Microcontroller

A microcontroller is a computer present in a single integrated circuit which is dedicated to perform one task and execute one specific application.

It contains memory, programmable input/output peripherals as well a processor. Microcontrollers are mostly designed for embedded applications and are heavily used in automatically controlled electronic devices such as cellphones, cameras, microwave ovens, washing machines, etc.

A microcontroller is abbreviated as µC, uC or MCU.

Advantages of Microcontrollers

a) Microcontrollers act as a microcomputer without any digital parts.
b) As the higher integration inside microcontroller reduces cost and size of the system.
c) Usage of a microcontroller is simple, easy to troubleshoot and system maintaining.
d) Most of the pins are programmable by the user for performing different functions.
e) Easily interface additional RAM, ROM,I/O ports.
f) Low time required for performing operations.

Disadvantages of Microcontrollers

a) Microcontrollers have got more complex architecture than that of microprocessors.
b) Only perform a limited number of executions simultaneously.
c) Mostly used in micro-equipments.
d) Cannot interface high power devices directly.

Applications of Microcontroller:

1. Consumer Electronics Products: Toys, Cameras, Robots, Washing Machine, Microwave Ovens etc. [any automatic home appliance]

2. Instrumentation and Process Control: Oscilloscopes, Multi-meter, Leakage Current Tester, Data Acquisition and Control etc.

3. Medical Instruments: ECG Machine, Accu-Check etc.

4. Communication: Cell Phones, Telephone Sets, Answering Machines etc.

5. Office Equipment: Fax, Printers etc.

6. Multimedia Application: Mp3 Player, PDAs etc.

7. Automobile: Speedometer, Auto-breaking system etc.

Microcontroller structure

A microcontroller basically contains one or more following components:

- Central processing unit(CPU)
- Random Access Memory)(RAM)
- Read Only Memory(ROM)
- Input/output ports
- Timers and Counters
- Interrupt Controls
- Analog to digital converters
- Digital analog converters
- Serial interfacing ports
- Oscillatory circuits

