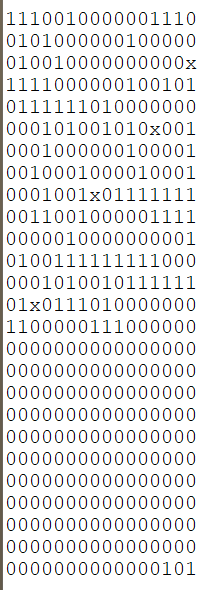
**实验报告——lab4 Reveal Yourself**

PB20151793 宋玮

**1.task1**

**原程序（有4个未知x）**



**分析：**

不妨假设程序从x3000开始

以下序号代表相应指令，如①代表第1条指令（1110010000001110）

①LEA指令，R2=x300f

②AND指令，R0清0

③JSR指令，x只能为0或1，若x为0，则程序跳转至结束（halt），不符合。因此x为1.

即本条完整指令为0100 1 00000000001。跳转至x3004。并将x3003保存至R7

④TRAP指令，即halt

⑤STR指令，将R7的值存入mem(R2)

⑥ADD指令，R2=R2+x001，根据最后R2的值仍为x300f，可知此条为R2=R2+1，即本条完整指令为0001 010 010 1 00001

⑦ADD指令，R0=R0+1

⑧LD指令，R1=mem（x3019）

⑨ADD指令，R1=R(x01)-1，x为1或0，因此R(x01)为R1或R5，考虑到最后R1变为0，因此此处为R1。即本条完整指令为0001 001 001 1 11111。R1=R1-1

⑩ST指令，R1的值存入x3019

⑪BRz指令，若R1为0，则跳转至下下条指令，即跳转至R2=R2-1

⑫JSR指令，跳转至x3004

⑬ADD指令，R2=R2-1

⑭01x0指令，可能为LDR指令或者JSR指令，根据后面12位可知不可能是JSR指令，否则会跳到程序之外，因此为LDR指令，即本条完整指令为0110 111 010 000000。R7=mem(R2)

⑮return指令，return R7

程序执行结束：R0 = 5, R1 = 0, R2 = 300f, R3 = 0

R4 = 0, R5 = 0, R6 = 0, R7 = 3003

因此四个x分别为：1,0,0,1

四条完整指令分别为：

0100 1 00000000001

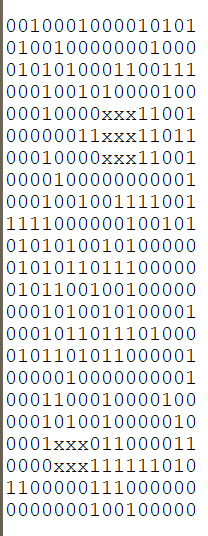
0001 010 010 1 00001

0001 001 001 1 11111

0110 111 010 000000

**2.task2**

**原程序（共有5个xxx）**



**分析：**

不妨假设程序从x3000开始

以下序号代表相应指令，如①代表第1条指令（0010001000010101）

①LD指令，R1=mem(x3016)=0000 0001 0010 0000

②JSR指令，程序跳转至x300a，R7=x3002

③AND指令，R2 = R1 AND 0000 0000 0000 0111

④ADD指令，R1=R2+R4

⑤ADD指令，R0=Rx-7

⑥BRp指令，若为正数，跳转，由于xxx只可能为111，否则将跳转至程序外，因此本条完整指令为0000 001 111111011，跳转至x3001 JSR指令

⑦ADD指令，R0=Rx-7

⑧BRn指令，若为负数，跳转至halt，结束程序

⑨ADD指令，R1=R1-7

⑩TRAP指令，即halt，结束程序

⑪（x300a）AND指令，R2清0

⑫AND指令，R3清0

⑬AND指令，R4清0

⑭ADD指令，R2=R2+1

⑮ADD指令，R3=R3+8

⑯AND指令，R5=R3 AND R1

⑰BRz指令，若为0，跳转至下下条指令，即R2=R2+R2

⑱ADD指令，R4=R2+R4

⑲ADD指令，R2=R2+R2，即R2左移

⑳ADD指令，Rx=R3+R3，根据分析可知，该条指令也是起了左移作用，因此Rx为R3，即xxx为011，也即本条完整指令为0001 011 011 000 011

BR指令，跳转至x300f（R5=R3 AND R1）

RETURN指令，return R7（x3002）

首先，根据hint，我们可以知道，从x300a开始的子程序是在求R1/8，因此第21条完整指令应该是BRp指令，即R3左移至1为最高位（为负数），之后停止循环。因此第21条完整指令为0000 101 111111010.

再回到主程序，应该是让R1一直减7，根据不同条件，再决定是跳转至子程序，还是继续减7，还是跳转至停止（halt），因此第⑤和⑦条指令中的xxx都为001，即完整指令都是0001 000 001 1 11001，R0=R1-7

程序执行结束时R1中的值即为x3016中存储数据mode 7 的值

因此5个xxx分别为：001, 111, 001，011，101

5条完整指令分别为：

0001 000 001 1 11001

0000 001 111111011

0001 000 001 1 11001

0001 011 011 000 011

0000 101 111111010