

Introduction to Microcontrollers

1.1 INTRODUCTION

Microcontrollers have only been with us for a few decades but their impact (direct or indirect) on our lives is profound. Usually these are supposed to be just *data processors performing exhaustive numeric operations*. But their presence is unnoticed at most of the places like

- *At supermarkets* in Cash Registers, Weighing Scales, etc.
- *At home* in Ovens, Washing Machines, Alarm Clocks, etc.
- *At play* in Toys, VCRs, Stereo Equipment, etc.
- *At office* in Typewriters, Photocopiers, Elevators, etc.
- *In industry* in Industrial Automation, safety systems, etc.
- *On roads* in Cars, Traffic Signals, etc.

What inside them makes these machines “smart”? The answer is microcontroller.

Creating applications for the microcontrollers is different than any other development job in electronics and computing. Before selecting a particular device for an application, it is important to understand what the different options and features are and what they can mean with regard to developing the application.

The purpose of this chapter is to introduce the concept of a microcontrollers, how it differ from microprocessors, different type of commercial microcontrollers available as well as their applications. The reminder of the book will go through and present different types of microcontrollers and also programming and interfacing techniques of microcontroller, mainly 8051, in detail.

1.2 EMBEDDED CONTROLLER

Simply an embedded controller is a controller that is embedded in a greater system. One can define an embedded controller as a controller (or computer) that is embedded into some device for some purpose other than to provide general purpose computing.

Is an embedded controller is the same as a microcontroller? The answer is definitely no. One can state devices such as 68000, 32032, x86, Z80, and so on that are used as embedded controllers but they aren't microcontrollers.

We might be correct by stating that an embedded controller controls something (for example controlling a device such as a microwave oven, car braking system or a cruise missile). An embedded controller may also embed the on-chip resources like a microcontroller. Microcontrollers and microprocessors are widely used in embedded systems. Though microcontrollers are preferred over microprocessors for embedded systems due to low power consumption.

1.3 MICROCONTROLLERS AND MICROPROCESSORS

A controller is used to control some process. At one time, controllers were built exclusively from logic components, and were usually large, heavy boxes. Later on, microprocessors were used and the entire controller could fit on a small circuit board. This is still common—one can find many controllers powered by one of the many common microprocessors (including Zilog Z80, Intel 8088, Motorola 6809, and others).

As the process of miniaturization continued, all of the components needed for a controller were built right onto one chip. A one chip computer, or microcontroller was born.

A CPU built into a single VLSI chip is called microprocessor. The simplified block diagram of the CPU is shown in the Fig. 1.1. It contains arithmetic and logic unit (ALU), Instruction decode and control unit, Instruction register, Program counter (PC), clock circuit (internal or external), reset circuit (internal or external) and registers. For example, Intel 8085 is 8-bit microprocessor and Intel 8086/8088 is 16-bit microprocessor.

Microprocessor is general-purpose digital computer central processing unit (CPU). The microprocessor is general-purpose device and additional external circuitry are added to make it microcomputer.

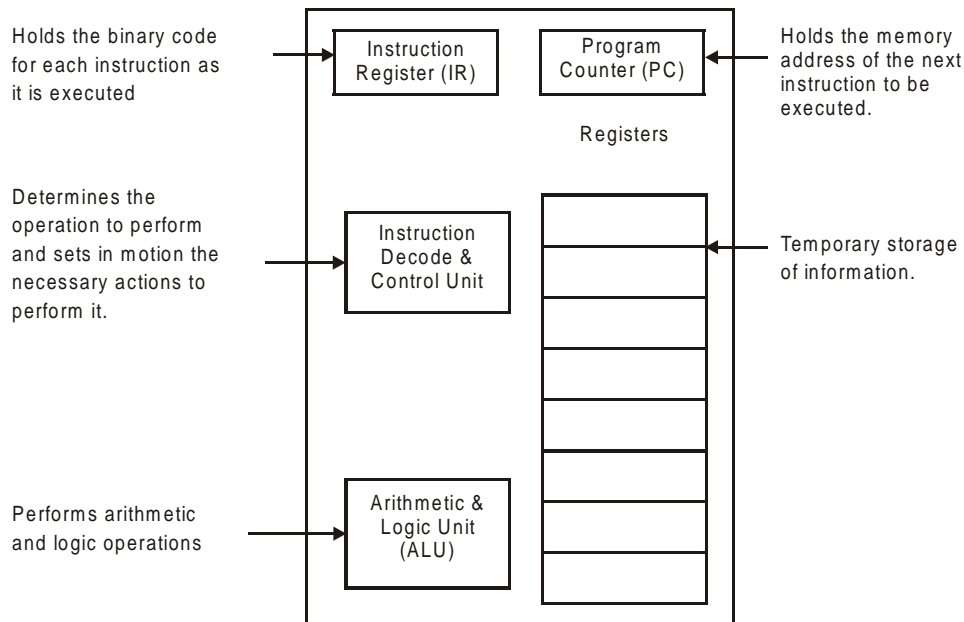


Figure 1.1 General block diagram of CPU (Microprocessor)

