ĐẠI HỌC TÔN ĐỨC THẮNG
TON DUC THANG UNIVERSITY

COMPUTER ORGANISATION
(TỔ CHỨC MÁY TÍNH)

IO
(Own reading only)

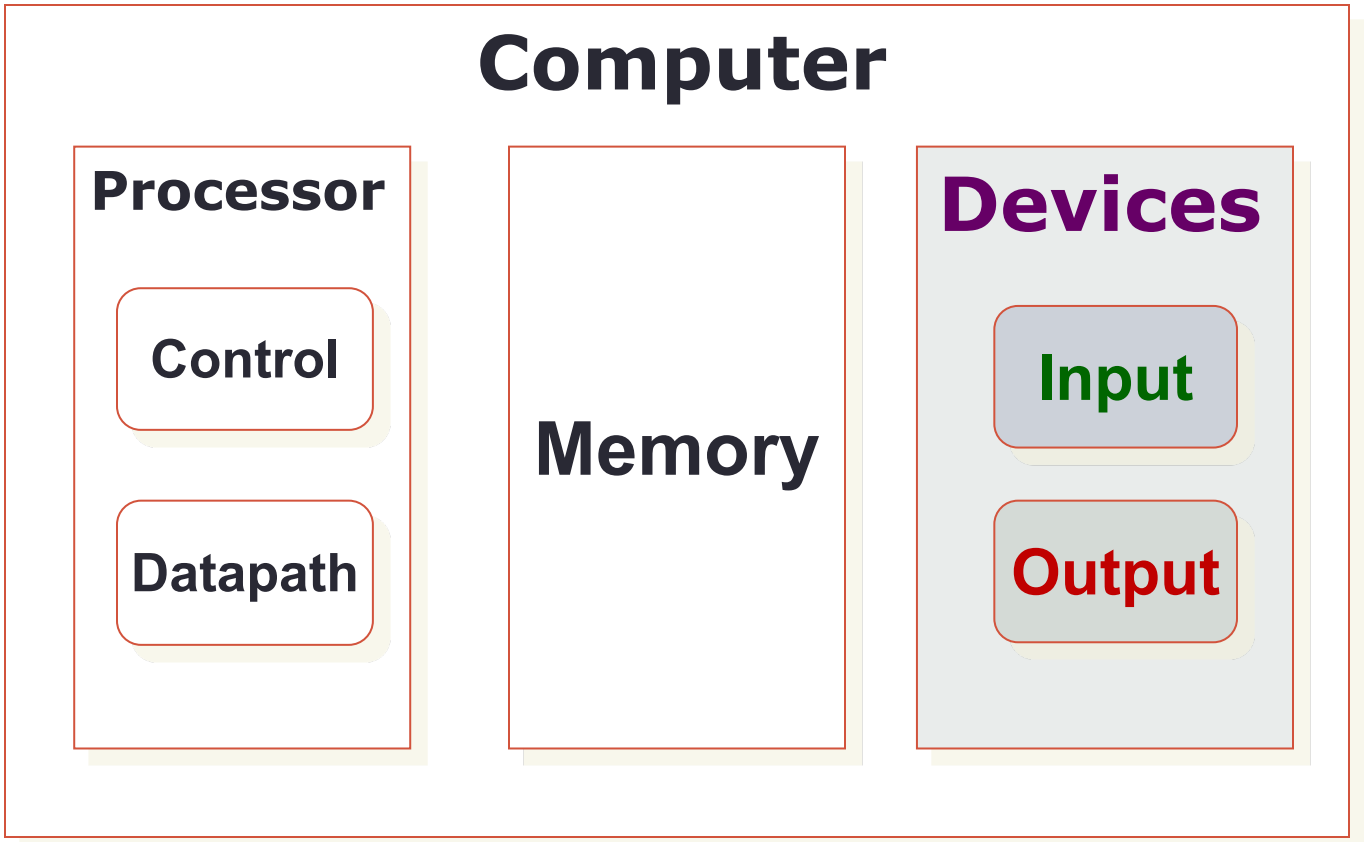
Acknowledgement

- The contents of these slides have origin from School of Computing, National University of Singapore.
- We greatly appreciate support from Mr. Aaron Tan Tuck Choy for kindly sharing these materials.

Policies for students

- These contents are only used for students PERSONALLY.
- Students are NOT allowed to modify or deliver these contents to anywhere or anyone for any purpose.

