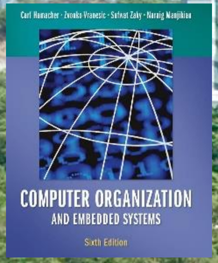
香港中文大學
The Chinese University of Hong Kong

CSCI2510 Computer Organization

Lecture 13: Basic Input & Output

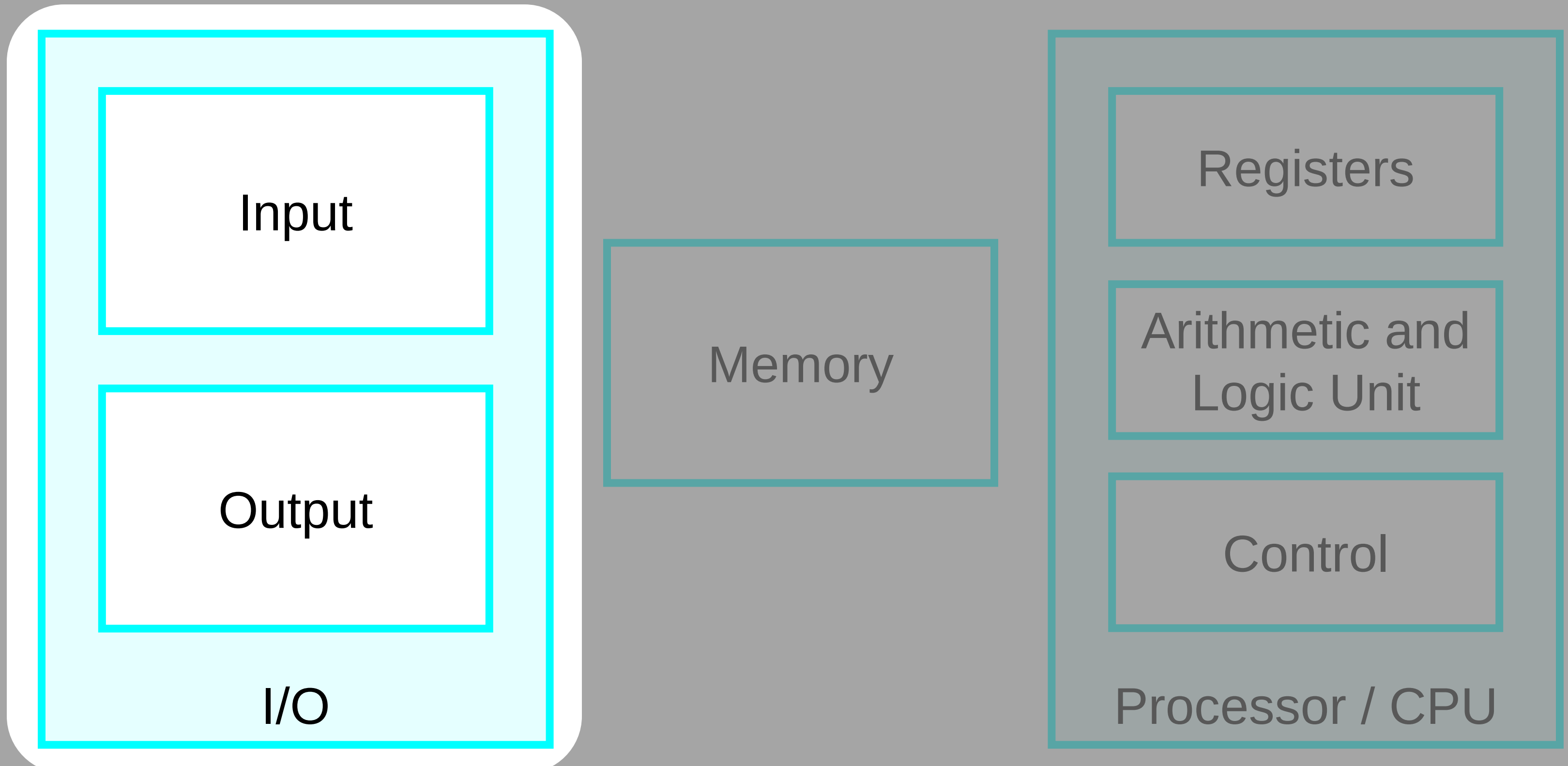
Ming-Chang YANG

mcyang@cse.cuhk.edu.hk



Reading: Chap. 3

Basic Functional Units of a Computer



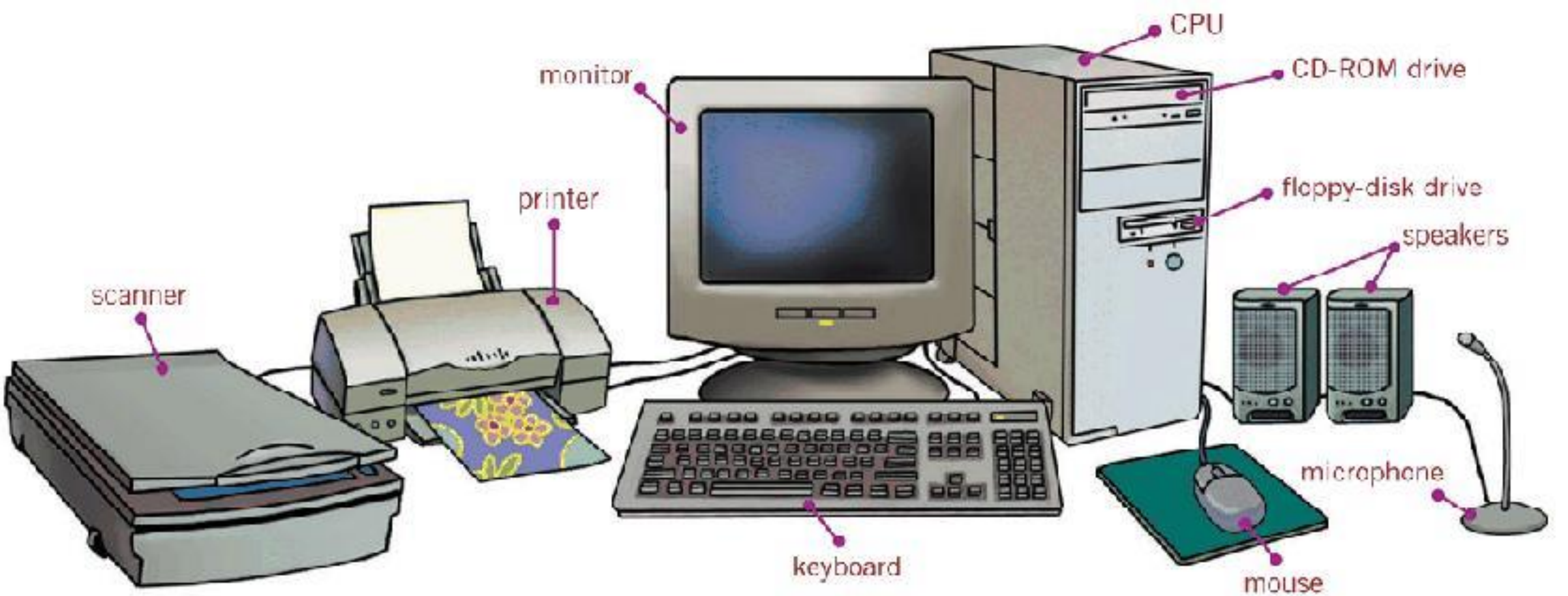
- **Input**: accepts coded information from human operators.
- **Memory**: stores the received information for later use.
- **Processor**: executes the instructions of a program stored in the memory.
- **Output**: sends back to the outside world.
- **Control**: coordinates all of these actions.

