_{10} \rightarrow (\quad)_2$

$(37)_{10} \rightarrow (\quad)_2$

$(21)_{10} \rightarrow (\quad)_2$

$(110000)_2 \rightarrow (\quad)_{10}$

$(0.25)_{10} \rightarrow (\quad)_2$

H.W3 :

Convert the following numbers?

$(150.F)_{16} \rightarrow (\quad)_2$

$(227.025)_8 \rightarrow (\quad)_2$

Complements

Complements are used in digital computers for simplifying the subtraction operation and for logical manipulation.

Two types:

1- 1's Complement and 2's Complement for binary numbers.

2- 9's Complement and 10's Complement for decimal numbers.

1's and 9's called Diminished Radix Complement

2's and 10's called Radix Complement

1- 1's complement: It done by changing the (0) to (1) and vice versa.

Example:	<u>Binary</u>	<u>1's complement</u>
	0000	1111
	111	000
	010	101
	1011000	0100111
	0101101	1010010

2- 9's Complement:

Example: The 9's Complement for 546700 is

999999- 546700 =453299

999999- 012398 =987601

3- 2’s Complement : is equal 1’scomplement +1

Example: Take a 2’s Complement for 101100?

Solution: 1- Take a 1’scomplement 010011

2- Pulse 1

$$\begin{array}{r} 010011 \\ +\quad 1 \\ \hline 010100 \end{array}$$

Example: Take a 2’s Complement for 1101100?

Solution: 1- Take a 1’scomplement 0010011

2- Pulse 1

$$\begin{array}{r} +\quad 1 \\ \hline 0010100 \end{array}$$

4- 10’s Complement : is equal 9’s Complement +1

Example: Take a 10’s Complement for 2389?

Solution: 1- Take a 9’s Complement 9999-2389=7610

2- Pulse 1

$$\begin{array}{r} 7610 \\ +\quad 1 \\ \hline 7611 \end{array}$$

Example: Take a 10’s Complement for 012398?

Solution: 1- Take a 9’s Complement 999999-012398=987601

2- Pulse 1

$$\begin{array}{r} 987601 \\ +\quad 1 \\ \hline 987602 \end{array}$$

Subtraction With Complements

The subtraction of two n-digit unsigned numbers $M - N$ in base r can be done as follows:

1. Add the minuend M to the is complement of the subtrahend N. This performs $M + (r^n - N) = M - N + r^n$.
2. If $M \geq N$, the sum will produce an end carry, r^n, which is discarded; what is left is the result $M - N$.
3. If $M < N$, the sum does not produce an end carry and is equal to $r^n - (N - M)$, which is the is complement of $(N - M)$. To obtain the answer in a familiar form, take the is complement of the sum and place a negative sign in front.

Example: Using 10's complement, subtract 72532 – 3250?

Solution:

M =	72532
10's complement of N =	+ 96750
Sum =	169282
Discard end carry 10^5=	-100000
Answer =	69282

Note that M has 5 digits and N has only 4 digits. Both numbers must have the same number of digits; so we can write N as 03250. Taking the 10's complement of N produces a 9 in the most significant position. The occurrence of the end carry signifies that $M \geq N$ and the result is positive.

Example: Using 10's complement, subtract 3250 – 72532?

Solution:

M =	03250
10's complement of N =	+ 27468
Sum =	30718

There is no end carry.

Answer: -(10's complement of 30718) = -69282

Note that since 3250 < 72532, the result is negative. Since we are dealing with unsigned numbers, there is really no way to get an unsigned result for this case. When subtracting with complements, the negative answer is recognized from the absence of the end carry and the complemented result. When working with paper and pencil, we can change the answer to a signed negative number in order to put it in a familiar form.

Example: Given the two binary numbers X = 1010100 and Y = 1000011, perform the subtraction (a) X - *Y* and (b) Y - *X* using 2's complements.

Solution:

(a)

X =	1010100
2's complement of Y =	+ 0111101
Sum =	10010001
Discard end carry 2^7 =	-10000000
Answer: X - Y =	0010001

(b)

Y =	1000011
2's complement of X =	+ 0101100
Sum =	1101111

There is no end carry.

Answer: Y - X = - (2's complement of 1101111) = -0010001

Example: Given the two binary numbers X = 1010100 and Y = 1000011, perform the subtraction (a) X - *Y* and (b) Y - *X* using 1's complements.