Input/Output Devices



Input/Output Devices

Why I/O Matters?

- CPU performance increase ~ 60% per year
- I/O performance increase < 10% per year
  - Limited by mechanical delays
- **Amdahl's Law:**
  - System speedup is limited by the slowest part
- Example:
  - Suppose 1 sec I/O + 4 sec CPU ➔ 5 seconds
  - Increase CPU performance by 100% ➔ 3 seconds

  = Only get a 66% speedup ➔ I/O bottleneck

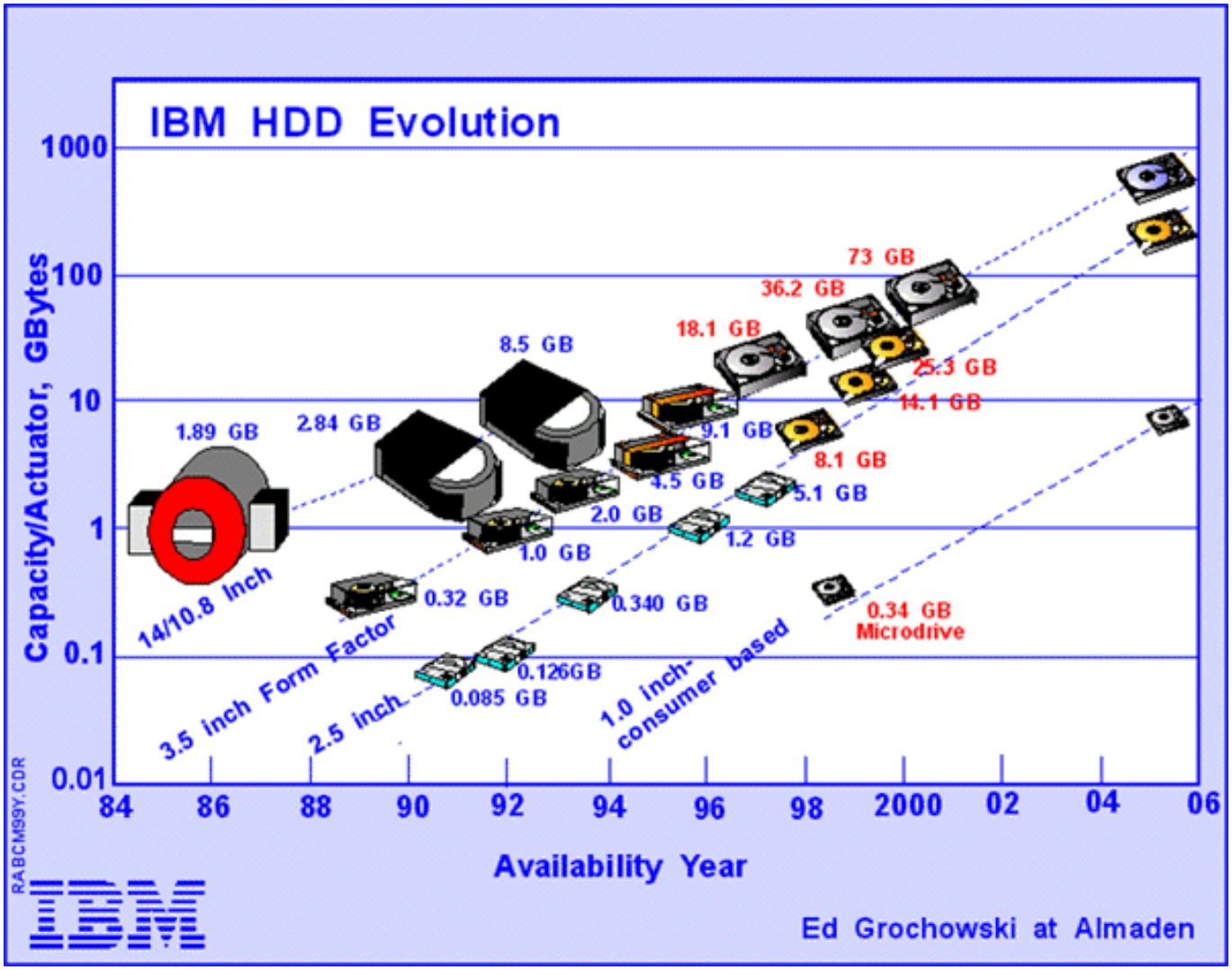
I/O Devices: Types & Characteristics

- **Behavior**
  - **Input:** read once
  - **Output:** write only, cannot be read
  - **Storage:** can be reread and usually rewritten
- **Partner**
  - What's on the other end? Human or machine
- **Data Rate**
  - Peak rate of transfer between I/O and memory/CPU

