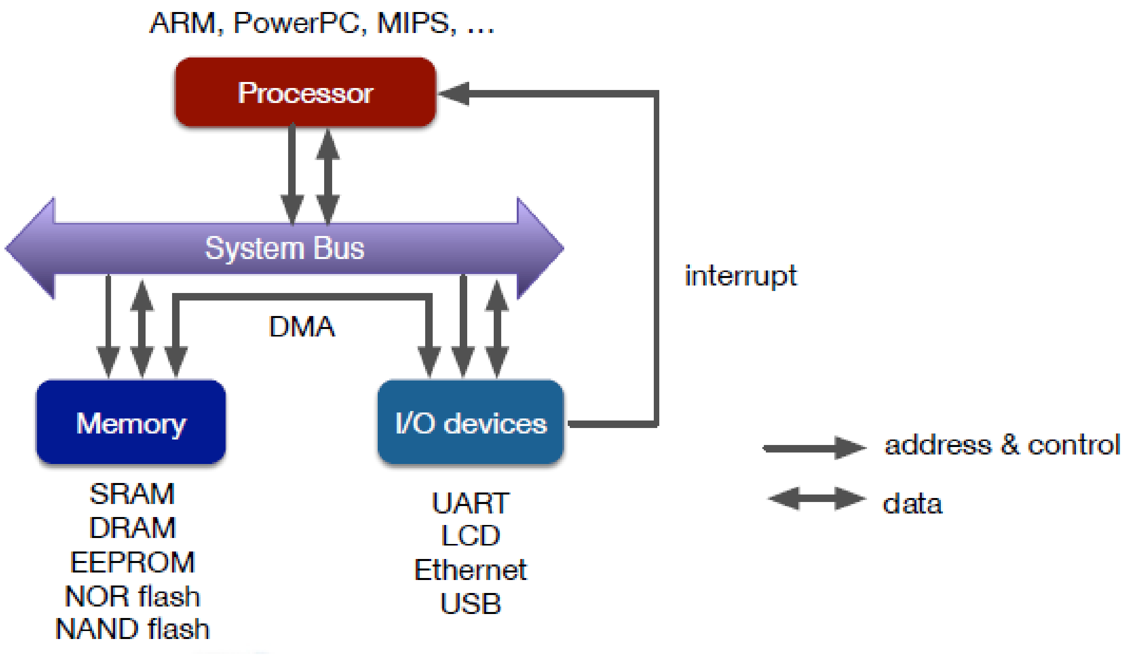
Memory Systems

Embedded HW Structure



Role

- Workspace for computer programs
- Permanent storage for programs and data

Abstraction of memory

- Virtually **flat** memory space
  - A single **logical address space** from an application perspective
  - **Assume** an ideal memory : read/write any location in 1 CPU clock

Implementation

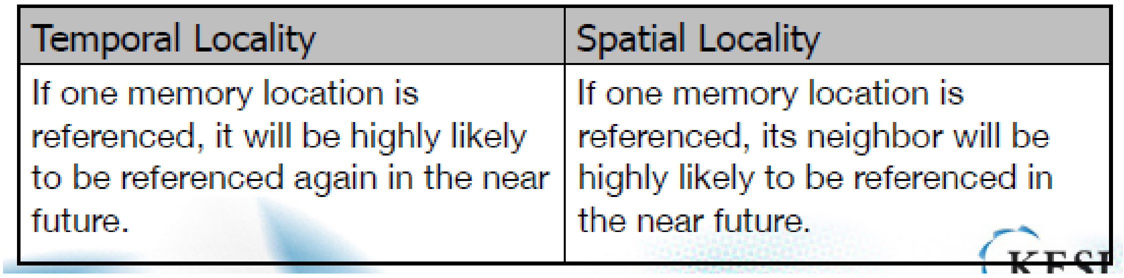
- **Impossible** to implement using a single type of memory device
- A well-implemented **"Memory Hierarchy"**
- Composed of a set of memory **subsystems**
  - Caches, DRAMs, disks, tapes, distribtued file systems, etc.

