Fundamentals of Microprocessor and Microcontroller

Dr. Farid Farahmand

Updated: 4/2/16

A little History

- *What is a computer?*
  - [Merriam-Webster Dictionary] one that computes; *specifically :* programmable electronic device that can store, retrieve, and process data.
  - [Wikipedia] A computer is a machine that manipulates data according to a list of instructions.
- Classification of Computers (power and price)
  - Personal computers
  - Mainframes
  - Supercomputers
  - Dedicated controllers – Embedded controllers

Mainframes

- The First Mainframes
  - Big businesses with big needs required big computers. Economies of scale also favored large, consolidated computer systems.
- The Second Mainframes
  - Transistor-based computers were replacing vacuum-tube machines in the late 1950s, spurred developments in hardware and software. Manufacturers commonly built small numbers of each model, targeting narrowly defined markets.

Mainframes

- Massive amounts of memory
- Use large data words…64 bits or greater
- Mostly used for military defense and large business data processing
- Examples: IBM 4381, Honeywell DPS8

Personal Computers

- Any general-purpose computer
  - Intended to be operated directly by an end user
- Range from small microcomputers that work with 4-bit words to PCs working with 32-bit words or more
- They contain a **Processor** - called different names
  - Microprocessor – built using Very-Large-Scale Integration technology; the entire circuit is on a single chip
  - Central Processing Unit (CPU)
  - Microprocessor Unit (MPU) – similar to CPU

http://en.wikipedia.org/wiki/Personal_computer

Supercomputers

- Fastest and most powerful mainframes
  - Contain multiple central processors (CPU)
  - Used for scientific applications, and number crunching
  - Now have teraflops performance
    - ***FL**oating Point **O**perations **P**er **S**econd* (FLOPS)
    - Used to measure the speed of the computer
- Examples of special-purpose supercomputers:
  - Belle, Deep Blue, and Hydra, for playing chess
  - Reconfigurable computing machines or parts of machines
  - GRAPE, for astrophysics and molecular dynamics
  - Deep Crack, for breaking the DES cipher
  - MDGRAPE-3, for protein structure computation

http://en.wikipedia.org/wiki/Supercomputer

Microcontrollers – Embedded Systems

- An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions often with real-time
- An integrated device which consists of multiple devices
  - Microprocessor (MPU)
  - Memory
  - I/O (Input/Output) ports
- Often has its own dedicated software

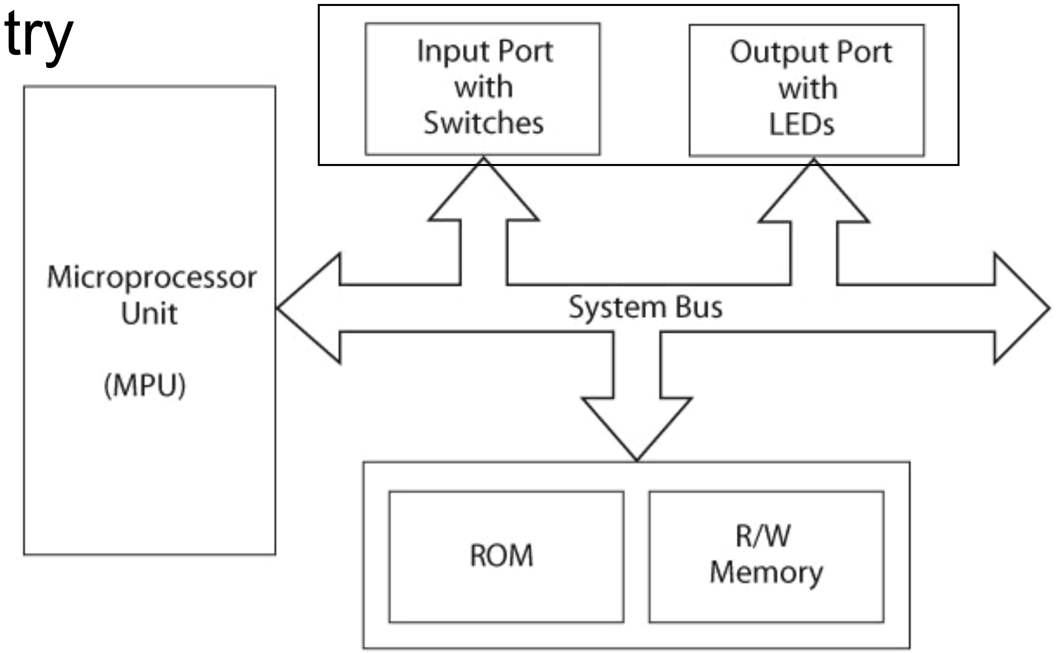
A little about Microprocessor-based Systems ……

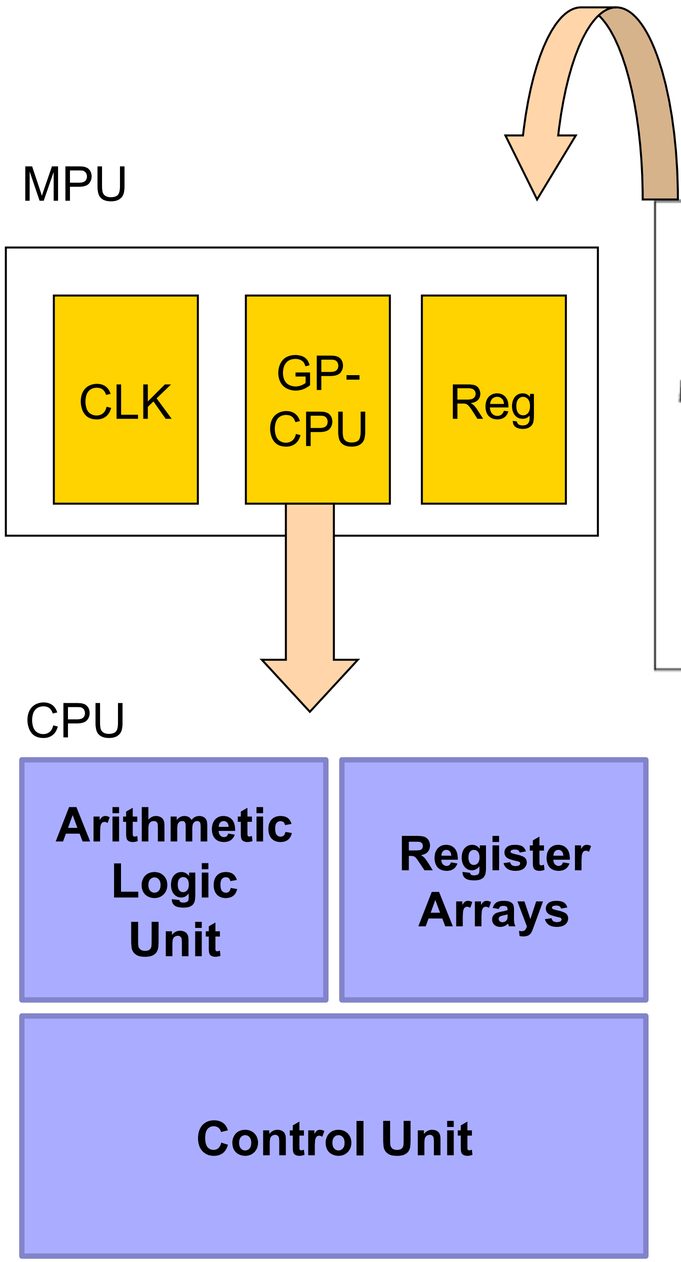
Evolution

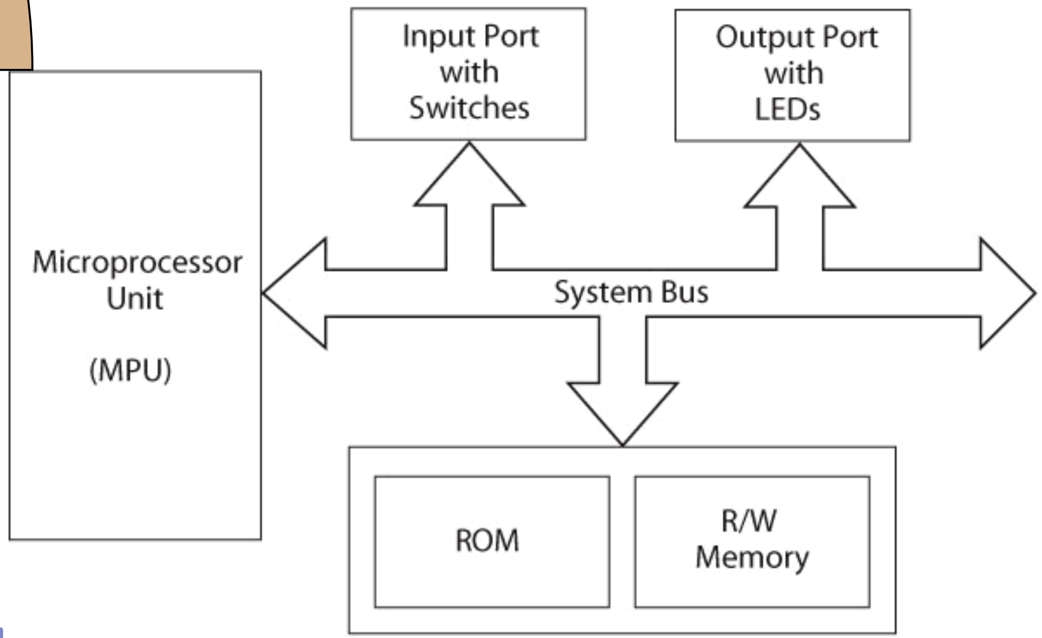
- First came transistors
- Integrated circuits
  - SSI (Small-Scale Integration) to ULSI
  - Very Large Scale Integration circuits (VLSI)
- 1- Microprocessors (MPU)
  - Microcomputers (with CPU being a microprocessor)
  - Components: Memory, CPU, Peripherals (I/O)
  - Example: Personal computers
- 2- 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 Systems

- Central Processing Unit (CPU)
- Memory
- Input/Output (I/O) circuitry
- Buses
  - Address bus
  - Data bus
  - Control bus

