2

COMPUTER SYSTEMS ORGANIZATION

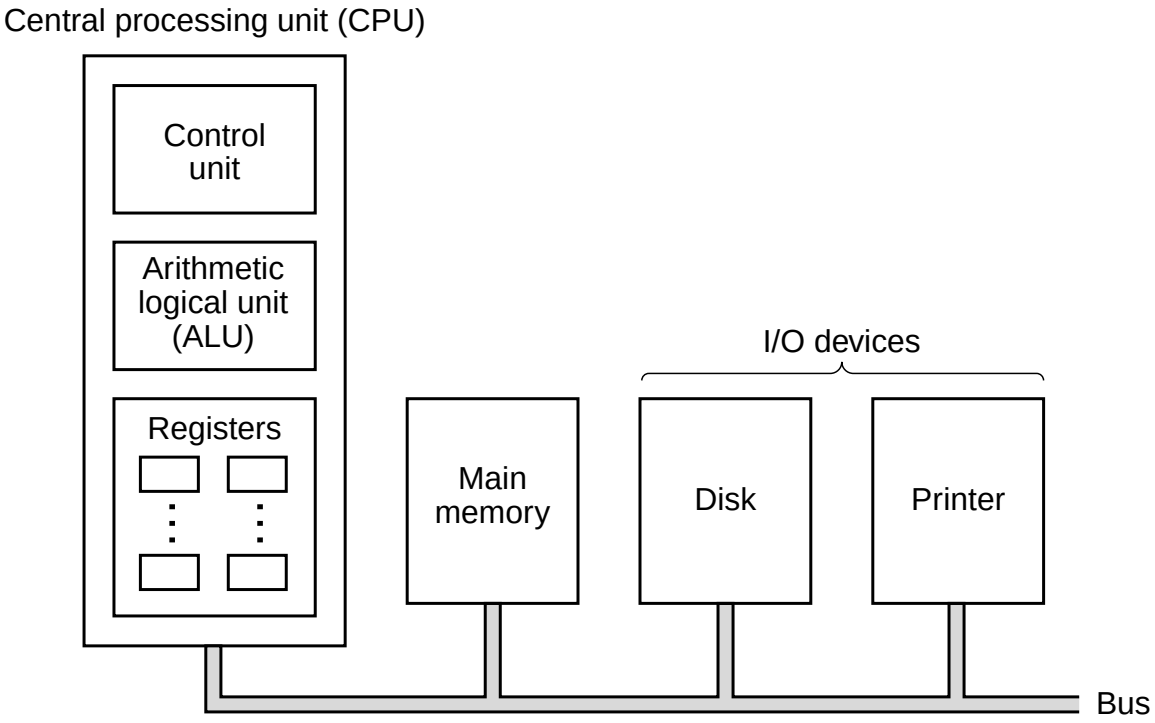

Figure 2-1. The organization of a simple computer with one CPU and two I/O devices.

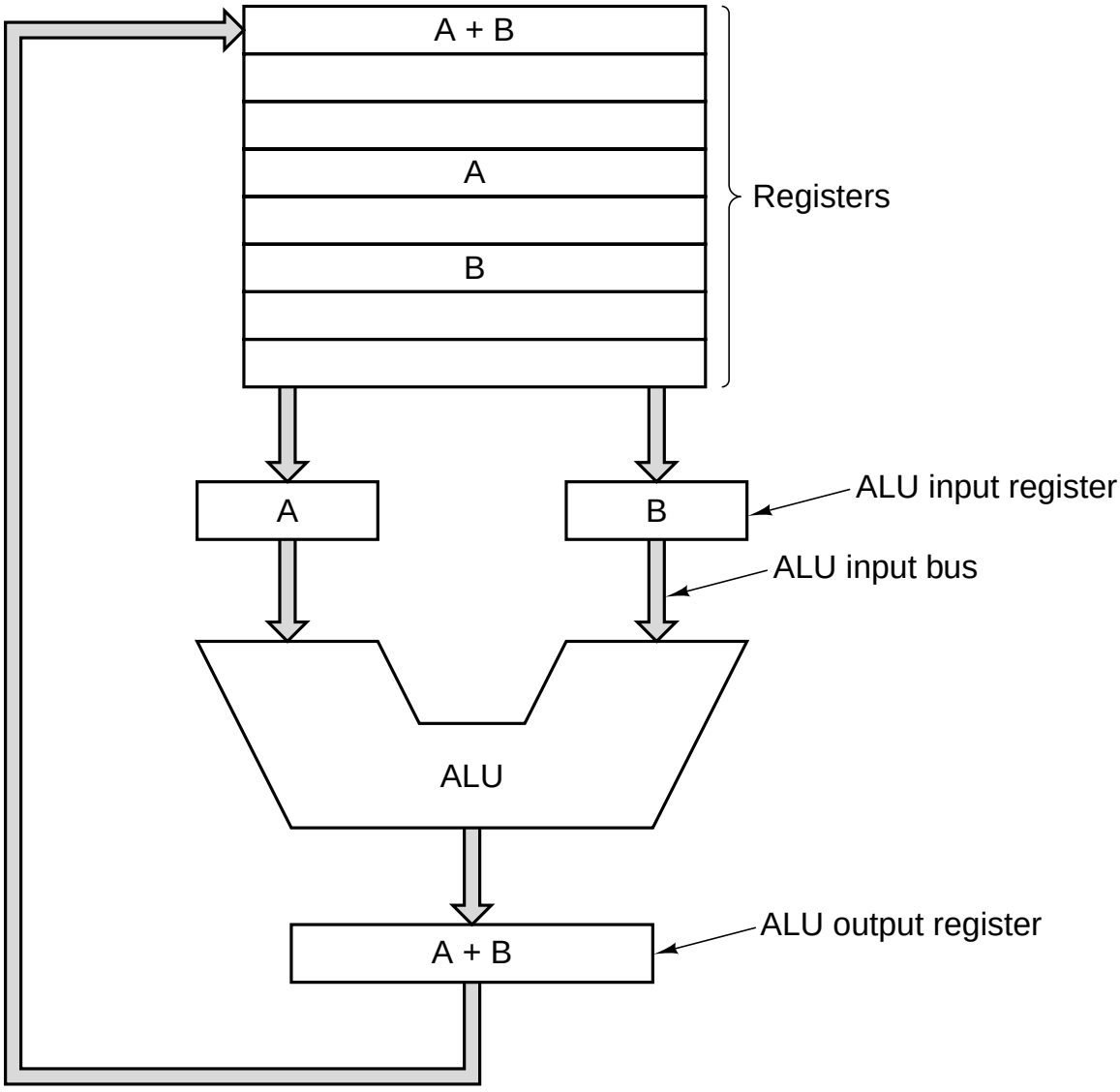

Figure 2-2. The data path of a typical von Neumann machine.

```
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 am 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.

      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).

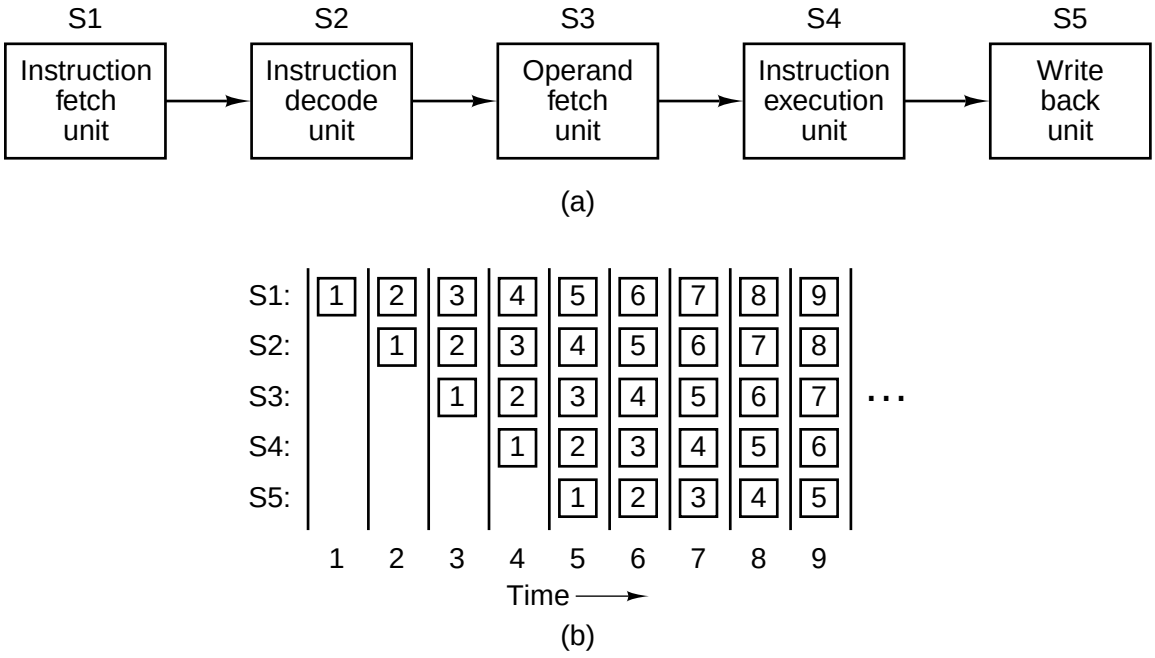

Figure 2-4. (a) A five-stage pipeline. (b) The state of each stage as a function of time. Nine clock cycles are illustrated.

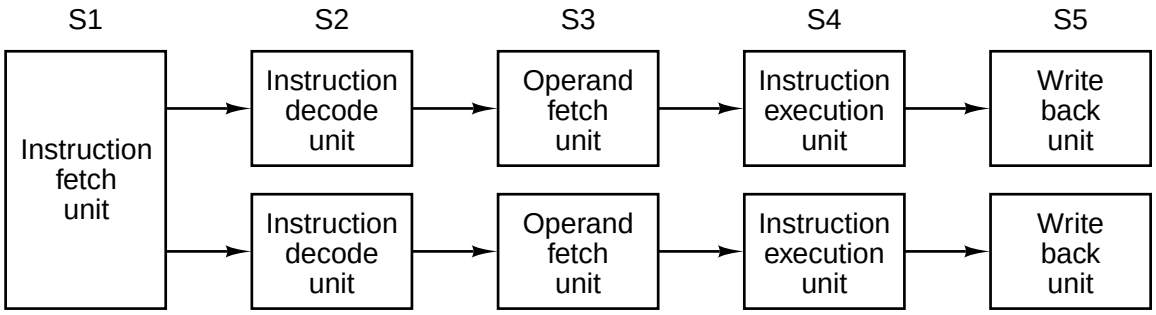

Figure 2-5. (a) Dual five-stage pipelines with a common instruction fetch unit.