Some I/O Devices

Device	Behavior	Partner	Data Rate (MBit/sec)
Keyboard	Input	Human	0.0001
Mouse	Input	Human	0.0038
Laser Printer	Output	Human	3.2000
Optical Disk	Storage	Machine	80.0000 – 220.0000
Flash Memory	Storage	Machine	32.0000 – 200.0000
Network-LAN	Input or output	Machine	11.0000 – 54.00000
Graphics Display	Output	Human	800.0000 – 8000.00000

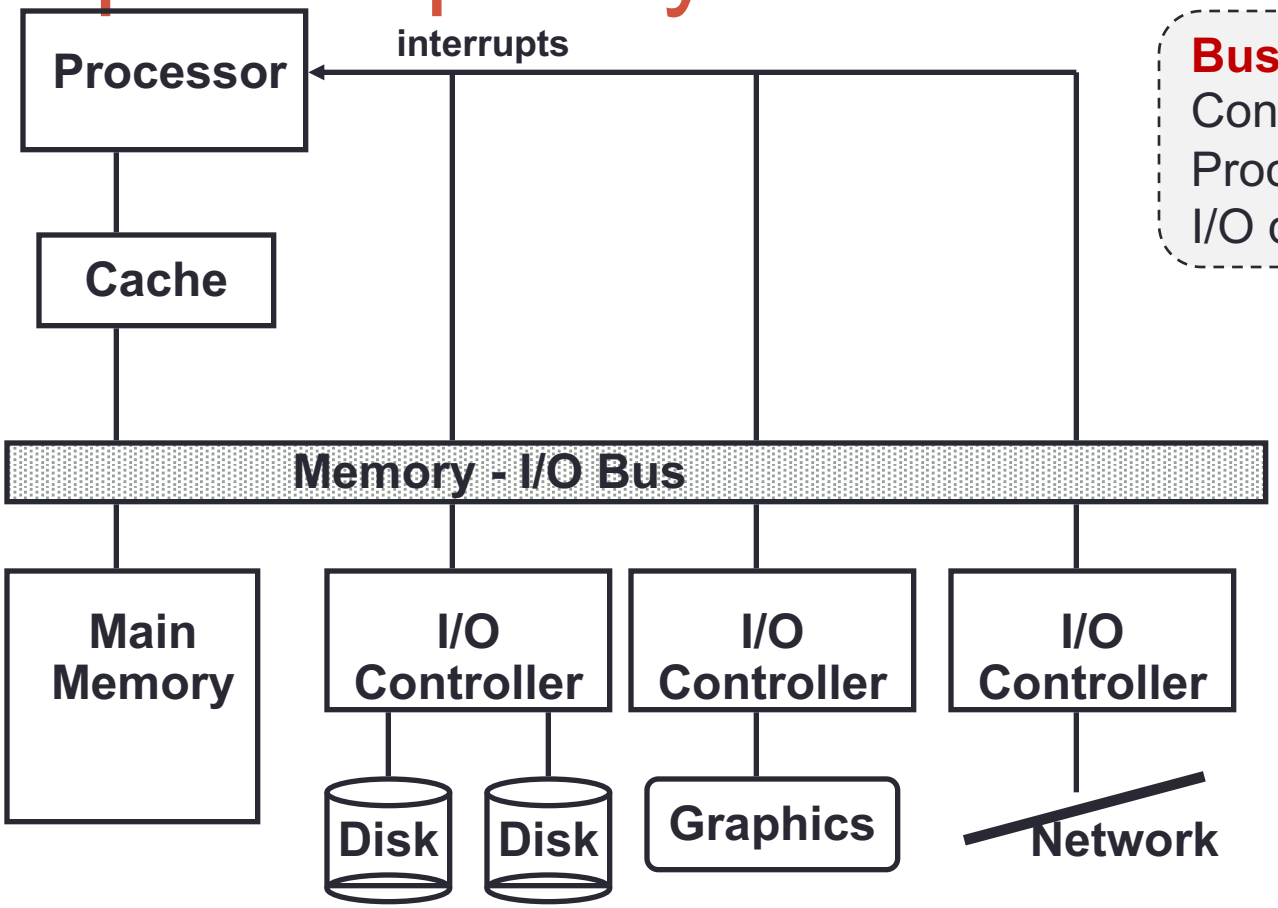
Hard Disk Drive Evolution



Floppy Disks

8-inch	5¼-inch	3½-inch
1971	1976	Mid-1980s
80KB	360KB	1.44MB

Input/Output System



Bus:
Connection between Processor, Memory and I/O devices

Communication between Processor and devices is via *bus protocols* and *interrupts*

