Introduction to System Programming

Computer System Overview, Operating System Overview.

Outline

- Basic Elements
  - Processor Registers
  - Instruction Execution
  - Interrupts
  - Memory Hierarchy
  - Cache Memory

Outline

- Operating System
  - Introduction
  - Objectives
  - Functions
  - Evolution
  - Major Achievements
  - Characteristics of Modern Operating System

Computer Software

- Computer software can be divided into two main categories: application software and system software.
- Application software consists of the programs for performing tasks particular to the machine's utilization.
  - This software is designed to solve a particular problem for users.
  - Examples of application software include spreadsheets, database systems, desktop publishing systems, program development software, and games.

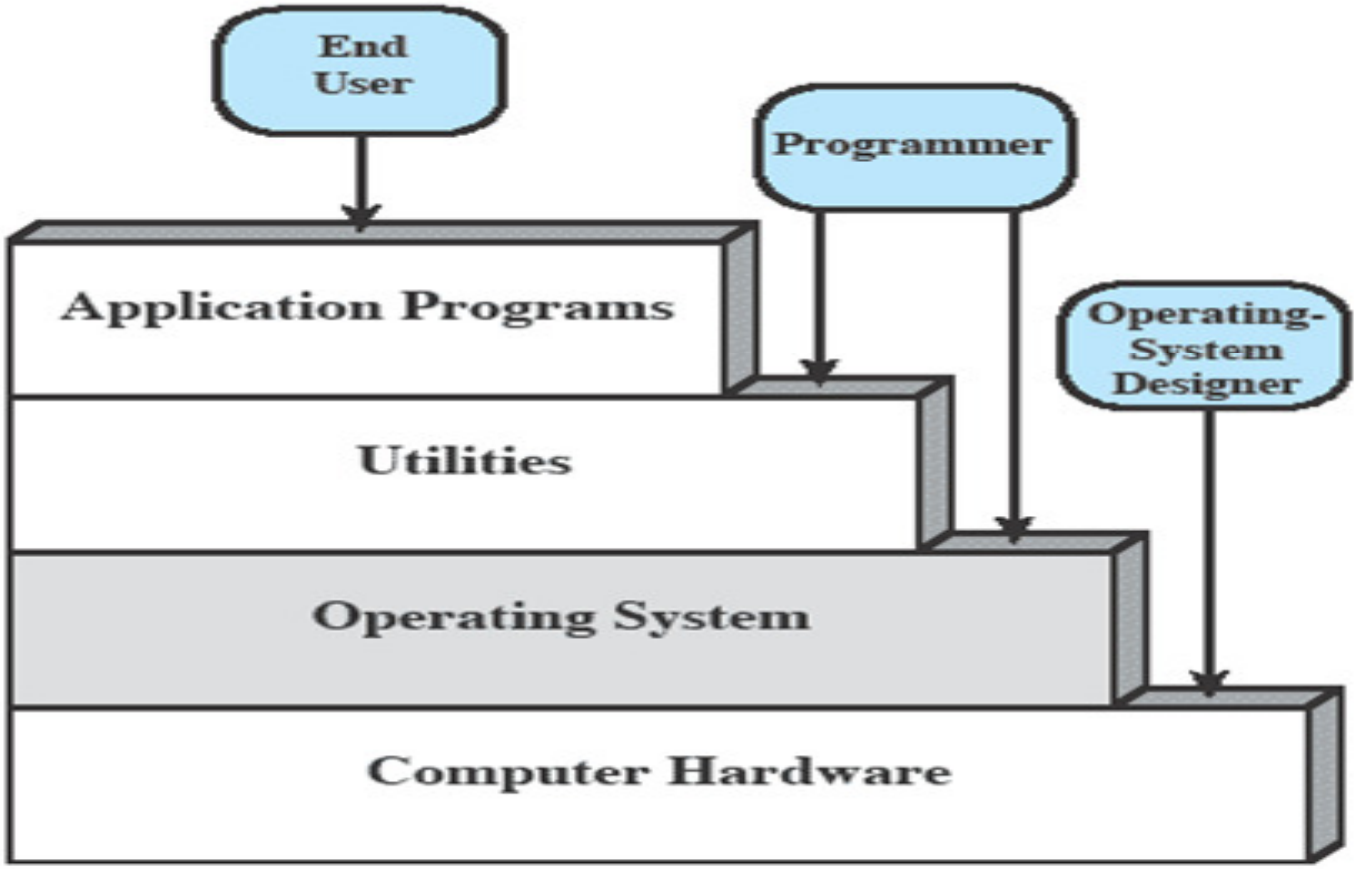
Computer Software

- On the other hand, system software is more transparent and less noticed by the typical computer user.
  - This software provides a general programming environment in which programmers can create specific applications to suit their needs.
  - This environment provides new functions that are not available at the hardware level and performs tasks related to executing the application program.
  - System software acts as an interface between the hardware of the computer and the application software that users need to run on the computer.
  - The most important type of system software is the operating system.

Operating System

- An **Operating System (OS) is a collection of programs that acts as an interface** between a user of a computer and the computer hardware.
- The purpose of an operating system is to provide an environment in which a user may execute the programs.
- Exploits the hardware resources of one or more processors.
- Provides a set of services to system users.
- Manages secondary memory and I/O devices

Layered View of Computer System



Layers and Views of a Computer System

Operating System

- A computer's operating system is a group of programs designed to serve two basic purposes:
  - To control the allocation and use of the computing system's resources among the various users and tasks.
  - To provide an interface between the computer hardware and the programmer that simplifies and makes feasible the creation, coding, debugging, and maintenance of application programs.

Operating System Functions

- Should act as a command interpreter by providing a user friendly environment.
- Should facilitate communication with other users.
- Facilitate the directory/file creation along with the security option.
- Provide routines that handle the intricate details of I/O programming.
- Provide access to compilers to translate programs from high-level languages to machine language.
- Provide a loader program to move the compiled program code to the computer's memory for execution.

Operating System Functions

- Assure that when there are several active processes in the computer, each will get fair and non-interfering access to the central processing unit for execution.
- Take care of storage and device allocation.
- Provide for long term storage of user information in the form of files.
- Permit system resources to be shared among users when appropriate, and be protected from unauthorized or mischievous intervention as necessary.

