实验报告

1. 实验内容

详见lab3assignment

1. 算法分析

User\_program ：

enable keyboard interrupts：MASK OR KBSR 将KBSR的bit[14]改为1

MASK AND KBSR 将KBSR的bit[14]改为0

user program to print the checkerboard：

将当前程序的某种状态保存在FLAG，传递给Interrupt\_service\_routine

FLAG表示checkerboard的类型 1—\* ，--1—#

输出checkerboard的类型由Print Judger判断，FLAG会在Interrupt\_service\_routine中被修改

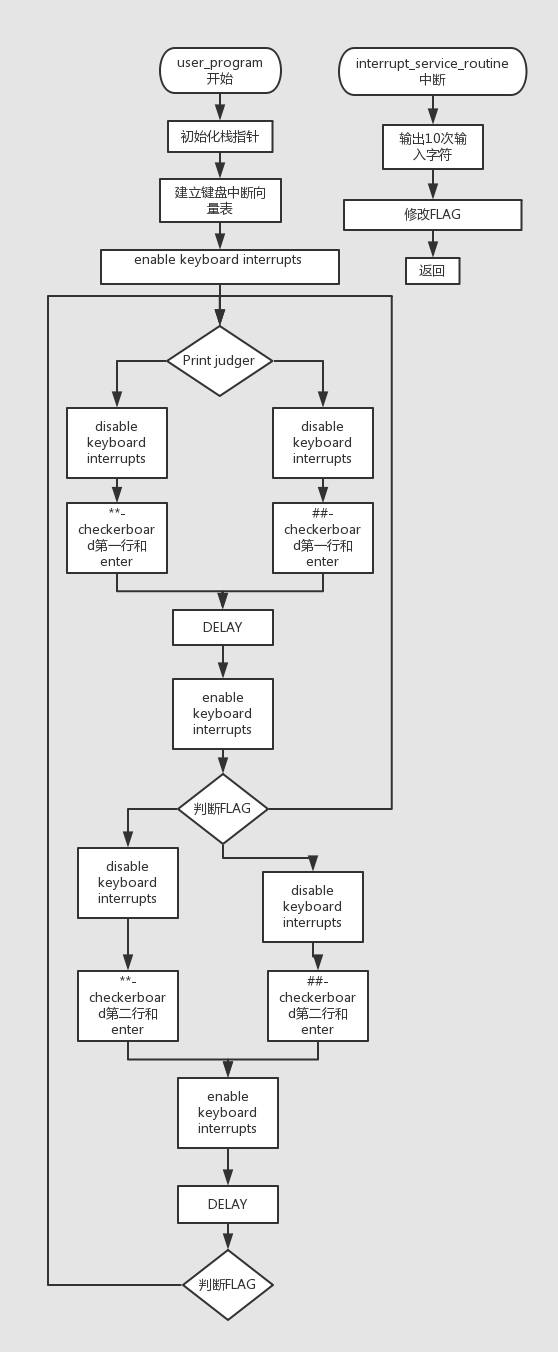
TRAP屏蔽：在PUTS和OUT之前disable keyboard interrupt，输出字符串之后enable keyboard interrupt

Interrupt\_service\_routine：

修改FLAG，若FLAG=1，则减二，若FLAG=-1，则加二

将字母输出十次

1. 流程图



1. 代码和注释

User\_program

.ORIG x3000

; initialize the stack pointer

LD R6,STACK

; set up the keyboard interrupt vector table entry

ADD R4,R4,0

ADD R5,R5,0

LD R4,KV

LD R5,SA

STR R5,R4,0

; enable keyboard interrupts

JSR ENABLE

; start of actual user program to print the checkerboard

LEA R3,FLAG

PJ LD R5,FLAG ;Print Judger

BRp LP1

BRn LP2

LP1 LEA R0,CB1

JSR DISABLE

PUTS

LD R0,ENTER

OUT

JSR ENABLE

JSR DELAY

LD R5,FLAG

BRn PJ

LEA R0,CB2

JSR DISABLE

PUTS

LD R0,ENTER

OUT

JSR ENABLE

JSR DELAY

BRnzp PJ

LP2 LEA R0,CB3

JSR DISABLE

PUTS

LD R0,ENTER

OUT

JSR ENABLE

JSR DELAY

LD R5,FLAG

BRp PJ

LEA R0,CB4

JSR DISABLE

PUTS

LD R0,ENTER

OUT

JSR ENABLE

JSR DELAY

BRnzp PJ

DELAY ST R1,SaveR1

LD R1,COUNT

REP ADD R1,R1,-1

BRp REP

LD R1,SaveR1

RET

ENABLE ADD R4,R4,0

ADD R5,R5,0

LDI R4,KBSR

LD R5,MASK

NOT R4,R4

NOT R5,R5

AND R4,R4,R5

NOT R4,R4

STI R4,KBSR

RET

DISABLE ADD R4,R4,0

ADD R5,R5,0

LDI R4,KBSR

LD R5,MASK2

AND R4,R4,R5

STI R4,KBSR

RET

FLAG .FILL x0001

COUNT .FILL #2500

SaveR0 .BLKW 1

SaveR1 .BLKW 1

ENTER .FILL x000A

CB1 .STRINGZ "\*\* \*\* \*\* \*\* \*\* \*\* \*\* \*\*"

CB2 .STRINGZ " \*\* \*\* \*\* \*\* \*\* \*\* \*\* "

CB3 .STRINGZ "## ## ## ## ## ## ## ##"

CB4 .STRINGZ " ## ## ## ## ## ## ## "

STACK .FILL x3000

KV .FILL x0180 ;Keyboard Vector

SA .FILL x2000 ;Starting Address

KBSR .FILL xFE00

MASK .FILL x4000

MASK2 .FILL xBFFF

.END

Interrupt\_service\_routine

.ORIG x2000

ST R0,SAVER0

ST R1,SAVER1

ST R2,SAVER2

ST R4,SAVER4

ST R3,FLAG1

;ST R2,PJ1

LD R2,COUNTER

START LDI R1,KBSR

BRzp START

LDI R0,KBDR

ECHO LDI R1,DSR

BRzp ECHO

STI R0,DDR

ADD R2,R2,1

BRn ECHO

LD R0,ENTER

ECHO1 LDI R1,DSR

BRzp ECHO1

STI R0,DDR

LDI R4,FLAG1

BRp L1

BRn L2

L1 ADD R4,R4,-2

STI R4,FLAG1

;LD R5,DIS

;ADD R0,R0,R5

BRnzp EXIT

L2 ADD R4,R4,2

STI R4,FLAG1

;LD R5,NEGDIS

;ADD R0,R0,R5

BRnzp EXIT

EXIT LD R0,SAVER0

LD R1,SAVER1

LD R2,SAVER2

LD R4,SAVER4

RTI

FLAG1 .FILL x0000

;PJ1 .FILL x0000

SAVER0 .BLKW 1

SAVER1 .BLKW 1

SAVER2 .BLKW 1

SAVER4 .BLKW 1

KBSR .FILL xFE00

KBDR .FILL xFE02

DSR .FILL xFE04

DDR .FILL xFE06

COUNTER .FILL xFFF6

ENTER .FILL x000A

.END

1. 运行结果

