Computer System Overview

Chapter 1

Operating Systems - Making computing power available to users by controlling the hardware

Basic Elements

- Processor
- Main Memory
  - referred to as real memory or primary memory
  - volatile
- I/O modules
  - secondary memory devices
  - communications equipment
  - terminals
- System interconnection
  - communication among processors, memory, and I/O modules

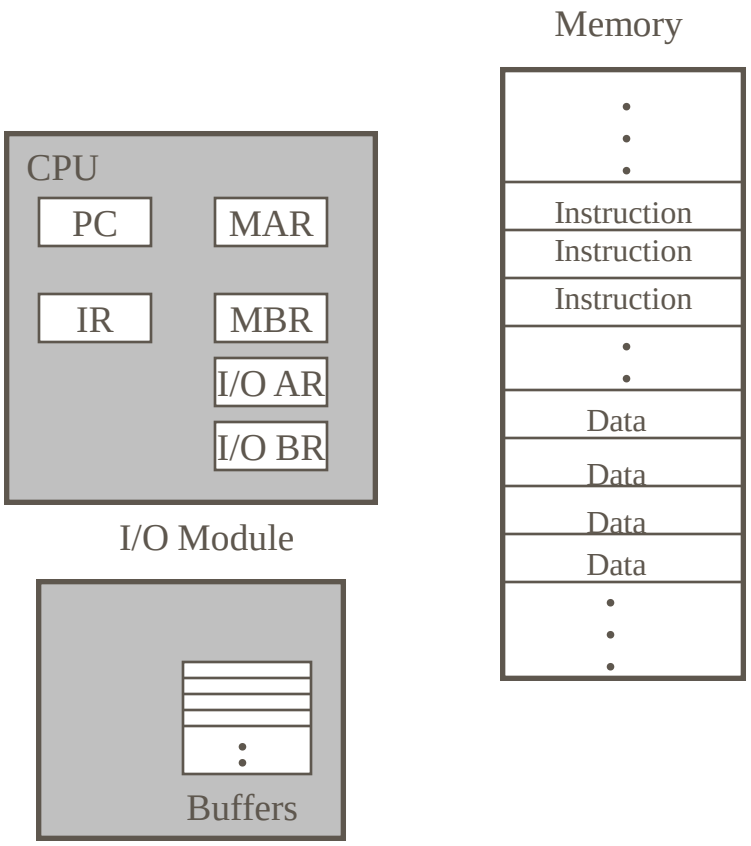
Registers

- ✓ Memory that is faster and smaller than main memory
- ✓ Temporarily stores data during processing

Top-level Components (Registers)

- ✓ IR - Instruction Register
  - most recently fetched instruction
- ✓ PC - Program counter
  - address for next instruction
- ✓ MAR - Memory Address Register
  - address for next read or write by CPU
- ✓ MBR - Memory Buffer Register
  - CPU puts data to be written into memory
  - CPU receives data read from memory
- ✓ I/OAR - I/O Address
  - specifies a particular I/O device
- ✓ I/OBR - I/O Buffer
  - exchange of data between an I/O module and the processor

Computer Components: top-level view



Processor Registers

- User-visible registers
  - May be referenced by machine language
  - Available to all programs - application programs and system programs
  - Types of registers
    - Data
    - Address
    - Condition Code

User-Visible Registers

- ✓ Data Registers
  - can be assigned by the programmer
- ✓ Address Registers
  - contain main memory address of data and instructions
  - may contain a portion of an address that is used to calculate the complete address
    - index register
    - segment pointer
    - stack pointer

User-Visible Registers

- Address Registers
  - Index
    - involves adding an index to a base value to get an address
  - Segment pointer
    - when memory is divided into segments, memory is referenced by a segment and an offset
  - Stack pointer
    - points to top of stack

