Microcontrollers Programming

PG-DIOT

-GAYLE FERNANDES

ACTS, CDAC PUNE

SEPTEMBER 2022

Why are we studying this?

- Microcontrollers are the ‘brains’ of the embedded system
  - They control the system, they reside in
- Microcontroller offer programmability
  - Useful for providing functionality of the embedded system
- Microcontroller offer connectivity through IO Interfaces

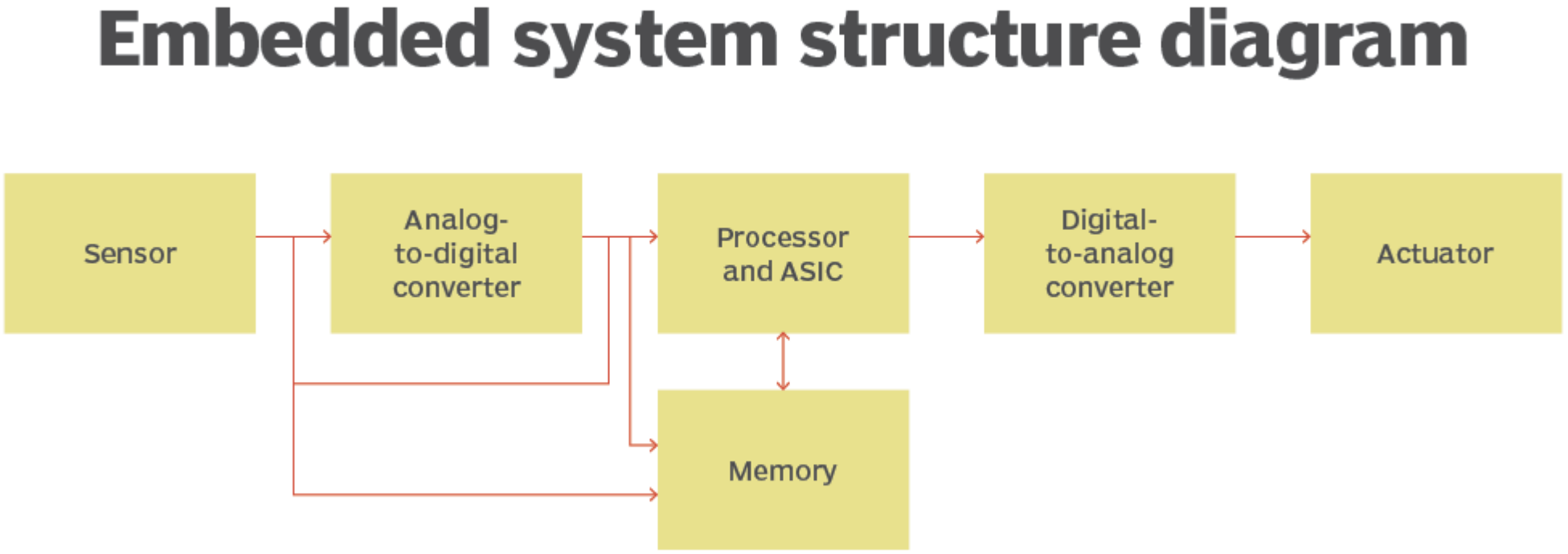
What are we studying?

- Basics of embedded system
- Basics of Microcontrollers
- Detailed study Microcontroller - ARM
- Memory and IO
- Communication Protocols
  - UART, SPI, I2C
  - CAN
  - USB

What is an embedded system?

- An embedded system is a combination of computer hardware and software designed for a specific function.
- is a special-purpose computer system
- completely encapsulated by the device it controls.
- has specific requirements and performs pre-defined tasks
- is a programmed hardware device.
- A programmable hardware chip is the 'raw material’
- Programmed with particular applications.
- The software controls the operation and functionality

What is an embedded system?



Where do we find Embedded Systems

- Home appliances
  - Microwave ovens, washing machines,
- Entertainment devices
  - TV, DVD players, MP3 players
- Telecommunication
  - Mobile phones, Telephones
- Computers
  - Keyboards, Disk drives
- Automobiles
  - Dashboard, Fuel Injection, suspension, GPS

Types of embedded systems

- **Mobile embedded systems** are small-sized systems that are designed to be portable. **Digital cameras**

