Page - 3

CHAPTER 1

Topic - I

INTRODUCTION TO MICROCONTROLLERS

Objectives

- Understand the basic building blocks of a microcontroller device in general.
- Know the history and developments in microcontrollers and microprocessors
- Know the terminologies like embedded and external memory devices, CISC and RISC processors, etc.
- Know some commercial microcontrollers.

1.1 INTRODUCTION

Today, we see many industrial and domestic products like remote controllers, telephone bill printing machines, automatic power regulators, automatic or semi-automatic washing machines, microwave ovens, automobiles, engines, indicating and measuring instruments and similar products. Automation is needed to facilitate the process or mechanism for its operation and control. Data storage and processing is an integral part of any automatic control system. The need is to have a device, so called *'microcontroller'*, which allows controlling the timing and sequencing of these machines and processes. Further, with the help of microcontroller, it is possible to carry out simple arithmetic and logical operations. Any system that has a remote controller, almost certainly contains a microcontroller.

Microcontrollers are single-chip microcomputers, more suited for control and automation of machines and processes. Microcontrollers have central processing unit (CPU), memory, input/output ports (I/O), timers and counters, analog-to-digital converter (ADC), digital-to-analog converter (DAC), serial ports, interrupt logic, oscillator circuitry and many more functional blocks on chip. Figure 1.1 shows a general functional block diagram of a microcontroller. Note that there may be variations in the functional blocks from device to device and from one manufacturer to another. All these functional blocks on a single Integrated Circuit (IC), results into a reduced size of control board, low power consumption, more reliability and ease of integration within an application design. The usage of microcontrollers not only reduces the cost of automation, but also provides more flexibility. The designer is little bit relieved from the complex interfacing of external peripherals like ADC/DACs, etc. and can concentrate on applications and development aspects. The device can be programmed to make the system intelligent. This is possible because of the data processing and memory capability of microcontrollers. Once a typical microcontroller is learnt and practiced, one may wish to go for other microcontrollers from various makes. Some of the commonly used microcontrollers are Intel MCS-51, MCS-96, Motorola 68HC12 family, Microchip's peripheral interface controller (PIC) family of microcontrollers 16CXX, 17CXX, etc.

