COMPUTER SYSTEM OVERVIEW

OPERATING SYSTEM

- Exploits the hardware resources of one or more processors
- Provides a set of services to system users
- Manages secondary memory and I/O devices

BASIC ELEMENTS

Processor

I/O Modules

Main Memory

System Bus

PROCESSOR

Controls the operation of the computer

Performs the data processing functions

Referred to as the *Central Processing Unit* (CPU)

MAIN MEMORY

- Stores data and programs
- Typically, volatile
  - Contents of the memory is lost when the computer is shut down
- Referred to as *real memory* or *primary memory*

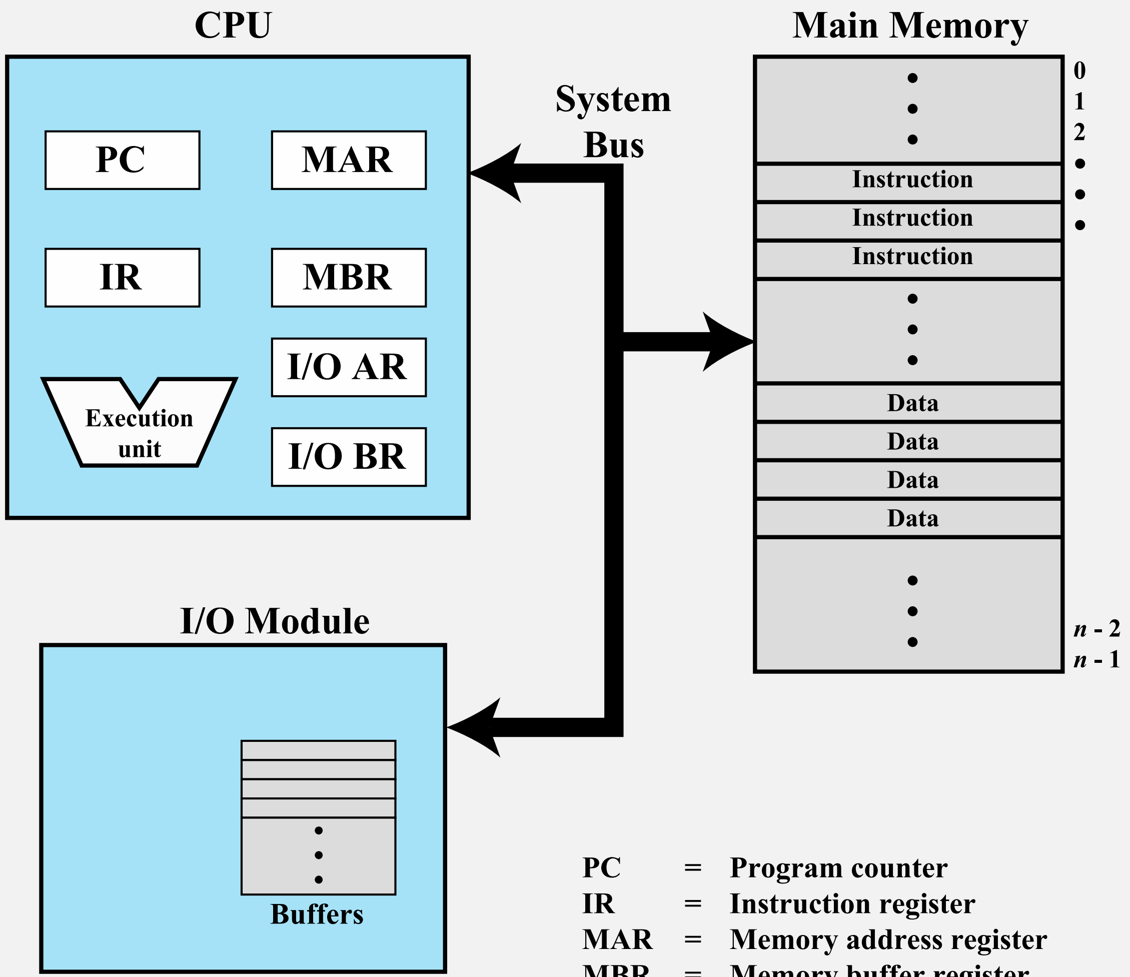
I/O MODULES

Move data between the computer and its external environment

- Secondary memory devices (e.g. disks)
- Communications equipment
- Terminals

SYSTEM BUS

- Provides for communication among processors, main memory, and I/O modules



PC	=	Program counter
IR	=	Instruction register
MAR	=	Memory address register
MBR	=	Memory buffer register
I/O AR	=	Input/output address register
I/O BR	=	Input/output buffer register

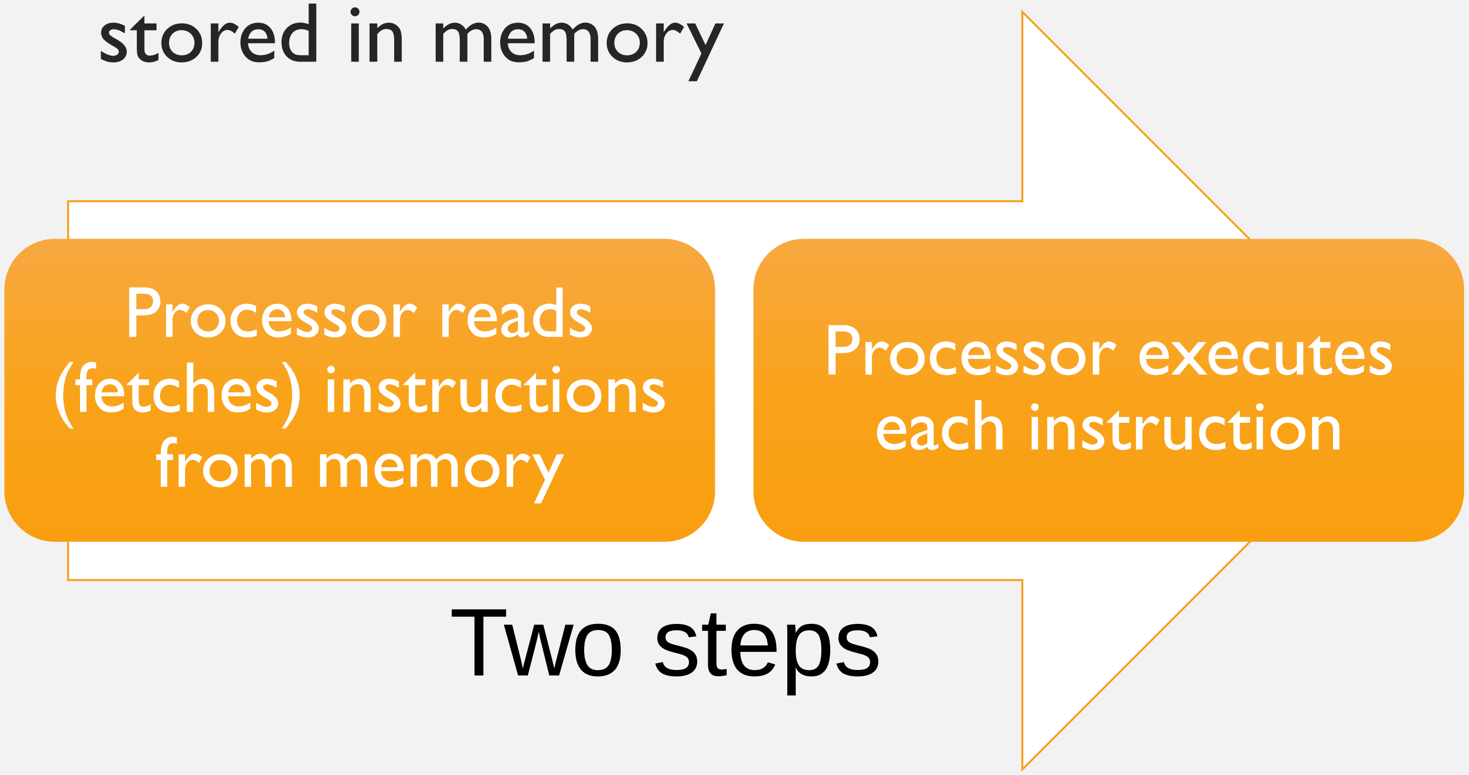
Figure 1.1 Computer Components: Top-Level View

