Introduction to Microprocessors

Course Teacher:

Md. Fahmidur Rahman Sakib

Lecturer, Department of Computer Science & Engineering

Metropolitan University

Course ID: CSE 237

Course Title: Microprocessor and Interfacing

Topics to be Covered in this Course !!

- Concept of microprocessor
- Evolution and Internal architecture of microprocessors
- Intel 8086 Microprocessor: internal architecture, register structure, programming model, addressing modes, instruction set
- Coprocessors, Multiprocessor system
- An overview of Intel 80186, 80286, 80386 & Pentiums
- Stepper Motor, Transducers, printers, motors and peripherals.
- Assembly language programming

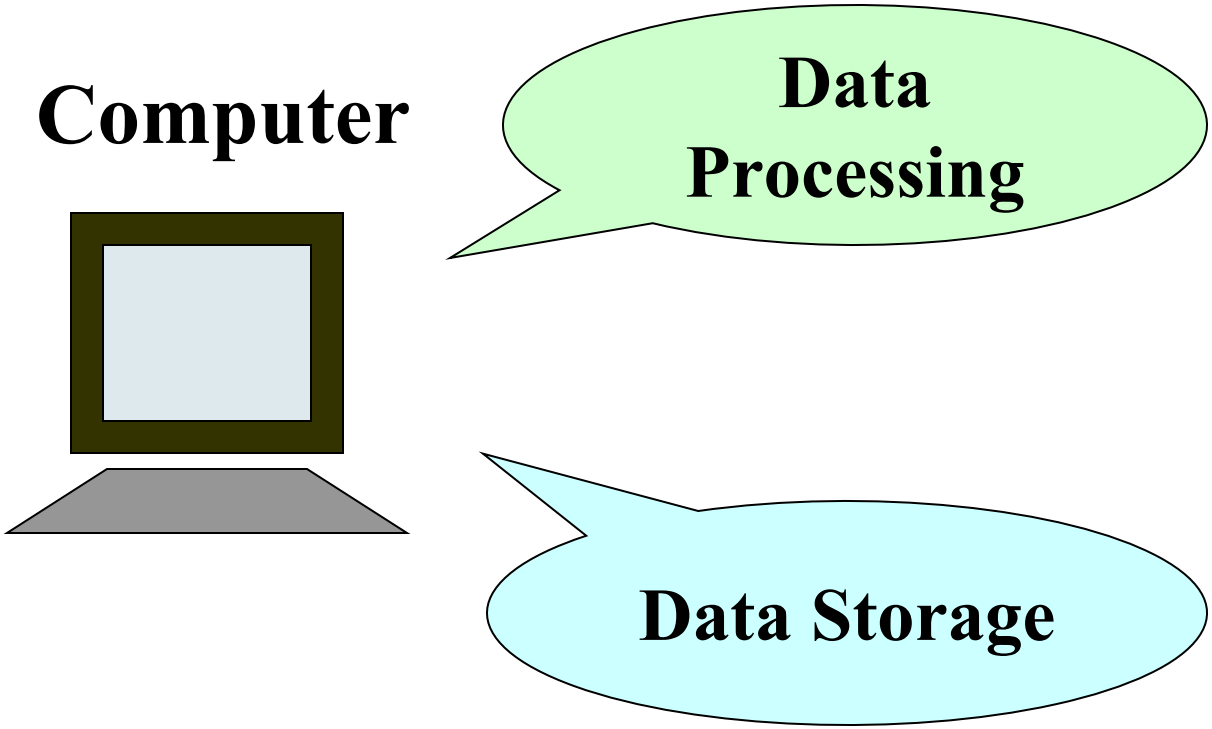
Recommended Texts

- Microprocessors *and Interfacing: Programming and Hardware,* **Author:** Douglas V. Hall
- Microprocessors and Interfacing , **Author:** D. A. Godse, A.P. Godse
- Computer Peripheral & Interfacing, **Author:** Gourav Gupta, Eagle Prakashan, Jalandhar

