

DBIT15



ANNAMALAI UNIVERSITY
DIRECTORATE OF DISTANCE EDUCATION

B.Sc INFORMATION TECHNOLOGY

FIRST YEAR

FUNDAMENTALS OF DIGITAL COMPUTERS

**Copyright Reserved
(For Private Circulation Only)**

Fundamentals of digital computers

Table of content

| Unit – I | Page no |
|---|----------------|
| 1.0 Introduction | 1 |
| 1.1 Objective | 1 |
| 1.2 Content | 1 |
| 1.2.1 Boolean Algebra Functions | 1 |
| 1.2.2 Boolean Laws | 2 |
| 1.2.3 Minimization of Boolean Expressions | 4 |
| 1.2.4 Logic Diagrams | 6 |
| 1.2.5 Combination Logic | 10 |
| 1.2.6 Variable entered Map | 22 |
| 1.2.7 Tabular Minimization | 25 |
| 1.2.8 Analysis of logic Schematics | 29 |
| 1.2.9 Synthesis of Combinations functions | 31 |
| 1.3 Revision Points | 32 |
| 1.4 Intext Questions | 32 |
| 1.5 Summary | 32 |
| 1.6 Terminal Exercises | 32 |
| 1.7 Supplementary Materials | 33 |
| 1.8 Assignments | 33 |
| 1.9 Reference Books | 33 |
| 1.10 Learning Activities | 33 |
| 1.11 Keywords | 33 |
| Unit-II | |
| 2.0 Introduction | 34 |
| 2.1 Objective | 34 |
| 2.2 Content | 34 |
| 2.2.1 Arithmetic-Logic Unit (ALU) | 34 |
| 2.2.2 Control Unit | 47 |
| 2.2.3 Micro Programming | 56 |
| 2.3 Revision Points | 61 |
| 2.4 Intext Questions | 61 |
| 2.5 Summary | 61 |
| 2.6 Terminal Exercises | 62 |
| 2.7 Supplementary Material | 62 |
| 2.8 Assignments | 62 |
| 2.9 Reference Books | 62 |
| 2.10 Learning Activities | 62 |
| 2.11 Keywords | 62 |

Unit-III

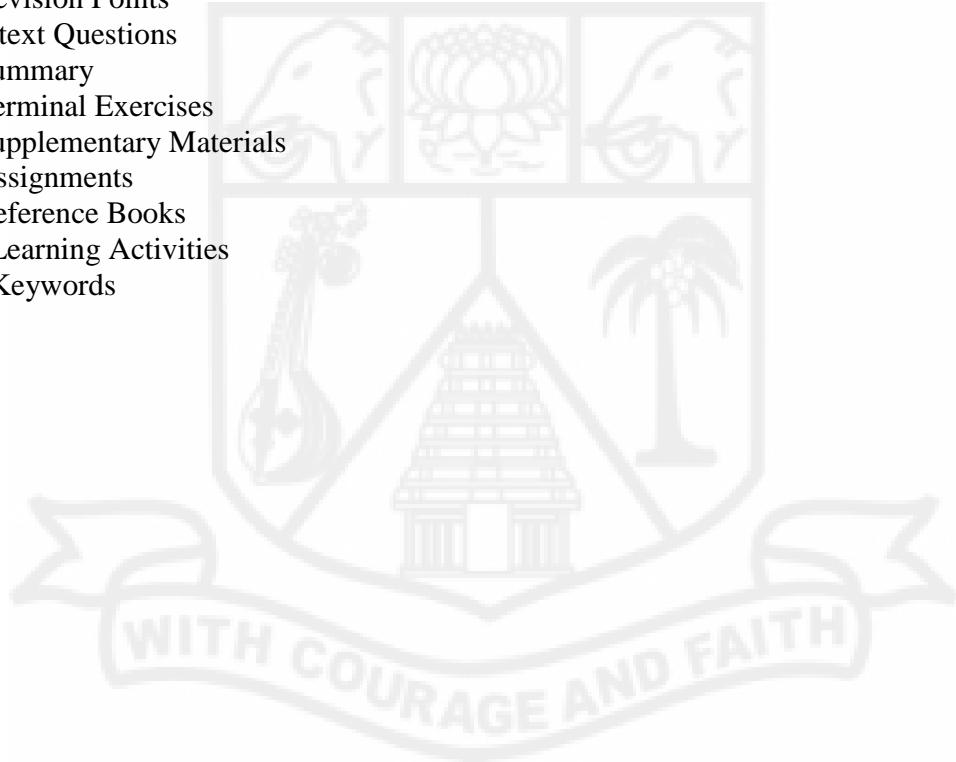
| | |
|---|----|
| 3.0 Introduction | 63 |
| 3.1 Objective | 63 |
| 3.2 Content | 63 |
| 3.2.1 Basic concepts | 63 |
| 3.2.2 Microprocessor Programming | 64 |
| 3.2.3 Instruction Set | 65 |
| 3.2.4 Instruction Types | 65 |
| 3.2.5 Addressing Modes | 68 |
| 3.2.6 Microprocessor Types And Applications | 70 |
| 3.3 Revision Points | 73 |
| 3.4 Intext Questions | 73 |
| 3.5 Summary | 73 |
| 3.6 Terminal Exercises | 73 |
| 3.7 Supplementary Materials | 73 |
| 3.8 Assignments | 74 |
| 3.9 Reference Books | 74 |
| 3.10 Learning Activities | 74 |
| 3.11 Keywords | 74 |

Unit-IV

| | |
|---|-----|
| 4.0 Introduction | 75 |
| 4.1 Objective | 75 |
| 4.2 Content | 76 |
| 4.2.1 Random Access Memories | 76 |
| 4.2.2 Organization | 77 |
| 4.2.3 Connecting Memory Chips To A Computer Bus | 81 |
| 4.2.4 Random Access Memory | 85 |
| 4.2.5 Mass Storage | 87 |
| 4.2.6 Storage Hierarchy | 102 |
| 4.2.7 Digital Recording Techniques | 104 |
| 4.2.7 Virtual Memory | 109 |
| 4.2.8 Cache Memory | 114 |
| 4.3 Revision Points | 119 |
| 4.4 Intext Questions | 119 |
| 4.5 Summary | 119 |
| 4.6 Terminal Exercises | 120 |
| 4.7 Supplementary Materials | 120 |
| 4.8 Assignments | 120 |
| 4.9 Reference Books | 120 |
| 4.10 Learning Activities | 120 |
| 4.11 Keywords | 120 |

Unit-V

| | |
|----------------------------------|-----|
| 5.0 Introduction | 121 |
| 5.1 Objective | 121 |
| 5.2 Content | 121 |
| 5.2.1 Buses And Interfaces | 121 |
| 5.2.2 Keyboard | 123 |
| 5.2.3 Character recognition | 126 |
| 5.2.4 CRT | 130 |
| 5.2.5 Flat Panel Display Systems | 132 |
| 5.2.6 Printers | 133 |
| 5.2.7 Teletype | 137 |
| 5.3 Revision Points | 141 |
| 5.4 Intext Questions | 141 |
| 5.5 Summary | 141 |
| 5.6 Terminal Exercises | 141 |
| 5.7 Supplementary Materials | 142 |
| 5.8 Assignments | 142 |
| 5.9 Reference Books | 142 |
| 5.10 Learning Activities | 142 |
| 5.11 Keywords | 142 |



ANNAMALAI UNIVERSITY

UNIT- I

1.0) Introduction

Boolean Algebra has and Extensive wide spread application in the design of digital computers. For expressing logic circuit functions, as well as for analyzing and designing logic circuits. Boolean algebra provides a mathematical basis which is almost essential for a proper understanding of digital circuits. In boolean algebra every number Is either 0 or is no negative or factional numbers.

1.1) Objective

At the end of this unit student will get knowledge about the following topics:

- Boolean Algebra
- Laws of Boolean Algebra
- Boolean expressions
- Combination Logic
- Sum-of-products and Product of sums
- Multiple-Output minimization
- Analysis of logic schematics
- Tabular minimization

1.2) Content

1.2.1 Boolean Algebra Functions

Boolean algebra has three basic logic functions. These functions are AND. OR. NOT operations.

AND operation

A logical product (multiplication) of two terms A and B can be expressed as $A \times B$ and Is read as "A and B" or $A \cdot B$ " which is read as "A dot B". This logical product represents an AND operation and the terms are said to be AND ed.

OR Operation

OR operation indicates the logical sum of terms $A + B$ represents a logical sum and is read as A or B.

NOT operation

In boolean algebra we have an operation called complementation and the symbol we use is \sim . Thus we write ' X' meaning "take the complement of X" and read as "NOT X'. The process of complementing is called negation.

1.2.2 Boolean Laws

Boolean laws have made it possible to design and analyze logic circuits mathematically. There are several laws of Boolean algebra which will be discussed in the following section.

(a) Laws of Intersection

Rule 1: $A \cdot 1 = A$

Rule 2: $A \cdot 0 = 0$

Proof for Rule 1

The variable A can have only the value 0 or 1. If A has the value 1 Then $1 \cdot 1 = 1$; if A has the value 0 then $0 \cdot 1 = 0$ (Refer Table 1.1).

Proof for Rule 2

Let $A = 1$ Then $1 \cdot 0 = 0$; Let $A = 0$ Then $0 \cdot 0 = 0$ (Refer Table 1.1)

(b) Laws of Union

Rule 3: $A + 1 = 1$

When $A = 0$; $A+1 = 0+1 = 1$ (Refer Table 1.1)

Rule 4: $A + 0 = 0$

When $A = 0$; $A+0 = 0+0 = 0$ (Refer Table 1.1)

(c) Laws of Tautology

Rule 5: $A \cdot A = A$

When $A = 0$; $A \cdot A = 0 \cdot 0 = 0$

Rule 6: $A+A=A$

When $A = 1$; $A \cdot A = 1 \cdot 1 = 1$

(d) Laws of Complements

Rule 7: $A \cdot \overline{A} = 0$

Table 1.1

Logic Function Truth Table

| AND (A . B) | | | Logical Multiplication (1) 0.0 = 0 (2) 0.1 = 0 (3) 1.0 = 0 (4) 1.1 = 1 |
|-------------------|---|---|--|
| A | B | C | |
| 0 | 0 | 0 | |
| 0 | 1 | 0 | |
| 1 | 0 | 0 | |
| 1 | 1 | 1 | |
| OR (A+B) | | | Logical Addition (1) 0 + 0 = 0 (2) 0 + 1 = 1 (3) 1 + 0 = 1 (4) 1 + 1 = 1 |
| A | B | C | |
| 0 | 0 | 0 | |
| 0 | 1 | 1 | |
| 1 | 0 | 1 | |
| 1 | 1 | 1 | |
| NOT (\bar{A}) | | | Logical Complementation $\bar{0} = 1$ $\bar{1} = 0$ |
| A | B | C | |
| 0 | 0 | | |
| 1 | 1 | | |

When A= 1; A . A = 1.0 = 0

Rule8: $A + \bar{A} = 1$

When A=0; $0 + \bar{A} = X = 1$

(e) Law of Double negation

Rule $A = \bar{\bar{A}}$

When $A=0; \bar{\bar{A}}=1; \bar{A}=0$ (ReferTable1.1)

When $A = 1; \bar{A} = 1;$

(f) Laws of Commutation

Rule 10: $A . B = B . A$

Rule 11: $A+B=B+A$

(g) Laws of Association

Rule 12: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

Rule 13: $(A + B) + C = A + (B + C)$

(h) Laws of Distribution

Rule 14: $A \cdot B + A \cdot C = A \cdot (B + C)$

Rule 15: $(A + B) (A + C) = A + (B \cdot C)$

(I) Laws of Absorption

Rule 16: $A(A + B) = A$

Rule 17: $A + A \cdot B = A$

Rule 18: $A(\overline{A} + B) = A \cdot B$

Rule 19: $AB + \overline{B} = A + \overline{B}$

Rule 20: $A \cdot B + \overline{B} = A + B$

(J) Demorgan's Theorem

Rule 21: $\overline{A + B} = \overline{A} \cdot \overline{B}$

Rule 22: $\overline{A \cdot B} = \overline{A} + \overline{B}$

1.2.3 Minimization of Boolean Expressions

Minimization of Boolean expressions leads to reduction of Boolean expression. Since each logic operator represents a logic hardware, reduction of a Boolean expression can be reduced by using Boolean laws which we have already considered. A stage may be reached when no further reduction is possible and that will represent the simplest form of logic circuitry which satisfies the Boolean expression.

Let us consider the function

$$F = A \cdot B + (B + C) \quad (1)$$

Let us now simplify the expression

Fundamentals Of Digital Computers

$$F = A \cdot B + (B + C) = A \cdot B + B + C$$

$$= B(A+1)+C \text{ Rule 17}$$

$$= B \cdot 1 + C \text{ Rule 3}$$

$$= B+C \quad (2)$$

From equation (1) and (2) we can know that both are same.

Now let us verify this using truth tables as follows

| A | B | C | A . B | B+C | A .B+B+C |
|---|---|---|-------|-----|----------|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

Example 1

Simplify the following function

$$F = A (A+B) (A+AB)$$

$$= A (A+AB) \text{ Rule 16}$$

$$= A \cdot A (1 + B)$$

$$= A \cdot A \text{ Rule 3}$$

$$F = A \text{ Rule 5}$$

Example 2

Simplify the following function

$$F = \bar{A}BCD + \bar{A}\bar{B}CD + ABC + BCD$$

$$= \bar{A}BC (D + \bar{D}) + BC (A + \bar{D})$$

$$= \bar{A}BC(1) \div BC(A+\bar{D}) \text{ Rule8}$$

$$= \bar{A}BC + BC (A + \bar{D})$$

— — —

$$= BC [A + A + D]$$

$$= BC(1+D) \quad \text{Rule8}$$

Example 3

Simplify the following function

$$\begin{aligned} F &= A \cdot B (A+B) \\ &= (A+B)(A+B) \text{ Rule 22} \end{aligned}$$

$$\begin{aligned} &= AA+AB+AB+BB \quad \text{rule14} \\ &= 0+A \cdot B + A \cdot B + 0 \quad \text{Rule7} \end{aligned}$$

$$= A \cdot B + A \cdot B$$

Example 4

Simplify the following function

$$F = (A + B) (A + C) (B + C)$$

$$= (A + B) + (A + C) + (B + C) \quad \text{Rule22}$$

$$+ B) + (A^* C) + (B^* C) \quad \text{Rule21}$$

$$= AB + AC + BC \quad \text{Rule 9}$$

1.2.4 Logic Diagrams

Logic gates are very important to understand the concepts of logic diagram corresponding to a Boolean expression. Logic gates are used to perform the logical operations like addition, subtraction, division and multiplication. These electronic devices are normally called as GATES because it can accept one or more Inputs but It can produce only one output. That output may be a HIGH or LOW (also called as TRUE or FALSE) condition which is dependent on the inputs.

The logic gates can be classified as BASIC GATES and COMBINATIONAL GATES. AND, OR and NOT gates are called as basic logic gates because they can perform basic logic functions like logical multiplication, logical addition and logical complementation respectively. Then the other gates derived from the basic logic gates are called as combinational gates. NAND, NOR, XOR, XNOR are called combinational gates. Moreover NAND and NOR gates are called universal gates because any basic logic gate functions like AND, OR, NOT can be derived only by using NAND or NOR gates.

In addition to that there are some logic conventions which help to simplify logic

diagrams and make it easier to follow logic systems.

The following are the some of the coimnonly used conventions

1. Positive logic
2. Negative logic

In any electrical system there are two voltage levels, one of which is more positive than the other or one is more negative than the other. The positive terminal is more positive than the negative terminal and negative terminal is more negative than the positive terminal. Defining the logic 1 signal as the more positive of the two logic signals and the logic 0 signal as the more negative of the two signals is known as positive logic. If we assign the binary 1 state to the more negative of the two voltage levels then it is called as negative logic.

In our discussion, positive logic is used. The following table (Fable 1.2) will give you all the details regarding logic gates. Here the truth table is used to summarize the output available under all the combination of inputs. You will notice from the truth table that the input combinations follow the natural binary progression (ascending order). This is to ensure that all possible can be extended to several variables by using the associative law. So all the gates can have more than two inputs.

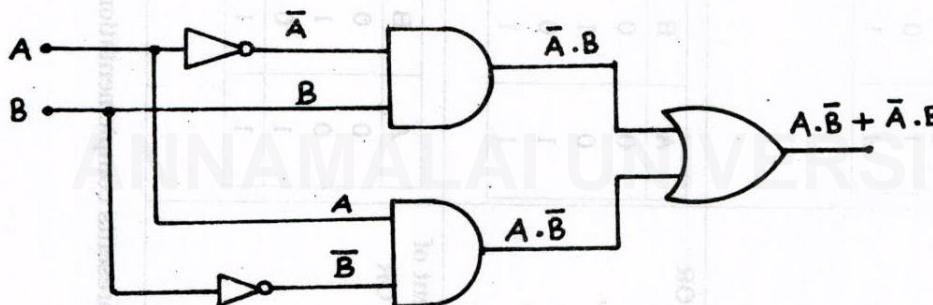
The basic logic gates can be interconnected to form logic networks or logic diagrams, which is also called combinational networks. So we can draw logic diagrams corresponding to any Boolean expression.

For example let us consider Exclusive OR gate.

Boolean expression for XOR gate is

$$y = A \cdot B + \bar{A} \cdot \bar{B}$$

The logic diagram for this boolean expression will be as follows:



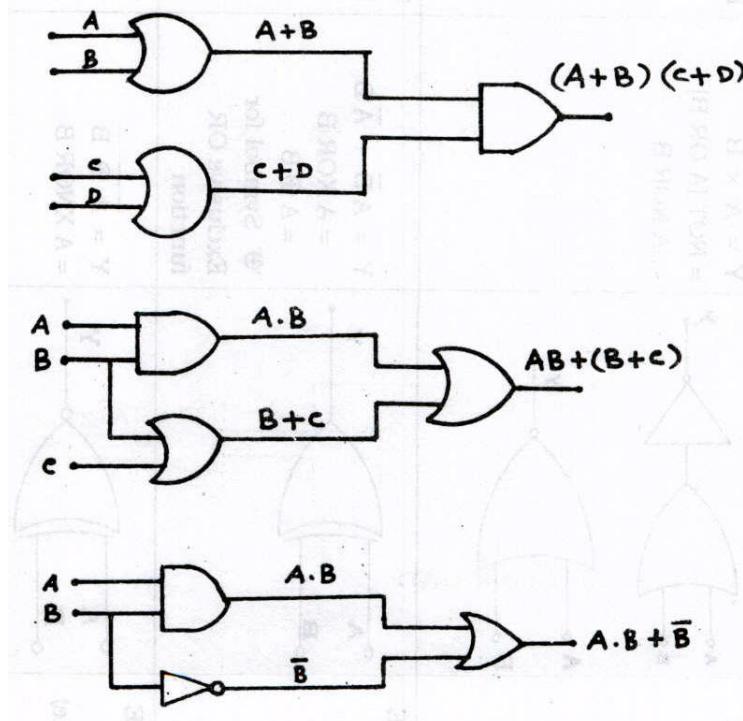
Example

Draw the logic diagram for the following boolean expressions.

(a) $F = (A + B)(C + D)$

(b) $F = A \cdot B + (B + C)$

(C) $F = A \cdot B + B$



Universal Building Blocks

Already we have studied about universal gates. Now let us realize the basic gate functions like AND, OR, and NOT using universal gates (NAND, NOR gates).

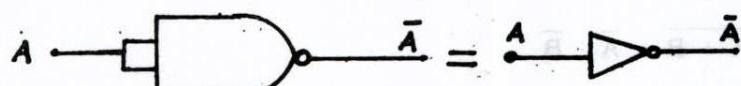
Nand Gate As Universal Gate

Realization of NOT functions using NAND gates

From law 5

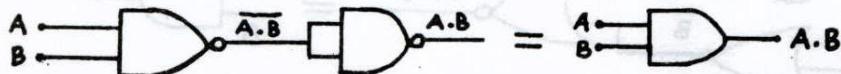
$$A \cdot A = A$$

Complement: $\overline{A \cdot A} = \overline{A} \rightarrow \text{NOT function}$



Realization of AND function using NAND gate

$\overline{A \cdot B}$; complement: $\overline{A \cdot B} = A \cdot \overline{B} \rightarrow$ AND FUNCTION (from law 9)



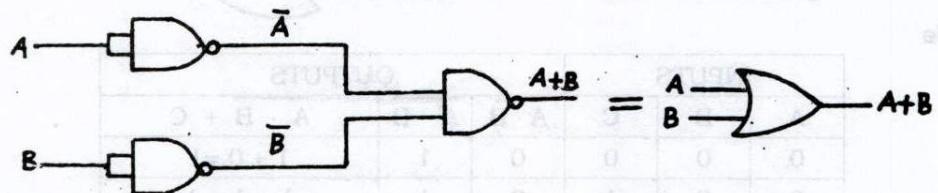
From this we can know that by combining NAND and NAND version NOT gate we can get the AND function.

Realization of OR function using NAND gate

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

$$\text{Manipulate : } A \cdot B = A + B$$

from law 9 : $= A + B \rightarrow$ OR function



NOR gate as universal gate:

Realization of NOT Gate Using NOR Gate

from law 6

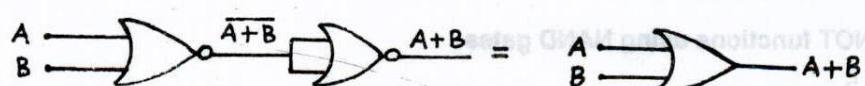
$$A + A = A$$

$$\text{Complement : } \overline{A + A} = \overline{\overline{A} + 0} = A \rightarrow \overline{A} = A \rightarrow \overline{A}$$

Realization of OR gate using NOR gate

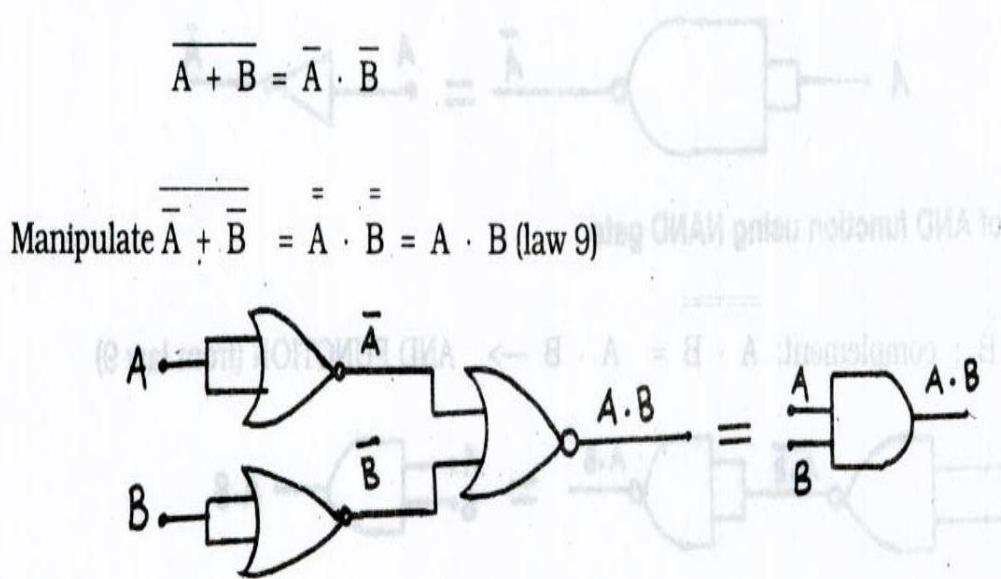
NOR function : $\overline{A + B}$

Complement : $A + B = A + B$ (law 9)



Realizaiton of AND gate using NOR gate

from Demorgan's theorem (law 21)



1.2.5 Combination Logic

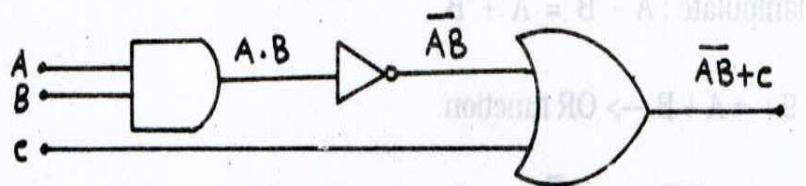
For various decision making function as combination of AND, OR and NOT, logic gates Is employed. The logic circuits so formed are known as combinational logic circuit, For analyzing .the performance of a logic circuit, it is useful to develop a truth table from which the output can be evaluated for all possible combination of input values.
Now let us develop the truth table for the given logic circuit;

Starting from a truth table, it is possible to design logic circuits in the sum-of-products and product-of-sums forms. This will enable to choose simple out of the two logic circuits.

For the main sum of products expression (SOP), consider only the product terms for the rows for which the output is 1. From this we can define the sum-of-products expression as a product term or several product terms logically added.

For the product of sums expressions (POS) we need to consider the rows for which the output is 0. Then write down the product term for that particular row and complement It. This will give you the sum term. After that logically multiply all the sum terms. Now this final expression is called as product of sums.

Logic Circuit



Truth Table

| INPUTS | | | OUTPUTS | | |
|--------|---|---|---------|-------|-----------|
| A | B | C | A · B | A · B | A · B + C |
| 0 | 0 | 0 | 0 | 1 | 1 + 0 = 1 |
| 0 | 0 | 1 | 0 | 1 | 1 + 1 = 1 |
| 0 | 1 | 0 | 0 | 1 | 1 + 0 = 1 |
| 0 | 1 | 1 | 0 | 1 | 1 + 1 = 1 |
| 1 | 0 | 0 | 0 | 1 | 1 + 0 = 1 |
| 1 | 0 | 1 | 0 | 1 | 1 + 1 = 1 |
| 1 | 1 | 0 | 1 | 0 | 0 + 0 = 0 |
| 1 | 1 | 1 | 1 | 0 | 0 + 1 = 1 |

We now take up a circuit having three inputs, for which we will consider the truth table give In Table 1.3 and derive SOP and P0s expressions for the required logic circuit.

Table 1.3

| Inputs | | | Output | Product Terms | Sum Terms |
|--------|---|---|--------|---------------------------------------|---|
| A | B | C | X | | |
| 0 | 0 | 0 | 1 | $\bar{A} \cdot \bar{B} \cdot \bar{C}$ | |
| 0 | 0 | 1 | 1 | $\bar{A} \cdot \bar{B} \cdot C$ | |
| 0 | 1 | 0 | 1 | $\bar{A} \cdot B \cdot \bar{C}$ | |
| 0 | 1 | 1 | 0 | | $\bar{A} + \bar{B} + \bar{C} \rightarrow \bar{A} \bar{B} \bar{C} = A + \bar{B} + \bar{C} = A + \bar{B} + \bar{C}$ |
| 1 | 0 | 0 | 1 | $A \cdot \bar{B} \cdot \bar{C}$ | |
| 1 | 0 | 1 | 1 | $A \cdot \bar{B} \cdot C$ | |
| 1 | 1 | 0 | 0 | | $\bar{A} + \bar{B} + C \rightarrow A \cdot B \cdot C = \bar{A} + \bar{B} + C = \bar{A} + \bar{B} + C$ |
| 1 | 1 | 1 | 0 | | $\bar{A} + \bar{B} + \bar{C} \rightarrow \bar{A} \bar{B} \bar{C} = \bar{A} + \bar{B} + \bar{C}$ |

The sum of products expression will be as follows:

$$F = A \cdot B \cdot C + A \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C}$$

$$= A \cdot B (C + \bar{C}) + A \cdot B (C + \bar{C}) + A \cdot \bar{B} \cdot C$$

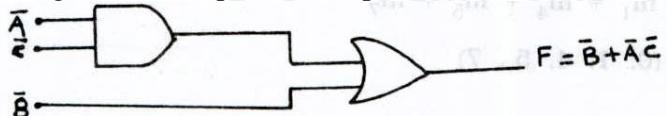
$$= A \cdot B + A \cdot \bar{B} + A \cdot \bar{B} \cdot C$$

$$= B(A + A) + A \cdot \bar{B} \cdot C \quad (\text{Rule 9})$$

$$= B + A \cdot \bar{B} \cdot C$$

$$= B + A \cdot C$$

The logic circuit required to Implement this function will be as follows:



The product of sums expression will be as follows:

$$\begin{aligned} F &= (A + \bar{B} + \bar{C})(\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C}) \\ &= (A + \bar{B} + \bar{C})(\bar{A} + \bar{B}) \\ &= \bar{B}(A + \bar{A}) + \bar{B}(1 + \bar{C}) + (\bar{A} + \bar{C}) \\ &= \bar{B} + \bar{A} \cdot \bar{C} \quad (\text{Refer Boolean laws}) \end{aligned}$$

This expression is same as that of sum of products expressions.

Karnaugh Map

In the previous section we have considered the sum of products and product of sum logic circuits received from truth table. But this method does not always lead to the simplest possible expression. So we are going for Karnaugh Map technique, which Is faster as well as better system for reducing Boolean expressions, which provides a graphical method for this purpose. Now let us discuss In detail about Karnaugh map technique. In mapping technique there are too Important terms. One such term Is referred to as 'Min term' another one is Max term'.

Each mm term is a product of all the variables in the system. A max term Is a sum of all the variables within a logic system. So we can derive an SOP expression from a sum of max terms and POS expressions as product of max terms.

To get clear idea regarding mm term and max term let us consider the following example

Fundamentals Of Digital Computers

| Input | | | Output | Min Term | | Max Term | |
|-------|---|---|--------|---------------------------|-------------|-------------------------------|-------------|
| A | B | C | | Expression | Designation | Expression | Designation |
| 0 | 0 | 0 | 1 | $\bar{A} \bar{B} \bar{C}$ | m_0 | $A + B + C$ | M_0 |
| 0 | 0 | 1 | 1 | $\bar{A} \bar{B} C$ | m_1 | $A + B + \bar{C}$ | M_1 |
| 0 | 1 | 0 | 0 | $\bar{A} B \bar{C}$ | m_2 | $A + \bar{B} + C$ | M_2 |
| 0 | 1 | 1 | 0 | $\bar{A} B C$ | m_3 | $A + \bar{B} + \bar{C}$ | M_3 |
| 1 | 0 | 0 | 1 | $A \bar{B} \bar{C}$ | m_4 | $\bar{A} + B + C$ | M_4 |
| 1 | 0 | 1 | 1 | $A \bar{B} C$ | m_5 | $\bar{A} + B + \bar{C}$ | M_5 |
| 1 | 1 | 0 | 0 | $A B \bar{C}$ | m_6 | $\bar{A} + \bar{B} + C$ | M_6 |
| 1 | 1 | 1 | 1 | $A B C$ | m_7 | $\bar{A} + \bar{B} + \bar{C}$ | M_7 |

Now let us write the SOP and P05 expressions using mm term and max term

$$\begin{aligned} F &= \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} C + A \bar{B} \bar{C} + A \bar{B} C + ABC \\ &= m_0 + m_1 + m_4 + m_5 + m_7 \end{aligned}$$

$$F = \Sigma m (0, 1, 4, 5, 7)$$

POS Expression

$$\begin{aligned} F &= (A + \bar{B} + C) (A + \bar{B} + \bar{C}) (\bar{A} + \bar{B} + C) \\ &= M_2 \cdot M_3 \cdot M_6 \\ &= \Pi M (2, 3, 6) \end{aligned}$$

A Karnaugh map is an alternative for a truth table. It can display in a visual form all the information contained in a truth table.

Two Variable Karnaugh Map

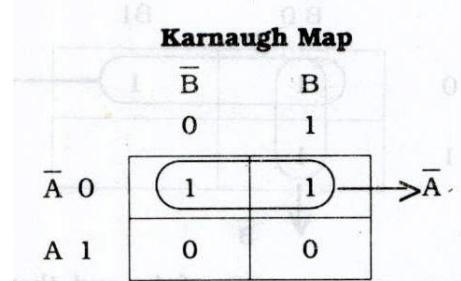
A Karnaugh Map consists of no of squares which corresponds to no of variables present in boolean function. Generally a boolean function of `n variables will require 2^n squares. So a boolean function with two variables will require 2^2 (ie) 4 squares. Then the columns in a map are assigned to one variable and rows are assigned to another variable.

Now let us consider the representation of the following logic function on a Karnaugh Map.

$$F = \bar{A} \bar{B} + \bar{A} B$$

Truth Table

| Input | | Product term | Output Y |
|-------|---|------------------|----------|
| A | B | | |
| 0 | 0 | $\bar{A}\bar{B}$ | 1 |
| 0 | 1 | $\bar{A}B$ | 1 |
| 1 | 0 | $A\bar{B}$ | 0 |
| 1 | 1 | AB | 0 |



Hence the given Boolean expression is

$$F = \bar{A}\bar{B} + \bar{A}B$$

We have to consider the following steps to form a Karnaugh Map.

1. Note down the number of variables present In the expression. In this case 2
2. Then find out the possible Input combinations using 2^n formula. Here $2^2 = 4$. So no of squares in K - Map are 4.
3. Write down the truth table for this. In the output column mark 1 for the product terms present in the given boolean expressions and mark 0 for the rest.
4. Then transfer the truth table to K-Map

K-Map Reduction

Two is, which form a pair, have been enclosed as it is adjacent to each other and from this pair the variable B has been dropped because B has changed from the complemented to the uncompiemneted form In this pair. and variable A which has not changed has been retained. So according to the K-Map reduction procedure, the reduced equations is $F = A$

Let us now reduce using boolean laws.

$$\begin{aligned} F &= \bar{A}\bar{B} + \bar{A}B \\ &= \bar{A}(\bar{B} + B) \\ &= \bar{A} \end{aligned}$$

(Rule 8)

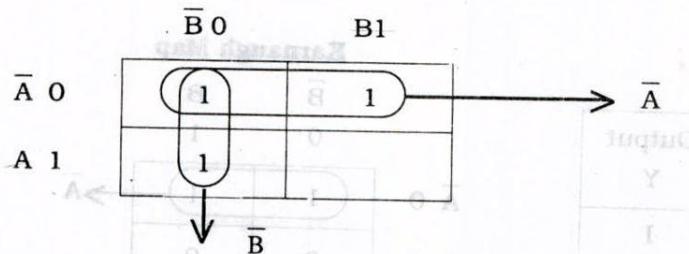
Now we can conclude that both the results are same.

Moreover we can plot a K-map directly from the boolean expression without using truth table as follows.

Example

Reduce the following boolean functions

$$F = \bar{A} \bar{B} + \bar{A} B + A \bar{B}$$



There are two pairs of 1s and they can be closed as shown in the map. It is permissible to use the same 1 more than once. So a 1 has been enclosed twice here. Now drop those variables from each pair, that have complemented and retain those variables that have not complemented and form an SOP equation by ORing them. This gives the following reduced function

$$F = \bar{A} + \bar{B}$$

Three variable functions

A 3 variable function will have or 8 product terms. Having two squares in a row and four in each column or vice versa can accommodate these product terms. Then assign one variable in any row and two variables in any column or vice versa.

In columns as well as rows, only one variable changes from the complemented to the uncomplemented form. This means that there is a change in only one digit as you move vertically or horizontally but not diagonally from one square to another and each square has been given an address which represents its location in the row and column.

Three Variable K-Map

| | $\bar{B} \bar{C}$ | $\bar{B} C$ | $B \bar{C}$ | $B C$ |
|-------------|---|---------------------------------------|---------------------------------------|---------------------------------|
| $\bar{A} 0$ | $\bar{A} \bar{B} \bar{C}$ 0 0 0 m_0 | $\bar{A} \bar{B} C$ 0 0 1 m_1 | $\bar{A} B \bar{C}$ 0 1 1 m_3 | $\bar{A} B C$ 1 0 0 m_2 |
| $A 1$ | $A \bar{B} \bar{C}$ 1 0 0 m_4 | $A \bar{B} C$ 1 0 1 m_5 | $A B \bar{C}$ 1 1 1 m_7 | $A B C$ 1 1 0 m_6 |

Example

Reduce the following Boolean function:-

$$F = \bar{A} \bar{B} \bar{C} + \bar{A} B \bar{C} + A B \bar{C} + A \bar{B} C$$

In K-map 1's have been entered In square which correspond to product terms

| | | $\bar{B} \bar{C}$ | $\bar{B} C$ | $B \bar{C}$ | $B C$ |
|-------------|---|-------------------|-------------|-------------|-------|
| | | 0 0 | 0 1 | 1 1 | 1 0 |
| | | $\bar{A} 0$ | | | |
| $\bar{A} 0$ | 1 | | | | 1 |
| | 0 | | 1 | 3 | 2 |
| $A 1$ | | 1 | | 1 | 6 |
| | 4 | 5 | 7 | | |

- As min term m_5 cannot form a group with any other mm term. Keep It as it Is.
- Mm terms m_5 and m_6 will form a pair and after dropping a variable A we will get BC
- The squares on the extreme left and right of a row or column can be considered adjacent. if it has only one variable different. So mm terms M_0 and M_2 will form a pair and after dropping variable B we will get AC

The reduced function is as follows:

$$F = A \bar{B} C + B \bar{C} + \bar{A} \bar{C}$$

Four Variable Karnugh Map

A logic function with four variables will have 2⁴ or 16 combinations of variables. The following table gives the fundamental products binary values and mm term designations of a 4 variables logic system.

Fundamentals Of Digital Computers

16

| Input | | | | Min. Term | |
|-------|---|---|---|--------------------------------|--------------|
| A | B | C | D | Fundamental Products | Designations |
| 0 | 0 | 0 | 0 | $\bar{A}\bar{B}\bar{C}\bar{D}$ | m_0 |
| 0 | 0 | 0 | 1 | $\bar{A}\bar{B}\bar{C}D$ | m_1 |
| 0 | 0 | 1 | 0 | $\bar{A}\bar{B}CD$ | m_2 |
| 0 | 0 | 1 | 1 | $\bar{A}\bar{B}C\bar{D}$ | m_3 |
| 0 | 1 | 0 | 0 | $\bar{A}B\bar{C}\bar{D}$ | m_4 |
| 0 | 1 | 0 | 1 | $\bar{A}B\bar{C}D$ | m_5 |
| 0 | 1 | 1 | 0 | $\bar{A}BC\bar{D}$ | m_6 |
| 0 | 1 | 1 | 1 | $\bar{A}BCD$ | m_7 |
| 1 | 0 | 0 | 0 | $A\bar{B}\bar{C}\bar{D}$ | m_8 |
| 1 | 0 | 0 | 1 | $A\bar{B}\bar{C}D$ | m_9 |
| 1 | 0 | 1 | 0 | $A\bar{B}C\bar{D}$ | m_{10} |
| 1 | 0 | 1 | 1 | $A\bar{B}CD$ | m_{11} |
| 1 | 1 | 0 | 0 | $AB\bar{C}\bar{D}$ | m_{12} |
| 1 | 1 | 0 | 1 | $AB\bar{C}D$ | m_{13} |
| 1 | 1 | 1 | 0 | $ABC\bar{D}$ | m_{14} |
| 1 | 1 | 1 | 1 | $ABC\bar{D}$ | m_{15} |

| | | | | |
|--|------------------|------------|------|------------|
| | $\bar{C}\bar{D}$ | $\bar{C}D$ | CD | $C\bar{D}$ |
| | 00 | 01 | 11 | 10 |

| | | | | |
|----------------------|--------------------------------|--------------------------|--------------------|--------------------------|
| $\bar{A}\bar{B}\ 00$ | $\bar{A}\bar{B}\bar{C}\bar{D}$ | $\bar{A}\bar{B}\bar{C}D$ | $\bar{A}\bar{B}CD$ | $\bar{A}\bar{B}C\bar{D}$ |
| m_0 | 0000 | 0001 | 0011 | 0100 |
| $\bar{A}\bar{B}\ 01$ | $\bar{A}B\bar{C}\bar{D}$ | $\bar{A}B\bar{C}D$ | $\bar{A}BC\bar{D}$ | $\bar{A}BC\bar{D}$ |
| m_4 | 0100 | 0101 | 0111 | 0110 |
| $A\bar{B}\ 11$ | $AB\bar{C}\bar{D}$ | $AB\bar{C}D$ | $ABC\bar{D}$ | $ABC\bar{D}$ |
| m_{12} | 1100 | 1101 | 1111 | 1110 |
| $A\bar{B}\ 10$ | $A\bar{B}\bar{C}\bar{D}$ | $A\bar{B}\bar{C}D$ | $A\bar{B}CD$ | $A\bar{B}C\bar{D}$ |
| m_8 | 1000 | 1001 | 1011 | 1010 |

A 4 variable function karnaugh map has four rows and four columns. The rows are assigned to combinations of variables A and B and their complements in such a manner that there is a change in the nature of only one variable as you move from one row to the next In the same column. Columns have been assigned in a same manner to variable C and D

Example

Reduce the following three variable function with the help of Karnaugh map.

$$F = m_0 + m_2 + m_5 + m_7 + m_8 + m_{10} + m_{13} + m_{15}$$

The min terms of this function have been plotted on a Karnaugh map

| | | $\bar{C} \bar{D}$ | $\bar{C} D$ | $C \bar{D}$ | $C D$ | |
|-------------|-----|-------------------|-------------|-------------|-------|---|
| | | 0 0 | 0 1 | 1 1 | 1 0 | |
| | | $\bar{A} \bar{B}$ | 0 0 | 1 | | 1 |
| $\bar{A} B$ | 0 1 | 0 | 1 | 3 | 2 | |
| | 1 | 4 | 1 | 1 | 6 | |
| $A \bar{B}$ | 1 1 | 12 | 13 | 15 | 14 | |
| | 1 0 | 1 | 1 | 1 | 1 | |
| | | 8 | 9 | 11 | 10 | |

Four 1's located In the three corners can be enclosed in one group, as they can be considered to be adjacent. From this group we can eliminate 2 variables as It Is changed. Those are C. C and A. A and retain the variables B and D. Similarly is In the centre of the map can be enclosed from which we can retain the variables B and D. The reduced fraction will be as follows.

$$F = \bar{B} \bar{D} + BD$$

Incompletely specified functions (Don't care states) h the binary coded decimal systems [BCDJ] and decimal equivalent of binary number are using a 4 bit codes In which the following codes never occur:

1010, 1011, 1100, 1101, 1110, 1111.

Fundamentals Of Digital Computers

The outputs corresponding to those binary numbers also do not appear as the output. These states are known as 'Don't care' states and are commonly designated by X In truth tables.

