

**CPS104**  
**Computer Organization and Programming**  
**Lecture 18: Input-Output**

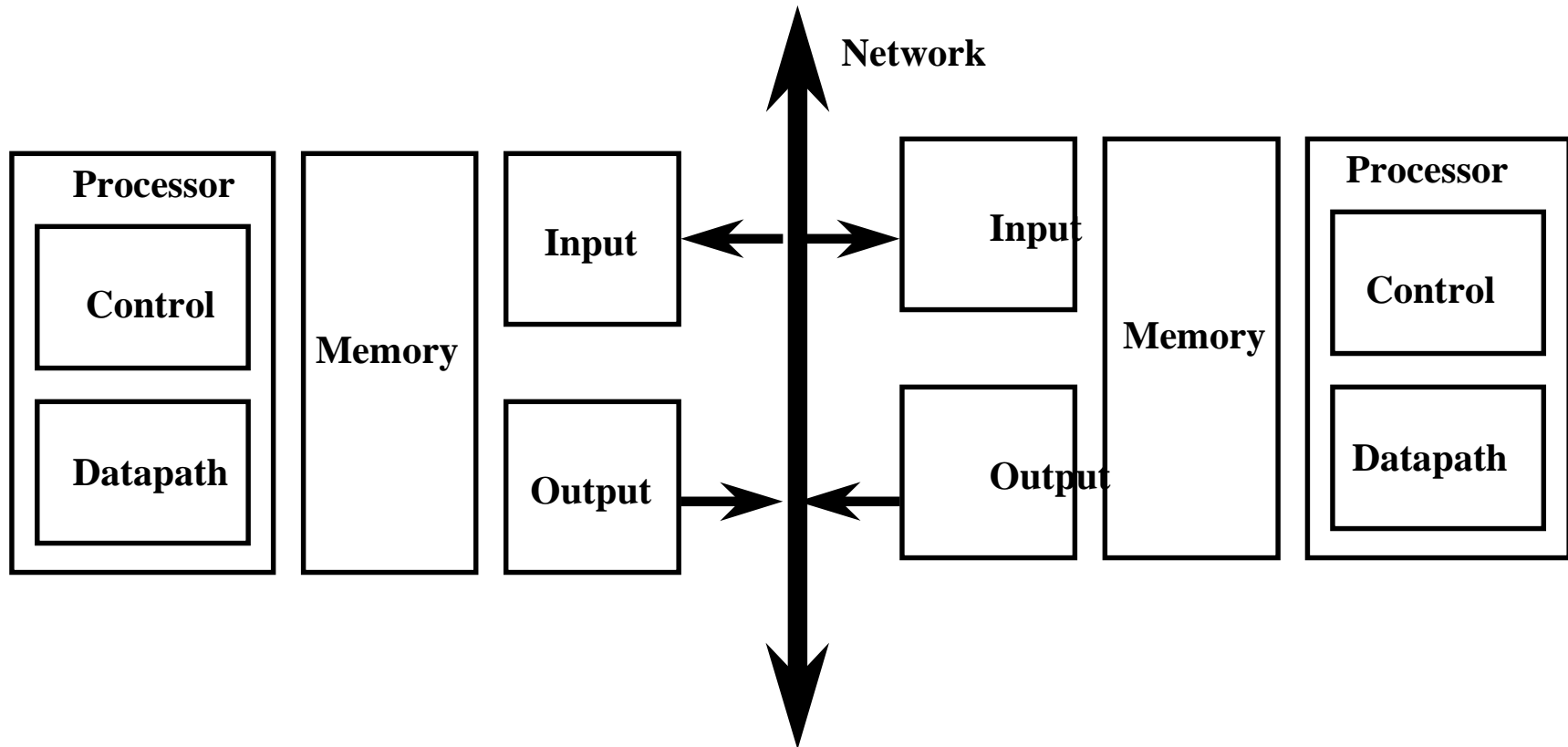
**Robert Wagner**

# Outline of Today's Lecture

- The I/O system
- Magnetic Disk
- Tape
- Buses
- DMA

# The Big Picture: Where are We Now?

- Today's Topic: I/O Systems

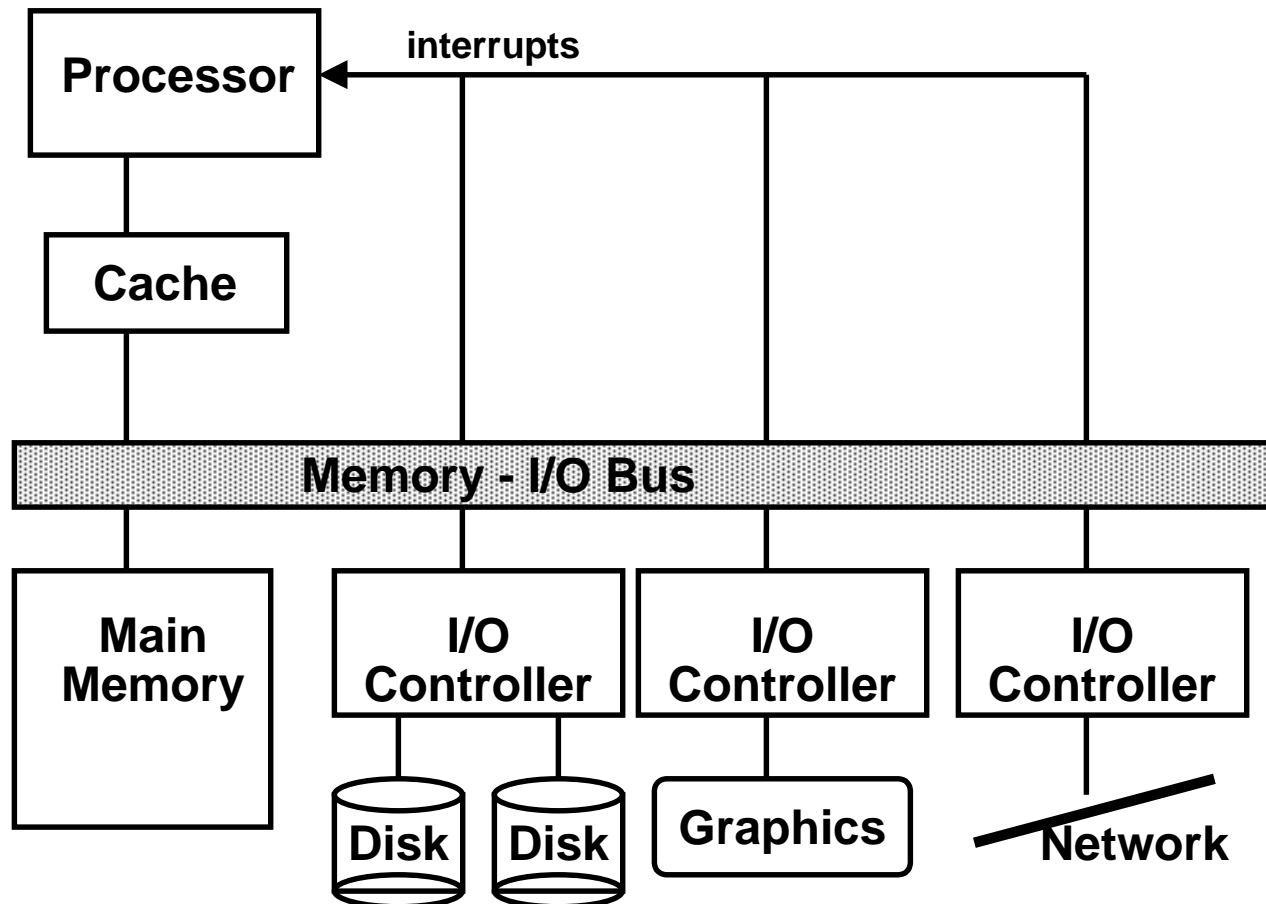


# Motivation: Who Cares About I/O?

- CPU Performance goes up : 50% - 80% per year
- I/O system performance limited by *mechanical* delays  
< 5% per year (IO per sec or MB per sec)
- Amdahl's Law: system speed-up limited by the slowest part!  
10% IO & 10x CPU => 5x Performance (lose 50%)  
10% IO & 100x CPU => 10x Performance (lose 90%)
- I/O bottleneck:  
Diminishing fraction of time in CPU  
Diminishing value of faster CPUs

# I/O System Design Issues

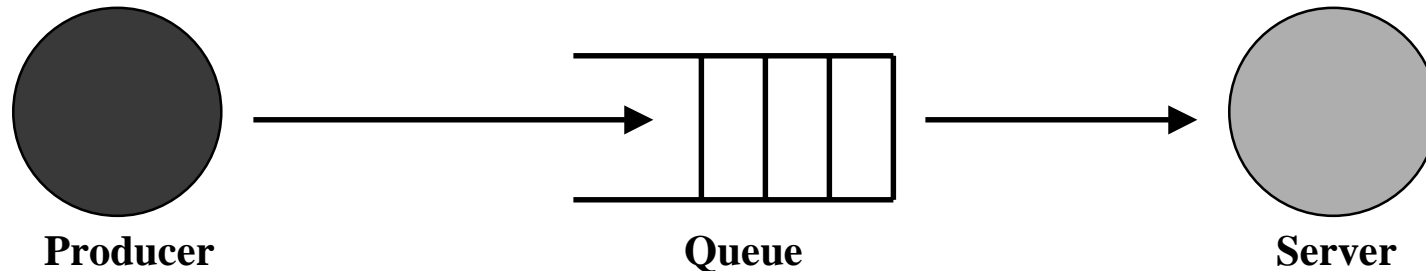
- \* Performance
- \* Expandability
- \* Resilience in the face of failure



# I/O System Performance

- **I/O System performance depends on many aspects of the system (“weakest link in the chain”):**
  - \* **The CPU**
  - \* **The memory system:**
    - **Internal and external caches**
    - **Main Memory**
  - \* **The underlying interconnection (buses)**
  - \* **The I/O controller**
  - \* **The I/O device**
  - \* **The speed of the I/O software (Operating System)**
  - \* **The efficiency of the software’s use of the I/O devices**
- **Two common performance metrics:**
  - \* **Throughput: I/O bandwidth**
  - \* **Response time: Latency**

# Producer-Server Model



- **Throughput:**

- \* The number of tasks completed by the server in unit time
- \* In order to get the highest possible throughput:
  - The server should never be idle
  - The queue should never be empty

