CD4002/CN4002 Computer Systems and Networks

Week 4 Input Output Methods

University of East London

Pioneering Futures Since 1898

Agenda

- Programmed I/O
- Interrupts
- Direct Memory Access

Basic Model

- Processing speed or program execution
  - determined primarily by ability of I/O operations to stay ahead of processor.

I/O Considerations

Speed Issues

- CPU operates at speeds much faster than the fastest I/O device
- Devices operate at different speeds
- Bursts of data
- Block data transfer required for some devices

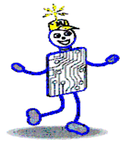
Coordination

- Several devices perform I/O simultaneously
- Unexpected input
- Various input formats
- Status information needed for each device

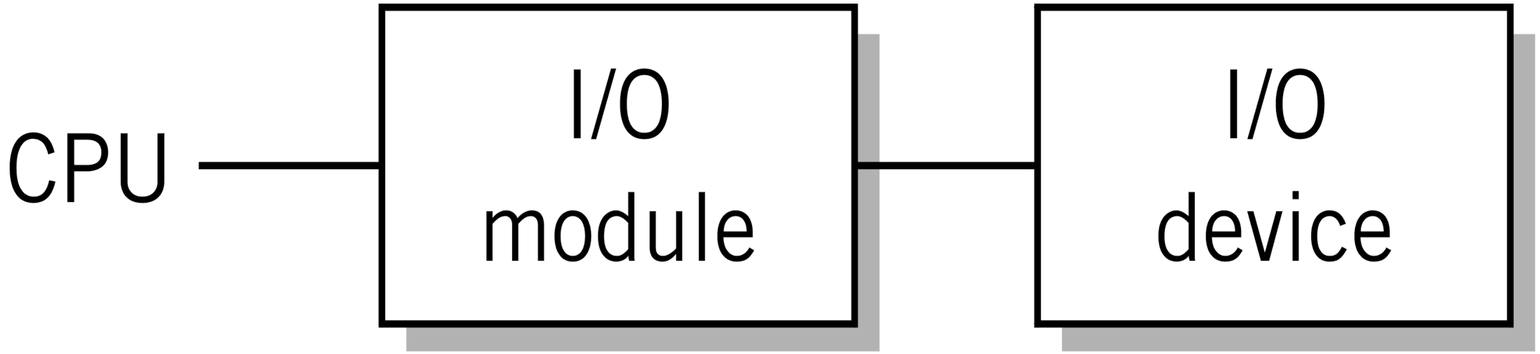
The Characteristics of Typical I/O Devices

Device	Input/Output	Data Rate	Control	Type
Keyboard	Input	Very low	External & program	Character
Mouse	Input	Low	External	Character
Sound	Input/Output	Medium	Program	Block burst or steady
Network	Input/Output	High to very high	External & program**	Block burst
Printer	Output	Low to medium	Program	Block burst
Graphics display	Output	High	Program	Steady
Flash drive	Storage	Medium	External & program**	Block burst
Magnetic disk	Storage	Medium	Program	Block burst
Solid state drive	Storage	Medium to high	Program	Block burst

** External initiation features, mostly program control



Examples of I/O Devices



Device	Input/Output	Data rate (in Kbytes/sec)	Type
Keyboard	Input	0.01	char
Mouse	Input	0.02	char
Voice input	Input	0.02	block burst
Scanner	Input	200	block burst
Voice output	Output	0.5	block burst
Inkjet printer	Output	1.5	block burst
Laser printer	Output	100-1,000	block burst
Graphics display	Output	30,000	block burst or steady
Local area network	Input or output	200-20,000	block burst or steady
Optical disk	Storage	500-15,000	block burst or steady
Magnetic tape	Storage	1,000-15,000	block burst or steady
Magnetic disk	Storage	2,000-60,000	block burst or steady
Non-compressed video source	Input	10,000	block steady
Non-compressed audio source	Input or output	100	block steady

I/O Modules Functions

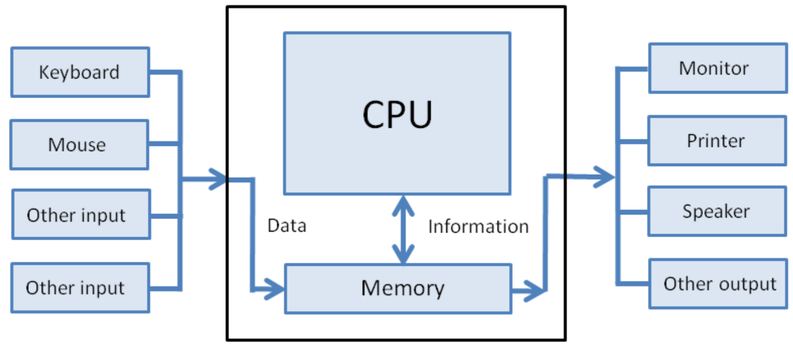
- Recognizes messages from device(s) addressed to it and accepts commands from the CPU
- Provides a buffer where the data from memory can be held until it can be transferred to the disk
- Provides the necessary registers and controls to perform a direct memory transfer
- Physically controls the device
- Copies data from its buffer to the device/from the CPU to its buffer
- Notifies with interrupts

Programmed I/O

- I/O data and address registers in CPU
- One word transfers
- Address information for each I/O device
  - LMC I/O capability for 100 devices
- Full instruction fetch/execute cycle
- Primary use:
  - Keyboards and similar devices
  - Communication with I/O controllers

