基于 8259A、8255A 和 8253A 的流水灯设计与仿真

徐　旭

(通信工程)

[摘要]本文使用 8086CPU 作为主控芯片，选择可编程计数器 8253A 芯片引入计时信号，选择可编程中断控制器 8259A 产生中断并按照相应的中断服务子程序控制 8255A 可编程并行接口芯片输出控制信号，驱动电路中的 8 只 LED 产生流水灯效果。本课程设计主要包括硬件设计和软件设计，其中所有汇编代码均使用 EMU8086 编写，硬件电路在 Proteus7.8 进行搭建和仿真，仿真结果显示，本设计中程序与电路运行正常，LED 会间隔相同时间后被循环点亮。

[关键词]微机原理课程设计；LED 流水灯；接口芯片仿真

一、绪论

1.1. 任务分析

本次设计是基于 8086 处理器的 LED 流水灯电路，通过 CPU 与总线、译码器、8 只 LED 灯以及各接口芯片之间的配合，实现循环点亮的效果。本设计中，8086CPU 的主频为 1500KHz，由外部时钟频率为 2MHz 的 8253A 芯片采用两计数通道级联实现初值 N 为 10000 的计数，产生周期为 0.005s 的方波作为中断触发信号送入 8259A 中。对该中断控制芯片进行设置，在中断服务子程序对 8255A 芯片并行口 PA 输出进行定义，使得 PA 口八只引脚循环输出高电平从而点亮对应的 LED 灯。

软件代码使用汇编语言编写，并在 EMU8086 中编译产生可执行文件；硬件电路在 Proteus7.8 中搭建，配合代码中各芯片地址进行电路布线，在引入编译产生的可执行文件后对电路仿真。

1.2. 总体设计思路

在本设计中 8086CPU 通过译码电路对各个接口芯片进行片选，并通过总线进行通信，设计思路框图如图 1 所示：

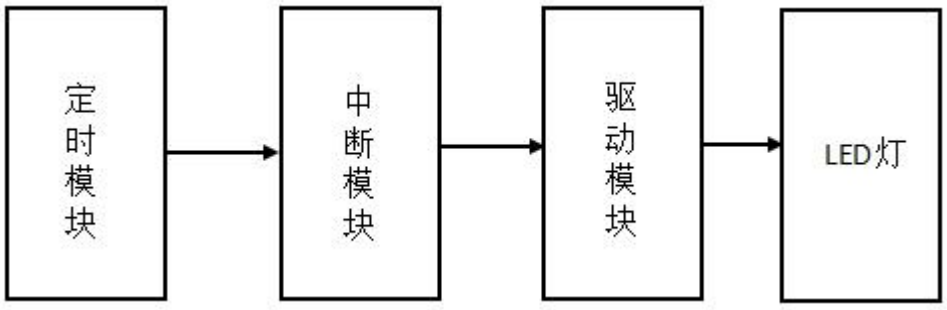

图 1 总体设计框图

（1）在定时模块中，笔者使用了 8253A 芯片的计数通道 0 和通道 1 均采用模式 3（即方波发生器）产生定时信号，外部时钟频率 $f_{CLK_0}=2MHz$，两计数器初值均为 100。

（2）在中断模块中，笔者对 8259A 芯片各控制字以及地址线进行设置，使用 IR0 引脚引入 8253A 产生的方波作为触发信号，中断服务子程序 INT0 的中断类型号设置为 40H 并使用循环移位指令 ROR 控制并行口的输出值。

（3）在驱动模块中，笔者设置 8255A 的 PA 口为输出，其初值为 80H 以点亮 LED8 作为初始态，在中断子程序 INT0 中采用循环移动 1 位的方法更改 PA 各端口的输出电平，从而驱动 PA 口所接的 LED 灯使之循环点亮。

二、硬件设计分析

图 2 为笔者在 Proteus7.8 软件中搭建的电路，使用到的元件包括一片 8086 处理器芯片、3 片 74273 芯片、一片 8259A 芯片、一片 8255A 芯片、一片 8253A 芯片、8 只 LED 二极管和四个非门。为了方便调试 8253A 的输出波形，故在电路中接入一个示波器。

