Introduction to Microprocessors

Department of Computer Science and Engineering

BRAC University

Course ID : CSE 341

Course Title : MICROPROCESSORS

Instructors

- **Aminul Huq**
  Lecturer, Department of Computer Science & Engineering
- **Saadat Hasan Khan**
  Lecturer and Lab Coordinator, Department of Computer Science & Engineering
- **Ramkrishna Saha**
  Lecturer, Department of Computer Science & Engineering
- **Sifat Tanvir**
  Lecturer, Department of Computer Science & Engineering
- **Ragib Morshed**
  Lecturer, Department of Computer Science & Engineering
- **Syed Zamil Hasan Shoumo**
  Lecturer and Theory Coordinator, Department of Computer Science & Engineering

Topics to be Covered

- Microprocessors and Microcontrollers
- Applications of microprocessors and microcontrollers
- Intel 8086 Microprocessor: Internal architecture, Register structure, Addressing modes, Instruction set etc.
- An overview of Intel 80186, 80286, 80386, 80486 and Pentium microprocessors
- RISC and CISC processors.
- Coprocessors.
- Assembly language programming

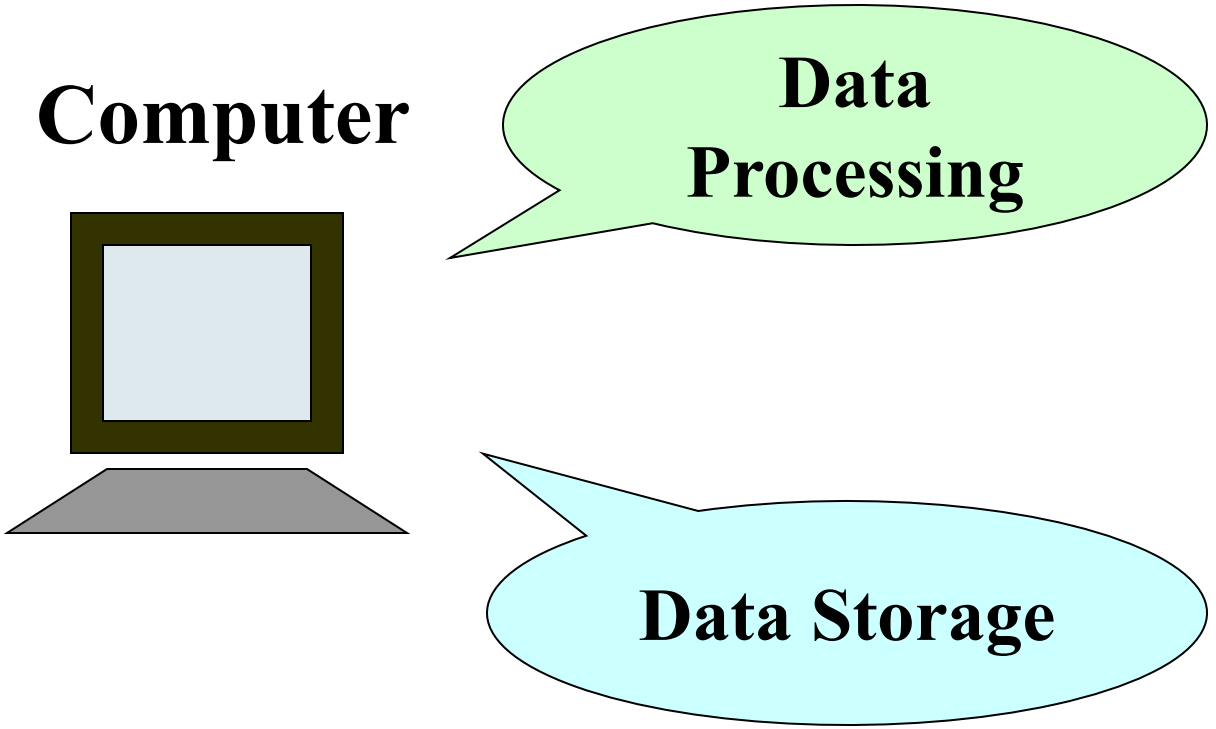
Recommended Texts

- *Microprocessors and Interfacing: Programming and Hardware, by* **Douglas V. Hall**
- *Assembly Language Programming and Organization of the IBM PC, by* **Ytha Y. Yu, Charles Marut**
- *Microprocessor, architecture, programming & application with the 8085, by* **Ramesh Gaonkar**
- *The Intel Microprocessor, by* **Barry B. Bray**
- *Microprocessor and Microcomputer – Based System Design, by* **Mohamed Rafiquzzaman**

