

---

---

## 第 6 章 振荡器

---

---

### 目录

本章包括下列主题：

6.1	简介 .....	6-2
6.2	CPU 时钟分配 .....	6-3
6.3	振荡器配置 .....	6-3
6.4	控制寄存器 .....	6-5
6.5	主振荡器 (POSC) .....	6-10
6.6	锁相环 (PLL) .....	6-14
6.7	低功耗辅助振荡器 (SOSC) .....	6-15
6.8	内部快速 RC 振荡器 (FRC) .....	6-16
6.9	内部低功耗 RC 振荡器 (LPRC) .....	6-16
6.10	故障保护时钟监视器 (FSCM) .....	6-17
6.11	时钟切换操作 .....	6-18
6.12	双速启动 .....	6-21
6.13	交流电气规范 .....	6-22
6.14	设计技巧 .....	6-23
6.16	相关应用笔记 .....	6-25
6.17	版本历史 .....	6-26

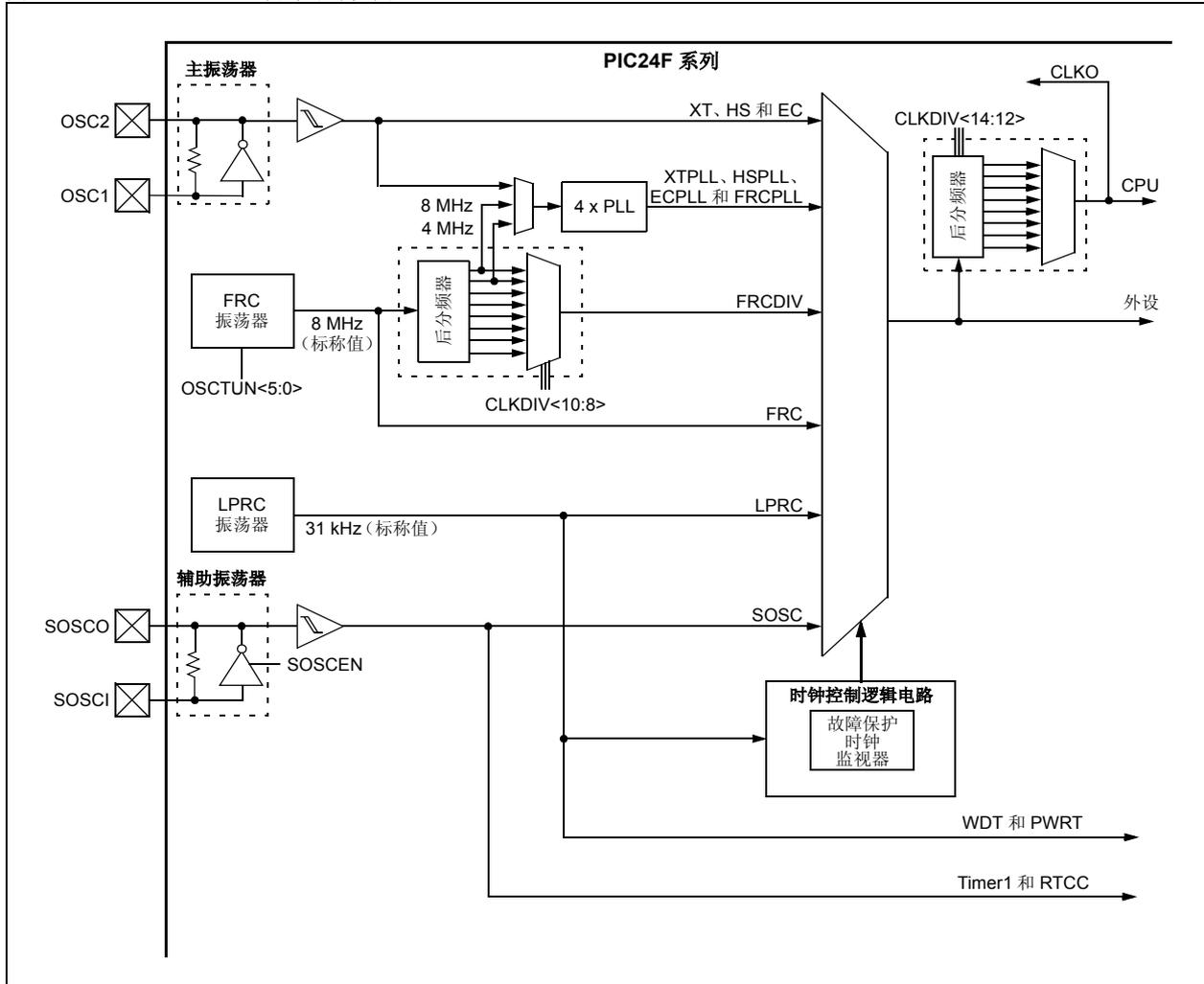
## 6.1 简介

本章描述 PIC24F 振荡器系统及其操作。PIC24F 振荡器系统具有以下模块和特性：

- 可选择 4 个外部和内部振荡器作为时钟源，提供 11 种不同的时钟模式
- 具有片上 4x PLL，可基于所选的内部和外部振荡器源升高内部工作频率
- 可通过软件控制在各种时钟源之间切换
- 可通过软件控制预分频器有选择地为 CPU 提供时钟，以节省系统功耗
- 具有故障保护时钟监视器（FSCM），可检测时钟故障，以使应用可安全地恢复或关闭

图 6-1 给出了振荡器系统的简化图。

图 6-1: PIC24F 系列时钟框图



## 6.2 CPU 时钟分配

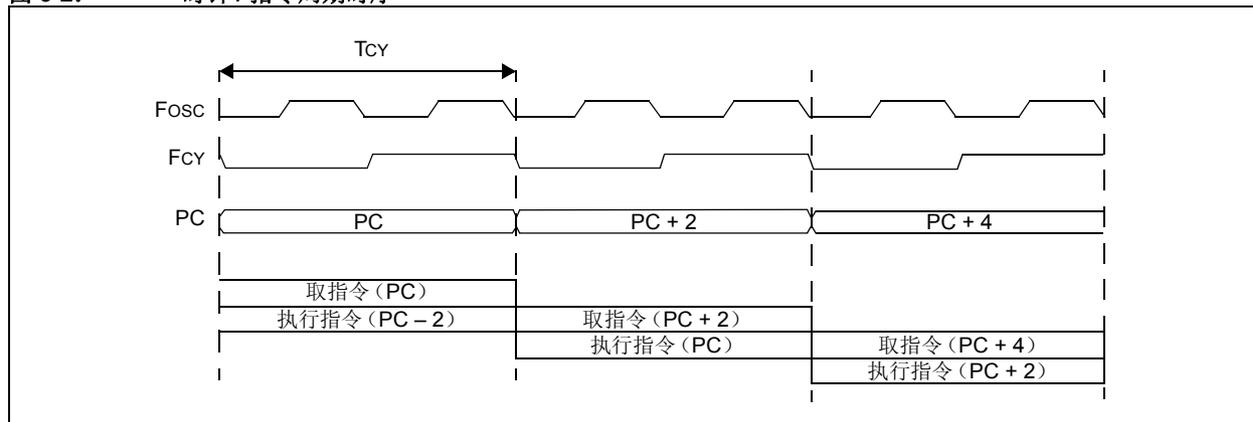
系统时钟源可以由以下 4 种之一提供：

- OSC1 和 OSC2 引脚上的主振荡器（POSC）
- SOSCI 和 SOSCO 引脚上的辅助振荡器
- 内部快速 RC 振荡器（FRC）
- 内部低功耗 RC 振荡器（LPRC）

主振荡器和 FRC 源可以选择使用内部 4x PLL。FRC 时钟源的频率可以通过可编程时钟分频器降低。选定的时钟源将产生处理器和外设时钟源。

处理器时钟源需进行二分频，以产生内部指令周期时钟  $F_{CY}$ 。在本文档中，指令周期时钟也表示为  $F_{osc}/2$ 。图 6-2 中的时序图显示了处理器时钟源和指令执行之间的关系。内部指令周期时钟  $F_{osc}/2$  可以在 OSC2 I/O 引脚提供，用于主振荡器的一些工作模式。