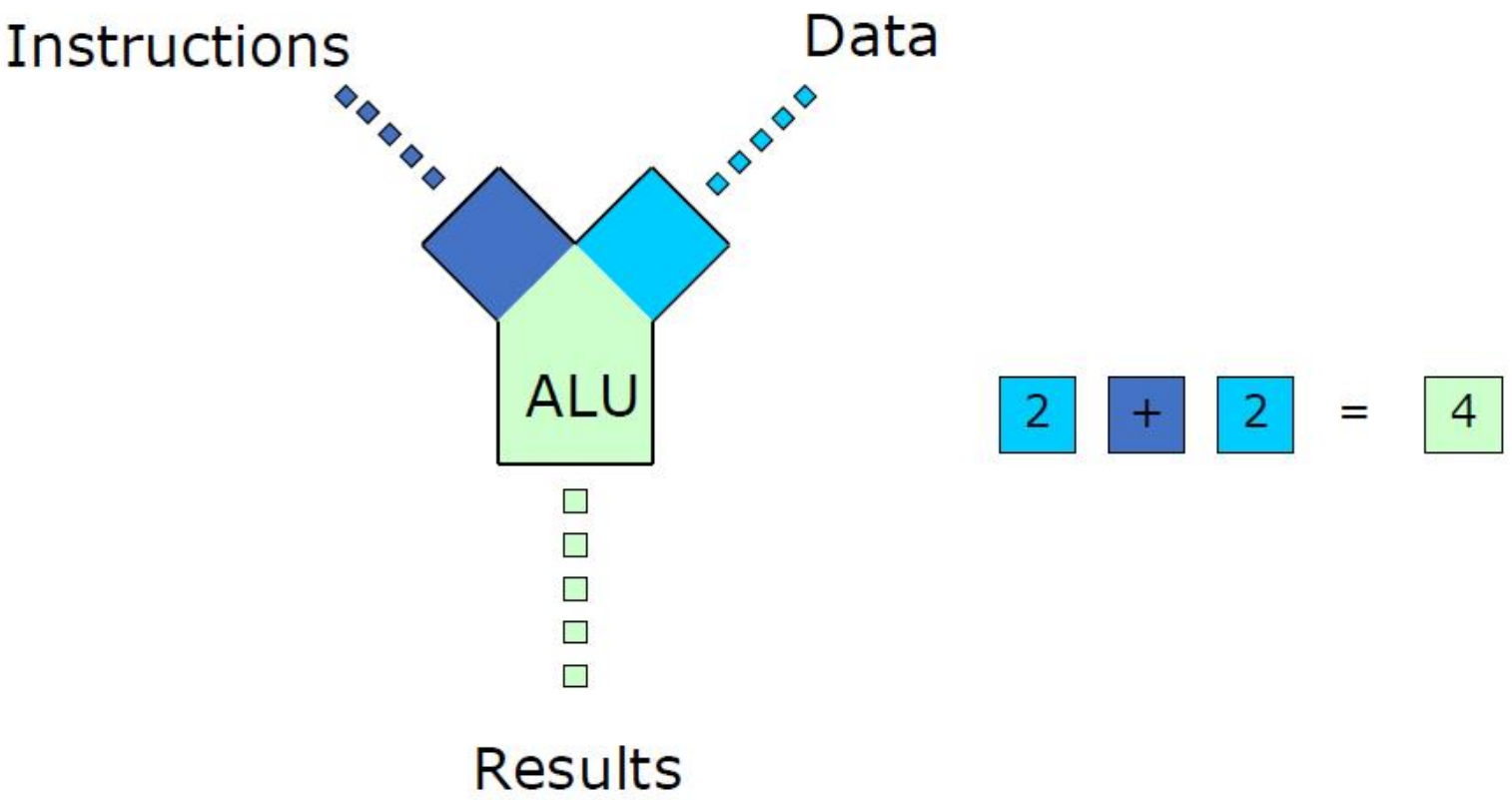
- **Networked embedded systems** are connected to a network to provide output to other systems. **Home security systems and point of sale (POS) systems.**

- **Real-time embedded systems** give the required output in a defined time interval. They are often used in medical, industrial and military sectors because they are responsible for time-critical tasks. **Traffic control**