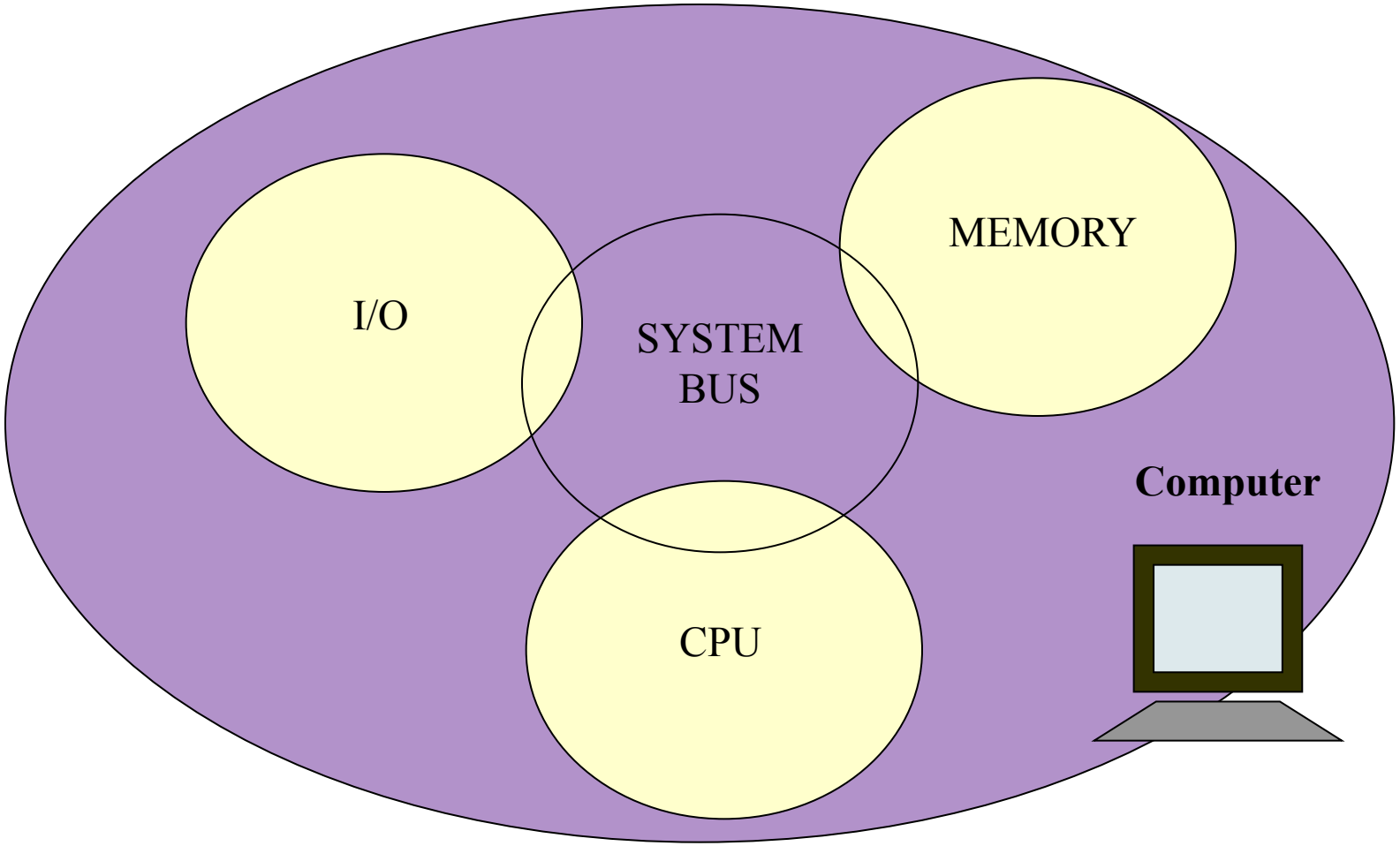
Some tips before we begin

- Number Systems and their Conversion
- Basics of “Digital Logic Design”
- Basics of “Computer Architecture”
- Basic Programming