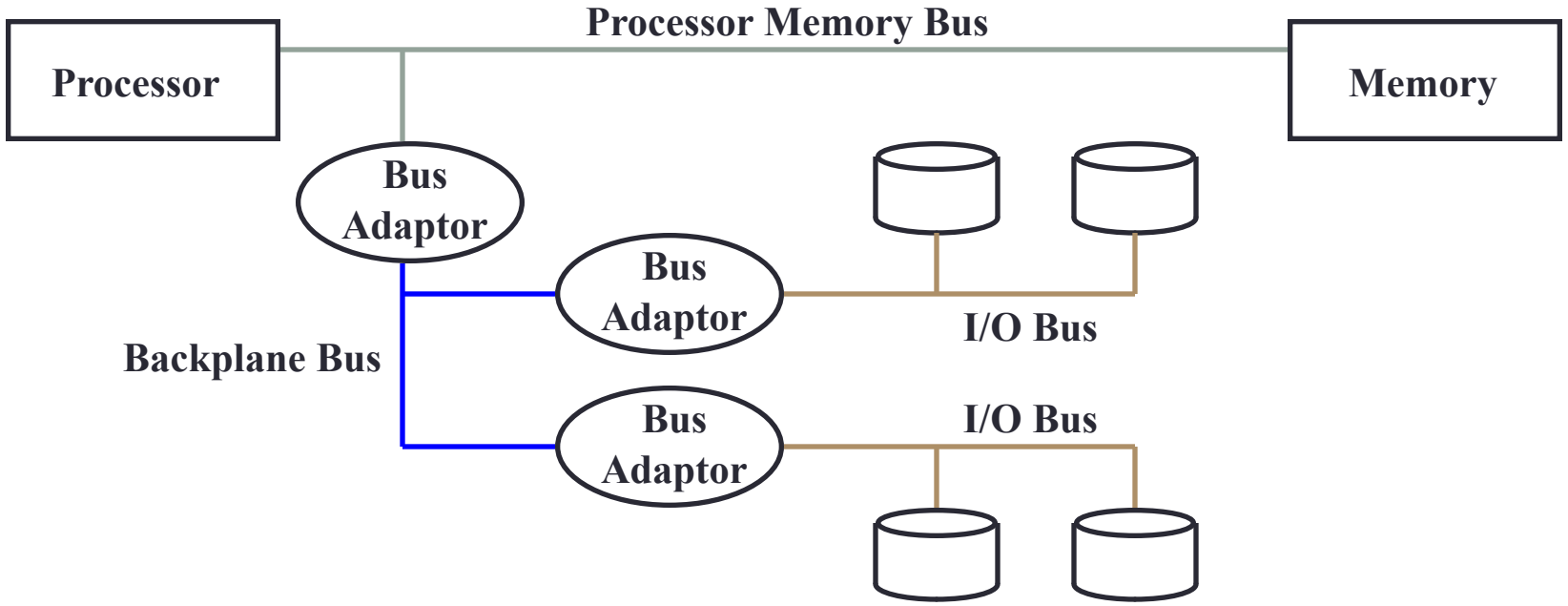
Bus Basics

- Bus consists of:
  - **Control lines:** Signal requests and acknowledgments
  - **Data lines:** Carry information between the source and the destination
- **Bus Transactions**
  - Sending the **address**
  - Receiving or sending the **data**
- **Advantages**
  - Versatility: single connection scheme for easy add-ons
  - Low cost: single set of wires shared in multiple ways
- **Disadvantages**
  - Communication bottleneck: bandwidth limits the maximum I/O throughput
  - Devices will not be able to use the bus when they need to

