EL-GY 6483 Real Time Embedded Systems

DEFINITION: EMBEDDED SYSTEM

- "Any device that includes a programmable computer but is not itself intended to be a general purpose computer."[1]
- "Information processing systems embedded into enclosing products." [2]
- Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems." [3]

1Wayne Wolf
2Peter Marwedel, TU Dortmund
3Edward Lee, Berkeley

EXAMPLES

Embedded systems:

- Mars rover
- Cardiac pacemaker
- Google glass

Not embedded systems:

- Laptop computer
- Desktop computer
- Cloud server

CHARACTERISTICS:

Real time: responds to input within a predictable, finite period (deadline). The correctness depends not only on the logical result but also the time it was delivered.

SYSTEM	TYPE	COMMENT
Missile Launcher	Hard	Missing a deadline is a total system failure
Weather Predictions	Firm	Result is not useful after its deadline. Occasional misses degrade quality of service, but are not catastrophic.
Live Video	Soft	Result is less useful after its deadline. Misses degrade quality of service

CHARACTERISTICS

Reactive: responds to input from physical environment.

This input is not sequential, often "messy":

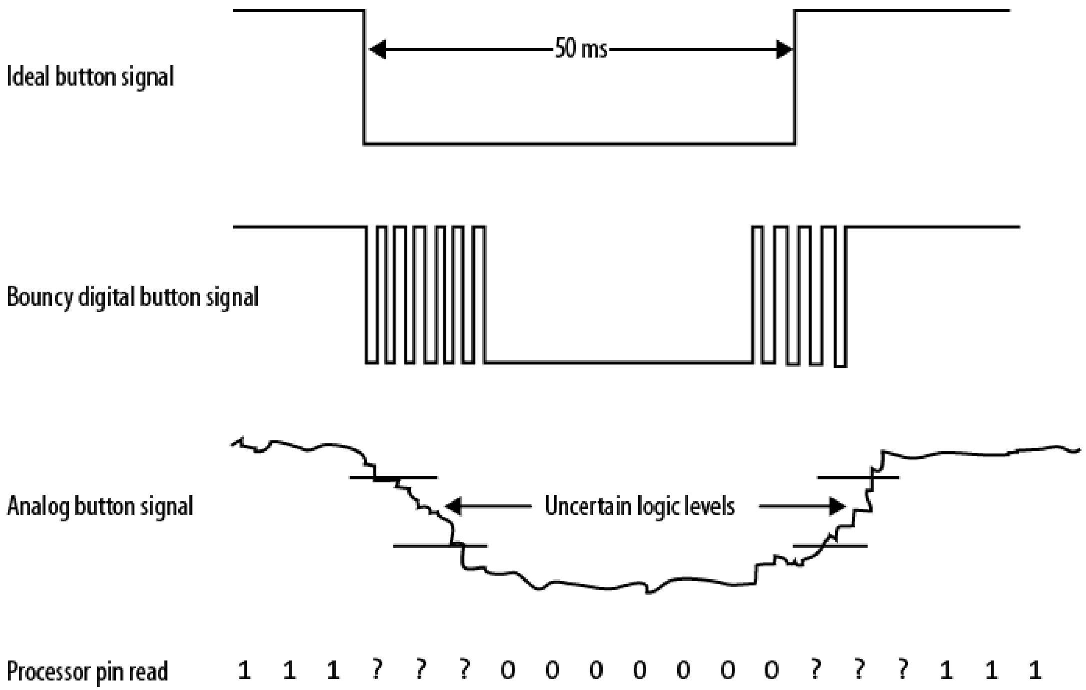


Image source: "Making embedded systems", Elicia White, O'Reilly

CHARACTERISTICS

Dependability: impact on physical world, may have safety implications.

Example: Therac-25

Therac-25 was a radiation therapy machine used in the 1980s. Because of programming errors, several people suffered massive radiation overdoses and some died.

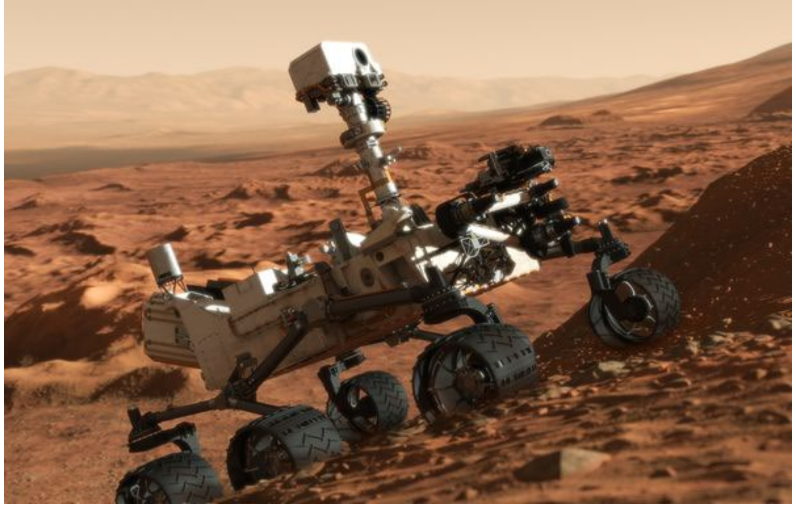
CHARACTERISTICS

Reliability: survival probability.

It is often impossible to fix bugs "in the field" and sometimes it's impossible to replace the system.

Example: Mars Curiosity Rover

Chipset used is allowed to have one and only one "bluescreen" every 15 years (even under heavy radiation).



CHARACTERISTICS

Efficiency: use limited resources effectively.

Embedded system hardware is typically subject to strict size, weight, and manufacturing cost restraints.

REVIEW: COMPUTER ORGANIZATION

UNITS OF DATA

- Bit: A single binary digit, that can have either value 0 or 1.
- Byte: 8 bits.
- Nibble: 4 bits.
- Word: varies by CPU, typically equals size of register. (For this class: 32 bits)

REPRESENTING DATA

- Binary (base 2): 0's and 1's. One digit is a bit.
- Hexadecimal (base 16): 0,…9, A,B,C,D,E,F. One digit is a nibble.
- Octal (base 8): 0,…7. One digit is a byte.

Examples:

0xE in hex = 14 in decimal = 1110 in binary

0x64 in hex = 100 decimal = 1100100 in binary

Data Format (8-bit)

(1 of 4)

- Unsigned Byte: All eight bits (Bit0 to Bit7) represent the magnitude of a number
  - Range 0 to FF in Hex and 0 to 255 in decimal

Unsigned

Signed

Data Format (8-bit)

(2 of 4)

Unsigned
Signed

- Signed Byte: Seven bits (Bit0 to Bit6) represent the magnitude of a number
  - The eighth bit (Bit7) represents the sign of the number. The number is positive when Bit7 is zero and negative when Bit7 is one.
  - Positive Numbers: 0 to 7F (0 to 127)
  - Negative Numbers: 80 to FF (-1 to -128)
  - All negative numbers are represented in **2's compliment**

Data Format (8-bit)

(3 of 4)

- Binary Coded Decimal Numbers (BCD)
  - 8 bits of a number divided into groups of four, and each group represents a decimal digit from 0 to 9
  - Four-bit combinations from A through F in Hex are invalid in BCD
  - Example: **0010 0101** represents the binary coding of the decimal number 25d which is different in value from 25H

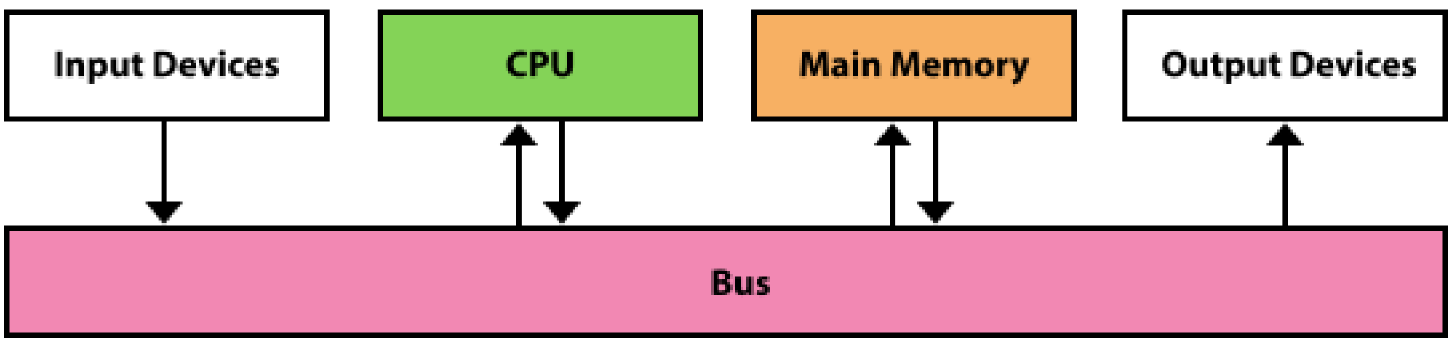
Data Format (8-bit)

