# 实验1

;-----------------------------------------------------------

;实验一交通灯示例程序 |

; |

;功能：模拟交通灯的转换 |

;74LS244带有8位拨位开关，地址为80H or 82H or 84H or 86H |

;74LS273带有8个发光二极管，地址为88H or 8AH or 8CH or 8EH |

; |

;-----------------------------------------------------------

DOSSEG

.MODEL TINY ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 符号定义 |

;-----------------------------------------------------------

;

PortIn EQU 80h ;定义输入端口号

PortOut EQU 88h ;定义输出端口号

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

;以下开始放置用户指令代码

;

;--------------------------------------------

; |

; State machine style control system |

; |

;--------------------------------------------

;

; State11 全是红灯

;

MOV AL,36H;NS红EW红

MOV DX,PortOut;

OUT DX,AL;

CALL Delay1;

;

; State12 南北绿灯

;

Main:

MOV AL,33H;NS绿EW红

MOV DX,PortOut;

OUT DX,AL;

CALL Delay1;

;

; State13 南北绿闪

;

MOV CX,6;

LOOP1:MOV AL,33H;NS绿EW红

MOV DX,PortOut;

OUT DX,AL;

CALL Delay2;

MOV AL,37H;NS不亮EW红

MOV DX,PortOut;

OUT DX,AL;

CALL Delay2;

LOOP2:DEC CX;

JNZ LOOP1;

;

; State14 南北黄灯

;

MOV AL,35H;NS黄EW红

MOV DX,PortOut;

OUT DX,AL;

CALL Delay3;

;

; State15 东西绿灯

;

MOV AL,1EH;NS红EW绿

MOV DX,PortOut;

OUT DX,AL;

CALL Delay1;

;

; State16 东西绿闪

;

MOV CX,6;

LOOP3: MOV AL,1EH;NS红EW绿

MOV DX,PortOut;

OUT DX,AL;

CALL Delay2;

MOV AL,3EH;NS红EW不亮

MOV DX,PortOut;

OUT DX,AL;

CALL Delay2;

LOOP4: DEC CX;

JNZ LOOP3;

;

; State17 东西黄灯

;

MOV AL,2EH;NS红EW黄

MOV DX,PortOut;

OUT DX,AL;

CALL Delay3;

;

; State18

;

JMP MAIN;

;

;--------------------------------------------

; |

; Delay system running for a while |

; CX : contains time para. |

; |

;--------------------------------------------

DELAY1 PROC

PUSH CX

MOV CX,DelayLong ;

D0: LOOP D0

POP CX

RET

DELAY1 ENDP

;--------------------------------------------

; |

; Delay system running for a while |

; |

;--------------------------------------------

DELAY2 PROC

PUSH CX

MOV CX,DelayShort

D1: LOOP D1

POP CX

RET

DELAY2 ENDP

;--------------------------------------------

; |

; Delay system running for yellow |

; |

;--------------------------------------------

DELAY3 PROC

PUSH CX

MOV CX,DelayYellow

D1: LOOP D1

POP CX

RET

DELAY3 ENDP

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段

DelayShort dw 4000 ; 短延时参量

DelayLong dw 40000 ; 长延时参量

DelayYellow dw 20000 ; 长延时参量

END Main ;指示汇编程序结束编译

# 2-3

;-----------------------------------------------------------

;

; Build this with the "Source" menu using

; "Build All" option

;

;-----------------------------------------------------------

;

; 实验二示例程序

;-----------------------------------------------------------

; |

; |

; 功能：控制7段数码管的显示 |

; 编写：《嵌入式系统原理与实验》课程组 |

;-----------------------------------------------------------

DOSSEG

.MODEL SMALL ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 符号定义 |

;-----------------------------------------------------------

;

; 8255芯片端口地址 （Port number）分配:

PortA EQU 90H ; Port A's port number in I/O space

PortB EQU 92H ; Port B's port number in I/O space

PortC EQU 94H ; Port C's port number in I/O space

CtrlPT EQU 96H ; 8255 Control Register's port number in I/O space

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段;

