1

INTRODUCTION TO COMPUTER, MICROPROCESSOR AND MICROCONTROLLER

Learning Objectives

After you have completed this chapter, you should be able to

- Define or explain the following terms: Computer, CPU, Memory, Input unit, Output unit, Bus, RAM and ROM
- Explain the difference between microprocessor and microcontroller
- Compare features of commercial microcontrollers
- Explain the difference between Von Neumann (Princeton) architecture and Harvard architecture
- Explain the difference between RISC and CISC machines
- Explain the difference between machine language, assembly language and high level language programming of a computer
- Explain embedded system architecture, design challenges and embedded operating system

1.1 WHAT IS A COMPUTER?

A computer is a multipurpose programmable machine that reads binary instructions from its memory, accepts binary data as input, processes data according to those instructions and provides result as output. It is a programmable device, made up of *hardware* and *software*. The various components of the computer are called *hardware*. A set of instructions written for the computer to solve a specific task is called a *program* and a collection of programs is called *software*. The computer hardware consists of four main components:

1. *Central Processing Unit (CPU)*, which acts as the computer's brain.
2. *Input unit*, through which program and data can be entered into the computer.
3. *Output unit*, from which the results of computation can be displayed or viewed and
4. *Memory*, in which the programs and data are stored. Figure 1.1 shows a simple block diagram of a computer.

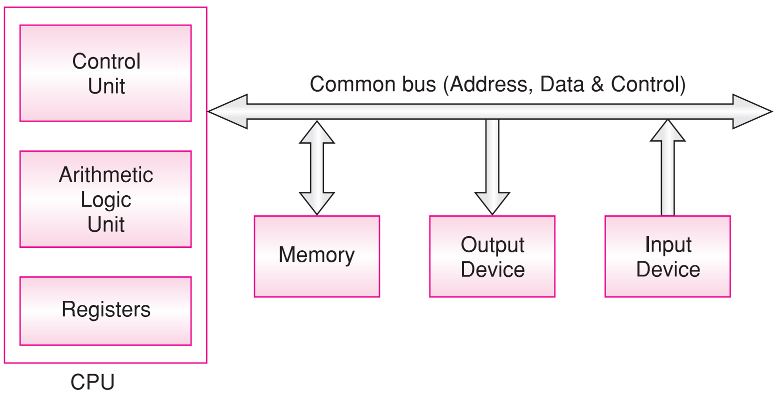


Figure 1.1 *Block diagram of a computer*

The processor communicates with the memory and input/output devices using three sets of lines called *buses—address bus, data bus*, and *control bus*. The bus is a communication path between the processor and the peripherals.

1.1.1 CENTRAL PROCESSING UNIT

The Central Processing Unit (CPU), which is also called the processor, can be further divided into three major parts:

Register File The register file consists of one or more registers. A register is a storage location in the CPU. It is used to hold data and/or a memory address during execution of an instruction.

Arithmetic Logic Unit The arithmetic logic unit is the computer's numerical calculator and logical operation evaluator. Under command from the control unit, it receives information from the memory. It analyses and rearranges the data and carries out the sequence of arithmetic and logical operations to accomplish the desired job.

Control Unit It controls and coordinates all the activities in the computer. It decodes the instructions and generates the necessary control signals for the execution of instructions. The system clock synchronises the activities of the control unit. All CPU activities are measured by clock cycles. The control unit also contains a register called the Program Counter (PC), which contains the memory address of the next instruction to be executed.

1.1.2 MEMORY

The memory is a place where programs and data are stored. It is a storage device that stores instructions, data and intermediate results and provides that information to the other units of the computer. A computer contains Semiconductor, Magnetic or Optical memory. Semiconductor memory has been discussed in this chapter. Figure 1.2 shows the classification of semiconductor memory into two major types—Random Access Memory (RAM) and Read Only Memory (ROM).

