# CHAPTER 8

## Exercises

**E8.1** The number of bits in the memory addresses is the same as the address bus width, which is 20. Thus the number of unique addresses is $2^{20}$ = 1,048,576 = 1024 $\times$ 1024 = 1024K.

**E8.2** (8 bits/byte) $\times$ (64 Kbytes) = 8 $\times$ 64 $\times$ 1024 = 524,288 bits

**E8.3** Starting from the initial situation shown in Figure 8.9a in the book, execution of the command PSHB results in:

```
            0907:
            0908:
SP →        0909:
            090A: A2
```

Then, execution of the command PSHA results in:

```
            0907:
SP →        0908:
            0909:  34
            090A:  A2
```

Then, the PULX command reads two bytes from the stack and we have:

```
            0907:              X:  34A2
            0908:
            0909:  34
SP →        090A:  A2
```

**E8.4** Starting from the initial situation shown in Figure 8.9a in the book, execution of the command PSHX results in:

```
            0907:
SP →        0908:
            0909:  00
            090A:  00
```

Then, the command PSHA results in:

| | | |
|---|---|---|
| SP → | 0907: | |
| | 0908: | 34 |
| | 0909: | 00 |
| | 090A: | 00 |

Next the PULX command reads two bytes from the stack, and we have:

| | | |
|---|---|---|
| | 0907: | X: 3400 |
| | 0908: | 34 |
| SP → | 0909: | 00 |
| | 090A: | 00 |

**E8.5** The results are given in the book.

**E8.6** (a) LDAA $0202
This instruction uses extended addressing. The effective address is 0202. In Figure 8.11 we see that this location contains 1A. Thus the content of the A register after this instruction is 1A.
(b) LDAA #$43
This instruction uses immediate addressing. The effective address is the one immediately following the op code. This location contains the hexadecimal digits 43. Thus the content of the A register after this instruction is 43.
(c) LDAA $05,X
This instruction uses indexed addressing. The effective address is the content of the X register plus the offset which is 05. Thus the effective address is 0205. In Figure 8.11 we see that this location contains FF. Thus the content of the A register after this instruction is FF.
(d) LDAA $06
This instruction uses direct addressing. The effective address is 0006. In Figure 8.11 we see that this location contains 13. Thus the content of the A register after this instruction is 13.
(e) LDAA $07,X
This instruction uses indexed addressing. The effective address is the content of the X register plus the offset which is 07. Thus the effective address is 2007. In Figure 8.11 we see that this location contains 16. Thus the content of the A register after this instruction is 16.

**E8.7** (a) Referring to Table 8.1 in the book, we see that CLRA is the clear accumulator A instruction with a single byte op code 4F. Furthermore execution of this command sets the Z bit of the condition code register. The BEQ $15 command occupies two memory locations with the op code 27 in the first byte and the offset of 15 in the second byte. Thus starting in location 0200, the instructions appear in memory as:

0200: 4F
0201: 27
0202: 14

(b) When the instructions are executed, the CLRA command sets the Z bit. Then if the Z-bit was clear the next instruction would be the one starting in location 0203 following the BEQ $15 command. However since the Z bit is set the next instruction is located at 0203 + 15 = 0218.

**E8.8** One answer is given in the book. Of course, other correct answers exist.

**E8.9** One answer is given in the book. Of course, other correct answers exist.

## Answers for Selected Problems

**P8.7*** 262,144 bytes

**P8.9*** ROM is read-only memory. Some types are:

1. Mask-programmable ROM in which the data is written when the memory is manufactured.
2. Programmable read-only memory (PROM) in which data is written by special circuits that blow tiny fuses or not depending on whether the data bits are zeros or ones.
3. Erasable PROMs (EPROMs) that can be erased by exposure to ultraviolet light (through a window in the chip package) and rewritten using special circuits.
4. Electrically erasable PROMs (EEPROMs) that can be erased by applying proper voltages to the chip.

All types of ROM are nonvolatile and are used for program storage in embedded computers.

**P8.13*** In the ignition control system for automobiles, we need to use ROM for the programs and fixed data, because ROM is nonvolatile. Some RAM would be needed for temporary data such as air temperature and throttle setting. Presumably many units would be needed for mass production and mask programmable ROM would be least expensive.

**P8.19*** A digital sensor produces a logic signal or a digital word as its output. An analog sensor produces an analog output signal that varies continuously with the variable being measured.

**P8.23*** A D/A is a digital-to-analog converter that converts a sequence of digital words into an analog signal. They are needed when an analog actuator must be controlled by a digital computer.

**P8.27*** A stack is a sequence of locations in RAM used to store information such as the contents of the program counter and other registers when a subroutine is executed or when an interrupt occurs. Information is added to (pushed onto) the top of the stack and then read out (pulled off) in the reverse order that it was written. After data are pulled off the stack, they are considered to no longer exist in memory. The stack pointer is a register that keeps track of the address of the location immediately above the top of the stack.

