**深 圳 大 学 实 验 报 告**

|  |
| --- |
| **课程名称 计算机系统1**  **项目名称 实验六 LC-3中断实验**  **学 院 计算机与软件学院**  **专 业 软件工程（腾班）**  **指导教师 陈飞**  **报 告 人 叶茂林 学号 2021155015**  **实验时间 2022.6.11**  **提交时间 2022.6.18** |

**教务处制**

# 一、实验目的与要求

（1）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)。（2）试验中使用键盘作为输入设备，中断正在运行的程序。

# 二、实验内容与方法

试验任务包括实现以下三部分程序：

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**。**

**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那些寄存器。

**C. 操作系统支持的代码**

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(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。

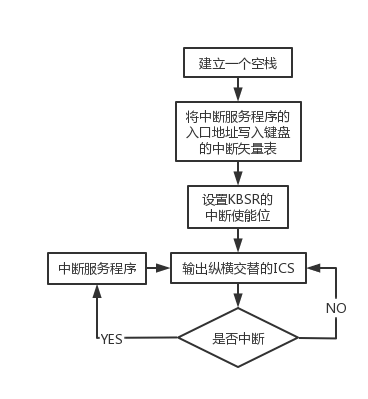
2.正常情况下，操作系统会建立中断向量表，它包含对应中断服务程序的起始地址，必须为键盘中断先建立一个中断向量表。中断向量表的开始地址是x0100，键盘中断的中断向量是x80。必须在中断向量表提供一个入口供本实验使用。

3.操作系统应该设置KBSR的IE(Interrupt Enable)位。

# 三、实验步骤与过程

（依照实验内容，逐条撰写实验过程与实验所得结果：包括程序总体设计，核心数据结构及算法流程，调试过程。请附上核心代码，及注意格式排版的美观。实验提交时，以上为评分依据，请不删除本行）

