Lab4 Reveal Yourself

姓名: 杨涛

学号: PB20020599

rec.txt

首先将机械码翻译为汇编语言用于理解,可以得到下面的结果

- 第一个x若为0则程序halt,不成立,故第一个x为1。
- 第二个x注意到为加立即数,一般都是加一,故暂时设x为0。
- 第三个x, 影响第二个寄存器, 若x=1则为R5, 而前面没有R5相关的程序, 可知x=0。
- 第四个x影响指令, 若x=0为JSR, 将会跳转到未知的地方, 不可能, 故x=1

回过头来考虑第二个x=0成立,用于栈指针的增加。

```
1110010000001110; LEA R2, LEBAL
0101000000100000; AND R0, R0, #0
010010000000000x; JSR Foo
                                     ->x=1?
1111000000100101;HALT
0111111010000000; Foo STR R7, R2, #0
000101001010x001;ADD R2,R2,#1
                                     ->x=0?
000100000100001;ADD R0,R0,#1
0010001000010001;LD R1,LEBAL2
0001001x01111111;ADD R1,R1,#-1
                                    ->x=0?
0011001000001111;ST R1,LEBEL2
0000010000000001;BRz LEBAL3
0100111111111000; JSR Foo
0001010010111111; LEBAL3 ADD R2, R2, #-1
01x0111010000000;LDR R7,R2,#0
                                   ->x=1?
1100000111000000; RET
000000000000000; LEBAL .BLKW 10
0000000000000000
0000000000000000
0000000000000000
00000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000000
0000000000000101; LEBAL2 .FILL #5
```

mod.txt

同样考虑将其翻译为汇编语言

先看下面的函数,使用掩码求第三位开始的每一位,即为逻辑右移三位,即R1/8。则可写出空缺分别为011和101。

而对于主函数,考虑 $x=8*(x/8)+x\%8\equiv(x/8+x\%8)(mod7)$,则易得主程序的缺省。

0010001000010101;LD R1,LEBAL 0100100000001000;Again JSR Foo

0101010001100111;AND R2,R1,#7;R2=R1%8

0001001010000100; ADD R1, R2, R4; R1=R1%8+R1/8

00010000xxx11001;ADD R0,R1,#-7 00000011xxx11011;BRp Again

->xxx=011 ->xxx=111

00010000xxx11001;ADD R0,R1,#-7

->xxx=011

0000100000000001;BRn skip

0001001001111001;ADD R1,R1,#-7

1111000000100101;skip TRAP

;Foo input:R1,output:R4=R1/8

0101010010100000; Foo AND R2, R2, #0

0101011011100000; AND R3, R3, #0

0101100100100000; AND R4,R4,#0

0001010010100001;ADD R2,R2,#1

0001011011101000;ADD R3,R3,#8

0101101011000001; LOOP AND R5, R3, R1

0000010000000001;BRz skip2

0001100010000100;ADD R4,R2,R4

0001010010000010; skip2 ADD R2,R2,R2

0001xxx011000011;ADD R3,R3,R3

0000xxx111111010;BRnp LOOP

1100000111000000; RET

->xxx=011

->xxx=101

000000100100000; LEBAL .FILL x120