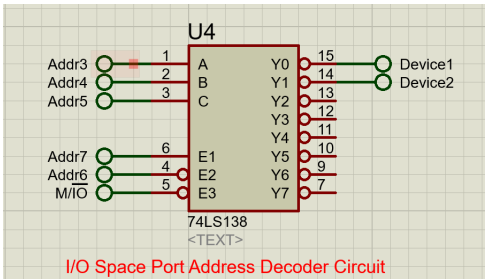


# 8086 实验报告

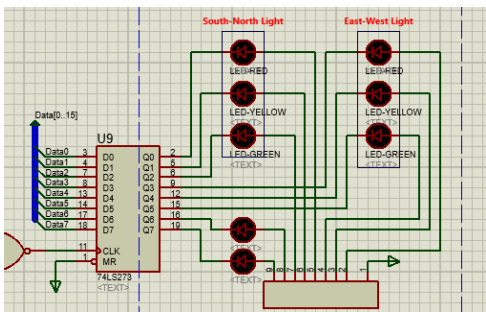
张浩宇 522031910129

## 1 实验一 I/O 译码

### 1.1 读入开关状态并输出



(a)



(b)

图 1: (a)IO 译码电路. (b) 交通灯 LED 连接

输入口（74LS244）和输出口（74LS273）的片选端口分别为 Device1 和 Device2，通过 74LS138 进行 IO 地址译码，如图1(a)。当 Device1 为低电平，选中 74LS244，即为输入；当 Device2 为低电平，选中 74LS273，即为输出。根据译码器连接方式，输入端口地址 80H，输出端口地址 88H。

因此从输入端口读入开关状态，取反输出到输出口，即可实现开关状态的读入和输出。

### 1.2 模拟交通灯

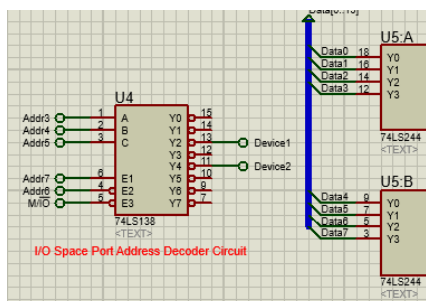
如图1(b)，交通灯为 6 个分为 2 组的共阳极 LED，其另一端接在 74LS273 输出口，当对应位输出低电平时，LED 亮，则每一种亮灯状态对应一个 8 位编码，将该编码从 74LS273 的端口地址输出就可以按照对应状态点亮 LED。

代码的运行是很快的，为了让亮灯的效果保持以便看清，需在改变 LED 状态后进行延时操作以保留。延时等待的效果可以通过循环重复多次操作实现。由于程序中需多次进行延时，可以将延时程序写为一个子过程，在需要时调用。

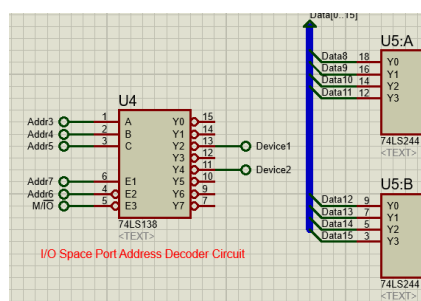
该程序可以看作一个状态机，每个状态对应一段包括输出 LED 编码、延时、状态跳转的操作。从初始状态进入后，在不同状态之间循环，实现交通灯的模拟。