A digital computer having microprocessor as the CPU along with I/O devices and memory is known as microcomputer. The block diagram in the Fig. 1.2 shows a microcomputer.

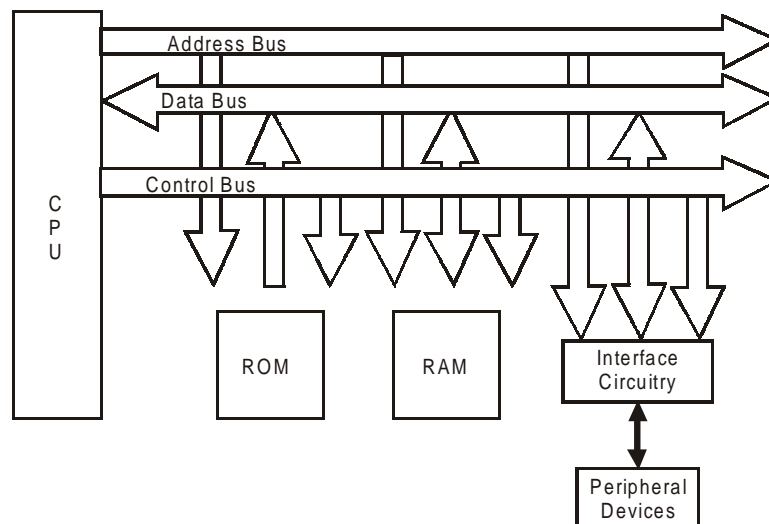


Figure 1.2 Microcomputer block diagram

A microcontroller is a highly integrated chip, which includes on single chip, all or most of the parts needed for a controller. The microcontroller typically includes: CPU (Central Processing Unit), RAM (Random Access Memory), EPROM/PROM/ROM (Erasable Programmable Read Only Memory), I/O (input/output) – serial and parallel, timers, interrupt controller. For example, Intel 8051 is 8-bit microcontroller and Intel 8096 is 16-bit microcontroller.

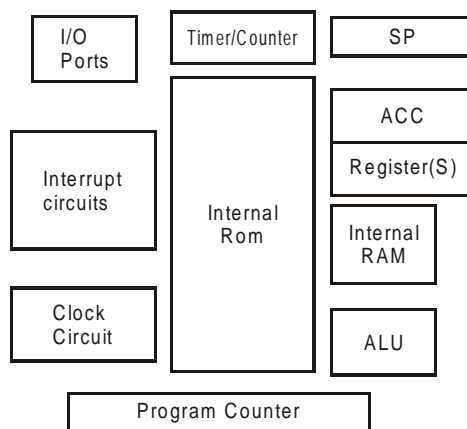


Figure 1.3 A block diagram of a microcontroller

By only including the features specific to the task (control), cost is relatively low. A typical microcontroller has bit manipulation instructions, easy and direct access to I/O (input/output), and quick and efficient interrupt processing. Figure 1.3 shows the block diagram of a typical microcontroller.

COMPARING MICROPROCESSORS AND MICROCONTROLLERS

- Microprocessor is a single chip CPU, microcontroller contains, a CPU and much of the remaining circuitry of a complete microcomputer system in a single chip.
- Microcontroller includes RAM, ROM, serial and parallel interface, timer, interrupt schedule circuitry (in addition to CPU) in a single chip.
 - RAM is smaller than that of even an ordinary microcomputer, but enough for its applications.
 - Interrupt system is an important feature, as microcontrollers have to respond to control oriented devices in real time. E.g., opening of microwave oven's door cause an interrupt to stop the operation. (Most microprocessors can also implement powerful interrupt schemes, but external components are usually needed).
- Microprocessors are most commonly used as the CPU in microcomputer systems. Microcontrollers are used in small, minimum component designs performing control-oriented activities.
- Microprocessor instruction sets are “processing intensive”, implying powerful addressing modes with instructions catering to large volumes of data. Their instructions operate on nibbles, bytes, etc. Microcontrollers have instruction sets catering to the control of inputs and outputs. Their instructions operate also on a single bit. E.g., a motor may be turned ON and OFF by a 1-bit output port.

