

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

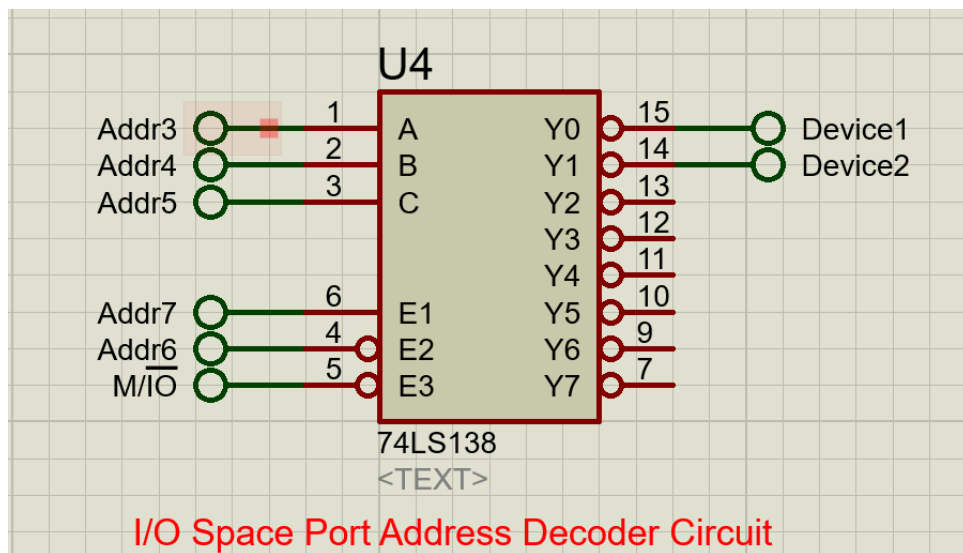


图 1: I/O 译码电路

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。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，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

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，而地址总线连接在译码器的输入端，输入地址决定译码器的哪个输出为低电平从而使能 I/O 端口芯片。因此改变 Device1 和 Device2 在译码器上的位置就可以改变对应的片选地址。要改为，只需把，如图。

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

8255 的 \overline{CS} 为引脚，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。要选片 8255，Addr7-Addr3 应为 10010。此外 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，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

2.2.1 内存地址和大小

根据原理图中内存地址译码部分和内存芯片部分，如图，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，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 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