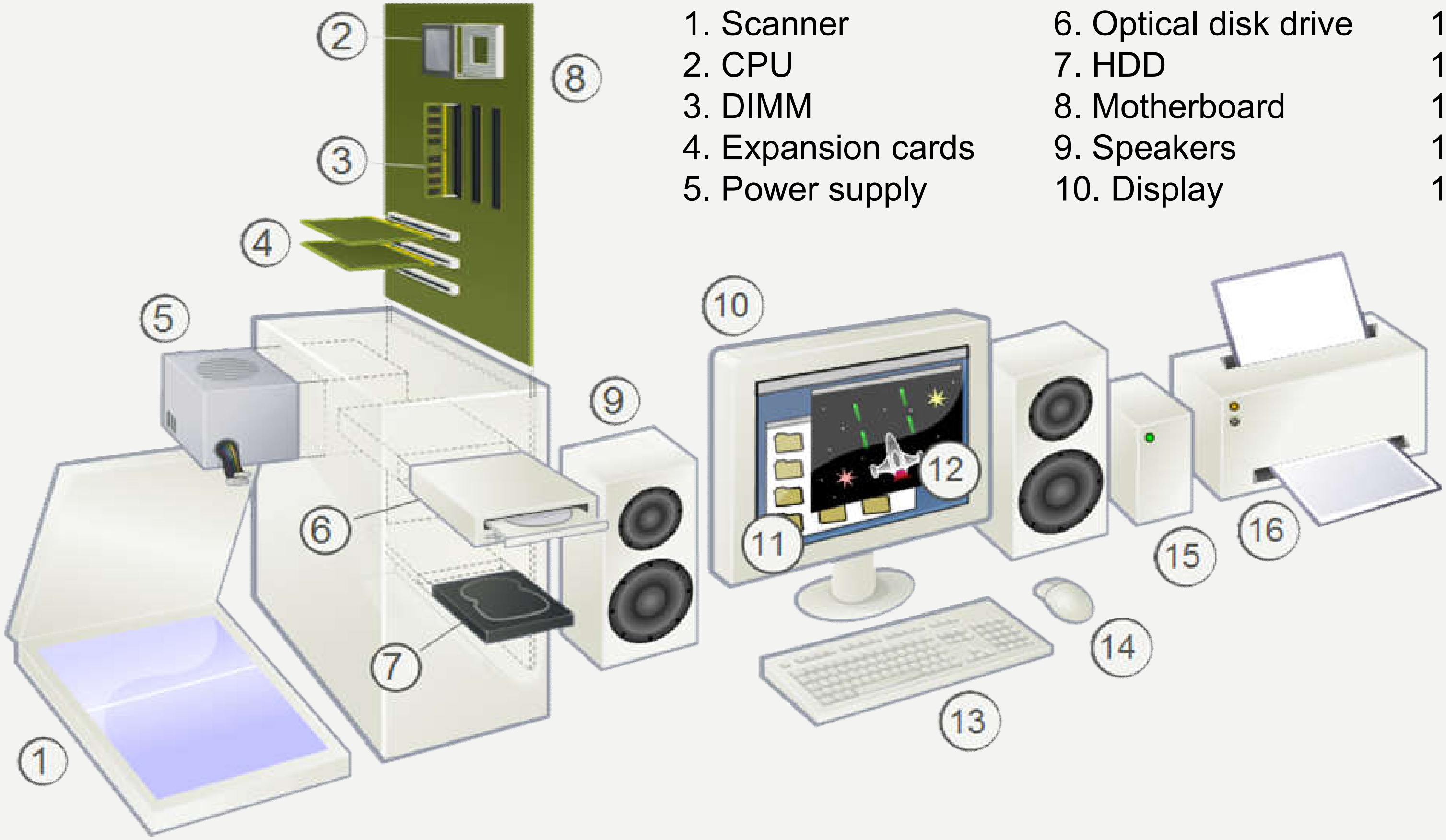
COMPUTER BASICS

Module 1.1
COP4600 – Operating Systems
Richard Newman

WHAT IS A COMPUTER?

- Stored program concept
- Components
  - CPU(s) a.k.a. cores
  - RAM
  - I/O Devices – keyboard, display, mouse, joystick, game controller, printer, NIC, …
  - Persistent Storage – disk, floppy, CD/DVD, flash drive, drum, tape …
  - Bus
  - Power Supply
- Fetch-decode-execute cycle

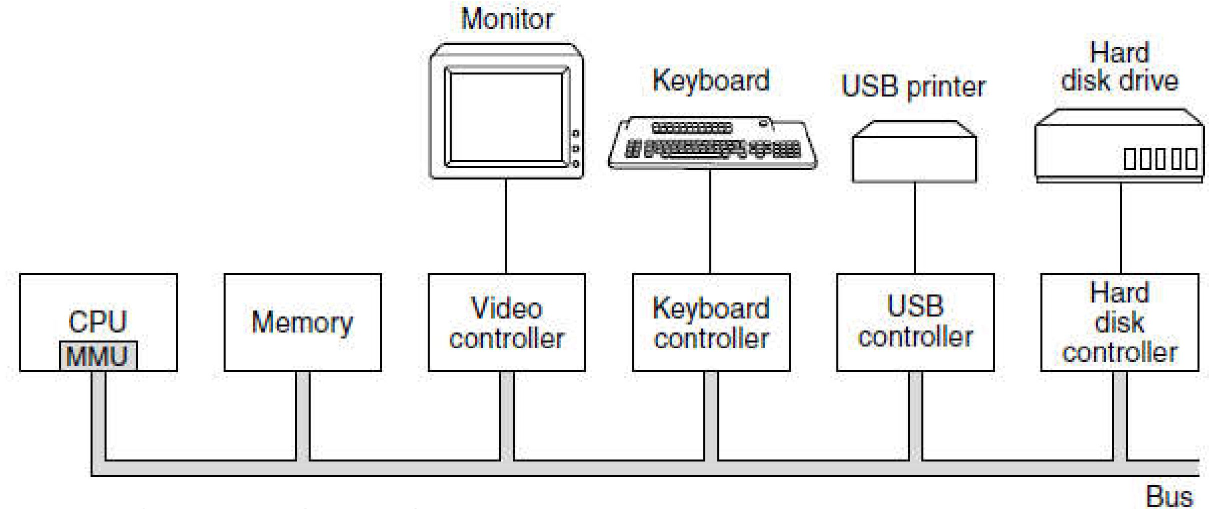
WHAT IS A COMPUTER?