Solution:

(a) X - Y = 1010100 - 1000011

X =	1010100
1's complement of Y =	+ 0111100
Sum =	10010000
End-around carry	+ 1
Answer: X - Y =	0010001

(b) Y - X = 1000011 - 1010100

Y =	1000011
1's complement of X =	+ 0101011
Sum =	1101110

There is no end carry.

Answer: $Y - X = -$ (*1's* complement of 1101110) = -0010001

Note that the negative result is obtained by taking the 1's complement of the sum since this is the type of complement used. The procedure with end-around carry is also applicable for subtracting unsigned decimal numbers with 9's complement.

Signed Binary Numbers

•In ordinary arithmetic, a negative number is indicated by a minus sign and a positive number by a plus sign. Because of hardware limitations, computers must represent everything with binary digits, commonly referred to as *bits*. It is customary to represent the sign with a bit placed in the leftmost position of the number. The convention is to make the sign bit 0 for positive and 1 for negative.

•Symbol (+ or -) or a bit (0 or 1) indicating the sign.

• Example: 9 1001 unsigned binary

+9 01001 signed binary or 0000 1001

-9 11001 signed number or 1111 1001

Arithmetic Addition For Signed Binary Numbers

The addition of two numbers in the signed magnitude system follows the rules of ordinary arithmetic.

The addition of two signed binary numbers with negative numbers represented in signed-2's-complement form is obtained from the addition of the two numbers, including their sign bits. A carry out of the sign-bit position is discarded.

Example: Add the following <u>signed</u> numbers?

+ 6	- 6	+ 6	- 6
+13	+ 13	- 13	- 13

Solution: + 6	0000 0110	-6	1111 1010
+ 13	0000 1101	+13	0000 1101
+19	0001 0011	+7	0000 0111 (over flow)
	(Not Over Flow)		

+6	0000 0110	- 6	1111 1010
-13	1111 0011	-13	1111 0011
-7	1111 1001	-19	1110 1101 (over flow)
	(Not Over Flow)		

Arithmetic Subtraction For Signed Binary Number

- Subtraction of two signed binary numbers when negative numbers are in 2's-complement form is very simple and can be stated as follows:

Take the 2's complement of the subtrahend (including the sign bit) and add it to the minu-end (including the sign bit). A carry out of the sign-bit position is discarded.

- This procedure occurs because a subtraction operation can be changed to an addition operation if the sign of the subtrahend is changed. This is demonstrated by the follow-ing relationship:

$$(\pm A) - (+B) = (\pm A) + (-B) \quad \text{---------}1$$
$$(\pm A) - (-B) = (\pm A) + (+B) \quad \text{----------}2$$

Example: Consider the subtraction of (-6) – (-13)=? according to arithmetic subtractions rules for signed binary number?

Solution: (1111 1010) – (1111 0011)=00000111 it is not arithmetic subtraction

(-6) + (+13)=(1111 1010) + (0000 1101) = 00000111

Codes

Binary Code

1- BCD (Binary Coded Decimal)

It composed of four bits representing the decimal digits 0 through 9.

Decimal	(Normal) Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
29	11101	0010 1001
30	11110	0011 0000

Example: Convert each of the following numbers to BCD ?

1. $(18)_{10}$
2. $(15.3)_{10}$
3. $(1976.84)_{10}$

Solution: 1. $(18)_{10} \rightarrow (0001\ 1000)_{BCD}$

2. $(15.3)_{10} \rightarrow (0001\ 0101.\ 0011)_{BCD}$

3. $(1976.84)_{10} \rightarrow (0001\ 1001\ 0111\ 0110.\ 1000\ 0100)_{BCD}$

2- Gray Code

Digital systems can be designed to process data in discrete form only. Many physical systems supply continuous output data. These data must be converted into digital form before they are applied to a digital system. Continuous or analog information is converted into digital form by means of an analog-to-digital converter. It is sometimes convenient to use the Gray code shown in Table 1-4 to represent the digital data when it is converted from analog data. The advantage of the Gray code over binary numbers is that only one bit in the code group changes when going from one number to the next. For example, in going from 7 to 8, the Gray code changes from 0100 to 1100.Only the first bit from the left changes from 0 to 1; the other three bits remain the same. When comparing this with binary numbers, the change from 7 to 8 will be from 0111 to 1000, which causes all four bits to change values.