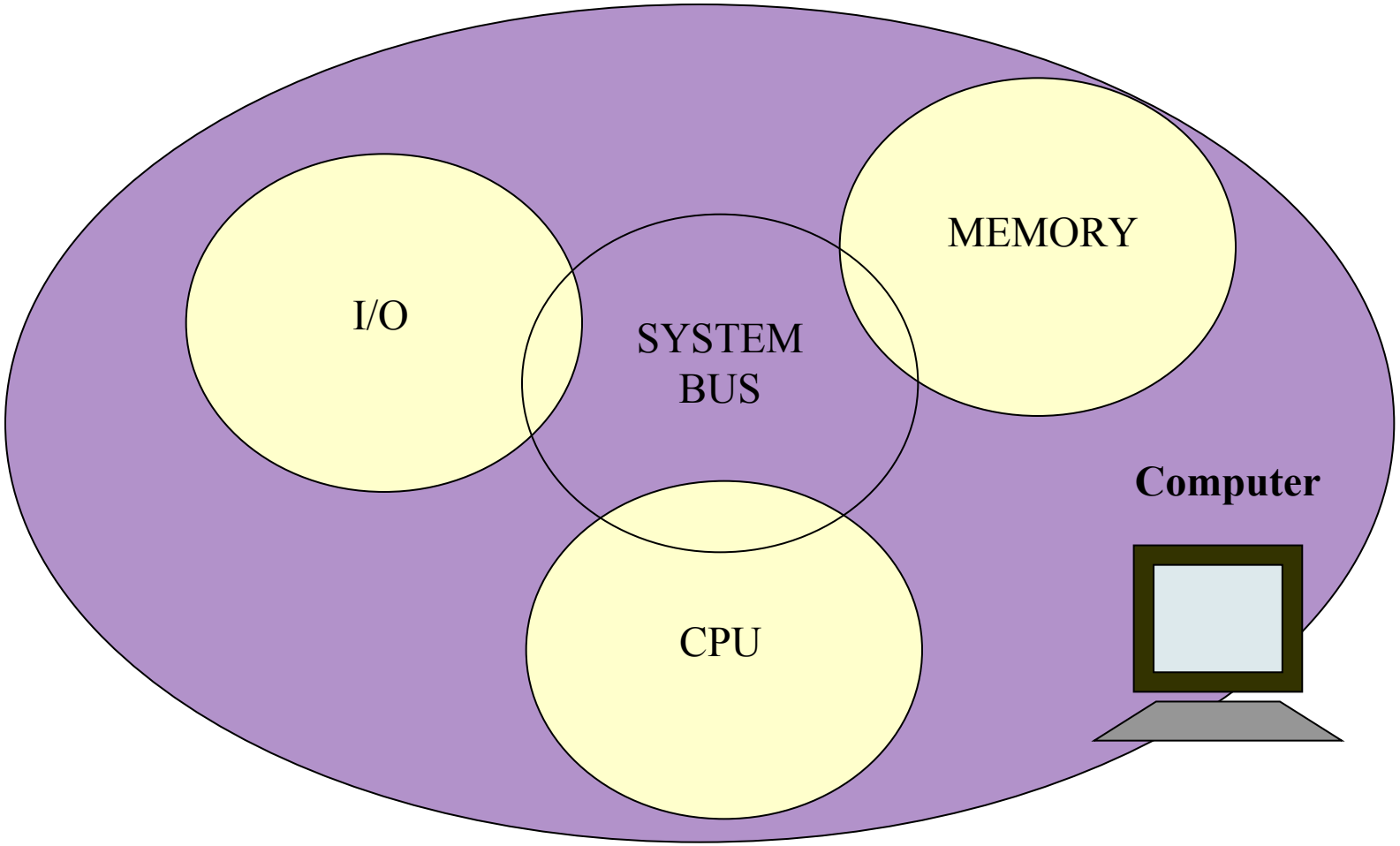
Does Earlier Knowledge Require ??

- You should have the knowledge about –
  - *Number Systems.*
  - Basics of "*Digital Logic Design*" course.
  - Basics of "*Computer Organization and Architecture*" course.
  - "*Basic Programming*".

