



NEW AGE

FOURTH EDITION

COMPUTER FUNDAMENTALS

Architecture and Organization



B. Ram



NEW AGE INTERNATIONAL PUBLISHERS

COMPUTER FUNDAMENTALS

Architecture and Organization

**THIS PAGE
IS BLANK**

COMPUTER FUNDAMENTALS

Architecture and Organization

B. Ram

Formerly Professor and Head
PG Electrical Engineering, N.I.T., Patna
&
Dean Faculty of Engineering
Patna University, Patna



PUBLISHING FOR ONE WORLD

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS

New Delhi • Bangalore • Chennai • Cochin • Guwahati • Hyderabad
Jalandhar • Kolkata • Lucknow • Mumbai • Ranchi

Visit us at www.newagepublishers.com

Copyright © 2007, 2000, 1996 New Age International (P) Ltd., Publishers
Published by New Age International (P) Ltd., Publishers

All rights reserved.

No part of this ebook may be reproduced in any form, by photostat, microfilm,
xerography, or any other means, or incorporated into any information retrieval
system, electronic or mechanical, without the written permission of the publisher.
All inquiries should be emailed to rights@newagepublishers.com

ISBN : 978-81-224-2446-1

PUBLISHING FOR ONE WORLD

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS
4835/24, Ansari Road, Daryaganj, New Delhi - 110002
Visit us at www.newagepublishers.com

PREFACE TO THE FOURTH EDITION

The fourth edition of the book has been thoroughly revised and enlarged. It is suitable for the first course on computer organization and architecture, which is taught at B.Tech., BCA and MCA level. The fourth edition of the book gives the latest information on processors, peripherals, supporting chips, bus standards and softwares. New topics included in the book are as mentioned below:

Chapter 1: Centrino Notebooks, Centrino Duo Mobile Technology, middleware, freeware, multithreading, terminologies of mobile phone standards and data communication, short range wireless, microprogrammed control unit, virus, MP3 compression standard, MPEG, PMP, DivX etc.

Chapter 3: Alternative circuits for Mod 3, Mod 5, Mod 6, Mod 10 and Mod 12 counters.

Chapter 5: Intel 8086 instructions, Pentium 4, EPIC and Itanium processors, Athlon 64, Athlon 64 FX, Opteron and Athlon 64 \times 2 processors, Transmeta Corporation Crusoe chips, VIA C3, ARM and Cell processors.

Chapter 6: DDR SDRAM, RDRAM (Rambus RAM), Magneto-resistance, non-volatile RAM, WPCMCIA etc.

Chapter 7: TFTLCD monitor, OLED (Organic Light Emitting Diode) displays, MFDs, Chipsets, GPU (Graphical Processing Unit), etc.

Chapter 8: Optical mouse, SQL, DDL, DCL, DML, WINDOWS-XP, WINDOWS-2000, WINDOWS-2003, Microsoft Office-XP, FrontPage, etc.

Chapter 9: Corel Draw, Instant Messaging, Groupware, etc.

Chapter 10: USB, IEEE1394, PCI Express etc.

Chapter 11: Pipelining, Array Processors, multiprocessor system, Vector processor; UMA, SMP and NUMA systems, Cache coherence, snoopy cache scheme, directory-based scheme, Clusters, distributed memory type multiprocessor system, Flinn's classification of computers—SISD, SIMD, MISD and MIMD etc.

Dr. B. RAM

**THIS PAGE
IS BLANK**

ACKNOWLEDGEMENT

The author wishes to thank Dr. U.S. Triyar, Principal, APEEJAY College of Engineering, Sohna, Haryana and Prof. A.K. Sahni, Department of Electrical Engineering, NIT, Patna for their valuable suggestions and encouragement.

Finally, the author wishes to thank M/s New Age International (P) Ltd., Publishers, New Delhi, for bringing out the fourth edition of the book.

Dr. B. RAM

**THIS PAGE
IS BLANK**

CONTENTS

Preface (v)

Acknowledgement (vii)

1. INTRODUCTION	1.1
1.1 Digital and Analog Computers	1.1
1.2 Evolution of Digital Computers	1.2
1.3 Major Components of a Digital Computer	1.7
1.4 Memory Addressing Capability of a CPU	1.13
1.5 Word Length of a Computer	1.13
1.6 Processing Speed of a Microprocessor	1.14
1.7 Microprocessors	1.14
1.8 Single-Chip Microcomputers (Microcontrollers)	1.17
1.9 Computer Classification	1.18
1.10 User Interface	1.21
1.11 Hardware, Software, Firmware, Middleware and Freeware	1.21
1.12 Batch Processing, Multiprogramming and Multiuser System	1.23
1.13 Multithreading	1.24
1.14 Computer Network	1.24
1.15 Distributed Processing (Computing) or Multiprocessing	1.24
1.16 Local Area Network (LAN) and Wide Area Network (WAN)	1.25
1.17 Some Terminologies of Mobile Phone Standards and Data Communication	1.28
1.18 Short Range Wireless	1.30
1.19 Control Flow And Data Flow Computers	1.30
1.20 RISC, CISC and EPIC Processors	1.31
1.21 Computer Applications	1.32
1.22 Impact of Computers on Society	1.33
1.23 Future Developments	1.35
1.24 Virus	1.35

1.25	MP3 Compression Standard	1.36
1.26	MPEG Compression Standard	1.36
1.27	Personal Media Player PMP	1.36
1.28	Div X	1.36
	<i>Problems</i>	1.36
2.	NUMBER SYSTEM	2.1
2.1	Introduction	2.1
2.2	Decimal Number System	2.1
2.3	Binary Number (or Base-2) System	2.2
2.4	Conversion of a Binary Number to Decimal Number	2.3
2.5	Conversion of a Decimal Number to a Binary Number	2.4
2.6	Addition of Binary Numbers	2.6
2.7	Binary Subtraction	2.7
2.8	Use of Complements to Represent Negative Numbers	2.8
2.9	Conversion of a Binary Fraction to a Decimal Fraction	2.15
2.10	Conversion of a Decimal Fraction to a Binary Fraction	2.16
2.11	Binary Coded Decimal (BCD Codes)	2.17
2.12	Hexadecimal Number System	2.19
2.13	Octal Number System	2.23
2.14	ASCII and ISCII Codes	2.27
2.15	EBCDIC Code	2.30
2.16	Gray Codes	2.30
2.17	Fixed Point Representation of Numbers	2.33
2.18	Floating-Point Representation	2.34
2.19	Overflow and Underflow	2.39
	<i>Problems</i>	2.40
3.	DIGITAL DEVICES	3.1
3.1	Logic Gates	3.1
3.2	Flip-Flops	3.8
3.3	Latches	3.17
3.4	Registers	3.17
3.5	Shift Registers	3.17
3.6	Buffers/Drivers	3.18
3.7	Encoders, Decoders and Code Converters	3.18
3.8	Counters	3.19

3.9	Digital Multiplexers/Data Selectors	3.28
3.10	Digital Demultiplexers/Decoders	3.29
	<i>Problems</i>	3.30

4. LOGIC DESIGN 4.1

4.1	Boolean Algebra	4.1
4.2	Minimum Boolean Expression	4.23
4.3	Karnaugh Map Method of Simplification of Logic Expression	4.23
4.4	Combinational and Sequential Circuits	4.42
4.5	Arithmetic Circuits	4.43
	<i>Problems</i>	4.49

5. CENTRAL PROCESSING UNIT (CPU) 5.1

5.1	Introduction	5.1
5.2	CPU Organization	5.1
5.3	Instruction	5.4
5.4	Addressing Modes	5.8
5.5	Interrupts and Exceptions	5.12
5.6	Pin Configuration or Intel 8085	5.13
5.7	Generation of Some Special Control Signals for Memory and I/O devices	5.15
5.8	Instruction Cycle	5.16
5.9	Instruction and Data Flow	5.17
5.10	Timing Diagram	5.18
5.11	Microprocessors	5.24
5.12	Intel 8085	5.24
5.13	Intel 8086	5.24
5.14	Brief Description of Intel Microprocessors	5.35
5.15	Intel 486 (Intel 80486)	5.37
5.16	Pentium Processor	5.47
5.17	Pentium MMX Processor	5.48
5.18	Pentium PRO	5.48
5.19	Pentium II and Pentium II XEON	5.50
5.20	Celeron Processor	5.50
5.21	Pentium III Processor	5.51
5.22	Pentium 4 Processor	5.52
5.23	Itanium Processor	5.53

5.24	PowerPC Microprocessor	5.54
5.25	MIPS Microprocessors	5.57
5.26	DEC'S Alpha Microprocessors	5.57
5.27	Cyrix Microprocessors	5.57
5.28	AMD (Advanced Micro Devices) Processors	5.58
5.29	Sun's Ultrasparc Microprocessor	5.58
5.30	National Semiconductors Microprocessors	5.58
5.31	Motorola Microprocessors	5.58
5.32	Transmeta Corporation's Crusoe Chips	5.59
5.33	VIA's C3 Processor	5.60
5.34	ARM Microprocessors	5.60
5.35	Cell Processors	5.60
5.36	RISC, CISC and EPIC Machines	5.60
	<i>Problems</i>	5.61

6. MEMORY **6.1**

6.1	Main Memory, Secondary Memory and Backup Memory	6.1
6.2	Cache Memory	6.2
6.3	Real (or Physical) and Virtual Memory	6.4
6.4	Semiconductor Memory	6.5
6.5	Memory Controllers	6.9
6.6	Magnetic Memory	6.11
6.7	Optical Disks	6.21
6.8	CCD (Charge Coupled Devices)	6.25
6.9	Direct Access Storage Devices (DASDs)	6.26
6.10	Destructive and Nondestructive Readout	6.26
6.11	Program and Data Memory	6.26
6.12	Memory Management Unit (MMU)	6.26
6.13	PCMCIA or WPCMCIA Cards and Slots	6.29
	<i>Problems</i>	6.29

7. I/O DEVICES, MICROCONTROLLERS, I/O PROCESSORS AND ARITHMETIC PROCESSORS **7.1**

7.1	Input Devices	7.1
7.2	Output Devices	7.11
7.3	Printers	7.19
7.4	Plotters	7.24

7.5	Other Forms of Output Devices	7.25
7.6	Input and Output Port	7.25
7.7	Serial Data Transfer	7.28
7.8	Programmable Interval Timer/Counter	7.33
7.9	Chipsets	7.34
7.10	Data Transfer Schemes	7.34
7.11	Single-Chip Microcomputers (Microcontrollers) and Embedded Microprocessors	7.45
7.12	DSP (Digital Signal Processor)	7.47
7.13	I/O Processor	7.47
7.14	Arithmetic Processors	7.49
7.15	Graphics Processors	7.51
	<i>Problems</i>	7.51
8.	SYSTEM SOFTWARE AND PROGRAMMING TECHNIQUES	8.1
8.1	Machine Language	8.2
8.2	Assembly Language	8.2
8.3	Low-Level and High-Level Languages	8.3
8.4	Stack	8.8
8.5	Subroutines	8.9
8.6	Debugging of Programs	8.11
8.7	Macro	8.12
8.8	Macro Processor (or Keyboard Enhancer)	8.13
8.9	Program Design	8.14
8.10	Software Development	8.18
8.11	Flow Charts	8.20
8.12	Multiprogramming or Concurrent Programming	8.22
8.13	Multiuser or Time-Share System	8.23
8.14	Multitasking System	8.23
8.15	Protection	8.24
8.16	Problems in Accessing Resources	8.24
8.17	Operating System	8.25
8.18	Utility Programs	8.31
8.19	Application Packages	8.32
8.20	BIOS	8.45
	<i>Problems</i>	8.46

9. COMPUTER APPLICATIONS	9.1
9.1 Computer Applications in Offices	9.1
9.2 Use of Computers in Books Publication	9.2
9.3 Desktop Publishing System	9.3
9.4 Application of Computers for Data Analysis	9.4
9.5 Accounting, Investment, Inventory Control etc.	9.4
9.6 Application of Computers in Graphics	9.5
9.7 CAM (Computer Aided Manufacture)	9.8
9.8 Application of Computers in Database Management	9.8
9.9 Application of Computers in Project Management	9.9
9.10 Use of Computers in Communications	9.10
9.11 Application of Computers in Education	9.14
9.12 Application of Computers in Medical Field	9.15
9.13 Application of Computers in Industrial Control and Instrumentation	9.16
9.14 Airlines and Railways Ticket Reservation	9.17
9.15 Robots	9.18
9.16 Artificial Intelligence	9.18
9.17 Military Application of Computers	9.20
9.18 Application of Computers in Banks	9.20
9.19 Application of Computers in Design and Research Work	9.20
9.20 Real-Time or Online Applications	9.21
9.21 Point of Sale (POS) Terminals	9.21
9.22 Financial Transaction Terminals	9.22
9.23 Some Other Application of Computers	9.22
9.24 Computers with Visions	9.23
<i>Problems</i>	9.25
10. BUS ARCHITECTURE	10.1
10.1 Introduction	10.1
10.2 ISA BUS	10.1
10.3 EISA BUS	10.2
10.4 PCI Bus	10.3
10.5 AGP (Accelerated Graphic Port)	10.3
10.6 PCI Express Bus	10.3
10.7 Universal Serial Bus (USB)	10.5
10.8 IEEE 1394	10.5

10.9 IEEE-1284	10.6
10.10 IEEE-488 BUS	10.6
10.11 Bus System in a Multiprocessor System	10.6
<i>Problems</i>	10.9
11. PARALLEL PROCESSING	11.1
11.1 Introduction	11.1
11.2 Pipelined Processors	11.1
11.3 Array Processors	11.3
11.4 Vector Processors	11.4
11.5 Multiprocessor system	11.4
11.6 Flinn's Classification of Computers	11.11
<i>Problems</i>	11.11
<i>Appendix</i>	A-1
<i>Bibliography</i>	B-1
<i>Index</i>	I-1

**THIS PAGE
IS BLANK**

1

CHAPTER

INTRODUCTION

INTRODUCTION

Computer is the most powerful tool man has ever created. Computers have made a great impact on our everyday life. Their presence is felt at almost every working place, viz. homes, schools, colleges, offices, industries, hospitals, banks, retail stores, railways, research and design organizations and so on. Computers, large and small, are used nowadays by all kinds of people for a variety of tasks in a modern and industrialized society.

A computer is basically a programmable computing machine. Earlier, computers were used for complex computations and used by only scientists and engineers. The trend was to design large and powerful computers to handle large data and solve complex problems. They were very costly and hence, only large organizations could afford them. The technological breakthrough in design and fabrication of semiconductor devices has made now possible to manufacture powerful microcomputers which are within the reach of small organizations and even individuals. These computers being very fast can be used not only for computation but also to store and retrieve information, to control certain processes and machines, to measure and display certain physical and electrical quantities and so forth. Developments in software allow massive applications of computers for non-computational jobs like text preparation, manipulation, storage and retrieval; transmission of texts, graphics and pictures from one place to another; and artificial intelligence and expert systems, for example, robots, and so on.

1.1 DIGITAL AND ANALOG COMPUTERS

Computers which are in use today are digital computers. They manipulate numbers. They operate on binary digits 0 and 1. They understand information composed of only 0s and 1s. In the case of alphabetic information, the alphabets are coded in binary digits. A binary digit is called *bit*. A group of 8 bits is called a *byte*. Computers do not operate on analog quantities directly. If any analog quantity is to be processed, it must be converted into digital quantity before processing. The output of a computer is also digital. If analog output is needed, the digital output has to be converted into analog quantity. If output is to be displayed in the form of text, the digital output is converted to alphabets. The components which convert alphanumeric characters to binary format and binary output to alphanumeric characters are the essential parts of a digital computer. But the electronic components which convert

analog quantity to digital quantity or digital quantity to analog quantity are connected to a digital computer as peripherals where needed. Processing of analog quantity is usually encountered in industrial control and instrumentation, not in general purpose computation, text manipulation or information storage, retrieval or transmission.

The computer which can process analog quantities is called an *analog computer*. Today, analog computers are rarely used. Earlier, analog computers were used to simulate certain systems. They were used to solve differential equations.

1.2 EVOLUTION OF DIGITAL COMPUTERS

Electronic computers using valves appeared in 1940s. The successful general purpose mechanical computers were developed in 1930s. Before 1930 mechanical calculators were built for automatic addition, subtraction, multiplication and division. A calculator is not a programmable device. Calculations are performed using step-by-step technique. The user does not prepare program for his calculation. A computer is a programmable machine. A program is to be prepared to solve a problem.

1.2.1 The Mechanical Era

The first mechanical calculator was developed in 1623 by Wilhelm Schickhard, a professor at the University of Tübingen. His machine did not become popular. A popular mechanical calculator was developed in 1642 by the great French philosopher and scientist Blaise Pascal. His machine was capable of performing addition and subtraction automatically. For this the machine employed counter wheels. There were two sets of six dials or counter wheels to represent decimal numbers. The calculator contained a mechanism for automatic transfer of carry while performing the sum of two numbers. The numbers were represented by the positions of the counter wheels. Around 1671 Pascal's machine was extended to perform multiplication and division automatically by German philosopher and scientist Gottfried Leibniz. This machine consisted of two parts: one part to perform addition and subtraction and the other part to perform multiplication and division. The part which performed addition and subtraction was similar to the calculating box of Pascal. It further included two additional sets of wheels to represent multiplier and multiplicand. Chains and pulleys were used to implement multiplication.

In 1823, Charles Babbage tried to build a mechanical computing machine capable of performing automatic multistep calculations. He named his machine a difference engine. This was designed to compute tables of functions such as logarithms and trigonometric functions. A polynomial was used to represent a function. The method of finite differences was used to evaluate a function. He could not complete the machine. Swede George Scheutz successfully built a difference engine which could handle third-degree polynomials and 15-digit numbers.

In 1830s Charles Babbage conceived of a much more powerful mechanical computer. He called this machine an analytical engine. This machine was designed to perform any mathematical calculation automatically. It contained all the essential components of a modern digital computer, namely:

- (i) A processor capable of performing addition, subtraction, multiplication and division.
He called it a 'mill'.

(ii) A memory unit. It was constructed from decimal counting wheels. Its capacity was 1000 numbers, each number consisting of 50 digits.

(iii) Several I/O devices such as a card punch, a punch-card reader and a printer.

The analytical machine was a programmable machine. It had a mechanism for enabling a program to change the sequence of its operations automatically. In other words there were conditional branches of instructions in the program. The condition was based on the sign of a number. One sequence of operations was to be performed if the sign were positive, and another one, if negative. Babbage's analytical machine was also not completed.

In the late nineteenth century punched cards were commercially used. Herman Hollerith was the inventor of punched-card tabulating machine. The major application of his machine came about in the 1890 United States Census. In 1896 he formed the Tabulating Machine Company to manufacture his machines. In 1911 his company was merged with several others to form the Computing-Tabulating Recording Company. This very company was renamed as the International Business Machines Corporation (IBM) in 1924.

Successful general purpose mechanical computers were built in 1930s. Konrad Zuse developed a mechanical computer, the Z1, in 1938 in Germany. The Z1 used binary number system instead of decimal system. Konrad was unaware of Babbage's work. He built several small mechanical computers. The Z3 was completed in 1941. It is believed to be the first operational general purpose computer. The Z3 employed relays (electromechanical binary switches) to construct arithmetic unit. The machine used floating-point number representation. Howard Aiken, a professor of Physics at Harvard University, designed a general purpose mechanical digital computer. This machine was called an Automatic Sequence Controlled Calculator and later as Harvard Mark I. It was constructed in cooperation with IBM, a leading manufacturer of office equipment at that time. Aiken was aware of Babbage's work. He used decimal counters wheels for its main memory. Its memory capacity was seventy two 23-digit decimal numbers. Punched paper tape was used to program and control the machine. Mark I started working in 1944. Later, Mark II was built by Aiken and his colleagues. Mark II employed electromechanical relays for its operation. Many computers using electromechanical relays were built in the 1940s. But they were quickly superseded by faster and more reliable electronic computers.

1.2.2 The Electronic Era

The first electronic computer using valves was developed by John V. Atanasoff in the late 1930s at Iowa State University. It contained an add-subtract unit. It was relatively a small computer and used about 300 valves. Its memory unit consisted of capacitors mounted on a rotating drum. It used binary numbers for its operation. Each capacitor was capable of storing one binary digit. It used a number of input/output (I/O) devices including a card punch and a card reader. It was completed in 1942. It was a special purpose computer to solve simultaneous equations. Several other electronic computers using valves were successfully constructed in the early 1940s. Some important computers were the series of computers called Colossus developed in England.

The first popular general purpose electronic digital computer was the ENIAC (Electronic Numerical Integrator and Calculator). It was developed at the University of Pennsylvania under the guidance of John W. Mauchly and J. Presper Eckert. John von Neumann was the consultant of the ENIAC project. It was a very large machine weighing about 30 tons and containing about 18000 vacuum tubes. It took 200 microseconds for addition and 3 milliseconds

to perform a 10-digit multiplication. It used decimal numbers for its operation rather than binary numbers. Its working memory was composed of 20 electronic accumulators. Each accumulator was capable of storing a signed 10-digit decimal number. A decimal digit was stored in a ring counter consisting of 10 vacuum-tube flip-flops connected in a closed loop. Like Analytical Engine and Mark I, in ENIAC also programs and data were stored in separate memories. Introducing a new program or modifying a program was an extremely tedious job with separate memories for program and data.

The ENIAC designers, most notably John von Neumann, gave an idea to use a high-speed memory to store both program as well as data during program execution. This idea is known as *stored program concept* and was first published by Neumann for a new computer EDVAC (Electronic Discrete Variable Automatic Computer) in 1945. This machine started operation in 1951. It used binary rather than decimal numbers for its operation. It used serial binary-logic circuits. It used a larger main memory (mercury-delay line) 1 K words and a slow secondary memory (magnetic wire memory) 20 K words (where K stands for Kilo which is equal to 1024 to be exact). Access to the main memory was bit by bit, i.e., serial.

Neumann and his colleagues designed and built a new computer called IAS (Institute of Advanced Studies) at the Institute for Advanced Studies in Princeton during 1946-1952. This machine had the features of a modern computer. It used random access main memory consisting of cathode-ray-tube. An entire word could be accessed in one operation. It used parallel binary circuits. The CPU contained several high-speed (vacuum tube) registers to store operands and results. This computer served as the prototype for most subsequent general purpose computers. The basic logical structure proposed by Neumann is still used in a standard computer. The term Neumann Computer became synonymous with standard computer architecture. A standard architecture includes a CPU, memory and input/output devices. In future the architecture may change; instead of a centralized processing, distributed processing may be used with corresponding other changes in the design and architecture.

The transistor was invented in 1948 at AT & T Bell Laboratories. In the 1950s the engineers started using transistors in place of vacuum tubes to construct computers. One of the earliest computers using transistors was TX-O. It was an experimental computer built at the Massachusetts Institute of Technology's Lincoln Laboratories. It started operation in 1953. Commercial computers using transistors were constructed in the late 1950s and early 1960s by many companies. For example, IBM introduced a large computer, the 7090, for scientific applications. It was a transistorized version of the IBM 709, a vacuum-tube computer. The transistorized computers used transistors as the components of CPU. These computers used ferrite core main memory and magnetic disk, drum and tapes as secondary memory. Ferrite core memories consist of tiny rings (cores) of magnetic material called ferrite. Each ferrite core stores a single bit of information. Transistorized computers were faster and compact, and consumed much less power compared to vacuum tube computers.

Integrated Circuits (ICs) were first designed and fabricated in 1958-1959 by Jack S. Kilby at Texas Instruments, and by Robert S. Noyce at Fairchild independently. The first commercial IC was introduced in 1961 by Fairchild. ICs began to replace transistor circuits since 1965. The examples of computers using ICs are IBM 370 and PDP-8. By 1970 all new computers used ICs, SSI and MSI as CPU components and LSI for main memory. SSI, MSI, LSI, VLSI and ULSI are the classification of ICs based on components density. SSI contains components, usually transistors, 1 to 100, MSI 100 to 1000, LSI 1000 to 10,000, VLSI more than 10,000 and ULSI millions.

The first LSI chips were introduced in 1970 in the form of computer memory units. With the advent of LSI and VLSI chips it became possible to fabricate the whole CPU unit on a single chip called microprocessor. The first microprocessor, the 4004 was introduced in 1971 by Intel Corporation. The first single-chip microcomputer TMS 1000, a 4-bit microcontroller, was developed by Texas Instruments in the year 1974. An 8-bit microcontroller, the 8048 was introduced in 1976 by Intel. Computers built in 1970s and onwards used microprocessors and other LSI, VLSI and ULSI components.

Computer Generations

First Generation (1946-1954). The digital computers using electronic valves (vacuum tubes) are known as first-generation computers. Some examples of the first-generation computers are: IBM 700 series-IBM 701, IBM 704, IBM 709, EDVAC and UNIVAC. The first-generation computers usually used vacuum tubes as CPU components. The high cost of vacuum tubes prevented their use for main memory. So less costly but slower devices such as acoustic delay lines were used for memory. They stored information in the form of propagating sound waves. Electrostatic memories have also been used in the first generation computers. Magnetic tape and magnetic drums were used as secondary memory. A first generation computer, Whirlwind I, constructed at MIT was the first computer to use ferrite core memory. The first generation computers used machine language and assembly language for programming. They used fixed-point arithmetic. Punched cards and paper tapes were developed to feed programs and data and to get results. Punched card and paper tape readers and printers were in use.

Second Generation (1955-1964). The second-generation computers used transistors for CPU components and ferrite cores for main memory, and magnetic disks and tapes for secondary memory. They used high-level languages such as FORTRAN (1956), ALGOL (1960) and COBOL (1960) for programming. Floating-point arithmetic hardware was widely used. I/O processor was included to control input/output operations. It relieved CPU from many time-consuming routine tasks. Examples of second generation computers are: IBM 1620 (1960), IBM 7090 (1960), IBM 7094I (1962), 7094II (1964); Control Data Corporation's CDC 1604; and Digital Data Corporation's PDP 1 (1957), PDP 5 (1963) and PDP 8 (1965). PDP (Programmed Data Processor) series is a series of minicomputers. PDP 8 was a 12-bit minicomputer. Its earlier units used transistors; IC version was introduced in 1967. Punched cards and paper tapes and their readers were used as I/O devices. Printers were in use.

Third Generation (1965-1974). The third-generation computers used ICs (SSI and MSI) for CPU components. In the beginning third generation computers used magnetic core memory, but later on semiconductor memories (RAMs and ROMs) were used. Semiconductor memories were LSI chips. Magnetic disks, and tapes were used as secondary memories. Cache memory was also incorporated in the computers of third generation. Microprogramming, parallel processing (pipelining, multiprocessor system, etc.), multiprocessing, multiprogramming, multiuser system (time-share system), etc. were introduced. The concept of virtual memory was also introduced. The examples of third generation computers are: IBM/370 series (1970), CDC 7600 (1969), PDP 11 (16-bit minicomputer, 1970), CDC's CYBER-175 and STAR-100, etc. I/O devices were punched cards, magnetic tapes and printers.

Fourth Generation (1975-1990). In the fourth-generation computers microprocessors were used as CPU. VLSI chips were used for CPU, memory and supporting chips. The electronic circuitry of up to 1.2 million transistors were placed on a single silicon chip.

Computers of earlier generation used separate ICs for cache memory, FPU (Floating-Point Unit *i.e.*, Math Processor), MMU (Memory Management Unit) etc. Now microprocessor chips contained all such units besides CPU on a single chip. They were packed in a single IC. Multifunctional peripheral chips were available. They contained interrupt controller, DMA controller, timer-counters, bus controller etc. in a single IC. These are essential components required for a computer. Computer of this generation were very fast. They performed internal operations in microseconds. 8, 16 and 32-bit microprocessors were developed during this period.

Main memory used fast semiconductor chips up to 4 Mbits size. Hard disks were used as secondary memory. Hard disk drives of hundreds of megabytes were available. Floppy disks and magnetic tapes were used as backup memory. Keyboard, CRT display (monitor), dot-matrix printers etc. were used as peripherals. Inkjet, laser and line printers, were developed during this period. PCs (Personal Computers) were available. Such computers can be easily placed on a desk and hence, they were also known as desk computers. They were single-user computers. During this period computers were within the reach of small organization, institutions, professionals and individuals. The desktop computers were more powerful than the mainframe computers of 1970s. Computers became very powerful and small in size. During this period computer network: LANs and WANs were also developed. Operating systems MS-DOS, UNIX, Apple's Macintosh etc. were available. Apple's Macintosh with GUI (Graphical User Interface) was developed. Object-oriented language C++ was developed. Single-chip microcomputers (microcontrollers) were available. They were widely used in industrial control, instrumentation, commercial appliances etc. Software packages for word processing, spread-sheet, database management etc. were developed. Examples of fourth-generation computers were: Intel's 8088, 80286, 80386 and 80486 based computers; Motorola's 6800, 68020, 68030 and 68040 based computers, IBM 3090, VAX 9000, Supercomputers-Cray-1, Cray-2, Cray X-MP, Cray Y-MP, Hitachi 828/80 etc.

Fifth-Generation (1991-Continued). Fifth-generation computer use ULSI (Ultra-Large Scale Integration) chips. Millions of transistors are placed in a single IC in ULSI chips. Intel's Pentium 4 Prescott contains 160 million transistors and Itanium 2 processor contains more than 400 million transistors. 64-bit microprocessors have been developed during this period. Data flow and EPIC architectures of processors have been developed. Intel's processors Pentium Pro onwards use data flow architecture and Itanium uses EPIC architecture. Von Neumann architecture are still used in less powerful CPUs. RISC and CISC both types of design are used in modern processors. Intel's up to Pentium 4 and AMD's processors use CISC design. SUN, MIPS, HP etc. use RISC design for their processors. 32-bit microcontrollers (Single-Chip Microcomputers) have been developed for industrial and commercial application. Nowadays multimedia computers are becoming common. A multimedia computer can accept text, sound and/or image as input and give output in the form of text, sound and/or images. They can handle animation. Computers using artificial intelligence (expert systems) are now available. Robots have been developed. They can work in environment where human beings can not do. Powerful handheld and notebook computers are now available. Fifth-generation computers use extensive parallel processing, multiple pipelines, multiple processors etc.

Memory chips and flash memory up to 1Gbits, hard disk drives up to 600 Gbytes and optical disks up to 50 GB have been developed. Inkjet and laser printers are commonly used. In some applications dot-matrix printers are still used. Computers with vision have been developed. Internet is becoming popular and people are getting all kinds of information from distant places using Internet. Video conferencing is also in use. Object-oriented language Java for Internet programming is widely used. Heterogeneous computers are connected to Internet.

Heterogeneous computers mean computers of different make and having different operating systems. Programs written in Java for one computer can run on any other computer. It means that Java is quite suitable for Internet programming for heterogeneous computers. Operating systems available today are : WINDOWS-95, 98, XP, 2000, 2003; Apple's Mac OS-8, 9, 10 and X; SUN's Solaris, LINUX etc. All OS now include GUI. Examples of processors are: Intel's Pentium to Pentium 4, Itanium; Motorola's Power PC, MIPS, Compaq's Alpha, SUN's Ultra SPARC III, AMD's Athlon, Athlon 64, ARM processors, Cell processors etc. Internal processing time is now in nanoseconds.

Superscalar processors, vector processors, DSP (Digital Signal Processor), symbolic processors, SIMD (Single Instruction Multiple Data) processors, multicore processors, expert systems employing artificial intelligence, etc. have been developed. Supercomputers available today are IBM's BlueGene/L DD2 Beta-system having speed of 70.7 TFLOPS, Columbia (NASA) having speed of 51.9 TFLOPS, Earth simulator (NEC) having speed of 40 TFLOPS etc.

1.3 MAJOR COMPONENTS OF A DIGITAL COMPUTER

The major components of a digital computer are: CPU (central processing unit), memory, input device and output device. The input and output devices are also known as peripherals. Fig. 1.1 shows a schematic diagram of a digital computer.

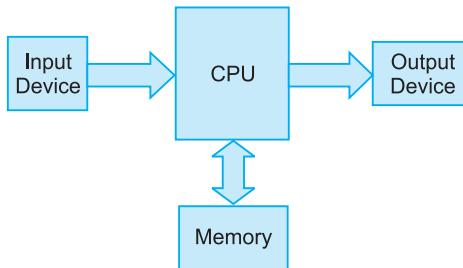


Fig. 1.1 Schematic diagram of a digital computer.

1.3.1 CPU

The CPU is the brain of a computer. Its primary function is to execute programs. Besides executing programs, the CPU also controls the operation of all other components such as memory, input and output devices. Under its control, programs and data are stored in the memory and displayed on the CRT screen or printed by the printer.

The CPU of a small computer is a microprocessor. Fig. 1.2 shows the schematic diagram of a microcomputer. The CPU of a large computer contains a number of microprocessors and other ICs on one or more circuit boards. Each microprocessor in a large CPU performs a specific task.

Fig. 1.3 shows the schematic diagram of a CPU or microprocessor. The major sections of a CPU are:

(i) Arithmetic and Logic Unit (ALU)

The function of an ALU is to perform arithmetic and logic operations such as addition, subtraction, multiplication, and division; AND, OR, NOT (complement) and EXCLUSIVE-OR operations. It also performs increment, decrement, shift and clear operations.

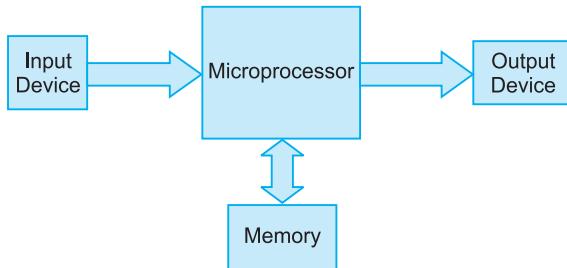


Fig. 1.2 Schematic diagram of a microcomputer.



Fig. 1.3 Schematic diagram of a CPU or microprocessor

(ii) Timing and Control Unit

The timing and control unit generates timing and control signals necessary for the execution of instructions. It provides status, control and timing signals necessary for the operation of other parts of the CPU, memory and I/O devices. It controls the entire operation of a computer. It is actually the control section of the CPU, which acts as the brain of a computer.

(iii) Accumulator, General and Special Purpose Registers

The accumulator is a register which holds one of the operands prior to the execution of an instruction and services result of the most arithmetic and logical operations. It is the most frequently used register. Some CPUs contain a single accumulator, and some contain several accumulators. General purpose registers store data and intermediate results during the execution of a program. They are accessible to programmers through instructions if they are working in an assembly language. Special purpose registers are not accessible to users. They are used by the computer for different purposes during program execution. Examples of special purpose registers are: program counter, stack pointer, index registers, instruction register, etc.

1.3.2 Memory

The function of the memory is to store information. It stores program, data, results or any other kind of information. Two or three levels of memories such as main memory, secondary memory and cache memory are provided in a digital computer. The *main memory* (or primary memory) is a fast memory. It stores programs along with data, which are to be

executed. It also stores necessary programs of the system software, which are required to execute the user's program. The main memory is directly addressed by the CPU. Semiconductor memories, RAMs are used as main memory. It possesses random access property, and has smaller access time, about 50 ns (nanosecond). *Secondary (or auxiliary) memory* stores operating system, data files, compilers, assemblers, application programs, etc. The CPU does not read information (residing in the secondary memory) directly from the secondary memory. The programs and data (residing in secondary memory), if needed by CPU, are first transferred from the secondary memory to the primary memory. Then the CPU reads them from the primary memory. The results are also stored in the secondary memory. The secondary memory is a mass storage memory. It is slow but cheap. It is a permanent memory while the main memory (RAM) is volatile memory. The capacity of the main memory is comparatively much smaller than that of the secondary because of its high cost. Hard disks are used as secondary memory. Their access time is about 5-10 ms (millisecond).

The *cache memory* is placed in between the CPU and the main memory. It is much faster than the main memory; access time about 10 ns. It stores instructions and data which are to be immediately executed. It is much costlier than the main memory. Hence, from cost consideration its capacity is kept much less than that of the main memory.

destructive and Nondestructive Readout

In some memories the process of reading the memory destroys the stored information. This property is called *destructive readout (DRO)*. Example of a memory having DRO characteristic is a dynamic RAM. In some memories the process of reading information does not destroy the stored information. This characteristic of the memory is called *nondestructive read-out (NDRO)*. Examples of memories having NDRO features are static RAM, hard disks, floppy disks, magnetic tapes, etc.

Real (or Physical) and Virtual Memory

The real or physical memory is the actual main memory available in a computer system. It is directly addressed by the CPU.

The technique which allows a program to use main memory more than a computer really has is called *virtual memory technique*. For example, the 80386 microprocessor can have the maximum physical memory capacity 4 gigabytes (GB) but its virtual memory capacity is much larger, 64 terabytes (TB) [see details in Chapter 6].

Direct Access Storage Devices (DASD), On-Line and Off-Line Memory Devices

While processing data it is often required to access any record at any time. It may be desired to access a single record, update it and put it back in its original place. This type of data processing is called direct processing or random processing. It needs locating, retrieving and updating any record stored in a file without reading the preceding or succeeding records in the file. These requirements can be fulfilled with direct access storage devices (DASD equipment). DASD includes hard disks, floppy disks and several forms of optical disks.

Memory devices which always remain connected to a computer system are called on-line devices. Hard disks are on-line secondary memory. The devices that can be connected to the system when needed are known as off-line memory. Magnetic tape is an example of off-line memory.

Memory Management

In a multiuser, multitasking or multiprogramming system, memory must be specifically managed to handle multiple programs. The physical size of the main memory is usually not large enough to accommodate the operating system and all of the application programs which are needed to execute the programs of various users. In a multiuser system users should not interfere with one another, and also they should not interfere with the operating system. This is achieved by providing suitable memory management scheme. Memory management can be provided totally by the operating system or with the help of hardware called MMU (memory management unit).

In a uniprogramming system, the main memory is partitioned into two portions: one portion for the operating system and the other portion for the program currently being executed. In a multiprogramming system the user's portion of the memory must be further subdivided to accommodate multiple tasks. The task of subdivision is done dynamically by the memory management scheme. Modern MMUs provide virtual memory to handle large program or a large number of programs. This is achieved by using swapping technique.

Memory Devices. There are three types of memories from technology point of view: semiconductor, magnetic and optical memory. Semiconductor memory is static, faster, lighter, smaller in size and consumes less power. It is used as main memory of a computer. Magnetic memory is slower but cheaper than semiconductor memory. It is used as secondary and back up memory of a computer for mass storage of information. RAMs, ROMs, EPROMs, flash memory etc. are semiconductor memories, and hard disks, floppy disks and magnetic tapes are magnetic memories. Optical disks and tapes are used as mass storage and back up memory.

Semiconductor Memory

Semiconductor memories are of two types: RAM (random access memory) and ROM (read only memory). RAM is a read/write memory. Information can be written into and read from a RAM. It is a volatile memory. It stores information so long as power supply is on. When power supply goes off or interrupted the stored information in the RAM is lost. ROM is a permanent type memory. Its contents are not lost when power supply goes off. The user cannot write into a ROM. Its contents are decided by the manufacturer and written at the time of manufacture. RAMs up to 1 Gbits capacity are available. ROMs store permanent programs and other types of information which are needed by the computer to execute user's programs.

Programmable ROMs are also available. They are called PROMs. Further, different types of PROMs such as erasable PROM called EPROM, electrically erasable PROM called E² PROM are available. User can write permanent information in PROMs. Such information is required while executing user's programs. Flash memory which is electrically erasable and programmable, is available. It is similar to EEPROM, but has higher packing density, lower cost and greater reliability.

Magnetic Memory

Magnetic memories are nonvolatile memory. They store information permanently. They are slower than semiconductor memory. The commonly used magnetic memories are of three types: hard disks, floppy disks and tapes. These devices are bulk storage devices. They are used to store information at a lower cost compared to semiconductor devices. These are not static devices. They are rotated while reading or writing information.

Floppy Disks. These are thin circular plastic disks coated with magnetic material (iron oxide or barium ferrite) on the surface. They are used as backup memory. The capacity of a 3.5 inch floppy is 1.44 MB. The use of floppy disks is diminishing day by day. Now people prefer to use optical disks. Floppy disks are cheaper than optical disks.

Hard Disks. Hard disks are made of aluminium or other metal or metal alloy which are coated on both sides with magnetic material usually iron oxide. Unlike floppy disks, hard disks are not removable from the computer. To increase the storing capacity several disks are packed together and mounted on a common drive to form a **disk pack**. A disk is also called **platter**. The disks unit packed in a sealed container is called **Winchester** disk drive. As the sealed containers are dust-free, they allow very high speed, usually 7200 rpm-15,000 rpm. A hard disk is more stable as it is rigid and contained in dust-free environment. Its track and bit densities are much higher than those of floppy disks. A hard disk may have more than 10,000 tracks per surface and bit density 15,000 bits per inch of a track. The data transfer rate is 33.3-700 MB/sec. The average access time is about 5-10 ms. The reliability of data is $1 \text{ in } 10^{11}$ which is much better than that of floppy disk, that is $1 \text{ in } 10^8\text{--}10^{10}$. Hard disks come in 2.5 inch and 3.5 inch diameter. The storing capacity per disk is upto 133 GB. The capacity of hard disk drive unit is upto 600 GB. A hard disk unit contains more than one platter.

Hard disk controllers are used to interface hard disks to a processor. An example of hard disk controller is Intel 82064. There are two types of hard disk controllers: IDE (Integrated Drive Electronics) and SCSI (Small Computer System Interface). SCSI are intelligent controller and they connect a number of I/O devices such as hard disks, floppy disks, tape drive, optical disks, printers, scanners etc. It is costlier than IDE controller. Actually, SCSI and IDE are not controllers; they are adapters. But people call them controllers. SATA (Serial ATA Interface) is now available for hard disk drives. Its data transfer rate is 3GB/s.

Magnetic Tape. Magnetic tape is a mass storage device. It is used as back up storage. It is serial access type storage device. Its main disadvantage is that it stores information sequentially. It is made up of plastic material. Standard sizes are 1/2 inch, 1/4 inch, 8 mm and 3 mm wide. Earlier, tapes used 9 tracks to store a byte with parity bit. Today tapes use 18 or 36 tracks to store a word or double word with parity bits. Newer tape is packed in cassette form which is called cartridge tape. The storing capacity is 2 GB-800 GB of compressed data. The data density of 18-track tape is about 40,000 characters per inch.

Optical Memory. Information is written to or read from an optical disk or tape using laser beam. Optical memory is used as archival and backup memory. Optical disks are not suitable for secondary memory because their access time is more than that of hard disks. Their advantage is that they have very high storage capacity. Types of optical memory are: CD-ROM, CD-R (CD Recordable), CD-RW, DVD-ROM, DVD-R and DVD-RW. CD-ROM is also called compact disk ROM. Information on CD-ROM is written at the time of manufacture. It is a read-only type memory. Disk size is 5.25 inch diameter. 650MB CD-ROMs are available. Their access time is 80 ms. Data transfer rate is 4800KB/s. A typical value of track density is 16000 tracks per inch.

CD-R/W (Read/Write) of 700 MB capacity are now available.

A DVD-ROM is similar to CD-ROM. It uses shorter wavelength of laser beam and hence, stores much more data than CD-ROM. DVD-ROMs of capacity 4.7 GB to 50 GB are now available.

1.3.3 Input Devices

Information is entered into a computer through input devices. An input device converts input information into suitable binary form acceptable to a computer. The commonly used input device is a keyboard. Several input devices which do not require typing of input information have been developed, for example, mouse, joystick, light pen, graphic tablet, touch screen and trackballs. Each of these allows users to select one of the items or images displayed on the screen. Therefore, these devices are called *pointing devices*. The required input is fed to the computer when control button is pressed. In industrial control electrical signals representing physical or electrical quantities such as temperature, pressure, force, current, voltage, frequency, etc. are entered a computer for their measurement and control. The sensors, transducers and data acquisition system act as input devices. Nowadays voice input systems have also been developed. A microphone is used as an input device. In many applications, computers with vision are required, for example, robots, computer-based security system, etc. The input systems for this type of computers use optical system, semiconductor devices sensitive to light, devices based on ultrasonic waves, etc. Such input devices produce digital signals corresponding to images, pictures etc. A multimedia computer accepts input in the form of text, images, graphics and voice.

1.3.4 Output Devices

The output devices receive results and other information from the computer and provide them to users. The computer sends information to an output device in the binary form. An output device converts it into a suitable form convenient to users such as printed form, display on a screen, voice output, etc. In some applications the computer's output may also be converted by an output unit in the form which can be used as an input to other devices, equipment, machines, etc. This is particularly true in industrial applications. The commonly used output devices are CRT screen and printers. Other output devices are LEDs (light emitting diodes), LCDs (liquid crystal displays), plasma displays, plotters, microfilm, microfiche, speaker or telephone system, etc.

The display screen is also called monitor or CRT (Cathode Ray Tube) display. Two types of display units are available: monochrome and colour monitor. Monochrome monitor displays texts in a single colour: blue, white, yellow or amber. A colour monitor displays text or graphics in multicolour. It may be desired in art/graphics applications. For graphics display, screens of higher resolutions are required. To provide higher resolution, screens contain more number of pixels to display text or images.

1.3.5 Buses

Memory and I/O devices are connected to the CPU through a group of lines called a *bus*. These lines are meant to carry information. There are three types of buses: address bus, data bus and control bus. An address bus carries the address of a memory location or an I/O device that the CPU wants to access. The address bus is unidirectional. The data and control buses are bidirectional because the data can flow in either direction; from CPU to memory, (or I/O device) or from memory (or I/O device) to the CPU. Examples of control signals are: \overline{RD} , \overline{WR} , ALE, etc. Fig. 1.4 shows the schematic diagram of I/O or memory connection to CPU.

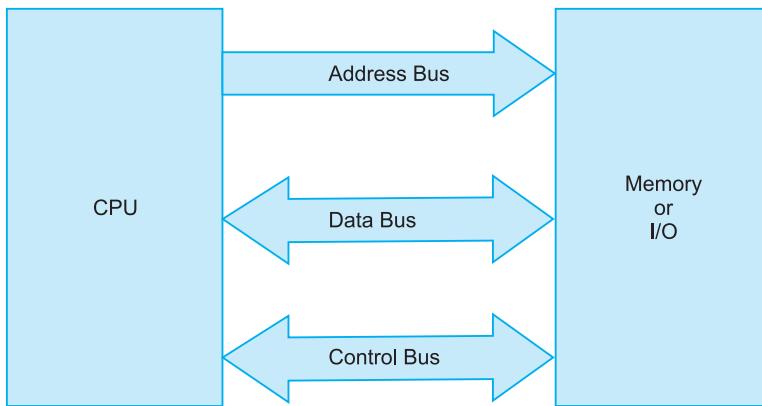


Fig. 1.4 I/O or Memory Connection to CPU.

Different types of bus architectures such as ISA, PCI, AGP, USB etc. have been described in Chapter 10.

1.4 MEMORY ADDRESSING CAPABILITY OF A CPU

The memory addressing capability of a CPU depends on the number of lines available in an address bus, that is, width of the address bus. With n -bit wide address bus a CPU can directly address up to 2^n memory locations. For example, a CPU with 20 bit wide address bus will address 2^{20} memory locations directly. $2^{20} = 1$ million. One memory location stores one byte of information and hence a CPU with 20-bit address bus will directly address up to 1 MB memory, with 16-bit address bus 64 KB memory, with 24-bit address bus 16 MB memory and with 32-bit address bus 4 GB memory.

It can be very easily shown that with n address lines, 2^n memory locations can be addressed. First take a simple case of only 2 lines. A line carries either 0 or 1 binary digit. Using two lines only, possible memory addresses are: 00, 01, 10 and 11. These are $2^2 = 4$. If there are 3 lines, the possible addresses will be 000, 001, 010, 011, 111 which are $2^3 = 8$. Similarly, with four address lines possible addresses are: 0000, 0001, 0010, 0011, 0100,, 1111 which come out to be $2^4 = 16$. Similarly, with n address lines up to 2^n memory locations can be addressed.

1.5 WORD LENGTH OF A COMPUTER

A digital computer operates on binary digits, 0 and 1. It can understand information only in terms of 0s and 1s. As already mentioned a binary digit is called a *bit*. The word bit is the short form of *binary digit*. A group of 8 bits is called a *byte*. The number of bits that a computer can process at a time in parallel is called its *word length*. The commonly used word lengths are: 8, 16, 32 or 64 bits. It is a measure of the computing power of a computer. Computer with longer word length are more powerful. When we talk of a 32-bit computer, it means that its word length is 32 bits. Similarly when we say 8-bit, 16-bit or 32-bit microprocessor, 8-bit, 16-bit or 32-bit indicates the word length of the microprocessor.

1.6 PROCESSING SPEED OF A MICROPROCESSOR

The processing speed of a microprocessor is usually measured in millions of instructions per second. In short it is written as **MIPS**. In computer literature term **throughput** is also used for the number of instructions executed per second. The MIPS rating is used to specify the integer computation performance of a processor. The processing speed of a microprocessor for floating-point computation is measured in millions of floating-point instructions per second, **MFLOPS**. As the instructions differ from microprocessor to microprocessor for a high-level language program, the MIPS rating does not give correct idea of processing speed of a microprocessor. If the same program runs on a RISC as well as on a CISC processor, the MIPS rating of the RISC processor will be higher because the instructions of a RISC processor are simpler than those of a CISC processor.

Further, to compare processors with different clock cycles and different instruction sets is not totally correct. Today **SPEC** ratings are widely used to specify processor's performance. SPEC is the abbreviation of System Performance Evaluation Committee. This committee was formed in 1989 to develop industry-standard benchmark to evaluate processor's rating. SPEC ratings are given in SPECint95 to measure integer performance, and SPECfp95 to measure floating-point performance. The 'int' stands for integer, 'fp' for floating-point, 95 is the year in which this standard was developed. SPECint95 is written in C language, and SPECfp95 in FORTRAN.

Some other ratings are as follows

TPS (Transactions Per Second). It is used for on-line processing application of a computer. On-line applications demand rapid interactive processing for large number of relative simple transactions. Each transaction may involve a database search, query, answering, and database update operations. They are supported by very large databases. Examples are: railways reservations, airlines reservations, automated teller machines, etc.

KLIPS. For a knowledge-based computer, performance can be measured in kilo logical inferences per second (KLIPS).

iCOMP. It is Intel's Comparative Microprocessor Performance. It consists of a collection of benchmarks to evaluate an index of relative performance of Intel microprocessors.

LINPACK Rating. It uses FORTRAN programs for solving linear system of equations of the order of 100 and higher. Its programs contain high percentage of floating-point operations. It is very sensitive to vector operation and the degree of vectorization by the computer. Hence, it is given with specific compiler and degree of linear equations. It is measured in MFLOPS or GFLOPS.

Dhrystone. It is synthetic testing benchmark. It gives integer performance. Its unit is Kdhrystone per second. Its disadvantage is that it is sensitive to compilers.

Whetstone. It is a FORTRAN based synthetic testing benchmark. It measures both integer and floating-point performance. Its programs take into account array indexing, subroutine calls, parameter passing, conditional branching, and trigonometric/transcendental functions. Its unit is KWhetstone per second. It is sensitive to compilers. Whetstone tests do not perform I/O or system calls.

1.7 MICROPROCESSORS

With the advances in LSI and VLSI technology it became possible to build the whole CPU of a digital computer on a single IC. A CPU built on a single LSI, VLSI or ULSI chip

is called a *microprocessor*. It is the latest development in the field of computer technology as well as semiconductor technology. A digital computer has a microprocessor as its CPU. A microprocessor combined with memory, an input device and an output device forms a microcomputer. The CPU of a large computer contains a number of microprocessors. Each microprocessor performs a specified task within the CPU. The microprocessors in the CPU of a large computer operate in parallel.

Table 1.1 Important Intel Microprocessors

<i>Microprocessor</i>	<i>Year of Introduction</i>	<i>Word Length</i>	<i>Memory Addressing Capacity</i>	<i>Pins</i>	<i>Clock</i>	<i>Remarks</i>
4004	1971	4-bit	1 KB	16	750KHz	First microprocessor
8085	1976	8-bit	64KB	40	3-6MHz	Popular 8-bit microprocessor
8086	1978	16-bit	1 MB	40	5-10MHz	
8088	1980	16-bit	1 MB	40	5-8 MHz	Data bus 8-bit, Internal architecture 16-bit, widely used in PC XT.
80286	1982	16-bit	16MB real, 4GB Virtual	68	6-12.5MHz	Widely used in PC/AT
80386	1985	32-bit	4GB real, 64 TB Virtual	100	20MHz	Popular 32-bit Microprocessor
80486	1989	32-bit	4GB real, 64TB Virtual	168 (17×17)	25-100MHz	Improved 32-bit processor; contains FPU and cache on the chip.
Pentium	1993	32-bit	4GB real	237PGA	233MHz	Contains 2 ALUs, data bus 64-bit, address bus 32-bit.
Pentium Pro	1995	32-bit	64GB real	387 pin PGA	150-200MHz	Data flow architecture, contains 2nd-level cache, operates at 3.3V.
Pentium II		32-bit	64GB real		450MHz	Pentium Pro with MMX technology
Celeron	1998	32-bit			2.6MHz	Cheaper 32-bit processor, based on Pentium Pro core.
Pentium III	1999	32-bit	64GB real	370PGA	500-1000MHz	pentium II + 70 multi-media instructions.
Pentium 4	2000	32-bit	64 GB	423PGA	1.3-3.2 GHZ	
Pentium 4EE and Pentium 6XX series	2004	64-bit		-	3-3.7 GHZ	
Itanium	2001	64-bit		423PGA	-	EPIC Processor

Nowadays microprocessors also perform tasks other than those of a CPU. A number of microprocessors are also used to control input and output devices of a large computer. For example, a microprocessor is used to control the operation of a keyboard and CRT display unit. It is used to control the operation of a printer and so on.

The first microprocessor, Intel 4004, a 4-bit microprocessor, was introduced in 1971 by Intel Corporation. In 1972 Intel introduced the first 8-bit microprocessor, Intel 8008. These microprocessors used PMOS technology. A more powerful and faster microprocessor, the Intel 8080, using NMOS technology was introduced in 1973. The 8-bit microprocessors were introduced by a number of companies; examples are: Motorola's MC 6809, Zilog's Z80 and Z800, MOS Technology's 6500 series, National Semiconductor's NSC 800 etc. The latest 8-bit microprocessor of Intel is 8085 introduced in 1976. It is very popular and widely used. The first Indian 8-bit microprocessor was SCL 6502, manufactured by Semiconductor Complex Ltd. 8-bit microprocessors were soon followed by 16-bit microprocessors. Examples of 16-bit microprocessors are: Intel 8086, 80186 and 80286; Motorola's 68000, 68010, 68012; Texas Instrument's TMS 9900, Fairchild 9440, Digital Equipment's LSI 11 and so on. In the 1980s, 32-bit microprocessors were introduced, and they are still widely used. Examples of 32 bit microprocessors are Intel 80386, 80486, Pentium, Pentium Pro, Pentium II, Pentium III, Celeron and Pentium 4.

Pentium Pro, Pentium II, Pentium III and Pentium 4 use data flow architecture. Earlier, Intel's 4004 to Pentium were Von Neumann type processors. Pentium III is an improved version of Pentium II. It includes MMX pipeline to provide MMX features. Furthermore, it contains Internet Streaming SIMD instructions to enhance multimedia performance on the Internet such as streaming audio and video, animation, 3-D simulation, advanced imaging, speech recognition etc. Celeron processor is a low-cost 32-bit processor. It includes MMX features and Internet streaming SIMD instructions. Pentium 4 is an improved version of Pentium III. It contains more Internet Streaming SIMD instructions and it is faster than Pentium III. Pentium M is low-cost processor for notebook computers. 32-bit processors of other companies are: Motorola's 68020, 68030, 68040 and 68060, Power PC 601, 603, 604, 740 and 750, National Semiconductor's NS 32032, NS 32332 NS32C532 and M300, AMD's K5, K6 and Athlon (K7), Cyrix 586 and 686 etc.

Pentium 4 EE (Extreme Edition) and Pentium 4 6XX series are 64-bit processors. Pentium 4 EE 840 is a dual core processor suitable for servers. Itanium is a 64-bit processor of Intel Corporation. 64-bit processors of other companies are: AMD's Athlon 64, Athlon 64 FX series, Opteron, Athlon 64X2; PowerPC 620, PowerPC G4, IBM's G5, SUN's Ultra SPARC III, Compaq's Alpha 21264, MIPS 12000; C-DAC's Param 10,000 and Param Padma, HP's PA 8500 series etc. AMD's Opteron and Athlon 64X2 are dual core processors.

Table 1.2 Important Microprocessors of Companies other than Intel

Microprocessor	Make	Year of Introduction	Word length	Clock	Number of Transistors	Remarks
6809	Motorola	1979	8-bit	4-8 MHz		Popular 8-bit microprocessor
68000	Motorola	1979	32-bit internal architecture, 16-bit data bus	10-25 MHz	70,000	Popular and widely used

(Contd.)...

68040	Motorola	1989	32-bit	20-33 MHz	1.2 million	Contains FPU, MMU, on-chip data cache and instruction cache
Power PC 601	Motorola, IBM and Apple	1993	32-bit	120 MHz	2.8 million	RISC processor
Power PC 750	Motorola, IBM and Apple	1997	32-bit	400 MHz		Suitable for notebooks, mobile and desktop
K6-3, Athlon	AMD	1999	32-bit	500 MHz		Contains 2nd level and 3rd level cache. K7 is called Athlon
586, 686 and 6X86MX	Cyrix		32-bit	233 MHz		6X86MX outperforms Pentium II
Power PC 620	Motorola, IBM and Apple		64-bit	250 MHz	7×10^6	Suitable for workstation
Alpha 21164, 21264	DEC Compaq		64-bit	700 MHz-1000MHz for 21264	9.3×10^6 (21164)	RISC processor.
ULTRA-SPARC	SUN		64-bit	200 MHz	3.8×10^6	RISC processor
MIPS 10,000, 12000	MIPS	1999	64-bit	300 MHz	6.4×10^6 (MIPS 10000)	RISC processor. Suitable for workstations.
PA 8500	H.P.	1999	64-bit	440 MHz		
68060	Motorola	1995	32-bit	-	-	Suitable for embedded application
Athlon 64, Athlon 64 FX	AMD	2003	64-bit	2.66 GHZ	-	0.13 micron process technology
Optiron, Athlon 64X2	AMD	2005	64-bit	2.4 GHZ	-	Dual-core processors

1.8 SINGLE-CHIP MICROCOMPUTERS (MICROCONTROLLERS)

With the development of VLSI technology it became possible to fabricate a digital computer on a single IC chip. A digital computer fabricated on a single IC chip is called **single-chip microcomputer**. Since it is widely used for control application, it is also called **microcontroller**. It is very small and compact. It forms the part of the device or equipment which is to be controlled. It is used for industrial control, process control, consumer and appliances control, instrumentation, etc. It contains a CPU, memory (RAM, and ROM/EPROM/Flash memory) and I/O lines. A powerful microcontroller may contain some other components

which are needed for control applications such as analog-to-digital converter, digital-to-analog converter, interrupt controller, DMA controller, wave generator, etc.

Intel developed 8-bit microcontrollers 8048 series, in 1976. In 1980 improved 8-bit microcontrollers 8051 series was developed . 8-bit microcontrollers are used for simple and low-cost control applications. In 1983, Intel developed 16-bit microcontrollers, 8096 series. Later on, it developed 80196 series of 16-bit microcontrollers. These were more powerful and were used in sophisticated industrial control, intelligent computer peripherals, instrumentation, etc.

Other manufacturers also developed 4-bit, 8-bit, 16-bit and 32-bit microcontrollers. Motorola developed 32-bit microcontrollers, MPC-505. IBM developed 32-bit microcontrollers, 403GA. 32-bit microcontrollers are used for complex control applications. See more detail in Chapter 7.

1.9 COMPUTER CLASSIFICATION

Modern computers are classified as follows:

- (i) Palmtop computers, also known as palm PCs or PDA (Personal Digital Assistant).
- (ii) Notebook computers, also known as laptop computers. Some trade names of notebook computers are: IBM's Thinkpad 570, Compaq's Armada -E700, M700, M300 series, H.P.'s Omnibook, Siemen's Scenic mobile 750 AGB (high-end multimedia notebook), Apple's iBook, etc.
- (iii) Desktop Computers
- (iv) Workstations
- (v) Servers
- (vi) Super Computers

Palmtop or Handheld Computers. These are the smallest computers available. They can be held in palm and hence, they are called palmtop computers. They can easily be kept in a shirt pocket. They are used for tracking appointments, maintaining lists, jotting notes, etc. They use tiny keyboard and have small disk memory. They can be connected to wireless network. Some palmtops use touch screen. Palmtops also accept handwritten inputs using an electronic pen which can be used to write on the palmtop screen. The system has to be trained on the user's handwriting before it can be used. A palmtop can be used as a mobile phone, fax and E-mail. Some palmtops use a proprietary operating system, but it can swap data with a WINDOWS PC. Some palmtops use Microsoft's WINDOWS-CE operating system. WINDOWS-CE is a stripped-down version of Microsoft WINDOWS. It is designed to provide a WINDOWS interface for palm PC's, some other types of very small computers, and tools and appliances other than PCs.

Notebook PCs or Notebook Computers (Laptop Computers). These are portable computers. They contain 32-bit CPU, hard disks, floppy disks, CD-ROM drive, modem and flat LCD screen. Colour displays are available. They consume less power and use batteries for their operation. They are used for word processing and spreadsheet computing while a person is travelling. They can be connected to computer network. Wireless connection can be provided to laptop computers so that they can get information from large stationary computers. They generally use WINDOWS-XP operating system. Notebook PC with LINUX operating system has also been developed. Hard disk capacity up to 60 GB, RAM capacity up to 256 MB,

etc. are available on a notebook. Multimedia system is now available. Some notebook computers are provided with finger recognition system.

Centrino Notebooks. The notebooks which are manufactured using Intel's Pentium M processor, Intel's mobile chipset (the 855 GME or 915 M) and Intel PROSet wireless LAN card, are said to have **Centrino technology**.

Centrino Duo Mobile Technology (NAPA)

Intel India has launched Centrino Duo Mobile technology in January, 2006. Its code name is NAPA. Centrino Duo is the next generation of mobile computing platform. It consists of the following three parts:

- (i) **Intel Core Duo Processor Called Yonah.** It uses 65 nm (nanometer) process technology. It is a dual core processor designed for mobile computers. It has on-die 32 KB instruction cache and 32 KB data cache. A 2 MB L2 cache is shared by two cores using the Intel Smart Cache technology which allows dynamic allocation of cache to the cores depending on processing load on the cores.
- (ii) **Mobile Intel 945 Express chipset family.** It gives dual channel DDR2 667 support. This chipset has two variants-945 GM and 945 PM. The 945 PM is pure performance chipset which consumes less power. The 945 GM offers Intel Media Accelerator 950 to deliver improved graphics performance.
- (iii) **Intel PRO/Wireless 3945ABG Network Connection.** It is a smarter, smaller and sleeker wireless solution. It is compatible with the latest 802.11e standard.

Desktop Computers. These computers are single-user personal computers (PCs) and can be placed on a desk and hence, they are called desktop computers. They use 32-bit processors such as Pentium 4, Celeron, Athlon-XP etc. The hard disk capacity of 80GB and RAM capacity of 512 MB are used. Optical disks and 3.5 inch floppy disks are used as backup memory. Operating system used are: WINDOWS-XP, WINDOWS-NT, LINUX, Mac OS-X etc. Some desktop computers use fingerprint recognition system to provide better security. Such a system falls under "Biometry". For finger recognition, the user has to slide his fingers across a scanner, which senses the patterns caused by the ridges and furrows on the fingertip.

Workstation. Workstations are more powerful computers than desktop computers. They are suitable for numeric and graphic intensive applications. They are used in scientific and engineering applications such as computer aided design (CAD), simulation etc., which require greater processing power, larger storage capacity and better graphics capability. They have longer colour video display unit (monitor of 19 inch or more). They have hard disk and RAM capacity more than those of a desktop computer. The RAM capacity may be of a few GB and hard disk capacity of a few hundred GB. They use RISC processors such as SUN's UltraSPARC III, HP's PA-8500 or 8800, Compaq's Alpha 21264, MIPS 12000, etc. Operating systems used are multiuser such as UNIX, SUN's Solaris, HP's HP-UX etc.

Servers. These are powerful computers. A number of PCs and terminals can be connected to a server. Servers are provided with large disk and RAM capacity. In a low-end server only one microprocessor is used to act as a CPU. On the other hand in a high-end server, a number of microprocessors are provided in the CPU. Microprocessors in a multiprocessor CPU operate in parallel. The user working on a PC connected to a server, makes simple computation on his own PC, but for more complex computation he can connect his PC to the server through the LAN, WAN or Internet. He can utilize computing power, all facilities and database available with the server. He can also avail the facilities available at other PCs

connected to the server. The computer connected to the server through a network is called **client**. The aforesaid type of computing is known as client-server computing. Recently multicore 64-bit processors have been developed by many companies. These are quite suitable for server computers. Servers use multiuser operating systems such as UNIX, LINUX, HP-UX, SUN's Solaris etc.

Super Computers. These are the most powerful computers. They are used for very complex computation work. They use vector processors. Intensive parallelism is used in supercomputers. A number of RISC microprocessors are used in the CPU of a supercomputer. Supercomputers are used for weather forecasting, in aerodynamics, seismology; atomic, plasma and nuclear analysis; for weapons research and development, sending rockets into space, etc. In some applications in aerodynamics and nuclear physics, as many as 10^{13} arithmetic operations are needed for a single problem. This may take a number of hours of computing time on a supercomputer. Hence, there is a constant demand to increase the power of a supercomputer. Examples of supercomputers are: Cray-1 (1976), Cray 2 (1985), Cray T3D (1993), NEC's SX-S/44 (1991), Fujitsu VP 2600/10 (1991), Hitachi 820/80 (1987), C-DAC's PARAM series of supercomputers etc. Current world's top supercomputers are: BlueGene/L DD2 Beta-system of IBM which has a speed of 70.7 TFLOPS and uses 400 PowerPC processors; Columbia (NASA) which has a speed of 51.9 TFLOPS and uses 10,240 Itanium 2 processors; Earth Simulator (NEC) with a speed of 40 TFLOPS and uses 5120 NEC CPUs; etc. C-DAC's PARAM 10,000 uses 160 UltraSPARC III processors. Its computing power is 100 GFLOPS. It has open frame architecture and can scale upto TFLOPS level. It uses C-DAC's own designed communication processor and network. C-DAC's PARAM Padma supercomputer has a peak computing power of 1 TFLOPS and 5 terabytes of storage. It is powered by 248 IBM's Power 4 RISC processors of 1 GHZ clock frequency. It runs AIX 5.1L operating system. Its primary interconnect is ParamNet-II, 2.5 GFLOPS, full duplex with backup gigabyte Ethernet network. Its parallel programming is done through C-DAC HPCC software. It ranked 171 in June 2003 list of top 500 supercomputers in the world. Other Indian supercomputers are: C-Dot's (Center for Development of Telematics) Chip-152, National Aeronautical Laboratory's Mark-3, Bhabha Atomic Research Centre's (BARC) Anupam; PACE series of supercomputers of Defence Research and Development Organization (DRDO), Hyderabad, etc.

Earlier Classification. Earlier, computers were classified as microcomputers, minicomputers, mainframe (or large) computers and supercomputers. This classification is no longer used. Microcomputers are low-cost small computers. They include portable computers, personal computers i.e., PCs (single-user desktop computers), computers for dedicated applications like industrial control, instrumentation, appliance control etc. Minicomputers are more powerful multiuser computers. They contain more RAM and hard disk capacity compared to microcomputers. High-end minicomputers contain more than one microprocessor in their CPU. Large or mainframe computers are more powerful than minicomputers. They use very large capacity of RAM and hard disk. They use UNIX and other multiuser operating system. Examples of mainframe computers are: IBM's ES-9000, DEC's VAX-9000, CDC Cyber-2000 V, etc. Supercomputers have already been described in new classification.

1.9.1 Computer Pen

It is a computer housed within a pen. It can function as a diary, alarm, notetaker and can receive E-mail and pager messages. Unlike a conventional handheld computer, it has no keyboard. It does not need a screen to work. It saves in its memory what user is writing. It links to a printer, mobile phone, modem or PC allowing handwritten notes to be transmitted.

It has been developed by BI, a research laboratory in U.K. The prototype is called SmartQuill. It has ability to record handwriting not only on paper but also on any flat surface-horizontal or vertical. A tiny light at the tip allows writing in dark. SmartQuill can even translate invisible writing in the air.

1.10 USER INTERFACE

User interface provides communication means between an user and the computer. There are two types of user interface : text/typing type and icon/mouse type. In text/typing type user interface, the user has to type commands using a keyboard. In DOS operating system the user has to type commands. It provides text/typing type user interface. In icon (small graphical symbol) type user interface, the user tells the computer to carry out certain commands by pointing to an icon. Such user interface is also called **GUI** (Graphical User Interface). A pointing device such as mouse can be used to point to an icon. Such system also provides a list of available commands called a **menu**. The user can point to a command in the menu. Today GUI is commonly used by most of the operating systems.

1.11 HARDWARE, SOFTWARE, FIRMWARE, MIDDLEWARE AND FREEWARE

The physical components of a computer are called *hardware*. A physical component may be electronic, electrical, magnetic, mechanical or optical. Examples of hardware are microprocessors and other ICs, hard disks, floppy disks, optical disks, cathode ray tube (CRT), keyboard, printer, plotter, etc.

A sequence of instructions given to a computer to perform a particular task is called a *program*. A set of programs written for a computer is called *software*. The software required to execute user's program is known as *system software*. The term software includes both system software and user's programs. The system software includes operating system, assembler, compiler, interpreter, debugging programs, text editors, etc. The *operating system* is a collection of programs which controls the overall operation of a computer. The programs stored in ROMs, PROMs, EPROMs or Flash memory are called *firmware*. Nowadays a large variety of prewritten programs are available to solve specific tasks. Users need not prepare programs for such tasks. They should simply know how to use such prewritten programs. Prewritten programs for specific tasks are called *application programs* or *application packages*. Important application packages available are WordStar and MS-Word for text manipulation, LOTUS 1-2-3 and MS-Excel for preparation of spreadsheet; MS-ACCESS, ORACLE, UNIFY and FOXBASE for handling database, etc. MS-Office is an integrated package. It includes a word processing package-Word, a spreadsheet package-Excel, a database management package-Access, a presentation package-PowerPoint, and a Scheduling and Organization package-Outlook. Software package for designing buildings, structures, power systems, inventory control, accounting, dealing with projects, etc. are available.

Middleware. It is software that operates at the level between an application program and a network. It can mediate the interaction between separate applications across heterogeneous computing platforms on a network.

Some software are available free of cost. They can be downloaded from Internet. Such software are called **freeware**.

1.11.1 Operating System

An operating system is a collection of programs which controls the overall operation of a computer. It manages files on a disk. It permits users to create, print, copy, delete, read, write to files. It formats disks and controls input and output devices. It executes programs, allocates memory space to users, schedules jobs, provides user interface to computers, prevents interference between users in a multiuser system, it processes user's commands, and so on and so forth.

Several operating systems have been developed over the years. Some important operating systems which are for single-user systems are: MS-DOS, OS/2, WINDOWS-98, WINDOWS-XP etc. At present WINDOWS-XP is commonly used. Operating systems for multiuser system are: UNIX, LINUX, Novell Netware, SUN's Solaris, WINDOWS-2003, Mac OS X(ten), etc.

1.11.2 Programming Languages

A computer understands information composed of only zeros and ones. A program written in terms of 0s and 1s is called a *machine language program*. Computer instructions are written in binary codes. A machine language uses only binary codes. The writing of programs in machine language is a very difficult, tiresome and very boring job. Moreover, it is errorprone. To overcome this difficulty a program can be written in alphanumeric symbols instead of 0s and 1s. Meaningful symbols called *mnemonics* are used for this purpose. For example, ADD is used for addition, SUB for subtraction, CMP for comparison, etc. A language which uses mnemonics is called an *assembly language*. A program written in an assembly language is called an assembly language program.

An *instruction* is a command given to a computer to perform specified task. The instruction set of a processor is the collection of instructions that the processor is designed to execute. In assembly language a mnemonic is an instruction. Instructions are classified into groups like data transfer, arithmetic, logical, branch control, and I/O and machine control instructions.. Data transfer group includes instructions for transferring data from register to register, register to memory and memory to register. Instruction of arithmetic group perform addition, subtraction, multiplication, division, etc. on data placed in a register or memory. Logical group instructions perform AND, OR, EX-OR, comparison, rotate, etc. operations on the contents of registers. Instructions of branch control group perform conditional and unconditional jumps, subroutine call and return, and restart. I/O and machine control group instructions perform input, output, stack and machine control operations.

When a program is written in a language other than machine language, the computer will not understand this. Therefore, a program written in other langauges must be translated into machine language before it is executed. The task of translation is done by software. A program which translates an assembly language program into a machine language program is called an *assembler*.

A language in which each statement or an instruction is directly translated into a single machine code is known as a *low-level* language. Each mnemonic of an assembly language has a unique machine code. An assembly language is a low-level language. A machine language is also a low-level language. An assembly language depends on the internal architecture of a processor. Each processor has its own assembly language. Assembly language of one processor cannot be used for another processor. In other words it is not *portable*. To write an assembly language program, a programmer must have the detailed knowledge of the instruction set of the particular processor, its internal architecture, registers, and connection of peripherals

to ports etc. It is not very fast and efficient programming language. To overcome the difficulties associated with assembly language, high-level (procedure-oriented/object-oriented) languages have been developed. In a high-level language an instruction is called *statement* rather than mnemonic. Statements more closely resemble English and Mathematics than mnemonics. High-level languages permit programmers to describe tasks in the forms which are problem oriented rather than computer oriented. Programming in a high-level language does not require precise knowledge of the architecture of a computer which is to be used. A program written in a high-level language will run on any computer which has a compiler for that language. In other words a high-level language is portable.

A program which translates a high-level language program into a machine language program is called a *compiler*. An *interpreter* is also a program which translates a high-level language program into machine language program. It reads one statement at a time, translates it into machine codes, executes it and then goes to the next statement of the program. On the other hand a compiler reads an entire program once or twice and then translates it. A compiler is faster and more powerful than an interpreter. A compiler is a larger program and occupies more memory space. It is costlier than interpreter.

Important high-level languages are: BASIC, FORTRAN, COBOL, PASCAL, C and C++ languages, PROLOG, JAVA etc. BASIC is a abbreviation for Beginners All-purpose Symbolic Instruction Code. It is a very simple and easy language for beginners. It is suitable for scientific calculation. FORTRAN stands for Formula Translation. It is a powerful language for scientific and engineering computations. COBOL stands for Common Business Oriented Language. It is suitable for business data processing. PASCAL is a multipurpose language suitable for both scientific and business applications. This language has been named in honour of Blaise Pascal, a great mathematician and inventor. PROLOG stands for Programming in LOGIC. It is suitable for artificial intelligence applications. It has been chosen for fifth generation computers. A large number of high-level languages have been developed. A particular language may be very efficient for a particular field. JAVA is an object oriented language. It is suitable for Internet applications.

1.12 BATCH PROCESSING, MULTIPROGRAMMING AND MULTIUSER SYSTEM

In a batch processing system a computer serves one user at a time. When the program of one user is completed, then only, another program is started. This type of system does not allow a user to interact with the computer. The large computers used in the 1960s, executed programs one by one using this method.

In multiprogramming several programs are processed by a computer simultaneously. Usually, a CPU is much faster as compared to I/O devices. While I/O devices are performing certain tasks the CPU may not be doing any task, it may be lying idle. To keep CPU busy for most of the time, it is desirable to process a number of programs concurrently. This is achieved by overlapping CPU and I/O operations when several programs are running simultaneously. The multiprogramming is an improvement over batch processing, but it still does not allow users to interact with the computer.

In a *multiuser system* a number of users can work simultaneously. It allows each user to interact with the computer. It is also known as *time-shared system*. A number of video terminals are connected to a computer. Users enter data at very slow rate, and computer processes data at very fast rate. Making use of this fact a computer serves several users

simultaneously by attending them in turn. It usually gives 20 milliseconds time to each user. Each user feels that he is using the computer exclusively because the computer processes his data as fast as he enters it. Each user can utilize the common resources such as high-speed printer, computer's memory, etc. A demerit of a multiuser system is that when computer is down all computing jobs are stopped. They can be resumed only when the computer is up and running again. Another demerit is that its response time to each user becomes unreasonably long when the number of users becomes more. Server-client type of computing is also a kind of system. Nowadays it is widely used. Time-shared type multiuser systems were used earlier.

In an industry a number of processes are controlled by a computer simultaneously. This is called *multitasking*. The term multitasking is more often used in real-time control in industry. When this term is used elsewhere in general sense, it simply means multiprogramming. The time-shared approach of multiuser system is not suitable for multitasking system. In a multiuser system all users are attended at approximately equal time intervals. In multitasking system for industrial control priority-based job scheduling is used. A high priority task can interrupt a low priority task.

1.13 MULTITHREADING

Multithreading. A thread is the smallest executable element of an application. Therefore, if an application has the ability to execute, at least two threads simultaneously, it will run twice as fast on a dual-core processor. So a dual-core is beneficial, if an application is multithreaded. But most applications which are available today are not multithreaded. However, some PC users run more applications at a time, so they will be benefited by using dual-core processor.

1.14 COMPUTER NETWORK

A processing complex consisting of two or more interconnected computers is called a *computer network*. When the computers of a computer network are within a building or campus, they are interconnected through a local area network (LAN). If computer units are situated at large distances, they are interconnected through telecommunication links. Satellites are also used for this purpose. A computer network spread over a wide area is known as WAN. The Internet is also a kind of WAN.

1.15 DISTRIBUTED PROCESSING (COMPUTING) OR MULTIPROCESSING

In a computer network each computer operates independently on separate tasks. When a number of users work on separate computers interconnected in a computer network, the processing technique is known as *distributed processing* (computing) or multiprocessing. The term multiprocessing is also used in case of a multiprocessor system (i.e., when a computer contains several processors). For distributed processing a large powerful central computer with large memory and high-speed printer is used in a computer network (LAN or WAN). A number of PCs or workstations are connected to the central computer (Server). Each user with a PC, performs his task locally in an independent manner. If he wants to use the server for complex computation or to access database available with the server, he can do so. In this system the user can select the local computer best suited to the local computational needs.

A company or organization also can use distributed computation technique. The company's work are done partwise at different computers which are connected through a LAN or WAN or Internet.

Disadvantages of a time-shared system is eliminated to a great extent in a distributed processing system. If the large computer is down, all the jobs are not stopped. Most of the tasks are carried out on the local microcomputer. The burden on the large computer is reduced to a great extent, as much of the computing work is done by local microcomputers.

1.16 LOCAL AREA NETWORK (LAN) AND WIDE AREA NETWORK (WAN)

In an industry, commercial organization or an office which has several computers in a building or a campus, need may arise to transmit data between computers. In such a situation a high-speed interconnecting network called *local area network* (LAN) is needed to interconnect the computers so that they can communicate with each other. There are various ways for interconnecting computers such as star, common-bus, ring type LAN, and so on. In a star type LAN there is a central controller that coordinates all communication between computers on the LAN. The common-bus type LAN also known as **Ethernet**, does not have any central controller. Rather, the control of the bus is spread among all the computers connected to the common-bus. The common-bus is simply a wire usually a co-axial cable to which any number of computers can be easily connected. Any computer can take over the bus to transmit data. Two computers are not allowed to transmit data at the same time. Also, in the ring type LAN, the control is distributed among all of the computers on the network. A binary code called *token* is passed around the ring from workstation to workstation. All workstations are linked to form a continuous loop. A computer which wants to transmit data must possess the token. It takes the token from the bus, to prevent any other computer from transmitting. After the transmission is completed, it puts the token on the ring so that another computer which has to transmit data can receive it.

LANs are classified according to their data transfer speed, as: high speed, medium-speed and low-speed LANs. In high-speed LANs data are transmitted at the rate a few gigabits per second (Mbps). Such LANs are designed to link server computers. In medium-speed LANs data are transmitted at the rate of about a gigabits per second. Such LANs are suitable to link smaller servers and PCs. Example of a medium-speed LAN is Ethernet. Its speed is 1 Gbits/s. Low-speed LANs transmit data at a few hundred Mbps. They are suitable to link PCs and other workstations. Wireless LAN, called WLAN has also been developed.

WAN. When computers/terminals are spread over a wide area, they can be connected through public or private communication system. This type of network is called *wide area network* (WAN). Internet is also a kind of WAN.

1.16.1 Internet

A worldwide computer network is called Internet. Any two computers on the Internet can communicate to each other. Each computer on the Internet has an address which is universally recognized throughout the network. Web, also called World Wide Web (WWW) is a collection of resources that one can access from anywhere in the world, over the Internet. These resources can provide textual documents, static graphic images, video clips, programs, database or any other kind of information. Web is just one of the many services available on

the Internet. The services available on the Internet other than Web are : E-mail, FTP, Mailing lists, Usenet news and articles, Chat sessions, etc. The WWW is a vast library of information in the field of art, science, engineering, commerce and so on. On Internet exchange of information takes place on client/server model. A **client** is a computer which receives information from the server. A **server** is a computer which gives information to other computers on the Internet. Servers and clients communicate through a protocol, **HTTP** (Hyper Text Transfer Protocol). The set of rules for information exchange between server and client is called **protocol**.

Important terminologies related to Internet are given below:

URL (Uniform Resource Locator). It is pronounced as earl. It is descriptive address for a Web page or any other file on the Internet. An example is <http://www.intel.com/Pentium III/>

This address is to obtain information regarding Pentium III microprocessor from Intel Corporation. A simple URL for a Web page includes (*i*) the name of the Web protocol (http), (*ii*) a colon, (*iii*) two forward slashes, (*iv*) a domain name and (*v*) a final slash. The domain name is the Internet name of the server which serves information.

HTML (Hyper Text Markup Language). It is a computer language to prepare Web pages. Hypertext is a text with extra features like formatting, images, multimedia and links to other documents. **Markup** is the process of adding extra symbols to ordinary text. Each symbol which is used in HTML has its own syntax, slang and rules. It is not a programming language. It is a markup language. It classifies the parts of a document according to their function. In other words it indicates which part is title, which part is a subheading, which part is the name of the author, and so on.

Web Site. Web server is known as Web site. It is the location at which pages are stored.

Web Page and Home Page. Web page is an information (document) written in HTML, which can be put on a Web server. It may give information about a person, a company, a group of people, an organization, a product, and so on. Web pages provide easy and efficient method for distributing information, files and softwares. **Home page** contains introductory information and/or master menu of the document.

Web Browser. The client software package called Web browser accesses the Web and contacts a server computer on the Internet, and exchanges information with the server. It understands how to communicate to a Web server through HTTP protocol, displays information and gives a way to represent hyper text links. The two most popular Web browsers are: Netscape's Navigator and Microsoft's Internet Explorer (IE). These Web browsers use graphical user interface. Other services can also be used from a Web browser such as Usenet, FTP, etc. The server computer is also provided with proper software to facilitate the exchange of information. Such a software is known as **Web server software**. Netscape's Navigator and Internet Explorer also allow user to use other services on the Internet, such as E-mail, to download files from FTP servers, read and post articles to Usenet newsgroups, etc.

Some other Web browsers are : Lynx (a text-only browser), Mosaic (the first graphical browser on which Navigator and IE are based), Web TV (runs on a television set), Amaya, UdiWWW, GNUScape, Opera, Arena, DOSLynx, etc.

Usenet News. It is forum for online discussion or exchange of information. Information may be on any topic. News client programs such as Microsoft's Internet News, Netscape News, etc. are available, which allow you to get news or information or article on any topic

from news servers through Usenet. Netscape News comes with Netscape Navigator. The Usenet service is available on Internet. NNTP (Network News Transfer Protocol) is used to distribute news on Internet.

FTP (File Transfer Protocol). It is a service used on Internet to exchange files. Files may be text file, programs or images. There are FTP servers. FTP client can access a FTP server.

Telnet. It is a protocol to connect users to database, library catalogs and other information resources on the Internet. WINDOWS-95 has telnet program which can be used to access telnet servers on the Internet. Neither Navigator nor IE include telnet.

Gopher. It is an Internet service that allows users to access gopher servers on the Internet. A few gopher servers still exist on the Internet, but it is almost extinct now.

TCP/IP (Transmission Control Protocol/Internet Protocol). TCP works with IP. They are simply different layers of control. They work on both LANs and WANs. They provide error checking, flow control (to prevent swamping of an user by another faster or more powerful user), and status and synchronization control. TCP is a transport layer. IP is a network layer which handles routing and delivery. To connect a computer to the Internet, TCP/IP is installed. Then ISP or online service is required to connect the computer to the Internet. TCP/IP includes standards for many common applications including E-mail, FTP, remote login, etc.

ISP (Internet Service Provider). In India Videsh Samachar Nigam Limited (VSNL) provides access to the Internet through the Gateway Internet Access (GIAS). Some other companies are also ISP, for example, Satyam, MTNL (Mahanager Telephone Nigam Ltd.), Bharti BT, etc.

Search Engine. A search engine is a program which looks through its database for information that matches your request. Information in the database are about Web sites and their contents. Examples of search engines are : Alta Vista, Yahoo, HotJava, Excite, Infoseek, AOL NetFind, Lycos, HotBot, LookSmart, SEARCH.COM, etc. Some Indian search engines are : Jadoo, Khoj, I Love India, 123 India, etc. Alta Vista is for Web and Usenet. Yahoo is for Web, Usenet, E-mail addresses, current news, people search, city maps and stocks. Yahoo is not a search engine. It has a huge list of Web sites, stored into categories. Yahoo provides links to search engines. HotBot is good to find site which uses a particular technology, such as JavaScript or VRML. HotJava is written in Java. It has been developed by SUN Microsystems. It is available for SPARC/Solaris platform as well as WINDOWS platform.

Web Crawler. It is a program that crawls through the Web and collects information regarding the Web sites. These information are put into the database of a search engine. Worms, Spiders or robots are the types of crawlers.

Packet. It is the basic data building block. A packet is self-contained data structure which can be sent over the network. It consists of two parts : the header and data. The header includes address information regarding the origination, destination and the type of the packet. The data is a block of data.

Applets. These are small Java application programs developed for Internet applications. One can develop new application programs using applets. Applets can be combined for an application using a scripting language JavaScript. Applets available at Web sites can be downloaded via Internet, and executed on the local computer.

Hub. It is central place to connect computers to a network, traditionally for a star topology. There are two types of hub: passive and smart. A passive hub simply connects the various nodes. A smart hub can be remotely controlled by the system operator. It can send back the information of packet traffic for analysis.

Bridge. A bridge connects networks (or two segments of LAN) that have the same Network Operating System (NOS), but different hardware.

Gateways. Gateways are similar to bridges except that they can translate protocol and convert data. They connect networks which operate at different network operating systems. The connected networks through a gateway may or may not have similar hardware. For example, a gateway can allow a Token Ring running IPX packet and an Ethernet running TCP/IP packet to communicate freely.

Routers. They control and direct network traffic. A network can be segmented into small networks using a router. The small networks are known as zones. If the packet's destination is in the same zone, the packet is not exposed to the other zones. This limits traffic across the entire network.

Intranet. A corporate or organizational network which uses the same protocols, (namely TCP/IP) as used on the Internet to share files and send E-mail, is called Intranet. An Intranet is cut off from the outside world. It permits its users to access the Internet, but it prevents outside access to internal files by hiding behind a security system known as a **firewall**. In many large organization intranet is a wide area network.

Extranet. When computers of many organization are connected through TCP/IP protocol, such a network is called Extranet.

1.17 SOME TERMINOLOGIES OF MOBILE PHONE STANDARDS AND DATA COMMUNICATION

Broadband. The minimum bandwidth for broadband must be 256 kbps (kilobits per second). Using broadband multiple channels can be sent over a single link. If there is insufficient bandwidth, then multiple services can not run reliably over the same connection (link). With broadband the user can go beyond Web browsing, E-mail and downloads; for example, now he can go for audio/video streaming to listen to songs or watch movies etc.

DSL (Digital Subscriber Line). It is a broadband technology. It uses existing telephone lines to transmit data. Its data transmission rate lies in the range of 1.5-8 mbps.

ADSL (Asymmetrical DSL). In ADSL the data flow is faster in one direction than in the other. Usually, download rates are higher than upload (outwards rates) rates.

DTH (Direct To Home). In DTH a dish antenna is used. One can get a number of channels on his TV, say 100 channels or more. Electricity consumption is 40 watts. It is cheap and compact. The diameter of the dish is 45-60 cm. It is based on digital technology. A set-top-box is also there with the dish to control channels. It is not connected through cable. It receives signals directly from a satellite.

Wi-Fi. Wireless LAN standards IEEE 802.11, namely 802.11 b, g and a are called Wi-Fi. Wireless LAN is abbreviated as WLAN. In Wi-Fi, Wi is for wireless but Fi is not meaningful. Wi-Fi is similar to Hi-Fi. 802.11a, b and c operate at 2.4 GHZ, 2.4 GHZ and 5 GHZ respectively. Their throughputs are 54 Mbits/s, 11 Mbits/s and 54 Mbits/s respectively.

WEP (Wired Encrypted Piracy). It is a built-in security method in Wi-Fi. Improved security methods described below have been developed for Wi-Fi.

RADIUS (Remote Authentication Dial-In User Service). It is a security protocol for Wi-Fi.

VPN. It is a point-to point tunneling Protocol (PPTP) to provide encryption over Wi-Fi network. It is a security measure for Wi-Fi network.

WPA (Wi-Fi Protected Access). It is an extension of 802.11 wireless LAN standards, 802.11i, which is for more robust security.

WiMAX. IEEE 802.16 standard is also called WiMax (Worldwide Interoperability for Microwave Access). Microwave radio or terrestrial wireless can be used to build wireless LAN and by ISPs to provide wireless Internet services. Microwave wireless system work in the 900 MHZ to 40 GHZ frequency band. It can be used in a point-to-point (P-T-P) or point-to-multipoint (P-T-M) configurations.

FSO (Free Space Optics). It uses laser technology to provide wireless data communication. It imposes line of sight limitation.

VSAT. Internet access to remote areas can be provided by satellite.

D-AMPS (Digital Advanced Mobile Phone System). It was a second-generation mobile phone standard. It was widely used in USA.

GSM (Global System for Mobile Communication). It is a second-generation mobile phones standard. The first-generation of mobile phones was analog voice. The second-generation was for digital voice. It was used everywhere in the world except USA. In USA, D-AMPS was used. The GSM channels are much wider (200 KHZ) than the D-AMPS (30KHZ).

CDMA (Code Division Multiple Access). It is a standard for mobile phones and technically superior to D-AMPS and GSM. It is the basis for third-generation mobile phone systems. It is also a second-generation digital scheme. Its speed is 144 kbps.

GPRS (General Packet Radio Service). It is a mobile phone standard in between the second-generation and the third-generation. It is called 2.5G scheme. It is an overlay packet network on the top of D-AMPS or GSM. Its maximum speed is 56 kbps. For GSM users, Internet access is provided through GPRS.

EDGE (Enhanced Data rates for GSM Evolution). It is also called 2.5G scheme. Its maximum data transmission speed is 230 kbps. It is a radio based high-speed mobile data standard.

W-CDMA (Wideband CDMA). It is a third-generation (3G) mobile phone standard. It runs at 5MHZ bandwidth and its speed is 1920 kbps. It is designed to interwork with GSM networks although it is not backward compatible with GSM. Furthermore, it has the property that a caller can leave a W-CDMA cell and enter a GSM cell without losing the call. The W-CDMA does not work with CDMA. The third-generation schemes are for digital voice and data. With 3G, users have the facilities for video through the Internet, video conferencing etc.

EV-DO (Evolution Date Only). It is a 3G mobile phone standard. It provides 2.4 mbps bandwidth to mobile users on a CDMA network.

UMTS (Universal Mobile Telecommunication System). It is a 3G mobile technology. Its data transfer speed is 1920 kbytes/s.

1.18 SHORT RANGE WIRELESS

Short range wireless system is used for communication between personal digital devices. There is a term PAN (Personal Area Network) which is used when personal devices are connected in a network. Wireless PAN is called **WPAN**. Nowadays, people can connect their PDA to their PC or laptop, answer mobile calls through a wireless headset, do shooting a video with a digital camera and streaming it to a television etc. All this type of things can be done using WPAN. The three technologies, namely Bluetooth 2.0, WUSB (Wireless USB) and ZigBee are used nowadays for WPAN.

Bluetooth 2.0. It operates in the frequency band of 2.45 GHZ. It is same as that for its previous version Bluetooth 1.2. But data transfer rate of Bluetooth 2.0 is 2.1 Mbps against 1 Mbps for version 1.2. Increased data transfer rate enables faster transmission of music and video. Bluetooth 2.0 consumes less power and the maximum distance can go up to 100 metres. It gives improved quality of service for better audio and video streaming. It also supports multicast using which one Bluetooth device can simultaneously transfer data to many Bluetooth devices.

WUSB (Wireless USB). It gives up to 480 Mbps throughput. It is suitable for transferring large amount of data over wireless system. It uses Ultra-Wideband technology for transferring data. It can use a band which is 7 GHZ wide ranging from 3.1 to 10.6 GHZ.

ZigBee. It is suitable for remote monitoring applications such as home, building and industrial automation. It is used for sensor devices for air conditioning, cooling, smoke alarms etc. For such applications very high throughputs are not required. Such applications require low power consumption so that batteries can go for longer periods.

1.19 CONTROL FLOW AND DATA FLOW COMPUTERS

The basic architecture of a conventional computer consists of a CPU, memory and input-output devices. This architecture was given by John von Neumann in mid 1940s, and hence such an architecture is called von Neumann architecture. The program is stored in the memory. The CPU fetches one instruction from the memory at a time and executes it. Thus the instructions are executed sequentially which is inherently a slow process. Neumann machines are called *control flow* computers because instructions are executed sequentially as controlled by a program counter. The speed of a sequential machine is limited by the speed at which the CPU fetches instructions and data from the memory, and the speed at which the CPU executes instructions. To increase the speed of data processing parallel computers have been developed in which several CPUs are connected in parallel to solve a problem. Even in parallel computers the basic building blocks are Neumann processors. The parallelism is not tackled at basic level. The entire burden of solving a problem using parallel processing rests on the programming technique.

The data flow design approach treats the issue of parallelism at a fundamental level. In data flow approach an instruction is executed whenever its required operands become available. No program counter is needed in a data flow computer. Operations are carried out asynchronously as soon as operands required for the operation become available. The fetch/decode unit of the processor, fetches 20 to 30 instructions in advance. The instructions are decoded and their opcodes are kept in an **instruction pool**. The dispatch/execute unit of the

processor checks the opcode of an instruction in the instruction pool, and examines whether its data are available for its execution. If data are available, the instruction is executed. If data are not available for the execution of an instruction, dispatch unit goes ahead and examines the next instruction in the instruction pool. Thus it executes only those instructions for which data are available. The instruction which are not executed due to non-availability of data, are executed in the next round of checking of the instructions in the instruction pool. Whenever data become available remaining instructions are executed. Thus it is seen that the processor does not wait, when data are not available for some instructions. This technique increases the speed of the processor. Pentium Pro, Pentium II, Pentium III and Pentium 4 are data flow processors.

1.20 RISC, CISC AND EPIC PROCESSORS

RISC stands for Reduced Instruction Set Computer. CISC stands for Complex Instruction Set Computer. Where RISC and CISC are used as adjective 'C' in RISC and CISC stands for 'Computing'. There are two techniques of designing a control unit of a processor : hardware technique and software technique. For the execution of an instruction a number of steps are needed. For each step a number of control signals are generated by the control unit of the processor. In a RISC processor, all the necessary signals for the execution of an instruction are generated by an electronic circuit of the control unit. For each instruction there is an electronic circuit in the control unit, to generate necessary control signals. This is a hardware technique of designing a control unit. In this technique microprogramming is not used.

In CISC processor, a microprogram is executed to generate necessary control signals for the execution of an instruction. A microprogram is written for each instruction using a sequence of microinstructions. This technique is known as **microprogramming**. When a microinstruction is executed a set of control signals is produced for the operation of certain step of an instruction. The microinstructions of a microprogram are executed one by one, and in turn, the necessary control signals are produced in steps for the execution of an instruction. For each microinstruction, there is a microcode. Microcodes are stored in a ROM which is within the processor. More details are given in Chapter 5.

The checking of microinstructions can be done more easily than the checking of computer instructions to see that they work properly. If there is an error in a computer instruction, the designer can correct it by changing microinstructions. On the other hand in a RISC processor (hard-wired CPU), if an instruction is erroneous, the designer has to change complex connections or the chip design. In a microprogrammed design of computer, new instructions can be implemented by writing new sets of microinstructions. This is similar to writing various programs using an instruction set of a computer. Such a computer can also be made to execute the instructions of an another computer. This process is known as **emulation**. These advantages of microprogramming are achieved at the cost of processing speed.

EPIC (Explicitly Parallel Instruction Computing) processor has been discussed in Section 5.11.

1.20.1 Microprogrammed Control Unit

Microprogramming has already been explained in the Section 1.20. Fig 1.5 shows a schematic diagram of microprogrammed control unit. The instruction register contains the operation control code (opcode) of an instruction which is to be executed. The decoder decodes

the opcode and gives starting address of the microprogram. The microprogram counter, μ PC contains the address of the next microinstruction to be executed. It is also called control address register. The control memory stores the sets of microinstructions. The microinstruction is read from the control memory and sent to the control buffer register. Then the microinstruction is decoded and control signals are generated and issued. The control signals activate the necessary arithmetic and logic unit etc. to perform the required operations defined by the microinstruction code. In this way a microinstruction is executed. The next address information is also issued and sent to the sequencing logic. The sequencing logic unit sends the address of next microinstruction to μ PC. It also sends a read command to the control memory. In this way the microinstructions of a microprogram are executed one by one to execute an instruction.

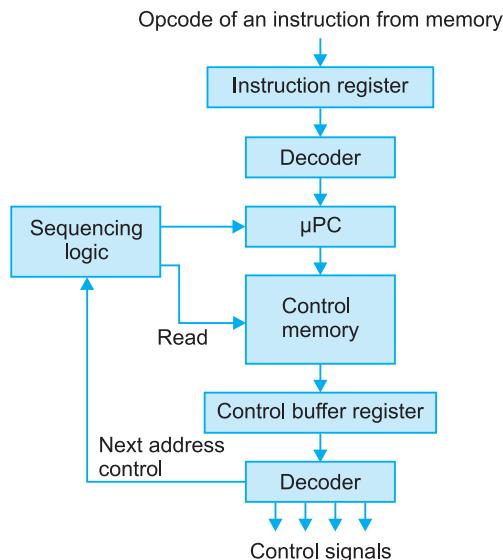


Fig. 1.5 Microprogrammed Control Unit

1.21 COMPUTER APPLICATIONS

Computer is the most versatile tool man has ever created. Nowadays they are being used almost in every sphere of our life. Earlier, a computer was used only for scientific and engineering computational work. But nowadays, besides computational work computers are extensively used for non-computational work such as storing information, transmitting information, creating and handling files in offices, controlling industrial machines and processes, controlling home, business and commercial appliances; ticket reservation in planes and railways, telephone exchange, diagnosing diseases, recording games events, composition of music, painting, book printing, printing newspapers, preparing drawings, in security arrangements to watch and supervise certain areas, to help police in crime investigation, in education, and so on. About 75% of the work done by computers today is of non-computational nature.

Computers are extensively used in offices for handling files, storing information, creating files, administrative purposes, in keeping records of staff, preparing salary bills and for communication. If all work of an office are done by computers, it is called an *electronic office*. The term 'office automation' is used when computers are extensively used for office work.

Wordprocessing softwares are used for creating, manipulating and storing text such as letters, notes, invoices, reports, legal briefs, etc.

Computers are being widely used for preparing accountant's ledger, budget, inventory control; analysis of profit and loss in an investment scheme, purchasing shares and debentures of a company, and many other accounting work. Accounting tasks are quickly analyzed and results may be tabulated or produced in graphical forms.

Modern computer systems can store huge amount of information. A collection of information may be kept in the form of a database. A large collection of data is called a **database**. From a database the required information can be sorted in the desired manner. To handle database special programs called database programs have been developed. Various organizations build and maintain their databases for their work. Some organizations do business to supply information in certain specialized areas such as medicine, engineering, business, etc. Customer becoming a subscriber can get information from such organization through the Internet.

Banks are also using computers for office work, accounting work and dealing with customers. Computers allow customers to make deposits and withdrawals. Cash dispensing machines are also used to make payment in cash and also to receive cash. These are also computer-based machines, called ATM (Automatic Teller Machine).

Computers are now widely used in home and office appliances such as washing machines, TV, telephone receivers, mobile phones etc.

In the medical field computers are used in the diagnosis of diseases, in clinical tests, in office work; in keeping records of cases of patients, treatments, drugs, etc. Some computer expert systems have been developed to diagnose diseases and prescribe medicines. These expert computers are based on the expert knowledge of specialists. Their knowledge is kept updated. The doctors have to supply information such as symptoms, medical history, clinical test results etc. At present expert systems are used to help doctors, not to substitute doctors.

Robots are computer-controlled programmable machines. They have hands, legs, optical sensors to see objects etc. They are used in industry to move materials, parts; to assemble machines etc. They can work in places where human beings cannot work, for example in places where temperature is unbearable and where nuclear radiation are present. They can handle harmful chemicals, radioactive materials and so on.

Internet is a worldwide computer network. There are a large number of Web sites to provide different kinds of information through the Internet in almost all spheres of life. Information is available in the field of arts, science, engineering, commerce, and so on.

Presently, there are numerous more applications of computers. For more details see Chapter 9.

1.22 IMPACT OF COMPUTERS ON SOCIETY

Nowadays computers are being widely used in industries, offices, schools, colleges, universities and research organizations; games, military, communications, hospitals, banks, hotels, homes and so on. They are being used in almost every field concerning the society. They have a great impact on the society. They are changing the way of our working, even the way of life. The use of computers in industries has improved their efficiency. The cost of production goes down. The quality of products is improved. Unpleasant and hazardous work can be performed by robots. Working hours of labourers are reduced; they get more leisure.

The use of computers in offices improves their working efficiency. Record keeping becomes very easy. Information can be searched immediately. Reports, notes, memos, etc. can be prepared immediately. The bureaucratic delay can be minimized to a great extent. It improves office administration. The management task becomes easy.

Computers are capable of manipulating and transmitting texts at very fast rate. They have revolutionized the field of communications. A person sitting at home can contact any office, bank and information supplying organizations through a computer. Instead of using a computer he may use his television set and an adapter and a keyboard attached to it for such purposes. Sitting at home, he may do office work, bank transactions, marketing, purchase of shares, reservation of plane and bus tickets and so on. Officers and managers can contact any branch office anywhere in the world through the computer network. Text, voice, data and picture can easily be transmitted nowadays using modern communication systems which heavily depend upon computers. Information is transmitted over long distances through satellites. Video conferencing allows two-way communication between persons. They can see each other on the screen. Nowadays video conferences are being arranged. People participate in conference while sitting at their homes or offices. Electronic mail is being used to transmit information. Information can be stored if a person is physically not available. Later, he can have the information. Information may be either in text form displayed on the screen or voice form.

Many organizations have set up information centres. They have created databases for specialized as well as general services. Professionals can get information in specialized fields such as engineering, medicine, law etc. for their guidance. General people can get information regarding certain products, stock exchange, market information, reservations for tours, weather information, news and so on.

Computers enable people getting better services from government or private organizations. There will be shorter waiting lines in queue in banks, railway ticket reservation counters, airline ticket offices; better and fast services at hospitals, improved clinical tests and diagnosis of diseases and so on.

Computers are playing an important role in education. One can get lessons on certain topics, special lectures prepared by experts etc. on the screen as many times as he wants until he understands the topic. Computers are working as teachers or helping teachers in educational institutions.

Thus we see that computers have great impact on our society. They are even changing the way we work. There are certain disadvantages also. There may be cases of embezzlement in banks, leakage or misuse of personal information, etc. When a person is using computer for marketing, office work, collection of information, etc. his activity is recorded. One can misuse this record. So individual privacy is not guaranteed. An efficient security measure has to be taken. Hardwares and softwares are available for such purposes. Someone can play mischief on computer. He can erase certain important data. He may take away some important and private data. Extensive use of computers for the job which is labour oriented will cause unemployment. But in industries computers can be used for machine control and process control etc. for which electronic or electromechanical control systems were employed in the past. Such applications of computers do not much affect labour employment.

Industrial revolution changed our society from an agricultural society to an industrial society. Now computer is changing our society from an industrial society to an information society. In U.S.A., around 1900, 35% of labour force was engaged in agriculture, 27 per cent in industries and 13% in information sector. By 1980, about 50% were engaged in information sector, 23% in industries and only 2% in agriculture.

The technology itself does not decide the future of the society. The way we apply the technology for the welfare of mankind is important. This is also true for computers. As we use automobiles and electric motors today extensively, computers will also be widely used by us. As computers are more powerful, their impact on the society will be much more than the application of any other tools.

1.23 FUTURE DEVELOPMENTS

At present silicon is used as semiconductor material to manufacture semiconductor devices: LSI and VLSI chips. Electrons can move more easily in gallium-arsenide (GaAs) than in silicon. Research is going on to use GaAs to manufacture semiconductor components of a computer. Fujitsu of Japan is making efforts in this direction. It introduced a supercomputer VP-200 in 1982, which incorporates GaAs technology to some extent.

Research is also going on to use light beam instead of electrons in semiconductor devices. Bell Labs. Research worker Devid Miller has developed an optical transistor or chip. It controls light beam in the same way as a transistor controls electric current in it. Computers which will use photons (that is, light beam) instead of electrons for their operation will be called optical computer or quantum computer. An optical computer will be much more faster and powerful than computers using silicon technology.

Further, research is also going on to develop biochips. Such chips will use organic material for their fabrication. Molecules of a substance will be organized to act as electronic memory or switching devices. Research is also going on in Japan and U.S.A. to study the behaviour of some worms, which will be helpful to develop computers of tomorrow. It has been observed that some worms are more intelligent than today's computers and more dexterous than robots. Some show information processing capability and can learn something which today's computer cannot do. For details the reader may refer to the book mentioned in Ref. 4.

1.24 VIRUS

A computer virus is a small program written with bad motive to affect badly the operation of a computer. Viruses are prepared by antisocial elements to damage other's computers, their important files etc. They may corrupt files, erase files, make a computer slow and so on. Such harmful programs are called virus because they can spread from one computer to another computer. They reproduce themselves and then cause damage. They infect other files and then spread. They spread via E-mail or through files which are downloaded. When virus is present in a file that is stored in a CD or floppy, it will spread to another computer which uses that CD/floppy.

A virus hiding in a program is known as Trojan horse. When the Trojan horse program is run, the virus loads itself into the computer's memory. Once it is there, it can secretly attach itself to other files/programs or store itself on any other disks run on the computer including the hard disk.

To protect computer from viruses, anti-virus softwares are available. Before downloading any file from Web, it must be scanned whether it is free from viruses. When any file is to be copied from CD or floppy, it must be scanned.

1.25 MP3 COMPRESSION STANDARD

MP3 is a standard method of compression of audio signals. It is the short form for Moving Pictures Expert Group-Version2-Layer 3 audio compression standard. The memory space requirement is reduced by a factor about 10 when MP3 standard is used to store an audio file. A 60 minutes MP3 music file requires only about 50 MB memory space. Without compression, 60 minutes music file in digital form requires about 600 MB memory space.

1.26 MPEG COMPRESSION STANDARD

MPEG is a standard compression method for video signals. It is short form for Motion Pictures Group. It is pronounced as 'em-peg'. It is not a single standard, but a series of standards. MPEG-1 is for TV quality video and low resolutions visual display unit (VDU) of a personal computer (640 X 480 pixels resolution). MPEG-2 supports HDTV (High Definition TV). HDTV is a form of digital transmission. It gives better quality of pictures and exceptional sound. Earlier, most routine broadcasts used analog television signals, which were transmitted like radio signals. MPEG-2 is an emerging new standard for TV broadcasts.

1.27 PERSONAL MEDIA PLAYER PMP

After MP3, it is the personal media player taking portable entertainment world by storm. PMPs play almost all kinds of digital audio and video formats depending upon the device capability. A PMP contains a mini hard drive, a LCD screen and a removable rechargeable battery. The user can view video files on the LCD screen. Most of the PMPs contains at least 20 GB hard drive. A 20 GB hard drive can carry about 15 Div X movies along with 500 songs.

1.28 DIV X

Div X is a media format. A new package 'Div X Create Bundle' is available. It is video compression utility. It includes the new improved Div X 6 codec, Div X converter and Div X pro. The new version of the Div X codec represents a significant update over version 5 and offers better image quality. The Div X converter converts a video file in Div X format. The user can select Div X profile. If there are a number of video files having same resolution and frame rate, they can be combined into one single file with the selection of menu. There is an extension to Div X which allows to add menu, chapter titles, subtitles etc. A Div X player plays a Div X-encoded file.

PROBLEMS

1. What are the essential components of a computer? Draw the schematic block diagram of a computer showing its essential components. Discuss the function of each component.
2. Discuss the important features of various generations of computers. Give some examples of computers for each generation.

3. Explain primary memory, secondary memory and cache memory. What type of memory devices are used in each of these categories of memory?
4. Explain what you understand by real and virtual memory.
5. Discuss the features of direct access storage devices and serial access storage devices.
6. Show that the memory addressing capability of a CPU is equal to 2^n bytes, where n is the number of address lines of the CPU.
7. Explain what are MIPS, MFLOPS, SPECint, SPECfp, KLIPS, Dhystone and Whetstone.
8. Which microprocessor is suitable for (i) Client computer, (ii) Server, (iii) Workstation and (iv) Supercomputer?
9. Discuss the important features of (a) Palm PC, (b) Notebook (c) Desktop, (d) Server, (e) Workstation and (f) Supercomputer.
10. Explain the terms hardware, software, firmware and freeware.
11. Explain what you understand by system software and operating system. Give examples of some important operating systems.
12. What are application programs? What is their utility? Discuss the features of an integrated package.
13. What is an assembler, compiler and interpreter?
14. Explain what you understand by low-level language and high-level language. Give examples of each.
15. Discuss multiprogramming, multitasking and multiuser system.
16. Explain (i) Computer network and (ii) Distributed processing.
17. What is multithreading? Explain.
18. What is LAN, WAN and Internet? What is Ethernet?
19. What do you understand by the term, 'broadband';
20. Discuss Wi-Fi and DTH.
21. What is computer virus? Explain.
22. What are RISC, CISC and EPIC processors?
23. Explain MP3, MPEG, PMP and DivX.
24. Distinguish between controlflow and dataflow computers.
25. Discuss the important applications of computers in different areas.
26. What is the impact of computers on our society?
27. What is artificial intelligence? Give examples and application of some systems employing artificial intelligence.
28. What is a search engine? Give some examples of search engines. What is the role of a Web crawler?
29. Define client and server computers?
30. What is Web browser? Give its examples.
31. Explain the following :
URL, Webpage, Website, FTP, TCP/IP, HTML, Applets, Packets, and ISP
32. Explain the following:
Bridge, Gateways, Routers and Hub.
33. What are Usenet and FTP?
34. What is Intranet? Where is it used.

**THIS PAGE
IS BLANK**

2

CHAPTER

NUMBER SYSTEM

2.1 INTRODUCTION

We are familiar with the decimal number system which is used in our day-to-day work. In the decimal number system there are ten digits which are used to form decimal numbers. Ten separate symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 are used to represent ten decimal digits. A digital computer stores, understands and manipulates information composed of only zeros and ones. A programmer (or user) who works on a computer is allowed to use decimal digits; letters A, B, C, . . . Z, a, b, c, . . . , z, usual special symbols, +, -, etc. for his convenience. The decimal digits, letters, symbols, etc. are converted to binary codes in the form of 0s and 1s within the computer. To understand the operation of a computer the knowledge of binary, octal and hexadecimal number system is essential. This chapter deals with these number systems.

2.2 DECIMAL NUMBER SYSTEM

As the ten fingers of our hands are the most convenient tools nature has given, human beings have always used them in counting. So the decimal number system followed naturally from their use. The base or radix of a number system is defined as the number of digits it uses to represent numbers in the system. Since the decimal number system uses ten digits, from 0 to 9, its base or radix is 10. The decimal number system is also called base-10 number system. The weight of each digit of a decimal number depends on its relative position within the number. This is explained by the following example.

Example. Take the decimal number 6498 as an example to explain the weight of each digit of the number.

$$\begin{aligned} 6498 &= 6000 + 400 + 90 + 8 \\ &= 6 \times 10^3 + 4 \times 10^2 + 9 \times 10^1 + 8 \times 10^0 \end{aligned}$$

The weight of each digit of a decimal number depends on its relative position within the number as explained below:

The weight of the 1st digit of the number from the right hand side = 1st digit $\times 10^0$.

The weight of the 2nd digit of the number from the right hand side = 2nd digit $\times 10^1$.

The weight of the 3rd digit of the number from the right hand side = 3rd digit $\times 10^2$.

The weight of the 4th digit of the number from the right hand side = 4th digit $\times 10^3$.

The above expressions can be written in general form as follows:

The weight of the n th digit of the number from the right hand side

$$\begin{aligned} &= n\text{th digit} \times 10^{n-1} \\ &= n\text{th digit} \times (\text{Base})^{n-1} \end{aligned}$$

The number system in which the weight of each digit depends on its relative position within the number, is called **positional number system**. The above form of general expression is true only for positional number system.

It is India that gave this positional method of expressing any number using ten symbols, each symbol receiving a value of position as well as an absolute value. It was a profound and important idea. Its merit has been appreciated by a famous mathematician Marquis de Laplace.

2.3 BINARY NUMBER (OR BASE-2) SYSTEM

The base (or radix) of the binary number system is 2. It uses only two digits, 0 and 1. In short a binary digit is called a bit. The storing or computing electronic elements of a computer have only two stable states. The output of such an element at any time is either HIGH (5 volts) or LOW (0 volt). These are the only two stable states. There is no other stable state. These stable states can be represented by 1 and 0 respectively, that is HIGH is represented by 1 and LOW by 0. Due to this very limitation a computer can understand information composed of only 0s and 1s. So all computers perform their internal operations and manipulations on binary digits. For the convenience of the users (programmers) they are allowed to use data and other information in the form of decimal digits, usual alphabets and special symbols. This information is converted to binary codes within the computer as the computer operates on binary bits. Thus we see that the knowledge of the binary number system is needed for those who want to understand the operating principle of a computer. It is not required by those who have to simply use a computer for their work.

In the decimal number system there is no difficulty in representing numbers upto 9. But there is no symbol or digit to represent ten and therefore, it is represented by 10. It is simply a positional technique. Again, after 99 we have to represent hundred and utilizing positional technique it is written as 100. In the binary number system zero is represented by 0 and one by 1. There is no digit in the binary number system to represent two. Therefore, using positional technique it is written as 10. Three is written as 11. Again four is represented by 100. In this way utilizing positional technique we proceed further. Thus it is seen that a binary number becomes very long and cumbersome. The weight of each binary bit of a binary number depends on its relative position within the number. It has been explained by the following example.

Example. Take the binary number 1101 as an example to explain the weight of each bit of the number.

$$\begin{aligned} 1101 \text{ (Binary Number)} &= 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 8 + 4 + 0 + 1 = 13 \text{ (Decimal Number)} \end{aligned}$$

The weight of each bit of a binary number depends on its relative position within the number as explained below:

The weight of the 1st bit of the binary number from the right hand side = 1st bit $\times 2^0$.

The weight of the 2nd bit of the number from the right hand side = 2nd bit $\times 2^1$.

The weight of the 3rd bit of the number from the right hand side = 3rd bit $\times 2^2$.

The weight of the 4th bit of the number from the right hand side = 4th bit $\times 2^3$.

The above expressions can be written in the form of a general expression given below.

The weight of the n th bit of the number from the right hand side

$$\begin{aligned} &= n\text{th bit} \times 2^{n-1} \\ &= n\text{th bit} \times (\text{Base})^{n-1} \end{aligned}$$

It is seen that this rule for a binary number is same as that for a decimal number. The above rule holds good for any other positional number system. The weight of a digit in any positional number system depends on its relative position within the number and the base of the number system. Table 2.1 shows binary equivalent of decimal numbers.

Table 2.1. Binary Equivalent of Decimal Numbers

Decimal Number	Binary Equivalent	Decimal Number	Binary Equivalent
0	0	11	1011
1	1	12	1100
2	10	13	1101
3	11	14	1110
4	100	15	1111
5	101	16	10000
6	110	31	11111
7	111	32	100000
8	1000	63	111111
9	1001	64	1000000
10	1010	128	10000000

2.4 CONVERSION OF A BINARY NUMBER TO DECIMAL NUMBER

To convert a binary number to its decimal equivalent we use the following expression:

The weight of the n th bit of the number from the right hand side = n th bit $\times 2^{n-1}$.

First we mark the bit position and then we give the weight of each bit of the number depending on its position. The sum of the weights of all bits gives the equivalent number. The following example illustrates the process.

Example 1. Convert the binary number 10 to its decimal equivalent.

$$\begin{aligned} \text{The first bit counting from the right hand side} &= 0 \\ \text{Its weight} &= 0 \times 2^{1-1} = 0 \times 2^0 \\ \text{The 2nd bit from the right hand side} &= 1 \\ \text{Its weight } 1 \times 2^{2-1} &= 1 \times 2^1. \\ \text{The decimal equivalent} &= 1 \times 2^1 + 0 \times 2^0 \\ &= 2 + 0 = 2 \end{aligned}$$

Example 2. Convert the binary number 101 to its decimal equivalent.

First we can write the binary number in the spread form and then show the bit position as shown below.

The binary number in the spread form = 1 0 1

Bit position from right hand side: 3rd 2nd 1st

The weight of each bit is assigned and added together to give the decimal equivalent.

$$101 \text{ (Binary number)} = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 4 + 0 + 1$$

$$= 5 \text{ (Decimal Number).}$$

Example 3. Convert the binary number 1010 to its decimal equivalent.

The binary number in the spread form = 1 0 1 0

Bit position from the right hand side: 4th 3rd 2nd 1st

$$1010 \text{ (Binary Number)} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$= 8 + 0 + 2 + 0$$

$$= 10 \text{ (Decimal Number).}$$

Example 4. Convert the binary number 11001 to its decimal equivalent.

The binary number in the spread form = 1 1 0 0 1

Bit position from the right hand side: 5th 4th 3rd 2nd 1st

$$11001 \text{ (Binary Number)} = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$= 16 + 8 + 0 + 0 + 1$$

$$= 25 \text{ (Decimal Number).}$$

To indicate the base of a number the following technique can be used:

$$(25)_{10} = (11001)_2$$

or

$$25_{10} = 11001_2$$

The suffix 10 indicates that the number 25 is a decimal number.

The suffix 2 indicates that the number 11001 is a binary number.

2.5 CONVERSION OF A DECIMAL NUMBER TO A BINARY NUMBER

In a decimal number the 1st position from the right hand side is for 1s, 2nd for 10s, 3rd for 100s, 4th for 1000s and so on. Similarly, in a binary number the 1st position from right hand side is for 1, 2nd for 2, 3rd for 4, 4th for 8, 5th for 16 and so on. This fact is utilized for the conversion of a decimal number to its binary equivalent. A method is to be developed to determine which multiples of 2 are present in a given decimal number. For example, the decimal number 13 is equal to $(8 + 4 + 1)$. In other words it has one 8, one 4 and a 1. It does not have any 2. Therefore, its binary equivalent is 1101. Based on this concept, for the conversion of a decimal number to binary number, the decimal number is divided by 2 successively. The quotient and remainder are noted down at each stage. The quotient of one stage is divided by 2 at the next stage. The process is repeated until the quotient becomes zero. The binary number equivalent to the decimal number is given by the following expression.

$$\text{Binary Number} = R_k R_{k-1} R_{k-2} \dots R_3 R_2 R_1$$

where R_1, R_2, \dots, R_k are the remainders at 1st, 2nd,... and k th stage respectively.

Example 1. Convert the decimal number 41 to its binary equivalent.

	<i>Quotient</i>	<i>Remainder</i>	<i>Remark</i>
$41 \div 2 = 20$,	1 (LSB)	There are 20 twos and one 1.
$20 \div 2 = 10$,	0	There are 10 fours and no 2. This is equivalent to dividing by 4.
$10 \div 2 = 5$,	0	There are 5 eights and no 4. Equivalent to dividing by 8.
$5 \div 2 = 2$,	1	There are two 16s and one 8. Equivalent to dividing by 16.
$2 \div 2 = 1$,	0	There is one 32 and no 16. Equivalent to dividing by 32.
$1 \div 2 = 0$,	1 (MSB)	There is no 64 and one 32. Equivalent to dividing by 64.

Therefore, 41 (Decimal Number) = 101001 (Binary Number). The first remainder is the least significant bit (LSB) and the last remainder the most significant bit (MSB).

Example 2. Convert the decimal number 73 to binary.

	<i>Quotient</i>	<i>Remainder</i>
$73 \div 2 = 36$,	1 (LSB)
$36 \div 2 = 18$,	0
$18 \div 2 = 9$,	0
$9 \div 2 = 4$,	1
$4 \div 2 = 2$,	0
$2 \div 2 = 1$,	0
$1 \div 2 = 0$,	1 (MSB)

The decimal number $73 = 1001001$ (Binary Number).

Checking of the answer.

The decimal equivalent of the above binary number is given by

$$\begin{aligned}1001001 &= 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\&= 64 + 0 + 0 + 8 + 0 + 0 + 1 = 73 \text{ (Decimal Number)}\end{aligned}$$

Example 3. Convert the decimal number 153 to binary.

	<i>Quotient</i>	<i>Remainder</i>
$153 \div 2 = 76$,	1 (LSB)
$76 \div 2 = 38$,	0
$38 \div 2 = 19$,	0
$19 \div 2 = 9$,	1
$9 \div 2 = 4$,	1
$4 \div 2 = 2$,	0

$$2 \div 2 = 1$$

,

0

$$1 \div 2 = 0$$

,

1 (MSB)

153 (Decimal Number) = 10011001 (Binary Number).

Checking of the answer.

$$\begin{aligned} 10011001 &= 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 128 + 0 + 0 + 16 + 8 + 0 + 0 + 1 \\ &= 153 \text{ (Decimal Number).} \end{aligned}$$

2.6 ADDITION OF BINARY NUMBERS

In the binary number system $1 + 0 = 1$. When 1 is added to 1, the sum is 0 with a carry 1. If the sum is written upto 2 bits, it is equal to 10 (2 decimal). The Table 2.2 shows the rules for binary addition.

Table 2.2 Binary Addition

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1 0 ↑ Carry

When 1 is added to 1, the sum is 0 with a carry 1. This carry is added to the sum of the adjacent bits.

Example 1

$$\begin{array}{r} 1001 \quad (9 \text{ decimal number}) \\ + 0101 \quad (5 \text{ decimal number}) \\ \hline 1110 \quad (14 \text{ decimal number}) \end{array}$$

Example 2

$$\begin{array}{r} 0111 \quad (7 \text{ decimal number}) \\ + 0011 \quad (3 \text{ decimal number}) \\ \hline 1010 \quad (10 \text{ decimal number}) \end{array}$$

Example 3

$$\begin{array}{r} 1010 \quad (10 \text{ decimal number}) \\ + 1101 \quad (13 \text{ decimal number}) \\ \hline 10111 \quad (23 \text{ decimal number}) \\ \uparrow \\ \text{Carry} \end{array}$$

2.7 BINARY SUBTRACTION

The following examples will illustrate binary subtraction.

Example 1

$$\begin{array}{r}
 \text{Borrow} \\
 \downarrow \\
 \begin{array}{r}
 1110 \quad (14 \text{ decimal number}) \\
 - 0101 \quad (- 5 \text{ decimal number}) \\
 \hline
 1001 \quad (9 \text{ decimal number})
 \end{array}
 \end{array}$$

Example 2

$$\begin{array}{r}
 1010 \quad (10 \text{ decimal number}) \\
 - 0101 \quad (- 5 \text{ decimal number}) \\
 \hline
 0101 \quad (5 \text{ decimal number})
 \end{array}$$

Example 3

$$\begin{array}{r}
 1010 \quad (10 \text{ decimal number}) \\
 - 0011 \quad (- 3 \text{ decimal number}) \\
 \hline
 0111 \quad (7 \text{ decimal number})
 \end{array}$$

Example 4

$$\begin{array}{r}
 1101 \quad (13 \text{ decimal number}) \\
 - 0111 \quad (- 7 \text{ decimal number}) \\
 \hline
 0110 \quad (6 \text{ decimal number})
 \end{array}$$

In the above examples smaller number has been subtracted from a larger number. Let us see what happens if a larger number is subtracted from a smaller number.

Example 1

$$\begin{array}{r}
 \text{Borrow} \\
 \downarrow \\
 \begin{array}{r}
 0101 \quad (5 \text{ decimal number}) \\
 - 0111 \quad (- 7 \text{ decimal number}) \\
 \hline
 1110 \quad (- 2 \text{ decimal number})
 \end{array}
 \end{array}$$

The result is not a simple representation of -2 i.e., -0010 . The result is the 2's complement of 2. This will be explained in the next section.

Example 2

$$\begin{array}{r}
 \text{Borrow} \\
 \downarrow \\
 \begin{array}{r}
 0111 \quad (7 \text{ decimal number}) \\
 - 1000 \quad (- 8 \text{ decimal number}) \\
 \hline
 1111 \quad (-1 \text{ decimal number})
 \end{array}
 \end{array}$$

The result is 2's complement of 1.

2.8 USE OF COMPLEMENTS TO REPRESENT NEGATIVE NUMBERS

Most of the computers now perform subtraction using complemented number. This is economical because subtraction can also be performed using the same electronic circuitry which is used for addition. In the binary number system there are two types of complements, 1's complement and 2's complement. To represent a negative binary number 2's complement is most widely used at present. To understand 1's and 2's complement in the binary number system one should first understand 9's and 10's complements in the decimal number system.

2.8.1 9's Complement

To form the 9's complement of decimal number each digit of a decimal number is subtracted from 9. The result so obtained is known as 9's complement of the number. For example, the 9's complement of 37 is $(99 - 37) = 62$. The 9's complement of 235 is $(999 - 235) = 764$.

2.8.2 10's Complement

The 10's complement of a decimal number is equal to the 9's complement of the number plus 1.

The 10's complement of a decimal number = Its 9's complement + 1.

The 10's complement of 37 = $62 + 1 = 63$.

The 10's complement of 235 = $764 + 1 = 765$.

Now let us examine the sum of a decimal number and its 10's complement.

Example 1

$$\begin{array}{r}
 37 \quad \text{(decimal number)} \\
 + 63 \quad \text{(its 10's complement)} \\
 \hline
 1\ 00
 \end{array}$$

↑ Ignore carry

In the above example the given decimal number is of two digits. If the sum of the number and its 10's complement is considered only upto two digits, the sum is equal to zero. In other words the sum of a number and its 10's complement is equal to zero if the carry of the last stage is neglected.

Example 2

$$\begin{array}{r}
 235 \quad \text{(decimal number)} \\
 + 765 \quad \text{(its 10's complement)} \\
 \hline
 1\ 000
 \end{array}$$

↑ Ignore carry

The decimal number 235 is of three digits. If the sum of a decimal number and its 10's complement is considered only upto three digits, the sum is equal to zero. Therefore, it is seen that the 10's complement represents the negative value of the number.

The 10's complement of a decimal number = - decimal number.

2.8.3 Addition in 10's Complement System

If 10's complement of a number is added to any number it is equivalent to its subtraction from that number. 10's complement subtraction is used in computers which employ BCD codes for the representation of decimal numbers.

Example 1. Add 86 and (- 21).

The 9's complement of 21 = 78

The 10's complement of 21 = 78 + 1 = 79

$$\begin{array}{r}
 86 \\
 + 79 \quad \text{(10's complement of 21)} \\
 \hline
 1\ 65 \\
 \uparrow \quad \text{Ignore carry}
 \end{array}$$

If the carry of the last stage is dropped, the result is correct.

Example 2. Add 59 and (- 84)

The 9's complement of 84 = 15

The 10's complement of 84 = 15 + 1 = 16

$$\begin{array}{r}
 59 \\
 + 16 \quad \text{(10's complement of 84)} \\
 \hline
 75 \quad \text{(10's complement of 25)}
 \end{array}$$

There is no carry and the result is in 10's complement. It is the 10's complement of 25. To get the correct result take 9's complement of the result, add 1 and put a - sign before it.

The 9's complement of 75 = 24

The 10's complement of 75 = 24 + 1 = 25

Result = - 25, which is the correct result.

Example 3. Add (- 26) and (- 43)

The 9's complement of 26 = 73

The 10's complement of 26 = 73 + 1 = 74

The 9's complement of 43 = 56

The 10's complement of 43 = 56 + 1 = 57

$$\begin{array}{r}
 74 \\
 + 57 \\
 \hline
 1\ 31 \\
 \uparrow \quad \text{Ignore carry}
 \end{array}$$

The 10's complement of 31 = 69. The correct result is - 69.

Example 4. Add 34 and 58

$$\begin{array}{r}
 34 \\
 58 \\
 \hline
 92
 \end{array}$$

As the both numbers are positive, they are simply added. The 10's complement is not required. The result is positive and correct. There is no carry at the last stage.

Conclusion

Whether the result is positive or negative the carry of the last stage is to be dropped. If the result is positive it is correct as usual. If the result is negative, it is in 10's complement. Computer can represent signed numbers, and hence recognize whether the result is positive or negative. If the result negative, computer understands that it is in the complemented form. So it converts it to get the result in the usual form. This has been explained while discussing 2's complement which is similar to 10's complement in the decimal system.

2.8.4 1's Complement

The 1's complement in the binary number system is similar to 9's complement in the decimal system. To obtain 1's complement of a binary number each bit of the binary number is subtracted from 1. For example, the 1's complement of the binary number 010 is 101. The 1's complement of 1110 is 0001. Thus 1's complement of a binary number may be formed by simply changing each 1 to a 0 and each 0 to a 1.

Example 1. Find 1's complement of the binary number 101100.

The 1's complement of 101100 = 010011

Example 2. Find 1's complement of the binary number 0000.

The 1's complement 0000 = 1111

Example 3. Find 1's complement of the binary number 1111.

The 1's complement of 1111 = 0000

2.8.5 2's Complement

The 2's complement in the binary number system is similar to 10's complement in the decimal number system. The 2's complement of a binary number is equal to the 1's complement of the number plus one.

The 2's complement of a binary number = Its 1's complement + 1.

The 2's complement of 0101 = 1010 + 1 = 1011.

Example 1. Find 2's complement of the binary number 101100.

The 2's complement of 101100 = 010011 + 1 = 010100

Example 2. Find 2's complement of the binary number 0000.

The 2's complement of 0000 = 1111 + 1 = 10000

Example 3. Find 2's complement of the binary number 111.

The 2's complement of 111 = 000 + 1 = 001

Let us examine the sum of a binary number and its 2's complement.

Example 1. Add the binary number 1100 and its 2's complement.

The binary number = 1100

Its 1's complement = 0011

Its 2's complement = 0011 + 1 = 0100

$$\begin{array}{r} \text{The number + its 2's complement} = 1100 \\ + 0100 \\ \hline 10000 \\ \uparrow \quad \text{Ignore carry} \end{array}$$

If the carry of the last stage is neglected the sum of a binary number and its 2's complement is equal to zero. In the above example the binary number is of 4 bits. If a 4-bit processor is used and the last carry is neglected, the sum will be considered only upto 4 bits.

For an 8-bit processor the binary number and its 2's complement is written upto 8 bits as shown below in the examples 2 and 3.

Example 2. Add the binary number 1011 and its 2's complement.

$$\begin{array}{lcl} \text{The binary number} & = 00001011 \\ \text{Its 1's complement} & = 11110100 \\ \text{Its 2's complement} & = 11110100 + 1 \\ & = 11110101 \\ \text{The given binary number} & = 00001011 \\ + \text{Its 2's complement} & + 11110101 \\ \hline 100000000 \\ \uparrow \quad \text{Ignore carry} \end{array}$$

The sum only upto 8 bits is equal to zero.

Example 3. Add 5 and (- 5).

5 (decimal) = 00000101 (binary)

- 5 = Its 2's complement = 11111010 + 1 = 11111011

$$\begin{array}{r} 5 = 00000101 \\ + (-5) + 11111011 (+ 2's \text{ complement of } 5) \\ \hline 100000000 \\ \uparrow \quad \text{Ignore carry} \end{array}$$

If the sum is considered only upto 8 bits, it is equal to zero.

2.8.6 Binary Subtraction Using 2's Complement

The addition of 2's complement of a number is equivalent to its subtraction. This will be clear from the following example.

Example 1. Subtract 2 from 6.

Simple binary subtraction

$$\begin{array}{r} 0110 \quad (6 \text{ decimal}) \\ - 0010 \quad (-2 \text{ decimal}) \\ \hline 0100 \quad (4 \text{ decimal}) \end{array}$$

Subtraction using 2's complement

1's complement of 0010 (2 decimal) = 1101

2's complement of 0010 (2 decimal) = 1101 + 1 = 1110

$$\begin{array}{r}
 0110 \quad (6 \text{ decimal}) \\
 + 1110 \quad (+ 2\text{'s complement of } 2) \\
 \hline
 1 0100 \quad (4 \text{ decimal})
 \end{array}$$

↑ Ignore carry

The carry of the last stage is to be neglected if we are using 2's complement technique.

Example 2. Subtract 3 from 5

$$\begin{array}{lcl}
 1\text{'s complement of } 0011 \text{ (3 decimal)} & = & 1100 \\
 2\text{'s complement of } 0011 \text{ (3 decimal)} & = & 1100 + 1 \\
 & & = 1101
 \end{array}$$

$$\begin{array}{r}
 0101 \quad (5 \text{ decimal}) \\
 + 1101 \quad (+ 2\text{'s complement of } 3) \\
 \hline
 1 0010 \quad (2 \text{ decimal})
 \end{array}$$

↑ Ignore carry

The carry of the last stage is neglected.

2.8.7 Representation of Signed and Unsigned Numbers

In the decimal system we put + or - sign before a number to represent its sign. In computer such notation can not be employed and therefore, a different technique has been adopted. To represent positive sign a 0 is placed before the binary number. For example + 9 is represented by 0 1001. To represent negative number a 1 is placed before the number. For example, -9 will be represented as 1 1001. There is only one way to represent a positive number. But there are three different ways to represent a negative number. These are:

- (1) Signed-Magnitude Representation
- (2) Signed-1's Complement Representation
- (3) Signed-2's Complement Representation

The representation of - 9 in above three representations are shown below:

Signed-magnitude representation 1 1001

Signed-1's complement representation 1 0110

Signed-2's complement representation 1 0111

Hence, 9 is represented by 4 binary bits and a separate bit is used to represent sign. In a computer the MSB can be used to represent the sign of the number. For example an 8-bit computer will represent - 9 as shown below. 7 bits are used to represent the number and the MSB is used to represent the sign of the number.

Signed-magnitude representation 1 0001001

Signed-1's complement representation 1 1110110

Signed-2's complement representation 1 1110111

When all the bits of the computer word are used to represent the number and no bit is used for sign representation it will be called unsigned representation of the number.