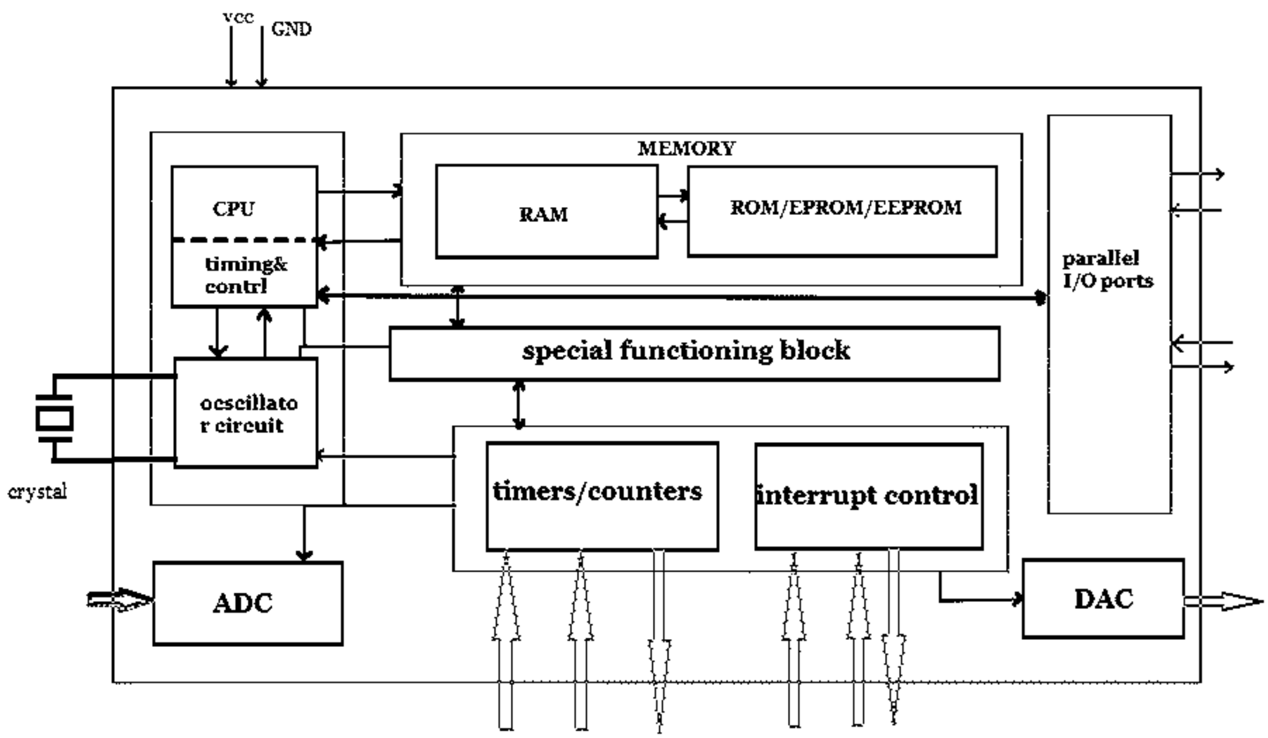


Fig. Microcontroller structure

CPU

CPU is the brain of a microcontroller. CPU is responsible for fetching the instruction, decodes it, then finally executed. CPU connects every part of a microcontroller into a single system. The primary function of CPU is fetching and decoding instructions. The instruction fetched from program memory must be decoded by the CPU.

Memory

The function of memory in a microcontroller is the same as a microprocessor. It is used to store data and program. A microcontroller usually has a certain amount of RAM and ROM (EEPROM, EPROM, etc) or flash memories for storing program source codes.

Parallel input/output ports

Parallel input/output ports are mainly used to drive/interface various devices such as LCD’S, LED’S, printers, memories, etc to a microcontroller.

Serial ports

Serial ports provide various serial interfaces between a microcontroller and other peripherals like parallel ports.

Timers/counters

This is the one of the useful function of a microcontroller. A microcontroller may have more than one timer and counters. The timers and counters provide all timing and counting functions inside the microcontroller. The major operations of this section are performed clock functions, modulations, pulse generations, frequency measuring, making oscillations, etc. This also can be used for counting external pulses.

Analog to Digital Converter (ADC)

ADC converters are used for converting the analog signal to digital form. The input signal in this converter should be in analog form (e.g. sensor output) and the output from this unit is in digital form. The digital output can be used for various digital applications (e.g. measurement devices).

Digital to Analog Converter (DAC)

DAC perform reversal operation of ADC conversion. DAC converts the digital signal into analog format. It usually used for controlling analog devices like DC motors, various drives, etc.

Interrupt control

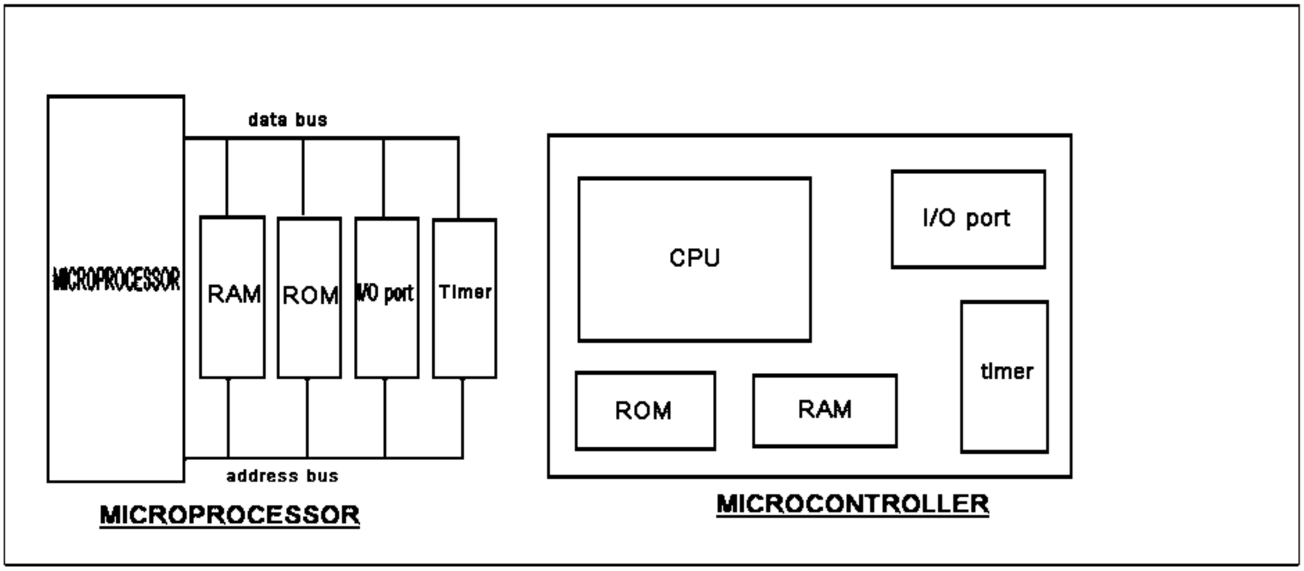
The interrupt control used for providing interrupt (delay) for a working program. The interrupt may be external (activated by using interrupt pin) or internal (by using interrupt instruction during programming).

