

微机原理与接口技术

第八至十一章复习

第八章、中断系统

- 中断的基本概念
- 中断系统的组成及结构
- 中断分类，中断类型号（中断向量码）、中断向量表
- 中断服务程序基本概念及他们之间的联系
- 中断向量设置及中断处理程序组成
- 可屏蔽中断的响应过程
- 8259中断控制器功能、基本结构，8259中ISR、IRR、IMR的作用，初始化控制字ICW2的作用，操作控制字OCW1的作用
- 优先级方式、嵌套方式
- 8259级连工作方式以及连接

1. PC 机的中断系统有哪些主要组成部分？

1. 硬件部分：

PC 机中断机制的硬件结构如图 8-1 所示，它们可以分为以下三个组成部分：

- (1) CPU 内部中断逻辑。包括：内部中断源，外部中断信号引入脚，内部中断的仲裁逻辑。
- (2) 外部非屏蔽中断源控制逻辑。
- (3) 外部可屏蔽中断源管理控制器 8259。

2. 软件部分：(编程可设定部分)

- (1) 中断向量表：中断处理程序入口地址表
- (2) 中断相关的专属指令：产生软中断的指令（INT n；INTO）；开/关外部可屏蔽中断指令 STI/CLI；中断返回指令 IRET。
- (3) 主程序中为处理外部可屏蔽中断所需的对中断控制器 8259 设定的程序段以及用于设置中断向量表的程序段。
- (4) 中断处理程序。

2. PC 机的中断系统有哪些中断源？

1. 来自 CPU 内部的中断源（软件中断，由指令执行时的异常结果触发）：

- 除法错误（除数为零；商超出表示范围）
- 溢出（运算产生溢出使 OF=“1”且又执行了 INTO 指令）
- 指令单步（也称为自陷或陷阱）执行设置（TF=“1”），。
- 执行中断指令 INT n

2. 来自 CPU 外部的中断源（硬件中断，由 CPU 专用引脚上的电信号触发）：

- 非屏蔽中断引脚 NMI 引入的外部不可屏蔽中断请求源。

该引脚引入的中断请求用于处理一些外部的紧急事件，一旦发生了这类事件的中断请求，CPU 必须给与响应。例如 PC 机中引入的 RAM 奇偶校验错；I/O 通道校验错；数学协处理器错等中断请求。

- 可屏蔽中断引脚 INT 引入的外部可屏蔽中断请求源。

这类中断请求 CPU 是有条件的加以响应，响应条件为标志寄存器中的中断允许位 IF=“1”。用户可以通过 STI/CLI 指令来设定/清除 IF 位，从而决定是否响应该引脚引入的中断请求。

3. 请说明中断向量表的结构及作用。

中断系统的一个重要职能是根据发出请求的不同中断源向 CPU 提供不同的中断处理程序入口地址。在 PC 微机中采用的方法是首先为不同的中断源分配一个唯一的编号，称之为中断类型号或中断向量号。然后将各中断处理程序的入口地址按中断类型号的顺序编排成一个称之为中断向量表的地址表。这个表之所以称为中断向量表，大概是由于表中的每个最终用于合成 20 位物理地址（向量）的表项都是由一个 16 位段地址（分量）和一个 16 位偏移地址（分量）组成的缘故。中断向量表的结构示意图见图 8-2。

1. 中断向量表的结构

中断向量表实际上是所有中断处理程序的入口地址表。每个中断处理程序的入口地址也称为中断向量，为 4 个字节，是由一个 16 位的代码段值（CS）和一个 16 位的偏移地址值（IP）组成。PC 机共可设置 256 个中断向量（对应 256 个中断处理程序），依次编号为中断类型（向量）号 0~255，共占用 $256 \times 4 = 1024$ （1K）个字节的地址空间。PC 机的中断向量表见后表。其中的部分中断类型号已被 PC 机固定用于某些专用的中断处理，用户不可使用。

