BACS1024 INTRODUCTION TO COMPUTER SYSTEMS

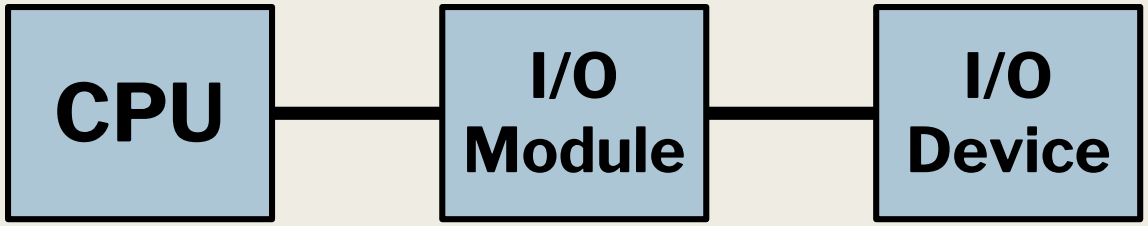
Chapter 8: I/O Facilities

0. Overview

1. I/O Modules
2. I/O Handling Techniques
3. I/O Architecture

1. I/O Modules

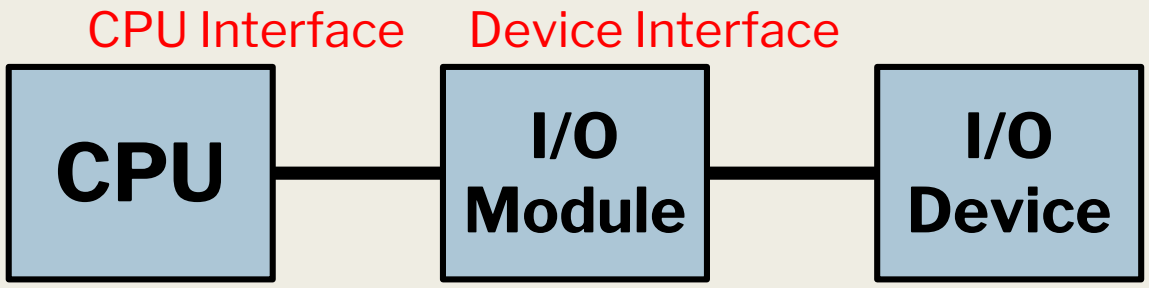
1. I/O Modules



- **I/O Modules**
  - Serves as an interface between the CPU and specific device, accepting commands from CPU and controlling device
  - Functions
    - Recognizes message addressed to it and accepts commands from the CPU
    - Provides a buffer where the data from memory can be held until it can be transferred to the disk
    - Provides the necessary registers and controls to perform a direct memory transfer
    - Interrupt capability, to notify CPU when the operation is completed.

1. I/O Modules

- **I/O Modules**



- CPU interface: Performs CPU interfacing tasks
  - Accept I/O commands from the CPU
  - Sending interrupts and status information to CPU
- Device interface: supplier control of the device
  - Moving the head to the correct track in a disk drive & rewinding tape

1. I/O Modules

- **I/O Modules**
  - Simple CPU I/O instruction can be used to control complex operations
  - Simplify the task f interfacing peripheral devices to a CPU. This off load a considerable amount of work from the CPU
  - Make it possible to control I/O to a peripheral with a few simple I/O commands
  - Support Direct Memory Access (DMA) ➔ CPU is free to perform other tasks
  - Provides the specialized circuitry to interface different types of peripherals.