Let us consider a logic system, which will give a binary 1 output for the following decimal numbers in the BCD truth tables.

Let us consider a logic system, which will give a binary 1 output for the following decimal numbers in the BCD systems.

Decimal No BCD Code

| | |
|---|------|
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

First of all draw a truth table with four variables and put 1 against those input states where binary output is desired, 0 against those states where no output is desired, and X against those states which will not occur, while forming pairs Xs that can be treated as Os or Is according to convenience.

| Intr. | | D | Product Term | Decimal No. | Output |
|-------|---|---|-----------------------------------|-------------|--------|
| A | B | | | | |
| | | 0 | $\bar{A} \bar{B} \bar{C} \bar{D}$ | 0 | 0 |
| | | 0 | $\bar{A} \bar{B} \bar{C} D$ | 1 | 0 |
| | 0 | 1 | $\bar{A} \bar{B} C \bar{D}$ | 2 | 0 |
| 0 | 0 | 1 | $\bar{A} \bar{B} C D$ | 3 | 0 |
| 0 | 1 | 0 | $\bar{A} B \bar{C} \bar{D}$ | 4 | 0 |
| 0 | 1 | 0 | $\bar{A} B \bar{C} D$ | 5 | 1 |
| 0 | 1 | 1 | $\bar{A} B C \bar{D}$ | 6 | 1 |
| 0 | 1 | 1 | $\bar{A} B C D$ | 7 | 1 |
| 1 | 0 | 0 | $A \bar{B} \bar{C} \bar{D}$ | 8 | 1 |
| 1 | 0 | 0 | $A \bar{B} \bar{C} D$ | 9 | 1 |
| 1 | 0 | 1 | $A \bar{B} C \bar{D}$ | 10 | x |
| 1 | 0 | 1 | $A \bar{B} C D$ | 11 | x |
| 1 | 1 | 0 | $A B \bar{C} \bar{D}$ | 12 | x |
| 1 | 1 | 0 | $A B \bar{C} D$ | 13 | x |
| 1 | 1 | 1 | $A B C \bar{D}$ | 14 | x |
| 1 | 1 | 1 | $A B C D$ | 15 | x |

Now let us draw the Karnaugh map to find out SOP and POS expression.

SOP function

To derive SOP function we will ignore squares in which Os have been entered and xs are considered as Is.

| | | CD | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
| | | AB | 00 | 0 | 0 | 0 |
| | | | 00 | 1 | 1 | 1 |
| | | | 01 | x | x | x |
| | | | 11 | 1 | 1 | x |
| | | | 10 | 1 | x | x |

$F = A + BC + BD$

POS function

To derive POS function we will ignore squares in which 0's have been entered and X3 are considered as 0'S

| | | CD | 00 | 01 | 11 | 10 |
|--|--|----|----|----|----|----|
| | | AB | 00 | 0 | 0 | 0 |
| | | | 00 | 0 | 1 | 1 |
| | | | 01 | x | x | x |
| | | | 11 | 1 | 1 | x |
| | | | 10 | 1 | x | x |

$\overline{A} \overline{C} \overline{D}$ and $\overline{A} \overline{B}$

Product terms are $\overline{A} \overline{C} \overline{D}$ and $\overline{A} \overline{B}$. These product terms can form the following function in the P05 form

$$\begin{aligned}
 \text{Complement } \overline{\overline{A} \overline{C} \overline{D}} &= \overline{A} + \overline{C} + \overline{D} \\
 &= A + C + D \\
 \text{Complement } \overline{\overline{A} \overline{B}} &= \overline{A} + \overline{B} \\
 &= A + B \\
 F = (A + C + D)(A + B)
 \end{aligned}$$

Note:

Consider don't care (X) if it is necessary otherwise ignore it.

Hybrid Functions

Sum of products and product of sums reduction process produces an input, which feeds an input gate as well as an output gate. Therefore it is called two level logic. Since each input signal has to pass through two gates before reaching the output. Many times it is possible to reduce SOP and POS circuit further by factoring which Produces a Hybrid Circuit.

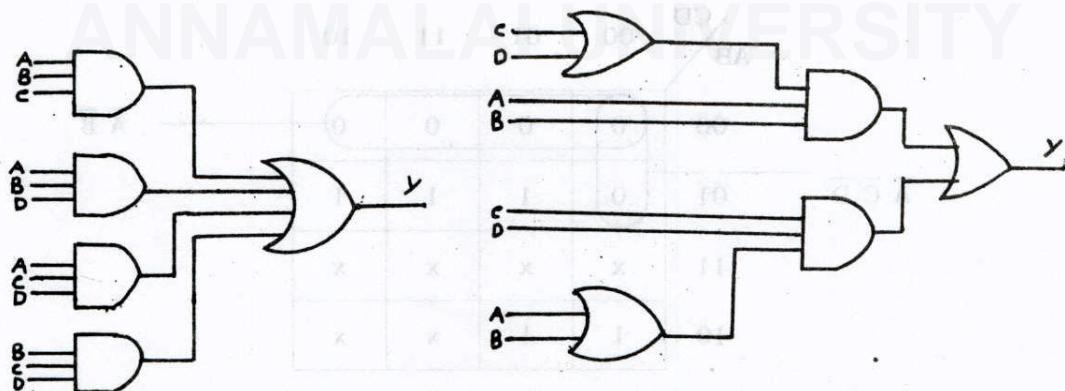
For example. consider the equation

$$F = ABC + ABD + ACD + BCD$$

This is a minimum sum of products form and requires 16 Inputs. The equation can be reduced by factoring as follows.

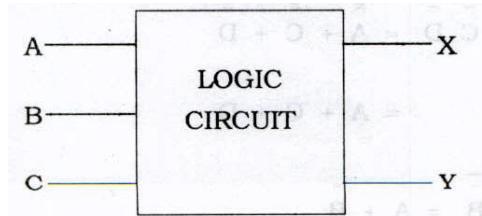
$$\begin{aligned}
 F &= ABC + ABD + ACD + BCD \\
 &= AB(C+D)+CD(A+B)
 \end{aligned}$$

this equation requires only 12 inputs



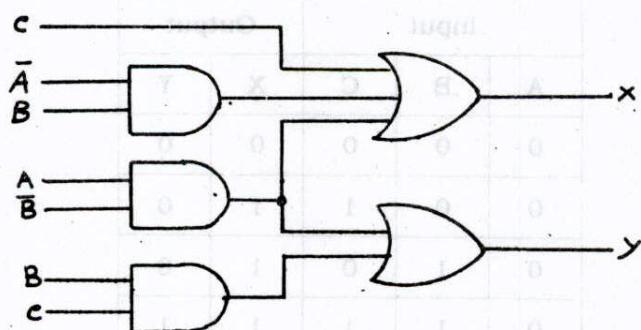
Multiple Output Minimizations

So far we have considered logic systems that have multiple Input and one output. However, there are many systems requiring several outputs. Now let us minimize a dual output function X and Y where input lines are A,B and C.



BLOCK DIAGRAM

Logic Diagram



Hence the term $\bar{A}B$ is common between the X and Y outputs. So sharing of gate is possible. This sharing of gates forms the basis for multiple output minimization.

1.2.6 Variable entered Map

The variable mapping technique is one that allows us to reduce a large mapping to a small map, consider the equation.

This Is a four variable problem. But we can solve this using only 3 variables map.

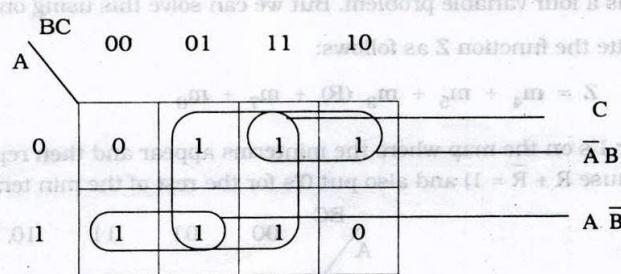
Rewrite the function Z as follows:

$$Z = m_4 + m_5 + m_6 + m_7 + m_8$$

TRUTH TABLE

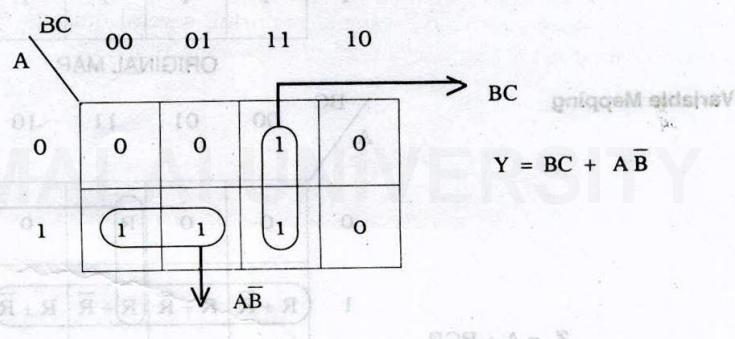
| Input | | | Output | |
|-------|---|---|--------|---|
| A | B | C | X | Y |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

X Map



$$X = C + \bar{A}B + AB'$$

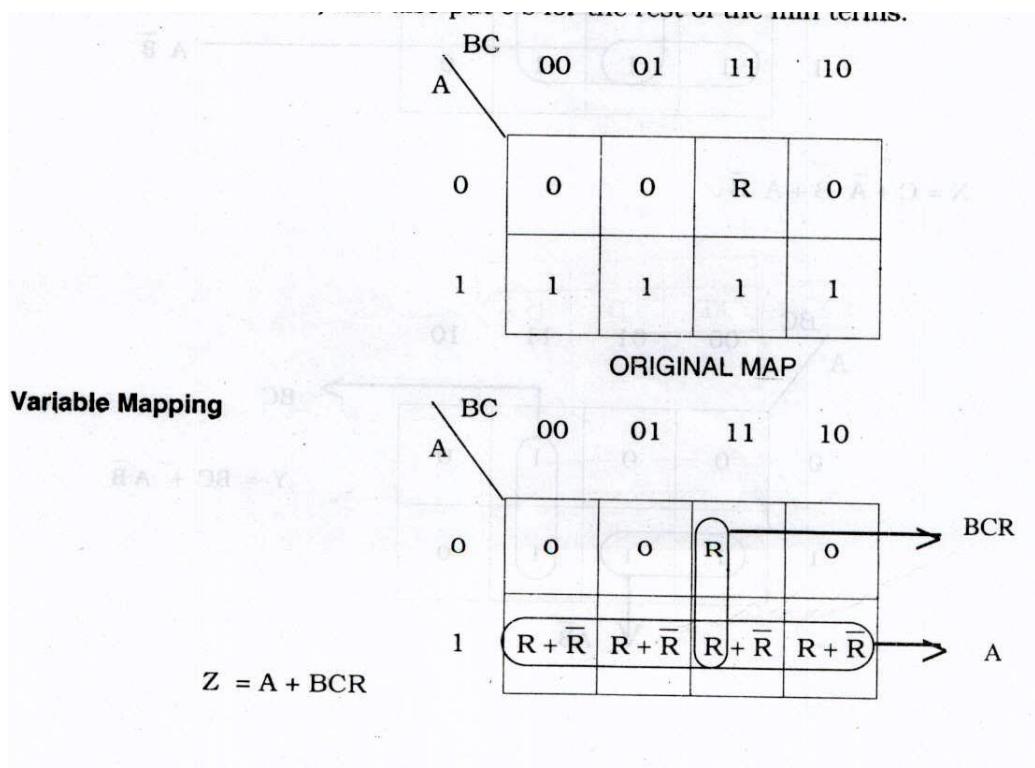
Y Map



$$Z = A\bar{B}\bar{C} + A\bar{B}C + \bar{A}BC(R) + ABC + A\bar{B}\bar{C}$$

Fundamentals Of Digital Computers

Enter 1's on the map where the min terms appear and then replace these 1's using it + R term (because $R + R = 1$) and also put 0's for the rest of the min terms.



Example

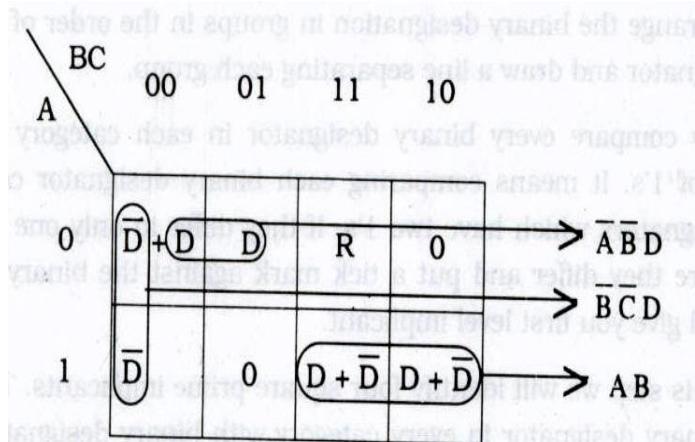
1. Solve using 3 variable map.

$$N = \bar{A} \bar{B} \bar{C} + \bar{A} \bar{B} CD + AB \bar{C} \bar{D} + ABC + ABC$$

We can convert each of the four variable mm terms to 3 variable mm terms multiplied by a D term.

$$N = m_0 + m_1 \bar{D} + m_4 D + m_7 + m_6$$

Now we can map each term, including the D variable on a 3 variable map. All 1's are entered as $D + D$.

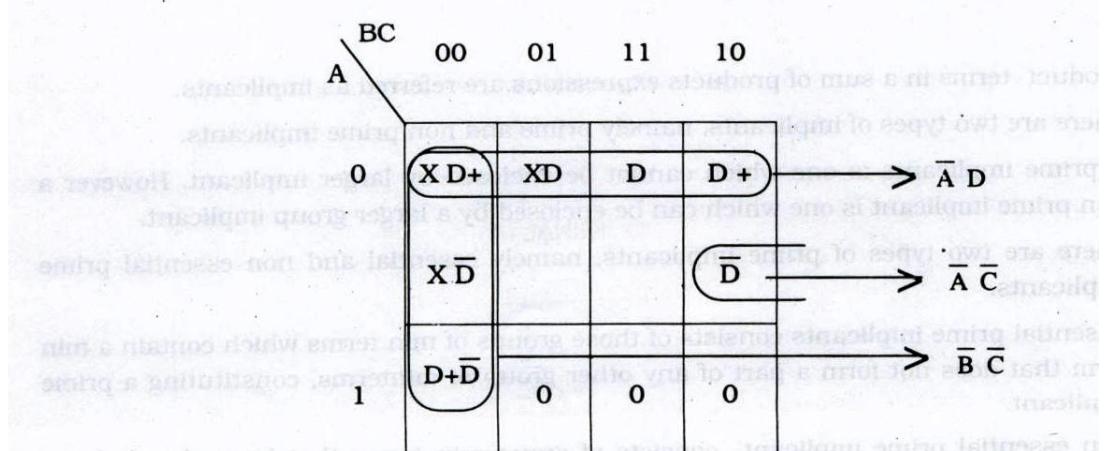


The result Is

$$N = AB + \bar{A}'\bar{B}D + \bar{B}'\bar{C}D$$

2. Solve using a 3 variable map

$$Y = Dm_3 + m_2 + m_4 + d(m_0 \text{ t } Dm_1)$$



The result is $Y = \bar{A}DC + \bar{A}\bar{C} + \bar{B}\bar{C}$

The procedure used for don't care problems has to be modified slightly to accommodate variable mapping. An entry of X on the map represents $XD + X\bar{D}$

1.2.7 Tabular Minimization

The mapping method is convenient when the number of variables is small, But It is a tedious method with a larger number of variables and as computerization is impossible.

Fundamentals Of Digital Computers

To satisfy these conditions we are going for tabular minimization or Quine -Mecluskey's tabular method.

The following are the minimization procedure for the tabular method;

Step 1: Convert all mm term designates to binary designators.

Step 2: Rearrange the binary designation in groups In the order of the number of 1's in each binary designator and draw a line separating each group.

Step 3: Now compare every binary designator In each category of 1's with the next higher category of 1's. It means comparing each binary designator containing one 1 with those binary designators which have two 1's. If they differ In only one position, put an X in the position where they differ and put a tick mark against the binary designator you have checked. This will give you first level implicant.

Step 4: In this step we will identify four square prime implicants. To do this, we have to compare each binary designator In every category,with binary designator in the next higher category of first level implicants. but the following condition must be satisfied In this comparison.

- I) The two Implicants must differ by only one 1.
- ii) Both Implicants must have Xs In the same column.

This will give you second level implicant.

Step 5: In this step we will determine the essential prime Implicants in rows, In a descending order of the squares covered.

NOTE:

1. Product terms in a sum of products expressions are referred as implicants.
2. There are two types of implicants, namely prime and non prime Implicants.
3. A prime implicants is one which cannot be enclosed by larger implicant. However a non prime implicant is one which can be enclosed by a larger group implicant.
4. There are two types of prime implicants, namely essential and non essential prime implicants.
5. Essential prime implicants consists of those groups of mm terms which contain a mm term that does not form a part of any other group of min terms, constituting a prime implicant.
6. Non essential prime implicant consists of group mm terms that have already been covered by other essential prime implicants. An essential prime Implicant is used to see whether It can be removed without leaving any mm term unclosed.

Basic principle of tabular method

| m ₂ | m ₃ | m ₁₀ | m ₁₁ | minterm designators |
|---|----------------|---|-----------------|------------------------|
| 0010 | 0011 | 1010 | 1011 | Binary designators |
| $\bar{A} \bar{B} C \bar{D} + \bar{A} \bar{B} C D$ | | $+ 0 \quad 0 \quad A \bar{B} C \bar{D} + A \bar{B} C D$ | | |
| $\bar{A} \bar{B} C X$ | | $A \bar{B} C X$ | | |

0 0 1 X 1 0 1 X

| | |
|-----------------------|-----------------|
| + | + |
| $\bar{A} \bar{B} C X$ | $A \bar{B} C X$ |
| $X \bar{B} C X$ | |
| $X 0 1 X$ | |

Min terms First level Implicants X stand for \bar{D} Second level Implicants X for (A+A)

Reduce the following equation using the tabular method of minimization.

$$F = m_0 + m_2 + m_3 + m_5 + m_8 + m_{10} + m_{11} + m_{13}$$

| Step 1 | | Step 2 | |
|--------------------|-------------------|--------------------|-------------------|
| Minterm Designator | Binary Designator | Minterm Designator | Binary Designator |
| m ₀ | 0 0 0 0 | 0 | 0 0 0 0 ✓ 0 |
| m ₂ | 0 0 1 0 | 2 | 0 0 1 0 ✓ 1 |
| m ₃ | 0 0 1 1 | 8 | 1 0 0 0 ✓ 1 |
| m ₅ | 0 1 0 1 | 3 | 0 0 1 1 ✓ 2 |
| m ₈ | 1 0 0 0 | 5 | 0 1 0 1 ✓ 2 |
| m ₁₀ | 1 0 1 0 | 10 | 1 0 1 0 ✓ 2 |
| m ₁₁ | 1 0 1 1 | 11 | 1 0 1 1 ✓ 3 |
| m ₁₃ | 1 1 0 1 | 13 | 1 1 0 1 ✓ 3 |

First level implicant

| Minterm Compared | Binary Designator | | | | |
|------------------|-------------------|---|---|---|---|
| 0, 2 | 0 | 0 | x | 0 | ✓ |
| 0, 8 | x | 0 | 0 | 0 | ✓ |
| 2, 3 | 0 | 0 | 1 | x | ✓ |
| 2, 10 | x | 0 | 1 | 0 | ✓ |
| 8, 10 | 1 | 0 | x | 0 | ✓ |
| 3, 11 | x | 0 | 1 | 1 | |
| 5, 13 | x | 1 | 0 | 1 | |
| 10, 11 | 1 | 0 | 1 | x | ✓ |

Step 4

Second Level implicants

| Min term designator | First level implicants compared | Second level implicant |
|---------------------|---------------------------------|------------------------|
| 0, 2 and 8, 10 | 0 0 x 0 and 1 0 x 0 | x 0 x 0 |
| 0, 8 and 2, 10 | x 0 0 0 and x 0 1 0 | x 0 x 0 |
| 2, 3 and 10, 11 | 0 0 1 x and 1 0 1 x | x 0 1 x |

Step 5

Selecting Output Terms

1. First draw a table listing all the mlii terms in columns and the prime Impilcants In rows in a descending order of the squares covered.

| Order of Prime Implicant | Prime Implicants | | Min terms | | | | | | | | | |
|---|------------------|---|-----------|---|---|---|---|---|---|----|----|----|
| | A | B | C | D | 0 | 2 | 3 | 5 | 8 | 10 | 11 | 13 |
| Second level | X | 0 | X | 0 | ✓ | ✓ | | | ✓ | ✓ | | |
| | X | 0 | 1 | X | | ✓ | ✓ | | | ✓ | ✓ | |
| First level (not covered by Second level) | X | 0 | 1 | 1 | | | ✓ | | | | ✓ | |
| | X | 1 | 0 | 1 | | | | ✓ | | | | ✓ |
| | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

2. Against all the implicants put marks Indicating the terms covered. Then find out the essential prime Impilcant.

3. Min term 0 is covered by only prime implicant $X_0 X_0$. So this is an essential prime implicant.
 4. Let us assume $X_0 1 X$ as an essential prime Implicant as It is covering Number of mln terms like m_2, m_3, m_{10}, m_{11}
 5. We can know that $X_0 1 1$ Is not essential prime Implicants because mm terms in 3 and are already covered.
 6. Again $X_{10} 1$ is an essential prime implicant as the min terms m_5 and m_{13} are not covered by any other Implicant.
- So essential prime Implicants ar $X_0 X_0 + X_0 1 X + X_{10} 1$.

Then the reduced expression Is

$$F = \overline{B} \overline{D} + \overline{B} C + \overline{B} \overline{C} D$$

1.2.8 Analysis of logic Schematics

The analysis of a combinational circuit starts with a given logic diagram and ends with a set of Boolean functions, a truth table or a verbal explanation of the circuit operation.

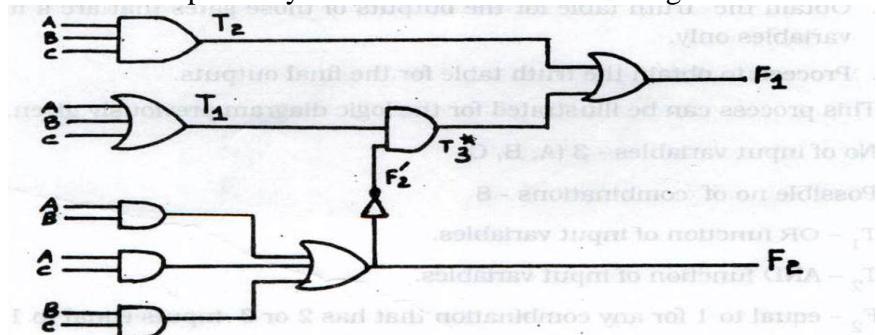
The first step In the analysis is to ascertain whether the given circuit is combinational or not. A combinational circuit has logic gates with no feedback paths.

Once the logic diagram Is verified as a combinational circuit, one can proceed to obtain the output boolean functions and / or the truth tables.

To obtain the output Boolean functions from a logic diagram we have to proceed as follows.

1. Start with arbitrary symbol for Input variable. Then obtain the Boolean functions for each gate using these variables.
2. The same procedure Is repeated until the outputs of the circuit are obtained in terms of input variables (or) previously labeled gates.
3. By repeated substitution and manipulation, obtain the output Boolean function In terms of Input variables only.

Now for example analyses the combinational circuit given below:



Fundamentals Of Digital Computers

This circuit has three binary inputs A,B and C and two binary outputs F_1 and F_2 . T_2 , T_1 and F_2 are Intermediate outputs those are in terms of input variables.

The Boolean functions will be as follows:

$$F_2 = AB + AC + BC$$

$$T_1 = A + B + C$$

$$T_2 = ABC$$

The Boolean function fur output T_3 , F_1 will be as follows.

$$T_3 = F_2' T_1, [\text{where } F_2' = 1 / (AD \div AC+BC)]$$

$$F_1 = T_3 + T_2$$

The Boolean functions fer the final outputs will be as follows;

$$F_2 = AB \div AC+BC$$

$$\begin{aligned} F_1 &= T_2 + T_3 \\ &= ABC \div F \\ &= ABC + (AB-s-AC+BC)' (A+B+C) \\ &= (A' + B') (A' + C') (B' + C') (A + B + C) + A B C \\ &= (A' + WC')(AB' + AC' + BC' + B'C) + ABC \\ &= A' B C' + A' B' C + AW C \div AS C \end{aligned}$$

We can derive the truth table directly from the Boolean functions and try to recognize a familiar operation.

To obtain the truth table directly from the logic diagram without going through the deviations of the Boolean functions we shall proceed as follows.

1. Determine the number of Input variables to the circuit. For n Inputs, form the 2^n possible input combinations of 1's arid 0's by listing the binary number from 0 to 2^n-1 .
2. Label the outputs of selected gates with arbitrary symbols.
3. Obtain the truth table for the outputs of those gates That are a function of the Input variables only.
4. Proceed to obtain the truth table for the final outputs.This process can be illustrated for the logic diagram previously given.

No of Input variables - 3 (A, B, C)

Possible no of combinations - 8

T_1 - OR function of input variables.

T_2 - AND function of Input variables.

F_2 - equal to 1 for any combination that has 2 or 3 inputs equal to 1.

F_2' - Complement of F_2 .

F_1 -equal to 1 If either T_2 or T_3 are both equal to 1

T_3 -equal to 1 when both T_1 and F_2 are equal to 1 and 0 otherwise.

| A | B | C | T_1 | T_2 | F_2 | F_2' | T_3 | F_1 |
|---|---|---|-------|-------|-------|--------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

1.2.9 Synthesis of Combinations functions

The design of combinational function starts from the verbal outline and ends in a logic circuit diagram, or a set of Boolean functions from which the logic diagram can be easily obtained. The procedure involves the following steps.

1. The number of available Input variables and required output variables are determined from the given problem.
2. Symbols are assigned to input and output variables.
3. Then the truth table is derived from the relationships between inputs and outputs.
4. The simplified Boolean function for each output is obtained.
5. Finally logic diagram is drawn.

For example let us design a combinational circuit which consists of three Inputs and two outputs. It should give sum and carry outputs.

Step 1:

The number of Input Variables - S

The number of Output variables - 2

Step 2:

Symbols assigned to Inputs - x, y, z.

Symbols assigned to outputs - s, c.

Step 3:

This combinational circuit should give $s = 0, c = 0$ output when all Inputs are equal to 0. The S output is equal to 1 for odd numbers of 1's at the input. The C output will be equal to 1 if more than one input is equal to 1.

1.3) Revision Points

Boolean algebra

Boolean algebra provides a mathematical basis, which is almost essential for a proper understanding of digital circuits

Karnaugh Map

Karnaugh Map technique, which is faster as well as better system for reducing Boolean expressions

Two level logic

Sum of products and product of sums reduction process produces an input, which feeds an input gate as well as an output gate. Therefore it is called two level logic.

Analysis of logic Schematics

The design of combinational function starts from the verbal outline and ends In a logic circuit diagram, or a set of Boolean functions from which the logic diagram can be easily obtained.

1.4) Intext Questions

- 1.Explain in detail about Boolean Algebra
- 2.What are Boolean expressions and logic diagrams? Discuss in detail.
- 3.What is Combination Logic? Explain in detail with one example.
- 4.Give detail about Tabular minimization

1.5) Summary

- Boolean Algebra has and Extensive wide spread application in the design of digital computers.
- Logic gates are used to perform the logical operations like addition, subtraction, division and multiplication.
- Minimization of Boolean expressions leads to reduction of Boolean expression.
- The analysis of a combinational circuit starts with a given logic diagram and ends with a Set of Boolean functions, a truth table or a verbal explanation of the circuit operation.

1.6) Terminal Exercises

- 1.Explain about Boolean expression.
- 2.What is a logic diagram?
- 3.Explain about Karnaugh map.
- 4.Mention the types involved in Karnaugh map.
- 5.What is the use of Karnaugh map?
- 6.Define about Two level logic.

1.7) Supplementary Materials

The design and analysis of spatial data structures – Samet

1.8) Assignments

Prepare assignment about Karnaugh Map.

1.9) Reference Books

Bhujade, ‘Digital Computer Design Principles’, M.R.Pitamber publishing company, 1989.

1.10) Learning Activities

An individual or group of peoples goes to library for future evaluation of this unit.

1.11) Keywords

- **Boolean Algebra**
- **Combination Logic**
- **Tabular minimization**
- **Truth tables**
- **Logic diagrams**

UNIT- II

2.0) Introduction

A digital computer is having the following fundamental units.

1. Input Devices.
2. Output Devices.
3. Central Processing Unit
4. Memory

The central processing unit consists of Arithmetic Logic unit and Control unit. The arithmetic logic unit of most computers is capable of performing addition, subtraction, multiplication and division as well as some logical operations. The control unit of a computer sequences the operation of a computer and controls the actions of all other units. Now let us discuss the arithmetic logic unit and control unit in detail.

2.1) Objective

At the end of this unit student will get knowledge about the following topics:

- Arithmetic-Logic Unit
- Control unit
- Organization of control registers
- Instruction set
- Micro programming applications

2.2) Content

2.2.1 Arithmetic-Logic Unit (ALU)

Construction of the ALU

The Information handled in a computer is generally divided into 'words'. Each consists of a fixed number of bits. If the word length of the computer is 32 bits, the ALU should be capable of adding, subtracting etc., words of 32 bits In length.

An accumulator Is a basic storage register of the arithmetic element which Is used to store the results temporarily.

If we want to add two binary words, the addend Is located in memory and the addend is transferred to the accumulator. The control unit directs the ALU to add these two. Then

the result is stored in the accumulator. Like this all the operations are carried out. In the following section let us discuss the arithmetic circuits and various operations done by the ALU.

Integer Representation

In digital computer the numbers are stored in flip-flops and an integer representation system is used to represent the numbers. Each flip-flop can store one bit of information. So to store binary number 1100 or decimal 12. four flip-flops are required and one more bit is used to represent the sign of the number indicating whether the number is positive or negative. This bit is called sign bit. This system is called signed-Integer binary system or signed -magnitude binary integer system.

In this signed integer binary system, if a register contains eight flip-flops, 7 bits are used to represent the magnitude and a single bit represents sign bit. So 00011111 (+ 15) to 10011111 (-15).

Binary Half adder

The function of the half adder is to add two binary bits and produce a sum and a carry according to the binary addition rule.

The following table shows the rules for binary addition.

| Augend A | Addend B | Sum Σ | Carry C |
|-------------|-------------|-----------------|------------|
| 0 | + | 0 | 0 |
| 0 | + | 1 | 1 |
| 1 | + | 0 | 1 |
| 1 | + | 1 | 0 |

For example let us take the following binary addition

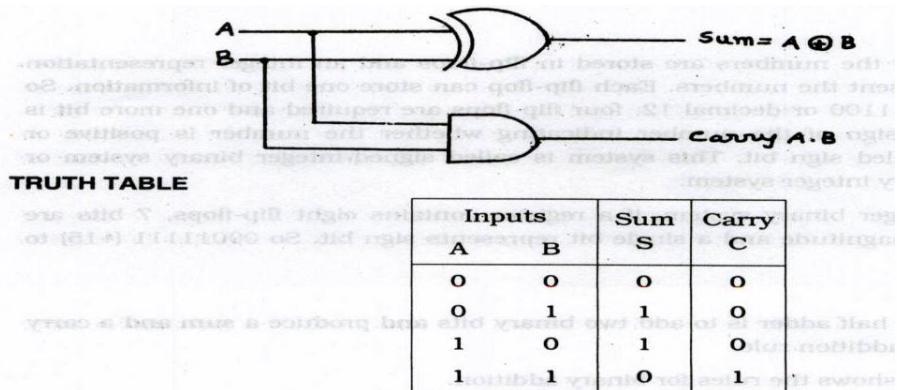
$$\begin{array}{r}
 \begin{array}{ccccccc}
 & & & & \text{Start from LSB [least} \\
 & & & & & & \text{significant bit]} \\
 \text{A} & 1 & 1 & 1 & 0 & 0 & \\
 \text{B} & 1 & 0 & 1 & 0 & & \\
 \hline
 & 0 & 1 & 1 & 0 & \text{Sum} & \\
 & 1 & 0 & 0 & 0 & \text{carry} &
 \end{array}
 \end{array}$$

If we apply A and B inputs to the XOR gate, the output will satisfy the sum column as well and an AND gate will satisfy the carry column. So we can conclude that if the same two inputs are applied to XOR and AND gate, the output of the XOR gate will represent the sum and the AND gate will represent the carry.

Therefore Boolean equation for

$$\begin{aligned} \text{sum} &= A \oplus B \\ &= A \bar{B} + \bar{A} B \\ \text{carry} &= A \cdot B \end{aligned}$$

Based on the above consideration let us draw the circuit diagram of the half adder as



Full adder

A full adder can add three binary bits that is for adding two digit plus a carry digit from a previous column.

A---> ---> Sum
 Full
 B---> adder ----> Carry
 C--->

TRUTH TABLE

| Inputs | | | Sum | Carry |
|--------|---|---|-----|-------|
| A | B | C | S | C |
| 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 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Now let us design a full adder circuit from the truth table by using Karnaugh map

Sum Map

| BC | | 00 | 01 | 11 | 10 |
|----|---|----|----|----|----|
| A | | 0 | 1 | 2 | 3 |
| 0 | 0 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 1 | 0 | |
| | 4 | 5 | 7 | 6 | |

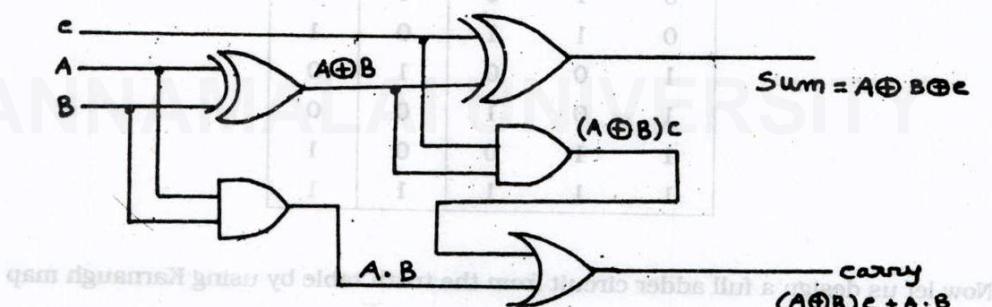
Sum = $\bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + ABC$
 $= \bar{A} [\bar{B}C + B\bar{C}] + A[\bar{B}\bar{C} + BC]$
 $= \bar{A} [B \oplus C] + A [B \odot C]$
 $= A \oplus B \oplus C$

Carry Map

| AB | | 00 | 01 | 11 | 10 |
|----|---|----|----|----|----|
| C | | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 3 | 2 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| | 4 | 5 | 7 | 6 | |

$$\begin{aligned} \text{Carry} &= AB + \bar{A}BC + A\bar{B}C \\ &= AB + (A \oplus B)C \end{aligned}$$

Circuit Diagram



From this circuit diagram we can know that a full adder can be constructed using two half adders and an OR gate. Moreover a full adder can be constructed easily as follows, From the truth table let us write the Boolean equation for sum and carry.

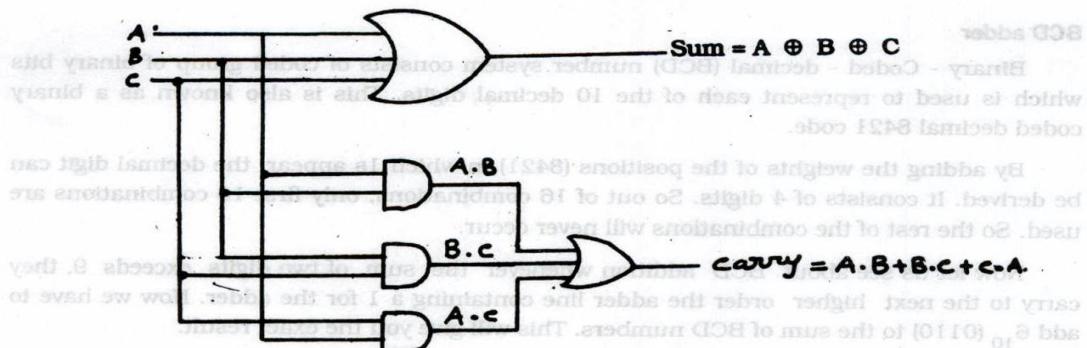
$$\begin{aligned} \text{Sum} &= \bar{A} \bar{B} C + \bar{A} B \bar{C} + A \bar{B} \bar{C} + A B C \\ &= A \oplus B \oplus C \end{aligned}$$

The carry outputs are obtained by feeding inputs A and B, B and C, C and A, three 2 inputs AND gates and connecting their outputs to a 3 Input OR gate

$$\text{Carry} = AB + BC + AC$$

| | AB | 00 | 01 | 11 | 10 |
|---|----|----|----|----|----|
| C | 0 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 1 | 1 | 1 |

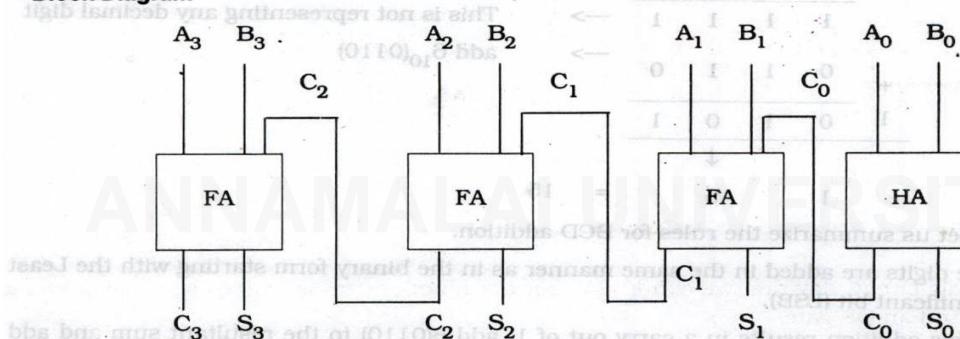
AB
AC
BC



Parallel Binary Adder

A 4 bit parallel adder can add two 4 bit numbers by combining three full adders and a half adder.

Block Diagram



BASIC PRINCIPLE

| C ₄ | C ₃ | C ₂ | C ₁ | C ₀ | |
|----------------|----------------|----------------|----------------|----------------|-----|
| A ₃ | A ₂ | A ₁ | A ₀ | | |
| B ₃ | B ₂ | B ₁ | B ₀ | | |
| S ₃ | S ₂ | S ₁ | S ₀ | | Sum |

The carry output of each adder goes to the carry Input of the next adder to the left. There is no carry output from the L.SB. The addition of these two digits (A_0, B_0) produces a sum S_0 and a carry out of C_0 . This C_0 will be added with input A_1, B_1 , which produces output S_1, C_1 . The third adder adds A_2, B_2 and carry output C_1 from the second adder then produces outputs S_2, C_2 . The last adder will add A_3, B_3 and the carry out of the previous adder and gives S_3 and C_3 as the output. Similarly we can design 8 bit and 16 bit adders.

BCD adder

Binary - Coded - decimal (BCD) number system consists of coded group of binary bits which is used to represent each of the 10 decimal digits. This is also known as a binary coded decimal 8421 code.

By adding the weights of the positions (8421) In which Is appear, the decimal digit can be derived. It consists of 4 digits. So out of 16 combinations, only first 10 combinations are used. So the rest of the combinations will never occur. Now let us see about BCD addition whenever the sum of two digits exceeds 9, they carry to the next higher order the adder line containing a 1 for the adder. Now we have to add 6₁₀ (0110) to the sum of BCD numbers. This will give you the exact result.

Examples

| (8) | (4) | (2) | (1) | \rightarrow | Weights of the position |
|--------------|-------|-------|-------|---------------|--|
| $8 + 7 = 15$ | 1 | 0 | 0 | \rightarrow | Augend |
| | 0 | 1 | 1 | \rightarrow | Addend |
| | 1 | 1 | 1 | \rightarrow | This is not representing any decimal digit |
| $+$ | 0 | 1 | 1 | \rightarrow | add 6_{10} (0110) |
| | 1 | 0 | 1 | \downarrow | |
| | | | | = | 15 |

Now let us summarize the i-tiles for BCD addition.

1. The digits are added in the same manner as in the binary form starting with the Least Significant bit (LSB).
2. If this addition results In a carry out of 1. add 6(0110) to the resultant sum and add carry to the next most significant bit (MSB). . r

Circuit diagram

Let us Verify the circuit operations as follows.

