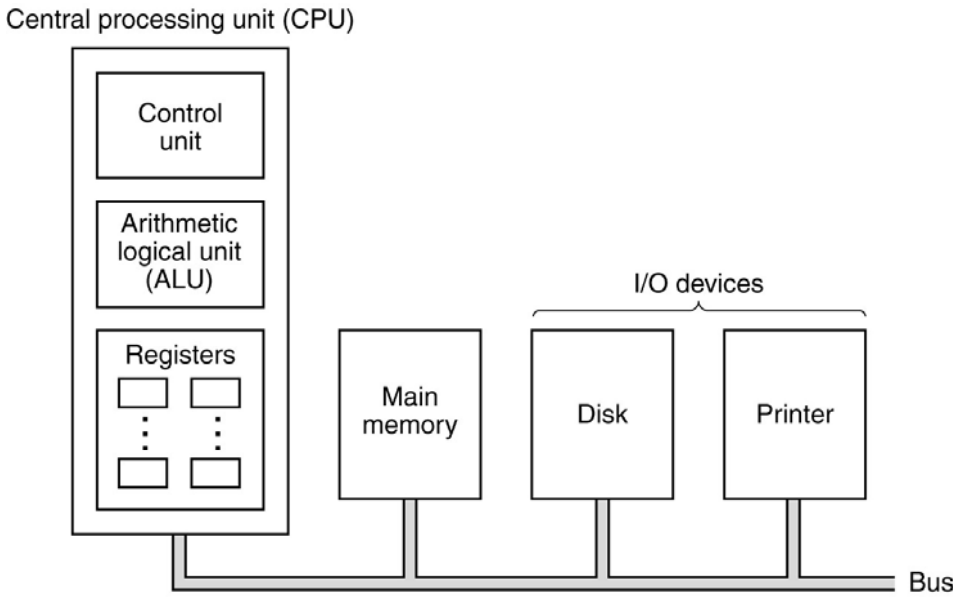
Computer Systems Organization

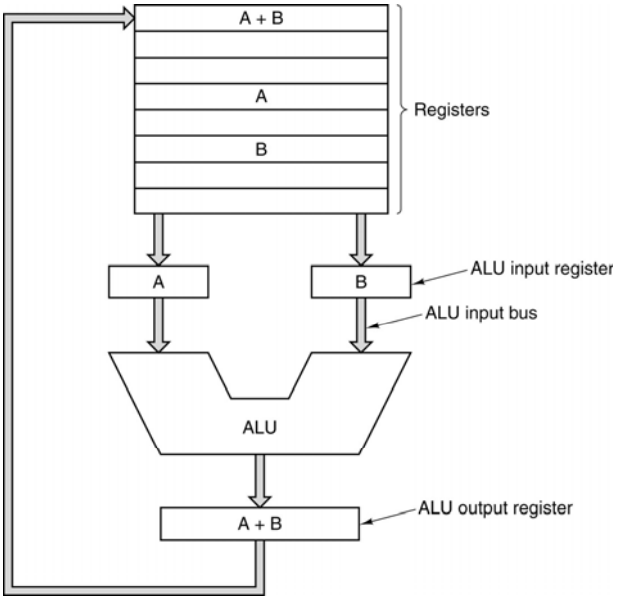
Chapter 2

Central Processing Unit



The organization of a simple computer with one CPU and two I/O devices

CPU Organization



The data path of a typical Von Neumann machine.

Instruction Execution Steps

1. Fetch next instruction from memory into instr. register
2. Change program counter to point to next instruction
3. Determine type of instruction just fetched
4. If instructions uses word in memory, determine where Fetch word, if needed, into CPU register
5. Execute the instruction
6. Go to step 1 to begin executing following instruction

Interpreter (1)

```
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
      // of the memory image and the starting address.
                                  . . .
```

An interpreter for a simple computer (written in Java).

Interpreter (2)

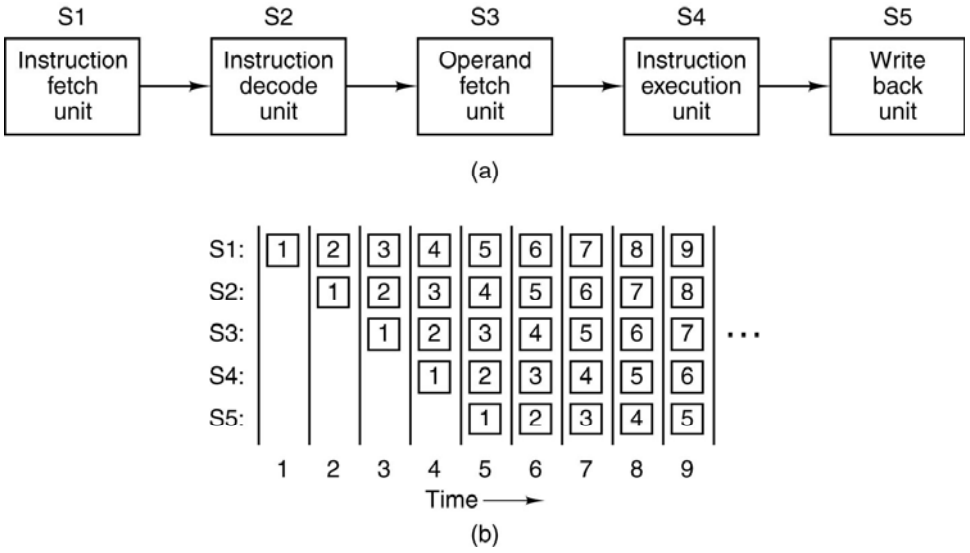
```
      PC = starting_address;
      while (run_bit) {
         instr = memory[PC];                    // fetch next instruction into instr
         PC = PC + 1;                           // increment program counter
         instr_type = get_instr_type(instr);    // determine instruction type
         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) { ... }
}
```

An interpreter for a simple computer (written in Java).

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, stores should reference memory
- Provide plenty of registers

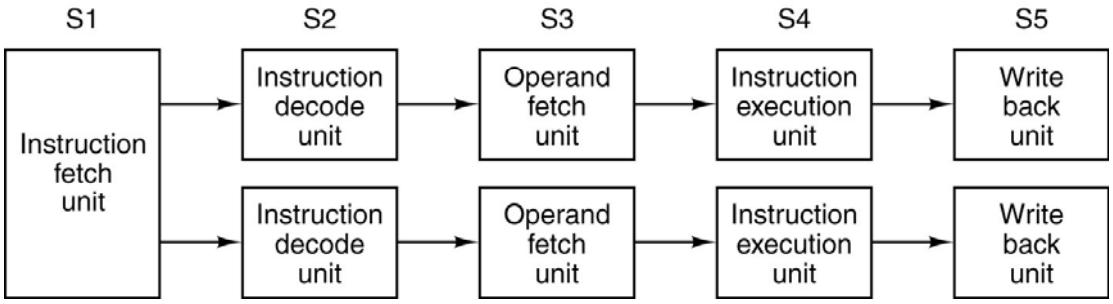
Instruction-Level Parallelism



a) A five-stage pipeline

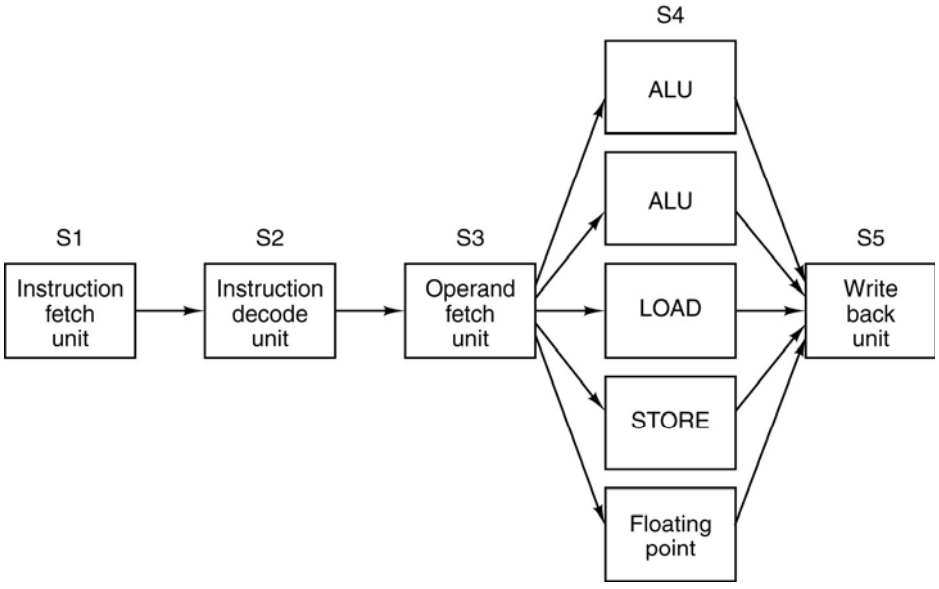
b) The state of each stage as a function of time. Nine clock cycles are illustrated

