

Assignment 9

11510837 吴昊宇

1. In each of the following cases, write a short program in the machine language described in Appendix C to perform the requested activities. Assume that each of your programs is placed in memory starting at address 00.[6 points]

- a. Move the value at memory location D8 to memory location B3.
- b. Interchange the values stored at memory locations D8 and B3.
- c. If the value stored in memory location 44 is 00, then place the value 01 in memory location 46; otherwise, put the value FF in memory location 46.

ANSWER:

a.

```
1 10D8
2 30B3
3 C000
```

Address	Cells
00	10
01	D8
02	30
03	B3
04	C0
05	00

b.

- 1 10D8
- 2 11B3
- 3 30B3
- 4 31D8
- 5 C000

Address	Cells
00	10
01	D8
02	11
03	B3
04	30
05	B3
06	31
07	D8
08	C0
09	00

c.

- 1 1144
- 2 2000
- 3 2201
- 4 B10A
- 5 22FF
- 6 3246
- 7 C000

Address	Cells
00	11
01	44
02	20
03	00
04	22
05	01
06	B1
07	0A
08	22
09	FF
0A	32
0B	46
0C	C0
0D	00

2. Perform the indicated operations.[2 points]

a. 11111111

OR 00101101

b. 01001011

XOR 10101011

ANSWER:

a. 11111111

b. 11100000

3.What is the result of performing a 1-bit left circular shift on the following bytes represented in hexadecimal notation? Give your answer in hexadecimal form.[8 points]

a. AB

b. 5C

c. B7

d. 35

ANSWER:

a.

$$AB_{16} = 10101011_2$$

10101011 1-bit left circular shift -> **01010111**

$$\mathbf{01010111_2 = 57_{16}}$$

b.

$$5C_{16} = 1011100_2$$

1011100 1-bit left circular shift -> **0111001**

$$\mathbf{0111001_2 = 39_{16}}$$

c.

$$B7_{16} = 10110111_2$$

10110111 1-bit left circular shift -> **01101111**

01101111₂ = **6F**₁₆

d.

35₁₆ = **00110101**₂

00110101 1-bit left circular shift -> **01101010**

01101010₂ = **6A**₁₆

4. Using the machine language of Appendix C, write a program that copies the middle 4 bits from memory cell E0 into the least significant 4 bits of memory cell E1, while placing 0s in the most significant 4 bits of the cell at location E1. [4 points]

ANSWER:

```
1 15E0    // E0 -> R5
2 A502    // R5 right move 2 bits
3 260F    // 0F -> R6  0000 1111
4 8056    // R5 AND R6 -> R0
5 30E1    // R0 -> E1
6 C000
```

Address	Cells
00	15
01	E0
02	A5
03	02
04	26
05	0F
06	80
07	56
08	30
09	E1
10	C0
11	00