

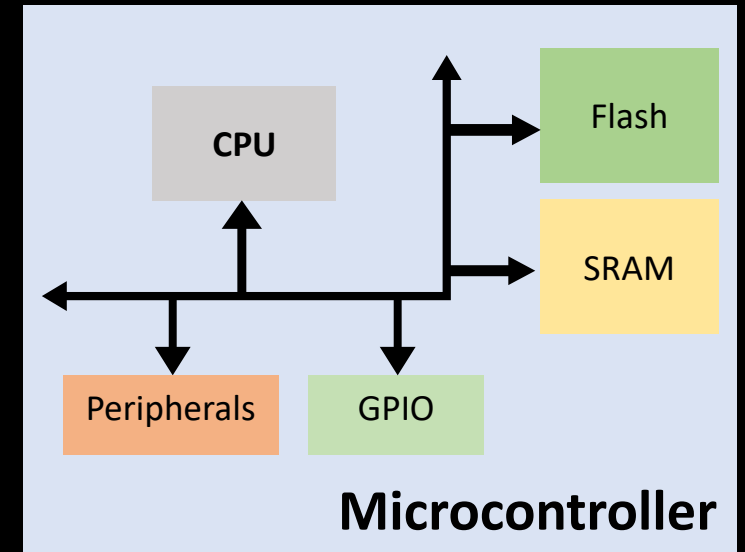
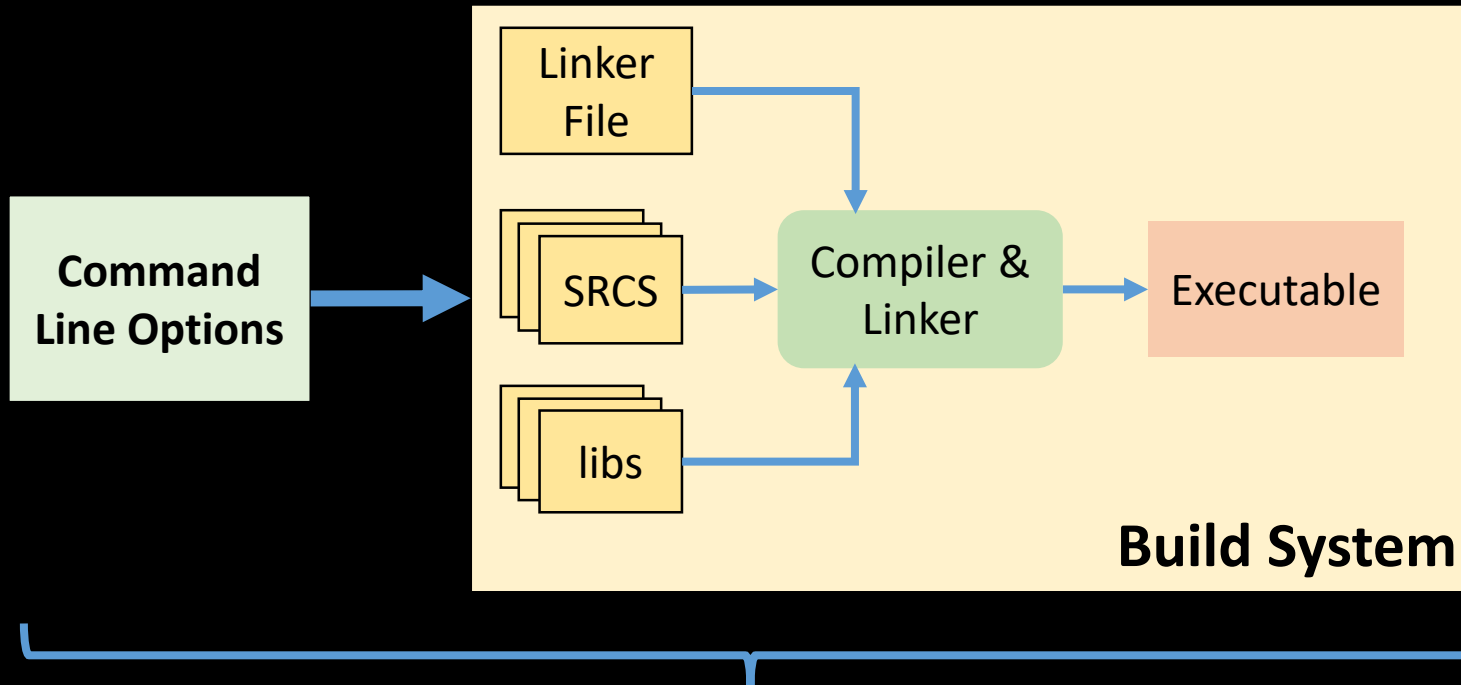
# Memory Architectures