Goal

- Simultaneously achieve **speed & capacity**
  - Speed : to provide toe performance of the fastest component
  - Capacity : At a cost of the cheapest component

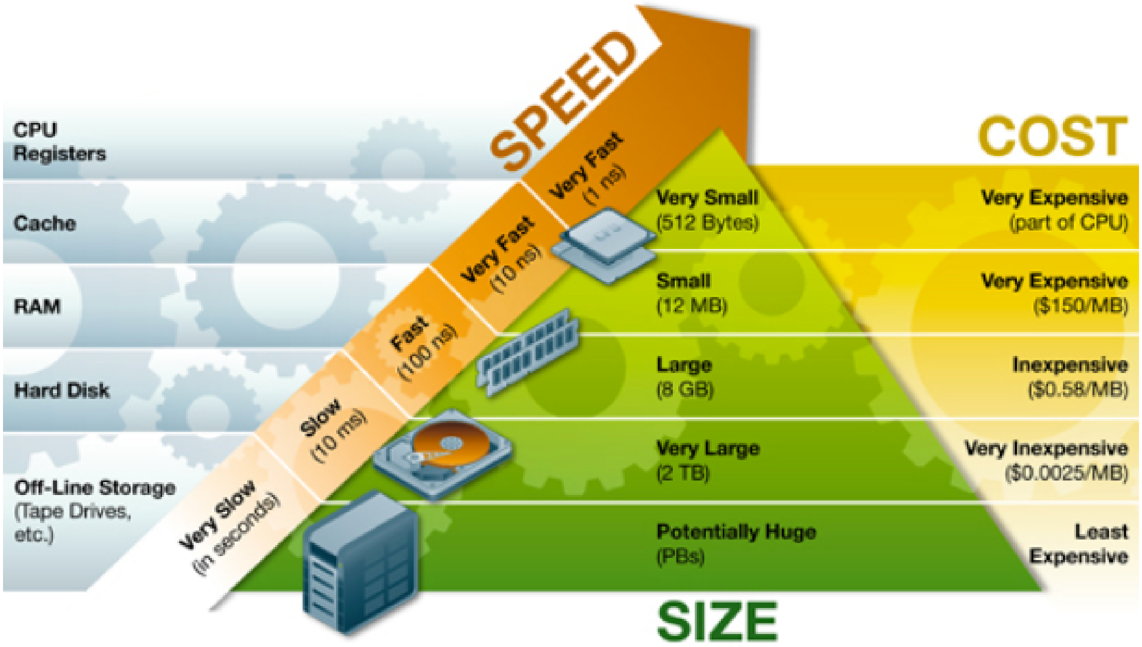
Locality of reference

- The key to memory hierarchy design

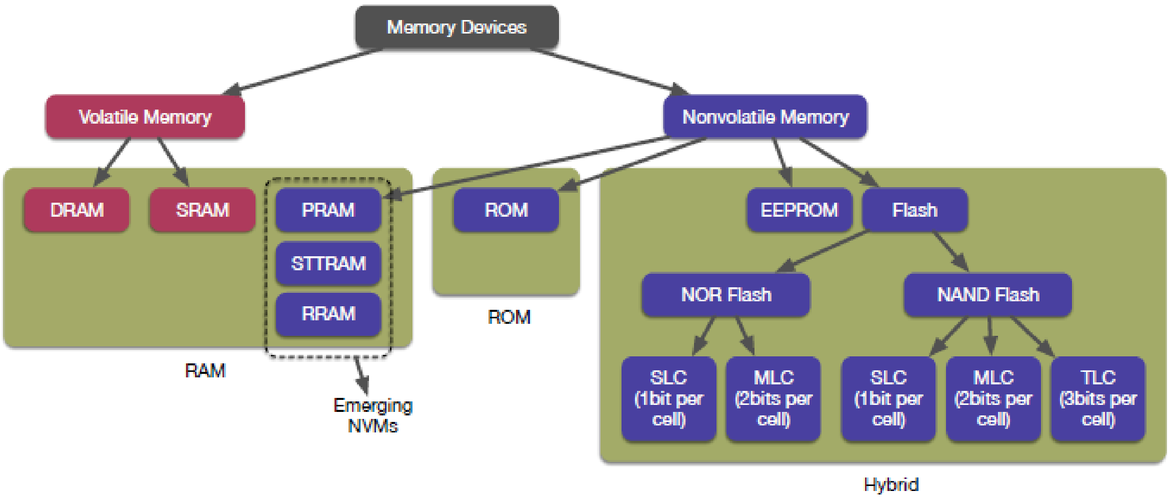


Temporal Locality	Spatial Locality
If one memory location is referenced, it will be highly likely to be referenced again in the near future.	If one memory location is referenced, its neighbor will be highly likely to be referenced in the near future.

Memory Hierarchy



Memory Taxonomy



SRAM(Static RAM)

- 4T or 6T Strutcure (T = Transistor)
- **Volatile** : Data is lost when not powered
- **Stable** : Holds value as long as power applied
- Very **fast** (~tens of ns) : Suitable for L1 and L2 caches
- Very **expensive** : Typical size range : tens of Kbits ~ several Mbits