Superscalar Architectures (1)



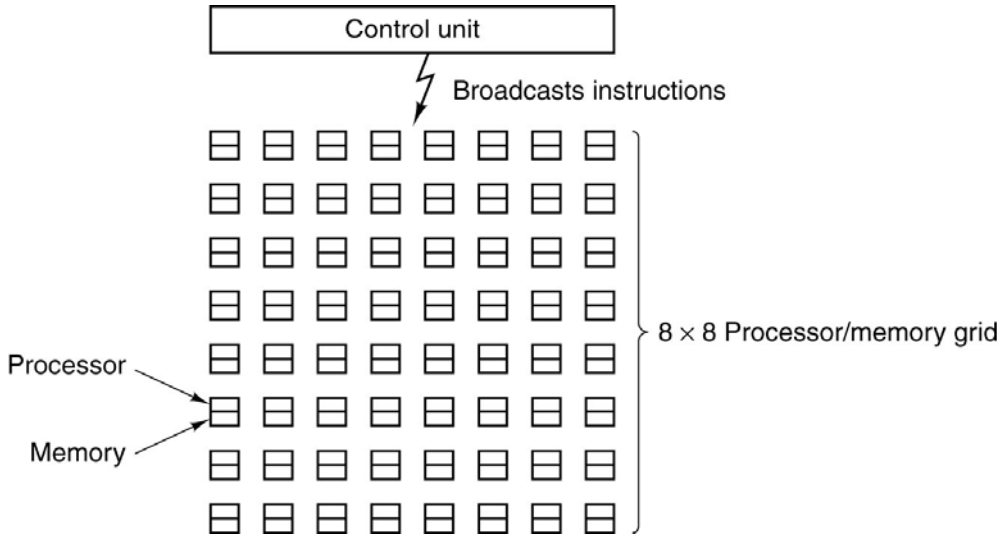
Dual five-stage pipelines with a common instruction fetch unit.

Superscalar Architectures (2)



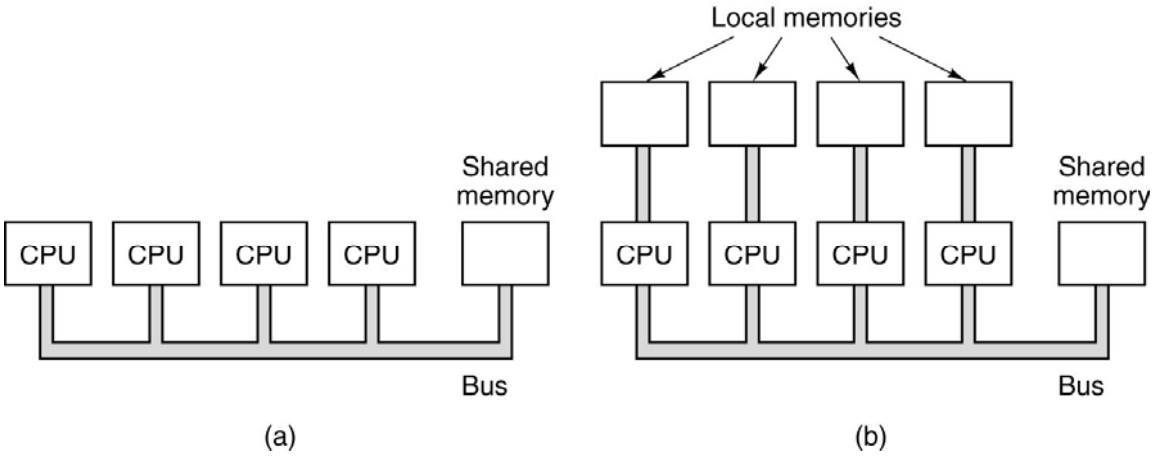
A superscalar processor with five functional units.

Processor-Level Parallelism (1)



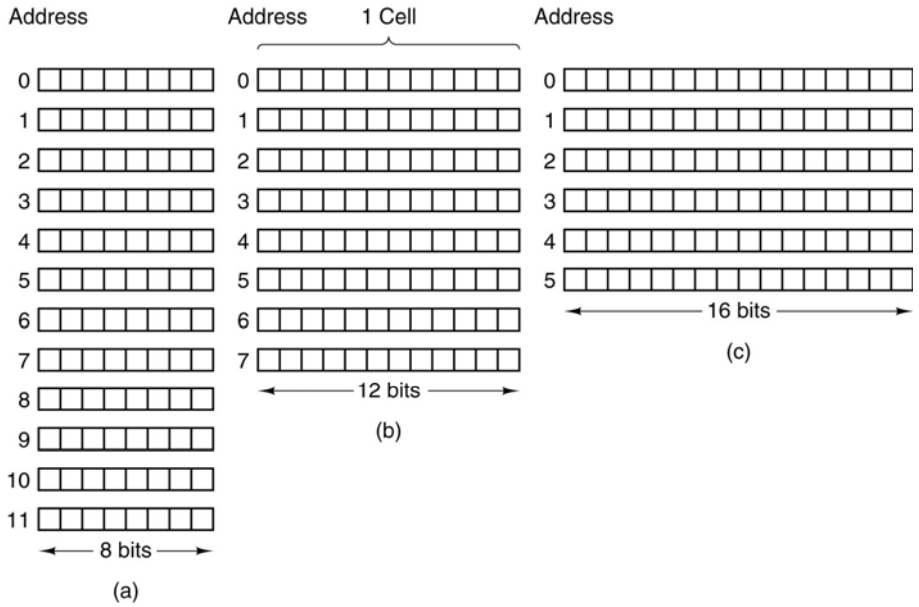
An array of processor of the ILLIAC IV type.

Processor-Level Parallelism (2)



a) A single-bus multiprocessor.
b) A multicomputer with local memories.

Primary Memory
Memory Addresses (1)



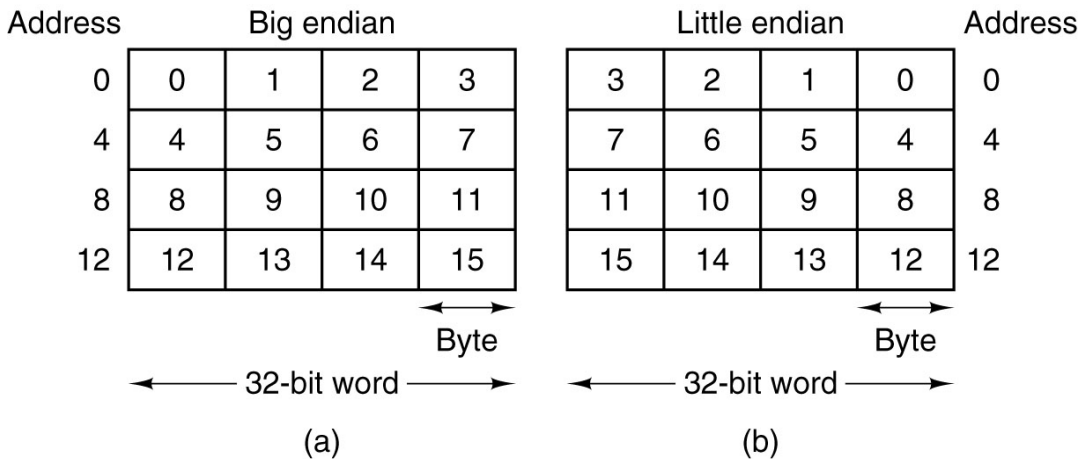
Three ways of organizing a 96-bit memory.