User-Visible Registers

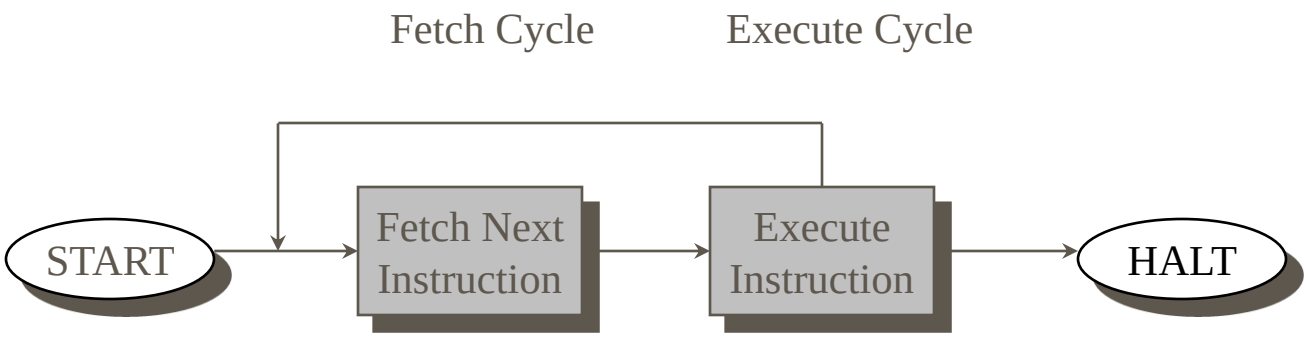
- ✓ Condition Codes or Flags
  - Bits set by the processor hardware as a result of operations
  - Can be accessed by a program but not changed
  - Examples
    - positive result
    - negative result
    - zero
    - overflow

Control and Status Registers

- ✓ 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)
  - condition codes
  - Interrupt enable/disable
  - Supervisor (a.k.a monitor) mode flag

Instruction Execution

- ✓ Processor executes instructions in a program
- ✓ Instructions are fetched from memory one at a time