Input/Output Modules

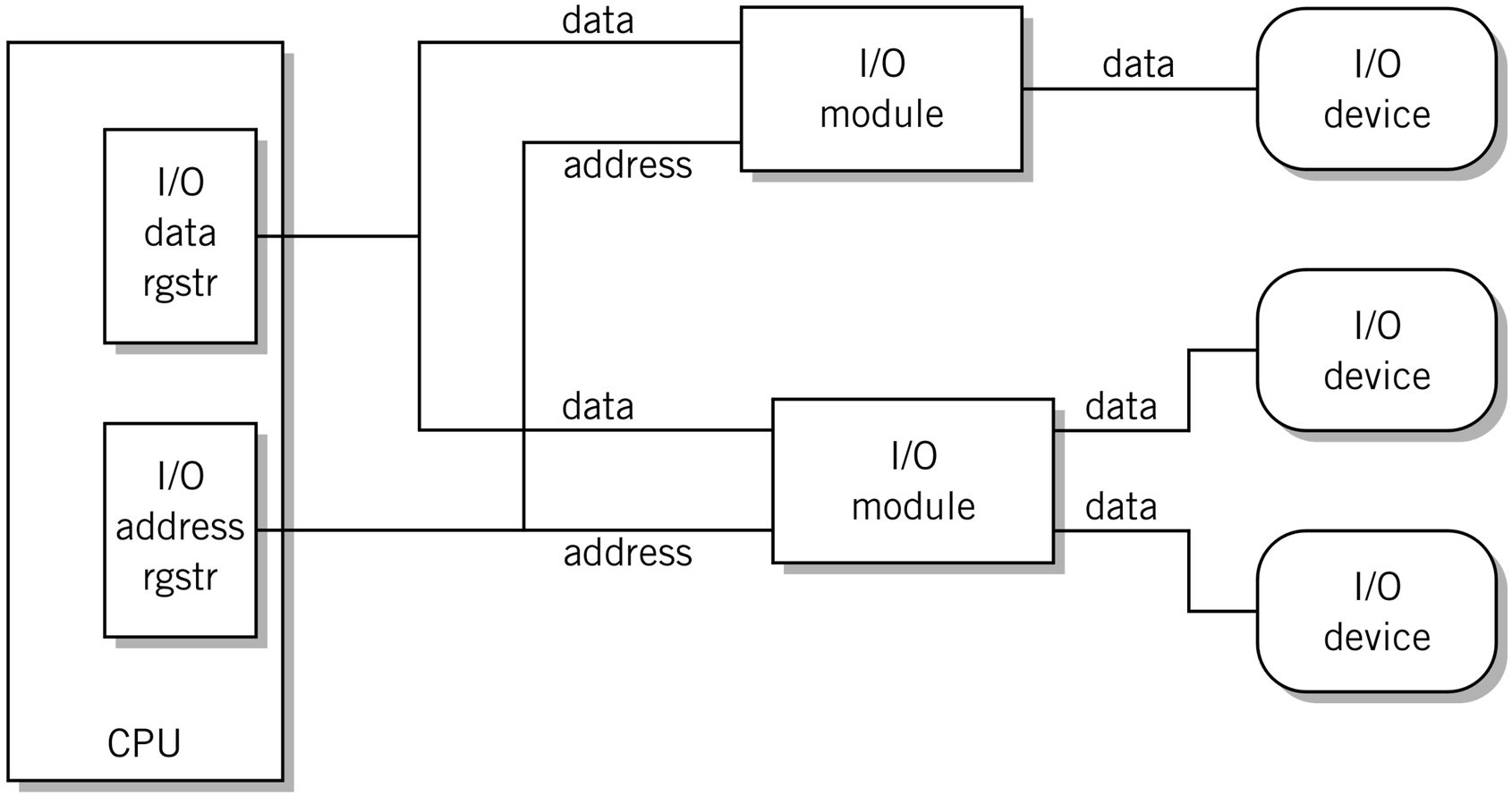
- Programmed I/O
  - CPU controlled I/O
- Interrupt Driven I/O
  - External input controls
- Direct Memory Access Controllers
  - Method for transferring data between main memory and a device that bypasses the CPU



Programmed I/O

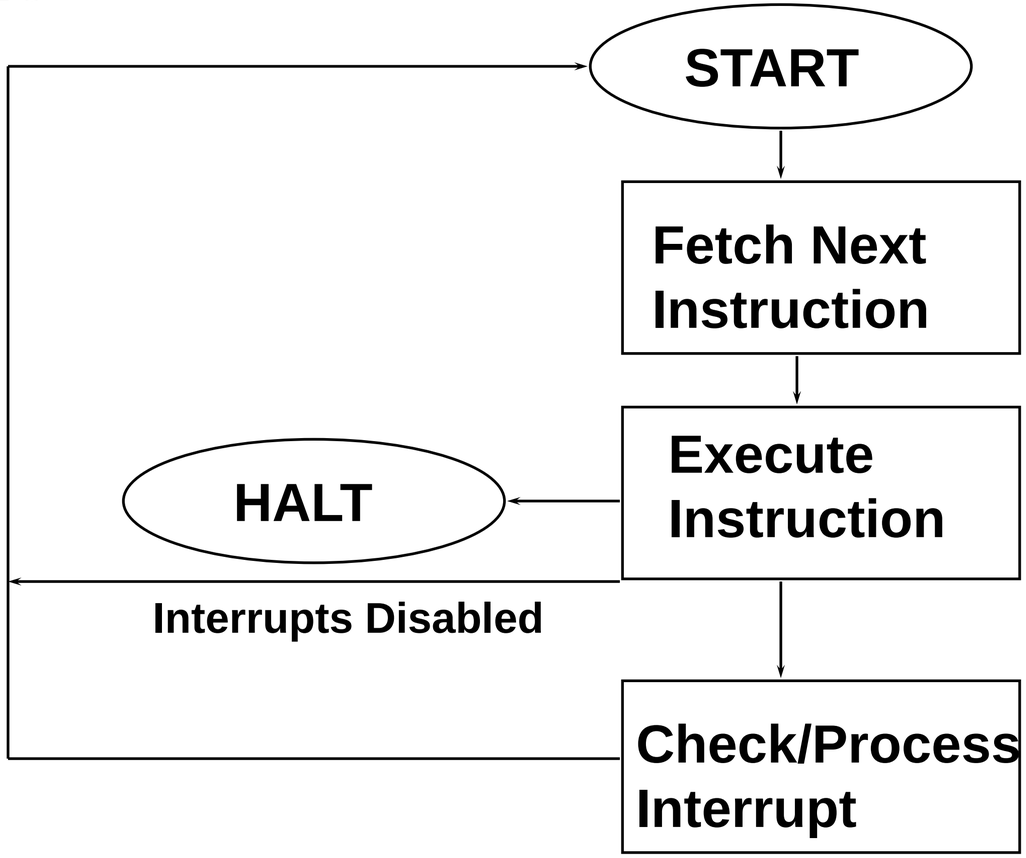
- I/O data and address registers in CPU
- One word transfers
- Address information for each I/O device
  - LMC I/O capability for 100 devices
- Full instruction fetch/execute cycle
- Primary use:
  - keyboards
  - communication with I/O modules (see DMA)