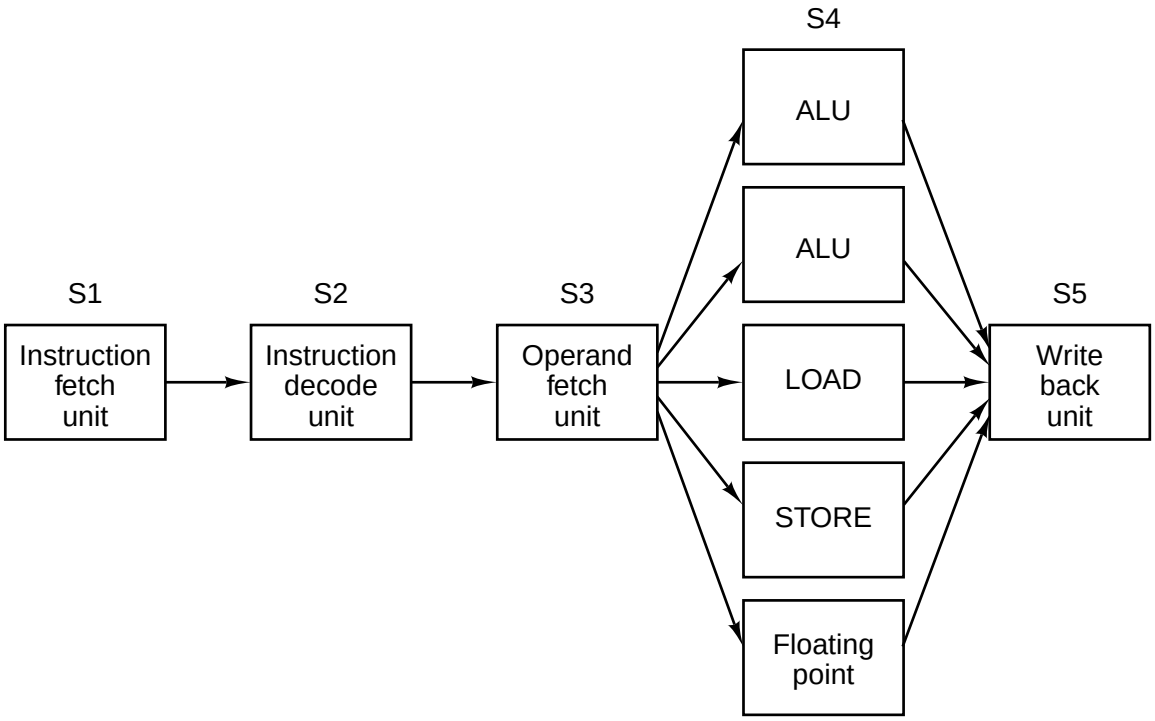

Figure 2-6. A superscalar processor with five functional units.

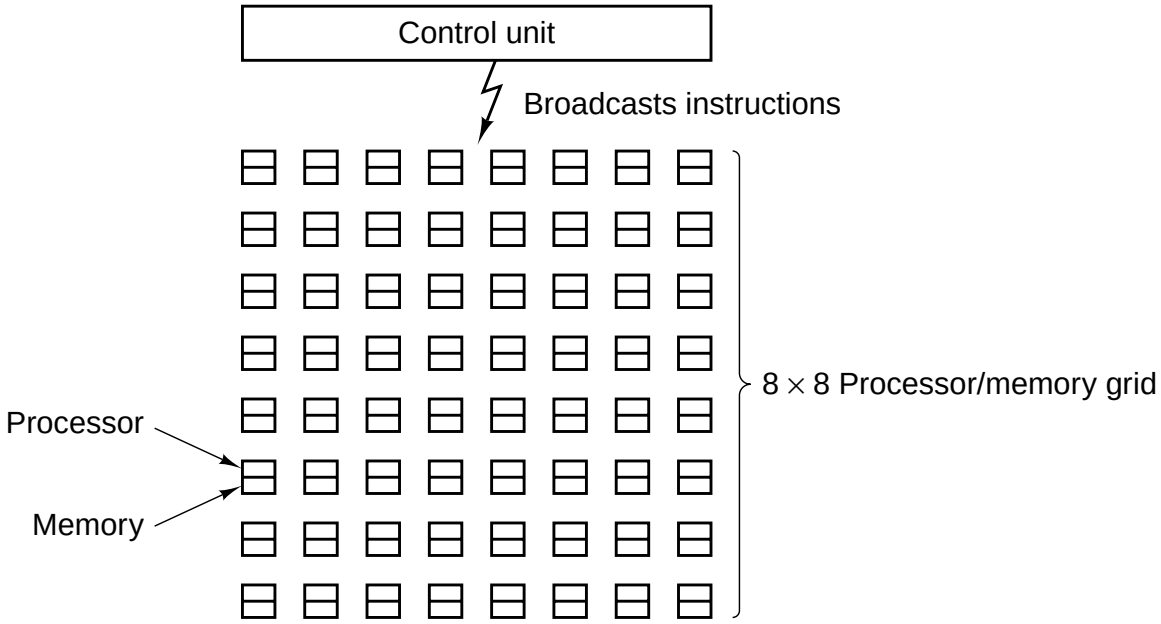

Figure 2-7. An array processor of the ILLIAC IV type.

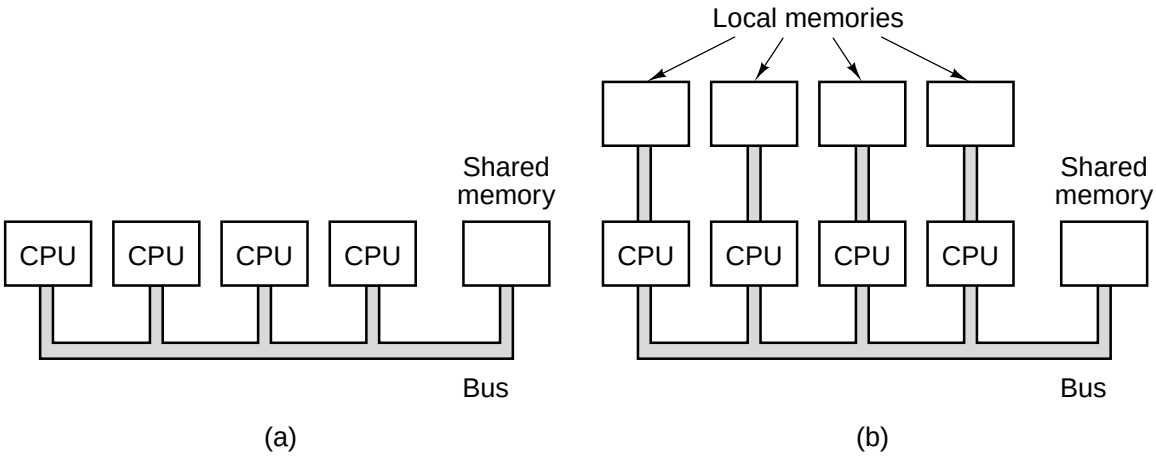

Figure 2-8. (a) A single-bus multiprocessor. (b) A multicomputer with local memories.

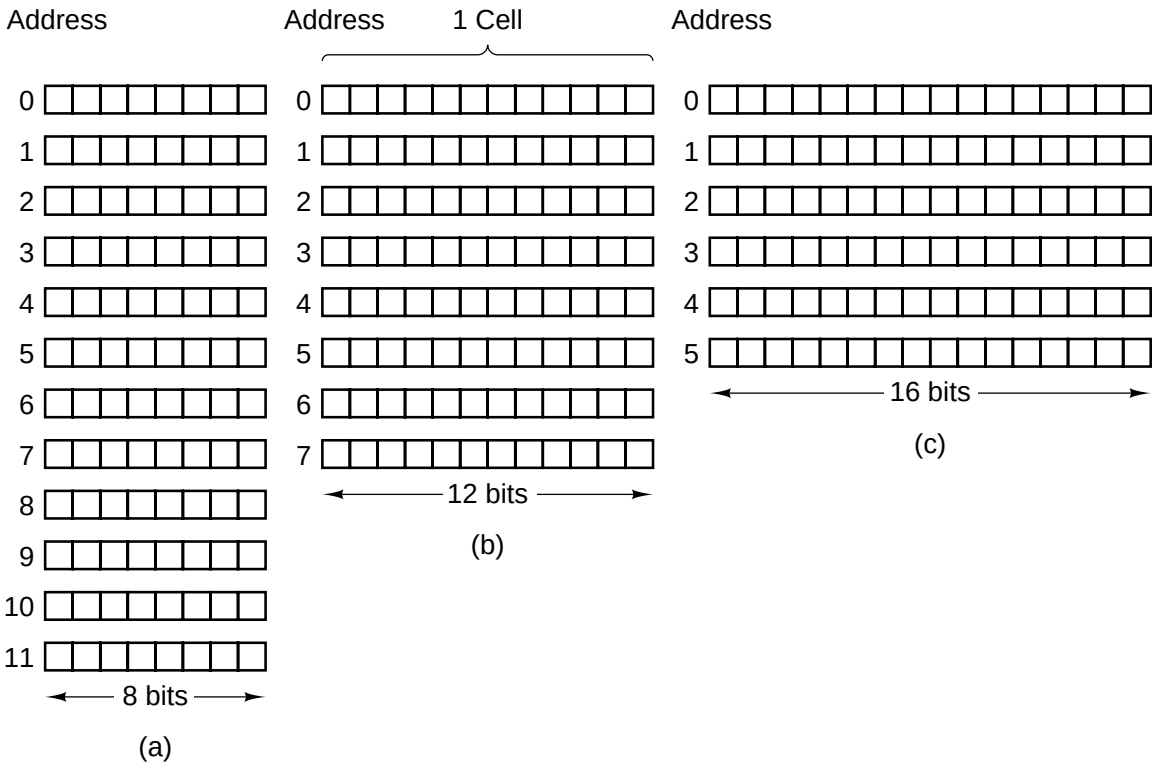

Figure 2-9. Three ways of organizing a 96-bit memory.

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.