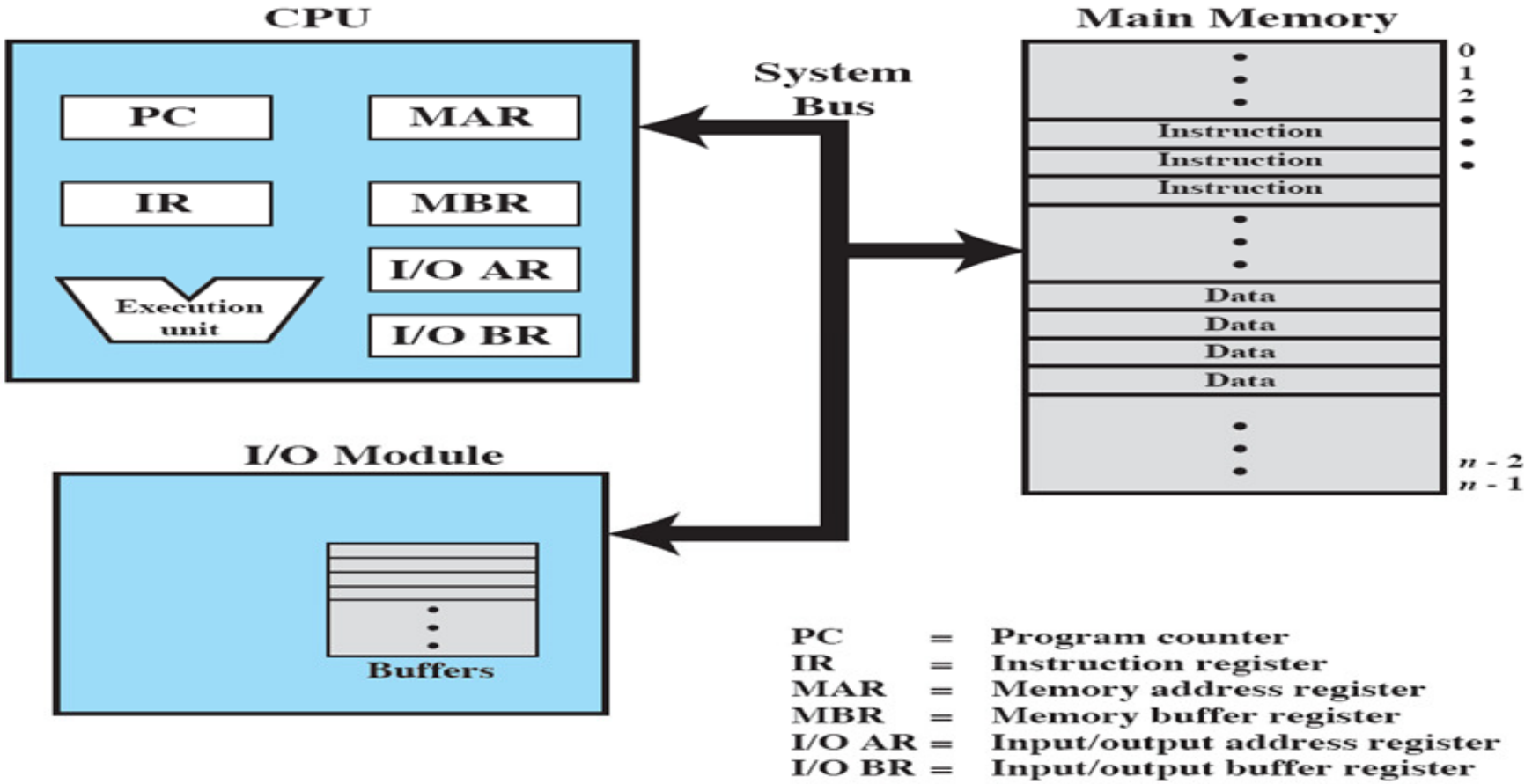
Goals Of Operating System

- The primary goal of an operating system is to make the computer **convenient to** use.
- The secondary goal is to use the hardware in an **efficient manner.**

Basic Elements

- The Four main structural elements:
  - Processor: Controls the operation of the computer and performs its data processing functions. (CPU)
  - Main Memory: Stores data and program.
    - It is volatile, referred as primary memory.
  - I/O Modules: Moves data between the computer and its external environment.
  - System Bus: Provides for communication among processor, main memory and I/O modules.

Computer Components: Top-Level View



Computer Components: Top-Level View

Processor Register

- A processor includes a set of register that provide memory that is faster and smaller than main memory.
- The processor registers serves two functions:
  - The **User-visible registers** : Enable programmer to minimize main memory references by optimizing register use.
  - **Control and status registers**: Used by processor to control operations of the processor and by privileged OS routines to control the execution of programs.

User-Visible Register

- May be referenced by machine language processor
- Available to all programs – application programs and system programs
- Typical Registers available are :
  - **Data Register** : can be assigned to a variety of functions by the programmer
  - **Address Registers** : contains main memory addresses of data and instructions, or portion of address used for calculating full address. These registers can be used for general purpose or devoted to particular way, they are Index registers, Segment Pointer, Stack Pointer.

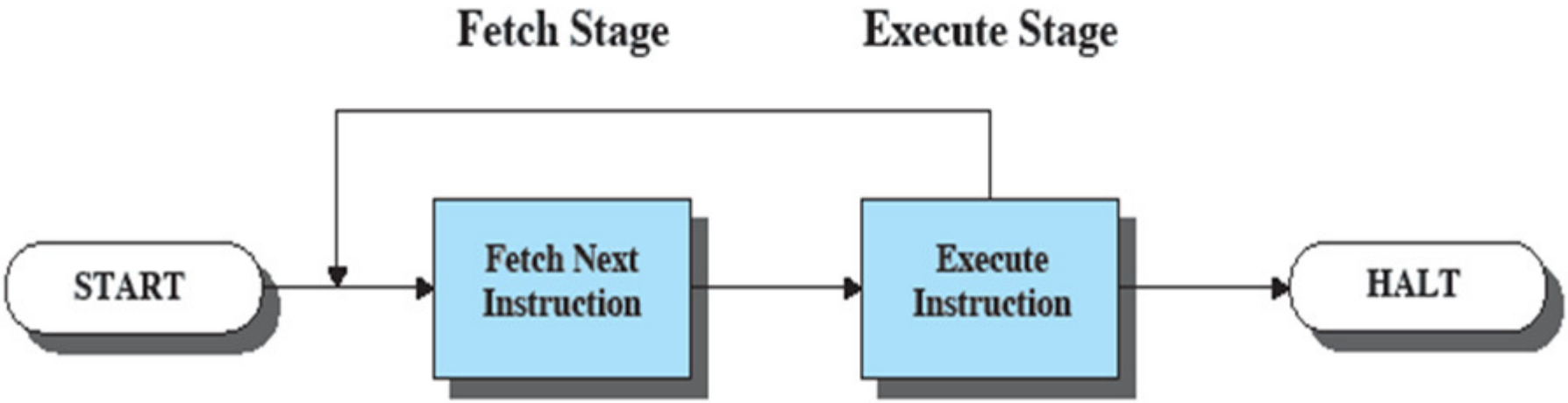
Control and Status Register

- Used to control the operations of the processor.
- **Program counter (PC)** : Contains the address of an instruction to be fetched
- **Instruction register (IR)** :Contains the instruction most recently fetched
- **Program status word (PSW)**: Contains status information

Instruction Execution

- A program to be executed by a processor consists of a set of instructions stored in memory.
- The instructions processing consists of two steps:
  - Processor reads (fetches) instructions from memory one at a time.
  - Processor executes each instruction
- The processing required for a single instruction is called an *instruction cycle*.