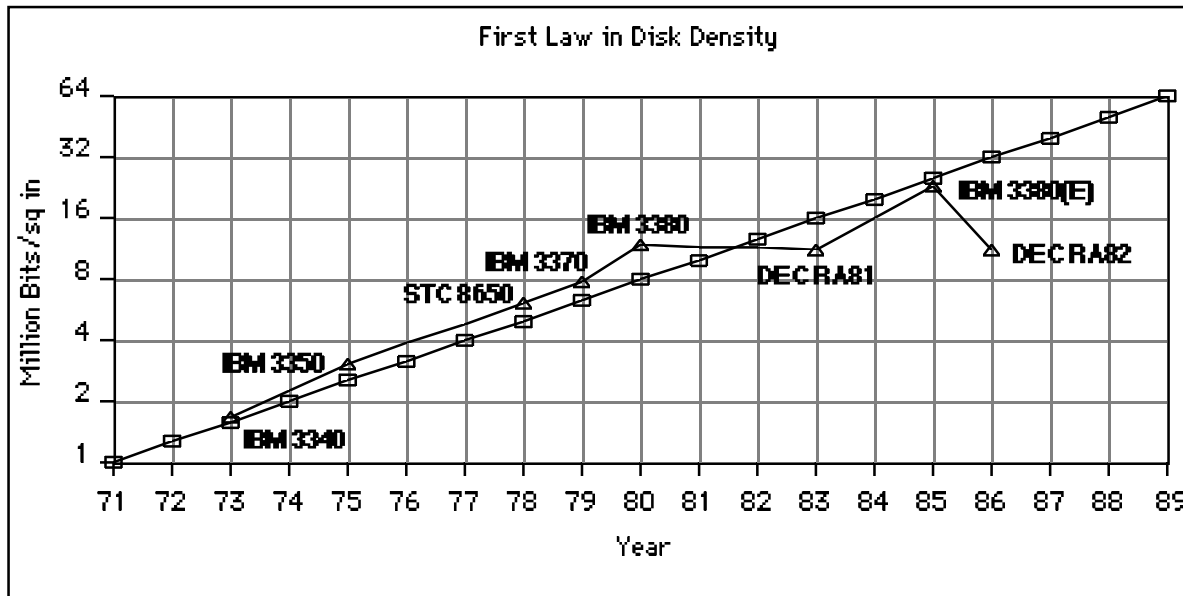
- **Response time:**

- \* Begins when a task is placed in the queue
- \* Ends when it is completed by the server
- \* In order to minimize the response time:
  - The queue should be empty
  - The server will be idle

# I/O Device Examples

<b>Device</b>	<b>Behavior</b>	<b>Partner</b>	<b>Data Rate (KB/sec)</b>
<b>Keyboard</b>	<b>Input</b>	<b>Human</b>	<b>0.01</b>
<b>Mouse</b>	<b>Input</b>	<b>Human</b>	<b>0.02</b>
<b>Line Printer</b>	<b>Output</b>	<b>Human</b>	<b>1.00</b>
<b>Laser Printer</b>	<b>Output</b>	<b>Human</b>	<b>100.00</b>
<b>Graphics Display</b>	<b>Output</b>	<b>Human</b>	<b>30,000.00</b>
<b>Network-LAN</b>	<b>Input/Output</b>	<b>Machine</b>	<b>10,000.00</b>
<b>Floppy disk</b>	<b>Storage</b>	<b>Machine</b>	<b>50.00</b>
<b>Optical Disk</b>	<b>Storage</b>	<b>Machine</b>	<b>500.00</b>
<b>Magnetic Disk</b>	<b>Storage</b>	<b>Machine</b>	<b>5,000.00</b>