1. Scanner
2. CPU
3. DIMM
4. Expansion cards
5. Power supply
6. Optical disk drive
7. HDD
8. Motherboard
9. Speakers
10. Display

11&12. GUI
13. Keyboard
14. Mouse
15. UPS
16. Printer

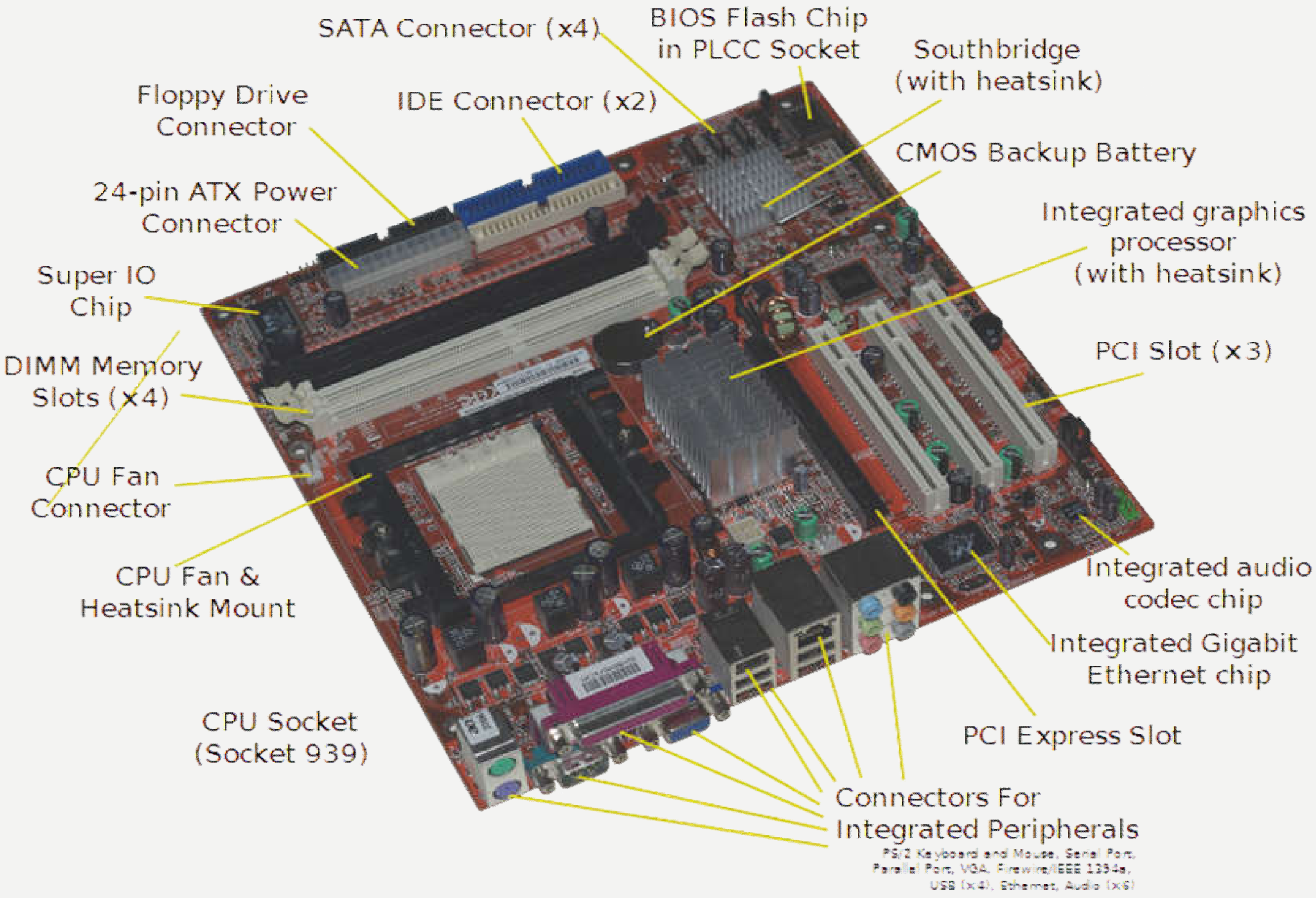
GENERIC PERSONAL COMPUTER



Tanenbaum & Bo, Modern Operating Systems:4th ed., (c) 2013 Prentice-Hall, Inc. All rights reserved.

Figure 1-6. Some of the components of a simple personal computer.

