

Practice Problem 2.18 (solution page 185)

In Chapter 3, we will look at listings generated by a *disassembler*, a program that converts an executable program file back to a more readable ASCII form. These files contain many hexadecimal numbers, typically representing values in two's-complement form. Being able to recognize these numbers and understand their significance (for example, whether they are negative or positive) is an important skill.

For the lines labeled A–I (on the right) in the following listing, convert the hexadecimal values (in 32-bit two's-complement form) shown to the right of the instruction names (**sub**, **mov**, and **add**) into their decimal equivalents:

| | | | | |
|---------|----------------------|-----|-------------------------|----|
| 4004d0: | 48 81 ec e0 02 00 00 | sub | \$0x2e0,%rsp | A. |
| 4004d7: | 48 8b 44 24 a8 | mov | -0x58(%rsp),%rax | B. |
| 4004dc: | 48 03 47 28 | add | 0x28(%rdi),%rax | C. |
| 4004e0: | 48 89 44 24 d0 | mov | %rax,-0x30(%rsp) | D. |
| 4004e5: | 48 8b 44 24 78 | mov | 0x78(%rsp),%rax | E. |
| 4004ea: | 48 89 87 88 00 00 00 | mov | %rax,0x88(%rdi) | F. |
| 4004f1: | 48 8b 84 24 f8 01 00 | mov | 0x1f8(%rsp),%rax | G. |
| 4004f8: | 00 | | | |
| 4004f9: | 48 03 44 24 08 | add | 0x8(%rsp),%rax | |
| 4004fe: | 48 89 84 24 c0 00 00 | mov | %rax,0xc0(%rsp) | H. |
| 400505: | 00 | | | |
| 400506: | 48 8b 44 d4 b8 | mov | -0x48(%rsp,%rdx,8),%rax | I. |

| | | | | | | | | |
|---------------|------|------|------|------|------|------|------|------|
| Hex digit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Decimal value | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Binary value | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| Hex digit | 8 | 9 | A | B | C | D | E | F |
| Decimal value | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Binary value | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

Figure 2.2 Hexadecimal notation. Each hex digit encodes one of 16 values.

- A. $0x2e0 = 001011100000 = 2 \times 16^2 + 14 \times 16^1 + 0 \times 16^0 = 736$
- B. $-0x58 = -(01011000) = -(5 \times 16^1 + 8 \times 16^0) = -88$
- C. $0x28 = 00101000 = 2 \times 16^1 + 8 \times 16^0 = 40$
- D. $-0x30 = (00110000) = -(3 \times 16^1 + 0 \times 16^0) = -48$
- E. $0x78 = 01111000 = 7 \times 16^1 + 8 \times 16^0 = 120$
- F. $0x88 = 0000000010001000 = 8 \times 16^1 + 8 \times 16^0 = 136$
- G. $0x1f8 = 000000011111000 = 1 \times 16^2 + 15 \times 16^1 + 8 \times 16^0 = 504$
- H. $0xc0 = 0000000011000000 = 12 \times 16^1 + 0 \times 16^0 = 192$
- I. $-0x48 = -(01001000) = -4 \times 16^1 + 8 \times 16^0 = -72$