Programmed I/O



The CPU - The Interrupt Cycle

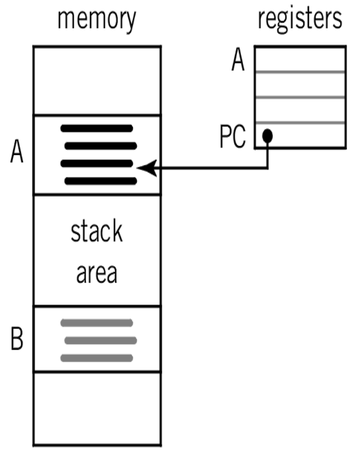
- Fetch / Execute cycle
- Interrupt cycle



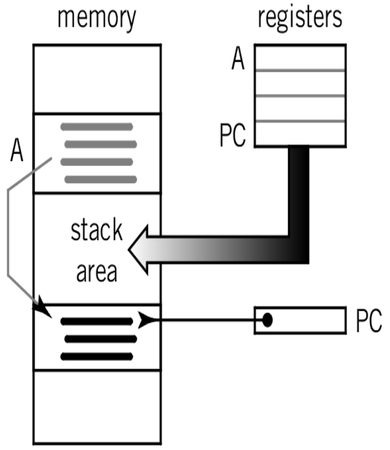
Interrupt Terminology

- Interrupt lines (hardware)
- Interrupt request
- Interrupt handlers
  - Program that services the interrupt
  - Also known as an interrupt routine
- Process Control Block (PCB)
  - Located in a part of memory known as the stack area
  - All registers of a program are saved here before control is transferred to the interrupt handler
- Abnormal event indicator (generated from within the CPU)
  - Illegal operations, hardware error
- Software interrupts

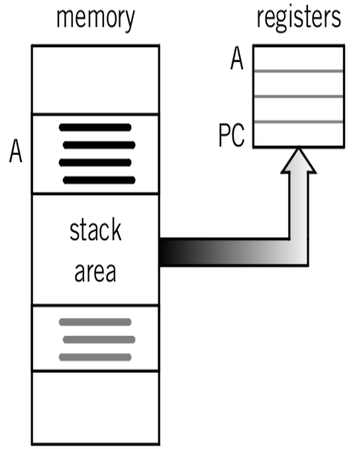
Servicing an Interrupt



1. Before interrupt arrives, program A is executing. The program counter points to the current instruction.



2. When the interrupt is received by the CPU, the current instruction is completed, all the registers are saved in the stack area (or in a special area known as a process control block). The PC is loaded with the starting location of program B, the interrupt handler program. This causes a jump to program B, which becomes the executing program.



3. When the interrupt routine is complete, the registers are restored, including the program counter, and the original program resumes exactly where it left off.

Quiz Time

Q. For which of the following devices is programmed I/O suitable? (Choose ALL that apply.)

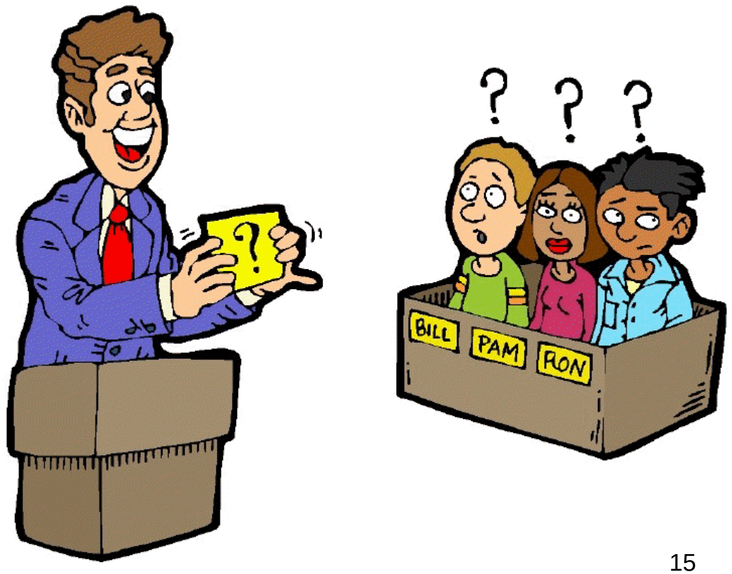
a. Keyboards

b. Mice

c. Sound cards

d. Flash drives

e. Solid state drives



Quiz Time

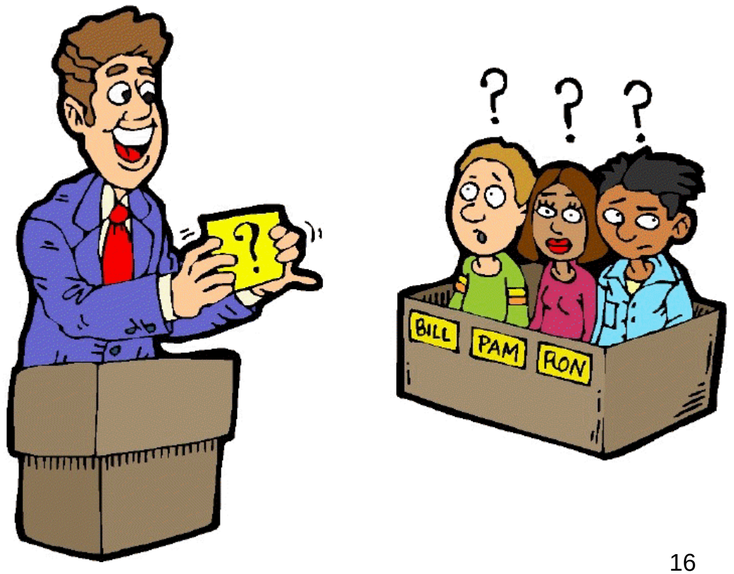
Q. Which of the following statements is TRUE? Programmed I/O (Choose ALL that apply.)

a. Is controlled by the CPU.

b. Transfers data one word at a time.

c. Requires a full fetch execute cycle for every word transferred.