Types of Buses

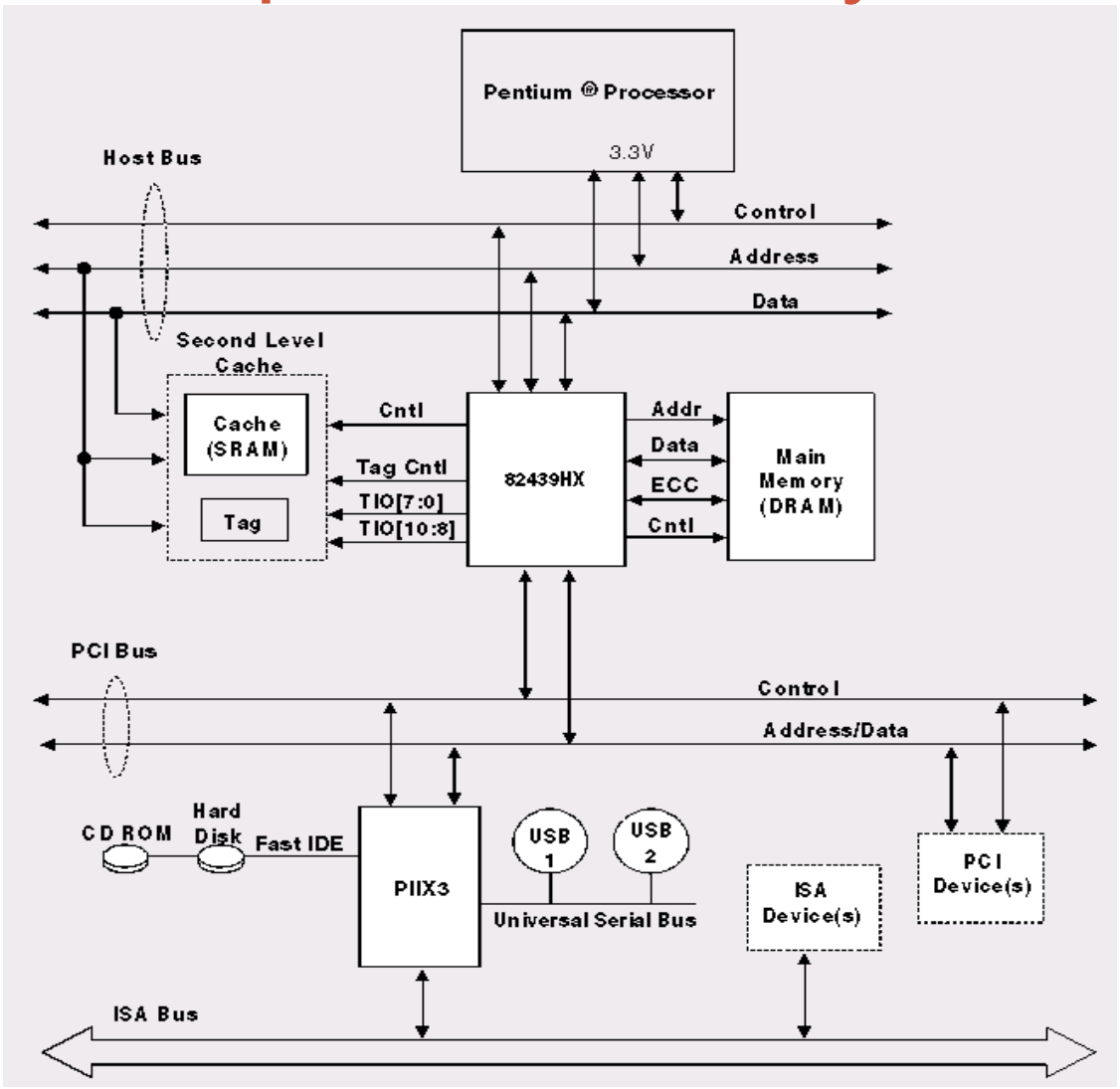
- **Processor-Memory Bus** (design specific)
  - Short and high speed
  - Only need to match the memory system
    - Maximize memory-to-processor bandwidth
  - Connects directly to the processor
  - Optimized for cache block transfers
- **I/O Bus** (industry standard)
  - Usually is lengthy and slower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