4. 采用 INT **n** 指令和由 CPU 的 INT 引脚引发的中断过程有什么不同的地方？

- INT **n**：来自CPU内部的中断源（软件中断，由指令执行时的异常结果触发），不可以被可屏蔽中断控制器8259屏蔽。
- **INT**：可屏蔽中断引脚INT引入的外部可屏蔽中断请求源。
- 这类中断请求CPU是有条件的加以响应，响应条件为标志寄存器中的中断允许位IF=“1”。
- 用户可以通过STI/CLI指令来设定/清除IF位，从而决定是否响应该引脚引入的中断请求。

5. 8259 中的中断请求寄存器 **IRR** 和中断服务寄存器 **ISR** 各自的作用是什么？

中断请求寄存器 (**IRR**): 8 位的锁存器, 只允许读。用于锁存外部 8 个中断请求信号。对应 **IR0~IR7**, 有请求的位为 “1”。

中断服务寄存器 (**ISR**): 一个 8 位的寄存器, 只允许读。用于记录 CPU 正在响应的中断请求位。**IR0 ~ IR7** 中哪个正被 CPU 响应, **ISR** 中对应位就为 “1”。

8. 8259 都有哪些中断优先级方式；它们各自的特点是什么？

- **普通全嵌套方式** (只能嵌套高优先级的中断)

8 个中断请求源 IR7 ~ IR0 的优先级是固定的，IR0 的优先级最高 IR7 最低。且某中断响应过程中只允许 8259 转发比当前优先级高的中断请求（只允许高优先级嵌套）。此种方式用于单片 8259 或级联方式下的从片 8259 中。

- **特殊全嵌套方式** (可以嵌套等优先级或高优先级的中断)

8 个中断请求源 IR7 ~ IR0 的优先级是固定的，IR0 的优先级最高 IR7 最低。且在中断响应过程中允许 8259 转发高于或等于当前优先级的中断请求（允许高优先级或等优先级嵌套）。

此种方式用于级联方式下的主片 8259 中。（因为从片 8259 的 8 个中断请求都要经主 8259 的同一中断请求输入端送入主 8259。对于主 8259 来说，这个中断请求输入端的优先级都是相同的，尽管从片 8259 各中断请求输入端的优先级不同。所以要想使从片 8259 的高优先级中断嵌套入本片的低优先级中断，就需要主片提供嵌套同级优先级中断请求的能力）。

- **优先级自动循环方式** (起始优先级固定)

这是一种优先级可变方式。初始状态下 IR0 优先级最高；IR1 次之，IR7 优先级最低。当 IR0 被响应后，其优先级自动降为最低，IR1 优先级成为最高。其它的优先级都自动升高一级。从而形成一优先级的可变循环体。

- **优先级特殊循环方式** (起始优先级可设定)

这是一种初始优先级可设定的优先级自动循环方式。并且通过对 OCW2 中的 L2、L1、L0 三位编程来设定初始状态下的最低优先级。

例如将 L2、L1、L0 设定为 011，则初始状态的最低优先级为 IR3，最高优先级为 IR4。同样，当最高优先级的中断被响应后自身的优先级降为最低，其它优先级自动增高一级。

9. 什么是 8259 的中断结束方式；这些不同的中断结束方式其目的各是什么？

● 自动中断结束方式

这种方式下，当 CPU 响应中断后，8259 会在 CPU 回送的第二个 INTA 负脉冲作用下自动将 ISR 中的当前响应对应位清零（在 CPU 开始运行中断处理程序之前），不需要程序员在中断处理程序内添写将 ISR 对应位清零的指令段。

这种方式使得在中断处理程序中只要开放了中断允许（IF=1）位，马上就可以响应对等优先级和低优先级中断请求的嵌套，而且嵌套点可以在之后随时产生，是无法预测的。这就使得中断处理程序可能会受到一些与本处理无关的中断请求的干扰，甚至影响到本次中断处理的正常完成。所以这种方式通常用于单级 8259 且不会产生其它中断嵌套的应用系统中。

