

Microprocessor and Microcontroller Fundamentals

AJLONTECH



Admin

- ◆ Course materials available online
- ◆ <http://ece.citadel.edu/hayne/>
 - Students are encouraged to print lecture slides in advance and use them to take notes in class

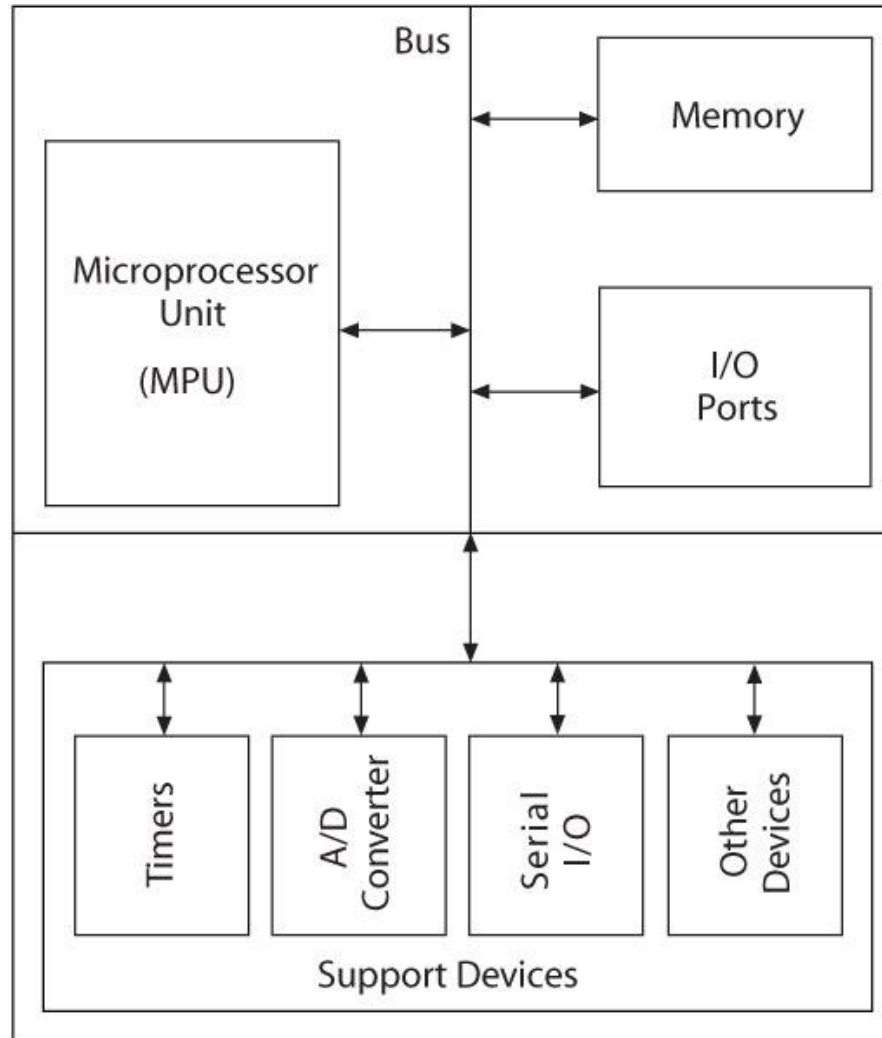
Microcontrollers

- ◆ Embedded Systems
 - Operations managed behind the scenes by a microcontroller
- ◆ Microcontroller (MCU)
 - Integrated electronic computing device that includes three major components on a single chip
 - Microprocessor (MPU)
 - Memory
 - I/O (Input/Output) ports

