

PIC 单片机 16F84 的内部硬件资源（三）

10 复位

复位是单片机的初始化操作。其主要功能是把程序计数器 PCL 初始化为 000H，可使 16F84 单片机从 000H 单元开始执行程序。

PIC16F84 芯片有下列几种不同的复位方式。

- (1) 芯片上电复位 POR。
- (2) 正常工作状态下通过外部 MCLR 引脚加低电平复位。
- (3) 在省电休眠状态下通过外部 MCLR 引脚加低电平复位。
- (4) 监视定时器 WDT 超时溢出复位。

PIC16F84 片内集成有“上电复位”POR 电路，对于一般应用，只要把 MCLR 引脚接高电位即可。

在正常工作或休眠状态下用 MCLR 复位，只需在 MCLR 引脚上加一按键瞬间接地即可。单片机 16F84 复位操作，对其它一些寄存器会有影响，如表 1 所示。

11 监视定时器 WDT

单片机系统常用于工业控制，在操作现场通常会有各种干扰，可能会使执行程序弹飞到一种死循环，从而导致整个单片机控制系统瘫痪。如果操作者在场，就可进行人工复位，摆脱死循环。但操作者不能一直监视着系统，即使监视着系统，也往往是引起不良后果之后才进行人工复位。由于 PIC16F84 中具有程序运行自动监视系统，即监视定时器 WDT(Watch Dog Time)，直译为“看门狗”定时器。这好比是主人养了一条狗，主人在正常干活时总不忘每隔一段时间就给狗喂食，狗就保持安静，不影响主人干活。如果主人打瞌睡，不干活了，到一定时间，狗饿了，发现主人还没有给它吃东西，就会大叫起来，把主人唤醒。由此可见，WDT 有如下特性：

- (1) 本身能独立工作，基本上不依赖 CPU。
- (2) CPU 在一个固定的时间间隔中和 WDT 打一次交通(如使其清零，即喂一次狗)，以表明系统目前工作正常。
- (3) 当 CPU 落入死循环后，能被 WDT 及时发觉(如 WDT 计数溢出)，并使系统复位。

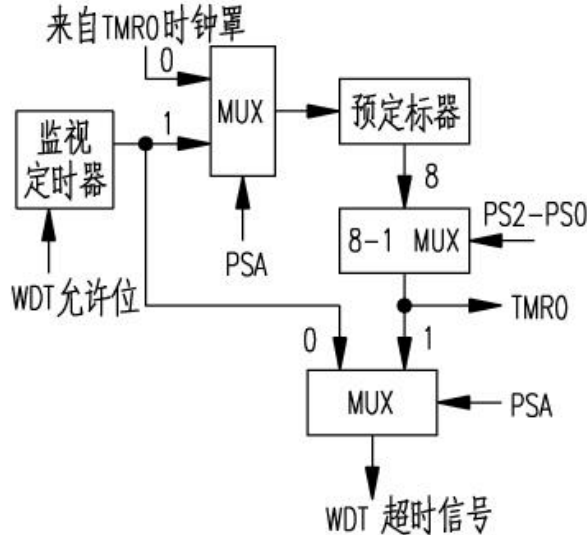
PIC16F84 内的 WDT，其定时计数的脉冲序列由片内独立的 RC 振荡器产生，所以它不需要外接任何器件就可以工作。而且这个片内 RC 振荡器与 OSC1/CLKIN(引脚{16})上的振荡电路无关，即使 OSC1 和 OSC2 上的时钟不工作，WDT 照样可以监视定时。例如：当 PIC16F84 在执行 SLEEP 指令后，芯片进入休眠状态，CPU 不工作，主振荡器也停止工作，但是，WDT 照样可监视定时。当 WDT 超时溢出后，可激活(唤醒)芯片继续正常的操作。而在正常操作期间，WDT 超时溢出将产生一个复位信号。如果不需要这种监视定时功能，在固化编程时，可关闭这个功能。附图是监视定时器的结构框图。表 2 是与 WDT 有关的寄存器。

WDT 的定时周期在不加分频器的情况下，其基本定时时间是 18ms，这个定时时间还受温度、VDD 和不同元器件的工艺参数等的影响。如果需要更长的定时周期，还可以通过软件控制 OPT/ON 寄存器把预分频器配置给 WDT，这个预分频器的最大分频比可达到 1：128。这样就可把定时周期扩大 128 倍，即达到 2.3 秒。

如果把预分频器配置给 WDT，用 CLRWDT 和 SLEEP 指令可以同时给 WDT 和预分频器清零，从而防止计时溢出引起芯片复位。所以在正常情况下，必须在每次计时溢出之前执行一条 CLRWDT 指令(即喂一次“狗”)，以避免引起芯片复位。当系统受到严重干扰处于