● 一般（命令字）中断结束方式

这种方式的主要特点是在中断处理子程序内需要开放等/低优先级中断的地方插入使 8259 内 ISR 当前响应位清零的指令。具体操作是插入设置 OCW2 寄存器 EOI=1；SL=0；R=0 的程序段。执行这段程序就会使 8259 ISR 内的当前响应位，即序号最低为“1”的位清零（固定优先级方式下 ISR 为“1”的位中序号最低的位优先级最高，也就是当前响应位）。

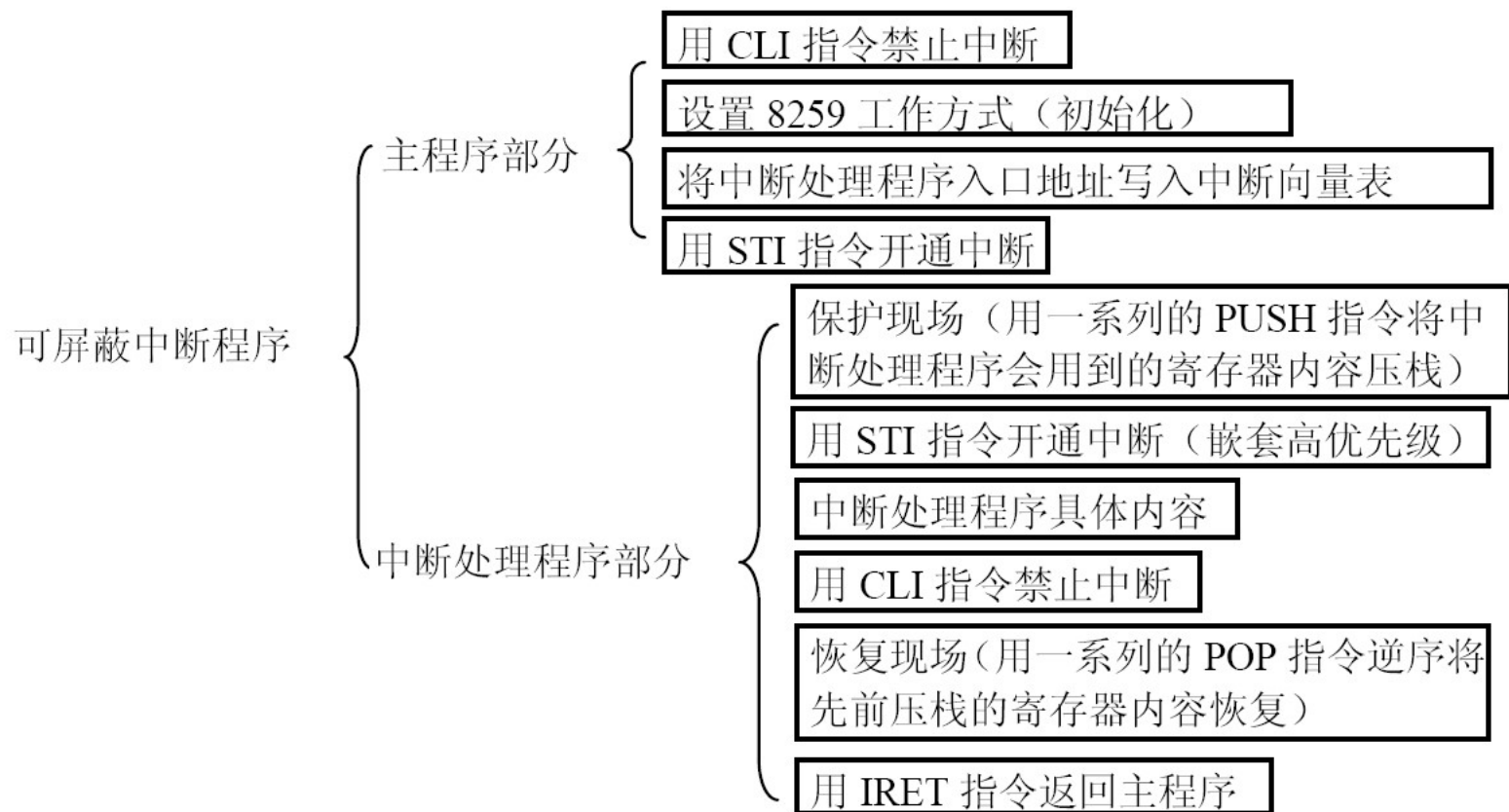
这种方式是大多数可屏蔽中断采用的方式，而且通常用于多个 8259 级联且采用固定优先级全嵌套优先方式下。

