

计算机科学系实验报告（首页）

课程名称 微型计算机技术

班级 17 软 2 班

实验名称 I/O 地址译码

指导教师 唐明星

姓名 陈庆辉 学号 陈庆辉

组号 01 日期 2020 年 5 月 18 日

一、实验内容

当 CPU 执行 I/O 指令且地址在 280H~2BFH 范围内，译码器选中，必有一根译码线输出负脉冲。利用这个负脉冲控制 L7 闪烁发光（亮、灭、亮、灭、……），时间间隔通过软件延时实现。

二、实验分析

1. 预备知识

与非门：输入全为高电平时才输出低电平。

38 译码器：使能端 E3 为高，#E1、#E2 为低电平有效。ABC 三个输入端选择 Y0~Y7 某端口输出有效低电平。

74LS74D 触发器：CLK 端口上升沿有效时，D 端输入，Q 端输出同 D 端，#Q 端跟 Q 端相反。另#SD 端口置位、低电平有效，#CD 端口复位、低电平有效。

真值表：

输入		输出	
D	CLK	Q	#Q
0	↑	0	1
1	↑	1	0
x	0	Q	#Q

2. 程序分析

```
code        segment
assume cs:   code
start: mov   dx,2a0h      ;输入地址 2a0h
          out  dx,al       ;执行写指令，将 al 数据写到 2a0h，执行一个总线写周期 1。
          call delay       ;调延时子程序
          mov  dx,2a8h      ;输入地址 2a8h
          out  dx,al       ;执行写指令，将 al 数据写到 2a8h，执行一个总线写周期 2。
          call delay       ;调延时子程序
          mov  ah,1        ;将立即数 1 传送给 ah
```

int 16h ;执行中断指令，根据中断类型为 16h，转到键盘 I/O 中断调用的中断服务子程序。该子程序根据入口参数 ah 为 1。转到判断是否有键按下的子模块，若有按键操作，则 ZF=0；若无键按下，则标志位 ZF=1。

je start ;je 条件转移指令，看 ZF 标志，ZF 为 1 跳转，即无键按下转 start；ZF 为 0 不跳转，即按键继续往下执行。

mov ah,4ch

int 21h ;执行中断指令，根据中断类型为 21h，转到 Dos 功能调用的中断服务子程序。该子程序根据入口参数 ah 为 4ch，转到返回 Dos 的子模块，然后返回 dos 操作系统，程序结束。

delay proc near ;延时子程序

mov bx,5 ;外循环次数

lll: mov cx,0 ;内循环次数

ll: loop ll ;先执行 cx-1=0-1=65535，即循环 65536 次

dec bx ;bx 递减，减至 0 时，ZF 标志位为 1；否则为 0

jne lll ;jne 条件转移指令，若 ZF 为 0，跳转；ZF 为 1 不跳转

ret ;返回调用它的程序

delay endp

code ends

end start

3. 实验内容



三. 实验内容、程序清单

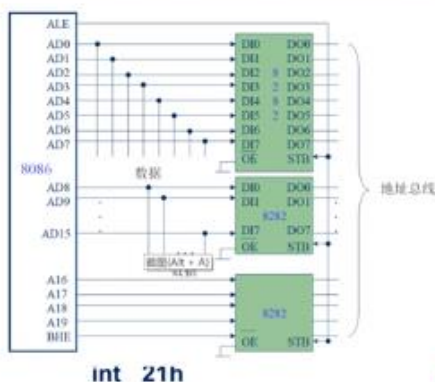
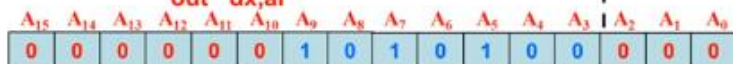
2. 程序清单

code segment

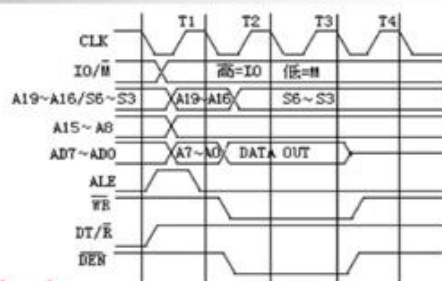
assume cs: code

start: mov dx,2a0h

out dx,al



int 21h



①

②

总线写周期 1:

T1 时刻，产生由 DX 寄存器产生的 16 位有效地址 02a0h，通过 AD0~AD15 输出到地址锁存器的输入端。

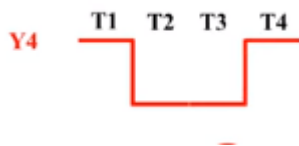
同时 ALE 引脚产生有效的高电平地址锁存允许信号输出到地址锁存器的 STB 引脚。当 T1 结束时，ALE

变为低电平，产生一个下降沿。地址锁存器开始工作，将输入端提供的地址信号锁存到输出端的地址总线上。地址总线连接到实验箱上的 I/O 译码电路上，提供地址信号 02a0h。现在分析电路图，要使 38 译码器工作，使能端#E2、#E3 要为低电平。#E2 接地为低电平，#E3 连接的是多路与非门的输出端。我们知道，与非门是全高为低。根据电路图，多路与非门 1~11 输入端都为高电平，12 输入端连接的两路与非门 U2:A 的输入端连接的是 I/O 读控制信号引脚（在写周期始终为无效的高电平）和写控制信号引脚（T1 时刻为高电平），所以多路与非门输出高电平，38 译码器也因此不工作。因为 A5~A3 为 100，所以 3-8 译码器的输出端 Y4 输出无效的高电平。