(4 of 4)

- American Standard Code for Information Interchange (ASCII)

    - Seven-bit alphanumeric code with 128 combinations (00 to FF)
    - Represents English alphabet, decimal digits from 0 to 9, symbols, and commands

SIMPLE MODEL

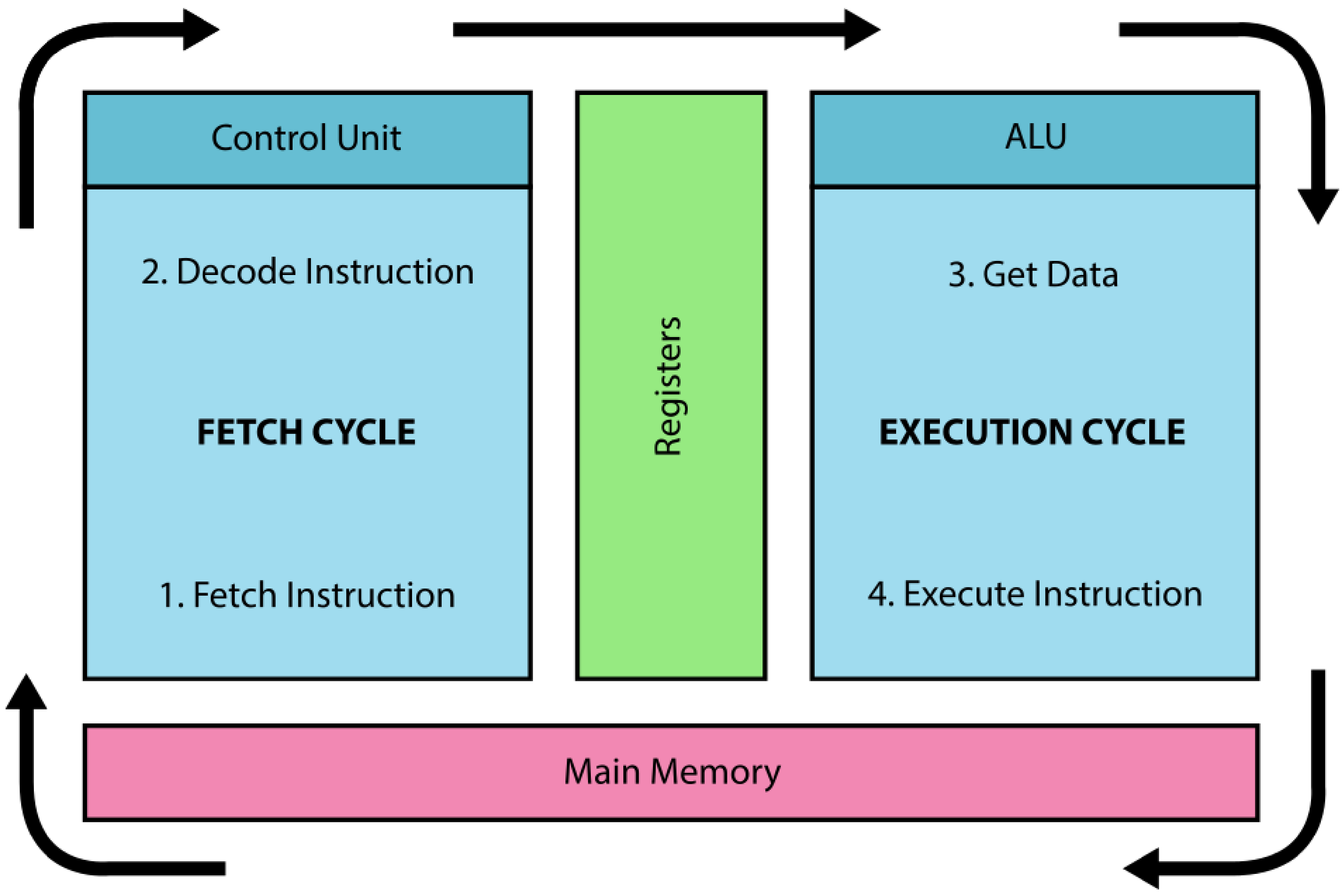


CPU

The role of the CPU is to execute a list of instructions (program), usually to manipulate data.

- Control Unit: directs the operation of the units with timing and control signals. Includes two special registers:
  - Instruction Register (IR) contains the instruction that is being executed
  - Program Counter (PC) contains the address of the next instruction to be executed
- Arithmetic/Logic Unit: performs basic arithmetic operations (add, subtract, multiply) and logical operations (and, or not). Uses registers for fast access to data operands.