要实现灯闪烁的效果可以通过在 LED 的亮和灭的状态中快速切换实现，多次切换的过程可以通过循环语句完成。在切换中也需要短时间的延时才能看清变化。

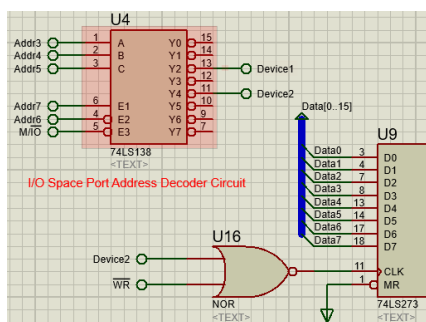
### 1.3 修改选片地址



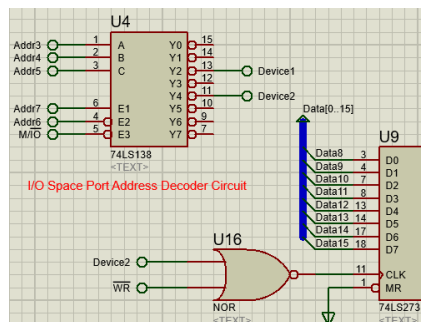
(a) 74LS244 地址 90、92、94、96



(b) 74LS244 地址 91、93、95、97



(c) 74LS273 地址 A0、A2、A4、A6



(d) 74LS273 地址 A1、A3、A5、A7

图 2: 修改选片地址

在该电路中，74LS244 和 74LS273 的片选端口分别为 Device1 和 Device2，这两个端口连接在译码器的输出端，而地址总线 Addr7-Addr3 连接在译码器的输入端，输入地址决定译码器的哪个输出口为低电平从而使能 I/O 端口芯片，因此改变 Device1 和 Device2 的在译码器上的位置就可以改变对应的片选地址的 Addr7-Addr3。而 IO 芯片使用低 8 位数据线时，对应偶地址，Addr0=0，使用高 8 位数据线时，对应奇地址，Addr0=1。Addr2-Addr1 与 IO 选片地址无关。

对于 74LS244，要将其地址改为 90H-97H，其中 90H、92H、94H、96H 均为偶地址且译码器 Y2 输出 0，有相同的连接方式，如图2(a)；91H、93H、95H、97H 均为奇地址且译码器 Y2 输出 0，有相同的连接方式，如图2(b)。

对于 74LS273，要将其地址改为 A0H-A7H，其中 A0、A2、A4、A6 均为偶地址且译码器 Y4 输出 0，有相同的连接方式，如图2(c)；A1、A3、A5、A7 均为奇地址且译码器 Y4 输出 0，有相同的连接方式，如图2(d)。

要在修改选片地址后完成开关状态的写入和输出，只需将源代码中的输入和输出端口地址修改为新的地址即可。

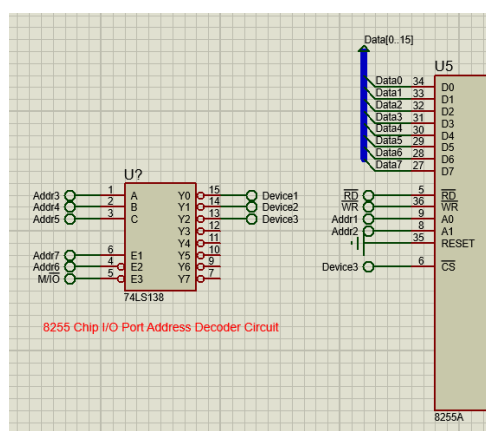
## 1.4 实验中的问题与解决

在编写延时程序时，通过一层 LOOP 循环实现延时，但是在实际运行中发现延时时间不够长，LED 状态切换过快。在一层 LOOP 循环中在添加一层由跳转语句控制的第二层循环后，延时时间变长，效果符合预期。

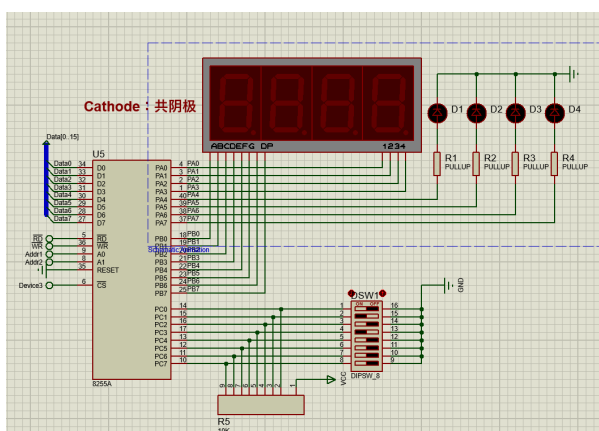
# 2 内存扩展和 I/O 空间操作

## 2.1 8255 接口操作

### 2.1.1 8255 在 I/O 空间的地址



(a)



