UNIT 2 FUNCTIONING OF A COMPUTER

Structure **Page No.**

2.0 INTRODUCTION

We have discussed in the previous unit, that the computers have undergone through a long history of evolution. The evolutionary history is often divided into mechanical and electronic era, with electronic computers started being developed in second half of the 20^{th} century. Since then electronic computers have undergone major transformations in terms of architectural design, components used and the IC fabrication technologies. Modern computers primarily use electronic components for processing element & primary memory and both magnetic & optical components (and solid state devices very recently) for secondary storage.

This unit aims to identify the major components of a digital computer and to describe the functions performed by them. As the unit progresses, we will discuss about the architectural blueprint of digital computers identifying the major components and their roles. This is followed by an introduction to decimal & binary number systems and popularly used binary code systems. We then move to understand the concept of a machine instruction through a simple example, followed by description of functioning of Control Unit and Arithmetic and Logic Unit. The unit concludes with a brief summary of learning outcomes.

2.1 OBJECTIVES

After going through this unit, you should be able to:

- identify the major components of a digital computer;
- understand the role of each component and their interconnection
- understand the language of digital computers;
- describe the decimal and binary number systems;
- describe the binary codes and their use;
- perform inter number system conversions;
- distinguish between weighted and un-weighted codes;
- describe the ASCII and Unicode;
- understand the format of machine instruction; and
- describe the function performed by Control Unit and Arithmetic & Logic Unit.

2.2 COMPONENTS OF A DIGITAL COMPUTER & THEIR ROLE

A digital computer is an electronic device that receives data, performs arithmetic and logical operations and produces results according to a predetermined program. It receives data through an input device (usually keyboard) and displays the results to some output device (usually monitor). All data processing in a digital computer is done by a central processing unit, also known as processor. A working memory is used to store data and instructions. Figure 2.1 presents a block diagram of a digital computer identifying the key components and their interconnection.

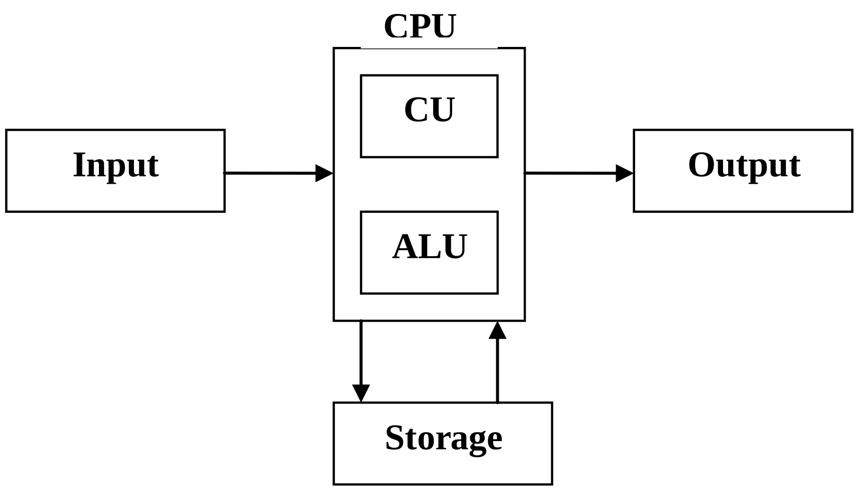

Figure 2.1: Block Diagram of a Digital Computer

2.2.1 Components of a Digital Computer

The key elements of a digital computer, as elaborated in the block diagram given in Figure 2.1 include: Central Processing Unit, Input, Output and Memory. The Central Processing Unit (CPU) is like the brain of the computer. It is responsible for executing instructions. It controls and coordinates the execution of instructions. It is comprised of a Control Unit (CU), an Arithmetic & Logic Unit (ALU) and registers. The CU controls the execution of instructions by decoding the instruction and generating micro-operations to be performed for executing that instruction. The ALU is responsible for performing arithmetic and logic operations. Execution of an instruction involves almost all parts (CU, ALU & Registers) of the CPU. Hence, CPU is known as the most vital component of a computer system.