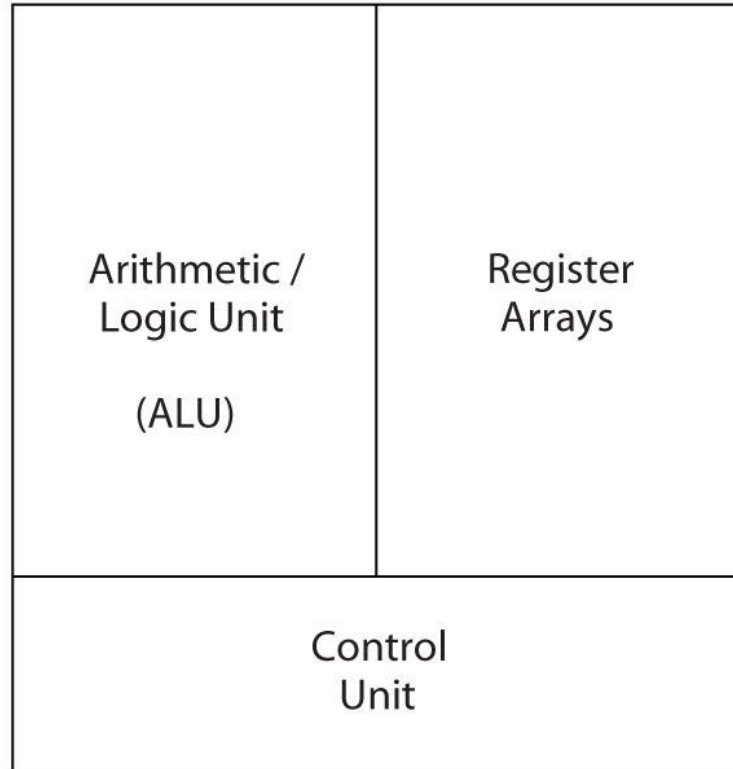
Microcontrollers

- ◆ Support Devices
 - Timers
 - A/D converter
 - Serial I/O
- ◆ Common communication lines
 - System Bus

Block Diagram



Microprocessor (MPU)