DelayShort dw 4000 ; 短延时参量

DelayLong dw 40000 ; 长延时参量

; 显示数字

DISCHAR DB 01,02,03,04

; SEGTAB是显示字符0-F，其中有部分数据的段码有错误，请自行修正

SEGTAB DB 3FH ; 0011 1111 7-Segment Tube, 共阴极类型的7段数码管示意图

DB 06H ; 0000 0110

DB 5BH ; 0101 1011 a a a

DB 4FH ; 0100 1111 f b

DB 66H ; 0110 0110 f b

DB 6DH ; 0110 1101 f b

DB 7DH ; 0111 1101 g g g

DB 07H ; 0000 0111 e c

DB 7FH ; 0111 1111 e c

DB 6FH ; 0110 1111 e c

DB 77H ; 0111 0111 d d d h h h

DB 7CH ; 0111 1100 ----------------------------------

DB 39H ; 0011 1001 b7 b6 b5 b4 b3 b2 b1 b0

DB 5EH ; 0101 1110 DP g f e d c b a

DB 79H ; 0111 1001

DB 71H ; 0111 0001

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

;

; Init 8255 in Mode 0

; PortA Output, PortB Output

;

MOV AL,10001001B

OUT CtrlPT,AL ;

;

; 把数字1、2、3、4显示在数码管上

;

s: IN AL,PortC

NOT AL

MOV BL,AL

AND AL,0F0H;高四位

ADD AL,0EH;第一根管子

OUT PortA,AL

AND BL,0FH;低四位

MOV AL,SEGTAB[BX]

OUT PortB,AL

JMP s

RET

;--------------------------------------------

; |

; Delay system running for a while |

; CX : contains time para. |

; |

;--------------------------------------------

DELAY1 PROC

PUSH CX

MOV CX,DelayLong ;

D0: LOOP D0

POP CX

RET

DELAY1 ENDP

;--------------------------------------------

; |

; Delay system running for a while |

; |

;--------------------------------------------

DELAY PROC

PUSH CX

MOV CX,DelayShort

D1: LOOP D1

POP CX

RET

DELAY ENDP

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

END

# 2-4

;-----------------------------------------------------------

;

; Build this with the "Source" menu using

; "Build All" option

;

;-----------------------------------------------------------

;

; 实验二示例程序

;-----------------------------------------------------------

; |

; |

; 功能：控制7段数码管的显示 |

; 编写：《嵌入式系统原理与实验》课程组 |

;-----------------------------------------------------------

DOSSEG

.MODEL SMALL ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 符号定义 |

;-----------------------------------------------------------

;

; 8255芯片端口地址 （Port number）分配:

PortA EQU 90H ; Port A's port number in I/O space

PortB EQU 92H ; Port B's port number in I/O space

PortC EQU 94H ; Port C's port number in I/O space

CtrlPT EQU 96H ; 8255 Control Register's port number in I/O space

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段;

DelayShort dw 40 ; 短延时参量

DelayLong dw 40000 ; 长延时参量

; 显示数字

DISCHAR DB 01,02,03,04

; SEGTAB是显示字符0-F，其中有部分数据的段码有错误，请自行修正

SEGTAB DB 3FH ; 0011 1111 7-Segment Tube, 共阴极类型的7段数码管示意图

DB 06H ; 0000 0110

DB 5BH ; 0101 1011 a a a

DB 4FH ; 0100 1111 f b

DB 66H ; 0110 0110 f b

DB 6DH ; 0110 1101 f b

DB 7DH ; 0111 1101 g g g

DB 07H ; 0000 0111 e c

DB 7FH ; 0111 1111 e c

DB 6FH ; 0110 1111 e c

DB 77H ; 0111 0111 d d d h h h

DB 7CH ; 0111 1100 ----------------------------------

DB 39H ; 0011 1001 b7 b6 b5 b4 b3 b2 b1 b0

DB 5EH ; 0101 1110 DP g f e d c b a

DB 79H ; 0111 1001

DB 71H ; 0111 0001

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

;

; Init 8255 in Mode 0

; PortA Output, PortB Output

;