DRAM(Dynamic RAM)

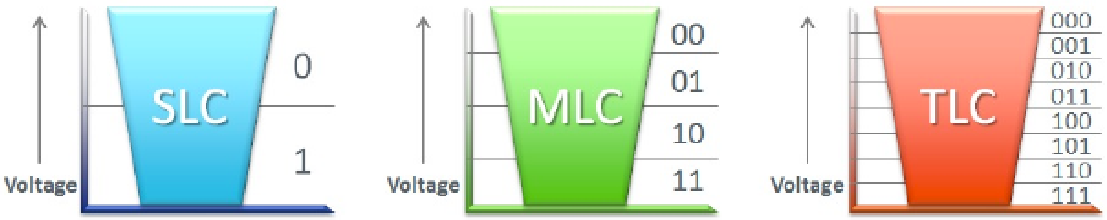
- 1T1C Structure (C = Capacitor) : The charge gradually leaks off
- **Volatile** : Data is lost when not powered
- **Unstable** : Data is lost without a periodic refresh
- **Destructive read** (charge consumed by read) : Must rewrite after read
- **Fast** (~hundreds of ns) : Suitable for main memory
- Very **Expensive** : Typical size range : several **Gbits**

Flash Memory

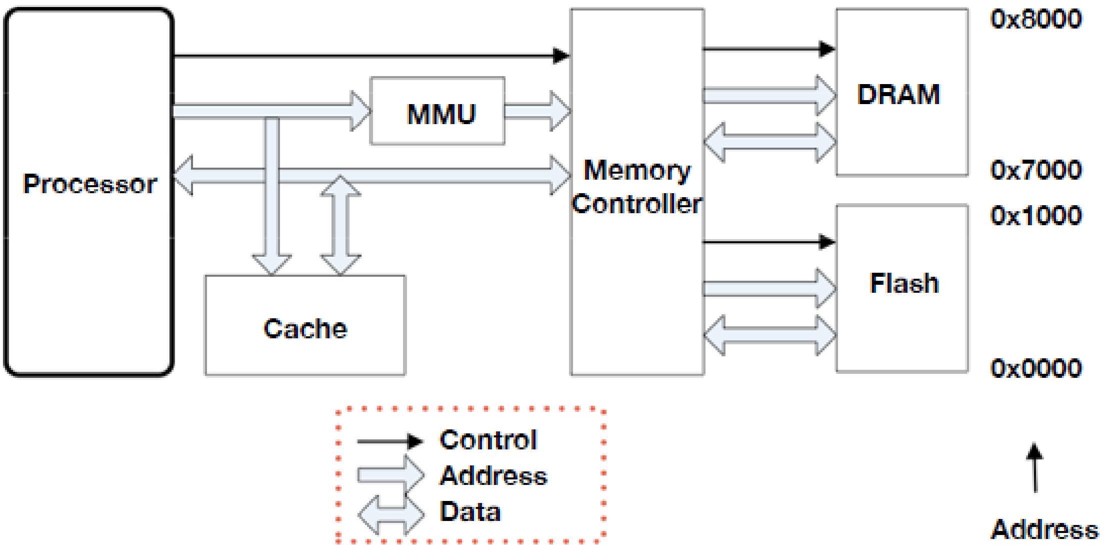
- **Operation principle**
  - Electrons trapped in the floating gate
- **Nonvolatile**
  - Electrons remain without power supply
  - Can be removed when high voltage supplied(erase operation)
- **Read/Write(Program) Unit**
  - Page (1KB ~ 4KB)
- **Erase Unit** : Block tens of pages
- **Out-place update**
  - Erase-before write constraint
  - Requires garbage collection