Outline

- Accessing I/O Devices
  - Memory-Mapped I/O
  - I/O Device Interface
  - Program-Controlled I/O
  - Interrupts
- Storage I/O
  - Hard Disk Drive (HDD)
  - Solid State Drive (SSD)

Input and Output Units (I/O)

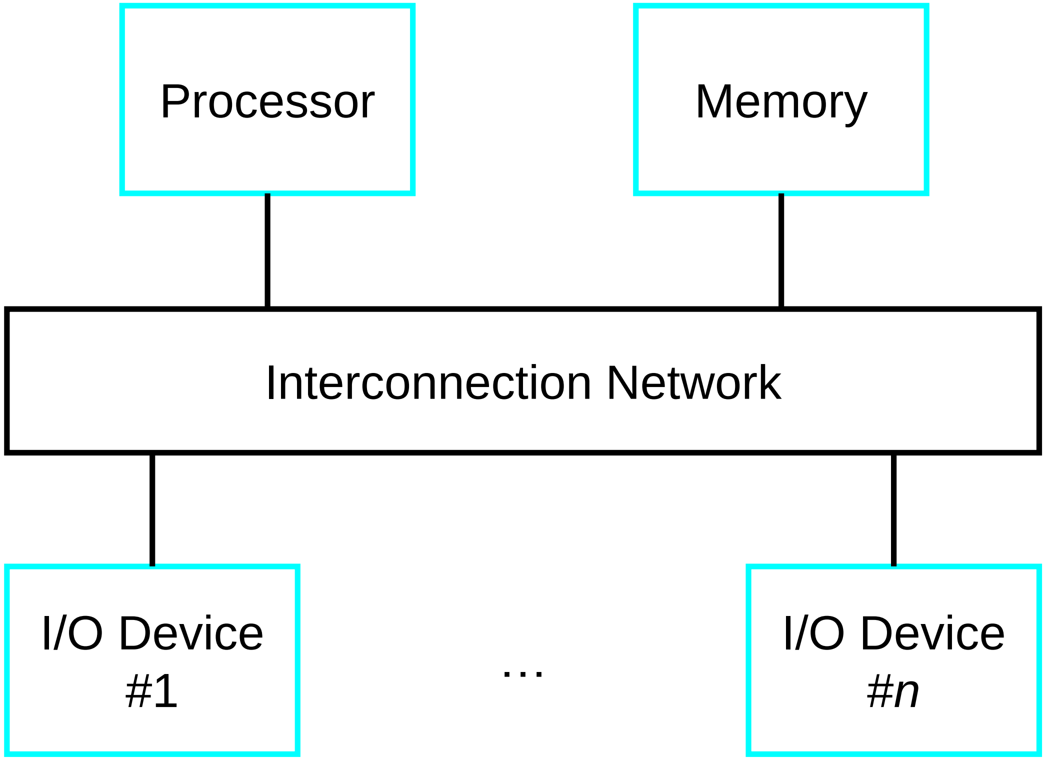
- Computers should have the ability to exchange digital and analog information with a wide range of devices.
- The collective term input/output (I/O) units: input units, output units, disk drives, etc.



https://norizman.wordpress.com/notes/

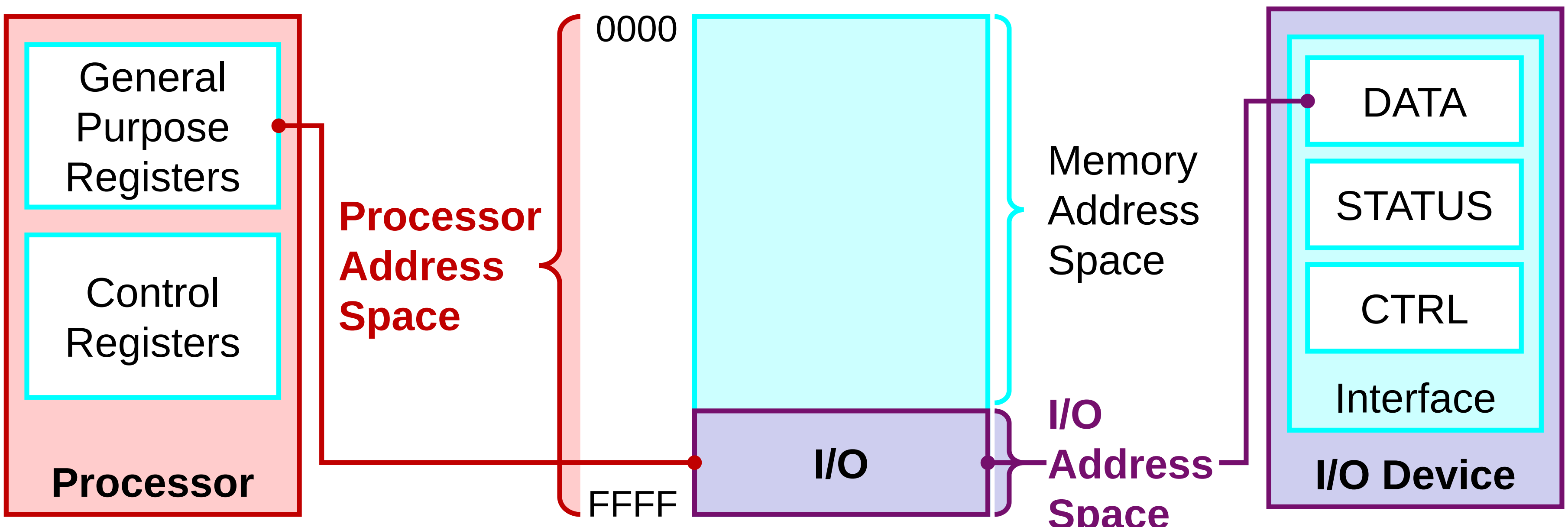
Accessing I/O Devices

- The components of a computer system communicate with each other through an interconnection network.
  - The network enables the information transfer between the processor, the memory unit, and a number of I/O devices.



