Digital Techniques

Digital Computer and Digital System:

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

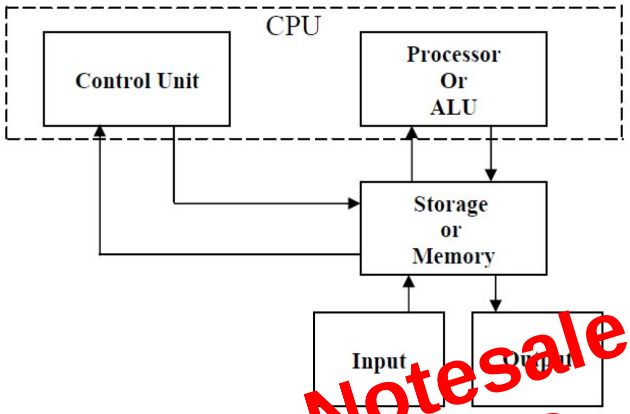


Figure (1) Digital Computer

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.

- Decimal to Octal conversions:

- $(30.5)_{10} \longrightarrow (?)_8$

8	30	6	LSB	$0.5 \times 8 = 4.0$
8	3	3	MSB	

The answer is $(36.4)_8$

- Hexadecimal to decimal conversions:

<u>EX</u>:

- $(1BF)_{16} \longrightarrow (?)_{10}$

$F \times 16^0 + B \times 16^1 + 1 \times 16^3 = 447$

The answer is $(447)_{10}$

- Decimal to Hexadecimal conversions:

<u>EX:</u>

- $(28.3)_{10} \longrightarrow (?)_{16}$

16	28	12	LSB	$0.3 \times 16 = 4.8$	4
16	1	1	MSB	$0.8 \times 16 = 12.8$	12
				$0.8 \times 16 = 12.8$	12

The answer is $(1C.4CC)_{16}$

<u>EX</u>: Subtract using 1's complement:

M	1000100	⇒	1000100
N	1010100 −		0101011
	no carry	→	☐ 1101111

Take 1's complement to the result and the answer is - 0010000.

Logic Gate

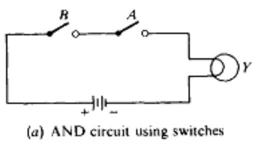
5.1 INTRODUCTION

The logic gate is the basic building block in digital systems. Logic gates operate with binary numbers. Gates are therefore referred to as binary logic gates. All voltages used with logic gates will be either HIGH or LOW. In this lecture, a HIGH voltage will mean a binary 1. A LOW voltage will mean a binary 0. Remember that logic gates are electronic circuits. These circuits will respond only to HIGH voltages (called 1s) or LOW (ground) voltages (called 0s).

All digital systems are constructed by using only three basic logic gates. These basic gates are called the AND gate, the OR gate, and the NOT gate. This chapter deals with these very important basic logic gates, or functions.

5.2 THE AND GATE

The AND gate is called the "all or nothing" gate. The schematic in Fig. 5.la shows the idea of the AND gate. The lamp (Y) will light only when both input switches (A and B) are closed. All the possible combinations for switches A and B are shown in Fig. 5.lb. The table in this figure is called a truth table. The truth table shows that the output (Y) is enabled (lit) only when both inputs are closed.



(a) AND circuit using switches

Input switches		Output light
B	A	Y
open	open	no
open	closed	no
closed	open	no
closed	closed	yes

(b) Truth table

Fig. 5.1

The laws of Boolean algebra govern how AND gates operate. The formal laws for the AND function are:

$$A \cdot 0 = 0$$
$$A \cdot 1 = A$$
$$A \cdot A = A$$
$$A \cdot \overline{A} = 0$$

5.3 THE OR GATE

The OR gate is called the "any or all" gate. The schematic in Fig. 5.3a shows the idea of the OR gate. The lamp (Y) will glow when either switch A or switch B is closed. The lamp will also glow when both switches A and B are closed. The lamp (Y) will no2 glow when both switches (A and B) are open. All the possible switch combinations are shown in Fig. 5.3b. The truth table details the OR function of the switch and lamp circuit. The output of the OR circuit will be enabled (lamp lit) when any or all input switches are closed

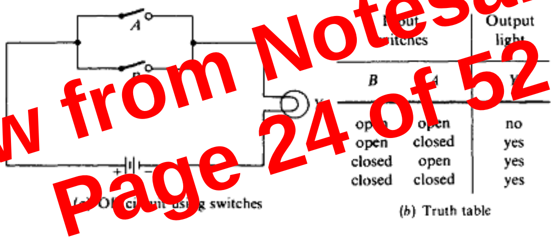
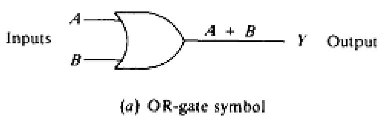


Fig. 5.3

The standard logic symbol for an OR gate is drawn in Fig. 5.4a. Note the different shape of the OR gate. The OR gate has two inputs labeled A and B. The output is labeled Y. The shorthand Boolean expression for this OR function is given as A + B = Y. Note that the plus (+) symbol means OR in Boolean algebra. The expression (A + B = Y) is read as A OR (+ means OR) B equals output Y. You will note that the plus sign does not mean to add as it does in regular algebra.