图 6-2: 时钟 / 指令周期时序



## 6.3 振荡器配置

在发生器件上电复位事件时使用的振荡器源（以及工作模式）使用配置位设置进行选择。振荡器配置位在程序存储器中的配置寄存器中进行设置（更多详细信息，请参见具体产品数据手册）。主振荡器配置位  $POSCMD1:POSCMD0$ （配置字 2<1:0>）和振荡器配置位  $FNOSC2:FNOSC0$ （配置字 2<10:8>）用于选择在上电复位时使用的振荡器源。默认情况下（未编程时）将选择 FRC 振荡器和预分频器（ $FRCDIV$ ）。通过编程这些位单元，可以选择辅助振荡器或一个内部振荡器。这些配置位使用户可以选择 11 种不同的时钟模式，如表 6-1 所示。

表 6-1: 时钟选择的配置位值

振荡器模式	振荡器源	POSCMD1: POSCMD0	FNOSC2: FNOSC0	注
快速 RC 振荡器和预分频器 (FRCDIV)	内部	xx	111	1, 2
(保留)	内部	xx	110	1
低功耗 RC 振荡器 (LPRC)	内部	xx	101	1
辅助 (Timer1) 振荡器 (SOSC)	辅助	xx	100	1
主振荡器 (HS) 和 PLL 模块 (HSPLL)	主	10	011	
主振荡器 (XT) 和 PLL 模块 (XTPLL)	主	01	011	
主振荡器 (EC) 和 PLL 模块 (ECPPL)	主	00	011	
主振荡器 (HS)	主	10	010	
主振荡器 (XT)	主	01	010	
主振荡器 (EC)	主	00	010	
快速 RC 振荡器和 PLL 模块 (FRCPLL)	内部	xx	001	1
快速 RC 振荡器 (FRC)	内部	xx	000	1

注 1: OSC2 引脚功能由 OSCIOFCN 配置位决定。

2: 未编程 (已擦除) 器件的默认振荡模式。

### 6.3.1 时钟切换模式配置位

FCKSM 配置位 (配置字 2<7:6>) 用于联合配置器件时钟切换和故障保护时钟监视器 (FSCM)。只有将 FCKSM1 编程 (0) 时, 才会使能时钟切换。只有同时将 FCKSM1:FCKSM0 编程 (00) 时, 才会使能 FSCM。

### 6.3.2 非晶振模式下的 OSC1 和 OSC2 引脚功能

当 OSC1 和 OSC2 上的主振荡器未配置为时钟源 (POSCMD1:POSCMD0 = 11) 时, OSC1 引脚自动重新配置为数字 I/O。在该配置中, 以及主振荡器配置为 EC 模式 (POSCMD1:POSCMD0 = 00) 时, 也可以通过编程 OSCIOFCN 配置位 (配置字 2<5>) 将 OSC2 引脚配置为数字 I/O。

当 OSCIOFCN 未编程 (1) 时, OSC2 上会产生  $F_{osc}/2$  时钟输出, 用于进行测试或同步。OSCIOFCN 编程 (0) 时, OSC2 引脚成为通用 I/O 引脚。在这两种配置中, OSC1 和 OSC2 之间的反馈器件被关闭, 以节省功耗。

## 6.4 控制寄存器

振荡器的操作由 3 个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN

### 6.4.1 振荡器控制寄存器（OSCCON）

OSCCON 寄存器（寄存器 6-1）是振荡器的主要控制寄存器。它控制时钟源切换和时钟源监控。

COSC 状态位是只读位，指示器件当前工作所使用的振荡器源。在上电复位和主清零复位时，COSC 位默认为选择内部快速 RC 振荡器和预分频器（FRCDIV），配置的频率为 4 MHz。器件将自动执行时钟切换，将时钟源切换为由 FNOSC 配置位（配置字 2<10:8>）选择的新振荡器源。COSC 位将发生改变，在时钟切换操作结束时指示新的振荡器源。

NOSC 状态位选择下一个时钟切换操作的时钟源。在上电复位和主清零复位时，这些位将自动选择 FNOSC 配置位指定的振荡器源。这些位可以由软件修改。

**注：** 必须先执行解锁序列，之后才能写 OSCCON。更多信息，请参见第 6.11.2 节“振荡器切换序列”。

在 FCKSM1 配置位置 1 时，将 CLKLOCK 位（OSCCON<7>）置 1 可防止发生时钟切换。如果 FCKSM1 位清零，则 CLKLOCK 位状态会被忽略，器件可以发生时钟切换。

LOCK 状态位（OSCCON<5>）是只读位，指示 PLL 电路的状态。在 PLL 实现频率锁定时该位置 1，在产生有效的时钟切换序列时该位复位。每当 PLL 不与振荡器配合用作当前时钟源时，它读为 0。

CF 状态位（OSCCON<3>）是可读 / 可清零状态位，指示发生时钟故障。每当发生有效的时钟切换时，该位复位。

SOSCEN 控制位（OSCCON<1>）用于使能或禁止 32 kHz SOSC 晶体振荡器。

OSWEN 控制位（OSCCON<0>）用于启动时钟切换操作。OSWEN 在成功进行时钟切换或任何冗余时钟切换后自动清零，并在切换到 FRC 后由 FSCM 清零。

# PIC24F 系列参考手册

寄存器 6-1: **OSCCON: 振荡器控制寄存器**

U-0	R-0	R-0	R-0	U-0	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0	U-0	R-0 <sup>(2)</sup>	U-0	R/CO-0	U-0	R/W-0	R/W-0
CLKLOCK	—	LOCK	—	CF	—	SOSCEN	OSWEN
bit 7							bit 0

<b>图注:</b>	U = 未实现位, 读为 0		
R = 可读位	W = 可写位	CO = 只可清零位	SO = 只可置 1 位
-n = 全部复位时的值	1 = 置 1	0 = 清零	x = 未知

- bit 15 **未实现:** 读为 0
- bit 14-12 **COSC2:COSC0:** 当前振荡器选择位
  - 111 = 快速 RC 振荡器和预分频器 (FRCDIV)
  - 110 = 保留
  - 101 = 低功耗 RC 振荡器 (LPRC)
  - 100 = 辅助振荡器 (SOSC)
  - 011 = 主振荡器和 PLL 模块 (XTPLL、HSPLL 或 ECPLL)
  - 010 = 主振荡器 (XT、HS 或 EC)
  - 001 = 快速 RC 振荡器和通过预分频器 (FRCPLL) 使用 PLL 模块<sup>(3)</sup>
  - 000 = 快速 RC 振荡器 (FRC)
- bit 11 **未实现:** 读为 0
- bit 10-8 **NOSC2:NOSC0:** 新振荡器选择位
  - 111 = 快速 RC 振荡器和预分频器 (FRCDIV)
  - 110 = 保留
  - 101 = 低功耗 RC 振荡器 (LPRC)
  - 100 = 辅助振荡器 (SOSC)
  - 011 = 主振荡器和 PLL 模块 (XTPLL、HSPLL 或 ECPLL)
  - 010 = 主振荡器 (XT、HS 或 EC)
  - 001 = 快速 RC 振荡器和通过预分频器 (FRCPLL) 使用 PLL 模块<sup>(3)</sup>
  - 000 = 快速 RC 振荡器 (FRC)
- bit 7 **CLKLOCK:** 时钟选择锁定使能位
  - 如果使能 FSCM (FCKSM1 = 1):
  - 1 = 时钟和 PLL 选择被锁定。
  - 0 = 时钟和 PLL 选择不锁定, 可以通过将 OSWEN 位置 1 而修改。
  - 如果禁止 FSCM (FCKSM1 = 0):
  - 时钟和 PLL 选择始终不锁定, 可以通过将 OSWEN 位置 1 而修改。
- bit 6 **未实现:** 读为 0
- bit 5 **LOCK:** PLL 锁定状态位
  - 1 = PLL 模块处于锁定状态, 或者满足 PLL 模块起振定时器
  - 0 = PLL 模块脱离锁定状态, PLL 起振定时器正在运行或 PLL 被禁止
- bit 4 **未实现:** 读为 0
- bit 3 **CF:** 时钟故障检测位
  - 1 = FSCM 检测到时钟故障
  - 0 = 未检测到时钟故障
- bit 2 **未实现:** 读为 0