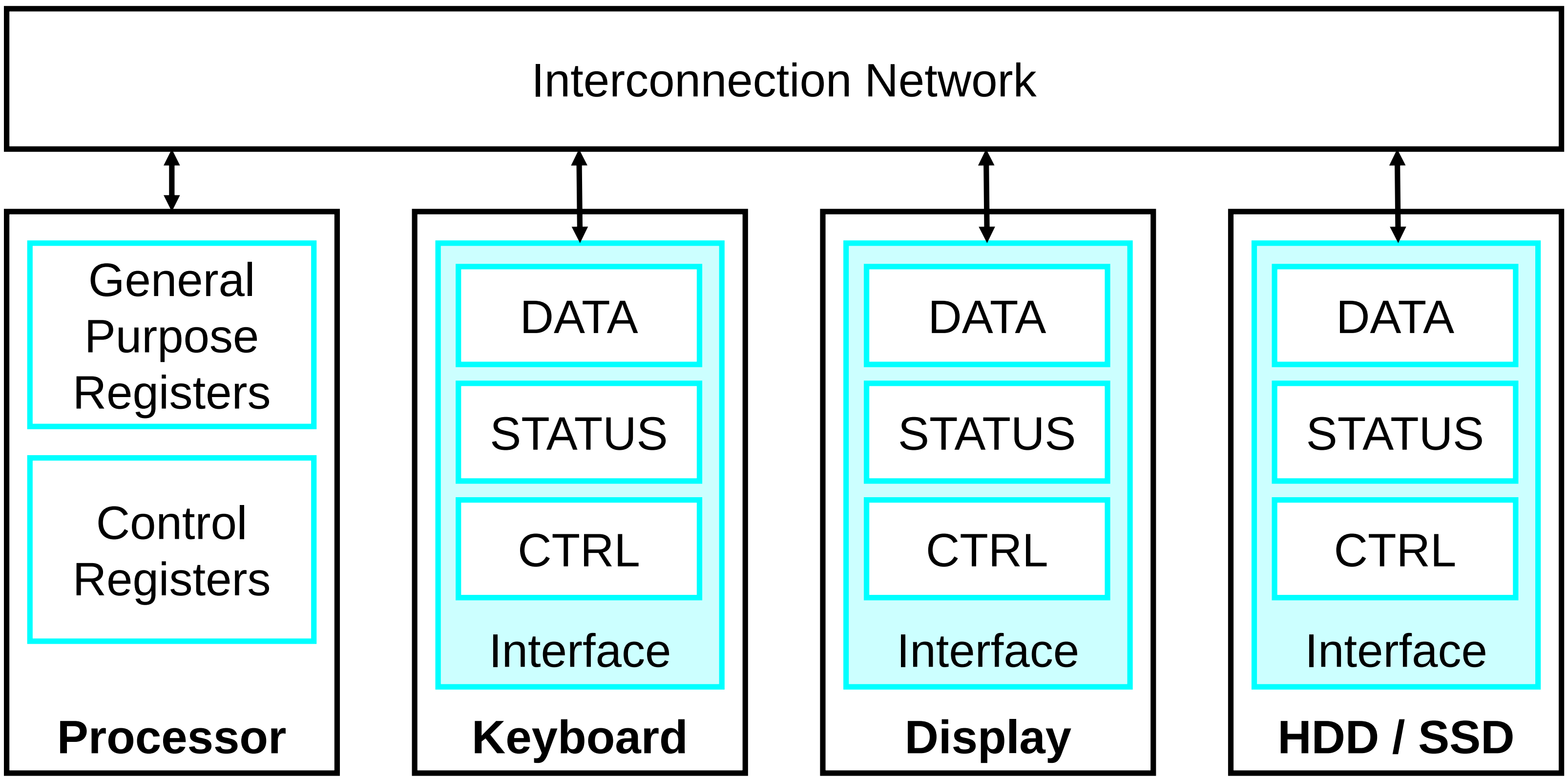
Memory-Mapped I/O

- The idea of using addresses to access (i.e. load/store) memory can be extended to deal with the I/O devices.
  - I/O devices consist of addressable locations, like memory.
  - E.g., `Load R2, DATA`, `Store R2, DATA`.
- **Memory-Mapped I/O**: I/O devices and the memory share the same address space of the processor.



I/O Device Interface

- An I/O device is connected to the interconnection network via the device interface.
  - The interface has some registers, accessible by the CPU, for data transfer, exchange of status, and control.



Program-Controlled I/O

- Let us begin with two most essential I/O devices for human-computer interaction—keyboard and display.
  - *Consider a task that reads characters typed on a keyboard, stores these data in the memory, and displays the same characters on a display screen.*
    - A processor executes billions of instructions per second.
    - Characters can be transmitted to and displayed on the display, typically several thousand characters per second.
    - The typing speed of the user is few characters per second.

- **Program-Controlled I/O**: Use a program to perform all functions needed to realize the desired action.
  - The speed difference in speed between the CPU and I/O devices creates need to be synchronized.

An Example of a RISC-Style I/O Program

- The program reads, stores, and displays a line of characters typed at the keyboard.

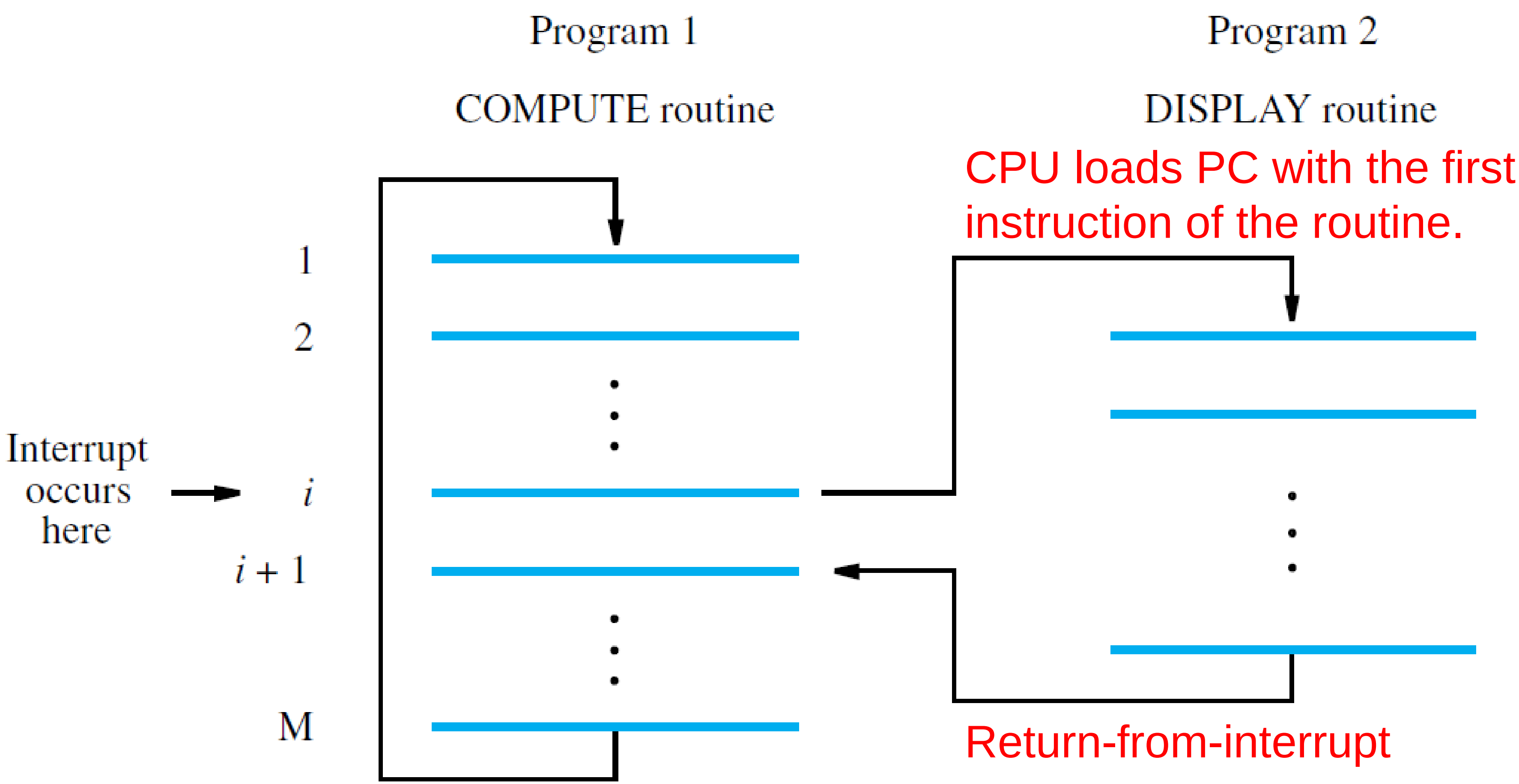
	Move	R2, #LOC	Initialize pointer register R2 to point to the address of the first location in main memory where the characters are to be stored.
	MoveByte	R3, #CR	Load ASCII code for Carriage Return into R3.
READ:	LoadByte	R4, KBD_STATUS	Wait for a character to be entered.
	And	R4, R4, #2	Check the KIN flag.
	Branch_if_[R4]=0	READ	
	LoadByte	R5, KBD_DATA	Read the character from KBD_DATA (this clears KIN to 0).
	StoreByte	R5, (R2)	Write the character into the main memory and
	Add	R2, R2, #1	increment the pointer to main memory.
ECHO:	LoadByte	R4, DISP_STATUS	Wait for the display to become ready.
	And	R4, R4, #4	Check the DOUT flag.
	Branch_if_[R4]=0	ECHO	
	StoreByte	R5, DISP_DATA	Move the character just read to the display buffer register (this clears DOUT to 0).
	Branch_if_[R5]≠[R3]	READ	Check if the character just read is the Carriage Return. If it is not, then branch back and read another character.