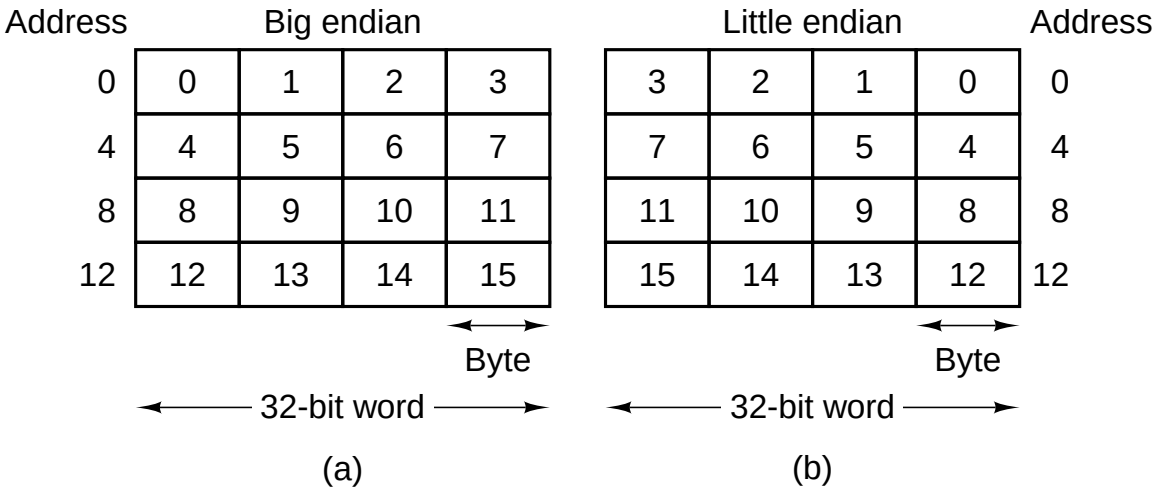

Figure 2-11. (a) Big endian memory. (b) Little endian memory.

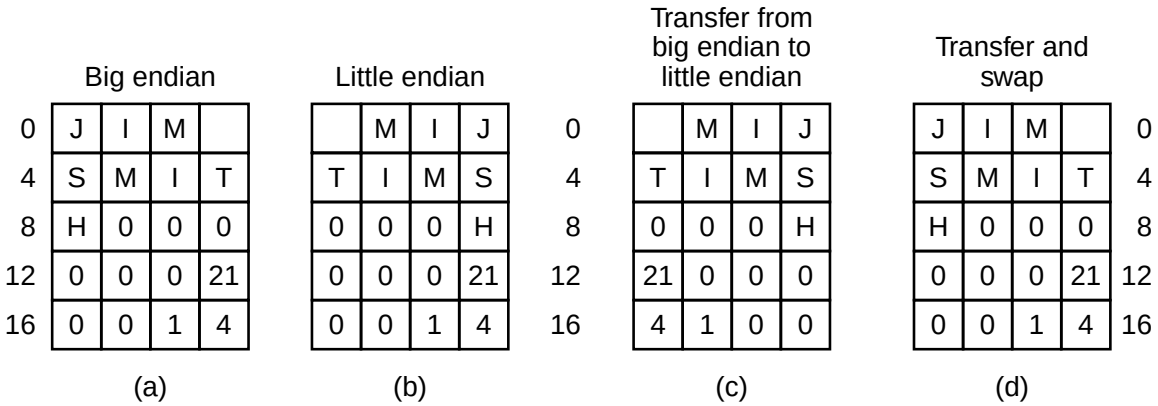

Figure 2-12. (a) A personnel record for a big endian machine. (b) The same record for a little endian machine. (c) The result of transferring the record from a big endian to a little endian. (d) The result of byte-swapping (c).

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.

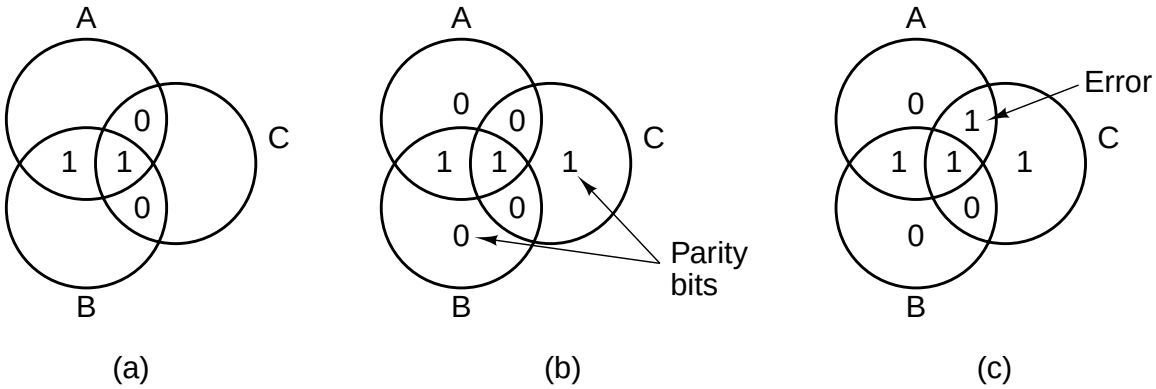

Figure 2-14. (a) Encoding of 1100. (b) Even parity added. (c) Error in *AC*.

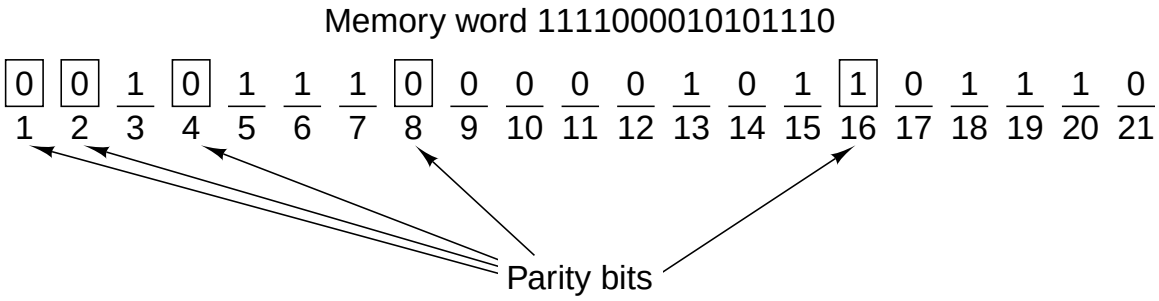

Figure 2-15. Construction of the Hamming code for the memory word 1111000010101110 by adding 5 check bits to the 16 data bits.

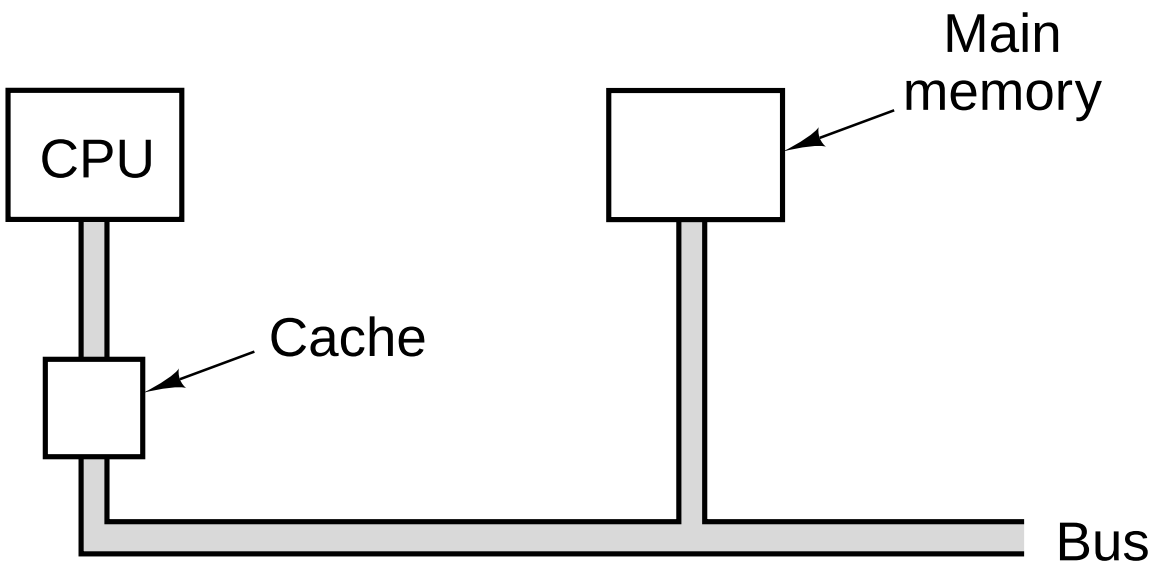

Figure 2-16. The cache is logically between the CPU and main memory. Physically, there are several possible places it could be located.

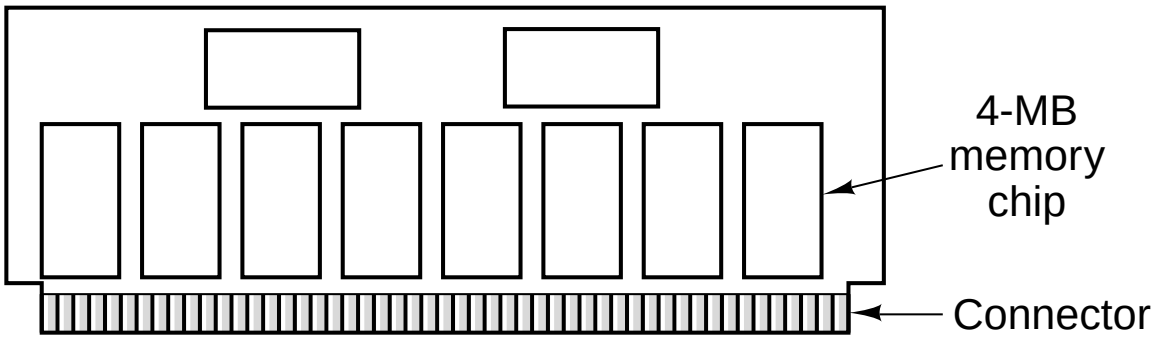

Figure 2-17. A single inline memory module (SIMM) holding 32 MB. Two of the chips control the SIMM.