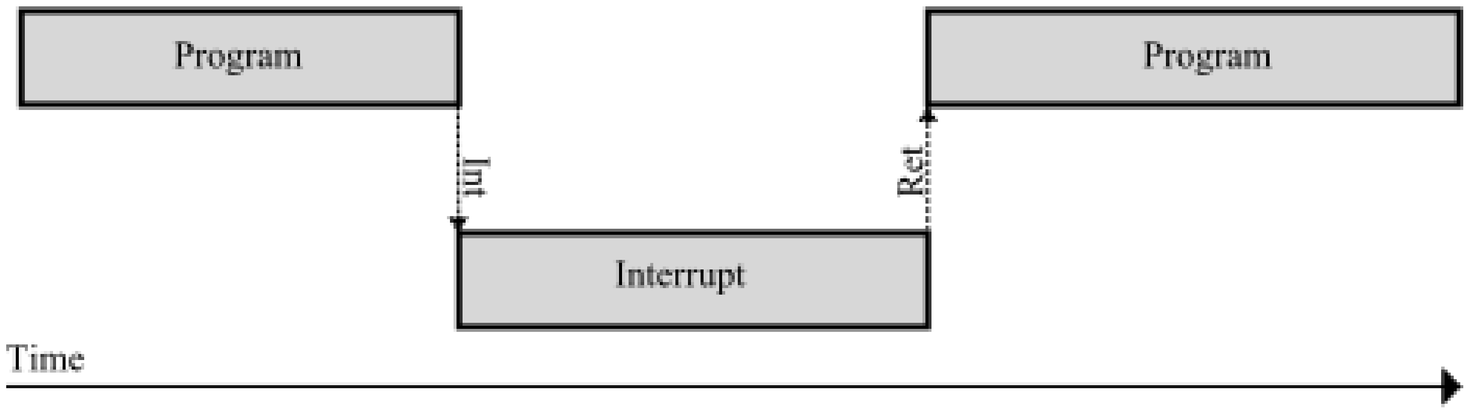
d. Requires the use of I/O address and data registers within the CPU.



Interrupts

- Signals that cause the CPU to suspend execution of the current program
  - free CPU from waiting for events
  - provide control for external input

- Examples
  - unexpected input
  - abnormal situations
  - illegal instructions
  - multitasking



Interrupt Terminology

- Interrupt lines (hardware)
- Interrupt requests
- Interrupt handlers
  - Programs that service interrupts
  - Also known as interrupt routines
- Process Control Block (PCB)
  - Located in a part of memory known as the stack area
  - All registers of a program are saved here

Interrupt Terminology (Contd.)

- Servicing interrupts
  - Executing program is suspended
  - Pertinent information including last instruction executed and data values in registers are saved in PCB
  - Control of CPU passes to interrupt handler

Quiz Time

Q. Which of the following devices use interrupts? (Choose ALL that apply.)

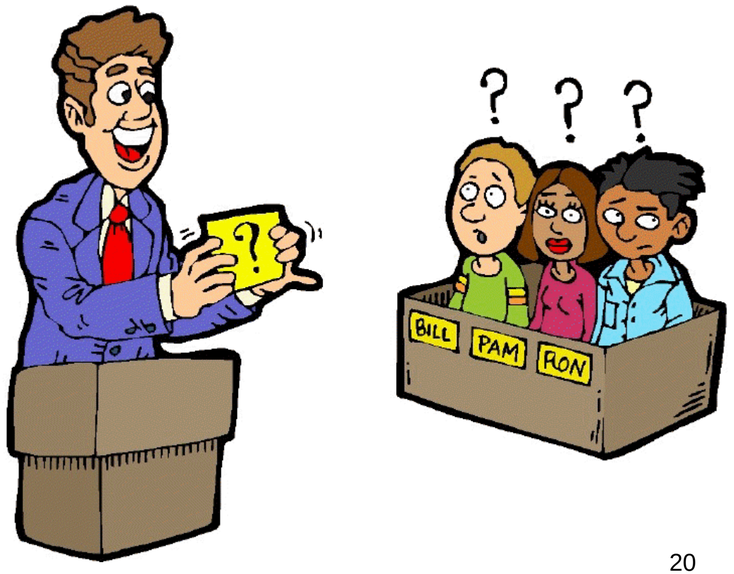
a. Keyboards

b. Mice

c. Printers

d. Flash drives

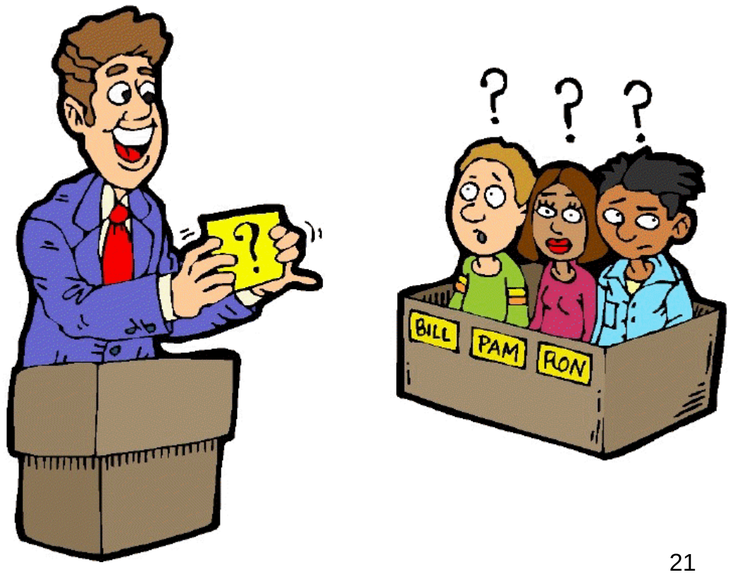
e. Magnetic drives



Quiz Time

Q. It is more efficient to use interrupts than polling as a way of notifying a CPU of an external event. Why?

CLUE: How often does a typical device interrupt the CPU?