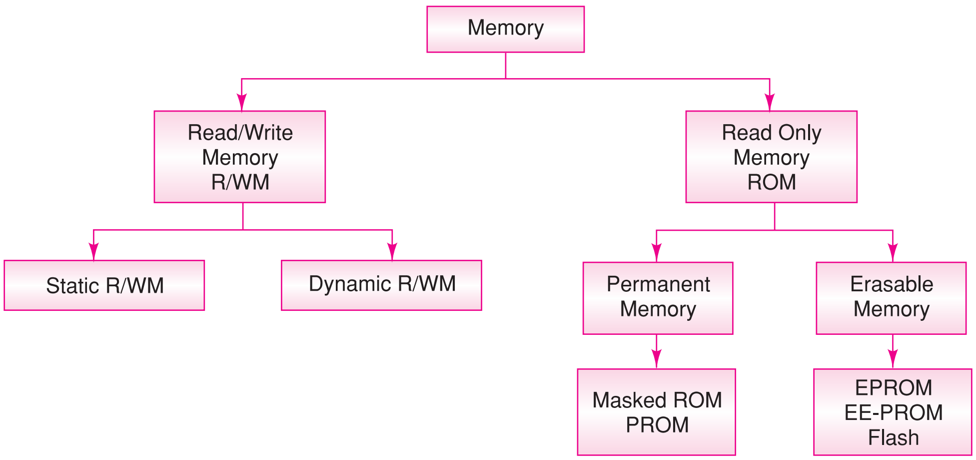


Figure 1.2 *Memory classification*

Random Access Memory Random access memory, popularly known as user memory, is used to store user program and data. It is also known as read/write (R/W) memory because it is equivalent to a group of addressable registers—you can either read the stored contents of memory location or write new contents into the memory location. The R/W memory is volatile, which means that when the power is turned off, the contents are lost. To communicate with the memory, the CPU should be able to

- Select the memory chip
- Select the register in the memory chip
- Read from or write into the register

Model of a typical read/write memory is shown in Fig. 1.3(a). The figure shows the chip select as $\overline{CS}$, write signal as $\overline{WR}$ and read signal as $\overline{RD}$—these are active low signals indicated by the bar. They help to select RAM and perform write and read operations. A flip-flop that can store one binary bit is called a *memory cell*. If more than one flip flop are grouped together, it is called a *register*.

In Fig. 1.3(b), the random access memory contains eight registers; each register contains eight memory cells and are arranged in a sequence. Here, the size of the memory is 8 × 8 bits or 8 bytes. To write into or read from any one of the registers, a specific register has to be selected. This is achieved by using a 3 to 8 decoder. Three address lines A2, A1 and A0 are required for the decoder. These input lines can have eight different bits combination (000, 001, 010, 011, 100, 101, 110, 111), and each combination can select one of the registers.

Registers are made up of flip-flops and it stores bit as a voltage. *Dynamic memory* uses one MOS transistor and one capacitor to store one bit of information in the form of electrical charge. Static memories are designed to store binary information without needing periodic refreshes and require one flip-flop to store one bit of information (four to six transistors are needed to store one bit of information). Hence, it requires the use of more complicated circuitry for each bit. The advantage of dynamic memory is that a large number of transistor gates can be placed on the memory chip; thus it has high density and is faster than static memory. The disadvantage is that the charge (bit information) leaks and hence, it requires periodic refreshing, i.e. information needs to be read and written again after every few milliseconds.

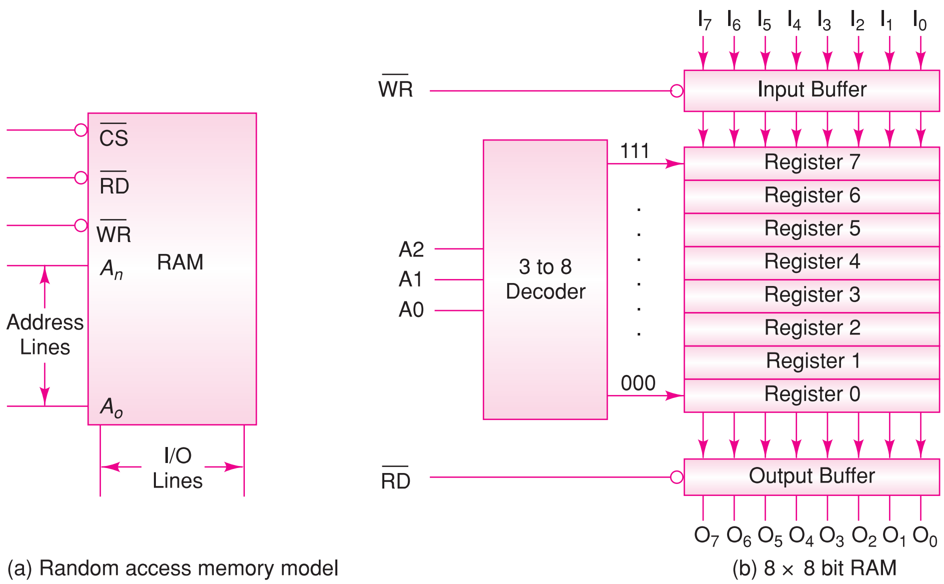
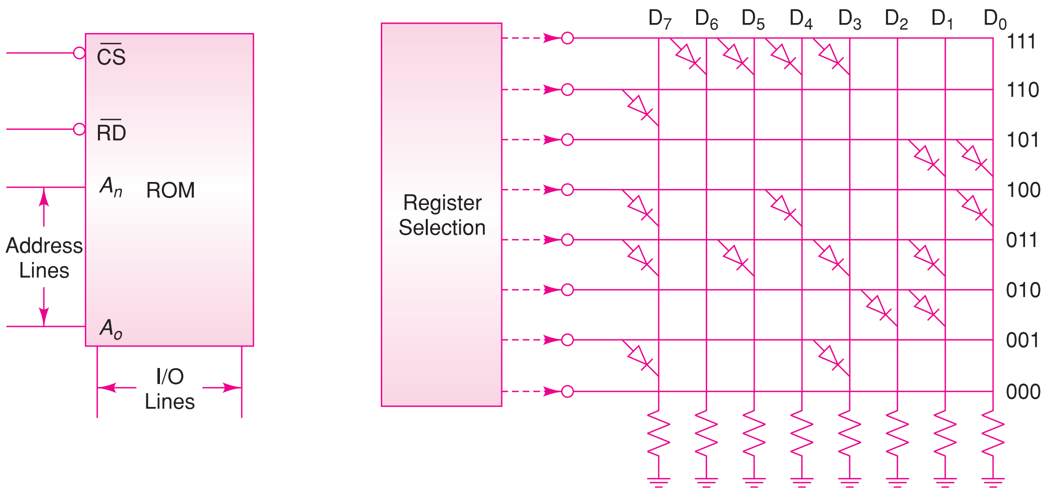


