Computer Systems Organization

Chapter 2

Processors

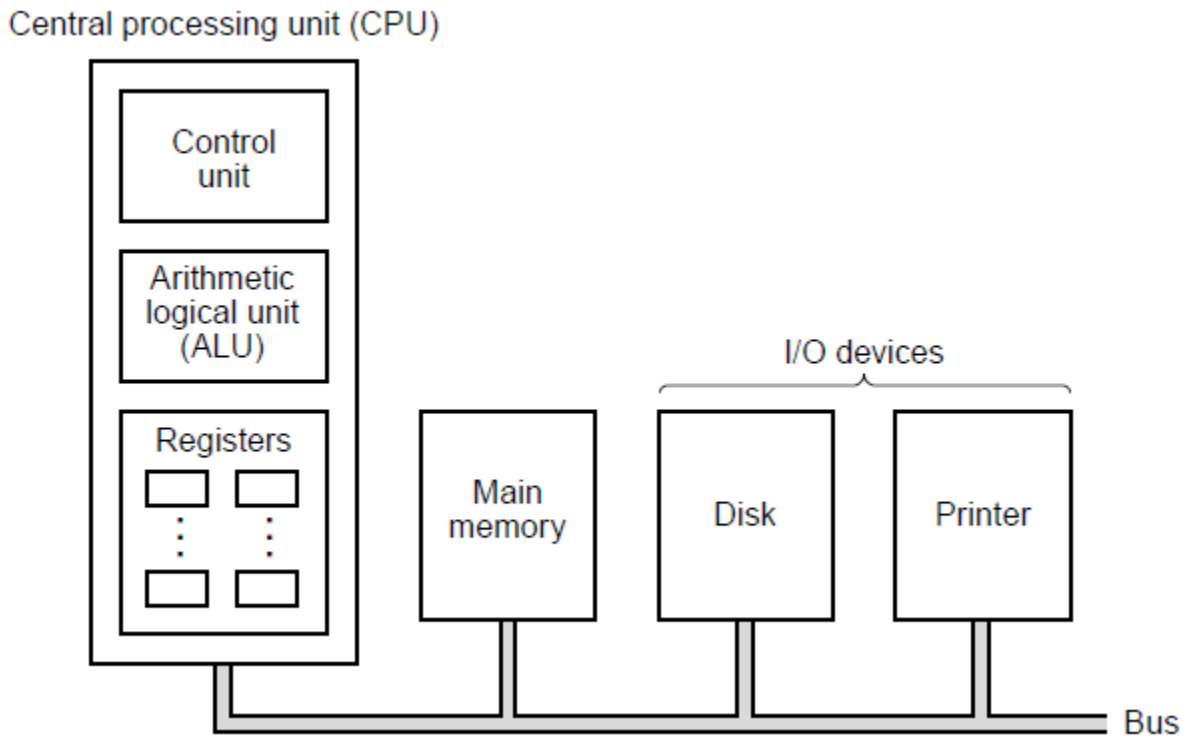

Figure 2-1. The organization of a simple computer with one CPU and two I/O devices.

CPU Organization

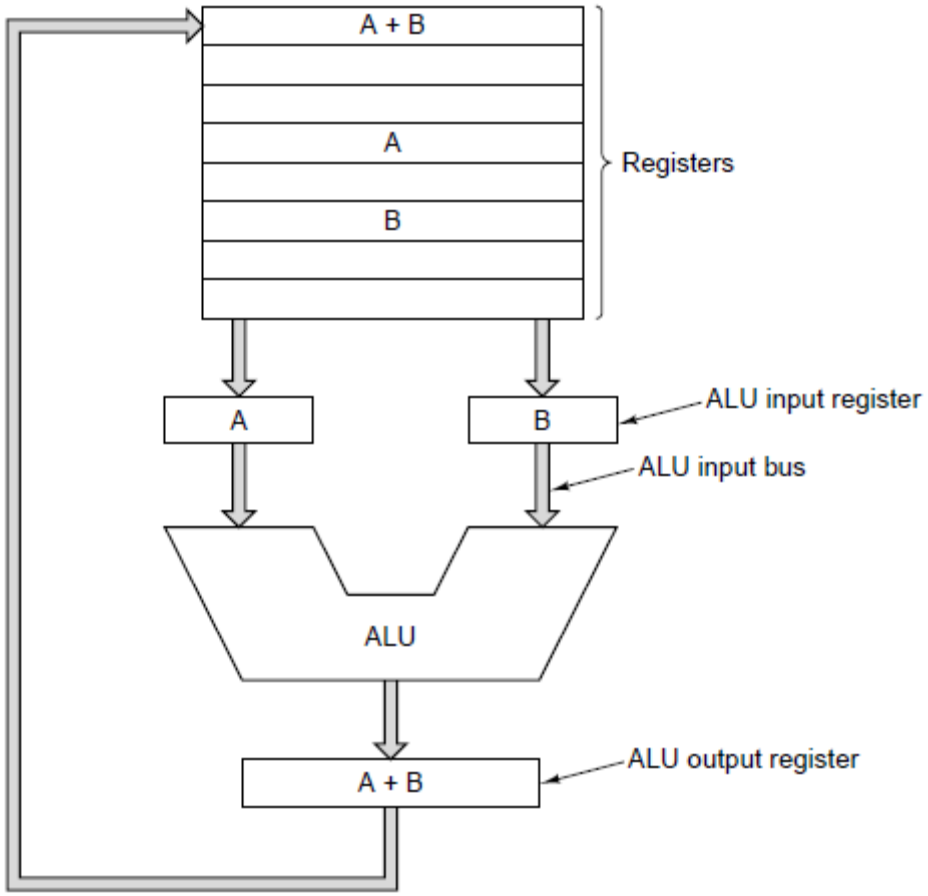

Figure 2-2. The data path of a typical von Neumann machine.

Instruction Execution (1)

- Fetch next instruction from memory
- Change program counter to point to next instruction
- Determine type of instruction just fetched
- If instruction uses a word in memory, locate it
- Fetch word, if needed, into a CPU register.
- Execute instruction.
- Go to step 1 to begin executing following instruction

Instruction Execution (3)

```
public class Interp {
   static int PC;                          // program counter holds address of next instr
   static int AC;                          // the accumulator, a register for doing arithmetic
   static int instr;                       // a holding register for the current instruction
   static int instr_type;                  // the instruction type (opcode)
   static int data_loc;                    // the address of the data, or –1 if none
   static int data;                        // holds the current operand
   static boolean run_bit = true;          // a bit that can be turned off to halt the machine

   public static void interpret(int memory[], int starting_address) {
      // This procedure interprets programs for a simple machine with instructions having
      // one memory operand. The machine has a register AC (accumulator), used for
      // arithmetic. The ADD instruction adds an integer in memory to the AC, for example.
      // The interpreter keeps running until the run bit is turned off by the HALT instruction.
      // The state of a process running on this machine consists of the memory, the
      // program counter, the run bit, and the AC. The input parameters consist of
      // the memory image and the starting address.

         . . .
```

Figure 2-3. An interpreter for a simple computer (written in Java).

Instruction Execution (4)

```
. . .

    PC = starting_address;
    while (run_bit) {
        instr = memory[PC];                    // fetch next instruction into instr
        PC = PC + 1;                           // increment program counter
        instr_type = get_instr_type(instr);    // determine instruction type
        data_loc = find_data(instr, instr_type); // locate data (–1 if none)
        if (data_loc >= 0)                     // if data_loc is –1, there is no operand
            data = memory[data_loc];           // fetch the data
        execute(instr_type, data);             // execute instruction
    }

  }

  private static int get_instr_type(int addr) { ... }
  private static int find_data(int instr, int type) { ... }
  private static void execute(int type, int data) { ... }
}
```

Figure 2-3. An interpreter for a simple computer (written in Java).

Instruction Execution (5)

Benefits of machines with interpreted instructions

- Ability to fix incorrectly implemented instructions in field, even make up for design deficiencies in basic hardware
- Opportunity to add new instructions at minimal cost, even after delivery of machine
- Structured design that permitted efficient development, testing, documenting of complex instructions

Design Principles for Modern Computers

- All instructions directly executed by hardware
- Maximize rate at which instructions are issued
- Instructions should be easy to decode
- Only loads and stores should reference memory
- Provide plenty of registers

Pipelining

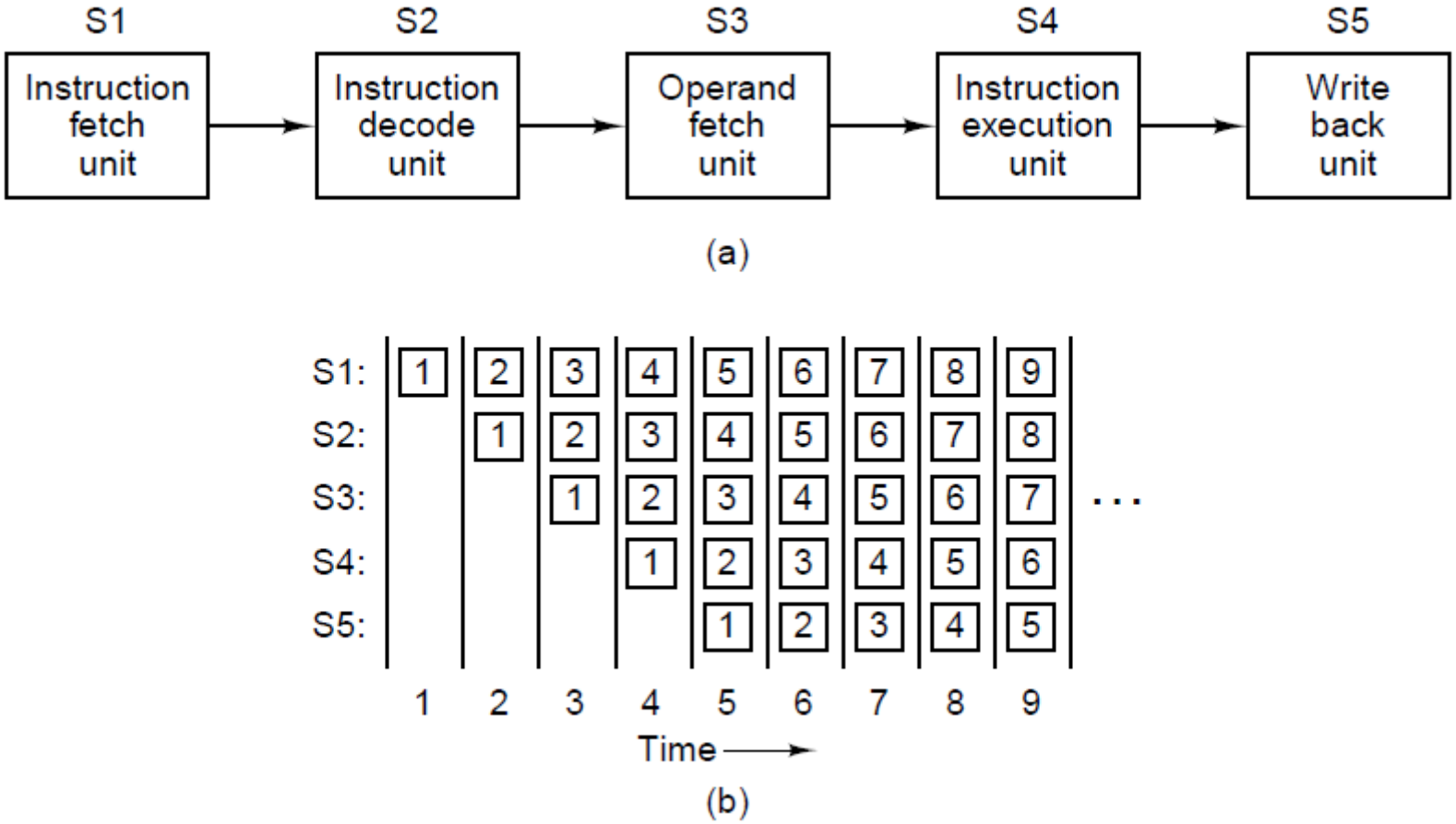

Figure 2-4. (a) A five-stage pipeline. (b) The state of each stage as a function of time. Nine clock cycles are illustrated

Superscalar Architectures (1)

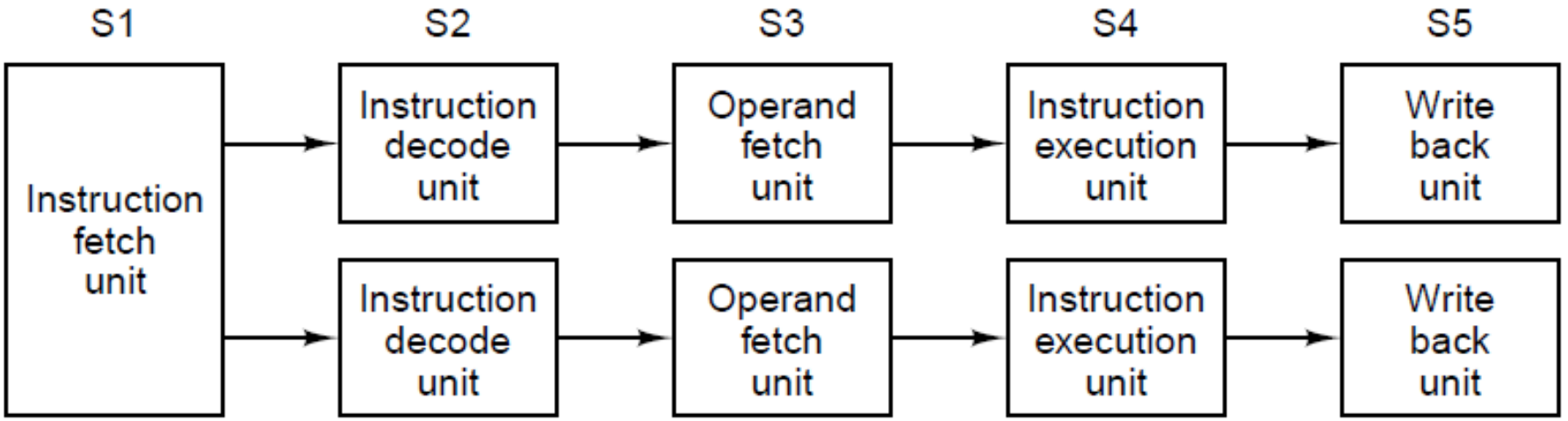

Figure 2-5. Dual five-stage pipelines with a common instruction fetch unit.

Superscalar Architectures (2)

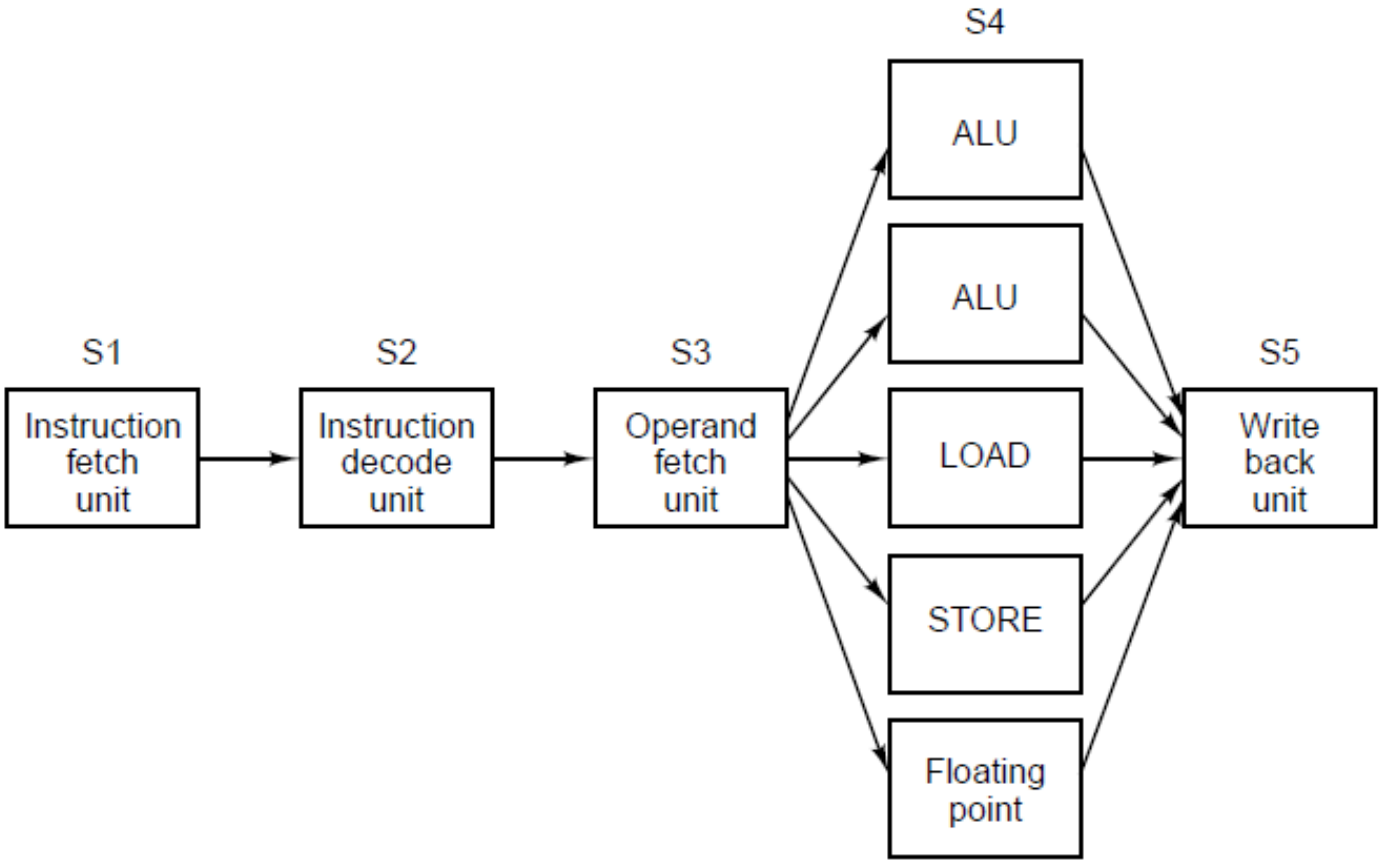

Figure 2-6. A superscalar processor with five functional units.