Basic Instruction Cycle



Basic Instruction Cycle

Instruction Fetch and Execute

- The processor fetches the instruction from memory.
- Program counter (PC) holds address of the instruction to be fetched next.
- PC is incremented after each fetch.
- The fetched instruction is loaded into the instruction register (IR).
  - The instructions contains bits that specify the action the processor is to take. The actions fall into 4 categories : Processor-memory, Processor-I/O, Data processing, Control.

Characteristics of a Hypothetical Machine

(a) Instruction format

(b) Integer format

Program counter (PC) = Address of instruction
Instruction register (IR) = Instruction being executed
Accumulator (AC) = Temporary storage

(c) Internal CPU registers

0001 = Load AC from memory
0010 = Store AC to memory
0101 = Add to AC from memory

(d) Partial list of opcodes

Characteristics of a Hypothetical Machine

Interrupts

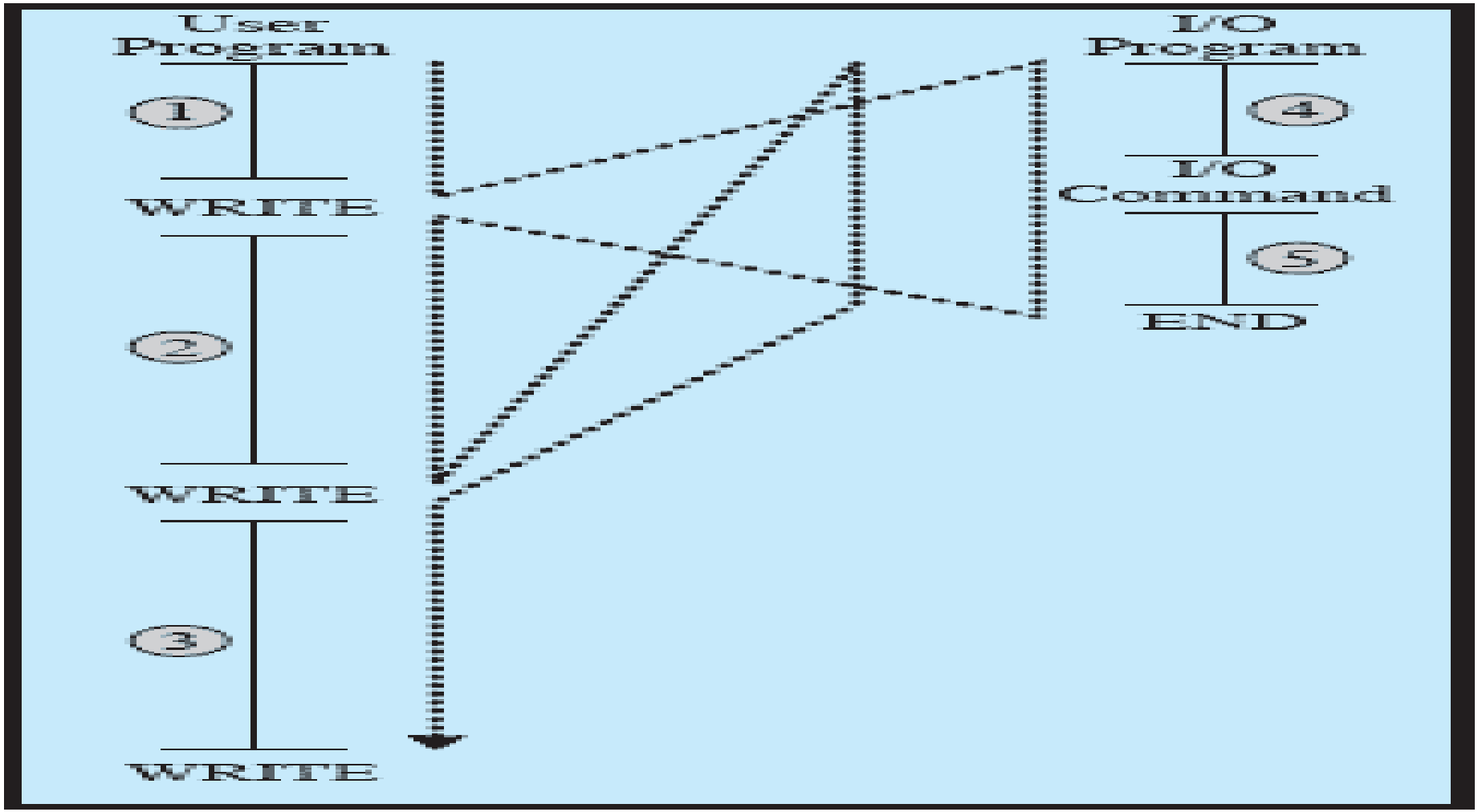
- Interrupt the normal sequencing of the processor
- Most I/O devices are slower than the processor
  - Processor must pause to wait for device
- Steps of handling interrupt

1. Hardware stacks program counter, etc.
2. Hardware loads new program counter from interrupt vector.
3. Assembly language procedure saves registers.
4. Assembly language procedure sets up new stack.
5. C interrupt service runs (typically reads and buffers input).
6. Scheduler decides which process is to run next.
7. C procedure returns to the assembly code.
8. Assembly language procedure starts up new current process.

