

# 实验报告——lab4 Reveal Yourself

PB20151793 宋玮

## 1.task1

原程序 (有 4 个未知 x)

[illegible]

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)

```
0010001000010101
0100100000001000
0101010001100111
0001001010000100
00010000xxx11001
00000011xxx11011
00010000xxx11001
0000100000000001
0001001001111001
1111000000100101
0101010010100000
0101011011100000
0101100100100000
0001010010100001
0001011011101000
0101101011000001
0000010000000001
0001100010000100
0001010010000010
0001xxx011000011
0000xxx111111010
1100000111000000
0000000100100000
```

**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,  $R5=R3 \text{ AND } R1$
- 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
- ㉑BR 指令, 跳转至 x300f ( $R5=R3 \text{ AND } R1$ )
- ㉒RETURN 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
```