Operating Systems: Internals and Design Principles

Chapter 1 Computer System Overview

Seventh Edition

By William Stallings

Operating Systems: Internals and Design Principles

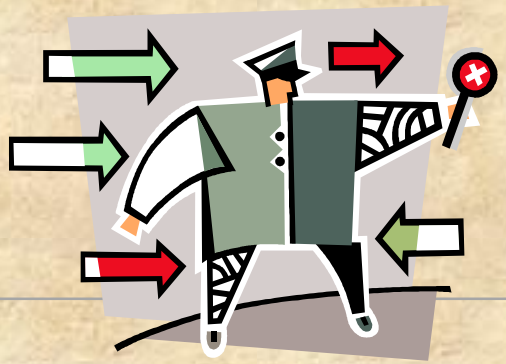
"No artifact designed by man is so convenient for this kind of functional description as a digital computer. Almost the only ones of its properties that are detectable in its behavior are the organizational properties. Almost no interesting statement that one can make about on operating computer bears any particular relation to the specific nature of the hardware. A computer is an organization of elementary functional components in which, to a high approximation, only the function performed by those components is relevant to the behavior of the whole system."

THE SCIENCES OF THE ARTIFICIAL ,

Herbert Simon

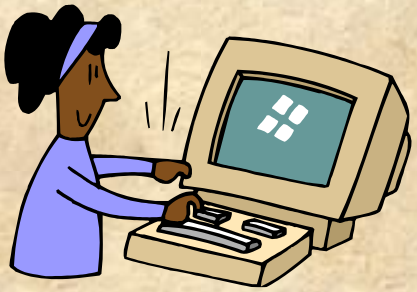
Operating System

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



Basic Elements

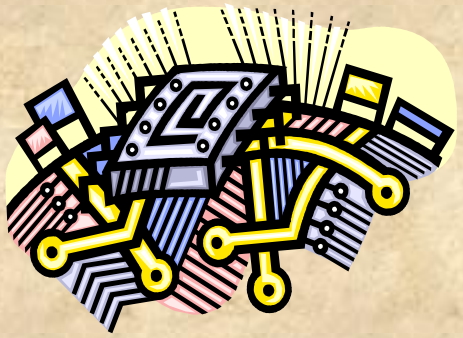
- Processor
- I/O Modules
- Main Memory
- System Bus



Processor

Controls the operation of the computer

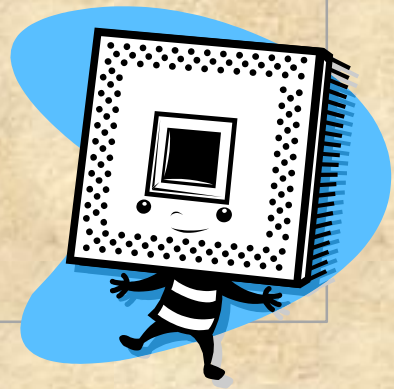
Performs the data processing functions



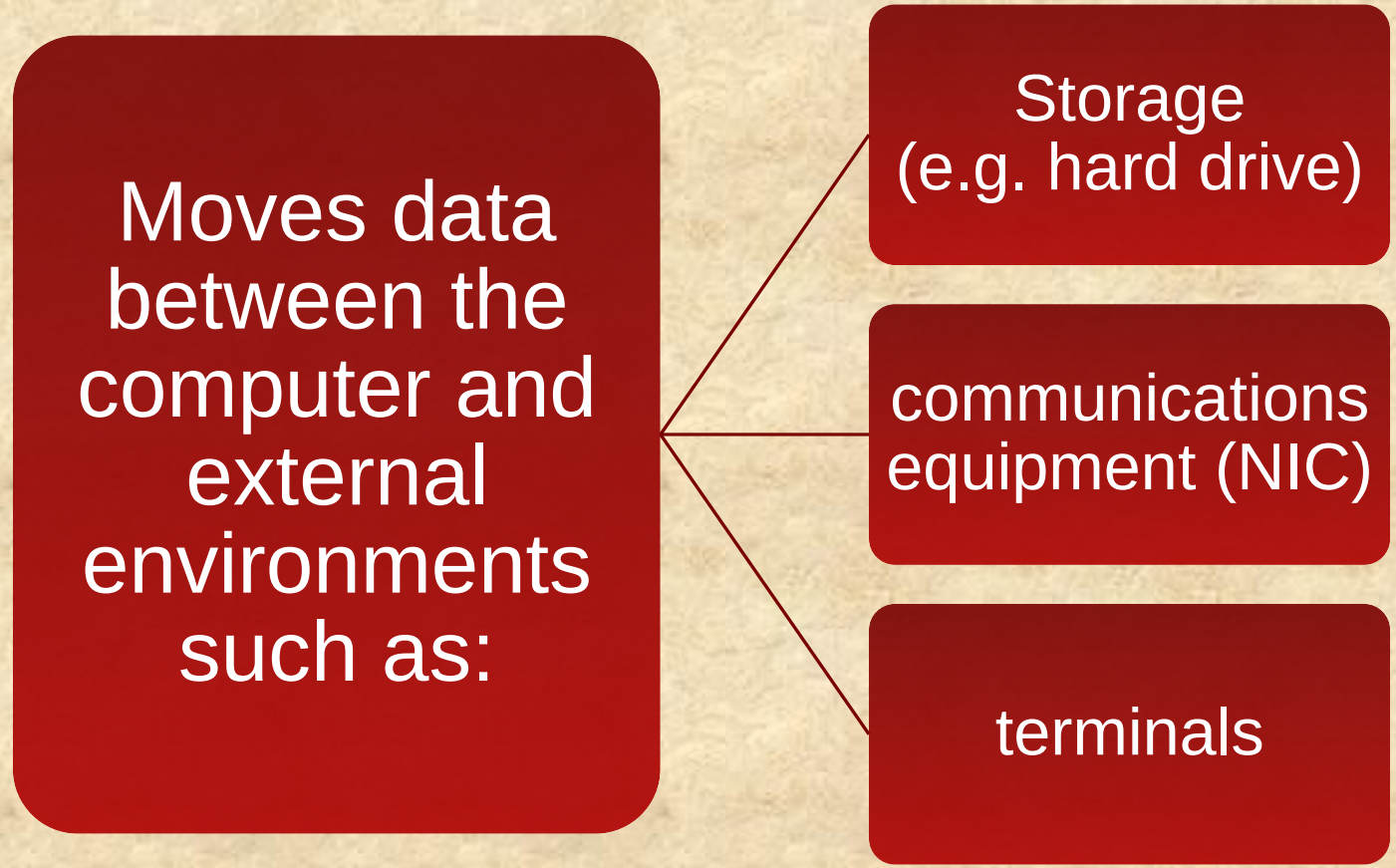
Referred to as the *Central Processing Unit* (CPU)

Main Memory

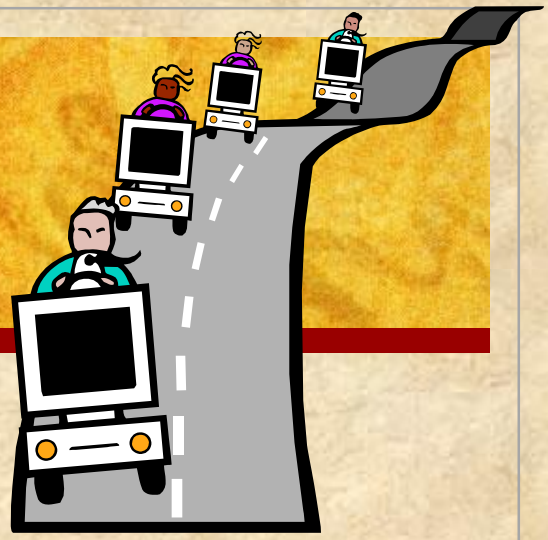
- Volatile
- Contents of the memory is lost when the computer is shut down
- Referred to as real memory or primary memory



I/O Modules


Moves data between the computer and external environments such as:
Storage (e.g. hard drive)
communications equipment (NIC)
terminals

System Bus



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

Top-Level View

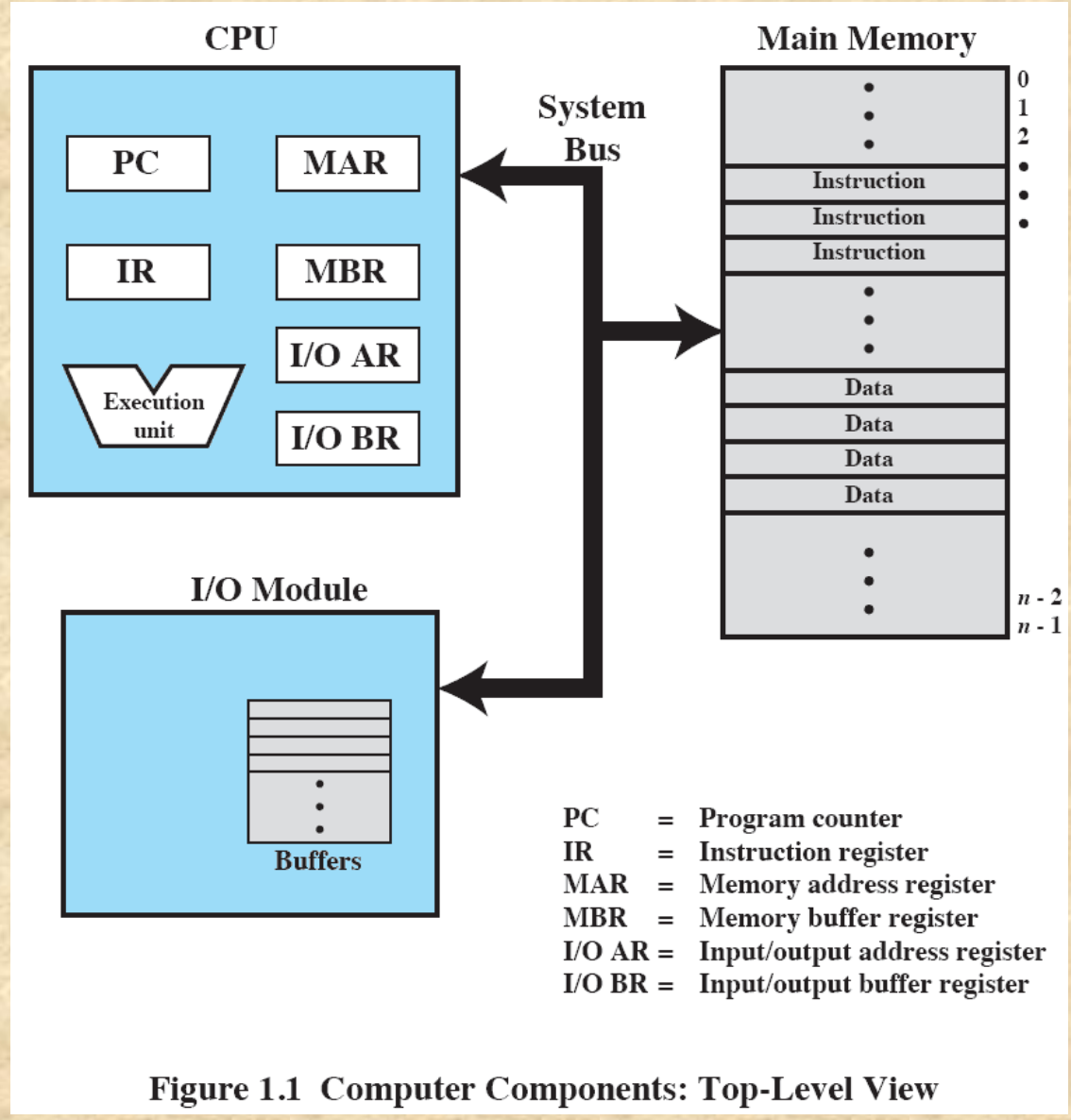
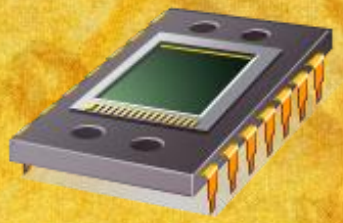


Figure 1.1 Computer Components: Top-Level View



Microprocessor

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

Graphical Processing Units (GPU's)

- Provide efficient computation on arrays of data using Single-Instruction Multiple Data (SIMD) techniques
- Used for general numerical processing
- Physics simulations for games
- Computations on large spreadsheets

G P U

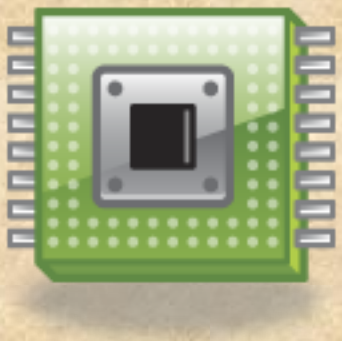
Digital Signal Processors (DSPs)

- Deal with streaming signals such as audio or video
- Used to be embedded in devices like modems
- Encoding/decoding speech and video (codecs)
- Support for encryption and security