2.8.8 Addition in Signed 2's Complement

The 2's complement in the binary number system is similar to 10's complement in the decimal system. We have already discussed the rules for addition in 10's complement. Those rules also hold good for 2's complement. A sign bit adjacent to the most significant bit is used to distinguish a positive number from a negative number. A 1 in the sign bit is used to represent a negative number and a 0 in the sign bit a positive number. The use of sign bit clearly indicates whether the result is positive or negative. The following examples will show the different situations in 2's complement addition.

Example 1. Addition of two positive numbers.

Add 5 and 3.

Normal Notation

$$\begin{array}{r} +0101 \\ +0011 \\ \hline +1000 \end{array} \quad (+5 \text{ decimal}) \quad (+3 \text{ decimal})$$

Computer Notation

$$\begin{array}{r} 0\ 0101 \\ 0\ 0011 \\ \hline 0\ 1000 \end{array} \quad (+5 \text{ decimal with sign bit}) \quad (+3 \text{ decimal with sign bit})$$

In computer representation sign bit has also been included. The first bit from the left (or fifth bit from the right) is the sign bit. The addition is also performed on the sign bit. In the above example the sum 1000 is correct in the binary form and it is equal to 8 decimal. A 0 in the sign bit indicates that the result is positive. There is no carry at the last stage.

Example 2. Addition of a positive and a negative number, the positive number being greater than the negative number.

Add 9 and (-4).

The 1's complement of 0100 (4 decimal) = 1011

The 2's complement of 0100 = $1011 + 1 = 1100$

Normal Notation

$$\begin{array}{r} +1001 \\ -0100 \\ \hline +0101 \end{array} \quad (+9 \text{ decimal}) \quad (-4 \text{ decimal})$$

Computer Notation

$$\begin{array}{r} 0\ 1001 \\ 1\ 1100 \\ \hline 1\ 0\ 0101 \end{array} \quad (+9 \text{ decimal with sign bit}) \quad (2\text{'s decimal of } 4 \text{ with sign bit})$$

↑ Neglect carry

The sum 0101 is correct in the binary form and equal to 5 decimal. The sign bit 0 indicates the sum is positive. There is carry at the last stage. The carry is to be dropped.

Example 3. Addition of a positive and negative number, the negative number being greater than the positive number.

Add (-9) and 3.

The 1's complement of 1001 (9 decimal) = 0110

The 2's complement of 1001 = $0110 + 1 = 0111$

$$\begin{array}{r} -9 \\ -1 \\ 0111 \end{array} \quad (2\text{'s complement of } 9 \text{ with sign bit})$$

$$\begin{array}{r} +3 \\ +0 \\ 0011 \end{array} \quad (+3 \text{ with sign bit})$$

$$\begin{array}{r} -6 \\ -1 \\ 1010 \end{array} \quad (2\text{'s complement of } 6 \text{ with sign bit})$$

The sum is negative as indicated by the sign bit. It is in 2's complement. There is no carry at the last stage.

When the result is negative and in 2's complement the following procedure can be adopted to get it in the correct form.

Take 1's complement of the result, add 1 and put a – sign before it. In the above example the result in 2's complement is 1010. The correct result will be:

$$\begin{array}{r} \text{The 1's complement of 1010} = 0101 \\ + \quad \quad \quad 1 \\ \hline -0110 \quad (-6 \text{ decimal}) \end{array}$$

Example 4. Add (- 12) and (- 2).

$$\begin{array}{lcl} \text{The 1's complement of 1100 (12 decimal)} & = 0011 \\ \text{The 2's complement of 1100} & = 0011 + 1 \\ & = 0100 \\ \text{The 1's complement of 0010 (2 decimal)} & = 1101 \\ \text{The 2's complement of 0010} & = 1101 + 1 \\ & = 1110 \end{array}$$

Computer Notation

$$\begin{array}{r} -12 \quad 10100 \quad (\text{2's complement of 12 with sign bit}) \\ -2 \quad 11110 \quad (\text{2's complement of 2 with sign bit}) \\ \hline -14 \quad 110010 \quad (\text{2's complement of 14 with sign bit}) \\ \uparrow \quad \text{carry} = 1 \end{array}$$

The sum is negative and it is in 2's complement. There is a carry through the sign bit. It is to be dropped.

The above result is in 2's complement. The correct result can be obtained as follows:

$$\begin{array}{lcl} \text{2's complement of 0010 (result)} & = 1110 \\ & = -14 \text{ (decimal)} \end{array}$$

Conclusion

The carry of the last stage is to be dropped. When the sum is positive, it is in the correct binary form. When the sum is negative it is in 2's complement. The sign bit indicates whether the sum is positive or negative. When the sum is in 2's complement, it will be converted to the correct form for display by the computer very easily. This is true only when there is no overflow. The case of overflow has been discussed later on in this chapter in Sec. 2.19.

In the above examples we have taken a separate sign bit. In a computer the most significant bit may be reserved for sign bit; and the rest of the bits of the data word will represent the magnitude of a number. Consider an 8-bit computer. The magnitude of the number is represented by 7 bits. The MSB represents the sign of the number. If the sign bit is 0, the number is positive. If the sign bit is 1, the number is negative. The negative numbers are represented in 2's complemented. The following example will illustrate the addition of signed binary numbers.

Example 5. Add + 14 and + 9

$$\begin{array}{r} +14 \quad 00001110 \\ +9 \quad 00001001 \\ \hline +23 \quad 00010111 \\ \uparrow \quad \text{Sign bit is 0, so the sum is positive.} \end{array}$$

Example 6. Add + 14 and (- 9).

$$\begin{array}{r}
 + 14 & 00001110 \\
 - 9 & 11110111 \text{ (2's complement of 9 with sign bit).} \\
 \hline
 + 5 & 100000101 (+ 5 decimal).
 \end{array}$$

↑ Sign bit is 0, so the result is positive.
 ↑ Carry = 1, it is neglected.

Example 7. Add (- 14) and + 9.

$$\begin{array}{r}
 - 14 & 11110010 \text{ (2's complement of 14 with sign bit).} \\
 + 9 & 00001001 \\
 \hline
 - 5 & 11111011 \text{ (2's complement of 5 with sign bit).}
 \end{array}$$

↑ Sign bit is 1, so the result is negative.
 00000100 1's complement of the result.
 + 1 Add 1
 ——————
 - 00000101 (-5 decimal).

Prefix – is to show the result in usual standard binary form.

Example 8. Add (- 14) and (- 9).

$$\begin{array}{r}
 - 14 & 11110010 \text{ (2's complement of 14 with sign bit).} \\
 - 9 & 11110111 \text{ (2's complement of 9 with sign bit).} \\
 \hline
 - 23 & 111101001 \text{ (2's complement of 23 with sign bit).}
 \end{array}$$

↑ Sign bit is 1, so the result is negative.
 ↑ Carry is 1, it is neglected.
 00010110 1's complement of the result.
 + 1 Add 1
 ——————
 - 00010111 (- 23 decimal).

Prefix – is to show the correct result in usual standard binary form.

The above examples show that the addition and subtraction in 2's complement is very simple. The sign bit clearly indicates whether the result is positive or negative. If it is negative it means that it is in 2's complement. Then the computer converts it to a standard decimal form to show it on the screen. The case of overflow is discussed in Sec. 2.19.

2.9 CONVERSION OF A BINARY FRACTION TO A DECIMAL FRACTION

In the decimal number system the weights of the digits which come after the decimal point, are represented as:

$$\begin{aligned}
 0.635 &= 0.6 + 0.03 + 0.005 \\
 &= 6 \times \frac{1}{10} + 3 \times \frac{1}{100} + 5 \times \frac{1}{1000} \\
 &= 6 \times 10^{-1} + 3 \times 10^{-2} + 5 \times 10^{-3}
 \end{aligned}$$

Similarly, in the binary number system the weights of the binary bits which come after the binary point, can be expressed as:

$$\begin{aligned} 0.1101 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \\ &= 1 \times \frac{1}{2} + 1 \times \frac{1}{4} + 0 \times \frac{1}{8} + 1 \times \frac{1}{16} \\ &= 0.5 + 0.25 + 0 + 0.0625 \\ &= 0.8125 \text{ (decimal)} \end{aligned}$$

Example 1. Convert the binary fraction 0.10111 to decimal fraction.

$$\begin{aligned} 0.10111 &= 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} + 1 \times 2^{-5} \\ &= 1 \times \frac{1}{2} + 0 \times \frac{1}{4} + 1 \times \frac{1}{8} + 1 \times \frac{1}{16} + 1 \times \frac{1}{32} \\ &= 0.5 + 0 + 0.125 + 0.0625 + 0.03125 \\ &= 0.71875 \text{ (decimal).} \end{aligned}$$

Example 2. Convert the binary real number 1101.1010 to a decimal real number.

A binary real number consists of two parts an integer and a fraction. The decimal equivalent is determined for both integer as well as fraction and these are added to give the equivalent decimal real number.

$$\begin{aligned} 1101.1010 &= (1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0) + (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &\quad + 0 \times 2^{-4}) \\ &= (8 + 4 + 0 + 1) + (0.5 + 0 + 0.125 + 0) \\ &= 13.625 \text{ (decimal)} \end{aligned}$$

2.10 CONVERSION OF A DECIMAL FRACTION TO A BINARY FRACTION

To convert a decimal fraction to its binary equivalent a technique of successive multiplication by 2 is used. The integer part is noted down after the multiplication by 2 at each stage and the remainder new fraction is used for the multiplication by 2 at the next stage. The following example will illustrate the procedure.

Example 1. Convert the decimal fraction 0.8125 to an equivalent binary fraction.

Fraction	Fraction $\times 2$	Remainder New Fraction	Integer
0.8125	1.625	0.625	1 (MSB)
0.625	1.25	0.25	1
0.25	0.50	0.50	0
0.50	1.00	0.00	1 (LSB)

$$0.8125 \text{ (decimal)} = 0.1101 \text{ (Binary)}$$

Example 2. Convert the decimal fraction 0.635 to its binary equivalent.

Fraction	Fraction $\times 2$	Remainder New Fraction	Integer
0.635	1.27	0.27	1 (MSB)
0.27	0.54	0.54	0
0.54	1.08	0.08	1

(Contd.)...

0.08	0.16	0.16	0
0.16	0.32	0.32	0
0.32	0.64	0.64	0
0.64	1.28	0.28	1 (LSB)
.	.	.	.
.	.	.	.
.	.	.	.

In this example it is seen that the fraction has not become zero, and the process will continue further. For such a case an approximation is made. For this example, we may take the result up to 6th binary bit after the binary point.

$$0.635 \text{ (decimal)} = 0.1010001 \text{ (binary)}$$

Example 3. Convert the decimal real number 12.625 to an equivalent binary real number.

For the above decimal real number the binary equivalent is obtained for both integer as well as fraction separately.

12 (decimal) is first converted to its binary equivalent.

Quotient	Remainder
$12 \div 2 = 6$	0 (LSB)
$6 \div 2 = 3$	0
$3 \div 2 = 1$	1
$1 \div 2 = 0$	1 (MSB)

$$12 \text{ (decimal)} = 1100 \text{ (binary)}.$$

Then the decimal fraction 0.625 is converted to its binary equivalent.

Fraction	$2 \times \text{Fraction}$	Remainder New Fraction	Integer
0.625	1.25	0.25	1 (MSB)
0.25	0.50	0.50	0
0.50	1.00	0.00	1 (LSB)

$$0.625 \text{ (decimal)} = 0.101 \text{ (binary)}.$$

Now adding the binary equivalents of 12 and 0.625, we get

$$12.625 \text{ (decimal)} = 1100.101 \text{ (binary)}$$

2.11 BINARY CODED DECIMAL (BCD CODES)

The BCD is the simplest binary code to represent a decimal number. In BCD code a decimal number is represented by four binary bits. For example, 3 is represented by 0011. If a decimal number consists of two or more than two digits, each decimal digit is individually represented by its 4-bit binary equivalent. For example, 56 is represented by 0101 0110.

Numbers are usually represented by some sort of binary codes. There is a difference between a binary equivalent of a decimal number and the binary code of a decimal number. For example, the binary equivalent of the decimal number 43 is 101011, but it is represented in BCD code as 0100 0011.

Table 2.3 shows the standard BCD codes for one-digit and two-digit decimal numbers.

Table 2.3. Standard BCD Codes

Decimal Number	Standard BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	0001 0000
11	0001 0001
12	0001 0010
13	0001 0011
56	0101 0110
84	1000 0100
99	1001 1001

In the standard BCD code the weights of four binary bits which represent an individual digit is 8, 4, 2, 1. At present modern computers perform subtraction using complements. There is difficulty in forming complements when numbers are represented by standard BCD. For example, the 1's complement of 0010 (2 decimal) is 1101 (13 decimal) which is not an acceptable BCD code in this system. To overcome this difficulty other BCD codes such as Excess-3 code; 2,4,2,1 code etc. have been used in earlier computers. In excess-3 code 3 is added to the individual digit of a decimal number, then their binary equivalents are written. For example, 5 (decimal) = 1000 (8 decimal) in excess-3 code. 253 (decimal) in excess-3 code will be 0101 1000 0110. The drawback of this code is that it is not a weighted code, that is the sum of weights of binary bits is not equal to the corresponding decimal digit.

Another BCD code is 2, 4, 2, 1 code. It is a weighted code and it has complements. For example, 5 is 0101; 9 = 1111, 7 = 1101.

BCD codes are used where the decimal information is directly (in coded form) transferred into or out of a digital system. Electronic calculators, digital voltmeters, frequency counters, electronic counters, digital clocks etc. work with BCD numbers. BCD codes have also been used in early computers. Modern computers do not use BCD numbers as they have to process names and other nonnumeric data.

In addition to difficulty in forming complements for binary subtraction there is also difficulty in performing addition in standard BCD. If the result lies in the range 10 to 15 or if there is carry from the 4th bit of any BCD digit, a correction of + 6 has to be made to obtain the correct result. This will be clear from the following example:

Example 1. Add 8 and 5.

BCD
8
+ 5
<hr/>
13
1000
0101
<hr/>
1101
Incorrect BCD
+
0110
<hr/>
00010011
Correct BCD 13

Example 2. Add 8 and 9.

BCD
8
+ 9
<hr/>
17
1000
1001
<hr/>
00010001
Incorrect BCD
+
0110
<hr/>
00010111
Correct BCD 17

2.12 HEXADECIMAL NUMBER SYSTEM

The hexadecimal number system is now extensively used in computer industry. Its radix (base) is 16. Its digits from 0 to 9 are same as those used by decimal number system. In the hexadecimal number system 10 is represented by A, 11 by B, 12 by C, 13 by D, 14 by E and 15 by F. The decimal number 16 is represented by 10, 17 by 11 and so on.

A hexadecimal digit is represented by four binary bits. For example, 5 is represented by 0101, A by 1010 and D by 1101. If a hexadecimal number consists of two or more than two digits, each digit is represented individually by four binary bits. For example, 86 is represented by 1000 0110, 5E by 0101 1110, 3B7 by 0011 1011 0111. Table 2.4 shows hexadecimal numbers and their binary representations. The binary representation of a hexadecimal number is also called binary coded hexadecimal number. The reason for using 4 binary bits to represent a hexadecimal digit is that the largest hexadecimal digit 'F' requires only 4 binary bits for its representation.

Let us examine the binary equivalent of a decimal number and its hexadecimal representation. The binary equivalent of the decimal number 94 is 1011110. The hexadecimal equivalent of 94 (decimal) is 5E. The binary representation of 5E is 0101 1110. If we neglect the leading zero, that is zero in MSB position, both binary representations are identical.

Table 2.4. Hexadecimal Numbers and their Binary Representations

<i>Decimal Number</i>	<i>Hexadecimal Number</i>	<i>Binary Coded Hexadecimal Number</i>
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101

(Contd.)...

6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000
37	25	0010 0101
43	2B	0010 1011
94	5E	0101 1110
223	DF	1101 1111
255	FF	1111 1111

2.12.1 Hexadecimal Versus BCD

From Table 2.4 it is clear that the hexadecimal system utilizes the full capacity of four binary bits, whereas BCD codes do not utilize the same. The BCD codes do not utilize the binary codes from 1010 to 1111. In the hexadecimal system an 8-bit word can represent up to FF, that is 11111111 (255 decimal) whereas in BCD only up to 10011001 (99 decimal). Hence the hexadecimal is a compact form of representation, and it occupies less memory space, thereby reducing the hardware cost. The arithmetic operations are also simpler in hexadecimal system.

When a computer gives hardware or software error it may be required to see the information stored in the memory. The memory stores information in binary form which is very difficult to examine. Therefore, the stored information is printed out in hexadecimal form which is very compact and easy to examine. Computers which work in binary produce memory dumps in hexadecimal as printouts that are very compact. Listings are also produced in hexadecimal.

Usually, the input data and the result (output) are in the decimal form. Before a binary processor (hexadecimal or octal) performs arithmetic operations the input data have to be converted into binary form (hexadecimal or octal). Again results which are in the binary form are to be converted to decimal form. The binary system will be efficient for small input-output requirements and a large number of computations. A processor working with BCD will be efficient if large input-output requirements are involved and small number of computations are required. The computers using BCD codes perform arithmetic operations directly on decimal data (BCD codes) and eliminate the requirement for conversion to binary and back to decimal.

Many decimal fractions do not have exact binary equivalents. In some business computers dealing with accounting applications precise calculations are desired. Therefore, BCD may be desired for the exact representation of decimal numbers including the digits after decimal points. But BCD arithmetic takes more time for execution and hence it is slower than binary

arithmetic. Thus the price paid for precise decimal fractions is less efficient use of memory and slower processing.

Some computers and all calculators employ BCD codes. Most of the modern computers use hexadecimal system. Some large computers may have the hardware facility for arithmetic calculations with both binary (hex.) as well as BCD.

2.12.2 Conversion of a Hexadecimal Number to a Decimal Number

For the conversion of a hexadecimal number to an equivalent decimal number the following well known expression for the weights of digits of the number will be used:

The weight of n th digit of the number from the right hand side = n th digit \times (Base) $^{n-1}$.

The following examples will illustrate the procedure.

Example 1. Convert the hexadecimal number 4B8 to its equivalent decimal number.

8 is the 1st digit of the number from the right hand side, therefore, its weight is 8×16^0 .

B is the 2nd digit of the number from the right hand side, therefore, its weight is $B \times 16^1$.

4 is the 3rd digit of the number from the right hand side, therefore, its weight is 4×16^2 .

$$\begin{aligned}\text{Therefore, } 4B8 \text{ (hex.)} &= 4 \times 16^2 + B \times 16^1 + 8 \times 16^0 \\ &= 4 \times 256 + 11 \times 16 + 8 \times 1 \\ &= 1024 + 176 + 8 \\ &= 1208 \text{ (decimal).}\end{aligned}$$

Example 2. Convert the hexadecimal number 6E to its decimal equivalent.

$$\begin{aligned}6E \text{ (hex.)} &= 6 \times 16^1 + E \times 16^0 \\ &= 6 \times 16 + 14 \times 1 \\ &= 94 + 14 \\ &= 110 \text{ (decimal).}\end{aligned}$$

Example 3. Convert the hexadecimal number 2B6D to its equivalent decimal number.

$$\begin{aligned}2B6D \text{ (hex.)} &= 2 \times 16^3 + B \times 16^2 + 6 \times 16^1 + D \times 16^0 \\ &= 2 \times 4096 + 11 \times 256 + 6 \times 16 + 13 \times 1 \\ &= 8192 + 2816 + 96 + 13 \\ &= 11117 \text{ (decimal).}\end{aligned}$$

2.12.3 Conversion of a Hexadecimal Fraction to a Decimal Fraction

In the hexadecimal system the weights of the hexadecimal digits after the hexadecimal point are as follows:

$$\begin{aligned}0.5A6B &= 5 \times 16^{-1} + A \times 16^{-2} + 6 \times 16^{-3} + B \times 16^{-4} \\ &= 0.3125 + 0.0390625 + 0.0014648437 + 0.00016784667 \\ &= 0.35319519037 \text{ (hexadecimal).}\end{aligned}$$

2.12.4 Conversion of a Decimal Number to a Hexadecimal Number

For the Conversion of a decimal number to an equivalent hexadecimal number, the decimal number is divided by 16 successively.

Example 1. Convert the decimal number 67 to an equivalent hexadecimal number.

<i>Quotient</i>	<i>Remainder</i>
$67 \div 16 = 4$, 3 (LSD)
$4 \div 16 = 0$, 4 (MSD)

The decimal number $67 = 43$ (hexadecimal).

Example 2. Convert the decimal number 952 to its hexadecimal equivalent.

<i>Quotient</i>	<i>Remainder</i>
$952 \div 16 = 59$, 8 (LSD)
$59 \div 16 = 3$, 11 = B
$3 \div 16 = 0$, 3 (MSD)

The decimal number $952 = 3B8$ (hexadecimal).

2.12.5 Conversion of a Decimal Fraction to a Hexadecimal Fraction

For the conversion of decimal fraction to its equivalent hexadecimal fraction the technique of repeated multiplication by 16 is used. The integer part is noted down after each multiplication and the new remainder fraction is used for multiplication at the next stage.

Example. Convert the decimal fraction 0.62 to its equivalent hexadecimal fraction.

<i>Fraction</i>	<i>Fraction $\times 16$</i>	<i>Remainder</i>	<i>Integer</i>
0.62	9.92	0.92	9 (MSD)
0.92	14.72	0.72	14 = E
0.72	11.52	0.52	11 = B
0.52	8.32	0.32	8
0.32	5.12	0.12	5
0.12	1.92	0.92	1 (LSD)

The process will further continue. Therefore, the result has been taken upto 6 places of hexadecimal point.

The decimal fraction $0.62 = 0.9EB851$ (hex.) approximately.

2.12.6 Conversion of a Binary Number to a Hexadecimal Number

The conversion of binary number to hexadecimal number uses a very simple technique. The base of the hexadecimal number system is 16. To convert a binary number to hexadecimal, 4-bit binary groups (each group contains 4 binary bits) are formed in the binary number. After forming the groups, each group of 4 binary bits is converted to its hexadecimal equivalent.

Example 1. Convert the binary number 01101110 to its equivalent hexadecimal number.

$$\begin{aligned} (01101110)_2 &= (0110) (1110) \\ &= 6E \text{ (hex.).} \end{aligned}$$

Example 2. Convert the binary number 1101001101 to its equivalent hexadecimal.

The formation of groups of 4 bits each in an integer binary number is made from right hand side to left.

$$(1101001101)_2 = (11) (0100) (1101)$$

In the above expression we see that the group of the most significant binary bits contains only 2 binary bits. This can be extended to 4 binary bits by adding zeros in MSB positions. If MSBs are extended by adding zeros the number remains unaffected. Therefore, the given number may be grouped as follows:

$$(1101001101)_2 = (0011) (0100) (1101) \\ = 34D \text{ (hex.)}$$

Example 3. Convert the binary real number $1011100 . 1000101$ to its equivalent hexadecimal real number.

In case of binary fraction, the formation of grouping of binary bits which are after the binary point, is made from left to right. For the integer part group is made from right to left.

$$1011100.1000101 = (101) (1100).(1000) (101) \\ = (0101) (1100).(1000) (1010) \\ = 5C.8A \text{ (hex.)}$$

2.12.7 Conversion of a Hexadecimal Number to a Binary Number

To convert a hexadecimal number to its equivalent binary number each digit of the given hexadecimal number is converted to its 4-bit binary equivalent.

Example 1. Convert the hexadecimal number 6B9 to its equivalent binary number.

$$(6B9)_{16} = (0110) (1011) (1001) \\ = (011010111001)_2 \\ = (11010111001)_2$$

Example 2. Convert the real hexadecimal number 6D.3A to its equivalent binary number.

$$(6D.3A)_{16} = (0110) (1101).(0011) (1010) \\ = (01101101.00111010)_2 \\ = (1101101.0011101)_2$$

2.13 OCTAL NUMBER SYSTEM

The base of the octal number system is 8. This system is also used in computer industry. It uses eight digits 0, 1, 2, 3, 4, 5, 6 and 7. The decimal number 8 is represented by 10, 9 by 11, 10 by 12 and so on in the octal number system. An octal number is represented by a group of three binary bits. For example, 4 is represented by 100, 6 by 110 and 7 by 111. The reason for using 3 binary bits to represent an octal number is that the largest octal digit is 7 which can be represented by only 3 binary bits i.e., $7 = 111$. If an octal number contains two or more than two digits, each digit is individually represented by a group of three binary bits. For example, 46 is represented by 100 110 and 354 by 011 101 100. Table 2.5 shows octal numbers and their binary representations. The binary representation of an octal number is also called binary coded octal number.

Let us examine the binary equivalent and binary representations of a decimal number in octal and hexadecimal. The binary equivalent of the decimal number 45 is 101101. The decimal number 45 is 2D hexadecimal and 55 octal respectively. 2D (hex.) can be represented in binary as 00101101, and 55 (octal) as 101101. If neglect zeros in MSB positions, all the three binary representations are same.

Table. 2.5 Octal Number and their Binary Representations

<i>Decimal Number</i>	<i>Octal Number</i>	<i>Binary Coded Octal Number</i>
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
8	10	001 000
10	12	001 010
15	17	001 111
16	20	010 000
61	75	111 101
63	77	111 111
64	100	001 000 000
100	144	001 100 100

2.13.1 Conversion of an Octal Number to a Decimal Number

For the conversion of a number from any number system to decimal number system we make the use of the following well known expression:

The weight of the n -th digit of the number from right hand side = n th digit \times (Base) $^{n-1}$.

Example 1. Convert the octal number 56 to its equivalent decimal number.

As the base of the octal number system is 8, applying the general rule of conversion, we get

$$\begin{aligned} 56 \text{ (octal)} &= 5 \times 8^1 + 6 \times 8^0 \\ &= 40 + 6 \\ &= 46 \text{ (decimal)} \end{aligned}$$

Example 2. Convert the octal number 364 to its decimal equivalent.

$$\begin{aligned} 364 \text{ (octal)} &= 3 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 \\ &= 3 \times 64 + 6 \times 8 + 4 \times 1 \\ &= 192 + 48 + 4 \\ &= 244 \text{ (decimal)} \end{aligned}$$

2.13.2 Conversion of an Octal Fraction to a Decimal Fraction

In the octal system the weight of the octal digits after the octal point are as explained below:

$$0.563 \text{ (octal)} = 5 \times 8^{-1} + 6 \times 8^{-2} + 3 \times 8^{-3}$$

$$\begin{aligned}
 &= 0.625 + 0.09375 + 0.005859375 \\
 &= 0.724609375 \text{ (decimal)}
 \end{aligned}$$

2.13.3 Conversion of a Decimal to an Octal Number

For the conversion of a decimal number to an equivalent octal number, the technique of repeated division by 8 is used.

Example 1. Convert the decimal number 62 to its equivalent octal number.

<i>Quotient</i>	<i>Remainder</i>
62 ÷ 8 = 7	, 6 (LSD)
7 ÷ 8 = 0	, 7 (MSD)
62 (decimal)	= 76 (octal)

Example 2. Convert the decimal number 958 to its equivalent octal number.

<i>Quotient</i>	<i>Remainder</i>
958 ÷ 8 = 119	, 6 (LSD)
119 ÷ 8 = 14	, 7
14 ÷ 8 = 1	, 6
1 ÷ 8 = 0	, 1 (MSD)
958 (decimal)	= 1676 (octal)

Checking the answer:

$$\begin{aligned}
 1676 \text{ (Octal)} &= 1 \times 8^3 + 6 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 \\
 &= 512 + 384 + 56 + 6 \\
 &= 958 \text{ (decimal)}
 \end{aligned}$$

2.13.4 Conversion of a Decimal Fraction to an Octal Fraction

For the conversion of a decimal fraction to its equivalent octal fraction the technique of repeated multiplication by 8 is used. The integer part is noted down and new remainder fraction is used for the multiplication at the next stage.

Example 1. Convert the decimal fraction 0.96 to its equivalent octal fraction.

<i>Fraction</i>	<i>Fraction × 8</i>	<i>Remainder Fraction</i>	<i>New Integer</i>
0.96	7.68	0.68	7 (MSD)
0.68	5.44	0.44	5
0.44	3.52	0.52	3
0.52	4.16	0.16	4
0.16	1.28	0.28	1 (LSD)

The process will continue further so we may take the result upto 5 places of octal point.

$$0.96 \text{ (decimal)} = 0.75341 \text{ (octal) approximately}$$

2.13.5 Conversion of a Binary Number to an Octal Number

The octal number system is a base-8 system. For binary to octal conversion, groups of 3 binary bits each are formed in the binary number. After forming the groups, each group of 3 binary bits is converted to its octal equivalent.

Example 1. Convert the binary number 101110 to its equivalent octal number.

The formation of groups of 3 bits each in an integer binary number is made from right to left.

$$\begin{aligned}(101110)_2 &= (101) (110) \\ &= 56 \text{ (octal)}\end{aligned}$$

Example 2. Convert the binary number 1101011 to its equivalent octal number.

$$(1101011)_2 = (1) (101) (011)$$

As the leftmost group consists of only one binary bit, this group is extended by adding zeros in MSBs.

$$\begin{aligned}(1101011)_2 &= (001) (101) (011) \\ &= 153 \text{ (octal)}\end{aligned}$$

Example 3. Convert the binary real number 1011.1011 to its equivalent octal number.

In the integer part of the binary number the group of 3 bits is formed from right to left. In the binary fraction the group of 3 bits is formed from left to right.

$$\begin{aligned}(1011.1001)_2 &= (1) (011).(101) (1) \\ &= (001) (011).(101) (100) \\ &= 13.54 \text{ (octal).}\end{aligned}$$

2.13.6 Conversion of an Octal Number to a Binary Number

To convert an octal number to its equivalent binary number each digit of the given octal number is converted to its 3-bit binary equivalent.

Example 1. Convert the octal number 376 to its equivalent binary number.

$$\begin{aligned}(376)_8 &= (011) (111) (110) \\ &= (01111110)_2 \\ &= (11111110)_2\end{aligned}$$

Example 2. Convert the real octal number 56.34 to its equivalent binary number.

$$\begin{aligned}(56.34)_8 &= (101) (110).(011) (100) \\ &= (101110.011100)_2 \\ &= (101110.0111)_2\end{aligned}$$

2.13.7 Conversion of an Octal Number to a Hexadecimal Number and Vice-Versa

The conversion of a hexadecimal number to its equivalent octal number or vice-versa can easily be done through binary as illustrated by the example given below.

Example 1. Convert the hexadecimal number 3DE to its equivalent octal number.

The hexadecimal number 3DE is first converted to its binary equivalent.

$$\begin{aligned}3DE \text{ (hex)} &= (0011) (1101) (1110) \\ &= (001111011110)_2\end{aligned}$$

Now the above binary equivalent is divided into groups of 3 bits to obtain its octal equivalent.

$$\begin{aligned}001111011110 &= (001) (111) (011) (110) \\ &= 1736 \text{ (octal)}\end{aligned}$$

Example 2. Convert the real hexadecimal number 5B.3A to its equivalent octal number. The number 5B.3A is first converted to its binary equivalent.

$$\begin{aligned} 5B.3A &= (0101) (1011).(0011) (1010) \\ &= 01011011.00111010 \end{aligned}$$

Now forming the groups of 3 binary bits to obtain its octal equivalent we have,

$$\begin{aligned} 01011011.00111010 &= (01) (011) (011).(001) (110) (10) \\ &= (001) (011) (011).(001) (110) (100) \\ &= 133.164 \text{ (octal).} \end{aligned}$$

Example 3. Convert the octal number 536 to its equivalent hexadecimal number.

Converting 536 (octal) first to its binary equivalent, we get,

$$\begin{aligned} (536)_8 &= (101) (011) (110) \\ &= (101011110)_2 \end{aligned}$$

Now forming the groups of 4 binary bits to obtain its hexadecimal equivalent we get,

$$\begin{aligned} (101011110) &= (1) (0101) (1110) \\ &= (0001) (0101) (1110) \\ &= 15E \text{ (hex).} \end{aligned}$$

Example 4. Convert the real octal number 46.57 to its equivalent hexadecimal number.

Converting 46.57 (octal) first to its binary equivalent we get,

$$\begin{aligned} 46.57 &= (100) (110).(101) (111) \\ &= (100110.101111)_2 \end{aligned}$$

Now forming the groups of 4 binary bits to obtain its hexadecimal equivalent we have,

$$\begin{aligned} 100110.101111 &= (10) (0110).(1011) (11) \\ &= (0010) (0110).(1011) (1100) \\ &= 26.BC \text{ (hex).} \end{aligned}$$

2.14 ASCII AND ISCII CODES

ASCII is pronounced as “ask-ee”. It stands for American Standard Code for Information Interchange. ASCII code is used extensively in small computers, peripherals, instruments and communication devices. It has replaced many of the special codes that were previously used by manufacturers. It is a 7-bit code. Microcomputers using 8-bit word length use 7 bits to represent the basic code. The 8th bit is used for parity or it may be permanently 1 or 0. Table 2.6 shows ASCII codes. With 7 bits up to 128 characters can be coded. A letter, digit or special symbol is called a character. It includes upper-and lower-case alphabets, numbers, punctuation marks, special characters and control characters. Table 2.7 shows the definitions of control characters. Some control characters which are used in serial communications are: ENQ, ACK, NAK, etc. ENQ is used for enquiry. ACK is for acknowledgement. It is used to indicate successful reception after completing error checking. NAK is negative acknowledgement, used to indicate errors in reception.

ISCII stands for Indian Standard Code for Information Interchange. It is an 8-bit code for Indian languages.

Table 2.6 ASCII Codes

<i>Characters or Abbreviations</i>	<i>Hex codes for 7-bit ASCII</i>	<i>Characters or Abbreviations</i>	<i>Hex Codes for 7-bit ASCII</i>
NUL	00	&	26
SOH	01	'	27
STX	02	(28
ETX	03)	29
EOT	04	*	2A
ENQ	05	+	2B
ACK	06	,	2C
BEL	07	-	2D
BS	08	.	2E
HT	09	/	2F
LF	0A	'	60
VT	0B	:	3A
FF	0C	;	3B
CR	0D	<	3C
SO	0E	=	3D
SI	0F	>	3E
DLE	10	?	3F
DC1	11	@	40
DC2	12	[5B
DC3	13	\	5C
DC4	14]	5D
NAK	15	^	5E
SYN	16	—	5F
ETB	17	{	7B
CAN	18		7C
EM	19	}	7D
SUB	1A	~	7E
ESC	1B	DEL	7F
FS	1C	a	61
GS	1D	b	62
RS	1E	c	63
US	1F	d	64
SP	20	e	65
!	21	f	66
"	22	g	67
#	23	h	68
\$	24	i	69
%	25	j	6A

(Contd.)...

k	6B	K	4B
l	6C	L	4C
m	6D	M	4D
n	6E	N	4E
o	6F	O	4F
p	70	P	50
q	71	Q	51
r	72	R	52
s	73	S	53
t	74	T	54
u	75	U	55
v	76	V	56
w	77	W	57
x	78	X	58
y	79	Y	59
z	7A	Z	5A
A	41	0	30
B	42	1	31
C	43	2	32
D	44	3	33
E	45	4	34
F	46	5	35
G	47	6	36
H	48	7	37
I	49	8	38
J	4A	9	39

Table 2.7 Definitions of Control Characters

NUL	NULL	DC2	DIRECT CONTROL 2
SOH	START OF HEADING	DC3	DIRECT CONTROL 3
STX	START TEXT	DC4	DIRECT CONTROL 4
ETX	END TEXT	NAK	NEGATIVE ACKNOWLEDGE
EOT	END OF TRANSMISSION	SYN	SYNCHRONOUS IDLE
ENQ	ENQUIRY	ETB	END TRANSMISSION BLOCK
ACK	ACKNOWLEDGE	CAN	CANCEL
BEL	BELL	EM	END OF MEDIUM
BS	BACKSPACE	SUB	SUBSTITUTE
HT	HORIZONTAL TAB	ESC	ESCAPE
LF	LINE FEED	FS	FORM SEPARATOR
VT	VERTICAL TAB	GS	GROUP SEPARATOR
FF	FORM FEED	RS	RECORD SEPARATOR
CR	CARRIAGE RETURN	US	UNIT SEPARATOR
SO	SHIFT OUT		
SI	SHIFT IN		
DLE	DATA LINK ESCAPE		
DC1	DIRECT CONTROL 1		

2.14.1 ASCII-8

A newer version of ASCII is known as ASCII-8. It is an 8-bit code. With 8 bits code capacity is extended to 256 characters. Table 2.8 shows ASCII-8 codes.

Table 2.8 ASCII-8 Codes

Characters	ASCII-8 Codes	Characters	ASCII-8 Codes
0	50	J	AA
1	51	K	AB
2	52	L	AC
3	53	M	AD
4	54	N	AE
5	55	O	AF
6	56	P	B0
7	57	Q	B1
8	58	R	B2
9	59	S	B3
A	A1	T	B4
B	A2	U	B5
C	A3	V	B6
D	A4	W	B7
E	A5	X	B8
F	A6	Y	B9
G	A7	Z	BA
H	A8		
I	A9		

2.15 EBCDIC CODE

It is pronounced as “ebb-see-dick”. It is Extended Binary-Coded Decimal Interchange Code. It is the standard character code for large computers. It is an 8-bit code without parity. A ninth bit can be used for parity. With 8 bits up to 256 characters can be coded. Table 2.9 shows EBCDIC codes.

In ASCII-8 and EBCDIC first 4 bits are known as zone bits and remainder 4 bits represent digit values. In ASCII first 3 bits are zone bits and remaining 4 bits represent digit values. When an ASCII-8 (or any other code) character is entered, the computer receives the binary equivalent of the hex code for ASCII-8. For example, when ASCII-8 key for digit 9 is pressed, the computer receives the binary equivalent of 59 hex, which must be converted to the binary 1001 for arithmetic operations. Before arithmetic operations data must be converted to proper format which is acceptable to arithmetic logic circuitry. The data also indicate whether they are positive or negative or unsigned.

2.16 GRAY CODES

The Gray Code is a binary code and is shown in Table 2.10. It is used in shaft encoder which indicates the angular position of a shaft in digital form. The binary bits are arranged

Table 2.9 EBCDIC Codes

<i>Characters or Control Characters</i>	<i>Hex Codes for EBCDIC</i>	<i>Characters or Control Characters</i>	<i>Hex Codes for EBCDIC-</i>
NUL	00	¢	4A
SOH	01	.	4B
STH	02	,	6B
ETS	03	?	6F
HT	05	:	7A
DEL	07	;	5E
VT	0B	!	5A
FF	0C	,	7D
CR	0D	"	7F
SO	0E	+	4E
SI	0F	-	60
DLE	10	—	6D
DC1	11	*	5C
DC2	12	/	61
DC3	13	=	7E
RES	14	<	4C
NL	15	>	6E
BS	16	(4D
CAN	18)	5D
EM	19	{	8B
FLS	1C	}	9B
GS	1D	[AD
RDS	1E]	DD
US	1F		4F
BYP	24	&	50
LF	25	\$	5B
EOB	26]	5F
ENQ	2D	%	6C
ACK	2E	#	7B
BEL	2F	@	7C
SYN	32	a	81
DC4	35	b	82
EOT	37	c	83
NAK	3D	d	84
SUB	3F	e	85
SP	40	f	86
BLANK	E0	g	87

(Contd.)...

h	88	J	D1
i	89	K	D2
j	91	L	D3
k	92	M	D4
l	93	N	D5
m	94	O	D6
n	95	P	D7
o	96	Q	D8
p	97	R	D9
q	98	S	E2
r	99	T	E3
s	A2	U	E4
t	A3	V	E5
u	A4	W	E6
v	A5	X	E7
w	A6	Y	E8
x	A7	Z	E9
y	A8	0	F0
z	A9	1	F1
A	C1	2	F2
B	C2	3	F3
C	C3	4	F4
D	C4	5	F5
E	C5	6	F6
F	C6	7	F7
G	C7	8	F8
H	C8	9	F9
I	C9		

in such a way that only one binary bit changes at a time when we make a change from any number to the next. Its use reduces the error in reading shaft position. The largest possible error will be one least significant digit. Suppose that the shaft changes its position from 7 to 8, the Gray code will change from 0100 to 1100. If the binary bit is not picked up by the sensor, the encoder, will show the previous position that is 7. But, in case of standard binary code 7 is represented by 0111 and 8 by 1000. If sensor picks up the least significant bits that is 000, but fails to pick up the most significant bit 1, the output will be 0000 instead of 1000. There will be large error if standard binary code is used for shaft encoding. In this case an error of the most significant bit may occur. The Gray code is often used in computer controlled machines such as lathes etc. Photoelectric coders or shaft position encoder disks are used as sensors. For details the reader may consult Ref. 5 and 10 (for books).

Table 2.10 Gray and Excess-3 Gray Codes

<i>Decimal</i>	<i>Gray Codes</i>	<i>Excess-3 Gray Codes.</i>
0	0000	0010
1	0001	0110
2	0011	0111
3	0010	0101
4	0110	0100
5	0111	1100
6	0101	1101
7	0100	1111
8	1100	1110
9	1101	1010
10	1111	
11	1110	
12	1010	
13	1011	
14	1001	
15	1000	

The Gray code is not suitable for a decimal code. This is due to the fact that the transition from 9 back to 0 involves a change of three bits (from 1101 to 0000). To overcome this difficult Excess-3 Gray code as shown in Table 2.10 is used. In this case for a change from 9 to 0, the corresponding change in excess-3 code Gray code will be from 1010 to 0010, that is a change of only one bit.

2.17 FIXED POINT REPRESENTATION OF NUMBERS

In a fixed-point system of number representation all numbers are represented as integers or fraction. Signed integers or BCD numbers are referred to as fixed point numbers because they contain no information regarding the location of the binary point or decimal point. The binary or decimal point is assumed at the extreme right or left of the number. If the binary or decimal point is at the extreme right of the computer word, all numbers are positive or negative integers. If the radix point is assumed to be at the extreme left, all numbers are positive or negative fraction. Suppose one has to multiply 5.85×90.38 . This will be represented as 585×9038 . The result will be 5287230. The decimal point has to be placed by the programmer to get the correct result, that is 528.7230. Thus in the fixed-point of representation the user has to keep the track of radix point which is a tedious job.

In scientific applications of computers fractions are frequently used. So a system of representation which automatically keeps track of the position of the binary or decimal point is required. Such a system of representation is called floating point representation of numbers. This is discussed in the next section. Many computers and all electronic calculators use floating-point arithmetic operations. The computers which do not have internal circuitry for floating-point operations can solve the scientific problems involving fractions with the help of floating-point software.

2.18 FLOATING-POINT REPRESENTATION

A number which has both an integer part as well as a fractional part is called real number or floating-point number. A real number may be either positive or negative. Examples of real decimal numbers are 215.58, 0.739, - 516.46, - 0.586 etc. Examples of binary real numbers are 101.110, 0.1011, - 101.1011, - 0.1101 etc.

The real number 546.98 can also be written as 5.4698×10^2 , 0.009863 as 9.863×10^{-3} , 146.58 as 0.14658×10^3 etc. Such representation is known as scientific form of representation.

In scientific computations it is often necessary to carry out calculations with very large or very small numbers. Hence, scientists have used a convenient notation in which a number is expressed as a combination of a mantissa and an exponent. For example 350000 may be written as 0.35×10^6 , where 0.35 is the mantissa and 6 is the value of the exponent.

In the general form a number N may be written as $N = MR^e$, where M is the mantissa, e the exponent and R the radix of the number system.

In a computer a real or floating-point number is represented by two parts: mantissa and exponent. The first part, the mantissa, is a signed fixed point number. The second part, the exponent indicates the position of the binary or decimal point. For example, the decimal number 3584.69 is represented in floating-point representation as shown below.

Sign	Sign
$0 . \underbrace{358469}_{\text{Mantissa}}$	$0 \underbrace{04}_{\text{Exponent}}$

A zero in the leftmost position of the mantissa denotes plus sign. The mantissa may be either a fraction or integer. Most computers use fractional system of representation for mantissa, but some manufacturers use the integer system. The decimal point shown above is an assumed decimal point. It is not physically indicated in the register. The exponent shown above is +4 to indicate that the actual position of the decimal point lies four decimal positions to the right of the assumed decimal point. In the above illustration the mantissa has been shown as a fraction.

If we use the integer system of representation for mantissa the number $3584.69 = 358469 \times 10^{-2}$ will be represented as shown below:

Sign	Sign
$0 \underbrace{358469}_{\text{Mantissa}}$	$1 \underbrace{02}_{\text{Exponent}}$

In this representation sign of the exponent has been shown negative to indicate that the actual position of the decimal point lies two decimal position to the left of the assumed decimal point.

The decimal number $0.0049586 = 0.49586 \times 10^{-2}$ can be represented in the fractional system of representation for mantissa as illustrated below:

Sign	Sign
$0 . \underbrace{49586}_{\text{Mantissa}}$	$1 \underbrace{02}_{\text{Exponent}}$

A negative number -563.5896 can be represented as shown below:

Sign	Sign
$1 \ .5635896$	$0 \ 03$
Mantissa	Exponent

A negative fraction -0.000258637

$= -0.258637 \times 10^{-3}$ can be represented as

Sign	Sign
$1 \ .258637$	$1 \ 03$
Mantissa	Exponent

The floating-point binary number is also represented in the same manner. For example, the binary number 1011.1010 can be represented in a 16-bit register as shown below:

Sign	Sign
$0 \ .10111010$	$1 \ 000100$
Mantissa	Exponent

The mantissa occupies 9 bits (including sign bit) and the exponent 7 bits (including sign bit). The binary point is not physically indicated in the register, but it is only assumed to be there.

In the general form a floating-point number is expressed as

$$N = M \times R^e$$

The mantissa M and the exponent e are physically present in a register of a computer. But, the radix R and the radix-point (decimal or binary point) are not indicated in the register. These are assumed things which are taken into account by the electronic circuitry which makes computation and manipulation.

2.18.1 Normalized Floating-Point Number

If the most significant digit of the mantissa is nonzero, the floating-point number is said to be normalized. For example, 0.00038695×10^5 and 0.058638×10^{-6} are not normalized numbers. The corresponding normalized numbers will be 0.38695×10^2 and 0.58638×10^{-7} are normalized numbers. Similarly, the binary numbers 0.10110×2^6 and 0.11010×2^{-7} are normalized numbers. The binary numbers 0.0010110×2^8 and 0.0011010×2^{-5} are not normalized numbers. While storing the floating-point numbers the mantissa is shifted left until its most significant bit becomes nonzero, and the exponent is adjusted accordingly. In other words the floating-point numbers are stored in the normalized (or standard) form. The process of shifting of mantissa left to make its most significant bit nonzero is called normalization.

2.18.2 Range of Stored Numbers

The range of the stored numbers depends on the word length of the computer and the number of bits assigned for the mantissa and the number of bits assigned for the exponent. Suppose, a 16-bit word is employed to store numbers. Out of 16 bits, 9 bits are used for the storage of mantissa including sign bit and 7 bits for exponent including sign bit. The maximum value of the number which can be stored is shown in Fig. 2.1.

The maximum value = 0.11111111 E 0111111

Where E indicates exponent; 0 after E indicates that the exponent is positive.

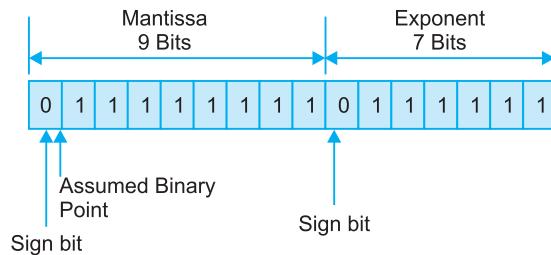


Fig. 2.1 Maximum value of stored number.

The maximum value

$$\begin{aligned}
 &= (1 - 2^{-8}) \times 2^{(2^6 - 1)} \\
 &= (1 - 2^{-8}) \times 2^{63} \\
 &\approx 1 \times 2^{63} \\
 &= 10^{19}
 \end{aligned}$$

The minimum value of the number which can be stored is shown in Fig. 2.2.

The minimum value = 0.10000000 E 1111111

The bit 1 just after E indicates that the exponent is negative.

The minimum value

$$\begin{aligned}
 &= (2^{-1}) \times 2^{-(2^6 - 1)} \\
 &= (2^{-1}) \times 2^{-63} \\
 &= 2^{-64} \\
 &\approx 10^{-19}
 \end{aligned}$$

Then the range will be 2^{-64} to 2^{63}

or 10^{-19} to 10^{19}

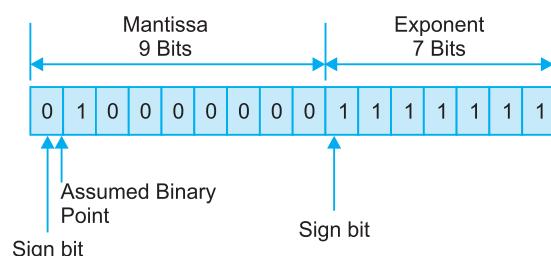


Fig. 2.2 Minimum value of stored number.

Similarly, the range can be obtained for 32-bit word length. Suppose out of 32 bits, 24 bits are employed for the mantissa including sign bit and 8 bits for the exponent including sign bit.

$$\begin{aligned}
 \text{The maximum value} &= (1 - 2^{-23}) \times 2^{(2^7 - 1)} \\
 &\approx 2^{127} \\
 &\approx 10^{38}
 \end{aligned}$$

$$\begin{aligned}\text{The minimum value} &= (2^{-1}) \times 2^{-(2^7-1)} \\ &= (2^{-1}) \times 2^{-127} \\ &= 2^{-128} \\ &\approx 10^{-38}\end{aligned}$$

The range of the numbers = 2^{-128} to 2^{127}
or 10^{-38} to 10^{38}

Biased Exponent

While determining the range of numbers which a 32-bit CPU can handle, we have seen that the exponent varies from -128 to $+127$. The exponent is negative, when a number is less than one. To eliminate the negative sign of the exponent, we can add a constant $+128$ to the exponent. This modification eliminates the need of a sign bit for the exponent. The modified technique is known as **biased exponent** technique. In the modified technique 8 bits are used to represent a exponent instead of 7 bits.

Example 1. Represent a binary number 1101011 in the floating-point representation using 32-bit word length. 24 bits represent mantissa and 8 bits represent exponent.

The binary number 1101011 = 0.1101011×2^7

For the above number, mantissa = 0.1101011

The exponent = 7.

For biased representation the exponent can be modified. The modified exponent = $7 + 128 = 135$

$$135 = 10000111$$

Since the mantissa is positive, its sign bit will be 0. The 32-bit representation of the given number is

Sign bit



01101011 00000000 00000000

↑.....↑

Mantissa
|

10000111

↑.....↑

Exponent
|

Example 2. Represent -1101011 in the floating-point representation for 32-bit word length.

The given binary number = -0.1101011×2^7

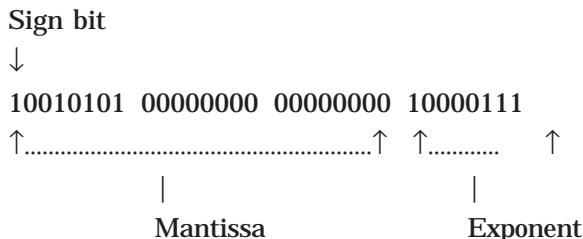
Since the given number is negative, its 2's complement is used in the representation. Its 2's complement is 0010101.

For the above negative number, mantissa = 0.0010101

Exponent = 7. The biased exponent = $7 + 128 = 135$

$$135 = 10000111$$

Since the given number is negative, the sign bit for the mantissa will be 1. The 32-bit representation of the given number is



2.18.3 Floating-Point Arithmetic Operations

Arithmetic operations with floating-point number are more complicated compared to those with fixed-point numbers. More complex hardware circuitry is required for their processing. Also, their processing takes longer time. When addition or subtraction is to be performed with normalized floating-point numbers, the exponents of the two numbers must be made equal. To do this the mantissa of the smaller number is shifted right by a number of places which is equal to the difference in exponents. The process of making exponents equal is known as *scaling* or *alignment* of numbers. After making the exponents equal the mantissas are added. After addition or subtraction the results are normalized. The process will be clear by the example given below:

Example 1. Add $.586293 \times 10^5$ and $.235000 \times 10^3$.

Before adding these numbers, their exponents are made equal.

$$\begin{array}{r} .586293 \times 10^5 \\ + .002350 \times 10^5 \\ \hline .588643 \times 10^5 \end{array}$$

Example 2. Add $.8794 \times 10^6$ and $.7635 \times 10^6$

$$\begin{array}{r} .8794 \times 10^6 \\ + .7635 \times 10^6 \\ \hline 1.6429 \times 10^6 \\ = .16429 \times 10^7 \end{array}$$

Example 3. Subtract $.39 \times 10^3$ from $.4935 \times 10^5$.

$$\begin{array}{r} .4935 \times 10^5 \\ - .0039 \times 10^5 \\ \hline .4896 \times 10^5 \end{array}$$

In case of floating-point multiplication and division an alignment of mantissas is not required. The product of two numbers is obtained by multiplying the two mantissas and adding the exponents. The division of the number X by Y is performed by dividing the mantissa of X by the mantissa of Y, and subtracting the exponent of Y from the exponent of X. After multiplication or division the results are normalized.

Example 4. Multiply $.486 \times 10^3$ by $.38 \times 10^4$

$$\begin{aligned} & (.486 \times 10^3) \times (.38 \times 10^4) \\ &= 1.8468 \times 10^7 \\ &= .18468 \times 10^8 \end{aligned}$$

Example 5. Divide $.93800 \times 10^5$ by $.3500 \times 10^2$

$$\begin{aligned} & (.93800 \times 10^5) \div (.3500 \times 10^2) \\ &= 2.6800 \times 10^3 \\ &= 0.26800 \times 10^4 \end{aligned}$$

2.19 OVERFLOW AND UNDERFLOW

2.19.1 Integer Representation

If two numbers of n digits each are added and the sum contains $n + 1$ digits, an overflow will occur. This will hold good whether the numbers are signed or unsigned; binary or decimal. As a computer register of n bits can not accommodate the result of $n + 1$ digits, the computer will indicate overflow. If a positive number is added to a negative number, overflow does not occur because the sum is smaller than the larger of the two original numbers. An overflow occurs when numbers to be added are both positive or both negative and the results exceeds the storing capacity of the register.

Let us examine the case of addition in 2's complement. The following examples will show the addition of two positive or two negative numbers. An 8-bit register stores numbers. 7 bits are used to represent numbers. The leftmost bit is used to represent the sign of the number.

Example 1. Add 69 and 78.

$$\begin{array}{r} + 69 \\ + 78 \\ \hline + 147 \end{array} \quad \begin{array}{l} 0\ 1000101 \text{ (Binary equivalent of 69)} \\ 0\ 1001110 \text{ (Binary equivalent of 78)} \\ \hline 1\ 0010011 \end{array}$$

Carries : 01

In this case there is a carry from the MSB of the numbers but no carry (or carry = 0) out of the sign bit. The sign bit is 1, which indicates negative sum. So result is not correct.

Example 2. Add -69 and -78.

$$\begin{array}{r} - 69 \\ - 78 \\ \hline - 147 \end{array} \quad \begin{array}{l} 1\ 0111011 \text{ (2's complement of -69)} \\ 1\ 0110010 \text{ (2's complement of -78)} \\ \hline 0\ 1101101 \end{array}$$

Carries : 10

In this case there is no carry (carry = 0) from MSBs of the numbers but there is a carry from the sign bit. The sign bit of the result is 0, so result is not correct.

From the above example it is clear that there is an overflow if two numbers are positive but the sum has a negative sign bit or two numbers are negative but the result has a positive sign bit. In other words the overflow produces an erroneous sign reversal. To detect the overflow condition the carry out of the MSBs of number (or carry into the sign bit) and the carry out of the sign bit are examined. If these two carries are not equal that is one of them is 0 and the other is 1, an overflow condition exists. If both the carries are 0 or both are 1 there is no overflow. Examples of addition in 2's complement in Sec. 2.8.8 indicate the conditions where there is no overflow. In those examples both carries are zero or both are 1.

2.19.2 Floating-point Representation

When the result is too large or too small to be presented by the computer, an overflow or underflow condition exists. When two floating-point numbers of the same sign are added, a carry may be generated out of high-order bit position. This is known as *mantissa overflow*. In case of addition or subtraction floating-point numbers are aligned. The mantissa is shifted right for the alignment of a floating-point number. Sometimes, the low order bits are lost in the process of alignment. This is referred to as *mantissa underflow*. To perform the multiplication of two floating-point numbers, the exponents are added. In certain cases the sum of the exponents may be too large and it may exceed the storing capacity of the exponent field. This is called *exponent overflow*. In case of division the exponent of the divisor is subtracted from the exponent of the dividend. The result of subtraction may be too small to be represented. This is called *exponent underflow*.

Overflow or underflow resulting from a mantissa operation can be corrected by shifting the mantissa of the result and adjusting the exponent. But the exponent overflow or underflow can not be corrected and hence, an error indication has to be displayed on the computer screen.

PROBLEMS

1. Why do digital computers use binary numbers for their operation?
2. Users of computers work with usual alphabets, special symbols, decimal numbers etc., whereas a computer can accept only binary digits. Then how do computers understand the information fed by the user?
3. Convert the following binary numbers to equivalent decimal numbers:
(i) 11010 (ii) 1011001 (iii) 1001011 (iv) 10011010
[Ans. (i) 26 (ii) 89 (iii) 75 (iv) 154]
4. Convert the following decimal numbers to equivalent binary numbers:
(i) 19 (ii) 43 (iii) 154 (iv) 97.
[Ans. (i) 10011 (ii) 101011 (iii) 10011010 (iv) 1100001]
5. Convert the following binary fractions to equivalent decimal fractions:
(i) 0.1011 (ii) 0.1010 (iii) 0.011 (iv) 0.0101
[Ans. (i) 0.6875 (ii) 0.625 (iii) 0.375 (iv) 0.3125]
6. Convert the following real binary numbers to equivalent decimal numbers:
(i) 1001.101 (ii) 1101.11 (iii) 11001.0101 (iv) 101001.1101
[Ans. (i) 9.625 (ii) 13.75 (iii) 25.3125 (iv) 41.8125]
7. Convert the following decimal numbers to equivalent binary numbers:
(i) 17.71875 (ii) 50.7 (iii) 74.635 (iv) 100.5
[Ans. (i) 10001.10111 (ii) 110010.1011001 (iii) 1001010.1010001 (iv) 1100100.10]
8. Perform the following additions and check the result by converting the binary numbers to decimal:
(i) 1100 + 1001 (ii) 101011 + 110010 (iii) 1011001 + 11010
[Ans. (i) 10101 (ii) 1011101 (iii) 1110011]
9. Perform the following additions and check the result by converting the binary numbers to decimal:

- (i) 101.011 + 11.110 (ii) 1011.1010 + 1000.011 (iii) 11001.1011 + 10011.0110
[Ans. (i) 1001.001 (ii) 10100.00 (iii) 101101.0001]
10. Perform the following subtractions and check the result by converting the binary numbers to decimal:
(i) 1101 – 1001 (ii) 11001 – 10110 (iii) 10011010 – 1100001
[Ans. (i) 100 (ii) 11 (iii) 111001]
11. Perform the following subtractions and check the result by converting the binary numbers to decimal:
(i) 101.101 – 11.011 (ii) 1100.01 – 1001.11 (iii) 1011.1 – 100.11
[Ans. (i) 10.01 (ii) 10.1 (iii) 110.11]
12. Find 9's and 10's complements of the following decimal numbers:
(i) 75 (ii) 438 (iii) 589 (iv) 183
[Ans. (i) 24, 25 (ii) 561, 562 (iii) 410, 411 (iv) 816, 817]
13. Find 9's and 10's complements of the following decimal numbers:
(i) 53.48 (ii) 0.863 (iii) 69.45 (iv) 54.99 (v) 48.9
[Ans. (i) 46.51, 46.52 (ii) 9.136, 9.137 (iii) 30.54, 30.55 (iv) 45.00, 45.01 (v) 51.0, 51.1]
14. Find 1's and 2's complements of the following binary numbers:
(i) 1010 (ii) 11001 (iii) 000 (iv) 111
[Ans. (i) 0101, 0110 (ii) 00110, 00111 (iii) 111, 000 (iv) 000, 001]
15. Find 1's and 2's complements of the following binary numbers:
(i) 0.101 (ii) 10.011 (iii) 100.11 (iv) 101.1
[Ans. (i) 1.010, 1.011 (ii) 01.100, 01.101 (iii) 011.00, 011.01 (iv) 010.0, 010.1]
16. Perform the following subtractions of binary numbers using 2's complements:
(i) 1101 – 1001 (ii) 101 – 111 (iii) 101 – 0.11 (iv) 0.11 – 0.101
[Ans. (i) 0100 (ii) -10 (iii) 100.01 (iv) 0.001]
17. Why are subtractions using 2's complements used in modern computers? What are the difficulties with BCD or simple binary subtractions?
18. How is a negative number represented? Represent – 5 in binary number:
[Ans. Signed-magnitude representation 10101
Signed 1's complement representation 11010
Signed 2's complement representation 11011]
19. Convert the following hexadecimal numbers to equivalent decimal numbers:
(i) 5A (ii) 1A5E (iii) AB7 (iv) 3A9 (v) 6B
[Ans. (i) 90 (ii) 6750 (iii) 2743 (iv) 937 (v) 107]
20. Convert the following decimal numbers to equivalent hexadecimal numbers:
(i) 70 (ii) 950 (iii) 10761 (iv) 6747 (v) 428
[Ans. (i) 46 (ii) 3B6 (iii) 2A09 (iv) 1A5B (v) 1AC]
21. Convert the following decimal numbers to equivalent octal numbers:
(i) 63 (ii) 460 (iii) 2749 (iv) 201 (v) 3965
[Ans. (i) 77 (ii) 714 (iii) 5725 (iv) 311 (v) 7575]

22. Convert the following octal numbers to equivalent decimal numbers:
(i) 72 (ii) 642 (iii) 5264 (iv) 704
[Ans. (i) 58 (ii) 418 (iii) 2740 (iv) 452]
23. Convert the following hexadecimal numbers to equivalent binary numbers:
(i) 6BC (ii) 5D, (iii) F2E (iv) 4C3F (v) 6E
[Ans. (i) 1101011100 (ii) 1011101 (iii) 111100101110 (iv) 10011000011111, (v) 1101110]
24. Convert the following binary numbers to equivalent hexadecimal numbers:
(i) 10110110111 (ii) 10010101 (iii) 1111110 (iv) 1101001100 (v) 101101
[Ans. (i) 5B7 (ii) 95 (iii) 7E (iv) 34C (v) 2D]
25. Convert the following octal numbers to equivalent binary numbers:
(i) 435 (ii) 637 (iii) 256 (iv) 134
[Ans. (i) 100011101 (ii) 110011111 (iii) 10101110 (iv) 1011100]
26. Convert the following binary numbers to equivalent octal numbers:
(i) 100011101 (ii) 1100101 (iii) 10110111 (iv) 11010110
[Ans. (i) 435 (ii) 145 (iii) 267 (iv) 326]
27. Convert the following hexadecimal numbers to equivalent octal numbers:
(i) 4B (ii) D9 (iii) 3F2 (iv) 5A3
[Ans. (i) 113 (ii) 331 (iii) 1762 (iv) 2643]
28. Convert the following octal numbers to equivalent hexadecimal numbers:
(i) 134 (ii) 67 (iii) 1527 (iv) 4753
[Ans. (i) 5C (ii) 37 (iii) 357 (iv) 9EB]
29. What are BCD codes? Discuss their merits and demerits. What are their areas of applications?
30. What is the full form of ASCII? What is the maximum number of characters, which can be represented by these codes? Discuss its areas of applications.
31. What are ASCII-8 codes? What is the maximum number of characters, which can be represented by these codes? Discuss their areas of applications.
32. What are EBCDIC codes? How many characters can be represented by these codes? Discuss their areas of applications.
33. What are the advantage of Gray codes? Discuss their areas of applications.
34. Discuss overflow and underflow phenomena which occur in a digital computer.

3

CHAPTER

DIGITAL DEVICES

In this chapter our aim is not to discuss all digital devices. Some important devices like logic gates, flip-flops, registers, multiplexers etc. will be described. Some special purpose digital devices will be discussed in other chapters.

3.1 LOGIC GATES

A digital computer uses binary number system for its operation. In the binary system there are only two digits 0 and 1. The computer receives, stores, understands and manipulates information composed of only 0s and 1s. The manipulation of binary information is done by logic circuits known as logic gates. The important logic operations which are frequently performed in the design of digital systems are: AND, OR, NOT, NAND, NOR and EXCLUSIVE-OR. An electronic circuit which performs a logic operation is called a *logic gate*. For example, the electronic circuit which performs AND operation is called AND gate, which performs OR operation is known as OR gate and so on. The logic gates are the basic building blocks of a digital system. A logic gate manipulates binary data in a logical way, i.e., the manipulation has a logical meaning. The knowledge of logic gates is essential to understand the important digital circuits used in computers. The example of such circuits are: the digital circuits used for addition, subtraction, multiplication etc. in a digital computer.

3.1.1 AND Gate

An AND gate has two or more inputs but it has only one output. An input signal applied to a gate has only two stable states, either 1 (HIGH) or 0 (LOW). There is no intermediate state. In case of a 2-input AND gate the output is 1 (high) only if both inputs are 1 (high), otherwise the output is 0 (low). The logical operation of an AND gate is called the AND operation (or AND function). The logical AND operation can easily be explained with the help of two switches connected in series, Fig. 3.1(b). The current will flow in the circuit only when both switches, A and B, are closed. A switch has two stable states, ON and OFF. The ON state is taken as logic 1, the OFF as logic 0. When both switches are ON, i.e., A and B both have logic 1, the output is 1. When anyone of the switches is OFF, there is no output (no current in the circuit). The output is taken as 0.

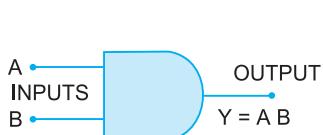
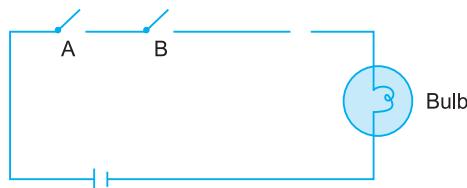
**Fig. 3.1(a) AND Gate.****Fig. 3.1(b) A circuit containing two switches in series.**

Fig. 3.1(a) shows standard symbol for AND gate. The figure shows only two inputs, A and B. If there are more than two inputs, they will be shown on the input side. Table 3.1 shows the truth table for a two-input AND gate. The output of an AND gate is 1 only when all the inputs are 1. If anyone of the inputs is 0, the output will be 0. Therefore, the output of an AND gate is same as the product of inputs i.e., the output $Y = A \cdot B$. The symbol \wedge is also used to denote AND operation. $A \wedge B$ denotes AND operation of A and B. ICs are available for AND gate, for example, 7408, 7415 and 7421 are 2-input, 3-input and 4-input AND gates respectively. Refer to section 4.1.1 for more details.

Table 3.1. Truth Table for AND Gate

INPUTS		OUTPUT
<i>A</i>	<i>B</i>	$Y = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

The realization of various types of logic gates is described in Appendix I.

Truth Table

The output of an AND gate is $Y = A \cdot B$. In this case A and B (inputs) are independent variables. Y (output) is a dependent variable. The table which shows the values of the dependent variable for all possible values of independent variables is called *truth table*.

3.1.2 OR Gate

An OR gate is also known as INCLUSIVE-OR gate. It has two or more inputs, but only one output. The output of an OR gate will be high if at least anyone of the inputs is high. The output will be 0 only when all inputs are 0. This logical operation will be called OR operation (or OR function). Fig. 3.2(a) shows a two-input OR gate. Its truth table has been shown in Table 3.2. Its output is given by $Y = A + B$. The logical OR operation can easily be explained taking an example of two switches connected in parallel as shown in Fig. 3.2(b). The current will flow in the circuit when either switch is in ON position (i.e., logic 1). The current will not flow at all when both switches are in OFF position (i.e., logic 0). The \vee symbol denotes OR operation. $A \vee B$ denotes OR operation of A and B. Example of OR gate is: 7432, a 2-input OR gate. The rules for OR operation are not same as those of arithmetic operation. Here, $1 + 1 = 1$ is not valid for arithmetic addition. For more details refer to section 4.1.2.

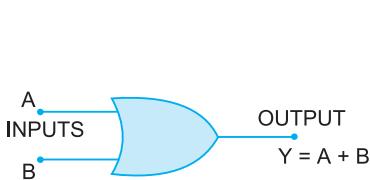


Fig. 3.2(a) OR Gate.

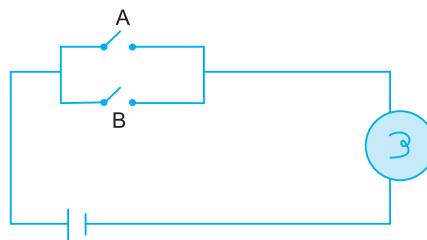


Fig. 3.2(b) A circuit containing two switches in parallel.

Table 3.2 Truth Table for OR Gate

INPUTS		OUTPUT
A	B	$Y = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

3.1.3 NOT Gate

A NOT gate has only one input, and only one output signal. It is also called INVERTER. Its output is the complement of the input signal. The output is 1 (HIGH) if the input is 0 (Low). The output is 0 when input is 1. Its output is given by $Y = \bar{A}$. NOT gates are used to invert the logic levels i.e., from LOW to HIGH or HIGH to LOW where required. Fig. 3.3 shows standard symbol for a NOT gate. Table 3.3 shows its truth table. Examples of inverter ICs are 7404, 7405 etc.

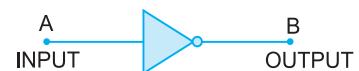


Fig. 3.3 NOT Gate.

Table 3.3. Truth Table for NOT Gate

INPUT Signal	OUTPUT Signal
A	$Y = \bar{A}$
0	1
1	0

3.1.4 NAND Gate

A NAND gate has two or more inputs but only one output. Fig. 3.4(a) shows a two-input NAND gate. The NAND function is the complement of the AND function. An AND gate can be combined with an INVERTER to form a NAND gate. The abbreviation NAND is the short form of NOT-AND. Its output is given by $Y = \overline{A \cdot B}$. Table 3.4 shows the truth table for a 2-input NAND gate.

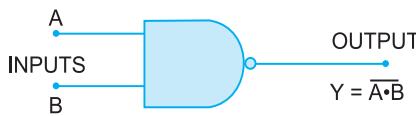


Fig. 3.4(a) NAND Gate

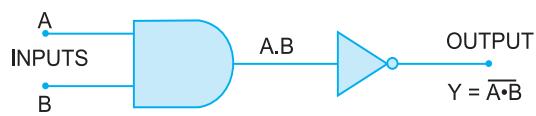


Fig. 3.4(b) Realisation of NAND Gate using AND gate and inverter

Table 3.4 Truth Table for NAND Gate

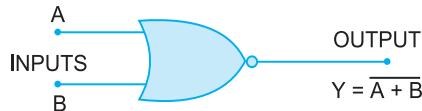
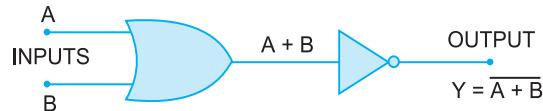
INPUTS		OUTPUT
A	B	$Y = \overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

The NAND operation is sometimes called *universal operation* because other logic operations can be realized using only NAND gates. A universal operation is one that can be used to implement the three basic logic operations AND, OR and NOT. It is easier to fabricate NAND and NOR gates employing modern IC technology compared to AND and OR gates. NAND and NOR gates also consume less power. Therefore, combinational logic networks are realized using only NAND gates. NAND gates are used as basic building blocks in fabricating a digital circuit.

NAND gates are available as a 2-, 3-, 4-, 8-, 12-, 13-input gate. For example, 7400, 7410, 7420, 7430, 74134, 74133 are 2-input, 3-input, 4-input, 8-input, 12-input, and 13-input NAND gates respectively.

3.1.5 NOR Gate

A NOR gate has two or more inputs but only one output. Fig. 3.5(a) shows a two input NOR gate. The NOR function is the complement of OR function. An OR gate can be combined with an INVERTER to form a NOR gate as shown in Fig. 3.5(b). The abbreviation NOR is the short form of NOT-OR. Its output is given by $Y = \overline{A + B}$. Table 3.5 shows the truth table for a 2-input NOR gate.

**Fig. 3.5(a) NOR Gate.****Fig. 3.5(b) Realisation of NOR using OR gate and inverter.**

The NOR operation is also a universal operation. Other logic gates such as AND, OR and NOT can be realized using only NOR gates. Combinational logic network can be realized using only NOR gates. NOR gates are used as basic building blocks in fabricating a digital network. NOR gates are available in IC forms. For example, 7402, 7427, 7425 and 74260 are 2-input, 3-input, 4-input, 5-input NOR gates respectively.

Table 3.5. Truth Table for NOR Gate

INPUTS		OUTPUT
A	B	$Y = \overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

3.1.6 Exclusive-OR (XOR) Gate

The output of XOR gate is high only when its inputs are different. Its output is low when both inputs A and B are same i.e., either both are high or both are low. Its output is given by

$$\begin{aligned} Y &= A \oplus B \\ &= \bar{A} \oplus \bar{B} \\ &= \bar{A}\bar{B} + A\bar{B} \end{aligned}$$

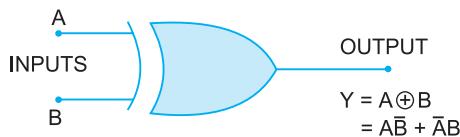


Fig. 3.6 EXCLUSIVE-OR gate.

The XOR operation is also called Modulo-2-sum operation. Fig. 3.6 shows the standard symbol for a 2-input XOR gate. Table 3.6 shows its truth table. Examples of 2-input XOR gates are 7486 and 74136.

Table 3.6. Truth Table for XOR Gate

INPUTS		OUTPUT
A	B	$Y = A \oplus B$ $= A\bar{B} + \bar{A}B$
0	0	0
0	1	1
1	0	1
1	1	0

An XOR operation can be distinguished from OR function as follows:

A OR B means A or B or both whereas A XOR B means A or B but not both.

3.1.7 Exclusive-NOR (XNOR) Gate

The XNOR operation is the complement of XOR operation. The output of XNOR is high only when the logic values of both inputs A and B are same i.e., either both are 1 or both are 0. Its output is 0 when its inputs are different. Table 3.7 shows its truth table. The output is given by

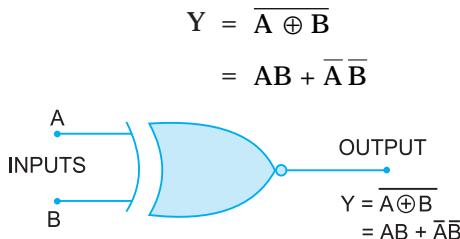


Fig. 3.7 EXCLUSIVE-NOR gate

Fig. 3.7 shows a two-input XNOR gate. The 74135 is an XOR/XNOR gate. It has three inputs A, B and C. If C input is low it operates as an XOR gate. If C input is high it operates as XNOR gate.

Table 3.7 Truth Table for XNOR Gate

INPUTS		OUTPUT
A	B	$Y = \overline{A \oplus B}$ $AB + \overline{A} \overline{B}$
0	0	1
0	1	0
1	0	0
1	1	1

3.1.8 Realization of Other Logic Functions Using NAND/NOR

The NAND operation is said to be universal operation because other logic operations can be realized using only NAND gates. A logic operation is said to be universal that can be used

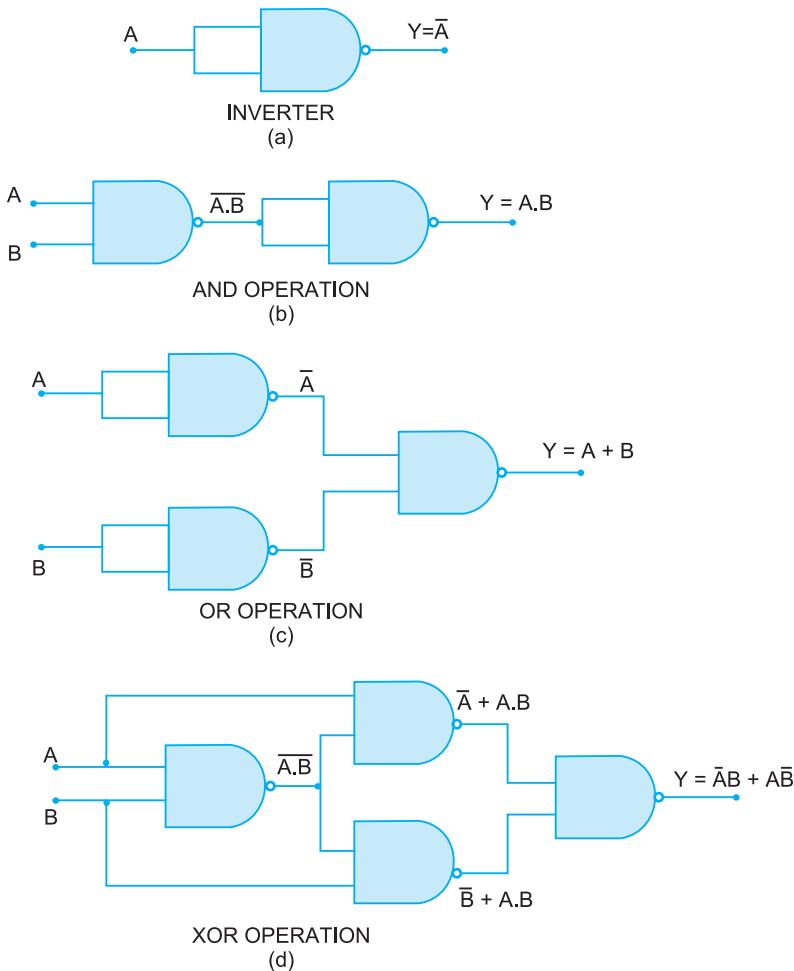


Fig. 3.8 Realisation of other logic operations using NAND gates.

to implement other basic logic operations AND, OR and NOT. NOR gate is also a universal gate. Other basic logic operations can be realized using only NOR gates. It is easier to fabricate NAND or NOR gates using IC technology than AND or OR gates. Another advantage of NAND and NOR gates is that they consume less power. Due to these reasons they are used as basic building blocks in fabricating digital networks. Fig. 3.8 and Fig. 3.9 show the realization of other logic gates using only NAND gates and using only NOR gates respectively.

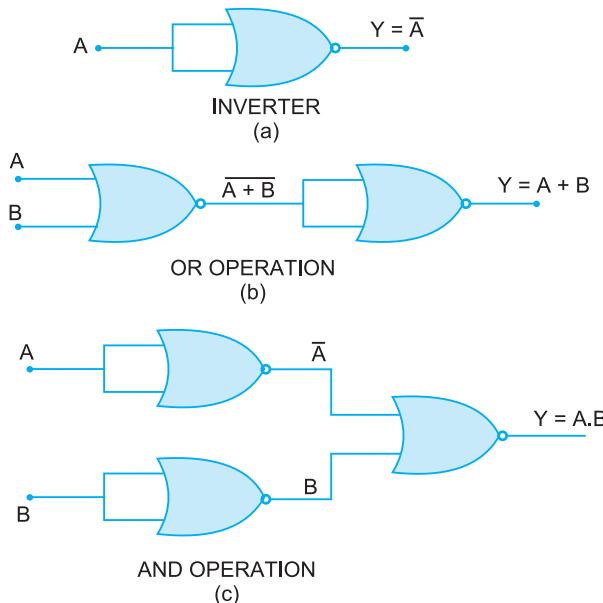


Fig. 3.9 Realisation of other logic operations using NOR gates

3.1.9 Tri-State Logic Gate

A tri-state logic gate has three distinct output-state conditions out of which two states are: high and low, similar to a conventional gate. The third state is a high impedance state. In the third state the gate behaves as if its output is disabled. The gate is controlled by a separate control input signal C. If C is high the gate behaves like a normal logic gate providing output 1 or 0. When C is low the output is disabled irrespective of the values of the inputs. In certain cases the control signal is designed to enable the gate, when it is low.

When a number of outputs of conventional gates are connected to a common bus, there will be loading problem. To solve this tri-state gates are employed. When tri-state gates are connected to a common bus only one such gate whose control is enabled gets access to the bus. Fig. 3.10(a) and (b) show a tri-state NAND gate with active-high and active-low control respectively. Examples of tri-state logic gate ICs are:

74134 – 12-input positive tri-state NAND gate

74125 – Tri-state quad bus buffer

74366 – Tri-state hex inverter.

3.1.10 Fan-Out and Fan-In

The fan-out of a logic gate is defined as the number of other logic gates that can be connected to the output of the gate. In other words the fan-out of a logic gate is the maximum

number of other gates which can be driven by the gate. Similarly, the fan-in is defined as the maximum number of gates which can be connected to the input lines of the gate. Obviously, the fan-in will be equal to the number of input lines of the gate.

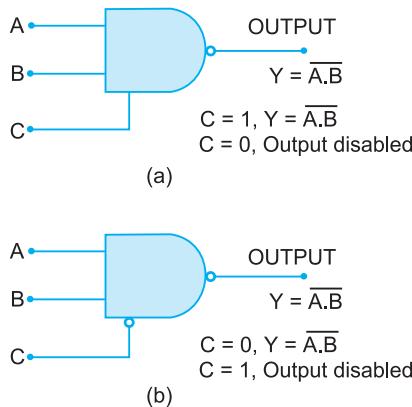


Fig. 3.10 Tri-state NAND gate (a) with active-high control (b) with active-low control.

3.2 FLIP-FLOPS

A device is said to be bistable which has two stable states. A flip-flop is a bistable device. It has two stable states: its output remains either high or low. The high stable state, i.e., 1 is called SET. The other stable state (low, i.e., 0) is called RESET. Its property is to remain in one state indefinitely until it is changed by an input signal to switch over to the other state. So it can store binary bit either 1 or 0. Thus it has storing capability, i.e., memory. It is a basic memory element or storage cell.

3.2.1 S-R Flip-Flop

An S-R flip-flop can be realized connecting two NOR gates as shown in Fig. 3.11 (a). R and S are two input signals to switch the flip-flop from one stable state to the other. Fig. 3.11 (b) is another way of drawing the same circuit. Q is the output of the flip-flop. \bar{Q} is the complement of the output. To understand the operating principle of the flip-flop circuit one should recall that a high on any input of a NOR gate will force its output low irrespective of the logic value of the other input. Thus a high on S (keeping R = 0) input will make $\bar{Q} = 0$. \bar{Q} is one of the inputs of the upper NOR gate. As both inputs (R and \bar{Q}) of the upper NOR gate are now low, its output Q will be high. Thus the flip-flop stores binary bit 1

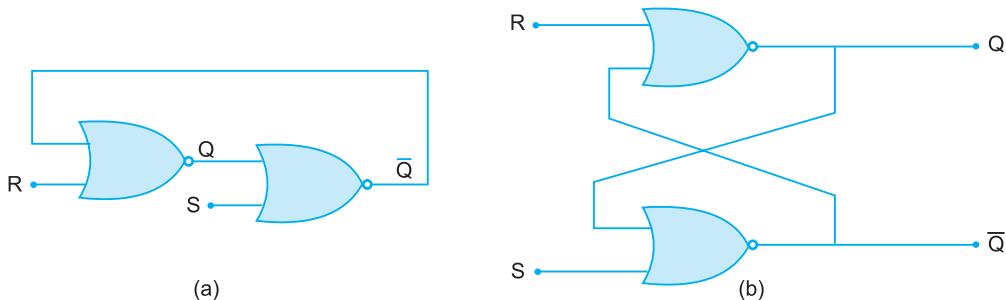


Fig. 3.11 S-R Flip-flop using NOR gates.

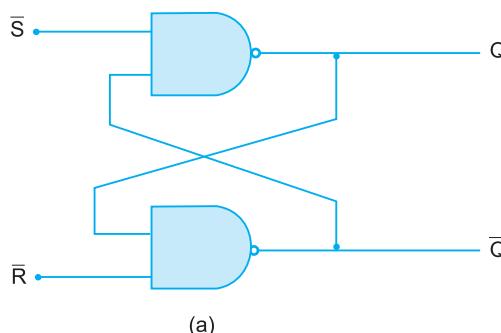
when S is made high. Even if the set input S is removed, the output Q will remain 1 because Q is one of the inputs of the lower NOR gate (i.e., Q acts as feedback). Similarly, when reset input R is made high keeping S = 0, Q will become low. So the flip-flop now stores binary bit 0. Q is also one of the inputs of the lower NOR gate. As both inputs of the lower gate (Q and S) are now low, \bar{Q} is high. When R = 0, and S = 0, the flip-flop will remain in the previous state, i.e., Q remains unchanged.

Table 3.8 shows the truth table for S-R flip-flop. An S-R flip-flop is also called a bistable multivibrator.

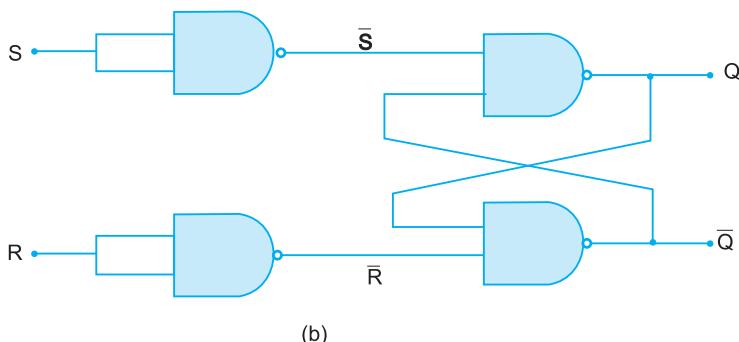
Table 3.8. Truth Table for S-R Flip-Flop

R	S	Q	Action
0	0	Last value	No change
0	1	1	Set
1	0	0	Reset
1	1	—	Invalid condition

When both S and R are made high simultaneously, it will make the outputs of both NOR gates low which is against the basic definition of a flip-flop. However, if such condition occurs incidentally, Q will be unpredictable being 1 or 0 depending upon the circuit condition.



(a)



(b)

**Fig. 3.12 (a) $\bar{S} - \bar{R}$ Flip-Flop employing NOR gates.
(b) S - R Flip-Flop employing NAND gates.**

A slightly different flip-flop can be realized using NAND gates as shown in Fig. 3.12(a). It may be called $\bar{S} - \bar{R}$ flip-flop. A 0 on any input to a NAND gate will make its output high. Therefore, when we make $\bar{S} = 0$ and $\bar{R} = 1$, Q will be 1 and $\bar{Q} = 0$. Similarly, when \bar{R} is made 0 keeping $\bar{S} = 1$, Q will become 0 and $\bar{Q} = 1$. If both \bar{S} and \bar{R} are made high, the flip-flop will remain in its previous output state. Both $\bar{S} = 0$ and $\bar{R} = 0$ will make the output of the flip-flop unpredictable. To convert an $\bar{R} - \bar{S}$ flip-flop into an R-S flip-flop the technique shown in Fig. 3.12(b) is adopted.

Flip-flop in the simplest forms as shown in Fig. 3.12(a) and (b) is called a *latch*. There are unclocked latches. There are some clocked latches also. They will be discussed later on in this chapter.

3.2.2 Clocked R-S Flip-Flop

A clock signal is added to the flip-flop to control the instant at which the flip-flop changes the state of its output. Figs. 3.13(a) and 3.13(b) show clocked R-S flip-flop. In Fig. 3.13(a) two additional AND gates have been provided so as to apply the clock signal. When clock is low the outputs of AND gates will be forced to be low and hence the state of the flip-flop is not changed. It remains in the previous state. The changes in R and S has no effect on the flip-flop output Q. Thus the flip-flop is disabled when clock is low.

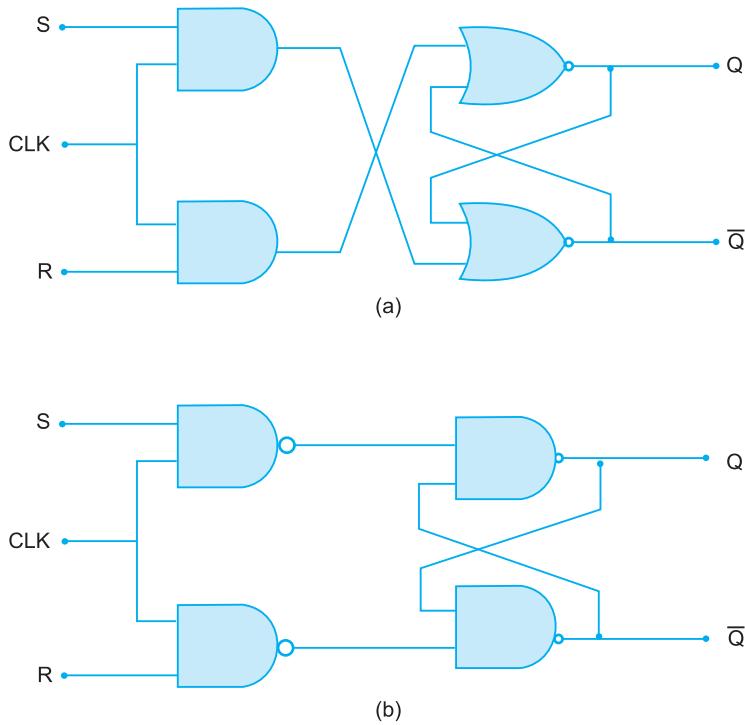


Fig. 3.13 Clocked R-S Flip-Flop

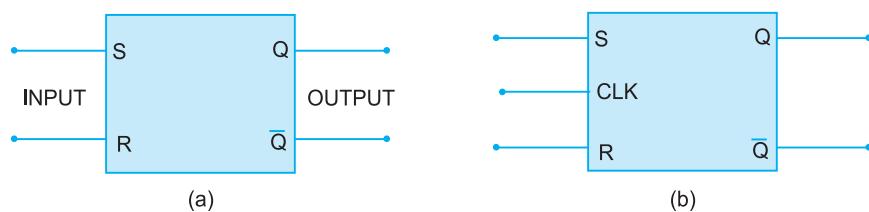


Fig. 3.14 Symbol for S-R Flip-Flop (a) Unclocked, (b) Clocked

When clock is high the outputs of AND gates will respond to the changes in inputs R and S. The flip-flop will now change its output according to the set or reset input. Thus the flip-flop is enabled when clock is high. The S-R flip-flop shown in Fig. 3.13(b) will also be enabled when clock is high and disabled when clock is low.

Figs. 3.14(a) and (b) show standard symbol for unclocked and clocked S-R flip-flop respectively. Table 3.9 shows the truth table for clocked S-R flip-flop. Q_{n+1} denotes the output state after the n^{th} clock-pulse and Q_n before the n^{th} clock-pulse.

Table 3.9 Truth Table for Clocked S-R Flip-Flop

INPUTS		OUTPUT
S_n	R_n	Q_{n+1}
0	0	Q_n (No change, remains in last state)
1	0	1
0	1	0
1	1	Unpredictable (Not to be used)

S-R flip-flops can be designed to be edge-triggered. The level-triggered S-R flip-flop is often called an S-R *latch*. The unclocked simple flip-flops and level triggered S-R and D flip-flops come under the category of *latches*. In an unclocked latch any change in input information is transferred immediately to the output. In a clocked latch the change in input data is transferred immediately when clock is high. The output will change in response to the changes in input as long as the clock is high. On the other hand in a flip-flop the output changes at a particular moment. So edge-triggered and master-slave devices fall under the category of flip-flops.

3.2.3 J-K Flip-Flop

In an S-R flip-flop the state of the output is unpredictable when $S = R = 1$. A J-K flip-flop allows inputs $J = K = 1$. In this situation the state of the output is changed; the complement of the previous state is available at the output terminal. Fig. 3.15(a) shows the schematic diagram of a J-K flip-flop. It has been realized by making $S = J$, \bar{Q} and $R = K \cdot Q$. Fig. 3.15(b) shows its standard symbol. Its truth table is given in Table 3.10.

Table 3.10 Truth Table for J-K Flip-Flop

	INPUTS		OUTPUT
CLK	J	K	Q_{n+1}
X	0	0	Q_n (No change, remains in last state)
↑	1	0	1
↑	0	1	0
↑	1	1	\bar{Q}_n (toggle)

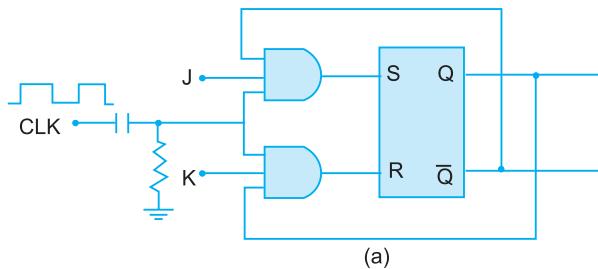
↑ indicates positive-edge of the clock.

When J and K both are 0, the outputs of the AND gates in Fig. 3.15(a) will be low, i.e., S and R are both low. When S and R are both low, there will be no change in the output state. It will remain in previous state.

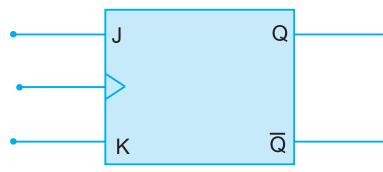
When J = 0 and K = 1, the output of the upper AND gate, i.e., S becomes low. Therefore, it is not possible to set the flip-flop. As K = 1, the output of the lower AND gate, i.e., R will be high if the other input of this gate Q is high. So the flip-flop can be reset. But, if the flip-flop is already in the reset state, i.e., Q = 0, R will be low. So, the flip-flop will remain in the reset state.

When J = 1 and K = 0, the output of lower AND gate, i.e., R becomes low. Therefore, it is not possible to reset the flip-flop. As J = 1, the output of the upper AND gate, i.e., S will be high if the other input of this gate \bar{Q} is high. So the flip-flop can be set. But, if the flip-flop is already in the set state, i.e., $\bar{Q} = 0$, S will be low. Therefore, the flip-flop will remain in the set state.

When J = 1 and K = 1, the flip-flop can be set or reset depending upon the previous state of the output. If Q = 1 and $\bar{Q} = 0$, the upper AND gate is disabled and the lower AND gate enabled. This makes S = 0 and R = 1. So the flip-flop is reset, i.e., Q becomes 0 and $\bar{Q} = 1$. If Q = 0 and $\bar{Q} = 1$, S will be 1 and R = 0. Hence, the flip-flop will be set, i.e., Q will become high and $\bar{Q} = 0$. Thus it is seen that the flip-flop is forced to take the complement of the previous state.



(a)



(b)

Fig. 3.15 J-K Flip-Flop (a) Schematic diagram (b) Standard symbol.

In Fig. 3.15 (a) assume that R-C circuit has been removed. With $J = 1$, $K = 1$ and high CLK, the output of the flip-flop will change to the complement of the previous state. The new outputs are again fed back to the input AND gates. After the propagation delay the output will again change if the high CLK persists. Thus the output will go on changing so long as clock is high. So during positive half cycle of the clock the output of the J-K flip-flop oscillates. This situation is called race-around condition or racing. To solve this problem the clock pulses are converted to narrow spikes using R-C circuit having a short time constant. Because of AND gates, the flip-flop becomes positive-edge-triggered. The output changes after the positive edge of the spike strikes. By the time new values of Q and \bar{Q} return to the input AND gates, the positive spikes die out. So only one change in the output occurs during each clock cycle. Propagation delay time prevents J-K flip-flop from racing.

The J-K flip-flop is the most versatile and commonly used flip-flop. It is an ideal flip-flop to be used for counting purposes.

3.2.4 Preset and Clear

When power is switched on, the flip-flops go to random states. So presetting or clearing is required before applying inputs. For such purpose preset PR and clear CLR terminals are provided as shown in Figs. 3.16(a) and (b). Small bubbles shown at PR and CLR terminals indicate that they are effective when they are low. A low PR sets Q to 1. Similarly, a low CLR resets Q to 0. PR and CLR can not be made low at the same time.

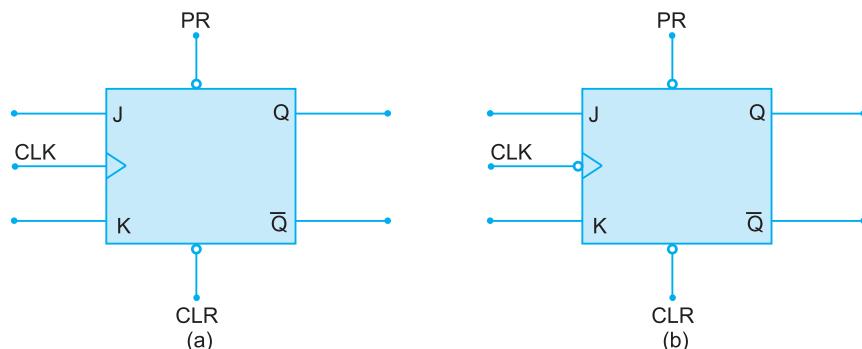


Fig. 3.16 Triggering of Flip-flops (a) Positive-edge-triggered (b) Negative-edge-triggered.

3.2.5 Triggering of Flip-Flops

The flip-flops can be triggered to set/reset either at one of the edges or levels of the clock pulse. There are three types of triggering as described below.

Positive-Edge-Triggered Flip-Flops

Positive-edge-triggered flip-flops set or reset at the positive (rising or leading) edge of the clock pulse depending upon the state of the input signals. The set or reset state of the output remains steady for one clock period and the clock again samples the input signal on the next positive edge of the clock. Fig. 3.16(a) shows the schematic representation of a positive-edge-triggered flip-flop. The arrowhead symbol is called dynamic signal indicator. Examples of positive edge-triggered flip-flops are:

- 7470 AND gated J-K Flip-Flops with preset and clear.
- 7474 Dual D-type Flip-Flops with preset and clear.
- 74109 Dual J-K Flip-Flops with preset and clear.

Negative-Edge-Triggered Flip-Flops

Negative-edge-triggered flip-flops set or reset at the negative (falling or trailing) edge of the clock pulse. Fig. 3.16(b) shows the symbolic representation of negative-edge-triggered flip-flops. A small circle known as bubble is placed before the arrowhead to show negative-edge-triggering. Examples of negative edge-triggered flip-flops are:

- 74H102 AND gated J-K Flip-Flops with preset and clear.
- 74LS73A Dual J-K Flip-Flops with clear.
- 74LS113A Dual J-K Flip-Flops with preset.
- 74276 Quad J – \bar{K} Flip-Flops, separate clocks, common direct clear and preset.

Level-Triggered (or Pulse-Triggered) Flip-Flops

The level triggering may be either positive triggering or negative triggering. In positive triggering the flip-flop sets or resets according to the state of the input lines when clock is high. If the input changes during high clock period, the output will also change, accordingly. When clock is low the output does not change, it remains in the previous state which was at the end of the positive clock pulse.

Similarly, in negative triggering the flip-flop sets or resets according to the state of the input lines when clock is low. When clock is high the output does not change. It remains in the previous state which was at the end of the negative clock pulse.

Level-triggered D type flip-flops or D latches are transparent latches. In a transparent latch the output follows the state of the input lines when clock is high. If input is changing the output goes on changing when the clock is high. This type of flip-flops are not suitable for computer applications. But they are suitable for some applications in other digital systems. Level-triggered master-slave flip-flops are not transparent. Examples of level-triggered flip-flops and latches are:

- 7475, 7477 D type dual transparent latches
- 74279 Quad $\bar{S} - \bar{R}$ latch
- 7473, 7476 Dual J-K Flip-Flops
- 74H71, 7472 Master-slave J-K Flip-Flops
- 74L71 AND-gated S-R Master-Slave Flip-Flop
- 74259 8-bit addressable latches with clear

3.2.6 Master-slave J-K Flip-Flops

Fig. 3.17 shows the schematic diagram of a master-slave J-K flip-flop. It consists of two clocked S-R flip-flops. One is called master and the other slave. When the clock is high, the master is active and the slave inactive. The master sets or resets according to the state of the input signals. As the slave is inactive during this period its output remains steady at the previous state. When the clock goes low the master is inactive and the slave is active. The slave sets or resets according to its inputs. The final output Q of a master-slave flip-flop is the output of the slave. The output of the slave is available at the end of the clock pulse.

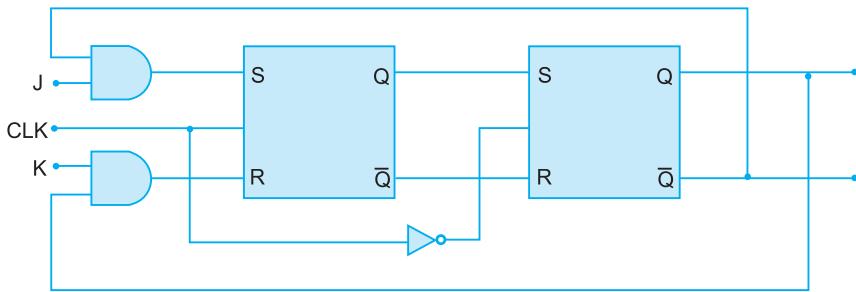


Fig. 3.17 Master-slave J-K Flip-flop.

When inputs, i.e., J and K both are high the master-slave flip-flop toggles once. The master toggles once and then slave copies the action of the master. There is no race-around condition because the feedback is from the output of the slave which is steady during positive half cycle of the clock. J-K master-slave flip-flops are very popular in industry. They are ideal flip-flops to be used in counting devices. Master-slave flip-flops are also realized using D flip-flops. Examples of J-K master-slave flip-flop ICs are: 74H71, 7472, 74110 and 74111. An example of S-R master-slave flip-flop is 74L71. All these master-slave flip-flops are level-triggered. A problem associated with level triggered master-slave J-K flip-flops is that the J and K inputs must not change while clock is high otherwise the output will be unpredictable. If a master-slave unit contains edge-triggered flip-flops there will be no such difficulty.

3.2.7 D Flip-Flop

An S-R flip-flop has two inputs, S and R. To store 1 a high S and low R are required. To store 0 a high R and low S are needed. Thus two signals are to be generated to drive an S-R flip-flop. This is a serious disadvantage in many applications. To overcome this difficulty D flip-flop has been developed which has only one input line. A D flip-flop can be realized using an S-R flip-flop as shown in Fig. 3.18(a). There is no racing problem with D flip-flop. A J-K flip-flop can also be used to realize a D flip-flop using the technique shown in Fig. 3.18(a).

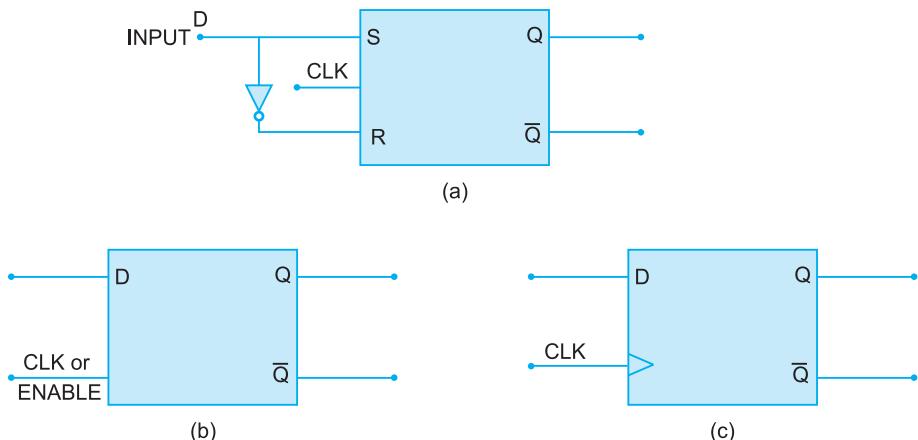


Fig. 3.18 D Flip-flop (a) schematic diagram. (b) Level triggered, (c) Positive-edge-triggered.

If the device shown in Fig. 3.18(a) is level-triggered, it is often called a D latch (or bistable latch). Fig. 3.18(b) shows its symbolic representation. In level triggering when clock

(or enable signal) is high the output Q follows the state of input line D. When D goes high, Q goes high. When D goes low, Q goes low. When clock is low, the output does not change. It remains in the previous state. Table 3.11 shows the truth table for a D latch. A D latch is a transparent latch. Such latches are not suitable for computer applications but they are suitable for other applications in digital circuits. Examples of transparent latch ICs are: 7475, 7477, 74100, 74116 and 74LS375.

Table 3.11 Truth Table for D Latch

CLK	D INPUT	Q_{n+1}
0	X	Q_n (previous state)
1	0	0
1	1	1

If the device shown in Fig. 3.18(a) is positive-edge-triggered it is called a D flip-flop. The flip-flop sets or resets according to the state of the input D at the positive edge of the clock. The set or reset state of the output remains for one complete clock period. Fig. 3.18(c) shows its schematic diagram. This type of D flip-flops are suitable for computer circuits. Examples of positive-edge-triggered D flip-flop ICs are: 7474, 74174, 74175, and 74273.

Table 3.12 shows truth table for a D flip-flop. When clock is low, Q remains in the previous state. On the arrival of positive edge of the clock Q is set to 1 if D is 1; Q is set to 0 if D is 0. The positive edge of the clock is marked by an upward arrow. On the negative edge of the clock (marked by downward arrow), Q remains in the last state.

Table 3.12. Truth Table for a D Flip-Flop

CLK	D INPUT	Q_{n+1}
0	X	Q_n (last state)
↑	1	1
↑	0	0
↓	X	Q_n (last state)

3.2.8 T Flip-Flop

A T flip-flop acts as a toggle switch. Toggle means to switch over to the opposite state. It can be realized using a J-K flip-flop by making T = J = K = 1, as shown in Fig. 3.19. Its truth table is shown in Table 3.13.

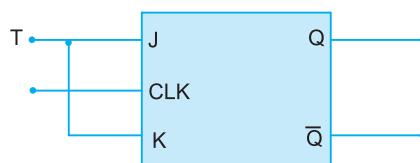


Fig. 3.19 T Flip-flop

Table 3.13. Truth Table for T Flip-Flop

INPUT	OUTPUT
T	Q_{n+1}
0	Q_n
1	\bar{Q}_n

3.3 LATCHES

A flip-flop in its simplest form is called a latch. A latch stores a binary bit 1 or 0. The unclocked simple flip-flops, and level-triggered S-R and D flip-flops fall under the category of latches. An n -bit latch consists of n -number single-bit latches. It stores a binary word of n bits. The n -bits of the binary word are transferred to the latch simultaneously in parallel. In a latch there is no facility to read its contents. The latches are temporary storage devices. They are ideally suited for storings information between processing units and I/O units or indicator units. The examples of latch ICs are:

- 74100 – 8-bit bistable latches
- 74259 – 8-bit addressable latches
- 7475 – 4-bit bistable latches
- 7477 – 4-bit bistable latches

3.4 REGISTERS

Registers consist of flip-flops rather than latches. The flip-flops are connected in parallel. All n -bits of an n -bit register are transferred in parallel. A register stores a binary word. The binary word remains stored until a new word is applied. Unlike a latch, the contents of a register can be read at any time without changing its value. Usually, registers consist of D type edge-triggered flip-flops. IC 74273 is an 8-bit register. It consists of 8 edge-triggered D flip-flops, with common clock and clear lines. The registers of microprocessors are within the microprocessor's IC package.

3.5 SHIFT REGISTERS

In a shift register the flip-flops are connected in series. The output of each flip-flop is connected to the input of the adjacent flip-flop. The contents of a shift register can be shifted within the register without changing the order of the bits. Data are shifted one position left or right at a time when one clock pulse is applied. Shift registers are used for the temporary storage of data. They are used where bit shifting is required for manipulating, computing or processing data. They are ideally suited to processing serial data, converting serial data to parallel data, and converting parallel data to serial data. They can be built using D, J-K or S-R flip-flops. Depending upon the modes of loading and reading out the data there are four types of shift-registers:

- (i) Serial-in, serial-out
- (ii) Serial-in, parallel-out
- (iii) parallel-in, serial-out
- (iv) parallel-in, parallel-out.

Examples of shift register ICs are:

- (i) 7491 8-bit serial-in, serial-out shift registers.
- (ii) 74164 8-bit serial-in, parallel-out shift registers.
- (iii) 74165 and
- 74166 8-bit serial/parallel-in, serial-out shift registers.

(iv) 74198 8-bit bidirectional universal shift registers.

(v) 74199 8-bit serial/parallel-in, parallel-out shift registers.

Universal shift registers are provided with mode select pins for I/O data transfer.

3.6 BUFFERS/DRIVERS

The function of a buffer/driver is to increase the output current/voltage ratings. When the output current of a digital device is insufficient to drive another device which is to be connected to the output terminal of the device, a buffer is employed to amplify the current. A buffer is a current amplifier. It is represented by a triangle as shown in Fig. 3.20(a). For an inverting buffer a bubble is placed at the output point of triangle (Fig. 3.20(b)). Sometimes, an increased voltage is required to drive relays, lamps etc. Suitable drivers are employed for such applications.



Fig. 3.20 (a) Buffer, (b) Inverter buffer.

As the output current of standard TTL gates may be 10 times the input current, a certain amount of buffering is done by a gate. But the device is said to be a buffer only when the manufacturers optimize the design for high current output. For example, 7426 is a quad 2-input NAND buffer. Its NAND gates are optimized for high current output.

Examples of buffer/driver ICs are:

7407, 7417 Hex buffers/drivers with open collector outputs

7406, 7416 Hex inverter buffer/drivers with open collector outputs

7428 Quad 2-input NOR buffers

7440 Dual 4-input NAND buffers.

3.7 ENCODERS, DECODERS AND CODE CONVERTERS

Digital computers use binary numbers for their operation. But users work with alphabets and decimal numbers. So electronic devices are required to convert alphanumeric characters into binary codes. The function of an encoder is to produce binary codes. The decoding is the reverse process of encoding. Examples of encoders and decoders are:

7441, 74141 BCD to decimal decoders/drivers, open collector outputs

7442 BCD to decimal decoders/drivers totem-pole outputs

7417 Decimal to BCD encoders

74148, 74348 Octal to BCD encoders.

The hexadecimal to binary decoder is realized employing two 74148 encoders and a data selector.

There are a number of binary codes to represent numbers and alphanumeric characters such as BCD, hexadecimal, octal, excess-3 code, Gray codes etc. Sometimes it is required to convert information from one binary code to another. Devices used for such conversion are called *code converters*. Examples of code converters are:

- 74184 BCD to binary converter
- 74185 Binary to BCD converter

3.8 COUNTERS

Digital counters consist of a number of flip-flops. Their function is to count electrical pulses. To count certain event, electrical pulses proportional to the event are produced for the counting purpose. Counters can also be used to count time interval or frequency. For such purposes clock pulses are applied to the counter. As the clock pulses occur at certain known intervals, the number of pulses are proportional to time. Therefore, time interval can easily be measured. The frequency is inversely proportional to time. So frequency can also be measured.

There are basically two types of counters: synchronous and asynchronous. In a synchronous counter all flip-flops are clocked simultaneously. On the other hand in asynchronous counters the flip-flops are not clocked simultaneously. Each flip-flop is triggered by the previous flip-flop.

3.8.1 Asynchronous or Ripple Counters

A binary asynchronous counter can be realized by the use of edge-triggered J-K flip-flops as shown in Fig. 3.21(a). The flip-flops are connected in series. The Q output of one flip-flop is connected to the clock terminal of the next flip-flop. In other words the output of one flip-flop drives the next flip-flop. So it is also called *serial counter*. It is inherently slow in operation due to series connection. The triggers move through the flip-flops like a ripple in water. So it is called ripple counter.

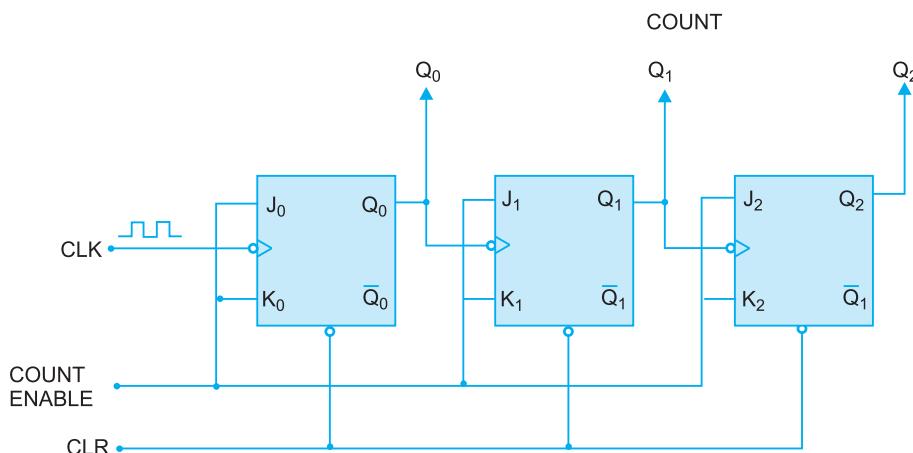
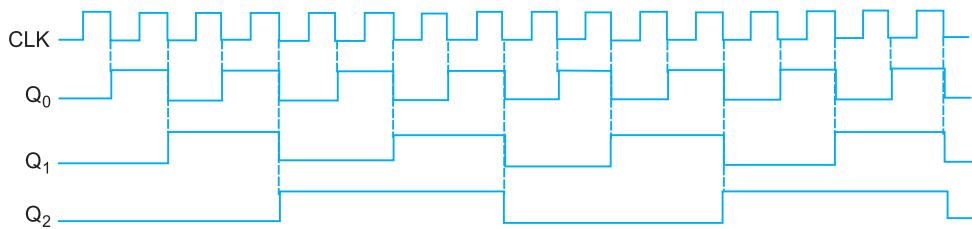


Fig. 3.21(a) Binary asynchronous (Ripple) counter.

**Fig. 3.21(b) Timing diagram**

Flip-flops are connected to operate as T flip-flops. When J and K inputs of all flip-flops are made high (*i.e.*, when count enable is high); each flip-flop will change its state (toggle) on the receipt of the negative edge of its input pulse. Suppose that the output of a T flip-flop is 0. When clock is applied its output will change from 0 to 1 on the receipt of one clock pulse. When clock goes low, the output will remain high. On the receipt of the second clock pulse, its output will change from 1 to 0. Thus a flip-flop gives an output of one pulse on the receipt of two pulses at its input. Fig. 3.21(a) shows a binary ripple counter consisting of 3 flip-flops. If at the input terminal of the first flip-flop 8 clock pulses are applied, first flip-flop will give 4 pulses at its output terminal. Now Q₀ acts as the clock input for the 2nd flip-flop. At the input terminal of the 2nd flip-flop 4 pulses are now applied. The 2nd flip-flop will give 2 pulses at its output terminal. Q₁ acts as the clock input for the 3rd flip-flop. At the input terminal of the 3rd flip-flop 2 pulses are applied. The third flip-flop will give one pulse at its output terminal.

Table 3.14 shows how the counter contents advance on the receipt of clock pulses. It is clear from the table that counter content advances one count on the receipt of each clock pulse.

Table 3.14 Truth Table of a Binary Ripple Counter

Clock pulses	Q ₂	Q ₁	Q ₀
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

On the receipt of 7 clock pulses all the three flip-flops show high outputs. On the receipt of 8th clock pulse the 3rd flip-flop gives an output of one pulse and all flip-flops show 0 states. A three flip-flop counter counts from 0 to 7. It has 8 states. It is called modulus-8 (or mod-8) counter. Similarly, a four flip-flop counter counts from 0 to 15. So it is a mod-16 counter. A six flip-flop counter is a mod-64 counter. Thus the number of output states of a counter is called *modulus* of a counter.

In a ripple counter a decoding gate is used, which detects a given state of the counter and produces an output when that state is detected.

Examples of asynchronous counters are:

- 7490 – BCD counters (divide by 2 and divide by 5).
- 7492 – Divide-by 12 counters (divide by 2 and divide by 6).
- 7493 – 4-bit binary counters (divide by 2 and divide by 8).

The ripple counters are very simple to build but they are not suitable for high frequency applications. They are used as frequency dividers. Their operation at lower frequency is satisfactory. Each flip-flop has a delay time. In a ripple counter the delay time of all the flip-flops are added resulting in greater settling time. Another problem with ripple counters is that glitches (unwanted pulses) occur at the output of the gates. Both of these problems are eliminated in synchronous counters discussed below.

3.8.2 Synchronous (or Parallel) Counters

In synchronous counters all the flip-flops are directly clocked at the same time so that all the flip-flops change their output state at the same time. Fig. 3.22 shows the schematic diagram of synchronous counter. Examples of synchronous counters are:

- 74161 – 4-bit binary UP counters.
- 74162 – Decade Up counters.
- 74192 – Decade UP/DOWN counters.
- 74193 – 4-bit binary UP/DOWN counters.

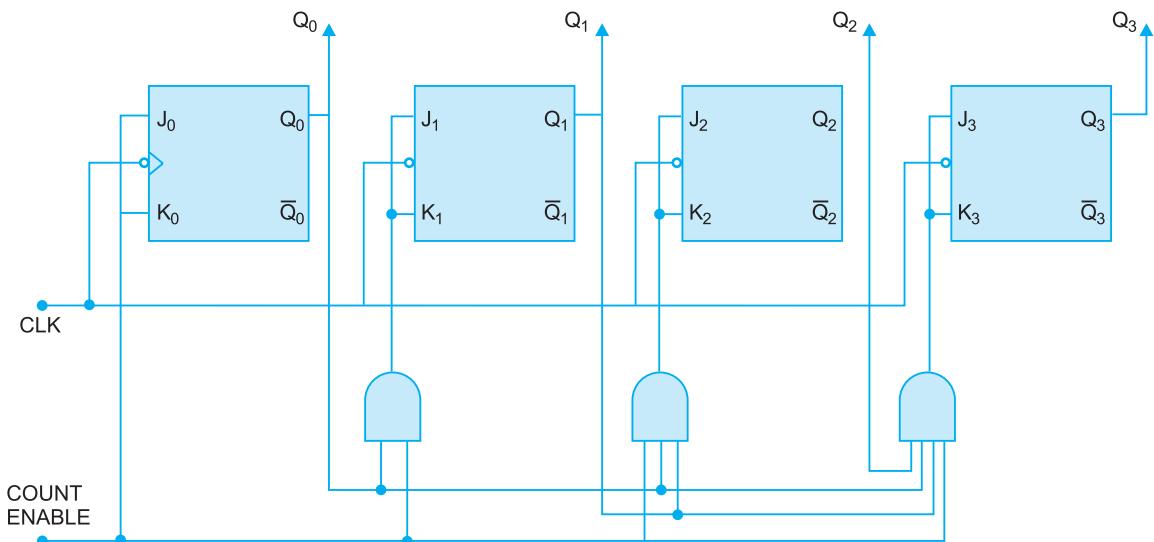


Fig. 3.22 Synchronous Counter.

3.8.3 Mod-3, Mod-6 and Mod-12 Counters

The synchronous and asynchronous counters discussed earlier have a modulus of 2^n , where n is the number of flip-flops in the counter. The count 2^n is called natural count. The counters consisting of 1, 2, 3 and 4 flip-flops have their natural count 2, 4, 8 and 16 respectively. These counters progress one count at a time in binary progression. Sometimes it is desirable to have the modulus of 3, 5 etc., other than 2^n . Such modulus are called modified

count. A counter of smaller modulus can be realized from a counter of larger modulus by skipping certain states. Fig. 3.23(a) shows a counter of modulus 3. Its truth table is shown in Table 3.15. The timing diagram is shown in Fig. 3.23(b).

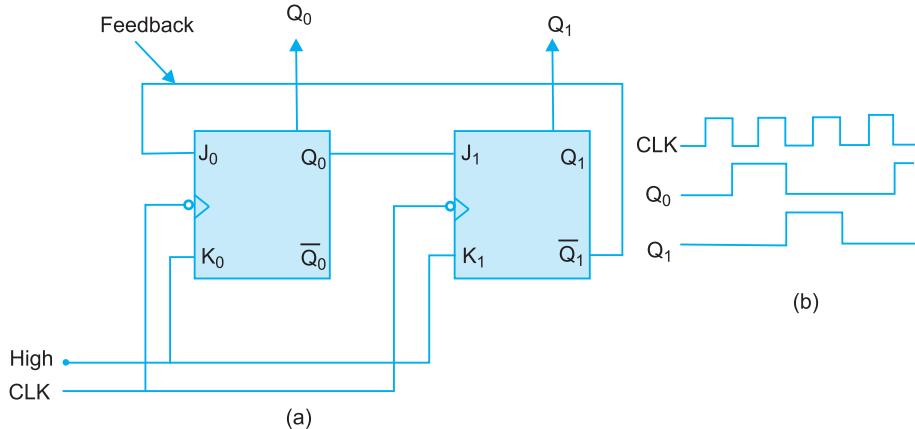


Fig. 3.23 Mod-3 counters (a) Schematic diagram (b) Timing diagram

The counter progresses one count at a time upto 2. After two clock pulses Q_0 is low and Q_1 is high. \bar{Q}_1 is feedback to J_0 . When Q_1 is high, $\bar{Q}_1 = 0$. So $J_0 = 0$ and $K_0 = 1$. On the arrival of the 3rd clock pulse the first flip-flop will remain in the reset state. So Q_0 will remain 0. At this moment $J_1 = 0$ and $K_1 = 1$. Hence the second flip-flop will also reset. Thus Q_1 will also become 0. Thus due to feedback of \bar{Q}_1 to J_0 , the counter resets to 00 after receiving 3 clock pulses. Mode-3 counter can also be realised using an alternative circuit. Refer to section 3.8.4.3.

Table 3.15. Truth Table of Mod-3 Counter

Q_1	Q_0	Count
0	0	0
0	1	1
1	0	2
0	0	3

Mod-6 Counters

To realize mod-6 counter a flip-flop is added to Mod-3 counter as shown in Fig. 3.24. Mode-6 counter can also be realised using an alternative circuit. Refer to section 3.8.4.2.

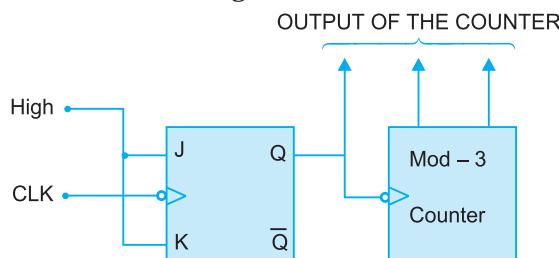


Fig. 3.24 Mod-6 counter

Mod-12 Counters

Similarly, using 2 flip-flops and one mod-3 counter ($2 \times 3 \times 2$) we can realize a mod-12 counter.

The 7492A is an asynchronous divide by 12 counter (divide by 2 and divide by 6). It consists of 4 Master-slave J-K flip-flops. 74LS92A is also a divide by 12 (2×6) counter.

Mode-12 counter can also be realised using an alternative circuit. Refer to section 3.8.4.6.

3.8.4 Mod-5 Counters

Fig. 3.25 (a) shows a mod-5 counter. \bar{Q}_2 is feedback to J_0 . Q_0 and Q_1 are applied to J_2 through an AND gate. Table 3.16 shows the truth table of a mod-5 counter.

The counter progresses one count at a time upto 4. At this moment $Q_2 = 1$, $\bar{Q}_2 = 0$. As \bar{Q}_2 is feedback to J_0 , $J_0 = 0$. K_0 is always 1. So the first flip-flop will remain in reset state. $Q_0 = 0$ is also applied to J_2 through the AND gate resulting in $J_2 = 0$. Therefore, on receipt of 5th clock the third flip-flop will now reset changing Q_2 from 1 to 0. Q_1 was already 0. Thus the counter's output is 000. Thus the counter resets at 000 after receiving 5 clock pulses.

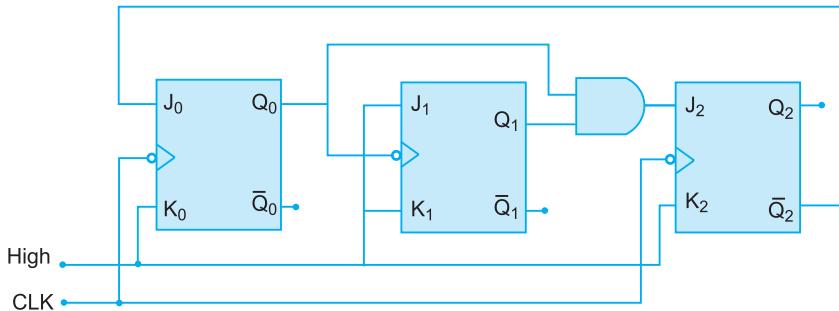


Fig. 3.25 (a) Mod-5 counter

Table 3.16 Truth Table of a Mod-5 Counter

Q_2	Q_1	Q_0	Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
0	0	0	5

3.8.4.1 Alternative Circuit For Mode-5 Counter

Fig. 3.25(b) shows the schematic diagram of a mode-5 counter. Q_0 and Q_2 are applied to a NAND gate. When Q_0 and Q_2 both are high, then only it gives 0 (low) output. The output of the NAND gate is applied to an AND gate. The other input to the AND gate CLR (clear) signal. When either CLR is low or the output of the NAND gate is low, the output of the AND gate is low which clears all the flip-flops of the counter. This resets the counter. The counter progresses one count at a time up to 4 as shown in Table 3.16. In this situation $Q_0 = 0$, $Q_1 = 0$ and $Q_2 = 1$. When fifth clock pulse is applied, Q_0 as well as Q_2 , both are 1, and hence the output of the NAND gate becomes 0 (low) which further makes the output of AND gate

low. This resets all the flip-flops of the counter thus the counter is reset. When the counter is reset, it shows 000. This counter counts from 0 to 4 (five number). Hence, it is called mode-5 counter. The principle of its circuit is easy to understand.

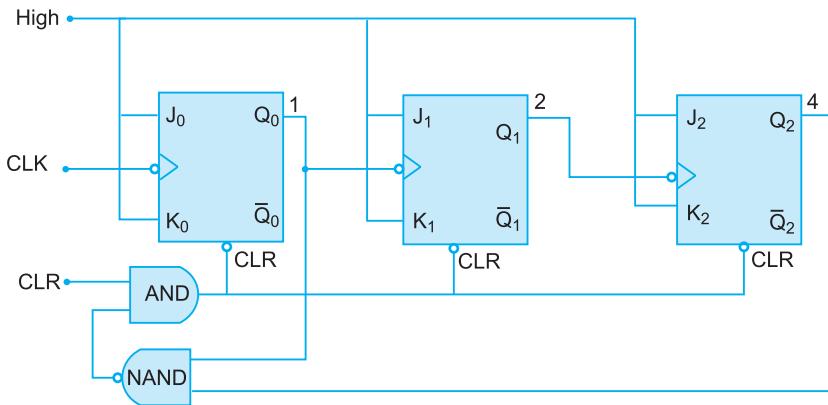


Fig. 3.25 (b) Mode-5 Counter.

3.8.4.2 Mode-6 Counter

Mode-6 counter can also be realized modifying the circuit of Fig. 3.25(b) slightly. When Q_1 and Q_2 are applied to the NAND gate, it becomes Mode-6 counter. Though this circuit requires more electronic components, its principle is easy to understand.

3.8.4.3 Mode-3 Counter

To obtain a mode-3 counter, a modified circuit based on the principle of the circuit shown in Fig. 3.25(b) can be used. For this only two flip-flops will be used. There will be a NAND gate and an AND gate. Q_0 and Q_1 will be applied to the NAND gate. The resulting circuit will be circuit for a Mode-3 counter.

3.8.4.4 Mod-10 (Decade or BCD) Counters

A decade counter can be realized using a flip-flop and a mod-5 counter ($2 \times 5 = 10$). Fig. 3.26 (a) shows a decade counter. Table 3.17 shows its truth table.

Table 3.17 Truth Table of a Decade Counter

Q_3	Q_2	Q_1	Q_0	Count
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
0	0	0	0	10

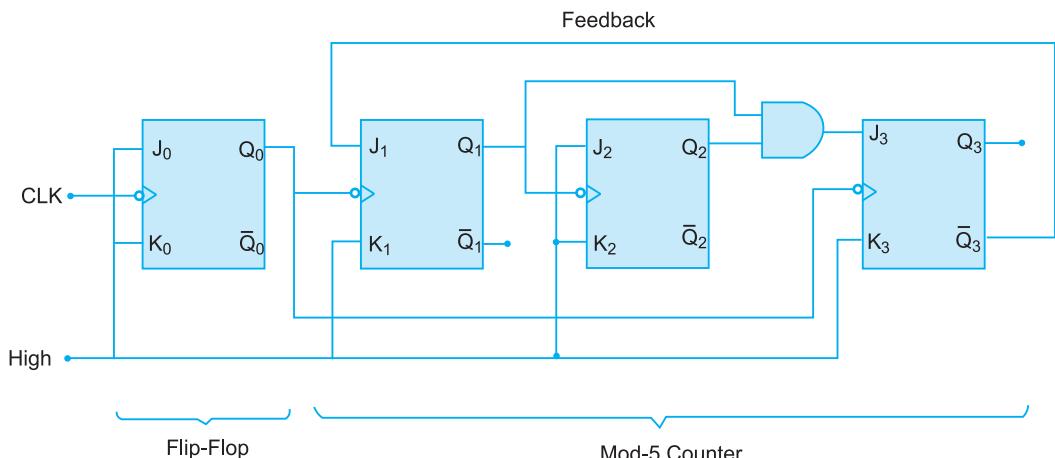


Fig. 3.26(a) Decade Counter

The ICs 7490, 74L90, 74LS90, 74290, 74LS290, 74390 and 74490 are asynchronous decade counters. They are 2×5 counters. 74190, 74LS190, 74L192 and 74LS192 are BCD synchronous counters.

3.8.4.5 An Alternative Circuit for Mode-10 (Decade or BCD) Counter

Fig. 3.26(b) shows the schematic diagram of a mode-10 (decade or BCD) counter. Q_1 and Q_3 are applied to a NAND gate. When Q_1 and Q_3 , both are high, then only the output of the NAND gate is low. The output of the NAND gate is applied to an AND gate. The other input to the AND gate is CLR (clear) signal. When either CLR or the output of the NAND gate is low, the output of the AND gate becomes low which clears all the flip-flops of the counter. Thus the counter progresses one count at a time up to 9 as shown in the Table 3.17. At this point $Q_0 = 1$, $Q_1 = 0$, $Q_2 = 0$ and $Q_3 = 1$. On the receipt to 10th clock pulse Q_1 and Q_3 , both are high and hence, the output of the NAND gate becomes low which further makes the output of the AND gate low. This resets all the flip-flops of the counter. Thus this counter is reset. When the counter is reset, it shows 0000. This counter counts from 0 to 9 (ten numbers). Therefore, it called mode-10 (decade or BCD) counter. The principle of its circuit is easy to understand.

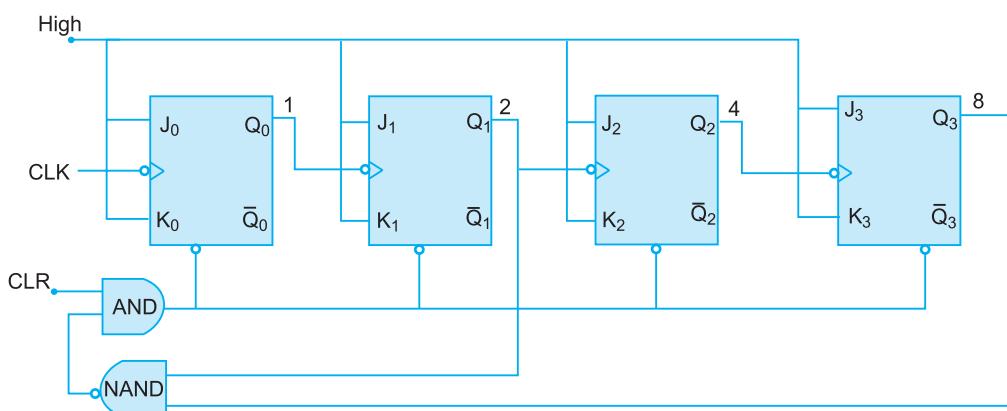


Fig. 3.26(b) Mode-10 (Decade or BCD) counter

3.8.4.6 Mode-12 counter

To obtain a mode-12 counter, the circuit of Fig. 3.26(b) can be modified. When Q_2 and Q_3 are applied to the NAND gate, it becomes a mode-12 counter.

3.8.5 Presettable Counters

A presettable counter starts counting from a number greater than zero. This number can be set before counting starts. Take an example of presettable counter consisting of 4 flip-flops. Its counting states are from 0000 to 1111. Suppose, it has been preset at 0100 before counting starts. When counting starts it will count 0101 on the receipt of the first clock pulse, 0110 on the receipt of the second clock pulse and so on. In this way it will count upto 1111 on the receipt of subsequent clock pulses. Thus we see that a presettable counter has variable modulus. It has programmable modulus. Examples of presettable counter ICs are:

74176, 74196 Presettable decade (biquinary) counters

74177, 74197 Presettable 4-bit binary counters.

3.8.6 Up Counters, Down Counters and Up/Down Counters

Up Counters

If a counter counts upward towards higher numbers, it is called up counter. The examples of up counters are:

74160, 74162 Decade Upcounters

Down Counters

The counters which count downward towards lower numbers are down counters.

Up/Down Counters

A counter which is designed to count either towards higher numbers or towards lower numbers is called an up/down counter. An U/\bar{D} terminal is provided to control up/down counting. If U/\bar{D} is high the counter becomes an up counter. If it is low the counter becomes a down counter. The examples of UP/DOWN counters are:

74168, 74190, 74192 Decade UP/DOWN synchronous counters.

74169, 74191, 74193, 4-bit binary UP/DOWN synchronous counters.

3.8.7 Controlled Counters

A controlled counter counts electrical pulses when it is asked to do so. There is a control terminal COUNT to control counting. When it is high, the counter counts pulses applied to it. When COUNT is low it does not do counting even though pulses may remain applied to it.

3.8.8 Ring Counters

Instead of counting numbers, a ring counter uses a binary word which has only a single high bit. This word is shifted within the counter. Fig. 3.27 shows a 4-bit ring counter. In a ring counter flip-flops are connected in series. The output Q of the one stage is connected to D input of the next stage. The output of the last stage is feedback to the first stage. All flip-flops are clocked simultaneously.

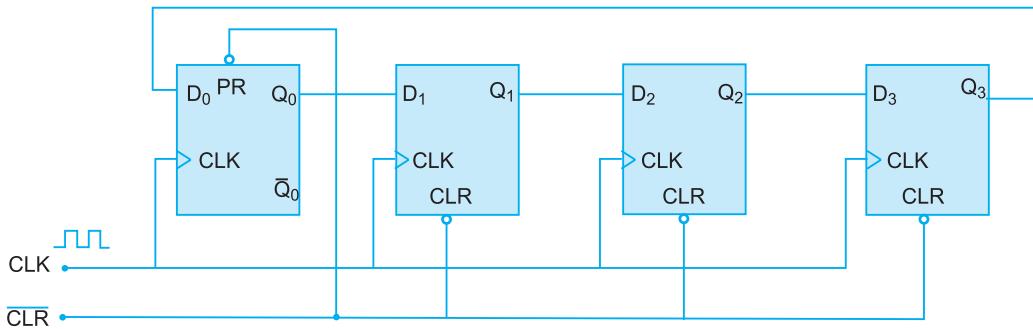


Fig. 3.27 Ring counter.