Data Parallel Computers

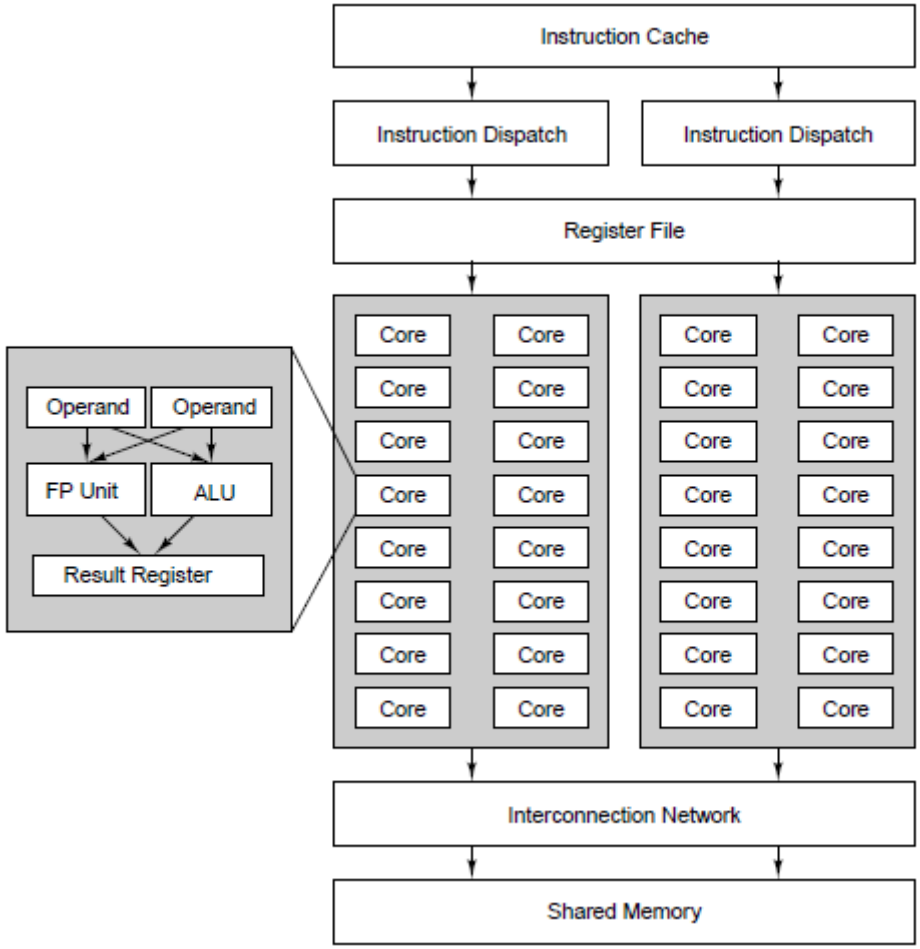

Figure 2-7. The SIMD core of the Fermi graphics processing unit.

Multiprocessors (1)

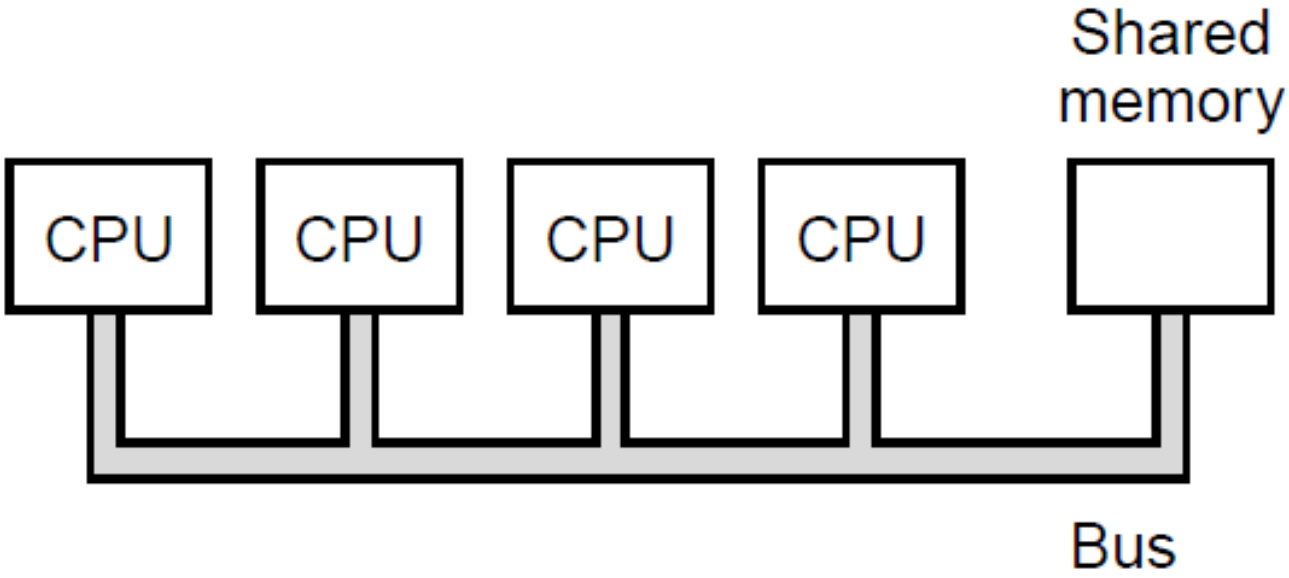

Figure 2-8. (a) A single-bus multiprocessor.

Multiprocessors (1)

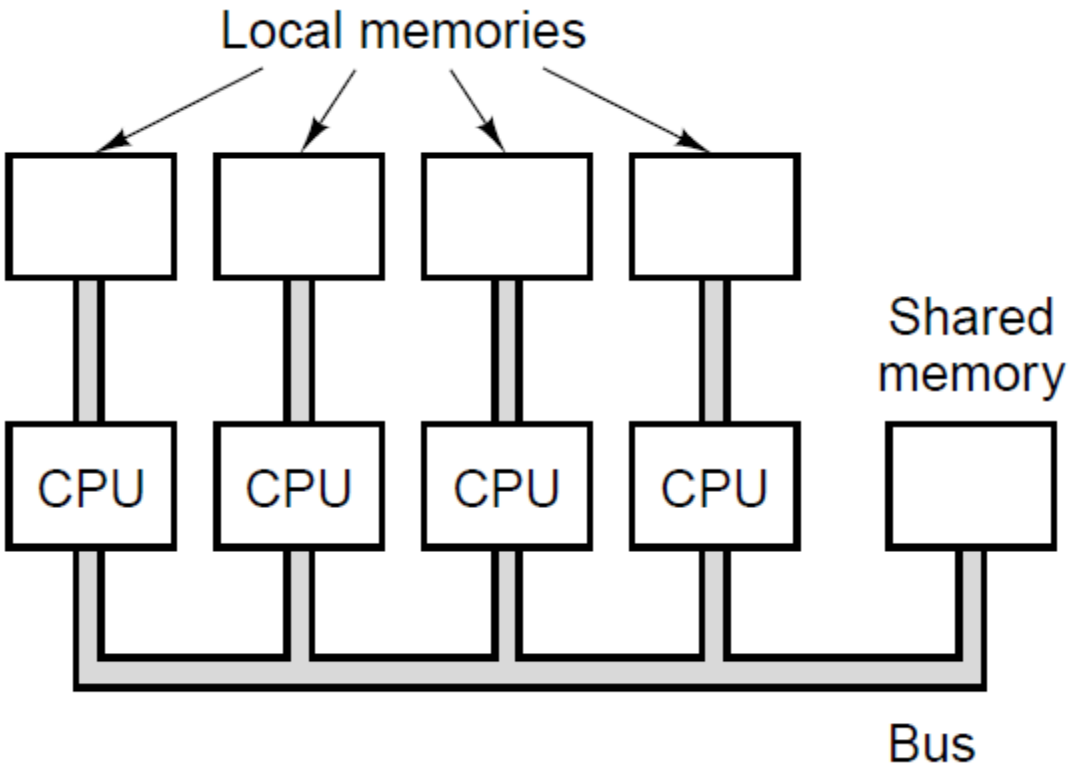

Figure 2-8(b) A multicomputer with local memories.

Primary Memory (1)

The part of computer where programs and data are stored

Bit: binary digit

Memory address: location in memory of a cell containing data

Primary Memory (2)

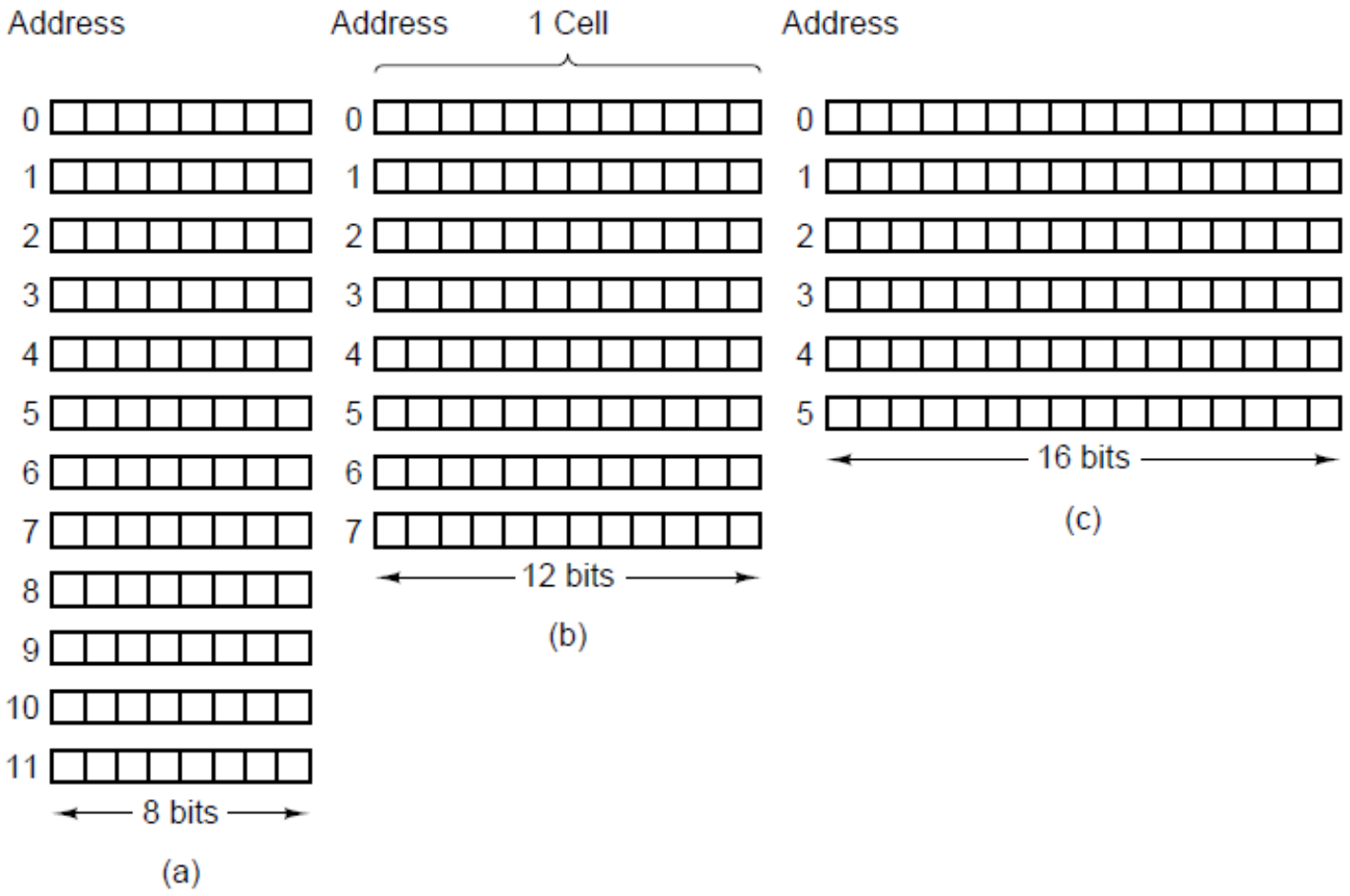

Figure 2-9. Three ways of organizing a 96-bit memory

Primary Memory (3)

Computer	Bits/cell
Burroughs B1700	1
IBM PC	8
DEC PDP-8	12
IBM 1130	16
DEC PDP-15	18
XDS 940	24
Electrologica X8	27
XDS Sigma 9	32
Honeywell 6180	36
CDC 3600	48
CDC Cyber	60

Figure 2-10. Number of bits per cell for some historically interesting commercial computers.

Byte Ordering (1)

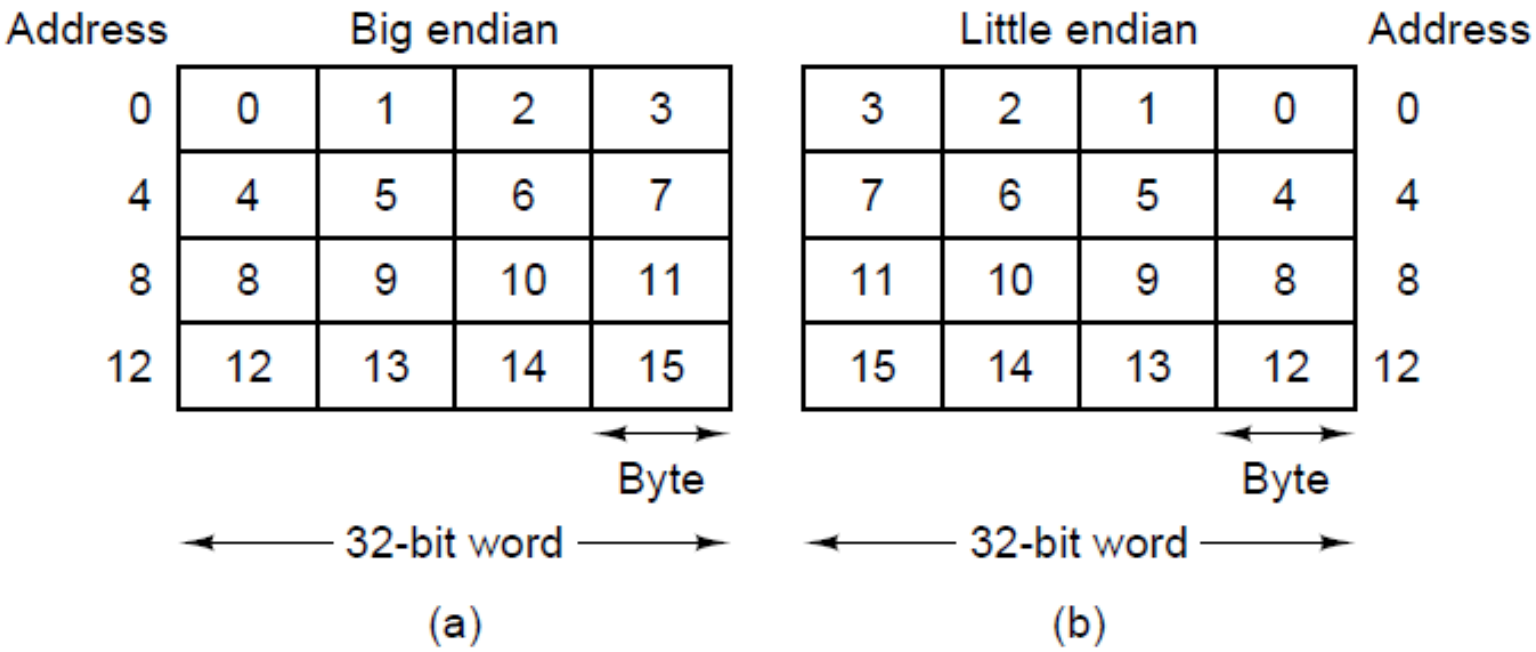

Figure 2-11. (a) Big endian memory. (b) Little endian memory.