Input devices are used to read the instructions and data to be processed and output devices display the results obtained after executing the program. Keyboard, Mouse and Scanner are examples of input devices, whereas Monitor, Printer and Plotter are examples of output devices. Memory is used as a working storage for temporarily storing the data and intermediate results generated during program execution. Computers use two kinds of memories: primary & secondary. The primary memory is often referred to as RAM in everyday language. It is a read/write memory used to store both the program and data. Since RAM is volatile, computers also use a second level of memory- secondary memory- to permanently store the contents. Hard Disk is the non-removable secondary storage device which stores virtually everything on the machine. Computers also use other removable secondary memories like CD-ROMs, Magnetic tapes and recently Flash Drives to permanently take backup of the data onto Hard Disk or to transfer data from one machine to another.

A more practical description of a digital computer can be given by describing the major units and their interconnections for a simple personal computer (PC). If you open the CPU cabinet of your PC, you will notice that it contains a printed circuit board on which a number of devices are plugged in. This printed circuit board is often called the mother board. All other major components of the computer are either plugged in directly to this mother board or connected through a bunch of wires. CPU, RAM and Device Cards are plugged in various slots of the mother board. Devices like Hard Disk, Floppy Drive, CDROM Drive, which are attached to the CPU cabinet, are connected through wire ribbons. The mother board has printed circuitry which allows all these components to communicate with each other. CPU cabinet also houses a power supply unit which provides power to all the components of the computer system. On the back end of the CPU cabinet, you can notice a number of connection slots. These slots are used to connect various input/output devices such as keyboard, mouse, printer, scanner, to the computer.

2.2.2 Computer as a Data Processor

Computer takes the raw data as input and performs several operations on these data in order to produce the desired output so it acts as a processing unit for those data

The main function of a computer is to process the input data according to a specific program to produce the desired output. This is the reason why a computer is often viewed as a data processing device. Various components of a computer work coherently to perform different operations to process the data according to program instructions. The word 'Data' refers to any raw collection of facts, figures and statistics. The input data to a computer may include both numbers and characters. Processing the data thus means manipulation of letters, numbers and symbols in a specific manner. The processing may include calculations, decision making, comparisons, classification, sorting, aligning & formatting etc. The processing of data results in some meaningful values, often termed as 'information'. As we see, computer takes the raw data as input and performs several operations on these data in order to produce the desired output so it acts as a processing unit for those data.

The programmers write programs for various data processing tasks. Writing a program to sort a given list of names, to search for a roll number in the list of qualified candidates, preparing and formatting a curriculum vitae, doing an accounting job, are all examples of data processing. As described in Unit I, digital computers have versatile capabilities and that is the reason why they are being used by a variety of people for different purposes. Now-a-days computers have something to offer to everyone. Whether it is an engineer, a business tycoon, a graphic designer, an accountant, a statistician, a student or even a farmer; everyone is now making use of computers. Thanks to the rich set of application programs, even novice users can now make effective use of computers.

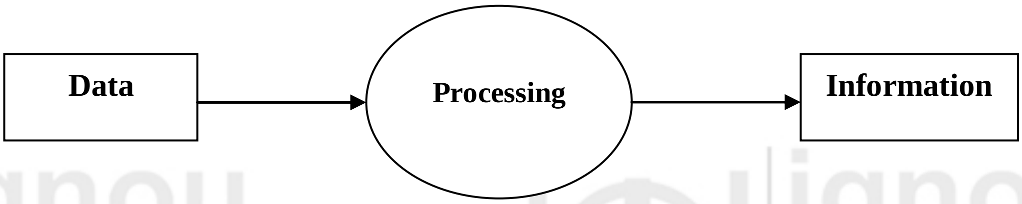

Figure 2.2: Data Processing

2.2.3 Language of Digital Computers

Digital computers are electronic devices which operate on two valued logic (On and OFF). The ability of a transistor to act as a switch is the key to designing digital computers. The digital circuits used in computers are bi-stable, one state each corresponding to ON and OFF values. The two valued Boolean logic (using two distinct symbols 0 and 1) serves as an appropriate representation of states of digital circuits. Every instruction and data item therefore needs to be represented only by using two symbols 0 and 1. Since machines are capable of executing 400-500 distinct instructions and a unique binary code is required to specify every instruction, the machine instructions are specified using multiple bits (binary digits). Similarly the data items also need to be specified using 0 and 1 only. Numbers, alphabets and other characters can be represented by using some binary code system. The next section describes in detail the binary number system and different binary codes.