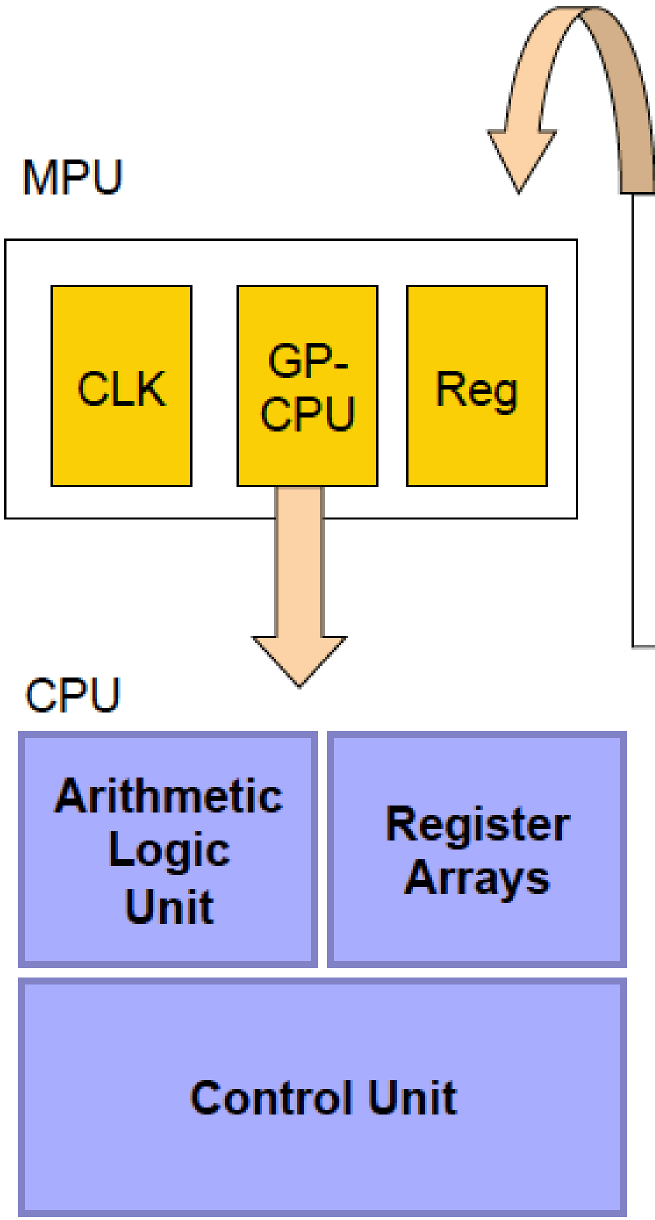
FETCH – DECODE – EXECUTE - STORE

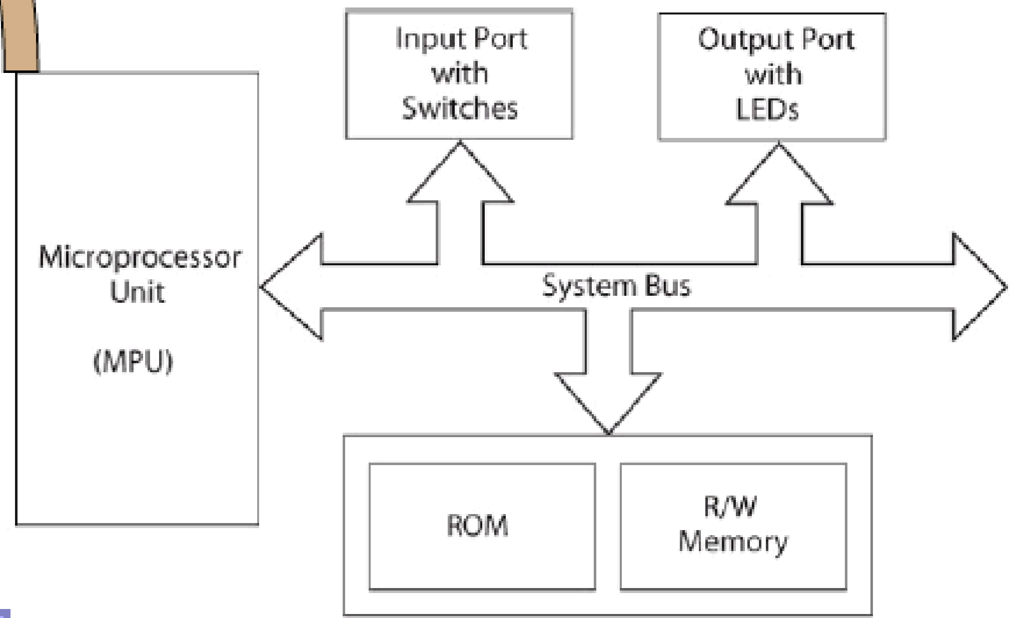


A little about Microprocessor-based Systems ….

Evolution

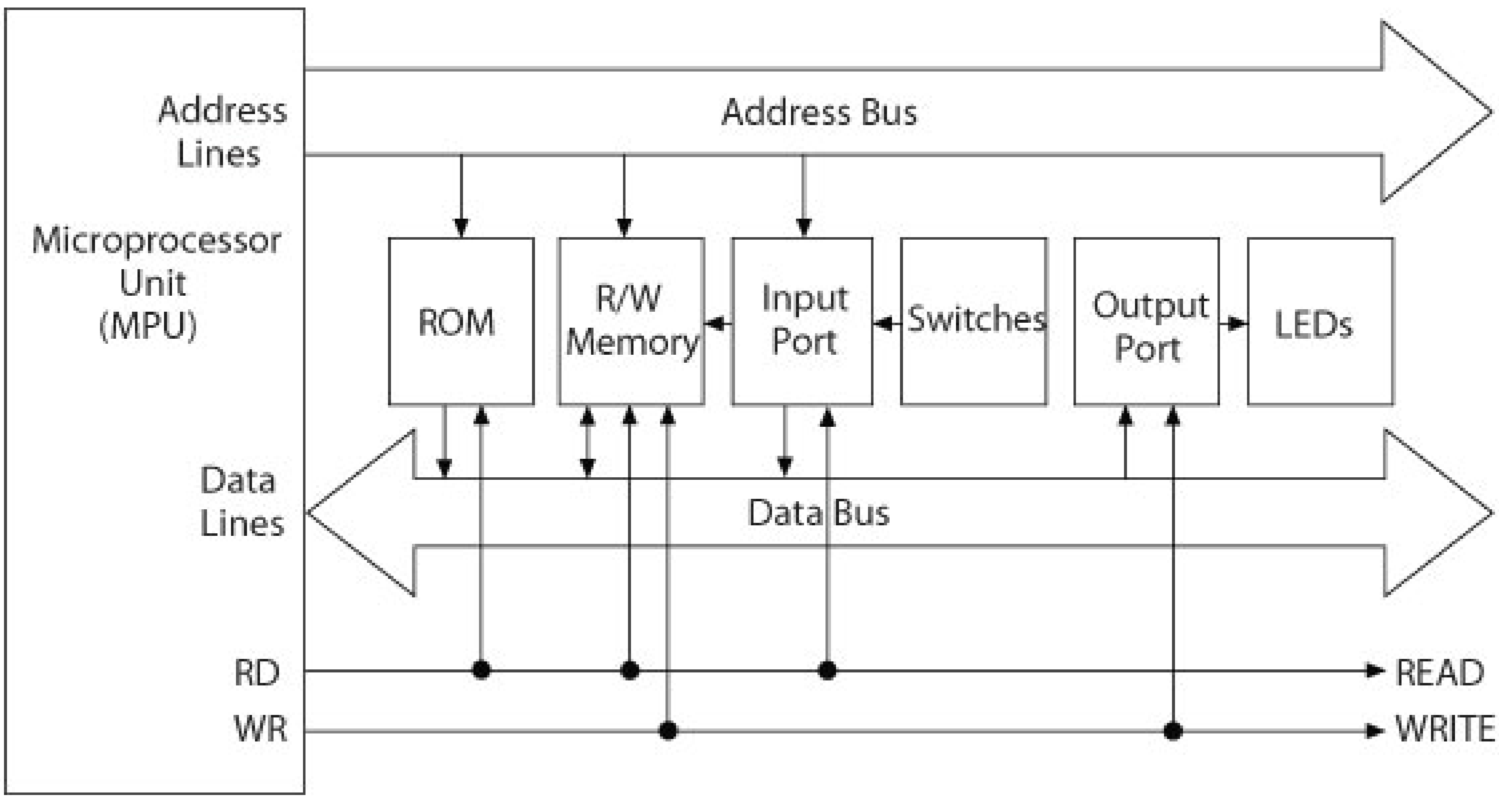
- First came transistors
- Integrated circuits
  - SSI (Small-Scale Integration) to ULSI
  - Very Large Scale Integration circuits (VLSI)
- 1 – Microprocessors (MPU)
  - Microcomputers (with CPU being a microprocessor)
  - Components: Memory, CPU, Peripherals (I/O)
- 2 – Microcontroller (MCU)
  - Microcomputers (with CPU being a microprocessor)
  - Many special function peripheral are integrated on a single circuit
  - Types: General Purpose or Embedded System (with special functionalities)