Byte Ordering (2)

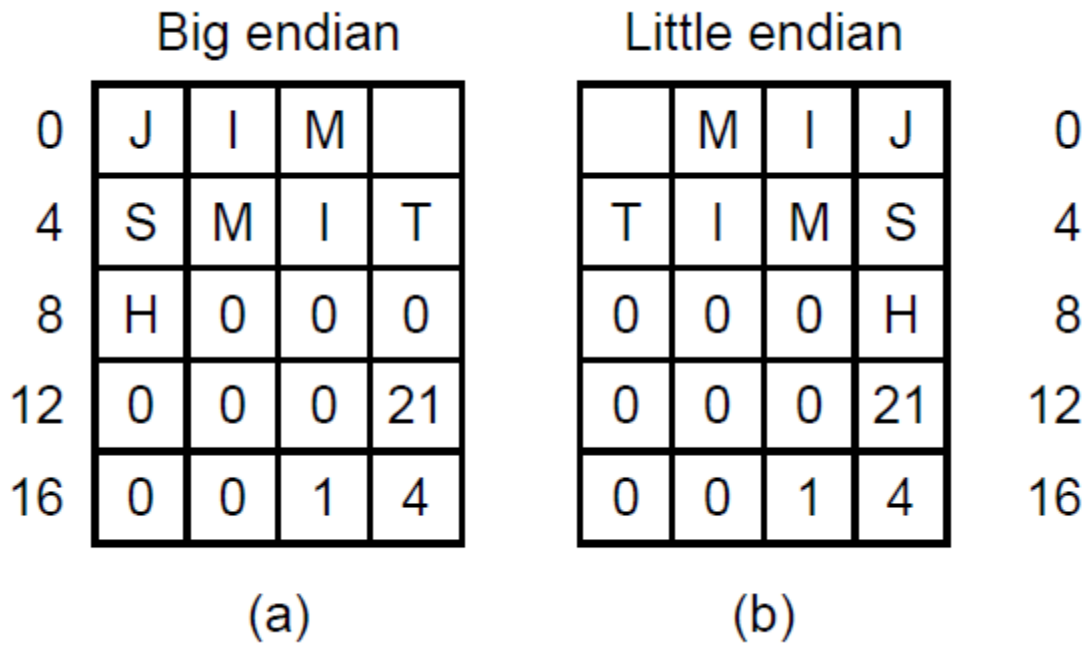

Figure 2-12. (a) A personnel record for a big endian machine. (b) The same record for a little endian machine.

Byte Ordering (3)

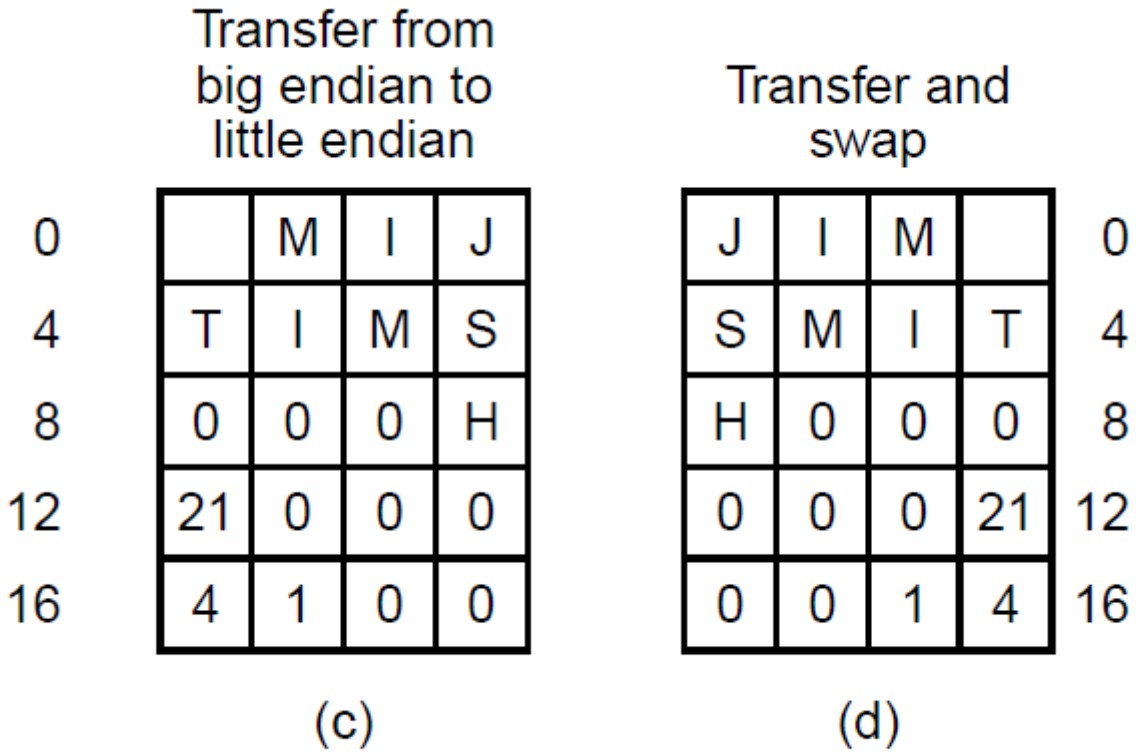

Figure 2-12. (c) The result of transferring the record from a big endian to a little endian. (d) The result of byte swapping (c).

Error-Correcting Codes (1)

Word size	Check bits	Total size	Percent overhead
8	4	12	50
16	5	21	31
32	6	38	19
64	7	71	11
128	8	136	6
256	9	265	4
512	10	522	2

Figure 2-13. Number of check bits for a code that can correct a single error.

Error-Correcting Codes (2)

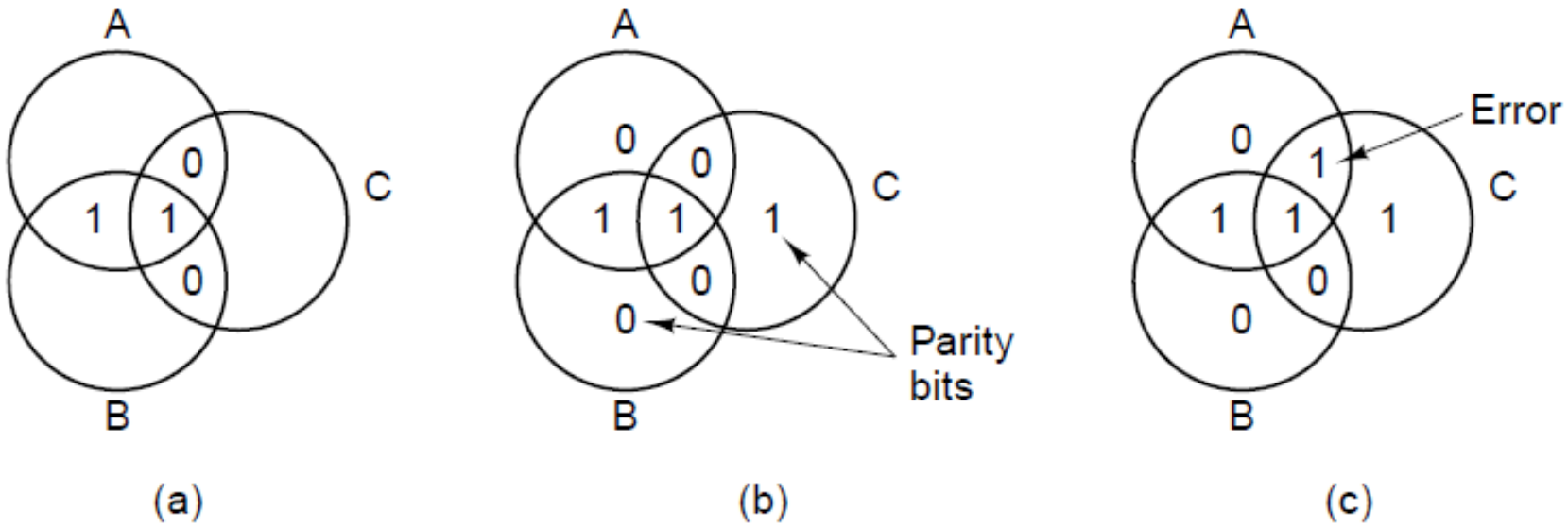

Figure 2-14. (a) Encoding of 1100.
(b) Even parity added.
(c) Error in AC.

Error-Correcting Codes (3)

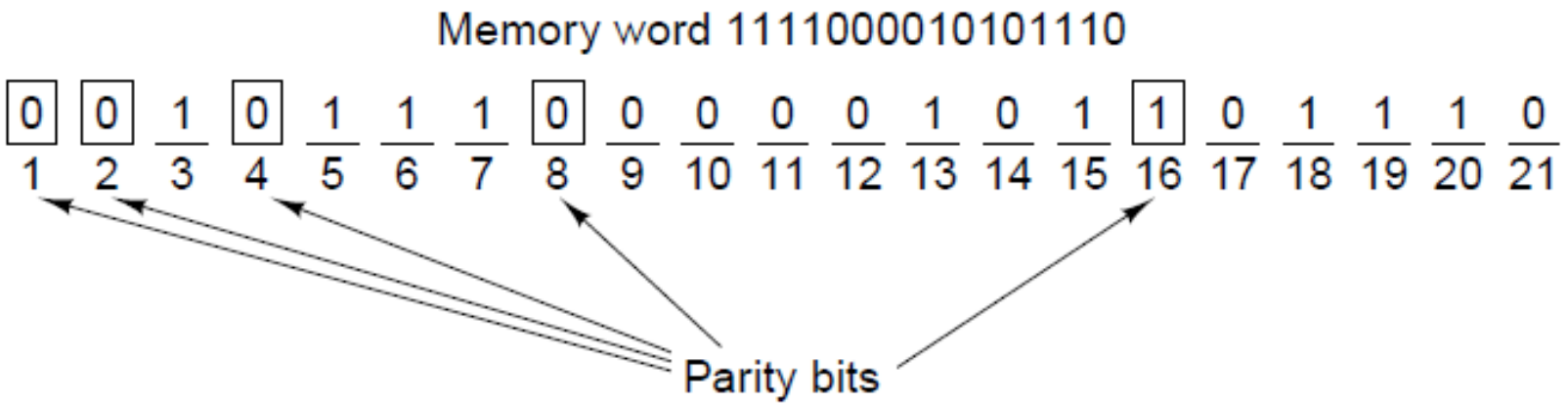

Figure 2-15. Construction of the Hamming code for the memory word 1111000010101110 by adding 5 check bits to the 16 data bits.

Cache Memory

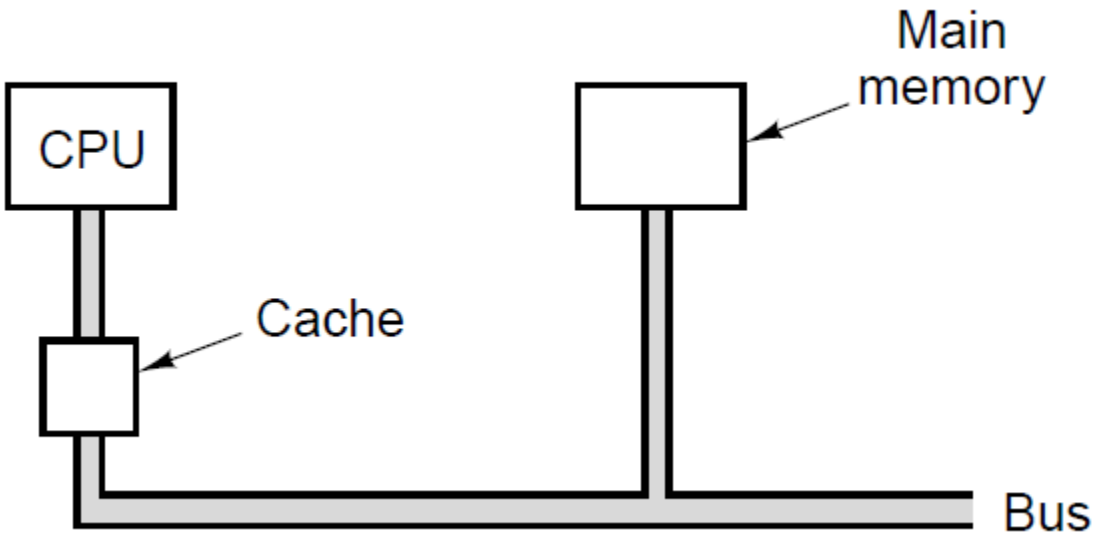

Figure 2-16. The cache is logically between the CPU and main memory. Physically, there are several possible places it could be located.

Memory Packaging and Types

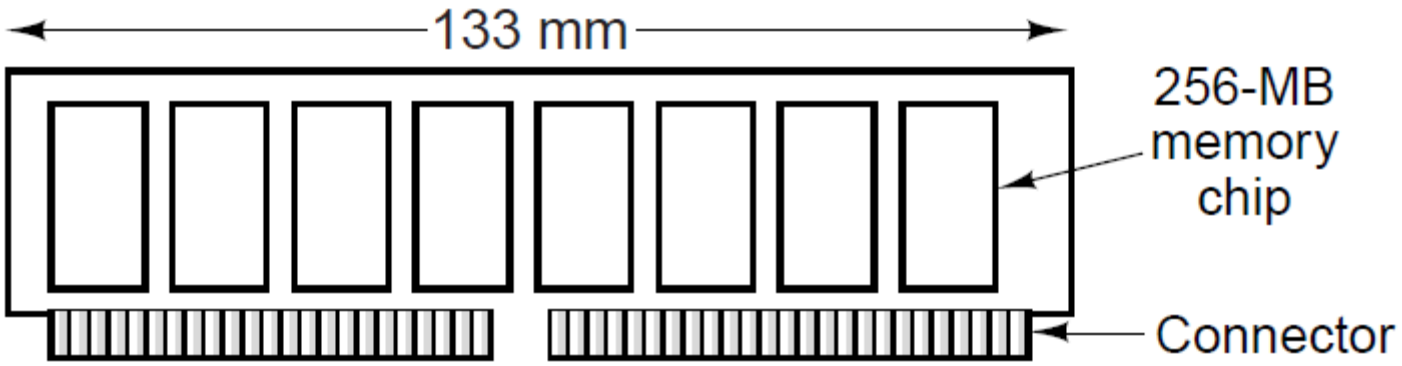

Figure 2-17. Top view of a DIMM holding 4 GB with eight chips of 256 MB on each side. The other side looks the same.

Secondary Memory
Memory Hierarchies

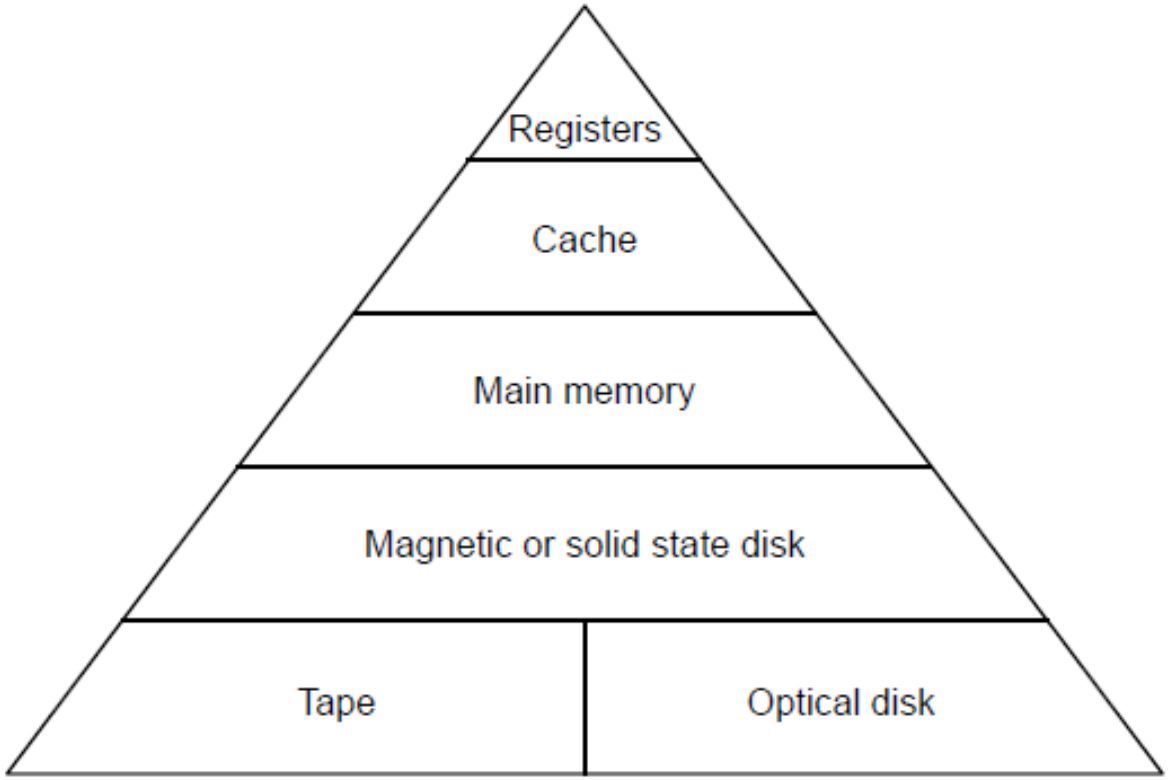

Figure 2-18. A five-level memory hierarchy.