- **特殊（命令字）中断结束方式**

这是一种用于多个 8259 级联且不采用固定优先级全嵌套优先方式的中断结束方式。因为这种情况下由于最高优先级是变化的，ISR 内序号最低的为“1”的位不一定是当前响应位，8259 无法自行判定那位是当前响应位。因此需要用户在程序中指定当前响应位。即在中断处理程序内除了向 8259 内 OCW2 寄存器写入 EOI=1；SL=1；R=0 外还要向 L2；L1；L0 位写入当前响应中断的编号（000B~111B，对应 IR0~IR7），指明将要清零的位。

说明：无论是采用一般的中断结束方式还是特殊的中断结束方式，如果是多片 8259 级联结构，中断结束命令字主片和从片都要写入。且需注意它们的地址不同。

11. 可屏蔽中断程序都有哪些必要组成部分？

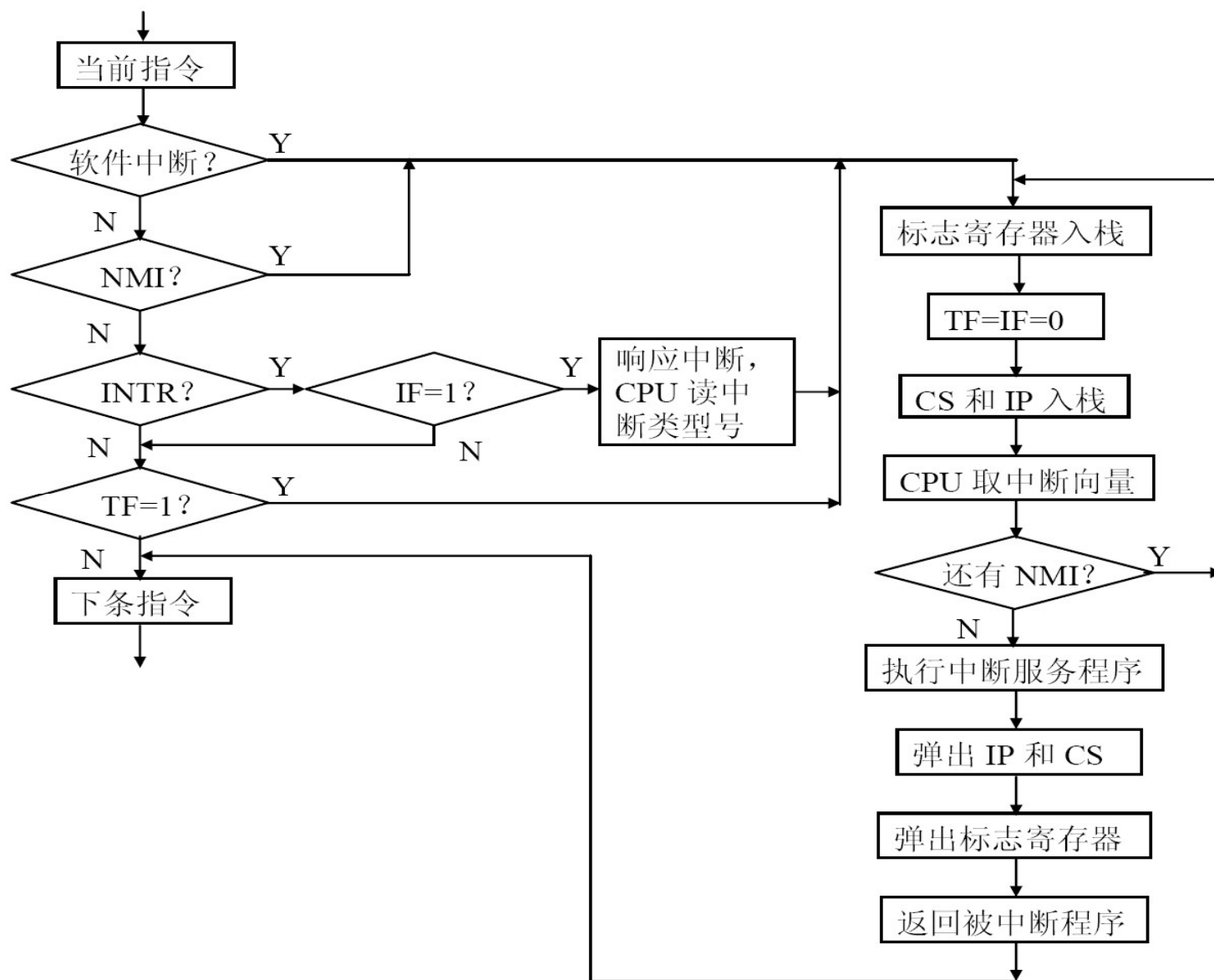


12. 为什么在 PUSH 和 POP 指令前必须要禁止中断？

(1) 如果程序中所用到的寄存器，其内容在中断返回后可能是有用的，则必须在程序主体内容开始之前用压栈指令 PUSH 逐一加以保存，并且在程序主体完成后采用与压栈指令相反的顺序用出栈指令 POP 逐一加以恢复。

(2) 如果在中断处理程序执行过程中需要进行中断嵌套，则必须在中断处理程序内合适的程序点插入 STI 指令打开中断，但是插入点必须在 PUSH 指令完成后，而且还必须在 POP 指令开始前插入 CLI 指令关闭中断。

