Computer Science

1. (10 pts.) Convert the following hexadecimal numbers to decimal. a. $(AB2)_{16}$ b. $(123)_{16}$ c. $(ABB)_{16}$ d. $(35E.E1)_{16}$

E→14 D→13 F→15 C→12 B→11 A→10

10

a、$10\times16^2+11\times16^1+2\times16^0=2560+176+2=2738$

b、$1\times16^2+2\times16^1+3\times16^0=256+32+3=291$

c、$10\times16^2+11\times16^1+11\times16^0=2560+176+11=2747$

d、$3\times16^2+5\times16^1+14\times16^0+14\times16^{-1}+1\times16^{-2}=768+80+14+\frac{14}{16}+\frac{1}{256}=862+\frac{224}{256}=862\frac{7}{8}=862.875$

2. Convert the following binary numbers to hexadecimal. a. $(01101)_2$ b. $(1011000)_2$ c. $(01110.01)_2$ d. $(111111.111)_2$

F→15 E→14 D→13 C→12 B→11 A→10

5

a、$0\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0=8+4+1=13$ $(13)_{10}=D_{16}$

b、$1\times2^6+0\times2^5+1\times2^4+1\times2^3+0\times2^2+0\times2^1+0\times2^0=64+16+8=88_{10}=5\times16^1+8\times16^0=(58)_{16}$

c、$0\times2^4+1\times2^3+1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+0\times2^{-2}=8+4+2+\frac{1}{4}=14.25$

d、$1\times(2^5+2^4+2^3+2^2+2^1+2^0+2^{-1}+2^{-2}+2^{-3})=32+16+8+4+2+1+\frac{1}{2}+\frac{1}{4}+\frac{1}{8}=63+\frac{7}{8}=63.875$

3. Change the following 8-bit two's complement numbers to decimal. a. 01110111 b. 10010100 c. 01110100 d. 11001110

5

a、$0\times2^7+1\times2^6+1\times2^5+1\times2^4+0\times2^3+1\times2^2+1\times2^1+1\times2^0=64+32+16+4+2+1=119$

b、$1\times2^7+0\times2^6+0\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+0\times2^0=128+16+4=148$

c、$0\times2^7+1\times2^6+1\times2^5+1\times2^4+0\times2^3+1\times2^2+0\times2^1+0\times2^0=64+32+16+4=116$

d、$1\times2^7+1\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+1\times2^1+0\times2^0=128+64+8+4+2=206$

4. Change the following decimal numbers to 8-bit two's complement integers. a. -12 b. -145 c. 56 d. 142

2

a、$-(1\times2^3+1\times2^2+0\times2^1+0\times2^0)=(-1100)_2$

b、$-(1\times2^7+0\times2^6+0\times2^5+1\times2^4+0\times2^3+0\times2^2+0\times2^1+1\times2^0)=-(10010001)_2$

c、$56=1\times2^5+1\times2^4+1\times2^3+0\times2^2+0\times2^1+0\times2^0=(111000)_2$

d、$142=1\times2^7+0\times2^6+0\times2^5+0\times2^4+1\times2^3+1\times2^2+1\times2^1+0\times2^0=(10001110)_2$

5. Convert the following numbers in 32-bit IEEE format. (Hint: 32-bit IEEE format

exponent size is 8 bits

mantissa size is 23 bits

bias is 127

0

a. $-2^0\times1.10001_2$

b. $+2^3\times1.111111_2$

6. Convert the following numbers in 32-bit IEEE format. a. 7.1875 b. -12.640625

0

7. We need to unset (force to 0) the four leftmost bits of an 8-bit pattern. Show the mask and the operation.

10

let the 8-bit pattern: 10101100

mask: 00001111

```
      10101100
And)  00001111
      --------
      00001100
```

8. We need to flip the three rightmost and the two leftmost bits of a pattern. Show the mask and the operation.

10

let the patern: 10001110 mask: 11000111

the mask of the three rightmost bits: 111

the mask of the two leftmost bits: 11

```
      10001110
XOR)  11000111
      --------
      01001001
```

a 9. Which of the following operation creates an overflow if numbers and the result are represented in 8-bit two's complement representation? a. $11000010_2 + 00111111_2$ b. $00000010_2 + 00111111_2$

0

a, 11000010
+ 00111111
100000001

b, 00000010
+ 00111111
01000001

A: a

10. Using the instruction set of the Table 5.4 and the initial state of CPU and memory, write the content of memory after that the CPU reaches a HALT instruction.

0

145 | 2
72 1
36 0
18 0
9 0
4 1
2 0
1 0

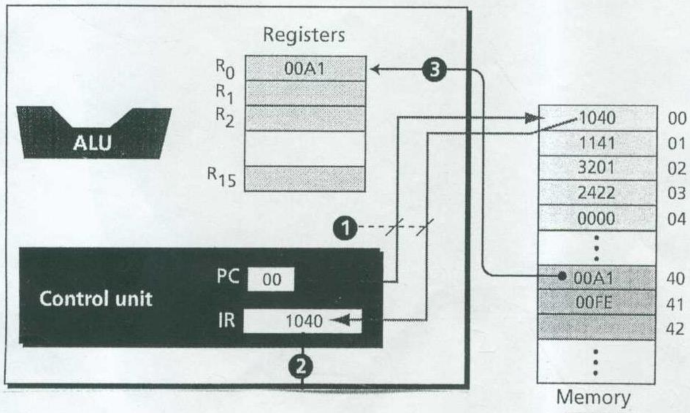

Table 5.4 List of instructions for the simple computer

Instruction	*Code*	*Operands*			*Action*
	d_1	d_2	d_3	d_4	
HALT	0				Stops the execution of the program
LOAD	1	R_D	M_S		$R_D \leftarrow M_S$
STORE	2	M_D		R_S	$M_D \leftarrow R_S$
ADDI	3	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} + R_{S2}$
ADDF	4	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1} + R_{S2}$
MOVE	5	R_D	R_S		$R_D \leftarrow R_S$
NOT	6	R_D	R_S		$R_D \leftarrow R_S$
AND	7	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ AND R_{S2}
OR	8	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ OR R_{S2}
XOR	9	R_D	R_{S1}	R_{S2}	$R_D \leftarrow R_{S1}$ XOR R_{S2}
INC	A	R			$R \leftarrow R + 1$
DEC	B	R			$R \leftarrow R - 1$
ROTATE	C	R	n	0 or 1	$Rot_n R$
JUMP	D	R	n		IF $R_0 \neq R$ then PC = n, otherwise continue

Key: R_S, R_{S1}, R_{S2}: Hexadecimal address of source registers
R_D: Hexadecimal address of destination register
M_S: Hexadecimal address of source memory location
M_D: Hexadecimal address of destination memory location
n: hexadecimal number
d_1, d_2, d_3, d_4: First, second, third, and fourth hexadecimal digits