Figure 1.3 *Random Access Memory*

Read Only Memory The read only memory is the simplest kind of memory. It is equivalent to a group of registers and each store a word permanently. It is a nonvolatile memory which means that it retains the stored information even if the power is turned off. By applying control signal, we can read the contents of any memory location and if the processor attempts to write data to a ROM location, ROM will not accept the data. The model of a typical read only memory is shown in Fig. 1.4(a). The concept of ROM can be explained with the diodes arranged in a matrix format, as shown in Fig. 1.4(b). The horizontal lines are connected to the vertical lines using diodes to store '1'. The presence of diode stores '1' and the absence of diode stores '0'. When a register is selected, the voltage of that line goes high, and the output lines, where the diodes are connected also go high. When the memory register 010 is selected, the data byte 00000110 (06H) can be read at the data lines.

Two types of ROM are presently available, permanent ROM and erasable ROM. *Permanent* ROM includes two types of memory, Masked read only memory and Programmed read only memory.

Masked Read Only Memory (MROM) MROM is a type of ROM where binary instruction/data are stored during its fabrication. The semiconductor manufacturer places binary instruction/data in the memory according to the request of the customer. MROM is used to hold microcontroller application program and constant data.



(a) Read only memory model (b) 8 × 8 bit ROM

Figure 1.4 *Read Only Memory*

Programmed Read Only Memory (PROM) PROM is a type of read only memory that can be programmed in the field using a device called PROM programmer and here the programming is permanent. In other words, the stored contents cannot be erased.

Erasable memory includes three types of memory—*EPROM, EEPROM*, and *Flash memory*.

Erasable Programmable Read Only Memory (EPROM) The information stored in this memory is semi-permanent. Data is stored with PROM programmer. Subsequently, all the information can be erased by exposing the memory to ultraviolet light through a quartz window installed on top of the EPROM chip. It can then be reprogrammed.

Electrically Erasable Programmable Read Only Memory (EEPROM) EEPROM is a type of nonvolatile memory that can be erased by electrical signals and reprogrammed. EEPROM allows the user to selectively erase a single location, a row, or the whole chip. This feature requires a complicated programming circuitry. Because of this, EEPROM cannot achieve the density of EPROM technology.

Flash Memory Flash memory was invented to incorporate the advantages and avoid the drawback of EPROM and EEPROM technologies. It achieves the density of EPROM, and can be programmed and erased electrically like EEPROM. However, it does not allow individual location to be erased, but the user can erase the whole chip.

1.1.3 INPUT DEVICES

The input device transfers data and instructions in binary form, from the outside world to the CPU. The input device, also known as a peripheral is the means through which the user communicates data to the computer. The user can enter instructions and data through an input device like a keyboard. Analog to digital converters, and switches can also be used as input devices.

1.1.4 OUTPUT DEVICES

It receives the stored result from the memory or from the CPU, converts it into a form that the user can understand and transfers this data to the outside world. Monitor, Liquid Crystal Display (LCD), seven-segment display, printer, etc. are used as output devices.

SECTION REVIEW

1. Name the four main components of a computer.
2. List three types of buses in a computer.
3. ___________ register contains the memory address of the next instruction to be executed.
4. Differentiate volatile and nonvolatile memory.
5. List the signals that helps the CPU to communicate with the memory.
6. 8×8 bit memory stores ___________ bit of information.
7. ___________ number of address lines are required to address 16×8 bit memory.
8. Analog to digital converter is a ___________ device.
9. LCD is a ___________ device.
10. ___________ memory can be erased electrically.

1.2 WHAT IS A MICROPROCESSOR?

In the late 1960s, the CPU was designed with discrete logic gates. As semiconductor technology advanced, it was possible to fabricate more than thousand gates on a single silicon chip—this came to be known as *Large-scale integration*. As technology moved from Small-scale Integration to Large-scale Integration, it became possible to build a whole CPU with its related timing functions on a single chip. This came to be known as *Microprocessor*. A computer that is designed using a microprocessor as its CPU, is known as a Microcomputer.