Magnetic Disks (1)

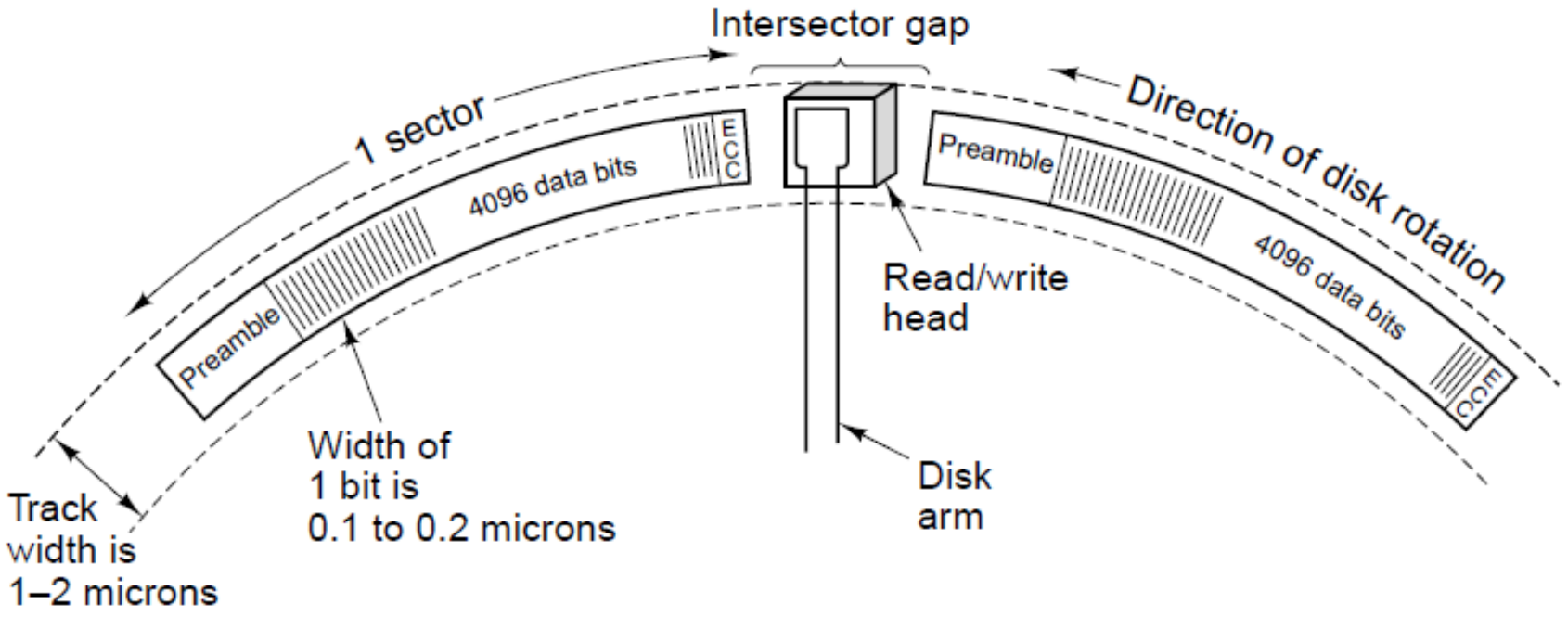

Figure 2-19. A portion of a disk track. Two sectors are illustrated.

Magnetic Disks (2)

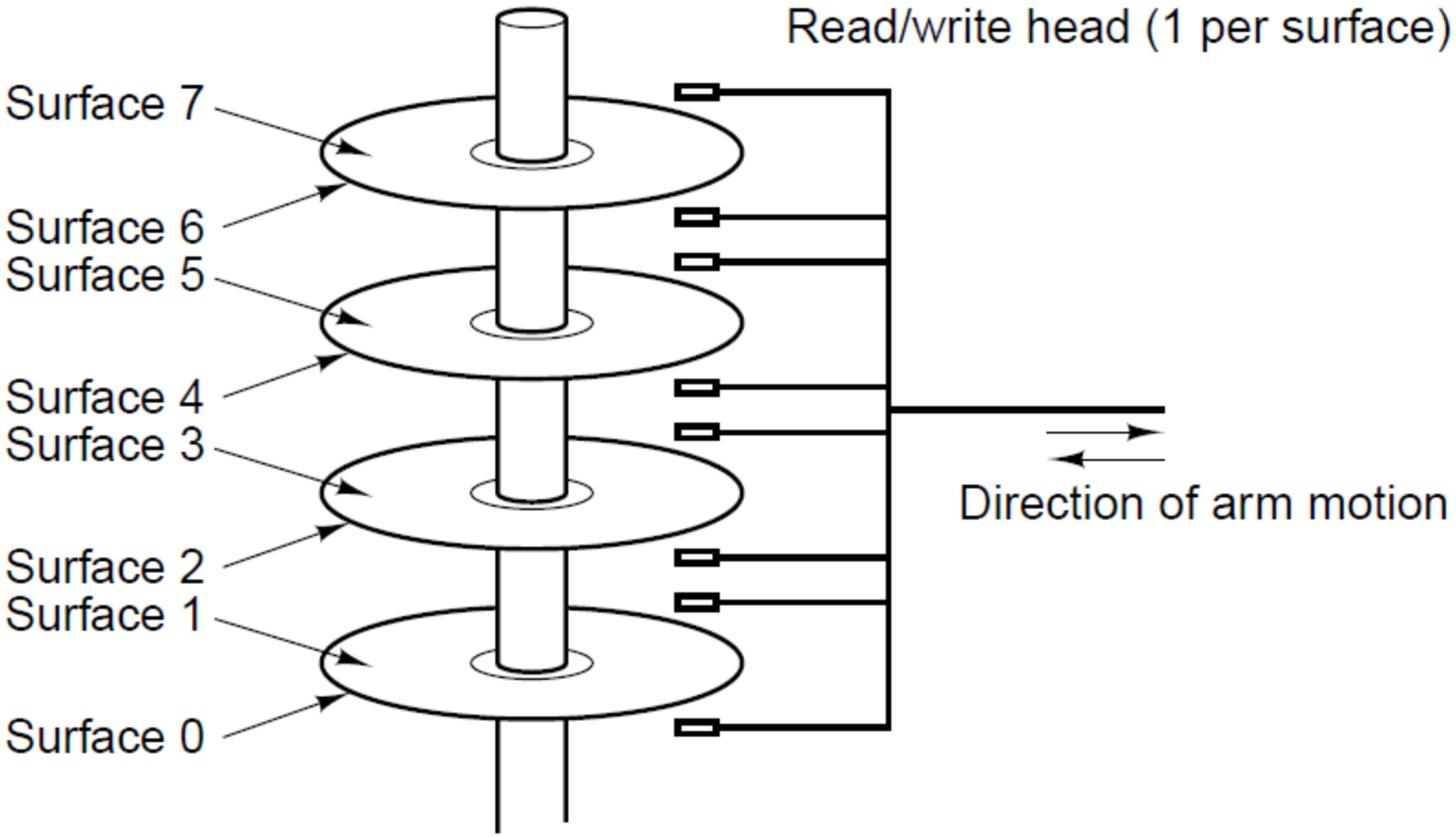

Figure 2-20. A disk with four platters.

Magnetic Disks (3)

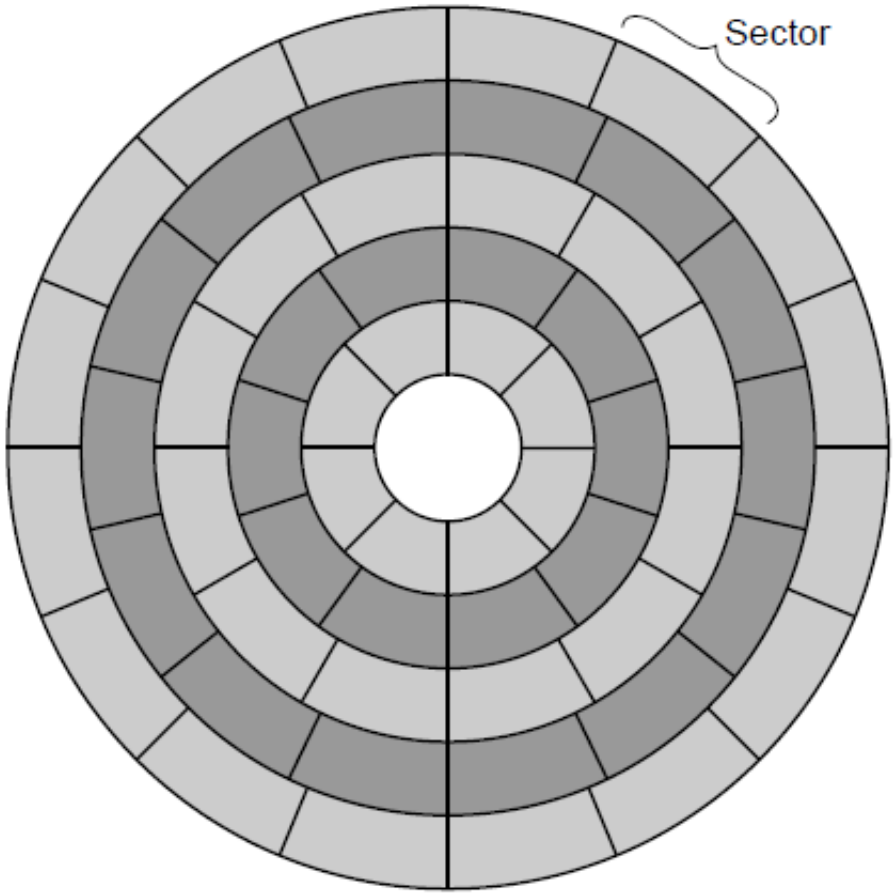

Figure 2-21. A disk with five zones. Each zone has many tracks.

SCSI Disks

Name	Data bits	Bus MHz	MB/sec
SCSI-1	8	5	5
Fast SCSI	8	10	10
Wide Fast SCSI	16	10	20
Ultra SCSI	8	20	20
Wide Ultra SCSI	16	20	40
Ultra2 SCSI	8	40	40
Wide Ultra2 SCSI	16	40	80
Wide Ultra3 SCSI	16	80	160
Wide Ultra4 SCSI	16	160	320
Wide Ultra5 SCSI	16	320	640

Figure 2-22. Some of the possible SCSI parameters

RAID (1)

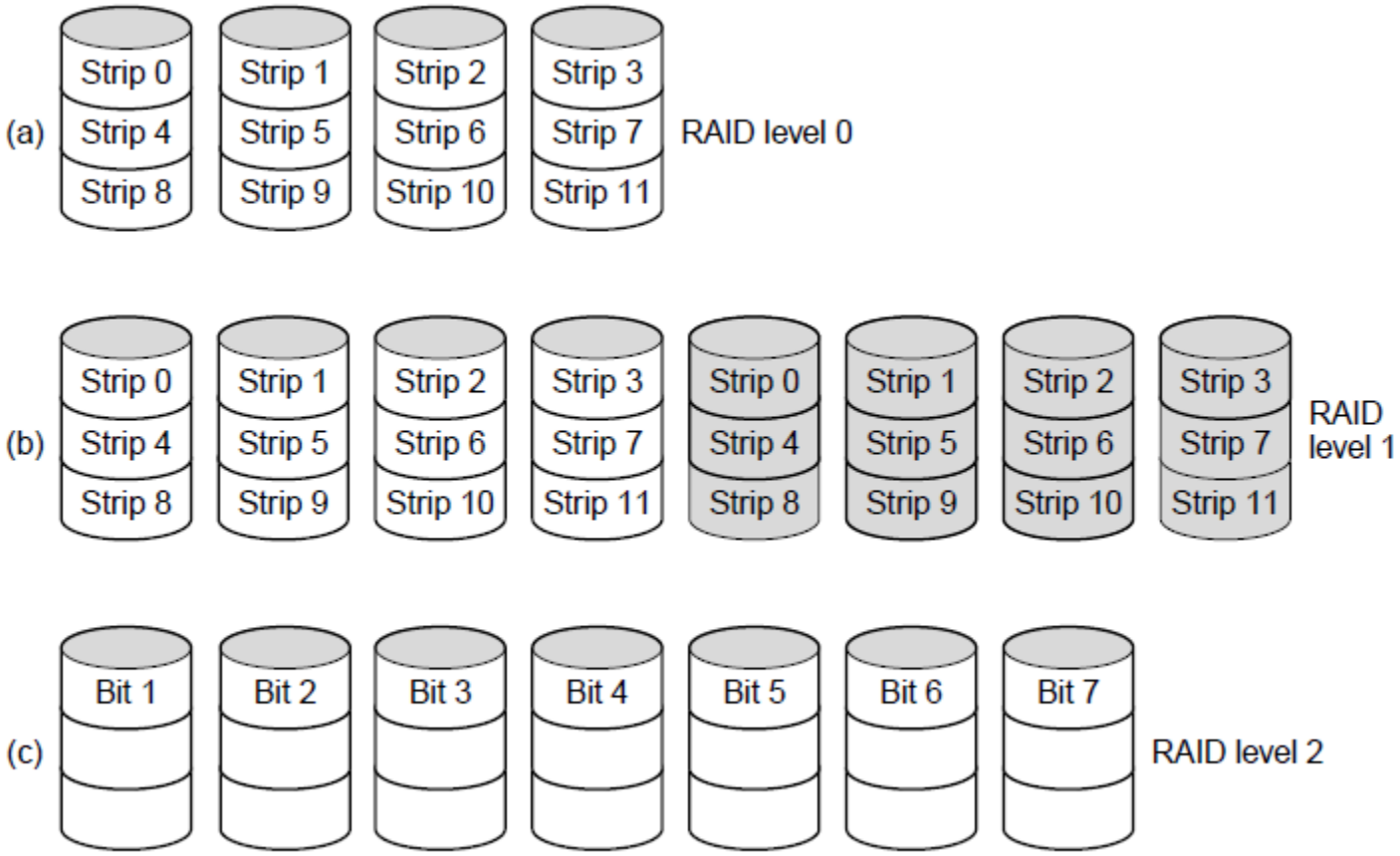

Figure 2-23. RAID levels 0 through 5. Backup and parity drives are shown shaded.

RAID (2)

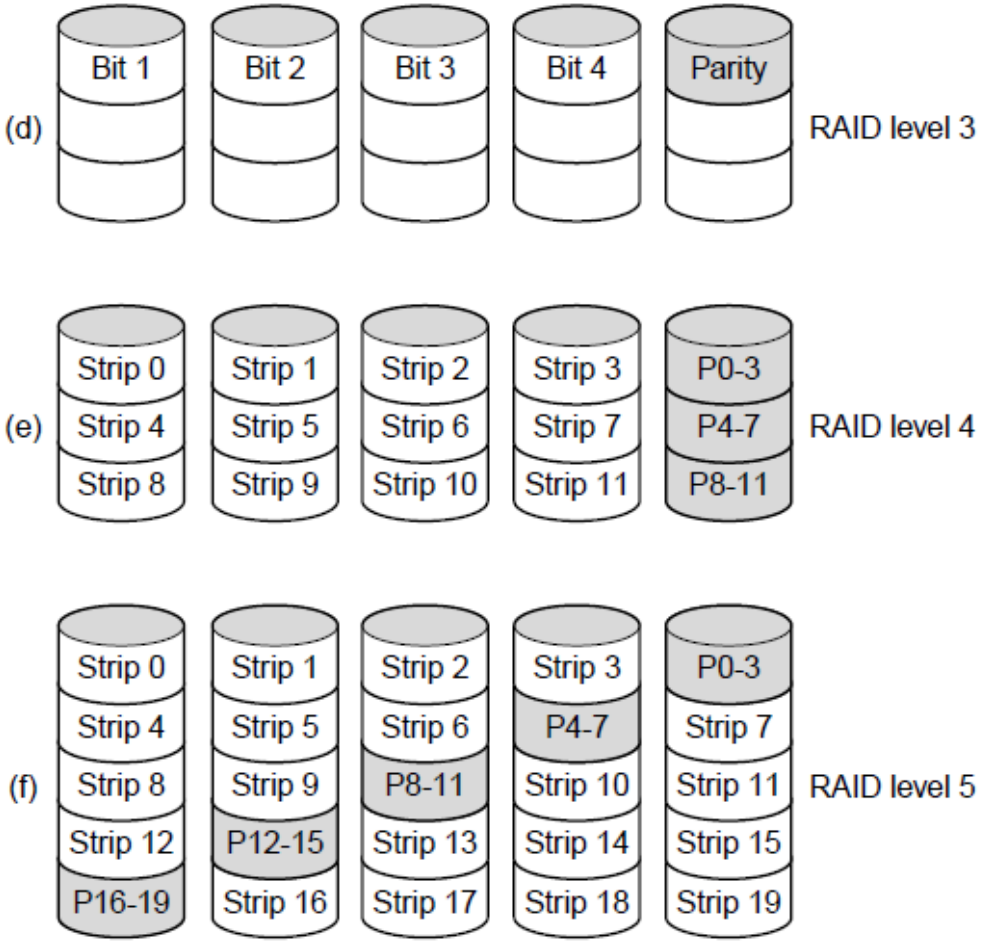

Figure 2-23. RAID levels 0 through 5. Backup and parity drives are shown shaded.

