

深圳大学实验报告

课程名称 计算机系统(1)

项目名称 实验 5: LC-3 中断实验

学 院 计算机与软件学院

专 业 数计班

指导教师 李庚辉

报 告 人 詹耿羽 学号 2023193026

实验时间 2024. 6. 22

教务处制

一、实验目的与要求

1. 实验背景

在计算机科学中，中断是由硬件或软件向处理器发出的一个信号，表明一个需要立即注意的事件。一个中断提醒处理器有一个高优先级的条件，需要中断处理器正在执行的当前代码，即当前线程。处理器通过暂停其当前活动，保存其状态，并执行一个称为中断处理程序（或中断服务例程，ISR）的小程序来处理该事件。此中断是暂时的，在中断处理程序完成后，处理器将恢复执行上一个线程。

中断有两种类型：硬件中断和软件中断。

2. 实验目的

- 1) 学会分析和理解给定的编程问题；
- 2) 掌握中断驱动的基本原理；
- 3) 掌握键盘数据寄存器（KBDR）及键盘状态寄存器（KBSR）工作基本原理；
- 4) 掌握输出数据寄存器（DDR）及输出状态寄存器（DSR）工作基本原理；
- 5) 利用 LC-3 设计并编写用户程序及中断程序。

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:

```
1.  ICS  ICS  ICS  ICS  ICS  ICS
2.   ICS  ICS  ICS  ICS  ICS
3.  ICS  ICS  ICS  ICS  ICS  ICS
4.   ICS  ICS  ICS  ICS  ICS
5.  ICS  ICS  ICS  ICS  ICS  ICS
6.   ICS  ICS  ICS  ICS  ICS
7.  ICS  ICS  ICS  ICS  ICS  ICS
8.   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。

二、实验内容

1. 实验任务

1) 用户程序

用户程序将会包含连续地输出纵横交替的 ICS，通过交替输出两个不同行，如下：