Table 1-4 Four –bit Gray Code

Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

Binary To Gray

Down the first binary number and compare each one with the other. If

Same = 0 but Difference=1

Example: Convert the following binary numbers to Gray Code?

1. $(1010110)_2 \rightarrow$ ( )Gary Code

2. $(1101101)_2 \rightarrow$ ( )Gary Code

Solution: 1. (1 0 1 0 1 1 0)

1 1 1 1 1 0 1

2. 1 1 0 1 1 0 1

1 0 1 1 0 1 1

Gray To Binary

Down the first binary number and compare each one with the above. If

Same = 0 but Difference=1

Example: Convert the following Gary Codes to binary?

1. (1 1 1 1 1 0 1) Gary Code → ($)_2$
2. (1 0 1 1 0 1)Gary Code → ($)_2$

Solution: 1. (1 1 1 1 1 0 1) 2. 1 0 1 1 0 1 1

1 0 1 0 1 1 0 1 1 0 1 1 0 1

Binary Storage And Registers

Binary Cell is a device that possesses two stable states and is capable of storing one bit of information.

Registers is a group of binary cells. The state of a register is an n-tuple number of 1's and 0's.

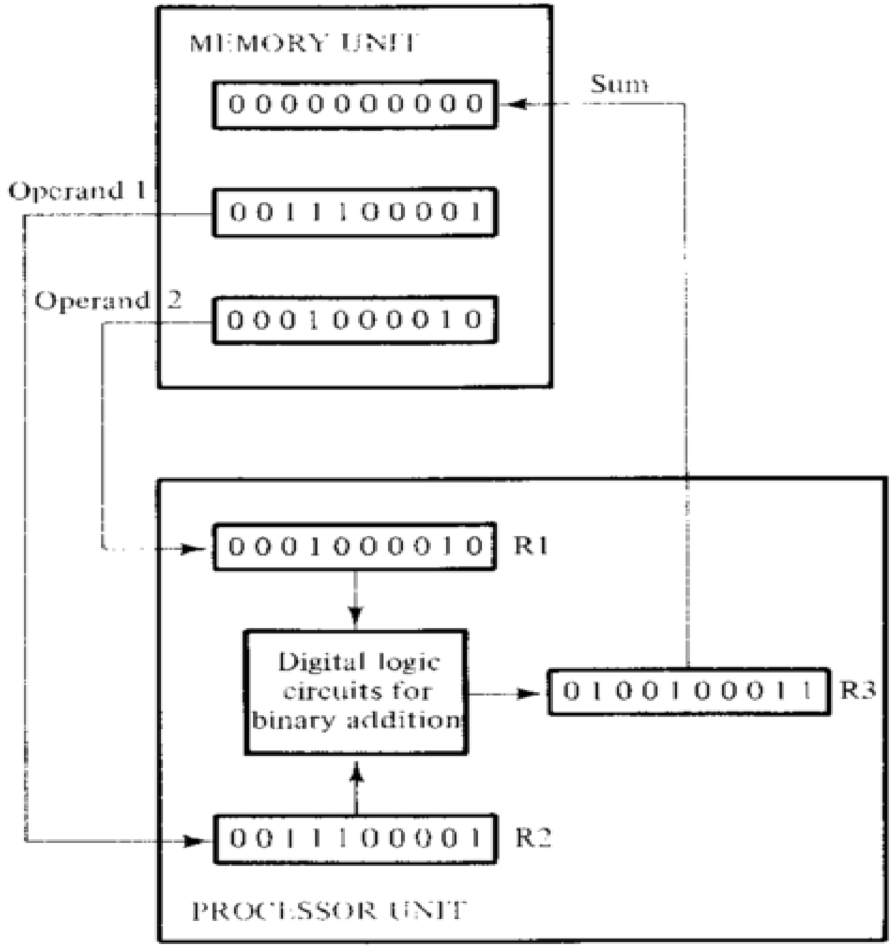
Example: Configure 1100001111001001 in the register?

Solution: This information can be represent at 16-cell register as follows.