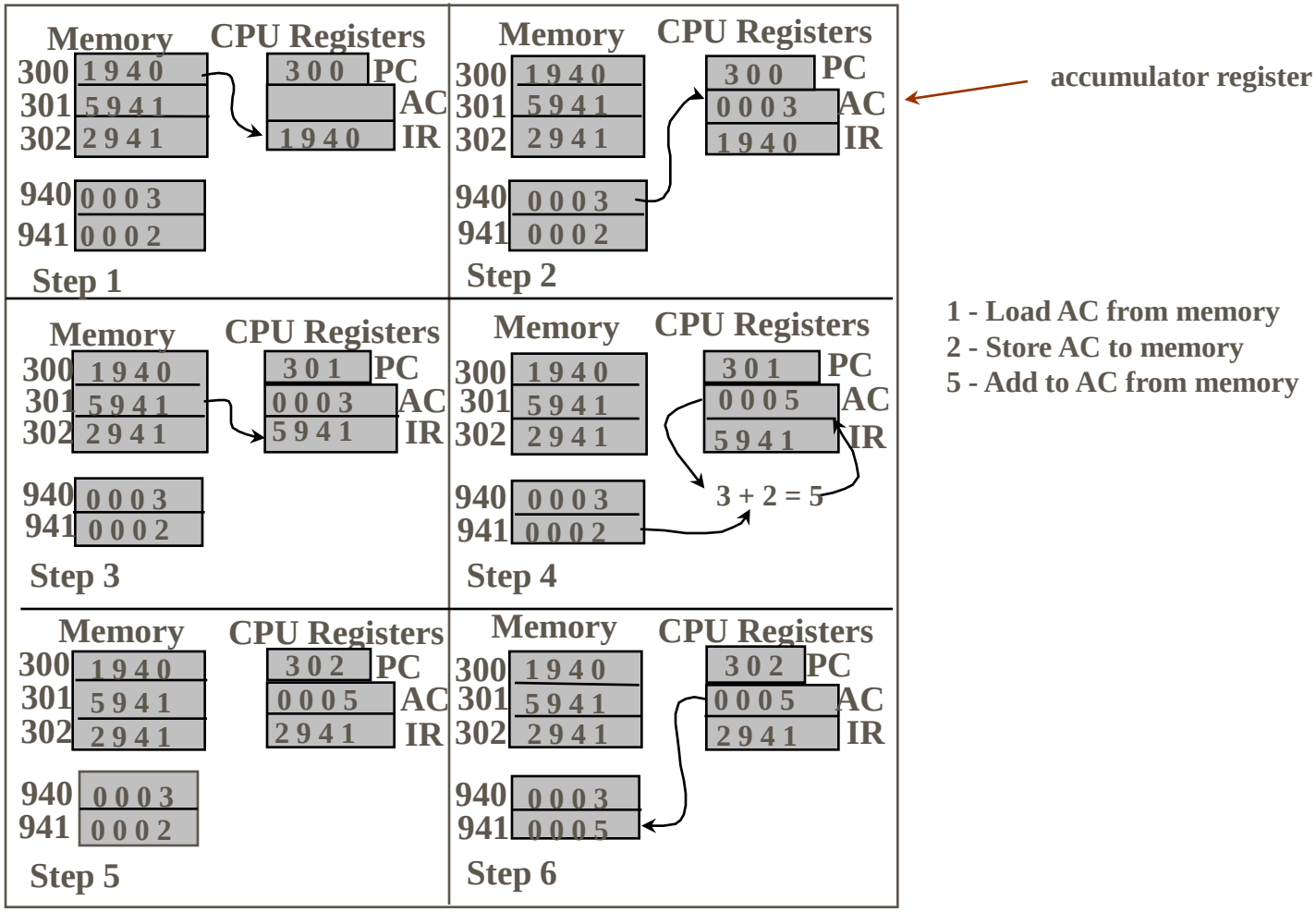
Instruction Fetch and Execute

- The processor fetches the instruction from memory
- Program counter (PC) holds address of the instruction to be fetched next
- Program counter is incremented after each fetch

Instruction Register

- ✓ Fetched instruction is placed here
- ✓ Types of instructions
  - Processor-memory
    - transfer data between processor and memory
  - Processor-I/O
    - data transferred to or from a peripheral device
  - Data processing
    - arithmetic or logic operation on data
  - Control
    - alter sequence of execution