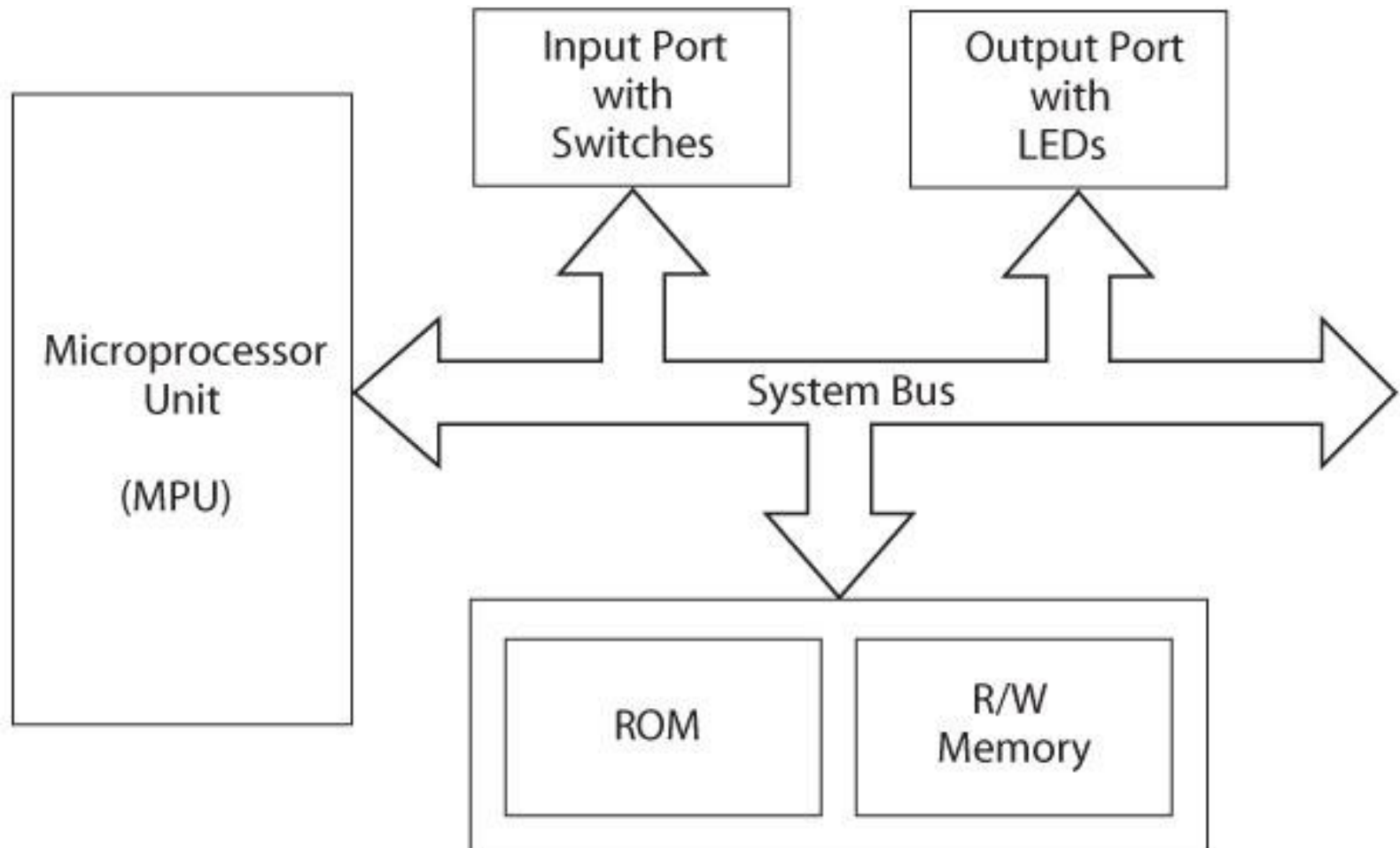


- ◆ MPU (CPU)
 - Read instructions
 - Process binary data

Input/Output (I/O)

- ◆ Input Devices
 - Switches and Keypads
 - Provide binary information to the MPU
- ◆ Output devices
 - LEDs and LCDs
 - Receive binary information from the MPU