Special functioning block

Some microcontrollers used only for some special applications (e.g. space systems and robotics) these controllers containing additional ports to perform such special operations. This considered as special functioning block.

Comparison between Microprocessor and Microcontroller

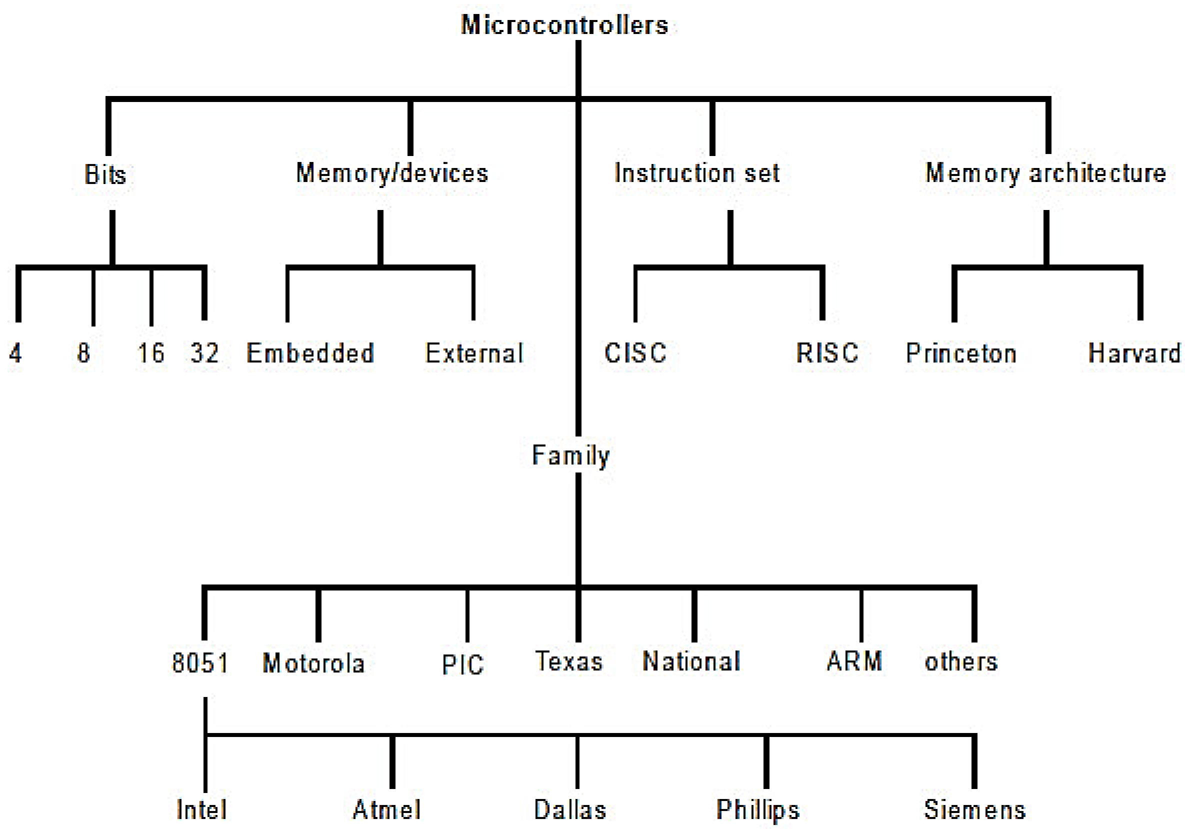
The main comparison between microprocessor and microcontroller shown in fig.