**P8.29*** Initially, we have:

| | |
|---|---|
| A: 07 | 0048: 00 |
| B: A9 | 0049: 00 |
| SP: 004E | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: 00 |
| | 004E: 00 |
| | 004F: 00 |

After the command PSHA, we have:

| | |
|---|---|
| A: 07 | 0048: 00 |
| B: A9 | 0049: 00 |
| SP: 004D | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: 00 |
| | 004E: 07 |
| | 004F: 00 |

After the command PSHB, we have:

| | |
|---|---|
| A: 07 | 0048: 00 |
| B: A9 | 0049: 00 |
| SP: 004C | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: A9 |
| | 004E: 07 |
| | 004F: 00 |

After the command PULA, we have:

| | |
|---|---|
| A: A9 | 0048: 00 |
| B: A9 | 0049: 00 |
| SP: 004D | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: A9 |
| | 004E: 07 |
| | 004F: 00 |

After the command PULB, we have:

| | |
|---|---|
| A: A9 | 0048: 00 |
| B: 07 | 0049: 00 |
| SP: 004E | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: A9 |
| | 004E: 07 |
| | 004F: 00 |

After the command PSHX, we have:

| | |
|---|---|
| A: A9 | 0048: 00 |
| B: 07 | 0049: 00 |
| SP: 004C | 004A: 00 |
| X: 34BF | 004B: 00 |
| | 004C: 00 |
| | 004D: 34 |
| | 004E: BF |
| | 004F: 00 |

**P8.31*** A sequence of instructions that results in swapping the high and low bytes of the X register is:

```
PSHD        save the original contents of A and B
PSHX
PULD        move the content of X to D
PSHA
PSHB
PULX
PULD        restore contents of A and B
```

**P8.32**

| | | | | |
|---|---|---|---|---|
| (a)* | LDAA | $2002 | Extended addressing | A:20 |
| (c)* | LDAA | $04 | Direct addressing | A:9A |
| (e)* | INCA | | Inherent addressing | A:02 |
| (g)* | LDAA | $2007 | Extended addressing | A:F3 |

**P8.33** *(a) The command starting at location 200A is the next to be executed after the BMI $07 command.

**P8.34**

| | Mnemonics | Machine codes | Memory locations occupied by instruction |
|---|---|---|---|
| (a)* | CLRA | 4F | 1 |
| (b)* | ADDA $4A | 9B 4A | 2 |

**P8.36*** (a) 48
(b) 09

**P8.37*** A:14 and B:E0

**P8.39*** * ANSWER FOR PROBLEM 8.39

```
        ORG     0200    Directive to begin in location 0200
        LDAB    #$0B    0B is hex equivalent of decimal 11
        MUL             compute product
        STD     $FF00   store result
        STOP
        END
```

**P8.41*** This subroutine first clears register B which will hold the quotient after the program has been executed. Then 3 is repeatedly subtracted from the content of A, and the content of B is incremented until the value in A is negative.

```
*DIVIDE BY 3
*
DIV3    CLRB
LOOP    SUBA    #$03    subtract 3 from content of A.
        BMI     END     quit if result is negative
        INCB
        JMP     LOOP
END     ADDA    #$03    restore remainder to register A
        RTS
```

## Practice Test

**T8.1.** **a**. 11, **b**. 17, **c**. 21, **d**. 24, **e**. 27, **f**. 13, **g**. 26, **h**. 9, **i**. 20, **j**. 12, **k**. 15, **l**. 16, **m**. 8, **n**. 29, **o**. 23, **p**. 30.

**T8.2.** **a**. direct, 61; **b**. indexed, F3; **c**. inherent, FF; **d**. inherent, 01; **e**. immediate, 05; **f**. immediate, A1.

**T8.3** Initially, we have:

| | |
|---|---|
| A: A6 | 1034: 00 |
| B: 32 | 1035: 00 |
| SP: 1038 | 1036: 00 |
| X: 1958 | 1037: 00 |
| | 1038: 00 |
| | 1039: 00 |
| | 103A: 00 |
| | 103B: 00 |
| | 103C: 00 |

After the command PSHX, we have:

| | |
|---|---|
| A: A6 | 1034: 00 |
| B: 32 | 1035: 00 |
| SP: 1036 | 1036: 00 |
| X: 1958 | 1037: 19 |
| | 1038: 58 |
| | 1039: 00 |
| | 103A: 00 |
| | 103B: 00 |
| | 103C: 00 |

After the command PSHB, we have:

| | |
|---|---|
| A: A6 | 1034: 00 |
| B: 32 | 1035: 00 |
| SP: 1035 | 1036: 32 |
| X: 1958 | 1037: 19 |
| | 1038: 58 |
| | 1039: 00 |
| | 103A: 00 |
| | 103B: 00 |
| | 103C: 00 |

After the command PULA, we have:

| | |
|---|---|
| A: 32 | 1034: 00 |
| B: 32 | 1035: 00 |
| SP: 1036 | 1036: 32 |
| X: 1958 | 1037: 19 |
| | 1038: 58 |
| | 1039: 00 |
| | 103A: 00 |
| | 103B: 00 |
| | 103C: 00 |

After the command PSHX, we have:

| | |
|---|---|
| A: 32 | 1034: 00 |
| B: 32 | 1035: 19 |
| SP: 1034 | 1036: 58 |
| X: 1958 | 1037: 19 |
| | 1038: 58 |
| | 1039: 00 |
| | 103A: 00 |
| | 103B: 00 |
| | 103C: 00 |