```
1.  ICS  ICS  ICS  ICS  ICS  ICS
2.   ICS  ICS  ICS  ICS  ICS
```

确保输出不至于太快，以至于肉眼不能察觉。用户程序应该包含一小段代码用于每行间的计数，间隔为从 25000 开始倒计时输出在屏幕上。

2) 键盘中断程序

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。

需要注意的是，中断服务程序中不能使用 TRAP 指令。如在屏幕上显示一个字符，必须检测先 DSR 寄存器，然后写进 DDR 寄存器，并且也不可以调用 TRAP x21 (OUT)，或者其它 TRAP

程序。

3) 编写出汇编代码

为了使得代码具有可读性，我会给代码每句话都加上详细的注释。

三、实验步骤与过程

1. 实验思路

1) 用户程序：

经过分析我画出了如下的思维导图：

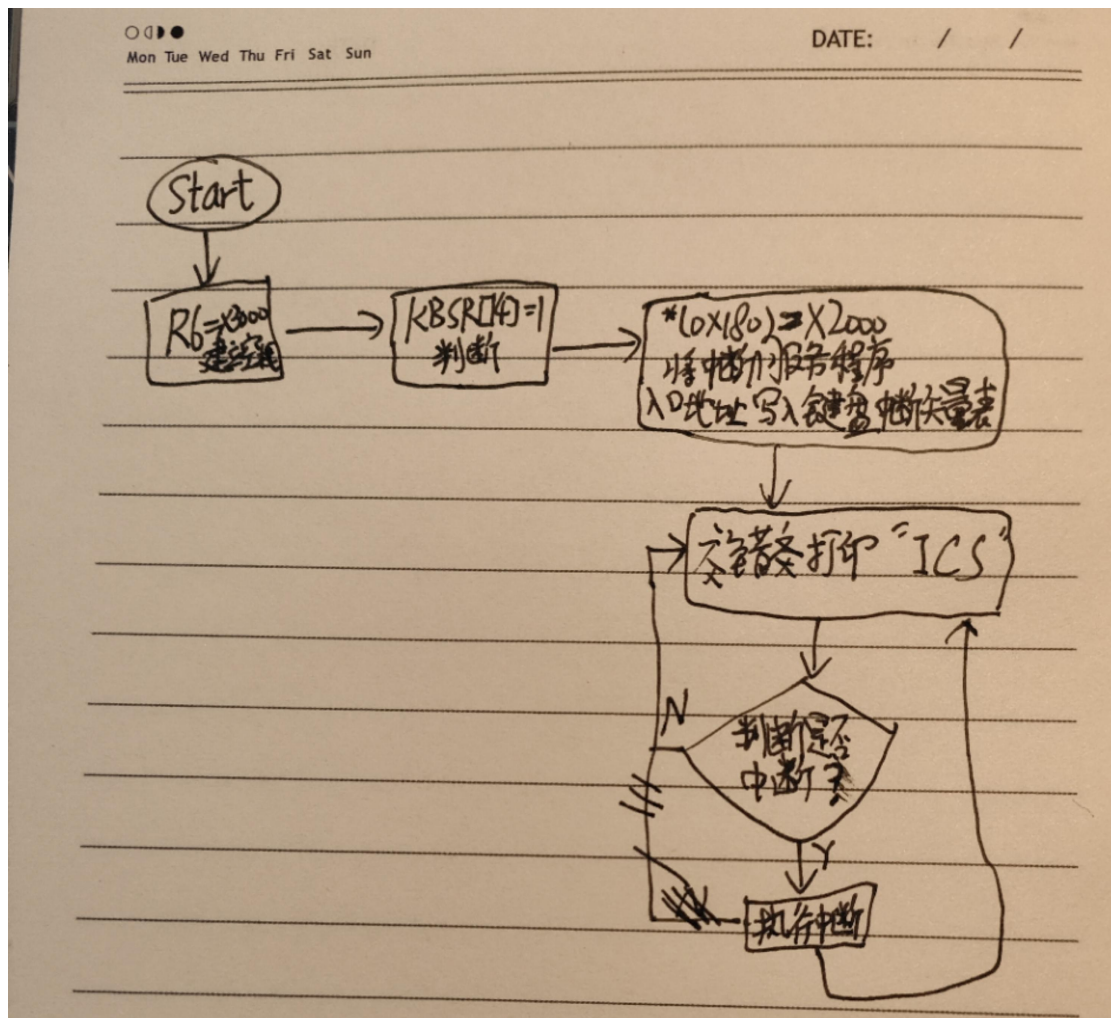


图 1：用户程序思维导图

2) 中断程序

经过分析，我画出了如下的思维导图。

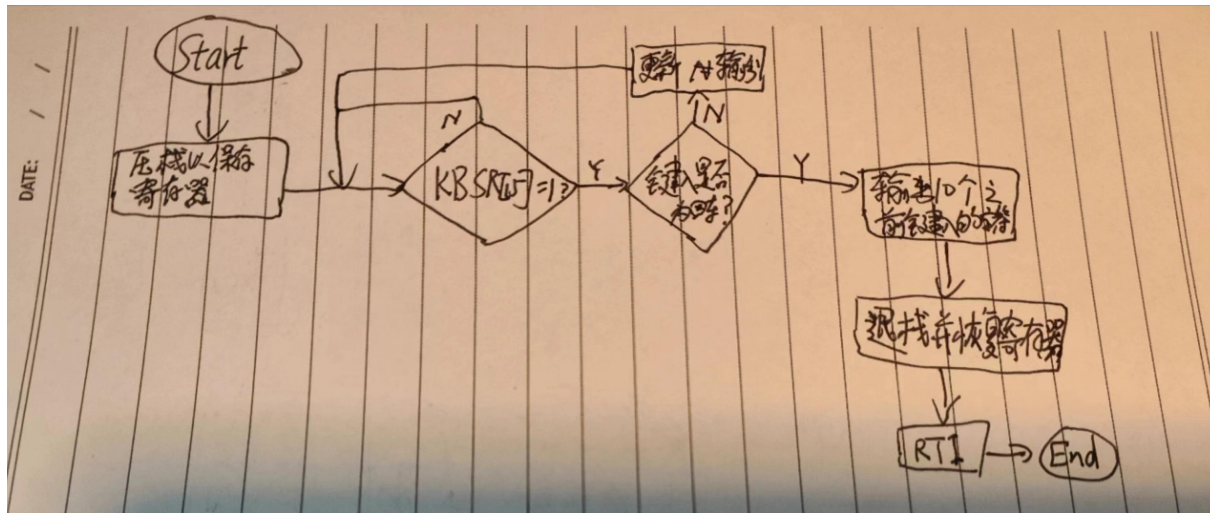


图 2: 中断程序思维导图

2. 根据思维导图写出代码

1) 思路解释

<1> 用户程序

要实现从键盘输入一个字符后执行中断程序,就需要将中断程序的起始地址写入键盘的中断矢量表中,且要将 KBSR 的中断使能位设置为 1。

具体步骤如下:

A.把 R6 初始化为 x3000,表示一个空栈,用于存放 PC 和 PSR 等需要保存和恢复的内容;

B.将中断服务程序的入口地址 x2000 和键盘的中断矢量 x0180 联系起来;

C.接着,将 KBSR 的中断使能位设置为 1。

接着,依次完成以下三个步骤:

D.在用户程序中,连续地输出纵横交替的 ICS;

E.在读入来自键盘的字符后,保护现场并进入中断服务子程序;

F 在子程序中,如果键入回车,则输出十个之前键入的字符,恢复现场,并回到用户程序,重复 C 步骤。

<2> 中断程序

该程序分为以下步骤:

A.压栈保存寄存器(保护现场);

B.反复检测 KBSR[15],如果 KBSR[15]=1,则检测键入的字符是否是回车。如果是回车,则执行 C 步骤;如果不是回车,则更新待输出的字符后回到 B 步骤;

C.连续输出十个之前在用户程序键入的字符并换行;

D.退栈以恢复寄存器(恢复现场)并执行 RTI;

2) 代码呈现

<1> 用户程序