	Microprocessors	Microcontrollers
1	It is only a general purpose computer CPU	It is a microcomputer itself
2	Memory, I/O ports, timers, interrupts are not available inside the chip	All are integrated inside the microcontroller chip
3	This must have many additional digital components to perform its operation	Can function as a microcomputer without any additional components.
4	Systems become bulkier and expensive.	Make the system simple, economic and compact
5	Not capable for handling Boolean functions	Handling Boolean functions
6	Higher accessing time required	Low accessing time
7	Very few pins are programmable	Most of the pins are programmable
8	Very few number of bit handling instructions	Many bit handling instructions
9	Widely Used in modern PC and laptops	widely in small control systems

Types of Microcontroller:

The microcontrollers are characterized regarding bus-width, instruction set, and memory structure. This article is going to describe some of the basic types of the Microcontroller that newer users may not know about. So the microcontroller can be classified according to their characteristics as shown in graph below:



Types of microcontrollers

Bits:
It indicates the number of bit it can process at a time.

- 4-bit microcontroller: Atmel MARC4, Toshiba T3400

- The bits in microcontroller are 8-bits, 16-bits and 32-bits microcontroller. In **8-bit** microcontroller, the point when the internal bus is 8-bit then the ALU is performs the arithmetic and logic operations. The examples of 8-bit microcontrollers are Intel 8031/8051, PIC1x and Motorola MC68HC11 families.
- The **16-bit** microcontroller performs greater precision and performance as compared to 8-bit. For example 8 bit microcontrollers can only use 8 bits, resulting in a final range of 0×00 – 0xFF (0-255) for every cycle. In contrast, 16 bit microcontrollers with its 16 bit data width has a range of 0×0000 – 0xFFFF (0-65535) for every cycle. A longer timer most extreme worth can likely prove to be useful in certain applications and circuits. It can automatically operate on two 16 bit numbers. Some examples of 16-bit microcontroller are 16-bit MCUs are extended 8051XA, PIC2x, Intel 8096 and Motorola MC68HC12 families.
- The **32-bit** microcontroller uses the 32-bit instructions to perform the arithmetic and logic operations. These are used in automatically controlled devices including implantable medical devices,

engine control systems, office machines, appliances and other types of embedded systems. Some examples are Intel/Atmel 251 family, PIC3x.

Memory/Devices:

Embedded microcontroller: When an embedded system has an MCU that has all the hardware and software units in a single unit, the MCU is called embedded microcontroller. Very few or no other external unit or system is present for processing during the control or use of the external devices. For example, a telephone handset circuit uses an embedded microcontroller.

External memory microcontroller: When an embedded system has an MCU that has all the hardware and software units present not as a single unit and has all or part of the memory unit externally interfaced using an interfacing circuit which is called the glue circuit, the MCU is called an external memory microcontroller. For example, 8031 has the program memory which is interfaced externally to it. The 8051 has both internal as well as external program memory.

Family:

The microcontrollers can also be classified according to their family. family is usually a company or a manufacturer in which controller is fabricated. Each family have its own architecture and instruction set. some though 8051 is famous but PIC and ARM are going more popular than 8051.

Instruction Set:

RISC is the way to make hardware simpler whereas CISC is the single instruction that handles multiple work. In this article, we are going to discuss RISC and CISC in detail as well as the Difference between RISC and CISC, Let’s proceed with RISC first.

Reduced Instruction Set Architecture (RISC)

The main idea behind this is to simplify hardware by using an instruction set composed of a few basic steps for loading, evaluating, and storing operations just like a load command will load data, a store command will store the data.

Characteristics of RISC

- Simpler instruction, hence simple instruction decoding.
- Instruction comes undersize of one word.
- Instruction takes a single clock cycle to get executed.
- More general-purpose registers.
- Simple Addressing Modes.
- Fewer Data types.
- A pipeline can be achieved.

Advantages of RISC

- **Simpler instructions:** RISC processors use a smaller set of simple instructions, which makes them easier to decode and execute quickly. This results in faster processing times.
- **Faster execution:** Because RISC processors have a simpler instruction set, they can execute instructions faster than CISC processors.
- **Lower power consumption:** RISC processors consume less power than CISC processors, making them ideal for portable devices.

Disadvantages of RISC

- **More instructions required:** RISC processors require more instructions to perform complex tasks than CISC processors.