Multi-Level Cell Flash

- Store **two or more bits per cell**
  - Fine-grained control of the amount of charge to be stored
- Features
  - Lower **cost-per-bit**
  - Slower **program speed**
  - Slower **read speed**
  - Decreased **write endurance**
  - Decreased **data retention time**

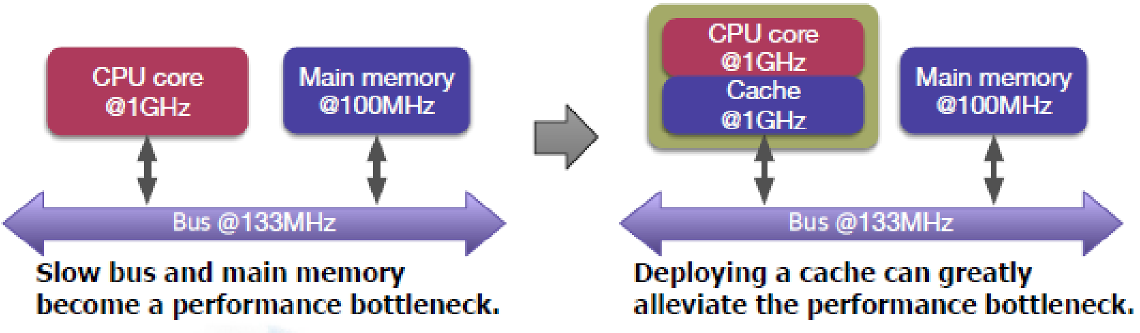


Memory System Structure



Cache Memory

- A **small size & high speed** memory
    - stores a part of frequently used data
- Most of memory accesses hit in the cache
    - Provide lower average memory access time
    - Exploit temporal & spatial localities


CPU core @1GHz
Main memory @100MHz
Bus @133MHz
Slow bus and main memory become a performance bottleneck.
CPU core @1GHz
Cache @1GHz
Main memory @100MHz
Bus @133MHz
Deploying a cache can greatly alleviate the performance bottleneck.

Cache Performance

- Design goal
    - supply instructions and data to the CPU as fast as possible
- Cache hit
    - requested data found in the cache
    - hit time : access latency for data in the cache
- Cache miss
    - requested data not found in the cache
    - miss penalty : elapsed time to fetch the missed cache line from main memory (line fill)
- Average access time
    - access time = hit time + miss rate * miss penalty

Cache Writing Policy

- Write Hit
    - Write Through
        - Write is done both to the cache and to the main memory

