



**MICROCHIP**

---

**PIC16(L)F1933**  
**数据手册**

带 LCD 驱动器、  
采用 nanoWatt XLP 技术的  
28 引脚 8 位 CMOS 闪存单片机

---

**请注意以下有关 Microchip 器件代码保护功能的要点:**

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信: 在正常使用的情况下, Microchip 系列产品是当今市场上同类产品中 safest 的产品之一。
- 目前, 仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知, 所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案 (Digital Millennium Copyright Act)》。如果这种行为导致他人在未经授权的情况下, 能访问您的软件或其他受版权保护的成果, 您有权依据该法案提起诉讼, 从而制止这种行为。

---

提供本文档的中文版本仅为为了便于理解。请勿忽视文档中包含的英文部分, 因为其中提供了有关 Microchip 产品性能和使用情况的有用信息。Microchip Technology Inc. 及其分公司和相关公司、各级主管与员工及事务代理机构对译文中可能存在的任何差错不承担任何责任。建议参考 Microchip Technology Inc. 的英文原版文档。

本出版物中所述的器件应用信息及其他类似内容仅为为您提供便利, 它们可能由更新之信息所替代。确保应用符合技术规范, 是您自身应负的责任。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保, 包括但不限于针对其使用情况、质量、性能、适用性或特定用途的适用性的声明或担保。Microchip 对因这些信息及使用这些信息而引起的后果不承担任何责任。如果将 Microchip 器件用于生命维持和 / 或生命安全应用, 一切风险由买方自负。买方同意在由此引发任何一切伤害、索赔、诉讼或费用时, 会维护和保障 Microchip 免于承担法律责任, 并加以赔偿。在 Microchip 知识产权保护下, 不得暗或以其他方式转让任何许可证。

#### 商标

Microchip 的名称和徽标组合、Microchip 徽标、dsPIC、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> 徽标、rfPIC 和 UNI/O 均为 Microchip Technology Inc. 在美国和其他国家或地区的注册商标。

FilterLab、Hampshire、HI-TECH C、Linear Active Thermistor、MXDEV、MXLAB、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Analog-for-the-Digital Age、Application Maestro、chipKIT、chipKIT 徽标、CodeGuard、dsPICDEM、dsPICDEM.net、dsPICworks、dsSPEAK、ECAN、ECONOMONITOR、FanSense、HI-TIDE、In-Circuit Serial Programming、ICSP、Mindi、MiWi、MPASM、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICkit、PICtail、REAL ICE、rfLAB、Select Mode、Total Endurance、TSHARC、UniWinDriver、WiperLock 和 ZENA 均为 Microchip Technology Inc. 在美国和其他国家或地区的商标。

SQTP 是 Microchip Technology Inc. 在美国的服务标记。

在此提及的所有其他商标均为各持有公司所有。

© 2011, Microchip Technology Inc. 版权所有。

ISBN: 978-1-61341-535-1

**QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
== ISO/TS 16949:2002 ==**

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。Microchip 的 PIC<sup>®</sup> MCU 与 dsPIC<sup>®</sup> DSC、KEELOQ<sup>®</sup> 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外, Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

---

---

**带 LCD 驱动器、采用 nanoWatt XLP 技术的  
28 引脚 8 位 CMOS 闪存单片机**

---

---

**本数据手册中包括的器件有：**

- PIC16F1933
- PIC16LF1933

**其他可用的 PIC16(L)F193X 器件有：**

- PIC16(L)F1934/6/7 (DS41364E\_CN)
- PIC16(L)F1938/9 (DS41574A\_CN)

**高性能 RISC CPU：**

- 只需要学习 49 条指令：
  - 除了跳转指令以外所有的指令都是单周期的
- 工作速度：
  - DC——振荡器 / 时钟的输入频率为 32 MHz
  - DC——指令周期为 125 ns
- 最多 16K x 14 字的闪存程序存储器
- 最多 1024 字节的数据存储器 (RAM)
- 带自动现场保护的中断功能
- 16 级深硬件堆栈
- 直接、间接和相对寻址模式
- 处理器对程序存储器的读访问
- 引脚布局与其他 28 引脚的 PIC16CXXX 和 PIC16FXXX 单片机兼容

**单片机的特性：**

- 高精度内部振荡器：
  - 出厂时校准为  $\pm 1\%$  (典型值)
  - 软件可选频率范围是 31 kHz 到 32 MHz
- 节能的休眠模式
- 上电复位 (Power-on Reset, POR)
- 上电延时定时器 (Power-up Timer, PWRT) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 欠压复位 (Brown-out Reset, BOR)
  - 可在 2 个跳变点之间进行选择
  - 可选择在休眠模式下禁止
- 与上拉 / 输入引脚复用的主复位
- 可编程代码保护
- 高耐用性闪存 /EEPROM 单元：
  - 闪存可经受 100,000 次写操作
  - EEPROM 可经受 1,000,000 次写操作
  - 闪存 / 数据 EEPROM 的数据保持时间: >40 年
- 宽工作电压范围：
  - 1.8V-5.5V (PIC16F1933)
  - 1.8V-3.6V (PIC16LF1933)

**PIC16LF1933 低功耗特性：**

- 待机电流：
  - 1.8V 时, 典型值为 60 nA
- 工作电流：
  - 32 kHz、1.8V 时, 典型值为 7.0  $\mu$ A (PIC16LF1933)
  - 1 MHz、1.8V 时, 典型值为 150  $\mu$ A (PIC16LF1933)
- Timer1 振荡器电流：
  - 32 kHz、1.8V 时, 典型值为 600 nA
- 低功耗看门狗定时器电流：
  - 1.8V 时, 典型值为 500 nA (PIC16LF1933)

**外设特性：**

- 最多 35 个 I/O 引脚和 1 个仅用作输入的引脚：
  - 高拉 / 灌电流可直接驱动 LED
  - 独立的可编程电平变化中断引脚
  - 独立的可编程弱上拉
- 集成的 LCD 控制器：
  - 最多 96 段
  - 可变的时钟输入
  - 对比度控制
  - 内部参考电压选择
- 电容触摸传感模块 (mTouch™)
  - 最多 16 路可选通道
- A/D 转换器：
  - 10 位分辨率并且最多 14 路通道
  - 可选择 1.024/2.048/4.096V 参考电压
- Timer0: 带 8 位可编程预分频器的 8 位定时器 / 计数器
- 增强型 Timer1
  - 专用的低功耗 32 kHz 振荡器驱动器
  - 带有预分频器的 16 位定时器 / 计数器
  - 外部门控输入模式 (具有翻转和单事件模式)
  - 门控事件完成时中断
- Timer2、4 和 6: 带 8 位周期寄存器、预分频器和后分频器的 8 位定时器 / 计数器
- 2 个捕捉 / 比较 / PWM 模块 (Capture/Compare/PWM, CCP)
  - 16 位捕捉, 最大分辨率为 125 ns
  - 16 位比较, 最大分辨率为 125 ns
  - 10 位 PWM, 最高频率为 31.25 kHz
- 3 个增强型捕捉 / 比较 / PWM 模块 (Enhanced Capture/Compare/PWM, ECCP)
  - 3 个 PWM 时基选项
  - 自动关闭和自动重启
  - PWM 控制
  - 可编程死区延时

# PIC16(L)F1933

## 外设特性（续）：

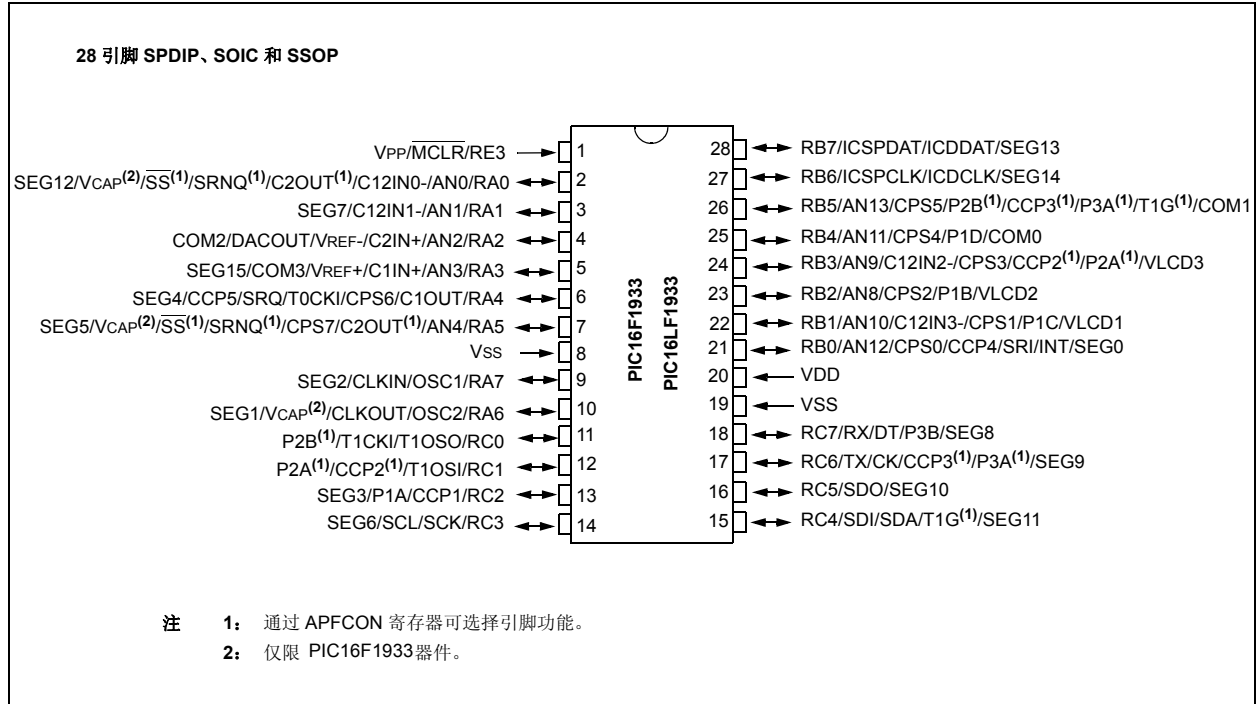
- 支持 SPI 和 I<sup>2</sup>C™ 的主同步串行端口（Master Synchronous Serial Port, MSSP）具有以下功能：
  - 7 位地址掩码
  - SMBus/PMBus™ 兼容
  - 启动时自动唤醒
- 增强型通用同步 / 异步收发器（Enhanced Universal Synchronous Asynchronous Receiver Transmitter, EUSART）
  - 与 RS-232、RS-485 和 LIN 兼容
  - 自动波特率检测
- SR 锁存器（555 定时器）：
  - 多个置 1/ 复位输入选项
  - 仿真 555 定时器应用
- 2 个比较器：
  - 轨到轨输入 / 输出
  - 功耗模式控制
  - 软件使能的迟滞
- 参考电压模块：
  - 固定参考电压（Fixed Voltage Reference, FVR）的输出级别有：1.024V、2.048V 和 4.096V
  - 具有正负参考电压选择功能的 5 位轨到轨阻式 DAC

## PIC16(L)F193X 系列类型

器件	闪存程序存储器 (字)	数据 EEPROM (字节)	SRAM (字节)	I/O	10 位 A/D (通道数)	电容触摸传感 (通道数)	比较器	8/16 位 定时器	EUSART	I <sup>2</sup> C™/SPI	ECCP	CCP	LCD
PIC16F1933 PIC16LF1933	4096	256	256	25	11	8	2	4/1	有	有	3	2	16 <sup>(1)</sup> /4