MICROPROCESSOR

- Invention that brought about desktop and handheld computing
- Contains a processor on a single chip
- Fastest general -purpose processors
- Multiprocessors
- Each chip (socket) contains multiple processors (cores)

INSTRUCTION EXECUTION

- A program consists of a set of instructions stored in memory



Processor reads (fetches) instructions from memory

Processor executes each instruction

Two steps

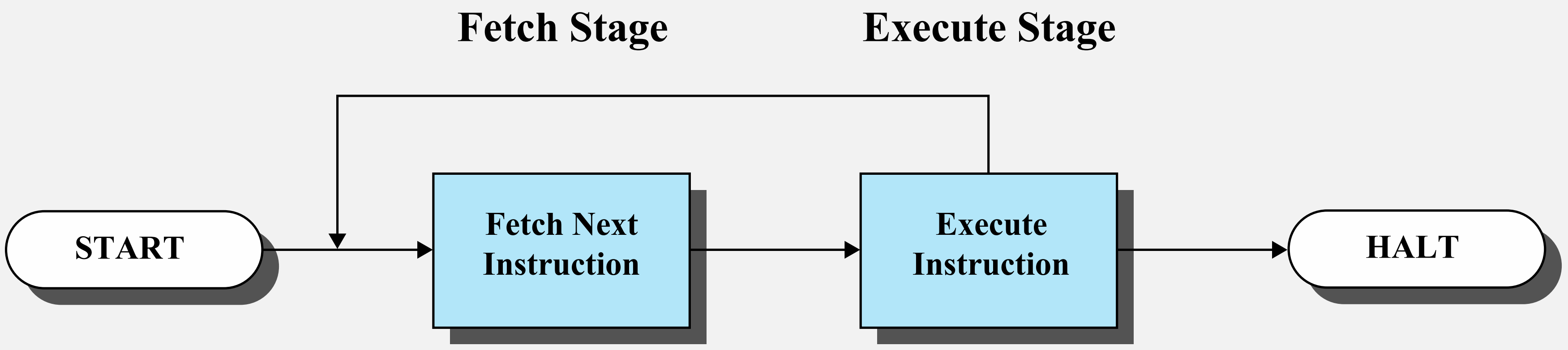


Figure 1.2 Basic Instruction Cycle

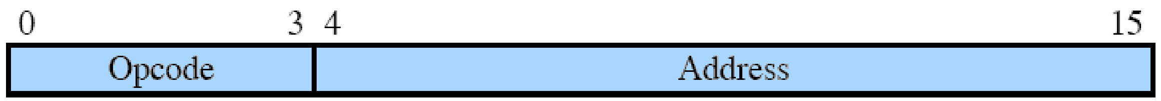
INSTRUCTION FETCH AND EXECUTE

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

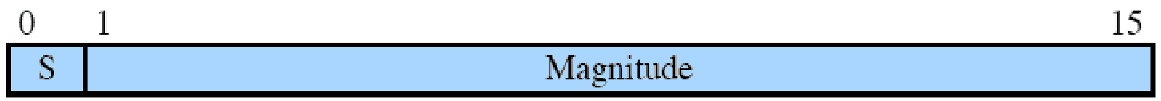
INSTRUCTION REGISTER (IR)

Fetched instruction is loaded into Instruction Register (IR)

- Processor interprets the instruction and performs required action:
  - Processor-memory
  - Processor-I/O
  - Data processing
  - Control



(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

Figure 1.3 Characteristics of a Hypothetical Machine

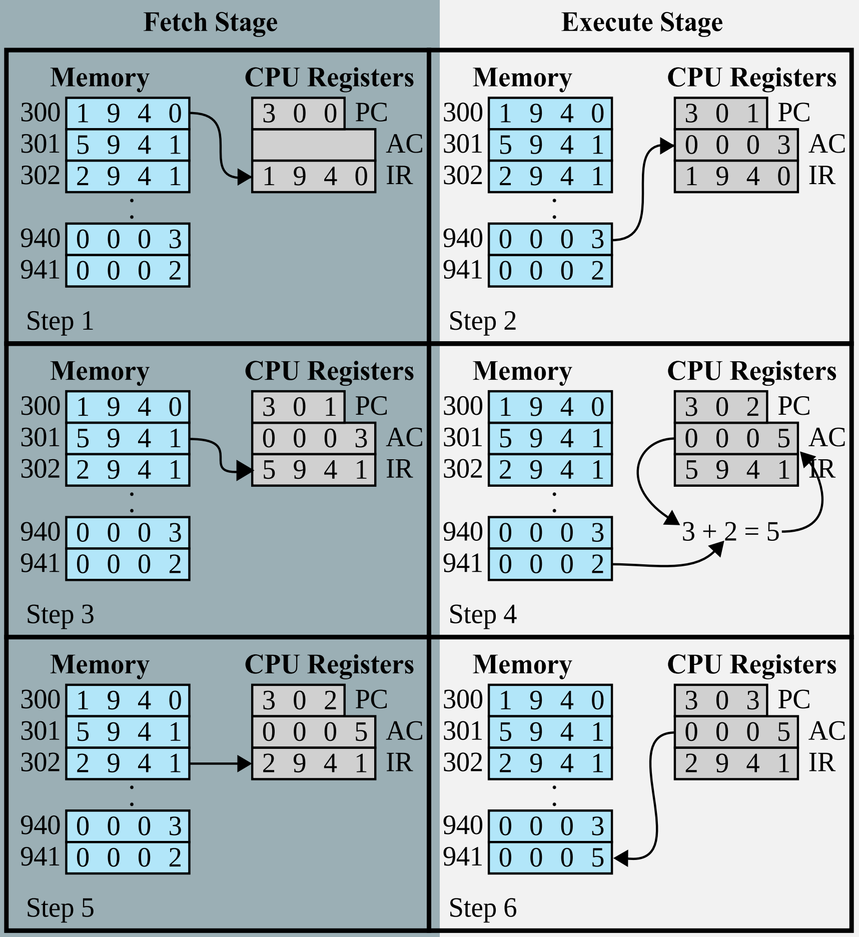


Figure 1.4 Example of Program Execution (contents of memory and registers in hexadecimal)