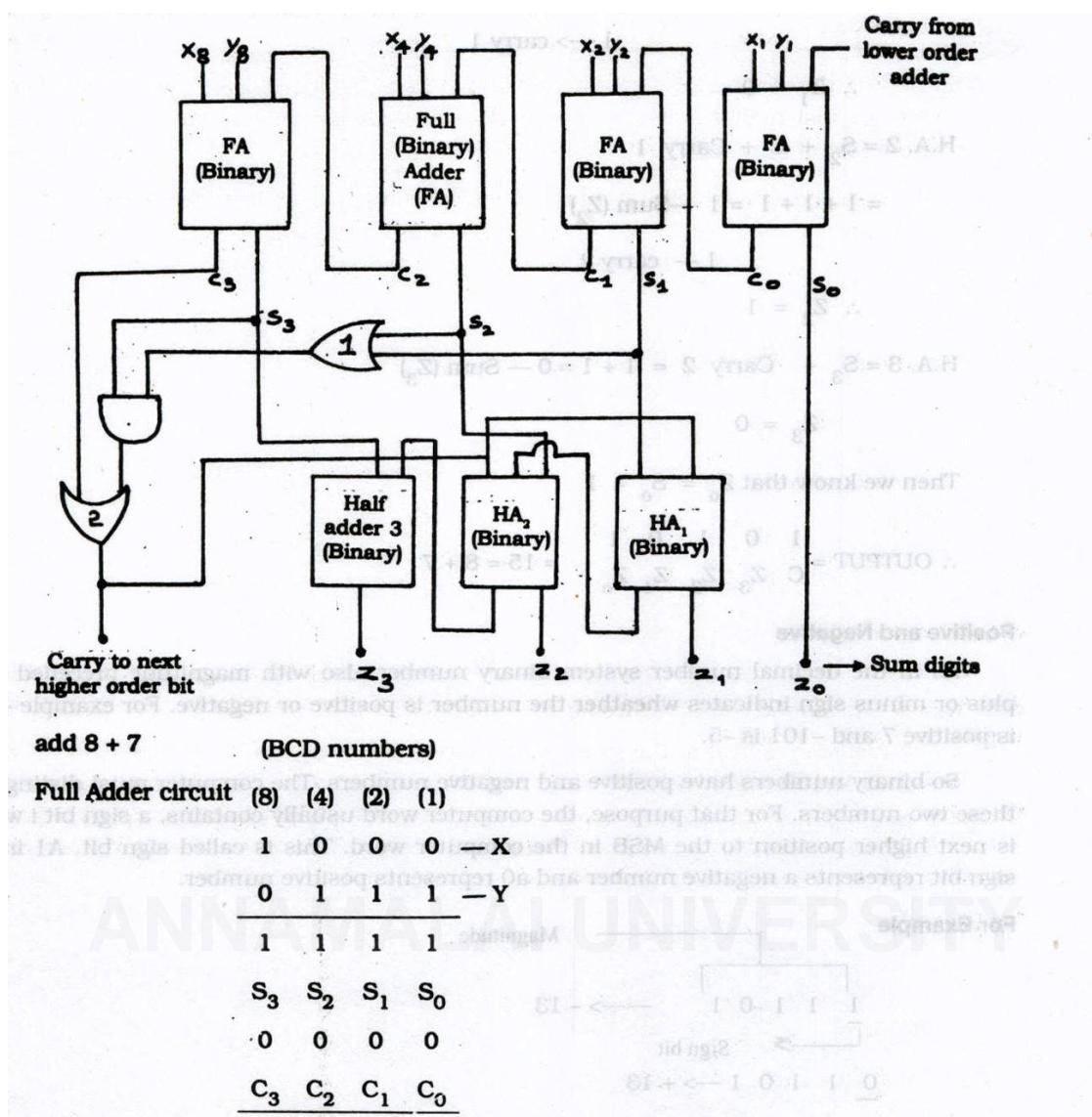
Gate Network

$S_1 + S_2 = 1 + 1 = 1$ (output of OR gate)

S_3 . output of OR gate $1 = 1 \cdot 1 =$ (output of AND gate)

$C_3 +$ output of AND gate $= 0 + 1 = 1$ (output of OR gate 2)

$C = 1$



Halt adder Circuits

$$I - LA \cdot 1 = 3 + C = I + I = O \rightarrow \text{Sum}(Z1)$$

$$\begin{aligned}
 & 1 \rightarrow \text{carry } 1 \\
 z_1 &= 0 \\
 \text{H.A.2} &= S_2 + C + \text{Carry } 1 \\
 \\
 & = 1+1+1 = 1 - \text{Sum}(Z_2)
 \end{aligned}$$

$$\begin{aligned}
 & 1 - \text{carry } 2 \\
 Z_2 &= 1 \\
 \text{H.A.3} &= 53 + \text{Carry}_2 = 1+1=0 - \text{Sum}(Z_3) \\
 za &= 0
 \end{aligned}$$

Then we know that $Z = S = 1$

$0 \ 0$

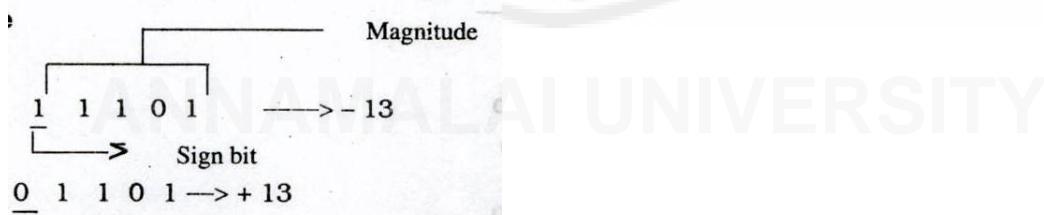
$$\begin{array}{r}
 10101 \\
 \therefore O'U'I'l, f'l' = \sim \sim \sim \sim \\
 3 \ 2 \ lo
 \end{array}
 \quad = 15 = 8 + 7$$

Positive and Negative

As in the decimal number system binary number also with magnitude preceded by a plus or minus sign Indicates whether the number is positive or negative. For example + iii is positive 7 and -101 is -5.

So binary numbers have positive and negative numbers. The computer must distinguish these two numbers. For that purpose, the computer word usually contains, a sign bit i which is next higher position to the MSB in the computer word. This is called sign bit. A 1 in the sign bit represents a negative number and a 0 represents positive number.

For Example



This is called sign-magnitude representation of binary numbers. In addition to it the negative may be represented as 1's complement and 2's complement. 1's Complement The 'Radix minus I' complement is commonly referred to as the 1's complement.

For example let us find out the 1's complement of the binary number 1010 as follows

$$\text{Radix-I} = 1 \ 1 \ 1 \ 1$$

Binary number = 1 0 1 0
Subtract = 0 1 0 1 -> 1's complement

Since the radix is 2 for binary numbers subtracting 1 from 2 gives 1. From this example we can find out that 1's complement of a binary number can be easily obtained by complementing each bit that is by changing all 0's to 1's and all 1's to 0's.

Binary Number 1 0 1 0
0 1 0. 1 -> 1's complement
We can easily say that both the results are same.

2's Complement

The 2's complement of the binary numbers can be easily obtained by adding 1 to the LSB of the 1's complement.

For example find out the 2's complement of 1 0 0 1
Given Binary number = 1 0 0 1
1's Complement = 0 1 1 0 +
AddltoLSB
2's complement = 0 1 1 1

The representation of BCD numbers is the same as that of binary numbers, but It will have 9's or 10's complement instead of 1's and 2's complement.

A shift operation moves the digits to a new position. That means a shift operation moves each bit to the left or right by some specified number of digits. If it is moved to the left, it is called shift-left operation. If it is moved to the right, It is called shift-right operation.

For example consider binary number

000110

After shift left operation by 1 digit

001100

After shift right operation by 1 digit

000011

Basic Operations

A digital computer consists of a number of registers which are used to store binary digits. So It is possible to perform certain operations on the bits stored In the registers. The

following are some of the basic operations which can be performed by the arithmetic circuits present In the arithmetic- logic unit.

1. The register may be reset to all 0's
2. The contents of a register may be 1's complemented or 2's complemented.
3. It may be shifted left or right
4. The content of the register can be incremented or decremented.
5. Adding to or subtracting from the contents of one register the contents of another register.
6. Comparison of magnitudes of two words are also possible.
7. Multiplications and division are also possible.

Logic Operations

In addition to the arithmetic operation mentioned above, many logical operations also are performed by ALUs. These logical operations are AND operation OR operations. complementation and the exclusive OR operations. Already we are familiar with these operations. These operations will be performed between registers, and then the results will be stored in one of the register.

Arithmetic And Logic Unit

Arithmetic and logic units ALUs) are available in single Integrated Circuited package. This single chip can perform all the arithmetic and logic operations so far mentioned. This device accepts two to four input words A.3. A2, A1. A0 designated A and B3, 82.81. B designated as B. The output is represented by F which is also A4 bit word F~, F2, F1, F0. This device can perform a number of arithmetic and logic functions. These operations are selected by the four selection lines S3~ ~2' ~1* S~)~ The mode input Is used to distinguish both logical and arithmetic operations.

Arithmetic Logic Circuits

Fundamentals Of Digital Computers

| Select Input | | | | Active-High Data | | |
|----------------|----------------|----------------|----------------|--------------------------------|----------------------------------|--------------------------------------|
| | | | | Arithmetic and logic functions | | |
| | | | | M = 0 | $\bar{C}_N = 0$ (With carry) | |
| | | | | F | F | F |
| S ₃ | S ₂ | S ₁ | S ₀ | | | |
| 0 | 0 | 0 | 0 | $F = \bar{A}$ | $F = A$ | $F = A$ Plus 1 |
| 0 | 0 | 0 | 1 | $F = \bar{A} + B$ | $F = A + B$ | $F = (A + B)$ Plus 1 |
| 0 | 0 | 1 | 0 | $F = \bar{A} \cdot B$ | $F = A + \bar{B}$ | $F = (A + \bar{B})$ Plus 1 |
| 0 | 0 | 1 | 1 | $F = 0$ | $F = \text{minus 1 (2's compl)}$ | $F = 0$ |
| 0 | 1 | 0 | 0 | $F = \bar{A} \cdot B$ | $F = A$ Plus $\bar{A}B$ | $F = A$ Plus $A\bar{B}$ plus 1 |
| 0 | 1 | 0 | 1 | $F = \bar{B}$ | $F = (A + B)$ Plus $\bar{A}B$ | $F = (A + B)$ Plus $A\bar{B}$ Plus 1 |
| 0 | 1 | 1 | 0 | $F = A \oplus B$ | $F = A$ Minus B Minus 1 | $F = A$ Minus B |
| 0 | 1 | 1 | 1 | $F = A \cdot \bar{B}$ | $F = A\bar{B}$ Minus 1 | $F = A \cdot \bar{B}$ |
| 1 | 0 | 0 | 0 | $F = \bar{A} + B$ | $F = A + AB$ | $F = A$ Plus AB Plus 1 |
| 1 | 0 | 0 | 1 | $F = A \oplus B$ | $F = A$ Plus B | $F = A$ Plus B Plus 1 |

$$F=B$$

$$F=AD$$

$$F=1$$

$$F=(A + B) \text{ Plus AD}$$

$$F=AS \text{ Minus 1}$$

$$F=A \text{ Plus A}$$

$$F=(A + B) \text{ Plus A}$$

$$F=(A + B) \text{ Plus A}$$

$$F=A \text{ Minus I}$$

$$\begin{array}{cccc} 1 & 0 & 1 & 0 \end{array}$$

$$\begin{array}{cccc} 1 & 0 & 1 & 1 \end{array}$$

$$\begin{array}{cccc} 1 & 1 & 0 & 0 \end{array}$$

$$F=A \div B$$

$$F=A+B$$

$$F=A$$

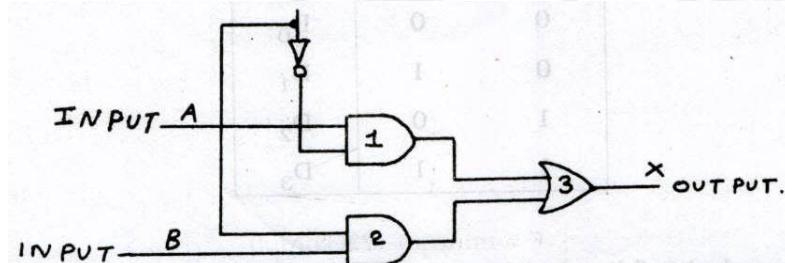
A: represents 4-bit long word applied at the A Inputs
 B represents 4-bit long word applied at the S Inputs
 F: represent 4-bit long output word on F3, F2, F1 and F0 ouputs

+ represents logical OR operation
 Plus : Represents addition
 CN : represents the logic level on the carry Input. It is active low.

Multiplexers

A multiplexers is a combinational logic circuit which can select any one of a number of inputs and route It to a single output. We shall illustrate the concept of digital Implementations of multiplexers with twoInputs and a single output.

In this diagram, A and B are the two Inputs, X Is the output and C Is the control input. When Input B Is to be routed to the output and C is the control input Is made high C=1, which enables AND gate 2 and its output passes through the OR gatea to the output. Similarly C=0 to route Input A to the output.



4 Line to 1 Line multiplexer

In the following diagram the Inputs are marked as D0, D1, D2, and D3. The output Is Y and the control signals are A and B

$$F = (A + B) \text{Plus AS Plus 1}$$

$$F = AD$$

$$F = A \text{Plus} A \text{Plus 1}$$

$$F = (A+B) \text{Plus} A \text{Plus 1}$$

$$F = (A+B) \text{Plus} A \text{Plus 1}$$

$$F = A$$

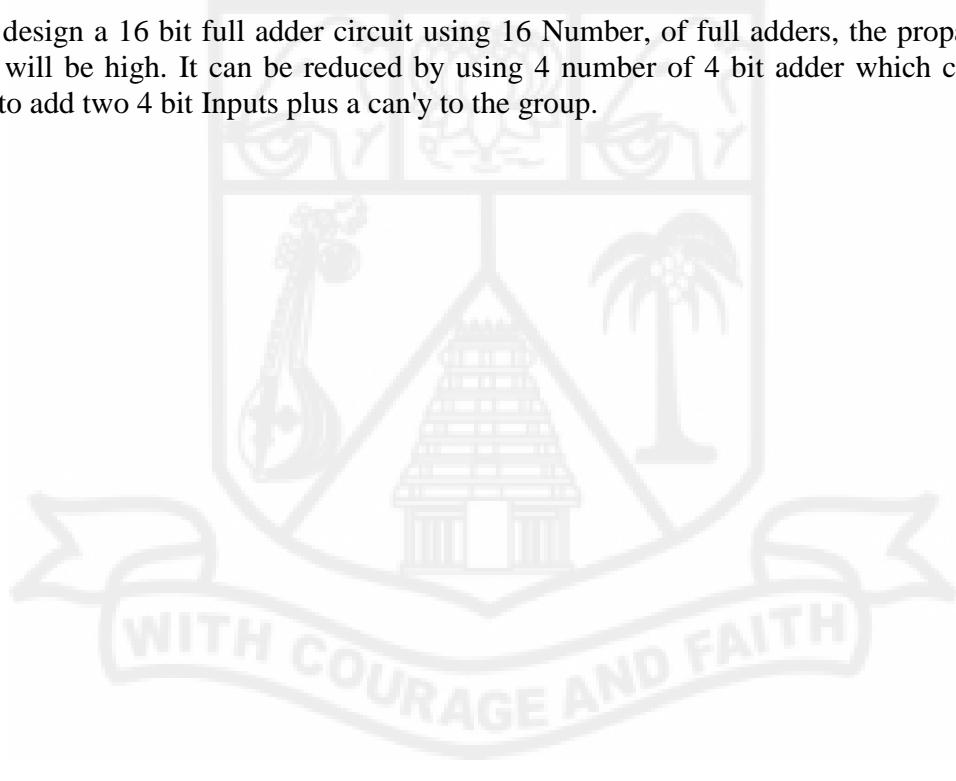
Like this we can design 8 line to 1 line multiplexer and 16 lines to 1 line multiplexer etc.

High Speed Arithmetic

Since additions and subtractions are often performed in computers, it is desirable to perform them quickly. Moreover it will speed up multiplication and division as these involve a number of addition or subtraction.

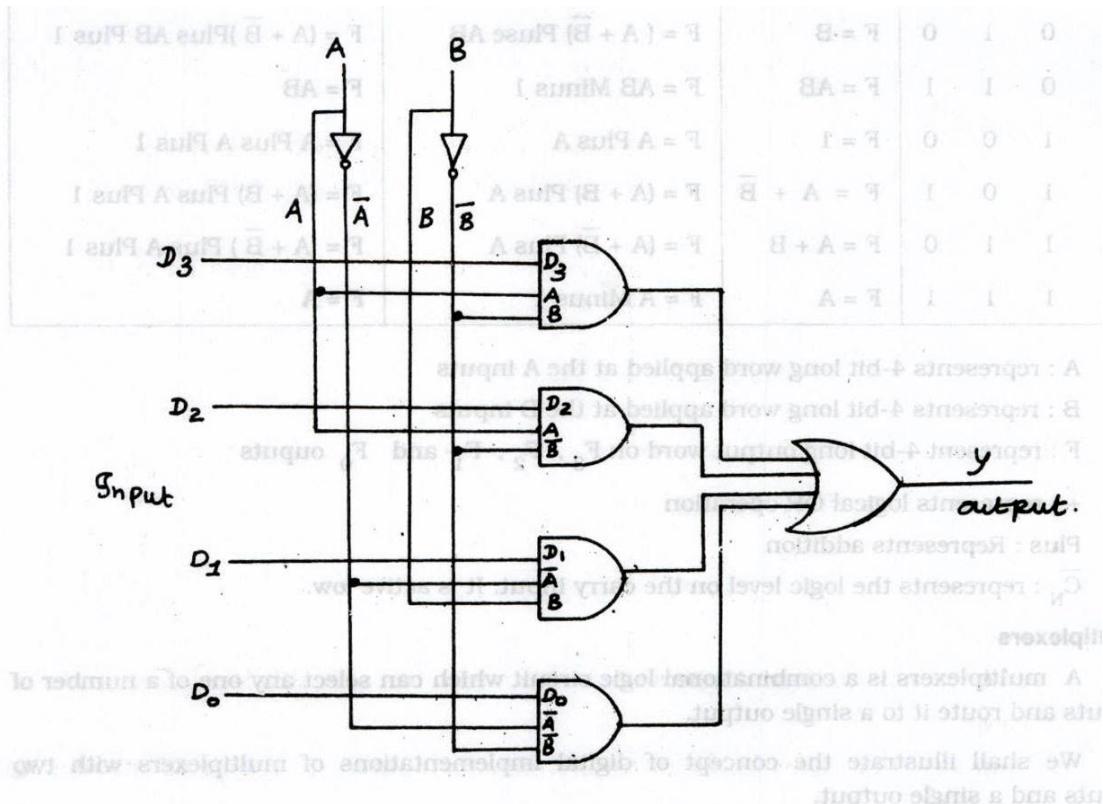
For example, consider a set of circuit each gate in a network delays a signal by some time period. Thus if a set of new inputs is placed on the inputs to the adder configuration, It will take some time to reach the outputs. This is called carry propagation delay for high speed arithmetic. The carry propagation delay must be reduced. This can be achieved by reducing the complexity of the gating network used.

If we design a 16 bit full adder circuit using 16 Number, of full adders, the propagation delay will be high. It can be reduced by using 4 number of 4 bit adder which contains gates to add two 4 bit Inputs plus a can'y to the group.



ANAMALAI UNIVERSITY

Fundamentals Of Digital Computers



Truth Table

| Control Signals | | Output |
|-----------------|---|----------------|
| A | B | Y |
| 0 | 0 | D ₀ |
| 0 | 1 | D ₁ |
| 1 | 0 | D ₂ |
| 1 | 1 | D ₃ |

In large computer and in signal processing computer, there is a need for high speed multiplication. For this arrays of gates are used which multiply several binary digits at the same time. It is called Parallel multiplier. If a₁, a₀ and b₁, b₀ are inputs given to the multiplier, then the product in binary integer form will be P₃ p₂ p₁ p₀*

$$P_0 = a_0, b_0$$

$$P_1 = a_0 b_1 \$ a_1 b_0$$

$$P2 = a1\ b1\$ a,0\ b1\ a1\ I\sim$$

$$P3 = a1\ b1\ a0\ a1\ b1\ a0\ b0$$

For large numbers these multipliers can be grouped with some full adders.

2.2.2 Control Unit

The control circuitry provides the necessary timing and control signals to all operations. Now let us discuss it in detail.

Construction of an Instruction Word

An Instruction word consists of a set of binary digits to represent an Instruction to the computer. It is used to determine the sequence of operations which the computer performs.

An Instruction word in a digital computer consists of a number of sections. Basically each instruction word contains two sections namely operation code (Opcode) and address part. The opcode defines the operation to be performed like addition, subtraction etc and the address part contains the memory location of the number on which the operation is to be performed.

Instruction word format

Opcode address part

The possible no of basic Instruction types are determined by the no of bits in the opcode part. If it Is 3 bit opcode, 2³ or 8 basic instruction types are possible. Normally opcodes are represented by Assembly language or mnemonics

Example

11 10 9 8 7 6 5 4 3 2 1 0

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

OPCODE ADDRESS PART

Here OPcode 001 represents the instruction TAD. It tells the computer to add the number located in memory at the address given in the address part (here address is 0 0 0 0 1 1 1. location 7) of the instruction to the number currently In the accumulator and to store the sum In the accumulator.

Instruction cycle and execution cycle

The execution of a program in a digital computer consists of an instruction cycle and an execution cycle.

During the instruction cycle, an instruction word is obtained from the memory and the address of the operand is given to the memory. During the execution cycle the operand is obtained and the operation given by the opcode is performed upon this. During each instruction cycle, the Instruction word is transferred from memory to memory buffer register then It is Interpreted and the address of the operand is given to the memory address register. During execution cycle an operand is obtained from the memory depending on the instruction word which will be Interpreted.

Organization of control register

Every digital computer consists Of several registers. These are described as follows:

1. Instruction Counter: (Program Counter)

This register is used to sequence the execution of instructions. It is used to point to the memory address from which the next instruction is to be fetched. The program counter Is Incremented by one during the performance of each Instruction and the contents are transferred to memory address register at the beginning. The length of this counter the same as the address part of the instruction word.

2. Op Code register

When an instruction word is read from the memory, the op-code part is transferred to the opcode register In order to identify the operation to be performed. The length of the register Is same as the length of the opcode part.

3. Memory address register

This register contains the location of the word, which is to be read from or written into the memory

4. R. flip flop

It tells the memory to read a word when it is turned on.

5. W. flip flop

It tells the memory to write the word from memory buffer register at the location given by the memory address register when it is turned on.

6. I flip flop

When this is on, the computer is In an instruction cycle.

7. E. flip flop

When this is on, the computer is in an execution cycle.

Instruction formats

Computers may have instructions of several different lengths containing various number of addresses. They are

1. One address instructions.
2. Two address instructions
3. Three address Instructions
4. Zero address instructions.

Already we have discussed about one address Instructions. Now let us discuss about all other instructions.

Two address instructions

It consists of three sections. The first consisting of the OP code and the second and third sections containing the address of the memory locations or processor register.

| OPCODE | ADDRESS OF OPERAND (A) | ADDRESS OF OPERAND (B) |
|--------|------------------------|------------------------|
|--------|------------------------|------------------------|

Example

(1) ADD R1,B [It means $\leftarrow R_1 + M(B)$]

Add - opcode

- Specifies register address to obtain augend

B -> Specifies memory address to obtain addend

(2) MOV R1, A [it means $c \leftarrow M(A)$]

MOV - Opcode

A -> specifies memory address

-> specifies register address

Three Address Instructions

It consists of opcode part and three address part. For example consider the following operation.

ADD R1, A, B

It means that add the operand at memory address A and B and store the result at processor register R1. The instruction format will be as follows;

| | | | |
|--------|----------------|---------|---------|
| ADD | R ₁ | A | B |
| OPCODE | ADDRESS | ADDRESS | ADDRESS |

PART PART PART
1 2 3

Zero Address instructions

These instructions do not specify any location in memory for an operand. It means that it does not use an address field for the instructions, but a stack is used to provide operands. A stack is a set of memory locations into which operands can be stored. Placing and removing an operand on the stack is called pushing and popping respectively. The stack pointer is a register which contains the address of the top operand in the stack. The stack pointer is incremented or decremented when an operand is pushed or popped.

Instruction Set

Computers provide a set of instructions to carry out various operations. The instruction set of different computers differ from each other. But there is a set of basic operations that must be included in most of the computer's instruction set. In this section we are discussing about the basic set of operations available in a computer.

Most computer instructions can be classified into three categories.

1. Data transfer instructions
2. Data manipulation instructions.
3. Program control instructions.

1. Data Transfer Instructions

Data transfer instructions move data from one place in the computer to another without changing the data content. The following table gives a list of eight data transfer instructions used in many computer.

| Name | Mnemonic |
|----------|----------|
| Load | LD |
| Store | ST |
| Move | MOV |
| Exchange | XCH |
| Input | IN |
| Output | OUT |
| Push | PUSH |
| POP | POP |

2. Data Manipulation Instructions

Data manipulation instructions perform operations on data. There are three types of data manipulations instructions in a typical computer.

1. Arithmetic instructions.
2. Logical and bit manipulation instructions
3. Shift instructions.

Fundamentals Of Digital Computers

Typical Arithmetic Instructions

| Name | Mnemonic |
|----------------------|----------|
| Increment | INC |
| Decrement | DEC |
| Add | ADD |
| Subtract | SUB |
| Multiply | MUL |
| Divide | DIV |
| Add with carry | ADDC |
| Subtract with borrow | SUBB |
| 2'scomplement | NEG |

Typical logical and Bit manipulation Instructions

| | |
|-------------------|------|
| Clear | CLR |
| Complement | COM |
| AND | AND |
| OR | OR |
| Exclusive - OR | XOR |
| Clear Carry | CLRC |
| Set Carry | SETC |
| Complement Carry | COMC |
| Enable Interrupt | EI |
| Disable Interrupt | DI |

Typical logical and Bit manipulation Instructions

| Name | Mnemonic |
|----------------------------|----------|
| Logical shift right | SHR |
| Logical shift left | SHL |
| Arithmetical shift right | SHRA |
| Arithmetic shift left | SHLA |
| Rotate right | ROR |
| Rotate left | ROL |
| Rotate right through carry | RORC |
| Rotate left through carry | ROLC |

Fundamentals Of Digital Computers

3. Program control instructions

Program control instructions specify conditions for altering the sequence of execution of a program by altering the content of the program counter. These instructions are important in digital computer to control the flow of program execution and providing a capability for branching to different program segments.

| Name | Mnemonic |
|---------|----------|
| Branch | BR |
| Jump | JMP |
| Skip | S1W |
| Call | CALL |
| Return | RET |
| Compare | CMP |
| Test | `1ST |

Branch and jump instructions may be conditional or unconditional. An unconditional branch instruction causes a branch to the specified address without any condition. The conditional branch instruction specifies a condition. While encountering the condition, the program sequence is changed to the branch address otherwise it will follow the normal sequence.

The condition in the conditional branch instruction is depending on status bits like C, S, Z and V which is also called as flag bits.

- C --> Carry Bit
- S --> Sign Bit
- Z --> Zero bit
- V --> Overflow bit

Conditional branch instructions

| Name | Mnemonic |
|-----------------------|----------|
| Branch if Zero | BZ |
| Branch if not Zero | BNZ |
| Branch if Carry | BC |
| Branch if no Carry | BNC |
| Branch if Plus | BP |
| Branch if Minus | BM |
| Branch if Overflow | BV |
| Branch if no Overflow | BNV |

Unsigned Compare Conditions (A-B)

| | |
|---------------------------|-------|
| Branch if Higher | RHI |
| Branch if Higher or equal | B1-IE |
| Branch if Lower | BLO |
| Branch if Lower or equal | BLOE |
| Branch if Equal | BE |
| Branch if not Equal | BNE |

Signed Compare Conditions (A-B)

| | |
|----------------------------|-----|
| Branch if Greater than | BGT |
| Branch if Greater or equal | BGE |
| Branch if Less than | BLT |
| Branch if Less or equal | BLE |
| Branch if Equal | BE |
| Branch if Not equal | BNE |

NOTE:

1. The relative magnitude of two unsigned number is determined from status bit C and Z but for signed numbers It is from S, V and Z. So both are having different instructions.
2. Interrupt refers to the program control from a currently running program to another program as a result of an external or internal request.

Controlling Arithmetic Operations

Let us now consider the problem of directing the control register to perform arithmetic operations. For that purpose control signals must be distributed in an orderly manner. A time base will indicate where we are in the sequence of operations to be performed is required. Each memory cycle is broken into four equal time periods, namely T0, T1, T2 and T3.

There is a clock signal input, and each memory cycle contains four clock pulses. Timing signal distributor is generating timing signal. The circuit has four output lines designated as T0, T1, T2 and T3. When the computer is in time period, the output line T0 will carry a 1 signal and T1, T2 and T3 will be 0s: at time T1 only output line T1 will have a 1 signal, rest will be 0, etc.

In an instruction cycle

- I. When it is T0. READ Flip flop will be turned on and directing the memory to read the instruction word located at the address in the memory register then places this word the memory buffer register.
2. At time T1, OPcode part of the instruction word is transferred to the OP-Code register.

3. Next step is always dependent on the Op-Code register. Then a decoder is connected with that register. It will produce a set of signal corresponding to the operation.

I. Signal line 0 0 0 0 0 will carry 1 signal for ADD operation.

2. Signal line 0 0 0 0 1 will carry I signal for SUB operation.

3. Signal line 0 0 0 1 O_w!!! be a I for CIA (ie clear and ADD operations).

4. Signal line 0 0 0 1 1 will be a I signal for STORE Operations.

All the combinations can be used by adding more instructions.

These lines and the timing signal distributor lines and the I and L flip flop lines are combined to give all the control signals needed to run the computer.

Typical Sequence of Operations

Let us analyze the sequence of operations during ADD instruction and a STORE instruction. During ADD instruction

INSTRUCTION CYCLE

I. At first, I flip flop is turned on to initiate the instruction cycle and to output line also turned on. Setting READ flip flop to the I state. So the MAR is assumed to have the address of the instruction will be read into MBR.

2. At time T1 the OP-Code section of the instruction will be transferred into the OP-Code register. Now computer will decode the OP-Code to determine the type of instruction.

3. By time T2 the instruction counter (IC) is incremented by 1 to transfer the address of the next instruction to IC.

4.. At time T3 MBA is transferred into MAR, thus transferring the address part of the instruction word into MAR. Also the instruction cycle flip flop is cleared by the RESET I signed and E flip flop is turned on by the SET E Signal to change the computer from instruction cycle to execution cycle.

EXECUTION CYCLE

1. At time E and T0, during an ADD instruction. R flip flop is set on. To read the word at the address in the address part of the instruction word to be executed.

2. At time E and T1 we transfer the contents of the MBR into the B register. Now the word will be available in the B register for the addition. Then the READ flip flop will be cleared.

3. At time E and T3 address of the instruction counter is transferred from instruction counter into the MAR. Then clearing E flip flop and setting I flip flop to change the computer from execution cycle to an instruction cycle.

During STORE instruction

Instruction Cycle

Instruction Cycle operations are same as that of the ADD instruction.

Execution Cycle

- I. At time E and T0, WRITE flip flop will be turned on to write into the memory.
Transfers word to be read into memory from accumulator into the MBR.
2. At time E and T1, WRITE flip flop will be cleared.
3. At time E and T2 contents of MBR are written into memory.
4. At time E and T3 the content of the instruction counter is transferred into the MAR and giving the location of the next instruction word to the memory. The I flip flop is turned on and the execution cycle flip flop is turned off.

Register transfer language

The way of writing register operations is called register transfer language. Now let us see some register transfer languages.

(a) transfer between register is register transfer language

It means that transfer the contents of register A Into register B.

(b) 'Reset 1' flip flop' is register transfer language $0 \rightarrow D$

(c) $A \div B \rightarrow A$

This regst~r transfer language says that add the number in A and B and place the sum mA.

(d) $A + 1 \rightarrow A$

It means that add 1 to A and place the sum in A.

(e) In some cases, register transfer only part of registers. For example only OP-Code part from Instruction word must be transferred into the OP-Code register.

Fundamentals Of Digital Computers

This says that only B0, B1, B2, B3 and B4 bits will be transferred into P0, p1, p2, p3 and P4 respectively.

(I) A specific bit in a register also can be transferred.

This transfers A3 bit of register A into B2 of register B.

(g) Sometimes transfers are dependent on certain conditions.

$$R = OR \rightarrow B$$

This means that if R is equal to 0 then the content of Register A is transferred to register

B. The colon is used to indicate a condition operations.

Moreover different companies using different transfer language. For example

$$Bc \rightarrow A$$

$$B = A \quad \text{Instead of } B \rightarrow A$$

2.2.3 Micro Programming

The function of the control unit in a digital computer is to initiate sequence of micro operations. Micro programming is one of the methods for designing the control unit of a digital computer. The control unit initiates a series of sequential steps of micro operations. Each micro operation is a binary variable. These are called as control variables.

The control variables at any given time can be represented by string of 1's and 0's called a control word. To perform various operations, control words should be programmed by varying the combinations of 1's and 0's.

A control unit whose binary control variables are stored in memory is called a micro programmed control unit.

A micro instruction specifies one or more micro operations. A sequence of micro instructions constitutes a micro program. A memory that is part of a control unit is referred as a control memory. This can be a read only memory (ROM).

The content of ROM is fixed and cannot be altered by simple programming. MI the microinstructions are stored in ROM during the hardware production itself.

Microinstruction Format

The microinstruction format for the control memory is shown in

figure. It is a 20 bit micro instruction.

| | | | | | |
|----|----|----|----|----|----|
| 3 | 3 | 3 | 2 | 2 | 7 |
| F1 | F2 | F3 | CD | BR | AD |

Each micro instruction is divided into four fields namely.

1. Micro operations fields [F1, F2, F3]
2. condition for branching field (CD)
3. Branch field (BR)
4. Address field (AD)

The micro operation field consists of three fields F1, F2 and F3 specifying micro operations for a computer with three bits in each field,

The CD field selects status bit conditions of which branching can be done. It consists of 2 bits.

The BR field specifies the type of branch to be used. It consists of 2 bits.

The AD field contains a branch address. It consists of 7 bits.

Simple Micropogram

Each line of the assembly language micropogram defines a symbolic microinstruction. Each symbolic microinstruction is divided into five fields: label, micro operations, CD, BR and AD. The fields specify the following information.

- 1 The label field may be empty or it may specify a symbolic address. A label is terminated with a colon (:)
2. The micro operations field consists of one, two or three symbols separated by commas. There may be no more than one symbol from each F field.
3. The CD field has one of the letters U, I, S. or Z
4. The BR field contains one of the four symbols defined in the given Table.
5. The AD field specifies a value for the address field of the microinstruction. It may be symbolic address or may be symbol NEXT to designate the next address in the sequence. When the BR field contains a RET or MAP symbol, the AD field is left empty.

Fundamentals Of Digital Computers

Symbols and Binary Code for Microinstruction Fields

| F1 | Microoperation | Symbol |
|-----|-----------------------------|--------|
| 000 | None | NOP |
| 001 | AC \leftarrow AC + DR | ADD |
| 010 | AC \leftarrow 0 | CLRAC |
| 011 | AC \leftarrow DR | INCAC |
| 100 | AC \leftarrow DR | DRTAC |
| 101 | AR \leftarrow DR (0 - 10) | DRTAC |
| 110 | AR \leftarrow PC | PCTAR |
| 111 | M[AR] \leftarrow DR | WRITE |

| F2 | Microoperation | Symbol |
|-----|--------------------------------|--------|
| 000 | None | NOP |
| 001 | AC \leftarrow AC - DR | SUB |
| 010 | AC \leftarrow AC V DR | OR |
| 011 | AC \leftarrow AC \wedge DR | AND |
| 100 | DR \leftarrow M[AR] | READ |
| 101 | DR \leftarrow AC | ACTDR |
| 110 | DR \leftarrow DR + 1 | INCDR |
| 111 | DR(0-10) \leftarrow PC | PCTDR |

ANNAMALAI UNIVERSITY

Fundamentals Of Digital Computers

54

| F3 | | Microoperation | Symbol |
|-----|---------------------------------|----------------|--------|
| 000 | None | | NOP |
| 001 | AC \leftarrow AC \oplus DR | | XOR |
| 010 | AC \leftarrow \overline{AC} | | COM |
| 011 | AC \leftarrow Shl AC | | SHL |
| 100 | AC \leftarrow Shr AC | | SHR |
| 101 | PC \leftarrow PC + 1 | | INCPC |
| 110 | PC \leftarrow AR | | ARTPC |
| 111 | RESERVED | | |

| CD | Condition | Symbol | Comments |
|----|------------|--------|----------------------|
| 00 | Always = 1 | U | Unconditional branch |
| 01 | DR(15) | I | Indirect address bit |
| 10 | AC(15) | S | Sign bit of AC |
| 11 | AC = 0 | Z | Zero value in AC |

| CD | Symbol | Functions |
|----|--------|---|
| 00 | JMP | CAR \leftarrow AD if condition = 1 CAR \leftarrow CAR + 1 if condition = 0 |
| 01 | CALL | CAR \leftarrow AD, SBR \leftarrow CAR + 1 if condition = 1 CAR \leftarrow CAR + 1 if condition = 0 |
| 10 | RET | CAR \leftarrow SBR (Return from subroutine) |
| 11 | MAP | CAR (2-5) \leftarrow DR (11-14) CAR (0,1,6) \leftarrow 0 |

Symbolic Microprogram (Partial)

| Label | Microoperation | CD | BR | AD |
|---------------|----------------|----|------|--------|
| ORG 0 | | | | |
| | NOP | I | CALL | INDRCT |
| ADD: | READ | U | JMP | NEXT |
| | ADD | U | JMP | FETCH |
| ORG 4 | | | | |
| BRANCH: | NOP | S | JMP | OVER |
| | NOP | U | JMP | FETCH |
| | NOP | I | CALL | INDRCT |
| OVER: | ARTPC | U | JMP | FETCH |
| ORG 8 | | | | |
| STORE: | NOP | I | CALL | INDRCT |
| | ACTDR | U | JMP | NEXT |
| | WRITE | U | JMP | FETCH |
| ORG 12 | | | | |
| EXCHNAGE: | NOP | I | CALL | INDRCT |
| | READ | U | JMP | NEXT |
| | ACTDR, DRTAC | U | JMP | NEXT |
| | WRITE | U | JMP | FETCH |
| ORG 64 | | | | |
| FETCH: | POCTAR | U | JMP | NEXT |
| | READ, INCPC | U | JMP | NEXT |
| | DRTAR | U | MAP | |
| INDRCT: | READ | U | JMP | NEXT |
| | DRTAR | U | RET | |

2.3) Revision Points

Arithmetic logic unit

The arithmetic logic unit of most computers is capable of performing addition, subtraction, multiplication and division as well as some logical operations.

Control unit

The control unit of a computer sequences the operation of a computer and controls the actions of all other units. Now let us discuss the arithmetic logic unit and control unit in detail.

Opcode

Opcode defines the operation to be performed like addition, subtraction etc and the address part contain the memory location of the number on which the operation is to be performed.

Register transfer language

The way of writing register operations is called register transfer language.

Simple Microprogram

Each line of the assembly language microprogram defines a symbolic microinstruction. Each symbolic microinstruction is divided into five fields: label, micro operations, CD, BR and AD.

2.4) Intext Questions

1. Explain in detail about ALU.
2. Mention the various types of adders and explain in detail.
3. Discuss about Micro Programming.

2.5) Summary

- Instruction Counter is used to point to the memory address from which the next instruction is to be fetched.
- Micro programming is one of the methods for designing the control unit of a digital computer.
- The central processing unit consists of Arithmetic Logic unit and Control unit.
- Multiplexer is a combinational logic circuit, which can select any one of a number of inputs and route it to a single output.

2.6) Terminal Exercises

- 1.Explain about ALU.
- 2.What is the function of adder?
- 3.Give details about Full adder with design.
- 4.What is BCD Adder?
- 5.Explain about Binary half adder
- 6.What is Shift operation?
- 7.Give details about Multiplexer.
- 8.Instruction cycle and execution cycle?
- 9.What is Micro programming?
- 10.Define about Instruction set
- 11.What is control registers?
12. Give details about control set.

2.7) Supplementary Material

Fundamentals of digital logic with VHDL design – Brown

2.8) Assignments

Prepare assignment about Control Unit.

2.9) Reference Books

Malvino, A.P., ‘Digital Computer Electronics’, Tata McGraw Hill, 1991.

2.10) Learning Activities

An individual or group of peoples goes to library for future evaluation of this unit.

2.11) Keywords

- **Central Processing Unit**
- **Arithmetic Logic Unit**
- **Multiplexer**
- **Control Unit**
- **Micro programming**

UNIT - III

3.0) Introduction

Microprocessor is an Integrated Circuit (IC) chip used as the central processing unit (CPU) in a microcomputer. It has a limited set of memory locations known as registers to store the information.

3.1) Objective

At the end of this unit student will get knowledge about the following topics:

- Basic concepts
- Microprocessor Programming
- Instruction sets
- Instruction types and classifications
- Addressing Modes
- Microprocessor types & application

3.2) Content

3.2.1 Basic concepts

Microprocessor can understand a set of basic instructions. It can generate signals to control external devices like memory and input/output devices. Inside the Chip, there is an Arithmetic Logic Unit (ALU), which performs the arithmetic, and Logic Operations. The registers inside the microprocessor store the data on which the operations are performed. There is a control unit (CU) inside the microprocessor generates control signals which also controls the operations of the internal circuitry. Fig 3.1 shows the block diagram of a general microprocessor and the signals available in a typical microprocessor chip.

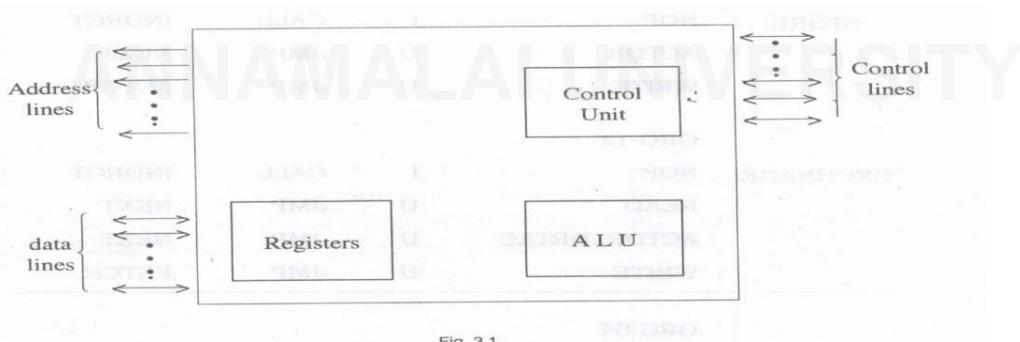


Fig. 3.1

The microprocessor controls memory and input/output devices through a series of connections called buses. Microprocessors select an Input/output device or memory through a set of lines known as Address Bus. For example, if there are 16 lines in an address bus of a microprocessor then it implies $2^{16} (= 65536)$ different addresses can be transmitted through these lines. To transfer data between an Input/output device or memory and the microprocessor, it has a set of lines known as data base. The microprocessor has lines for controlling the external devices like memory, input/output and for providing information about its status. These lines are referred to as control bus. The power of the microprocessor is in its ability to execute millions of Instructions per second from program (set of Instructions) stored in the memory system.

Another feature that makes the microprocessor powerful is Its ability to make simple decisions. Decisions are based upon numerical facts. For example, a microprocessor can decide if a number is Zero, if it is positive and so forth. These simple decisions allow the microprocessor to modify the program flow so programs appear to think.