Though computers use a kind of binary language, we would never like to use them if we are forced to learn and use binary language for working on a computer. In fact, had this been the case, computers would have never become popular. To bridge the gap between the language used by computers and human beings, software got evolved. Software acts as an interface between the machine and the user. Initially the software performed only simple tasks of making the better utilization of machine resources and to make it more convenient for the user to use the machine. The convenience is the simplicity of the way through which users can give instructions to a computer. But as computers started becoming cheaper and popular, softwares to perform more complicated and domain specific tasks were written. Now we refer to these two varieties of softwares as System and Application software respectively. While systems softwares are concerned with driving the machine, application softwares provide task specific functionalities. System software is must to operate a computer system. Operating system is a kind of systems software whereas wordprocessing, database, accounting packages are all examples of applications software. Figure 2.3 below presents the layered view of a computing system.

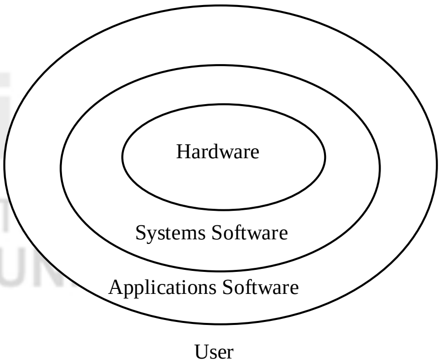

Figure 2.3: Layered View of a Computing System

☞ Check Your Progress 1

1) Fill in the blanks:

   (a) A digital computer is made of following four basic components:

   (i)………………(ii)……………. (iii) ……………… (iv)……………….

   (b) The CPU stands for ....

   (c) Modern processors are capable of executing ……………. distinct machine instructions.

2) What is the significance of a computer being termed as a data processer?

   ………………………………………………………………………………………

   ………………………………………………………………………………………

   ………………………………………………………………………………………

3) What kind of language is used to specify machine instructions of a digital computer?

   ………………………………………………………………………………………

   ………………………………………………………………………………………

4) Choose the best alternative:

   (a) CPU is like ………………… of the computer.

   (i) heart (ii) eyes (iii) brain (iv) ears.

   (b) The language of digital computers comprise of:

   (i) decimal numbers (ii) alphabets (iii) binary numbers

   (c) Which part of the CPU is responsible for arithmetic computations and comparisons?

   (i) Control Unit (ii) Arithmetic & Logic Unit (iii) Registers

2.3 NUMBER SYSTEM

A number system with base r will have r distinct symbols, from 0 to r-1. Binary number system (r = 2), octal number system (r = 8) and hexadecimal number system (r = 16) are some of the frequently used number

We are familiar with decimal number system which uses ten distinct symbols from 0…9, and has base 10. In the decimal number system a number $n_4n_3n_2n_1$ is interpreted as $n_4 \times 10^3 + n_3 \times 10^2 + n_2 \times 10^1 + n_1 \times 10^0$. Thus decimal number 5632 represents 5000+600+30+2. It is a weighted code system since numbers 5632, 2563, 3562, 6532 all represent different quantities despite the fact that all of them use the same symbols (2,3,5,6). The magnitude/ value of a number is determined both by the symbols used and the places at which they are present. Thus, symbol 3 at ten's place represent 30, but when written at thousands' place it represent 3000. Although we use only the decimal number system in everyday applications but there are many other number systems possible. In fact, we can have number system with any base r.

A number system with base r will have r distinct symbols, from 0 to r-1. Binary number system (r = 2), octal number system (r = 8) and hexadecimal number system (r = 16) are some of the frequently used number systems in computer science. Binary number system has two distinct symbols 0 & 1; Octal has seven distinct symbols 0,1,2,3,4,5,6,7; and Hexadecimal number system has sixteen distinct symbols namely 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. The numbers written in a particular number system can be transformed to an equivalent value in a different number system. For example a number 3F in hexadecimal is equivalent to 63 ($3 \times 16^1 + F \times 16^0$) in decimal number system. And similarly a number 302 in octal is equivalent to 194 ($3 \times 8^2 + 0 \times 8^1 + 2 \times 8^0$) in decimal number system.

2.3.1 Binary Number System

As stated above, the binary number system has base 2 and therefore uses only two distinct symbols 0 and 1. Any number in binary is written only by using these two symbols. Though it uses just two symbols but it has enough expressive power to represent any number. A binary number 1001 thus represents the number $1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$. It is equivalent to number 9 in decimal number system. Similarly 11001 and 10001 represent numbers 25 and 17 respectively in decimal number system. Please note that the base in binary number system is 2.

