深圳大学实验报告

课程名称 _	计算机系统(1)	
项目名称 _	LC-3 中断实验	
77 F R _	== 1 7/2/4	
学 院_	计算机与软件学院	
, ,, <u>,</u>	11 21 No 4 N(11 4 120	
专 业_	软件工程(腾班)	
指导教师	俞 航	
4 4 700 F	14.4 /4/ 6	
报告人	黄亮铭 学号 2022155028	
_		
实验时间	2023年06月08日	

教务处制

一、实验目的

- (1) 使用 LC-3 编写汇编程序:输出指定字符串的用户程序;打印 10 次键盘输入内容的键盘中断程序;
 - (2) 通过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;
 - (3) 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。

二、实验内容

实验中使用键盘作为输入设备,中断正在运行的程序。

实验任务包括实现以下三部分程序:用户程序;键盘中断服务程序;操作系统支持的代码。

A. 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:

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。

B. 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那些寄存器。

C. 模拟操作系统支持

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。

三、实验过程与步骤

- (1) 用户程序(采用文档中所给的模板):
 - I. 首先设置栈指针(如图1):

```
;初始化栈指针
LD R6,NUM
```

图 1

II. 然后设置中断矢量表(如图3);

```
;设置键盘中断表项
LD R2, INTERUPT
STI R2, KEY
```

图 3

III. 设置中断使能位(如图 4);

```
;设置键盘中断使能位
LD R1, NUM
STI R1, KBSR
```

图 3

IV. 循环打印两行不同的字符串,通过通用寄存器 R2 的数值的正负性来判断下一行被输出的字符串(如图 4)

```
;循环打印ICS
AND R1, R1, #0
AND R2, R2, #0
ADD R1, R1, #6
ADD R2, R2, #1
LOOP
ADD R2, R2, #0 ;根据R2的正负性判断输出哪一个字符串
BRn ST2
ST1
LEA RØ, STR1
BRnzp PRINT
ST2
LEA RØ, STR2
PRINT
PUTS
JSR DELAY ;延时输出
ADD R1, R1, #-1
BRP LOOP
LD RO, ENTER
OUT ;换行
ADD R2, R2, #0
BRn SET6
SET5
AND R1, R1, #0
ADD R1, R1, #5
BRnzp NOTR2
SET6
AND R1, R1, #0
ADD R1, R1, #6
NOTR2
NOT R2, R2
ADD R2, R2, #1
BRnzp LOOP
HALT
```

V. 为了更好地观察输出结果,使用文档提供的延时输出程序(如图 5);

```
;文档提供的延时输出
DELAY ST R1, SaveR1
LD R1, COUNT
REP ADD R1,R1,#-1
BRp REP
LD R1, SaveR1
RET
COUNT .FILL #10000;自行将数值改大,方便观察
SaveR1 .BLKW 1
```

图 5

- (2) 键盘中断服务程序:
 - I. 判断输入的是否为回车,如是,则中断结束,否则继续(如图 6);

```
LOOP
LDI R3, KBDR
BRz LOOP
LD R4, Enter
ADD R3, R3, R4
BRnp READ
RTI
```

图 6

II. 将从键盘上读取的字符在屏幕上输出(如图7);

READ

```
AND R4, R4, #0
ADD R4, R4, #10
LDI R0, KBDR

STROUT
LDI R5, DSR
BRZP STROUT
STI R0, DDR
ADD R4, R4, #-1
BRP STROUT
AND R4, R4, #0
STI R4, KBDR
```