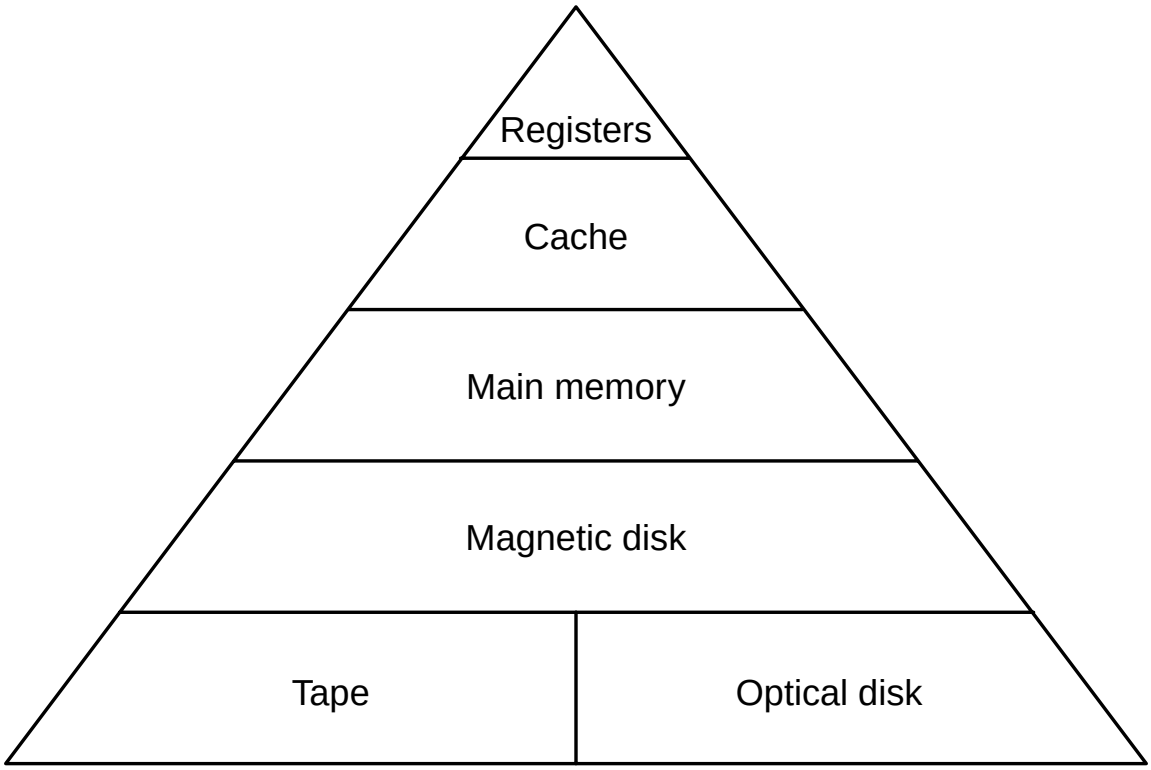

Figure 2-18. A five-level memory hierarchy.

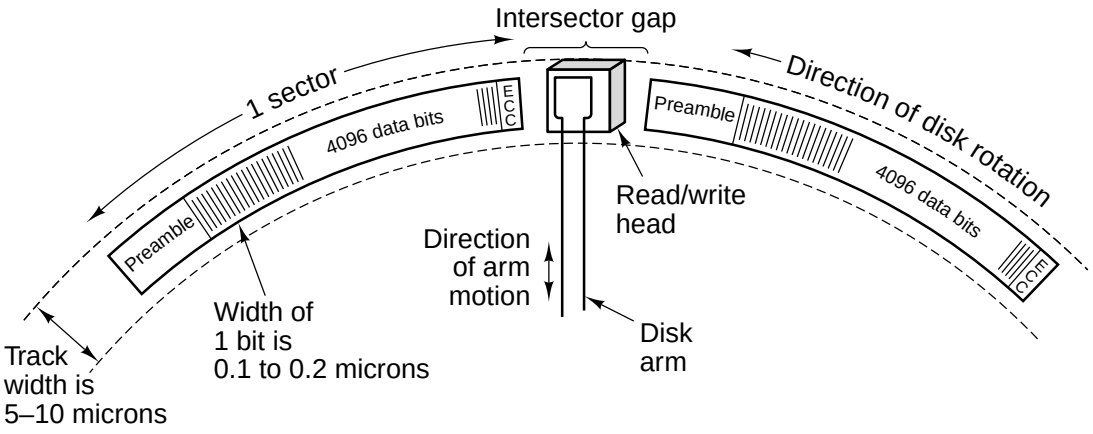

Figure 2-19. A portion of a disk track. Two sectors are illustrated.

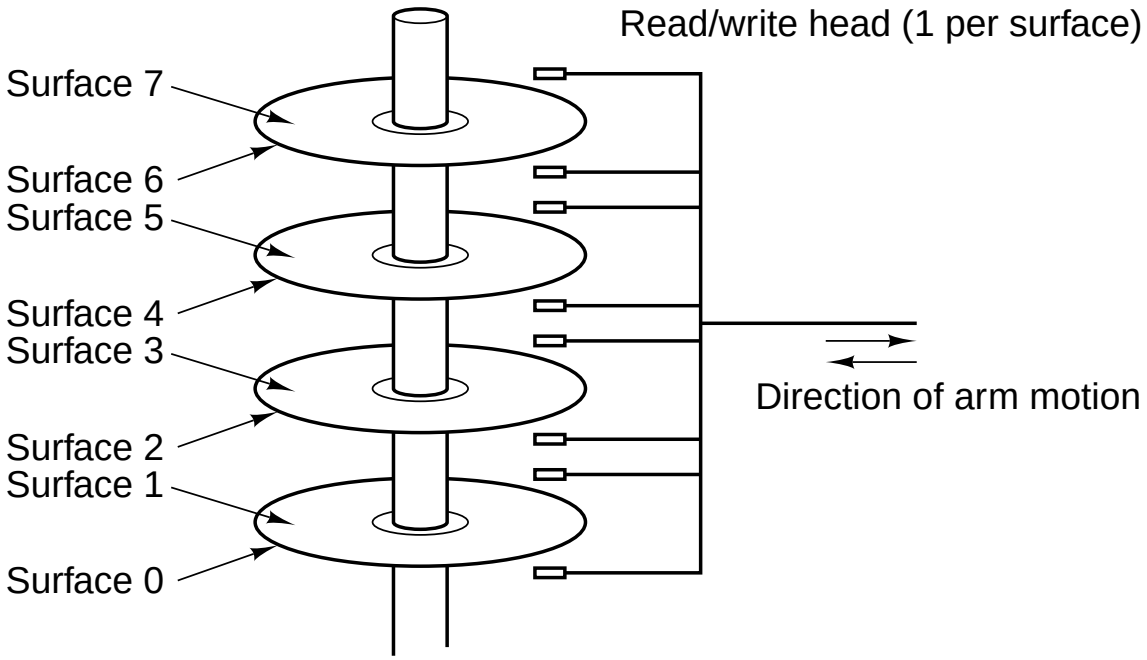

Figure 2-20. A disk with four platters.

Parameters	LD 5.25″	HD 5.25″	LD 3.5″	HD 3.5″
Size (inches)	5.25	5.25	3.5	3.5
Capacity (bytes)	360K	1.2M	720K	1.44M
Tracks	40	80	80	80
Sectors/track	9	15	9	18
Heads	2	2	2	2
Rotations/min	300	360	300	300
Data rate (kbps)	250	500	250	500
Type	Flexible	Flexible	Rigid	Rigid

Figure 2-21. Characteristics of the four kinds of floppy disks.

Name	Data bits	Bus MHz	MB/sec
SCSI-1	8	5	5
SCSI-2	8	5	5
Fast SCSI-2	8	10	10
Fast & wide SCSI-2	16	10	20
Ultra SCSI	16	20	40

Figure 2-22. Some of the possible SCSI parameters.

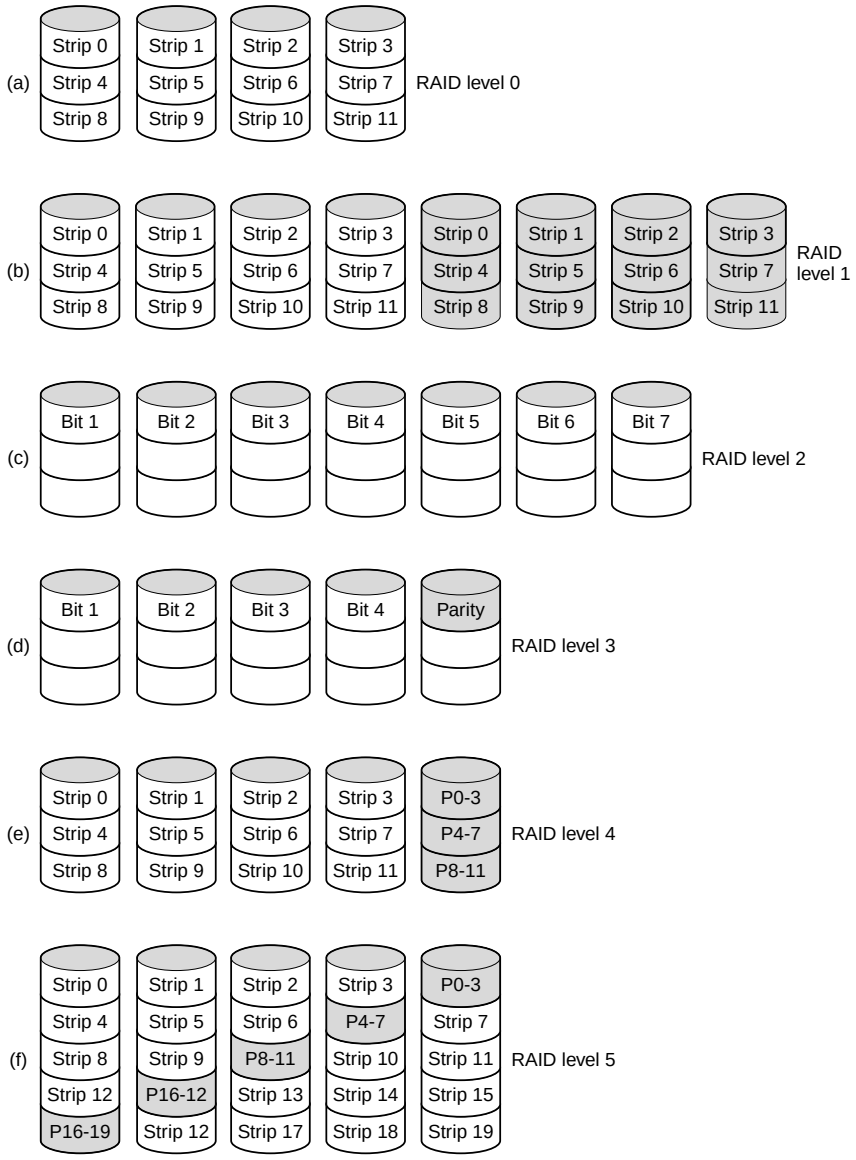

Figure 2-23. RAID levels 0 through 5. Backup and parity drives are shown shaded.