A binary number can be converted to its decimal equivalent by forming the sum of powers of 2 of those coefficients whose value is 1.
For example:

$$(10101)_2 = 2^4 + 2^2 + 2^0 = (21)_{10}$$
$$(100011)_2 = 2^5 + 2^1 + 2^0 = (35)_{10}$$
$$(1010.011)_2 = 2^3 + 2^1 + 2^{-2} + 2^{-3} = (10.375)_{10}$$

The conversion from decimal to binary or to any other base-r system is done by separating the number into an integer part and a fraction part and then converting each part separately. For example the decimal number 41.6875 can be converted to binary equivalent, by converting the integer and fraction parts separately, as follows:

Operation	Quotient	Remainder
41 / 2	20	1
20 / 2	10	0
10 / 2	5	0
5 / 2	2	1
2 / 2	1	0
1 / 2	0	1

The number is divided by 2 and the remainder part is extracted. The quotient obtained is again divided by 2 and this process is repeated until the quotient becomes 0. Every time the remainder obtained is recorded. The set of remainders obtained, read from the bottom to top form the binary equivalent of the integer part of the number. Thus $(41)_{10}$ = $(101001)_2$.

In order to convert the fraction part, it is multiplied by 2 to obtain resultant integer and fraction parts. The integer part in the resultant is extracted and the fraction part is again multiplied by 2. This multiplication process is repeated till the fraction part becomes 0 or a desired precision is achieved. The integer part of the given number (.6875) can be converted to binary as follows:

Operation	Resulting Integer part	Resulting Fraction part
0.6875 X 2	1	.3750
0.3750 X 2	0	.7500
0.7500 X 2	1	.5000
0.5000 X 2	1	.0000

The binary equivalent of fraction 0.6875 is 1011, obtained by reading the integer parts from top to down. Thus, the decimal number 41.6875 is equivalent to 101001.1011 binary number system.

Conversion

Decimal to Binary:

Q. Convert $(13)_{10}$ to an equivalent binary number**.**

	Quotient	**Remainder**
13/2	6	1
6/2	3	0
3/2	1	1
1/2	0	1

We take the remainder from bottom to top. Hence $(13)_{10} = (1101)_2$

Binary To Decimal

Q. Convert $(101011)_2$ to an equivalent decimal number.

5 4 3 2 1 0

$(1\ 0\ 1\ 0\ 1\ 1)_2$

$= 2^5 + 2^3 + 2^1 + 2^0 = 32 + 8 + 2 + 1 = 43$

Hence, $(101011)_2 = 43$

Hexadecimal to Binary

Q. Convert $(2D5)_{16}$ to an equivalent Binary number.

$(2D5)_{16}$ = 2 D 5

0010 1101 0101

We replaced each digit in the given number by its 4-bit binary equivalent. So

$(2D5)_{16}$ = 0010 1101 0101

Thus, $(2D5)_{16} = (001011010101)_2$

Binary to Hexa Decimal

Q. Convert $(10100110101111)_2$ to its equivalent Hexa Decimal Number.

Starting from the least significant bit, each group of 4 bits is replaced by its decimal equivalents.

$(10100110101111)_2$ = 0010 1001 1010 1111

2 9 A F

Thus, $(10100110101111)_2 = (29AF)_{16}$

2.3.2 Binary Codes

We have seen earlier that digital computers use signals that have two distinct values and there exists a direct analogy between binary signals and binary digits. Computers not only manipulate numbers but also other discrete elements of information. The distinct discrete quantities can be represented by a group of binary digits (known as bits). For example, to represent two different quantities uniquely two symbols are sufficient and hence one binary digit (either 0 or 1) will be sufficient to uniquely represent the two symbols. But one bit will not suffice if one has to represent more than two quantities. In those cases more than one bits are required, i.e. the bits have to used repeatedly. For example if we have to give unique codes for three distinct items we need at least 2 bits. With two bits we can have codes 00, 01, 10 and 11. Out of this we can use first three to assign unique codes to three distinct quantities and leave the fourth one unused. In general, an n-bit binary code can be used to represent 2^n distinct quantities. Thus group of two bits can represent four distinct quantities through unique symbols 00, 01, 10 & 11. Three bits can be used to represent eight distinct quantities by unique symbols 000, 001, 010, 011, 100, 101, 110 & 111. In other words, to assign unique codes to m distinct items we need at least *n* bit code such that $2^n >= m$.