Primary Memory
Memory Addresses (2)

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

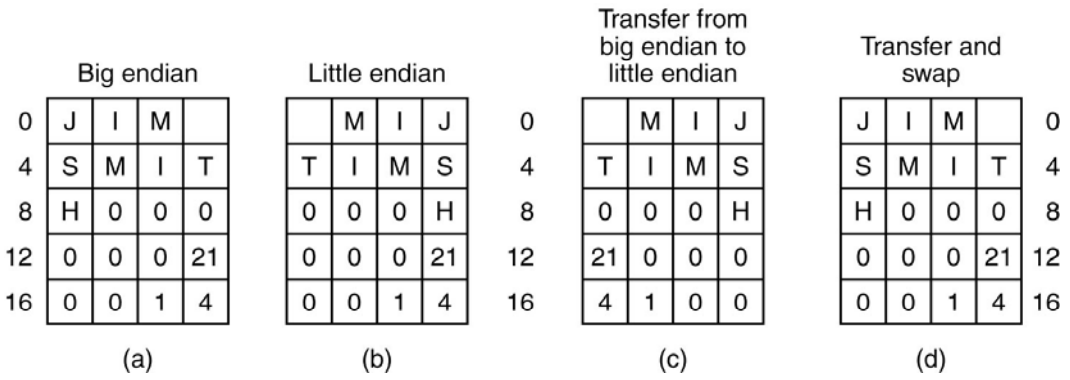
Number of bits per cell for some historically interesting commercial computers

Byte Ordering (1)



(a) Big endian memory (b) Little endian memory

Byte Ordering (2)



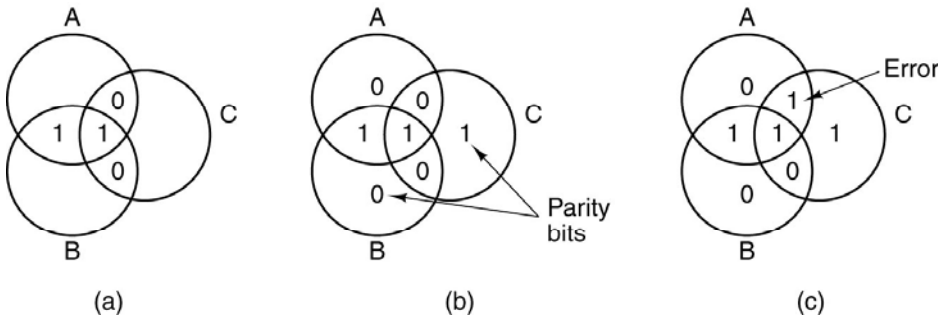
(a) A personal record for a big endian machine.
(b) The same record for a little endian machine.
(c) The result of transferring from big endian to little endian.
(d) The result of byte-swapping (c).

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

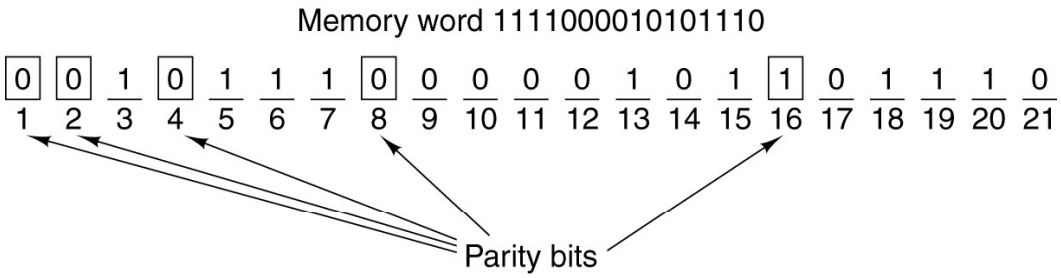
Number of check bits for a code that can correct a single error

Error Correcting Codes (2)



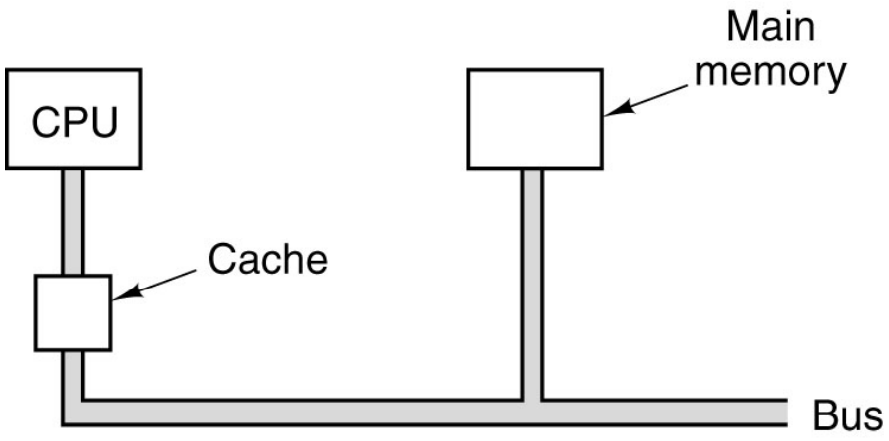
(a) Encoding of 1100
(b) Even parity added
(c) Error in AC

Error Correcting Codes (3)



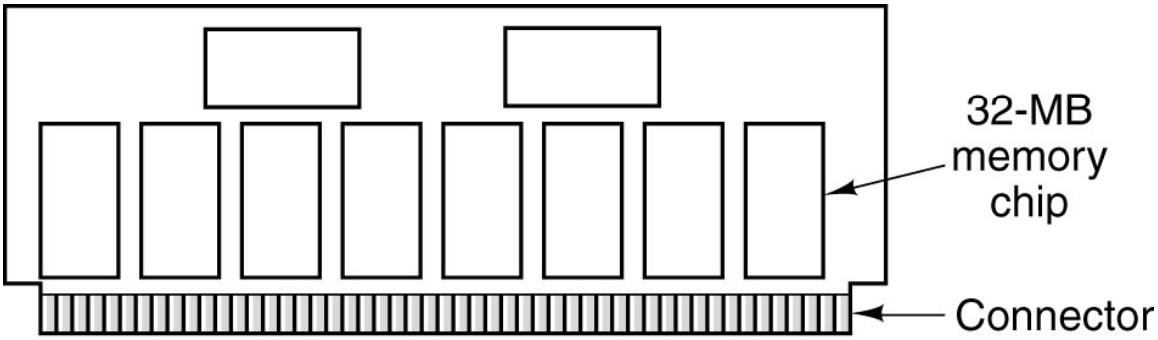
Construction of the Hamming code for the memory word 1111000010101110 by adding 5 check bits to the 16 data bits.

