

Microcontroller basics

Department of Instrumentation Engineering

| Lecture 1 |

Microprocessor Basics

In this Lecture

- ◆ Microprocessors and microcontrollers, and their applications
- ◆ Microprocessor...
 - ⊕ Basics
 - ⊕ System components
 - ⊕ Internal architecture
 - ⊕ Instruction fetch
 - ⊕ Instruction execution

Introduction

Microprocessor

- ⊕ A microprocessor *processes* data

Microcontroller

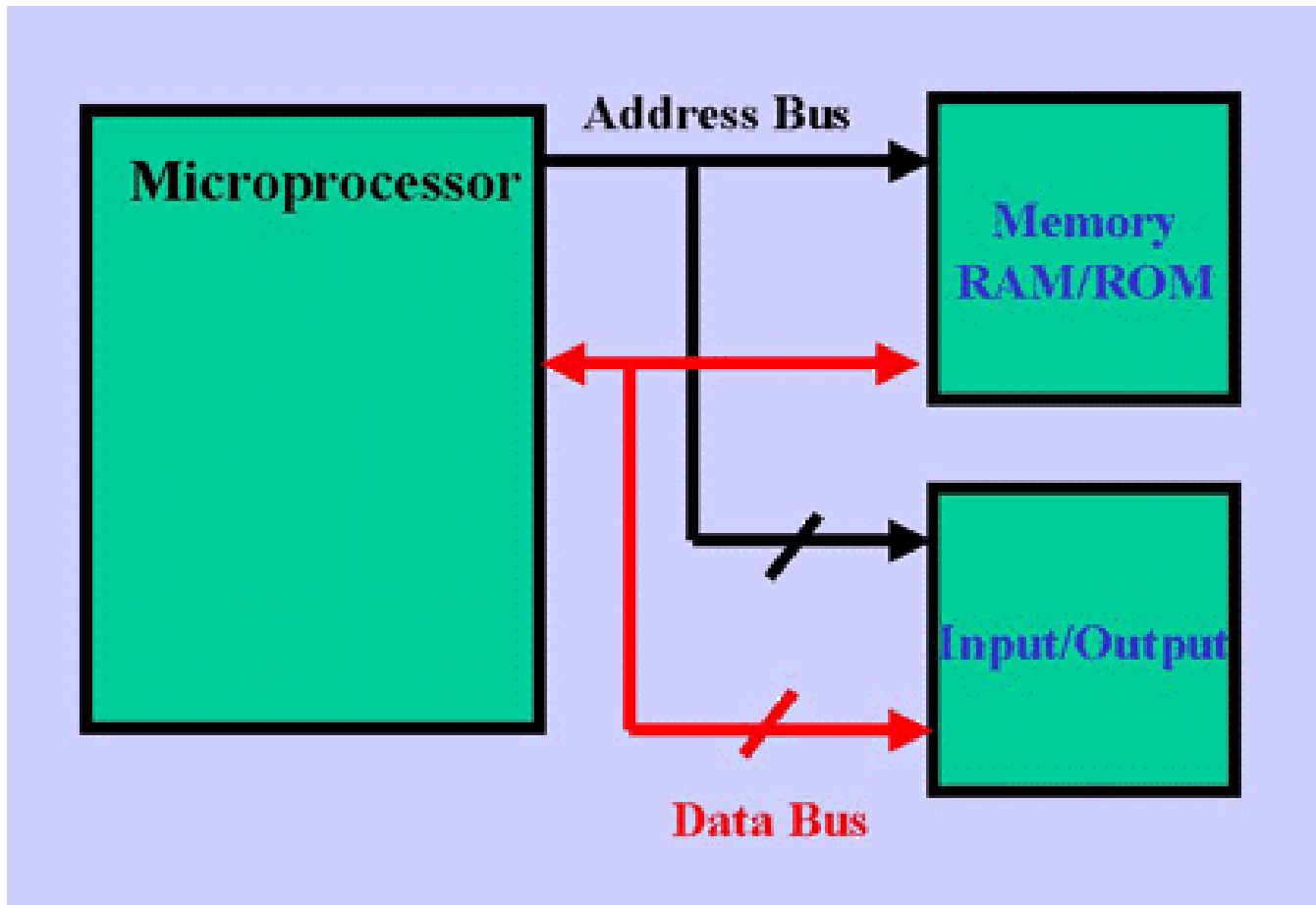
- ⊕ A microcontroller *controls* the operation of a machine (e.g. with a program)