1	1	0	0	0	0	1	1	1	1	0	0	1	0	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Example: Represent Binary information processing for adding two 10-bit binary numbers? R1=0001000010 and R2=0011100001

Solution:



Binary Information Processing

Binary Logic

- Binary logic is used to describe in a mathematical way. It is particularly suited for analysis and design digital systems.
- The binary logic to be introduced in this section is equivalent to algebra called Boolean Algebra.
- Binary Logic consists of binary variables and logical operations.
- Variables such as A,B,C, x, y, z,…etc with each variable having values 1 or 0
- Logical operations are AND , OR, and NOT

1- **AND :** This operation is represented by dot or by absence of an operator

For example, X . Y= Z or X Y= Z is read "X AND Y is equal to Z".

2- **OR:** This operation is represented by plus sign. For example, X+Y=Z is read "X OR Y is equal to Z"

3- **NOT**: This operation is represented by prime (sometimes by bar). For example, X' =Z (or X=Z) is read "NOT X is equal to Z" meaning that Z is what X is not.

Note: In binary arithmetic we have 1+1=10 (read: "one plus one is equal to Two")

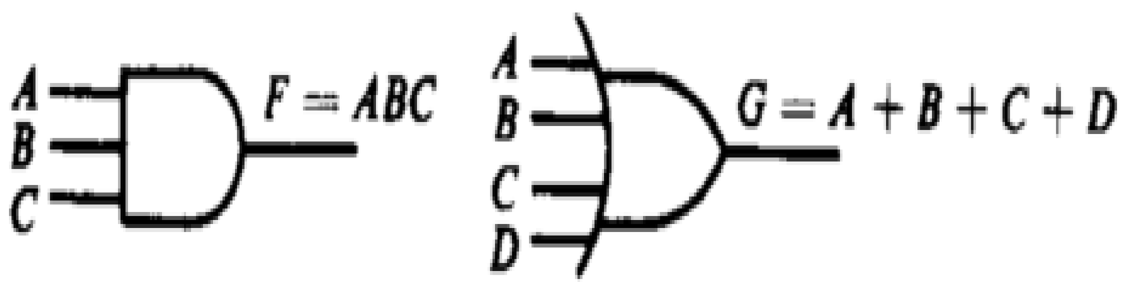
But in binary logic we have 1+1=1 (read: "one OR one is equal to one")

Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

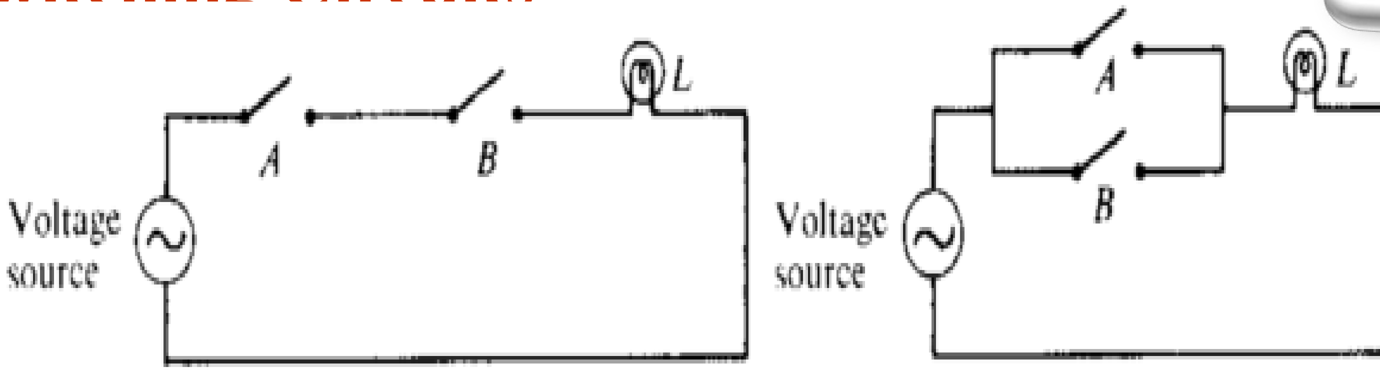
Logic Gates

(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter



(d) Three-input AND gate (e) Four-input OR gate

Switching Circuits



(a) Switches in series – logic AND

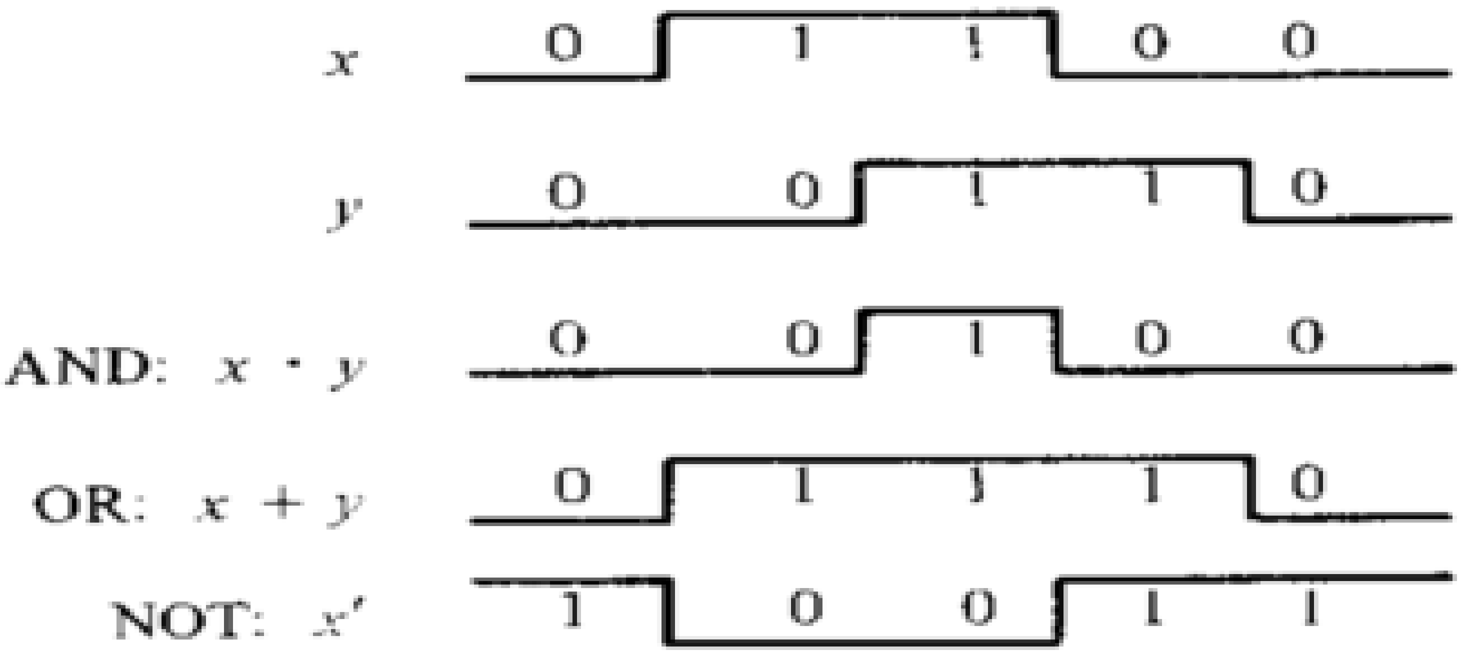
(b) Switches in parallel – logic OR

L= A.B

L= A+B

- Electronic digital circuits are sometimes called *switching* circuits because they behave like a switch, with the active element such as a transistor either conducting (switch closed) or not conducting (switch open).

Input – Output Signal for Gates



Summary

- Computer Number Systems:
  - 1- Decimal System.
  - 2- Binary System.
  - 3- Octal System.
  - 4- Hexadecimal System.

- Conversions between number systems
- Binary Arithmetic:
  - 1- Addition.
  - 2- Subtraction.
  - 3- Multiplication.

- Complements :

    1- 1's and 2's complement for binary number.

    2- 9's and 10's complement for decimal number.

- Signed Binary Numbers
- Arithmetic addition for signed binary number.
- Arithmetic subtraction for signed binary number.

- Codes (Binary Code):

    1- BCD

    2- Gray Code

    3- Error Detection Code (Parity Bit)

- Conversions between (Binary to BCD and vice versa) and (Binary to Gray Code and vice versa).

- Binary Storage and Registers

- Binary Logic:

  1- AND

  2- OR

  3- NOT

- Switching Circuits
- Input – Output Signal for Gates

Chapter One Examination

Quiz 1

Quiz Title: **Digital Computers & Digital Systems**

Quiz Duration: **30 minutes**