Digital computers use binary codes to represent all kinds of information ranging from numbers to alphabets. Whether we have to input an alphabet, a number or a punctuation symbol; we need to convey it to machine through a unique code for each item. Thus, the instruction to be performed by the CPU and the input data which form the operands of the instruction are represented using a binary code system. A typical machine instruction in a digital computer system could therefore look like a set of 0s and 1s. Many binary codes are used in digital systems. BCD code for representing decimal numbers, ASCII code for information interchange between computer and keyboard, Unicode for use over Internet and Reflected (Gray) code are some commonly studied binary code systems.

2.3.3 ASCII & Unicode

An alphanumeric code has to represent 10 decimal digits, 26 alphabets and certain other symbols such as punctuation marks and special characters. Therefore, a minimum of six bits is required to code alphanumeric characters ($2^6 = 64$, but $2^5 = 32$ is insufficient). With a few variations this 6 bit code is used to represent alphanumeric characters internally. However, the need to represent more than 64 characters (to incorporate lowercase and uppercase letters and special characters), have given rise to seven- and eight- bit alphanumeric codes. ASCII code is one such seven bit code that is used to identify key press on the keyboard. ASCII stands for American Standard Code for Information Interchange. It's an alphanumeric code used for representing numbers, alphabets, punctuation symbols and other control characters. It's a seven bit code, but for all practical purposes it's an eight bit code, where eighth bit is added for parity. Table 2.1 below presents the ASCII code chart.

ASCII is an alphanumeric code used for representing numbers, alphabets, punctuation symbols and other control characters

Table 2.1 : ASCII Code Chart

Bits												
b_7 → b_6 → b_5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FC	,	<	L	\	l	/
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

ASCII codes represent text in computers, communications equipment, and other devices that use text. Most modern character-encoding schemes are based on ASCII, though they support many more characters than did ASCII. Historically, ASCII developed from telegraphic codes. Its first commercial use was as a seven-bit teleprinter code promoted by Bell data services. Work on ASCII formally began on October 6, 1960, with the first meeting of the American Standards Association's (ASA) X3.2 subcommittee. The first edition of the standard was published during 1963 a major revision during 1967, and the most recent update during 1986. ASCII includes definitions for 128 characters: 33 are non-printing control characters (now mostly obsolete) that affect how text and space is processed; 94 are printable characters, and the space is considered an invisible graphic. The most commonly used character encoding on the World Wide Web was US-ASCII until December 2007, when it was surpassed by UTF-8.

Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems. Developed in conjunction with the Universal Character Set standard and published in book form as *The Unicode Standard*, the latest version of Unicode consists of a repertoire of more than 107,000 characters covering 90 scripts, a set of code charts for visual reference, an encoding methodology and set of standard character encodings, an enumeration of character properties such as upper and lower case, a set of reference data computer files, and a number of related items, such as character properties, rules for normalization, decomposition, collation, rendering, and bidirectional display order (for the correct display of text containing both right-to-left scripts, such as Arabic and Hebrew, and left-to-right scripts. Unicode can be implemented by different character encodings. The most commonly used encodings are UTF-8 (which uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters), the now-obsolete UCS-2 (which uses two bytes for each character but cannot encode every character in the current Unicode standard), and UTF-16 (which extends UCS-2 to handle code points beyond the scope of UCS-2).

Unicode is a computing industry standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems

The Unicode Consortium, the nonprofit organization that coordinates Unicode's development, has the ambitious goal of eventually replacing existing character encoding schemes with Unicode and its standard Unicode Transformation Format (UTF) schemes, as many of the existing schemes are limited in size and scope and are incompatible with multilingual environments. Unicode's success at unifying character sets has led to its widespread and predominant use in the internationalization and localization of computer software. The standard has been implemented in many recent technologies, including XML, the Java programming language, the Microsoft .NET Framework, and modern operating systems.