Intel Corporation announced the first 4 bit microprocessor 4004 in 1971. The number of bits refer to the number of binary digits that the microprocessor can manipulate in one operation. Soon after this, Intel developed an 8 bit microprocessor 8080 in 1974. The 8085 microprocessor that followed 8080, had few more additional features compared to 8080 architecture. The instruction sets of 8080 and 8085 are practically the same. The 8085 microprocessor has an 8 bit data bus, so it can read data from or write data to memory or I/O ports only 8 bit at a time. It has a 16 bit address bus, so it can address any one of 2^{16} or 65,536 memory locations. It operates with 3MHz clock signal. The limitations of 8085 microprocessor are

- It operates with low speed
- Less powerful addressing mode and instruction set
- Limited number of 8 bit general purpose registers
- Low memory (64 Kbytes) addressing capability

Because of these limitations, Intel Corporation announced the first 16 bit microprocessor 8086 in 1974. The 8086 microprocessor has 16 bit data bus and 20 bit address bus. It can read or write data to memory or I/O ports either 16 bit or 8 bit at a time and it can address any one of 1,048,576 (2^{20}) memory locations. Subsequently, Intel developed 16 bit processors such as 8088, 80186, 80188 and 80286. The Intel 8088 has the same instruction set and arithmetic unit as the 8086. It has 8 bit data bus and 20 bit address bus and can read or write data to memory or I/O devices only 8 bit of data at a time. 80186 and 80188 are an improved version of 8086 and 8088 respectively. Both the processors have a few additional instruction sets compared to 8086 instruction set. 80286 is an improved version of 80186. It is designed to be used as CPU in multitasking computers. It operates in real and virtual addressing mode. Later, in 1985, Intel developed 32 bit microprocessor 80386. The 80386 can address directly up to 4 gigabytes of memory. Intel's second generation of 32 bit microprocessor 80486 was available in the year 1989. Intel introduced a third generation 80586 (Pentium processor) in 1998. Motorola, Zilog, etc. also developed 8 bit, 16 bit and 32 bit Microprocessors. Microprocessors have been widely used after their invention. However, the following limitations of the microprocessor led to the invention of the microcontroller.

- A microprocessor requires external memory to execute a program.
- A microprocessor cannot be directly interfaced with I/O devices. Peripheral chips are needed to interface I/O devices.

SECTION REVIEW

1. The 8085 microprocessor has ___________ bit data bus and ___________ bit address bus.
2. List the limitations of the 8085 microprocessor.
3. The 8088 microprocessor has ___________ bit data bus and ___________ bit address bus.
4. The 80386 is a ___________ bit microprocessor.
5. List the limitations of microprocessor.

1.3 WHAT IS A MICROCONTROLLER?

As technology moved from LSI to VLSI, it became possible to build the microprocessor, memory and I/O devices on a single chip. This came to be known as the 'Microcontroller.'

A microcontroller contains a microprocessor and also one or more of the following components.

- Memory
- Analog to Digital (A/D) converter
- Digital to Analog (D/A) converter
- Parallel I/O interface
- Serial I/O interface
- Timers and Counters

Figure 1.5 shows the block diagram of a typical microcontroller, which is a true computer on-chip. The first 4 bit microcontroller was developed by different companies like Hitachi, National, Toshiba, etc. Soon after this, 8 bit microcontrollers were developed by Intel, Motorola, Zilog, Philips, Microchip technology, etc.

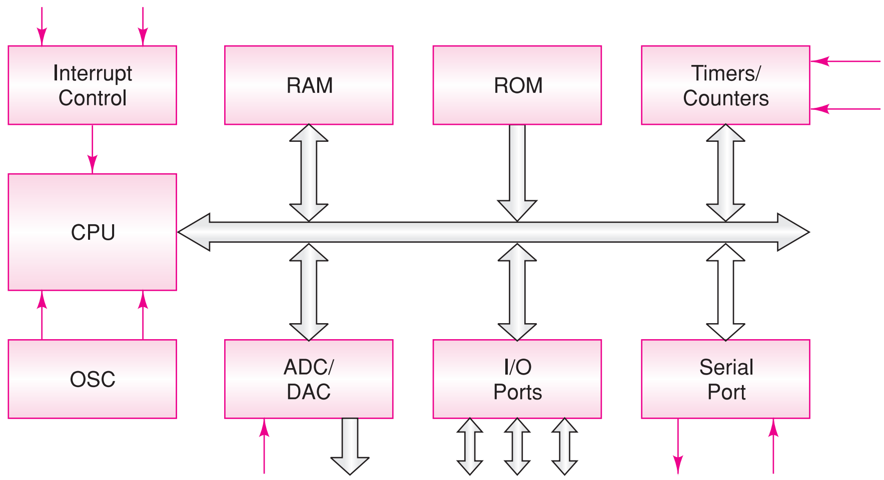


Figure 1.5 *Block diagram of a typical microcontroller*

1.3.1 APPLICATIONS OF MICROCONTROLLER

Microcontrollers have been widely used in home appliances such as refrigerators, washing machines and microwave ovens. It is used in displays, printers, keyboards, modems, charge card phones and also in automobile engines, etc. as controllers.