MOV AL,10001001B

OUT CtrlPT,AL ;

;

; 把数字1、2、3、4显示在数码管上

;

s:

IN AL,PortC

NOT AL

MOV BL,AL

AND AL,0F0H;高四位

AND BL,0FH;低四位

PUSH AX

PUSH AX

PUSH AX

ADD AL,0EH

OUT PortA,AL

POP AX

PUSH AX

PUSH BX

MOV BX,AX

MOV CX,4

S1: SHR BX,1

LOOP S1

MOV AL,SEGTAB[BX]

POP BX

OUT PortB,AL

CALL DELAY

POP AX

ADD AL,0DH

OUT PortA,AL

MOV AL,SEGTAB[BX]

OUT PortB,AL

CALL DELAY

POP AX

ADD AL,0BH

OUT PortA,AL

POP AX

PUSH AX

PUSH BX

MOV BX,AX

MOV CX,4

S2: SHR BX,1

LOOP S2

MOV AL,SEGTAB[BX]

POP BX

OUT PortB,AL

CALL DELAY

POP AX

ADD AL,07H

OUT PortA,AL

MOV AL,SEGTAB[BX]

OUT PortB,AL

CALL DELAY

JMP s

RET

;--------------------------------------------

; |

; Delay system running for a while |

; CX : contains time para. |

; |

;--------------------------------------------

DELAY1 PROC

PUSH CX

MOV CX,DelayLong ;

D0: LOOP D0

POP CX

RET

DELAY1 ENDP

;--------------------------------------------

; |

; Delay system running for a while |

; |

;--------------------------------------------

DELAY PROC

PUSH CX

MOV CX,DelayShort

D1: LOOP D1

POP CX

RET

DELAY ENDP

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

END

# 2-5

;-----------------------------------------------------------

;

; Build this with the "Source" menu using

; "Build All" option

;

;-----------------------------------------------------------

;

; 实验二示例程序

;-----------------------------------------------------------

; |

; |

; 功能：控制7段数码管的显示 |

; 编写：《嵌入式系统原理与实验》课程组 |

;-----------------------------------------------------------

DOSSEG

.MODEL SMALL ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 符号定义 |

;-----------------------------------------------------------

;

Patch\_Protues EQU IN AL, 0 ; Simulation Patch for Proteus, please ignore this line

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段;

;

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

Patch\_Protues ; Simulation Patch for Proteus,

; Please ignore the above code line.

;-----------------------------------------------------------

;主程序部分,向扩展的存储器内放数据 |

;-----------------------------------------------------------

MOV AX,4000H ;指定DS开始地址

MOV DS,AX

MOV BX,0H

; 内存中写入0FFh

L:

;MOV AX,BX

MOV BYTE PTR [BX],BL ;将AL中的数据以字节为单位送到DS:BX所指字节单元

INC BX

JNZ L

WT:

JMP WT

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

END ;指示汇编程序结束编译

# 实验3-1、2

;--------------------------------------------------------------------------

;

; Build this with the "Source" menu using

; "Build All" option

;

;--------------------------------------------------------------------------

;

; 实验三程序通用框架

;

;--------------------------------------------------------------------------

; 功能： 程序框架 |

; 编写：《嵌入式系统原理与实验》课程组 |

; 版本：3.5

; 修订：B

;--------------------------------------------------------------------------

DOSSEG

.MODEL SMALL ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

;-------------------------------------------------------------------------

; 符号定义 |

;-------------------------------------------------------------------------

;

;

; 8253芯片端口地址 （Port Address):

L8253T0 EQU 100H ; Timer0's port number in I/O space

L8253T1 EQU 102H ; Timer1's port number in I/O space

L8253T2 EQU 104H ; Timer2's port number in I/O space

L8253CS EQU 106H ; 8253 Control Register's port number in I/O space

;

; 8255芯片端口地址 （Port Address):

L8255PA EQU 121H ; Port A's port number in I/O space

L8255PB EQU 123H ; Port B's port number in I/O space

L8255PC EQU 125H ; Port C's port number in I/O space

L8255CS EQU 127H ; 8255 Control Register's port number in I/O space

;