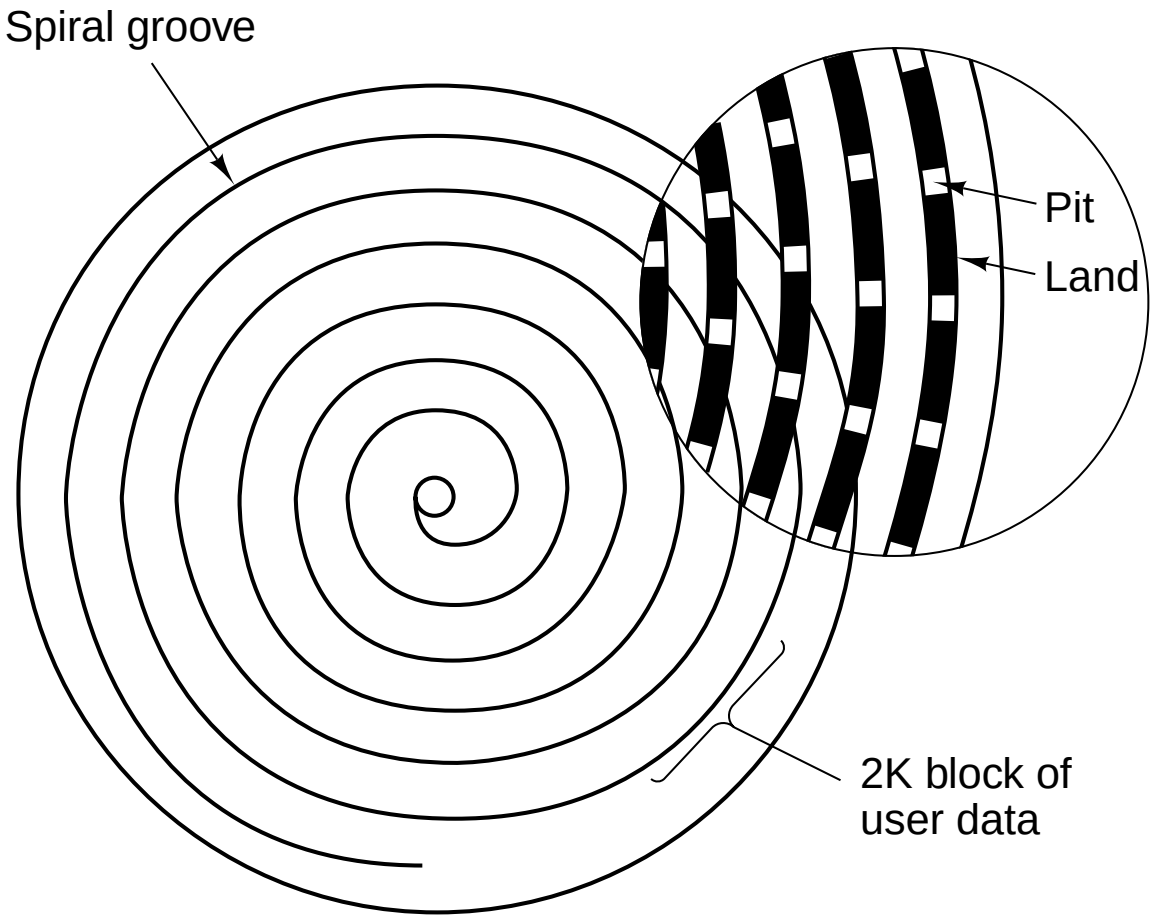

Figure 2-24. Recording structure of a Compact Disc or CD-ROM.

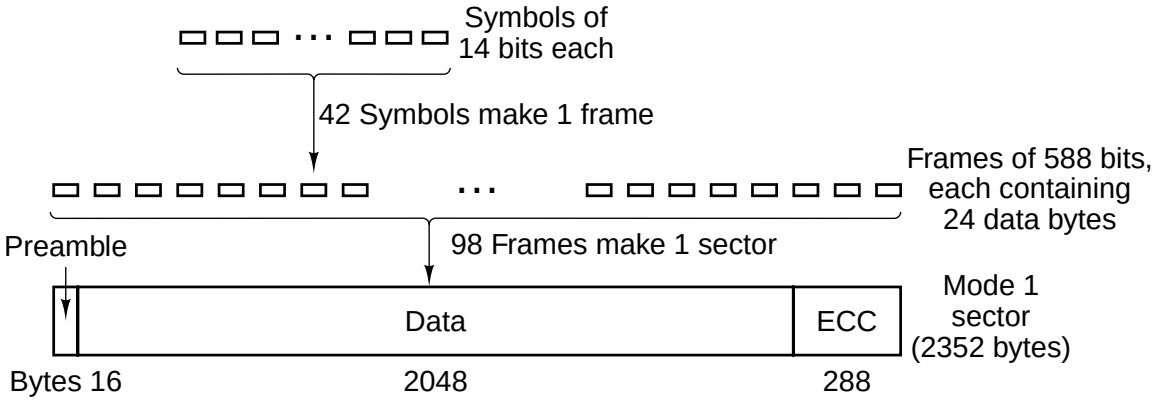

Figure 2-25. Logical data layout on a CD-ROM.

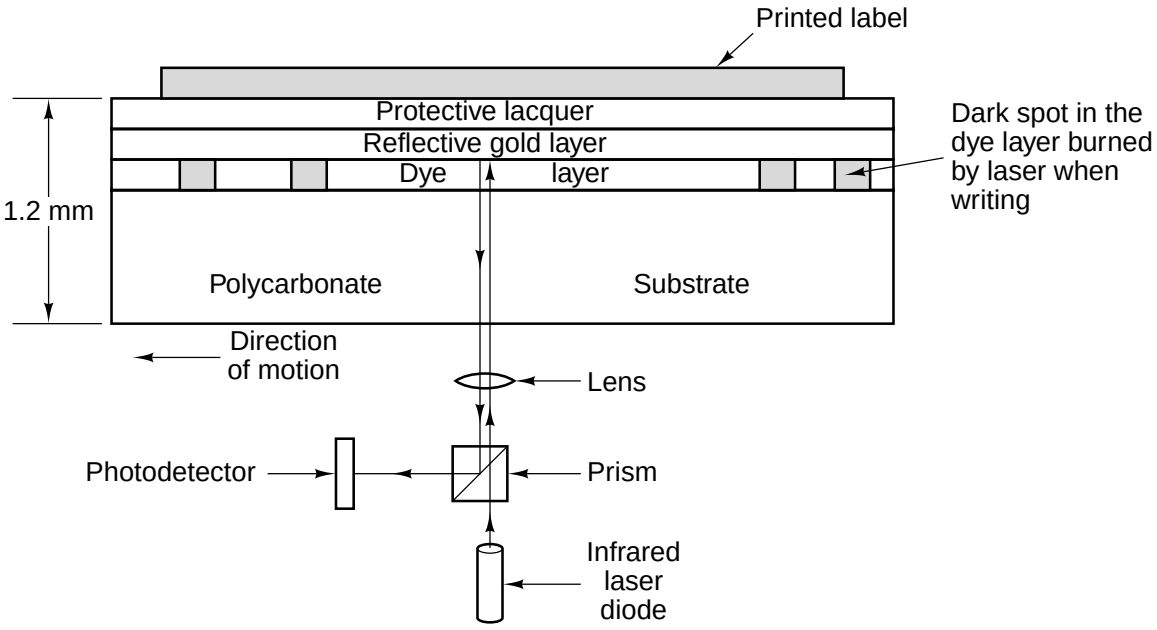

Figure 2-26. Cross section of a CD-R disk and laser (not to scale). A silver CD-ROM has a similar structure, except without the dye layer and with a pitted aluminum layer instead of a gold layer.

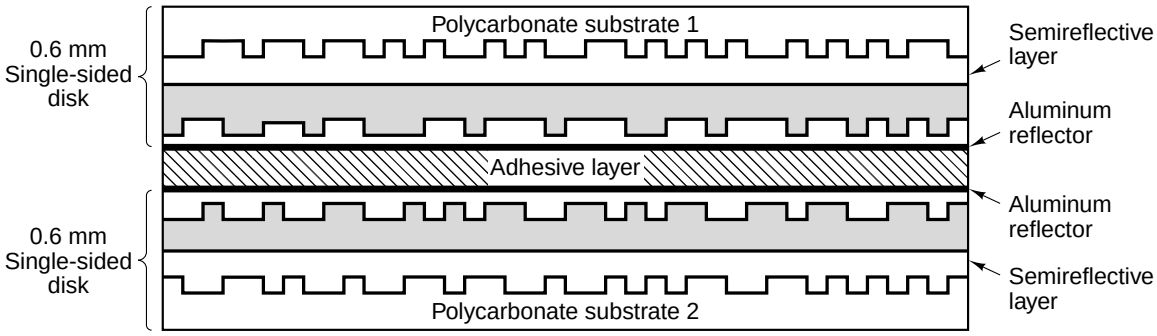

Figure 2-27. A double-sided, dual layer DVD disk.

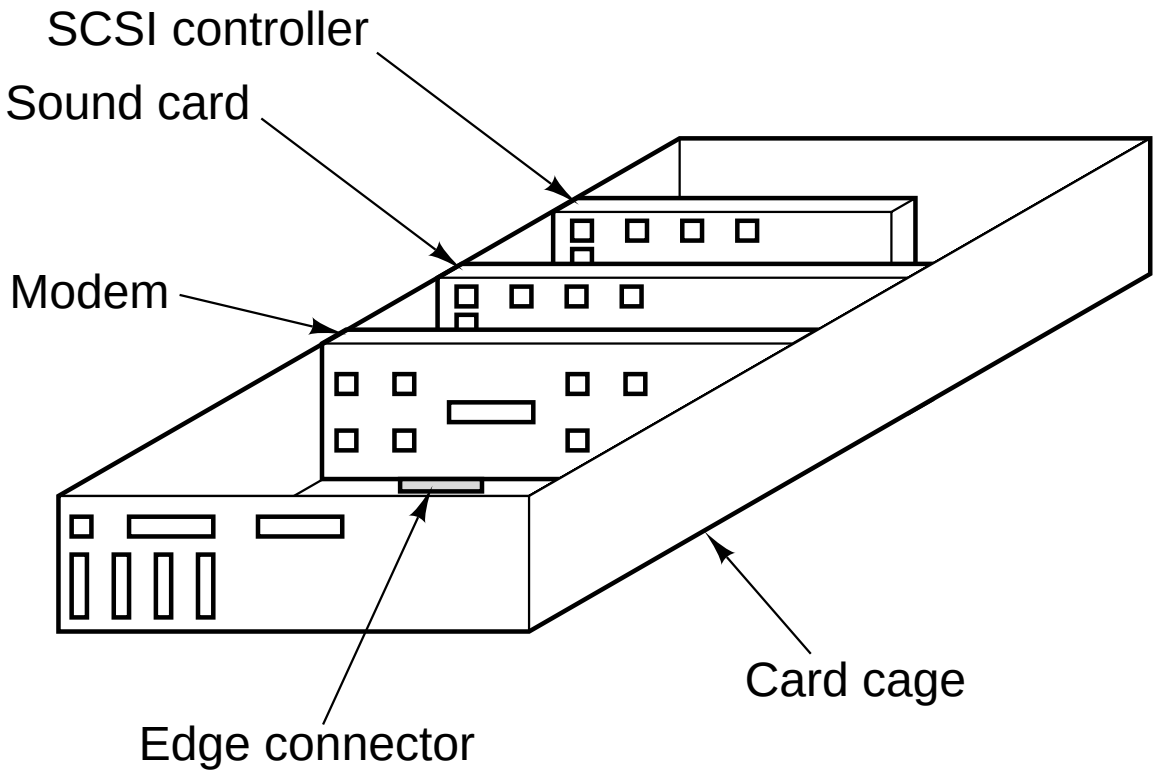

Figure 2-28. Physical structure of a personal computer.

