HW04

一． 相同点：长度相同，都为 16 位

不同点：操作码不同，代表着两种指令

二．

1. 使用 ADD 指令：

```
ADD     R3,   R2,   #0
```

2. 三条指令如下：

```
NOT     R3,   R3,
ADD     R1,   R3,   #1
ADD     R1,   R2,   R3
```

3. 将 R1 中的内容加 0 再放入 R1 即可

```
ADD     R1,   R1,   #0
```

4. 不会。LC-3 的每条指令最多只会涉及一个目的寄存器，且内容用补码表示，每次只能为正、负、零中的一种情况

5. 将 R2 的内容和 0 进行与运算即可：

```
AND     R2,   R2,   #0
```

机器码格式：0101 010 010 1 00000

三．0001 为 ADD 指令

xFFF8 显然不能用寄存器中的值直接相加得到，所以考虑负立即数（5 位）扩展后前面补 1 得到

所以 x1000 处指令应为

0001 101 000 1 11000

四．

如下图所示：

PC、IR、MAR、MDR 必定会用到；计算地址时要对指令的后五位进行扩展，变成 16 位后进行加法运算得出地址；访问内存；访问源寄存器与目的寄存器；修改条件寄存器的值……

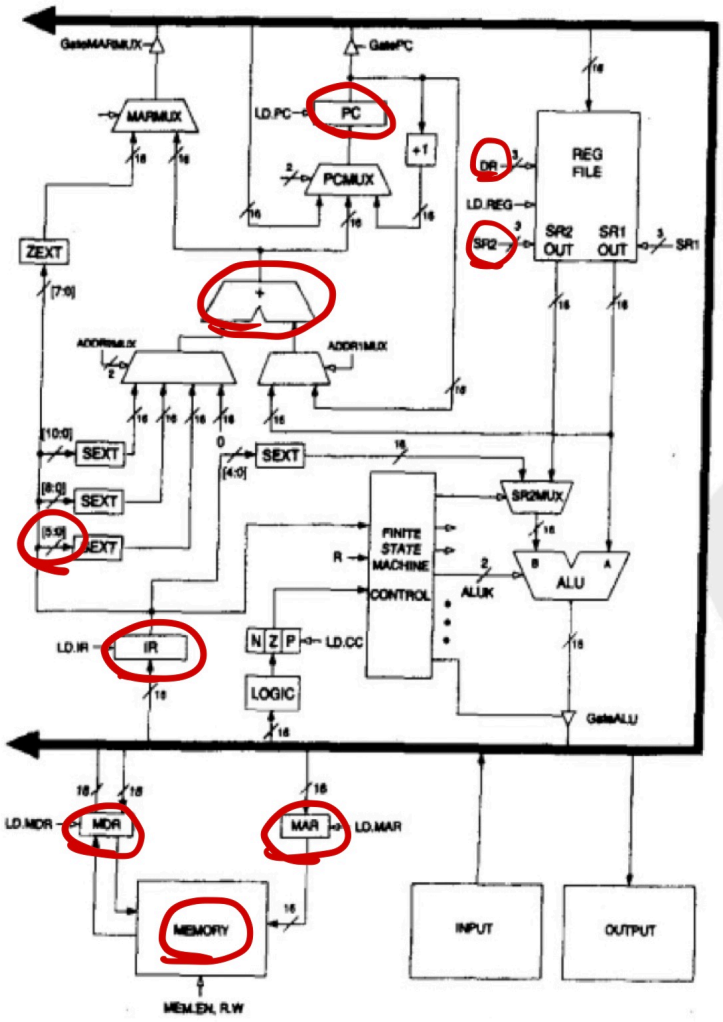

图5-18 LC-3的数据通路

五．

分析程序如下：

```
0101 000 000 1 00000     //清零 R0
0001 000 000 1 00101     //R0 <- R0+5
0010 001 000000100       //此时已增量 PC 值为 x3003，加 4 后为 x3007
                         //之后将 x3007 中的内容 x0004 放入 R1 中
0001 000 000 000 000         //R0 <- R0 + R0
0001 001 001 1 11111         //R1 <- R1-1
0000 001 111111101           //若 R1 为正数则 PC-3，（即回到第 5 行）
1111 0000 00100101           //陷入矢量 x25,终止程序
0000 000 000000100           //指令无效
```

1. 计算 5*2^4，并将结果放入寄存器 R0 中
2. R0 值为 x50，R1~R7 值均为 x0，PC 值为 x3006
   N,P 值为 0，Z 值为 1
3. 5+5+7+（5+5+6）* 3 +5+5+5 = 80

六．

分析如下：

```
x3000 0101 000 000 1 00000  //清零 R0
x3001 0010 001 011111110    //将 PC+x00FE 地址的内容放入 R1
x3002 0000 010 000000100    //若该内容为 0 则向下跳四步（程序结束）
x3003 0000 011 000000001    //若为正值则向下跳一步，
x3004 0001 000 000 1 00001  //R0 <- R0 + 1
x3005 0001 001 001 000 001  //R1 <- R1 + R1
x3006 0000 111 111111011    //无条件向上跳 5 步
x3007 1111 0000 0010 0101   //终止
```

七．

分析程序如下：

```
x3000 1110 000 011111111    //将 PC+x00FF = x3001 + x00FF = x3100 放入 R0 中
x3001 0001 000 000 1 00000  //R0 <- R0+0
x3002 0000 100 000000010    //若找到这个复负数，则向下跳两步，到 x3005
x3003 0001 000 000 1 00001  //R0 <- R0+1
x3004 0000 111 111111100    //无条件跳转，-4，回到 x3001
x3005 0011 000 001001010    //PC+x4A = x3006+x008A = x3050，把 R0 中的值放入
//x3050 中
x3006 1111 0000 0010 0101   //终止
```

故缺少的两条指令应为：

x3001 0001 000 000 1 00000

x3002 0000 100 000000010