Example of Program Execution



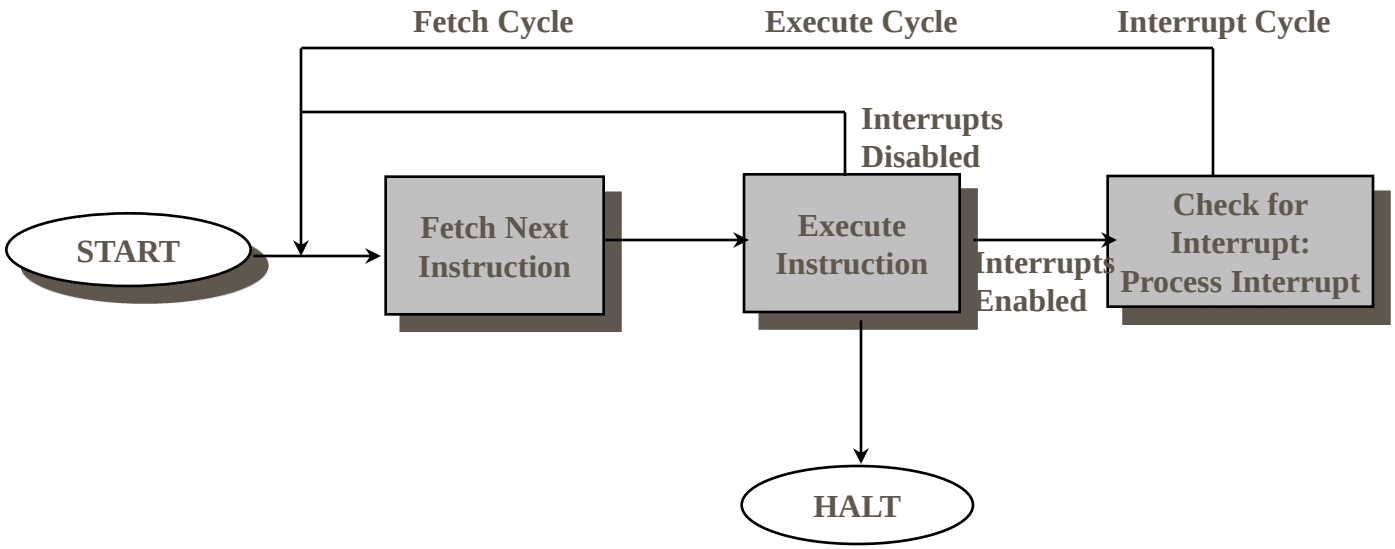
Interrupts

- An interruption of the normal processing of processor
- Improves processing efficiency
- Allows the processor to execute other instructions while an I/O operation is in progress
- A suspension of a process caused by an event external to that process and performed in such a way that the process can be resumed

Classes of Interrupts

- ✓ Program
  - arithmetic overflow
  - division by zero
  - execute illegal instruction
  - reference outside user's memory space
  - request for an OS service
- ✓ Timer
- ✓ I/O
- ✓ Hardware failure