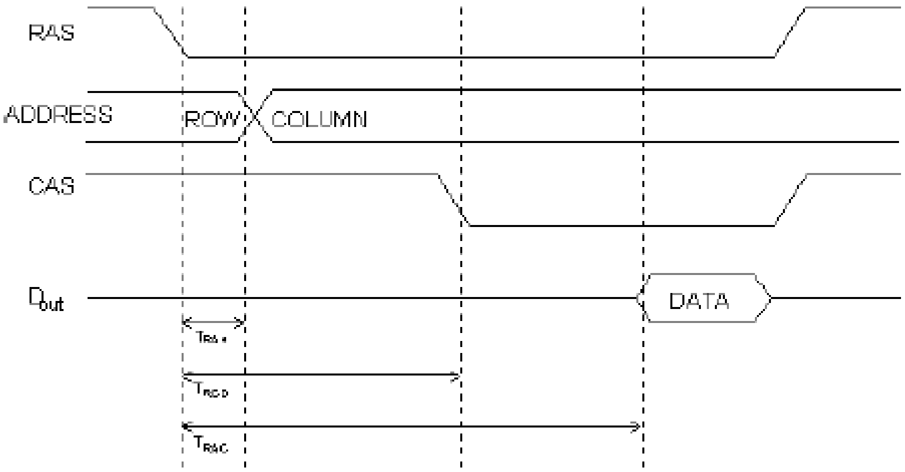
  - Write Back
    - Write is done only to the cache
    - Main memory is updated when the associated cache block is evicted from the cache
      (lazy write)
- Write Miss
  - Write Allocate
    - Missed cache block is first loaded to the cache, followed by a write-hit operation
  - No-wrtie Allocate
    - Missed cache block is not loaded to the cache
    - Data is directly written to the main memory

MMU(Memory Management Unit)

- Address Translation
  - Convert virtual memory addresses to physical memory addresses
- Memory Protection
  - To prevent a process from accessing memory that has not been allocated to it

Memory Controller

- Role
  - Manage the flow of data going to and from the main memory
- Functions
  - Address decode
    - Determine a target memory device
    - Assert a proper memory control signal (chip select)
  - Address multiplexing (for DRAM)
    - Split a given address into row and column address
    - Same address pins used to feed both ROW and COL addresses
  - Refresh operation (for DRAM)
    - Periodic read and write of every memory cells



I/O Devices

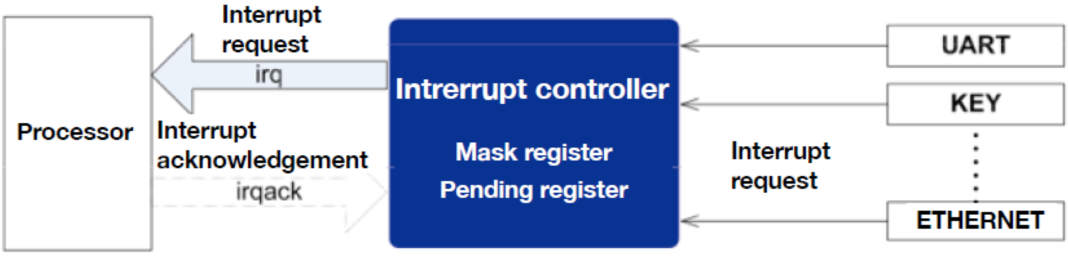
- Similar to memory devices
  - use of address bus, data bus, and control signals
- I/O-mapped I/O (or port-mapped I/O)
  - Dedicated address space for I/O devices
  - ex) intel X86
- Memory-mapped I/O
  - The same address bus for both memory and I/O devices
  - Typically used for most embedded processors
  - Caution
    - I/O memory region should be configured to be "non-cacheable"
    - I/O variables should be declared as "volatile"