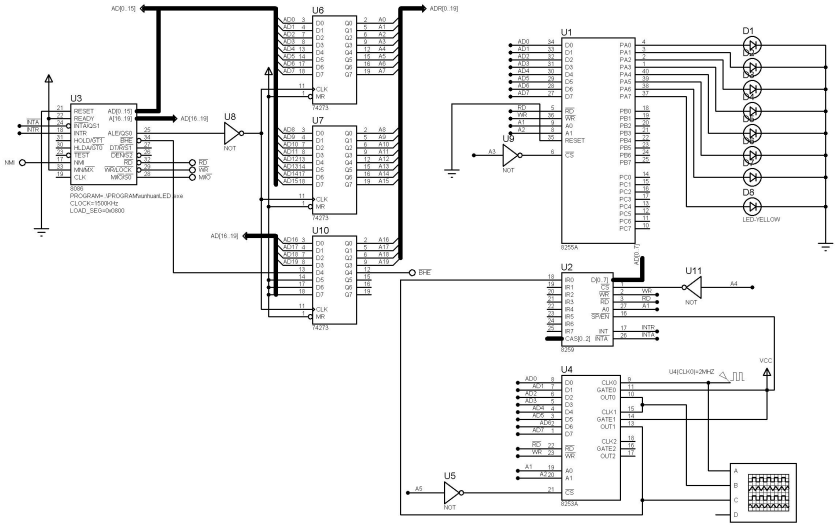

图 2 硬件电路图

端口	地址
计数器 0	100000B
计数器 1	100010B
计数器 2	100100B
控制端口	100110B

表 1 8253A 端口地址

对于 8253A 芯片，电路中各计数器以及控制端口的地址如表 1 所示。笔者考虑设置计数器初值为 10000，为了更充分体会 8253A 芯片的使用方式，采用两计数器级联的方式实现，即设置计数器 0 的外部时钟频率为 $f_{CLK_0}=2MHz$，初值为 100，输出的 OUT0 作为计数器 1 的外部时钟信号，接着设置计数器 1 的初值也为 100，GATE0 和 GATE1 引脚均接高电平，此时 OUT1 便可以输出周期为 0.005s 的连续方波信号。

计数器	控制字
计数器 0	00010110B
计数器 1	01010110B

表 2 8253A 控制字

根据上述需求，计数器 0 和计数器 1 均工作在方式 3，计数初值均为 100，采用二进制数表示，所以对应的控制字如表 2 所示。

对于 8259A 芯片，电路中控制端口的奇偶地址分别为 10010B 和 10000B。设置 ICW1 为边沿触发，使用单片 8259A 并使用 ICW4 控制字；中断类型号即 ICW2 设置为 40H；ICW4 不使用缓冲方式，采用非特殊的全嵌套方式并且正常结束中断，上述控制字中 ICW1 写入偶地址，其它写入奇地址。设置中断优先级为非循环方式并发出中断结束命令，OCW2 操作字写入偶地址。其命令字如表 3 所示。

命令字	数据
ICW1	00010011B
ICW2	40H
ICW4	00000001B
OCW2	00100000B

表 3 8259A 命令字

对于 8055A 芯片，电路中设置各端口地址如表 4 所示。由于设置初始状态为只有 PA8 输出高电平，所以要求 PA 口输出工作在方式 0，方式选择控制字中其他位默认为 0，故控制字应为 80H。