Concept of Computer

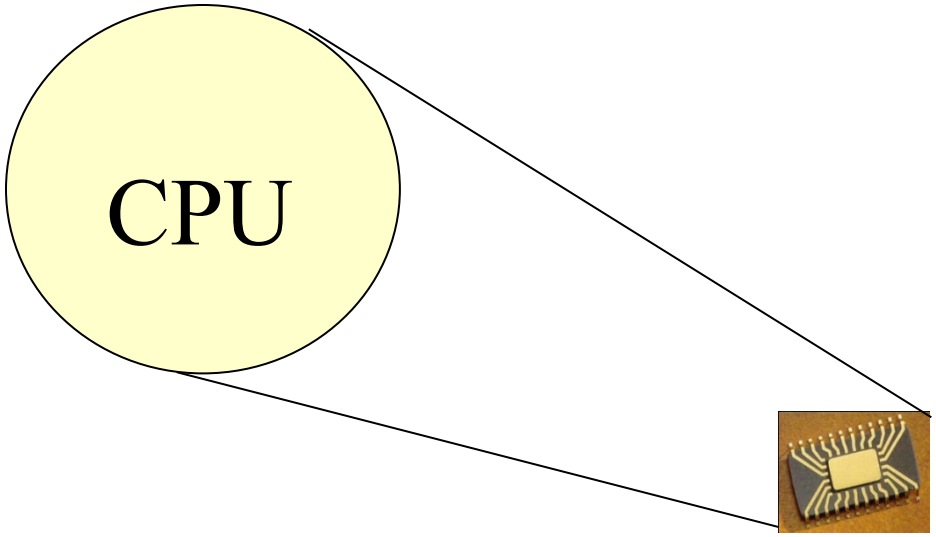


Major Components of Computer

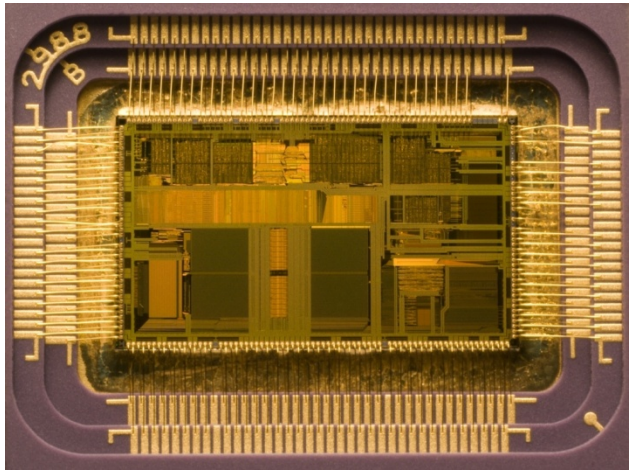


Concept about Microprocessor

- 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

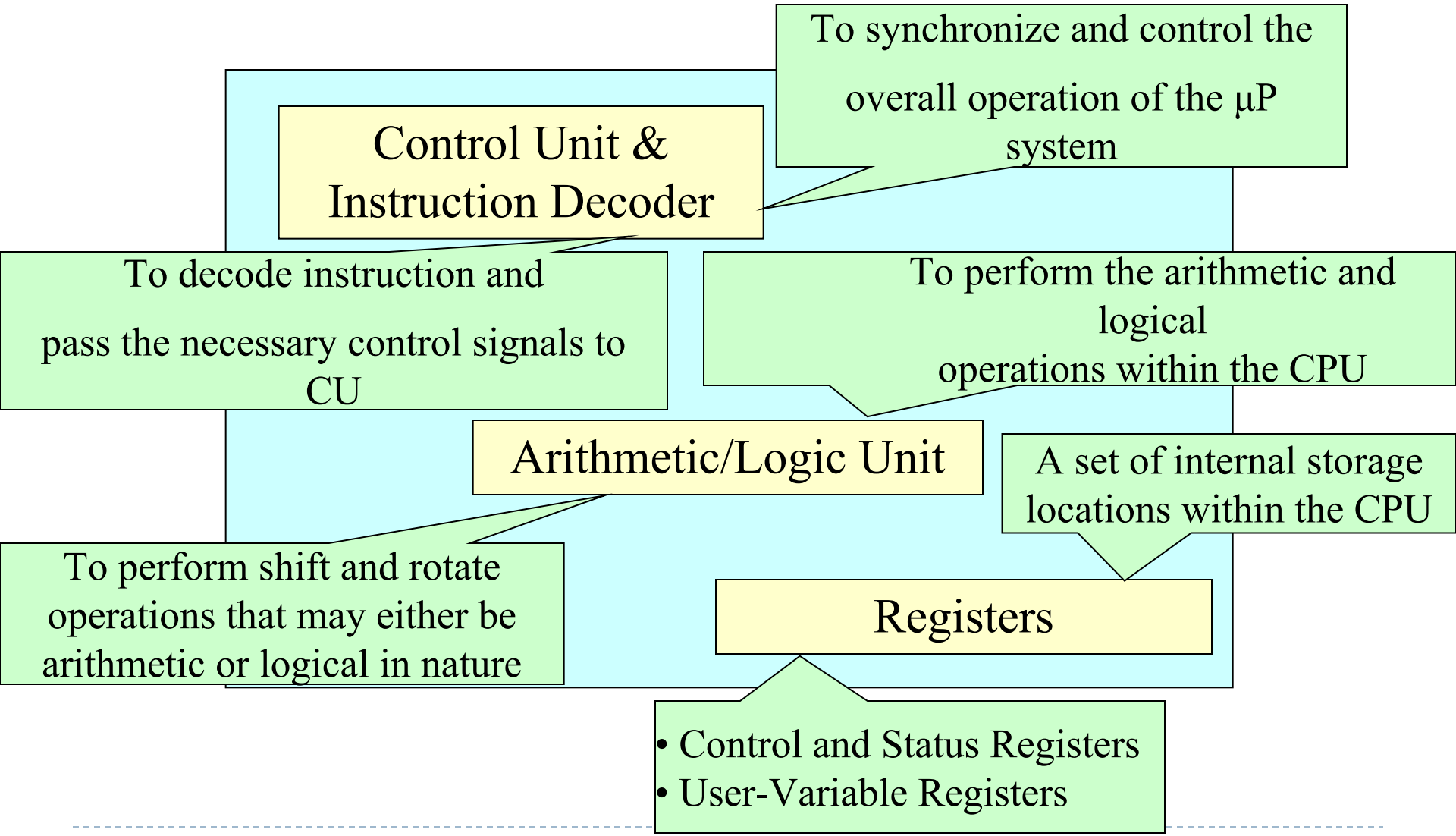


Central Processing Unit

- A **central processing unit** (**CPU**), or sometimes just called **Microprocessor (µP)**, is a description of a class of logic machines that can execute **computer programs**.

- This broad definition can easily be applied to many early computers that existed long before the term "CPU" ever came into widespread usage. However, the term itself and its initialism have been in use in the computer industry at least since the early 1960s.
- 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/Microprocessor (μP)



So .. What is Microprocessor?

- A microprocessor (abbreviated as µP or uP) is a Silicon Chip that contains an electronic central processing unit (CPU). In the world uP or CPU used interchangeably, which 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:**
  - 8-bit designs
  - 16-bit designs
  - 32-bit designs
  - 64-bit designs
  - Multi-core designs
  - RISC (Reduced Instruction Set Computer)
  - CISC (Complex Instruction Set Computer)
  - Special-purpose designs: Microcontrollers, Digital Signal Processors (DSP) and Graphics Processing Units (GPU).