; 中断矢量号定义

IRQNum EQU 20H ; 中断矢量号,要根据学号计算得到后更新此定义。

Patch\_Protues EQU IN AL, 0 ; Simulation Patch for Proteus, please ignore this line

;=======================================================================

; 宏定义

;=======================================================================

; 修补Proteus仿真的BUG，参见程序段中的使用说明

WaitForHWInt MACRO INTNum ; INTNum is the HW INT number

MOV AL, INTNum ;

OUT 0,AL ;

STI

CLI

ENDM

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段;

DelayShort dw 40 ; 短延时参量

DelayLong dw 4000 ; 长延时参量

; SEGTAB是显示字符0-F，其中有部分数据的段码有错误，请自行修正

SEGTAB DB 3FH ; 7-Segment Tube, 共阴极类型的7段数码管示意图

DB 06H ;

DB 5BH ; a a a

DB 4FH ; f b

DB 66H ; f b

DB 6DH ; f b

DB 7DH ; g g g

DB 07H ; e c

DB 7FH ; e c

DB 6FH ; e c

DB 77H ; d d d h h h

DB 7CH ; ----------------------------------

DB 39H ; b7 b6 b5 b4 b3 b2 b1 b0

DB 5EH ; DP g f e d c b a

DB 79H ;

DB 71H ;

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

;------------------------------------------------------------------------

Patch\_Protues ; Simulation Patch for Proteus,

; Please ignore the above code line.

;------------------------------------------------------------------------

START: ; 代码需要修改，否则无法编译通过

; 以下的代码可以根据实际需要选用

;CLI ; 关中断

;MOV AX, @DATA ;

;MOV DS, AX ; 初始化DS段寄存器

CALL INIT8255 ; 初始化8255

CALL INIT8253 ; 初始化8253

; MOV BL, IRQNum ; 取得中断矢量号

; CALL INT\_INIT ; 初始化中断向量表

Display\_Again:

CALL DISPLAY8255 ; 驱动四位七段数码管显示

CALL AccessPC

JMP Display\_Again

HLT

;===================================================================================

; Attention:

; The following codes is a Patching for Proteus 8086 Hardware Simulation Bug.

; Use these codes in the case you want the 8086 to halt and waiting for HW INT only!

; You can treat it as if it doesn't exist.

;

; If you need use HW INT, please uncomment it, or

; Don't modify it, leave it here just as it is.

; WaitForHWInt IRQNum ; 8086中断仿真补丁代码，直接忽略此代码行的存在即可

;====================================================================================

JMP Display\_Again

HLT ; 停止主程序运行

;=====================================================================================

;--------------------------------------------

; 子程序定义

;

;--------------------------------------------

; |

; INIT 8255 |

; |

;--------------------------------------------

INIT8255 PROC

; Init 8255 in Mode 0, L8255PA outPUT, L8255PB outPUT, L8255PCU inPUT, L8255PCL outPUT

; 计算8255的控制字

MOV DX, L8255CS

MOV AL, 10001000B

; 发送控制字到8255

OUT DX, AL

RET

INIT8255 ENDP

;--------------------------------------------

; |

; INIT 8253 |

; |

;--------------------------------------------

INIT8253 PROC

; 设定Timer0的初始值

MOV AL, 00110111B

MOV DX, L8253CS

OUT DX, AL

MOV AX, 10000;10ms

MOV DX, L8253T0

OUT DX, AL;设置低字节

MOV AL, AH

OUT DX, AL;设置高字节

; 设定Timer1的初始值

MOV AL, 01010111B

MOV DX, L8253CS

OUT DX, AL

MOV AX, 100;