☞ Check Your Progress 2

1) Choose the best alternative:

(a) ASCII is practically a ………………. bit code:

(i) six (ii) seven (iii) eight (iv) nine

(b) A number system having base r will have ……….. distinct symbols:

(i) r/2 (ii) 2^r (iii) r (iv) r^2

(c) Unicode is used primarily for:

(i) processing (ii) Internet technologies (iii) logical operations

(d) The binary equivalent of $(25.25)_{10}$ is:

(i) 11001.001 (ii) 11101.010 (iii) 10001.110

2) State **True** or **False**:

(a) A decimal number can be converted into its binary equivalent integer and fraction parts separately. True ☐ False ☐

(b) An n-bit binary code can represent n^2 distinct codes. True ☐ False ☐

(c) The most commonly used Unicode Transformation Format currently is UTF-8. True ☐ False ☐

3) Briefly state the reason why ASCII can not be less than 7 bit code?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

4) What was the main motivation behind the development of Unicode?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

2.4 CONCEPT OF INSTRUCTION

The CPU is a semiconductor integrated circuit chip consisting of a large number of transistors. In personal computers, the CPU is also referred by the term Microprocessor. Every CPU is capable of performing certain instructions (known as machine instruction). Modern CPUs have the logic built in to perform 400-550 machine instructions. The machine instructions that a CPU can execute demonstrates its capability. Every processor is capable of performing certain operations. An instruction refers to an operation that can be performed by the processor directly. The entire set of instructions that can be executed by the processor directly, through the logic in hardware, form the instruction set of the processor. An instruction tells the processor what task is to be performed and what micro-operations need to be completed to perform the task. Every instruction execution requires execution of a set of arithmetic and logical operations (micro-operations). The size and format of the instruction varies with different processors.

> The entire set of instructions that can be executed by the processor directly, through the logic in hardware, form the instruction set of the processor. An instruction tells the processor what task is to be performed and what micro-operations need to be completed to perform the task

Every instruction is comprised of two parts: **opcode** and **operands**. The opcode specifies the operation to be performed and the operands provide the data on which the operation is to be performed. To understand the concept of instruction more clearly let us assume a simple hypothetical computer which the capability to perform eight different operations. Every operation is specified by a unique opcode as given in Table 2.2.

> The opcode specifies the operation to be performed and the operands provide the data on which the operation is to be performed

Table 2.2 : Unique OPcode

Operation	Opcode
Addition	000
Subtraction	001
Multiplication	010
Division	011
Modulus	100
Complement	101
Bitwise AND	110
Bitwise OR	111

Let us further assume that our computer can process only two-digit decimal numbers, i.e. there can be a maximum of two operands each of a maximum of two digits. Thus the computer can add or subtract numbers containing a maximum of two digits. A simple instruction can thus be written as a combination of an opcode and its associated operands. Opcode is denoted by its unique binary code. The operands are decimal digits and therefore also need to be converted to binary code system to pass them as operands to the processor. Suppose BCD code is used to represent the operands. Then following are examples of some valid instructions on the processor:

Instruction	**Effect**
0001001001100100101	93 + 25
10110000101	Complement 85
0110010010100000101	25 / 05

In the first instruction, the first three bits represent the opcode and the remaining sixteen bits represent the two operands each a two digit decimal number expressed using BCD code. The opcode for addition as described in the table is 000 and the BCD codes for 9,3,2 and 5 are 1001, 0011, 0010 and 0101 respectively. Thus the instruction 0001001001100100101 represents 93 + 25. Similarly, in the second instruction, first three bits represent the opcode and the remaining eight bits specify the operand to perform the operation. However, this is the case of a very simple hypothetical computer. Real world processors are much more complex and capable of performing more than 500 machine instructions. Further they can take their operands in a number ways: directly, from registers, from memory etc. Moreover, modern processors can perform calculations on large numbers. Thus an instruction in a modern CPU could easily comprise more than 50 bits.

Execution of a machine instruction on modern processors involves a complex sequence of operations with multiple cycles. One instruction cycle typically involves Fetch, Decode, Execute and Write back cycles. During Fetch cycle the instruction is fetched from memory. During Decode cycle the instruction is processed by the control unit of the CPU, which generate the set of micro-operations and timing signals required to execute the instruction. The micro-operations are then executed during Execute cycle to complete the instruction and any results generated are then written back to memory during Write back cycle. An instruction cycle may also involve one or more operand fetch cycles. The control unit is responsible for overall control and coordination of instruction execution. It generates the set of micro-operations either through a hard wired logic or with the help of micro program sequencer. Executing a program therefore involves executing a large number of machine instructions, where every machine instruction execution requires executing several micro-operations.