Embedded Software Essentials

C1M3V2

# Programing Embedded Systems

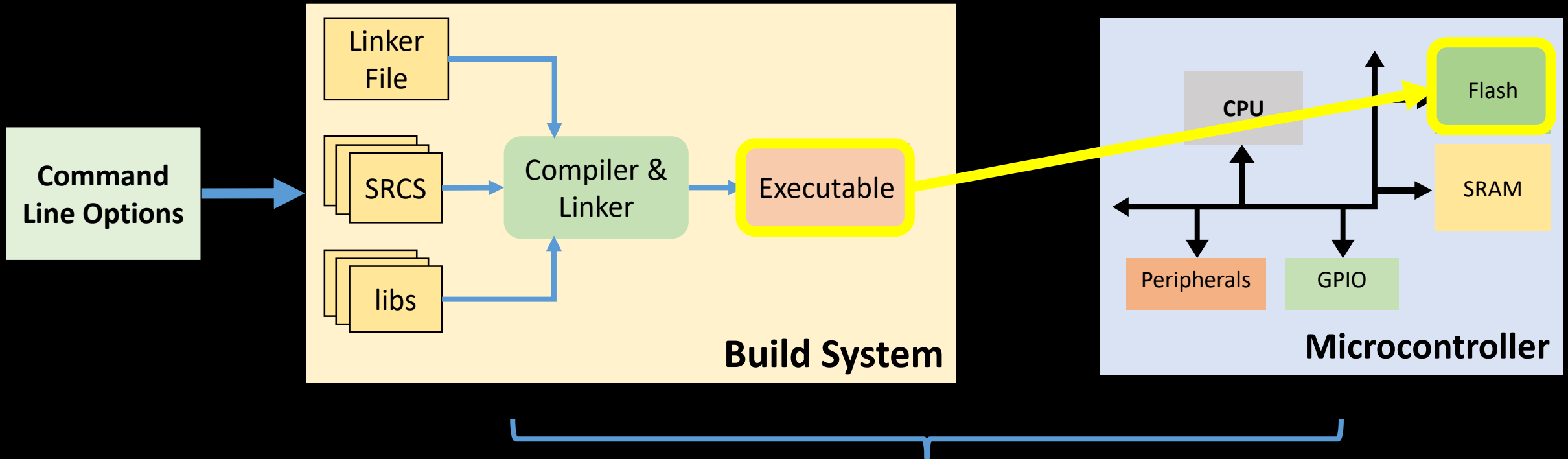
- Executable Program consists of **program code** and **program data** compiled for a particular **architecture** and **platform**