MOTHERBOARD

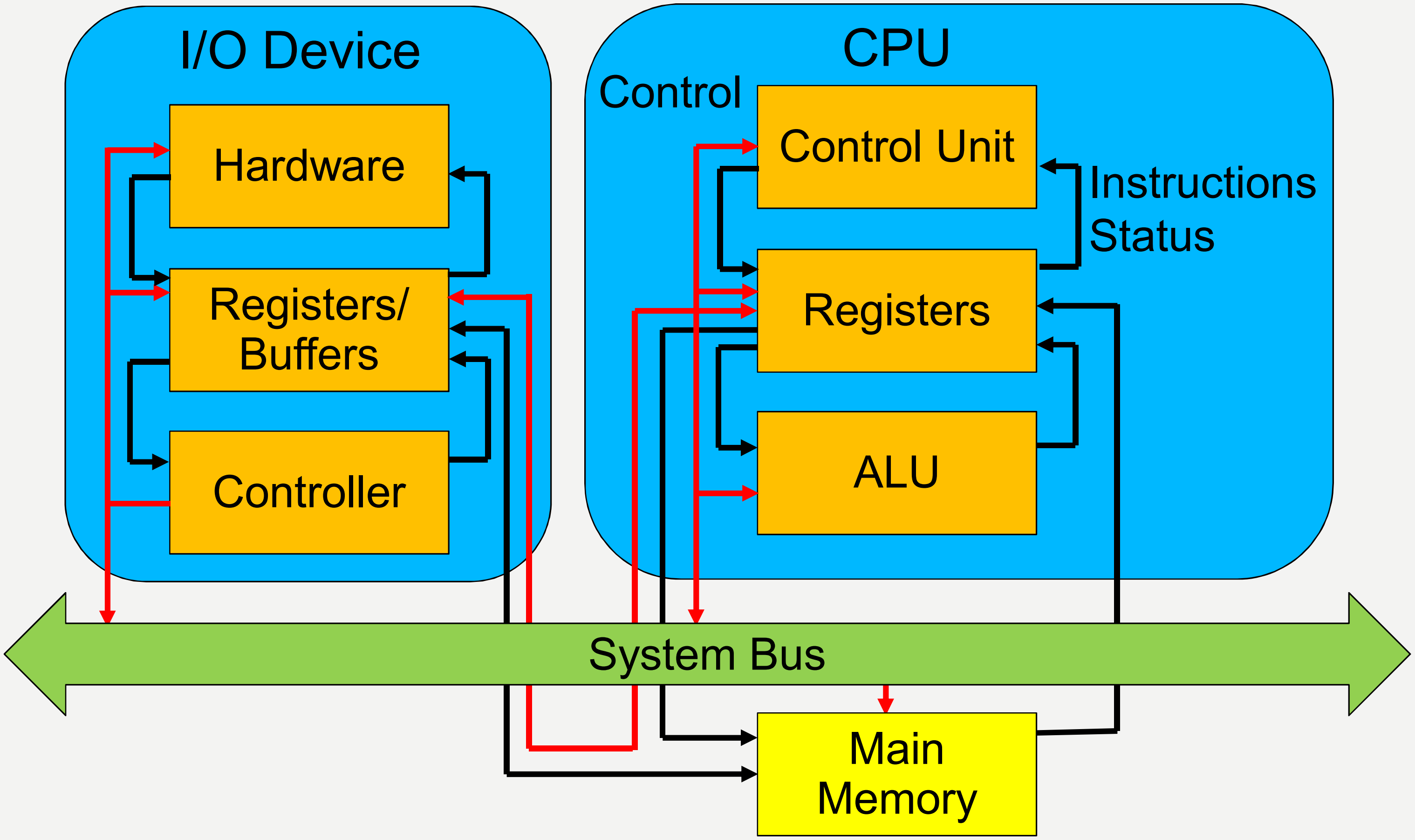

SATA Connector (x4)
BIOS Flash Chip in PLCC Socket
Southbridge (with heatsink)
Floppy Drive Connector
IDE Connector (x2)
CMOS Backup Battery
24-pin ATX Power Connector
Integrated graphics processor (with heatsink)
Super IO Chip
DIMM Memory Slots (x4)
PCI Slot (x3)
CPU Fan Connector
CPU Fan & Heatsink Mount
Integrated audio codec chip
Integrated Gigabit Ethernet chip
CPU Socket (Socket 939)
PCI Express Slot
Connectors For Integrated Peripherals
PS/2 Keyboard and Mouse, Serial Port, Parallel Port, VGA, Firewire/IEEE 1394a, USB (x4), Ethernet, Audio (x6)

INSIDE THE CPU

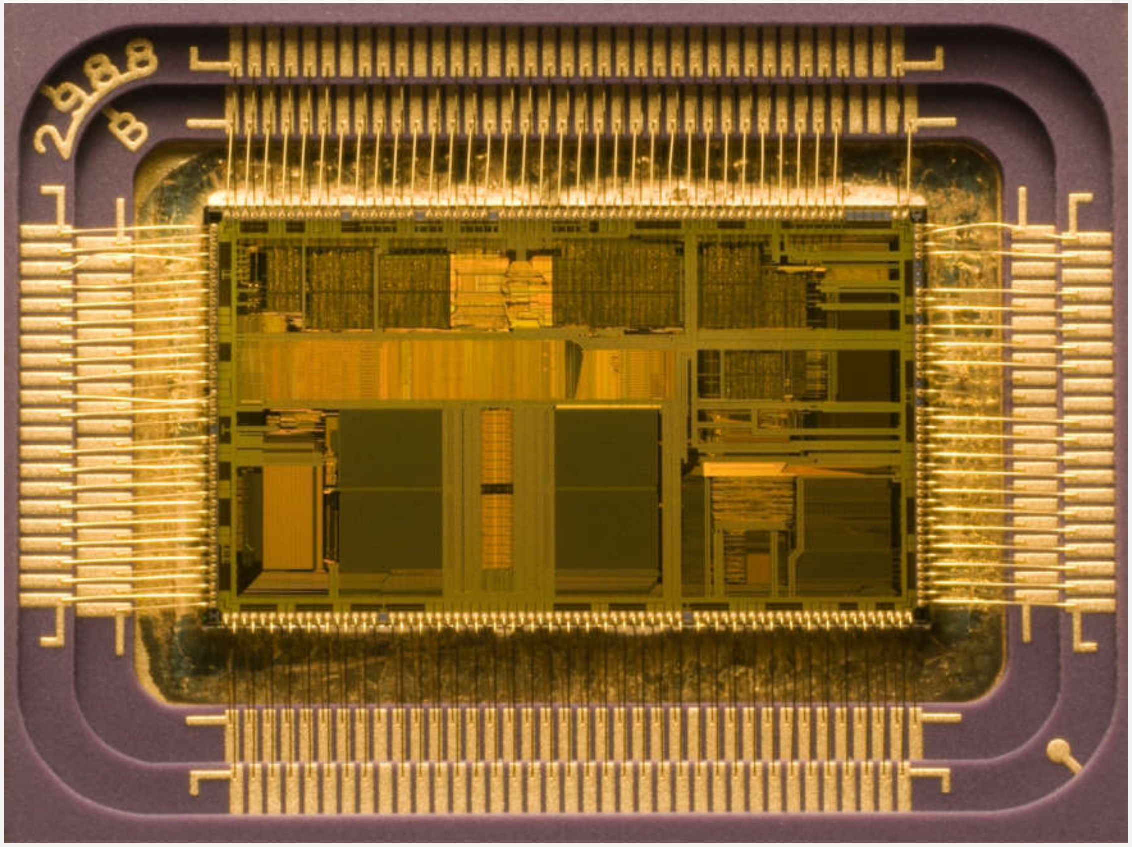
- Central Processing Unit (CPU)
- Registers – very small, very fast storage
  - Special purpose – MAR, MBR, PC, IR, SR
  - General purpose – two or more for data
- Arithmetic Logic Unit (ALU)
- Decoder – interprets instruction opcode