**注** 1: 这些位的复位值由 FNOSC 配置位决定。  
 2: 在任意有效时钟切换期间, 或者每当选择了非 PLL 时钟模式时, 也复位为 0。  
 3: 仅在选择 4 MHz 和 8 MHz FRC 后分频器选项时可用。

**寄存器 6-1: OSCCON: 振荡器控制寄存器 (续)**

bit 1	<b>SOSCEN:</b> 32 kHz 辅助振荡器 (SOSC) 使能位 1 = 使能辅助振荡器 0 = 禁止辅助振荡器
bit 0	<b>OSWEN:</b> 振荡器切换使能位 1 = 启动振荡器切换, 切换为 NOSC2:NOSC0 位指定的振荡器 0 = 振荡器切换完成

- 注**
- 1: 这些位的复位值由 FNOSC 配置位决定。
  - 2: 在任意有效时钟切换期间, 或者每当选择了非 PLL 时钟模式时, 也复位为 0。
  - 3: 仅在选择 4 MHz 和 8 MHz FRC 后分频器选项时可用。

**6.4.2 时钟分频寄存器 (CLKDIV)**

时钟分频寄存器 (寄存器 6-2) 控制与打盹模式关联的功能, 以及 FRC 振荡器的后分频器。

ROI 位 (CLKDIV<15>) 允许发生中断时退出打盹模式, 并自动为处理器和外设时钟选择 1:1 的比率。DOZEN 位 (CLKDIV<11>) 在从打盹模式退出后清零。清零 ROI 位可以阻止中断影响打盹模式。

DOZE 位 (CLKDIV<14:12>) 用于选择处理器时钟与外设时钟的比率。其范围为 1:1 至 1:128, 可由软件选择。主复位和上电复位时默认为 1:1 的比率。该功能使 CPU 可以节省功耗, 而不会破坏外设的操作。

将 DOZEN 位置 1 会将器件置为打盹模式, 并预定处理器时钟后分频器。该位在 ROI 位置 1 和发生中断时清零。

RCDIV 位 (CLKDIV<10:8>) 用于选择 FRC 振荡器输出的后分频器选项, 使用户可以选择低于标称频率 8 MHz 的较低时钟频率。第 6.8.2 节“FRC 后分频器模式 (FRCDIV)”和第 6.8.3 节“使用 PLL 的 FRC 振荡器模式 (FRCPLL)”中详细介绍了该选项。

# PIC24F 系列参考手册

**寄存器 6-2: CLKDIV: 时钟分频寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN <sup>(1)</sup>	RCDIV2	RCDIV1	RCDIV0
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 -n = 全部复位时的值          1 = 置 1                      0 = 清零                      x = 未知

**bit 15 ROI:** 中断恢复位  
 1 = 发生中断时清零 DOZEN 位, 并将 CPU/ 外设时钟比率复位为 1:1  
 0 = 中断对 DOZEN 位无影响

**bit 14-12 DOZE2:DOZE0:** CPU/ 外设时钟比率选择位  
 111 = 1:128  
 110 = 1:64  
 101 = 1:32  
 100 = 1:16  
 011 = 1:8  
 010 = 1:4  
 001 = 1:2  
 000 = 1:1

**bit 11 DOZEN:** 打盹使能位 <sup>(1)</sup>  
 1 = DOZE2:DOZE0 位指定 CPU 外设时钟比  
 0 = 处理器时钟 / 外设时钟比设置为 1:1

**bit 10-8 RCDIV2:RCDIV0:** FRC 后分频器选择位  
 111 = 31.25 kHz (256 分频)  
 110 = 125 kHz (64 分频)  
 101 = 250 kHz (32 分频)  
 100 = 500 kHz (16 分频)  
 011 = 1 MHz (8 分频)  
 010 = 2 MHz (4 分频)  
 001 = 4 MHz (2 分频)  
 000 = 8 MHz (直接 FRC 输出)

**bit 7-0 未实现:** 读为 0

**注 1:** 该位在 ROI 位置 1 和发生中断时自动清零。

### 6.4.3 振荡器调节寄存器 (OSCTUN)

FRC 振荡器调节寄存器 (寄存器 6-3) 使用户可以对 FRC 振荡器在大约  $\pm 12\%$  的范围内进行微调。将寄存器值递增或递减都会将 FRC 振荡器的出厂校准频率改变一个固定的量。

寄存器 6-3: OSCTUN: FRC 振荡器调节寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0
bit 7						bit 0	

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 全部复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6

未实现: 读为 0

bit 5-0

**TUN5:TUN0:** FRC 振荡器调节位

011111 = 最高频率偏移

011110 =

•

•

•

000001 =

000000 = 中心频率, 振荡器以出厂校准频率工作

111111 =

•

•

•

100001 =

100000 = 最低频率偏移

## 6.5 主振荡器 (POSC)

PIC24F 系列在 OSC1 和 OSC2 引脚提供了主振荡器。通常，主振荡器可以配置为使用外部时钟输入或外部晶振。后面几节描述了主振荡器工作模式的更多详细信息。主振荡器具有 6 种工作模式，表 6-2 中列出了这些工作模式。

表 6-2: 主振荡器工作模式

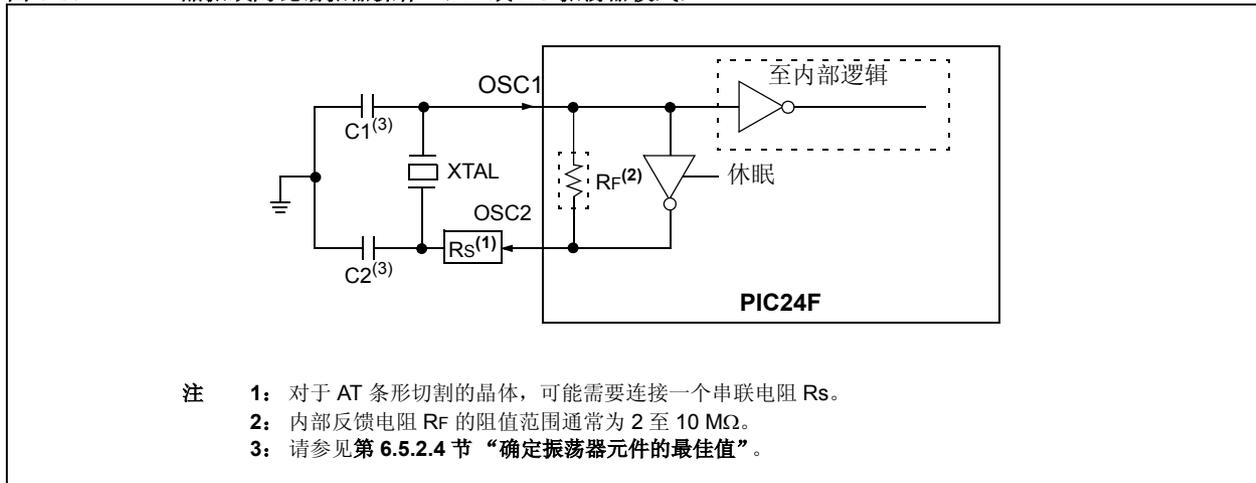
振荡器模式	说明	OSC2 引脚功能
EC	外部时钟输入 (0-32 MHz)	Fosc/2
ECPLL	外部时钟输入 (4-8 MHz), PLL 使能	Fosc/2
HS	10 MHz-32 MHz 晶振	(注 1)
HSPLL	4 MHz-8 MHz 晶振, PLL 使能	(注 1)
XT	3.5 MHz-10 MHz 晶振	(注 1)
XTPLL	3.5 MHz-8 MHz 晶振, PLL 使能	(注 1)

注 1: 在这些模式中，外部晶振连接到 OSC1 和 OSC2。

POSCMD 和 FNOSC 配置位（分别为配置字 2<1:0> 和 <10:8>）用于选择主振荡器的工作模式。POSCMD1:POSCMD0 位用于选择要使用的特定子模式（XT、HS 或 EC），而 FNOSC2:FNOSC0 位决定振荡器是单独使用还是与内部 PLL 一起使用。每当 COSC 位（OSCCON<14:12>）设为 010 或 011 时，PIC24F 使用主振荡器工作。

请参见具体器件数据手册中的“电气特性”一节，了解关于每种晶振模式的频率范围的更多信息。

图 6-3: 晶振或陶瓷谐振器操作 (XT 或 HS 振荡器模式)