端口	地址
PA	01000B
PB	01010B
PC	01100B
控制字	01110B

表 4 8255A 端口地址

三、软件设计分析

3.1. 流程图设计

根据前面硬件电路的设计以及总体思路要求，程序主流程图以及相应的中断服务子程序 INT0 流程图如图 3 所示。

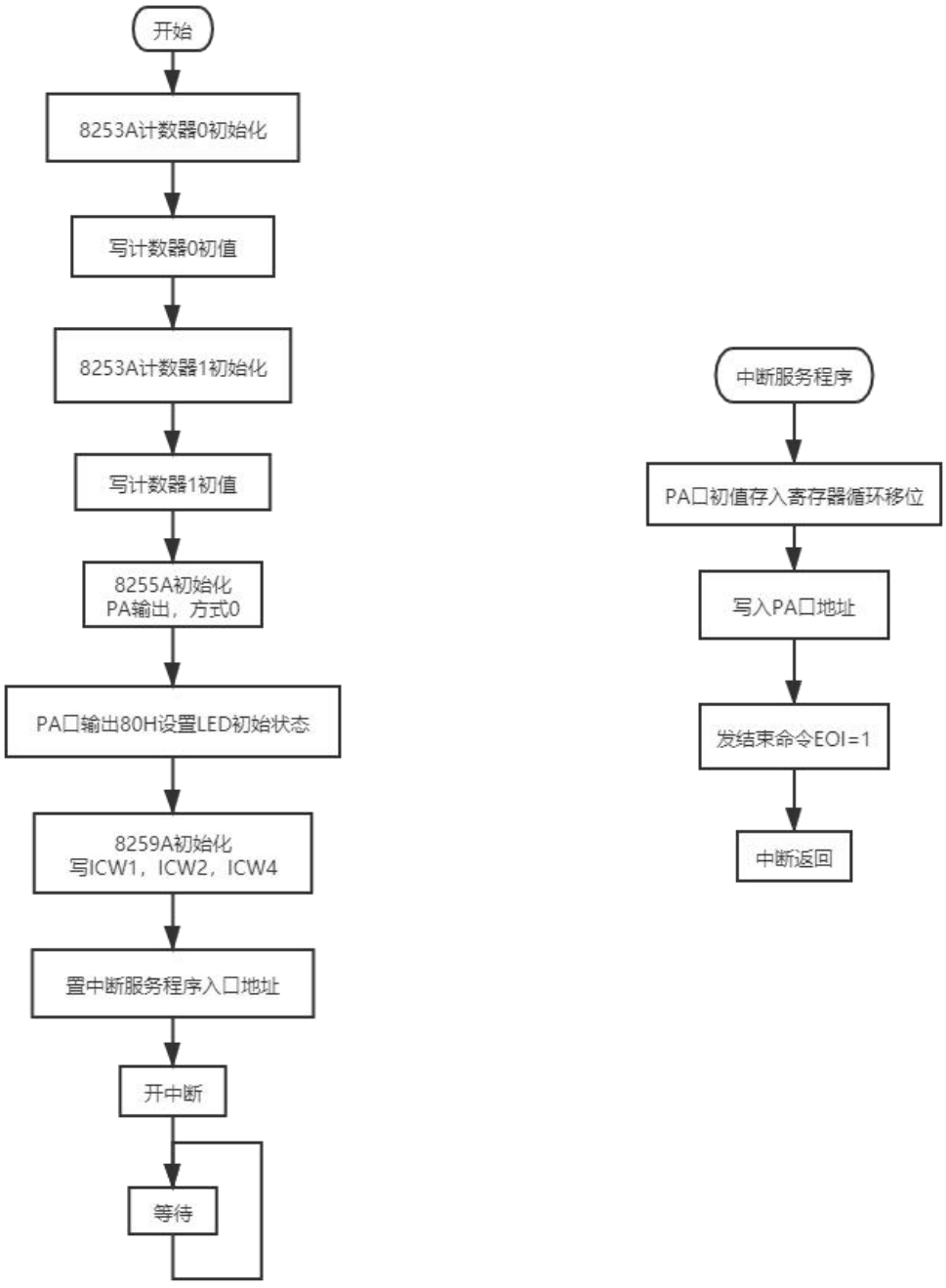

图 3 程序流程图

3.2. 程序代码

```
DATA SEGMENT
DATA ENDS

STACK SEGMENT STACK
    ST DB 10 DUP(0)
STACK ENDS

CODE SEGMENT
    ASSUME CS:CODE,DS:DATA,SS:STACK
START:
    MOV AX,DATA
    MOV DS,AX
    MOV DX,100110B    ;8253 INIT
    MOV AL,00010110B  ;COUNTER0,MODE3,BINARY-DIGITS
    OUT DX,AL

    MOV DX,100000B    ; COUNTER0
    MOV AL,64H      ; 100
    OUT DX,AL

    MOV DX,100110B    ;8253 INIT
    MOV AL,01010110B  ;COUNTER1,MODE3,BINARY-DIGITS
    OUT DX,AL

    MOV DX,100010B    ; COUNTER1
    MOV AL,64H      ; 100
    OUT DX,AL

    MOV DX,01110B     ; 8255 初始化
    MOV AL,80H      ; A 口输出，方式 0,B OUT,MODE 0
    OUT DX,AX

    MOV BL,80H      ; LED0 灯亮(高电平灯亮)
    MOV AL,BL
    MOV DX,01000B