1.3.2 COMMERCIAL MICROCONTROLLER DEVICES

A brief overview of some commercial microcontrollers, PIC microcontrollers, Intel microcontrollers and Atmel microcontrollers is given in this section. Microcontrollers must be selected depending on the needs of a given application. Table 1.1 lists the various microcontrollers with important features like on-chip memory, number of timers, DMA, A/D converter and UART.

TABLE 1.1 *Microcontrollers with important features*

Device	Register Memory On-chip (bytes)	On-chip program memory ROM/EPROM	Speed MHz	No. of timers/ Counters	No. of I/O lines	On-chip Peripherals
8031 (MCS51-family)	128	ROM less	12	2	32	UART
8051 (MCS51-family)	128	4K ROM	12	2	32	UART
8052 (MCS51-family)	256	8K ROM	12	3	32	UART

(*Contd*)

(*Contd*)

8751 (MCS51-family)	256	8K EPROM	12	3	32	UART
87C58 (MCS51-family)	256	32K EPROM	12-24	3	32	UART
87C51GB (MCS51-family)	256	8K EPROM	12-16	3	48	UART, 8 channel ADC DMA
89C61x2 (MCS51-family)	1024	64K Flash	20-33	3	32	UART
AT89S8252 Atmel	256	8K Flash 2K EPROM	24	3	32	UART SPI
16C74 (Microchip)	192	4K ROM	20	3	32	USART 8 bit ADC SPI
16F874/877 (Microchip)	256	8K	20	3	32	USART 10 bit ADC

SECTION REVIEW

1. List the components in a microcontroller.
2. The 8051 is a ____________ bit microcontroller.
3. List few applications of a microcontroller.
4. List few commercial microcontrollers with their important features.
5. ____________ Intel microcontroller is ROM less.
6. The 8052 has ____________ bytes of on-chip RAM and ____________ bytes of on-chip ROM.
7. The 8751 has ____________ number of I/O lines.
8. 16F874 microchip operates with speed ____________ MHz.
9. List the features of ATMEL AT 89S252 microcontroller.
10. List the on-chip peripherals in an 8051 microcontroller.

1.4 VON NEUMANN (PRINCETON) AND HARVARD ARCHITECTURE

A microprocessor that fetches instruction and data using a single bus is called Von Neumann or Princeton architecture. In Von Neumann architecture, data memory (RAM) and Program memory (ROM) are connected by using single address and data bus as shown in Fig. 1.6.

In Harvard architecture, program memory and data memory are connected using separate address and data bus to achieve fast execution speed for a given clock rate as shown in Fig. 1.7. For example, 8051 microcontroller by Intel and PIC microcontroller by microchip have Harvard architecture. Motorola 68HC11 microcontroller has Von Neumann architecture.

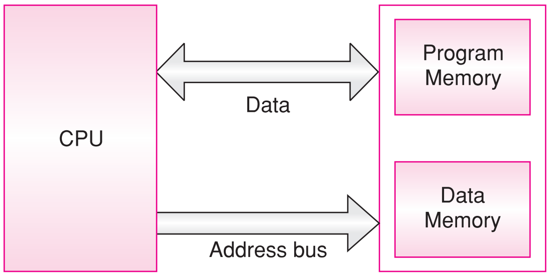


Figure 1.6 *Von Neumann architecture*

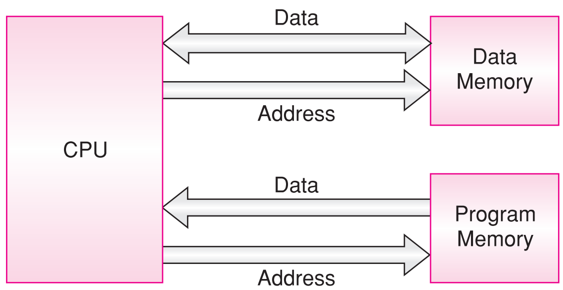


Figure 1.7 *Harvard architecture*

SECTION REVIEW

1. ____________ architecture contains single address and data bus.
2. In ____________ architecture, two memory access can be made in one instruction cycle.

1.5 RISC AND CISC MACHINES

The microcontrollers with small instruction set are called *Reduced Instruction Set Computer (RISC) machines* and those with complex instruction set are called *Complex Instruction Set Computer (CISC) machines*. Intel 8051 microcontroller is an example of CISC machine and Microchip PIC16F87X is an example of RISC machine. Comparison of features of RISC and CISC is shown in Table 1.2.