D S P

System on a Chip (SoC)

- To satisfy the requirements of handheld devices, the microprocessor is giving way to the SoC
- Components such as DSPs, GPUs, codecs and main memory, in addition to the CPUs and caches, are on the same chip



Instruction Execution

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

Two steps:

- processor reads (fetches) instructions from memory
- processor executes each instruction

Basic Instruction Cycle

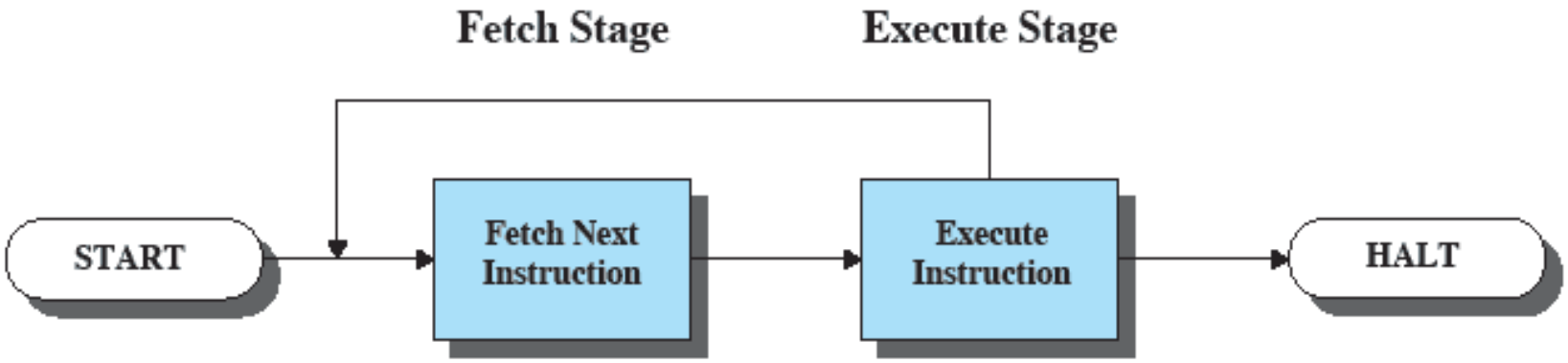
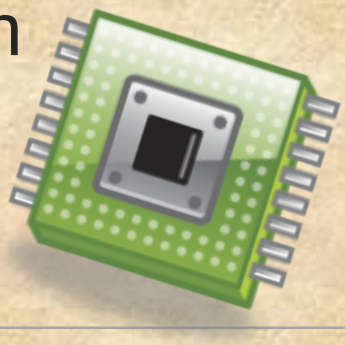


Figure 1.2 Basic Instruction Cycle

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

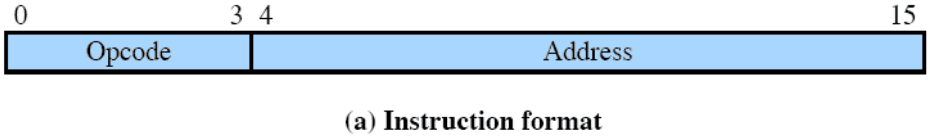


Instruction Register (IR)

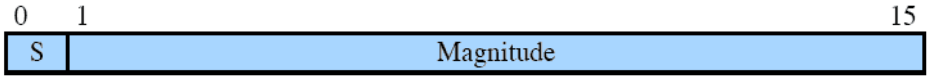
Fetched instruction is loaded into Instruction Register (IR)

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

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

Figure 1.3 Characteristics of a Hypothetical Machine

Example of Program Execution

load 940

add 941

store 941

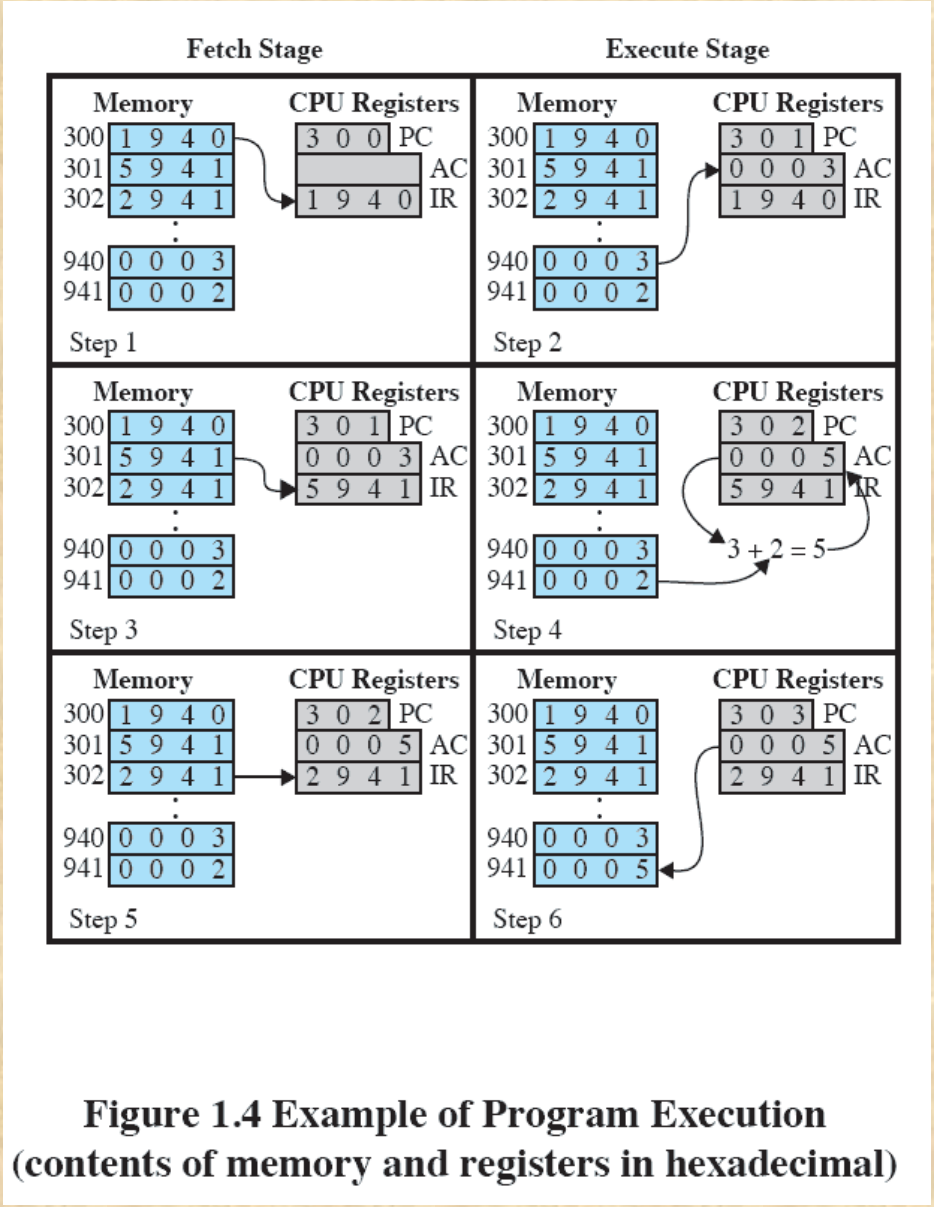
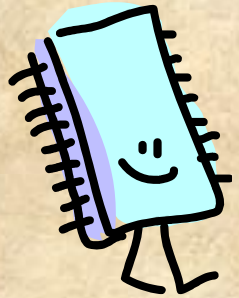


Figure 1.4 Example of Program Execution (contents of memory and registers in hexadecimal)

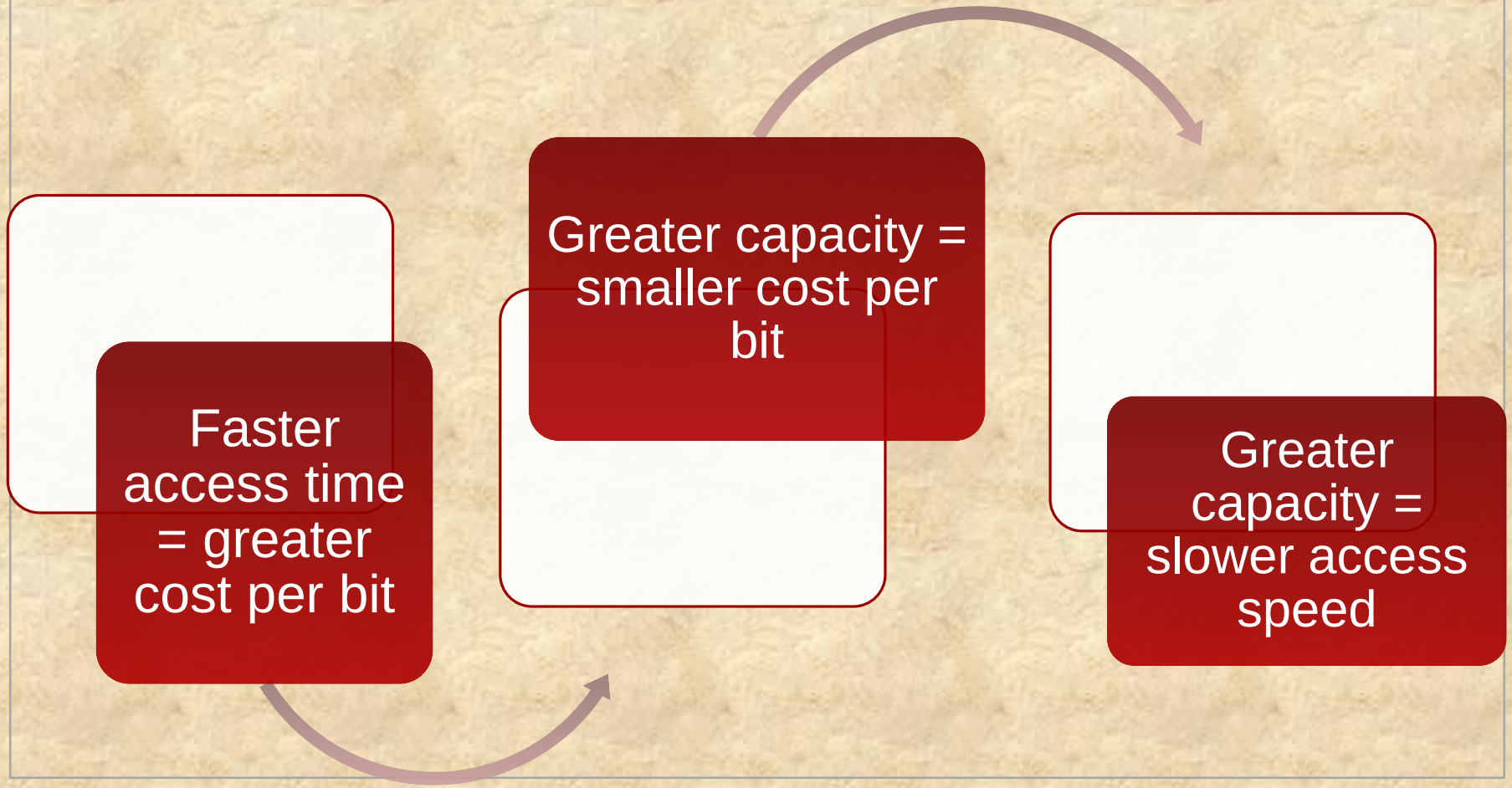
Memory Hierarchy

- Major constraints in memory
  - amount
  - speed
  - Expense (cost)



- Memory must be able to keep up with the processor
- Cost of memory must be reasonable in relationship to the other components

Memory Relationships



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

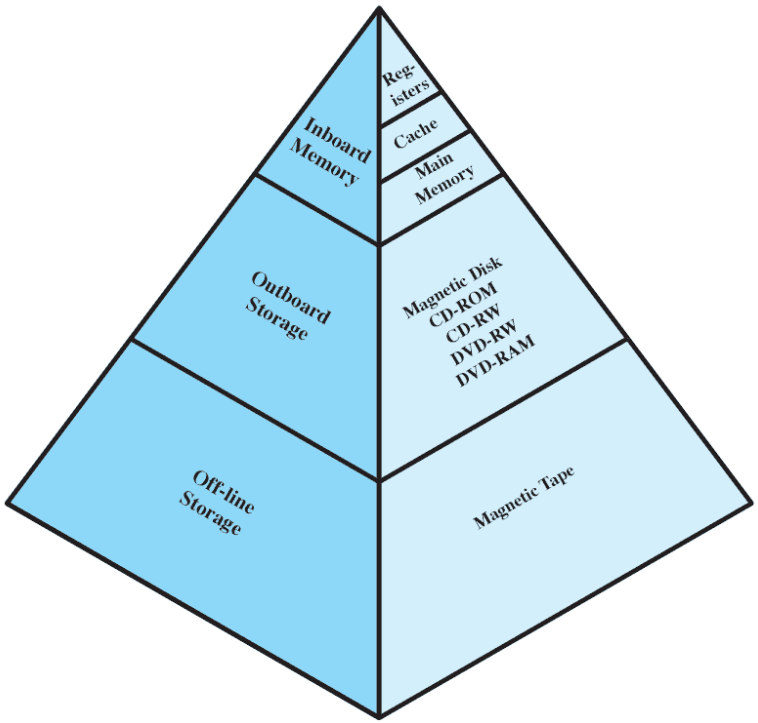
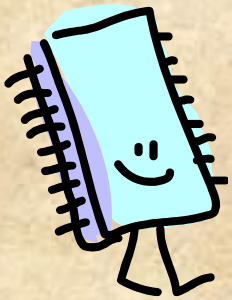


