Chapter 1
The General Purpose Machine

1.1 The Intel 8086 processor addresses a maximum of 2^{20} bytes of memory. What is the maximum number of 16-bit words that can be stored in this memory? (§1.2)

Solution: One byte = 8 bits, so a 16-bit word = 2 bytes. 2^{20} bytes of memory is equivalent to 2^{19} 16-bit words. In decimal this is about a half million words.

1.3 If the cost of RAM is $95 for a 4MB module, what will it cost to install 32 Mwords in an originally empty machine if the word size is 32 bits? (§1.2)

Solution: With a word size of 32 bits, one word = 4 bytes. So 32MW = $32\text{M} \times 4\text{B}$ = 128MB. At $95 for 4MB, this gives 128MB/memory × $95/4MB = $3,040/memory.

1.5 a. A certain machine requires 25 μs to process each data record in a database. How long will it take to process a database containing 100×10^6 records?
b. How many 1.0MB-capacity floppy disks will be required to store the database? (§1.2)

Solution: a. 25 μs/record $\times 100 \times 10^6$ records/database = 25×10 seconds/database ≈ 41.7 minutes.

b. If a record takes 16 bytes, then there are 16×10^8 bytes/database. At 10^6 B/floppy, this means 1600 floppies/database. Unreasonable, even for small records!

1.7 If one printed character can be stored in 1 byte, approximately how many bytes will be required to store the text of this chapter? Do not include the graphics, and do not count the characters one by one. Show your work and state your assumptions. (§1.2)

Solution: In manuscript form there are about 96 characters (including spaces and punctuation) in a full line of text. About 3/4 of a page is full lines on the average, and there are 29 pages in the chapter. A full page could contain 50 lines.

50 lines/page × 29 pages/chap × 75% × 96char/line ≈ 34,800 char/chapter ≈ 35 KB.

1.9 Describe the tools used by the assembly language programmer. (§1.3)

Solution: The tools used by the assembly language programmer are editor, assembler, linker, loader, debugger, and development system. The *editor* is used to edit the source code (the assembly language). The *assembler* allows the programmer to generate machine language program from assembly language programs. It translates assembly language statements to their binary equivalents. The *linker* links separately assembled modules together into a single module suitable for loading and execution. The *loader* loads the executable binary code into the memory and changes some logical addresses to appropriate physical addresses. The *debugger* allows the programmer to observe the details of program execution. The *development system* is a collection of hardware and software that is used to support system development.

1.11 What is the difference between a programmable calculator and a personal computer? (§1.4)

Solution: The programming capability of a programmable caculator is limited, and there's only one "language." It can only solve some particular kinds of numerical problems. The I/O capability is very poor.

A personal computer is a general purpose machine. Nearly all kinds of operating systems, programming languages, and application software can run on it. It can solve nearly all kinds of problems, such as mathematical computation, controlling, design, information processing, and artificial intelligence problems. It can drive all sorts of I/O devices—vastly more than a calculator. Different computers can also be connected via a network.

1.13 What is an ISA and what are its components? (§1.3)

Solution: The ISA is the collection of instructions and resources. It includes the instruction set, the machine's memory, and all of the programmer-accessible registers in the CPU and elsewhere in the machine.

1.15 Using only the information in Table 1.2, encode the following MC68000 assembly language instructions into machine language. Express your results in binary. (§1.3)

a. `MOVE.W D1, D7`

b. `ADDI.W #511, D6`

Solution: a.

```
MOVE.W D1, D7

0011 111 000 000 001
```

b.

```
ADDI.W #511, D6

00000110 01 000 110
0000 0001 1111 1111 <-- 511
```

1.17 Define the difference between a simulator and an emulator. Which would you prefer to work with, and why? (§1.4)

Solution: Simulators are software tools that mimic aspects of the system's behavior and allow a certain amount of performance estimation at an early part of the design process. Since simulators mimic hardware performance in software, they are usually slower in operation by orders of magnitude. Emulators can be thought of as hardware-assisted simulators that provide operation speed closer to the speed of the hardware being emulated.

Emulators are preferred because they provide results closer to the behavior of the real machine being emulated.

1.19 Provide a diagram similar to Figure 1.5 for the computer you work at most frequently. (§1.5)

Solution: The diagram should include CPU, memory, I/O, and their interconnection. The name of each bus should be specified. I/O devices may include keyboard, mouse, bit pad, display, printer, disk drive, CD ROM, tape, network, other computers, and so on.

1.21 What natural separation distinguishes computer logic design from classical logic design? (§1.5)

Solution: The computer designer does not design an entire digital computer using state machine design techniques. There is a natural separation or partitioning of concerns between the data path and the control path. Whereas the logic designer sees logic gates and flip-flops, the computer designer sees multiplexers, decoders, and register files.

1.23 Describe as accurately as you can the implementation domains of the computer proposed by Charles Babbage. (§1.5, §1.6)

Solution: Gears, shafts, wheels, cranks, and dials.