lab5

PB21111618 侯超群

1.Report name: **lab5**

2.Purpose:

编写题目所要求的用户程序，以及键盘中断服务例程；
a)题目要求根据特定输入解决汉诺塔问题，其将汉诺塔参数记作N，并初始化为xFFFF，存储在内存的x3FFF处；
b)用户程序要求从x3000开始，并打印学号PB21111618，如：PB21111618 PB21111618 PB21111618 ...... 等待参数N变成有效值时调用HONOI子例程解决问题；
c)键盘中断服务例程从x1000开始，检查键入的是否为十进制，若不是，则从新行打印"< the input character > is not a decimal digit."并换行以RTI结束，重新循环打印学号；若是，则从新行打印"< the input character > is a decimal digit."并保存到x3FFF处，并换行以RTI终止；

3.Principles:

1)如何制定算法：
a)首先对于汉诺塔算法，采取递归形式，先深入到最底层N减至0时，此时H(n)为0，再根据之前保存的N值即R3记录需要跳出的次数，逐层跳出计算H(n)；
b)对于学号输出采取了实验要求中所给的延迟程序，同时根据N的初始值xFFFF加1判断是否为0，是则继续循环，否则调用HONOI子例程解决问题
c)对于多位数显示，通过不断减100至负数，得到其百位数，再加上100还原至十位和个位，之后对于十位和个位的操作同理；再根据各位数是否为零，利用BRz跳转，选择是否显示；
d)对于输入是否为有效值的判断，通过比较其是否满足ASCLL码范围，分别跳转输出，需要则储存N值，在键盘中断服务例程中用到R0,R1,R2三个寄存器，因此需要保存并恢复其值；

2)如何编写程序
在编写程序中，采取了模块化编程，开始分别对学号显示，汉诺塔问题求解进行编程；
在完成学号显示后编写了键盘中断服务例程，实现对于N有效值的判断及储存；
同时通过固定汉诺塔问题的N值，在对该模块的多位数输出显示做了优化，最后整合至用户程序，通过对获取N值的稍加调整完成该程序的编写；

3)如何设计测试用例
在编写程序过程中，主要测试在于终端显示以及对于汉诺塔问题的测试；
终端显示特定值时可能陷入无限循环或计算错误，测试经常在于JSR与RET处出现问题；
对于汉诺塔问题的测试，采取在模块中固定N值，对于模块测试，可以适当减少问题发生的可能；

4.Procedure:

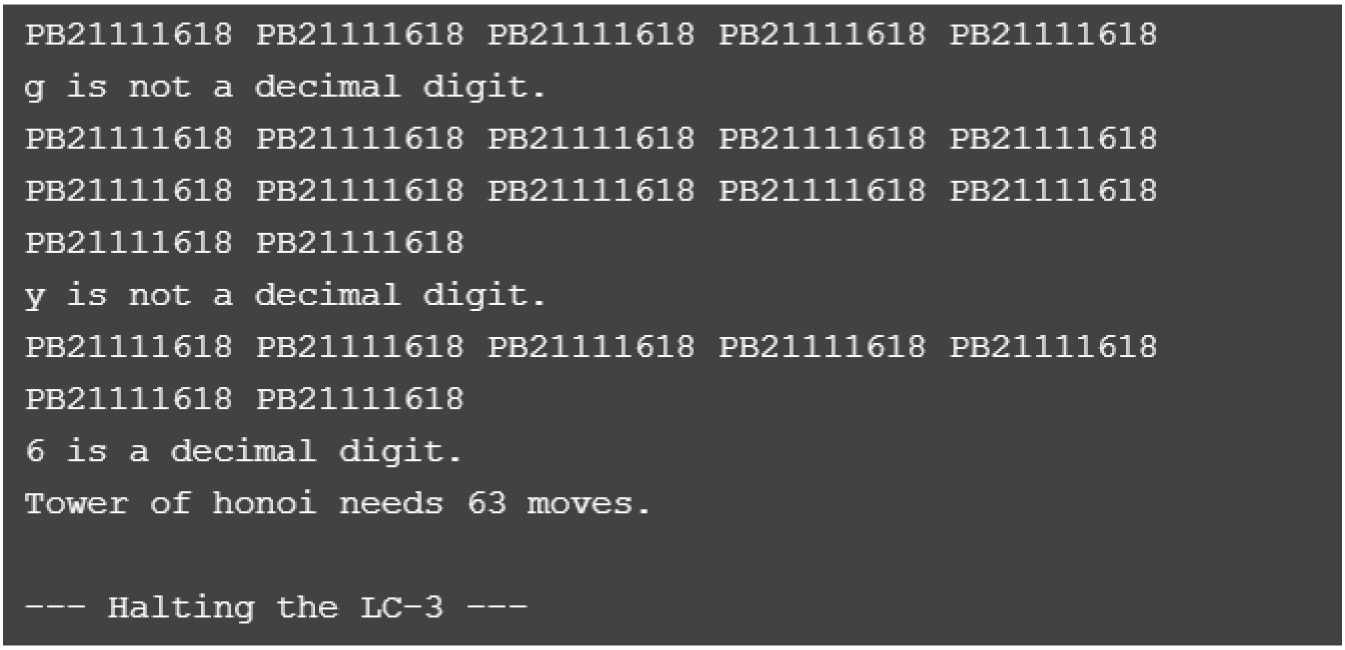
```
.ORIG x800
       ; (1) Initialize interrupt vector table.
       LD R0, VEC
       LD R1, ISR
       STR R1, R0, #0
```

```
        ; (2) Set bit 14 of KBSR.
        LDI R0, KBSR
        LD R1, MASK
        NOT R1, R1
        AND R0, R0, R1
        NOT R1, R1
        ADD R0, R0, R1
        STI R0, KBSR

        ; (3) Set up system stack to enter user space.
        LD R0, PSR
        ADD R6, R6, #-1
        STR R0, R6, #0
        LD R0, PC
        ADD R6, R6, #-1
        STR R0, R6, #0
        ; Enter user space.
        RTI

VEC     .FILL x0180
ISR     .FILL x1000
KBSR    .FILL xFE00
MASK    .FILL x4000
PSR     .FILL x8002
PC      .FILL x3000
        .END

        .ORIG x3000
        ; *** Begin user program code here ***
PRIN    LEA R0, NUM
        TRAP x22        ;"PB21111618 "
        JSR DELAY
        LDI R0, N
        ADD R0, R0, #1
        BRz PRIN        ;根据N判断是否循环输出
        BRnzp HONOI

DELAY   ST R1, SAVERR1  ;延迟便于显示
        LEA R1, COUNT
REP     ADD R1, R1, #-1
        BRp REP
        LD R1, SAVERR1
        RET

HONOI   LDI R1, N       ;HONOI子例程解决问题
        ADD R3, R1, #0
        ADD R1, R1, #1
        JSR SKIP
        BRnzp DONE

SKIP    ADD R1, R1, #-1
        BRz ZERO
        JSR SKIP
        ADD R2, R2, R2
        ADD R2, R2, #1
        ADD R3, R3, #-1 ;根据R3即N的值记录跳出次数
        BRz DONE
        RET

ZERO    AND R2, R2, #0  ;减至0时则开始跳出递归
        RET

DONE    LD R4, HUNDRED
        LD R5, TEN
        ADD R3, R2, #0  ;R3表示计算结果

        AND R1, R1, #0
HUND    ADD R1, R1, #1
        ADD R3, R3, R4
        BRzp HUND
        ADD R1, R1, #-1 ;循环减100得到百位数

        NOT R4, R4
        ADD R4, R4, #1
        ADD R3, R3, R4  ;恢复到减完百位

        AND R2, R2, #0
TENC    ADD R2, R2, #1
        ADD R3, R3, R5
        BRzp TENC
        ADD R2, R2, #-1 ;循环减10得到十位数
        ADD R3, R3, #10 ;恢复到个位数R3

        LEA R0, RESUL
        TRAP x22    ;“Tower of honoi needs ”

        ADD R6, R1, #0  ;输出百位
        BRz TENP
        JSR APPE

TENP    ADD R6, R2, #0  ;输出十位
        BRz BITP
        JSR APPE

BITP    ADD R6, R3, #0  ;输出个位
        JSR APPE
        BRnzp OVERD

APPE    ADD R0, R6, #15
        ADD R0, R0, #15
        ADD R0, R0, #15
        ADD R0, R0, #3  ;转化为ASCLL码输出
        TRAP x21
        RET

OVERD   LEA R0, RESULT
        TRAP x22    ;“ moves.”
        HALT
```