Cache Memory



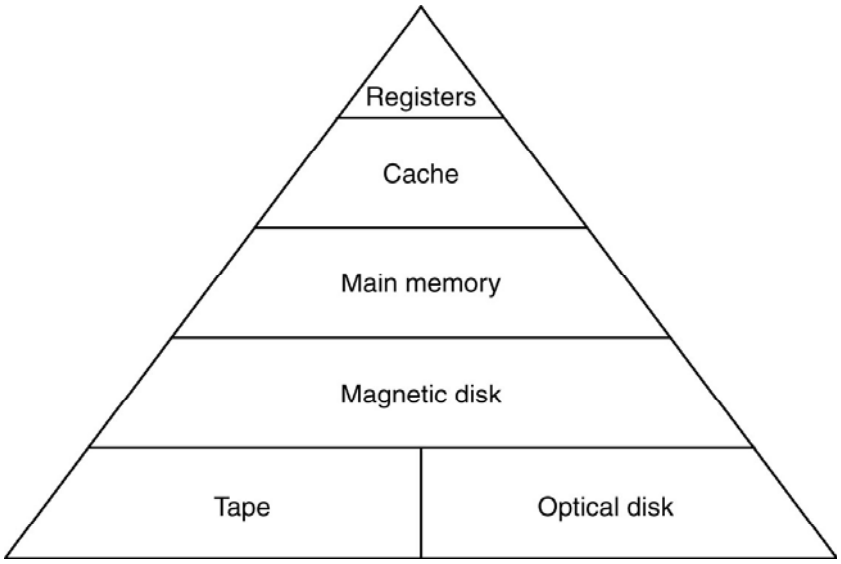
The cache is logically between the CPU and main memory. Physically, there are several possible places it could be located.

Memory Packaging and Types



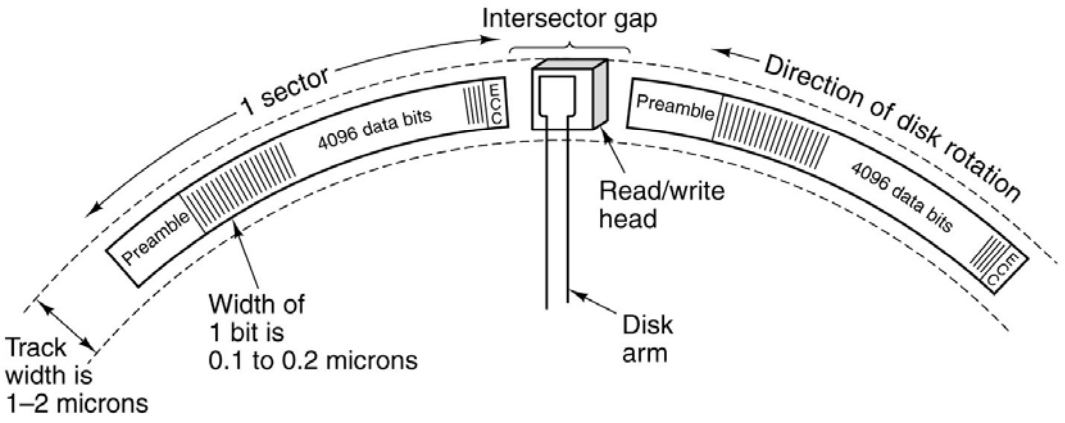
A single inline memory module (SIMM) holding 256 MB. Two of the chips control the SIMM.

Memory Hierarchies



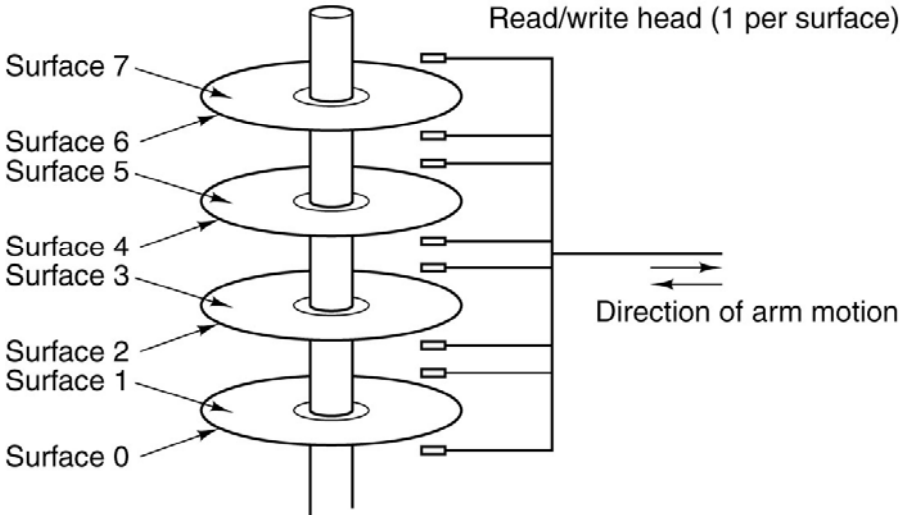
A five-level memory hierarchy.

Magnetic Disks (1)



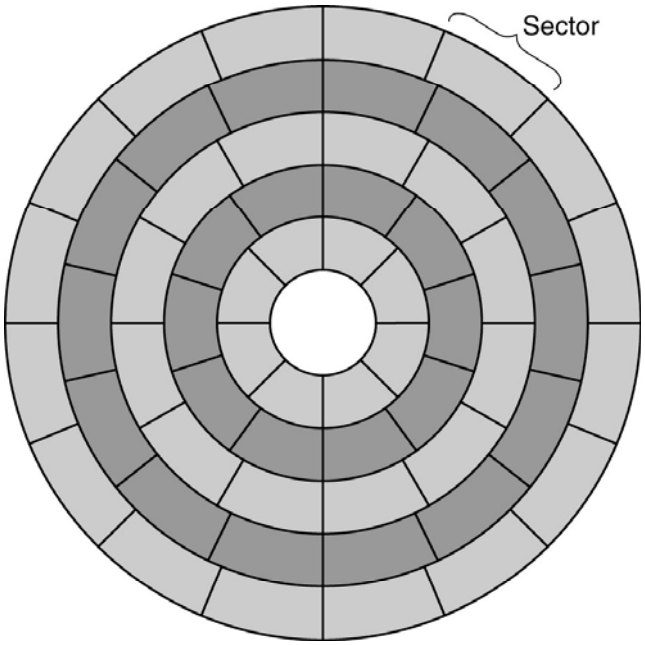
A portion of a disk track. Two sectors are illustrated.

Magnetic Disks (2)



A disk with four platters.

Magnetic Disks (3)



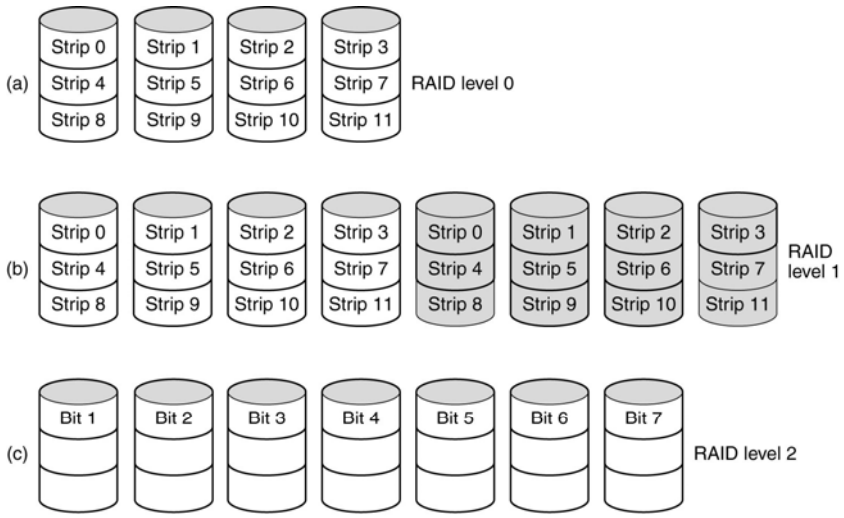
A disk with five zones. Each zone has many tracks.