A ring counter is employed to control a number of operations in a sequence. The devices may be connected to Q_0 , Q_1 , Q_2 , ... Q_n outputs of the counter. The outputs Q_0 , Q_1 , Q_2 , ... Q_n go high in a sequence. For initial setting a binary word 0001 is set in a 4-bit ring counter. First CLR goes low. Q_0 is made high. Then CLR becomes high. The device connected to Q_0 is activated. Now clock pulses applied to CLK terminal activate devices. On the receipt of the first clock pulse Q_1 becomes high, and the device connected to Q_1 is activated. The second clock pulse makes Q_2 high, the 3rd clock pulse makes Q_3 high and so on. At a time only one output becomes high and all others remain low. As the outputs Q_0 , Q_1 , Q_2 , ... Q_n are made high in a sequence, the devices connected to these outputs are activated in a sequence. Ring counters are used in computers to control certain operations in a sequence. Their other applications are: to control stepper motors, encoding the outputs of electronic keyboards etc.

3.8.9 Shift Counter (Johnson or Moebius Counter)

It is similar to a ring counter. In a ring counter the Q output of the last stage is feedback to the D input of the 1st stage whereas in a Johnson counter the complemented output \bar{Q} of the last stage is feedback to D input of the 1st stage. If J-K flip-flops are used in the counter \bar{Q} is feedback to J input and Q is feedback to K input of the first stage. It is also called a *walking-ring*, a *twisted-ring* or *switched tail* counter.

Tables 3.18 and 3.19 show the truth table for 3-bit and Mod-10 shift counters respectively. Such counters do not count in binary or decimal progression. Their counting sequence is shown in the truth tables. In counting sequence one flip-flop gives high output, then two flip-flops give high outputs, then three and so on. At one stage all flip-flops give high outputs. After this one flip-flop gives 0 output, then two flip-flops give 0 output, then three and so on. Finally, all flip-flops give 0 output.

Table 3.18. Truth Table for 3-Bit Johnson Counter

Q_2	Q_1	Q_0	Count	State
0	0	1	1	1
0	1	1	3	2
1	1	1	7	3
1	1	0	6	4
1	0	0	4	5
0	0	0	0	6
0	0	1	1	

Table 3.19. Truth Table for a Mod-10 Johnson Counter

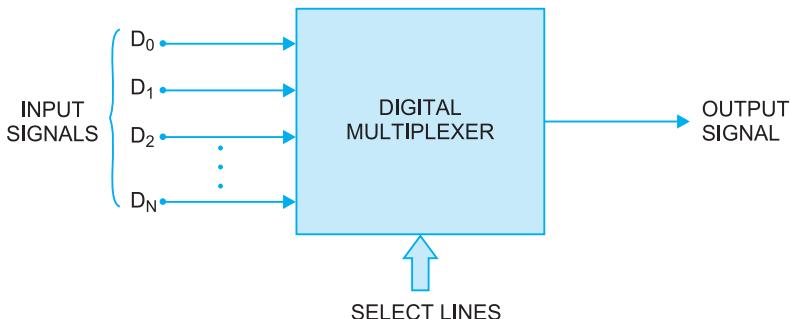
Q_4	Q_3	Q_2	Q_1	Q_0	Count	State
0	0	0	0	1	1	1
0	0	0	1	1	3	2
0	0	1	1	1	7	3
0	1	1	1	1	15	4
1	1	1	1	1	31	5
1	1	1	1	0	30	6
1	1	1	0	0	28	7
1	1	0	0	0	24	8
1	0	0	0	0	16	9
0	0	0	0	0	0	10
0	0	0	0	1	1	

Mod-10 shift counters employ five flip-flops. For 5 flip-flop counters there should be 32 states. But there are only 10 states. Others are invalid states.

3.9 DIGITAL MULTIPLEXERS/DATA SELECTORS

A digital multiplexer has N inputs and only one output. By applying control signals anyone input can be made available at the output terminal. It is also called data selector. Fig. 3.28 shows the schematic diagram of a digital multiplexer. The control signals are applied to the select lines to select the desired input. Examples of digital multiplexer/data selector ICs are:

- 74150 – 1 of 16 Data Selectors/Multiplexers
- 74151A – 1 of 8 Data Selectors/Multiplexers
- 74152 – 1 of 8 Data Selectors/Multiplexers
- 74153 – Dual 4-line to 1-line data Selectors/Multiplexers

**Fig. 3.28** Digital multiplexer.

3.10 DIGITAL DEMULTIPLEXERS/DECODERS

A digital demultiplexer has 1 input and N outputs. The meaning of demultiplexer is one into many. By applying control signals the input signal can be made available at anyone of output terminals. It performs reverse operation of a multiplexer. Fig. 3.29 shows the schematic diagram of a demultiplexer.

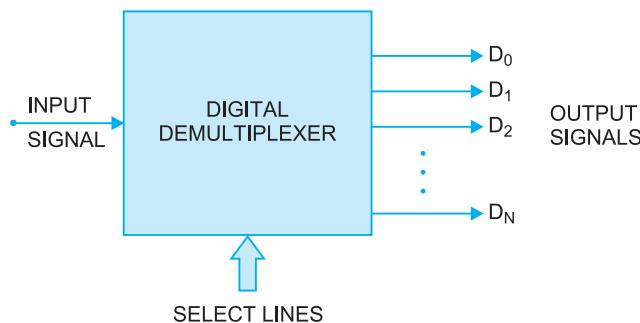


Fig. 3.29 Digital demultiplexer.

The 74154 is a 1 to 16 demultiplexer/4 to 16 decoder. Fig. 3.30 shows the connection of 74154 as a demultiplexer. The STROBE has to be low to activate the 74154. G₁ (Pin 18) and G₂ (pin 19) are two strobe inputs. To use 74154 as a demultiplexer the input signal can be applied to anyone of the strobe inputs (*i.e.*, either to G₁ or G₂) with other strobe input low. Suppose G₁ is selected as data input line, then G₂ is used as strobe and it is kept low. When strobe is low, the control signals applied to select lines select one of the output lines.

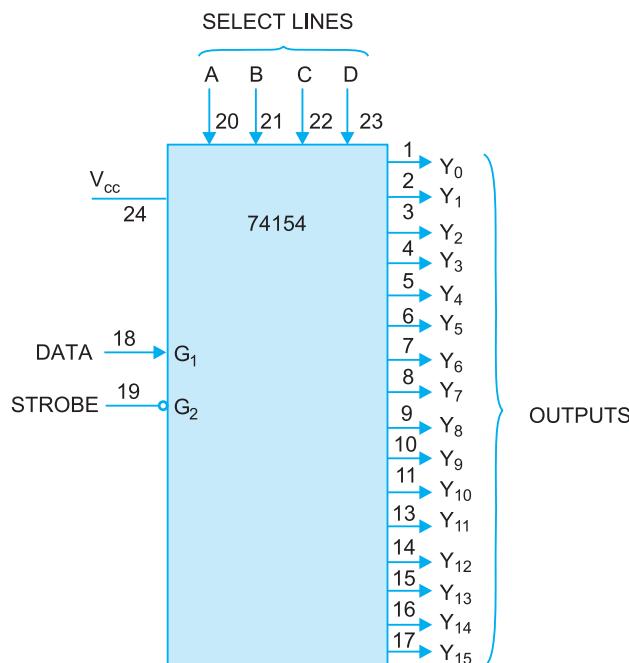


Fig. 3.30 Connections of 74154 as a Demultiplexer.

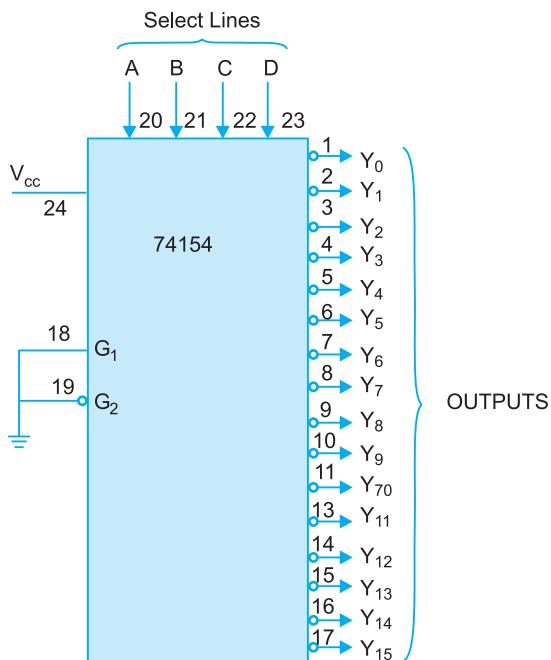


Fig. 3.31 Connections of 74154 as a Decoder.

The selected output line becomes low when input data is low. When the data input is high the selected output becomes high. Thus the input signal can be made available at the selected output terminal.

To use 74154 as a 4 to 16-lines decoder connections can be made as shown in Fig. 3.31. A decoder is similar to a demultiplexer, with one exception—there is no data input. G_1 (Pin 18) and G_2 (Pin 19) are grounded in this case. The control signals applied to select lines select one of the outputs. The selected output line goes low. Other output lines remain high. Due to this reason bubbles have been shown on the output lines. Decoder is used to connect several devices to a microprocessor. By making the selected output line of the decoder low, a device is enabled.

Examples of demultiplexer/decoder ICs are:

- 74138 – 1 to 8-lines demultiplexers or 3 to 8-lines decoders
- 74154 – 1 to 16-lines demultiplexers or 4 to 8-lines decoders
- 74155 – Dual 1 to 4-lines demultiplexers or 2 to 4 lines decoders; 1 to 8-lines demultiplexers or 3 to 8-lines decoders.

PROBLEMS

1. What are logic gates? Discuss AND operation. Draw the schematic block diagram of a 3-input AND gate. Give its truth table.
2. Explain OR operation. Give truth table for 2-input and 3-input OR gates.
3. What are XOR and XNOR gates? Give truth tables for 2 input XOR and XNOR gates.
4. What are NOR and NAND gates? Why are they called universal gates? Give truth tables for 3-input NAND and NOR gates.

5. What is tri-state logic gate? Explain in brief.
6. What do you understand by fan-in and fan-out of a logic gate?
7. What is a flip-flop? What is its function? Explain how an S-R flip-flop is realized employing (i) NOR gates, (ii) NAND gates. What is the difference between a flip-flop and latch?
8. What is the shortcoming of an S-R flip-flop? Explain how this shortcoming is removed in J-K flip-flops. Discuss how a J-K flip-flop is realized. Describe its operating principle. What is racing problem in J-K flip-flop?
9. Draw the schematic diagram of a master-slave J-K flip-flop. Discuss its working principle. What are its advantages over other types of flip-flops?
10. What is D flip-flop? What is its advantage over S-R flip-flop? What is the difference between a D latch and a D flip-flop?
11. What is triggering of flip-flops? Discuss different types of triggering with examples.
12. What are registers? What is the difference between a n -bit register and a n -bit latch?
13. What is the function of a shift register? What are its various types? Where are such registers used?
14. What are buffers/drivers? Explain in brief.
15. Discuss the function of encoders, decoders and code converters.
16. Draw the schematic diagram of an asynchronous (ripple) counter. Why is it called ripple counter? Give some examples of asynchronous counter ICs.
17. Discuss the working principle of a synchronous (parallel) counter with its block diagram. What is the advantage of synchronous counters over serial counters.
18. What is modulus of a counter? Discuss the working principle of a mod-3 counter. How are mod-6 and mod-12 counters realized using mod-3 counters?
19. What is a mod-5 counter? How is it built? How is a decade counter realized using mod-5 counter?
20. Explain the terms' 'UP counter', "DOWN counter" and 'UP/DOWN counter'.
21. What are presettable counters? What is the nature of the modulus of a presettable counter?
22. Draw the schematic block diagram of a ring counter. Describe its working principle. Mention its applications.
23. What is a shift-counter (Johnson counter)? How is it constructed?
24. What are digital multiplexers/selectors? Explain their operating principle. Give a few examples of multiplexer ICs.
25. What are demultiplexers/decoders? What is the difference between a demultiplexer and a decoder? Show connection diagram of a demultiplexer and a decoder.

**THIS PAGE
IS BLANK**

4

CHAPTER

LOGIC DESIGN

This chapter deals with Boolean algebra, minimization techniques for boolean expressions, combinational and sequential logic circuits.

4.1 BOOLEAN ALGEBRA

Boolean algebra is an algebra of logic. It is one of the most basic tools to analyze and design logic circuits. It is named after George Boole who developed it in 1854. The original purpose of this algebra was to simplify logical statements and solve logic problems. It had no practical application until 1939 when Shannon applied it to telephone switching circuits. Shannon's work gave an idea that Boolean algebra could be applied to computer electronics. Today it is the backbone of design and analysis of computer and other digital circuits.

Now consider the statement "The man is tall". This statement may be either true or false. So it is a logical statement. A symbol can be used to represent a logical statement. For example,

X = the man is tall

Further, 1 is used to represent true, and 0 to represent false. If the above mentioned statement is true, we write X = 1. If the above statement is false, we write X = 0. Here 1 and 0 represent only truth and falsehood. They do not represent numerical values. 1 and 0 have no arithmetic significance. They have only logical significance.

4.1.1 AND Operation

We have already discussed AND operation in Chapter 3. Here, the approach is different i.e., it is from logical statement point of view. Now consider the following statements:

The man is tall (= X)

The man is wise (= Y)

These statements can be written as a compound statement given below:

The man is tall and the man is wise

This compound statement can be written in the symbolic form as follows:

X AND Y

X may be true or false; and Y may be true or false. Therefore, the compound statement

X AND Y will be true only when both X and Y are true. Table 4.1 shows all possibilities of truth and falsehood of X and Y, and corresponding truth and falsehood of X AND Y. Such a table is called truth table.

Table 4.1 Truth Table for AND Operation

<i>X</i>		<i>Y</i>	<i>X AND Y</i>
True	AND	True	True
True	AND	False	False
False	AND	True	False
False	AND	False	False

A “.” is used to represent AND operation. So X AND Y will be represented as X.Y. Representing true by 1, false by 0 and AND by “.” the above table can be replaced by Table 4.2.

Other symbols which are used to represent AND operation are \wedge and \cap . In ordinary algebra “.” represents multiplication but in Boolean algebra it represents only logical AND operation. The rules for AND operation are exactly same as those of simple arithmetic multiplication. This is just a coincidence which enabled us to remember these rules without any additional effort, though the AND operation has nothing to do with the arithmetic multiplication. However, it is used in computer’s multiplication circuits because of the such coincidence.

Table 4.2 Truth Table for AND Operation in Symbolic Form

<i>X Y</i>	<i>X.Y</i>
1.1	= 1
1.0	= 0
0.1	= 0
0.0	= 0

4.1.2 OR Operation

Now consider the following two basic logical statements:

He will give me a pen (= X)

He will give me a pencil (= Y)

These two statements can be written as a compound statement given below:

He will give me a pen or a pencil or both

This compound statement can be written in the symbolic form as shown below:

X OR Y OR both

This is called inclusive-OR. Both is not written in the statement. If simply OR is written, it is understood that it is inclusive-OR. X OR Y means X OR Y OR both. Therefore, an inclusive-OR is simply written as given below:

X OR Y

There is another connective: exclusive-OR. It is written as XOR. It means X OR Y but not both.

Here, we shall consider inclusive-OR. X may be true or false, and Y may be true or false. The compound statement X OR Y will be true when anyone or both statements are true. Table 4.3 shows all possibilities of truth and falsehood of X and Y, and corresponding truth and falsehood of X OR Y.

Table 4.3 Truth Table for OR Operation

X		Y	X OR Y
True	OR	True	True
True	OR	False	True
False	OR	True	True
False	OR	False	False

A “+” is used to represent OR operation. So X OR Y can be written as $X + Y$. Representing true by 1, false by 0 and OR by “+”, the above table can be replaced by Table 4.4.

Table 4.4 Truth Table for OR Operation in Symbolic Notation

X	Y	$X + Y$
1	+ I	= 1
1	+ 0	= 1
0	+ 1	= 1
0	+ 0	= 0

Other symbols used to represent OR operation are \vee and \cup . In ordinary algebra “+” means addition, but in Boolean algebra it simply represents logical OR operation. The rules for OR operation are not exactly same as those of arithmetic addition. The 1st rule $1 + 1 = 1$ is not valid for addition. It simply indicates that if both statements are true, the compound statement is true. So we have to remember this particular rule for OR operation. The other three rules for OR operation are exactly same as those for arithmetic addition.

4.1.3 NOT Operation

Now consider the following statement:

The man is wise (assume it is $= X$)

This statement may be true or false. If this statement is true, the statement given below will be false:

The man is NOT wise ($= \text{NOT } X = \bar{X}$)

If the statement “The man is wise” is false, the statement “The man is NOT wise” will be true. Table 4.5 is the truth table for NOT operation.

Table 4.5 Truth Table for NOT Operation

X	NOT X
True	False
False	True

NOT X is represented by \bar{X} or X' . \bar{X} is called the **complement** of X.

If $X = 1$, $\bar{X} = \bar{1} = 0$

If $X = 0$, $\bar{X} = \bar{0} = 1$

It also follows that

$$\overline{\overline{X}} = X$$

$$\overline{\overline{1}} = 1$$

$$\overline{\overline{0}} = 0$$

Table 4.6 shows the truth table for NOT operation in symbolic form.

Table 4.6. Truth Table for NOT Operation in Symbolic Form

X	\bar{X}
1	0
0	1

4.1.4 Examples of Switches to Illustrate Logic Operations

Electrical switches are very good examples to give clear concept of AND, OR operations and many Boolean theorems. A switch has only two states: either closed or open. These are similar to truth and falsehood of a statement. We can assume closed = 1 and open (off) = 0.

Now consider two switches connected in series as shown in Fig. 4.1. It is a very good example to illustrate AND operation. The bulb will glow only when both the switches X and Y are closed. Table 4.7 shows its truth table.

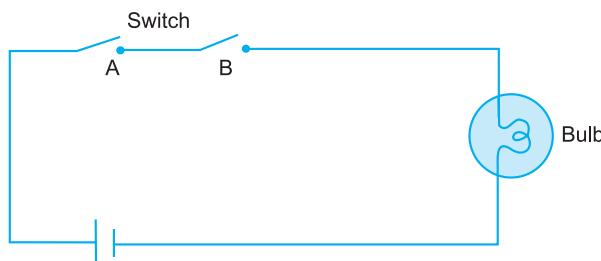


Fig. 4.1 Two switches in series.

We can assume closed = 1, open = 0, ON = 1 and OFF = 0, then Table 4.8 will replace the Table 4.7.

Similarly, two switches connected in parallel as shown in Fig. 4.2 is an example to illustrate OR operation. The bulb will glow when either or both switches are on. Table 4.9 shows its truth table.

Table 4.7. The Behaviour of Two Switches Connected in Series

Switch X	Switch Y	Bulb B
Closed	Closed	ON
Closed	Open	OFF
Open	Closed	OFF
Open	Open	OFF

Table 4.8. Truth Table for Two Switches Connected in Series

Switches		Bulb B
X	Y	$B = X \cdot Y$
1	1	1
1	0	0
0	1	0
0	0	0

Table 4.9. Behaviour of Two Switches Connected in Parallel

Switch X	Switch Y	Bulb B
Closed	Closed	ON
Closed	Open	ON
Open	Closed	ON
Open	Open	OFF

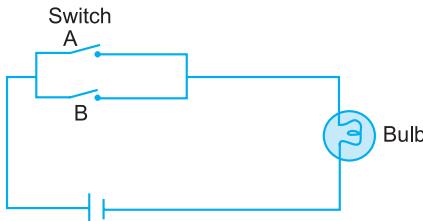
**Fig. 4.2** Two switches in parallel.

Table 4.10 shows the truth table in symbolic form for two switches in parallel.

Table 4.10. Truth Table for Two Switches in Parallel

Switches		Bulb
X	Y	$B = X + Y$
1	1	1
1	0	1
0	1	1
0	0	0

Now consider the following:

A switch is closed = A

A switch is NOT closed, i.e., open = \bar{A}

4.1.5 Boolean Variables, Operations and Functions

A computer is a binary digital system. Such a system operates on electrical signals which have only two possible states: HIGH (5 volts) and LOW (0 volt). These high and low states are represented by 1 and 0 respectively. A signal that does not change its state (or value) in time is called a **constant** signal. The value of a constant signal will always remain same: either 1 or 0. On the other hand, a signal which changes its state in time is known as a

variable signal. The value of a variable signal may be 1 at some times and 0 at some other times. For design and analysis of digital systems a constant signal and a variable signal will be treated just as a constant and as a variable respectively. Thus the variables which have only two values 1 and 0, are called **Boolean variables** (or logic variables). These variables may be denoted by A, B, C, X, Y, Z etc. There are only two Boolean constants 0 and 1. In ordinary algebra a variable can have very large (or even infinite) number of values, but in Boolean algebra they can have only one of the two possible values, 0 and 1. Thus, Boolean algebra becomes much simpler than ordinary algebra.

The only algebraic operations used in Boolean algebra are AND, OR and NOT. All other operations can be expressed in terms of these basic operations. For example, NAND operation is the combination of AND followed by NOT. Similarly, NOR is the combination of OR followed by NOT.

In ordinary algebra we have the concept of expression or function. Similarly, in Boolean algebra, we have the concept of expression or function. A Boolean function or expression consists of Boolean variables. Consider the following example:

$$X = A + B.C + C.(\bar{D} + E)$$

A, B, C, D and E are Boolean variables. The right hand side of the above equation is known as an *expression*. Each occurrence of a variable or its complement in an expression is called *literal*. In the above expression there are five variables and six literals. Generally, “.” is not written in an expression. It can be written where additional clarity is required. Boolean expression (or Boolean function) is also called logic expression (or logic function).

In the above equation the variable X is the function of A, B, C, D, and E. This can be written as

$$X = f(A, B, C, D, E)$$

4.1.6 Boolean Postulates

Fundamental conditions or self-evident propositions are called postulates. The postulates for Boolean algebra originate from the three basic logic operations—AND, OR and NOT. The properties of these basic operations given in Tables 4.2, 4.4 and 4.6 are the postulates for Boolean algebra. These are called Boolean postulates and they are summarized in Table 4.11. These postulates define the operation of the AND, OR and NOT. In other words these are the results of these basic operations.

Table 4.11. Boolean Postulates

(1) $0.0 = 0$	}	Derived from AND operation
(2) $0.1 = 0$		
(3) $1.0 = 0$		
(4) $1.1 = 1$		
(5) $0 + 0 = 0$	}	Derived from OR operation
(6) $0 + 1 = 1$		
(7) $1 + 0 = 1$		
(8) $1 + 1 = 1$		

$$\left. \begin{array}{l} (9) \quad \bar{0} = 1 \\ (10) \quad \bar{1} = 0 \end{array} \right\} \quad \text{Derived from NOT operation}$$

4.1.7 Boolean Theorems

With the help of Boolean postulates many useful theorems known as Boolean theorems have been derived. These theorems are very useful in simplifying logical expressions or transforming them into other useful equivalent expressions. Table 4.12 presents the list of Boolean theorems.

In Boolean algebra, it is possible to test the validity of the theorems by substituting all possible values of the variables because the variables have only two values, viz. 0 and 1. This technique of proving theorems is known as *proof by perfect induction*.

Table 4.12. Boolean Theorems

(1) $0.X = 0$	Properties of AND operation
(2) $X.0 = 0$	
(3) $1.X = X$	
(4) $X.1 = X$	
(5) $X + 0 = X$	Properties of OR operation
(6) $0 + X = X$	
(7) $X + 1 = 1$	
(8) $1 + X = 1$	
(9) $X.X = X$	Combining a variable with itself or its complement
(10) $X.\bar{X} = 0$	
(11) $X + X = X$	
(12) $X + \bar{X} = 1$	
(13) $\bar{\bar{X}} = X$	Double complementation
(14) $X + Y = Y + X$	
(15) $X.Y = Y.X$	
(16) $X.(Y.Z) = (X.Y).Z = X.Y.Z$	Associative laws
(17) $(X + Y) + Z = X + (Y + Z)$	
$= X + Y + Z$	
(18) $X.(Y + Z) = X.Y + X.Z$	Distributive laws
(19) $X + Y.Z = (X + Y).(X + Z)$	
(20) $X + XY = X$	Absorption
(21) $X(X + Y) = X$	
(22) $XY + X\bar{Y} = X$	
(23) $(X + Y)(X + \bar{Y}) = X$	

- (24) (a) $X + \bar{X}Y = X + Y$
 (b) $XZ + Z\bar{X}Y = ZX + ZY$
- (25) $X(\bar{X} + Y) = XY$
- (26) $(Z + X)(Z + \bar{X} + Y) = (Z + X)(Z + Y)$
- (27) $XY + \bar{X}Z + YZ = XY + \bar{X}Z$
- (28) $(X + Y)(\bar{X} + Z)(Y + Z) = (X + Y)(\bar{X} + Z)$
- (29) $XY + \bar{X}Z = (X + Z)(\bar{X} + Y)$
- (30) $(X + Y)(\bar{X} + Z) = XZ + \bar{X}Y$
- (31) $\overline{X.Y.Z\dots} = \bar{X} + \bar{Y} + \bar{Z} + \dots$
- (32) $\overline{X+Y+Z+\dots} = \bar{X}.\bar{Y}.\bar{Z}\dots$

} DeMorgan's Theorem

Proofs of some theorems by perfect induction and/or some other techniques are given below:

Theorem 1

$$\begin{array}{lll}
 & 0.X = 0 & \\
 \textit{Proof.} & \text{If } X = 0, & 0.X = 0.0 \\
 & & = 0 \text{ by Postulate 1.} \\
 & \text{If } X = 1, & 0.X = 0.1 \\
 & & = 0 \text{ by Postulate 2.} \\
 & \text{Therefore,} & 0.X = 0
 \end{array}$$

Theorem 4

$$\begin{array}{lll}
 & X.1 = X & \\
 \textit{Proof.} & \text{If } X = 0, & X.1 = 0.1 \\
 & & = 0 \text{ by Postulate 2.} \\
 & & = X \\
 & \text{If } X = 1, & X.1 = 1.1 \\
 & & = 1 \text{ by Postulate 4.} \\
 & & = X \\
 & \text{Therefore,} & X.1 = X
 \end{array}$$

Theorem 5

$$\begin{array}{lll}
 & X + 0 = X & \\
 \textit{Proof.} & \text{If } X = 0, & X + 0 = 0 + 0 \\
 & & = 0 \text{ by Postulate 5.} \\
 & & = X \\
 & \text{If } X = 1, & X + 0 = 1 + 0 \\
 & & = 1 \text{ by Postulate 7.} \\
 & & = X \\
 & \text{Therefore,} & X + 0 = X
 \end{array}$$

Theorem 8

$$\begin{aligned}
 & 1 + X = 1 \\
 \text{Proof.} \quad & \text{If } X = 0, \quad 1 + X = 1 + 0 \\
 & \qquad\qquad\qquad = 1 \text{ by Postulate 7.} \\
 & \text{If } X = 1, \quad 1 + X = 1 + 1 \\
 & \qquad\qquad\qquad = 1 \text{ by Postulate 8.} \\
 & \text{Therefore,} \quad 1 + X = 1
 \end{aligned}$$

Similarly, upto theorem 13, all of them can be proved very easily with the help of postulates.

Theorem 14

$$\begin{aligned}
 & X + Y = Y + X \\
 \text{Proof.} \quad & \text{If } Y = 0, \quad X + Y = X + 0 \\
 & \qquad\qquad\qquad = X \text{ by Theorem 5.} \\
 & \qquad\qquad\qquad Y + X = 0 + X \\
 & \qquad\qquad\qquad = X \text{ by Theorem 6.} \\
 & \text{Therefore,} \quad X + Y = Y + X \\
 & \text{If } Y = 1, \quad X + Y = X + 1 \\
 & \qquad\qquad\qquad = X \text{ by Theorem 7.} \\
 & \qquad\qquad\qquad Y + X = 1 + X \\
 & \qquad\qquad\qquad = X \text{ by Theorem 8.} \\
 & \text{Therefore,} \quad X + Y = Y + X.
 \end{aligned}$$

Alternatively, this can also be proved taking an example of a 2-input OR gate. If two inputs of a 2-input OR gate are interchanged, the output remains same. Therefore, $X + Y = Y + X$.

Another proof can be obtained by preparing a truth table for an OR gate having inputs X and Y. Take all possible values of X and Y, find the output $X + Y$, and $Y + X$. You will see that $X + Y = Y + X$.

Yet another way is to consider two switches X and Y connected in parallel to illustrate OR operation. If X and Y are interchanged, the result is not affected. So $X + Y = Y + X$.

Theorem 15

$$\begin{aligned}
 & X \cdot Y = Y \cdot X \\
 \text{Proof.} \quad & \text{If } Y = 0, \quad X \cdot Y = X \cdot 0 \\
 & \qquad\qquad\qquad = 0 \text{ by Theorem 2.} \\
 & \qquad\qquad\qquad Y \cdot X = 0 \cdot X \\
 & \qquad\qquad\qquad = 0 \text{ by Theorem 1.} \\
 & \text{If } Y = 1, \quad X \cdot Y = X \cdot 1 \\
 & \qquad\qquad\qquad = X \text{ by Theorem 4.}
 \end{aligned}$$

$$\begin{aligned}
 Y.X &= 1.X \\
 &= X \text{ by Theorem 3.} \\
 \text{Therefore, } X.Y &= Y.X
 \end{aligned}$$

Alternatively, take an example of 2-input AND gate. If inputs of 2-input AND gate are interchanged, the result remains unaffected. Therefore, $X.Y = Y.X$.

Another proof can be obtained by considering two switches X and Y connected in series to illustrate AND operation. If the X and Y are interchanged, the result is not affected. So $XY = YX$.

Yet another way is: prepare truth table for $X.Y$ and $Y.X$ for a 2-input AND gate taking all possible values of inputs X and Y. You will see that $X.Y = Y.X$.

Theorem 20

$$X + XY = X$$

Proof.

$$\begin{aligned}
 X + XY &= X(1 + Y) \\
 &= X.1 \text{ by Theorem 8.} \\
 &= X
 \end{aligned}$$

Theorem 21

$$X(X + Y) = X$$

Proof.

$$\begin{aligned}
 X(X + Y) &= X.X + X.Y \\
 &= X + XY \\
 &= X(1 + Y) \\
 &= X.1 \text{ by Theorem 8.} \\
 &= X
 \end{aligned}$$

Theorem 22

$$XY + X\bar{Y} = X$$

Proof.

$$\begin{aligned}
 XY + X\bar{Y} &= X(Y + \bar{Y}) \\
 &= X.1 \text{ by Theorem 12.} \\
 &= X.
 \end{aligned}$$

Theorem 23

$$(X + Y)(X + \bar{Y}) = X$$

Proof.

$$\begin{aligned}
 (X + Y)(X + \bar{Y}) &= X.X + X\bar{Y} + YX + Y\bar{Y} \\
 &= X + X(Y + \bar{Y}) + 0 \\
 &= X + X.1 \\
 &= X + X \\
 &= X
 \end{aligned}$$

Theorem 24(a)

$$X + \bar{X}Y = X + Y$$

Proof.
rem 20.

$$X + \bar{X}Y = X + XY + \bar{X}Y \text{ putting } X = X + XY \text{ by Theorem 20.}$$

$$\begin{aligned} &= X + Y(X + \bar{X}) \\ &= X + Y \end{aligned}$$

Alternatively, prove this theorem using truth table. Prepare a truth table for all values of X and Y as shown in Table 4.13.

Table 4.13

X	Y	$X + \bar{X}Y$	$X + Y$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	1	1

From the above table we conclude that

$$X + \bar{X}Y = X + Y$$

Theorem 27

Proof.

$$\begin{aligned} XY + \bar{X}Z + YZ &= XY + \bar{X}Z \\ XY + \bar{X}Z + YZ &= XY + \bar{X}Z + YZ(X + \bar{X}) \\ &= XY + \bar{X}Z + YZX + YZ\bar{X} \\ &= XY(1 + Z) + \bar{X}Z(1 + Y) \\ &= XY + \bar{X}Z \end{aligned}$$

Theorem 31

$$\overline{XY} = \bar{X} + \bar{Y} \text{ (Taking only two variables).}$$

Proof. This theorem can easily be proved by truth table approach. For all values of X and Y, we calculate \overline{XY} and $\bar{X} + \bar{Y}$, and tabulate the result as shown in Table 4.14.

Table 4.14

X	Y	\overline{XY}	$\bar{X} + \bar{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

From the above truth table we conclude that

$$\overline{XY} = \bar{X} + \bar{Y}$$

Theorem 32

$$\overline{X + Y} = \bar{X} \cdot \bar{Y} \text{ (Taking only two variables)}$$

Proof. To prove this theorem we can prepare the truth table as shown in Table 4.15.

Table 4.15

X	Y	$\overline{X + Y}$	$\bar{X} \cdot \bar{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

From the above truth table we conclude that

$$\overline{X + Y} = \bar{X} \cdot \bar{Y}$$

4.1.8 Simplification of Boolean Expression by Algebraic Method

Boolean theorems are very useful tools for simplifying logical expressions. Some examples of simplification are given below:

Example 1. Simplify the logical expression

$$X\bar{Y}\bar{Z} + X\bar{Y}\bar{Z}W + X\bar{Z}$$

The above expression can be written as

$$\begin{aligned} X\bar{Y}\bar{Z}(1 + W) + X\bar{Z} &= X\bar{Y}\bar{Z} + X\bar{Z}, \text{ as } 1 + W = 1 && \text{by Theorem 8.} \\ &= X\bar{Z}(\bar{Y} + 1) \\ &= X\bar{Z}, \text{ as } \bar{Y} + 1 = 1 && \text{by Theorem 8.} \end{aligned}$$

Example 2. Simplify the Boolean expression

$$X + \bar{X}Y + \bar{Y} + (X + \bar{Y})\bar{X}Y$$

The above expression can be written as

$$\begin{aligned} X + \bar{X}Y + \bar{Y} + X\bar{X}Y + \bar{Y}\bar{X}Y \\ &= X + \bar{X}Y + \bar{Y}, \quad \text{as } X\bar{X} = 0, \quad X\bar{X}Y = 0, \\ &\qquad\qquad\qquad \text{as } \bar{Y}Y = 0, \quad \bar{Y}\bar{X}Y = 0. \\ &= X + Y + \bar{Y} \quad \text{as } X + \bar{X}Y = X + Y && \text{by Theorem 24(a).} \\ &= X + 1 \quad \text{as } Y + \bar{Y} = 1 && \text{by Theorem 12.} \\ &= 1 && \text{by Theorem 7.} \end{aligned}$$

Example 3. Simplify the logical expression

$$Z(Y + Z)(X + Y + Z)$$

The above expression can be written as

$$\begin{aligned}
 (ZY + ZZ)(X + Y + Z) &= (ZY + Z)(X + Y + Z), & \text{as } ZZ = Z && \text{by Theorem 9.} \\
 &= Z(X + Y + Z), & \text{as } Z + ZY = Z, && \text{by Theorem 20.} \\
 &= ZX + ZY + ZZ \\
 &= ZX + ZY + Z, & \text{as } ZZ = Z, && \text{by Theorem 9.} \\
 &= ZX + Z, & \text{as } Z + ZY = Z, && \text{by Theorem 20.} \\
 &= Z, & \text{as } Z + ZX = Z && \text{by Theorem 20.}
 \end{aligned}$$

Example 4. Simplify the logical expression

$$\bar{X}\bar{Y} + \bar{X}Z + YZ + \bar{Y}Z\bar{W}$$

At first glance it looks that it cannot be reduced further. But applying a trick it can be reduced as follows:

The above expression can be written as

$$\begin{aligned}
 \bar{X}\bar{Y} + \bar{X}Z \cdot 1 + YZ + \bar{Y}Z\bar{W} &= \bar{X}\bar{Y} + \bar{X}Z(Y + \bar{Y}) + YZ + \bar{Y}Z\bar{W} \\
 &= \bar{X}\bar{Y} + \bar{X}ZY + \bar{X}Z\bar{Y} + YZ + \bar{Y}Z\bar{W} \\
 &= \bar{X}\bar{Y}(1 + Z) + YZ(\bar{X} + 1) + \bar{Y}Z\bar{W} \\
 &= \bar{X}\bar{Y} + YZ + \bar{Y}Z\bar{W}
 \end{aligned}$$

Example 5. Simplify the expression

$$(X + Y)(\bar{X} + Z)(Y + Z)$$

The above expression can be written as

$$\begin{aligned}
 (X\bar{X} + XZ + Y\bar{X} + YZ)(Y + Z) &= (XZ + Y\bar{X} + YZ)(Y + Z), \text{ as } X\bar{X} = 0 \\
 &= XZY + YY\bar{X} + YYZ + XZZ + Y\bar{X}Z + YZZ \\
 &= XZY + Y\bar{X} + YZ + XZ + Y\bar{X}Z + YZ
 \end{aligned}$$

Rearranging the terms we get

$$\begin{aligned}
 XZY + XZ + Y\bar{X} + Y\bar{X}Z + YZ, \text{ as } YZ + YZ = YZ \\
 &= XZ(Y + 1) + Y\bar{X} + YZ(\bar{X} + 1) \\
 &= XZ + Y\bar{X} + YZ
 \end{aligned}$$

Now it seems that it cannot be reduced further. But apply the following trick:

The above expression can be written as

$$\begin{aligned}
 XZ + Y\bar{X} + YZ(X + \bar{X}), \text{ as } X + \bar{X} = 1 \\
 &= XZ + Y\bar{X} + YZX + YZ\bar{X} \\
 &= XZ(1 + Y) + Y\bar{X}(1 + Z) \\
 &= XZ + Y\bar{X}
 \end{aligned}$$

4.1.9 Dual and Complement of a Boolean Expression

Two expressions are called *equivalent* only when both are equal to 1 or equal to 0. Two expressions are *complements* of each other if one expression is equal to 1 while the other is equal to 0, and vice versa.

To obtain the complement of a Boolean expression the following changes are made:

- (i) all . signs are changed to + signs
- (ii) all + signs are changed to . signs
- (iii) all 1s are changed to 0s
- (iv) all 0s are changed to 1s
- (v) all literals are complemented.

Example. Find complement of $1.X + \bar{Y}Z + 0$

The complement of the above expression will be

$$(0 + \bar{X})(Y + \bar{Z}).1$$

The *dual* of a Boolean expression is obtained by performing the following operations:

- (i) all . signs are changed to + signs
- (ii) all + signs are changed to . signs
- (iii) all 1s are changed to 0s
- (iv) all 0s are changed to 1s

In finding a dual of an expression literals are not complemented. The following examples illustrate the principle. There is no general rule for the values of dual expressions. Both expressions may be equal to 1 or both may be equal to 0. One may be equal to 1 while the other is equal to 0.

Example 1. Find the dual of the Boolean expression

$$1.X + \bar{Y}Z + 0$$

The dual of the above Boolean expression is

$$(0 + X)(\bar{Y} + Z).1$$

Example 2. Find the dual of the equation

$$X.(Y + Z) = X.Y + X.Z$$

The dual of the above equation will be

$$X + Y.Z = (X + Y)(X + Z)$$

Example 3. Find the dual of $X + XY = X$

The dual of the above expression is

$$X.(X + Y) = X$$

Example 4. Find the dual of $X + \bar{X}.Y = X + Y$

The dual of the above equation is

$$X.(\bar{X} + Y) = X.Y$$

Example 5. Find the dual of the expression $X(\bar{Y} + YZ) + Y\bar{Z}$

The dual of the above expression is

$$\begin{aligned}[X + \bar{Y}.(Y + Z)].(Y + \bar{Z}) &= (X + \bar{Y}Z)(Y + \bar{Z}) \\ &= XY + X\bar{Z}\end{aligned}$$

Boolean postulates 5, 6, 7 and 8 are the duals (or complements, as no literals are involved) of 4, 3, 2 and 1 respectively and vice versa (see Table 4.11).

From Table 4.12, Boolean theorems 5, 6, 7 and 8 are duals of 4, 3, 2 and 1 respectively and vice versa. Again 9 is the dual of 11, 10 is the dual of 12, 14 is the dual of 15, 18 is the dual of 19, 20 is the dual of 21, 22 is the dual of 23 and so on, and vice versa. If any theorem is proved, it is not necessary to prove its dual.

4.1.10 Sum of Products and Product of Sums Forms of Logic Expressions

A logic expression should be in such a form which can easily be realized using logic gates. There are two such forms discussed below.

(i) Sum of Products Form

A sum of products expression consists of several product terms logically added. A product term is a logical product of several variables. The variables may or may not be complemented. The following are examples of sum of products expressions:

- (a) $XY + \bar{X}Y + X\bar{Y}$
- (b) $AB + ABC + B\bar{C}$
- (c) $A + A\bar{B} + \bar{B}C$
- (d) $ABC + \bar{A}B + A\bar{B}C + \bar{A}\bar{B}\bar{C}$

Sometimes a product term may consist of a single variable.

(ii) Product of Sums Form

A product of sums expression consists of several sum terms logically multiplied. A sum term is the logical addition of several variables. The variables may or may not be complemented. The following are examples of product of sums expressions:

- (a) $(A + B)(\bar{A} + \bar{B})$
- (b) $A(\bar{B} + \bar{C})(B + C)$
- (c) $(X + \bar{Y})(X + Y + Z)(Y + Z)$
- (d) $(X + Y + Z)(\bar{X} + Y + Z)(X + \bar{Y} + \bar{Z})$

Sometimes a sum term may consist of a single variable.

4.1.11 Canonical Form of a Logic Expression

When each term of a logic expression contains all variables, it is said to be in the *canonical form*. When a sum of products form of logic expression is in canonical form, each product term is called a *minterm*. Each minterm contains all variables. The canonical form of a sum of products expression is also called *minterm canonical form* or *standard sum of products*.

Similarly, when a product of sums form of logic expression is in canonical form, each sum term is called a *maxterm*. Each maxterm contains all variables. The canonical form of a product of sums expression is also called *maxterm canonical form* or *standard product of sums*.

When a logic expression is not in the canonical form, it can be converted into canonical form. In the canonical form there is a uniformity in the expression, which facilitates minimization procedure.

The following are examples of the canonical form of sum of products expressions (or minterm canonical form):

$$(i) \quad Z = XY + X\bar{Y}$$

$$(ii) \quad F = XY\bar{Z} + \bar{X}YZ + \bar{X}\bar{Y}\bar{Z} + X\bar{Y}Z + XYZ$$

$$(iii) \quad Y = A\bar{B}CD + ABC\bar{D} + \bar{A}BC\bar{D} + \bar{A}\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D}$$

In case of 2 variables, the maximum possible product terms are 4, for 3 variable possible product terms 8, for 4 variables: 16, for n variables, 2^n . In the above examples the expression (ii) contains 5 of 8 possible product terms and (iii) contains six out of 16 possible combinations. When the expression is in the canonical form all terms are *mutually exclusive*. It means that for a given set of values of the variables, when one of the terms is equal to 1, all others must be 0. Of course, it is possible that all terms may be 0.

The following are examples of canonical form of product of sums expressions (or maxterm canonical form):

$$(i) \quad Z = (X + Y)(X + \bar{Y})$$

$$(ii) \quad F = (\bar{X} + Y + \bar{Z})(\bar{X} + Y + Z)(\bar{X} + \bar{Y} + \bar{Z})$$

$$(iii) \quad Y = (A + \bar{B} + C + D)(A + B + \bar{C} + D)(\bar{A} + B + C + D)(A + \bar{B} + C + \bar{D})$$

4.1.12 Conversion of Sum of Products Expressions into Canonical Form

The following examples will illustrate how logic expressions can be converted into canonical form.

Example 1. Convert the expression $X + X\bar{Y}$ into canonical form.

This expression has two variables. The first term has only one variable. So to make it of two variables it can be multiplied by $(Y + \bar{Y})$, as $Y + \bar{Y} = 1$. After multiplication the given logic expression can be written as

$$\begin{aligned} & X(Y + \bar{Y}) + X\bar{Y}, \text{ as } Y + \bar{Y} = 1 \\ &= XY + X\bar{Y} + X\bar{Y} \\ &= XY + X\bar{Y} \end{aligned}$$

Example 2. Convert the following logic expression into canonical form:

$$X + Y\bar{Z}$$

This expression has three variables. The first term has only one variable. It requires two more variables. So it is to be multiplied by $(Y + \bar{Y})$ and $(Z + \bar{Z})$. The second term has two variables. To make it of three variables it is multiplied by $(X + \bar{X})$. After multiplication the given logic expression can be written as

$$\begin{aligned} & X(Y + \bar{Y})(Z + \bar{Z}) + Y\bar{Z}(X + \bar{X}), \text{ as } Y + \bar{Y} = 1, Z + \bar{Z} = 1 \text{ and } X + \bar{X} = 1 \\ &= (XY + X\bar{Y})(Z + \bar{Z}) + XYZ + \bar{X}YZ \\ &= XYZ + XY\bar{Z} + X\bar{Y}Z + X\bar{Y}\bar{Z} + XY\bar{Z} + \bar{X}YZ \end{aligned}$$

$$= XYZ + XY\bar{Z} + X\bar{Y}Z + X\bar{Y}\bar{Z} + \bar{X}YZ,$$

as $XY\bar{Z} + X\bar{Y}Z = XYZ$

Example 3. Convert the following expression into canonical form:

$$XZ + \bar{X}YW + XZ\bar{W}$$

To introduce the missing variables in each term the above expression can be written as

$$\begin{aligned} & XZ(Y + \bar{Y})(W + \bar{W}) + \bar{X}YW(Z + \bar{Z}) + XZ\bar{W}(Y + \bar{Y}) \\ &= (XYZ + X\bar{Y}Z)(W + \bar{W}) + \bar{X}YZW + \bar{X}Y\bar{Z}W + XYZ\bar{W} + X\bar{Y}Z\bar{W} \\ &= XYZW + XYZ\bar{W} + X\bar{Y}ZW + X\bar{Y}Z\bar{W} + \bar{X}YZW + \bar{X}Y\bar{Z}W + XYZ\bar{W} + X\bar{Y}Z\bar{W} \\ &= XYZW + XYZ\bar{W} + X\bar{Y}ZW + X\bar{Y}Z\bar{W} + \bar{X}YZW + \bar{X}Y\bar{Z}W \end{aligned}$$

as 2nd and 7th terms, and 4th and 8th terms are the same.

4.1.13 Conversion of Product of Sums Expressions into Canonical Form

Before we proceed for such conversion a few identities should be examined.

(i) We can write $A = (A + B)(A + \bar{B})$

This can be proved as follows:

$$\begin{aligned} A &= A + A + 0 \\ &= A(B + \bar{B}) + A.A + B.\bar{B}, \text{ as } B + \bar{B} = 1 \\ &= AB + A\bar{B} + AA + B\bar{B} \\ &= A(A + B) + \bar{B}(A + B) \\ &= (A + B)(A + \bar{B}) \end{aligned}$$

(ii) Similarly, we can write $A + B = (A + B + C)(A + B + \bar{C})$

$$(A + B + C)(A + B + \bar{C}) = AA + AB + A\bar{C} + AB + BB + B\bar{C} + AC + BC + C\bar{C}$$

Rearranging the terms we get

$$\begin{aligned} & AA + BB + A\bar{C} + B\bar{C} + AC + BC + AB + AB, \quad \text{as } C\bar{C} = 0 \\ &= (A + B) + \bar{C}(A + B) + C(A + B) + AB + AB \\ &= (A + B) + (A + B)(C + \bar{C}) + AB + AB \\ &= A + AB + B + AB \\ &= A(1 + B) + B(1 + A) \\ &= A + B \end{aligned}$$

This technique can be extended to any number of variables such as

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

Example 1. Convert the following expression into canonical form:

$$(A + B)(B + C)$$

To convert the above expression into canonical form the following identity can be used:

$$X + Y = (X + Y + Z) (X + Y + \bar{Z})$$

Applying the above identity, the given logic expression can be written as

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

Example 2. Convert the following expression into canonical form:

$$X(\bar{Y} + \bar{Z})$$

The first term is of a single variable. It can be first extended to two variables

$$X = (X + Y) (X + \bar{Y})$$

Now the given logic expression can be written as

$$(X + Y) (X + \bar{Y}) (\bar{Y} + \bar{Z})$$

Now extending each term to three variables we get

$$\begin{aligned} & (X + Y + Z) (X + Y + \bar{Z}) (X + \bar{Y} + Z) (X + \bar{Y} + \bar{Z}) (X + \bar{Y} + \bar{Z}) (\bar{X} + \bar{Y} + \bar{Z}) \\ &= (X + Y + Z) (X + Y + \bar{Z}) (X + \bar{Y} + Z) (X + \bar{Y} + \bar{Z}) (\bar{X} + \bar{Y} + \bar{Z}) \end{aligned}$$

as the 4th and 5th terms are the same.

Example 3. Write the following expression in canonical form:

$$(A + B + \bar{C}) (A + D)$$

The above expression can be written as

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

as the 1st and 5th terms are the same.

4.1.14 Truth Tables and Boolean Expressions

The behaviour of a logic network may be described by truth tables or logic expressions. For a given logic expression a truth table can be formed. Similarly, for a given truth table a logic expression can be obtained. A table containing all possible values of the input variables (or independent variables) and the corresponding values of the output (or dependent variables), is known as *truth table*.

Truth Table for Boolean Expression

Take a function $F = X + \bar{Y}$. To find its truth table, all possible values of X and Y are written as shown in Table 4.16. For a given set of values of X and Y, corresponding value of Z is obtained and tabulated as shown in Table 4.16.

Table 4.16 Truth Table of $F = X + \bar{Y}$

X	Y	$Z = X + \bar{Y}$
0	0	1
0	1	0
1	0	1
1	1	1

Let us consider another function $F = X + YZ$, and write all possible values of X, Y and Z. Find corresponding values of F as shown in Table 4.17.

Table 4.17 Truth Table of $F = X + YZ$

X	Y	Z	$F = X + YZ$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

4.1.15 Boolean Expression in Sum of Products Form from a Given Truth Table

If a truth table to describe the behaviour of a logic network is given, the corresponding logic expression can be obtained either in sum of products form or product of sums form. The following example will illustrate the procedure to obtain the logic expression in sum of products form.

Example. Table 4.18 describes the behaviour of a digital circuit. Find the corresponding logic expression in sum of products form to describe the behaviour of the circuit.

Table 4.18

X	Y	Z	$F = f(X, Y, Z)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$F = f(X, Y, Z)$ is the desired logic expression. Its values are given in the table corresponding to each set of values of X, Y and Z. A sum of products expression is obtained by selecting those product terms for which the function $F = 1$ and logically adding them. In the

5th, 6th and 8th rows we find $F = 1$. The product term for the 5th row will be $X\bar{Y}\bar{Z}$ which makes $F = 1$ for the given set of values of variables X, Y and Z. The product term $X\bar{Y}\bar{Z}$ will be 0 for any other set of values of X, Y and Z. So it describes the condition of the 5th row of the truth table. Similarly, product terms which give $F = 1$ are $X\bar{Y}Z$ and XYZ for 6th and 8th rows respectively. The desired logical expression is given by

$$F = X\bar{Y}\bar{Z} + X\bar{Y}Z + XYZ$$

This function will satisfy the Truth Table 4.18. This expression is in canonical sum of products form. Each term of this expression is mutually exclusive (See definition in Sec. 4.1.11). Each term of the above expression corresponds to only one row in which $F = 1$. Thus the expression corresponds to the collection of three rows. For the remaining five rows this expression will have $F = 0$. So the expression satisfies Truth Table 4.18.

The above expression can be simplified as

$$\begin{aligned} F &= X\bar{Y}\bar{Z} + XZ(Y + \bar{Y}) \\ &= X\bar{Y}\bar{Z} + XZ \\ &= X(Z + \bar{Z}\bar{Y}) \\ &= X(Z + \bar{Y}) \quad \text{by Theorem 24(a)} \\ &= X\bar{Y} + XZ \end{aligned}$$

4.1.16 Boolean Expression in Product of Sums Form from a Given Truth Table

A product of sums expression can be formed for Truth Table 4.18 by selecting those sum terms for which $F = 0$ and multiplying them. The 1st row gives $F = 0$. The sum term is $(X + Y + Z)$. The 2nd terms is also 0, the corresponding sum term which gives $F = 0$ is $(X + Y + \bar{Z})$. Similarly, all other sum terms are formed and the desired expression is given below:

$$F = (X + Y + Z)(X + Y + \bar{Z})(X + \bar{Y} + Z)(X + \bar{Y} + \bar{Z})(\bar{X} + \bar{Y} + Z).$$

This expression can be simplified on applying identity (ii) of Sec. 4.1.13 to

$$\begin{aligned} F &= (X + Y)(X + \bar{Y})(\bar{X} + \bar{Y} + Z) \\ &= X(\bar{X} + \bar{Y} + Z), \quad \text{applying identity (i) of Sec. } 4.1.13. \\ &= X\bar{X} + X\bar{Y} + XZ \\ &= X\bar{Y} + XZ \end{aligned}$$

Now it can be seen that the logic expressions (for Truth Table 4.18) in sum of products form and product of sums form are equivalent. A generalized proof for this is also given in Section 4.1.17.

4.1.17 Proof for Obtaining Expression in Product of Sums Form from a Given Truth Table

Our aim is to obtain a logical expression of a digital system which behaves according to the truth table given in Table 4.18. This table can be rewritten as shown in Table 4.19 with an additional column \bar{F} .

For the complementary function \bar{F} we can write an expression in minterm canonical form:

$$\begin{aligned}\bar{F} &= \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + XYZ \\ F = \bar{\bar{F}} &= (\bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + \bar{X}YZ + XYZ)\end{aligned}$$

Applying Demorgan's theorem this can be written as

$$\begin{aligned}F &= (\bar{X}\bar{Y}\bar{Z})(\bar{X}\bar{Y}Z)(\bar{X}Y\bar{Z})(\bar{X}YZ)(XYZ) \\ &= (X+Y+Z)(X+Y+\bar{Z})(X+\bar{Y}+Z)(X+\bar{Y}+\bar{Z}) \\ &\quad (\bar{X} + \bar{Y} + Z)\end{aligned}$$

This expression can also be written directly by collecting those sum terms for which $F = 0$, as we have done in Sec. 4.1.16. So algebraically it has been proved that the technique used in Sec. 4.1.16 is correct.

Table 4.19

X	Y	Z	F	\bar{F}
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

4.1.18 Implementing Logic Expressions with Logic Gates

(i) Logic Expression in Sum of Product Form

If logic expressions are given in sum of products form the logic network can be realized using AND-OR gates or only NAND gates. Take a simple expression given below:

$$Z = AB + CD$$

The given expression will require two AND gates and one OR gate. The logic circuit is shown in Fig. 4.3. As NAND gates are universal gates, they are used as building blocks for the realization of logic networks. An AND-OR network shown in Fig. 4.3 can be replaced by an equivalent NAND network as shown in Fig. 4.4. Similarly, a sum of products expression having more variables can be realized using AND-OR or equivalent NAND network.

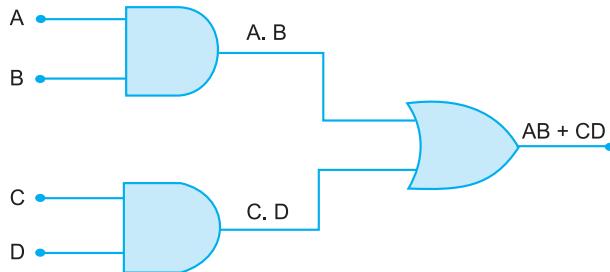


Fig. 4.3 AND-OR Network to realise logic expression $Z = AB + CD$.

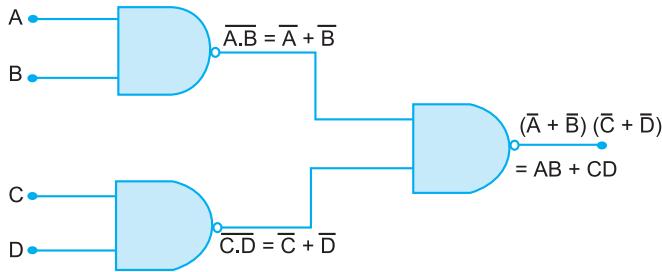


Fig. 4.4 NAND network to realize logic expression $Z = AB + CD$.

Take another example given below:

$$Y = A\overline{B}C + A\overline{B}\overline{C} + ABC$$

Figs. 4.5 and Fig. 4.6 show AND-OR and NAND network respectively for the above logic expression.

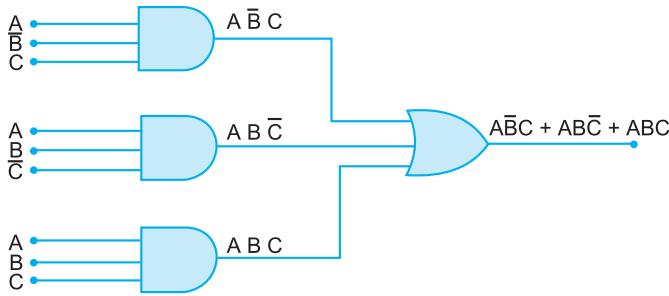


Fig. 4.5 AND-OR Network for logic expression $Y = A\overline{B}C + A\overline{B}\overline{C} + ABC$.

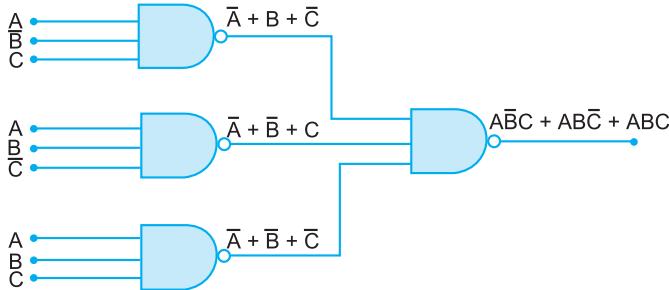


Fig. 4.6 NAND Network for logic expression $Y = A\overline{B}C + A\overline{B}\overline{C} + ABC$.

(ii) Logic Expressions in Product of Sums Forms

If logic expressions are given in product of sums forms, the logic network can be realized using OR-AND gates or only NOR gates. Take an example given below:

$$W = (A + B + C)(X + Y + Z)$$

Figs. 4.7 and 4.8 show OR-AND and NOR network respectively for the above logic expression.

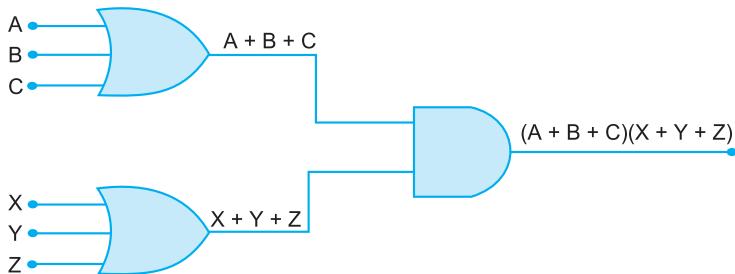


Fig. 4.7 OR-AND Network for logic expression $W = (A + B + C) (X + Y + Z)$.

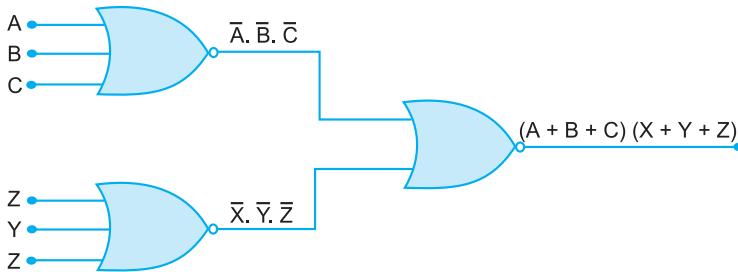


Fig. 4.8 NOR Network for logic expression $W = (A + B + C) (X + Y + Z)$.

4.2 MINIMUM BOOLEAN EXPRESSION

A logic expression is simplified before a logic network is realized using logic gates. The simplification is essential to minimize the cost of the logic network. The expression must be minimized before implementation. It should be either a minimum sum of products or minimum product of sums. A minimum sum of products is one which contains either minimum number of terms of products or minimum number of literals. Similarly, a minimum product of sum is one which contains minimum number of literals or minimum number of factors. The type of minimum expression desired, containing minimum terms or minimum literals, depends on the circumstances.

We have already seen that a logic expression can be simplified by algebraic method using Boolean theorems. The drawback of this technique is that there are no specific rules for proceeding step by step to manipulate the process of simplification. Even if a minimum is obtained, one may not be sure that it is minimum. If the expression is quite complex, it is very difficult to obtain a minimum by algebraic manipulation. For minimization of Boolean functions having more than three variables the algebraic method becomes inconvenient. The Karnaugh map method provides a simple straightforward technique of simplification of logic expressions. This is discussed in the next section.

4.3 KARNAUGH MAP METHOD OF SIMPLIFICATION OF LOGIC EXPRESSION

The Karnaugh map method is a graphical technique for simplifying Boolean functions. The Karnaugh map is a two-dimensional representation of a truth table. It provides a simpler method for minimizing logic expressions. The map method is ideally suited for four or less variables. But it becomes cumbersome for five or more variables. A Karnaugh map is a diagram consisting of squares. Each square of the map represents a minterm. Any logic

expression can be written as a sum of products, i.e. sum of minterms. Therefore, a logic expression can easily be represented on a Karnaugh map.

A Karnaugh map for n variables is made up of 2^n squares. Each square designates a product term of a Boolean expression. For product terms which are present in the expression, 1s are written in the corresponding squares; 0s are written in those squares which correspond to product terms not present in the expression. For clarity of the map writing of 0s can be omitted. So blank squares indicate that they contain 0s.

First let us consider a map of two variables as shown in Fig. 4.9(a) to explain the principle. The first row is for \bar{B} , the 2nd row for B . Similarly, 1st column is for \bar{A} and 2nd column for A . For any square, see the variables in both row as well as column. For the 1st square of the 1st row, variables are \bar{B} and \bar{A} , and hence it will represent the product term $\bar{A}\bar{B}$. For the 2nd square of the 1st row, the variables are A and \bar{B} , so it represents $A\bar{B}$.

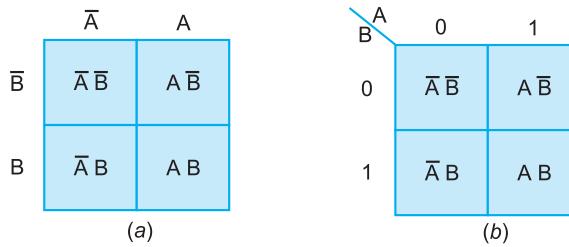


Fig. 4.9 Two variable Karnaugh Map.

0 and 1 written at the top of the map shown in Fig. 4.9(b) indicate 0 and 1 logic for variable A . In other words 0 represents \bar{A} , and 1 represents A . Similarly, 0 and 1 logic are for variable B (i.e. 0 represents \bar{B} and 1 represent B).

Suppose the Karnaugh maps are to be drawn for the Boolean functions given below:

$$(a) \quad Y = A\bar{B} + AB$$

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

In these functions only two product terms are present. So 1s are written in the corresponding squares as shown in Fig. 4.10(a) and (b).

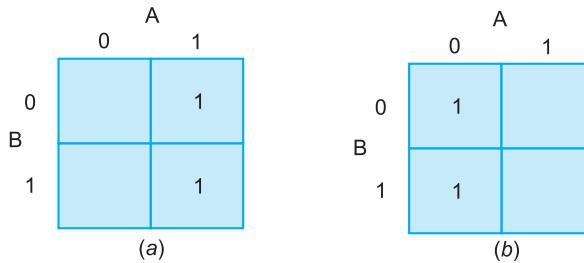


Fig. 4.10 Karnaugh map for (a) $Y = A\bar{B} + AB$ (b) $Y = \bar{A}\bar{B} + \bar{A}B$.

From the map shown in Fig. 4.9 it can be seen that two adjacent squares differ only by one variable. They can be grouped for simplification. Larger number of adjacent squares showing 1 can also be grouped for simplification. The following examples will show the procedure.

Example 1. Simplify

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

Draw Karnaugh map of the above function as shown in Fig. 4.11.

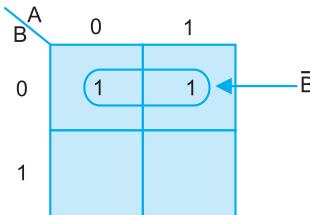


Fig. 4.11 Karnaugh map for $Y = \bar{A}\bar{B} + A\bar{B}$.

Two adjacent squares containing 1 have been grouped together. To show the grouping, they have been encircled. For simplification we have to see that which variable is common to both squares. In this case \bar{B} is common to both as the 1st row is for \bar{B} . So their simplification will result $Y = \bar{B}$. This can be verified also algebraically as follows:

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

So the variable which is common to adjacent squares is selected, and the variable which is not common is discarded.

Example 2. Simplify

$$Y = \bar{A}B + AB + A\bar{B}$$

Fig. 4.12 shows the Karnaugh map of the above function. Groups of adjacent 1-squares are made as shown in the figure. In this case the 2nd column of the 2nd row is common to both groupings. The group of horizontal 1-squares gives result = B, and vertical 1-squares = A. So the simplified expression is $Y = A + B$.

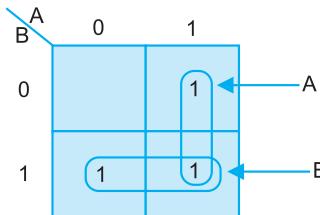


Fig. 4.12 Karnaugh map of $Y = \bar{A}B + AB + A\bar{B}$.

The result obtained by map method can also be verified algebraically as follows:

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

Fig. 4.13(a), (b), (c) and (d) show Karnaugh maps and simplification of the following functions:

- (a) $Y = \bar{A}\bar{B} + \bar{A}B$
 (b) $Y = A\bar{B} + AB$
 (c) $Y = \bar{A}\bar{B} + A\bar{B} + \bar{A}B$
 (d) $Y = \bar{A}B + AB$

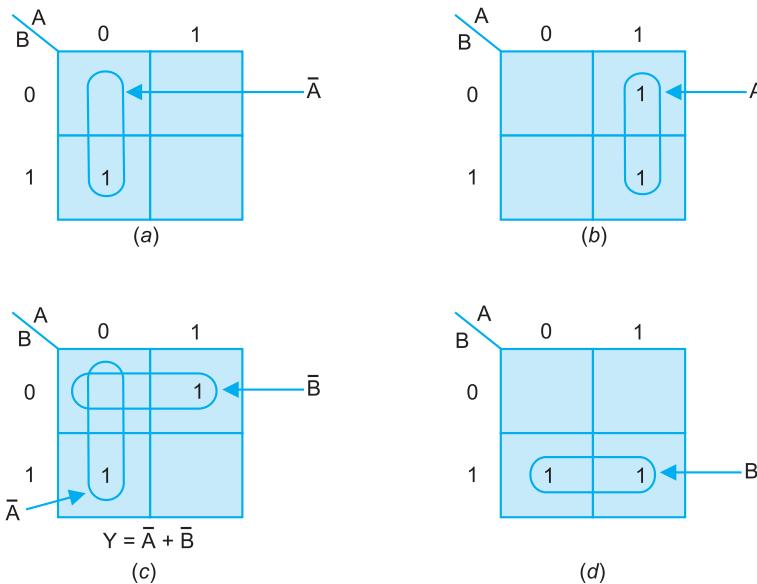


Fig. 4.13 Karnaugh map for a few functions.

4.3.1 Karnaugh Map for Three Variables

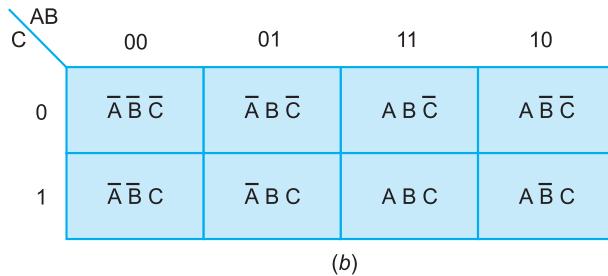
Fig. 4.14(a) shows Karnaugh map for three variables. Fig. 4.14(b) shows the alternative way of representing the variables. The ordering of the variables, i.e., 00, 01, 11, 10 is in Gray (reflected binary) code. One should not write straight binary codes, i.e., 00, 01, 10 and 11. The straight binary codes are used in Veitch diagram. Mr. Karnaugh modified the Veitch diagram and used reflected binary codes.

While forming groups of adjacent squares containing 1s the following considerations must be kept in mind:

- (i) Every square containing 1 must be considered at least once.
- (ii) A square containing 1 can be included in as many groups as desired.
- (iii) A group must be as large as possible.

		$\bar{A}\bar{B}$	$\bar{A}B$	$A\bar{B}$	$A\bar{B}$	
		\bar{C}	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$A\bar{B}\bar{C}$	$A\bar{B}\bar{C}$
		C	$\bar{A}\bar{B}C$	$\bar{A}B C$	$A\bar{B} C$	$A\bar{B} C$

(a)

**Fig. 4.14** Karnaugh map for three variables.

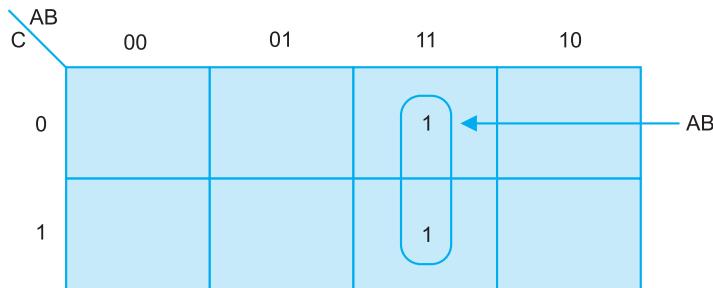
- (iv) A square containing 1 should not be left alone to be included in the final expression, if there is possibility of its inclusion in a group of two squares containing 1. Similarly, a group of two 1-squares (squares containing 1) should not be made if these 1-squares can be included in a group of four 1-squares; and so on.
- (v) The number of squares in a group must be equal to 2^n , such as 2, 4, 8, 16 etc. It cannot be 3, 5, 6, 7, 9, etc.
- (vi) The map is considered to be folded or cylindrical. Therefore, squares at the ends of a row or column are treated as adjacent squares.
- (vii) The simplified logic expression obtained from Karnaugh map is not unique. Groupings of 1-squares can be made in different ways, which results in more than one logic expressions.
- (viii) Before drawing Karnaugh map the logic expressions must be in canonical (or expanded) forms.

The following examples will illustrate the procedure of simplification.

Example 1. Simplify the function $Y = ABC + AB\bar{C}$, by Karnaugh map method.

Fig. 4.15 shows the Karnaugh map for Example 1. The squares containing 1s are adjacent squares. So they can be grouped together. The variables AB are common to both squares, so they are selected. There is no other variable common both squares. So uncommon variables, i.e. C and \bar{C} are discarded. The simplified function will be

$$Y = AB$$

**Fig. 4.15** Karnaugh map for Example 1.

Example 2. Using Karnaugh map method simplify the expression

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

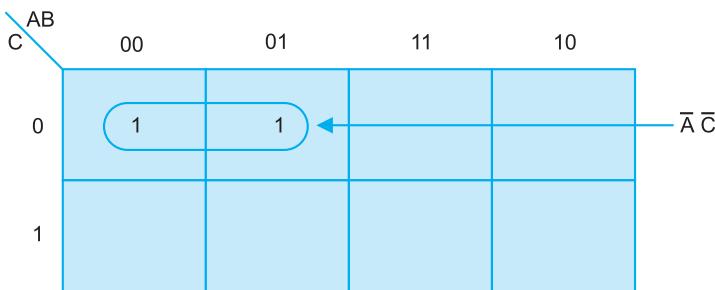


Fig. 4.16 Karnaugh map for Example 2.

Figure 4.16 shows the Karnaugh map for Example 2. Observe the diagram from both row wise and column wise for common variables. Row wise it is seen that \bar{C} is common to both 1-squares. Column wise it is seen that \bar{A} is common to both. So considering all the three variables $\bar{A}\bar{C}$ is common to both squares. The simplified function is

$$Y = \bar{A}\bar{C}$$

Example 3. Simplify the function $Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C$

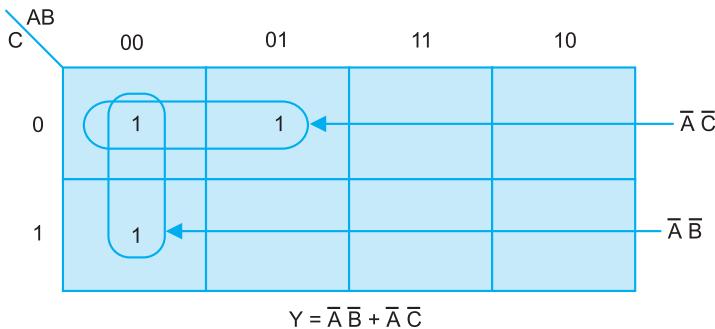


Fig. 4.17 Karnaugh map for Example 3.

Figure 4.17 shows the Karnaugh map for Example 3. Groups of adjacent 1-squares have been made as shown in the figure. In this case the square representing $\bar{A}\bar{B}\bar{C}$ is common to two groups. The simplified function is

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

This expression is the minimum sum of products. It contains 4 literals. It will require two AND gates and one OR gate for its realization. But if it is expressed in product of sums form it will be

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

This contains only 3 literals. It will require one AND gate and one OR gate. Thus we see that we have to consider both forms of simplified expression before the implementation of Boolean function.

Example 4. Simplify the function $Y = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$

Figure 4.18 shows the Karnaugh map for Example 4. The squares at the ends of a row or column are considered to be adjacent squares because the map is treated as a folded or cylindrical one. So two 1-squares of Fig. 4.18 are adjacent squares. They can be grouped together. The simplified function will be

$$Y = \overline{B}\overline{C}$$

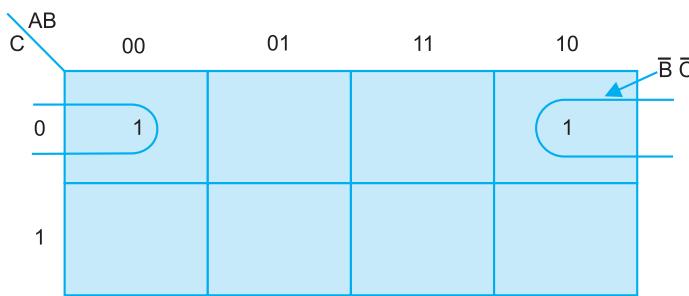


Fig. 4.18 Karnaugh map for Example 4.

Example 5. Simplify $Y = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C}$

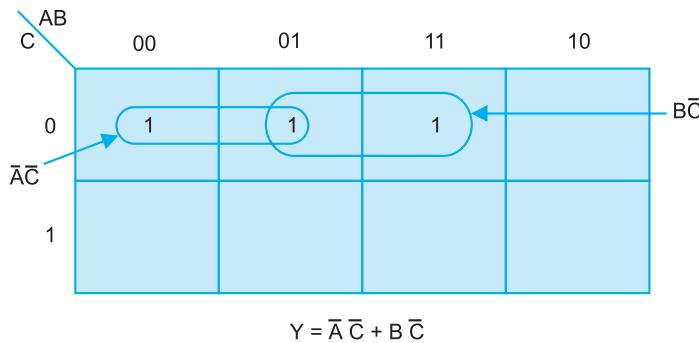


Fig. 4.19 Karnaugh map for Example 5.

Figure 4.19 shows the Karnaugh map for Example 5. In this example we see that there are three adjacent squares containing 1. As the number of 1-squares in a group must be some power of 2 such as 2, 4, 8, 16 and so on. We cannot form the group of 3. So two groups of 1-squares have been formed as shown in the figure. The simplified function will be

$$Y = \overline{A}\overline{C} + B\overline{C}$$

Example 6. Simplify $Y = ABC + A\overline{B}\overline{C} + A\overline{B}C + A\overline{B}\overline{C}$

Figure 4.20 shows the Karnaugh map for Example 6. If we observe column wise, we find that the variable A is common to all the four squares. If we see row wise, no variable is common to all the four squares. Therefore, the function after simplification is

$$Y = A$$

Example 7. Fig. 4.21 represents a Boolean function on a three variable Karnaugh map. Find the Boolean function.

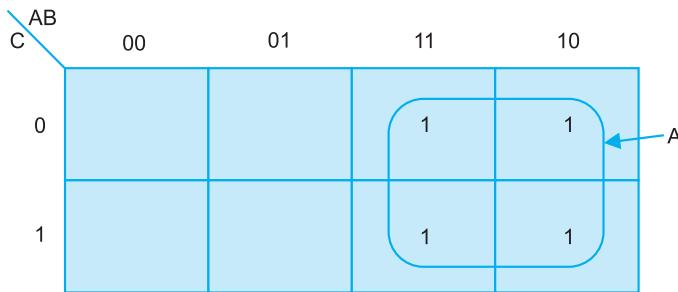


Fig. 4.20 Karnaugh map for Example 6.

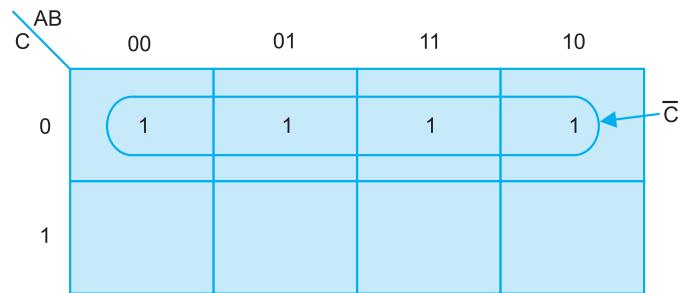


Fig. 4.21 Karnaugh map for Example 7.

In Fig. 4.21 four 1-squares are adjacent squares. So they have been grouped together. From row side \bar{C} is common to all 1-squares. From column side no variable is common to all. So the common variable is selected, and uncommon are discarded. The simplified function will be

$$Y = \bar{C}$$

Example 8. Fig. 4.22 represents a Boolean function on a three variable Karnaugh map. Find the Boolean function.

As the Karnaugh map is treated as a cylindrical map, four 1-squares of Fig. 4.22 are adjacent squares. The variable \bar{B} is common to all 1-squares. The simplified function is given by

$$Y = \bar{B}$$

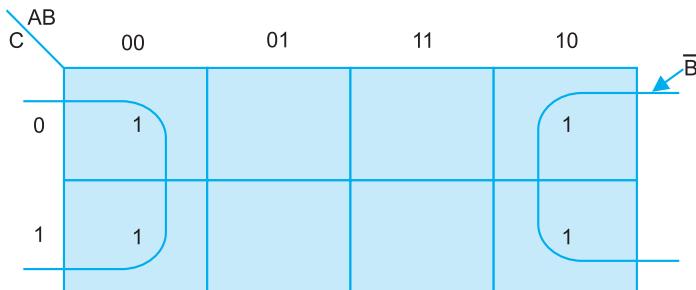


Fig. 4.22 Karnaugh map for Example 8.

4.3.2 Karnaugh Map for Four Variables

The representation of variables and product terms of a four variable Karnaugh map is shown in Fig. 4.23.

		AB	00	01	11	10
		CD	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
		00	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$	$A\bar{B}\bar{C}\bar{D}$
		01	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}B\bar{C}D$	$A\bar{B}\bar{C}D$	$A\bar{B}\bar{C}D$
		11	$\bar{A}\bar{B}CD$	$\bar{A}BCD$	$ABC\bar{D}$	$A\bar{B}CD$
		10	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}BC\bar{D}$	$ABC\bar{D}$	$A\bar{B}C\bar{D}$

Fig. 4.23 Four variable Karnaugh map.

The following examples illustrate the method of simplification of four variable Boolean functions.

Example 9. Simplify $Y = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D + \bar{A}BCD + ABCD + \bar{A}\bar{B}\bar{C}\bar{D}$

Fig. 4.24 shows the Karnaugh map of Example 9.

Four 1-squares being adjacent squares have been grouped together. Column wise the variable B is common to all the four squares. Row wise the variable D is common to all the four squares. So these four squares will give BD after simplification. The square representing $\bar{A}\bar{B}\bar{C}\bar{D}$ is left alone, so it is to be considered as a group of single 1-square. The simplified function is given by

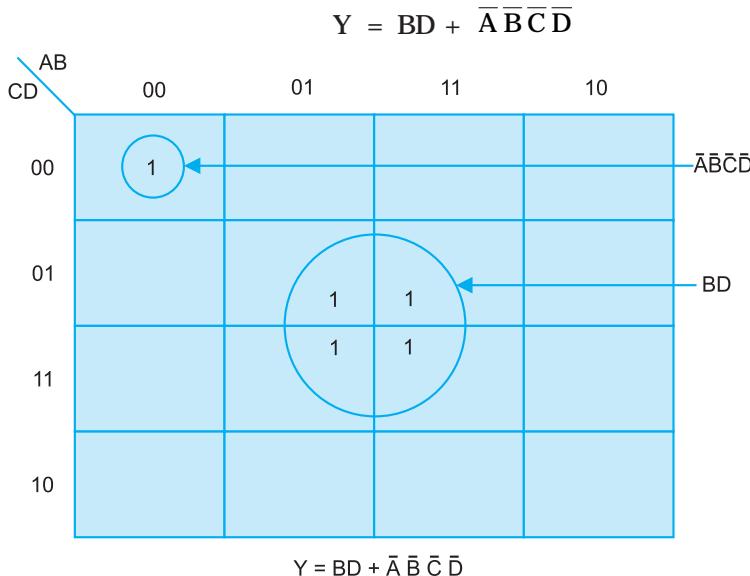


Fig. 4.24 Karnaugh map for Example 9.

Example 10. Figure 4.25 represents a logic function on a four variable Karnaugh map. Find the logic function.

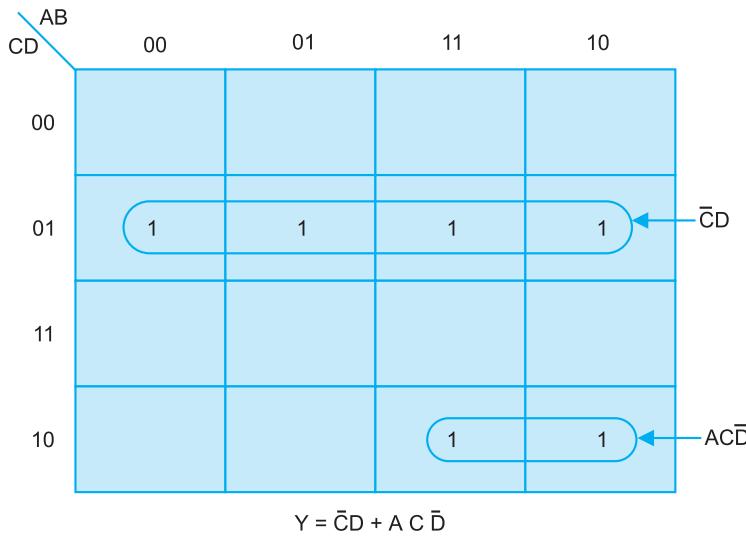


Fig. 4.25 Karnaugh map for Example 10.

Groupings of 1-squares have been shown in the figure. In the 1st group there are four adjacent 1s. From row wise $\bar{C}D$ is common to these 4 adjacent squares. From column wise no variable is common to all the four squares. So these four squares result in $\bar{C}D$. In the 2nd group there are two 1-squares. From row wise $\bar{C}D$ is common to them. From column wise A is common to both. So these two squares give $A\bar{C}\bar{D}$. The simplified expression is

$$Y = A\bar{C}\bar{D} + \bar{C}D$$

Example 11. Figure 4.26 represents a Boolean function on a four variable Karnaugh map. Find the Boolean function.

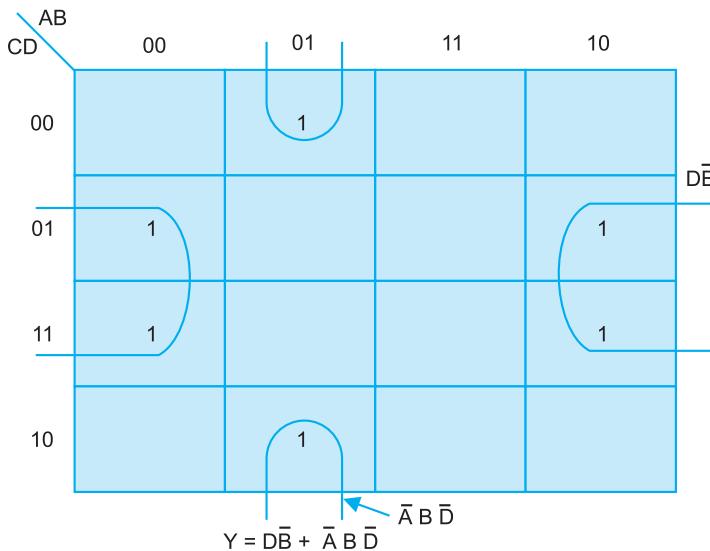


Fig. 4.26 Karnaugh map for Example 11.

Two 1-squares at the left edge and two 1-squares at the right-edge are considered adjacent because the map is treated as a cylindrical map with left and right edges touching each other. This group of four squares corresponds to the term $D\bar{B}$. Again one square at the top and one at the bottom is grouped together as the map is also treated as top and bottom edges touching each other. This group of two squares corresponds to the term $\bar{A}BD$. The simplified Boolean function is given by

$$Y = \bar{A}BD + D\bar{B}$$

Example 12. Figure 4.27 represents a Boolean function on a four variable Karnaugh map. Find the Boolean function.

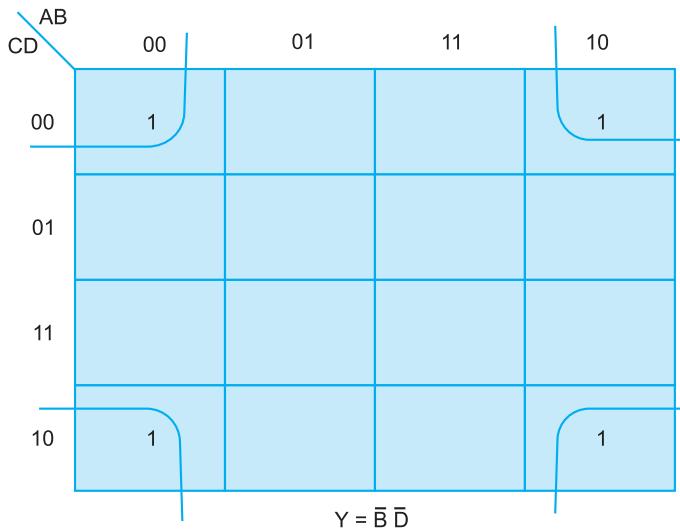
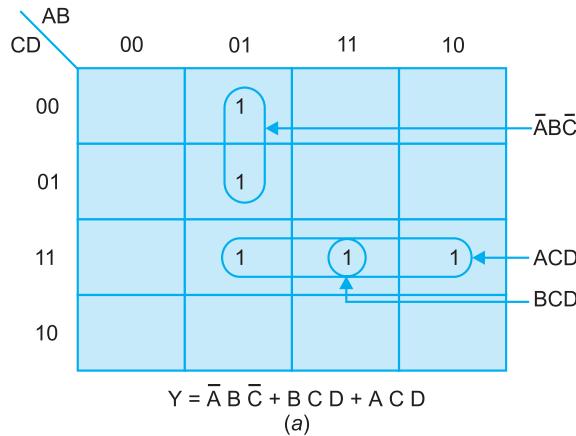


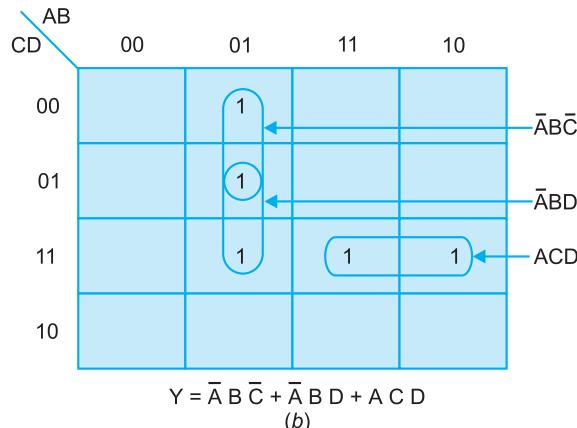
Fig. 4.27 Karnaugh map for Example 12.

Seeing the map row wise, the variable \bar{D} is common to all the four squares. Seeing the map column wise, the variable \bar{B} is common to all the four squares. Therefore, B is common to all the four squares. The simplified Boolean function is given by

$$Y = \bar{B}\bar{D}$$

Example 13. Figure 4.28 represents a Boolean function on a four variable Karnaugh map. Find the Boolean function.



**Fig. 4.28** Karnaugh map for Example 13.

There are two possible groups of 1-squares as shown in Figs. 4.28(a) and (b). Corresponding to Fig. 4.28(a) the simplified Boolean function is

$$Y = \bar{A}B\bar{C} + BCD + ACD$$

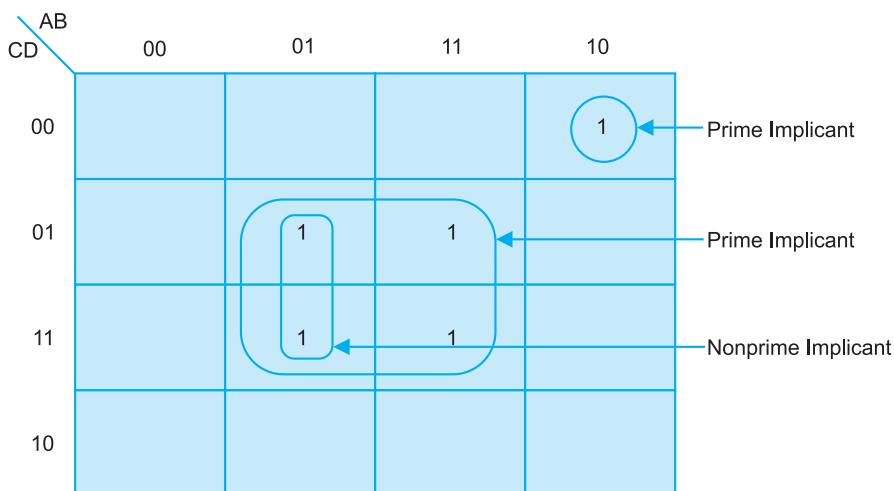
Corresponding to Fig. 4.28(b) the simplified Boolean function is

$$Y = \bar{A}B\bar{C} + \bar{A}BD + ACD$$

Thus we see that the Boolean function obtained from a Karnaugh map may not be unique.

4.3.3 Prime Implicant

In a sum of products expression each product term is known as *implicant*. On a Karnaugh map each implicant relates to a single 1-square or a group of adjacent 1-squares. In other words on a Karnaugh map an implicant is denoted by an enclosure containing adjacent 1-squares or a single 1-square. A *prime implicant* is an implicant which cannot be wholly enclosed by a larger implicant on a Karnaugh map. A nonessential prime implicant can be removed without leaving any 1-square unenclosed. Fig. 4.29 shows prime and nonprime implicants. Fig. 4.30 shows essential and nonessential prime implicants.

**Fig. 4.29** Prime and nonprime implicant.

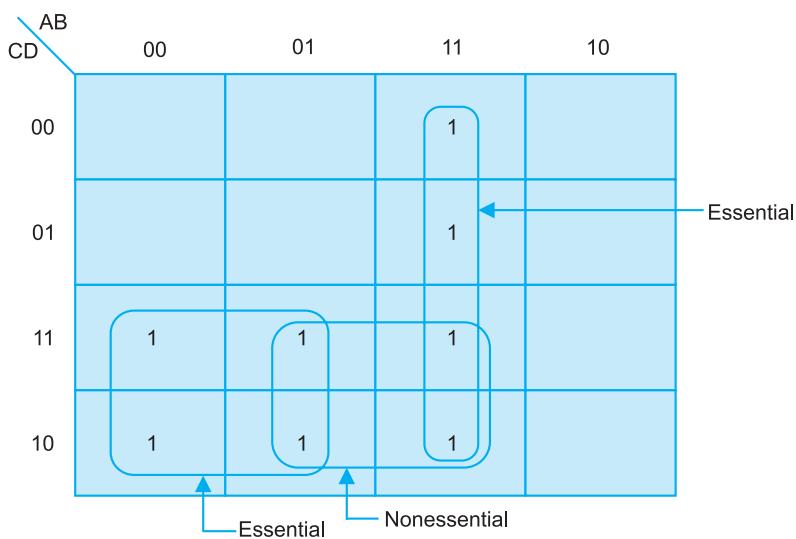


Fig. 4.30 Essential and nonessential prime implicant.

4.3.4 Labelling of Karnaugh Map Squares

The squares of a Karnaugh map can be numbered as shown in Fig. 4.31. These numbers are equal to the binary equivalent of the logic values of the corresponding minterm variables.

The number in the 3rd column of the 1st row is 12. This square is for the minterm $AB\bar{C}\bar{D}$. The logic values of $AB\bar{C}\bar{D}$ are 1100 which is equal to the decimal number 12. If positions of AB and CD are interchanged as shown in Fig. 4.32, the numbering of square will also change. In this diagram the minterm $AB\bar{C}\bar{D}$ is in 3rd row and 1st column. The logic values for the 3rd row, 1st column are 1100, i.e. 12 (decimal).

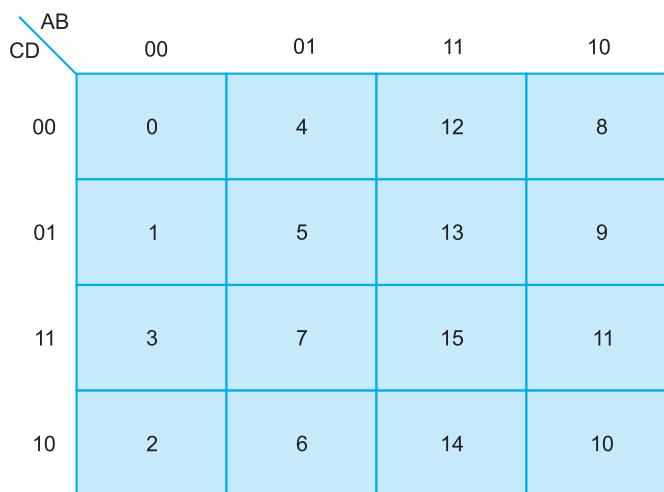


Fig. 4.31 Numbering of 4-variable Karnaugh map squares.

		CD AB	00	01	11	10
		00	0	1	3	2
		01	4	5	7	6
		11	12	13	15	14
		10	8	9	11	10

Fig. 4.32 Labelling of 4-variable Karnaugh map squares.

Similarly, 3-variable Karnaugh map can also be numbered as shown in Figs. 4.33(a) and (b). The minterm for the 1st row, 3rd column of Fig. 4.33(a) is ABC . The logic values of variables of this minterm are 110 (i.e. 6 decimal). Similarly, for 1st row, 4th column, the minterm is $A\bar{B}\bar{C}$. The logic values of the variables of this minterm are 100 (i.e. 4 decimal).

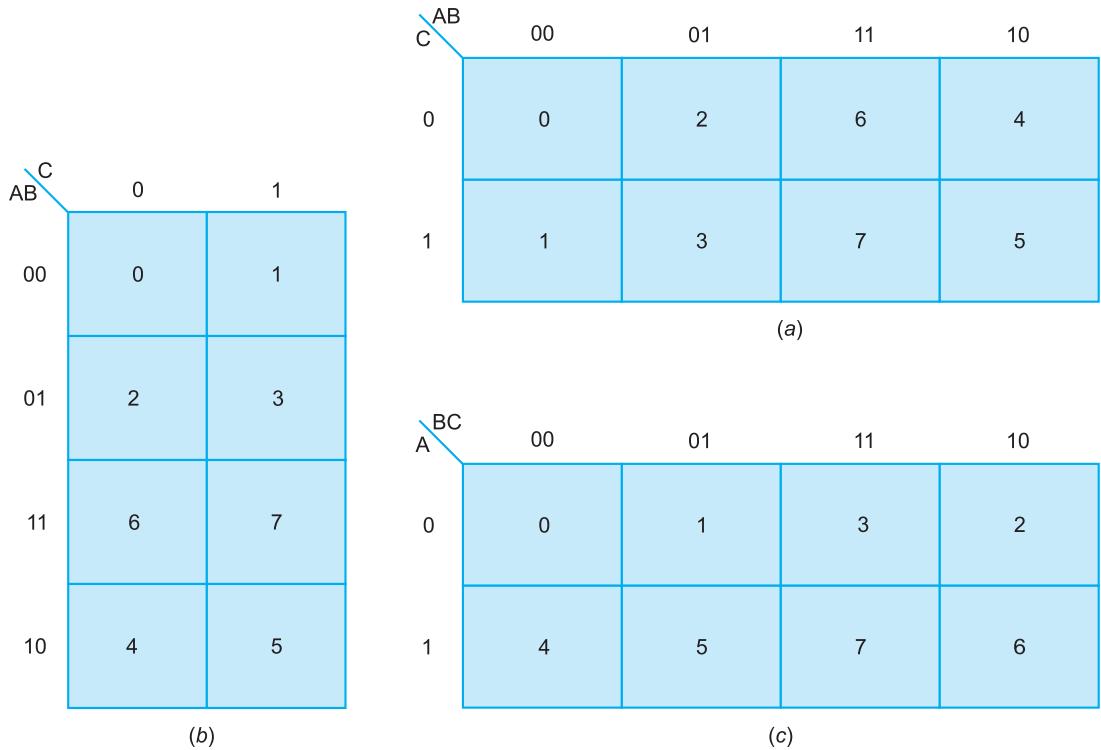


Fig. 4.33 Numbering of squares of 3-variable Karnaugh map.

4.3.5 Alternative Way of Representing Sum of Products Expression

The following examples will illustrate the procedure:

Example 14. Consider the following Boolean function:

$$Y = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC$$

This can also be represented as

$Y(A, B, C) = \Sigma(2, 6, 7)$ or $\Sigma m(2, 6, 7)$, where m stands for minterm.

The above expression means that the minterm corresponding to squares 2, 6 and 7 of the Karnaugh map are present in the given Boolean function. The squares 2, 6 and 7 will be marked by 1 and other squares of the map by 0 as shown in Fig. 4.34.

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

Fig. 4.34 Karnaugh map for Example 14.

Example 15. Draw the Karnaugh map for the following Boolean function:

$$Y(A, B, C, D) = \Sigma m(4, 5, 10, 11, 15)$$

$$\text{or } \Sigma(4, 5, 10, 11, 15)$$

The Karnaugh map of the above Boolean function is as shown in Fig. 4.35.

		AB	00	01	11	10
		CD	00	1	0	0
			0	1	0	0
0	0	0	1	0	0	0
0	1	0	1	0	0	0
1	0	0	0	1	1	0
1	1	0	0	0	0	1

Fig. 4.35 Karnaugh map for Example 15.

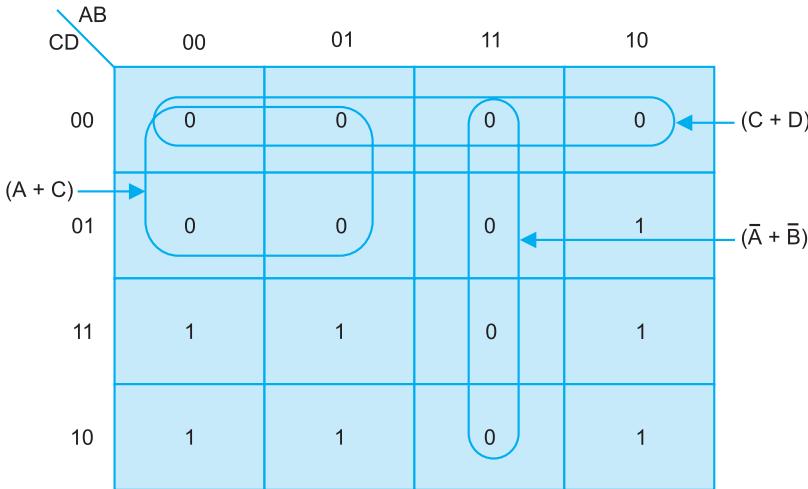
4.3.6 Realization of Product of Sums Expression from Karnaugh Map

To realize a product of sums expression from a Karnaugh map groups of 0-squares are made, and first a sum of products of complementary function is obtained. Then applying Demorgan's law the product of sums expression is obtained.

Example 16. Find the minimum product of sums expression from the Karnaugh map shown in Fig. 4.36.

Groups of 0-squares are made as shown in Fig. 4.36. The complementary function will be

$$\bar{F}(A, B, C, D) = AB + \bar{A}\bar{C} + \bar{C}\bar{D}$$



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

Fig. 4.36 Karnaugh map for Example 16.

Applying Demorgan's law, the following minimum product of sums is obtained

$$\begin{aligned} F(A, B, C, D) &= \overline{(AB + \bar{A}\bar{C} + \bar{C}\bar{D})} \\ &= (\bar{A} + \bar{B})(A + C)(C + D) \end{aligned}$$

This final expression can also be written directly by applying the technique given in Sec. 4.3.7.

The squares containing 0s of Fig. 4.36 can also be represented as

$$F(A, B, C, D) = \pi(0, 1, 4, 5, 8, 12, 13, 14, 15)$$

or

$$\pi M(0, 1, 4, 5, 8, 12, 13, 14, 15)$$

where M represents maxterm.

Example 17. Find the minimum product of sums from the Karnaugh map shown in Fig. 4.37.

Groups of 0-squares are made as shown in Fig. 4.37. The complementary function is given by

$$\bar{F}(A, B, C) = \bar{A}\bar{C} + BC$$

Applying Demorgan's law, we have

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

This expression can also be written directly by applying the technique given in Sec 4.3.7.

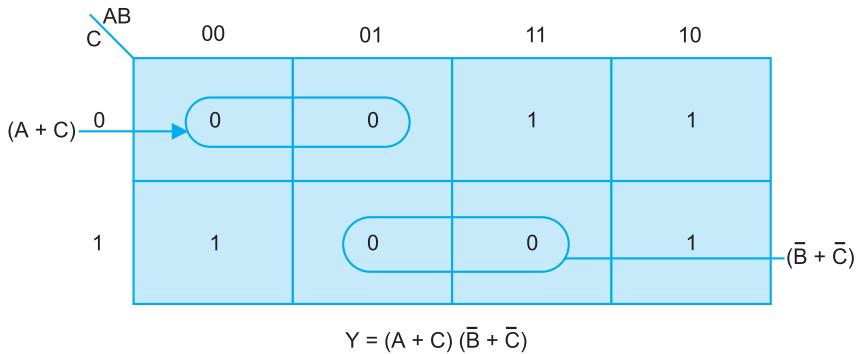


Fig. 4.37 Karnaugh map for Example 17.

4.3.7 Karnaugh Map for Product of Sums Expression

In this case we have to see that the sum term for a particular square of a Karnaugh map should be 0. Consider the Karnaugh map shown in Fig. 4.38. The 1st row and 1st column will be 0, if $A + B + C = 0$. In other words the sum term for the 1st row, 1st column is $(A + B + C)$. Similarly, the sum terms for the 1st row, 4th column will be $\bar{A} + B + C$. Again, the sum term for the 2nd row, 4th column will be $\bar{A} + B + \bar{C}$. Applying this rule a Karnaugh map for a given product of sums expression can be drawn.

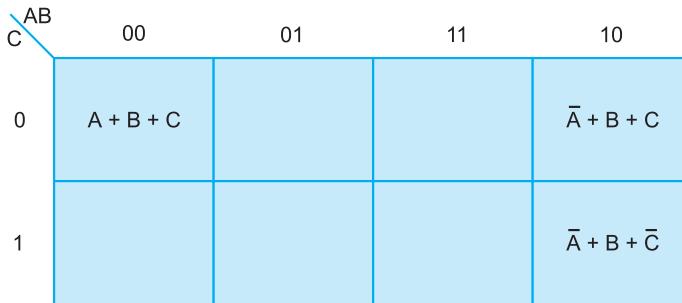


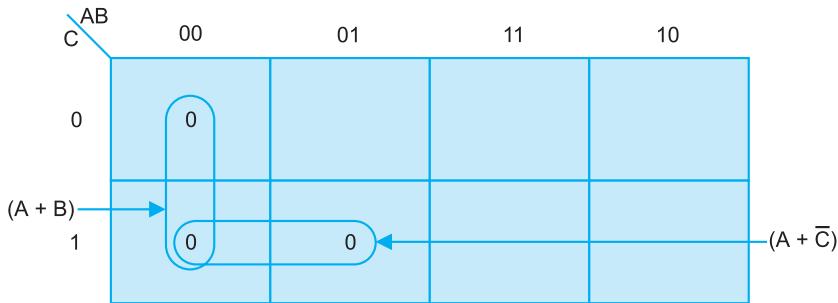
Fig. 4.38 Karnaugh map showing sum terms.

Example 18. Represent the following product of sums expression on Karnaugh map and simplify:

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

or

$$Y(A, B, C) = \pi M(0, 1, 3) \text{ or simply } \pi(0, 1, 3)$$



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

Fig. 4.39 Karnaugh map for Example 18.

Figure 4.39 shows the Karnaugh Map of Example 18. Two groups of adjacent 0-squares have been made as shown in the figure. While simplifying the product of sums expressions by Karnaugh map method, the common variables which have 0-logic are taken as they are, but the variables which have 1-logic are complemented.

The vertical two 0-squares have common variables A and B with 0-logic. There is no common variable from rowside. So their sum ($A + B$) is taken. Again the 2nd enclosure containing two horizontal 0-squares has C common with logic 1, so its complement is taken. From column side A with 0 logic is common and so A is taken. Their sum is ($A + \bar{C}$). The result obtained from these two enclosures is multiplied to obtain the simplified Boolean function. The simplified function is

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

Example 19. Represent the following Boolean functions on Karnaugh map, and simplify it:

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

or $Y(A, B, C, D) = \pi M(6, 7, 15)$

Figure 4.40 shows the Karnaugh map of the above Boolean function. The two groups of adjacent 0-squares have been made as shown in the figure. For the enclosure I column wise, the variable B with 1-logic is common, so it is complemented, i.e. \bar{B} is taken. Row wise CD with 1-logic is common, so $(\bar{C} + \bar{D})$ is to be considered. The sum term for the I enclosure will be $(\bar{B} + \bar{C} + \bar{D})$. For the enclosure II column wise, variables are A with 0-logic and B with 1-logic. So $(A + \bar{B})$ is to be taken. Row wise the common variable is C with 1-logic, so \bar{C} is to be taken. The sum term for II enclosure will be $(A + \bar{B} + \bar{C})$. The simplified Boolean function is given by

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

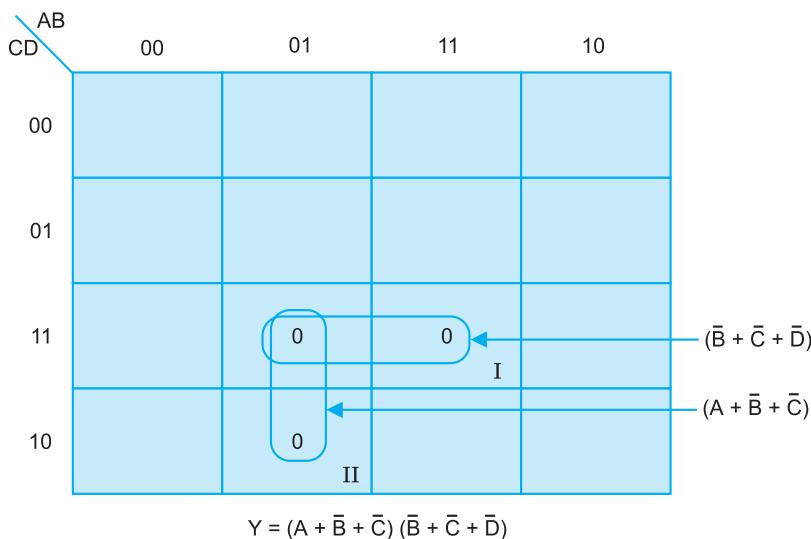


Fig. 4.40 Karnaugh map for Example 19.

4.3.8 Don't-care Conditions

Boolean expressions describe the behaviour of logic networks. Each square of a Karnaugh map represents the response (i.e., output) of the network corresponding to a combination of logic values of the input variables. Sometimes, certain input combinations never occur. In such situations the output of the logic network (i.e., minterms on the Karnaugh map) are not specified. These situations are referred to as don't-care conditions. A don't-care condition is indicated by a dash or cross on a Karnaugh map. The squares on the Karnaugh map with dash entries are known as don't-care squares. A don't-care square may be assumed either "as a 1-square or 0-square as desired while forming the groups of squares for simplification. Any one of such squares or some of them-may be included or may not be included while forming groups.

The Karnaugh map shown in Fig. 4.41(a) with don't care conditions can also be represented as

$$F(A, B, C, D) = \Sigma m(0, 2, 3, 6, 7, 12, 13, 14) + \Sigma d(1, 4, 11, 15)$$

where d denotes don't-care conditions

The above expression is for sum of product expression.

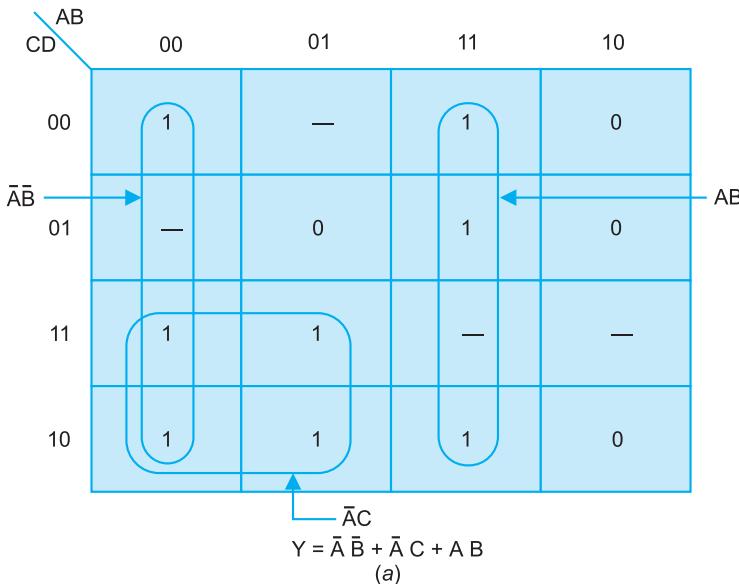
The K-map shown in Fig. 4.41(b) with don't-care conditions for product of sums expression can also be written as

$$F(A, B, C, D) = \pi M(5, 8, 9, 10). \pi d(1, 4, 11, 15).$$

Example 20. Fig. 4.41 shows a Karnaugh map with don't-care conditions. Find Boolean expression (a) in sum of products form, and (b) in product of sums form.

Figure 4.41(a) shows groupings for sum of product expression. The simplified Boolean function is given by

$$Y(A, B, C, D) = \bar{A}\bar{B} + \bar{A}C + AB$$



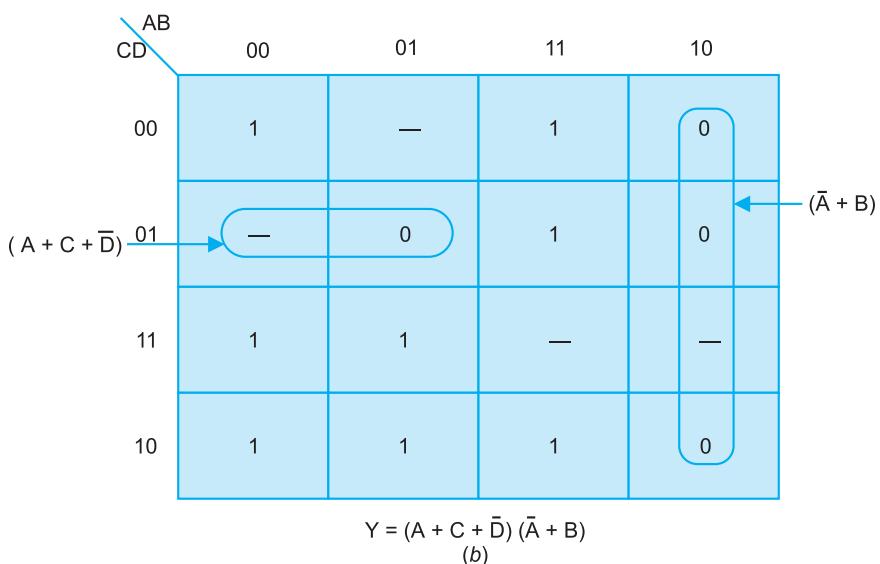


Fig. 4.41 Karnaugh map with Don't care Conditions for Example 20.

Figure 4.41(b) shows groupings for product of sums expression. The simplified Boolean expression will be

$$Y(A, B, C, D) = (A + C + \bar{D})(\bar{A} + B)$$

4.3.9 Quine-McCluskey Method

This is a tabular method of simplification. This is applicable to any number of variables. It is an algorithmic method and programs have been developed to implement it on a computer. The main aim of this book is to describe computer organization. Chapters on digital devices and logic design are background chapters. The interested readers can refer to the books mentioned in references 17, 18, 19 for this method of simplification of Boolean expressions.

4.4 COMBINATIONAL AND SEQUENTIAL CIRCUITS

There are two types of logic circuits—combinational and sequential. A combinational circuit is one in which the state of the output at any instant is entirely determined by the states of the inputs at that time. Combinational circuits are those logic circuits whose operations can be completely described by a truth table/Boolean expression. A combinational circuit is realized using AND, OR and NOT gates (or NAND or NOR gates). Examples of combinational circuits are: adders, subtractors, digital comparators, code converters, decoders, encoders, digital multiplexers/selectors, demultiplexers, parity checker-generators, programmable logic arrays, ROMs, etc.

A sequential circuit consists of a combinational logic and storage elements. The output of a sequential circuit is not only a function of the present inputs but also of past inputs. The state of the storage elements depends upon the preceding inputs and the preceding states of the elements. Therefore, the output of a sequential circuit becomes a function of both the present inputs and the present internal states of the sequential circuit. To realize sequential circuits in addition to AND, OR and NOT gates, flip-flops are also required. Examples of

sequential circuits are: registers, shift-registers, counters, etc. The two major uses of sequential circuits in digital systems are:

1. as memories to store information while processing.
2. as control circuits to generate control signals which are essential to select and enable a sequence of data transfer or data processing steps in the execution of multistep tasks.

The sequential circuits which employ clock are called synchronous sequential circuits. In a synchronous sequential circuit all memory elements are clocked latches or clocked flip-flops. The design and operation of sequential circuits is greatly simplified by the use of clock signals. Hence, the sequential circuits encountered in digital systems are mostly synchronous sequential circuits. The sequential circuits which do not employ clock are known as unclocked or asynchronous sequential circuits. Unclocked sequential circuits are difficult to design and therefore, they are relatively uncommon.

4.5 ARITHMETIC CIRCUITS

Arithmetic operations, such as addition, subtraction, multiplication, division etc. are performed by a digital computer, calculator or other digital system. Logic circuits for some important arithmetic operations are discussed in the following sections.

4.5.1 Half-Adder

A logic-circuit which performs addition of two binary bits is called a half-adder. Table 4.20 is the truth table for the addition of two binary bits.

Table 4.20 Truth Table for Half-Adder

Inputs		Outputs	
A	B	Sum S	Carry C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

From Table 4.20 it is concluded that the sum is equal to A XOR B. It means that the outputs of an EXCLUSIVE-OR gate will give the sum. The carry is equal to A AND B. The output of an AND gate will give the carry. Fig. 4.42(a) shows the logic circuit for a half-adder and Fig. 4.4.2(b) its symbol. The expression for the sum and carry are as follows:

$$\begin{aligned} S &= \bar{A}B + A\bar{B} \\ &= A \oplus B \\ C &= AB \end{aligned}$$

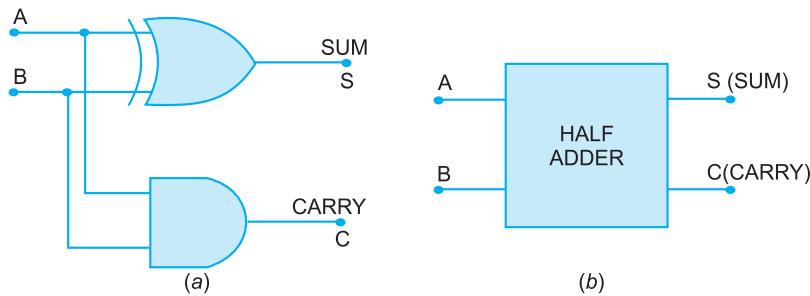


Fig. 4.42 Half adder (a) Logic circuit (b) Symbol.

4.5.2 Full-Adder

When two n -bit numbers are added, there may be a carry from one stage to the next stage. The carry coming out from one stage is to be added to the next stage. A half-adder cannot add 3 bits as it has only 2 input terminals. A logic circuit which can add 3 bits is required. The logic circuit which can add 3 binary bits is known as full-adder. Table 4.21 is the truth table for a full-adder.

Table 4.21. Truth Table for a Full-Adder

INPUTS			OUTPUTS	
A_n	B_n	C_{n-1}	SUM	$CARRY$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A full-adder can be built using two half-adders and an OR gate as shown in Fig. 4.43.

A logic circuit to behave according to the truth table given in Table 4.21 can be realised. The sum of product expression for SUM and CARRY for Truth Table 4.21 can be written as follows:

It will be easy to write logic expression if we assume $A_n = A$, $B_n = B$, $C_{n-1} = C$. We write the product terms when logic value of S_n or C_n is equal to 1 .

$$\begin{aligned}
 \text{SUM } (S_n) &= \overline{\overline{A} \overline{B} C} + \overline{A} \overline{B} \overline{C} + A \overline{B} \overline{C} + ABC \\
 &= \overline{A} (\overline{B} C + B \overline{C}) + A (\overline{B} \overline{C} + BC) \\
 &= \overline{A} (B \oplus C) + A (\overline{B} \oplus C) \\
 &= A \oplus B \oplus C \\
 &= A_n \oplus B_n \oplus C_{n-1}
 \end{aligned}$$

$$\text{CARRY } (C_n) = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

This expression can easily be reduced by Karnaugh map method. It can also be reduced by algebraic method as follows:

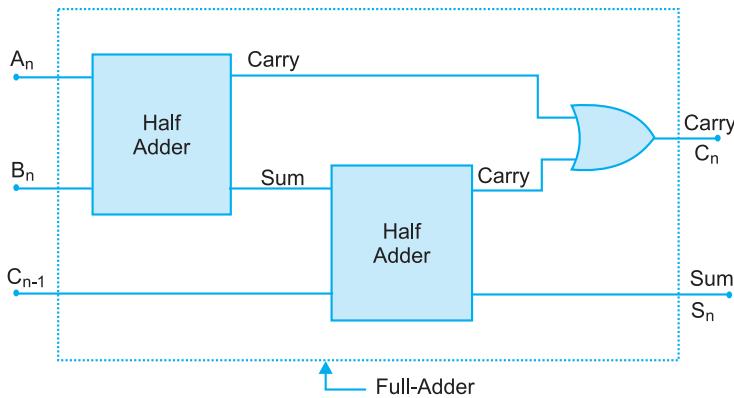


Fig. 4.43 A full-adder realized from two half-adders

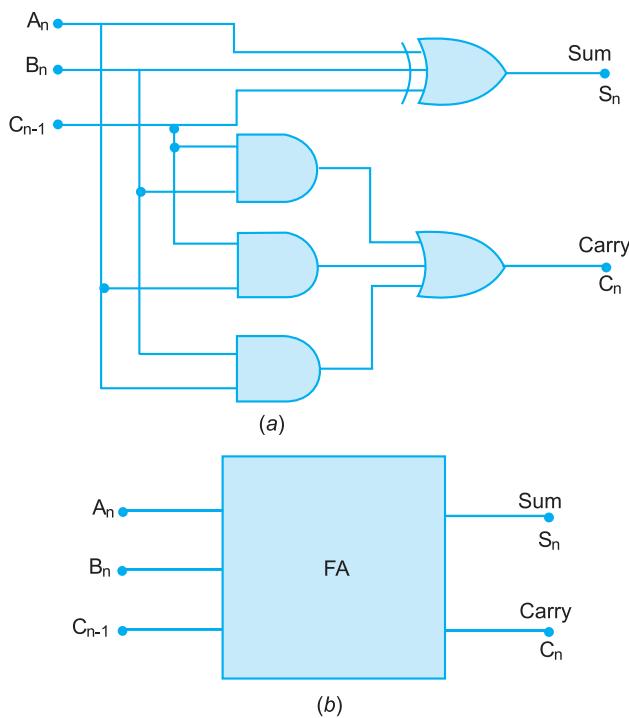


Fig. 4.44 Full-Adder (a) Logic circuit; (b) Symbol

If we add two more ABC terms we have,

$$\begin{aligned} C_n &= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC + ABC + ABC \\ &= \bar{A}BC + ABC + A\bar{B}C + ABC + AB\bar{C} + ABC \end{aligned}$$

$$\begin{aligned}
 &= BC(A + \bar{A}) + AC(B + \bar{B}) + AB(C + \bar{C}) \\
 &= AB + BC + AC \\
 &= A_n B_n + B_n C_{n-1} + A_n C_{n-1}
 \end{aligned}$$

A logic circuit for full-adder can be realized as shown in Fig. 4.44(a). Its symbolic representation is shown in Fig. 4.44(b).

4.5.3 *n*-Bit Adder

To obtain *n*-bit adder *n* number of full-adders are connected as shown in Fig. 4.45. In the least significant stage a half-adder instead of a full-adder can be employed. The sum of *n*-bit adder is of *n* bits as the storing capacity of the register is only of *n* bits. Therefore, the carry out of the last stage indicates overflow.

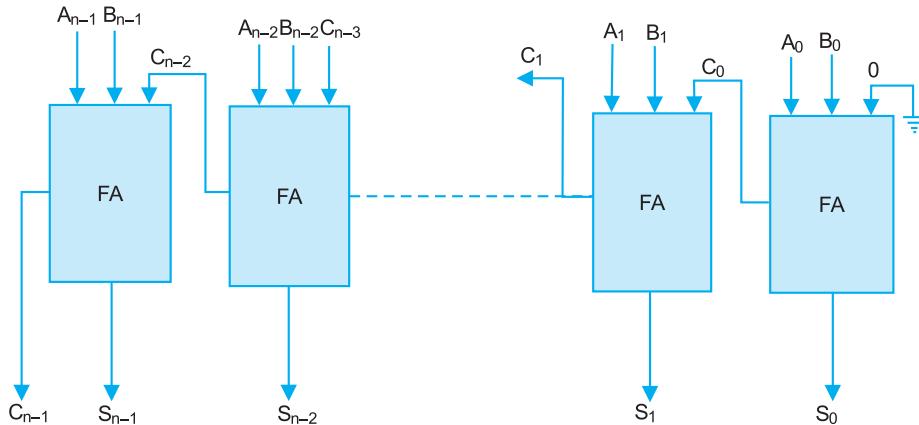


Fig. 4.45 *n*-bit adder.

4.5.4 Adder-Subtractor

In a computer the same circuit performs addition as well as subtraction. The subtraction is performed using 2's complement. An adder-subtractor circuit consists of a controlled inverter and *n* number of full-adders. A controlled inverter is shown in Fig. 4.46. When INVERT terminal is low, the controlled inverter transmits the *n*-bit input to the output. When INVERT signal is high, it transmits the 1's complement of the input. To obtain 2's complement 1 is added to 1's complement.

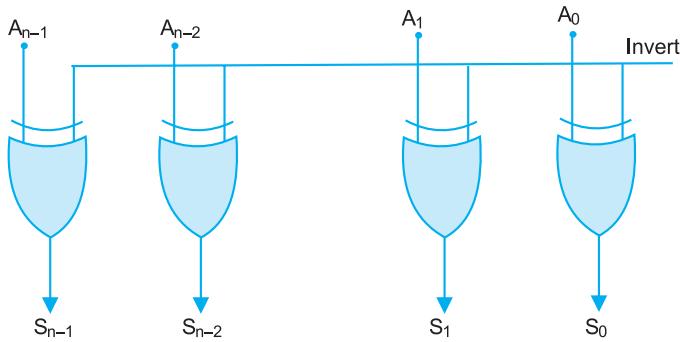


Fig. 4.46. Controlled inverter.

An adder-subtractor circuit is shown in Fig. 4.47. It includes controlled inverter and n number of full adders. For addition A_0 and B_0 are added to give S_0 , A_1 and B_1 are added to give S_1 and so on. So B_0, B_1, \dots, B_n should be applied to full adders. For this, INVERT signal of the controlled inverter portion of the circuit has to be low. In Fig. 4.47, this is marked as SUB. This low signal is also applied to the carry-in terminal of the full-adder of the LSB stage. The full adders perform addition of A_n and B_n . Thus addition is performed when SUB is low.

To perform subtraction SUB signal is made high. So the controlled inverter portion of the circuit inverts B_n . So 1's complement of B_n is obtained. The SUB signal is also applied to the carry-in terminal of the full-adder of the LSB stage. As SUB is high, 1 is added to the sum of A_0 and B_0 . It is a step to make 2's complement of B_n . The full adders add A_n and the 2's complement of B_n . Thus subtraction is performed when SUB is high.

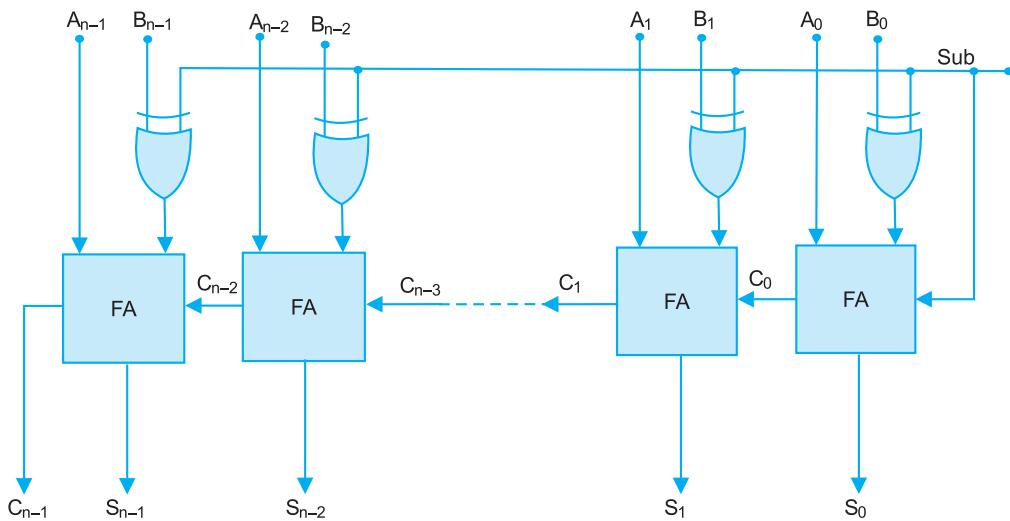


Fig. 4.47 Binary adder-subtractor.

4.5.5 High-Speed Addition and Subtraction

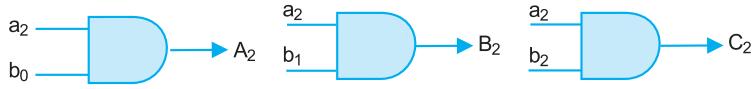
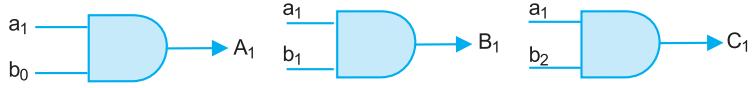
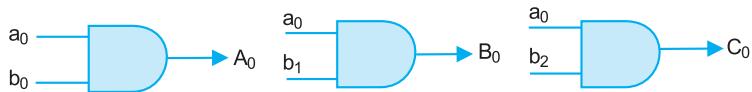
Adders and subtractors described so far have considerable amount of delay because they are basically ripple-carry adders and subtracters. At each stage the carry out of the previous stage is required. So unless the operation of the previous stage is performed the carry bit is not available. To alleviate this delay in determination of the carry bits, each carry bit can be determined directly from A_n and B_n bits of input numbers and some lower order carry. The circuit which generates each carry bit C_n should be able to see beyond the intermediate carry bits and work directly with the inputs A_n, B_n and some lower order carry bit. The adders and subtracters based on such a scheme for generating carry bits are known as look-ahead-carry adders and subtracters. For detailed analysis and design refer to the books in Ref. 12 and 13.

4.5.6. Multiplication of Binary Numbers

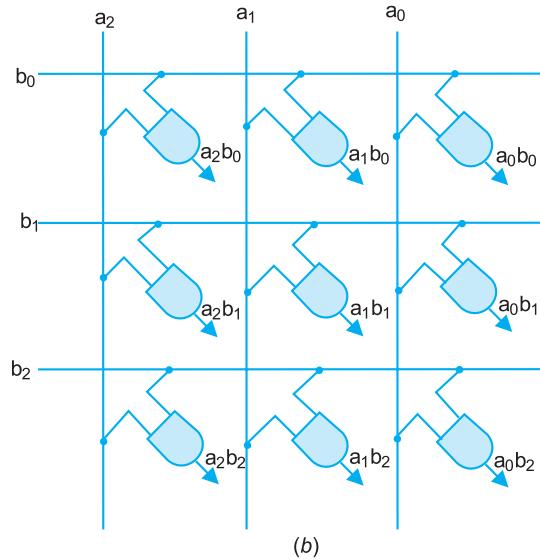
Suppose we want to multiply $a_2\ a_1\ a_0$ by $b_2\ b_1\ b_0$. This is performed as follows:

$$\begin{array}{r}
 & a_2 & a_1 & a_0 \\
 & b_2 & b_1 & b_0 \\
 \hline
 a_2b_0 & a_1b_0 & a_0b_0 \\
 a_2b_1 & a_1b_1 & a_0b_1 \\
 a_2b_2 & a_1b_2 & a_0b_2 \\
 \hline
 P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array}$$

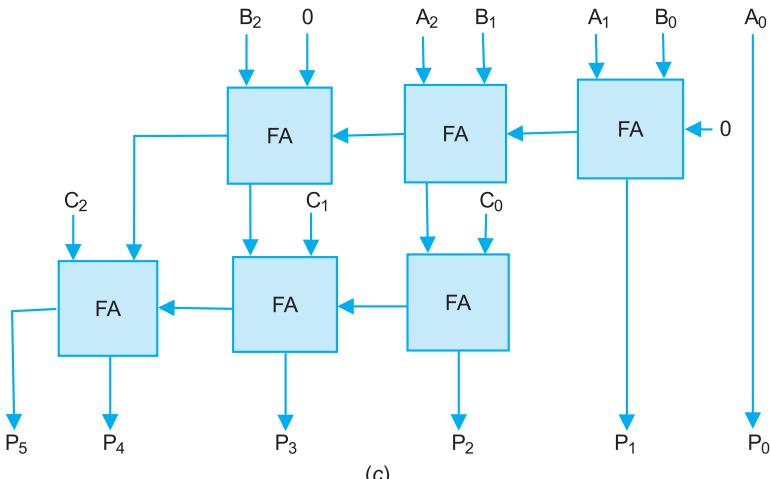
In Fig. 4.4(a) AND gates are shown to obtain the products of $a_i\ b_j$. These gates can be connected as shown in Fig. 4.48(b). The multiplier circuit made of full-adders is shown in Fig. 4.48(c). $P_5\ P_4\ P_3\ P_2\ P_1\ P_0$ is the product of $a_2\ a_1\ a_0$ and $b_2\ b_1\ b_0$. If the number of bits increases in the numbers to be multiplied, the size of the array of full-adders also increases. This will increase the delay through the array. There are improved methods of rearranging the adders to reduce the delay. For details of various types of multiplication and other arithmetic circuits refer to the book in Ref. 12.



(a)



(b)

**Fig. 4.48** A 3-bit multiplier made of full-adders.**PROBLEMS**

1. Explain AND and OR operations with suitable examples of logic statements and electrical switches. Does AND operation perform multiplication? Does OR operation perform addition?
 2. What is the difference between EXCLUSIVE-OR and INCLUSIVE-OR?
 3. Explain Boolean variables, Boolean operations and Boolean expressions.
 4. What are Boolean postulates?
 5. What are truth tables? Explain with suitable examples.
 6. Find the truth tables of the following functions:
 - (a) $X\bar{Y} + \bar{X}Y$
 - (b) $XY + \bar{X}\bar{Y}$
 - (c) $XY + YZ$
- [Ans.** The values in order will be:
- (a) 0, 1, 1, 0;
 - (b) 1, 0, 0, 1 and
 - (c) 0, 0, 0, 1, 0, 0, 1, 1]
7. What are Boolean theorems? Discuss their usefulness.
 8. Show that:
 - (a) $X + XY = X$
 - (b) $X + \bar{X}Y = X + Y$
 - (c) $\bar{X} \cdot \bar{Y} = \bar{X} + \bar{Y}$
 - (d) $(Z + X)(Z + \bar{X} + Y) = (Z + X)(Z + Y)$
 9. Simplify the following logic expressions:
 - (a) $A\bar{B} + \bar{A}\bar{B} + \bar{A}B + AB$

(b) $A\bar{C} + B\bar{C} + \bar{A}\bar{C} + \bar{B}\bar{C}$

(c) $ABC + A\bar{B}C + AB\bar{C} + A\bar{B}\bar{C}$

(d) $A\bar{B}C + \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$

[Ans. (a) 1, (b) \bar{C} , (c) A and (d) \bar{B}]

10. Simplify the following Boolean expressions

(a) $\bar{A}\bar{B}\bar{C} + ABC + \bar{A}\bar{B}C + ABC$

(b) $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$

(c) $\bar{A}\bar{B}C + A\bar{B}C + ABC + \bar{A}\bar{B}C$

(d) $\bar{A}\bar{B}\bar{C} + ABC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}$

[Ans. (a) B, (b) \bar{A} , (c) C and (d) \bar{C}]

11. Simplify the following Boolean functions:

(a) $\bar{A}B + AB + B\bar{C}$

(b) $\bar{A} + B + \bar{B}A + \bar{C}$

(c) $(\bar{A} + B + \bar{C})A\bar{B}C$

(d) $(\bar{A} + B + \bar{B}A)D$

[Ans. (a) B, (b) 1, (c) 0, (d) D and (e) $C\bar{A} + CB$]

12. Simplify the following Boolean functions:

(a) $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C}D + B\bar{C}$

(b) $(\bar{A} + B)AB + A + \bar{A}D$

(c) $(A + B)(A + \bar{B})$

(d) $(A + B + C)(A + B + \bar{C})$

[Ans. (a) $B\bar{C}$, (b) $A + D$, (c) A and (d) $A + B$]

13. Show that:

(a) $XZ + Y\bar{X} + YZ = XZ + Y\bar{X}$

(b) $\bar{X} + XY = \bar{X} + Y$

(c) $Z\bar{X} + ZXY = Z\bar{X} + ZY$

(d) $(XY + \bar{Y}X + Y\bar{Z})\bar{X}\bar{Y} + X\bar{Y} + Y\bar{Z} = X\bar{Y} + Y\bar{Z}$

(e) $X + \bar{X}Y = (X + \bar{X})(X + Y)$

14. What is the dual of a Boolean expression? What is the difference between dual and complement?

15. Find the dual of the following logic expressions:

(a) $1.X.Y + \bar{Y}.X.Z + 0$

(b) $X.Y(Y + Z + X) + X.\bar{Y}$

(c) $X\bar{Y} + XYZ + Y\bar{Z}$

- (d) $(X + Y + Z)(\bar{X} + \bar{Y} + Z)(X + \bar{Y} + \bar{Z})$
 (e) $XY(Y + \bar{Y}Z) + \bar{X}Z$

- [Ans.] (a) $(0 + X + Y)(\bar{Y} + X + Z).1$
 (b) $(X + Y + XYZ)(X + \bar{Y})$
 (c) $(X + \bar{Y})(X + Y + Z)(Y + \bar{Z})$
 (d) $XYZ + \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z}$
 (e) $[X + Y + Y(\bar{Y} + Z)](\bar{X} + Z)$

16. Find the dual of the following expressions:

- (a) $(X\bar{Y} + YZ + XYZ)XYZ$
 (b) $(\bar{X}Z + YZ)(\bar{Z}Y + XY)$
 (c) $XY(\bar{Y}X + YZ) + Y\bar{Z} + XY$
 (d) $(X + Y)(\bar{Y}X + YZ) + Y\bar{Z}$

- [Ans.] (a) $(X + \bar{Y})(Y + Z)(X + \bar{Y} + Z) + X + Y + Z$
 (b) $(\bar{X} + Z)(Y + Z) + (\bar{Z} + Y)(X + Y)$
 (c) $[X + Y + (\bar{Y} + X)(Y + Z)](Y + \bar{Z})(X + Y)$
 (d) $[XY + (\bar{Y} + X)(Y + Z)](Y + \bar{Z})$

17. Find the complement of the following logic expressions:

- (a) $X.\bar{Y} + \bar{X}.Z$
 (b) $1.\bar{X} + Y.\bar{Z} + 0$
 (c) $X\bar{Y}(Y + \bar{Z}) + \bar{X}Y$
 (d) $X(\bar{Y} + Y\bar{Z}) + Y\bar{Z}$

- [Ans.] (a) $(\bar{X} + Y)(X + \bar{Z})$
 (b) $(0 + X)(\bar{Y} + Z).1$
 (c) $(\bar{X} + Y + \bar{Y}Z)(X + \bar{Y})$
 (d) $[\bar{X} + Y.(\bar{Y} + Z)](\bar{Y} + Z)$

18. What are (a) sum of products form, and (b) product of sums form of logic expressions? Explain with suitable examples.
 19. What is the canonical form of logic expressions? Explain minterms and maxterms.
 20. Explain (a) minterm canonical form or standard sum of products, and (b) maxterm canonical form or standard product of sums.
 21. Convert the following logic expressions into canonical form:

- (a) $A + \bar{A}B$
 (b) $A + \bar{B}C$
 (c) $AB + BC$
 (d) $A\bar{B} + ABC + BCD$

- [Ans.] (a) $AB + A\bar{B} + \bar{A}B$
 (b) $ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC$
 (c) $ABC + ABC\bar{C} + \bar{A}BC$
 (d) $ABCD + A\bar{B}CD + ABCD + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}CD + AB\bar{C}\bar{D}$

22. Convert the following logic expressions into canonical form:

$$(a) (A + \bar{B})(\bar{B} + C)$$

$$(b) B(A + \bar{C})$$

$$(c) (A + \bar{B})(A + B + \bar{D})$$

$$(d) A(A + B + C)$$

[Ans. (a) $(A + \bar{B} + C)(A + \bar{B} + \bar{C})(A + \bar{B} + C)(\bar{A} + \bar{B} + C)$

(b) $(A + B + C)(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + B + \bar{C})(A + \bar{B} + C)(\bar{A} + \bar{B} + C)$

(c) $(A + \bar{B} + C + D)(A + \bar{B} + C + \bar{D})(A + \bar{B} + \bar{C} + D)(A + \bar{B} + \bar{C} + \bar{D})(A + B + C + \bar{D})(A + \bar{B} + \bar{C} + \bar{D})$

$(A + B + C + \bar{D})(A + \bar{B} + \bar{C} + \bar{D})$

(d) $(A + B + C)(A + B + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C})$

23. Realize (a) an AND-OR logic network and (b) NAND network for the following Boolean function:

$$Z = BC + AD$$

[Ans. See (a) Fig. 4.49, (b) Fig. 4.50.]