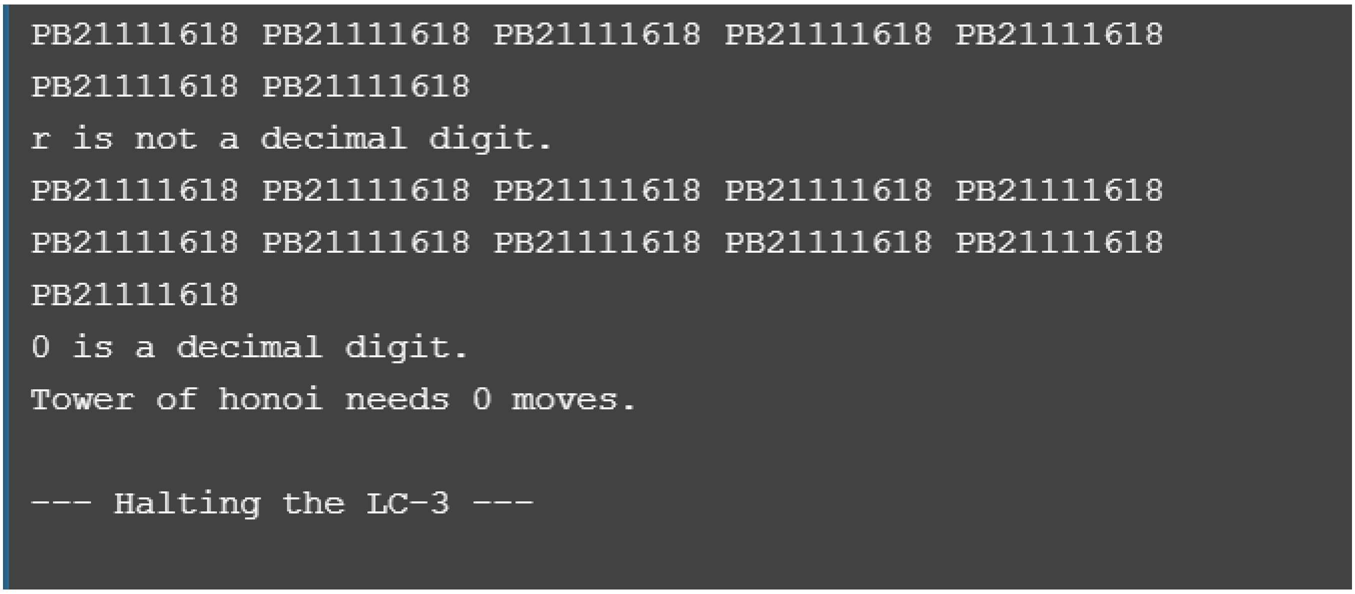
```
HUNDRED .FILL #-100
TEN .FILL #-10

SAVERR1 .FILL x0000
COUNT .FILL #2500
NUM .STRINGZ "PB21111618 "
RESUL .STRINGZ "Tower of honoi needs "
RESULT .STRINGZ " moves."
N .FILL x3FFF
        ; *** End user program code here ***
        .END

        .ORIG x3FFF
        ; *** Begin honoi data here ***
HONOI_N .FILL xFFFF
        ; *** End honoi data here ***
        .END

        .ORIG x1000
        ; *** Begin interrupt service routine code here ***
        ST R0, SAVER0
        ST R1, SAVER1
        ST R2, SAVER2
        TRAP x20
        ADD R2, R0, #0
        AND R0, R0, #0
        ADD R0, R0, x000A
        TRAP x21    ;显示换行
        ADD R0, R2, #0
        TRAP x21    ;显示N

        ADD R1, R0, #-15
        ADD R1, R1, #-15
        ADD R1, R1, #-15
        ADD R1, R1, #-3 ;根据ASCLL码判断是否为十进制数
        BRn NONUM
        ADD R1, R1, #-10
        BRzp NONUM
        BRnzp YESNUM

YESNUM  LEA R0, YESD
        TRAP x22    ;" is a decimal digit."
        ADD R1, R1, #10
        STI R1, HN  ;当N有效时更改N的值
        BRnzp OVER

NONUM   LEA R0, NOD
        TRAP x22    ;" is not a decimal digit."

OVER    AND R0, R0, #0
        ADD R0, R0, x000A
        TRAP x21    ;显示换行
        LD R0, SAVER0
        LD R1, SAVER1
        LD R2, SAVER2
        RTI

SAVER0 .FILL x0000
SAVER1 .FILL x0000
SAVER2 .FILL x0000
YESD .STRINGZ " is a decimal digit."
NOD .STRINGZ " is not a decimal digit."
HN .FILL x3FFF
        ; *** End interrupt service routine code here ***
        .END
```

5.Result of test:

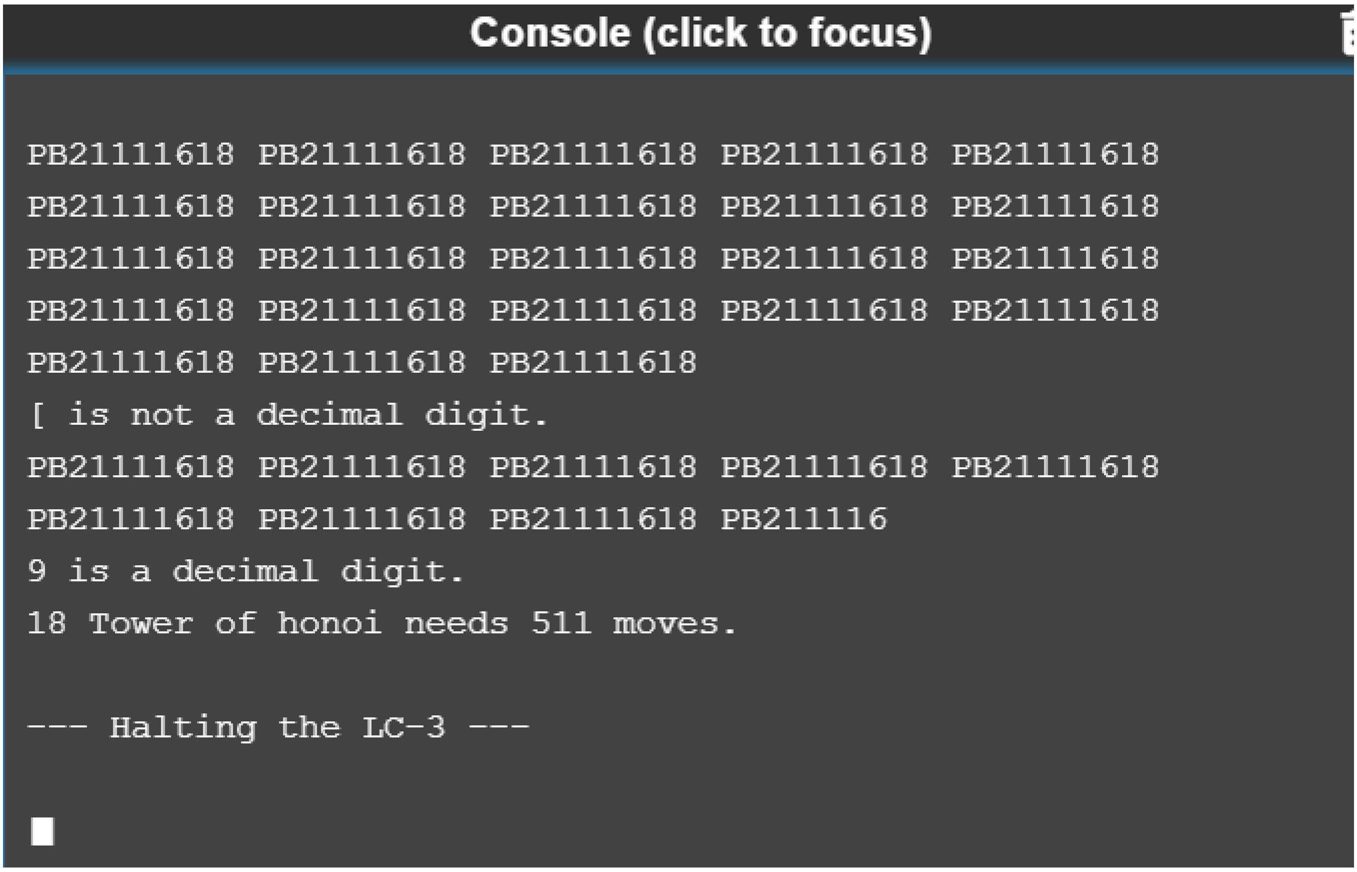
部分评测如下



```
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
g is not a decimal digit.
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618
y is not a decimal digit.
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618
6 is a decimal digit.
Tower of honoi needs 63 moves.

--- Halting the LC-3 ---
```



```
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618
r is not a decimal digit.
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618
0 is a decimal digit.
Tower of honoi needs 0 moves.

--- Halting the LC-3 ---
```



```
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618
[ is not a decimal digit.
PB21111618 PB21111618 PB21111618 PB21111618 PB21111618
PB21111618 PB21111618 PB21111618 PB211116
9 is a decimal digit.
18 Tower of honoi needs 511 moves.

--- Halting the LC-3 ---
```