LAB_04 REVEAL YOURSELF 实验报告

PART_01 阅读代码并补全

首先我们将助教给出的机器码转换为汇编语言进行分析。

```
x3000   LEA R2,#14;          将地址x300F存入R2（如果首位为x3000）
x3001   AND R0,R0,#0;        将R0清零
x3002   JSR #?;              此处为0或者1，如果是0则无意义(会直接HALT)
x3003   TRAP x25;            HALT
x3004   STR R7,R2,#0;        把R7存入mem[R2]（实际上是mem[x300F],初始是x3003）
x3005   ADD R2,R2,#?;        此处为1或者#9(实际上是对地址的操作，所以很明显也是1)
x3006   ADD R0,R0,#1;        R0自加1
x3007   LD R1,#17;           将mem[x3019](x0005)赋值给R1
x3008   ADD R1,R?,#-1;       R5或者R1(R5从没有出现过)
x3009   ST R1,#15;           将R1的值存入x3019
x300A   BRz #1;              如果R1为0则进入下下条ADD指令
x300B   JSR #-8;             回到STR指令
x300C   ADD R2,R2,#-1;       R2自减1
x300D   ?(LDR R7,R2,#0);     LDR R7,R2,#0或者JSR x180（这条指令过于离谱），效果是
R7=men[R2]
x300E   RET;                回到JSR的下一次指令
x300F   .FILL x0;
x3010   .FILL x0;
x3011   .FILL x0;
x3012   .FILL x0;
x3013   .FILL x0;
x3014   .FILL x0;
x3015   .FILL x0;
x3016   .FILL x0;
x3017   .FILL x0;
x3018   .FILL x0;
x3019   .FILL x5;
```

通过对汇编代码的分析，我们可以对代码的功能进行猜测，得到以下流程。

- 将各寄存器清零。将mem[x3019]赋值给R1(5)。在R2中存入x300F.同时把x3003存入对应地址
- R1减1，R0加1，直到R1=0。此时R0=5.
- R7读入R2对应地址的值。

  所以得到正确代码如下。

```
LEA R2,#14;
AND R0,R0,#0;
JSR #1;
TRAP x25;
STR R7,R2,#0;
ADD R2,R2,#1;
ADD R0,R0,#1;
LD R1,#17;
ADD R1,R1,#-1;
```

```
ST R1,#15;
BRz #1;
JSR #-8;
ADD R2,R2,#-1;
LDR R7,R2,#0;
RET;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x0;
.FILL x5;
```

PART_02 分析代码含义并补全（模7）

如上文所述，我们把机器码转换成汇编码并加以分析。

```
LD R1 x15;  存入x120
JSR #8;     开始halt之后的部分
AND R2,R1,#7;   R2=R0mod8
ADD R1,R2,R4;   R1=R2+R4
ADD R0,?,#-7;   R0=R? -7
BRp #-5;        如果是正数重新进行上述操作
ADD R0,R?,#-7;  R0=R?-7;
BRn #1;         如果小于零则不需要减7
ADD R1,R1,#-7;  R1=R1-7
TRAP x25;   HALT
AND R2,R2,#0;   R2清零
AND R3,R3,#0;   R3清零
AND R4,R4,#0;   R4清零
ADD R2,R2,#1;   R2=R2+1
ADD R3,R3,#8;   R3=R3+8
AND R5,R3,#1;   R5=R3&R1
BRz #1;         R5为0则不对R4操作
ADD R4,R2,R4;   R4=R2+R4;
ADD R2,R2,R2;   R2=R2*2
ADD ?,R3,R3;    ;很明显这是通过对R3进行操作判断
                ;循环是否结束
BR ? #-6;       ;如果R3是正/负数，重新进行上述操作（应该是p）
RET;            执行取模操作
.FILL x120;
```

分析上述代码，可以得到代码思路如下。

观察代码得到本次代码的思想为将待求数存入R1中，依次判断R1与8的正整数倍的与，若为1，则得到该数与7求模的值依次为1，2，4,...,相加得到代码如下。

```
.orig x3000;
LD R1 x15;
```

```
JSR #8;
AND R2,R1,#7;
ADD R1,R2,R4;
ADD R0,R1,#-7;
BRp #-5;
ADD R0,R1,#-7;
BRn #1;
ADD R1,R1,#-7;
TRAP x25;
AND R2,R2,#0;
AND R3,R3,#0;
AND R4,R4,#0;
ADD R2,R2,#1;
ADD R3,R3,#8;
AND R5,R3,R1;
BRz #1;
ADD R4,R2,R4;
ADD R2,R2,R2;
ADD R3,R3,R3;
BRp #-6;
RET;
.FILL x120;
.END
```