**Developing Code and Building an Executable is only  
Part of Design Process**

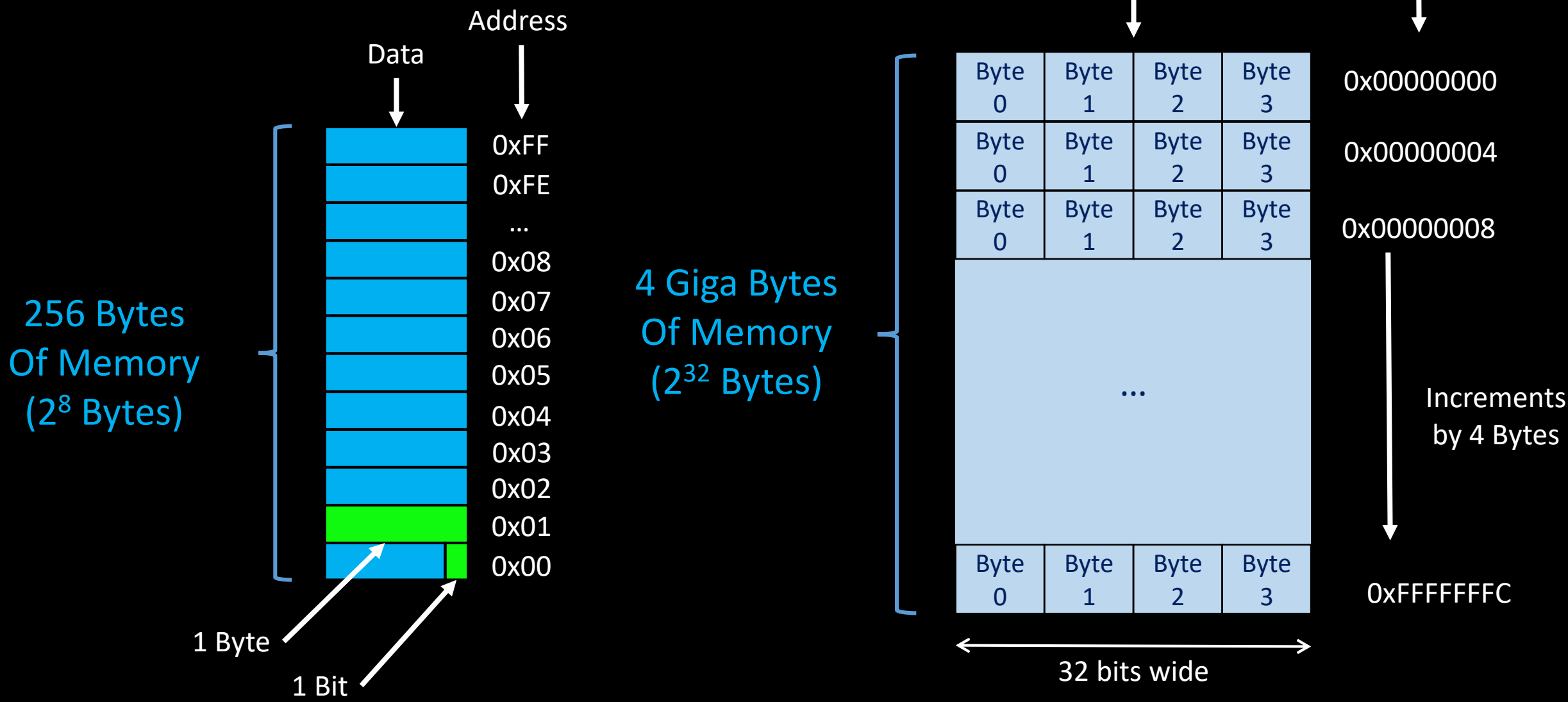
# Programing Embedded Systems

- Executable Program consists of **program code** and **program data** compiled for a particular **architecture** and **platform**



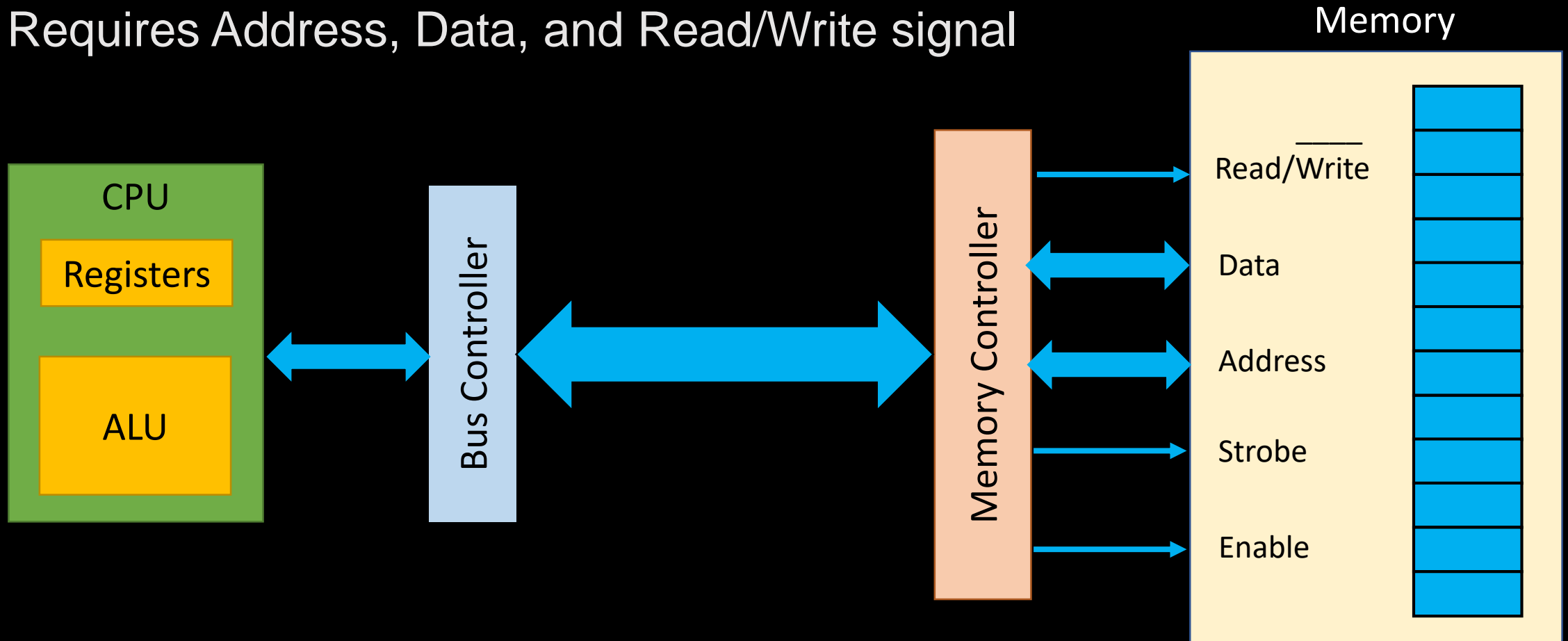
Installation and Testing is needed!!!