Concept of Computer



Major Components of Computer

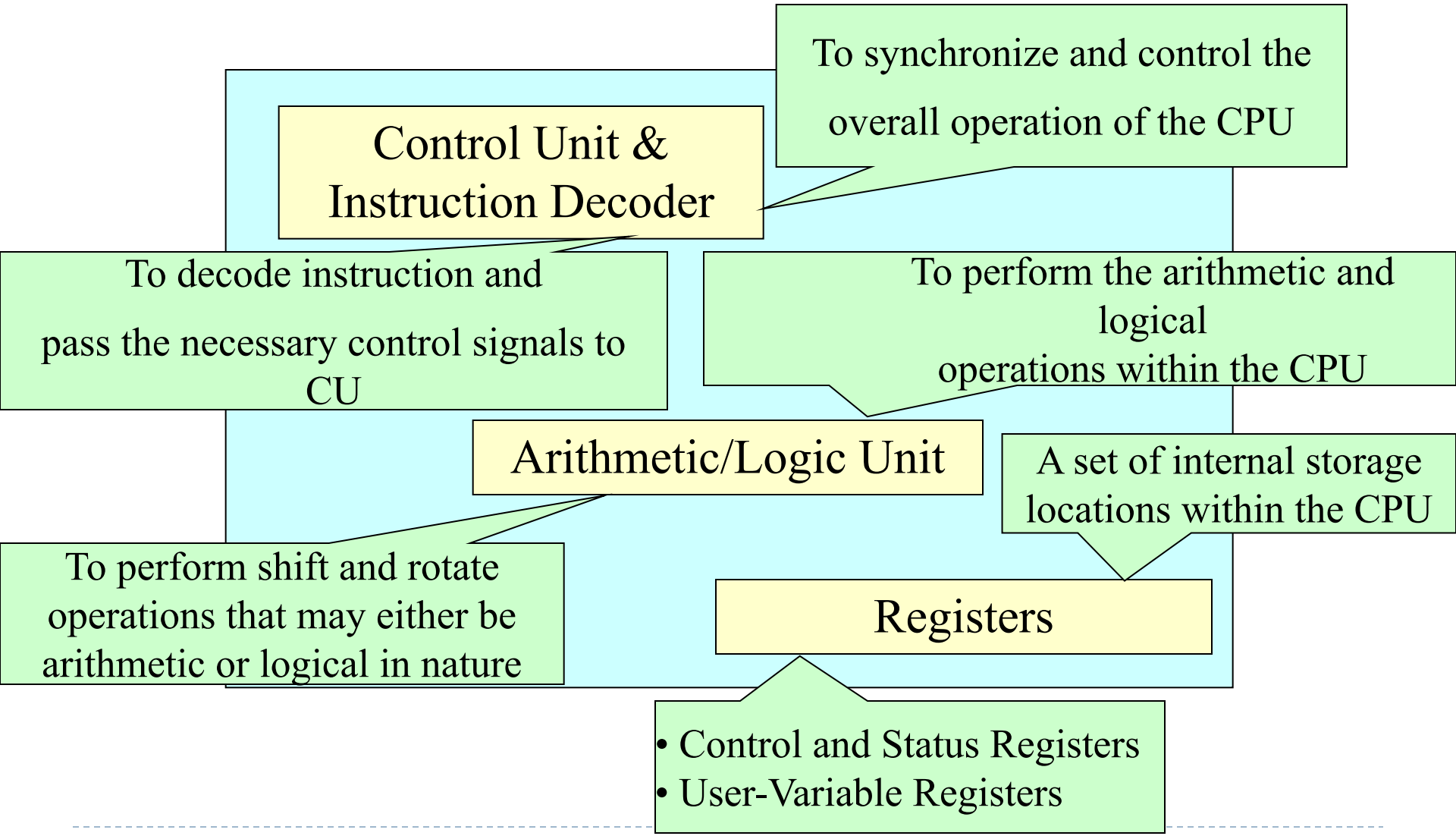


Central Processing Unit

- A **central processing unit** (**CPU**) is a description of a class of logic machines that can execute **computer programs**.

- The form, design and implementation of CPUs have changed dramatically since the earliest examples, but their fundamental operation has remained much the same.

Central Processing Unit



So .. What is Microprocessor?

