Microcontroller (EEC421)

Lecture 1

Dr. Islam Mohamed

Electrical Engineering Department
Shoubra Faculty of Engineering, Benha University
Islam.ahmed@fen.bu.edu.eg

References

The 8051 Microcontroller
Kenneth J. Ayala

Chapters 1:6

8051 Microcontrollers An Applications-Based Introduction
David Calcutt, Hassan Parchizadeh

Chapters 1&7

Course Topics

1- INTRODUCTION (MICROPROCESSORS AND MICROCONTROLLERS)
2- THE 8051 ARCHITECTURE:
- ✓8051 Microcontroller Hardware
- ✓Counters, Registers, Memory
- ✓Input / Output Pins, Ports, and Circuits
- ✓Timers, Serial data

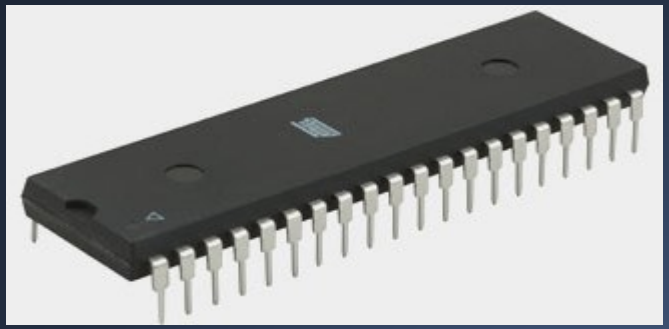
3-MOVING DATA
- ✓ Addressing Modes
- ✓External Data Moves
- ✓Data Exchanges

4- LOGICAL OPERATIONS
5- ARITHMETIC OPERATIONS
6-PROJECT APPLICATIONS
- ✓speed control of a small DC motor
- ✓speed control of a stepper motor
- ✓function generator
- ✓Measuring Frequency

An introduction to Microcontrollers

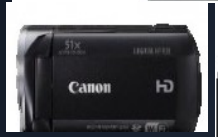
What is a microcontroller?

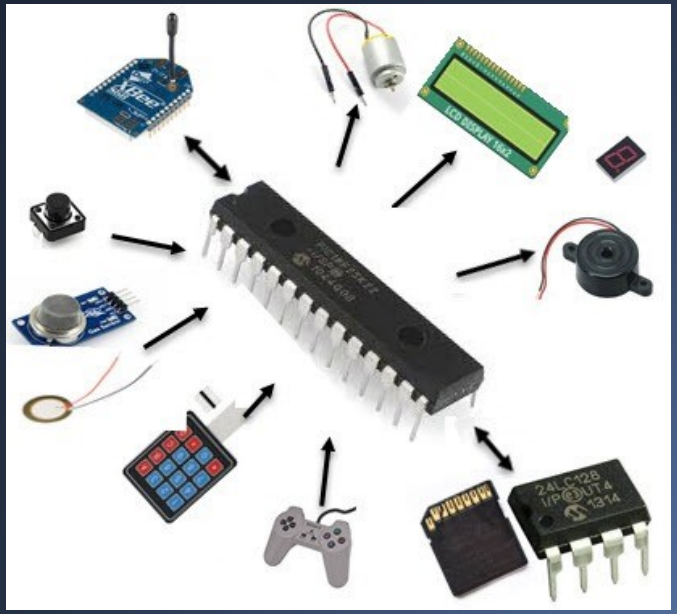
- A microcontroller is an integrated circuit that is programmed to do a specific task.

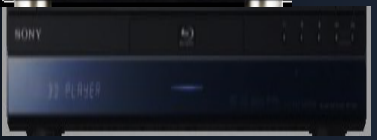


- Microcontrollers are really just "mini-computers".

Where we can find microcontroller

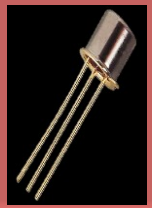






History of Microcontrollers

Vacuum Tube
1939



Transistor
1947

Logic Gate
1960

Microcontroller
1971

Microprocessor –vs- Microcontroller

- You may have heard of the term “microprocessor” or just “processor” before. You may ask, “Is there a difference between a microprocessor and microcontroller?”