Classes of Interrupts

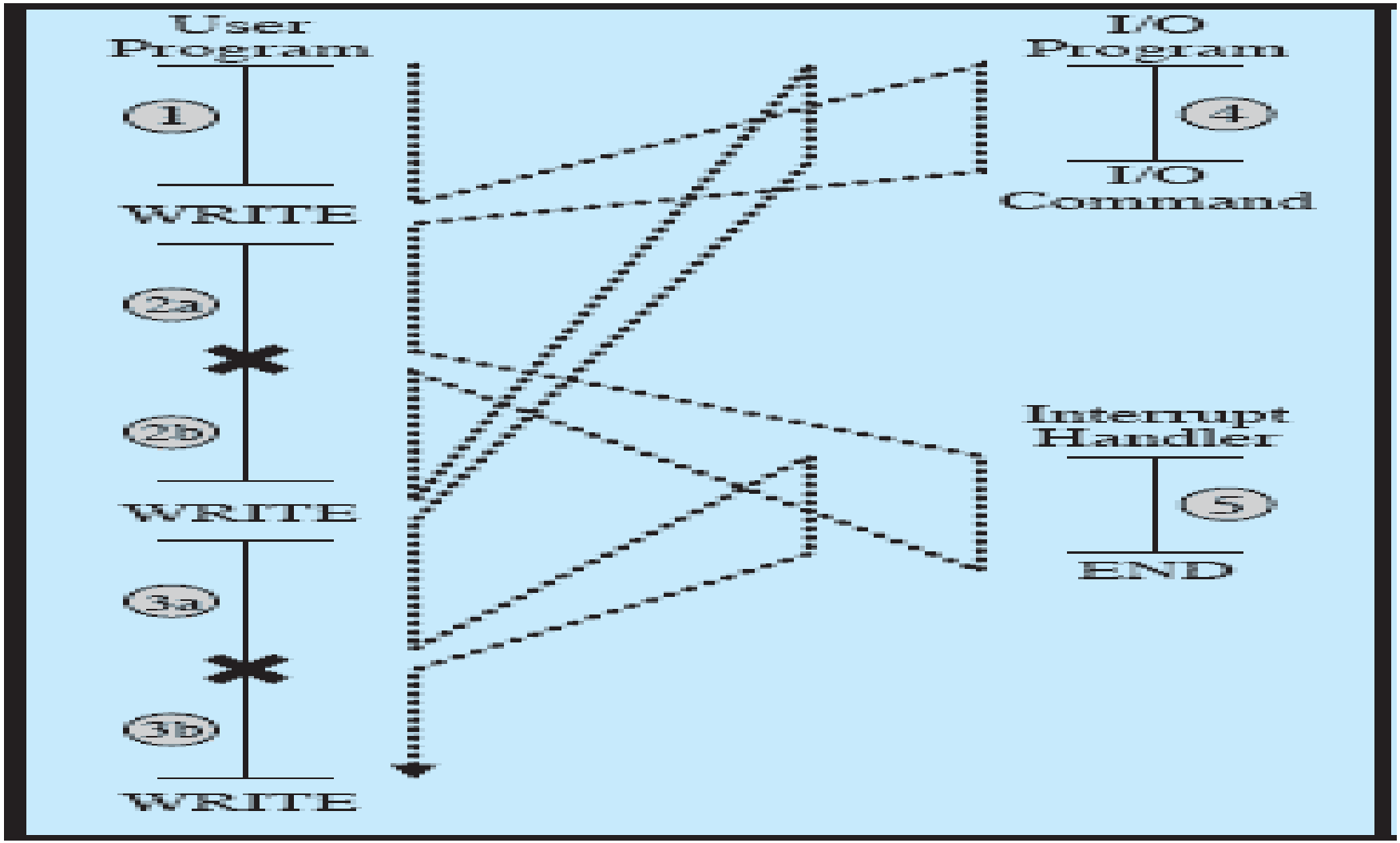
Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows the operating system to perform certain functions on a regular basis.
I/O	Generated by an I/O controller, to signal normal completion of an operation or to signal a variety of error conditions.
Hardware failure	Generated by a failure, such as power failure or memory parity error.