Microprocessor-based System

Microprocessor-Based System with Buses: Address, Data and Control



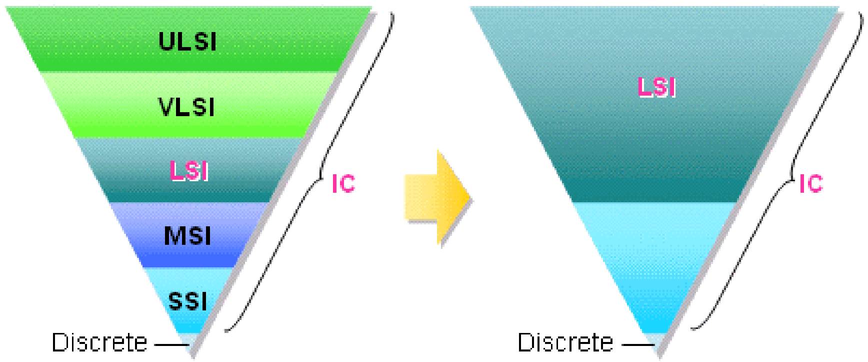
Microprocessor-Based Systems

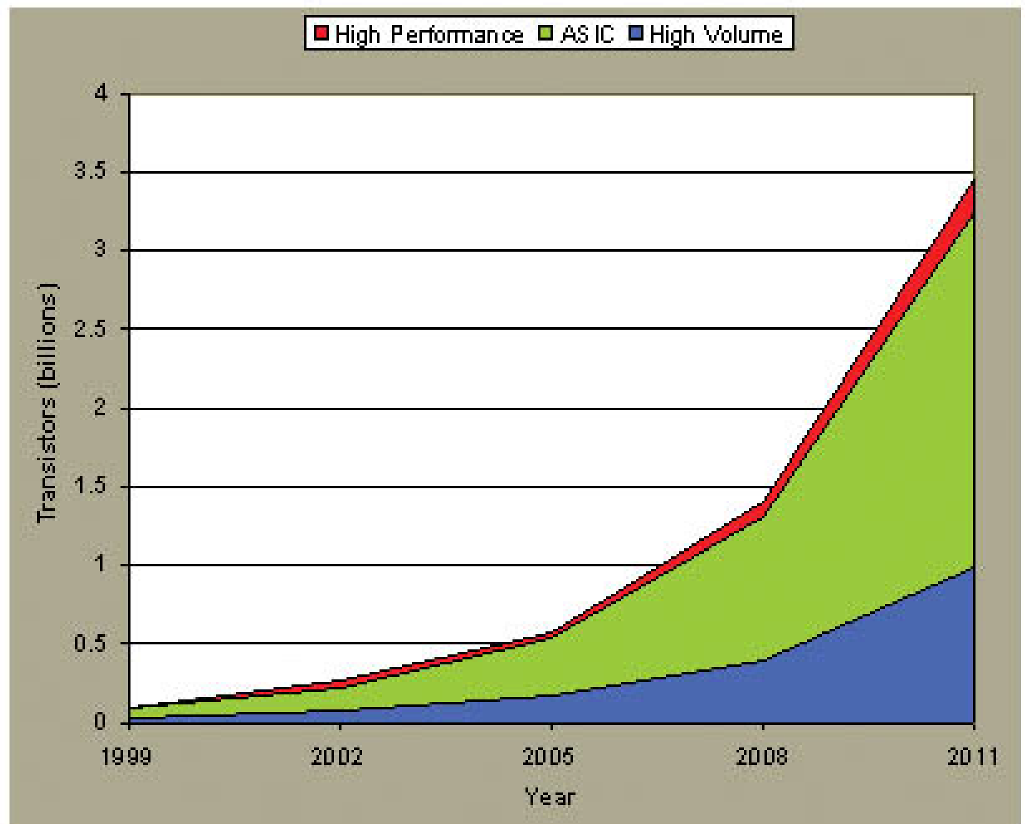
Microprocessor

- The microprocessor (MPU) is a computing and logic device that executes binary instructions in a sequence stored in memory
- Characteristics:
  - General purpose central processing unit (CPU)
  - Binary
  - Register-based
  - Clock-driven
  - Programmable

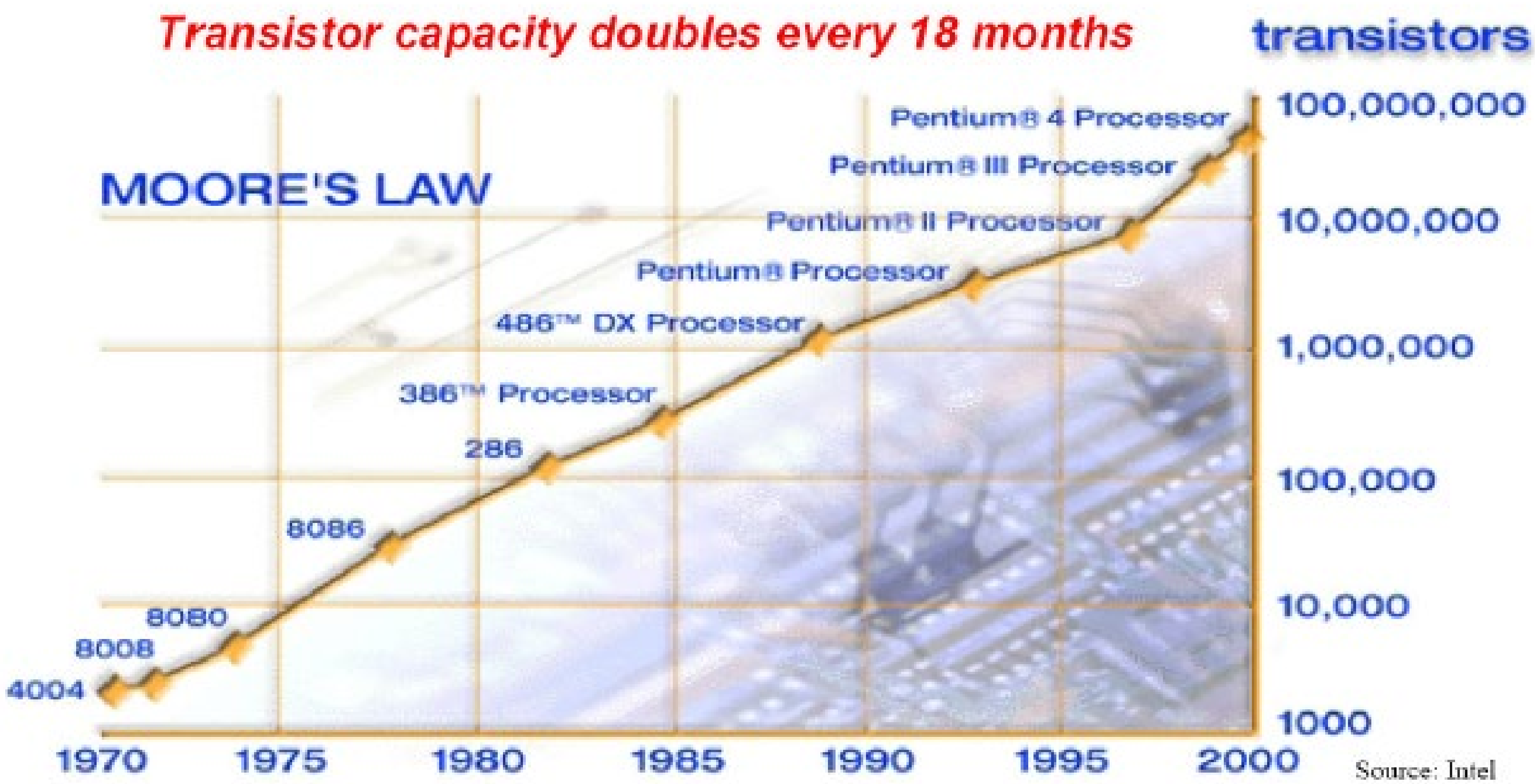
Integrated Circuits

- Advances in manufacturing allowed packing more transistors on a single chip
- Transistors and Integrated Circuits from SSI(Small-Scale Integration) to ULSI
- Birth of a microprocessor and its revolutionary impact





Evolution of CPUs



In 1965, Gordon Moore, co-founder of Intel, indicated that the number of transistors per square inch on integrated circuits had doubled every year since the integrated circuit was invented. Moore predicted that this trend would continue for the foreseeable future.

Microprocessors

- Noyce and Gordon Moore started Intel
- Intel designed the first calculator
- Intel designed the first programmable calculator
- Intel designed the first microprocessor in 1971
  - Model 4004
  - 4-bit; 2300 transistors, 640 bytes of memory, 108 KHz clock speed

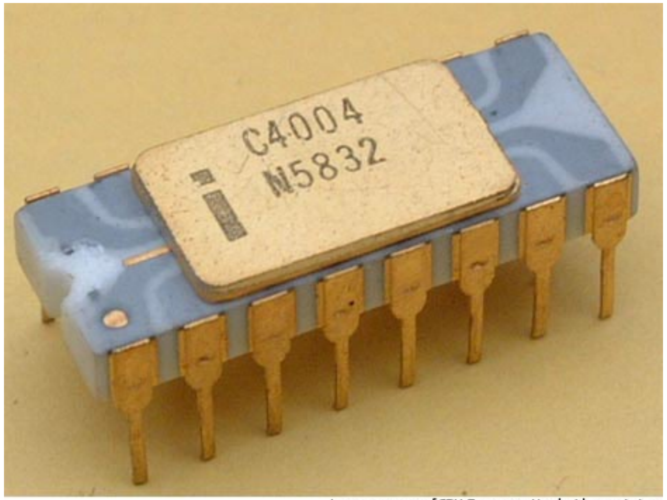
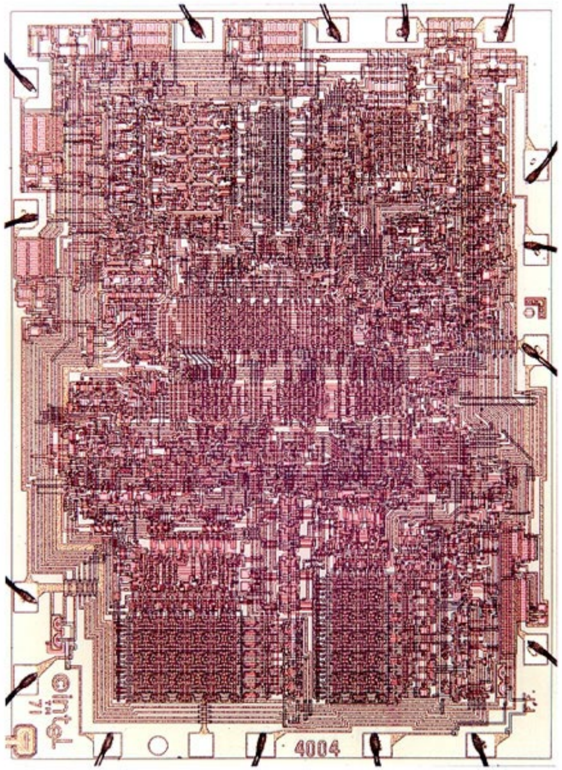


Image courtesy of CPU-Zone.com. Used with permission.