Microprocessor unit (MPU)

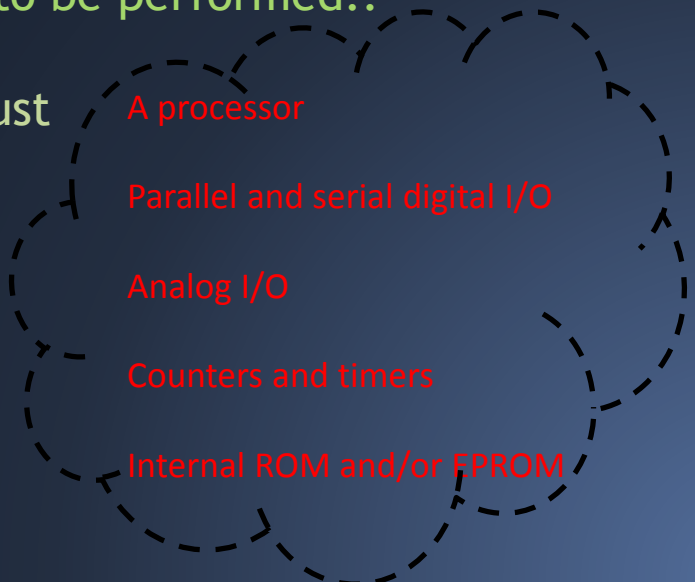
- Microprocessors are general purpose computing devices which incorporate all the functions of the central processing unit on a chip but do not include peripherals like memory and input and output pins like the microcontroller

Note: By itself, it is totally useless.

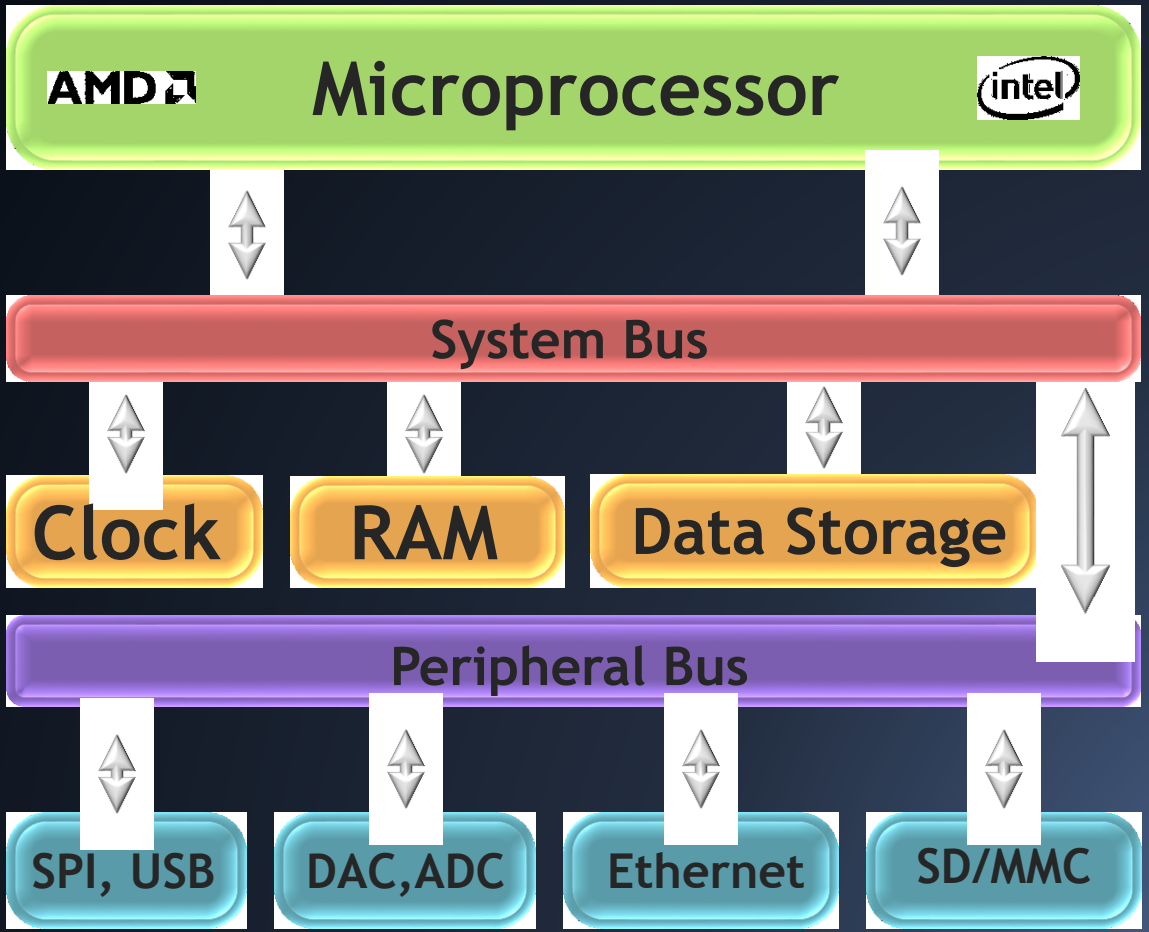
Microcontroller unit (MCU)