(b)

图 3: (a)8255 地址译码. (b)8255IO 连接

8255 的  $\overline{CS}$  引脚为低电平时 8255 才工作，其连接在译码器 Y2，由地址 Addr7-Addr3 控制，如图3(a)。要选片 8255，Addr7-Addr3 应为 10010B。8255 的 A、B、C 三个口和控制寄存器的选择由 A1、A0 决定，分别由 Addr2、Addr1 控制。8255 使用低 8 为数据线，对应偶地址，Addr0 为 0。因此 8255 的地址分别为：

A(A1A0=00):10010000B=90H

B(A1A0=01):10010010B=92H

C(A1A0=10):10010100B=94H

控制寄存器 (A1A0=11):10010110B=96H

### 2.1.2 接收输入并通过数码管和 LED 输出

根据 8255 连接方式，如图3(b)，A 口高 4 位连接 LED，A 口低 4 位连接数码管位选，B 口连接数码管段选，C 口连接开关输入。故应该将 8255 设置为方式 0，A 口为输出，B 口为输出，C 口为输入，对应控制字 10001001B。

将控制字输出至 8255 的控制寄存器完成 8255 的初始化，然后通过 C 口读入开关量，取其高 4 位并将低 4 位变为 1110B 输出至 A 口实现 LED 显示和数码管位选，根据低 4 位输出段码至 B 口实现数码管段选。在输出段码时，可预先将 1-F 的段码存储在数据段，输出时根据相应的值寻址并取出。

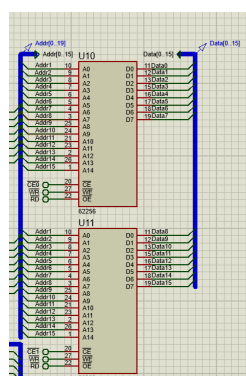
### 2.1.3 接收输入并通过多位数码管输出

该实验 8255 连接和初始化配置与上个实验相同。由于需要数码管多位显示不同的数字，而数码管同时只能接收一个段码的输入，因此需要在数码管的显示周期内快速切换显示的段码和位选，以实现多位显示。

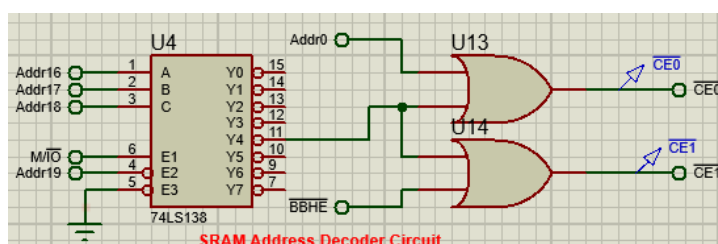
### 2.1.4 实验中的问题与解决

- (1) 在进行 I/O 操作时，由于输入和输出只能通过 AX/AL 寄存器进行，直接操作输入量会被覆盖。应当在输入后保存输入量以便进行多次操作。
- (2) 在该实验中需要对一个寄存器进行移位 SHR 右移 4 位，但直接 SHR AL,4 会报错，因为用立即数来移位只能移动 1 位，移动多位应当先在一个寄存器中保存移位值，再进行移位操作。
- (3) 在通过快速切换段码和位选来实现数码管多位显示不同数字时，会出现显示乱码的情况。这是因为切换过快导致重叠，因此应该在切换之间加上短时间的延迟。

## 2.2 内存扩展



(a)



(b)

图 4: (a) 内存芯片. (b) 内存地址译码

### 2.2.1 内存地址和大小

根据原理图中内存芯片部分，如图4(a)，内存由两片 32KB 的芯片组成，为 U10 和 U11，分别对应偶地址和奇地址。总的内存容量为 64KB。根据地址译码电路，如图4(b)，当 Addr19-Addr16 为 0100B 时，Y4 为 0，能选中内存芯片，Addr15-Addr1 为片内地址，Addr0 在 U10 和 U11 中选择，决定奇偶地址，故地址范围是 40000H-4FFFFH。