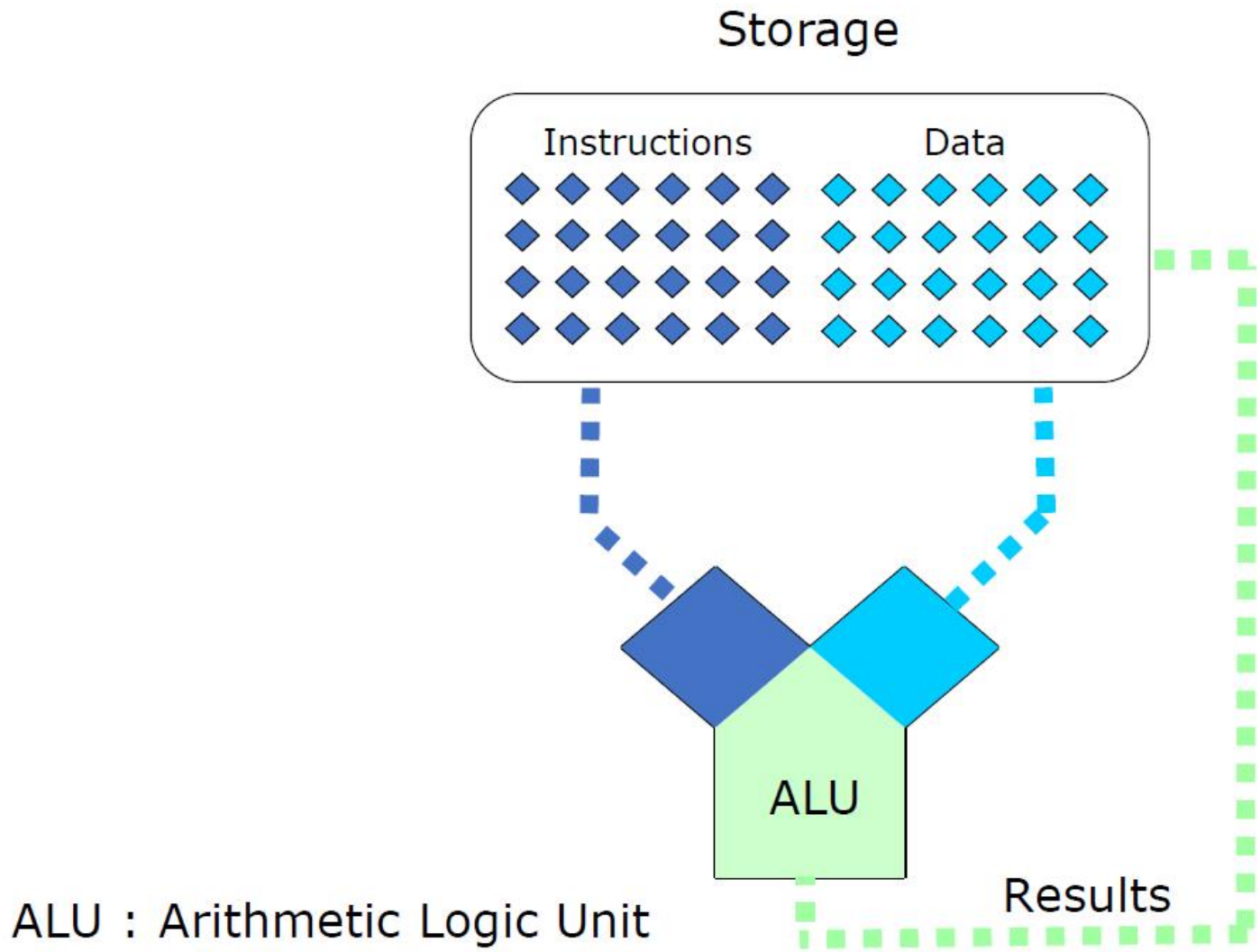
Features

- Constituents: hardware and software
- Timelessness
  - Controller must respond fast to keep operation within safe region
  - Operate in real time
- System Interconnection
  - For interaction between sub-systems e.g. Car
  - Internet enabled embedded systems
- Reliability
  - Products cannot afford to have bugs
  - Upgrade not easy