INTERRUPTS

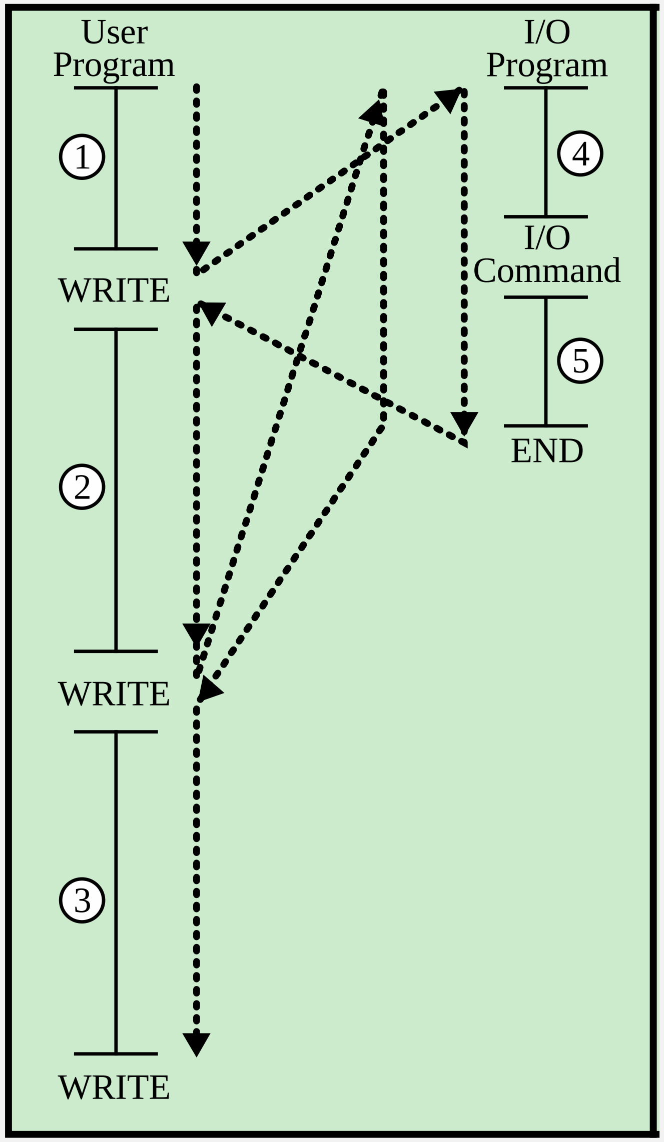
- Mechanism by which other modules may interrupt the normal sequencing of the processor
- Provided to improve processor utilization
  - Most I/O devices are slower than the processor
  - Processor must pause to wait for device
  - Wasteful use of the processor

Table 1.1 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.