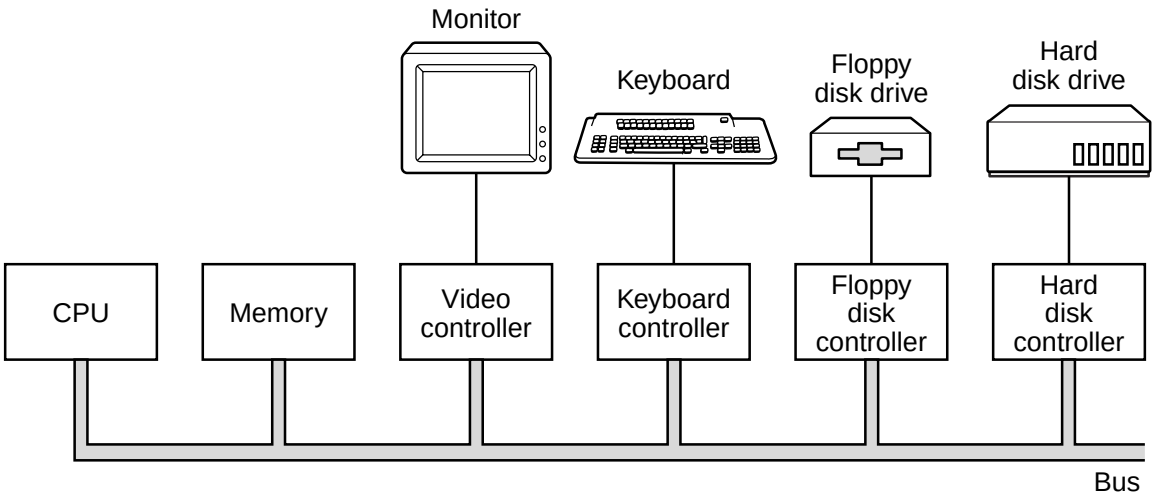

Figure 2-29. Logical structure of a simple personal computer.

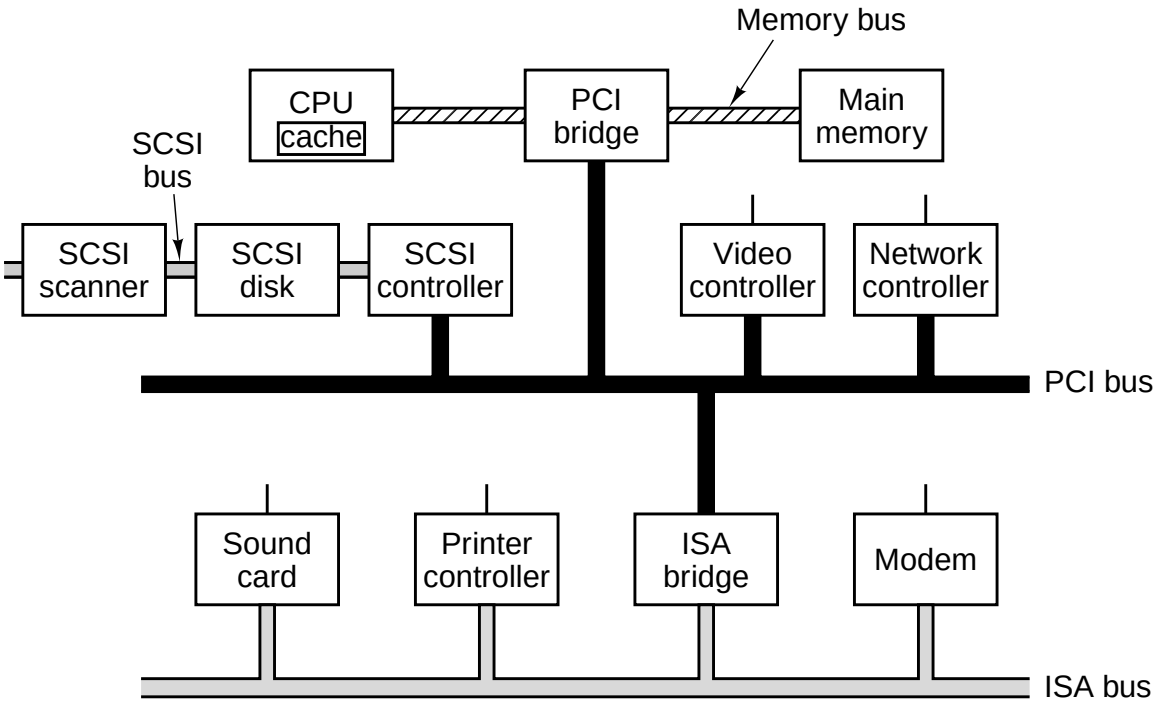

Figure 2-30. A typical modern PC with a PCI bus and an ISA bus. The modem and sound card are ISA devices; the SCSI controller is a PCI device.

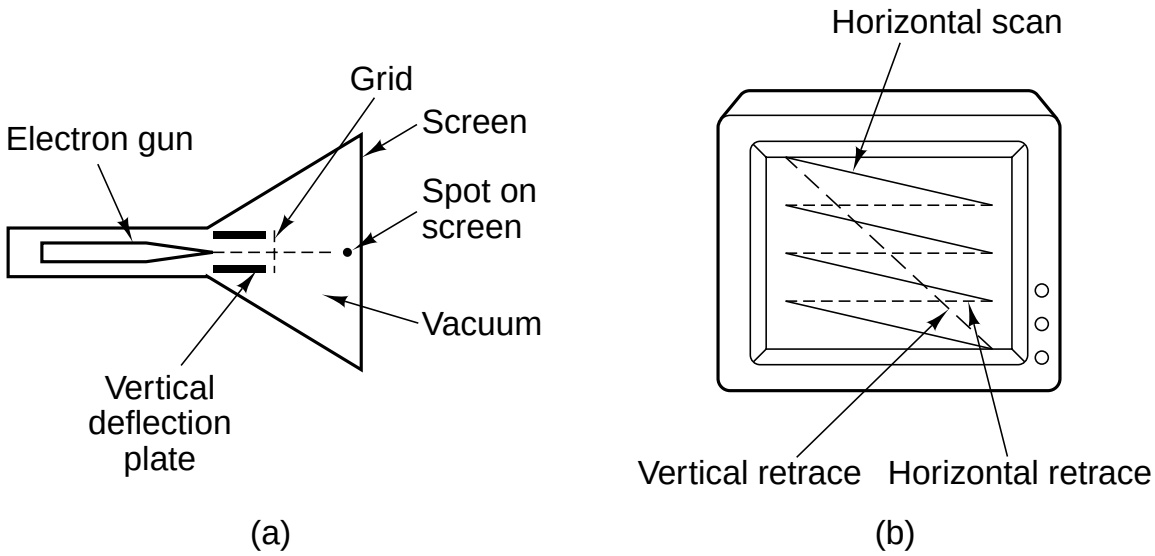

Figure 2-31. (a) Cross section of a CRT. (b) CRT scanning pattern.

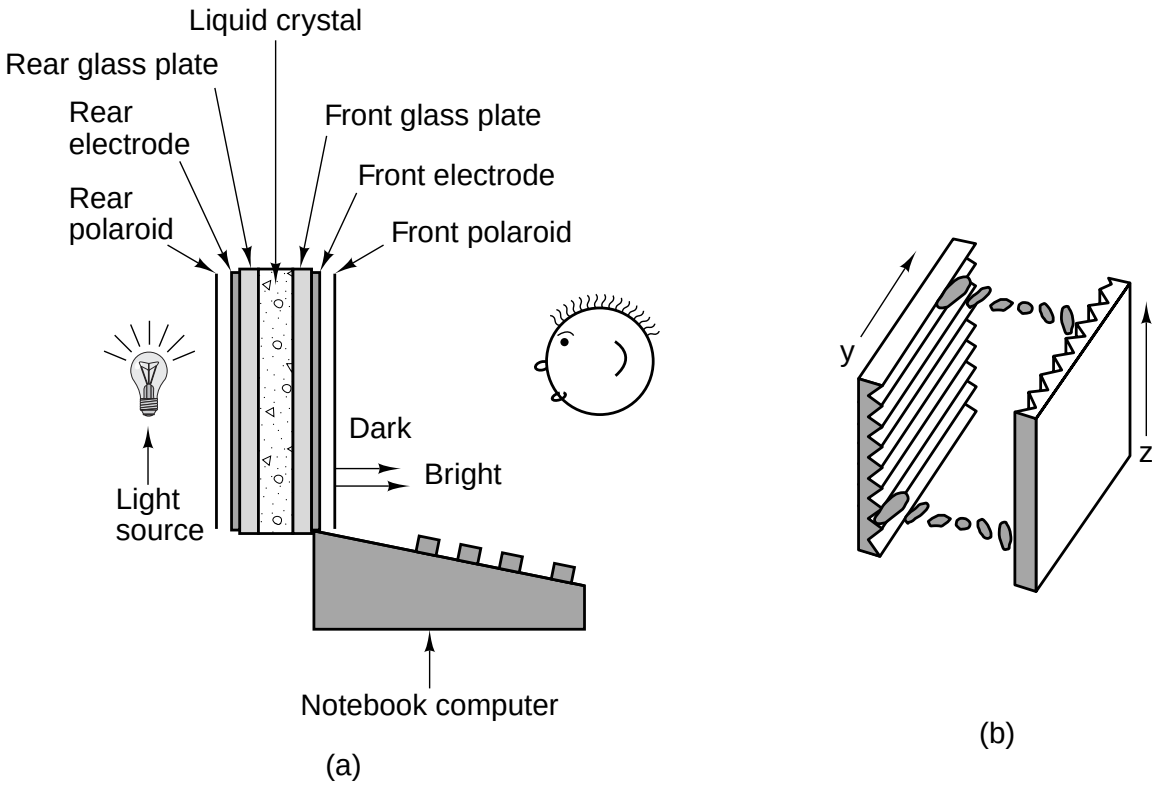

Figure 2-32. (a) The construction of an LCD screen. (b) The grooves on the rear and front plates are perpendicular to one another.