What is a Microprocessor

- ◆ A general-purpose digital computer ***central-processing-unit*** (CPU)
- ◆ ***Not a complete computer***; need to add memory and other items (peripherals)
- ◆ ROM/RAM/keyboard...
- ◆ Read, ***process***, and store/display data

Basic Microprocessor System



Microprocessor System Components

- ◆ Memory
- ◆ Input/output (I/O) device
- ◆ The microprocessor chip
- ◆ System buses

Microprocessor System Components

◆ *Memory*

- ⊕ Read-only-memory (ROM): store *program*
- ⊕ Random-access-memory (RAM): store *data*

◆ *I/O device*

- ⊕ Keyboard and mouse (input)
- ⊕ Monitor (output)

Microprocessor System Components

◆ *Microprocessor chip*

- ⊕ The heart of the microprocessor
- ⊕ Controls all activities
- ⊕ Perform mathematical & logical (i.e. Boolean) operations
- ⊕ Determines the next action to be taken based on the result of preceding operation

Microprocessor System Components

- ◆ ***System buses***
 - ⊕ Bus is a collection of wires
 - ⊕ Transfer data between components inside the computer or between computers
 - ⊕ ***Address bus*** (carry address)
 - ⊕ ***Data bus*** (carry data)
 - ⊕ ***Control bus*** (control & coordination)

Microprocessor System Components

- ◆ **Address bus** (carry address signals)
 - A set of wires along which is sent out the binary code (the address signals) that identifies the memory location to be used
 - Addresses can only be issued by the microprocessor, so it is uni-directional
- ◆ **Data bus** (carry data signals)
 - A set of wires along which data is transferred between units
 - Signals are bi-directional, since data can flow into or out of the microprocessor

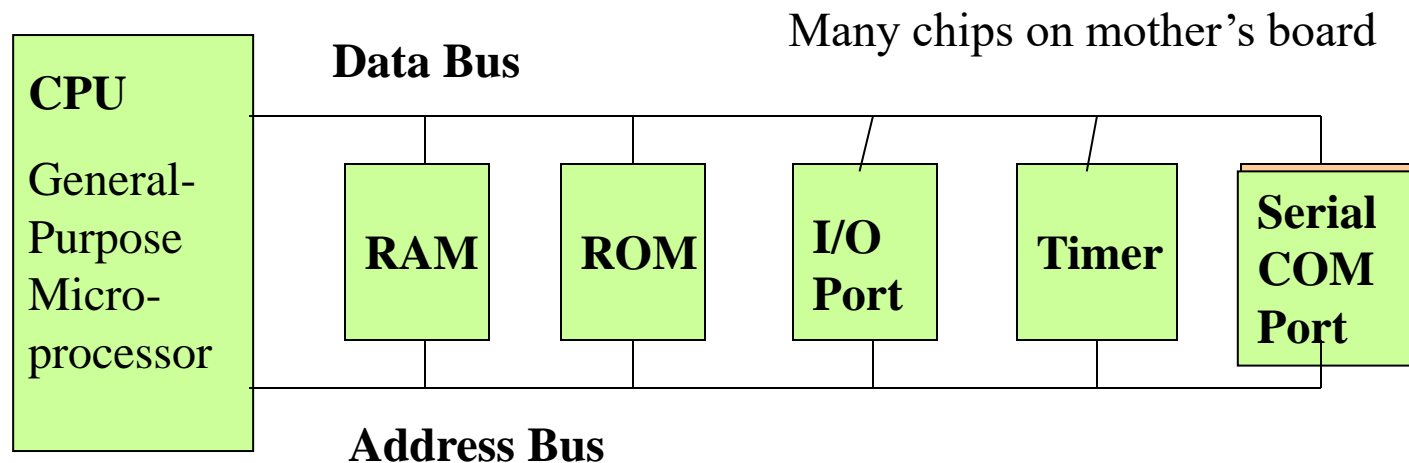
Microprocessor System Components

- ◆ ***Control bus*** (control & coordination signals)
 - A collection of special signals that coordinated the activities of the various units
 - Each type of microprocessor has its own particular set of signals