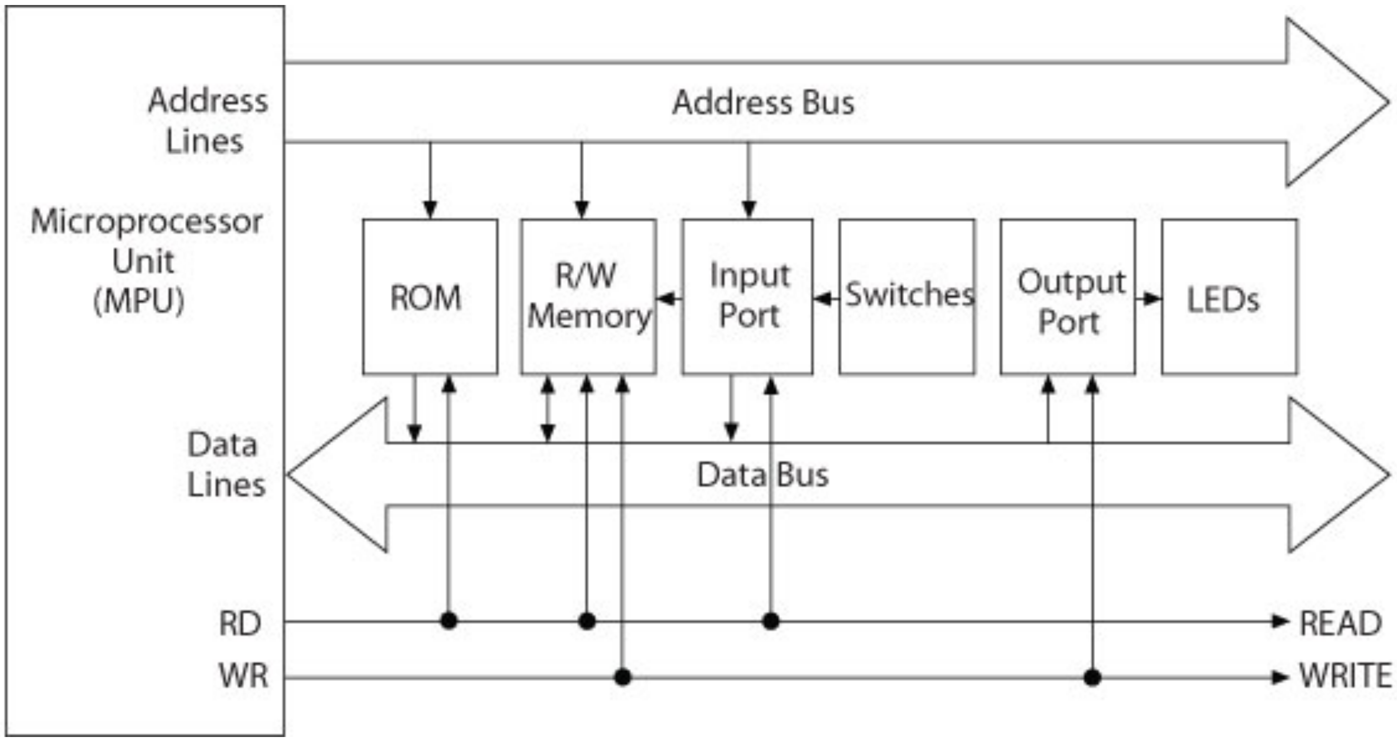






Microprocessor-based System

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

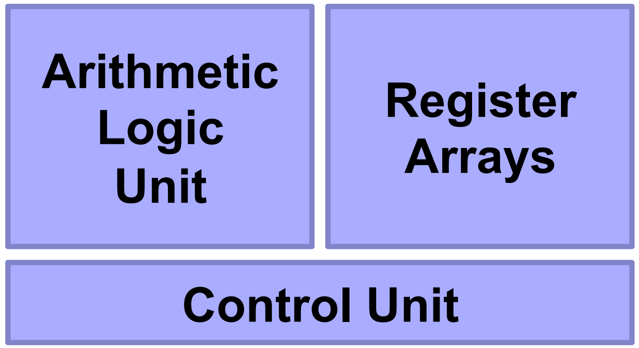


Microprocessor-based Systems
Microprocessor

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

Microprocessor-based Systems
Microprocessor

- the "brains" of the computer
  - its job is to fetch instructions, decode them, and then execute them
  - 8/16/32/etc –bit (how it moves the data
- contains:



ALU performs computing tasks – manipulates the data/ performs numerical and logical computations
Registers are used for temp. storage
Control unit is used for timing and other controlling functions – contains a program counter (next instruction's address and status register)

System software: A group of programs that monitors the functions of the entire system

Let’s Review a Few Things First…

Unsigned
Signed

Data Format (8-bit) (1 of 4)

- Unsigned Integers: 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)

- Signed Integers: Seven bits (Bit0 to Bit6) represent the magnitude of a number.
  - The 8th bit (Bit7) represents the sign of a 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 complement

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 numbers
    - 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 7F)
  - Represents English alphabet, decimal digits from 0 to 9, symbols, and commands

Back to the Main Point…
Evolution of CPUs

Digital Logic

George Boole (1815-1864)

English mathematician George Boole laid the foundations for the logic system that now bears his name: Boolean logic. His system of logical operations based on simple principles is the bedrock of modern computers.

Putting Boolean Logic to Work

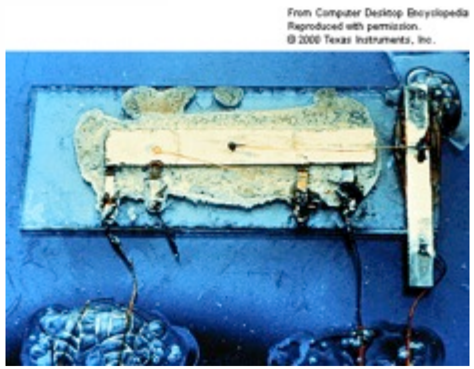
Claude Shannon encountered George Boole's ideas in a college philosophy class in the 1930s. He recognized its value for real world problems.

Shannon's 1937 MIT master's thesis, *A Symbolic Analysis of Relay and Switching Circuits*, applied Boolean algebra to the design of logic circuits using electromechanical relays. Shannon is also remembered for a seminal 1948 paper on information theory, *A Mathematical Theory of Communication*.

Claude Shannon wasn't the first to apply Boole's concepts. Victor Shestakov proposed similar ideas in 1935, but didn't publish until 1941—and then only in Russian.

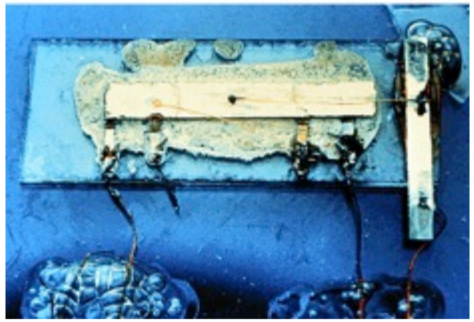
Transistors

- Vacuum Tubes: A devise to control modify, and amplify electric signals
- Then came transistors
  - Designed by John Bardeen, Will Shockley, and Walter Brattain, scientists at the Bell Telephone Laboratories in Murray Hill, New Jersey – 1947



Transistors

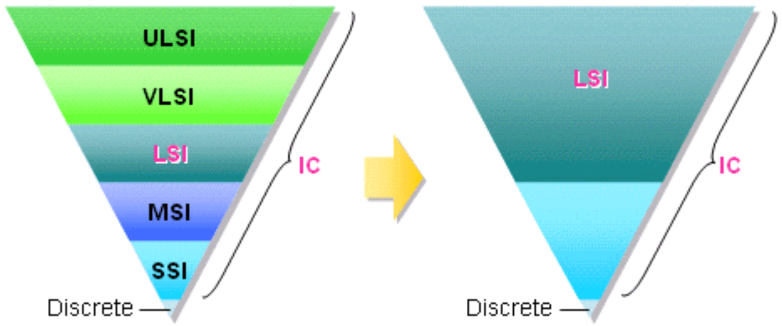
- In September 1958 Jack Kilby of Texas Instruments, Dallas, TX demonstrated the industry's first integrated flip-flop.
- TI announced Kilby’s germanium “Solid Circuit” concept
- Robert Noyce had co-founded the Fairchild Semiconductor Corporation – he was also working on *how to make more of less*.

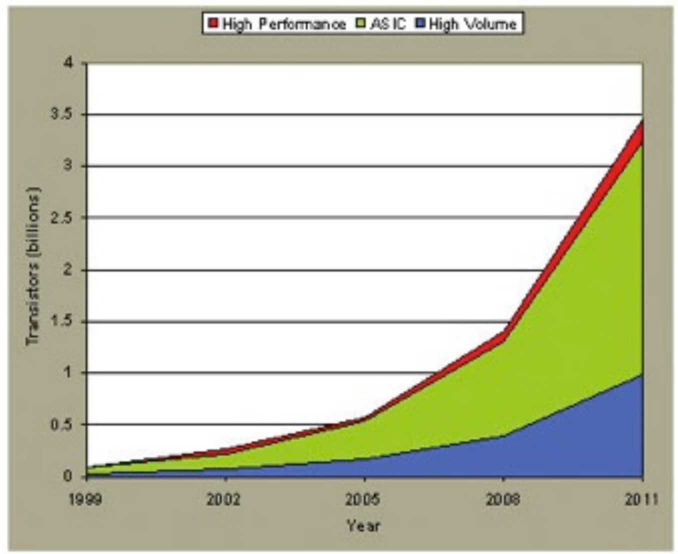


Jack Kilby was awarded the Nobel Prize in Physics in 2000 for his role as co-inventor with Robert Noyce of the integrated circuit. Noyce did not receive the award as he died on June 3, 1990.

Integrated Circuits

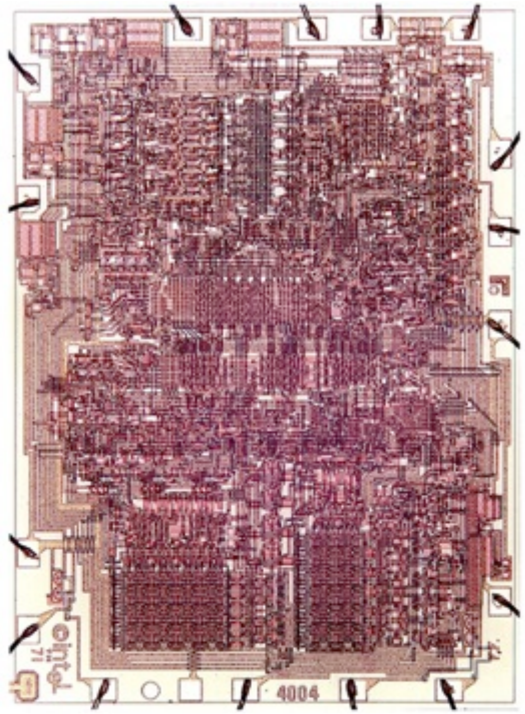
- Advances in manufacturing allowed packing more transistors on a single chip
- Transistors and Integrated Circuits from SSI (Small-Scale Integration) to ULSI
- Birth of a microprocessor and its revolutionary impact