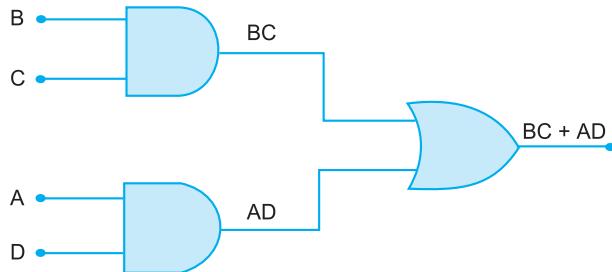


Fig. 4.49 AND-OR Network for problem 23(a).

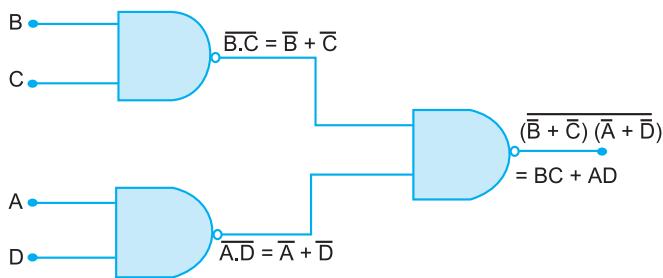


Fig. 4.50 NAND Network for problem 23(b).

24. Realize (a) an AND-OR network and (b) NAND network for the following Boolean function:

$$Z = XYZ + \bar{X}YZ + Y\bar{Z}$$

[Ans. See (a) Fig. 4.51 and (b) Fig. 4.52]

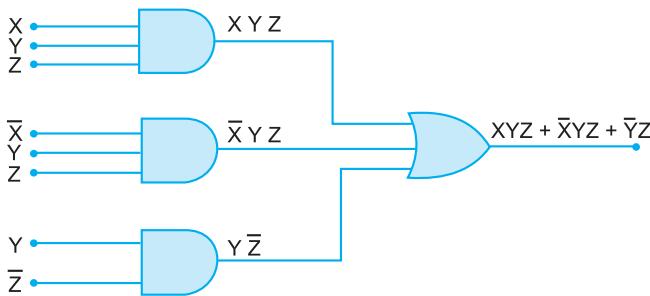


Fig. 4.51 AND-OR Network for problem 24(a).

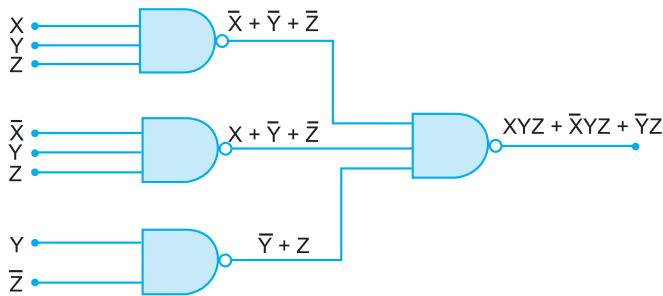


Fig. 4.52 NAND Network for problem 24(b).

25. Realize (a) an OR-AND network and (b) NOR network for the following Boolean function:

$$Y = (A + B)(C + D)$$

[Ans. See (a) Fig. 4.53 and (b) Fig. 4.54]

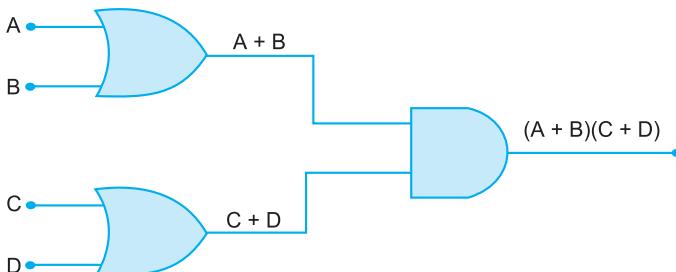


Fig. 4.53 OR-AND Network for problem 25(a).

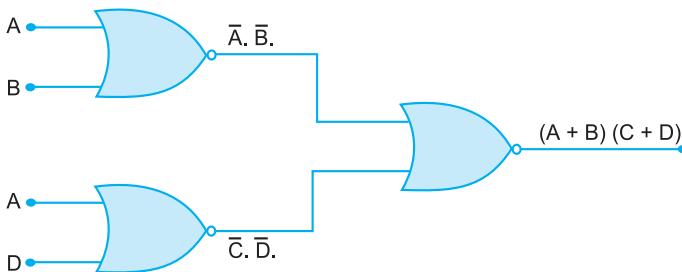


Fig. 4.54 NOR Network for problem 25(b).

26. Simplify the following expressions by Karnaugh map method:

- (a) $\overline{ABC} + ABC + \overline{ABC} + ABC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C}$
- (b) $ABC + ABC + A\overline{B}\overline{C} + \overline{ABC} + \overline{A}\overline{B}\overline{C} + \overline{ABC}$
- (c) $\overline{ABC} + \overline{ABC} + \overline{ABC}$
- (d) $\overline{ABC} + ABC + A\overline{B}\overline{C}$

[Ans. (a) $B + \overline{C}$, (b) $A + C$, (c) $\overline{AB} + \overline{AC}$ and (d) $B\overline{C} + A\overline{C}$]

27. Draw Karnaugh map and simplify the following Boolean expressions:

- (a) $AB\overline{CD} + A\overline{B}\overline{C}\overline{D} + AB\overline{CD} + A\overline{B}\overline{CD}$
- (b) $\overline{A}\overline{B}\overline{CD} + A\overline{B}\overline{CD} + \overline{ABC}D + \overline{ABC}\overline{D}$
- (c) $\overline{ABC}\overline{D} + A\overline{B}\overline{CD} + \overline{ABC}D + \overline{ABC}\overline{D} + \overline{ABCD}$
- (d) $\overline{ABC}\overline{D} + \overline{ABCD} + A\overline{B}\overline{CD} + A\overline{BCD} + A\overline{B}\overline{D} + ABC\overline{D}$

[Ans. (a) $A\overline{C}$, (b) $\overline{A}\overline{B}$, (c) $\overline{A}\overline{B} + \overline{AC}D$ and (d) $\overline{BD} + ABD$]

28. How are the squares of a Karnaugh map labelled? How are the sum of product form of logical expressions are written in $\Sigma(N_1, N_2, \dots)$ form?

29. Draw Karnaugh map and simplify the following Boolean expressions:

- (a) $Y(A, B, C) = \Sigma(2, 3, 4, 6, 7)$
- (b) $Y(A, B, C) = \Sigma(0, 1, 3, 4, 5)$
- (c) $Y(A, B, C, D) = \Sigma(3, 4, 5, 6, 7, 11, 12, 13, 14, 15)$
- (d) $Y(A, B, C, D) = \Sigma(0, 2, 4, 6, 8, 10, 12, 14, 15)$

[Ans. (a) $B + A\overline{C}$, (b) $\overline{B} + \overline{AC}$, (c) $B + CD$ and (d) $\overline{D} + ABC$]

30. Explain the following terms:

- (a) Prime implicant
- (b) Nonprime implicant
- (c) Essential prime implicant
- (d) Nonessential prime implicant

31. Draw Karnaugh map and simplify the following Boolean functions:

- (a) $Y(A, B, C, D) = \Sigma m(0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)$
- (b) $Y(A, B, C, D) = \Sigma m(0, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 14 \text{ and } 15)$

[Ans. (a) $A + D + \overline{AC} + \overline{AB}$]

Hint: make the enclosures as follows:

Enclosure I: 8, 9, 10, 11, 12, 13, 14 and 15.

Enclosure II: 1, 3, 5, 7, 9, 11, 13 and 15.

Enclosure III: 2, 3, 6 and 7.

Enclosure IV: 0, 1, 2 and 3.

There are other alternatives also.

[Ans. (b) $\overline{B} + A + \overline{AD}$]

Hint: make the enclosures as follows:

Enclosure I: 0, 1, 2, 3, 8, 9, 10 and 11.

Enclosure II: 8, 9, 10, 11, 12, 13, 14 and 15.

Enclosure III: 0, 2, 4 and 6.

32. Draw Karnaugh map and simplify the Boolean function:

$$Y(A, B, C, D) = \pi M(1, 5, 7, 8, 9, 10, 11, 12, 13, 14, 15)$$

[Ans. $\bar{A}(\bar{B} + \bar{D})(C + \bar{D})$]

Hint: make the following enclosures:

Enclosure I: 8, 9, 10, 11, 12, 13, 14 and 15.

Enclosure II: 5, 7, 13 and 15.

Enclosure III: 1, 5, 9, 13.

33. Explain what you understand by don't care conditions.

34. Find the simplified Boolean function (a) in sum of products form and (b) in product of sums form for the Karnaugh map represented by

$$Y(A, B, C, D) = \Sigma m(0, 3, 7, 8, 9, 11, 12, 13) + \Sigma d(1, 4, 14, 15)$$

$$\text{or } \pi M(2, 5, 6, 10) + \pi d(1, 4, 14, 15)$$

[Ans. (a) $\bar{C}\bar{D} + CD + A\bar{C}$]

Hint: make groupings of squares containing 1s as follows:

Group I : 0, 4, 8 and 12, assume 4 to contain 1.

Group II : 3, 7, 11 and 15, assume 15 to contain 1.

Group III : 8, 9, 12 and 13.

[Ans. (b) $(A + \bar{B} + C)(\bar{C} + D)$]

Hint: make groupings of squares containing 0s as follows:

Group I: 2, 6, 10 and 14.

Group II: 4 and 5.

35. What is a half-adder? Write truth table for a half-adder and develop its logic circuit.

36. What is a full-adder? How is a full-adder built using half-adders?

37. Write the truth table for a full-adder and develop its logic network.

38. How is an n -bit adder realized using n full-adders? Draw the logic network.

39. How is an adder-subtractor network built using full-adders? Draw its logic network. What is the function of a controlled inverter?

40. How is multiplication performed using arrays of full-adders? Explain with a suitable example.

**THIS PAGE
IS BLANK**

5

CHAPTER

CENTRAL PROCESSING UNIT (CPU)

5.1 INTRODUCTION

The central processing unit (CPU) is the brain of a computer. Its primary function is to execute programs. The program, which is to be executed, is stored in the main memory. A program is a sequence of instructions to perform a specified task. The CPU fetches instruction codes from the memory and decodes them. The necessary data for the execution of an instruction may be in registers of the CPU or in the memory. The CPU also reads data from the memory, which are required for instruction execution. When the required data for the execution of an instruction is at hand, the CPU executes the instruction. When all the instructions of a program are executed, the CPU gives the result. The result is placed in the memory or sent to an output device according to the instruction given in the program.

Besides executing programs, the CPU also controls input devices, output devices and other components of the computer. It controls input and output devices to receive and send data. Under its control programs and data are stored in the memory and displayed on the CRT screen.

The CPU of a small computer is a microprocessor. The CPU of a large computer contains a number of microprocessors. They work in parallel to execute a program. This makes computer more powerful and faster. Each microprocessor in a large central processing unit performs a specified task.

5.2 CPU ORGANIZATION

The following are the main sections of a CPU:

- (i) Arithmetic and logical unit (ALU).
- (ii) Control unit.
- (iii) Accumulator and general and special purpose registers.

5.2.1 Arithmetic and Logic Unit (ALU)

The function of an arithmetic and logic unit is to perform arithmetic and logic operations. Usually an ALU performs the following arithmetic and logic operations:

- (i) Addition
- (ii) Subtraction
- (iii) Multiplication
- (iv) Division

- (v) Logical AND
- (vi) Logical OR
- (vii) Logical EXCLUSIVE-OR
- (viii) Complement (logical NOT)
- (ix) Increment (*i.e.*, addition of 1)
- (x) Decrement (*i.e.*, subtraction of 1)
- (xi) Left or right shift (the content of the accumulator can be shifted left or right by one bit)
- (xii) Clear (the content of the accumulator or carry flag can be made zero).

Other mathematical operations such as exponential, logarithmic, trigonometric and floating-point operations are not performed by ALU. These operations are performed by special purpose math processor called floating-point unit (FPU). Modern microprocessors contain an FPU on the microprocessor chip itself (*i.e.*, an on-chip FPU). Some cheaper microprocessors which are used for simple automatic control applications may not contain on-chip FPU. Such processors use either software for above mentioned mathematical operations or employ a math processors IC (or math coprocessor) in the microprocessor-based system. The use of software for such mathematical operations makes execution slower. Math processors speed up program execution and reduce programming complexity. The choice depends on actual requirement and cost involved in a particular application.

5.2.2 Control Unit

The control unit of a CPU controls the entire operation of the computer. This very section of the CPU really acts as the brain of the computer. It also controls all other devices such as memory, input and output devices connected to the CPU. It fetches instruction from the memory, decodes the instruction, interprets the instruction to know what tasks are to be performed and sends suitable control signals to other components to perform further necessary steps to execute the instruction. It maintains order and directs the operation of the entire system. It gives order to ALU what operations are to be performed. It generates timing and control signals, and provides them for all operations. It controls the data flow between CPU and peripherals (including memory). It provides status, control and timing signals that the memory and I/O devices require.

Under the control of the control unit the instructions are fetched from the memory one after another for execution until all the instructions are executed. In a Von Neumann type processor such as Intel 8085, for fetching and executing an instruction the following steps are performed under its control:

- (i) The address of the memory location where instruction lies, is placed on the address bus.
- (ii) Instruction is read from the memory.
- (iii) The instruction is sent to the decoding circuitry for decoding.
- (iv) Addresses and data required for the execution of the instruction are read from the memory.
- (v) These data/addresses are sent to the other section for processing.
- (vi) The results are sent to the memory or kept in some register.

- (vii) Necessary steps are taken to fetch next instruction. For this the content of the program counter is incremented.

5.2.3 Registers

A CPU contains a number of registers to store data temporarily during the execution of a program. The number of registers differs from processor to processor. Some processor contains more registers, some less. Registers are classified as follows:

General Purpose Registers. These registers store data and intermediate results during the execution of a program. They are accessible to users through instructions if the users are working in assembly language.

Accumulator. It is the most important general-purpose register having multiple functions. It is most efficient in data movement, arithmetic and logical operation. Its functions depend on the design of the processor whether the processor is an accumulator-based processor or general-purpose registers-based processor (Refer to Section 5.3.3 for more details). It has some special features that the other general-purpose registers do not have. In a accumulator-based processor design, the accumulator holds one of the operands of arithmetic and logical operation. After the execution of arithmetic and logical instructions, the result is placed in the accumulator. All data transfer between the CPU and device/port are performed through the accumulator.

Special Purpose Registers. A CPU contains a number of special purpose registers for different purposes. These are:

- (i) Program Counter (PC)
- (ii) Stack Pointer (SP)
- (iii) Status Register
- (iv) Instruction Register (IR)
- (v) Index Register
- (vi) Memory Address Register (MAR)
- (vii) Memory Buffer Register (MBR) or Data Register (DR)

All CPUs do not contain all of these special registers. A powerful CPU contains most of them. The brief description of these registers is given below:

Program Counter (PC). The Program counter keeps track of the address of the instruction which is to be executed next. So it holds the address of the memory location, which contains the next instruction to be fetched from the memory. Its content is automatically incremented after an instruction has been fetched assuming that instructions are normally executed sequentially. In case of a jump instruction its contents are modified, and program jumps to the memory location which contains the desired instruction to be executed next. The processors which employ Von Neumann architecture contain a program counter. The processors which use data flow architecture do not contain program counter (Refer to Sections 1.19 and 5.18.1 for more details).

Stack Pointer (SP). The stack is a sequence of memory location defined by the user. It is used to save the contents of a register if it is required during the execution of a program. The stack pointer holds the address of the last occupied memory location of the stack. Thus it indicates upto what memory locations the stack is already filled up. In other words it indicates from which memory location onward the stack is vacant for further storage. The stack will be discussed in detail in Section 8.4.

Status Register (or Flag Register). A flag register contains a number of flags either to indicate certain conditions arising after arithmetic and logical operations or to control certain operations. The flags which indicate conditions are known as condition flags. The flags which are used to control certain operations are called control flags. A flag is a flip-flop which is set or reset by the processor to indicate certain conditions or set/reset by the programmer to control certain operations.

A simple microprocessor like Intel 8085 contains the following condition flags:

- | | |
|-------------|---|
| Carry flag | - It indicates whether there is carry or not after an arithmetic or logical operation. |
| Zero flag | - It indicates whether the result of an arithmetic or logical operation is zero or nonzero. |
| Sign flag | - It indicates whether the result is positive or negative. |
| Parity flag | - It indicates whether the result contains odd number of 1s or even number of 1s. |

Auxiliary carry (or Half carry) *i.e.*, carry from the 3rd bit to 4th is also indicated.

The Intel 8085 does not contain any control flag.

The Intel 8086 microprocessor contains six condition flags and three control flags (Refer to Section 5.13.2 for details)

Instruction Register: It holds an instruction until it is decoded. Some computers have two instruction registers, and so they can fetch and save the next instruction while the execution of the previous instruction is going on.

Index Register: Index registers are used for addressing. One or more registers are designated as index registers. The address of an operand is the sum of the contents of the index register and a constant. Instructions involving index register contain constant. This constant is added to the contents of index register to form the effective address, *i.e.* the address of the operand. Usually, special instructions are provided to increment or decrement an index register so as to access the entries of a table in a successive manner. Alternatively, some computers have the facility of auto-indexing by which the index register is automatically incremented or decremented.

Memory Address Register (MAR). It holds the address of the instruction or data to be fetched from the memory. The CPU transfers the address of the next instruction from the program counter (PC) to the memory address register. From MAR it is sent to the memory through the address bus. Sometimes it is called simply address register (AR).

Memory Buffer Register (MBR) or Data Register (DR). It holds the instruction code or data received from or sent to the memory. It is connected to data bus. The data, which are written into the memory, are held in this register until the write operation is completed. It is also called data register (DR). Thus the flow of data from the CPU to the memory or from the memory to CPU is always through MBR. It is within CPU.

5.3 INSTRUCTION

An instruction is a command given to the computer to perform a specified operation on given data. Each instruction consists of two parts: an opcode (operation code) and an operand. The first part of an instruction, which specifies the operation to be performed, is known as

opcode. The second part of an instruction called operand is the data on which computer performs the specified operation. Important instructions of Intel 8085 are given in section 5.3.1 and those of Intel 8086 in section 5.13.9.

5.3.1 Intel 8085 Instructions

Intel 8085 is an 8-bit microprocessor, very popular in India. Microprocessor kits based on Intel 8085 are widely used in laboratories for students' training. Intel 8085 has one 8-bit accumulator designated as A. It has six 8-bit general-purpose registers: B, C, D, E, H and L. Two 8-bit registers can be combined to form a register-pair, which can handle 16-bit data. The valid register pairs of Intel 8085 are: B-C, D-E and H-L. Special purpose register are: one 16-bit program counter (PC), one 16-bit stack pointer (SP) and an instruction register. It does not have a status register, but contains a set of flip-flops to store status flags. The combination of the binary bits, which indicate status flags is called *program status word* (PSW). Five bits of PSW show status flags, 3-bits are undefined. PSW and the accumulator are treated as a 16-bit unit for stack operation. In addition to these registers it also contains a temporary register, address buffer and data buffer.

Some of instructions of 8085 are described here as they are required to explain a few topics in the subsequent sections.

MOV r_1, r_2 . The contents of register r_2 are transferred to register r_1 . For example, the instruction MOV A, B transfers the contents of register B to register A. The instruction MOV B, A transfers the contents of register A to register B.

MOV r, M . The contents of the memory location M whose address is in H-L pair, are transferred to the register r . For example, MOV A, M will transfer the contents of the memory location M specified by H-L pair, to the accumulator.

MVI r, data . The data specified in the instruction will be transferred to the register r . For example, MVI B, 08 will transfer 08 to register B.

LXI rp, data 16-bit. 16-bit data specified in the instruction will be transferred to the register pair rp . For example, the instruction LXI H, 2500H transfers 2500 to H-L pair.

LDA addr . The contents of the memory address specified in the instruction are transferred to the accumulator. For example, the instruction LDA 2200 will transfer the contents of the memory location 2200 to the accumulator.

STA addr . The contents of the accumulator are transferred to the memory location whose address is specified in the instruction. For example, the instruction STA 2000 will transfer the contents of the accumulator to the memory location 2000.

ADD r . The contents of the register r are added to the contents of the accumulator and the result is placed in the accumulator. For example, the instruction ADD C will add the contents of the register C to the contents of the accumulator, and the result is placed in the accumulator.

ADD M . The contents of the memory location, whose address is in H-L pair, are added to the contents of the accumulator and the result is placed in the accumulator. For example, ADD M will add the contents of the memory location specified by H-L pair, to the contents of the accumulator, and the result is placed in the accumulator.

ADI, data . The data in the instruction are added to the contents of the accumulator, and the result is placed in the accumulator. For example, ADI, 05 will add 05 to the contents of the accumulator, and the result is placed in the accumulator.

SUB r. The contents of the register *r* are subtracted from the contents of the accumulator, and the result is placed in the accumulator.

RAL. The contents of the accumulator are rotated left one bit through carry.

IN port address. The instruction transfers data from the input device or input port to the accumulator.

OUT port address. This instruction transfers the contents of the accumulator to the output device or output port.

HLT. This instruction is halt. After the execution of this instruction the microprocessor stops.

5.3.2 Intel 8085 Instruction Format or Instruction Word Size

As a computer understands instructions only in the form of 0 and 1, instructions and data are fed into the computer in binary form. They are written in binary codes known as *machine codes*. For the convenience of the user the codes can be written in hexadecimal form.

As there are different ways of specifying data the binary codes of all instructions are not of the same length. Instructions are classified into the following three types according to their word length (*i.e.*, length of the binary code):

- (i) Single-byte instruction
- (ii) Two-byte instruction
- (iii) Three-byte instruction

The first byte of an instruction is the opcode of the instruction. Other bytes may be either data or address.

Single-Byte Instruction. In a single-byte instruction, the binary code of the instruction is of one byte. For example, MOV A,B; ADD B; RAL etc. are single-byte instructions. The machine codes for these instructions are: 78H, 80H and 17H respectively. H denotes hexadecimal.

Two-Byte Instruction. In a two-byte instruction, the machine code is of two bytes. For example, MVI A, 05 is a two-byte instruction. Its machine code is 3E, 05. The first byte, *i.e.*, 3E is the opcode of the MVI A instruction, 05 is the data which is to be transferred to register A. As a memory location stores only one byte, the two bytes are stored in two consecutive memory locations.

Another example of a two-byte instruction is IN 01. In coded form it is written as DB, 01. DB is the operation code. The second byte 01 is the address of an input device.

Three-Byte Instruction. In a three-byte instruction the first byte is the opcode of the instruction. The 2nd and 3rd bytes are either data or address. Some examples are:

- (i) LXI H, 2500H : Load H-L pair with 16-bit data 2500H.
21,00,25 : The instruction LXI H, 2500 in code form.

The first byte 21 is the opcode of the instruction LXI H. The second byte 00 is 8 LSBs of the data 2500H. 8 LSBs are loaded into register L. The third byte 25 is 8 MSBs of the data. It is loaded into register H.

- (ii) LDA 2400H : Load accumulator with the contents of the memory location 2400H.
3A, 00,24 : The instruction LDA 2400H in the code form.

The first byte 3A is the opcode of the instruction LDA. The second byte 00 is 8 LSBs of memory address 2400H. The third byte 24 is the 8 MSBs of memory address 2400H.

5.3.3 Alternative Way of Classification of Instructions

Instructions can also be classified depending upon the number of operand addresses they contain. Such a classification is as follows:

- (1) 0-address instruction
- (2) 1-address instruction
- (3) 2-address instruction
- (4) 3-address instruction

Shorter instructions are faster and they can reduce the overall cost.

0-Address Instruction

The 0-address type instructions do not contain any operand address. The operand addresses are implied.

1-Address Instruction

In 1-address instructions only one operand address is specified in the instruction. The other operand address is implied. The other operand is in the accumulator. The result is placed in the accumulator.

2-Address Instruction

In 2-address instructions both operand addresses are specified. The result is placed in one of the specified addresses.

3-Address Instruction

In 3-address instructions two addresses are specified for the two operands and one address for the result.

5.3.4 Classification of Processors/Computers

Faster arithmetic, logical and comparison instructions make a processor faster and more powerful. Two or three address machines are faster and more powerful as compared to zero-address and one-address machines. Depending on the number of addresses given in arithmetic, comparison and logical instructions, processors (or computers) are classified as follows:

- (1) One-Address Machine or Accumulator- Based machines
- (2) Two or Three-Address Machines or General-Purpose Registers Based Machines.
- (3) 0-Address Machines or Stack Machines.

One-Address Machines or Accumulator-Based Machines

Arithmetic, logical and comparison instructions of accumulator-based machines contain the address of only one operand. It is implied that the other operand is in the accumulator. The result of the operation is placed in the accumulator. It is the design feature of the processor. The first-generation computers were accumulator-based machines. Also, the first generation of minicomputers (DEC's PDP-8, HP 2116, etc) and the first generation of 8-bit microprocessors (MC 6800, MC 6809, Intel 8080, Intel 8085, etc) had accumulator-based architecture.

Two or Three-Address or General-Purpose Registers Based Machines

Machines of this type contain a set of general-purpose registers which also act as accumulator for arithmetic, logical and comparison operations. Their instructions contain two or three addresses. In case of two addresses processors the addresses for both operands are specified in the instruction. The result is placed in one of the specified addresses. This type of architecture was designed for the second generation of computers and microprocessors such as IBM 360/370, DEC's PDP-11, Intel 8086, Z8000, Motorola, 68000, etc. Examples of three address machines are : Cyber-170, CDC 6600, etc. Two or three-address machines are faster and more powerful than accumulator-based machines.

0-Address or Stack Machines

Stack oriented machines do not contain any accumulator or general-purpose registers. All operations except data transfer are performed on the operands which are at or near the top of the stack. A stack-oriented machine contains only a stack-pointer (SP), which points to the stacktop. An arithmetic instruction ADD does not require any address. The ADD instruction pops two operands from the stack, adds them and pushes the result back into the stack. But the data transfer instructions of a stack-oriented machine contain operand address. For example, the LOAD and STORE instructions have an operand address. The LOAD X instruction fetches operand from the memory location X, and pushes it into the stack. Similarly, STORE X instruction pops operand from the stack, and stores in the memory location X.

In stack machines arithmetic, logical and comparison instructions do not require any operand address. Hence, they are called 0-address machines. The computing power of a machine depends on the performance of arithmetic, logical and comparison instruction. These instructions are the deciding factors, not the data transfer and some other types of instructions. Examples of stack-oriented machines are : Burrough's B5000, 5500 and 6700 (mainframe computers), and HP 3000 (minicomputers). Many calculators employed stack-oriented processors. This type of architecture is not suitable for current RISC and CISC processors, which have intensive parallelism. Modern processors contain several operands in large register set for high performance.

5.4 ADDRESSING MODES

Each instruction needs data on which it has to perform the specified operation. The operand (data) may be in the accumulator, general purpose register or in some specified memory location. Therefore, there are various ways to specify data. The techniques of specifying the address of the data are known as addressing modes. The important addressing modes are as follows:

- (i) Direct (or absolute) addressing
- (ii) Register addressing
- (iii) Register indirect addressing
- (iv) Immediate addressing
- (v) Implicit addressing
- (vi) Indexed addressing
- (vii) Based addressing
- (viii) Based indexed addressing

- (ix) Relative addressing
- (x) Relative indexed addressing
- (xi) Page addressing
- (xii) Stack addressing

Intel 8085 uses addressing modes only from (i) to (v). In addition to (i) to (v) addressing modes other addressing modes are also used by other microprocessors. Addressing modes of Intel 8086 with examples are given in Section 5.13.10. Examples of addressing modes given in Sections 5.4.1 to 5.4.5. are for Intel 8085.

5.4.1 Direct (or Absolute) Addressing

In direct addressing the address of the data (operand) is specified within the instruction itself. Examples of direct addressing are:

(i) STA 2500H : Store the contents of the accumulator in the memory location 2500H.

(ii) LDA 2500H : Load accumulator with the contents of the memory location 2500H.

In example (i) above, 2500H is the memory address where data are to be stored. The memory address 2500H is given in the instruction itself. In this case it is understood that the source of data is accumulator.

(iii) IN 01 : Read data from an input device whose address is 01.

In this instruction 01 is the address of an input device from where data are to be read. In this instruction it is implied that the data will be transferred from the input device to the accumulator.

5.4.2 Register Addressing

In register addressing the operands are located in general-purpose registers. In other words the contents of a register is the operand. Therefore, only the names of the registers are to be specified in the instruction. Examples of register addressing are:

(i) MOV A, B : Transfer the contents of register B to register A.

The opcode of this instruction is 78H. In addition to the operation to be performed, the opcode also specifies the addresses of the registers mentioned in the instruction. The opcode 78H in binary form is 01111000. The first two bits 01 denote MOV operation, the next three bits 111 are the binary code of register A and the last three bits 000 are the binary code of register B of Intel 8085.

(ii) ADD B : Add the contents of the register B to the contents of the accumulator.

The opcode of this instruction is 80H. In the binary form the code is 10000000. The first five bits 10000 specify the operation to be performed, i.e., ADD. The last three bits 000 are the binary code of register B.

5.4.3 Register Indirect Addressing

In register indirect addressing the address of the operand is given indirectly. The contents of a register or a register-pair are the address of the operand. Compare this with the register addressing where the content of the register is the operand. Examples are:

(i) LXI H, 2400H : Load H-L pair with 2400H.

MOV A, M : Move the content of the memory location, whose address is in H-L pair (*i.e.*, 2400H), to the accumulator.

In this example MOV A, M is an example of register indirect addressing. For MOV A, M instruction the operand is in a memory location whose address is not directly given in this instruction. The address of the memory location is stored in H-L pair, which has been specified by the earlier instruction in the program, *i.e.*, LXI H, 2400H.

(ii) LXI H, 2200H : Load H-L pair with 2200H.

ADD M : Add the contents of the memory location, whose address is in H-L pair, to the contents of the accumulator.

In this example the instruction ADD M is an example of register indirect addressing.

5.4.4 Immediate Addressing

In immediate addressing the operand is given in the instruction itself. The examples of immediate addressing are:

(i) MVI A, 06 : Move 06 to the accumulator.

(ii) ADI 05 : Add 05 to the content of the accumulator.

(iii) LXI H, 2500H : Load H-L pair with 2500H.

5.4.5 Implicit (or Implied) Addressing

This mode of addressing is also called implied addressing or inherent addressing. Some instructions operate on only one operand, which is in the accumulator. So address need not be specified. Such addressing is known as implicit addressing. Examples are:

(i) RAL : Rotate the contents of the accumulator left through carry

(ii) RLC : Rotate the contents of the accumulator left.

(iii) CMA : Take complement of the contents of the accumulator.

Many instructions use two operands. For one of the operands they use implicit addressing while for the other operand they employ any one of the other addressing modes.

Definitions of Certain Terms Related to Addressing Modes

Before discussing other addressing modes, certain terms are to be defined. A large memory is divided into segments. The memory address of an operand consists of two components: the starting address of the segment and an offset. The starting address of the segment is supplied by the processor. The operand is placed at an offset within the segment with reference to the starting address of the segment. The offset is determined by adding any combination of three offset address elements: displacement, base and index. The combination depends on the addressing mode of an instruction to be executed. The offset is also called **effective address**. The memory address of an operand is given by

The memory address of an operand = Starting address of the memory segment + offset

Displacement. It is an 8-bit or 16-bit immediate value given in the instruction.

Base. It is the content of the base register.

Index. It is the content of the index register.

The addressing modes described in sections 5.4.6 to 5.4.12 are for Intel 8086 and other microprocessors. They are not for Intel 8085. The addressing modes of Intel 8086 with examples are given in section 5.13.10.

5.4.6 Indexed Addressing

The operand's offset is determined by adding an 8-bit or 16-bit displacement (given in the instruction) to the content of the index register.

5.4.7 Based Addressing

In this mode of addressing the operand's offset is the sum of the content of the base register and the 8-bit or 16-bit displacement given in the instruction.

5.4.8 Based Indexed Addressing

In based indexed addressing the contents of the base register and the contents of the index register are added together to form the effective address. The base register contains a base address and the index register contains an index.

5.4.9 Relative Addressing

In relative addressing a signed displacement is added to the current value of the program counter to form the effective address. This mode of addressing is commonly used in branch (or jump) instructions. The displacement is a signed quantity so as to allow either a forward or a backward jump from the location pointed out by the program counter. This mode of addressing is also known as *PC relative addressing*. The effective address specifies memory location in relation to the current value of the program counter.

5.4.10 Relative Indexed Addressing

In this mode of addressing the contents of the program counter and the contents of the index register are added together to form the effective address.

5.4.11 Page Addressing

In paged mode of addressing the memory is divided into a number of equal length pages. The page size is 256 bytes for 8-bit microprocessors and 4KB for 16-bit microprocessors. The microprocessor contains a page register to hold page number. The instruction contains an offset. The offset indicates the address within the page with reference to the starting address of the page. The numbering of pages is done in such a way that in case of 16-bit memory address, the 8-bit page number contained in the page register indicates 8 MSBs of the operand address. The advantage of this mode of addressing is that a fewer bits in the instruction are required to indicate the memory address. This results in shorter instruction and faster execution.

0 Page Addressing

Some microprocessors provide paged addressing in the limited form without employing a page register. In such a case the paged addressing is restricted only to the first 256 memory locations. The 8 MSBs of the address of these 256 memory locations are zero and this part of the memory space is called 0-page (or base page). Such mode of addressing is called *0-page addressing*.

5.4.12 Stack Addressing

In this mode of addressing the address of the operand is specified by the stack pointer (SP). The length of instruction is shortest because it does not include any address of the memory location or mention any register (just like implied mode of addressing). The contents of SP are automatically incremented or decremented after each stack operation. PUSH

instruction is used to save the contents of a register pair into the stack. The POP instruction is used to transfer the contents from the stack to the register-pair. SP contains the address of the memory location of the stack from (or to) which data are to be transferred.

The stack addressing is employed in the following cases:

- (i) When PUSH and POP instructions are used in a program by the programmer.
- (ii) When CALL instruction is used to call a subroutine. Before a program jumps from the main program to a sub-routine the contents of the PC are saved. When program goes back from the subroutine to the main program the contents of PC are restored. For this RET instruction is used at the end of a subroutine. This uses stack addressing.
- (iii) When interrupt occurs the contents of important registers are saved into the stack. For this stack addressing is used.

5.4.13 Use of Segment Registers

In case of very large memory, it is partitioned into sections called *segments*. To facilitate the memory addressing segment registers are employed. A segment register points out the starting address of a memory segment (i.e., segment base address). The memory address for an operand is given by:

Segment base address + offset within the segment

The effective address is an offset within the segment with reference to the starting address of the segment (segment base address). It depends on the addressing mode. The effective address (i.e. offset within the segment) is computed using displacement, base, index, scale etc depending upon the addressing mode employed. See details how to calculate the effective address (offset) in case of Intel 8086 and Intel 80486 in Sections 5.13.9 and 5.15.3 respectively. These microprocessors employ segment registers. The Intel 8086, 80486 etc. use segment registers. The intel 8085 does not use segment registers.

5.5 INTERRUPTS AND EXCEPTIONS

When data are ready an I/O device can interrupt CPU. After completing the current instruction at hand the CPU attends to the I/O device. The CPU enters into a subroutine known as *interrupt service subroutine* (ISS) to transfer data from the device. Each CPU has interrupt lines through which I/O device can be connected to the CPU. Slow I/O devices are connected through interrupt lines. When data transfer is over the CPU returns to the program it was executing. This is discussed in detail in the chapter dealing with I/O devices.

An interrupt caused by an external signal applied to an interrupt input line of a CPU is known as *hardware interrupt*.

Enabling of Interrupts

When interrupts are to be used they are enabled (made effective) using instruction EI (it is an 8085 instruction) in the main program.

Disabling of Interrupts

When the microprocessor is performing certain important task it may be desired to prevent the occurrence of interrupts during that period. This is achieved using DI instruction. The use of DI instruction in the program prevents occurrence of all the interrupts.

Masking of Interrupts

In certain situations it may be desired that some of the several interrupts should not occur while microprocessor is performing some important task. This is done by masking technique. Certain interrupts can be masked off (made ineffective) by software technique. The interrupt that can be masked off is called *maskable interrupt*.

Software Interrupts

The normal program execution of a microprocessor can be interrupted by a special instruction in the program. This is known as software interrupt. A special instruction is included in the program for this type of interrupt. In the 8085 there are RST n instructions for this purpose. If an RST n instruction is inserted in a program, the program is executed up to the point where RST n has been inserted. Then microprocessor stops. This is used in checking a program. By this process one checks whether the program runs correctly upto certain point. Then the program is checked by inserting RST n at another point and so on. The 8086 used INT instruction for software interrupt.

Exceptions

The internal events, which cause the processor to go out of its normal processing sequence, are called *exceptions*.

The definition of interrupts and exceptions differs from literature to literature. According to Intel handbooks the external events caused by external I/O devices, which prevent the further processing are called *interrupts*. Interrupts handle external asynchronous events. Exceptions handle internal abnormal or unusual conditions, which prevent further processing. The processor treats software interrupts as exceptions.

According to Motorola, exceptions include interrupts, traps and reset. The trap includes exceptional conditions, program faults and hardware faults.

5.6 PIN CONFIGURATION OF INTEL 8085

Fig. 5.1 shows the pin configuration of Intel 8085. It is a 40 pin IC. Pin descriptions are as follows:

A₈-A₁₅ (output). These are address lines. They form the address bus. They carry 8 MSBs of the memory address or the 8 bits of I/O address.

AD₀-AD₇ (input/output). These are address/data lines. They form time-multiplexed address/data bus, i.e., they serve dual purpose. They carry 8 LSBs of the memory address or I/O address during the first clock cycle of a machine cycle. Again they are used to carry data during the second and third clock cycles.

ALE (output). It is an address latch enable signal. It goes high during the first clock cycle of a machine cycle and enables 8 LSBs of the address to get latched into the on-chip latch of peripherals.

IO/M (output). It is a status signal to indicate whether the address sent by the microprocessor is for a memory or an I/O device. When it is high the address on the address bus is for an I/O device. When it is low the address on the address bus is for the memory.

S₀ and S₁ (output). These are status signals issued by the microprocessor to identify the various types of operations given in Table 5.1.

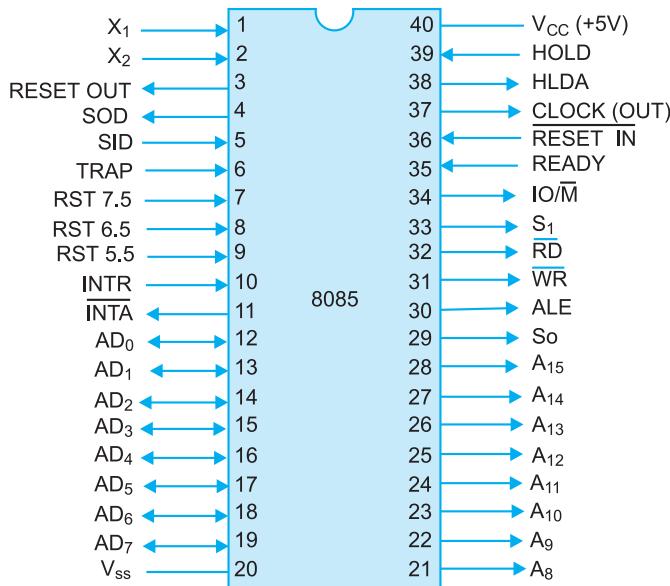


Fig. 5.1 Pin configuration of Intel 8085.

Table 5.1 Status codes for Intel 8085

S_1	S_0	Operations
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

RD (output). It is a control signal sent by the microprocessor to control read operation. The selected memory or I/O device is read when RD is low.

WR (Output). It is a control signal issued by the microprocessor to control write operation. The data on the data bus are written into the selected memory or I/O device when WR is low.

HOLD (input). When an external device wants to use the address and data bus, it sends HOLD signal to the microprocessor. Having received the HOLD signal the microprocessor completes its current instruction at hand, and then relinquishes the control of buses to allow the external device to use them.

HLDA (output). It is a HOLD acknowledgement signal. It is sent by the microprocessor to the external device to indicate that the HOLD request has been received. On the completion of data transfer, the external device removes the HOLD request. The HLDA goes low after the removal of HOLD request. The CPU takes over the control of buses half cycle after HLDA goes low.

INTR (input). It is an interrupt signal of the lowest priority.

INTA (output). It is an interrupt acknowledgement signal. It is issued by the microprocessor after INTR is received.

RST 5.5, 6.5, 7.5 and TRAP (input). These are interrupts. The TRAP is a nonmaskable interrupt and has the highest priority. Others are maskable interrupts. The order of priority is TRAP, RST 7.5, RST 6.5 and RST 5.5.

RESET IN (input). When this signal is applied the CPU is brought to reset condition. The contents of the program counter becomes zero.

RESET OUT (output). This signal indicates that the CPU is being reset.

READY (input). It is an input signal to the microprocessor. It is sent by a peripheral device to indicate whether it is ready to transfer data or not. The microprocessor examines READY signal before data are transferred. If READY is high it shows that peripherals are ready to transfer data. If READY is low the microprocessor waits till READY becomes high. The status of READY is examined in the second clock cycle of the machine cycle.

X₁ and X₂ (input). An external crystal oscillator is connected to these terminals to supply clock for the microprocessor. The crystal oscillator drives an internal circuitry which is within the microprocessor to produce a suitable clock for the operation of the microprocessor.

CLK (output). The clock is also required by some other ICs of the computer. Hence, CLK is a clock output from the microprocessor, which can be utilized for the operation of other ICs.

SID (input). It is an input line for serial data. The data received from this are loaded into the 7th bit of the accumulator when RIM instruction is executed.

SOD (output). It is an output line for serial data. The 7th bit of the accumulator is sent through this line when SIM instruction is executed.

5.7 GENERATION OF SOME SPECIAL CONTROL SIGNALS FOR MEMORY AND I/O DEVICES

The control signal \overline{RD} is used for both reading memory as well as reading an input device. Therefore, it is essential to produce two separate read signals: one for the memory and the other for the input device. Similarly, two separate write signals are generated. These special control signals are as follows:

MEMR. It is a memory read signal. It is obtained by ORing \overline{RD} with IO/\overline{M} as shown in Fig. 5.2. The memory read operation is performed when both IO/\overline{M} and \overline{RD} are low. MEMR will be low only when both IO/\overline{M} and \overline{RD} are low. The logic gates shown in the figure are equivalent to OR gates. This symbol is used where the output required is active low with both inputs low.

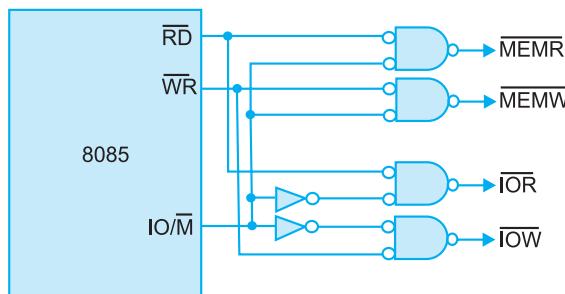


Fig. 5.2 Read/write control signals for memory and I/O.

MEMW. It is a memory write signal. It will be low when both $\overline{IO/M}$ and \overline{WR} are low.

IOR. It is an I/O read signal. It is used to read an input device. $\overline{IO/M}$ is inverted and then applied to an OR gate. Therefore, IOR will be low when IO/M is high and \overline{RD} low.

IOW. It is an I/O write signal. It will be low when IO/M is high and \overline{WR} low.

5.8 INSTRUCTION CYCLE

The main function of a CPU is to execute programs. A program consists of a sequence of instructions to perform a particular task. Programs are stored in memory. In a Von Neumann type processor such as Intel 8085, the CPU fetches one instruction at a time from the memory and executes it. First of all the CPU fetches the first instruction of the program and executes it. Then it fetches the next instruction to execute it. The CPU repeats this process till it executes all the instructions of the program. Thereafter, it may take another program if any, to execute.

The necessary steps that the processor has to carry out for fetching an instruction from the memory and executing it, constitute an *instruction cycle*. An instruction cycle consists of two parts—a *fetch cycle* and an *execute cycle*. In fetch cycle the CPU fetches the machine code of the instruction (opcode) from the memory. The necessary steps that are carried out to fetch an opcode from the memory constitute a *fetch cycle*. In execute cycle an instruction is executed. The necessary steps, which are carried to execute an instruction, constitute an *execute cycle*.

5.8.1 Fetch Operation

To fetch an opcode from a memory location the following steps are performed:

- (i) The program counter places the address of the memory location in which the opcode is stored, on the address bus.
- (ii) The CPU sends the required memory control signals so as to enable the memory to send the opcode.
- (iii) The opcode stored in the memory location is placed on the data bus and transferred to the CPU.

All the above steps require three clock cycles. If memory is slow the time taken may be more. In that case the CPU has to wait for some time till the memory transfers the opcode to the CPU. The extra clock cycles for which the CPU waits are known as *wait cycles*. Most of the microprocessors have circuitry to introduce wait cycles to cope with slow memories.

5.8.2 Execute Operation

The opcode which is fetched from the memory is placed first of all in the data register, DR (data/address buffer in case of Intel 8085). Thereafter it goes to the instruction register, IR. From the instruction register it goes to the decoder circuitry, which is within the CPU. The decoder circuitry decodes the opcode. After the opcode is decoded the CPU comes to know what operation is to be performed, and then execution begins. If the operand is in a general purpose register, the execution is immediately performed. In such a situation the time required for decoding and executing the instruction is only one clock cycle. If the required data or operand address is still in the memory, the CPU reads them from the memory. For reading data or the operand address from the memory the CPU performs read operation. The

read cycle is similar to an opcode fetch cycle. In a read cycle the quantity received from the memory is data or address instead of opcode. After receiving data from the memory, CPU performs execute operation. Some instructions may require write operation. In write cycle data are transferred from the CPU to the memory or an output device. Thus we see that an execute cycle may involve one or more read or write cycles or both.

5.8.3 Machine Cycle and State

The necessary steps, which are carried out to access a memory or I/O device, constitute a machine cycle. In other words necessary steps, which are carried out to perform a fetch, read or write operation, constitute a *machine cycle*. An instruction cycle consists of a number of machine cycles. In one machine cycle only one operation such as opcode fetch, memory read, memory write, I/O read or I/O write is performed. The first machine cycle of an instruction cycle is an opcode fetch cycle. The single-byte instructions are executed in only one machine cycle. Two-byte and three-byte instructions need more machine cycles, as additional machine cycles are required for reading/writing data from/into the memory or I/O devices.

A state (or T-state) is one subdivision of an operation performed in one clock period. These subdivisions are internal states synchronized with the system clock. So one clock cycle of the system clock is referred to as a state.

5.9 INSTRUCTION AND DATA FLOW

Two types of words: instruction word (opcode) and data word are processed during an instruction cycle. At the beginning of a fetch cycle the contents of the program counter (*i.e.*, the next memory address) are sent to the memory address register, MAR, (Fig. 5.3). From MAR the memory address is placed on the address bus so as to transfer it to the memory.

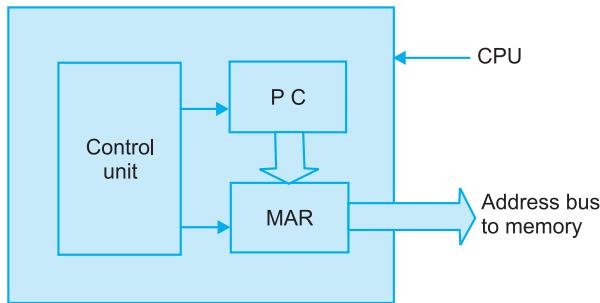


Fig. 5.3 Address transfer to memory.

A read control is sent by the CPU to the memory. Having received the address, the memory places the opcode on the data bus. Then the opcode is received in the data register DR (Fig 5.4). From DR the opcode is sent to the instruction register, IR. Thereafter the opcode is decoded by the decoder circuitry and then executed.

Any data (or address) read from the memory is also received in data register. From the data register the data are transferred to the accumulator or any other general purpose register depending on the instruction, as shown in Fig 5.5. If any data are sent to the memory from the CPU, that data also flow through DR.

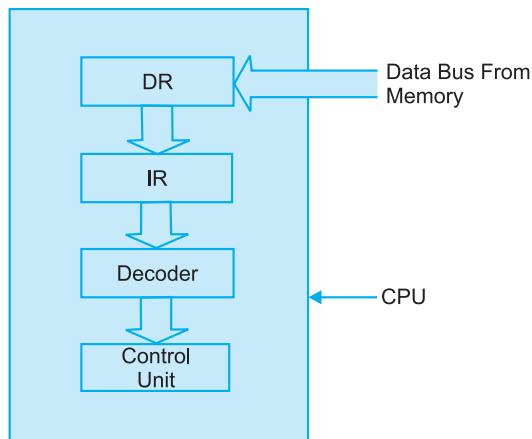


Fig. 5.4 Flow of instruction word (Opcode).

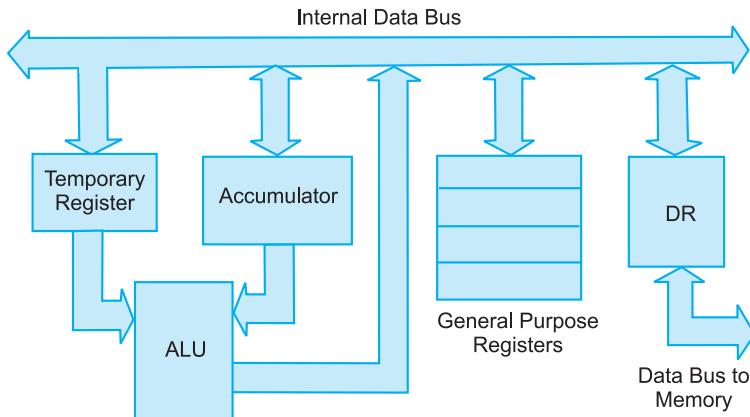


Fig. 5.5 Flow of data word.

5.10 TIMING DIAGRAM

The timing diagram is the graphical representation of the necessary steps, which are performed in a particular machine cycle. The timing diagram for opcode fetch, memory read, memory write, I/O read and I/O write of Intel 8085 is discussed in this section.

5.10.1 Timing Diagram for Opcode Fetch Operation

In a fetch operation the CPU fetches the opcode of an instruction from the memory. Fig. 5.6 shows the timing diagram for an opcode fetch operation. T_1 , T_2 , T_3 and T_4 represent clock cycles. The sequence of the operation is given below:

- (1) A low IO/M signal is sent out by the microprocessor to indicate that it wants to communicate with the memory.
- (2) Microprocessor sends out high status signals S_0 and S_1 to indicate that fetch operation will be performed.

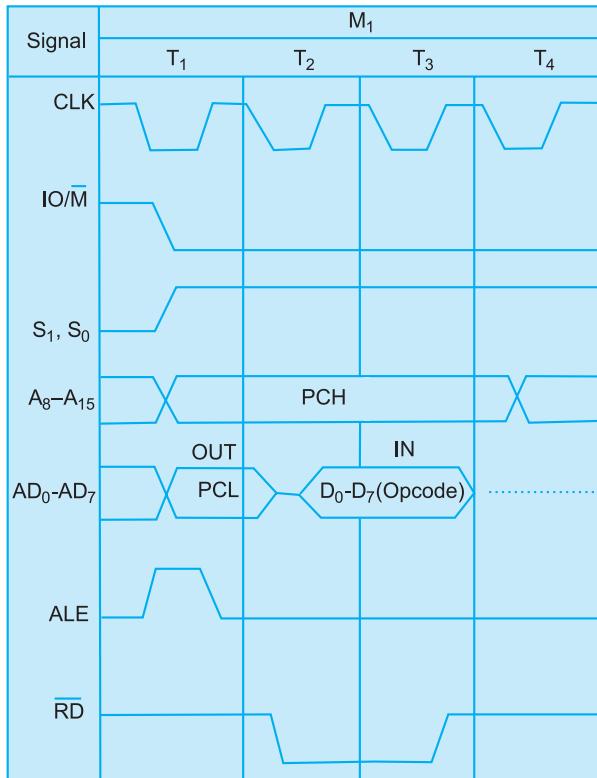


Fig. 5.6 Timing diagram for opcode fetch operation.

- (3) Microprocessor sends out the address of the memory location where the opcode is stored. The 16-bit memory address is sent over the address bus A (A₈ – A₁₅) and address/data bus AD (AD₀ – AD₇) during the first clock cycle (T₁) of a machine cycle. The 8 MSBs of the address are transmitted through A-bus and 8 LSBs of the address through AD-bus.
- (4) As the AD-bus is required for data transfer during subsequent clock cycles T₂ and T₃, it has to be made available for this purpose. Therefore, the 8 LSBs of the address have to be latched into peripheral. The microprocessor sends out an ALE signal to latch the 8 LSBs of the address into peripheral so that complete 16-bit address is available in the subsequent clock cycles. 8 LSBs of the address remain on AD bus only for T₁ state.

AD bus is a time-multiplexed bus, *i.e.*, it serves dual purpose. During the first clock cycle it carries 8 LSBs of the address, and during the second and third clock cycles it carries data.

- (5) The AD-bus becomes available to carry data during T₂.
- (6) A control signal RD sent out by the microprocessor goes low during T₂ to enable the memory for read operation.
- (7) The memory places the opcode on the data bus.
- (8) The opcode is received by the microprocessor through the data bus.
- (9) The opcode is placed into data register, DR. From DR it goes to IR.

- (10) \overline{RD} goes high during T_3 . When \overline{RD} goes high the memory is disabled.
- (11) In T_4 the opcode is sent to decoding circuitry from IR. The opcode is decoded in T_4 .
- (12) If the instruction is of single-byte, the execution is also completed in T_4 , as operands are in the general-purpose registers. Decoding and execution are completed in T_4 . Examples of single-byte instructions are: MOV r_1, r_2 ; SUB r , ADD r , RAL etc.
- (13) If the instructions are of 2- or 3-bytes, more machine cycles are required. The first machine cycle M_1 is for fetching the opcode from the memory. Subsequent machine cycles M_2, M_3 etc. are required either to read data or address from the memory or I/O devices (or to write data into memory or I/O devices).

5.10.2 Timing Diagram for Memory Read Operation

In a memory read operation the CPU reads either data or address from the memory. The data are received in DR and then go to the accumulator or any other register of the CPU according to the instruction. Let us take an example of a two-byte instruction.

MVI A, 08 : Move 08 in the accumulator

3E, 08 : The above instruction in the coded form.

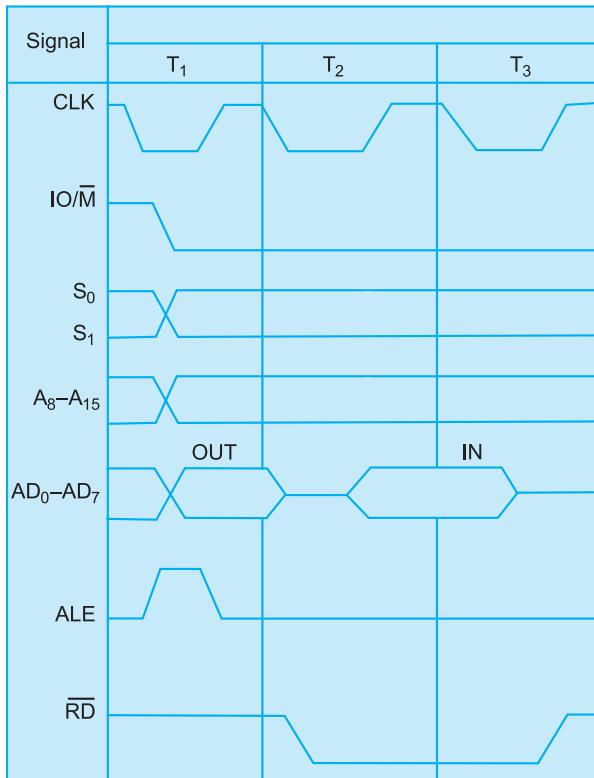


Fig. 5.7 Timing diagram for memory read operation.

3E is the opcode for MVI A instruction and 08 is the data. This instruction needs two machine cycles M_1 and M_2 . The first machine cycle M_1 is to fetch the opcode 3E from the memory. The timing diagram for opcode fetch operation was shown in Fig. 5.6. The second

machine cycle M_2 is for reading the data, 08 from the memory as shown in Fig. 5.7. The memory read cycle is similar to opcode fetch cycle except that the status signals S_0 and S_1 are 0 and 1 respectively for read operation. All other descriptions given for fetch cycle are also true for a read cycle. The memory address sent by the microprocessor in this case is the address of the memory location where the data 08 are stored.

Now let us take an example of a single-byte instruction, which requires a memory read cycle in addition to opcode fetch cycle:

- (1) **ADD M** : Add the content of the memory location whose address is in H-L pair, to the content of the accumulator.

86 : The instruction in the coded form.

This instruction requires two machine cycles. In the first machine cycle the opcode 86 is fetched from the memory. The second machine cycle is a memory read cycle in which the contents of the memory location, whose address is in H-L pair, are brought into the CPU. Thereafter the addition operation is also performed in the second machine cycle.

- (2) **MOV A, M** : Move the content of the memory location, whose address is in H-L pair, to the accumulator.

7E : The instruction in the coded form.

This instruction requires two machines cycles. In the first machine cycle the opcode 7E is fetched from the memory. The second machine cycle is a memory read cycle in which the contents of the memory location, whose address is in H-L pair, are moved to the accumulator.

- (3) Let us now consider three-byte instructions:

- (i) **LXI H, 2400H** : Load H-L pair with 2400H

21, 00, 24 : The above instruction in the coded form.

This instruction needs three machine cycles: one opcode fetch cycle and two consecutive memory read cycles. In the first machine cycle M_1 , the CPU fetches the opcode, 21. In the second machine cycle M_2 , the CPU reads 8 LSBs (i.e., 00) of the data 2400H. In the third machine cycle M_3 , the CPU reads 8MSBs (i.e., 24) of the data 2400.

- (ii) **LDA 2500H** : Load accumulator with the contents of the memory location 2500H.

3A, 00,25 : The above instruction in the coded form

This instruction requires 4 machine cycles. The first machine cycle is the fetch cycle in which the opcode 3A is fetched from the memory. The second machine cycle is a memory read cycle in which the 8 LSBs (i.e., 00) of the memory address 2500H are read. The third machine cycle is a memory read cycle in which 8 MSBs (i.e., 25) of the memory address 2500H are read. The fourth machine cycle is also a memory read cycle in which the CPU reads the content of the memory location 2500H and places it in the accumulator.

5.10.3 I/O Read Operation (IOR)

In an I/O read operation the CPU reads data from the input device or input port. The data are placed in the accumulator. Read operation is not performed with an output device. The terminology IOR or I/O is used because input/output (I/O) are used together in the literature. An I/O read cycle is similar to memory read cycle except that IO/M signal is high in case of I/O read operation. In the timing diagram for I/O read cycle all other signals remain same as shown in Fig. 5.7. A high IO/M signal indicates that the address on the address bus

is for an input device or input port. An I/O device or an I/O port has address only 8-bit long, and hence the I/O address is duplicated on A and AD-buses. In case of an I/O read operation the external latching of 8 LSBs of the I/O address is not required because the I/O address is available on the A-bus till the end of the I/O read cycle. IN instruction is used for I/O read operation.

Example:

IN 02 : Read data from input device whose address is 02.

DB,02 : The above instruction in the coded form.

This instruction is 2-byte long. It takes three machine cycles. In the first machine cycle the opcode DB is fetched from the memory. The second machine cycle is a memory read cycle to read the address (i.e., 02) of input device from the memory. The third machine cycle is an I/O read cycle in which the CPU reads data from the input device whose address is 02.

5.10.4 Memory Write Operation

In a memory write operation data are transferred from the CPU to the memory. The data are transferred from the accumulator or any other general-purpose register to the memory depending on the instruction. Fig. 5.8 shows the timing diagram for memory write cycle. The IO/M signal is low to indicate that the CPU will communicate with memory. The status signal S_0 and S_1 are 1 and 0 respectively for write operation to be performed. The CPU sends 16-bit memory address through A and AD-buses. The ALE signal is sent in T_1 state to latch 8 LSBs of the memory address. During T_2 the AD-bus is not disabled as it is done in the case of memory read operation. Rather data to be transferred to the memory from the accumulator or any other register are placed on the AD-bus. The control signal for write operation WR goes low during T_2 to enable the memory for write operation. The data sent by the CPU are stored in the memory location whose address was sent by the CPU. WR goes high during T_3 . When WR goes high the memory is disabled.

Example 1

STA 2400H : Store the content of accumulator in the memory location 2400H

32,00,24 : The instruction in the coded form.

This instruction needs four machine cycles. The first machine cycle is opcode fetch cycle to read opcode, 32 from the memory. The second machine cycle is a memory read cycle to read 00, 8 LSBs of the memory address. The third machine cycle is also a memory read cycle to read 24, the 8 MSBs of the memory address. The fourth machine cycle is a memory write cycle in which CPU sends data to store in the memory location 2400H.

Example 2

MOV M, A : Move the content of the accumulator to the memory location, whose address is in H-pair.

77 : The instruction in the coded form.

This is a single-byte instruction. It requires 2 machine cycles. The first machine cycle is an opcode fetch cycle to fetch 77 from the memory. The second machine cycle is a memory writes cycle in which the contents of the accumulator are transferred to the memory location, whose address is in H-L pair.

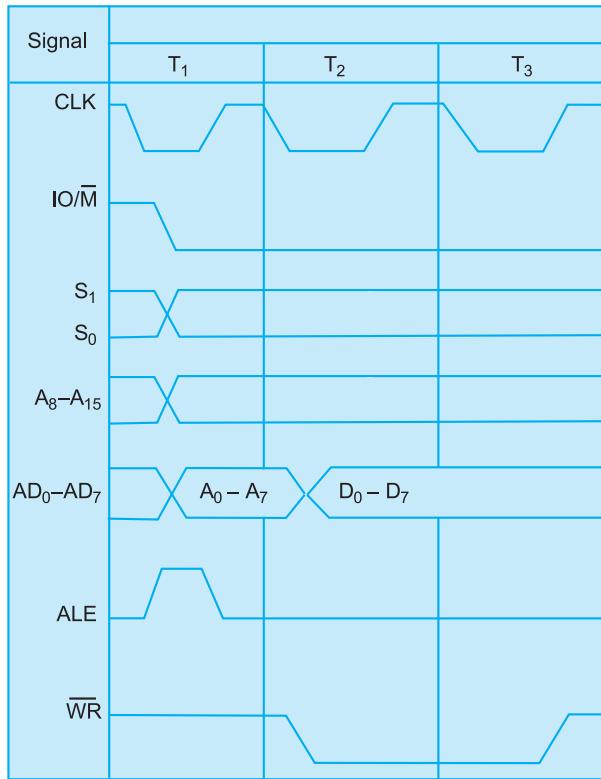


Fig. 5.8 Timing diagram for memory write operation.

5.10.5 I/O Write Operation (IOW)

In an I/O write operation the CPU transfers data from the accumulator to an input device of input port. An I/O write cycle is similar to a memory write cycle except that the control signal I/O M is high in case of I/O write operation. In the timing diagram for I/O write cycle all other signals remain same as shown in Fig. 5.8. As the address of an I/O device is of only 8 bits, it is duplicated on A and AD-bus. In case of an I/O write operation the external latching of 8 LSBs of the I/O address is not required because it is available on the A-bus till the end of the I/O write cycle. OUT instruction is used for I/O write operation.

Example

OUT 01 : Transfer the contents of the accumulator to an output device whose address is 01.

D3, 01 : The instruction in the coded form.

It is a 2-byte instruction. It requires 3 machine cycles. In the first machine cycle the opcode D3 is fetched from the memory. The second machine cycle is a memory read cycle in which the address of the output device, 01 is read from the memory. The third machine cycle is an I/O write cycle in which the content of the accumulator is transferred to the output device whose address is 01.

5.11 MICROPROCESSORS

Microprocessors of Intel, AMD (Advanced Micro Devices), Motorola and SUN are popular in India. Some important microprocessors of these manufacturers are described in this and subsequent sections.

8-Bit Microprocessors. Important 8-bit microprocessors are: Intel 8085, Motorola 6809, Zilog Z80, Zilog Z800 etc.

Intel 8088 has its data bus of 8-bits but its internal architecture is of 16-bits.

16-Bit Microprocessors. Important 16-bit microprocessors are: Intel 8086, Intel 80286, Motorola 68000, Zilog 8000 etc.

32-Bit Microprocessors. Important 32-bit microprocessor are: Intel 486, Pentium, Pentium Pro, Pentium II, Celeron, Pentium III, Pentium 4, AMD's Athlon, Power PC 601, 603, 604, 740 and 760, etc.

64-Bit Microprocessors. 64-bit microprocessors are DEC's Alpha 21264, Power PC 620, SUN's ULTRASPARC, MIPS-12000, Intel's Pentium 4EE, Itanium, HP's PA8500 etc.

5.12 INTEL 8085

It is an 8-bit microprocessor. It is implemented in NMOS technology. It is a 40-pin LSI chip. It employs 5 Vd.c. supply for its operation. 3, 5 and 6 MHz clocks are employed for the different versions of 8085. It has 80 basic instructions and 246 opcodes.

Its pin configuration, register organization, important instructions, instruction format, addressing modes, timing diagram etc. have already been discussed in the previous sections of this chapter. It is widely used in microprocessor kits in the laboratory for training students. Its description gives a clear idea of main features of a CPU, and it is simple to understand for beginners. Due to these reasons it is taught in detail in the first course on microprocessors in Electrical, Electronics and Computer Engineering. Once a student understands 8085, it becomes easy to understand other more complex microprocessors. 8-bit microprocessor-based system and 8-bit microcontrollers are widely used in simple industrial control, equipment control, instrumentation etc.

5.13 INTEL 8086

It is a 16-bit, N-channel, HMOS microprocessor. It is built in 40-pin IC package. It employs 5 Vd.c. supply for its operation. It was introduced in 1978. It contains a circuitry of 29000 transistors. It has 20 address lines. It can directly address up to 1 MB memory. Its data bus is 16-bit wide. Its address lines are time multiplexed. 16 low order address lines are multiplexed with data lines and 4 high order address lines are multiplexed with status signals. It performs arithmetic operations in binary or decimal including multiplication and division. Its clock rates are: 5, 8 and 10 MHz for its different versions. It can be interfaced with 8087 numeric data coprocessor for fast mathematical computation. The knowledge of 8086 microprocessor is essential as it forms the basis of other Intel's microprocessors. 16-bit processor-based system and 16-bit microcontrollers are used in industrial, commercial and equipment control applications, instrumentation etc. The 8086-microprocessor kits are used in the laboratory for engineering students training.

5.13.1 Operating Modes of 8086

It has two operating modes: the minimum mode and the maximum mode. The minimum mode of operation is for a simple system, which employs only one CPU and has a few peripherals. The maximum mode of operation is used when 8086 operates in multiprocessor environment. It is equipped with a strap pin MN/MX which defines its modes of operation. When MN/MX is high it operates in the minimum mode. When MN/MX is low it operates in the maximum mode. Two different sets of signals are issued by the processor through the pins 24 to 31. One set of signals are issued by the microprocessor through these pins when it operates in the minimum mode. The second set of signals are issued through the same pins when it operates in the maximum mode.

5.13.2 Register Organization

The 8086 has fourteen 16-bit registers as shown in Fig. 5.9A. They are classified into the following groups:

- (i) General purpose registers
- (ii) Pointer and index registers
- (iii) Segment registers
- (iv) Instruction pointer and status register.

General Purpose Registers

There are four 16-bit general purpose registers: AX, BX, CX, and DX. Each general purpose register consists of two 8-bit registers. For example, AX consists of AH and AL. AH is an 8-bit high-order register and AL is an 8-bit low order register. Similarly, BX consists of BH and BL, and so on as shown in Fig. 5.9A. AX serves as an accumulator. BX, CX and DX are used as general purpose registers. Besides serving as general purpose registers these registers also serve as special purpose registers. BX serves as a base register for the computation of effective memory address. CX is also used as a counter. DX is called data register. Some I/O instructions move data between an identified I/O port and the memory location addressed by register DX.

Pointer and Index Registers

Pointer and index registers as shown in Fig. 5.9A are as follows:

- (i) Stack pointer, SP
- (ii) Base pointer, BP
- (iii) Source index, SI
- (iv) Destination index, DI

SP acts as a stack pointer as discussed in case of 8085. BP, SI and DI are employed in the calculation of memory address.

Segment Registers

The 8086 has the following segment registers:

- (i) Code segment register, CS
- (ii) Data segment register, DS
- (iii) Stack segment register, SS
- (iv) Extra segment register, ES

The segment registers point to the starting address of the various memory segments. See details in Sec. 5.13.6.

Instruction Pointer

It acts as a program counter. It points to the next instruction to be executed. The content of the code segment register is added to the content of instruction pointer to compute the address of next instruction.

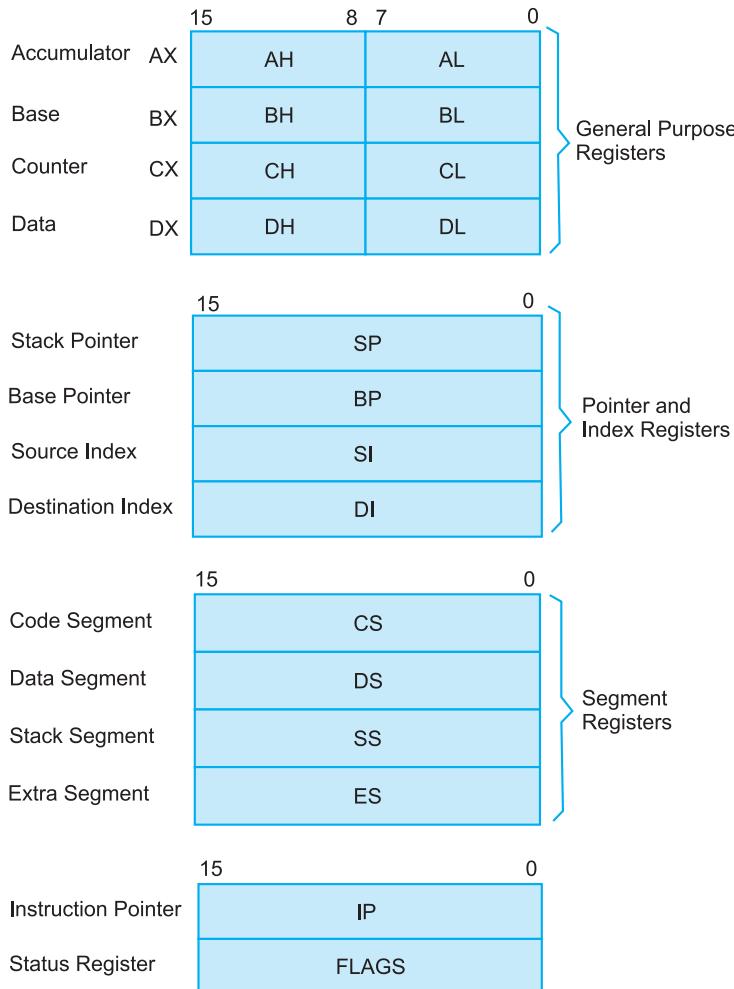


Fig. 5.9A Intel 8086 register organisation.

Status Flags

The 8086 has a 16-bit status register. It has 9 status flags as shown in Fig. 5.10. The O flag indicates overflow. The flag D is direction flag. When it is 1, the index registers SI and DI are decremented resulting in the access of strings from the highest memory address down the lowest memory address. If the direction status is 0, the index registers SI and DI are

incremented resulting in the access of strings from the lowest memory address. The trap status T is a debugging aid for single step operation of 8086. The interrupt status, I enables or disables interrupts. When it is 1, the interrupts are enabled. When I is 0, the interrupts are disabled. All other flags are similar to those of 8085. D, T and I flags are control flags and others are condition flags. Fig. 5.9B shows the block diagram of Intel 8086.

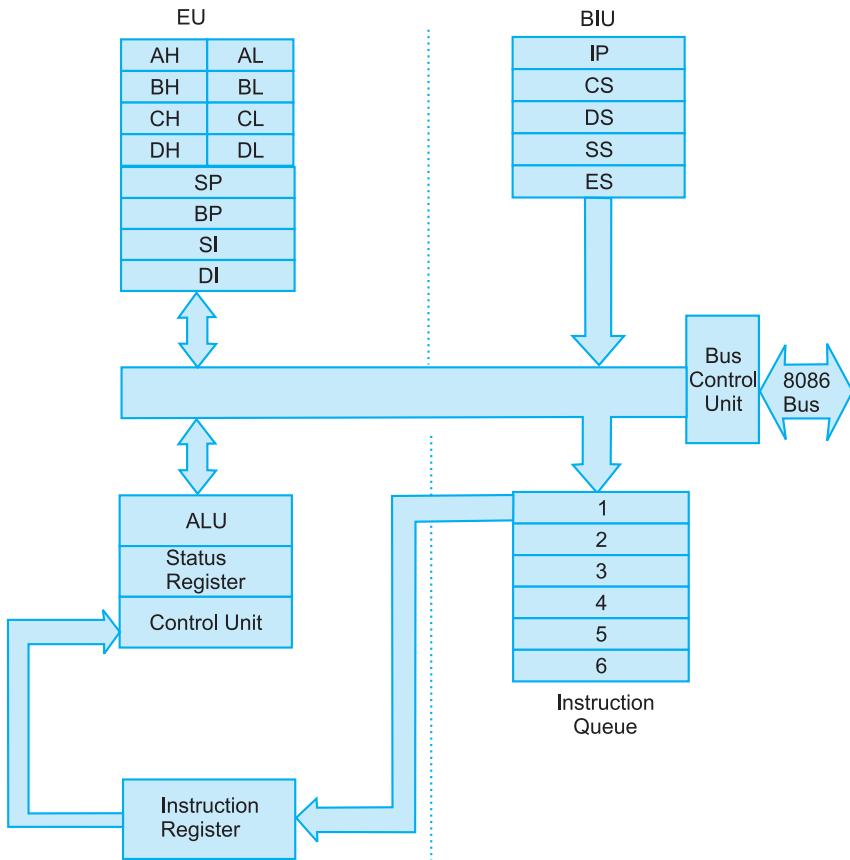


Fig. 5.9B Block Diagram of Intel 8086.

5.13.3 Pin Configuration of 8086

Fig. 5.11 shows the pin diagram of 8086. For pin description see Appendix 2. The signals shown by the side of pins 24 to 31 are for the maximum mode of operation. The signals shown in the brackets against the pins 24 to 31 are for the minimum mode of operation. In Fig. 5.11 the direction shown for signals at pin 30 and 31 are for $\overline{RQ}/\overline{GT}_0$ and $\overline{RQ}/\overline{GT}_1$ in the maximum mode of operation. In the minimum mode of operation the direction of HOLD will be inward and that for HLDA outward. The 8288, a bus controller is designed to be used with 8086 in the maximum mode of operation. It generates control signals for memory and I/O devices such as memory read, memory write, I/O read, I/O write, interrupt acknowledge, etc.

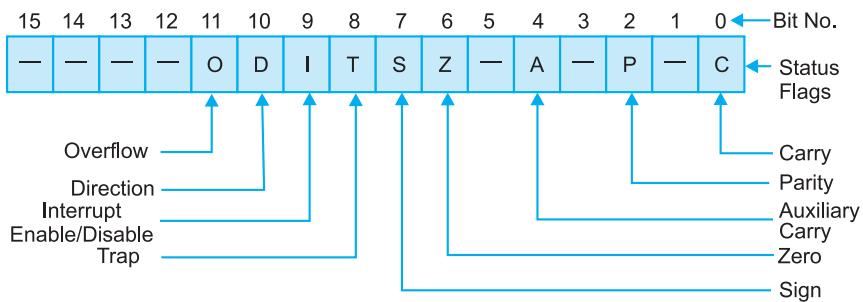


Fig. 5.10 Status flags of Intel 8086.

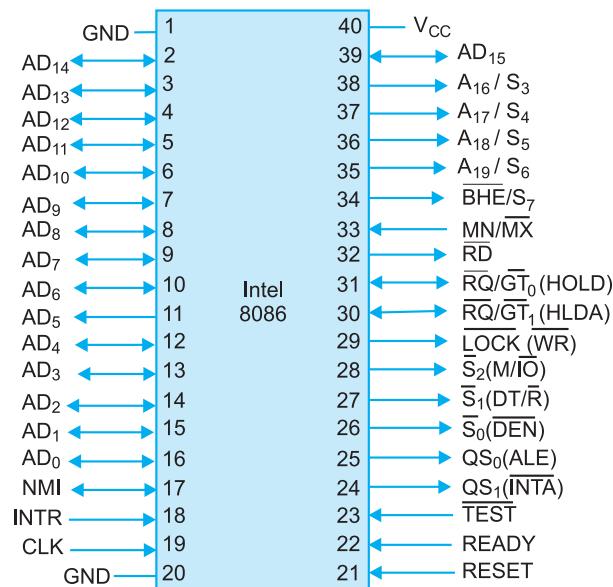


Fig. 5.11 Pin diagram of 8086.

5.13.4 Operating Principle

The internal architecture of 8086 is partitioned logically into two processing units: a *bus interface unit* (BIU) and an *execution unit* (EU). The memory and I/O devices connected to the 8086 are handled by BIU. Its function is to transfer instruction and data between the processor and peripherals. The function of the execution unit (EU) is to decode and execute instructions.

The BIU fetches opcodes from the memory, and maintains a 6-byte queue for the opcodes. Besides opcode fetching and maintaining a queue its other functions are address relocation, operand fetching and storing, and result storing. When EU desires I/O or memory access, it makes bus access request to BIU. If BIU is not currently busy, it acknowledges the request. The EU sends un-relocated operand address to BIU. The EU receives operand through BIU.

The EU receives the opcode from the queue, decodes and executes it. The queue is a first-in-first-out (FIFO) buffer. The queue is always filled-up by the BIU when at least two bytes of the queue fall vacant. The operation of BIU and EU are independent of each other,

but they can interact if required. While EU is executing instructions, the BIU fetches opcodes from the memory. The overlapping of execution of instructions and fetching of instructions makes 8086 a faster processor. Fetching of next instruction while the current instruction is being executed is called *pipelining*. Fig. 5.9B shows the block diagram of Intel 8086.

5.13.5 Interrupts

The 8086 has two hardware interrupts: INTR and NMI. The NMI is a nonmaskable interrupt. It has higher priority than INTR. The INTR is a maskable interrupt. On receipt of INTR, the 8086 issues an acknowledgement $\overline{\text{INTA}}$. On the occurrence of interrupts; CS, IP and status flags are saved into stack. If necessary, the programmer can also save other registers into the stack. The INTR can be masked by clearing interrupt flag I. The interrupt flag is cleared by executing CLI instruction. The interrupt flag is automatically cleared when an interrupt is recognized. The flag can be reset on the return from an interrupt service subroutine by executing IRET instruction. It can also be set explicitly by executing the instruction STI.

The maskable hardware interrupt INTR is used by I/O devices. It can be shared by a number of external devices. Each external interrupting device has its own ISS. When an interrupt occurs the CPU sends an acknowledgement. During the interrupt acknowledge sequence the interrupting external device supplies an 8-bit vector. This 8-bit vector points to an entry of a look-up table stored in the memory. The look-up table contains the starting addresses for ISS. This look-up table is known as *interrupt vector table*. It contains 256 entries. Each entry contains 4-byte quantity consisting of two 16-bit addresses, which provides the address of the next instruction to be executed. These addresses are loaded into CS and IP to update them. Before updating CS, IP and flags are saved in the stack on the occurrence of an interrupt.

There are INT and INTO instructions to generate software interrupts.

The 8086 generates an interrupt when divisor is zero in case of division operation.

For checking a program, it can be executed step by step. In one step one instruction is executed. This process is known as single stepping or tracing. The 8086 has a hardware feature to control single step operation. It is done by the trap flag, T. When T flag is set to 1, an interrupt is generated.

5.13.6 Memory Organization of 8086

The 8086 has 20 address lines. It can directly address upto 1 MB memory. The 16 address lines $\text{AD}_0\text{-}\text{AD}_{15}$ are multiplexed with data lines. When they carry address they can be abbreviated as $\text{A}_0\text{-}\text{A}_{15}$. When they carry data they can be abbreviated as $\text{D}_0\text{-}\text{D}_{15}$. The address lines $\text{A}_{16}\text{-}\text{A}_{19}$ are also multiplexed with status signals (for details see Appendix 2).

The memory used with 8086 is divided into 4 segments:

- (i) Code segment
- (ii) Data segment
- (iii) Stack segment
- (iv) Extra segment

The capacity of each segment may be upto 64 KB. A segment register points to the starting address of a particular memory segment. For example, the code segment register points to the starting address of the code segment, data segment register indicates the

starting address of the data segment, and so on. The memory address is computed using the contents of a segment register and an effective memory address. The effective memory address is computed through a variety of addressing modes. The instruction codes reside in the code segment. If any stack is used in the program, it resides in the stack segment. Local data, variables and constants reside in the data segment. External (global) data reside in the extra segment.

Physically, the memory is organized into two banks of 512 K bytes (maximum size of each bank) as shown in Fig. 5.12. The low order memory bank uses D_0-D_7 lines for data transfer. D_8-D_{15} are used by high order memory banks. A_0 and BHE are used as memory chip select signals. When both A_0 and BHE are low a word is transferred. When A_0 is low and BHE is high a byte from/to even memory address is transferred. When BHE is low and A_0 high, a byte from/to odd memory address is transferred. Any two consecutive bytes in the memory form a word. The address of the least significant byte forms the word address. A word can start either from an even address or from an odd address. To transfer a word the BIU requires 1 or 2 memory cycles depending on whether the starting byte of the word is an even or odd address respectively. Consequently, for transferring a word the performance can be optimized if data start from even addresses. This is specially a useful technique for stack handling.

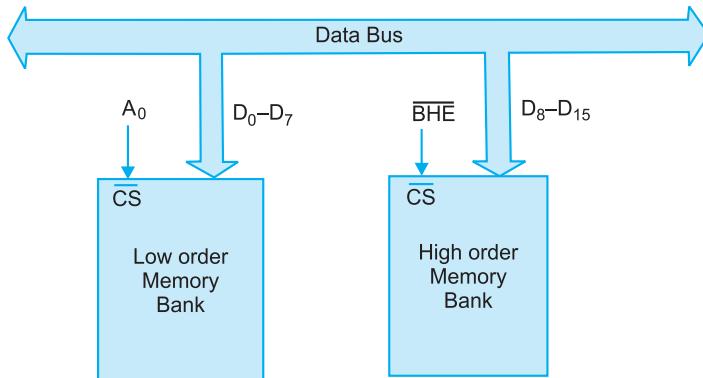


Fig. 5.12 Memory organisation of 8086 for word and byte addressing.

5.13.7 Semaphore

In a time shared system common resources may be used by more than one user. A printer can be taken as an example of a common resource. Suppose, one user is printing his result. As the computer gives only 20 millisecond time to each user, his turn is over before his printing work is completed. Now the turn of the second user comes. Suppose, he also wants to print his result. His turn also may be over before his printing work is completed. Thus the printing of both users are mixed. To overcome this difficulty *semaphore* is used. It is a software technique. It is a flag. When a user is using a common resource, the semaphore is set to 1. When it is set to 1, the common resource can not be used by any other user. On the completion of the work the semaphore is set to 0.

5.13.8 Lock

It is an instruction prefix. It prevents another processor from taking control over the buses while the next instruction is being executed.

In a multiprocessor system each microprocessor has its own local buses and memory. The microprocessors are connected through the system bus so that each microprocessor can access the common resources when needed. When a microprocessor is executing certain critical instruction using system bus, any other processor should not take over the control of the system bus. This is achieved using LOCK instruction as prefix before the critical instruction. It prevents any other processor from taking control over the system bus when one processor is in the middle of a critical instruction which uses the system bus. The LOCK instruction activates the pin LOCK which is active low and connected to the bus controller. When LOCK is low, the other processor is not granted request to take the control over the system bus.

5.13.9 Intel 8086 Instructions

Detailed discussion of Intel 8086 instructions is beyond the scope of the book. Only a few important instructions are described in this section.

5.13.9.1 MOV Instructions

By MOV instruction data can be transferred from register to register, from register to memory or from memory to register. But data transfer from memory to memory is not allowed. By MOV instruction immediate data can also be transferred to register or memory. Examples are:

- (i) MOV AX, BX ; the contents of BX register is transferred to AX register, 16-bit data transfer.
- (ii) MOV DX, CX ; [DX] ← [CX]. The content of CX is moved to DX.
- (iii) MOV BL, CL ; the content of CL register is transferred to BL register, 8-bit data transfer.
- (iv) MOV [0301H], AX ; the content of AX register is transferred to the memory location 0301 H and 0302H, 16-bit data transfer.
- (v) MOV [0301], CL ; the content of CL register is transferred to the memory location 0301H; 8-bit data transfer.
- (vi) MOV AL, [BX] ; 8-bit data transfer from the memory location addressed by BX register to AL register.
- (vii) MOV AX, [BX] ; 16-bit data transfer from two consecutive memory locations addressed by BX register to the register AX.
- (viii) MOV AL, 65H ; immediate data 65H is transferred to AL register, 8-bit data transfer.
- (ix) MOV CX, 3456H ; 16-bit data transfer. 3465H is transferred to CX register.

5.13.9.2 ADD Instructions

By ADD instruction the following operations are allowed:

- (a) Add register to register
- (b) Add register to memory

- (c) Add memory to register
- (d) Add immediate data to register
- (e) Add immediate data to memory

Examples are:

- (i) ADD AL, CL ; the content of CL is added to the content of AL and the result is placed in AL.
- (ii) ADD CL, AL ; the content of AL is added to the content of CL and the result is placed in CL.
- (iii) ADD AX, BX ; the content of BX is added to the content of AX and the result is placed in AX.
- (iv) ADD BX , AX ; the content of AX is added to the content of BX and the result is placed in BX.
- (v) ADD AX, [0301H] ; the contents of memory locations 0301H and 0302H are added to the contents of AX ; the result is placed in AX.
- (vi) ADD [0301H], AX ; the contents of AX is added to the contents of two consecutive memory locations 0301H and 0302H; result is placed in the memory locations 0301H and 0302H.
- (vii) ADD AL, [0201H] ; the content of the memory location 0201H is added to the content of AL. The result is placed in AL.
- (viii) ADD [0201H], AL ; $[0201H] \leftarrow [0201H] + [AL]$
- (ix) ADD AL, [BX] ; the content of the memory location addressed by BX register is added to the content of AL register. The result is placed in AL register
- (x) ADD [BX], AL ; The content of AL register is added to the content of memory location addressed by BX register and the result is placed in the memory location addressed by BX register.
- (xi) ADD AX, [BX] ; The content of memory locations addressed by BX register is added to the content of AX register and the result is placed in AX register.
- (xii) ADD [BX], AX ; The content of AX register is added to the content of memory locations addressed by BX register and the result is placed in the memory locations addressed by BX register.
- (xiii) ADD AX, 3594H ; Add 3594H to the content of AX register. Place the result in AX.
- (xiv) ADD CL, 59H ; Add 59H to the content of CL register. Place result in CL register.

5.13.9.3 Multiplication Instructions

There are two types of multiplication instruction:

- (i) **MUL reg or memory.** This instruction is for the multiplication of two unsigned numbers. Multiplication of an 8-bit number by an 8-bit number or 16-bit number by 16-bit number is allowed.
- (ii) **IMUL reg or memory.** This instruction is for the multiplication of two signed numbers.

MUL reg or memory. In this instruction both operands are unsigned operands. The result is an unsigned number. For 8-bit multiplication one operand is in AL register and the other operand is in any other 8-bit register or a memory location. The result is stored in AX register. For 16-bit multiplication one operand is in AX register and the other operand is in any other 16-bit register or two consecutive memory locations. The result is stored in DX and AX registers. The low-order 16 bits are stored in AX register and the high-order 16 bits are stored in DX register. Examples are:

- (i) MUL CL ; one operand is in AL. The content of AL is multiplied by the content of CL. The result is in AX.
- (ii) MUL CX ; one operand is in AX. The content of AX is multiplied by the content of CX. The result is in DX : AX.
- (iii) MUL BYTE PTR [BX] ; the content of AL is multiplied by the memory location addressed by BX register. BYTE PTR is a prefix used in assembly language for 8-bit multiplication. The result is in AX.
- (iv) MUL WORD PTR [BX] ; the content of AX is multiplied by the content of two consecutive memory locations addressed by BX register. WORD PTR is a prefix to indicate 16-bit operation. The result is in DX : AX.

IMUL reg or memory. In this instruction both operands are signed numbers. The result is a signed number. Multiplication of an 8-bit number by an 8-bit number or 16-bit number by a 16-bit number is allowed. For 8-bit multiplication one operand is held in AL register and the other operand is in any other 8-bit register or a memory location. The result is stored in AX register. For 16-bit multiplication one operand is in AX register and the other operand in any other 16-bit register or two consecutive memory locations.. The result is stored in DX and AX registers. The 16 LSBs are stored in AX register and 16 MSBs are stored in DX register. Examples are:

- (i) IMUL CL ; The content of AL is multiplied by the content of CL. The result is in AX.
- (ii) IMUL CX ; Multiply [AX] by [CX]. The result is in DX : AX.
- (iii) IMUL BYTE PTR [BX] ; Multiply [AL] by [BX]. The result is in AX.
- (iv) IMUL WORD PTR [BX] ; Multiply [AX] by [BX]. The result is in DX : AX.

5.13.9.4 Division Instructions

There are two types of division instruction:

- (i) **DIV reg or mem.** This instruction is used to divide an unsigned number by an unsigned number.
- (ii) **IDIV reg or memory.** This instruction is used to divide a singed number by a signed number.

DIV reg or mem. This instruction allows to divide a 32-bit unsigned number by a 16-bit unsigned number or to divide a 16-bit unsigned number by an 8-bit unsigned number. The 32-bit dividend is held in DX and AX registers. The 16-bit divisor is held in a 16-bit register or two consecutive memory locations. The 16-bit quotient is stored in AX register and the 16-bit remainder is stored in DX register. If the dividend is of 16-bit, it is held in AX register. The 8-bit divisor is held in an 8-bit register or a memory location. The 8-bit quotient is stored in AL register and the 8-bit remainder is stored in AH register. The examples are:

- (i) DIV CL ; this instruction divides the content of AX register by the content of CL register.
- (ii) DIV CX ; this instruction divides the content of DX : AX registers by the content of CX register.
- (iii) DIV BYTE PTR [BX] ; this instruction divides the content of AX register by the content of the memory location addressed by BX register.
- (iv) DIV WORD PTR [BX] ; this instruction divides the content of DX : AX registers by the content of two consecutive memory locations addressed by BX register.

IDIV reg or mem. This instruction allows to divide a 32-bit signed number by a 16-bit signed number or to divide a 16-bit signed number by an 8-bit signed number. The 32-bit dividend is held in DA : AX registers. The 16-bit divisor is held in a 16-bit register or two consecutive memory locations. The 16-bit quotient is stored in AX register and the 16-bit remainder is stored in DX register. If the dividend is of 16-bit, it is held in AX register. The 8-bit divisor is held in an 8-bit register or a memory location. The 8-bit quotient is stored in AL register and the 8-bit remainder is stored in AH register. The examples are:

- (i) IDIV CL
- (ii) IDIV CX
- (iii) IDIV BYTE PTR [BX]
- (iv) IDIV WORD PTR [BX]

5.13.10 Addressing Modes of Intel 8086

Intel 8086 is provided with a total of 8 addressing modes. Two addressing modes are provided for instructions which operate on the content of a register or immediate operands. These two addressing modes are:

Register Addressing. In this mode of addressing the operand is placed in one of the 8 or 16-bit general-purpose registers. Examples are:

- (i) MOV AX, BX ; Move the content of BX register to AX register.
- (ii) ADD AL, CH ; ADD the content of CH register to the content of AL register.
- (iii) ADD CX, DX ; Add the content of DX register to the content of CX register.

Immediate Addressing. In this mode of addressing the operand is given in the instruction itself. Examples are:

- (i) MOV AL, 64H ; Move 64H to AL register.
- (ii) MOV BX, 0493H ; Move 0493H to BX register.
- (iii) MOV [offset address], data ; This instruction will move 92H to the memory location 0300H and 78 to the memory location 0301H.
or MOV [0300], 7892H
- (iv) ADD AX, 0286H ; Add 0286H to the content of AX register.

The remaining 6 addressing modes are for specifying location of an operand in the memory. To understand these addressing modes see the definitions of displacement, base, index and offset in the Section 5.4.5. These addressing modes are as follows:

Direct Addressing. In this mode of addressing the operand's offset or effective address is given in the instruction itself. Examples are:

MOV AL, [0400H] ; Move the content of the effective address 0400H to AL register.

MOV [0301 H], AX ; Move the content of AL to 03101H and the content of AH to 0302H.

Register Indirect Addressing. The operand's offset is in one of the registers: BX, BP, SI or DI. Examples are:

ADD AX, [BX] ; Add the contents of memory locations addressed by BX register to the contents of AX register.

Suppose, the BX register contains 0401H. The content of 0401H is 95H and the content of the next memory location 0402H is 63H. Now 6395H will be placed in AX.

Based Addressing. In this mode of addressing the operand's offset (or effective address) is the sum of the contents of the base register (BX or BP) and an 8-bit or 16-bit displacement.

Offset (or Effective address) = [BX + 8-bit or 16-bit displacement]. Examples are:

(i) ADD AL, [BX+05]; It is a case of an 8-bit displacement. Suppose, BX register contains 0401H. Hence, the effective address (or offset) is $0401+5 = 0406H$. The content of 0406H will be added to the content of AL, and the result will be placed in AL.

(ii) ADD AL, [BX+1045H]; It is a case of 16-bit displacement.

Indexed Addressing. In this mode of addressing the operand's offset (or effective address) is the sum of an 8-bit or 16-bit displacement and the contents of an index register SI or DI.

Offset = [SI or DI + 8-bit or 16-bit displacement]

Examples are:

(i) ADD AX, [SI + 05]

(ii) MOV AX, [SI + 1239H]

Based Indexed Addressing. The operand's offset (or Effective Address) is the sum of the contents of a base register and an index register.

Offset (or Effective Address) = [BX or BP] + [SI or DI]

Examples are:

(i) MOV AX, [BX+SI]

(ii) ADD CX, [BX+SI]

Based Indexed with Displacement. In this mode of addressing the operand's offset (or effective address) is determined by adding a base register's contents, an index register's contents and an 8-bit or 16-bit displacement.

Offset (or Effective Address) = [BX or BP] + [SI or DI] + Displacement. Examples are:

(i) MOV AX, [BX + SI + 05H]

(ii) ADD AX, [BX + SI + 1085H]

5.14 BRIEF DESCRIPTION OF INTEL MICROPROCESSORS

We have already described Intel 8085 and Intel 8086 microprocessors in detail as their knowledge is necessary to understand the basic principles of computer technology. In this

section we want to discuss some microprocessors which were popular in the past. The microprocessors which are popular now will be described in the subsequent sections. Though the 486, Pentium and Pentium Pro are not used today, they will be discussed in the subsequent sections in detail as they give the background to understand current Intel microprocessors. The latest Intel's microprocessors are Pentium 4 and Itanium which will be described in the subsequent sections.

5.14.1 Intel 8088

Shortly after the introduction of 8086 (year 1978), Intel introduced the 8088 in the year 1980. The 8088 had exactly the same instruction set and internal architecture as those of Intel 8086. The main difference between the two processors was the width of their external data buses; the 8086 had 16-bit data bus; on the other hand the 8088 had only 8-bit wide data bus. It contained 16-bit registers and 16-bit internal data path. Its address bus was 20-bit wide. Its computing power was of a 16-bit microprocessor because its internal architecture was of a 16-bit microprocessor. As it had 8-bit wide data bus, it used 8-bit input/output devices which were cheaper than 16-bit input/output devices. Personal computers built around Intel 8088 microprocessor were cheaper than personal computers built around Intel 80286. Low-cost personal computer PC/XT using 8088 as CPU were very popular and widely used in 1980s. PC stands for personal computer and XT for extended technology. It could be interfaced with 8087 math coprocessor for fast mathematical computation.

5.14.2 Intel 80286

The 80286 introduced in 1982, was an enhanced version of Intel 8086. It was designed with multiuser and multitasking capability. It contained integrated memory management system, four-level memory protection and supports for virtual memory and operating system. It could directly address up to 16 MB memory. It could manage up to 1 GB of virtual memory. It contained an electronic circuitry of 1,30,000 transistors. It could be interfaced with Intel 80287 math coprocessor for fast mathematical computation. Personal computers, PC/AT, built around 80286 were very popular and widely used in 1980s. They were costlier than PC/XT which was built around 8088. AT stands for advanced technology. They used 16-bit input/output devices.

5.14.3 Intel 80186 and Intel 80188

Intel 80186. It is highly integrated 16-bit microprocessor. Besides CPU, it also contains a number of commonly used computer components on a single chip. It contains an 8086-2 CPU (an enhanced version of 8086), a programmable interrupt controller, two independent DMA channels, three programmable 16-bit timers/counters, a clock generator, programmable memory and peripheral chip-select logic, programmable wait-state generator, local bus controller, etc. Its improved versions are 80C186, 80C186EB, etc. It can be interfaced with 8087 math coprocessor for fast mathematical computation. The aim behind the development of 80186 was to make it possible to build a computer with smaller number of ICs compared to an equivalent 8086-based computer. This made 80186 an attractive microprocessor for embedded control applications such as test instruments, industrial robots, telephone switching systems etc. It did not become popular for general-purpose microcomputers.

Intel 80188. It is a highly integrated microprocessor with 8-bit wide data bus and 16-bit internal architecture. It contains 8088-2 CPU (an enhanced version of 8088), a programmable

interrupt controller, two independent high speed DMA channels, three programmable 16-bit timers/counters, a clock generator, programmable memory and peripheral chip-select logic, programmable wait-state logic, local bus controller, etc. Its improved version are : 80C188, 80C188EB etc. These processors are suitable for low-cost embedded control applications. They did not become popular for low-cost general-purpose microcomputers. It can be interfaced to 8087 math coprocessor for fast mathematical computation.

5.14.4 Intel 80386

In short, the Intel 80386 is written as Intel 386. It was Intel's popular 32-bit microprocessor. It was introduced in 1985 and widely used in 32-bit general-purpose computer before the introduction of 486 microprocessor. The 386DX version had 32-bit address bus and 32-bit data bus. It could directly address up to 4 GB of physical (real) memory. Its memory management unit (MMU) could manage up to 64TB of virtual memory. It contained an electronic circuitry of 2,75,000 transistors. Its general-purpose registers and ALU were of 32-bits. It contained MMU and protection mechanism on the processor chip itself.

The 386SL version was developed in 1990 for laptop (notebook) computers. It contained a 32-bit CPU, memory controller, cache controller, power management unit etc on the chip itself. It had only 24-bit address bus and 16-bit data bus. Its internal architecture was of 32 bits. It was designed to consume less power. An I/O subsystem 82360SL was developed to build a personal computer with 386SL microprocessor. The 82360SL contained two 8237 DMA controllers, two 8259 interrupt controllers, two 8254 programmable timer/counters, two 16450 compatible serial port controllers, one 8-bit parallel port, ISA bus logic, etc.

The 386SX version had 32-bit internal architecture but only 24 address lines and 16 data lines. It was a 32-bit entry-level processor. The computers built around this CPU were cheaper and less powerful compared to the computers built around 386DX.

5.15 INTEL 486 (INTEL 80486)

The 486 (80486) is an advanced high performance 32-bit CMOS microprocessor. It was introduced in the year 1989. It contains a 32-bit CPU, a floating-point coprocessor, 8 KB or 16 KB cache and a memory management unit on the chip. Its CPU has all the features of 386 microprocessor with enhancements to improve its speed and computing power. The 8KB cache is unified for both codes as well as data. The math coprocessor contained in the 486 microprocessor is same as 80387 math coprocessor, but it is integrated on the 486 chip itself. The 486 contains a circuitry of 1.2 million transistors. It is CISC processor. It is highly pipelined. It executes frequently used instructions in one clock cycle.

The available versions of the 486 microprocessor are: 486SX, 486DX, 486DX2 and 486DX4. The 486SX operates at 25 MHz, the 486DX at 33 MHz, the 486DX2 at 66 MHz and 486DX4 at 100 Mz. The 486DX4 contains 16KB cache whereas other versions contain only 8KB cache. The 486SX version doesn't contain math coprocessor on the chip. The math coprocessor Intel 487SX is an optional. All other versions of 486 microprocessor contain math coprocessor on the chip. All other features are same for the 486SX and other versions of 486 microprocessor. The 486SX is packed in a 168-pin grid array or 196-lead plastic quad flat package. The 486DX is packed in a 168-pin (17 × 17) grid array package. Its computing power is 54 and 80 MIPS at 66 MHz and 100 MHz respectively. Corresponding data transfer rates are 212 and 320 MB/sec respectively.

The 486 operates in two modes: real and virtual (or protected) modes. In real mode its memory addressing capacity is 1 MB. In this mode paging is not allowed. 8086 application programs are run in this mode. The primary aim of the real mode is to set up the 486 microprocessor in the protected mode. In the protected mode memory management and protection facilities are available. The 486 can directly address 4 GB of memory. The memory management unit can manage 64TB virtual memory per task. In the protected mode virtual 8086 environment is available. In virtual 8086 environment 8086 programs are run with protection. Paging facility is also available. Memory addressing capacity is 1 MB. The 486 microprocessor can run 8086 programs and 486 programs simultaneously in the protected mode of operation.

5.15.1 Functional Unit of Intel 486

The Intel 486 contains the following functional units:

- (i) Bus interface unit
- (ii) Code prefetch unit
- (iii) Instruction decoding unit
- (iv) Control and protection test unit
- (v) Execution unit
- (vi) Floating-point coprocessor
- (vii) Segmentation unit
- (viii) Paging unit
- (ix) Cache unit

Fig. 5.13A and Fig. 5.13B show the block diagram of Intel 486. The bus interface unit connects I/O devices and memory chips to the 486 microprocessor. On the receipt of request from the code prefetch unit for fetching instructions and transferring data, the processor generates address and control signals for current bus cycles.

The bus interface unit gives higher priority to data transfer over instruction fetching. The code prefetch unit fetches instruction code from cache memory and stores them in a 32-byte code queue. The instruction decoding unit gets instruction codes from 32-byte code queue and decodes them. The decoded instructions are first sent to the control and protection test unit. Before the execution of an instruction, protection test is performed. The protection test unit checks segment violations. To execute an instruction the 486 microprocessor executes a sequence of microinstructions called microprogram. The binary code for a microinstruction is called microcode. The control ROM (shown in Fig. 5.13A) stores microcodes. The execution unit executes microcodes corresponding to an instruction. It contains an ALU, register file and barrel shifter. The barrel shifter is a special type of shift register which can perform multiple number of shifts in a single operation.

The segmentation unit calculates the linear address from the logical address. It also provides four-level of protection for isolating and protecting tasks and the operating system from each other. The paging unit provides paging facility within a segment. It translates linear address into physical address. The memory management unit consists of segmentation unit and paging unit. The 8 KB or 16 KB cache memory is a unified cache for code as well as data.

5.15.2 Register Organization of Intel 486

The basic register set of Intel 486 microprocessor is shown in Fig. 5.14. It contains eight 32-bit general purpose registers: EAX, EBX, ECX, EDX, ESI, EDI, EBP and ESP. The low-

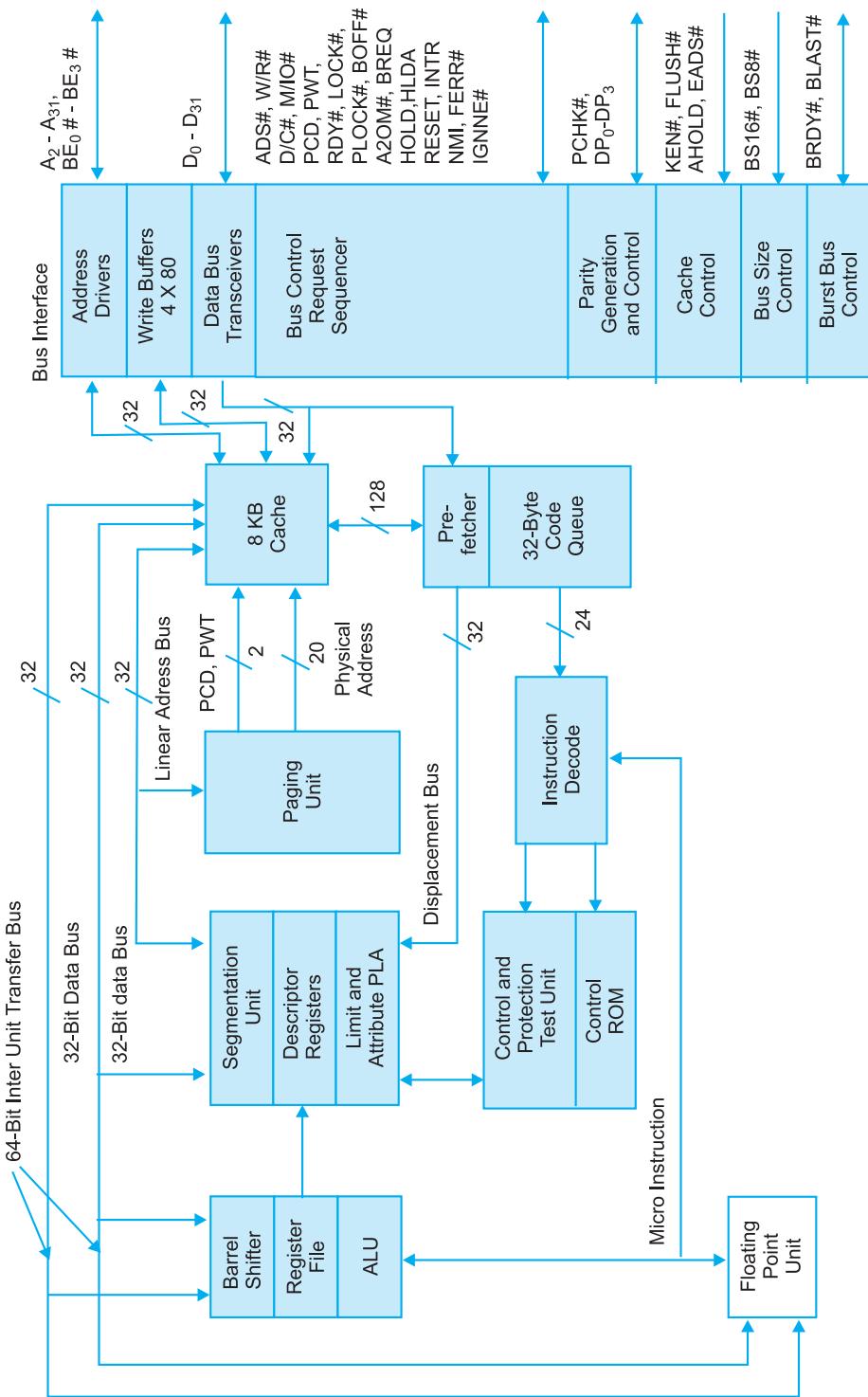


Fig. 5.13A Block Diagram of Intel 486.

order 16-bits of general purpose register are used for 16-bit operation. They can be accessed separately. The names of the low-order 16-bit registers are: AX, BX, CX, DX, SI, DI, BP and SP. Further, AL, BL, CL, DL, AH, BH, CH and DH can be accessed separately for 8-bit operation.

There are six 16-bit segment registers in the 486 microprocessor. They are: CS, DS, SS, ES, FS and GS. DS, ES and GS are data segment registers. These data segment registers have been provided to access four different data areas in the memory to permit programs to get different types of data structures. The instruction pointer EIP is a 32-bit register. It acts as a program counter. It holds the effective address (offset) of the next instruction to be executed. The offset is always relative to the starting address of the code segment, which is contained in CS. The 486 contains a 32-bit EFLAGS. Fig. 5.15 shows the status flags of Intel 486 microprocessor. It contains 14 flags. Out of 14 flags six flags are conditional flags and remaining flags are control and system flags. Six condition flags are: carry flag (CF), parity flag (PF), auxiliary flag (AF), zero flag (ZF), sign flag (SF) and overflow flag (OF). The control and system flags are: direction flag (DF), interrupt flag (IF), trap or trace flag (TF), virtual 8086 mode flag (VM), alignment check flag (ACF), resume or restart flag (RF), nested task flag (NTF) and input/output privilege-level flag (IOPL). The low-order 16-bits of EFLAGS are used for 16-bit operation. They are useful for executing 8086 and 80286 instructions. The flag register for 16-bit operation is called FLAGS.

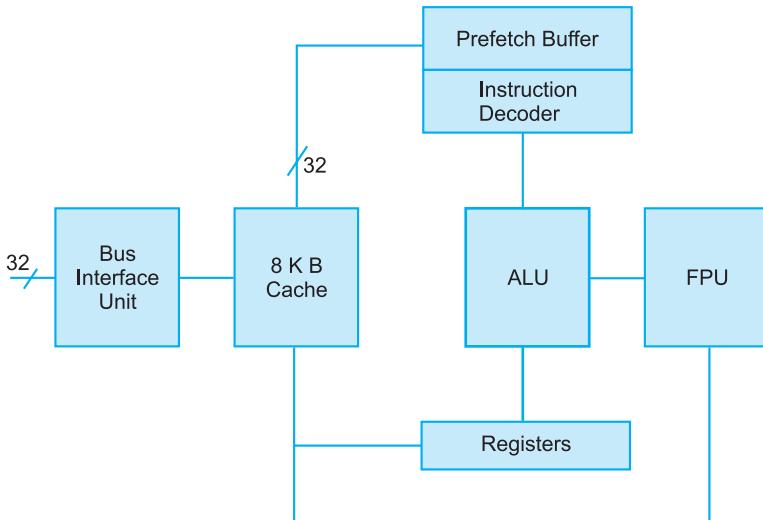
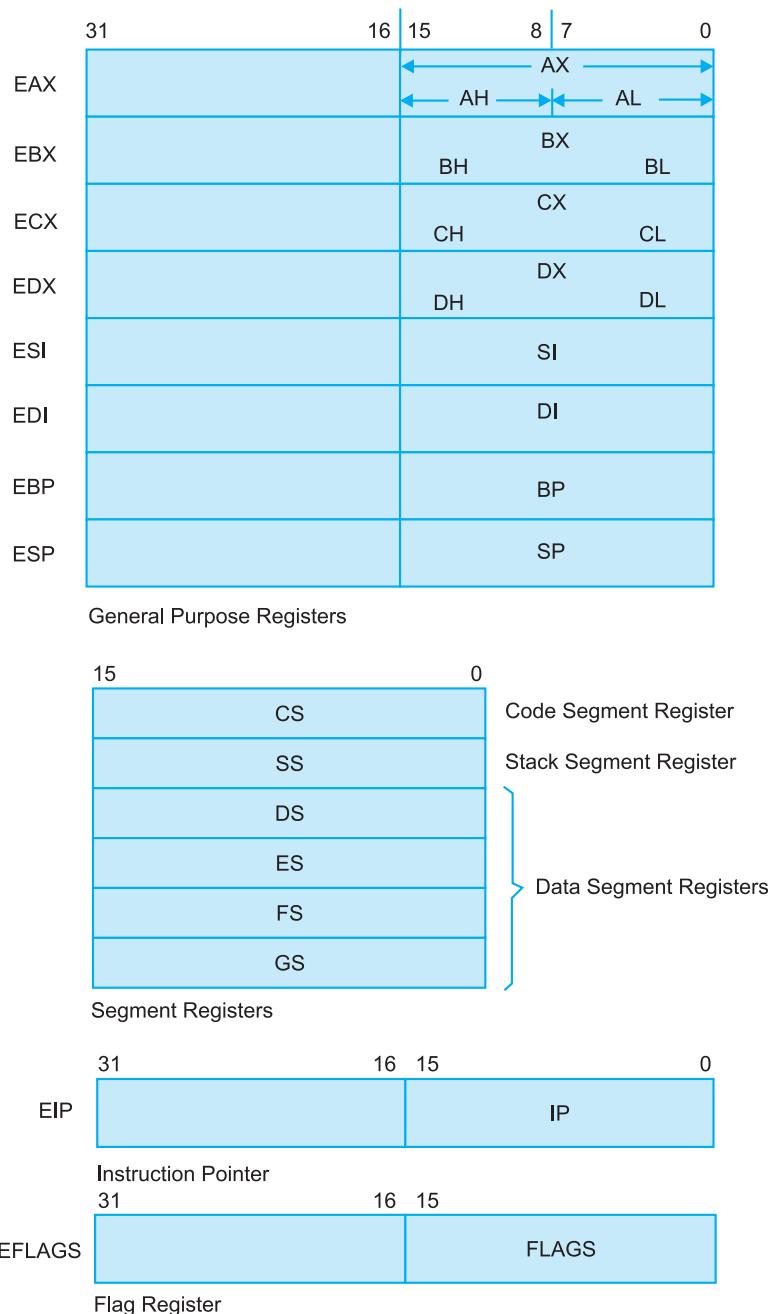


Fig. 5.13B Simplified Block Diagram of Intel 486.

Other registers of Intel 486 are: three control registers - CR1, CR2 and CR3 (CR1 is reserved for future Intel microprocessors); four system address registers - GDTR (global descriptor table register), IDTR (interrupt descriptor table register), LDTR (local descriptor table register) and TR (task state segment); seven debug registers - DR0-DR7 (DR4 and DR5 reserved for future microprocessors); five test registers, floating-point registers and segment descriptor cache registers. A segment descriptor cache register is associated with each segment register. It holds 32-bit base address for the segment, 32-bit segment limit and other necessary segment attributes. Segment descriptor cache register are invisible (not accessible) to users whereas segment registers are visible to users.

**Fig. 5.14** Basic Register Set of Intel 486.

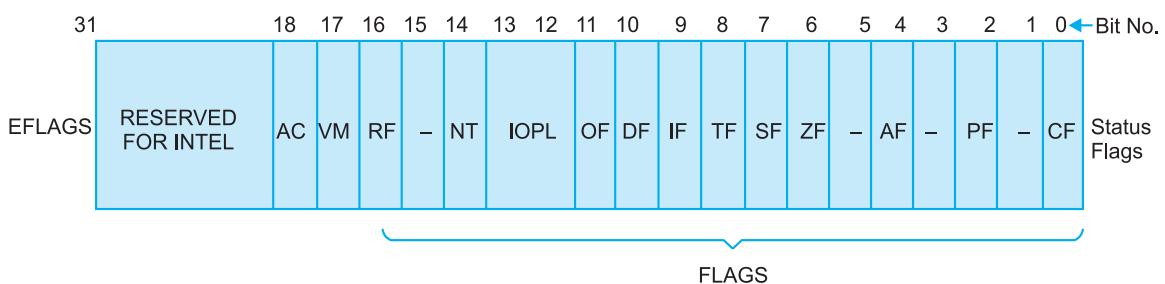


Fig. 5.15 Status flags of Intel 486.

5.15.3 Addressing Modes of Intel 486

The 486 microprocessor has 11 addressing modes. Two addressing modes are to specify register or immediate operand. They are as follows:

Register Addressing. The operand is contained in one of the 8, 16 or 32-bit general purpose registers.

Immediate addressing. The operand is included in the instruction itself. It is a part of an instruction.

The remaining 9 addressing modes are used to access memory. They specify the location of an operand in the memory. The **linear address** is the sum of the starting address (base address) of the segment and the operand's offset (effective address) within the segment. The following four address elements are used to determine operand's offset.

Displacement. It is an 8, 16, or 32-bit immediate data following the instruction.

Base. The contents of any general purpose register can be used as a base address to determine an offset. In other words any general purpose register can be used as a base register.

Index. The contents of any general purpose register except ESP can be used as index. The index registers are generally used to contain index. The index is used to access the elements of an array or a string of characters. In other words any general purpose register except ESP can be used as an index register.

Scale. The contents of an index register can be multiplied by a scale factor such as 1, 2, 4 or 8. For accessing arrays or structures the scaled index addressing is very useful.

The effective address is an offset within the segment with respect to the starting address (base address) of the segment.

The effective address or offset is computed as follows:

$$\text{Effective Address or Offset} = [\text{Base reg.}] + [\text{Index Reg.} \times \text{Scaling}] + \text{Displacement.}$$

The remaining 9 addressing modes are as follows:

Direct Addressing Mode. The operand's offset is contained in the instruction itself as an 8-, 16- or 32-bit displacement.

Register Indirect Addressing Mode. In this mode of addressing any general purpose register can be used to hold the effective address (offset) of the operand.

Based Addressing Mode. The operand's offset is determined by adding displacement to the contents of a base register.

Index Addressing Mode. The operand's offset is determined by adding displacement to the contents of an index register.

Scaled Index Addressing Mode. The contents of an index register is multiplied by a scale factor such as 1, 2, 4 or 8, and the result is added to a displacement to determine operand's offset.

Based Index Addressing Mode. The contents of an index register is added to the contents of a base register to obtain operand's offset.

Based Scaled Index Addressing Mode. To get the operand's offset the contents of an index register is multiplied by a scale factor and the result is added to the contents of a base register.

Based Index Addressing with Displacement. In this addressing mode three address elements: the contents of an index register, the contents of a base register and a displacement are added to form operand's offset.

Based Scaled Index Addressing with Displacement. The contents of an index register is multiplied by a scale factor and the result is added to the contents of a base register and a displacement to form operand's offset.

5.15.4 Memory Organization of Intel 486

The 486 handles a byte, a word and a double word. The memory is partitioned in bytes. A word (16-bit quantity) is stored in two consecutive bytes. A dword (double word, a 32-bit quantity) is stored in 4 consecutive bytes. The address of word or double word is the address of its low order byte. Memory for Intel 486 can be organized into segments. In real mode the segment is of fixed size of 64 KB. In protected mode the segment size is of variable length (1 byte to 4 GB). The memory can also be organized into 4 KB pages. The segmentation and paging can be combined to get the advantages of both techniques.

The address given in a program is known as **logical address**. The memory (RAM and ROM) actually present in a computer system is called **physical memory or real memory**. The segmentation unit translates logical address into 32-bit **linear address**. The linear address is the sum of the starting address (base address) of the segment and the operand's offset (effective address) within the segment. The paging unit translates linear address into physical address. In real mode paging is not provided and therefore, the physical address are same as the linear address. The memory is addressed by a pointer consisting of a 16-bit selector and a 32-bit offset. In other words logical address consists of a 16-bit selector and a 32-bit offset. The selector is contained in a segment register. The offset is obtained by combining address components base, index and displacement. This combination depends on the types of addressing modes.

The 16-bit selector and 32-bit offset are employed for memory addressing in both real and protected modes. In real mode the segmentation unit shifts selector (the contents of a segment register) by 4-bits left and adds 16-bit offset to the result to get 20-bit linear address. In protected mode, the contents of a segment register does not give the base address of the segment directly. Rather it points to a descriptor stored in a table in the memory called **descriptor table** (local descriptor table, LDT or global descriptor table, GDT) which gives a 32-bit base address and other information about the segment.

In real mode of operation only 20 address lines are used for memory addressing. Using 20 address lines, 2^{20} bytes = 1 MB memory can be addressed. In protected mode 32 address lines are used for memory address. Using 32 address lines, 2^{32} bytes = 4 GB memory can be addressed. The maximum capacity of physical memory of a 486-based system may be 4 GB. This capacity is used in the protected mode of operation.

A memory model in which segmentation is eliminated and the entire memory space is used in a single block, is called **flat memory model**.

Memory Management Unit (MMU) and Protection. In a multiuser/multitasking system tasks must be isolated from each other and operating system to avoid interference. Each user is assigned specific memory area for his use. The user is not allowed to write (or read) into (or from) memory area assigned to other users and the operating system. This feature is called protection. It is provided by MMU. In addition to protection MMU also computes linear address from the logical address and physical address from the linear address. The 486 microprocessor contains a hardware mechanism within the chip itself to provide protection and address translation. This hardware unit is called MMU. The MMU consists of memory segmentation unit and the paging unit. The segmentation unit provides 4 levels of protection. The protection is available only in the protected mode, not in the real mode.

Swapping and Virtual Memory. The MMU can provide more memory space than the actual existing physical memory to a program by swapping technique. When a program requires very large memory size, a part of the program currently needed for execution is brought into the memory from the hard disk. Similarly, the part of the program which is not required is pushed back into the hard disk. This to and fro movements of the parts of a program between the main (physical) memory and the secondary memory (hard disk) is called **swapping**. By this technique 64TB (tera bytes) memory space can be provided by the MMU of the a 486 microprocessor. The 64TB memory space is the virtual memory space of a 486 based system. The maximum memory space which can be provided by MMU of a system is called **virtual memory**.

5.15.5 Interrupt and Exception of Intel 486

The interruption of normal program execution of a microprocessor caused by external events is called **interrupt**. When an I/O device wants to transfer data, it informs microprocessor by sending a signal to the microprocessor. Thus the microprocessor is interrupted by an external device. The microprocessor can also be interrupted by internal abnormal conditions. An interrupt caused by an abnormal condition is called **exception**. Further, the microprocessor can also be interrupted by inserting a special instruction in a program. This type of interruption is called **software interrupt**. A software interrupt is treated as an exception by the microprocessor. The interruption caused by external devices is also called **hardware interrupt**. After an interrupt/exception occurs, the microprocessor completes its current instruction at hand and then it attends the interrupting device or takes necessary steps in case of internal abnormal conditions as specified by the system designer. For each interrupting device or abnormal condition there is a subroutine which is executed by the microprocessor after the occurrence of an interrupt/exception. This subroutine is called **interrupt service subroutine (ISS)**. When an I/O device interrupts microprocessor, the microprocessor executes an ISS to transfer data or to perform any other task as desired by the programmer. In the case of internal abnormal condition (or software interrupt), the microprocessor executes an ISS to perform the desired operation predetermined by the computer designer. Interrupts are also used to perform single step-operation and basic I/O operation (BIOS).

The 486 microprocessor has two interrupt lines: NMI and INTR. NMI is a nonmaskable interrupt. It is not available to users. It is used by the microprocessor for certain specific operation. INTR is a maskable interrupt. It is available to user. A number of I/O devices can be connected to the microprocessor through an interrupt controller.

In a 486-based system an identification number is assigned to an interrupt/exception. This identification number is called **interrupt type** or **interrupt vector**. A 486 system can entertain up to 256 interrupts/exceptions. The interrupt controller sends an 8-bit vector to the microprocessor in response to interrupt acknowledge signal. The 8-bit vector identifies the interrupting I/O device.

5.15.6 Pins and Signals of Intel 486

The 486 is packaged in a 168-pin PGA (pin grid array) package. Fig. 5.16 shows the pin diagram for Intel 486. Fig. 5.17 shows the signals of 486. $BE_0\# - BE_3\#$ are byte enable signals. They are active-low signals. They indicate which data bytes are to be accessed during read and write operation. $PCHK\#$ indicates parity error. $LOCK\#$ indicates that the current bus cycle is locked. When 64-bit or 128-bit read/write operation is to be performed, $PLOCK\#$ indicate that the current bus transaction requires more than one bus cycle for completion. $AHOLD$ is an address hold request. It allows another bus master to access the address bus of the 486 to perform cache invalidation. $BS16\#$ is the bus size 16 signal. When it is asserted 16-bit data are transferred per cycle. It is asserted by an external device which cannot send or accept 32-bit data in a single cycle. $BS8\#$ is bus size 8-signal. $KEN\#$ is cache enable signal. $EADS\#$ is used in cache invalidation cycle.

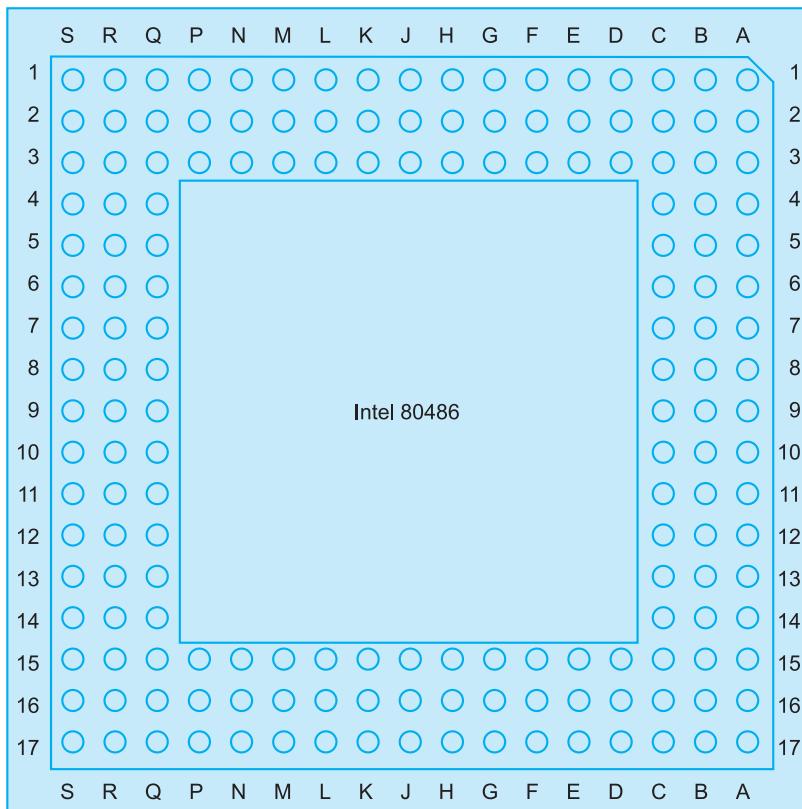


Fig. 5.16 Pin package of Intel 486.

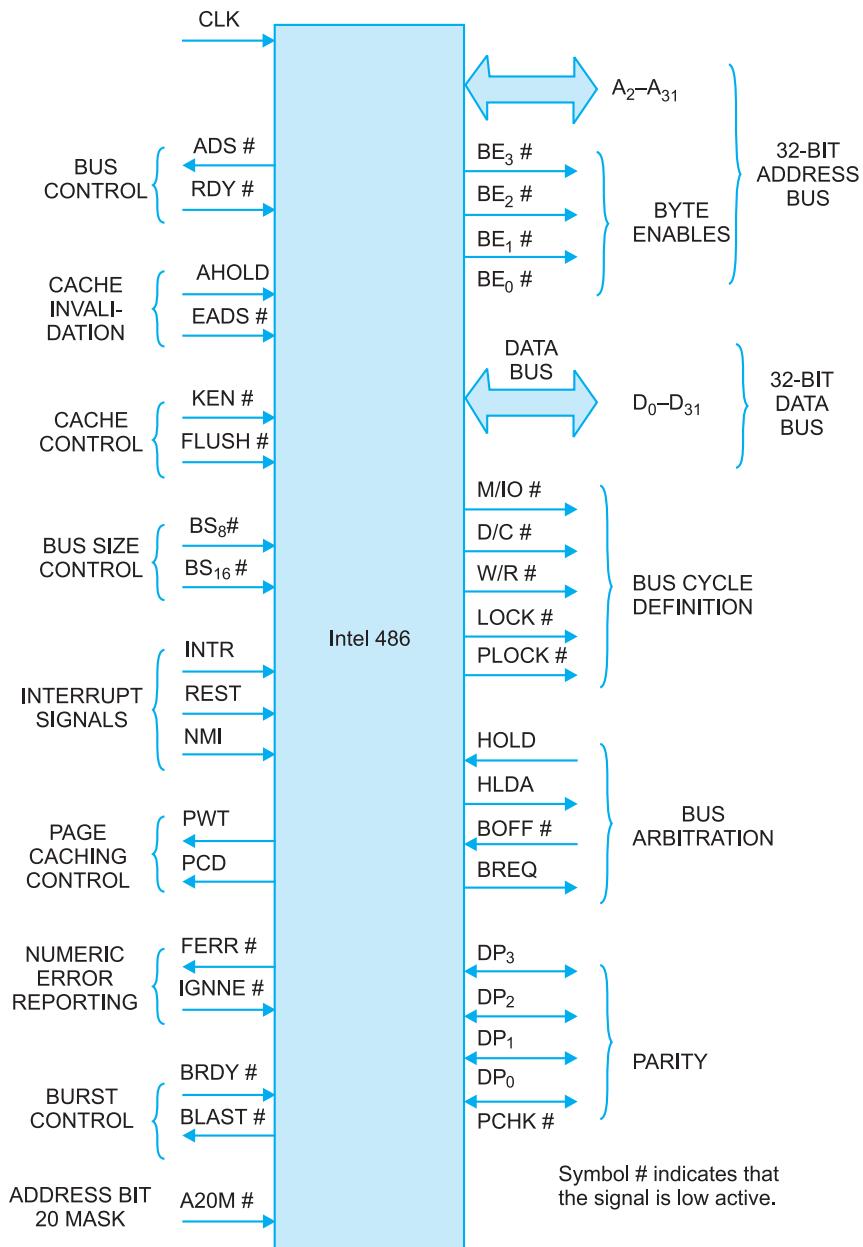


Fig. 5.17 Signals of Intel 486.

PWT controls the write operation of the external cache memory. PCD controls on-chip cache memory. FERR# indicates floating-point error. For more details see Intel microprocessor hand book.

Before the introduction of Pentium processor, the 486 was widely used CPU for 32-bit general-purpose computers.

5.16 PENTIUM PROCESSOR

The Pentium is Intel's 32-bit superscalar CISC microprocessor. It was introduced in 1993. The term **superscalar** is used for the processor which contains multiple ALUs to execute more than one instruction simultaneously in parallel per clock cycle. The Pentium contains two ALUs and can execute two instructions per clock cycle. Besides two ALUs, it also contains one on-chip FPU (Floating-Point Unit) and two 8 KB cache memory (one for instruction and the other for data). FPU is a math processor which can calculate trigonometric, logarithmic, exponential etc and can handle floating-point numbers. The Pentium has a 32-bit address bus and 64-bit data bus. The data bus used is of 64-bit with a view to supply data at faster rate to meet the need of two ALUs. Its operating clock frequency for its different versions was 60 MHZ to 233 MHZ.

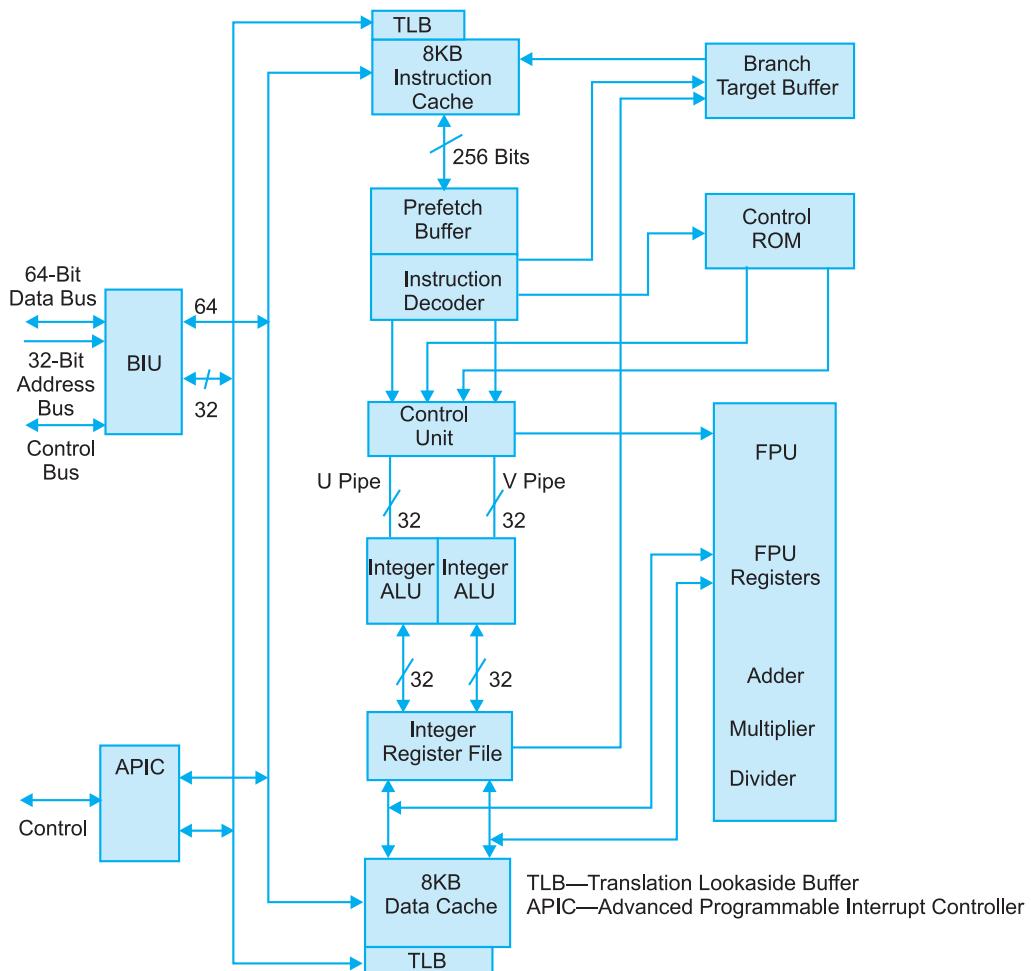


Fig. 5.18 Block Diagram of Pentium Processor.