2. I/O Handling Techniques

2. I/O Techniques

- **I/O techniques**
  - Data transfer between the CPU and I/O devices can be handled in three types of models

  (1) Programmed I/O : CPU controlled I/O

  (2) Interrupt-driven I/O : External input control

  (3) Direct Memory Access (DMA): Data transfer by bypassing CPU

2. I/O Techniques

- **I/O techniques**

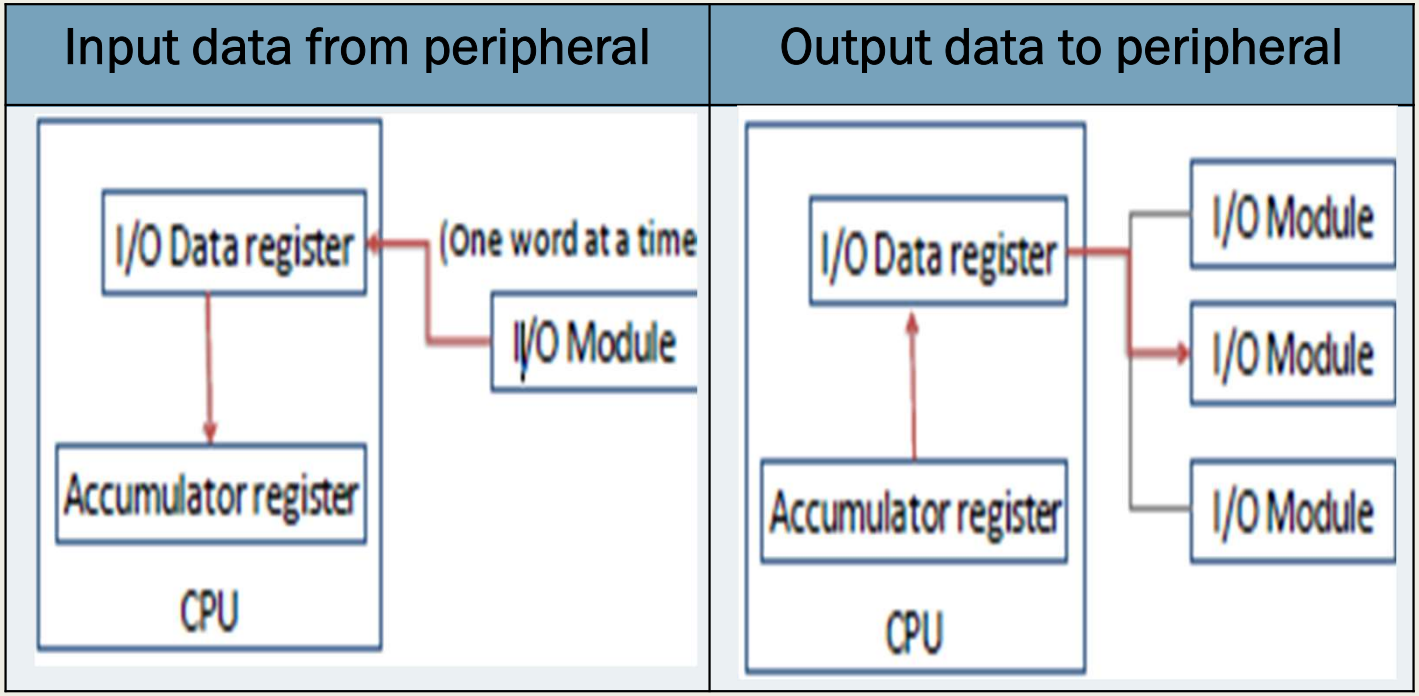
  (1) Programmed I/O

  - ❑ The simplest method performing I/O
  - ❑ An I/O module is connected to a pair of I/O registers in the CPU via a bus
  - ❑ Each instruction produces a single input / output
  - ❑ Address information must be sent with the I/O devices individually
  - ❑ Full instruction cycle for each I/O data word (very slow)
  - ❑ Primary use: Keyboard

2. I/O Techniques

- **I/O techniques**

  (1) Programmed I/O



2. I/O Techniques

- **I/O techniques**

  (2) Interrupt-driven I/O

  - Signal that causes the CPU to alter its normal flow on instruction execution
    - Free CPU from waiting for events
    - Provides control for external input
  - E.g.:
    - An unexpected user input
    - Illegal instructions
    - Multitasking, multiprocessing
    - An abnormal situation