### 2.2.2 内存写入

已知 U10 和 U11 内存范围是 40000H-4FFFFH，可将 4000H 作为段基址，就是对 U10 和 U11 进行操作。

要操作内存，可以通过寄存器相对寻址的方法，将 BX 寄存器作为偏移地址，AX 寄存器存储要放入内存的数值，通过循环，每次放入内存并增加寄存器的值，实现对内存的遍历操作。

当操作完所有内存后，需要结束循环，由于偏移地址寄存器（16bit）的可表示的内存单元数与总的内存容量一致（64K），当出现进位时恰操作完，因此进位标志可以作为循环终止的判断条件。

### 2.2.3 修改内存地址

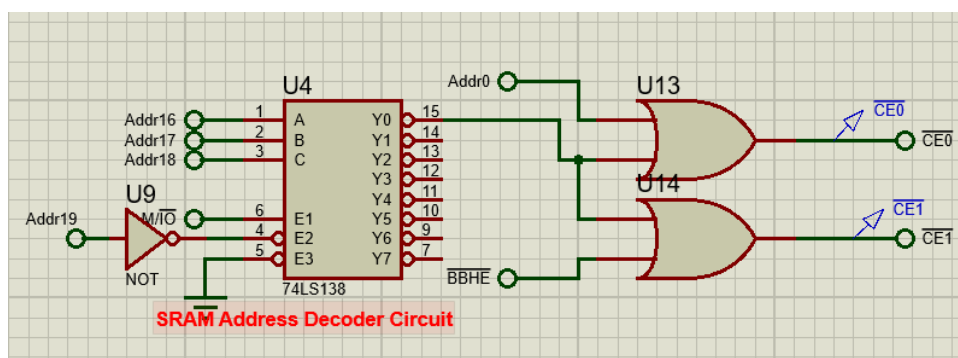


图 5: 修改内存起始地址

要使内存地址从 80000H 开始，可修改译码器输出的连接位置，如图5。内存地址从 80000H 开始，则 Addr19-Addr16 应为 1000B，此时译码器 Y0 输出低电平，应将 Y0 与后续电路连接。

修改内存地址后的写入程序与之前完全一致，只需将段基址改为 8000H 即可。

### 2.2.4 实验中的问题与解决

(1) 在该实验中，需要修改段寄存器 DS 的值，而段寄存器只能通过寄存器寻址方式赋值，不能用立即数寻址。

(2) 8086 在操作内存时，默认是以字为单位进行操作，因此在本实验需要在代码中指明内存操作数的宽度属性是字节。

## 3 定时器、计数器的应用

### 3.1 8255 配置和应用

#### 3.1.1 8255 初始化

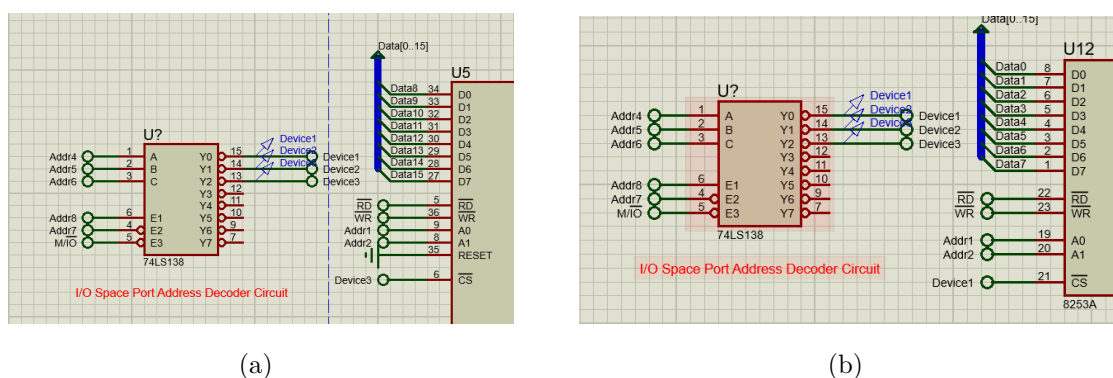


图 6: (a)8255 地址译码. (b)8253 地址译码

8255 的  $\overline{CS}$  引脚为低电平时 8255 才工作，其连接在译码器 Y2，由地址 Addr8-Addr4 控制，如图6(a)。要选片 8255，Addr8-Addr4 应为 10010B。8255 的 A、B、C 三个口和控制寄存器的选择由 A1、A0 决定，分别由 Addr2、Addr1 控制。8255 使用高 8 位数据线，对应奇地址，Addr0 为 1。因此 8255 的地址分别为：

