HW4 Solutions

T1

register addressing, immediate addressing,PC-relative, indirect,Base+offset

(a) ADD: operate;

register addressing for destination and source 1

register or immediate addressing for source 2

(b)JMP: control;   register addressing

(c)LEA: data movement;   immediate addressing

(d)LDR: data movement; Base+offset

(e)NOT: operate;   register addressing

T2

a. 8 bits.

> LC-3是按字寻址的，字长为16bits，256个字共需要 8 位地址 。

b. 6 bits.

> 注意是前后20个位置。

c. the offset is 6.

> 增量PC + offset = 10

T3

a. 0101 011 010 1 00100（AND R3，R2，#4）

b. 0101 011 010 1 01100（AND R3，R2，#12）

c. 0101 011 010 1 11111 （AND R3，R2，#-1）

d. 不可以，因为只有五位立即数

T4

(a) Add R1, R1, #0 => differs from a NOP in that it sets the CC's.(It will effect when the next instrction is BR.)

(b)BRnzp #1 => Unconditionally branches to one after the next address in the PC. There- fore no, this instruction is not the same as NOP.

(c)Branch that is never taken. Yes same as NOP.

T5

(a) 0001 011 010 1 00000 (ADD R3, R2, #0)

(b)1001 011 011 111111 (NOT R3, R3)
0001 011 011 1 00001 (ADD R3, R3, #1)
0001 001 010 0 00011 (ADD R1, R2, R3)

(c) 0001 001 001 1 00000 (ADD R1, R1, #0) or 0101 001 001 1 11111 (AND R1, R1, #-1)

(d)Can't happen. The condition where N=1, Z=1 and P=0 would require the contents of a register to be both negative and zero.

(e) 0101 010 010 1 00000 (AND R2, R2, #0)

T6

> 运用摩根律即可： a XOR b = (NOT a AND b) OR (NOT b AND a)

(1): 1001 100 001 111111 NOT R4, R1

(2): 0101 100 100 000 010 AND R4, R4, R2

(3):1001 101 010 111111 NOT R5, R2

(4):0101 101 100 000 001 AND R5, R5, R1

(5):1101 011 100 000 101 OR R3, R4, R5

T7

1110 001 000100000（LEA R1, 0x20 ）   R1 <- 0x3121

0010 010 000100000（LD R2, 0x20 ）   R2 <- Mem[0x3122] = 0x4566

1010 011 000100001（LDI R3, 0x20 ）   R3 <- Mem[Mem[0x3123]] = 0xabcd

0110 100 010 000001（LDR R4, R2, 0x1） R4 <- Mem[R2 + 0x1] = 0xabcd

1111 0000 0010 0101（TRAP 0x25 ）

T8

(a) LD: two, once to fetch the instruction, once to fetch the data. EXECUTE.

(b) LDI: three, once to fetch the instruction, once to fetch the data address, and once to fetch the data. EXECUTE.

(c) LEA: once, only to fetch the instruction. EVALUATE ADDRESS,OPERAND FETCH.

T9

```
x30ff    1110   0010 0000 0001    (LEA R1, #1) R1 <- 0x3101
x3100    0110  010 001 00 0010   (LDR R2, R1, #2) R2 <- 0x1482
x3101    1111  0000  0010  0101  (TRAP 0x25)
x3102    0001  0100  0100  0001
x3103    0001  0100  1000  0010
```

T10

说明R5 的低8位中有5位为1

```
x2FFF    0101 000 000 1 00000        AND R0,  R0,  #0
x3000    0101 111 111 1 00000        AND R7, R7,  #0
x3001    0001 110 111 1 00001        ADD R6, R7,  #1

x3002    0101 100 101 000 110        AND R4, R5, R6
x3003    0000 010 00000 0001       BRz, #1
x3004    0001 000 000 1 00001        ADD R0, R0,  #1    //若R5第i位为1，R0加1
x3005    0001 110 110 000 110        ADD R6, R6, R6
x3006    0001 111 111 1 00001        ADD R7, R7,  #1
x3007    0001 001 111 1 11000        ADD R1, R7,  #-8      //说明循环8次
x3008    0000 100 11111 1001       BRn, #-7

x3009    0101 111 111 1 00000         AND R7, R7,  #0
```

T11

（参考）d.MUL Ri, Rj, Rk

这几个指令的功能都可以用现有指令组合实现，d的组合实现相对其它选项复杂得多。

T12

	11	10	01	00
Mux 1 D[15:12]	3	2	1	0
Mux 2 D[11:8]	0	1	2	3
Mux 3 D[7:4]	3	1	2	0
Mux 4 D[3:0]	2	3	1	0

R/W	MDR	MAR
W	x72A3	00
W	x8FAF	11
R	x72A3	00
R	xFFFF	10
W	x732D	11
R	xFFFF	01
W	x37A3	01
R	x37A3	01
R	x732D	11

Contents after accesses:

00: x72A3

01: x37A3

10: xFFFF

11: x732D