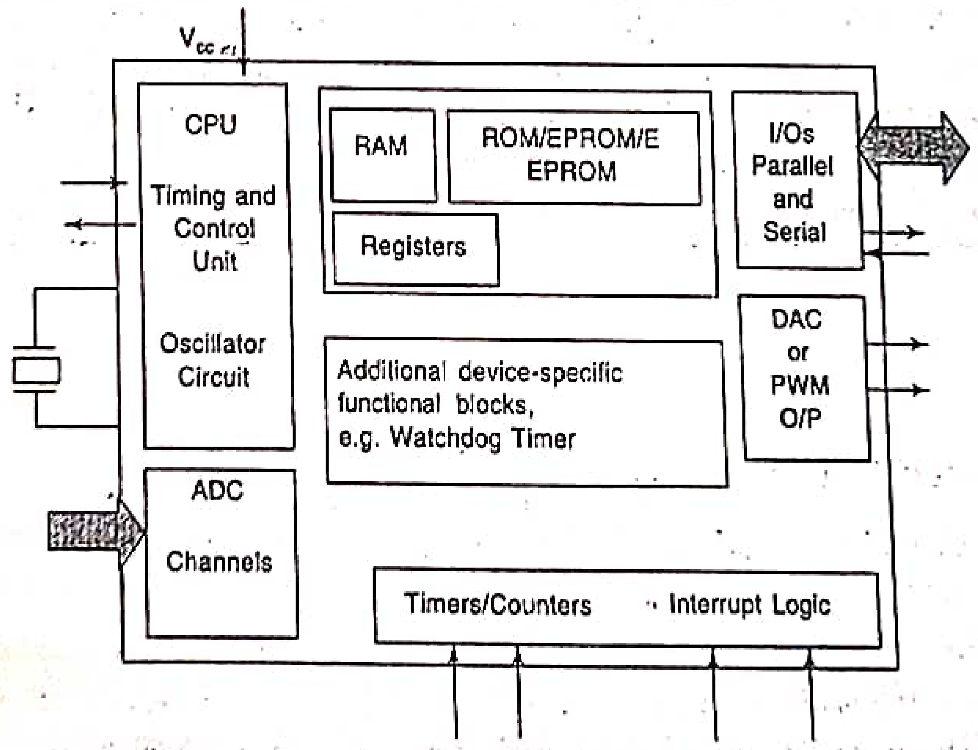

Fig. 1.1 : Generalized Functional Block Diagram of a Microcontroller

Microcontrollers have all the functional blocks which can fulfill the general needs of automation. Just to get an idea about the use of microcontrollers, consider an example of a temperature indicator as shown in the Fig. 1.2. A single microcontroller IC can directly get the analog input, say in the range 0–5 V typically, from a temperature sensor and drives LCD display. Note that there is no need to connect any external memory; the ADC function is also built inside the single microcontroller IC. The programmer has to write simple software modules to operate the LCD display and ADC. For example, the ADC operation involves the selection of ADC channel, generating the start of conversion command, checking the ADC status and getting the data into memory. The data may be averaged over some samples to improve the signal-to-noise ratio.

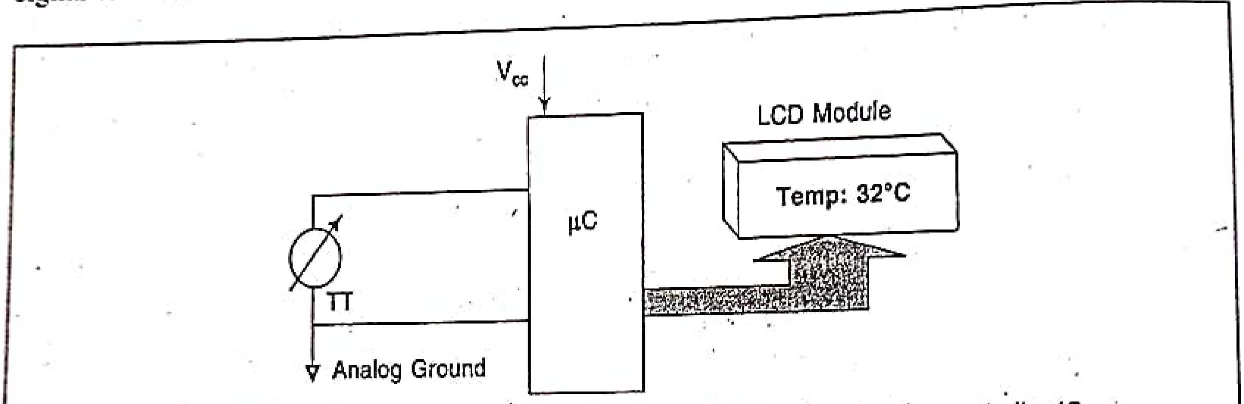

Fig. 1.2 Microcontroller-based Temperature Indicator

1.2 MICROCONTROLLERS AND MICROPROCESSORS

The word microprocessor, in a broader sense, means a CPU only. The functional blocks like memory, and other peripherals are to be connected externally to a microprocessor chip to form a complete microprocessor board. The system, which is built this way, is called as a "Single-Board Microcomputer". For example, 8085, 8086 and 80486 are microprocessors.

If we see the design requirements of automation, as discussed in the earlier section, we need a device which has all the functional blocks inside a single IC. Therefore, the concept of "single-chip" microcomputers came into reality. Single-chip microcomputers are 'Microcontrollers'. The examples are Intel MCS-51, PIC family by Microchip, Atmel 89CXX, 89CXX51. These are the microcontrollers used for general-purpose applications in the sense that they are user programmable and have functional blocks suitable to meet a more general design requirement. There are general-purpose and application-specific microcontroller products as well. Application-specific standard products (ASSPs) are tailored for a specific application but are not proprietary to a single customer, while general-purpose products are neither application nor customer specific. The application may be like embedded motor control. The job of application-specific microcontrollers in this particular application is mainly for controlling motors of any types; like DC, brushless DC, brushless permanent magnet, and AC induction motor.