It contained an electronic circuitry of 3.3 million transistors and used 0.35 micron process technology. One **micron** is equal to 10^{-6} metre. 0.35 micron process technology means that the distance between two components (transistors etc) in the IC is 0.35 micron. It operates at 3.3 Vdc. It uses 7 stage pipeline. It was widely used in desktop computers before the

development of Pentium Pro and Pentium II processors. Fig 5.18 shows the block diagrams of Pentium processor. TLB is translation look-aside buffer. Its function is to keep most recently accessed pages. The function of control ROM is to store micro-codes of instructions. APIC is advanced programmable interrupt controller. The Pentium processor is provided with branch prediction capability to predict which set of instructions is to be executed next. The pipelines are kept full accordingly.

5.17 PENTIUM MMX PROCESSOR

Pentium MMX processor is Pentium processor with MMX technology. MMX stands for Multi-Media extension. Pentium with MMX technology has been developed to handle multimedia computer. A multimedia computer accepts input in the form of text, graphics, images and sound. Similarly, it produces output in the form of text, graphics, images and sound. The Pentium with MMX technology contains one additional pipeline to handle multimedia. This pipeline is called MMX pipeline. The MMX pipeline is in addition to two integer pipelines and one floating-point pipeline. The MMX pipeline uses intensive parallel processing of data so that it can handle graphics, video, speech, image, etc. It employs SIMD (Single Instruction Multiple Data) technology for fast data processing. Multiple data are handled simultaneously using SIMD. Intel's Pentium with MMX contains 4.5 million transistors, has operating voltage 3.3 V, operating frequency 200 MHz, two cache memory : 16 KB each, one for instruction and one for data, etc. The Celeron, Pentium II, Pentium III and Pentium 4 also use MMX technology. The Pentium with MMX was widely used before the introduction of these processors.

5.18 PENTIUM PRO

The Pentium Pro, a 32-bit CISC processor, introduced in 1995, has data flow architecture. The data flow architecture is quite different than the architecture of Intel's earlier microprocessors such as Pentium, Pentium with MMX, 486, 386, 8086, etc. The Pentium Pro contains a second-level cache memory of 256 KB or 512 KB in addition to the first-level two caches (8 KB each, one for instruction and one for data). It also contains a cache controller and an advanced programmable interrupt controller. Its first-level instruction cache is write-through cache. The first-level data cache may be write-back or write-through. The Pentium Pro is provided with 36 address lines which can directly address up to 64 GB memory. It is packed in a 387-pin PGA (Pin Grid Array) package. It operates at 200 MHz and voltage 3.3 V. It contains 21 million transistors, including L2 cache. It was used before the introduction of Pentium II. It uses 0.35 micron process technology.

5.18.1 Data Flow and Von Neumann Architecture

The Pentium Pro uses data flow architecture whereas earlier Intel microprocessors used Von Neumann architecture. In **Von Neumann Architecture** the CPU executes instructions in a sequential order. The program is stored in the memory. The CPU fetches one instruction from the memory, decodes it and then gets data from the memory, if any. After receiving data it executes the instruction. Then it fetches the next instruction from the memory for execution. Unless it executes the instruction at hand, it does not fetch the next instruction. Thus it executes instruction in a sequential order. It uses a program counter to implement the sequential order. The program counter points to the address of the next instruction to be executed.

In **data flow architecture**, the processor (fetch and decode unit of the processor) fetches several instructions (say 20–30 instructions) from the instruction cache, decodes them and keeps their microcodes in an **instruction pool**. Then the processor (dispatch/execute unit of the processor) checks the first instruction of the instruction pool, whether its necessary data are available for its execution. Suppose, its data are not available in the data cache. There is a cache miss. Now the processor does not wait for data and hence, it does not waste its precious time. Then it checks the second instruction. Suppose, it requires the result of the first instruction for its execution. Now, processor checks the third instruction which can be executed. The processor executes it and then goes to the fourth instruction which can also be executed. The processor will execute the fourth instruction. In this way the processor checks up all the instructions which are in the instruction pool. Thus we see that the processor executes only those instructions for which data are available. The instructions which are not executed due to non-availability of data, are executed in the next round of checking of instructions in the instruction pool. When data becomes available, the remaining instructions are executed. Thus it is seen that in data flow architecture, the processor does not wait, when data are not available in data cache for some instructions. This technique increases the speed of the processor. Pentium Pro, Pentium II and Pentium III are data flow processors.

5.18.2 Functional Units of Pentium Pro

Fig 5.19 shows the functional units of Pentium Pro. The fetch and decode unit receives instructions from the instruction cache and decodes them. It sends micro-opcodes of the decoded instructions to the instruction pool. The instruction pool is a content addressable memory (*i.e.*, associative memory). The dispatch unit checks the status of micro-opcodes which are in the instruction pool. It selects and receives those micro-opcodes of the instructions from the instruction pool, which can be executed. The dispatch unit sends executable micro-opcodes to the execution unit. The execution unit contains two integer units (ALU) and one FPU (floating-point unit). It can process two integer instructions and one floating-point instruction simultaneously. After the execution of an instruction, its result is kept in the instruction pool temporarily for processing. The dispatch and execute unit selects and executes instructions in an **out of order fashion**, not in the sequential order. The retire unit checks the status of the micro-opcodes of the instructions in the instruction pool. It looks for the micro-opcodes that have been executed and can be removed from the instruction pool. It does not only remove executed instructions but also gives final result in the same order in which the instructions were written originally by the programmer. The retire unit can remove three executed instructions per clock cycle. Fig. 5.20 shows the interface of Pentium Pro to the system bus.

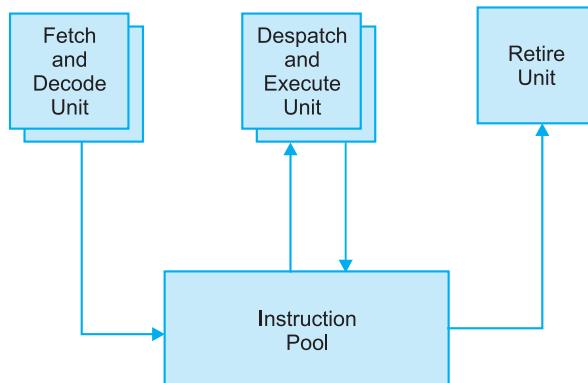


Fig. 5.19 Functional Units of Pentium Pro.

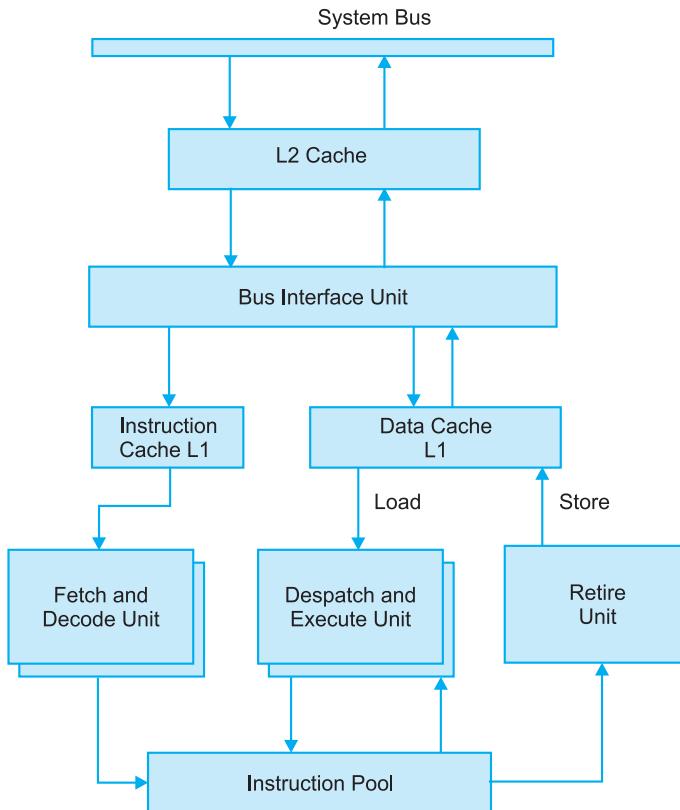


Fig. 5.20 Interfacing of Pentium Pro to System Bus.

5.19 PENTIUM II AND PENTIUM II XEON

The Pentium II processor is a multimedia 32-bit CISC processor. It was introduced in the year 1997. It is built around Pentium Pro architecture. It includes on-chip MMX pipeline to implement MMX technology. The MMX technology enables the processor to add multimedia feature. The Pentium II has 57 MMX instructions. It contains two first-level on-chip cache memory 16 KB each, one for instructions and the other for data. It operates at 3.3 V and uses 0.25 micron process technology. Its clock frequency for its different versions are in the range of 233 MHz–700 MHz. It contains 7.5 million transistors (Excluding L2 cache). The second-level cache memory is of 512 KB. It is not an on-chip cache. It is a separate IC. Both ICs *i.e.*, processor IC and second-level cache IC are packed in a single package. Pentium II was widely used in desktop computers before the introduction of Pentium III processor.

Pentium II Xeon processor is a version Pentium II having a larger second-level cache memory of 1 MB or 2 MB. It is suitable for workstations and servers.

Mobile Pentium II processor for portable computers was also developed.

5.20 CELERON PROCESSOR

The Celeron is a low-cost superscalar CISC 32-bit processor. It was introduced in the year 1999 by Intel Corporation. It is built around Pentium Pro architecture. It contains one

additional pipeline to implement MMX technology, which enables the processor to handle multimedia. It contains two first-level on-chip cache memory, 16 KB each (one for instructions and the other for data). It also contains an on-chip 128 KB second-level cache memory. Its clock frequency for its different versions was gradually increased from 266 MHz to 2.60 GHz. Earlier, it was built around Pentium II core. Later on, when Pentium III was developed, its new versions were built around Pentium III core. The new versions used 0.18 micron process technology. They used Internet streaming SIMD instructions which are to enhance multimedia performance on the Internet such as advanced imaging, streaming audio and video, 3-D simulations, animation, speech recognition etc. The Celeron uses 810E, 810E2, 815EP, 815E, 845GL, 845GV etc chipsets to interface peripherals. The system bus frequencies are 100 MHz and 400 MHz. The Celeron processor is used in cheaper desktop computers. The recent version of Celeron is called **Celeron D**. It uses 90 nm process technology and has the second-level cache of 256 KB capacity. It runs on 533 MHz bus frequency. Its some models are available for the LGA 775 sockets. LGA stands for Land Grid Array. The LGA type of socket is found on motherboards supporting Intel's new chipsets *i.e.*, the 915 and 925.

5.21 PENTIUM III PROCESSOR

The Pentium III processor is Intel's 32-bit superscalar CISC processor. It was introduced in the year 1999. It includes on-chip MMX pipeline to have multimedia features. It also includes 70 additional instructions to enhance multimedia performance on Internet. These instructions are called Internet Streaming SIMD (Single Instruction Multiple Data) Instructions. These instructions are known as Internet SSE (Internet Streaming SIMD extensions). These instructions enhance multimedia performance on Internet such as streaming audio and video, animation, 3-D simulation, advanced imaging, speech recognition etc.

The Pentium III employs data flow architecture of Pentium Pro processor. It contains 36 address lines and 64 data lines. Using 36 address lines it can address up to 64 GB memory. The 64 data lines are used to supply data at very fast rate to meet the need of two ALUs, one FPU, MMX pipeline etc. The Pentium III contains about 40 million transistors and uses 0.18 micron process technology. It operates at 3.3 volts. It contains two on-chip first-level cache memory, 16 KB each (one for instructions and the other for data). It also contains on-chip 256 KB or 512 KB second-level cache memory. The chipsets designed for Pentium III are 810, 815 and 820. Examples of some versions of Pentium III are: Coppermine, Katmai etc. The block diagram of Pentium III processor is same as those shown in Fig. 5.19 and Fig. 5.20. It was widely used in desktop computers before the introduction of Pentium 4. The clock frequency for its different versions was gradually increased from 650 MHZ to 1.2 GHz.

Mobile version of Pentium III was also developed. The operating voltages for mobile versions were: 0.975 V, 1.10 V and 1.70 V. The power consumption of mobile Pentium III was 1/2 watt to 1 watt. It used 0.13 micron process technology. Tualatin named Pentium III mobile was also available for notebook computers. The mobile version of Pentium III is written as **Pentium III M**. M stands for mobile. Pentium M is for notebook computers.

Pentium III Xeon uses larger size of second-level cache memory of 1 MB or 2 MB. It is suitable for workstations and servers.

5.22 PENTIUM 4 PROCESSOR

The Pentium 4 processors are available in 32-bit as well as 64-bit versions.

5.22.1 32-bit Pentium 4

The Pentium 4 is Intel's 32-bit superscalar CISC processor. It was introduced in the year 2000. It is an improved version of Pentium III processor. It has 144 Internet Streaming SIMD Instructions, whereas the Pentium III contains only 70 such instructions. These instructions are called Internet SSE2 (Streaming SIMD Extension 2) Instructions. These instructions enhance multimedia performance on Internet such as streaming video and audio, animation, 3-D simulations, advanced imaging, speech recognition, etc.

The 32-bit Pentium 4 employs data flow architecture of Pentium Pro processor. It contains 36 address lines and 64 data lines. Using 36 address lines it can address up to 64 GB memory. The 64 data lines are used to supply data at faster rate to meet the need to two ALUs, one FPU, MMX pipeline etc. The 32-bit Pentium 4 contains 55 million transistors and uses 0.13 micron process technology. It operates at 3.3 V. It has 423 pins and uses PGA (Pin Grid Array) package. It has two first-level cache (one for instruction and the other for data). The capacity of the data cache is 8 KB. The instruction cache is also known as **trace cache**. Its capacity is 12 KB. The instruction cache in 32-bit Pentium 4 processor is placed after the fetch and decode unit. The fetch and decode unit fetches instructions from L2 (second-level) cache. The trace cache (instruction cache) receives decoded instructions in micro-ops. This results in more efficient use of cache storage space and high volume of instruction delivery to the processor's execution unit. On the other hand, the fetch and decode unit of Pentium III processor is placed after the instruction cache. Consequently, the fetch and decode unit fetches instructions from the instruction cache. The Pentium 4 uses second-level cache memory of 256 KB or 512 KB.

A 20-stage pipeline is used in 32-bit Pentium 4 processor whereas the Pentium III processor employs a 10-stage pipeline, and Pentium processor uses a 5-stage pipeline. The clock frequency for the different versions of Pentium 4, was gradually increased from 1.2 GHz to 3.2 GHz. The system bus frequencies available are: 800 MHz, 500 MHz and 400 MHz. The chipsets used are: 865P, 850, 850E, 845E, 845PE, 845G, 845GE and 845GV. RDRAM (Rambus DRAM) is used with Pentium 4 processor. It is very fast and its data transfer rate 1.6 GB per second.

32-bit Pentium 4 Xeon processor uses a larger second-level cache memory of 1 MB–4 MB capacity. It is suitable for workstations and servers.

Pentium 4 Mobile. Mobile version of 32-bit Pentium 4 processor is also available. It is used in portable computers. The chipset 845MP is used with mobile 32-bit Pentium 4 processor.

5.22.2 64-Bit Pentium 4

Intel has also developed 64-bit Pentium 4 processors. They are 6XX series of Pentium 4 processors. Pentium 4 Extreme Edition *i.e.*, Pentium 4 EE is also a 64-bit processor. Its bus frequency is 1066 MHz. The bus frequency of 6XX series is 800 MHz. The clock frequency range for 64-bit processors is 3 GHz–3.8 GHz. These processors have L2 cache of 2 MB capacity. They use new motherboards using 915 and 925 chipsets. The 6XX series includes 630, 640, 650 and 660 series of Pentium 4 processors.

Intel has also developed dual core processor, **Pentium Extreme Edition 840, Intel Xeon Processor MP etc.** Dual core processors have less heating problem and are efficient in multitasking. In dual core processor two processors are built on a single silicon chip. They are packed in a single IC unit. In future more than two processors will be built on a single silicon chip. The two processors execute instructions independently. The program is split into two sequence of instructions. One sequence of instructions is executed by one processor. Pentium EE 840 has 2 MB of L2 cache and runs at 3.2 GHz. It uses 955 chipset.

Intel Xeon Processor MP is a dual core processor. It is a 64-bit processor suitable for larger general-purpose upto mid-size database computing tasks.

5.23 ITANIUM PROCESSOR

The Itanium is Intel's 64-bit EPIC (Explicitly Parallel Instruction Computing) processor. It was introduced in the year 2001. It is jointly designed and developed by Intel Corporation and HP (Hewlett Packard). The EPIC architecture is quite different than CISC and RISC design. Its design is based on the following techniques:

- (i) Instruction-level parallelism
- (ii) Long or very long instruction words (LIW or VLIW)
- (iii) Branch predication (not the same thing as branch prediction)

The Itanium processor fetches several bundles of instructions at a time to implement instruction-level parallelism. Multiple bundles are examined to decide which instructions can be executed in parallel. The compiler reorders these instructions to keep them in contiguous bundles. Such instructions are independent and can be executed in parallel.

In EPIC architecture several parallel execution units are used to increase the processing speed of the processor. Many work which are done by hardwares in conventional RISC and CISC design, are done by software in explicit parallelism approach of design. The compiler statically schedules instructions at compile time, rather than processor doing scheduling dynamically at run time. The instructions which can be executed in parallel are reordered and checked for dependencies by the compiler. The compiler also makes sure whether the functional units needed for the execution of instructions are available. Since the compiler is doing all such work, the processor hardware becomes much simpler.

Predication is a technique to eliminate conditional branches. After a conditional branch, instruction sets for both the conditions are executed. But at the time of retirement the result for correct condition is kept and the undesired result is discarded. Unlike CISC and RISC processors, which generally have 32 or less registers, the Itanium has 128 64-bit registers for integer, logical and general-purpose use; 128 64-bit registers for floating-point and graphic use, 64 1-bit predicate registers, instruction pointer, etc. Thus it is seen that it has much more number of registers as compared to a RISC or CISC processor. It uses multiple 10-stage pipelines for execution of multiple instructions simultaneously. It contains 4 integer units, 4 floating-point units, 4 MMX units, 2 load/store units etc. It has two first-level cache memory, 16 KB each, the second-level cache of 256 KB and third-level cache 6MB/4MB/2MB. It contains 325 million transistors (25 million for CPU and 300 million for cache memory). It uses 0.13 micron process technology. It has 64-bit address and uses RDRAM having 1.6 GB/S data transfer rate. Its operating frequency is 1.3 GHz, 1.4 GHz and 1.5 GHz. Its system bus frequency is 400 MHz. The chipsets used are 460GX, E8870 etc.

The Itanium is suitable for servers. It performs best on scientific applications with large data sets and lots of complex floating-point math. It also has good performance on transaction processing and other high-end business applications. It works on different operating system like MS-Windows, Linux and Unix.

5.24 POWERPC MICROPROCESSOR

PowerPC microprocessors are jointly developed by IBM, Motorola and Apple. They are high performance, superscalar, RISC processors. The term superscalar is used for the architecture which uses more than one pipeline for the execution of instructions.

5.24.1 Versions of PowerPC

The different versions of PowerPC are : 601, 602, 603, 604, 620, 740, 750, 760 and 970. All of these microprocessors except 620 are 32-bit microprocessors. They have 32-bit address bus, 64-bit data bus, 32 general purpose registers (each 32-bit) and 32 floating-point registers (each 64-bit). 32-bit PowerPC processors use 32-bit address for physical memory and 52-bit address for virtual memory. The PowerPC620 is a 64-bit microprocessor. It has a 64-bit address bus, a 64-bit data bus, 32 general purpose registers (each 64-bit) and 32 floating-point registers (each 64-bit).

PowerPC microprocessors are designed to work in multiprocessor system. All PowerPC microprocessors contain floating-point math processor and memory management unit on the chip.

The PowerPC601 is a entry-level general-purpose microprocessor. It is suitable for desktop computers and low-end workstations. It operates at 3.6 volts, and it is available at 66, 80 and 100 MHz. It contains 2.8 million transistors. It has a 32 KB unified instruction and data cache. Its performance ratings at 80 MHz are 105 SPECint92 and 115 SPECfp92. It executes three instructions per clock cycle.

The PowerPC602 is a 32-bit, 66 MHz microprocessor. It contains an on-chip floating-point processor. It is designed with superscalar architecture. It is suitable for graphical, multimedia, voice-recognition/synthesis and other functions that are required in new entertainment, education and information devices.

The PowerPC603 has been designed for low-power consumption. It consumes 1.2 watts of power whereas the 601 consumes 4 watts. It is suitable for portable computers such as notebook, laptop, and personal digital assistant (PDA) computers. It operates at 3.3 volts and it is available at 66, 80 and 100 MHz. It contains 2.6 million transistors. It has a 8 KB instruction cache and 8 KB data cache. Its performance ratings at 80 MHz are 75 SPECint92 and 85 SPECfp92. It executes three instructions per clock cycle. It is smaller and cheaper than PowerPC601.

The PowerPC604 is a high performance 32-bit RISC microprocessor. It is designed with superscalar architecture. It is more powerful than PowerPC601. It is a general purpose microprocessor suitable for desktop, server and workstation computers. It operates at 3.3 volts, and it is available at 80, 100, 120 and 133 MHz. Its performance rating at 133 MHz is 200 SPECint92. It contains 3.6 million transistors. It has a 16 KB instruction cache and 16 KB data cache.

The Power PC620 is a 64-bit RISC microprocessor. It operates at 133 MHz. Its performance rating is 225 SPECint92 and 300 SPECfp92. It is designed with a 64-bit address bus, 64-bit

data bus, 32 general purpose registers (each 64-bit), 32 floating-point registers (each 64-bit), 32 KB instruction cache and 32 KB data cache. It has 64-bit effective memory addressing capacity and 80-bit virtual memory addressing capacity. It executes 4 instructions per cycle. Power PC G4, G5 etc have also been developed which are 64-bit processor. G4 and G5 stand for fourth and fifth generation respectively. Earlier, Power PC processors were used in computers of IBM, Apple and some other companies. Today they are used in embedded applications, and network computers etc.

The 603e and 604e were low-power processors. The latest low-power processors are : 740 and 750. They are suitable for notebooks, mobile and desktop. The operating frequency is 400 and 450 MHz.

5.24.2 Addressing Modes of PowerPC Processors

The PowerPC processors being RISC processors have a few simple addressing modes for processing/computing. All instructions except LOAD/STORE instructions use register or immediate addressing. Only LOAD/STORE instructions use memory addressing. They are as follows:

- (i) Register Indirect Addressing
- (ii) Based Indirect Addressing with Displacement
- (iii) Based Indirect Addressing with Index.

Direct memory addressing is not used by PowerPC processors.

5.24.3 Register Organization of PowerPC Processors

The 32-bit PowerPC processors contain the following registers:

General Purpose Registers

PowerPC processors contain 32 general purpose registers (each 32-bit). They are associated with the integer unit of the processor. These registers are available for integer arithmetic calculation and address computation.

Floating-Point Registers

PowerPC processors contain 32 floating-point registers (each 64-bit). They are associated with floating-point execution unit. They are available for floating-point arithmetic instructions.

The next class of registers includes the status and control registers. Their brief descriptions are as follows:

Condition Register

There is one 32-bit condition register in the PowerPC architecture. It stores the status of result of an instruction execution. These status determine the direction of a branch. In other words the status decides where the program will jump or which is the next instruction to be executed.

Fixed-Point Exception Register

A PowerPC processor is provided with a 32-bit fixed-point exception register. It is also known as integer exception register. This register stores information regarding exceptional conditions which arise during the execution of an integer instruction. This register also stores the carry bit out of most significant bit, which arises during the execution of certain arithmetic operation.

Floating-Point Status and Control Register

A 32-bit floating-point status and control register is provided in the architecture of a PowerPC processor. This register is used to record exceptional conditions that arise during the execution of a floating-point instruction. It is also used to control the floating-point processing unit.

Link Register (LR)

A 32-bit link register is associated with the branch instructions. It is used for subroutine linkage. The subroutine linkage address is automatically loaded into the link register. The subroutine linkage address is the return address associated with a subroutine call.

Count Register (CTR)

A 32-bit count register is provided with PowerPC. It is associated with branch instructions. When a loop instruction is to be executed, the count is loaded into this register. This register is automatically decremented and tested whether the count has reached zero.

Segment Registers

There are sixteen 32-bit segment registers provided in the architecture of PowerPC processors. These registers are used for memory management.

In addition to above registers there are a number of special purpose registers such as 18 data storage interrupt status registers (DSISR), 19 data address registers (DAR), 22 decrementors (DEC), 26 machine status save restore registers 0 (SRR0), machine state register (MSR), 27 machine status save restore registers 1 (SRR1), etc.

Registers for 64-Bit PowerPC Processors

The registers for 64-bit PowerPC processors are same as those for 32-bit PowerPC processors, but all of them are 64-bit registers.

5.24.4 Memory Organization

32-bit PowerPC processors use 32-bit address bus for memory addressing. Using 32-bit address bus, up to 4 GB of physical memory can be addressed. The MMU uses 52-bit address for virtual memory addressing. The virtual memory capacity is 2^{52} bytes which is equal to 4×2^{50} bytes (4PB). Memory is divided into segments. The capacity of each segment is 256 MB (2^{28} bytes). Virtual memory is split into 2^{24} segments (2^{52} bytes = $2^{24} \times 2^{28}$ = $2^{24} \times 256$ MB). Each segment has been split into $2^{16} \times 4$ KB ($2^{16} \times 4 \times 2^{10}$ bytes = 2^{28} bytes).

Finally, 12-bit (4 KB = 2^{12}) offset is used to address a memory location which contains a byte.

A 32-bit effective address is used by the programmer to access memory. The upper-order 4 bits of the effective address are used to access one of 16 segment registers of the processor. Each segment register contains a 24-bit identification number corresponding to one 256 MB segment. The next 16 bits of the effective address identify the page within the segment which is to be accessed. The remaining 12 bits in the effective address identify the bytes within the page being accessed.

64-bit PowerPC processors use 64-bit address bus. Using 64-bit address bus, 2^{64} bytes of physical memory can be addressed. 64-bit PowerPC processors use 80-bit address for virtual memory addressing.

5.24.5 Interrupts

PowerPC processors are enabled by exception-handling mechanism to change state and react to program requests, signals, errors, or external events. The bits in the machine state register (MSR) of the processor select supervisor mode, enable and disable some interrupts and determine how the processor reacts to some system events. Two interrupts of particular interest to the assembly language programmer are: system call interrupt and program interrupt. When a programmer wants to use some operating system facility, he can use a system call instruction. Such an instruction causes a system call interrupt. The system call instruction is a subroutine call to an operating system subroutine. The program interrupt can be caused by a trap instruction. A trap instruction checks for an error condition and then traps (causes a program interrupt) if such an error occurs. From that point, the operating system may try to fix the error.

5.25 MIPS MICROPROCESSORS

MIPS (brought out by Silicon Graphics) company has developed a number of 64-bit RISC processor such as R4000, R4400, R5000 R10,000 and R12,000. R4400 is a low-power processor suitable for desktop computers. MIPS 10,000 contains 32 KB instruction cache, 32 KB data cache, 5 independent functional units, thirty-two 64-bit general purpose registers, thirty-two 64-bit floating-point registers, etc. It contains 6.4 million transistors. Its performance ratings at 200 MHz are: 300 SPECint92 and 600 SPECfp92. MIPS 12000 is the latest 64-bit RISC processor.

These processors are also used in embedded applications. They are particularly successful in video game consoles and networking applications. They are used in Sony play Station, Sony's 3-D graphics engines, Ninetendo 64 game console, high-speed routers in Internet etc.

5.26 DEC'S ALPHA MICROPROCESSORS

The Digital Equipment Corporation (DEC) has developed 64-bit RISC processor: Alpha 21064, 21164, and 21264. The operating frequency gradually increased from 200 MHz to 1000 MHz for its different versions. The Alpha 21164 contains 8 KB instruction cache and 8 KB data cache, 96 KB second-level cache, 4 functional units, thirty-two 64-bit general-purpose registers, thirty two 64-bit floating-point registers, etc. It contains 9 million transistors. Its performance ratings at 300 MHz are: 330 SPECint92 and 500 SPECfp92. The Alpha 21164 was also developed at 600 and 622 MHz. The latest Alpha processor is 21264 which operates at 700 MHz and 1000 MHz and uses 0.25 micron process technology. DEC company has been acquired by Compaq company in the year 1998.

5.27 CYRIX MICROPROCESSORS

Cyrix was competitor of Intel corporation. It has developed 32-bit microprocessors: 586, 686, 6X86MX. etc. The 586 was developed to compete with Intel's Pentium. The 6X86MX competes with Pentium II. It was designed by Cyrix and manufactured by IBM. It outperforms Pentium II. Cyrix microprocessors are cheaper than Intel's microprocessors. National Semi-

conductors purchased Cyrix in the year 1997. In the year 1999, National Semiconductors sold some parts of Cyrix to Via Technology. It kept some parts which can be used by National Semiconductors in its single-chip embedded solutions.

5.28 AMD (ADVANCED MICRO DEVICES) PROCESSORS

AMD is the competitor of Intel Corporation. It has developed 32-bit processors:

K5, K6, K6-MMX, K6-2, K6-3 and K7. The K5 was developed to compete with Pentium. The K6-MMX is a multimedia processor. It is cheaper than Pentium II. It is suitable for server computer. It is used for high-end engineering, scientific, and graphics applications. It works with all leading operating systems. It used 0.25 micron process technology. K6-2 operates at 475 MHz, K6-3 at 500 MHz and K7 at 700 MHz, 800 MHz and 1.1 GHz. The K6-3 contains 256 KB L2 (second-level) cache and 2 MB L3 (third-level) cache memory. The K7 will compete with Pentium III. K7 is now called Athlon. 1.1 GHz Athlon uses copper wires, rather than aluminium, to connect circuits. It contains 22 million transistors. Its bus frequency is 200 MHz. The code name of the next generation Athlon chip is Thunderbird. It will operate at 1 GHz and 1.1 GHz. K7 is a CISC processor. After this, AMD produced Athlon XP and Sempron processors. AMD has also produced 64-bit processor-Athlon 64, Athlon 64FX, Opteron and Athlon 64 X2 series of processors. The processors of FX series are Athlon 64 FX51, Athlon 64 FX55, Athlon 64 FX57 etc. Whenever a new processor in FX series comes, the older FX is renamed, for example, The previous FX53 is now called Athlon 64 4000+. The AMD's Opteron and Athlon 64 X2 are a 64-bit dual core processors.

5.29 SUN'S ULTRASPARC MICROPROCESSOR

The SUN's ULTRASPARC processor is a 64-bit RISC processor. SPARC stands for Scalable Processor Architecture. It contains 9 independent functional units, 16KB on-chip instruction cache, 16 KB on-chip data cache, etc. It contains 3.8 million transistors. It can issue 4 instructions per clock cycle. Its performance ratings at 167 MHz are : 275 SPECint92 and 305 SPECfp92. Its latest version is ULTRASPARC III. It contains 29 million transistors. It operates at 750 and 900 MHz. It is used in servers, workstations and supercomputers. It is suitable for multiprocessor environment.

5.30 NATIONAL SEMICONDUCTORS MICROPROCESSORS

32-bit microprocessors developed by National Semiconductors are : NS 32032, NS32332, NS32C532, M2 300, etc. The NS 32332 uses 32-bit address bus, ALU with barrel shifter, 20 bytes instruction queue, dynamically configurable bus-width of 8,16, or 32 bits depending upon the memory or I/O area addressed, burst mode of bus transactions, etc. Its latest 32-bit processors M2 operates at 300 MHz and can outperform Intel's Celeron. It is to be manufactured by National's contract manufacturer IBM.

5.31 MOTOROLA MICROPROCESSORS

Motorola introduced its first 8-bit microprocessor 6800 in 1974. It was widely used in

industry for controlling equipment in 1970s. It was followed by an advanced 8-bit microprocessor 6809. The 6809 is faster and more powerful than 6800.

In 1979, Motorola introduced an advanced powerful 16-bit microprocessor 68000. Though its data bus is 16-bit wide its internal architecture is of 32 bits. It can directly address upto 16 MB memory. It is a hybrid computer between 16-bit and 32-bit architecture. It was very popular and widely used.

Motorola introduced 68008 microprocessor in 1982. It is similar to 68000 but it has only 8-bit data bus so that it can use 8-bit I/O devices. This was developed to build cheaper computers using 8-bit I/O devices. It was not widely used.

In 1983, Motorola introduced 68010 microprocessor an improved version of 68000. The 68010 contained hardware features to support virtual memory. Its direct memory addressing capacity is 16 MB. Again in 1983 Motorola developed 68012 similar to 68010 but its memory addressing capacity is more, *i.e.*, 2 GB.

In 1984, Motorola introduced 68020, an advanced 32-bit microprocessor. Its data bus is 32-bit wide. Its direct memory addressing capacity is 4 GB. It has on-chip instruction cache. It was widely used.

Motorola introduced 68030, an advanced 32-bit microprocessor in 1987. It contains 68020 plus a MMU on the chip itself. Besides instruction cache it also contains data cache. It was widely used in minicomputers.

In 1989, Motorola introduced 68040, an advanced microprocessor. It contains a CPU, MMU, floating-point coprocessor and cache memory. It contains a circuitry of 1.2 million transistors. The 68040 was suitable to be the CPU of a minicomputer.

Motorola 680X0 series of microprocessors were similar from programmer's point of view. An improved new machine of this series can run the software of its predecessor of the series. In 1980s the 680X0 series of processors were used in desktop and server computers. They are no longer used in desktop computers or servers. But they are used for embedded applications. In mid 1990s the latest member of this series was 68060. It is designed for embedded applications. For embedded applications ColdFire processors and 68300 microcontroller have also been developed. These products are used in laser printers, network equipment, communication devices, etc.

5.32 TRANSMETA CORPORATION'S CRUSOE CHIPS

The Crusoe chips have been designed and developed for handhelds, notebooks, Net applications etc. These chips consume much less power but they are less powerful than other chips for similar applications. These chips are designed to emulate the functions of Intel's X86 microprocessors. In microprogrammed approach, new instructions can be introduced by writing new microprograms. Instructions for any other computer can be executed. This process is known as **emulation**. The software called code-morphing converts Intel's computing instructions into VLIW (Very Long Instruction Word) instruction to speed up computing by executing several simple instructions in parallel. The TM3120 chip is used in handheld computers, WebPad-like information appliances and Web browsers. The TM5400 chip is used in laptop computers and runs MS-Windows operating system. The Crusoe chips consume 1-2 watts power as compared to 25 watts power consumed by conventional processors of Intel and AMD.

5.33 VIA'S C3 PROCESSOR

The C3 processor is designed for desktop computers. It consumes much less power (only 8.5 watts) compared to Intel's Celeron (21 watts at 800 MHz) and AMD's mobile Athlon processor (22 watts at 850 MHz). Its performance is acceptable in integer operations but lacklustre in floating-point ones. Its first-level cache memory is of 128 KB and the second-level cache memory of 64 KB. It uses 133 MHz fast serial bus (FSB).

5.34 ARM MICROPROCESSORS

ARM (Advanced RISC Machines) microprocessors were developed in 1980s for small computers. The ARM does not manufacture chips, it gives designs and issues license to other chip manufacturing companies such as Intel, Hewlett-Packard, Texas Instruments, Digital Equipment etc. ARM microprocessors are fast and consume less power. They are designed for portable computer, cellular phones, communication modems, automotive engines, network appliances and other embedded applications. They are low-cost processors for low-power applications. They are 32-bit processors, can address 4 GB memory and contain 3-stage pipeline. The different versions of an ARM processor are: ARM 7, 9 and 10. Today many products by different companies based on ARM processors are in the market.

5.35 CELL PROCESSORS

The Cell processor is being jointly developed by Sony, Toshiba and IBM. It is a multi-core processor which contains 8 processors on a single silicon chip. Its clock speed is 4.6 GHz and its operating voltage 1.3 volt. It contains 234 million transistors. Its computing power is 256 GFLOPS whereas that of Pentium 4 at 3 GHz is only 6 GFLOPS. It supports multiple operating systems. The Cell processor is built for high performance, vector-based distributed computing. Each processor in the Cell processor is a vector processor.

5.36 RISC, CISC AND EPIC MACHINES

RISC is an abbreviation for reduced instruction set computer. The examples of RISC machines are DEC's Alpha, PowerPC, ULTRASPARC, etc. CISC is an abbreviation for complex instruction set computer. The examples of CISC machines are: Intel 80486, Pentium, Pentium II, Celeron, Pentium III, Pentium 4, AMD's Athlon etc.

The aim of employing RISC design is to increase the computing speed by reducing the execution time of instructions. It is seen that usually 80% of a computer's instructions are used less frequently, only for 20% of the time. The 20% of the instructions are used more frequently, for 80% of the time. The philosophy of RISC machine is based on the fact that frequently used 20% of the instructions are implemented in the hardwired circuitry of the chip itself *i.e.*, by hardware technique. The remaining 80% of the instructions are implemented by computer's software. Almost all computations are performed using simple instructions. Usually, a RISC machine is 5 to 10 times faster than a CISC machine.

The important features of a RISC microprocessor are as follows:

- (i) There are few instruction types and few addressing modes.

- (ii) Instruction set is simpler so that the processor runs faster. RISC machines have fixed and easily decoded instruction formats. All instructions are decoded by hardware.
- (iii) Most instructions are executed in a single cycle.
- (iv) Most RISC instructions involve only register-to-register operations. Memory access is limited only to load and store instructions.
- (v) Several general-purpose registers and large cache memories are employed.
- (vi) Hardwired controls are used. Microprogramming is not used. In other words microcodes are not employed to generate instructions.

The CISC machines use microprogramming. With the help of microprograms highly complex instruction set is generated. Complex instructions lead to a number of complications in both hardware and software design. In scientific computations if there are more floating-point operations CISC machines are likely to give better performance. CISC machines are more complex and less efficient.

The earliest digital computers were very simple. They employed few instructions and few addressing modes. The IBM 360 series was introduced in 1964 which employed microprogramming. The microprogramming technique leads to a complex instruction set. Within a few years even minicomputers were designed with 200 instructions and a dozen of addressing modes. To solve the problem of complexity and comparatively slow speed of CISC machines computer industry is now producing simple and fast RISC machines. EPIC (Extensively Parallel Instructions Computing) processors have been developed by Intel. For details Refer to Section 5.23.

PROBLEMS

1. Discuss the function of a CPU. What are the main sections of a CPU ? Discuss the function of each section.
2. What operations are performed by an ALU ?
3. Discuss the role of the timing and control unit of a computer.
4. What is the purpose of providing registers in a CPU? Describe various registers which are usually provided in a microprocessor ?
5. Explain the requirements of a program counter, a stack pointer and status flags in a microprocessor.
6. What status flags are normally provided in a microprocessor ?
7. Discuss the function of an index register, memory address register (MAR) and memory buffer register (MBR).
8. Explain instruction format. What are the various types of instruction formats ? Discuss them with suitable examples.
9. Discuss the various types of addressing modes which are usually provided in a microprocessor. Give suitable examples.
10. Explain what you understand by interrupts. Discuss enabling, disabling and masking of interrupts.
11. Explain the terms hardware interrupts and software interrupts. What do you understand by exceptions ?

12. Explain instruction cycles, machine cycles and states.
13. What are fetch cycles and execute cycles ?
14. Draw and explain the timing diagram for opcode fetch cycle for Intel 8085.
15. Draw and explain the timing diagram for memroy read cycle for Intel 8085.
16. Draw and explain the timing diagram for memory write cycle for Intel 8085.
17. Draw and explain the timing diagram for I/O read cycle for Intel 8085.
18. Draw and explain timing diagram for I/O write cycle for Intel 8085.
19. Draw and explain the timing diagram for $MOV r_1, r_2$ for Intel 8085.
20. Draw and explain the timing diagram for $MVI A, \text{data}$ for Intel 8085.
21. What are the various registers provided in Intel 8086 ? Discuss their function.
22. What do you understand by pipelining ? How is it achieved in Intel 8086 ?
23. Discuss the memory organization of Intel 8086.
24. What is the purpose of semaphore and LOCK provided in Intel 8086 ?
25. Explain what are Von Neumann and data flow architectures of processors.
26. In what way operating principle of Pentium Pro, Pentium II, Pentium III and Pentium 4 differ from earlier microprocessors such as Pentium or 486 processor ?
Discuss the operating principle of Pentium Pro in detail.
27. What is MMX? Discuss its need and applications.
28. Discuss the advantage of a RISC processor over CISC processor.
29. Give some example of 32-bit and 64-bit microprocessors which are manufactured by companies other than Intel.
30. Discuss the operating principle of Pentium 4.
31. Discuss the operating principle of Itanium.

6

CHAPTER

MEMORY

Memory is an essential component of a digital computer. It is a storing device. It stores programs, data, results etc. At present the following three kinds of memory are commonly used in modern computers:

- (i) Semiconductor memory
- (ii) Magnetic memory
- (iii) Optical memory

The semiconductor memory is faster, compact and lighter. It consumes less power. The semiconductor memory is static device. There is no rotating part in it. The magnetic and optical memory are slow compared to semiconductor memory. But they are cheaper than semiconductor memory. They are not static devices. They are either in the form of rotating disk or tape.

6.1 MAIN MEMORY, SECONDARY MEMORY AND BACKUP MEMORY

All computers except very small computers contain both semiconductor as well as magnetic memory. The semiconductor memory is employed as the *main memory* (or *primary memory*) of the computer. It stores programs and data which are currently needed by the CPU. The magnetic memory is used as *secondary* (or *auxiliary*) *memory*. The information which is not being currently processed resides in the secondary memory. The information which is needed by the CPU for current processing is transferred from the secondary memory to the main memory. The size of the main memory is comparatively much smaller than that of the secondary memory because of its high cost. The CPU communicates directly with the main memory. As the CPU employs semiconductor technology and has very high speed, its matching memory must be very fast. Only semiconductor devices can provide the matching speed. Thus the main memory also must employ semiconductor technology. RAM and ROM ICs are used as the main memory of the computer. RAMs are volatile, *i.e.*, their contents are erased when power goes off.

The secondary memory is employed for bulk storage (mass storage) of programs, data and other informations. It has much larger capacity than main memory. It stores system software, assemblers, compilers, useful packages, large-data files etc. The secondary memory should not be of volatile nature. It must store information permanently. The magnetic memory has this property. It retains the information once stored in it. The magnetic memories such as hard disks are the most common secondary memories used in computers.

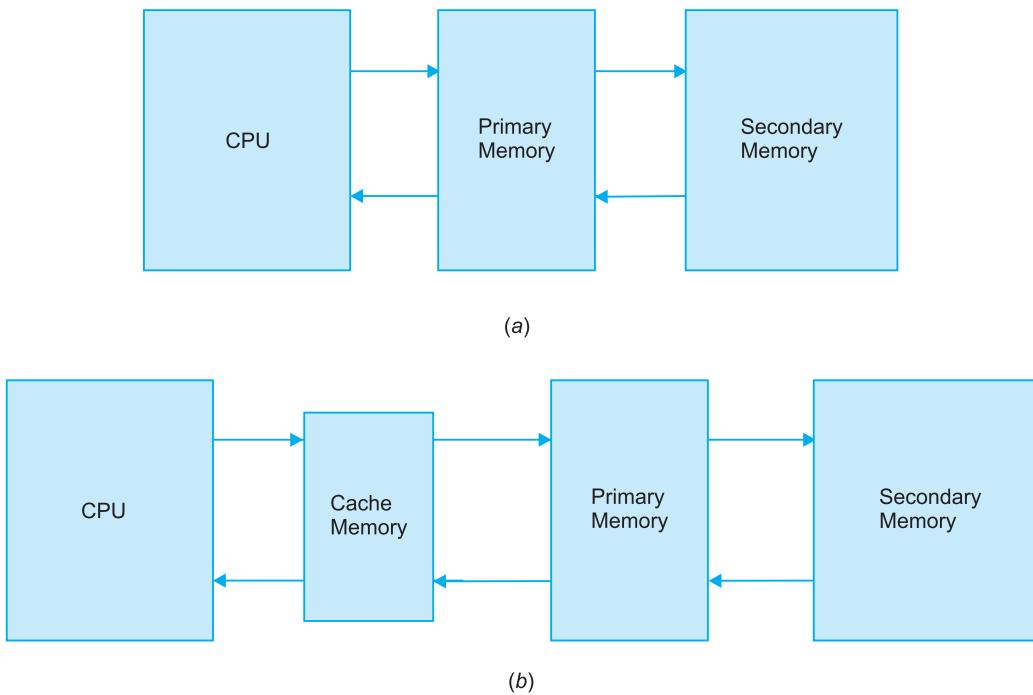


Fig. 6.1 Memory hierarchies (a) without cache memory (b) with cache memory.

The two most common memory hierarchies are shown in Fig. 6.1. Fig. 6.1 (a) shows a computer system which has only main and secondary memories but no cache memory. Fig. 6.1(b) shows a system which includes a cache memory.

Backup memory is used to store the copy of the important programs such as operating system, compilers etc. These programs are generally available in the secondary memory but their copies are also kept in the backup memory so that they can be reloaded into the secondary memory in case the programs stored in the secondary memory are lost accidentally or due to any other reason. Floppy disks, optical disks, and magnetic tapes can be used as backup storage.

6.2 CACHE MEMORY

The word cache is pronounced as cash. The cache memory is placed in between CPU and main memory. The processor is connected to the cache memory through a cache controller. It is a semiconductor memory. It consists of static RAMs. Its access time is about 10 ns (nanosecond). 1 nanosecond = 10^{-9} second. which is much less than that of the main memory. The access time of the main memory is about 50 ns. The capacity of the cache memory is 2 to 3 percent of that of the main memory. It stores instruction codes and data, which are to be currently executed by the CPU. It is used to reduce the average access time for instructions and data, which are normally stored in the main memory. A cache memory also needs a cache controller. Cache controller ICs are available.

The modern 32-bit and 64-bit microprocessors operate at very high speed. Their clock rates are usually in the range of 1GHz–3.8GHz. The memory matching with high-speed microprocessor must be very fast. But very fast memory is very expensive. If a fast

microprocessor operates with conventional main memory, it has to operate with several wait states. This will reduce the speed of the computer. A compromise is made, and a high-speed cache memory is used to supply currently needed instructions and data to CPU. The main memory stores program and data, which is to be processed by the CPU. The currently needed instructions and data of the program are loaded into the cache from the main memory.

There are two types of cache schemes: **write-through** and **write-back**. In a write through cache the main memory is updated each time the CPU writes into the cache. The advantage of the write-through cache is that the main memory always contains the same data as the cache contains. This characteristic is desirable in a system which uses direct memory access (DMA) scheme of data transfer. The I/O devices communicating through DMA receive the most recent data. In a write-back cache, only the cache memory is updated during a write operation. The updated locations in the cache are marked by flags so that later on when the word is removed from the cache, it is copied into the main memory. The words are removed from the cache time to time to make room for a new block of words. The contents of updated cache memory locations are copied into the main memory when there are free processor cycles. This requires additional hardware support, but improves performance, since the exchanges between cache and the main memory are fewer and better timed. The write-back scheme increases performance by reducing the utilization of buses and preventing unnecessary bottlenecks in the system. This scheme is faster and hence it is preferred.

The technique of accessing a cache memory differs from that of the main memory. To access main memory the CPU sends an address to it. In response of this the main memory sends data contained at the specified memory address. On the other hand cache memory uses parallel searching of the required data. It first compares the incoming address to the addresses present in the cache. If the address matches, it is said that a '**hit**' has occurred. Then the corresponding data is read by the CPU. If the address does not match, it is said that a '**miss**' has occurred. When a miss occurs, the data is read from the main memory. The data read from the main memory is also written to the cache memory, so that when this specific address is accessed next time a hit occurs. Usually, hit ratio is above 90 per cent. The hit ratio is defined as the ratio of the number of hits to the total number of read requests sent to the cache memory by the CPU. Usually, setassociative cache memory is used in a computer. The memory in which parallel search of the required data is done, is known as associative memory. Set associative memory is a kind of associative memory.

Cache memory is placed at two or three levels. They are called first-level cache (L1), second-level cache (L2), and third-level cache (L3). Some microprocessors contain L1 and L2 within the microprocessors. Some microprocessors contain L1, L2 and L3 within the microprocessor. Cache within the microprocessor is called internal cache. The cache outside the processor is called external cache.

In a write-through cache the main memory is to be updated each time the CPU writes into the cache. If CPU waits for write operation of the main memory, it is slowed down. Moreover, the CPU does not need the result of the write operation immediately. Therefore, it is not necessary for the CPU to wait for the write operation to be completed. Hence, to improve performance a **write buffer** can be included for temporary storage for write requests. The CPU keeps each write request into this buffer, and executes the next instruction. The information from the write buffer is sent to the main memory whenever the memory is not responding to read requests. In case of write-back cache also a write buffer may be included. It holds temporarily the flag marked blocks of data, which are to be removed from the cache and to be stored in the main memory. The read operation from the main memory for a new block of data needed by the CPU must be performed immediately. The CPU should not wait

till the room is made in cache by removing a block and then to perform read operation to get the new block of data from the main memory. For this purpose also the buffer is needed.

6.2.1 Cache Controllers

When the CPU sends an address of instruction code or data, the cache controller examines whether the content of the specified address is present in the cache memory. If the instruction code or data is available in the cache memory, the cache controller enables the cache memory to send the addressed code/data to the CPU. If the addressed instruction code or data is not present in the cache memory the cache controller enables the controller of the main memory to send the addressed code/data from the main memory. The main memory outputs the addressed code/data on data bus. From the data bus the code/data is sent to the CPU as well as cache memory.

Cache Controllers, Intel 82496, 82497 and 82498

These cache controllers have been developed to control the second-level cache memory in Pentium-based computers. They can implement write-back cache for multi-processing support. Dual ported buffers and registers allow them to concurrently handle CPU bus, memory bus and internal cache operation for maximum performance. They have the features of synchronous, asynchronous and strobed memory bus operation, selectable bus width and the line sizes, transfers and burst orders. The 82496 and 82497 support 32, 64 and 128-bit wide memory bus; and 16, 32 and 64-byte line sizes. They support 256KB and 512KB cache memory with parity. The 82498 supports 64 and 128-bit wide memory bus; and 32, 64-byte line sizes. Cache memory is arranged in lines of data. The entire line is transferred to or from the memory at a time. The 82498 supports 1M bytes to 2M bytes cache memory. The 82496, 82497 and 82498 cache controllers control 82491, 82492 and 82493 SRAM cache memory respectively. $1\text{KB} = 1 \text{ Kilo Bytes} = 2^{10} \text{ Bytes}$. $1\text{MB} = 1 \text{ Mega Bytes} = 2^{20} \text{ Bytes}$.

In Pentium Pro, Pentium II, Pentium III and Pentium 4, the second-level cache memory and its controller are within the processor itself.

6.3 REAL (OR PHYSICAL) AND VIRTUAL MEMORY

The real or physical memory is the actual main memory provided in the system. It is directly addressed by the CPU. The address of a location of the physical memory is called **physical address**.

The technique which allows a program to use main memory more than what a computer really has is known as *virtual memory technique*. It gives the programmers an illusion that they have main memory available more than what is physically provided in the computer. The entire program and its data are not placed in the main memory. Only the instructions and data which are to be currently executed are brought from the secondary memory into the main memory. The remaining part of the program resides in the secondary memory. When the part of the program which is in the main memory has been executed, it is sent back to the secondary memory. Thereafter another part is brought into the primary memory for its execution. The to and fro movements of the parts of a program between the main memory (physical or real memory) and the secondary memory (hard disks) is called **swapping**. Using swapping technique a program requiring much larger capacity than that of the main memory can be executed. This technique is called virtual memory technique. The virtual memory space is much larger than the physical memory (main memory) space. For example

the 80386 can directly address 4 GB of physical memory. Its virtual memory addressing capacity is 64 TB. The capacity of virtual memory depends on the design aspects of a processor. (Terabytes). $1 \text{ Tera} = 2^{40}$.

The addresses which are used in a program are called *logical addresses*. These addresses indicate the positions of instructions and data in the program. The logical addresses are translated into physical addresses by memory management unit, MMU. The MMU is placed in between the processor and main memory. The CPU generates logical address while executing the program. The MMU receives a logical address from the processor and determines whether the logical address is present in the physical memory (i.e., main memory). If it is present in the main memory, the MMU determines the corresponding physical address. If the logical address is not present in the physical memory, the MMU interrupts CPU. The CPU executes a service routine to bring the required information from the secondary memory to the physical memory. Thereafter the MMU determines the corresponding physical address. The details of MMU will be discussed later on in this chapter. The logical address is also called *virtual address* in a system which has provision for virtual memory.

6.4 SEMICONDUCTOR MEMORY

Semiconductor memories are of two types: RAM (random access memory) and ROM (read only memory). The various kinds of RAMs and ROMs are as described below.

6.4.1 RAM

The read and write memory (R/W memory) of a computer is called RAM. The users can write information into RAM and read information from it. It is accessible to users. The user enters his program and data into RAM. It possesses random access property. In a random access memory any memory location can be accessed in a random manner without going through any other memory location. The access time is same for each memory location. RAM is a volatile memory. The information written into it is retained in it as long as the power supply is on. As soon as the power supply goes off (or interrupted) its stored information is lost. The programmer has to reload his program and data into the RAM when the power supply is resumed.

RAM is not the correct name of R/W memory because both RAM and ROM possess random access property. But R/W memory has been called RAM since long and in practice it has become an established name. Thus R/W memory is commonly called RAM.

There are two important types of RAMs: static RAM and dynamic RAM. Static RAMs retain stored information only as long as the power supply is on. But a dynamic RAM loses its stored information in a very short time (a few milliseconds) even though the power supply is on. In a dynamic RAM a binary state is stored on the gate to source stray capacitor of a transistor of the MOS memory circuit. The presence of charge on the stray capacitor represents 1, and the absence of charge on the stray capacitor represents 0. The charge on these capacitors leak away as a result of normal leakage in a few milliseconds. Therefore, dynamic RAMs have to be refreshed periodically, generally every 2 milliseconds. The dynamic RAMs are cheaper and have high packing density and moderate speed. They consume less power. They are used where large capacity of memory is needed. Static RAMs are costlier and consume more power. They do not need refreshing circuitry. They have higher speed than dynamic RAMs. Static RAM and dynamic RAM are also written as SRAM and DRAM respectively.

Dynamic RAM being cheaper is used for main memory. Static RAM being faster is used in cache memory. Both static and dynamic RAMs use CMOS technology. CMOS devices consume less power. Static RAMs hold information in a flip-flop.

Static RAMs hold information in a flip-flop circuit consisting of two cross-coupled inverters. In a RAM the memory cell must be associated with read and write facility. Six transistors are needed per memory cell in a static RAM. Dynamic RAMs require less number of transistors per memory cell because information is stored on stray capacitors. Only one transistor is needed to form a memory cell of the dynamic RAM. This results in high packing density of the dynamic RAM compared to static RAM.

EDO (Extended Data Output) RAM. In an EDO RAM any memory access (including refresh) stores 256 bytes of data into latches. The latches hold next 256 bytes of information, so that in most programs which are sequentially executed, the data are available without wait states.

SDRAM(Synchronous DRAM) and SGRAM (Synchronous Graphics RAM). These RAM chips use the same clock rate as the CPU uses. As a result the memory chips remain ready to transfer data when the CPU expects them to be ready. They run at the processor-memory bus without imposing wait states. SDRAM is mainstream mass memory choice, and WRAM or SGRAM the high-end graphics memory choice.

DDR SDRAM. The abbreviation DDR is for double data rate. SDRAM transfers data only on the rising edge of the clock, whereas the DDR SDRAM transfers data on both edges of the clock. This results in double data transfer rate. Improved versions of DDR are DDR2 and DDR3. The data transfer rate of DDR3 is 1066 mbps. DDR SDRAM has been developed by the company Samsung. The company Rambus has developed XDR RAM (eXtreme Data Rate dynamic RAM).

RDRAM (Rambus DRAM). It is the fastest semiconductor memory available, 1.6 GB/S. Rambus DRAM (RDRAM) ICs have been developed by the company Rambus. Several RDRAMs are connected through a special bus interface.

Dual-Ported DRAM. This type of RAM allows to access two memory locations simultaneously. It has been provided with two ports for reading data. Sometimes, it is also called video RAM (or VRAM). WRAM (WINDOWS RAM) is a special version of VRAM which is optimized for the types of access which are common in PCs running WINDOWS and WINDOWS applications.

SIMM and DIMM. They stand for Single Inline and Double Inline Memory Modules. These are small printed circuit cards, on which several DRAM memory chips are placed. Such cards are plugged into the system board of the computer. The SIMM circuit cards contain several memory chips with contacts placed on only one edge of the printed circuit card. The complete module is organized as a byte-wide memory module. In DIMM circuit cards memory chips and contacts are placed on both edges of the printed circuit card.

Bit, Nibble and Byte-Oriented Memory Chips

The capacities of memory chips are given in the form of $m \times n$ bits. For example, the capacity of a RAM is $256 \text{ M} \times 1$ bits, it means that its each bit can be addressed individually. One bit can be read from or written into the memory at a time. Such a memory is called **bit-oriented** memory. In a **nibble-oriented** memory four bits i.e., a nibble, are read from or written into the memory at a time. Each nibble of data can be addressed individually. The rating of nibble-oriented memory chip is given as K (or M) \times 4 bits, example $256 \text{ K} \times 4$ bits.

In a **byte-oriented** memory a byte of data can be read from or written into the memory at a time. Each byte of data can be addressed individually. Example is $4M \times 8$ bits. If a parity bit is used, the rating is given as $4M \times 9$ bits; 8 bits are for data and 1 bit for parity.

The memory system of a processor uses a number of memory banks in its memory system. Each bank is byte-oriented. The Pentium processor has 32 address lines, which can address up to 4GB of memory. It contains 64 data lines, and hence its memory system employs 8 numbers of byte-oriented memory banks of DRAM chips (which may be arranged in SIMM or DIMM). The capacity of each bank is 512 MB. The total memory capacity is 4 GB. The Pentium Pro has 36 address lines which can directly address up to 64 GB of memory. It has 64 data lines, and hence, its memory system also employs 8 numbers of byte-oriented memory banks. The capacity of each bank is 4 GB. Total memory capacity is 64 GB.

6.4.2 ROM

ROM stands for "Read Only Memory". It is nonvolatile memory, i.e., the information stored in it is not lost even if the power supply goes off. It is used for permanent storage of information. It also possesses random access property. ROMs are much cheaper compared to RAMs when produced in large volumes. The stored information can only be read from ROMs at the time of operation. Information can not be written into a ROM by the users/programmers. In other words ROMs are not accessible to users. The contents of ROMs are decided by the manufacturers. The contents are permanently stored in a ROM at the time of manufacture. From programming mode point of view the ROMs are placed in broad categories: masked-programmed and user-programmed. ROMs in which contents are written at the time of IC manufacture are called mask-programmed ROMs. PROM, EPROM, E²PROM or any other kind of PROM are user-programmable ROMs. If we simply write or say ROM it means masked programmed ROM because user programmable ROMs are called PROM, EPROM, EEPROM etc.

ROMs store information which is not subject to change. They store fixed programs. In microprocessor-based systems which are used for industrial controls, ROMs store monitor, assembler, debugging package, function tables such as sine, cosine, logarithm, square root, exponential and code conversion tables, etc. An example of a ROM is Toshiba Mask ROM, TCS 534000, 512 K \times 8 bits.

6.4.3 PROM

PROM is a programmable ROM. Its contents are decided by the user. The user can store permanent programs, data or any other kind of information in a PROM. A special equipment called PROM programmer is available for the programming of PROMs. With the help of PROM programmer the user stores his programs in a PROM. PROMs are once programmable, i.e., the user can write his information in a PROM only once. PROMs are more cost effective if small number of chips are to be produced to store certain fixed programs. An example of PROM is 74S287.

6.4.4 EPROM

An EPROM is an erasable PROM. The stored data in EPROMs can be erased by exposing it to high intensity short wave ultraviolet light for about 20 minutes. An UV source of 2537 Å wavelength can be used for the purpose. The technique of erasing contents is not easy and convenient because the EPROM IC has to be removed from the computer for the exposure to the ultraviolet light. When an EPROM is exposed to ultraviolet light the entire data are

erased. The user cannot erase the contents of certain selected memory locations. EPROMs are cheap, reliable and hence they can be used.

EPROMs are used to store programs which are permanent but need updating. The permanent programs which are at the research and development stage also need to be stored in EPROM because they are modified several times. Where updating or changes in the programs at the development stages are required, EPROMs are far more economical than PROMs because they can be reused.

EPROMs employ MOS technology. They store 1 or 0 as the charge or no charge on the insulated floating-gate of the transistor. The UV light causes the stored charge to leak off. Thus the data are erased and it allows the user to reprogram the EPROM. For programming it requires high voltage, 30 V or higher. But its operating voltage, i.e., the voltage for read operation is only 5 V.

Examples of EPROMS are: Intel's 87C257, 256K (32K × 8) CHMOS EPROM; Intel's 27C010, 1M (128K × 8) Byte-wide High-speed CMOS EPROMs; Toshiba, TC571000, 128 K × 8 EPROM. Intel produces EPROMs in the range of 16K to 4M bit. Intel 27C040 is a 4M (512 K × 8) CHMOS EPROM and 27C240 is a 4M(256 K × 16) CHMOS EPROM.

6.4.5 EEPROM (or E²PROM)

EEPROM is an electrically erasable PROM. It is also known as EAPROM (Electrically Alterable PROM). The chip can be erased and reprogrammed on the board easily on a byte by byte basis. Either a single byte or the entire chip can be erased in one operation. It requires much shorter time, a few milliseconds for erasing as compared to 10–20 minutes for EPROM. There is a limit on the number of times the EEPROMs can be reprogrammed, usually 10,000 times. It need not be removed from the computer's board for erasing. Compared to RAMs E²PROMs take much longer time for both writing and erasing a byte. But the access time for reading RAMs, ROMs, PROMs and EEPROMs are comparable.

Intel 2816A is a 16K (2K × 8) EEPROM. The access time for its various versions lies in the range of 200 ns–450 ns. Any byte can be erased in 9 milliseconds without affecting the data in any other byte. Alternatively, the entire memory can be erased in 9 milliseconds. It takes 9–15 ms for writing a byte. The Intel's 2817A is also an 16K (2K × 8) E² PROM. It possesses automatic byte-erase facility before write operation. It takes 20 milliseconds for a combined erase/write operation.

E²PROM requires 21 volt pulses for writing or erasing and 5 V for read operation. Some EEPROM chips require external high voltage pulse for erase/write operation, for example Intel 2816. But some do not require because high voltage pulses for erasing and writing are generated by on-chip circuitry, for example, Intel 2816A.

6.4.6 Flash Memory

It is electrically erasable and programmable permanent type memory. Unlike EEPROM, the flash memory uses one transistor memory cell resulting in high-packing density, low power consumption, lower cost and higher reliability. Its entire contents can be erased in one operation. It is not byte by byte alterable like EEPROM. Some new flash memory chips provide block erasure of its contents. It can be erased and reprogrammed on the system board itself. It has the feature of quick pulse programming. The entire unit or the entire block is erased and reprogrammed as a whole at a time. The name "Flash Memory" has been given due to fast reprogramming capability. It is suitable for storing firmware (codes) whereas

conventional EEPROMs are suitable for storing parameters. When codes are updated the entire codes are rewritten to avoid errors. For updating firmware codes, flash memory is quick and economical as compared to EEPROM. In case of parameter updating they are to be rewritten individually on-line byte by byte. Hence, EEPROMS (which are byte by byte erasable and programmable) are best suited for parameter updating.

Due to low power consumption flash memory is used in battery driven digital devices such as handheld computers, cell phones, digital cameras, MP3 players etc. Flash memory chip upto 1 Gbyte has been developed. In many cases a single flash memory chip may not provide sufficient storage capacity. For such applications larger flash memory modules consisting of a number of flash memory chips have been developed. Flash memory modules upto 4 GBytes capacity are available. The modules are available as flash cards and flash drives. Flash cards can be conveniently plugged into a slot. Flash drives can replace smaller hard disk drives in some applications. They are static and faster but they are costlier.

6.4.7 Nonvolatile RAM

IBM has recently developed a magneto-resistive nonvolatile RAM. It uses magnetization to store information. The capacity of the first magneto-resistive RAM (MRAM) which was a prototype unit, was 16 Mbit.

6.5 MEMORY CONTROLLERS

A dynamic RAM (DRAM) requires refreshing circuitry to refresh its contents periodically every 2 milliseconds or less. For this purpose Intel and some other companies have developed DRAM controllers to refresh several DRAMs in a system. Some of them will be described in this section. Chipsets which are used to interface memory and I/O devices, contain memory controllers. Also, some recent processors contain memory controllers on the processor chip itself. Error detection and correction unit has also been described.

6.5.1 Intel 8203

The Intel 8203 is 64K DRAM controller. It is a 40 pin IC; uses +5 V supply. It provides all signals required to control 64K or 16K DRAMs in a microcomputer system. It is capable of directly addressing and driving up to 64 devices without external drivers. It provides multiplexed addresses, address strobes, refresh logic and refresh/access arbitration. It is fully compatible with 8080A, 8085A, Intel 8088 and Intel 8086 family of microprocessors. It contains refresh timer and refresh counter. It has two modes of operation; one for 64K DRAM and the other for 16K DRAM.

The 8203 can accept the following two types of memory read requests:

1. Normal read via \overline{RD} input.
2. Advanced read, employing S_1 and ALE inputs (the advanced read capability is only for 16K mode).

The advanced read cycles are requested by pulsing ALE when S_1 is active (high). If S_1 is inactive (low), ALE is ignored.

The user can select the desired read request.

Refresh Cycle: Refresh cycles can be started either internally or externally. An on-chip refresh timer generates internal refresh. The REFRQ input is used for external refresh.

6.5.2 Intel 8207

The 8207 is a dual-port DRAM controller. It can interface 16K, 64K and 256K DRAMs. It is a 68-pin IC. It can directly address and drive up to 2 MB without external drivers. A dual-port interface permits two different buses to access memory independently. When used with error detection and correction unit (EDCU), 8206, it provides the necessary logic to directly control 8206. This helps designing of a large error-corrected memory. The combination of 8206 and 8207 provides automatic memory initialization and memory error scrubbing (correcting errors in memory during refresh). When a correctable error is detected during refresh, the RAM refresh cycle is lengthened slightly to allow 8206 to correct error and for the corrected word to be written into the memory. If uncorrected errors are detected during scrubbing, they are ignored. The 8207 is capable of interfacing fast or slow RAMs. Its internal timings are adjusted and optimized as programmed either for fast or slow RAMs.

6.5.3 Intel 82C08

The 82C08 is a CMOS DRAM controller. It is available in 48 pin plastic DIP or 68 pin PLCC. It can interface 64K and 256K DRAMs to Intel and other microprocessors. It can directly address up to 1 MB without external drivers. In case of power failure the 82C08 is capable of performing the task of memory refreshing employing battery backup. The power failure is to be detected by some other device and a signal is to be sent to 82C08. A clock for the operation of 82C08 is to be supplied by the system during power failure.

The status signals of the CPU are connected to the control signals of 82C08. The 82C08 decodes these signals received from the CPU and produces read and write signals it requires. Most of the time CPUs will be able to read a byte or word from DRAMs without any WAIT states being required. If the 82C08 happens to be in middle of a refresh cycle when the CPU attempts to read a DRAM location, the 82C08 will hold its AACK signal high and force the CPU to insert a WAIT state. In such a situation the CPU has to wait one cycle while the 82C08 is completing its refresh cycle before it can access the DRAM.

6.5.4 Error Detection and Correction in RAMs

One of the sources of errors is a noise pulse. In case of DRAM alpha particles and some other radiations also cause errors. Such radiations cause charge on the tiny capacitors to change. These charges on tiny capacitors in DRAM represent data. In large memory chances of errors are more.

One simplest technique of error detection is to use parity bit. When data is written, the parity of a data word is determined. A parity bit is generated such that the overall parity of the data bits plus the parity bit is always odd. The generated parity bit is also written in. When the data word and the parity bit are read out, the parity is checked. If the parity of the result is not odd, it means that there is somewhere an error. The drawback of the simple parity check is that two errors may cancel each other. Another drawback of the simple parity checking is that it does not indicate which bit of the data word is incorrect. More complex error detecting/correcting codes are used in detecting and correcting errors in RAMs. One of such codes is Hamming code. It can detect single-bit and multiple-bit errors and correct at least single-bit errors.

When the data word is written to the memory the several encoding bits are generated by the error detecting and correcting circuitry. The encoding bits are also stored in the memory. In case of Hamming codes the encoding bits for different data bits are different as shown in Table 6.1.

Table 6.1 Encoding Bits

<i>Data Word Bits</i>	<i>Number of Encoding Bits</i>
8	5
16	6
32	7
64	8
80	8

The encoding bits are not simply placed at the one end of the data bit. They are interspersed in the data word. When data words are read from the memory, the encoding bits for the read out data word are recalculated by the detecting and correcting circuitry. These encoding bits are exclusive-NORed with the encoding bits which were stored in the memory. The word obtained from this operation is called *syndrome word*. The value of the syndrome word tells which bit is incorrect in the data word. The correction is made by simply inverting the incorrect bit. There are several different schemes for this purpose from hardware cost consideration. Several ICs are available for automatic detection and correction, such as Intel 8206 etc.

Intel 8206

The Intel 8206 is an error detection and correction unit. It is built in 68-pin grid array package. It uses +5 V single supply. It is a high speed device and uses HMOS technology. It provides error detection and correction for static and dynamic RAMs. It can detect and correct all single bit errors. It is also capable of detecting all double bit and higher multiple bit errors. It interprets odd multiple bit errors as single bit errors, for example 5-bit error. It interprets even multiple bit errors (e.g. 4-bit error) as no error, but detects most of them as double bit error. The 8206-1 takes 35 ns for error detection and 67 ns for correction. The 8206 uses modified Hamming code for error detection and correction. It is able to handle 8 or 16 data bits and 8 check bits. Detection and correction can be provided up to 80 bit of data using 8206 in cascaded connection. Its other features are: ability to handle bytes writes, memory initialization and error logging. It supports read with and without correction, writes, partial (bytes) write and readmodify-writes.

6.6 MAGNETIC MEMORY

Magnetic memory is a permanent type memory. It is not volatile. It is used as secondary and backup memory. Digital data 1 or 0 is stored on the magnetic coating (film) applying electric pulses of suitable polarity to the magnetizing coil of a write head. Fig 6.2 shows a read/write head. The binary digit 1 or 0 depends on the direction of magnetization of a very small area of the magnetic film, which comes under the write head. There are two techniques of data recording: longitudinal recording and vertical recording. In the longitudinal recording the magnetic regions are oriented parallel to the disk surface along the track. In the vertical recording the magnetized regions are vertical to the disk surface. The vertical recording stores many times more data as compared to longitudinal recording. To read data a separate read head, positioned for the convenience close to the write head, is used. The read head is a magnetoresistive (MR) sensor. The electrical resistance of the magnetoresistive material depends on the direction of magnetization of the medium moving under it. By passing current through the MR sensor, resistance changes are detected as voltage signals for read operation.

In low-capacity disks and older hard disks a single write/read head is employed. In other words the same head acts as a write head during write operation, and as a read head during read operation. In case of single head system, during read operation, the movement of the magnetic field relative to the coil of the head, produces electric pulses across the coil. The polarity of the pulses depends the direction of magnetization of a very small area of the magnetic film which was created while recording the information in terms of 1s and 0s. The pulses produced across the coil are utilized for read operation.

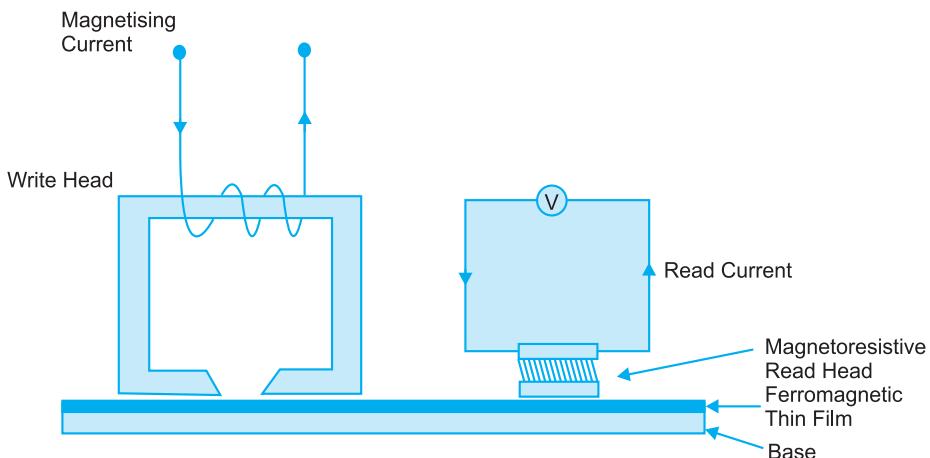


Fig. 6.2 Read/Write head for a magnetic disk.

A modern computer uses the following types of magnetic memory:

- (i) Magnetic Disks : Hard disks and floppy disks,
- (ii) Magnetic tapes.

6.6.1 Magnetic Disks

Magnetic disks include hard disks and floppy disks. Working principle is same for both hard disks and floppy disks. A magnetic disk is a surface device. It stores data on its surface. Its surface is divided into circular concentric tracks, and each track is divided into sectors, as shown in Fig. 6.3. In the configuration of a magnetic disk as shown in Fig. 6.3 (a), all tracks have same number of sectors. The number of bytes stored in each sector is kept same. Therefore, all tracks store the same amount of data. This results higher bit density in inner tracks than that of the outer tracks. Since the same number of bytes is stored in each sector, the size of the inner sectors decides the storage capacity for all other sectors on the disk. Due to this reason some storage space remains unused on the outer tracks. To utilize this unused space, the disk surface is divided into zones. In a zone the number of sectors in each track is same, but it is different than the number of sectors in a track of the other zone. In outer zones the number of sectors per track is more than that in inner zones, as shown in Fig. 6.3(b). Thus storage capacity of the disk is increased by this technique.

Magnetic disks are semirandom devices. A track on a disk is selected in random fashion, but data is written to or read from a sector in serial fashion. The disks are mounted on a rotary drive (D.C. motor) to rotate the disk. After the read/write head is moved to the addressed track, it remains stationary while the disk rotates to bring the starting position of

the addressed sector under the read/write head. Afterwards, the disk continues to rotate, and information is read from or written to the addressed sector. The time required to move the read/write head to the addressed sector is known as **access time**. Access time is the sum of **seek time** and **latency time**. The time required to move the read/write head to the addressed track is called seek time. The latency time (or search time) is the time required to bring the starting position of the addressed sector under the read/write head.

Disk Controller. Magnetic disk drives require controller. The controller converts instructions received from software to electrical signals to operate disks. The functions of a disk controller are:

- (i) To interface a disk drive system to the CPU.
- (ii) Disk drive selection, because a computer uses more than one disk drive.
- (iii) Track and sector selection.
- (iv) To issue commands to the disk drive system to perform read/write operation.
- (v) Data separation.
- (vi) Serial-to-parallel and parallel-to-serial conversion.
- (vii) Error detection, etc.

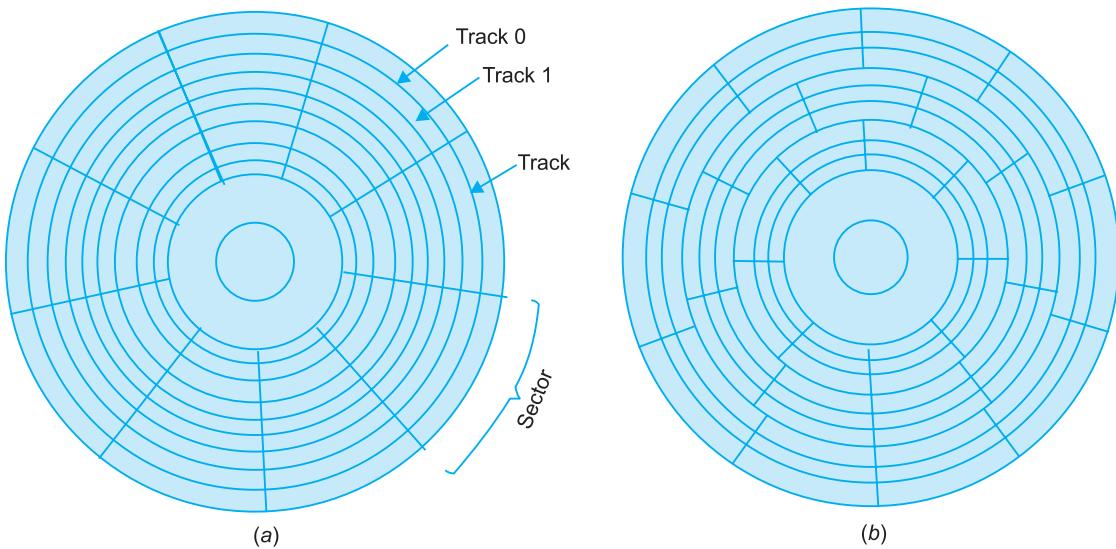


Fig. 6.3 Tracks and Sectors
 (a) All tracks have same number of sectors
 (b) Outer tracks have more sectors than inner tracks.

Data processed by the CPU or stored in the main memory is in the byte form. The bytes to be stored on a magnetic disk must be converted into serial format. Data read from a magnetic disk is in the serial format, which has to be converted to parallel format so that it can be sent to the CPU for processing or to the main memory for storage. Data recorded on a magnetic disk is combined with timing signal (it is done for encoding purpose). Therefore, data read from a magnetic disk must be separated from the timing signal (*i.e.*, clock). Hard disk controller and floppy disk controller are available in IC form. A few of them will be described later on in this chapter.

6.6.2 Hard Disk

Hard disks are on-line storage devices. The term on-line means that the device (hard disks) is permanently connected to the computer system. When computer is on, the device (hard disks) is available to store information or to give information. Hard disks are used as secondary memory for mass (bulk) storage of information permanently. They store programs, data, operating system, compilers, assemblers, application programs, database, etc.

A hard disk is made of aluminium (or other metal or metal alloy) with a thin coating of magnetic material (iron oxide) over it. Standard size of hard disks is 3.5 inch. The 2.5 inch disks are used in portable computers. Hard disks and read/write heads are kept in a sealed, air filtered enclosure. This technique is known as **Winchester** technique. Hard disks together with read/write heads, access mechanism and driving motor constitute an unit called **hard disk drive** unit. The whole unit is fixed. It is not removable. A hard disk is also called a **platter**. It cannot be removed or inserted into a hard disk drive unit. Hard disk drive unit under discussion is of large capacity, and is used as secondary memory. Recently removable hard disks and external hard disks of smaller capacity for backup memory have also been developed, which will be discussed later on.

To increase the storage capacity several hard disks (platters) are mounted on common drive to constitute a disk pack, as shown in Fig. 6.4. The set of all tracks at the same distance from the spindle on the recordable surfaces of the disk pack will lie on the surface of an imaginary **cylinder**. All the tracks which lie on a cylinder have same number. For example, track-10 of each recordable surface of the pack will lie on the cylinder which is numbered as cylinder-10. The disks are addressed by drive number (because more than one drives are used in a computer), cylinder number (*i.e.*, track number), surface number and sector number.

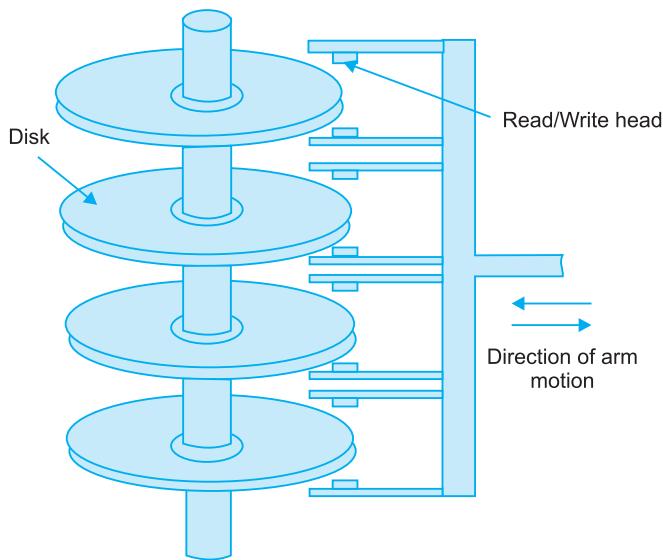


Fig. 6.4 Hard disk pack.

Speed and Capacity. A floppy disk stretch and swell with changes in temperature and humidity. Hard disks do not or at least they do not as much. Due to this fact the read/write heads used for hard disk are much smaller as compared to those for floppy disks. This allows more tracks and sectors on a hard disk resulting in more storage capacity. A hard disk uses

several thousand tracks per surface and hundreds of sectors per track. The capacity of hard disk drives which are used in notebooks, desktop, workstation and servers varies in the range of 40 GB – 750 GB. 1 GB = 1 Giga bytes = 2^{30} bytes. The access time is in the range of 5 ms–10 ms (millisecond). The capacity of single a disk upto 133 GB is possible. A 400 GB hard drive of Seagate company uses only 3 disks. The speed of hard disks varies from 5400 rpm to 15000 rpm. The large capacity disk drives provide SATA (serial ATA) interface. Seagate's 500 GB hard disk drive provides SATA interface having 3 GB/s data transfer rate. The hard disk drives of smaller capacity (1 GB–5 GB) are also available for smaller system. Such microdrives are used as external drive. They are connected to the computer through USB 2.0 interface. The diameter of micro hard disks may be of 1 inch to 2.5 inch, depending upon their capacity. Micro drives are also used in some digital equipment. Recently, 2.5 inch hard disk drives of 40 GB, 80 GB and 150 GB have also been developed.

Disk Cache. Semiconductor RAM memory is much faster than magnetic disk memory. Hence, disk access is much slower than that of the main memory which is semiconductor DRAMs. To solve the problem of mismatch in data transfer rate, a semiconductor memory, called disk cache is placed in between the main memory and hard disk memory. The capacity of the disk cache is a few megabytes. The disk cache stores information which are needed for the execution of the current program. It enhances the performance of the hard disk drive. Other mass storage devices may also be provided with this type of cache memory.

RAID System. In Redundant Arrays of Independent Disks (RAID) system multiple disks operate in parallel and store the same information. It improves storage reliability. It eliminates the risk of data loss when one of the multiple disk fails. Furthermore, a large file is stored in several disk units by breaking the file up into a number of smaller pieces and storing these pieces on different disks. This is called data stripping. When file is accessed for read operation, all disks deliver data in parallel. Thus total file transfer time is reduced. To improve reliability of storage system, some information can also be stored on additional disk units. Such system is economical on a large computer system.

Air Cushion between Head and Hard Disk. The high speed of hard disks creates a thin air cushion of about 10 microinch between the rotating disk and the head. When disk is rotating the head floats on air cushion and remains 10 microinch off the disk surface. The head does not make any physical contact with the disk surface. Hard disk must be kept in dust free environment because the diameter of the dust particle may be as many as 10 times the thickness of the air cushion. Any dust particles or any other contamination present in the sealed container, will cause the heads to crash into the disks. It is known as **head crash**. In head crash the coating of the magnetic material on the disk surface is damaged where head strikes. The data of that portion of disk surface is lost.

Removable and External Hard Disk. Removable hard disks have also been developed. They are used as backup memory. They can use either EIDE or SCSI interface. A docking bay can be added to PC for removable hard disk. The docking bay connects internally to PC's power supply and also to either an EIDE or a SCSI host adapter. Examples of removable hard disks are: SyQuest syjet 1.5 GB, Iomega Jaz 1 GB and 2 GB, etc. External hard disk drives of 1 GB–30 GB have been developed, which can be interfaced to IEEE 1394 serial bus or USB 2.0.

IDE, SCSI, ATA and ATAPI. Actually IDE and SCSI are not controllers, rather they are host adapters. The controllers are built into the drive unit itself. IDE stands for

Integrated Drive (or Device) Electronic, and SCSI stands for Small Computer System Interface. It is a high-speed, intelligent pathway to connect peripherals to the computer. IDE is a standard according to which IDE interface is made.

The enhanced IDE (EIDE) standard is available. It is also called improved IDE or new IDE. EIDE is almost universal on new PCs. It can interface hard disk drive, floppy disk drive, optical disk drive and tape drive. It provides up to four channels. Two EIDE devices can be connected to each channel. Thus a total of up to 8 EIDE devices can be interfaced to a PC. No commercial motherboards support more than two channels, but one can add plug-in cards on the ISA bus to support one or two additional channels. IDE or EIDE drives offer great ease of use, one can just plug them in to work.

A PC motherboard has two connectors for EIDE interface, as shown in Fig 6.5. From each connector one flat cable runs to provide an EIDE channel. Two devices can be connected to each channel. For each pair of devices on the same cable, one is set as a master device and the other as a slave device. It does not matter which one (master or slave) is at the end of the cable.

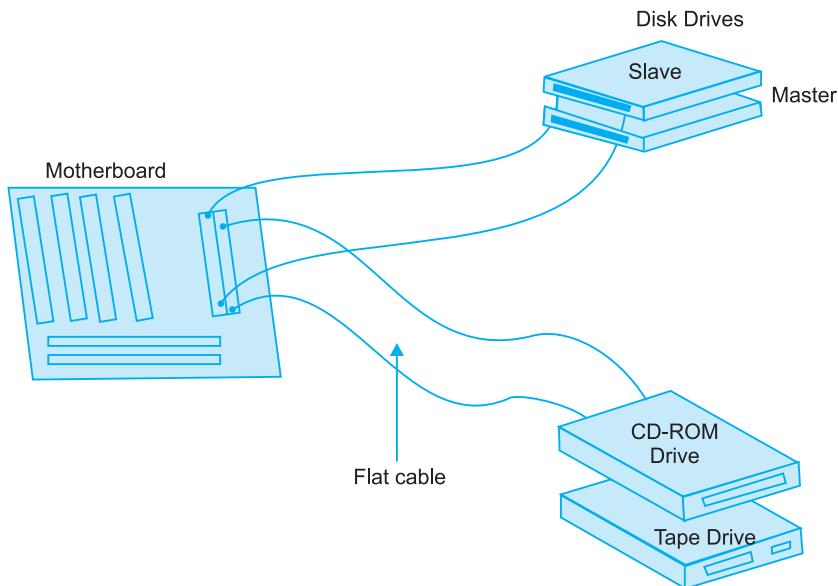


Fig. 6.5 EIDE interface.

ATA and **ATAPI** are also the names of standards. ATA is AT Attachment, and ATAPI is AT Attachment Packet Interface. ATAPI is an extension of ATA. ATA and ATAPI are specifications for EIDE. These are published by Small Form Factor (SFF) Committee. Data transfer rate of IDE or EIDE is much less than that of SCSI. A typical value is 5.5 MB/s.

SCSI. It is an intelligent adapter. It mediates between the activity on the SCSI bus and that on the PCs I/O bus. SCSI has a separate I/O bus called SCSI bus. Some PCs have SCSI host adapters built into their motherboards, but much more common practice is to use an optional circuitry on a plug-in card. As the data transfer rate for SCSI interface is high, the SCSI host adapter is connected to a fast input/output bus. Today the most popular choice is PCI bus. The latest version of SCSI called SCSI-3 can connect up to 15 devices. Each device connected to SCSI bus is assigned an identification number. The highest ID (identification) number is used by the host adapter. SCSI devices may be hard disk drive, optical disk drive,

ZIP drives, printer, image scanner, graphics tablet, mouse, etc. The data transfer rate for SCSI-3 is 80 MB/s. Even higher data transfer rate can be achieved using optical fiber link. Western Digital has announced ULTRA 160 SCSI with data transfer rate of 160 MB/s.

Fig. 6.6 shows how SCSI devices are connected to SCSI bus (it is known as daisy-chain connection). SCSI bus consists of 50 wires. A flat cable runs from the SCSI controller card to the SCSI devices. The cable is terminated at the end i.e., the last device on the daisy chain must have a terminator installed on the connector that is not being used. The SCSI controller sends ID number of the device it wants to communicate.

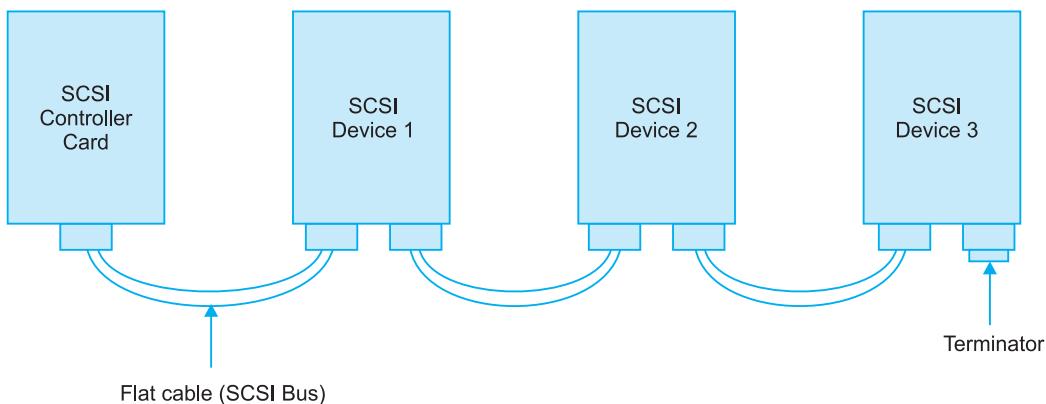


Fig. 6.6 SCSI interface.

SCSI devices have their own controllers and SCSI interface. The controllers are single-chip microcomputer. Hard disks and floppy disks are very important devices. The functions of their controllers have already been discussed in the Section 6.6.1 under the subheading "Magnetic Disk Controllers". SCSI devices can be moved from a host adapter of one computer to another, except hard disk drive with data. It is treated as a new disk. This problem is not there in case of CD-ROMs, CD-R and removable hard disks. In these cases the device driver software which supports their use has been provided with features needed for such movement. SCSI system has high rate of data transfer and it is a costly system. It is used on servers and other powerful computer. Desktop computers use, IDE or EIDE interface.

6.6.3 Floppy Disk

Floppy disks are made of Mylar (a plastic material) coated with magnetic material (iron oxide or barium ferrite). The disk is not a hard plate, rather it is very thin piece of flexible plastic, and hence, it is called floppy disk. Floppy disks are also called **diskette**. They are removable disks. A floppy disk is inserted into the computer system when needed. It is an inexpensive storage device and it is used as backup memory. Users also store their programs, documents and other important information on floppy disks. The size of floppy disks is 3.5 inch diameter. A floppy disk rotates at 360 rpm. Its average access time is 150–250 ms.

3.5 Inch Floppy Disks

Fig. 6.7 shows the schematic diagram of a 3.5 inch floppy disk. On a 3.5 inch floppy disk the read/write head window is covered with a sliding metal cover i.e., a metal shield. The disk drive pushes the cover aside when a disk is inserted into the drive unit. There is a hole at the centre for the drive spindle. There is a write protect notch to protect the stored information.

It uses a sliding plastic button, which can slide to either of the two positions. One position allows reading, writing and erasing. The other position allows only reading. The button slides in a track. When light is visible through the track, then only the disk is write protected. The protective cover of a 3.5 inch floppy is hard plastic. It is permanently sealed, and contains lubricants and cleaning agents that prolong the life of the disk. The storing capacity of a 3.5-inch high density floppy is 1.44 MB. It uses 80 tracks per surface, 135 TPI (tracks per inch), 18 sectors per track, 360 rpm and stores 512 bytes per sector. Both sides of the disk store information. The data transfer rate is about 40 KB/s. With the advent of optical disks, the use of floppy disks is diminishing day by day.

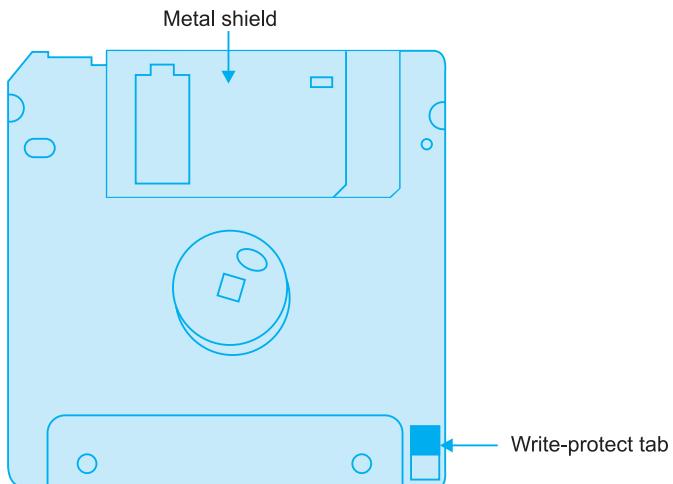


Fig. 6.7 3.5 inch floppy disk.

Cautions in Handling Floppy Disks

A floppy disk should always be kept in a plastic box when it is not in use. It should be kept in vertical position. One should not touch the exposed area of the disk surface with fingers. Sharp and hard-point pen such as a dot pen (ball-point-pen) should not be used for writing label. A felt-tipped pen such as an art pen should be used for writing label. Floppy disks should be kept away from magnetic fields such as that created by televisions, tape recorders, electric radios, electric motors, telephones and other electrical appliances. Heat, moisture and dust also damage floppy disks, and therefore, they should be kept away from excessive heat produced by direct sun or any other radiation. They should be kept in dust and moisture free environment. A disk should not be removed while drive is running. The disk should not be bent.

Floppy Disk Controllers

The function of a magnetic disk controller has already been described in Section 6.6.1 under the subheading “Magnetic Disk Controllers”. Intel has developed 82078, 82077AA and 82077SL floppy disk controllers. The 82077AA, is a CHMOS single-chip floppy disk controller. It can drive up to four floppy disks. It can support both tape and floppy disk drives. It has been provided with integrated tape drive support. It has capacity to interface directly perpendicular recording 4 MB floppy drives. The 82077SL, is a superset of 82077AA with additional feature of power management that makes it suitable for portable computer. The 82078, is an enhanced floppy disk controller. It supports standard 5 V as well low-voltage 3 V system. Its

data transfer rate is 2Mbps, which is double of the data rate of earlier versions. It is provided with enhanced power management.

6.6.4 Formatting of Magnetic Disks

Magnetic disks are specially prepared before they are used first time. The special preparation is known as **formatting** of the disk. When a new blank disk is formatted the tracks and sectors are marked on the disk. Each sector is numbered, going round each track, from outermost track. In addition to data, several other informations are also recorded on the disk for correct data transfer between the disk and the main memory. The outermost tracks contain disk's main directory, information about disk space allocation, etc. In the beginning of each sector there are some bytes of information, which are used by the disk drive system. These are address of the sector. It indicates where on the disk this sector is located. Then comes the data section of the sector, which is 512 bytes of space to record data. Each sector has a few bytes space to record CRC value or ECC bytes.

6.6.5 Error Checking in Magnetic Disk Memory

When data are read from a magnetic disk, it is checked whether the data are the same, which were written. Error detecting codes are used to detect errors. The Cyclic Redundancy Checking (CRC) method is most popular and widely used. In this method all data bytes of a sector of a track are treated as a single large number. This binary number is divided by a constant number. The 16-bit remainder is recorded as CRC bytes after the data bytes. When data are read, the data bytes as well CRC bytes are read out. The CRC bytes are subtracted from the read out data. The result is divided by the original constant. If the remainder is zero, the read out data is correct; otherwise data read out is not correct. If an error is found in CRC checking, the checking is repeated a few times. In case of an error, the disk controller and operating system declare a failure of the sector, and prevent users from using damaged data.

Floppy disks use CRC technique for error checking Hard disks use ECC (Error Correction Codes) for error detection and correction. ECC allows not only to detect errors, but also indicates where and what the error is. Each bit must be either 0 or 1. If one knows exactly which bits are wrong, they can be corrected by reversing those individual bits. Error detection/correction is done by the controller.

6.6.7 Magnetic Tapes

Magnetic tapes are used for backup memory. They are sequential access device whereas a disk drive is a direct access device. In disk drive system the head moves to the position of the desired record. But in the case of tape drive system, the head moves sequentially. It has to move through the adjacent records until it reaches the desired record. However, recently Exabyte company has developed a technology using which the heads can read data from the physical location on the tape, without having to follow tracks from beginning to end. This breakthrough gives a boost to reliability and data retrieval. The magnetic tape is made of flexible polyester coated with magnetizable material. Tape width varies from 3 mm to 12.7 mm. Most of the tapes are available in cartridge form. Tape drives may contain multiple cartridges. The capacity varies from a few hundred megabytes to a few hundred gigabytes. The manufacturing companies are: IBM, HP, Quantum, Sony, Exabyte, etc. Some trade names of tapes are: LTO-Ultrium (LTO is Linear Tape Open), DLT (Digital Linear Tape), DDS (Digital Data Storage), DAT (Digital Audio Tape), AIT (Advanced Intelligent Tape) etc. Some

typical capacities are: 20 GB (40 GB compressed), 40 GB (140 GB compressed), 80 GB (160 GB compressed), 100 GB, 200 GB, 400 GB (800 GB compressed) etc.

Modern magnetic tapes contain a number of parallel tracks running lengthwise. Earlier tapes used 9 tracks. Modern tapes use 18 or 36 tracks. A 9 track system stores 1 byte of data on 8 tracks and a parity bit on the 9th track. In case of 18 and 36 track systems 2 bytes and 4 bytes of data and corresponding parity bits are stored respectively. If the recording of data on all the tracks is done simultaneously using multiple heads, it is called parallel recording. But modern tape system use serial recording. A typical serial recording called serpentine recording is used. In this technique the first set of bits are recorded on a track lengthwise. When the end of the tape is reached, the heads are repositioned on the next track. Now the recording is done in the reverse direction. In this way recording continues back and fourth, until the tape becomes full. To increase the speed, reading/writing can be performed on a number of adjacent tracks (typically 2 to 8 tracks) simultaneously. Data are still recorded serially on the individual tracks. But the blocks in sequence are stored on the adjacent tracks, as shown in Fig. 6.8.

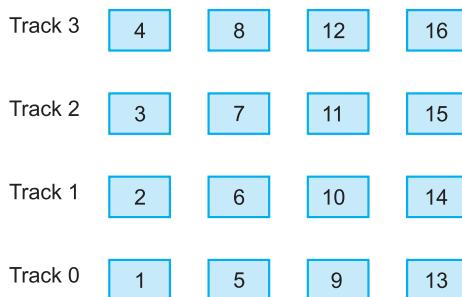


Fig. 6.8 Block Layout for a tape system to record 4 tracks simultaneously.

The data on the tape is organized in the form of records separated by gaps. A gap between two consecutive records is called **inter-block gap (IBG)**.

A half-inch magnetic tape is divided into horizontal rows called **tracks** or **channel** and vertical columns called **frames**. For parallel recording a 9-track tape drive system is provided with 9 read/write heads. 9 bit are written/read in a vertical column simultaneously. 8 bits are for the binary code of a character and the 9th bit is the parity bit for error checking. A parity bit detects only single-bit errors. It cannot detect multibit errors. To detect multibit errors a second parity bit called **longitudinal parity bit** is introduced. After recording character bits in a number of columns, one column is used to record longitudinal parity bits. Using both vertical and longitudinal parity check bits, both types of errors; single-bit and multi-bit errors are detected and corrected.

When a number of records are stored in a tape an interrecord gap is provided between two adjacent records. Some portion of the tape at one end is left unrecorded for threading purpose. A reflective marker called *load point* shows the beginning of the usable tape. There is also a similar marker at the end of the usable tape, known as *end of reel marker*. At the beginning of the tape, after the load point marker, there is a *header control label* to identify the tape contents and to store other control information. At the end of the tape there is a transfer control label, which contains the number of records in the file. There is also a special one-character record to signal the end of a file. All these labels and markers are shown in Fig. 6.9 (a).

If the records are of smaller lengths, due to presence of interrecord gaps only a small percentage of the tape is utilized. To improve the efficiency of the tape utilization a number of records can be grouped together in a *block* and block is recorded as single unit on the tape. There are suitable programs to separate records within a block for processing.

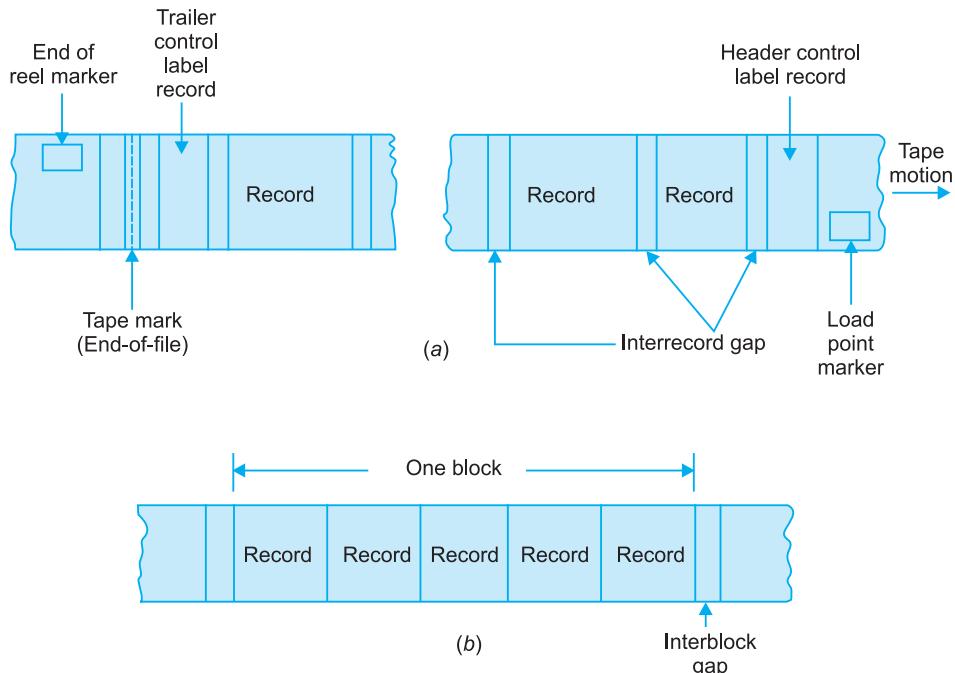


Fig. 6.9 Magnetic tape (a) markers and labels, (b) multiple record block.

6.7 OPTICAL DISKS

Optical disks are used for backup memory. Information is written to or read from an optical disk using laser beam. It has very high storing capacity as compared to magnetic floppy disks. With the advent of optical disks, the use of magnetic floppy disk is diminishing day by day. The capacity of optical disks varies from 650 MB to 17 GB. DVDs of 15, 25, 30 and 50 GB capacity are expected soon. It has very long life. An optical disk is a direct access device. It can be randomly accessed for music, video, a file or any other information. It is costlier than magnetic floppy disk, but it is not expensive. As its read/write head does not touch the disk surface, there is no disk wear and problem of head crash. Elaborate error checking codes can be used as there is no problem of space because of its high storage capacity. The greatest drawback of an optical disk drive system is its large access time as compared to magnetic hard disk drive. In case of a hard disk drives system the read/write head is a tiny magnet whereas in an optical disk system the drive has to move on a sizable optical assembly across the disk surface. This results in an increased access time. There are the following types of optical disks:

- (i) CD (Compact Disk) or CD-ROM
- (ii) CD-R (CD Recordable)
- (iii) CD-RW (CD-Read/Write)
- (iv) DVD (Digital Versatile Disk)-ROM

- (v) DVD-R
- (vi) DVD-RW

The diameter of CD disks are 12 cms. DVDs come in 8 cm diameter as well as 12 cm diameter. The capacity of CDs may be 650 MB or 700 MB. The capacity of DVDs is 4.7 GB, 8.5 GB or 17 GB. The information written on optical disks are stored permanently.

6.7.1 CD Technology

The bottom layer of a CD disk is polycarbonate plastic. It acts as a clear glass base. The surface of the polycarbonate substrate is coated with a reflecting material usually aluminium. The aluminium surface is then coated with a protective acrylic layer. Finally, there is the topmost layer to have label. Fig 6.10 shows the different layers of a CD. The CD disk has a single spiral track for recording data. The spiral track can store more data as compared to concentric circular tracks. To write information in terms of 1s and 0s laser beam is used. To write 1 the laser beam is turned on, which burns a **pit** upto the reflecting layer. The laser beam is focussed from polycarbonate surface side. The laser source and the photodetector are positioned below the polycarbonate plastic. The emitted laser beam travels through the polycarbonate plastic, reflected back from the aluminium layer and comes back towards photodetector. To write 0, the laser beam is not turned on and hence, no pit is burnt. The surface where there is no pit is called **land**.



Fig. 6.10 Different Layers of a CD-ROM or CD

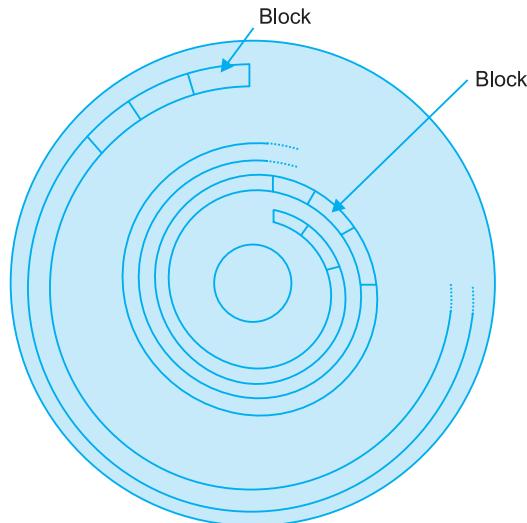


Fig. 6.11 Spiral track of an optical disk.

Fig. 6.11 shows a spiral track of an optical disk. The track is divided into sectors of the same size as shown in the figure. The outside sectors near the outer edge of the disk are of

the same length as that of the inside sectors. The pits and lands are read by the laser at constant linear velocity. In other word the information is read at the same constant rate. An optical disk is rotated at a variable rotational speed so that pits/lands are read by the laser beam at constant linear speed. The disk rotates more slowly near the outer edge than near the center.

6.7.2 CD-ROM and Audio CD

From audio CD and CD-ROM prerecorded information are read out. Both use similar technology. If CD stores digital audio and video data, some errors in data can be tolerated because they are unlikely to affect the reproduced sound and image in a perceptible way. But if a CD stores data for computer applications, errors are not acceptable. Therefore, CD meant for computer applications store additional bits also, for error detection and correction. CD-ROM players/computers also have to store error detection and correction programs. CD which stores computer data are called CD-ROM. The CDs which are used for audio/video are called audio CD/video CD.

The capacity of a CD-ROM is 650 MB, diameter 5.25 inch and speed 360 r.p.m. Its data transfer rate is nx , where $x = 150 \text{ KB/S}$ and n is a factor. If a CD-ROM has specified speed $52x$, its data transfer rate is $52 \times 150 \text{ KB/S} = 7.8 \text{ MB/S}$. 650 MB can store about 3 lacs pages of printed text. A 650 MB audio CD can store music of 75 minutes. CD-ROMs are widely used for application programs, large texts, video games, distribution of softwares, data bases, etc.

To write data on CDs, high-intensity laser is used whereas to read data, laser beam of less intensity is employed. CDs are mass manufactured. Information on read-only CDs are recorded at the time of manufacture. Data are recorded using high-power laser beam and a master disk is prepared. Then a die is made from the master disk. From the die copies of the master disk are produced on mass scale.

6.7.3 CD-R (CD-Recordable)

CD-R is used when a single copy or a few copies of a set of data is needed. It is a write-once read-many times (WORM) type CD. The user can write data on CD-R only once, as it is done in case of a semiconductor PROM. A CD-R has a spiral track on the disk. There is a photosensitive organic dye on the track. To write 1s, laser beam is focused to create pits into the organic dye on the track. To write 0s, the laser beam is not turned on. In this case land is formed. CD-R is used for the storage of documents and files. The written data are stored permanently. Different layers of a CD-R are shown in Fig. 6.12. The storing capacity is 700 MB.

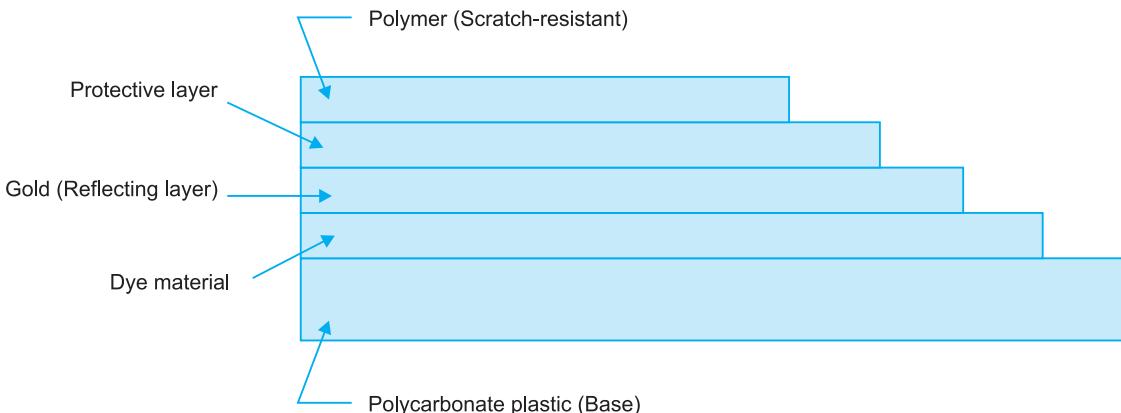


Fig. 6.12 Different Layers of a CD-R

6.7.4 CD-RW (CD-Rewritable)

CD-RW is a read/write optical disk. The user can erase the recorded data of a CD-RW and rewrite new data in that space. The capacity of a CD-RW is 700 MB. Now people use CD-RW instead of magnetic floppy disk as it has high storage capacity, more reliable and has longer life. The construction of a CD-RW optical disk is similar to that of a CD-R. Instead of an organic dye in the recording layer the CD-RW contains an alloy of silver, indium, antimony and tellurium. This alloy has two states: crystalline state and amorphous (non-crystalline) state. The crystalline state allows light to pass through it. The amorphous state absorbs light. Using laser beam the structure of the alloy material can be changed from one state to another. This technology is called **phase change technology**. In normal state the alloy has crystalline nature. When the alloy is heated above 500 degree C using high-power laser beam and then cooled down, the alloy becomes amorphous. This state reflects light poorly and corresponds to a pit to represent bit 1. The Crystalline state represents 0 bit. To erase data, the alloy is heated to about 200 degree C and this temperature is maintained for an extended period. This process is called annealing. The annealing process brings amorphous material back to crystalline state. The CD-RW drive employs laser beam of three different power. A high-power laser beam is used to record pits to represent 1s. The middle-power laser beam is used to erase data and to bring material back to crystalline state. A low-power laser beam is used to read information from the disk. A CD-RW drive is also capable to read CD-ROM and read/write CD-R. The CD-RW drives provide EIDE, SCSI and USB interface. With the advent of CD-RW, the use of magnetic floppy disk is diminishing day by day. CD-RWs are used for backup purposes. As CD-RWs have made CD-Rs less important as they are superior to CD-Rs.

6.7.5 DVD (Digital Versatile Disks)

DVDs have much more storage capacity as compared to CDs. They use laser beam of shorter wavelength as compared to the wavelength of laser beam used in CDs. Pits in DVDs are much smaller and tracks are placed closer together. This gives 4.7 GB storage capacity for a single layer single sided DVD. Double layer single sided disk has a capacity of 8.5 GB and double layer double sided disks have capacity 17 GB.

The structure of a single layer single sided DVD is almost the same as that for a CD-ROM shown in Fig. 6.10. The data transfer rate of DVDs is nx , where $x = 1.38 \text{ MB/S}$ and n is a factor. The access time of DVDs is also low and similar to CDs. But the data transfer rate is much higher than that of CDs, because of higher density of pits.

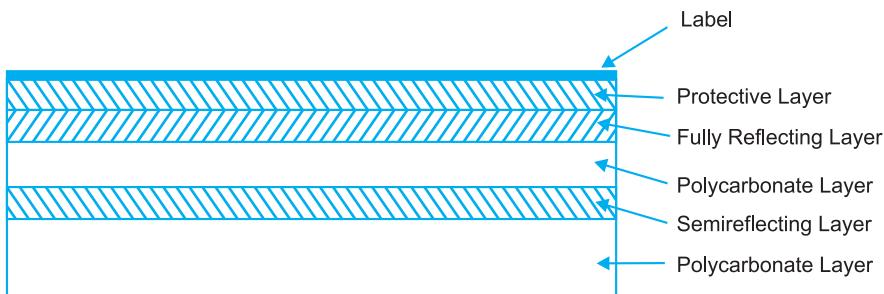


Fig. 6.13 Dual Layer DVD

Fig. 6.13 shows a dual layer DVD. In dual layer disk, the first layer contains a translucent material (semireflector) instead of aluminium reflecting aluminium material. A fully reflecting material is provided at the top of the second layer. By focussing the laser beam on

the desired layer, the information can be read. When the laser beam is focused on the first layer, sufficient light is reflected back by the semireflecting (translucent) layer to detect pits and lands corresponding to the data stored on this layer. When the laser beam is focussed on the second layer, the fully reflecting layer, which is at the top of the second layer, reflects the laser beam to detect the pits and lands corresponding to the stored data on this layer. In both cases, the layer on which the laser beam is not focussed, reflects much less amount of laser beam, which is filtered by detecting electronic circuitry.

The next generation of DVDs will have two competing formats: Blue-ray and HD-DVD. The Blue-ray consortium is headed by Sony. Blue-ray DVDs of 25 GB and 50 GB capacity are expected soon. The HD-DVD consortium is Toshiba. HD-DVDs of 15 GB and 30 GB are expected very soon. These disks will be less costly than Blue-ray disks. HD-DVDs will soon be produced at near DVDs prices.

There are the following types of DVDs:

- (i) DVD-ROM
- (ii) DVD-R
- (iii) DVD-RW

DVD-ROM. The structure of a DVD-ROM is similar to that of a CD-ROM as shown in Fig. 6.10. Its bottom layer is polycarbonate plastic. It forms the base of the disk. The next layer is a reflecting layer, usually aluminium. Then there is a protective acrilic layer. The topmost layer is for label. The capacity of a single layer single sided disk is 4.7 GB, double layer single sided disk 8.5 GB and dual layer double sided disk 17 GB. DVDs of 15, 25, 30 and 50 GB capacity are expected soon.

DVD-R or DVD-Writable. The structure of a DVD-R is similar to that of a CD-R. Its different material layers are as shown in Fig. 6.12. Pits are formed in the dye material. Users can write information on a DVD-R only once and read as many times as they want. It is used where a single copy or a few copies of information is desired. The capacity of a CD-R is 4.7 GB.

DVD-RW or DVD-RAM or DVD-Rewritable. The structure of a DVD-RW is similar to that of a CD-RW. Instead of an organic dye material, its recording layer is an alloy. The alloy consists of silver, indium, antimony and tellurium. It uses **phase change technology** for data recording. This technology has already been explained while discussing CD-RW. The capacity of a DVD-RW is 4.7 GB and 9.4 GB. CD-RWs are used as backup memory to store a large amount of data. DVD-RW uses laser beam of different power for writing, erasing and reading. High power laser beam for writing, middle power laser for erasing and low power laser for reading are used. EIDI, SCSI and USB interface can be used for a DVD-RW.

6.8 CCD (CHARGE COUPLED DEVICES)

CCD is a semiconductor memory device. It is fabricated in the form of a long shift register to act as serial shift register. Extremely high density shift register and serial memories can be constructed with CCDs. When light falls, charge is created proportional to light intensity. The charge is converted to digital signal by analog to digital converter. Because serial memories have limited use as compared to RAMs. CCDs are not widely used memories in digital computers. Applications of CCDs, however, are encountered in image processing and digital signal processing system, where high density serial nature of the device is desirable feature. CCD memory is used for computer's vision, in digital camera, in robots, etc.

6.9 DIRECT ACCESS STORAGE DEVICES (DASDs)

The online secondary storage which is always available to a processor is called direct access storage device (DASD). The commonly used DASDs are magnetic disks. The optical disks, if incorporated in a system are also DASDs. This term DASD is in the context of permanent type directly accessible long term storage device (*i.e.*, directly accessible secondary storage). As far as the term direct access is concerned all semiconductor memories RAMs, ROMs, PROMs are directly accessible and have random access property. Out of these RAMs are for users and they have been categorized as primary memory. In the literature DASDs are also called random access memory. Earlier, we have discussed random access property with respect to memory locations. From memory locations point of view a disk memory is semirandom memory whereas RAMs, ROMs and PROMs are random access memory. There is another point of consideration, *i.e.*, file storage consideration. Files can be stored/read on random basis on disks. But they cannot be stored on random basis on a magnetic tape. From this consideration disks are *random access memory* and tapes are *sequential memory*. In case of a magnetic tape the entire tape has to be read and processed to update the sequentially organized records in a file. The tape is a offline device. Recently some tape drives have provision to access files directly.

A disk can also be used to record files of sequential records. If files are to be processed sequentially, it can be stored on a disk in that very order. In such a case, the first file is accessed on random basis but all other files are accessed sequentially. The random access capability of disks is then not utilized.

6.10 DESTRUCTIVE AND NONDESTRUCTIVE READOUT

If in the reading operation the contents of a memory is destroyed, it is called *destructive readout*. In case of destructive readout memory every read operation must be followed by a write operation. The dynamic RAM is an example of destructive readout memory. If the reading operation does not change the contents of a memory, it is called *nondestructive readout*. Examples of nondestructive readout memory are: static RAM, magnetic disk, magnetic tape etc.

6.11 PROGRAM AND DATA MEMORY

These terms are used in the context with single chip microcomputers, *i.e.*, microcontrollers which are used for dedicated applications. In industrial applications or some other dedicated applications the programs are usually fixed, already developed in the laboratory or industry, and tested for the desired applications. These fixed programs are written into a PROM, EPROM or flash memory. The memory of a microcontroller, which stores programs is called *program memory*. PROM or EPROM or flash memory is used as a program memory in a microcontroller to store the fixed program. The memory of a microcontroller, which stores data during processing, is called data memory. A RAM is used as a data memory in a microcontroller to store data. In some cases EPROM is within the microcontroller IC. In some cases there is no EPROM in the IC; an external EPROM is used. But RAM is within the IC.

6.12 MEMORY MANAGEMENT UNIT (MMU)

The function of a memory management unit is to convert logical memory address into actual physical (real) memory address and provide protection. The MMU is a hardware

component placed in between the processor and the main memory. Some modern CPUs contain on-chip MMU such as 80286, 80386, 80486, Pentium family of processors, 68030, 68040, 68060, powerPCs etc. The CPU which does not contain on-chip MMU uses external MMU. The MMU is used in a multiprogramming or multiuser system. The tasks which are performed by MMU can be performed by the operating system. Then the operating system will not get sufficient time for program execution and its other tasks. The MMU reduces the burden of the operating system. The memory addresses used in a program are *logical addresses*. They indicate the logical position of instructions and data in the program. A logical address is the location of an instruction or data relative to the beginning of the program. For example, the instruction JNZ LOOP. The label LOOP represents a logical address to which the program will jump if the zero flag is not set. When a program is compiled each logical address is represented by a memory pointer consisting of two components. In a segment oriented system the two components are: segment selector and offset. In a page oriented system the two components are: page address and page offset. In other words a logical address is represented by a page number and a page offset (*i.e.*, a relative address within the page).

The modern MMU provides the facility of *virtual memory* to provide very large memory space to users. Virtual memory is a technique which allows programmers to use more memory than what a computer actually has. The main memory which is actually present in a computer system is known as *physical* or *real memory*. The processor executes a program which resides in the main memory. By virtual memory technique a programmer can make a program which uses more memory than the actual capacity of the main memory. His program is stored in the secondary memory, usually disks. The MMU transfers a part of the program from the secondary memory to the main memory, which is currently needed. Similarly, it sends back those parts of the program from the main memory to the secondary memory, which are not being currently used. This to and fro movement of instructions and data between main and secondary memory is known as *swapping*.

To explain how MMU manages more memory than the actual capacity of the main memory, we shall take the example of 80286. It has 24 address lines. With 24 address lines only 16 MB memory can be addressed. But its virtual memory capacity is 1 GB. In its virtual mode of operation the memory pointer consists of 16-bit segment selector and a 16-bit offset. The segment selector has 14 address bits and 2 privilege-level bits. The segment selector specifies an index into the memory resident table (descriptor table) called *descriptor*. The descriptor gives a 24-bit segment address. The physical address is obtained by adding the 16-bit offset to the 24-bit segment address. This computation of physical address from a logical address is carried out by MMU.

With the help of 14 address bits the segment selector can select any one of 16384 (2^{14}) descriptors. As the offset component of the logical address is of 16-bits, each segment will contain 64 KB. As there is one descriptor for each segment, the total logical address will be 64 KB \times 16384, about 1 GB. Thus the MMU can manage 1 GB of memory by virtual memory technique. Now the question is "How are 1 GB addresses managed with only 16 MB of real memory?" This is managed by keeping only the currently used segments in the physical memory.

When the CPU sends a logical address to MMU, it checks whether the segment containing this logical address is present in the physical memory. If the segment is present in the physical memory, the MMU calculates physical address corresponding to the supplied logical address. When the segment corresponding to the supplied logical address is not present in the physical memory, the MMU interrupts the CPU. On receipt of an interrupt from MMU, the CPU reads the desired code or data segment from the disk (*i.e.*, secondary memory). The

MMU then calculates the required physical address. 1 GB logical addresses (in case of 80286) available to users constitute *virtual memory* and corresponding logical addresses are called *virtual addresses*. The term virtual refers to something which appears to be present but actually it is not.

The advantage of a segmented memory is that it offers a simple handling technique for growing data structures. When it is not known how large a particular data structure will become, it is not necessary to make a guess. The data structure can be assigned its own segment. The operating system will expand or shrink it as required. Another advantage is that a few descriptors are required for large programs or data spaces. In segment oriented system the size of the segments which are swapped in and out of physical memory is quite large. Their loading into the memory is a time consuming process. The paging technique overcomes this difficulty.

6.12.1 Demand-Paged Virtual Memory Scheme

In page oriented memory, the memory is divided into pages. A page has a fixed length, 4 KB or 4 MB length. The logical address is represented by the page address and the page offset. The page address points to a descriptor table. The page address is just like the selector in the case of memory segment scheme. The function of a descriptor is same as that in case of a memory segment scheme. When the demanded page is not present in the physical memory, a page fault is triggered. This informs the operating system to swap in the desired page. The swapping operations are much easier due to the smaller and fixed length of the pages. During the program execution a particular logical page can be stored in different physical pages as a page is swapped in and out several times. At any time only a few pages of any program (or process) are kept in the memory. Therefore, more programs (or processes) in a multi-programming system can be maintained in the memory. Thus it becomes possible that a program or process can be larger than the actual capacity of the memory. This memory management scheme is called *demand-paged virtual memory scheme*. As the unused pages are not to be swapped in and out of the memory there is appreciable time saving.

The modern approach is to store the descriptors for currently used segments or pages in the cache memory instead of main memory. This makes the access of descriptors much faster. Some systems have combined memory mapping scheme employing both segmentation as well as paging techniques, e.g. 80386. In this technique each segment contains a number of pages. A logical address has three components: segment, page and page offset. A segment component specifies the desired segment. The page component specifies the page within a segment. The page offset component specifies a word within a page.

6.12.2 Protection

In a multiuser system users should not interfere with each other. The operating system also should not be interfered by the users. The MMU provides necessary protection so that these conditions are fulfilled. Take an example of 80286. The 16-bit segment selector contains 14 bits for address and 2 bits for privilege-level checking. The privilege-level checking is done for the protection. The 2 bits in the segment selector represent the privilege-level of a portion of a program, which is making a request to access a segment. There are also 2 special bits for privilege-level in the descriptor. The MMU compares the privilege-level of the segment selector with the privilege level of the descriptor. If a segment selector contains same or higher privilege-level, then only the memory access is allowed by MMU. If the privilege-level of the segment selector is lower than that of the descriptor, the CPU is interrupted by the

MMU, which indicates a privilege-level violation. This mechanism is utilized to protect operating system from users.

6.12.3 MC68851

It is memory management unit of Motorola. It uses paging technique of memory division. It is used with those microprocessors of Motorola which do not have on-chip MMU. This chip is popular and widely used with 68000 family of microprocessors.

6.13 PCMCIA OR WPCMCIA CARDS AND SLOTS

PCMCIA stands for Personal Computer Memory Card International Association. It is a standard for cards for peripherals whose size is that of a credit card. PCMCIA cards are becoming standard features on portable and desktop machines. These cards are sometimes called PC cards. These are small cards approximately 2 inches wide and 3.5 inches long. The thickness varies from 1/8 inch to 1.5 inch, depending on its type. Originally, the standards were developed for removable memory cards for portable computers. Today, the standards have been developed for extremely diverse devices such as fax, modem, SCSI adapter, an Ethernet adapter, disk drives, etc. The standards specify the physical design of the cards, the physical design of the connector, the electrical interface to cards, etc. PCMCIA slots (socket) are provided on notebooks and other portable computers. Today these are also available on desktop computers. For data exchange purpose, the desktop PCMCIA slots are often designed to fit into floppy drive mounting bays. These slots can be used for exchanging data between a mobile computer system and a desktop computer using PCMCIA cards. With the PCMCIA storage, data can easily be saved into a PCMCIA card (with floppy) on a portable computer, and then the card can be taken out and inserted directly into the desktop machine. A PCMCIA slot supports hot insertion which means that devices can be plugged and unplugged without switching off the power supply to the computer.

There are different types of PCMCIA cards: Type I, Type II, Type III and Type IV. Type I cards are 3.3 mm thick and have 34-pin connector. These are memory cards. Type II cards are 5 mm thick and have 68-pin connector. These are for fax, cellular modem, LAN adapter, wireless LAN adapter, SCSI adapter, etc. Type III cards are 10.5 mm thick and have 68-pin connector. These are for hard disk drive up to 1 GB. Type IV cards are 16 mm thick and have been developed by Toshiba for removable hard disks.

Wireless PCMCIA cards are also available. They are called WPCMCIA cards. These cards are required to interface wireless devices to PC.

PROBLEMS

1. What is the function of memory in a computer?
2. What are the different types of memory? Discuss their merits, demerits and area of applications.
3. What are the various types of semiconductor memory? Discuss their merits, demerits and area of applications.
4. Describe RAM, ROM and nonvolatile RAM with their merits, demerits and area of applications.
5. What are static and dynamic RAMs? Discuss their merits, demerits and area of applications.

6. Describe ROM, PROM, EEPROM and flash memory with their merits, demerits and area of applications.
7. What are main, secondary and backup memory?
8. What is the function of a cache memory? Explain the terms cache hit and cache miss. What is the role of a cache controller?
9. Discuss real and virtual memory? Explain the term swapping. What is logical address and physical address?
10. What is memory controller? For what type of semiconductor memory is it used? What are its functions?
11. Discuss the error detection and correction techniques which are used for RAMs.
12. What are the different types of magnetic memory? Describe them briefly.
13. What are the advantages of magnetic disk memory over magnetic tape memory? What are the advantages of hard disks over floppy disks?
14. Describe the construction and working principle of 3.5 inch floppies.
15. Describe the construction and working principle of hard disks. What do you understand by head crash?
16. What precautions are to be taken in handling floppy disks?
17. What do you understand by formatting of magnetic disks?
18. Discuss error checking techniques which are used for magnetic disk memory.
19. Discuss the functions of floppy disk and hard disk controllers. Give some example of controller ICs.
20. What is an optical disk? Discuss their advantages and disadvantages as compared to magnetic disks. What are the different types of optical disks? Discuss their area of applications.
21. What is CCD memory?
22. What are direct access storage devices? What are their advantage over sequential memory?
23. What do you understand by destructive and nondestructive read out?
24. What are program and data memory?
25. Discuss the role of a memory management unit. Discuss how the MMU generates more logical memory addresses than the actually existing physical addresses.
26. Discuss the merits and demerits of segment oriented memory system and page oriented memory system.
27. Discuss the demand-paged virtual memory scheme of memory management.
28. What is protection? How is it achieved?
29. What are SCSI and EIDE adapters? Discuss their applications.
30. What are DVD-ROM, DVD-R and DVD-RAM? In what way they differ from CD-ROM, CD-R and CD-RAM?
31. What are PCMCIA and WPCMCIA cards and slots? Discuss their applications.

7

CHAPTER

I/O DEVICES, MICROCONTROLLERS, I/O PROCESSORS AND ARITHMETIC PROCESSORS

The input/output (I/O) devices and secondary storage units of a computer are called *peripherals*. The term peripheral is used in a wider sense, it also includes interfacing devices such as I/O port, programmable peripheral interface, programmable interrupt controller, DMA controller, communication interface, counter/interval timer, CRT controller, memory controller, floppy disk controller, hard disk controller, keyboard interface etc. Some of these controllers have already been discussed in Chapter 6. I/O devices, remaining interfacing devices, I/O processors, arithmetic processors etc. will be discussed in this chapter.

7.1 INPUT DEVICES

Data and instructions are entered into a computer through input devices. An input device converts input data and instructions into suitable binary form which can be accepted by the computer. The commonly used input device is a keyboard. A number of input devices have also been developed which do not require typing for inputting information. Examples are: mouse, light pen, graphic tablet, joy stick, track ball, touch screen etc. Each of these devices permits the user to select something on CRT screen by pointing to it. Therefore, these devices are called pointing devices. Voice input systems have also been developed. A microphone is used as an input device.

7.1.1 Keyboards

Programs and data are entered into a computer through a keyboard which is attached to a microcomputer or the terminal of a server or supercomputer. A keyboard is similar to the keyboard of a typewriter. It contains alphabets, digits, special characters, functions and some control keys. When a key is pressed an electronic signal is produced which is detected by an electronic circuit called *keyboard encoder*. A keyboard encoder may be special IC or a single-chip microcomputer used as encoder. The function of an encoder is to detect which key has been pressed and to send a binary code (corresponding to the pressed key) to the computer. The binary code may be an ASCII, EBCDIC or HEX code.

In some computer systems the keyboard encoder sends simply a *scan code* to indicate which key has been pressed. There is a computer program called *keyboard translation program* to assign the meaning of the scan code. The program will give the desired binary code corresponding to the pressed key. This approach is known as *soft keys*. This approach makes possible to change the meaning associated with various keys on the keyboard.

A single chip microcomputer used as encoder contains a lookup table in a ROM. The binary code is obtained from the lookup table. By changing the lookup table in the ROM the output code can be changed. For example, the keyboard system which previously was outputting ASCII, can be made to output EBCDIC or any other codes by changing lookup table in the ROM.

The codes from the encoder may be transmitted to the computer usually in serial mode. In serial transmission number of connecting wires is reduced and data is sent one bit at a time. Computers use standard keyboards having **QWERTY** type layout of characters, which is commonly used in mechanical typewriters. For faster typing keys are rearranged to give a different layout known as **Dvorak keyboard**. It is available at extra cost as an optional.

Fig. 7.1 shows key arrangement of a QWERTEY type keyboard. It is called QWERTY type keyboard because the letters in the word 'QWERTY' are the first six letters of the third row from the top of the keyboard. The keyboard contains the following major categories of keys:

- (i) *Alphabet Keys.* There are 26 letters of English alphabets. When an alphabet key is pressed, small letter is typed. Keeping Shift key pressed, when an alphabet key is pressed, capital letter is typed. Keyboard of other languages are also available.
- (ii) *Digit Keys.* There are 10 keys for digits 0 to 9. Digit keys are on the second row from the top of the keyboard. There is one additional set of digit keys, called numeric key pad, at the lower right side of the keyboard. It is for faster entry of numbers.
- (iii) *Special Character Keys.* These are for full stop, comma, :, ;, ?, /, \, &, %, @, #, \$, !, ^, *, +, -, _, <, >, {, [, (, },],), " , ', etc.
- (iv) *Control Keys.* These are: Esc (Escape), Backspace, Enter (Return), Ins (Insert), Del (Delete), Arrows keys, Tab, Shift, Ctrl, Alt, PgUp (Page up), PgDn (Page down), Home, End, PrtSc (Print Screen), Windows, etc.
- (v) *Function Keys.* Function keys from F1 to F12 are on the first row of the keyboard. The functions of some control keys are as follows:

Enter Key. When Enter (Return) key is pressed, the cursor comes in the beginning of the next line. This key is pressed at the end of a line typing to go to the next line.

Home Key. It allows users to move cursor to the beginning of line. Keeping Ctrl pressed when 'Home' is pressed, cursor goes in the very beginning of the screen.

End Key. It brings the cursor at the end of the line of the text. Keeping Ctrl pressed, when 'End' is pressed, the cursor goes at the end of the text (or any other stored information).

PgUp (Page Up). It allows users to move one page of the text (or any other stored information) upward at a time on the screen.

PgDn (Page Down). It allows users to move one page of the text (or any other stored information) downward at a time.

Arrow keys: They can move cursor, up, down, left or right as desired.

Del (Delete) Key. It allows to erase character.

Ins (Insert) Key. It allows to insert characters.

PrtSc (Print Screen) Key. In DOS operating system, when this key is pressed, the text or any other information which is on the screen, is sent to the printer for printing. In WINDOWS operating system, this key is used to obtain Screen Capture.

Tab Key. It moves cursor to the next tab stop.

Backspace Key. When this key is pressed, the cursor comes to the previous character's position and the current character is deleted.

Shift Key. Keeping this key pressed, when an alphabet key is pressed, capital letter is typed. In case of digit and some other keys, the symbols which are written at the top of the key, is printed.

Ctrl Key. It is a control key. It is used in the combination of other keys either a letter or number. Keeping this key pressed, when certain key is pressed, some control action is performed. Examples are:

Ctrl + B → Add or remove bold formatting.

Ctrl + P → Print

Ctrl + S → Save

Ctrl + U → Add or remove underline

Ctrl + Z → Undo last command

Alt (Alternate) Key. It gives alternate function of any other key. Keeping this key pressed, some other key is pressed to get some special function. The special function depends on the application software under which certain special function is to be performed. For example, when F4 is pressed after pressing 'Alt', application program is closed.