    OUT DX,AL       ;PA7 灯亮

    MOV AL,13H      ; 00010011B，ICW1：边沿触发，单片，要 ICW4
    MOV DX,10000B   ; 8259 地址
    OUT DX,AL
    MOV AL,40H      ; ICW2 中断类型号为 40H
    MOV DX,10010B
```

```
    OUT DX,AL
    MOV AL,01H     ; ICW4 不用缓冲方式，正常中断结束，非特殊的全嵌套方式
    OUT DX,AL

    MOV AX,0
    MOV DS,AX      ; 数据段清零

    LEA AX,INT0    ; 写 8259 中断程序的入口地址
    MOV DS:[4*40H],AX  ; 把中断服务程序的入口地址偏移量送中断向量表
    MOV AX,CS
    MOV DS:[4*40H+2],AX ; 把中断服务程序的入口地址段地址送中断向量表

    STI            ; 开中断

AGAIN:
    MOV DX,8000H
    MOV AL,40H
    OUT DX,AL
    JMP AGAIN

INT0 PROC NEAR     ; 8259 中断服务程序
    ROR BL,1       ; 右循环 1 次
    MOV AL,BL
    MOV DX,01000B
    OUT DX,AL      ; PA 口灯亮

    MOV DX,10000B
    MOV AL,20H     ; OCW2 发结束命令 EOI=1
    OUT DX,AL

    IRET
INT0 ENDP
CODE ENDS
    END START
```

四、仿真结果

汇编程序在 EMU8086 中成功编译，将可执行文件写入 Proteus7.8 软件的 8086CPU 中仿真后发现，8 只 LED 灯可以被循环点亮（图 4）。8253A 计数器输出波形如图 5 所示。