FIGURE 1.5A

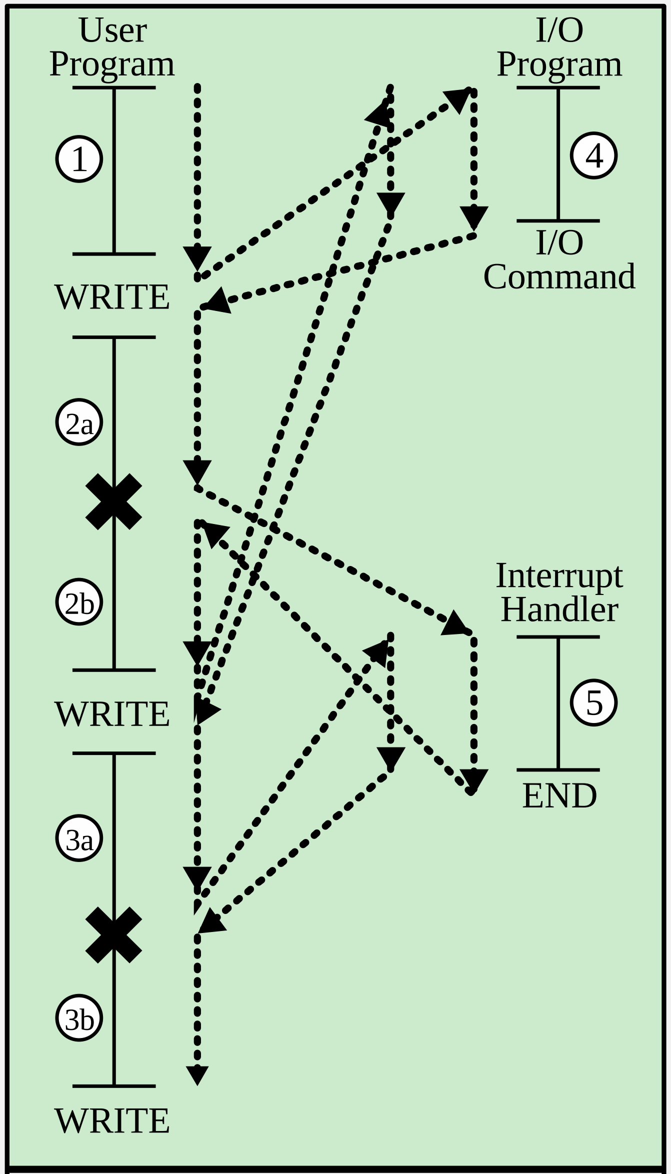
FLOW OF CONTROL WITHOUT INTERRUPTS



(a) No interrupts

Figure 1.5b

Short I/O Wait

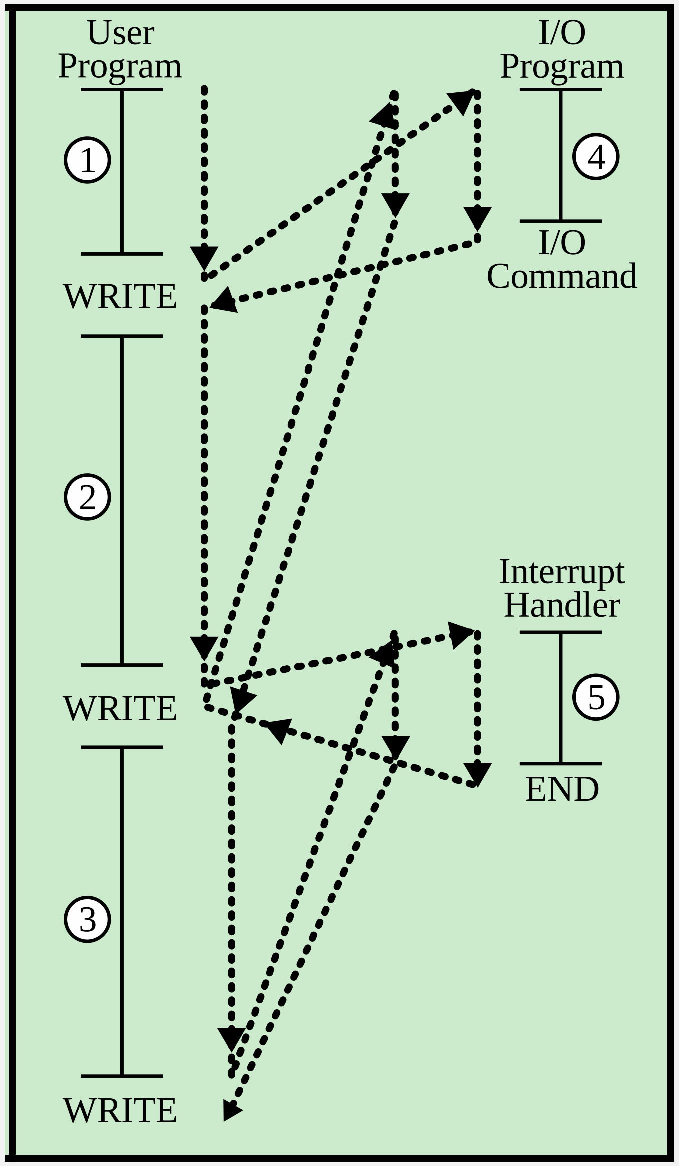


(b) Interrupts; short I/O wait

X = interrupt occurs during course of execution of user program

Figure 1.5c

Long I/O Wait



(c) Interrupts; long I/O wait

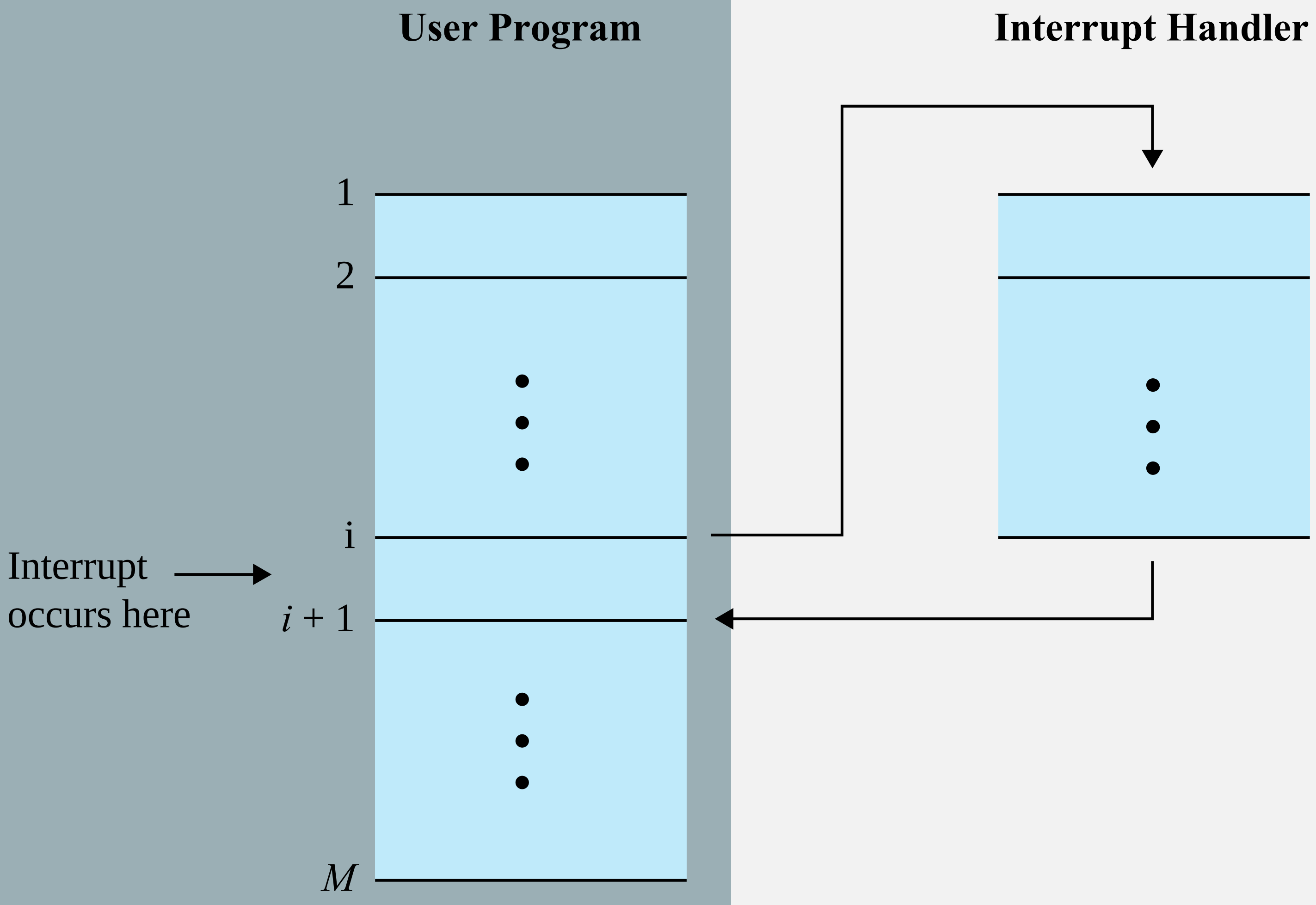


Figure 1.6 Transfer of Control via Interrupts

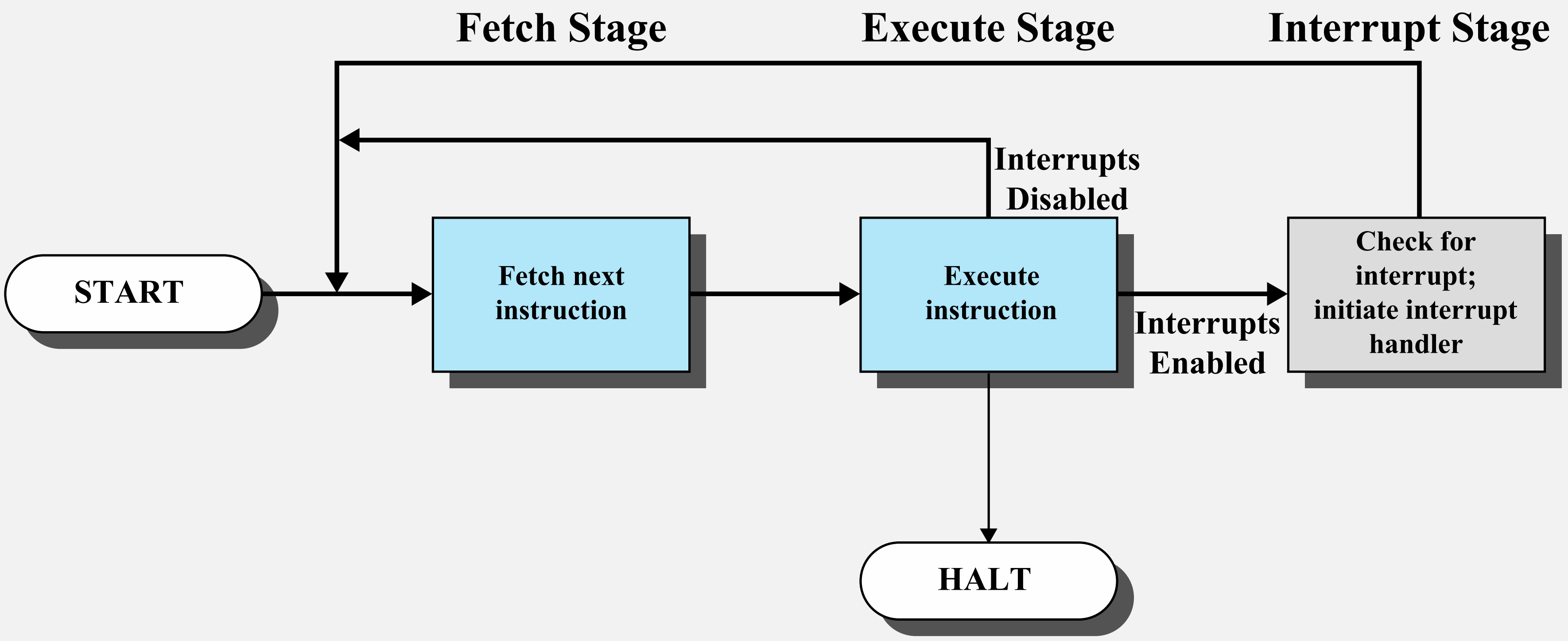