Another important class of processors is the so called 'Bit-Slice Processors'. The term bit-ce processor means that the processors can be interconnected to form a processor of any esired word length. Bit slice processors consist of an ALU typically of 4 or 8 bits[1], registers nd control lines. The control lines are connected to each processor slice and all processors an perform the same operation. An example of a bit-slice processor is AMD's 2900 series. The bit-slice microprocessor design has some advantages. The first main advantage is that the ALUs can be assembled together horizontally to form computers that can handle very large data at a time. Another advantage of the bit-slice design is that it makes possible the use of bipolar chip technology that is very fast. Further, the bit-slice design allows users to create their own instruction sets for their applications. The discussions in this text will be focused only on the general-purpose microcontrollers and their applications.

1.3 HISTORY OF MICROCONTROLLERS AND MICROPROCESSORS

Since the inception of microprocessors, 4-, 8-, 16- and 32-bit microprocessors/ microcontrollers have developed and appeared in market. Intel 4004 was the first 4-bit processor which appeared in 1971. The instructions were 8 bits long but the data processed was 4-bit data. It had separate external memories for program (4K) and data (1K). There were 46 instructions and the clock frequency was 740 kHz. Then, during 1972, Intel 4040, the advanced version of 4004, was developed. 4040 had 14 more instructions with 8K program memory and interrupt capability.

In 1974, Texas Instruments introduced the first microcontroller TMS 1000. TMS 1000 had on-chip RAM, ROM, and I/Os. Intel 8080 was then introduced in 1974, as the advanced version of 8008, which also appeared in 1972. The popular, Intel 8085 was developed in 1976. This could operate on +5V supply and 3 MHz frequency. In the same year, Zilog Z-80 appeared as an improvement over Intel 8080. Z80 operating frequency was 2.5 MHz, and in the CMOS (Complementary Metal Oxide Semiconductor) version it was 10 MHz.

Immediately after Intel's 8080, in 1975, Motorola introduced its 6800, later followed by the 6502 and 6809, etc. Next, Intel 8048 was developed as a microcontroller around 1976. This is the MCS-48 family. On-chip data storage was possible, but the code was stored in the external program memory. It had 1-byte instructions. This was however immediately replaced by the MCS-51 family of microcontrollers in 1980. Intel MCS-51 uses more flexible 2-byte instructions. On-chip (RAM/ROM/EPROM) program memory as well as the 128 bytes of data memory is provided. External memory connections are also possible. In 1982, Motorola introduced microcontroller 6805. Peripheral interface controller (PIC) originated at Harvard University

[1]Bit is the basic unit of digital information. Four bits form a nibble and a byte consists of eight bits.

during 1975. PIC family of microcontrollers was introduced around 1985, by Microchip. PIC uses Harvard architecture, and has a Reduced Instruction Set. In 1978, Intel 8086, a 16-bit processor, was developed. Followed by this, Motorola introduced a 16-bit microprocessor 68000, and Zilog introduced Z8000, a 16-bit microprocessor. The historical developments in microprocessor products are listed in Table 1.1.

Table 1.1 Historical Developments in Microprocessors and Microcontrollers

Year	Microprocessor/microcontroller	Remark
1971	Intel 4004	4-bit microprocessor
1972	Intel 4040	4-bit microprocessor
1974	Intel 8080	8 bit microprocessor
	TMS 1000	8 bit microcontroller
1975	Motorola 6800	8 bit microprocessor
1976	MCS-48	8-bit microcontroller
	Intel 8085	8-bit microprocessor
1978	8086, Motorola 68000, Zilog Z-8000	16-bit microprocessor
1979		
1980	8051	8-bit microcontroller
1982	68010, 6805	Microcontroller
	80186, 80188, 80286	16-bit microprocessors
	8096 (MCS-96)	16-bit microcontrollers
1984	68020	32-bit microprocessor
1985	80386	32-bit microprocessor
	PIC microcontrollers by Microchip	8-bit microcontrollers
1987	Z280	16-bit microprocessor
1989	80386SX, 80486	32-bit microprocessor
1993	Pentium	32-bit microprocessor
1997	Atmel 8-bit AVR family	8-bit RISC microcontrollers

In June 1997, ATMEL 8-bit AVR microcontrollers were introduced which also have reduced instruction set. Today, we see a number of microcontroller families, from multiple sources and in many versions. Before discussing more about microcontrollers, let us look at some important terms related to microcontrollers.