Microprocessors:

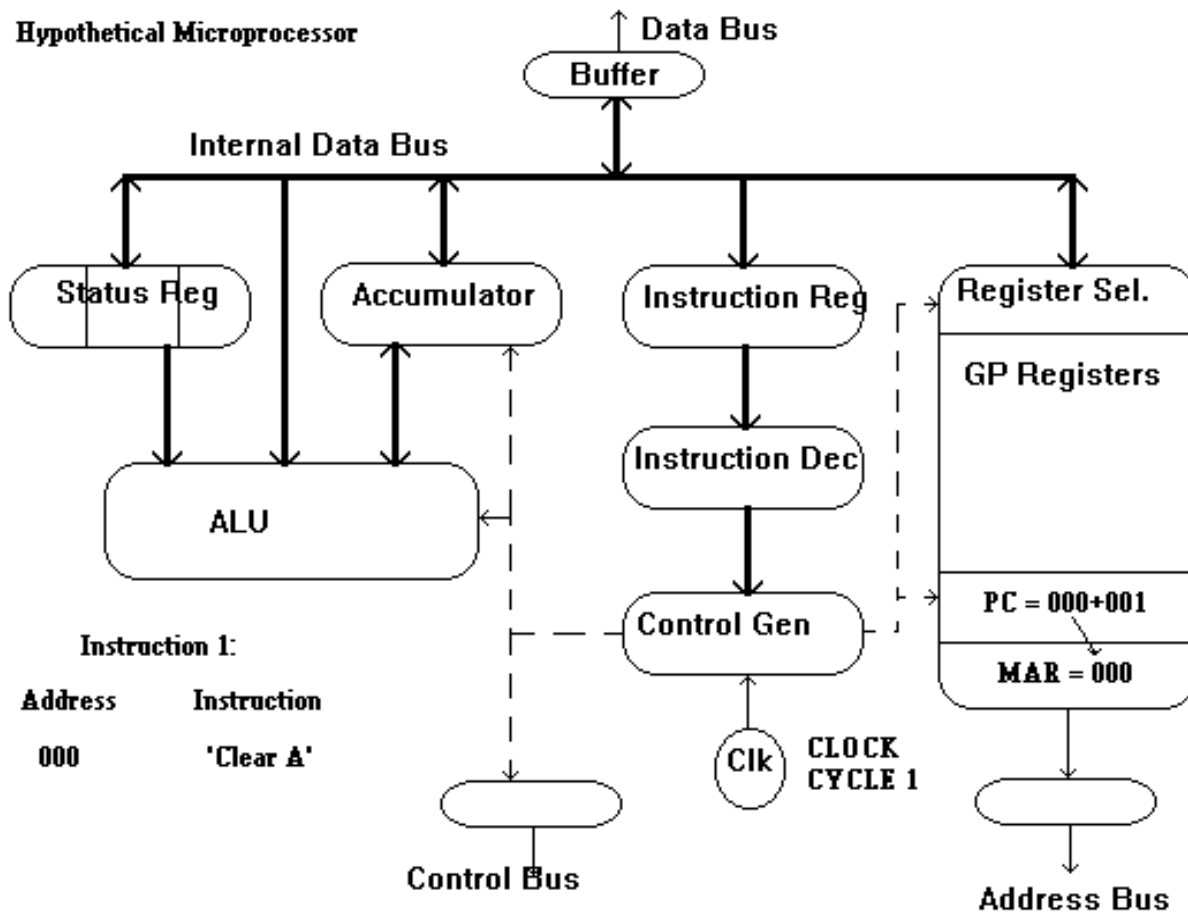
General-purpose microprocessor

- CPU for Computers
- No RAM, ROM, I/O on CPU chip itself
- Example : Intel's x86, Motorola's 680x0