MOV DX, L8253T1

OUT DX, AL

; 设定Timer2的初始值

RET

INIT8253 ENDP

;-------------------------------------------

; |

; Function: Polling 8255 PC0, send to PC6 |

; Input: None |

; Output: None |

;-------------------------------------------

AccessPC PROC

; 读取C口数值，并把PC0的值输出到PC6。

MOV DX, L8255PC ; DX指向C端口

IN AL, DX ; 读取C口数值

MOV AH, AL ; Save it

AND AL, 00000001b ; Test bit 0

AND AH, 10111111b ; 清除PC6

JZ APCOut

OR AH, 01000000b ; 置位PC6

APCOut:

MOV AL, AH ; Restore and Modify

OUT DX, AL ; 更新PC6

RET

AccessPC ENDP

;--------------------------------------------

; |

; DISPLAY STUDENTS ID |

; |

;--------------------------------------------

DISPLAY8255 PROC

; 点亮第一个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FEh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+0 ; 获取字模‘0’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第二个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FDh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+7 ; 获取字模‘7’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第三个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FBh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+2 ; 获取字模‘2’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第四个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0F7h ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+5 ; 获取字模‘5’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

RET

DISPLAY8255 ENDP

;--------------------------------------------------------------

; | |

; Function：DELAY FUNCTION |

; Input：None |

; Output: None |

;--------------------------------------------------------------

DELAY PROC

PUSH CX

MOV CX, DelayShort

D1: LOOP D1

POP CX

RET

DELAY ENDP

;-------------------------------------------------------------

; | |

; Function：INTERRUPT Vector Table INIT |

; Input: BL = Interrupt number |

; Output: None |

; |

;-------------------------------------------------------------

INT\_INIT PROC FAR ; 此部分程序有删节，需要修改

CLI ; Disable interrupt

MOV AX, 0

MOV ES, AX ; 准备操作中断向量表

; Put your code here

; 提示：可参考使用SET、OFFSET运算子获取标号的段地址值和段内偏移量

INT\_INIT ENDP

;--------------------------------------------------------------

; | |

; FUNCTION: INTERRUPT SERVICE Routine （ISR） |

; Input:: |

; Output: |

; |

;--------------------------------------------------------------

MYIRQ PROC FAR ; 此部分程序有删节，需要修改

; Put your code here

; 中断返回

MYIRQ ENDP

END ; 指示汇编程序结束编译

# 实验3-中断

;--------------------------------------------------------------------------

;

; Build this with the "Source" menu using

; "Build All" option

;

;--------------------------------------------------------------------------

;

; 实验三程序通用框架

;

;--------------------------------------------------------------------------

; 功能： 完成中断服务程序的编写，用BT1 按钮产生手动硬件中断的请求信号 |

; 编写：《嵌入式系统原理与实验》课程组 |

; 版本：3.4

; 修订：B

;--------------------------------------------------------------------------

DOSSEG

.MODEL SMALL ; 设定8086汇编程序使用Small model

.8086 ; 设定采用8086汇编指令集

;-----------------------------------------------------------

; 定义堆栈段 |

;-----------------------------------------------------------

.stack 100h ; 定义256字节容量的堆栈

;-------------------------------------------------------------------------

; 符号定义 |

;-------------------------------------------------------------------------

;

;

; 8253芯片端口地址 （Port Address):

L8253T0 EQU 100h ; Even Timer0's address in I/O space, 108H as well.

; 101H, 109H for high order byte access.

L8253T1 EQU 102h ; Timer1's address in I/O space

; 103H, 10BH for high order byte access.

L8253T2 EQU 104h ; Timer2'saddress in I/O space

; 105H, 10DH for high order byte access.

L8253CS EQU 106h ; 8253 Control Register's address in I/O space

; 107H, 10FH for high order byte access.

;

; 8255芯片端口地址 （Port Address):

L8255PA EQU 121h ; Odd Port A's address in I/O space, 129H as well.

; 120H, 128H for low order byte access.

L8255PB EQU 123h ; Odd Port B's address in I/O space, 12BH as well.

; 122H, 12AH for low order byte access.

L8255PC EQU 125h ; Odd Port C's address in I/O space, 13DH as well.

; 124H, 12CH for low order byte access.

L8255CS EQU 127h ; 8255 Control Register's port number in I/O space, 12FH as well

; 126H, 12EH for low order byte access.