3.2.2 Microprocessor Programming

Programming is the process of telling the microprocessor exactly how to solve a problem. To do this a program has to be written in a language (assembly language) which the microprocessor understands. A program is a sequence of instructions that operates on data to produce the desire result. The microprocessor has to be programmed before it can execute any task. To program the microprocessor, the problem needs to be organized so that the microprocessor can solve the problem. This is called designing the program. In the design process, the exact step-by step method that will be followed is developed and written down. For example to find the sum of two values stored in a memory location, a sequence of instructions have to be written as follows.

1. Move the contents fromThe memory to the processor register.
2. Addthetwovalues
3. Output the result (or transfer the result to a memory location)

These instructions can be written In assembly language. In an assembly language program the instructions are written using mnemonic codes and symbolic addresses. For example, the instruction. MOV B,A will copy the contents of the register A to another register B. Here MOV is referred as the mnemonic and A and B are the symbolic addresses. However, a microprocessor can understand only machine language programs. Hence assembler program is used to translate the assembly language instructions to the corresponding machine instructions (binary codes) before it is executed.

These instructions (program) are stored in the Random Access Memory (RAM). The microprocessor having been instructed to execute the program, fetches the Instructions (one at a time) from memory and executes it.

3.2.3 Instruction Set

The Instruction set of a microprocessor defines the basic operation that a microprocessor can perform. A program is constructed using the instruction set of that particular microprocessor.

An instruction essentially consists of an operation code (known as OP-Code) and the address(es) of the data on which it has to operate. The data on which an Instruction operates is known as the operend for that instruction. Thus an Instruction is viewed as an entity having two portions namely the Op-Code and the address(es) of the operend(s) as shown below

| | |
|---------|---------|
| OP-CODE | ADDRESS |
|---------|---------|

3.2.4 Instruction Types

An instruction may be one byte or more bytes in length. In any of these types, the first byte indicates the operation to be performed. The remaining bytes, if present contain either the operand or address of the operand on which the operation is to be performed.

Classification of Instructions

Most microprocessors provide the following types of instructions in their instruction set.

1. Data transfer instructions
2. Arithmetic instructions
3. Logic instructions
4. Branch instructions
5. Input/output instructions

1. Data Transfer instructions are provided to move data either between its internal registers or between an internal register and a storage location in memory.
For example.

BLDA 1240H

instruction used by 8085 transfers the contents of memory location 1240 H to the accumulator register A inside the processor.

ii) **MOV DX, CX**

instruction of 8086 transfers the 16 bit information from CX register to DX register within the processor.

2. Arithmetic instructions are provided to perform arithmetic operations like addition, subtraction etc.

For example.

i)**ADD B**

instruction of 8085 adds the contents of B register with that of A register and stores the result in A register.

ii) ADD AL, 1OH

instruction in 8086 adds the value 1OH to the contents of AL register.

3. Logic Instructions perform logic operations like AND. OR. NOT etc.,

For example.

AND AL, OFH

instruction marks the upper 4 bits of the digit stored in AL register.

4. Branch instructions are provided to transfer the control of flow of Instruction execution either conditionally or unconditionally.

For example,

JC LOOP

instruction transfer the control to LOOP if carry flag is set to I.

5. I/O instruction transfer data between the input/output device and the processor register.

Some example programs are given below using the instruction set of 8085 and 8086 microprocessors. Similarly assembly language programs can be written for other microprocessors using the corresponding instruction set of that particular microprocessor.

Example I (for 8085 microprocessor)

To add two 8 bit numbers present in the memory locations 2000 H and 2001 H and store the result in the memory locations 2002 H LDA 2000 H : Move the content from the memory location 2000 H to accumulator register

MOV B, A Move the content from register A to register B

LDA 2001; Move the contents from 2001 H to A

ADD B ; Add the contents of A and B and store the result in A

STA 2002H ; Move the contents (result) from the accumulator to the memory location 2002 H

HLT

The same task Is executed by 8086 microprocessor using the following program.

DATA - HERE SEGMENT

DATA I DB 45H

DATA2DB 121-1

SUM DB ?

DATA - HERE ENDS

CODE - HERE SEGMENT

ASSUME CS: CODE - HERE. DS: DATA - HERE

MOV AL, [1000 H]

MOV BL, (1001 H)

ADD AL, BL

MOV [1002 H]. AL

CODE - HERE ENDS

Fundamentals of Digital Computers

END

Example 2:

To multiply two 8 bit numbers in memory location and store the product in another memory location

```
LDA 2000 H ; transfer the content of memory 2000 H to accumulator  
MOV B, A Move the content from A to B  
LDA 2001 H Transfer the content of memory 2001 H to accumulator  
MOV C, A ; Move the content from A to C
```

```
60  
MVI A, OOH ; Clear accumulator register  
LOOP : ADD C ; Add the content of C to A  
DCR B ; Decrement B by 1  
JNZ LOOP ; If B ≠ 0 then repeat the loop  
STA 2002 H ; transfer the result (product) from accumulator to the  
memory location 2002 H  
HLT
```

The 8085 doesn't have multiply instruction in its instruction set. Therefore, when a programmer wants to multiply two numbers, he has to devise his own method for doing so. The method followed in the above example for multiplication is to add the content of one memory location (multiplication) to the accumulator, multiplier times with the initial value of Accumulator being zero. The same task (multiplying two 8 bit numbers) is implemented with the help of MUL instruction in Intel 8086 as follows:

```
DATA_HERE SEGMENT  
MULTPLICAND DB 42 H ; first byte of multiplicand  
MULTPLIER DB 13 H ; second byte of multiplier  
PRODUCT DW ? ; result  
DATA _ HERE ENDS  
CODE _ HERE SEGMENT  
ASSUME CS : CODE _ HERE, DS: DATA _ HERE  
MOV AX, DATA _ HERE ; initialize DS register  
MOV DS, AX  
MOV AL, MULTPLICAND  
MUL MULTIPLIER  
MOV PRODUCT, AX  
CODE _ HERE ENDS  
END
```

3.2.5 Addressing Modes

The operation code of the Instruction specifies the operation that will be executed after it is fetched from memory and placed in the control unit of the microprocessor. The microprocessor must know where to find the operand or operands upon which the operation is to be performed. Operand (s) may be located in processor registers or memory locations or interface registers. There are different techniques by which the address of the data to be operated upon may be specified. These techniques are known as Addressing modes.

In 8 bit microprocessors, the first byte of an instruction is a combined binary code that specifies both the operation and the addressing mode. This byte, when placed In the instruction register during the fetch cycle, is interpreted by the control unit to determine the operation that must be executed and also the way to go about locating the operands. The availability of different addressing modes gives the programmer flexibility for writing programs that are more efficient with respect to the number of instructions and executions time.

Implied Addressing Mode

In this mode, the operand is specified implicitly in the instruction. Instructions of this type are 1-byte instructions. For example, the Instruction CMA (complement accumulator) is an implied mode instruction because the operand is placed In the accumulator.

Immediate Addressing Mode

Memory immediate mode instructions having an 8 bit operand Is a 2 byte instruction. If the operand is 16 bits then the instruction is a 3 byte instruction. For example, MOV AL, 25H Instruction copies a byte - sized 25 H into register AL. in the 80386 and above, a double word of source - immediate data can be transferred Into a register or memory location (example: MOVE EBY. 1111 1111 H) Instruction copies a double word sized 1 11 1 1111 H into the 32 bit wide EBY register.

Direct Addressing Mode

In this mode, the address of the operand Is explicitly specified given in the address part of the Instruction. In S bit microprocessors with 16-bit address, an instruction consists of 3 bytes. In microprocessors with wider memory words the address part is combined with the operation and mode code bits to combine the entire Instruction in one memory word. Most direct addressing mode instructions assume that the other operands reside in processor registers. If more than one operand resides in memory, the instruction must include additional addresses to specify their locations. Some 8 bIt microprocessors with 16 bit addresses have special direct addressing modes that require only one byte to specify an address. Such microprocessors divide the 2¹⁶ (65,536) bytes of memory into blocks called pages. Each page is usually assigned 256 bytes of consecutive memory

space. A page in memory is then specified by the 8 high-order bits of an address. The 8 low-order bits give the byte within the page. Thus a 64k memory can be divided into 256 pages of 256 bytes each. The first page is called page 0 and the last is page 255. By means of the paging scheme. It is possible to develop some variations in the direct mode of addressing.

For example: MOV CX, DATA
instruction copies the word contents of memory location DATA into register CX

Register Addressing Mode

In this mode the operands are in registers which reside within the CPU. Register-mode instructions can be executed within the CPU without the need to reference memory for operands.

Example: MOV AX, BX

Instruction copies the contents of register BX into register CX. In the 80386 and above, a double word can also be transferred from the source register or memory location to the destination register or memory location.

Example; the MOV ECX, EDX instruction copies the double word sized contents of register EDX into register ECX.

Register Indirect Addressing Mode

In this mode the instruction specifies a register or a pair of registers in the processor whose contents give the address of the operand in memory. Before using a register - indirect - mode instructions, the programmer must ensure that the address of the operand is placed in the processor register with a previous transfer instruction. A reference to the register is then equivalent to specifying a memory addft"3s.

Example: MOV AX, IBX)

Instruction copies the word sized data from the data segment offset address indexed by BX into register AX in the 80386 and above, a byte. a word or double word is transferred between a register and a memory location addressed by any register. EAX, EBX, ECX, EDX, EBP, EDI or ESI.

Relative Addressing Mode

In relative addressing mode instruction, the second part specifies a signed number (e.g. +45 or -52). The address of the operand is calculated by adding the content of the program counter to the signed address in the instruction. Since microprocessors use various addressing modes to calculate the address of an operand, to differentiate among the various addresses involved in the computation, we must distinguish between the

control unit when it executes the instruction. The address of the operand or the address where control branches in response to a jump branch or call Instruction is called the Effective address. In a direct addressing mode instruction, the effective address is equal to the address part of the instruction. In the relative mode, the effective address is computed from the value in program counter plus the address part of the instruction.

Indexed Addressing Mode

The address part of the instruction is added to the value presently stored in the index register to obtain the effective address. The index register is often incremented or decremented to facilitate the execution of program loops and to have access to tables of data stored in memory.

Base Register Addressing

This is similar to the indexed addressing mode, except that the address part of the instruction consists of a number of bits that is less than the number of bits required for a full address. The effective address is calculated by adding the contents of the index register to the partial address in the instruction. The register used in this mode is called a base register. The base register holds a base address, and the truncated address in the instruction specifies a displacement with respect to the base address.

3.2.6 Microprocessor Types And Applications

The 8085 and Z80 Microprocessors: In 1977. Intel Corporation introduced the microprocessor 8085. This was to be the last 8 bit general - purpose microprocessor developed by Intel. The main advantages of 8085 are its Internal clock generator, internal system controller and higher clock frequency. This higher level of component integration reduced the cost and increased the usefulness of 8085. Applications that contain the 8085 are still being used and designed and will likely to continue to be popular well in future. Another company Zilog Corporation, Z80 Is also an 8 bit microprocessor which uses a language code compatible with 8085.

The 8086 / 8088 Microprocessors: Intel released the 8086 microprocessor in 1978 and the microprocessor 8088 later. Both are 16 -bit microprocessor which execute instructions in as little as 400ns (2.5 millions Instructions per second). This is a major improvement over the execution speed of 8085. In addition, the 8086 and 8088 address 1 MB of memory. 16 times more than 8085.(1MB memory contains 1,048,576 bytes of memory locations). This higher execution speed and larger memory size allowed the 8086 and 8088 to replace smaller mini computers in many applications. Another feature found in 8086 / 8088 is an instruction cache or queue that prefetches a few instructions before they are executed. The queue speeds up the operation of many sequences of instructions and proved to be the basis for the much larger instruction caches found in modern microprocessors. The increase memory size (I.e capability to access more memory locations) and additional instruction capacity of 8086 and 8088 have led to many sophisticated applications for microprocessors. improvements to the Instruction included

Fundamentals of Digital Computers

multiply and divide instructions, which were, missing on earlier micro processors. These microprocessors are called CISC (complex instruction set computer) because of- the number and complexity of instructions. Also the number of Instructions are more compared to the earlier microprocessors. This additional instructions ease the task of developing efficient and sophisticated applications. The 16 bit microprocessors also provides more internal storage (register) space than the 8 bit microprocessor. The additional registers allow software to be written more efficiently. The 16 - bit microprocessor evolved mainly because of the need for larger memory systems. The 8088 microprocessor is used In IBM personal computers. Applications such as spread sheets, word processors spelling checkers are memory intensive and require more than 64K bytes of memory found In 8 - bit microprocessors address 1MB memory for these applications. Soon even 1MB memory proved limiting for large spread sheets and other applications. This led to the introduction of the 80286 microprocessor.

The 80286 Microprocessor The 80286 microprocessor Is also 16 bit microprocessor. It is almost identical to the 8086 and 8088, except it addresses a 16 MB memory system Instead of 1 MB. The instruction set of the 80286 is almost Identical to the 8086 / 8088. except for a few additional instruction that manage the extra 15MB of memory. The clock speed of 80286 is increased so it executes some instructions In 250 ns. Some changes In the internal execution of the instructions that led to an eight fold increase in speed for many instructions when compared to 8086 / 8088 instructions.

The 80386 Microprocessor Applications began to demand more memory, faster microprocessor speeds and wider data paths. This led to the introduction of intel 80386 which is a 32 bit microprocessor. The 80386 addresses up to 4 Giga Bytes (1 Gb = 1024 MB) (its address bus is 32 bit wide). The 80386 is also available in a few modified versions such as the 80386 SX. which addresses 16MB of memory through 24 bIt address bus and 16 bit data bus, and the 80386 SL / 80386 SW which address 32 MB of memory through 25 bit address bus and 16 bit data bus. The 80386 SW version contains an internal cache memory that allows it to process data at higher rates.

Another version of 80386 is 80386 Ex which is called an embedded PC because it contains all the components of the AT class personal computer on single integrated circuit. The 80386 Ex also contains 24 tImes for Input / output data, a 26 - bit address bus, a 16 bit data bus, a DRAM refresh controller and programmable chip selection logic. The 32 bit microprocessor is needed because of the size of its data bus, which transfers real numbers that require 32 bit wide memory. In order to efficiently process 32- bit real numbers, the microprocessors must efficiently pass them between itself and memory. It takes 4 read-or-write cycles for them to pass through an 8 bit data bus, but only one read-or- write cycle for them to pass through a 32 bit data bus. This significantly increases the speed of any program that manipulates Real numbers. Most high level languages, spreadsheets and database management systems use real numbers for data storage. Real numbers are also used in graphical design packages that use vectors to plot images on the video screen. The 80386 also Includes a memory management unit that allows memory resources to be allocated and managed by the operating system. Earlier microprocessor left memory management completely to the software. The 80386 includes

hardware circuitry for memory management and memory assignment, which improves its efficiency and reduces software overhead. The instruction set of 80386 is compatible with 8086 / 8088 and 80286 microprocessors. Additional instructions reference the 32 - Bit registers and manage the memory system. The 80486 Microprocessor In 1989, Intel Corporation released the 80486 microprocessor, which incorporates some 80386 like microprocessor, an 80387 like numeric co-processor and an 8KB cache memory system in a single integrated package. The internal structure of 80486 is modified so that nearly half of its instructions executes in one clock instead of two clocks. The 80486 is available in a 50 MHZ executes half of the instruction in 25 ns. The average speed Improvement for a typical mix of instructions is about 50% over an 80386 operated at the same clock speed. Another version, of 80486 with a 66 MHZ double clocked version executes instructions at the rate of 66 Mhz with memory transfers executed at the rate of 33 MHZ. A triple clocked version, the 80486 DX4, improves the Internal execution speed to 100 MHZ. With memory transfers at 33 MHZ. It also contains an expanded 16 KB. cache In place of 84 KB cache that is standard on 80486 microprocessor. Other versions of 80486 are called over drive processors. The overdrive processors is actually a double- clocked version of an 80486 DX that replaces an 80486 SX slower speed 80486 DX.when overdrive processor is plugged into its socket, it displaces or replaces the 80486 SX or 80486 DX functions as a double - clocked versions of the microprocessors.

For example , if an 80486 SX operating at 25 MHZ is replaced with an over drive microprocessor. it functions as a 80486 DX2 operating at 50 MHZ using a memory transfer rate of 25 MHZ. The pentium microprocessor is similar to 80386 and 80486 originally labeled as P5. The two introductory versions of pentium operate with a clock frequency of 60 MHZ and 66 MHZ. The double clocked pentium operates at 120 MHZ. The cache size of pentium that transfers a large amount of memory data to benefit from a cache. The pentium microprocessor address up to 4 GB of memory with a 64 bit data bus. These wider speed vector-generated graphical displays. It also transfers data between the memory system and microprocessor at a high fate.

The Pentium executes two instructions not dependent on each other simultaneously, because It contains two Independent internal integer processors called super scalar technology. This allows the Pentium to execute two instructions per clocking period.

3.3) Revision Points

Microprocessor

Microprocessor is an Integrated Circuit (IC) chip used as the central processing unit (CPU) in a microcomputer.

Instruction Set

The Instruction set of a microprocessor defines the basic operation that a microprocessor can perform. A program is constructed using the instruction set of that particular microprocessor.

3.4) Intext Questions

1. Give details about Instruction set and Instruction types.
2. Mention the types of Addressing Modes. Explain detail.
3. Discuss in detail about Microprocessor types & application

3.5) Summary

- Intel released the 8086 microprocessor in 1978 and the microprocessor 8088 later. Both are 16 -bit microprocessor which execute instructions in as little as 400ns (2.5 millions Instructions per second). This is a major improvement over the execution speed of 8085.
- Another version of 80386 is 80386 Ex which is called an embedded PC because it contains all the components of the AT class personal computer on single integrated circuit

3.6) Terminal Exercises

1. Give the advantages of Microprocessor.
2. Define Microprocessor programming.
3. Mention the types of addressing modes.
4. What is instruction set?
5. Give hints about 8085 and Z80 Microprocessors.
6. What is the difference between 8085/8086 Microprocessors?
7. Give hints about 80286/ 80386 Microprocessor.

3.7) Supplementary Materials

Fundamentals of Digital Logic with Verilog Design – Brown

3.8) Assignments

Prepare assignment about Microprocessor programming.

3.9) Reference Books

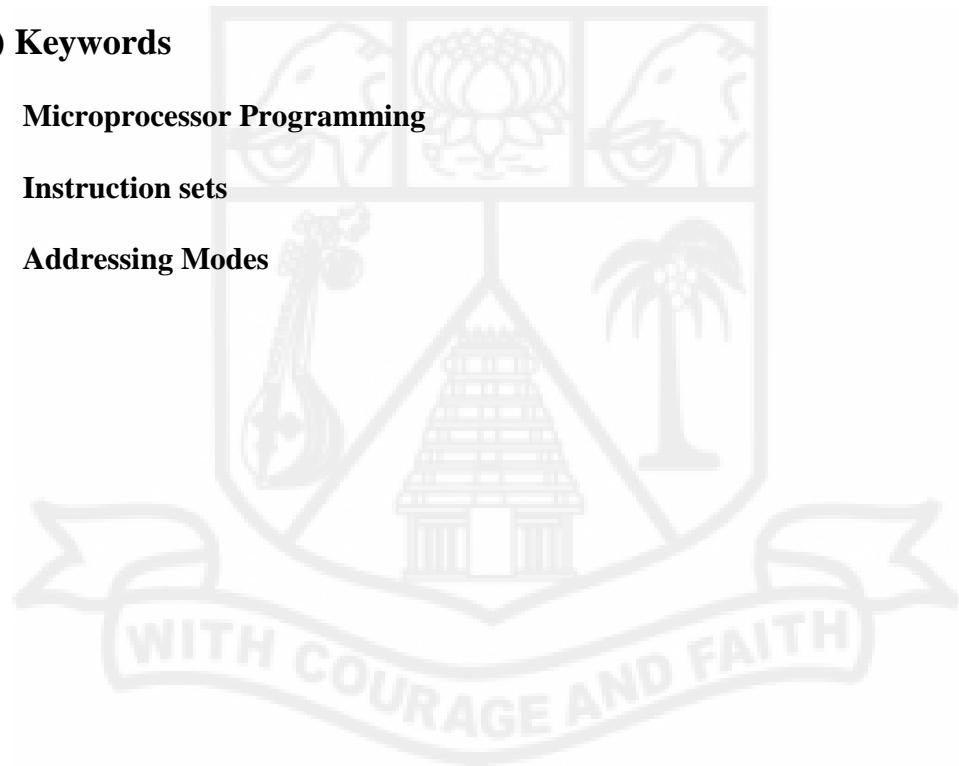
Schneider, 'The Principles of Computer Organizations', G.M.John Wiley Sons, 1985.

3.10) Learning Activities

An individual or group of peoples goes to library for future evaluation of this unit.

3.11) Keywords

- **Microprocessor Programming**
- **Instruction sets**
- **Addressing Modes**



ANAMALAI UNIVERSITY

UNIT- IV

4.0) Introduction

The memory component of a computer system can be divided into three main groups.

1. Internal Processor Memory: This comprises a small set of high speed registers used as working memory for temporary storage of instructions and data.
2. Main Memory: Also called as primary memory is a relatively fast memory used for program and data storage during computer operations. The principal technology used for main memory is based on semiconductor integrated circuits (ICs) .
3. Secondary Memory: Also called as auxiliary memory is much larger in capacity but much slower than main memory. It is used to store system programs, large data files etc. which are not continually required by the Cpu.

Memory Device Characteristics

1. Access Time(tA): It Is usually calculated from the time that a read request is received by a memory unit to the time at which all the requested implementation has been made available at the memory output terminals.
2. Cycle Time(tM): The minimum time that must elapse between the initiation of two different access by the memory can be greater than tA; this rather loosely defined time is called cycle time tM of the memory.
3. Data Transfer Rate or Bandwidth(bM): It is generally convenient to assume at tM is the time needed to complete any read or write operation. Hence the maximum amount of information that can be transferred to and from the memory every second is $1/tM$. This quantity is called the data transfer rate or bandwidth bM.

4.1) Objective

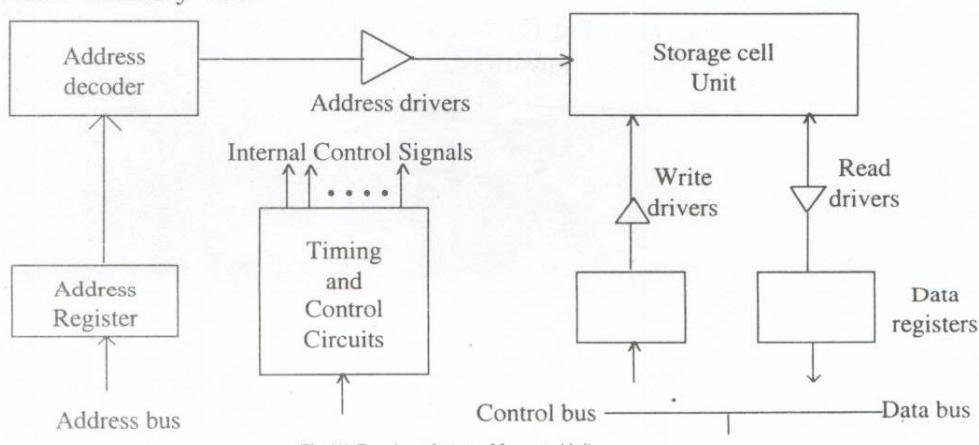
At the end of this unit student will get knowledge about the following topics:

- Random accesses memories
- Decoders
- Connecting memory chips to computer bus
- Static and dynamic ROM, RAM and EPROM memories
- Mass storage's
- Optical storage devices
- Storage hierarchies
- Digital recording techniques
- Virtual and cache memory

4.2) Content

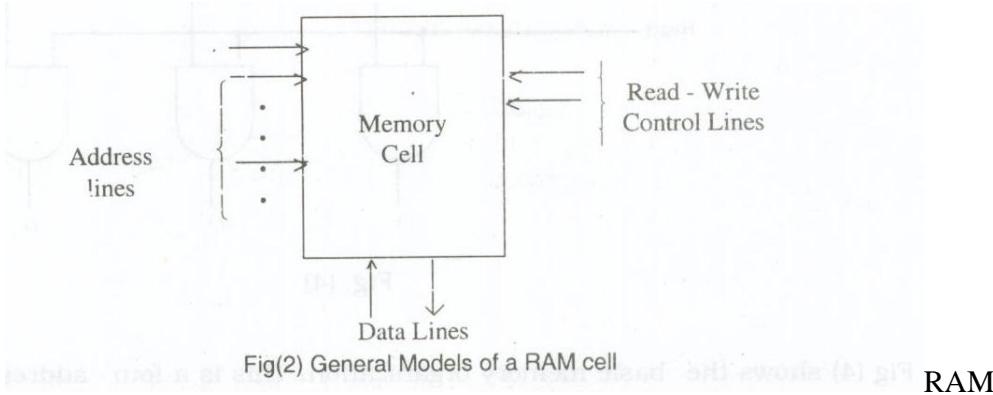
4.2.1 Random Access Memories

Random-Access Memories (RAMs) are characterized by the fact that every location can be accessed independently. The access and cycle times for every location are constant and independent of its position. Fig (1) shows the main components of a Random- Access Memory unit.



The storage cell unit comprises N cells each of which can store I bit of information. The memory operates as follows. The address of the required location is transferred via the address bus to the memory address register. The address is then processed by the address decoder which selects the required location in the storage cell unit. A read -write select control line specifies the type of access to be performed. If read is requested, the contents of the selected location is transferred to the output data register. If write is requested, the word to be written is first placed in the memory input data register and then transferred to the selected cell. Since it is not usually desirable to permit simultaneous reading and writing in the input and output data register. The input and output parts of the data bus may then be merged to form a single bidirectional data bus.

Fig(2) shows an idealized model of a RAM cell and its external connections. The address lines are used to select the cell for either reading or writing as determined by the read-write control lines. A set of data line is used for transferring data to and from the memory. In each line connected to the storage cell unit one can expect to find a driver which acts as either an amplifier or a transducer of physical signals. Thus we find in fig(1) a set of address line drivers and a set of data line drivers. The various drivers, decoders and control circuits are collectively referred to as the access circuitry of the memory unit.



4.2.2 Organization

Fig (2) can be drawn as Fig (3) which depicts a memory cell consisting of an RS flip-flop with associated control circuitry

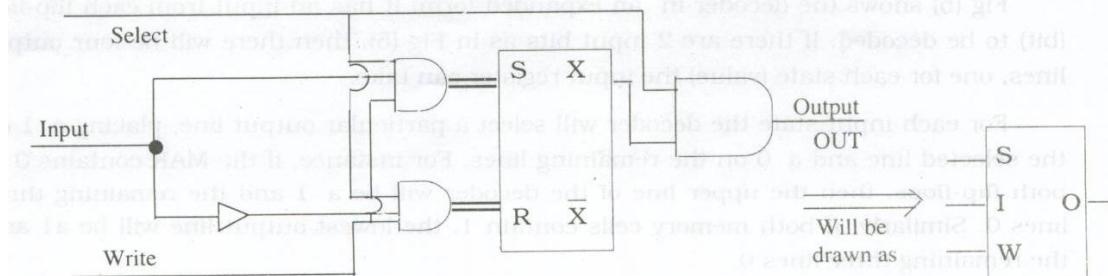


Fig. 3. Basic Memory Cell

Fig (4) shows the basic memory organization.

This is a four- address memory with 3 bits per word. The memory address register (MAR) selects the memory cells (flip-flop) to be read from or written into through a decoder which selects three flip-flops for each address that can be in memory address register.

Fig (5) shows the decoder in an expanded form. It has an input from each flip-flop (bit) to be decoded. If there are 2 input bits as in Fig (5), then there will be four output lines, one for each state (value) the input register can take.

For each input state the decoder will select a particular output line, placing a 1 on the selected line and a 0 on the remaining lines. For Instance. if the MAR contains 0 in both flip-flops. then the upper line of the decoder will be a 1 and the remaining three lines 0. Similarly, if both memory cells contain 1, the lowest output line will be 1 and the remaining three lines 0.

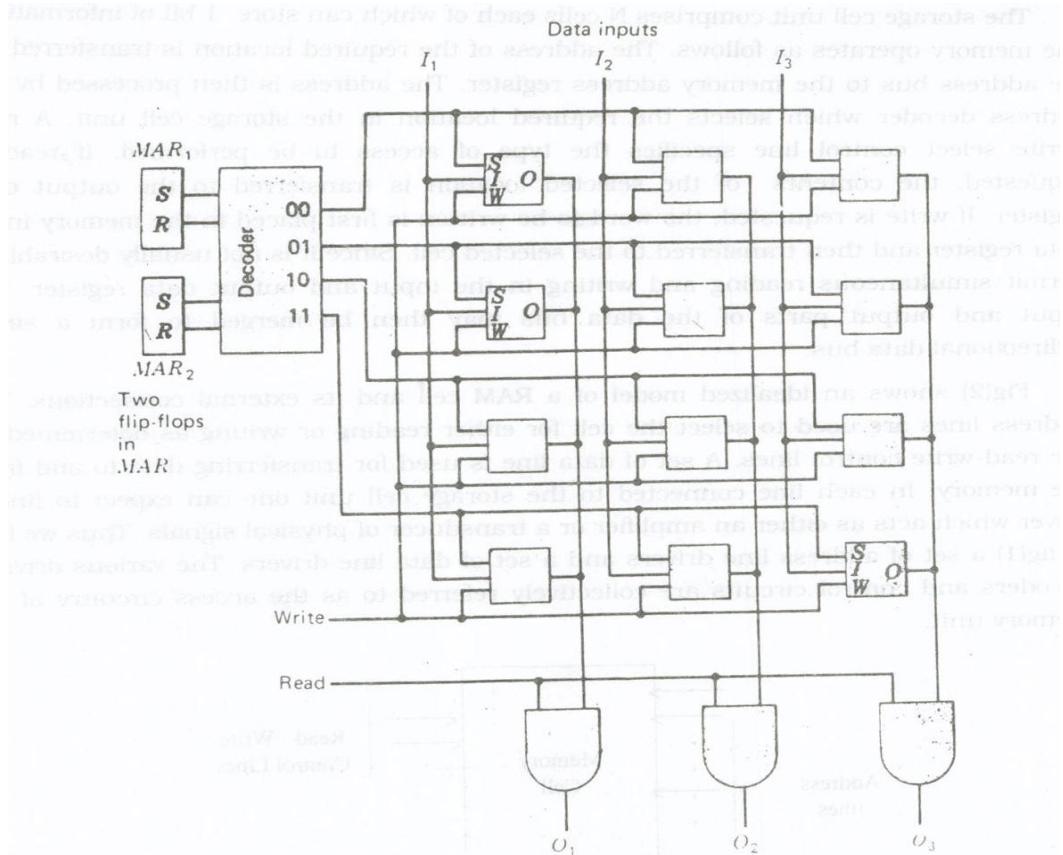


Fig. (4)

The memory iii Fig (4) is organized as follows

There are four words, and each row of three memory cells comprises a word. At any given time the MAR selects a word in memory. If the READ line is a 1, the contents Of the three cells in the selected word are read out on the 01, 02 and 03 lines. If the WRITE line is al the values on 11. 12 and 13 are read into the memory.

The AND gates connected to the OUT lines on the memory cells in Fig (3) must have the property that when a number of AND gate output lines are connected, the output goes to the highest level. If any OUT Is 1. the line goes to 1; otherwise it is 0. This Is called wired OR. In fig (4) all four memory cells in the first column are wire ORED, so If any output is a the entire line will be a 1.

For example. if the second row In the memory contaIns 110 In the three memory cells. andif the MAR contaIns 01. then the second output line from the decoder (marked 01) will be 1, and the input gates and output gates to these three memory cells will be selected.

If the READ is 1, then the outputs from the three memory cells in the second row will be 110 to the AND gates at the bottom of the figure which will transmit the value 11 as on output from the memory.

69

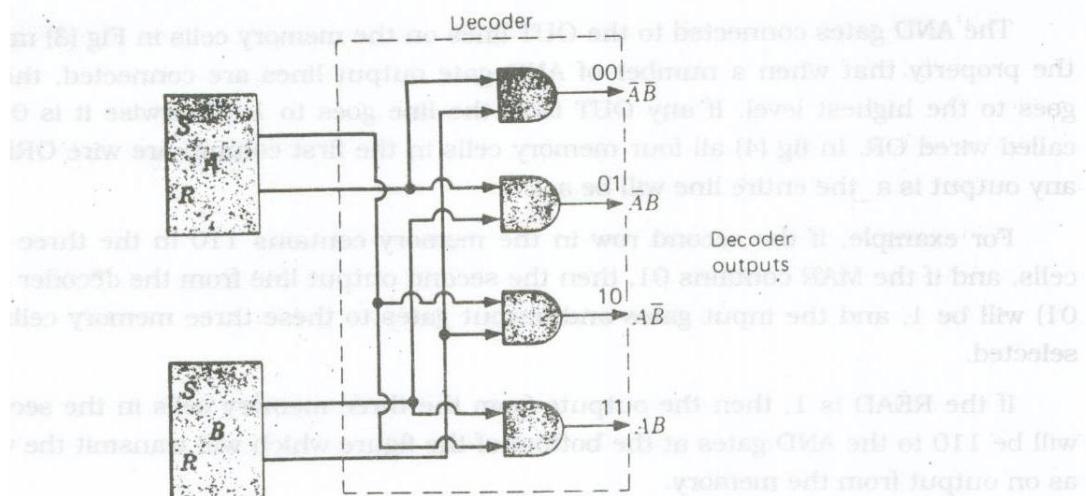


Fig. 5 (a)

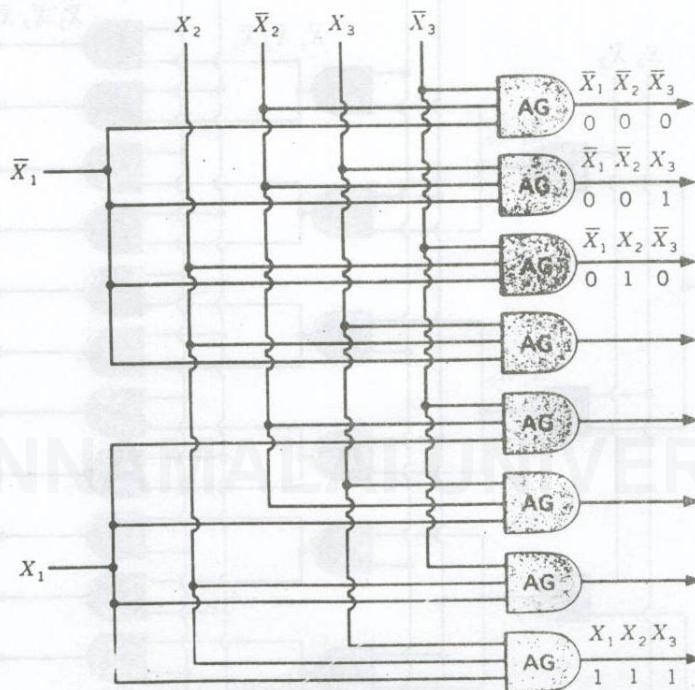
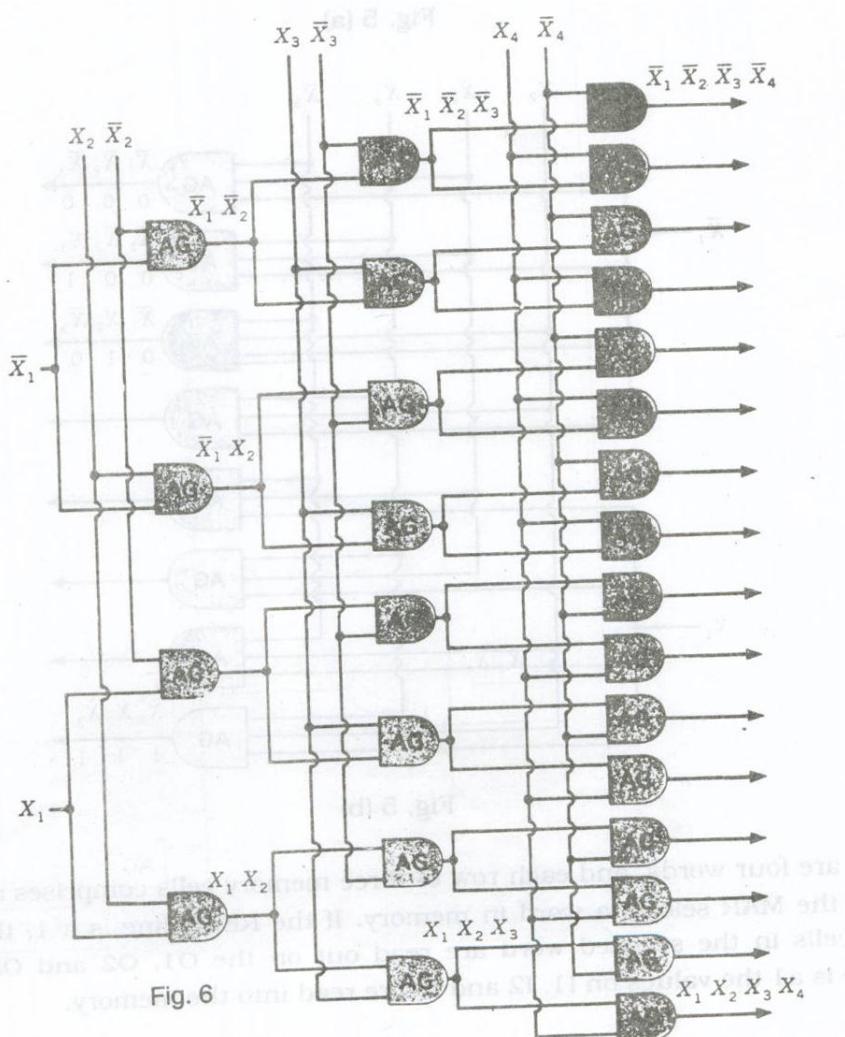


Fig. 5 (b)



If the WRITE line is a 1 and the MAR again contains 01, the second row of flip-flops will have selected inputs. Then the input values on 11, 12 and 13 will be read into the flip-flop in the second row.

Further the number of AND gate is equal to the number of output lines which is equal to 2^n where n is the number of input flip flops being decoded. Total number of decodes is equal to $n \times 2^n$. For Instance, to decode an eight flip-flops register, we would requftt $8 \times 2^8 = 2048$ diodes if the decoder were constructed in this manner.

Fig (6) shows a different types of structure used in building decoder networks. This is called a tree-type decoding networks. This tree network decodes four flip-flops and so has $2^4 = 16$ output lines, a unique one of which Is selected for each state of the flip-flops. An examination will show that 56 diodes are required. while $2^4 \times 4 = 64$ diodes would be required to build the parallel type shown in Fig(5b).

Fig (7) shows another type of decoder network called balanced multiplicative decoder network. This network requires only 48 diodes. It can be shown that decoder network in Fig(3) requires the minimum number of diodes.

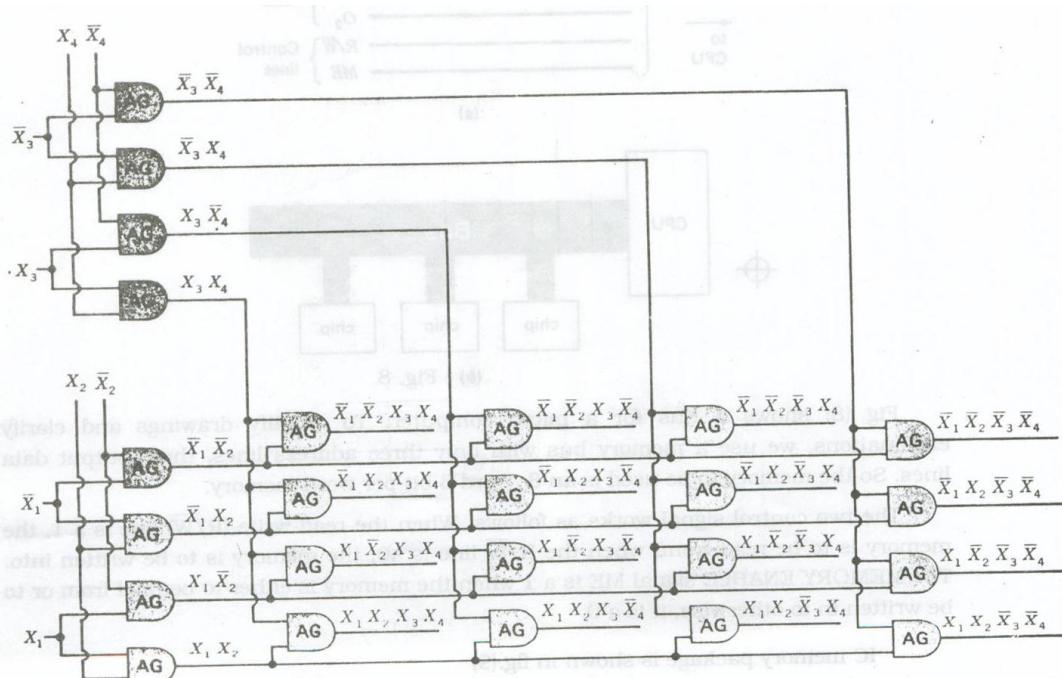


Fig. 7

In general we simply draw the decoder network as a box with n inputs and $2n$ outputs, with the understanding that one of the three types of circuits shown In Fig (I). (2) or (3) will be used In the bx.

4.2.3 Connecting Memory Chips To A Computer Bus

The CPU Is connected to memory by means of a bus. The bus used to connect the memories generally consists of (1) a set of address lines to give the address of the word In memory to be used (2) a set of data wires to Input data from the memory arid output data to the memory and (3) a set of control wires to control the read and write operations.