# Technology Trends

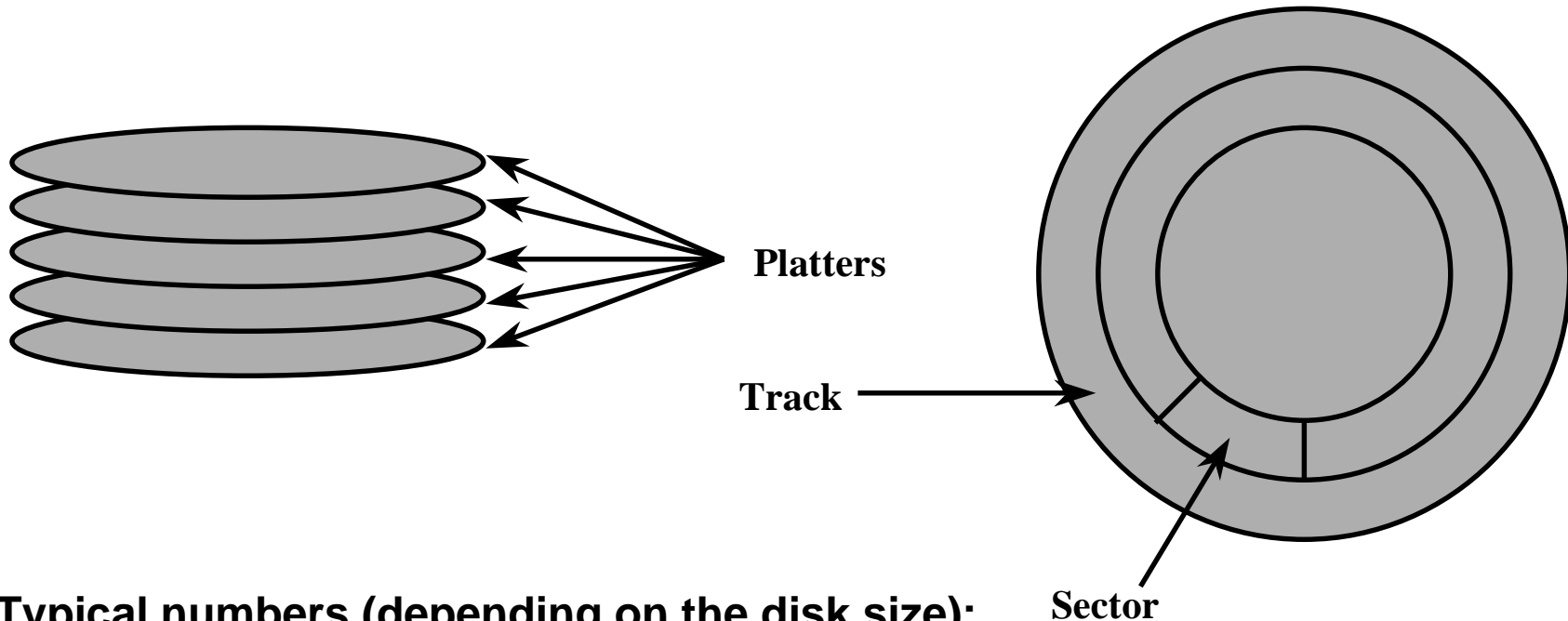


**Disk Capacity  
doubles every  
1.5 years**

- Today: Processing Power Doubles Every 18 months
- Today: Memory Size Doubles Every 18 months(?)
- Today: Disk Capacity Doubles Every 12-18 months
- *Disk Positioning Rate (Seek + Rotate) Doubles Every Ten Years!*

**The I/O  
GAP**

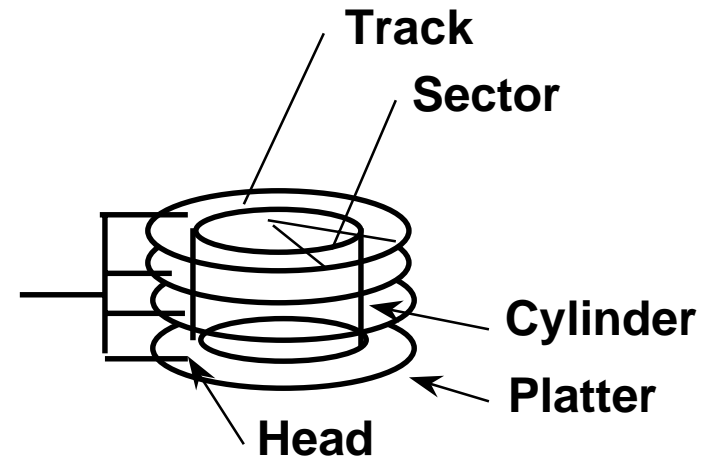
# Organization of a Hard Magnetic Disk



- **Typical numbers (depending on the disk size):**
  - \* **500 to 2,000 tracks per surface**
  - \* **32 to 128 sectors per track**
    - **A sector is the smallest unit that can be read or written**
- **Traditionally all tracks have the same number of sectors:**
  - \* **Constant bit density: record more sectors on the outer tracks**
  - \* **Recently relaxed: constant bit size, speed varies with track location**

# Magnetic Disk Characteristic

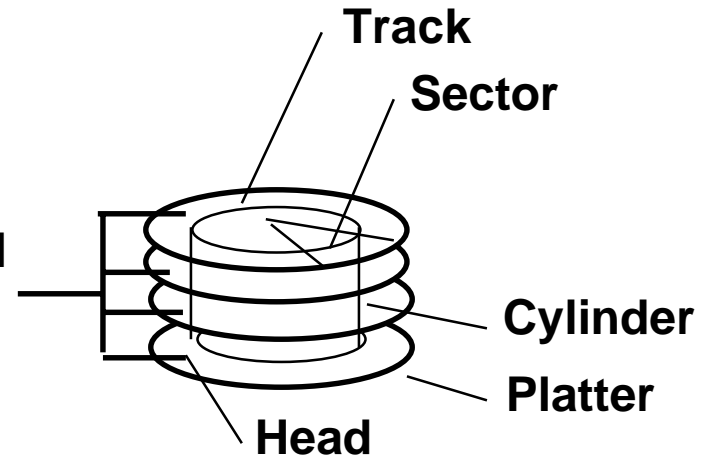
- **Cylinder:** all the tracks under the head at a given point on all surfaces
- **Read/write data is a three-stage process:**
  - \* **Seek time:** position the arm over the proper track
  - \* **Rotational latency:** wait for the desired sector to rotate under the read/write head
  - \* **Transfer time:** transfer a block of bits (sector) under the read-write head
- **Average seek time as reported by the industry:**
  - \* Typically in the range of 8 ms to 12 ms
  - \*  $(\text{Sum of the time for all possible seeks}) / (\text{total \# of possible seeks})$
- **Due to locality of disk reference, observed average seek time may:**
  - \* Only be 25% to 33% of the advertised number