A Three-Bus System



- A small number of backplane buses tap into the processor-memory bus
  - Processor-memory bus is used for processor memory traffic
  - I/O buses are connected to the backplane bus
- **Advantage**: workload on the processor bus is greatly reduced

Example: Pentium System Organisation



Processor/Memory Bus

PCI Bus [Backplane]

I/O Busses [IDE, SCSI]

Processor to I/O Devices

- Two methods for communicating with the device

1. Special I/O instructions:

- Each instruction specifies:
  - **Device number**: Uniquely identifies the device
  - **Command word**: Operation to be performed
- Information is sent using I/O bus

2. Memory-mapped I/O:

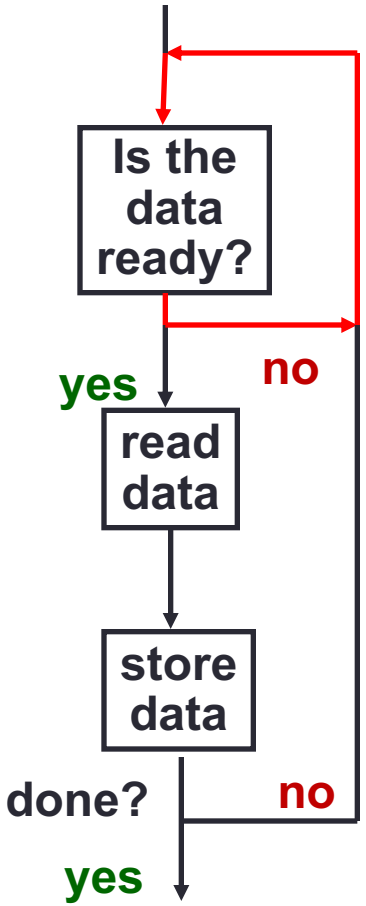
- Designates a portion of the memory address space to I/O device communication
- Read and writes to those addresses are interpreted as commands to the I/O devices

I/O Device to Processor

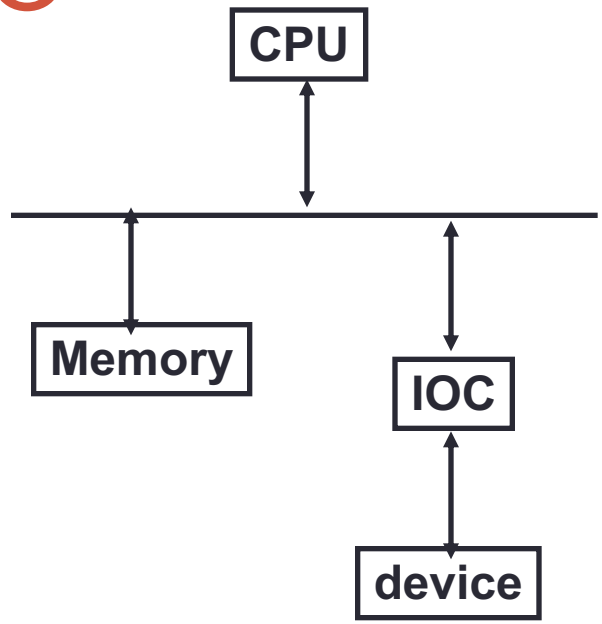
- Two methods to communicate with the processor
- **Polling:**
  - The I/O device put information in a **status register**
  - The processor periodically check the **status register**
- **Interrupt driven I/O:**
  - The I/O device causes the processor to be interrupted

  ➔ Processor will cease doing it's original task and handle the interrupt accordingly
- The process above is commonly handled by part of the operating system (OS)