Quiz Time

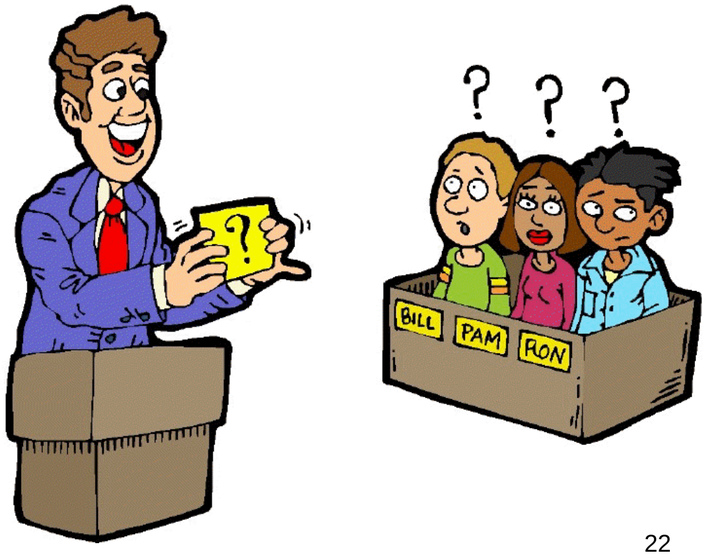
Q. To write data to a hard disk, a user program must generate a software interrupt which is handled by the operating system. Why?

a. Because disk I/O can only be performed using privileged instructions.

b. To avoid conflicts i.e. 2 programs writing to the same file simultaneously.