Functions of function keys depends on the software which is being used currently. A function key invokes certain program which is stored in the computer. When F1 is pressed, 'Help' is activated. While working in an application, pressing this key brings up the application help menu. If there is no application, F1 opens WINDOWS Help.

Sometimes someone can type some words, and then he realizes that the Caps Lock key was mistakenly switched on. So everything is in capital letters. To convert it to small letters, select the text and press Shift + F3.

Windows Key. There are two Windows keys in the bottom row of the keyboard. There is a Windows logo on the Windows key. When it is pressed, Start Menu is displayed. Holding the Windows key down, when certain keys are pressed, some actions are performed. An example is:

Windows + F1 → Displays Windows Help

Application Key. This is in the bottom row of the keyboard. It has an image of mouse pointer on a menu. It performs the same function as the right-button clicking of a mouse.

Symbol * in the second row is for multiplication. The symbol ^ in this very row is for 'raise to the power', for example, if we write 2^3 , it means 2 raise to the power 3. Its value is $2 \times 2 \times 2 = 8$.

In a multimedia keyboard buttons for volume control, CD player's control, sleep, cut, copy, paste, rotate, close DOS, etc. have also been provided. Wireless laser keyboards, wireless radio frequency operated keyboards, illuminated keyboards, keyboards with adjustable wrist rest for ergonomic comfort, keyboards specially designed for polluted industrial environment, waterproof keyboards, keyboards specially designed for games, disable persons etc. are also available.

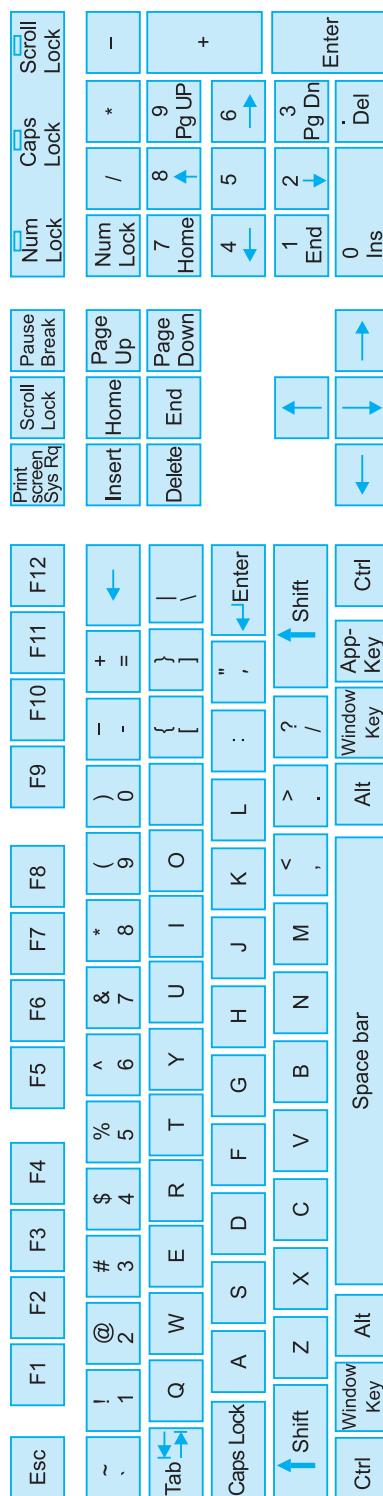


Fig. 7.1 Keyboard

Construction of the Keys

Mechanical Keyswitches. This type of key switches are mechanical type switches. When a key is pressed two metallic pieces make contact. The switch elements are usually made of phosphor-bronze alloy. When a key is pressed an electrical signal is produced. Mechanical type keys suffer from contact bounce. A pressed key may make and break the contact several times before it makes a perfect contact. They may become dirty or oxidized. This makes their contact imperfect and they are no longer reliable. The life of high quality mechanical keys is about 1 million keystrokes.

Capacitive Keyswitches. A capacitive type keyswitch uses plates: one movable and two fixed. When a key is pressed, the movable plate comes closer to the fixed plates. This causes a change in the capacitance between the two fixed plates. An electronic circuitry detects this change and produces a signal. It is better and more reliable than mechanical switches, but it requires special circuitry to detect the change in the capacitance. Its life is more than mechanical type keyswitches, 20 million keystrokes.

Hall Effect Keyswitches. In this type of keyswitches a current is passed between two opposite faces of a semiconductor crystal. This current acts as a reference. A magnet is moved to provide magnetic field. When a key is pressed, a small magnet moves. This produces a small voltage across the other two opposite faces of the crystal. Such keyboards are expensive but more reliable. Their life is 100 million or more keystrokes.

Two-Key Lockout

When more than one keys are pressed simultaneously a problem arises that the processor should not perform wrong operation. In the technique of two-key lockout a single key pressing is recognized. Additional key closure is ignored until the previous key is released. Only after the release of the previous key, next one is recognized.

Two-Key Rollover

When two keys are pressed simultaneously or at nearly the same time, each one is recognized. The ASCII code for the first key and a strobe signal for it is sent out. Then the ASCII code for the second one and a strobe signal for it is sent out.

N-Key Rollover

When N-keys are pressed at a time each key is treated independently. The information of all the key closures is stored in some internal buffer. The operations are performed in a sequence.

7.1.2 Light Pen

A light pen is a pointing device. It is used to select a displayed menu option on the CRT. It is a photosensitive penlike device. It is capable of sensing a position on the CRT screen when its tip touches the screen. When its tip touches the screen surface, its photocell sensing element detects the light coming from the screen and the corresponding signals are sent to the processor. The menu is a set of programmed choices offered to the user. The user indicates his choice by touching light pen against a desired description of the menu. The signals sent by the light pen to the processor identifies the menu option.

A light pen can also be used for graphics work. A user can draw directly on the CRT screen with the light pen if the computer system is provided with CAD package. The user can select various colours and line thicknesses, can add or erase lines and can enlarge or reduce the size of the drawings.

7.1.3 Mouse (Puck)

A mouse is a device to move the cursor on the CRT screen (video screen) of the computer at faster rate conveniently. Besides moving the cursor of the screen, certain operations are also performed by pressing buttons provided on the mouse. When a mouse is moved, its distance traveled and the direction is communicated to the screen by a microcontroller which is embedded into the mouse. When a mouse is moved, the cursor on the CRT screen moves accordingly. Cordless mouse is also available. Scroll mouse has a scrolling feature i.e. text on the screen can be moved up or down moving a wheel which is on the mouse. A mouse may have three or four buttons for its operation. Modern mouse are optical mouse. An optical mouse contains camera, light emitting diode (LED) and an embedded digital processing unit. The LED illuminates surface area. A small area of the surface underneath is focused on a tiny digital camera. The image taken by the digital camera is converted into digital signals which are sent to the embedded digital processor. Any surface usually contains microscopic patterns having lines, changes in brightness, shadows etc. The processor compares the successive images taken by the camera to determine the distance traveled and the direction of movement of the mouse. The processor uses digital signal processing technique to measure the distance traveled from one picture to the next picture, and the direction of mouse movement. The camera takes more than a thousand pictures every second. A very powerful low-cost embedded processor is used to perform intensive computation. This type of mouse does not require any pad. It can be moved on almost any type of surface. A cordless mouse uses RF (Radio Frequency) signal to interface it to the computer.

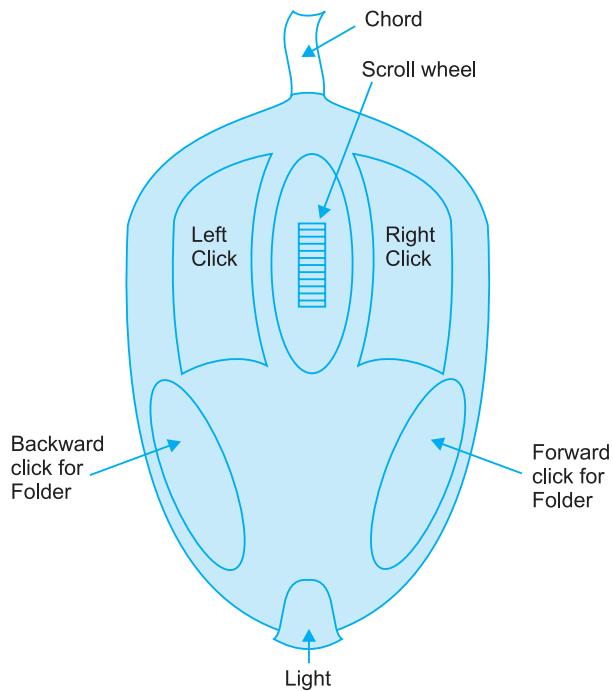


Fig. 7.2 Mouse

The optical mouse discussed in the above para is a complex optical mouse. In a simple optical mouse, there is a light sensitive device to sense the reflected light from the surface. When mouse is moved, the reflected light changes from light to dark area underneath. By counting such changes the distance traveled is measured by the microcontroller and corresponding signals are sent to the computer.

Earlier, a mouse was a mechanical one. It employed small wheels and was moved on a rubber pad. Electrical pulses were generated when wheels moved. These pulses were proportional to the incremental movement of the wheel. A microcontroller is used to send these pulses and pulses corresponding to the button pressed. The cursor moves on the CRT screen corresponding to the movement of the mouse.

Mouse has also been developed to give indications in three dimensions: X, Y and Z. Such mouse is used for 3-D simulations. Laser mouse has also been developed.

7.1.4 Joystick, Pointing Stick and Track Pads

A **joystick** is also a pointing device. It is used to move the cursor position on a CRT screen. Its function is similar to that of a mouse. A joystick is a stick which has spherical ball at its lower end as well as at its upper end as shown in Fig. 7.3. The lower spherical ball moves in a socket. The joystick can be moved right or left, forward or backward. The electronic circuitry inside the joystick detects and measures the displacement of the joystick from its central position; the information is sent to the processor. Buttons mounted on the stick or elsewhere on the joystick, can be pressed to execute commands. Joystick is used in games. Potentiometer and capacitor are used in the sensing element. Digital joysticks have also been developed. They use piezo-electric sensor which uses a crystal to produce an electrical signal when it is pressed. The electrical signal is fed to ADC (Analog-to-Digital Converter) which gives digital output. The digital values are processed by the computer. Cordless joysticks are also available.

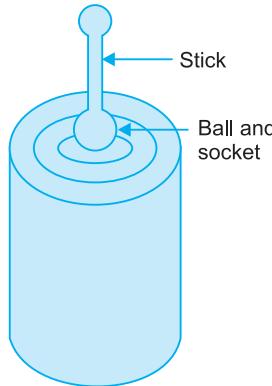


Fig. 7.3 Joystick

Eraser or Pointing Stick

It is a pointing device. It is used on laptops where there is no space for conventional mouse. It looks like a pencil eraser. It is embedded among G, H and B keys on a keyboard. When a finger applies lateral pressure to the eraser, the pressure is passed on to the four force-sensing resistors employed to sense up, down and sideways forces. A single-chip microcomputer (microcontroller) is provided to measure the forces and send the information about cursor movement on the screen.

Track Pads

It is one of the latest pointing devices. It looks like a small, gray window, about two inches wide. One merely slides his finger tip across the window to move the cursor. When window is tapped, it works as the clicking of primary mouse button. A larger track pad called Power Cat has also been developed. It has a wider window about three inches wide. There is a special region on the window. When the user taps this with his finger, it produces the effect of secondary mouse button click. Some area of the window is used to scroll up, down, left or right in the document. It has also two places for triggering the effect of the forward and backward buttons in a Web browser.

7.1.5 Trackballs

Trackball is also a pointing device and contains a ball which can rotate in any direction. The user spins the ball in different directions to move the cursor on the CRT screen. The associated electronic circuitry detects the direction and speed of the spin. The information is sent to the processor. It is used on a laptop computer where there is no space for conventional mouse.

7.1.6 Scanners

Scanners are a kind of input devices. They convert printed text, graphics, pictures etc. to digital form. They are capable of entering information directly into the computer. The main advantage of direct entry of information is that users do not have to key the information. This provides faster and more accurate data entry. Important types of scanners are optical scanners and magnetic-ink character readers.

Optical Scanners

Flatbed scanner. In a flatbed scanner, the page to be scanned is placed over a glass plate. A light source is kept below the glass plate, which moves from left to right horizontally. Each page is treated as a matrix of dots. Each dot is sensed whether it reflects light or absorbs light. The light source focuses light on very thin line called scan line. The dots lying on a scan line is sensed. The entire page is divided into a number of scan lines. The scanner senses each scan line one by one. The dot (spot) which reflects light are taken as 1 and the spots which do not reflect light (i.e. absorbs light) are taken to represent 0. Reflected light by dots are sensed by CCDs (Charge Coupled Devices). In many scanner 300 hundred of CCDs are provided per inch on a horizontal line to give 300 bits per inch resolution. After scanning one scan line, the scanner scans the next scan line. In this way it scans the entire page. The entire page is represented by matrix of bits. The scanned page is stored in the computer's memory in bit mapped form. If the page contains pure typed text, the bit mapped form can be converted to ASCII form. Bit mapped form requires much more storage. Another disadvantage of bit mapping is that individual characters are not stored and hence, it is not possible to search for character strings. Software called optical character recognizer, is available to convert bit mapped form to ASCII form. But pictures, graphics etc. can not be converted to ASCII form.

Optical character recognizer works on comparison principle. It can recognize texts which are typed in standard type fonts. Handwritten or any different kind of fonts will not be recognized correctly and efficiently. At present what OCRs (optical character recognizers) are available, do not recognize 100 percent correctly. So proof reading has to be done and errors are to be corrected.

Optical Mark Reader (OMR)

Special marks such as square or bubble are prepared on examination answer sheets or questionnaires. The users fill in these squares or bubbles with soft pencil or ink to indicate their choice. These marks are detected by an optical mark reader and the corresponding signals are sent to the processor. If a mark is present, it reduces the amount of reflected light. If a mark is not present, the amount of reflected light is not reduced. This change in the amount of reflected light is used to detect the presence of a mark. This method is used where one out of a few number of alternatives is to be selected and marked. For example, market survey, population survey, objective type answer sheets etc. where choice is restricted to one out of a few choices.

Optical Bar-Code Readers

This method uses a number of bars (lines) of varying thickness and spacing between them to indicate the desired information. Barcodes are used on most grocery items. An optical-bar reader can read such bars and convert them into electrical pulses to be processed by a computer. The most commonly used bar-code is universal product code (UPC). The UPC code uses a series of vertical bars of varying width. These bars are detected as ten digits. The first five digits identify the supplier or manufacturer of the item. The second five digits identify individual product. The code also contains a check digit to ensure that the information read is correct or not.

A point-of-sale terminal often contains an optical-bar reader. Such a terminal is used to process the sale transactions. It is an on-line terminal connected to a computer for processing. The optical-bar reader reads the bar-code of an item. The bar-code is decoded and data are sent to the computer. The computer prints the name of the item, its price and other necessary informations on the receipt. The stock at hand and sale records are also updated.

Magnetic-Ink Character Reader (MICR)

MICR is widely used, by banks to process large volumes of cheques and deposit forms written every day. A special ink called magnetic ink (i.e. an ink which contains iron oxide particles) is used to write characters on the cheques and deposit forms which are to be processed by an MICR. MICR is capable of reading characters on a paper written with magnetic ink. The magnetic ink is magnetized during the input process. The MICR reads the magnetic pattern of the written characters. To identify the characters these patterns are compared with special pattern stored in the memory. Before cheques are issued to customers the identification number of the bank and the depositor's account number are printed on the lower left-side of the cheques with the magnetic ink. The amount is printed on the lower-right side of the cheque by the bank employee with the same magnetic ink after the cheque is received from the customer. When a cheque is entered into an MICR, it passes through a magnetic field. The iron oxide particles are magnetized under the magnetic field. The read head reads the characters written on the cheque with the magnetic ink. It interprets the characters and sends the corresponding data directly to the computer for processing. Upto 2600 cheques are processed per minute by an MICR.

7.1.7 Voice Input Systems

Data entry into a computer manually using keyboard is a time-consuming and laborious task. It will become very easy if we can talk to a computer. Attempts have been made to develop a computer that can listen to the users and talk to them. The voice input to the

computer i.e. voice recognition by a computer is much more difficult than the voice output. It is because of the fact that the rules for generating voice through a speaker or a telephone system can easily be defined compared to the rules for interpreting words spoken by a person. The tones of speech, speed, accent and pronunciation differ from person to person. These differences in speech makes voice recognition a difficult job. In a voice input system the speech is converted into electrical signals employing a microphone. The signals are sent to a processor for processing. The signal pattern is compared with the patterns already stored in the memory. A word is recognised only when a choice match is found, and then the computer gives a corresponding output. At present a voice recognition system is costly. In future it is expected to become cost effective and will be widely used for direct entry of data. IBM has developed a Talkwriter with 6000 words. It is capable of detecting words with 95% accuracy. It is meant for business correspondence. A voice recognition system can be used in factories at places where both hands of worker are engaged in the job he is doing and he wants to input some data into the computer. It can also be used to assist bedridden and handicapped persons in a number of tasks; to control access to restricted areas; to identify a customer in a bank etc.

7.1.8 Touch Screen

Some computers have touch screen which is sensitive to user's touch. One can use finger to point the command displayed on the screen. It is popular on laptops. Many techniques have been used to make the screen sensitive to touch as described below :

- (i) Capacitive screen uses a device which can sense changes in capacitance when and where the user touches the screen with a stylus or finger.
- (ii) Infrared screens employ light-emitting diodes and photo detector cells to cover the screen with invisible light. LEDs emit infrared light, and photo detectors receive it. When the user touches the screen, some light beams are interrupted, and the computer then senses the position of the finger.
- (iii) Pressure-sensitive screens of Mylar, separated by a small space are used. Each sheet of Mylar contains rows of invisible wires. The sheets are placed in such a way that the wires run horizontally in one sheet and vertically in the other. When the user applies pressure on the screen, the wires at that point make contact and a circuit is closed. This is sensed and fed to the computer.

7.1.9 Some Other Forms of Input Devices

Microcomputers or microprocessor-based systems are now widely used in industry for automatic control. Physical quantities like temperature, pressure, speed, deflection, strain, stress, force, vibration etc. are measured and controlled by microcomputers. An electrical or electronic device called transducer is used to sense physical quantity and give proportional electrical signal. The electrical signals are amplified and then converted to digital signals. The digital signals are fed to the processor for measurement, display and control purposes. Transducers, amplifiers, analog-to-digital converter etc. form a circuitry called data acquisition system. The data acquisition system acts as an input device. Electrical quantities like voltage, current, frequency, power, energy etc. are also measured, displayed and controlled by microcomputers. The data acquisition system for electrical quantities do not need transducers. They employ amplifiers, analog-to-digital converter, analog multiplexer, sample and hold circuits etc. Also, in the case of processing of electrical quantities the data acquisition system acts as an input device. In some cases switches are used to supply electrical signals to computers for control purposes. In those cases switches act as input devices.

In many applications it is desired that a computer should be able to see its environment. For example, a robot must be able to see to perform its job, a computer-controlled security system must be able to see its environment etc. To provide vision to computers, sensors like video cameras, CCD cameras, OPTICRAM cameras etc. are employed. These cameras act as sensors to provide signal proportional to the intensity of light falling on the various spots of the image of an object. The computer can process these signals and recognize and display the image of the object. Such sensing devices which provide the required signals to computers act as input devices. Details of such sensors is discussed in Chapter 9 under computer vision.

7.2 OUTPUT DEVICES

The output devices receive informations from the computer and provide them to users. The computer sends informations to the output devices in binary coded forms. The output devices convert them into a form which can be used by users such as printed form or display on a screen. In some applications the computer's output may also be converted by an output unit in a form which can be used as input to other devices, equipment, machines etc. The commonly used output devices with general purpose computers are: CRT screen and printers.

Computers or microprocessor-based systems are now widely used for automatic control applications in industry and other commercial organizations. In such cases the computer outputs electrical signals which are sent directly for control purposes. In some cases digital-to-analog converters are used as output devices to give control signals to controllers, actuators, relays etc.

There are certain devices which act as both input as well as output devices. Examples are: teleprinter terminal, visual display terminal etc. A teleprinter terminal contains a keyboard for input and a typewriter like printer for output. A visual (or video) display terminal (VDT) contains a keyboard for input and a visual display unit for output. The visual display unit is called *monitor* or *video monitor*. Most VDT units contain CRT for visual display. Other types of display units are also available such as LED (light emitting diodes) display, LCD display and plasma screens. The VDT which contains CRT for visual display is also called CRT terminal.

7.2.1 CRT Terminals

A CRT *terminal* consists of a CRT display unit, a keyboard, CRT-refresh RAM, CRT controller and USART or UART for communication with the computer. Modern CRT terminals contain one or more built-in microcontrollers to control and co-ordinate keyboard, CRT display unit and data transmission from the terminal to the computer and vice-versa. The data are entered into the computer through the keyboard. Each entered character is also displayed on the CRT screen, so that the user can see what he has typed. When data are keyed in, they are held in a small memory called a *buffer*, within the terminal itself. The data are not transmitted to the computer until the user presses an enter key on the keyboard. A small square pointer on the screen, called a *cursor* indicates the spot on the screen where the next character to be keyed will be displayed. To correct keystroke errors the cursor is moved to the position where the correction is to be made. Then the key for the correct

character is pressed. Since the terminals are used for interaction with operators, fast data transmission is not required. Therefore, usually data are transmitted from the terminal to the computer and vice versa in a serial mode i.e. one bit at a time.

7.2.2 CRT Display Unit

A CRT (Cathode Ray Tube) display unit is a commonly used output device. It displays the data or information received from the computer. It can display alphanumeric characters and graphs. Though the CRT is basically an output device it can perform limited input function when used in conjunction with a light pen. The CRT screen is similar to a TV screen. A CRT is a vacuum tube. An electron beam is produced by the electron gun located at the back portion of the tube. The electron beam is directed towards the front of the CRT (i.e. CRT screen). A coating of phosphor material is made on the inner surface of the screen. The phosphor emits light when it is struck by electron beam. The colour of the emitted light depends on the phosphor substance used. A CRT display may be either monochrome (i.e. only one colour) or colour (i.e. multicolour). Monochrome displays are available in green, blue, orange, yellow, pink, amber, red, and white depending on the type of the phosphor material used. Colour displays are produced on CRT screen on the same principle as they are produced on colour TV screen. To produce colour display three phosphors: red, blue, and green are used. The coating of these phosphor materials is made in such a way that dots of these three phosphors in a triangular pattern are spread over the entire screen. The three dots of the three colours placed on a triangular spot are so close that they appear as a single dot. Three separate electron beams are employed to illuminate the dots of three different phosphors. By varying the intensity of the three electron beams the intensity of red, blue and green dots is varied. This gives the appearance of a triangular spot (consisting of red, blue and green dots) of the desired colour. Black and white colours can also be obtained. When all the three beams are off the spot will be a black one. When red is 30%, green 59% and blue 11%, the spot will become white. By turning on and off the different combinations of red, blue and green beams 16 types of colour can be produced. A much wider variety of colours can be obtained by controlling the intensity of red, blue and green electron beams employing digital-to analog (D/A) converters.

Raster Scan and Vector Scan Method of Display

Characters or graphics displayed on the CRT screen are formed of a number of dot points. The dot points are arranged on horizontal scan lines on the screen. In the raster scan method the electron beam is first directed at the top left-hand corner of the screen and then it is moved along the first horizontal scan line. The beam illuminates the selected dots along the first horizontal scan line, which are needed to produce characters or graphics. When the beam reaches right end of the first scan line, it is turned off (blanked) and retraced rapidly left side to the starting point of the second scan line. Now it moves along the second scan line and illuminates the required dots on it. This process is repeated and all the scan lines of the screen are illuminated to display images, characters or graphics on the screen. When the beam reaches at the end of the last line it is blanked and retraced back to the starting point of the first scan line again and the entire process is repeated again to refresh the illumination of the desired dots so that one can always see the display due to persistence of vision. In this method the beam is scanned over the entire screen.

In many applications we want only to draw an array of straight lines. In such cases it will be wasteful to move the electron beam along all the scan lines over the entire screen.

Moreover, the diagonal lines displayed by raster technique look like stair steps. In vector scan (or random scan) method deflecting plates are used to deflect the beam left or right, up or down. The beam can be directed to any point on screen directly. This technique illuminates the selected dots on the screen directly. Straight lines can be drawn connecting any two points on the screen. By applying the proper analog voltage to horizontal and vertical deflecting plates the electron beam can be directed to any desired spot. D/A converters are used in the circuitry of deflecting plates to give appropriate analog voltages. The vector scan method is quite suitable to display graphics consisting of straight lines. But it is not suitable to display curves.

Display of Characters on CRT Screen

A character is displayed on the screen by light dots. A matrix of dots is used for this purpose. To display a character the size of the dot-matrix may be 5×7 , 7×9 or 7×12 . The desired dots are lighted to display a character. A ROM called *character generator ROM*, stores the dot pattern for the display of each character. The ASCII or EBCDIC codes of the character of a text to be displayed at a time on the screen, are stored in a RAM, called *display RAM* or *video RAM* or *display refresh RAM*. When new text is to be displayed the contents of the RAM are changed accordingly. A CRT screen displays 25 lines (rows), each line (row) containing 80 characters. So 2KB (25×80 bytes) display RAM is required for this purpose. A *row counter* and a *character counter* are employed to address the ASCII code of a character in the display RAM.

Each row of the dot matrix is called *dot row*. Therefore, to display one row of characters (i.e. one line of the text) 9 dot rows will be needed, if 7×9 dot-matrix is used. The dot-rows of the same characters-row form *scan lines*. A dot-row of dot-matrix is a section of the scan lines as shown in Fig. 7.4. A schematic diagram to explain the principle of displaying a character is shown in Fig. 7.5. The ASCII codes of the character to be displayed on the screen are stored in the display RAM. By setting the row counter and character counter the address of the first character of the first character-row is applied to the display RAM. The display RAM gives an output, i.e. ASCII code of the first character of the first character-row. This code is applied to the character generator ROM which contains the dot-matrix pattern of the character.

A dot matrix has 9 dot-rows. At a time one dot-row is taken up for scanning by the electron beam. A *dot-row counter* is also employed to count the dot-rows of a character (or scan lines of the characters). The output of this counter controls the character generator ROM. So the ROM will give the dot pattern of the first scan line of the first character. This output is applied to a shift-register. The function of the shift register is to convert the parallel input to serial output because the scanning of dots of a dot-row is to be done one dot at a time. The output of the shift register is amplified and then used to lighten the required dots of the first scan line of the first character (i.e. the dots on ab). The eighth bit of the data input of the shift register is grounded. This gives one dark dot or undot between adjacent characters. After this the first scan line of the second character of the first character row (i.e. cd) is taken up. Similarly, first scan line for all 80 characters is taken up. Then the electron beam is banked and retraced to the starting point of the second scan line of the first row of characters. In this way all 9 scan lines are taken up one by one for the first row of characters. After this the second row of the characters (i.e. second line of the text) will be taken up. This process will be repeated to take up all the 25 character-rows. When the electron beam reaches at the right end of the last scan line of last character row, it is retraced back to the

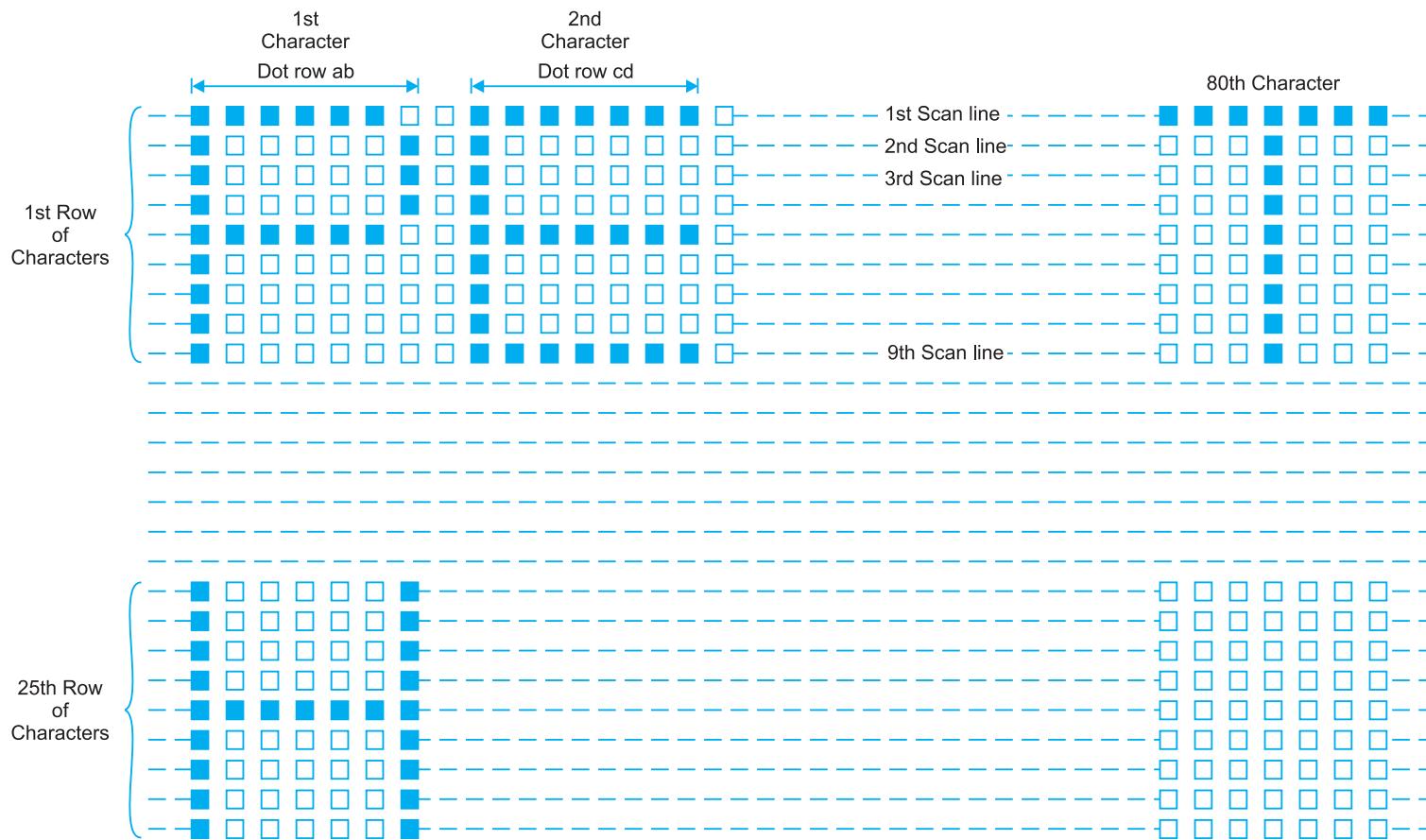


Fig. 7.4 Character display by dot-matrix.

starting point of the first scan line of first character row. The entire process of scanning is again repeated to refresh the characters displayed on the screen. The refreshing is done 60 times per second.

In some arrangements the ASCII codes of the characters are stored in alternate memory locations of the display RAM. Associated with each character an attribute byte is also stored. The attribute byte specifies the quality of the character, such as underline, increased or decreased intensity, whether it is blinking and so on. In some cases two RAMs are employed. One stores codes of the characters while the other stores attributes.

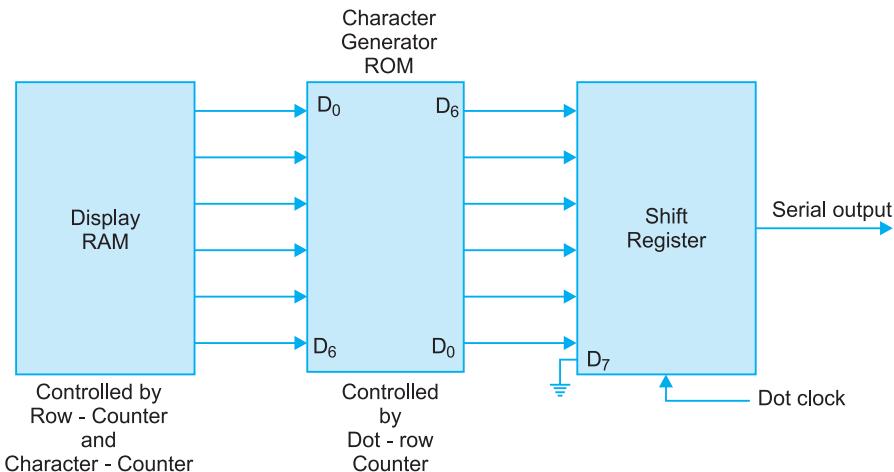


Fig. 7.5 Schematic diagram of character displaying circuit.

Bit-Mapped Raster Scan Method

This technique is used to display graphics or pictures. The screen is not treated as 25 rows of characters and each row containing 80 characters. The entire screen is treated as a matrix of dots. Each dot is programmed whether it will be on or off. There is no display RAM and character generator ROM. The dot pattern for each 8 dots of a scan line is stored in successive memory locations. A byte is read from the memory and applied directly to the shift register which gives serial output. The desired dot pattern for a section of the scan line is displayed on the screen. Then next byte from the memory is taken up. The process of the raster scan is followed and the dots of the entire screen are scanned. Each dot is called a picture element or in short *pixel* or *pel*. This method is very flexible. It has two major drawbacks. It requires larger RAM capacity, i.e. 16 KB compared to 2 KB in character display method. To produce colour graphics still more memory locations are required. In this technique tremendous amount of copying is required. It needs a very fast CPU or special hardware for controlling pixels quickly.

Programming of each pixel is a very tedious work. To make programming easy graphics programs are available. One has to learn how to use these programs. In graphics program subroutines are available to draw lines, circles, arcs etc. using which complex drawings can be drawn and print out can be obtained. A mouse is very helpful in preparing such drawings.

A bit-map terminal can support displays having a number of windows. A window is defined as an area of the screen used by one program. The facility of multiple windows allows to run several programs at the same time. Each program displays its results independent of the others.

Some video terminals have both facilities: they can work either as a bit-map terminal or a character-map terminal.

7.2.3 CRT Controllers

A CRT controller contains the necessary electronic circuitry to control CRT display. Several controller ICs are available. A few of them are described below.

Intel 8275

It is a single chip programmable CRT controller to interface CRT raster scan displays. It is compatible with 8085, 8086 and 8088 Intel microprocessors. Its basic function is to display characters and refresh the display. It also has graphics capability. It contains a row counter, a character counter, a scan line counter, a raster timing circuitry and video control unit. It also contains two 80-byte buffers, one to hold the ASCII codes of characters of the row to be displayed and the other to hold the ASCII codes of characters of the next row. The ASCII codes of characters of a page to be displayed are stored in the main memory of the computer. While the contents of one 80-byte buffer of 8275 are being used to refresh the display, the other 80-byte buffer is filled up with the ASCII codes for the next row using DMA technique. Character generator ROM and shift register are external chips. The 8275 has capability of 4 types of cursor control. It is also provided with light pen detection capability. It has programmable screen and character format. It possesses 6 independent visual field attributes.

Intel 8276

It is small system CRT controller. It is meant to interface CRT raster scan displays with Intel 8085, 8086, 8088 microprocessors and 8051 single chip microcomputer. It has 4 types of cursor control, programmable screen and character format, 6 independent visual field attributes. It contains all necessary counters, buffers, raster timing circuitry and video control, etc.

Intel 82706

It is an Intel video graphics array (VGA) compatible display controller for IBM PC/XT/AT and PS/2. All video monitors developed for IBM PS/2 system are provided with VGA controller. The 82706 can support 256 KB video memory. It is designed to be compatible with 80286 and 80386 and other microprocessors. It has EGA/CGA/MDA BIOS compatibility. It acts as CRT controller and video memory controller.

Intel 82716

It is a video storage and display device. It is a low cost, highly integrated video controller. It displays texts and graphics. It can manage upto 16 display objects on the screen at any time. It can select upto 16 colours from a range of 4096. It contains on-chip DRAM controller and on-chip D/A converters. It can support upto 512 KB of display memory. It is compatible with 8- and 16-bit microprocessors and microcontrollers.

Motorola 6845

It is a CRT controller. It is used in both a monochrome monitor and a colour/graphics monitor. It uses 2 external 2KB memory: one to store ASCII codes of characters and the other to store an attribute code for each character. Character generator and shift registers employed are external. A multiplexer is used to allow either the CPU or CRT controller to access the display-refresh RAMs. It can address upto 16 KB of display and attribute memory.

7.2.4 Non-CRT Displays

Non-CRT displays include LED, LCD and plasma displays. LED displays are used in microprocessor-based industrial controls, instruments etc. where only a small amount of data are to be displayed. CRT screen display is used where a large amount of data are to be displayed. In portable battery powered instruments usually LCD displays are used because they consume less power. Non-CRT displays have been described below in brief.

Liquid Crystal Display (LCD)

In LCDs a liquid crystalline material is sandwiched between two glass or plastic plates. The front plate is transparent and the back plate is reflective. There is a coating of thin film on the front plate. The coating is transparent and conductive. Its sections (segments) are in the shape of desired characters. An electrically conductive film or backplane is put on the back plate. A voltage is applied between a segment and the back plate to create electric field in the region under the segment. The electric field makes a change in the transmission of light through the region under the segment. Commonly available LCDs are of two types: dynamic scattering type and field effect type. In dynamic scattering type the molecules are aligned in one direction under the segment where field is present. The crystalline molecules reflect more light in this condition and this produces etched-glass looking light characters on a dark background. In field effect type, molecules are polarized to absorb light where electric field is present. This produces dark characters on a silver-gray background. LCDs do not emit their own light. Therefore, a light source is to be used. LCDs simply change the reflection of available light. Today, most LCDs used are of the type that produce dark images on a silver background. The colour displays have also been developed. Monochrome and colour displays are expected to grow rapidly in future. D.C. voltage is not used in case of LCDs as it will damage them. A square wave signal of frequency 30–150 Hz is used for the purpose. A 2–3 V is required between the backplate and segment.

Screen type LCDs have also been developed. For screen type LCDs the liquid crystal elements are arranged in a large X-Y matrix of dots. The elements of each row are connected together for driving purpose. Similarly, the elements of each column are also connected together. An individual element is energized by energizing both the row and the column which contain the element. This type of display has a problem of low resolution. The resolution is the ability to distinguish things which are close together. LCDs are lightweight and consume little power which make them attractive for portable computers.

TFT LCD Monitor

At present TFT LCD monitors are widely used in notebooks. TFT stands for Thin Film Transistors. A matrix of tiny switching transistors called TFT is placed on a glass substrate which controls on/off state of each pixel (dot or spot) depending on the signal applied. The advantages of LCD screen is that it is slim, light weight and has low eye-strain compared to a CRT monitor. A best-of-class LCD comes with USB port, widescreen format screen size from 17–30 inches, built-in TV tuner, memory card readers etc. CRT monitors are known to emit harmful radiation, whereas LCD monitors do not. Nowadays good quality LCD monitors are available. A major weak point of LCD monitor is its response time, which is the time taken for a pixel to switch off completely from white to black. The greater lag is not desirable in fast moving videos.

OLED (Organic Light Emitting Diode) Displays

OLED is a solid state device. It contains a couple of organic material layers. It is thinner, lighter, flexible and consumes less power. It is better than liquid crystal displays and LEDs. It emits brighter and more coherent light. The disadvantages of OLEDs are: shorter life and high cost. There is anode and cathode across which voltage is applied. In between anode and cathode, there are two organic layers. The principle of emission of light is “whenever an electron moves into a hole, energy is released”. This energy is in the form of light. The colour of light depends on the type of organic molecules used and the intensity depends on the amount of current flowing in the diode. In an organic layer holes are created, electrons move from the cathode. Electrons move into holes in an organic layer. In an active-matrix OLED, there is Thin Film Transistors (TFT) matrix as a part of the circuitry which decides which pixel is to be turned on.

Plasma Displays

In plasma displays ionized gas is sandwiched between two glass plates. A number of parallel wires run horizontally as well as vertically. A small amount of current is passed through one horizontal and one vertical wire to cause the gas to glow at a spot at the intersection of the wires. The problems encountered with LCDs are eliminated in plasma displays. Unlike LCD, plasma display is brighter. It can be seen from a wide angle, 160 degrees. Gas plasma display screen can easily be made larger than 40 inches (diagonal). LCD screen can not be built in this size. A plasma screen uses simplified circuits. Being less expensive it is suitable for high-volume production. In case of plasma display, phosphor material is used for each cell. For colour display red, blue and green phosphors are used in three adjacent cells.

Light Emitting Diode (LED) Displays

Three types of LED displays as shown in Fig. 7.6 are available. A 7-segment display is very simple and it is used to display only digits and hexadecimal letters. To display numbers and entire alphabets 18-segment displays or dot-matrix displays can be used.

The 7-segment displays being simple are widely used. Drivers/decoders for 7-segment displays are available in IC form. The displays are interfaced to microprocessors through drivers/decodes. Such displays with buffers can also be connected directly to the processor through ports. There are binary codes to display digits and some selected alphabets. These codes are stored in memory. For more details including interfacing Ref. 1 may be consulted.

7.2.5 Display and Keyboard Interfacing Chip, Intel 8279

The 8279 is a programmable keyboard/display device. The data input and display is the basic requirement of many microprocessor-based systems. The 8279 can interface a keyboard and LED displays to 8-bit microprocessors. It relieves CPU from the burden of scanning keyboard and refreshing displays. It has two sections: a keyboard section and a display section. The keyboard section is capable of interfacing a regular typewriter style keyboard or random toggle or thumb switches. It acts as a keyboard encoder for 8 x 8 keyboard. It can provide interface to 64-contact key matrix or to an array of sensors or strobed interface keyboard, such as the Hall Effect and ferrite variety. It has 2-key lockout or N-key rollover with contact debounce features. It has been provided with 8-character keyboard FIFO to store keyboard information.

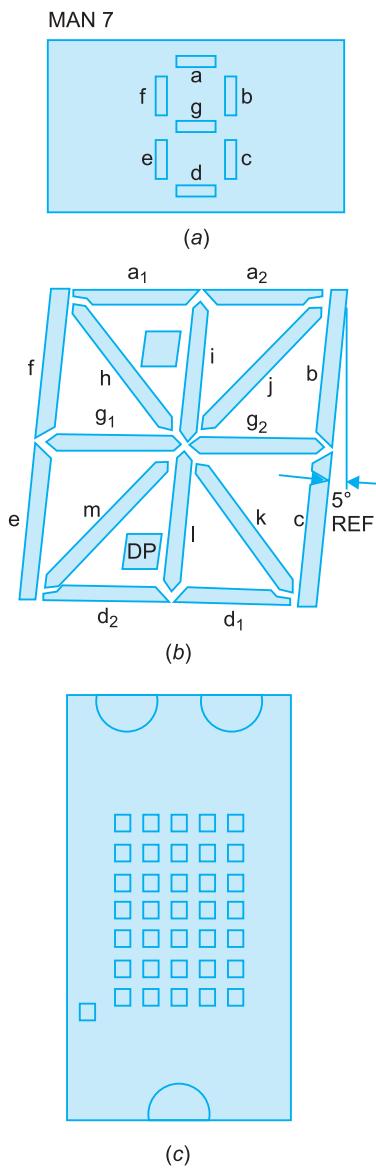


Fig. 7.6 LED display **(a)** 7-segment display, **(b)** 18-segment display, **(c)** 5 × 7 dot matrix display.

Its display section drives alphanumeric displays or indicator lights. It can interface LED, incandescent and other popular displays. It controls display and refreshing upto 16 characters. It contains 16 × 8 display RAM which can be loaded by the CPU. Both reading and writing of the display RAM can be made with auto-increment of the display RAM address.

7.3 PRINTERS

Printers are commonly used output devices. They provide information in a permanent readable form. They produce printed output of results, programs and data. Printers which are used with computers are classified as follows:

- (i) Character printers
- (ii) Line printers
- (iii) Page printers

A character printer prints one character of the text at a time. A line printer prints one line of the text at a time. A page printer prints one page of the text at a time.

The printers have been classified above as to how they print. There is one more classification which depends on the technology used in their manufacture. According to this consideration the printers are classified into the following two broad categories:

- (i) Impact printers
- (ii) Nonimpact printers

Impact printers use electromechanical mechanism that causes hammers or pins to strike against a ribbon and paper to print the text. Non-impact printers do not use electromechanical printing head to strike against ribbon and paper. They use thermal, chemical, electrostatic, laser beam or inkjet technology for printing the text. Usually, a nonimpact type printer is faster than an impact type printer. The disadvantage of nonimpact type printers is that they produce single copy of the text whereas impact printers produce multiple copies of the text. To solve this shortcoming of the nonimpact printers the manufacturers have developed nonimpact type printers that can be used as offline devices to produce additional copies of computer prepared output. For this purpose the printer accepts data from magnetic tape and produce the output.

7.3.1 Character Printers

Character printers print one character at a time. They are low-speed printers. Their printing speed lies in the range of 30–600 characters per second depending upon the type of the printer. They are used with personal computers for low volume printing work. Characters to be printed are sent serially to the printer. A modern printer has its own microcontroller. The microcontroller controls the positioning of the print head or decides which pins are to be employed to form the desired character. It allows printer to receive the next set of characters to be printed from the computer while printing of the current set is going on. Two types of impact character printers are available: dot-matrix printers and letter quality printers.

7.3.2 Dot-Matrix Impact Type Character Printers

A character is printed by printing the selected number of dots from a matrix of dots. Fig. 7.7 shows how a character is printed. The formation of a character has been shown using 7-dot rows and 5-dot columns. This pattern is called 5×7 dot-matrix. The print head contains a vertical array of 7, 9, 14, 18 or even 24 pins. A character is printed in a number of steps. One dot-column of the dot-matrix is taken up at a time. The selected dots of a column (i.e. the column of dot-matrix) are printed by the print head at a time as it moves across a line. A dot-matrix printer is faster than a letter quality printer. Its printing speed lies in the range of 30–600 cps (character per second).

Dot-matrix printers are very flexible. They do not have fixed character font. The term font is used to refer to a character set of a printer. As the fonts are not fixed a dot-matrix printer can print any shape of a character by the software. This permits for many special characters such as α , β , σ , $\sqrt{ }$ etc; various sizes of print, bold or expanded characters, italic, character of any language and provides the ability to print graphics. To print graphics the dot pattern for each column of dots are sent out to the print head from the memory. The principle

is similar to that used in bit mapped raster scan graphics display on CRT. Fig. 7.7 shows how characters are printed by a dot matrix printer. With the advent of inkjet and laser printers, the use of dot-matrix printers has diminished much. They are used in certain applications such as banks, railways etc.

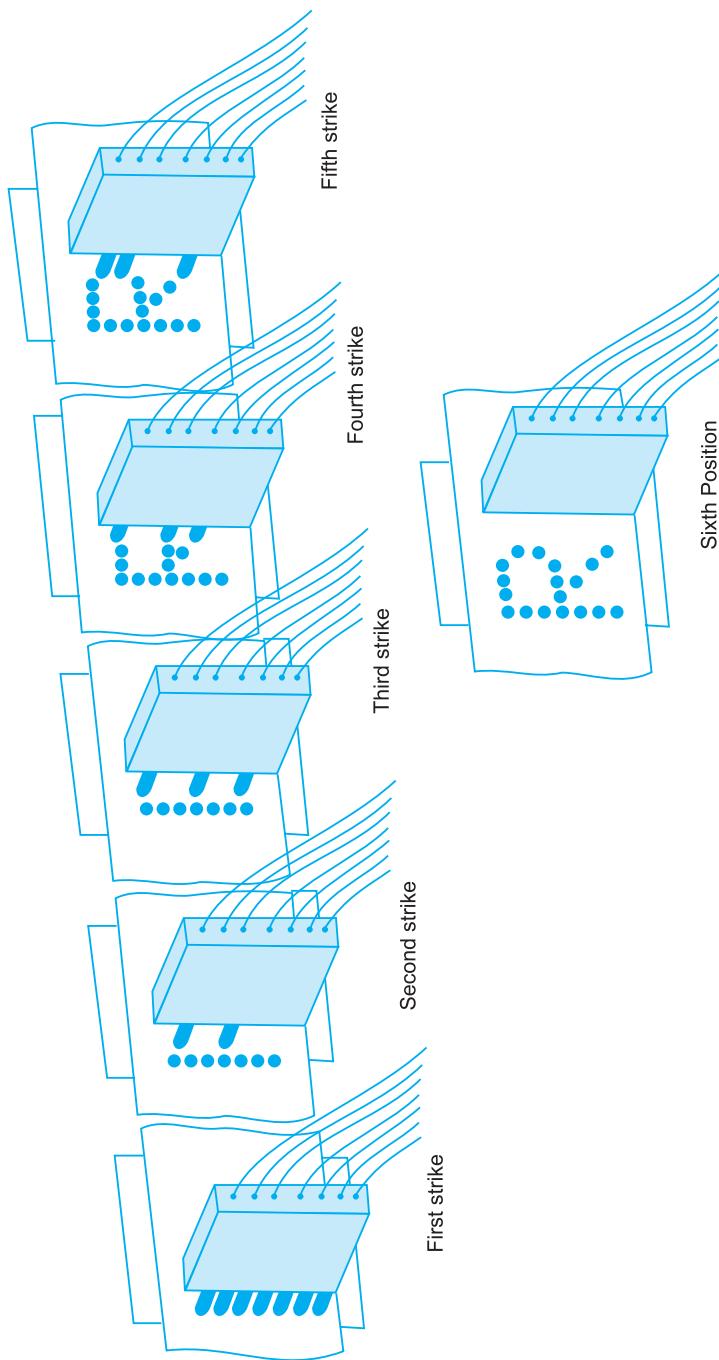


Fig. 7.7 Character printing by dot-matrix printer.

7.3.3 Ink-Jet Printers

An ink-jet printer uses dot-matrix approach to print text and graphics. Most ink-jet printers use multiple jets vertically aligned on a cartridge, which moves horizontally across a page. The ink cartridges contain a column of tiny heaters. When a heater is activated a drop of ink is exploded onto the paper. The print head contains an ink cartridge which is made up of a number of ink-filled firing chambers, each attached to a nozzle thinner than a human hair. When an electric current is passed through a resistor, the resistor heats a thin layer of ink at the bottom of the chamber. The bottom ink layer of the chamber is heated up to 900 degree Fahrenheit for a fraction of a second. This causes the ink to boil and form a vapour bubble. The vapour bubble expands and pushes ink through the nozzle to form a droplet at the tip of the nozzle. The pressure of vapour bubble forces the droplet to move to the paper. When the resistor cools down, the bubble collapses. This results in a suction which pulls fresh ink from the reservoir into the firing chamber.

A colour ink-jet printer employs four ink cartridges : one each for cyan (blue-green), yellow, magenta (purple-red) and black. This system of colour is called CYMK (K stands for black). Unlike colour laser printers and colour thermal printers, colour ink-jet printers do not need multiple passes over the same sheet of paper for colour printing. The ink of desired colour can be placed at any desired point of the page, in a single pass.

Photo Quality Colour Ink-Jet Printers

Printers can give output that matches the clarity, gloss and sharpness of conventional photographs from studios. These true-to-life images are known as photo-quality prints. Traditionally, dye sublimation printers have been used for photo-quality prints. They can achieve photo-quality results even at resolutions as low as 200 dpi. But cost per unit with these printers turns out to be very high. Today, ink-jet printers have been developed to give photo-quality prints.

A colour ink-jet printer uses four primary colours CYMK. It can produce dots of these four primary colours. When several dots of different colours are placed close enough and viewed from a distance, they blend to produce a new colour. This technique is called **half-toning**. Ordinary ink-jet printers use four primary colours, and place two drops of different ink in each dot. In this way they can produce dots of eight colours. Further shades are obtained by half-toning.

There is another technique called **layering** to get wider range of shades. Hewlett-Packard photo-quality printers use this technique. In this technique several ink drops are placed in each dot. This gives better colours without half-toning. It can give 250 colour shades per dot by layering 16 drops of ink per dot. Epson uses six colour cartridges in its photo-quality ink-jet printers to get wider range of colours. Extra colours are low-density magenta and low-density cyan (m and c), which are difficult to obtain. Epson also employs variable-size droplet technology, which can produce dots of different intensity from the same nozzle.

7.3.4 Laser Printers

Laser printers are page printers. An entire page is processed at a time. They use laser beam to produce an image of the page containing text/graphics, on a photosensitive drum. The drum is coated with negatively charged photo-conductive material. A in-built computer controls the laser beam to turn it on and off when it is sent to the drum. The page image is produced on raster scan principle as it is produced on a CRT. The points where laser beam strikes become positively charged. The laser beam causes a negatively charged film (usually,

made of zinc oxide and other material on the surface of the drum) to change its negative charge to positive charge. The areas of the drum surface where laser beam does not strike, retain their negative charge. Negatively charged ink powder called **toner** is used in a laser printer for printing. The laser-exposed areas of the drum attract toner. Paper is positively charged when it passes through the printer. The positive charge on the paper is stronger than the positive charge on the laser-exposed surface of the drum. The paper charge being more positive, pulls the toner off the drum and puts it on the paper. The paper then moves to a fusing station where toner is permanently fused on the paper with heat and pressure. After this the drum is discharged and cleaned. Now the drum is ready to process the next page. In this case black letters are written on white paper. The surface of the drum which does not attract toner, gives white areas on the paper. Printers writing black image on white paper are called **write-black printers**. Low-speed laser printers print 8–10 pages per minute. High-speed laser printers print 200 pages per minute.

A **write-white printer** prints white characters on black background. In such a printer laser-beam creates negative charge on the surface of the drum, where it strikes. Toner is negatively charged. Positively charged photoconductive material is used for the coating of the drum surface. The surface area of the drum, which is not exposed to laser-beam, remains positively charged. Toner is attracted by the area of the drum which is not exposed to laser-beam. Paper is positively charged but more stronger. It attracts toner, and a white print against black background is obtained.

Colour laser printers are also available. They use toner of different inks, and they make multiple passes over the same sheet of paper for colour printing. A colour laser printer has ability to provide the most precise details, but it is slow, complicated and expensive. It needs four separate print engines which must take their turn to apply coloured toner to the page.

The difference between a xerox machine and a laser printer is that in a xerox machine the page image is formed on the photosensitive drum with a camera lens, whereas in a laser printer page image is formed by scanning laser beam using computer control.

A laser printer contains a powerful embedded single-chip microcomputer or a microprocessor-based system to handle very large amount of pixels involved in the page printing.

7.3.5 Dye-Sublimation Type Printer

A dye-sublimation type printer gives photo-quality print. It uses special paper that accepts printer's colour. Paper sheet is placed on a drum. It uses a roll of plastic film which contains cyan, magenta, yellow and black dyes in bands. Each band has the same width and height as those of the paper sheet. The adjacent bands are of different colours. There are a number of heating elements which can cause the dye to sublimate. The solid ink when sublimates becomes gaseous and absorbed into the fiber of the paper. The ink does not go into liquid state. Each heating element can produce 256 different temperatures. The amount of dye that is sublimated depends on the temperature. More the temperature, more the dye is sublimated and transferred to the paper. This gives different shades of a particular colour. By this technique 16 million colours can be produced without mixing different colours as it is done in case other type of printers.

After the printing of one colour is completed, the drum reverse its direction and the paper returns to its starting position. Then the printing process for the other colour is repeated. The entire process is repeated for all the four colours. In other words the paper sheet goes through multiple passes for multicolour print. A dye-sublimation printer produces 300 dpi.

In case of dye sublimation type printer, dots are not of the same size *i.e.* they are of variable size. A heating element which is at lower temperature, produces a smaller dot than that produced by the heating element which is at comparatively higher temperature.

7.3.6 MFDs (Multi Function Devices)

Multifunction devices can perform the function of a printer, copier, scanner and fax machine. An MFD may be either laser-based or inkjet-based machine.

7.4 PLOTTERS

Plotters are output devices. They are used to produce precise and good quality graphics and drawings under computer's control. They use ink pen or ink-jet to draw graphics or drawings. Either single colour or multicolour pens can be employed. The pens are driven by motor. Drawings can be prepared on paper, vellum or mylar (polyester film). Colour transparency can also be prepared.

Pen plotters are slow devices. The graphics and drawings produced by pen plotters are uniform and precise, and they are of very good quality. They are used for low to medium volume job. A pen plotter can take from several seconds (for simple drawings) to several minutes (for complex drawings) to produce a drawing. But it takes much less time as compared to traditional hand methods of producing drawings. A complex drawing which can take about a month by traditional hand method can be produced in less than an hour using plotters.

The pen plotters may be classified into the following types:

Drum Plotters

A drum plotter contains a long cylinder and a pen carriage. The paper is placed over the drum (*i.e.* cylinder). The drum rotates back and forth to give up and down movement. The pen is mounted horizontally on the carriage. The pen moves horizontally along with the carriage left to right or right to left on the paper to produce drawings. Under the computer control both the drum and the pen move to produce the desired drawings. Several pens with ink of different colours can be mounted on the carriage for multicolour drawings. Since each pen is program selectable, a multicoloured drawing can be produced.

Microgrip Plotters

Such plotters do not use drums. The paper or any other medium is held (gripped) on both sides at the edges by pinch wheels which give back and forth paper movement. With such plotters high performance is attained at a low cost.

Flat-Bed Plotters

Such plotters use horizontal flat surface on which paper, vellum, mylar or any other medium can be fixed. The pen moves along both axes: the X axis and Y axis. The pen carriage is controlled by computer in both the axes.

Inkjet Plotters

Some plotters employ ink-jets in place of ink pens. The paper is placed on a drum and the ink-jets with different coloured ink are mounted on a carriage. Such plotters are capable of producing multicolour large drawings.

7.4.1 Hard Copy Devices for Fast Plotting of Drawings

In this technique the drawing is first displayed on a CRT screen. The hard copy device copies the drawing from the CRT screen and prepares a hard copy i.e. copy on a paper. It is less expensive and it produces output much more quickly. Whatever is on the CRT screen is copied by simply a touch of a button. It includes any combination of graphic and nongraphic (text) display. The entire process takes only a few seconds. Quality of drawings is not as good as those produced by pen plotters.

The techniques used are electrostatic method, photo plotter, ink-jet process etc. Dot-matrix printers also produce hard copy of the drawings displayed on the CRT screen. The raster approach is used to produce the drawings. The output is in the dot-matrix form.

7.5 OTHER FORMS OF OUTPUT DEVICES

7.5.1 Computer Output on Microfilm and Microfiche

A technique has been developed to record computer output on a microfilm as microscopic film images. The information recorded on the microfilm can be read with the aid of a microfilm viewing system. The computer output is first displayed on the CRT screen. It is reduced in size 48 times or more and then recorded on a roll or sheet of microfilm. A high-speed camera takes the photographs of the information displayed on the CRT. This is very fast, 10 to 20 times faster than high-speed printers. The photograph can be taken at the speed upto 32000 lines/ minute. As it is very costly system, it is suitable for a very large volume job. The film storage is less costly than to keep printed information on paper.

A microfiche is a 4 by 6 inch film sheet. The word 'fiche' is pronounced as fish. It is a French word which means card. A microfiche can store upto 270 pages of information. Some ultrafiche can store upto 1000 standard pages of information in the same space. It is easier to read a microfiche as compared to a microfilm. It is also easier to send microfiches from one place to another.

7.5.2 Voice Response

Voice recognition (voice input) by a computer is a difficult problem while voice response (voice out) is very simple. As the rules for producing voice through a speaker or a telephone system can easily be formulated, a voice response system can easily be designed and developed. The banks are the largest user of voice response techniques. The bank computer gives voice response to telephone, enquiry of customers regarding the present position of their bank accounts. The voice response system usually is considered for applications requiring low-speed human-machine interaction. Audio response system is inexpensive. Such systems can be used with PCs. The Texas Instrument Co. has developed a "Speak and Spell" system to teach children to spell and pronounce over 200 elementary words. The selected sounds to be used for an audio response can be synthesized employing a low cost single IC. A multimedia computer gives output in the form of text, graphs, images and sound.

7.6 INPUT AND OUTPUT PORT

Input and output devices can not be interfaced to a microprocessor directly because they are not provided with necessary logic circuitry needed for direct interfacing to the processor buses. They are usually interfaced to the processor buses through electronic circuitry called I/O ports. An I/O port is supposed to contain device selection logic, bus drivers, data buffers, status register,

control lines etc. Standard I/O ports in IC forms are available. Fig. 7.8 shows the interfacing of I/O devices through I/O ports. An input device is connected to the processor through an input port. The port is the place for loading or unloading data. The input device unloads data into an input port. Then the microprocessor reads the data from the input port. Similarly, an output device is connected to the processor through an output port. The microprocessor unloads data into an output port. Then the output device receives data from the output port.

An I/O port may be programmable or nonprogrammable. A nonprogrammable port is permanently connected to the processor to behave either as input port or output port. If a port is connected as an input port, it will always act as an input port. Similarly, if a port has been connected as an output port, it will always act as an output port. The operating mode to act as input port or an output port can not be altered by the user. A programmable port is also permanently connected to the processor. But it can be made to act either as input port or output port by software technique. The same port can be programmed as an input port for one problem and as an output port for another problem. When it has been programmed to act as input port, an input device will be connected to it. When it has been programmed to act as an output port, an output device is connected to it.

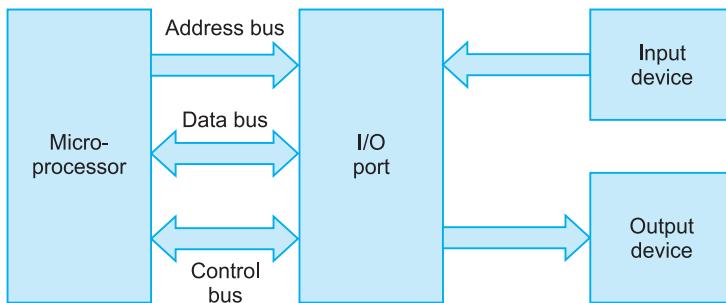


Fig. 7.8 Interfacing of I/O device through I/O port

7.6.1 Intel 8212

It is a nonprogrammable 8-bit I/O port. It can be used either as input port or output port. Once an 8212 IC is connected as input port, it will always work as input port. When it is connected as an output port, it will always work as an output port. Suppose we require one input port and two output ports in a particular system. In this case three 8212 ICs will be used. One unit will be connected as an input port and two units as output ports.

7.6.2 Programmable Peripheral Interface

A programmable peripheral interface is a multiport device. Its ports can be programmed in a variety of ways as required by the user. It is very useful device for interfacing input/output devices. Some manufacturers use the term 'Peripheral Interface Adapter' or 'Versatile Interface Adapter'.

Intel 8255A, 82C55A

The 8255A is a programmable peripheral interface (PPI). Fig. 7.9 shows its schematic diagram. It contains three 8-bit programmable ports, Port A, Port B and Port C. The Port C can further be divided into two 4-bit ports: Port C_{upper} and Port C_{lower}.

The 8255A operates in three modes: Mode 0, Mode 1 and Mode 2. In mode 0 all the three ports operate as simple I/O ports. In Mode 1 the Port A and Port B operate in strobbed input/

output mode. Pins of the Port C are used for their control (for handshaking signals, interrupt etc.). The combination of Mode 1 and Mode 0 is also possible. For example, the Port A can operate in Mode 1 and Port B in Mode 0. In Mode 2 only Port A operates. When the Port A operates in Mode 2, the Port B may operate in Mode 1 or Mode 0. Mode 2 is a strobed bidirectional mode of operation.

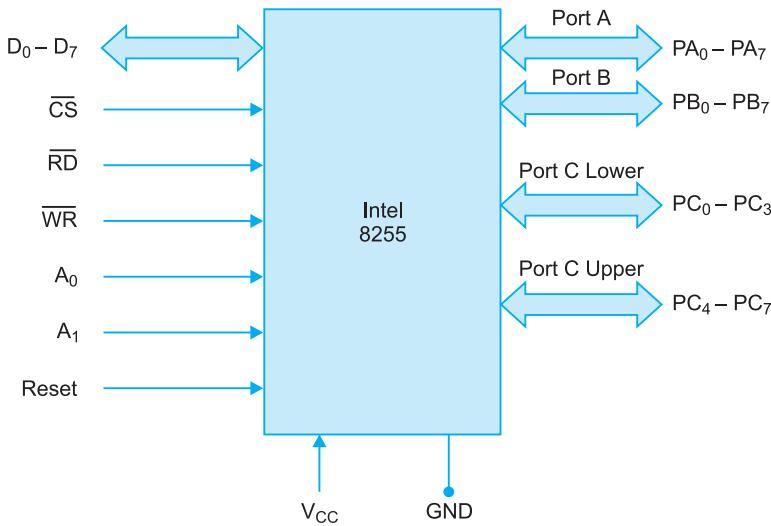


Fig. 7.9 Intel 8255

When the 8255A is programmed in Mode 1 or Mode 2, the Port C sends or accepts handshaking signals. The contents of Port C permit programmer to test and verify the status of each peripheral device and change the program flow accordingly. Pins of Port C are also used to interrupt the CPU. See details of 8255A given in Intel's Handbook.

The 82C55A is the CHMOS version of 8255A.

The programmer makes a control word that defines which port will act as an input or output and what is its mode of operation. The 8255A provides parallel ports. See more details in Ref. 1 and Intel's Handbook.

7.6.3 Parallel and Serial Ports

Computers have a few built in interfaces called ports. Peripheral devices are interfaced to the computer through these ports. Data flow in and out through these ports. Ports are of two types: parallel and serial. A parallel port allows the transfer of all the bits of a word simultaneously. In parallel interface there are multiple lines to connect the peripheral to the port. A parallel interface is used to transfer data at faster rate for high-speed peripherals such as disk and tape. A parallel interface is also known as Centronics interface as it was first popularized for interfacing printers manufactured by Centronics. Within the computer data is transferred in parallel so that the computer works at high speed.

A serial port allows serial data transfer. In serial data transfer one bit of data is transferred at a time. In serial interface only one line or a pair of lines is used to transmit data. It is used for slow-speed peripherals such as terminals. Printers employ either serial interface or parallel interface. The disadvantage of a serial/parallel port is that only one device can be connected to a port. From practical consideration there is limitation to have only two of each on a PC. USB has no such limitation.

7.7 SERIAL DATA TRANSFER

Over long distance data can not be transmitted in parallel mode as it requires too many wires resulting in high cost of transmission. The serial data transmission is economical in such a situation. In serial data transfer only one bit is transmitted at a time. As it requires only one wire or a pair of wires, it is less costly. It reduces the cost and complexity of interface components. It also minimizes the effect of noise. Serial data can be transmitted either in asynchronous mode or synchronous mode. The term asynchronous means that the transmission of data is not based on a predefined timing pattern. The asynchronous serial data transfer is commonly used. The term synchronous means that the transmission of data is based on a standard timing signal. The synchronous serial data transfer is not commonly used.

In asynchronous serial data transmission one character of data is transmitted at a time. Each data character has a bit to identify its start and 1 or 2 bits to identify its end. Thus a character is preceded by a start code and followed by a stop code. As each character is individually identified, characters can be sent at any time (asynchronously). In other words characters are sent independently. Characters can be transmitted at nonuniform rate. This method is not efficient because of the overhead for sending start bit, stop bits and parity bit. In sending an 8-bit ASCII character, four extra bits have also to be sent. This makes an overhead of 50%.

In synchronous serial data transmission blocks of characters or bits are transmitted at constant rate without start and stop codes. This is more efficient compared to asynchronous data transmission.

The serial data transfer can be classified into the following three types:

Simplex Mode

In this method of serial data transmission data are transmitted only in one direction. For example the transfer of data from a computer to a remote printer.

Half-duplex

In this method the communication can take place in either direction between the two systems using a single link. But data are transmitted only in one direction at a time.

Full-duplex

In this method each system can transmit and receive information at the same time. It requires two serial links: one for each direction. The transfer of information takes place in both directions simultaneously and independently.

7.7.1 Baud Rate

The term baud rate specifies the rate at which serial data are transmitted. It is equal to 1/(the time for a bit cell). If the time for one bit cell is 9.01 ms, the baud rate = 1/9.01 ms = 110 bd (baud). The baud rate is not the same thing as bits per second because 2 to 4 actual data bits are encoded within one transmitted bit cell. Commonly used baud rates are: 110, 300, 1200, 2400, 4800, 9600, 19200 and 38,400 Bd.

7.7.2 USART or UART

As a computer processes data in parallel, the data to be sent out using serial data transfer technique must be converted from parallel to serial. Similarly, serial data received must be converted into parallel for processing by the computer. A parallel-in-serial out shift

register and a serial-in-parallel out shift register can perform this task. In some cases hand shaking signals are also needed. Their purpose is to make sure that a transmitter should not send data faster than it can be received by the receiver. Several programmable ICs are available for serial data transfer. These devices contain all necessary circuitry required for serial data transfer. National INS 8250 has been developed for asynchronous serial data transfer. It is known as *universal asynchronous receiver-transmitter (UART)*. Intel has developed 8251, a *universal synchronous-asynchronous receiver-transmitter (USART)* which is capable of transmitting data either in synchronous or asynchronous mode.

7.7.3 Current Loop or Voltage Signal Approach

In current loop approach of sending data from the transmitting USART to the receiving USART, a current is used to represent 1 in the signal line and no current to represent 0. To isolate and protect such data transfer from electrical noise and disturbances produced by interference from AC lines, radio systems and other electrical appliances, opto-couplers are employed.

In voltage signal approach line drivers are included in the output circuit of USART to produce a voltage signal to represent 1.

7.7.4 Modem, DCE and DTE

The telephone lines are convenient and less costly means for serial data transfer. But their bandwidth is only about 300 to 3000 Hz. Therefore, digital signals can not be transmitted over these lines directly. To solve this problem, digital signals are converted to audio frequency tones which can be transmitted over telephone lines. A *modem* is a device which converts the digital signals to audio-frequency tones. At the receiving end a modem converts transmitted tones back to digital information. The term *modem* is the short form of modulator-demodulator.

Data communication equipment (DCE) includes modems and some other equipment which are used to transmit serial data over long distances. Computers and terminals which are used to send or receive serial data are called data terminal equipment (DTE). A printer is considered as DTE.

7.7.5 The RS-232C

The RS-232C is a standard for serial data transfer. It was developed by Electronic Industries Association (EIA). It specifies standard for 25 signals and handshake signals which are used between DCE and DTE. The voltage levels, maximum bit rate, rise and fall times, impedance levels and maximum capacitance for these signal lines are also described in this standard. The standard RS-232C interface is usually provided in computers for serial data transfer. The DTE connector is a male and DCE connector a female according to this standard. The DB-25P male and DB-25S female connectors are mostly used. A voltage between -3 V and -15 V under load (-25 V no load) is used for a high logic or mark. A voltage between +3 V and +15 V under load (+25 V for no load) is used for a low logic or space. A voltage supply ± 12 V is commonly used to get a low and high logic. Termination by a resistor is not necessary at the input or output end. The voltage levels are not TTL compatible. Manufacturers have developed special ICs for RS-232C to TTL conversion and vice versa. For example, MC1488 converts RS-232C signals level to TTL level. MC1489 converts TTL signal level to RS 232C level. The RS-232C interface is suitable for transmitting data reliably only upto 50 feet (16.4 m). The baud rate is limited only upto 20 K Bd. Its limitation is due to the use of a single-ended line. The single-ended line uses one line to carry the signal. The voltage reference is with respect to ground. The line is open, it is not terminated through resistance at the end. Such a line is affected by electromagnetic interference. Table 7.1 gives RS-232C pin names and signal descriptions.

Table 7.1 RS-232C Pin Names and Signal Descriptions

<i>Pin number</i>	<i>Common name</i>	<i>RS-232 C name</i>	<i>Signal direction on DCE</i>	<i>Description</i>
1		AA	—	PROTECTIVE GROUND
2	TXD	BA	IN	TRANSMITTED DATA
3	RXD	BB	OUT	RECEIVED DATA
4	<u>RTS</u>	CA	IN	REQUEST TO SEND
5	<u>CTS</u>	CB	OUT	CLEAR TO SEND
6	<u>DSR</u>	CC	OUT	DATA SET READY
7	GND	AB	—	SIGNAL GROUND (COMMON RETURN)
8	<u>CD</u>	CF	OUT	RECEIVED LINE SIGNAL DETECTOR
9		—	—	(RESERVED FOR DATA SET TESTING)
10		—	—	(RESERVED FOR DATA SET TESTING)
11			—	UNASSIGNED
12		SCF	OUT	SECONDARY RECEIVED LINE SIGNAL DETECTOR
13		SCB	OUT	SECONDARY CLEAR TO SEND
14		SBA	IN	SECONDARY TRANSMITTED DATA
15		DB	OUT	TRANSMISSION SIGNAL ELEMENT TIMING (DCE SOURCE)
16		SSB	OUT	SECONDARY RECEIVED DATA
17		DD	OUT	RECEIVER SIGNAL ELEMENT TIMING (DCE SOURCE)
18		—	—	UNASSIGNED
19		SCA	IN	SECONDARY REQUEST TO SEND
20	<u>DTR</u>	CD	IN	DATA TERMINAL READY
21		CG	OUT	SIGNAL QUALITY DETECTOR
22		CE	OUT	RING INDICATOR
23		CH/CI	IN/OUT	DATA SIGNAL RATE SELECTOR
24		DA	IN	TRANSMIT SIGNAL ELEMENT TIMING (DTE SOURCE)
25		—	—	UNASSIGNED

7.7.6 Other Standards for Serial Data Transfer

RS-422A

The RS-422A is a newer standard for serial data transfer. In RS-422A interface two wires are used. Each signal is sent differentially over the two lines. The differential lines are terminated by resistors. Differential signals are produced by differential line drivers, such as MC 3487. Differential line receivers are also used (MC 3486). Suppose the interface uses differential lines A and B. When a logic high is transmitted the line B is more positive than A. When a logic low is transmitted the line A is more positive than B. The voltage difference between the two lines is more than 0.4 V and less than 12 V. The range for the common-mode voltage on the signal lines is -7 V to +7 V. The differential line receiver rejects any common-mode electrical noise induced in the two lines. RS-422A allows a data transfer rate upto 10 MBd for a distance of 50 ft (16.4 m) or 100 KBd for a distance of 4000 ft (1220 m). IC chips are available for RS-422A to TTL conversion and vice versa.

RS-423A

The RS-423A is another standard for serial data transfer. The RS-423A interface allows a low-impedance single-ended signal to be transmitted. The low impedance signal can be transmitted over 50 ohms coaxial cables. To prevent reflections the cable is terminated at the receiving end. The data transfer rate is 300 Bd upto 4000 ft or 300 KBd for a distance of 40 ft.

RS-449 and RS-366

The connector pin number and handshake signals have not been specified in RS-422A and RS-423A standards. RS-449 is an additional EIA standard which gives these specifications. It gives specifications for 37 signal pins on a main connector and 9 additional pins on an optional connector. As the RS-449 signals are a superset of the RS-232C signals, RS-232C equipment can be interfaced with RS-449 equipment.

RS-366 is another standard which includes signals for automatic telephone dialing with modems.

7.7.7 Intel 8251A, a Serial Interface Device

The 8251 A is a programmable communication interface for serial data transmission. It is a universal synchronous/asynchronous receiver/transmitter (USART). It accepts data in parallel format from the CPU and converts them into a continuous serial stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data. The data so converted into parallel format are sent to the CPU for processing. Fig. 7.10 shows the schematic diagram of Intel 8251 A. It sends a signal to the processor whenever it can accept a new character for transmission or whenever it has received a data character for the processor. The processor sends data to or reads data from the 8251A on an interrupt basis or on a polled basis. The important signals of 8251 are as follows:

C/D	(Control/data). When it is low data can be transmitted on the data bus. When it is high control signals can be transmitted on the data bus.
WR	(Write). When it is low the processor can write data into 8251A.
RD	(Read). When it is low the processor can read data from 8251 A.
T × RDY	Transmitter ready
T × C	Transmitter clock
T × D	Line for serial data transmission
T × E	Transmitter empty
R × RDY	Receiver ready
R × D	Line for receiving data
R × C	Receiver clock
DTR	Data terminal ready
DSR	Data set ready
RTS	Request to send
CTS	Clear to send

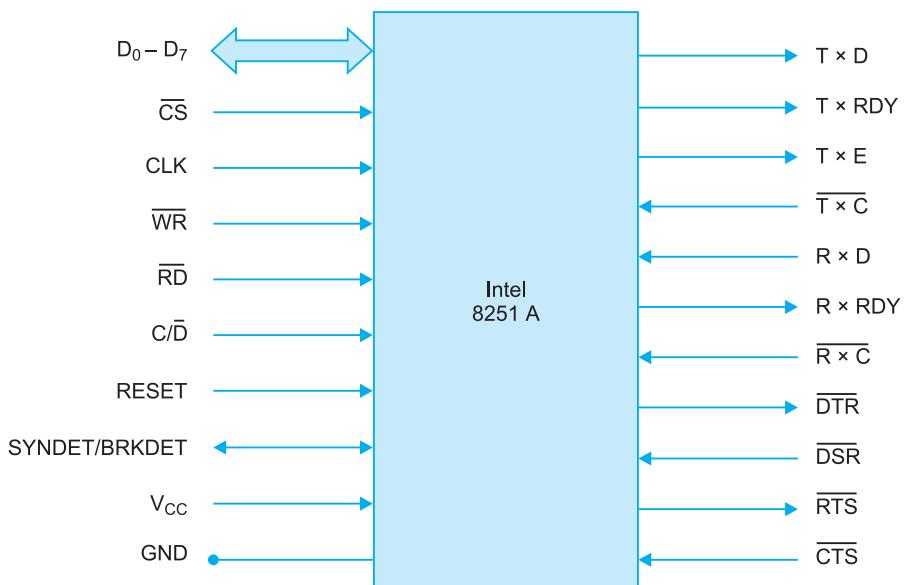


Fig. 7.10 USART Intel 8251

Before starting the data transmission or reception, a set of control words are formed. These control words are loaded into the 8251 by the CPU. There are two important control words: mode word and command word. The mode word selects the operating

mode: synchronous or asynchronous mode of operation. There are two formats for this purpose: one for asynchronous mode and the other for synchronous mode. The command word controls the actual operation of 8251. Besides these two control words, there is one more control word called status word. The programmer can read the status of 8251 at any time during operation. The status of $T \times RDY$, $R \times RDY$, $T \times E$, SYNDET/BRKDET etc. can be read.

The $T \times RDY$ is connected to interrupt pin of the processor or 8259A, an interrupt controller. By making $T \times RDY$ high the 8251 informs the processor that it can accept data. When data are received from the processor, this line goes low. The received data is converted into serial form and transmitted through $T \times D$ line. The rate of serial data transmission is controlled by the clock input at $\overline{T \times C}$ pin. The 8251A is double buffered. It means that one data character is being shifted out of the transmit shift register while another data character can be loaded into a holding buffer. When the holding buffer becomes empty, $T \times RDY$ goes high and then another character is transferred from the processor. When both the holding buffer and the transmit shift register become empty, the $T \times E$ goes high. In other words $T \times E$ goes high when the 8251A has no character to send out.

Serial data is received on $R \times D$ line. The $R \times RDY$ is normally low. When the 8251A contains a data character which is to be sent to the processor, the $R \times RDY$ becomes high. It can be connected to an interrupt line of the processor or 8259A.

\overline{DTR} , \overline{DSR} , \overline{RTC} and \overline{CTS} are standard modem control signals. If a modem has been included in the system, these control signals are connected to the corresponding modem pins. \overline{DTR} goes low when the 8251A has to transmit serial data. If modem is also ready to accept data and transmit it onward, it informs 8251A by making \overline{DSR} low. When the 8251A and modem both are ready to transmit data, the 8251A sends a low signal \overline{RTC} to the modem to initialize the data transmission. Then the modem makes \overline{CTS} low. A low logic on \overline{CTS} enables the transmission logic circuitry of 8251 A. If due to any reason data transmission has to be stopped, the \overline{CTS} is made high.

SYNDET is SYNC Detect and BRKDET is Break Detect. The SYNDET/BRKDET line goes high if there is intentional break in data transmission (for asynchronous mode only). In synchronous mode it goes high if a specified sync. character (s) is found in the incoming string of data bits.

7.8 PROGRAMMABLE INTERVAL TIMER/COUNTER

7.8.1 INTEL 8254

Intel 8254 is a programmable interval timer/counter. It has been developed to solve the common timing control problems in microcomputer system. It is compatible with all Intel and most other microprocessors. It uses HMOS technology with a single +5 V supply and is packed in 24-pin plastic or CERDIP package. Its clock rates are 5, 8, and 10 MHz for its different versions. It contains three independent 16-bit counters. As these counters are programmable the user can load a count into them, start them and stop them with instructions

in the program. It has six operating modes. It can be used to produce accurate delay, to act as programmable one-shot, rate generator, to generate square waves, software triggered strobe and hardware triggered strobe. Its modes are as follows:

Mode 0: Interrupt on terminal count

Mode 1: Hardware retriggerable one-shot

Mode 2: Rate generator

Mode 3: Square wave mode

Mode 4: Software triggered strobe

Mode 5: Hardware triggered strobe

Control Word

By defining the bits of the control format one can select one of the counters, mode of operation, binary counting or BCD counting etc. It is often desirable to read the contents of the counter while counting is in progress without disturbing the counting operation. For this purpose a counter latch command is given. There is also a control word for this propose.

7.8.2 Intel 8253

Intel 8253 is also a programmable interval timer/counter. The 8254 is a superset of 8253. The 8254 and 8253 are pin to pin compatible. They are nearly identical in function. The 8253 operates at lower frequency, 2.6 MHz. It is compatible with 8085 microprocessor. It also operates in 6 modes as mentioned in the case of 8254. It also contains three 16-bit programmable counters. Those who are interested in more details and want to program 8253, may consult Ref. 1.

7.9 CHIPSETS

A chipset is a set of two ICs which interface memory, input/output devices etc. to the CPU of a computer. It is placed on the motherboard of a computer. It provides a high-speed bus to connect memory, PCI or PCI Express and ISA bus slots for input/output and other peripherals. Chipsets of Intel corporation are: 845, 865, 875, 915, 915G, 915GV, 925 etc. The 915, 925 and their variants come with an on-board Intel Graphics Media Accelerator. Nvidia chipsets are: nForce chipsets and GeForce chipsets. The nForce chipsets are for motherboards. The GeForce is graphics chipsets. Via's chipsets are K8T800 and K8M800. The K8M800 chipset has integrated graphics capability.

7.10 DATA TRANSFER SCHEMES

In a computer data transfer takes place between two devices such as CPU and memory, CPU and I/O devices, and memory and I/O devices. Usually, memory is compatible with microprocessor while input and output devices are not. A computer is interfaced with a number of input/output devices of different speed. In such a situation a slow I/O device may not be ready to transfer data when microprocessor issues instruction for this purpose. To solve the problem of speed mismatch a number of data transfer schemes have been developed. The data transfer schemes are classified into the following two broad categories:

1. Programmed data transfer schemes
2. Direct memory access (DMA) data transfer scheme

Programmed Data Transfer Schemes

In a programmed data transfer scheme the data transfer takes place between the CPU and an I/O device under the control of a program which resides in the memory. The program is executed by the CPU. The data are transferred when the I/O device is ready for the same. The programmed data transfer scheme is used when small amount of data is to be transferred. There are the following three important types of programmed data transfer scheme:

- (i) Synchronous data transfer scheme
- (ii) Asynchronous data transfer scheme
- (iii) Interrupt driven data transfer scheme

DMA Data Transfer Scheme

In DMA data transfer data are directly transferred from an I/O device to the memory or vice versa without going through the microprocessor. The microprocessor (CPU) does not participate in this type of data transfer. This scheme is used when bulk data are to be transferred. If bulk data are transferred through microprocessor, it will be a time consuming process. The microprocessor holds on when data transfer takes place between an I/O device and memory using DMA technique. The I/O device which uses DMA technique of data transfer sends a HOLD signal to the microprocessor. Having received a HOLD request from an I/O device the microprocessor relinquishes the use of buses (i.e. the microprocessor gives up the control of the buses) as soon as the current cycle is completed. It sends HLDA (an HOLD acknowledgement signal) to the I/O device to indicate that HOLD request has been received, and the data and address buses has been relieved. In other words the microprocessor transfers the control of the buses to the I/O device. Now the I/O device gains control over the buses and transfers data directly to or from the memory without involving CPU. Thus data transfer takes place at very high speed. This scheme is used to transfer data from mass storage devices like hard disks, optical disks or high speed printers etc. When data transfer is completed the CPU regains the control over the system buses.

When data are being transferred from I/O device to the memory or vice versa using DMA technique, the microprocessor is not doing anything and it is in the hold state. The microprocessor can come out of this state only after the DMA request is withdrawn by the I/O device. The duration of the hold state depends on the speed of the I/O device, speed of the memory and the number of data bytes to be transferred. The DMA data transfer in which the I/O device relinquishes the control of the system buses only after all data bytes have been transferred, is called *burst mode data transfer*. In this mode a block of data is transferred. This type of DMA data transfer is used by magnetic disk drives where data transmission can not be stopped or slowed down without loss of the data and hence block transfer is a must. The block data transfer may require the CPU to remain inactive for relatively longer period. Another DMA data transfer scheme called *cycle stealing* allows the DMA controller to use the system bus to transfer one or perhaps several data bytes, after which it must return the control of buses to the CPU. In this technique a long block of data can be transferred by a sequence of DMA bus transactions interspersed with CPU bus transaction. Though this method reduces the maximum I/O data transfer rate, it also reduces the interference by the DMA controller in the CPU's activities. The interference can be eliminated completely by designing DMA interface in such a way that bus cycles are stolen only when the CPU is actually not using the system bus. This is known as *Transparent DMA*.

The *cycle stealing* technique can also be used for slow I/O devices. In this scheme the I/O devices request the processor for DMA data transfer. When DMA is granted the I/O device transfers one or two bytes of data and then withdraws the DMA request. After some time when the I/O device becomes again ready, it sends DMA request. It transfers one or two bytes again when DMA is granted. This process is repeated till all the data bytes are transferred.

The burst mode of data transfer is used when successive data bytes are transferred at short regular intervals. The cycle stealing is used when there is significant time delay between the transfer of two successive data bytes.

In DMA data transfer scheme data transfer takes place under the control of an I/O device and therefore, the I/O device must contain its own registers to store memory address and byte count. It must also contain electronic circuitry to generate control signals required for DMA data transfer. Generally, I/O devices are not equipped with such facilities. To solve this problem manufacturers have developed single chip programmable DMA controller to interface I/O devices to the microprocessor for DMA data transfer. Such chips are: Intel 8237A, 82C37A-5 etc. These controllers are described in subsequent sections.

7.10.1 Synchronous Data Transfer Scheme

When the processor and the I/O devices match in speed, this type of data transfer scheme can be used. The data can be transferred from or to the I/O devices using suitable instructions such as IN and OUT instructions. The IN instruction is used to transfer data from an input device or an input port to the processor. Similarly, the OUT instruction is used to send data from the processor to an output device or an output port. As I/O devices and the processor match in speed, I/O devices are ready to transfer data when the processor issues IN and OUT instructions. The I/O devices with known timing characteristics or speed-compatibility with processor are usually not available. Therefore, this scheme is rarely used alone for I/O devices. It is usually used with the other schemes discussed in subsequent sections. However, this scheme is invariably used with compatible memory devices.

7.10.2 Asynchronous Data Transfer Scheme

This type of data transfer scheme is used when the speeds of I/O devices and the microprocessor do not match and timing characteristic of the I/O devices is not predictable. In this method of data transfer the processor initiates the device to get ready and then goes on checking the status of the device. Thus the microprocessor waits till the device becomes ready to transfer data. When the device becomes ready, the processor issues instructions for data transfer. Fig. 7.11 shows a schematic diagram of asynchronous data transfer. This method is used when I/O devices are slow as compared to microprocessor. In this method the precious time of the processor is wasted in waiting. From this consideration it is an inefficient method of data transfer.

In this method two signals are used before the actual data transfer takes place. The microprocessor sends an initializing (or starting or get ready) signal to the I/O device. When data are ready the I/O device sends signal to the processor. This type of signals are called *handshake signals* and this mode of data transfer is called *handshake mode* of data transfer. The handshaking signals prevent the processor from writing new data before the device has accepted the previous data and become ready to accept new data.

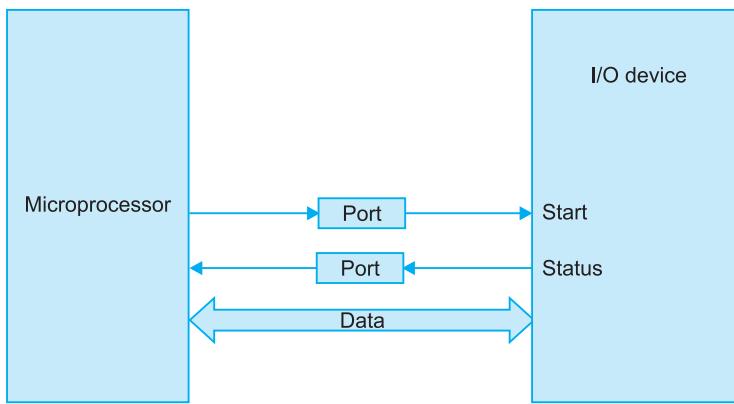


Fig. 7.11 Asynchronous data transfer

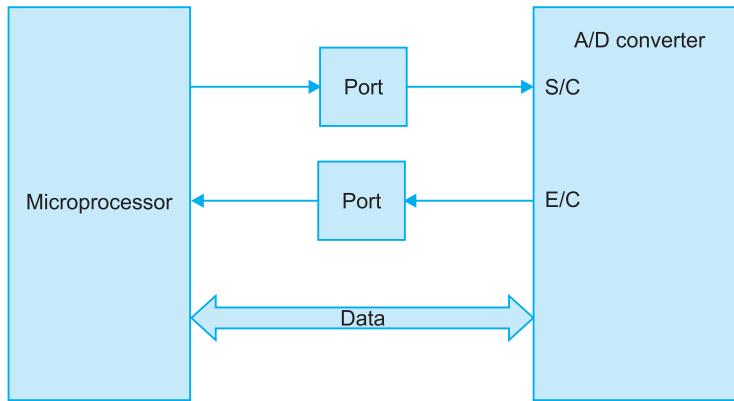


Fig. 7.12 Asynchronous data transfer scheme for an A/D converter

We can take an example of asynchronous data transfer scheme for an A/D (analog to digital) converter. Fig. 7.12 shows the schematic diagram for the scheme. First of all the microprocessor sends a start of conversion pulse to the A/D converter. The A/D converter takes some time to convert an analog signal to digital signal. It is slow as compared to microprocessor. The microprocessor has to wait till the task of conversion is over. When conversion is completed and data are ready, the A/D converter gives an end of conversion signal (E/C) to the processor. The program is prepared in such a way that the processor is kept in a loop to check the end of conversion signal E/C again and again until E/C becomes high. When E/C becomes high, indicating that conversion is completed, the processor issues instructions for data transfer.

In the above paragraph asynchronous data transfer has been discussed using software approach. The asynchronous data transfer scheme can also be implemented using hardware approach. In hardware approach READY pin (input) of the processor is used. The device status is connected to the READY pin of the processor. The microprocessor checks READY pin before the data are transferred. If READY is low the microprocessor introduces wait cycles. The state of the READY signal is checked by the microprocessor in T_2 state of a machine cycle. The microprocessor remains in wait state till the READY signal becomes high. Asynchronous data transfer scheme using READY signal is generally employed for slow memory devices.

Some very simple I/O devices do not have status indicating signals. In such a situation the microprocessor is kept in a loop to check whether data are available at the port. A simple keypad interfaced to the processor through a port may form an example of this type of asynchronous data transfer scheme.

7.10.3 Interrupt Driven Data Transfer Scheme

In this method of data transfer scheme the processor first initiates an I/O device to get ready and then goes on executing its original program instead of wasting its time in continuously checking the status of the I/O device. Whenever the I/O device becomes ready to send or receive data, it informs the processor through a specific control line called interrupt line. Having received an interrupt the processor completes the execution of the current instruction at hand. Thereafter, instead of executing the next instruction of the program which it was executing, it takes necessary steps to transfer data to or from the I/O device. First of all it saves the contents of program counter in the stack. Then the processor enters a subroutine called *interrupt service subroutine (ISS)*. The ISS saves the status of the processor in the stack and then performs data transfer from the I/O device which has interrupted the processor. After completing the data transfer it restores the processor status. Then the processor returns back to the original program which it was executing before the interrupt signal was received.

We can take an example of an A/D (analog to digital) converter to illustrate the principle of an interrupt driven data transfer, (Fig. 7.13). First of all the processor sends a start of conversion (S/C) signal to the converter. After initiating the A/D converter the processor continues the execution of the main program. The A/D converter takes some time to convert analog signal to digital signal. When the task of conversion is completed the A/D converter gives an end of conversion signal (E/C) to the microprocessor. The E/C signal is connected to an interrupt line INTR of the processor. On receiving an interrupt signal the processor will take all necessary steps to transfer data from the A/D converter. On the completion of data transfer the processor will return to execute the main program again.

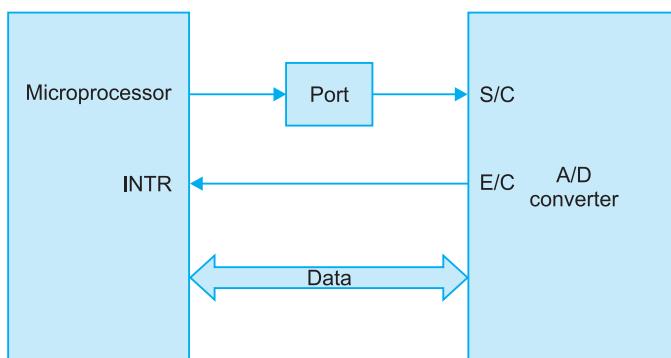


Fig. 7.13 Interrupt driven data transfer scheme for an A/D converter

Interrupt driven data transfer scheme is used to interface relatively slow I/O devices such as character printers, A/D converters etc. In the interrupt driven data transfer scheme the time of the processor is efficiently utilized as compared to asynchronous method of data transfer. In this method of data transfer the valuable time of the processor is not wasted in waiting while I/O devices are getting ready or preparing data for the processor. The processor utilizes its time efficiently in executing its program. When data become ready and the processor is interrupted by the I/O device, it executes ISS to transfer the data.

Device Polling

When a number of I/O devices are to be connected to a single interrupt line of the processor, they can be connected through an OR circuit. When an I/O device interrupts the processor, it has to be ascertained which I/O device has interrupted. The processor must check each device in turn whether it has interrupted or not. This method of checking is called *device polling*. There is likelihood that more than one device may interrupt the processor at the same time. To solve this problem priority may be assigned to the devices. The processor may poll the devices one by one in turn in the given priority order. The device polling scheme is a time consuming process and hence slow. The method of polling may be done either by software approach or hardware approach.

Vectored Interrupt

When a processor has several interrupt lines and one I/O device is connected to each interrupt line, there is no question of ambiguity regarding which device has interrupted. When a device interrupts the processor, the processor immediately comes to know which device has interrupted and it executes its ISS. This type of interrupt scheme is called *vectored interrupt*.

There may be a scheme which can employ vectored interrupt as well as polling. When a large number of I/O devices are to be connected to a few interrupt lines, a group of I/O devices is connected to each interrupt line. When an interrupt occurs, the processor immediately knows which group has interrupted. Then it checks by polling which device of that group has interrupted.

7.10.4 Programmable Interrupt Controllers (PIC)

When several I/O devices are to be connected to the same interrupt line of the processor, they can be connected through an interrupt controller. An interrupt controller operates as an overall manager in an interrupt driven system. It accepts interrupt signals from I/O devices and determines which of the interrupting devices is of the highest priority. Each I/O device has a special program or subroutine to implement its data transfer. This is known as ISS (interrupt service subroutine). Having decided the priority, the controller interrupts the processor through its INT line. The microprocessor sends acknowledgement through INTA line. Then the controller sends necessary information to the processor so that the processor executes the ISS for the interrupting device. The information contains the starting address of the ISS.

Intel 8259A

The Intel 8259A is a programmable interrupt controller. It can handle up to 8 vectored priority interrupts for the CPU. It uses NMOS technology and requires a single +5 V supply. It is packed in a 28-pin DIP or 28-lead PLCC package. Fig. 7.14 shows the schematic diagram of Intel 8259A. Fig. 7.15 shows the interfacing of 8259A chips to a microprocessor. INTA from the microprocessor goes to INTA pins of all 8259 chips. If there is only one 8259A chip, its INT will be directly connected to INTR pin of the microprocessor.

Its important signals are as follows:

IR₀-IR₇

Interrupt requests. An I/O device sends interrupt signals through one of these lines. The interrupting device makes IR_n high and keeps it high until it is acknowledged. These are asynchronous inputs.

D₀-D₇	Bidirectional data bus. Control, status and interrupt vector information is transmitted through these lines.
CS	Chip select.
WR	Write. When \overline{WR} is low the 8259 accepts command word from the CPU.
RD	Read. When \overline{RD} is low the 8259 sends various status signals on the data bus for the CPU.
INT	Interrupt. The 8259 interrupts CPU through this line.

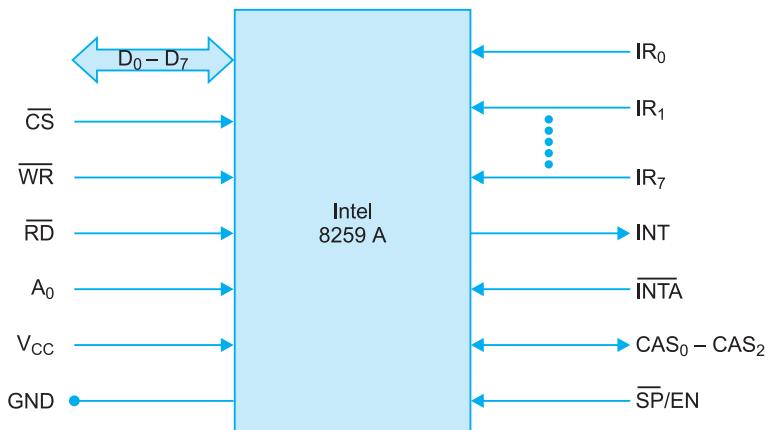


Fig. 7.14 Programmable interrupt controller Intel 8259A.

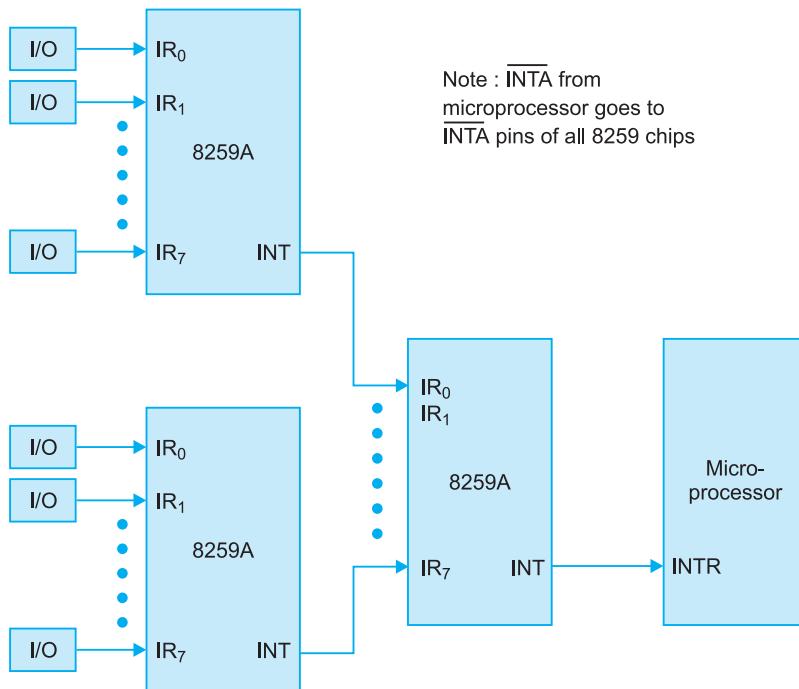


Fig. 7.15 Interfacing of 8259A to microprocessor

INTA.	Interrupt acknowledge. The CPU sends acknowledgement through this line to 8259. This enables 8259 to send necessary information to CPU.
CAS₀-CAS₂.	Cascade lines.
A₀.	Address line. This pin acts in conjunction with \overline{CS} , \overline{WR} and \overline{RD} .
SP / EN.	Slave program/enable buffer.

The 8259A is compatible with 8086, 8088 and 8085 microprocessors. Its circuitry is static, requiring no clock input. The 8259A chips can be cascaded to handle 64 vectored priority interrupts without additional circuitry. Its other versions are: 8259A-2 and 8259A-8.

Intel 82C59A-2

It is a high performance CHMOS version of Intel 8259A-2. It is compatible with 80C86/88, 8086/88 and 8080/8085 microprocessors.

7.10.5 Programmable DMA Controllers

The basic principle of DMA data transfer has already been discussed in the beginning of the section 7.10. The necessity of DMA controller has also been explained. The DMA data transfer scheme is used where bulk data transfer is required directly from an I/O device to memory and vice versa. In this section some DMA controllers are described.

Intel 8237A, 8237A-4, 8237A-5

The 8237A is a high performance programmable DMA controller. It has 4 independent DMA channels. It has the feature of independent autoinitialization of all channels. Each channel can be individually programmed to autoinitialize to its original conditions following an end of process (EOP). Each channel has a full 64 K address and word count capability. Memory-to-memory data transfer capability is provided. Data upto 1.6 MB/sec can be transferred at 5 MHz by 8237A-5. It can be expanded to any number of channels by cascading additional controller chips. It has the ability of address increment and decrement. It has also the feature of software DMA requests. It is compatible to 8086, 8088, and 8085 processors. It uses NMOS technology. It is available in 40-pin lead cerdip and plastic packages.

Fig. 7.16 shows the schematic diagram of 8237A. Its important signals are as follows:

DREQ₀-DREQ₃.	DMA request. An I/O device uses one of these lines to send its DMA request. DREQ should be maintained high until corresponding DACK goes active.
DACK₀-DACK₃.	DMA acknowledge. When DMA is granted the I/O device is informed by 8237A through this line.
DB₀-DB₇	Bidirectional 3-state data bus lines. During DMA cycles they carry 8 MSBs of memory address which are to be latched in an external latch. The address bits are transferred in the beginning of the DMA cycle. The bus is then released to carry data during the rest of the DMA cycle.
A₀-A₃	These are 4 bidirectional LSBs of the address lines. In idle cycle these are inputs and are used by the CPU to address the register to be loaded or read. In active cycle they are outputs and carry the lower 4 bits of the output address.

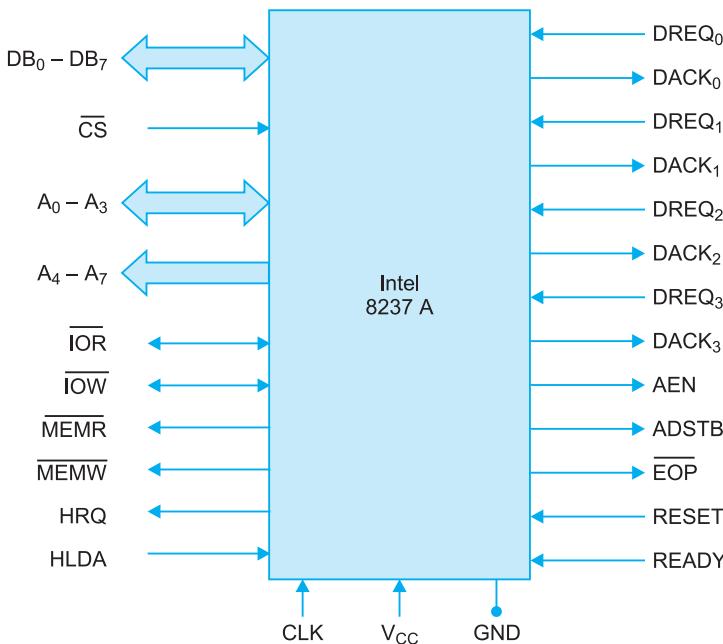


Fig. 7.16 Programmable DMA controller 8237A

A₄-A₇

These are 4 MSBs address lines. They give 3-state outputs. These are enabled only during DMA cycle.

HRQ

Hold request. The 8237A sends hold request to the CPU through this line.

HLDA

Hold acknowledge. The CPU sends acknowledgement to the 8237A through this line. It indicates that the CPU has relinquished the control of the system buses.

MEMR

Memory read. It is low when data are transferred from memory to the I/O device.

MEMW

Memory write. It is low when data are transferred from the I/O device to the memory.

IOR

I/O read. It is low when data are transferred from the I/O device to the memory.

IOW

I/O write. It is low when data are transferred from memory to the I/O device.

AEN

Address enable. It performs several tasks. It enables 8-bit latch containing upper 8 address bits onto the system address bus. It disables other system bus drivers during DMA transfer.

ADSTB

Address strobe. It is active high. It is used to latch the upper 8 address bits into an external latch.

EOP

End of process. When the transfer of programmed number of bytes is

completed the 8237A makes $\overline{\text{EOP}}$ low. It unasserts its hold request to the processor and makes $\overline{\text{AEN}}$ low to release buses back to the processor.

READY	Ready. It is an input used to extend the memory read and write pulses from 8237A to accommodate slow memories or I/O devices.
--------------	---

The 8237A has a number of internal registers. It stores the memory address, byte count etc. in internal registers. When the CPU grants DMA data transfer to some I/O device, it loads the device number, memory address, byte count, direction of data transfer and other necessary information into the registers of 8237A. For data transfer from the memory to the I/O device $\overline{\text{MEMR}}$ and $\overline{\text{IOW}}$ are made low. For the data transfer from the I/O device to the memory $\overline{\text{MEMW}}$ and $\overline{\text{IOR}}$ are made low. The 8237A sends 8 MSBs of the memory address over DB-bus. These 8 MSBs of the address are latched into an external latch. The 8 LSBs of the address are sent on A_0 - A_7 lines. As the 8237A does not have 20 address lines, so it sets up the bus address lines using some dedicated latches. When the processor 8086 supplies memory addresses, those are 20-bit addresses. The address bits A_{16} - A_{19} remain latched in external latches. These bits are available and they are utilized by 8237A during DMA data transfer. $\overline{\text{EOP}}$ goes low when all programmed bytes of data are transferred. In fixed priority $\overline{\text{DREQ}_0}$ has the highest priority and $\overline{\text{DREQ}_3}$ has the lowest priority.

Some terms used in DMA data transfer are as follows:

Idle Cycle

When there is no request from any channel, the 8237A enters the idle cycle. In this cycle the controller samples the DREQ lines every clock cycle to detect if any channel is requesting a DMA cycle. The controller also samples $\overline{\text{CS}}$ to check whether the microprocessor is attempting to write or read internal registers of the controller.

Active Cycle

When the controller gets DMA request, it outputs an HRQ to the microprocessor and enters the active cycle. DMA data transfer takes place in the active cycle.

The operating modes of 8237A are as follows:

Single Transfer Mode

In this mode the device is programmed to perform one transfer only. The word count is decremented and the address incremented or decremented after each transfer.

Block Transfer Mode

In this mode the 8237A is activated by DREQ to continue data transfer during the service until a TC (terminal count) caused by word count or an external end of process (EOF) is encountered. DREQ need only be held active until DACK becomes active. An autoinitialization occurs at the end of the service if the controller is programmed for it.

Demand Transfer Mode

In this mode the device is programmed to continue data transfer until a TC or external $\overline{\text{EOP}}$ is encountered or until DREQ goes inactive. Thus the data transfer may continue until the I/O device exhausts its data capacity.

Cascade Mode

In this mode more than one 8237A are cascaded as shown in Fig. 7.17. This permits the DMA requests of the additional 8237A units to propagate through the priority network circuitry of the preceding unit.

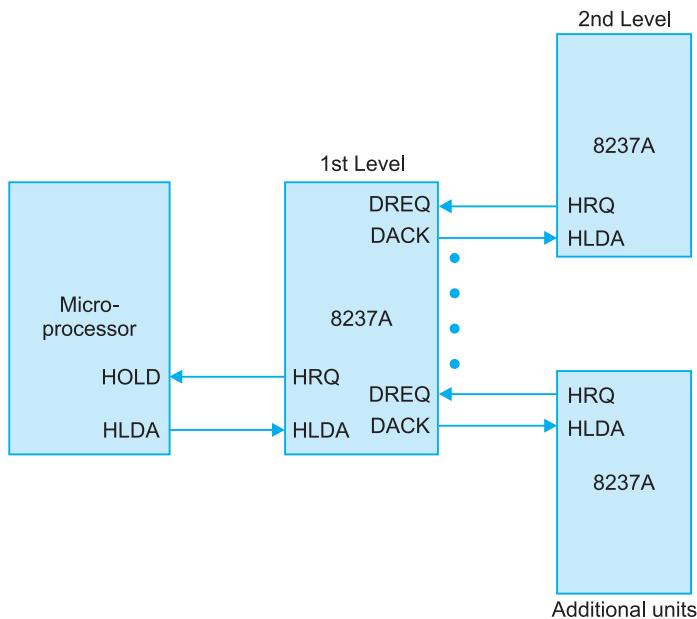


Fig. 7.17 Cascade connection of 8237A units.

Intel 82C37A-5

It is CHMOS version of 8237A-5. It is pin to pin compatible with NMOS 8237A-5.

Intel 8257/8257-5

It is a 4-channel DMA controller. The 8257-5 is compatible with 8085 processors. Each channel generates a sequential memory address which permits the I/O device to transfer data to or from the memory. Each channel contains two 16-bit registers: DMA address register and byte count register. These registers are initialized before a channel is enabled. Initially, the DMA address register is loaded with the address of the first memory location to be accessed. During DMA operation it stores the next memory location to be accessed in the next DMA cycle. Fourteen LSBs of the byte count register store the number of bytes to be transferred; $2^{14} = 16384$ bytes of data can be directly transferred to or from the memory.

Its operations are similar to those explained in case of 8237A. Pins are also similar. This chip contains $\overline{DACK_0}$ – $\overline{DACK_3}$ pins and a terminal count pin, TC. TC becomes high when all programmed bytes are transferred. The main features of 8257 are: priority DMA request logic, auto load mode, channel inhibit logic, etc.

7.10.6 Multifunction VLSI Chips

With the rapid advancement in VLSI technology it has become possible to build several functions on a single chip IC. DMA controller, interrupt controller, programmable timer/

counters, system power management, DRAM controller, etc. can be placed on a single-chip. Today a handful of ICs are used on the motherboard. VLSI chips for several functions are available today, and they are used in a digital computer, for example, Intel 915, 915G, 915GV, 925 etc. These are for Pentium 4 based computers.

7.11 SINGLE-CHIP MICROCOMPUTERS (MICROCONTROLLERS) AND EMBEDDED MICROPROCESSORS

7.11.1 Single-Chip Microcomputers

With the development of VLSI technology it became possible to fabricate a digital computer on a single IC chip. A digital computer fabricated on a single IC chip is called **single-chip microcomputer**. Single-chip microcomputers are widely used in industrial control, process control, consumer and home appliances control, instrumentation, etc. Since a single-chip microcomputer is used for control applications, it is also called **microcontroller**. It is very small and compact. It forms a part of the device or equipment which is to be controlled. A simple microcontroller contains a CPU, memory (RAM and ROM/EEPROM/Flash memory) and I/O lines. A powerful microcontroller may contain a few other components (which are needed for industrial and other control applications) such as A/D converter, D/A converter, DMA channels, pulse-width modulator, watch-dog timer, high-speed input/output system, wave generator, interrupt processor, etc.

Intel introduced 8048 series of single-chip 8-bit microcomputers in 1976. Its family members were : 8048, 8748, 8042, 8049, etc. In 1980, Intel developed more powerful 8-bit microcontrollers, Intel 8051 series. Microcontrollers of this series are faster, have enhanced instruction set including instructions for multiplication and division, full duplex serial port, power saving modes, etc. Its family members are : 80C51BH, 80C31BH, 87C51, 8X52/54/58, 8XC51FX, 8XL52/54/58, 8XL51FA/FB/FC, 8XC51RA/RB/RC, 8XC51GB, 8XC51SL, 8XC152JA/JB/JC, etc. X = 0 for ROMless version, X = 3 for versions with ROM, and X = 7 for versions with EPROM or OPT (one time programmable). Its different versions have different features designed for specific applications. Specific features such as A/D conversion, pulse-width modulation, serial communication facility, etc. are needed by certain applications. Intel has also developed 8XC151 and 8XC251 series of microcontrollers. These are more powerful and faster than 8051 series of microcontrollers. 8-bit microcontrollers are used for simple industrial control applications, consumer appliances, counting events, measurement of electrical and physical quantities, instrumentation, etc.

Other companies have also developed microcontrollers. Examples are : Texas Instruments 4-bit microcontroller - TMS1000, Motorola's 8-bit microcontroller - 6801 and MCS68HC11, Zilog's Z8, NEC's 7800 series, etc.

In 1983, Intel introduced 16-bit microcontrollers, 8096 series. Microcontrollers of this series are more powerful and faster than 8-bit microcontrollers. They are used in sophisticated industrial control, intelligent computer peripherals, instrumentation, etc. Examples are : control of large horse power electric motors, robots, guidance and control of missiles, etc. Intel developed improved versions of 16-bit microcontrollers, 80196 series. Its family members are : 80C194, 83C194, 8XC198, 8XC196KB/KC/KD/KR/KQ/JR/JQ/KT/NT/NP/NU/MC/MD/MH, 80296SA, etc.

Motorola has developed 32-bit RISC microcontroller, MPC505. It is used in advanced communication devices, speech processing system, automotive modules, etc. IBM has

developed 32-bit RISC embedded microcontroller, 403GA. It is used in office automation equipment (printers, copiers, fax machines, etc.), consumer electronics, video games, telecommunication, networking, personal digital assistant (PDA), etc.

7.11.2 Intel 8044 Microcontroller with Communication Controller

The 8044 is based on the 8051 core. The 8044 replaces 8051's serial port with serial interface unit, SIU. The SIU is an intelligent high performance HDLC/SDLC serial communication controller. HDLC stands for high-level data link control and SDLC for synchronous data link control. The CPU and SIU operate concurrently. The 8044 family has the following members:

- (i) 8344AH: An 8051 microcontroller with SIU.
- (ii) 8044AH: An 8344 with 4K bytes on-chip ROM program memory.
- (iii) 8744AH: An 8344 with 4K bytes on-chip EPROM program memory.

This family is also known as RUPI-44 family. It is used for sophisticated real-time control applications such as instrumentation, industrial control, and intelligent computer peripherals. The major features are:

- (i) A powerful 8-bit CPU.
- (ii) Two 16-bit timer/counters.
- (iii) Five source interrupt capability with two programmable priority levels. Any one of the priority levels can be used.
- (iv) The CPU executes most instructions in 1 μ sec. It can perform 8×8 multiplication in 4 μ sec.
- (v) It contains a Boolean processor.
- (vi) 4K bytes ROM program memory and 192 bytes RAM data memory.
- (vii) 64K bytes program memory address space. The lower 4K bytes of the 64 KB program memory address space is filled by internal (on-chip) ROM or EPROM. The remaining is for external program memory. The program memory uses 16-bit address.
- (viii) 64 KB external data memory address space. The external data memory can use either 8 or 16-bit address. The internal data memory use 8-bit address, which provide 256 addresses. The lower 192 addresses are for on-chip RAM. There are 35 special function registers which use various locations in the upper address space.
- (ix) 32 programmable I/O lines.
- (x) The SIU supports data rates upto 2.4 MB/sec when externally clocked and 375 KB/sec. when self clocked.
- (xi) 40-pins, 12 MHz clock and +5 V supply.

7.11.3 Embedded Microprocessors

There are certain control applications which require large amount of data handling. For such an application a microprocessor-based system is used instead of a microcontroller. Embedded microprocessors have been developed for data control applications. A microprocessor-based system which is used for data control application, uses embedded microprocessors. Memory requirement for data control applications differs from application to application, and therefore, memory is not included on the embedded processor chip. Depending upon the requirement

external memory chip is used in the embedded processor based system. Data control applications require multifunction control such as DMA control, data processing, data formatting, I/O control, etc. Embedded processors have been designed to meet such requirements.

Intel has developed 16-bit and 32-bit embedded processors for data control applications. 32-bit embedded processors are : 80960 (or i960), embedded versions of 486 and 386 processors, 80376, etc. 16-bit embedded processors are : 80186 and 80188. The 8096 is a powerful embedded processor and it is widely used for sophisticated applications which require large data processing. Examples are : laser printers, complex industrial automation, image processing, graphics, networking, etc.

Embedded versions of 486 are : 486GX, 486SX, etc. The embedded versions of 386 are: 386CXSA, 386CXSB, 386EX and 386SXSA. The 80186 is a 16-bit highly-integrated processor. It contains a CPU, interrupt controller, DAM channels, clock generator, RAM refresh control unit, etc. Its versions are : 80C186, 80C186XL/EA/EB/EC, etc. The 80188 is also an integrated chip and it has only 8 data lines whereas 80186 has 16 data lines. Its different versions are: 80C188, 80C188XL/EA/EB/EC.

7.12 DSP (DIGITAL SIGNAL PROCESSOR)

The processors specially designed to process digital signals is called DSP. It receives digitised signals, performs some mathematical operations and sends result to an output device or a host microprocessor. Most of DSP systems accept analog signal using ADC (Analog-to-Digital Converter), process them using DSP and send the result to a D/A converter or a host microprocessor. DSPs implement integration, differentiation, complex Fast Fourier Transform etc. using hardware. Many companies make DSP chips; for example, Texas Instrument's TMS32025, NEC μ PD7281, Motorola MC56000, National LM32900, Matsushita MN1900, Fujitsu MB8764, etc.

Earlier DSP chips also contained analog multiplexer, analog demultiplexer, A/D converter, D/A converter, etc. The later DSP chips excluded analog input and analog output sections because they can be best implemented with different types of technology. Instead they concentrated on increasing processing speed, increasing RAM and ROM capacity, improving bit manipulating capacity and adding floating-point processing capability. A DSP system employs other ICs for analog input and analog output sections. The input section includes an analog multiplexer, filters, sample and hold, and A/D converter. The output section includes D/A converter, demultiplexer, filters, amplifiers, etc.

7.13 I/O PROCESSOR

A microcontroller or a microprocessor-based system which is used to control an input/output device is called I/O processor. It relieves CPU from the task of input/output operations. A modern computer has several I/O devices. Some of them are very fast devices. To perform I/O operations I/O programs are to be executed. If CPU is kept busy in executing I/O programs, its appreciable time is wasted in it and it gets less time to perform its own task of data processing. Therefore, to make a modern computer efficient and faster, important I/O devices are provided with I/O processors. For example, a keyboard contains a microcontroller which scans the pressed key and gives its ASCII code (or any other binary code). The code is stored in the memory of the microcontroller. When certain number of keys are pressed or

after certain time intervals the microcontroller interrupts the CPU and sends ASCII codes of the pressed keys to the CPU for processing. An another example is a laser printer. It is a fast device. It uses a microprocessor-based system. Intel's 960, a fast embedded microprocessor is used for this purpose. A laser printer has to handle a large amount of data (i.e. data for pixels) to print a page. The CPU loads information which is to be printed into the memory of i960-based I/O processor. Now I/O processor performs the task of data handling for printing the information. When printing work is over, it interrupts CPU and asks for more information, if any for printing.

General purpose I/O processor called universal peripheral interface has also been developed such as UPI-452. It is a slave I/O processor. A slave processor works on the direction of the CPU. An UPI contains interfacing circuit besides a microcontroller. The interfacing circuit connects it to the host CPU.

7.13.1 UPI-452

The UPI-452 (universal peripheral interface) is a CMOS programmable I/O processor. It is a general purpose slave I/O processor. Actually the UPI-452 is a slave microcontroller. It incorporates an 80C51 (a single-chip microcomputer) with double program and data memory, a slave interface which permits the connection of UPI-452 directly to the host system bus as a peripheral, a FIFO buffer module, a two channel DMA controller and a fifth I/O port. Fig. 7.18 shows the functional block diagram of UPI-452. Its main features include 256×8 -bit internal RAM, 8-KB ROM/EPROM, two 16-bit timers/counters, Boolean processor, bit addressable RAM, 8 interrupt sources, programmable full duplex serial channel, 34 additional special function registers, 40 programmable I/O lines, 128-byte bidirectional FIFO slave interface, two DMA independent channels, 64K program memory space, 64K data memory space, 68-pin PGA and PLCC package, software compatible with the MCS-51 family of single-chip microcomputers etc.

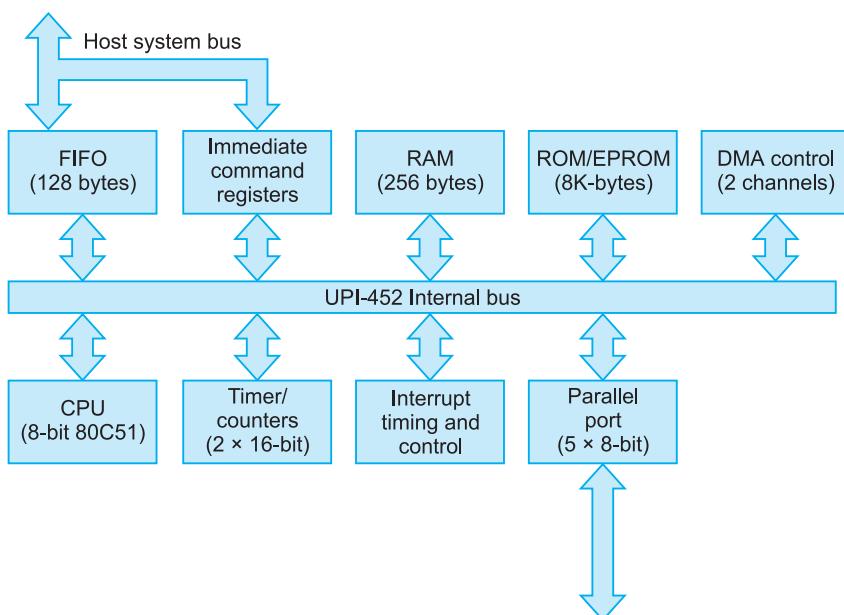


Fig. 7.18 Functional diagram of UPI-452

The on-chip DMA controller permits high speed data transfer between any of the three writable memory spaces: internal data memory, external local expansion bus data memory and the special function register array. The special function register array is treated as a set of unique dedicated memory addresses which can be used as either the source or destination address of a DMA transfer. Each DMA channel can be independently programmed. For this purpose there are dedicated special function registers for mode, source and destination addresses and byte count to be transferred. Up to 64 KB can be transferred in a single DMA operation. Each DMA channel may operate in either block mode or demand mode. In block mode of data transfer there are two techniques: burst mode or alternate cycle mode. In burst mode processor halts its execution and DMA data transfer takes place. In alternate cycle mode DMA cycle and instruction cycle take place alternately. The demand mode data transfer may be of two types: (i) FIFO or serial channel demand mode and (ii) external demand mode. In demand mode, a DMA data transfer takes place when it is demanded. Demand can be accepted from an external device (through external interrupt pins) or from either the serial channel or FIFO flags. In these cases the DMA data transfer can be synchronized to an external device, the FIFO or the serial port.

7.13.2 UPI-41, 42

These are general purpose universal peripheral interfaces to control I/O devices. They are essentially slave microcontrollers with interface included on the chip. They contain microcontrollers 8041AH, 8042AH, 8741AH or 8742AH. The UPI-41 operates at 6 MHz and UPI-42 at 12 MHz. They are compatible with all Intel and most other microprocessors. They contain 8-bit CPU plus ROM/EPROM, RAM, I/O, Timer/Counter, and DMA with interrupt or polled operation support in a single package. UPI-42 has 8042AH or 8742AH microcontroller. UPI-41 contains 8041AH or 8741AH microcontroller. The ROM/EPROM in different versions are as shown in Table 7.2.

Table 7.2 UPI-41, 42 Family

UPI device	ROM bytes	EPROM bytes	RAM bytes
8042 AH	2K		256
8742 AH		2K	256
8041 AH	1K		128
8741 AH		1K	128

Other important features are: 8-bit timers/counters, 18 programmable I/O lines, one 8-bit status and two data registers for asynchronous slave-to-master interface, expandable I/O, sync, mode available, over 90 instructions, intelligent programming algorithm, available in 40-lead cerdip, 40-lead plastic and 44-lead plastic leaded chip carrier packages, etc.

7.14 ARITHMETIC PROCESSORS

General purpose microprocessors such as 8086, 8088, 80286, 80386 etc. are not optimized to perform complex numerical calculations, CRT graphics manipulations or word processing. For these purposes specialized coprocessors have been developed. For example, the 8087 is a numeric data coprocessor for numerical calculation, and 82786 is a graphic coprocessor. These coprocessors operate in parallel with CPUs. Powerful 32-bit and 64-bit processors such as Intel 486, Pentium family of processors, etc. contain FPU (math processor unit) on the processor chip itself.

There are two ways to interface an arithmetic processor to a CPU. In one approach it is treated as a peripheral device. Such units are called peripheral processor. The CPU sends data and instructions for processing to such an unit and receives results from it. Examples of peripheral arithmetic processors are AMD 9511/12, Intel 8231A, etc. In another approach the arithmetic processor is connected as an extension of the CPU. The instructions and registers of arithmetic processor are extensions to those of the CPU. The instruction set of the CPU includes a special subset of opcodes reserved for arithmetic processor. Arithmetic processor of this type is called *coprocessor*. Unlike a peripheral processor, a coprocessor is designed for a particular CPU family, whereas a peripheral processor can be used with any host CPU. Each CPU is designed with a coprocessor interface that contains special control circuitry to link CPU with the coprocessor, and special instructions for coprocessor. Examples of numeric data coprocessors are Intel 8087, 80287, 80387 etc.

7.14.1 Intel 8087

It is a high performance numeric data coprocessor. Its other versions are 8087-1 and 8087-2. It uses HMOS III technology and is packed in 40-pin package. Clock rates for 8087 are 5 MHz, for 8087-2, 8 MHz and for 8087-1, 10 MHz. It has been designed to work with 8086, 8088, 80186 and 80188 microprocessors. It has 68 numeric processing instructions which are added to 8086/8088 instructions set. The 8087 includes the standard 8086/8088 instruction set for general data manipulation and program control. Its 68 numeric instructions are for extended precision integer, floating-point, trigonometric, logarithmic and exponential functions. It includes instructions for arithmetic operations. It contains eight 80-bit registers which are added to the register set of the CPU. It can handle 16, 32 and 64-bit integers; 64, 80-bit floating-point and 18-digit BCD operands. It also provides the capability to control round off, underflow, and overflow errors in each calculation. The trigonometric, logarithmic and exponential functions are built into the coprocessor hardware. At hardware level it is treated as the extension to the CPU, providing register, data types, control and instruction capabilities. As a coprocessor to 8086 or 8088 the 8087 is connected in parallel with the CPU. At the programmer's level the 8087 and the CPU are treated as a single unified processor. The 8087 executes instructions as a coprocessor to a maximum mode CPU.

7.14.2 Intel 80287, 80C287A

The 80287 is a HMOS numeric processor extension for 80286 microprocessor. It has 80-bit internal architecture. Its various versions are 80287-3, 80287-6, 80287-8 and 80287-10 to operate at different clock rates. It has over 50 instructions which are added to the instruction set of 80286. It is object code compatible with 8087. It executes instructions in parallel with an 80286. It directly extends 80286 instruction set to trigonometric, logarithmic, exponential and arithmetic instructions for all data types. The data types include 32, 64 and 80-bit floating point; 32, 64-bit integers and 18-digit BCD operands. It can perform full-range transcendental operations for sine, cosine, tangent, arctangent and logarithm. It contains 8×80 -bit individually addressable numeric register stack. Protected mode operation completely conforms to the 80286 memory management and protection. It is packed in 40-pin cerdip package.

The 80C287A is a CHMOS III math coprocessor designed for higher speed and low power consumption.

7.14.3 Intel 80387 DX

It is a high performance 80-bit CMOS IV numeric processor extension for 80386 DX microprocessor. It has 80-bit internal architecture. Its performance is 5 to 9 times more than that of 8087 and 80287. It handles 32, 64 and 80-bit floating-point; 32 and 64-bit integers and 18-bit BCD operands. It directly extends 80386 DX CPU instruction set to include trigonometric, logarithmic, exponential and arithmetic instructions for all data types. It is upward object-code compatible with 8087 and 80287. It has built-in exception handling. It effectively extends the registers and instruction set of 80386 microprocessor. It adds over 70 mnemonics to 80386 DX instruction set. It is packed in 68-pin PGA package. Programmers may use the registers of 80387 DX in addition to the registers of 80386 DX microprocessor.

Intel 80387 SX

It is a high performance 80-bit numeric processor extension for 80386SX microprocessor. Its internal architecture is of 80-bits. Its performance is 2–3 times more than that of 8087 and 80287. It is compatible with 80387 DX. It is upward object-code compatible with 8087 and 80287. It can handle 32, 64 and 80-bit floating-point; 32, 64-bit integer and 18-digit BCD operands. It directly extends 80386SX CPU instructions set to trigonometric, logarithmic, exponential and arithmetic instructions for all data types. It adds over seventy mnemonics to the instruction set of 80386SX microprocessor. It has built-in exception handling. It is available in 68-pin PLCC package. Programmers may use the registers of 80387SX in addition to those of 80386SX microprocessor.

7.14.4 Motorola MC68881

It is a floating-point coprocessor of Motorola. It contains eight 80-bit general/purpose registers, and 3 status and control registers. It contains a number in extended-precision format with sign, 15-bit exponent, and 64-bit mantissa.

7.15 GRAPHICS PROCESSORS

Several graphics chips specially designed for graphics/video have been developed. Their function is to generate pixels and process them and display them. The terminology used are graphics processors, graphics coprocessor, video accelerators, etc. Intel has developed Intel 740-3D graphics chip. It is optimised for Intel's multimedia processor based PCs, using a hyper-pipelined 3-D architecture with additional 2-D acceleration. Like most 3-D graphics chips, I-740 will be marketed in performance, not the main stream category. It will go mostly for heavy multimedia uses such as games and movies. Other graphics processors are: IBM's 8514/A, Texas Instruments TMS34010 and TMS34020. Intel i860, a 64-bit processor has graphics hardwares to implement 3-D graphics. Earlier Intel chips are : i750 a video processor, 82786 a graphics coprocessor, etc. GPU is Graphical Processing Unit. SLI (Scalable Link Interface) technology or SLI based machines allow multiple GPUs to be placed on the same motherboard, thereby nearly doubling the performance.

PROBLEMS

1. What do you understand by 'peripheral'? Explain with examples. Discuss the function of input and output devices.
2. What is the function of a keyboard in a computer? What is a keyboard encoder? What are its functions?

3. Explain the function of the following keys:
 - (i) Alt key
 - (ii) Function keys
 - (iii) Enter or return key
 - (iv) Home key
 - (v) Pg Up, Pg Dn, Insert, Delete keys
4. Discuss the construction of mechanical, capacitive and Hall effect key-switches. What do you understand by two-key roll-over and two-key lockout?
5. Discuss the function of a mouse, light pen, joystick and trackballs.
6. What is an optical scanner? Describe optical flat-bed scanner, optical mark reader and optical bar-code reader. What is a point-of-sale terminal?
7. Discuss magnetic-ink character reader with its area of applications.
8. Discuss the working principle of voice input and output system with their present state of technology.
9. What do you understand by CRT terminal? Discuss the working principle of a CRT display unit. What is raster scan and vector scan method of CRT display?
10. Discuss the principle how a character is displayed on a CRT screen. What is the function of character generator ROM? What is bit-mapped raster scan method of display?
11. What is the function of a CRT controller?
12. What are non-CRT displays? Give examples and discuss their operating principle.
13. What are the different types of printers? What is the difference between impact type and nonimpact type printers?
14. Discuss the working principle of a dot-matrix printer? Discuss its merits and demerits.
15. Distinguish among character printers, line printers and page printers.
16. Discuss the working principle of laser printers. Discuss the area of applications of slow speed and high speed laser printers.
17. What are the different types of plotters? Discuss their working principle. What are hard copy devices for fast plotting of drawings?
18. What are microfilm and microfiches? Discuss their working principle and area of applications.
19. What is the function of a port? What is an input port or an output port? What are programmable and nonprogrammable ports?
20. What is a programmable peripheral interface device? Describe Intel 8255. What are various operating modes of 8255?
21. What is serial data transfer? What is baud rate?
22. What is an USART or an UART? What is the function of a modem? What are DCE and DTE?
23. Discuss RS-232C and RS-422A standards for serial data transfer.
24. Describe Intel 8251. Where is it used?
25. What is programmable interval timer/counter? Discuss the major features of Intel 8254 and 8253.
26. What are the various schemes which are used for data transfer between the two devices of a computer? Discuss asynchronous data transfer with suitable examples.
27. What is DMA scheme of data transfer? Discuss its operating principle. What is burst mode of data transfer and cycle stealing method of data transfer?

28. What do you understand by interrupts? Discuss the interrupt driven data transfer scheme.
29. What is interrupt controller? Describe Intel 8259.
30. What is DMA controller? Describe Intel 8237A.
31. What is a microcontroller? Discuss its area of applications?
32. What is an I/O processor? What are the main features of UPI-452, and UPI-41 and 42?
33. What is the role of a math coprocessor in a computer? What operations are performed by such coprocessor? Which coprocessor is used with 8088, 80286 and 80386 CPU ?

**THIS PAGE
IS BLANK**

8

CHAPTER

SYSTEM SOFTWARE AND PROGRAMMING TECHNIQUES

A computer can only do what a programmer asks to do. To perform a particular task the programmer writes a sequence of instructions, called *program*. An *instruction* is a command given to the computer to perform certain specified operation on given data. A set of programs written for a computer is called *software*. The software needed to execute the user's program is known as *system software*. The system software consists of operating system, assembler, compiler, interpreter, debugging programs, text editors etc. A program which is prepared by a programmer to solve certain problem or to perform certain specified task is known as *user's program*. The *operating system* is a collection of programs that controls the overall operation of a computer. The term *firmware* is used for the software stored in read only storage devices. Those programs which are rarely or never expected to be changed are stored as firmware. Very important programs which must be in the primary memory and must not be lost when power goes off, are stored as firmware such as programs of a microcontroller etc. The microprograms of computers are stored as firmware. The term software includes both system software as well as user's program. The user's program is stored in RAM. The CPU executes all the instructions of the program one by one to obtain the desired result.

The softwares which are helpful to users in developing, writing, debugging and documenting programs are referred to as *utility programs*. These are software tools to help users while preparing programs. There are two types of utility programs: file management utilities and program development utilities. The file management utilities are helpful to users in creating, coping, printing, erasing and renaming the files. These programs are a part of the operating system. The program development utilities are useful in writing and testing programs. These are text editor, assembler, compiler, loader, linker, locator, debugger etc. A simple text editor forms a part of the operating system whereas an elaborate editor is stored separately. Compilers are stored separately. Programs loaded separately also run under the control of the operating system.

A large variety of prewritten softwares is available nowadays to solve specific tasks. Users need not prepare programs for many specific tasks. They should simply know how to use such prewritten softwares. Special purpose softwares are available for word processing, billing, inventory control, computer based engineering design such as design of electronic circuits, design of different aspects of power systems, building, structure etc. Such special-purpose programs (softwares) are called application programs or application packages. Some important application packages (programs) available are: ORACLE, MS-WORD, MS-Excel etc.