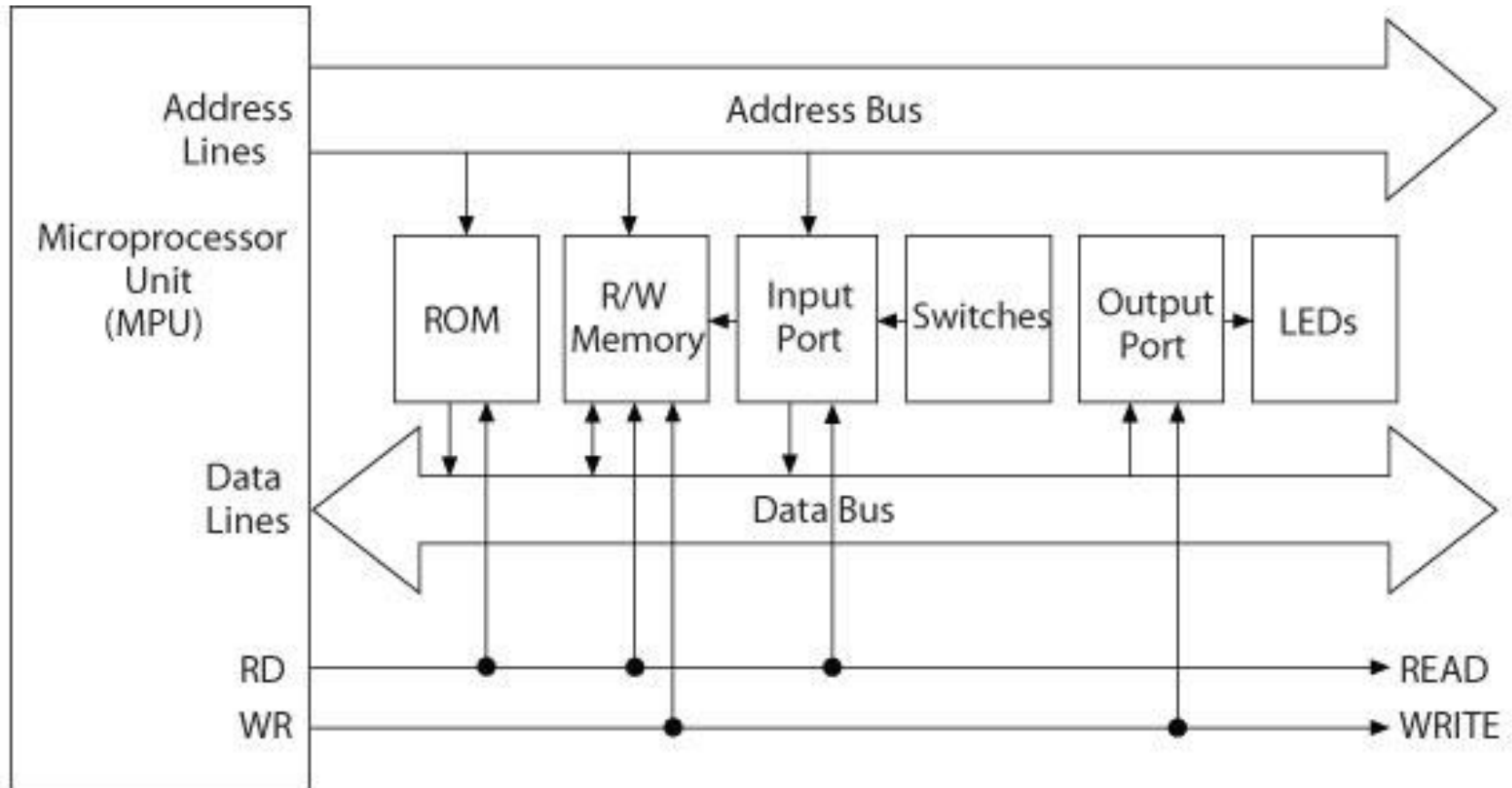
Microprocessor-Based Systems



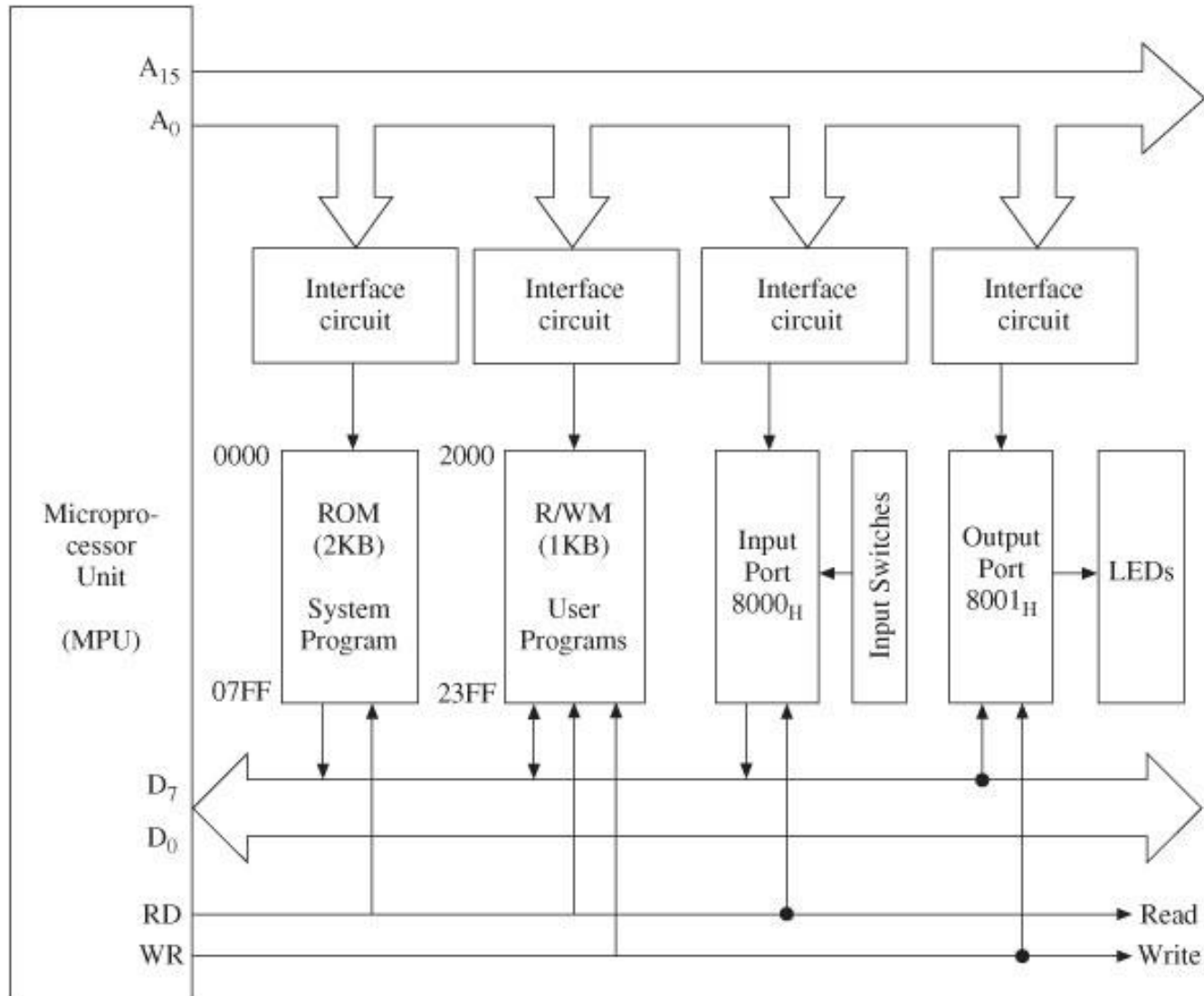
Microprocessor Architecture

- ◆ MPU communicates with Memory and I/O using the System Bus
 - Address bus
 - Unidirectional
 - Memory and I/O Addresses
 - Data bus
 - Bidirectional
 - Transfers Binary Data and Instructions
 - Control lines
 - Read and Write timing signals

Microprocessor-Based System



Example Microprocessor System



Software

- ◆ Machine Language
 - Binary Instructions
 - Difficult to decipher and write
 - Error-prone
 - All programs converted into machine language for execution

Instruction	Hex	Mnemonic	Description	Processor
10000000	80	ADD B	Add reg B to Acc	Intel 8085
00101000	28	ADD A, R0	Add Reg R0 to Acc	Intel 8051
00011011	1B	ABA	Add Acc A and B	Motorola 6811

Software

- ◆ Assembly Language
 - Machine instructions represented in mnemonics
 - One-to-one correspondence
 - Efficient execution and use of memory
 - Machine-specific

Software

- ◆ High-Level Languages
 - BASIC, C, and C++
 - Written in statements of spoken languages
 - Machine independent
 - Easy to write and troubleshoot
 - Larger memory and less efficient execution

Data Format (8-bit)

- ◆ Unsigned Integers
 - All eight bits represent the magnitude of a number
 - Bit7 to Bit0
 - Range 00_{H} to FF_{H} (0_{10} to 255_{10})

Data Format (8-bit)

- ◆ Signed Integers
 - 2's Complement
 - Bit7 is sign bit
 - Positive numbers: 00_{H} to $7F_{\text{H}}$ (0_{10} to 127_{10})
 - Negative numbers: 80_{H} to FF_{H} (-1_{10} to -128_{10})

Data Format (8-bit)

- ◆ Binary Coded Decimal Numbers (BCD)
 - 8-bit number divided into two groups of four
 - Each group represents a decimal digit from 0 to 9
 - A_H through F_H are invalid
 - Example: $0010\ 0101_{\text{BCD}} = 25_{10}$