Microprocessors

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



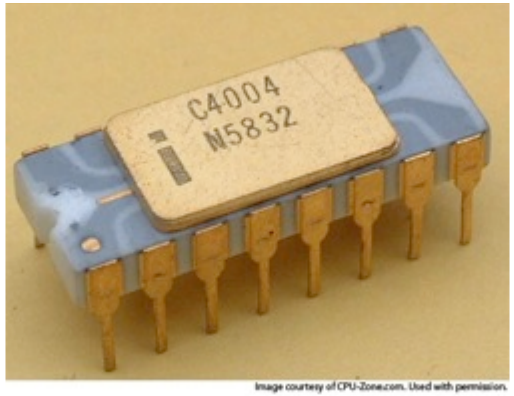
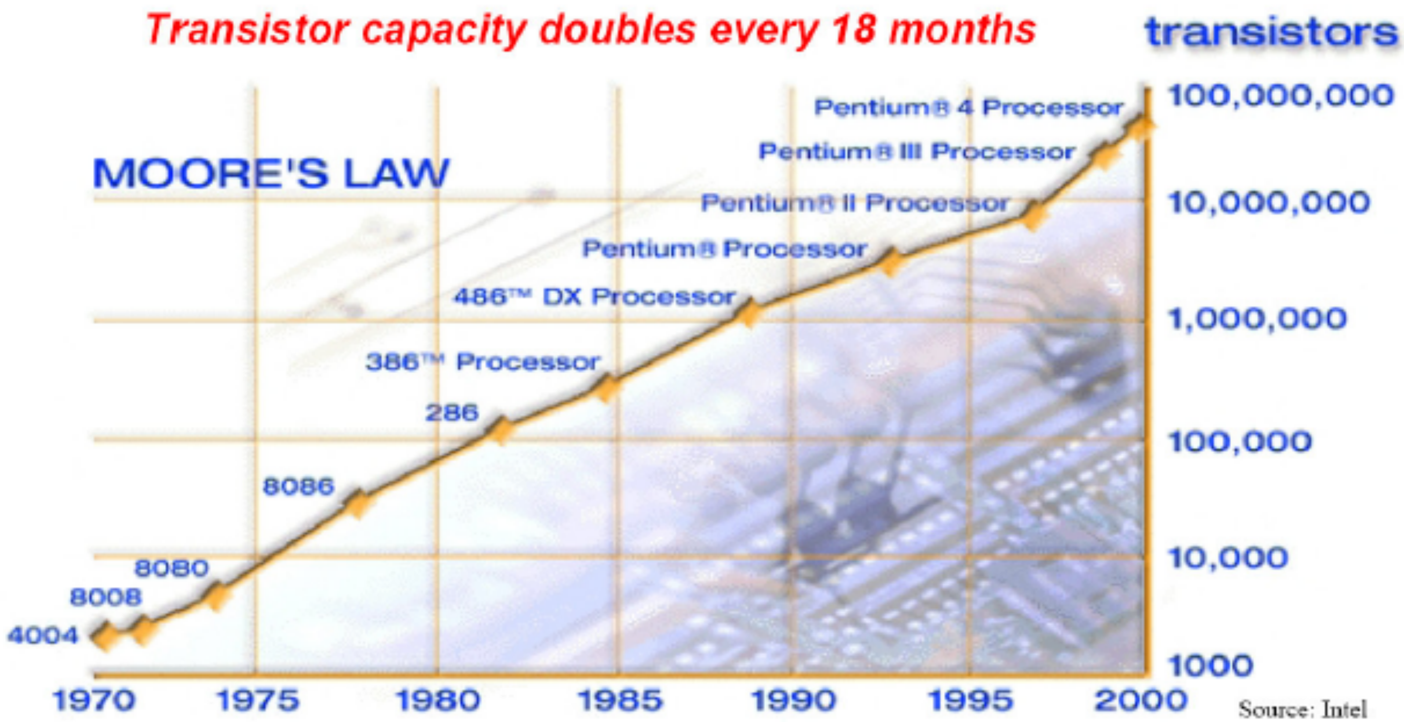


Image courtesy of CPU-Zone.com. Used with permission.

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 are 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 MC68020

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.

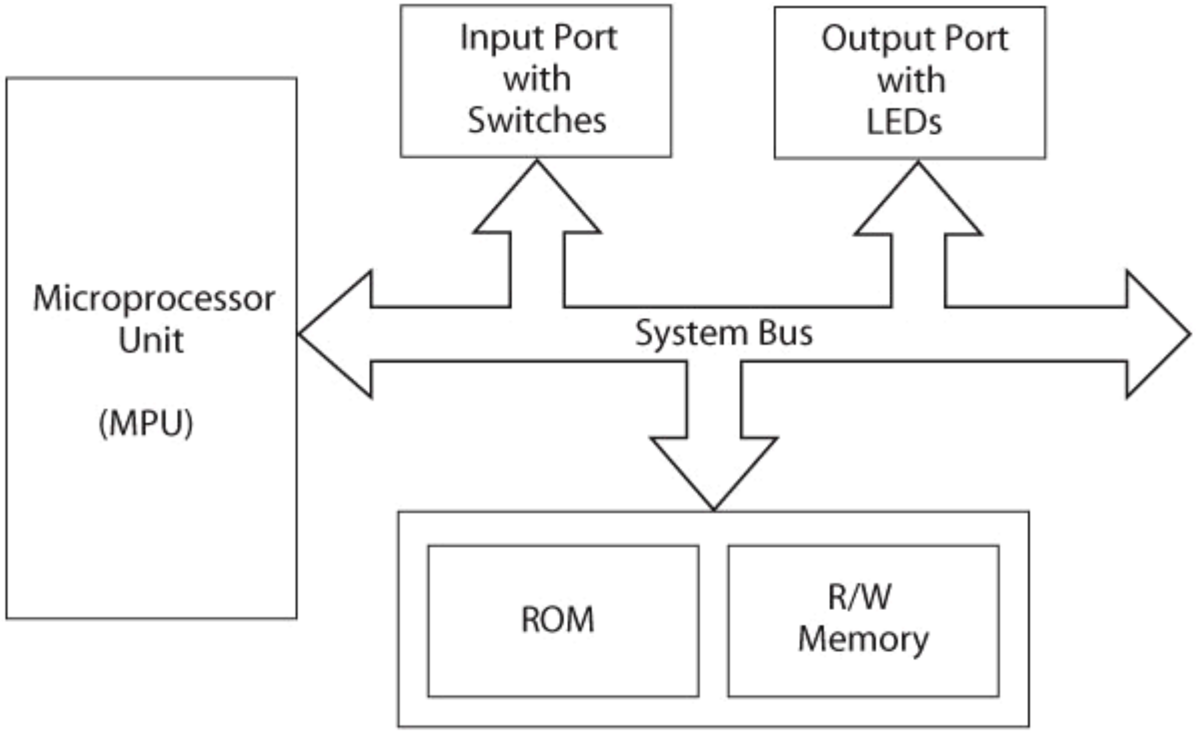
Evolution of CPUs

- Tukwila
  - World's First 2-Billion Transistor Microprocessor - Next-generation Intel® Itanium® processors (codenamed Tukwila)

2010: Intel Itanium Tukwila MPU, 2000M transistor
2010 AMD Operon 6100 MPU, 1800 transistors
2006 Nvidia G80, 681M transistors
1994 Motorola 68060 MPU, 2M transistors

http://www.intel.com/technology/architecture-silicon/2billion.htm

Remember



Input Port
with
Switches
Output Port
with
LEDs
Microprocessor
Unit
(MPU)
System Bus
ROM
R/W
Memory

Microprocessor-based Systems
Memory

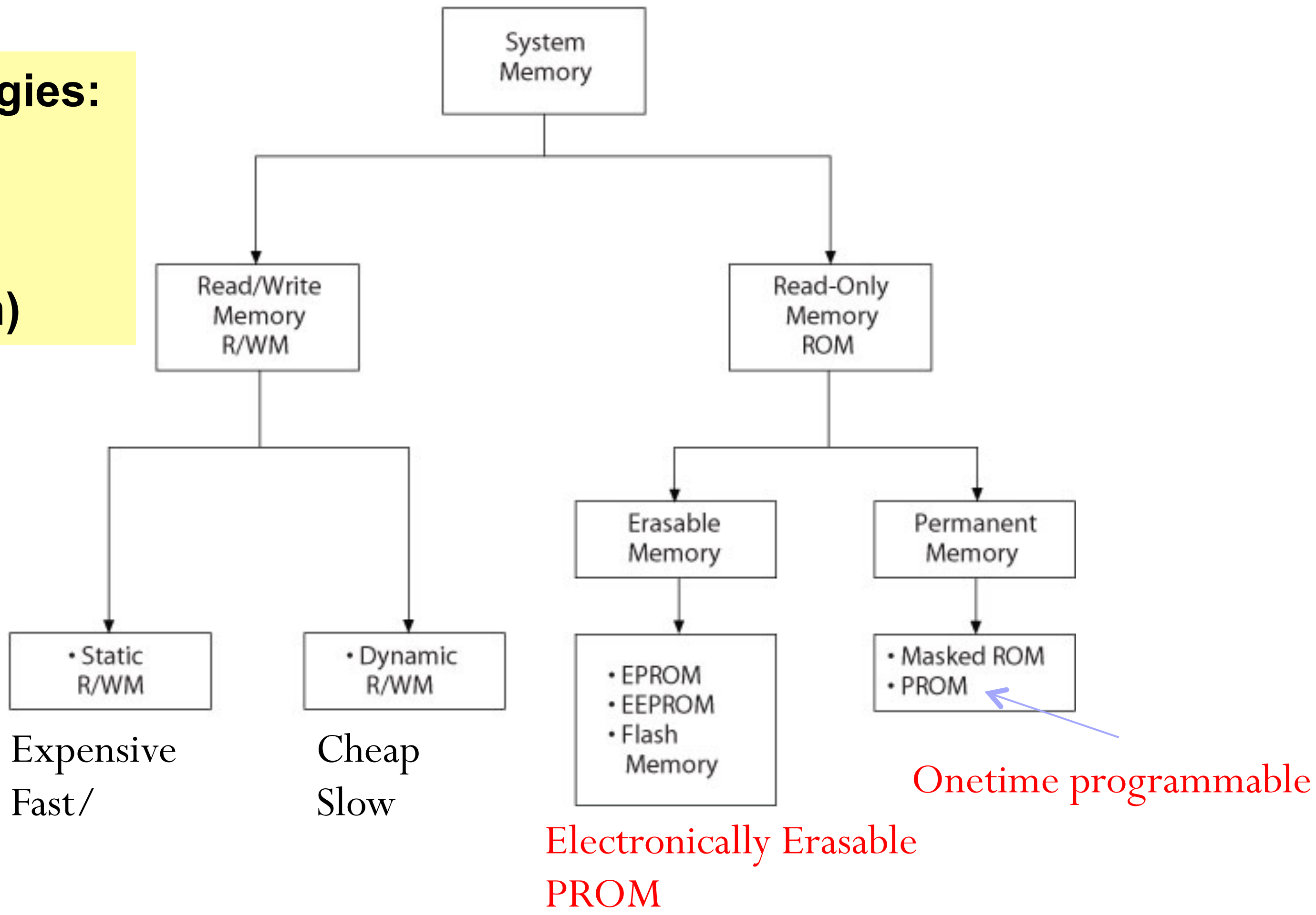
- Memory is a group of registers
  - 16 register – address: 0-15 – in binary: 0-1111; Address lines: A0-A3
- Serves two major purposes
  - storing the binary codes for the sequence of instructions specified by programs (program)
  - storing binary data that the computer needs to execute instructions (data)

Microprocessor-based Systems
Memory Types

- **R/W: Read/Write Memory; also called RAM**
  - It is volatile (losses information as power is removed)
  - Write means the processor can store information
  - Read means the processor can receive 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)