### 6.5.1 选择主振荡器模式

XT 和 HS 模式之间的主要区别在于振荡器电路的内部反相器的增益不同，从而使频率范围也不同。XT 模式是中等功耗的中等频率模式。HS 模式使用晶振提供最高的振荡器频率。在 HS 和 XT 振荡器模式下，OSC2 提供晶振反馈。

使用 PLL 电路的 EC 和 HS 模式可提供最高的器件工作频率。在这些模式下，振荡器电路将消耗大部分的电流，因为 PLL 被使能对振荡器的频率进行 4 倍频。

通常，用户应选择具有最低增益但仍满足规定的振荡器选项。这样可以产生较小的动态电流 (IDD)。每种振荡器模式的频率范围是建议的频率截止范围，但也可以选择不同的增益模式，但要先执行全面的验证（电压、温度和元件差异，如电阻、电容和内部振荡器电路）。

在所有的 EC 模式下，振荡器反馈电路均被禁止。OSC1 引脚是高阻抗输入，可以由 CMOS 驱动器驱动。

如果主振荡器配置为使用外部时钟输入，则不需要 OSC2 引脚支持振荡器功能。对于这些模式，OSC2 引脚可以用作附加的器件 I/O 引脚或时钟输出引脚。在 OSC2 引脚用作时钟输出引脚时，输出频率为  $F_{osc}/2$ 。

### 6.5.2 晶体振荡器和陶瓷谐振器

在 XT 和 HS 模式下，晶振或陶瓷谐振器连接到 OSC1 和 OSC2 引脚，以产生振荡（图 6-3）。PIC24F 振荡器的设计要求使用平行切割的晶体。使用顺序切割的晶体，可能会使振荡器产生的频率超出晶体制造厂商所给的参数范围。

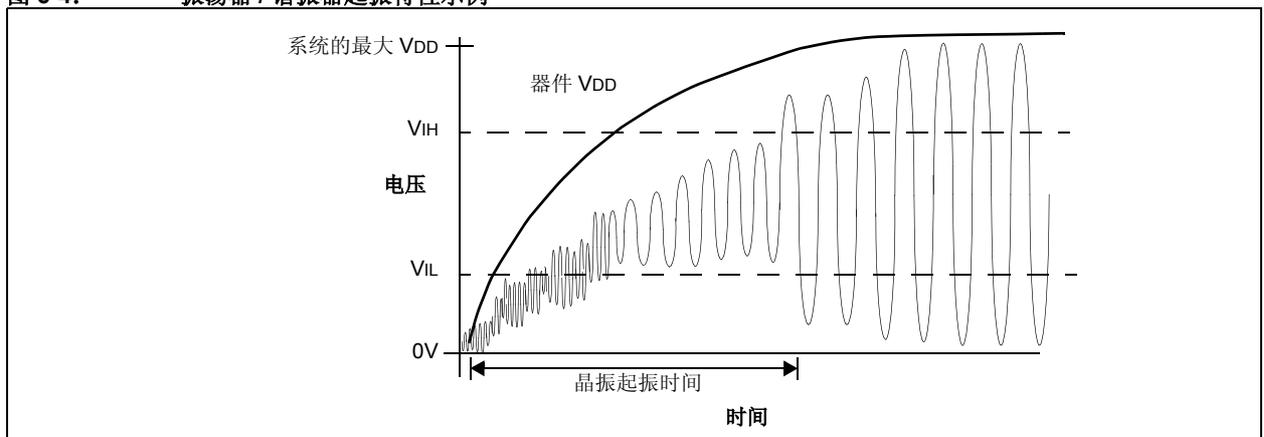
#### 6.5.2.1 振荡器 / 谐振器起振

当器件电压从 Vss 上升时，振荡器将开始振荡。振荡器开始振荡所需要的时间取决于许多因素，包括：

- 晶振 / 谐振器频率
- 所使用的电容值
- 串联电阻的阻值和类型（如使用）
- 器件 VDD 上升时间
- 系统温度
- 为器件选择的振荡器模式（选择内部振荡器反相器的增益）
- 晶振品质
- 振荡器电路布线
- 系统噪声

图 6-4 中显示了典型的晶振或谐振器的起振过程。请注意，振荡器并不是瞬间即达到稳定振荡状态。

图 6-4: 振荡器 / 谐振器起振特性示例



## 6.5.2.1.1 主振荡器从休眠模式起振

从休眠模式唤醒时是振荡器起振最困难的时候。这是因为两个负载电容都已不完全充电至某个静态电量值，在唤醒时的相位差最小。因而，达到稳定振荡需要更长的时间。另外请记住，低电压、高温和较低频率时钟模式也会限制环路增益，进而影响起振。

以下各因素都会延长起振时间：

- 低频设计（及低增益时钟模式）
- 无噪声环境（例如电池驱动的器件）
- 在屏蔽箱中工作（在嘈杂的 RF 区域之外）
- 低电压
- 高温
- 从休眠状态唤醒

另一方面，电路噪声实际上可以帮助“开动”振荡器，帮助缩短振荡器起振时间。

## 6.5.2.2 振荡器起振定时器

为了确保晶体振荡器（或陶瓷谐振器）已启动和稳定，提供了一个振荡器起振定时器（OST）。OST 是一个简单的 10 位计数器，在将振荡器时钟释放给系统的其他部分之前计数 1024 个  $T_{osc}$  周期。这个超时周期称为  $T_{OST}$ 。振荡器信号的振幅必须达到振荡器引脚的  $V_{IL}$  和  $V_{IH}$  阈值，之后 OST 才可以开始对周期进行计数。

每次振荡器必须重新启动（即，在 POR、BOR 和从休眠模式唤醒）时，都需要等待时长为  $T_{OST}$  的时间间隔。振荡器起振定时器应用于主振荡器的 XT 和 HS 模式和辅助振荡器 SOSC（见第 6.7 节“低功耗辅助振荡器（SOSC）”）。

## 6.5.2.3 调节振荡器电路

因为 Microchip 器件具有很宽的工作范围（频率、电压和温度；取决于所订购的部件和版本），可以使用各种不同质量和制造商的外部元件（晶振和电容等），所以需要执行工作验证来确保选择的元件满足应用要求。这些外部元件的选择和排列有许多需要考虑的因素。根据实际应用，这些因素可包括以下任意一项：

- 放大器增益
- 所需的频率
- 晶振的谐振频率
- 工作温度
- 电源电压范围
- 起振时间
- 稳定性
- 晶振寿命
- 功耗
- 电路的简化
- 标准元件的使用
- 元件数量

#### 6.5.2.4 确定振荡器元件的最佳值

选择元件的最佳方法是运用相关知识进行大量的试验评估和测试。晶振通常只需根据它们的并联谐振频率进行选择；不过，其他一些参数对于设计可能也很重要，例如温度或频率容差。Microchip 应用笔记 AN588 “PIC<sup>®</sup> Microcontroller Oscillator Design Guide” 是了解关于晶振工作和订购信息的上佳参考。

PIC24F 内部振荡器电路是并联振荡器电路，所以需要选择并联谐振晶振。负载电容通常规定在 22 pF 至 33 pF 范围内。负载电容处于该范围内时，晶振的振荡频率最接近所需频率。为了获得其他方面的好处，可能还需要更改这些值，我们将在后面说明。

时钟模式主要根据晶体振荡器的所需频率进行选择。XT 和 HS 振荡器模式之间的主要区别在于振荡器电路的内部反相器的增益不同，从而使频率范围也不同。通常，用户应选择具有最低增益但仍满足规定的振荡器选项。这样可以产生较小的动态电流 ( $I_{DD}$ )。每种振荡器模式的频率范围是建议的频率截止范围，但也可以选择不同的增益模式，但要先执行全面的验证（电压、温度和元件差异，如电阻、电容和内部振荡器电路）。C1 和 C2 也首先应根据晶振制造商建议的负载电容和器件数据手册中提供的几个表进行选择。器件数据手册中给出的值只能作为初始参考值，因为晶振制造商、电源电压和其他已提及的因素可能会导致电路不同于工厂特性测定过程中使用的电路。