- Bus(es) – move data around
  - Data – read from RAM or written to RAM
  - Address – where in RAM to access
  - Control
    - Enable (for read)
    - Set (for write)

INSIDE THE CPU



INSIDE THE CPU



Intel 80486 die

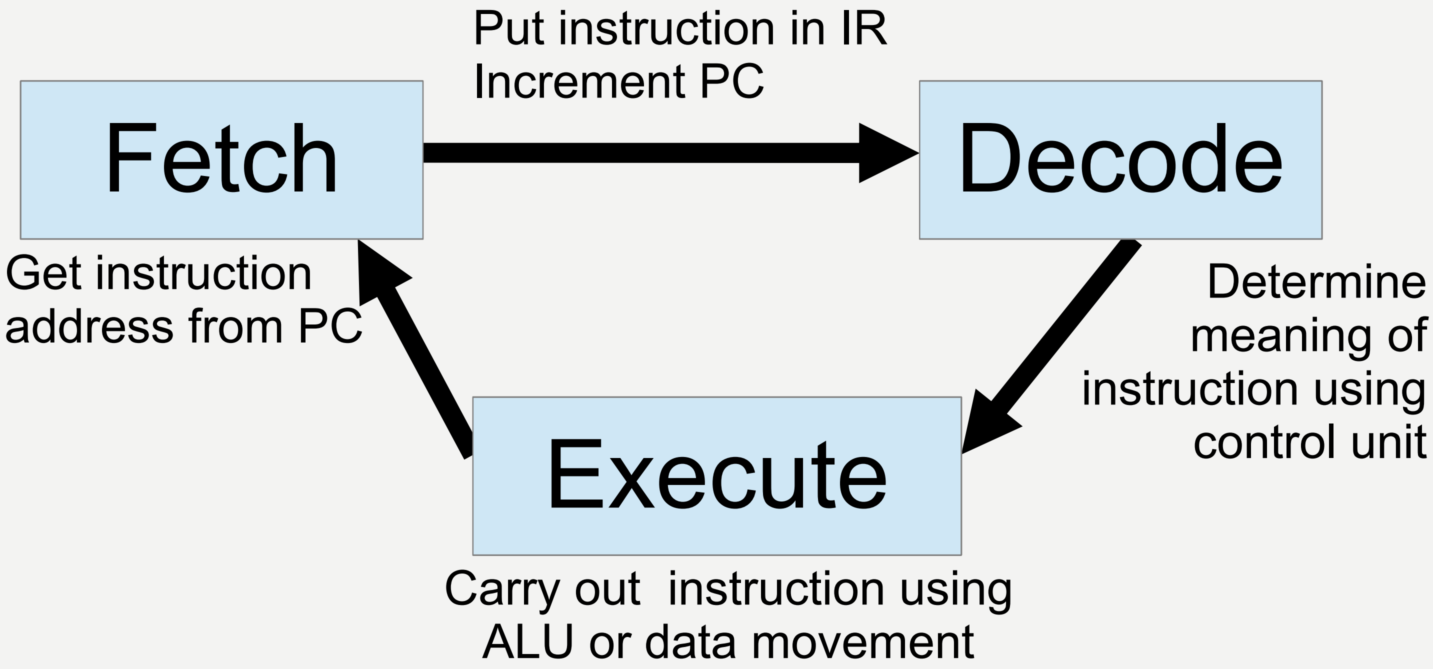
HOW DOES A COMPUTER WORK?

- Fetch-decode-execute cycle
- Fetch
  - Load MAR from PC
  - Read instruction from RAM into MBR
  - Move instruction from MBR to IR
  - Increment PC
- Decode
  - Interpret operation code (opcode) in IR
- Execute
  - Carry out instruction – get additional parts of multi-word instruction, move data to data registers, cause ALU to perform operation on data registers, test status register, change the PC, etc.

INSTRUCTIONS AND PROTECTION STATE

- Opcode and operands
  - Opcode determines what the instruction does
  - Operands determine what it does it to
- Extended instructions
  - Operands may be too large to fit into a single memory word
  - Instruction may require additional fetches
- Protected instructions
  - Some instructions are not available to user programs
  - Processor protection state must be compatible
- Protection state
  - Special register that hold current permission info
  - Changed when particular instructions are executed

FETCH-DECODE-EXECUTE CYCLE



Fetch

Get instruction
address from PC

Put instruction in IR
Increment PC

Decode

Determine
meaning of
instruction using
control unit

Execute

Carry out instruction using
ALU or data movement

SIMPLE PIPELINING

Fetch I-1	Fetch I-2	Fetch I-3	F I-4
	Decode I-1	Decode I-2	D I-3
		Execute I-1	E I-2
Get instruction address from PC Load MBR	Move instruction to IR While I-1 is decoded Load MBR with I-2	Execute I-1 in ALU or buses Move I-2 to IR and decode Load MBR with I-3	Etc. Etc. Etc.

PARALLELISM IN PROCESSORS

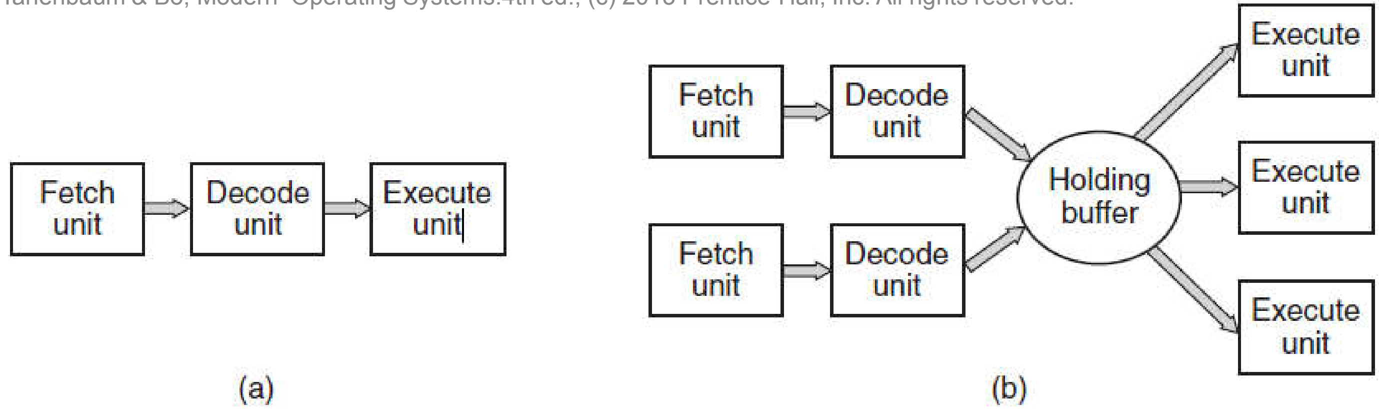


Figure 1-7. (a) A three-stage pipeline.
(b) A superscalar CPU.