The control unit is responsible for overall control and coordination of instruction execution

2.5 ELEMENTS OF CPU AND THEIR ROLE

The CPU is a complex IC chip having millions of transistors. As mentioned earlier, a CPU has three major identifiable parts: Control Unit (CU), Arithmetic & logic Unit (ALU) and a set of Registers. The CPU chip is interfaced with other components of the computer through a system bus (printed wires on the mother board) which has three sets of wires forming Control Bus, Data Bus and Address Bus. Figure 2.4 presents the components of a CPU :

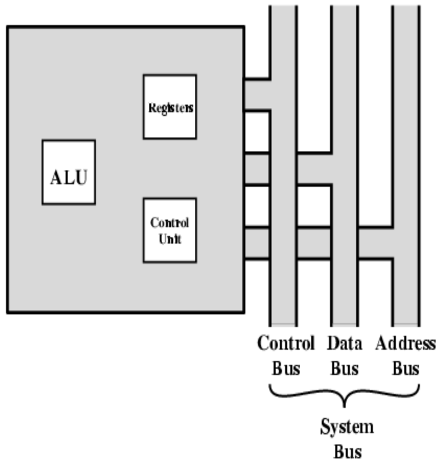

Figure 2.4 : Interfacing CPU and System Bus

CPU has a set of **Registers** which is used to store some data temporarily. Register lies above Cache and Main memory in memory hierarchy of the system. The registers in CPU perform two roles:

- **User-visible registers**: used to store temporary data items and other user accessible information useful for machine or assembly language programmers.
- **Control & Status Registers**: used by control unit to control and coordinate the operation of the processor.

The **Control Unit** of the processor is that unit which controls and coordinates the execution of instructions by the processor. It is responsible for defining and controlling the instruction cycle. In essence, it causes things to happen in the processor. It issues control signals external to the processor to cause data exchange with memory and I/O modules. It also issues control signals internal to the processor to move data between registers, to cause the ALU to perform a specified function, and to regulate other internal operations. It generates timing signals and initiates the Fetch cycle of instruction execution. When the instruction is fetched, it generates the sequence of micro-operations which need to be executed in order to execute the instruction. CU also generates timing signals for executing set of micro-operations. There are three different ways in which CU can generate these micro-operations: through a hardwired logic, by reading a programmable Array (PLA) table or by reading a Programmable Read Only Memory (PROM).

> The **Control Unit** of the processor is that unit which controls and coordinates the execution of instructions by the processor

In hardwired control, the mapping between machine instruction and consequent micro-operations to be generated is permanently wired within the processor. It is relatively faster way although it cannot be modified. In PLA control the sequence of micro-operations to be generated for executing an instruction is stored as a PLA table. In Micro program control, the logic of the control unit is specified by a microprogram. Microprogram specifies the micro-operations. The microprogram control has a control memory (a PROM chip) which stores the sequence of micro-operations. As a general rule processors having smaller instruction set (such as RISC processors) have hard wired control logic whereas microprogram control is used in processors having larger instruction set. Most of the modern CISC processors use microprogram control.

> The microprogram control has a control memory (a PROM chip) which stores the sequence of micro-operations

The **Arithmetic and Logic Unit** is that part of the CPU that actually performs arithmetic and logical operations on data. The CU, CPU registers and memory help in bringing the data into the ALU and then taking the results back. Figure 2.5 presents the ALU inputs and outputs.

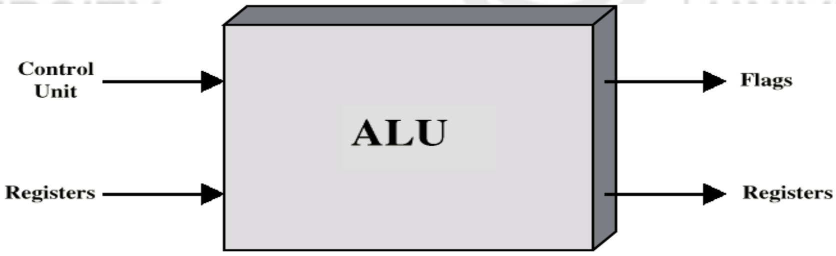

Figure 2.5 : The Arithmetic and Logic Unit

Data are presented to ALU in registers and the results are also stored in registers. Accumulator is one such register which is very frequently used during the ALU operation. ALU has many other registers such as flags and status register, which indicate information about the operation and its result. ALU has logic implemented to perform operations like addition, multiplication, division, shifting, complement etc. The operations are performed on represented numbers, both integer and floating point numbers.