# Typical Numbers of a Magnetic Disk

- **Rotational Latency:**

- \* **Most disks rotate at 3,600 or 7200 RPM**
- \* **Approximately 16 ms to 8 ms per revolution, respectively**
- \* **An average latency to the desired information is halfway around the disk: 8 ms at 3600 RPM, 4 ms at 7200 RPM**



- **Transfer Time is a function of :**

- \* **Transfer size (usually a sector): 1 KB / sector**
- \* **Rotation speed: 3600 RPM to 7200 RPM**
- \* **Recording density: bits per inch on a track**
- \* **Diameter typical diameter ranges from 2.5 to 5.25 in**
- \* **Typical values: 2 to 12 MB per second**

# Tape vs. Disk

- **Longitudinal tape uses same technology as hard disk; tracks its density improvements**
- **Inherent cost/performance based on geometry:  
fixed rotating platters with gaps  
(random access, limited area, fixed media, re-write any block)**

**vs.**

- removable long strips wound on spool  
(sequential access, "unlimited" length, write only at end )**
- **New technology trend:  
Helical Scan (VCR, Camcorder, DAT)  
Spins head at angle to tape to improve density**

# Current Drawbacks to Tape

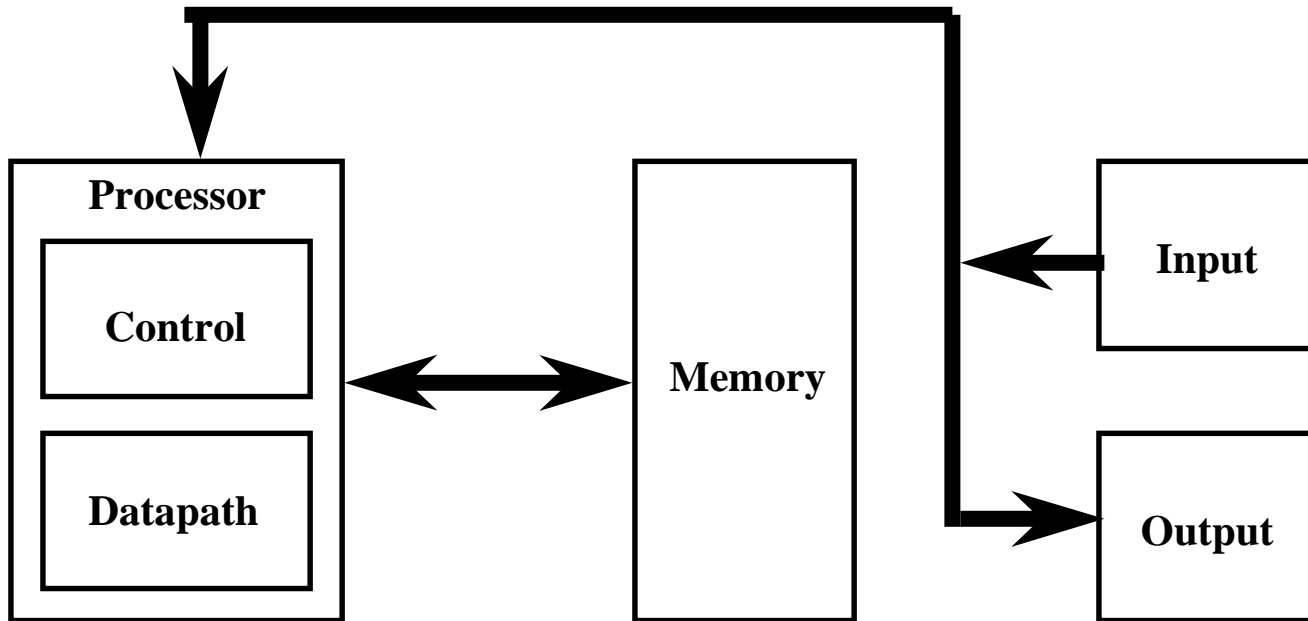
- **Tape wear out:**
  - **Helical 100s of passes to 1000s for longitudinal**
- **Head wear out:**
  - **2000 hours for helical**
- **Both must be accounted for in economic / reliability model**
- **Long rewind, eject, load, spin-up times;**

## CDR vs. Tape

	CDR	Helical Scan Tape
Type	5.25"	8mm
Capacity	0.75 GB	5-12 GB
Media Cost	\$3	\$8
Drive Cost	\$200	\$600
Access	Write Once	Read/Write
Robot Time	10 - 20 s	10 - 20 s

**Media cost ratio CDR vs. helical tape  
= 2.7 : 1**

# Buses: Connecting I/O to Processor and Memory

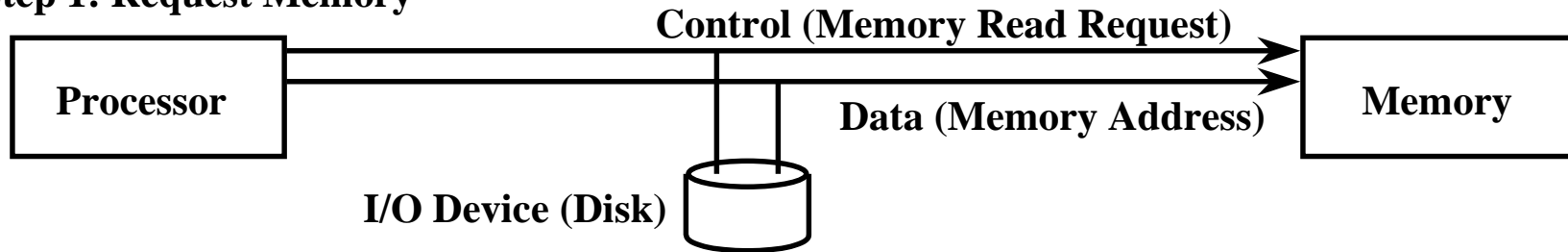


- A bus is a shared communication link
- It uses one set of wires to connect multiple subsystems