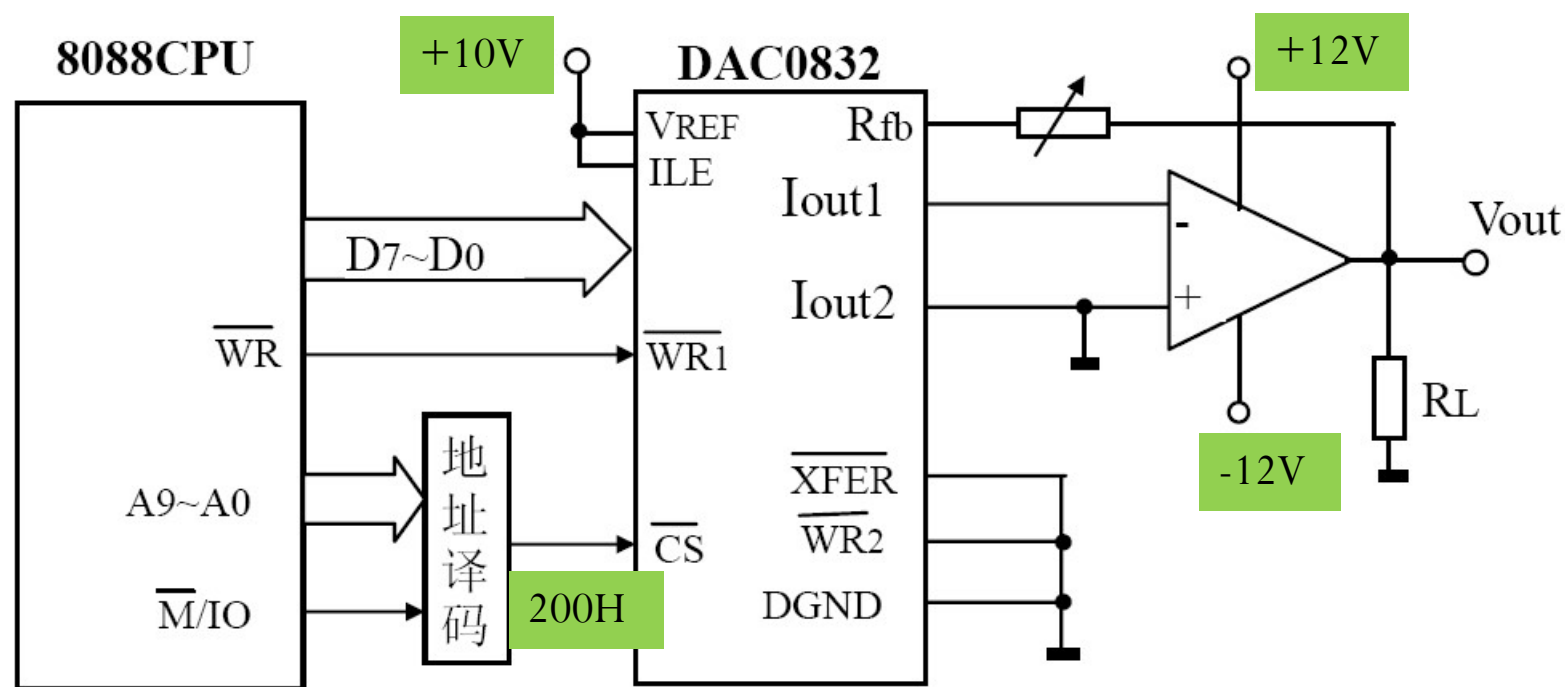
13. 请叙述 CPU 响应中断到中断返回的主要过程；并指明 CPU 将自动完成哪些工作。



第九章、A/D与D/A转换器与微机的接口方法

- A/D、D/A变换器的基本工作原理、主要技术参数
- DAC0832 的结构、引脚、应用电路连接及应用编程
- ADC0809的结构、引脚及工作时序、应用电路连接及应用编程
- 了解A/D转换原理，着重掌握逐次逼近式A/D转换原理，正确理解ADC参数（尤其是分辨率与精度的根本区别、转换时间）
- 如何利用0832产生特殊波形如：三角波，锯齿波。
- 对应电压与数值之间的转换

3. 设计一采用最小方式下的 8086 CPU 与 0832 构成的数/模转换电路。要求：0832 设定为一级单缓冲方式，口地址为 200H（用 74LS138 设计）；对应数字输入 00H~FFH 的模拟输出电压为 0V~10V。（假设 0832 转换时间为 50nS，8086 系统时钟为 10MHZ）画出电路图并编写输出频率为 1000HZ 三角波的有关程序。



```
AA: MOV DX,200H ; D/A 变换器口地址 (CS 端)
    MOV AL,00H;
    MOV CX,0200H ; 产生512 个锯齿波
    MOV BL 0H; 初始为上升波段
BB: OUT DX,AL;
    CALL DELAY;
    CMP BL,1H; 上升还是下降波段
    JZ DECAL;
    ADD AL,01H;
    JNZ BB;
    MOV BL 1H
    DEC AL,01H
DECAL: DEC AL,01H;
    CMP AL,00H
    JNZ BB
    MOV BL, 0H;
    JMP BB
```