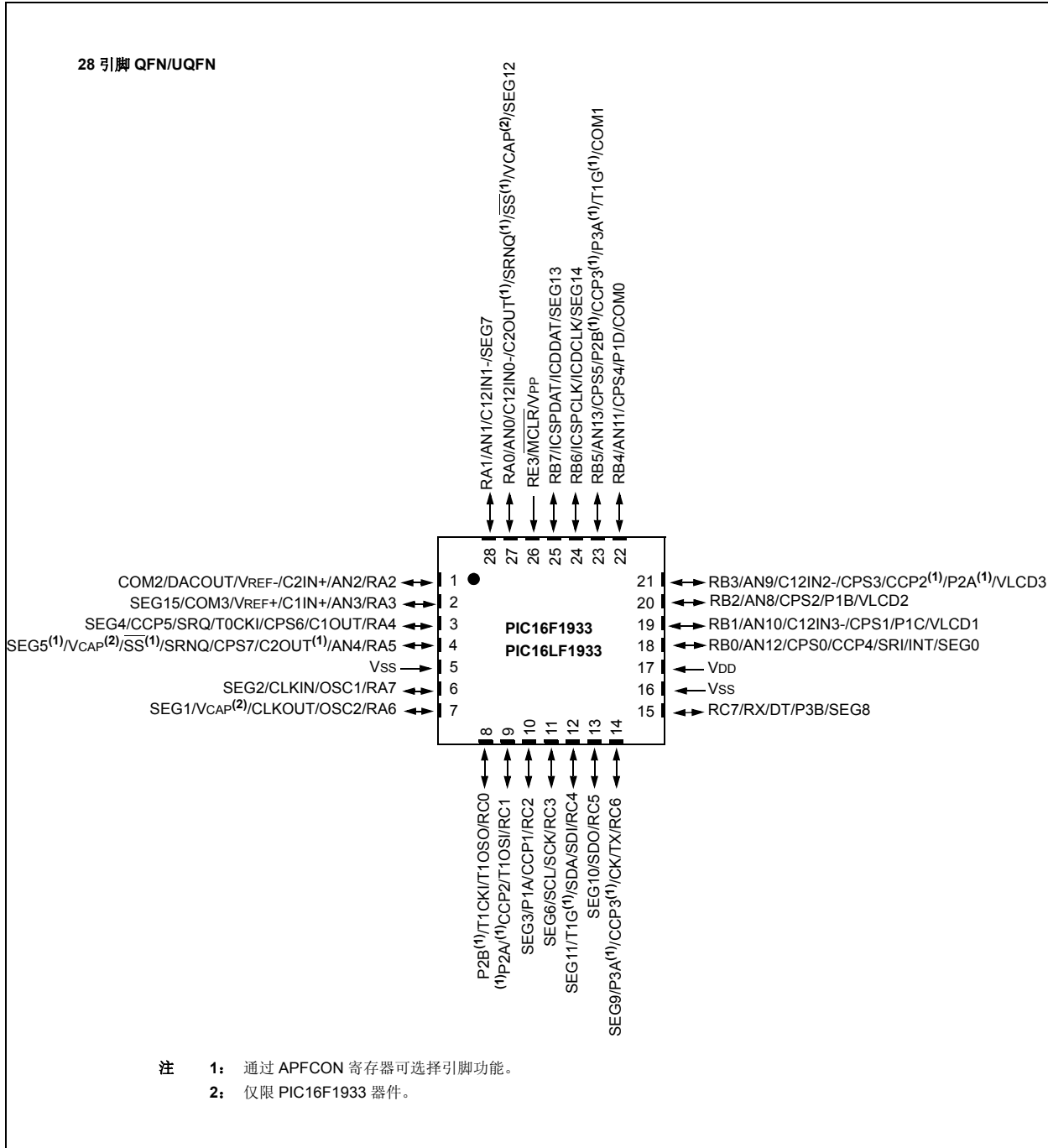
注 1： COM3 和 SEG15 共用同一物理引脚，因此，在使用 1/4 复用显示时，SEG15 不可用。

## 引脚图 —— 28 引脚 SPDIP/SOIC/SSOP (PIC16F1933 和 PIC16LF1933)



# PIC16(L)F1933

引脚图 —— 28 引脚 QFN/UQFN (PIC16F1933 和 PIC16LF1933)



# PIC16(L)F1933

表 1: 28 引脚汇总 (PIC16F1933 和 PIC16LF1933)

I/O	28 引脚 SPDIP	28 引脚 QFN/UQFN	ANSEL	A/D	电容触摸传感	比较器	SR 锁存器	定时器	CCP	EUSART	MSSP	LCD	中断	上拉	基本功能
RA0	2	27	有	AN0	—	C12IN0-/C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG12	—	—	VCAP <sup>(2)</sup>
RA1	3	28	有	AN1	—	C12IN1-	—	—	—	—	—	SEG7	—	—	—
RA2	4	1	有	AN2/VREF-	—	C2IN+/DACOUT	—	—	—	—	—	COM2	—	—	—
RA3	5	2	有	AN3/VREF+	—	C1IN+	—	—	—	—	—	SEG15/COM3	—	—	—
RA4	6	3	有	—	CPS6	C1OUT	SRQ	T0CKI	CCP5	—	—	SEG4	—	—	—
RA5	7	4	有	AN4	CPS7	C2OUT <sup>(1)</sup>	SRNQ <sup>(1)</sup>	—	—	—	SS <sup>(1)</sup>	SEG5	—	—	VCAP <sup>(2)</sup>
RA6	10	7	—	—	—	—	—	—	—	—	—	SEG1	—	—	OSC2/CLKOUT VCAP <sup>(2)</sup>
RA7	9	6	—	—	—	—	—	—	—	—	—	SEG2	—	—	OSC1/CLKIN
RB0	21	18	有	AN12	CPS0	—	SRI	—	CCP4	—	—	SEG0	INT/IO	有	—
RB1	22	19	有	AN10	CPS1	C12IN3-	—	—	P1C	—	—	VLCD1	IO	有	—
RB2	23	20	有	AN8	CPS2	—	—	—	P1B	—	—	VLCD2	IO	有	—
RB3	24	21	有	AN9	CPS3	C12IN2-	—	—	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	VLCD3	IO	有	—
RB4	25	22	有	AN11	CPS4	—	—	—	P1D	—	—	COM0	IO	有	—
RB5	26	23	有	AN13	CPS5	—	—	T1G <sup>(1)</sup>	P2B <sup>(1)</sup> / CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	—	—	COM1	IO	有	—
RB6	27	24	—	—	—	—	—	—	—	—	—	SEG14	IO	有	ICSPCLK/ ICDCLK
RB7	28	25	—	—	—	—	—	—	—	—	—	SEG13	IO	有	ICSPDAT/ ICDDAT
RC0	11	8	—	—	—	—	—	T1OSO/ T1CKI	P2B <sup>(1)</sup>	—	—	—	—	—	—
RC1	12	9	—	—	—	—	—	T1OSI	CCP2 <sup>(1)</sup> / P2A <sup>(1)</sup>	—	—	—	—	—	—
RC2	13	10	—	—	—	—	—	—	CCP1/ P1A	—	—	SEG3	—	—	—
RC3	14	11	—	—	—	—	—	—	—	—	SCK/SCL	SEG6	—	—	—
RC4	15	12	—	—	—	—	—	T1G <sup>(1)</sup>	—	—	SDI/SDA	SEG11	—	—	—
RC5	16	13	—	—	—	—	—	—	—	—	SDO	SEG10	—	—	—
RC6	17	14	—	—	—	—	—	—	CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup>	TX/CK	—	SEG9	—	—	—
RC7	18	15	—	—	—	—	—	—	P3B	RX/DT	—	SEG8	—	—	—
RE3	1	26	—	—	—	—	—	—	—	—	—	—	—	有	MCLR/VPP
VDD	20	17	—	—	—	—	—	—	—	—	—	—	—	—	VDD
VSS	8, 19	5, 16	—	—	—	—	—	—	—	—	—	—	—	—	VSS

注 1: 可使用 APFCON 寄存器改变引脚功能。  
2: 仅限 PIC16F1933 器件。

# PIC16(L)F1933

## 目录

1.0	器件概述	11
2.0	增强型中档 CPU	19
3.0	存储器构成	21
4.0	器件配置	51
5.0	振荡器模块（带故障保护时钟监视器）	57
6.0	复位	75
7.0	中断	83
8.0	低压差（LDO）稳压器	97
9.0	掉电模式（休眠）	99
10.0	看门狗定时器（WDT）	101
11.0	数据 EEPROM 和闪存程序存储器控制	105
12.0	I/O 端口	119
13.0	电平变化中断	135
14.0	固定参考电压（FVR）	139
15.0	模数转换器（ADC）模块	141
16.0	温度指示器模块	155
17.0	数模转换器（DAC）模块	157
18.0	比较器模块	161
19.0	SR 锁存器	171
20.0	Timer0 模块	175
21.0	带门控的 Timer1 模块	179
22.0	Timer2/4/6 模块	191
23.0	捕捉 / 比较 / PWM 模块	195
24.0	主同步串行端口（MSSP）模块	223
25.0	增强型通用同步 / 异步收发器（EUSART）	275
26.0	电容触摸传感（CPS）模块	305
27.0	液晶显示（LCD）驱动模块	315
28.0	在线串行编程（ICSP™）	349
29.0	指令集汇总	353
30.0	电气规范（PIC16(L)F1933）	367
31.0	直流和交流特性图表	399
32.0	开发支持	401
33.0	封装信息	405
附录 A:	数据手册版本历史	417
附录 B:	从其他 PIC® 器件移植	417
	索引	419
	Microchip 网站	427
	变更通知客户服务	427
	客户支持	427
	读者反馈表	428
	产品标识体系	429



## 致 客 户

我们旨在提供最佳文档供客户正确使用 Microchip 产品。为此，我们将不断改进出版物的内容和质量，使之更好地满足您的要求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 [CTRC@microchip.com](mailto:CTRC@microchip.com)，或将本数据手册后附的《读者反馈表》传真到 86-21-5407 5066。我们期待您的反馈。

### 最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

查看数据手册中任意一页下边角处的文献编号即可确定其版本。文献编号中数字串后的字母是版本号，例如：DS30000A是DS30000的 A 版本。

### 勘误表

现有器件可能带有一份勘误表，描述了实际运行与数据手册中记载内容之间存在的细微差异以及建议的变通方法。一旦我们了解到器件 / 文档存在某些差异时，就会发布勘误表。勘误表上将注明其所适用的硅片版本和文件版本。

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

- Microchip 网站：<http://www.microchip.com>
- 当地 Microchip 销售办事处（见最后一页）

在联络销售办事处时，请说明您所使用的器件型号、硅片版本和数据手册版本（包括文献编号）。

### 客户通知系统

欲及时获知 Microchip 产品的最新信息，请到我公司网站 [www.microchip.com](http://www.microchip.com) 上注册。

# PIC16(L)F1933

---

注:

## 1.0 器件概述

本数据手册描述了 PIC16(L)F1933 器件系列。该器件系列采用了 28 引脚封装形式。图 1-1 给出了 PIC16(L)F1933 器件的框图，而表 1-2 给出了其引脚排列说明。

每个器件的可用外设，请参见表 1-1。

**表 1-1: 器件外设汇总**

外设		PIC16F1933	PIC16LF1933
ADC		•	•
电容触摸传感模块		•	•
数模转换器 (DAC)		•	•
EUSART		•	•
固定参考电压 (FVR)		•	•
LCD		•	•
SR 锁存器		•	•
温度指示器		•	•
捕捉 / 比较 / PWM 模块			
	ECCP1	•	•
	ECCP2	•	•
	ECCP3	•	•
	CCP4	•	•
	CCP5	•	•
比较器			
	C1	•	•
	C2	•	•
主同步串行端口			
	MSSP1	•	•
定时器			
	Timer0	•	•
	Timer1	•	•
	Timer2	•	•
	Timer4	•	•
	Timer6	•	•