Polling: Programmed I/O

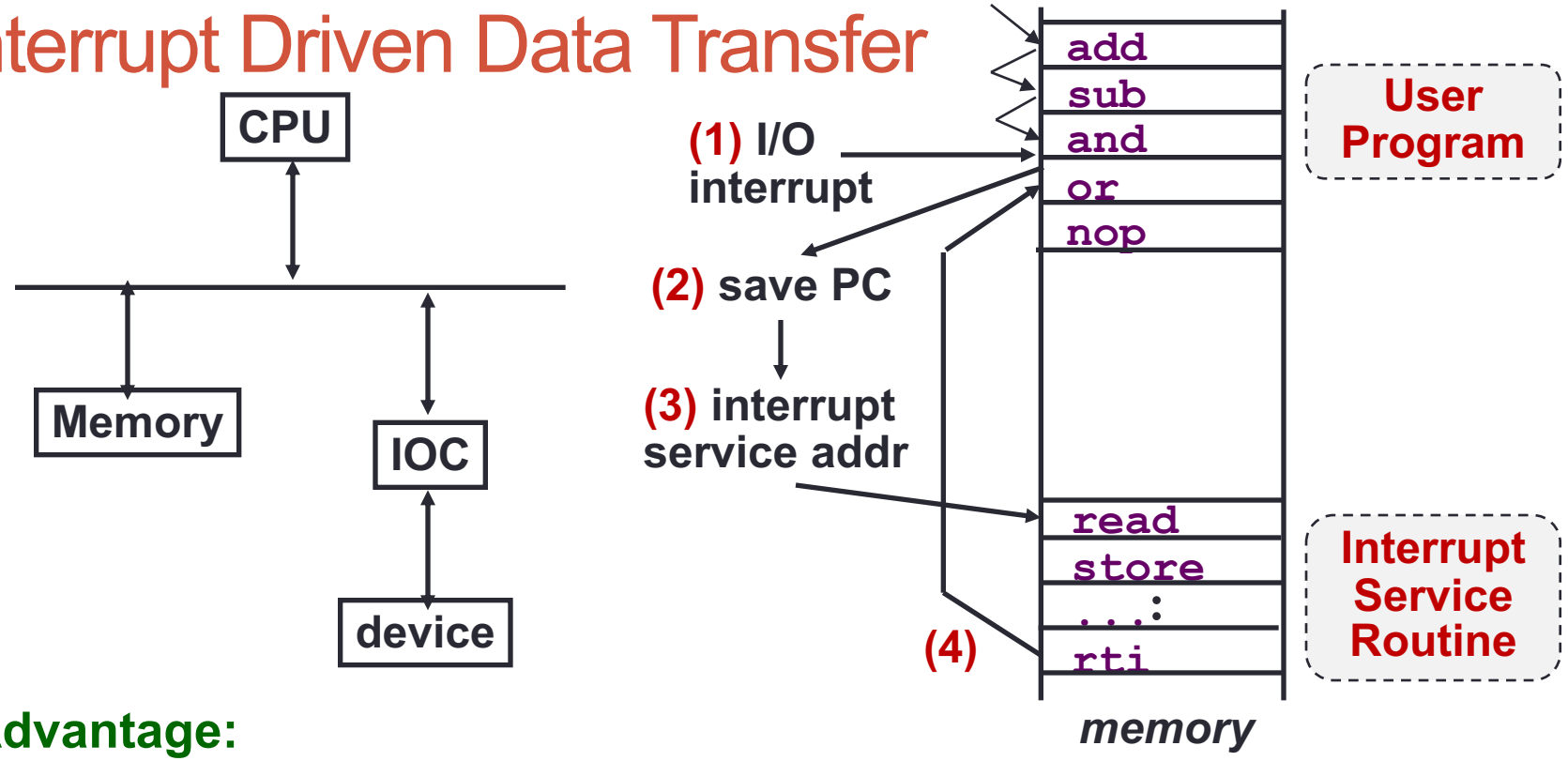


Busy wait loop
Inefficient use of the CPU time unless the device is very fast!



- **Advantage:**
  - Simple: the processor is totally in control and does all the work
- **Disadvantage:**
  - Polling overhead can consume a lot of CPU time

Interrupt Driven Data Transfer



- **Advantage:**
  - User program progress is only halted during actual transfer
- **Disadvantage**, special hardware is needed to:
  - Cause an interrupt (I/O device)
  - Detect an interrupt (processor)
  - Save the proper states to resume after the interrupt (processor)

I/O Interrupt

- 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
    - You can pick your own convenient point to take an interrupt
- I/O interrupt is complicated:
  - Needs to convey the identity of the device generating the interrupt
  - Interrupt requests can have different urgencies:
    - Interrupt request needs to be prioritized

More I/O Related Topics

- Bus access and Bus Arbitration Mechanism
- Networking
- Direct memory access

SUMMARY

- I/O performance is limited by weakest link in chain between OS and device
- Wide range of devices
- Communication between I/O device and Processor:
  - Polling: it can waste a lot of processor time
  - I/O interrupt: similar to exception except it is asynchronous

END