Program Flow of Control



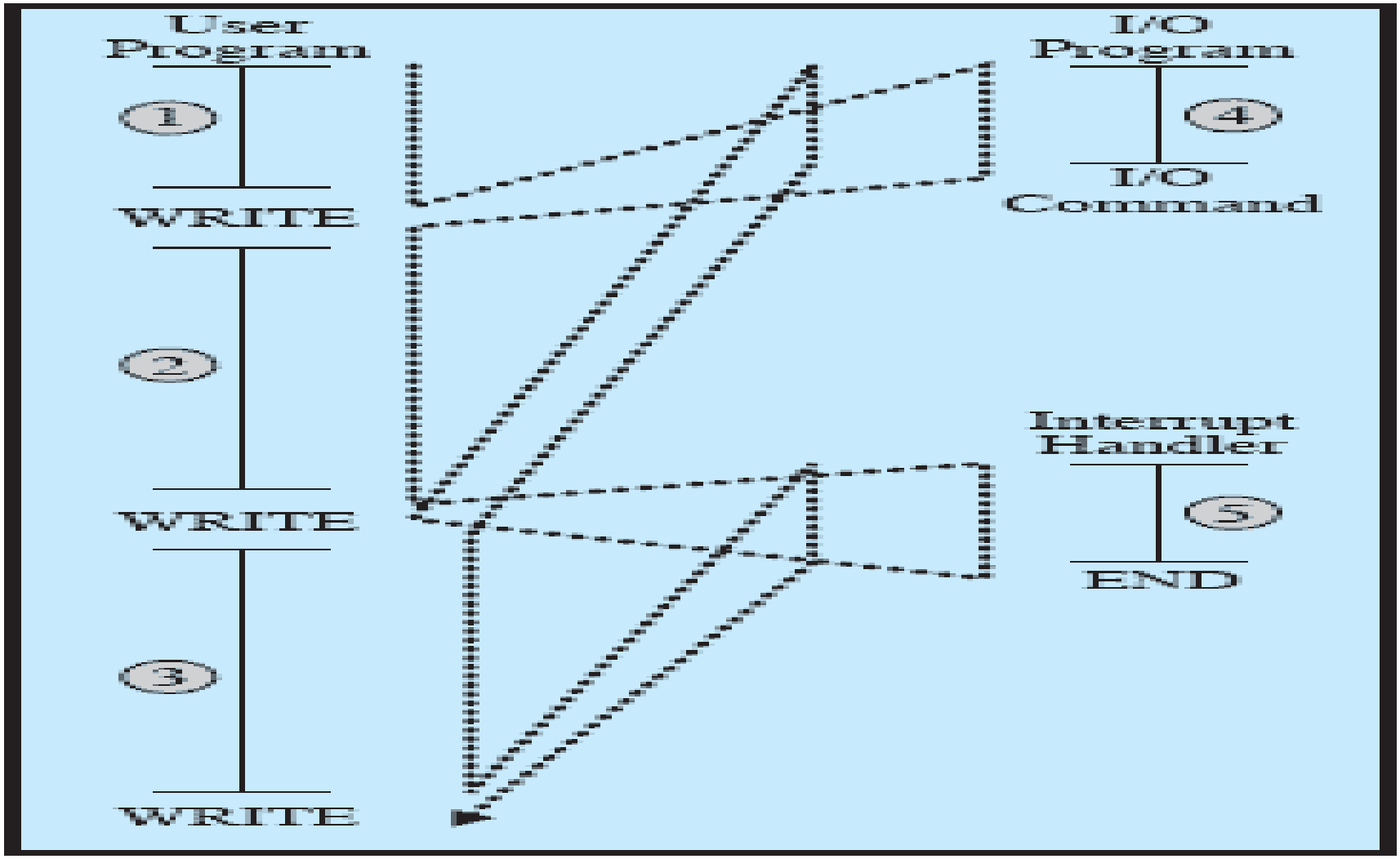
(a) No interrupts

Program Flow of Control



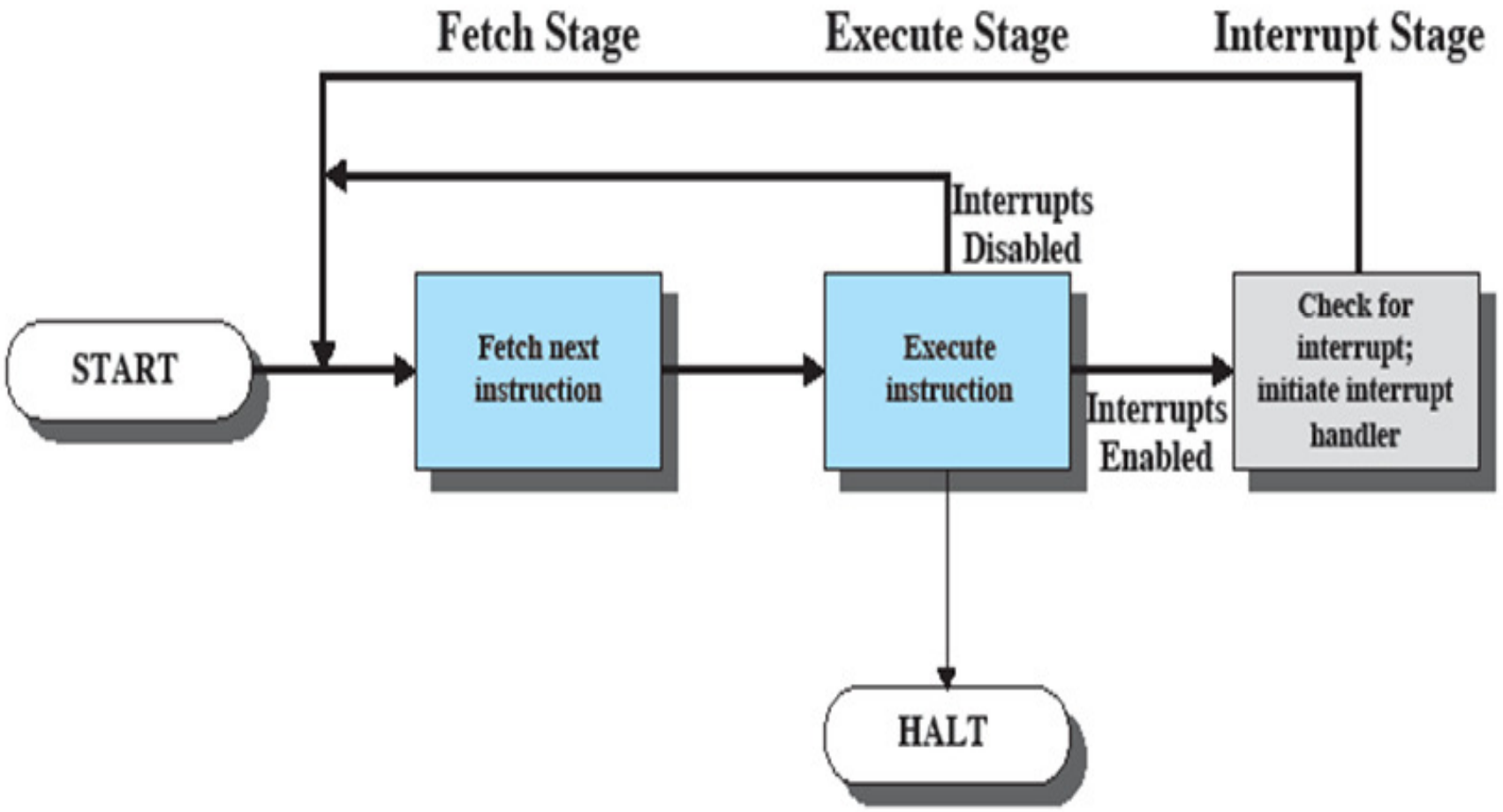
(b) Interrupts; short I/O wait

Program Flow of Control



(c) Interrupts; long I/O wait

Instruction Cycle with Interrupt



Multiprogramming

- Processor has more than one program to execute
- The sequence in which programs are executed depend on their relative priority and whether they are waiting for I/O
- After an interrupt handler completes, control may not return to the program that was executing at the time of the interrupt