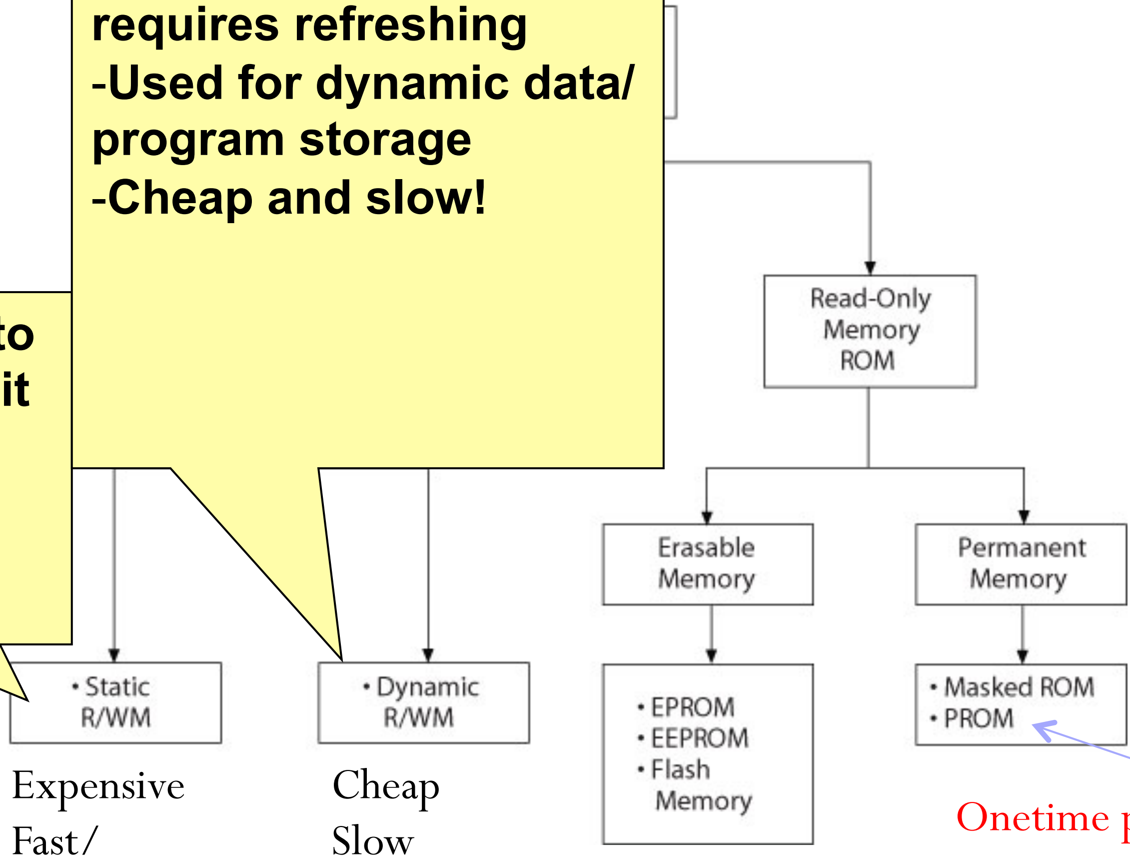


Microprocessor based Systems

Memory C

-one transistor and one capacitor to store a bit
-Leakage problem, thus requires refreshing
-Used for dynamic data/ program storage
-Cheap and slow!

-4/6 transistor to save a single bit
- Volatile
- Fast but expensive



Expensive Fast/

Cheap Slow

Electronically Erasable PROM

Onetime programmable

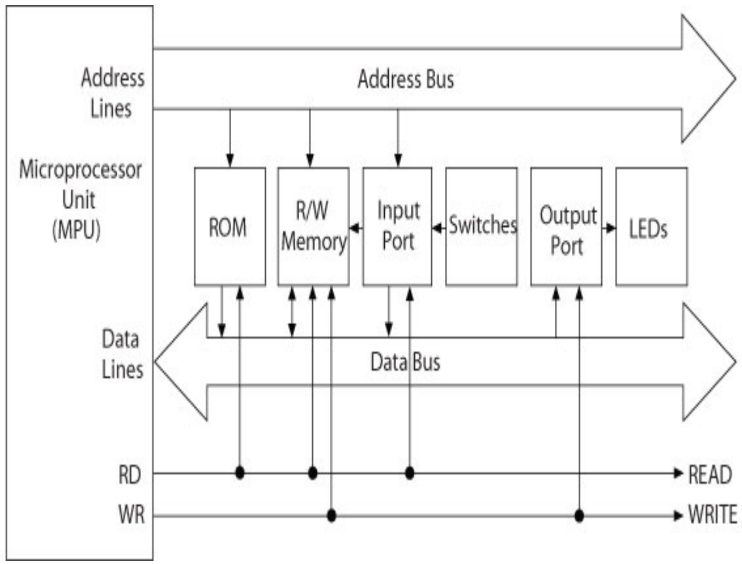
Erasable ROMs

- Marked Programmed ROM
  - Programmed by the manufacturer
- Programmable ROM (PROM)
  - Can be programmed in the field via the programmer
- Erasable Programmable ROM (EPROM)
  - 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)

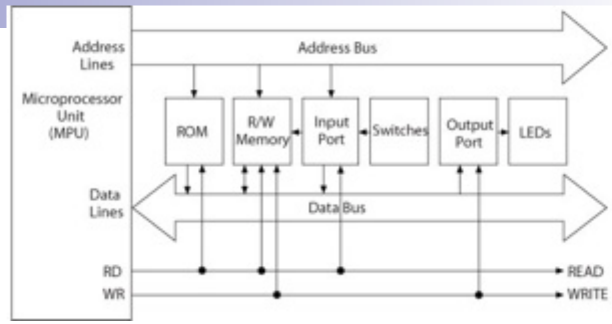
Microprocessor-based Systems

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)**



Microprocessor-based Systems - BUS



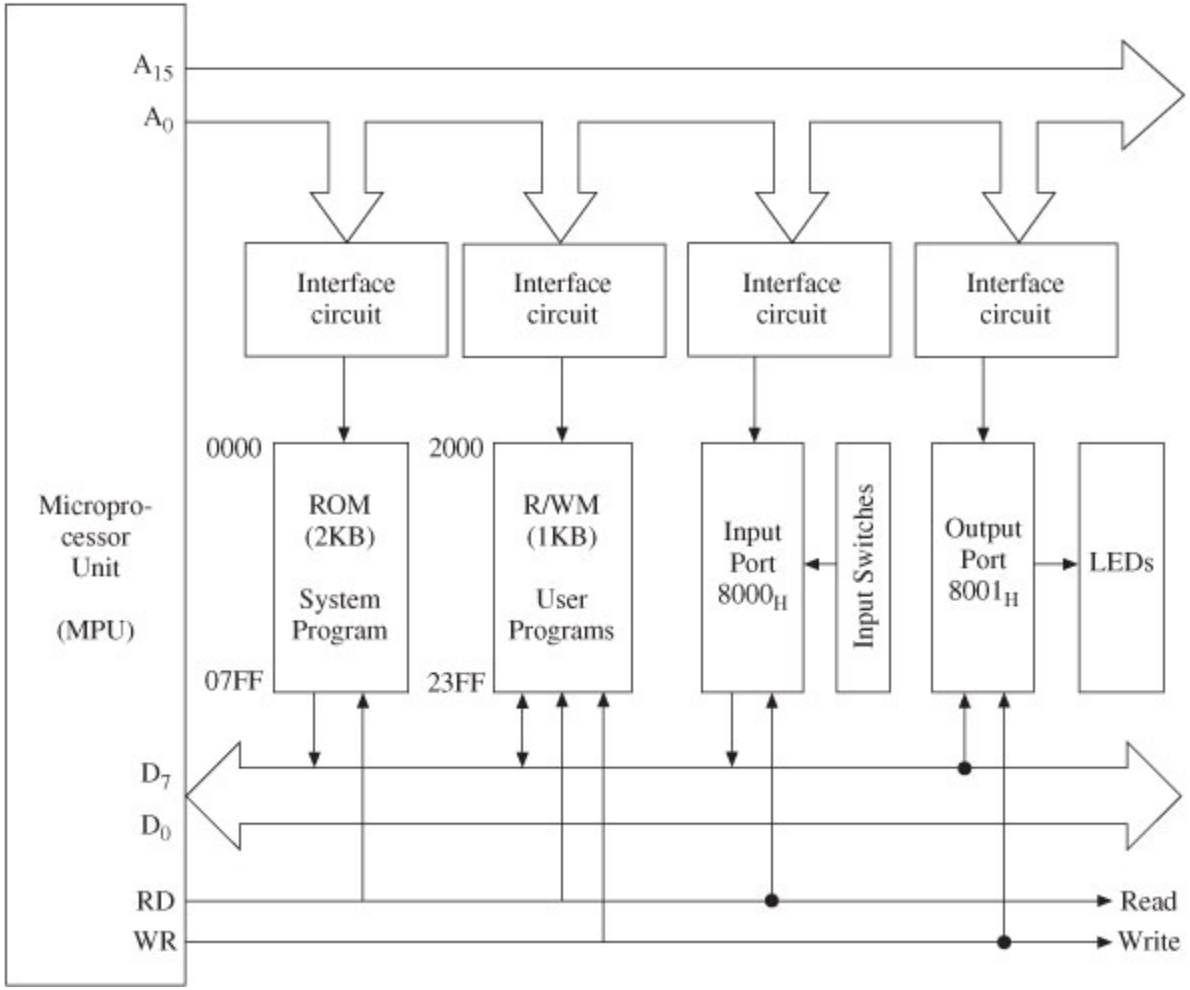
- **The three components – MPU, memory, and I/O – are connected by a group of wires called the BUS**

- **Address bus**
  - **consists of 16, 20, 24, or 32 parallel signal lines (wires) - <u>unidirectional</u>**
  - **these lines contain the address of the memory location to read or written**
- **Control bus**
  - **consists of 4 to 10 (or more) parallel <u>signal lines</u>**
  - **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 signal lines**
  - **<u>bi-directional</u>**
  - **only one device at a time can have its outputs enabled,**
  - **this requires the devices to have three-state output**