# Memory Models



# Memory Interfaces

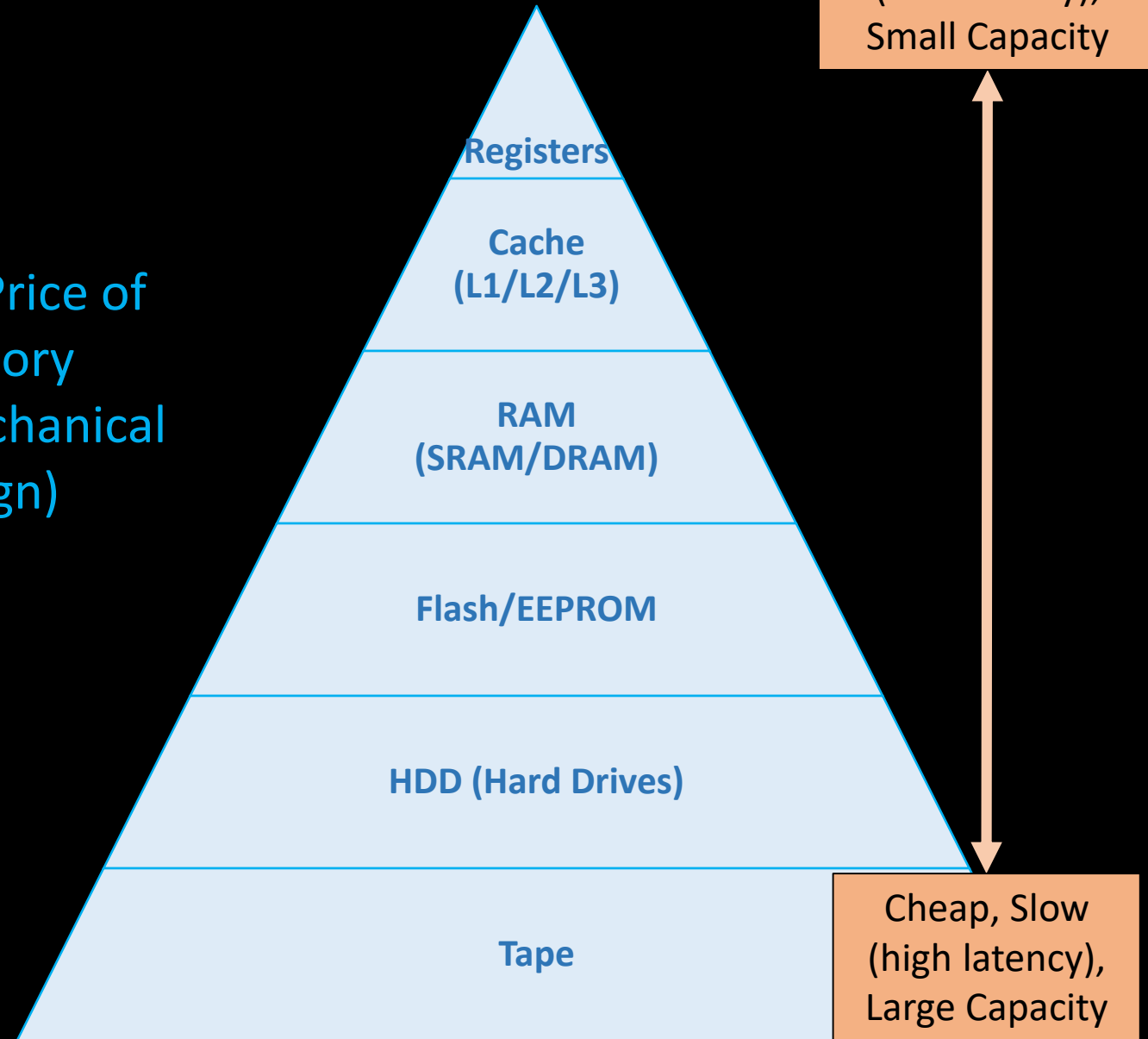
- CPU **Reads** or **Writes** data to Memory through Bus and Memory Controllers
  - Requires Address, Data, and Read/Write signal