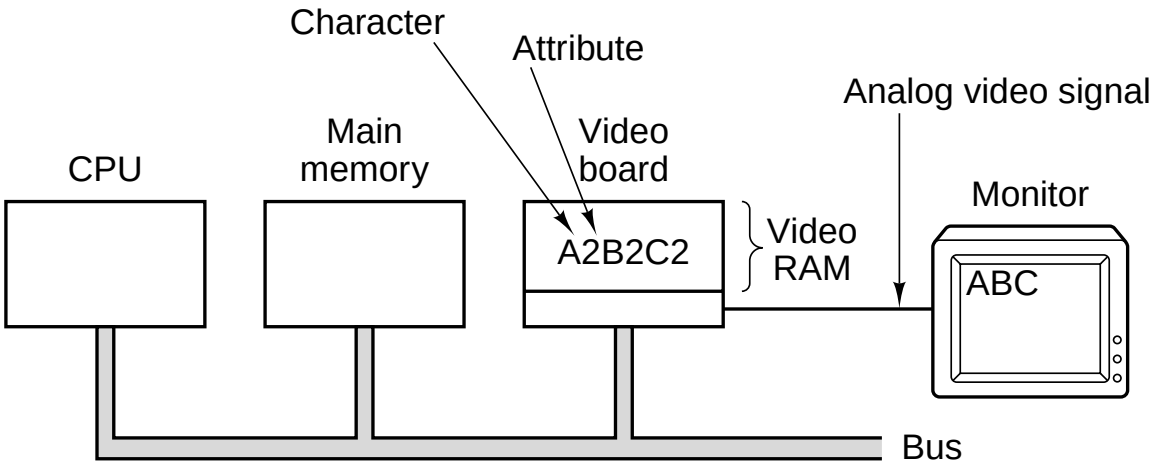

Figure 2-33. Terminal output on a personal computer.

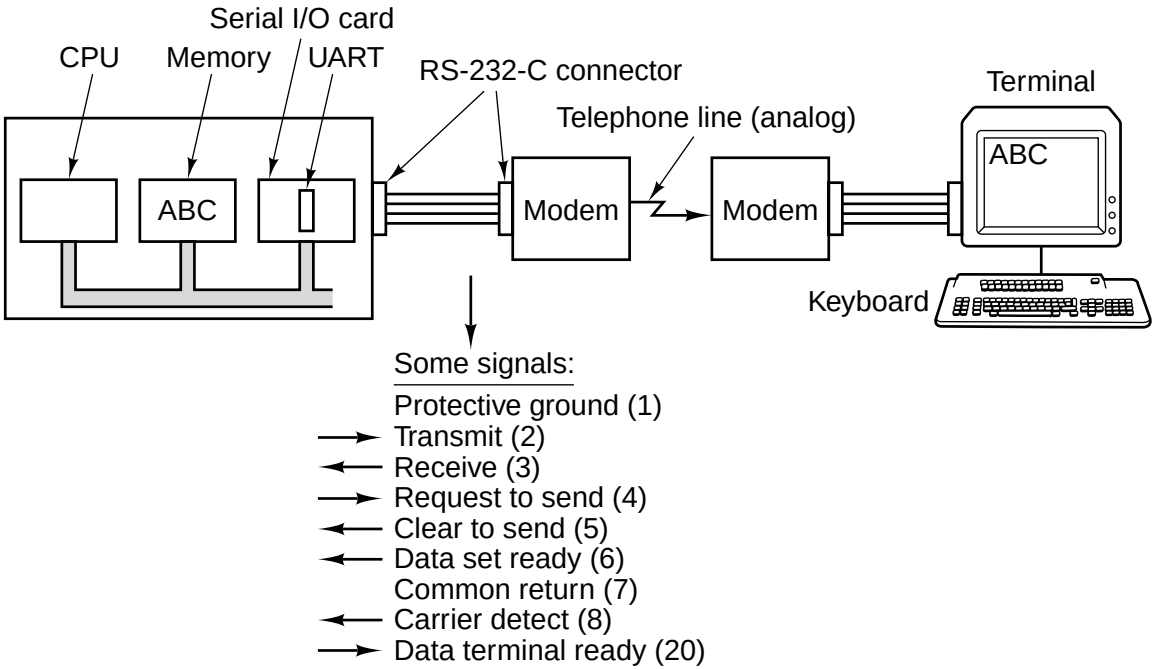

Figure 2-34. Connection of an RS-232-C terminal to a computer. The numbers in parentheses in the list of signals are the pin numbers.

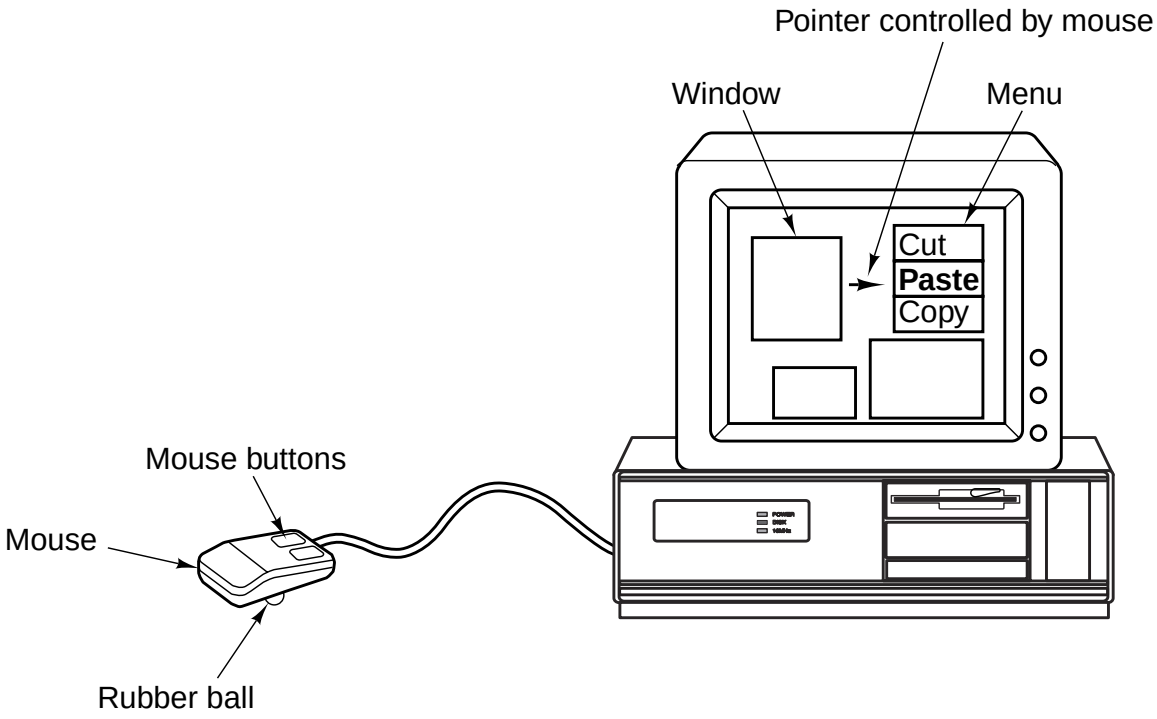

Figure 2-35. A mouse being used to point to menu items.

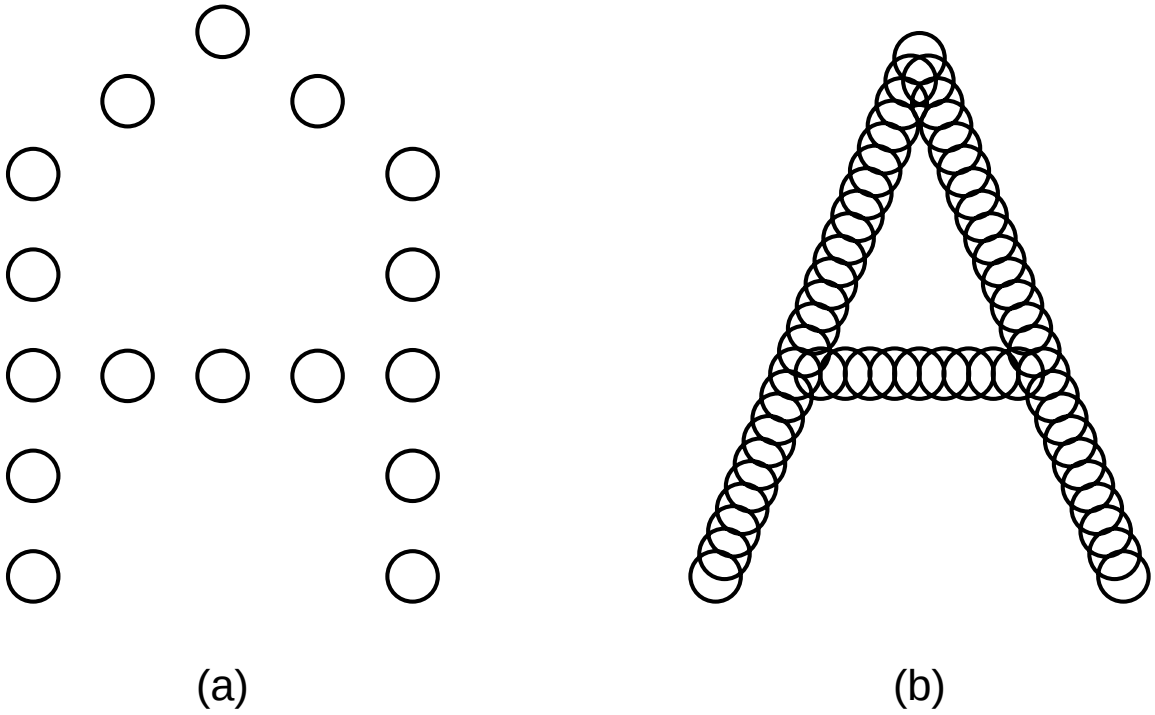

Figure 2-36. (a) The letter ‘‘A’’ on a 5×7 matrix. (b) The letter ‘‘A’’ printed with 24 overlapping needles.