理想情况下，所选择的电容应使电路可以在电路需工作的最高温度和最低 VDD 条件下振荡。高温和低 VDD 对环路增益都有一定限度的影响，所以如果电路可以在这些极限条件下工作，设计者就可确定电路能在其他温度和电源电压组合下正常工作。在最高增益环境下（最高 VDD 和最低温度），输出正弦波应不会被削波；在最低增益环境下（最低 VDD 和最高温度），正弦输出振幅应足够高，可以满足器件数据手册中列出的时钟的逻辑输入要求。OSC1 可能具有规定的 VIL 和 VIH 电平（请参见具体产品数据手册了解更多信息）。

改善起振的一种方法是使 C2 电容值大于 C1。这会使晶振在上电时产生更大的相移，从而加速振荡器起振。这两个电容除了辅助晶振产生适当的频率响应之外，提高它们的电容值还能降低环路增益。可以通过选择 C2 来影响电路的整体增益。如果晶振过驱动，使用较高的 C2 可以降低增益（另请参见关于 Rs 的讨论）。如果电容值过高，电容会存储和释放过多的电流通过晶振，所以 C1 和 C2 的电容不应过大。遗憾的是，测量晶振的功耗（瓦特数）很困难，但是如果偏离建议值不是太多，则可以不考虑这一点。

如果在选择了满足要求的所有其他外部元件之后，晶振仍然过驱动，则可以在电路中增加一个串联电阻 Rs。这可以通过使用示波器检查 OSC2 引脚（被驱动的引脚）确定。将探针连接到 OSC1 引脚会使引脚负载过大，对性能产生负面影响。请记住，示波探针会将其自身的电容加到电路中，所以在设计中可能必须要考虑这一点（即，如果电路在 C2 为 22 pF 时工作状态最佳，而示波探针电容为 10 pF，则实际的电容为 33 pF）。输出信号不应被削波或拉平。过驱动晶振还会导致电路迁变至高次谐波，或者甚至损坏晶振。

OSC2 信号应为平滑的正弦波，可以轻松跨越时钟输入引脚的输入最小值和最大值。有一个简单的方法可对其进行设置，即在器件需工作的最低温度和最高 VDD 条件下再次测试电路，然后检查输出。此时，时钟输出振幅应最大。如果正弦波被削波，或者正弦波在接近 VDD 和 VSS 时失真，升高负载电容会导致过多的电流流过晶振，或者导致电容值过于偏离制造商的规定值。要调节晶振电流，可以在晶振反相器输出引脚和 C2 之间添加一个微调电位器，并对它进行调节，直到输出平滑正弦波为止。在低温和高 VDD 的极限条件下，晶振将通过最高的驱动电流。

应在这些限制条件下对微调电位器进行调节，以防止过驱动。然后，加入最接近标准值的  $R_s$  来代替微调电位器。如果  $R_s$  阻值过大，超过 20 k $\Omega$ ，输入与输出的隔离度将过大，使时钟更容易受噪声影响。如果确定需要这么高的阻值来防止晶振过驱动，可以尝试升高 C2 来进行补偿，或者尝试改变振荡器工作模式。请尽量使用约为 10 k $\Omega$  或以下的  $R_s$  阻值，并且负载电容不要过于偏离制造商规定值的组合。

## 6.5.3 外部时钟输入

在 EC 模式下，OSC1 引脚处于高阻态，可以由 CMOS 驱动器驱动。OSC2 引脚可以通过选择 OSCIOFCN 位（配置字 2<5>）而配置为 I/O 或时钟输出（Fosc/2）。OSCIOFCN 置 1（图 6-5）时，时钟输出可用于进行测试或同步。OSCIOFCN 清零（图 6-6）时，OSC2 引脚成为通用 I/O 引脚。OSC1 和 OSC2 之间的反馈器件被关闭，以节省功耗。

图 6-5: 外部时钟输入的工作原理（OSCIOFCN = 1）

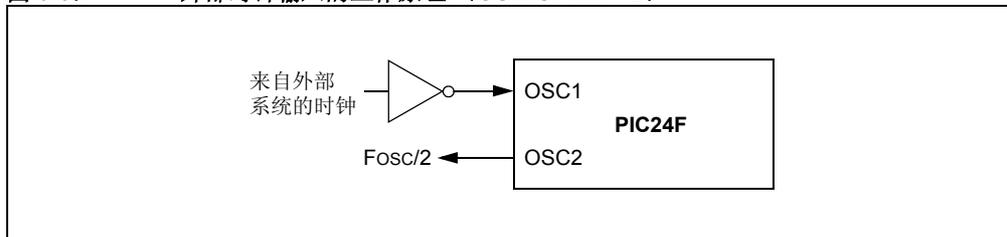
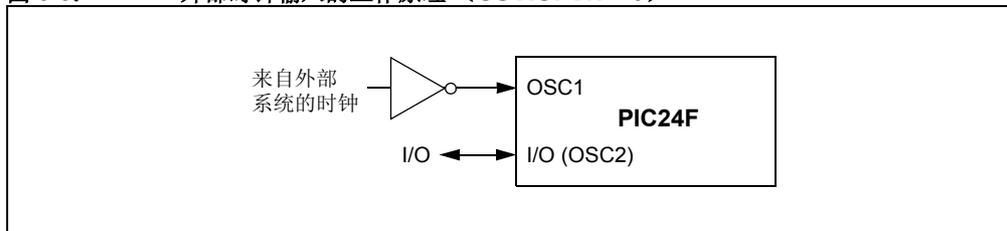


图 6-6: 外部时钟输入的工作原理（OSCIOFCN = 0）



## 6.6 锁相环（PLL）

PLL 提供固定倍频的 4x 倍频器，可以用于 XT、HS 和 EC 主振荡器和内部快速 RC 振荡器（FRC）。每当 PLL 的时钟源改变时，PLL 就绪定时器必须复位，以允许 PLL 与新的时钟源同步。在就绪定时器计数达到所需的时间后，PLL 输出即可供使用。

### 6.6.1 PLL 锁定状态

LOCK 位（OSCCON<5>）是只读位，指示 PLL 电路的锁定状态。在 PLL 实现锁定的典型时间延时（也称为 TLock）之后，该位自动置 1。如果 PLL 未在起振期间正常达到稳定，则 LOCK 可能不会反映实际的 PLL 锁定状态，该位也不会检测正常工作期间 PLL 失锁的情况。

LOCK 位在上电复位时和时钟切换（PLL 被选择作为目标时钟源）时清零。当选择任何不使用 PLL 的时钟源时，它保持清零。

请参见具体器件数据手册中的“电气规范”一节，了解关于 PLL 锁定时间间隔的更多信息。

## 6.7 低功耗辅助振荡器 (SOSC)

低功耗或辅助振荡器 (SOSC) 特别为使用 32.768 kHz 晶振的低功耗操作而设计。振荡器位于 SOSCO 和 SOSCI 器件引脚, 用作低功耗操作的辅助晶振时钟源。它还可以为实时时钟应用程序驱动 Timer1 或实时时钟 / 日历模块。

### 6.7.1 使能辅助振荡器

SOSC 的操作通过 FNOSC 配置位进行选择, 之后由 SOSCEM 位 (OSCCON<1>) 控制。将 SOSCEM 置 1 可以使能振荡器; SOSCO 和 SOSCI 引脚由振荡器控制, 不能用作端口 I/O 或其他功能。

**注:** 在写 OSCCON 之前, 必须先执行解锁序列。更多信息, 请参见第 6.11.2 节 “振荡器切换序列”。

### 6.7.2 辅助振荡器连续操作

在 SOSCEM 置 1 时, SOSC 始终使能。如果让振荡器在所有时间均运行, 则可以快速切换到 32 kHz 系统时钟进行低功耗操作。如果主振荡器是晶振类型的时钟源, 则恢复为使用较快速的主振荡器时, 将仍然需要振荡器起振时间 (见第 6.5.2.2 节 “振荡器起振定时器”)。

此外, 对于使用 Timer1 或 RTCC 模块的实时时钟应用程序, 振荡器需要在所有时间均保持运行。更多详细信息, 请参见第 14 章 “定时器” 和第 29 章 “实时时钟和日历 (RTCC)”。

### 6.7.3 辅助振荡器间断操作

SOSCEM 清零时, 振荡器将只有在被选择作为当前器件时钟源 (COS2: COS0 = 100) 时才会工作。如果它是当前器件时钟源, 器件进入休眠模式时, 它将自动被禁止。