# PIC16(L)F1933

图 1-1: PIC16(L)F1933 框图

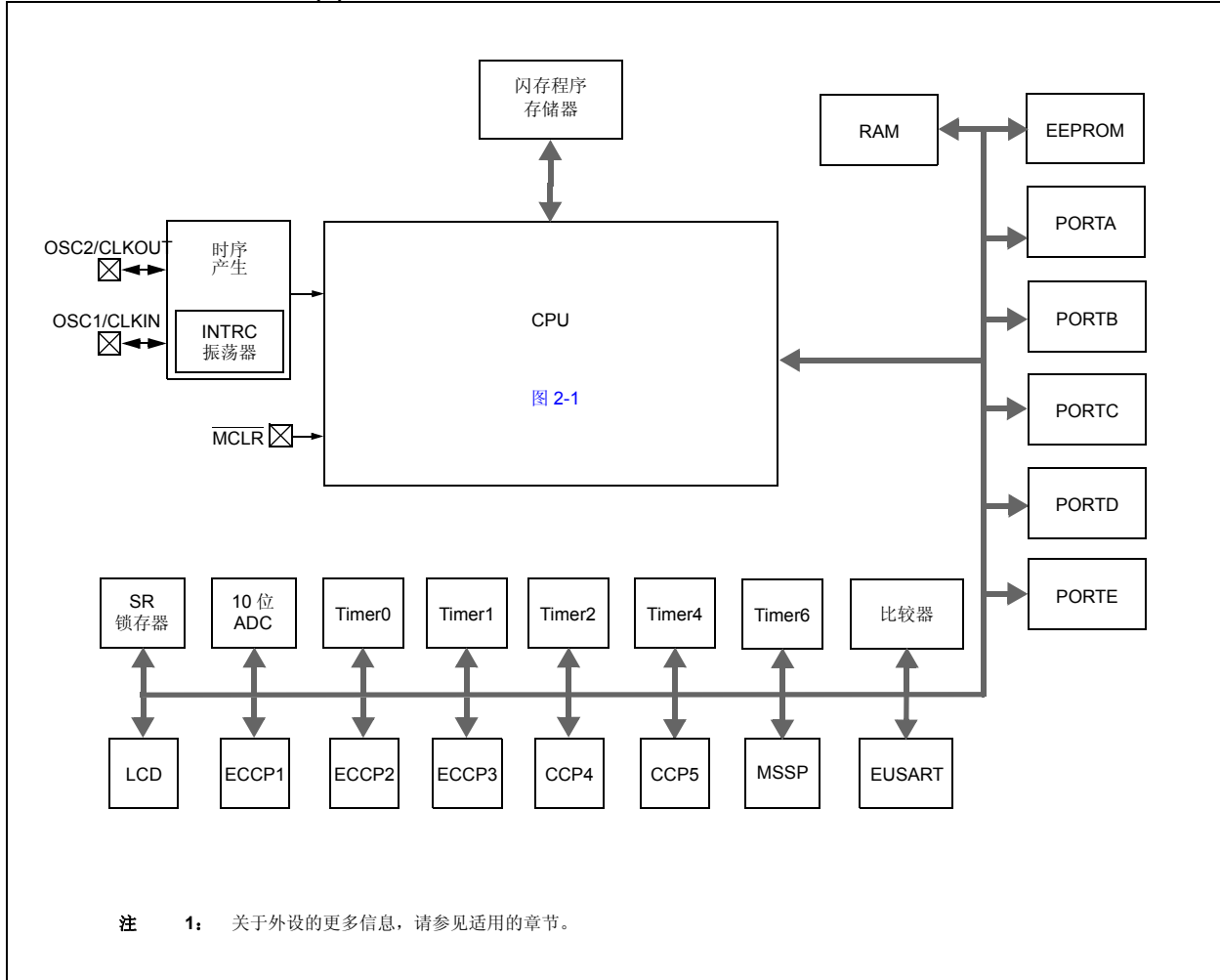


表 1-2: PIC16(L)F1933 引脚排列说明

名称	功能	输入类型	输出类型	说明
RA0/AN0/C12IN0-/C2OUT <sup>(1)</sup> /SRNQ <sup>(1)</sup> /SS <sup>(1)</sup> /VCAP <sup>(2)</sup> /SEG12	RA0	TTL	CMOS	通用 I/O。
	AN0	AN	—	A/D 输入通道 0。
	C12IN0-	AN	—	比较器 C1 或 C2 反相输入。
	C2OUT	—	CMOS	比较器 C2 输出。
	SRNQ	—	CMOS	SR 锁存器反相输出。
	SS	ST	—	从选择输入。
	VCAP	电源	电源	稳压器的滤波电容（仅限 PIC16F1933）。
SEG12	—	AN	LCD 模拟输出。	
RA1/AN1/C12IN1-/SEG7	RA1	TTL	CMOS	通用 I/O。
	AN1	AN	—	A/D 输入通道 1。
	C12IN1-	AN	—	比较器 C1 或 C2 反相输入。
	SEG7	—	AN	LCD 模拟输出。
RA2/AN2/C2IN+/VREF-/DACOUT/COM2	RA2	TTL	CMOS	通用 I/O。
	AN2	AN	—	A/D 输入通道 2。
	C2IN+	AN	—	比较器 C2 同相输入。
	VREF-	AN	—	A/D 负参考电压输入。
	DACOUT	—	AN	参考电压输出。
COM2	—	AN	LCD 模拟输出。	
RA3/AN3/C1IN+/VREF+/COM3/SEG15	RA3	TTL	CMOS	通用 I/O。
	AN3	AN	—	A/D 输入通道 3。
	C1IN+	AN	—	比较器 C1 同相输入。
	VREF+	AN	—	A/D 参考电压输入。
	COM3	—	AN	LCD 模拟输出。
SEG15	—	AN	LCD 模拟输出。	
RA4/C1OUT/CPS6/T0CKI/SRQ/CCP5/SEG4	RA4	TTL	CMOS	通用 I/O。
	C1OUT	—	CMOS	比较器 C1 输出。
	CPS6	AN	—	电容触摸传感输入 6。
	T0CKI	ST	—	Timer0 时钟输入。
	SRQ	—	CMOS	SR 锁存器同相输出。
	CCP5	ST	CMOS	捕捉 / 比较 / PWM5。
SEG4	—	AN	LCD 模拟输出。	
RA5/AN4/C2OUT <sup>(1)</sup> /CPS7/SRNQ <sup>(1)</sup> /SS <sup>(1)</sup> /VCAP <sup>(2)</sup> /SEG5	RA5	TTL	CMOS	通用 I/O。
	AN4	AN	—	A/D 输入通道 4。
	C2OUT	—	CMOS	比较器 C2 输出。
	CPS7	AN	—	电容触摸传感输入 7。
	SRNQ	—	CMOS	SR 锁存器反相输出。
	SS	ST	—	从选择输入。
	VCAP	电源	电源	稳压器的滤波电容（仅限 PIC16F1933）。
SEG5	—	AN	LCD 模拟输出。	

图注: AN = 模拟输入或输出  
TTL = TTL 兼容输入  
HV = 高电压

CMOS = CMOS 兼容输入或输出  
ST = 带 CMOS 电平的施密特触发器输入  
XTAL = 晶振

OD = 漏极开路  
I<sup>2</sup>C™ = 带 I<sup>2</sup>C 电平的施密特触发器输入

注 1: 通过 APFCON 寄存器可选择引脚功能。  
2: 仅限 PIC16F1933 器件。

# PIC16(L)F1933

表 1-2: PIC16(L)F1933 引脚排列说明 (续)

名称	功能	输入类型	输出类型	说明
RA6/OSC2/CLKOUT/VCAP <sup>(2)</sup> /SEG1	RA6	TTL	CMOS	通用 I/O。
	OSC2	—	XTAL	晶振 / 谐振器 (LP、XT 和 HS 模式)。
	CLKOUT	—	CMOS	Fosc/4 输出。
	VCAP	电源	电源	稳压器的滤波电容 (仅限 PIC16F1933)。
	SEG1	—	AN	LCD 模拟输出。
RA7/OSC1/CLKIN/SEG2	RA7	TTL	CMOS	通用 I/O。
	OSC1	XTAL	—	晶振 / 谐振器 (LP、XT 和 HS 模式)。
	CLKIN	CMOS	—	外部时钟输入 (EC 模式)。
	SEG2	—	AN	LCD 模拟输出。
RB0/AN12/CPS0/CCP4/SRI/INT/SEG0	RB0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN12	AN	—	A/D 输入通道 12。
	CPS0	AN	—	电容触摸传感输入 0。
	CCP4	ST	CMOS	捕捉 / 比较 /PWM4。
	SRI	—	ST	SR 锁存器输入。
	INT	ST	—	外部中断。
	SEG0	—	AN	LCD 模拟输出。
RB1/AN10/C12IN3-/CPS1/P1C/VLCD1	RB1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN10	AN	—	A/D 输入通道 10。
	C12IN3-	AN	—	比较器 C1 或 C2 反相输入。
	CPS1	AN	—	电容触摸传感输入 1。
	P1C	—	CMOS	PWM 输出。
	VLCD1	AN	—	LCD 模拟输入。
RB2/AN8/CPS2/P1B/VLCD2	RB2	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN8	AN	—	A/D 输入通道 8。
	CPS2	AN	—	电容触摸传感输入 2。
	P1B	—	CMOS	PWM 输出。
	VLCD2	AN	—	LCD 模拟输入。
RB3/AN9/C12IN2-/CPS3/CCP2 <sup>(1)</sup> /P2A <sup>(1)</sup> /VLCD3	RB3	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN9	AN	—	A/D 输入通道 9。
	C12IN2-	AN	—	比较器 C1 或 C2 反相输入。
	CPS3	AN	—	电容触摸传感输入 3。
	CCP2	ST	CMOS	捕捉 / 比较 /PWM2。
	P2A	—	CMOS	PWM 输出。
RB4/AN11/CPS4/P1D/COM0	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN11	AN	—	A/D 输入通道 11。
	CPS4	AN	—	电容触摸传感输入 4。
	P1D	—	CMOS	PWM 输出。
	COM0	—	AN	LCD 模拟输出。

图注: AN = 模拟输入或输出      CMOS = CMOS 兼容输入或输出      OD = 漏极开路  
TTL = TTL 兼容输入      ST = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C™ = 带 I<sup>2</sup>C 电平的施密特触发器  
HV = 高电压      XTAL = 晶振      输入

注 1: 通过 APFCON 寄存器可选择引脚功能。  
2: 仅限 PIC16F1933 器件。

**表 1-2: PIC16(L)F1933 引脚排列说明 (续)**

名称	功能	输入类型	输出类型	说明
RB5/AN13/CPS5/P2B/CCP3 <sup>(1)</sup> / P3A <sup>(1)</sup> /T1G <sup>(1)</sup> /COM1	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	AN13	AN	—	A/D 输入通道 13。
	CPS5	AN	—	电容触摸传感输入 5。
	P2B	—	CMOS	PWM 输出。
	CCP3	ST	CMOS	捕捉 / 比较 / PWM3。
	P3A	—	CMOS	PWM 输出。
	T1G	ST	—	Timer1 门控输入。
RB6/ICSPCLK/ICDCLK/SEG14	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	ICSPCLK	ST	—	串行编程时钟。
	ICDCLK	ST	—	在线调试时钟。
	SEG14	—	AN	LCD 模拟输出。
RB7/ICSPDAT/ICDDAT/SEG13	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使其上拉功能。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
	ICDDAT	ST	CMOS	在线数据 I/O。
	SEG13	—	AN	LCD 模拟输出。
RC0/T1OSO/T1CKI/P2B <sup>(1)</sup>	RC0	ST	CMOS	通用 I/O。
	T1OSO	XTAL	XTAL	Timer1 振荡器连接。
	T1CKI	ST	—	Timer1 时钟输入。
	P2B	—	CMOS	PWM 输出。
RC1/T1OSI/CCP2 <sup>(1)</sup> /P2A <sup>(1)</sup>	RC1	ST	CMOS	通用 I/O。
	T1OSI	XTAL	XTAL	Timer1 振荡器连接。
	CCP2	ST	CMOS	捕捉 / 比较 / PWM2。
	P2A	—	CMOS	PWM 输出。
RC2/CCP1/P1A/SEG3	RC2	ST	CMOS	通用 I/O。
	CCP1	ST	CMOS	捕捉 / 比较 / PWM1。
	P1A	—	CMOS	PWM 输出。
	SEG3	—	AN	LCD 模拟输出。
RC3/SCK/SCL/SEG6	RC3	ST	CMOS	通用 I/O。
	SCK	ST	CMOS	SPI 时钟。
	SCL	I <sup>2</sup> C	OD	I <sup>2</sup> C™ 时钟。
	SEG6	—	AN	LCD 模拟输出。
RC4/SDI/SDA/T1G <sup>(1)</sup> /SEG11	RC4	ST	CMOS	通用 I/O。
	SDI	ST	—	SPI 数据输入。
	SDA	I <sup>2</sup> C	OD	I <sup>2</sup> C™ 数据输入 / 输出。
	T1G	ST	—	Timer1 门控输入。
	SEG11	—	AN	LCD 模拟输出。
RC5/SDO/SEG10	RC5	ST	CMOS	通用 I/O。
	SDO	—	CMOS	SPI 数据输出。
	SEG10	—	AN	LCD 模拟输出。

**图注:** AN = 模拟输入或输出  
TTL = TTL 兼容输入  
HV = 高电压

CMOS = CMOS 兼容输入或输出  
ST = 带 CMOS 电平的施密特触发器输入  
XTAL = 晶振

OD = 漏极开路  
I<sup>2</sup>C™ = 带 I<sup>2</sup>C 电平的施密特触发器输入

**注** 1: 通过 APFCON 寄存器可选择引脚功能。  
2: 仅限 PIC16F1933 器件。

# PIC16(L)F1933

表 1-2: PIC16(L)F1933 引脚排列说明 (续)

名称	功能	输入类型	输出类型	说明
RC6/TX/CK/CCP3/P3A/SEG9	RC6	ST	CMOS	通用 I/O。
	TX	—	CMOS	USART 异步发送。
	CK	ST	CMOS	USART 同步时钟。
	CCP3	ST	CMOS	捕捉 / 比较 / PWM3。
	P3A	—	CMOS	PWM 输出。
	SEG9	—	AN	LCD 模拟输出。
RC7/RX/DT/P3B/SEG8	RC7	ST	CMOS	通用 I/O。
	RX	ST	—	USART 异步输入。
	DT	ST	CMOS	USART 同步数据。
	P3B	—	CMOS	PWM 输出。
	SEG8	—	AN	LCD 模拟输出。
RE3/MCLR/VPP	RE3	TTL	—	通用输入。
	MCLR	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
VDD	VDD	Power	—	正电源电压。
VSS	VSS	Power	—	参考地。

图注: AN = 模拟输入或输出      CMOS = CMOS 兼容输入或输出      OD = 漏极开路  
 TTL = TTL 兼容输入      ST = 带 CMOS 电平的施密特触发器输入      I<sup>2</sup>C™ = 带 I<sup>2</sup>C 电平的施密特触发器  
 HV = 高电压      XTAL = 晶振      输入

注 1: 通过 APFCON 寄存器可选择引脚功能。  
 2: 仅限 PIC16F1933 器件。



## 2.0 增强型中档 CPU

此器件系列包含了一个增强型中档 8 位 CPU 内核。该 CPU 具有 49 条指令。中断功能包括自动现场保护。硬件堆栈有 16 级深，还具有上溢复位功能和下溢复位功能。提供直接寻址、间接寻址和相对寻址模式。2 个文件选择寄存器（File Select Register, FSR）提供了读程序存储器和数据存储器的功能。