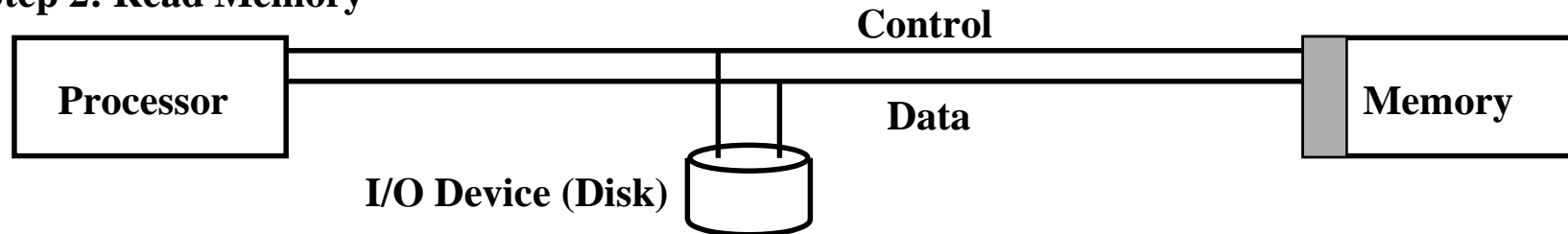
# Output Operation

- Output is defined as the Processor sending data to the I/O device:

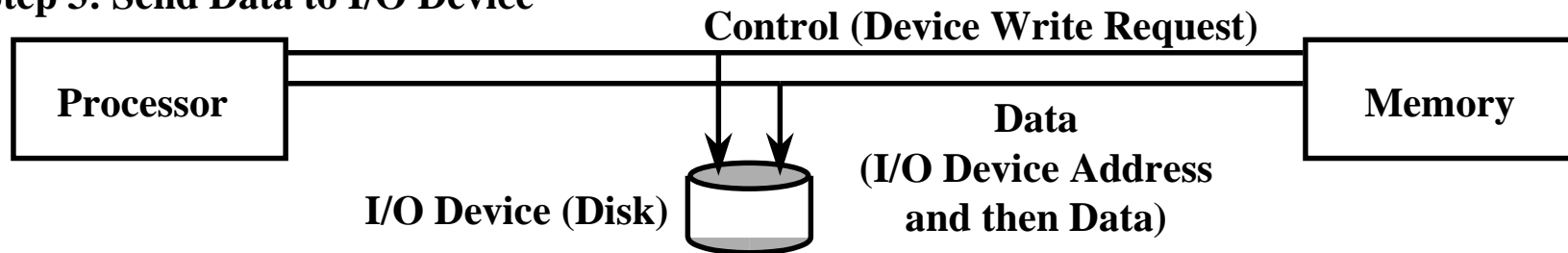
## Step 1: Request Memory



## Step 2: Read Memory



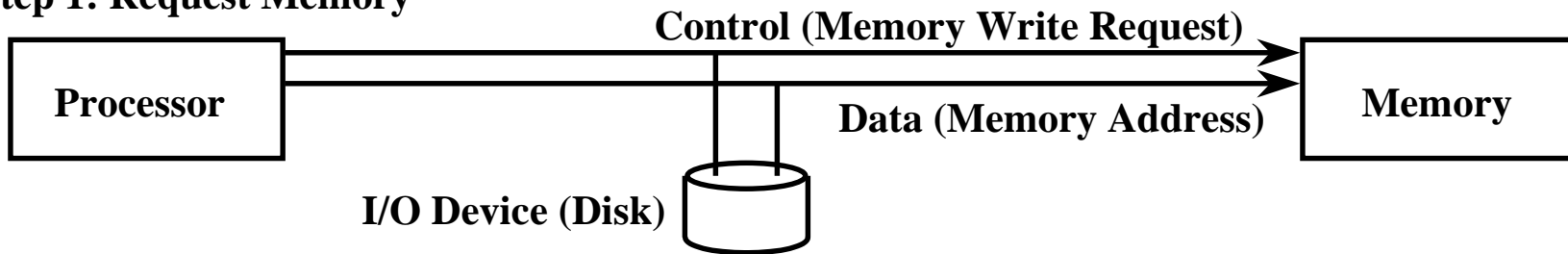
## Step 3: Send Data to I/O Device



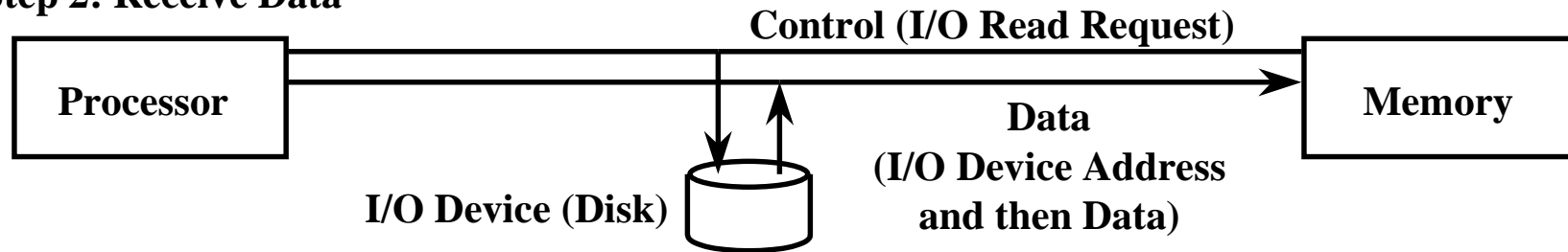
# Input Operation

- Input is defined as the Processor receiving data from the I/O device:

## Step 1: Request Memory



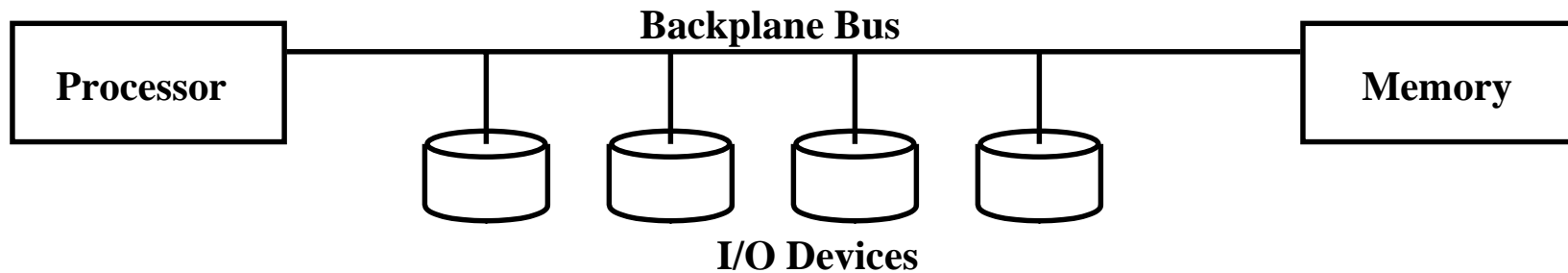
## Step 2: Receive Data



# Types of Buses

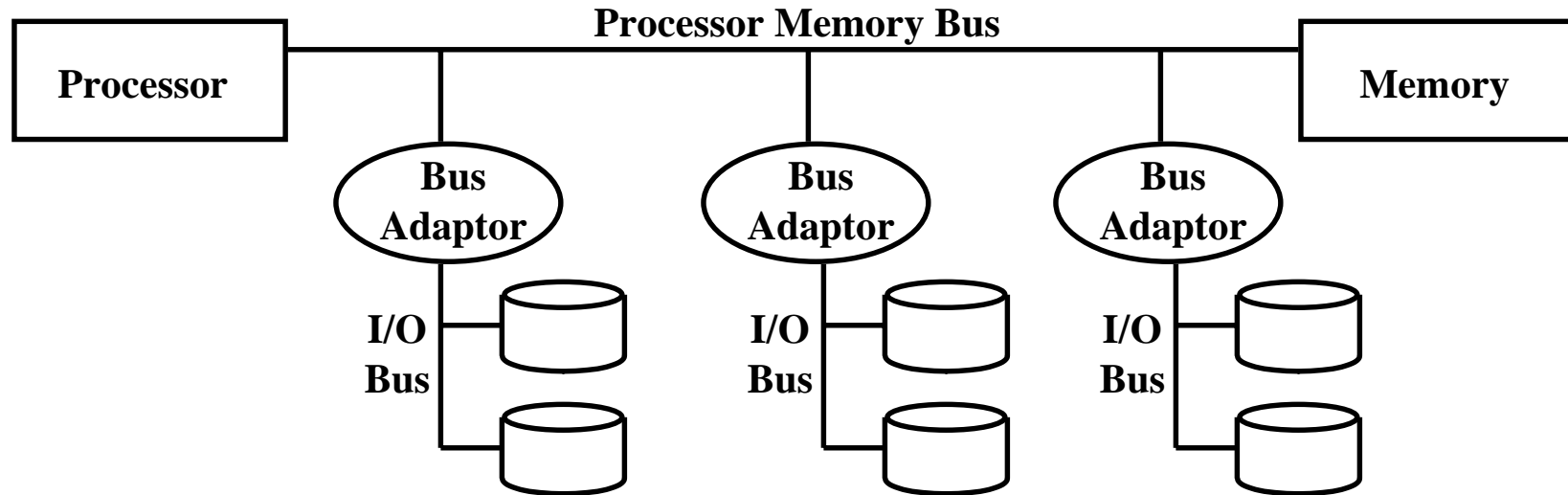
- **Processor-Memory Bus (design specific)**
  - \* **Short and high speed**
  - \* **Only need to match the memory system**
    - **Maximize memory-to-processor bandwidth**
  - \* **Connects directly to the processor**
- **I/O Bus (industry standard)**
  - \* **Usually is lengthy and slower**
  - \* **Need to match a wide range of I/O devices**
  - \* **Connects to the processor-memory bus or backplane bus**
- **Backplane Bus (industry standard)**
  - \* **Backplane: an interconnection structure within the chassis**
  - \* **Allow processors, memory, and I/O devices to coexist**
  - \* **Cost advantage: one single bus for all components**