Solid-State Disks

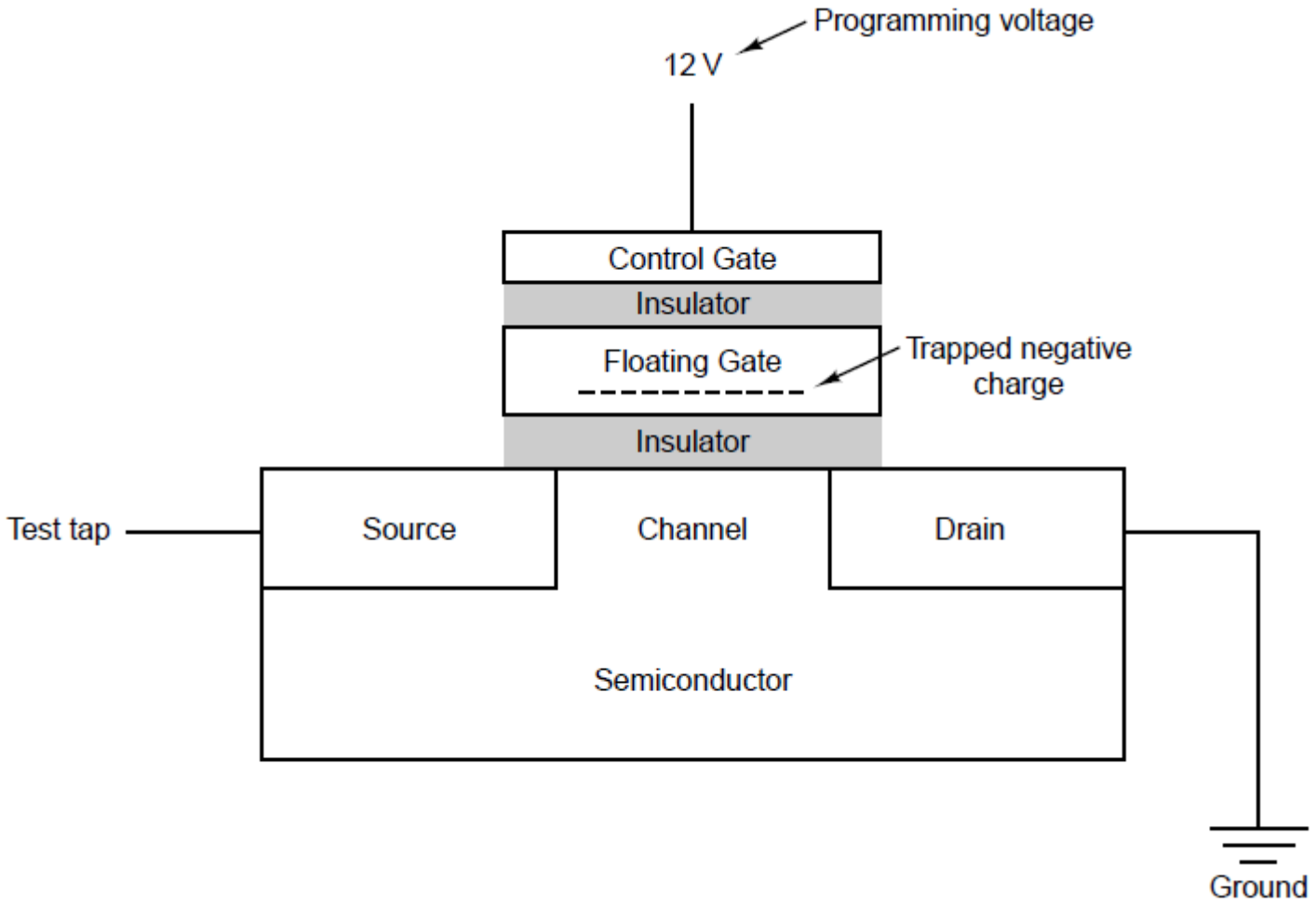

Figure 2-24. A flash memory cell.

CD-ROMs (1)

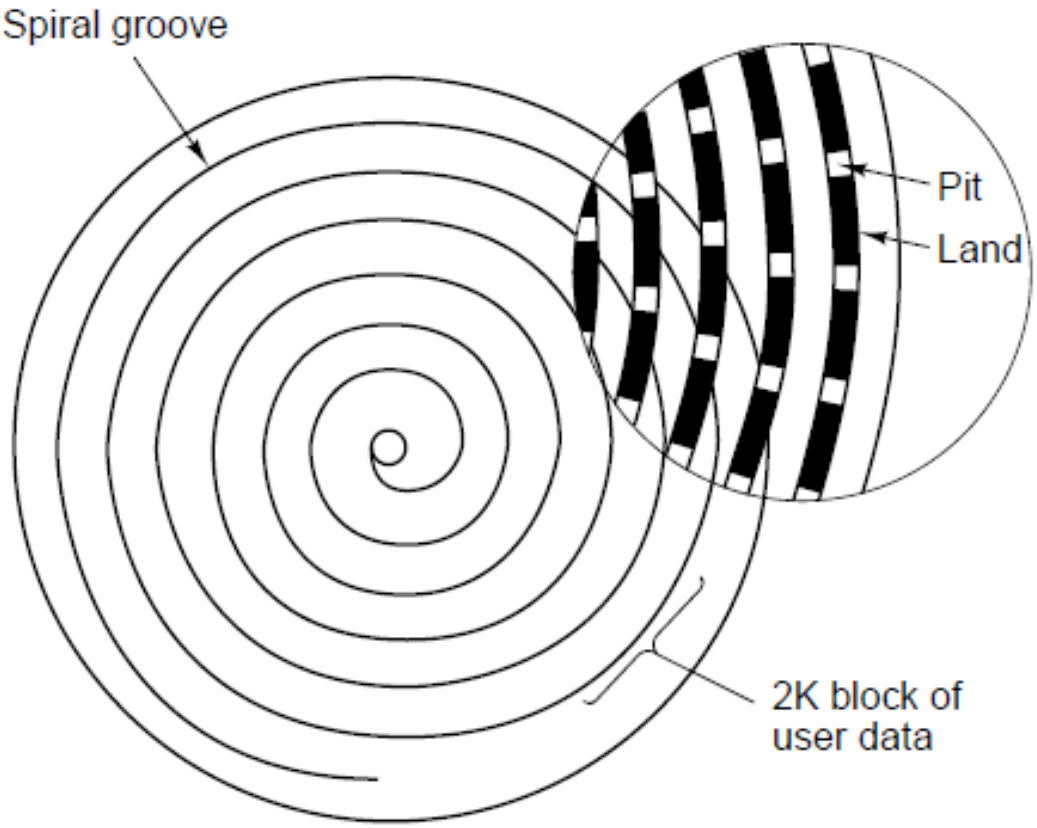

Figure 2-25. Recording structure of a Compact Disc or CD-ROM.

CD-ROMs (2)

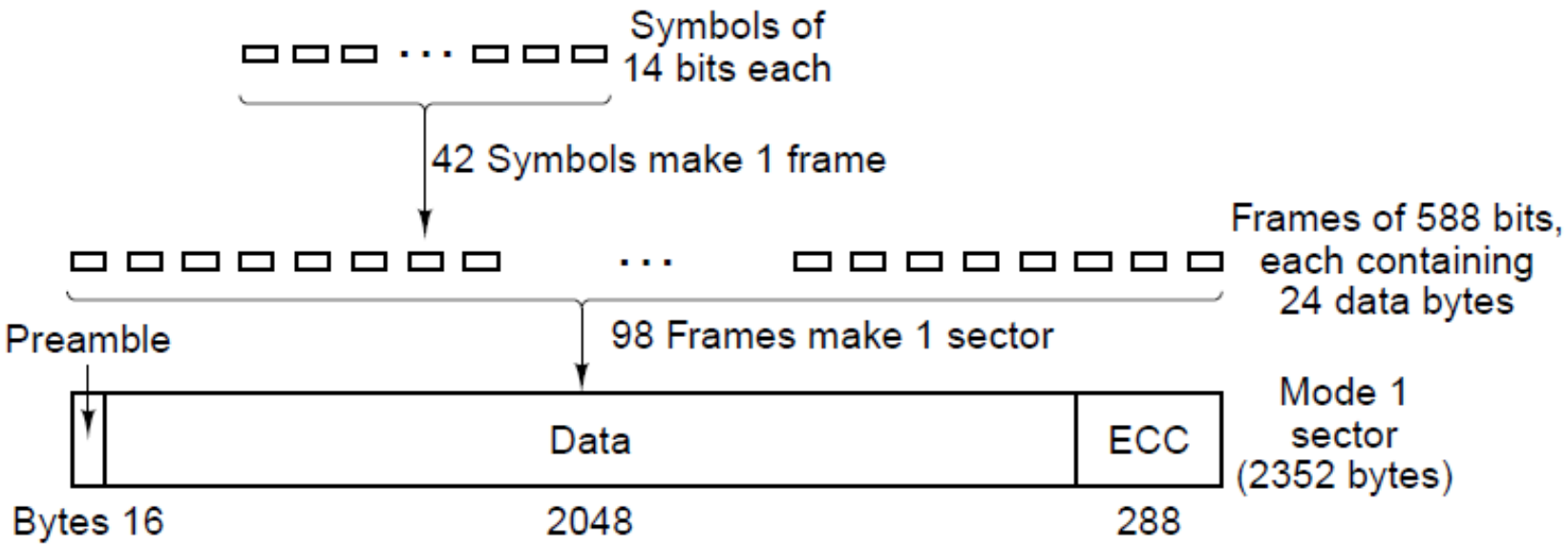

Figure 2-26. Logical data layout on a CD-ROM.

CD-Recordables

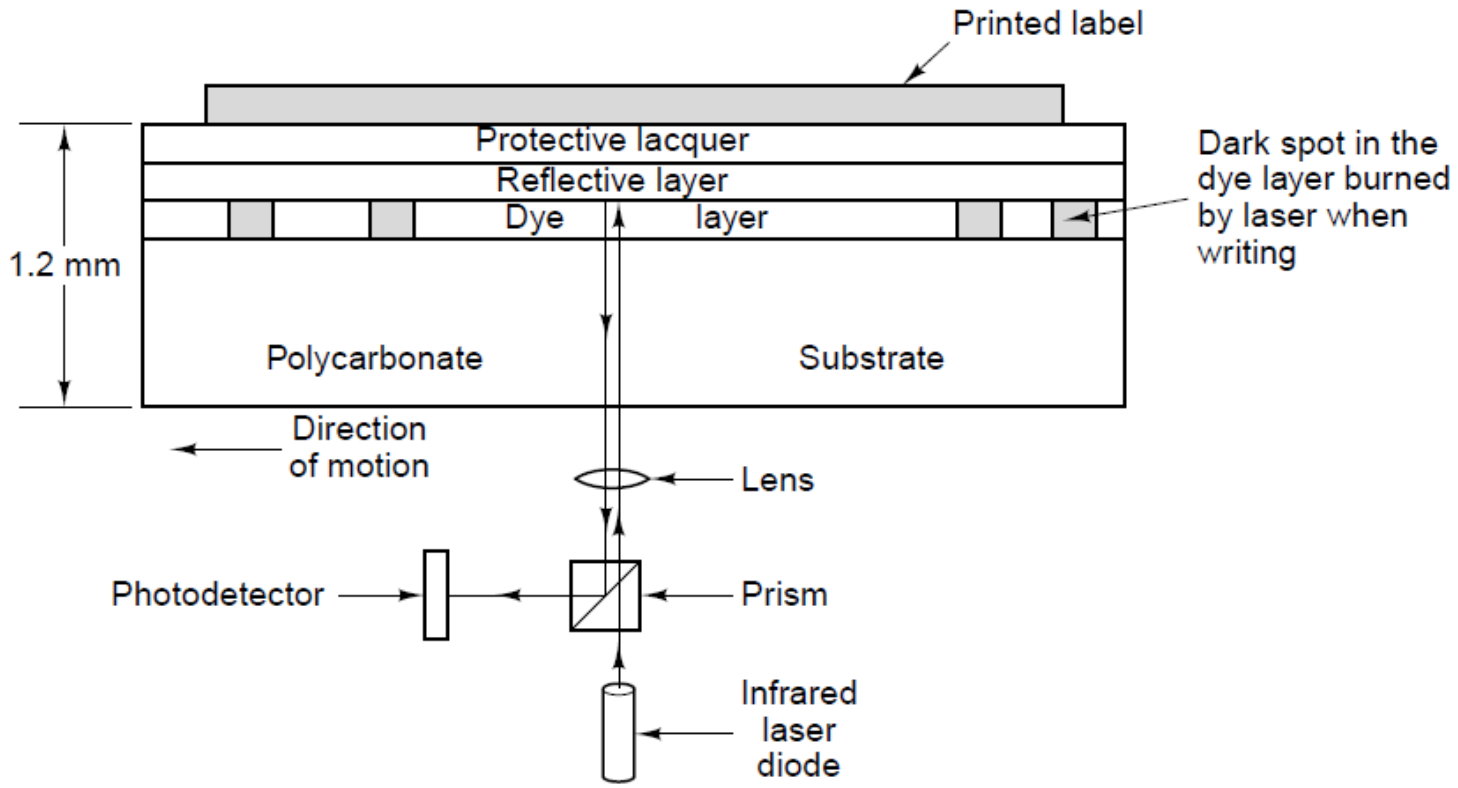

Figure 2-27. Cross section of a CD-R disk and laser (not to scale). A CD-ROM has a similar structure, except without the dye layer and with a pitted aluminum layer instead of a reflective layer.

DVD (1)

- Was **D**igital **V**ideo **D**isk
- Now **D**igital **V**ersatile **D**isk
- New features
  1. Smaller pits
  2. A tighter spiral
  3. A red laser

DVD (2)

Formats:

- Single-sided, single-layer (4.7 GB)
- Single-sided, dual-layer (8.5 GB)
- Double-sided, single-layer (9.4 GB)
- Double-sided, dual-layer (17 GB)

DVD (3)

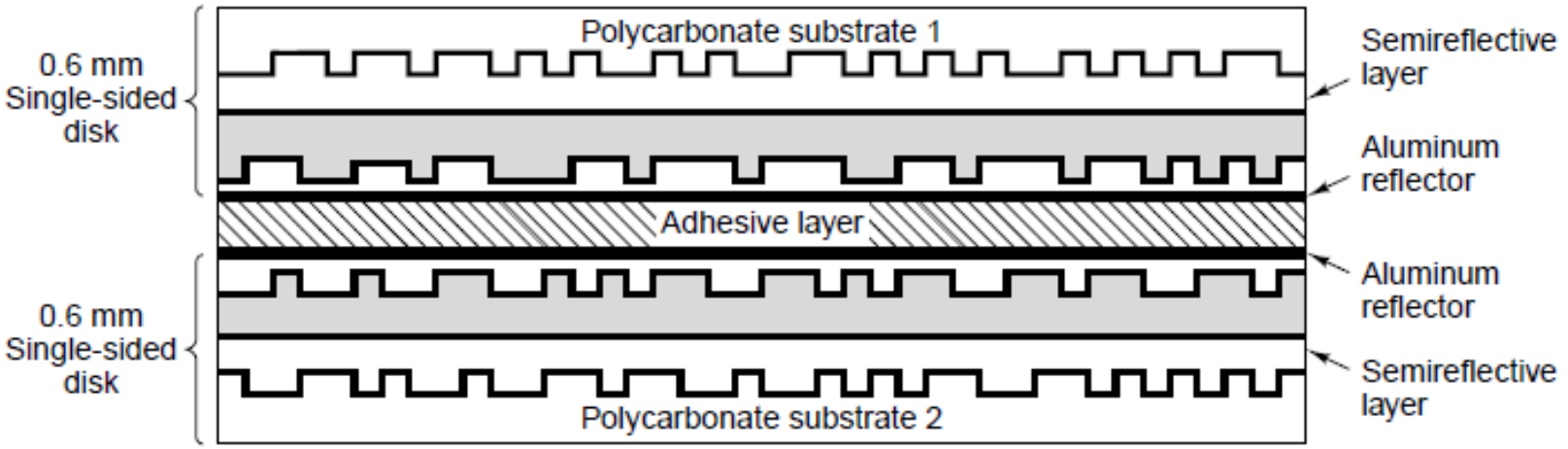

Figure 2-28 A double sided, dual layer DVD disk.