First Processors

- Intel released the 8086, a 16-bit microprocessor, in 1978
- Motorola followed with the MC68000 as their 16-bit processor
  - The 16-bit processor works with 16 bit words rather than 8 bit words
  - Instructions executed faster
  - Provide single instructions for more complex instructions such as multiply and divide
- 16-bit processors evolved into 32-bit processors
- Intel released the 80386
- Motorola released the MC 68020

Evolution of CPUs

- Intel®Core™i7
  - Intel®Core™i7-5960X Processor Extreme Edition
  - 20M Cache, up to 3.50 GHz
  - 8 Cores, 16 Threads
  - 64-bit Instruction Set

Microprocessor-based Systems

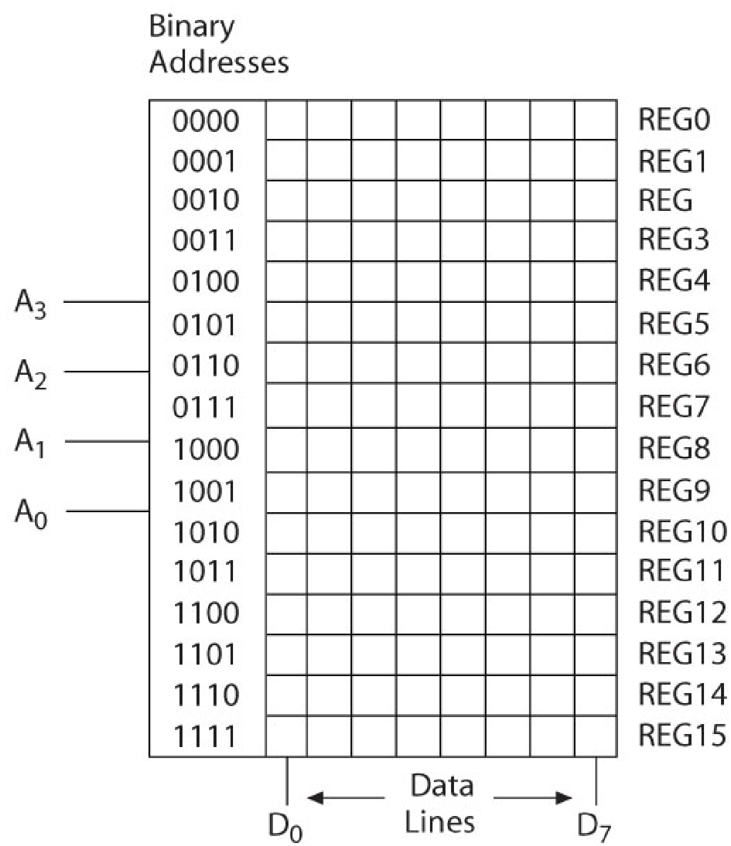
BUS

- **The three components – MPU, memory, and I/O, are connected by a BUS**

- **Address Bus**
  - Consists of 16, 20, 24, or 32 parallel lines (wires) – unidirectional
  - These lines contain the address of the memory location to read or written
- **Control Bus**
  - Consists of 4 to 10 (or more) parallel signal lines
  - CPU sends signals along these lines to memory and to I/O ports
    - Examples: Memory Read, Memory Write, I/O Read, I/O Write
- **Data Bus**
  - Consists of 8, 16, or 32 parallel lines
  - Bi-directional
  - Only one device at a time can have its outputs enabled
  - This requires the devices to have three-state output

Storing Bits in Memory

- We can store in different memory types
  - EEPROM, FLASH, RAM, etc.
- In an 8-bit RAM
  - Each byte is stored in a single memory register
  - Each word is stored in two memory locations (registers)
  - DATA 0x1234
    - 0x12 → REG11 (High-order byte)
      - 0001 0010
    - 0x34 → REG10 (Low-order byte)
      - 0011 0100



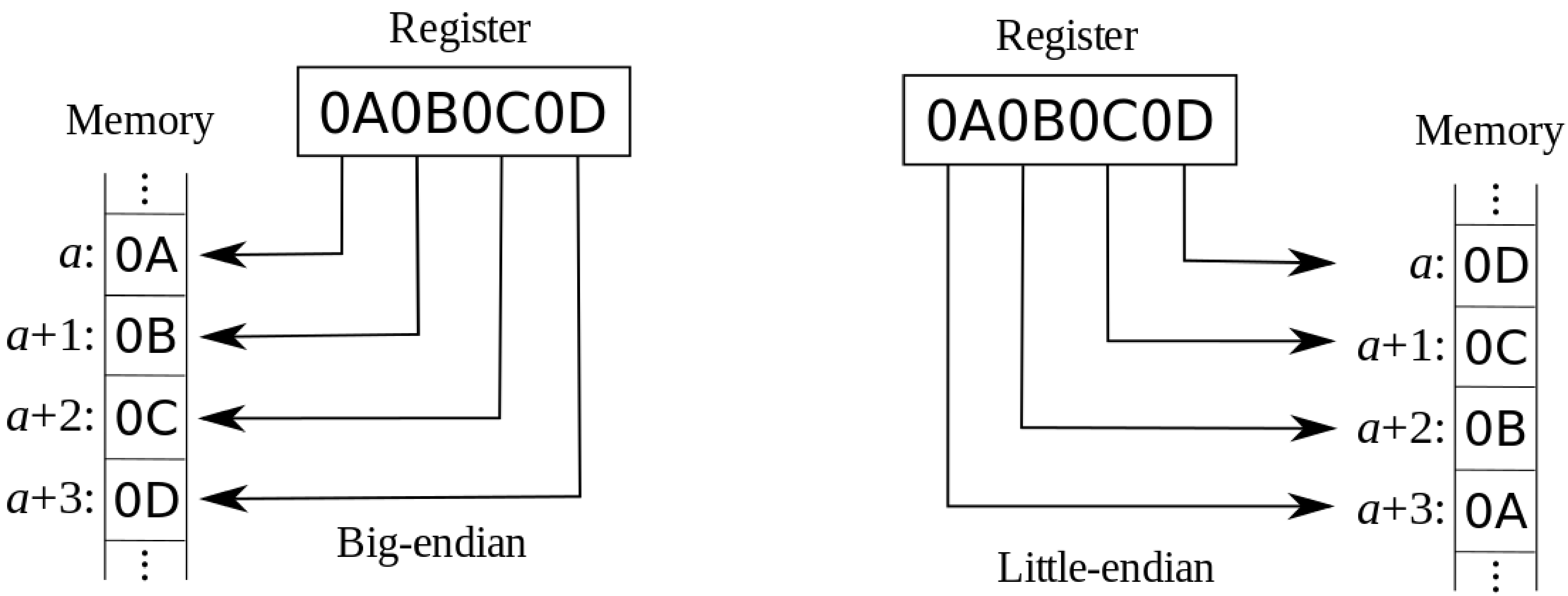
Remember: -8 -> 111 1000 (in two's complement)

MEMORY ADDRESSING

A unique ID (address) is assigned to each byte of memory (byte-addressable) or each word of memory (word-addressable)

Address	Content	Name	Type	Value
0x90000000	0x00			
0x90000001	0x00	anInt	int	0x000000FF(255)
0x90000002	0x00			
0x90000003	0xFF			
0x90000004	0xFF	aShort	short	0xFFFF(-1)
0x90000005	0xFF			
0x90000006	0x1F			
0x90000007	0xFF			
0x90000008	0xFF			
0x90000009	0xFF	aDouble	double	0x1FFFFFFFFFFFFF
0x9000000A	0xFF			(4.450147717014423E-308)
0x9000000B	0xFF			
0x9000000C	0xFF			
0x9000000D	0xFF			
0x9000000E	0x90			
0x9000000F	0x00	ptrAnInt	int*	0x90000000
0x90000010	0x00			
0x90000011	0x00			