图 7

- (3) 将上述程序命名保存并以 asm 编译;
- (4) 打开 Simulate, 并将上述程序 load 到 Simulate, 在 x301E 处设置断点(如图 8、9);

```
x2000
                                                                                      R3, KBDR
LOOP
R4, Enter
R3, R3, R4
            1010011000010000
                                             xA610 LOOP
  x2000
x2001
x2002
x2003
              x05FE
x2811
                                             x16C4
  x2004
              0000101000000001
                                             x0A01
                                                                          BRNP
                                                                                      READ
             RTI
AND
ADD
LDI
  x2005
                                             x8000
  x2006
x2007
x2008
                                                                                      R4, R4, #0
R4, R4, #10
R0, KBDR
R5, DSR
                                             xA008
             x2009
                                             xAA08 STROUT
  x2009
x200A
x200B
x200C
x200D
                                             xAA00
x07FE
xB007
x193F
                                                                                      STROUT
RO, DDR
R4, R4, #-1
                                             x03FB
                                                                          BRP
                                                                                      STROUT
 x200D 000000111111011

x200E 0101100100100000

x200F 101110000000001

x2010 000011111101111

x2011 11111100000010

x2012 111111100000010

x2013 1111111000000110
                                                                          AND
STI
BRNZP
                                                                                      R4, R4, #0
R4, KBDR
LOOP
                                             x5920
                                             xB801
x0FEF
xFE02 KBDR
                                                                          TRAP
                                                                                      x02
                                             xFE04 DSR
                                                                          TRAP
                                                                                      x04
                                                                          TRAP
             11111111111110110
```

图 8

```
→ x3000
        0010110000100110 x2C26
                                                     R6, NUM
                                                     R2, INTERUPT
R2, KEY
x3001
         0010010000100110 x2426
                                              LD
         1011010000100110
 x3002
                                              STI
 x3003
         0010001000100011
                            x2223
                                                     R1, NUM
         1011001000100101 xB225
 x3004
                                                     R1, KBSR
                                                     R1, R1, #0
R2, R2, #0
 x3005
         0101001001100000
                            x5260
                                              AND
         0101010010100000
 x3006
                            x54A0
                                                     R1, R1, #6
R2, R2, #1
R2, R2, #0
 x3007
         0001001001100110
                            x1266
                                              ADD
 x3008
         0001010010100001
                                              ADD
                            x14A1
 x3009
        0001010010100000
                            x14A0 LOOP
                                              ADD
        00001000000000010
 x300A
                            x0802
                                              BRN
                                                     ST2
                                                     RO. STRI
 x300B
        1110000000100000
                            xE020
                                              LEA
         0000111000000001
                                              BRNZP PRINT
 x300C
                            x0E01
                                                     RO, STR2
PUTS
        1110000000100110
1111000000100010
 x300D
                            xE026
                                              LEA
                            xF022 PRINT
                                              TRAP
= x300E
                            x480F
 x300F
         01001000000001111
                                              JSR
                                                     DELAY
        0001001001111111
                                              ADD
                                                     R1, R1, #-1
= x3010
                            x127F
 x3011
         0000001111110111
                            x03F7
                                              BRP
                                                     LOOP
                                                     RO, ENTER
x3012
        0010000000011000
                            x2018
                                              LD
                                                     OUT
R2, R2, #0
 x3013
         1111000000100001
                            xF021
                                              TRAP
x3014
        0001010010100000
                            x14A0
                                              ADD
                            x0803
                                                     SET6
R1, R1, #0
R1, R1, #5
 x3015
         0000100000000011
                                              BRN
= x3016
= x3017
        0101001001100000
                            x5260 SET5
                                              AND
         0001001001100101
= x3018
                                              BRNZP NOTR2
        0000111000000010
                            x0E02
 x3019
         0101001001100000
                            x5260
                                              AND
                                                     R1, R1, #0
                                                     R1, R1, #6
R2, R2
= x301A
        0001001001100110
                            x1266
                                              ADD
= x301B
= x301C
         1001010010111111
                            x94BF NOTR2
                                                     R2, R2, #1
        0001010010100001
                            x14A1
                                              ADD
 x301D
        0000111111101011
                                              BRNZP LOOP
                            x0FEB
                                              TRAP
*301E 1111000000100101
                            xF025
                                                     HALT
        0011001000000110 x3206 DELAY
                                                     R1, SaveR1
```

图 9

(5) 点击 基 运行程序,输出结果如图 10 所示;

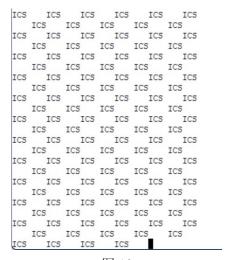


图 10

(6) 按下键盘上任意按键,观察程序输出结果,判断是否成功中断(图11);