T2 时刻，I/O 写控制信号为低电平，因此多路译码器的全部输入都为高电平，输出低电平。38 译码器开始工作，Y4 输出有效的低电平。

T3 时刻，I/O 写控制信号仍为低电平，因此 38 译码器的 Y4 仍输出有效的低电平。

T4 时刻，I/O 写控制信号为高电平，38 译码器的 Y4 输出高电平。



因为 T3 到 T4 产生一个上升沿，D 触发器开始工作，Q 端口输出高电平，LED 发光二极管亮。



三. 实验内容、程序清单

2. 程序清单

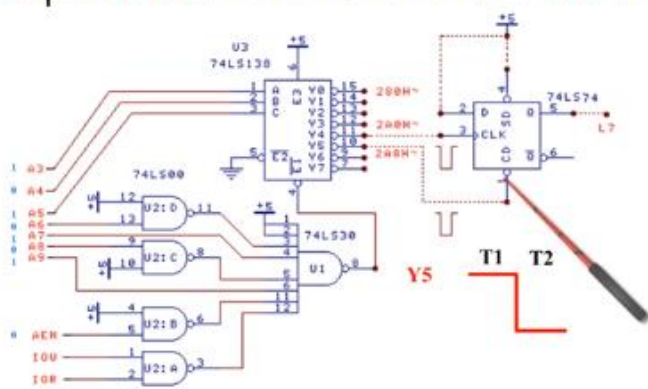
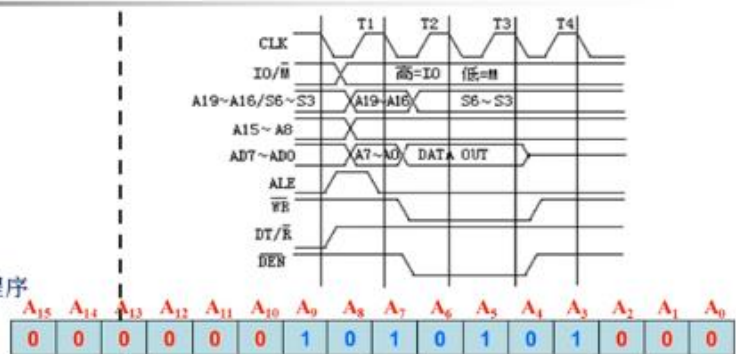
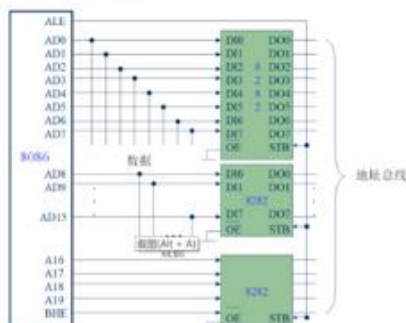
```
code segment
assume cs: code
start: mov dx,2a0h

out dx,al

call delay ;调延时子程序

mov dx,2a8h

out dx,al
```



总线写周期 2 与总线写周期 1 过程类似, 主要区别是地址信号变为 2a8h, 38 译码器输出 Y5, 连接 D 触发器的#CD 端(复位)。当 Y5 输出低电平时, #CD 端低电平有效, 将 Q 端复位清 0, 灯灭。

延时子程序的关键点

jne 跟 0 标志位有关, 0 标志位为 0 时跳转, 前面的 dec bx 指令会对 0 标志位有影响。循环 5 次后, 0 标志位为 1, jne 不跳出, 继续执行。

2.1.3 标号

标号是指令地址的符号表示, 也可以是过程名。过程名是过程入口地址的符号表示, 即过程的第一条指令的地址。

标号的定义方式有两种:

(1) 用“:”定义, 直接写在指令助记符前。如定义标号 CYCLE:

```
CYCLE: MOV AL,[SI]
```

(2) 用 PROC 和 ENDP 伪指令定义过程。

标号一般只在代码段中定义和使用, 标号代表指令的地址, 因而它也有 3 个属性: 段地址、偏移地址和类型。

标号的段地址是定义该标号所在段的段地址。

标号的偏移地址是标号所在段的段首址到该标号定义指令的字节距离。

标号的类型有 NEAR 和 FAR 两种。用“:”定义的标号为 NEAR 类型, 过程名可以定义为 NEAR 类型也可以定义为 FAR 类型, NEAR 类型的标号只能在定义该标号的段内使用, 而 FAR 类型的标号却无此限制。

过程用伪指令 PROC 和 ENDP 定义, 过程定义的格式请见 2.4.1 节。

标号同变量一样, 也可以和常量组成地址表达式。

2.4.1 过程的定义

过程由伪指令 PROC 和 ENDP 定义。定义过程的格式为:

过程名 PROC [NEAR] 或 FAR

:

过程名 ENDP

其中, PROC 和 ENDP 必须成对出现, 且前面都有同一过程名(过程名是必须有的), 过程的类型由 PROC 的操作数指出。若 PROC 后无操作数, 则默认为 NEAR 类型。

在汇编程序中, 可以使用过程定义伪指令定义子程序, 通过调用指令 CALL 调用子程序。过程名是为该子程序起的名字, 在调用子程序时, 作为调用指令 CALL 的操作数。PROC 和 ENDP 两伪指令之间, 是为实现某功能的程序段, 其中至少有一条子程序返回指令 RET 以便返回调用它的程序。

子程序也可以用“:”定义, “:”定义的标号是为该子程序起的名字。主程序把用“:”定义子程序作为远过程来调用。若子程序与调用它的主程序在同一代码段, 则必须用过程来定义该子程序。中断调用都是远调用, 所以中断服务子程序可以用“:”定义。若用过程来定义中断服务子程序, 则必须将中断服务子程序定义为远过程, 无论它与调用程序是否在不同的代码段。