Introduction to Microcontrollers
Introduction

• It is hard to imagine the present world of electronic devices without the microprocessor.
  – Cash register, scales, ovens, washing machine, alarm clock, thermostats, and ignition system, etc.
• 1971 (Microprocessors)
  – Intel 8080, Motorola 6800, RCA 1801, MOS Technology and Zilog Z80.
• 1976 (Microcontrollers)
  – Intel 8748 (MCS-48 family) \( \approx 17,000 \) transistors
    • CPU, 1KB EPROM, 64B RAM, 27 I/O pins, and an 8-bit timer.
    • Used in washing machines and traffic light controllers.
• 1980 (Microcontrollers)
  – Intel 8051 (MCS-51 family) \( \approx 60,000 \) transistors
    • CPU, 4KB ROM, 128B RAM, 32 I/O Pins, a serial port, two 16-bit timers.
  – Siemens SAB80515
    • 68-pin parade with six 8-bit I/O ports, 13 interrupt sources, and an 8-bit A/D converter with 8 input channels.
A Computer System

• Computer is defined by two key traits:
  – The ability to be programmed to operate on data without human intervention.
  – The ability to store and retrieve data.

• More generally, a computer system includes
  – Peripheral devices for communicating with the human. (hardware)
  – Program that process data. (software)

• A Computer system contains
  – Central process unit (CPU) connected to random access memory (RAM) and read-only memory (ROM) via the address bus, data bus, and control bus. Interface circuits connect the system buses to peripheral devices. (see next slide)
Block Diagram of A Microcomputer System

- CPU  $\rightarrow$ fetching instructions and executing instructions.
- Instruction Set
  - Add, subtract, multiply, divide, AND, OR, NOT, data movement, branch operations  $\rightarrow$ represented by a set of binary codes called the Instruction Set.
The Central Processing Unit (CPU)

- **Register** for the temporary storage of information.
- **ALU** for performing operations on this information.
- **Instruction decode and control unit** that determine the operation to perform.
- **IR** holds the binary code for each instruction as it is executed.
- **PC** holds the memory address of the next instruction to be executed.
Fetching and Executing the Instructions

- Fetching Steps
  1. The contents of PC are placed on the address bus.
  2. A READ control signal is activated.
  3. Data (the instruction opcode) are read from RAM and placed on data bus.
  4. The opcode is latched into the CPU’s IR.
  5. The PC is incremented to prepare for the next fetch from memory.

- Executing Steps
  - Decoding the opcode and generating control signals to gate internal registers in and out the ALU and to signal ALU to perform the specified operation.
RAM and ROM

- RAM is read/write memory. It is volatile (the contents are lost when power is removed). User program and data are loaded into RAM for execution.

- ROM is read-only memory. It is nonvolatile. ROM is used to hold the short, frequently used software routines that perform input/output operations.
The Buses: Address, Data, Control

- **Address Bus**
  - 16-bit address can access \(2^{16} = 65,536\) locations. \(2^{16} = 2^6 \times 2^{10} = 64K\) locations. The address bus is unidirectional.

- **Data Bus**
  - Evidently computers spend up to 2/3 of their time simply moving data. Since the majority of move operations are between a CPU register and external RAM or ROM, the number of lines (width) of the data bus is important for overall performance.
  - 16-bit computer refers to a computer with 16 lines on its data bus.
  - The overall computing power increasing as the width of the data bus increases. The limitation by-width is bottleneck!!
  - The data bus is bidirectional. (Read or Write)

- **Control Bus**
  - Control signals are timing signals supplied by CPU to synchronize the movement of the information on the address bus and data bus. Such as CLOCK, READ and WRITE.
I/O Devices

- Mass storage devices
  - Online storage
    - is available to CPU without human intervention upon the request of a program. (such as magnetic disk)
  - Archival storage
    - Holds data that are rarely needed and require manual loading onto the system. (such as magnetic tapes or disks, optical discs: CD-ROM)

- Human interface devices
  - Keyboard, mouse, Printer, CRT

- Control/monitor devices
  - Temperature control of a building, home security, elevator control, home appliance control.
  - The hardware and software interfacing of these devices to microcontroller is one of the main themes in this book.
Hardware vs. Software

- Whereas the early days of computing witnessed the materials, manufacturing, and maintenance costs of computer hardware far surprising the software costs.
- Today, with the mass-produced LSI (Large-Scale Integrated) chips. Hardware costs are less dominant. It is the labor-intensive job of writing, documenting, maintaining, updating, and distributing software that constitutes the bulk of the expense in automating a process using computers.
Three Levels of Software

• Input/output subroutines
  – Directly manipulate the hardware of the system
    • Reading characters from the keyboard.
    • Writing characters to the CRT.
    • Reading blocks of information from the disk.
  – Since these subroutines are so intimately linked to the hardware, they are written by the hardware designers and are stored in ROM (They are the BIOS – basic input/output system – on the IBM PC for example).

• The operating system
  – Is a large collection of programs that come with the computer system and provide the mechanism to access, manage, and effectively utilize the computer’s resources.
  – These abilities exist through the operating system’s command language and utility programs which in turn facilitate the development of application software.

• Application software (user interface).
Microprocessors vs. Microcontroller

• Hardware architecture
  – Whereas a microprocessor is a single-chip CPU, a microcontroller contains a CPU and much of the remaining circuitry of a complete microcomputer system in a single IC.
  – Microcontrollers (as control oriented devices) are called upon to respond to external stimuli in real time.
  – They must perform fast context switching, suspending, one process while executing another in response to an ‘event’ (opening the oven’s door).

• Applications
  – Microcontrollers are suitable to ‘control’ of I/O devices in designs requiring a minimum component count, whereas microprocessors are suitable to ‘processing’ information in computer systems.
Microprocessors vs. Microcontroller

• Instruction set features
  – Microprocessor instruction sets are ‘processing intensive’ implying they have powerful addressing modes with instructions catering to operations on large volumes of data. Microcontrollers, on the other hand, have instruction sets catering to the control of inputs and outputs.
  – Microcontrollers have built-in circuitry and instructions for input/output operations, event timing, and enabling and setting priority levels for interrupts caused by external stimuli. Microprocessors often require additional circuitry (serial interface Ics, interrupt controllers, timers, etc.) to perform similar operations.
  – A tight encoding scheme for the instruction set is essential for microcontrollers. This is rarely a feature of microprocessors; their powerful address modes bring with them a less-than-compact encoding of instructions.
New Concepts

• Consider an examples microwave ovens, programmable thermostats, electronic scales, and even cars.
  – The electronics within each of these products typically incorporates a microcontroller interfacing to push buttons, switches, lights, and alarms on a front panel.

• Computer systems have a high RAM-to-ROM ratio, with user programs executing in a relatively large RAM space and hardware interfacing routines executing in a small ROM space.

• Microcontrollers, on the other hand, have a high ROM-to-RAM ratio. The control program is stored in ROM, while RAM is used only for temporary storage.