

# 第四次实验报告

## 一、实验要求

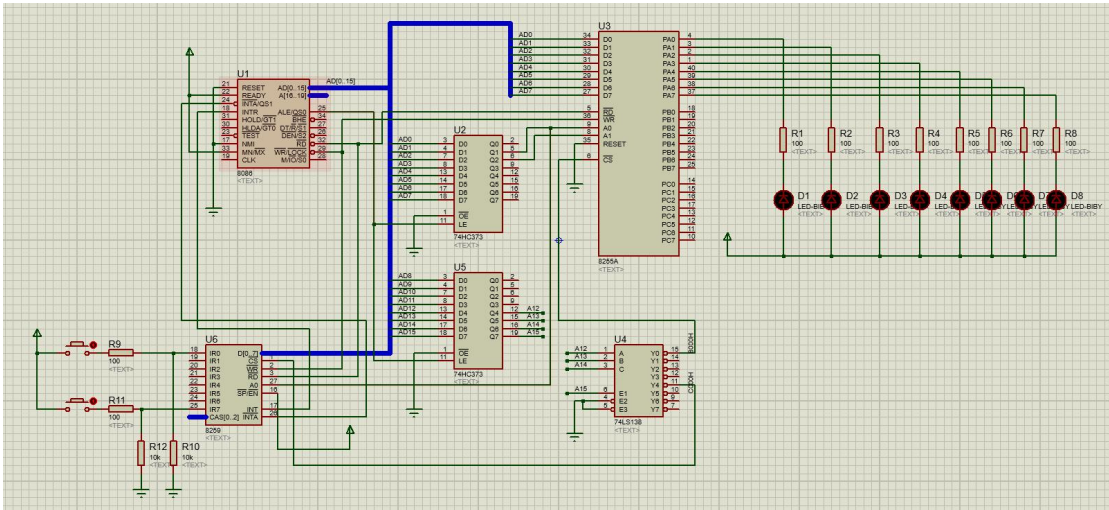
- 1. 利用 8086 控制 8259 可编程中断控制器，实现对外部中断的响应和处理。
- 2. 要求程序中每次中断进行计数，并将计数结果用 8255 的 PA 口输出到发光二极管显示。
- 3. 要求一个中断表示加一，一个中断表示减一，计数结果在二极管处显示。

## 二、实验目的

- 1. 实现 8086 与 8259 的连接。
- 2. 实现 8086 对 8259 的编程控制方法。
- 3. 实现 8259 的多片级联。

## 三、实验电路和连线

- 1. 实验元件  
两片 74HC373，一片 74LS138，一片 8086，一片 8255，一片 8259，两个 BUTTON，八个 LED-BIBY，八个 RES。
- 2. Proteus 实验电路



- 3. 硬件地址申明

接线孔 1	接线孔 2
8255CS	08000H-08FFFH
8259CS	0C00H-0CFFFH
PA0—PA7	D1—D8

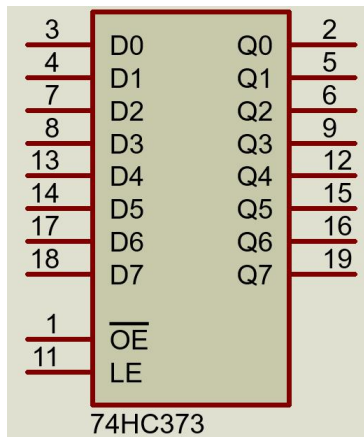
- 4. 注意事项

8086 的外部中断必须通过外接中断控制器才可以进行外部中断的处理. 在编程时应注意:

- 1、正确地设置可编程中断控制器的工作方式。
- 2、必须正确地设置中断向量表和中断服务程序的入口地址。

8259 可外接 8 个中断源，本实验只响应 INT0 中断，8259 也可以多级连接以响应多个中断源. 将单脉冲信号接到 8259 的 INT0 脚. 每次中断时，可以看到 LED 显示加 1。

## 四、实验说明

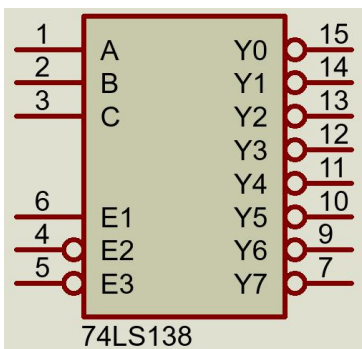


### 1. 74HC373 数据锁存芯片

1 脚 OE, 为使能输出, 低电平有效.

当 1 脚是高电平时, 不管输入 3、4、7、8、13、14、17、18 如何, 也不管 11 脚(锁存控制端 G)如何, 输出 2(Q0)、5(Q1)、6(Q2)、9(Q3)、12(Q4)、15(Q5)、16(Q6)、19(Q7) 全部呈现高阻状态. 当 1 脚是低电平时, 只要 11 脚(锁存控制端 G)上出现一个下降沿, 输出 2(Q0)、5(Q1)、6(Q2)、9(Q3)、12(Q4)、15(Q5)、16(Q6)、19(Q7) 立即呈现输入脚 3、4、7、8、13、14、17、18 的状态. 锁存端 LE 由高变低时, 输出端 8 位信息被锁存, 直到 LE 端再次有效.