c. Both a and b

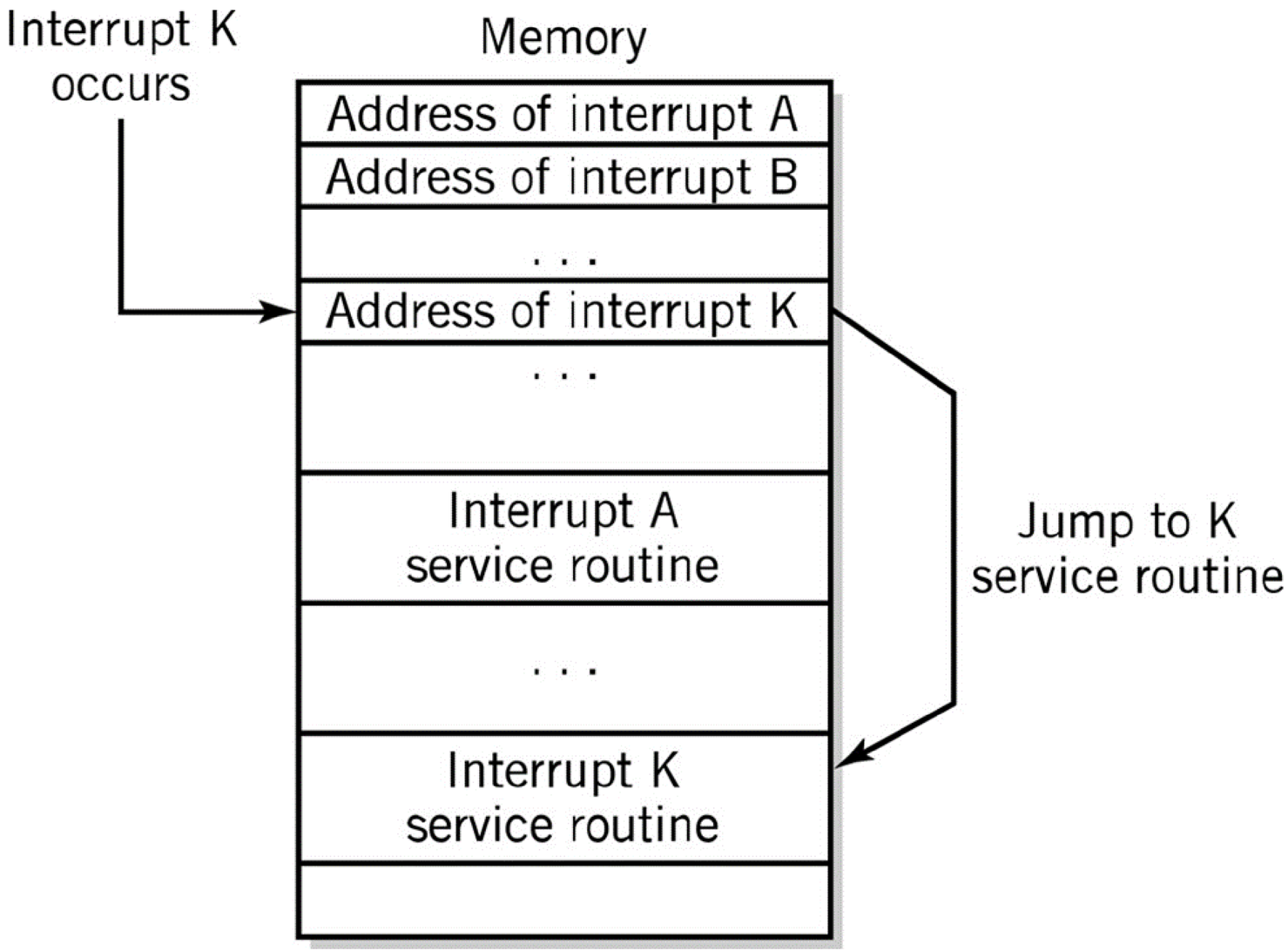
d. Neither a nor b



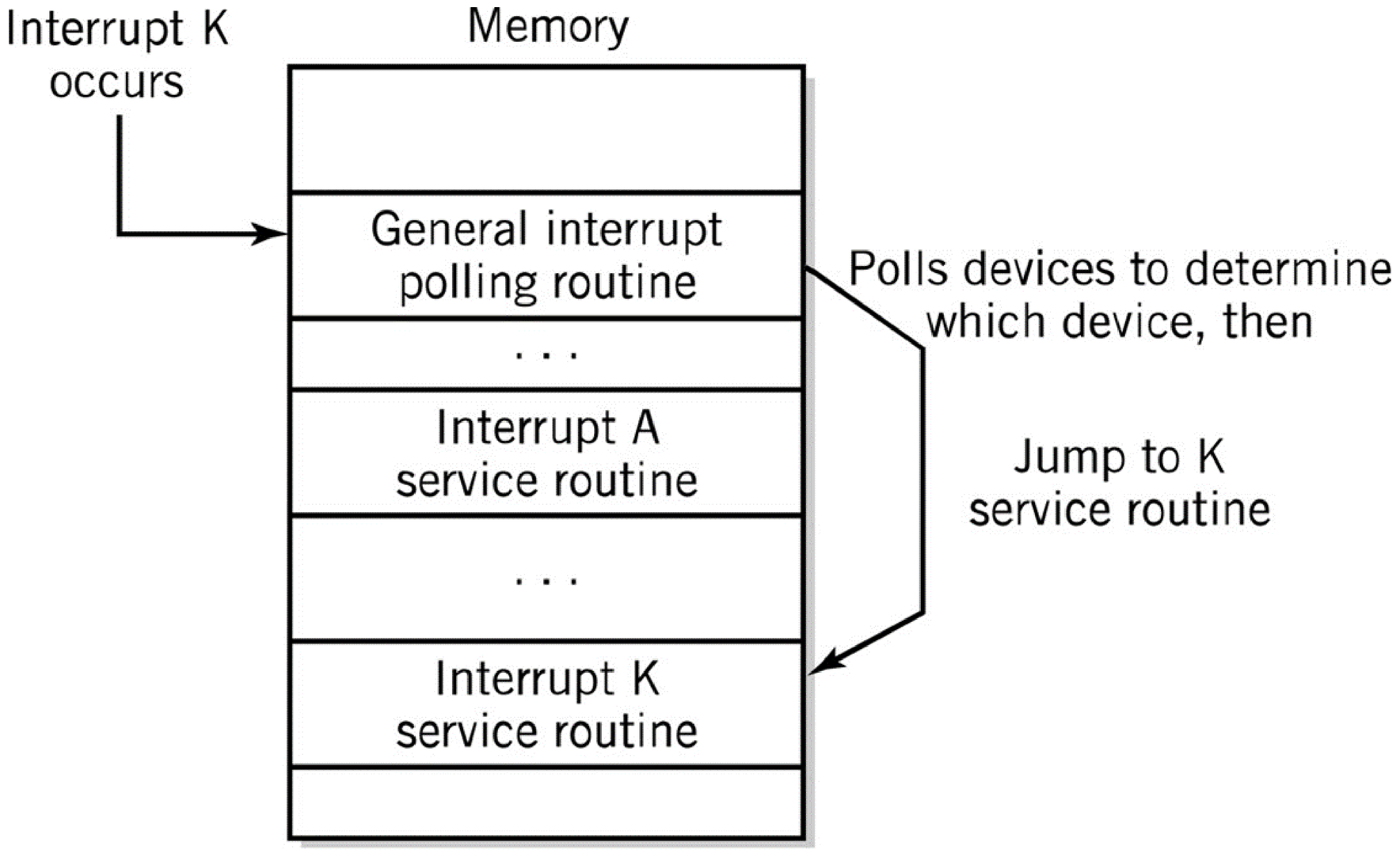
Multiple Interrupts

- Identifying the interrupting device
  - Vectored interrupts (include address of interrupting device)
  - Polling (checking for input in rotation)
- Interrupt priorities
  - Loss of data vs. task completion
- Maskable vs. non-maskable interrupts