Interrupts (1/2)

- The previous program enters a wait loop in which it repeatedly tests the device status.
  - During this period, the processor is not performing any useful computation.
  - There are many situations where other tasks can be performed while waiting for an I/O device to become ready.
- To allow this to happen, we can arrange for the I/O device to alert the processor when it becomes ready.
  - It can do so by sending a hardware signal called an interrupt request to the processor.
  - The routine executed in response to an interrupt request is called the interrupt-service routine.

Interrupts (2/2)

- Assume an interrupt arrives during instruction *i* of the COMPUTE routine, and the DISPLAY routine is the interrupt-service routine.



Outline

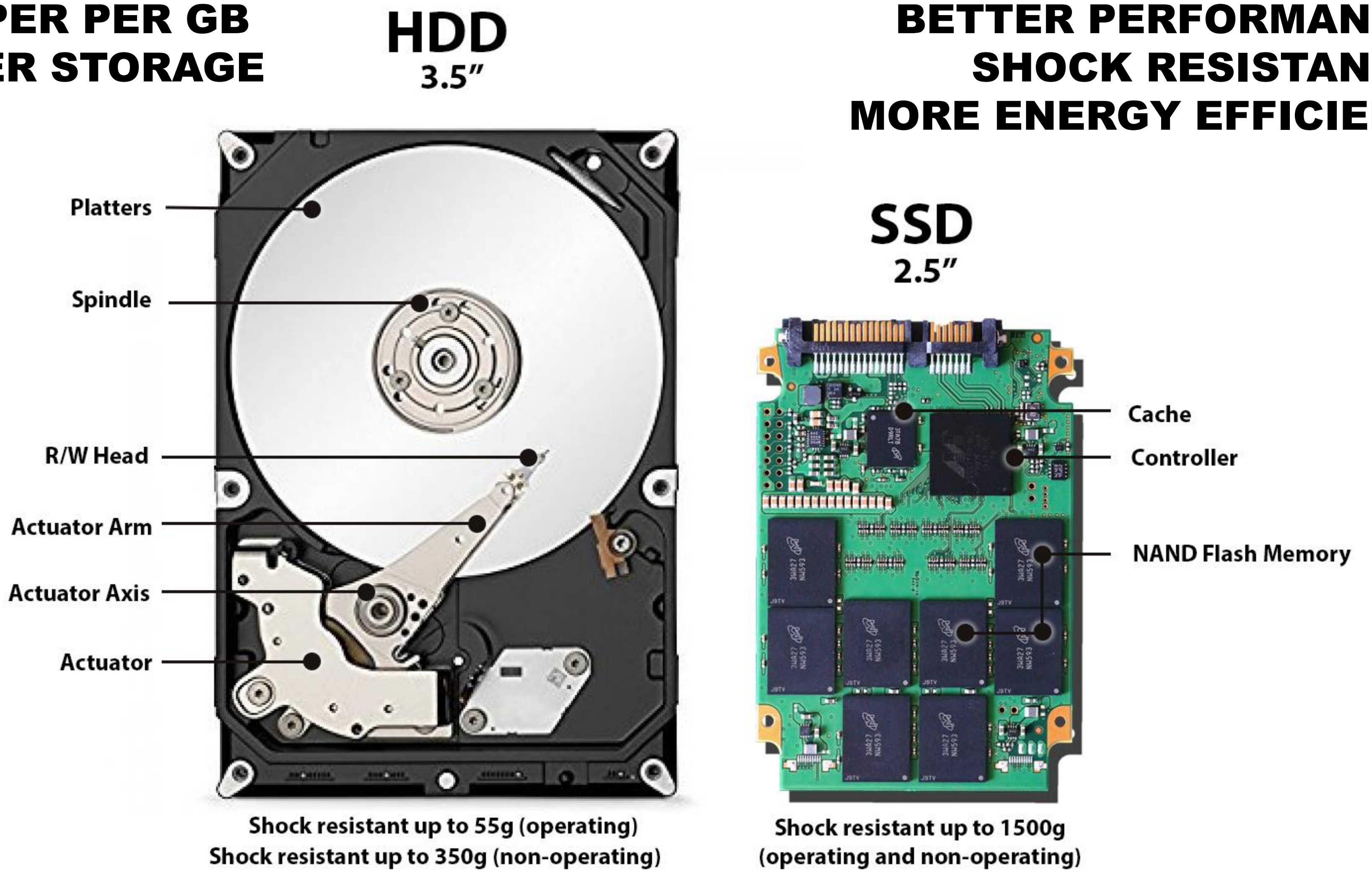
- Accessing I/O Devices
  - Memory-Mapped I/O
  - I/O Device Interface
  - Program-Controlled I/O
  - Interrupts
- Storage I/O
  - Hard Disk Drive (HDD)
  - Solid State Drive (SSD)