Remember: 111 1111 1111 = 7FF = 2^11-1 = 2047
2^11=2K=2048
2^11 Requires 11 bits

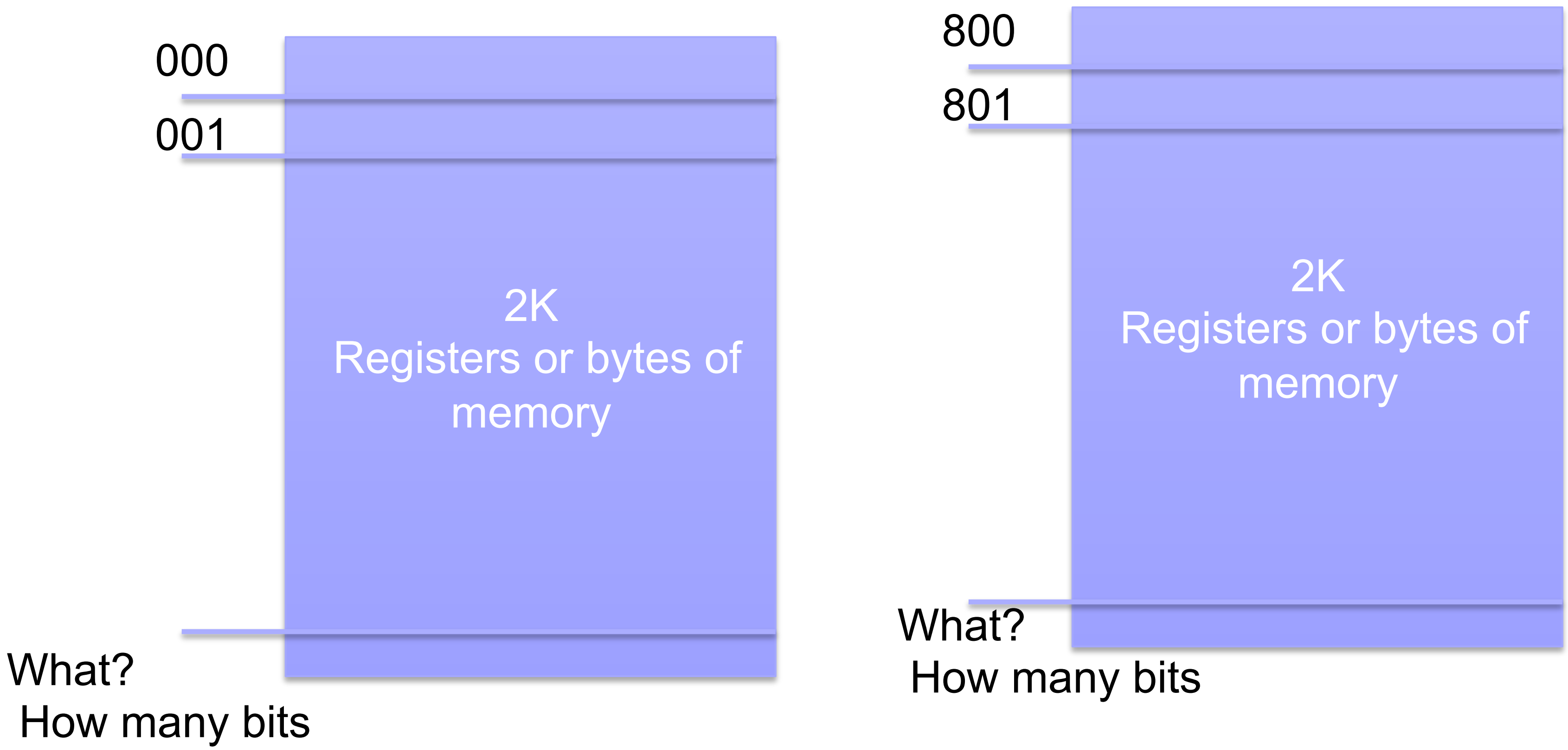
Expanded Microprocessor-Based System

1. **Note the directions of busses**
2. **What is the width of the address bus?**
3. **What is the value of the Address but to access the first register of the R/WM?**



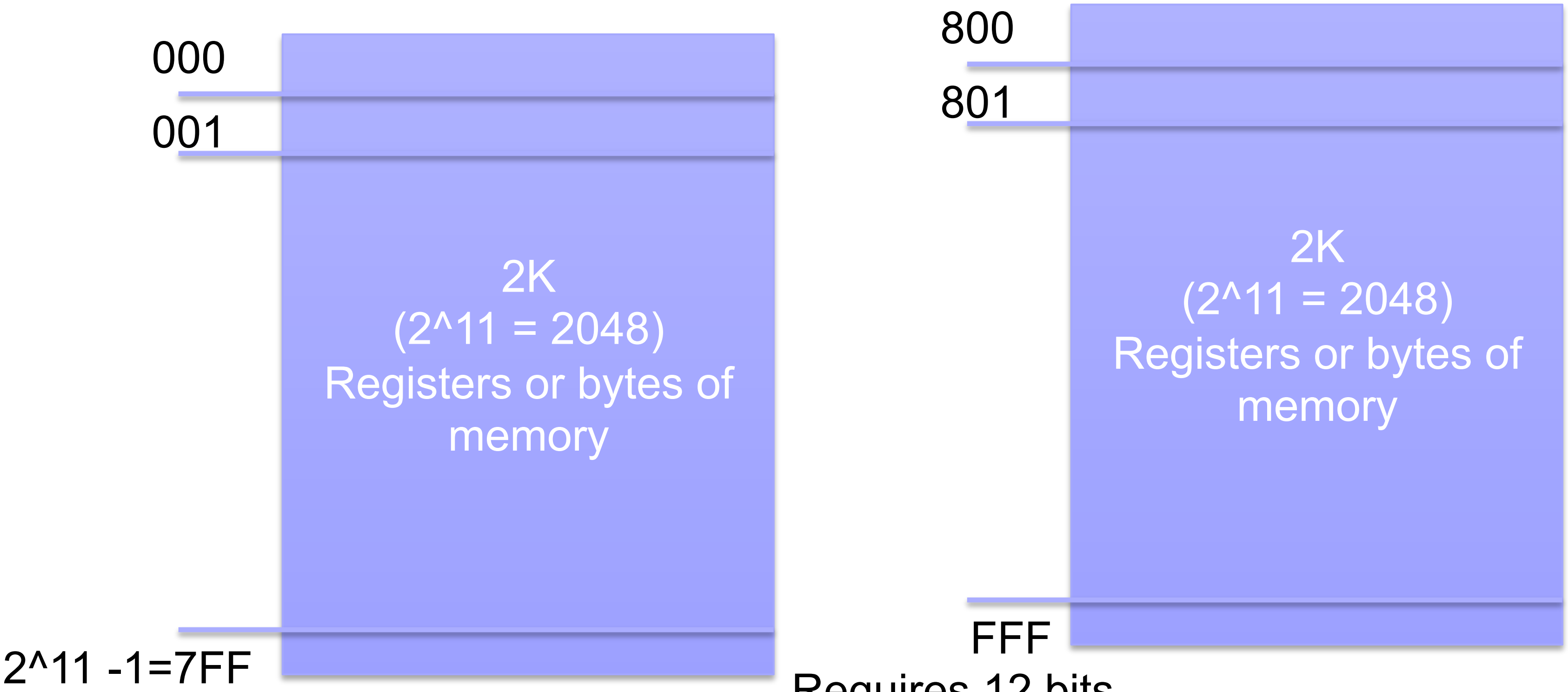
You must know how to draw it!

Example



How much memory do we have?