TABLE 1.2 *Comparison of RISC and CISC machines*

RISC	CISC
1. Instruction takes one or two cycles	1. Instruction takes multiple cycles
2. Only load/store instructions are used to access memory	2. In addition to load and store instructions, memory access is possible with other instructions also.
3. Instructions executed by hardware	3. Instructions executed by the micro program
4. Fixed format instructions	4. Variable format instructions
5. Few addressing modes	5. Many addressing modes
6. Few instructions	6. Complex instruction set
7. Most of them have multiple register banks	7. Single register bank

SECTION REVIEW

1. CISC instruction takes multiple cycles. True or False?
2. RISC has variable format instruction. True or False?
3. CISC has multiple register banks. True or False?
4. RISC has few addressing modes and less number of instructions. True or False?

1.6 COMPUTER SOFTWARE

A set of instructions written in a specific sequence for the computer to solve a specific task is called a *program*, and *software* is a collection of programs. The program stored in the computer memory in the form of binary numbers is called *machine instructions*. The machine language program is called *object code*.

For example, in 8051 microcontroller, the instruction CLR A is represented by binary code 11100100 (E4 in hexadecimal). Because it is difficult to write programs in sets of 0's and 1's, assembly language was then developed to simplify the programming job. Machine language and assembly language are low level languages and both are microprocessor specific. Assembly language programs are written using assembly instructions. An assembly instruction is the mnemonic representation of machine instructions.

For example, in the instruction CLR A, CLR stands for clear and A represents the accumulator. This symbol suggests the operation of storing 00H in the accumulator. The assembly language program that a programmer enters is called the *source program* or *source code*. A software program called an *assembler* is then developed to translate the program (source code) written in assembly language into machine language (object code), which is compatible with the microprocessor being used in the system as shown in Fig. 1.8. Mnemonics are specific to microprocessors and each microprocessor has its own assembler.

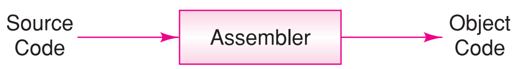


Figure 1.8 *Assembler*

The drawbacks in assembly language programming are:

- The programmer has to be very familiar with the processor of the computer in which the program is to be executed.
- It is difficult to understand an assembly language program without the use of comments.

To avoid the drawbacks of assembly language programming, high-level languages such as Fortran, Pascal, C and C++ were developed. A program written in high level language is also called a source program. A software program called a *compiler* or an *interpreter* was then developed to translate the source program into machine language as shown in Fig. 1.9.

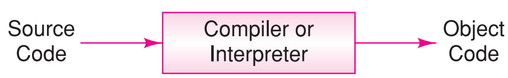


Figure 1.9 *Compiler/Interpreter*

Thus, a compiler or an interpreter translates high level language program into machine language. Each microprocessor needs its own compiler or an interpreter for each high level language. The primary difference between a compiler and an interpreter lies in the process of generating machine code. The compiler reads the entire program, translates it into object code and then it is executed by the processor. On the other hand, the interpreter takes one statement of a high level language program as input and translates it into object code and then executes. One of the major drawbacks of high level languages is that the machine code compiled from high level language program cannot run as fast as machine codes of assembly language program. For this reason, real time application programs are written in assembly language.

SECTION REVIEW

1. ___________ languages are low level languages.
2. In assembly language, programs are written using mnemonics. True or False?
3. ___________ software translates assembly language to object code.
4. List the drawbacks of assembly language programming.
5. Differentiate compiler and interpreter.

1.7 AN OVERVIEW OF EMBEDDED SYSTEM

In our daily lives, we are surrounded by number of embedded systems, such as TVs, DVD players, mobile phones, washing machines, digital cameras and automobiles. In this section, we will study definition of embedded systems, architecture of embedded systems, and embedded operating systems.

1.7.1 WHAT IS AN EMBEDDED SYSTEM?

An embedded system can be defined as a combination of computer hardware and software that does a specific job. Embedded systems are used in consumer electronics, food processing industry, chemical plants, cement plants, biomedical equipments, telecommunication and security. The embedded software that is executed for specific job is called *firmware*.

1.7.2 CHARACTERISTICS OF EMBEDDED SYSTEMS

Based on the processor and software, various types of embedded systems exist. Reliability, cost effective-ness, low power consumption, fast execution time, efficient use of memory, and processing power are the characteristics of most of the embedded systems.

Reliability Reliability of hardware and software is most important in embedded systems. The embedded system should reboot and rest by itself without human intervention during failure. The co-design of hardware and software is given importance in case of embedded system to meet these requirements.

Cost effectiveness The embedded systems are developed to meet the requirements of specific applications, in mass markets, and hence, keeping the product cost reasonable becomes essential. To meet the requirements, the system is developed using general-purpose processor during prototype. Then, an application specific integrated circuit is used to reduce the hardware and cost.