The heart of the processor - Simplified

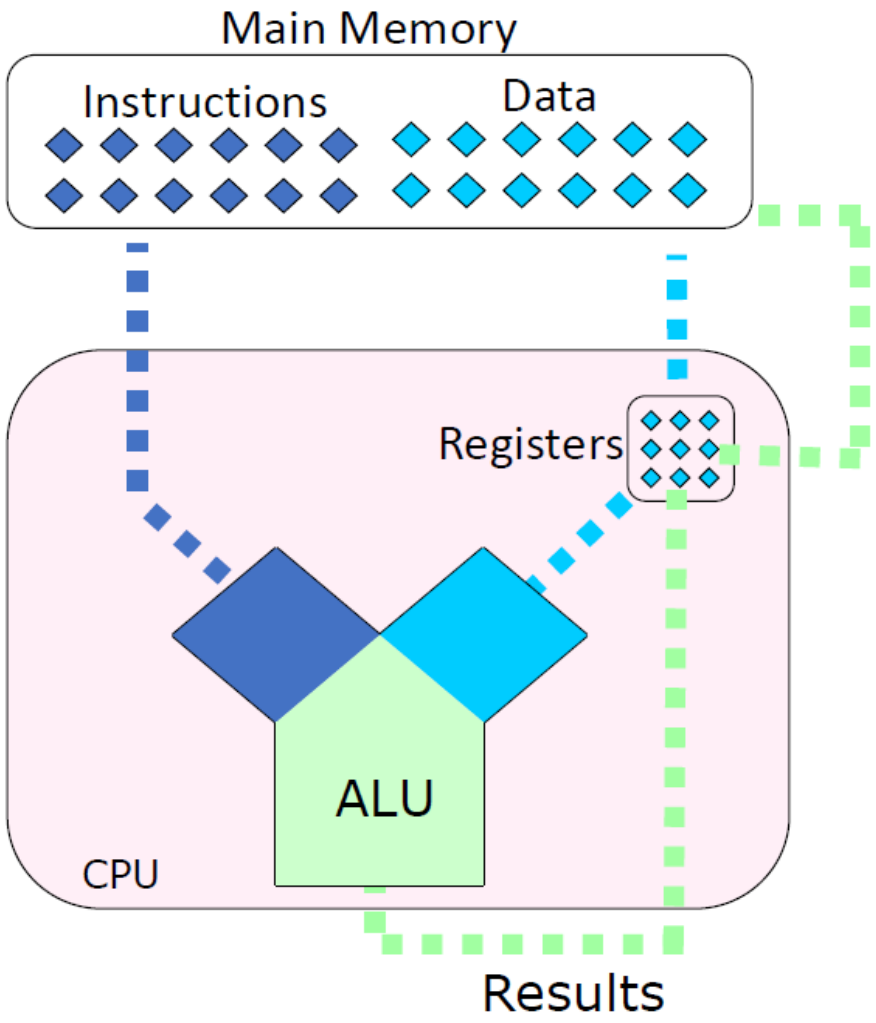


Instruction and Data

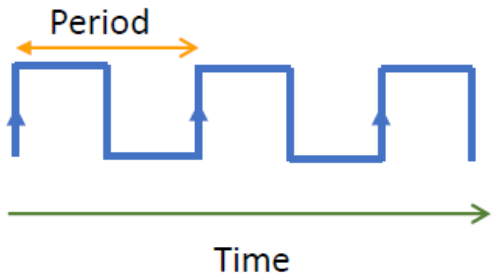


ALU : Arithmetic Logic Unit

CPU with registers



- CPU executes instructions, one after another in a sequential manner
- Each instruction operates on its corresponding data
- To make this happen, a clock signal is provided, which acts like a heartbeat