Storage I/O

- The most common two types: HDD and SSD

CHEAPER PER GB
LARGER STORAGE

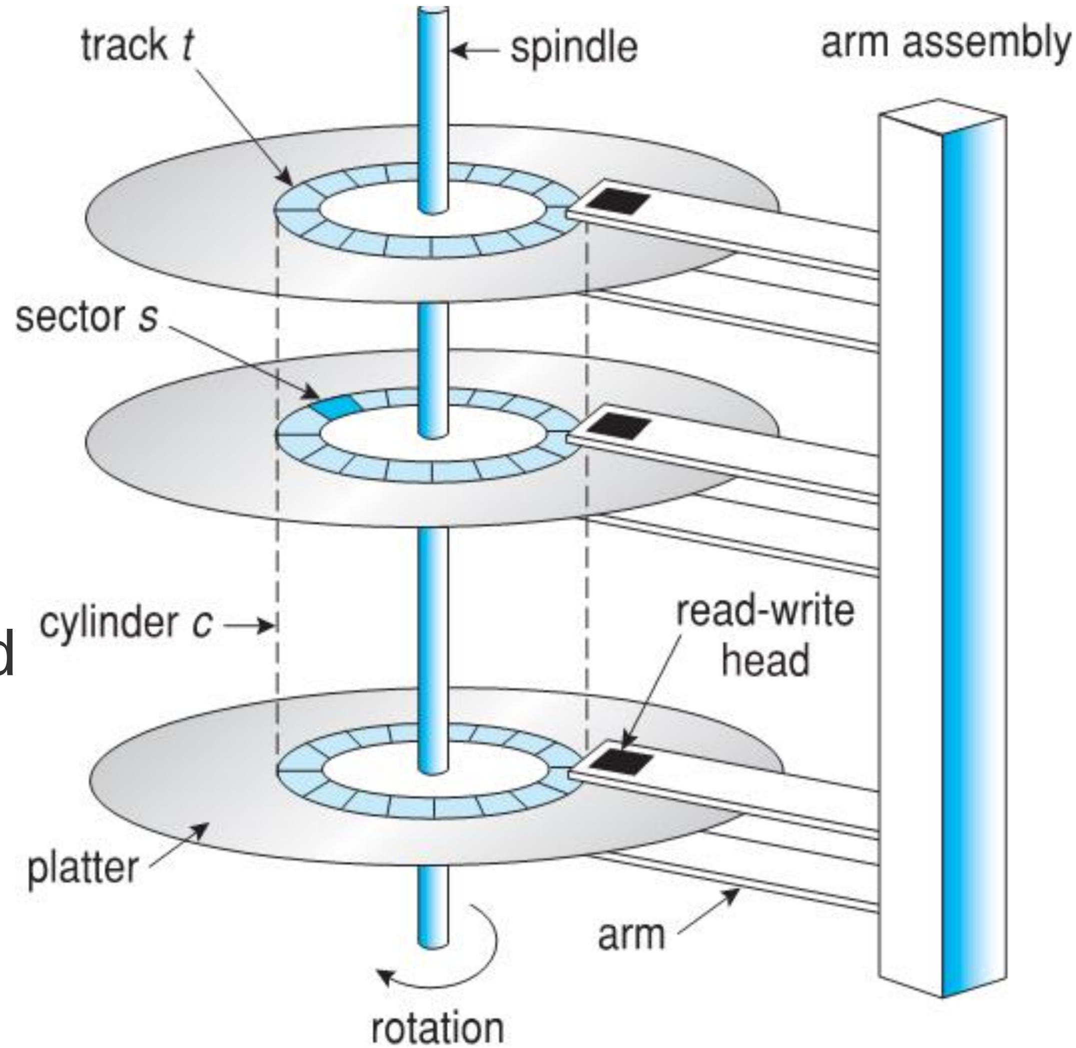
BETTER PERFORMANCE
SHOCK RESISTANCE
MORE ENERGY EFFICIENT



https://www.backblaze.com/blog/ssd-vs-hdd-future-of-storage/

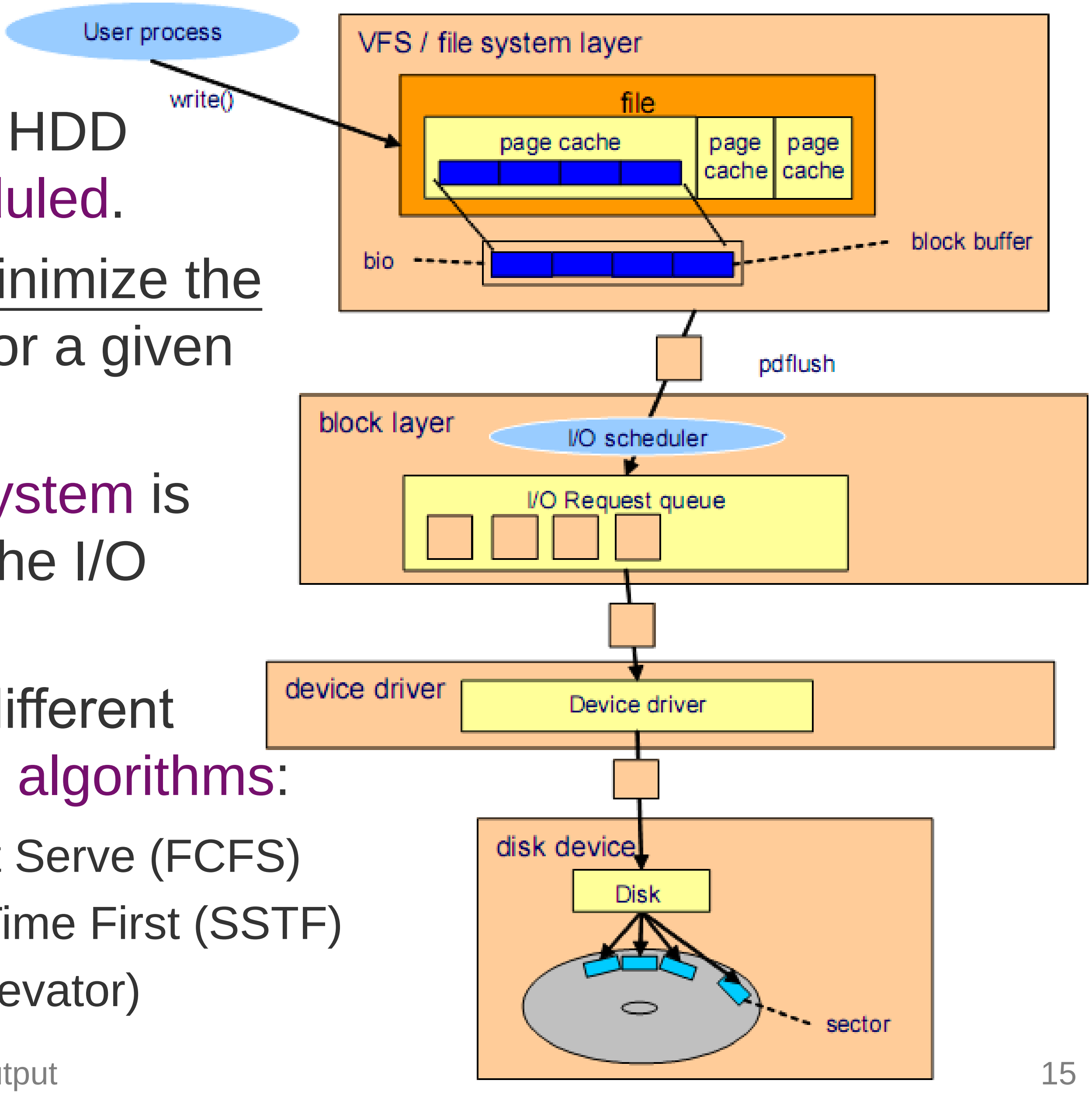
Hard Disk Drive (HDD)

- HDD provides bulk of storage for modern computers.
- Digital data can be stored in any sector s of any track t on any disk platter p.
- **HDD Seeking Time**:
  - Time to move disk arm to desired cylinder, and
  - Time to rotate the disk head to the sector.
    - Platter rotates at 60 to 250 times per second.