**程序总体设计**

****

**核心数据结构**

**1、用户程序**

R6初始化为x4000，表示一个空栈。

地址为x1080的内存空间存入值x2000，即将中断服务程序的入口地址写入键盘的中断矢量表。

地址为xfe00的内存空间存入值x4000，即设置KBSR的中断使能位。

R0存储字符串连续存储空间的首地址。

1. **中断服务程序**

R6是栈指针x4000。

R0存储KBSR的值，用于判断能否读取KBDR的内容。

然后R0继续存储KBDR的值，并于存储回车符的负ASCLL码的R1相加，其结果存放于R1，判断是否为回车符。

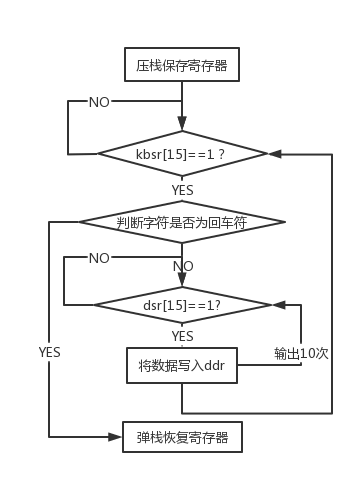
然后R1存储输出的次数10。

R2存储DSR的值，判断能否能否向DDR写入数据。

然后DDR存储输入的字符。

**算法流程**

中断服务程序

****

**调试过程**

相继将两个程序装载进LC-3模拟器，然后点击运行，发现输出正常，如图1所示：

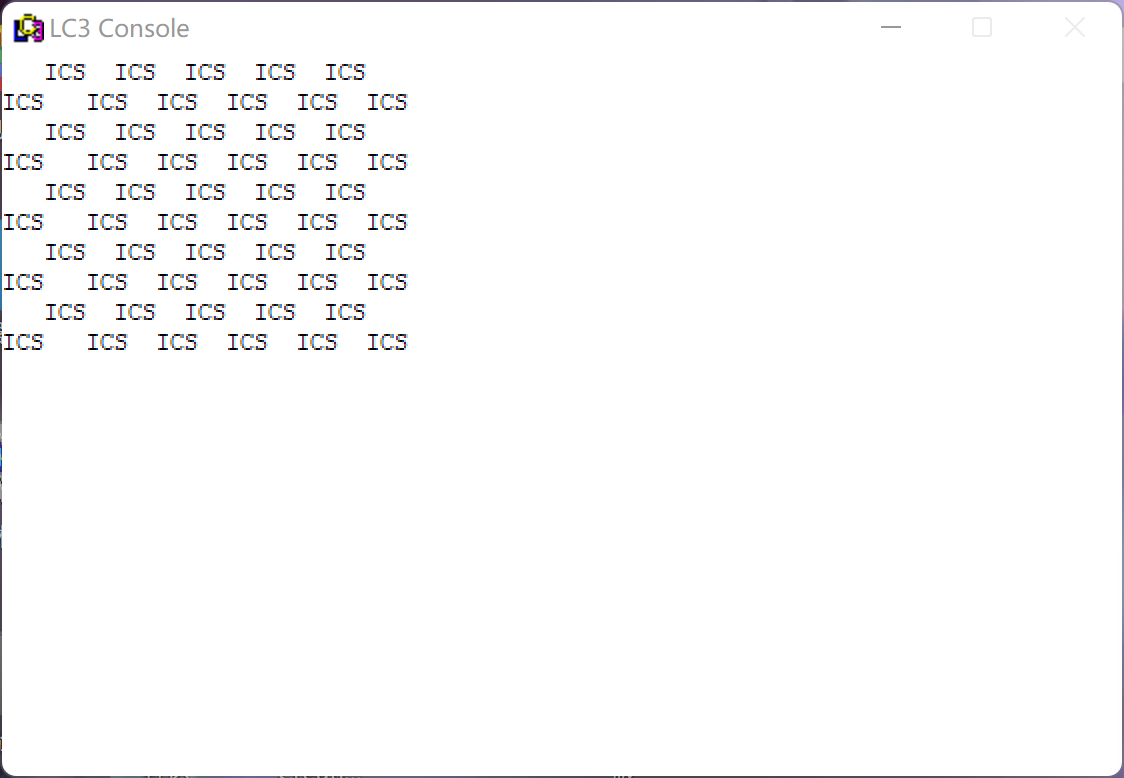


图1

然后随机键入一个字符，成功将输入的字符输出10次，但是发现不能再输入其他字符，如图2所示：

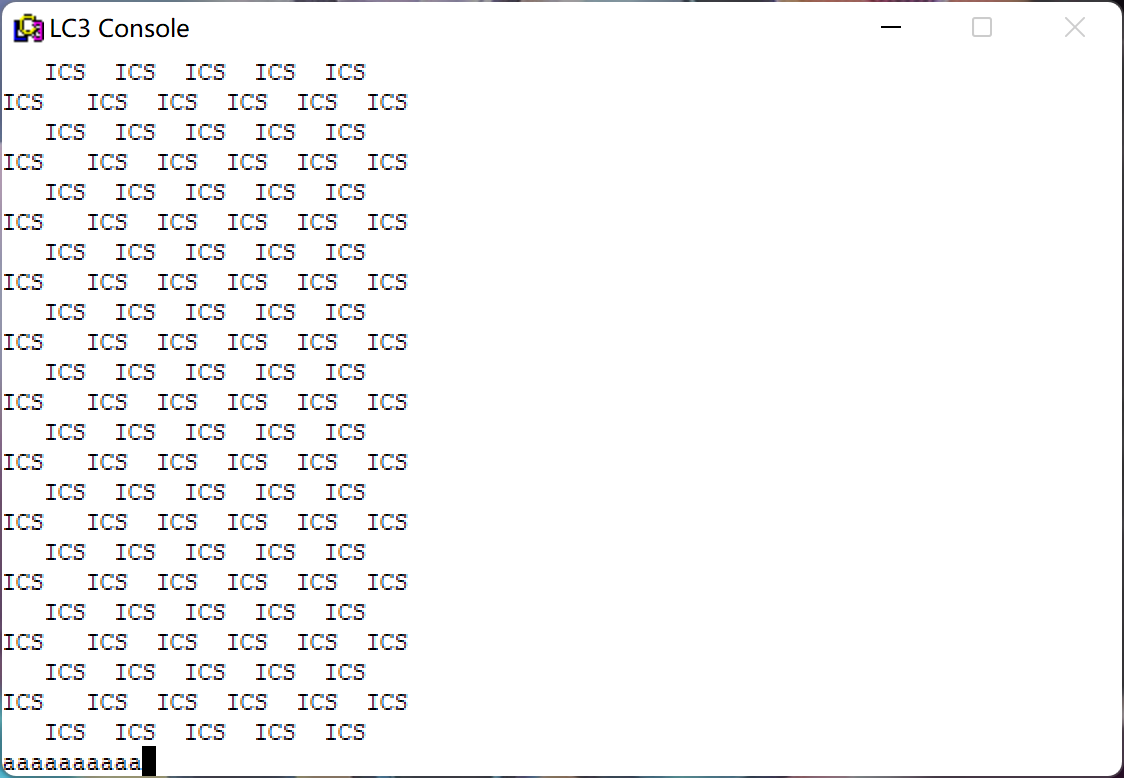


图2

然后查看代码，发现只写了读取一遍输出一遍的代码，于是修改代码，再次运行发现电脑出现卡顿，任务管理器无法终止程序运行，长按电源键强制关机，重新查看代码，发现LDI指令写成LD，如图3所示：