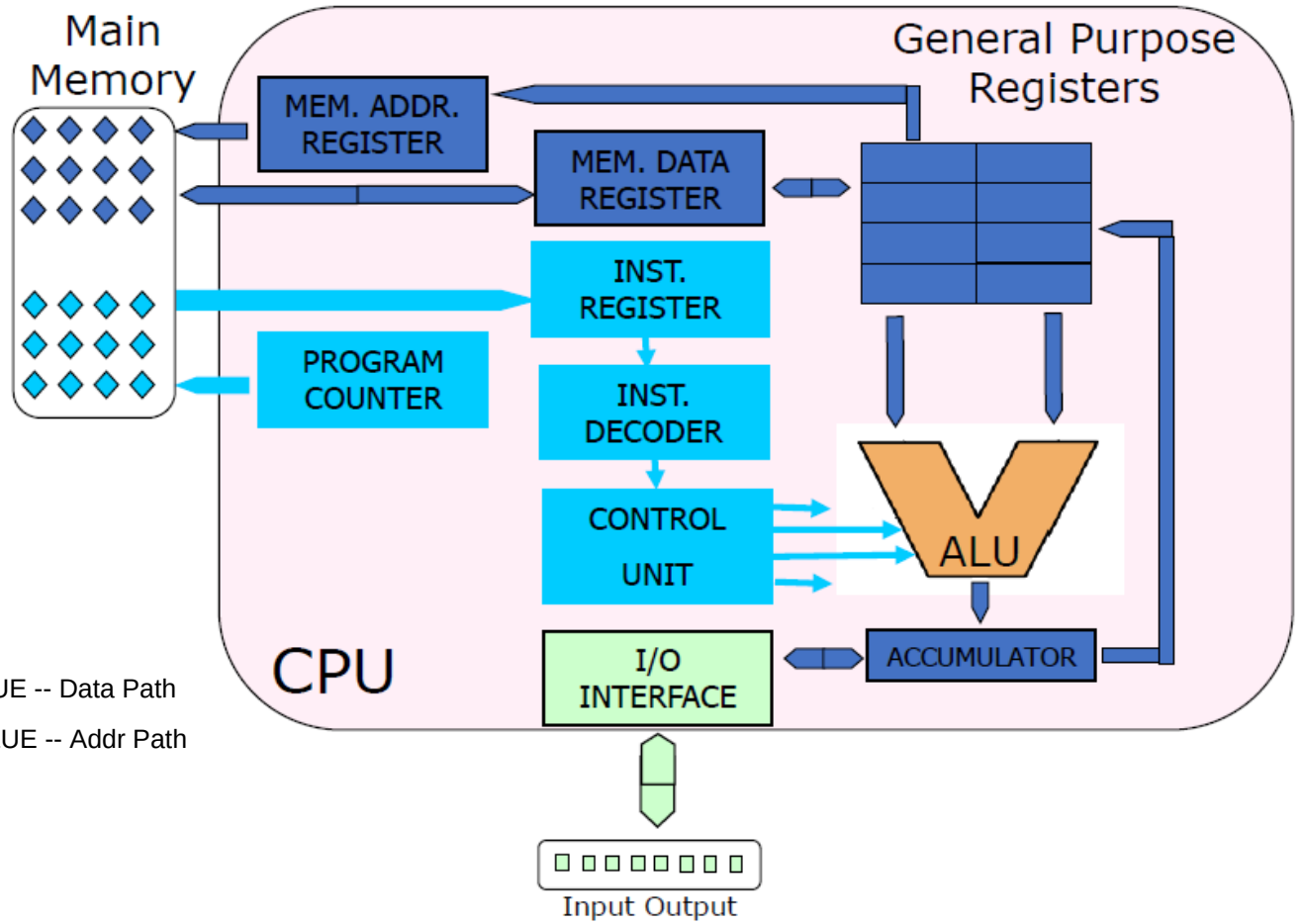


Binary Decimal Hexadecimal table

Decimal	Binary	Hexa
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Decimal	Binary	Hexa
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

CPU – Some more details



DARK BLUE -- Data Path

LIGHT BLUE -- Addr Path

- Executing instructions involves typically following tasks
  - Fetch
  - Decode
  - Execute
  - Writeback

Components of a Processor

ARITHMETIC AND LOGIC UNIT

- An arithmetic/logic unit (**ALU**) performs integer arithmetic and logic operations.
- It also performs shift and rotate operations and other specialized operations.

CONTROL UNIT

- Is in-charge of the computer. Control units fetch and decode machine instructions

Bus

BUSES

- A bus is a set (group) of parallel lines that carries data, addresses, instructions, and other information
- Information travels on buses as a series of electrical pulses, each pulse representing a one bit or a zero bit
- The size or width of a bus is how many bits it carries in parallel. Common bus sizes are: 8 bits, 16 bits,32 bits, etc
- An internal bus is inside the processor, while an external bus is a bus outside of the processor (but inside the computer),

Storage

- **Registers** and **flags** are a special kind of memory that exists inside a processor.
- Typically a processor will have several internal registers that are much faster than main memory.
- These registers usually have specialized capabilities for arithmetic, logic, and other operations.
- Registers are usually fairly small
  - 8, 16, 32, bits for integer data, address, and control registers;
  - 32, 64, 96, or 128 bits for floating point registers

Storage

- Some processors separate integer data and address registers, while other processors have general-purpose registers that can be used for both data and address purposes.
- A processor will typically have one to 32 data or general purpose registers
- Flags are single bit memory used for testing, comparison, and conditional operations
  - Typically conditional branching

Flags

- **Processor flags** store information about status of result.
  - This can include result flags that record the results of certain kinds of testing
- **Auxiliary carry:** Carry from LS nibble to MS nibble
- **Carry:** Carry out of the MS bit of a result
- **Negative:** Set if the MS bit of a result is set
- **Overflow:** Set if arithmetic overflow occurs
- **Sign:** Set for negative sign