2. I/O Techniques

- **I/O techniques**

(2) Interrupt-driven I/O

❑ Uses of interrupt

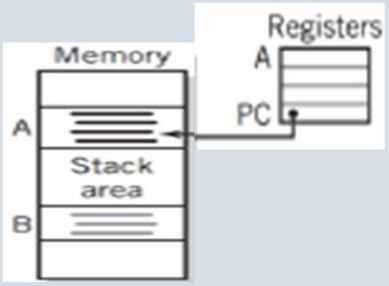
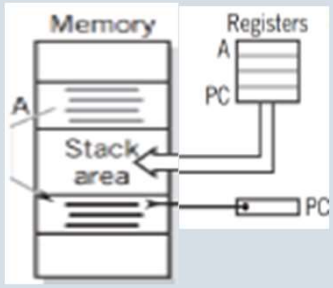
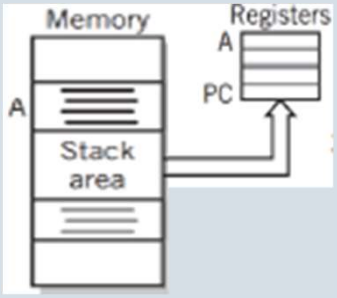
1. An unexpected / external user input	2. An a completion signal	3. Multitasking, multiprocessing	4. An abnormal situation
• Notifying the CPU of external events • Free CPU from polling. • To control the computer from input device.	• Controls the flow of data to the output devices. • Notify the computer of the completion of an action.	• Allocating CPU time to different programs that are sharing the CPU.	• Usage is similar to external input events, but in this case, the events are directed at the problems within the computer system itself.
E.g.: Keyboard Input, Real-time or time sensitive system	E.g.: printer to control the flow of characters to the printer in an efficient way.	E.g.: Time sharing	E.g.: Power failure, Illegal instruction, Hardware error.

2. I/O Techniques

- **I/O techniques**

  (2) Interrupt-driven I/O

  - ❑ Servicing an interrupt

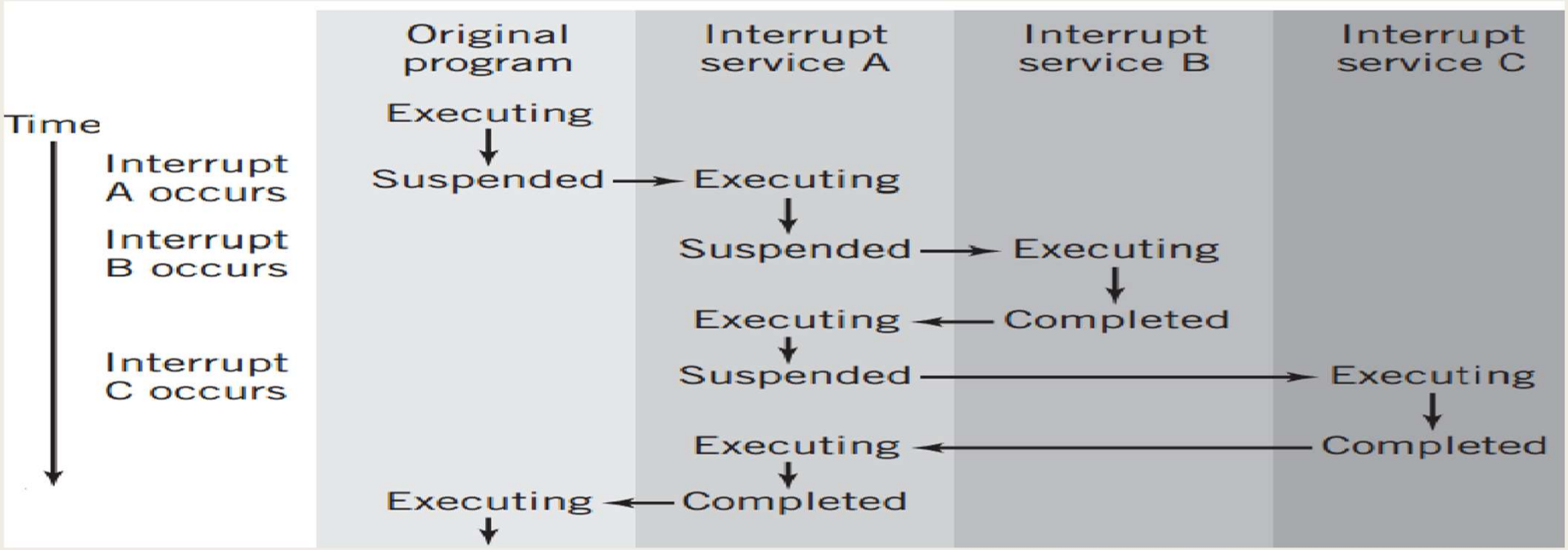
1. Suspend program in progress	2. Save context, including last instruction executed & data values in registers, in Process Control Block (PCB)	3. Branches to interrupt handler program (interrupt routine)	4. CPU resumes control to the interrupt program
			When the interrupt routine is complete, the registers are restored, including the program counter and original program resumes exactly where it left off