- 自动中断现场保护
- 具有上溢和下溢功能的 16 级堆栈
- 文件选择寄存器
- 指令集

### 2.1 自动中断现场保护

中断期间，某些寄存器的内容自动保存在影子寄存器中，并在从中断返回时自动恢复，从而节省了堆栈空间和用户代码。更多信息，请参见第 7.5 节“自动现场保护”。

### 2.2 具有上溢和下溢功能的 16 级堆栈

这些器件的外部堆栈存储器有 15 位宽、16 字深。如果使能了堆栈上溢或堆栈下溢，那么在发生溢出事件时，PCON 寄存器中的相应位（STKOVF 或 STKUNF）将会置 1，并会导致软件复位。更多详细信息，请参见第 3.4 节“堆栈”。

### 2.3 文件选择寄存器

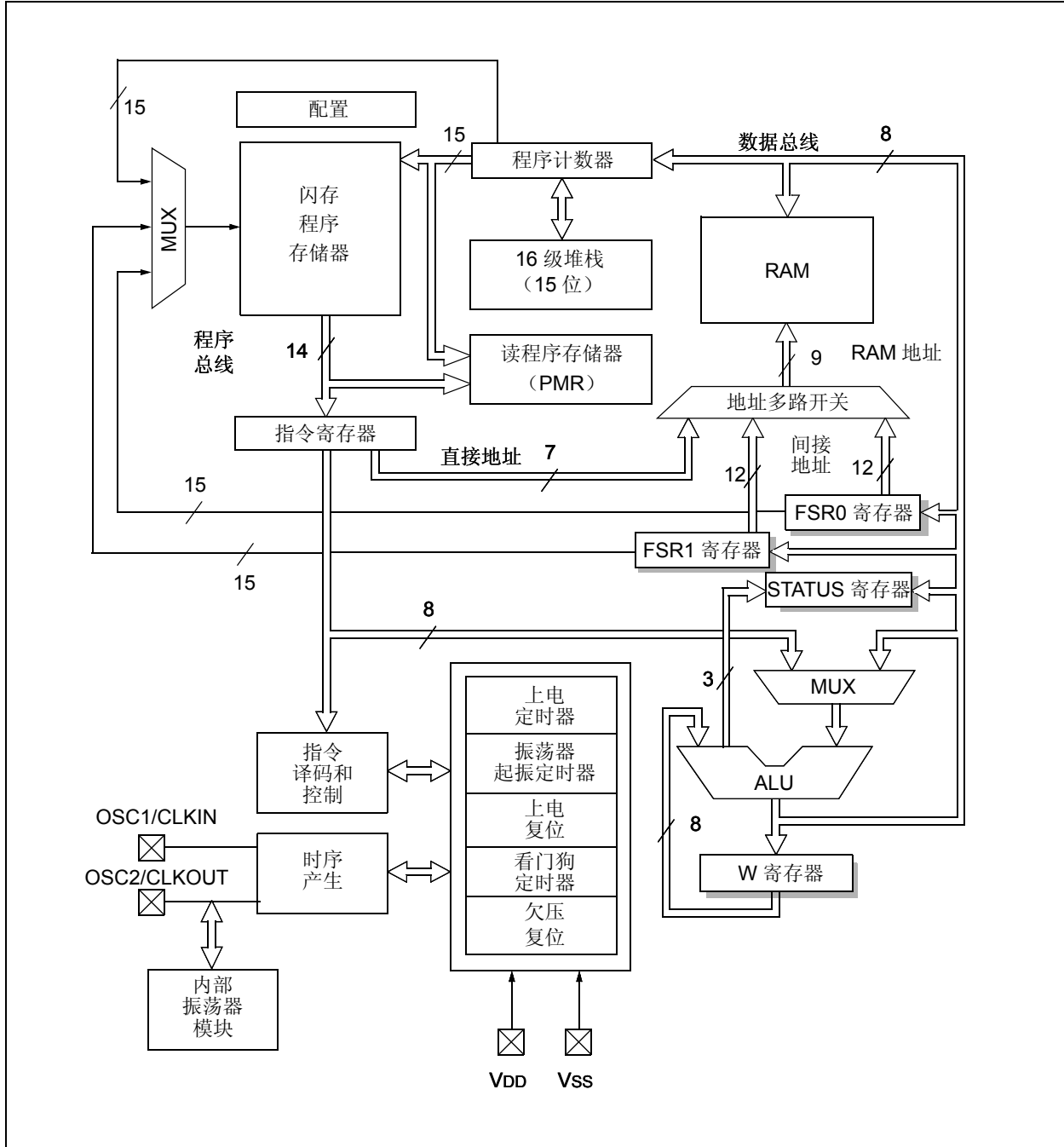
有 2 个 16 位文件选择寄存器（FSR）。FSR 可访问所有的文件寄存器和程序存储器，从而允许对所有存储器使用一个数据指针。当 FSR 指向程序存储器时，在使用 INDF 的指令中提供了 1 个额外的指令周期以完成取数据的操作。现在可以线性寻址通用存储器，来访问 80 字节以上的连续数据。还提供了新的指令来支持 FSR。更多详细信息，请参见第 3.5 节“间接寻址”。

### 2.4 指令集

增强型中档 CPU 提供了 49 条指令来支持 CPU 的功能。更多详细信息，请参见第 29.0 节“指令集汇总”。

# PIC16(L)F1933

图 2-1: 内核框图



## 3.0 存储器构成

PIC16(L)F1933 器件中有三种类型的存储器：数据存储器、程序存储器和数据 EEPROM 存储器<sup>(1)</sup>。

- 程序存储器
- 数据存储器
  - 内核寄存器
  - 特殊功能寄存器
  - 通用 RAM
  - 公共 RAM
  - 器件存储器映射
  - 特殊功能寄存器汇总
- 数据 EEPROM 存储器<sup>(1)</sup>

**注 1：**第 11.0 节“数据 EEPROM 和闪存程序存储器控制”中描述了数据 EEPROM 存储器和通过 EECON 寄存器访问闪存存储器的方法。

以下功能与程序存储器和数据存储器的访问和控制有关：

- PCL 和 PCLATH
- 堆栈
- 间接寻址

## 3.1 程序存储器构成

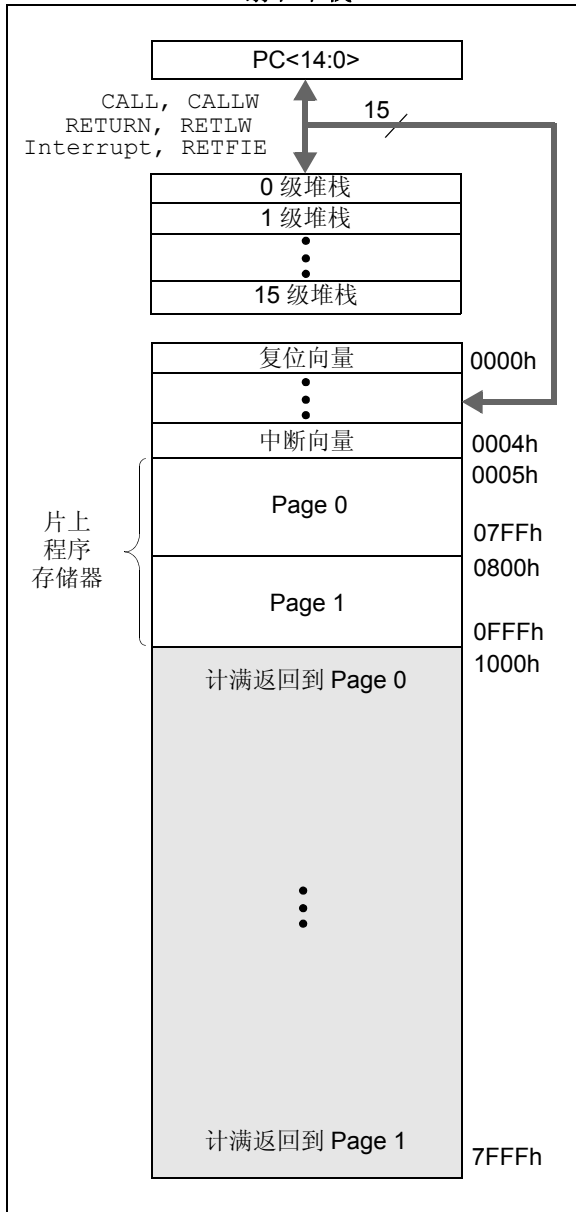
这个增强型中档内核具有一个 15 位程序计数器，可对 32K x 14 的程序存储空间进行寻址。表 3-1 给出了 PIC16(L)F1933 器件系列所能实现的存储容量。访问超出地址边界的单元将导致操作返回到所实现的存储空间内。复位向量地址为 0000h，中断向量地址为 0004h（见图 3-1）。

表 3-1: 器件程序存储器大小和地址

器件	程序存储空间（字）	程序存储器的最后一个地址
PIC16(L)F1933	4,096	0FFFh

# PIC16(L)F1933

图 3-1: 4 KW 器件的程序存储器映射和堆栈



## 3.1.1 将程序存储器当作数据存储器读取

有两种方法可访问程序存储器中的常数。第一种方法是使用 RETLW 指令表。第二种方法是设置 FSR 指向程序存储器。

### 3.1.1.1 RETLW 指令

RETLW 指令用于提供对常数表的访问。例 3-1 给出了创建这种表的推荐方法。

例 3-1: RETLW 指令

```
constants
    BRW                ;Add Index in W to
                       ;program counter to
                       ;select data
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3

my_function
    ;... LOTS OF CODE...
    MOVLW    DATA_INDEX
    call constants
    ;... THE CONSTANT IS IN W
```

BRW 指令使得这种类型的表实现起来非常简单。如果代码必须保持与前几代单片机之间的可移植性，则 BRW 指令不适用，因此必须使用较老的表读取方法。

### 3.1.1.2 使用 FSR 间接读取

可将程序存储器当作数据存储器进行访问，方法是将对 FSRxH 寄存器的 bit 7 置 1 并读取与之配对的 INDFx 寄存器。MOVIW 指令会将已寻址到的字的低 8 位保存在 W 寄存器中。无法通过 INDF 寄存器执行写程序存储器操作。通过 FSR 访问程序存储器的指令需要一个额外的指令周期才能完成。例 3-2 演示了通过 FSR 访问程序存储器的过程。

如果标号指向程序存储器中的单元，HIGH 伪指令将 bit<7> 置 1。

#### 例 3-2: 通过 FSR 访问程序存储器

```
constants
    RETLW DATA0      ;Index0 data
    RETLW DATA1      ;Index1 data
    RETLW DATA2
    RETLW DATA3
my_function
;... LOTS OF CODE...
    MOVLW LOW constants
    MOVWF FSR1L
    MOVLW HIGH constants
    MOVWF FSR1H
    MOVIW 0[FSR1]
;THE PROGRAM MEMORY IS IN W
```

### 3.2.1 内核寄存器

内核寄存器包含直接影响 PIC16(L)F1933 基本操作的寄存器。这些寄存器如下所列：

- INDF0
- INDF1
- PCL
- STATUS
- FSR0 低字节
- FSR0 高字节
- FSR1 低字节
- FSR1 高字节
- BSR
- WREG
- PCLATH
- INTCON

**注：** 内核寄存器位于数据存储器每个存储区的前 12 个地址单元。

## 3.2 数据存储器构成

数据存储器可划分为 32 个存储区，每个存储区 128 个字节。每个存储区包含（图 3-2）：

- 12 个内核寄存器
- 20 个特殊功能寄存器（Special Function Register, SFR）
- 最多 80 字节的通用 RAM（General Purpose RAM, GPR）
- 16 个公共 RAM

可通过将存储区号写入存储区选择寄存器（Bank Select Register, BSR）来选择有效存储区。未实现的存储器将读为 0。所有的数据存储器都可以直接访问（通过使用文件寄存器的指令），或通过 2 个文件选择寄存器（FSR）间接访问。更多信息，请参见第 3.5 节“间接寻址”。

# PIC16(L)F1933

## 3.2.1.1 STATUS 寄存器

如寄存器 3-1 所示，STATUS 寄存器包含：

- ALU 的算术运算状态
- 复位状态

和任何其他寄存器一样，STATUS 寄存器也可以作为任何指令的目标寄存器。如果 STATUS 寄存器是影响 Z、DC 或 C 位的指令的目标寄存器，那么将禁止对这 3 位进行写操作。根据器件逻辑，这些位会被置 1 或清零。此外，TO 和 PD 位也是不可写的。因此，当执行一条把 STATUS 寄存器作为目标寄存器的指令后，结果可能和预想的不一样。