## 6.8 内部快速 RC 振荡器 (FRC)

FRC 振荡器是快速 (标称频率为 8 MHz) 内部 RC 振荡器。该振荡器用于在不使用外部晶振或陶瓷谐振器的情况下, 提供适当的器件工作速度。每当 COSC 位为 111、001 或 000 时, PIC24F 使用 FRC 的时钟工作。

### 6.8.1 使能 FRC 振荡器

因为在器件初始化期间作为参考时钟, 所以在上电复位时, FRC 振荡器始终使能。在器件经过配置和 PWRT 计时期满之后, FRC 仅在被选择作为器件时钟源时才继续工作。

### 6.8.2 FRC 后分频器模式 (FRCDIV)

如果用户希望使用快速内部振荡器作为时钟源, 他们并不是只能使用标称频率 8 MHz FRC 的输出。另一种 FRC 模式 (FRCDIV) 采用了一个可选的后分频器, 允许用户选择 7 种不同的较低时钟频率输出, 或选择直接 8 MHz 输出。后分频器使用 RCDIV2:RCDIV0 位 (CLKDIV<10:8>) 进行配置。标称频率输出为 8 MHz, 而可供选择的较低频率输出范围为 4 MHz (2 分频) 至 31 kHz (256 分频)。所提供的频率范围使用户可以随时在应用程序中通过更改 RCDIV 位来节省功耗。

每当 COSC 位为 111 时, 就将选择 FRCDIV 模式。

### 6.8.3 使用 PLL 的 FRC 振荡器模式 (FRCPLL)

FRC 后分频器的输出还可以与 4x PLL 进行组合, 以产生 16 MHz 或 32 MHz 的标称系统时钟。虽然与使用晶振或谐振器的主振荡器相比, 这样产生的频率精度较低, 但仍可使器件进行高速操作, 而无需使用外部振荡器元件。

每当 COSC 位为 001 时, 选择的即是 FRCPLL 模式。此外, 只有在选择直接或 2 分频的 FRC 后分频器选项 (RCDIV2:RCDIV0 = 000 或 001) 时, 该模式才工作。

## 6.9 内部低功耗 RC 振荡器 (LPRC)

LPRC 振荡器独立于 FRC, 它的标称振荡频率为 31 kHz。LPRC 是上电延时定时器 (PWRT)、WDT 和 FSCM 电路的时钟源。在那些有严格的功耗要求但不要求时序准确性的应用中, 也可将其用于为器件提供低频时钟源。

### 6.9.1 使能 LPRC 振荡器

由于 LPRC 振荡器用作 PWRT 时钟源, 因此只要板载稳压器被禁止, 它就会在上电复位时使能。在 PWRT 计时期满后, 只有在以下任一条件为真时, LPRC 振荡器才会保持工作:

- 故障保护时钟监视器被使能。
- WDT 被使能。
- LPRC 被选择作为系统时钟 (COSC2:COSC0 = 100)。

如果以上条件全不为真, 则 LPRC 将在 PWRT 计时期满后关闭。

## 6.10 故障保护时钟监视器 (FSCM)

故障保护时钟监视器 (FSCM) 使器件即使在出现振荡器故障时仍能继续工作。FSCM 功能通过编程配置字 2 中的 FCKSM (时钟切换和监视) 位而使能。只有在两个位都被编程时 (00)，FSCM 才会使能。当 FSCM 使能时，LPRC 内部振荡器将始终运行 (休眠模式期间除外)。

在发生振荡器故障时，FSCM 将产生时钟故障陷阱，并将系统时钟切换到 FRC 振荡器。之后，用户可以选择尝试重新启动振荡器或执行受控关闭。无论系统时钟源或振荡器模式如何，FSCM 都会对系统时钟源进行监视。这包括主振荡器的所有振荡器模式，以及辅助振荡器 SOSC (配置为系统时钟时)。

在切换到 FRC 振荡器时，FSCM 模块执行以下操作：

1. COSC 位中装入 000。
2. CF 状态位置 1，指示发生时钟故障。
3. OSWEN 控制位清零，以取消所有待执行的时钟切换。

**注：** 关于振荡器故障陷阱的更多信息，请参见第 8 章 “中断”。

### 6.10.1 FSCM 延时

在发生 POR、BOR 或从休眠模式唤醒事件时，在 FSCM 开始监视系统时钟源之前，可能会插入一个标称时长的延时 (TFSCM)。FSCM 延时的目的是确保在未使用上电延时定时器 (PWRT) 时，为振荡器和/或 PLL 提供时间来达到稳定状态。FSCM 延时将在内部系统复位信号 (SYSRST) 被释放之后产生。请参见第 7 章 “复位” 了解 FSCM 延时的时序信息。

每当 FSCM 被使能，并且 EC、HS 或 SOSC 振荡器模式被选择作为系统时钟时，都会有一个 TFSCM 时间间隔。

**注：** 请参见具体器件数据手册的 “电气特性” 一节，了解 TFSCM 规定值。

### 6.10.2 FSCM 和慢速振荡器起振

如果所选择的器件振荡器在从 POR、BOR 或休眠模式退出时的起振速度很慢，则 FSCM 延时可能会在振荡器启动之前计时期满。这种情况下，FSCM 将产生时钟故障陷阱。发生这种情况时，COSC 位中会装入 FRC 振荡器选项。这实际上会关闭正在尝试启动的原始振荡器。用户可以在陷阱服务程序中检测这种情况并启动时钟切换，切换回所需的振荡器。

### 6.10.3 FSCM 和 WDT

FSCM 和 WDT 都使用 LPRC 振荡器作为时基。在发生时钟故障时，WDT 不受到影响，并继续使用 LPRC 时钟运行。

## 6.11 时钟切换操作

在软件控制下，应用程序可以随时在 4 种时钟源（主振荡器、SOSC、FRC 和 LPRC）之间自由切换，几乎没有什么限制。为了限制这种灵活性可能带来的副作用，PIC24F 器件在切换过程中采用了保护锁定。

**注：** 主振荡器模式具有三种不同的子模式（XT、HS 和 EC），它们由 POSCMD 配置位决定。虽然在软件中，应用程序可以从其他模式切换为主振荡器模式并从主振荡器模式切换为其他模式，但它不能在不重新编程器件的情况下在不同的主振荡器子模式之间切换。

### 6.11.1 使能时钟切换

要使能时钟切换，FCKSM1 配置位必须编程为 0（请参见具体器件数据手册了解更多详细信息）。如果 FCKSM1 配置位未编程（1），则时钟切换功能和故障保护监视器功能被禁止。这是默认设置。

当时钟切换被禁止时，NOSC 控制位（OSCCON<10:8>）不控制时钟选择。不过，COSC 位（OSCCON<14:12>）将反映由 FNOSC 配置位选择的时钟源。

在时钟切换被禁止时，OSWEN 控制位（OSCCON<0>）不起作用。它始终保持为 0。

### 6.11.2 振荡器切换序列

执行时钟切换至少需要以下基本序列：

1. 根据需要读 COSC 位（OSCCON<14:12>），以确定当前的振荡器源。
2. 执行解锁序列，以允许写 OSCCON 寄存器高字节。
3. 向 NOSC 控制位（OSCCON<10:8>）写入新振荡器源的对应值。
4. 执行解锁序列，以允许写 OSCCON 寄存器的低字节。
5. 将 OSWEN 位置 1，以启动振荡器切换。