1.4 EMBEDDED VERSUS EXTERNAL MEMORY DEVICES

Embedded devices are becoming very popular now-a-days. Such a device has all the functional blocks on chip, including the program and data memory. There is no external data/address bus provided. For example, ATMEL 89C2051 is one example of Embedded Controller, which has timers/counters, on-chip RAM, EEPROM, I/Os, a precision comparator along with CPU and timing and control unit. It has only 20 pins. The code is executed from the internal program memory only. One can see that this type of IC is definitely better than the control boards in earlier designs, which used many logic gates and other digital ICs.

However, we see normally that devices like 8031 from MCS-51 family need external program memory interface. These devices are External Memory Devices. In case of 8051/8751, a part of program execution is possible from the on-chip program memory and rest of the program execution is from the external memory. Further, there is an option of executing the whole program only from the external program memory as well. This feature of external memory is useful, only when the program memory requirement exceeds the one which is available on-chip. In external memory devices, there is an external address/data bus. Naturally, the number of pins is more and an additional Printed Circuit Board (PCB) space is needed for memories and latches, etc.

1.5 8-BIT AND 16-BIT MICROCONTROLLERS

Several features define the word length of a processor. One can define an 8-bit microprocessor or microcontroller in a much broader sense, as a device which has most of its registers 8-bit wide and most of its instructions use operands which are 8-bit wide. There is almost no direct dependency of this definition on the width of the data or address bus. For example, 8088 has 8-bit external data bus, but it is a 16-bit processor. 8085 is an 8-bit microprocessor. MCS-51 is an 8-bit microcontroller family. MCS-96 is a 16- bit microcontroller family.

From the application point of view, it is very important to decide if an 8-bit or a 16-bit microcontroller is to be used in a typical design. 8-bit microcontrollers have dominated over the 16-bit microcontrollers. The reason is that many designers are familiar with 8-bit microcontrollers and we can always perform 16-bit operations on 8-bit controllers, by writing suitable programs.

1.6 CISC AND RISC PROCESSORS

Complex Instruction Set Computers (CISC) and Reduced Instruction Set Computers (RISC) are common terminologies used while talking about microcontrollers or microprocessors. CISC processors have large number of instructions. A larger instruction set helps assembly language programmers by providing flexibility to write effective and short programs. The objective of CISC architecture is to write a program in as few lines of assembly language code as possible. This is made possible by developing processor hardware that can understand and execute a number of operations. For example, in MCS-51, the MUL (multiply) instruction is a complex instruction for which only operands are to be specified in the instruction, and the operation of multiplication is done by hardware. In such type of instructions, the building of complex instructions directly into the hardware helps in two different ways. Not only is the hardware implementation faster, but also it saves the program memory space in the sense that the instruction code is very short as compared to the one needed for the multiply operation using ADD instruction. Thus, the programmer works at a relatively higher level.

Programmers want to have fewer, simpler and faster instructions, than the large, complex and slower CISC instructions. This is, however, at the cost of writing more instructions to accomplish a task. One advantage of RISC is that because of the more simple instructions, RISC chips require lesser hardware implementation, which makes them simpler to design and hence lesser cost of production. And it is always easier to write optimized compilers, because of a small number of instructions. One example of RISC microcontrollers is the popular PIC family of microcontrollers by Microchip.

More importance is given to how fast a chip can execute the instructions and how it runs available or designed software.

1.7 HARVARD AND VON NEUMANN ARCHITECTURES

There are two major classes of computer architectures, namely, 'Harvard architectures', and the 'Von Neumann (or Princeton) architectures'. Figure 1.3 shows the Harvard architecture. Many special designs of microcontrollers and DSP (Digital Signal Processor) use Harvard architectures.

Harvard architecture uses separate memories for program and data with their independent address and data buses. Because of two different streams of data and address, there is no need to have any time division multiplexing of address and data buses. Not only the architecture supports parallel buses for address and data, but also it allows a different internal organization such that instruction can be pre-fetched and decoded while multiple data are being fetched and operated on. Further, the data bus may have different size than the address bus. This allows the optimal bus widths of the data and address buses for fast execution of the instruction. For example, the MCS-51 family of microcontrollers by Intel has Harvard architecture because there are different memory spaces for program and data and separate (internal) buses for address and data. Similarly, PIC microcontrollers by Microchip use Harvard architecture.