General-Purpose Microprocessor System

Internal Architecture



Internal Architecture

- ◆ Internal architecture: the parts within a microprocessor chip
- ◆ Arithmetic logic unit (ALU)
- ◆ Accumulator
- ◆ Clock signals
- ◆ Control unit
- ◆ Internal data bus
- ◆ Program counter (PC)
- ◆ General purpose registers
- ◆ Instruction register (IR)
- ◆ Memory address register (MAR)
- ◆ Status registers (SR)

Internal Architecture

- ◆ **Arithmetic logic unit (ALU)**: perform logical (AND, OR, NOT, XOR, ...) (Boolean) and mathematical operations (+, −, x, / ...), and store the results in the accumulator
- ◆ **Accumulator**: store all immediate results from ALU
- ◆ **Clock signals**: synchronize and coordinate operations

Internal Architecture

- ◆ **Control unit:** carry out instructions
- ◆ **Internal data bus:** transfer data between the components within the computer
- ◆ **Program counter (PC):**
 - ⊕ A pointer to the current location of the program
 - ⊕ Keeps track of the memory location
 - ⊕ Once an instruction is fetched, it will increment by 1 to move to next instruction
 - ⊕ Should not be modified, or serious error can result
 - ⊕ Need to use special instructions (e.g. JUMP) to modify

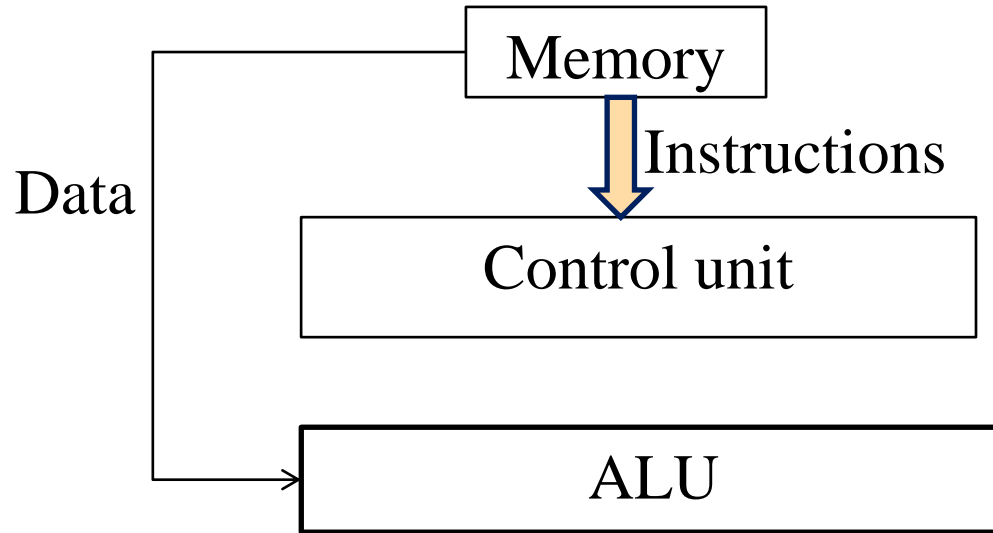
Internal Architecture

- ◆ **General purpose registers**: temporary storage of binary numbers, or additional data during the execution of a program
- ◆ **Instruction register (IR)**: temporary storage of the current instruction of a program
- ◆ **Memory address register (MAR)**:
 - ⊕ Holds the address of the next instruction
 - ⊕ Feeds the address bus with the addresses of the current program
- ◆ **Status register (SR)**:
 - ⊕ Contains flags which indicate certain conditions
 - ⊕ SR must be either 0 or 1

How the CPU Executes Program Instructions

- Before execute an instruction, program instructions and data must be placed into memory from input device or secondary memory
- The CPU performs four steps in executing an instruction:
 1. The control unit gets the instruction from memory.
 2. The control unit decides what the instruction means and directs the necessary data to be moved from memory to the arithmetic/logic unit.
 3. The arithmetic/logic unit performs the actual operation on the data.
 4. The result of the operation is stored in memory or a register.