# A Computer System with One Bus: Backplane Bus



- A single bus (the backplane bus) is used for:
  - \* Processor to memory communication
  - \* Communication between I/O devices and memory
- Advantages: Simple and low cost
- Disadvantages: slow and the bus can become a major bottleneck
- Example: IBM PC

# A Two-Bus System



- **I/O buses tap into the processor-memory bus via bus adaptors:**
  - \* **Processor-memory bus:** mainly for processor-memory traffic
  - \* **I/O buses:** provide expansion slots for I/O devices
- **Example: Apple Macintosh-II**
  - \* **NuBus:** Processor, memory, and a few selected I/O devices
  - \* **SCCI Bus:** the rest of the I/O devices

# OS and I/O Systems Communication Requirements

- **The Operating System must be able to prevent:**
  - \* **The user program from communicating with the I/O device directly**
- **If user programs could perform I/O directly:**
  - \* **Protection to the shared I/O resources could not be provided**
    - **User could change the OS copy on disk**
    - **Data privacy could not be maintained**
- **Three types of communication are required:**
  - \* **The OS must be able to give commands to the I/O devices**
  - \* **The I/O device must be able to notify the OS when the I/O device has completed an operation or has encountered an error**
  - \* **Data must be transferred between memory and an I/O device**

# OS to I/O Interface

- **I/O device control registers are addressable as memory locations**
- **OS initiates operation of I/O device X by storing into X's control register**
  - \* **X's control register might be location 0x100 in memory**
  - \* **Each device uses different memory location(s)**
- **Device indicates "operation complete" (and reports errors) by setting code in another control register, also addressed as memory location**
  - \* **OS can "poll" devices -- examine control registers of each, in turn**
  - \* **Devices can interrupt processor to signal need for attention**
    - **Each device, or possibly "type" of device, can use a different interrupt location, or cause code**
- **This approach avoids special I/O op-codes**
- **Protection of I/O devices handled by memory protection system**
  - \* **OS prevents user program from read/write access to I/O locations**

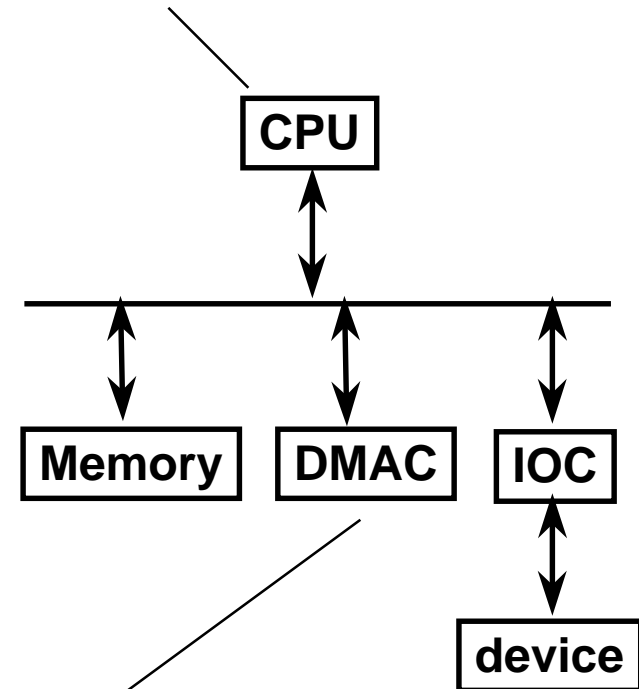
# I/O Interrupt

- **An I/O interrupt is just like any exception except:**
  - \* **An I/O interrupt is asynchronous**
  - \* **Information about the interrupt needs to be conveyed to OS**
- **An I/O interrupt is asynchronous with respect to instruction execution:**
  - \* **I/O interrupt is not associated with any instruction**
  - \* **I/O interrupt does not prevent any instruction from completion**
    - **Hardware designer can pick any convenient point to take an interrupt**
- **I/O interrupt is more complicated than exception:**
  - \* **Needs to convey the identity of the device generating the interrupt**
  - \* **Interrupt requests can have different urgencies:**
    - **Interrupt request needs to be prioritized**

# Delegating I/O Responsibility from the CPU: DMA

- **Direct Memory Access (DMA):**
  - \* **External to the CPU**
  - \* **Act as a “master” on the bus**
  - \* **Transfers blocks of data to or from memory without CPU intervention**

CPU sends a starting address, direction, and length count to DMAC. Then issues "start".



DMAC provides handshake signals for Peripheral Controller, and Memory Addresses and handshake signals for Memory.

# Summary:

- I/O performance is weakest link in chain between OS and device
- Disk I/O Benchmarks: I/O rate vs. Data rate vs. latency
- Three Components of Disk Access Time:
  - \* Seek Time: advertised to be 8 to 12 ms. May be lower in real life.
  - \* Rotational Latency: 4.1 ms at 7200 RPM and 8.3 ms at 3600 RPM
  - \* Transfer Time: 2 to 12 MB per second
- Three types of buses: Processor-memory, I/O, Backplane
  - \* performance vs. cost
- I/O device notifying the operating system:
  - \* Polling: it can waste a lot of processor time
  - \* I/O interrupt: similar to exception except it is asynchronous
- Delegating I/O responsibility from the CPU: DMA, or even IOP