8.1 MACHINE LANGUAGE

A computer understands information composed of only zeros and ones and hence, it uses binary digits for its operation. The computer's instructions are coded and stored in the memory in the form of 0s and 1s. A program written in the form of 0s and 1s is called *machine language program*. There is a specific binary code for each instruction. For example, to add the contents of register A and register B, the binary code is 10000000 for Intel 8085. The binary code (machine code or object code) for a certain operation differs from microprocessor to Microprocessor. Each microprocessor has its own instruction set and corresponding machine codes.

8.2 ASSEMBLY LANGUAGE

The writing of programs in machine language is very difficult, tiresome and boring job for a programmer. Moreover, it is errorprone. To solve this problem and to facilitate programmer easily understandable languages have been developed. Assembly language is one of them. Programs can easily be written in alphanumeric symbols instead of 0s and 1s. Meaningful and easily rememberable symbols are selected for this purpose. For example, ADD for addition, SUB for subtraction, CMP for comparison, etc. Such symbols are known as *mnemonics*. A program written in mnemonics is called *assembly language program*. The writing of programs in assembly language is easier and faster as compared to the writing of programs in a machine language.

8.2.1 Assembler

When a program is written in a language other than the machine language of computer, the computer will not understand it. Hence, the programs written in other languages must be translated into the machine language of the computer before they are executed. Such translation is performed with the aid of software. A program which translates an assembly language program into a machine language program is called an *assembler*. An assembler which runs on a computer for which it produces object codes (machine codes) is called a self assembler (or resident assembler). A less powerful and cheaper computer may not have enough software and hardware facilities for program development and convenient assembly. In such a situation a faster and powerful computer can be used for program development. The programs so developed are to be run on smaller computers. For such program development a cross assembler is required. A *cross assembler* is an assembler that runs on a computer other than that for which it produces machine codes.

One-Pass Assembler. It is an assembler which reads the assembly language programs only once. Such assembler must be equipped with some means to assign addresses to the labels used in the assembly language program.

Two-Pass Assembler. It is an assembler which goes through the assembly language program twice. On the first pass the assembler reads the assembly language program and collects all labels. It assigns addresses to the labels counting their position from the starting address. On the second pass the assembler produces the machine code for each instruction and assigns address to each.

8.2.2 Advantages of Assembly Language

The advantage of assembly language over high-level languages is that the computation time for an assembly language program is less. An assembly language program runs faster to produce the desired result.

8.2.3 Disadvantages of Assembly Language

- (i) Programming is difficult and time consuming.
- (ii) The assembly language is machine oriented. The programmer must have the detailed knowledge of the structure of the computer he is using. He must have the knowledge of registers and instruction set of the computer, connections of ports to the peripherals etc.
- (iii) The program written in assembly language for one computer can not be used on any other computer, *i.e.*, the assembly language program is not portable. Each processor has its own instruction set and hence its own assembly language.
- (iv) An assembly language program contains more instructions compared to high-level language program. Each statement of a program in a high-level language (such as FORTRAN, PASCAL etc.) corresponds to many instructions in an assembly language program.

8.3 LOW-LEVEL AND HIGH-LEVEL LANGUAGES

A language in which each statement (instruction) is directly translated into a single machine code is known as *low-level language*. Examples of the low-level language are machine language and the assembly languages of various processors. The demerits of assembly language were already discussed. The basic problem of an assembly language is that it is more closely related to the structure of a computer than to the performance of the task which is to be performed. The programmer spends more time in manipulating registers and considering instruction sequences than solving actual problems. The programmer must have very detailed knowledge of the instruction set, architecture and connection of peripherals to the ports etc.

To overcome the difficulties associated with assembly languages *high-level or procedure-oriented* languages have been developed. High-level languages permit programmers to describe tasks in a form which is problem oriented or object oriented rather than computer oriented. One can formulate problems more efficiently in a high-level language and need not have a precise knowledge of the architecture of the computer he is using.

The instructions written in a high-level language are called *statement*. The statements more clearly resemble English and mathematics as compared to mnemonics in assembly languages. Examples of high-level languages are BASIC, PASCAL, FORTRAN, COBOL, ALGOL, C, C++, PROLOG, LISP, JAVA etc. A high-level language is independent of a computer. The same program will run on any other computer which has a compiler for that language. The compiler is machine dependent but not the language. The advantages of high-level languages are:

- (i) They are easier to learn as compared to assembly languages.
- (ii) They make programs easier and faster to write.
- (iii) Provide better documentation.
- (iv) The programmer does not have limitations to use a single type of machine.
- (v) Programs are portable, *i.e.*, they will run on any computer which has a similar compiler.

One statement of a high-level language corresponds to many instructions of the assembly language program. Hence, a high-level language program is much shorter compared to

assembly language program. Many high-level languages have been developed; some are for general purposes and some for special purposes. For example, PASCAL, C and C++ are general purpose languages. FORTRAN and ALGOL are for scientists and engineers. They are designed to solve mathematical problems. COBOL was used in the past for business applications. BASIC is for newcomers to programming. PROLOG is based on logical reasonings and used for artificial intelligence (i.e. for expert systems). SNOBOL is suitable for text processing. APT is used in manufacturing applications to control machine tools. Java is suitable for the Internet.

8.3.1 Source Language and Object Language

The language in which a programmer writes programs is called *source language*. It may be a high-level language or an assembly language. The language in which the computer works is called *object language* or machine language. Machine codes are also known as *object codes*. A program written in a source language is called a *source program*. When a source program is converted into machine code by an assembler or compiler, is known as an *object program*. In other words a machine language program ready for execution is called an object program.

8.3.2 Brief Description of High-Level Languages

BASIC. It is an abbreviation for Beginners All-purpose Symbolic Instruction Code. It is a very simple and easy language. It is suitable for scientific computations. But it is not as powerful as FORTRAN. It was introduced in 1965 by Dartmouth College. It is a widely used language for simple computation and analysis. It is now by far the most popular high-level language used in personal computers. To translate BASIC instructions into machine-language codes interpreters are frequently used in PC systems. But BASIC language compilers are also available for these systems.

The latest version of BASIC is QBASIC. It was developed by Microsoft. It replaces the earlier version of BASIC, which was called GWBASIC. QBASIC provides graphical user interface (GUI). It is provided with an intelligent editor which checks syntax as the program lines are entered. It is provided with an interpreter, which translates program line by line. In QBASIC it is not necessary to assign statement number to each and every statement as it is done in GWBASIC. Statement numbers are assigned where it is necessary.

Visual BASIC was introduced in 1991 by Microsoft. It is a powerful tool for developing Windows applications (GUIs) in BASIC. Earlier, Windows applications were developed in C language. The development of Windows applications was much harder task as compared to the development of DOS applications. Programmers had too much worry about, such as what the mouse was doing, where the user was inside a menu, and whether the programmer was clicking or double clicking at a given place. Development of Windows application needed expert C programmers and hundreds of lines of codes for the simplest task. Even the experts were facing problems.

Visual BASIC made the development of Windows applications an easy task. It is now used to develop Graphical User Interfaces (GUIs). It is a scripting language to combine small programs written in BASIC. It allows to add menu, text boxes, command buttons, options buttons (for making exclusive choices), check boxes (for non-exclusive choices), list boxes, scroll bars, and file and directory boxes. One can use grids to handle tabular data, communicate with other Windows applications and access databases. One can have multiple windows on the screen. Visual Basic has some features of OOP.

FORTRAN. It is an abbreviation for Formula Translation. It was introduced by IBM in 1957. It is a very useful language for scientific and engineering computations as it contains many functions for complex mathematical operations. It is a compact programming language. Huge libraries of engineering and scientific programs written in FORTRAN are available to users. It is not suitable for processing large business files as COBOL is. It has a number of versions. Earlier, FORTRAN IV was very popular. In 1977 the American National Standards Institute (ANSI) published a standard for FORTRAN called FORTRAN 77 with an idea that all manufacturers may use the same form of the language. FORTRAN 90 and FORTRAN 95 versions are also available.

COBOL. It is an abbreviation for Common Business-Oriented Language. It was developed specially for business data processing. It was introduced by U.S. industry/government committee in 1960. It was used for large business and commercial applications such as handling of ledgers, accounts, payroll files etc. It supports simple and limited numeric operations, but it can handle complex non-numeric operations. It is more suitable to manipulate alphanumeric characters than FORTRAN. It can be written in a quasi-english form that may employ commonly used business terms. Its English like statements can be understood very easily, for example, SUBTRACT WITHDRAWALS FROM OLD BALANCE GIVING NEW BALANCE. Its recent version is COBOL-2002. Visual COBOL has also been developed, which is an object-oriented version of COBOL.

Its demerit is that it is not a compact language. It is not easy to learn, and it can not handle complex mathematical computations as FORTRAN does.

C Language. It is a general purpose high-level language. This language has been designed by a group at Bell Telephone Laboratories, U.S.A. in the early 1970s. It permits manipulation of internal processor registers and hence, a programmer can write low-level machine instructions. It has the features of assembly language programming. It is a small and concise language. It makes use of a library of functions that are equivalent to subroutines. C program can use modular and structured concept. A problem may be divided into smaller tasks, a function may be used to solve each task. C program may represent a grouping of functions that are linked together to produce the solution of the problem. C programs can be shifted from one machine architecture to another without much difficulty. This language is used by system programmers to develop complex programs such as an operating system and application program. AT and T (Bell Lab's parent) produce C compiler and UNIX operating system tools as a single software package. It has many versions which run on PCs and larger machines.

C++. C++ is an extension to C language. Commands of C language are subset of C++. In other words C++ is a superset of C. Almost every correct statement in C is also a correct statement in C++, but the reverse is not true. It was developed in the year 1980. It is an object-oriented language. It is a high-level language. Other high-level languages such as BASIC, COBOL, PASCAL, FORTRAN, C etc. are procedure-oriented language. While writing a program in an object-oriented language, the programmer has to tell only 'what to do'. In procedure-oriented language the programmer has to tell 'what to do' as well as 'how to do'. Now-a-days object-oriented programming is getting wide acceptance. OOP stands for Object Oriented Programming. C++ has all features of both structured programming and object-oriented programming.

JAVA. It is an object-oriented programming language. It has been derived from C++. SUN Micro System Inc. has developed this language. This language is suitable for Internet,

desktop computers, servers, microcontrollers, etc. Internet software and application programs are being written in JAVA. Application programs for a network of heterogeneous computers can easily be written in JAVA. Application programs written in JAVA can run on any platform i.e. on any computer built around the CPU of any company. Here, question arises how Java achieves machine independence. It is achieved using a **Java Virtual Machine**. The Java Virtual Machine is not a real machine. It is a simulated computer which resides in a computer's memory. It performs all major functions of a real machine. Java programs are translated by Java compiler into machine codes, which are acceptable to Java Virtual Machine. These machine codes are called virtual machine codes. These codes are processed by the Java Virtual Machine. The output of the Java Virtual Machine goes to a Java interpreter which produces machine codes for the computer on which Java programs are running.

JAVA SCRIPT. It is a scripting language. It is written in Java language. It is used to develop application programs using applets. **Applets** are small Java application programs. A scripting language assumes that a collection of useful programs, each performing a task, is available. It can combine these programs to make a complex program. The new complex program can perform a complex task.

SQL. It stands for Structured Query Language. It is a language which enables users to create and operate on relational database which is the sets of related information stored in the form of tables. It is used in database packages such as ORACLE, SYBASE, INGRESS, INFORMIX, etc. It is also a standard language for accessing client/server databases. SQL is designed to handle a database. It caters to most of the routine queries put by the user on the database. It can extract data from a database in meaningful ways. It is used to create database tables, to define relationship between two or more databases, to modify tables, to access/retrieve a database, to manipulate data in a database, update data in a database, and so on.

DDL (Data Definition Language). It is a part of SQL. It consists of those commands which create objects such as tables, indexes, views etc., in the database. Examples of some commands of DDL are: Create, Drop and Alter.

DCL (Data Control Language) is another part of DDL. It is related to security mechanism or scheme for the protection of data in a system from unauthorised access. It provides features which can determine whether the user is allowed to perform certain action. Its commands are: Grant, Revoke etc.

DML (Data Manipulating Language). It provides features like retrieving data, manipulating data and updating data. Retrieving data allows to get information from the database. Manipulating data allows users to perform functions like summing and averaging of columns and sum other arithmetic functions. Updating allows to update data, insert data and delete data. Corresponding commands are: Update, Insert and Delete.

PROLOG. It is a suitable language for developing programs involving complex logical operations. It is used primarily for artificial intelligence. It was developed in France. The Japanese have chosen this language as a standard language for their fifth generation computer project. It is quite suitable for handling large databases and for producing rules-based expert systems applications. PROLOG stands for PROgramming in LOGic. It is based on mathematical logic. Most of today's-high-level languages like BASIC, COBOL, FORTRAN or PASCAL are not based on the principles of mathematical logic. These languages were designed to provide efficient computation and data manipulation. They enable us to use

computers for these purposes. But today computers are also being used to provide conclusions based on intelligent reasoning. For such a purpose programming languages based on the principle of mathematic logics are needed. PROLOG is based on the first order predicate calculus. PROLOG consists of a set of facts and rules that describe objects and relations between objects in a given domain. The statements that are unconditionally true are called facts, while rules provide properties and relations which are true depending on given conditions. Many expert systems have been developed. They perform operations based on logical reasoning and provide conclusions.

LISP. It stands for LISt Processing. This language was developed by McCarthy in the early 1960s. It is suitable for nonnumeric operations involving logical operations. It is used extensively in artificial intelligence and pattern recognition. It is also used in game playing, theorem proving etc. It is capable of searching, handling and sorting long strings or lists of text. So it has often been used to implement computerized translators. It is used primarily on larger computers but LISP compilers are also available for PCs.

SNOBOL. It stands for StriNg Oriented SymBOLic Language. This language was developed by a group led by Griswold in the mid 1960s. It can manipulate strings of characters and hence it is used in text processing. It is capable of performing various types of operations on strings of characters such as combining strings, splitting strings, matching strings etc.

LOGO. It was developed by Seymour Papert and his colleagues at MIT in the late 1960s. It is used in serious scientific work in universities. It has also been popularized as a first educational language that children can use to achieve intellectual growth and problem-solving skills. LOGO has graphics capability. Children can easily use it to make drawings. They can draw, colour and animate images. It runs on PCs. It is used to compose music, manipulate text, manage data etc.

APT. It stands for Automatically Programmed Tooling. It is used in manufacturing applications to control machine tools.

8.3.3 Programming Language Generations

First Generation Language, 1GL. Machine language is known as the first-generation language.

Second Generation Language, 2GL. An assembly language is the second-generation language. An assembly language is machine oriented. It is very close to the hardware of the CPU (i.e. microprocessor). An assembly language programmer must know the architecture of the microprocessor for which he is writing assembly language programs. He must have the knowledge of microprocessor's registers and its instruction set.

Third Generation Language, 3GL. High-level languages are the third-generation-languages. The third-generation languages are procedure-oriented languages. The programmer must tell 'what-to-do' as well as 'how-to-do'. He has to write the steps how to perform the desired task. Examples of 3GLs are : BASIC, FORTRAN, PASCAL, COBOL, C etc.

Fourth Generation Language, 4GL. Non-procedural or object-oriented languages are the fourth generation languages. In a 4GL the programmer has to tell only 'what-to-do'. 4GLs are considered to work at higher-level than normal high-level languages. They are usually application programs for RDBMS (Relational Database Management Systems) and some new languages. Most of 4GLs use SQL as a tool. Examples of 4GLs are : ORACLE, SYBASE, etc. JAVA and C++ are fourth-generation languages. These are suitable for writing object-oriented programs.

8.3.4 Compiler

A program which translates a high-level language program into a machine language program is called a compiler. A compiler is more intelligent than an assembler. It checks all kinds of limits, ranges, errors etc. But its program execution time is more, and occupies a larger part of the memory. It has low speed and low efficiency of memory utilization. If a compiler runs on a computer for which it produces the object code, then it is known as a *self* or *resident compiler*. If a compiler runs on a computer other than that for which it produces object code, then it is called a *cross-compiler*.

8.3.5 Interpreter

An interpreter is a program which translates statements of a high-level language program into machine codes. It translates one statement of the program at a time. It reads one statement of a high-level language program, translates it into machine code and executes it. Then it reads the next statement of the program, translates it and executes it. In this way it proceeds further till all the statements of the program are translated and executed. On the other hand a compiler goes through the entire high-level language program once or twice and then translates the entire program into machine codes. A compiler is 5 to 25 times faster than an interpreter. An interpreter is a smaller program as compared to compiler. It occupies less memory space. It can be used in a smaller system which has limited memory space. The object program produced by the compiler is permanently saved for future reference. On the other hand the object code of the statement produced by an interpreter is not saved. If an instruction is used next time, it must be interpreted once again and translated into machine code. For example, during the repetitive processing of the steps in a loop, each instruction in the loop must be reinterpreted every time the loop is executed.

8.4 STACK

During program execution the contents of certain registers are needed to be saved because the registers are required for some other operations in the subsequent steps and the saved contents will be needed at the later stage of the program execution. The contents of registers are saved in certain memory locations set aside by the programmer for this purpose in the very beginning while writing the program. The memory space set aside for this purpose is called *stack*. Thereafter the registers are used for other operations. After completion of these operations the saved contents are brought back from the memory to the registers. The last occupied memory location of the stack is known as *stacktop*. A special register known as stack pointer (SP) holds the address of the stacktop to keep track of the stack memory locations. Any area of the RAM can be used as stack, but usually the last user's area of the RAM is set aside for this purpose.

The PUSH instruction is used to transfer the contents of the registers to the stack, and the POP instruction to bring back the contents from the stack to the registers. The data are stored in the stack on last-in-first-out (LIFO) principle. Fig. 8.1 (a) shows the stack position before PUSH operation. Suppose the contents of B-C register-pair are to be saved. Fig. 8.1 (b) shows the stack position after PUSH operation. The stack positions before and after the POP operation are shown in Fig. 8.2 (a) and 8.2 (b) respectively. Stack access is faster than memory access.

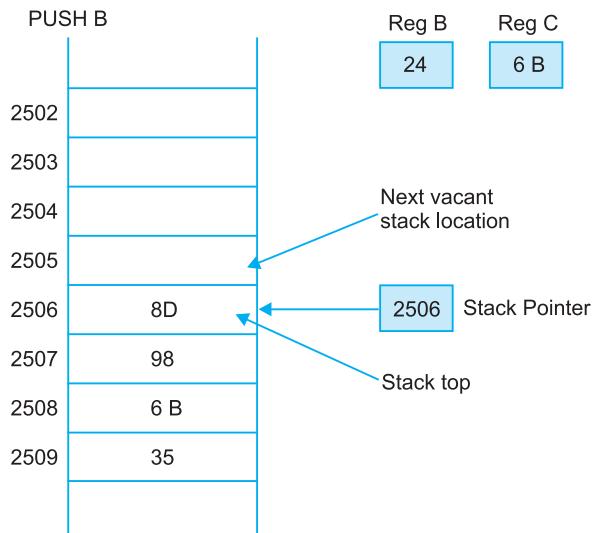


Fig. 8.1 (a) Stack position before PUSH operation.

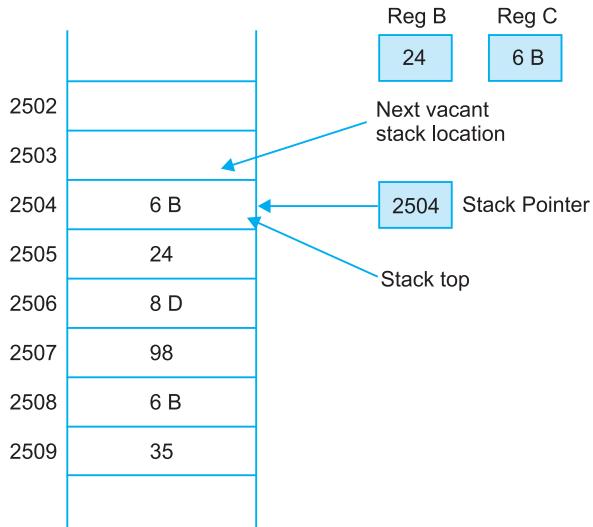


Fig. 8.1 (b) Stack position after PUSH operation.

8.5 SUBROUTINES

Often, we need to execute several times a sequence of instructions to perform a subtask within a program. In such cases, we write the sequence of instructions which perform subtask as a separate subprogram. This subprogram can be called at any point in the main program whenever required using a special instruction (CALL) for this purpose. Such subprograms are known as *subroutines or procedures*. On the completion of the subroutine, the execution of the main program begins from the next instruction after the CALL instruction in the main

program. Before starting the execution of a subroutine the content of the program counter, PC is saved on the stack. The contents of PC point to the instruction following the CALL instruction (*i.e.*, instruction next to the CALL instruction in the main program). At the end of a subroutine RET (return) instruction is used. It restores the contents of PC so that the control is transferred back to the main program. Fig. 8.3 shows the main program, subroutine and CALL-RET structure.

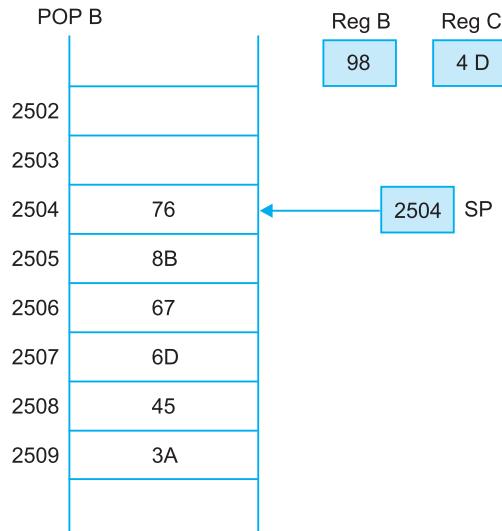


Fig. 8.2 (a) Stack position before POP operation.

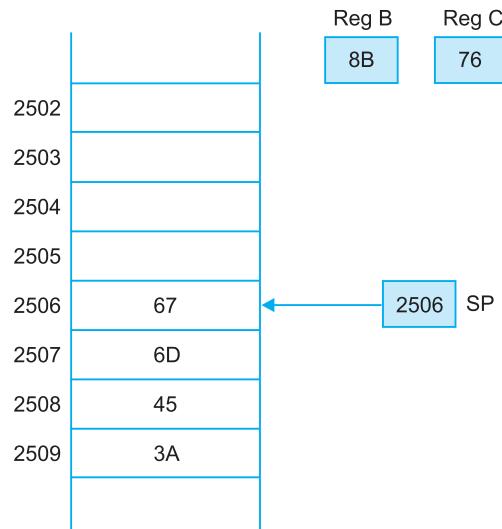


Fig. 8.2 (b) Stack position after POP operation.

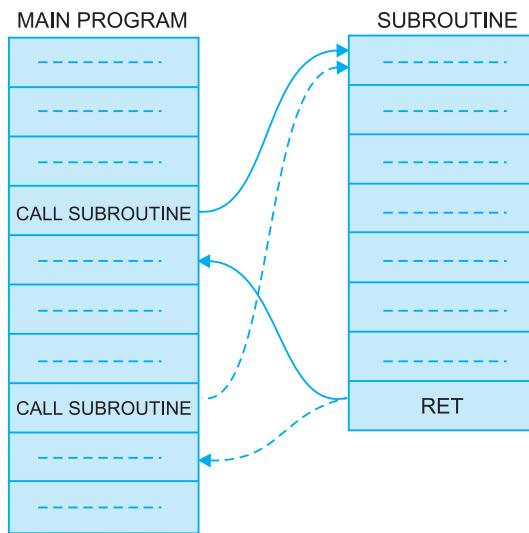


Fig. 8.3 Subroutine CALL operation

8.6 DEBUGGING OF PROGRAMS

In computer's terminology the word bugs means errors. It is used in the context of errors in programs. To debug means to eliminate program errors. *Debugger* is a program that helps in finding and correcting errors in user's programs. The program debugging and testing are also called *verification* and *validation* respectively. When programs are prepared, usually they contain some errors. They do not run correctly the first time. Compilers and interpreters include error-diagnostic features. Such features detect errors which are due to the incorrect application of the language used to prepare the source program. Such errors are called *syntax errors*. The computer shows the errors on the screen and also indicates the erroneous statements.

A simple method of checking a program is single step control. By this technique program is executed step by step and intermediate results are observed. After executing one instruction, the programmer can examine registers, memory etc. to see the result after the execution of that instruction. At a time only one machine instruction is executed. The program checking by single step method becomes very difficult and time-consuming in case of a long program particularly when the error is towards the end of the program. Furthermore, if a loop occurs in a program, it becomes a tedious job to cross the loop by single step control. For such situations there is another technique called *break point* technique which can be conveniently used for program checking. In this technique a break point using a software interrupt can be introduced at the suitable point in the program. To introduce software interrupt an instruction meant for software interrupt is used such as RST n (RST 5, RST 6 etc.) in case of Intel 8085 and instruction INT or INTO in case of Intel 8086. When a break point is inserted in a program, the program is executed only upto the break point. So by inserting break points at appropriate points the program can be checked and ascertained which part of the program contains errors. The part which contains errors can be executed by single control method to find out which instruction contains errors.

A debugger allows to execute a single statement or a single section of a program at a time. After the execution of a statement or a section of the program, the values of the variables, the contents of registers or the code that has just been executed or is about to be executed can be displayed. It also allows to insert break points.

Programs need thorough debugging and testing before they are used. There are a number of tools to debug programs. Some important tools are:

- (i) Simulators
- (ii) Logic analyzers
- (iii) Break points
- (iv) Trace routines
- (v) Memory dumps etc.

Simulators. It is a computer program which simulates the execution of a program on another computer. Simulators are large programs and run on a large computer. These programs are useful in checking the logic of user's program, as the user can change data, examine registers and use other debugging facilities. But they do not fully model input/output or give much information about timing problems.

Logic Analyzer. It is testing instrument. It is the digital bus-oriented version of the oscilloscope. It can detect the states of digital signals during each clock cycle and store them in a memory. The stored information can be displayed on the screen. Like oscilloscope it has the features of monitoring and displaying many inputs, specifying triggering of events and setting of threshold. These are convenient means to display rapidly changing parallel digital signals. Some logic analyzers are capable of triggering on a particular instruction or a sequence of instructions, recalling previous data and capturing very short noise spikes (or glitches). Logic analyzers are complementary to software simulators, since they are mainly used in solving timing problems.

Trace Routines. It is a program which can print information regarding the status of the processor at specified intervals. Some microcomputers and most of simulator programs are provided with trace facilities. The programmer has to select variables and formats carefully so that he should be able to examine their traces.

Memory Dumps. The listing of the current contents of a section of the memory is called memory dump. It is not an effective means for debugging. But it is used when other methods fail. Most of simulator programs, MDS and monitors are capable of producing memory dumps.

8.7 MACRO

If a sequence of instructions is used frequently in a program, it will be convenient for the programmer to assign a name to the sequence. Some assemblers have facility to assign a name to a sequence of instructions. The sequence of instructions to which a name is assigned is called a *macro*. Once a sequence of instructions is written and a macro name is assigned to it, the assigned macro name will be used in the program whenever needed instead of writing the sequence over and over again. The assembler will replace the macro name by the corresponding sequence of instructions. It makes the task of the programmer easier. Also, the program becomes shorter and easier to read and understand. Macros and subroutines are similar. Macros are used for short sequences of instructions whereas subroutines for longer

ones, preferably for 10 or more instructions. Subroutines need CALL and RET instructions while macros do not. Each macro is clearly defined and unique name is assigned to it. A few examples are as given below.

Example 1

COMP2	MACRO	ADDRESS
	LDA	ADDRESS
	CMA	
	ADI	01
	ENDM	

In this example COMP2 is the name of the macro. The word MACRO is to be written in the beginning of the definition as shown in Example 1. ADDRESS is a parameter. ENDM is used to end a macro. If we write COMP2 1601 in an assembly language program, the assembler will replace this macro by the following sequence of instructions in the program.

LDA	1601
CMA	
ADI	01

This sequence of instruction will take the 2's complement of the contents of the memory location 1601.

Example 2

SHIFT2	MACRO
	ADD A
	ADD A
	ENDM

SHIFT2 is the name of the macro. If SHIFT2 is written in the program, the assembler will replace it by

ADD A
ADD A

This series of instructions will shift the content of the accumulator left by 2 bits. The first ADD A instruction adds the content of the accumulator to itself; which will result in the shifting of the content of the accumulator left by one bit. The 2nd ADD A instruction will shift the accumulator's content once again, so finally the accumulator's content is shifted left by 2 bits.

8.8 MACRO PROCESSOR (OR KEYBOARD ENHANCER)

It is a memory-resident package (or desktop organizer or desk accessory) which permits the use of macros. It is a program that allows the user to write and store a string of characters, and then assign a name (or abbreviation) to it. A string of characters is a combination of letters, digits, and/or other symbols. It may be a word, a part of a word, a group of words, a number and so forth. When the programmer types that macro or abbreviation using computer's keyboard, the macro processor substitutes the stored string of characters and sends it to the primary program. Most macro processors can store about 300

abbreviations at any time, and each macro can represent thousands of characters if required. Thus by pressing a few keys a set of thousands of characters can be brought into the primary program which the user is executing.

Besides macro processor, memory resident package for some other functions are also used. For example, a spelling checker, appointment calender and alarm clock, notepad, calculator etc. A memory-resident function can be called up by the programmer with a few keystrokes.

8.9 PROGRAM DESIGN

The important techniques which are useful in designing programs are as follows:

- (i) Modular programming.
- (ii) Structured programming.
- (iii) Top-down and bottom-up design.
- (iv) Object Oriented Programming (OOP).

8.9.1 Modular Programming

When a program becomes very long and complex, it becomes a very difficult task for the programmer to design, test and debug such a program. Therefore, a long program can be divided into smaller programs called *modules*. As the modules can be designed, tested and debugged separately, the task of the programmer becomes easy and convenient. The division of a long program into smaller programs (or modules) is called *modular programming*. Advantages of modular programming are as follows:

- (i) It is easier to design, test and debug a single module as compared to an entire program.
- (ii) Usually, a module of general nature is prepared so that it can be used elsewhere.
- (iii) If any changes are to be made, they are made into particular module where needed rather than into the entire program.
- (iv) This technique allows the programmer to divide tasks into subtasks and use previously written programs.

Disadvantages of the modular programming are:

- (i) Since separate modules may repeat certain functions, the modular programming often need extra time and memory.
- (ii) The combining of the modules may be a problem particularly when different persons design and develop different modules.
- (iii) In modular programming the documentation needs special attention because a module may effect the other part of the program.
- (iv) While testing a module there arises a problem that it may require data from other modules. This problem can be solved by using special programs called *drivers*. The drivers produce sample data for testing. The development of drivers needs extra effort and time.

Though the modular programming technique has been developed for designing long programs, this technique can also be used for comparatively shorter programs for

microprocessors. The modules are most often divided on the basis of functional lines. This type of division is particularly useful in microprocessor programming. The modules can be kept in a library of programs to be used for later work. The modules of 20 to 50 lines are better because longer modules are seldom of general nature and may be difficult to integrate. Preparation of shorter modules is waste of time. Modules should be prepared for common tasks. They should be as distinct and logically separate as possible.

8.9.2 Structured Programming

With the increasing capacity of the memory, the programs also became longer and longer. The long and complex programs may be well understood by the programmers who developed but not by the persons who had to maintain them. To overcome this difficulty a technique known as *structured programming* was developed to write a program. The basic idea behind this technique is that any part of the program can be represented by elements from three basic logic structures. Each structure has single entry and single exit. The three basic logic structures are:

- (i) **Simple Sequence Structure.** It is a linear structure in which instructions or statements are executed consecutively in a sequence.
- (ii) **Conditional Structure.** In this structure a condition is tested. The condition is followed by two alternative program control paths. The selection of the path depends on the result of the test. If the condition is satisfied a particular program (PROGRAM 1) is executed. If the given condition is not satisfied the other program (PROGRAM 2) is executed. Fig. 8.4 shows a flowchart for conditional structure. It is also called IF-THEN-ELSE structure.

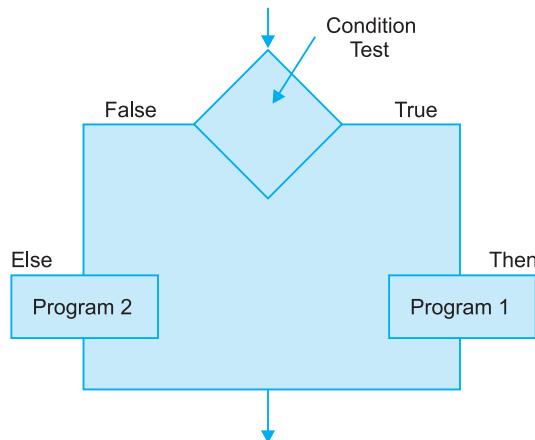


Fig. 8.4 Conditional structure

- (iii) **Loop Structure.** Fig. 8.5 (a) shows a loop structure in which the computer checks the condition. If the given condition C is true, the given program P is executed. The program P is not executed when C is false. This is a DO WHILE structure. In this structure the program is executed once or more while the condition is true. When the condition becomes false, the looping process ends. When C is false in the very beginning, P is not at all executed.

Figure 8.5 (b) shows a variation of the loop structure. It is known as DO UNTIL structure. In this structure the condition is tested at the end whereas in DO WHILE structure

the condition is tested at the beginning. In DO UNTIL structure the looping process is repeated until a condition becomes true.

The structured programming technique is more useful when program length exceeds 1000 lines. The development costs for larger programmes is less but this technique needs more memory space. It is not suitable for applications which involve complex data structure. In this method it is very simple thing to trace the sequence of operations. This permits easy testing and debugging. Structured programs can easily be described with the help of flowchart or other graphic methods.

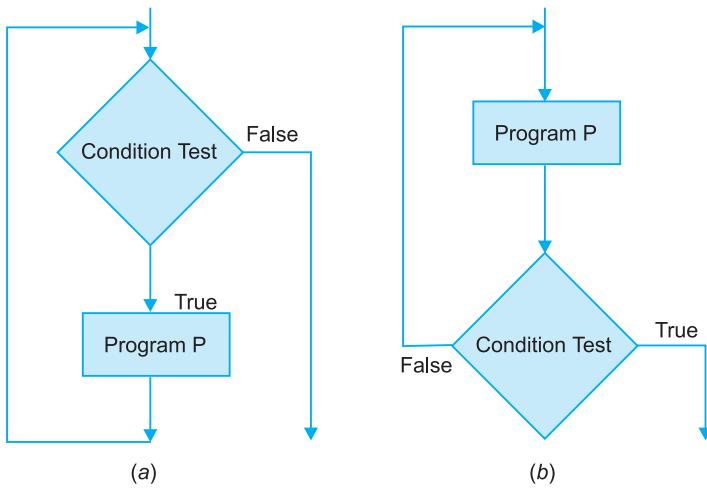


Fig. 8.5 Loop structures **(a)** DO WHILE loop structure, **(b)** DO UNTIL, loop structure

8.9.3 Top-Down and Bottom-Up Design

In top-down technique the design of the system program is started at the system level. The programmer first develops the overall supervisor program which is used to outline and control subprograms. The whole system work is divided into a number of subtasks. To perform each subtask there is a subprogram. The supervisor program (or main-control program) is then tested to see that its logic is correct. The undefined subprograms are replaced by temporary programs called stubs (dummy subprograms). A stub represents a subtask. A stub may either record the entry of a subprogram or to a selected test program. Then the programmer proceeds by expanding stubs. Testing and debugging is made at each step as a stub is replaced by a working program. The process of expansion, testing and debugging is continued until all the stubs are replaced by working programs. In this technique testing and integration is made along the way at each level, rather than all at the end. No special drivers or data generating softwares are necessary. Therefore, the testing is done in the actual system environment rather than using driver programs. This method gives an idea of exactly where the programmer is in the design. The top-down design technique assumes modular programming, and is compatible with structured programming as well. The disadvantage of top-down design is that the overall system design may not take good advantage of the hardware.

The bottom-up design is a traditional design technique. In this technique inner subprograms are prepared first for specific tasks and then integrated into a complete system. This technique should be used for frequently used subprograms whose speed is critical to the speed of the whole problem and whose functions are clearly understood initially. This ensures that correct parameters are selected for the subprograms. The top-down technique is better if the precise nature of the subprogram cannot be determined until a much more detailed analysis is completed. It permits easier debugging of the main program. In bottom-up design extra work is required for testing and debugging. The entire integration work is to be done at the end.

8.9.4 Object-Oriented Programming (OOP)

In object-oriented programming the programmer has to tell only 'what to do'. On the other hand in procedure-oriented programming the programmer has to tell 'what to do' as well as 'how to do'.

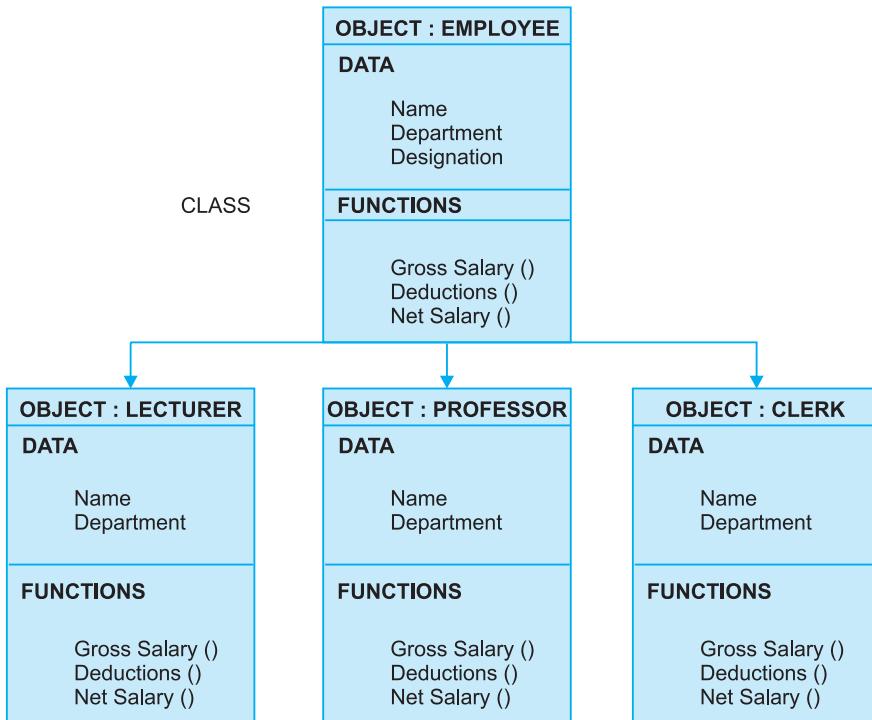
The main problem with computer programs is complexity. Large programs are very much complicated. Because of complexity, programs are prone to errors. Software errors in a large program may create several problems. Object-oriented programming offers a new and powerful way to cope with complexity. Its goal is clearer and more reliable. It results in the availability of more easily maintained programs. A few terms in connection with OOP are explained below :

Object: An object consists of data and functions. It is a self-contained unit of data and functions. An object-oriented language relates data to its functions. An object may be an item, person or any other entity. The data and functions related to an object are given within the object.

Take examples: Two objects—Lecturer and Professor contain data and function as shown below:

OBJECT : LECTURER	OBJECT : PROFESSOR
DATA	DATA
FUNCTIONS	FUNCTIONS
Name Department	Name Department
Gross Salary () Deductions () Net Salary ()	Gross Salary () Deductions () Net Salary ()

Class: The two examples given above—Lecturer and Professor are similar. Both are employees of an institution. Both objects have the same set of data and functions, and hence, they can be placed in the same class (or category). This class can be named—"Employee". The class employee includes clerks, librarians, demonstrators, peons, darwans, etc. This class can be shown as :



In OOP objects are the members of a class. In the above example, it is to be noted that data and functions to be used in an object are specified in the class. Hence, the class must be defined before the creation of objects of that class. We can give some more examples of classes : The class 'Fruit' includes orange, apple, guava, mango, banana, etc. The class 'Grain' includes rice, wheat, maize, etc.

Inheritance: In OOP a class can be divided into **subclasses**. For example, the class of vehicles can be divided into subclasses like cars, buses, trucks, scooters and motor cycles. The original class is called the **base class**. Subclasses are **derived classes**. In the given example vehicle is the base class and cars, buses, etc. are derived classes. Derived classes inherit some characteristics from their base class, but add some new characteristics of their own. All members of vehicles have wheels and engine. These are common characteristics. In addition to these common characteristics the members of different subclasses have some own characteristics, for example, a bus has seats for many people, while a truck has space for hauling heavy loads. The idea of classes and subclasses leads to the idea of inheritance.

8.10 SOFTWARE DEVELOPMENT

While preparing programs the following factors should get due considerations:

Reliability. A program must work reliably. It should perform the task properly for which it has been developed.

Speed. A program must execute the specified task quickly. The time taken by a program to perform a given task should be as minimum as possible. A faster program will give more output in a given time than a slower one.

Programming Time and Cost. The cost of processors, memory and peripherals are decreasing but the cost of programming is rising. Due to this reason more attention is being

given on programming techniques like structured programming and top-down design which increase programmer's output. Proper design, testing, debugging and documentation reduce the overall cost of programming.

Ease of Use. A program must be easily understood by others. A program with strictly defined and complicated data formats is difficult to use and expensive to debug and maintain.

Error Tolerance. A program must react to errors. It should give some information regarding errors or malfunctions without shutting the entire system down.

Extendibility. A program that can be extended to tasks other than for which it has been designed and developed is definitely a better program. The modular programming is more useful in attributing the feature of extendibility to a program.

8.10.1 Stages of Software Development

The software development may be divided into the following stages:

Problem Definition. At this stage the problem to be solved or the task to be performed is defined. Inputs, outputs, processing requirements, system constraints such as execution time, accuracy etc., and error handling methods are specified.

Program Design. At this stage the program is designed to meet the specified requirements according to its definition. The important design techniques are top-down, structured programming, modular programming and flowcharting. The flowcharts help in explaining programs and describing program structure. Flowcharts are more helpful in documentation than in design. The program design has already been discussed in detail.

Preparation of Actual Program. At this stage computer instructions are written according to its design.

Testing. At this stage the program is tested to check whether it performs the required task or solves the given problem. This stage is also called *validation*.

Debugging. At this stage errors in the program are detected and corrected. This is also called *verification*. This has already been discussed in detail. Important debugging tools are simulators, logic analyzers, breakpoints, trace routines, memory dumps, etc.

Documentation. It indicates what functions are performed by the program and how these functions are carried out. It helps users to understand and maintain the program. Important techniques of documentation are flowcharts, comments, program listings etc.

Maintenance. At this stage programs are corrected and updated to meet the need of changing conditions. It should be corrected or modified on the basis of field experience.

Extension and Redesign. A program can be extended to other tasks. If necessary, it can be redesigned to get its improved version or to perform other tasks.

8.10.2 Testing of a Program

The purpose of testing a program is to determine whether it runs and produces correct results. First of all testing is done with the inputs which will produce known results. The testing is also done with unusual but valid data, which will check the program whether it is capable of handling exceptions. The program is also tested with incorrect, incomplete or inappropriate data which checks the error-handling capability of the program. Today many specialized testing packages are available for testing programs. When a program passes the tests, it is released for use. However, errors may still remain and it may give wrong results even after running for some time. It is due to the fact that a simple program that takes in

16 bits of data and produces 16-bit results involves 4 billion possible combinations of inputs and outputs. In a complex program there may be tens of thousands of different possible paths through the program. It is not practical and may be not even possible to trace through all these paths during testing.

8.11 FLOW CHARTS

As a computer cannot think on its own, the programmer has to provide a method to solve a problem. After defining the problem, the programmer writes the procedure how to solve the problem. The procedure must be written in the form of a series of steps in a logical sequence. A precise statement of the procedure required for solving a problem is called an *algorithm*. Having obtained an algorithm for solving a problem, the algorithm is expressed in a pictorial form called a *flow chart*. A flow chart is essentially a diagram or picture which defines the procedure how to solve the problem. A flow chart shows the order of operations. It also shows the relationship between the sections of the program. Flow charts are independent of a particular computer or computer language.

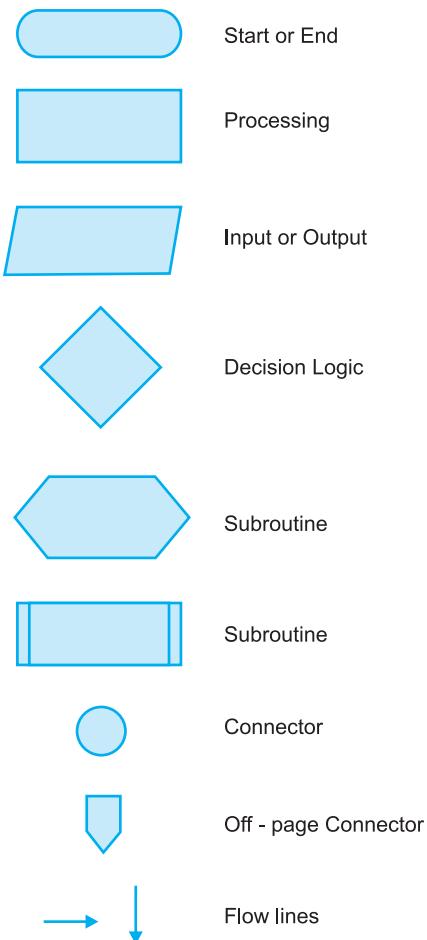


Fig. 8.6 Standard symbols for flow charts.

Fig. 8.6 shows the standard symbols which are usually used in drawing flowcharts. There are also some special symbols as shown in Fig. 8.7. In literature sometimes a circle is used to represent the start or end of a program.

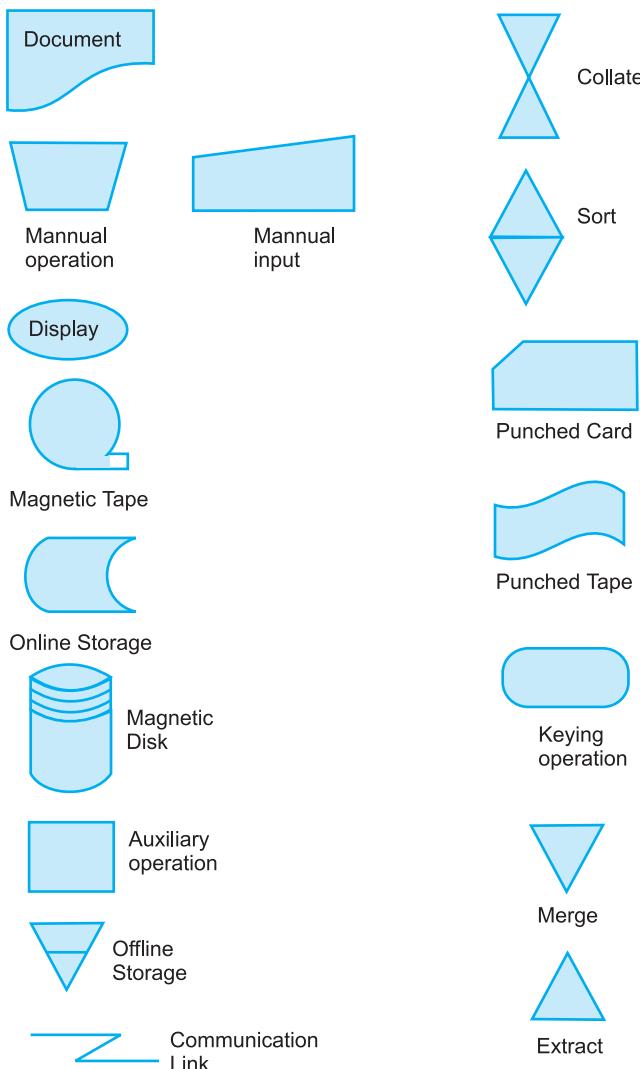


Fig. 8.7 Special symbols for flow charts.

In a complex system there may be an overall flow chart called *macro flow chart* which shows the general input/processing/output components of the system. There may then be a hierarchy of more detailed flow charts called *micro flow charts*, each of which describes a module in a higher level flow chart.

Figure 8.8 shows a flow chart to find the smallest number from a data array.

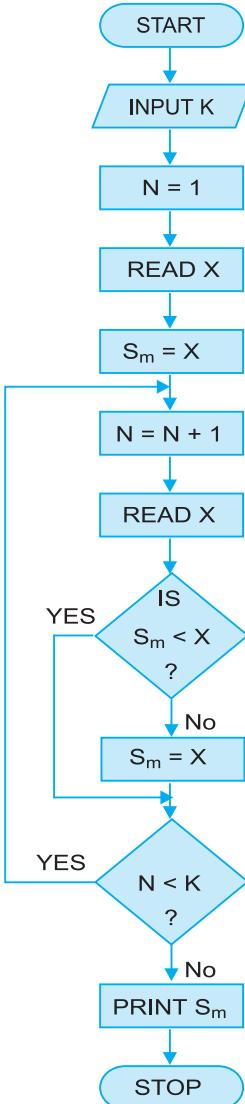


Fig. 8.8 Flow chart to find smallest number from a data array.

8.12 MULTIPROGRAMMING OR CONCURRENT PROGRAMMING

Usually, a CPU is much faster compared to peripheral devices. While peripherals are performing certain operations the CPU may not be doing any work. It may be lying idle. To keep CPU busy for most of the time it is desirable to process a number of programs concurrently. When a computer processes several programs concurrently, such processing or execution of programs is called *multiprogramming*. Nowadays, with the advent of VLSI and ULSI technology very powerful and fast CPUs are available. They are very much suitable for multiprogramming.

In multiprogramming several programs are stored in the disk memory. The operating system transfers a few programs from the disk memory to the main memory. The CPU begins the execution of one program and goes ahead until it reaches a point where it requires the operation of a slow I/O device such as a printer or it needs certain data which are not in the main memory. Suppose it needs such data. Then the CPU suspends the execution of the first program and begins the execution of another program. When the required data for the first program becomes available in the main memory the CPU is interrupted. Now the CPU resumes the first program that was suspended earlier, and suspends the second program.

Take another example, suppose the CPU needs a printer after executing the first program upto certain point. If printer is available, it prints out the result of the first program. If printer is not available, it stores data (which are to be printed) on a magnetic disk. Then it begins executing the second program. When printer becomes available, it takes necessary steps to print the result of the first program and thereafter return to the second program.

Thus we see that the CPU need not wait when a slow peripheral device is making certain operation or preparing data for the CPU. In multiprogramming system the CPU is kept busy most of the time. This improves the efficiency of the computer system. In a multiprogramming system the user cannot interact with the computer.

8.13 MULTIUSER OR TIME-SHARE SYSTEM

In a multiuser (or time-share) system several users work on the computer simultaneously. Several terminals are connected to a single CPU and operate on a time-share basis. The CPU gives a few milliseconds time (usually 20 ms) to each user and serves them in turn. As the computer operates at very high speed, each user feels that he is receiving the full attention of the computer. In a multiuser system the user can interact with the computer. The disadvantage of a multiuser system is that the computer response time to each user becomes unreasonably long when the number of the users becomes more. Another disadvantage of this system is that when the computer goes down all work carried out at different terminals is stopped.

When a number of programs or tasks are to be executed by a single CPU, they are to be scheduled in some way so that they can be executed properly. A user's program or a section of the program is called a *process* or *task*. The *scheduling* is a term which decides the order in which certain things will be performed or operated. There may be several methods of scheduling. In multiuser system several users get time on the basis of time-slice method. This is one type of scheduling. The operating system UNIX uses this scheduling approach for a multiuser system.

8.14 MULTITASKING SYSTEM

In industries a number of processes are controlled by a computer simultaneously. This is known as *multitasking*. The term multitasking is more often used in real-time control in industry. The operation of machine control or process control in an industry is a slower phenomenon as compared to the speed of a computer. Hence, a computer can check, measure and control a number of processes in turn. For example, it can measure and control temperature of a number of furnaces, speeds of motors, steam pressure at many points etc. before it is needed to return to the first process to recheck it. It cycles through all of them over and over again. It provides its services to all in turn in certain predetermined sequence. This

type of operation is called *multitasking* in real-time operation. In this system priority based scheduling is used. A low priority task can be interrupted by a high priority task.

The term multitasking is also used at places other than real-time applications, i.e. in general sense. In those cases it means multiprogramming.

The time-shared approach of multiuser system is not suitable for multitasking system. In a multiuser system all users are attended to by the computer at approximately equal time intervals. In multitasking system for industrial control, priority-based job scheduling is used. A high priority task can interrupt a low priority task. UNIX is not suitable for real-time industrial control which requires job scheduling on priority basis. UNIX is suitable for multiuser system which works on time-slice basis. Several companies have developed operating systems for industrial multitasking real-time operation. Intel RMX-86 is one example of this type of operating system. It operates on the principle of priority-based scheduling. It has been developed for industrial multitasking real-time control applications. It contains highly structured functional modules and utilities. The system designer selects the required modules and utilities for the industrial control he is going to design.

8.15 PROTECTION

In a multiuser system users should not interfere with each other. User should also not interfere with the operating system. To prevent such an interference a user should not use the memory area of the other user or the memory area of the operating system. There is hardware mechanism to prevent unauthorized access of memory by different users. The hardware of Intel's 16-bit and 32-bit processors includes mechanism that permits up to four levels of protection which can be used by the operating system running on it. The four levels are kernel, system services, operating system extensions and applications. In a multiuser system a password (a code) is given to each user. When the user enters his password into the computer then only he is allowed to work on it. Thus unauthorized access to computer is prevented. Each user is assigned a section of usable memory area and he is not allowed to go out of his assigned memory area. In chapter 6, Sec. 6.13 it was explained under the heading "protection" how hardware mechanism protects operating system from users. Also, the next section (i.e., Sec. 8.16) explains what steps are taken to prevent some typical interferences which arise between users while accessing common resources like printers, hard disks etc.

8.16 PROBLEMS IN ACCESSING RESOURCES

In a multiuser system common resources such as printers, magnetic disks etc. are used by several users. Conflicts may arise in using common resources. Take an example of the use of a printer. Suppose a user wants to print out his results. Obviously, his printing work cannot be completed in one of the 20 ms time slices. When he starts printing the results his turn will be over before his printing work is completed. Now the turn of the next user comes. Suppose he also wants to print something. His printing work also cannot be completed in 20 ms time. So his turn also will be over before his printing is complete. Thus the printouts of both users will be mixed up. To overcome this difficulty a software technique called *semaphore* is used. It is a flag. When a user is using a common resource such as a printer the semaphore is set to 1. When it is set to 1 no other user can use that common resource. When the work is completed the semaphore is set to zero.

Take another example when two users need the same two resources, for example, a printer and a magnetic disk. Both want to read some information from the disk and take a print out. Suppose that one user accesses the disk and sets its semaphore for the disk. His turn is over before information is read from the disk. Now the turn comes to the second user. He finds that the disk is engaged so he puts his request for the disk in the queue. Next, the second user tries for the printer which is free. He sets the printer semaphore. When the turn of the first user comes again he tries to get access to the printer but he finds that the printer is busy. He puts his request in the queue for the printer. In such a situation neither user can proceed. This situation is called *deadlock*. To overcome this type of difficulty a priority among the tasks may be assigned. Hence, when a deadlock occurs the task of the higher priority uses both resources first. Another approach may be to put the magnetic disk and the printer under one semaphore.

Take one more example when two users want to read and change the contents of some memory locations at the same time. Suppose that a computer system using time-slice approach is being used for reservation of seats in railways. Now, further suppose that the clerk on one reservation counter examines the memory location which represents a seat, and finds the seat vacant, just before the end of his time slice. A clerk from another reservation counter examines the same memory location in his time slice. He finds it vacant and marks it full. He issues a reservation ticket. When the first clerk gets the turn again, he marks the seat full because he had already checked the seat during his previous time-slice. He also issues a reservation ticket. Thus two persons get tickets for the same seat. To overcome this difficulty protection has to be provided. This is based on the principle that when a value of variable is being examined and changed by one user, it must be protected from access by an other user until the operation is complete. The section of code to be protected is referred to as *critical region*. *Mutual exclusion* principle is employed to prevent two users from accessing a critical region at the same time. For details one can refer to Chapter 15 of the book in Ref. 2.

8.17 OPERATING SYSTEM

An operating system is a collection of programs that controls the overall operation of a computer. It allows users to format disks; create, print, copy, delete and display files, read data from files, write data to files, control most input/output operations, execute programs, allocate memory locations, process interrupts etc. It provides users an interface to computing resources. It processes user's commands. In a multiuser system it allows several users to share CPU on the time-share basis, to share other system resources, it performs job scheduling, prevents interference between different users, between users and operating system; provides memory management etc. In short we can say that an operating system monitors the execution of user programs and the use of resources. The physical resources available in a computer are CPU, memory and I/O devices. In a multiuser system a resource is used by a number of programs and hence, the operating system has to maintain scheduling. The scheduling will decide the order in which a resource will be allocated if several programs want to use it.

Usually, operating systems are large. Most of them are too large to be stored in the memory at a time. Therefore, they can be divided into a number of parts. Some portions of the operating system must always be present in the memory. These sections perform the basic operations such as starting and terminating user programs, allocations of memory and

files, and basic input/output operations. Interrupts are also handled by these resident portions of the operating system. The portion of the operating system which is always present in the memory is called *nucleus or kernel*. The kernel is a master program of operating system. It co-ordinates all other parts of the operating system. It is also called *supervisor*. Other portions of the operating system which are brought into the memory when needed and removed when not needed, are called transient programs.

Booting (or Bootstrapping)

The process of loading the operating system into memory is known as *booting*. When a computer is turned on the operating system must be brought into the computer's memory from the hard disk memory. The process is normally started by a small program called *bootstrap loader*. This program resides in a ROM as firmware. A computer is designed to fetch its very first instruction from the ROM when power is turned on. The first instruction is bootstrap loader. It is a very simple program sufficient only to direct the CPU to look for a specific file (i.e., operating system file) on the disk memory, and execute the instructions stored in the file. The file contains machine codes of the operating system. The first part of instructions in the file contains codes to direct the CPU to continue loading the rest of the operating system into the memory. When the operating system is fully loaded into the memory, the computer is ready to accept user's commands.

Some important operating systems are described in the following sections.

8.17.1 UNIX and XENIX

UNIX is a multitasking and multiuser operating system developed by Bell Telephone Research Laboratories (which are now a part of AT and T Information System) in 1969. It was developed for larger machines. Now it runs on servers and supercomputers. It is also used with powerful 32-bit personal computers. In this system a user is identified with a user ID. A user has to enter a password to get access to the computer.

XENIX is a version of UNIX. It has been developed by Microsoft.

A multiuser system consists of two or more layers. This technique is used from protection point of view. Its inner layers are well protected. Users can not write into the memory area where these layers are stored because user programs can only access operating system resources through very specific mechanism rather than simple, accidental call or jump. UNIX consists of three layers as shown in Fig. 8.9. The innermost layer, *kernel*, is the most important layer. It contains a process scheduler, a hierarchical file structure and mechanisms for processes to communicate with each other. The middle layer is called *shell* which contains command interpreter. The user's commands are decoded and interpreted by this layer. The shell executes user commands and programs. This layer of UNIX provides interface to users. The outermost layer contains programming tools and application programs. The programming tools include assemblers, compilers, editors, debuggers etc. The application programs include packages for database management, word processing, graphics etc.

UNIX is a multiuser operating system. It permits many users to share a CPU on a time-slice basis. Each user's program is known as a *process*. If multiple programs or tasks are to be processed by a CPU, a schedule must be prepared so that they can be processed properly. The parts of the operating system, which perform the task of scheduling is called *scheduler, dispatcher* or *supervisor*.

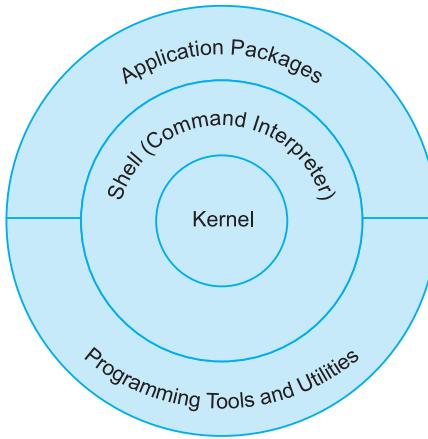


Fig 8.9 UNIX Layers

The UNIX Kernel

The major functions of the UNIX kernel are:

- (i) To schedule and to service the needs of processes (*i.e.*, tasks or programs)
- (ii) To maintain the system file structure
- (iii) To provide a means of communication between processes

The UNIX Shell

This layer of UNIX executes users commands and programs. To perform these tasks it calls kernel procedures (subroutines) as required. Some important features of UNIX shell are:

- (i) A feature of UNIX shell is pipe command. Using pipe command the output data from one program can be sent directly as input data to another program.
- (ii) The UNIX shell permits a user to execute two commands concurrently.
- (iii) A simple method is provided by the UNIX shell to execute a series of commands over and over again.
- (iv) The shell and kernel provide spooling facility. A spooler alternate the user's ongoing activity and controlling of another activity such as printing. A user can send files for printing without worrying about the printer whether it is free at that time. If printer is available the printing work is done by stealing small amount of time between other operations. If printer is not free the print request is put in the queue.

UNIX Utilities/Applications Layer

UNIX has many utilities such as powerful editors, compilers for many high-level languages, debuggers etc. It also includes a large number of application programs. Examples are: database management packages, accounting packages, computer-aided engineering design packages, etc.

Some UNIX Commands

File names under UNIX may consist of any length. A file name may contain uppercase and lowercase characters freely mixed. All UNIX commands must be in small letters. UNIX

has a broad range of disk-related commands. Besides disk commands it also provides more than 100 small commands or utility programs. Some examples of commands are given below.

- ls** This command is used to show a list of files on the disk. The files are normally arranged in alphabetical order.
- mv** mv stands for move. This command is used in renaming a disk file.
- rm** This command will erase (i.e. remove) a file from the disk.
- cat** cat stands for concatenate. This command is used to display the contents of a data file. It is also used to combine two or more files.
- grep** It is the short form of “global replace”. This command is used to search particular word or phrase in one or more data files. It provides a means to locate a particular file even if some one has forgotten its name.
- login** It is used to log in to a computer. It means that one can inform the operating system to be identified. When one logs in the operating system will ask for a password.
- passwd** This command is used to give the password.

Several versions of UNIX are available. These are : XENIX, VENIX, MICRONIX, Berkeley Version, Apple's Version A/UX, IBM's AIX, HP's UX, LINUX, UNIXWARE-7, etc. 64-bit UNIX is available from Digital Equipment Corporation, called 64-bit Digital UNIX. It can address up to 14GB memory. 64-bit UNIX is useful for Internet, Multimedia, very large database, etc.

8.17.2 LINUX

It is a multitasking, multiuser operating system. It is a version of UNIX. It has been developed by hundreds of programmers scattered around the world. The aim was to develop UNIX clone free of any commercially copy righted software, that the entire world can use. It was started by a computer science student, Linus Torvalds. It includes most of the commands of UNIX. It includes several computer science concepts, which were available in MINIX operating system, developed by Prof. A. Tannenbaum. It runs on widely available IBM PCs, DEC's Alpha processors, SUN's SPARC processors, PowerMac computers, etc. It comes with complete implementation of TCP/IP networking protocol.

8.17.3 Windows Family of Operating System

The WINDOWS family of operating system has been developed by Microsoft. It includes the following operating systems:

- WINDOWS – 95
- WINDOWS – 97
- WINDOWS – 98
- WINDOWS – Me (Millennium)
- WINDOWS – XP
- WINDOWS – NT
- WINDOWS – 2000
- WINDOWS – 2003

The WINDOWS – 95, 97, 98, Me and XP are single user operating systems. The WINDOWS – NT, WINDOWS – 2000 and WINDOWS – 2003 are multiuser operating systems.

8.17.4 Windows-95

It is an operating system developed by Microsoft. It includes WINDOWS graphical user interface. It is a mixed 16-bit and 32-bit multitasking operating system. Unlike DOS it does not have the limitation of memory addressing capacity. DOS has memory limitation of 640 KB. WINDOWS-95 takes full advantage of memory addressing capacity of 32-bit microprocessors. It overcomes the entire real-mode access problems by including all operating system functions into a 32-bit architecture. It always runs under protected mode.

8.17.5 WINDOWS-98

It is an improved version of WINDOWS-95 operating system. It is faster and easier to use. It includes graphical user interface. It is a multitasking operating system. It provides multiple windows. It does not need as many mouse clicks and double clicks as needed by its earlier version. It is a 32-bit operating system. A new Outlook Express and E-Mail Manager permits programmers to handle incoming and outgoing messages very easily. The Internet's online environment is more closely associated with WINDOWS-98 desktop computers. It provides improved plug-and-play facility. This enables programmers to plug new devices into the computer without having to set hardware switches or determine appropriate interrupt settings. It includes Microsoft Web browser—Internet Explorer. It provides a feature called Direct Cable Connection which allows to transfer files from one computer to another. This scheme is only for two computers which are connected through a high-speed parallel or serial cable. These computers can share files and printer resource with one another. This technique replaces an expensive network when users want only two computers share files and other resources.

8.17.6 WINDOWS-Me (Millennium)

Microsoft introduced WINDOWS-Me an improved single user operating system in the year 2000. It is an extension of WINDOWS-98. It includes new and improved music, film, digital photo and Internet features. It contains Windows Media Player 7, Movie Maker and Internet Explorer version 5.5. It is faster than WINDOWS-98 and occupies less hard disk space.

8.17.7 WINDOWS-XP

The WINDOWS-XP was developed by Microsoft in the year 2001. It is a widely used single-user operating system for desktop and notebook computers. The term 'XP' stands for eXPerience. It includes features which make it easier to use Internet, to download music and to process digital photos. Microsoft has developed an interface called **Luna** for home and business computers. It is delivered as a part of WINDOWS-XP. It helps people in sharing applications over Internet, online shopping and publishing photos to the Web. It gives users a more streamlined Windows look. It retains the Windows Start button but makes Net connectivity, E-mail access and interaction with system setting via the control panel more intuitive. It has the following four versions:

WINDOWS-XP Home Edition. It is meant for home user. It gives simplicity to home computers. It includes cleaner, more intuitive visual design, advanced media features which make working with digital photograph or playing digital music simple.

WINDOWS-XP Professional. It is the superset of Home Edition. It is designed for professionals and business applications. It is more powerful and faster than Home Edition. It includes all the major features and new visual design of WINDOWS-XP Home Edition, plus premier security and privacy features, advanced recovery options, improved ability to connect to large networks etc.

WINDOWS Media Center Edition. It makes easy to enjoy video, audio, pictures and TV, through convenient user interface.

WINDOWS-XP Tablet PC Edition. It is suitable for notebook computers. The Tablet PC provides pen and speech facilities. The user can use digital pen to write directly on the screen and control computer just as one can do with a mouse. The Tablet PC has also built in wireless technologies. The WINDOWS-XP Tablet PC Edition provides support for all such things.

8.17.8 WINDOWS-2000

The fifth version of WINDOWS-NT which was a multiuser operating system, was renamed as WINDOWS-2000. Its new synchronization manager helps synchronise folders, offline files, Web pages, E-mail calendar information and other applications. It gives better reliability. It is a server platform for business Internet. It acts as an all-in-one Web server, file and print server, application server and network server, and fortified by enhancements like the Active Directory Services (ADS). Its different versions are:

WINDOWS-2000 Professional

WINDOWS-2000 Server Standard Edition

WINDOWS-2000 Advanced Server

WINDOWS-2000 Data Center Server

The WINDOWS-2000 Professional is for workstation and other versions are for different type of server requirements.

8.17.9 WINDOWS Server-2003

The WINDOWS Server 2003 is a multiuser operating system, developed in the year 2003 by Microsoft. It is an ideal server platform for organization of any size. It provides infrastructure platform to serve connected applications, networks and Web services from the workgroup to the data center. It also provides an application environment for building, deploying, managing and running XML Web Services. Its different versions are:

WINDOWS Server-2003 Standard Edition

WINDOWS Server-2003 Enterprise Edition

WINDOWS Server-2003 Data Center Edition

WINDOWS Server-2003 Web Edition

8.17.10 SUN's Solaris

Solaris is a 32-bit operating system of SUN company. It provides graphical user interface. At present its version-2.6 is available. SUN's subsidiary SUNSOFT developed its own version of UNIX, called Open Windows. The Open Windows includes Berkeley derivative of UNIX along with Open Look graphical and look-and-feel desktop manager. This entire bundle is known as Solaris.

SUN developed 64-bit operating system, Solaris version-7. Solaris-7 installation and administration have been made more user friendly with 'Wizards' and other user-oriented automation tools. SUN's SPARC Processors or Intel processors can run under Solaris-7. 64-bit Solaris is downward compatible with 32-bit programs and in fact runs them faster than they run under 32-bit Solaris.

Recently, SUN has developed Solaris-8 operating system. It is SUN's version of UNIX. Future SUN server computers using UltraSparc III processor will use Solaris-8 OS. Applications running on Solaris-2.6 and Solaris-7 will run on Solaris-8.

8.17.11 Apple's Macintosh Operating system

It was introduced in 1984. It was the first operating system, which used graphical user interface (GUI). It was widely used on Apple's computer built around Motorola 680X0 series of microprocessors. Its latest version is Mac OS 8.5. It is used on PowerPC processors. It provides full memory protection and process management. While its microkernel enables UNIX-like process management, the protected memory puts each service in its specified slot of dynamically allocated memory. It is faster and it can easily be installed. It has improved Internet search functionality. From a single dialog box, a user can search files by content, and search Internet using several different engines, such as Excite and HotBot, simultaneously. Coloursync printout, video or web-site of what users see on their screen is an area where Macintosh operating system excels over WINDOWS platform.

Recently Apple has developed Mac OS-X operating system. It provides object-oriented programming environment. Its microkernel provides LINUX like facilities.

8.17.12 Novell NetWare

It is a network operating system. It is a multitasking, multiuser operating system. It gives fair time-slices to processes. Computers on a network share files and resources like printers, plotters, storage devices and computing power. It provides E-Mail facility. Its recent version Novell-5.0 offers services in Internet environment besides features of providing facilities to computers on the LAN to share files and resources. Novell-5 offers centralized network management services IPX-to IP support which enables IPX-based LANs to connect to IP-based network.

8.18 UTILITY PROGRAMS

The utility programs are software tools to help programmers in developing, writing, debugging and documenting programs. The utility programs can be placed into two categories: file management utilities and program development utilities. The file management utilities are the parts of the operating system. They help programmers in creating, copying, printing, erasing and renaming files. Program development utilities help users in writing and testing programs. These are text editor, assemblers, compilers, loader, locator, linker, debugger etc. The loader, linker, locator and debugger are included in the operating system. A simple text editor may be included in the operating system but an elaborate text editor is stored separately. Compilers are loaded separately. Programs loaded separately also run under the operating system. The operating system includes the file and memory management subroutines and utility programs. Some utility programs which have not been discussed earlier are discussed in the following subsections.

Editor

An editor is a program which permits programmers to enter, modify and store a source program or text. Some examples of editor are WordStar and Edlin. To write a program or create text, an editor is called under the control of the operating system. When an editor program is transferred from the disk memory to RAM, the control is transferred from the operating system to the editor. The editor has its own commands which are used to enter and modify text or a program.

File Manager

A file manager is a program which permits programmers to create, copy, update or delete files on the disks. A file is a collection of information supplied by the programmer. It is stored on the disk as a single entity and is accessible by a name. A directory of files stored on the disk is also maintained by the file manager. The directory itself is also stored on the disk. The directory includes file name, size of the file, date and time, and unused memory capacity of the disk etc.

Loader

The loader is a program that loads machine codes of a program into the system memory. It accepts programs either in absolute or relocatable format. If a program is in absolute format (i.e. the actual addresses of the instructions and data are supplied by the programmer), the loader simply loads the program into the system memory. If a program is in relocatable format, the locator assigns specific addresses to each instruction and data before the loader loads the program into memory.

Locator

A locator is a program that assigns specific memory addresses for each machine code of a program which is to be loaded into the system memory.

Linker

Usually, a longer program is divided into a number of smaller subprograms called modules. It is easier to develop, test and debug smaller programs. A linker is a program that links (combines) smaller programs to form a single program. While developing a program subroutines are frequently used. The subroutines are stored in a library file. The linker also links subroutines with the main program. The linker links machine codes of the programs. Therefore, it accepts user's programs after editor has edited the program and compiler has produced machine codes of the program.

8.19 APPLICATION PACKAGES

An application package or application program is the software that has been written to process or perform specific job. The term application indicates that the software is applied to the job at hand. Application packages are available for all types of tasks. They may be for business applications, engineering designs, home applications, teaching aids etc. Some examples are: word processing packages for processing and manipulating text, spreadsheet packages for calculating finances and data analysis, CAD for designing or drafting, dBASE to work with records (management of files, management of database, storage and retrieval of information), engineering design packages, MATLAB for scientific and engineering computation, SIMULINK for building models of dynamic system, communication packages etc. Some integrated software

packages have also been developed which combine the functions of word processing, spreadsheets, graphics, database management and communication into a single software, for example, Framework, Symphony, MS-Office etc.

General features of some application packages are described in the following sections. Some application packages for specific purposes are also discussed in the next chapter.

8.19.1 Wordprocessing Packages

A word processing package (or program) is the software which is used to process text. It allows user to enter, view, edit, manipulate, transmit, store, retrieve and print text material. The text material may be letters, reports, notes, thesis, book, invoices, projects, or anything else. It is very easy to make corrections for typing or any other kind of errors. Changes and modifications in text can also be made at any time very easily. Editor provides commands to insert or delete words, sentences or even paragraphs. Editing commands are there to move word, sentences or paragraphs from one place to another. The left and right margins, line spacing can be set as desired while printing the text.

A word processing package runs on a personal or any other computer. A PC or any other computer is a general purpose machine. It can perform different types of tasks including word processing. They are not limited to the task of only word processing. Dedicated machines are also available which perform only the task of word processing. More powerful and costly system, desktop publishing system, is also available, it will be discussed in Chapter 9.

As an operator enters characters from keyboard, they are displayed on the screen. When the operator reaches the end of a line, the word processing program automatically moves the text to the next line. This feature of the word processing program is known as *word wrap*. When a word being typed is too long to fit at the end of the current line, it is automatically shifted to the next line. Unlike the typewriter, the operator need not press the return key at the end of each line of the text. He has to press the return key only at the end of a paragraph. Most of the word processing programs allow to use underlining, *boldface* type, *italics*, capitals, subscript (below the line) and superscript (above the line). If *justification* feature of word processing program is used the right margin is automatically aligned. It adjusts the spacing between the words to produce alignment at the right margin.

Page Breaks. This command begins the next page of the text. So it is inserted in the text at places from where we want to begin the next page.

Formatting or Page Design. The operator has to mention left and right margins, top and bottom margins etc. Spacing between lines or number of lines per page, number of characters per inch or per line etc. are also to be specified. All such information can be specified at the time of entering a text. But these can be changed as desired before taking prints out. A heading can be centred if desired. While printing a text the word processing program automatically makes the numbering of its pages. The program also maintains the proper numbering sequence of the footnotes even if some footnotes are added or deleted at the time of editing.

Editing of Text. Powerful set of editing commands are available in a word processing program, to edit the text after it has been typed in. One can make corrections and modifications in the text. Something can be inserted or deleted. There is also a block move command. A paragraph or a sentence can be moved from one place to another. Margins and line spacing can be changed. Then the revised version of text can be printed.

Scrolling. When a text becomes long the user can move the text up or down using Pg Up or Pg Dn key to view the text. This is called scrolling. The scrolling is also used to move through the document quickly until the desired page is located. Besides push buttons a modern mouse has also a scroll wheel. The scroll wheel can be rotated with a finger. By rotating the scroll wheel, the content displayed on the screen can be moved up or down as desired.

Search and Replace. This command searches through the entire document to locate each occurrence of a particular character string and replace it with another. A character string is a combination of letters, numbers, and/or other symbols. It may be a single word, a part of a word, a group of words, a number and so forth. Suppose there is a typing mistake, "computor" instead of "computer". This mistake can be corrected automatically throughout the document using this command. One has to give three answers when this command asks the following.

Find? Computor

Replace with? Computer

Options? Global (*i.e.*, entire document)

Similarly, we make some changes throughout some text. For example, Information System is to be replaced by Information Services. Take another example: suppose one has to type "Electrical Engineering Department, Bihar College of Engineering, Patna-5" at several places in a project report. One can type a symbol for this such \$ as each time when needed. After finishing the writing work he can ask the program to replace \$ by the desired group of words.

On and Off-Screen Formating. In on-screen formating what one sees on the screen he gets the same when printed. In this type of formating boldface type, italics, capitals etc. are shown on the screen as desired. This type of formating is called WYSIWYG (What You See Is What You Get), pronounced as wizzy-wig. In some older word processors there is off screen formating. In this scheme some control symbols are displayed for boldface, italics etc. on the screen. The results are not visible until the text is printed out.

Spelling Checker. A word processor program includes a spelling checker component. It checks every word of the document. For this purpose there is an electronic dictionary stored in the memory. The spelling checking is a mechanical process. If a word does not match any word stored in the dictionary, it is identified. The operator is asked to correct the identified word. In some systems spelling checkers check the spelling of the words as they are entered (typed) and give some sort of indication if a word is not found in the dictionary. In other systems, the spelling checker can be called on after the typing work is completed. If an operator uses some unusual or technical words, these can be added to the dictionary so that the spelling checker will recognize them when they appear again.

Syntax Checkers. A word processing package may also include a syntax checker. It checks for improper syntax, poor usage of words and other related items.

The Electronic Thesaurus. Sometimes we feel that a particular word is not suitable or quite right in the text where it has been used. In such a situation we want to know suitable synonyms. An online thesaurus displays a list of alternative words on the screen. A synonym is selected from the list. It can be inserted in the document to replace the original word.

Mail-merge. Many word processing packages include mail-merge facility. Using this facility some information (data) can be copied into a document (which is being processed by

word processor package) from other package or file or database. Suppose one wishes to send same letter to many persons. Let the file containing the names and addresses of the persons who will receive letters is available separately. First, letter's contents are entered, omitting names and addresses, using word processing package. Using mail-merge command word processing package gets names and addresses from the file. For each name one copy of the letter is printed. Offices, institutions, companies often issue the same letter to many persons. For this type of work this facility is very useful.

Ability to Print Graphics and Text Together. Some word processors are provided with this facility. This is very useful for advertising material, financial reports and brochures. Bar graphs and pie charts can easily be included in financial reports using this facility.

Important word processing packages are : MS-Word, WordStar, Lotus WordPro, Corel's WordPerfect, Akshar (Hindi/English), Sabdaratna (Hindi/English), Sabda Sagar, Shree Lipi (multilingual for Indian languages), etc.

8.19.2 Electronic Spreadsheet (or Worksheet) Packages

Electronic spreadsheet package is a software tool for data analysis. It allows users to quickly create, manipulate and analyze data organized in columns and rows. A spreadsheet is a sheet of paper containing rows and columns in which data can be entered, manipulated and analyzed. When a computer prepares spreadsheet it is known as *electronic spreadsheet*. It is displayed on the screen, stored in the memory and its prints out can be taken. Electronic spreadsheets are very helpful where large tabulated computations are used. Original spreadsheets were created in business context to maintain accountant's ledger and to perform other accounting work. Nowadays, they are used in several fields which deal in numbers. They are also very useful in engineering. Spreadsheet programs are widely used in accounting, sales, inventory control, financial aspects of business, tabulation of marks, and a number of business and nonbusiness problems. Today several spreadsheet packages are available which can be used to prepare budgets, forecast future cash needs, determine break-even sale points, prepare examination results etc. Examples of spreadsheet packages are: Lotus 1-2-3, MS-Excel etc.

Creation of Spreadsheet

When a spreadsheet program is initialized, the CRT screen displays blank rows and columns as shown in Fig 8.10 (a). The rows are numbered and columns are lettered. A spreadsheet program allows users to have hundreds of columns and thousands of rows. The screen will show a few columns and rows at a time. By scrolling technique the user can see any portion of the spreadsheet. The intersection of each row and column creates a rectangular box called a *cell* on the spreadsheet. The address of a cell is given by the letter of the column and number of the row to which the cell belongs. For example, the cell created by the intersection of the first column and first row is addressed by A1. Similarly, C5 is the address of the cell created by the intersection of the third column and the fifth row, and so on. A cell can contain either text or number. To enter a continuous text adjacent cells can be used together. The contents of a cell also may be classified as *label*, *value* and *formula*. A formula gives relationship between two or more cells. A label is a set of characters. It is used for headings, titles and descriptions. A label begins with a letter. A value is a number (data) to be manipulated or analyzed. A formula can also be put in a cell to compute its contents. It may be a function of values of other cells. A formula begins with digits and +, -, @, (, \$, and ^. A data or value also begins with digits and these special characters. To locate

a cell a special rectangular cursor or screen marker called *cell pointer* is used. Its movement can be controlled with control keys or a mouse. To enter a value or label or formula into a particular cell, first locate the cell and then type in the desired data. The text or number can be left-justified, right-justified or centered within a cell.

A1		READY			
	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

Fig. 8.10 (a) Empty spread sheet

Two or more adjacent cells are referred to as a **range** of cells. The ranges indicate the first and the last cell in a row, column or a rectangular area. Examples of range are:

B3:B10—All cells in the column B from the row 3 to 10.

B3:H3—All cells in the 3rd row from the column B to H.

C3:G20—All cells in the rectangular area bounded by C3 at the upper left and G20 at the lower right.

The display screen is divided into two sections: the control panel (or summary panel) and the spreadsheet itself. The upper left portion of the screen (*i.e.*, the left portion of the control panel) is known as *edit area*. The edit area is also called *scratch area* or *status line*. In Fig. 8.10 (a), the cursor is located at cell A1. This location is displayed in the edit area of the control panel. The cell pointed out by the cursor is called *active cell*. The upper right portion of the control panel (called mode indicator) displays READY indicating that entry can be made in the spreadsheet. A title “STUDENTS GRADE SHEET” is entered in the first row of spreadsheet. The spreadsheet in Fig 8.10 (b) shows name of the students, their marks, class average in each test and final examination marks, total marks, and average grade of each student in percentage. The class average and average grade of the student are calculated by the spreadsheet program. The user has to type the formula in the corresponding cell. For example, in cell B10, a formula = (B6 + B7 + B8)/3 is to be used to calculate class average. The user will type this formula in cell B10. This formula will not appear in the cell B10. It will appear in the edit area as shown in Fig. 8.10 (b). The calculated value will appear in the cell B10. Similarly, total marks of each student is calculated in column E and the percentage average grade in column F.

It is not necessary to type = (B6 + B7 + B8)/3 in B10. It can be easily obtained by some tricks. When the cursor is in B10, the edit area will show B10. After this type = (Then bring the cursor in B6. So B6 will appear in the edit area. Type + after B6. Again the cursor will automatically come to B10. Again move it to B7. So B7 will appear in edit area after the

+ sign, i.e., edit area will now show = (B6 + B7). Again type + sign. It will bring the cursor to B10 again. Now bring cursor to B8. The edit area will show = (B6 + B7 + B8). Then type). It will bring the cursor to B10. Then type /and 3 to complete the required formula. By this procedure you get the correct addresses of the required cells automatically in the formula. You need not type them. There are chances of errors and the process will be slow, if you try to type them in the formula. Instead of writing the formula = (B6 + B7 + B8)/3, one can use a built-in function = AVG(B6:B8) in the cell B10(B6:B8) indicate a range from the cell B6 to B8. This will give the average value in B10. Similarly, to get total in E6 one can use the formula + B6 + C6 + D6. Alternatively, one can use the built-in function = SUM (B6 : D6).

B10: (B6 + B7 + B8)/3							READY
	A	B	C	D	E	F	G
1							
2							
3		STUDENTS GRADE SHEET					
4	Full Marks	Test 1 50	Test 2 50	Final 100	Total 200	Av. Grade Per cent	
5							
6	Mohan	45	47	90	182	91	
7	Ram	44	43	85	172	86	
8	Ganesh	31	36	77	144	72	
9							
10	Class	40	42	84	166		
11	Average						
12							
13							
14							
15							

Fig. 8.10 (b) Spreadsheet

Some Important Commands of Spreadsheet

Copy Command. The copy command can be used to copy the contents of one cell (or group of cells) into another cell (or group of cells). This feature is particularly very useful when the same formula structure is to be used in several cells. For example, the formula for the cell E6 is = (B6 + C6 + D6). The same structure is to be used for the cell E7 and E8 i.e. = (B7 + C7 + D7) and = (B8 + C8 + D8) respectively. Instead of writing these formulas for E7 and E8 a relative copy command will be used, which will use the formula structure of cell E6 and make the modifications accordingly. Similarly, the formula for cell B10 is = (B6 + B7 + B8)/3. The formula for C10 and D10 are = (C6 + C7 + C8)/3 and = (D6 + D7 + D8)/3. Instead of writing these formulas, the formula structure of the cell B10 will be copied for C10 and D10.

Insert and Delete Command. An *insert* or *delete* command allows users to add or delete a column or a row anywhere in the spreadsheet. The addresses of all the cells which move, are automatically changed by the spreadsheet program. The program also adjusts the formula reference to those cells.

Window Command. The screen can be split either vertically or horizontally using this command. Suppose there are some columns which are not appearing on the screen. When those columns are brought on the screen, the column A which contains certain labels disappears. So one can split the screen using window command, and see column A in one window at the left of the screen, and those last columns in a second screen window on the right side of the screen. This facilitates data analysis of those columns which are brought on the screen later on by scrolling.

Format, Save and Print Commands. There are various fromatting commands which can be used to control the way the columns, rows and cells appear on the screen. For example, the width of a column can be adjusted; left or right justification of labels or values can be made. The *save* (load) command is used to store the spread sheet on a disk. The print command is used to print the spreadsheet or part of it on paper.

Built-in Functions. Instead of writing a formula for sum = (B6 + B7 + B8 + B9 + B10) one can write @ SUM (B6 .. B10) in a particular cell. It will take the sum. Similarly, a function can also be written for = (B6 + C6 + D6 + E6 + F6) as = SUM (B6 : F6). Other built-in functions are: AVG (range), MAX (range), MIN (range) etc. AVG is used to find average, MAX is used to search cells for maximum and so on.

MS-Excel: It is spreadsheet (worksheet) application program. It is a part of MS-Office which is a suite of application programs. It is also available as a stand-alone application program. It is a software tool for data analysis. It allows users to create, manipulate and analyze data organized in columns and rows. The spreadsheet is displayed on the screen, stored in the memory and its prints out can be taken. Spreadsheet programs are widely used in accounting, sales, inventory control, financial aspects of business, salary preparation, income-tax calculation, tabulation of marks, etc. Excel provides various chart (pie chart, bar chart etc.) and layout options. Microsoft Office-XP provides features of voice dictation and voice activated commands for MS-Excel. The user can dictate information and can use voice commands to activate the menu system.

General description of spreadsheet (also called worksheet) has already been given in the Section 8.19.2. There, it has been discussed how to calculate SUM, AVERAGE etc. using built in formulas or making own formulas. Besides these features, the MS-Excel has also an AutoCalculate, one can easily find the SUM, AVERAGE, MAXIMUM, MINIMUM etc. of the selected cells easily. One has to initiate AutoCalculate, select the cells of a column and choose the formula for the desired function from the functions of AutoCalculate displayed on the screen. In MS-Excel there is also an AutoSum button on the Standard toolbar. Using this one can find the sum of the desired cells of a row or column.

Templates

Many people use spreadsheet for same type of data analysis. They do not like to enter labels and formula needed for data analysis. They want to enter only values (data) and make analysis of their common type of problem. For such purpose spreadsheet templates are available. A *template* is a spreadsheet which contains all the labels and formulas needed to create a spreadsheet for a specific type of data analysis. A template does not contain values in the data cells. The user merely enters values in the data cells while making data analysis. A large number of spreadsheet template products are on the market now, and the number is growing every day.

8.19.3 Special Purpose Data Analysis Packages

General purpose spreadsheet packages handle any type of data analysis problem in which data are organized in columns and rows. The problem may be of investment, accounting, engineering, tabulation of examination results, statistical analysis etc. But today packages for special purpose data analysis are also available. Such a package is for one type of data analysis, for example, investment analysis packages, accounting packages, statistical analysis packages. Investment packages are used to analyze investment in securities (stocks, bonds). The types of investment packages available are: fundamental analysis packages, technical

analysis packages, portfolio management packages and personal financial management packages. The types of accounting packages available are for sales analysis, inventory control, payroll calculation, order entry/shipping/billing, general ledger etc. Integrated package for accounting is also available for small business organizations. Such a package includes programs for general ledger, order processing, accounts receivable, accounts payable, inventory control and payroll. Statistical analysis packages are similar to spreadsheets. Some statistical functions such as @ AVG are included in the package. But other formulas are to be entered for analysis. Some statistical packages preprogrammed with all the specialized formulas and built-in procedures are available. Many packages can accept data from other files; sort, merge and make manipulation in various ways; select samples; perform analysis on single and multiple sets of data; produce charts and graphs and print results. For details one can refer to the book mentioned in Ref. 4.

8.19.4 Graphics Packages

It is a well known fact that information presented in the form of graphical and pictorial form is very easy to understand. It is often remarked that one picture is worth a thousand words. Preparation of drawing, graphs, pictures etc. with hand is a time-consuming process. Today, computer with graphics softwares facilitates the preparation of graphs, picture, drawings etc. It enables users to quickly convert tabular data to graphical form. Users no longer depend on draftsman and artist. Sophisticated graphs or drawings in multicolour can be drawn in minutes by pressing a few keys of computer keyboard. Computers have also the capability of drawing pictures and three-dimensional drawings or pictorial views.

Business organizations use graphic packages to present data in the form of bar charts, pie charts, line charts etc. Fig. 8.11 shows bar charts and pie charts. A graphics package is also used in text publishing. A desktop publishing package includes graphics to produce pictures and other forms of graphics in the text where needed. In the field of engineering graphics packages are used to draw engineering drawings and various kinds of graphs. The graphics packages are not only the means of displaying high-resolution drawings but also they provide the capability of manipulating and modifying drawings and pictorial views. A package can provide an engineer with a succession of different views of a design, such as building or an aircraft. The scale of drawing can be changed so that either an entire assembly or a small section of the assembly can be examined. Pictures can be rotated, contracted or expanded while displaying. Another important application of computer graphics is in animatin *i.e.*, to add the dimension of time to spatial dimension. Moving pictures can be seen and analyzed. Moving pictures are of great value in understanding the dynamics of objects and systems. Computer graphics have abiltiy to produce such images.

Statistical data stroed in a number of pages can be represented by a few graphs, charts, maps and other visual aids. The huge amounts of data stored in alphanumeric form can easily be highlighted with a few graphic presentations. Graphics packages are alos used to recognize pictures and to process images. Paint packages are also available. Artists can use cimputer to design and draw paintings.

CAD/CAM. CAD stands for computer aided drafting or computer aided design. CAM stands for computer aided manufacture. CAD facilitates preparation of engineering design and drawings. It is used to design electrical, electronic, mechanical and structural components and systems, and so on. Extensive use of graphics is made in designing printed circuit board

layouts, design of LSI/VLSI chips, aircraft structures, ships' hulls, building structure, chemical plant piping layout etc. Engineering workstations are commonly used for CAD purposes, but CAD packages are also available which run on PCs. AutoCAD is a CAD package developed by Autodesk Inc. It is registered in U.S. Patent and Trade mark office. A CAD system provides an environment for manipulating graphical symbols with a computer. AutoSLIP is a language to manipulate symbols. AutoSLIP is a very good language for programming a CAD system. Its programming and other features are used for advanced work with CAD.

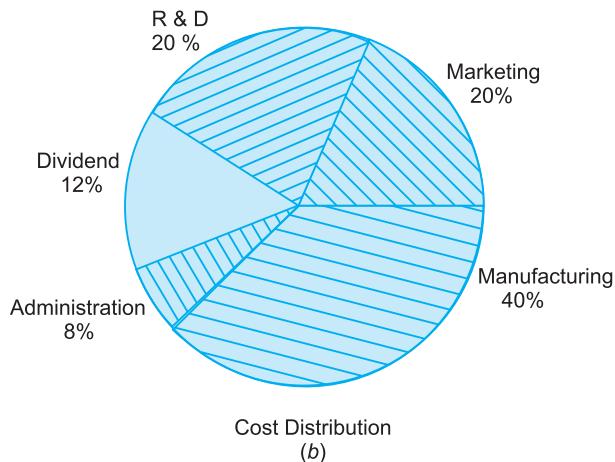
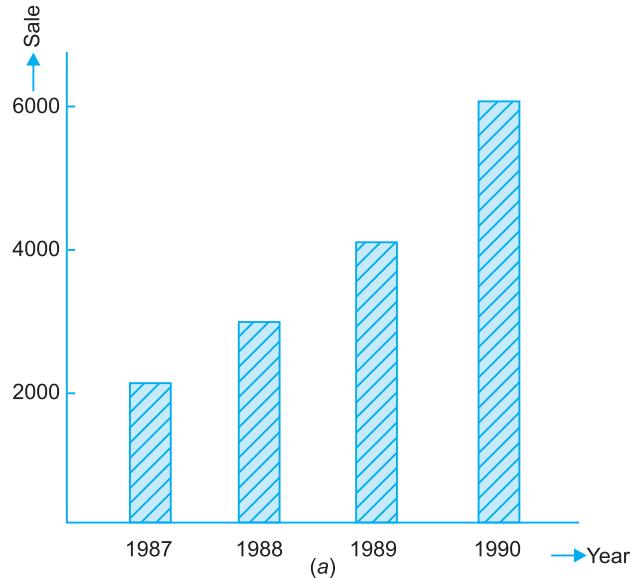


Fig. 8.11 (a) Bar charts, (b) pie charts

Nowadays computers are widely used in manufacturing process also. Information produced by CAD system are utilized in CAM. CAD and CAM are used as an integrated system. In such a system engineers and designers design new machines and their parts using CAD. The drawings are prepared with the help of CAD immediately. New drawings are available for manufacturing process. In a CAD/CAM system database is used to store all design data,

drawings, dimensions, specifications, etc. On the factory floor manufacturing personnel are on the same computer network. The design information is obtained from database. This information is utilized to run lathes, milling machines, multiple drill presses, assembly lines and so on. A series of computer-controlled machines perform all tasks to fabricate parts. Thus an industry may use full automation for its products using CAD/CAM. Some more details are discussed in the next chapter.

Graphics packages are often integrated with spreadsheet, data analysis, investment and other data analysis packages. After data analysis and manipulations users want to see the results quickly. Graphics are the convenient means for such purposes. Graphics packages are also used in fields other than data analysis for example, they are included in desktop publishing packages, they are also used by designers, engineers, architects, artists etc.

8.19.5 Data Base Management Packages

The word *data* is used to refer to a fact or facts about a person, place or thing. Data and information are not the same thing. Data arranged in certain order and form which is useful to us is called *information*. Data are raw material to get information. Data are to be processed to produce information. Thus information is the output of data processing operations. We get information to enhance our knowledge and to achieve certain purpose.

Database management systems are software packages which permit users to work with records. A *database* consists of one or more related files. A *file* is a collection of records. A *record* consists of *fields* (or *data elements* or *data items*) about a given subject. A database management package allows users to create database. Users define data fields and make entry of fields into records. Then they combine records to create files. Now users can manipulate and retrieve stored data in database in various ways to prepare reports and collect certain desired information arranged in certain order.

In a relational database a file is arranged in a tabular form. The table contains rows and columns. Each column of a table represents field and each row represents a record. The relational database management package has ability to manage data in more than one file. The data of one file is related to the data of the another file through the use of a common field.

We consider some examples to illustrate what type of work can be done using a database management package. Suppose the fields of the records of a file are: customer number, name, locality, city, PIN. Collection of these informations about customers creates a file. From these stored data (information) we can get the answer to certain questions. We can ask the computer to prepare the list of the customer who belong to Patna. Again we can ask computer to print the names of customers who came from Bailey Road, Patna. Similarly, we can give certain customer number and ask to print all the details about the customer who has been allotted that customer number.

Take another example of a database created to keep the detail records of an employee of an industry. The information stored is: name, identification number, age, experience, qualification, specialization, the section in which the employee is working etc. From this database one can find the list of persons who specialized in a particular field say electronics, computer etc. Again one can find the list of persons who are electronics engineers and have an experience of 15 years and more. Further, one can find the list of persons who are electronics engineers, have 15 years of experience and have knowledge of designing microprocessor-based automatic control system such as a system to control temperature of

furnaces, to control speed of electric motors, to measure stress, strain, force etc. This type of sorting is very easy by database management packages. Similarly, if any one wants to know the detailed information of an employee, he can immediately find it supplying identification number of the employee. So searching is also very easy. If some information required involves two or more related files, one can find the information immediately.

Indexing

To maintain the records in a file in certain sequence is a problem because records are added or deleted time to time. Suppose, a file keeps records in alphabetical order. If a name starting from letter B is added, it needs movement of some records, changes in their record number which is in a serial order. To solve this problem an index is provided. Each record has an additional field indicating a pointer which indicates the next record. So any addition can be made at the end and the pointer provided for it will indicate its correct position in the file where it is. A few more changes are also to be made. Suppose, a file contains 5 names. A sixth name is to be added. Its position in the file should be third according to alphabetical order, so an index against this name will be 3. The name which is in position 2 and shows next record 3 in its pointer, requires a change in the pointer from 3 to 6. Fig. 8.12 (a) and (b) shows this technique. They keep record to their cash balance, names are in alphabetical order.

Record Number	Customer Name	Balance Rs.	City	Pointer to Next Record
1	Akhilesh Kumar	500	Patna	2
2	Bihari Prasad	400	Delhi	3
3	Chhotan Singh	800	Patna	4
4	Devi Pandey	300	Patna	5
5	Hari Prasad	200	Kanpur	EOF

EOF = End of file

Fig. 8.12 (a) Indexing technique

Record Number	Customer Name	Balance Rs.	City	Pointer to Next Record
1	Akhilesh Kumar	500	Patna	2
2	Bihari Prasad	400	Delhi	6
3	Chhotan Singh	800	Patna	4
4	Devi Pandey	300	Patna	5
5	Hari Prasad	200	Kanpur	EOF
6	Brajesh Prasad	700	Delhi	3

EOF = End of file

Fig. 8.12 (b) Technique to change pointer to next record.

The main objective of any database management system is to provide information from the stored data. In fact data and information are not the same thing. Data are converted to information by certain processes such as sorting, selection and calculation. The database management packages have also data manipulating (data analyzing) capability which helps providing some desired information. Many database management packages can also produce graphs from data in the database.

A number of database management packages are available such as dBASE, MS-Access, ORACLE, UNIFY, FOXBASE, MS-Visual Foxpro, INGRESS, SYBASE, INFORMIX, etc.

MS-Access: It is a relational database management system (RDBMS). In a relational database a file is arranged in a tabular form. MS-Access is designed to create database tables, data entry forms and reports based on the records which programmer creates. It allows to arrange data and use them for queries and searches to give desired information. It helps programmers to manage large amounts of information and to show relationships among records.

Oracle: Oracle is a relational database management system. It has been used in multiuser system. Now it is available for single-user computers also. It is SQL-based database system. SQL (pronounced as SEQUEL) stands for Structured Query Language. It is a language which enables us to create and operate on relational database.

8.19.6 Integrated Packages

In the previous sections we have discussed separate application packages for word processing, spreadsheet analysis, database management, graphics and communications. But on many occasions users require more than one application packages to prepare certain documents. Suppose, a user is writing a report using word processing package and decides to include some data from a spreadsheet. With separate packages it is not an easy task. He will face a lot of difficulties. Take another example in which a user wants to take data from database for the spreadsheet he is preparing. Further, he wants to print graph from spreadsheet data. All these are very cumbersome and difficult operations to achieve. For such tasks integrated packages are available. An integrated package includes word processing, spreadsheet, database management, graphics and communications capability. An integrated package allows users to change from one application package to another. It also allows to transfer data back and forth between application packages. For example, an integrated package permits users to transfer some data from a database directly into spreadsheet, from the spreadsheet data can be sent to draw graphs. A document produced by the word processor can include both the spreadsheet and the graphs. All these are done in a co-ordinated manner without switching program and/or shuffling disks.

MS-Office, LOTUS 1-2-3 are integrated package. It has already been described. More powerful integrated packages are *Framework* and *Symphony*. Framework has been introduced by Ashton-Tate. It combines word processing, spreadsheets, a database management, graphics and communications capabilities into a single application program. Symphony is developed by Lotus Development Corporation. It combines a spreadsheet with graphics, wordprocessing, database management and communications, plus an advanced windowing capability and an advanced command language into one comprehensive integrated package. It is very powerful and flexible spreadsheet for microcomputers. It is bigger, faster, and offers more commands compared to any other electronic spreadsheet package. MS-Office is an integrated package, described in the Section 8.19.7.

If user has separate single function packages, he can purchase a software to co-ordinate them. Such a software is called *system integrator* or *integrating software shells*. The application packages designed specifically to operate in a given shell environment should be purchased. In such a situation all the user programs will respond to the same commands and queries.

In many cases a user needs one applications program for most of the time but occasionally needs some simple operations from other application programs. For example, a person working with word processor may require a calculator for simple arithmetic operations (he may not need a spreadsheet), a person working with spreadsheet program may require a notepad (he may not need a word processor program) to write a short report, and both of these persons may require a program (not a database package) to store and retrieve names, addresses etc. For such work, a simple integration approach will be to use memory-resident package along with the single function application. Examples of memory resident packages are keyboard enhancer, calculator, notepad, card filer and auto dialer, appointment calender and alarm clock etc.

8.19.7 Desktop Applications

Usually, an office needs word processors, spreadsheet package, presentation package, etc. Various companies have developed software suite consisting of several useful software packages. Examples are : MS-Office, Lotus SmartSuite, QuarkXPress, Corel WordPerfect Suite, Leap Office, Karyalaya 2000 (multilingual office automation).

MS-OFFICE: Microsoft Office is a set of application programs packed together. It includes a word processing package-'WORD', a spreadsheet (worksheet) package-'EXCEL', a database management package-'ACCESS', a presentation package-'POWERPOINT', and a scheduling and organization package-'OUTLOOK'. Each of the packages of MS-Office can be used separately, or they can be used together where needed. For example, a document prepared in MS-Word can use charts and graphs created in Excel. MS-Office is an integrated package. Its different application programs can be integrated easily. A brief description of these packages (application programs) is given below. Its latest version is Microsoft Office-XP. It includes **Front Page**, a Web design tool.

MS-Word: It is a word processing application program (package). It is a versatile application program, which allows to create documents that can contain texts, tables, charts, drawing, etc. It includes a spelling checker which checks the spelling of the words as they are entered (typed) and gives an indication (underline with red colour) if the word is not found in the dictionary. It also indicates grammatical mistakes, improper spacing between words etc. Such mistakes are underlined with green colour. MS-Word is available as a part of MS-Office or as a stand-alone application program.

Microsoft Office-XP includes the feature of voice dictation and voice-activated commands for MS-Word package. The speech feature of the Office-XP has to be trained for the user's speech patterns and intonation (rise and fall of the voice in speaking). Then, the user can dictate text into the Office document. Before giving dictation the user has to select 'Tools menu', 'Speech' and click 'Dictation'. To give voice commands one can click the Voice Command icon or can speak "voice command". After giving voice command, the user can open a menu by telling the name of the menu. For example, the user can speak 'format' to open format menu. Then to open a submenu such as font, the user can say "font". After this the user can navigate a particular dialog box using voice commands. For example, in the Font dialog box, to change the size of the font, the user can say "size". This activates the size box which can control font size. Then the user has to tell the size of the font, such as "14". Similarly, the user can also activate some other font attributes in the dialog box. After the user has finished working with a particular dialog box, he can tell "OK" or "Cancel". One can also activate buttons on the various toolbars using voice commands. For example, one can give command bold by

saying “bold”. The Bold button on the formatting toolbar will become active. To turn bold off, user has to say “bold” again.

MS-PowerPoint: It is a presentation package. It helps user to create interactive, self-running, or speaker-controlled visual displays. It uses multimedia technology to include text, photographs, drawing, graphs, video and audio clips in a presentation. Presentation created in PowerPoint can be used to accompany lectures or as the basis for Websites. PowerPoint can also be used to prepare 35 mm slides, overhead projections and printed handouts.

MS-Outlook: It is a scheduling and organization package. It provides user with a calendar, contact list, task lists and notes. It also creates and manages E-Mail service. Its components are fully integrated. Suppose a user sends E-Mail scheduling a meeting, the appointment will automatically shows up on the calendar. Outlook is very useful in a network environment.

Front Page. It is a Web design tool. It provides wizards to create several types of Web sites. Using FrontPage user can quickly create a personal, business or any other kind of Web site. The user can include graphics and navigational tool, such as hyperlinks.

MS-Excel and MS-Access have already been described in Sections 8.19.3 and 8.19.5 respectively.

8.19.8 Desktop Publishing

Desktop publishing packages provide facilities to include text, images, graphs, etc. on a page. These types of facilities are needed in publishing books, news papers, bulletins, magazines, etc. Examples of some desktop publishing packages are : Corel's Ventura, Adobe's Page Maker, Quark Inc.'s Quark XPress, Microsoft Publisher 2000, Akruti Publisher, etc. Desktop publishing has been discussed in Chapter 9.

8.20 BIOS

The abbreviation BIOS stands for Basic Input Output System. Some system programs that manage the computer's operations are stored in ROM as firmware. These programs perform the most fundamental types of supervisory and support work. They provide essential services that all programmers need. These service programs (or subroutines) are called Basic Input/Output Services. They are referred to as BIOS or ROM-BIOS. Each program (or subroutine) performs one specific task such as writing some characters to the CRT, reading some data from disk memory or reading a character from the keyboard etc. All these are done using software interrupt instruction (INT instruction in IBM PCs).

The services provided by ROM-BIOS are: print screen, video services, disk services, serial port (RS-232) services, cassette port services, keyboard services, parallel port (printer) services, boot strap etc.

Other system programs are built on the foundation of ROM-BIOS programs and provide higher-level support services. Operating systems are examples of higher-level system programs. WINDOWS-XP is an important operating system. There are two important differences in ROM-BIOS and WINDOWS-XP. ROM-BIOS provides services to users at very low-level, while WINDOWS-XP provides at higher-level, for example, file input and output. WINDOWS-XP does much services to users directly, which the ROM-BIOS does not.

PROBLEMS

1. Explain the terms: system software, operating system, firmware and utility programs.
2. What do you understand by low-level and high-level languages? Explain with examples.
3. Compare the merits and demerits of assembly languages and high-level languages.
4. Give examples of important high-level languages. Discuss their areas of applications.
5. Discuss the areas of applications of BASIC, FORTRAN, COBOL, PROLOG, LISP, SNOBOL and APT.
6. Describe assembler, compiler and interpreter. What are one-pass and two-pass assemblers, resident and cross-assemblers?
7. What do you understand by stack and stacktop? Explain how data are pushed to the stack or popped from the stack. What is stack pointer?
8. Discuss subroutines and macros. What are the differences between subroutines and macros? Mention the areas of applications of subroutines and macros.
9. What are SQL, DDL, DML and DCL? Discuss their applications.
10. What do you understand by the term “debugging”? Explain single-step control and break point method of debugging.
11. Discuss modular programming and structured programming, top-down and bottom-up design techniques.
12. What are the common and special symbols for flow charts? Discuss the application of flowchart.
13. Discuss multiprogramming, multiuser (time-share) and multitasking system.
14. Explain protection and semaphore.
15. What are object-oriented languages and procedure-oriented languages ? Give some examples of these languages.
16. Give examples of single user and multiuser operating systems.
17. What is WINDOWS-XP? Explain.
18. Give the names of important operating systems. Explain the main features of WINDOWS-2000 and WINDOWS-2003.
19. What are UNIX and LINUX? Discuss their main features and area of applications.
20. What is booting ?
21. What are utility programs ?
22. What is the function of a file manager, loader, linker and locator?
23. Discuss the function of an editor.
24. What is ROM-BIOS ? In what way does it differ from an operating system?
25. What is the difference between a programming language and an application package ?
26. What are application packages ? Give examples of some important application packages. Discuss their field of applications.
27. Describe the main features of word processing package.
28. Can a word processor program check spelling of all words used in the text? Can it check syntax errors ?
29. Describe the important features of spreadsheet packages.
30. What is MS-Exel? Discuss its important features. Is it a single function application package or an integrated package?

31. What is database management package ? Discuss its main features and area of applications.
32. What is MS-Access ? Discuss its main features and area of applications.
33. Describe the main features of graphics packages. What types of work are performed with graphics packages ?
34. What do you understand by integrated packages ? What types of work can be performed by this type of packages ? Give some examples of integrated packages.
35. What is MS-Office? Discuss application programs it consists of.
36. What is mail-merge feature of a word processing package ?

**THIS PAGE
IS BLANK**

9

CHAPTER

COMPUTER APPLICATIONS

Computer is the most powerful tool man has ever created. It made great impact on every activity of mankind and plays an important role in daily functioning of all industrialized societies. Earlier, computers were used for scientific and engineering computations. But now about 75% of work done by computer is of non-computational nature. Nowadays they are used for office work, games, for preparing manuscripts of books, automatic control in factory, ticket issue and reservation in railways and other places, control of military equipment; to store, display, send and receive information; to draw drawings, graphs and pictures; to control missiles and to control airplanes; to diagnose diseases and prescribe medicines in hospitals, to perform clinical tests, to maintain and sort the records of clients; to teach students; to perform accounting and financial work, tabulation of results; to prepare horoscopes, to select life partner; to supervise and watch certain areas; help police in crime investigation; to control robots; and so on.

In this chapter some important applications of computers are discussed.

9.1 COMPUTER APPLICATIONS IN OFFICES

Important work performed in offices are: preparation of letters, reports, memorandum, copy of advertisement, publicity, contracts, forms, notes, etc. Today these tasks are efficiently performed by computers. Nowadays desktop computers are available at reasonable cost. Even small offices, organizations and institutions can afford to have a computer. A personal computer costs in the range of Rs. 25000 – Rs. 50,000. A computer helps user in preparing, storing, retrieving and displaying text. As already explained in Chapter 8 software for this purpose called wordprocessing package is available. It helps users write, edit, manipulate, store, retrieve, display, and print the text.

The term electronic office is used when all work of the office is done employing computers. In an electronic office computers are used for the following tasks:

- (i) to prepare, store and retrieve files.
- (ii) to handle correspondence and office communication.
- (iii) to facilitate administrative work and preparation of related papers.
- (iv) to assist decision making.

The term office automation is used when computers are extensively used for office work.

The computer with word processing programs has many advantages over typewriters. In computerised system errors can be corrected very easily. Addition of text can be made to the document. When additions are made the contents after the addition are automatically pushed down (or shifted) as much as necessary. Similarly, any word, sentence, paragraph or page can be deleted from the document. As the word processor has storing capacity, the document can be stored for future need or reference. If and when desired changes can be made in the document. All these things are done using the editing capability of the word processing programs which helps users insert, delete and modify the text. It also provides the capability to move a word, sentence or paragraph from one place to another.

Word processing programs have many other powerful commands and facilities such as searching, mail merging, spelling checking etc. Suppose someone makes mistakes in spelling or he wants to change some word with some other word; for such purposes the search and replace command is used. This command searches a particular word throughout the document and replaces it with another word supplied by the user. A word processor program includes spelling checker which checks the spelling of all the words of the document. A dictionary is provided with the word processor. The words of the document are compared with the words available in the dictionary. If a word of the document is not found in the dictionary, it is identified. The user can add some unusual or technical words to the dictionary.

A syntax checker is also included. It checks syntax error, poor uses of words and other related items. Using mail merge command some information (data) can be included into the document (which is being processed by the word processor) from other files or database. There is also a facility to get synonyms or opposite words corresponding to a particular word if user so desires. Then he can select the proper word to use at certain place where he is unable to use the suitable word himself. Some word processing programs provide graphics facility also. In advertising materials, financial reports and brochures graphs can be included. Bar graph and pie charts can be included in financial reports.

There is also a facility to provide special mathematical symbols, notations, underline, bold letters, suffix, italics etc. in the document. Word-processing packages have powerful formatting commands. Once the text has been entered, it can be formatted as desired before printing the text. Line spacing, left margin, right margin, right justification etc. can easily be specified before printing the text.

In offices accounting, billing, preparation of pay rolls, data analysis etc. are also done using computers. These are discussed under separate headings.

9.2 USE OF COMPUTERS IN BOOKS PUBLICATION

Nowadays conventional letter composition in press for book publishing has become obsolete. The manuscripts of books are now prepared using computers. These computers contain word processing programs. Text of the manuscript is entered into a computer. Entered text is displayed on the CRT screen. Text is stored on an optical disk. Usually, laser printers are connected to the computer which is used for text processing. Laser printers produce high quality print. Prints of the text are taken out. First proof reading is done in the press. The

stored text is retrieved and displayed on the screen. The programmer makes correction as pointed out by the proof reader. Again prints of the corrected text are taken out. Finally, these prints are sent to the author of the book for final proof reading. Errors pointed out by the author are corrected again. Then final prints are taken out. Nowadays sketches, illustrations, diagrams etc. are also prepared on computers using suitable software.

For mass printing, a negative of the document is produced and it is used to prepare plates for offset printing.

Nowadays thesis, project reports etc. are also produced using word processing package. In case of thesis, project reports, etc. 4 – 6 copies are taken out using laser printers. Alternatively, only one copy of the text is printed out using laser printer and rest of the copies are produced by photocopying.

9.3 DESKTOP PUBLISHING SYSTEM

A desktop publishing package is more powerful than word processor package. The system includes computer and a number of peripherals with powerful software that can produce page layouts complete with pictures and text printed in a variety of attractive ways. Such pages are used in manuals, bulletins, newsletters etc. They are more attractive and efficient compared to the simple clear print outputs of word processors. It contains an art library containing over one thousand pictures which can be used by the programmer in his documents. The system contains an output device which can produce text and pictures. Digitizers or scanners are used to convert art, photo and text images into suitable signals to be fed into processors. The screen used is a high resolution screen. Though the package may run on PC, powerful computers are generally used. Phototypesetter is used which gives better output than dot-matrix or laser printers. For inferior quality work dot-matrix printer or letter quality printer can be used with desktop publishing package. But these devices do not produce pictures having quality as good as those produced by phototypesetter. The desk top publishing package can accept the text which has already been prepared by word processing package and stored in a file. Examples of desktop publishing packages (DTPs) are : Ventura, Page Maker, Quark XPress, MS-Publisher 2000, Akruti Publisher etc.

The desktop publishing package includes a program called page layout (or page make up or page composition) program which permits operators to format pages of the text and merge text and pictures on display screen. The operator designs the pages taking suitable pictures from the picture library. He combines pictures and texts and views on the screen. He tries a number of alternatives, then he selects the final design of the page layout. The page layout editor allows him to arrange the text around the picture as he likes. He can add pictures, can move the position of the picture, can change the style and size of headlines, design of borders etc.

When the design is completed, text is entered, the output of the desktop publishing package, i.e. text and images are sent to graphics printer (an output device used with desktop publishing unit) for multiple copies. Alternatively, phototypesetting equipment can be used for output, or a negative can be prepared. The negative is sent to printing firms for mass production.

9.4 APPLICATION OF COMPUTERS FOR DATA ANALYSIS

Computers are widely used for data analysis. There is a special software called spreadsheet program for data analysis. Data are entered into rows and columns. Using spreadsheet program one can tabulate results, calculate number of students in first class, second class, failed student, percentage of students in different classes, list of failure students, average marks etc.

One can prepare product sales, profit, investment computation for an organization using spreadsheet programs. Yearwise numbers of unit sold, price per unit, production cost, selling cost etc. are entered into the spreadsheet. Revenue are calculated from the units sold and per unit price. Total expenditure is the sum of production cost and the selling cost. Profit is equal to production cost minus total expenditure.

Similarly, any kind of computation for data entered in the tabulated form can be made using computers with spreadsheet program.

Computers are widely used for statistical data analysis. Statistical packages are available for such purposes. They have functions to calculate average, maximum, minimum etc. They can accept data from other files and sort, merge and manipulate data as needed.

Spreadsheet templates are available for various kinds of data analysis. The user has only to enter data and get the desired results. Details of spreadsheet package were discussed in Chapter 8. Lotus 1-2-3 and MS-Excel are popular spreadsheet package used for data analysis.

The spreadsheet programs are also used for accounting, sale analysis, inventory control, financial aspect of business, preparation of budgets etc. These items are discussed under separate heading.

9.5 ACCOUNTING, INVESTMENT, INVENTORY CONTROL ETC.

A spreadsheet program is a general tool for the analysis of accounting, investment, inventory control, preparation of budget, students grade sheet etc. But special purpose programs are also available for financial calculation, accounting etc.

Accounting

Specialized programs are available to handle accountant's ledger. The task of maintaining ledger becomes easy when computers are used for this purpose with special accounting packages. Other accounting routines are order entry, billing, account receivable, account payable, inventory control, sales analysis, tax calculation, pay roll preparation, auditing, budget analysis etc. Accounting packages are available for one or more jobs stated above. Integrated packages are also available to handle general ledger, order processing, inventory control, pay roll calculation, accounts receivable, accounts payable etc. Small organizations use integrated packages. Large firms prefer separate specialized packages for each type of accounting work.

Investment Analysis

Computers are used for investment analysis. Many companies issue shares, debentures etc. A large number of persons nowadays make investment in shares. Some people take decision on the advice of brokers or financial managers. But others like to analyze themselves using computers. Software packages for such analysis are available. The user has to simply

enter data and get the results of analysis. To handle this type of work software packages are available for fundamental analysis, technical analysis, portfolio management and personal financial management.

Fundamental Analysis Packages

Such a package analyzes the present financial position, prospect etc. of a company which issues securities. It analyzes its increase in sales in previous years, its present rate of securities etc. A person can collect information regarding different companies from newspapers, magazines, companies annual reports and databases available for such purposes. Then he makes screening to select a company on certain basis what he thinks right in his interest. He makes analysis on computers and then decides to purchase securities of certain companies. For screening purposes stock-screening packages are also available.

Technical Analysis Packages

Such a package helps users decide the best time to purchase and sell securities. It is based on changes in prices of shares, trend of price rise, trading volumes of the company etc. Information is collected from database, newspapers, magazines and annual reports of companies. It helps users in purchasing or selling securities.

Portfolio Management Packages

Such a package makes analysis of the present status of securities owned by a person. One has to supply the number of shares he is having, the purchase rate, current price etc. The program will calculate gain or loss, total investment in purchasing the shares, total current value, also these values companywise, total gain or loss, percentage gain or loss etc.

Personal Financial Management

Nowadays people use computers for keeping record and making analysis of their investment, income, expenditures, and savings. A personal financial management package is available for this type of work. It helps in preparing budget, calculating income, interest, insurance, printing cheques to make payment of bills, income tax calculation etc. It also calculates savings. The person can decide how much of the saving can be invested in securities etc.

9.6 APPLICATION OF COMPUTERS IN GRAPHICS

Computers are being used to draw drawings, graphics and pictures. One need not depend on draftsman or artist for these purposes. For this work graphic packages are available. Preparation of drawings is a time consuming task if it is done by a draftsman. But computers can draw sophisticated drawings in multicolour in minutes. It can produce three-dimensional pictorial views. Graphics packages are often integrated with spreadsheet, investment analysis and statistical analysis. Graphics such as bar graphs, pie charts and line graphs are plotted from spreadsheet data and other tabulated form of stored data. Graphics packages are also used to recognize picture, process images, draw pictures etc.

Drawings or pictures can also be modified. They can be reduced, enlarged or rotated while displaying on the screen. When specifications and dimensions are given the computer will draw three-dimensional views or two-dimensional drawings. Moving pictures can be presented and analyzed. Drawings from different angles can be presented to designers. For example, a designer will like to see the different views of buildings, aircraft etc. All such facilities are provided by graphics packages.

Graphics packages in different categories are available, for example, design, paint, analysis and presentation packages.

Corel Draw

It is a package to prepare illustration and page layout. It has been developed by Corel Corporation. Its version 9 has improved interface, a new palette, tools for Web and Acrobat (standard for electronic document exchange) output, better workflow and powerful production capabilities. Corel DRAW-9 Premium Colour Edition includes Corel Photo-Paint-9, Microsoft Visual Basic for Applications version 6 for scripting functionality and Bitstream Font Navigator—a leading font management system.

Paint Packages

A paint package is used to create drawings or pictures. It is used by artists. The package also provides art library of stored images. The artist can select picture from the library. Then he can select colours of his choice and can do painting work. The package allows to rotate or flip a shape, add titles or captions in a style as he likes. He is allowed to make changes in scales. There is a zoom command which is used to increase or decrease the scale of a viewed object. The paint package permits to zoom in on part of the emerging drawing to refine image details. When the picture is completed it is printed and stored.

Analysis Packages

An analysis package presents masses of data in graphical or picture form which gives a better concept of the relationship, changes and trends that are hidden in their data. Bar charts, pie charts and line charts are widely used by analysis. The package allows users to vary colours, add and delete lines and headings. It also allows to change scales and edit the look to the graphs. The results can be printed and stored for future use.

Presentation Packages

A presentation package is used to communicate information to people. It includes all features of an analysis package. Bar and pie charts are used by this package also. But it is capable of producing multiple three-dimensional images. It permits to present charts in such a way that appeals to audience. For example, one can explode a pie chart segment or separate a segment from other segments to draw the attention of the persons to an important point. The features available in paint packages may be included in a presentation package to permit programmers to dress up charts with art library pictures or special drawings.

A presentation package utilizes colours in a better way as compared to an analysis package. It provides a number of styles to draw graphs. It is also provided with animation facility. For example, it can show bars growing, pictures moving etc. on the screen. It requires a powerful computer to show its ability, but PC presentation packages are also available.

Design Packages

In the field of engineering two-dimensional (2-D) and three-dimensional (3-D) drawings, pictorial views and pictures are widely used. Computers are extensively used for these purposes. When a design engineer designs a new machine or modifies the design of an existing machine, he prepares its drawing. The drawing specifies all dimensions. It shows its various parts. The engineering drawings are very complex and sophisticated. Draftsman using hand technique takes much time. A computer offers a powerful, quick and convenient tool.

It helps designer create, view on the screen and modify the drawing. Then prints are taken out and stored for future use.

CAD, computer aided drafting (or computer aided design) is a software package which helps design engineers and architects to design and prepare drawings. CAD is widely used to prepare machine drawings, building drawings, presentation of solid model (3-D views) of machine parts on the screen. It also shows dimensions. The output of CAD helps other engineers in manufacturing the machines, erecting buildings, etc. using drawings, specifications and dimensions. CAD is also used to draw electric and electronic circuit diagrams, to design printed circuit board layouts, LSI/VLSI chips, aircraft structure; bridge, tower structures; chemical plants piping layout etc. AutoCAD is a popular CAD package used for preparing drawings.

9.6.1 Graphics Tools

There are a number of input devices which are used in graphics applications, for example: graphic tablet, mouse, joystick, light pen, etc. Some of these devices have already been discussed in Chapter 7 and some others are discussed here. The input devices which convert graphic and pictorial data into digital form are called digitizers. The digital data are sent to the computer for processing and storage in the memory. An example of digitizer is a graphic tablet.

Graphic Tablet

It is an input device to create graphics. It is commonly used at CAD terminals, engineering workstations and other data entry sites. It is a flat surface which can be kept on a table as shown in Fig. 9.1. It contains hundreds of copper wires embedded below the surface of the tablet. These wires form a grid. The grid is connected to a computer. A *stylus* (an electronic pen) is provided with the tablet. When the stylus is pressed down at a point on the tablet, a switch is closed and the tip of the stylus is electromagnetically coupled to the grid wires. When the stylus is moved on the surface of the tablet, there is a corresponding movement of the cursor on the CRT screen. The grid wires detect the co-ordinates of a point where stylus is positioned. The X-Y co-ordinates of the points are transferred to the computer. The graphic tablet digitizes a drawing or graphic while it is created. A graphic tablet is also called digitizer tablet.

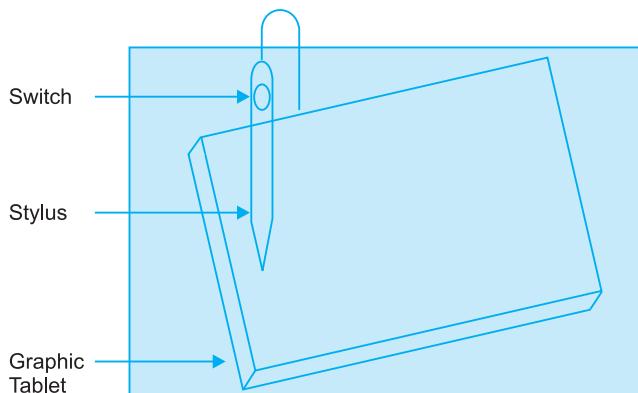


Fig. 9.1 Graphic tablet

The stylus is not used to mark directly on the tablet. Rather, a piece of paper is placed on the tablet and drawings are made on the paper sheet. The drawing is displayed on the screen. There is a difference between the actually drawn sketches and displayed drawings. Poorly sketched lines are displayed straight, poor lettering is replaced by neat printing and poorly drawn corners are made smooth curves. Changes and modifications in the drawings can be made quickly.

Image-Scan Digitizers

This type of digitizers can scan and reproduce entire drawings or photographs automatically. They are capable of distinguishing intensities of the spot on the photograph.

9.7 CAM (COMPUTER AIDED MANUFACTURE)

Computers are nowadays used to control production of certain engineering products, chemical products etc. Take an example of a mechanical industry. Designers use CAD and prepare the design of machines and machine parts to be manufactured. After the design is complete all data, drawings, dimensions are available. This information is available to engineers and operators who are the incharge of manufacturing. They use computers to control manufacturing machines such as lathes, drill, milling machines etc. These computers which are controlling manufacturing use design data produced by CAD.

In many industry drawings, dimensions etc. are not needed for manufacturing for example chemical industries. In such a situation information is transmitted from CAD system directly to product manufacturing equipment.

Integrated approach of CAD and CAM is also used. This is known CAD/CAM system. The designers design new machines using CAD. In integrated system there is a database which stores all data regarding design, drawings, specifications, etc. On the factory floor the manufacturing personal are on the same computer network. There are appropriate softwares to get design informations from the database and convert them automatically into other programs which activate and control manufacturing machines such as lathes, drills, milling machines etc. Further, they control assembly lines and so forth. A series of computer-controlled machines perform the whole task of production. So full automation for the manufacturing of products is achieved in a CAD/CAM system.

Another term CIM (Computer Integrated Manufacture) is used when all the work of a factory is done by computers, such as design, manufacturing, costing, sales management, inventory control, planning and other business and administrative functions. CIM is used for total automation.

9.8 APPLICATION OF COMPUTERS IN DATABASE MANAGEMENT

Nowadays computers are extensively used to store and retrieve information. Offices are using computers to handle files which store a variety of information. When needed, we get information from the files. The term database is used to refer to one or more related files. The files are now stored in magnetic memory. There is no need of paper to store information. Information can be seen on the computer screen, when required prints are taken out. Information from one office to another can be transmitted by electronic mail system.

To handle files database management packages are widely used. Suppose computer stores information in a file regarding students: name of the students, roll number, marks

subjectwise, total marks, percentage of total marks, class awarded etc. Now some one asks computer to print the list of students who have secured total marks 75% and above, one can ask to print the name of the students who have secured the highest marks in different subjects. Someone may provide roll number or name of the student and ask to print his marks sheet.

The database package not only stores information and supplies information, but also makes computations on data to supply certain information. Actually, in the above example, the total marks, percentage of marks, average etc. are calculated and then stored in tabulated form by the computer. Class is awarded by the computer.

A database management system can provide information from more than one related files. A user would like to have certain information which is not in a single file. Some information is in one file, some in other files. The program will search all the required information from different files and supply it to users. Take an example of records maintained by an advocate. He maintains two files. One file contains the records of different clients. It may be named as client file. It records client number, client identification number, name of the client, address, phone number, amount due etc. He maintains another file which keeps the records of different cases. It may be named as case file. It records case number, identification number of the cases, identification number of the client of the different cases, type of the case, name of the opponent, attorney, court, judge, opening date, court date etc. Suppose the advocate wants information about which cases are to be taken up on a particular date, with the details of the cases and details of clients. To obtain information one may have to search both the files—the client file as well as the case file. The software package will collect information from both the files. The case file gives the information which cases are due on a particular date and their details. It also gives the identification number of the client corresponding each case. For these identification numbers the details of clients are obtained from the client file.

Examples of database management packages are: ORACLE, SYBASE, INGRESS, MS-Access, INFORMIX, FOXBASE etc.

There may be a number of examples of the utilization of database packages. Some examples are given under other headings in subsequent sections.

9.9 APPLICATION OF COMPUTERS IN PROJECT MANAGEMENT

Various organizations and departments take up new project work, it may be to install a new factory, build a new building, construct a power plant, open a new college, a hospital and so on. The project work has a series of tasks which must be completed in scheduled time period. Nowadays computers can be used to manage project work. Project management packages are available for such purposes. A project management package is used to plan, schedule and control man power, material and other resources needed to complete the project work in the scheduled time period at reasonable cost. For computer analysis the information required is: what are the tasks to be performed, the sequence in which tasks are to be performed, time required to complete each task and the starting data. All these informations should be supplied to the computer. Certain tasks can not be started until certain other tasks are completed. This type of information is also to be supplied. The project management package shows the longest sequence of events in the project called *critical path*. It calculates total project time. It also checks for dangless (the task which do not follow the schedule i.e. either lag or lead or create some other type of mismatching problem).

In case a project lags behind the schedule, speed can be increased to reduce the time of completion of one or more tasks on the critical path to catch up the schedule. Project management package will show the impact of the delay. It will quickly show the modified schedule. Some packages show an automatic resource leveling feature. It will show the increased manpower required to catch up the schedule. Most project management packages employ critical path method but some packages provide best, worst and most likely estimates for task durations.

9.10 USE OF COMPUTERS IN COMMUNICATIONS

Computers are extensively used in sending and receiving information. The information transfer may take place either between two points or throughout a computer network that connects a number of computers or workstations. Data communication packages are available which are used to send and receive information. For communication purposes one must have a computer, a data communication package and a modem. Usually, telephone lines are used over which data are transmitted from one place to another. The telephone lines have been built for voice transmission. To transmit human voice continuously variable analog wave patterns are used. Thus a telephone line is designed to transmit analog signals. On the other hand, a computer processes digital signals consisting of pulse or no pulse, i.e., presence or absence of an electrical voltage. Digital signals can not be transmitted over existing telephone lines which have been designed to transmit analog signals. Hence, modems are needed to convert or modulate digital signals to analog signals and vice versa when existing telephone lines are to be used. A modem connects a computer to the telephone lines. The modem receives data from a computer or a terminal which are to be transmitted over telephone lines. The modem converts (modulates) the digital signals into analog signals which can be transmitted over the telephone lines. The modem at the other end receives the data from telephone lines and converts (or demodulates) the received analog signals into digital signals. These digital signals are processed by a computer at the receiving end and they are reproduced as the original information. Data can be transmitted in both directions. A modem performs both tasks: modulation and demodulation. When data is being transmitted from an end, the modem placed at that end performs modulation. When data is received at an end the modem placed at this end performs demodulation.

From physical design consideration the modems are of two types: external modem (or stand-alone or freestanding modem) and internal modem. An external modem has its own cabinet. An internal modem is installed inside a PC. It is cheaper than external modem. It does not require additional desk space. An external modem is recommended only in situations where computers are likely to be changed.

Both types of modems are equipped with microprocessors, memory chips and special communication chips. Such modems are intelligent modems. They can be programmed to automatically perform dialing, answering and disconnecting tasks.

Communication lines are being designed and built in USA, Japan and most of European countries to transmit digital signals directly. Such systems do not need modems because they do not use analog signals. Until such networks are fully developed, existing telephone lines will continue currently to be in use for data transmission.

Computer communications are being used to send and receive electronic mail, engage in online conferencing (computer conferencing, teleconferencing or videoconferencing); to

view and post news on computer bulletin boards; for electronic shopping, banking and brokerage and to get information services etc. These information utilities fall in the following three categories:

- (i) Communication services
- (ii) Transaction services
- (iii) Information services

The communication services include electronic mail, online conferencing, to view and post news on computer bulletin boards etc. The transaction services include electronic shopping, banking and brokerage services. The information services include online information given by some organizations which maintain database for such purposes. A customer can access the database to get the desired information. Information may be of general nature or deal with some specialized topic or business.

9.10.1 Electronic Mail (E-Mail)

When messages are delivered and received employing computers it is called electronic mail. In E-mail it is not necessary to locate the person who had to receive the message. The persons who are connected through this system have an individual mailbox to store messages to be received. The messages are sent on the first try. The messages may be either spoken or keyed text. They may be sent at any time, day or night. The receiver need not be interrupted or disturbed at odd time. He will link up with the system periodically at his convenience to review the stored messages. The message can be sent to an individual or a number of persons. The keyed messages are usually entered and received on PCs, teleprinters or visual display terminals. Apnamail.com has been developed to send messages in Indian languages.

Documents and pictures can also be transmitted. A facsimile machine (or fax machine) is used to send or receive pictures or documents. At the sending end the fax machine scans the document and at the receiving end a fax machine reproduces the scanned image. The receiving machine produces duplicate or facsimile of the original.

9.10.2 Bulletin Board Systems (BBS)

The appropriate software program permits bulletin board computer to answer telephone calls from user's computers, accept and store messages from outside users and allow access to the information stored in its files. Users can actively exchange ideas and participate in ongoing discussions or passively observe the electronic conversation of others. This type of service is offered to a group of users who have similar interests. For example, bulletin may be for a group of employees, customers, clients, patients, readers, students and so on. Life insurance company may provide BBS to give general information and answers to the questions of policy holders or who want to have new policies. Health organizations may have BBS for doctors. Industries can provide BBS for their employees and so on.

9.10.3 Computer Conferencing (Teleconferencing or Videoconferencing)

The videoconferencing allows persons sitting at different places to participate in a conference. It is also known as online conferencing. The participants are at their PCs or terminals. This allows people to meet at a common time and communicate over wide distances. It allows users to have voice, video and data communication.

9.10.3.1 Instant Messaging

An user can have a conversation with somebody sitting anywhere in the world in real-time using instant messenger. It is a Web-based communication means. It allows users to do text based chatting in real-time. Its capability can be extended to include voice chats or video conferencing. In the public domain, IMs (Instant Messengers) are provided by Yahoo, MSN and ICQ. They are available for download and are very easy to use. In the case of public IMs, there is problem of control and security. Big organizations may have their own IM server for their work.

9.10.3.2 Groupware

It provides more facilities than **collaboration**. In collaboration, user can share calendars, tasks and similar objects which can help them work together on a project more efficiently. **Groupware** is a platform which allows groups to work on projects, providing all resource and tools for all activities required for completing a project. It provides three categories of functions: communication, conferencing and central repository. A Website is created to provide all types of information. Audio, video and data conferencing can be done. Conferencing also provides facilities to share information in an interactive way. Groupware has facility of instant messaging. Chat rooms may be provided for real-time collaboration among the people. The central repository is the location where all information is stored. The information include what people create, use and modify. The repository provides all necessary tools to collect, organize, manage and track information. Different types of data required for the project can be stored in the central repository. The data may be about project information, tasks, meetings, documents, calendars, responsibilities etc.

9.10.4 Computer Information Services

Many organizations and individuals develop and maintain databases for specific subjects. They allow subscribers to use their databases. They provide online services. They provide subscribers a communication package which allows to access the database. The subscriber has to give his identification number and password, and then he can access the stored information.