In Von Neumann architecture, programs and data share the same memory space. Figure 1.4 shows the Von Neumann architecture. Von Neumann architecture allows storing or modifying

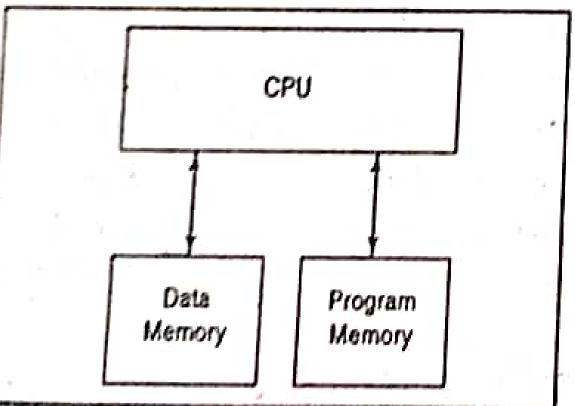

Fig. 1.3 Harvard Architecture

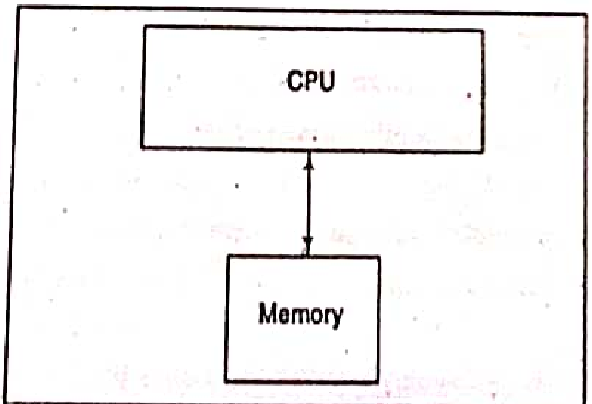

Fig. 1.4 Von Neumann Architecture

the programs easily. However, the code storage may not be optimal and requires multiple fetches to form the instruction. Program and data fetches are done using time division multiplexing which affect the performance. One example of the microcontroller using the Von Neumann (Princeton) architecture is Motorola 68HC11 microcontroller.

1.8 COMMERCIAL MICROCONTROLLER DEVICES

For a given application, it is necessary to find out the functional needs and select a suitable microcontroller. One should be aware of some of the commercially available microcontrollers. A brief overview of some commercial microcontrollers is given below. The families, which we will discuss in this book, are PIC by Microchip, Intel MCS-51, and Atmel 89CXX /89CXX51.

1.8.1 MCS-51 and Atmel 89CXX, 89CXX51 Microcontrollers

MCS-51 and Atmel 89CXX, 89CXX51 microcontrollers are 8-bit microcontrollers. Atmel 89CXX and 89CXX51 are compatible with the MCS-51 family. Therefore, while learning MCS-51, one can learn these Atmel microcontrollers. Let us consider MCS-51 as a starting point for learning the microcontrollers for which the literature and development systems are already available in many institutions and industries where some work on microcontrollers is already going on. However, today one can get these development tools and application notes easily from the manufacturers, for the other microcontroller families as well.

1.8.2 The MCS-51 Family

MCS-51 is the family of 8-bit microcontrollers, operating at the frequency of 12 MHz, and it was introduced to replace the earlier MCS-48 microcontrollers. Table 1.2 lists the various devices of this family. The design is based on HMOS (High-Speed Metal Oxide Semiconductor) technology. CHMOS (Complementary High-Speed Metal Oxide Semiconductor) versions of these devices are also available and are represented by the part number with an additional letter 'C', e.g. 80C51, 87C51, etc. CHMOS is the name given to Intel's high-speed CMOS process. There are some advantages of using these CHMOS versions of MCS-51 microcontrollers over the HMOS versions. CHMOS version has lower power consumption, higher noise immunity and higher speed. The CHMOS devices are architecturally compatible with their HMOS counterparts. The only difference is that the CHMOS devices have additional features like power down and idle modes, etc.