;

; 中断矢量号定义

IRQNum EQU 85 ; 中断矢量号,要根据学号计算得到后更新此定义。

INTNum EQU 140h

Patch\_Proteus EQU IN AL, 0 ; Simulation Patch for Proteus, please ignore this line

;=======================================================================

; 宏定义

;=======================================================================

; 修补Proteus仿真的BUG，参见程序段中的使用说明

WaitForHWInt MACRO INTNum ; INTNum is the HW INT number

MOV DX, INTNum

IN AL,DX ;

OUT 0,AL

CLI

STI;

ENDM

;-----------------------------------------------------------

; 定义数据段 |

;-----------------------------------------------------------

.data ; 定义数据段;

DelayShort dw 40 ; 短延时参量

DelayLong dw 4000 ; 长延时参量

; SEGTAB是显示字符0-F，其中有部分数据的段码有错误，请自行修正

SEGTAB DB 3FH ; 7-Segment Tube, 共阴极类型的7段数码管示意图

DB 06H ;

DB 5BH ; a a a

DB 4FH ; f b

DB 66H ; f b

DB 6DH ; f b

DB 7DH ; g g g

DB 07H ; e c

DB 7FH ; e c

DB 6FH ; e c

DB 77H ; d d d h h h

DB 7CH ; ----------------------------------

DB 39H ; b7 b6 b5 b4 b3 b2 b1 b0

DB 5EH ; DP g f e d c b a

DB 79H ;

DB 71H ;

;-----------------------------------------------------------

; 定义代码段 |

;-----------------------------------------------------------

.code ; Code segment definition

.startup ; 定义汇编程序执行入口点

;------------------------------------------------------------------------

Patch\_Proteus ; Simulation Patch for Proteus,

; Please ignore the above code line.

;------------------------------------------------------------------------

START: ; 代码需要修改，否则无法编译通过

; 以下的代码可以根据实际需要选用

CALL INIT8255 ; 初始化8255

CALL INIT8253 ; 初始化8253

MOV BL, IRQNum

; 取得中断矢量号

CALL INT\_INIT

Display\_Again:

;INT IRQNum ; Poll PC0, Modify PC6

WaitForHWInt INTNum ;

CALL DISPLAY8255 ; 驱动四位七段数码管显示

;CALL AccessPC

; 取得中断矢量号

; 初始化中断向量表

; 开中断

;===================================================================================

; Attention:

; The following codes is a Patching for Proteus 8086 Hardware Simulation Bug.

; Use these codes in the case you want the 8086 to halt and waiting for HW INT only!

; You can treat it as just the following assembly line:

;

;Locallable: JMP Locallable ; Endless loop

;

; If you need this function, uncomment it, or

; Please don't modify it, leave it just as it is.

;====================================================================================

; 8086中断仿真补丁代码，可根据需要使用。

JMP Display\_Again

HLT ; 停止主程序运行

;=====================================================================================

;--------------------------------------------

; 子程序定义

;

;--------------------------------------------

; |

; INIT 8255 |

; |

;--------------------------------------------

INIT8255 PROC

; Init 8255 in Mode x, L8255PA xPUT, L8255PB xPUT, L8255PCU xPUT, L8255PCL xPUT

; 计算8255的控制字

; Init 8255 in Mode 0, L8255PA Output, L8255PB Output, L8255PC UP Input, L8255PC LOW Output