Example

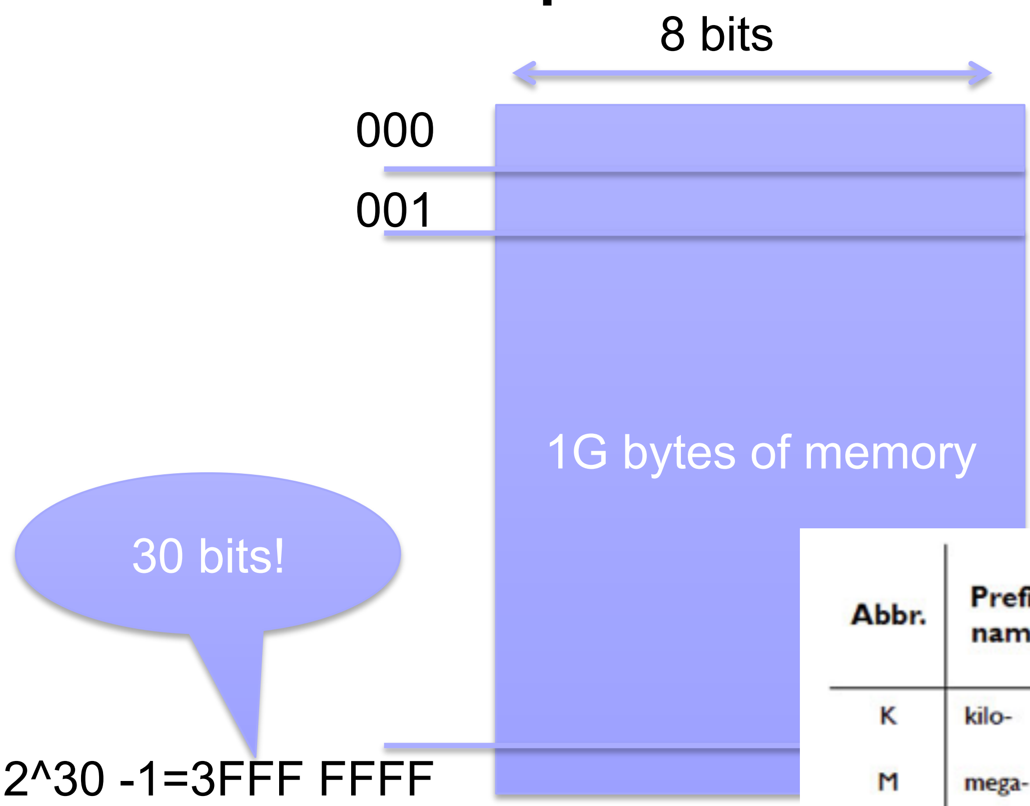


Requires 11 bits

Requires 12 bits
For a total of 4095 bytes

Total of 4K bytes of memory: 2^12 (FFF) → 12 bits ; last values 2^12-1 = 4096-1

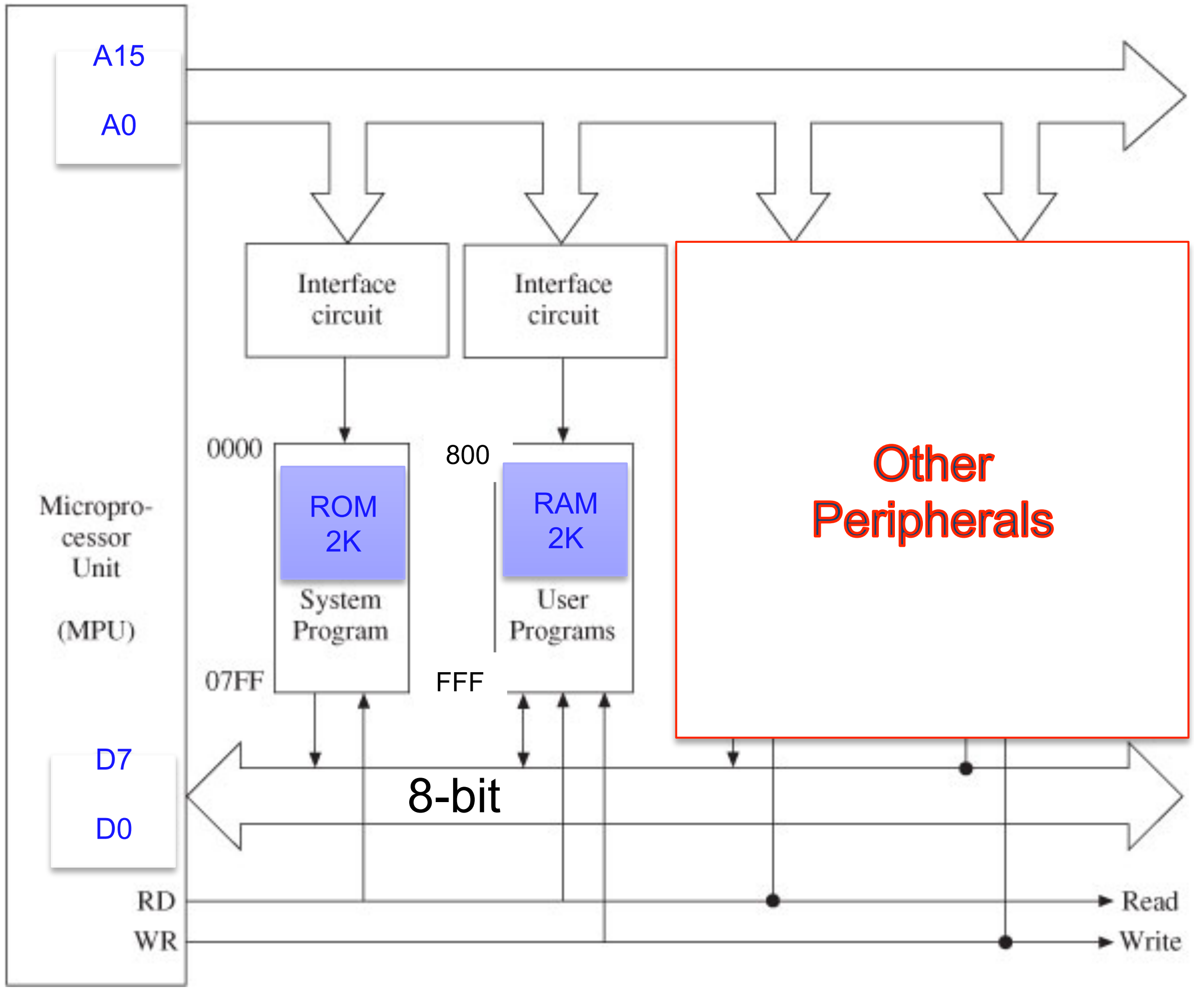
Example



Next number: 4000 0000 (in Hex)

Abbr.	Prefix name	Decimal size	Size in thousands	Binary approximation	Address variable size
K	kilo-	10^3	1,000	$1{,}024 = 2^{10}$	10
M	mega-	10^6	$1{,}000^2$	$1{,}024^2 = 2^{20}$	20
G	giga-	10^9	$1{,}000^3$	$1{,}024^3 = 2^{30}$	30
T	tera-	10^{12}	$1{,}000^4$	$1{,}024^4 = 2^{40}$	40
P	peta-	10^{15}	$1{,}000^5$	$1{,}024^5 = 2^{50}$	50
E	exa-	10^{18}	$1{,}000^6$	$1{,}024^6 = 2^{60}$	60

Example of an 8-bit MPU



So what are microcontrollers?

What is a Microcontroller?

- A microcontroller is a small computer on a single integrated circuit containing
  - processor core,
  - memory,
  - programmable input/output peripherals
- Used for specific (embedded) applications

Embedded controllers

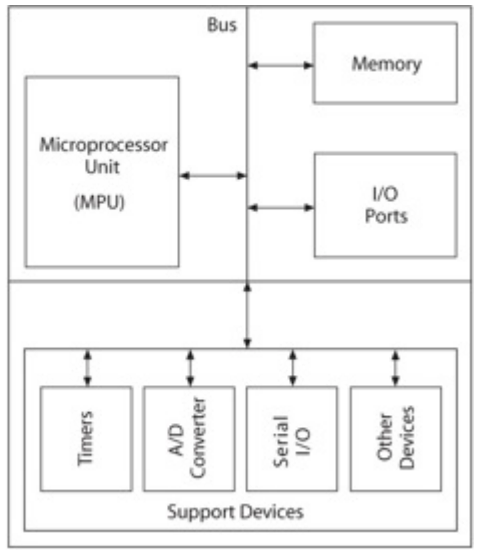
- Used to control smart machines
- Examples: printers, auto braking systems
- Also called microcontrollers or microcontroller units (MCU)

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 need to know CPU architecture
  - Speed
  - Program size
  - Uniqueness

Microcontroller Unit (MCU)
Block Diagram

- An integrated electronic computing and logic device that includes <u>three</u> major components on a <u>single chip</u>
  - 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.



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
- 1980 Intel abandoned microcontroller business
- By 1989 Microchip was a major player in designing microcontrollers
  - PIC: Peripheral Interface Controller

Different Microcontrollers (MCU)

- ARM core processors (from many vendors)
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Cypress Semiconductor PSoC (Programmable System-on-Chip)
- Freescale ColdFire (32-bit) and S08 (8-bit)
- Freescale 68HC11 (8-bit)
- Intel 8051
- Infineon: 8, 16, 32 Bit microcontrollers[9]
- MIPS
- Microchip Technology PIC, (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24), (32-bit PIC32)
- NXP Semiconductors LPC1000, LPC2000, LPC3000, LPC4000 (32-bit), LPC900, LPC700 (8-bit)
- Parallax Propeller
- PowerPC ISE
- Rabbit 2000 (8-bit)
- Renesas RX, V850, Hitachi H8, Hitachi SuperH (32-bit), M16C (16-bit), RL78, R8C, 78K0/78K0R (8-bit)
- Silicon Laboratories Pipelined 8051 Microcontrollers
- STMicroelectronics ST8 (8-bit), ST10 (16-bit) and STM32 (32-bit)
- Texas Instruments TI MSP430 (16-bit)
- Toshiba TLCS-870 (8-bit/16-bit).

What is the difference?
8/16/24/32 bits
Architecture
Package
Capability
Memory
Software (IDE)/cloud
ADC (10-12 bit)

http://en.wikipedia.org/wiki/Microcontroller A more complete list is here: http://en.wikipedia.org/wiki/Category:Microcontrollers

MCU Architecture

- RISC
  - Reduced instruction set computer
  - Simple operations
  - Simple addressing modes
  - Longer compiled program but faster to execute
  - Uses pipelining
  - Most embedded system
- CISC
  - Complex instruction set computer
  - More complex instructions (closer to high-level language support)
  - x86 standard (Intel, AMD, etc.), but even in the mainframe territory CISC is dominant via the IBM/390 chip

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

CISC vs RISC

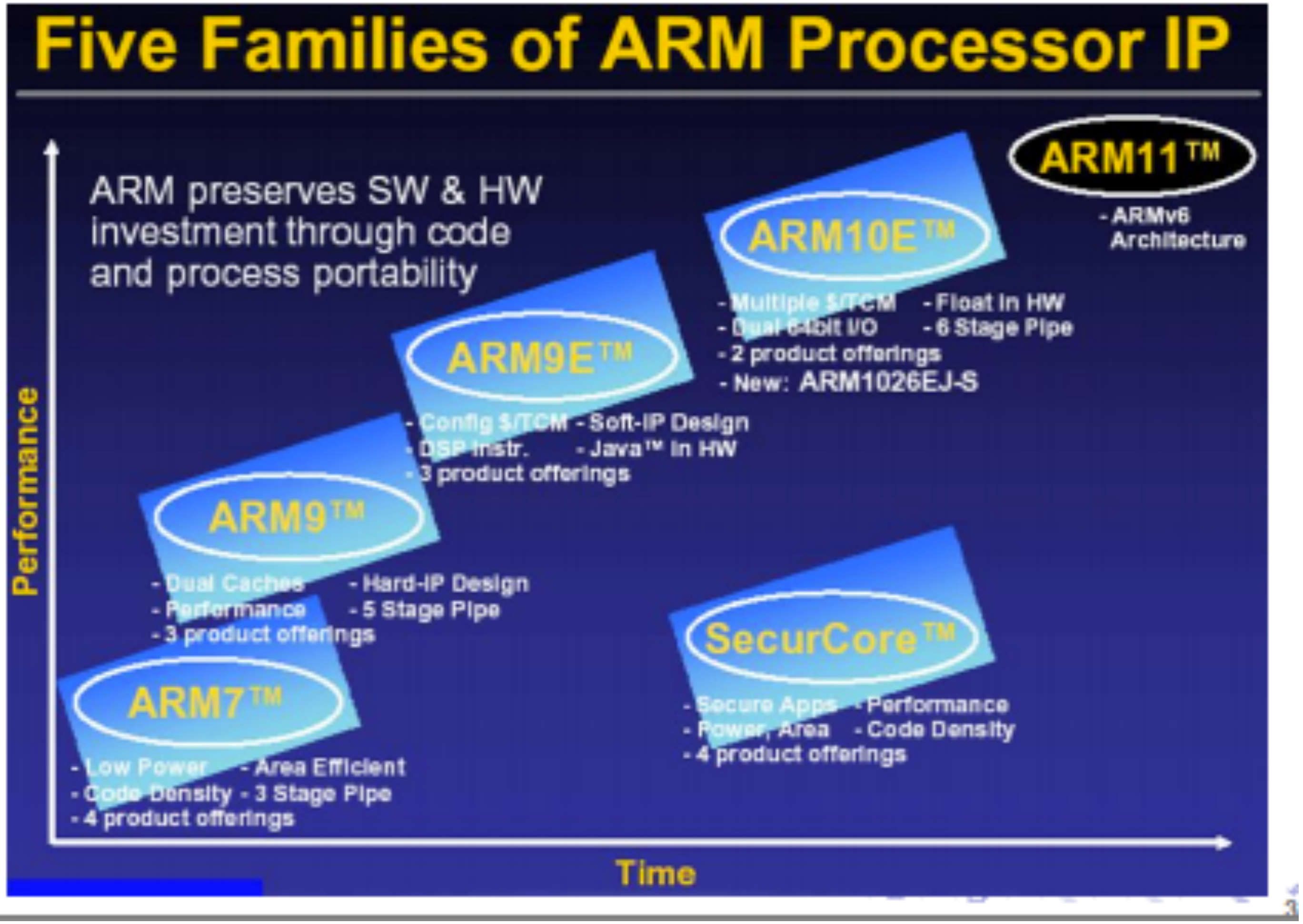
CISC **Pentium/x86 are CISC-based**	**RISC** **ARM-based** **Most mobile-phones**
Complex instructions require multiple cycles	Reduced instructions take 1 cycle
Many instructions can reference memory	Only Load and Store instructions can reference memory
Instructions are executed one at a time	Uses pipelining to execute instructions
Few general registers	Many general registers

RISC and CISC architectures are becoming more and more alike.
Read the LINK on the web site!

A Bit About ARMs Architecture (Advanced RISC Machine)

- ARM design takes the RISC based computer design approach – Linux –like architecture
- ARM is a British semiconductor (and software) design company that designs and licenses ARM processor cores to semiconductor manufacturers
  - They just sell the ARM *core*
  - Other manufacturers license the core from them and then design microcontrollers around that core by adding in peripherals and memory to suit their design goals
- There are different cores for different applications
  - Cortex-M0/M0+, Cortex-M3, or Cortex-M4.