4. A/D 变换器中分辨率的含义是什么？对于一个 10 位的 A/D 变换器，若电源电压为 0~10V，其分辨率是多少？

1. 分辨率：

数字输出量每增加一位所对应的模拟输入量的变化数值。对于输入模拟程相同的 A/D，其输出数字量的位数越多其分辨率越高。

例如：若输入量程同为 0 ~ 5V，则 8 位 A/D 的分辨率为 $5V/256 \approx 20mV$ ，

而 10 位 A/D 的分辨率为 $5V / 1024 \approx 5mV$

可见数字量位数越多分辨率越高。所以通常习惯用数字量位数来表示分辨率。

常用的 A/D 分辨率有 8 位、10 位、12 位、16 位等。

- $10V / 2^{10} = \mathbf{0.009765625\ V}$

5. A/D 变换器中 $V_{REF}(-)$ 和 $V_{REF}(+)$ 引脚的作用是什么？与电源电压有何关系？

1) $V_{REF}(+)$ 的值将限定输入模拟电压的最大值。

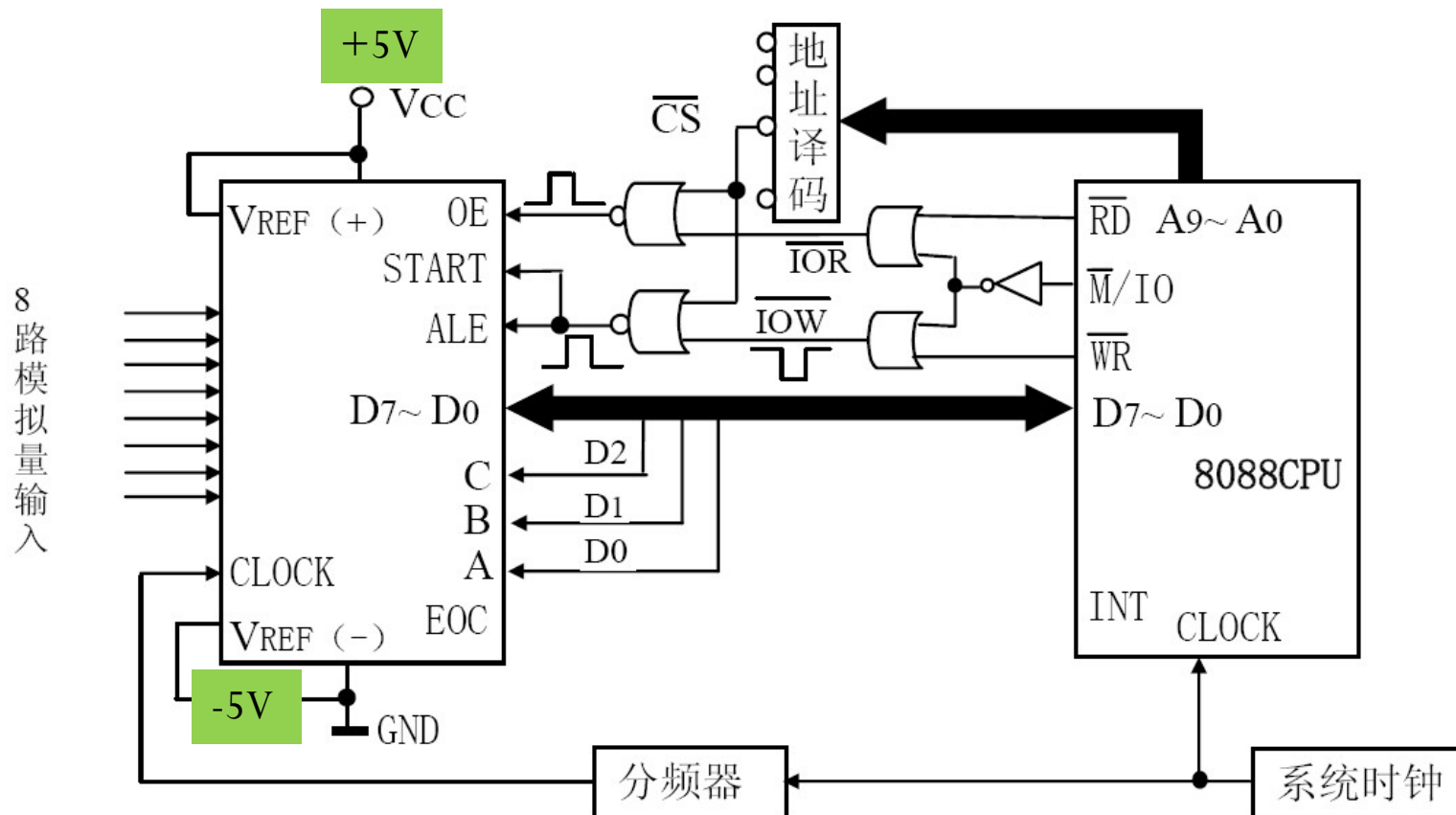
例如：若 $V_{REF}(+) = 5V$ ，则输入模拟电压的最大值不得超过 $5V$ ，且对应最大输出为 FFH 。对于输入超过 $V_{REF}(+)$ 的输入电压，也只能产生错误的结果 FFH 。