Figure 1.7 Instruction Cycle with Interrupts

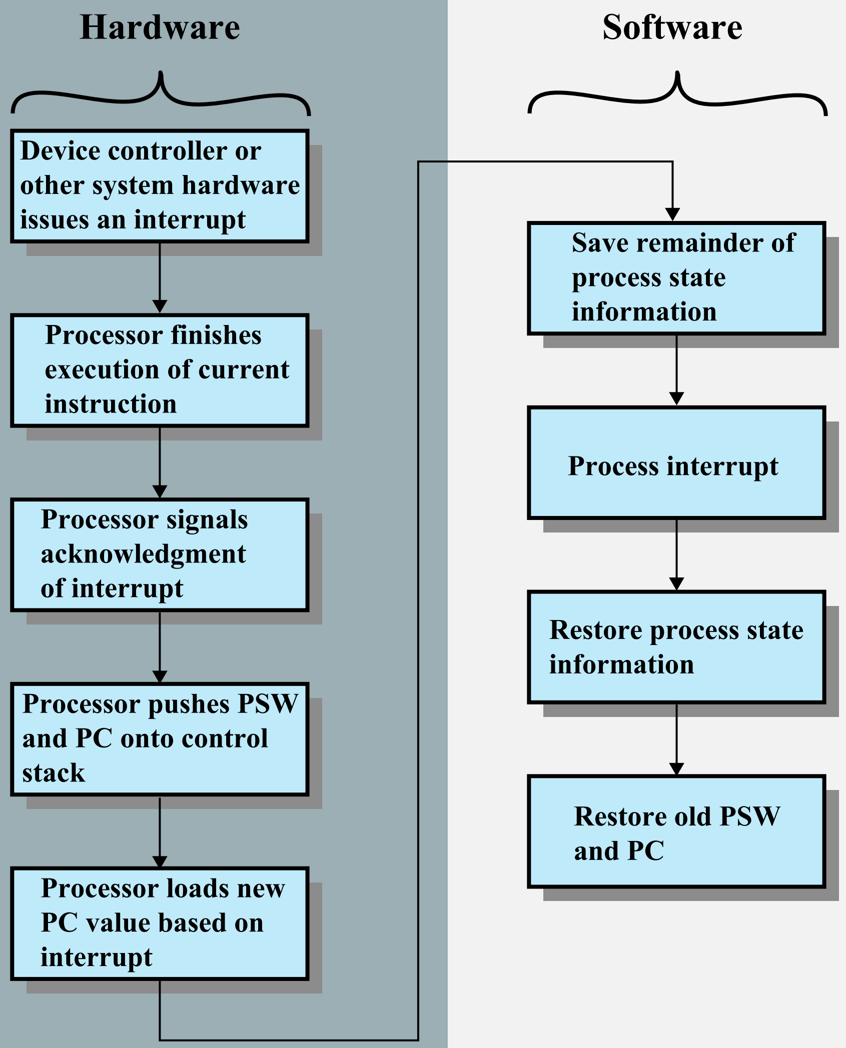

Hardware
Software
Device controller or other system hardware issues an interrupt
Processor finishes execution of current instruction
Processor signals acknowledgment of interrupt
Processor pushes PSW and PC onto control stack
Processor loads new PC value based on interrupt
Save remainder of process state information
Process interrupt
Restore process state information
Restore old PSW and PC