A microprocessor (abbreviated as µP) is a Silicon Chip that contains an electronic central processing unit (CPU). In the world the word µP or CPU is now used interchangeably. It is made from miniaturized transistors and other circuit elements on a single semiconductor integrated circuit (IC).

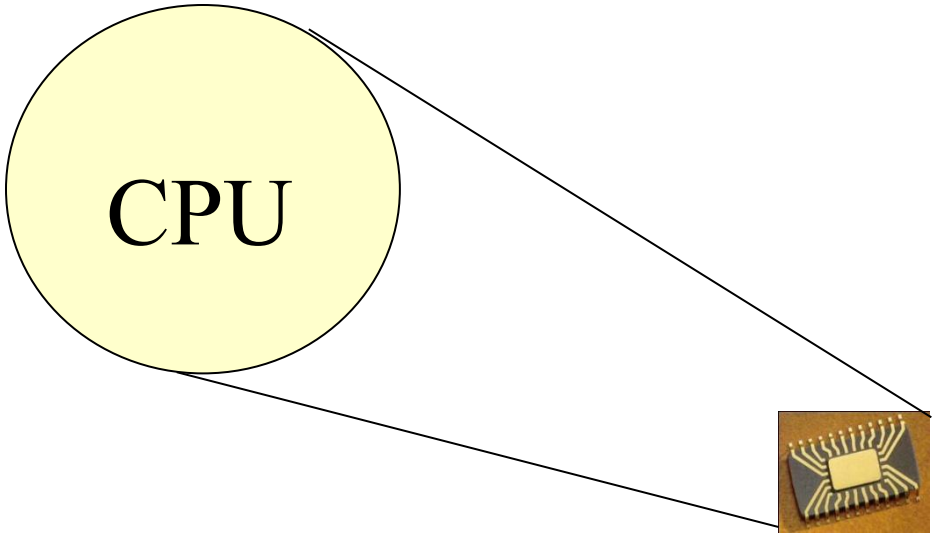
The integration of the whole CPU onto a single **VLSI Chip** therefore greatly reduced the cost of processing capacity.

Architectures of Microprocessors:

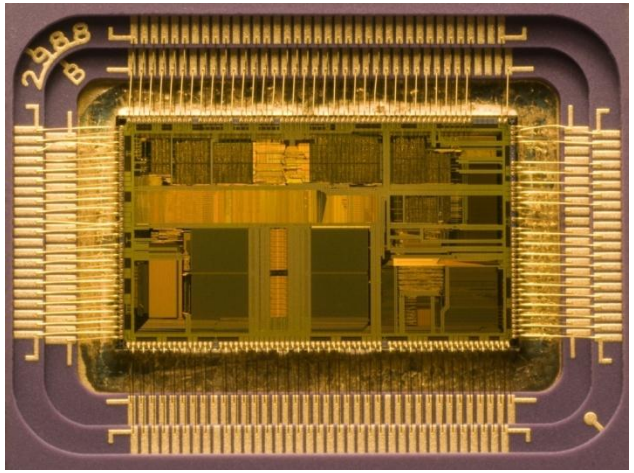
- RISC (Reduced Instruction Set Computer)
- CISC (Complex Instruction Set Computer)
- Special-purpose designs: Microcontrollers, Digital Signal Processors (DSP) and Graphics Processing Units (GPU).

Concept about Microprocessor

- A **microprocessor** incorporates most or all of the functions of a central processing unit (CPU) on a single **integrated circuit** (IC).



Die of an Intel **80486DX2** microprocessor (actual size: 12×6.75 mm) in its packaging



List of Microprocessors

1971 - Intel 4004, 1st single chip CPU, 4-bit processor, 46 instructions
1972 - Intel 4040, enhanced 4004, 60 instructions