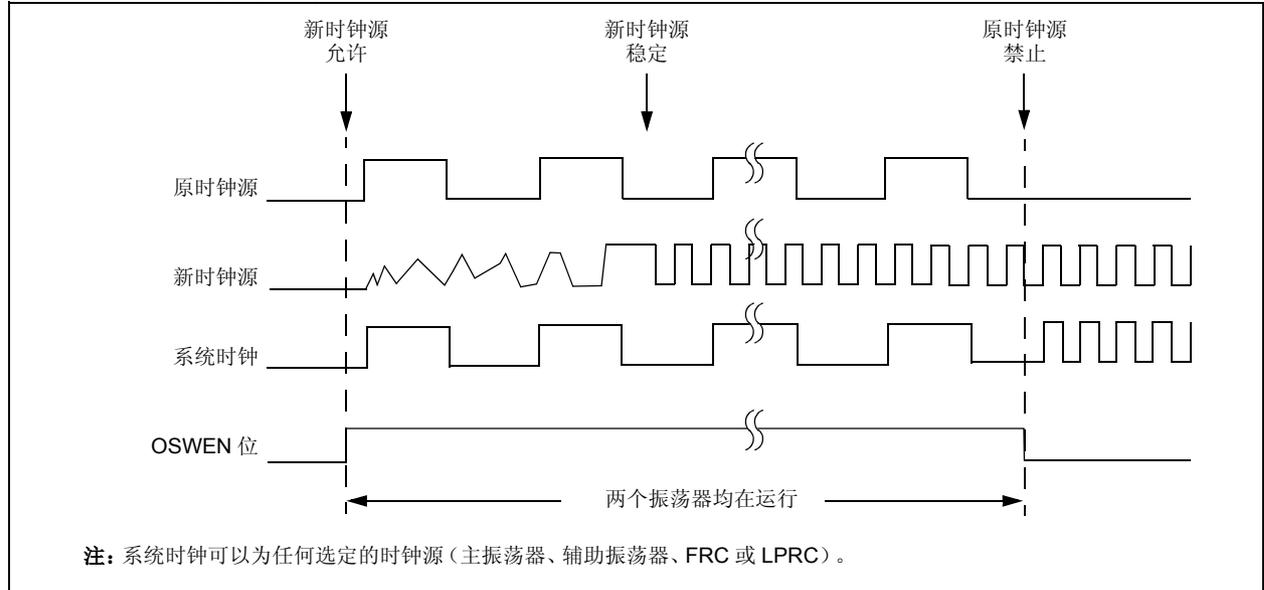
在基本序列完成后，系统时钟硬件按以下方式自动响应：

1. 时钟切换硬件将 COSC 状态位与 NOSC 控制位的新值进行比较。如果它们相同，则时钟切换是冗余操作。这种情况下，OSWEN 位自动清零，时钟切换终止。
2. 如果启动了有效的时钟切换，则 LOCK（OSCCON<5>）和 CF（OSCCON<3>）状态位清零。
3. 如果新的振荡器当前不在运行，则硬件会将其启动。如果必须要启动晶体振荡器，则硬件会等待到 OST 计时期满。如果新的振荡器源需要使用 PLL，则硬件将等待检测到 PLL 锁定（LOCK = 1）。
4. 硬件将等待新的时钟源达到稳定，然后执行时钟切换。
5. 硬件清零 OSWEN 位，指示时钟切换成功。此外，NOSC 位值被传送到 COSC 状态位。
6. 此时，原来的时钟源被关闭，LPRC（如果 WDT 或 FSCM 使能）或 SOSC（如果 SOSREN 保持置 1）除外。

图 6-7 中显示了时钟源切换的时序。

**注：** 在时钟切换序列的整个过程中，处理器将继续执行代码。在此时间期间，不应执行对时序敏感的代码。

图 6-7: 时钟转换时序图



时钟切换的建议代码序列包括以下:

1. 禁止在 OSCCON 寄存器解锁和写序列期间的中断。
2. 在两个紧接的指令中将 78h 和 9Ah 写入 OSCCON<15:8>, 以执行 OSCCON 高字节的解锁序列。
3. 在紧接解锁序列之后的指令中将新的振荡器源写入 NOSC 控制位。
4. 在两个紧接的指令中将 46h 和 57h 写入 OSCCON<7:0>, 以执行 OSCCON 低字节的解锁序列。
5. 在紧接解锁序列之后的指令中将 OSWEN 位置 1。
6. 继续执行对时钟时序要求不高的代码 (可选)。
7. 调用相应数量的软件延时 (周期计数), 以使选中的振荡器和 / 或 PLL 启动并稳定。
8. 检查 OSWEN 是否为 0。如果为 0, 则说明切换成功。

例 6-1 中显示了解锁 OSCCON 寄存器和启动时钟切换的核心序列。

例 6-1: 时钟切换的基本代码序列

```

;Place the new oscillator selection in W0
;OSCONH (high byte) Unlock Sequence
MOV     #OSCONH,w1
MOV     #0x78, w2
MOV     #0x9A, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Set new oscillator selection
MOV.b   WREG, OSCCONH
;OSCONL (low byte) unlock sequence
MOV     #OSCONL,w1
MOV.b   #0x01, w0
MOV     #0x46, w2
MOV     #0x57, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Start oscillator switch operation
MOV.b   w0, [w1]

```

## 6.11.2.1 时钟切换的注意事项

在应用程序中加入时钟切换功能时，用户在设计代码时应注意一些事项。

- **OSCCON** 解锁序列对时序的要求极高；**OSCCON** 寄存器字节仅在序列之后的一个指令周期内可写。一些高级语言（如 C 语言）在编译时可能不会保留对时序敏感的指令序列。如果以高级语言编写的应用程序需要进行时钟切换，则最好在汇编程序中编写程序，然后将其连接到应用程序中，在需要时将其作为函数进行调用。
- 如果目标时钟源是晶体振荡器，则时钟切换时间将主要是振荡器起振时间。
- 如果新时钟源不启动，或者不存在，则时钟切换硬件将无限期地等待新时钟源。用户可以检测到这种情况：**OSWEN** 位无限期地保持置 1。
- 如果新时钟源需使用 **PLL**，则只有在实现锁定之后才会发生时钟切换。用户可以检测是否脱离 **PLL** 锁定：**LOCK** 位将清零，**OSWEN** 位将置 1。
- 切换到低频时钟源（例如辅助振荡器）将导致器件工作速度非常慢。

**注：** 在使能故障保护时钟监视器时，应用程序不应尝试切换到频率低于 100 kHz 的时钟。在这些情况下进行时钟切换可能会产生错误的振荡器故障陷阱，并导致其切换到内部快速 RC 振荡器。

## 6.11.3 中止时钟切换

在时钟切换尚未完成时，可以通过清零 **OSWEN** 位而复位时钟切换逻辑。这将放弃时钟切换过程，停止并复位 **OST**（如适用），以及停止 **PLL**（如适用）。例 6-2 中显示了用于中止时钟切换的典型汇编代码。

时钟切换过程可以随时中止。当前已在执行中的时钟切换也可以通过执行第二次时钟切换而中止。

### 例 6-2: 中止时钟切换

```
MOV      #OSCCON,W1      ; pointer to OSCCON
MOV.b    #0x46,W2        ; first unlock code
MOV.b    #0x57,W3        ; second unlock code
MOV.b    W2, [W1]        ; write first unlock code
MOV.b    W3, [W1]        ; write second unlock code
BCLR     OSCCON,#OSWEN   ; ABORT the switch
```

## 6.11.4 在时钟切换期间进入休眠模式

如果器件在时钟切换操作期间进入休眠模式，则时钟切换操作被中止。处理器将保持原来的时钟选项，而 **OSWEN** 位被清零。然后，器件将正常执行 **PWRSV** 指令。

## 6.12 双速启动

双速启动是一种自动时钟切换功能，它独立于前面描述的手动受控时钟切换。该功能允许单片机在主时钟源稳定之前使用 FRC 振荡器作为时钟源，从而帮助器件最大限度地缩短从振荡器起振到代码执行之间的延时。该功能由 IESO 配置位（配置字 2<15>）控制，其操作与 FCKSM 配置位的状态无关。

双速启动功能在 FNOSC 配置位选择使用外部振荡器时非常有用，因为基于晶振的振荡器（主振荡器或辅助振荡器）的起振时间较长。作为内部 RC 振荡器，FRC 时钟源在 POR 或器件唤醒后几乎立即可用。

利用双速启动，器件在上电复位时以其默认的振荡器配置（FRC）开始执行代码。它继续在该模式工作，直到 FNOSC 配置位所指定的外部振荡器源达到稳定，这时，它将自动切换到该振荡器源。

在从省电休眠模式唤醒时，也使用双速启动功能。器件将使用 FRC 时钟源，直到选中的主时钟就绪。在空闲模式下将不使用该功能，因为器件将使用当前选定的时钟源直到主时钟源可用为止。