Figure 1.10 Simple Interrupt Processing

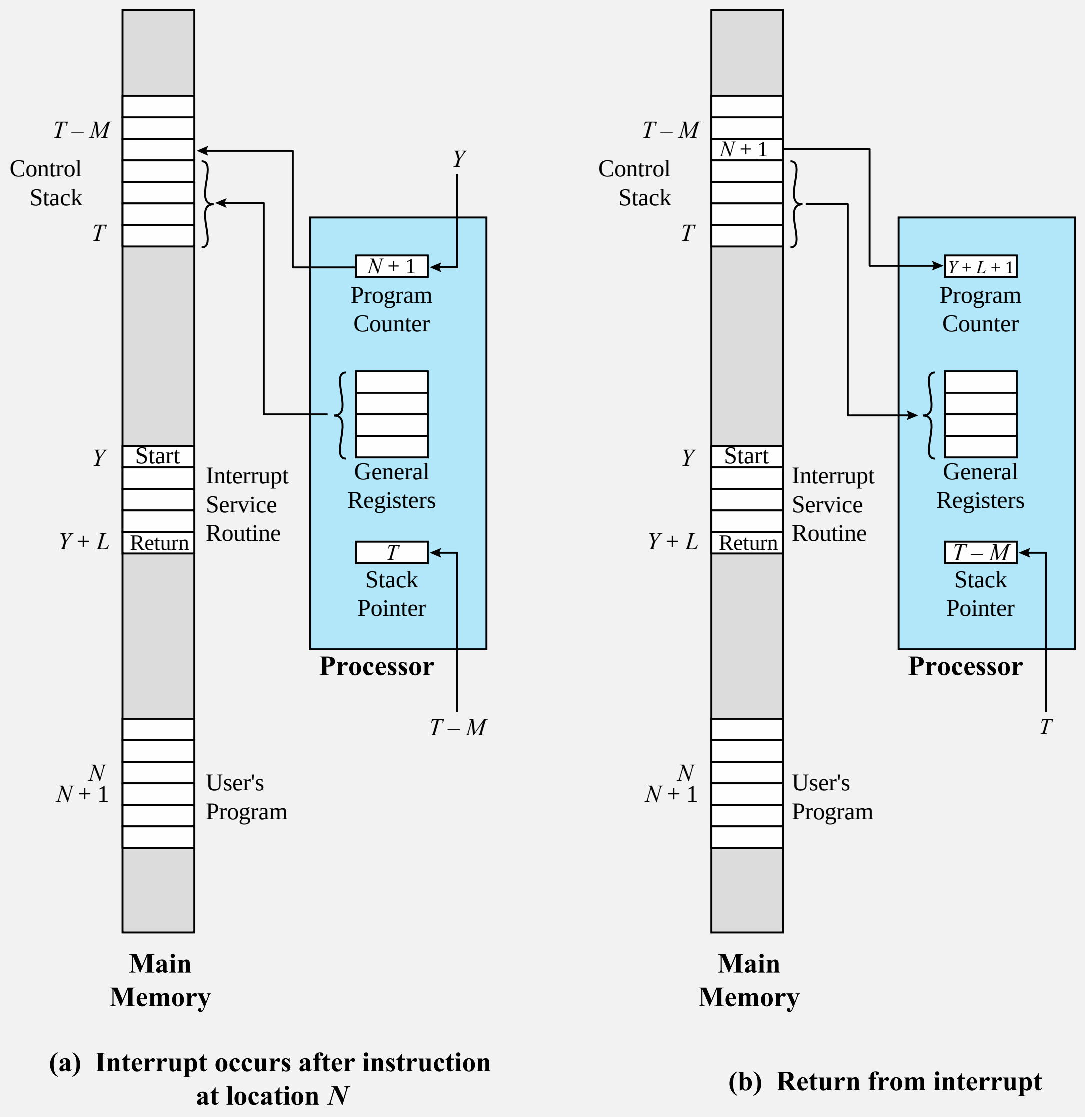


Figure 1.11 Changes in Memory and Registers for an Interrupt

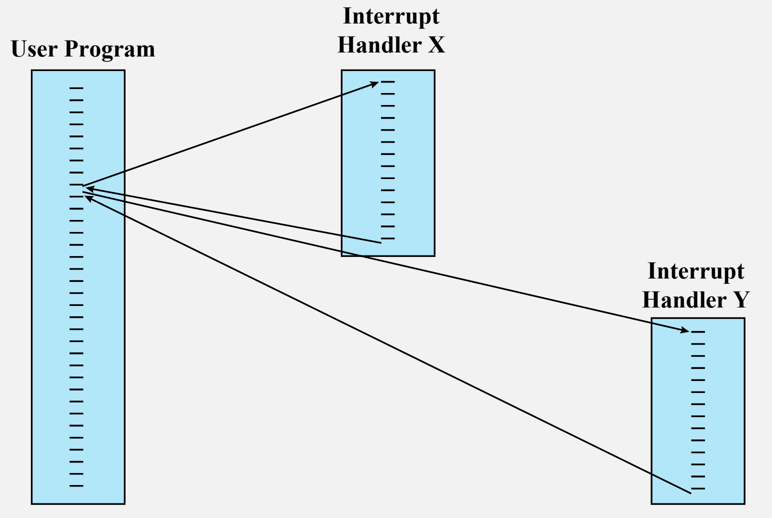
MULTIPLE INTERRUPTS

An interrupt occurs while another interrupt is being processed

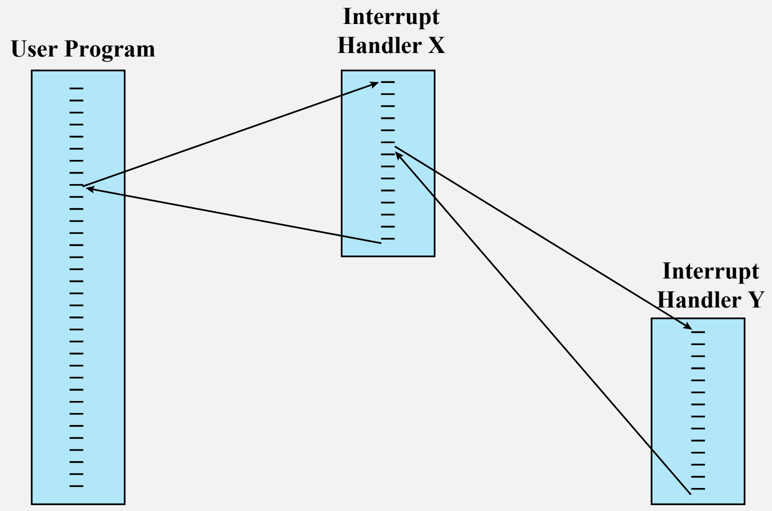
- e.g. receiving data from a communications line and printing results at the same time

Two approaches:

- Disable interrupts while an interrupt is being processed
- Use a priority scheme



(a) Sequential interrupt processing



(b) Nested interrupt processing

Figure 1.12 Transfer of Control with Multiple Interrupts

MEMORY HIERARCHY

- Design constraints on a computer's memory
  - How much?
  - How fast?
  - How expensive?
- If the capacity is there, applications will likely be developed to use it
- Memory must be able to keep up with the processor
- Cost of memory must be reasonable in relationship to the other components