1972 - Intel 8008, 8-bit µP

1972 - Texas Instrument TMS 1000, 1st single µC, 4-bit

1974 - Intel 8080, successor to the 8008, used in Altair 8800

1975 - Motorola 6800, used MOS technology

1976 - Intel 8085, updated 8080, +5V power supply

1976 - Zilog Z80, improved 8080

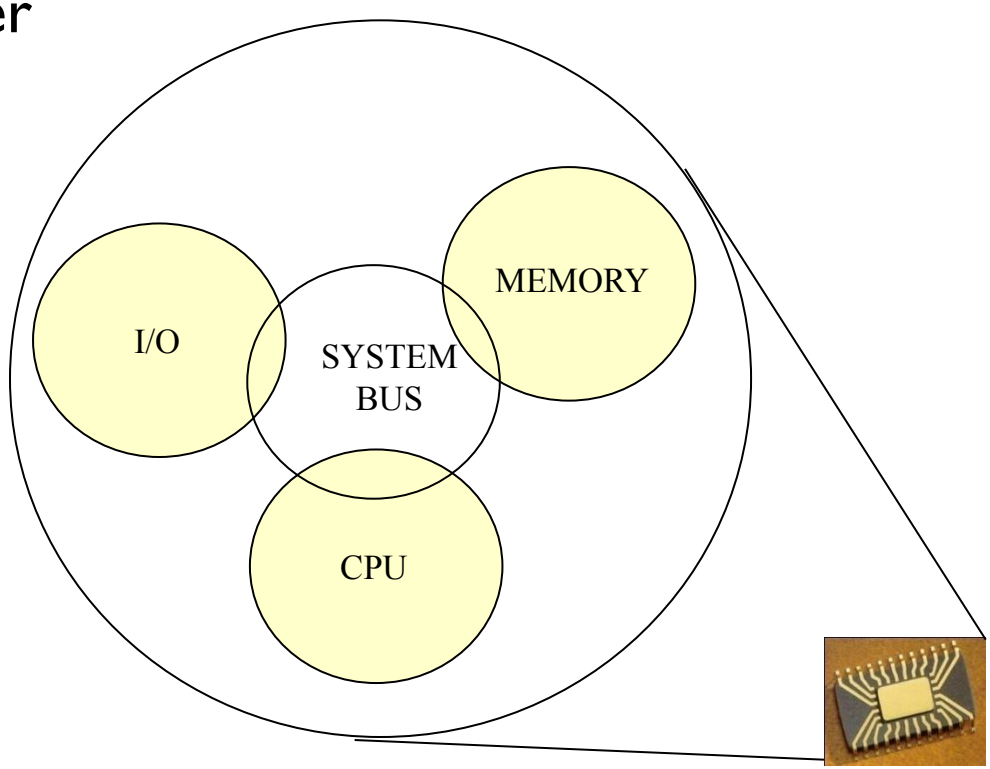
1976 - TI TMS 9900, 1st 16-bit µP

1978 - Zilog Z8000, Motorola 68000, 16-bit µP

1978 - Intel 8086, 16-bit, IBM's choice...

Microcontroller (µC)

- **Microcontroller** is an IC dedicated to perform simpler tasks.
  - A microcontroller is the integration of
    - Processor
    - Memory (RAM, ROM)
    - I/O ports



List of Microcontrollers

1972 - Texas Instrument TMS 1000, 1st single µC, 4-bit

1976 - Intel 8048, 8-bit µC, 1k ROM, 64b RAM, 27 I/O

1980 - Intel 8051, 4k ROM, 128b RAM, 32 I/O, 2 16-bits timers

1980s -

(MCS-51 family)

- Intel 8031, 8052, 8751, …
- Atmel AT89C51, AT 89C1052/2051,…
- Dallas Semiconductor DS5000 series…