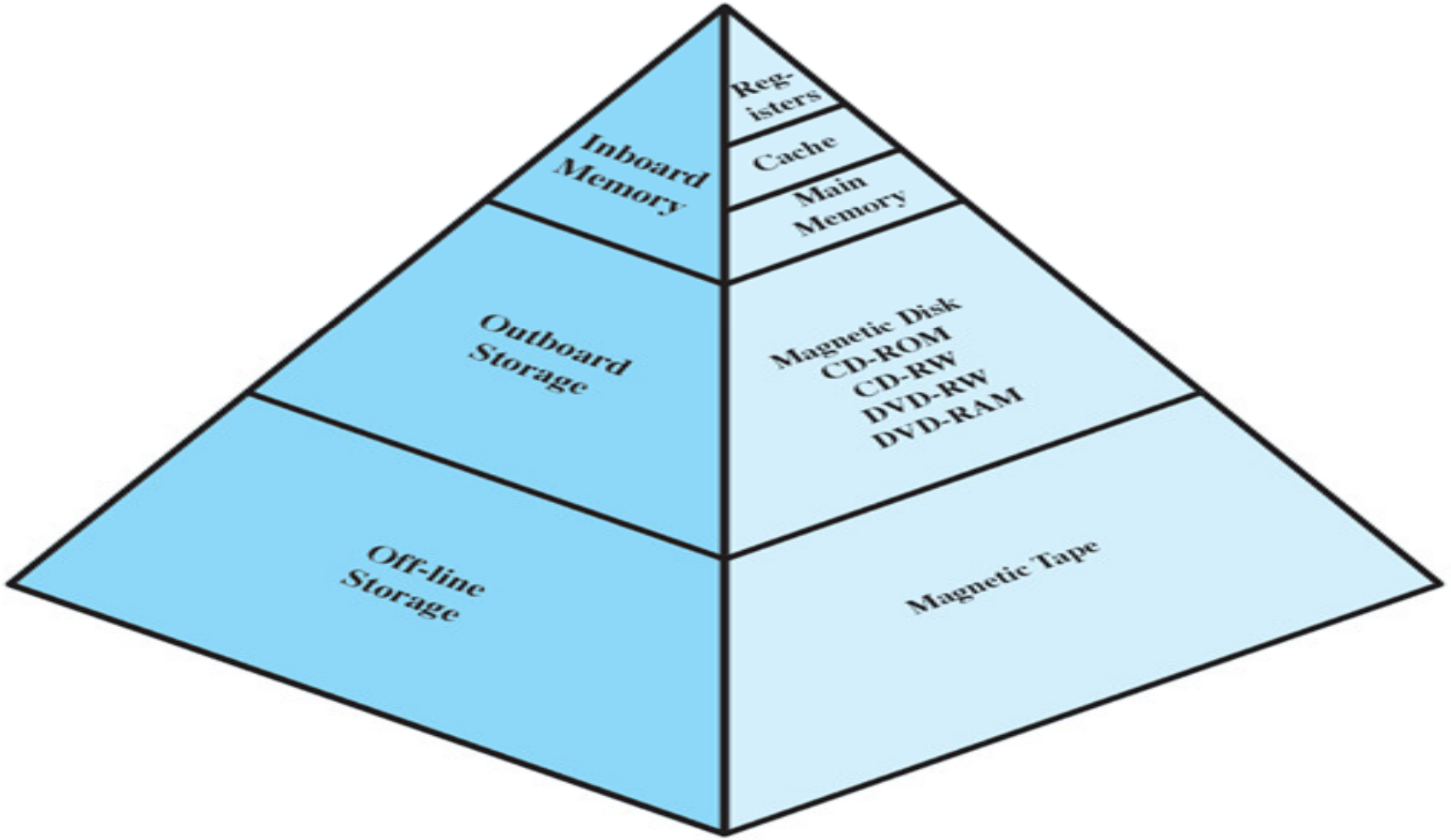
Memory Hierarchy

- Design constraints of memory are: How much?, How fast?, How expensive?.
- Memory parameters: Capacity, Speed (access time and cycle time), Latency, Bandwidth.
- Faster access time, greater cost per bit
- Greater capacity, smaller cost per bit
- Greater capacity, slower access speed

Classification of Memory

- It is based on technology, access method, function and usage mode.
- Access: Random access, sequential access, semi random access
- Capability: R/W memory, Read only memory.
- Technology : Core memory, semiconductor memory, magnetic bubble memory.
  - Semiconductor: RAM, ROM
    - RAM : SRAM, DRAM
    - ROM : ROM, PROM, EPROM, EEPROM, Flash memory
- Role: Main memory, secondary memory, cache memory, virtual memory.

Memory Hierarchy



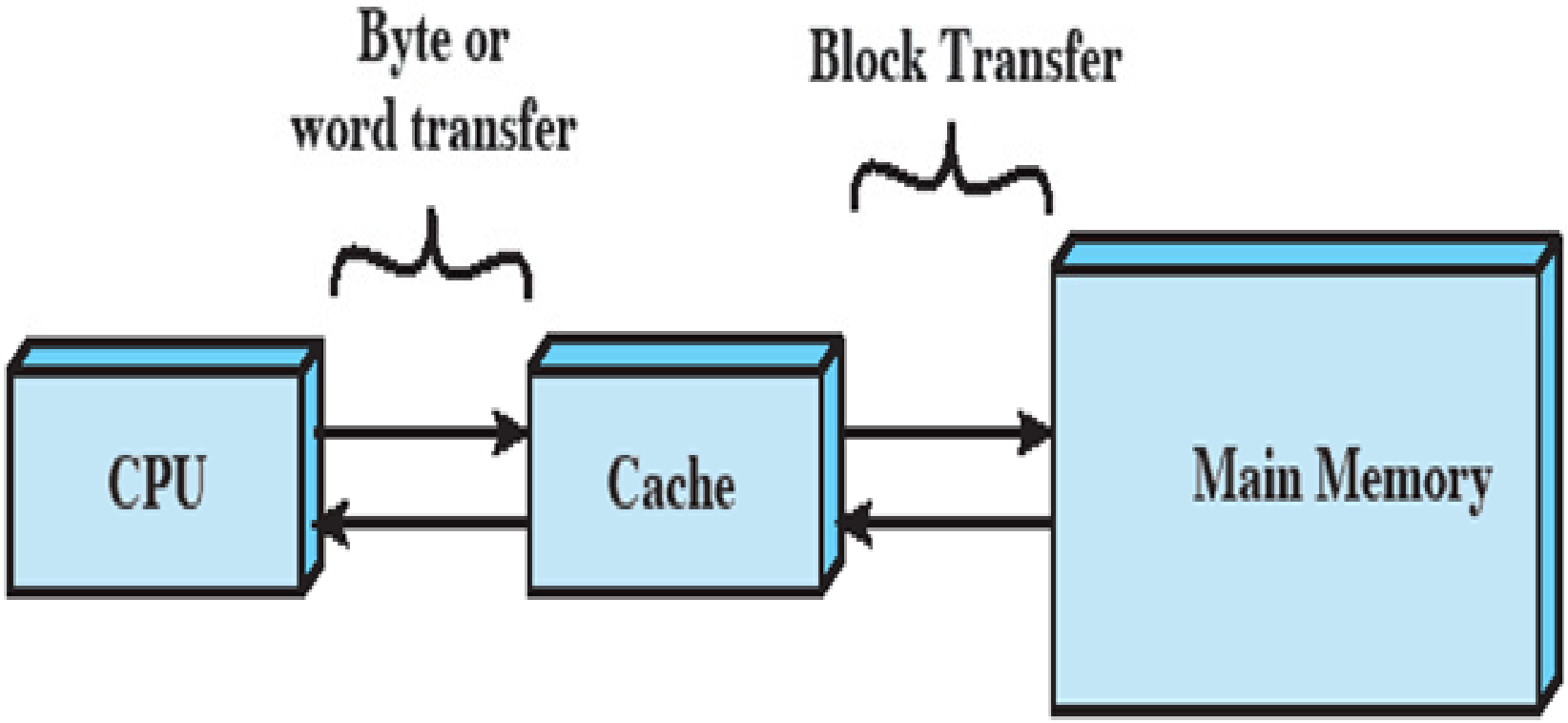
Going Down the Hierarchy

- Decreasing cost per bit
- Increasing capacity
- Increasing access time
- Decreasing frequency of access to the memory by the processor

Memory

- Secondary Memory
  - Auxiliary memory
  - External
  - Nonvolatile
  - Used to store program and data files
- Cache Memory
  - Processor speed faster than memory access speed
  - Exploit the principle of locality with a small fast memory