例如，CLRF STATUS 会清零高 3 位，将 Z 位置 1，从而使得 STATUS 寄存器的内容变为 000u u1uu（其中 u = 不变）。

因此，建议仅使用 BCF、BSF、SWAPF 和 MOVWF 指令来改变 STATUS 寄存器的内容，因为这些指令不影响任何状态位。如需了解其他不影响状态位的指令，请参见第 29.0 节“指令集汇总”。

**注 1:** 在减法运算中，C 和 DC 分别作为借位和半借位。

**寄存器 3-1: STATUS: STATUS 寄存器**

U-0	U-0	U-0	R-1/q	R-1/q	R/W-0/u	R/W-0/u	R/W-0/u
—	—	—	TO	PD	Z	DC <sup>(1)</sup>	C <sup>(1)</sup>
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 取值视条件而定

- bit 7-5      **未实现:** 读为 0
- bit 4      **TO:** 超时标志位  
1 = 在上电复位、执行了 CLRWDT 指令或 SLEEP 指令后  
0 = 发生了 WDT 超时
- bit 3      **PD:** 掉电标志位  
1 = 上电复位或执行了 CLRWDT 指令后  
0 = 执行了 SLEEP 指令
- bit 2      **Z:** 零标志位  
1 = 算术运算或逻辑运算结果为零  
0 = 算术运算或逻辑运算结果不为零
- bit 1      **DC:** 半进位 / 半借位标志位 (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) <sup>(1)</sup>  
1 = 结果的第 4 个低位向高位发生了进位  
0 = 结果的第 4 个低位未向高位发生进位
- bit 0      **C:** 进位 / 借位标志位 <sup>(1)</sup> (ADDWF、ADDLW、SUBLW 和 SUBWF 指令) <sup>(1)</sup>  
1 = 结果的最高位发生了进位  
0 = 结果的最高位未发生进位

**注 1:** 对于借位，极性是相反的。减法指令通过加上第二个操作数的二进制补码实现。对于移位指令 (RRF 和 RLF)，此位值来自源寄存器的最高位或最低位。

### 3.2.2 特殊功能寄存器

特殊功能寄存器是应用程序用来控制器件中外设功能所需操作的寄存器。与外设操作有关的寄存器将在本数据手册中的相应外设章节中讲述。

### 3.2.3 通用 RAM

数据存储器的每个存储区中的 GPR 最多有 80 个字节。

#### 3.2.3.1 GPR 的线性访问

可以通过 FSR 以非分区方法访问通用 RAM。这可以简化对大存储器结构的访问。更多信息，请参见第 3.5.2 节“线性数据存储器”。

### 3.2.4 公共 RAM

从所有存储区都可以访问 16 字节的公共 RAM。

### 3.2.5 器件存储器映射

此器件系列的存储器映射如表 3-2 所示。

表 3-2: 存储器映射表

器件	存储区	表号
PIC16F1933	0-7	表 3-3
PIC16LF1933	8-15	表 3-4, 表 3-7
	16-23	表 3-5
	23-31	表 3-6, 表 3-8

图 3-2: 分区存储器划分

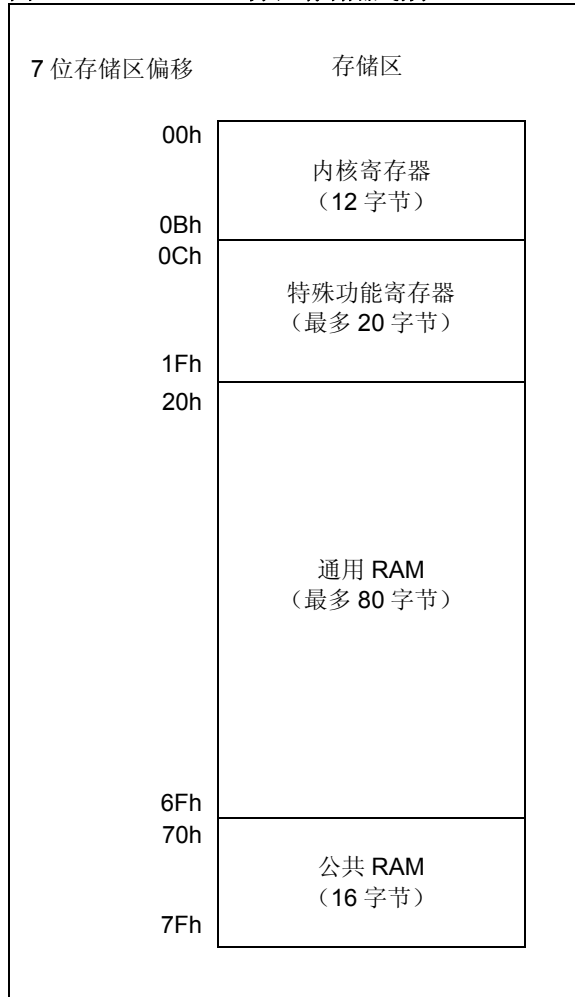


表 3-3: PIC16(L)F1933 存储器映射, BANK 0-7

BANK 0		BANK 1		BANK 2		BANK 3		BANK 4		BANK 5		BANK 6		BANK 7	
000h	INDF0	080h	INDF0	100h	INDF0	180h	INDF0	200h	INDF0	280h	INDF0	300h	INDF0	380h	INDF0
001h	INDF1	081h	INDF1	101h	INDF1	181h	INDF1	201h	INDF1	281h	INDF1	301h	INDF1	381h	INDF1
002h	PCL	082h	PCL	102h	PCL	182h	PCL	202h	PCL	282h	PCL	302h	PCL	382h	PCL
003h	STATUS	083h	STATUS	103h	STATUS	183h	STATUS	203h	STATUS	283h	STATUS	303h	STATUS	383h	STATUS
004h	FSR0L	084h	FSR0L	104h	FSR0L	184h	FSR0L	204h	FSR0L	284h	FSR0L	304h	FSR0L	384h	FSR0L
005h	FSR0H	085h	FSR0H	105h	FSR0H	185h	FSR0H	205h	FSR0H	285h	FSR0H	305h	FSR0H	385h	FSR0H
006h	FSR1L	086h	FSR1L	106h	FSR1L	186h	FSR1L	206h	FSR1L	286h	FSR1L	306h	FSR1L	386h	FSR1L
007h	FSR1H	087h	FSR1H	107h	FSR1H	187h	FSR1H	207h	FSR1H	287h	FSR1H	307h	FSR1H	387h	FSR1H
008h	BSR	088h	BSR	108h	BSR	188h	BSR	208h	BSR	288h	BSR	308h	BSR	388h	BSR
009h	WREG	089h	WREG	109h	WREG	189h	WREG	209h	WREG	289h	WREG	309h	WREG	389h	WREG
00Ah	PCLATH	08Ah	PCLATH	10Ah	PCLATH	18Ah	PCLATH	20Ah	PCLATH	28Ah	PCLATH	30Ah	PCLATH	38Ah	PCLATH
00Bh	INTCON	08Bh	INTCON	10Bh	INTCON	18Bh	INTCON	20Bh	INTCON	28Bh	INTCON	30Bh	INTCON	38Bh	INTCON
00Ch	PORTA	08Ch	TRISA	10Ch	LATA	18Ch	ANSELA	20Ch	—	28Ch	—	30Ch	—	38Ch	—
00Dh	PORTB	08Dh	TRISB	10Dh	LATB	18Dh	ANSELB	20Dh	WPUB	28Dh	—	30Dh	—	38Dh	—
00Eh	PORTC	08Eh	TRISC	10Eh	LATC	18Eh	—	20Eh	—	28Eh	—	30Eh	—	38Eh	—
00Fh	—	08Fh	—	10Fh	—	18Fh	—	20Fh	—	28Fh	—	30Fh	—	38Fh	—
010h	PORTE	090h	TRISE	110h	—	190h	—	210h	WPUE	290h	—	310h	—	390h	—
011h	PIR1	091h	PIE1	111h	CM1CON0	191h	EEADRL	211h	SSPBUF	291h	CCPR1L	311h	CCPR3L	391h	—
012h	PIR2	092h	PIE2	112h	CM1CON1	192h	EEADRH	212h	SSPADD	292h	CCPR1H	312h	CCPR3H	392h	—
013h	PIR3	093h	PIE3	113h	CM2CON0	193h	EEDATL	213h	SSPMSK	293h	CCP1CON	313h	CCP3CON	393h	—
014h	—	094h	—	114h	CM2CON1	194h	EEDATH	214h	SSPSTAT	294h	PWM1CON	314h	PWM3CON	394h	IOCBP
015h	TMR0	095h	OPTION_REG	115h	CMOUT	195h	EECON1	215h	SSPCON1	295h	CCP1AS	315h	CCP3AS	395h	IOCBN
016h	TMR1L	096h	PCON	116h	BORCON	196h	EECON2	216h	SSPCON2	296h	PSTR1CON	316h	PSTR3CON	396h	IOCBF
017h	TMR1H	097h	WDTCON	117h	FVRCON	197h	—	217h	SSPCON3	297h	—	317h	—	397h	—
018h	T1CON	098h	OSCTUNE	118h	DACCON0	198h	—	218h	—	298h	CCPR2L	318h	CCPR4L	398h	—
019h	T1GCON	099h	OSCCON	119h	DACCON1	199h	RCREG	219h	—	299h	CCPR2H	319h	CCPR4H	399h	—
01Ah	TMR2	09Ah	OSCSTAT	11Ah	SRCON0	19Ah	TXREG	21Ah	—	29Ah	CCP2CON	31Ah	CCP4CON	39Ah	—
01Bh	PR2	09Bh	ADRESL	11Bh	SRCON1	19Bh	SPBRGL	21Bh	—	29Bh	PWM2CON	31Bh	—	39Bh	—
01Ch	T2CON	09Ch	ADRESH	11Ch	—	19Ch	SPBRGH	21Ch	—	29Ch	CCP2AS	31Ch	CCPR5L	39Ch	—
01Dh	—	09Dh	ADCON0	11Dh	APFCON	19Dh	RCSTA	21Dh	—	29Dh	PSTR2CON	31Dh	CCPR5H	39Dh	—
01Eh	CPSCON0	09Eh	ADCON1	11Eh	—	19Eh	TXSTA	21Eh	—	29Eh	CCPTMRS0	31Eh	CCP5CON	39Eh	—
01Fh	CPSCON1	09Fh	—	11Fh	—	19Fh	BAUDCTR	21Fh	—	29Fh	CCPTMRS1	31Fh	—	39Fh	—
020h	—	0A0h	—	120h	—	1A0h	—	220h	—	2A0h	—	320h	—	3A0h	—
06Fh	通用寄存器 96 字节	0EFh	通用寄存器 80 字节	16Fh	通用寄存器 80 字节	1EFh	未实现 读为 0	26Fh	未实现 读为 0	2EFh	未实现 读为 0	36Fh	未实现 读为 0	3EFh	未实现 读为 0
070h	—	0F0h	快速操作存储区 70h - 7Fh	170h	快速操作存储区 70h - 7Fh	1F0h	快速操作存储区 70h - 7Fh	270h	快速操作存储区 70h - 7Fh	2F0h	快速操作存储区 70h - 7Fh	370h	快速操作存储区 70h - 7Fh	3F0h	快速操作存储区 70h - 7Fh
07Fh	—	0FFh	—	17Fh	—	1FFh	—	27Fh	—	2FFh	—	37Fh	—	3FFh	—