当三态门使能信号 OE 为低电平时, 三态门导通, 允许 Q0~Q7 输出, OE 为高电平时, 输出悬空.



### 2. 74LS138 三线八线译码器

A0~A2: 地址输入端

STA(E1): 选通端

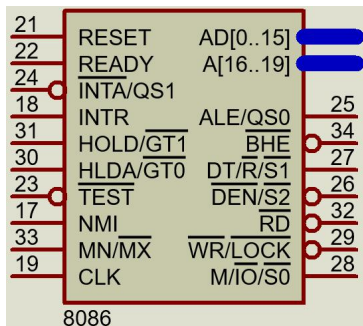
/STB(/E2)、/STC(/E3): 选通端(低电平有效)

/Y0~Y7: 输出端(低电平有效)

VCC: 电源正

GND: 地

A0~A2 对应 Y0——Y7; A0, A1, A2 以二进制形式输入, 然后转换成十进制, 对应相应 Y 的序号输出低电平, 其他均为高电平.



### 3. 8086 处理器

AD15—AD0 (AddressDataBus): 地址/数据复用信号输入/输出引脚(16 个), 分时输出.

A19/s6—A15/s3 (AddressStatusBus): 地址/状态复用信号输出引脚(4 个), 分时输出地址的高 4 位及状态信息.

NMI (Non-MaskableInterrupt)、INTR (InterruptRequest): 中断请求信号输入引脚(2), 引入中断源向 CPU 提出的中断请求信号, 高电平有效, 前者为非屏蔽中断请求, 后者可屏蔽中断请求信号.

/RD (Read): 读控制输出信号引脚(1).

CLK/ (Clock): 时钟信号输入引脚(1).

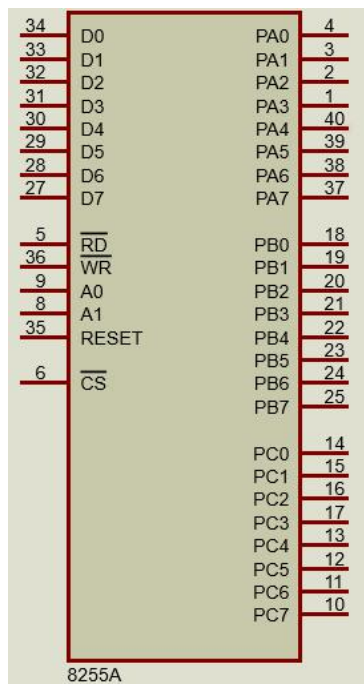
Reset (Reset): 复位信号输入引脚(1), 高电平有效.

READY (Ready): “准备好” 状态信号输入引脚(1), 高电平有效, 该信号是协调 CPU 与内存单元或 I/O 端口之间进行信息传送的联络信号.

/TEST (Test): 测试信号输入引脚(1), 低电平有效, TEST 信号与 WAIT 指令结合起来使用, CPU 执行 WAIT 指令后, 处于等待状态, 当 TEST 引脚输入低电平时, 继续执行被暂停执行的指令.

MN/MX (Minimum/MaximumModelControl) 最小/最大模式设置信号输入引脚(1), 该输入引脚电平的高、低决定了 CPU 工作在最小模式还是最大模式, 高电平 CPU 工作于最小模式下.

/BHE/S7 (BusHighEnable/Status): 高 8 位数据允许/状态复用信号引出引脚(1), 输出. 表示高 8 为数据线 D15—D8 上的数据有效



#### 4. 8255 可编程并行接口

D7~D0: 双向数据线.

CPU 通过它向 8255A 发送命令、数据;

8255A 通过它向 CPU 回送状态、数据.

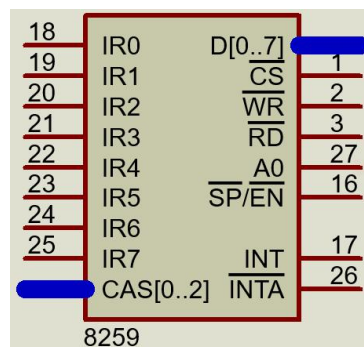
CS: 选片信号线, 该信号低电平有效, 由系统地址总线经 I/O 地址译码器产生.

CPU 通过发高位地址信号使它变成低电平时, 才能对 8255A 进行读写操作. 当 CS 为高电平时, 切断 CPU 与芯片的联系. A1, A0: 芯片内部端口地址信号线, 与系统地址总线低位相连. 该信号用来寻址 8255A 内部寄存器. 两位地址, 可形成片内 4 个端口地址.

RD: 读信号线, 该信号低电平有效. CPU 通过执行 IN 指令, 发读信号将数据或状态信号从 8255A 读至 CPU.

