

# 计算机系统概论 A 实验报告



实验题目：\_\_\_\_\_实验五\_\_\_\_\_

学生姓名：\_\_\_\_\_牛庆源\_\_\_\_\_

学生学号：\_\_\_\_\_PB21111733\_\_\_\_\_

完成日期：\_\_\_\_\_2022.12.30\_\_\_\_\_

### 【实验题目】

不断循环输出自己的学号，并在输入合法的汉诺塔输入后中断重复过程，并给出该输入的解。

### 【实验目的】

1. 学会用汇编写出循环输出的代码。
2. 学会写一个键盘中断处理程序。
3. 使用汇编写一个汉诺塔程序。

### 【实验原理】

1. 首先我们需要读入键盘键入的内容，并判断是否合法，从而使程序输出学号或者是汉诺塔问题的解。
2. 首先循环输出学号这一部分，为了避免输出过快，采取延迟的方式，每延迟 COUNT 后输出一次。

```
DELAY  ST R1, SAVER1
        LD R1, COUNT; 每次延迟count后输出一次
REP     ADD R1, R1, #-1
        BRp REP
        LD R1, SAVER1
        RET
```

```
COUNT .FILL x2500;延迟量
```

3. 用中断服务例程的起始地址初始化中断向量表。我们用值 x1000（键盘中断服务）初始化内存位置 x0180（+键盘中断向量后）。设置 KBSR 的第 14 位以启用中断。将 PSR 和 PC 处理堆栈，使得它可以使用 RTI 指令跳转到 x3000 的用户程序。

4. 打印学号为无限循环，先检测输入，若为合法（数字）则处理汉诺塔，否则跳转到学号输出例程 print，经过 delay 例程延迟输出频率后回到 loop 循环，继续检测键入。

```

LOOP      LD R0, ADDR ;
          LDR R1, R0, #1
          ADD R1, R1, #0
          BRz PRINT;若R1为合法输入(数字)则继续，否则跳转到输出学号
          LDR R1, R0, #0;R1为数字
          AND R3, R3, #0;R3 = 0
          JSR HANOI;//
          LEA R0, OUT3
          TRAP x22
          JSR OUTPUT
          LEA R0, OUT4
          TRAP x22
          HALT

PRINT     LEA R0, idOUT
          TRAP x22
          JSR DELAY;先经过delay例程避免循环输出过快
          BR LOOP;输出学号后继续跳转到循环过程

```

5. 由给出的汉诺塔递归的公式：

$$H(n) = \begin{cases} 0 & ,n = 0 \\ 2H(n-1) + 1 & ,n > 0 \end{cases}$$

C 语言写出一个函数：

```

int Hanoi(int n)
{
    if(n == 0)
        return 0;

    return 2 * Hanoi(n - 1) + 1;
}

```

改写为汇编语言

```
HANOI    STR R7, R6, #0
          ADD R1, R1, #-1;R1为从键盘读入的合法数字
          BRn RECURSIVE
          STR R1, R6, #-1
          ADD R6, R6, #-2
          JSR HANOI
          ADD R6, R6, #1
          LDR R7, R6, #0
          ADD R3, R3, R3
          ADD R3, R3, #1

RECURSIVE ADD R6, R6, #1;指针移动
          LDR R1, R6, #0;取R1
          RET
```

上述是栈操作，对地址和内容入栈然后出栈，进行递归即可。

6. 循环输出学号和汉诺塔都写好之后，接下来写中断部分。首先检查是否从键盘读入了 R1，若没有，则跳出，若读入了，则检查是否是数字输入，若不是则跳出，是的话存入汉诺塔处理例程中，更新数字，并且中断循环输出学号的例程。具体的每一步骤注释标注。

```
OUTPUT2  LD R0, LF
          TRAP x21
          ADD R0, R1, #0;R0 = R1
          TRAP x21;
          LEA R0, OUT1
          TRAP x22
          BR EXIT

EXIT      RTI
```

```

LDI R1, KBSR1;检查是否从键盘读入内容
LD R2, MASK_1;xc000
AND R1, R1, R2
BRz EXIT

LDI R1, KBDR1;读入了内容
LD R2, MASK_2;x00ff
AND R1, R1, R2;取后8位
LD R2, NUM0;数字0的ASCII码
ADD R2, R1, R2;判断是否大于0的ASCII码
BRn OUTPUT2;从键盘读取的内容不进入汉诺塔例程
LD R2, NUM9;数字9的ASCII码
ADD R2, R1, R2;判断是否小于9的ASCII码
BRp OUTPUT2
LD R0, LF;是合法的数字输入，则存到Hanoi_n并且更新
TRAP x21
ADD R0, R1, #0; R0 = R1
TRAP x21
LEA R0, OUT2
TRAP x22
LD R3, ADDR1;汉诺塔输入存放
LD R2, NUM0
ADD R1, R1, R2;// R1 = R1 - 48此时从ASCII码转化为数字
STR R1, R3, #0
LDR R1, R3, #1;以下将number都设为1
ADD R1, R1, #1
STR R1, R3, #1
RET

```

7. 关于 ASCII 码和符号码，从百位开始处理，然后十位。参照了网上的代码。

```

OUTPUT  ST R0, SAVER0
        ST R1, SAVER1
        ST R2, SAVER2
        ADD R2, R3 #0
        LD R1, NUM100
        NOT R1, R1
        ADD R1, R1, #1
        AND R0, R0, #0

MINUS1  ADD R0, R0, #1
        ADD R2, R2, R1
        BRzp MINUS1
        NOT R1, R1
        ADD R1, R1, #1
        ADD R2, R2, R1
        ADD R0, R0, #-1;循环多了一次
        BRnz JUMP;跳过百位
        LD R1, zero
        ADD R0, R0, R1
        TRAP x21

JUMP    LD R1 NUM10
        AND R0, R0, #0
        NOT R1, R1
        ADD R1, R1, #1

```

```

MINUS2  ADD R0, R0, #1
        ADD R2, R2, R1
        BRzp MINUS2
        NOT R1, R1
        ADD R1, R1, #1
        ADD R2, R2, R1
        ADD R0, R0, #-1
        BRnz JUMP2;跳过十位
        LD R1, zero
        ADD R0, R0, R1
        TRAP x21

JUMP2   ADD R0, R2, #0
        LD R1, zero
        ADD R0, R0, R1
        TRAP x21
        LD R0, SAVER0
        LD R1, SAVER1
        LD R2, SAVER2
        RET

```

```

SAVER0  .FILL x0000
SAVER1  .FILL x0000
SAVER2  .FILL x0000

NUM100  .FILL #100
NUM10   .FILL #10
zero    .FILL x30

```

【实验结果】

如图：

```
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733
a is not a decimal number!
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733
i is not a decimal number!
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733 PFB21111733
PFB21111733 PFB21111733 PFB21111733 PFB21111733
6 is a decimal number!
Tower of hanoi needs 63 moves.
```

--- Halting the LC-3 ---