# Memory Characteristics

- Capacity
- Volatility
- Access
- Power Consumption
- Latency
- Durability
- Transaction Size

Affects Price of  
Memory  
(And Mechanical  
design)



# Capacity

- **Capacity**: the amount of storage a memory can hold
  - Embedded Systems do **NOT** need a lot of memory, they need **optimized performance**

↑ **Capacity** ≠ ↑ **Performance**

- Increasing capacity increases the complexity of design and size
  - Physical size and connection circuitry (potentially)

You want to **LIMIT**  
size, power & cost of  
system

↑ **Capacity** = ↑ **Size** = ↑ **Power** = ↑ **Cost**

# Volatility

- **Volatility**: The ability for memory to hold data without power
  - Volatile Memory – Loses data when power removed
  - Non-Volatile – Retains data when power is removed

## Volatile Memories:

- SRAM
- DRAM
- SDRAM
- Register (most)

## Non-Volatile Memories:

- ROM/PROM/EPROM/EEPROM
- Flash
- Disk<sup>1</sup>
- Tape<sup>1</sup>

<sup>1</sup>Not used in modern design





**Many pros and cons to the different technologies**



# Volatility

- **Volatility**: The ability for memory to hold data without power
  - Volatile Memory – Loses data when power removed
  - Non-Volatile – Retains data when power is removed

## Non-Volatile Memories:

- ROM/PROM  Programmable only once
- EPROM/EEPROM/Flash  Exhibits endurance issues, limited number of write-erase cycles

# Write Endurance

- **Endurance**: Non-volatile semiconductor memories have a limited number of write-erase cycles before failure

Programmable Memory Evolution:

ROM → PROM → EPROM → EEPROM → Flash



Programmable ONLY once

# Write Endurance

- **Endurance**: Non-volatile semiconductor memories have a limited number of write-erase cycles before failure

Programmable Memory Evolution:

ROM → PROM → EPROM → EEPROM → Flash



Erasable (UV exposure)  
and Programmable Multiple  
times

# Write Endurance

- **Endurance**: Non-volatile semiconductor memories have a limited number of write-erase cycles before failure

Programmable Memory Evolution:



# Write Endurance

- **Endurance**: Non-volatile semiconductor memories have a limited number of write-erase cycles before failure

Programmable Memory Evolution:

ROM → PROM → EPROM → EEPROM → **Flash**

Flash has an Endurance  
lasting 10,000 to 100,000+  
Write-Erase Cycles

# Memory Access

- Random Access: Allows for access to any part of memory given the address of that location
  - Random Access Memory (RAM) – SRAM/DRAM
- Access Security: Require credentials to Read/Write parts of memory
  - Read-Only Memory (ROM) – No ability to write without extra permissions or process

# Memory Comparison

## SRAM

- Simple Read/Write Process
- No Access Security
- Byte level read/write
- Volatile
- No write/erase endurance issues

## Flash (as Read-Only Memory)

- Complex Read/Write process
- Secure Write Access
- Page level read/write
- Non-Volatile
- Endurance issues (Limited Write/erase cycles)

# Latency

- Latency: Time it takes for memory to respond to a read/write request
- Code and Data need to be read/written to memory
  - Assembly Instructions are read
  - Instruction operands need to be read
  - Instruction results need to be written
- Any data read/written interacts with the CPU, Bus Controller, Bus, memory controller, & memory

These read/write operations  
take a certain amount of time