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
8031	[illegible]	[illegible]	[illegible]	[illegible]	1
8032	[illegible]	[illegible]	[illegible]	6	1
8051	128	4K ROM	[illegible]	[illegible]	1
8052	256	8K ROM	3	6	1
8751	128	4K EPROM	2	[illegible]	1
8752	256	8K EPROM	3	6	1

The selection criteria among these are only based on the functional requirements power supply constraints, available printed circuit board space, pin count and the cost of production. For a mass production, the designer may go for ROM versions. However, for lesser production, EPROM version or CPU version with external program memory is suitable. Many a times it is preferred to go for the EPROM/ROM versions to avoid the external memory connections, provided the size of the code is less than the size of on-chip program memory. However, one should note that the EPROM has limited write/erase cycles, typically 1000 write/erase. From maintenance point of view, one may prefer EPROM or ROM version, because it requires replacing only a single IC from the board. Further, during the product development, one spends a lot of intellectual man-hours to design software. So, it is better to use the built-in 'security fuse' facility to avoid stealing or unauthorized copying of software. For the development work, an EPROM eraser and programmer is needed to write or change the code, along with the testing and debugging tools.

1.8.3 Atmel Microcontrollers

Few devices from the Atmel family are listed in Table 1.3. Mainly we discuss here about the reprogrammable flash devices. The emphasis is given on 20-pin devices. These devices support fully static operation from 0 to 24 MHz. The low-frequency operation is very important when the power consumption is to be kept minimum. For example, in case of the battery operated instruments, power consumption is crucial. Note, however that the execution speed is also reduced at low frequencies. Timer-clocking periods are also dependent on the operating frequency. The power-down modes and IDLE modes can be used to keep the power consumption to a minimum level. One can select a device with low-voltage operation, typically in the range 2.7–6 V. Atmel 89C1051/2051 devices support this low-voltage operation. For Atmel 89C4051, this range is 3–6 V.

Table 1.3 Atmel Microcontrollers

Device	On-chip data memory (bytes)	On chip program memory (flash)	No. of 16-bit timers/counters	Digital I/Os	Full duplex serial I/Os	No. of pins in (PDIP)	Precision on-chip analog comparators
AT89C51	128	4K	2	32	1	40	None
AT89C52	256	8K	3	32	1	40	None
AT89C55WD	256	20K	3	32	1	[illegible]	None
AT89C1051	64	1K	2	15	1	20	1
AT89C2051	128	2K	2	15	1	20	1
AT89C4051	128	4K	2	15	1	20	1
AT89LV52	256	8K	3	32	1	40	None

Twenty-pin devices are very much suited when the PCB space available is small, and the need of I/O lines is fulfilled with 15 I/Os. The precision analog comparator along with timers can be used for building the counter type ADC. Flash PEROM (Programmable and Erasable Read Only Memory) is another very useful feature. Ultraviolet EPROM eraser is not needed at all. The device can be programmed and erased electrically. Atmel 89LV52 is a low-voltage 8K flash microcontroller compatible with 8032. The operating voltage range is 2.7 V–6 V. For more details one may refer the Atmel microcontroller product literature. It is also interesting to see the Atmel AVR family of RISC processors.

1.8.4 PIC Microcontrollers

PIC16CXX and PIC17CXX, 8-bit microcontrollers by Microchip, use CMOS technology. PIC microcontrollers are popular because of high performance, low cost and small size. It uses high-speed RISC architecture. PIC 16CXX has only 33 single-word instructions. The operating frequency typically for 16CXX parts ranges from DC to 20 MHz. It is possible to add external program memory, up to 64K words. PIC 17C42 has a number of counter/timer resources and I/O handling capabilities. 16C71 has built in 8-bit ADC with 4 channels.

To have a 10-bit ADC with 12 channels, one can go for 17C752 device. PIC microcontroller devices are briefly described in Table 1.4. There are many more devices by Microchip in many versions, which are not listed in the table. General features include, timers, watchdog timers, embedded ADC, extended instruction/data memory, serial communication, pulse width modulated (PWM) outputs, and ROM, EPROM and EEPROM memories.