8086 Lab1 实验报告

**汪润中 515021910285**

第三题，十字路口红绿灯源代码：

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

;实验一 第三题 |

; |

;功能：实现路口红绿灯循环控制和显示 |

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

; |

;作者：汪润中 515021910285 |

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

.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 |

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

PortOut EQU 88h ;定义输出端口号

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

; 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

DELAY DW 23500

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

; 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

;

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

;主程序部分,循环控制路口红绿灯并送显示 |

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

MAIN PROC

Again**:**

;STAGE 1

**MOV** **AL,**00110011B ;Green on & Red on

**OUT** PortOut**,AL**

**CALL** LongLongDelay ;delay of STAGE 1

;STAGE 1.5, Green light flash

**MOV** **CX,**04H ;set flash counter

S1p5**:**

**MOV** **AL,**00110011B ;Green on & Red on

**OUT** PortOut**,AL**

**CALL** ShortDelay

**MOV** **AL,**00110111B ;Green off & Red on

**OUT** PortOut**,AL**

**CALL** ShortDelay

**LOOP** S1p5 ;Loop STAGE 1.5 - green light flash

;STAGE 2

**MOV** **AL,**00110101B ;Yellow on & Red on

**OUT** PortOut**,AL**

**CALL** LongDelay

;STAGE 3

**MOV** **AL,**00011110B ;Red on & Green on

**OUT** PortOut**,AL**

**CALL** LongLongDelay

;STAGE 3.5, Green light flash

**MOV** **CX,**04H ;set flash counter

S3p5**:**

**MOV** **AL,**00011110B ;Red on & Green on

**OUT** PortOut**,AL**

**CALL** ShortDelay

**MOV** **AL,**00111110B ;Red on & Green off

**OUT** PortOut**,AL**

**CALL** ShortDelay

**LOOP** S3p5

;STAGE 4

**MOV** **AL,**00101110B ;Red on & Yellow on

**OUT** PortOut**,AL**

**CALL** LongDelay

**JMP** Again

MAIN ENDP

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

;函数部分，三种不同时长的延迟函数 |

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

LongLongDelay PROC

**PUSH** **CX**

**MOV** **CX,**5

longlongloop**:** **CALL** LongDelay

**Loop** longlongloop

**POP** **CX**

**RET**

LongLongDelay ENDP

LongDelay PROC

**PUSH** **CX**

**MOV** **CX,**2

longloop**:** **CALL** ShortDelay

**LOOP** longloop

**POP** **CX**

**RET**

LongDelay ENDP

ShortDelay PROC

**PUSH** **CX**

**MOV** **CX,DS:**DELAY

shortloop**:** **LOOP** shortloop

**POP** **CX**

**RET**

ShortDelay ENDP

END ;指示汇编程序结束编译

第四题，更改选片地址源代码:

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

;实验一 第四题 |

; |

;功能：读取开关量的状态取反后送到LED显示 |

;74LS244带有8位拨位开关，地址为90H or 92H or 94H or 96H |

;74LS273带有8个发光二极管，地址为A0H or A2H or A4H or A6H |

; |

;作者：汪润中 515021910285 |

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

.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 |

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

;

PortIn EQU 90H ;定义输入端口号

PortOut EQU 0A0H ;定义输出端口号

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

; 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

;MyVar DB ? ; 定义MyVar变量

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

; 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

;

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

;主程序部分,读取开关量状态取反后送显示 |

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

Again**:**

**MOV** **AL,**0 ;

**IN** **AL,**PortIn ;读取开关量状态

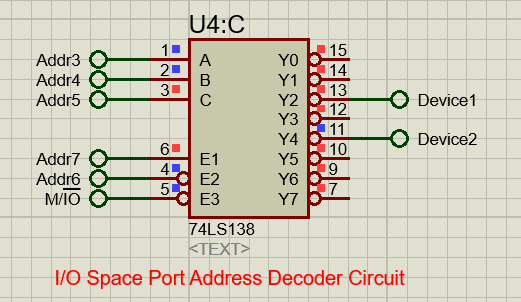
**NOT** **AL** ;取反

**OUT** PortOut**,AL** ;送显示

**JMP** Again ;跳转循环执行

END ;指示汇编程序结束编译

第四题，更改的连线图部分



第五题，向00h~FFh的内存地址中写入00h~FFh，从0x00000到0x09FFF

源代码：

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

;实验一 第四题 |

; |

;功能：初始化0x00000到0x09FFF内存 |

;内容为00h到FFh循环 |

; |

;作者：汪润中 515021910285 |

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

.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 |

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

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

; 定义堆栈段 |

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

.stack 100h ; 定义256字节容量的堆栈

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

; 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

;

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

;主程序部分,将0x0000到0x8FFF按要求初始化， |

;0x9000开始为指令和堆栈 |

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

MAIN PROC

**MOV** **AX,**0000H ;初始化AX

setAllWords**:** ;按要求初始化全部内存

**MOV** **DS,AX** ;将DS移到开始位置

**CALL** setAWord ;第一次循环设置0x0000至0x00FF的值，以此类推

**ADD** **AX,**10H ;更新AX，后续赋值给DS

**CMP** **AX,**1000H ;比较是否超出上界

**JBE** setAllWords ;没有超出上界，则继续执行

**HLT** ;程序执行完毕，挂起

MAIN ENDP

setAWord PROC

**PUSH** **CX** ;堆栈中暂存数据

**PUSH** **BX**

**PUSH** **AX**

**MOV** **CX,**80H ;初始化循环变量

wordloop**:**

**MOV** **BX,CX**

**DEC** **BX**

**SHL** **BX,**1

**MOV** **AL,BL**

**MOV** **AH,BL**

**INC** **AH**

**MOV** **[BX],AX** ;BX存储目标地址，AX存储将要赋给改地址的值

**LOOP** wordloop ;循环

**POP** **AX** ;从堆栈中取出数据

**POP** **BX**

**POP** **CX**

**RET**

setAWord ENDP

END ;指示汇编程序结束编译

程序运行结果：



实验感想：

1. 调用函数时，要将使用到的寄存器值存入堆栈，保证调用前后不会对其他程序产生影响。例如如果使用到CX，就应在代码段开头PUSH CX，最后再POP CX

2. DS只接受寄存器对它的修改，不接受立即数。因此，只能MOV DS, AX

3. 字母开头的十六进制数需要在前面加上数字0才能被正常读取，比如F3H应当写作0F3H

4. 使用方括号作为offset直接寻址时，只有SI、DI和BX寄存器是可以使用的

5. proteus程序在仿真时，点击菜单栏中的debug选项，可以监视8086寄存器、内存等

6. 需要给单次、顺序执行的8086程序设定终点。在主程序末尾加上HTL或者死循环都可以解决程序运行遇到未知代码的报错