- Microcontrollers are small computing devices on a single chip that contain one or more processing cores, with memory devices embedded alongside programmable special and general purpose input and output (I/O) ports. They are used especially in applications where only specific repetitive tasks need to be performed..

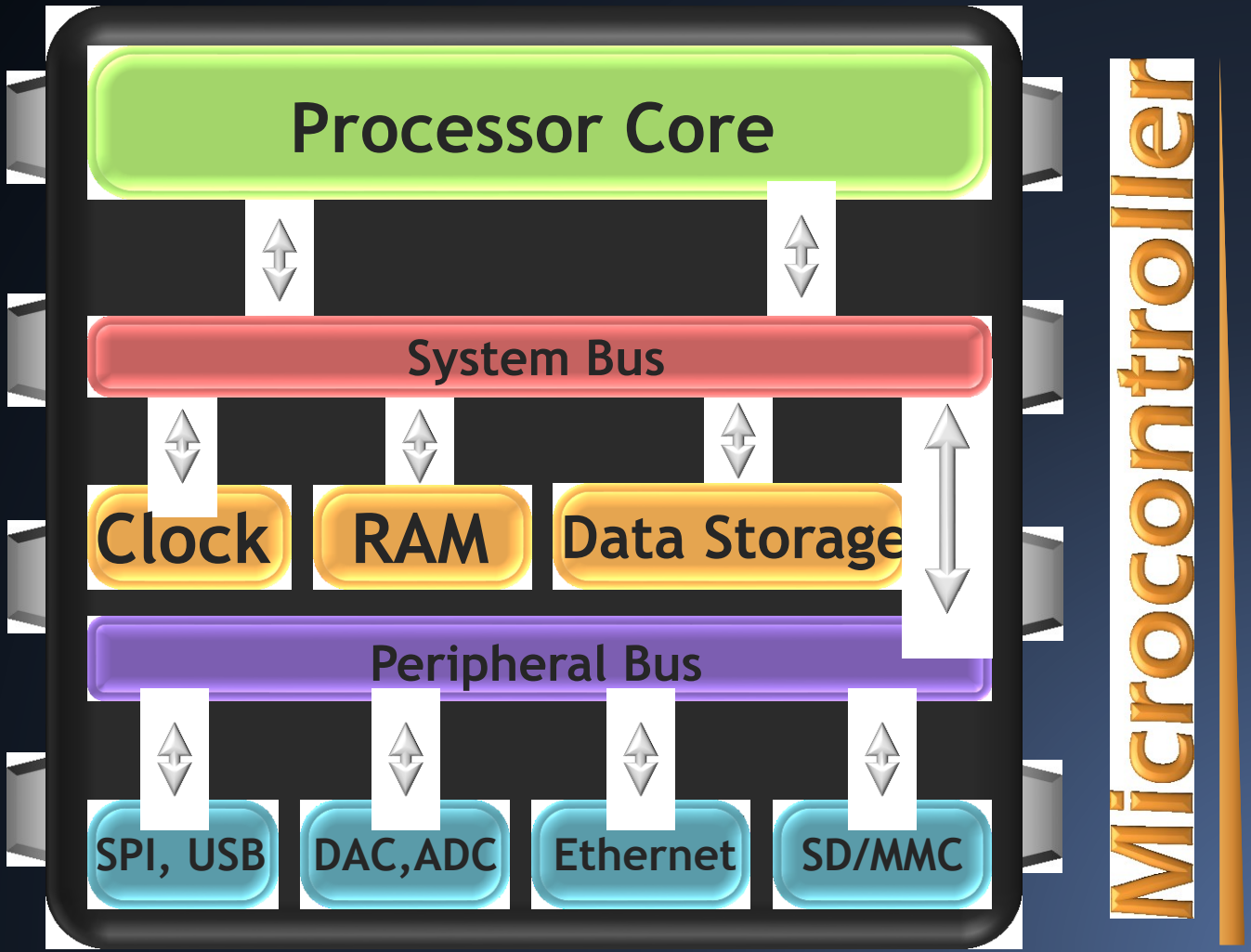
- Microcontrollers are really just "mini-computers" contain