Figure 1.14 The Memory Hierarchy

Performance of a Simple Two-Level Memory



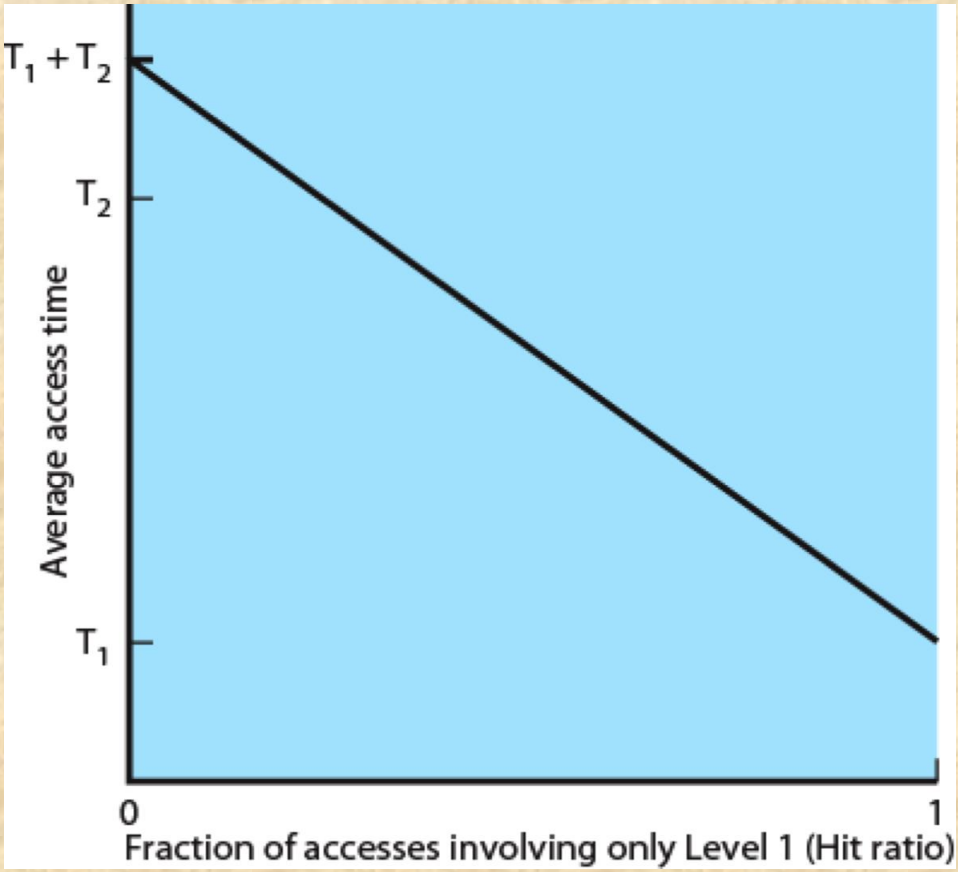


Figure 1.15 Performance of a Simple Two-Level Memory

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

Types of Locality

- Spatial locality: tendency of execution to involve a number of memory **locations** that are clustered
- Temporal locality: tendency for a processor to access memory locations that have been used **recently**

Clicker: Locality

Processor access instructions/data sequentially...

A. Spatial locality: tendency of execution to involve a number of memory locations that are clustered

B. Temporal locality: tendency for a processor to access memory locations that have been used recently

Clicker: Locality

When an iteration (`for`) loop is executed...

A. Spatial locality: tendency of execution to involve a number of memory locations that are clustered

B. Temporal locality: tendency for a processor to access memory locations that have been used recently

How to Exploit Locality?

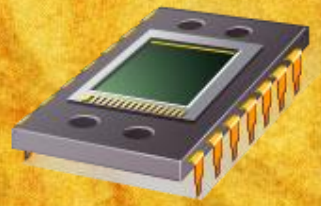
- Spatial locality: use larger cache and pre-fetching
- Temporal locality: keep recently used instruction/data in cache and exploit cache hierarchy

Secondary Memory

Also referred to as auxiliary memory

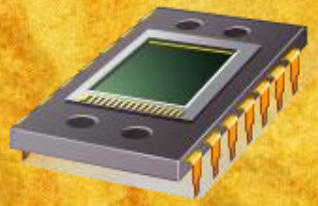
- External
- Nonvolatile
- Used to store program and data files

Cache Memory



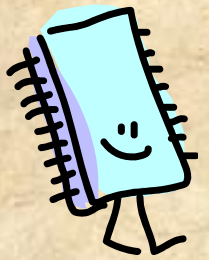
- Invisible to the OS
- Interacts with other memory management hardware
- Processor must access memory at least once per instruction cycle
- Processor execution is limited by memory cycle time
- Exploit the principle of locality with a small, fast memory

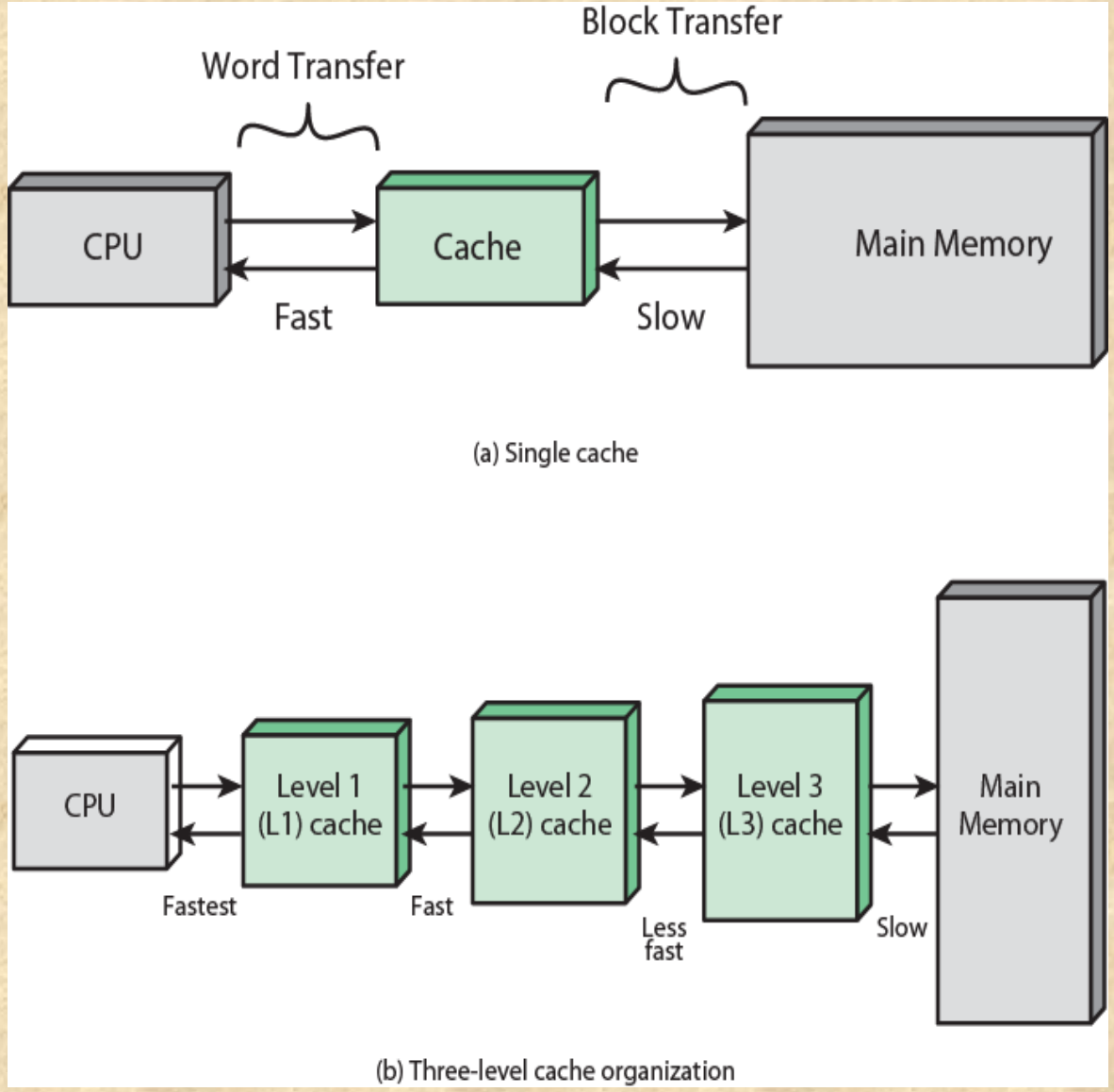
Cache Principles



- Contains a copy of a portion of main memory
- Processor first checks cache
- If not found, a block of memory is read into cache
- Because of locality of reference, it is likely that many of the future memory references will be to other bytes in the block

Cache and Main Memory





Cache/Main-Memory Structure

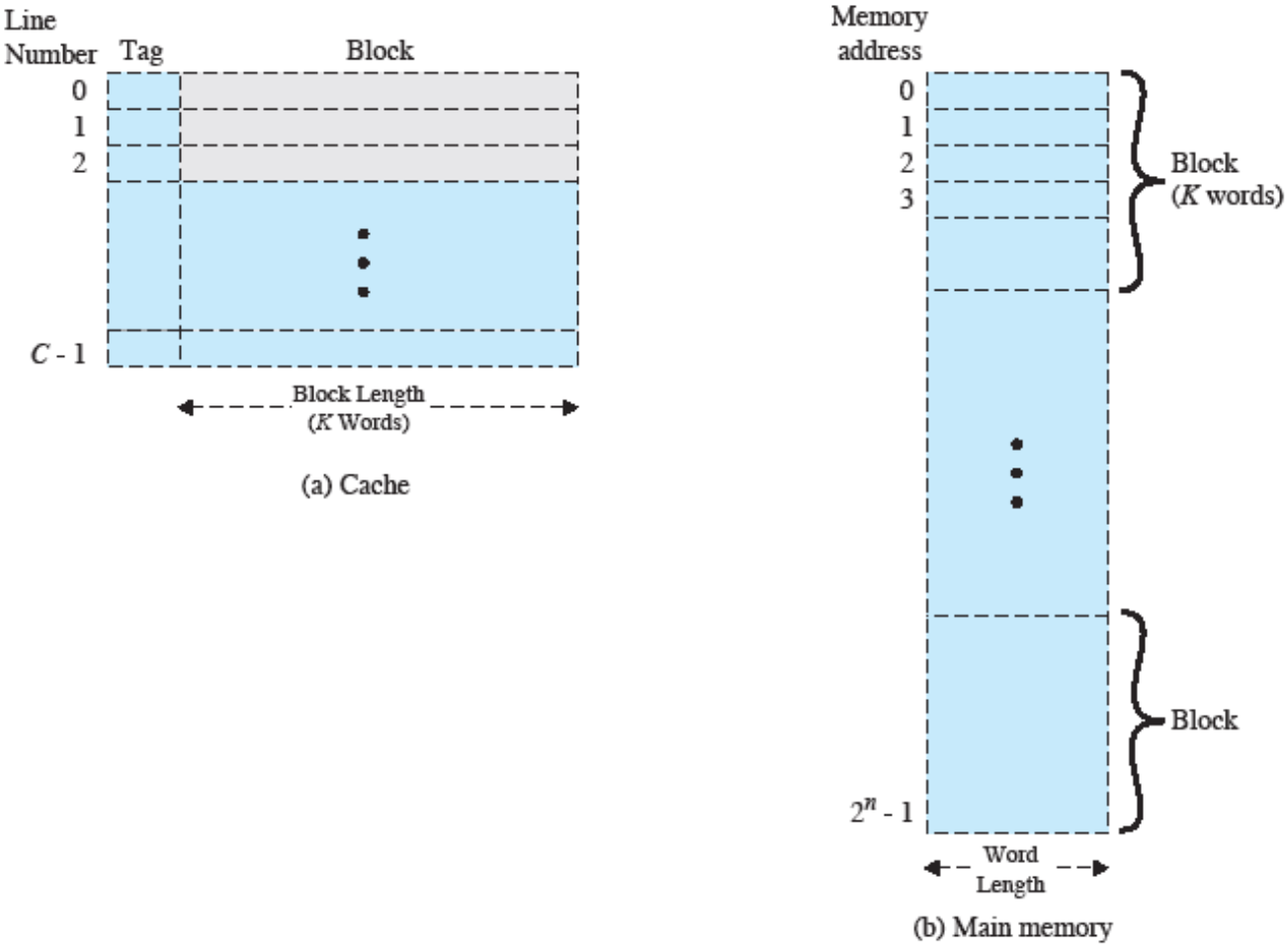
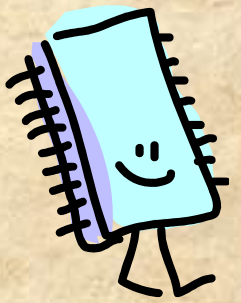


