ICS 2021　LAB04

实验报告

PB20111699　吴骏东

2021.12.23

一．任务一

这是一个有些意义不明的程序.....至少从翻译出来的结果看十分让人迷惑。由于程序中空缺的 bit 位为单独出现的，因此理论上可以通过穷举法直接暴力破解。

在通读程序并初步翻译后，我们可以做出如下分析：对于第一处代码“JSR #?”，可知该跳转指令会将增量 PC 存储到 R7 中，并跳转到偏移后的位置。由于该代码下一行即为 halt 指令，所以这里必定为“JSR　#1”。对于第二处代码“ADD　r2, r2, #?”，这里可供选择的数为 1 或 9。通读整段指令后我们可以推断出这里应为“ADD　r2, r2, #1”。对于第三处“ADD　r1, r？, #-1”，可供选择的数为 1 或 5.但整段程序中没有出现过 r5，所以结果为“ADD　r1, r1, #-1”。对于最后一段指令，我们并不知道它的操作数，但由于只缺少一位我们可以大胆进行推测其为 LDR(0110)。最终程序也验证了我们的猜想。

以下为完整代码。

```
.orig x3000
LEA        r2, #14
AND        r0, r0, #0
JSR        #1
halt
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 #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #0
.FILL #5

.END
```

二．任务二

这个程序相对前者而言功能性更强。我们已知它可用来计算模 7 的余数。但在计算过程中我们惊人地发现他竟然先与 7 做按位与(这是模 8 的操作)程序中包含了一个函数，它的功能是计算 R1 中存储的值整除 8 的结果，并将其存储在 R4 中。执行完函数之后，程序会计算 R1 中存储的值模 8 的结果，并将其存在 R2 中。接下来是代码段的核心部分。这里我们用 n 代表目标整数，x 代表整除 8 的结果，y 代表模 8 电结果。有：

$$\mathrm{n} = 8x + y$$

我们令 k = x + y，若 k = 7，则有：

$$n = 8(7 - y) + y = 7(8 - y)$$

所以 n 整除 7 等价于 k 整除 7。若 k = 7t + h，则有

$$n = 8(7t + h - y) + y = 7(8t + h - y) + h$$

所以 n ≡ k (mod7)。因而我们可以对 k 重复 n 的操作，直到结果 k<8 为止。最终得到的结果 k 范围在 1~7 ，刚好为 n 模 7 的结果(k = 7 相当于整除)。

以下为完整代码。

```
.orig x3000
LD          r1, number
start1
      JSR         fun
      AND          r2, r1, #7
      ADD          r1, r2, r4
      add          r0, r1, #-7
      brp          start1
add         r0, r1, #-7
BRn         fend
ADD          r1, r1, #-7
fend
      halt
fun
      and          r2, r2, #0
      and          r3, r3, #0
      and          r4, r4, #0
      add          r2, r2, #1
      add          r3, r3, #8    ;r2 = 1, r3 = 8, r4 = 0
      start2
            and          r5, r3, r1
            brz          #1
            add          r4, r2, r4

            add          r2, r2, r2
            add          r3, r3, r3
            brnp         start2
      RET
number
      .fill     #-4 ;#288
.end
```

三. 小结

这次的实验还是很有意思的。第二题的创新思路确实是一个很好的启发。希望这样的题目可以多一点。