图注: ■ = 未实现的数据存储单元, 读为 0。



表 3-4: PIC16(L)F1933 存储器映射, BANK 8-15

BANK 8		BANK 9		BANK 10		BANK 11		BANK 12		BANK 13		BANK 14		BANK 15	
400h	INDF0	480h	INDF0	500h	INDF0	580h	INDF0	600h	INDF0	680h	INDF0	700h	INDF0	780h	INDF0
401h	INDF1	481h	INDF1	501h	INDF1	581h	INDF1	601h	INDF1	681h	INDF1	701h	INDF1	781h	INDF1
402h	PCL	482h	PCL	502h	PCL	582h	PCL	602h	PCL	682h	PCL	702h	PCL	782h	PCL
403h	STATUS	483h	STATUS	503h	STATUS	583h	STATUS	603h	STATUS	683h	STATUS	703h	STATUS	783h	STATUS
404h	FSR0L	484h	FSR0L	504h	FSR0L	584h	FSR0L	604h	FSR0L	684h	FSR0L	704h	FSR0L	784h	FSR0L
405h	FSR0H	485h	FSR0H	505h	FSR0H	585h	FSR0H	605h	FSR0H	685h	FSR0H	705h	FSR0H	785h	FSR0H
406h	FSR1L	486h	FSR1L	506h	FSR1L	586h	FSR1L	606h	FSR1L	686h	FSR1L	706h	FSR1L	786h	FSR1L
407h	FSR1H	487h	FSR1H	507h	FSR1H	587h	FSR1H	607h	FSR1H	687h	FSR1H	707h	FSR1H	787h	FSR1H
408h	BSR	488h	BSR	508h	BSR	588h	BSR	608h	BSR	688h	BSR	708h	BSR	788h	BSR
409h	WREG	489h	WREG	509h	WREG	589h	WREG	609h	WREG	689h	WREG	709h	WREG	789h	WREG
40Ah	PCLATH	48Ah	PCLATH	50Ah	PCLATH	58Ah	PCLATH	60Ah	PCLATH	68Ah	PCLATH	70Ah	PCLATH	78Ah	PCLATH
40Bh	INTCON	48Bh	INTCON	50Bh	INTCON	58Bh	INTCON	60Bh	INTCON	68Bh	INTCON	70Bh	INTCON	78Bh	INTCON
40Ch	—	48Ch	—	50Ch	—	58Ch	—	60Ch	—	68Ch	—	70Ch	—	78Ch	—
40Dh	—	48Dh	—	50Dh	—	58Dh	—	60Dh	—	68Dh	—	70Dh	—	78Dh	—
40Eh	—	48Eh	—	50Eh	—	58Eh	—	60Eh	—	68Eh	—	70Eh	—	78Eh	—
40Fh	—	48Fh	—	50Fh	—	58Fh	—	60Fh	—	68Fh	—	70Fh	—	78Fh	—
410h	—	490h	—	510h	—	590h	—	610h	—	690h	—	710h	—	790h	—
411h	—	491h	—	511h	—	591h	—	611h	—	691h	—	711h	—	791h	—
412h	—	492h	—	512h	—	592h	—	612h	—	692h	—	712h	—	792h	—
413h	—	493h	—	513h	—	593h	—	613h	—	693h	—	713h	—	793h	—
414h	—	494h	—	514h	—	594h	—	614h	—	694h	—	714h	—	794h	—
415h	TMR4	495h	—	515h	—	595h	—	615h	—	695h	—	715h	—	795h	—
416h	PR4	496h	—	516h	—	596h	—	616h	—	696h	—	716h	—	796h	—
417h	T4CON	497h	—	517h	—	597h	—	617h	—	697h	—	717h	—	797h	—
418h	—	498h	—	518h	—	598h	—	618h	—	698h	—	718h	—	798h	—
419h	—	499h	—	519h	—	599h	—	619h	—	699h	—	719h	—	799h	—
41Ah	—	49Ah	—	51Ah	—	59Ah	—	61Ah	—	69Ah	—	71Ah	—	79Ah	—
41Bh	—	49Bh	—	51Bh	—	59Bh	—	61Bh	—	69Bh	—	71Bh	—	79Bh	—
41Ch	TMR6	49Ch	—	51Ch	—	59Ch	—	61Ch	—	69Ch	—	71Ch	—	79Ch	—
41Dh	PR6	49Dh	—	51Dh	—	59Dh	—	61Dh	—	69Dh	—	71Dh	—	79Dh	—
41Eh	T6CON	49Eh	—	51Eh	—	59Eh	—	61Eh	—	69Eh	—	71Eh	—	79Eh	—
41Fh	—	49Fh	—	51Fh	—	59Fh	—	61Fh	—	69Fh	—	71Fh	—	79Fh	—
420h	未实现 读为 0	4A0h	未实现 读为 0	520h	未实现 读为 0	5A0h	未实现 读为 0	620h	未实现 读为 0	6A0h	未实现 读为 0	720h	未实现 读为 0	7A0h	未实现 读为 0
46Fh	—	4EFh	—	56Fh	—	5EFh	—	66Fh	—	6EFh	—	76Fh	—	7EFh	—
470h	快速操作存储区 70h - 7Fh	4F0h	快速操作存储区 70h - 7Fh	570h	快速操作存储区 70h - 7Fh	5F0h	快速操作存储区 70h - 7Fh	670h	快速操作存储区 70h - 7Fh	6F0h	快速操作存储区 70h - 7Fh	770h	快速操作存储区 70h - 7Fh	7F0h	快速操作存储区 70h - 7Fh
47Fh	—	4FFh	—	57Fh	—	5FFh	—	67Fh	—	6FFh	—	77Fh	—	7FFh	—

图注: ■ = 未实现数据存储单元, 读为 0。

见表 3-7

表 3-5: PIC16(L)F1933 存储器映射, BANK 16-23

BANK 16		BANK 17		BANK 18		BANK 19		BANK 20		BANK 21		BANK 22		BANK 23	
800h	INDF0	880h	INDF0	900h	INDF0	980h	INDF0	A00h	INDF0	A80h	INDF0	B00h	INDF0	B80h	INDF0
801h	INDF1	881h	INDF1	901h	INDF1	981h	INDF1	A01h	INDF1	A81h	INDF1	B01h	INDF1	B81h	INDF1
802h	PCL	882h	PCL	902h	PCL	982h	PCL	A02h	PCL	A82h	PCL	B02h	PCL	B82h	PCL
803h	STATUS	883h	STATUS	903h	STATUS	983h	STATUS	A03h	STATUS	A83h	STATUS	B03h	STATUS	B83h	STATUS
804h	FSR0L	884h	FSR0L	904h	FSR0L	984h	FSR0L	A04h	FSR0L	A84h	FSR0L	B04h	FSR0L	B84h	FSR0L
805h	FSR0H	885h	FSR0H	905h	FSR0H	985h	FSR0H	A05h	FSR0H	A85h	FSR0H	B05h	FSR0H	B85h	FSR0H
806h	FSR1L	886h	FSR1L	906h	FSR1L	986h	FSR1L	A06h	FSR1L	A86h	FSR1L	B06h	FSR1L	B86h	FSR1L
807h	FSR1H	887h	FSR1H	907h	FSR1H	987h	FSR1H	A07h	FSR1H	A87h	FSR1H	B07h	FSR1H	B87h	FSR1H
808h	BSR	888h	BSR	908h	BSR	988h	BSR	A08h	BSR	A88h	BSR	B08h	BSR	B88h	BSR
809h	WREG	889h	WREG	909h	WREG	989h	WREG	A09h	WREG	A89h	WREG	B09h	WREG	B89h	WREG
80Ah	PCLATH	88Ah	PCLATH	90Ah	PCLATH	98Ah	PCLATH	A0Ah	PCLATH	A8Ah	PCLATH	B0Ah	PCLATH	B8Ah	PCLATH
80Bh	INTCON	88Bh	INTCON	90Bh	INTCON	98Bh	INTCON	A0Bh	INTCON	A8Bh	INTCON	B0Bh	INTCON	B8Bh	INTCON
80Ch	—	88Ch	—	90Ch	—	98Ch	—	A0Ch	—	A8Ch	—	B0Ch	—	B8Ch	—
80Dh	—	88Dh	—	90Dh	—	98Dh	—	A0Dh	—	A8Dh	—	B0Dh	—	B8Dh	—
80Eh	—	88Eh	—	90Eh	—	98Eh	—	A0Eh	—	A8Eh	—	B0Eh	—	B8Eh	—
80Fh	—	88Fh	—	90Fh	—	98Fh	—	A0Fh	—	A8Fh	—	B0Fh	—	B8Fh	—
810h	—	890h	—	910h	—	990h	—	A10h	—	A90h	—	B10h	—	B90h	—
811h	—	891h	—	911h	—	991h	—	A11h	—	A91h	—	B11h	—	B91h	—
812h	—	892h	—	912h	—	992h	—	A12h	—	A92h	—	B12h	—	B92h	—
813h	—	893h	—	913h	—	993h	—	A13h	—	A93h	—	B13h	—	B93h	—
814h	—	894h	—	914h	—	994h	—	A14h	—	A94h	—	B14h	—	B94h	—
815h	—	895h	—	915h	—	995h	—	A15h	—	A95h	—	B15h	—	B95h	—
816h	—	896h	—	916h	—	996h	—	A16h	—	A96h	—	B16h	—	B96h	—
817h	—	897h	—	917h	—	997h	—	A17h	—	A97h	—	B17h	—	B97h	—
818h	—	898h	—	918h	—	998h	—	A18h	—	A98h	—	B18h	—	B98h	—
819h	—	899h	—	919h	—	999h	—	A19h	—	A99h	—	B19h	—	B99h	—
81Ah	—	89Ah	—	91Ah	—	99Ah	—	A1Ah	—	A9Ah	—	B1Ah	—	B9Ah	—
81Bh	—	89Bh	—	91Bh	—	99Bh	—	A1Bh	—	A9Bh	—	B1Bh	—	B9Bh	—
81Ch	—	89Ch	—	91Ch	—	99Ch	—	A1Ch	—	A9Ch	—	B1Ch	—	B9Ch	—
81Dh	—	89Dh	—	91Dh	—	99Dh	—	A1Dh	—	A9Dh	—	B1Dh	—	B9Dh	—
81Eh	—	89Eh	—	91Eh	—	99Eh	—	A1Eh	—	A9Eh	—	B1Eh	—	B9Eh	—
81Fh	—	89Fh	—	91Fh	—	99Fh	—	A1Fh	—	A9Fh	—	B1Fh	—	B9Fh	—
820h	未实现 读为 0	8A0h	未实现 读为 0	920h	未实现 读为 0	9A0h	未实现 读为 0	A20h	未实现 读为 0	AA0h	未实现 读为 0	B20h	未实现 读为 0	BA0h	未实现 读为 0
86Fh	快速操作存储区 70h - 7Fh	8EFh	快速操作存储区 70h - 7Fh	96Fh	快速操作存储区 70h - 7Fh	9EFh	快速操作存储区 70h - 7Fh	A6Fh	快速操作存储区 70h - 7Fh	AEFh	快速操作存储区 70h - 7Fh	B6Fh	快速操作存储区 70h - 7Fh	BEFh	快速操作存储区 70h - 7Fh
870h	快速操作存储区 70h - 7Fh	8F0h	快速操作存储区 70h - 7Fh	970h	快速操作存储区 70h - 7Fh	9F0h	快速操作存储区 70h - 7Fh	A70h	快速操作存储区 70h - 7Fh	AF0h	快速操作存储区 70h - 7Fh	B70h	快速操作存储区 70h - 7Fh	BF0h	快速操作存储区 70h - 7Fh
87Fh	快速操作存储区 70h - 7Fh	8FFh	快速操作存储区 70h - 7Fh	97Fh	快速操作存储区 70h - 7Fh	9FFh	快速操作存储区 70h - 7Fh	A7Fh	快速操作存储区 70h - 7Fh	AFFh	快速操作存储区 70h - 7Fh	B7Fh	快速操作存储区 70h - 7Fh	BFFh	快速操作存储区 70h - 7Fh