Low power consumption Many embedded systems are battery operated. Power consumption has to be limited to increase the life of the battery. This can be achieved by reducing the number of hardware components or by designing the processor to revert to sleep mode when no operation is to be performed.

Fast execution time In real time embedded systems, certain tasks must be performed within a specific time. To meet the performance constraints of real time systems, special operating systems known as real time operating systems run on these embedded systems. To fulfill the performance requirement, the code should be optimal. The software is generally developed using high-level languages, some computationally intensive units are developed in assembly language.

Efficient use of memory Most of the embedded systems contain only ROM and RAM without any secondary storage. Flash memory is used to store the program, including the operating system. Most of the microcontrollers and digital signal processors are available with on-chip flash memory.

Processing power The number of instructions executed per second is the processing power of the processor. Most of the processors execute one instruction in one clock cycle. Therefore, the processing power is in terms of million instructions per second (MIPS).

1.7.3 EMBEDDED SYSTEM ARCHITECTURE

Embedded system is a dedicated computer-based system for an application. The software is embedded in the read only memory. The hardware of an embedded system consists of the following six main components:

1. Central processing unit
2. Memory
3. Input unit
4. Output unit
5. Application specific circuitry
6. Communication channels

Central processing unit The central processing unit (CPU), which is also called processor is the heart of the embedded system. The CPU can be any of the following

- Microprocessor
- Microcontroller
- Digital signal processor
- Application specific processor

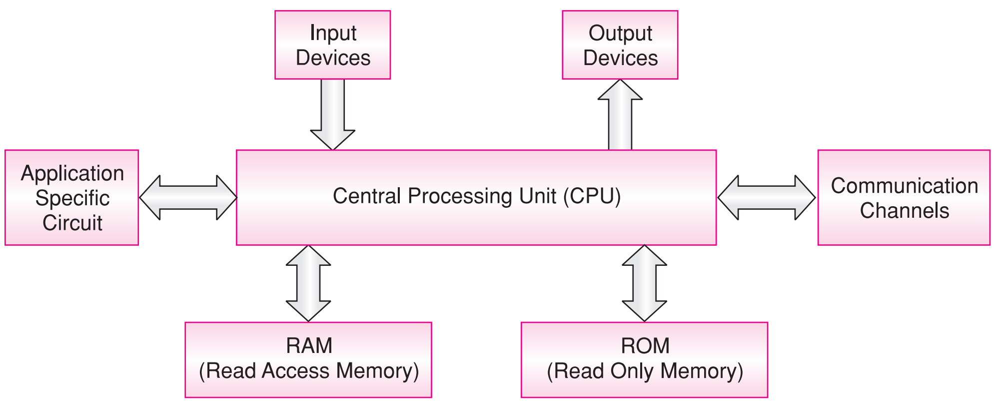


Figure 1.10 *Block diagram of an Embedded System*

Microprocessor Microprocessor is a single VLSI chip that has a CPU. It is used when large embedded software is to be executed. Microprocessors are powerful and result in faster processing of instructions. Table 1.3 lists the important microprocessors used in the embedded systems.

Microcontrollers Microcontrollers are also single VLSI chip, which have a CPU, memory, parallel ports, serial ports, analog to digital converters and timers. Microcontrollers are used when small-embedded software is to be executed and stored in its internal memory. Table 1.3 lists the important microcontrollers used in the embedded systems.

Digital signal processors Digital signal processors are used where signal processing is involved such as filtering, signal conversion from time to frequency domain, etc., and in applications like biomedical and speech signal processing. Table 1.3 lists the important digital signal processors used in the embedded systems.

Application specific processors In application specific processors, processors are designed for running the application specific tasks. They are the best choice where faster solution is required, as compared to microprocessor and microcontroller. Application specific processors are used in mobile phones, TV decoders, etc.

TABLE 1.3 *Important processors used in Embedded Systems*

Microprocessors	Microcontrollers	Digital Signal Processors
68HCxxx Motorola	68HC11xx Motorola	5600xx Motorola
80x86 Intel	8051, 80196, 89c61x2 Intel	TMS320Cxx Texas
SPARC SUN	PIC16F84, PIC16F874 Microchip	SHARC Analog Devices
Power PC IBM	ARM 7 ARM	ADSP 21xx Analog Devices

Memory An embedded system contains two types of semiconductor memory—random access memory (RAM) and read only memory (ROM). Random access memory stores user program and data. Firmware (embedded software) is stored in read only memory. When power is switched on, the processor executes the embedded software stored in read only memory.

Input Devices Some embedded systems have input devices like small keypad, analog to digital converters, switches whereas others do not have any input device for user interaction.

Output Devices Embedded systems have output devices like seven segment displays' light emitting diodes (LED) and liquid crystal displays (LCD) to display important parameters.

Application Specific Circuitry Depending on the application, embedded system contains transducers, sensors, amplifiers, and current to voltage conversion circuits. Function of these circuits is to convert incoming signal to match the range of Analog-to-Digital (A/D) converter.

