

深圳大学实验报告

课程名称：计算机系统(1)

实验名称：中断试验

学院：数学与统计学院

指导老师：俞航

报告人：王曦 组号：无

学号：2021192010 实验地点：LC-3模拟器

实验时间：2022年06月15日

提交时间：2022年06月15日

1. 实验目的

使用实验证明:中断驱动的输入、输出可以中断一个正在运行的程序,执行中断服务程序,返回被中断的程序,从被中断位置下一个地址继续执行(好像什么也没发生似的).试验中我使用键盘作为输入设备,中断正在运行的程序.

2. 实验内容

2.1 用户程序

用户程序将会连续地输出纵横交替的ICS,通过交替,输出两个不同行,如下:

ICS ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS ICS

ICS ICS ICS ICS ICS

确保输出不至于太快,以至于肉眼不能察觉.因而,需要延时操作,可以考虑如下实现:

即用户程序包含一小段代码用于每行间的计数,间隔为从2500开始倒计时,计时结束时,再进行输出.

2.2 键盘中断服务程序

键盘中断服务程序将会简单地在屏幕上写上十次,用户随机输入的字符并以Enter(x0A)结束.

中断服务程序中要求不使用TRAP指令.

注意,不要忘记保存和恢复在中端服务程序中使用的寄存器.

2.3 模拟操作系统支持

不幸的是,我们还没在LC-3上安装windows或Linux,所以我们必须要求你在你的用户程序代码前先做到以下三个步骤:

1.正常情况下,操作系统将会先安装一些栈空间,所以当中断发生的时候 PC和PSR可以被放进栈中(正如你知道的,当程序执行RTI,PC和PSR都会被弹出栈,处理器返回到执行被中断的程序)由于没有操作系统,请先把R6初始化为x4000,表示一个空的栈.

2.正常情况下,操作系统会建立中断向量表,它包含对应中断服务程序的起始地址,你必须为键盘中断先建立一个中断向量表.中断向量表的开始地址是x0100,键盘中断的中断向量是x80.你必须在中断向量表提供一个入口供本实验使用.

3.操作系统应该设置KBSR的IE(Interrupt Enable)位.

3. 实验结果

3.1 用户程序

用户程序实现持续间隔输出两行不同的"ICS".为使得肉眼能看清每行的内容,加入一些计数的延迟.

```

1  .ORIG x3000
2  LD R6,STACK ; initialize the stack pointer
3  LD R1,ENTRY1 ; set up the keyboard interrupt vector table entry
4  LD R2,STARTS
5  STR R2,R1,#0
6  LD R3,IE ; enable keyboard interrupts
7  STI R3,KBSR
8
9  ; start of actual user program to print ICS checkerboard
10 AGAIN LEA R0,STR1
11 PUTS
12 JSR DELAY
13 LEA R0,STR2
14 PUTS
15 JSR DELAY
16 BR AGAIN
17 HALT
18
19 ENTRY1 .FILL X0180
20 STARTS .FILL X2000
21 STACK .FILL X3000
22 IE .FILL X4000
23 KBSR .FILL XFE00
24 STR1 .STRINGZ "ICS ICS ICS ICS ICS ICS \n"
25 STR2 .STRINGZ " ICS ICS ICS ICS ICS ICS \n"
26
27 DELAY ST R1, SaveR1
28 LD R1, COUNT
29 REP ADD R1, R1, #-1
30 BRp REP
31 LD R1, SaveR1
32 RET
33 COUNT .FILL #2500
34 SaveR1 .BLKW 1
35
36 .END

```

3.2 中断程序

中断程序将用户键入的回车前的字符打印10次.

注意中断处理程序中不能使用TRAP,则只能用读写DSR的方式来显示字符.

注意备份和恢复寄存器的内容.

```

1  .ORIG x2000
2  ADD R6, R6, #-1 ; push into the stack
3  STR R0, R6, #0
4  ADD R6, R6, #-1

```

```

5   STR    R1, R6, #0
6   ADD    R6, R6, #-1
7   STR    R2, R6, #0
8   ADD    R6, R6, #-1
9   STR    R3, R6, #0
10  ADD    R6, R6, #-1
11  STR    R4, R6, #0
12
13  LD     R4, STRING ; the begin address of string
14  LOOP1  LDI R1,KBSR ; check the keyboard
15        BRZP  LOOP1
16        LDI R0, KBDR
17        LD  R2, BREAK ; is enter
18        ADD R2, R2, R0
19        BRZ  LOOP ; yes->stop input
20        STR R0, R4, #0 ; get the string
21        ADD R4, R4, #1
22        BRNZP LOOP1
23  LOOP   AND  R0, R0, #0 ; string ends with enter
24        ADD  R0, R0, #10
25        STR  R0,R4,#0
26  NEXT1  AND  R3, R3, #0
27        ADD  R3, R3, #10 ;cnt for 10 times
28  LOOP3  LD   R4, STRING
29  LOOP5  LDR  R0, R4, #0 ; read
30        ADD  R4, R4, #1
31  LOOP4  LDI  R1, DSR ; check the screen
32        BRZP  LOOP4
33        LD  R2, BREAK ; is enter?
34        ADD  R2, R2, R0
35        BRZ  NEXT3 ; yes->loop
36        STI  R0, DDR ; print char
37        BRNZP LOOP5
38  NEXT3  ADD  R3, R3, #-1
39        BRP  LOOP3
40  NEXT2  LDR  R4, R6, #0 ; top
41
42        ADD  R6, R6, #1
43        LDR  R3, R6, #0
44        ADD  R6, R6, #1
45        LDR  R2, R6, #0
46        ADD  R6, R6, #1
47        LDR  R1, R6, #0
48        ADD  R6, R6, #1
49        LDR  R0, R6, #0
50        ADD  R6, R6, #1
51        RTI  ; return
52
53  KBSR   .FILL  xFE00
54  KBDR   .FILL  xFE02
55  DSR    .FILL  xFE04
56  DDR    .FILL  xFE06
57  BREAK  .FILL  xFFF6 ; -x000A
58  STRING .FILL  x4000 ;begin address the of string
59        .END

```

3.3 运行结果

持续输出两行不同的"ICS":

[illegible]

中断:

[illegible]

4. 实验结论

1. 中断处理程序中不能使用TRAP指令,则要显示字符时只能通过读写DSR的方式.
 2. 中断处理程序中需备份用到的寄存器并恢复.
 3. 为避免输出过快导致肉眼看不清,可加入计数延迟或无实际作用的代码以消耗时钟周期,达到sleep效果.
 4. 通常情况下,当遇到中断发生之前,操作系统已经开辟好栈空间,保存PC和PSR,当执行到RTI时,PC和PSR会被弹栈. 因为没有操作系统m需要初始化R6为X3000m指示一个空栈.
 5. 操作系统会建立一张中断向量表,用来包含中断处理程序对应的起始执行地址,并将KBSR的IE(Interrupt Enable)位置1.
-
-