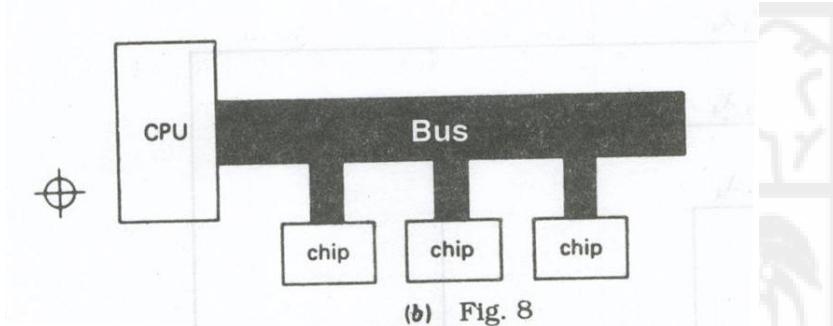
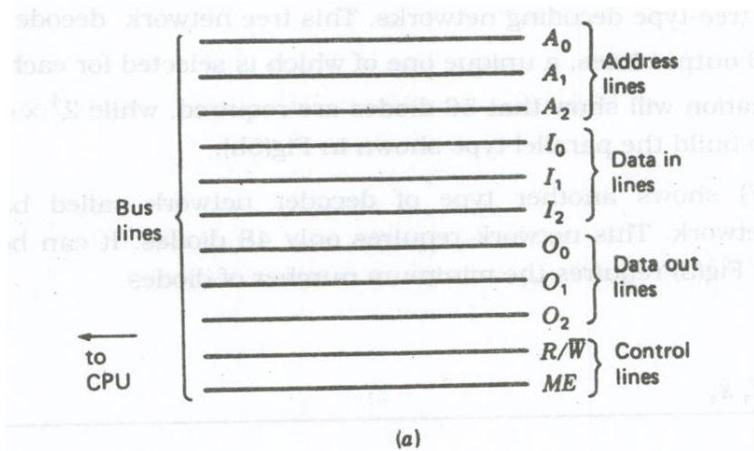


Fig 18) shows a bus for a micro computer. To amplify drawings and clarify explanations, we use a memory bus with only three address lines, three output data lines. So the memory to be used Is an 8- word 3 bit per word memory.

The two control signal works as follows. When the read-write (R/W) line Is a 1. the memory Is to be read from, when the R/W line Is a0. the memory Is to be written Into. The MEMORY ENABLE signal ME is a 1 when the memory Is either to be read from or to be written in to otherwise, it Is a 0.

IC memory package is shown in fig.(9)

The IC memory chip works as follows:

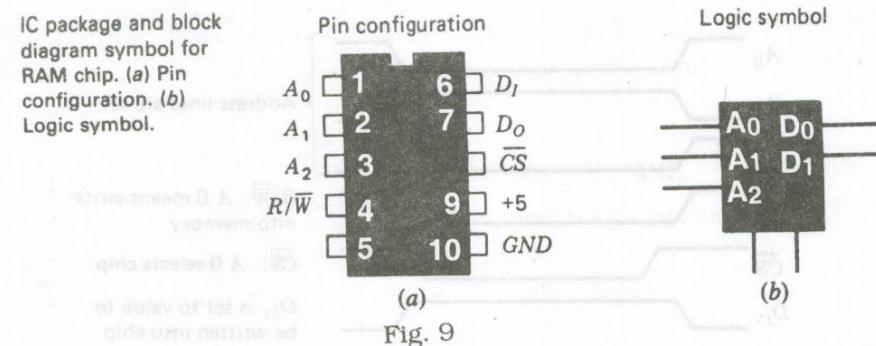
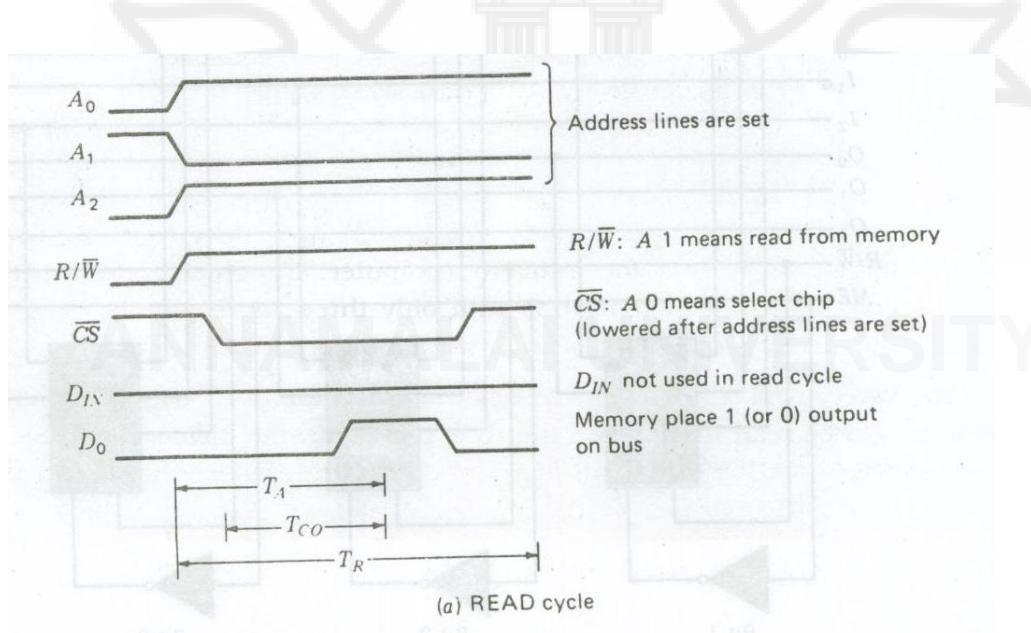
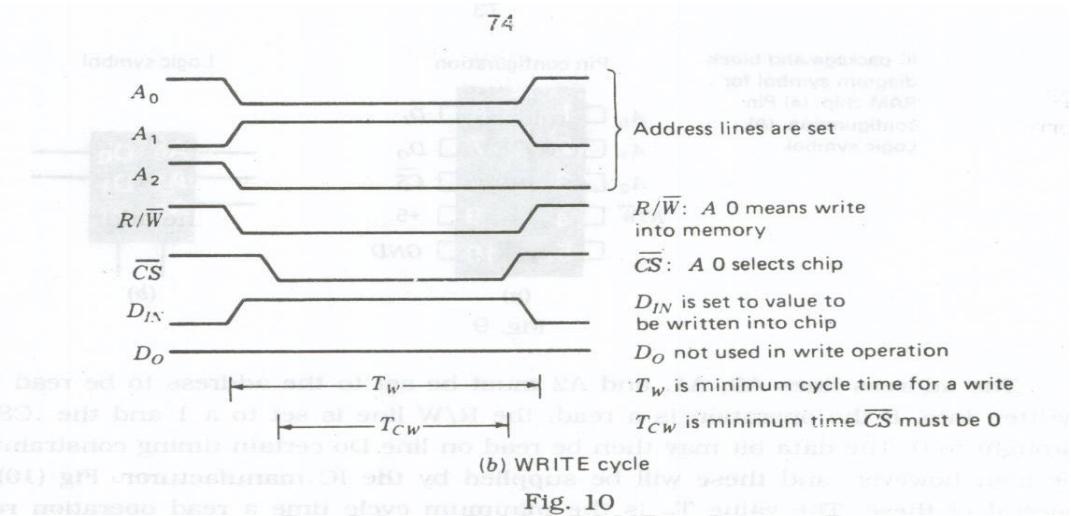


Fig. 9

The address lines A_0 , A_1 , and A_2 must be set to the address to be read from or written into. If the operation is a read, the R/W line is set to a 1 and the CS line is brought to 0. The data bit may then be read on line D_O certain timing constraints must be met, however, and these will be supplied by the IC manufacturer. Fig (10) shows several of these. The value T_A is the minimum cycle time a read operation requires. During this period the address lines must be stable. The value T_A is the access time, which is the minimum time when the address lines are stable until data can be read from the memory. The value T_{CO} is the minimum time when the CS line made a 0 until data can be read. The bus timing must accommodate the above times. It is important that the bus not operate too fast from the chip and that the bus wait for at least the time T_A after setting its address lines before reading and wait at least the time T_A after setting address lines before reading and wait at least T_{CO} after lowering the CS line before reading. Also the address line must be held stable for at least the period TR .





For a write operator, the address to be written into is set up on the address lines the R/W line is made 0, CS is brought down and the data to be read are placed on the D~ line.

The time interval T_w is the minimum time for a WRITE cycle; the time T_m is the time the data to be written into the chip must be held stable. Different types of memories have different tuning constraints which the bus accommodates.

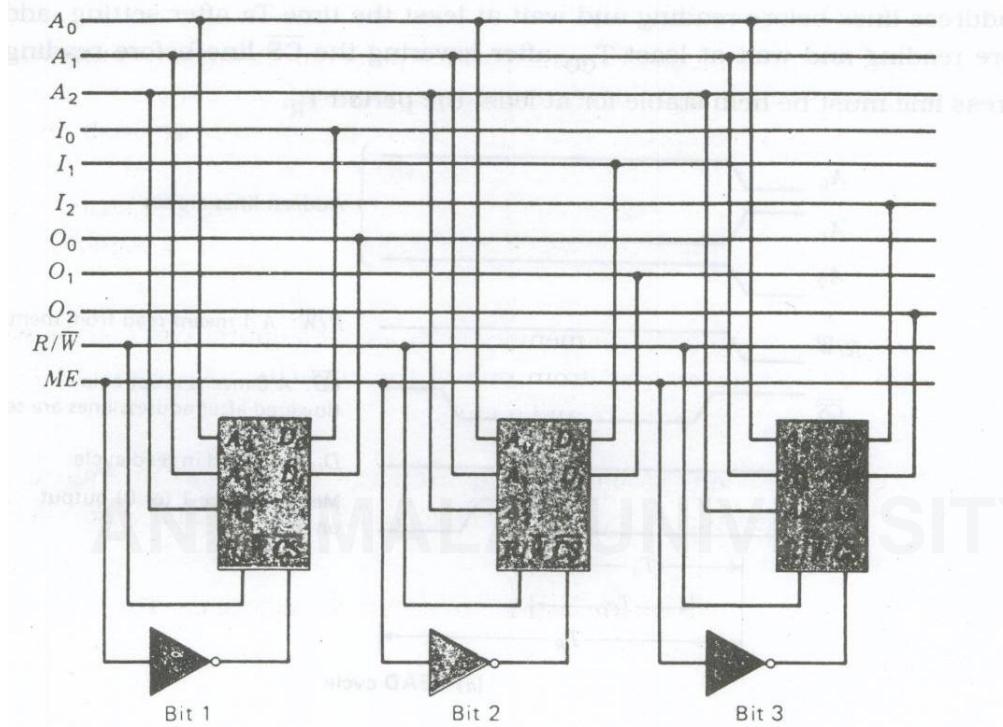


Fig. 11

To form an 8-word 3-bit memory from these IC packages, the interconnection scheme in Fig. (ii) is used. Here the address line to each chip is connected to a corresponding address output on the microcomputer bus. The CHIP SELECT input CS of each chip is connected

to the MEMORY ENABLE output ME from the microprocessor via an inverter. and the R/W bus line is connected to the R/W input on each chip.

If the CPU wishes to read from the memory, it simply places the address to be read from the address lines, put Cl on the [(W line and raises the ME line. Then each chip reads the selected bit onto its output line and the CPU can read these values on its I_i 12 md 13 lines.

Similarly. to write a word into the memory, the CPU places the address to be written into on the address lines, the bits to be written on the 01 . 02. 03 lines lowers [(1W and then raises ME. In practice for microprocessor the memory words now generally contain 8 bits each. There are generally 16 address lines and so 2¹⁶ words can be used in the memory. However memory chips tend to have from 8 to 14 memory address lines.

4.2.4 Random Access Memory

Random access memory can be classified as two kinds as follows:

1. DYNAMIC RAM
2. STATIC RAM

DYNAMIC RAM

The most common memory inside today's personal computers brings RAM to life using minute electrical charges to remember memory states. Charges are stored in small capacitors. A capacitor comprises two metal plates separated by a small distance that's insulated with an electrical insulator. A positive charge can be applied to one plate and because opposite charges attract, it draws a negative charge to the other nearby plate. The insulator separating the plates prevents the charges from mingling and neutralizing each other. The capacitor can function as memory because a computer can control whether the charge is applied to or removed from one of the capacitor plates. The charge on the plates can thus store a single state and a single lot of digital information.

There is always some possibility that a charge will sneak through any material. Moreover, the circuitry that charges and discharges the capacitor also allows some of the charge to leak off. This system will not reliably retain information very long. The capacitor system retains its charge for a few milliseconds before the disappearing charges make the memory unreliable. Those few milliseconds are long enough that practical circuits can be designed to periodically recharge the capacitor and refresh the memory. Because the changing nature of this form of memory and its need to be actively maintained by refreshing it is termed dynamic memory. Integrated circuits that provide this kind of memory are termed dynamic RAM or DRAM.

STATIC RAM

Although dynamic memory tries to trap evanescent electricity and hold it in place. static memory allows the current flow to continue on its way. altering the path taken by the

Fundamentals of Digital Computers

power, using one of two possible courses of travel to mark the state being remembered. Static memory operates as a switch that potentially allows or halts the flow of electricity. A simple mechanical switch will in fact suffice as a form of static memory. The switch, alas has the handicap that it must be manually toggled from one position to another by a human or robotic hand.

A switch that can be controlled by electricity is called a relay. Relay technology was one of the first used for computer memories. The typical relay circuits provides a latch. Like a door latch, this kind of relay circuit stays locked until some force or signal causes it to change. opening the door or the circuit.

Transistors, which can behave as switches, can also be wired to act as latches. A large number of these transistor circuits minimized and combined make a static memory chip. Static RAM Is often shortened to SRAM.

READ-ONLY-MEMORY (ROM)

Note that both the relay and the transistor latch must have a constant source of electricity to maintain their latched state. If the current supplying Item falters, the latch will relax and the circuit will forger. Similarly. if dynamic memory is not constantly refreshed, It alao forgets. When the electricity is removed from either type of memory circuit, the Information that it held simply evaporates, leaving nothing behind. Consequently, these electrically dependent memory systems are called volatile. A constant supply of electricity is necessary for them to maintain their integrity. When there occurs electricity loss the memory loses Its contents. In the old-fashioned world of relays, you could permanently set memory in one portion or another by careful application of a hammer. All that you need are switches that don't switch more accurately, that switch once and Jam. This permanent kind of memory is so valuable In computers and it is called READ ONLY MEMORY or ROM. They are called read-only since the computer that they are installed In can store new code in them. Only what is already there can be read from the memory.

PROM AND EPROM

PROM is programmable ROM. This style of circuit consists of an array of elements that work like fuses. Like ordinary fuses, the fuses In EPROM can be blown to stop the electrical flow. All it takes is a strong enough electrical current, supplied by a special machine called a PROM programmer or PROM burner. PROM chips are manufactured and delivered with all of their fuses intact. The PROM Is then customized for Its given application using a PROM programmer to be low the fuses one by one., according to the needs of the software to be coded inside the chip. This process Is usually termed burning PROM. The effects of burning a PROM are permanent. The chip cannot be changed to update or revise the program inside.

EPROM

Technology has brought an alternative to the PROM the Erasable Programmable Read_only-Memory chip or EPROM. Sort of self heading semiconductors, the tha Inside

Fundamentals of Digital Computers

an EPROM can be erased and the chip can be reused for other data or programs. EPROM chips are easy to spot because they have a clear window in the centre of the top of their packages. The chip is erased by shining high intensity ultraviolet light through the window. If ray light should leak through the window. The chip could be erased.

4.2.5 Mass Storage

HARD DISK DRIVERS

The Hard disk drive, also called "Winchester" drive, "nonremovable" disk, and "fixed" disk manages to support a huge amount of memory in a very small space. As you might suspect, this is not a simple matter, so a hard disk drive is a fairly complicated piece of equipment. The mechanical parts of the drive are important, of course, but the controlling software is just as important. the disk operating system (DOS) handles all of the "housekeeping" details.

The mechanical parts of the hard disk are sealed inside a metal case. Many hard disk drives are interfaced to the computer via a high-speed "SCSI" bus. The disk platter itself is made of aluminum, and Is coated with a very thin layers of nickel-cobalt or ferromagnetic material. This ferromagnetic material can store signals recorded by a Read/Write head. On the larger capacity drives, several of these platters are stacked together. A separate head is provided for each side of each disk. For example, a drive with two disks would have four separate Read/Write heads.

The Read/Write heads are moved by a stepper motor, which drive a mechanical linkage. All of the Read/Write heads move at the same time, but only one the heads is activated at a time.

The stack of disks is turned at 3600 rpm by a drive motor. (Note that this is twelve times faster than the 300-rpm speed of a floppy disk). The spindle braking/locking mechanism acts as a brake to slow the disks When the drive is switched off. Once the disks stop, the spindle lock holds them in place.

Much of the space on the disk Is used for "housekeeping" functions. On a drive that can store 20 megabytes of user data, the disk must be able to handle over 200 million digital "is"; and "Os". To fit this much information on a small disk, each of the digital storage spaces must be very small. The recorded tracks must be very close together and the disk itself must be very rigid. The motor bearings must hold the disk assembly very firmly, with no sign of free play. Most drive may be operated with the disk turning either horizontally or vertically.

When the drive is operating the Read/Write head never touches the surface of the disk. Instead the head flies above the disk on a very thin layer of air. This layer of air keeps the head 10 to 50 micro inches (0.0001 to 0.00005) above the surface of the disk.

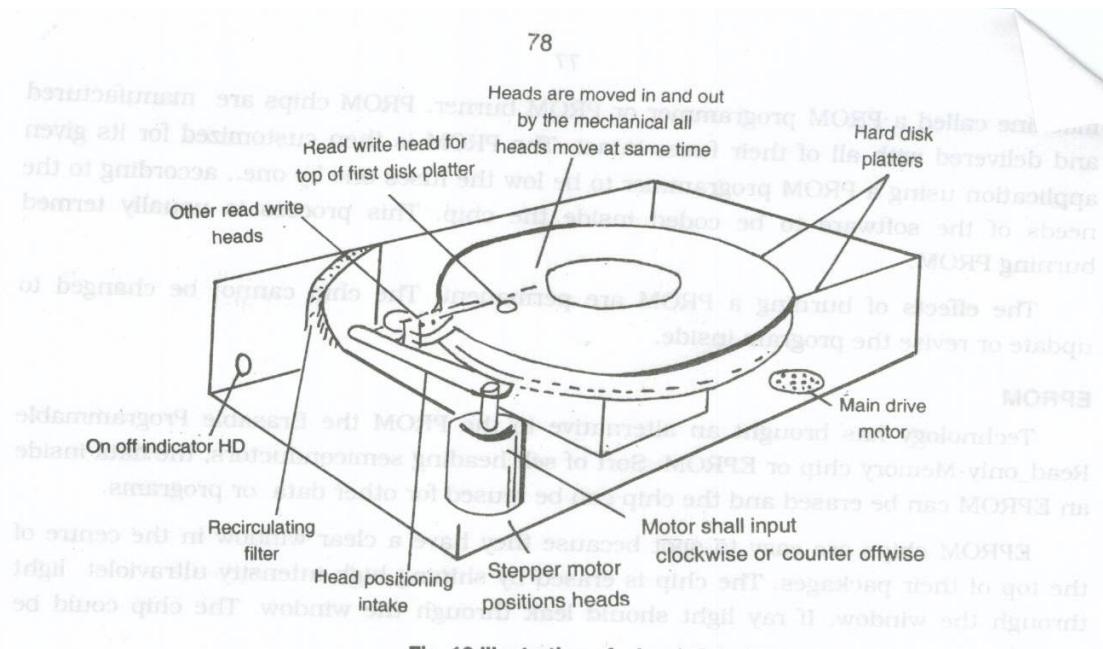


Fig. 12 Illustration of a hard disk drive

The drive mechanism is enclosed in an air-tight metal case, as hard disk drives can be damaged by the smallest particle of dust. The case is sealed to keep these dust particles out of the drive. A "recirculating filter" is provided inside the case. As the disks turn, they create an air flow inside the case. This air is routed through the recirculating filter, which then catches any dust particles that maybe present. The recirculating filter typically catches any particles bigger than 0.3 micron

(10 micro inches or 0.00001 inch).

A hard disk drive also has a opening through the case which is protected by a "barometric filter". This opening provides a way of equalizing the air pressures inside and outside of the case. Consider what would happen if the case was completely sealed and you took the drive up in an airplane. The case would tend to expand, since the air pressure outside the case would decrease but the pressure inside the case would be the same. The barometric filter allows a small amount of air to leak into or out of the case, so the pressures can equalize.

The disk itself is arranged into a system of "tracks" and "sectors" similar to the arrangement on a floppy disk. On a hard disk can hold much more data. As a result the hard disk however, the tracks one much closer together can hold much more data. A hard disk might have 300 tracks or more. The sectors used on hard disk systems also tend to be larger than those on floppy disks. Hard disks usually have 256 bytes or 512 bytes per sector. Track 000 usually holds the "housekeeping" files which tell the computer where to find the various files and sectors. As on the floppy disks, these are called the directory and the file allocation table (FAT). Before the drive can read a particular sector, it must

first refer to the directory and the FAT. This means the drive Is constantly using Track 000.

Since many hard disk drives have more than one disk platter, a special system Is used to address a particular sector. This is illustrated in Hard Disk sector addressing. Let's start by accessing a particular track, say Track 50. One track, labeled Track 50, is located on top of the first platter. and another is located on the underside of the platter. A third Track 50 is located on the top of the second platter, with still another Track 50 found on the underside of that platter. We could group all of these "Track 50" tracks together and say that the formed a "cylinder". Thus "Cylinder 50" includes any "Track 50" that Is included on any of the platters.

Now let's say we wanted top read the "Track 50" that is written on the top of theplatter. The Read/Write head that would be used for this Job would be "Head 1". If we wanted to read "Sector 5" on that track, the complete address would be "cylinder 50, "Head Sector 5" The whole surface of each disk is not used for memory storage. Some of the Inner tracks of the disk are left "blank" and are not coated with oxide material, The Read/Write heads may be "parked" over these blank tracks when the drive is not In use. FLOPPY DISK DRIVES The floppy disk drive offers the user a convenient way of storing large amounts àf information.

Construction of Floppy Disk Drive

Let's examine the mechanical components and the circuits inside a floppy disk drive. Fig. shows a single-sided 3.5-inch drive. A single-sided floppy disk drive has a single head that handles both the "read" and "write" functions. The read/write head is built around a core of soft Iron, as shown in Fig. A coil of wire is wrapped around this core. The coil is actually tapped in the center, so that it behaves as if it were two separate coils. When the control circuits want to write something on the diskette they send a current through these coils. This creates electromagnetic lines of force in the soft iron material. The "air gap" at the bottom of the core acts to "focus"this force on the oxide layer on the surface of the diskette. This creates a magnetized spot on the oxide layer.

80

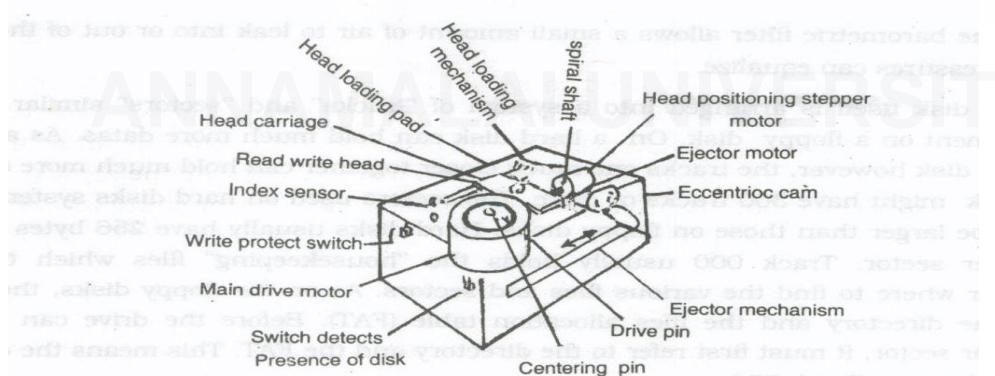


Fig. 13 Illustration of 3.5 inch disk drive mechanism

When the control circuits want to read from the disk, the process is reversed. As the magnetized spots pass under the head, they Induce lines of force in the soft iron core. which then creat tiny currents In the coils of wire. Other circuits then amplify this weak

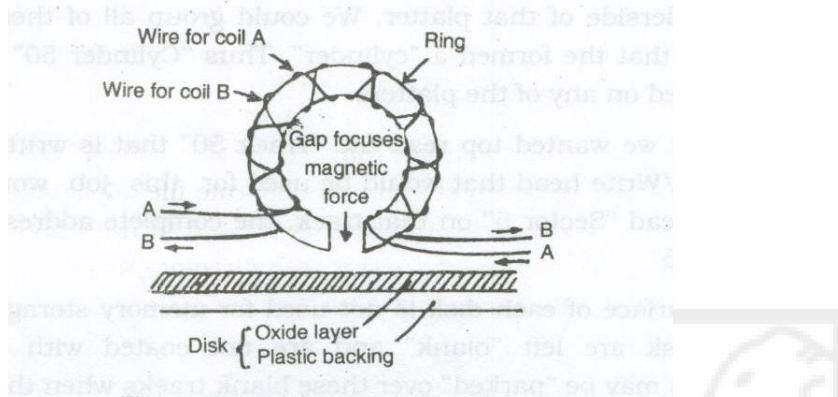


Fig. 14 Illustration of read/write head

Clamping Mechanism

The 3.5-inch-type of drive uses an Interesting clamping mechanism to hold the disk assembly firmly in position. First, the user slides the disk assembly into the drive. A metal plate is suspended above the disk assembly. When the disk in inserted, this plate is lowered, clamping the hub of the disk against the top of the drive motor.

Motor-Control Circuit

A motor-control circuit turns the drive motor on and off. As the motor turns, a sensor detects its speed and reports back to the control circuit. The controller uses this "feedback" to make corrective adjustments and keep the drive motor speed.

Head Carriage

The read/write head is attached to a carriage, which can move both toward and away from the center of the disk. By moving the head carriage in and out, the drive can position the head to read different parts of the disk. On this particular drive, the head carriage is moved by a stepper motor which drives a spiral shaft. As the shaft turns, the carriage Is forced in or our. The rotary motion of the stepper motor cause the carriage to move with a linear motion.

Ejection System

The drive illustrated if Fig. has an automatic ejection system. When the control circuits want to eject the disk, they turn on a small ejector motor. This motor drives an eccentric cain, which moves a lever. When this lever is moved, it triggers a series of mechanical actions, and the disk assembly is forced out of the drive.

Track 00 Sensor

The track 00 detector switch is tripped when the head is reading the outermost track on the disk. The index sensor generates a pulse each time the disk rotates. Another switch may also be included to tell the control circuits when a disk assembly has been inserted in the drive. A floppy disk can store many files, and each file may be stored on tracks in a large number of sectors. The computer must have a way of finding each file, and must have a way of finding the sectors that contain that file. In the type of DOS used by the IBM PC5, the information on the sectors is stored in a separate file allocation table, abbreviated as "FAT". A list of the files stored on the disk is located in a section of floppy disk called the "directory". On some computer system, the directory also tells the computer how to find the sectors which make up each file.

Over the years, engineers have been able to pack more and more data on to floppy disk. Every year seems to bring a new format, offering more data storage in the same space. As you work with floppy disks, you will find many differences. However, the arrangements we have just described applies to most of the floppy disk drives built to date.

MAGNETIC TAPE DRIVE MECHANISM

General aspects

In a tape-storage system the recording medium is tape which is usually between 0.38 cm and 2.54 wide, about 0.025 mm thick and made from polyester coated with magnetic oxide (about 100L'm thick). The tape is wound on reels, the larger variety being 26.7 cm in diameter. Reading and writing is effected by moving the tape past and in contact with, stationary read-write heads. Across the tape several tracks are usually read or written simultaneously, requiring one read/write head per track. Separate read and write heads can be used to allow a writing process to be immediately checked by a subsequent reading process.

Computer Peripherals

One data character (or byte,) together with an associated (vertical) parity bit are recorded at a time. A group of characters is recorded or read together and is known as a record (or a block). One record is separated from another record by a blank inter record gap, often 1.52 cm in length which allows the tape to be positioned for recording or replay. The number of character in a record is defined by software. Each record is identified by a header.

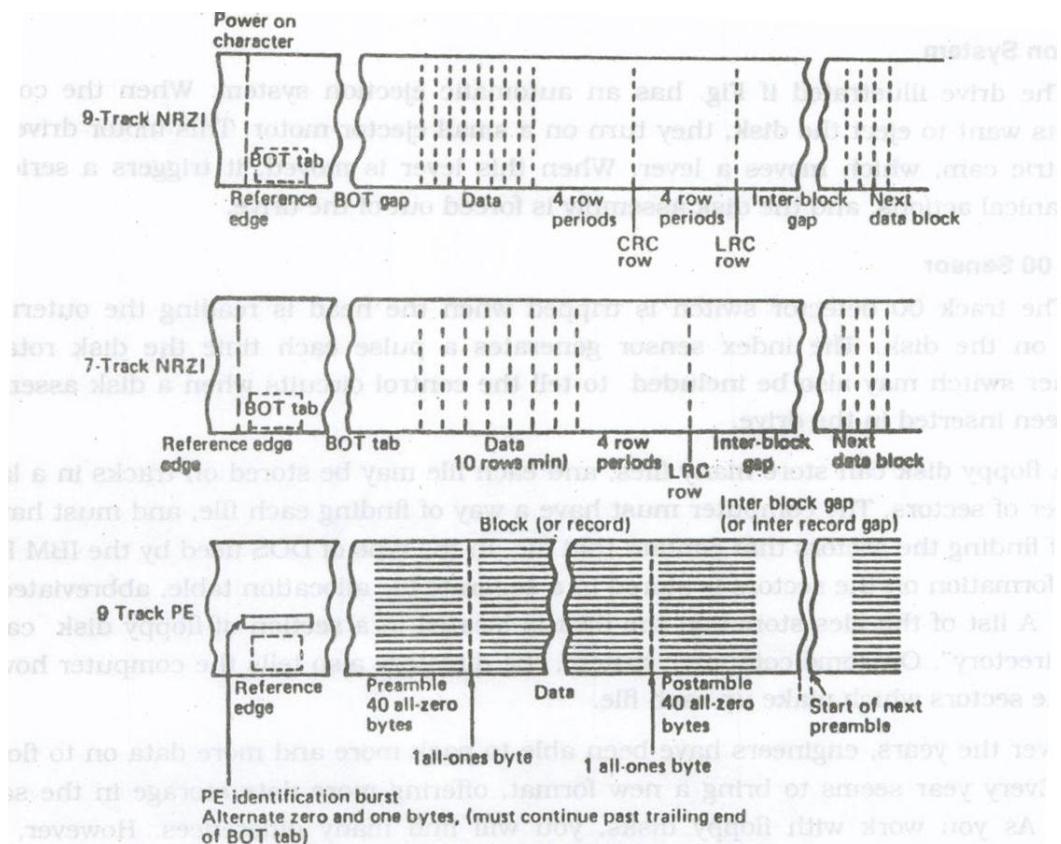


Fig. 15

There are physical markers to indicate the ends of the tape, in the form of metallic foil or small holes, that can be sensed, and software markers in the form of EOT (end of tape) and BOT (beginning of tape) characters recorded on the tape. The method of recording is traditionally NRZI but other codes are used, example MINRZI. Write protection of specific reels of tape can be afforded by having an annular groove in the reel. If a ring is fitted, this can be mechanically detected with the reel in position and this serves to indicate the write protection is not to take place. (Similar schemes can be applied to exchangeable discs).

Larger tape transports

The tape is often 1/2 inch (1.27) wide with 9 or 7 tracks (8 and 6 data bits plus parity), held on reels of 10 1/2 inch (26.7 cm) diameter. 1 inch (2.54 cm) wide tape with more tracks have been used. The tape is propelled past separate read and write heads at high speed, perhaps 300 cm/s for reading and writing. The tape is maintained at the correct tension at all times, even during acceleration and deceleration, by employing vacuum chambers which allow some slack in the tape and isolate the heavy reels from the capstan drives. A capstan is a roller that is rotated at constant speed by a motor. The tape is pressed against the capstan by a pinch roller, causing the tape to move at constant speed. An optical system senses the tape in the chamber and signals when the reels should release or take up more tape. The reels are driven separately from the capstan drives.

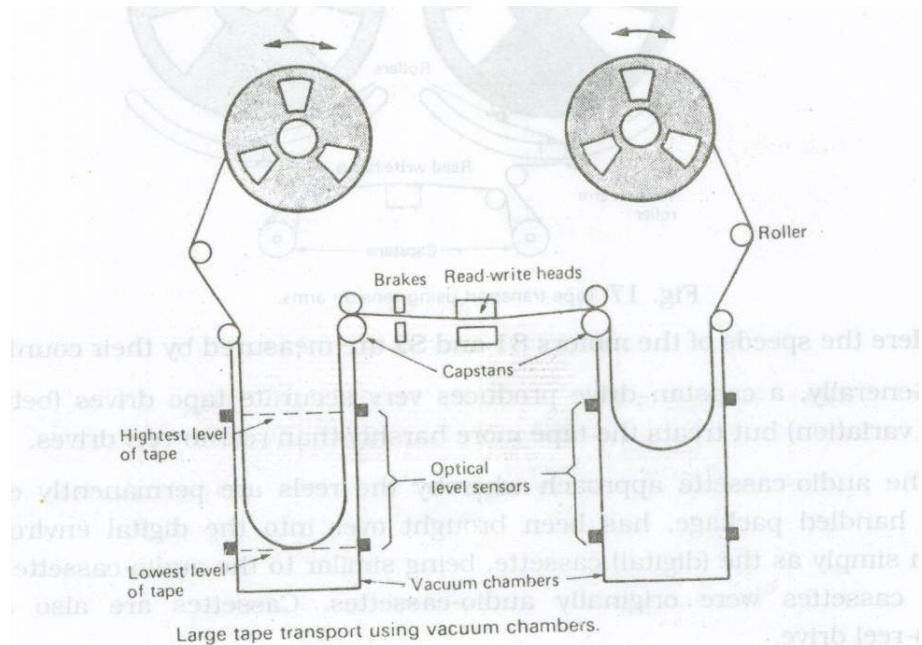


Fig. 16

For slower speed systems (less than 100 cm/s.) the vacuum chambers can be replaced by tension arms. The reels of tape with this arrangement often measure 7 inch(17.5 cm) or 8 1/2 (21.6 cm).

In the larger systems the reels often hold 730 m of tape of more and packing densities of 4046 bytes/cm are common, using NRZI recording with transfer rates of 320 KBs/s. Recent advances in design have enabled packing densities in excess of 15,000 bytes/cm to be achieved and transfer rates of over 1200 KBs/s width a reduced inter-record gap of 0.76 Cm. Large tape transport using vacuum chambers.

Smaller Tape Drives

In a smaller tape system where the reels of tape are a few centimeters in diameter vacuum chambers or tension arms are unnecessary. There are two principal methods of transporting the tape past a read/write head. One is to use capstans to drive the tape at constant speed and the other is to drive the reels only, the so-called reel-to-reel drive. In the latter there is a tape-speed variation of perhaps 2.5 to 1 between the situation in which one reel is full and that in which the other reel is full, unless there is a servo feedback mechanism, which is frequently incorporated.

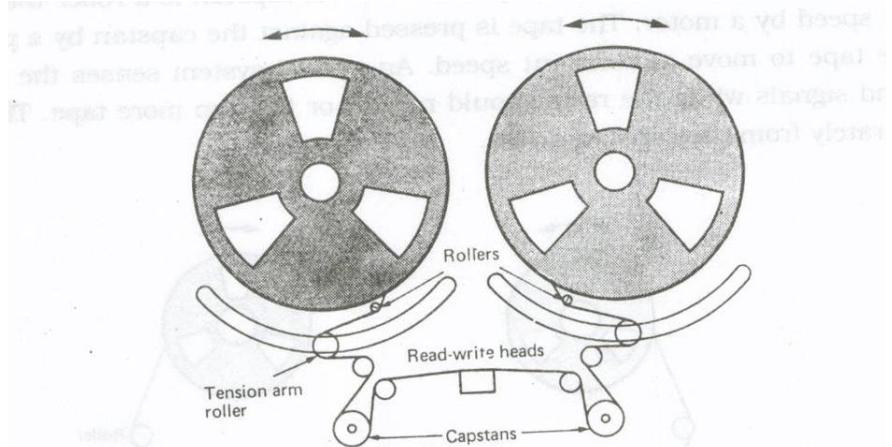


Fig. 17 Tape transport using tension arms.

Here the speeds of the motors S₁ and S₂ are measured by their counter emfs. Generally, a capstan drive produces very accurate tape drives (better than $\pm 2\%$ speed variation) but treats the tape more harshly than reel-to-reel drives. The audio-cassette approach whereby the reels are permanently enclosed in an easily handled package, has been brought over into the digital environment and is known simply as the (digital) cassette, being similar to the audio-cassette. Indeed, some early cassettes were originally audio-cassettes. Cassettes are also available with reel-to-reel drive.

Special digital cassettes, known as cartridges, have been designed specially for digital applications. For these tape tension should be constant, and stopping, starting forward and reverse directions is under digital control. There are two reels of unequal size. The tape passes in the forward direction from the supply reel (the larger reel) to the take-up reel. Tape tension and speed is maintained constant by a unique drive band that passes around the tape on the supply and take-up reels and around capstan that is driven from outside the package. This scheme does not require pinch rollers. Cassettes and cartridges can be single, two or four track, unidirectionally phase encoded and with packing densities of 3.14-628 bits/cm. Tape read/write speeds are in the region 7-100 cm/s (higher for wind-rewind), data capacity over a million bytes for cassettes and up to 20 million bytes for cartridges. The tape is either 0.38 cm or 0.64 cm wide.

These systems can be used to replace paper-tape input in addition to normal bulk storage. For increased freedom from environmental effects two track recordings using complementary codes are used.

Streaming

Tape systems are suited for archival storage and back-up storage. For example a copy of a Winchester disc can be made on a tape at the end of transaction to afford protection against loss of data due to a disc crash or other fault. A convenient tape mode known as streaming can be used. In a traditional tape operation, inter-record gaps (IRGs) are inserted into the recording to enable the tape to be stopped between recordings. In the

Fundamentals of Digital Computers

streaming mode, the tapes is not stopped between records (blocks) and the complete contents of a disc (for example) are recorded without stopping the tape. Hence it is not necessary to include TRG5 specifically to stop the tape at the beginning of a block, a process stoping the tape which causes an overrun and slow reversing process and forward motion necessary to position the tape. This process would be performed infrequently in the steaming mode. The tape format can be the same as the conventional start-stop mode to allow the tape to be accessed in a slower stan-stop operation.

Tape Cassettes And Cartridges

The chargeable tape cassette used in the familiar home recorder is an attractive means for recording digital data. The cassettes are small, chargeable and Inexpensive; they are frequently used in small and home computers. There are also larger tape cartridges which contain long strips of magnetic tape and which resemble large cassettes. These cartridges provide a more convenient way to package tape and greatly simplify the mounting of tape reels. The tape cartridges also provide, protection against dirt and contamination, since the tape is reeled in the cartridges. A number of different cassettes and cassette drives are now being produced.

Cartridges

These are high performance magnetic tape storage medium. The 3M cartridge and drive is shown in Fig (18). The cartridge contains 300ft of 1/4 inch tape capable of recording up to four tracks at 1600 bits/in for a maximum storage capacity of more than 2×10^7 bits. The 3M transport operates at 30 in/s during reading or writing and at 90 in/s in the search mode. A novel elastic band drive moves the tape and also supplies tape tension. Tape drive hub and guide components are referenced to the base of the cartridge and require no external guidance. Several new cartridge systems have been designed to back up hard disk drives. Because most Winchester (hard-disk-drives) drives have fixed disks, there is need to back up, that is to write and store elsewhere, the contents of Winchester disks.

This can be done with floppy disks, but they do not have great capacity per cartridge, or with standard 0.5 inch tape drives, but these are liable to be expensive for small systems. Thus a small inexpensive type of drive, the streaming tape drive, computers In the market for the Winchester backup function.

Magnetic Bubble Memories

Magnetic bubble memories are mainly used in specialized applications. These types of memories are highly reliable. Magnetic bubble memories have been manufactured with capacities of 1M or more bits per chip. Materials such as garnet exhibit properties such as surface magnetization. The natural direction of magnetization are perpendicular to the surface of the plates. These areas are long and tubular hence we shall refer to them as serpentine areas or DOMAINS. The materials inside two adjacent domains are magnetized in opposite directions. In the event of an opposing magnetic field being

Fundamentals of Digital Computers

applied perpendicular to the surface plate, opposite to the direction of magnetization of the domains, these domains tend to curl up and form cylindrical domains called MAGNETIC BUBBLE& A typical bubble diameter is around 1 micro meter.

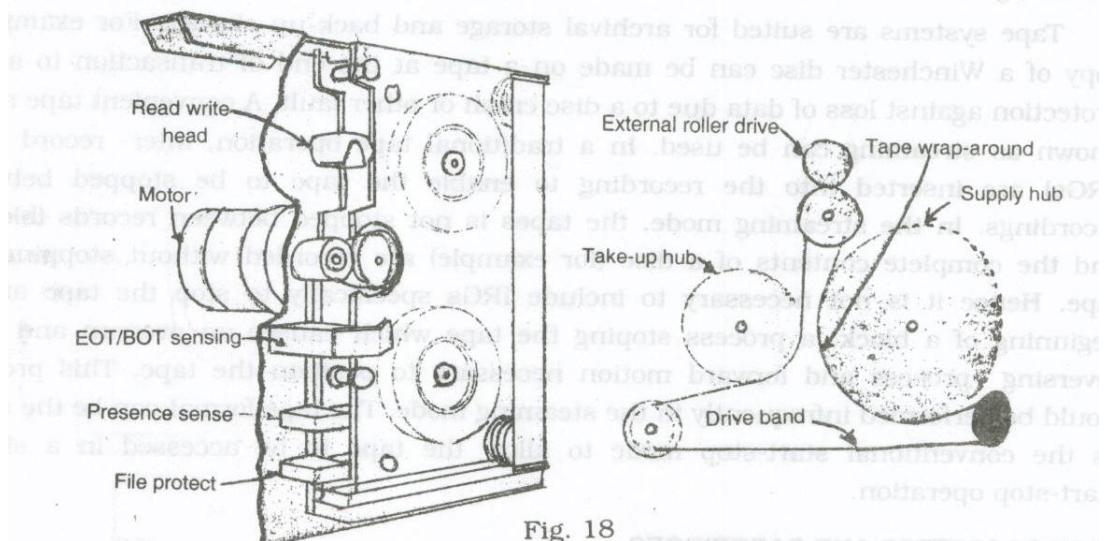
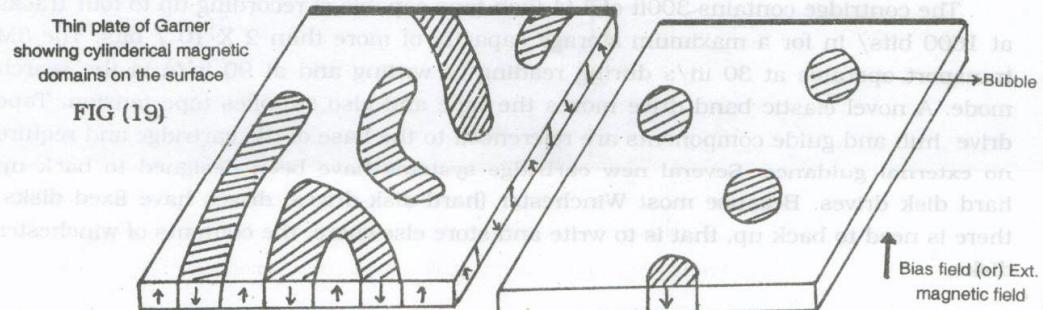


Fig. 18

Formation of Magnetic Bubbles



These bubbles can be moved at high speeds around the plate by applying a drive field whose direction is parallel to the plate surface. The rotating drive field (rotated at fixed rate) is produced by an electromagnet. Linear tracks of permalloy material are deposited on the surface of the plate so that the bubbles can be moved along these predetermined tracks. These tracks can be created in a surface layer of the substrate by ion implantation.