MEMORY

Tanenbaum & Bo, Modern Operating Systems:4th ed., (c) 2013 Prentice-Hall, Inc. All rights reserved.

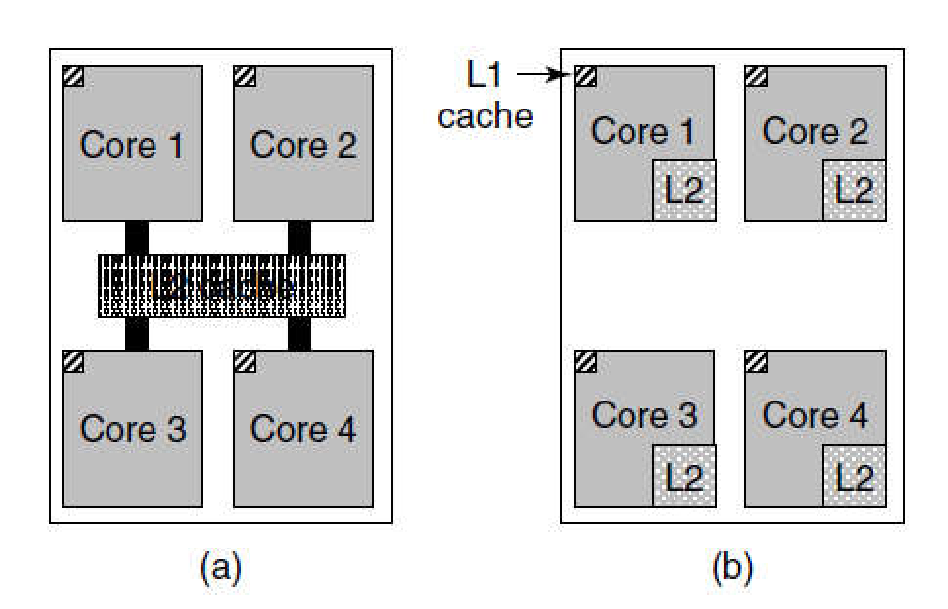


Figure 1-8. (a) A quad-core chip with a shared L2 cache.
(b) A quad-core chip with separate L2 caches.

MEMORY HIERARCHY

Tanenbaum & Bo, Modern Operating Systems:4th ed., (c) 2013 Prentice-Hall, Inc. All rights reserved.

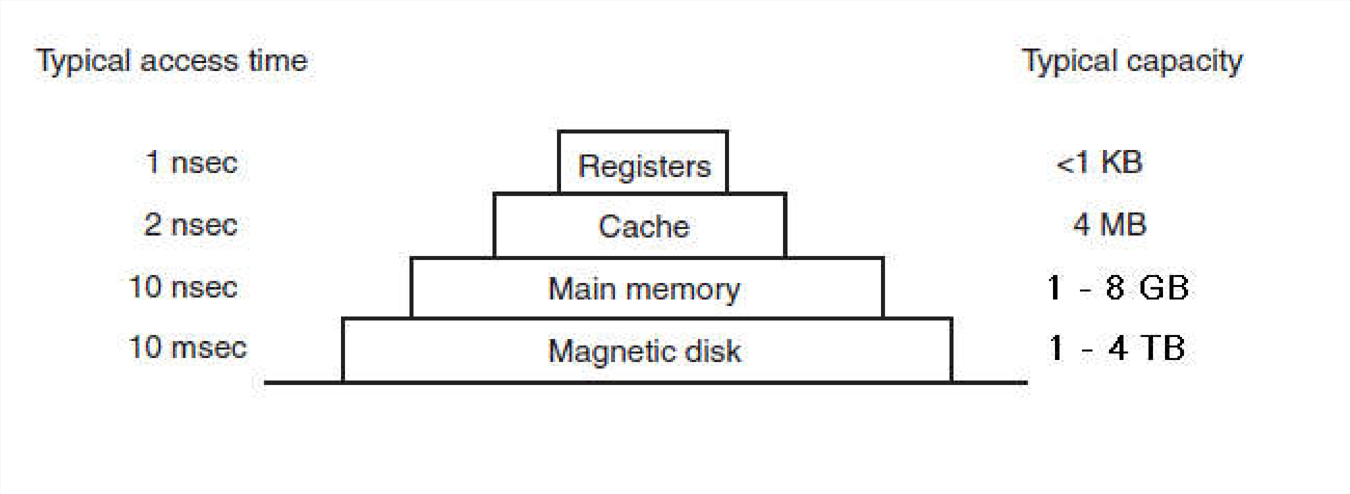


Figure 1-9. A typical memory hierarchy. The numbers are very rough approximations.

CACHING

Caching system issues:

- When to put a new item into the cache.
- How big a cache line should be.
- Which cache line to put the new item in.
- Which item to remove from the cache when a slot is needed.
- Where to put a newly evicted item in the larger memory.