List of Microprocessors

1971 - Intel 4004, 1st single chip CPU, 4-bit processor, 45 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

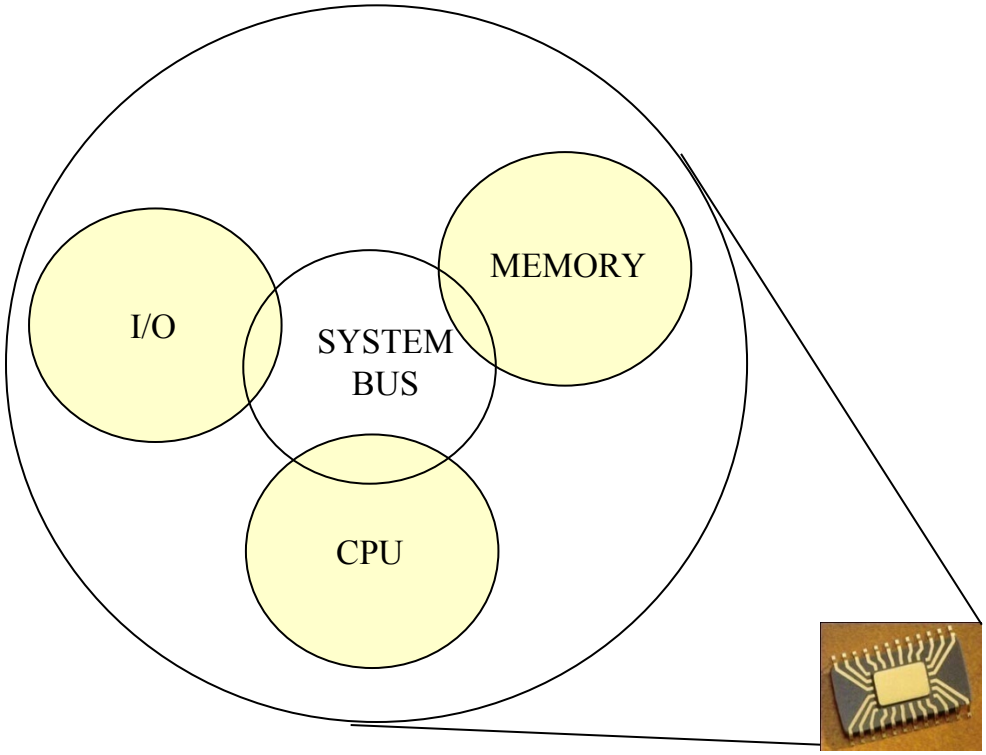
1976 - TI TMS 9900, 1st 16-bit µP

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

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

Similar but Different !! **Microcontroller (µC)**

- **Microcontroller** is an IC dedicated to perform one task.
  - Integrates the memory and other features of a microprocessor.
  - A microcontroller is the integration of
    - Microprocessor
    - Memory
      - ROM types – commonly flash PROM
      - RAM – Static ram



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, …

- (Other µCs) Microchip PIC16 series, Motorola 68HC11, Zilog's Z86

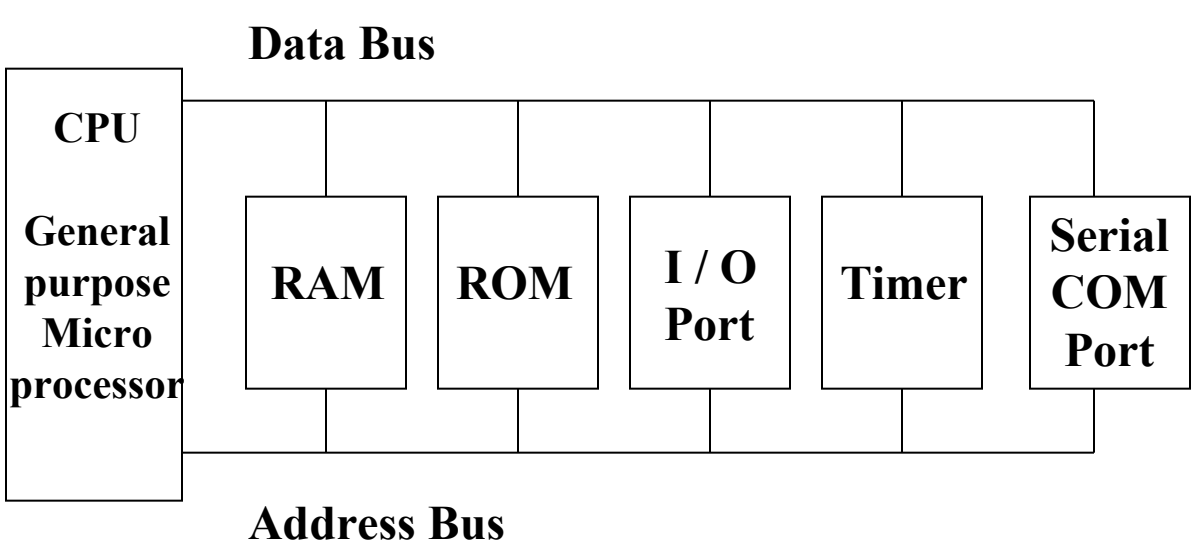
Microprocessor System Vs Microcontroller System