```

1.  .ORIG    x3000
2.  LD R6,STACK ; 初始化堆栈指针,保存在寄存器 R6 中
3.  ; 设置键盘中断向量表项
4.  LD R1,ENTRANCE ; 加载中断服务程序的入口地址到 R1
5.  LD R2,INTV ; 加载中断向量地址到 R2
6.  STR R1,R2,#0 ; 如果键盘有输入,程序计数器(PC)将跳转到 x2000
7.  ; 启用键盘中断

```

```

8. LD R3,IE ; 加载中断使能标志到 R3
9. STI R3,KBSR ; 设置 KBSR 的第 14 位为 1, 以启用键盘中断
10. ; 开始实际的用户程序, 用于打印 ICS 棋盘图案
11. PRINT LEA R0,STR1 ; 将第一行字符串的地址加载到 R0
12. TRAP x22 ; 打印 R0 指向的第一行字符串
13. LEA R0,STR2 ; 将第二行字符串的地址加载到 R0
14. JSR DELAY ; 调用延迟子程序
15. TRAP x22 ; 打印 R0 指向的第二行字符串
16. JSR DELAY ; 调用延迟子程序
17. BRnzp PRINT ; 无条件跳转到 PRINT 标签, 继续打印
18. HALT ; 停止程序
19. DELAY ST R1, SaveR1 ; 保存 R1 的值到 SaveR1
20. LD R1, COUNT ; 加载计数器初始值到 R1
21. REP ADD R1,R1,#-1 ; R1 减 1
22. BRp REP ; 如果 R1 为正, 跳转回 REP 继续循环
23. LD R1, SaveR1 ; 恢复 R1 的原始值
24. RET ; 返回调用者
25. STR1 .STRINGZ "ICS ICS ICS ICS ICS ICS\n"
26. STR2 .STRINGZ " ICS ICS ICS ICS ICS ICS\n"
27. IE .FILL x4000 ; 中断使能标志位, 位 14 设置为 1
28. KBSR .FILL xFE00 ; 键盘状态寄存器的地址
29. KBDR .FILL xFE02 ; 键盘数据寄存器的地址
30. COUNT .FILL #25000 ; 延迟计数, 用于控制显示速度
31. INTV .FILL x0180 ; 中断向量表的具体位置
32. ENTRANCE .FILL x2000 ; 中断服务程序的起始地址
33. SaveR1 .FILL #0 ; 用于保存 R1 寄存器的值
34. STACK .FILL x3000 ; 堆栈的初始地址
35. .END

```

<2> 中断服务程序

```

1. .ORIG x2000 ; 设置程序起始地址为 x2000
2. ADD R6,R6,#-1 ; 将栈指针向下移动 1 位
3. STR R0,R6,#0 ; 将 R0 寄存器的内容保存到栈顶
4. ADD R6,R6,#-1 ; 将栈指针再次向下移动 1 位
5. STR R1,R6,#0 ; 将 R1 寄存器的内容保存到栈顶
6. ADD R6,R6,#-1 ; 再次将栈指针向下移动 1 位
7. STR R2,R6,#0 ; 将 R2 寄存器的内容保存到栈顶
8. ADD R6,R6,#-1 ; 再次将栈指针向下移动 1 位
9. STR R3,R6,#0 ; 将 R3 寄存器的内容保存到栈顶
10.
11. LOOP ST R0,SaveR0 ; 在循环开始前保存 R0 的原始值
12. CHECK LDI R1,KBSR ; 从内存地址 KBSR 加载键盘状态寄存器的值到 R1
13. ADD R1,R1,#0 ; 检查 KBSR 的第 15 位 (键盘就绪位)
14. BRzp CHECK ; 如果 KBSR[15]=0, 继续等待键盘输入

```