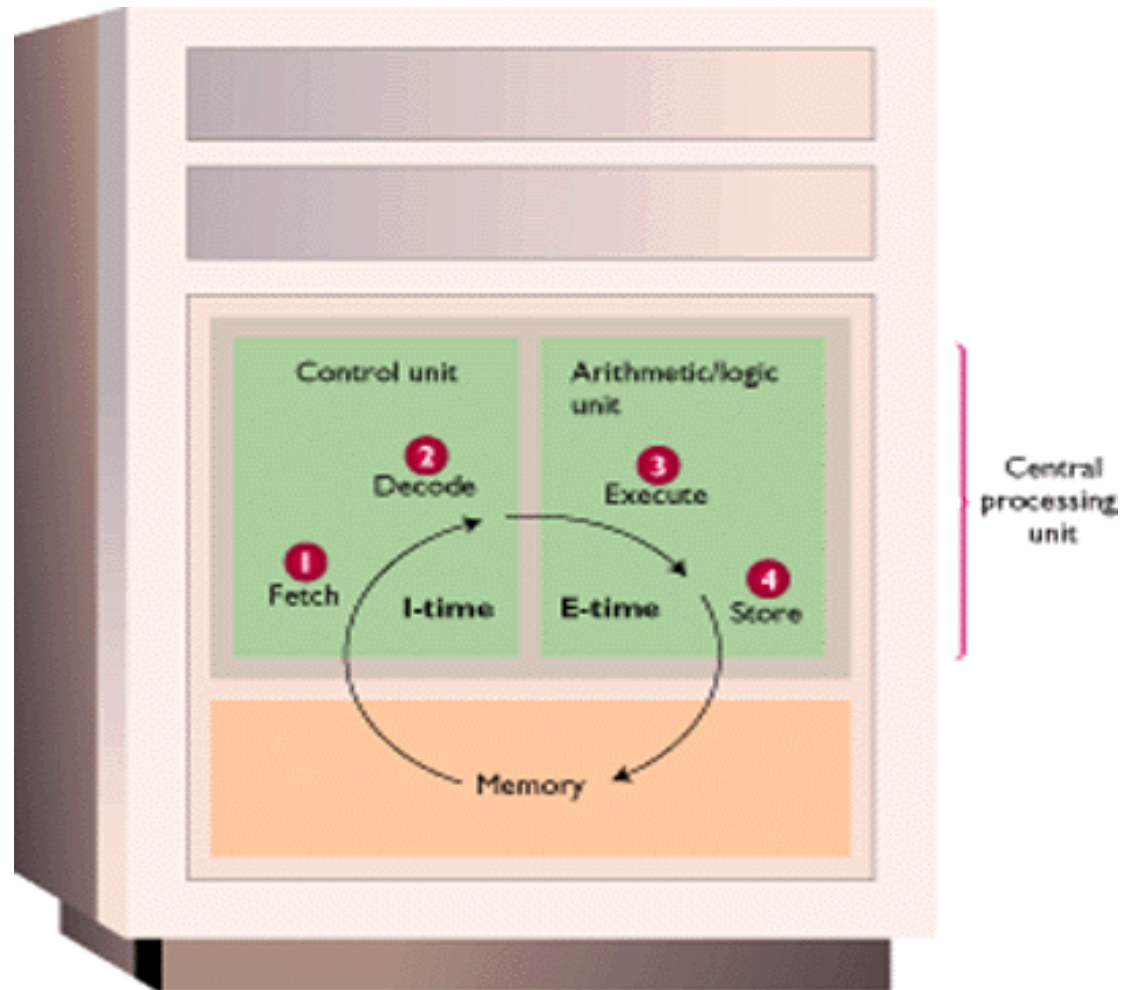
How the CPU Executes Program Instructions



How the CPU Executes Program Instructions

- ◆ The first two steps make up what is called the instruction time, or *fetch cycle*
- ◆ The last two steps make up what is called the execution time, or *execution cycle*.
- ◆ The combination of the fetch cycle and execution cycle is called a *machine cycle*
- ◆ Each type of central processing unit is designed to understand a specific group of instructions called the *instruction set*