### 6.12.1 使用双速启动时的注意事项

当在双速启动模式中使用 FRC 振荡器时，器件仍将遵守进入省电模式的正常指令顺序，包括执行多条 PWRSAV 指令。实际上，这意味着在 OST 超时前用户代码可以改变 NOSC2:NOSC0 位的设置或执行 PWRSAV #SLEEP 指令。这就使应用程序能短暂地唤醒器件，执行“日常事务”，并在器件开始使用外部振荡器前返回休眠状态。

用户代码还可以检查 COSC2:COSC0 位与 NOSC2:NOSC0 位的状态，以确定当前提供器件时钟的时钟源。如果两组位数据匹配，则说明时钟切换已成功完成，器件正在使用所需的时钟源工作；主振荡器正在提供时钟。否则，表示当器件复位或从休眠模式被唤醒期间由 FRC 为系统提供时钟。

## 6.13 交流电气规范

图 6-8: 外部时钟时序

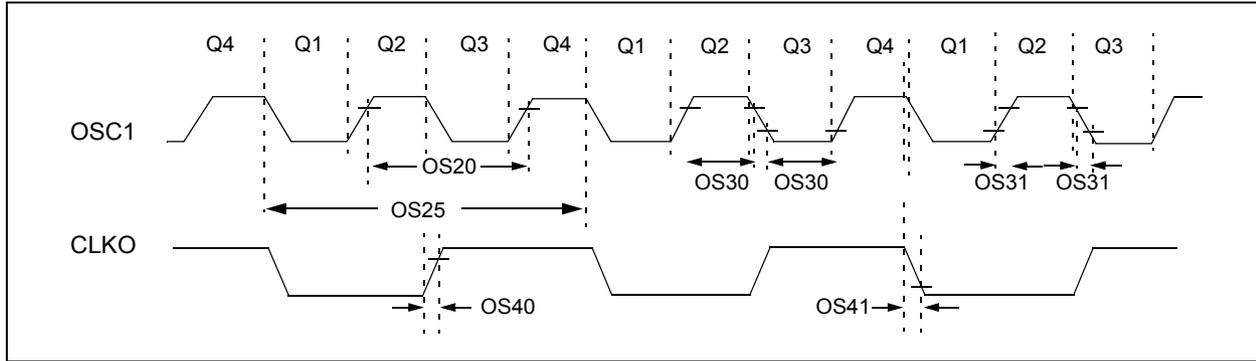


表 6-3: 外部时钟时序要求

交流特性			标准工作条件: 2.0V 到 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
OS20	Tosc	$T_{osc} = 1/F_{osc}$	—	—	—	—	
OS25	Tcy	指令周期时间 <sup>(2)</sup>	62.5	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 (OSC1) 的高电平或低电平时间	$0.45 \times T_{osc}$	—	—	ns	EC
OS31	TosR, TosF	外部时钟输入 (OSC1) 的上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 <sup>(3)</sup>	—	6	10	ns	
OS41	TckF	CLKO 下降时间 <sup>(3)</sup>	—	6	10	ns	

- 注 1: 除非另外声明, 否则“典型值”栏中的数据均为 3.3V 和 25°C 条件下的值。参数仅供设计参考, 未经测试。
- 注 2: 指令周期时间 (Tcy) 等于输入振荡器时基周期的两倍。所有规定值基于特性数据, 为器件在标准工作条件和特定振荡器类型下执行代码时的数据。超出这些规定的限定值, 可能导致振荡器运行不稳定和 / 或导致电流消耗超出预期值。所有器件在测试“最小”值时, 都在 OSC1/CLKI 引脚连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大”周期时间限制为“DC”(无时钟)。
- 注 3: 测量是在 RC 模式下进行的。CLKO 信号是在 OSC2 引脚上测量的。对于 Q1-Q2 周期 (1/2 Tcy) CLKO 为低电平, 对于 Q3-Q4 周期 (1/2 Tcy) CLKO 为高电平。

## 6.14 设计技巧

**问 1:** *在上电后使用示波器检查 OSC2 引脚时, 未检测到时钟。这是什么原因?*

**答:** 可能有几种原因:

1. 进入休眠模式后未提供唤醒方法 (如 WDT、 $\overline{\text{MCLR}}$  或中断)。请验证代码是否将器件置为休眠模式而未提供唤醒的方法。如果可能, 请尝试在  $\overline{\text{MCLR}}$  使用低电平脉冲唤醒器件。上电时  $\overline{\text{MCLR}}$  保持低电平也使晶体振荡器有更多时间起振, 但程序计数器会等到  $\overline{\text{MCLR}}$  引脚为高电平时才会递增计数。
2. 为所需的频率选择了错误的时钟模式。对于空白器件, 默认的振荡器为 FRCDIV。大多数部件在初始时使用的是在默认模式下选择的时钟, 默认模式不使用晶振或谐振器启动振荡。请验证是否已正确编程了时钟模式。
3. 未执行正确的上电序列。如果 CMOS 部件在上电之前通过 I/O 引脚供电, 则可能会发生错误的事件 (闭锁或起振不正常等)。欠压条件、起振时电源线有噪声和 VDD 上升速度太慢等情况都会导致问题。尝试在器件上电时 I/O 上不连接任何部件, 并使用确定可正常工作的快速升压电源。请参见具体器件数据手册中的上电信息, 了解关于欠压和上电序列的注意事项。
4. 连接到晶振的 C1 和 C2 电容未正确连接, 或者电容值不正确。请确定所有的连接都正确无误。这些元件使用器件数据手册中的特性值时, 通常可以使振荡器运行, 不过, 这些值可能不是对应于设计的最佳值。

**问 2:** *为什么我的器件的工作频率远高于晶振的谐振频率?*

**答:** 增益对于该振荡器电路过高。请参见第 6.5.2.4 节“确定振荡器元件的最佳值”来帮助选择 C2 (可能需要使用更大的电容)、Rs (可能需要) 和时钟模式 (可能选择了错误的模式)。对于低频晶振 (如常见的 32.768 kHz 晶振) 特别容易发生这种情况。

**问 3:** *器件运行良好, 但频率稍微偏低。如何进行调节?*

**答:** 改变 C1 的值可以对振荡器频率产生一些影响。如果使用了串联晶振, 则它的谐振频率将不同于相同频率输出的并联晶振。请确保使用的是并联晶振。

**问 4:** *我的应用程序开始时工作良好, 但之后突然退出或变慢。这是什么原因?*

**答:** 除了应当进行一般的软件检查来确定变慢原因之外, 还可以检查是否是振荡器的输出振幅不够高, 不足以可靠地触发谐振器输入。此外, 请检查 C1 和 C2 的电容值, 并确保器件配置位对于所需的振荡器模式是正确的。

**问 5:** *在我将示波器探针连接到振荡器引脚时, 我并没有看到预期的结果。这是什么原因?*

**答:** 请记住, 示波器探针也有一定的电容。将探针连接到振荡器电路会改变振荡器的特性。请考虑使用低电容 (有源) 探针。

## 6.15 寄存器映射

表 6-4 中提供了与 PIC24F 振荡器模块相关的寄存器汇总。

**表 6-4: 振荡器寄存器映射**

寄存器名称	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时
OSCCON	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	—	LOCK	—	CF	—	SOSCEN	OSWEN	xxxx <sup>(1)</sup>
CLKDIV	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	—	0300
OSCTUN	—	—	—	—	—	—	—	—	—	—	—	TUN<5:0>				0000	

**图注:** x = 复位时的未知值, — = 未实现, 读为 0。所示复位值为十六进制。

**注 1:** OSCCON 寄存器复位值取决于 FOSC 配置位和复位类型。

## 6.16 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC24F 器件系列而编写的，但其概念是相关的，通过适当修改即可使用，但在使用中可能会受到一定限制。当前与振荡器模块相关的应用笔记有：

标题	应用笔记编号
Crystal Oscillator Basics and Crystal Selection for rPIC <sup>®</sup> and PICmicro <sup>®</sup> Devices	AN826
Basic PICmicro <sup>®</sup> Oscillator Design	AN849
Practical PICmicro <sup>®</sup> Oscillator Analysis and Design	AN943
Making Your Oscillator Work	AN949

**注：** 如需获取更多 PIC24F 系列器件的应用笔记和代码示例，请访问 Microchip 网站 ([www.microchip.com](http://www.microchip.com))。

## 6.17 版本历史

### 版本 A（2006 年 9 月）

这是本文档的初始发行版。