Modern processors nowadays have two identifiable trends which improve their performance to a much higher level. These are use of on chip Cache memory and having more than one processor core on the same IC chip. Cache memory is a fast semiconductor memory which can be used to temporarily store instructions and data that are frequently referred by the processor. By having frequently referred instructions and data available in the processor, the wait cycles introduced due to memory references are minimized and hence the processor performance improves a lot. Another modern technique of having more than one processor core on the same IC chip tries to perform the execution of instructions in parallel and hence the performance of the processor improves a lot.

☞ Check Your Progress 3

1) Choose the best alternative:
   (a) A machine instruction consists of ……………………
   (i) ALU (ii) Operator (iii) Opcode & Operands (iv) CU
   (b) Executing an instruction involves execution of several……………….
   (i) micro-operations (ii) programs (iii) control signals
   (c) ALU is responsible for …………….
   (i) generating control signals (ii) microprogram sequencing
   (iii) arithmetic & logic
   (d) The hardwired control unit is ……………..
   (i) permanent, non-modifiable (ii) made of PROM (iii) erasable
   (e) CPU has following two kinds of registers……………
   (i) user-visible & control (ii) program counter & flags (iii) ALU & CU

2) Briefly explain the sub cycles involved in execution of a machine instruction.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………..

3) What are the different ways of designing the control unit of a processor?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………..

4) List the two key technological trends seen in modern processors for improving performance.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………..

2.6 SUMMARY

This unit has introduced you to the components of a digital computer and their roles. A digital computer has CPU, Memory, Input and Output as constituent elements. Digital computers use binary language. The binary number system is analogous to two valued Boolean logic. Decimal numbers and alphabets are represented as binary codes for being processed by the computer. The unit describes the ASCII and Unicode code systems in brief. ASCII is the code used for information interchange between keyboard and computer. Unicode is a universal code system widely used over the Internet. The unit also introduced you with the concept and format of a machine instruction. An instruction cycle involves fetch, decode, execute and write back cycles. The three key components of CPU are: CU, ALU and registers. CU controls and coordinates the execution of instructions. ALU actually performs the operations. CPU registers work as temporary storage for executing instructions.

2.7 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) (a) CPU, Memory, Input, Output (b) Central processing Unit
   (c) 400-550
2) A computer accepts raw data as input and processes it (performing computations, manipulations and formatting etc.) to produce a meaningful result. Due to its key job being processing according to user instructions, it is referred to as data processor.
3) A language comprising of binary numbers is used to specify machine instructions in digital computers. Every machine instruction is represented by a unique binary code.
4) (a) iii (b) iii (c) ii

Check Your Progress 2

1) (a) ii (b) iii (c) ii (d) i

2) (a) True (b) False (c) True

3) ASCII Code is required to have unique code for every key press on the keyboard. Through the keyboard we can enter 10 decimal digits, 26 letters of alphabets and certain other symbols such as punctuation marks and special characters. Therefore, a minimum of six bits is required to code alphanumeric characters ($2^6 = 64$, but $2^5 = 32$ is insufficient). However, the need to represent more than 64 characters (to incorporate lowercase and uppercase letters and special characters), made ASCII a 7-bit code.

4) Unicode was developed as a standard for the consistent encoding, representation and handling of text expressed in most of the world's writing systems. Unicode consists of a repertoire of more than 107,000 characters covering 90 scripts. It is the universal encoding scheme having special significance for Internet and multilingual computing

Check Your Progress 3

1) (a) iii (b) i (c) iii (d) ii (e) i

2) Execution of a machine instruction involves Fetch, Decode, Execute and Write Back cub cycles. An instruction cycle may also involve operand fetch cycles.

3) There are three ways of designing the control unit: Hard-wired control, PLA and Micro-program control.

4) The two technological trends seen in modern processors to improve performance include provision of on-chip cache memory and having multiple processing cores on the same chip.

2.8 FURTHER READINGS

William Stalling, *Computer Organization and Architecture*, 6th Ed., Pearson Education.
V. Rajaraman, *Fundamentals of Computers*, PHI.

Web Link:

- http://computer.howstuffworks.com