MOV DX, L8255CS

MOV AL, 10000001B ; Control Word

; 发送控制字到8255

OUT DX, AL

RET

INIT8255 ENDP

;--------------------------------------------

; |

; INIT 8253 |

; |

;--------------------------------------------

INIT8253 PROC

; 计算8253的控制字

MOV DX, L8253CS

MOV AL, 00110110B ; 【Timer0: 16bit, method3, binary】

OUT DX, AL

MOV AL, 01010110B ; 【Timer1: 8bit, method3, binary】

OUT DX, AL

MOV AL, 10010000B ; 【Timer2: 8bit, method0, binary】

OUT DX, AL

; 设定Timer0的初始值

MOV DX, L8253T0

MOV AX, 10000 ; 【f = 1MHz, 10000times = 10ms】

OUT DX, Al

MOV AL, AH ; 【get higher 8 bit】

OUT DX, AL

; 设定Timer1的初始值

MOV DX, L8253T1

MOV AL, 100 ; 【t = 10ms, 100times = 1s】

OUT DX, AL

; 设定Timer2的初始值

;MOV DX, L8253T2

;MOV AL, 200 ; 【f = 500kHz, 200times = 400us】

;OUT DX, AL

RET

RET

INIT8253 ENDP

;-------------------------------------------

; |

; Function: Polling 8255 PC0, send to PC6 |

; Input: None |

; Output: None |

;-------------------------------------------

AccessPC PROC

; 读取C口数值，并把PC0的值输出到PC6。

MOV DX, L8255PC ; DX指向C端口

IN AL, DX ; 读取C口数值

MOV AH, AL ; Save it

AND AL, 00000001b ; Test bit 0

MOV AH, 10111111b ; 清除PC6

JZ APCOut

OR AH, 01000000b ; 置位PC6

APCOut:

MOV AL, AH ; Restore and Modify

OUT DX, AL ; 更新PC6

RET

AccessPC ENDP

;--------------------------------------------

; |

; DISPLAY STUDENTS ID |

; |

;--------------------------------------------

DISPLAY8255 PROC

; 点亮第一个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FEh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+0 ; 获取字模‘0’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第二个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FDh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+7 ; 获取字模‘7’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第三个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0FBh ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+2 ; 获取字模‘2’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

; 点亮第四个七段管

MOV DX, L8255PA ; 指向PA

MOV AL, 0F7h ; 设定第一个七段管的PA值

OUT DX, AL ; 输出到PA

MOV AL, SEGTAB+5 ; 获取字模‘5’的值

MOV DX, L8255PB ; 指向PB

OUT DX, AL ; 输出字模到PB口，点亮

CALL DELAY ; 延时

RET

DISPLAY8255 ENDP

;--------------------------------------------------------------

; | |

; Function：DELAY FUNCTION |

; Input：None |

; Output: None |

;--------------------------------------------------------------

DELAY PROC

PUSH CX

MOV CX, DelayShort

D1: LOOP D1

POP CX

RET

DELAY ENDP

;-------------------------------------------------------------

; | |

; Function：INTERRUPT Vector Table INIT |

; Input: BL = Interrupt number |

; Output: None |

; |

;-------------------------------------------------------------

INT\_INIT PROC FAR ; 此部分程序有删节，需要修改

CLI ; Disable interrupt

PUSH AX

PUSH BX

PUSH CX

PUSH ES

MOV AX, 0

MOV ES, AX ; 准备操作中断向量表

XOR BH, BH

MOV CL, 2

SHL BX, CL

; 【Addr = IRQNum \* 4】

MOV AX, OFFSET MYIRQ ; 【IRQ IP】

MOV ES:[BX], AX

MOV AX, SEG MYIRQ ; 【IRQ CS】

MOV ES:[BX + 2], AX

; 提示：可参考使用SET、OFFSET运算子获取标号的段地址值和段内偏移量

POP ES

POP CX

POP BX

POP AX

RET

INT\_INIT ENDP

;--------------------------------------------------------------

; | |

; FUNCTION: INTERRUPT SERVICE Routine （ISR） |

; Input:: |

; Output: |

; |

;--------------------------------------------------------------

MYIRQ PROC FAR ; 此部分程序有删节，需要修改

; Put your code here

PUSH DX

PUSH AX

MOV DX, L8255PC

IN AL, DX

TEST AL, 10000000B ; 【if PC7 = 1】

JZ PC7ON

AND AL, 01111111B ; 【reset PC7】

PC7OUT: OUT DX, AL

POP AX

POP DX

IRET ; 中断返回

PC7ON: OR AL, 10000000B ; 【set PC7】

JMP PC7OUT

IRET ; 中断返回

MYIRQ ENDP

END ; 指示汇编程序结束编译