Figure 1.17 Cache/Main-Memory Structure

Cache Read Operation



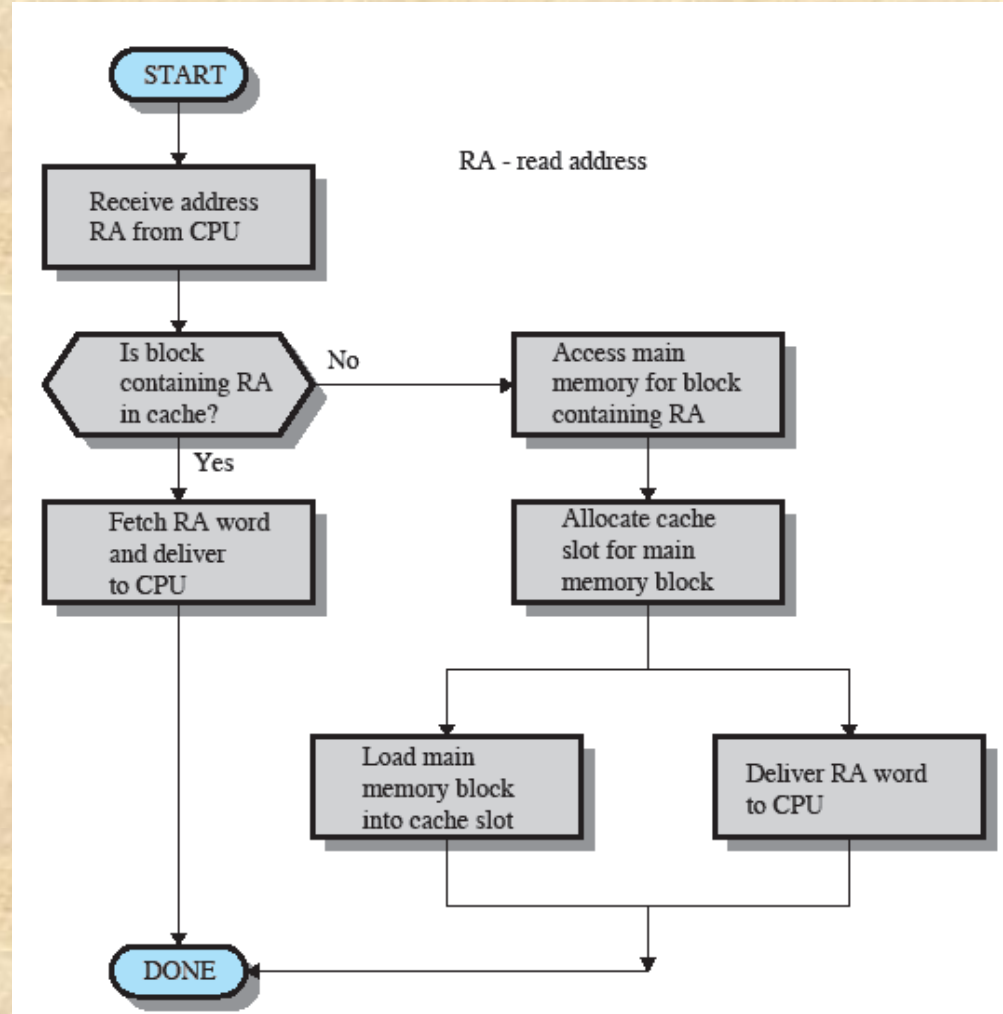
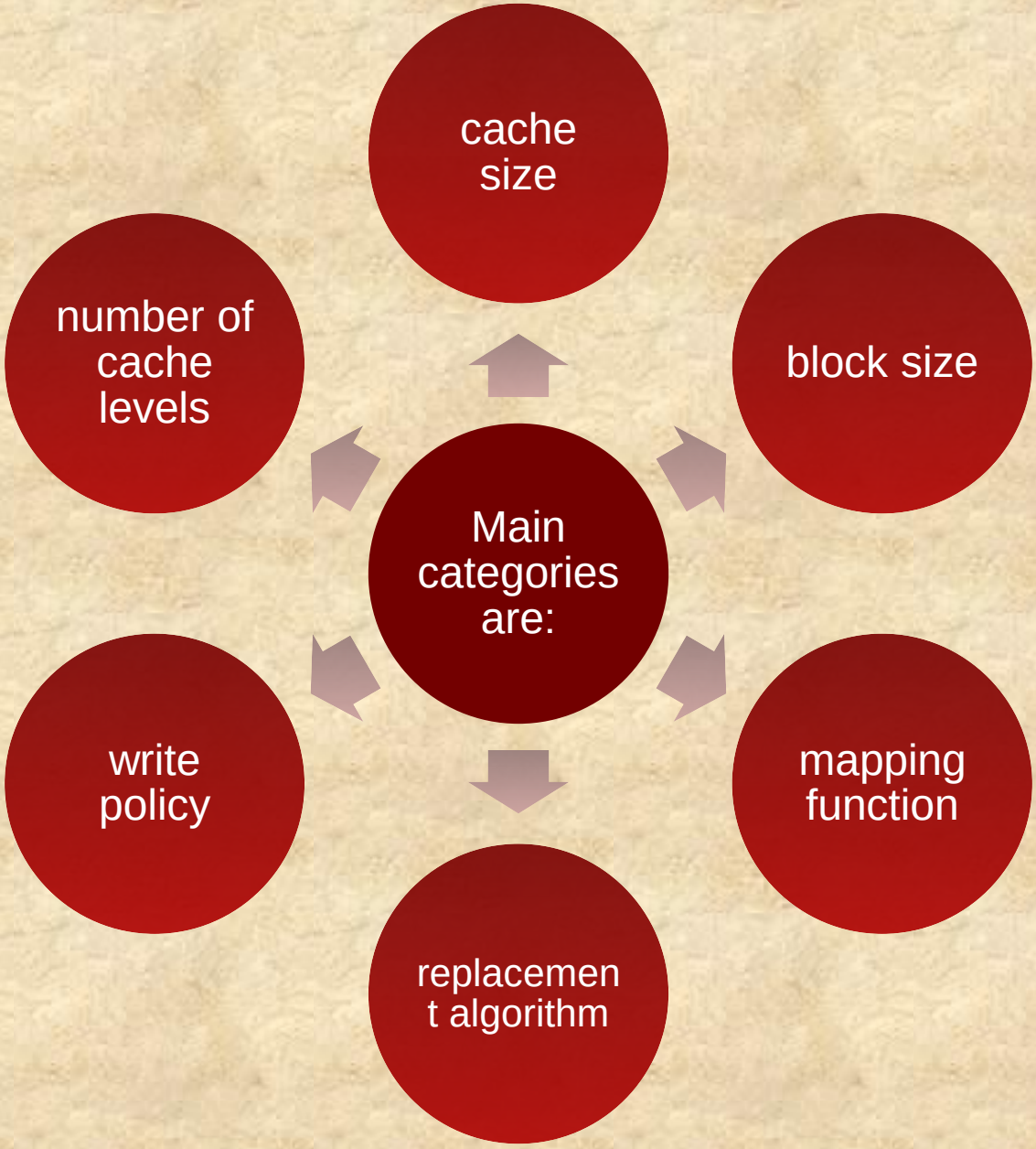


Figure 1.18 Cache Read Operation

Cache Design



Cache and Block Size

Cache Size

Small caches have significant impact on performance

Block Size

The unit of data exchanged between cache and main memory

Mapping Function

* Determines which cache location the block will occupy

Two constraints affect design:

When one block is read in, another may have to be replaced

The more flexible the mapping function, the more complex is the circuitry required to search the cache

Interrupts

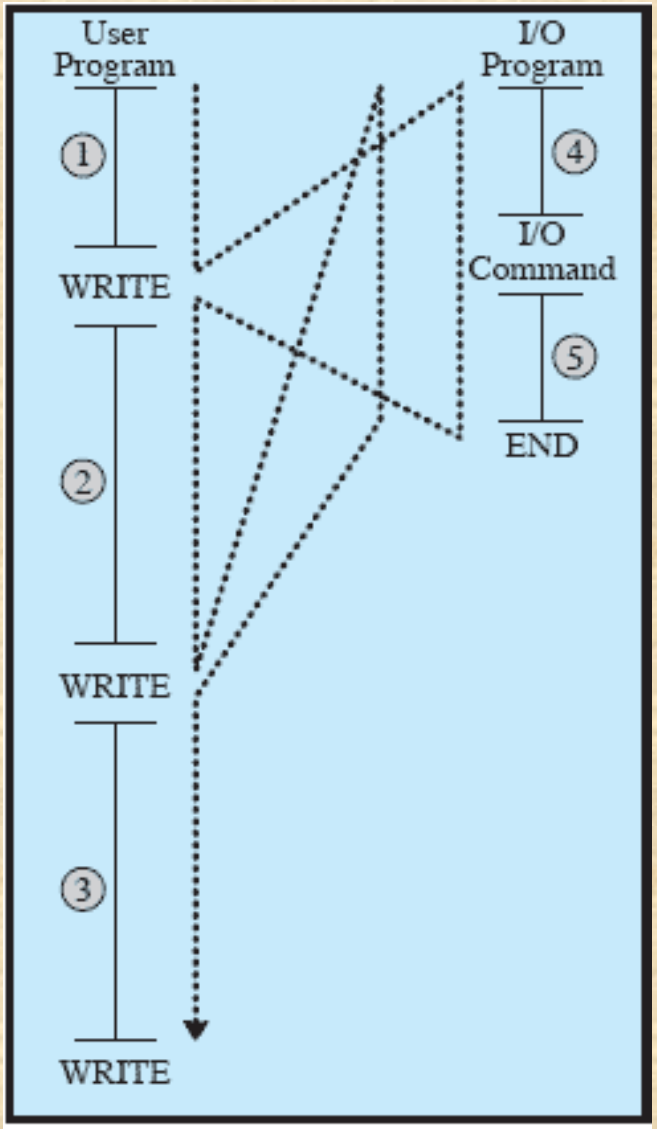
- Interrupt the normal sequencing of the processor
- Provided to improve processor utilization
  - most I/O devices are slower than the processor
  - processor must pause to wait for device
  - wasteful use of the processor

Common Classes of Interrupts

Table 1.1 Classes of Interrupts

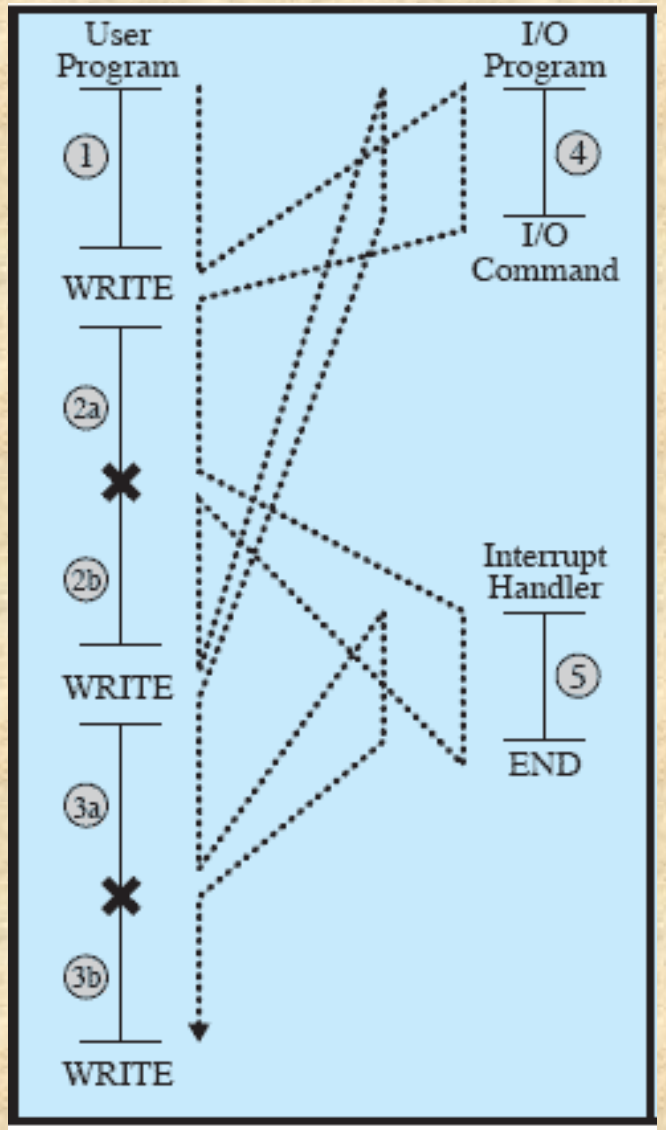
Program	Generated by some condition that occurs as a result of an instruction execution, such as arithmetic overflow, division by zero, attempt to execute an illegal machine instruction, and reference outside a user's allowed memory space.
Timer	Generated by a timer within the processor. This allows 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.