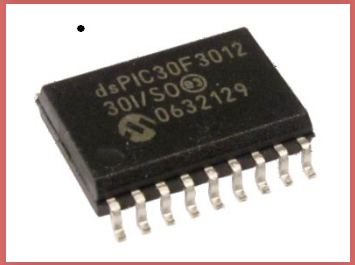
	Microprocessor	Microcontroller
Applications	General computing (i.e. Laptops, tablets)	Appliances, specialized devices
Speed	Very fast	Relatively slow
External Parts	Many	Few
Cost	High	Low
Energy Use	Medium to high	Very low to low
Vendors	  	  

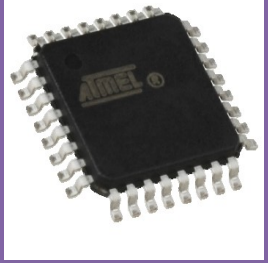


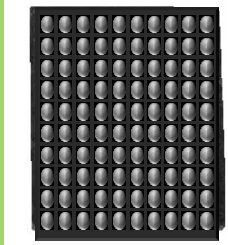
Microprocessor


Processor Core
System Bus
Clock
RAM
Data Storage
Peripheral Bus
SPI, USB
DAC,ADC
Ethernet
SD/MMC
Microcontroller

Microcontroller Packaging







DIP	SOIC	QFP	BGA
(Dual Inline Package)	(Small Outline IC)	(Quad Flat Package)	(Ball Grid Array)
Through hole	Surface Mount	Surface Mount	Surface Mount
8 pins	18 pins	32 pins	100 pins
9mm x 6mm	11mm x 7mm	7mm x 7mm	6mm x 6mm
0.15pins/mm^2	0.23pins/mm^2	0.65pins/mm^2	2.78pins/mm^2

Basic Principles of Operation

- Microcontrollers are used for specific applications.
- They do not need to be powerful because most applications only require a clock of a few MHz and small amount of storage.
- A microcontroller needs to be programmed to be useful.
- A microcontroller is only as useful as the code written for it. If you wanted to turn on a red light when a temperature reached a certain point, the programmer would have to explicitly specify how that will happen through his code.

Microcontroller Programming

1.) Code is written for the microcontroller in an integrated development environment, a PC program. The code is written in a programming language. (e.g. C, BASIC or Assembly).

2.) The integrated developer environment (IDE) debugs the code for errors, and then compiles it into binary code which the microcontroller can execute.

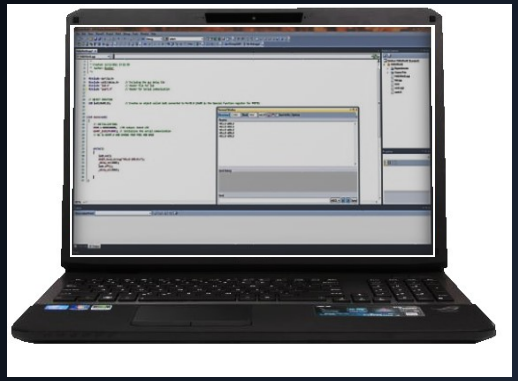
3.) A programmer (a piece of hardware, not a person) is used to transfer the code from the PC to the microcontroller. The most common type of programmer is an ICSP (In-circuit serial programmer).

