

# Lab5 of ICS

姓名：曾艺润

学号：PB21000009

## Lab5 of ICS

Purpose

Principles

Procedure

Result

## Purpose

1. 写一个递归计算HANOI塔结果的用户程序
2. 写 Interrupt Service Routine
3. 进一步掌握 LC - 3 tool 的使用
4. 理解 LC - 3 汇编语言的原理
5. 熟悉 LC - 3 汇编语言的使用
6. 了解 LC - 3 汇编语言的特性

## Principles

使用尽量少的代码，尽量清晰的码风，尽量低的时间复杂度

## Procedure

1. 学号循显：
  1. 使用死循环，每次循环时 DELAY 0x2500 次接着下一次显示，以防止显示太快
2. Interrupt Service Routine
  1. 每次输入 n 的时候，如果 n 是一个有效值(0 - 9)，那么输出 n is a decimal digit 存储 n 至 x3FFF 后调用递归程序计算 HANOI\_N 的结果，将结果存储在 x4000，展示之后继续循环显示学号。否则输出 n is not a decimal digit
  2. 要注意 ASCII 码的转换，调用递归程序之前要把输入的值从 ASCII 码转化为数字
3. 递归部分：
  1. 使用 STACK 实现
  2. STACK 初始化在 xFDFE，递归时每次递减直至 R0 的结果为 0。之后就开始回溯计算结果，每次  $R0 \leftarrow 2 \times R0 + 1$ ，最终得到结果。

#### 4. 将结果展示的部分：

1. 对递归所求得的结果首先每次减去 100 直至为负数，记录减去 100 的次数，为百位（减至负数的那次不计）。而后加上 100，求解十位同理。个位无需求解。输出时需要转为 ASCII 码

```
1  .ORIG x800
2      ; (1) Initialize interrupt vector table.
3      LD  R0, VEC
4      LD  R1, ISR
5      STR R1, R0, #0
6
7      ; (2) Set bit 14 of KBSRR.
8      LDI R0, KBSRR
9      LD  R1, MASK
10     NOT R1, R1
11     AND R0, R0, R1
12     NOT R1, R1
13     ADD R0, R0, R1
14     STI R0, KBSRR
15
16     ; (3) Set up system stack to enter user space.
17     LD  R0, PSR
18     ADD R6, R6, #-1
19     STR R0, R6, #0
20     LD  R0, PC
21     ADD R6, R6, #-1
22     STR R0, R6, #0
23     ; Enter user space.
24     RTI
25
26     VEC    .FILL x0180
27     ISR    .FILL x1000
28     KBSRR  .FILL xFE00
29     MASK   .FILL x4000
30     PSR    .FILL x8002
31     PC     .FILL x3000
32     .END
33
34     ; *** USER PROGRAM ***
35     .ORIG x3000
36     LOOP  LEA R0, STUID
37           TRAP x22
38     DELAY ST R1, SaveR1    ; store R1 first
39           LD  R1, COUNT
40     REP   ADD R1, R1, #-1
41           BRp REP
42           LD  R1, SaveR1    ; restore R1
43           BRnzp LOOP
44     SaveR1 .BLKW 1
45     COUNT .FILL x2500
46     STUID  .STRINGZ "PB21000009\n"
47           .END
48
49     .ORIG x3100
50
```

```

51  HANOI  ADD R6, R6, #-1 ; R6--
52        STR R1, R6, #0 ; mem[R6] <- R1
53        ADD R1, R0, #0 ; R1 <- R0
54        BRz EXIT
55        ADD R6, R6, #-1 ; R6--
56        STR R7, R6, #0 ; mem[R6] <- R7
57        ADD R6, R6, #-1 ; R6--
58        STR R0, R6, #0 ; mem[R6] <- R0
59        ADD R0, R0, #-1 ; R0--
60        JSR HANOI
61        LDR R1, R6, #0 ; R1 <- mem[R0]
62        ADD R6, R6, #1 ; R6++
63        ADD R0, R0, R0
64        ADD R0, R0, #1 ; R0 <- 2*R0+1
65        LDR R7, R6, #0 ; R7 <- mem[R6]
66        ADD R6, R6, #1 ; R6++
67  EXIT   LDR R1, R6, #0 ; R1 <- mem[R6]
68        ADD R6, R6, #1 ; R6++
69        RET
70        .END
71        ;
72        .ORIG x3200
73        LEA R0, SET1
74        TRAP x22
75        ;
76        LDI R1, RESULTP ; R1 is the result
77        LD R3, NUMB0 ; R3 <- x30
78        AND R5, R5, #0 ; R5 <- 0
79        LD R4, HUND
80        NOT R4, R4
81        ADD R4, R4, #1 ; R4 <- (-100)
82  GETH   ADD R1, R1, R4 ; R1 -= 100
83        BRn FH ; if(R1 < 0) goto FH(finish calculating hundred bit)
84        ADD R5, R5, #1 ; COUNT++
85        BRnzp GETH
86        ;
87  FH     ADD R5, R5, R3 ; To change to ASCII code
88        LD R4, HUND ; R4 <- 100
89  LOOPP  LDI R2, DSRR
90        BRzp LOOPP
91        STI R5, DRRR ; To present 'bai wei'
92        ADD R1, R1, R4 ; R1 += 100
93        LD R4, TEN
94        NOT R4, R4
95        ADD R4, R4, #1 ; R4 <- (-10)
96        AND R5, R5, #0 ; COUNT <- 0
97  GETT   ADD R1, R1, R4
98        BRn FT
99        ADD R5, R5, #1
100       BRnzp GETT
101
102  FT     ADD R5, R5, R3
103        LD R4, HUND
104  LOOPP2 LDI R2, DSRR
105        BRzp LOOPP2
106        STI R5, DRRR