Now-a-days Internet has made information service very easy. One can get almost all kinds of informations on Internet in every sphere of life, such as arts, science, medicine, engineering, agriculture, business, industry and so on. Some typical examples are:

WWW.indiandoctors.com, which allows exchange of technical and clinical data between doctors, with research institutions or hospitals.

WWW.krishiworld.com, which includes comprehensive information on a wide range of topics: soil, diseases, fertilisers, weather, agro products and crops.

WWW.architectmatters.com, which is for construction industry in India. Suppliers and manufacturers can list their services and products.

The databases may be basically of two types: one type includes specialised, well defined systems to serve the needs of a particular section of people. The other type may contain multipurpose information utilities. The specialised services may be for doctors, engineers, lawyers and so on. General purpose information services may be for general people. Some examples of general purpose information services are given below.

CompuServe

It is a general purpose information service, based in Columbus, Ohio, with more than 200,000 subscribers. It provides two types of services: consumer information service and executive information service. Under consumer information service customers get the facilities of electronic mail, home shopping via electronic mail, travel arrangements through TWA's reservation system, and national bulletin board service. In home shopping the customer makes the selection of goods from catalogs or newspaper advertisements. He does not enjoy the pleasure of window shopping. Some systems allow viewers to see items on the television screen. The viewer can rotate items displayed on the screen or view them in closeup.

Under executive information service, besides all above mentioned services, other services are: access to financial, demographic and editorial information, stock quotes, market and industry indexes etc.

The Source

It is a general purpose information service. It is based in McLean, Virginia. It provides news, weather and sports information, business and investment data (including online stock trading) and electronic shopping.

9.10.5 Home Banking

A person sitting at home can perform all bank transactions. For this type of work a PC or special input device attached to TV set can be employed. A communication package is used for the purpose. Information is transmitted over telephone lines. One can see his balance, make payment of bills, apply for loans, and deposit money. He can make payment for purchasing securities and can perform other kinds of bank transactions.

9.10.6 Online Investment Brokerage System

One can use computers for online investment brokerage which will connect user to investment brokerage houses. He can see the present rates of securities and can place order to purchase or sell securities.

9.10.7 Office-Work Sitting at Home

Using computers one can do the office-work sitting at home. He will connect himself to the office and will take instructions from his supervisor what tasks are to be performed. He will enter the text, prepare documents etc. into the computer. The documents will be sent to the office. It will be stored in his file. The officer-in-charge can see the work done by his junior or other staff on the screen and guide them for doing some modifications, improvement or assign other new tasks.

9.10.8 Telex and Teletex

World-wide standardised text communication is currently being offered by the telex service. At present millions of telex lines are in existence. Telex communication takes place between two teleprinters. In telex system an operator has to dial (or redial if number is engaged) to transfer the message. It is a slow system.

Today telex is facing challenge from teletex or electronic mail. In teletex the communication takes place between two word processors. In this system letters and documents can be transmitted via telecommunication lines from one word processor to another. The text can be transmitted and received in the form with which we are most familiar. It is a more

advanced public system for sending copy of a text providing a much wider character set that includes punctuation and lower and uppercase characters. Transmission rate is much higher. This system offers direct communication between two word processors. It has potential to out-perform the telex service. Its superiority lies in its rich character set and flexibility with regard to document layout, its error recovery features, more powerful memory based terminal system and higher speed.

9.10.9 Satellite Communications

To transmit large volume of data at high speed, over 100,000 CPS, broad channels are employed. Coaxial cables, microwave circuits and communication satellites are commonly used for broadband channels. To transmit data through space microwave channels are used. In microwave system very high frequency radio signals are employed. When microwave systems are used through ground routes repeater stations are installed at every 40 KM. The data signals are received, amplified and then transmitted onward at each repeater station. Alternatively, if a communication satellite is available, data can be routed through it. A satellite is the quickest and cheapest medium for transmission of voice, data, text and pictures over long distances. The data are sent to the satellite from a transmitting station on the earth. The satellite acts as a reflector. It accepts signals from one point on the earth and returns the same signals to some other point on the earth. If a satellite's speed matches the earth's rotation, it may look stationary from the earth. Several satellites are now in orbit to handle transmission of data in a certain zone or all over the world.

9.11 APPLICATION OF COMPUTERS IN EDUCATION

In educational institutions computers are used as teaching aid, information resource and computing and research tool. When computer is used as a teaching aid it is referred to as *computer-assisted instruction* (CAI). Software for CAI can be prepared by class teachers or software packages can be purchased. The subject material is displayed on CRT screen. The computer also asks questions to test the students. When a student gives correct answer, the computer gives some comments to indicate that answer is correct. If the answer is not correct, the computer gives some error signal. The CAI programs can be prepared in a variety of modes, such as tutorial, discovery, problem solving, modelling, and drill and practice mode. In *tutorial mode* the computer presents new ideas followed by test questions. In *discovery mode* the computer presents information and asks students to draw conclusions. In *problem solving mode* computer allows students to apply their concept in solving a problem. In *modelling mode* a process or a system is presented mathematically to make its analysis. Students are allowed to change variables to see their effects. In *drill and practice mode* it is assumed that the skill in question is known but repetition is essential to have command over the subject. The interactive feature of the computer makes computer-aided teaching attractive. Computers can repeat the presentation of the material as many times as desired. This helps students make certain concepts clear if they are not clear in earlier attempts. Other attractive features of computers are their graphics—particularly colour graphics ability. Pictures can move, rotate, computer can speak; all these features attract students. Graphics help students to see three-dimensional figures and view from different angles. Figures can be enlarged or reduced.

Computers are widely used for computation, design, and research. The word processing package is used for preparing text, typing thesis, office and administrative work, preparation

of test questions etc. Its vast storing capacity is utilized to store large volume of information. Data processing capability of computers are used by students, teachers and research scholars to analyze, manipulate and process data. Information stored in databases serve as a learning resource for students. Database may be purchased or developed by teachers and students. Remote databases can be accessed through computer network. Question banks may be prepared and stored in computer's memory. Test questions may also be stored so that students can manage their examination to test their knowledge. Teachers can use computers to maintain students grade, to analyze student performance, grade students, and to give home assignments etc.

General purpose languages such as BASIC and PASCAL which were used earlier by several programmers to prepare CAI software, are not well suited for CAI. An extended version of PASCAL, called UCSD has been developed for CAI software. It makes easy to process words, clauses, lines, curves and also student answers. LOGO is another language which is very much suitable for CAI programs. Its graphics ability enhances its utility for CAI programs. It allows to draw, colour, and animate pictures on the screen, which is interesting and attracting feature of graphics. LOGO is very easy to learn. It has been popularized as educational language. Children can learn it very easily. They can use it to achieve intellectual growth, and develop problem solving skills. It is also used to manage data, compose music and manipulate text. It can also perform sophisticated programming. It runs on PCs and is used in schools and colleges.

PILOT is another language used in CAI. It stands for programming Inquiry Learning or Teaching. It is a dialog-oriented language. Its syntax is simple. It handles words and text easily. It emphasizes drills, texts and dialogs.

9.12 APPLICATION OF COMPUTERS IN MEDICAL FIELD

Computers are widely used in hospitals to help doctors in diagnosis, getting information on patients, diseases, treatments, drugs etc. They are also used in administration and in keeping patient records. Doctors can get information from distant data banks and expert systems. They can discuss with colleagues using teleconferencing.

Many medical databases are being developed. They contain information on diseases and treatment. Doctors are taking help of expert systems in the diagnosis and treatment of a patient. An expert system is a program based on the expert knowledge of specialists. It is a diagnostic tool for doctors. The doctors have to supply the detailed information such as symptoms, medical history, test results etc. This information is processed by the computer and a diagnosis is made by the expert system. At present the expert systems are helping doctors; they do not substitute doctors. They are currently useful primarily in reminding doctors of diagnosis and treatment that should be considered. The databases in the expert systems must be constantly checked and updated. Some more details are given later on in this chapter.

Many devices take images and diagnose diseases. Such devices use computers to process huge amount of data collected by the scanners associated with such devices. Examples are: Computerized Axial Tomography (CAT), Positron Emission Tomography (PET), Magnetic Resonance Imaging (MRI) etc. These devices take pictures of brain and help in diagnosis of various kinds of diseases. These machines are at present very expensive.

Computers are also used in pathological tests. Programs have been developed to monitor various pathological processes, record data and analyze results. From a single sample of blood

computer can test hemoglobin, hematocrit, glucose, BUN, creatinine, total CO₂, chloride, sodium, potassium, calcium, phosphorous, uric acid, albumin, protein, globulin, A/C ratio, cholesterol, triglycerides, bilirubin, phosphate, white and red blood cell count etc. Computer controlled monitoring of tests increases speed, and give accurate and reliable results.

Microcontrollers are being used to control artificial limbs. Microcontroller controlled artificial arms and legs have been developed for disabled persons. For paralyzed persons, robot arms actuated by voice signals can handle eating utensils, turn pages of books and move certain objects. A person confined to bed can direct a robot by voice commands, and can watch the movement of the robot on a television or CRT screen. Microcontrollers will bring revolution in health care for disabled in near future.

Computers are also used in athletic training. Computers watch the movement, analyze positions at various moment and point out errors, weaknesses etc. if any. Coaches utilize computer to tabulate, process and analyze game statistics. Computers are widely used today in both judging and scoring athletic events.

9.13 APPLICATION OF COMPUTERS IN INDUSTRIAL CONTROL AND INSTRUMENTATION

Computers are extensively used in industry for automatic control of machines, process, measurement and display of electrical and physical quantities. Examples are: automatic control of temperature of a furnace, speed of a motor, generator's voltage and power, boiler's pressure, testing of products etc. Microprocessor-based system or single-chip microcomputers have been developed for industrial control and instrumentation. We shall take some typical examples how microprocessor-based systems are used in industrial control and instrumentation.

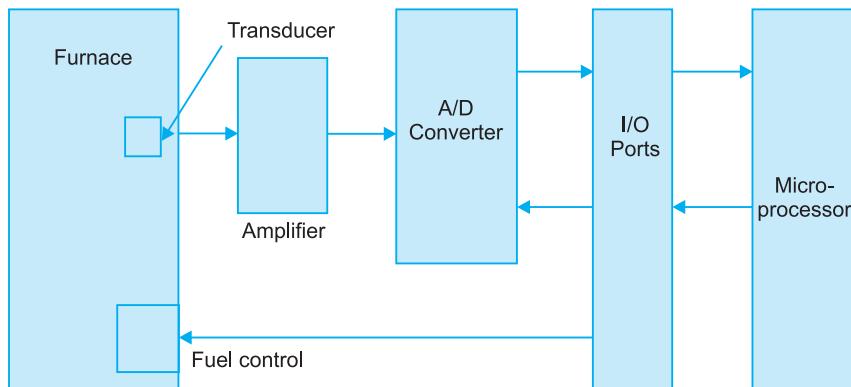


Fig. 9.2 Furnace temperature control

Fig. 9.2 shows the schematic diagram to measure and control the temperature of a furnace. A transducer is a device that gives an electrical signal proportional to a physical quantity. In this case a temperature transducer has been employed, which gives electrical voltage proportional to the temperature of the furnace. If electrical signal is small it is amplified by an amplifier. The amplified voltage signal is an analog quantity. As a computer can process only digital quantity, it is applied to an A/D (analog to digital) converter. A/D converter gives a digital signal proportional to the analog voltage input. This digital signal is processed by the microprocessor. Suppose, we want that temperature of the furnace should be maintained constant, say at T₁. A digital value proportional to T₁ is stored in the memory.

The microprocessor compares the measured digital voltage (which is proportional to the temperature of the furnace) with the reference value (*i.e.* T_1). If the measured temperature is less than T_1 , the microprocessor gives a signal to increase the fuel input to the furnace. If the measured temperature is more than T_1 , the microprocessor gives a control signal to reduce the fuel input to the furnace. The microprocessor also displays the measured temperature on LED display or CRT screen. The microprocessor checks the temperature at certain intervals and takes corrective measures if the temperature deviates from its normal value. As the microprocessor is very fast, it can measure and control the temperature of a number of furnaces in turn. For this purpose the transducers are connected to the A/D converter through an analog multiplexer. A multiplexer is a switch which has N inputs and it gives one output at a time. Under the control of a microprocessor any one of the inputs of the multiplexer can be made available at its output terminal for processing. Thus N number of signals can be controlled by the microprocessor in turn taking one at a time. To process and control one signal it takes time of the order of milliseconds.

A microprocessor-based system can measure and control any physical and electrical quantities, for example, pressure, displacement, deflection, force, strain, stress, voltage, current, power, energy, frequency, water level, etc. It controls the operation of any machine, device, equipment, instrument etc. Take an example of a power house. The microprocessor-based system measures and displays the pressure of boilers, temperature of furnaces, voltages of generators, power output of generators, frequency etc. It also controls these quantities. If any fault occurs on any part or the power house it immediately disconnects the faulty part or faulty generator or faulty transformer from the healthy part of the system. Similarly, a computerized system shows the diagram of the entire power system of a state or nation on CRT screen. It shows generating stations, connecting lines, transformers etc. It shows voltage, power outputs at various points. If there is any deviation from the normal values, it gives indication by flickering light at that point. It sends control commands to all power houses, substations and other important points.

All types of modern instruments contain embedded microprocessor-based system or microcontroller (single-chip microcomputer) for their operation and control. Also, computerized controllers for process, equipment or machines are microprocessor-based system or microcontrollers. Many of them are embedded in the equipment or machines to be controlled.

9.14 AIRLINES AND RAILWAYS TICKET RESERVATION

Indian Railways have introduced computerized ticket reservation system at many important stations. The computer stores all necessary information in connection with ticket reservation, such as train number, stations enroute, distance, number of seats available for each train in each class, fare etc. It also stores information regarding persons who have already reservation such as name of the passengers, age, destination, date of journey, class, fare, etc. The person sitting at the counter gets information immediately whether seats on a particular date in a particular train are available or not. If seats are available, computer prints and issues ticket and the details of the person getting reservation are recorded. The computer also prints reservation charts. It also gives information regarding train timings, train fares etc. on enquiry.

Computers are also used for plane ticket reservation. The passengers can also book ticket for onward journey if computers are connected to a computer network for such a

purpose. They can also get informations about seats in hotels in other cities, climate of other cities etc. through the computer network.

9.15 ROBOTS

Robots are computer-controlled programmable machines composed of mechanical hands, legs, and optical sensors to provide vision. A robot contains one or more microprocessors to control its actions. It is designed to move materials, parts, tools, or to perform certain specialized tasks. In industries they are used to perform tasks such as spray painting, welding, stamping, packing cartons etc. Robots are also used as domestic servants. They are used in hospitals to perform many tasks which are presently performed by nurses. In industry they can perform tasks in the environment where human being cannot, for example, robot can work at places where temperature is too high, they can handle toxic chemicals which cannot be handled by human beings, they can perform tasks at places where nuclear radiations are high enough to cause damage to human being, can handle radioactive substances, and so on. They can perform tasks in dirty drear, unpleasant environments. They can perform very hard task, can handle heavy loads and can place certain metallic parts in hearth for tempering and then remove them while they are white hot.

Robots can be programmed to perform different jobs at different time. A robot can make a series of spot welding while manufacturing a particular design of a car. It can be reprogrammed to change the welding pattern for other models of the car. When there is no welding job, the robot can be programmed for other job such as material handling. In future more sophisticated robots are expected. They may include artificial intelligence, and hence they can take certain decisions.

Some robots with vision can select and sort parts from a random mix. Some others can lift and position parts on an assembly line. They can also check the quality of assembled products. Still other robots can recognize, understand, insert and align various parts while making assembly of certain items. Robots may be provided with artificial intelligence along with sensory capabilities such as hearing, sense of touch and vision system.

9.16 ARTIFICIAL INTELLIGENCE

A machine with reasoning, learning and logic capability is said to possess artificial intelligence. For example, expert systems possess artificial intelligence. An expert system is a software which utilizes a knowledge base collected from human specialists in a certain area, and helps users refer to this expertise in making their own decisions.

An artificial intelligence system has a knowledge base and programming technique to probe and process the facts in the knowledge base. The development of a knowledge base is very difficult and time consuming job. To develop a knowledge base the designers and research workers consult experts of a particular field. They collect facts that are based on the knowledge and experience of the experts in that field. Such a collection forms a knowledge base. An expert system uses a knowledge base for its artificial intelligence. Besides expert's knowledge an expert system also uses the decision rules of human specialists to reach certain conclusion and to give recommendations. Natural language may be used to input the user's facts and queries. The recommendation made by an expert system is based on the collection of information in the knowledge base. Hence, the knowledge base must be updated regularly. An expert system selects one of many opinions depending on input data and information. The

machine uses a set of decision rules on logical basis to choose an opinion. The machine is designed to perform such logical functions. It asks users to supply information on a problem. Computer then processes the information and reaches conclusion making use of the knowledge base.

Due to various reasons many organizations want to develop expert systems. If an expert retires, dies or resigns the organization loses his expertise. So this is a very good reason to preserve expertise of experienced specialists. Another reason is that other person are benefitted by the specialist's expertise. Some examples of expert systems are as follows.

Several expert systems in various specialized fields have been developed and are in use. Important specialized fields in which expert systems are in use are: medicine, engineering, chemistry, biology, defence, oil industry, mining, geology, training, computer science and mathematics.

A number of medical diagnostic expert systems have been developed to diagnose diseases and recommend treatment. An expert system called HELP has been developed to diagnose heart diseases. Another expert package is PUFF which has been designed to diagnose lungs diseases. The expert system CADUCEUS is a generalized package. It can process and evaluate over 4000 symptoms and other information to diagnose over 600 diseases. The expert systems in the medical field at present do not replace doctors. They are currently helping in reminding doctors of diagnosis which should be considered.

An expert system called DELTA has been developed to detect malfunctioning of diesel electric locomotives. It helps maintenance staff correct malfunctioning. DENTRAL is an expert system developed by Stanford University to estimate the molecular structure of unknown compounds by analyzing mass spectrographic, nuclear magnetic resonance and other related data.

Some more expert systems are listed in Table 9.1.

Table 9.1 List of Some Important Expert Systems

Name of the Expert System	Field of Application	Function Performed
MYCIN	Medicine	Diagnose meningitis and bacterial infections.
ELF	Oil Industry	Oil exploration
PROSPECTOR	Geology	Mineral exploration
EL	Electrical Industry	Analyzes electrical circuits
MOLGEN	Biology	Planning DNA elements and genetic science
XCON	Computer Science	Diagnosis of Computer malfunction
SOPHIE	Electronics	Computer-aided instruction
STREAMER	Training	Teaches naval officers. It has been developed by U.S. Personal Research and Development Centre.

9.17 MILITARY APPLICATION OF COMPUTERS

Computers are extensively used in defence. Nowadays smart weapons are being widely used in military operations. Smart weapons use microprocessor-based control systems. Modern tanks, torpedos, missiles etc. employ computerized control systems. A guided missile uses internal computer for its control. Computer's memory holds the detailed map of the target area. Photographs of the target are taken and compared with stored map before the missile hits the target. Another advantage of computer guided missiles is that they fly faster and turn more quickly as compared to manned aircraft. They can fly into dangerous zones without the risk of pilot's life. Computers play very important role in military communications. Radar systems employ computers for processing data and display images. Computerized control system provide much more information about range, height etc. of the target. In future it is expected that computers will present three-dimensional picture of the target. The computer will check the profile and indicate what the profile represents. Nowadays aircraft employ computers for their controls and instrumentation. Fighters are equipped with computers for their quick activity. The X-29A, a fighter has three computers which make about 40 adjustments per second. The computers also allow pilot to interact with.

9.18 APPLICATION OF COMPUTERS IN BANKS

Banks use computers for general purpose computations, to maintain ledger, to handle transactions, to make entry in passbooks, to issue fixed deposit receipts, to provide online service to customers who want to perform bank transactions from home terminals and to answer customers at bank terminals regarding their balance etc. At bank counters computer allows customers to make withdrawals and deposits. The customer supplies his password and makes a few key strokes to give other details. To handle cash banks use cash dispensing machines. Such a machine is a microprocessor-based machine. This machine provides cash when identification number is typed in. Money can also be put into an account in the same way. The large volumes of cheques received everyday in banks nowadays are processed by computers. To handle this task magnetic-ink character reader is employed with computers. This has already been explained in Sec. 7.1.6.

With the use of a computer electronic transfer of funds from one account to another has become possible. A customer from his home, office or store can insert his identification number and type the transaction details on the terminal's keyboard. Immediately the amount he specifies will move out of his account and enter another's account.

9.19 APPLICATION OF COMPUTERS IN DESIGN AND RESEARCH WORK

Scientific and engineering design and research work involve complex and massive computations. In many cases simulation is also required. Design of bridges, towers, buildings, generators, motors, electrical transmission network, statistical analysis of molecular structure of complex proteins etc. need complex computations. In some cases massive data are to be processed particularly in statistical analysis. Computers are suitable for both complex computations as well as processing of massive amount of data. Research work in science and engineering field also requires complex computation. Such complex computation is not possible by hand or calculators. Nowadays computers are must for this type of work.

9.20 REAL-TIME OR ONLINE APPLICATIONS

When a computer is controlling certain machine, equipment, process or certain activity, it measures and receives data (status or other information) regarding the equipment, process or activity which is to be controlled. If the measured quantity deviates from its predetermined value, the computer sends control signal for its immediate correction. This type of computer control is known as *real-time* or *online control*. In real-time control the computer receives data, processes the received data immediately and gives output to control, direct or change the process or ongoing activity. In such a situation there is time constraint. Suppose a computer is controlling the launching of a satellite. It measures its distance and angle. If it deviates from the prescribed direction, the computer sends control signal for correction. The processing of data, computation, sending of control signals all are to be performed within a prescribed limit of time, otherwise the satellite will go far off and the project will fail. Take another example of computerized protective relays. Nowadays computers are used to protect power plants, electrical transmission lines etc. If a short-circuit occurs on the power network, it has to be detected and the faulty part must be isolated within a fraction of a second or otherwise the system will be damaged. So the computer is to be fast enough to perform its task of protection within some milliseconds. There may be certain control, where the time limit is of the order of a few seconds, for example control of temperature of a furnace. The computer may measure temperature and control it at every 15 seconds. So the entire temperature measurement and control process can be done in some seconds; there will be no harm. Computers are very fast they do computation and control tasks in some milliseconds. Most of the industrial controls are real-time control. The monitoring of spacecraft is also an example of real-time control. The computers are used to monitor the space-craft from launching to landing.

9.21 POINT OF SALE (POS) TERMINALS

A point of sale-terminal is used in departmental stores, shops and other retail centres to process sale transaction. Usually an optical bar reader is provided with a point of sale terminal. Bar codes are marked on most packets of commodities to be soled by merchants to identify them. For details please see Sec. 7.1.6. The point of sale-terminal is connected to a computer for processing sales. The sold packets are passed through a scanning window. The optical bar reader reads the bar code of the item to identify it. The computer stores its price and other details. The computer prints name of the item, rate, price and other details on the receipt and issues the receipt to the customer. If a customer purchases a number of items, the computer issues the receipt for all the items after making calculations. It mentions details of all items on the receipt and the total amount to be paid. It also updates sale records and inventory.

A point of sale-terminal is not restricted to reading only bar codes. It can read printed price and other descriptions also. If there is no optical character/bar reader provided with a point of sale terminal or PC, salesman enters item number of the item sold. The computer gives the name, price and other description on the receipt corresponding to the item number entered into the computer. All this information remains stored in the computer's memory. If the POS terminal is connected to banks and credit agencies through telecommunication network, it will entertain cheque or credit card.

9.22 FINANCIAL TRANSACTION TERMINALS

Besides online teller terminals which handle customer's withdrawals and deposits, financial transaction terminals are also commonly used in the premises of departmental stores, shops and other retail centers; hotels, railway stations etc. to receive and dispense cash and handle routine financial transactions. A plastic currency or debit card is issued to the customer. The card contains a magnetic tape on its backside to record account number and the amount at his credit. When the customer inserts his card into the terminal, the terminal reads the data on the tape and transmits it to the CPU which processes his account. The customer directs computer to perform financial transaction from the terminal point.

Smart cards have also been developed. They are alternative to magnetic cards. A smart card contains a built-in microcomputer chip instead of magnetic tape. In case of smart cards there are less chances of fraud. How much cash a customer has to his credit is stored in the chip before it is issued to him. When the customer uses the card to make purchases, the required amount is deducted from the balance by a special electronic machine used by merchants. The electronic machine used by merchants communicate with the card-issuing company's computer time to time for money transactions. Thus a card holder has facility of keeping electronic money with him conveniently and gradually use in making purchases. When his electronic money is used up he can replenish electronic money by depositing money at automatic banking machine of card-issuing company. A record of purchases made by the customer can also be stored in a smart card.

9.23 SOME OTHER APPLICATIONS OF COMPUTERS

Some general applications of computers which have not been discussed earlier are mentioned here. A microprocessor based system is used to control efficient burning of fuel in a car, bus, truck etc. Smart cars are also on market. Microprocessors control engine operations such as air-fuel mixture, ignition timing etc. They control door locks, air conditioning, braking, skidding and speed of the car. Other functions are vehicle diagnosis, collision avoidance, maintenance analysis, and vehicle performance analysis.

Smart watches are also on market. A smart product contains embedded microcontroller. The microcontroller gives time, date, month etc. Besides these things the user can also enter some important dates such as birthdays of friends and family members, dates of important functions etc. The watch will flash a reminder when each date comes.

A number of smart products such as smart crane, smart material handler, smart camera, smart TV, smart thermostat, smart taxi meter, etc. are available. The user can program a smart product to perform certain tasks as he desires. A smart taxi meter keeps record of charges of different passengers sharing a taxi even when they are travelling to different places.

Nowadays personal computers are being used in home for a variety of tasks such as playing games, communicating with database services, for word processing, computer aided education, home management, accounting etc.

Automatic photo laboratory uses microprocessor-based machine to produce positive photo copies from negatives. The adjustment of light, time etc. is done by the microprocessor. It improves the quality as well as output.

Computers are being used in automatic electronic telephone exchange. They improve the overall efficiency of the exchange. The exchange can handle larger number of customers in faster and efficient manner.

All sorts of communications are controlled by computers, they may be through telephone lines, through dish antennas or satellites. Telescopes are controlled by computers. The computer-controlled telescopes are able to detect and track satellites beyond radar range. All district headquarters in India are connected by computers. This is under the scheme of NIC (National informatics centre). This scheme uses dish antenna for interlinking. The zonal offices are at Delhi, Bhubaneshwar, Pune and Chennai. A person sitting at district headquarters can connect his computer with any other district headquater's computer of NIC network through dish antenna. He can send or receive information directly. The scheme uses satellite communication.

9.24 COMPUTERS WITH VISIONS

A computer can be provided with suitable sensors which enable it to see its environment. For many applications computers with vision are needed. For example, a computer-based security system is required to look around to see if any intruder has entered the restricted area. It gives warning to intruders not to move further and alerts security personnel. In industries there is a high demand of robots with vision. Some robots with vision can select and sort various components from a random mix. Other robots can lift and position parts on an assembly line. They can also check the quality of assembled products. Still other robots with vision system can recognize, understand, insert and align the various parts required while making assembly of certain items. The development of robots with vision system is expected to allow widespread automation of quality-control, processing, checking and making assembly of certain products in future.

There are several technologies which provide vision capability to computers. They use OPTICRAM cameras, video cameras, CCDs (charge-coupled devices) or devices employing ultrasonic sound. A brief description of some mechanisms is given in the following sections.

9.24.1 OPTICRAM Cameras

An OPTICRAM camera contains a 64 KB dynamic RAM. The cover of the dynamic RAM is made of glass instead of metal or plastic. An image of an object is directly focussed onto the glass surface of dynamic RAM by the lens on the front of the camera. The dynamic RAM contains 65,536 (64 KB) memory cells. These cells are divided into two sections. Each section is a matrix of 128×256 cells. Each cell functions as a pixel.

In a dynamic RAM 0 and 1 are stored as charge on tiny capacitors. As the charge gradually changes due to leakage, a dynamic RAM has to be refreshed periodically. If light falls on a dynamic RAM cell, the stored charge on the tiny capacitor changes at a faster rate as compared to the usual condition when there is no light falling on the memory cell. A dynamic RAM can be used as an image sensor. For this purpose all memory cells, are first charged to logic 1 level. After certain predetermined interval of time the logic level on each cell is read by a microcomputer. The cell whose logic has not changed and still retains logic 1 level corresponds to a dark spot. The cell whose logic level has changed from 1 to 0 represents a light spot. The logic level of an OPTICRAM can be read and stored directly in the microcomputer memory for processing. The light sensitivity of the camera can be adjusted.

It is done by adjusting the time interval between charging up all the cells and reading out the logic levels on the cells. For example for brighter light a shorter time interval is used.

A Micron Eye camera employing an OPTICRAM has been developed by Micron Technology. It contains a printed circuit board with suitable interfacing circuitry so that it can be interfaced to a microcomputer. With this type of sensors bit pattern of images are stored in the computer memory.

An OPTICRAM camera is relatively inexpensive. It can easily be interfaced to a microcomputer. It has enough resolution to be applied to robots and other similar applications.

9.24.2 Video Cameras

Video cameras and TV station cameras contain a special vacuum tube. There is a light-sensitive coating on inside of the face of the tube. This special vacuum tube is called vidicon. An electron beam is swept horizontally and vertically on the face of the vidicon in the same way as on a TV screen. When the beam is at a particular spot of the vidicon the beam current is proportional to the intensity of light falling on that spot. The vidicon gives an analog output signal for each scan line. The magnitudes of the analog signal at various points are proportional to the amount of light falling on different spots along the scan line. To convert analog signals to digital signal A/D converters are used. Corresponding to each dot on a picture there is an output from the A/D converter. Its value determines intensity of the dot. In this way a picture can be stored in the digital form. Video cameras are expensive, have high resolution so they can not be used for all types of applications. For less expensive systems Charge Coupled Device (CCD) cameras are used.

9.24.3 CCD Cameras

A CCD camera is less expensive but resolution is not as high as that of a video camera. It can easily be interfaced to a microcomputer. It is more rugged and smaller in size. It is used in robots to sense images.

CCDs are semiconductor devices. They are fabricated as long shift registers on semiconductor material. There is a P-type substrate. On this P-type substrate there is an oxide layer (an insulating layer). Finally on the oxide layer there are isolated gates. When a positive potential is applied to a gate with respect to the substrate, a potential well is developed under that gate. In this condition if a charge of electron is injected into the region under the gate, the charge will be retained there. If a sequence of clock pulses are applied to the gates, the stored charge will shift along the region under the next gate. In this way a CCD can operate as a digital or analog shift register.

To utilize CCDs in computer vision several hundred CCD shift registers are fabricated in parallel on the same IC. A photodiode is doped in under every other gate. To create potential well all the gates having photodiode under them are made positive. When light falls on a photodiode a charge proportional to the intensity of the light is produced. This charge is held in each well. These charges can be shifted. For each scan line of a picture dot by dot values are obtained by shifting these charges. A CCD register gives video information in discrete samples. As the charge held in a well is proportional to the intensity of light, the discrete samples are in analog form. They are converted to digital form by A/D converters before processing by computer.

9.24.4 Ultrasonic Vision

Sound waves above human hearing range are called ultrasonic waves. Bats can see in the dark. They emit ultrasonic pulses and determine the distance of the obstacle on the basis of time taken by echoes to return. This technique is used by some cameras to adjust distance automatically while taking photograph. Texas Instruments produces kits for this purpose. Such a kit with some simple circuitry can be interfaced to a microcomputer to provide ultrasonic vision.

Ultrasonic pulses are sent out. When each of the sound waves returns they are received by a transducer. The transducer then produces an analog electrical signal. The time between sending out the ultrasonic waves and receiving the echo back can be measured by microcomputer. The distance at which an obstacle is located can be calculated by the computer on the basis of the measured time and speed of the ultrasonic sound waves.

This type of vision can be used in robots. The range of this type of range finder is about 10 m and resolution is about 3 mm. For greater resolution optical vision systems are recommended.

PROBLEMS

1. Enumerate the applications of computers in our daily working life. Give some examples of home applications.
2. What do you understand by the term office automation and 'Electronic Office'? Can a person do the office work sitting at home?
3. What is a wordprocessor? Discuss the main features of a word processor. Can a general purpose computer function as a word processor?
4. What is a wordprocessing package? Give examples of some wordprocessing packages? What are the main features of a wordprocessing package?
5. How are the documents printed using computers? What is offset printing? Is computer's print essential for offset printing?
6. What is desktop publishing? What are its advantages over wordprocessor?
7. How are computers used in data analysis? What is a spreadsheet program? How is a spreadsheet program used in data analysis?
8. Discuss how are computers used in accounting, investment, preparation of budget and inventory control?
9. Discuss how are computers used in graphics?
10. What are CAD and CAM ? Discuss their utility.
11. What are graphic tablet, mouse, lightpen and joystick?
12. Discuss the application of computer in project management.
13. Discuss the role of computer in present day communications.
14. Discuss the role of satellite in communications.
15. What is electronic mail?
16. Discuss bulletin board and video conferencing?
17. What are telex, teletext and videotex?
18. Discuss the applications of computers in banks? How can a customer perform bank transactions from home?

19. Discuss the applications of computers in education.
20. What is artificial intelligence? What are expert systems? Give some examples of expert systems. Can an expert system do the job of a doctor?
21. Enumerate the applications of computers in industry. Give some examples of industrial control using computers.
22. What do you understand by real-time application of a computer? Give some examples of real-time applications.
23. Discuss how ticket reservations are done in airlines and railways using computers.
24. What is a robot? How does it work? Give some examples of robot applications.
25. What do you understand by computer vision? Give some examples of computer applications which need vision.
26. What is a point of sale terminal? Discuss its working principle.
27. What are financial transaction terminals? What is a cash card? What is a smart card? Discuss their applications.

10

CHAPTER

BUS ARCHITECTURE

10.1 INTRODUCTION

Memory and peripheral devices are connected to the processor through a group of lines called a bus. Three types of buses, namely address bus, data bus and control bus have already been discussed in Chapter 1, Section 1.3.5. There are different types of bus standards which will be described in this chapter. Important types of bus standards (architectures) used in modern computer systems are:

- (i) ISA Bus
- (ii) PCI Bus
- (iii) PCI Express Bus
- (iv) AGP
- (v) Universal Serial Bus (USB)
- (vi) IEEE 1394 BUS

10.2 ISA BUS

ISA is the abbreviation of industry standard architecture. It is pronounced as e-sah. It has 24 address lines and 16 data lines. It is used in 386 and 486 single user system. It doesn't take full advantage of the 32-bit address bus and 32-bit data bus of a 32-bit microprocessor. This reduces the data transfer rate of the system. This problem can be solved to some extent using a 32-bit bus between the CPU and memory, and employing a cache on the motherboard. The ISA bus is then connected to the 32-bit bus through an ISA expansion bus controller. This technique allows the CPU to access memory at the full speed of a 32-bit microprocessor. In such a system ISA bus (a 16-bit expansion bus) is used to transfer data from slow peripherals which are designed according to ISA, for example IDE hard disk drive etc. In single user system the limitations of an ISA bus do not have appreciable effect on the performance of the system. The advantages of an ISA bus are its low cost and availability of many peripheral boards for it. Fig. 10.1 shows a typical example of a bus architecture including an ISA bus.

10.3 EISA BUS

For a multiuser/multitasking system ISA bus is not suitable because of its low data transfer rate. Also, there is no mechanism for bus arbitration. EISA and MCA bus architecture is suitable for multiuser system. EISA is abbreviation for extended industry standard architecture. It uses 32-bit address bus and 32-bit data bus to fulfil the needs of a 32-bit microprocessor. Data transfer rate of EISA is twice that of ISA.

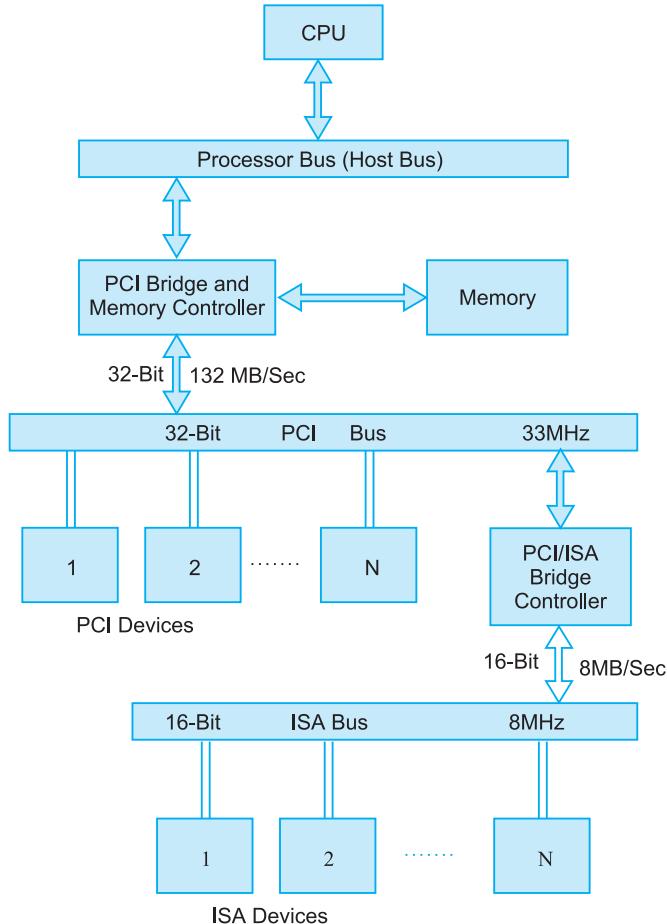


Fig. 10.1 PCI bus in combination with ISA bus.

In a multiprocessor system the processor which takes over the control of buses is called a **bus master**. The controller which takes over the control of buses for DMA data transfer is called a **DMA slave**. The EISA bus system can support up to 6 bus masters and 8 DMA slaves.

EISA bus system is 100% compatible with ISA 8-bit and 16-bit expansion boards and software. EISA connector is superset of the ISA connector to maintain full compatibility with ISA expansion cards. EISA connector contains two layers of contacts. The top layer contacts are the contacts for ISA signals. The bottom layer contacts are the contacts for additional EISA signals. The connectors of the EISA bus are of the same physical size as that of an ISA bus connector so that either ISA or EISA cards can be inserted in the EISA connector slot.

10.4 PCI BUS

PCI has been developed by Intel. It is a 32-bit bus which extends the processor's own local bus, and can be expanded up to 64-bit when need arises. It offers higher performance, automatic configuration of peripheral cards and superior compatibility. It is a costly system. The PCI bus system is able to support ten devices because PCI devices do not electrically load down the CPU bus. Logical mechanical and electrical specifications of PCI bus have been given very clearly. The PCI bus system can transfer data at a rate of 130 MB per second at 33 MHz.

PCI bus is a high performance connection between the motherboard components and expansion boards of a system. There is a bridge-chip between the processor and the PCI bus, which connects the PCI bus to the processor's local bus. This allows to connect PCI peripherals directly to the PCI bus. Once a host bridge chip is included in the system, the processor can access all available PCI peripherals. This makes the PCI bus standard processor independent. When a new processor is to be used, only the bridge-chip needs to be replaced; the rest of the system remains unchanged. The PCI bus employs a 124-pin, microchannel style connector (188 pins for a 64-bit system). PCI specifications are for two types of connection: 5 V system and 3.3 V low power system. PCI design have ability to support future generations of peripherals.

Figure 10.1 and 10.2 show PCI bus architecture. Since PCI bus slots do not accept 8-bit and 16-bit ISA cards, ISA bus is also used in combination with PCI bus to interface 8-bit and 16-bit older ISA cards. (See Fig. 10.1 and 10.3).

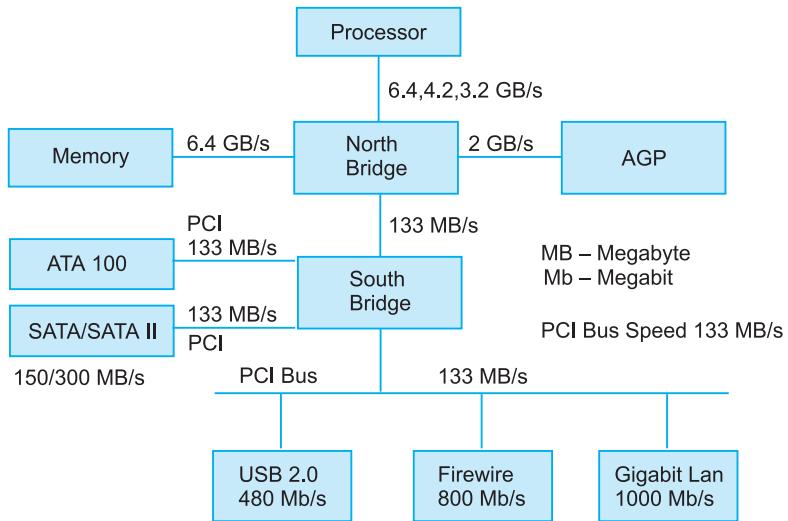
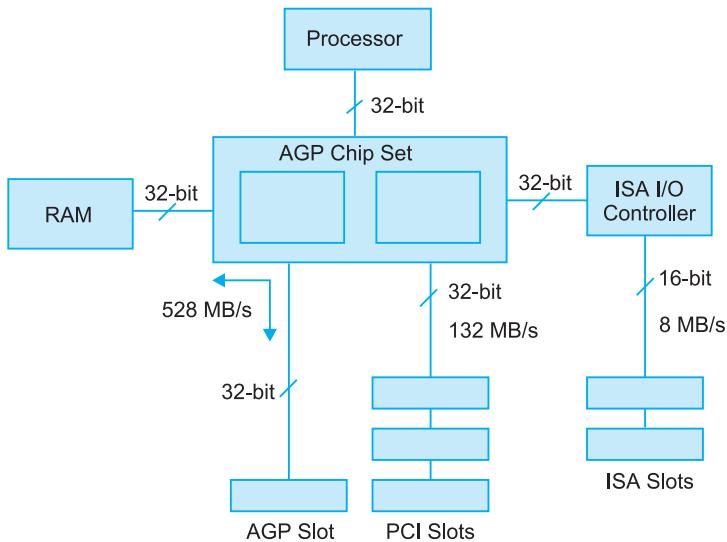
The North Bridge and South Bridge shown in Fig. 10.2 are the two chips of the chipset. The North Bridge is meant for memory interfacing and the South Bridge for I/O interfacing. The peripherals are connected to PCI or other buses. A bus is connected to the CPU through a bridge. A bridge contains necessary electronic circuits to interface a bus to the processor bus (CPU bus). The bridge translates the signals and protocols of one bus to those of the other.

10.5 AGP (ACCELERATED GRAPHIC PORT)

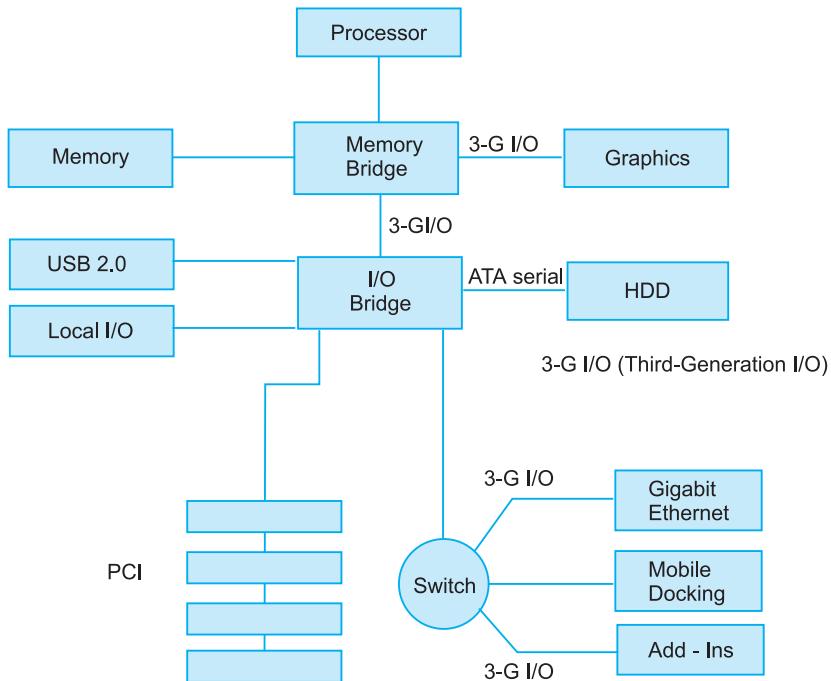
AGP is actually a bus. It is specially designed for video card. Its data transfer rate is 528 MB/sec or more. It is an improvement over PCI bus. It operates at processor bus speed. The video card contains video accelerator which can access main memory at high speed through AGP bus and the chipset. The video accelerator performs image calculation. It generates and processes pixels. It receives commands from the CPU, converts graphics commands into a data stream and keeps in the local memory. A video accelerator is provided with local memory. It contains DAC (Digital to Analog converter) which receives information from the local memory and controls the intensity of red, blue and green electron beam. An AGP slot has already been shown in Fig. 10.2 and 10.3.

10.6 PCI EXPRESS BUS

The PCI Express bus is also written as PCIe bus. It is faster and more powerful than PCI bus. Fig. 10.4 shows a schematic diagram of PCI Express Bus Architecture. The memory bridge and the I/O bridge are the two chips of a chipset. Unlike PCI bus, PCIe bus is a two-lane transfer system. This allows data to come and go simultaneously. PCI bus becomes a subset of PCIe (see Fig. 10.4). If an user has PCIe based motherboard, he has to buy a PCIe graphics card as well. PCIe is backward compatible with older PCI components.

**Fig 10.2** PCI bus architecture**Fig. 10.3** AGP slot along with PCI and ISA slot

That latest PCIe chipsets from Intel, 915, 925 and their variants, come with an on-board Intel graphics Media Accelerator, which has Direct-X 9 support. The 915G and 915V chipsets have native support for HDTV (High Definition TV) and widescreen LCD monitor. The motherboard containing 915 or 925 chipset has an x16 PCIe slot for a graphics card and 2 or more x1 PCIe lanes in addition. These chipsets have support for SATA drives and one ATA drive. The 915 and 925 boards are equipped with integrated 802.11g wireless LAN access, but manufacturer may remove this function to reduce cost.

**Fig 10.4** PCI express bus architecture

10.7 UNIVERSAL SERIAL BUS (USB)

It is a high-speed serial bus. Its data transfer rate is higher than that of a serial port. It allows to interface several devices to a single port in a daisy-chain. It supports interface for a wide range of peripherals such as monitor, keyboard, mouse, modem, speaker, microphone, telephone, scanner, printer, etc. It provides power lines along with data lines. USB cable contains four wires. Two out of four wires in the USB cable are used to supply electrical power to peripherals, eliminating bulky power supply. Low-power devices such as keyboard and mouse which need small amount of power, get power from USB cable. The device which requires larger amount of power for example a big loudspeaker, must have a local power supply. Out of four wires of USB cable, two wires are used to send data and commands. USB assigns address to each device. It permits devices to communicate to one another, without involving the host in the conversation. It can operate in two modes: low-speed mode and medium-speed mode. In low-speed mode data transfer rate is 1.5 Mbps. At medium-speed mode data transfer rate is 12 Mbps. It uses three types of data transfer : isochronous or real-time, interrupt driven and bulk data transfer. In isochronous data transfer there is no interruption in the flow of data such as video or sound. In such a case uniform amounts of data must be transferred every second, and fixed amounts of data must be transferred in chunks on a regular schedule. Wireless USB called WUSB has also been developed.

10.8 IEEE 1394

It is also a standard for high-speed serial data transfer. IEEE 1394 connector also connects a number of serial devices in daisy-chain fashion. It is limited to shorter distances than

USB bus. With both USB and IEEE 1394, a device serves as a link to another device. It is possible to plug in a device to a PC and then plug other device into the device which has already been connected. In this way the cable is extended as far as it is desired. IEEE 1394 bus is much faster than USB bus. IEEE 1394 cable also has four wires : two for power supply and two for sending data and commands. Apple computer helped the development of IEEE 1394 standard and they have trademarked it in the name of “**Firewire**”.

10.9 IEEE-1284

It is a standard to interface peripheral devices to the parallel port of a computer. Printer, CD-ROM, external hard disk drive, ZIP drive and other mass storage devices can easily be connected to a parallel port. If any such device is connected to a parallel port, it becomes slower than if the device would have been connected to the PC's I/O bus via a plug-in card. But it is convenient to hook up an external peripheral device to the parallel port of a PC.

IEEE-1284 has five modes of operation. One of them is also the centronics mode (compatibility mode) for printer connection. Two advanced modes are: EPP (Enhanced parallel port) and ECP (Extended capability port). EPP is super-parallel port design for bidirectional data flow. Its data transfer rate is up to 2 MBps which is over 10 times faster than compatibility mode. It is suitable for ZIP and other mass storage devices. It was developed by Intel, Xircon and Zenith.

ECP developed by Hewlett-Packard, provides data compression along with other features. It is capable of providing channels to transfer different data simultaneously. It makes possible to communicate with one channel even when another channel in the same peripheral is busy. For example, in a combined printer/fax/scanner device, it is possible to receive fax while the printer is busy in printing a separate documents.

10.10 IEEE-488 BUS

IEEE-488 standard was developed by Hewlett-Packard to interface smart test instruments with a digital computer. The IEEE-488 bus is also known as GPIB (General-Purpose Interface Bus) or HPIB (Hewlett Packard Interface Bus). It uses 24-wire cable. 8-10 devices can be connected in daisy-chain fashion. The bus includes 8 bidirectional data lines, five bus management lines and three handshake lines. The bidirectional lines are used to transfer data, addresses, commands and status bytes. The handshake signals coordinate the transfer of data bytes on the data lines. The bus management lines send signals for reset, interrupt, attention; to indicate end of data block transfer, signal to bring an instrument directly under the controller, etc.

10.11 BUS SYSTEM IN A MULTIPROCESSOR SYSTEM

A number of bus configurations are available for a multiprocessor system. Examples are: single shared bus system, dual bus system, crossbar interconnection, hypercube system, etc. A few bus systems are described below in brief.

Figure 10.5 shows a single shared bus architecture. In such a system only one processor is allowed to communicate with the memory or another processor at any given time. Since it is restricted to one transfer at a time, its data transfer rate within the system is limited by the speed of the single shared bus system. It has been used in computer systems because it can easily be designed and controlled.

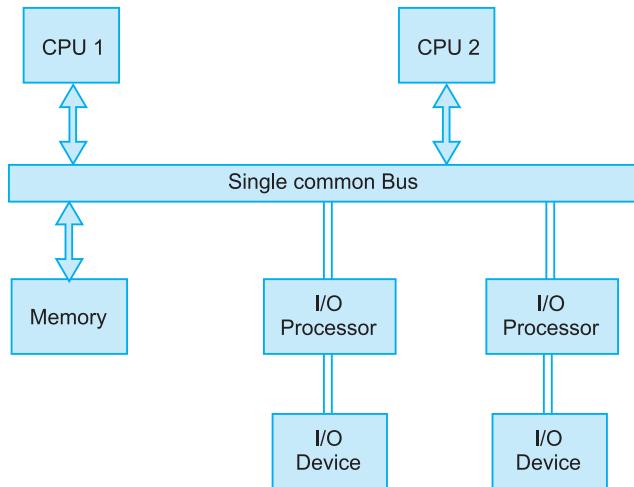


Fig. 10.5 Single-bus system

Figure 10.6 shows a dual bus configuration. In this system local buses are employed. Each local bus is connected to a CPU, the local memory and I/O. A system bus controller connects each local bus to a common system bus. The memory and I/O (if any) connected to the common bus can be shared by all processors. In this system only one CPU is allowed to communicate with the shared memory and other common resources through the system bus at any given time. The other processors use their local memory and I/O devices. It is an improvement over the single shared bus system. It has higher data transfer rate. But is a costly and complex bus system.

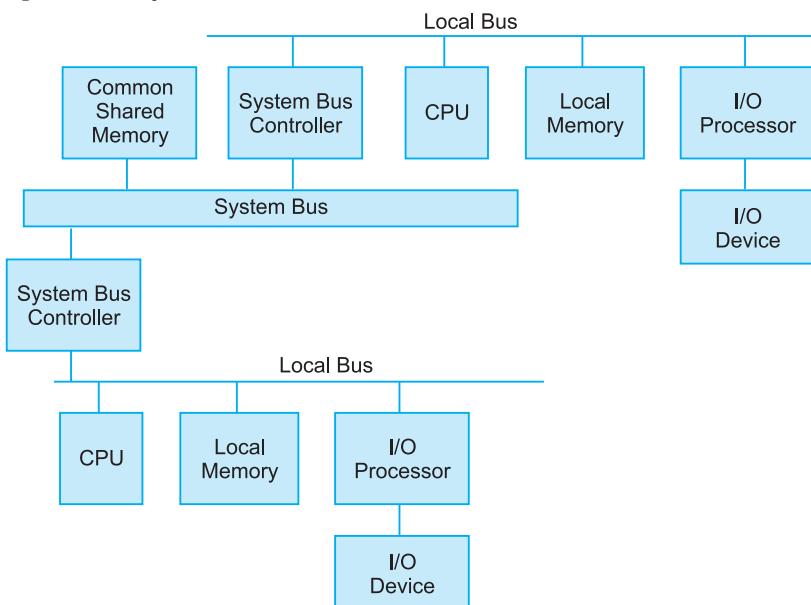


Fig. 10.6 Dual bus architecture

Figure 10.7 shows a crossbar interconnection. It is a type of multibus system. At each crosspoint there is a control logic to determine the path from a CPU to a memory module. It allows simultaneous transfers from all memory modules because there is a separate path

for each module. It permits multiple requests for access to the same memory module on a predetermined priority basis. It is a costly and complex bus system.

Figure 10.8 shows a hypercube interconnection. It has been used for loosely coupled multicomputer system. Each CPU forms a node and has direct connection to neighbouring CPUs. Each CPU has its own local memory and I/O devices.

Static and Dynamic Interconnection

In a multiprocessor system interconnecting paths may be either static or dynamic. In static interconnection unit-to-unit interconnecting paths (links) are static i.e., fixed and unchangeable. In dynamic interconnection the unit-to-unit interconnecting paths are reconfigurable under system control. Examples of static interconnection are: star, ring, hypercube, mesh, linear etc. Examples of dynamic interconnection are: single-bus, multibus, crossbar interconnection etc.

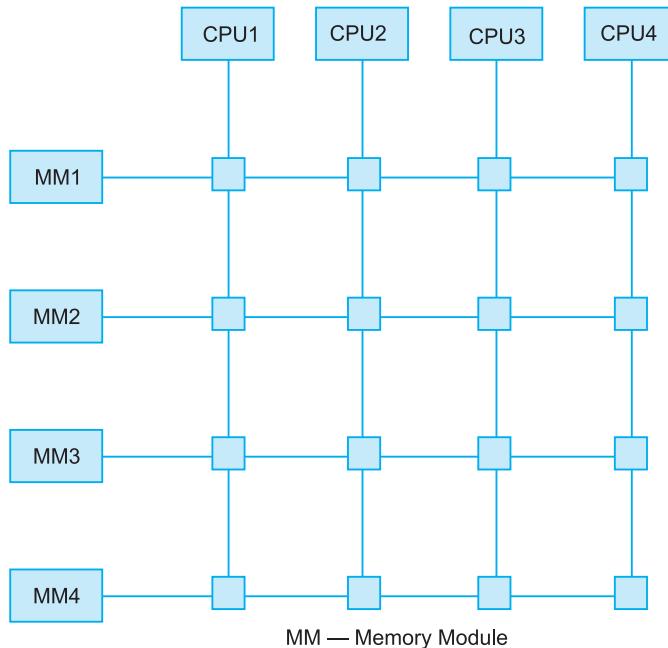


Fig. 10.7 Crossbar Interconnection.

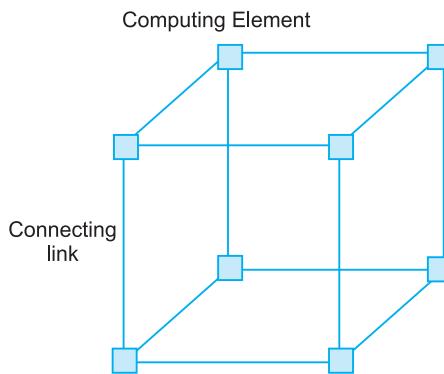


Fig. 10.8 Hypercube Interconnection.

PROBLEMS

1. Which bus architectures are used today in a modern digital computer? Discuss them in brief.
2. What is an AGP? For what purpose is it used?
3. What is ISA? Why is it still used in a modern digital computer?
4. What is PCI? Discuss its merits over other bus architectures.
5. What is IEEE-1284 standard? For what purpose has it been developed?
6. What is IEEE-488? For what purpose has it been developed?
7. What are EPP and ECP? Where are they used?
8. What is processor bus or a host bus? What is the operating frequency of a processor bus, when the CPU is in the range of 3.2–3.8 GHz ?
9. What is a local bus? What types of peripherals are connected to a local bus? Discuss some types of local bus.
10. Draw a figure showing AGP in combination with PCI and ISA bus. Why ISA is used in combination with PCI and AGP?
11. What is PCI Express bus ? What are its advantages over PCI bus ?
12. What is USB ? Discuss it briefly. What is WUSB ?
13. Discuss IEEE-1394. Where is it used ?
14. What is firewire ? Where is it used ?

**THIS PAGE
IS BLANK**

11

CHAPTER

PARALLEL PROCESSING

11.1 INTRODUCTION

Light or electromagnetic waves travel 3 cm in 0.1 nanosecond. The speed of electrons in electronic devices has a limit set by the speed of light. The electron speed in modern digital system is reaching its speed limit. Hence, it is not possible to increase the speed of a computer consisting of a single processing unit beyond a limit. Therefore, to get higher speed an alternative approach is to use several processing units to operate in parallel.

When several computations are performed concurrently in a computer, it is known as *parallel processing*. A multiprocessor system performs several computations concurrently. Parallel processing is also achieved in a uniprocessor system using pipelining. There are three types of parallel processors/computers based on their architectural configuration:

- (i) Pipelined processors
- (ii) Array processors
- (iii) Multiprocessor system

11.2 PIPELINED PROCESSORS

In a Von Neumann type processor, to execute an instruction the following steps are involved:

- (i) Instruction fetch (IF) from the main memory.
- (ii) Instruction decoding (ID). After decoding an instruction the processor comes to know what operations are to be performed.
- (iii) Operand fetch (OF), if any
- (iv) Execution of the decoded instruction (EX)

In a non-pipelined processor all the above steps for the execution of an instruction are completed, then the next instruction to be executed is fetched from the memory, as shown in Fig. 11.1(c). In a pipelined processor, while one instruction is being fetched, the 2nd is being decoded, the operand is being fetched for the 3rd and the 4th is being executed. The instruction fetching unit, the instruction decoding unit, operand fetching unit and the execution unit all operate simultaneously. This type of overlapped operation is shown in Fig. 11.1(b). In first cycle instruction No. 1, that is, I_1 is fetched from the memory. In the second cycle another

instruction I_2 is fetched and simultaneously I_1 is decoded by the instruction decoding unit. In the third cycle the instruction I_3 is fetched, I_2 is decoded and the operand for I_1 is fetched. In the fourth cycle the instruction I_4 is fetched, I_3 is decoded, the operand for I_2 is fetched and I_1 is executed. Such a cycle is called *pipelined cycle*. This type of processing technique is called pipelining. In the fifth pipelined cycle the instruction I_5 is fetched, I_4 is decoded, the operand for I_3 is fetched and I_2 is executed. In this way the execution of other instructions proceeds.

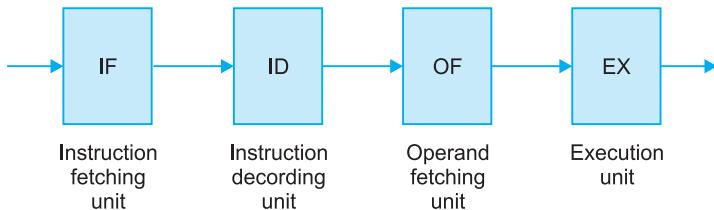


Fig. 11.1(a) Functional units of pipelined processor

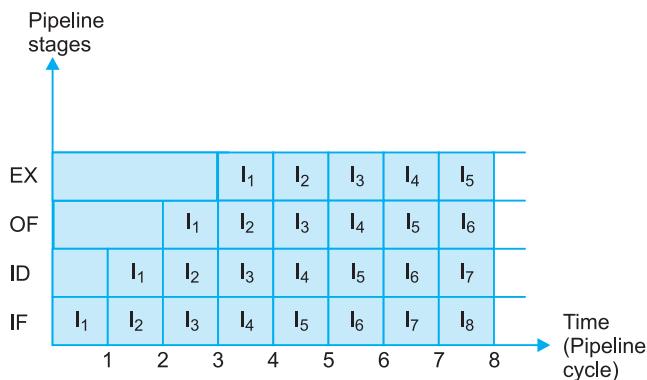


Fig. 11.1(b) Overlapped operations in a pipelined processor.

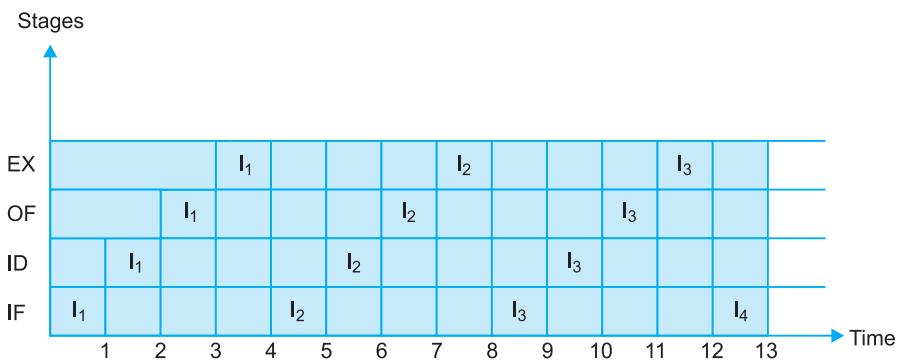


Fig. 11.1(c) Instruction execution in a non-pipelined processor.

Fig. 11.1(a) shows how the various functional units of a pipelined processor are connected. The concept of the pipelined processing is similar to assembly lines in an industrial plant. In the pipelined processing a task is divided into a number of subtasks. Each subtask is executed by a specialized hardware unit which operates concurrently with the other units in the pipeline. Successive tasks are streamed into the pipeline and get executed in an overlapped manner at the subtask levels. In the pipelined processing all the four functional

units of Fig. 11.1(a) operate in parallel. This type of parallel processing is called *temporal parallelism*. The operation of all the functional units is synchronized under a common clock. All the functional units are within a processor and operate under a single control unit. A pipelined computer is a uniprocessor system.

We can take a simple analogy to explain pipelined processing. Suppose there are 100 answer books to be examined. There were five questions Q_1, Q_2, \dots, Q_5 to be answered. If there is only one teacher to examine all the answer books, he will take one answer book at a time. He will evaluate all the answers in an answer book and add the marks given for each question. After completing one answer book he will take up another answer book for evaluation. This is an example of non-pipelined processing.

To speed up the task of evaluation five teachers can be asked to co-operatively evaluate the answer books. All the five teachers sit at a time in a line. The first teacher in the line will evaluate the answer to Q_1 and then he will pass on the answer book to the second teacher. The second teacher will evaluate the answer to Q_2 . The first teacher will take another answer book and evaluate Q_1 in it. In this way an answer book is passed through all the teachers. Flow of answer books is continuous and all teachers are working concurrently. This is an example of pipelined processing. The total time taken to evaluate all answer books by this method is about 1/5th of the time taken by one teacher.

11.3 ARRAY PROCESSORS

An array processor contains several processing elements (PEs) which operate in parallel under the control of a single control unit (CU). A processing element is an arithmetic and logic unit (ALU) with attached registers and local memory. It does not have instruction decoding capability. The control unit has its own memory. The control unit controls the execution of the system and user programs. The control unit decodes all the instructions. The scalar or control-type instructions are directly executed in the control unit. Vector instructions are transmitted to PEs for distributed execution. All the PEs execute the same instruction under the control of the control unit. Vector operands are stored in the local memories of PEs before they execute an instruction in parallel on different operands. These operands are loaded into the local memories from an external source via the system data bus or via the CU using control bus. An array processor processes single instruction and multiple data streams (SIMD), that is the same instruction is to be executed on an array of data. An array of data usually involves vector and matrix computations. Array processors are also called SIMD processor. Array computers are very much suitable for vector processing but difficult to program as compared to pipelined computers. Fig. 11.2 shows a schematic diagram of an array processor or SIMD processor.

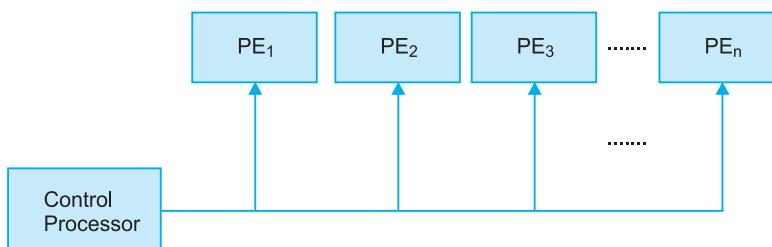


Fig. 11.2 Array processor or SIMD processor.

11.4 VECTOR PROCESSORS

A *vector* is a linear array of numbers (elements). Examples of vectors are:

Vector A ($a_1, a_2, a_3, \dots, a_n$)

where $a_1, a_2, a_3, \dots, a_n$ are the elements of the Vector A.

Vector B ($b_1, b_2, b_3, \dots, b_n$)

where $b_1, b_2, b_3, \dots, b_n$ are the elements of the Vector B.

Vector C = Vector A + Vector B

= Vector C ($c_1, c_2, c_3, \dots, c_n$)

where $c_1 = a_1 + b_1, c_2 = a_2 + b_2, \dots, c_n = a_n + b_n$

A vector processor can add all the elements of Vector A and Vector B by single add instruction using hardware approach. Similarly, a vector processor can also perform other arithmetic and some other operations using single vector instruction, on all the elements of a pair of vectors simultaneously. Array processors also do vector computation. A vector processor employs multiple pipelines which operate in parallel. On the other hand an array processor employs multiple processing elements which operate in parallel.

11.5 MULTIPROCESSOR SYSTEM

A computer system which contains one CPU is called *uniprocessor system*. A computer system that contains two or more CPUs is called a *multiprocessor system*. Some supercomputers and powerful servers contain two or more CPUs. These CPUs operate in parallel under the integrated control of an operating system to handle one task or job.

The multiprocessor computers are mainly of two types: shared memory type and distributed memory type. If the main memory or a major portion thereof, can be directly accessed by all the processors of a multiprocessor system, then the system is referred to as *shared-memory type multiprocessor computer*. The shared portion of the main memory is called *global memory*. In the shared memory system a small local memory or high-speed buffer (cache) may exist with each processor. Normally, each processor accesses instructions and data from its own local memory. An interconnecting network is provided to allow a processor to access the global memory and I/O devices. The global memory holds programs and data that are to be shared among the processors. It provides a means for interaction among the processors. A shared memory type multiprocessor machine is simple and attractive. Its limitation is that it can not scale to large sizes. It contains less number of CPUs as compared to a distributed memory type machine.

In the *distributed memory type multiprocessor computer*, each processor has a large local (private) main memory. Shared-memory and distributed-memory type multiprocessor computers are also called *tightly coupled* and *loosely coupled* multiprocessor computers respectively. A distributed memory system may have no global memory or little global memory. A distributed memory computer is also called *message passing computer*. In this type of computer, communication occurs by sending messages from one processor to the other.

In a loosely coupled multiprocessing system the interaction takes place at the file level, while in a tightly coupled system processors interact at the level of individual data element. In the latter case there is a high degree of cooperation between processors. In a loosely

coupled system individual processor has its local memory and I/O devices. The individual processor, its local memory and I/O devices together is called a *computer module*.

A multiprocessor system is a single computer that contains multiple processors. Processors of multiprocessor computer communicate and cooperate at different levels in solving a given problem. The communication may occur by sending messages from one processor to the other or by sharing a common memory. A multiprocessor system can be distinguished from a computer network which consists of several autonomous computers operating independently on separate tasks. They may or may not communicate with each other. The processors of a multiprocessor system share the resources such as communication facilities, I/O devices, program libraries, databases, etc. and are controlled by a common operating system. In computer network a person working on a small computer can utilize the computing power of a large computer on the network, may use database etc. though the two systems are not operating under the control of one operating system.

11.5.1 Shared Memory Type Multiprocessor System or Tightly Coupled Multiprocessor System

In a shared memory type multiprocessor system (computer), processors share the same main memory and I/O devices. They are connected by a bus or an interconnecting network. When all the processors have equal access time to all memory words, it is called uniform memory access. This type of shared-memory multiprocessor system is called **UMA (Uniform Memory Access)** model (system). Fig. 11.3 shows a UMA type shared memory multiprocessor system. The memory is usually organized so that multiple simultaneous access to separate blocks of memory can be done.

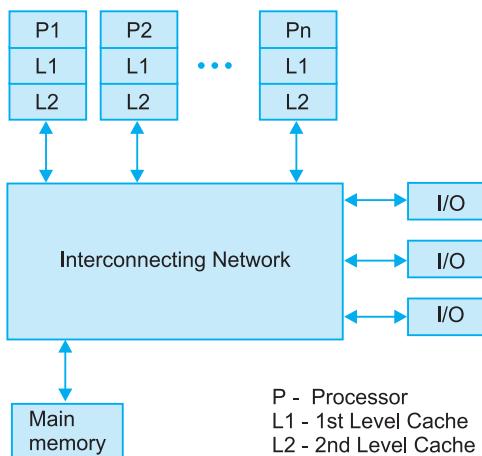


Fig. 11.3 UMA Type shared memory multiprocessor system

If all the processors have equal access time to memory and all peripherals, it is also called **symmetric multiprocessor (SMP) system**. In this case, all the processors are equally capable of running the executive programs, such as operating system kernel and I/O service routines. When only one or a subset of processors are executive capable, the system is known as **asymmetric multiprocessor system**. In asymmetric multiprocessor system an executive or master processor can execute the operating system and handle I/O devices. The remaining processors called attached processors (APs) execute user codes under

the control of master processor. They do not have I/O capability. Both master processors and attached processors share the global memory.

The organizational approaches for an SMP system are as follows:

- (i) Time-shared or common bus
- (ii) Multiport memory
- (iii) Central control unit

A time shared bus is the simplest mechanism to build an SMP system as shown in Fig. 11.4. There are a number of processors and I/O processor in the system. The processors attempt to gain access to one or more memory modules through the bus. The system is simple one and has flexibility. It can be easily expanded by adding more processors to the bus. It is very reliable. If any attached device fails, it does not cause the failure of the whole system.

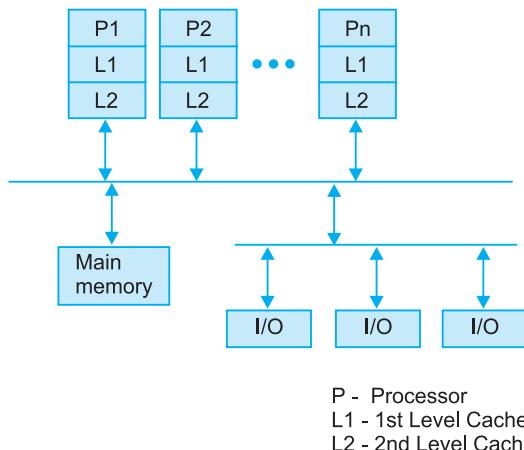


Fig. 11.4 Common bus type SMP multiprocessor system

A **multipoint memory** approach allows direct, independent access of main memory by each processor and I/O module. But it is more complex than the bus approach. However, it gives better performance because each processor has an independent path to each memory module. Fig. 11.5 shows a multiport memory type system.

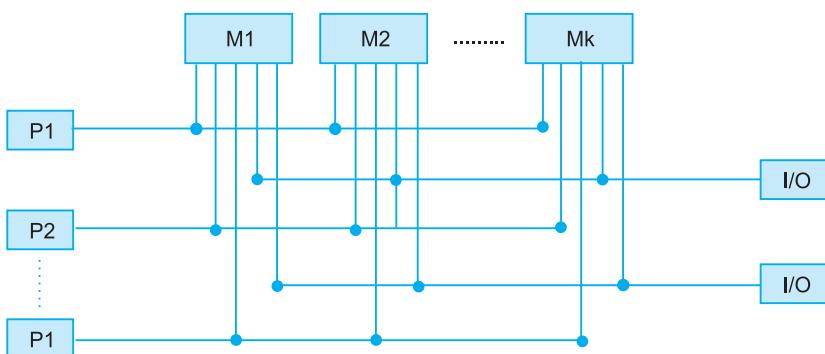


Fig. 11.5 Multiport Memory Type System

NUMA Model of Shared-Memory Type Multiprocessor System

The abbreviation NUMA stands for Non-Uniform Memory Access. Fig. 11.6 shows a schematic diagram of a NUMA type shared memory multiprocessor system. In this system there are memory modules directly attached to each processor. Besides accessing the local memory, each processor can also access other memory modules through the interconnecting network. As the access of remote memory module which is attached to other processor, is done through the interconnecting network, it takes considerably longer time than access of local memory module. Due to this difference in access time, this system is called Non-Uniform Memory Access (NUMA) type multiprocessor system. The collection of all local memories forms a global address space accessible to all processors. Besides the local memories which are distributed memories, globally shared memory can also be added. In such a case, there are three memory access time. The fastest is local memory access. The next is global memory access. The slowest is remote memory (attached to any other processor) access. A NUMA system with cache coherence is known as CC-NUMA system.

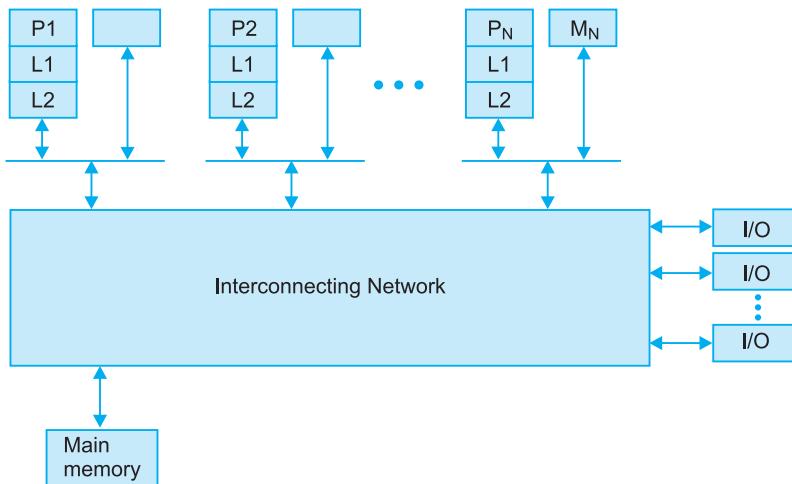


Fig. 11.6 NUMA Type Shared Memory Multiprocessor System

11.5.2 Cache Coherence

In a multiprocessor system, there is a possibility that two or more caches may hold different values of the same variable at the same time. It is known as cache coherence problem or cache consistency problem. When two or more caches contain different values of the same variable, it gives erroneous results because each processor is executing a part of the same large task. There are two approaches to solve cache coherence problem: software approach and hardware approach.

In **software technique**, the compiler checks which data may become unsafe for caching and they are marked accordingly. The operating system or hardware then prevents such data from being cached. This scheme is implemented using write-through cache policy. In write-through cache policy the main memory is also updated each time when CPU writes in the cache (i.e. cache is updated). So when other processors need such data, they read from main memory. This scheme is not very efficient. It degrades system performance.

In **hardware approach** the protocols may be a write-through protocol or a write-back protocol. A write-through protocol may use cache update technique or cache invalidate technique. In the write-through cache update technique, when a processor writes into its cache (i.e., it updates its cache), it also updates main memory. The processor also broadcasts the modified data to all processors. As each processor receives broadcast data, it updates its cache.

A write-through protocol may also use cache invalidate technique. In this technique, when a processor updates its cache, it also updates memory, and sends signal to invalidate caches of other processors. If any other processor wants to read data, it does not get the data from its cache as its cache for that data is invalidated. So it can get data from the main memory which has also been updated.

In write-back protocol, when a processor wants, it updates its cache, it also sends signal to invalidate other caches and main memory. Now this processor becomes exclusive owner of the data. If any other processor wants to read this data, data are sent by the current owner of the data because at present memory is not having the modified data as it is a write-back scheme. In write-back scheme, when processor performs write operation, only its cache is updated. Main memory is only updated when corresponding cache line is flushed from the cache. The write-back protocol generates less traffic than the write-through protocol. This is due to the fact that a processor is likely to update a cache block several times before this block is required by another processor.

11.5.2.1 Cache Coherence Problem Tackling Schemes

There are two important types of cache coherence problem tackling scheme: snoopy cache scheme and directory-based scheme. The snoopy cache scheme is suitable for smaller multiprocessor system which uses a single-bus system. Directory-based schemes are suitable for large multiprocessor systems.

Snoopy Cache Scheme. Each cache associated with a processor has a cache controller. In this scheme the responsibility for maintaining cache coherence is distributed among all the cache controllers. A cache must be able to know the situation when a data held by it is also shared with other caches. When the shared data is updated, all other caches must be informed about it by broadcast. Each cache controller must be able to snoop on the interconnecting network (which is a single-bus system in this case) to know the broadcast messages and take necessary actions accordingly.

The snoopy cache scheme can use either write-update protocol or write-invalidate protocol. In write-update scheme, when a processor updates its cache, it also broadcasts updated data. As all other processors also receive broadcast, they update their caches.

Usually write-invalidate scheme is used with write-back policy. In a write-invalidate scheme, when a processor wants to update any data, it sends message to all other processors to invalidate the cache line which contains that data. It also invalidates the main memory. Then it updates its cache and becomes the exclusive owner of that data. Now this processor may perform more updates on that data without sending any message to other processors, until any other processor needs to update that data. If any processor is an exclusive owner of a data and the scheme uses write-back protocol, the current owner of the data sends this data to any other processor which wants to read this data.

The write-invalidate scheme is widely used in commercial multiprocessor systems, such as multiprocessor systems employing Pentium 4 processors, multiprocessors system employing

PowerPC processors, etc. In this scheme, the state of every cache line is marked providing two additional bits in the cache tag. There are four states which are marked. They are modified, exclusive, shared or invalid. Due to this reason the write-invalidate protocol is known as MESI. M is for 'modified', E for 'exclusive', S for 'shared' and I for 'invalid'. This scheme uses write-back policy of cache writing. When any processor is exclusive owner of a data, it can modify it i.e., it can update its cache without informing other caches about it. When a number of processors are having shared data, a processor will update its cache and invalidate other caches and the main memory. As far as reading of cache data is concerned, any exclusive or shared owner of cache can read its cache and does not send any signal to bus. When any processor has its cache invalidated, it receives data from the current owner of the data because it is a write-back scheme and hence, memory is not having updated data. When memory is updated, all processors which have invalidates cache, come to know that memory has updated data. Moreover, when a processor tries to read data from memory, it knows status of memory whether it is update or invalidated.

Directory-Based Scheme

The scheme is suitable for a large multiprocessor system. The drawback of earlier discussed cache-update or cache-invalidate schemes is that large amount of unnecessary signal traffic is generated by full broadcast. Practically only a few processors may be having the copies of a particular data at a particular time. But all processors receive signals for updating their caches or invalidating their caches. Thus unnecessary much more signals are sent on the bus. To solve this problem, a directory is maintained to tell which caches have copies of the same data at any given time. Messages are sent to only those caches which have the copies of the data.

11.5.3 Clusters

A cluster is a set of computers which are interconnected (networked) to perform as one. It is meant for high performance computation. It is attractive for server application. Clustering is an alternative to symmetric multiprocessing. A cluster may have a number of machines, each of which is a multiprocessor. It has scalability feature. It is arranged in such a way that new systems can be added to it. Fig. 11.7 (a) and (b) shows a cluster configuration without and with shared hard disk.

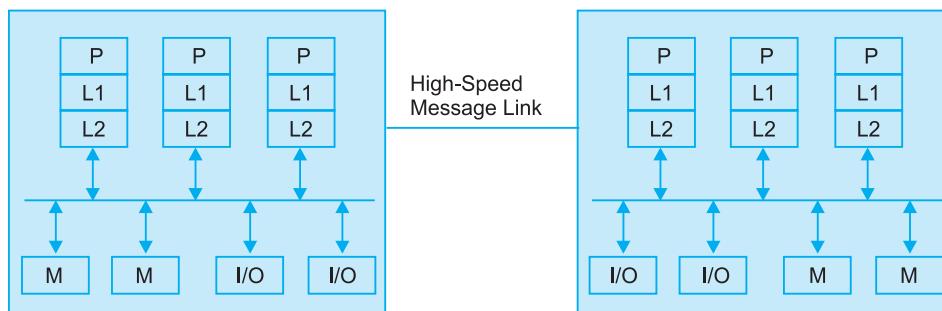


Fig. 11.7 (a) Cluster configuration without shared disk

A cluster is a multicomputer system. It is used to solve a computational task which requires multiple CPUs. All CPUs operate in parallel to solve a common task. It is a MIMD computer. On the other hand, in a computer network, different computers work on separate

independent tasks. However, a number of PCs or workstations can be placed in a big rack. They can be connected to work as a multicomputer system. PCs and workstations spread around a building or campus can also be used as a multicomputer system to solve large computational task which needs multiple CPUs. This requires special software. In day-time these PCs and workstations form a computer network and are connected through LAN. At night, when they are idle can be used as a multicomputer system.

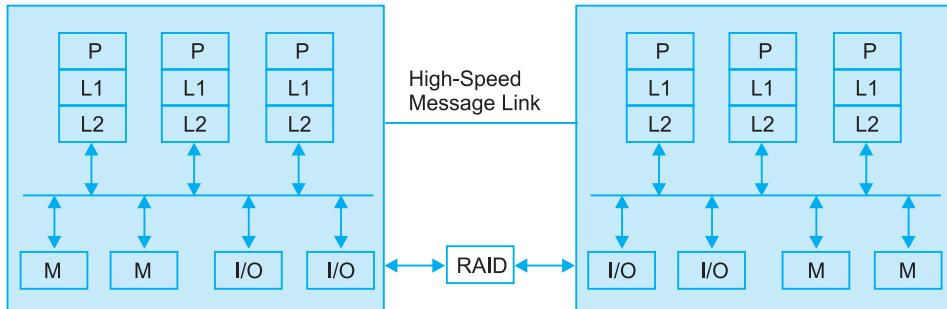


Fig. 11.7 (b) Cluster Configuration with Shared Disk

11.5.4 Distributed Memory System or Multicomputer System

Fig. 11.8 shows a distributed memory type system. In this system each processor has its own memory module. A processor can not access a remote memory without cooperation of the remote processor. In this system communication takes place by sending messages from one processor to another. This type of system is also called **message passing system** or **loosely coupled system**. In this system each node is an autonomous computer consisting of a processor, local memory and sometimes attached disks and I/O devices. Therefore, it becomes a multicomputer system. However, this type of multicomputer system constitutes a single computing machine, it may be a supercomputer, workstation etc. It is a MIMD type computer. All these multiple computers work in parallel, solve a common computational task, operate under the same operating system. On the other hand, in a computer network, different computers work independently on separate tasks. However, it is possible to use PCs and workstations spread around a building or campus to form a multicomputer system when they are idle, particularly in night. Usually, they are connected by a LAN and operate in a computer network in day time. When they are used as a multicomputer system, it requires different software. A large work which needs many CPUs for computation, uses a multicomputer system.

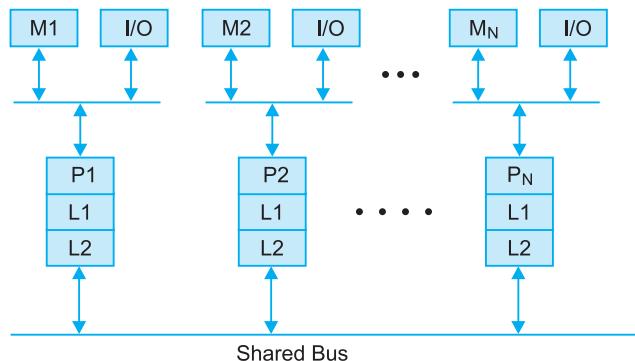


Fig. 11.8 Distributed Memory Type Multicomputer System

11.5.5 Comparison of Shared Memory System and Distributed Memory System

A shared memory multiprocessor system is simple and attractive. As the shared memory uses same memory space it can be accessed easily using ordinary LOAD and STORE instructions. The limitation of this type of scheme is that it can not scale to larger sizes. It can contain comparatively less number of CPUs as compared to distributed memory type machine. Hence, distributed memory type machines which are multicomputer type machine are much more powerful than shared memory type machines (multiprocessor type machines). An NUMA type machines can have more CPUs than UMA type machine. But it has non-uniform memory access. CPUs of multicomputer system can not use ordinary LOAD and STORE instructions for memory access because processor has its own private memory. For interactions they send messages over the interconnecting network.

11.6 FLINN'S CLASSIFICATION OF COMPUTERS

Flinn classified computers on the basis of multiplicity of instruction streams and data streams in a computer system. The following are the four types of computer organization according to this classification.

- (i) SISD—single-instruction stream, single-data stream.
- (ii) SIMD—single instruction stream, multiple-data stream.
- (iii) MISD—multiple-instruction stream, single-data stream.
- (iv) MIMD—multiple-instruction stream, multiple-data stream.

11.6.1 SISD Computer Organization

SISD is a uniprocessor system. Instructions are executed sequentially. They may be overlapped in their execution stages (that is pipelining). SISD may be further classified into two subclasses (i) SISD-S: the class of processors with single functional unit, and (ii) SISD-M; the class of processors with multiple functional units. All functional units operate under the supervision of one control unit. Pipelined computers are SISD-M type machines. They contain several functional units for the overlapped execution of instructions.

11.6.2 SIMD Computer Organization

SIMD machines contain several ALUs. All ALUs execute the same instruction on different data under the control of single control unit. Array type processors discussed in the Section 11.3 are SIMD machines.

11.6.3 MISD Computer Organization

No real computer system exists under this classification.

11.6.4 MIMD Computer Organization

Multiprocessor type computers fall into this class of computer organization.

PROBLEMS

1. Discuss the principle of a pipelined processor. Give examples of some pipelined processors.
2. What is an array processor ? Draw its schematic diagram and explain its operating principle.

3. What is a multiprocessor computer ? What are tightly coupled and loosely coupled multiprocessor systems ?
4. What do you understand by UMA type multiprocessor system ? Draw its schematic diagram.
5. What do you understand by SMP ? Discuss its configuration and draw its block diagram.
6. What is asymmetric type multiprocessor system ? Compare the merits and demerits of symmetric and asymmetric multiprocessor systems ?
7. What is NUMA type multiprocessor system ? Draw its schematic diagram.
8. What are shared memory and distributed type multiprocessor systems ? Discuss their merits and demerits.
9. What do you understand by cache coherence ? How are cache coherence problems solved ?
10. What are snoopy cache scheme and directory-based scheme of solving cache coherence problems ?
11. What do you understand by clusters ? Draw the schematic diagram of a cluster system.

APPENDIX

APPENDIX I REALIZATION OF LOGIC GATES

NOT Gate

Fig. A.1 (a) shows a simple circuit to perform NOT function. When the switch S is open (logic 0), the output voltage V_0 is equal to the supply voltage V (logic 1). When S is closed (logic 1), V_0 is equal to 0 (logic 0), because the point P goes to the ground potential in this condition. The switch S can be replaced by a transistor to perform the same operation, as shown in Fig. A.1 (b). When V_{in} is 0, the transistor does not conduct and V_0 is equal to V (logic 1). When V_{in} is equal to 5 volts (logic 1), the transistor conducts and V_0 is equal to zero (logic 0) as the point P goes to the ground potential. Thus this simple transistor circuit performs NOT function (i.e. inverter operation).

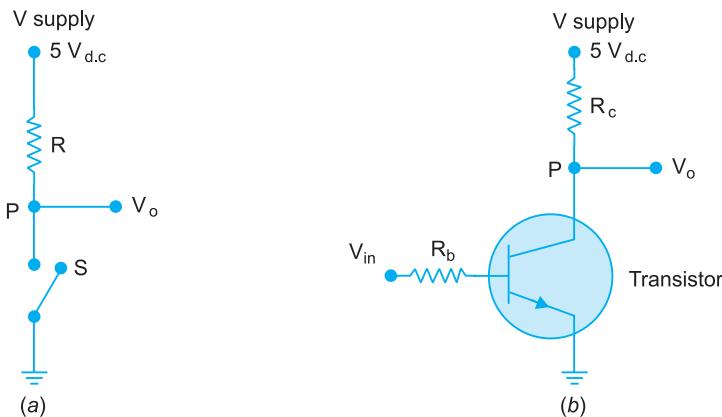


Fig. A.1 NOT Gate

NAND Gate

Fig. A.2 (a) shows a simple circuit to perform NAND function. When any one or both switches A and B are open (logic 0), the output voltage V_0 is equal to the supply voltage (i.e. logic 1). When both switches are closed (logic 1), V_0 is zero (i.e., logic 0). The switches A and B are replaced by Transistors T_A and T_B as shown in Fig. A.2(b). When both inputs A and B are at 5 volts i.e. logic 1, transistors T_A and T_B both conduct and the point P goes on the ground

potential. This makes V_0 equal to zero i.e., 0 logic. When both A and B are at 0 logic both transistors are open (they do not conduct) and V_0 is equal to the supply voltage V i.e., at logic 1. When either A or B is at zero logic one of the transistors will not conduct and hence the point P will not come to the ground potential. In this condition V_0 is at logic 1. Thus this transistor circuit performs NAND function. See the truth table for NAND gate in Chapter 3.

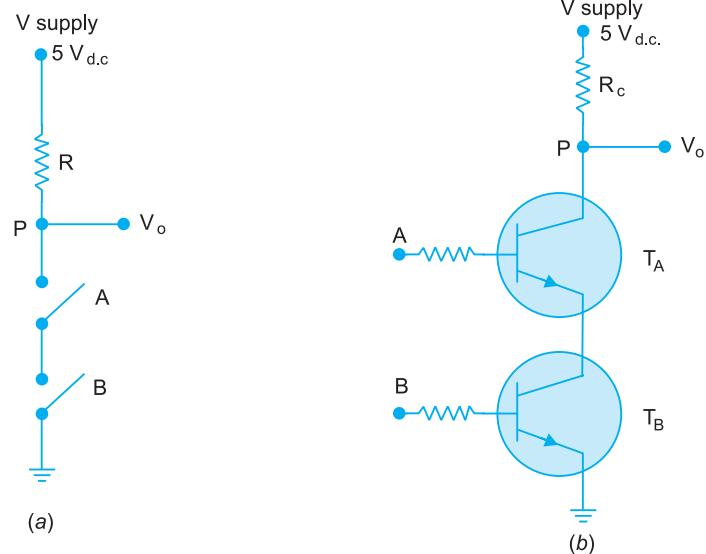


Fig. A.2 NAND Gate

NOR Gate

Fig. A.3(a) shows a simple circuit to perform NOR function. When any one of the switches A and B or both are closed (i.e., logic 1), V_0 is at zero logic. When both the switches are open (logic 0), V_0 is at logic 1. Switches A and B are replaced by transistors as shown in Fig. A.3(b).

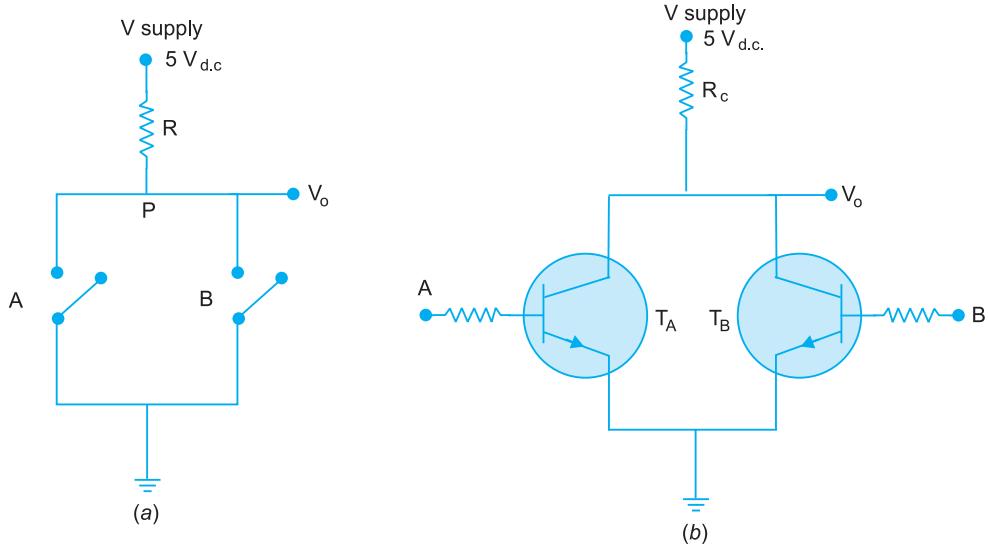


Fig. A.3 NOR Gate

When both inputs A and B are at logic 1, both transistors conduct and V_0 is at ground potential i.e., logic 0. When either A or B is at logic 1 one of the transistors (T_A or T_B) conducts, and the point P goes to the ground potential. This makes V_0 at logic 0. When both the inputs A and B are at 0 logic, none of the transistors conducts and V_0 is equal to the supply voltage i.e., logic 1. Thus this circuit performs NOR function. See the truth table for NOR gate in Chapter 3.

NAND gate is a universal gate. Other gates can be realized from NAND gates. NOR gates are also universal. It is easier to realize NAND and NOR gates than AND and OR gates.

APPENDIX II

PIN DESCRIPTION OF 8086

AD₀-AD₁₅ (output). Bidirectional multiplexed address/data bus.

When AD lines are used to send memory address, abbreviation A_0-A_{15} is used instead of AD_0-AD_{15} . When they are used to transmit data the symbol D_0-D_{15} is used.

A₁₆/S₃, A₁₇/S₄, A₁₈/S₅, A₁₉/S₆ (output). These are high-order address lines multiplexed with status signals. During T₁ these lines carry memory address. During T₂, T₃ and T₄ they carry status signals. S₃ and S₄ are segment identifier. S₅ is interrupt enable status. S₆ is a status signal.

BHE / S₇ (output). Bus high enable/status. During T₁ it is low and enables data onto the most significant data bus, lines D₈-D₁₅. 8-bit devices connected to the upper half of the bus normally use \overline{BHE} for chip select. It is low during T₁ for read, write and interrupt acknowledge cycles when a byte is to be transferred on high-order bus. During T₂, T₃ and T₄, the status S₇ is available on this pin.

A word or a byte can be moved to and from with the help of \overline{BHE} . For this purpose A₀ is also used with \overline{BHE} shown as below:

\overline{BHE}	A ₀	
0	0	one 16-bit word
0	1	high-order byte from/to odd address
1	0	low-order byte from/to even address
1	1	none

RD (output). It is a control signal for read operation. It is an active low signal.

READY (input). It is a signal sent by a peripheral to the microprocessor. It is active high. If READY is inactive, i.e., it is low it means that the accessed device is not ready. In such a situation a wait state is introduced. When READY is high data transfer takes place.

RESET (input). System reset. It is an active high signal.

CLK (input). Clock 5, 8 or 10 MHz.

INTR (input). Interrupt request. Active high signal. It is a level-triggered maskable interrupt.

NMI (input). It is an edge-triggered, non-maskable interrupt.

TEST (input). Wait for test control. If it is low execution continues, otherwise the microprocessor waits in an idle state till **TEST** goes low. This signal is used to allow the 8086 to test coprocessor status. The test is required when floating-point instruction is encountered. The CPU starts up the coprocessor. Then the CPU executes other instructions of the program. When CPU requires the result of the floating-point operation, it tests coprocessor to check whether the coprocessor has completed the operation. If the coprocessor has not completed the operation, the CPU waits for the result. It is connected to BUSY pin of 8087. When 8087 is executing an instruction, its BUSY pin remains high. The BUSY pin becomes low when 8087 completes the execution of the instruction.

V_{cc} power supply, 5 V_{d.c.}

GND. Ground.

Signals for Minimum Mode Operation

When MN/**MX** is high the 8086 operates in the minimum mode. The signals issued through the pins 24 to 31 as shown in the brackets against these pins in Fig. 5.11, are for the minimum mode of operation. Their description is as follows:

INTA (output) Pin 24. It is an interrupt acknowledge signal.

ALE (output) Pin 25. Address latch enable. It goes high during T₁.

DEN (output). Pin 26. It is for data enable when 8286/8287 bus transceiver are employed. It acts as output enable. It is active low.

DT/R (output). Pin 27. Data transmit/receive. When 8286/8287 is employed this signal controls the direction of data flow. When it is high data is transmitted. When it is low data is received.

M/IO (output). Pin 28. It is a status signal to control memory or I/O access. When it is high the CPU reads the memory or writes into the memory. When it is low, the CPU can access I/O device.

WR (output). Pin 29. It is a control signal for write operation. It is active low.

HLDA (output). Pin 30. It is the HOLD acknowledge signal. It is sent by the 8086 after receiving HOLD signal. It is active high. It goes low when HOLD is removed.

HOLD (input). Pin 31. When another external device in the system wants to use the address and data bus, it sends the HOLD request to the CPU through this pin. It is an active high signal.

Signals for Maximum Mode Operation

When MN/**MX** is low the 8086 operates in the maximum mode. The signals shown by the side of pins 24 to 31 in Fig. 5.11, are for the maximum mode of operation. In the maximum mode of operation the control signals such as **DEN**, **DT / R**, **ALE**, **INTA**, memory read, memory write, I/O read, I/O write etc. are not available directly from the 8086. These are available through the bus controller 8288.

QS₀, QS₁ (output). Pins 24 and 25. These are instruction queue status. Their logics are given below:

QS₁ QS₀

0	0	No operation
0	1	1st byte of the opcode from the queue
1	0	empty the queue
1	1	subsequent byte from the queue.

S₀, S₁, S₂ (output). Pins 26, 27 and 28. Status signals. These signals are applied to the bus controller, 8288. The bus controller generates control signals as shown below:

S ₂	S ₁	S ₀	
0	0	0	Interrupt acknowledge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	HALT
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive state

LOCK output. Pin 29. It is active low. When it is low all interrupts are masked and no HOLD request is granted.

RQ/GT₀, RQ/GT₁ (bidirectional). Pin 30, 31. These are request/grant lines. They are used by other devices to request 8086 to release address, data and control buses. They are similar to HOLD and HLDA pins in the minimum mode. RQ / GT₀ has higher priority than RQ / GT₁.

**THIS PAGE
IS BLANK**

BIBLIOGRAPHY

1. B. Ram, *Fundamentals of Microprocessors and Microcomputers*, 6th edition, Dhanpat Rai Publications, New Delhi, 2005.
2. D.V. Hall, *Microprocessors and Interfacing: Programming and Hardware*, TMH 2nd edition, 1992.
3. Hussain and Hussain, *Computers: Technology, Applications and Social Implications*, Prentice Hall of India, New Delhi, 1989.
4. D.H. Sanders, *Computer Today*, 3rd edition, McGraw-Hill, Singapore, 1988.
5. T.C. Bartee, *Digital Computer Fundamentals*, 6th edition, McGraw-Hill, Singapore, 1985.
6. J.P. Hayes, *Computer Architecture and Organisation*, 3rd edition, McGraw-Hill, Singapore, 1998.
7. J.P. Hayes, *Digital System Design and Microprocessors*, McGraw-Hill, Singapore, 1985.
8. M.R. Gore and J.W. Stubbe, *Computers and Information Systems*, 2nd edition, McGraw-Hill, Singapore, 1984.
9. S.C. Chapra and R.P. Canale, *Introduction to Computing for Engineers*, McGraw-Hill, Singapore, 1986.
10. D.V. Hall, *Microprocessors and Digital Systems*, 3rd edition, McGraw-Hill, Singapore, 1994.
11. M. Morris Mano, *Computer System Architecture*, 2nd edition, Prentice-Hall of India, New Delhi, 1982.
12. Joseph J.F. Cavanagh, *Digital Computer Arithmetic*, McGraw-Hill, Singapore, 1985
13. D.D. Givone and R.P. Rosser, *Microprocessors/Microcomputers: An Introduction*, McGraw-Hill, Singapore, 1980.
14. A.P. Malvino, *Digital Computer Electronics—An Introduction to Microcomputers*, 2nd edition, Tata McGraw-Hill, New Delhi, 1983.
15. A.P. Malvino and D.P. Leach, *Digital Principles and Applications*, 4th edition, McGraw-Hill, Singapore, 1986.
16. C.E. Strangio, *Digital Electronics—Fundamental Concept and Applications*, Prentice-Hall of India, New Delhi, 1980.

17. T.R. Padmanabhan, *Digital Systems and Microprocessors*, Tata McGraw-Hill, New Delhi, 1983.
18. M.P. Marcus, *Switching Circuits for Engineers*, 2nd edition, Prentice-Hall of India, New Delhi, 1962.
19. S.S. Alam and S.K. Sen, *Computer and Computing with FORTRAN 77*, Oxford and IBH, New Delhi, 1988.
20. James L. Turley, *PC Made Easy*, Osborne/McGraw-Hill, Berkely, California, 1989.
21. John F. Wakerly, *Microcomputer Architecture and Programming 68000 Family*, John Wiley and Sons, New York, 1989.
22. Peter Norton and Robet Jourdan, *The Hard Disk Companion*, Prentice Hall of India, New Delhi, 1989.
23. Peter Norton, *Inside the IBM PC*, Prentice Hall of India, New Delhi, 1997.
24. William Stallings, *Computer Organisation and Architecture*, 6th edition, PHI, 2003.
25. Thom Luce, *Computer Hardware, System Software, and Architecture*, McGraw-Hill, Singapore, 1989.
26. Mohamed Rafiquzzaman, *Microprocessors and Microcomputer-Based System Design*, Universal Book Stall, New Delhi, 1990.
27. Andrew S. Tanenbaum, *Structural Computer Organisation*, 4th edition, Prentice Hall of India, New Delhi, 1999.
28. Donald D. Voisinet, *Computer-Aided Drafting and Design*, McGraw-Hill, Singapore, 1987.
29. Donald Hearn and M. Pauline Baker, *Computer Graphics*, 2nd edition, Prentice-Hall of India, New Delhi 1994.
30. Edward M. Baras, *Guide to Lotus Using 1-2-3*, 2nd edition, Osborne/McGraw-Hill, Singapore, 1988.
31. Que Corporation Carmel, Indiana, *Using 1-2-3, special edition*, TECH Publication, Singapore, 1987.
32. Robert T. Grauer and Paul K. Sugrue, *Microcomputer Applications*, McGraw-Hill, Singapore, 1987.
33. Milan Milenkovic, *Operating Systems-Concepts and Design*, McGraw-Hill, Singapore, 1987.
34. A.S. Tanenbaum, *Operating Systems, Design and Implementation*, Prentice Hall of India, New Delhi, 1987.
35. Hwang K. and Briggs F.A., *Computer Architecture and Parallel Processing*, McGraw-Hill, Singapore, 1984.
36. V. Rajaraman, *Elements of Parallel Processing*, Prentice-Hall of India, New Delhi, 1990.
37. R. Rajagopalan, *Understanding Computers*, Tata McGraw-Hill, New Delhi, 1987.
38. Hamcher, Vranesic and Zaky, *Computer Organisation*, 5th edition, McGraw-Hill, 2002.
39. Peter Norton and Richard Wilton, *Programmer's Guide to the IBM PC and PS/2*, Microsoft Press, Washington, 1988.

40. S. Dasgupta, *Computer Architecture: A Modern Synthesis*, Vol. 1 and 2, John Wiley and Sons, Singapore, 1989.
41. Intel's *Microprocessor and Peripheral Handbook*, Vol. I and II, Santa Clara, U.S.A., 1989.
42. Intel's *Microprocessors Handbook*, Santa Clara, U.S.A., 1990, 1997.
43. Intel's *Memory Component Handbook*, Santa Clara, U.S.A., 1990.
44. C.H. Pappas and W.H. Murry III, *80386 Microprocessor Handbook*, Osborne/McGraw-Hill, Berkeley, California, 1988.
45. James T. Perry and J.G. Lateer, *Understanding Oracle*, B.P.B. Publication, Delhi, 1989.
46. Douglas Cobb, *Mastering Symphony*, 4th edition, B.P.B. Publication, Delhi, 1988.
47. Nelson & Jhonson, *AutoCAD: The Complete reference*, Osborne McGraw-Hills, 1989.
48. Public Telecommunication Services, *Philips Professional Journal*, p. 17, No. 4, 1983.
49. 64 Bit Microprocessors, p. 61, *Electronics For You*, New Delhi, July, 1991.
50. Building and Managing an Integrated Project Support Environment, *Hewllet Packard Journal*, Palo Alto, California, June, 1991, p. 90.
51. 386 PC Applications, *Computers Today*, New Delhi, January, 1992.

**THIS PAGE
IS BLANK**

INDEX

A

- Accounting, 9.4
- Accumulator, 1.8, 5.3
- Accumulator-based machine, 5.7
- Access, 8.43
- Adder, 4.43, 4.44
- Address bus, 1.12
- Addressing modes, 5.8, 5.34
 - Based addressing, 5.11, 5.35
 - Based indexed addressing, 5.11, 5.35
 - Direct or absolute addressing, 5.9, 5.35
 - Immediate addressing, 5.10, 5.34
 - Implicit addressing, 5.10
 - Indexed addressing, 5.11, 5.35
 - Paged addressing, 5.10
 - Register addressing, 5.9, 5.34
 - Register indirect addressing, 5.9, 5.35
 - Relative addressing, 5.11
 - Relative indexed addressing, 5.11
 - Stack addressing, 5.11
- ADSL, 1.28
- AGP, 10.3
- Airline reservation 9.17
- ALE, 5.13
- ALGOL, 8.3
- Algorithm, 8.20
- Alpha microprocessor, 5.57
- ALU, 1.7, 5.1
- Alta Vista, 1.27
- Applet, 1.27
- Apple's Mac OS, 8.31
- Application packages or application programs, 8.32
- Application of computers, 9.1
- APT, 8.7
- AMD processor, 5.58
- AND gate, 3.1
- Arithmetic circuits, 4.43
 - Adder-subtractor, 4.46
 - Full adder, 4.44
 - Half adder, 4.43
 - High-speed addition and subtraction, 4.47
 - Multiplication, 4.48
 - n*-bit adder, 4.46
- Arithmetic and logic unit (ALU) 1.7, 5.1
- ARM processor, 5.60
- Arithmetic processor, 7.49
- Array processors, 11.3
- Artificial intelligence, 9.18
- ASCII codes, 2.27
- Assembler, 1.22, 8.2
 - Cross assembler, 8.2
 - One-pass assembler, 8.2
 - Self assembler, 8.2
 - Two-pass assembler, 8.2
- Assembly language, 1.22, 8.2
- Associative memory, 6.3
- Asynchronous data transfer, 7.36

- ATA, ATAPI, 6.16
Atholon, 5.58
Auxiliary carry (AC), 5.4
Auxiliary memory, 1.9, 6.1
- B**
- Back-up memory, 6.1
Basic, 8.4
Base, 5.10
Based addressing, 5.11, 5.35
Based indexed addressing, 5.11, 5.35
Base register, 5.25
Batch processing, 1.23
Baud rate, 7.28
BCD, 2.17
Binary addition, 2.6
Binary counter, 3.19
Binary number system, 2.2
Binary subtraction, 2.7
Binary subtraction using 2's complement, 2.12
Binary to decimal conversion, 2.3
Binary to hexadecimal conversion, 2.22
Binary to octal conversion, 2.25
BIOS, 8.45
Bit, 1.1, 1.13
Bit-mapped raster scan, 7.15
Bluetooth, 1.30
Boolean algebra, 4.1
Boolean expression (function), 4.5
Boolean operations, 4.5
Boolean postulates, 4.6
Boolean theorems, 4.6
Boolean variables, 4.5
Booting, 8.26
Bottom-up design, 8.16
Bottom-down design, 8.16
Bootstrapping, 8.26
Break point, 8.11
Bridge, 1.28
Broadband, 1.28
- Buffers, 3.18
Bulletin board system, 9.11
Burst mode of DMA data transfer, 7.35
Buses, 1.12
- Address bus, 1.12
 Bus width, 1.13
 Control bus, 1.12
 Data bus, 1.12
- Byte, 1.1, 1.13
- C**
- Cache coherence, 11.7
Cache controller, 6.2
Cache memory, 1.9, 6.2
CAD, 8.39, 9.8
CARRY stayus (CS), 5.4
CAM, 8.39, 9.7
Canonical form of logic expression, 4.15
Cathode ray tube (CRT), 7.12
CCD, 6.25, 9.24
CC-NUMA, 11.7
CDMA, 1.29
CD-ROM, CD-R, CD-R/W, 6.22
Celeron processor, 5.51
Cebl processor, 5.60
Central processing unit (CPU), 1.7, 5.1
Centrino technology, 1.19
Character, 2.27
Character printer, 7.20
Chipsets, 7.34
CIM, 9.9
CISC processor, 1.31, 5.61
C++, C language, 8.5
Client, 1.26
CLK (clock), 5.15
Cluster, 11.9
Code converter, 3.18
COBOL, 1.23, 8.5
Combinational circuits, 4.42
Command, 1.23

- Compiler, 1.23, 8.8
 Complements, 2.8
 1's complement, 2.10
 2's complement, 2.10
 9's complement, 2.8
 10's complement, 2.8
 Complement of Boolean expression, 4.13
 Computer, 1.1
 Analog computer, 1.2
 Digital computer, 1.1
 Evolution of, 1.2
 Generation of, 1.5
 Mainframe (large) computer, 1.20
 Microcomputer, 1.20
 Minicomputer, 1.20
 Personal computer, 1.19
 Processing speed of 1.14
 Super computer, 1.20
 Wordlength of 1.13
 Computer aided design (CAD), 8.39, 9.8
 Computer aided drafting (CAD), 8.39, 9.8
 Computer aided manufacture (CAM), 8.39, 9.8
 Computer Applications, 1.32
 Computers in communications, 9.10
 Computer conferencing, 9.11
 Computer generation, 1.5
 Computer graphics, 8.39, 9.5
 Computer network, 1.24
 Computer pen, 1.21
 Computer vision, 9.23
 Concurrent programming, 8.22
 Control flow and data flow computers, 1.30
 Control unit, 1.8, 5.2
 Counters, 3.19
 Asynchronous (or ripple) counter, 3.19
 BCD counter, 3.25
 Binary counter, 3.19
 Decade counter, 3.25
 Modulus of a counter, 3.20
 Mode-3, Mode 6 and Mod 12 counters, 3.22
 Mod-10 counter, 3.25
 Presetable counter, 3.26
 Ring counter, 3.27
 Ripple counter, 3.19
 Shift counter, 3.28
 Synchronous counter, 3.22
 CPU, 1.7, 5.1
 Cross assembler, 8.2
 Crossbar interconnection, 10.7
 Cross compiler, 8.8
 CRT Display, 7.12
 CRT controller, 7.16

D

- Data analysis packages, 8.35, 9.4
 Database, 8.41
 Database management, 8.41
 Data bus, 1.12
 Data elements, 8.41
 Data items, 8.41
 Data flow Architecture, 1.30, 5.49
 Data flow computer, 1.30, 5.48
 Data register, 5.3
 Data selector, 3.29
 Data transfer schemes, 7.34
 Asynchronous data transfer, 7.36
 DMA data transfer, 7.35
 Interrupt driven data transfer, 7.38
 Synchronous data transfer, 7.36
 DCE, 7.29
 Debugging, 8.11
 Decimal number system, 2.1
 Decimal to binary conversion, 2.4
 Decimal to hexadecimal conversion, 2.22
 Decimal to octal conversion, 2.25
 Decoders, 3.18, 3.29
 D flip-flop, 3.15
 De Morgan's Theorem, 4.8
 Descriptors, 5.43, 6.27
 Demultiplexers (digital), 3.29

- Design package, 9.6
Desktop computer, 1.19
Despatcher, 8.26
Destructive and nondestructive readout, 1.9, 6.10
Device polling, 7.39
Dhrystone, 1.14
DIMM, 6.6
Digital computers, 1.1
 Classification of, 1.18
 Evolution of, 1.2
 Generation of, 1.5
Digital multiplexers, 3.29
Digital signal processor, 7.47
Digitizers, 9.7
Direct access storage devices (DASDS), 6.26
Direct addressing, 5.9, 5.35
Direct memory access (DMA), 7.34, 7.35
Displacement, 5.10
Display, 7.12
 CRT display, 7.12
 LED display, 7.18
 LCD display, 7.17
 Plasma display, 7.18
Distributed processing, 1.24
DIVX, 1.36
DMA controller, 7.41
DMA data transfer, 7.35
Documentation, 8.19
Don't care conditions, 4.41
Dot-matrix printer, 7.20
Drivers, 3.18
DSL, 1.28
DTE, 7.29
DTH, 1.28
DTP, 7.26
Dual of Boolean Expression, 4.13
Dual bus, 10.7
DVD, 6.24
Dvorak keyboard, 7.2
Dynamic RAM, 6.5
Dynamic RAM controller, 6.9
- E**
- EBCDIC codes, 2.30
Editor, 8.32
EDGE, 1.29
EDO RAM, 6.6
EEPROM, 6.8
EIDE, 6.16
EISA bus, 10.1
E-mail, 9.11
Electronic office, 9.1
Electronic spreadsheet, 8.35
Electronic thesaurus, 8.34
Embedded processor, 7.46
Encoders, 3.18
EPIC processor, 5.53
EPROM, 6.7
Eraser, 7.7
Error checking, 6.10, 6.19
Ethernet, 1.25
Execute cycle, 5.16
Exceptions, 5.13
Excel, 8.38
EXCLUSIVE-OR gate, 3.5
EXCLUSIVE-NOR gate, 3.5
Expert system, 9.18
Exponent, 2.34
Extranet, 1.28
- F**
- Fan-in, 3.7
Fan-out, 3.7
Fetch cycle, 5.16
Fields, 8.41
File, 8.41
File manager, 8.32
Firmware, 1.21, 8.1
Flags, 5.4
Flash memory, 6.8

- Flat-bed plotter, 7.24
 Flinn's classification of computers, 11.11
F
 Flip-flops, 3.8
 D flip-flop, 3.15
 J-K flip-flop, 3.11
 Level-triggered flip-flop, 3.14
 S-R flip-flop, 3.8
 T flip-flop, 3.17
 +ve-edge-triggered flip-flop, 3.13
 -ve-edge triggered flip-flop, 3.14
 Floating-point representation, 2.34
 Floppy disks, 1.11, 6.17
 Flowchart, 8.19
 Formatting of magnetic disks, 6.19
 Front page, 8.45
 FORTNAN, 1.23, 8.5
 Free ware, 1.22
 FTP, 1.27
 Full duplex, 7.28
- G**
- Gates, 3.1
 AND gates, 3.1
 EXCLUSIVE-OR gates, 3.5
 NAND gates, 3.3
 NOR gates, 3.4
 NOT gate, 3.3
 OR gates, 3.2
 Tri-state gagte, 3.7
 Gateways, 1.28
 General purpose registers, 1.8
 General registers machines, 5.8
 Global memory, 11.4
 Gopher, 1.27
 GPRS, 1.28
 Graphics packages, 8.39
 Graphic tablet, 9.7
 Graphics processor, 7.51
 Graphics tools, 9.7
 Gray codes, 2.30
- Groupware, 9.12
 GSM, 1.28
 GUI, 1.21
- H**
- Half-duplex, 7.28
 Handheld computer, 1.18
 Hand shaking mode, 7.36
 Hand shaking signal, 7.36
 Hard disks, 1.11, 6.14
 Hardware, 1.21
 Hardware interrupt, 5.12
 Hexadecimal number system, 2.19
 Hexadecimal to binary conversion, 2.23
 Hexadecimal to decimal conversion, 2.21
 Hexadecimal to octal conversion, 2.26
 High-level languages, 1.23, 8.3
 IGL to 4GL, 8.7
 ALGOL, 8.3
 APT, 8.7
 BASIC, Visual BASIC, 8.4
 C, C++ language, 8.5
 FORTRAN, 8.5
 JAVA, JAVASCRIPT, 8.6
 LISP, 8.7
 LOGO, 8.7
 PASCAL, 8.3
 PROLOG, 8.6
 SQL, 8.6
 SNOBOL, 8.7
 HLDA, 5.14
 HOLD, 5.14
 Home page, 1.26
 HTML, 1.26
 Hub, 1.27
- I**
- iCOMP, 1.14
 Icon, 1.21
 IDE, 6.16

- IEEE-488, 10.6
IEEE-1284, 10.6
IEEE-1394, 10.5
Image-scan digitizer, 9.7
Immediate addressing, 5.10
Impact of computers on society, 1.33
Implicants, 4.34
 Nonprime implicants, 4.34
 Prime implicants, 4.34
Implicit addressing, 5.10
INCLUSIVE-OR gate, 3.2
Index, 5.10
Indexing, 8.42
Indexed addressing, 5.11, 5.35
Index register, 5.4, 5.25
INFORMIX, 8.43
INGRESS, 8.43
Inhibitor (inverter), 3.3
Ink-jet printer, 7.22
Input devices, 1.12, 7.1
Instant messaging, 9.12
Instruction, 1.22, 5.4, 8.1
Instruction types, 5.6, 5.7
Instruction and data flow, 5.17
Instruction cycle, 5.16
Instruction pointer, 5.26
Instruction register, 5.4
INTA, 5.14
Integrated package, 8.43
Intel 8044, 7.46
Intel 8048, 8748, 1.18, 7.45
Intel 8051, 1.18, 7.45
Intel 8085, 1.15, 5.13, 5.24
Intel 8086, 1.15, 5.24
Intel 80486, 1.15, 5.37
Intel 8087, 80287, 80387, 7.50
Intel 8088, 80188, 1.15, 5.36
Intel 8096, 1.18, 7.45
Intel 8231, 7.50
Intel 8237A, 7.41
Intel 8251, 7.31,
Intel 8253, 8254, 7.33, 7.34,
Intel 8255, 7.26
Intel 8257, 7.44
Intel 8259, 7.39
Intel 8275, 8276, 7.16
Intel 8279, 7.18
Intel's Pentium, 1.15, 5.47
Intel's Pentium Pro, 1.15, 5.48
Intel's Pentium 4, 1.15, 5.52
Intel's Itanium, 5.33
Internet, 1.25
Internet Explorer (IE), 1.26
Intranet, 1.28
Interpreter, 1.23, 8.8
Interrupt, 5.12
 Hardware interrupt, 5.12
 Software interrupt, 5.13
Interrupt controller, 7.39
Interrupt disable, 5.12
Interrupt driven data transfer, 7.38
Interrupt enabling, 5.12
Interrupt masking, 5.13
Interrupt service subrouting (ISS), 5.12
INTR, 5.14
Investment package, 9.4
Inventory control, 9.4
IO/ $\overline{\text{S}}$, 5.13
I/O ports, 7.25
I/O processor, 7.47
 $\overline{\text{CE}}\ \overline{\text{R}}$, 5.16
I/O read cycle, 5.21
 $\overline{\text{CE}}\ \overline{\text{W}}$, 5.16
I/O write cycle, 5.23
ISA bus, 10.1
ASCII, 2.27
ISP, 1.27
ISS, 5.12
Itanium, 5.33

J

JAVA, 1.23, 8.5
 Java Script, 8.6
 J-K flip-flop, 3.11
 Master-slave J-K flip-flop, 3.14
 Joystick, 7.7

K

Karnaugh map, 4.23
 Don't care conditions, 4.41
 for two variables, 4.24
 for three variables, 4.26
 for four variables, 4.31
 K-map for product of sums, 4.39
 Labeling of, 4.35
 Product of sums for K-Map, 4.38

Kernel, 8.26

Keyboard, 7.1

Keyboard/display interface, 7.18

KLIPS, 1.14

L

LAN, 1.25
 Laser printer, 7.22
 Latch, 3.17
 LCD display, 7.17
 LED display, 7.18
 7-Segment display, 7.19
 18-Segment display, 7.19
 Light pen, 7.5
 Line printers, 7.20
 Linker, 8.32
 LINPACK rating, 1.14
 LINUX, 1.22, 8.28
 Liquid crystal display, 7.17
 LISP, 8.7
 Loader, 8.32
 Local area network (LAN), 1.25
 Locator, 8.32

LOCK, 5.31

સ્થાનકુલી, 5.31

Logic analyser, 8.12
 Logical address, 5.43, 6.27

Logical gates, 3.1
 AND gate, 3.1
 EXCLUSIVE-OR gate, 3.5

NAND gate, 3.3

NOR gate, 3.4

NOT gate, 3.3

OR gate, 3.2

LOGO, 8.7

Loosely coupled multiprocessor, 11.4

LOTUS 1-2-3, 8.35

Low-level language, 1.23, 8.3

LSI, 1.4

M

Machine cycle, 5.17
 Machine language, 1.22, 8.2, 8.4
 Macro, 8.13

Magnetic-ink character reader (MICR), 7.9

Magnetic memory, 1.10, 6.11

 Floppy disks, 1.11, 6.17

 Hard disks, 1.11, 6.14

 Magnetic tapes, 1.11, 6.19

Mail-merge, 8.34

Mainframe computer, 1.20

Main memory, 6.1

Mantissa, 2.34

MAR (Memory Address Register), 5.4

Markup language, 1.26

Masking, 5.13

Master-slave J-K flip-flop, 3.14

Maxterm, 4.15

MBR (Memory Buffer Register), 5.4

Macro, 8.12

Memory, 1.8, 6.1

Main memory, 1.8, 6.1

Backup memory, 6.1

- Cache memory, 1.9, 6.2
 Destructive and nondestructive readout, 1.9, 6.10
 Magnetic memory, 1.10
 Magnetic tape, 1.11, 6.19
 Optical memory, 1.11, 6.21
 Primary (Main) memory, 1.8, 6.1
 Program and data memory, 6.26
 Ral and virtual memory, 1.9, 6.4
 Secondary (Auxiliary) memory, 1.9, 6.1
 Semiconductor memory, 1.10, 6.5
 Memory address register (MAR), 5.4
 Memory addressing capacity of CPU, 1.13
 Memory buffer register (MBR), 5.4
 Memory controller, 6.9
 Memory dumps, 8.12
 Memory management unit (MMU), 1.10, 6.26
 Memory read cycle, 5.20
 Memory write cycle, 5.22
 _____, 5.15
 Menu, 1.21
 _____, 5.16
 MESI, 11.9
 MFDS, 7.24
 MICR (Magnetic-ink Character Reader), 7.9
 Microcomputer, 1.20
 Microcontroller (Single-chip microcomputer), 1.18, 7.45
 Microfiche, 7.25
 Microfilm, 7.25
 Microprocessor, 1.15, 5.24
 Microprogramming, 1.31
 MIMD, 11.11
 Minicomputer, 1.20
 Minterm, 4.15
 MIPS, 1.14
 MIPS microprocessor, 5.57
 MFLOPS, 1.14
 Middleware, 1.22
 MISD, 11.11
 MMU, 6.27
 Mnemonics, 1.22
 Modem, 7.29
 Modular programming, 8.14
 Monitor, 1.12
 Motorola processors, 5.59
 Mouse (Puck), 7.6
 MP3, 1.35
 MPEG, 1.35
 MS-DOS, 1.22
 MS-Office, 8.44
 MSI, 1.4
 MS-Word, 8.44
 MS-Powerpoint, 8.45
 MS-Excel, 8.38
 MS-Access, 8.43
 MS-Outlook, 8.45
 Multicomputer system, 11.9
 Multifunction devices (MFDs), 7.24
 Multiplexers, 3.29
 Multiplier circuit, 4.48
 Multiport memory, 11.6
 Multiprocessing, 1.24
 Multiprocessor system, 11.4
 Loosely coupled, 11.4, 11.10
 Distributed-memory type, 11.4, 11.10
 Tightly coupled, 11.4, 11.5
 Shared memory type, 11.4, 11.5
 Multiprogramming, 1.23, 8.22
 Miltitasting system, 1.24, 8.23
 Multiuser (or time-share) system, 8.23
 Multithreading, 1.24
- N**
- NAND gate, 3.3
 Negative number representation, 2.8
 Negative-edge-triggered flip-flop, 3.14
 Netscape Navigator, 1.26
 Neuman computer, 1.30, 5.48
 Nine's complement, 2.8
 Non-impact printer, 7.20

Non prime implicants, 4.34
 Nonvolatile RAM, 6.9
 NOR gate, 3.4
 Normalized floating-point number, 2.35
 Notebook computer, 1.19
 NOT gate, 3.3
 NUMA type multiprocessor system, 11.7

O

Object Oriented Programming, 8.17
 Object code, 8.4
 Octal number system, 2.23
 Octal to binary conversion, 2.26
 Octal to decimal conversion, 2.24
 Octal to hexadecimal conversion, 2.27
 Office automation, 1.32
 Offset, 5.10
 OLD display, 7.10
 One-address instruction, 5.7
 One-address machine, 5.7
 One-byte instruction, 5.6
 One's complement, 2.10
 Online applications, 9.21
 OOP, 8.5, 8.17
 Opcode, 5.5
 Opcode fetch cycle, 5.16
 Operand, 5.5
 Operating system, 1.22, 8.1, 8.25
 LINUX, 1.22, 8.28
 MS-DOS, 1.22
 OS/2, 1.22
 RMX-86, 8.24
 Solaries, 1.22, 8.30
 Mac OS, 8.31
 UNIX, 1.22, 8.26
 WINDOWS-98, 1.22, 8.29
 WINDOWS-XP, 8.29
 WINDOWS-2003, 8.30
 XENIX, 8.26
 Optical disk, 1.11, 6.21

Optical bar-code reader, 7.9
 Optical mark reader, 7.9
 Optical scanner, 7.8
 Optic RAM camera, 9.23
 OR gate, 3.2, 4.2
 ORACLE, 8.43
 OS/2, 1.22
 Organisation of computer, 1.7
 Output devices, 1.12, 7.11
 Outlook, 8.45
 Overflow, 2.39

P

Packet, 1.27
 Paged addressing, 5.11
 Pagemaker, 9.3
 Paint packages, 9.6
 Palmtop computer, 1.18
 PAN, 1.30
 Parallel ports, 7.27
 Parallel processing, 11.1
 Array processor, 11.3
 Multiprocessor system, 11.4
 Pipelining, 11.1
 Temporal parallelism, 11.3
 Parallel search memory, 6.3
 Parity status, 5.4
 PASCAL, 1.23
 PC, 1.19
 PCI bus, 10.3
 PCI Express, 10.3
 PCMCIA, 6.29
 Pentium, 5.47
 Pentium Pro, 5.48
 Pentium 4, 1.15, 5.52
 Peripheral, 7.1
 Personal computers, 1.19
 Personal Digital Assistant (PDA), 1.18
 Physical or real memory, 1.9, 6.4
 Pipelined processor, 11.1

- Pipelining, 11.2
Pipelined cycle, 11.2
Plasma display, 7.18
Plotters, 7.24
 Drum plotter, 7.24
 Flat-bed plotter, 7.24
 Ink-jet plotter, 7.24
 Microgrip plotter, 7.24
PMP, 1.36
Polling, 7.39
POP operation, 8.8
Pointing device, 7.1
Pointing stick, 7.7
Point of sale terminal, 9.21
Port, 7.25
Power PC, 5.54
Preset and clear, 3.13
Primary memory, 1.8, 6.1
Prime implicants, 4.34
Printers, 7.19
 Character printer, 7.20
 Impact printer, 7.20
 Laser printer, 7.22
 Ink-jet printer, 7.22
 Line printer, 7.20
 Nonimpact printer, 7.20
 Page printer, 7.20
Procedure oriented language, 8.1, 8.5, 8.17
Process, 8.26
Processing speed of computer, 1.14
Product of sums form of logic expression, 4.15
Program 1.21, 8.1
Program counter (PC), 5.3
Program design, 8.14
 Modular programming, 8.14
 Structured programming, 8.15
 Top-down and bottom-up design, 8.16
Program testing, 8.19
Project management, 9.9
PROLOG, 1.23, 8.6
PROM, 6.7
Protection, 6.28, 8.24
PSW, 5.5
PUCK (mouse), 7.6
PUSH operation, 8.8
- Q**
- Quine-McCluskey method, 4.42
QWERTY type key layout, 7.2
- R**
- Railway ticket reservation, 9.17
RADIUS, 1.28
RAM, 6.5
 Dynamic RAM, 6.5
 Dual ported RAM, 6.6
 RDRAM, 6.6
 EDO RAM, 6.6
 VRAM, 6.6
 WRAM, 6.6
 DDR RAM, 6.6
 Nonvolatile RAM, 6.9
 Static RAM, 6.5
 SDRAM, 6.6
 SGRAM, 6.6
Raster scan, 7.12
RDBMS, 8.43
READY signal, 5.15
Real (or floating-point) numbers, 2.34
Real (or physical) memory, 1.9, 6.4
Real-time applications, 9.21
Record, 8.41
Relative addressing, 5.11
Relative indexed addressing, 5.11
Registers, 3.17, 5.3
RESET signal, 5.15
Ring counter, 3.27
Ripple counter, 3.19
RISC processor, 1.31
RMX-86, 8.24

- Robot, 9.18
ROM, 1.10, 6.8
Routers, 1.28
RS-232C, 7.29
RS-422A, RS-423A; 7.30
RS-449, RS-366; 7.30
RST instructions, 5.13
RST 5.5, RST 6.5, RST 7.5; 5.15
- S**
- Sabdratna, 8.35
Satellite communication, 9.14
Scanners, 7.8
Scheduler, 8.26
Scheduling, 8.25
SCSI, 6.16
SDRAM, 6.6
Search engine, 1.27
Secondary memory, 1.9, 6.1
Self assembler, 8.2
Segment, 5.12
Segment register, 5.12, 5.25
Semaphor, 5.30
Semiconductor memory, 6.5
Flash memory, 6.8
Nonvolatile RAM, 6.7, 6.8
PROM, EPROM, EEPROM, 6.7, 6.8
RAM, 6.5
ROM, 6.7
Serial data transfer, 7.28
Baud rate, 7.28
Full duplex, 7.28
Half duplex, 7.28
Simplex mode, 7.28
Serial port, 7.27
Sequential circuits, 4.42
Servers, 1.20, 1.26
SGRAM, 6.6
Shared memory type multiprocessor system, 11.4, 11.5
Shift register, 3.17
- SID, 5.15
SIMM, 6.6
Single-bus system, 10.6
Single-chip microcomputer, 1.20, 7.45
Intel 8048, 1.20, 7.45
Intel 8051, 1.20, 7.45
Intel 8096, 1.20, 7.45
Intel 8748, 1.20, 7.45
Single step control, 8.11
Signed and unsigned numbers, 2.12
Simplex mode, 7.28
Simulator, 8.12
SIMD, 11.11
Single-byte instruction, 5.6
SISD, 11.11
Smart products, 9.22
SMP System, 11.5
SNOBOL, 8.7
Snoopy cache system, 11.8
SOD, 5.15
Software, 1.21, 8.1
Application programs, 1.21
Firmware, 1.21, 8.1
Middleware, 1.22
Operating system, 1.21, 8.1
System software, 1.21, 8.1
Utility programs, 8.1
Software development, 8.18
Software interrupt, 5.13
Source language, 8.4
SPEC int, 1.14
SPEC fp, 1.14
Spreadsheet, 8.35
SQL, 8.6
S-R flip-flop, 3.8
Stack, 5.3, 8.8
Stack addressing, 5.11
Stack pointer (SP), 5.4, 5.25
Stacktop, 8.8
Status flag, 5.3

Status register, 5.4
Stack-oriented machine, 5.8
State, 5.17
Statement, 1.23, 8.3
Structured programming, 8.16
SYBASE, 8.43
Symmetric multiprocessor System, 11.5
SYMPHONY, 8.43
Synchronous data transfer, 7.36
Subroutine, 8.9
Subtracter, 4.47
Sum of products form of logic expressions, 4.15
Supervisor, 8.26
Supercomputer, 1.18
Swapping, 6.4, 6.27

T

TCP/IP, 1.27
Teleconferencing, 9.11
Telnet, 1.27
Telex system, 9.13
Templates, 8.38
Ten's complement, 2.10
Text editor, 8.32
T flip-flop, 3.16
TFT LCD Moniter, 7.17
Thesaurus, 8.34
Three-address instruction, 5.7
Three-address machine, 5.8
Three-byte instruction, 5.6
Throughput, 1.14
Tightly coupled multiprocessor, 11.4
Timing and control unit, 1.8, 5.2
Timer/counter, 7.33
Time-Shard system, 1.24
Timing diagram, 5.18
Token, 1.25
Touch screen, 7.10
Top-down and bottom-up design, 8.16
TPS, 1.14
Trace routines, 8.12
Trakballs, 7.8
Trackpads, 7.8
Tristate logic, 3.7
Truth table, 3.2
Two-address instruction, 5.7
Two-address machines, 5.8
Two-byte instruction, 5.6
Two's complement, 2.10
Two-key lockout, 7.5
Two-key rollover, 7.5

U

UART, 7.29
ULSI, 1.4
ULTRASPARC, 5.58
Ultrasonic vision, 9.25
UMA type multiprocessor, 11.5
UMATS, 1.29
Underflow, 2.39
UNIFY, 8.43
Uniprocessor system, 11.4
Universal gate, 3.6
Unsigned number, 2.12
UNIX, 1.22, 8.26
Up counter, 3.26
Up-down counter, 3.26
UPI-41, UPI-42, 7.40
UPI-452, 7.39
URL, 1.26
USART, 7.29
USB, 10.5
Usenet, 1.26
User interface, 1.21
Utility programs, 8.1, 8.31

V

Validation, 8.11
Vector computer, 11.4
Vector scan, 7.12

INDEX

- Vectored interrupt, 7.39
 Vector processor, 11.4
 Ventura, 9.3
 Verification, 8.11
 Vidieo comera, 9.24
 Video conferencing, 1.33, 9.11
 Video monitor, 1.12
 Virus, 1.35
 Virtual address, 6.28
 Virtual memory, 1.9, 6.4, 6.28
 Visual BASIC, 8.4
 VLSI, 1.4
 Voice recognition, 7.9
 Voice response, 7.25
 Von Neumann Architecture, 1.30, 5.48
 VPN, 1.29
 VRAM, 6.6
 VSAT, 1.29
- W**
- Wait cycle, 5.16
 WAN, 1.25
 WCDMA, 1.29
 Web, 1.25
 Web browser, 1.26
 Web crawler, 1.27
 Web page, 1.26
 Web site, 1.26
 Wep, 1.29
 Whetstone, 1.14
- WiFi, 1.28
 WiMAX, 1.29
 Winchester disk drive, 6.14
 WINDOWS-95, 8.29
 WINDOWS-98, 1.22, 8.29
 WINDOWS-2003, 8.30
 WINDOWS-XP, 8.29
 Wordlength of computer, 1.13
 Word perfect, 8.35
 Word processing package, 8.33
 Word processor, 8.33
 Word Star, 8.35
 WPA, 1.29
 WPAN, 1.30
 Workstation, 1.19
 WUSB, 1.30
- X**
- XNOR gate, 3.5
 XOR gate, 3.5
- Y**
- Yahoo, 1.27
- Z**
- Zero-address instruction, 5.7
 Zero-address machine, 5.8
 Zero status, 5.4
 ZigBee, 1.30
 Zoom command, 9.6