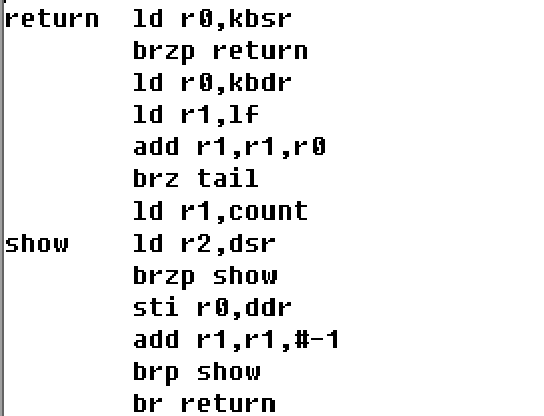


图3

然后修改代码，将LD指令换成LDI指令，运行程序，发现可以正常输出，如图4所示：

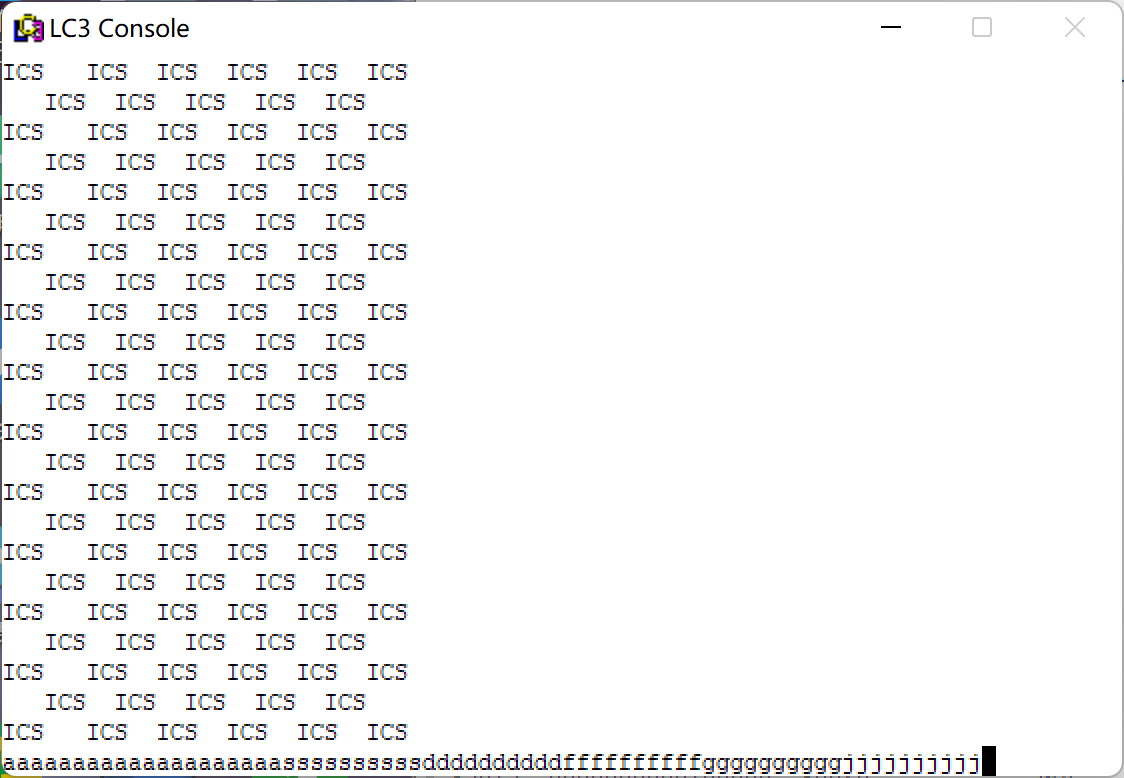


图4

然后敲入回车，发现程序终止，如图5所示：

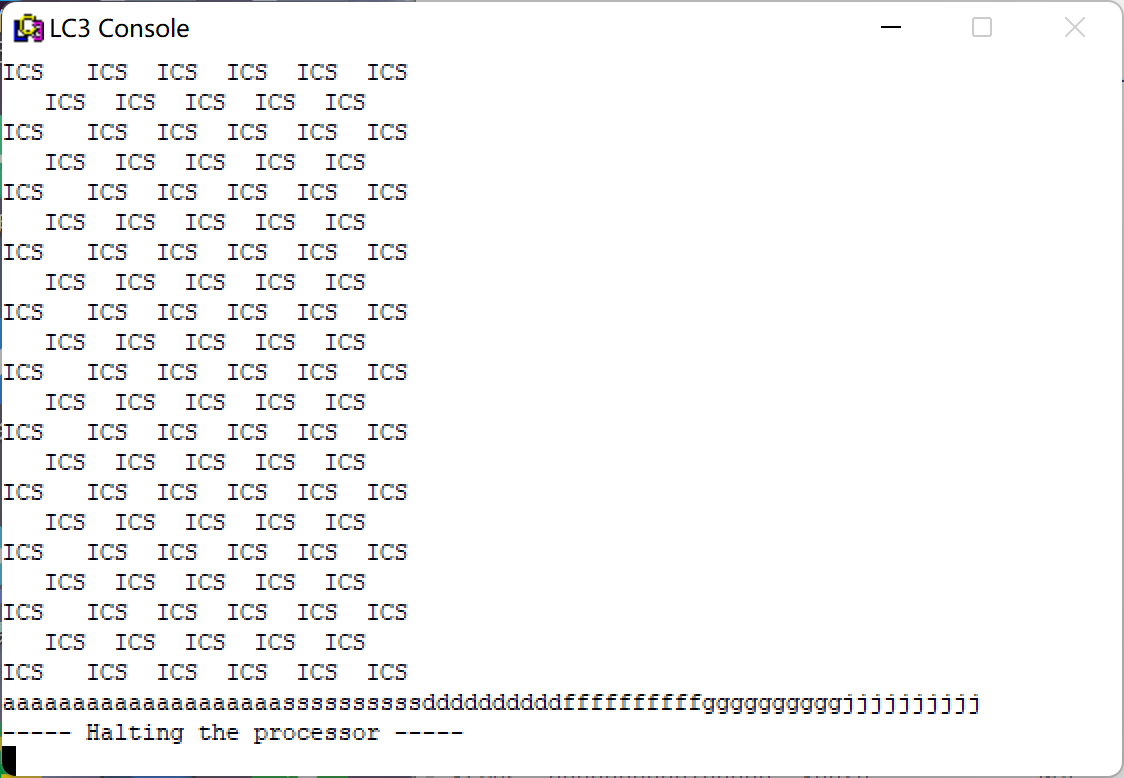


图5

查看代码，发现之前debug写的halt指令忘记删除，如图6所示：

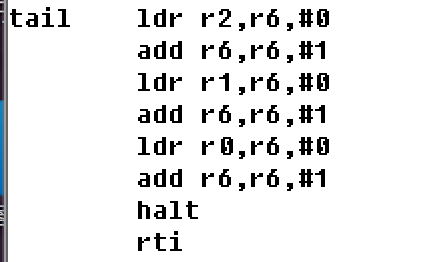


图6

然后去掉halt指令，重新运行程序，运行结果符合预期结果，键入回车后，程序从中断程序正常跳回，如图7所示：

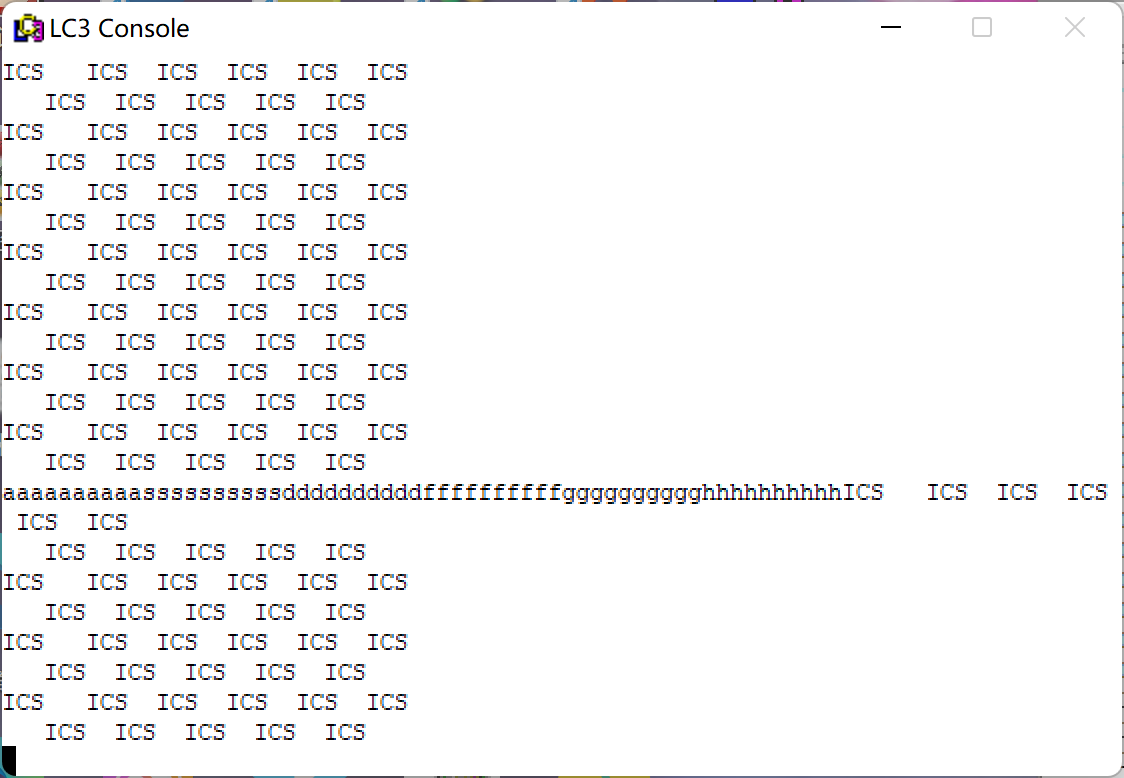


图7

# 四、实验结论或体会

（撰写实验收获及思考）

通过本次的LC-3中断实验，我学会了分析问题，基本掌握了中断驱动的基本原理，对KBSR和KBDR以及DSDR和DDR的基本原理有了更深的认识和体会，成功地设计并实现了中断程序，达到了实验的预期效果。

在本次实验中，我因为误将LDI指令写成LD指令而导致电脑出现了异常，使得操作系统无法终止异常程序的运行，这使我认识到，对于计算机而言，一切都是严谨的、确定的，我们应该尊重和理解计算机，从计算机角度出发去解决实际问题，要减少因人员操作失误而导致的不可估计的后果的发生。