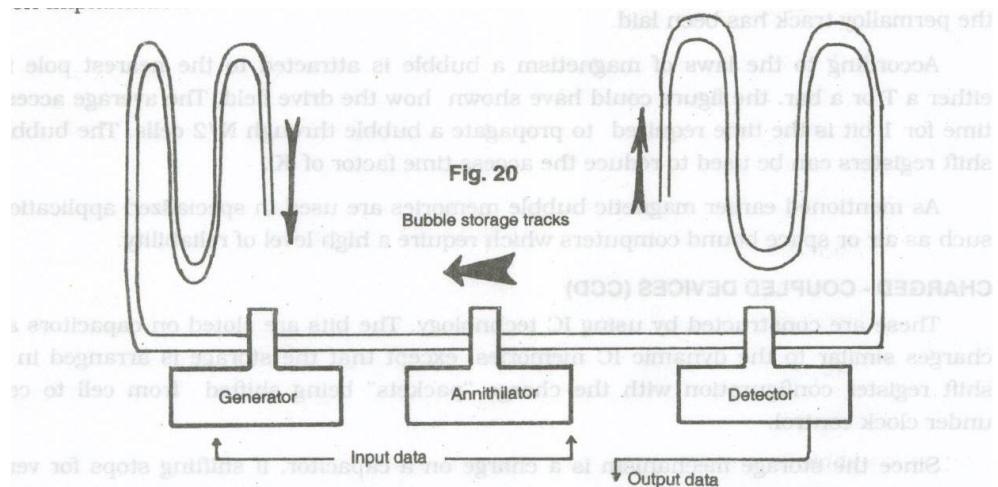


Fig. 20

The figure shows the basic bubble memory organization. It consists of input data (bubbles) and Detector to output data (detect presence of closed tracks are used to circulate the bubble around at a fixed rate. The used to remove bubbles.

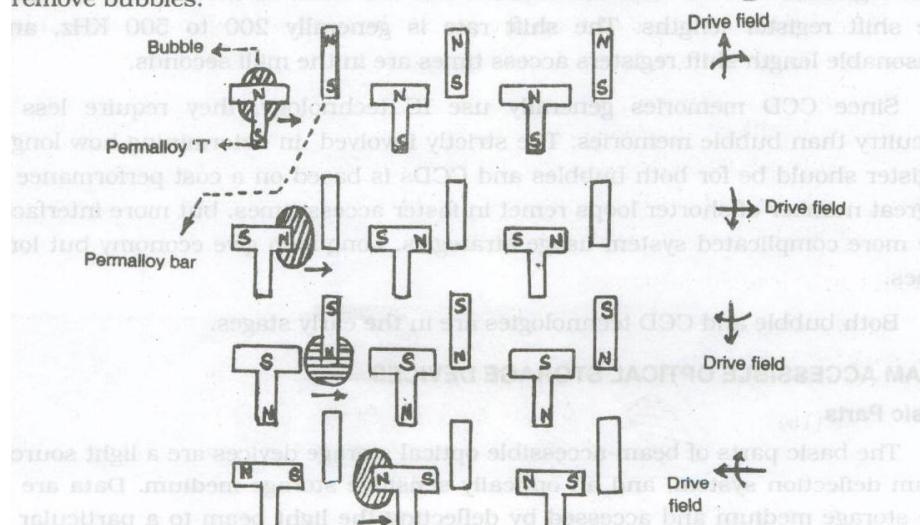


Fig.21 Propagation of Magnetic bubble along a T-Bar Track

The figure 21 shows T-Bar tracks used in early systems. They consist of a T and a bar shaped permalloy element in an alternating sequence. The rotating drive field induces a magnetic field in the permalloy elements parallel to the driver field. Depending on the orientation of the drive field the extremities of the T's and bars and the junctions of the T's become north (N) and south (S) magnetic poles at various times. Each magnetic bubble is like a small magnet, one of whose poles is at the surface on which the permalloy track has been laid.

Fundamentals of Digital Computers

According to the laws of magnetism a bubble is attracted to the nearest pole In either a T or a bar. the figure could have shown how the drive field. The average access time for I bit is the time required to propagate a bubble through N/2 cells. The bubble shift registers can be used to reduce the access time factor of K'.

As mentioned earlier magnetic bubble memories are used in specialized application such as air or space bound computers which require a high level of reliability.

Charged - Coupled Devices (Ccd)

These are constructed by using IC technology. The bits are slotted on capacitors as charges similar to the dynamic IC memories, except that the storage is arranged in a shift register configuration with the charge "packets" being shifted from cell to cell under clock control.

Since the storage mechanism is a charge on a capacitor, if shifting stops for very long (a few mill seconds), the charges will leak from the capacitors and the memory's contents will be lost. CCD memories generally have from 500 KBits to several megebits of storage. The shift registers are read from and written into the ends, so access time is dependent on the shift register lengths. The shift rate is generally 200 to 500 KHz, and so for reasonable length shift registers access times are in the mill seconds.

Since CCD memories generally use IC technology., they require less interface circuitry than bubble memories. The strictly involved in determining how long the shift register should be for both bubbles arid CCDs is based on a cost performance analysis. A great number of shorter loops remet in faster access limes, but more interface circuits are more complicated system usage strategies. Long loop give economy but long access times. Both bubble and CCD technologies are in the early stages.

Beam Accessible Optical Storage Devices

Basic Parts

The basic parts of beam-accessible optical storage devices are a light source, optical beam deflection system, and an optically sensitive storage medium. Data are stored in the storage medium and accessed by deflecting the light beam to a particular point in the storage medium. A laser Is usually used for the light source and the ability to focus laser beams down to very small spots means that high storage capacities are potentially achievable. This range covers that of magnetic and disk and tape storage.

Magnetic-Optic Memory

The basic form is shown below:

The laser beam is first passed into the deflector, one type of deflector is based on the change of optical properties that occur when certain materials, such as nitrobenzlnne are subject to an electric field. This can be used to produce a cell in which a light beam

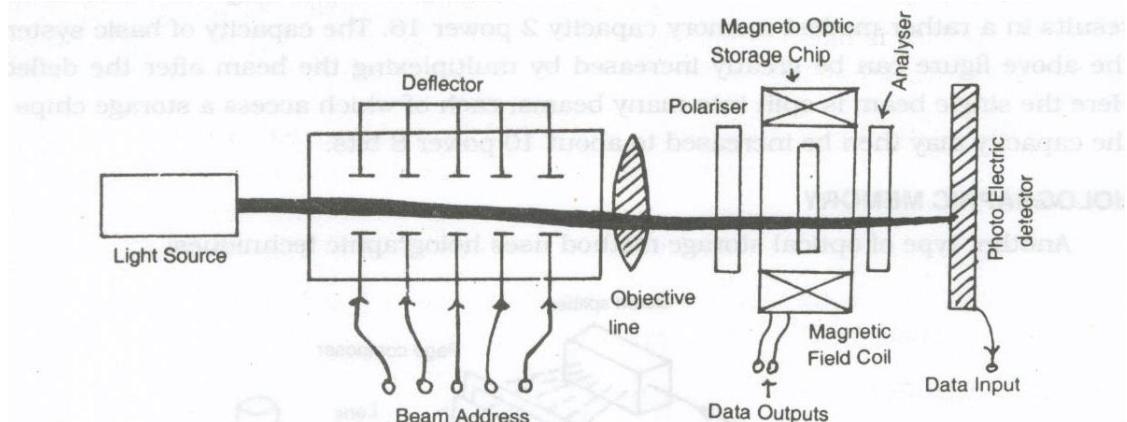


Fig. 22

passes straight through or moved to one side when a voltage is applied to pair of plates. This amount of sideways movement depends on cell dimensions. By cascading N cells the beam can be deflected to any one of 2^N positions, by applying N-bit binary address to the deflection system. Other deflection techniques have also been contemplated such as acousto-optic effects in which the sound waves perturb the refractive index of some materials.

From the deflector the beam is focused onto the magneto-optic storage chip. The chip contains larger number of tiny regions each of which can hold one data bit. When laser beam strikes one of these regions the material is locally heated and the magnetic properties are altered. The writing technique is based on the curie point thermomagnetic property. Here the material, which is ferromagnetic, when raised above the curie temperature by the laser spot, loses its magnetization. The spot is removed in the presence of a magnetic field and the material quickly cools and retains the sense of magnetization that is set by magnetic field. A binary 0 or 1 can therefore be written by choosing the direction of current flow in the magnetic-field coil. To read data it is necessary to detect the direction of magnetization of any selected magnetic cell. A popular technique uses the Faraday effect, in which material magnetization rotates the plane of polarization of a linearly polarized beam of light passes through. By placing the magnetic storage chip between polarizer and analyzer films, the laser beam (of reduced intensity to obtain non destructive readout) is blocked or passes through depending on the direction of magnetization of the addressed memory cell. This is covered to an electrical data signal by a photosensitive area covering the rear of storage chip.

The practical limit for the deflector complexity is something like 16 cells which results in a rather modest memory capacity 2^{16} . The capacity of basic system in the above figure can be greatly increased by multiplexing the beam after the deflector. Here the single beam is split into many beams, each of which access a storage chips and the capacity may then be increased to about 10^8 bits.

Holographic Memory

Another type of optical storage method uses holographic techniques;

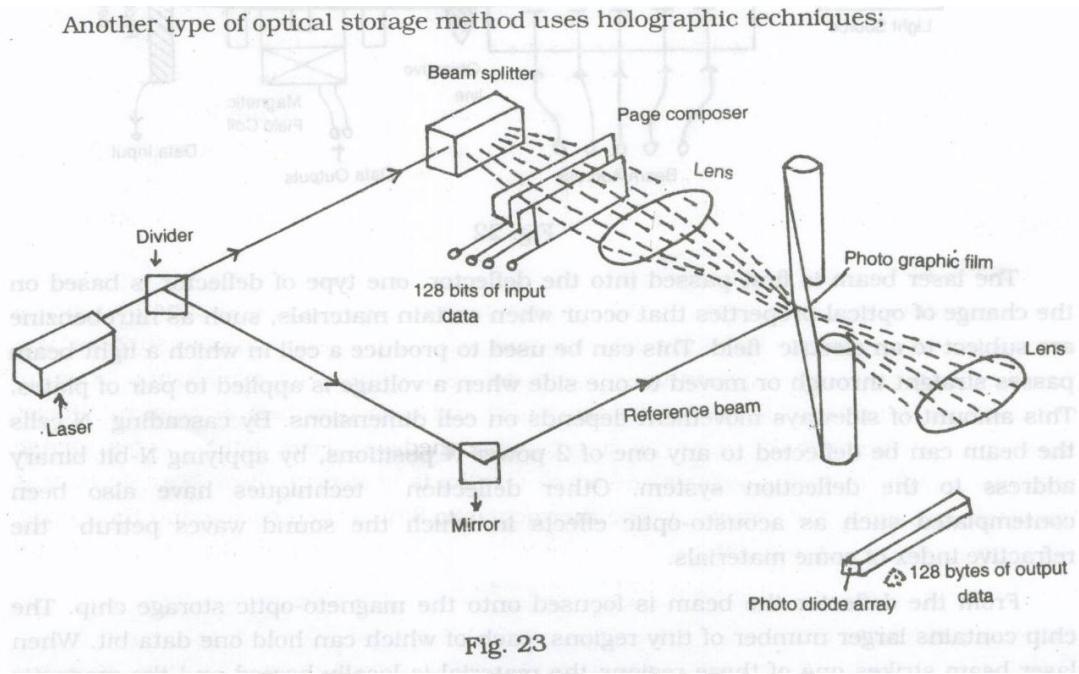


Fig. 23

To write data, the laser beam is separated by the beam splitter into 128 separate beams which pass into the page composer. The page composer assembles the data to be written into a group or ~page, of 128 bits and then uses each bit to open or close an electronic shutter which lies in the path of each beam. The data are then recorded by exposing an area of photographic emulsion on the film. However a holographic image rather than a conventional image is recorded. This is achieved by combining the light from the page composer with a reference beam of coherent light which has been divided from the main laser source. The wave fronts from the two sources of light add or subtract at individual points on the wave fronts and form an interference pattern which is unique to the recorded on the photographic film. The film is moved to another position to record the next page of data.

To read the data the light into the page composer is turned off. One of the remarkable properties of a hologram is that the reference beam then unscans the recorded interface pattern into an image of the original 128 data bits. This image then impinges on an array of 128 photo diodes to provide an electrical output.

Advantages

The main advantages of using holographic techniques are that blemishes on the film can be tolerated and high precision optics are not required. Also precise registration of the hologram is not required to read the data so the file-transport mechanism can be simple design.

Fundamentals of Digital Computers

The total data storage capacity is set by the length of the film. Very high capacities of upto 5×10^{11} bits have been suggested. The average time to access a piece of data depends principally on the speed at which the film can be moved past the reference beam. For storage capacity of 5×10^{11} bits the film is quite long and average access times of a couple of minutes might be expected. When lower capacities comparable to conventional magnetic tape stores are considered then average access times fall to a few seconds and promise to be less than those of magnetic tape stores.

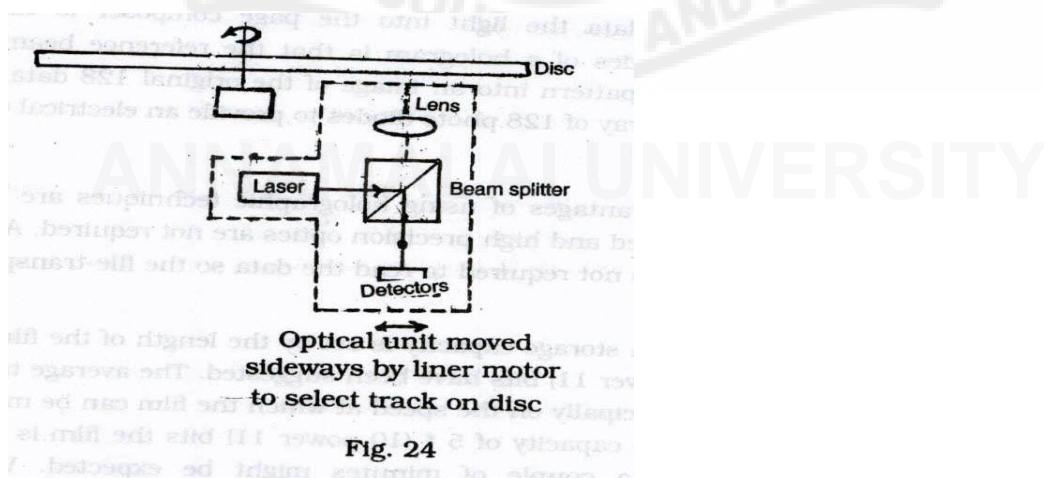
DISADVANTAGES

A possible disadvantage is that photographic film, unlike magnetic tape, is a write-once-only medium. This may not be significant for many applications, such as data logging. However some applications, such as real time data processing require reversible medium that allows data to be erased and then rewritten. Special multilayer thermoplastic film can be used as a reversible medium. Here light causes charge patterns to build up on the surface and when the film is heated these charge patterns ripple the surface of the structure. The thermoplastic film is then cooled and the ripples freeze, thus recording the data. To erase data the required area of film is heated and then cooled. In the absence of light and the film is ready for re-use.

Optical Disc Memory

This type of device has been developed from work on long-playing discs for recorded television programmes, where the inherently high information storage capacity and high signal rates are also desirable features for digital backing stores. One optical disc memory system is outlined in the below figure.

Information is stored on both sides of a removable 12 in (30 cm) disc made up of tellurium films sandwiched between plastic. A single spiral groove on each side runs for 45000 revolutions or tracks. Each track is split into 128 sectors which can hold 1k bits of data (Each sector contains a pre-recorded sector address). This amounts to a total storage capacity on both sides of the disc of 10^{10} bits.



4.2.6 Storage Hierarchy

The design constraints on a computer memory can be summed up by three questions. How much? How fast? and How expensive? There is a trade-off among the three key characteristics of memory namely capacity, access times and cost. The designer would like to use memory technologies that provide for large capacity memory both because the capacity is needed and because the cost per bit is low. However to meet performance requirements, the designer needs to use expensive, relatively lower capacity memories with fast access times.

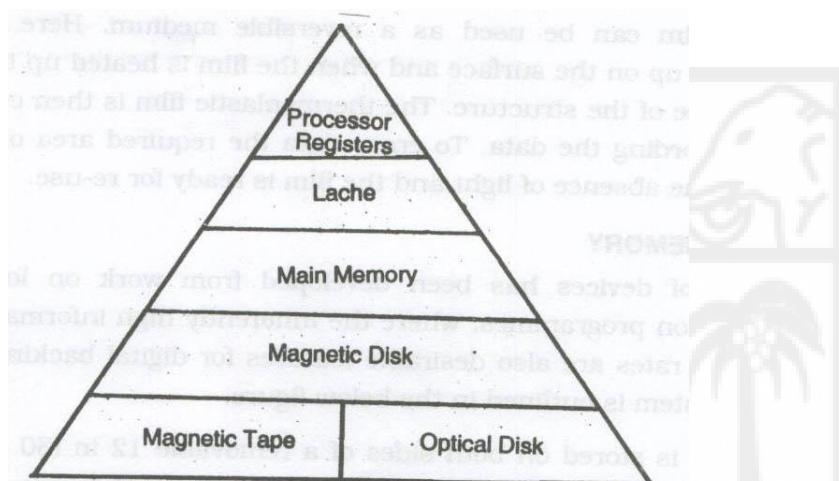


Fig. 25. Memory Hierarchy

The way out of this dilemma is not to rely on a single memory component or technology, but to employ a memory hierarchy. Fig (25) shows this hierarchy. As one goes down the hierarchy, the following occur:

- (a) Decreasing cost/bit
- Magnetic Disk
- Magnetic Tape
- (b) Increasing capacity
- (c) Decreasing access time
- (d) Decreasing frequency of access of the memory by the CPU

Thus smaller, costly, faster memories supplemented by larger, cheaper slower memories. Suppose CPU can access 2 levels of memory. Level 1 contains 1000 words and has an access time of 1 ps. Level 2 contains 100,000 words and has an access time of 10 ps. The CPU first accesses level 1 to find a word. If it is level 1 it is accessed directly. If the word

Fundamentals of Digital Computers

Is In level 2 to word Is transferred from level 2 to level 1 and then It is accessed by the CPU.

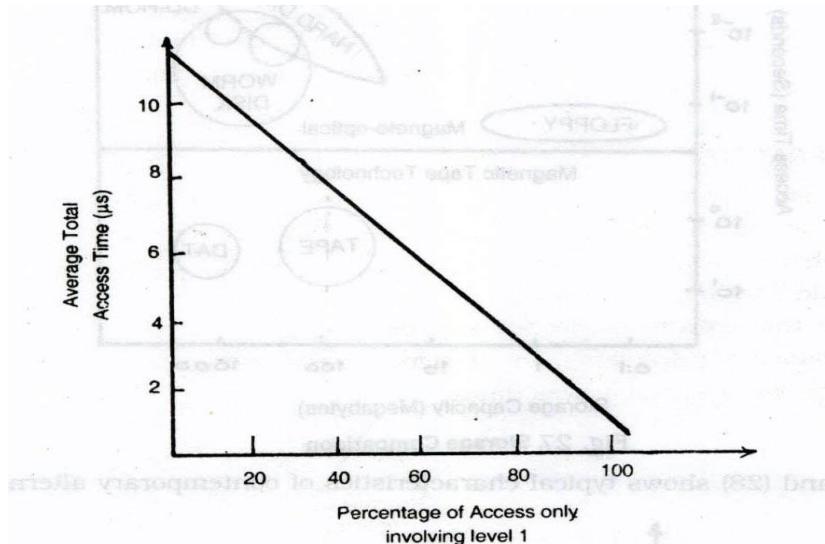


Fig. 26 Performance of a simple two-level memory

1

Fig (26) shows the average total access time Is much closer to that of level 1 than that of level 2.

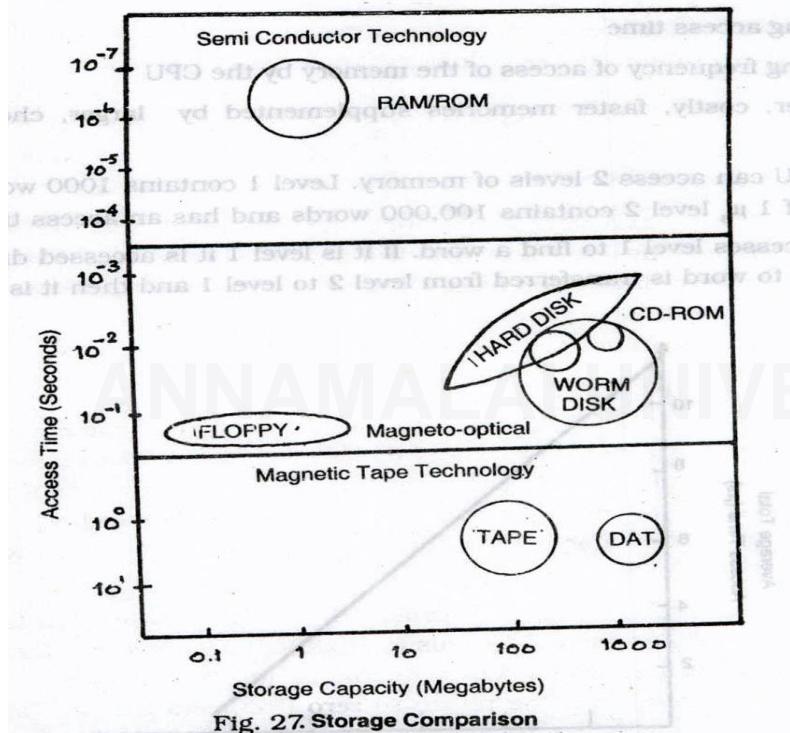


Fig. 27. Storage Comparison

Fig (27)

Fig (27) and (28) shows typical characteristics of contemporary alternative memory system.

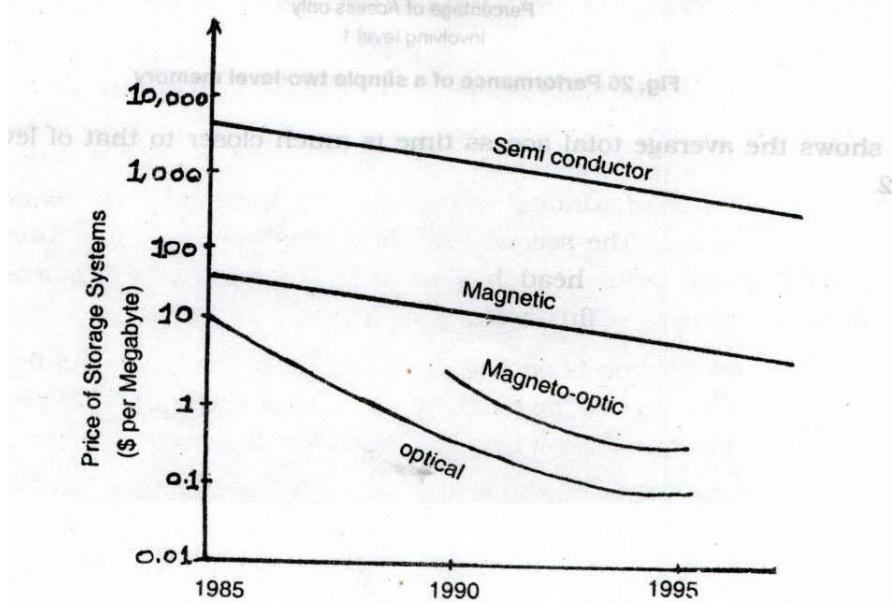


Fig. 28. Cost Forecasts for Secondary Storage Technologies

During the course of execution of program, memory references by the processor. For both Instructions and data, tend to cluster. Programs typically contain a number of iterative loops and subroutines. Once a loop or subroutine is entered, there are repeated references to a small set of Instructions. Over a long period of time the clusters in use change, but over a short period of time, the processor is primarily working with fixed clusters of memory references.

It is possible to organize data across the hierarchy such that the percentage of accesses to each succeedingly lower level is substantially less than to the level above. In a two level memory system most references will be to Instructions and data contained in level 1. This principle can be applied across more than two levels of memory consider the hierarchy shown in Fig (25).

4.2.7 Digital Recording Techniques

Considerable research has gone into both the development of the recorded patterns used to represent 0s and 1s and the means for determining the value recorded. There are two necessities here (1) Packing density should be made as great as it possible that is each cell or bit should occupy as little space as possible, thus reducing the amount of tape or disk are used to store a given amount of information (2) The reading and writing procedure should be made as reliable as possible. In writing information on magnetic surface, the digital information is supplied to the recording circuitry, which then codes this information into a pattern that is recorded by the write head. The techniques used to write information on a magnetic medium can be divided into several categories, the return to

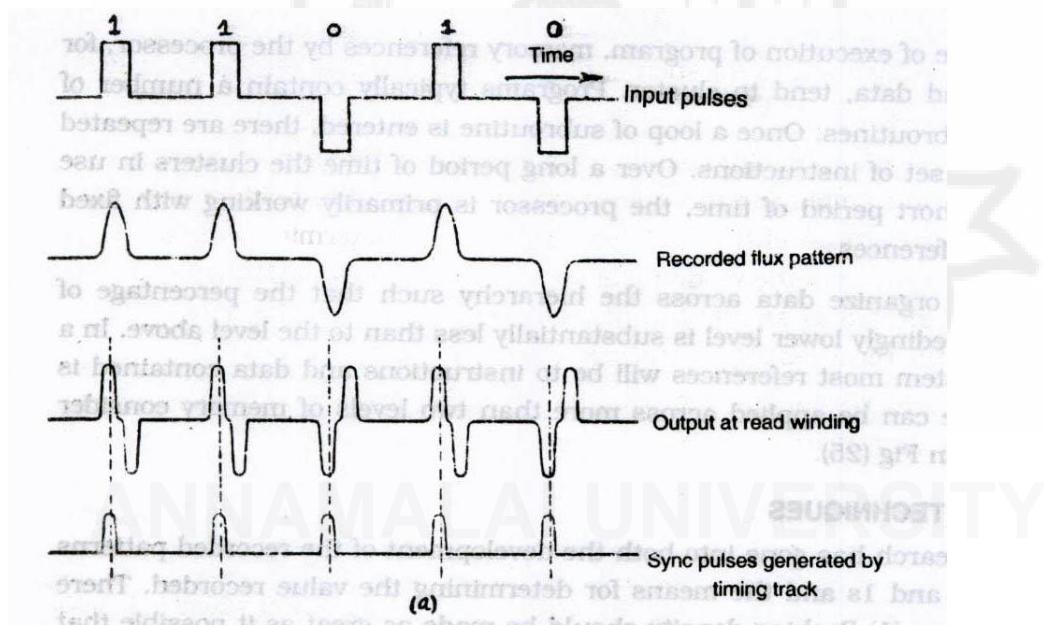
zero (RZ) technique and the return to bias (RB) technique and non return-to-zero (NRZ) technique.

RETURN TO ZERO AND RETURN-TO-BIAS RECORDING TECHNIQUES

Fig (29) illustrates return to zero recording technique. In Fig (29a) current passes through the winding of the write head except when a 1 or a 0 is to be recorded. If a 1 is to be recorded, a pulse of positive polarity is applied to the winding on the write head and if a 0 is to be written, a negative pulse is applied to the winding. In either case the current through the write-head winding is to zero after the pulse and remains there until the next bit is recorded. The second waveform illustrates the flux pattern on the magnetic surface alter the write head has passed. There is some distortion in this pattern because of the bringing of flux around the head.

In this pattern, magnetization is passed under the read head, some of the magnetic flux will be coupled into the core of the head. The flux lower the reluctance path through the core material of the head instead of bridging the gap in the head. (Fig 30)

And when the amount of flux through the core material changes, a voltage will be induced in the coil wound around the core. Thus a change in the amplitude of the recorded magnetic field will result in a voltage being induced in coil of the read head;



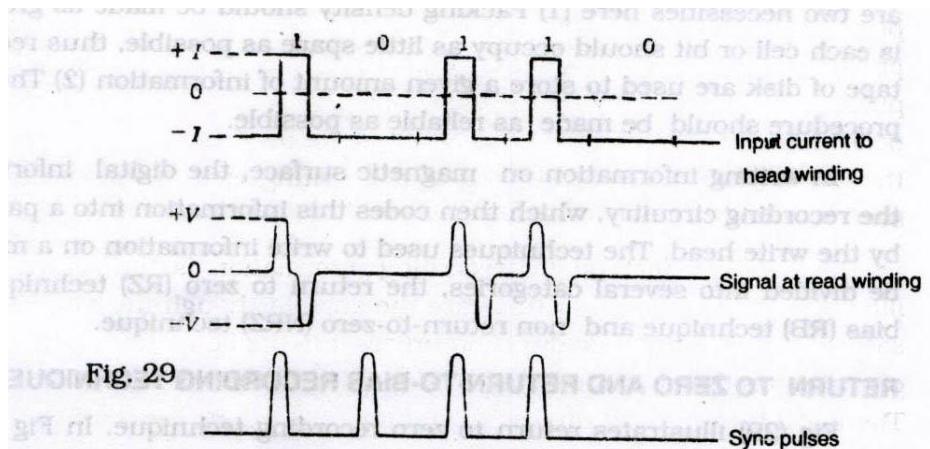


Fig. 29

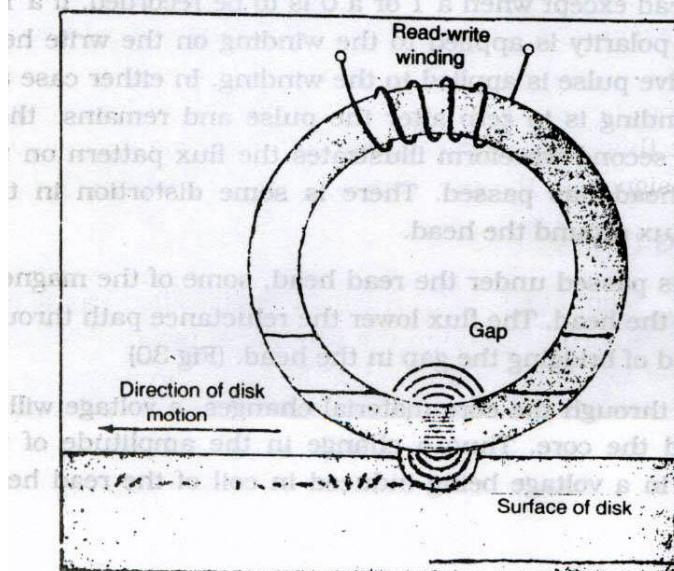


Fig. 30

The waveforms in Fig (29a) and (29b) show typical output signals on the read head winding for each technique.

The problem is therefore to distinguish a 1 or a 0 output at the sense winding. One technique consists of first amplifying the output waveform from the read winding in a linear amplifier. Then the output of this amplifier is strobed to determine whether a 1 or 0 was written. If the output from the read amplifier is connected to an AND gate and the strobe pulse is also connected as an input to the same AND gate, then the output will be a positive pulse when the recorded signal represents a 1.

A fundamental characteristic of HZ recording (Fig (29a)) is: for a 1 the output signal during the first half of each bit time will be positive with regard to the second half; for a 0 the first half of the output signal during each bit time will be negative with regard to the second half of the signal. This fact sometimes exploited in translating the signal read-back. In the HZ system in Fig (29a) the magnetic field returns to zero flux when a 1 or 0

pulse Is not present. This makes It impossible to write over information which has previously been written unless the position of each cell Is very accurately located. If a0 pulse Is written directly over a previously recorded 1. the flux generated will reverse the polarity of the recorded field only if the write head Is In exactly the right position when the 0 Is recorded. So the timing of the writing of imformation is very critical for this system, and It is rarely used.

The second method for recording Information Is the return to the bias system, Illustrated in Fig (29b)). In this case the current though the winding maintains the head saturated In the negative direction unless 1 is the be written. When 1 is written, a pulse of current Is the opposite direction Is applied to the winding at the center of the bit time. There will be output at the sense winding only when 1 Is written. The timing here Is not so critical when information is being written over, because the negative flux from the head will magnetic the surface in the correct direction, regardless of what was previously recorded. The primary problems here concerns sequence of OS. For magnetic tape either a clock track must be used or the code used must be such that at least one 1 occurs line of the tape. Notice that this Is in each because only is generate magnetic flux changes and, therefore, output signals at the read head.

NON RETURN - TO - ZERO RECODING TECHNIQUES

Fig (31) Illustrates 3 recording techniques each of which Is classified as a non return-to-zero system. In the first, the current through the winding is negative through the entire bit time when 0 Is recorded and is positive through the entire time when a1 Is reorded So the current through the winding will remain constant when a sequence of Os or 1s is being written and will change only when 0 Is followed by a1 or when a1 followed by 0 is written. In this case a signal will be induced in the sense winding only when the information recorded changes from 1 to 0 or vice versa.

The second technique Is sometimes referred to as a modified not return-to-zero.or non return-to-zero mark (NRZfl. technique. In this system the polarity of the cun-ent through the write winding is reversed each lime a1 is recorded and remains constant when a0 Is recorded. If a series of 1s is recorded, then the polarity of the recorded flux. will change for each 1- If a series of Os is recorded no changes will occur. Notice that the polarity has no meaning in this system only changes in polarity matters. Therefore a signal will be read back only when a1 has been recorded.

This system is often used for tape recording. When, in order to generate a clock or strobe, a 1 must be recorded somewhere In each cell along the tape width. That Is. If 10 tracks are recorded along the tape, other one of these must be a timing track which records a sequence of 1s, each of which defines a different set of cells to be read; or the information must be coded so that 0 1 occurs in each set of 10 cells which are read. Alphanumeric coded Information is often recorded on tape, and the code may be arranged so that a1 occurs in each code group.

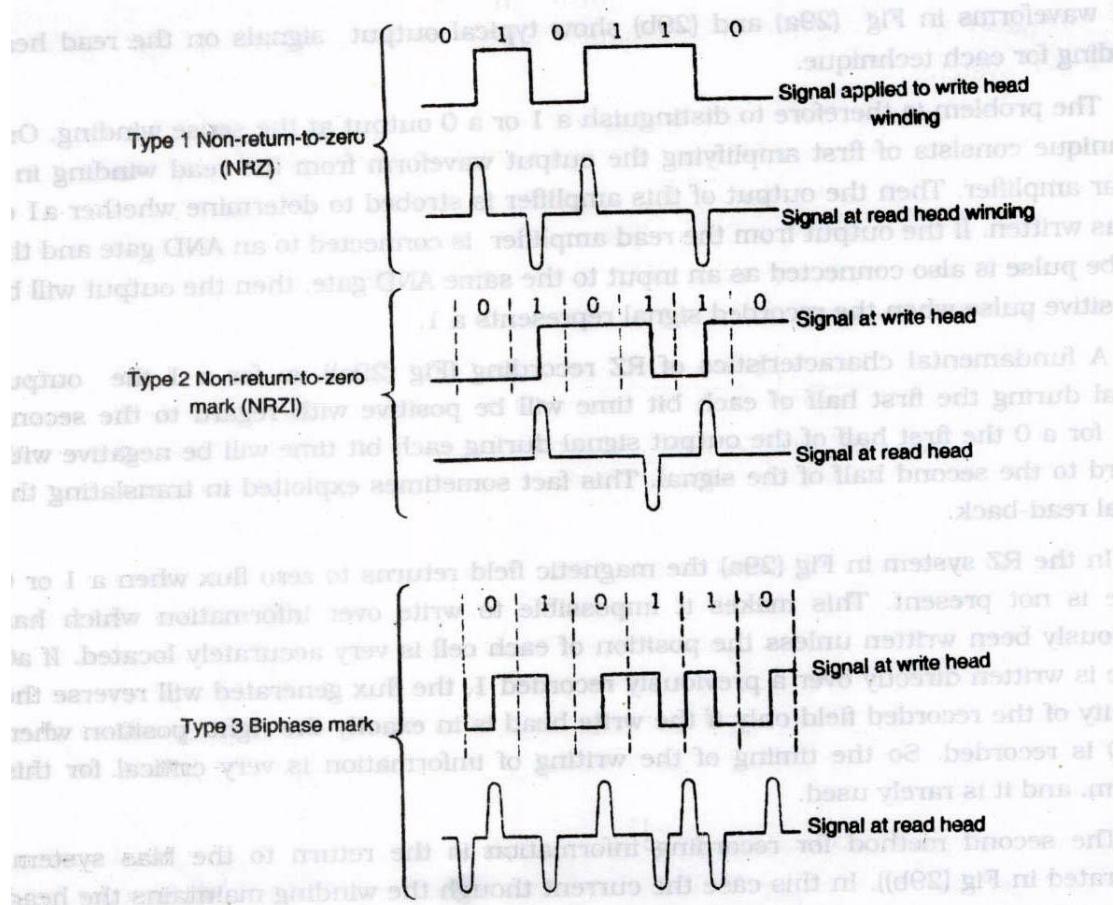


Fig. 31

The third non return-to-zero technique in Fig (31) is sometimes called a phase encoded (or) split frequency system. In this case 0 recorded as a 1/2 bit time negative

pulse followed by a 1/2 bit time positive pulse
 pulse followed by a 1/2 bit time negative
 high-speed systems.

ANAMALAI UNIVERSITY

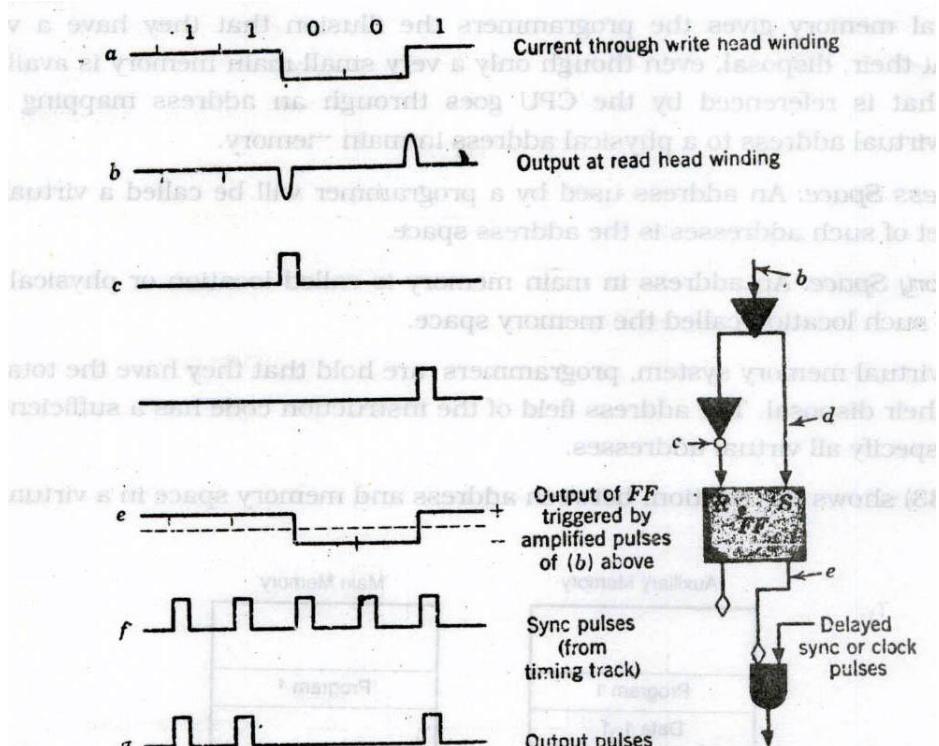


Fig. 32

The reading of information which has been recorded consists of two steps. First the output from the read head is amplified and the amplified signals are translated by logic circuitry. Fig(32) shows a translation technique for the first non return-to-zero system illustrated in Fig (30). The output signals may be either from the output flip-flop or from serial pulses. The sync pulses occur each time a cell passes under the read heads in the system.

The flip-flop (Fig(4)) responds to positive pulses only. Positive pulse signals at the recording head will therefore set the flip-flop to 1. The inverter at the input will cause negative pulses to be made positive. These positive pulses then will clear the flip-flop. The output of the flip-flop may be used directly by the computer or pulse outputs can be generated by connecting an AND gate to the 1 output, delaying the sync pulses, and connecting them to the AND gate. Also, a serial representation of the number stored along the surface may be formed.

4.2.7 Virtual Memory

Normally programs and data are first stored in secondary memory and only operations of program or data are brought into main memory as and when they are needed by the CPU. Virtual memory is a concept used in some large computer systems that permit the user to construct programs as though a large amount of memory space is available.

Fundamentals of Digital Computers

Virtual memory gives the programmers the illusion that they have a very large memory at their disposal, even though only a very small main memory is available. The address that is referenced by the CPU goes through an address mapping from the so-called virtual address to a physical address in main memory.

Address Space: An address used by a programmer will be called a virtual address and the set of such addresses is the address space. **Memory Space:** An address in main memory is called location or physical address. The set of such locations called the memory space. In a virtual memory system, programmers are told that they have the total address space at their disposal. The address field of the instruction code has a sufficient number of bits to specify all virtual addresses.