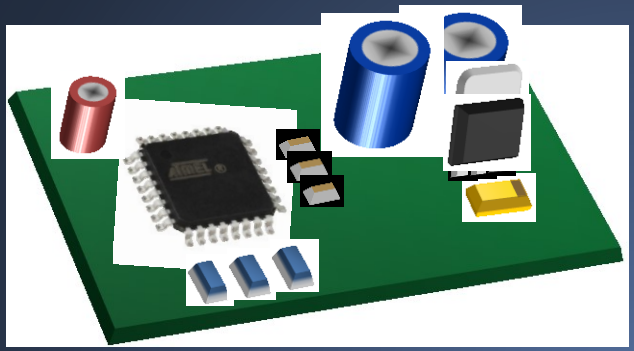
What languages are they being programmed in?

	1998-1999	1999-2000
Assembly	~ 21%	~ 10%
C	~ 69%	~ 80%
C++	~ 5%	~ 6%
Java	~ 1 %	~ 2%
Other	~ 3 %	~ 2%

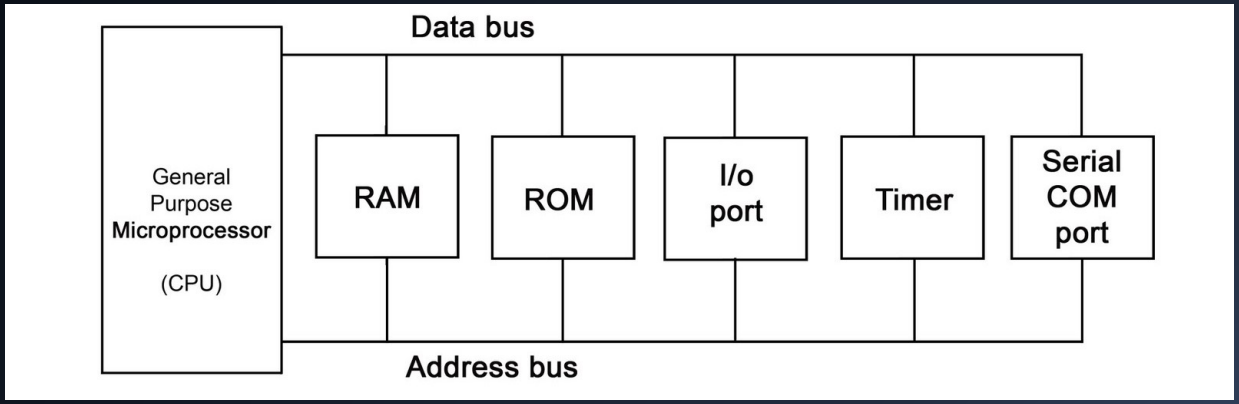
Source: TRON Association Survey 1998/99 & 1999/2000