图注: ■ = 未实现数据存储单元, 读为 0。

表 3-6: PIC16(L)F1933 存储器映射, BANK 24-31

BANK 24		BANK 25		BANK 26		BANK 27		BANK 28		BANK 29		BANK 30		BANK 31	
C00h	INDF0	C80h	INDF0	D00h	INDF0	D80h	INDF0	E00h	INDF0	E80h	INDF0	F00h	INDF0	F80h	INDF0
C01h	INDF1	C81h	INDF1	D01h	INDF1	D81h	INDF1	E01h	INDF1	E81h	INDF1	F01h	INDF1	F81h	INDF1
C02h	PCL	C82h	PCL	D02h	PCL	D82h	PCL	E02h	PCL	E82h	PCL	F02h	PCL	F82h	PCL
C03h	STATUS	C83h	STATUS	D03h	STATUS	D83h	STATUS	E03h	STATUS	E83h	STATUS	F03h	STATUS	F83h	STATUS
C04h	FSR0L	C84h	FSR0L	D04h	FSR0L	D84h	FSR0L	E04h	FSR0L	E84h	FSR0L	F04h	FSR0L	F84h	FSR0L
C05h	FSR0H	C85h	FSR0H	D05h	FSR0H	D85h	FSR0H	E05h	FSR0H	E85h	FSR0H	F05h	FSR0H	F85h	FSR0H
C06h	FSR1L	C86h	FSR1L	D06h	FSR1L	D86h	FSR1L	E06h	FSR1L	E86h	FSR1L	F06h	FSR1L	F86h	FSR1L
C07h	FSR1H	C87h	FSR1H	D07h	FSR1H	D87h	FSR1H	E07h	FSR1H	E87h	FSR1H	F07h	FSR1H	F87h	FSR1H
C08h	BSR	C88h	BSR	D08h	BSR	D88h	BSR	E08h	BSR	E88h	BSR	F08h	BSR	F88h	BSR
C09h	WREG	C89h	WREG	D09h	WREG	D89h	WREG	E09h	WREG	E89h	WREG	F09h	WREG	F89h	WREG
C0Ah	PCLATH	C8Ah	PCLATH	D0Ah	PCLATH	D8Ah	PCLATH	E0Ah	PCLATH	E8Ah	PCLATH	F0Ah	PCLATH	F8Ah	PCLATH
C0Bh	INTCON	C8Bh	INTCON	D0Bh	INTCON	D8Bh	INTCON	E0Bh	INTCON	E8Bh	INTCON	F0Bh	INTCON	F8Bh	INTCON
C0Ch	—	C8Ch	—	D0Ch	—	D8Ch	—	E0Ch	—	E8Ch	—	F0Ch	—	F8Ch	见表 3-8
C0Dh	—	C8Dh	—	D0Dh	—	D8Dh	—	E0Dh	—	E8Dh	—	F0Dh	—	F8Dh	
C0Eh	—	C8Eh	—	D0Eh	—	D8Eh	—	E0Eh	—	E8Eh	—	F0Eh	—	F8Eh	
C0Fh	—	C8Fh	—	D0Fh	—	D8Fh	—	E0Fh	—	E8Fh	—	F0Fh	—	F8Fh	
C10h	—	C90h	—	D10h	—	D90h	—	E10h	—	E90h	—	F10h	—	F90h	
C11h	—	C91h	—	D11h	—	D91h	—	E11h	—	E91h	—	F11h	—	F91h	
C12h	—	C92h	—	D12h	—	D92h	—	E12h	—	E92h	—	F12h	—	F92h	
C13h	—	C93h	—	D13h	—	D93h	—	E13h	—	E93h	—	F13h	—	F93h	
C14h	—	C94h	—	D14h	—	D94h	—	E14h	—	E94h	—	F14h	—	F94h	
C15h	—	C95h	—	D15h	—	D95h	—	E15h	—	E95h	—	F15h	—	F95h	
C16h	—	C96h	—	D16h	—	D96h	—	E16h	—	E96h	—	F16h	—	F96h	
C17h	—	C97h	—	D17h	—	D97h	—	E17h	—	E97h	—	F17h	—	F97h	
C18h	—	C98h	—	D18h	—	D98h	—	E18h	—	E98h	—	F18h	—	F98h	
C19h	—	C99h	—	D19h	—	D99h	—	E19h	—	E99h	—	F19h	—	F99h	
C1Ah	—	C9Ah	—	D1Ah	—	D9Ah	—	E1Ah	—	E9Ah	—	F1Ah	—	F9Ah	
C1Bh	—	C9Bh	—	D1Bh	—	D9Bh	—	E1Bh	—	E9Bh	—	F1Bh	—	F9Bh	
C1Ch	—	C9Ch	—	D1Ch	—	D9Ch	—	E1Ch	—	E9Ch	—	F1Ch	—	F9Ch	
C1Dh	—	C9Dh	—	D1Dh	—	D9Dh	—	E1Dh	—	E9Dh	—	F1Dh	—	F9Dh	
C1Eh	—	C9Eh	—	D1Eh	—	D9Eh	—	E1Eh	—	E9Eh	—	F1Eh	—	F9Eh	
C1Fh	—	C9Fh	—	D1Fh	—	D9Fh	—	E1Fh	—	E9Fh	—	F1Fh	—	F9Fh	
C20h	—	CA0h	—	D20h	—	DA0h	—	E20h	—	EA0h	—	F20h	—	FA0h	
	未实现 读为 0		未实现 读为 0		未实现 读为 0		未实现 读为 0		未实现 读为 0		未实现 读为 0		未实现 读为 0		
C6Fh	—	CEFh	—	D6Fh	—	DEFh	—	E6Fh	—	EEFh	—	F6Fh	—	FEFh	
C70h	快速操作存储区 70h - 7Fh	CF0h	快速操作存储区 70h - 7Fh	D70h	快速操作存储区 70h - 7Fh	DF0h	快速操作存储区 70h - 7Fh	E70h	快速操作存储区 70h - 7Fh	EF0h	快速操作存储区 70h - 7Fh	F70h	快速操作存储区 70h - 7Fh	FF0h	快速操作存储区 70h - 7Fh
CFFh	—	CFFh	—	D7Fh	—	DFFh	—	E7Fh	—	EFFh	—	F7Fh	—	FFFh	

图注: ■ = 未实现数据存储单元, 读为 0。

# PIC16(L)F1933

表 3-7: PIC16(L)F1933 存储器映射, BANK 15

Bank 15	
791h	LCDCON
792h	LCDPS
793h	LCDREF
794h	LCDCST
795h	LCDRL
796h	—
797h	—
798h	LCDSE0
799h	LCDSE1
79Ah	—
79Bh	—
79Ch	—
79Dh	—
79Eh	—
79Fh	—
7A0h	LCDDATA0
7A1h	LCDDATA1
7A2h	—
7A3h	LCDDATA3
7A4h	LCDDATA4
7A5h	—
7A6h	LCDDATA6
7A7h	LCDDATA7
7A8h	—
7A9h	LCDDATA9
7AAh	LCDDATA10
7ABh	—
7ACh	—
7ADh	—
7AEh	—
7AFh	—
7B0h	—
7B1h	—
7B2h	—
7B3h	—
7B4h	—
7B5h	—
7B6h	—
7B7h	—
7B8h	—
	未实现 读为 0
7EFh	—

图注:  = 未实现数据存储单元, 读为 0。

表 3-8: PIC16(L)F1933 存储器映射, BANK 31

Bank 31	
F8Ch	未实现 读为 0
FE3h	
FE4h	STATUS_SHAD
FE5h	WREG_SHAD
FE6h	BSR_SHAD
FE7h	PCLATH_SHAD
FE8h	FSR0L_SHAD
FE9h	FSR0H_SHAD
FEAh	FSR1L_SHAD
FEBh	FSR1H_SHAD
FECh	—
FEDh	STKPTR
FEEh	TOSL
FEFh	TOSH

图注:  = 未实现数据存储单元, 读为 0。

### 3.2.6 特殊功能寄存器汇总

该器件系列的特殊功能寄存器汇总如下:

器件	存储区	页码
PIC16(L)F1933	0	29
	1	30
	2	31
	3	32
	4	33
	5	34
	6	35
	7	36
	8	37
	9-14	38
	15	39
	16-30	40
	31	41

**表 3-9: 特殊功能寄存器汇总**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 0</b>												
000h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
001h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
002h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
003h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
004h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
005h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
006h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
007h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
008h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>				---	0 0000	---	0 0000
009h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
00Ah <sup>(1, 2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
00Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
00Ch	PORTA	写 PORTA 寄存器写的是 PORTA 数据锁存器; 读 PORTA 寄存器读的是 PORTA 引脚								xxxx xxxx	uuuu uuuu	
00Dh	PORTB	写 PORTB 寄存器写的是 PORTB 数据锁存器; 读 PORTB 寄存器读的是 PORTB 引脚								xxxx xxxx	uuuu uuuu	
00Eh	PORTC	写 PORTC 寄存器写的是 PORTC 数据锁存器; 读 PORTC 寄存器读的是 PORTC 引脚								xxxx xxxx	uuuu uuuu	
00Fh	—	未实现								—	—	
010h	PORTE	—	—	—	—	RE3	—	—	—	---- x---	---- u---	
011h	PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000	
012h	PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	0000 00-0	0000 00-0	
013h	PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	-000 0-0-	-000 0-0-	
014h	—	未实现								—	—	
015h	TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu	
016h	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu	
017h	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu	
018h	T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{T1SYNC}$	—	TMR1ON	0000 00-0	uuuu uu-u	
019h	T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		0000 0x00	uuuu uxuu	
01Ah	TMR2	Timer 2 模块寄存器								0000 0000	0000 0000	
01Bh	PR2	Timer 2 周期寄存器								1111 1111	1111 1111	
01Ch	T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		-000 0000	-000 0000	
01Dh	—	未实现								—	—	
01Eh	CPSCON0	CPSON	CPSRM	—	—	CPSRNG1	CPSRNG0	CPSOUT	T0XCS	00-- 0000	00-- 0000	
01Fh	CPSCON1	—	—	—	—	—	CPSCH<2:0>		----	-000	----	-000

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注**
- 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。
  - 2: 从任何存储区都可以寻址到这些寄存器。
  - 3: 未实现, 读为 1。

# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 1</b>												
080h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
081h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
082h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
083h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
084h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
085h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
086h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
087h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
088h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>			—	—	---0 0000	---0 0000	
089h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
08Ah <sup>(1, 2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
08Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
08Ch	TRISA	PORTA 数据方向寄存器								1111 1111	1111 1111	
08Dh	TRISB	PORTB 数据方向寄存器								1111 1111	1111 1111	
08Eh	TRISC	PORTC 数据方向寄存器								1111 1111	1111 1111	
08Fh	—	未实现								—	—	
090h	TRISE	—	—	—	—	— <sup>(3)</sup>	—	—	—	---- 1---	---- 1---	
091h	PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000	
092h	PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	0000 00-0	0000 00-0	
093h	PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	-000 0-0-	-000 0-0-	
094h	—	未实现								—	—	
095h	OPTION_REG	WPUEN	INTEDG	TMROCS	TMROSE	PSA	PS<2:0>			1111 1111	1111 1111	
096h	PCON	STKOVF	STKUNF	—	—	$\overline{RMCLR}$	$\overline{RI}$	$\overline{POR}$	$\overline{BOR}$	00-- 11qq	qq-- qquu	
097h	WDTCON	—	—	WDTPS<4:0>					SWDTEN	--01 0110	--01 0110	
098h	OSCTUNE	—	—	TUN<5:0>					—	--00 0000	--00 0000	
099h	OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		0011 1-00	0011 1-00	
09Ah	OSCSTAT	T1OSCR	PLLRL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	00q0 0q0-	qqqq qq0-	
09Bh	ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	uuuu uuuu	
09Ch	ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	uuuu uuuu	
09Dh	ADCON0	—	CHS<4:0>					GO/DONE	ADON	-000 0000	-000 0000	
09Eh	ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF1	ADPREF0	0000 -000	0000 -000	
09Fh	—	未实现								—	—	

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 2</b>												
100h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
101h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
102h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
103h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
104h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
105h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
106h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
107h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
108h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
109h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
10Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
10Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
10Ch	LATA	PORTA 数据锁存器								xxxx xxxx	uuuu uuuu	
10Dh	LATB	PORTB 数据锁存器								xxxx xxxx	uuuu uuuu	
10Eh	LATC	PORTC 数据锁存器								xxxx xxxx	uuuu uuuu	
10Fh	—	未实现								—	—	
110h	—	未实现								—	—	
111h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	0000 -100	0000 -100	
112h	CM1CON1	C1INTP	C1INTN	C1PCH1	C1PCH0	—	—	C1NCH<1:0>		0000 --00	0000 --00	
113h	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	0000 -100	0000 -100	
114h	CM2CON1	C2INTP	C2INTN	C2PCH1	C2PCH0	—	—	C2NCH<1:0>		0000 --00	0000 --00	
115h	CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	---- --00	---- --00	
116h	BORCON	SBOREN	—	—	—	—	—	—	BORRDY	1--- ---q	u--- ---u	
117h	FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR1	CDAFVR0	ADFVR<1:0>		0q00 0000	0q00 0000	
118h	DACCON0	DACEN	DACLPS	DACOE	---	DACPSS<1:0>		---	DACNSS	000- 00-0	000- 00-0	
119h	DACCON1	---	---	---	DACR<4:0>					---0 0000	---0 0000	
11Ah	SRCON0	SRLEN	SRCLK2	SRCLK1	SRCLK0	SRQEN	SRNQEN	SRPS	SRPR	0000 0000	0000 0000	
11Bh	SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E	0000 0000	0000 0000	
11Ch	—	未实现								—	—	
11Dh	APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	-000 0000	-000 0000	
11Eh	—	未实现								—	—	
11Fh	—	未实现								—	—	

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注** 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值		
<b>Bank 3</b>													
180h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx		
181h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx		
182h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000		
183h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu		
184h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu		
185h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000		
186h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu		
187h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000		
188h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000		
189h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu		
18Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000	
18Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000		
18Ch	ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	--11 1111	--11 1111		
18Dh	ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	--11 1111	--11 1111		
18Eh	—	未实现								—	—		
18Fh	—	未实现								—	—		
190h	—	未实现								—	—		
191h	EEADRL	EEPROM / 程序存储器地址寄存器低字节								0000 0000	0000 0000		
192h	EEADRH	—	EEPROM / 程序存储器地址寄存器高字节								-000 0000	-000 0000	
193h	EEDATL	EEPROM / 程序存储器读数据寄存器低字节								xxxx xxxx	uuuu uuuu		
194h	EEDATH	—	—	EEPROM / 程序存储器读数据寄存器高字节								--xx xxxx	--uu uuuu
195h	EECON1	EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	0000 x000	0000 q000		
196h	EECON2	EEPROM 控制寄存器 2								0000 0000	0000 0000		
197h	—	未实现								—	—		
198h	—	未实现								—	—		
199h	RCREG	USART 接收数据寄存器								0000 0000	0000 0000		
19Ah	TXREG	USART 发送数据寄存器								0000 0000	0000 0000		
19Bh	SPBRGL	BRG<7:0>								0000 0000	0000 0000		
19Ch	SPBRGH	BRG<15:8>								0000 0000	0000 0000		
19Dh	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x		
19Eh	TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	0000 0010	0000 0010		
19Fh	BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00		