Fig (33) shows the relation, between address and memory space in a virtual memory system.

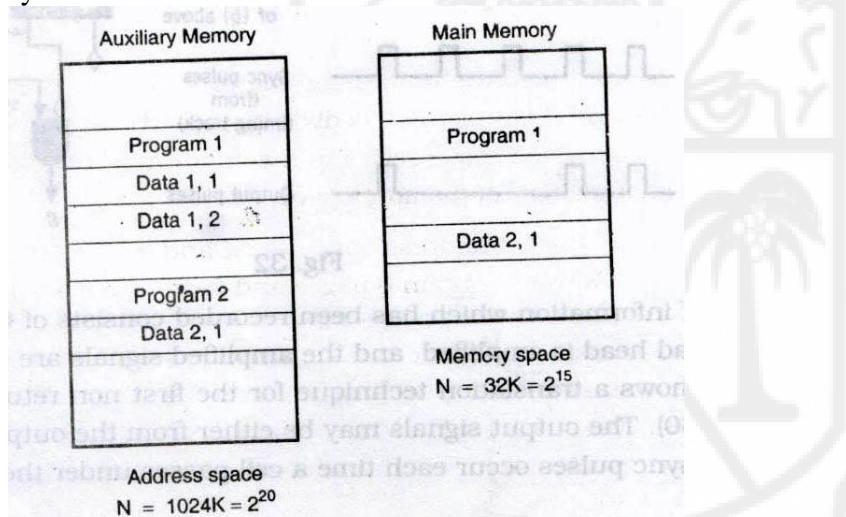


Fig.33 Relation between address and memory space

A table is needed as shown in Fig (34) to map a virtual address of 20 bits into a physical address of 15 bits. The mapping is a dynamic operation, which means that every address is translated immediately as a word is referenced by CPU.

- Program

Data 2,1

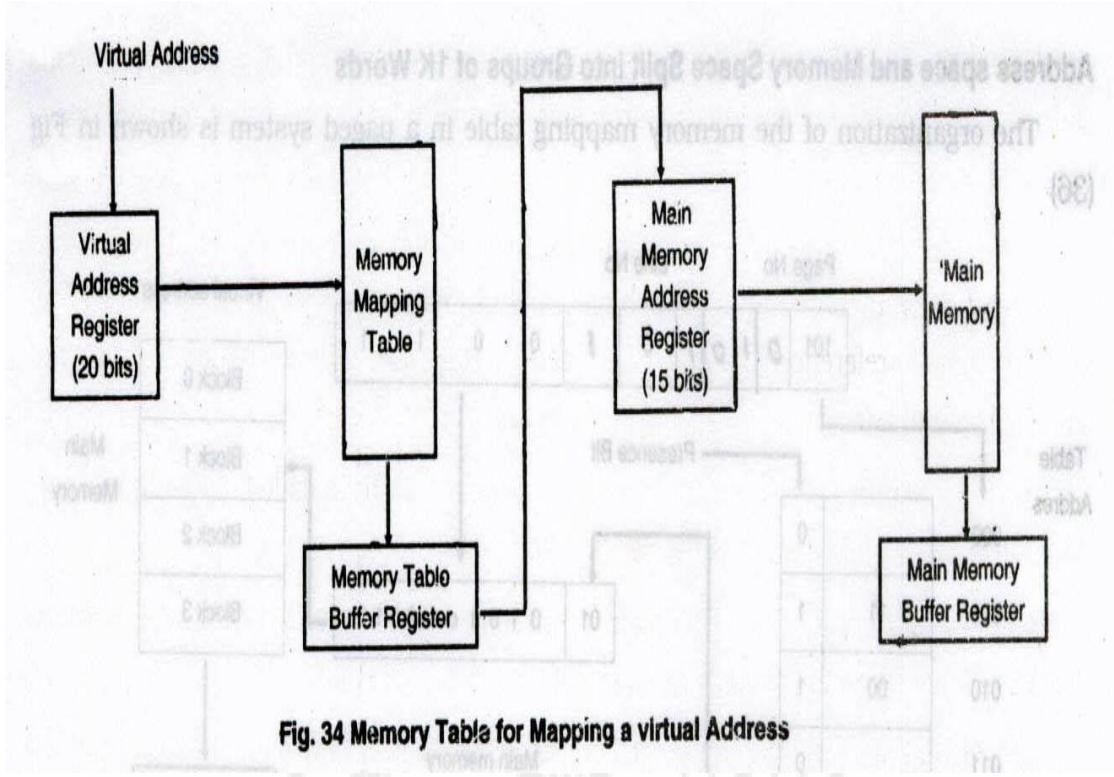


Fig. 34 Memory Table for Mapping a virtual Address

Memory Table for Mapping a Virtual Address

Pages and Blocks: The physical memory is broken down into groups of equal size called blocks, which may range from 64 to 4096 words each. The term page-frame is sometimes used to denote a block.

Page: Page refers to groups then of address space of the same size. For example, if a page or block consists of 1K words then using the previous example, address space is divided into 1024 pages and main memory is divided into 32 blocks. Although the block and the page are of the same size page refers to the organization of address space, while a block refers to the organization of memory space.

Consider a computer with an address space of 8K and a memory space of 4K. If we split each into groups of 1K words obtain 8 pages and four blocks as shown in Fig 35.

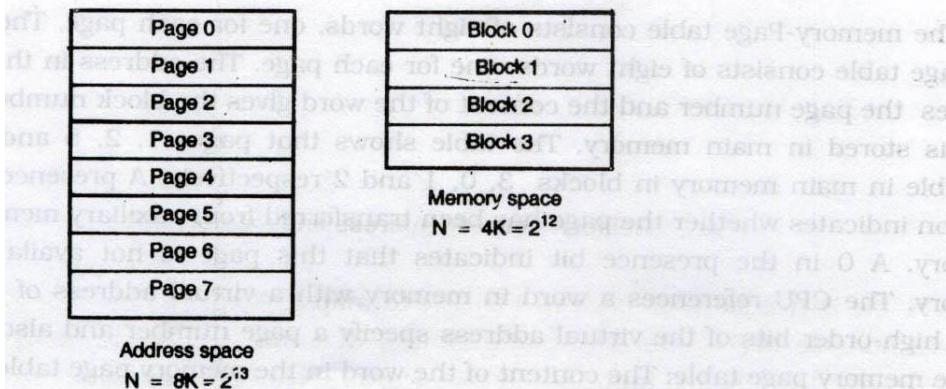


Fig. 35 Address space and Memory space split into groups of 1k words

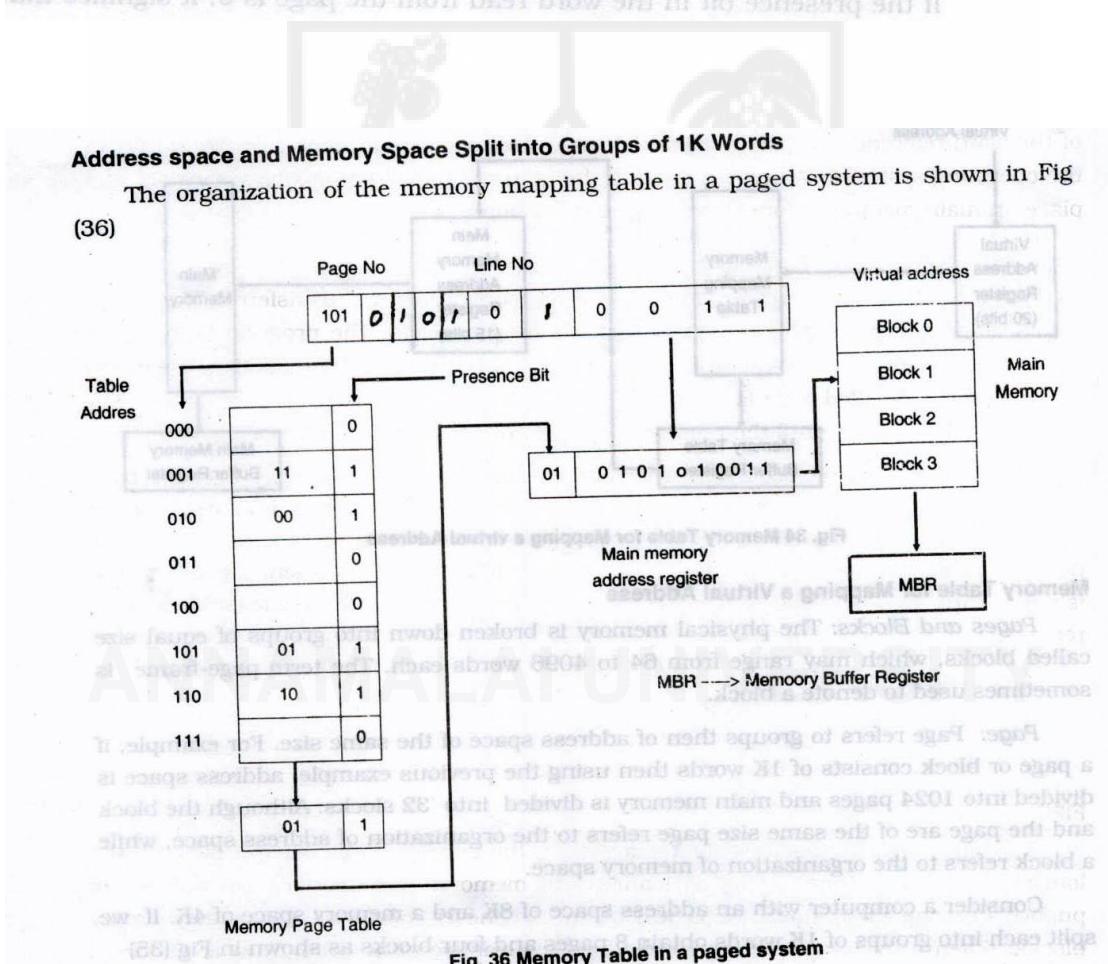


Fig. 36 Memory Table in a paged system

Virtual address

Fig. 36 Memory Table in a paged system The memory-Page table consists of eight words, one for each page. The address in the page table consists of eight words, one for each page. The address in the page table denotes the page number and the content of the word gives the block number where the page is stored in main memory. The table shows that pages 1, 2, 5 and 6 are now available In main memory in blocks 3, 0, 1 and 2 respectively. A presence bit in each location indicates whether the page has been transferred from auxiliary memory to main memory. A 0 in the presence bit indicates that this page is not available in main memory. The Cpu references a word in memory with a virtual address of 13 bits. The three high-order bits of the virtual address specify a page number and also an address for the memory page table. The content of the word in the memory page table at the page number address is read out into the memory table buffer register. If the presence bit is a 1, the block number thus read is transferred to the two high order bits of the main memory address register. The line number from the virtual address is transferred into 10 low order bits of the memory address register. A read signal to main memory transfer the contents of the word to the main memory buffer register ready to be used by the CPU. If the presence bit in the word read from the page is 0, it signifies that the content of the word referenced by the virtual address does not reside In main memory. A call to the operating system Is generated to fetch the required page from auxiliary memory and place In main memory before remaining computation.

Page Replacement

When a program starts execution, one or more pages are transferred into main memory and the page table Is set to Indicate their position. The program Is executed from main memory until it attempts to reference a page that Is still In auxiliary memory. This condition is called page fault. When page fault occurs, the execution of the present program Is suspended until the required page Is brought into main memory.

When a page fault occurs In a virtual memory system. It signifies that the page referenced by the Cpu Is not in main memory. If main memory is full, it would be necessary to remove a page from a memory block to make room for the new page. The policy for choosing pages to remove is determined from the replacement algorithm that is used. The goal of a replacement algorithm Is to try to remove the page least likely to be referenced In the Immediate future.

Two common replacement algorithms are

1. FIFO
2. LRU

FF0 (First In First Out)

The FIFO algorithm selects for replacement, the page that has been in memory the longest time. Each time a page is loaded into memory, the identification number is

pushed into the FIFO stack. FIFO will be full whenever memory has no more empty blocks. When a new page is to be loaded, the page least recently brought in is removed. The page to be removed is easily determined because its Identification number is at the top of the stack.

LRU (Least Recently Used)

LRU policy is more difficult to implement but it is more attractive on the assumption that the least recently used page is a better candidate for removal than the least recently loaded page as in FIFO. The LRU algorithm can be implemented by associating a counter with each page. The counter is set of zero. At fixed intervals of time the counters associated with all pages presently in memory are incremented by 1. The least recently used page is the page with the highest count. The counters are also called tag registers, as their count indicates their tag, that is how long their associated pages have been referenced.

4.2.8 Cache Memory

Reference to memory at any given time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference.

LOCALITY OF REFERENCE

This property states that over a short interval of time the address generated by a typical program refer to a few localized areas of memory repeated while the remainder of memory is accessed relatively infrequently. If the active portions of the program and data are placed in a fast small memory. The average memory access lime can be reduced, thus reducing the total execution time of the program. Such a fast memory is referred to as cache memory.

Basic Operation of Cache

When the Cpu needs to access memory, the cache is examined. If the word is found in the cache, it is said to produce a hit. If the word addressed by the CPu is not found In the cache, it produces a miss and the main memory is accessed to read the word.

HIT RATIO

The ratio of the number of hits divided by the number of misses during a given Interval of time. The basic characteristics of cache memory Is its fast access time. Therefore very little or no time must be wasted while searching for words In the cache. The transformation of data from main memory to cache memory is referred to as a mapping process.

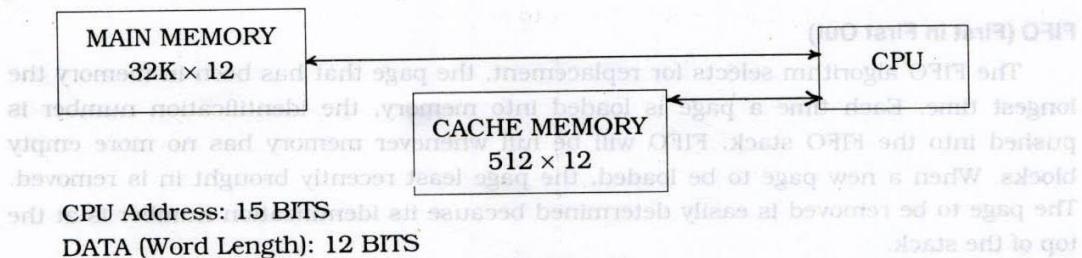


Fig. 37: CACHE MEMORY

Types of Mapping Procedures

1. Associative mapping
2. Direct mapping
3. Set-Associative mapping

ASSOCIATIVE MAPPING

CPU address (15 Bit)

Argument register

| \leftarrow Address \rightarrow | \leftarrow Data \rightarrow |
|------------------------------------|---------------------------------|
| 01000 | 3450 |
| 02777 | 6710 |
| 22345 | 1234 |
| | |
| | |

Fig. 38

The associative memory stores both address and contents of the memory word. A CPU address of 15 bits is placed in the argument register and the associative memory is searched, for a matching address. If the address is found, the corresponding 12 bit data is read and sent to the CPU. If no match, the main memory is accessed for the word. The address-data pair is then transferred to the associative cache memory. If the cache is full, an address data pair must be displaced to make room for a pair that is needed and not presently in the cache.

Direct Mapping

The CPU address of 15 bits is divided into 2 fields. The nine least significant bits constitute the Index field and the remaining six bits form the tag field. The number of bits in the Index is equal to the number of address bits required to access the cache memory.

In the general case

There are 2 words In cache memory

and 2 words in main memory

The n bit memory address is divided into 2 fields k bits for the Index field and n – k bits for the tag field. The direct mapping cache organization uses the n-bit address to access the main memory and the k-bit index to access the cache.

Fundamentals of Digital Computers

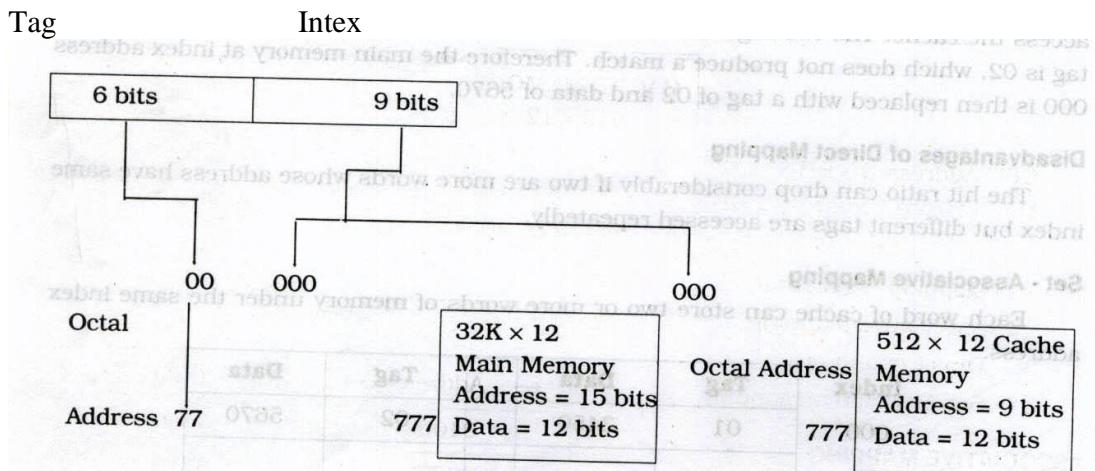


Fig. 39 DIRECT MAPPING

Internal Organization of the Words in the Cache Memory

Each word in cache consists of the data word and its associated tag.

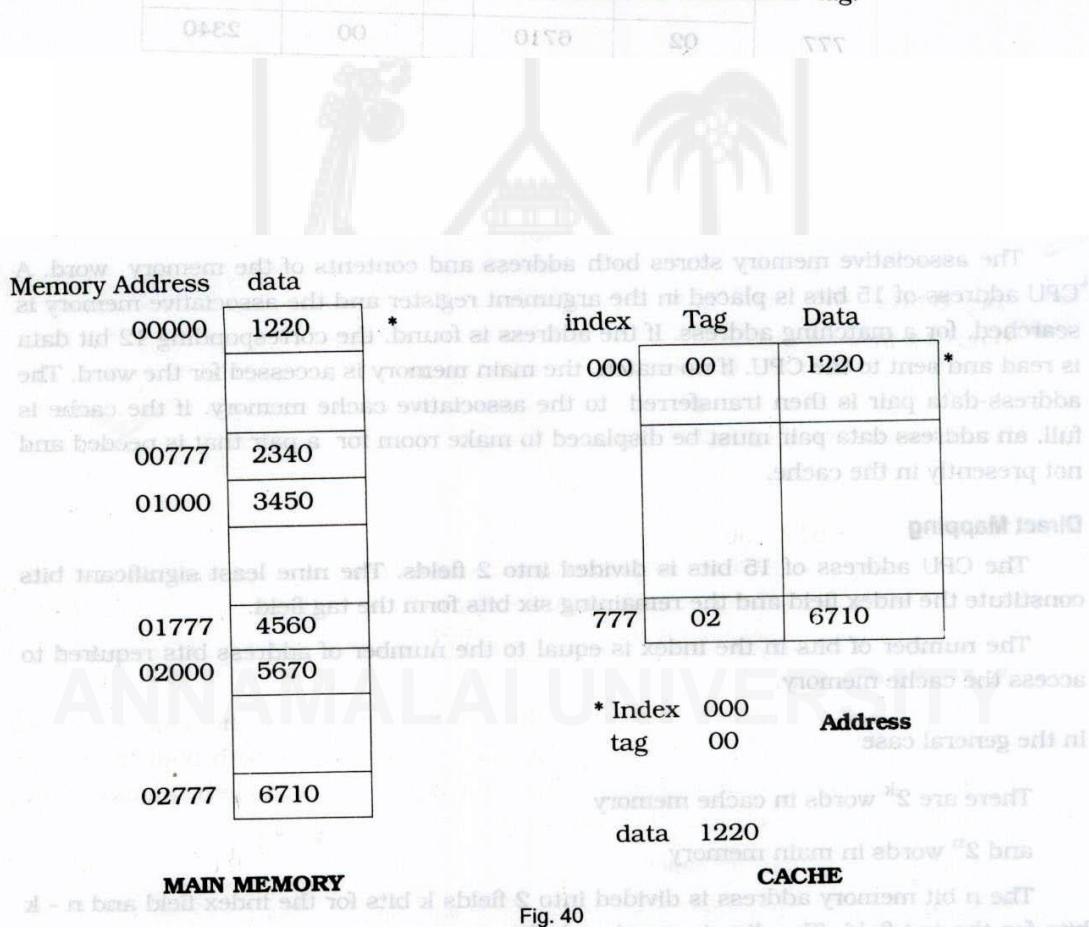


Fig. 40

The word at address zero is presently stored in the cache. Suppose that the CPU now wants to access the word at address 02000. The index address 000 is used to access the cache. The two tags are now compared. The cache tag is 00, but the address tag is 02.

which does not produce a match. Therefore the main memory at index address 000 is then replaced with a tag of 02 and data of 5670.

Disadvantages of Direct Mapping

The hit ratio can drop considerably if two or more words whose address have same index but different tags are accessed repeatedly.

Set - Associative Mapping

Each word of cache can store two or more words of memory under the same index address

| Index | Tag | Data | Tag | Data |
|--------------|------------|-------------|------------|-------------|
| 000 | 01 | 3450 | 02 | 5670 |
| | | | | |
| 777 | 02 | 6710 | 00 | 2340 |

Tag: 6 bits
Data: 12 bits

Index: 9 bits
Word length = 2 (6+12)

= 36 bits.

Size of Cache = $512 \times 36 = 1024$ words

TWO WAY SET-ASSOCIATIVE MAPPING CACHE

Each Index address refers to two data words and their associated tags. In general a set associative cache of set size K will accommodate k words of main memory In each word of cache. When the CPU generates a memory request the Index value of the address is used to access the cache. The tag field of the CPU address Is compared with both the tags In the cache to determine If a match occurs. The comparison logic is done by an associative search of the tags In the set; thus the name "set associative" advantage - hit ration increases disadvantage - requires more complex comparison logic.

REPLACEMENT ALGORITHMS

- o Random replacement
- o FIFO
- o LRU

WRITING INTO CACHE

Write Through Update main memory with every memory write operation with cache memory being updated in parallel if it contains the word at the specified address. Ensures data validity In systems with direct memory access transfers.

WRITE BACK

Only the cache location Is updated during the write operation. The location is then marked by a flag so that jater when the word is removed from cache It Is copied into main memory.

- a A word may be updated several times during its stay in cache.

ANAMALAI UNIVERSITY

4.3) Revision Points

Internal Processor Memory

This comprises a small set of high speed registers used as working memory for temporary storage of instructions and data.

Main Memory

Also called as primary memory is a relatively fast memory used for program and data storage during computer operations. The principal technology used for main memory is based on semiconductor integrated circuits (ICs).

Bandwidth(bM)

It is generally convenient to assume at t_M is the time needed to complete any read or write operation. Hence the maximum amount of information that can be transferred to and from the memory every second is $1/t_M$. This quantity is called the data transfer rate or bandwidth b_M .

Random Access Memories

Random-Access Memories (RAMs) are characterized by the fact that every location can be accessed independently. The access and cycle times for every location are constant and independent of its position

4.4) Intext Questions

1. Discuss about Random access memories.
2. Explain in detail about how to connect connecting memory chips to computer bus.
3. Give brief detail about Storage hierarchies.
4. Discuss about static and dynamic ROM,RAM and EPROM memories.
5. Give detail about CCD and optical storage devices.

4.5) Summary

- The event of an opposing magnetic field being applied perpendicular to the surface plate, opposite to the direction of magnetization of the domains, these domains tend to curl up and form cylindrical domains called MAGNETIC BUBBLE
- Address Space: An address used by a programmer will be called a virtual address and the set of such addresses is the address space.
- Memory Space: An address in main memory is called location or physical address. The set of such location called the memory space.
- Reference to memory at any given time tend to be confined within a few localized areas in memory. This phenomenon is known as the property of locality of reference.

4.6) Terminal Exercises

1. What is RAM?
2. Define decoders.
3. Give short notes about storage hierarchies.
4. Write about digital recording techniques.
5. What is hit ratio?
6. Give hints about hard disk and floppy disk.
7. What is the function of magnetic bubble?

4.7) Supplementary Materials

Multi-carrier Digital Communications: Theory and Applications of OFDM - ARS Bahai, BR Saltzberg, M Ergen

4.8) Assignments

Prepare assignment about virtual and cache memory.

4.9) Reference Books

M.M.Manoo, ‘Computer System Architecture’, Prentice Hall, 1982.

4.10) Learning Activities

An individual or group of peoples goes to library for future evaluation of this unit.

4.11) Keywords

- **Random accesses memories**
- **Decoders**
- **Mass storage’s**
- **Magnetic tape**
- **Magnetic bubble**
- **Optical storage devices**
- **Storage hierarchies**

UNIT - V

5.0) Introduction

The components of a computer system, that is the memories, input – output devices etc. must be interconnected to form a computer system. The way these components are put together and how they communicate with each other profoundly affect the system's performance characteristics.

5.1) Objective

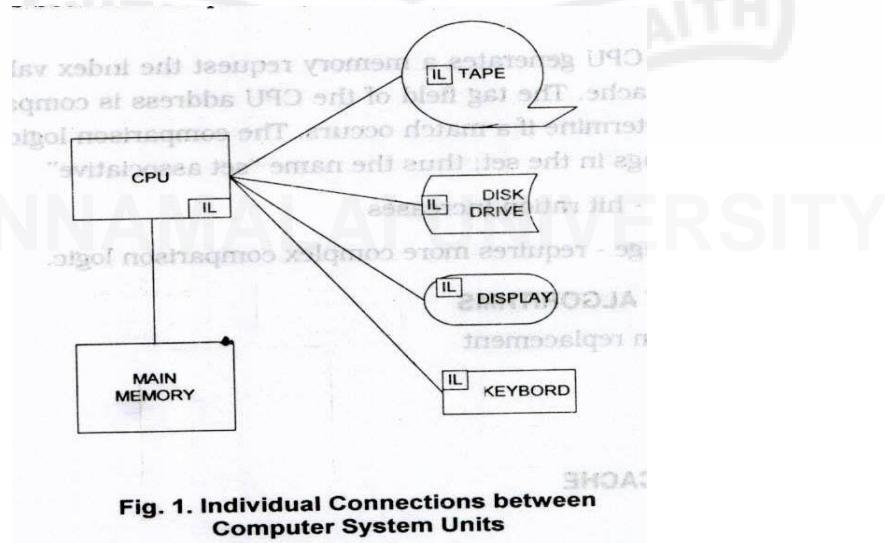
At the end of this unit student will get knowledge about the following topics:

- I/O Busses and Interfaces
- Keyboard
- Character recognition
- Output devices
- CRT
- Flat panel display
- Printers

5.2) Content

5.2.1 Buses And Interfaces

There are several ways to interconnect components in a computer, but the most used is a bus. To make interconnection of the system component less experience and to standardize the interface logic used a very popular techniques involves interconnecting all components by using a signal bus. Fig (1) shows individual connections between computer system units.



Fundamentals of Digital Computers

Fig (2) shows a unibus system. Note that the same wires are used to transfer data from the CPU to the main memory as from the CPU to a tape or other input-output device.

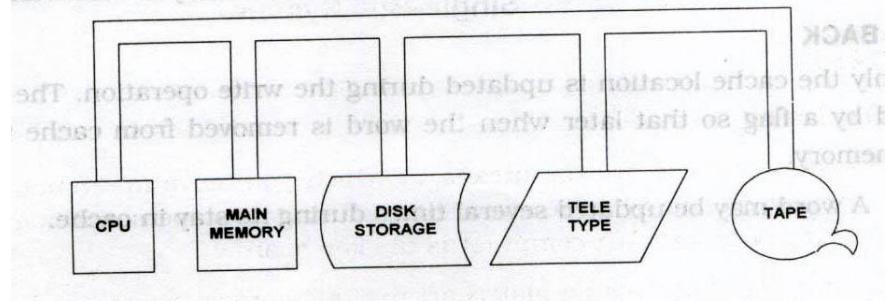


Fig. 2 All Devices are Connected Through a Single Bus

When one device or unit is connected to another, an interface is required which includes the memory logic.

A widely used technique to interface modules efficiently at low cost employs a single bus to interconnect all the units. This is shown in Fig (3), where the several lines or conductors which form the bus pass through and connect to a number of units or modules. In general each module can read from or write on to the bus. The bus interface is usually standardized since the same bus connects all units. Since each unit connects only once to the bus, the amount of interface circuitry and logic required tends to be lower than for separate connection between units. As a result, buses are widely used in microcomputers and minicomputers and even in large computer system for modules where the data flow is not excessive. Often the modules which are bussed together share the same data lines. Then it is necessary for each module to be able to both write into and read from a given line.

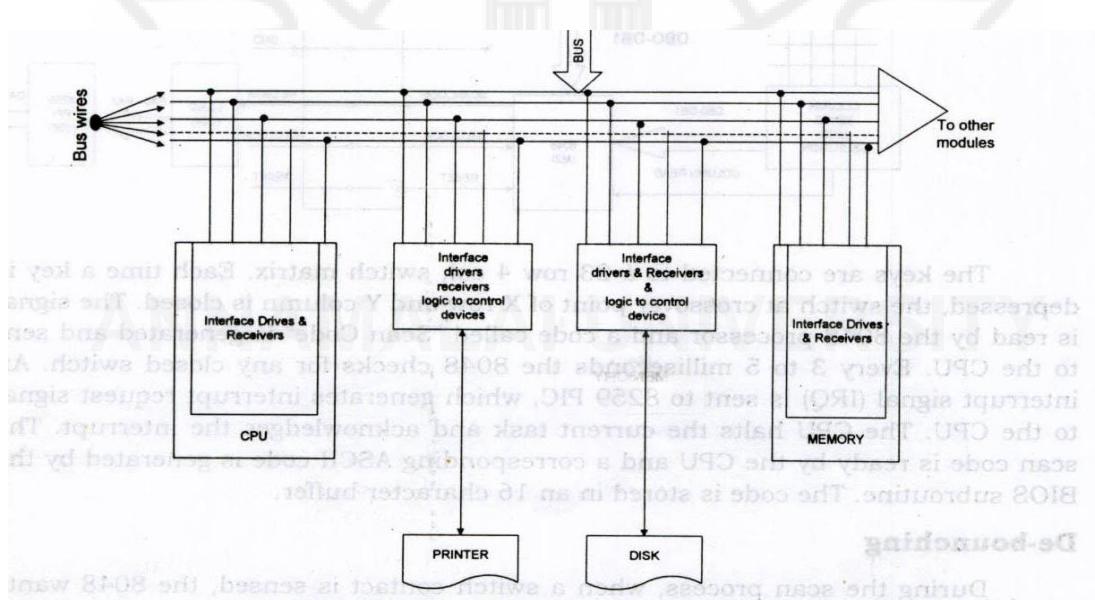
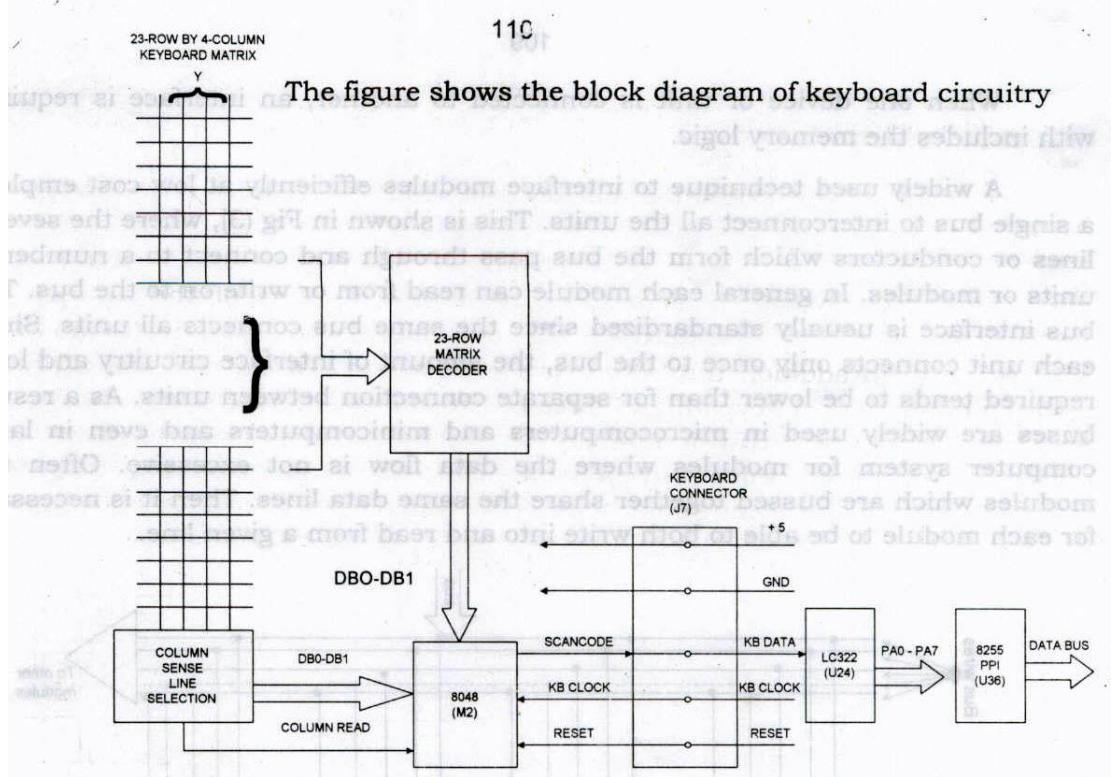


FIG. 3. Computer Organization of Single-Bus System

5.2.2 Keyboard

Input Devices

The input devices are the means by which you move information into your PC, the primary means by which you interact with your personal computer. The primary input device for most computer is the key board. The key board contains a matrix array of momentary contact switches and 8-bit 8048 single chip computer with an internal 2KB ROM and associated electronic *to handle control tasks.



The keys are connected to a 23 row 4 col, switch matrix. Each time a key is depressed, the switch at crossover point of X row and Y column is closed. The signal is read by the 8048 processor and a code called "Scan Code" is generated and sent to the CPU. Every 3 to 5 milliseconds the 8048 checks for any closed switch. An interrupt signal (IRQ) is sent to 8259 PIC, which generates interrupt request signal to the CPU. The CPU halts the current task and acknowledges the interrupt. The scan code is ready by the CPU and a corresponding ASCII code is generated by the BIOS subroutine. The code is stored in an 16 character buffer.

De-bouncing

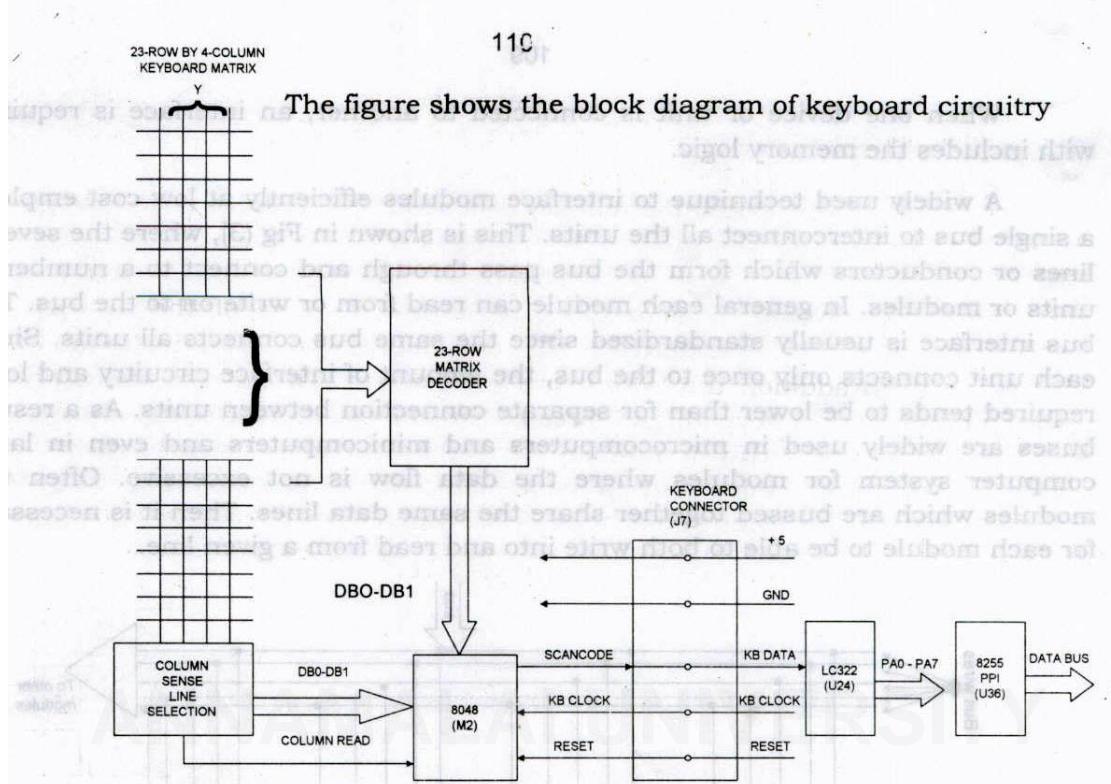
During the scan process, when a switch contact is sensed, the 8048 wants for a few milliseconds to let the key closure settle out. This is because the mechanical switches

such as keys don't close neatly i.e. they bounce several times before a solid contact is achieved. This bouncing can produce noise spikes that can be interpreted as noise signals. To counter this the 8048 provides a short (several milliseconds) delay before the key action is encoded and an interrupt is generated.

Scan Codes

When a key stroke is generated, a microprocessor built into the keyboard generates a scan code indicating which key is pressed. The scan code is converted to serial data and relayed to the microprocessor in the computer's system unit.

Each press of a key generates two different scan codes - one when the key is pushed down, and another when it is pushed back up. The two code technique allows your computer system, unit to tell when a key is pressed and held down-for example, when you hold down the ALT key while pressing a function key.



The keys are connected to a 23 row 4 col, switch matrix. Each time a key is depressed, the switch at crossover point of X row and Y column is closed. The signal is read by the 8048 processor and a code called "Scan Code" is generated and sent to the CPU. Every 3 to 5 milliseconds the 8048 checks for any closed switch. An interrupt signal (IRQ) is sent to 8259 PIC, which generates interrupt request signal to the CPU. The CPU halts the current task and acknowledges the interrupt. The scan code is ready by the CPU and a

Fundamentals of Digital Computers

corresponding ASCII code is generated by the BIOS subroutine. The code is stored in an 16 character buffer.

De-bouncing

During the scan process, when a switch contact is sensed, the 8048 wants for a few milliseconds to let the key closure settle out. This is because the mechanical switches such as keys don't close neatly i.e. they bounce several times before a solid contact is achieved. This bouncing can produce noise spikes that can be interpreted as noise signals. To counter this the 8048 provides a short (several milliseconds) delay before the key action is encoded and an interrupt is generated.

Scan Codes

When a key stroke is generated, a microprocessor built into the keyboard generates a scan code indicating which key is pressed. The scan code is converted to serial data and relayed to the microprocessor in the counter's system unit. Each press of a key generates two different scan codes - one when the key is pushed down, and another when it is pushed back up. The two code technique allows your computer system, unit to tell when a key is pressed and hold down-for example, when you hold down the ALT key while pressing a function key.

Each key also generate a unique scan code. Even if the same legend appears on two keys, such as duplicate number keys in the alphanumeric and numeric and cursor keypad, the individual keys generate the same codes. The code for a given key Is whether the Caps Lock or the shift key is in effect.

Key Layouts

The first keyboard was a 83 keys keyboard. Most of the arrangement followed the typewriting standard - a big set of alphabetical keys in the middle of the keyboard. In addition there were two rows of function keys at the left of the main alpha-numerical keypad, arid force cursor controls to share the same keypad with a calculator-style array of numbers for direct data entry. The enter key is small and ambiguously identified with a bent arrow legend, and no indicators were provided for the three locking shift keys (Caps, Num Lock, Scroll Lock).

The original 83-Keys keyboard layout is given.

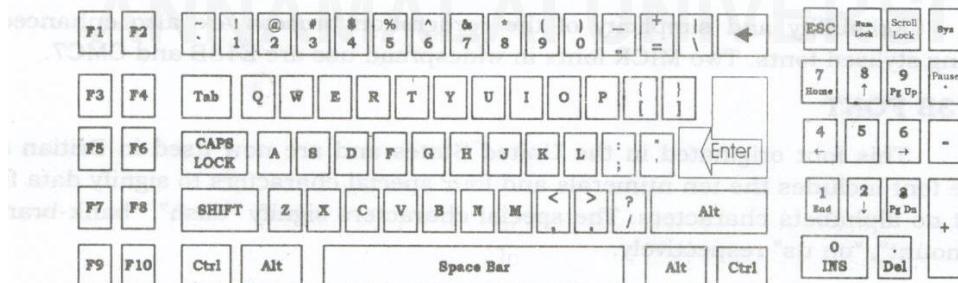


Figure 1. Original IBM AT Keyboard Layout

Fundamentals of Digital Computers

The keyboard released for the AT was the 101 keyboard. It was electrically similar to the original AT keyboard (84 keys). Here the layout was changed and several key additions were made. A new dedicated cursor pad was provided and several other functional keys were duplicated separately, two new function keys were provided, and the position of the function keys were changed.

The layout of a 101 keys Keyboard is given below:

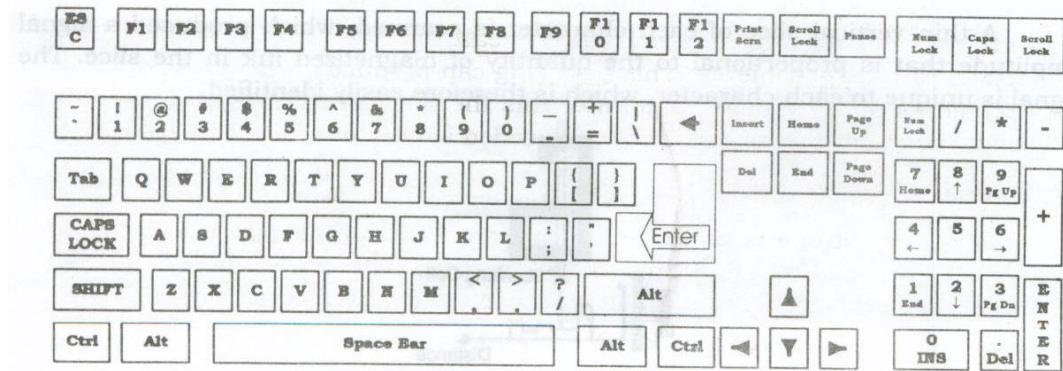


Figure 2. IBM Advanced Keyboard Layout

5.2.3 Character recognition

Magnetic Ink Character Reader

Characters are printed using ink loaded with iron oxide. The printed document is passed through the reading machine which first magnetizes the characters and then passes the document under a small coil. The magnetized characters induce a voltage signal in the coil from which the character is decoded. The term Magnetic Ink Character Reader (MICR) is used as well as magnetic character.