2. I/O Techniques

- **I/O techniques**

  (2) Interrupt-driven I/O

  - ❑ Servicing multiple interrupts
    - ❖ Can be handled by assigning priorities to each interrupt



2. I/O Techniques

- **I/O techniques**

  (2) Interrupt-driven I/O

  - 2 methods to identify devices that initiate interrupt:
    - Vectored interrupt

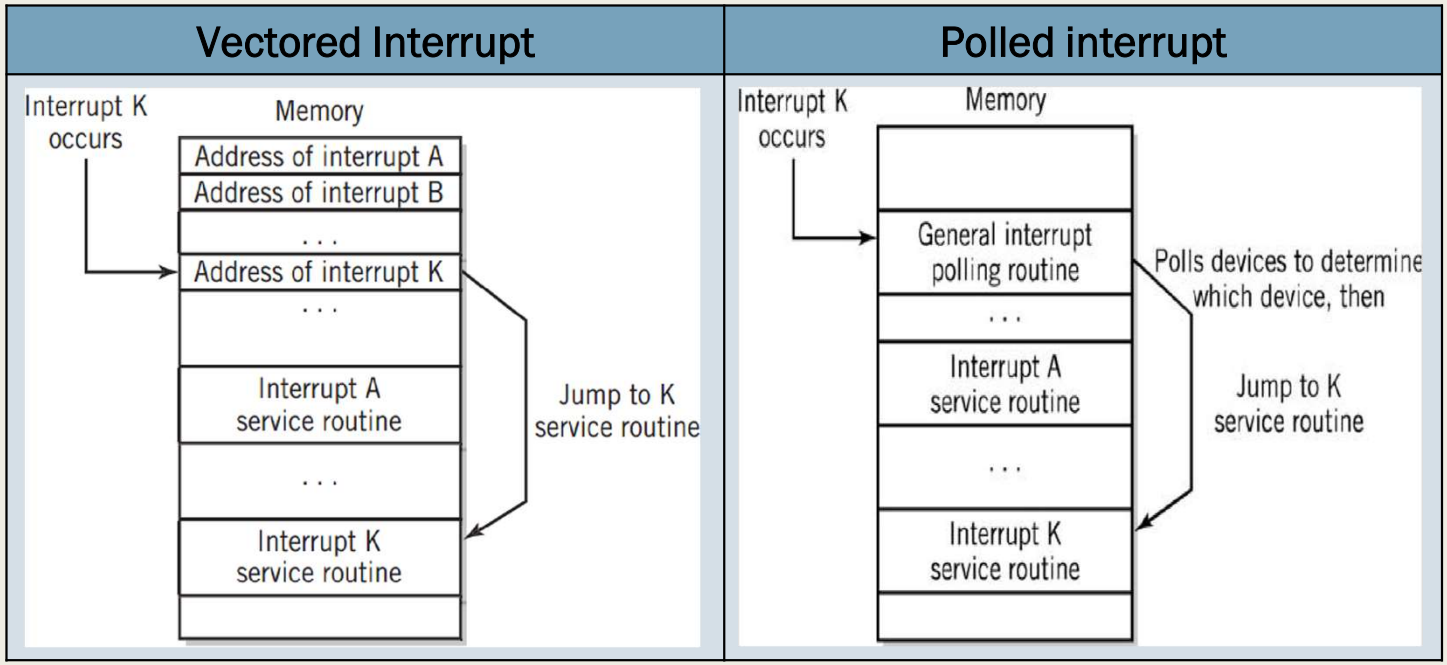
      The address of the interrupting device is included as part of the interrupt
    - Polled interrupt

      Uses the general interrupt that shared by all devices. CPU polls each device to identify the interrupt source.