```

15.
16.  LDI R0,KBDR          ; 从内存地址 KBDR 加载键盘输入的数据到 R0
17.  LD R2,_ENDLINE       ; 加载_ENDLINE 中的值到 R2
18.  ADD R2,R2,R0         ; 将 R0 的值（键盘输入的字符）添加到_ENDLINE
19.  BRnp LOOP           ; 如果结果为负（即输入非换行符），返回 LOOP
    继续处理输入
20.
21.  AND R3,R3,#0         ; 清除 R3 的内容（设置 R3 为 0）
22.  ADD R3,R3,#10        ; 将 R3 设置为 10（用于后续打印循环）
23.
24.  P_LOOP LD R0,SaveR0   ; 恢复 R0 的原始值
25.  START LDI R1,DSR      ; 从内存地址 DSR 加载显示状态寄存器的值到 R1
26.  BRzp START           ; 如果 DSR[15]=0，表示显示设备未就绪，继续等待
27.  STI R0,DDR           ; 向内存地址 DDR 发送数据（即输出 R0 的值到显示设备）
28.  ADD R3,R3,#-1        ; R3 减 1
29.  BRp P_LOOP           ; 如果 R3 为正，继续 P_LOOP 循环
30.
31.  P_ENDL LD R0,ENDLINE  ; 加载 ENDLINE 中的换行符到 R0
32.  LDI R1,DSR           ; 再次加载显示状态寄存器的值到 R1
33.  BRzp P_ENDL          ; 如果显示设备未就绪，继续等待
34.  STI R0,DDR           ; 输出换行符到显示设备
35.
36.  LDR R3,R6,#0          ; 从栈中恢复 R3 的原始值
37.  ADD R6,R6,#1          ; 将栈指针向上移动 1 位
38.  LDR R2,R6,#0          ; 从栈中恢复 R2 的原始值
39.  ADD R6,R6,#1          ; 将栈指针向上移动 1 位
40.  LDR R1,R6,#0          ; 从栈中恢复 R1 的原始值
41.  ADD R6,R6,#1          ; 将栈指针向上移动 1 位
42.  LDR R0,R6,#0          ; 从栈中恢复 R0 的原始值
43.  ADD R6,R6,#1          ; 将栈指针向上移动 1 位
44.
45.  RTI                  ; 返回从中断
46.
47.  ; buffer space as required
48.  _ENDLINE .FILL xFFF6  ; 设置_ENDLINE 为'\n'的 ASCII 值的补码
49.  ENDLINE .FILL x000A    ; 设置 ENDLINE 为'\n'的 ASCII 值
50.  SaveR0 .FILL #0       ; 为保存 R0 设置缓冲
51.  KBSR .FILL xFE00       ; 键盘状态寄存器的内存地址
52.  KBDR .FILL xFE02       ; 键盘数据寄存器的内存地址
53.  DSR .FILL xFE04        ; 显示状态寄存器的内存地址
54.  DDR .FILL xFE06        ; 显示数据寄存器的内存地址
55.  .END

```


3) 结果呈现

<1> 用户程序不断输出“ICS”：

[illegible]

<2> 中断（键入“abc”连续输出 10 次）：

```
LC3 Console
```

```
ICS ICS ICS ICS ICS ICS
  ICS ICS ICS ICS ICS ICS
ICS ICS ICS ICS ICS ICS
abcabcabcabcabcabcabcabcabc 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 ICS ICS ICS ICS
  ICS ICS ICS ICS ICS ICS
ICS ICS ICS ICS ICS ICS
  ICS ICS ICS ICS ICS ICS
```

中断后继续执行程序, 可见符合实验要求。

四、实验结论或体会

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。

指导教师批阅意见:

成绩评定:

指导教师签字:

年 月 日

备注:

注: 1、报告内的项目或内容设置,可根据实际情况加以调整和补充。