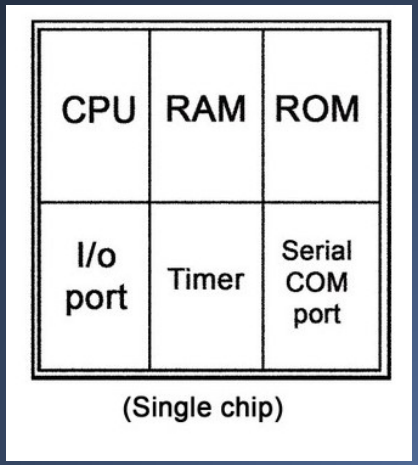
Microcontroller Programming



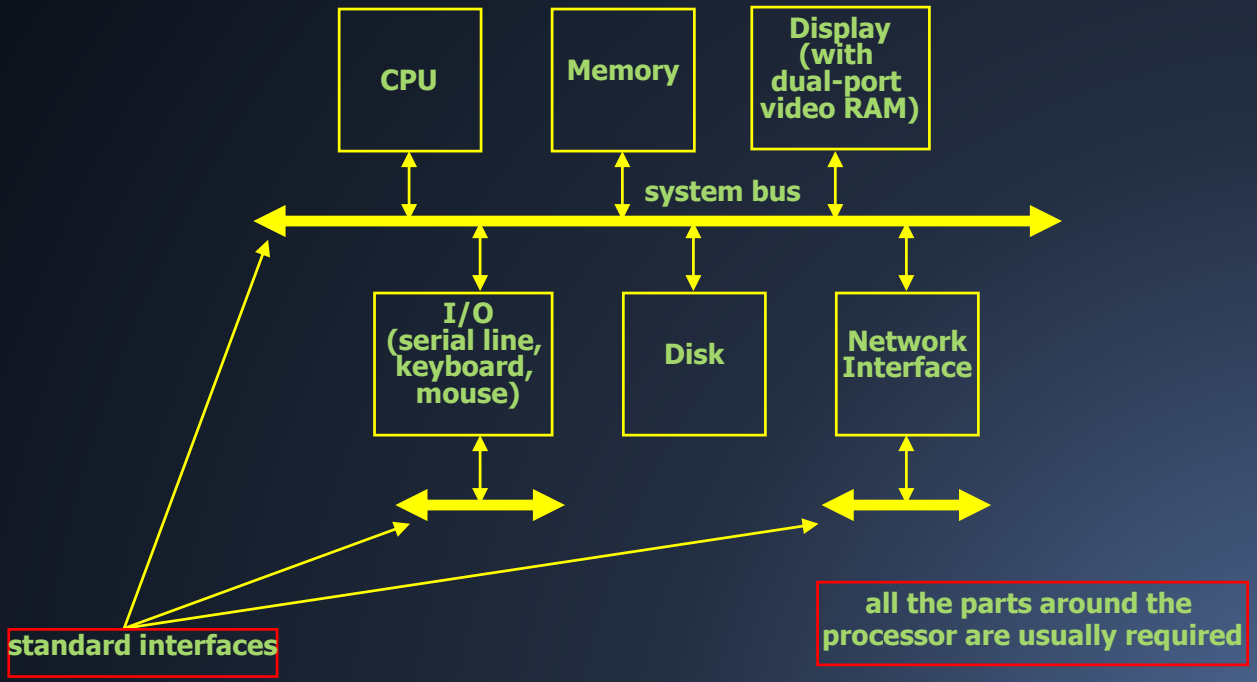


A Single Chip MCU





Typical general-purpose architecture

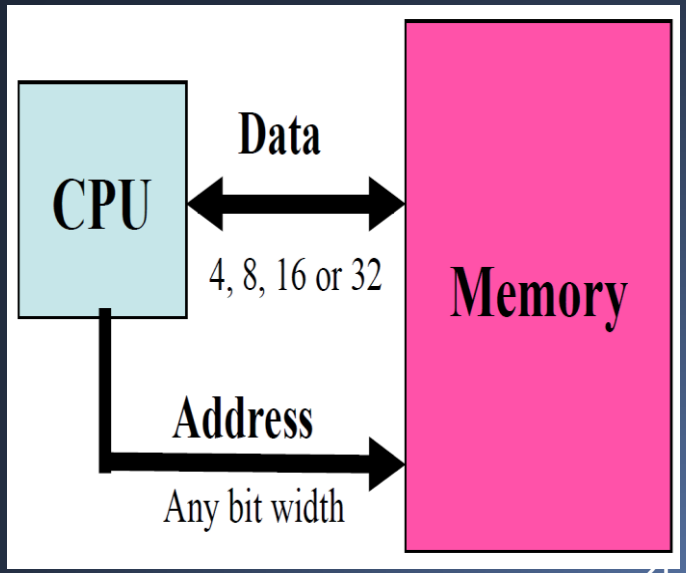


MCU Types

- **MCUs can be classified according:**
  - **MCU bits (e.g. 8-bit, 16-bit, 32-bit)**
  - **Instruction Set Architecture (ISA)**

MCU Bit Definition

The number of bits describing the data path defines Microcontroller Bit Definition