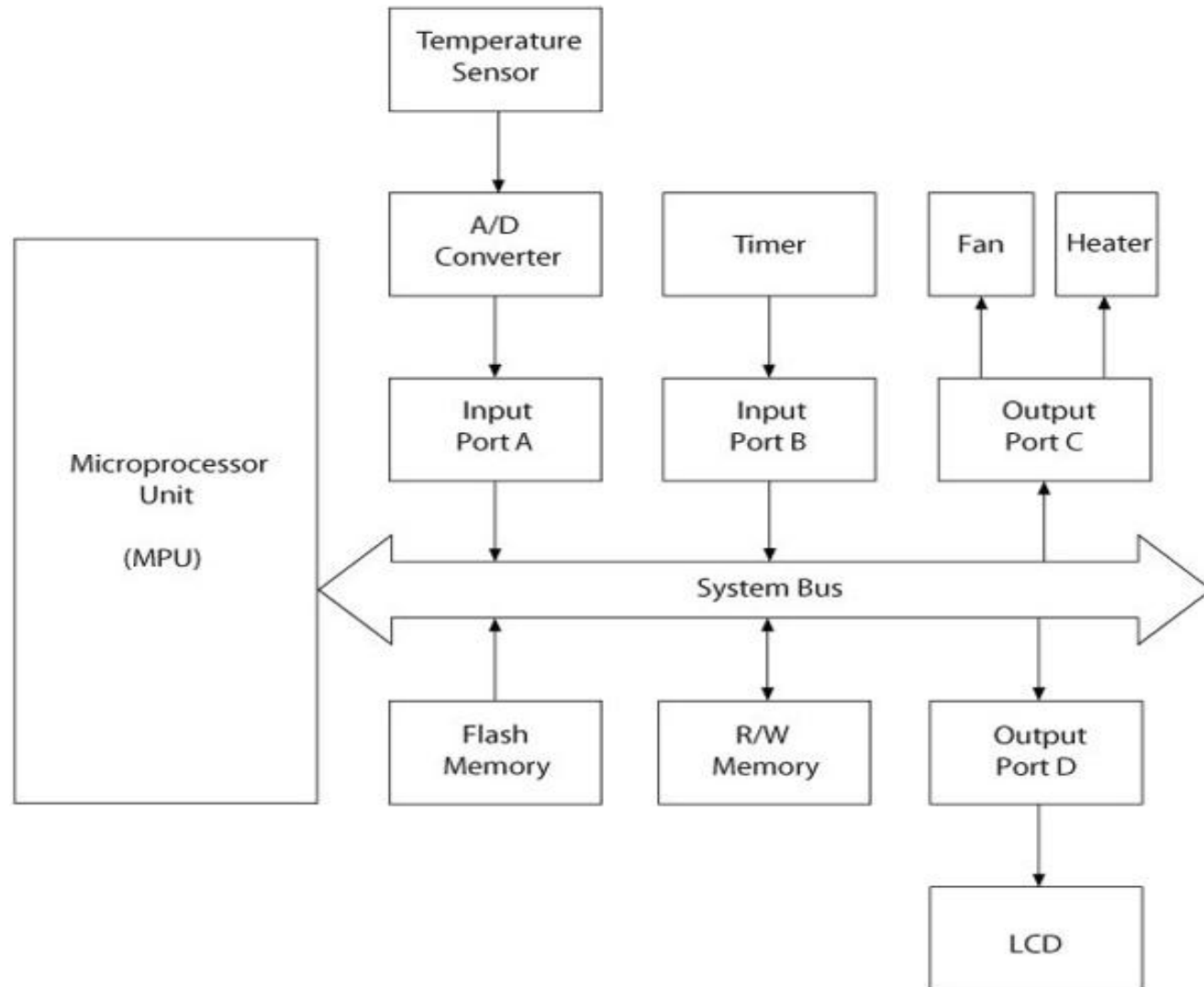
Data Format (8-bit)

- ◆ American Standard Code for Information Interchange (ASCII)
 - 7-bit alphanumeric code with 128 combinations (00_{H} to 7F_{H})
 - Represents English alphabet, decimal digits from 0 to 9, symbols, and commands

MPU-Based Systems

- ◆ System hardware
 - Discrete components
 - Microprocessor, Memory, and I/O
 - Components connected by buses
 - Address, Data, and Control
- ◆ System software
 - Group of programs that monitors the functions of the entire system