Instruction Cycle with Interrupts



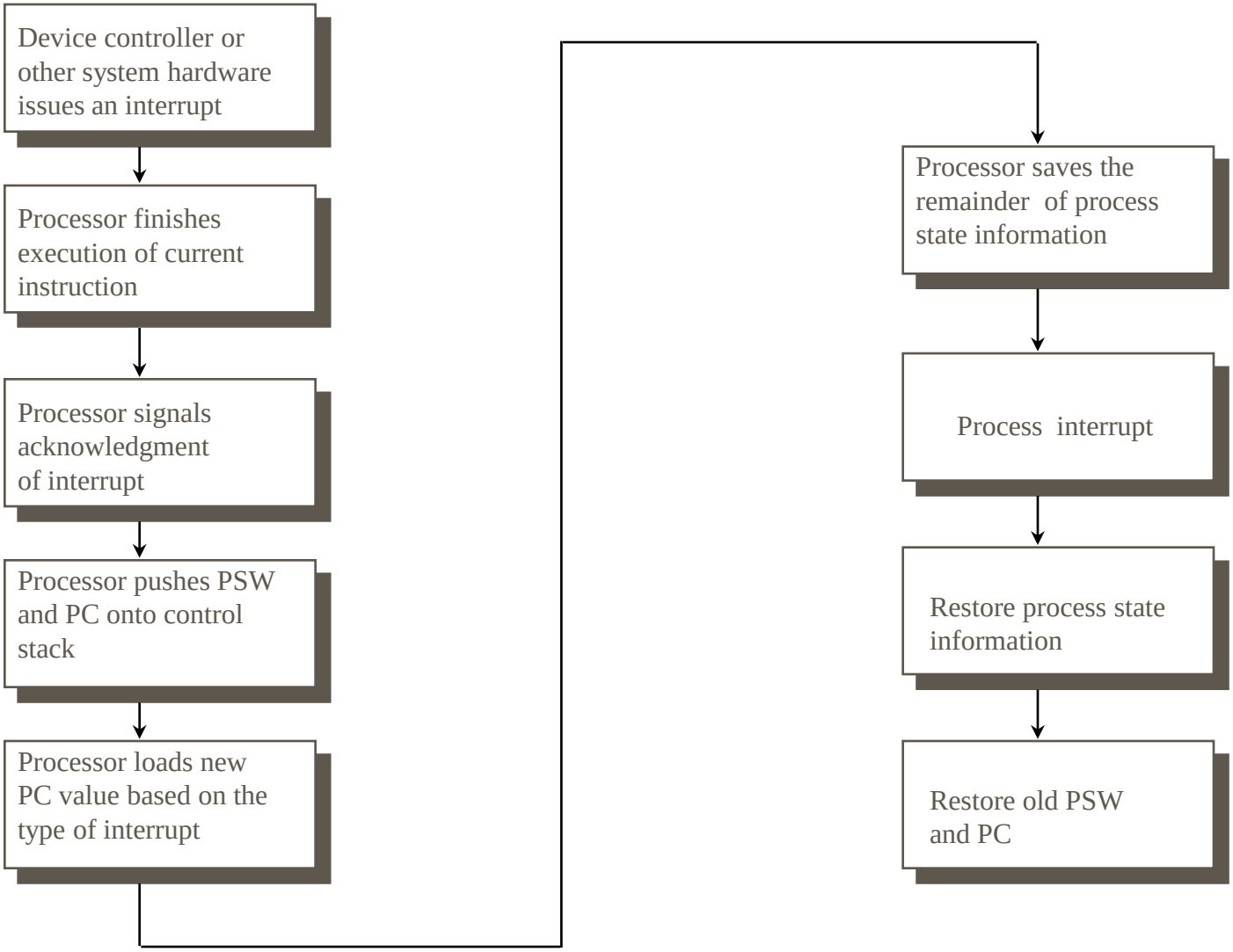
Interrupt Handler

- A program that determines nature of the interrupt and performs whatever actions are needed
- Control is transferred to this program
- Generally part of the operating system

Interrupt Cycle

- ✓ Processor checks for interrupts
- ✓ If no interrupts, fetch the next instruction for the current program
- ✓ If an interrupt is pending, suspend execution of the current program, and execute the interrupt handler