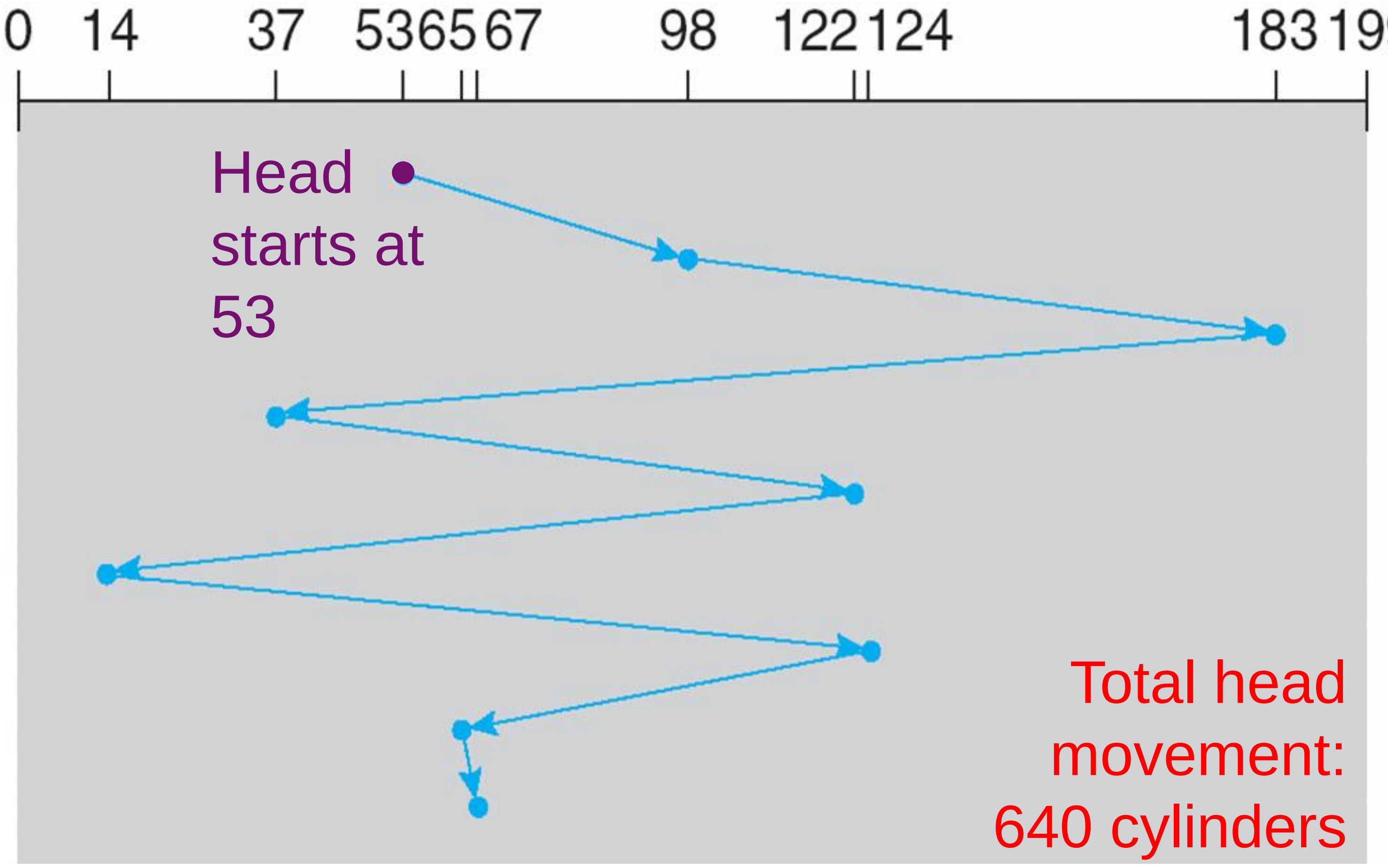
I/O Scheduling for HDDs

- I/O Scheduling
  - Accesses to the HDD should be scheduled.
  - The goal is to minimize the total seek time for a given set of accesses.
  - The operating system is responsible for the I/O scheduling.
  - There're many different HDD scheduling algorithms:
    - First Come First Serve (FCFS)
    - Shortest Seek Time First (SSTF)
    - SCAN (a.k.a. Elevator)



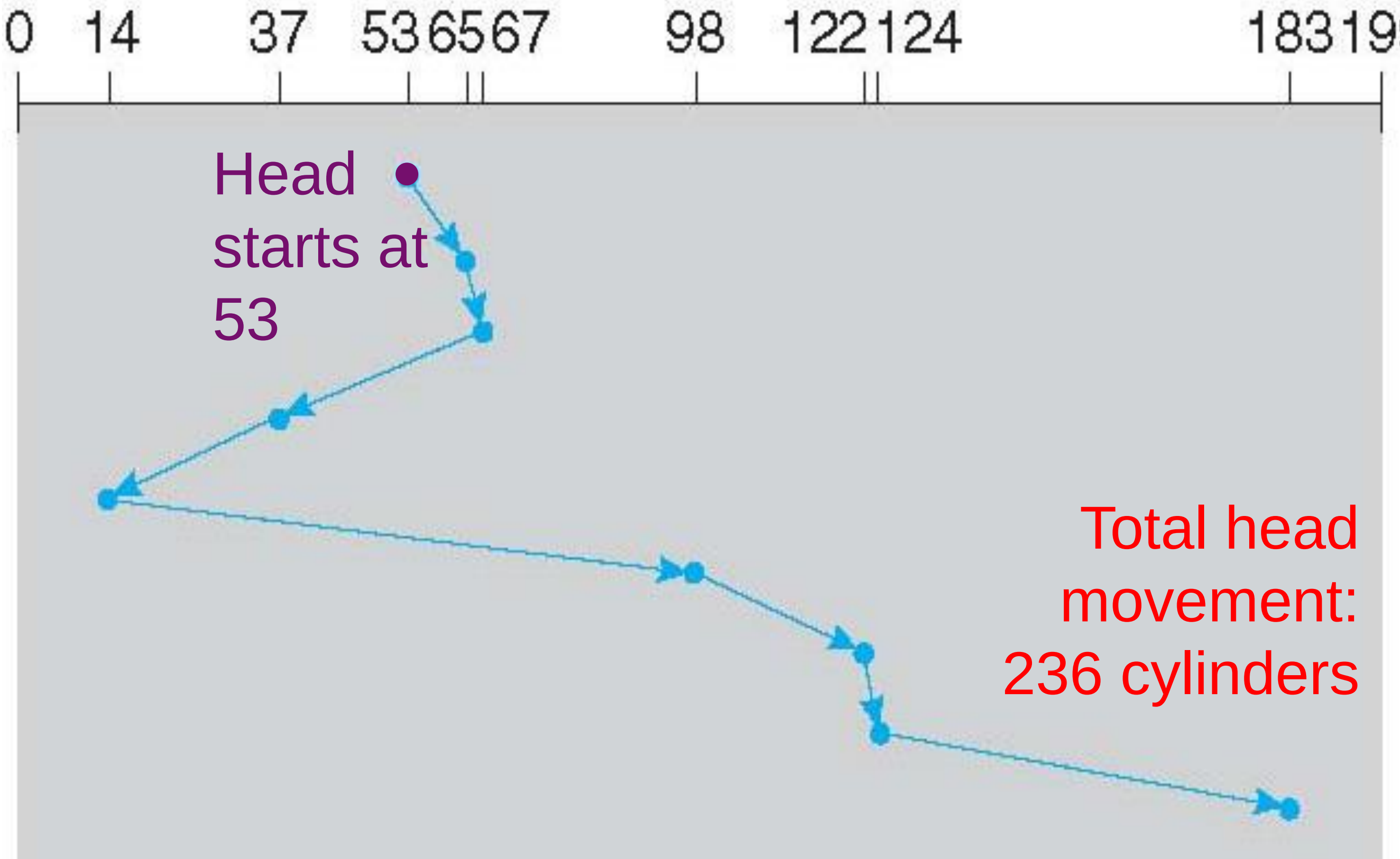
First Come First Serve (FCFS) Algo.

- **FCFS** serves accesses in order.
  – Given a set of accesses: 98, 183, 37, 122, 14, 124, 65, 67