-Philips, National Semiconductor, …

- Freescale S32K MCU, Renesas RL 78G1F

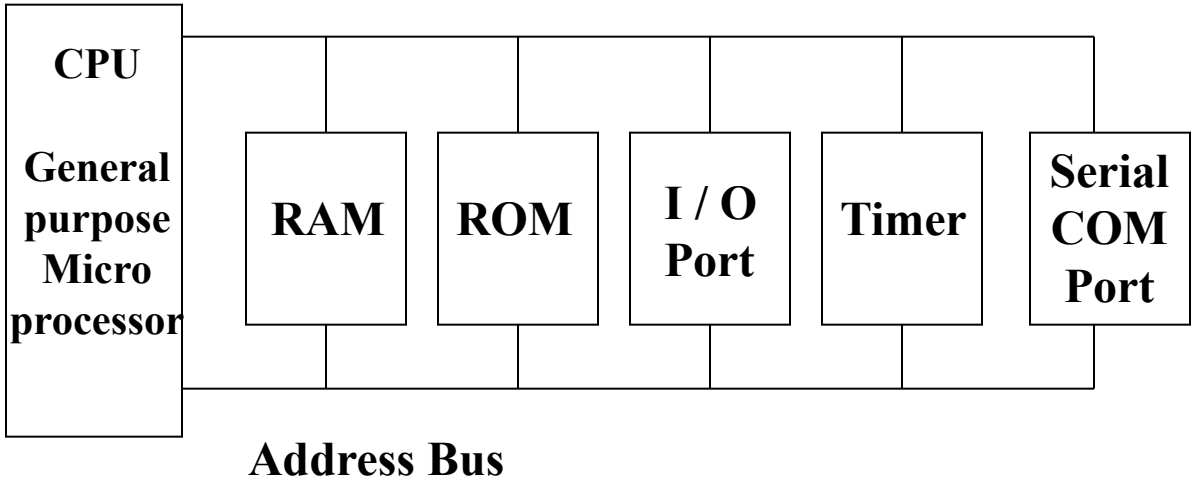
Microprocessor System Vs Microcontroller System

Microprocessor	Microcontroller
Used where intensive processing is required	Used where task is fixed and predefined
Only CPU is in the chip. Memory, I/O port are connected externally	CPU, Memory, I/O port – all are connected on the same single chip
Higher Clock speed and external RAM used is also higher	Lower Clock speed and RAM used is also lower
The program for the microprocessor can be changed for different applications.	The program for the microcontroller is fixed once it is designed
Cost is comparatively higher	Cost is comparatively lower
Power consumption is higher	Power consumption is lower
Overall size of the system is large	Overall size of the system is smaller
Applications include personal computers	Applications include washing machines, camera etc.

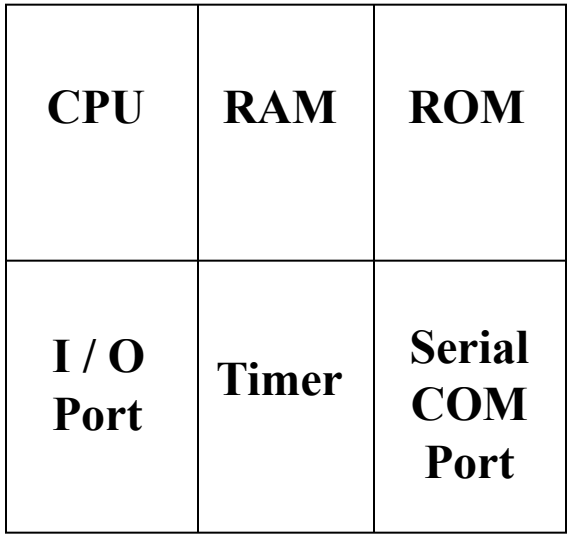
Food for thought

- We know that your computer uses a microprocessor. But what about your keyboard?

Microprocessor System Vs Microcontroller System



Microprocessor System



Microcontroller

Assembly Language

- **Assembly language:**
  - Assembly language is used in programming because it is difficult to program a microprocessor in its native machine language.

- **Assembler:**
  - An assembler is a program that converts assembly language into machine language.

  - Assemblers are similar to compilers in that they produce executable code. However, assemblers are more simplistic.

High level language vs Machine language

- int a, b, c;
  a = 83;
  b = -2; **// high level language**
  c = a + b;

- 0010 0001 0000 0100
- 0001 0001 0000 0101
- 0011 0001 0000 0110 **//machine language**
- 0111 0000 0000 0001
- 0000 0000 0101 0011
- 1111 1111 1111 1110

Example of Assembly Language

- **Add 2 with 3**

mov cl, 3 : copy the value 3 in the internal register cl *// so currently cl is holding the value 3*

add cl, 2 : add the value 2 with the current value of cl and store sum in cl *// after adding 2, cl is now holding the value 5*

- **Subtract 2 from 3**

mov cl, 3 : copy the value 3 in the internal register cl *//so currently cl is holding the value 3*

sub cl, 2 : sub the value 2 from the current value of cl *//after subtracting 2, cl is now holding the value 1*

mov, add, sub **--- *opcodes or instructions***

cl, 3, 2 **---- operands**

Food for thought

- **Using cl register show assembly code for the following expression :**

5 + 6 – 10

- mov cl, 5
- add cl, 6
- sub cl, 10