BUSES

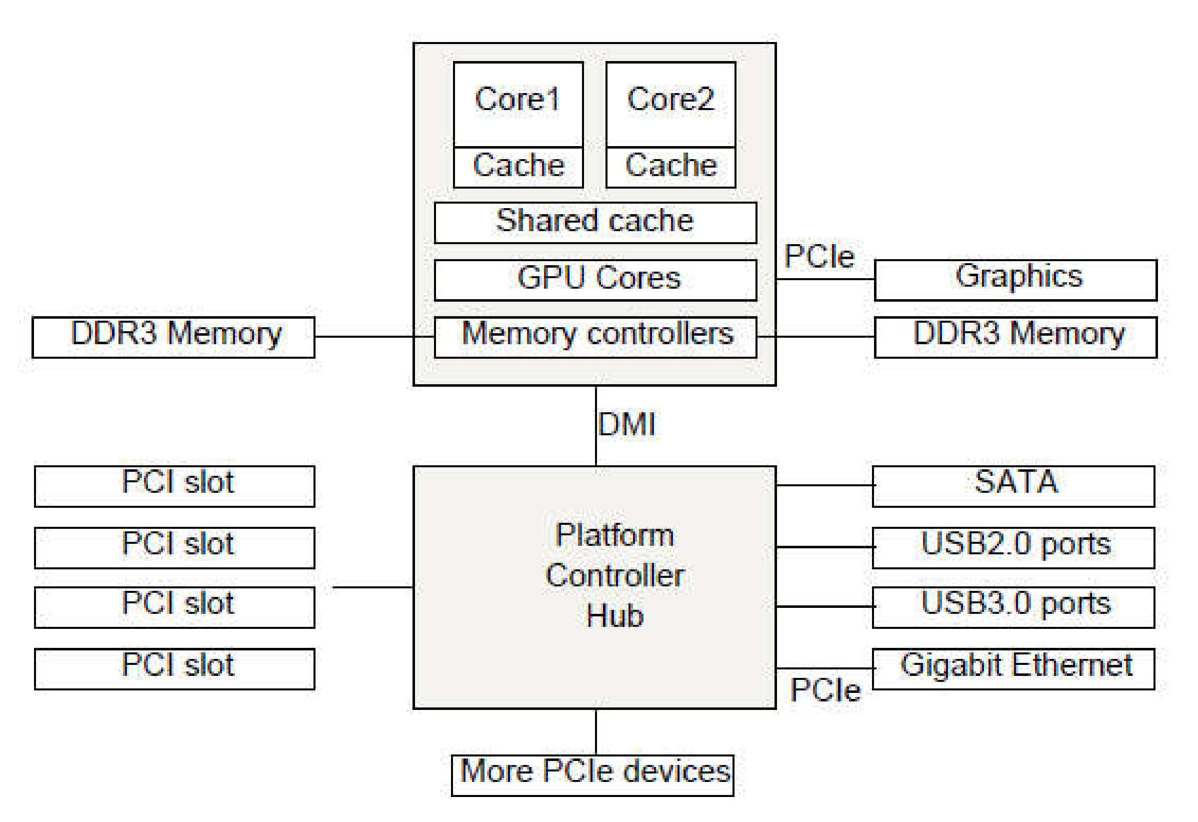


Figure 1-12. The structure of a large x86 system

HOW DOES A BUS WORK?

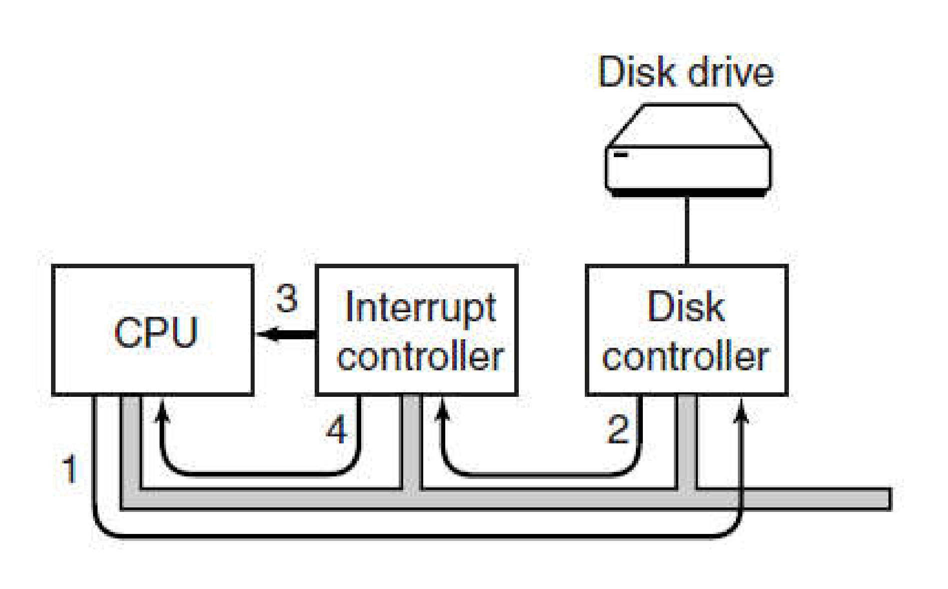
- Three types of bus lines
  - Data
  - Address
  - Control
- Bus Controller/Arbiter
  - Decides which competing device gets the bus

- Address/Data lines
  - Specify data and memory address to access
  - May reuse the same lines (different times)
- Control lines
  - Power, ground, clock
  - Read/Write (enable/set)
  - Interrupt line(s)

HOW MANY BUSES ARE THERE?

- Early computers had just two
  - One on CPU for routing between registers, ALU, CU, etc.
  - One “system bus” for access to memory, I/O devices
- Later, many evolved
  - Buses run synchronously at bus clock speed
  - CPU and memory speeds increased faster than I/O device speeds – needed separate bus to keep CPU fed
  - I/O devices may also run at different speeds
  - So multiple I/O buses have arisen to deal with this
- Multiple buses meant I/O controller /hub was needed
  - Southbridge/Northbridge

I/O DEVICES



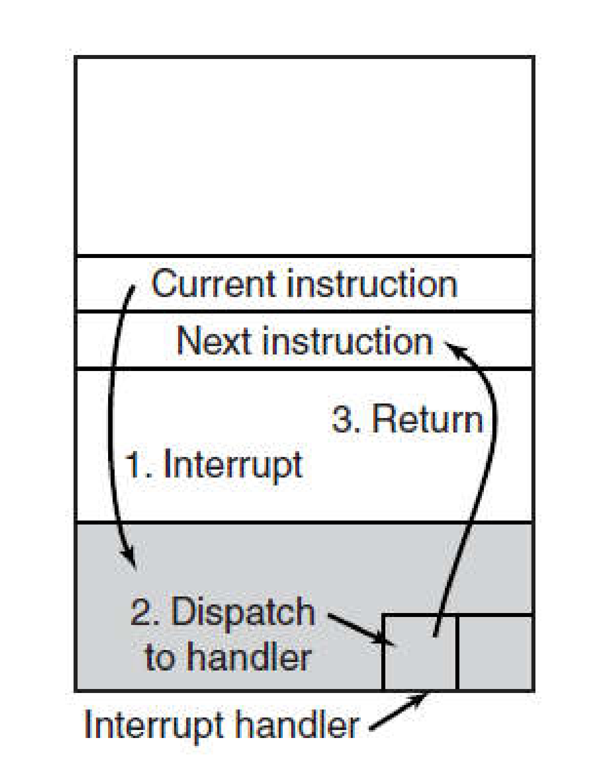


Figure 1-11. (a) The steps in starting an I/O device and getting an interrupt. (b) Interrupt processing involves taking the interrupt, running the interrupt handler, and returning to the user program.