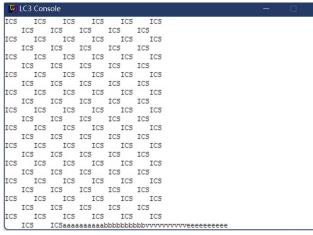


图 11

(7) 按下回车,观察程序是否继续打印之前的字符串(图12);

```
ICS ICS ICS ICS ICS
 ICS
CS
    ICS ICS ICS ICS
 ICS
ICS ICS ICS ICS ICS
  CS ICS ICS ICS ICS
 ICS
 ICS ICS ICS ICS ICS
CS
    ICS ICS ICS ICS
 ICS
ICS ICS ICS ICS ICS
ICS ICS ICS ICS ICS
ICS ICS ICS ICS ICS
ICS ICS ICS ICS bbbbbbbbbb
```

图 12

(8) 附上完整的代码;

用户程序:

- .ORIG x3000
- ; 初始化栈指针
- LD R6, NUM
- ;设置键盘中断表项
- LD R2, INTERUPT
- STI R2, KEY
- ;设置键盘中断使能位
- LD R1, NUM
- STI R1, KBSR

;循环打印 ICS

AND R1, R1, #0

AND R2, R2, #0

ADD R1, R1, #6

ADD R2, R2, #1

LOOP

ADD R2, R2, #0;根据 R2 的正负性判断输出哪一个字符串

BRn ST2

ST1

LEA RO, STR1

BRnzp PRINT

ST2

LEA RO, STR2

PRINT

PUTS

```
JSR DELAY;延时输出
ADD R1, R1, #-1
BRp LOOP
LD RO, ENTER
OUT ;换行
ADD R2, R2, #0
BRn SET6
SET5
AND R1, R1, #0
ADD R1, R1, #5
BRnzp NOTR2
SET6
AND R1, R1, #0
ADD R1, R1, #6
NOTR2
NOT R2, R2
ADD R2, R2, #1
BRnzp LOOP
HALT
;文档提供的延时输出
DELAY ST R1, SaveR1
       LD R1, COUNT
REP
       ADD R1, R1, #-1
       BRp REP
       LD R1, SaveR1
       RET
COUNT . FILL #10000;自行将数值改大,方便观察
SaveR1 .BLKW 1
NUM .FILL x4000
INTERUPT .FILL x2000
KEY .FILL x0180
KBSR .FILL xFE00
ENTER .FILL x000A
STR1 .STRINGZ "ICS "
STR2 .STRINGZ " ICS"
. END
```

键盘中断服务程序:

.ORIG x2000

LOOP

LDI R3, KBDR

BRz LOOP

LD R4, Enter

ADD R3, R3, R4

BRnp READ

RTI

READ

AND R4, R4, #0

ADD R4, R4, #10

LDI RO, KBDR

STROUT

LDI R5, DSR

BRzp STROUT

STI RO, DDR

ADD R4, R4, #-1

BRp STROUT

AND R4, R4, #0

STI R4, KBDR

BRnzp LOOP

KBDR .FILL xFE02

DSR .FILL xFE04

DDR .FILL xFE06

Enter .FILL xFFF6

. END

四、实验结论或体会

- (1)本次实验证明了中断驱动的输入、输出可以中断一个正在运行的程序;执行完中断服务程序后会返回被中断的程序,并从被中断位置下一个地址继续执行;
 - (2) 成功使用 LC-3 汇编语言编写用户程序和键盘中断服务程序;
- (3) 手动设置中断使能相当于将当前运行的程序的优先级降到最低,任何中断信号的发起,系统都会中断当前程序。
- (4) 不足: 写键盘中断服务程序时,错误地将回车的补码设置为 xFF06,导致中断服务程序不能结束。改进:将回车的补码设置为 xFFF6.

指导教师批阅意见:	
成绩评定:	
扌	旨导教师签字:
	年 月 日
备注:	
任 在:	