Communication Interfaces An embedded system can interact with other embedded systems using serial communication bus like RS232, RS422 and IEEE bus.

1.7.4 FULL-CUSTOM PROCESSOR, SEMI-CUSTOM PROCESSOR AND PROGRAMMABLE LOGIC DEVICES

Processors are implemented on an integrated circuit (IC). IC, often called as chip consists of a set of transistors interconnected with other devices. There are different processes to fabricate semiconductors namely nMOS, pMOS and CMOS. The most widely used process is CMOS (complementary metal oxide semiconductor).

Semiconductors devices are built with different layers. The bottom-most layers are of transistors. The middle layers form logic components. The top-most layers connect these logic components with wires to build larger circuits like a processor.

Full-custom design Full-custom IC design is also referred to as Very Large Scale Integration (VLSI) design. Full-custom Design is the name given to the technique where the function and layout of every transistor is optimised. A microprocessor is an example of full custom IC. In this design, care has been taken to squeeze every last square of micron of chip space. Full-custom ICs are the most expensive to manufacture and to design.

Semi-custom design In case of semi-custom ICs, all of the logic cells are pre designed and some of the mask layers are custom designed. The masks for the transistor and gate levels are already built. The remaining design is to connect these gates to achieve a particular implementation. They provide good performance with much less cost than full-custom ICs.

Programmable logic devices In Programmable Logic Devices (PLDs), all of the logic cells are pre-designed and none of the mask layers are customised. There are two types in this category, namely Programmable Logic Devices (PLD) and Field Programmable Gate Array (FPGA). In case of PLDs, all layers already exist. One can purchase the IC and program it for intended design. The programming consists of creating or destroying connections between connected gates, either by blowing a fuse, or setting a bit in a programmable switch. A new inclusion to this family is Field Programmable Gate Arrays (FPGAs) that offer more general connectivity among blocks of logic, rather than just arrays of logic as with PLDs. These ICs are bigger in size, have higher unit cost and are slower compared to full-custom or semi-custom but provide reasonable performance with reduced design time.

1.7.5 DESIGN CHALLENGES IN AN EMBEDDED SYSTEM

The embedded system designers and developers have common design challenges like co-design, embedding an operating system, and optimising the code.

Co-design In an embedded system design, the task is implemented using hardware and software. The design engineer has to decide which module needs to be implemented in hardware and which module is to be realised by software. Hardware implementation imposes the constraints of size and cost. Software implementation imposes the constraints of memory size and performance. To design an efficient embedded system, co-design is a challenge.

Choice of operating system In the design of an embedded system, the designer has the option to develop the software in assembly and C without using an operating system. This will result in efficient code, saving processing power and memory. The other option is to use a readily available operating system, so that the designer can focus on application software development. This saves development time and cost.

Optimizing of code In the design of embedded system, memory and execution time are the main constraints. It is a challenge for the designer to optimise the code to meet the constraints.

1.7.6 EMBEDDED OPERATING SYSTEM

In the olden days, development of application embedded software was done in assembly language. Debugging and maintaining assembly language program were very difficult and time consuming. Hence, most of the development of application embedded software is now done using object oriented languages. Embedded operating systems are designed to work on semiconductor memory with limited processing power, since there is no secondary storage device in an embedded system. The advantages of using operating systems in embedded systems are

- The program can be written in a high-level language. Object oriented programming languages particularly C++ and Java are extensively used for embedded software development.
- The programmer can focus on application program rather than memory and I/O management program.

Operating systems used in embedded systems can be broadly divided into three types:

Non real time embedded operating systems These operating systems are suitable for soft real time embedded systems to perform tasks like memory management, I/O management, etc. They are not suitable for hard real time applications. Examples are embedded windows XP, and embedded Linux.

Real time operating systems These operating systems are suitable for hard real time operating systems. They are used to provide the necessary system calls for real time deadlines. Examples are OS/9, RT Linux, and Vx works.

Mobile/handheld operating systems Operating systems that are used to work in mobile environments are known as mobile operating systems. Examples are Embedded Windows NT, window CE, palm OS, and symfian OS.

In embedded systems, the operating system and the application software are integrated and stored in the memory of the embedded system.

1.7.7 REAL TIME EMBEDDED SYSTEMS

In most embedded systems, it is important to perform some of the computations in a timely manner. Such embedded systems in which strict deadlines are imposed to complete specific task are called *real time embedded systems*.

In real time systems, an additional programming is necessary to meet the deadline constraints in its functions. An operating system, which handles these multiple tasks with real time constraints, is called as *real time operating system* (RTOS). Note that real time system refers to embedded system that exhibits real time characteristics. For example, a cell phone decodes audio signal and converts digital signal to voice. All this takes place in a defined time period, else there will be a delay in reaching to the listener. Other systems that have timing requirements are process control system, robots, networks and multimedia systems.