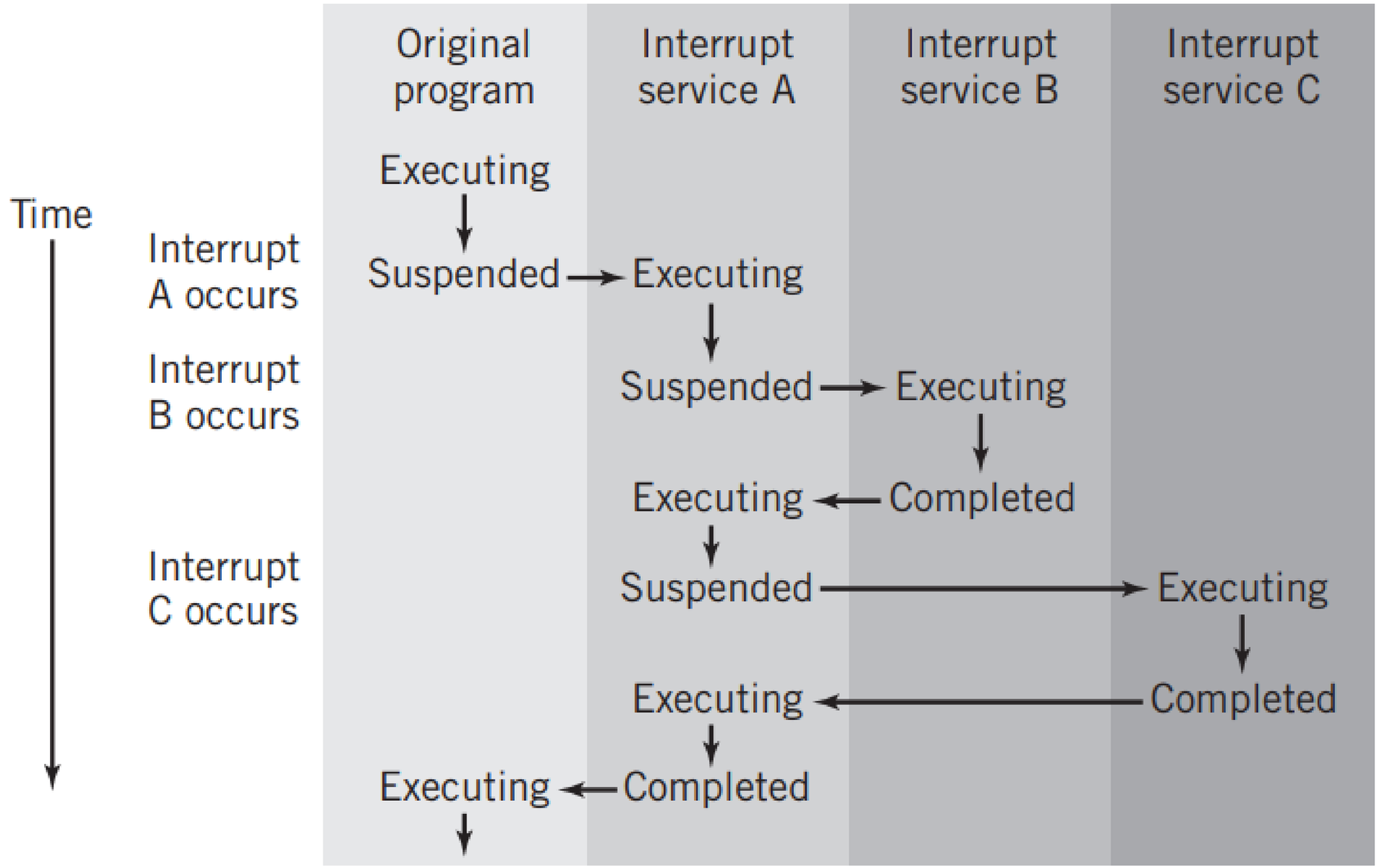
Vectored Interrupts



Polled Interrupts

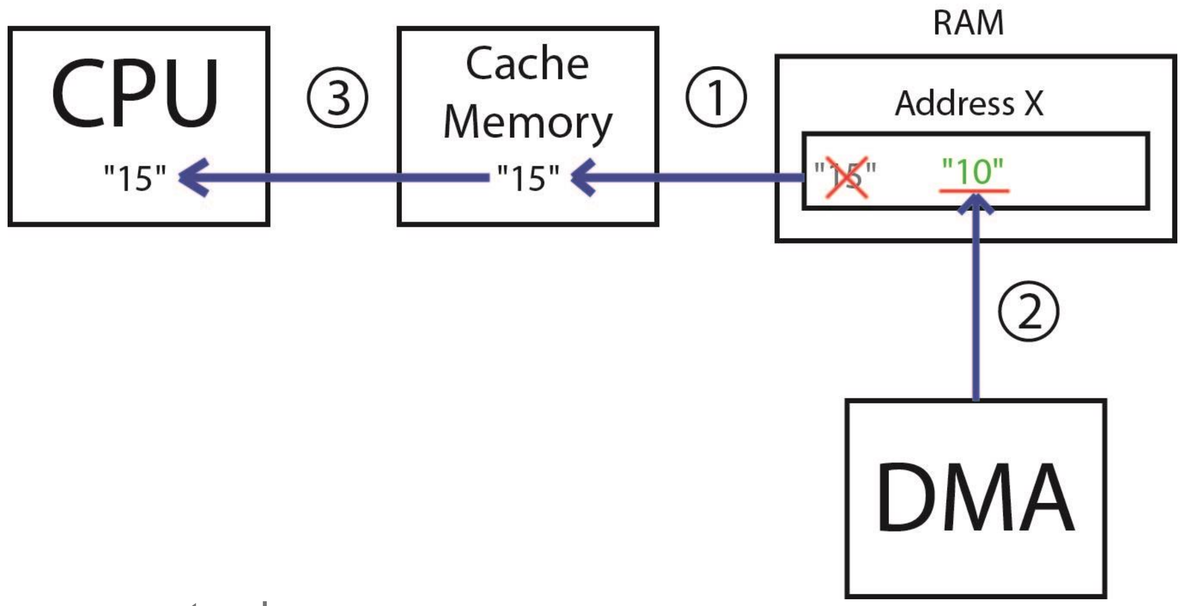


Multiple Interrupts



Direct Memory Access

- Transferring large blocks of data
- Direct transfer to and from memory
- CPU not actively involved in transfer itself
- Required conditions for DMA
  - The I/O interface and memory must be connected
  - The I/O module must be capable of reading and writing to memory
  - Conflicts between the CPU and the I/O module must be avoided