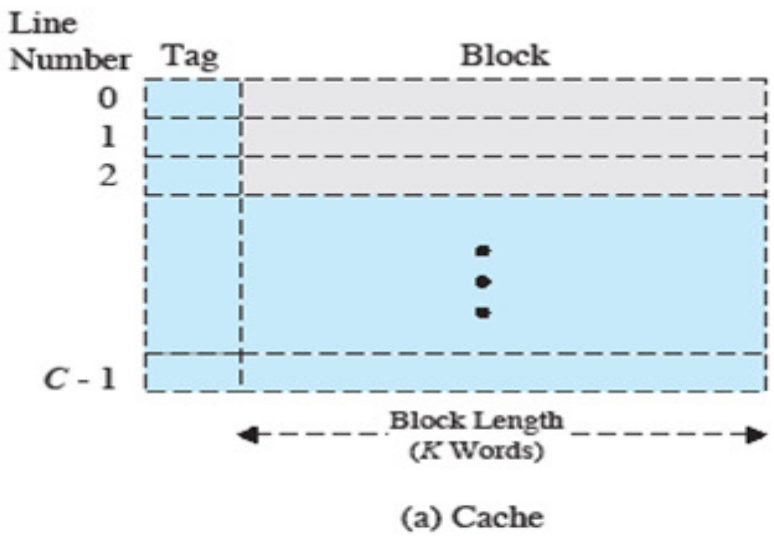
Cache and Main Memory



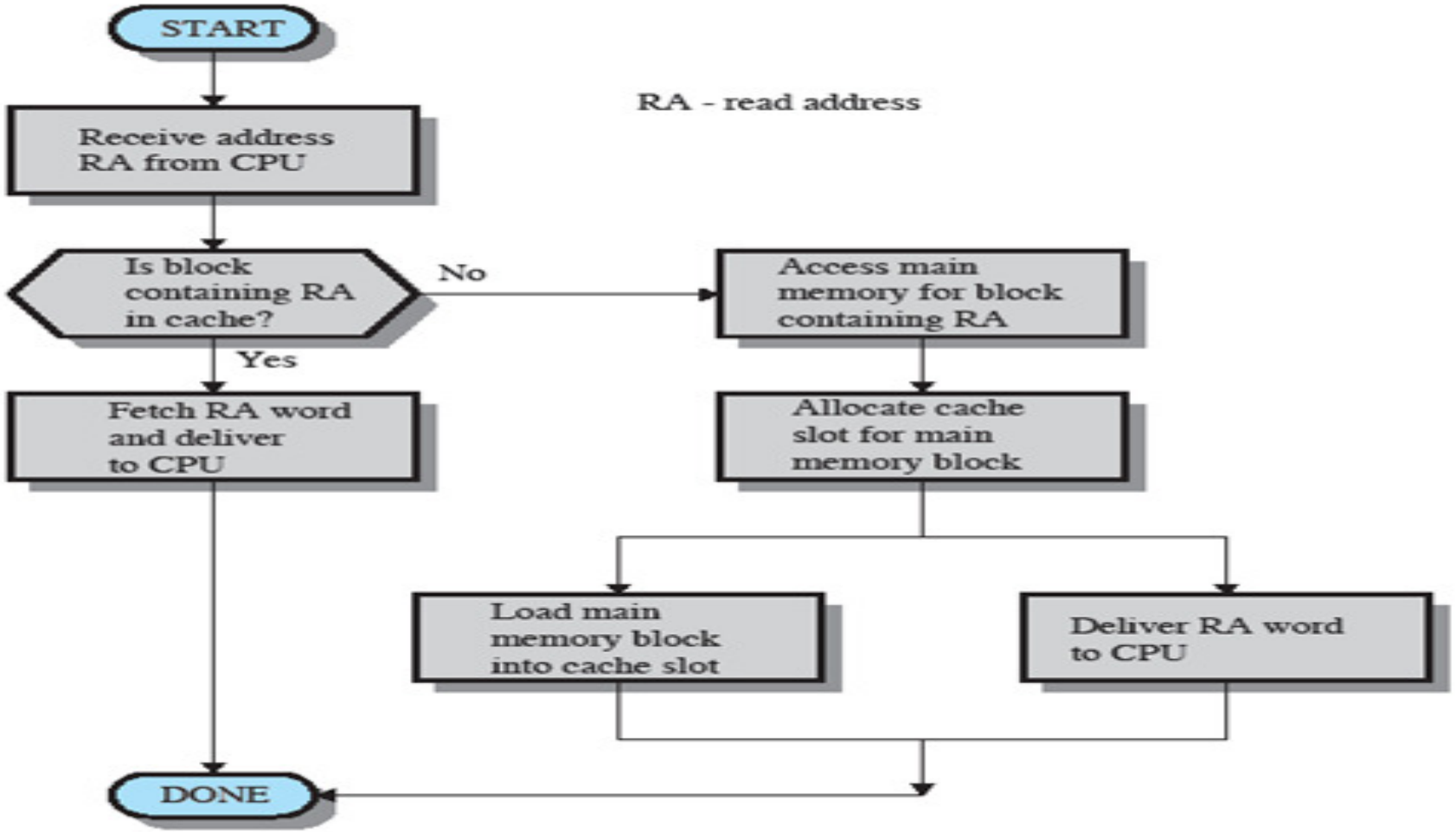
Cache Principles

- Contains copy of a portion of main memory
- Processor first checks cache
- If desired data item not found, relevant block of memory read into cache
- Because of locality of reference, it is likely that future memory references are in that block

Cache/Main-Memory Structure

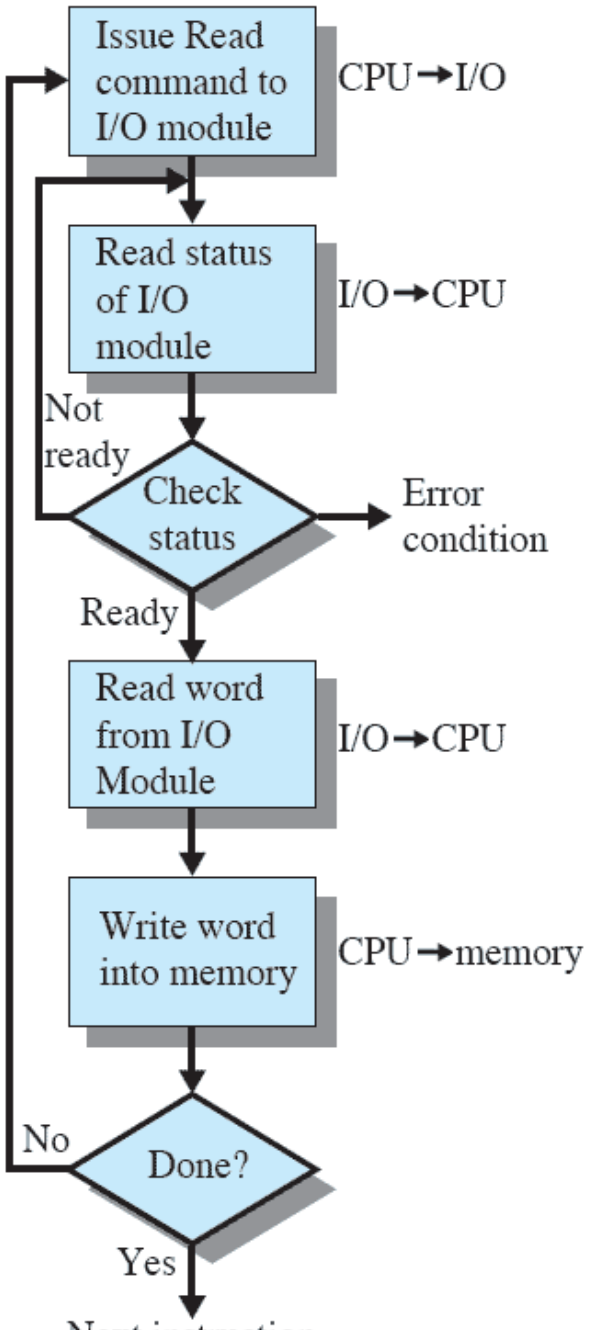


Cache Read Operation



I/O Communication Techniques [Programmed I/O]