```

```

107
108 GET1    LD R4, TEN
109         ADD R5, R1, R4
110         ADD R5, R5, R3
111 LOOPP3   LDI R2, DSRR
112         BRzp LOOPP3
113         STI R5, DDRR
114
115         LEA R0, SET2
116         TRAP x22
117         LD R1, SR1
118         RET
119 SR1      .BLKW 1
120 DSRR     .FILL xFE04
121 DDRR     .FILL xFE06
122 NUMB0    .FILL x30
123 HUND     .FILL #100
124 TEN      .FILL #10
125 SET1     .STRINGZ "Tower of hanoi needs "
126 SET2     .STRINGZ " moves.\n"
127 RESULTP .FILL x4000
128         .END
129
130         .ORIG x3FFF
131         ; *** Begin hanoi data here ***
132 HANOI_N .FILL xFFFF
133 H        .FILL xFFFF
134         ; *** End hanoi data here ***
135         .END
136
137
138 ; *** INTERRUPT SERVICE ***
139         .ORIG x1000
140         ADD R6, R6, #-1
141         STR R0, R6, #0
142         ADD R6, R6, #-1
143         STR R1, R6, #0
144         ADD R6, R6, #-1
145         STR R2, R6, #0
146         ADD R6, R6, #-1
147         STR R3, R6, #0
148
149 LOOP1    LDI R1, KBSR
150         BRzp LOOP1
151 ;
152         LDI R0, KBDR      ; Read from keyboard
153 LOOP2    LDI R1, DSR       ; State of output
154         BRzp LOOP2
155         STI R0, DDR
156         LD R2, N0
157         ADD R2, R2, R0
158         BRn NOTN
159         LD R2, N9
160         ADD R2, R2, R0
161         BRp NOTN
162

```

```

163 ;
164 LD R2, N0 ; R2 <- (-48)
165 ADD R0, R0, R2 ; Transfer R0 from ASCII to number n
166 STI R0, SAR0 ; Store n in x3FFF
167 LD R2, NUMSTR2
168 LEA R3, STR2
169 LOOP3 LDI R1, DSR
170 BRzp LOOP3
171 LDR R0, R3, #0
172 STI R0, DDR
173 ADD R3, R3, #1
174 ADD R2, R2, #-1
175 BRp LOOP3
176 ;
177 LDI R0, SAR0
178 ST R6, SAR6 ; Store R6
179 LD R6, STACK
180 LD R1, HANOII ; Begin hanoi program
181 JSRR R1
182 STI R0, SRESULT ; Store the result in x4000
183 LD R1, OUTPUT ; Begin output program
184 JSRR R1
185 LD R6, SAR6 ; restore R6
186 BRnzp DONE
187
188 NOTN LD R2, NUMSTR3
189 LEA R3, STR3
190 LOOP4 LDI R1, DSR
191 BRzp LOOP4
192 LDR R0, R3, #0
193 STI R0, DDR
194 ADD R3, R3, #1
195 ADD R2, R2, #-1
196 BRp LOOP4
197
198 DONE LDR R3, R6, #0
199 ADD R6, R6, #1
200 LDR R2, R6, #0
201 ADD R6, R6, #1
202 LDR R1, R6, #0
203 ADD R6, R6, #1
204 LDR R0, R6, #0
205 ADD R6, R6, #1
206 RTI
207
208 STACK .FILL xFDFF
209 HANOII .FILL x3100
210 OUTPUT .FILL x3200
211 KBSR .FILL xFE00
212 KBDR .FILL xFE02
213 DSR .FILL xFE04
214 DDR .FILL xFE06
215 N0 .FILL #-48
216 N9 .FILL #-57
217 SAR6 .BLKW 1
218 SAR0 .FILL x3FFF

```

```

219 SRESULT .FILL x4000
220 NUMSTR2 .FILL 21
221 STR2 .STRINGZ " is a decimal digit.\n"
222 NUMSTR3 .FILL 25
223 STR3 .STRINGZ " is not a decimal digit.\n"
224 .END

```

## Result

```

PB21000009
PB21000009
PB21000009
PB21000009
1 is a decimal digit.
Tower of hanoi needs 001 moves.
PB2100 ■

```

```

PB21000009
PB21000009
3 is a decimal digit.
Tower of hanoi needs 007 moves.
PB21000009
PB21000009

```

```

PB21000009
PB21000009
PB21000009
5 is a decimal digit.
Tower of hanoi needs 031 moves.

```

选取其中 3 张截图展示