

ICS_Lab5_Report

Xiaoma

2022 年 12 月 22 日

实验目的

使用 LC-3 汇编语言实现根据输入值计算汉诺塔问题的解。

- 通过中断驱动 I/O 键入一个字符值
- 判断键入字符是否为 0 – 9 的整数，若是则计算汉诺塔问题的解，若不是则输出相关信息
- 在没有键盘输入时，程序将无限循环输出学生学号

实验原理

汉诺塔问题

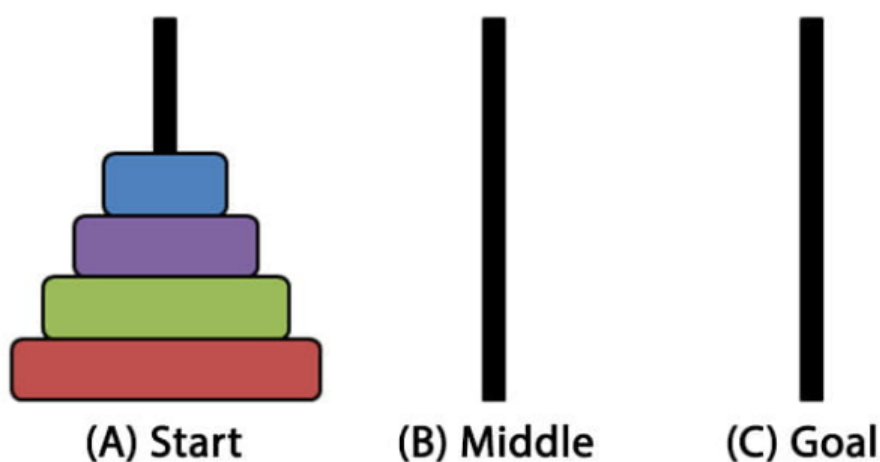
汉诺塔是一个数学问题，由三根杆子和许多不同直径的圆盘组成，这些圆盘可以处于任何杆子上。问题开始时，圆盘按大小递减的顺序堆叠在一根杆上，最小的在顶部，该问题的目标是将整个堆栈移动到最后一根杆上，移动时需遵守以下规则

- 一次只能移动一个圆盘
- 每次移动只能取出杆上最顶部的圆盘，并将其放置于另一根杆的顶部或空杆上

- 任何圆盘都不能放在比其小的圆盘上面

通过计算我们可以得到汉诺塔问题的求解公式

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



判断键入值是否为有效值

已知从键盘键入的值的类型为字符型，则当字符值为 0–9 时，其整型值实际为 48–57，故可以通过确定输入值是否在给定范围内来判断其是否为有效值，若为有效值，则中断服务程序执行结束后，用户程序计算汉诺塔问题的解，若不是有效值，则用户程序继续循环输出学号。

实验步骤

用户程序

循环输出学号的实现

首先使用 `.STRINGZ` 来存放需要输出的字符串，即

```
1 StNum .STRINGZ "PB2006xxx\n"
```

然后使用 *.FILL* 使用标签来存放字符串的指针，则可以使用 *trapx22* 指令来输出该字符串，即

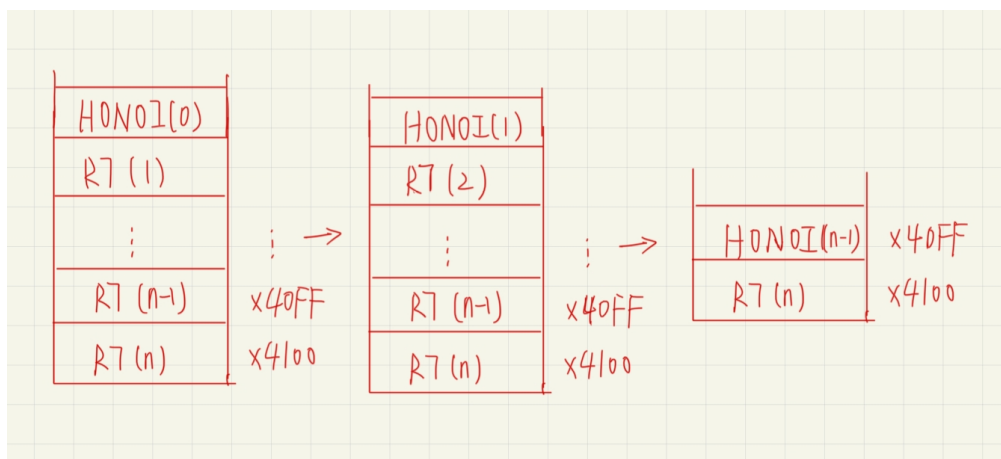
```
1 SITESTU .FILL StNum
2 LD R0, SITESTU
3 trap x22
```

轮询键盘输入

通过轮询键盘状态寄存器 *KBSR* 判断是否有键盘输入，若有则跳转至键盘中断服务程序。

求解汉诺塔问题

采用递归的方法实现求解汉诺塔问题，根据 LC-3 的 *JSR, RET* 操作，可知寄存器 *R7* 保存子程序结束后返回的地址，而如果采用递归的方式，必须使用一个栈来存储每个程序中的 *R7* 的值。该递归子程序的栈内容如图所示



当 $n > 0$ 时，进入下一级程序 *HONOI(n - 1)*，当下一级程序结束返回时，将结果从栈中取出，计算结果后再压入栈，返回上一级程序。
当 $n = 0$ 时，子程序得到结果 0，并将其压入栈中，返回上一级程序。

输出答案

若将结果直接输出到屏幕，则寄存器的值会被视为一个字符输出，故需要通过取余按位输出结果，并且要将每一位都的值都加 48，以保证输出的字符与整型值对应。

```

1  L5      ADD R6, R6, #1
2          LD R3, NUM2
3          ADD R0, R0, R3
4          BRzp L5
5          LD R3, NUM3
6          ADD R1, R0, R3
7          ADD R6, R6, #-1
8          LD R3, NUM4
9          ADD R0, R6, #0
10         BRz #2
11         ADD R0, R0, R3
12 OUT1    trap x21
13         ADD R0, R1, #0
14         AND R6, R6, #0
15 L6      ADD R6, R6, #1
16         ADD R0, R0, #-10
17         BRzp L6
18         ADD R1, R0, #10
19         ADD R6, R6, #-1
20         ADD R0, R6, #0
21         BRz #2
22         ADD R0, R0, R3
23 OUT2    trap x21
24         ADD R0, R1, #0
25         ADD R0, R0, R3
26 OUT3    trap x21

```

键盘中断服务程序

该程序判断输入值是否为有效值的过程中同样要考虑字符型和整型的对应关系，若转化为整型后为 0 – 9 的整数，则输出相应信息，并存储结果。

若不是有效值，输出相应的信息。已知只有当输入值为有效值时才进行汉诺塔问题的求解，故使用一个寄存器作为判断输入是否为有效值的标志位。

实验结果

程序运行的结果如图所示

```
PB20061376
q is not a decimal number
PB20061376
1 is a decimal number
HONOI move(s): 1
PB20061376
2 is a decimal number
HONOI move(s): 3
PB20061376
3 is a decimal number
HONOI move(s): 7
PB20061376
4 is a decimal number
HONOI move(s): 15
PB20061376
5 is a decimal number
HONOI move(s): 31
PB20061376
6 is a decimal number
HONOI move(s): 63
PB20061376
7 is a decimal number
HONOI move(s): 127
PB20061376
8 is a decimal number
HONOI move(s): 255
PB20061376
9 is a decimal number
HONOI move(s): 511
PB20061376
```