2) $V_{REF}(-)$ 的值将限定输入模拟电压的最小值。

例如：若 $V_{REF}(-) = 1V$ ，则输入模拟电压的最小值不得小于 $1V$ ，且对应最小输出为 $00H$ 。

3) 当 $V_{REF}(+)$ 和 $V_{REF}(-)$ 都不为零的情况下，输入模拟电压的可分辨下限值为 $V_{REF}(-)$ ，对应变换后数字量的最小值（8 位 A/D 为 $00H$ ）；输入模拟电压的可分辨上限值为 $V_{REF}(+)$ ，对应变换后数字量的最大值（8 位 A/D 为 FFH ）；而输入模拟电压的零值则对应 $(V_{REF}(+) - V_{REF}(-)) / 2$ 。如例 2 中 $0V$ 模拟电压的数字化值为 $80H$ ($128D$)。这种数字量的 0 值并不对应模拟量 0 值的数字编码，称为交叉二进制码。

6. 设计一采用最小方式下的 8086 CPU 与 0809 构成的模/数转换电路。已知 0809 转换时间为 $100\mu\text{S}$, 8086 系统时钟为 10MHz 。要求: CPU 采用软件延时, 口地址为 $3\text{F}0\text{H}$ (用 74LS138 设计); 对应数字输出 $00\text{H}\sim\text{FFH}$ 的模拟输入电压为 $-5\text{V}\sim+5\text{V}$ 。画出电路图 (无需采样保持) 并编写采样 100 个数据的程序。若某个采样值为 70H , 其对应的模拟电压值为多少?



$$70/255*(5V+5V)-5V = -2.25490196V$$

```

START:  MOV  DI, DATBUF
        MOV  BL, 100      ; BL=采样计数
        MOV  AX, 3F0H    ; 口地址 3F0H 对应的有效地址线为 A9~A0
LP1:    MOV  DX, AX       ; DX =  $\overline{CS}$  片选地址 (A9~A3) + 通道选择号 (A2~A0)
        OUT  DX, AL      ; 选择并锁存通道号; 启动转换。AL 中的内容无意义。
        PUSH AX          ;
        CALL DELAY       ; DELAY 为延时 100  $\mu$  S 子程序
        IN   AL, DX      ; 产生 OE 有效信号; 读入数据
        MOV  [DI], AL    ; 保存所采集的数据到数据缓冲区
        INC  DI
        POP  AX          ; 恢复地址
        INC  AX          ; 地址加 1, 指向下 1 个通道
        DEC  BL          ; 100 个通道采样完?
        JNZ  LP1
        结束

```

常考重点

- 8086/8088加电复位后执行的第一条指令地址：FFFF0H
- 8086的基本总线周期。8086与8088的主要区别。
- 中断源 NMI与INTR的区别。
- 8253中GATE的功能和作用，电路图，初始化程序。。
- 已知终端型号，中断处理程序名，编写设置中断向量的程序片段。
- 键盘电路行扫描法和行反转发的实现过程与区别。
- 0832电路图，D/A（A/D）转换基本方法，如何输出锯齿波、三角波、分辨率、精度。
- 8259中断屏蔽寄存器，内部结构，功能。