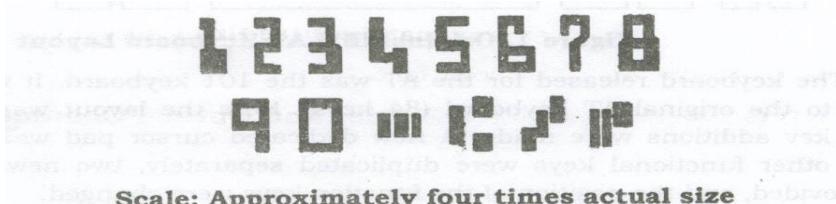
A special printer is used to print the characters on document. This method has found widespread use in banks (for example, on cheques) because of its security and reliability. The method is secure because characters are not easily altered without the special printer. Reliability is obtained because ordinary dirty marks and other blemishes which might cause errors in an optical character reader are less likely to cause errors here because they are generally non-magnetic.

Reliability and simplicity of the recognition process are also enhanced by using stylised fonts. Two MICR fonts in widespread use are E13B and CMC7.

E13B FONT

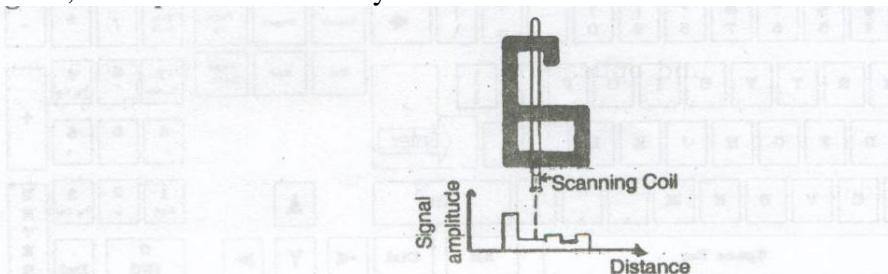
This font originated in the United States and is now used in Britain also. The font includes the ten numerals and four special characters to signify data fields but no alphabets characters. The special characters signify "dash", "bank-branch", "amount", "on us" respectively.

Fundamentals of Digital Computers



The E13B Magnetic Character Font

A thin vertical slice of each character is scanned, which produced a signal amplitude that is proportional to the quantity of magnetized ink in the slice. The signal is unique to each character, which is therefore easily identified.



Reading of E13B character.

CMC7 FONT

This font is in widespread use on the continent. Numerals, special characters and also alphabet characters are included. The coding principle is different from that of the E138 font. In this case each character is made up of seven vertical magnetic ink bars. The six spaces between the bars are binary valued: narrow (binary 0) and wide (binary 1). This creates a six-bit code which allows a maximum of 64 possible characters to be represented. CMC7 uses 41 of these. The magnetic character reader again scans a thin vertical slice of the character, but measures the time between the bars and from this the character is decoded. The method is therefore less sensitive to the length of each bar, (provide each bar is not too short). This allows pieces to be cut out of the bars to allow the characters to be visually readable.



(a)

Scale: approximately three times actual size

The CMC7 magnetic character font.

OMR AND OCR: (Optical Mark Reader and Optical Character Reader)

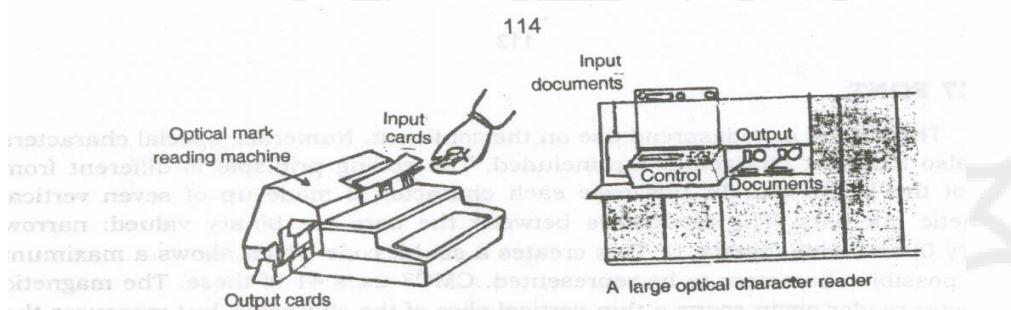
Introduction

The common way to enter data into a computer is by means of a keyboard. This process is a very lengthy process and prone to error. To overcome this problem, we can read the data through machines. Better solution can be to enter the data at source by means of a mark or character reading machine. This concept is called as source-data automation.

OMR AND OCR

Optical mark readers and optical character readers accept printed or hand written information directly from documents and output this information as data in computer-compatible form. OMR can detect pencil or ink mark made on special paper forms using light sensors. The special forms are designed with square circle marks which can be filled with soft pencil or ink.

OCR is capable of reading alphabetic and numeric characters printed on paper. The characters are of special design which are typed using a special font known as OCR font. OCR can read several thousands of printed characters per second.



Types of Marks and Characters that Can be Read

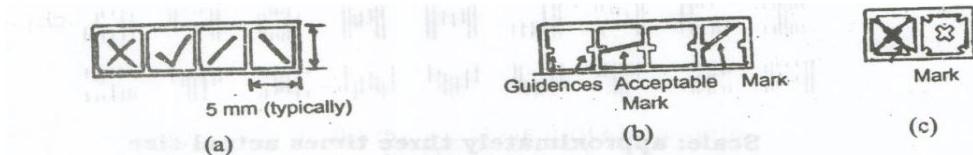
1. Location

Readers are generally designed to read marks or characters positioned in a matrix pattern formed by regularly spaced locations on each of several rows. This pattern is obtained automatically when a printer or typewriter is used.

2. Marks

Optical Mark readers detect the presence or absence of a mark placed in each permitted location, some machines will read a wide variety of marks.

Fundamentals of Digital Computers



Typical Marks that May be Used a. Unrestricted; b. Must fit guidelines

(iii) Printed or typed characters

(iii) Printed or typed characters
Various machine readable characters sets, or fonts are used. Two standard fonts OCR-A and OCR-B. Printers and typewriters are available with OCR-A and OCR-B fonts

(iv) Hand-Printed Characters

Because of difference in style of individuals, reliable recognition of hand-printed characters is more difficult than printed or typed characters. Consequently many OCR machines will read hand-printed numerals only, together with a few other characters.

Please print like this:

1 2 3 4 5 6 7 8 9 0

Various Types of OCR Machines

(i) Journal Roll Reader

These type of machines are able to read journal rolls (or tally rolls) that record transactions in cash register or adding machines.

(II) Document Readers

These can handle documents up to about 15 by 25 cm. Typical application include processing of membership subscription forms, bank documents and bills.

(iii) Pager Reader

Typically these can read pages up to about 25 by 35 cm. These machines are tend to be larger and more versatile.

APPLICATIONS

Optical Mark Readers and Optical Characters Readers can be used for a wide range of applications as an alternative to keying-in data. Optical character reading machines are usually expensive, so tend to be used for application which generate large quantities of data. OMR technique is suitable for areas where one out of small number of choices is to be marked. Eg: Objective type answer sheets, market survey, population survey etc where responses can be restricted to one out of a few alternatives. OMR are more flexible than others. An important concept that becomes possible with OMR and OCR is that of a turn-round document recipient.

Eg: in subscriber-renewal forms, stocks control warehouse etc.

5.2.4 CRT

The CRT which is commonly known as the picture tube, is a large glass envelope containing an electron gun. The basic elements of the conventional CRT are the filament, cathode grids and the anode are shown in the following Figure.

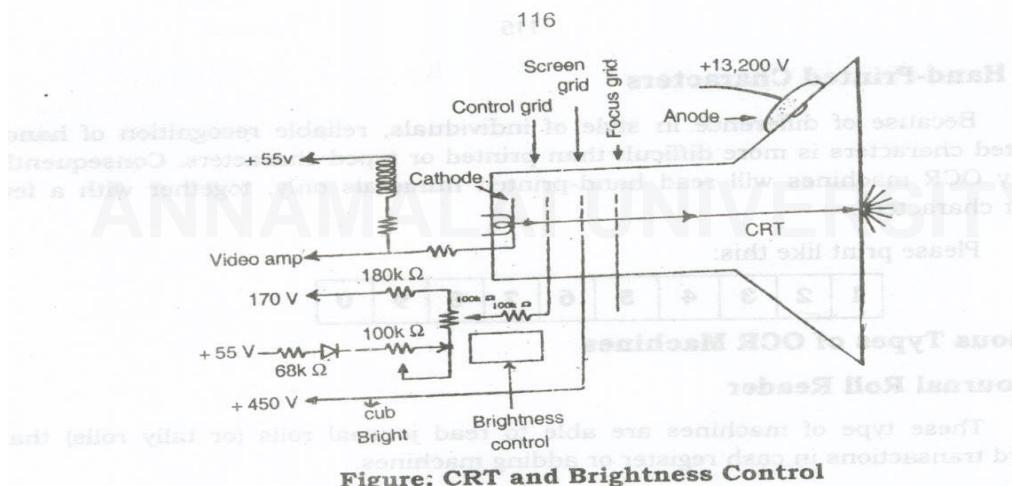


Figure: CRT and Brightness Control

When heated the cathode begins to emit electrons. It is heated by the filament which is sometimes called the hectare. The grids control the flow of the electrons from the cathode

Fundamentals of Digital Computers

to the anode, which is the most positively charged element. Since electrons are negatively charged particles, they are attracted towards the positive potential and repelled away from a negative potential. Electron beam is also affected by magnetic fields. Since current is flowing through the electron beam, a circular magnetic field is associated with it. When a magnetic field is placed through an electron beam, the beam is deflected within the externally induced magnetic field in the same way that two magnets attract or repel each other. This phenomenon is known as magnetic deflection.

The inside face of CRT is coated with phosphors, which is a material that exhibits luminescence (it glows) when excited by electrons (heat radiation) or other sources of radiation. When the electron beam from the cathode strikes the phosphor, it produces a bright dot of light which is known as cathode luminescence. The intensity of the dot of light is directly proportional to the intensity of the electron beam, which can be controlled by varying the voltage difference between the cathode and the control grid.

The last element is the anode. The inside surface of the CRT is coated with conductive paint to form the anode. Since the electrons must travel a great distance from the cathode to the anode high voltage must be present on the anode. Most colour CRTs require 25,000 volts. A colour CRT is similar to the monochrome CRT except that three electron guns are used, one for red, one for green, and one for blue, the primary these colours of light. By mixing these three colours, any colour is possible.

The other main difference from the monochrome CRT is that a shadow mask, a metal plate with many tiny holes in it, placed just before the face of the CRT. The face of the CRT is painted with many groups of red, green and blue phosphor dots called triads as shown in Figure. The shadow mask is set up in such a way that only the red electron beam hits the red dots, only the green beam hits these are green dots, and only the blue beam hits the blue dots. Since the three electron beams are not originate from the same point, an additional set of deflection magnets, called the 'Convergence Yoke' is required. This type of colour CRT is called a delta gun CRT because the electron guns are arranged in a delta form.

Many of the newer colour CRT designs put all three electron beams into one common electron gun. Another difference in the new CRT design is that the phosphor is painted as groups of vertical strips instead of dots. Directly behind the stripes is a slotted shadow mask instead of holes. The single-gun design greatly reduces the number of convergence components required on the CRT and in some cases eliminates the need for the convergence yoke.

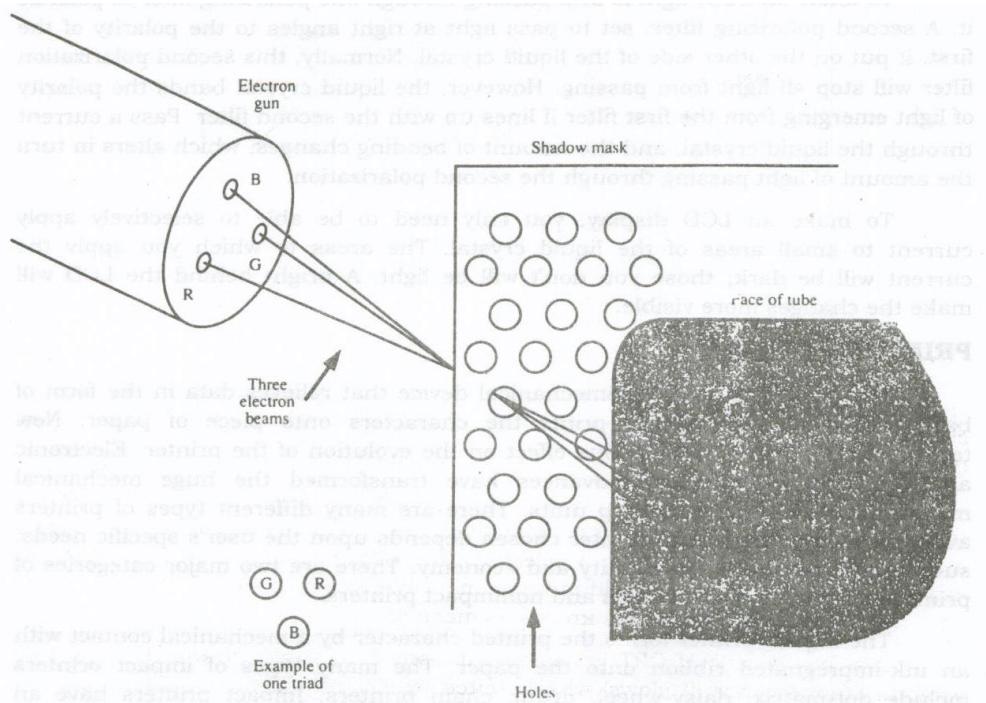


Figure: The Colour CRT

5.2.5 Flat Panel Display Systems

Because of their size and weight, CRTs are unpractical for portable computers, laptop (or) notebook computers. One alternative is the gas-plasma screen that uses a high voltage to ionise a gas and cause it to emit light. It need a lot of power, so when used in laptops the battery life of a gas-plasma equipped machine is quite brief, in the order of an hour.

Liquid Crystal Display (LCD) unlike gas-plasma displays do not waste energy by shining. An LCD display is actually as and which made from plastic sheets with a very special liquid. Polarization is key to the function of LCDs. A polarization filter creates polarized light by allowing light of a particular polarity to pass through.

To make an LCD, light is first passing through one polarizing filter to polarize it. A second polarizing filter, set to pass light at right angles to the polarity of the first, is put on the other side of the liquid crystal. Normally, this second polarization filter will stop all light from passing. However, the liquid crystal bands the polarity of light emerging from the first filter if lines up with the second filter. Pass a current through the liquid crystal, and the amount of bending changes, which alters in turn the amount of light passing through the second polarization.

To make an LCD display, you only need to be able to selectively apply current to small areas of the liquid crystal. The areas to which you apply the current will be dark; those you don't will be light. A bright behind the LCD will make the changes more visible.

5.2.6 Printers

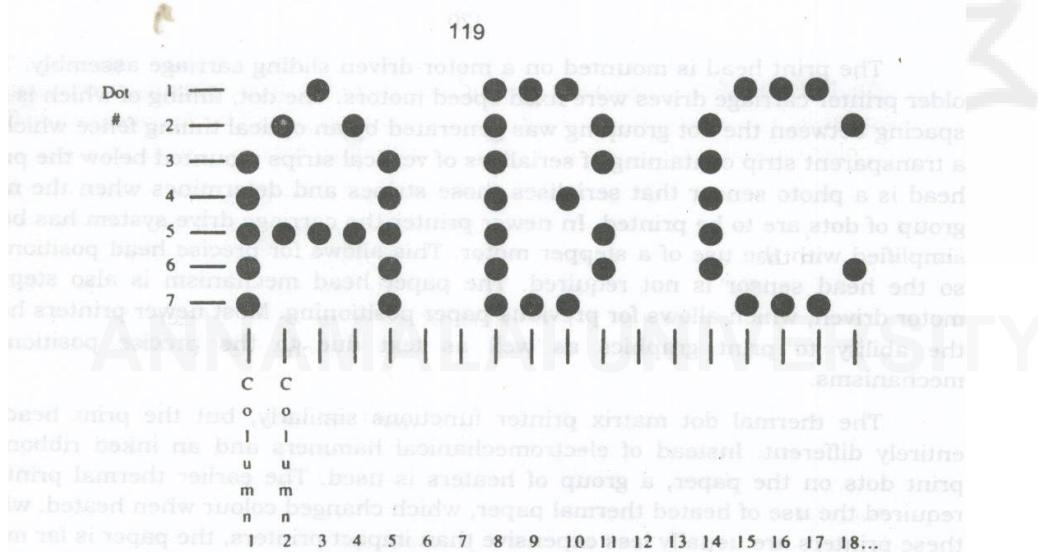
The printer is an electromechanical device that relieves data in the form of binary-coded characters and prints the characters onto piece of paper. New technology has had a tremendous effect on the evolution of the printer. Electronic and mechanical technology advances have transformed the huge mechanical machines into compact desktop units. There are many different types of printers available today. The type of printer chosen depends upon the user's specific needs, such as print speed, print quality and economy. There are two major categories of printers such as impact printers and nonimpact printers.

The impact printer forms the printed character by a mechanical contact with an ink-impregnated ribbon onto the paper. The many types of impact printers include dot-matrix, daisy-wheel, drum, chain printers. Impact printers have an electromagnet known as hammer, that supplies the mechanical energy to print the character.

A nonimpact printer forms the character without physically hitting an inked-ribbon on the paper. The many form of impact printing include thermal transfer, Ink jet, electrostatic and laser printers. Nonimpact printers have the unique advantage of being extremely quite.

Dot Matrix Printers (DMP)

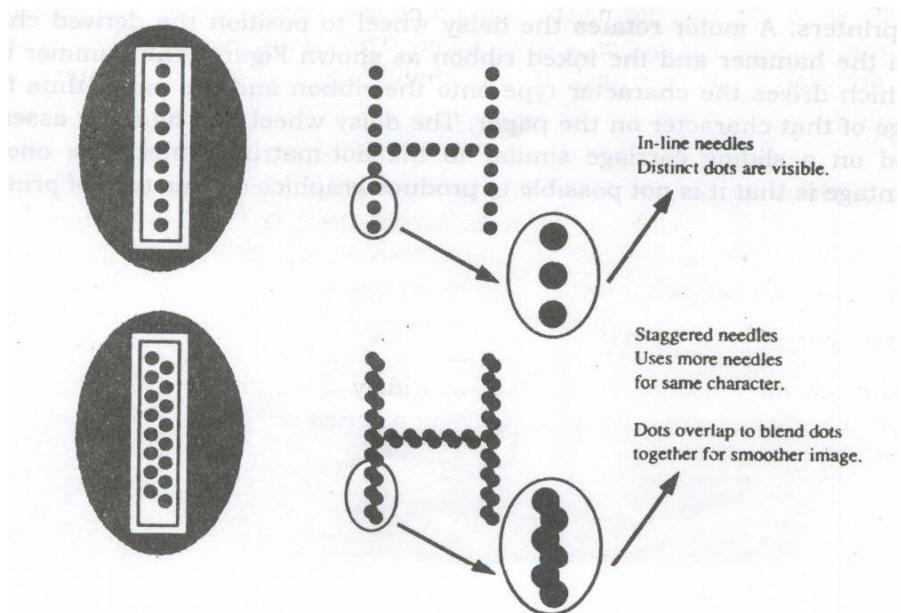
The dot matrix printer is - the most widely used printer on micro computer today. The characters are formed by printing a series of dots on the paper as shown in Figure.



Dot matrix printers are usually fast, but since the characters made up of dots, the character tend to be somewhat choppy. Some dot matrix printers offer a "near letter quality" mode by printing more dots per character but this slows down the print speed. The most widely used impact dot matrix printers use a group of needles or wires

Fundamentals of Digital Computers

connected to electro magnets. The needle and electro magnet assembly is known as print head. The needles are arranged in a group of 7 to 24 vertical dots, depending on the printer. The needles are often staggered to allow the dots to overlap, which causes the dots to blend together for a smoother image as shown in Figure.

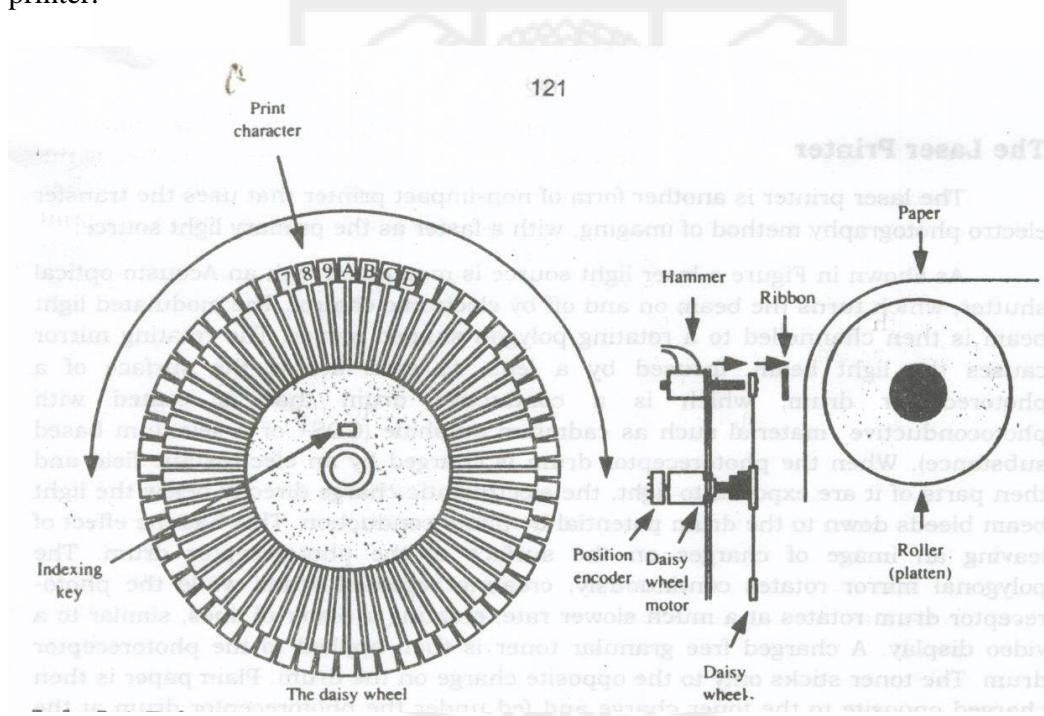


The print head is mounted on a motor-driven sliding carriage assembly. The older printer carriage drives were fixed speed motors. The dot timing of which is the spacing between the dot grouping was generated by an optical timing fence which is a transparent strip containing serialised vertical stripes mounted below the print head is a photo sensor that serialises those stripes and determines when the next group of dots are to be printed. In newer printers the carriage drive system has been simplified with the use of a stepper motor. This allows for precise head positioning so the head sensor is not required. The paper feed mechanism is also stepper motor driven, which allows for previous paper positioning. Most newer printers have the ability to print graphics as well as text due to the precise positioning mechanisms.

The thermal dot matrix printer functions similarly, but the print head is entirely different. Instead of electromechanical hammers and an inked ribbon to print dots on the paper, a group of heaters is used. The earlier thermal printers required the use of heated thermal paper, which changed colour when heated, while these printers are usually less expensive than impact printers, the paper is far more expensive. If the paper is accidentally heated or left out in the sunlight, it turns dark, making the printed copy unreadable. As a solution to this problem, the new printer used thermal ribbon.

Daisy Wheel Printer

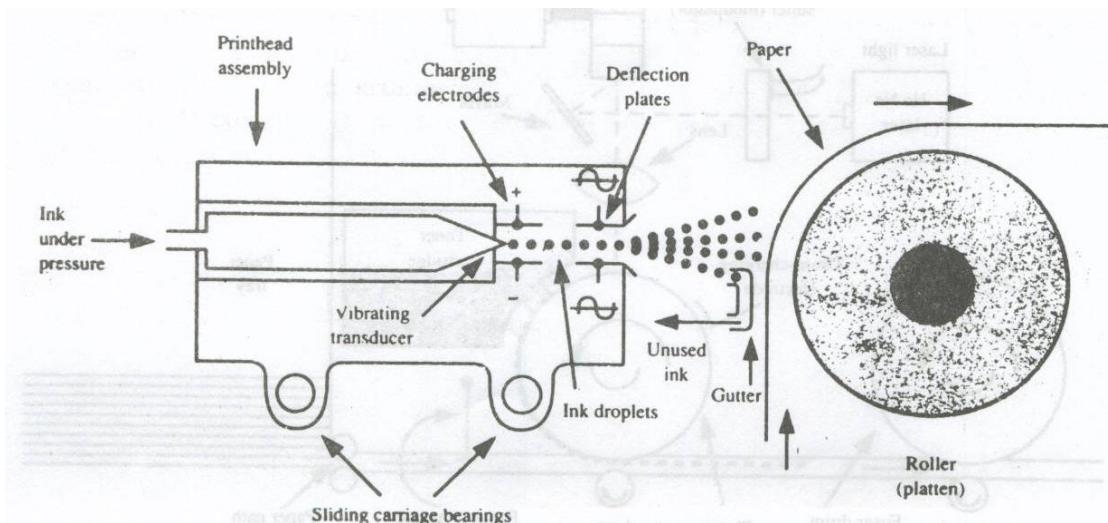
Daisy wheel printers are impact printers. The entire printable character set is contained on a plastic wheel that resembles a daisy. Each petal of the flower has a different character on it. This method produces the best type quality, since the entire character is typed just as on a typewriter. Since the daisy wheel must start, spin, stop and settle at the desired spot, these printers are much slower than dot matrix printers. A motor rotates the daisy wheel to position the derived character between the hammer and the inked ribbon as shown Figure. The hammer is often fired, which drives the character type onto the ribbon and the paper thus forming an image of that character on the paper. The daisy wheel and hammer assembly is mounted on a sliding carriage similar to the dot-matrix printer. The one major disadvantage is that it is not possible to produce graphics on this type of printer.



Ink Jet Printer

The ink jet printer is another form of nonimpact matrix printer. The characters or images are founded by firing ink droplets at a high velocity onto the paper. The ink droplets are formed by pumping liquid ink into a vibrating chamber with a small office at the opposite end. The ink droplets are electrically charged as they pass through an electrostatic field generated by charging plates. The droplets are then deflected up or down by the deflection plates to create the vertical matrix while the print head is sliding sideways to create the horizontal matrix as shown in Figure.

Fundamentals of Digital Computers



A variation of this method uses several orifices and fires the ink droplets without using a deflection system. The carriage and paper feed mechanism for the ink jet printer are similar to those used for the impact matrix printer.

The Laser Printer

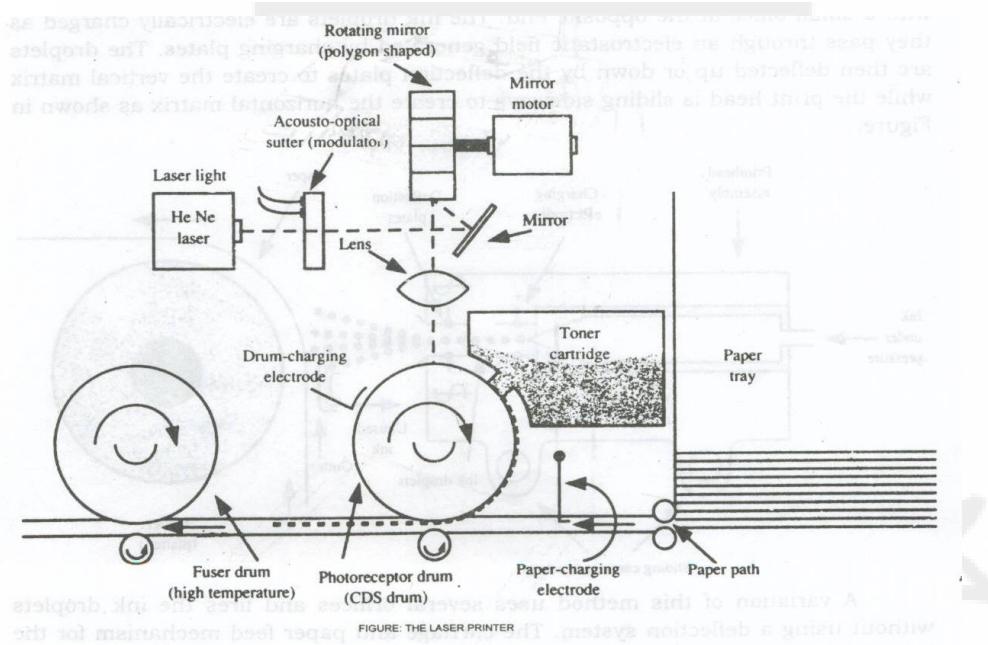
The laser printer is another form of non-impact printer that uses the transfer electro photography method of imaging, with a faster as the primary light source.

As shown in Figure a laser light source is modulated with an Acousto-optical light mirror based and light effect of photo- to a

optical light mirror causes the light beam, focused by a lens, to scan across the surface of a photoreceptor drum, which is a conductive drum that is coated with photoconductive material such as cadmium sulphide (CDS+ or a selenium substance). When the photoreceptor drum is charged by an electrostatic field and then parts of it are exposed to light, the electrostatic charge directly below the beam bleeds down to the drum potential by photo conduction. This has the leaving an image of charges on the surface of the photoreceptor drum. The polygonal mirror rotates continuously, creating repeated scans while the receptor drum rotates at a much slower rate, creating a roster of lines, similar to a

Fundamentals of Digital Computers

video display. A charged free granular toner is then applied to the photoreceptor drum. The toner sticks only to the opposite charge on the drum. Plain paper is then charged opposite to the toner charge and fed under the photoreceptor drum at the same speed in which the drum is rotating. The image formed by the toner transfers to the paper due to the higher positive charge on the paper. The paper is then sent to a fuses drum, which is heated and melts, a fuses, the toner image onto the paper, making the image permanent.



5.2.7 Teletype

One of the most successful printing terminal has been undoubtedly the teletype. There are three common teletypes, they are

- (1) Receive - Only teletype (RO)
- (ii) Keyboard Send-Receive teletype (KSR)
- (iii) Automatic Send-Receive teletype (ASR)

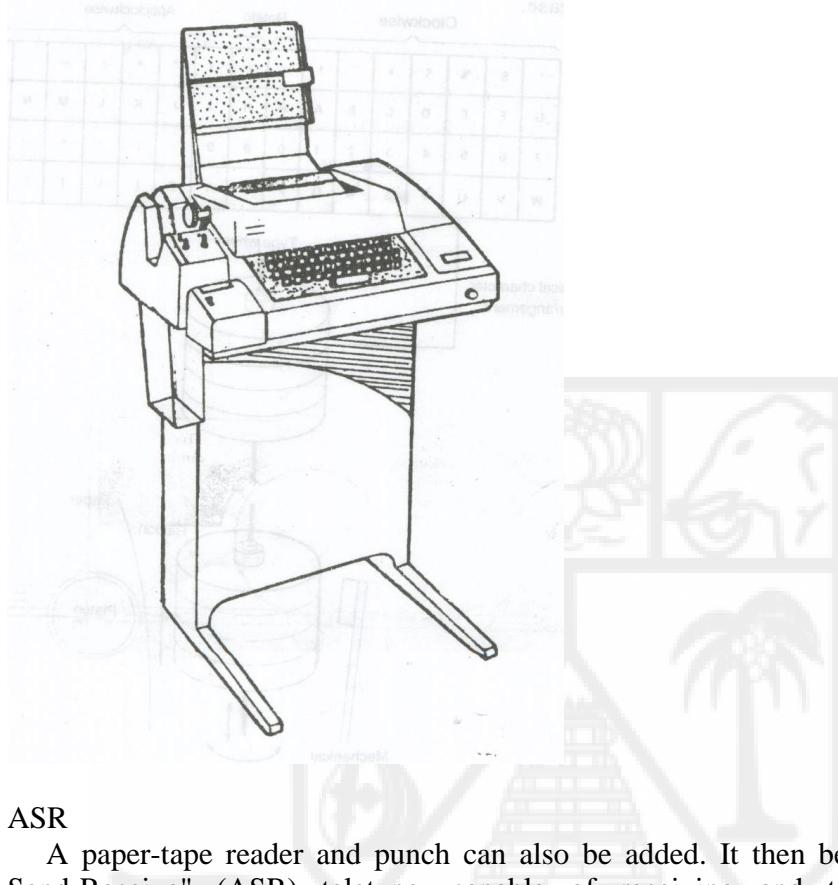
i) RO

When only a printer is incorporated the peripheral is known as "Receive (RO) teletype, having only the ability to receive and print the received

ii) KSR

Fundamentals of Digital Computers

With a keyboard added, the peripheral becomes known as a "Keyboard send-receive" (KSR) teletype. This has the ability to receive transmit and print messages.



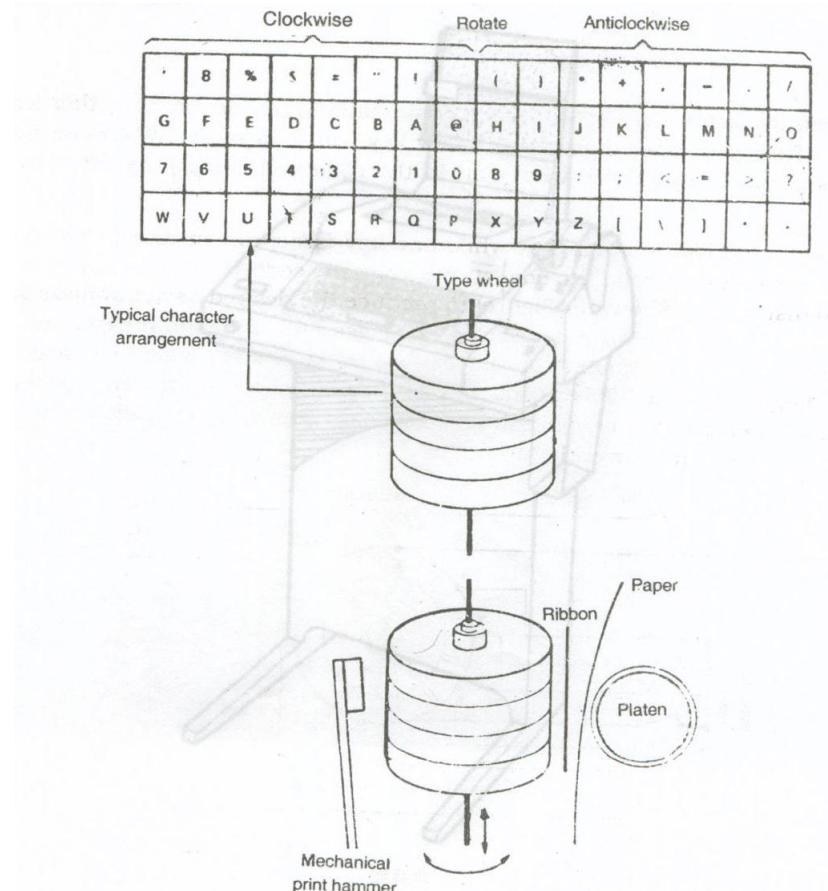
ASR

A paper-tape reader and punch can also be added. It then becomes an "Automatic Send-Receive" (ASR) teletype, capable of receiving and transmitting messages, producing a paper tape copy of the messages and printed copy. Paper tape can also be read for printing or transmission.

The terminal is almost completely mechanical with a mechanical keyboards, reader and punch. Given below are the brief details of the mechanical arrangements in a typical teletypes.

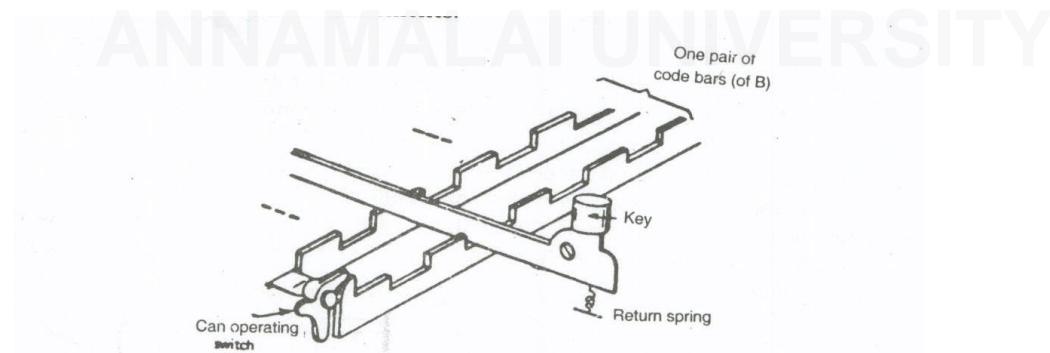
The cylinder print head used is as shown in Figure. The symbols to be printed are embossed in a metallic cylinder. With 64 symbols, there are four rows the 16 symbols on each row. The ribbon and paper are immediately in front of the cylinder and a hammer is positioned behind the head. The head is rotated clockwise or anti clockwise and moved vertically until the required strikes the back of the cylinder, forcing the cylinder to strike the ribbon. This causes the symbol to be printed on the paper. Only the upper case letter are provided, though it is feasible to have lower-case.

Fundamentals of Digital Computers



THE INSIDE STUFF

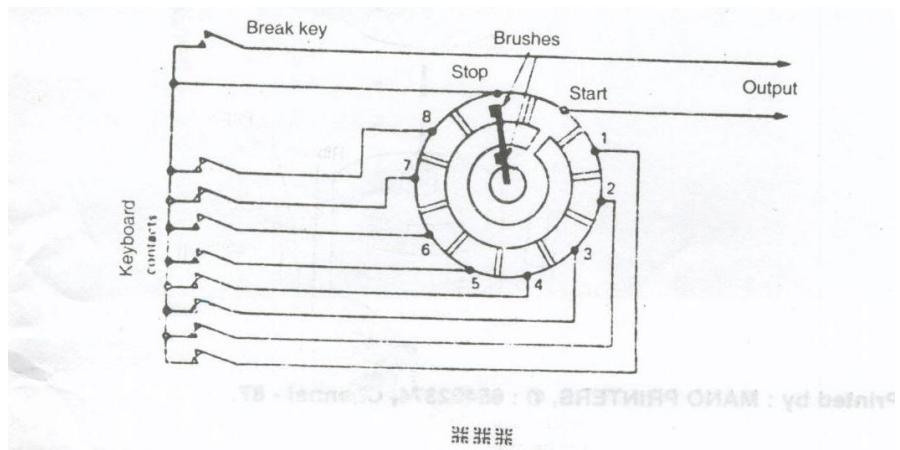
When a key is depressed the ASCII code is produced (with or without parity) by the keyboard contacts. This is done by using a pair of code bars for each bit of the code (as shown in Figure) one of which in the pair is depressed when the key is pressed depending upon slots in the bars. There is a slot in one bar to represent a '1' and a slot in the other bar to represent a '0'. The movements of the bar are translated into switch contact movements.



Fundamentals of Digital Computers

When the key is released the code combination remains until another key is depressed. The appropriate bars for the new key will then move. When one bar of the pair is pressed (only those whose bit in the code is different) the other in the pair returns.

The mechanism is such that while one key is depressed, others cannot be depressed because the solid portion of one or more code bars. The key contacts connect to distributor (Shown in Figure) to produce the desired asynchronous serial data output. The clutch of the distribution engages once a key is pressed and the shaft rotates once. The outer brush on the shaft moves over 10 segments corresponding to the start (S data) and stop bits. As the shaft rotates the signal path through the inner brush, outer brush and keyboard contacts produces the desired mark-space (current-no current) output.



ANNAMALAI UNIVERSITY

5.3) Revision Points

The Laser Printer

The laser printer is another form of non-impact printer that uses the transfer electro photography method of imaging, with a faster as the primary light source.

Printers

The printer is an electromechanical device that relieves data in the form of binary-coded characters and prints the characters onto piece of paper.

Liquid Crystal Display

Liquid Crystal Display (LCD) unlike gas-plasma displays do not waste energy by shining. An LCD display is actually as and which made from plastic sheets with a very special liquid

CRT

The CRT which is commonly known as the picture tube, is a large glass envelope containing an electron gun. The basic elements of the conventional CRT are the filament cathode

5.4) Intext Questions

1. Discuss about I/O Busses and Interfaces
2. Give details about the function of flat panel display
3. Explain in detail about CRT.
4. What is Character recognition? Explain the types.

5.5) Summary

- The many types of impact printers include dot-matrix, daisy-wheel, drum, chain printers. Impact printers have an electromagnet known as hammer, that supplies the mechanical energy to print the character.
- OMR can detect pencil or ink mark made on special paper forms using light sensors. The special forms are designed with square circle marks which can be filled with soft pencil or ink.
- OCR is capable of reading alphabetic and numeric characters printed on paper. The characters are of special design which are typed using a special font known as OCR font. OCR can read several thousands of printed characters per second.

5.6) Terminal Exercises

- 1.What is Teletype printing?
- 2.Give hints about Ink Jet Printer

3. Give short notes about Daisy Wheel Printer and Dot Matrix Printers
4. What is the function of Magnetic Ink Character Reader?
5. List the various types of printers.
6. Give difference and similarities of Optical mark readers & optical character readers.

5.7) Supplementary Materials

Multi-carrier Digital Communications: Theory and Applications of OFDM - ARS Bahai, BR Saltzberg, M Ergen

5.8) Assignments

Prepare assignment about the various types of printers.

5.9) Reference Books

Kai Hwan and Faye A. Briggs, 'Computer Architecture and Parallel Processing', McGraw Hill International Edition, 1985.

5.10) Learning Activities

An individual or group of peoples goes to library for future evaluation of this unit.

5.11) Keywords

- **Ink Jet Printer**
- **Daisy Wheel Printer**
- **Liquid Crystal Display**
- **CRT**
- **Magnetic Ink Character Reader**