Before going in to details of microcontrollers it will be beneficial to go through common and frequently used terminology encountered in the description of microcontrollers.

CENTRAL PROCESSING UNIT (CPU)

CPU is the brain of the computer system, administers all activity in the system and performs all operations on data. It continuously performs two operations: *fetching* and *executing* instructions. It understand and execute instructions based on a set of binary codes called the instruction set.

Machine Cycle

To execute an instruction–the processor must:

1. Fetch the instruction from memory
2. Decode the instruction

3. Execute the instruction
4. Store the result back in the memory. These four steps refer to Machine Cycle.

Generally one machine cycle = X clock cycles ("X" depends on the particular instruction being executed). Shorter the clock cycle, lesser the time it takes to complete one machine cycle, so instructions are executed faster. Hence, faster the processor.

FETCHING AND EXECUTING AN INSTRUCTION

Fetching involves the following steps:

- (a) Contents of PC are placed on address bus.
- (b) READ signal is activated.
- (c) Data (instruction opcode) are read from RAM and placed on data bus.
- (d) Opcode is latched into the CPU's internal instruction register.
- (e) PC is incremented to prepare for the next fetch from memory.

While execution involves decoding the opcode and generating control signals to gate internal registers in and out of the ALU and to signal the ALU to perform the specified operation.

THE BUSES: ADDRESS, DATA, AND CONTROL

A BUS is a collection of wires carrying information with a common purpose. For each *read* or *write* operation, the CPU specifies the location of the data or instruction by placing an address on the address bus, then activates a signal on the control bus indicating whether the operation is read or write.

–READ OPERATIONS retrieve a byte of data from memory at the location specified and place it on the **data bus**. CPU reads the data and places it in one of its internal registers.

–WRITE OPERATIONS put data from CPU on the data bus and store it in the location specified.

ADDRESS BUS carries the address of a specified location. For n address lines, 2^n locations can be accessed. E.g., A 16-bit address bus can access $2^{16} = 65,536$ locations or 64K locations ($2^{10} = 1024 = 1K$, $2^6 = 64$).

DATA BUS carries information between the CPU and memory or between the CPU and I/O devices.

CONTROL BUS carries control signals supplied by the CPU to synchronize the movement of information on the address and data bus.

CONTROL/MONITOR (INPUT/OUTPUT) DEVICES

CONTROL DEVICES are *outputs*, or *actuators*, that can affect the world around them when supplied with a voltage or current.

MONITORING DEVICES are *inputs*, or *sensors*, that are stimulated by temperature, pressure, light, motion, etc. and convert this to voltage or current read by the computer.

Note: The interface circuitry converts the voltage or current to binary data, or vice versa.

1.4 TYPES OF MICROCONTROLLERS

Microcontrollers can be classified on the basis of internal bus width, architecture, memory and instruction set. Figure 1.4 shows the various types of microcontrollers.