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
Ultra3 SCSI	8	80	80
Wide Ultra3 SCSI	16	80	160
Ultra4 SCSI	8	160	160
Wide Ultra4 SCSI	16	160	320

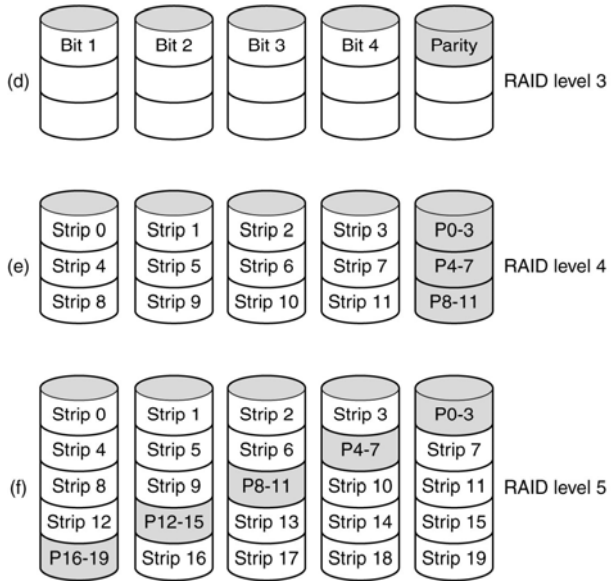
Some of the possible SCSI parameters.

RAID (1)
Redundant Array of Inexpensive Disks



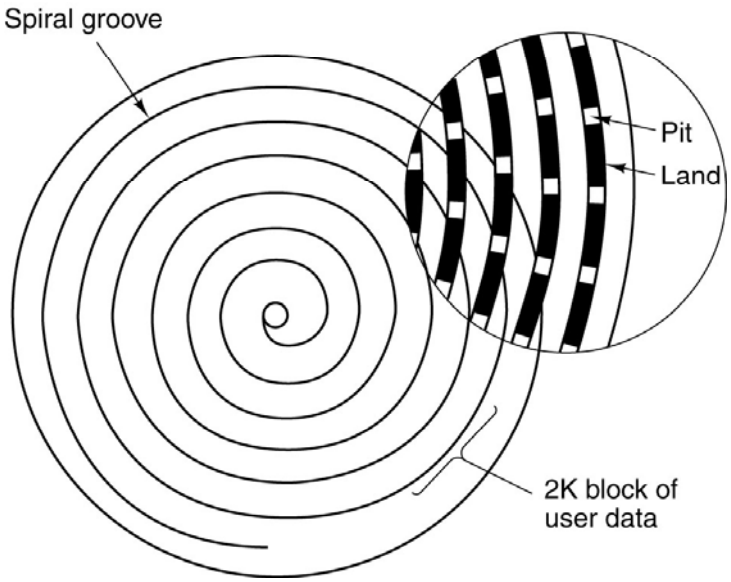
RAID levels 0 through 2.
Backup and parity disks are shown shaded.

RAID (1)
Redundant Array of Inexpensive Disks



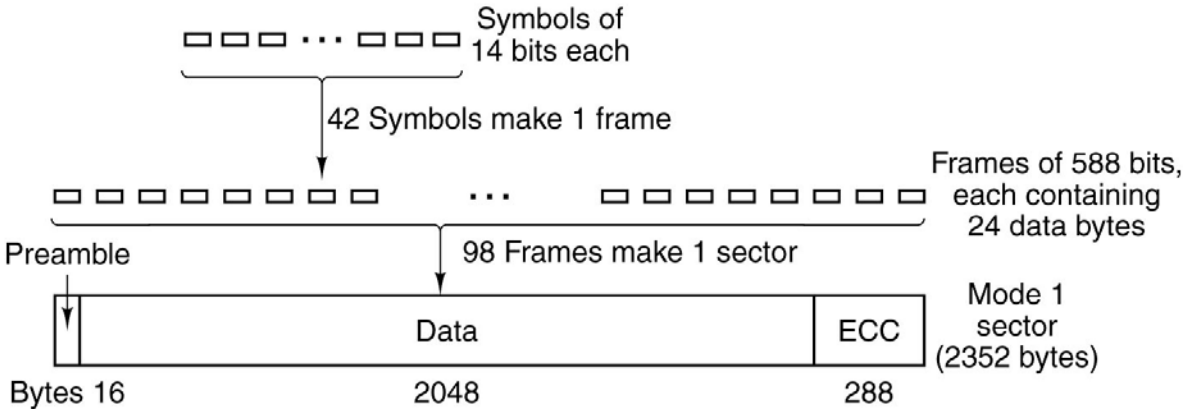
RAID levels 3 through 5.
Backup and parity disks are shown shaded.

CD-ROMs (1)



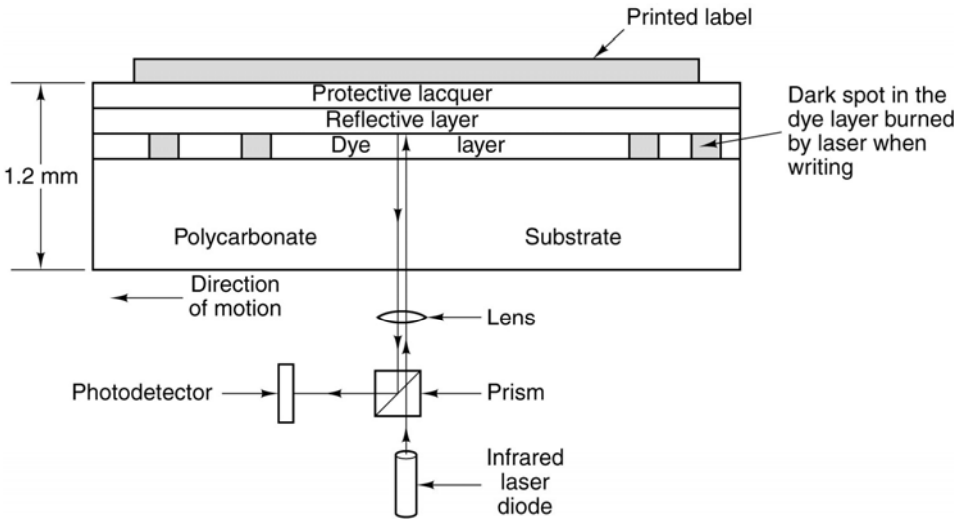
Recording structure of a Compact Disk or CD-ROM.

CD-ROMs (2)



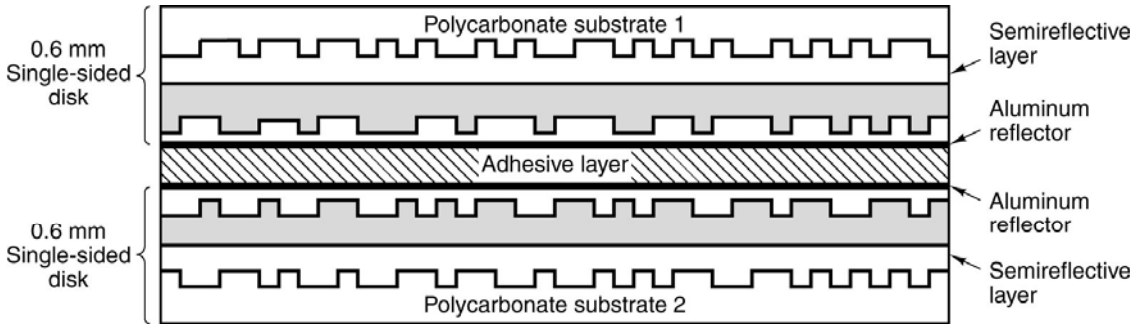
Logical data layout on a CD-ROM.

CD-Recordables



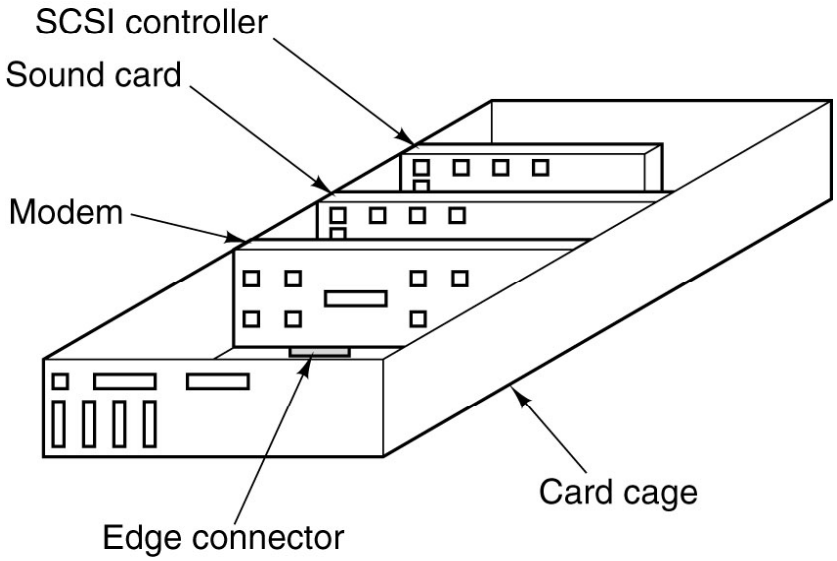
Cross section of a CD-R disk and laser (not to scale). A CD-ROM has a similar structure, except without the dye layer and with a pitted aluminum layer instead of a reflective layer.