2. I/O Techniques

- **I/O techniques**

  (2) Interrupt-driven I/O



2. I/O Techniques

- **I/O techniques**

  (3) Direct Memory Access (DMA)

  - ❑ DMA is a method of transferring data between peripherals & memory without using CPU
  - ❑ Transferring large block of data
  - ❑ Direct transfer, CPU not actively involved itself
  - ❑ There are 4 pieces of data required to initiate the DMA transfer:
    1. The location of data on I/O device
    2. The start location of the data block in memory
    3. The size of data to be transferred
    4. The direction of transfer (READ / WRITE)

2. I/O Techniques

- **I/O techniques**

(3) Direct Memory Access (DMA)

- For DMA to take place, 3 primary conditions must met:
  1. There must be a method of connect together the I/O interface & memory
  2. The I/O modules must be capable of reading & writing to memory
  3. There must be a mean to avoid conflict between the CPU & I/O modules
- E.g.: Consider a program that sorts a block of numbers
  - The entire block of umbers are transferred from HDD to memory
  - Then sorting is perform via memory
  - The sorted data is transferred back to HDD

2. I/O Techniques

- **I/O techniques**

  (3) Direct Memory Access (DMA)

  - Since the CPU is not actively involved during the transfer, the CPU can be used to perform other tasks during the time when I/O transfers are taking place
  - DMA is not limited to just disk-to-memory but also other high-speed devices
  - DMA takes several instructions to initiate a DMA transfer, therefore is not suitable for data transfer with small amount

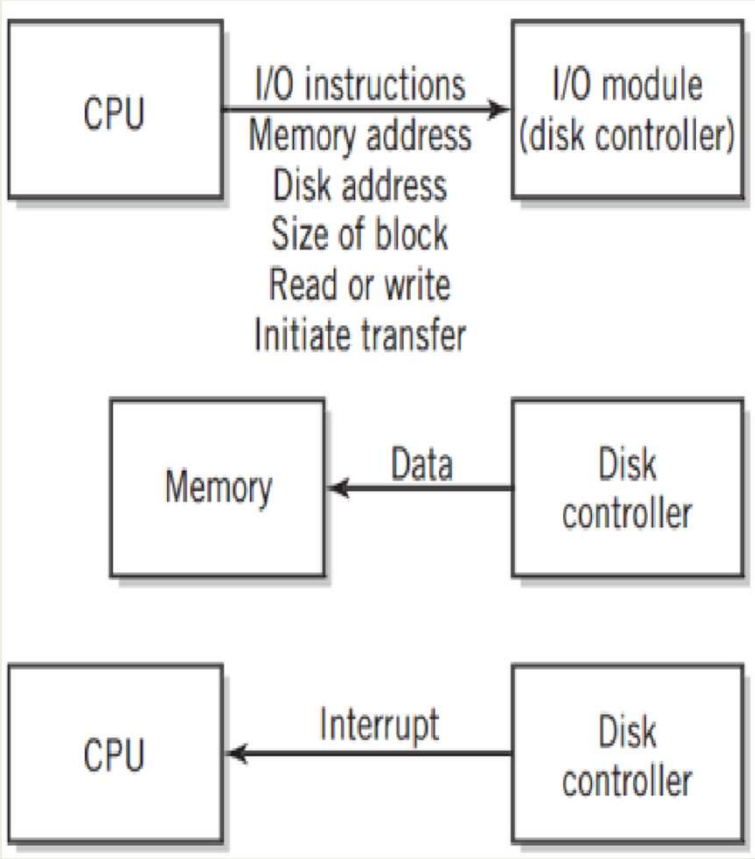
2. I/O Techniques

- **I/O techniques**

  (3) Direct Memory Access (DMA)