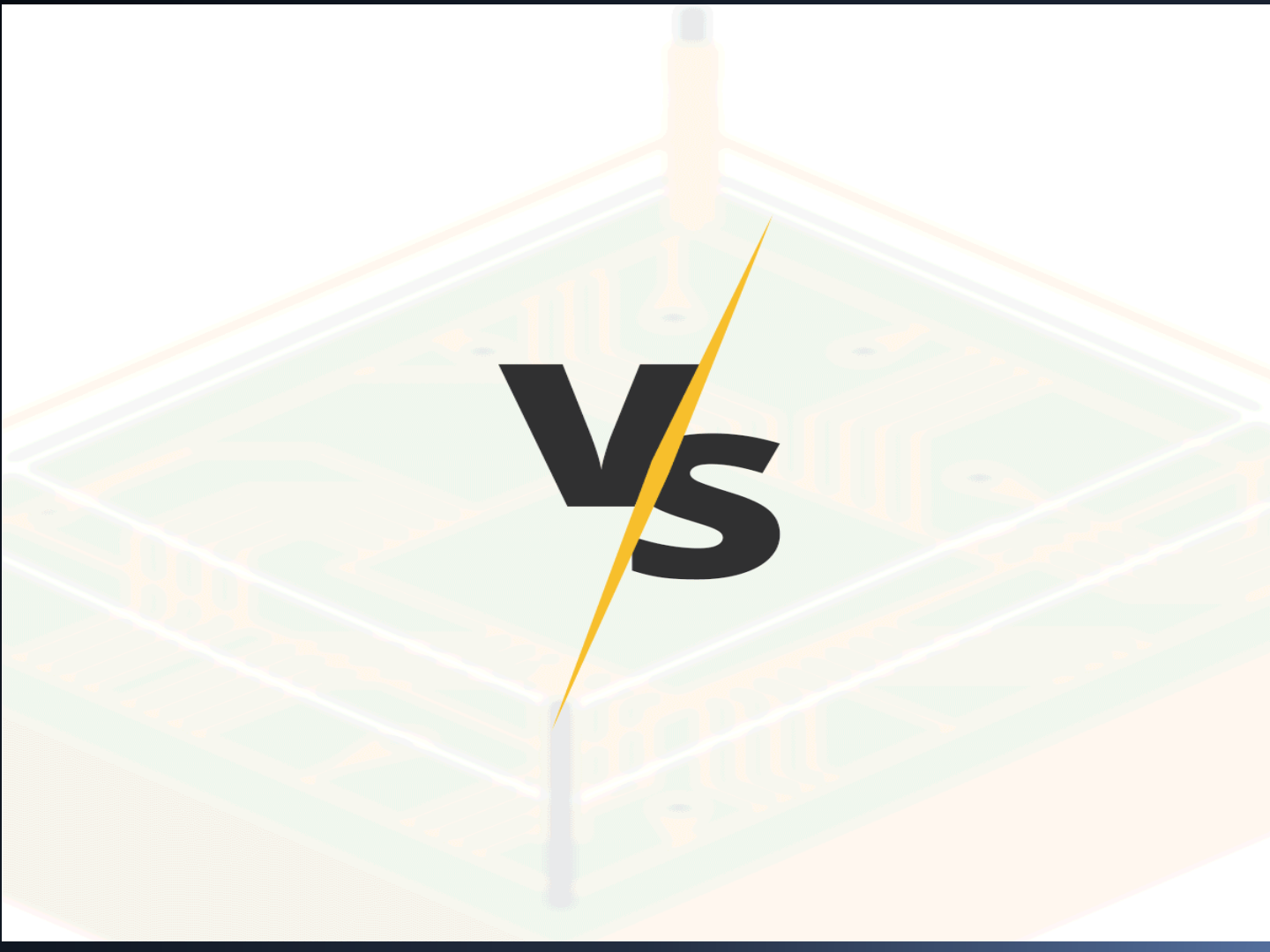


8-bit MCU

- processes 8-bits of data at any particular time.
- the size of the registers (8 bits per register)
- number of memory addresses (only 2^8 = 256 addresses)
- largest numbers they can process (2^8 = 256 integers, or integers 0 through 255 or from -127 to 128))
- processing speed 8 Mhz.

32-bit MCU

- processes 32-bits of data at any particular time.
- They have 32-bit arithmetic logic units, registers, and bus width.
- it can handle quadruple the amount of data, making it technically more data efficient
- It can theoretically handle numbers reaching 2^32 (0 to 4,294,967,295)
- Processing speed hundreds of Mhz


VS

MCU Types

Instruction Set Architecture (ISA):

is the part of the computer architecture related to programming, including the native data types, instructions, registers, addressing modes, memory architecture, interrupt and exception handling, and external I/O.

- The ISA defines:
  - Operations that the processor can execute,
  - The mechanism of Data Transfer and how to access data,
  - Control Mechanisms (branch, jump, etc)
  - “Contract” between programmer/compiler and hardware

Common Memory Types in MCUs

- RAM
- EEPROM
- Flash Memory

Common Memory Types in MCUs

RAM: Random Access Memory

- Volatile or Non-Permanent Memory, i.e., data is lost after the removal of power,
- Can be Written too Many Times,

- It is a general purpose memory which is typically used for storing user data, temporary or Changeable Data

- Most microcontrollers have some amount of internal RAM. Generally, 1 kbytes (“Embedded” in MCUs) is a common amount, although some microcontrollers have more, some less.