MEMORY RELATIONSHIPS

Faster access time = greater cost per bit

Greater capacity = smaller cost per bit

Greater capacity = slower access speed

THE 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

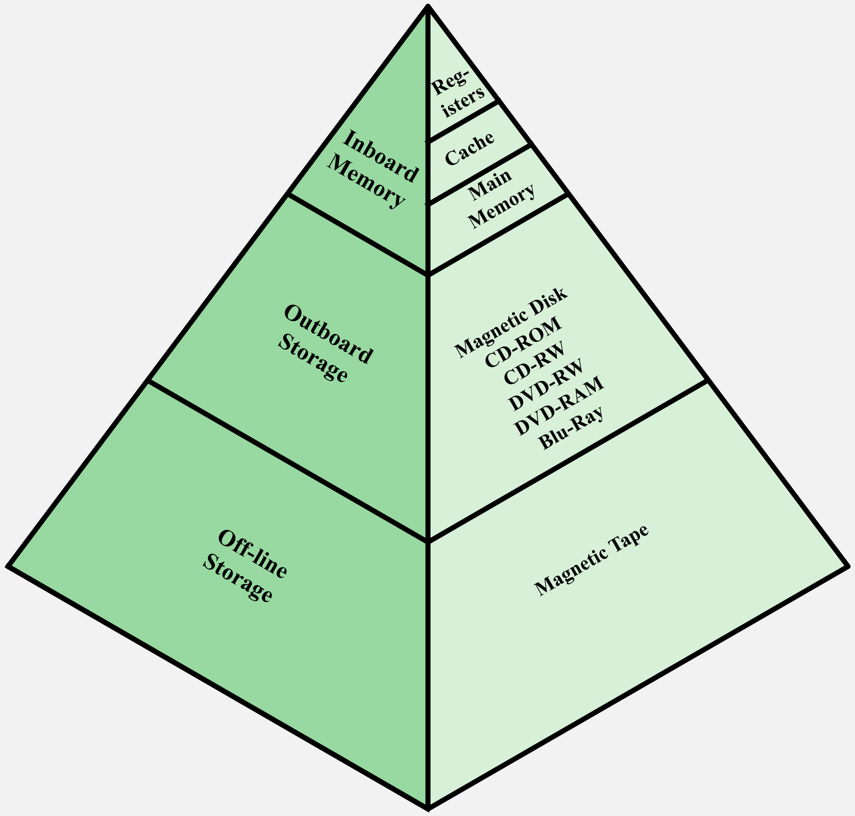


Figure 1.14 The Memory Hierarchy

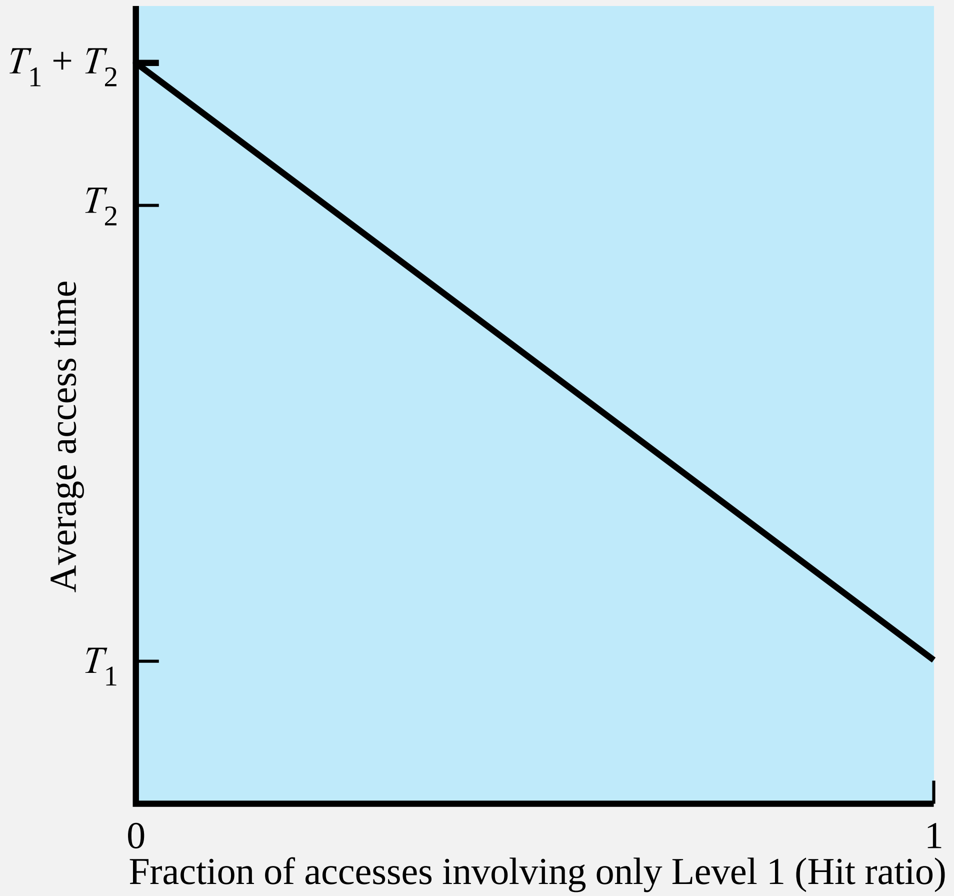


Figure 1.15 Performance of a Simple Two-Level Memory

PRINCIPLE OF LOCALITY

- Memory references by the processor tend to cluster
- Data is organized so that the percentage of accesses to each successively lower level is substantially less than that of the level above
- Can be applied across more than two levels of memory

Secondary Memory

Also referred to as auxiliary memory

- External
- Nonvolatile
- Used to store program and data files

CACHE MEMORY

- Invisible to the OS
- Interacts with other memory management hardware
- Processor must access memory at least once per instruction cycle
- Processor execution is limited by memory cycle time
- Exploit the principle of locality with a small, fast memory