DVD



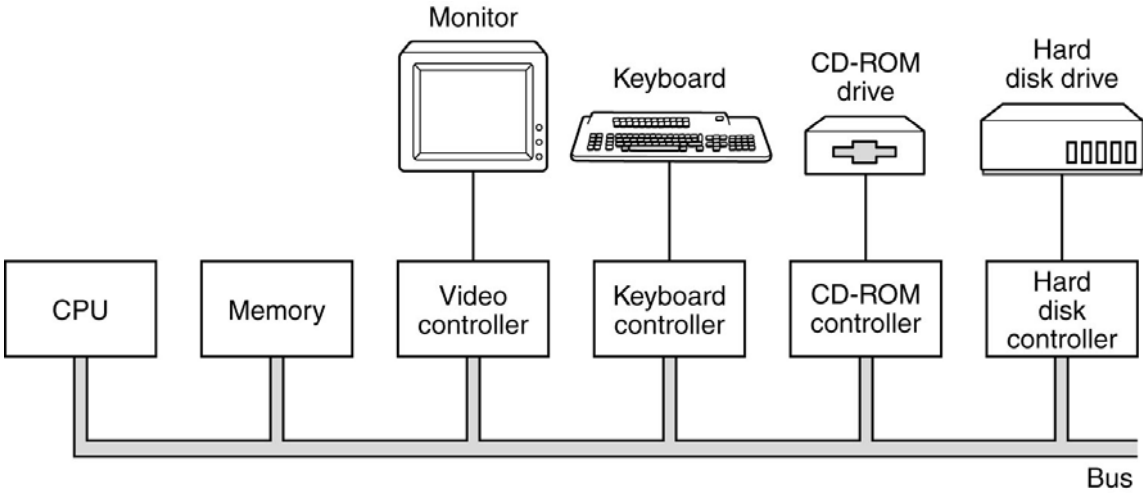
A double-sided, dual layer DVD disk.

Input/Output
Buses (1)



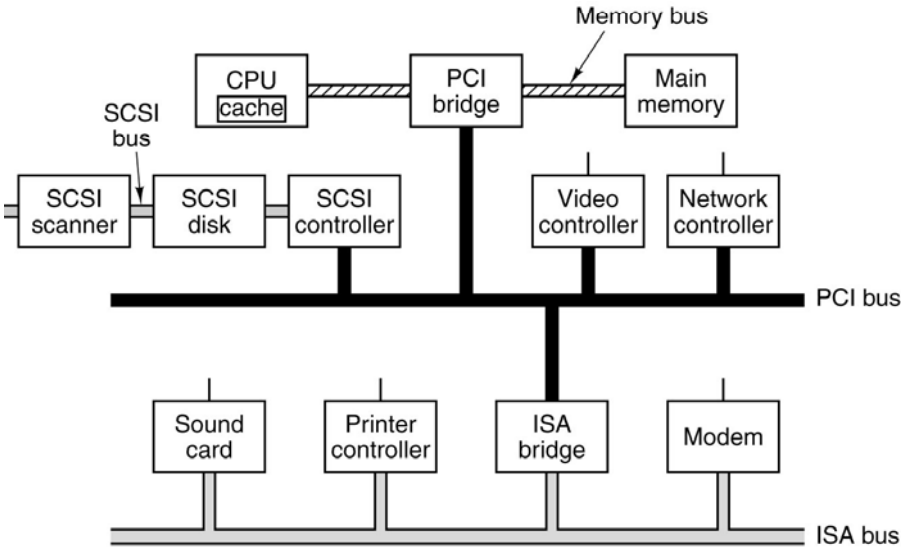
Physical structure of a personal computer.

Input/Output
Buses (2)



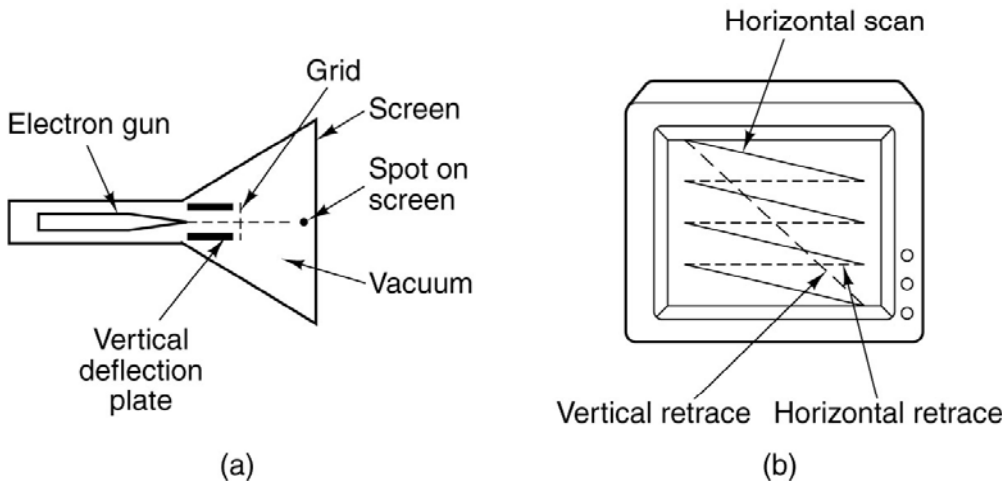
Logical structure of a simple personal computer.

Input/Output
Buses (3)



A typical modern PC with a PCI bus and an ISA bus.