- **Increased memory usage:** RISC processors require more memory to store the additional instructions needed to perform complex tasks.
- **Higher cost:** Developing and manufacturing RISC processors can be more expensive than CISC processors.

Complex Instruction Set Architecture (CISC)

The main idea is that a single instruction will do all loading, evaluating, and storing operations just like a multiplication command will do stuff like loading data, evaluating, and storing it, hence it's complex.

Characteristics of CISC

- Complex instruction, hence complex instruction decoding.
- Instructions are larger than one-word size.
- Instruction may take more than a single clock cycle to get executed.
- Less number of general-purpose registers as operations get performed in memory itself.
- Complex Addressing Modes.
- More Data types.

Advantages of CISC

- **Reduced code size:** CISC processors use complex instructions that can perform multiple operations, reducing the amount of code needed to perform a task.
- **More memory efficient:** Because CISC instructions are more complex, they require fewer instructions to perform complex tasks, which can result in more memory-efficient code.
- **Widely used:** CISC processors have been in use for a longer time than RISC processors, so they have a larger user base and more available software.

Disadvantages of CISC

- **Slower execution:** CISC processors take longer to execute instructions because they have more complex instructions and need more time to decode them.
- **More complex design:** CISC processors have more complex instruction sets, which makes them more difficult to design and manufacture.
- **Higher power consumption:** CISC processors consume more power than RISC processors because of their more complex instruction sets.

Difference between the RISC and CISC Processors

RISC	CISC
It is a Reduced Instruction Set Computer.	It is a Complex Instruction Set Computer.
It emphasizes on software to optimize the instruction set.	It emphasizes on hardware to optimize the instruction set.
It is a hard wired unit of programming in the RISC Processor.	Microprogramming unit in CISC Processor.
It requires multiple register sets to store the instruction.	It requires a single register set to store the instruction.
RISC has simple decoding of instruction.	CISC has complex decoding of instruction.
Uses of the pipeline are simple in RISC.	Uses of the pipeline are difficult in CISC.
It uses a limited number of instruction that requires less time to execute the instructions.	It uses a large number of instruction that requires more time to execute the instructions.
It uses LOAD and STORE that are independent instructions in the register-to-register a program's interaction.	It uses LOAD and STORE instruction in the memory-to-memory interaction of a program.
RISC has more transistors on memory registers.	CISC has transistors to store complex instructions.
The execution time of RISC is very short.	The execution time of CISC is longer.
RISC architecture can be used with high-end applications like telecommunication, image processing, video processing, etc.	CISC architecture can be used with low-end applications like home automation, security system, etc.
It has fixed format instruction.	It has variable format instruction.
The program written for RISC architecture needs to take more space in memory.	Program written for CISC architecture tends to take less space in memory.

Example of RISC: ARM, PA-RISC, Power Architecture, Alpha, AVR, ARC and the SPARC.	Examples of CISC: VAX, Motorola 68000 family, System/360, AMD and the Intel x86 CPUs.

Memory Architecture:

- **Harvard Memory Architecture Microcontroller**: The point when a microcontroller unit has a dissimilar memory address space for the program and data memory, the microcontroller has Harvard memory architecture in the processor.
- **Princeton Memory Architecture Microcontroller**: The point when a microcontroller has a common memory address for the program memory and data memory, the microcontroller has Princeton memory architecture in the processor.

Memory Addressing

Microprocessor Architecture, Programming, and Applications with the 8085> Pages 69-74, Pages 120-126

I/O Interfacing

Microprocessor Architecture, Programming, and Applications with the 8085> Pages 140-149

Memory Mapped I/O Vs I/O Mapped I/O

Isolated I/O	Memory-mapped I/O
Different address spaces are used for computer memory and I/O devices. I/O devices have dedicated address space.	Same address space is used for memory and I/O devices.
Separate control unit and control instructions are used in case of I/O devices.	Control units and instructions are same for memory and I/O devices.
More complex and costlier than memory -mapped I/O as more bus are used.	Easier to build and cheap as it's less complex.
Entire address space can be used by memory as I/O devices have separate address space.	Some part of the address space of computer memory is consumed by I/O devices as address space is shared.
Computer memory and I/O devices use different control instructions for read write.	Computer memory and I/O devices can both use same set of read and write instructions.
Separate control bus is used for computer memory and I/O devices. Though same address and data bus are used.	Address, data and control bus are same for memory and I/O devices.