Flow of Control Without Interrupts



(a) No interrupts

Interrupts: Short I/O Wait



(b) Interrupts; short I/O wait

Transfer of Control via Interrupts

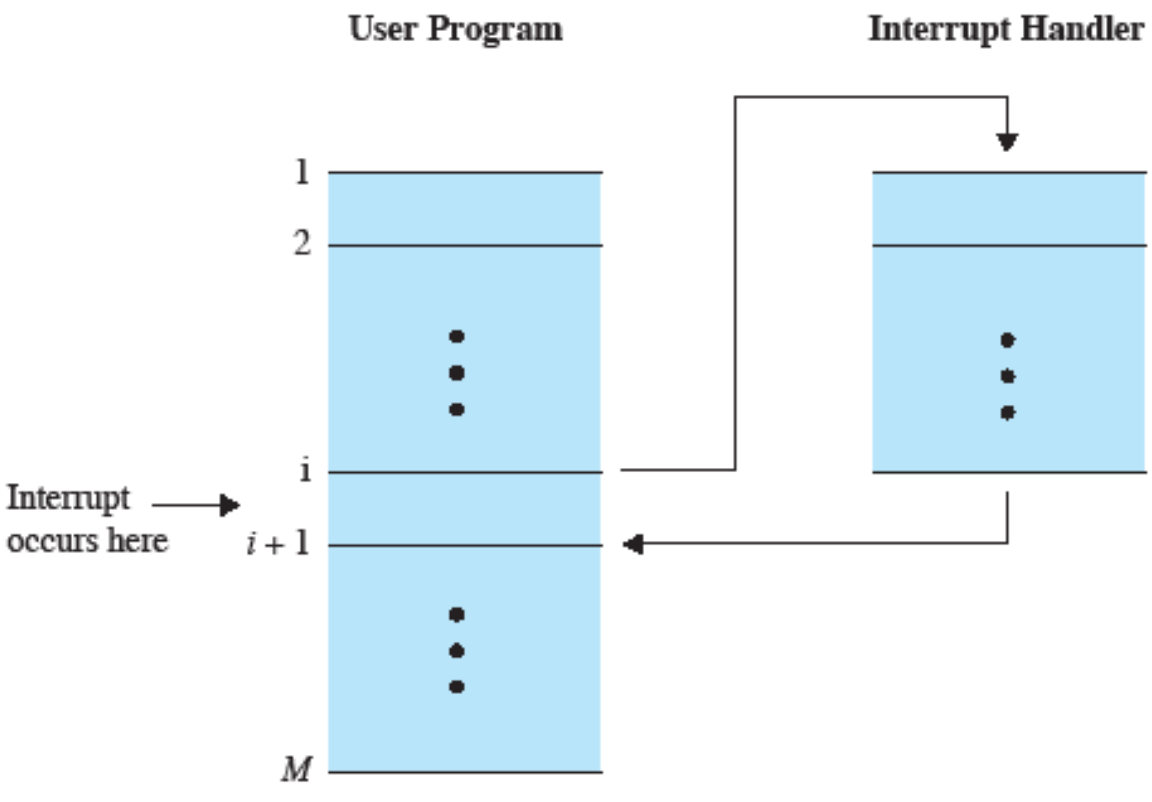


Figure 1.6 Transfer of Control via Interrupts

Instruction Cycle With Interrupts

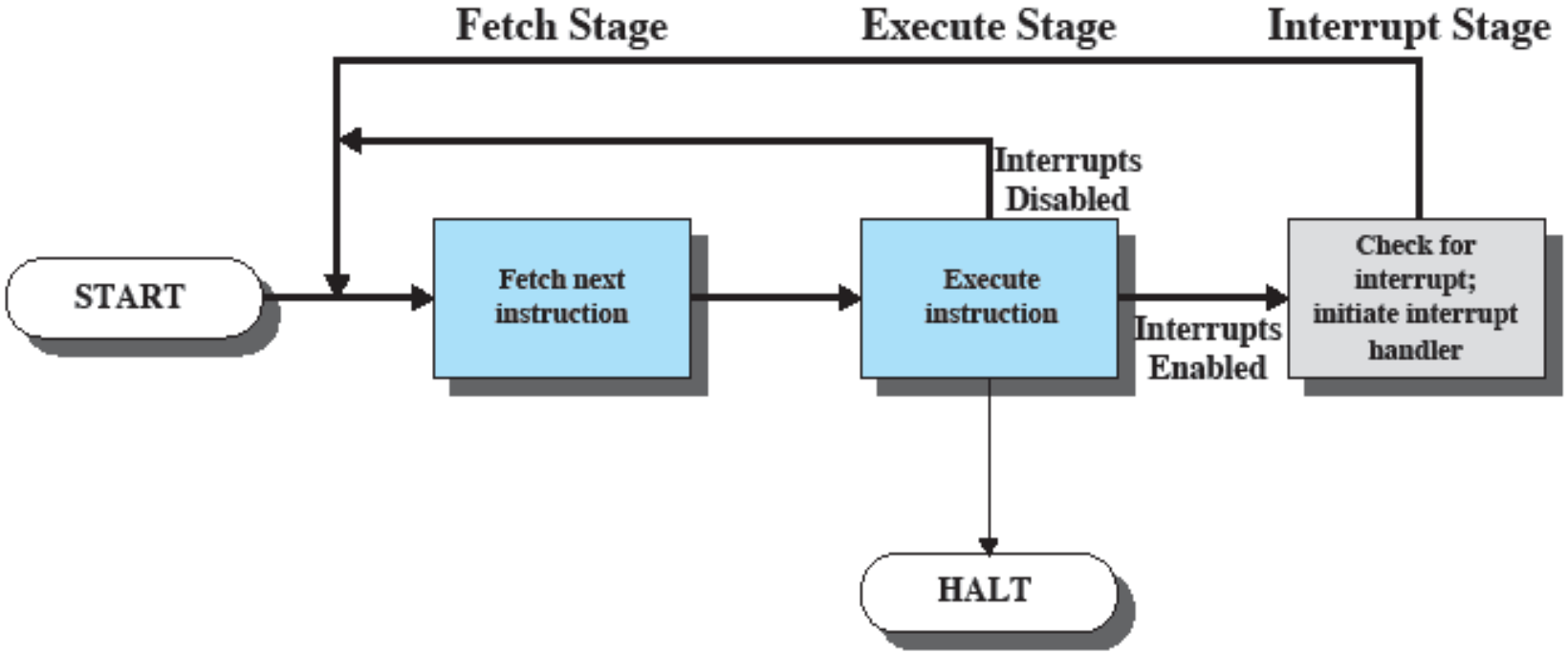
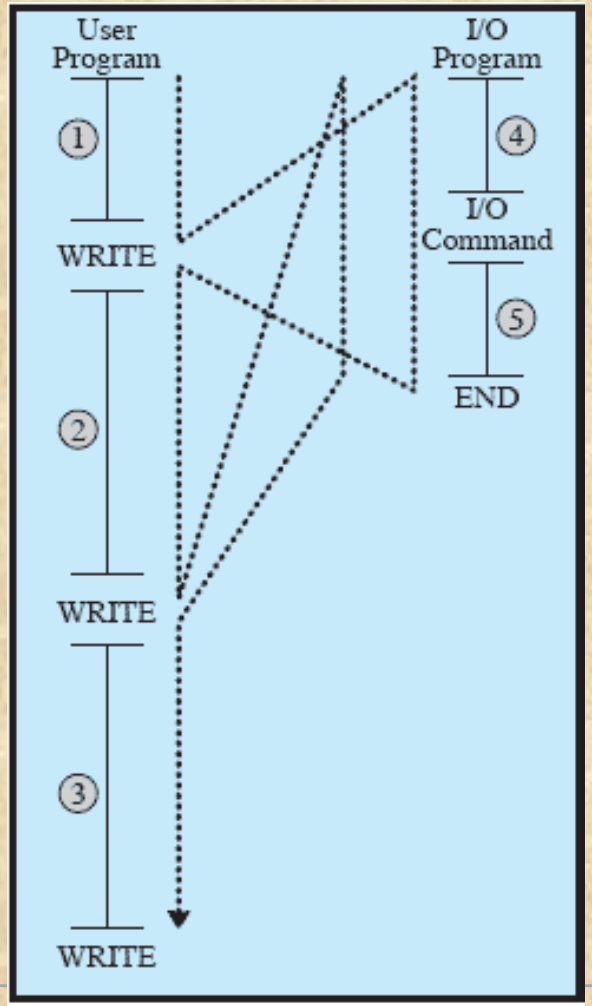


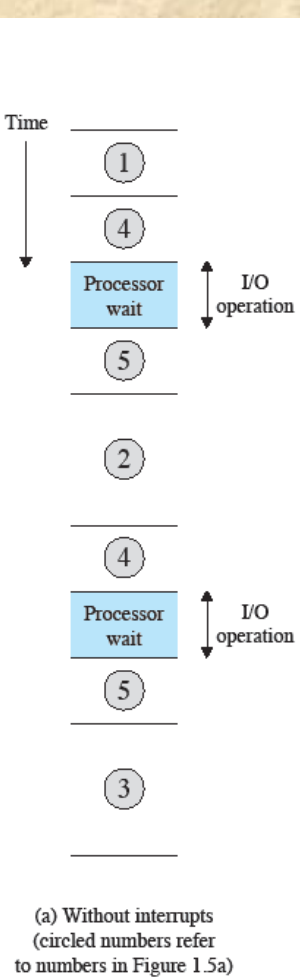
Figure 1.7 Instruction Cycle with Interrupts

Program Timing:

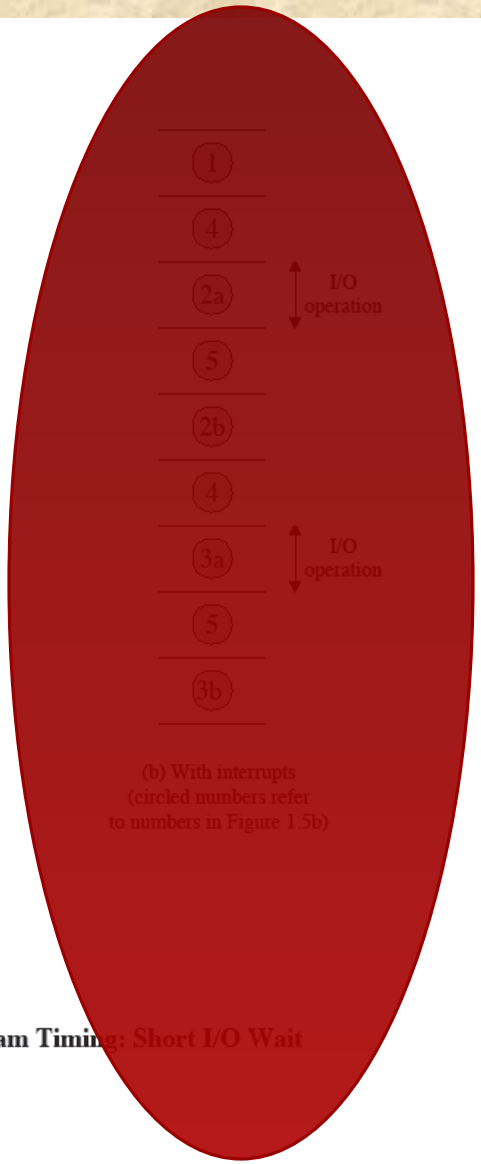
No Interrupt



(a) No interrupts



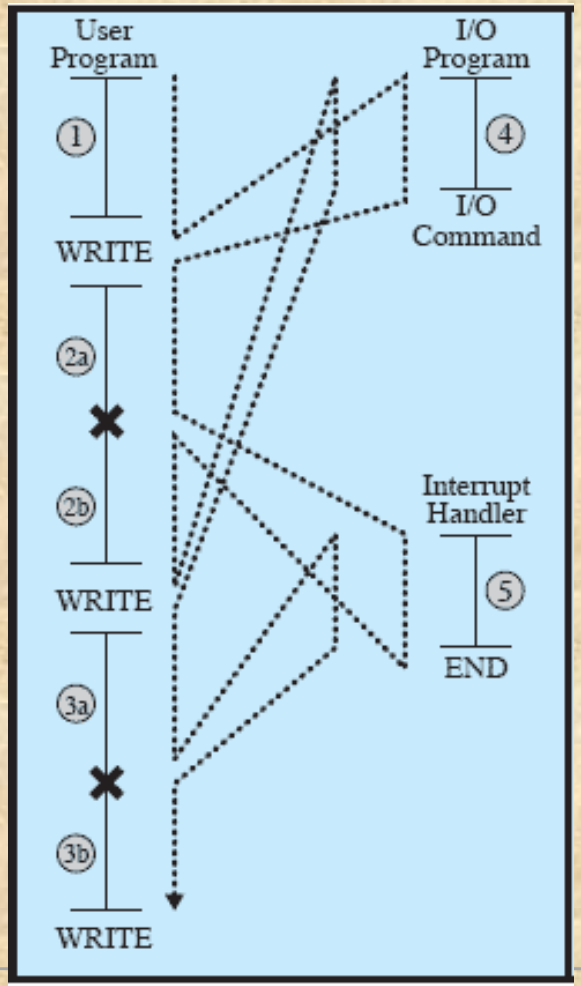
(a) Without interrupts
(circled numbers refer
to numbers in Figure 1.5a)