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.

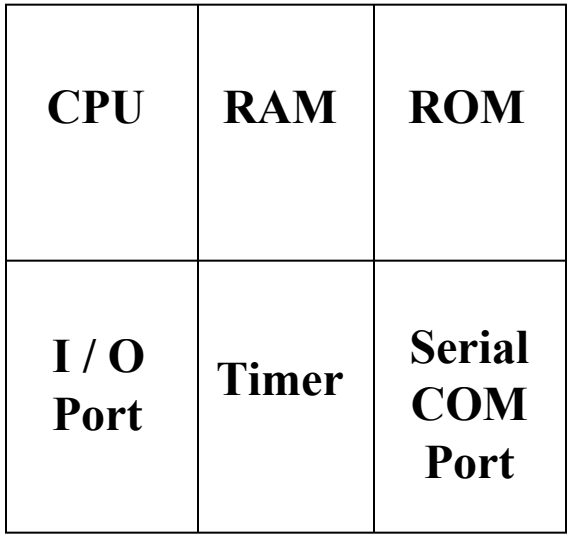
Food for thought

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

Microprocessor System Vs Microcontroller System



General-Purpose Microprocessor System



Microcontroller

Assembly Language

- **Assembly language:**
  - Assembly language is used for most 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 8085 Assembly Language

Address **Instruction**

```
202A    MVI   A, 21H     ; Copies 21 into accumulator
202C    MVI   B, 2AH     ; Copies 2A into B register
202E    ADD   B          ; Adds B reg. content with Acc
        and stores the result in Acc.
202F    STA   [41 FF]    ; Stores the Acc (the sum) into
        the memory location 41 FF.
2032    HLT              ; Stops the program
```

Memory storage of the Assembly language

Address	Instruction/Data
202A	MVI A,
202B	21
202C	MVI B,
202D	2A
202E	ADD B
202F	STA
2030	FF
2031	41
2032	HLT

Another Example of 8085 Assembly Language

Address **Instruction**

```
2020        MVI    B, 24        ; Copies 24 into B register
2022        INR    B        ; Increment B reg content by 1
2023        MOV    A, B         ; Copies B register into Acc.
2024        SUB    B        ; Subtracts B reg content from
        Acc and stores the result in                    Acc.
2025        STA    [5F FF]      ; Stores the Acc content into
                the memory location 5F FF.
2028        HLT                 ; Stops the program
```

Example of 8086 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

Assembly Language vs. Machine Language

- Machine Language vs Assembly Language
  - Machine language or object code is the only code a computer can execute but it is nearly impossible for a human to work with
  - E4 27 88 C3 E4 27 00 D8 E6 30 F4 the object code for adding two numbers input from the keyboard
- When programming a microprocessor, programmers often use assembly language
  - This involves 3-5 letter abbreviations for the instruction codes (mnemonics) rather than the binary or hex object codes

Address	Hex Object Code				Mnemonics		Comment
					Op-Code	Operand	
0100	E4	27			IN	AL,27H	Input first number from port 27H and store in AL
0102	88	C3			MOV	BL,AL	Save a copy of register AL in register BL
0104	E4	27			IN	AL,27H	Input second number to AL
0106	00	D8			ADD	AL,BL	Add contents of BL to AL and store the sum in AL
0107	E6	30			OUT	30H,AL	Output AL to port 30H
0109	F4				HLT		Halt the computer

Self Learning

- Assembly Language VS Machine Language
- Assembler VS Compiler

Thank You!!