  - ❑ The working of DMA

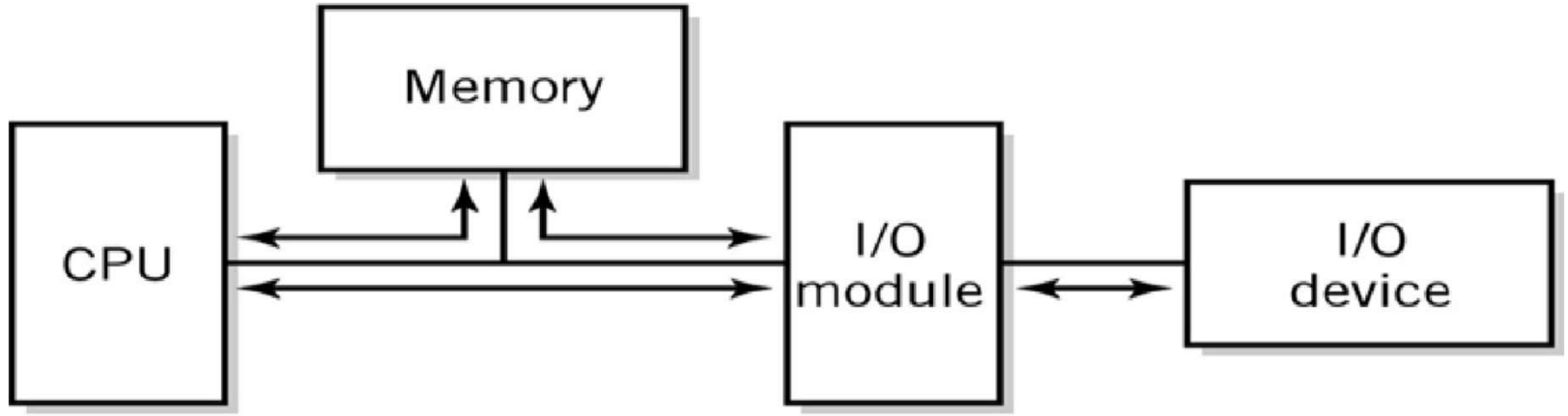
    1. Programmed I/O used to prepare I/O module for transfer by providing required information & initiating transfer
    2. DMA transfer. In this case, the data is transferred from disk to memory
    3. Upon completion, disk controller sends completion interrupt to CPU