(a) OR-gate symbol

Inputs		Output
B	A	Y
0	0	0
0	1	1
1	0	1
1	1	1

0 = low voltage
1 = high voltage

(b) OR truth table

Fig 5.4

Digital Techniques

USING PRACTICAL LOGIC GATES

Logic functions can be implemented in several ways. In the past, vacuum-tube and relay circuits performed logic functions. Presently tiny integrated circuits (ICs) perform as logic gates. These ICs contain the equivalent of miniature resistors, diodes, and transistors.

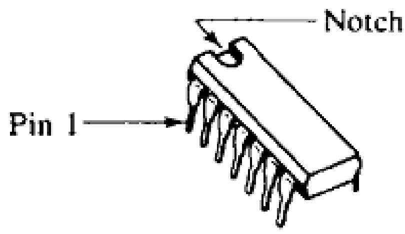


Figure 6.1: 14-pin DIP integrated circuit

A popular type of IC is illustrated in Fig. 6.1. This case style is referred to as a dual-in-line package (DIP) by IC manufacturers, This particular IC is called a 14-pin DIP integrated circuit. Note that immediately counterclockwise from the notch on the IC shown in Fig. 6.1 is pin number 1. The pins are numbered counterclockwise from 1 to 14 when viewed from the top of the IC. Manufacturers of ICs provide pin diagrams similar to the one in Fig. 6.2 for a 7408 IC. Note that this IC contains four 2-input AND gates; thus it is called a quadruple 2-input AND gate. Figure 6.2 shows the IC pins numbered from 1 through 14 in a counterclockwise direction from the notch. The power connections to the IC are the GND (pin 7) and Vcc (pin 14) pins. All other pins are the inputs and outputs to the four AND gates. The 7408 IC is part of a family of logic devices. It is one of many

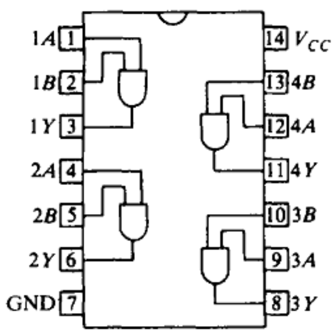


Fig. 6.2: Pin diagram for a 7408 IC

Digital Techniques

Boolean Algebra

Boolean algebra is a form of symbolic logic that shows how logic gates operate. A Boolean expression is a “shorthand” method of showing what is happening in a logic circuit.

Rules of Boolean algebra:

The following propositions are correct in and basic to Boolean algebra:

$A+1 = 1$	$A+\bar{A} = 1$	$A+0 = A$	$A+A = A$
$A \cdot 0 = 0$	$A \cdot \bar{A} = 0$	$A \cdot A = A$	$A \cdot 1 = A$
$\bar{\bar{A}} = A$	$A+AB=A$	$A+\bar{A}B=A$	$(A+B) \cdot (A+C) = A + BC$

Laws of Boolean algebra:

- The Associative law:

  $(a + b) + c = a + (b + c)$

  $(a.b) . c = a. (b.c)$

- The Commutative law:

  $a + b = b + a$

  $a . b = b . a$

- The Distributive law:

  $a (b + c) = ab + ac$

  $a + bc = (a + b)(a + c)$

Subject: Logic Techniques

Tutorial Sheet No.1

Computer and System Eng. Dept.

First class / 2009 - 2008

1. Convert the following decimal numbers to binary numbers:

$(572)_{10} \longrightarrow (\quad)_2$

$(72)_{10} \longrightarrow (\quad)_2$

$(127)_{10} \longrightarrow (\quad)_2$

$(255)_{10} \longrightarrow (\quad)_2$

$(17.325)_{10} \longrightarrow (\quad)_2$

$(9.152)_{10} \longrightarrow (\quad)_2$

$(0.572)_{10} \longrightarrow (\quad)_2$

2. Convert the following binary numbers to decimal numbers:

$(110101)_2 \longrightarrow (\quad)_{10}$

$(100011)_2 \longrightarrow (\quad)_{10}$

$(11101)_2 \longrightarrow (\quad)_{10}$

$(0.1010)_2 \longrightarrow (\quad)_{10}$