1.4.1 THE 8,16 AND 32-BIT MICROCONTROLLERS

THE 8-BIT MICROCONTROLLER

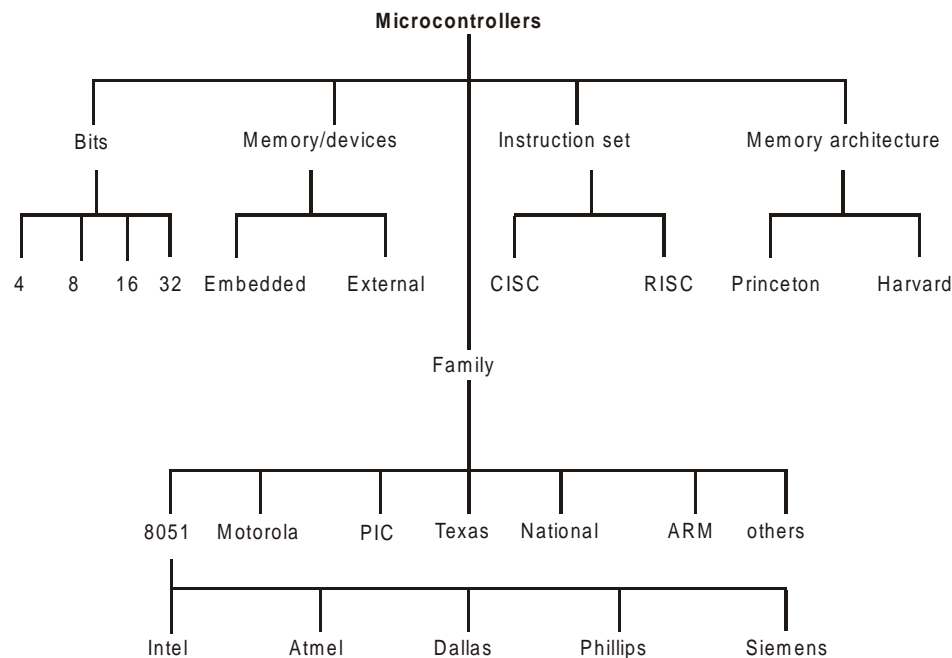
When the ALU performs arithmetic and logical operations on a byte (8-bits) at an instruction, the microcontroller is an 8-bit microcontroller. The internal bus width of 8-bit microcontroller is of 8-bit. Examples of 8-bit microcontrollers are Intel 8051 family and Motorola MC68HC11 family.

THE 16-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a word (16-bits) at an instruction, the microcontroller is an 16-bit microcontroller. The internal bus width of 16-bit microcontroller is of 16-bit. Examples of 16-bit microcontrollers are Intel 8096 family and Motorola MC68HC12 and MC68332 families. The performance and computing capability of 16 bit microcontrollers are enhanced with greater precision as compared to the 8-bit microcontrollers.

THE 32-BIT MICROCONTROLLER

When the ALU performs arithmetic and logical operations on a double word (32-bits) at an instruction, the microcontroller is an 32-bit microcontroller. The internal bus width of 32-bit microcontroller is of 32-bit. Examples of 32-bit microcontrollers are Intel 80960 family and Motorola M683xx and Intel/Atmel 251 family. The performance and computing capability of 32 bit microcontrollers are enhanced with greater precision as compared to the 16-bit microcontrollers.

**Figure 1.4** Types of microcontrollers

1.4.2 EMBEDDED AND EXTERNAL MEMORY MICROCONTROLLERS

EMBEDDED MICROCONTROLLERS

When an embedded system has a microcontroller unit that has all the functional blocks (including program as well as data memory) available on a chip is called an embedded microcontroller. For example, 8051 having Program & Data Memory, I/O Ports, Serial Communication, Counters and Timers and Interrupt Control logic on the chip is an embedded microcontroller.

EXTERNAL MEMORY MICROCONTROLLERS

When an embedded system has a microcontroller unit that has not all the functional blocks available on a chip is called an external memory microcontroller. In external memory microcontroller, all or part of the memory units are externally interfaced using an interfacing circuit called the glue circuit. For example, 8031 has no program memory on the chip is an external memory microcontroller.

1.4.3 MICROCONTROLLER ARCHITECTURAL FEATURES

There are mainly two categories of processors, namely, Von-Neuman (or Princeton) architecture and Harvard Architecture. These two architectures differ in the way data and programs are stored and accessed.

1.4.3.1 VON-NEUMAN ARCHITECTURE

Microcontrollers based on the Von-Neuman architecture have a single “data” bus that is used to fetch both instructions and data. Program instructions and data are stored in a common main memory. When such a controller addresses main memory, it first fetches an instruction, and then it fetches the data to support the instruction. The two separate fetches slows up the controller’s operation. Figure 1.5 shows the Von-Neuman Architecture. The Von-Neuman architecture’s main advantage is that it simplifies the microcontroller design because only one memory is accessed. In microcontrollers, the contents of RAM can be used for data storage and program instruction storage. For example, the Motorola 68HC11 microcontroller Von-Neuman architecture.

Example : An Instruction “Read a byte from memory and store it in the accumulator” as follows:

Cycle 1 :- Read instruction

Cycle 2 - Read data out of RAM and put into Accumulator

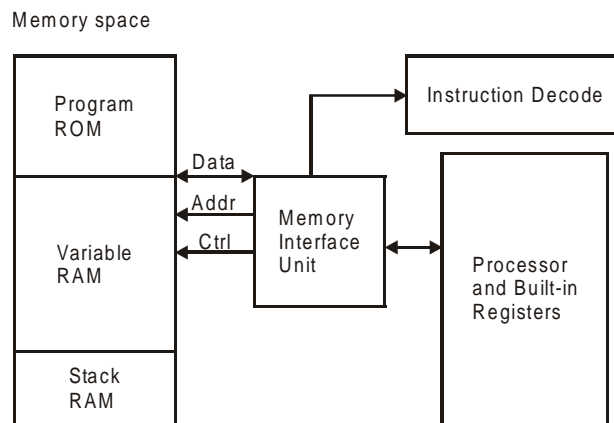


Figure 1.5 Von-neuman architecture block diagram

1.4.3.2 HARVARD ARCHITECTURE

Microcontrollers based on the Harvard Architecture have separate data bus and an instruction bus. This allows execution to occur in parallel. As an instruction is being “pre-fetched”, the current instruction is executing on the data bus. Once the current instruction is complete, the next instruction is ready to go. This pre-fetch theoretically allows for much faster execution than Von-Neuman architecture, on the expense of complexity. Figure 1.6 shows the Harvard Architecture. The Harvard Architecture executes instructions in fewer instruction cycles than the Von-Neuman architecture. For

example, the intel MCS-51 family of microcontrollers and PIC microcontrollers use Harvard Architecture.

The same instruction (as shown under Von-Neuman architecture) would be executed as follows:

- Cycle 1: - Complete previous instruction
 - Read the “Move Data to Accumulator” instruction
- Cycle 2: - Execute “Move Data to Accumulator” instruction
 - Read next instruction

Hence each instruction is effectively executed in one instruction cycle.

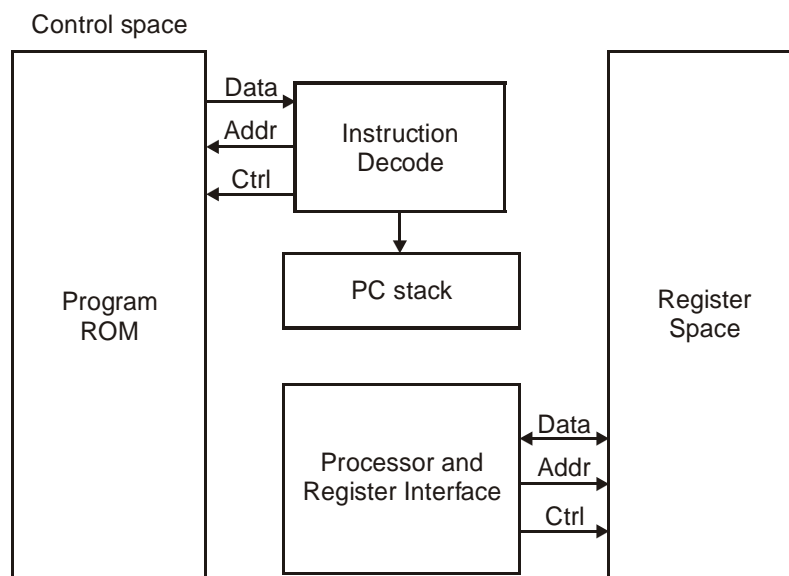


Figure 1.6 Harvard architecture block diagram

1.4.3.3 CISC (COMPLEX INSTRUCTION SET COMPUTER) ARCHITECTURE MICROCONTROLLERS

Almost all of today's microcontrollers are based on the CISC (Complex Instruction Set Computer) concept. When a microcontroller has an instruction set that supports many addressing modes for the arithmetic and logical instructions, data transfer and memory accesses instructions, the microcontroller is said to be of CISC architecture.

The typical CISC microcontroller has well over 80 instructions, many of them very powerful and very specialized for specific control tasks. It is quite common for the instructions to all behave quite differently. Some might only operate on certain address spaces or registers, and others might only recognize certain addressing modes.

The advantages of the CISC architecture are that many of the instructions are macro-like, allowing the programmer to use one instruction in place of many simpler instructions. An example of CISC architecture microcontroller is Intel 8096 family.

1.4.3.4 RISC (REDUCED INSTRUCTION SET COMPUTER) ARCHITECTURE MICROCONTROLLERS

The industry trend for microprocessor design is for Reduced Instruction Set Computers (RISC) designs. When a microcontroller has an instruction set that supports fewer addressing modes for the arithmetic and logical instructions and for data transfer instructions, the microcontroller is said to be of RISC architecture.

The benefits of RISC design simplicity are a smaller chip, smaller pin count, and very low power consumption.