3. I/O Architecture

3. I/O Architectures

- **Basic CPU-Memory-I/O Pathway**



- 2 key architectures
  - Bus architecture
  - Channel architecture

3. I/O Architectures

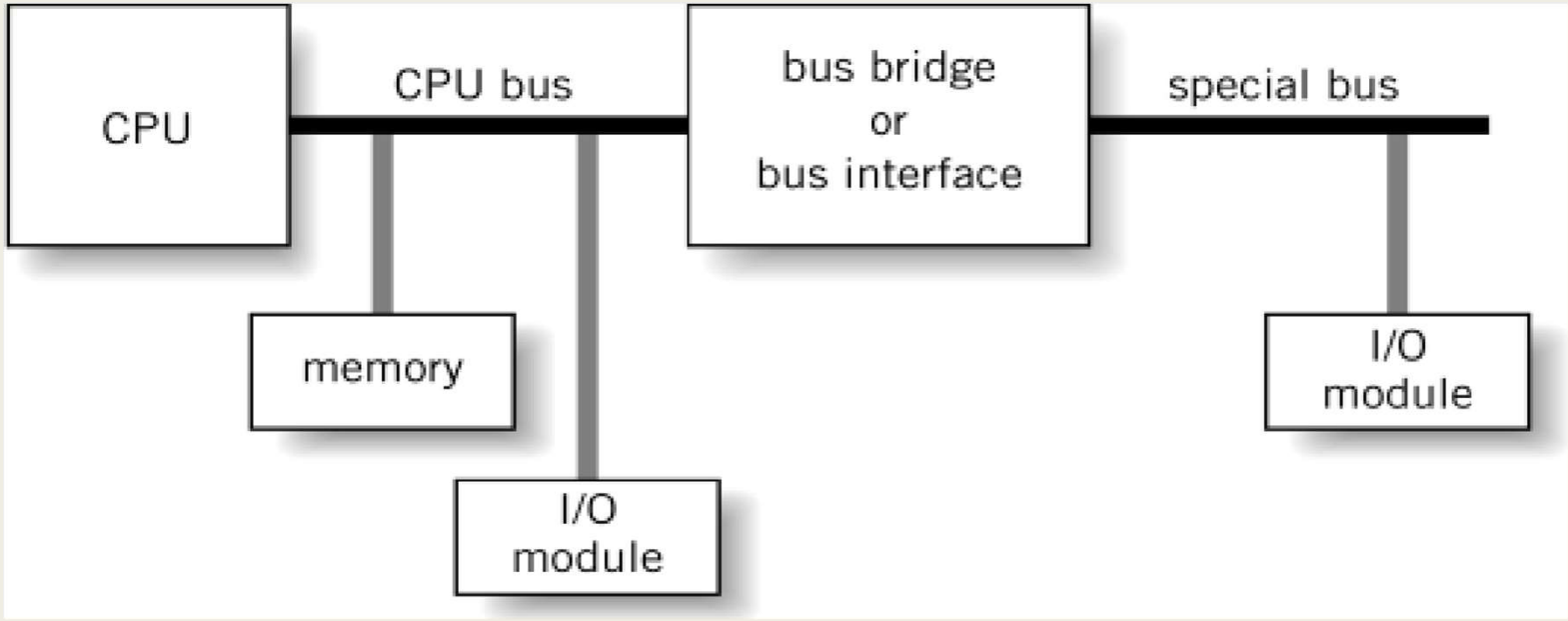
- **Basic CPU-Memory-I/O Pathway**
  - **Bus architecture**
    - Used in almost all personal computers
    - Uses backbone for connections of various components, memory and I/O to CPU
    - Simplest form of connection. Single system bus connects the CPU to memory & all to all various I/O modules that control I/O devices
    - Consists of a number of interconnected buses, e.g.: CPU bus, PCI bus, ISA bus, etc.

3. I/O Architectures

- **Basic CPU-Memory-I/O Pathway**
  - **Bus architecture**
    - These buses are interconnected by bus interface, e.g.: expansion bus interface, bus bridges, etc to expand the flexibility by converting bus signals from one to another
    - The interconnect-ability makes it possible to design & use industry standard buses on equipment of different vendors
      - ✔ Connection of I/O devices standardized across a wide range of equipment types and manufacturers
      - ✔ The major aspect of ***Open Architecture*** concept