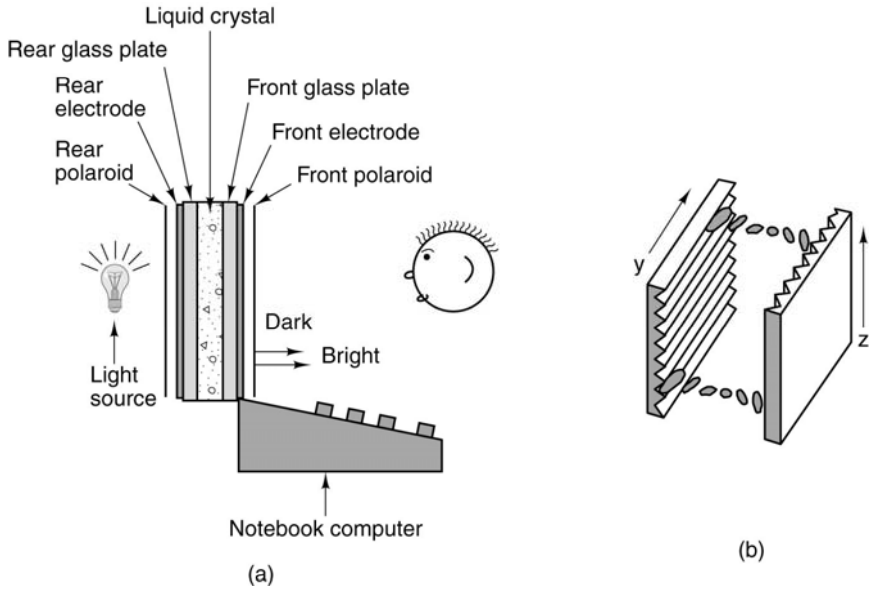
CRT Monitors



(a) Cross section of a CRT

(b) CRT scanning pattern

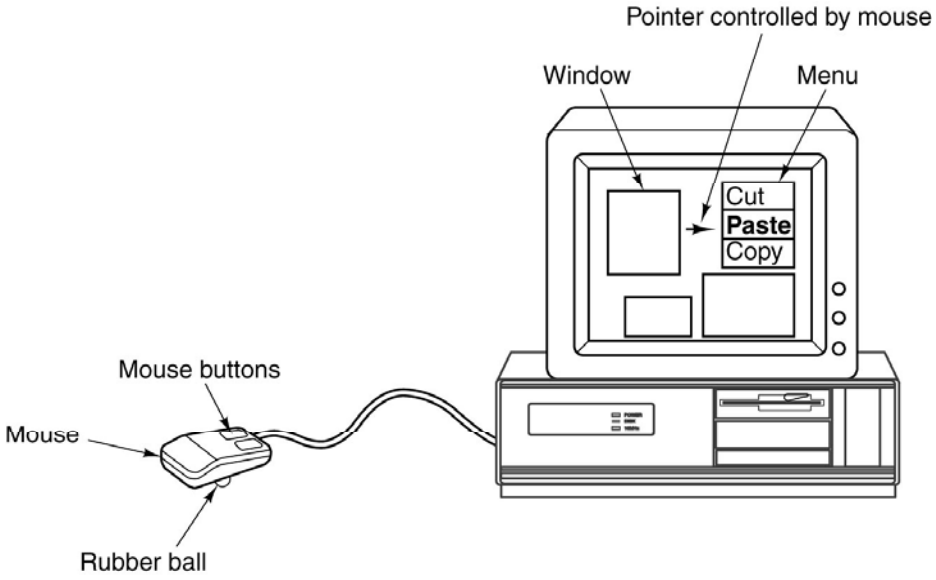
Flat Panel Displays



(a) The construction of an LCD screen.

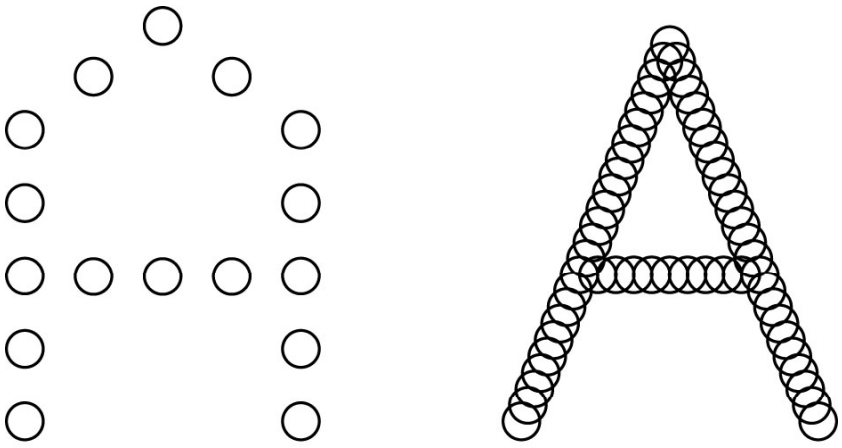
(b) The grooves on the rear and front plates are perpendicular to one another.

Mice



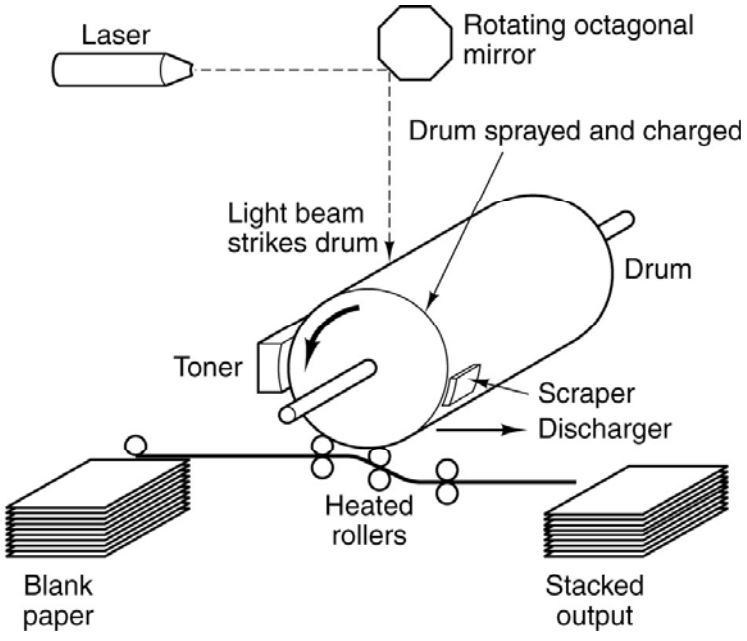
A mouse being used to point to menu items.

Printers (1)



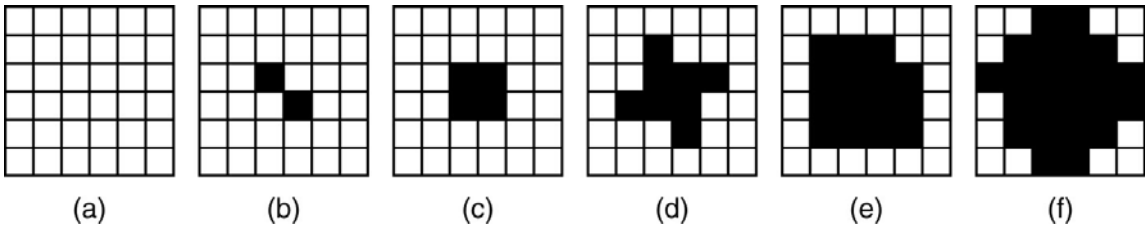
(a) The letter “A” on a 5 x 7 matrix.
(b) The letter “A” printed with 24 overlapping needles.

Printers (2)



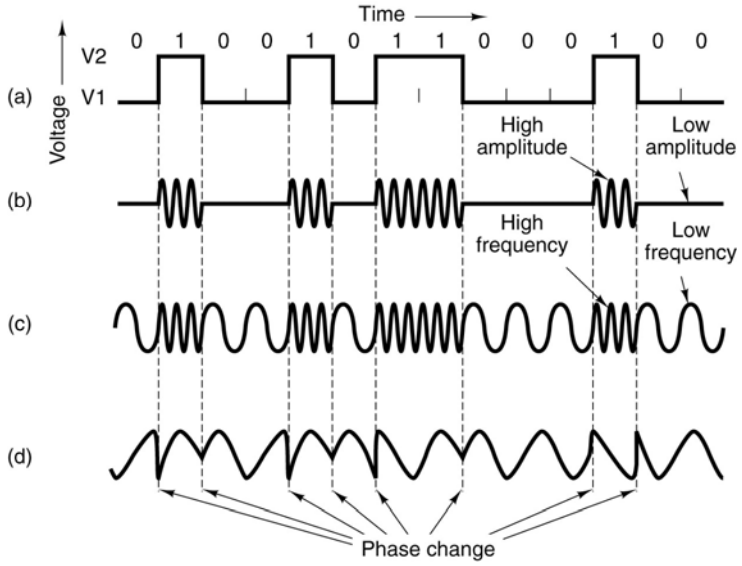
Operation of a laser printer.