ENDIANNESS



Register
0A0B0C0D
Memory
a: 0A
a+1: 0B
a+2: 0C
a+3: 0D
Big-endian
Register
0A0B0C0D
Memory
a: 0D
a+1: 0C
a+2: 0B
a+3: 0A
Little-endian

MEMORY

Main/Primary memory: Volatile (RAM), keeps programs when they are running and immediate data

Secondary Memory: Non-volatile (ROM), long-term storage for programs and data

Microprocessor-based Systems

Memory Types

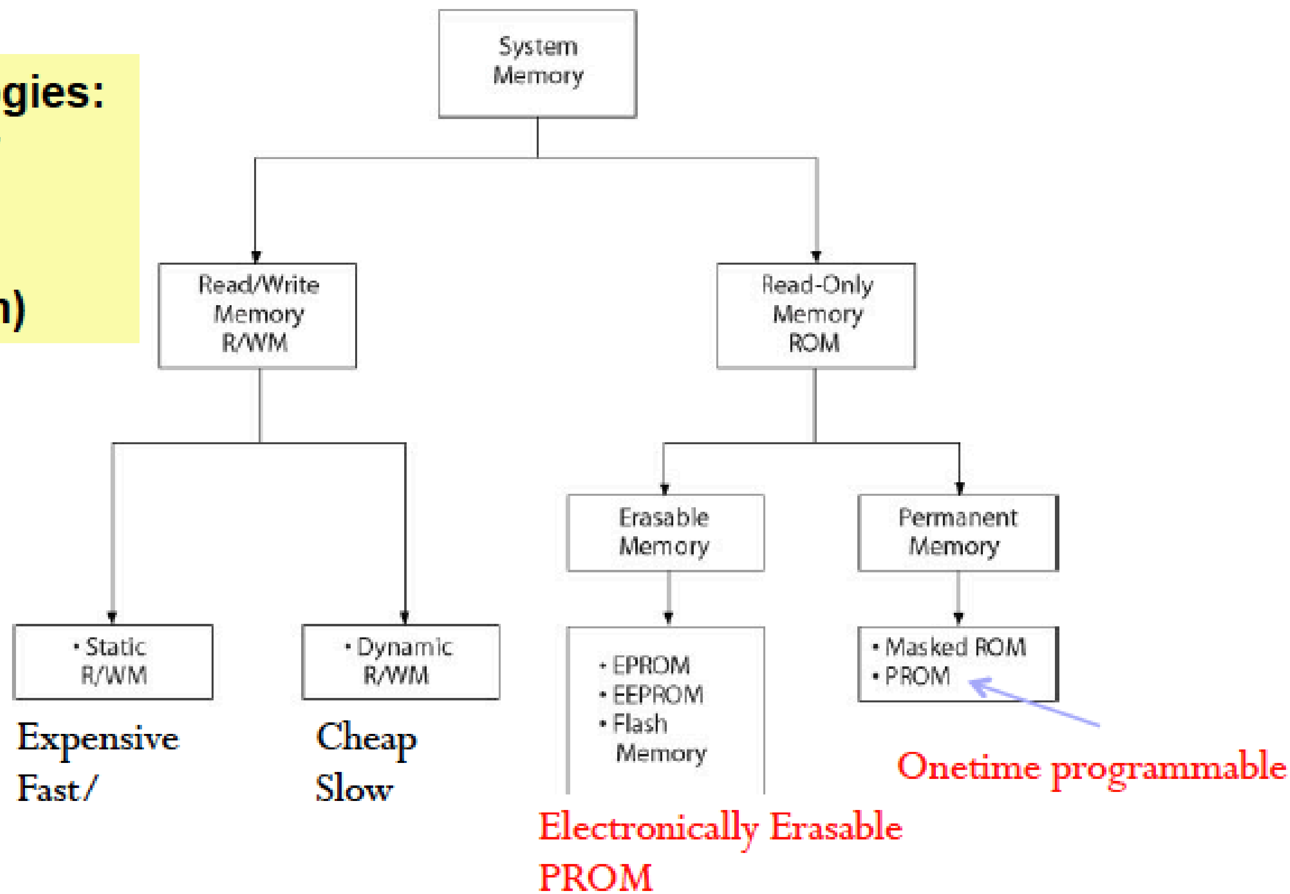
- **R/W: Read/Write Memory; also called RAM**
  - It is volatile (losses information as a power is removed)
  - Write means the processor can store information
  - Read means the processor can retrieve information from the memory
  - Acts like a Blackboard!

- **ROM: Read-Only Memory**
  - It is typically non-volatile (permanent) – can be erasable
  - It is similar to a page from your textbook

Microprocessor-Based Systems

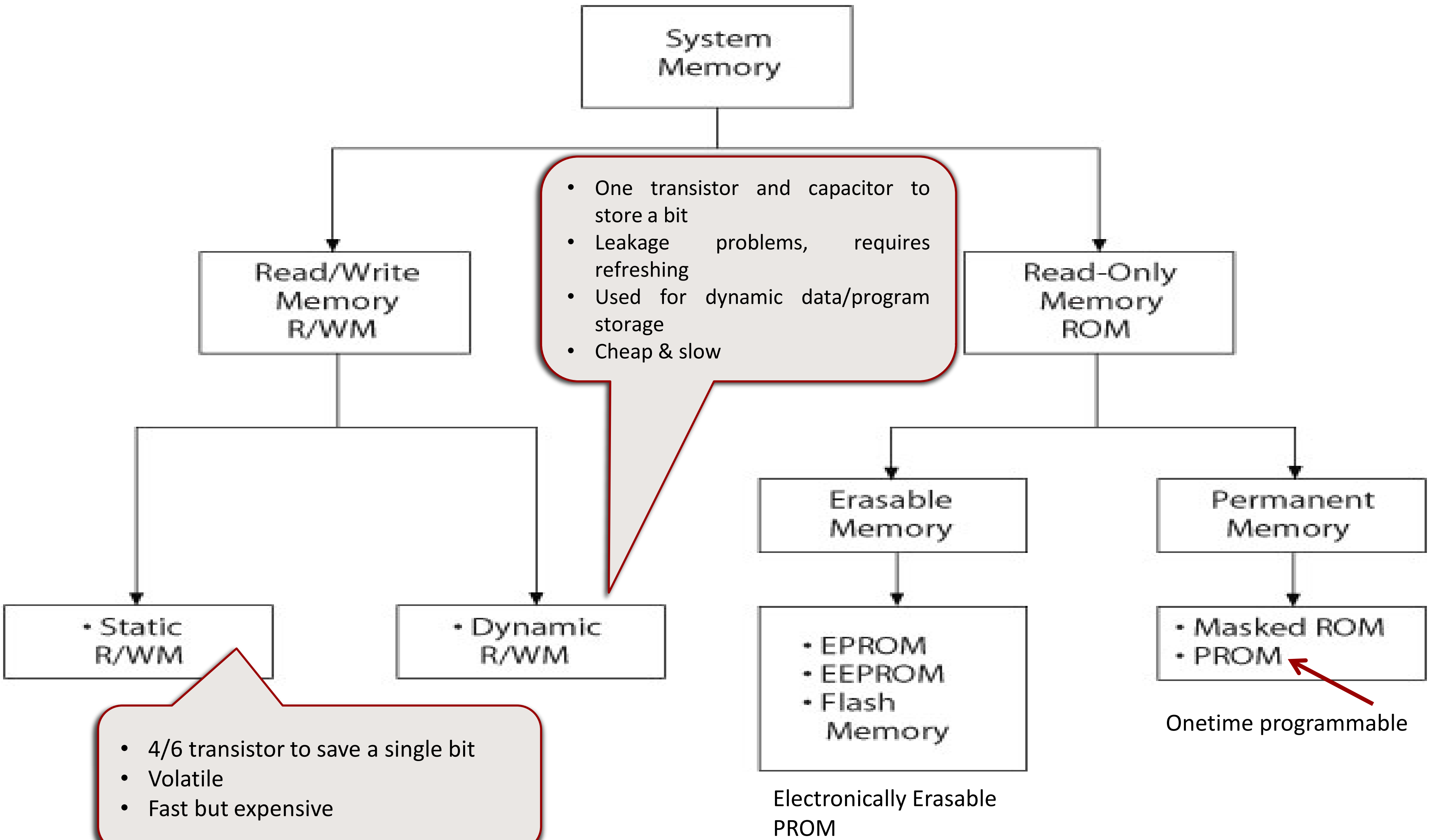
Memory Classification

Basic Technologies:
Semiconductor
Magnetic
Optical
(or combination)