How the CPU Executes Program Instructions

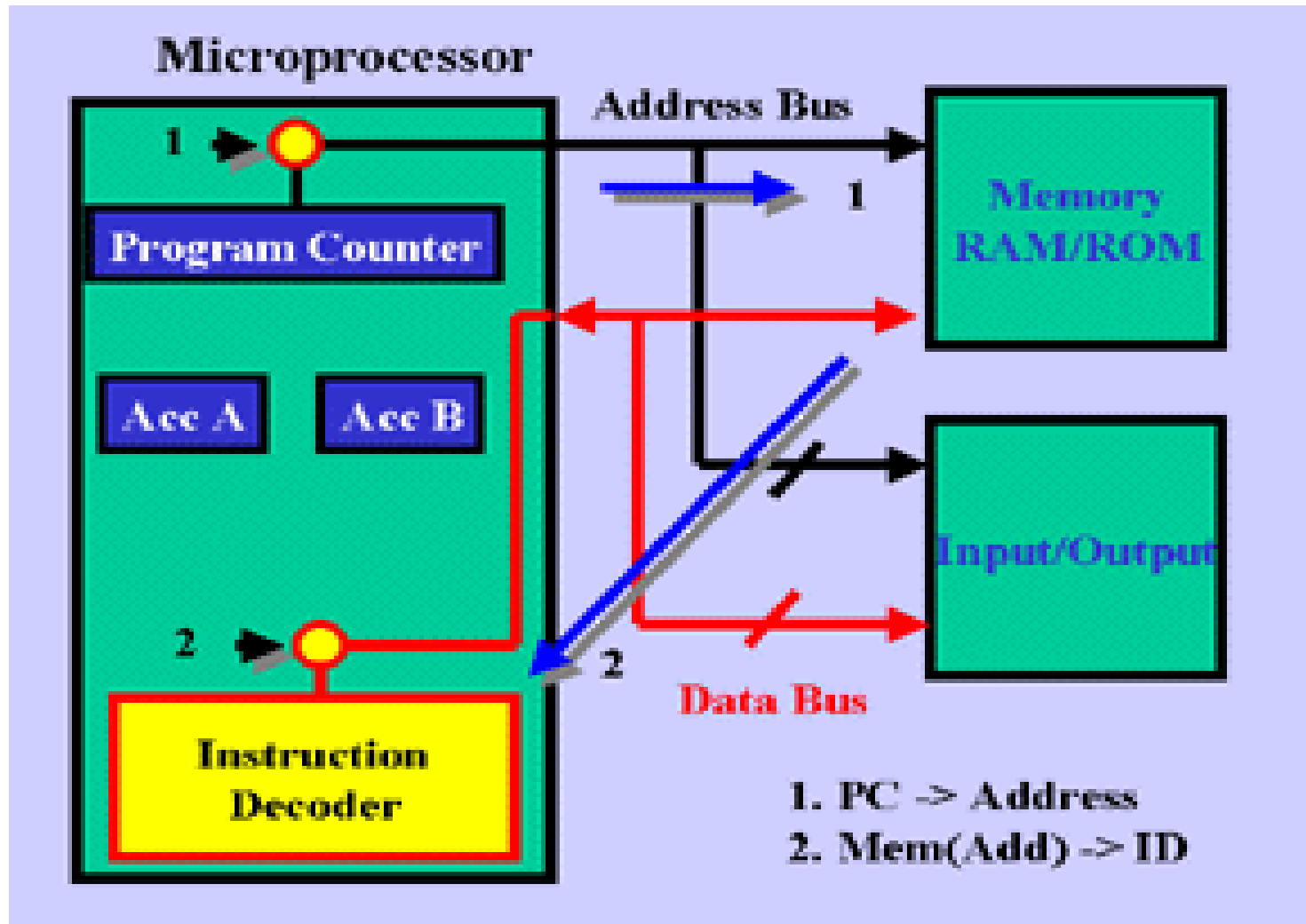


Instruction Fetch

Obtain instruction from input

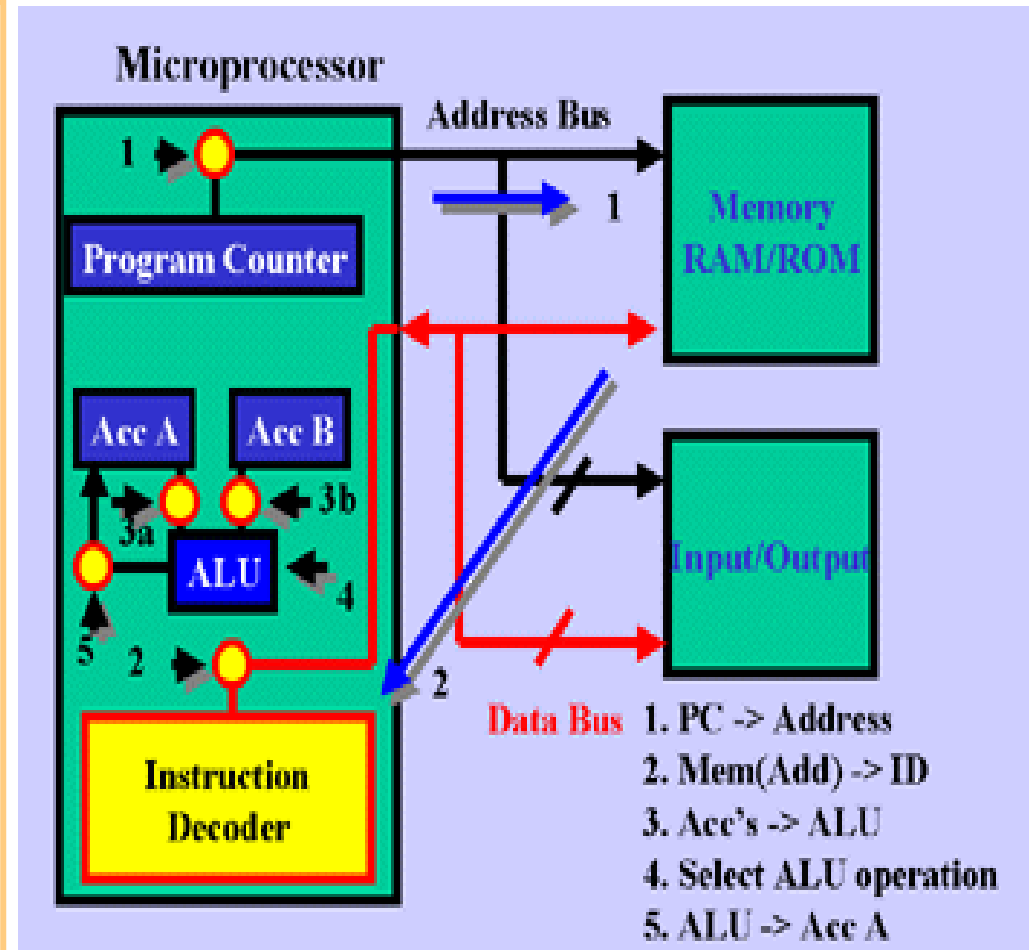
1. PC is enabled to address bus
2. The addressed memory is enabled to data bus
3. The data bus is enabled to the decoded instruction
4. PC is incremented by 1 (+1) to move to the next instruction
 - ◆ Repeat instruction fetch if the instruction is “no operation” (NOP)
 - ◆ Otherwise, execute the instruction

Instruction Fetch



Instruction Execution

- ◆ do the instruction (program)
- ◆ We shall defer the discussion of this topic later as it involves programming concepts