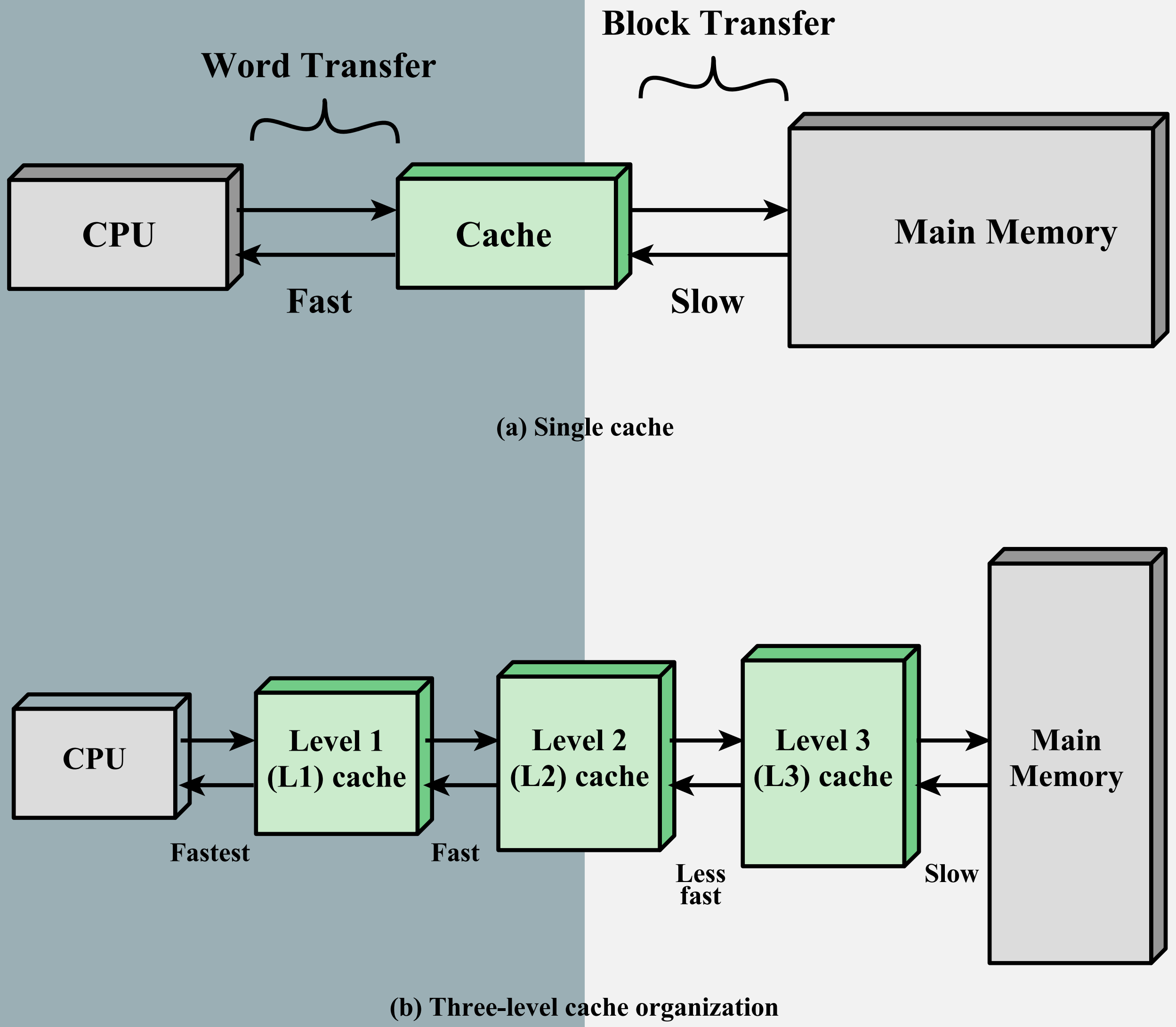


Figure 1.16 Cache and Main Memory

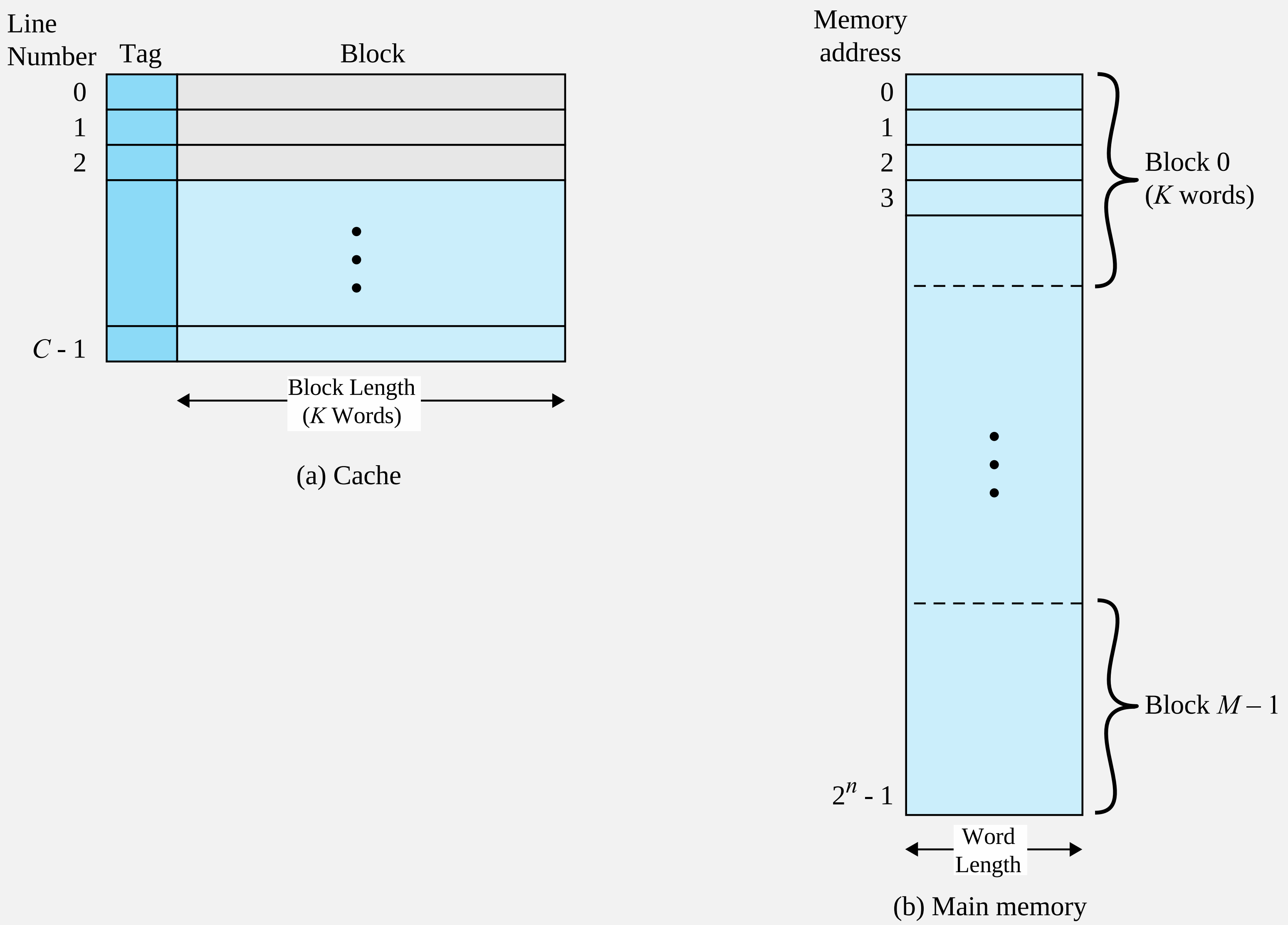


Figure 1.17 Cache/Main-Memory Structure

MULTICORE COMPUTER

- Also known as a chip multiprocessor
- Combines two or more processors (cores) on a single piece of silicon (die)
  - Each core consists of all of the components of an independent processor
- In addition, multicore chips also include L2 cache and in some cases L3 cache

SUMMARY

- Basic Elements
- Evolution of the microprocessor
- Instruction execution
- Interrupts
  - Interrupts and the instruction cycle
  - Interrupt processing
  - Multiple interrupts
- The memory hierarchy
- Cache memory
  - Motivation
  - Cache principles
  - Cache design
- Direct memory access
- Multiprocessor and multicore organization
  - Symmetric multiprocessors
  - Multicore computers