I/O Resource Management

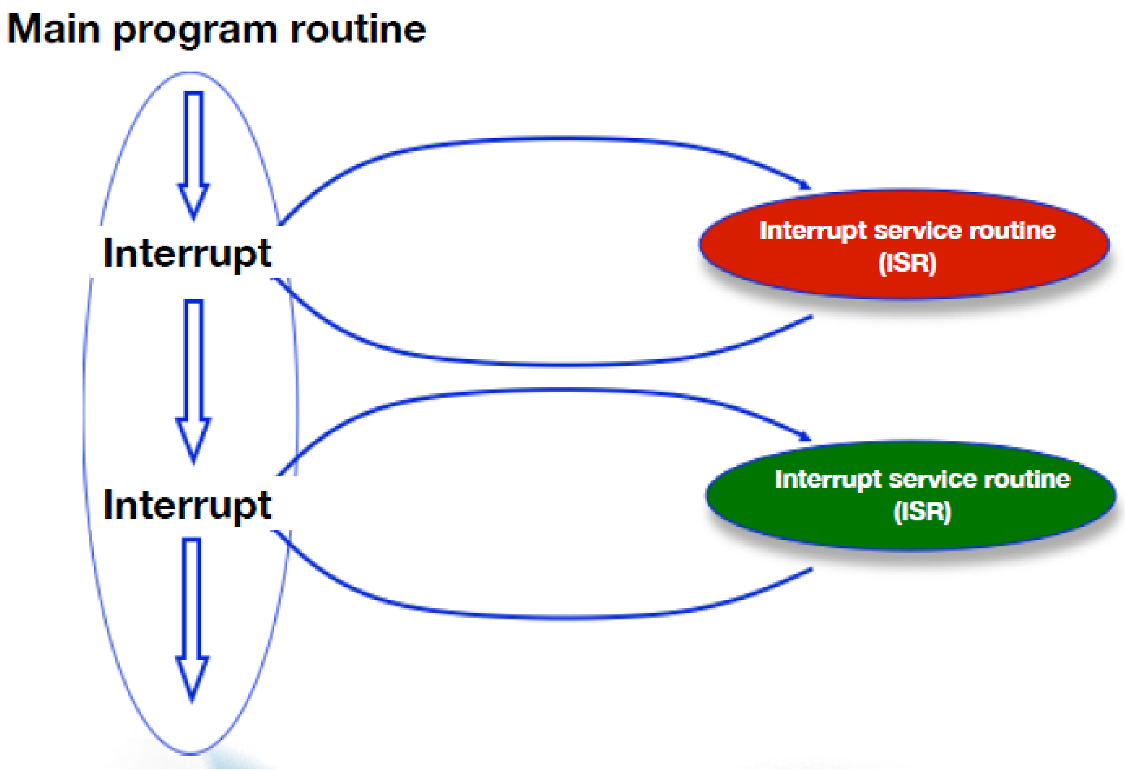
- Polling
  - Continuously check the readiness of an I/O device in a synchronous manner
  - CPU does nothing other than waiting (busy-wait)
- Interrupt
  - Allow an I/O device to send a signal for its readiness
  - CPU can run other processes
- DMA (Direct Memory Access)
  - Data transfer between I/O devices and memory without intervention of the CPU