DISKS

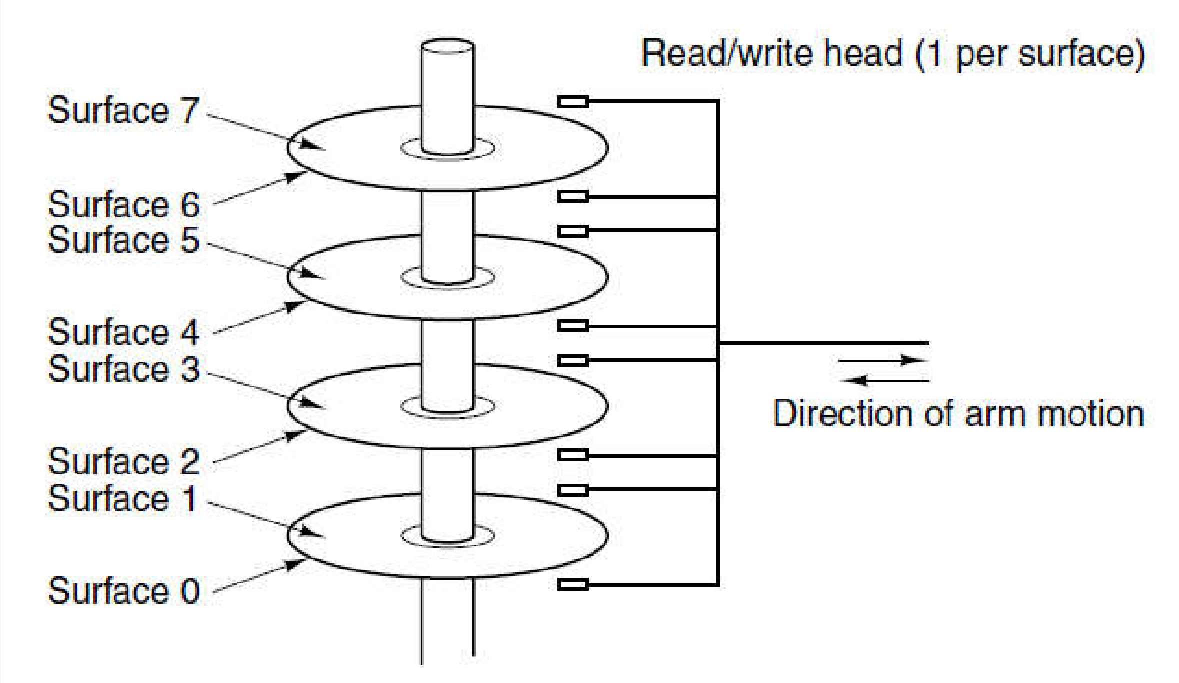
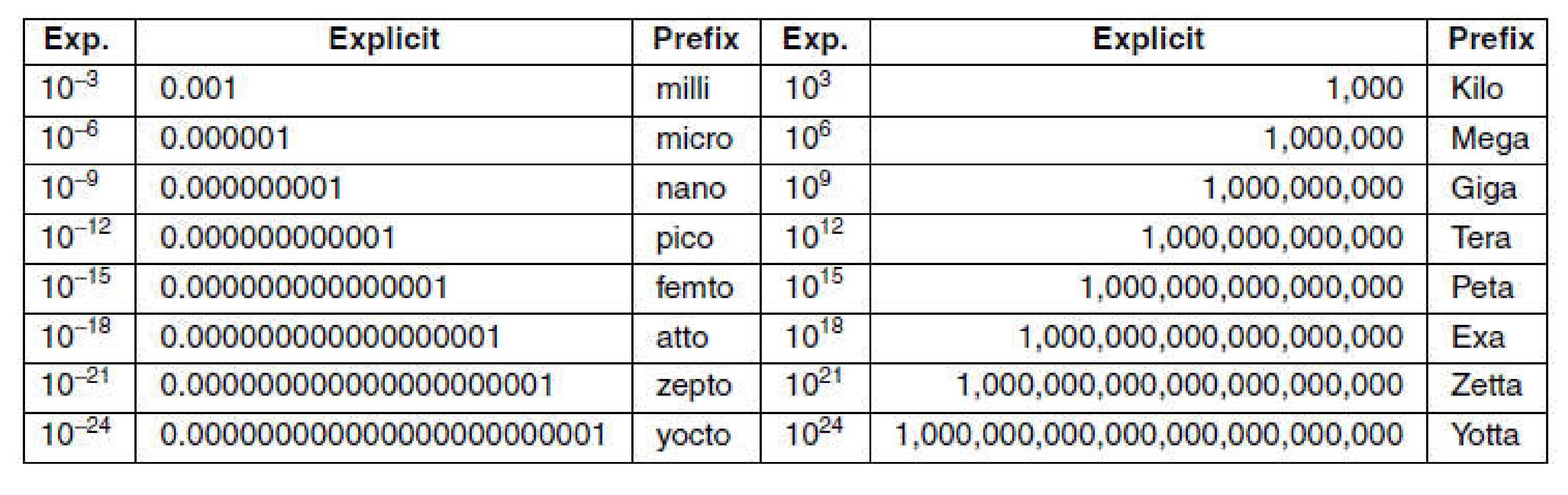


Figure 1-10. Structure of a disk drive.

Tanenbaum & Bo, Modern Operating Systems:4th ed., (c) 2013 Prentice-Hall, Inc. All rights reserved.

METRIC UNITS



Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
10^{-3}	0.001	milli	10^{3}	1,000	Kilo
10^{-6}	0.000001	micro	10^{6}	1,000,000	Mega
10^{-9}	0.000000001	nano	10^{9}	1,000,000,000	Giga
10^{-12}	0.000000000001	pico	10^{12}	1,000,000,000,000	Tera
10^{-15}	0.000000000000001	femto	10^{15}	1,000,000,000,000,000	Peta
10^{-18}	0.000000000000000001	atto	10^{18}	1,000,000,000,000,000,000	Exa
10^{-21}	0.000000000000000000001	zepto	10^{21}	1,000,000,000,000,000,000,000	Zetta
10^{-24}	0.000000000000000000000001	yocto	10^{24}	1,000,000,000,000,000,000,000,000	Yotta

Figure 1-31. The principal metric prefixes.