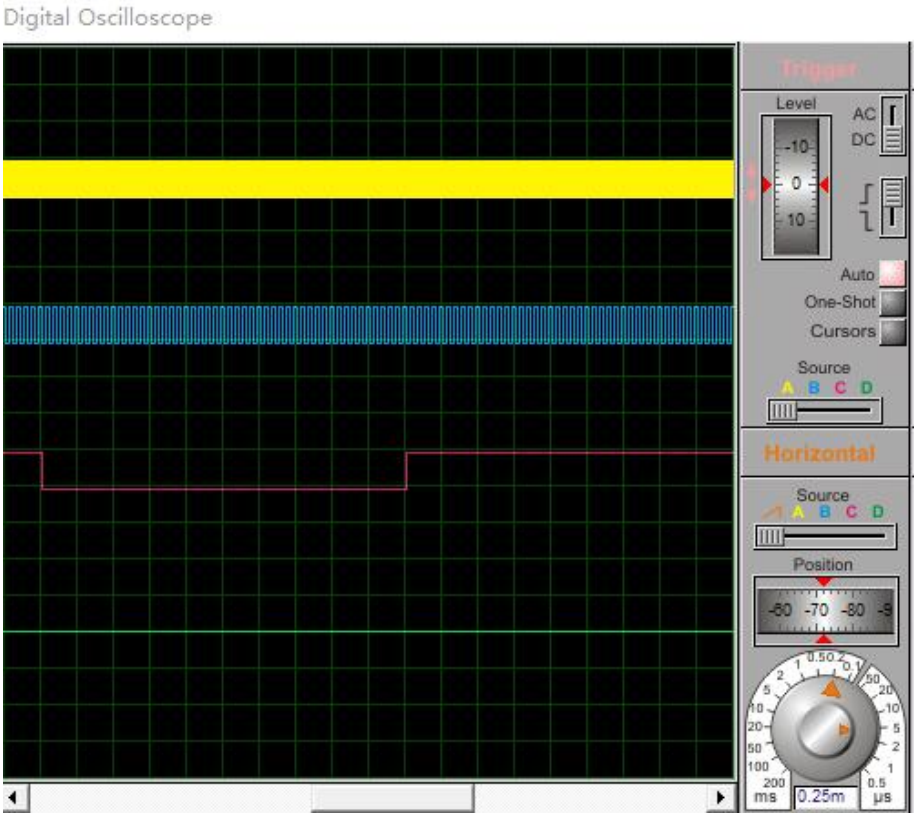

图 4 8253A 波形

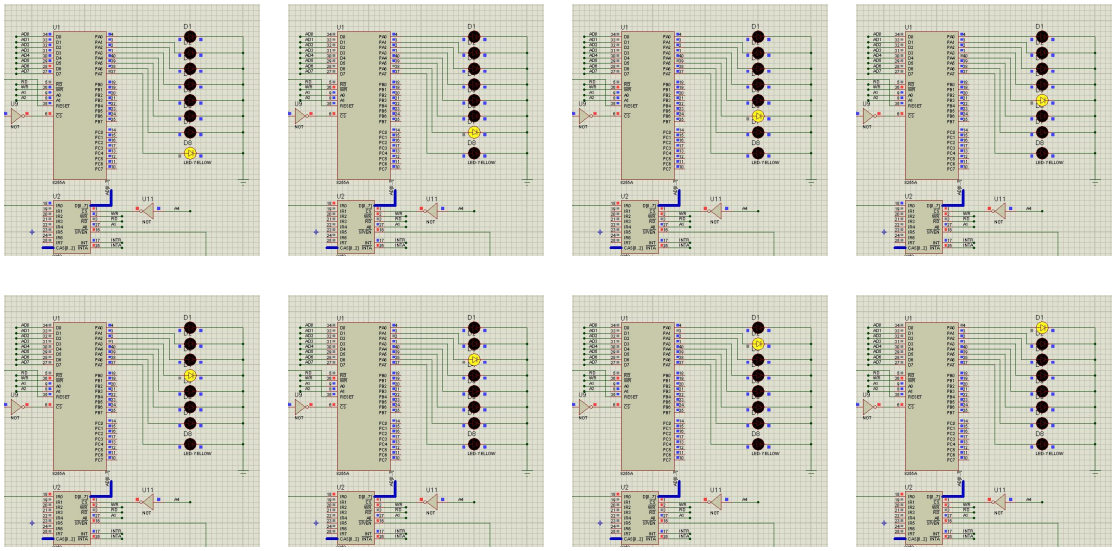

表 5 流水灯效果

五、分析体会

本次课程设计需要软件代码和硬件电路的具体结合，是对教材上知识原理的实际应用，在操作的过程中加深了对所学知识点的理解。不同于学习汇编时只需要程序没有错误正确执行，课程设计中每一个错误信息都需要通过代码和电路联动共同排查。

教材以及参考书上的例子与习题中一般会给出所有端口地址或其他需要用到的控制字条件，但在实际设计中需要自己按照接线规则进行电路布线并由此获得实际端口地址，所以不同的设计理念得到的端口地址各不相同。而各个接口芯片的命令字也会因设计者设计需求的不同而不同，这就要求对控制位的设置规则足够熟练。

笔者选择的设计较为基础，但尽可能使用了所学到的三种接口芯片，并利用8253A 两计数器级联完成了初值 10000 的计数，这为后续解决大数初值计时（或计数）的情景奠定了实际操作基础。