Microprocessor-Based Systems

Memory Classification



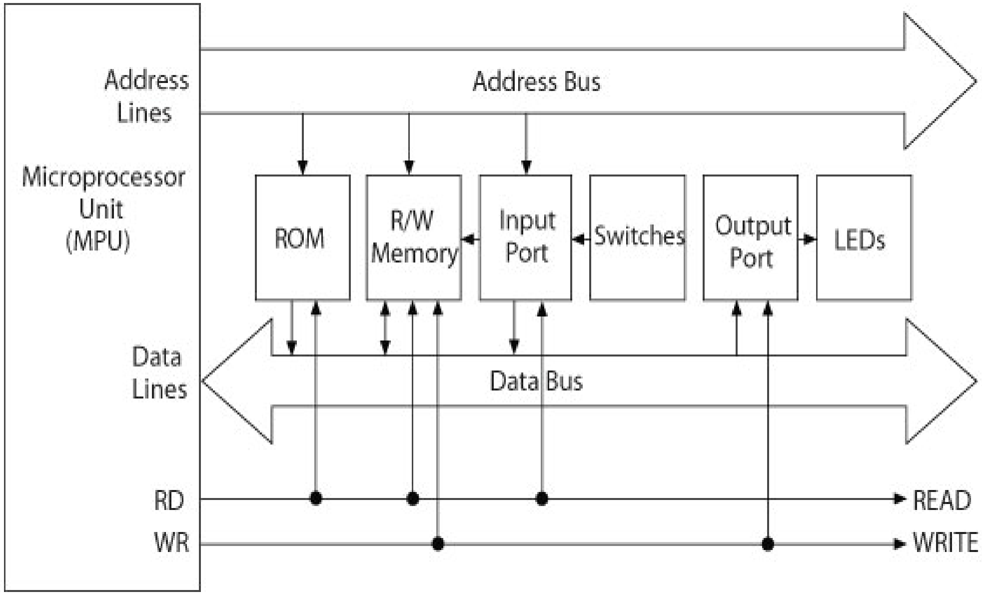
Erasable ROMs

- Masked Programmable ROM
    - Programmed by the manufacturer
- Programmable ROM (PROM)
    - Can be programmed in the field via the programmer
- Erasable Programmable ROM (EPOM)
    - Uses ultraviolet light to erase (through a quartz window)
    - OTP refers to One-Time Programmable
- Electrically Erasable Programmable ROM (EEPROM)
    - Each program location can be individually erased
    - Expensive
    - Requires programmer
- FLASH
    - Can be programmed in-circuit (in-system)
    - Easy to erase (no programmer)
    - Only one section can be erased/written at a time (typically 64 bytes at a time)

Microprocessor-based Systems

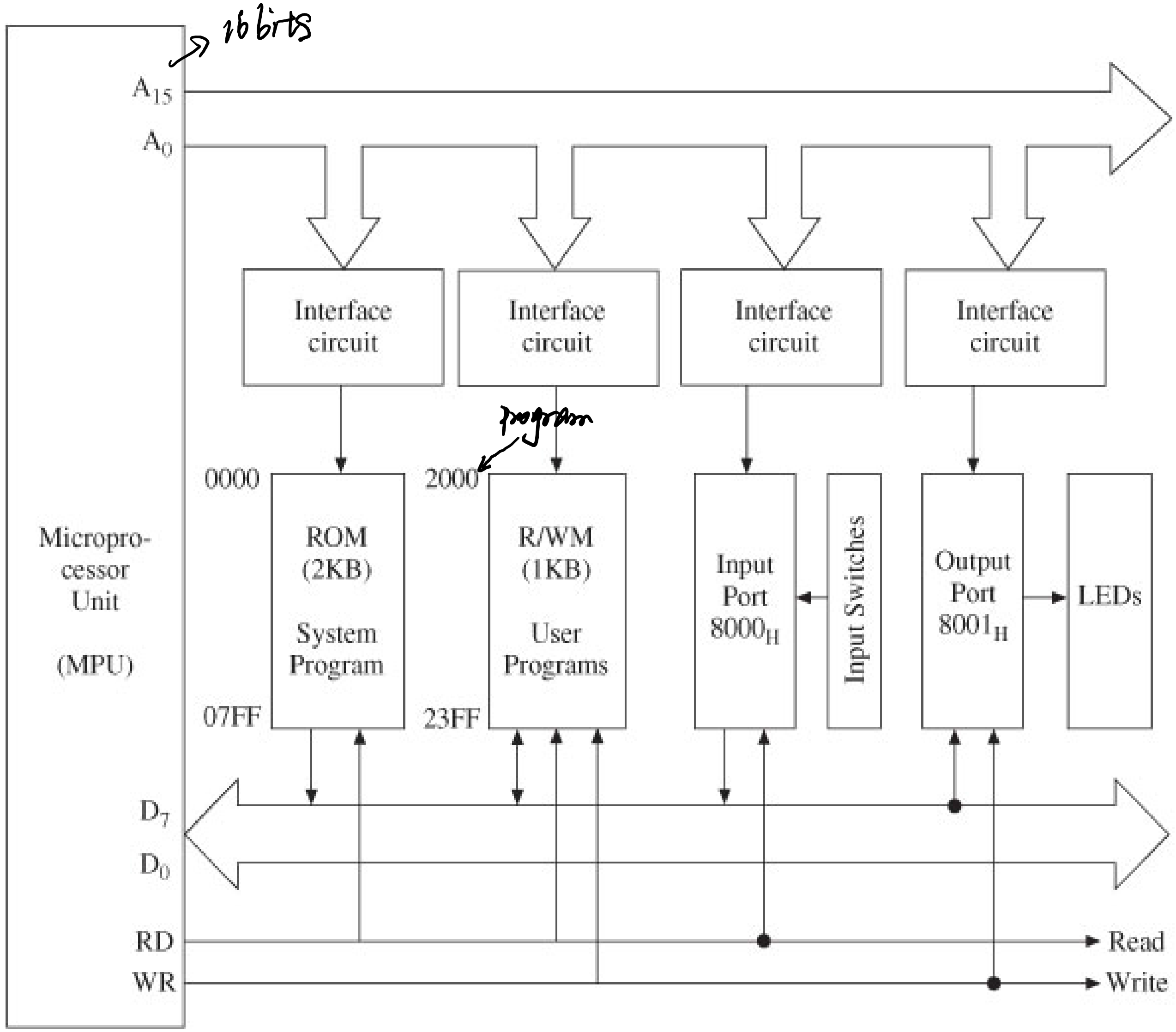
I/O Ports

- The way the computer communicates with the outside world devices
- I/O ports are connected to Peripherals
  - Peripherals are I/O devices
    - Input devices
    - Output devices
  - Examples
    - Printers and modems
    - Keyboard and mouse
    - Scanner
    - Universal Serial Bus (USB)



Expanded Microprocessor-Based System

- Note the direction of the busses.
- What is the width of the address bus?
- What is the value of the Address bus to access the first register of the R/WM?