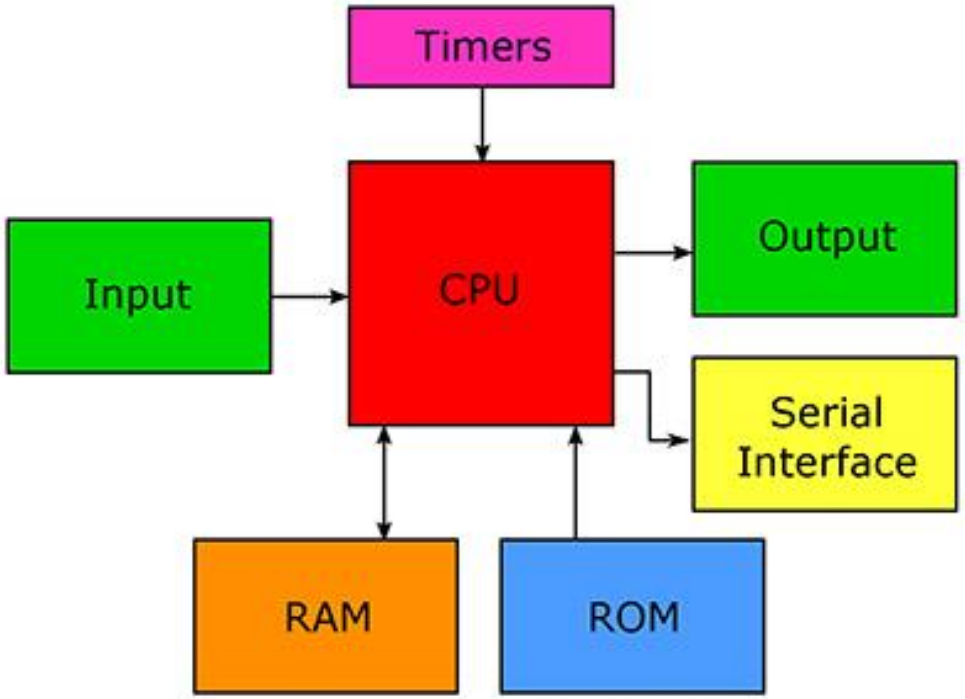
MS: Most Significant

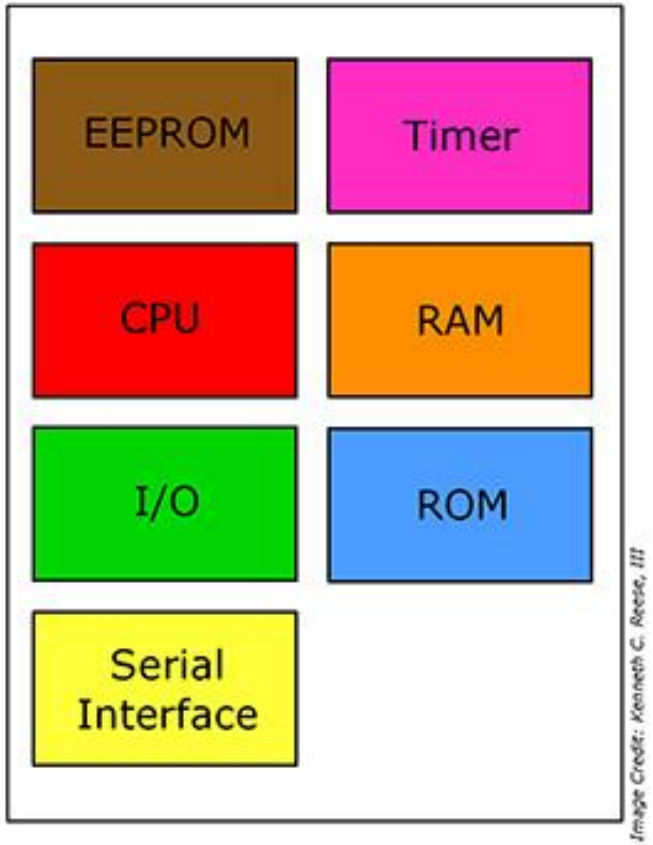
LS: Least Significant

microcontroller vs microprocessor

Microprocessor	Microcontroller
a. Microprocessors are widely used in computer systems.	a. Microcontroller is widely used in embedded systems.
b. It has only a CPU embedded into it.	b. It has a CPU, a fixed amount of RAM, ROM and other peripherals all embedded on it.
c. In case of microprocessors we have to connect all the components externally so the circuit becomes large and complex.	c. As all the components are internally connected in microcontroller so the circuit size is small.
d. It consumes more power.	d. It consumes less power than a microprocessor.
e. It has very less internal register storage so it has to rely on external storage. So all memory operations are carried out using memory based external commands which results in high processing time.	e. It has many registers so processing time is less.