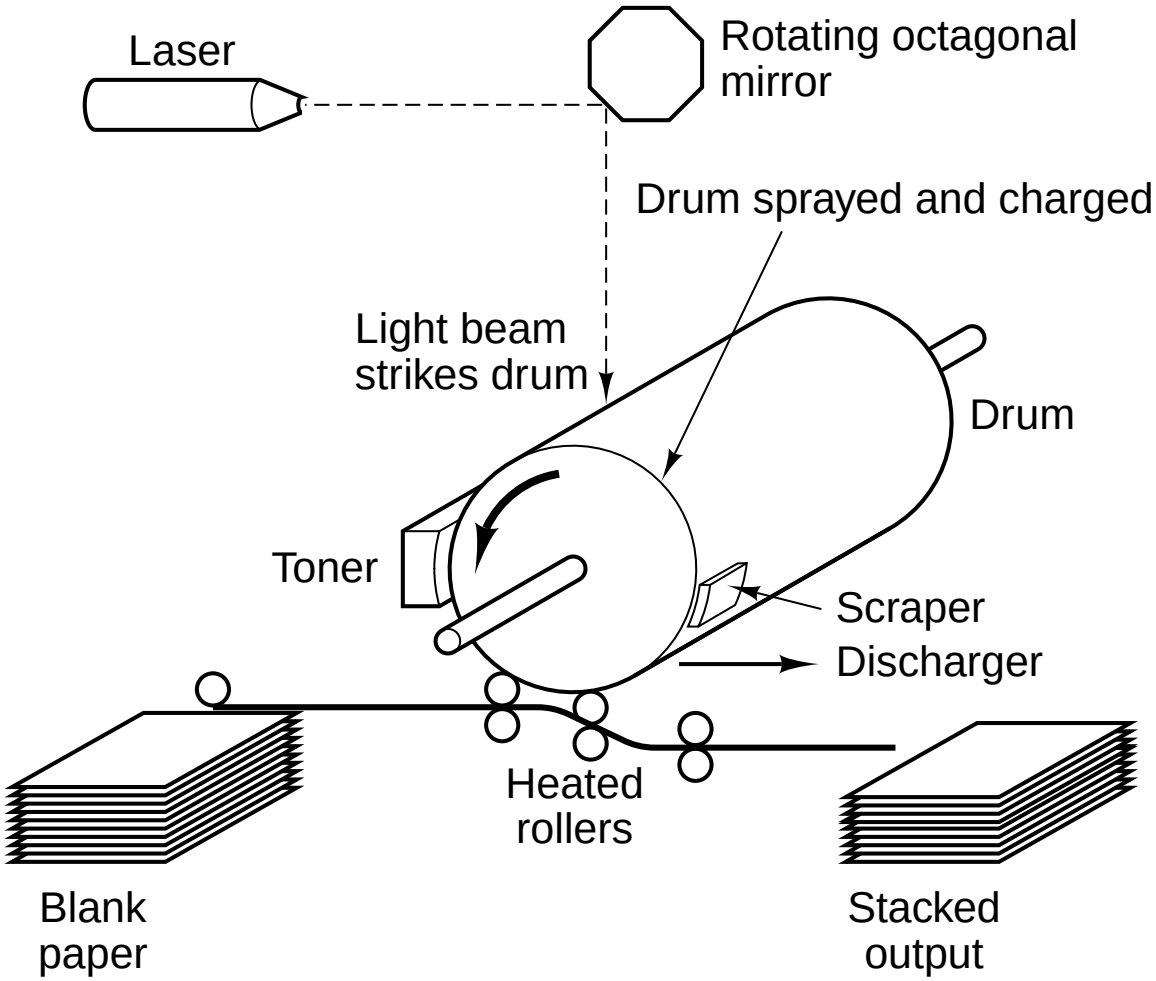

Figure 2-37. Operation of a laser printer.

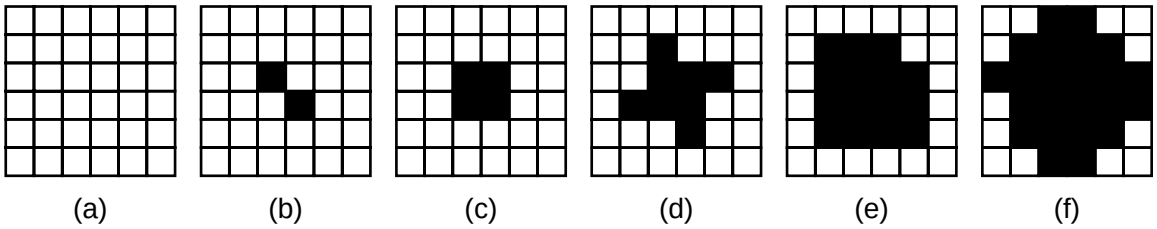

Figure 2-38. Halftone dots for various gray scale ranges. (a) 0–6. (b) 14–20. (c) 28–34. (d) 56–62. (e) 105–111. (f) 161–167.

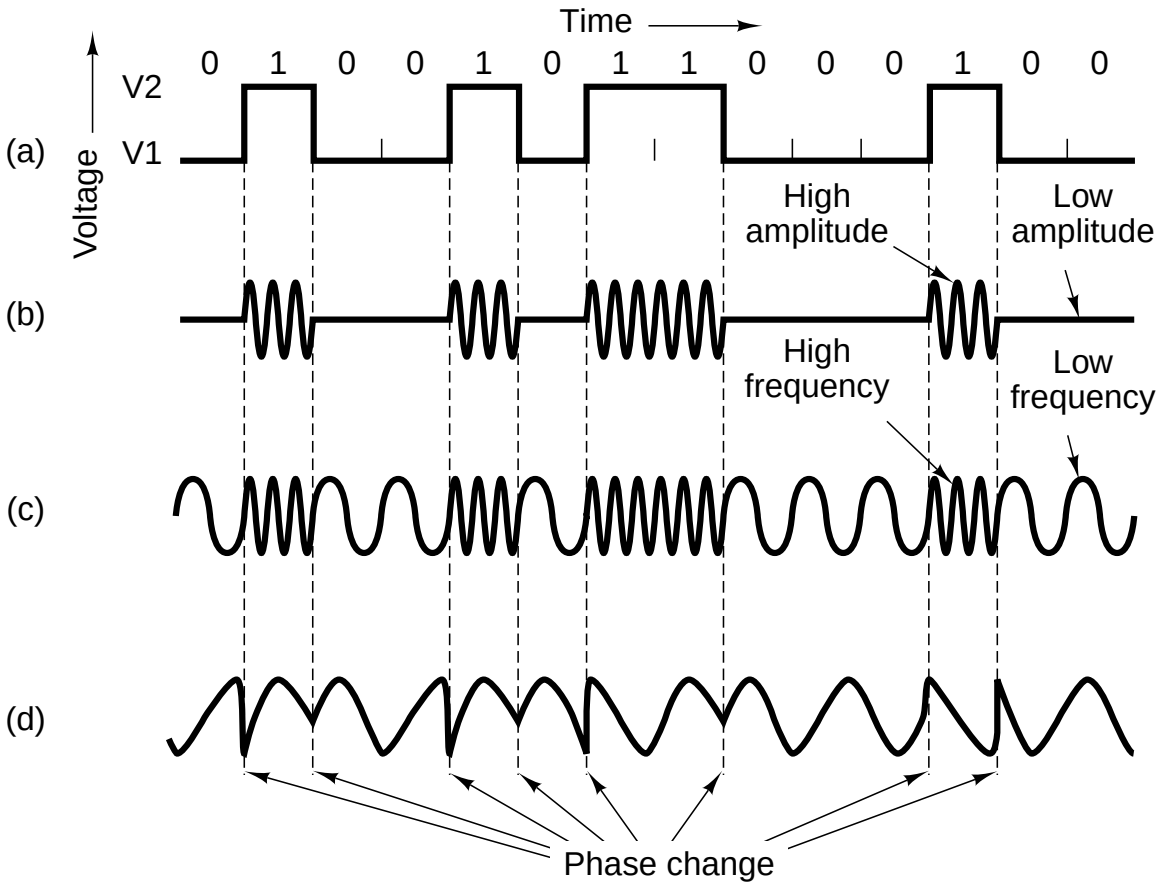

Figure 2-39. Transmission of the binary number 01001011000100 over a telephone line bit by bit. (a) Two-level signal. (b) Amplitude modulation. (c) Frequency modulation. (d) Phase modulation.

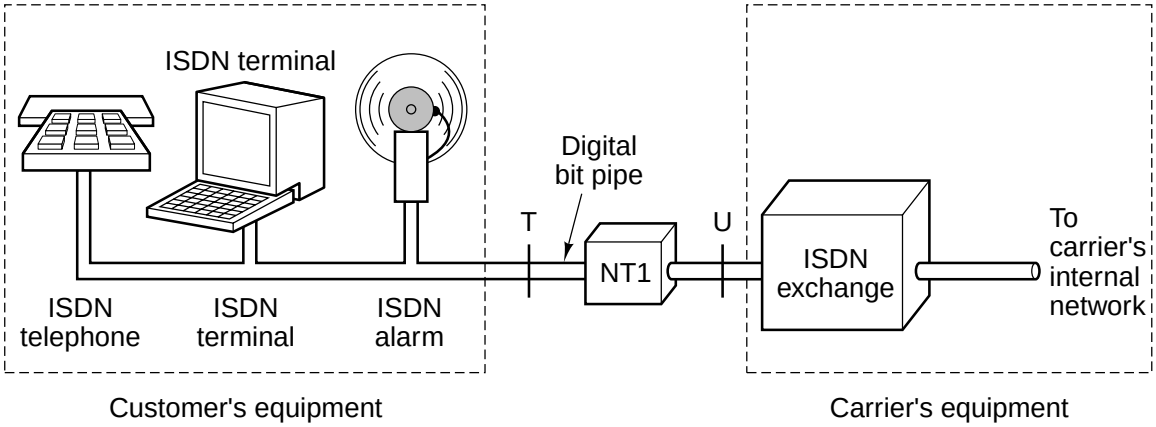

Figure 2-40. ISDN for home use.