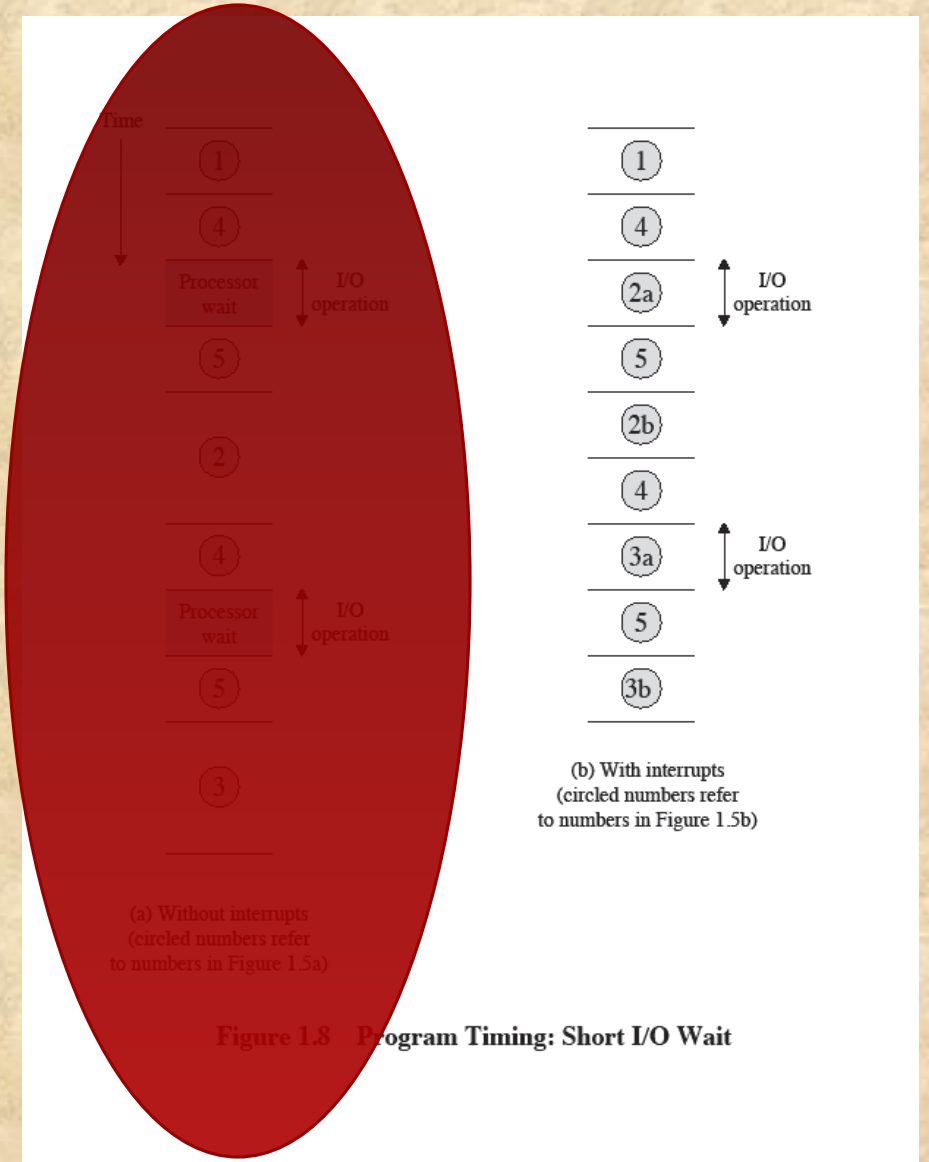
(b) With interrupts
(circled numbers refer
to numbers in Figure 1.5b)

Figure 1.8 Program Timing: Short I/O Wait

Program Timing: Short I/O Wait



(b) Interrupts; short I/O wait



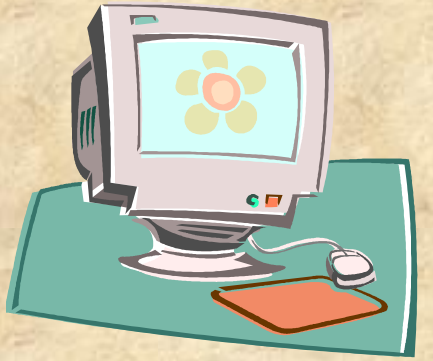
(a) Without interrupts (circled numbers refer to numbers in Figure 1.5a)

(b) With interrupts (circled numbers refer to numbers in Figure 1.5b)

Figure 1.8 Program Timing: Short I/O Wait

Program Timing:

Long I/O wait



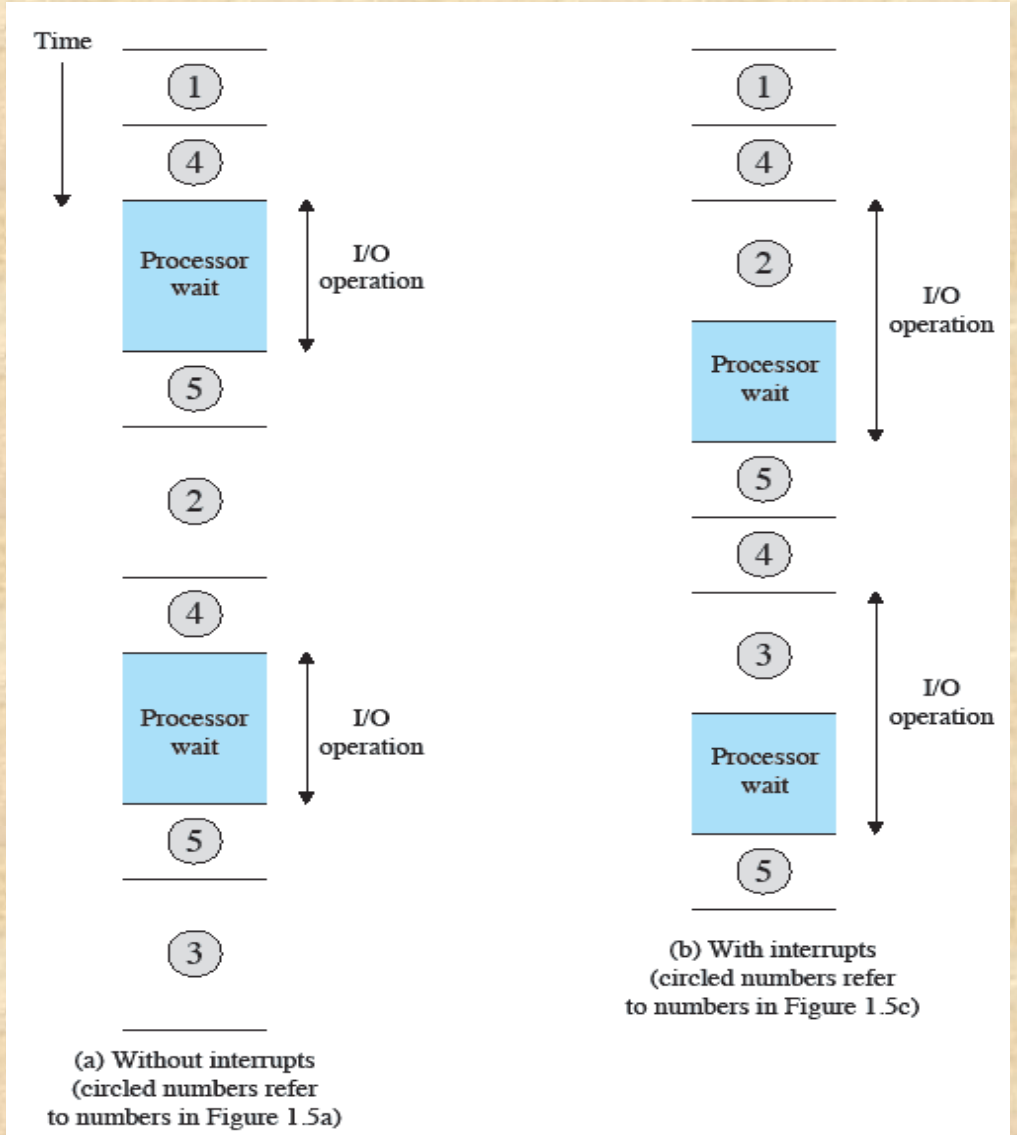
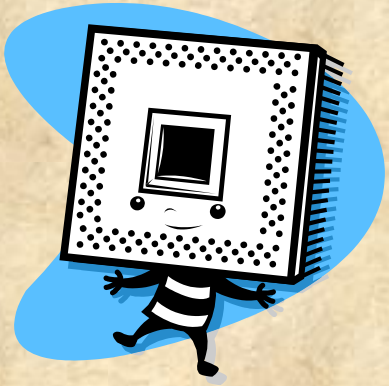