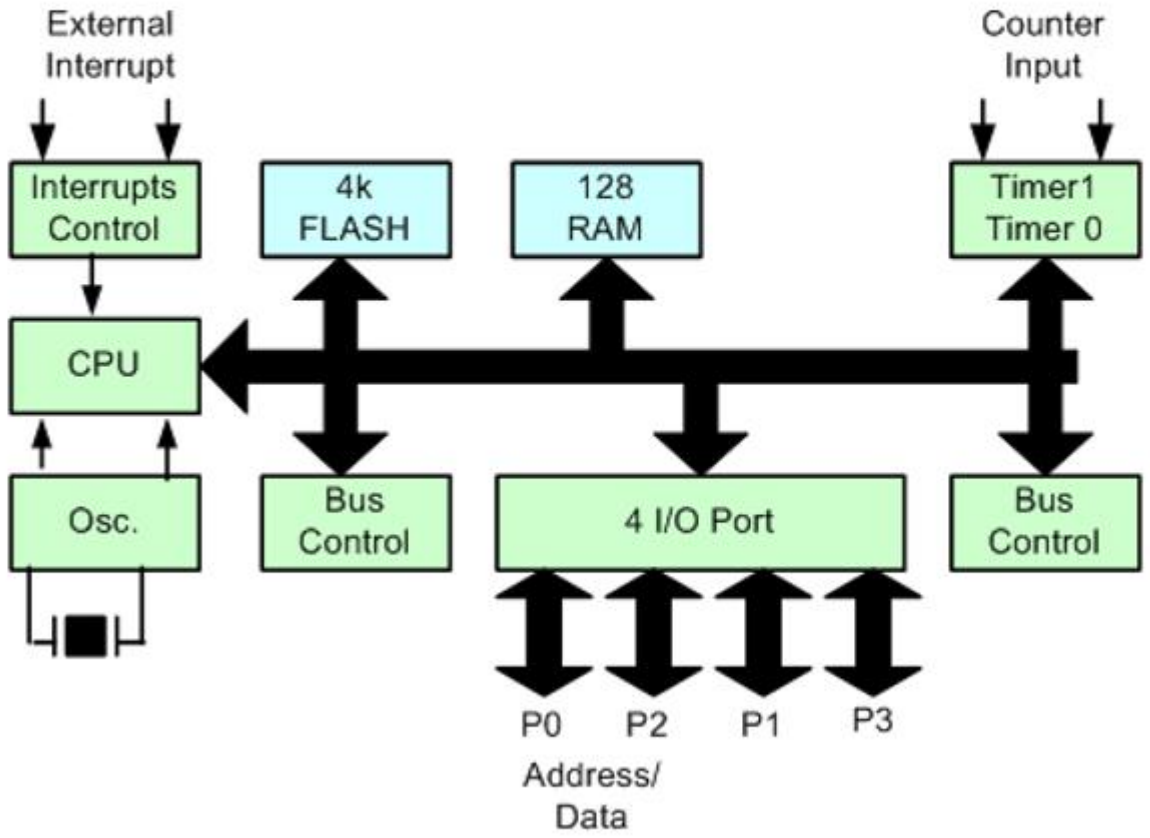
microcontroller vs microprocessor





Block Diagram

- CPU
- Program Memory
- Data Memory
- Peripherals
  - Timers
  - IO ports

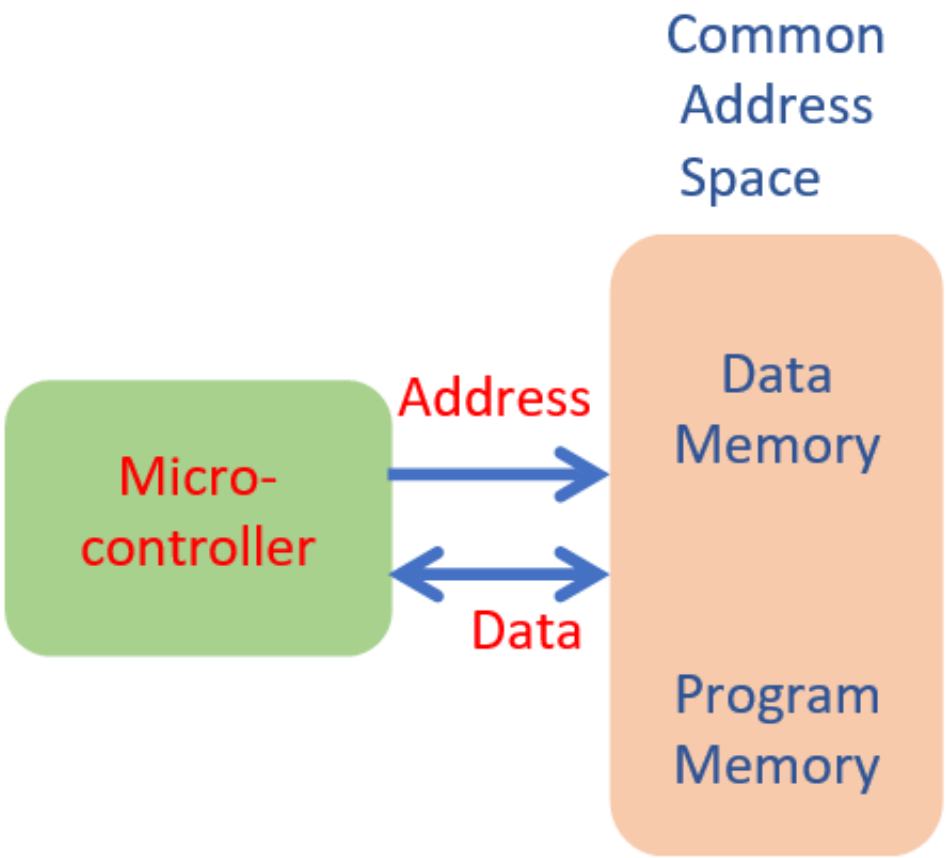


Memory Models

- There are two fundamental ways the CPU accesses memory
  - Von Neumann Architecture
  - Harvard Architecture
- There are two fundamental ways of storing numbers in memory
  - Little endian
  - Big endian

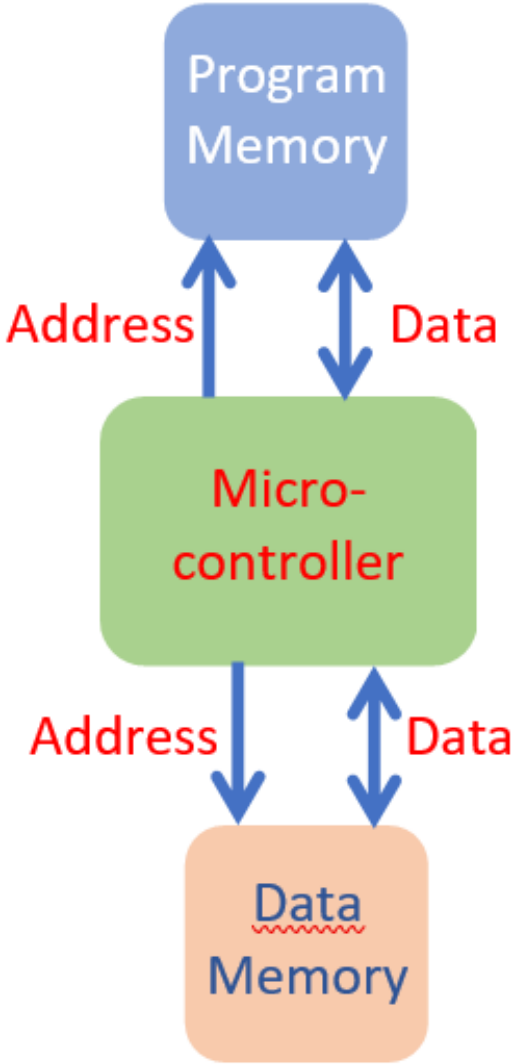
Von-Neumann

- Program and data memory share same address space
- Single path to access program or data
- Data memory is idle when program is fetched
- Storing or modifying programs is easy
- Bottleneck on buses
- Pin count is reduced
- Ex. 8085, x86



Harvard

- The term originated from the Harvard Mark I relay-based computer, which stored instructions on punched tape and data in relay latches
- Separate address spaces for Program and Data memory
  - Instructions can be fetched simultaneously with instruction execution.
  - Buses could be of different sizes
  - Easier to pipeline
- Hardware realization is complex
- eg. 8048, PIC, DSPs



Little Vs. Big Endian

- Although numbers are displayed in the same way, they are not stored in the same way, in memory
- Big Endian processors store the most significant byte (MSB) of data in the lower memory address
- Little Endian machines on the other hand, store the least significant byte (LSB) byte of data in the lower memory address

A big Endian processor

Stores 0x12345678 as

addrs 0 - 0x12

addrs 1 - 0x34

addrs 2 – 0x56

addrs 3 – 0x78

A little Endian processor

stores 0x12345678 as

addrs 0 – 0x78

addrs 1 – 0x56

addrs 2 – 0x34

addrs 3 - 0x12

END