ARM Processor IP

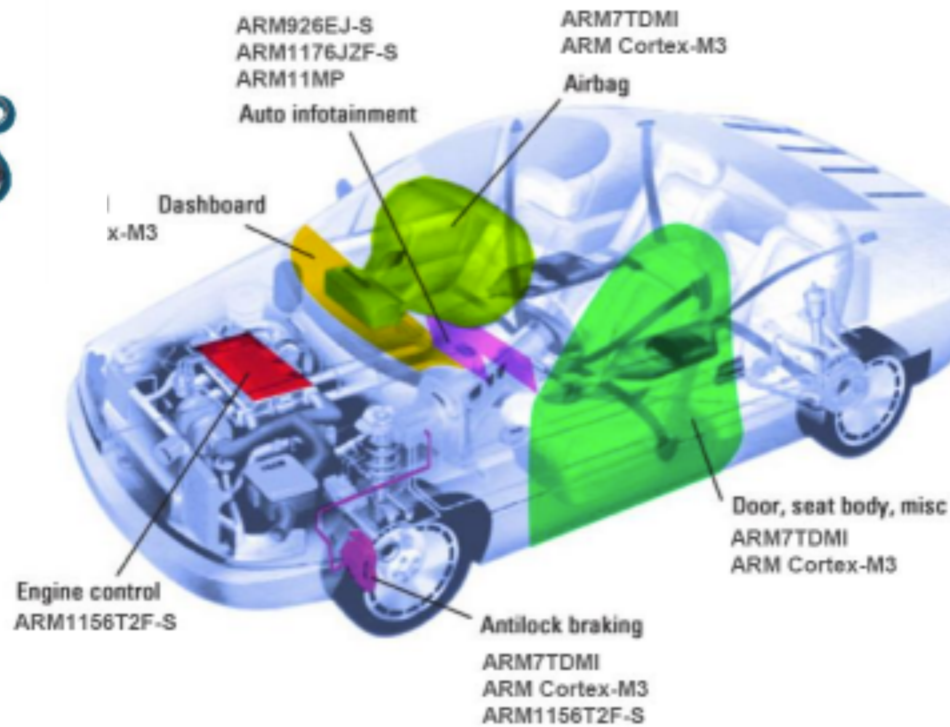


Applications of ARM-Based Microcontrollers

Most Cellphones!

Who is using ARM? Check this out!

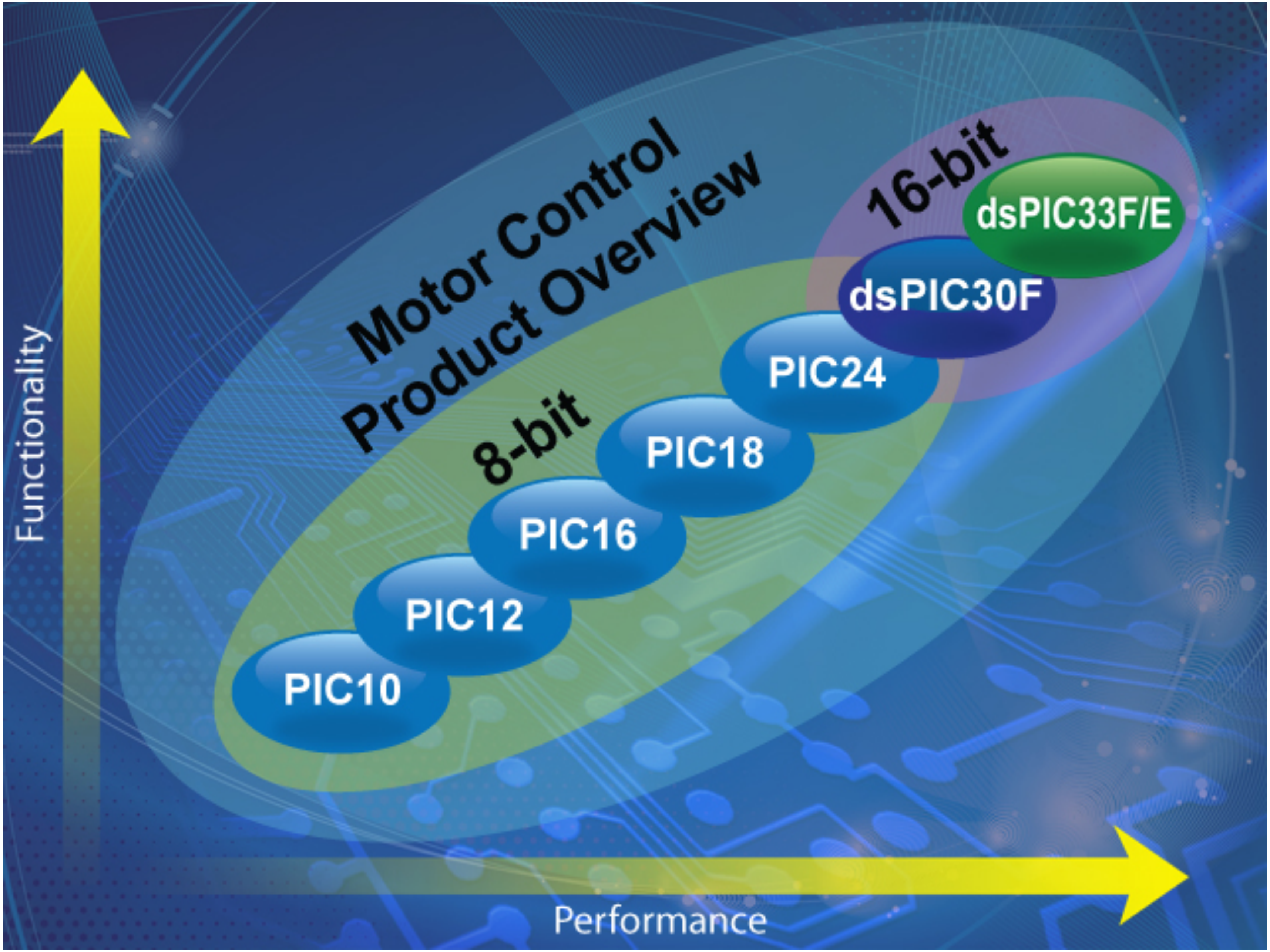
http://en.wikipedia.org/wiki/List_of_applications_of_ARM_cores



Back to Our 8-bit Controllers… (Main Players)

- Microchip
  - RISC architecture (reduced instruction set computer)
  - Has sold over 2 billion as of 2002
  - Cost effective and rich in peripherals
- Motorola
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 68HC05, 68HC08, 68HC11
- Intel
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 8051, 8052
  - Many difference manufacturers: Philips, Dallas/MAXIM Semiconductor, etc.
- Atmel
  - RISC architecture (reduced instruction set computer) – with CISC instruction set!
  - Cost effective and rich in peripherals
  - Claims to be very code efficient – less memory for the same code!
  - AVR (Advanced Virtual RISC): TunyAVR, MegAVR, XmegaAVR
- Freescale
- Ziglog (Z8)

What is the difference?
Speed
Package
Power
RAM/ROM
IO Pins
Software (IDE)/cloud


Motor Control
Product Overview
16-bit
dsPIC33F/E
dsPIC30F
PIC24
8-bit
PIC18
PIC16
PIC12
PIC10
Functionality
Performance

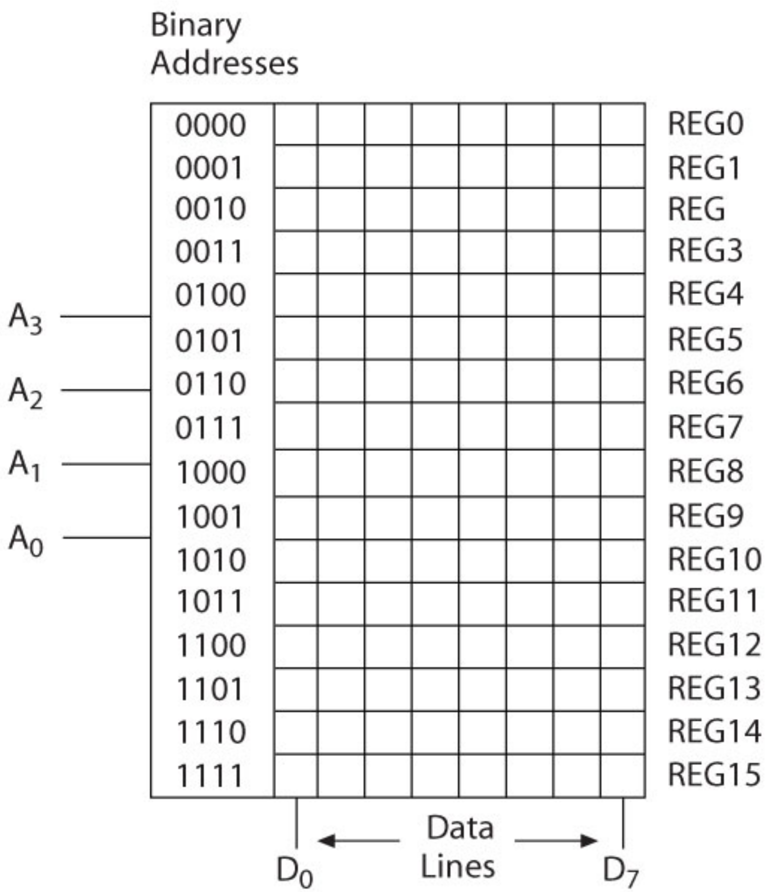
What you Need to Use Microcontrollers

- A *target* - the actual microcontroller
- A *toolchain* — this is the software you use to write your code
  - Most developers use an *IDE* — integrated development environment — which contains a text editor, plus functionality for compiling and downloading your programs to the target
  - The toolchain can be locally installed or on **cloud!**
- A *Programmer/debugger* — this is the device that connects the computer to the microcontroller to download code to it
  - Your PICKIT3!
  - Allows real-time debugging of the program

Programming MCUs….