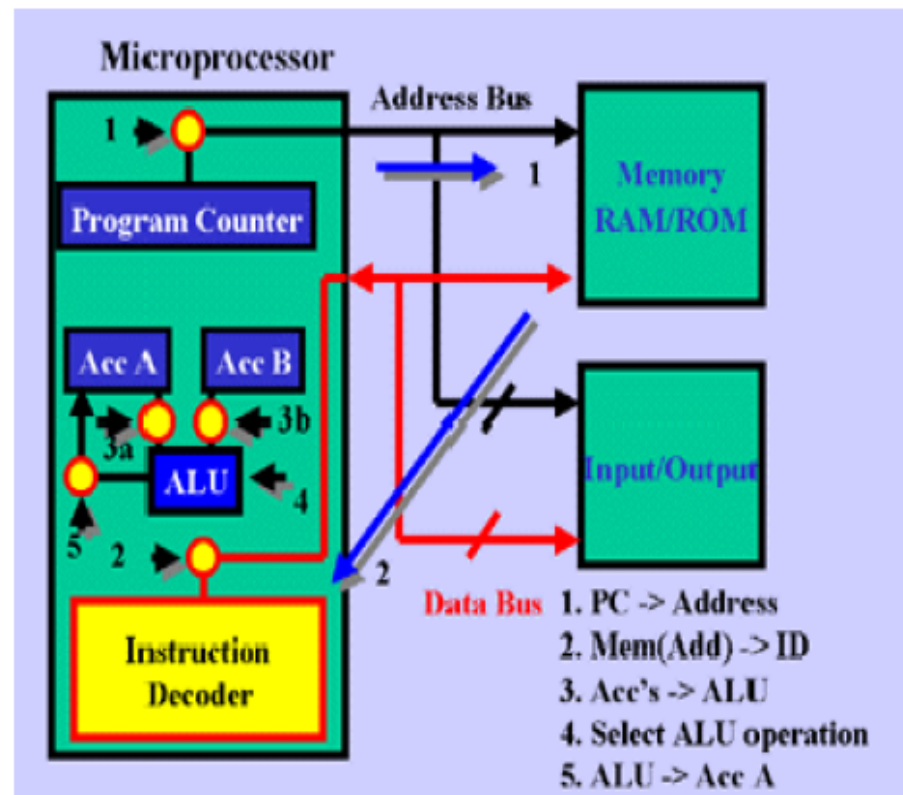




Instruction Execution

Program address	Operation code (opcode)	Mnemonic	Operand(s)
0200	00	NOP	
0201	7400	MOV	A, #00H
0203	245A	ADD	A, #5AH
0205	FF	MOV	R7, A

- ◆ do the instruction (program)
- ◆ We shall defer the discussion of this topic later as it involves programming concepts



Microcontroller

- ◆ a chip having a CPU (i.e. microprocessor chip) and some amount of RAM, ROM, and I/O ports
- ◆ In other words, a *single-chip* computer

Types of Microcontrollers

- 4-bit microcontroller (eg. Epson, Sharp)
- 8-bit microcontroller (eg. Intel 8051, Zilog Z8)
- 16-bit microcontroller (eg. Infineon C166)
- 32-bit microcontroller (eg. Philips ARM7)

Choosing a Microcontrollers

- ◆ Efficient handling of tasks at a reasonable cost
 - ⊕ Speed
 - ⊕ Power consumption
 - ⊕ Number of RAM, ROM...etc.
 - ⊕ Ease and cost of upgrading
 - ⊕ Cost per unit of microcontroller
- ◆ Easy to develop auxiliary products
- ◆ Readily available both at present and in future

Comparison

Microprocessor

- ◆ Data is in byte (8-bit) or word (16-bit) format
- ◆ Many instruction types and modes
- ◆ Hardware includes CPU only
- ◆ Mainly for processing
- ◆ Larger I/O scale – use system buses
- ◆ Memory is byte-oriented
- ◆ Instruction sets are more robust

Microcontroller

- ◆ Data is in bits or byte format
- ◆ Less instruction types and modes
- ◆ Hardware includes CPU and peripherals
- ◆ Mainly for control
- ◆ Smaller I/O scale – use parallel and serial ports
- ◆ Memory is bit-oriented
- ◆ Instruction sets are more compact

Summary

RISC	CISC
1. Instructions takes 1 or 2 cycles	Takes multiple cycles
2. Only load/store instructions are used to access memory.	In addition to load/store instructions, memory access is possible with other instructions also.
3. Instructions executed by hardware	Instruction executed by the micro program
4. Fixed format instructions	Variable format instructions
5. Few addressing modes	Many addressing modes
6. Few instructions	Complex instruction set
7. Most of them have multiple register banks	Single register bank

table: Comparison of RISC and CISC

Summary

- ◆ Understanding microprocessors and microcontrollers
- ◆ The components in a microprocessor system
- ◆ Internal architecture of a microprocessor chip
- ◆ Instruction fetch and execution
- ◆ Comparing microprocessors & microcontrollers

Read reference

- ◆ The 8051 Microcontroller and Embedded Systems - Using Assembly and C, Mazidi
 - Chapter 0 P.9 – P.19
 - Chapter 1 P.23 – P.32