Simple Interrupt Processing



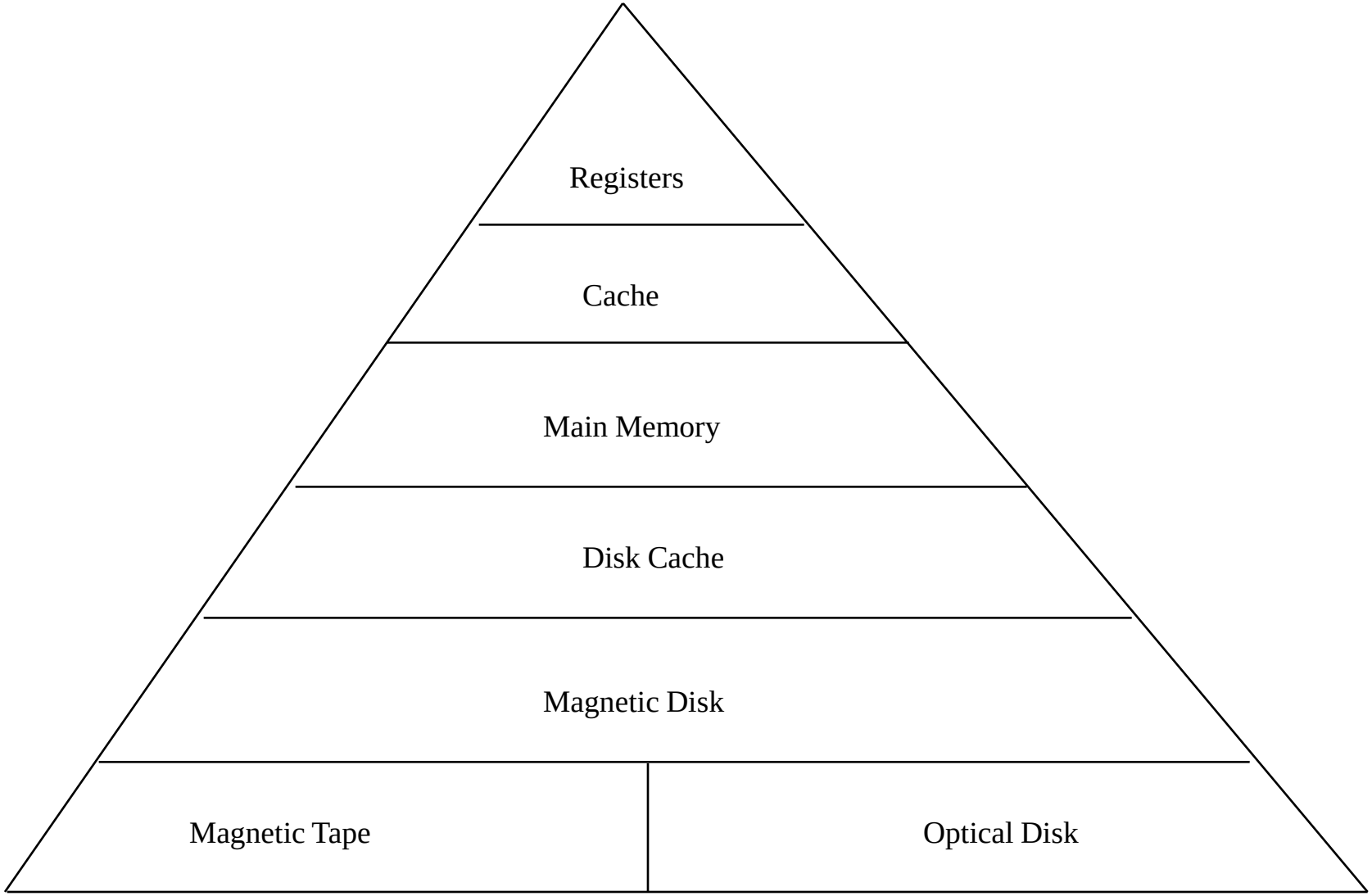
Multiple Interrupts Sequential Order

- ✓ Disable interrupts so processor can complete task
- ✓ Interrupts remain pending until the processor enables interrupts
- ✓ After interrupt handler routine completes, the processor checks for additional interrupts

Multiple Interrupts Priorities

- ✓ Higher priority interrupts cause lower-priority interrupts to wait
- ✓ Causes a lower-priority interrupt handler to be interrupted
- ✓ Example when input arrives from communication line, it needs to be absorbed quickly to make room for more input

Memory Hierarchy



Going Down the Hierarchy

- ✓ Decreasing cost per bit
- ✓ Increasing capacity
- ✓ Increasing access time
- ✓ Decreasing frequency of access of the memory by the processor
  - based on the principle of *locality of reference*

Disk Cache

- A portion of main memory used as a buffer to temporarily hold data for the disk
- Disk writes are clustered
- Some data written out may be referenced again. The data are retrieved *rapidly* from the software cache instead of *slowly*, from disk