此后应加强编程语言的学习，提高自身编程能力，为以后做工程项目做准备。

|  |
| --- |
| 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。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。

Code\_user\_program

.orig x3000

ld r6,stack

ld r0,inter

sti r0,vector

ld r0,enable

sti r0,kbsr

again lea r0,string1

puts

jsr DELAY

lea r0,string2

puts

jsr DELAY

br again

string1 .stringz "ICS ICS ICS ICS ICS ICS\n"

string2 .stringz " ICS ICS ICS ICS ICS \n"

DELAY ST R1, SaveR1

LD R1, COUNT

REP ADD R1,R1,#-1

BRp REP

LD R1, SaveR1

RET

COUNT .FILL x7fff

SaveR1 .BLKW 1

stack .fill x4000

kbsr .fill xfe00

vector .fill x0180

enable .fill x4000

inter .fill x2000

.end

Code\_interrupt\_service\_program

.orig x2000

add r6,r6,#-1

str r0,r6,#0

add r6,r6,#-1

str r1,r6,#0

add r6,r6,#-1

str r2,r2,#0

return ldi r0,kbsr

brzp return

ldi r0,kbdr

ld r1,lf

add r1,r1,r0

brz tail

ld r1,count

show ldi r2,dsr

brzp show

sti r0,ddr

add r1,r1,#-1

brp show

br return

tail ldr r2,r6,#0

add r6,r6,#1

ldr r1,r6,#0

add r6,r6,#1

ldr r0,r6,#0

add r6,r6,#1

rti

kbsr .fill xfe00

kbdr .fill xfe02

dsr .fill xfe04

ddr .fill xfe06

count .fill #10

lf .fill x-0a

.end