Remember: 111 1111 1111 = 2^11=2K

First Microcontrollers

- IBM started using Intel processors in its PC
  - Intel started its 8042 and 8048 (8-bit microcontroller) – using in printers
- Apple Macintosh used Motorola 68000
- In 1980 Intel abandoned microcontroller business
- By 1989, Microchip was a major player in designing microcontrollers
  - PIC: Peripheral Interface Controller

Embedded Controllers

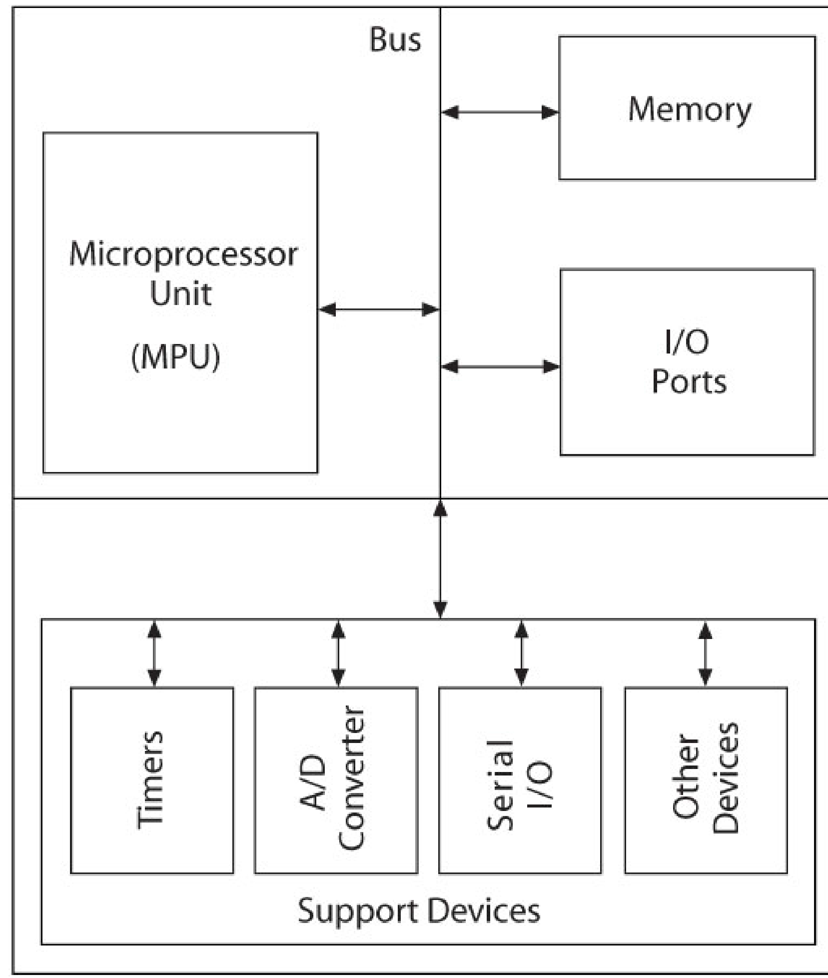
Software Characteristics

- No operating systems
- Execute a single program, tailored exactly to the controller hardware
- Assembly language (vs. High-level language)
  - Not transportable, machine specific
  - Programmer needs to know CPU architecture
  - Speed
  - Program size
  - Uniqueness

Microcontroller Unit (MCU)

Block Diagram

- An integrated electronic computing and logic device that includes three major components on a single chip
  - Microprocessor
  - Memory
  - I/O ports
- Includes support devices
  - Timers
  - A/D converter
  - Serial I/O
  - Parallel Slave Port
- All components connected by common communication lines called the system bus



MCU Architecture

- RISC (Harvard)
  - Reduced instruction set computer
  - Simple operations
  - Simple addressing modes
  - Longer compiled program but faster to execute
  - Uses pipelining
- CISC (Von Neuman)
  - Complex instruction set computer
  - More complex instructions (closer to high-level language support)

Bench marks: How to compare MCUs together
MIPS: Million Instructions / second (Useful when the compilers are the same)

Main 8-bit Controllers

- Microchip -PIC® Microcontrollers
  - RISC architecture (reduced instruction set computer)
  - Has sold over 2 billion as of 2002
  - Cost effective and rich in peripherals
- Motorola – now Freescale
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 68HC05, 68HC08, 68HC11
- Intel – now Marvell
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 8051, 8052
  - Many different manufacturerers: Philips, Dallas/MAXIM Semiconductor, etc.
- Atmel
  - RISC architecture (reduced instruction set computer)
  - Cost effective and rich in peripherals
  - AVR

Software: From Machine to High-Level Languages

(1 of 3)

- Machine Language: binary instructions
  - All programs are converted into the machine language of a processor for execution
  - Difficult to decipher and write
  - Prone to cause many errors in writing

High-Level Language
Assembly Language
Machine Language

Software: From Machine to High-Level Languages

(2 of 3)

- Assembly Language: machine instructions represented in mnemonics
  - Has one-to-one correspondence with machine instructions
  - Efficient in execution and use of memory; machine-specific and not easy to troubleshoot

High-Level Language
Assembly Language
Machine Language

Software: From Machine to High-Level Languages

(3 of 3)

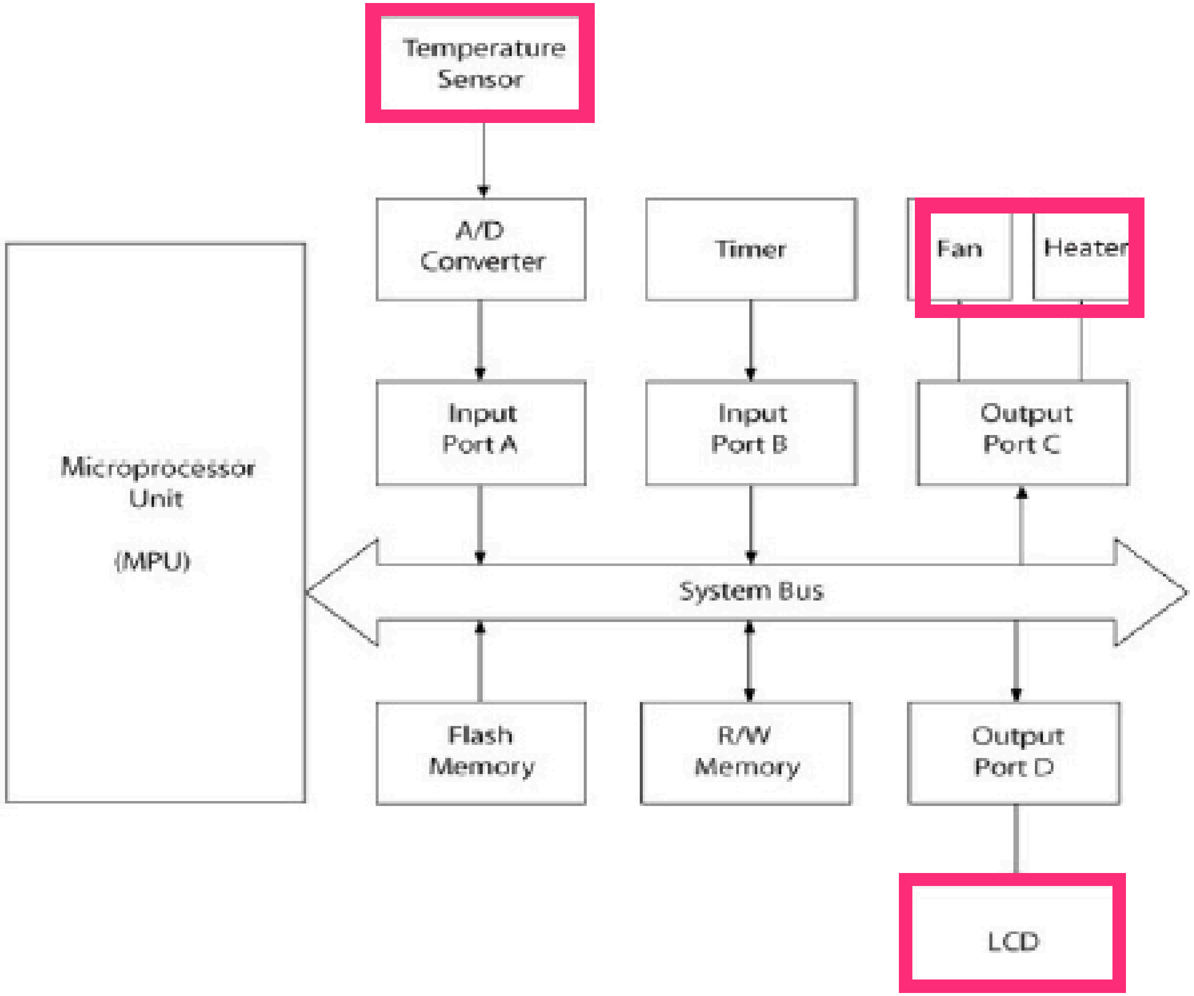
- High-Level Languages: Such as BASIC, C, and C++
  - Written in statements of spoken languages (such as English)
    - Machine independent
    - Easy to write and troubleshoot
    - Requires large memory and less efficient in execution

High-Level Language
Assembly Language
Machine Language

Design Examples …..

Microcontrollers vs. Microprocessors

MPU-Based Time and Temperature System



MCU-Based Time and Temperature System