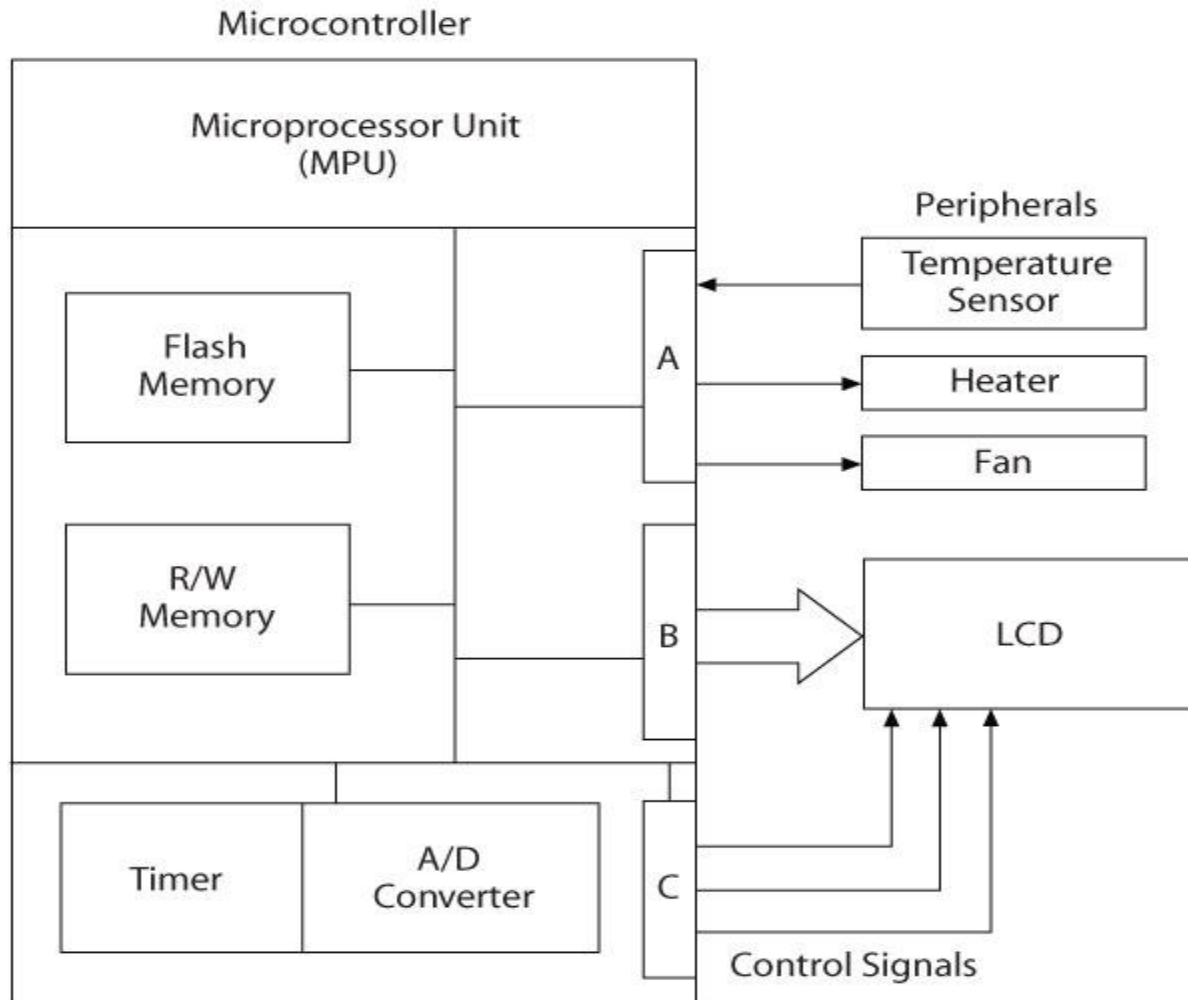
MPU-Based System



MCU-Based Systems

- ◆ Microprocessor, memory, I/O ports, and support devices on a single chip
- ◆ Buses generally not available to a system designer
- ◆ I/O ports generally multiplexed and can be programmed to perform different functions

MCU-Based System



Computer Architectures

- ◆ Princeton versus Harvard Architecture
- ◆ CISC versus RISC processors
- ◆ Microprocessors and Microcontrollers