Input/Output Buses (1)

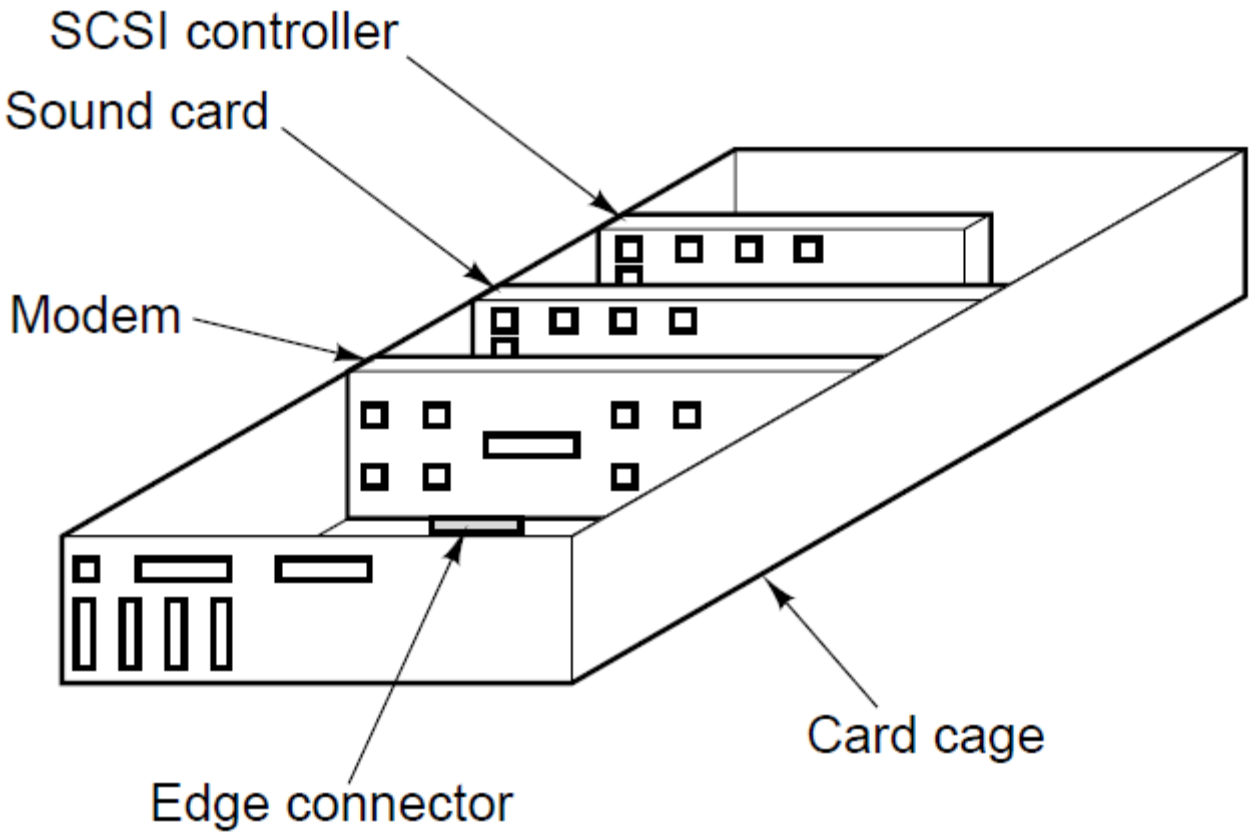

Figure 2-29. Physical structure of a personal computer.

Buses (2)

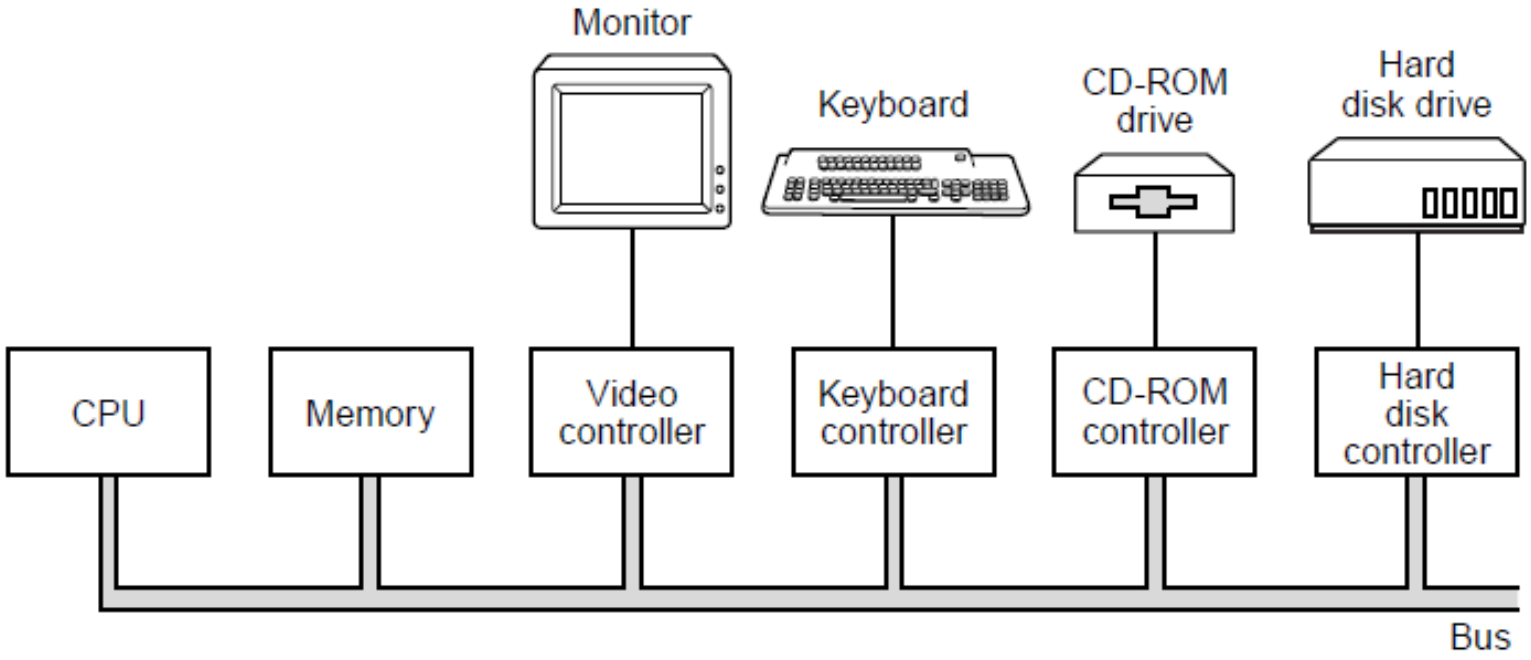

Figure 2-30. Logical structure of a simple personal computer.

PCI and PCIe Buses (1)

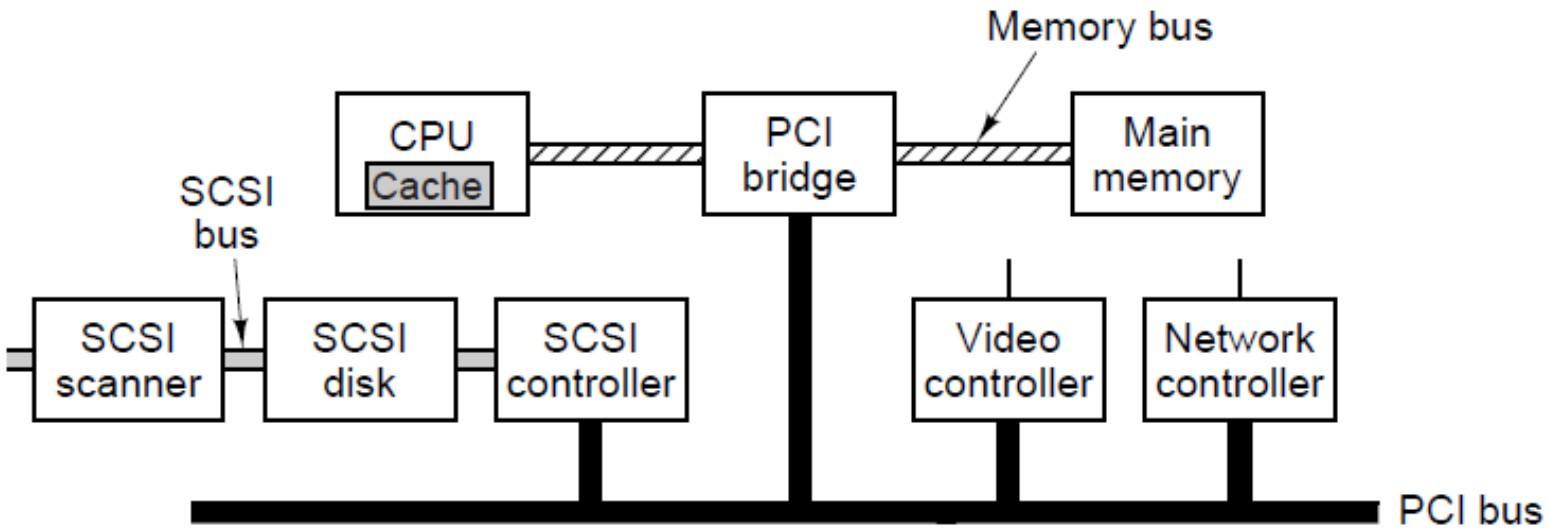

Figure 2-31. A typical PC built around the PCI bus. The SCSI controller is a PCI device.

PCI and PCIe Buses (2)

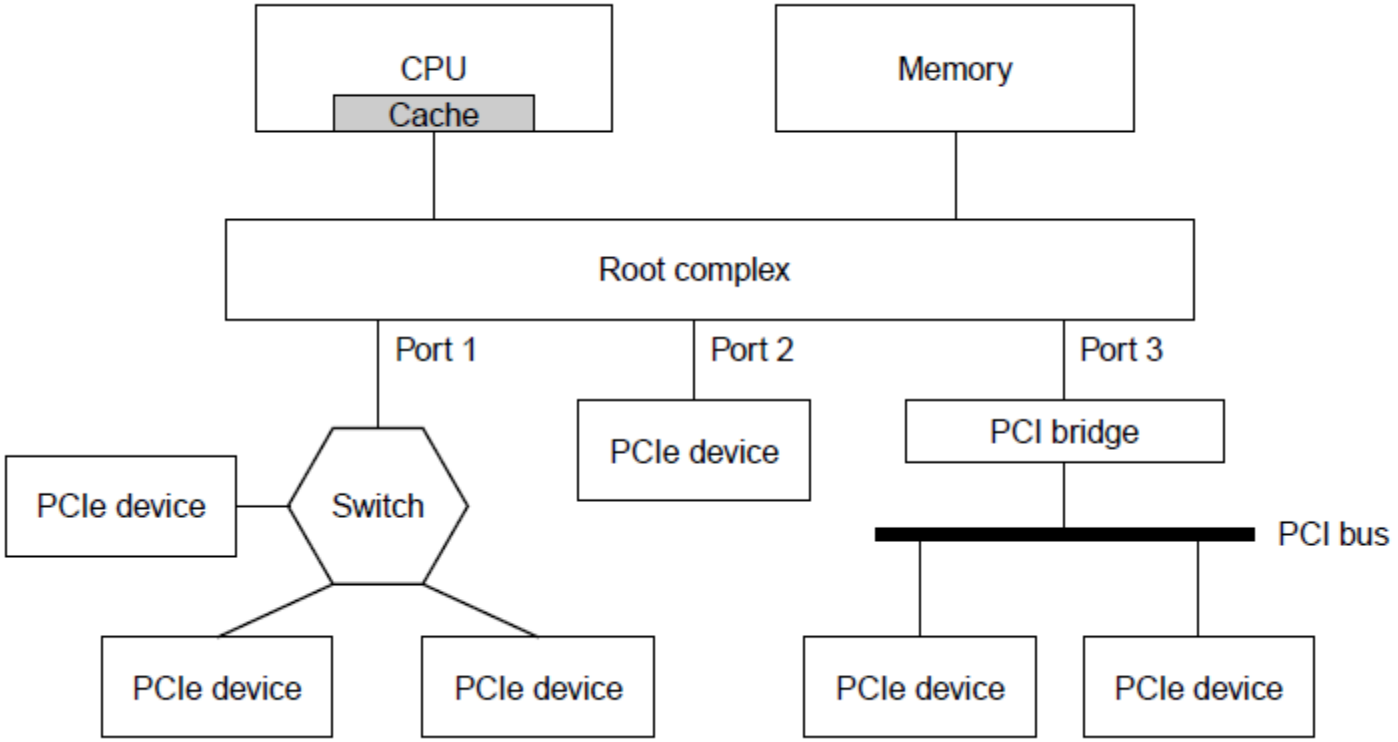

Figure 2-32. Sample architecture of a PCIe system with three PCIe ports.

Terminals

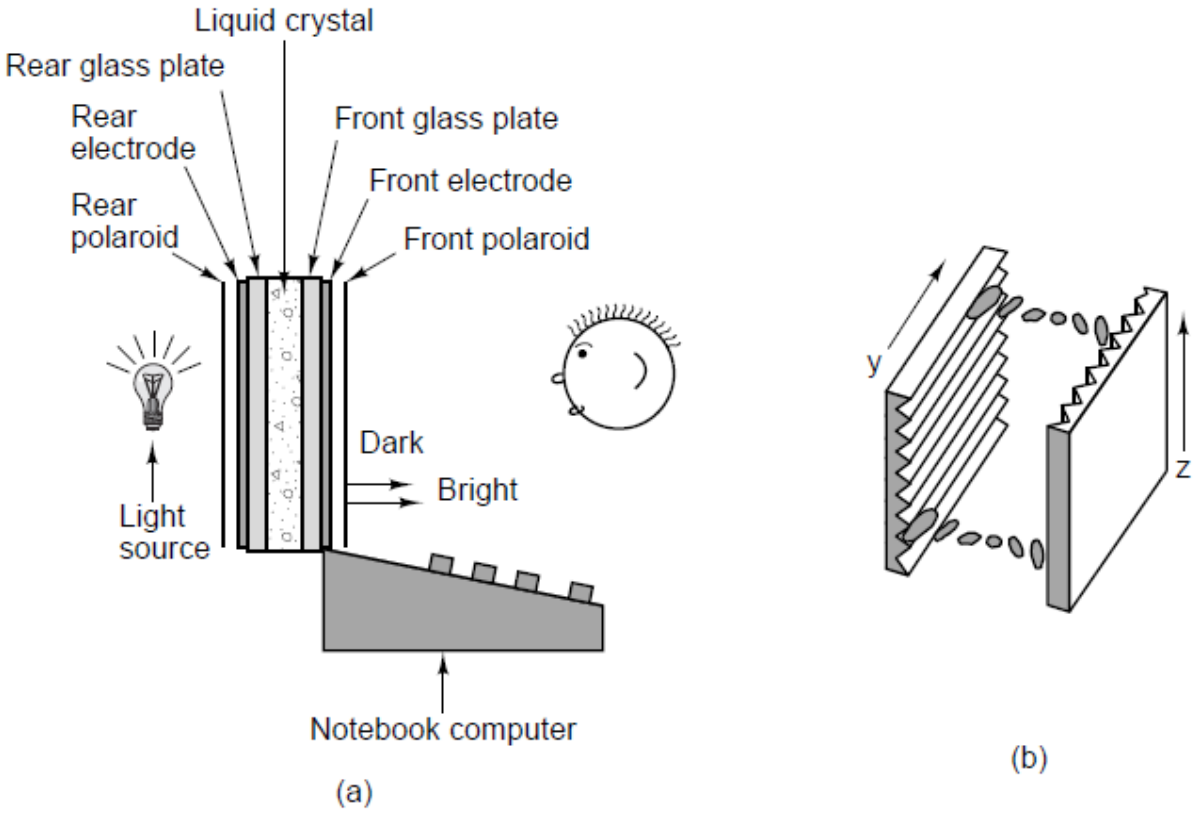

Figure 2-33. (a) The construction of an LCD screen. (b) The grooves on the rear and front plates are perpendicular to one another.