Figure 1.9 Program Timing: Long I/O Wait

Simple Interrupt Processing



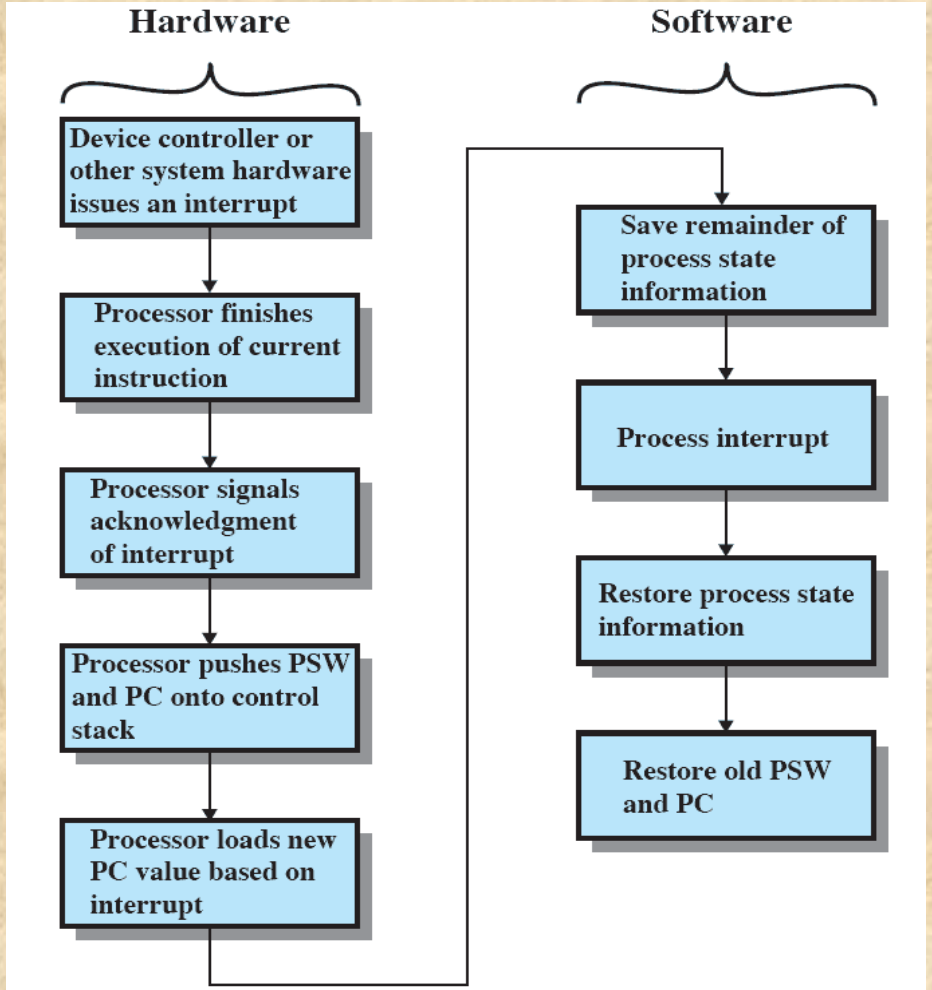
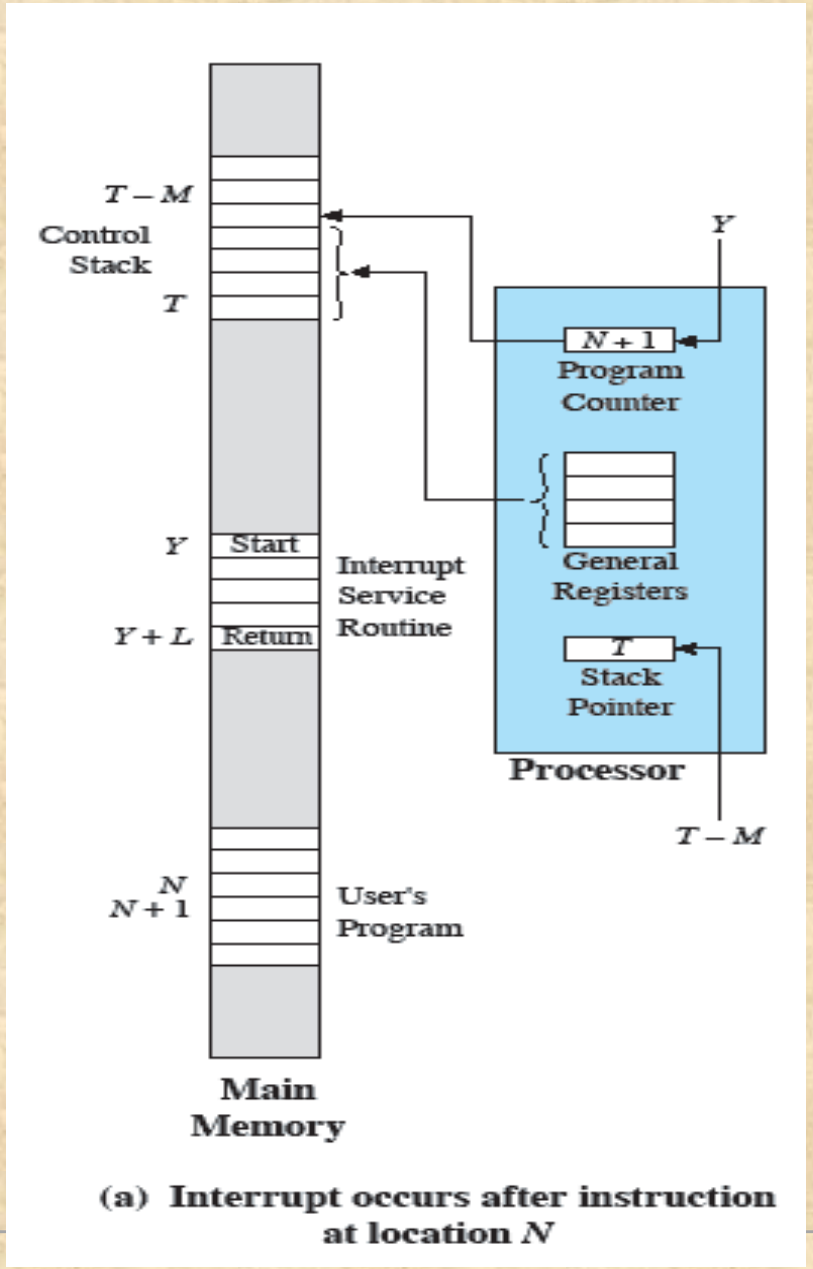
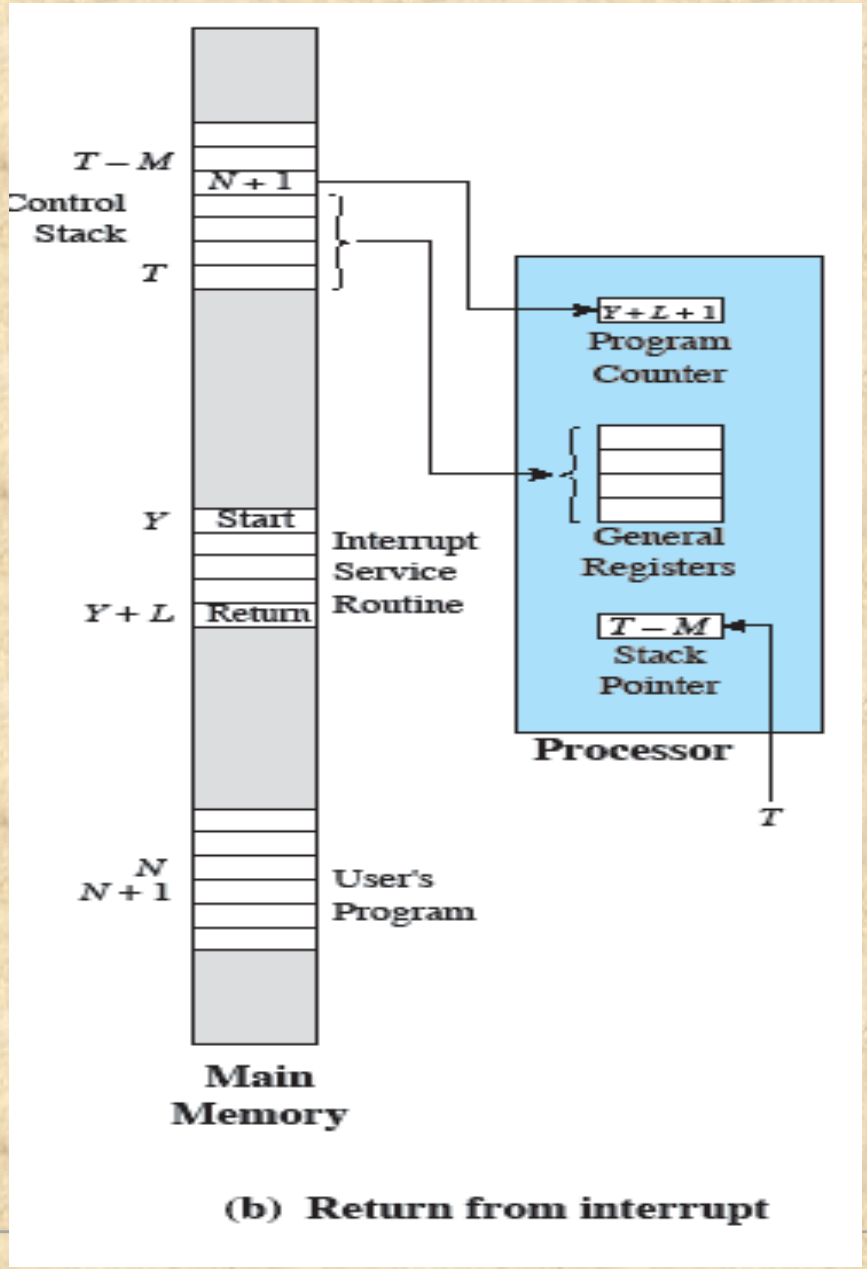


Figure 1.10 Simple Interrupt Processing



(a) Interrupt occurs after instruction at location N



(b) Return from interrupt

Multiple Interrupts

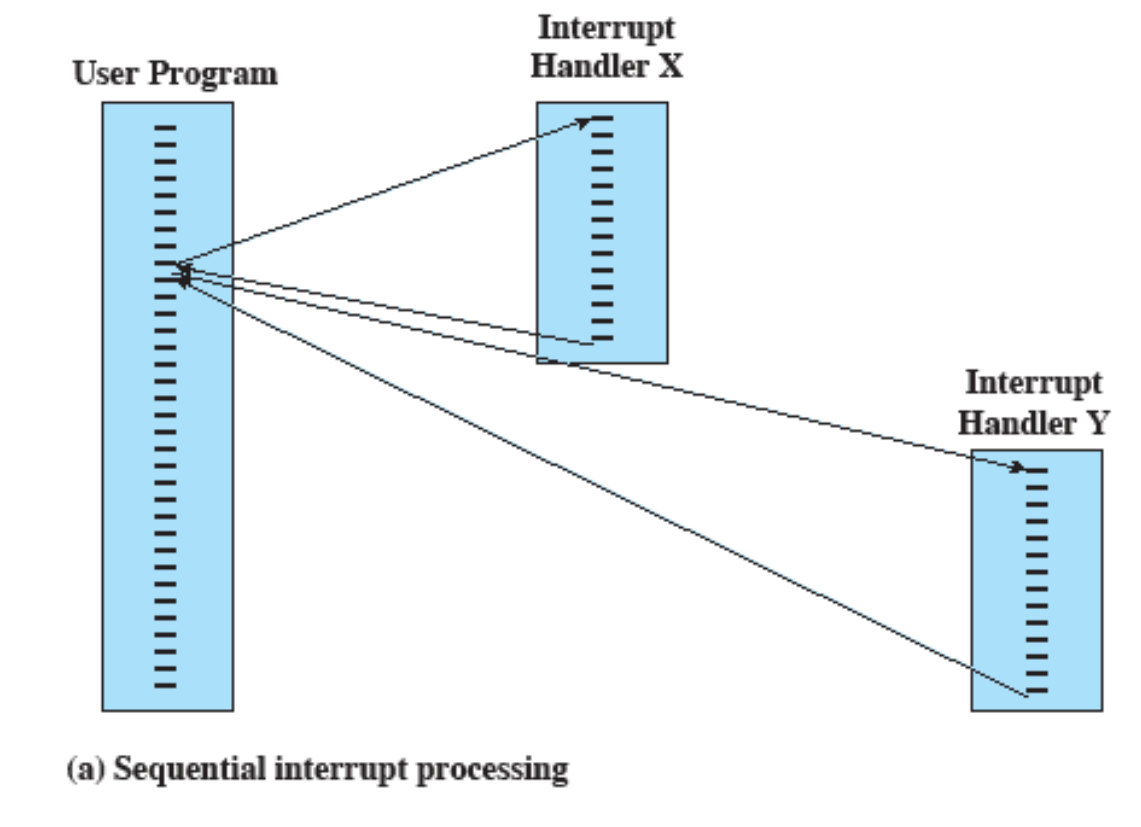
An interrupt occurs while another interrupt is being processed

- 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 (sequential)
- use a priority scheme (nested)

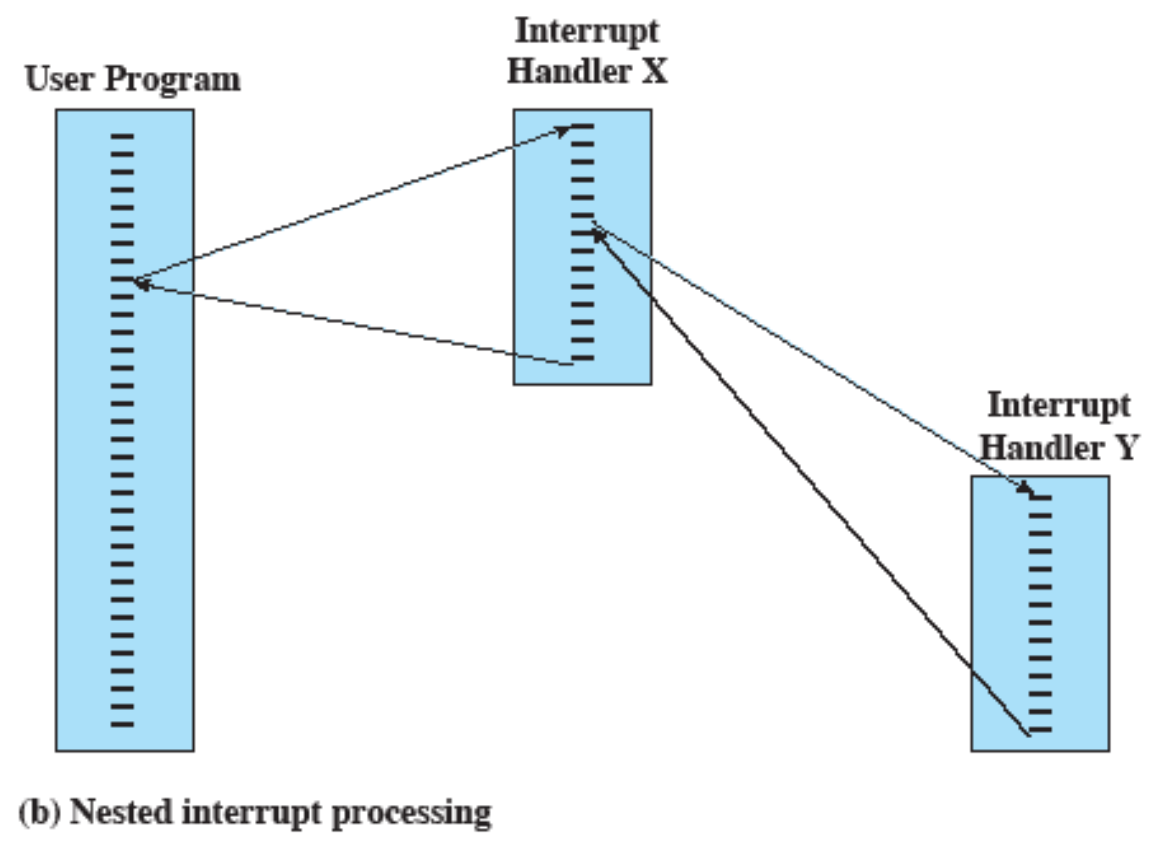
Transfer of Control With Multiple Interrupts:



(a) Sequential interrupt processing

Sequential

Transfer of Control With Multiple Interrupts:



Nested

Example Time Sequence of Multiple Interrupts

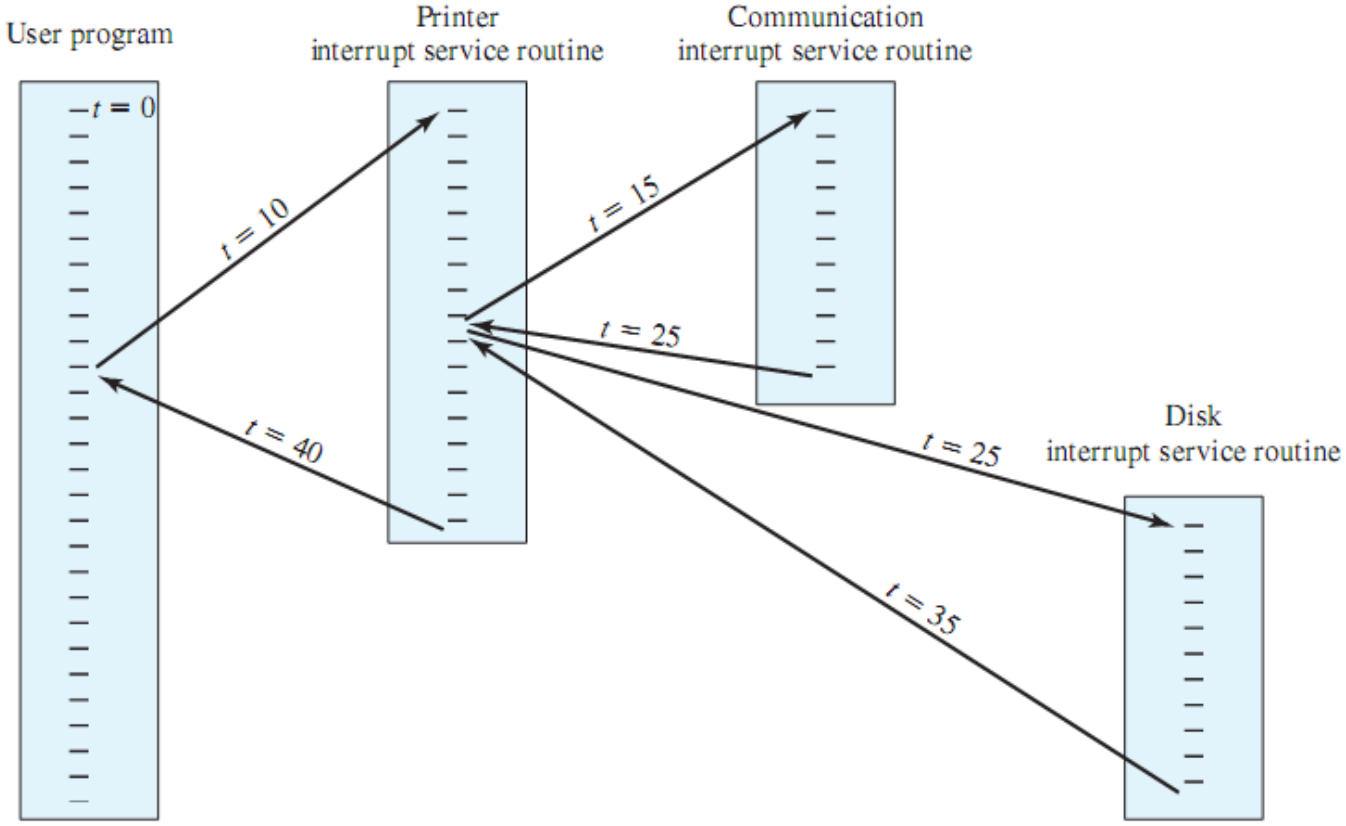
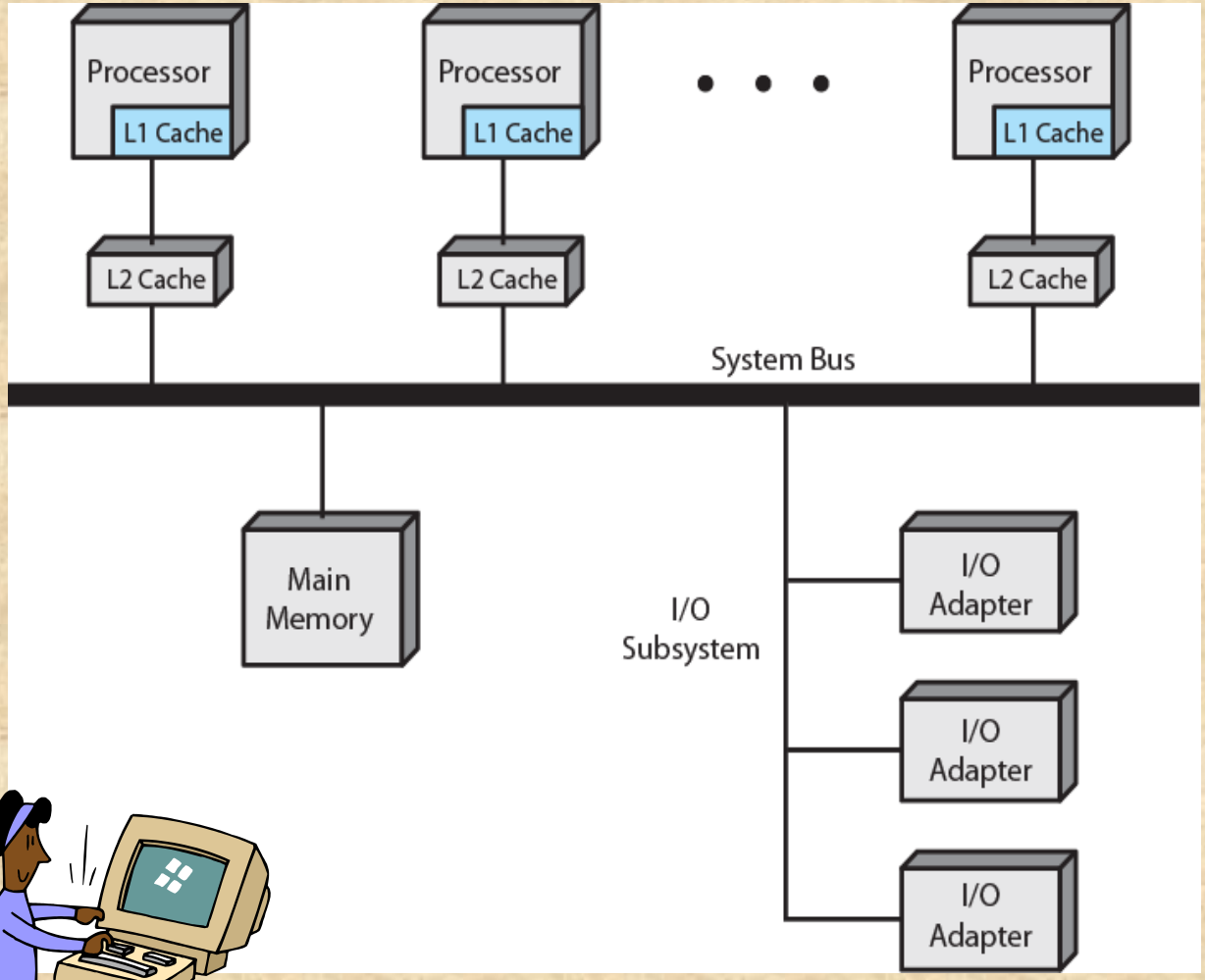


Figure 1.13 Example Time Sequence of Multiple Interrupts

Symmetric Multiprocessors (SMP)

- A stand-alone computer system with the following characteristics:
  - two or more similar processors of comparable capability
  - processors share the same main memory and are interconnected by a bus or other internal connection scheme
  - processors share access to I/O devices
  - all processors can perform the same functions
  - the system is controlled by an integrated operating system that provides interaction between processors and their programs at the job, task, file, and data element levels

SMP Organization



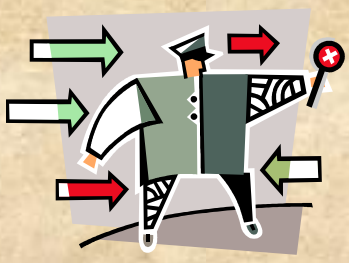
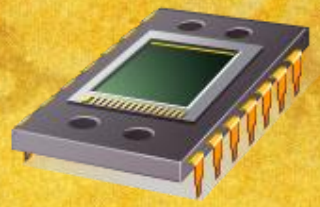


Figure 1.19 Symmetric Multiprocessor Organization



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

Intel Core i7

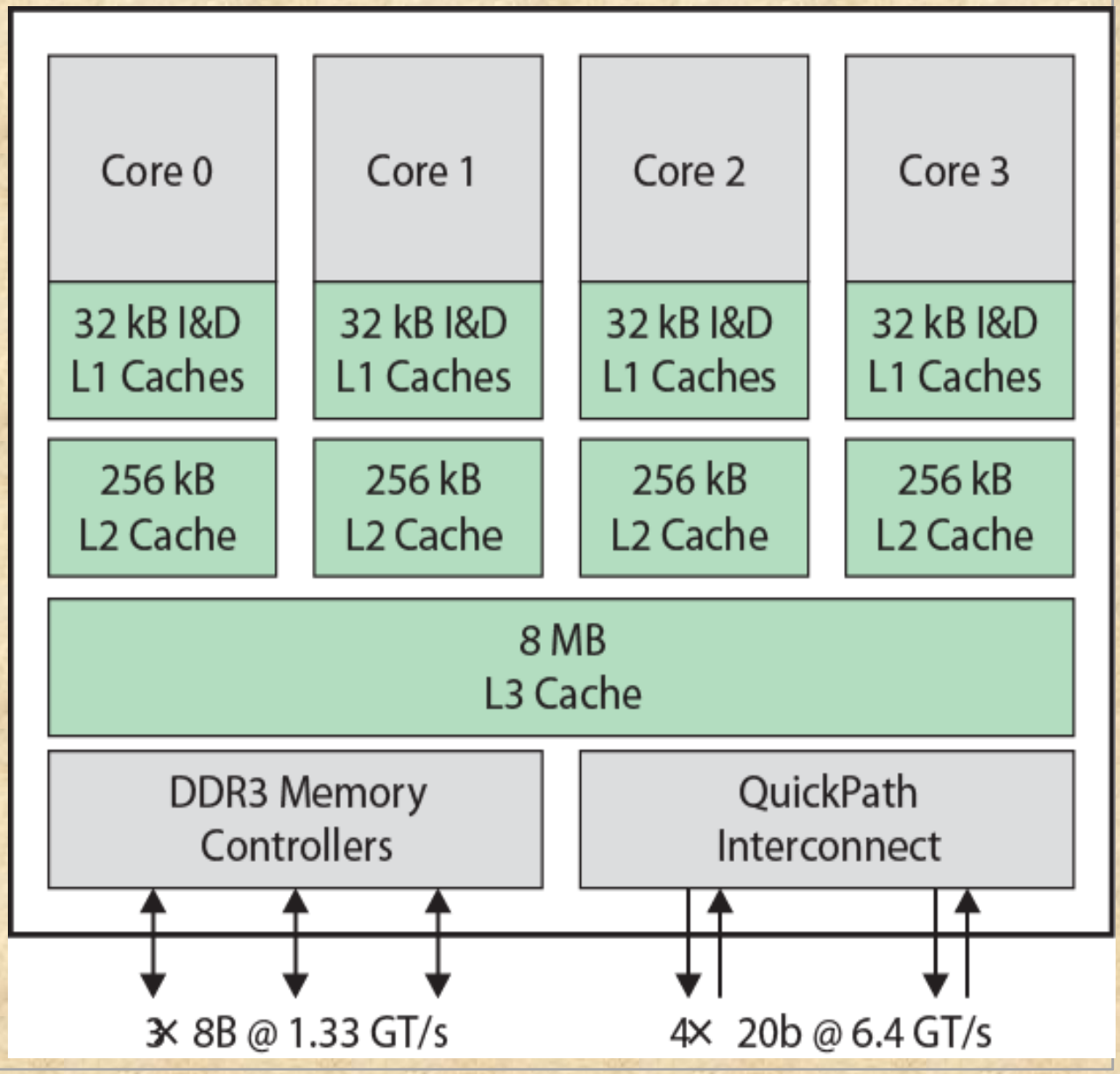


Figure 1.20 Intel Corei7 Block Diagram

Summary

- Basic Elements
  - processor, main memory, I/O modules, system bus
  - GPUs, SIMD, DSPs, SoC
  - Instruction execution
    - processor-memory, processor-I/O, data processing, control
  - Interrupt/Interrupt Processing
  - Memory Hierarchy
  - Cache/cache principles and designs
  - Multiprocessor/multicore