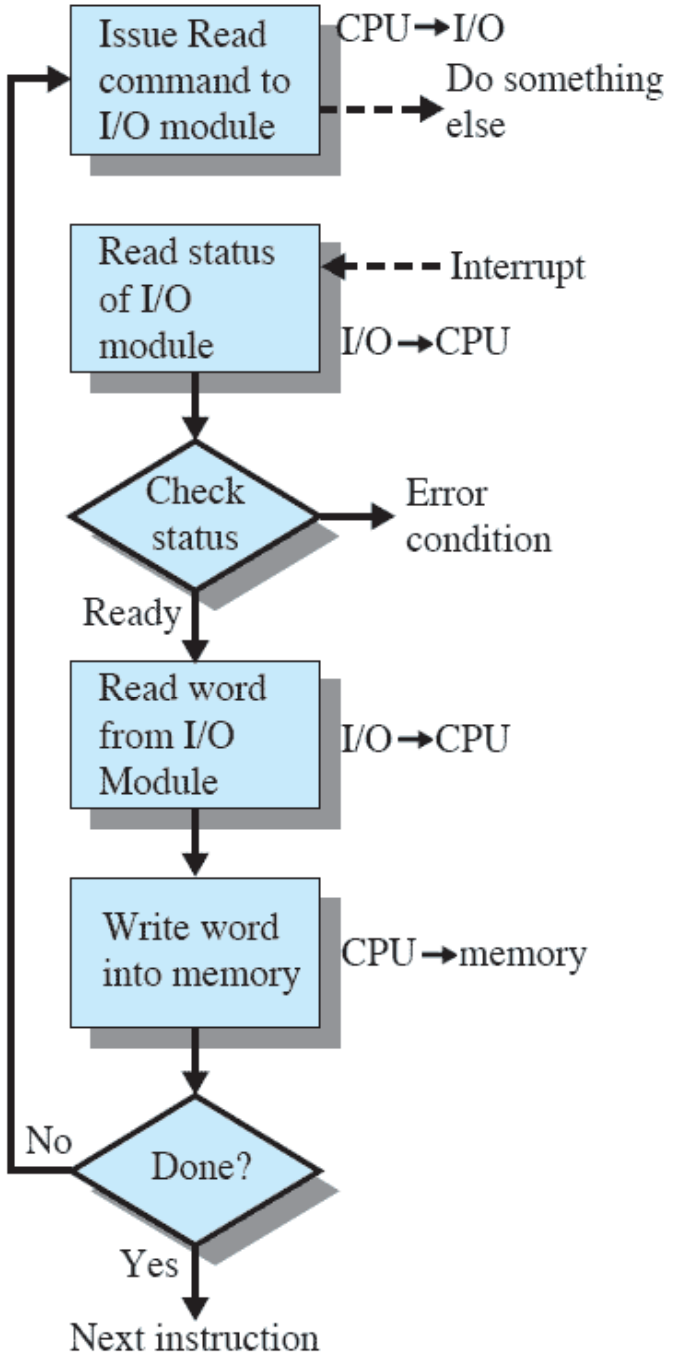
- I/O module performs the action, not the processor
- Sets the appropriate bits in the I/O status register
- No interrupts occur
- Processor checks status until operation is complete



(a) Programmed I/O

I/O Communication Technique [Interrupt-Driven I/O]

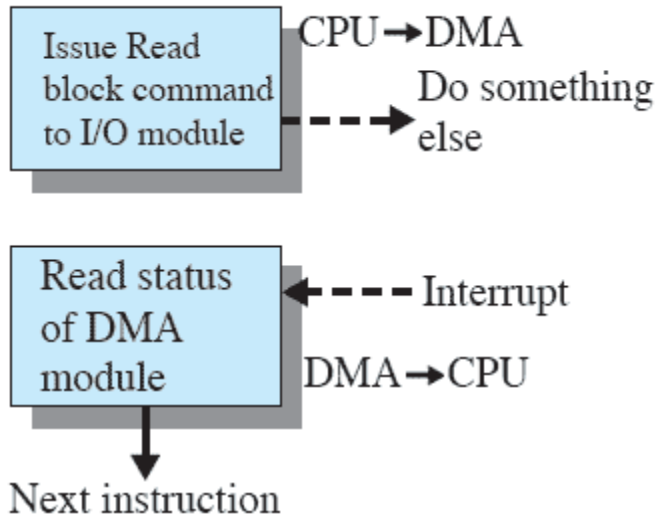
- Processor is interrupted when I/O module ready to exchange data
- Processor saves context of program executing and begins executing interrupt-handler



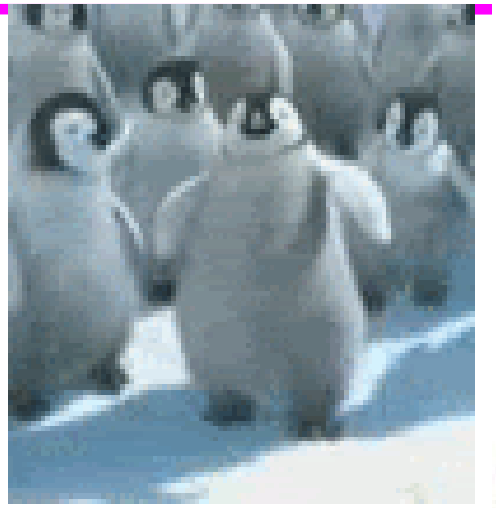
(b) Interrupt-driven I/O

I/O Communication Techniques [Direct Memory Access]

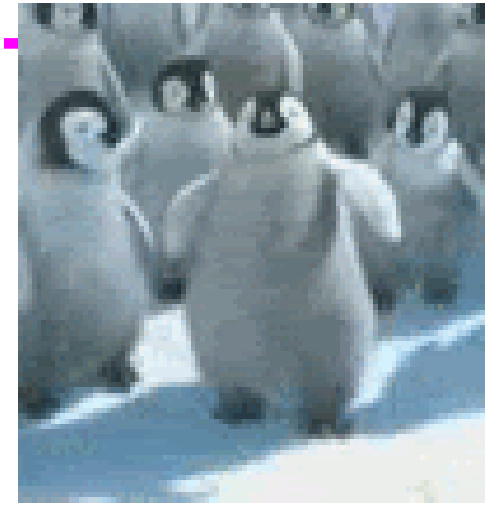
- Transfers a block of data directly to or from memory
- An interrupt is sent when the transfer is complete
- More efficient

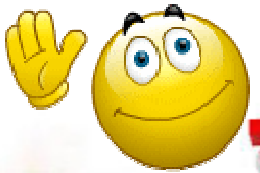


(c) Direct memory access




THANKS






THANKS


THANKS


THANKS