A(A1A0=00):100100001B=121H

B(A1A0=01):100100011B=123H

C(A1A0=10):100100101B=125H

控制寄存器 (A1A0=11):100100111B=127H

在本实验中，A 口输出，B 口输出，C 口高 4 位输出、低 4 位输入，则控制字 10000001B，将控制字写入控制寄存器即完成初始化。

#### 3.1.2 数码管输出学号

在数码管上同时多为显示不同数字的方法与之前相同，需要快速切换段码和位选，以实现多位显示。在切换之间需要延迟以免重叠。

#### 3.1.3 PC0 开关量输出至 PC6

要使 PC0 开关量输出到 PC6，可以将 C 口输入数据后，根据 PC0 的值设置 PC6 的值，再输出至 C 口。

在判断 PC0 的值时, 可以通过 AND 指令与 00000001B 进行与操作, 若结果为 0, 则 PC0 为 0, 否则为 1。

在改变 PC6 的值时, 为了不影响 C 口其他引脚的值, 可以通过 OR 指令与 01000000B 进行或操作来令其为 1, 通过 AND 指令与 10111111B 进行与操作令其为 0, 不会影响其他位。

### 3.1.4 8253 配置和应用

### 3.1.5 8253 初始化并输出 10ms 方波

8253 的  $\overline{CS}$  引脚为低电平时 8253 才工作, 其连接在译码器 Y0, 由地址 Addr8-Addr4 控制, 如图6(b)。要选片 8253, Addr8-Addr4 应为 10000B。8253 的 0, 1, 2 三个计数器的选择由 A1, A0 决定, 分别由 Addr2, Addr1 控制。8253 使用高 8 位数据线, 对应偶地址, Addr0 为 0。因此 8253 的地址分别为:

Timer0(A1A0=00):100000000B=100H

Timer1(A1A0=01):100000010B=102H

Timer2(A1A0=10):100000100B=104H

控制寄存器 (A1A0=11):100000110B=106H

要让 OUT0 输出 10ms 方波, 应采用 8253 的工作方式 3。输入时钟周期为 1 $\mu$ s, 要输出 10ms 方波, 则需要计数 10000 次, 应该设置初值为 10000, 在二进制计数下为 2710H, 需要 16 位。故 Timer0 的控制字为 00110110B。初始化时需要将初值放入其初值寄存器

当 PC0 上的开关拨到 OFF, PC0 为 1, PC6 为 1, 即 Timer0 的 GATE0 为 1, 能够工作, 输出方波; 当 PC0 上的开关拨到 ON, PC0 为 0, PC6 为 0, 即 Timer0 的 GATE0 为 0, 不能工作, 不输出方波。

### 3.1.6 级联输出方波

要让 OUT1 输出 1s 方波, 应采用方式 3。将 OUT0 级联到 Timer1 的时钟输入, 周期为 10ms, 则需要计数 100 次, 应该设置初值为 100, 在二进制计数下为 64H, 需要 8 位。故 Timer1 的控制字为 01010110B。初始化时需要将初值放入其初值寄存器。

当 PC0 上的开关拨到 OFF, PC0 为 1, PC6 为 1, 即 Timer0 的 GATE0 为 1, 能够工作, 输出方波, Timer1 有正常的时钟输入也能工作输出方波当 PC0 上的开关拨到 ON, PC0 为 0, PC6 为 0, 即 Timer0 的 GATE0 为 0, 不能工作, 不输出方波, 此时 Timer1 没有时钟输入, 不能工作, 不输出方波。

### 3.1.7 实验中的问题与解决

在对 8253 计数器的初值寄存器赋值时, 一次只能操作一个字节, 因此需要分别对高 8 位和低 8 位进行赋值。