Common Memory Types in MCUs

- **EEPROM: Electrically Erasable Programmable Read-Only Memory**

- It is electrically-erasable-and-programmable.

- Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light. Any byte within an EEPROM may be erased and rewritten.– Erased by higher voltage.
- The primary tradeoff for this improved functionality is higher cost, though write cycles are also significantly longer than writes to a RAM.
- EEPROM are typically used to store permanent data

Common Memory Types in MCUs

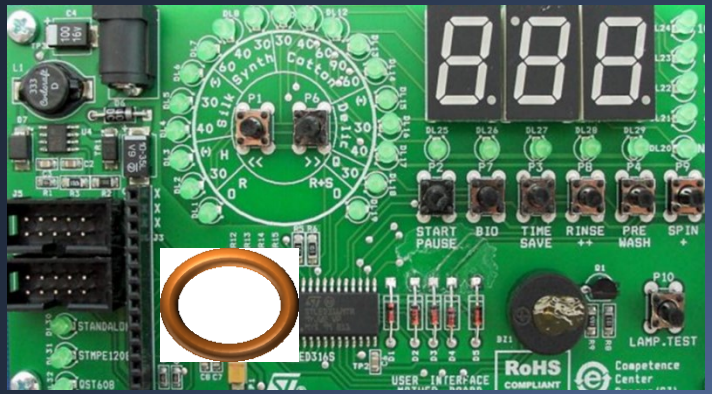
Flash Memory

- Flash memory devices are:
- ✓ high density,
- ✓ low cost,
- ✓ nonvolatile,
- ✓ fast (to read, but not to write), and
- ✓ electrically reprogrammable.
- ✓ Write/Erase large blocks of bytes
- ✓ Read bytes
- EEPROM is similar to flash memory (sometimes called flash EEPROM).
- The principal difference is that
- EEPROM requires data to be written or erased one byte at a time whereas flash memory allows data to be written or erased in blocks.
- Flash memory is typically used in MCUs for storing program code.

Microcontroller Applications

- This is the controller board for a washing machine. If a button is pushed or if a knob is turned, the microcontroller knows how to react to the event.

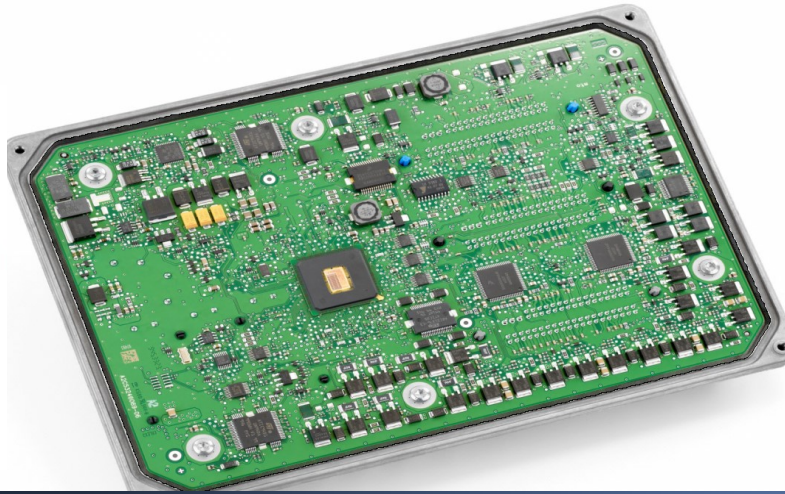
- Ex. If “start” is pushed, the microcontroller knows to switch a relay which starts the motor.



Microcontroller Applications

- This is the main controller from a Buick Regal. This board has several microcontrollers each for a specific task.

- Ex. A microcontroller may handle dashboard controls or it may even control something more complex like the ignition system.



Microcontroller Applications

- Many robots use microcontrollers to allow robots to interact with the real world.

- Ex. If a proximity sensor senses an object near by, the microcontroller will know to stop its motors and then find an unobstructed path.

Thank You