Decimal is used for clock rates (GHz) and network data rate (Gbps)

MEMORY SIZE UNITS

Prefix	Abbrev.	Exponent	Explicit
kilo	K	2^{10}	1024
mega	M	2^{20}	1,048,576
giga	G	2^{30}	1,073,741,824
tera	T	2^{40}	1,099,511,627,776
peta	P	2^{50}	1,125,899,906,842,624
exa	X	2^{60}	1,152,921,504,606,846,976
zetta	Z	2^{70}	1,180,591,620,717,411,303,424
yotta	Y	2^{80}	1,208,925,819,614,629,174,706,176

Memory sizes usually expressed in powers of 2 (GB), since addressing is in binary.

Powers of two may also be used for bus speeds (GBps).

HOW DOES A COMPUTER START?

- First instruction: initialize PC to 0 (usually)
  - Done by hardware
- Execute BIOS program stored in static RAM (is not erased when power off like DRAM)
- Power-on Self Test (POST) – make sure all components are OK
- Search for bootable device (DVD, HDD, etc.)
- Load boot block into RAM
- Start execution of program in boot block
- Bootstrap program loads operating system, starts OS execution

BOOTSTRAPPING

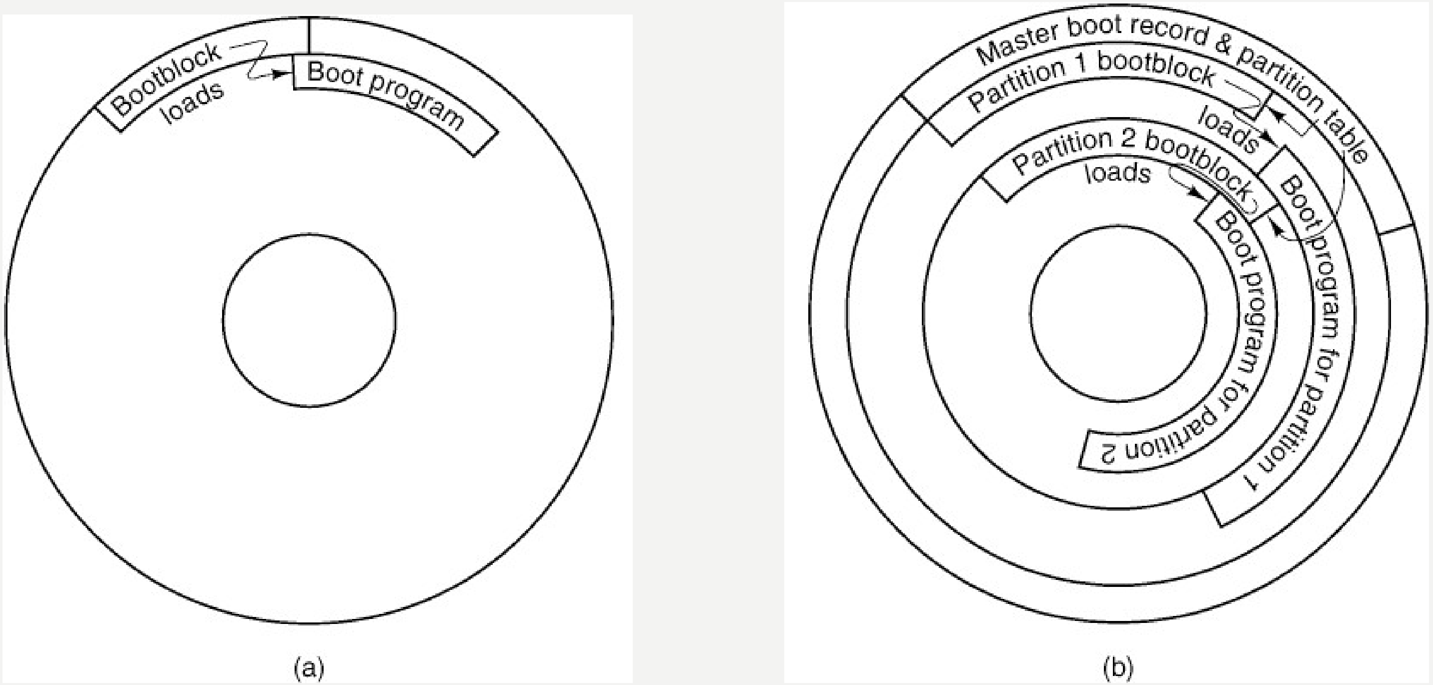

Bootblock
loads
Boot program
(a)
Master boot record & partition table
Partition 1 bootblock
loads
Boot program for partition 1
Partition 2 bootblock
loads
Boot program for partition 2
(b)

Figure 2-36. Disk structures used for bootstrapping.

(a) Unpartitioned disk. The first sector is the bootblock.

(b) Partitioned disk. The first sector is the master boot record, also called masterboot.

Tanenbaum & Woodhull, Operating Systems: Design and Implementation, (c) 2006
Prentice-Hall, Inc. All rights reserved. 0-13-142938-8

SUMMARY

- Stored Program Concept
  - Program determines functionality of machine
- Computer Organization
  - Components and how they work together
- CPU
  - Parts of the CPU
  - Fetch-Decode-Execute Cycle
  - Protected instructions
- Memory Hierarchy & Caching
  - I/O Devices & Interrupt Handling
- How a computer starts up – Bootstrapping

IMAGE SOURCES

- Exploded computer view: Self-published work by User:HereToHelp
- Motherboard: Moxfyre at en.wikipedia, CC BY-SA 3.0,
- 80486 die photo: Uberpenguin at the English language Wikipedia

LINKS

https://www.youtube.com/watch?v=XM4IGflQFvA

https://www.youtube.com/watch?v=xfJbpCJSpd8

https://quizlet.com/6585038/computer-hardware-and-ports-pictures-flash-cards/

https://www.youtube.com/watch?v=QGD4Gr4s6S8

https://commons.wikimedia.org/w/index.php?curid=6544498