Some of the typical features of a *RISC processor- Harvard architecture* are

1. Allows simultaneous access of program and data.
2. Overlapping of some operations for increased processing performance.
3. Instruction pipelining increases execution speed.
4. Orthogonal (symmetrical) instruction set for programming simplicity.
5. Allows each instruction to operate on any register or use any addressing mode.

1.4.3.5 SISC (SPECIFIC INSTRUCTION SET COMPUTER)

Actually, a microcontroller is by definition a Reduced Instruction Set Computer. It could really be called a Specific Instruction Set Computer (SISC). The basic idea behind the microcontroller was to limit the capabilities of the CPU itself, allowing a complete computer (memory, I/O, interrupts, etc) to fit on the single chip. At the expense of the more general purpose instructions that make the standard microprocessors (8088, 68000, 32032) so easy to use, the instruction set was designed for the specific purpose of control (powerful bit manipulation, easy and efficient I/O, and so on).

1.5 MICROCONTROLLER APPLICATIONS

In addition to control applications such as the home monitoring system, microcontrollers are frequently found in embedded applications. Among the many uses that you can find one or more microcontrollers: automotive applications, appliances (microwave oven, refrigerators, television and VCRs, stereos), automobiles (engine control, diagnostics, climate control), environmental control (greenhouse, factory, home), instrumentation, aerospace, and thousands of other uses.

Microcontrollers are used extensively in robotics. In this application, many specific tasks might be distributed among a large number of microcontrollers in one system. Communications between each microcontroller and a central, more powerful microcontroller (or microcomputer, or even large computer) would enable information to be processed by the central computer, or to be passed around to other microcontrollers in the system.

A special application that microcontrollers are well suited for is data logging. By stick one of these chips out in the middle of a corn field or up in a balloon, one can monitor

and record environmental parameters (temperature, humidity, rain, etc). Small size, low power consumption, and flexibility make these devices ideal for unattended data monitoring and recording.

1.6 COMMERCIAL MICROCONTROLLER DEVICES

Microcontrollers come in many varieties. Depending on the power and features that are needed, one might choose a 4 bit, 8 bit, 16 bit, or 32 bit microcontroller. In addition, some specialized versions are available which include features specific for communications, keyboard handling, signal processing, video processing, and other tasks. The examples of different types of commercial microcontroller devices are given in the following tables.

Table 1.1 4 Bit Microcontrollers

Model (Manufacturer)	I/O	Pins	RAM (bytes)	ROM (bytes)	Counters	Extra Features
COP400 Family (National)	23	28	64	1K	1	Serial bit I/O
HMCS40 (Hitachi)	10	28	32	512	-	10-bit ROM
TMS 1000 (Texas Instruments)	23	28	64	1K	-	LED display

Table 1.2 8 Bit Microcontrollers

Model (Manufacturer)	I/O	Pins	RAM (bytes)	ROM (bytes)	Counters	Extra Features
8048 (Intel)	27	40	64	1K	1	8k External memory
8051 (Intel)	32	40	128	4K	2	128k External memory, Boolean processing, serial port
COP800 Family (National)	24	28	64	1K	1	Serial bit I/O, 8- channel A/D converter
6805 (Motorola)	20	28	64	1K	1	PLL frequency synthe- sizer,
68hc11(Motorola)	40	52	256	8K	2	A/D, PWM generator, pulse accumulator
TMS370 (Texas)	55	68	256	4K	2	watchdog timer, Instru- ments) Serial ports, A/D (8 bit, 8 channel)
PIC (Micro Chip)	12	18	25	1K	0	small pin count, very low power consumption

Table 1.3. 16 Bit Microcontrollers

Model (Manufacturer)	I/O	Pins	RAM (bytes)	ROM (bytes)	Counters	Extra Features
80c196 (Intel)	40	68	232 8K	2		PWM generator, watchdog timer
HPC Family (National)	52	68	512 16K	4		PWM generator, watchdog timer, 8-channel A/D, serial port

1.7 PROBLEMS

1. Write major differences between a microprocessor and a microcontroller.
2. Discuss the evolution of microcontrollers.
3. What are the applications of microcontrollers?
4. Discuss the advantages and disadvantages of Harvard and Princeton architectures.
5. Discuss the advantages of microcontrollers over microprocessors in control operations.
6. What are the different ways of classifying the types of microcontrollers?
7. How does a microcontroller differ from a microprocessor?
8. What are the RISC and CISC microcontrollers?
9. Discuss microcomputer system. Is microcomputer as same as microcontroller?
10. What is the difference between an embedded and external memory microcontroller?