失控状态时，就不可能在每次计时溢出之前执行一条 CLR WDT 指令，WDT 就产生计时溢出，从而引起芯片复位，从失控状态又重新进入正常运行状态。

当 WDT 计时溢出时，还会同时清除状态寄存器中的 D4 位 T0，检测 T0 位即可知道复位是否由于 WDT 计时溢出引起的。



12 E2PROM 的使用方法

在 PIC16F84 单片机中，除了可直接寻址的由 SRAM 构成的数据存储器外，还另有可电擦、电写的 E2PROM 数据存储器。该 E2PROM 共有 64 字节，其地址为 00~3FH 单元。由于 E2PROM 具有在线改写，并在掉电后仍能保持数据的特点，可为用户的特殊应用提供方便。16F84 的 E2PROM 在正常操作时的整个 VDD 工作电压范围内是可读写的，典型情况下可重写 100 万次，数据保存期大于 40 年。

PIC16F84 的 E2PROM 并未映象在寄存器组空间中，所以它们不能像 SRAM 通用寄存器那样用指令直接寻址访问，而需要通过专用寄存器进行间接寻址操作。因此，在 16F84 中增加了以下四个专用寄存器，即 EECON1、EECON2、EEDATA、EEADR，专门用于片内对 E2PROM 的操作。该专用寄存器中，EEDATA 存放 8 位读/写数据，EEADR 存放正在被访问的 E2PROM 存储单元的地址。

EECON1 是只有低五位的控制寄存器，其高三位不存在，读作“0”。具体见下表。

D7	D6	D5	D4	D3	D2	D1	D0
—	—	—	EEIF	WRERR	WREN	WR	RD

控制位 RD 和 WR 分别用于读写操作的启动，这两位可以由软件置 1，以启动读、写操作，但不能用软件清零，原因是防止不恰当的软件操作会使写入失败。当读写操作完成后由硬件自动清零，表示此刻未对 E2PROM 进行读写操作。当 WREN 位被置 1 时，允许进行写操作，而在上电时该位被清零。在正常操作时，一旦有 MCLR 或 WDT 复位，WRERR 位就置 1，表示写操作被中止。当写操作完成时，EEIF 被置 1(需由软件清零)；当写操作未完成或尚未启动时，EEIF 为“0”。

EECON2 仅是一个逻辑上的寄存器，而不是一个物理上存在的寄存器，读出时将总是为零。它只在写操作时起作用。

(1)E2PROM 的读操作

为进行一次 E2PROM 读操作，需执行如下步骤：

1)将 E2PROM 的单元地址放入 EEADR。2)置 RD(EECON 的 D0 位)=1。3)读取 EEDATA 寄存器。

程序段举例，读取 25H 处的 E2PROM 存储器数据：

```
...
BCF    STATUS, RP0 ; 选 Bank0
MOVLW 25H
MOVWF EEADR      ; 地址 25H→EEADR
BSF    STATUS, RP0 ; 选 Bank1
BSF    EECON1, RD ; 启动读操作
BCF    STATUS, RP0 ; 选 Bank0
MOVF  EEDATA, W ; 将 E2PROM 数据
        ...      ; 读入 W 寄存器
```

(2)E2PROM 的写操作

要进行一次 E2PROM 写操作，需执行如下步骤：

1)将 E2PROM 单元地址放入 EEADR；2)将写入数据放入 EEDATA；3)执行一段控制程序段。

例如：将数据 99H 写入 E2PROM 的 25H 单元，需执行下列程序：

```
...
BCF    STATUS, RP0 ; 选 Bank0
MOVLW 25H
MOVWF EEADR      ; 地址→EEADR
MOVLW 99H
MOVWF EEDATA     ; 写入数据→EEDATA
BSF    STATUS, RP0 ; 选 Bank1
BSF    EECON1, WREN; 写操作功能允许
1 BCF  INTCON, GIE ; 关闭总中断
2 MOVLW 0x55
3 MOVWF EECON2
4 MOVLW 0xAA
5 MOVWF EECON2 ; 操作 EECON2
6 BSF  EECON1, WR; 启动写操作
7 BSF  INTCON, GIE ; 开放总中断
...
```

注意：上列程序中的 2~6 条各语句必须严格执行，否则不能启动 E2PROM 的写操作。而 1~7 条则是我们建议用户执行的操作，即在 E2PROM 写操作序列步骤中要关闭所有中断，以免这个序列被中断打断。

另外，WREN(EECON1 的 D2 位)是用来保证 E2PROM 不会被意外写入而设置的，所以，在平时，用户程序应保持 WREN=0 以禁止写操作。只有当需对 E2PROM 写入时才置 WREN=1，并在写入完成后将其恢复为 0。用户只有置 WREN=1 后才能置 WREN=1 启动写操作。上电复位后 WREN 位自动清零。

E2PROM 写操作约需 10ms 的时间才能完成。用户程序可通过查询 WR 位的状态(当 WR=0 时表示操作已完成)，或者用 E2PROM 写入完成中断来判断 E2PROM 写操作是否完成。如要使用中断，应先置 EEIF(INTCON 的 D6)为 1，以开中断。E2PROM 写完成要中断标志位 EEIF，只能用软件清零。

本文内容来自互联网，著作权归原作者所有。由电子零件城（<http://www.epcity.com/>）整理并制作成 PDF 文件，仅供个人学习之用，不得用于任何商业目的，否则后果自负。如果您认为本 PDF 文件侵犯了您的任何权利，请来信 epcity@epcity.com 通知，本站立即删除。

搜集整理：电子零件城-笨笨兔（QQ：154502842） 2004-04-10