- Memory devices can store two types of information:
  - Data (RAM)
  - Programs (a series of instructions that tell the MPU in the microcontroller what to do!) - ROM

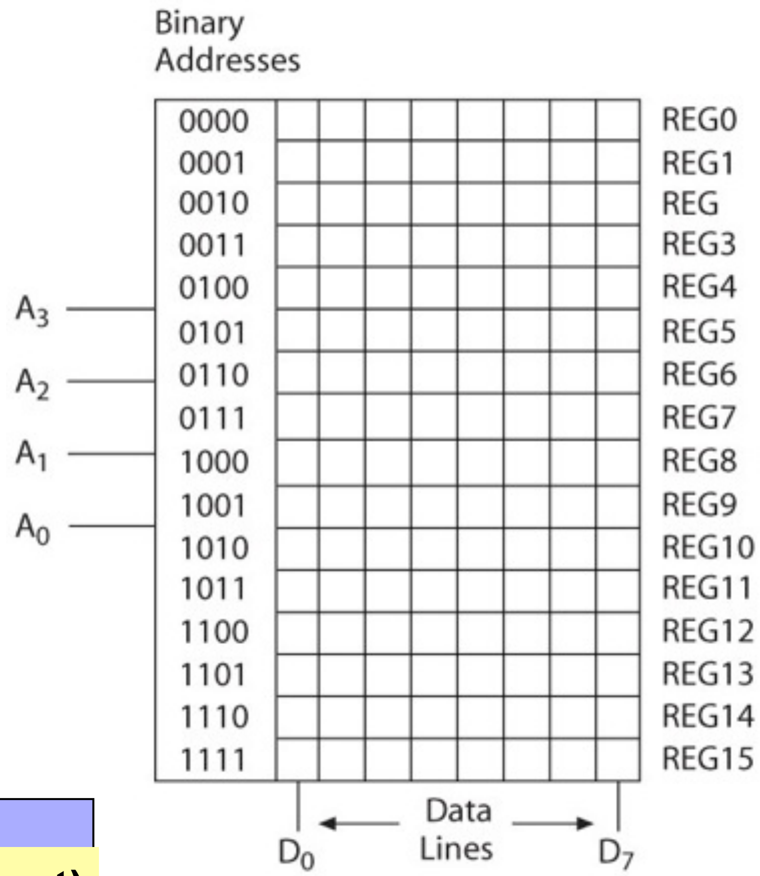
Memory



- A semiconductor storage device consisting of registers that store binary bits
- Two major categories
  - Read/Write Memory (R/WM)
  - Read-only-Memory (ROM)

Storing Bits in Memory

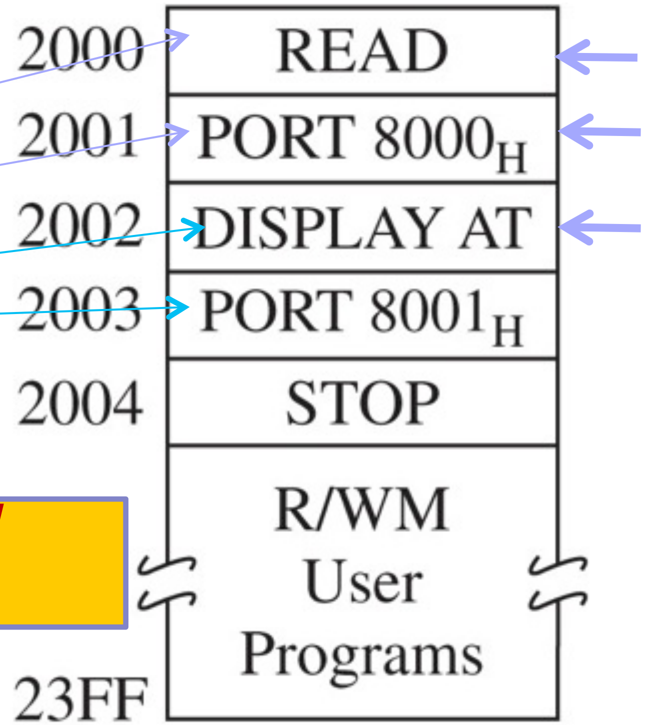
- We can store in different memory types
  - EEPROM, FLASH, RAM, etc.
- In an 8-bit RAM
  - Each byte is stored in a single memory register
  - Each word is stored in two memory locations (registers)
  - DATA 0x1234
    - 0x12→REG11 (High-order byte)
      - 0001 0010
    - 0x34→REG10 (Low-order byte)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



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

Symbolic Representation of Program Memory Contents

- Addresses   Registers



What is the address bus value?

PORT A = 8000H

PORT B = 8001H

CODE:

READ PORT A

WRITE PORT B

STOP

Fetch / Decode / Execute

So, How Do We Right the Instructions and Tell the MPU What to Do?

… We use a Software Language

Software: From Machine to High-Level Languages (1 of 3)

High-level Language
Assembly Language
Machine Language

- 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

Software: From Machine to High-Level Languages (2 of 3)

High-level Language
Assembly Language
Machine Language

- 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

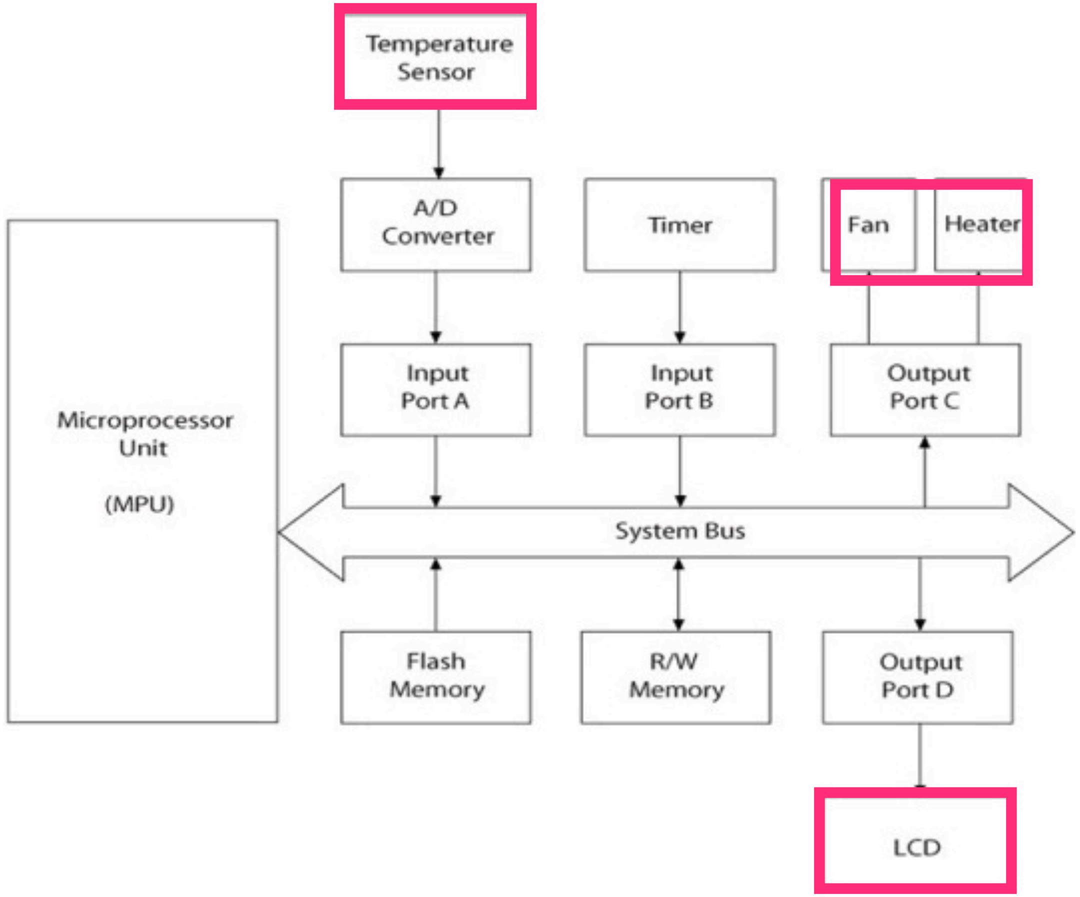
Software: From Machine to High-Level Languages (3 of 3)

- <u>High-Level Languages</u> (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

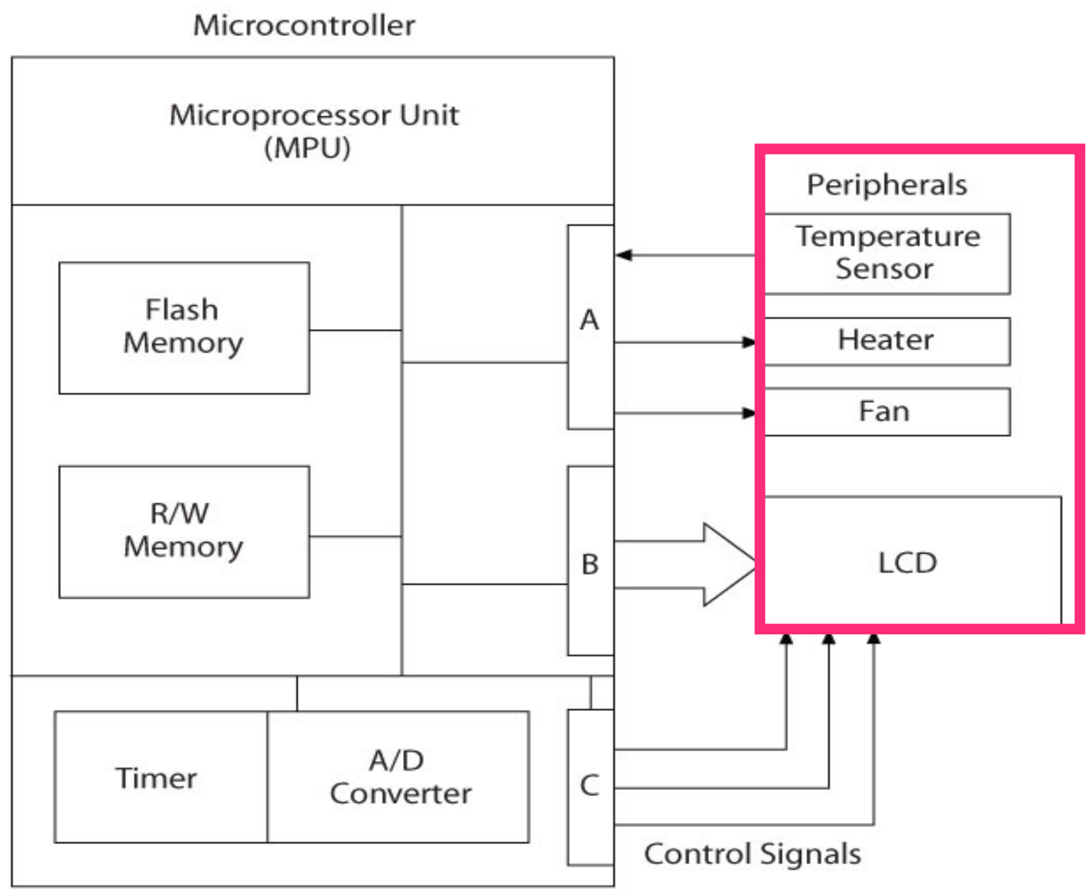
Design Examples …..

Microcontrollers vs. Microprocessors

MPU-Based Time and Temperature System



MCU-Based Time and Temperature System



References

- Computer History Museum: http://www.computerhistory.org/
- Read about microcontrollers: http://www.mikroe.com/en/books/picbook/2_01chapter.htm
- Lots of good information exist on Wikipedia about microcontrollers http://en.wikipedia.org/wiki/
- History of transistors: http://inventors.about.com/library/weekly/aa061698.htm
- Nice transistor timeline by Intel: http://www.intel.com/technology/timeline.pdf
- I used a few slides from here: http://www.ceng.metu.edu.tr/courses/ceng336/_documents/introduction.pdf
- ARM related references:
  - http://mc2.unl.edu/2013/10/03/getting-started-with-arm-microcontrollers/
  - http://www4.cs.fau.de/Lehre/SS06/HS_AKES/slides/ARM.pdf - Very good reference !

References - RISC

- http://cse.stanford.edu/class/sophomore-college/projects-00/risc/
- http://en.wikipedia.org/wiki/Complex_instruction_set_computer
- http://en.wikipedia.org/wiki/RISC
- http://arstechnica.com/articles/paedia/cpu/pipelining-1.ars/4