Quiz Time

Q. Which of the following devices uses DMA? (Choose ALL that apply.)

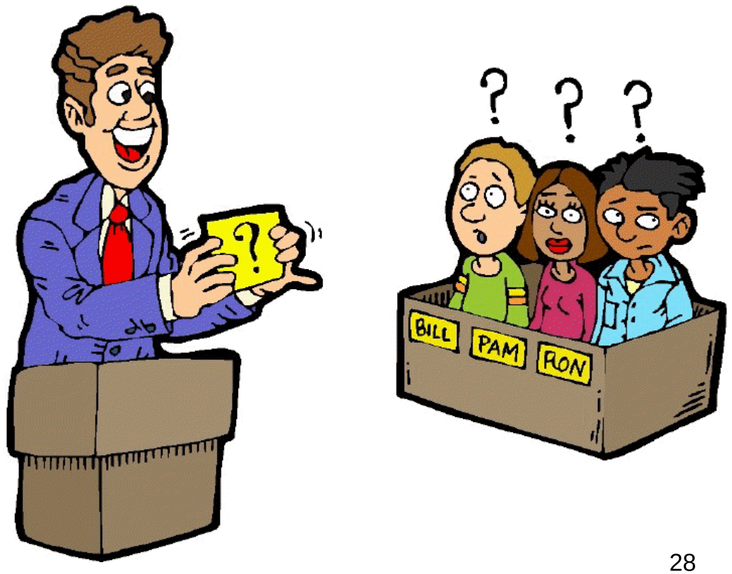
a. Disk drives

b. Graphics cards

c. Network cards

d. Mice

e. Sound cards



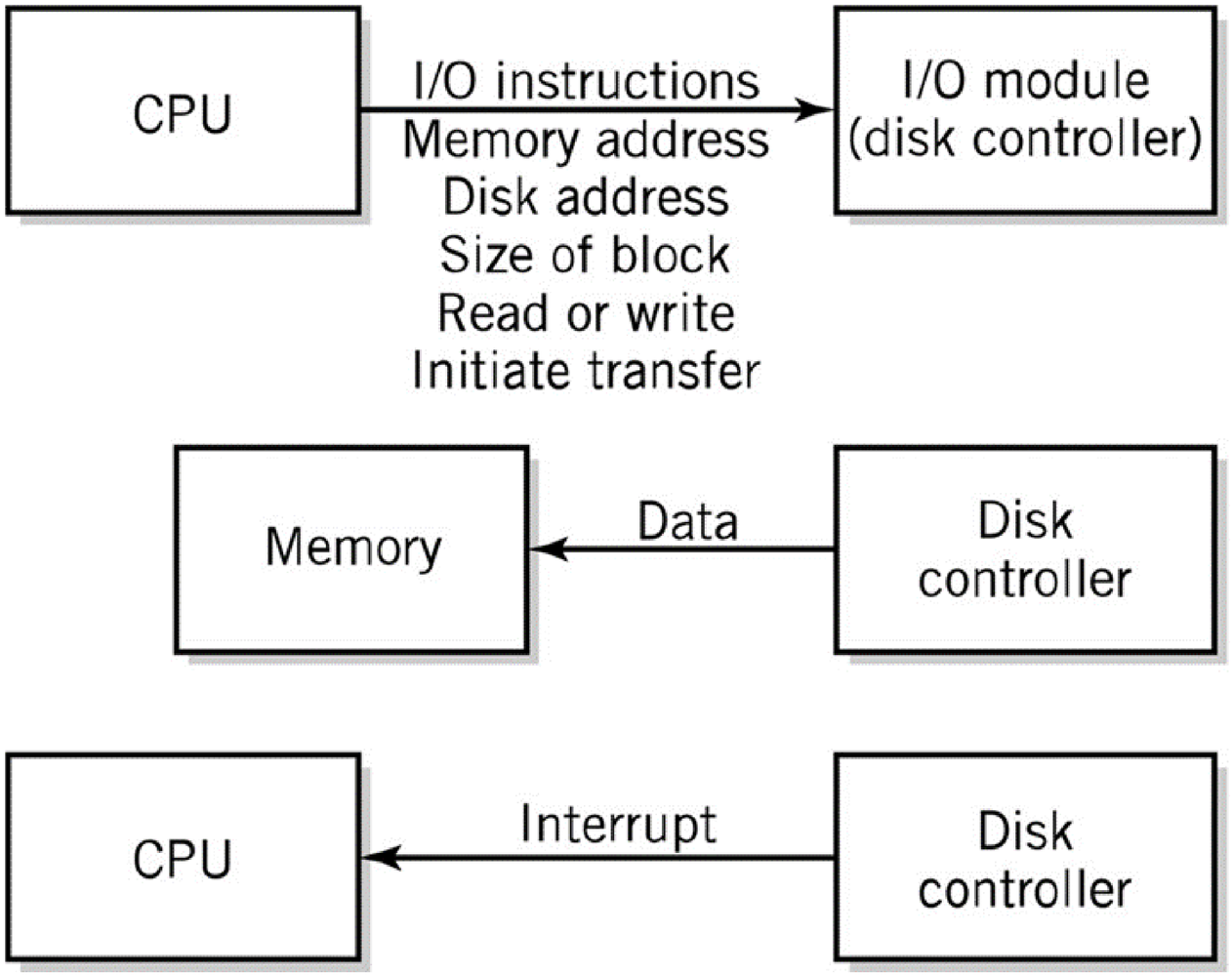
The DMA Process

- Application program requests I/O service from operating system
  - privileged instructions

- To initiate DMA, programmed I/O is used to send the following information:
  1. location of data on I/O device
  2. the starting location in memory
  3. the size of the block
  4. Read or write operation

- Interrupt to CPU upon completion

DMA Initiation and Control

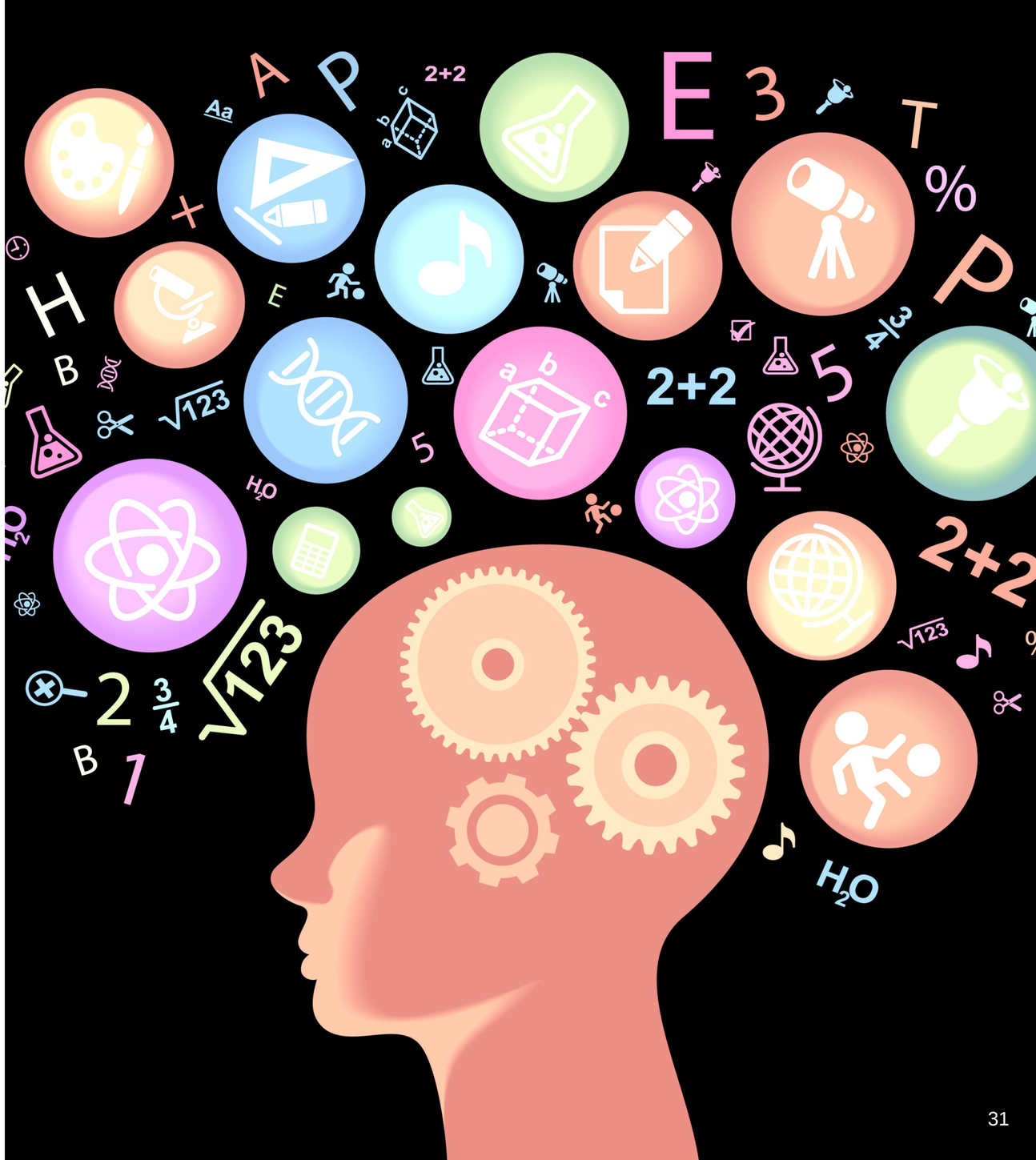


1. Programmed I/O used to prepare I/O module for transfer by providing required information and initiating transfer.
2. DMA transfer. In this case data is transferred from disk to memory.
3. Upon completion, disk controller sends **completion** interrupt to CPU.

Learning Objectives

On completion of this topic, you will be able to:

- Explain how the different methods of input output work
- Explain why the different input output methods are needed and give examples of their use



Directed Reading

- Englander, Chapter 9
- Stallings, Chapter 7