WR: 写信号线, 该信号低电平有效. CPU 通过执行 OUT 指令, 发写信号, 将命令代码或数据写入 8255A.

RESET: 复位信号线, 该信号高电平有效. 它清除控制寄存器并将 8255A 的 A、B、C3 个端口均置为输入方式; 输出寄存器和状态寄存器被复位, 并且屏蔽中断请求.



#### 5. 8259 可编程中断控制器

D7~D0: 双向、三态数据线, 与系统数据总线相连. 对 8259A 编程时, 命令字由此写入; 在第二个中断响应总线周期中, 中断类型码由此传给 CPU.

/RD: 读信号, 输入, 与系统控制总线/IOR 相连.

/WR: 写信号, 输入, 与系统控制总线/IOW 相连.

A0: 片内寄存器寻址信号, 输入, 用于对片内寄存器端口寻址.

/CS: 片选信号, 输入.

/SP//EN: 双功能双向信号.

INT: 中断请求信号, 输出, 与 CPU 的中断请求信号线 INT 相连. 在级联方式下, 从片的 INT 与主片的 IR7~IR0 中的某一根连接在一起.

/INTA: 中断响应信号, 输入, 与 CPU 的中断响应信号线/INTA 相连.

CAS2~CAS0: 级联控制线, 主片的 CAS2~CAS0 与从片的 CAS2~CAS0 对应相连.

IR7~IR0: 中断请求输入信号, 由外设输入.

VCC: +5V 电源输入信号.

GND: 电源地.

### 五、实验程序流程

```
MODE EQU 80H ; 8255 工作方式
MODE EQU 80H ; 8255 工作方式
PA8255 EQU 8000H ; 8255 PA 口输出地址
CTL8255 EQU 8006H
ICW1 EQU 00010011B ; 单片 8259, 上升沿中断, 要写 ICW4
ICW2 EQU 00100000B ; 中断号为 20H
```

```

ICW4 EQU 0000001B ; 工作在 8086/88 方式
OCW1 EQU 0000000B ; 只响应 INT0 中断
CS8259A EQU 0C000H ; 8259 地址
CS8259B EQU 0C002H
CODE SEGMENT
ASSUME CS:CODE, DS: DATA, SS:STACK
ORG 800H
START:
MOV AX, DATA
MOV DS, AX
MOV AX, STACK
MOV SS, AX
MOV AX, TOP
MOV SP, AX
MOV DX, CTL8255
MOV AL, MODE
OUT DX, AL
CLI
PUSH DS
MOV AX, 0
MOV DS, AX
MOV BX, 128 ; 0X20 * 4 中断号
MOV AX, CODE
MOV CL, 4
SHL AX, CL ; X 16
ADD AX, OFFSET INTDEC ; 中断入口地址 (段地址为 0)
MOV [BX], AX
MOV AX, 0
INC BX
INC BX
MOV [BX], AX ; 代码段地址为 0
MOV AX, 0
MOV DS, AX
MOV BX, 156 ; 0X27 * 4 中断号
MOV AX, CODE
MOV CL, 4
SHL AX, CL ; X 16
ADD AX, OFFSET INTINC ; 中断入口地址 (段地址为 0)
MOV [BX], AX
MOV AX, 0
INC BX
INC BX
MOV [BX], AX ; 代码段地址为 0

```

```

POP DS
CALL IINIT
MOV AL, CNT ; 计数值初始为 0xFF,全灭
MOV DX, PA8255
OUT DX, AL
STI
LP: ; 等待中断, 并计数.
NOP
JMP LP
IINIT:
MOV DX, CS8259A
MOV AL, ICW1
OUT DX, AL
MOV DX, CS8259B
MOV AL, ICW2
OUT DX, AL
MOV AL, ICW4
OUT DX, AL
MOV AL, OCW1
OUT DX, AL
RET
INTDEC:
CLI
MOV DX, PA8255
DEC CNT
MOV AL, CNT
OUT DX, AL ; 输出计数值
MOV DX, CS8259A
MOV AL, 20H ; 中断服务程序结束指令
OUT DX, AL
STI
IRET
INTINC:
CLI
MOV DX, PA8255
INC CNT
MOV AL, CNT
OUT DX, AL ; 输出计数值
MOV DX, CS8259A
MOV AL, 20H ; 中断服务程序结束指令
OUT DX, AL
STI
IRET

```

```
CODE ENDS
DATA SEGMENT
CNT DB 0FFH
DATA ENDS
STACK SEGMENT 'STACK'
STA DB 100 DUP(?)
TOP EQU LENGTH STA
STACK ENDS
END START
```

## 六、实验结果和体会

详细了解了 8086 控制 8259 可编程中断控制器的中断控制相应，以及 8086，8255，8259 的各个引脚的功能用法，对 Proteus 的程序应用开始逐渐熟练，但是在程序编写方面还是存在许多问题，很难完全复现出原程序，希望在后期的学习中可以加强对于芯片编程的学习。