

Lab04 Reveal Yourself

[实验目的]

- task1

根据给定的LC3机器码程序猜测其功能，从而补充其中丢失的4个bit

- task2

补充所给的LC3机器码程序中丢失的15个bit，使其能完成模7的功能

[实验过程]

task1:

- 根据机器码写出对应的汇编码

1		1110 010 000001110	; LEA R2, TARGET
2		0101 000 000 1 00000	; AND R0, R0, #0
3		0100 1 0000000000x	; JSR
4		1111 0000 00100101	; TRAP x25 (HALT)
5	QUEUE	0111 111 010 000000	; STR R7, R2, #0
6		0001 010 010 1 0x001	; ADD R2, R2, #
7		0001 000 000 1 00001	; ADD R0, R0, #1
8		0010 001 000010001	; LD R1, VALUE
9		0001 001 x01 1 11111	; ADD
10		0011 001 000001111	; ST R1, VALUE
11		0000 010 000000001	; BRz LOOP
12		0100 1 1111111000	; JSR QUEUE
13	LOOP	0001 010 010 1 11111	; ADD R2, R2, #-1
14		01x0 111010000000	;
15		1100 000 111 000000	; RET
16			
17	TARGET	0000000000000000	
18		0000000000000000	
19		0000000000000000	
20		0000000000000000	
21		0000000000000000	
22		0000000000000000	
23		0000000000000000	
24		0000000000000000	
25		0000000000000000	
26		0000000000000000	
27	VALUE	0000000000000101	

- 用C程序模拟其行为观察其功能

```

lab4.bin  lab4.cpp  x
ICS > lab4 > lab4.cpp > f()
1  #include<stdlib.h>
2  int R0,R1,R2,R3,R4,R5,R6,R7;
3  int value = 5;
4  int main()
5  {
6      R2 =(int) &value;
7      R7 = (int)(&value-22);
8      f();
9      system("pause");
10 }
11
12 void f()
13 {
14     *((int*)R2) = R7;
15     R2=R2+1;
16     R0=R0+1;
17     value=value-1;
18     if(value==0)
19     {
20         R2-=1;
21         R7 = *((int*)R2);
22     }
23     else f();
24 }

```

发现其实它的功能就是把value中存的值最后放入R0

- 补充完整后如下

```

lab4.bin  x  Generate  Sim
ICS > lab4 > lab4.bin
1  1110 010 000001110    ;LEA R2,TARGET
2  0101 000 000 1 00000    ;AND R0,R0,#0
3  0100 1 00000000001    ;JSR QUENE
4  1111 0000 00100101    ;TRAP x25(HALT)
5  QUEUE 0111 111 010 000000    ;STR R7,R2,#0
6  0001 010 010 1 00001    ;ADD R2,R2,#
7  0001 000 000 1 00001    ;ADD R0,R0,#1
8  0010 001 000010001    ;LD R1,VALUE
9  0001 001 001 1 11111    ;ADD R1,R1,#-1
10 0011 001 000001111    ;ST R1,VALUE
11 0000 010 000000001    ;BRz LOOP
12 0100 1 11111111000    ;JSR QUEUE
13 LOOP 0001 010 010 1 11111    ;ADD R2,R2,#-1
14 0110 111 010 000000    ;LDR R7,R2,#0
15 1100 000 111 000000    ;RET
16
17 TARGET 0000000000000000
18 0000000000000000
19 0000000000000000
20 0000000000000000
21 0000000000000000
22 0000000000000000
23 0000000000000000
24 0000000000000000
25 0000000000000000
26 0000000000000000
27 VALUE 0000000000000101

```

task2:

思路:

求一个数模7的余数，用小学知识可以定义为一直减7直到减不动为止，但这样太慢，更快的方法是采用这样一个知识

$$l = 8 * m + n \quad (0 \leq n < 8)$$

$$l \bmod 7 = (m + n) \bmod 7$$

观机器码，发现其正采用此方法，可根据给定的信息，补充完代码

- 根据机器码写出对应的汇编码

```
> lab4.bin
SP      0010 001 000010101      ;LD R1,TARGET
        0100 1 00000001000      ;JSR QUEUE
        0101 010 001 1 00111      ;AND R2,R1,#7
        0001 001 010 0 00 100      ;ADD R1,R2,R4
        0001 000 0xx x11001      ;ADD R0, ,
        0000 001 1xxx11011      ;BRp
        0001 000 0xx x11001      ;ADD R0, ,
        0000 100 000000001      ;BRn NEXT
        0001 001 001 1 11001      ;ADD R1,R1,#-7
NEXT     1111 000000100101      ;TRAP x25(HALT)
QUEUE    0101 010 010 1 00000      ;AND R2,R2,#0
        0101 011 011 1 00000      ;AND R3,R3,#0
        0101 100 100 1 00000      ;AND R4,R4,#0
        0001 010 010 1 00001      ;ADD R2,R2,#1
        0001 011 011 1 01000      ;ADD R3,R3,#8
ITER     0101 101 011 000 001      ;AND R5,R3,R1
        0000 010 000000001      ;BRz LOOP
        0001 100 010 0 00 100      ;ADD R4,R2,R4
LOOP     0001 010 010 0 00 010      ;ADD R2,R2,R2
        0001 xxx 011 0 00 011      ;ADD ,R3,R3
        0000 xxx 111111010      ;BR ,ITER
        1100 000 111 000000      ;JMP R7

TARGET   0000000100100000
```

- 用C程序模拟其行为观察其功能

```
ICS > lab4 > lab4.cpp > main()
1  #include<stdlib.h>
2  #include<stdio.h>
3  int R0,R1,R2,R3,R4,R5,R6,R7;
4  int target = 0b100100000;
5  int main()
6  {
7      R1 = target;
8      s();
9  }
10 void f()
11 {
12     R2 = 1;
13     R3 = 0b1000;
14     R4 = 0;
15     h();
16 }
17 void h()
18 {
19     R5 = R1&R3;
20     if(R5!=0) R4 = R2+R4;
21     R2 = R2*2;
22     R3 = R3*2;
23     if(R3!=0) h();
24 }
25
26
27 void s()
28 {
29     f();
30     R2 = R1&(0b111);
31     R1 = R2+R4;
32     R0 = R1-7;
33     if(R0>0) s();
34     R0 = R1-7;
35     if(R0<0) system("pause");
36     R1 = R1-7;
37     system("pause");
38 }
39
```

- 补充完整后如下

		0010 001 000010101	;LD R1,TARGET
SP		0100 1 00000001000	;JSR QUEUE
		0101 010 001 1 00111	;AND R2,R1,#7
		0001 001 010 0 00 100	;ADD R1,R2,R4
		0001 000 001 1 11001	;ADD R0,R1,#-7
		0000 001 111111011	;BRp SP
NEXT		0001 000 001 1 11001	;ADD R0,R1,#-7
		0000 100 000000001	;BRn NEXT
		0001 001 001 1 11001	;ADD R1,R1,#-7
		1111 000000100101	;TRAP x25(HALT)
		0101 010 010 1 00000	;AND R2,R2,#0
QUEUE		0101 011 011 1 00000	;AND R3,R3,#0
		0101 100 100 1 00000	;AND R4,R4,#0
		0001 010 010 1 00001	;ADD R2,R2,#1
		0001 011 011 1 01000	;ADD R3,R3,#8
		0101 101 011 000 001	;AND R5,R3,R1
ITER		0000 010 000000001	;BRz LOOP
		0001 100 010 0 00 100	;ADD R4,R2,R4
		0001 010 010 0 00 010	;ADD R2,R2,R2
		0001 011 011 0 00 011	;ADD R3,R3,R3
		0000 001 111111010	;BRp,ITER
LOOP		1100 000 111 000000	;JMP R7
TARGET		0000000100100000	

[实验总结]

这次实验主要锻炼了读机器码，理解机器码实现功能的能力，同时对理解高级语言中函数，指针的具体实现以及它们与地址之间的联系有了更深的理解，同时锻炼了对机器代码功能的分析，每一步具体实现怎样的功能，理解它们与高级语言相比的不同之处。