Mice

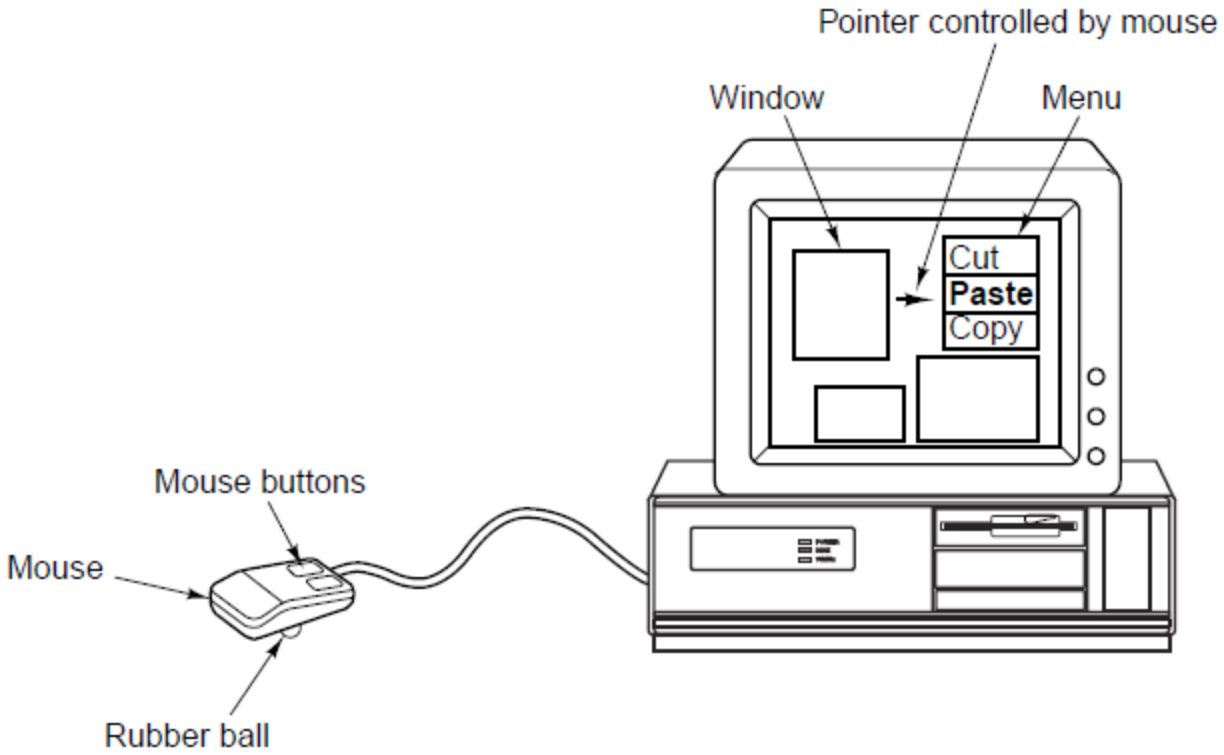

Figure 2-34. A mouse being used to point to menu items.

Game Controllers (1)

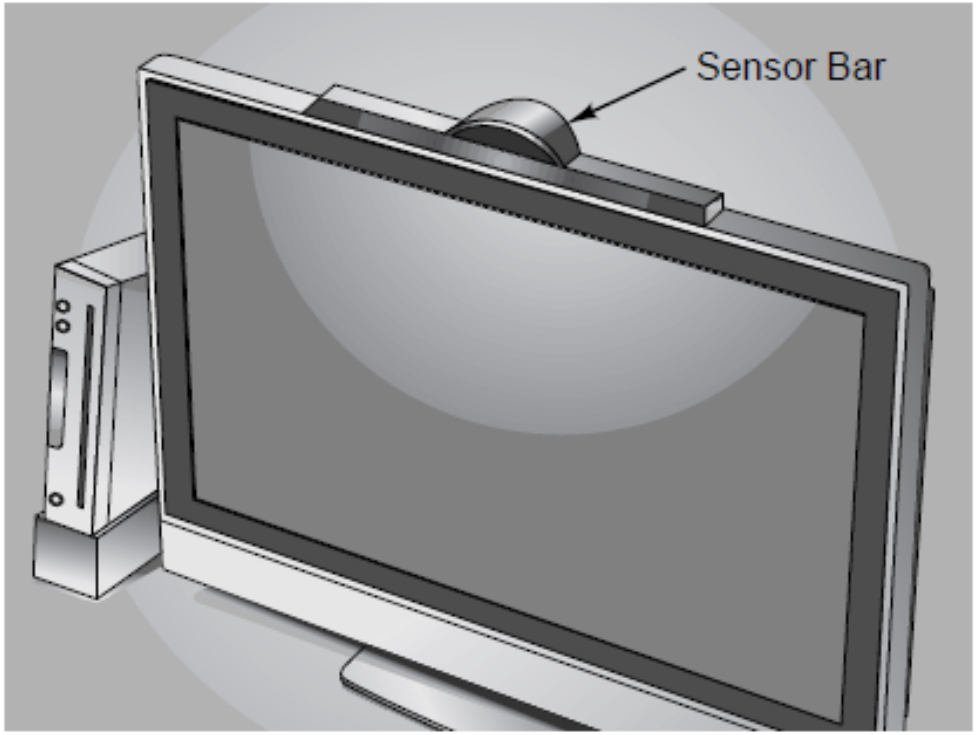

Figure 2-35. The Wiimote video game controller motion sensors.

Game Controllers (2)

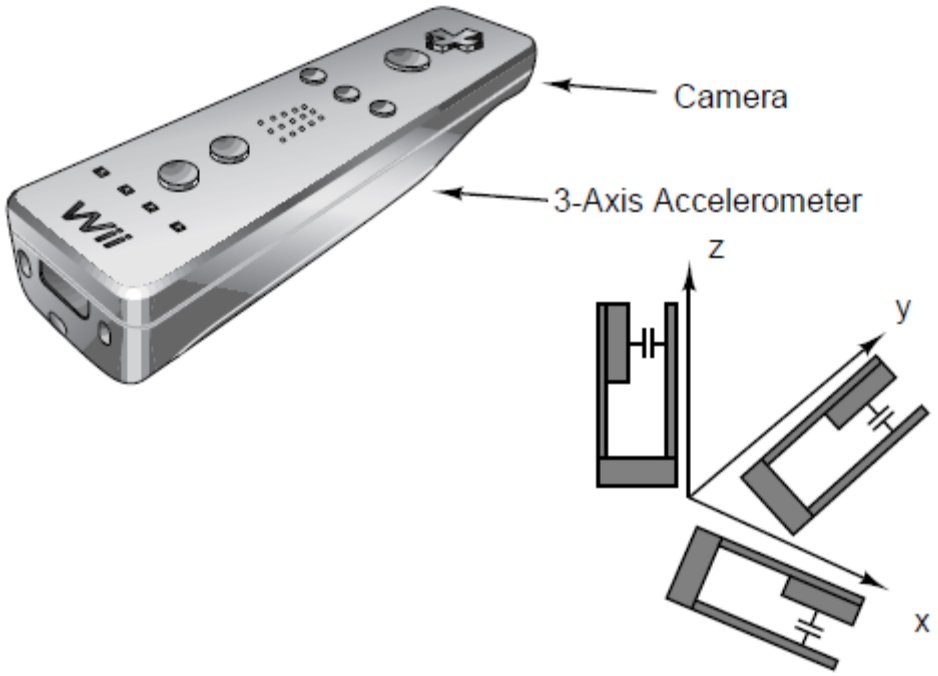

Figure 2-35. The Wiimote video game controller motion sensors.

Laser Printers (1)

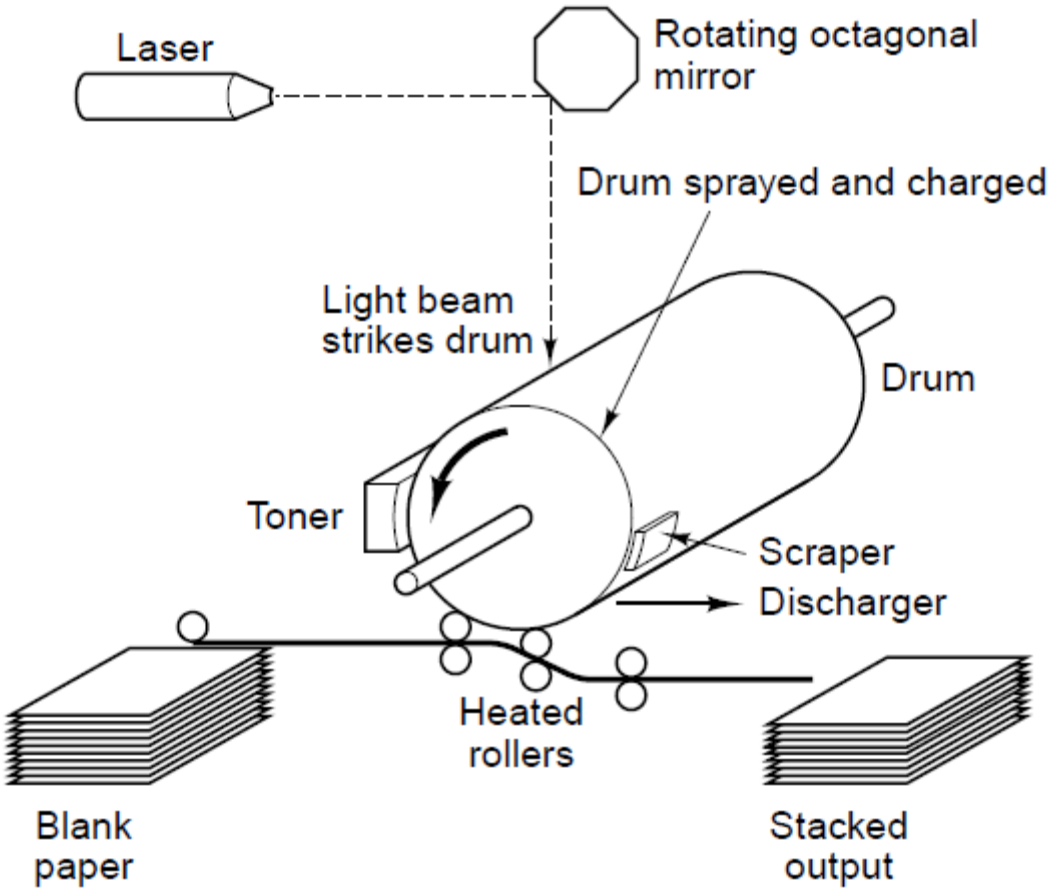

Figure 2-36. Operation of a laser printer.

Laser Printers (2)

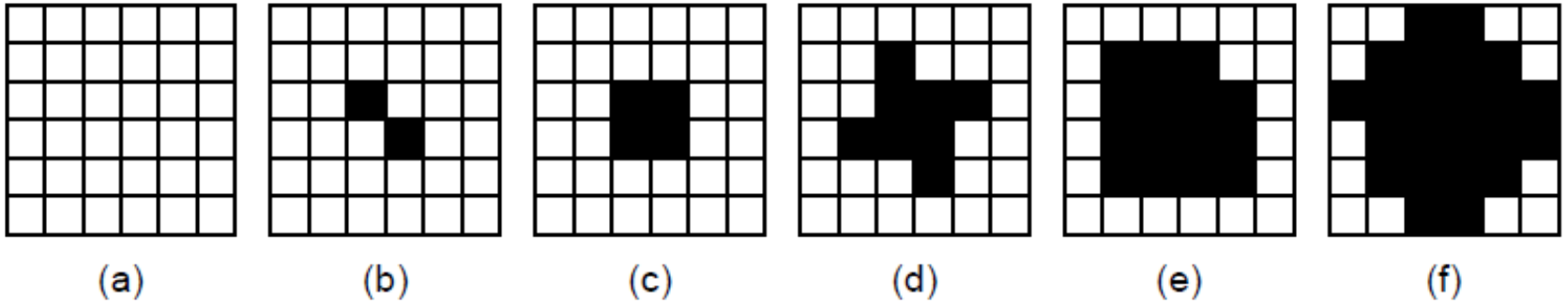

Figure 2-37. Halftone dots for various grayscale ranges. (a) 0–6. (b) 14–20. (c) 28–34. (d) 56–62. (e) 105–111. (f) 161–167.

Color Printing

- Color monitors use transmitted light;
  - Color printers use reflected light.
- Monitors have 256 intensities per color;
  - Color printers must halftone.
- Monitors have a dark background;
  - Paper has a light background.
- The RGB gamut of a monitor and the CMYK gamut of a printer are different.

Modems

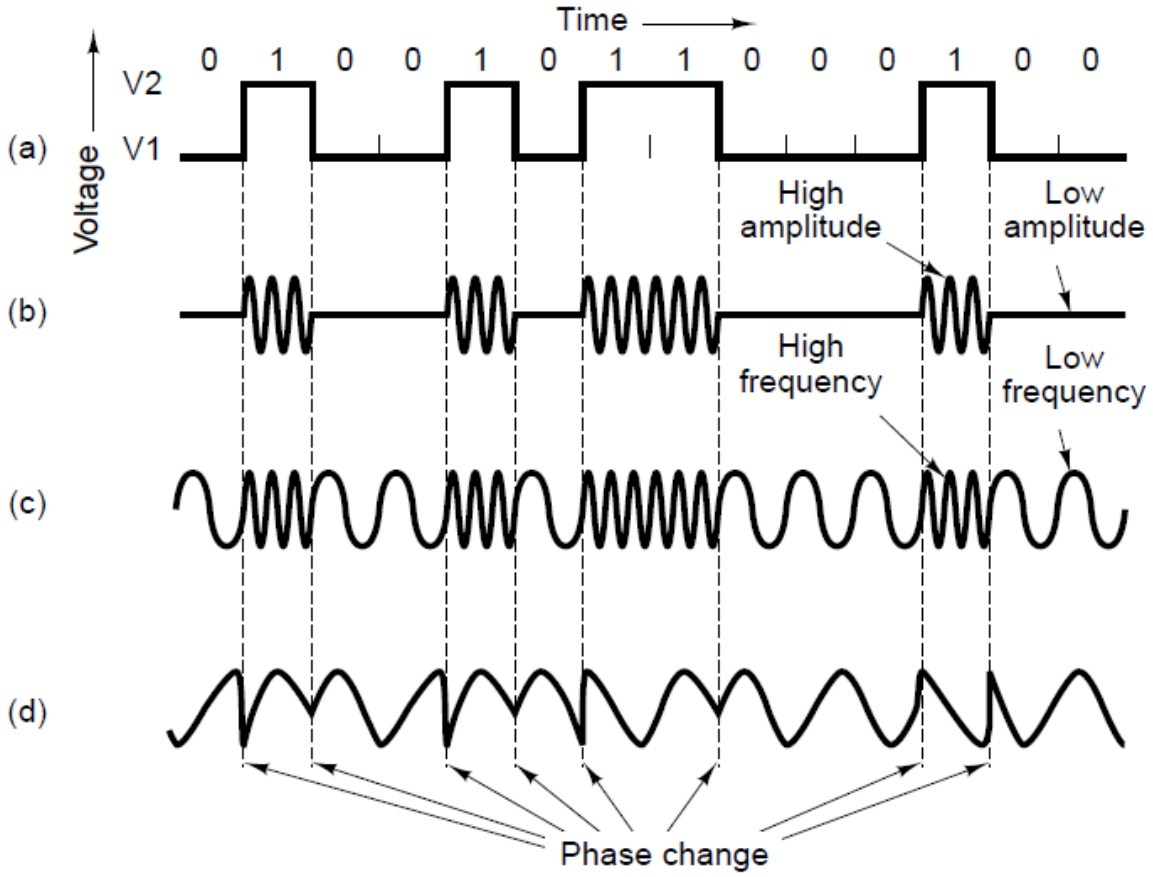

Figure 2-38. Transmission of the binary number 0100101100100 over a telephone line bit by bit. (a) Two-level signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

Digital Subscriber Lines (1)

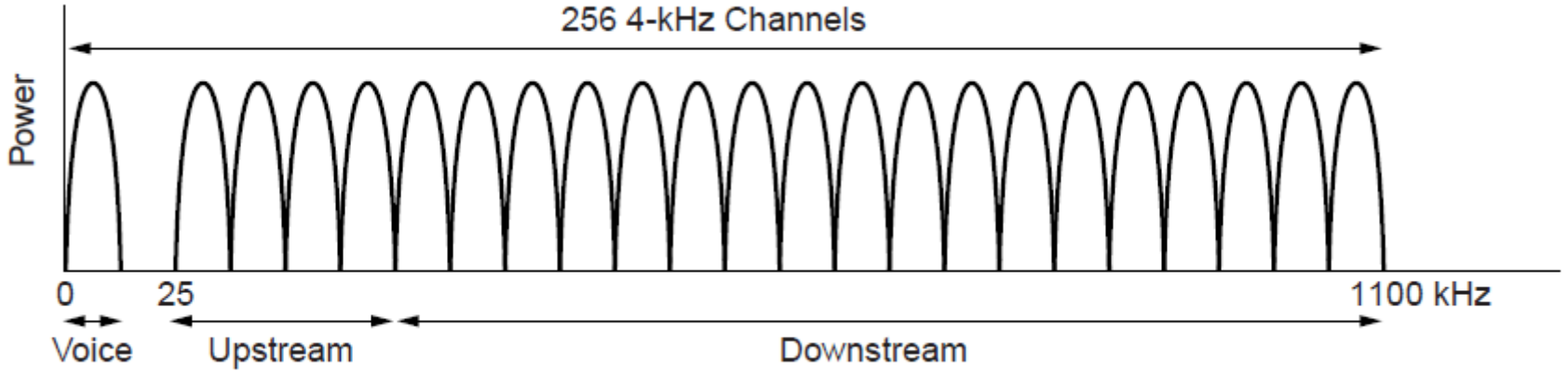

Figure 2-39. Operation of ADSL.