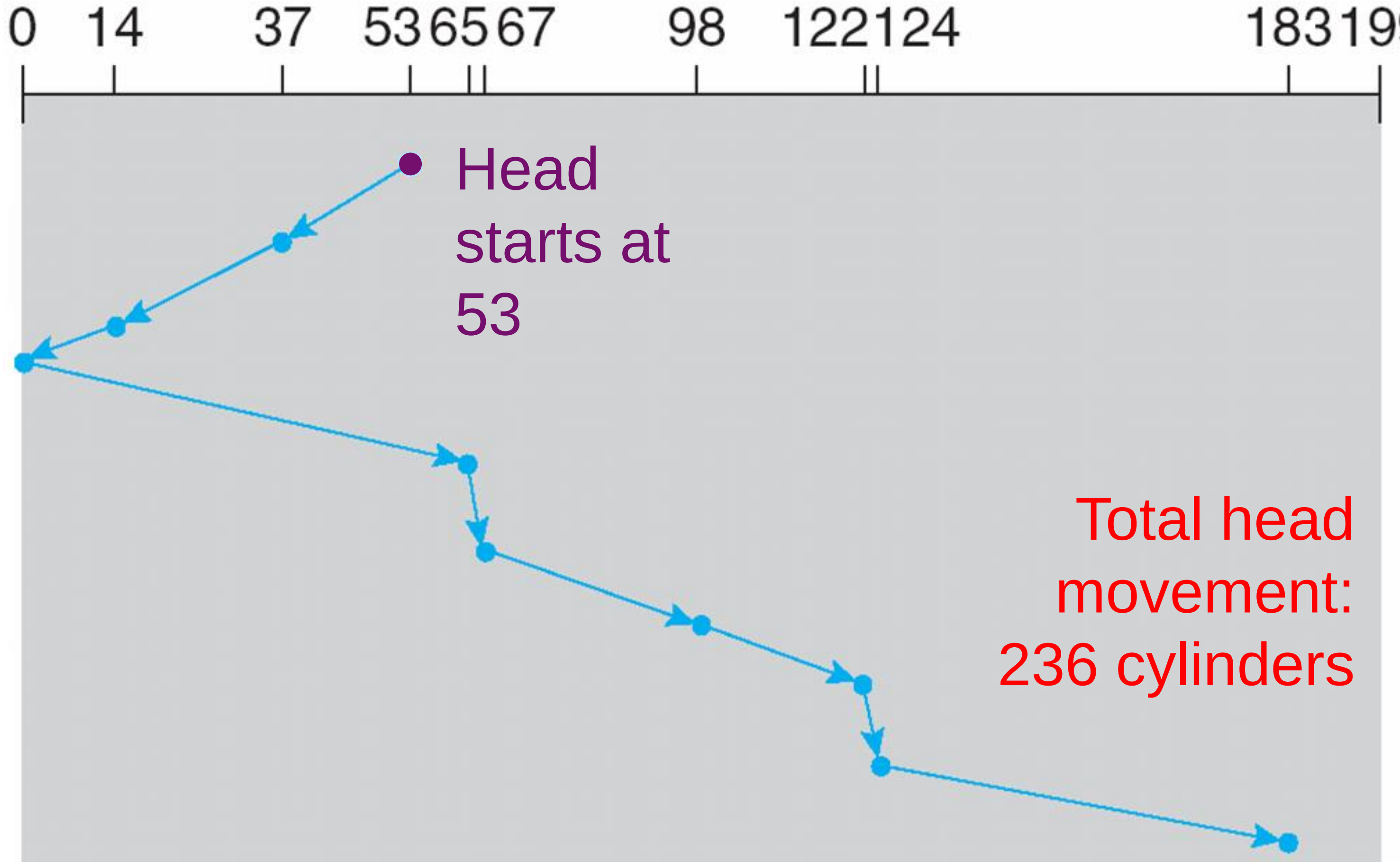
Shortest Seek Time First (SSTF) Algo.

- **SSTF** selects the request with the minimum seek time from the current head position.
- Given accesses: 98, 183, 37, 122, 14, 124, 65, 67



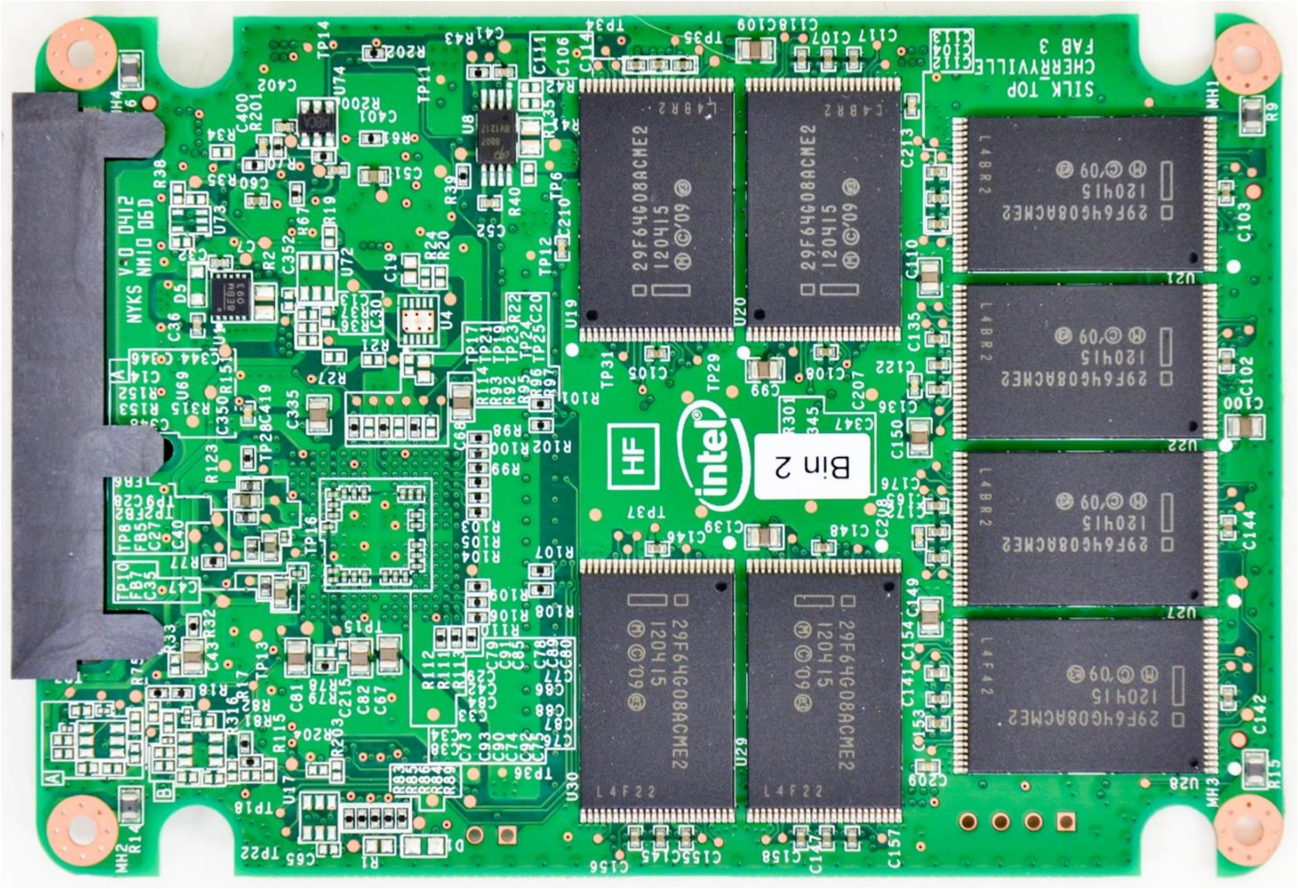
SCAN/Elevator Algorithm

- **SCAN** starts at one end of the disk, moves toward the other end, reverses until reaching any end.
- Given accesses: 98, 183, 37, 122, 14, 124, 65, 67