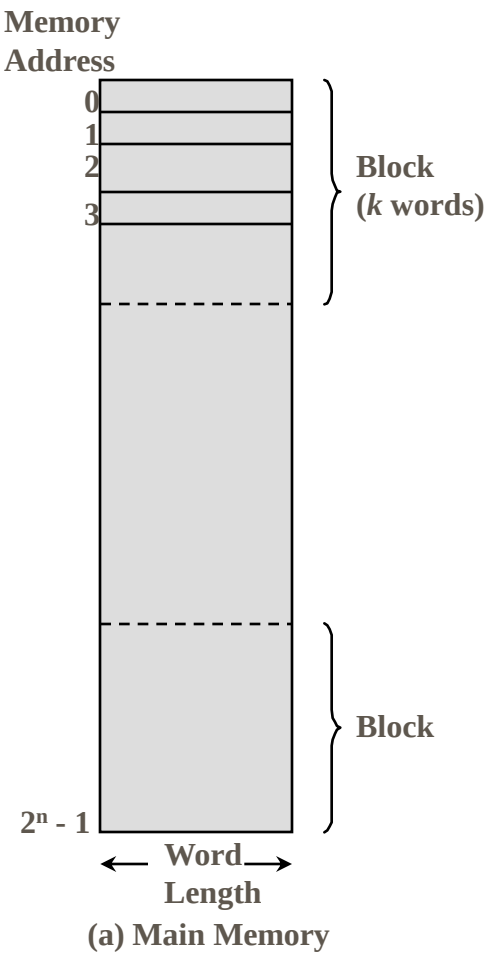
Cache Memory

- ✓ Invisible to operating system
- ✓ Used similarly to virtual memory
- ✓ Increases the speed of memory
- ✓ Processor speed is faster than memory speed

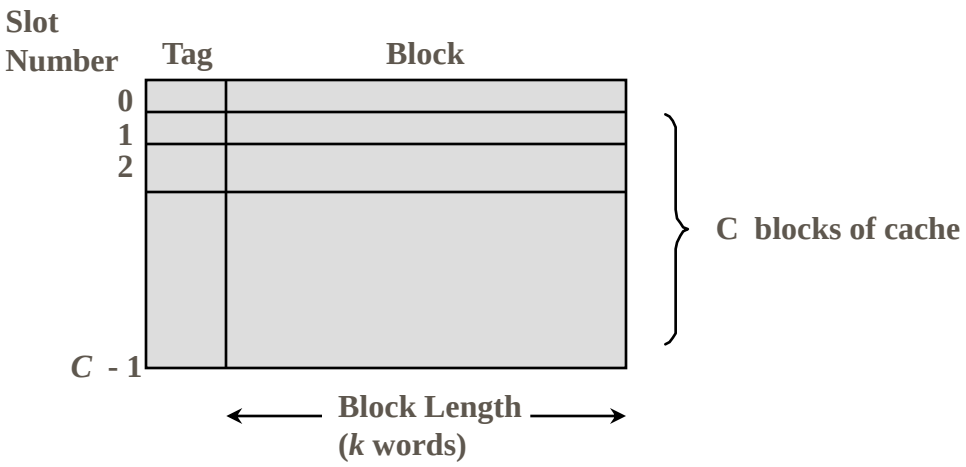
Cache Memory

- Contains an image of a portion of main memory
- Processor first checks cache
- If not found in cache, the block of memory containing the needed information is moved to the cache

Cache/Main-Memory Structure



(a) Main Memory



(b) Cache

Tag: address of the block + control bits

Cache Design

- Cache size
  - Even small caches have significant impact on performance
- Block size
  - the unit of data exchanged between cache and main memory
  - hit means the information was found in the cache
  - larger block size more hits until probability of using newly fetched data becomes less than the probability of reusing data that has been moved out of cache

Cache Design

- ✓ Mapping function
  - determines which cache location the block will occupy
- ✓ Replacement algorithm
  - determines which block to replace
  - Least-Recently-Used (LRU) algorithm