Digital Subscriber Lines (2)

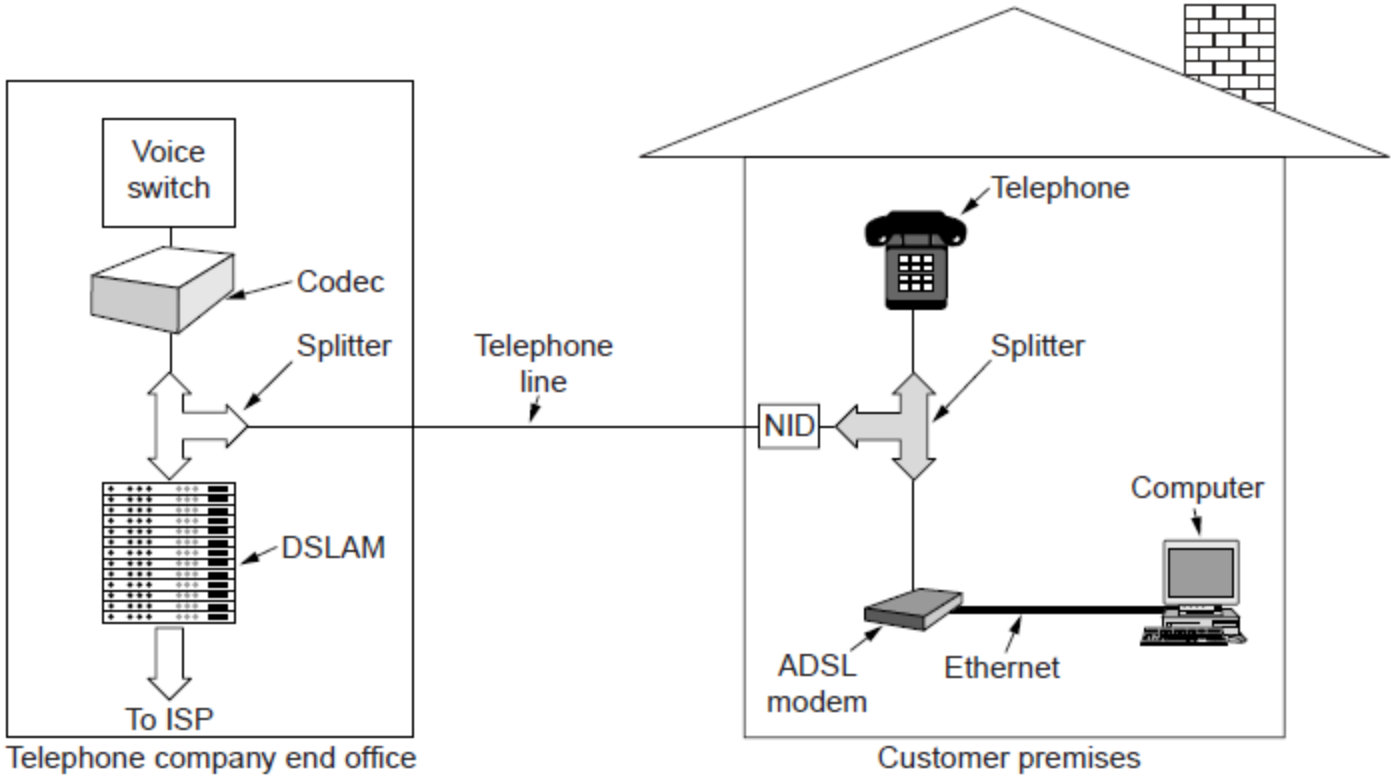

Figure 2-40. A typical ADSL equipment configuration.

Internet over Cable (1)

Problems to overcome:

How to add Internet access without interfering with TV programs.

How to have two-way traffic when amplifiers are inherently one way.

Internet over Cable (2)

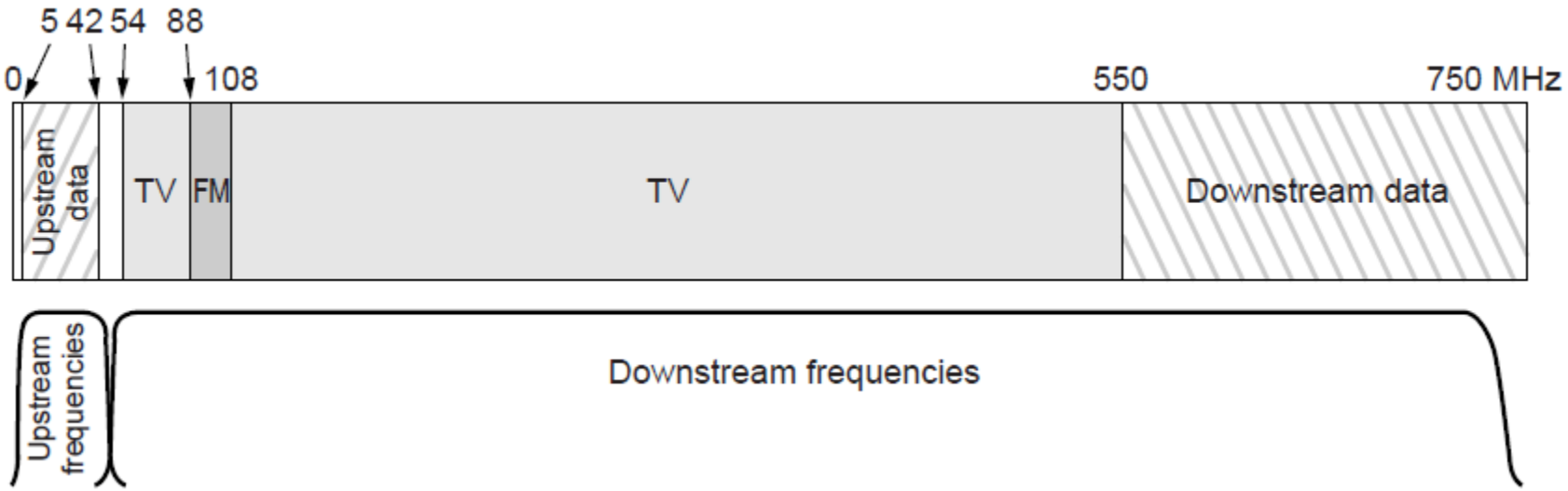

Figure 2-41. Frequency allocation in a typical cable TV system used for Internet access.

Internet over Cable (3)

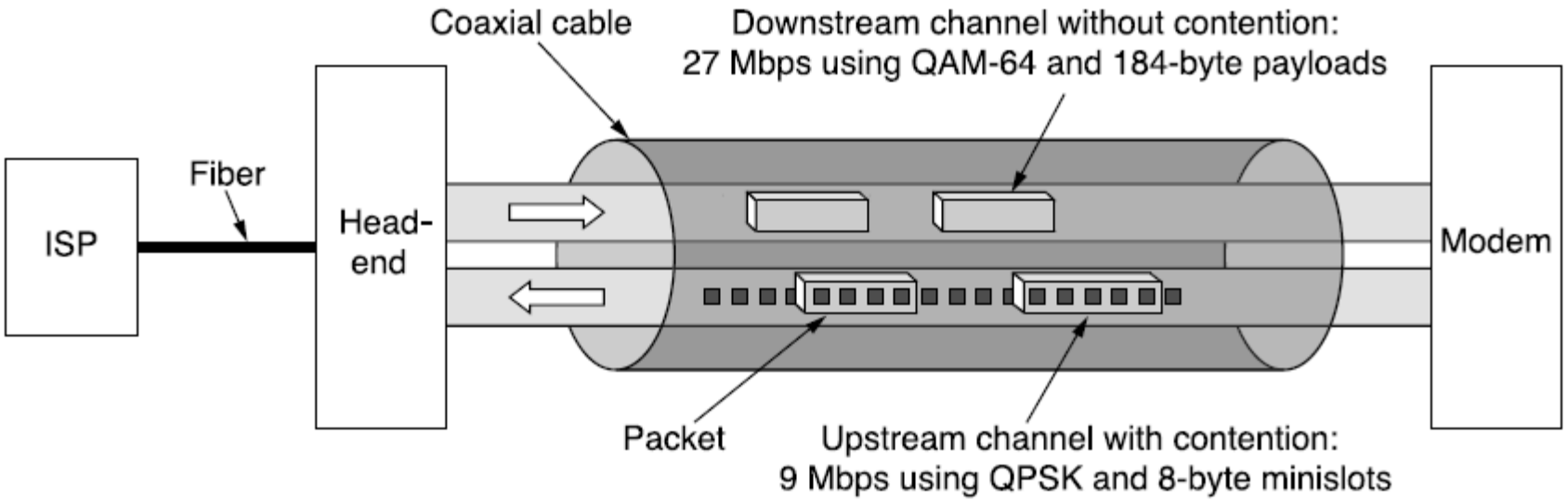

Figure 2-42. Typical details of the upstream and downstream channels in North America. QAM-64 allows 6 bits/Hz but works only at high frequencies. QPSK works at low frequencies but allows only 2 bits/Hz.

Digital Cameras

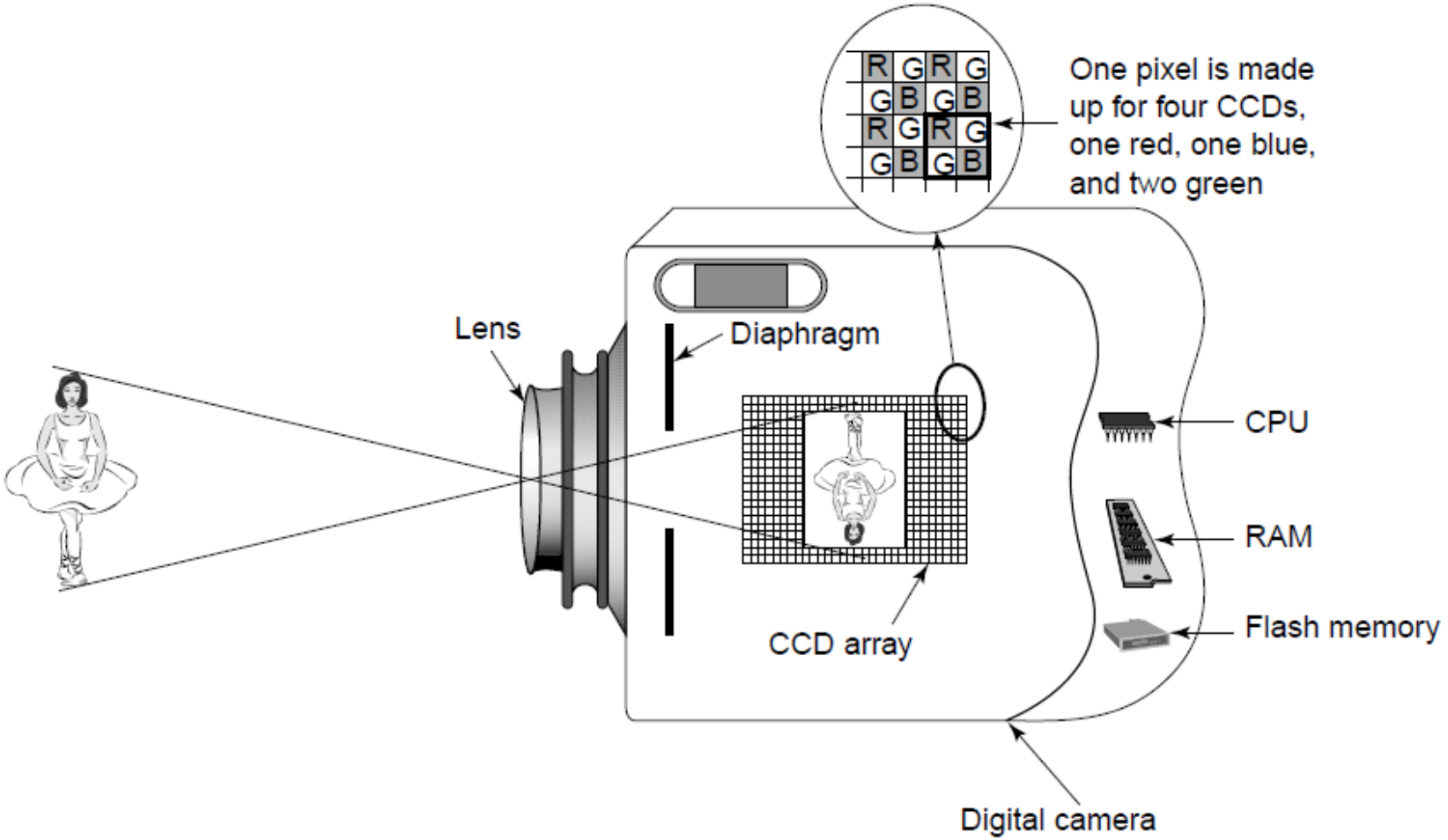

Figure 2-43. A digital camera.

Character Codes (1)

Hex	Name	Meaning	Hex	Name	Meaning
0	NUL	Null	10	DLE	Data Link Escape
1	SOH	Start Of Heading	11	DC1	Device Control 1
2	STX	Start Of TeXt	12	DC2	Device Control 2
3	ETX	End Of TeXt	13	DC3	Device Control 3
4	EOT	End Of Transmission	14	DC4	Device Control 4
5	ENQ	Enquiry	15	NAK	Negative AcKnowledgement
6	ACK	ACKnowledgement	16	SYN	SYNchronous idle
7	BEL	BELl	17	ETB	End of Transmission Block
8	BS	BackSpace	18	CAN	CANcel
9	HT	Horizontal Tab	19	EM	End of Medium
A	LF	Line Feed	1A	SUB	SUBstitute
B	VT	Vertical Tab	1B	ESC	ESCape
C	FF	Form Feed	1C	FS	File Separator
D	CR	Carriage Return	1D	GS	Group Separator
E	SO	Shift Out	1E	RS	Record Separator
F	SI	Shift In	1F	US	Unit Separator

Figure 2-44. The ASCII character set.

Character Codes (2)

Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char	Hex	Char
20	(Space)	30	0	40	@	50	P	60	‘	70	p
21	!	31	1	41	A	51	Q	61	a	71	q
22	"	32	2	42	B	52	R	62	b	72	r
23	#	33	3	43	C	53	S	63	c	73	s
24	$	34	4	44	D	54	T	64	d	74	t
25	%	35	5	45	E	55	U	65	e	75	u
26	&	36	6	46	F	56	V	66	f	76	v
27	’	37	7	47	G	57	W	67	g	77	w
28	(	38	8	48	H	58	X	68	h	78	x
29	)	39	9	49	I	59	Y	69	i	79	y
2A	*	3A	:	4A	J	5A	Z	6A	j	7A	z
2B	+	3B	;	4B	K	5B	[	6B	k	7B	{
2C	,	3C	<	4C	L	5C	\	6C	l	7C	|
2D	-	3D	=	4D	M	5D	]	6D	m	7D	}
2E	.	3E	>	4E	N	5E	^	6E	n	7E	~
2F	/	3F	?	4F	O	5F	_	6F	o	7F	DEL

Figure 2-44. The ASCII character set.

Character Codes (3)

Bits	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
7	0ddddddd					
11	110ddddd	10dddddd				
16	1110dddd	10dddddd	10dddddd			
21	11110ddd	10dddddd	10dddddd	10dddddd		
26	111110dd	10dddddd	10dddddd	10dddddd	10dddddd	
31	1111110x	10dddddd	10dddddd	10dddddd	10dddddd	10dddddd

Figure 2-45. The UTF-8 encoding scheme.

End

Chapter 2