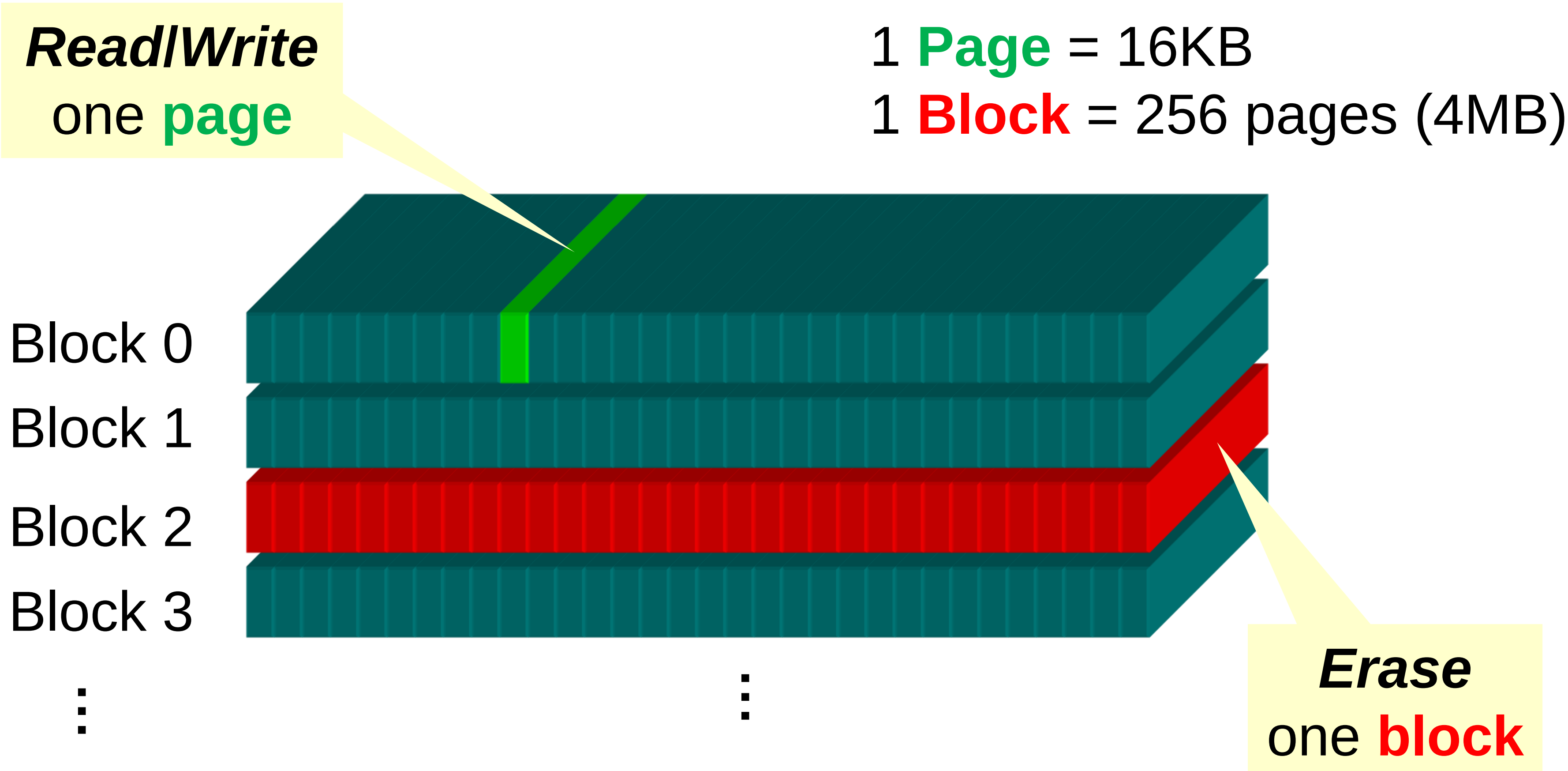


Solid State Drive (SSD)

- SSD is made by NAND flash memory.
- Digital data can be accessed randomly on memory cells of SSD (without introducing seek time as HDD).



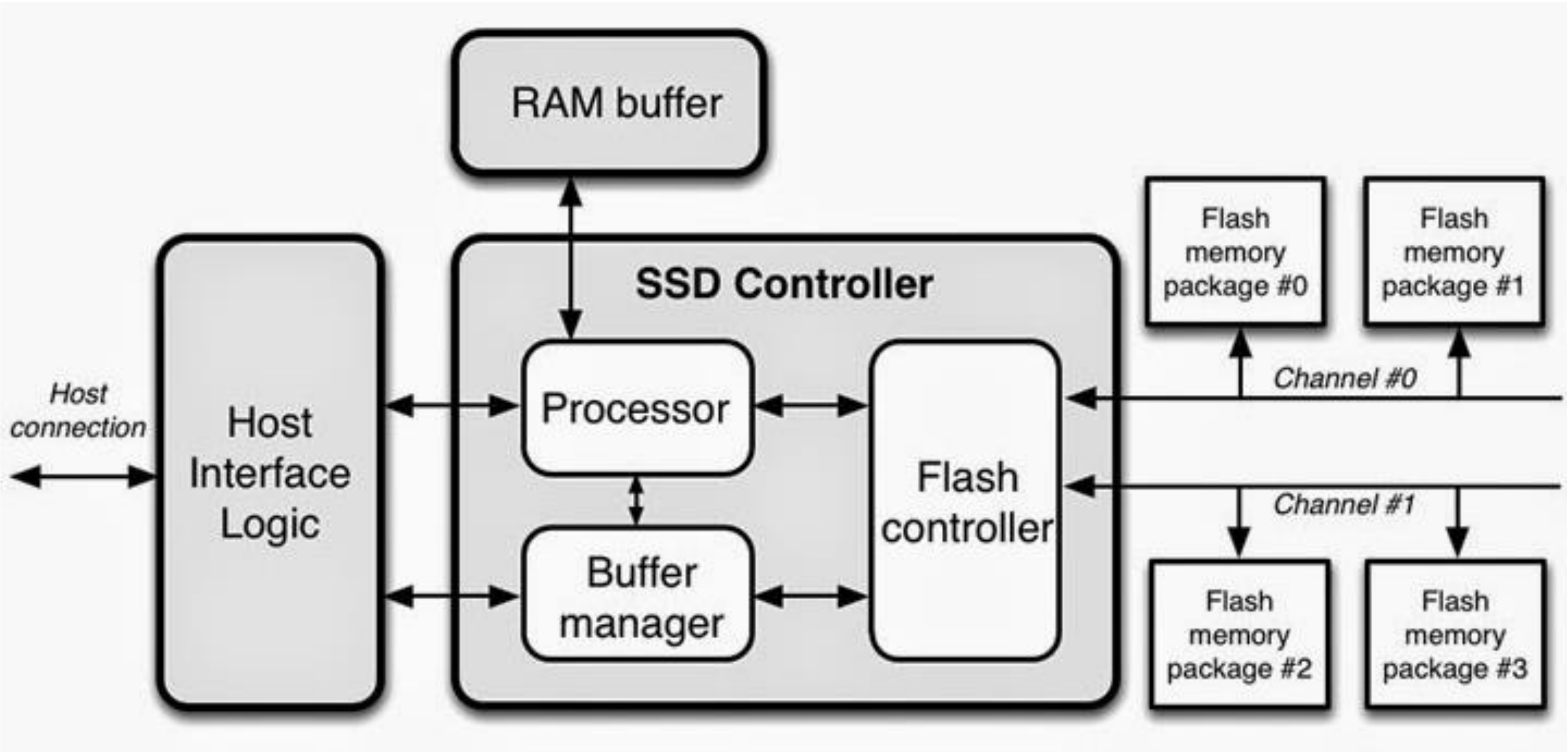
Flash Memory Characteristics



- **Write-Once Property**: Overwriting any page is not allowed unless its residing block is erased.
- **Endurance**: A block can be erased for a certain time.

SSD Management

- SSD internals require sophisticated management to deal with the flash memory characteristics.
  - There is a processor inside the SSD.



Summary

- Accessing I/O Devices
  - Memory-Mapped I/O
  - I/O Device Interface
  - Program-Controlled I/O
  - Interrupts

- Storage I/O
  - Hard Disk Drive (HDD)
  - Solid State Drive (SSD)