Printers (3)



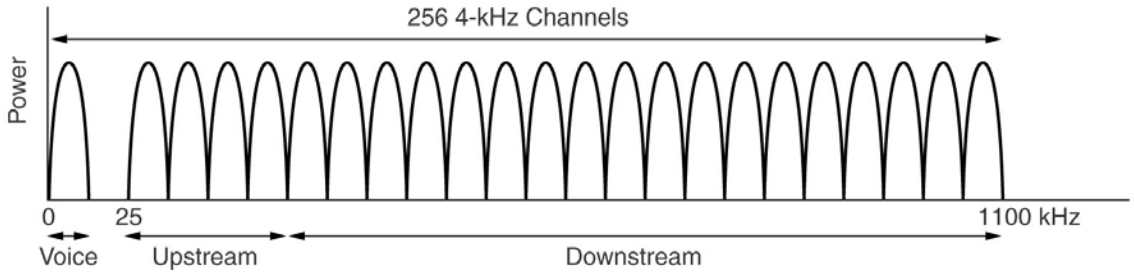
Halftone dots for various gray scale ranges.
(a) 0 – 6. (b) 14 – 20. (c) 28 – 34.
(d) 56 – 62. (e) 105 – 111. (f) 161 – 167.

Telecommunications



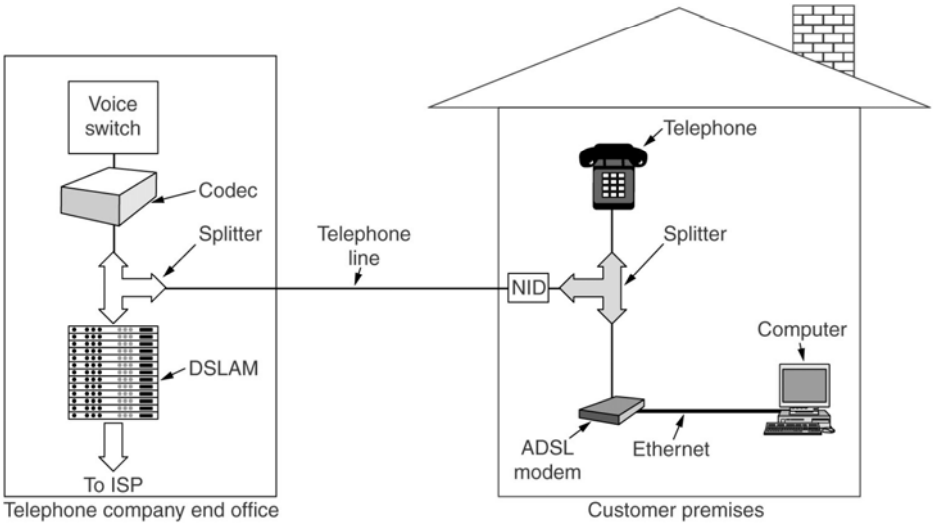
Transmission of the binary number 01001010000100 over a telephone line bit by bit. (a) Two-level signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

Digital Subscriber Lines (1)



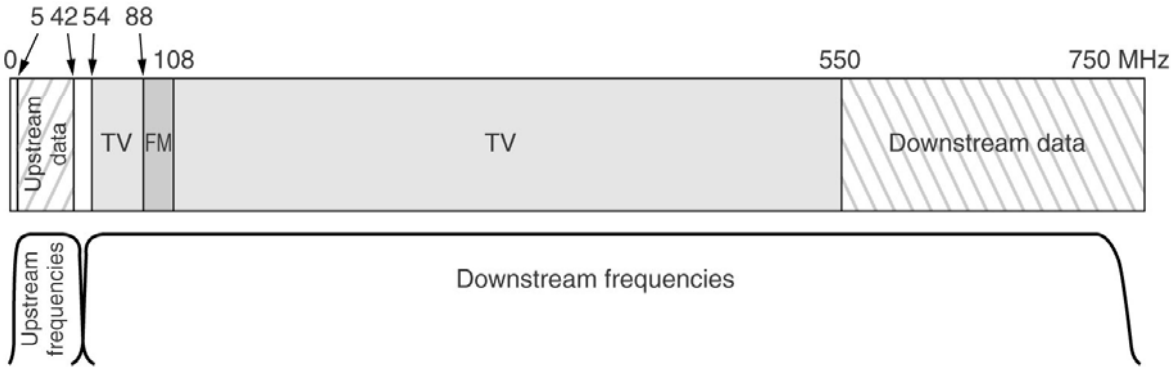
Operation of ADSL.

Digital Subscriber Lines (2)



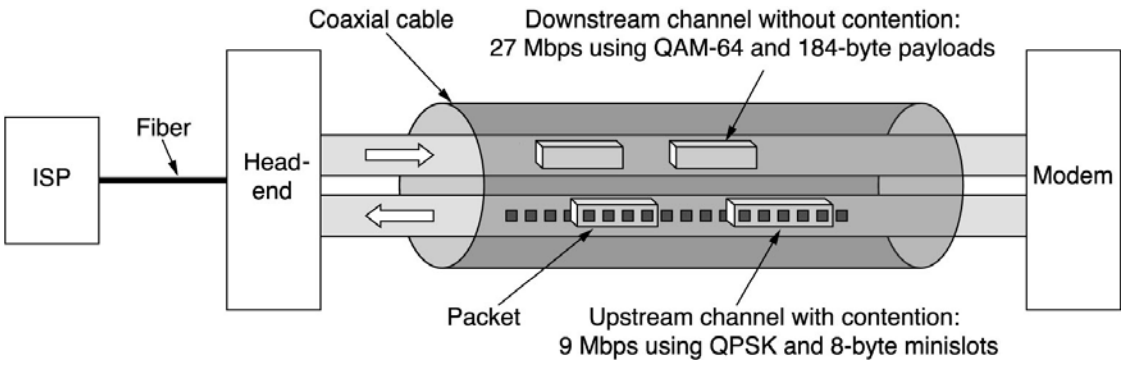
A typical ADSL equipment configuration.

Internet over Cable (1)



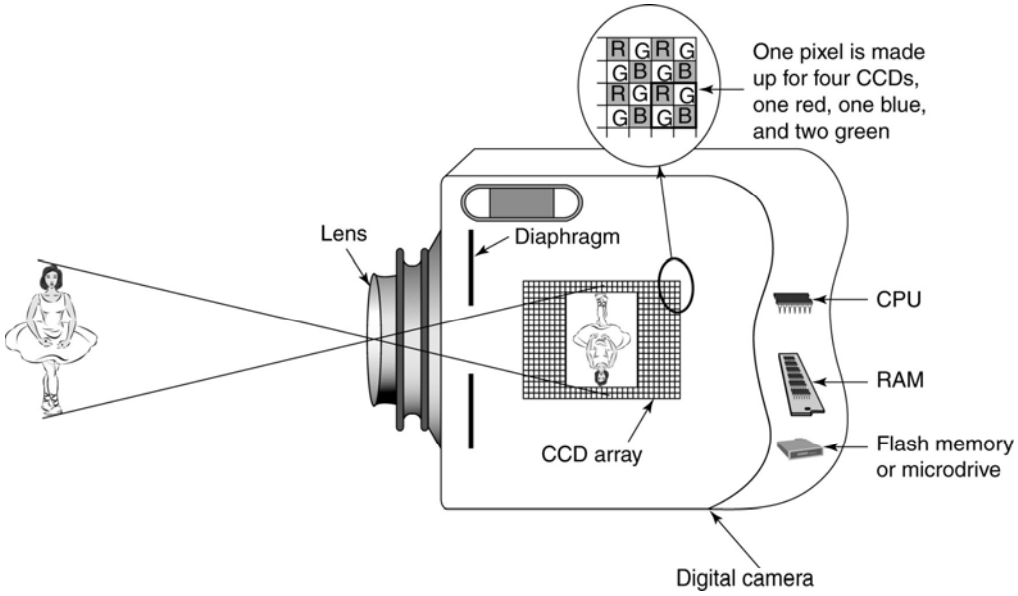
Frequency allocation in a typical cable TV system used for Internet access

Internet over Cable (2)



Typical details of the upstream and downstream channels in North America. QAM-64 (Quadrature Amplitude Modulation) allows 6 bits/Hz but only works at high frequencies. QPSK (Quadrature Phase Shift Keying) works at low frequencies but allows only 2 bits/Hz.

Digital Cameras



A digital camera.

ASCII 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

The ASCII Character set: characters 0 – 31.

ASCII 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

The ASCII Character set: characters 32 – 127.