3. I/O Architectures

- **Basic CPU-Memory-I/O Pathway**
  - **Bus architecture**

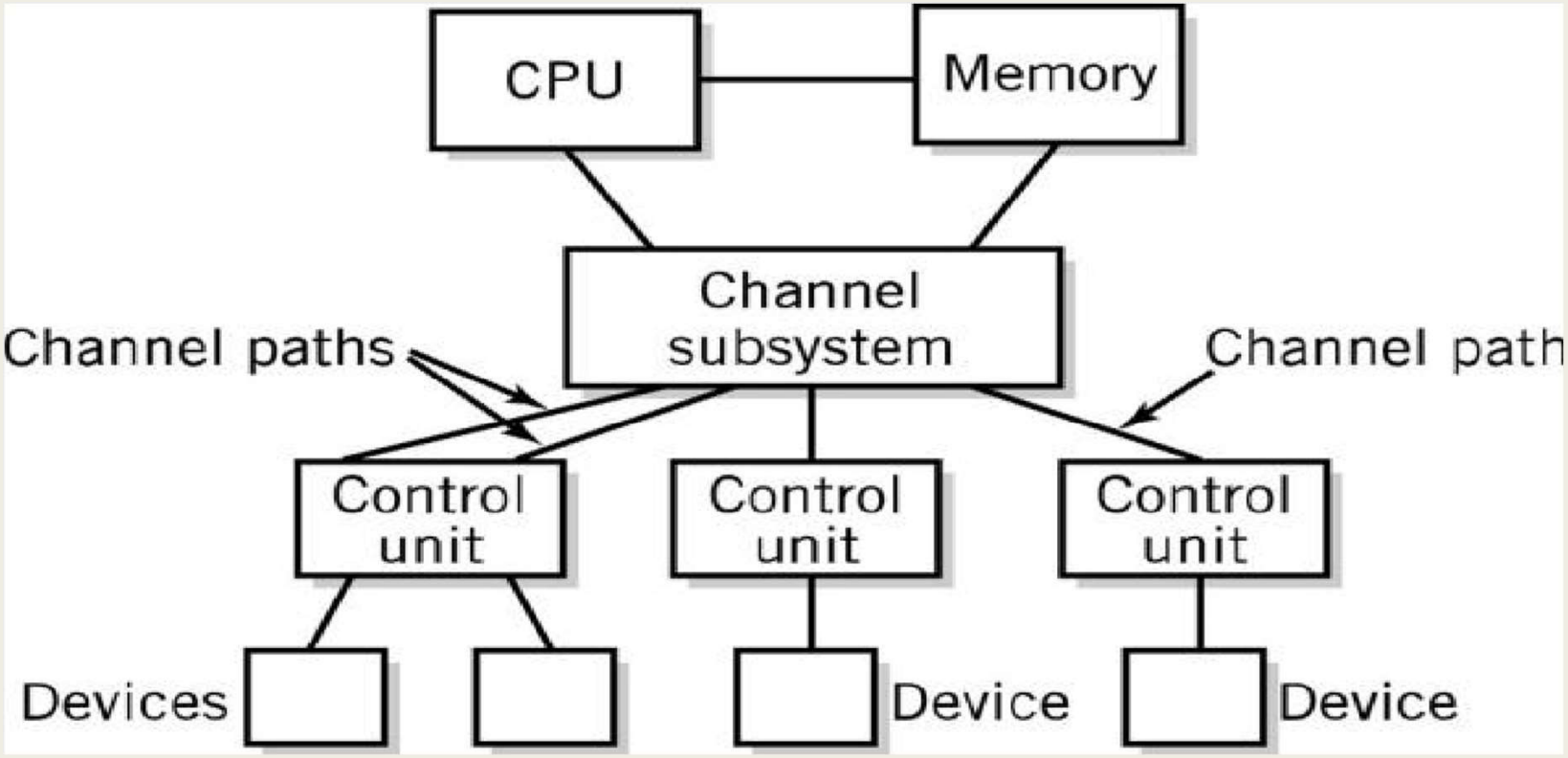


3. I/O Architectures

- **Basic CPU-Memory-I/O Pathway**
  - **Channel architecture**
    - Used in all IBM mainframe computers since late 70's
    - Uses channel subsystem / Separate I/O processor
      - ✔ Serves as a separate CPU for I/O operations
      - ✔ Channel control words
      - ✔ Programs stored in memory, independent of CPU
      - ✔ Transfer data between memory & and I/O device using DMA
    - Sub channel
      - ✔ Each of it is connected to a control unit module
      - ✔ Similar role to a device controller
    - Up to 8 different channel paths between channel subsystem & control unit (used as alternative, if busy)

3. I/O Architectures

- **Basic CPU-Memory-I/O Pathway**
  - ❑ **Channel architecture**



Chapter Review

Chapter Review

1. **I/O Modules**
   - ❑ I/O configuration
   - ❑ Functions of I/O modules
2. **I/O Handling Techniques**
   - ❑ Programmed I/O
   - ❑ Interrupt I/O
   - ❑ DMA
3. **I/O Architecture**
   - ❑ Bus architecture
   - ❑ Channel architecture