Interrupt Interface

- Interrupt controller
  - manage interrupt requests from I/O devices



Interrupt Flow Control

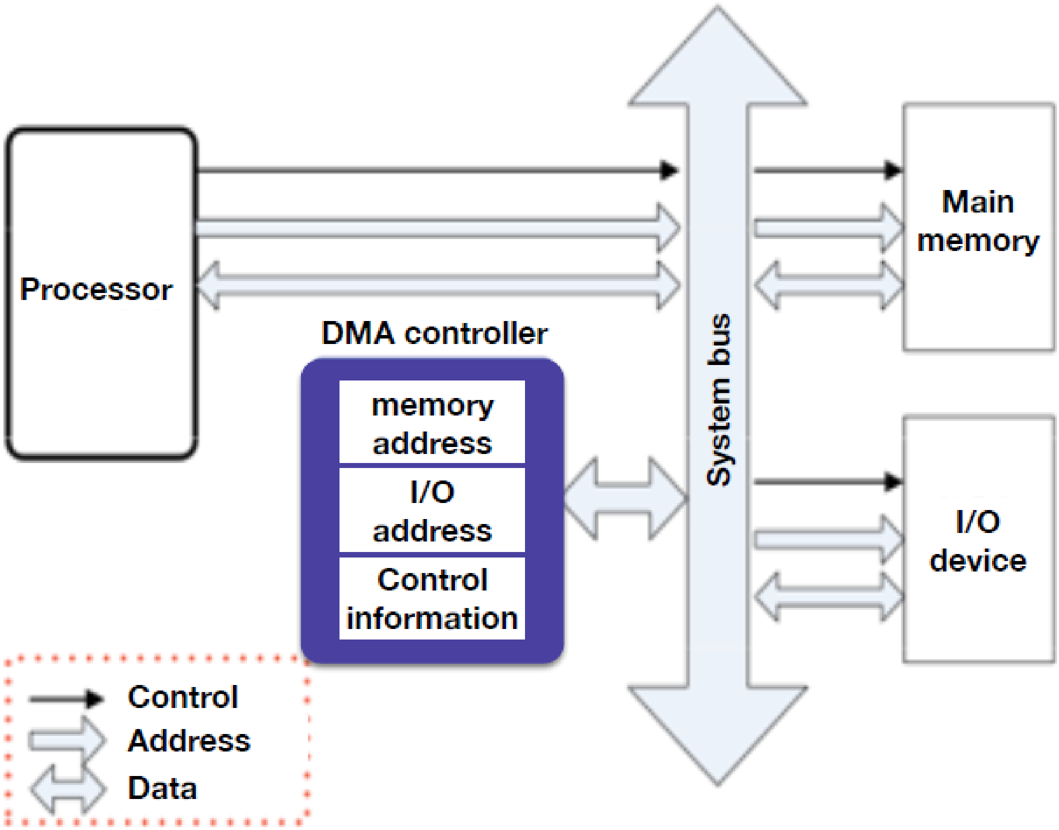

Main program routine
Interrupt
Interrupt
Interrupt service routine (ISR)
Interrupt service routine (ISR)

Interrupt Vector

- Interrupt vector table
  - memory space for storing information to associate an interrupt handler function to an interrupt request
- Fixed Interrupt
  - the address for interrupt handler is fixed
- Vectored Interrupt
  - The address for interrupt handler is configurable
  - Can be set by I/O devices

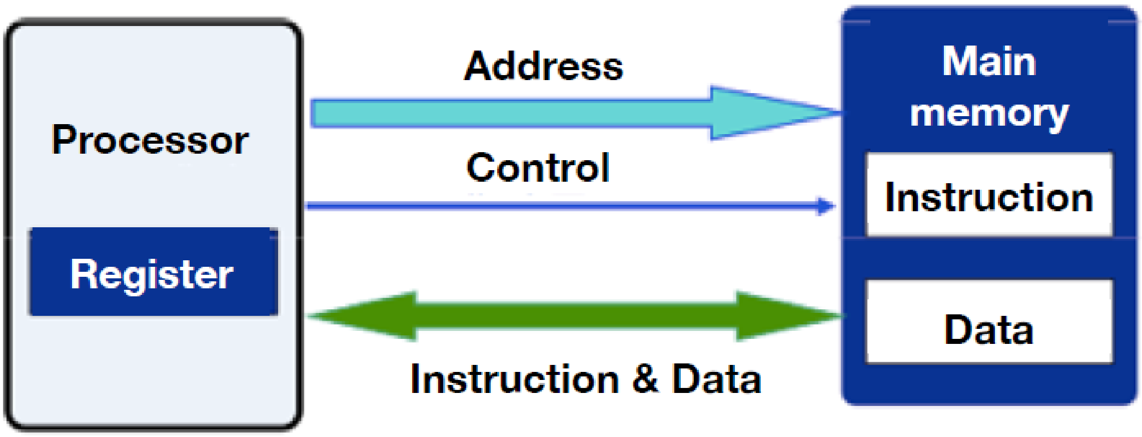
DMA (Direct Memory Access)

- Data can be transferred between I/O devices and memory without intervention of the CPU



Bus

- A set of signals shared by the components in a computer system
- Used to transfer address, data, and control information between components
- Consist of address bus, data bus, and control bus



Processor
Register
Address
Control
Instruction & Data
Main memory
Instruction
Data