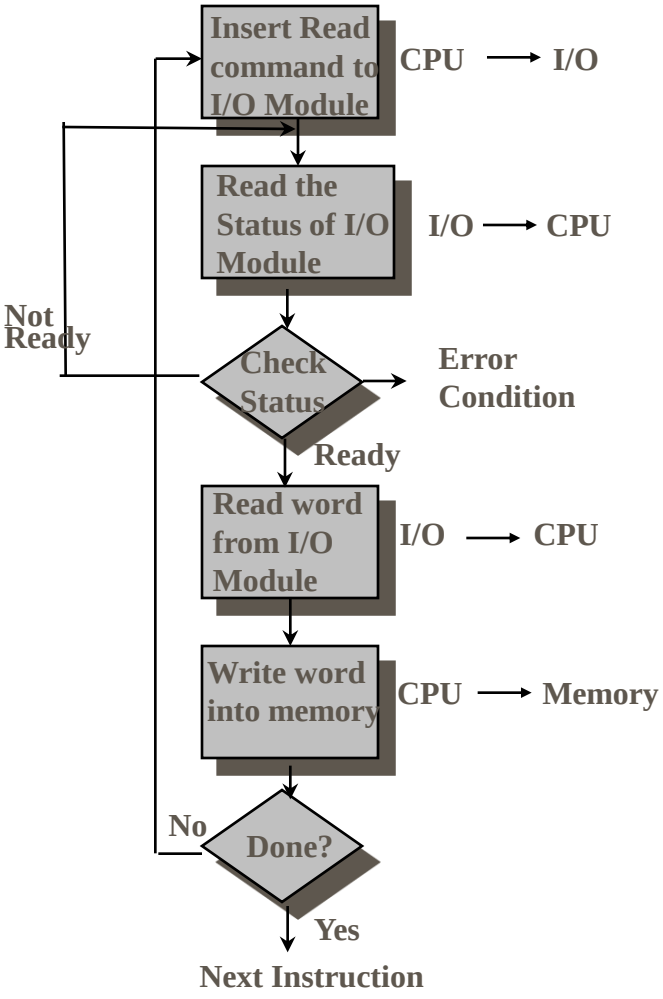
Cache Design

- Write policy
  - write a block of cache back to main memory
  - main memory must be current for direct memory access by I/O modules and multiple processors

Programmed I/O

- I/O module performs the action, not the processor
- Sets appropriate bits in the I/O status register
- No interrupts occur
- Processor is kept busy checking status

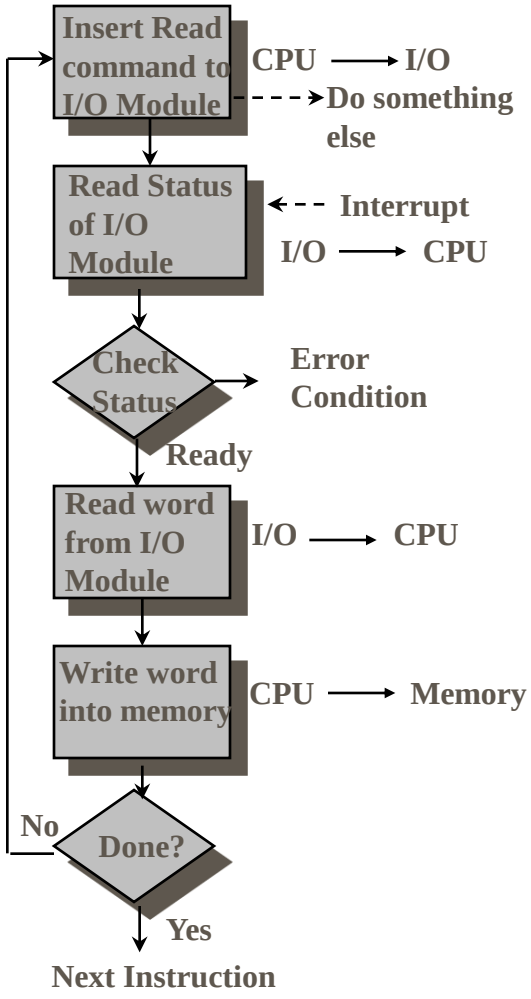
Programmed I/O



Interrupt-Driven I/O

- Processor is interrupted when I/O module ready to exchange data
- Processor is free to do other work
- No needless waiting
- Consumes a lot of processor time because every word read or written passes through the processor

Interrupt-Driven I/O



Direct Memory Access (DMA)

- I/O transfers a block of data directly to or from memory
- An interrupt is sent when the task is complete
- The processor is only involved at the beginning and end of the transfer.
  - Is free to do other things in-between

Direct Memory Access