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。



# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值		
<b>Bank 4</b>													
200h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx		
201h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx		
202h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000		
203h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu		
204h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu		
205h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000		
206h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu		
207h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000		
208h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---	0 0000	---	0 0000
209h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu		
20Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000	
20Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIF	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000		
20Ch	—	未实现								—	—		
20Dh	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	1111 1111	1111 1111		
20Eh	—	未实现								—	—		
20Fh	—	未实现								—	—		
210h	WPUE	—	—	—	—	WPUE3	—	—	—	---- 1----	---- 1----		
211h	SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								xxxx xxxx	uuuu uuuu		
212h	SSPADD	ADD<7:0>								0000 0000	0000 0000		
213h	SSPMSK	MSK<7:0>								1111 1111	1111 1111		
214h	SSPSTAT	SMP	CKE	$\overline{D/A}$	P	S	$\overline{R/W}$	UA	BF	0000 0000	0000 0000		
215h	SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				0000 0000	0000 0000		
216h	SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0000 0000	0000 0000		
217h	SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	0000 0000	0000 0000		
218h	—	未实现								—	—		
219h	—	未实现								—	—		
21Ah	—	未实现								—	—		
21Bh	—	未实现								—	—		
21Ch	—	未实现								—	—		
21Dh	—	未实现								—	—		
21Eh	—	未实现								—	—		
21Fh	—	未实现								—	—		

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 5</b>												
280h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
281h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
282h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
283h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
284h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
285h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
286h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
287h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
288h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
289h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
28Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
28Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
28Ch	—	未实现								—	—	
28Dh	—	未实现								—	—	
28Eh	—	未实现								—	—	
28Fh	—	未实现								—	—	
290h	—	未实现								—	—	
291h	CCPR1L	捕捉 / 比较 / PWM 寄存器 1 (LSB)								xxxx xxxx	uuuu uuuu	
292h	CCPR1H	捕捉 / 比较 / PWM 寄存器 1 (MSB)								xxxx xxxx	uuuu uuuu	
293h	CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				0000 0000	0000 0000	
294h	PWM1CON	P1RSEN	P1DC<6:0>								0000 0000	0000 0000
295h	CCP1AS	CCP1ASE	CCP1AS<2:0>			PSS1AC<1:0>		PSS1BD<1:0>		0000 0000	0000 0000	
296h	PSTR1CON	—	—	—	STR1SYNC	STR1D	STR1C	STR1B	STR1A	---0 0001	---0 0001	
297h	—	未实现								—	—	
298h	CCPR2L	捕捉 / 比较 / PWM 寄存器 2 (LSB)								xxxx xxxx	uuuu uuuu	
299h	CCPR2H	捕捉 / 比较 / PWM 寄存器 2 (MSB)								xxxx xxxx	uuuu uuuu	
29Ah	CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				0000 0000	0000 0000	
29Bh	PWM2CON	P2RSEN	P2DC<6:0>								0000 0000	0000 0000
29Ch	CCP2AS	CCP2ASE	CCP2AS<2:0>			PSS2AC<1:0>		PSS2BD<1:0>		0000 0000	0000 0000	
29Dh	PSTR2CON	—	—	—	STR2SYNC	STR2D	STR2C	STR2B	STR2A	---0 0001	---0 0001	
29Eh	CCPTMRS0	C4TSEL1	C4TSEL0	C3TSEL1	C3TSEL0	C2TSEL1	C2TSEL0	C1TSEL1	C1TSEL0	0000 0000	0000 0000	
29Fh	CCPTMRS1	—	—	—	—	—	—	C5TSEL<1:0>		---- --00	---- --00	

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 6</b>												
300h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
301h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
302h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
303h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
304h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
305h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
306h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
307h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
308h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
309h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
30Ah <sup>(1, 2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
30Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
30Ch	—	未实现								—	—	
30Dh	—	未实现								—	—	
30Eh	—	未实现								—	—	
30Fh	—	未实现								—	—	
310h	—	未实现								—	—	
311h	CCPR3L	捕捉 / 比较 / PWM 寄存器 3 (LSB)								xxxx xxxx	uuuu uuuu	
312h	CCPR3H	捕捉 / 比较 / PWM 寄存器 3 (MSB)								xxxx xxxx	uuuu uuuu	
313h	CCP3CON	P3M<1:0>		DC3B<1:0>		CCP3M<1:0>				0000 0000	0000 0000	
314h	PWM3CON	P3RSEN	P3DC<6:0>								0000 0000	0000 0000
315h	CCP3AS	CCP3ASE	CCP3AS<2:0>			PSS3AC<1:0>		PSS3BD<1:0>		0000 0000	0000 0000	
316h	PSTR3CON	—	—	—	STR3SYNC	STR3D	STR3C	STR3B	STR3A	---0 0001	---0 0001	
317h	—	未实现								—	—	
318h	CCPR4L	捕捉 / 比较 / PWM 寄存器 4 (LSB)								xxxx xxxx	uuuu uuuu	
319h	CCPR4H	捕捉 / 比较 / PWM 寄存器 4 (MSB)								xxxx xxxx	uuuu uuuu	
31Ah	CCP4CON	—	—	DC4B<1:0>		CCP4M<3:0>				--00 0000	--00 0000	
31Bh	—	未实现								—	—	
31Ch	CCPR5L	捕捉 / 比较 / PWM 寄存器 5 (LSB)								xxxx xxxx	uuuu uuuu	
31Dh	CCPR5H	捕捉 / 比较 / PWM 寄存器 5 (MSB)								xxxx xxxx	uuuu uuuu	
31Eh	CCP5CON	—	—	DC5B<1:0>		CCP5M<3:0>				--00 0000	--00 0000	
31Fh	—	未实现								—	—	

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注** 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 7</b>												
380h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
381h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
382h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
383h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
384h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
385h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
386h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
387h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
388h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>				---	0 0000	---	0 0000
389h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
38Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
38Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
38Ch	—	未实现								—	—	
38Dh	—	未实现								—	—	
38Eh	—	未实现								—	—	
38Fh	—	未实现								—	—	
390h	—	未实现								—	—	
391h	—	未实现								—	—	
392h	—	未实现								—	—	
393h	—	未实现								—	—	
394h	IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	0000 0000	0000 0000	
395h	IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	0000 0000	0000 0000	
396h	IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	0000 0000	0000 0000	
397h	—	未实现								—	—	
398h	—	未实现								—	—	
399h	—	未实现								—	—	
39Ah	—	未实现								—	—	
39Bh	—	未实现								—	—	
39Ch	—	未实现								—	—	
39Dh	—	未实现								—	—	
39Eh	—	未实现								—	—	
39Fh	—	未实现								—	—	

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

表 3-9: 特殊功能寄存器汇总 (续)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 8</b>												
400h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
401h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
402h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
403h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
404h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
405h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
406h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
407h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
408h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
409h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
40Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
40Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
40Ch	—	未实现								—	—	
40Dh	—	未实现								—	—	
40Eh	—	未实现								—	—	
40Fh	—	未实现								—	—	
410h	—	未实现								—	—	
411h	—	未实现								—	—	
412h	—	未实现								—	—	
413h	—	未实现								—	—	
414h	—	未实现								—	—	
415h	TMR4	Timer 4 模块寄存器								0000 0000	0000 0000	
416h	PR4	Timer 4 周期寄存器								1111 1111	1111 1111	
417h	T4CON	—	T4OUTPS<3:0>			TMR4ON	T4CKPS<1:0>			-000 0000	-000 0000	
418h	—	未实现								—	—	
419h	—	未实现								—	—	
41Ah	—	未实现								—	—	
41Bh	—	未实现								—	—	
41Ch	TMR6	Timer 6 模块寄存器								0000 0000	0000 0000	
41Dh	PR6	Timer 6 周期寄存器								1111 1111	1111 1111	
41Eh	T6CON	—	T6OUTPS<3:0>			TMR6ON	T6CKPS<1:0>			-000 0000	-000 0000	
41Fh	—	未实现								—	—	

图注: x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 9-14</b>												
x00h/ x80h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
x00h/ x81h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
x02h/ x82h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
x03h/ x83h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
x04h/ x84h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
x05h/ x85h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
x06h/ x86h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
x07h/ x87h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
x08h/ x88h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>				---	0 0000	---	0 0000
x09h/ x89h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
x0Ah/ x8Ah <sup>(1),(2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
x0Bh/ x8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
x0Ch/ x8Ch — x1Fh/ x9Fh	—	未实现								—	—	

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注** 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。  
2: 从任何存储区都可以寻址到这些寄存器。  
3: 未实现, 读为 1。

# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 15</b>												
780h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
781h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
782h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
783h <sup>(2)</sup>	STATUS	—	—	—	TO	PD	Z	DC	C	---1 1000	---q quuu	
784h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
785h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
786h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
787h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
788h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
789h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
78Ah <sup>(1,2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
78Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIF	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
78Ch — 790h	—	未实现								—	—	
791h	LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		000- 0011	000- 0011	
792h	LCDPS	WFT	BIASMD	LCDA	WA	LP<3:0>				0000 0000	0000 0000	
793h	LCDREF	LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	000- 000-	000- 000-	
794h	LCDCST	—	—	—	—	LCDCST<2:0>				---- -000	---- -000	
795h	LCDRL	LRLAP<1:0>		LRLBP<1:0>		—	LRLAT<2:0>			0000 -000	0000 -000	
796h	—	未实现								—	—	
797h	—	未实现								—	—	
798h	LCDSE0	SE<7:0>								0000 0000	uuuu uuuu	
799h	LCDSE1	SE<15:8>								0000 0000	uuuu uuuu	
79Ah — 79Fh	—	未实现								—	—	
7A0h	LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	xxxx xxxx	uuuu uuuu	
7A1h	LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	xxxx xxxx	uuuu uuuu	
7A2h	—	未实现								—	—	
7A3h	LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	xxxx xxxx	uuuu uuuu	
7A4h	LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	xxxx xxxx	uuuu uuuu	
7A5h	—	未实现								—	—	
7A6h	LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	xxxx xxxx	uuuu uuuu	
7A7h	LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	xxxx xxxx	uuuu uuuu	
7A8h	—	未实现								—	—	
7A9h	LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	xxxx xxxx	uuuu uuuu	
7AAh	LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	xxxx xxxx	uuuu uuuu	
7ABh — 7EFh	—	未实现								—	—	

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注**
- 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。
  - 2: 从任何存储区都可以寻址到这些寄存器。
  - 3: 未实现, 读为 1。

# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Banks 16-30</b>												
x00h/ x80h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
x00h/ x81h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
x02h/ x82h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
x03h/ x83h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
x04h/ x84h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
x05h/ x85h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
x06h/ x86h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
x07h/ x87h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
x08h/ x88h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>				---	0 0000	---	0 0000
x09h/ x89h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
x0Ah/ x8Ah <sup>(1),(2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
x0Bh/ x8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
x0Ch/ x8Ch — x1Fh/ x9Fh	—	未实现								—	—	

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

- 注**
- 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。
  - 2: 从任何存储区都可以寻址到这些寄存器。
  - 3: 未实现, 读为 1。



# PIC16(L)F1933

**表 3-9: 特殊功能寄存器汇总 (续)**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR 和 BOR 时的值	所有其他复位时的值	
<b>Bank 31</b>												
F80h <sup>(2)</sup>	INDF0	使用 FSR0H/FSR0L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
F81h <sup>(2)</sup>	INDF1	使用 FSR1H/FSR1L 的内容对数据存储器进行寻址来寻址此单元 (非物理寄存器)								xxxx xxxx	xxxx xxxx	
F82h <sup>(2)</sup>	PCL	程序计数器 (PC) 的低字节								0000 0000	0000 0000	
F83h <sup>(2)</sup>	STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	---1 1000	---q quuu	
F84h <sup>(2)</sup>	FSR0L	间接数据存储器地址 0 低字节指针								0000 0000	uuuu uuuu	
F85h <sup>(2)</sup>	FSR0H	间接数据存储器地址 0 高字节指针								0000 0000	0000 0000	
F86h <sup>(2)</sup>	FSR1L	间接数据存储器地址 1 低字节指针								0000 0000	uuuu uuuu	
F87h <sup>(2)</sup>	FSR1H	间接数据存储器地址 1 高字节指针								0000 0000	0000 0000	
F88h <sup>(2)</sup>	BSR	—	—	—	BSR<4:0>					---0 0000	---0 0000	
F89h <sup>(2)</sup>	WREG	工作寄存器								0000 0000	uuuu uuuu	
F8Ah <sup>(1),(2)</sup>	PCLATH	—	程序计数器高 7 位的写缓冲器								-000 0000	-000 0000
F8Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	0000 0000	0000 0000	
F8Ch — FE3h	—	未实现								—	—	
FE4h	STATUS_SHAD						Z_SHAD	DC_SHAD	C_SHAD	---- -xxx	---- -uuu	
FE5h	WREG_SHAD	工作寄存器正常 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FE6h	BSR_SHAD				存储区选择寄存器正常 (非 ICD) 影子寄存器					---x xxxx	---u uuuu	
FE7h	PCLATH_SHAD		程序计数器锁存器高字节寄存器正常 (非 ICD) 影子寄存器								-xxx xxxx	uuuu uuuu
FE8h	FSR0L_SHAD	间接数据存储器地址 0 低字节指针正常 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FE9h	FSR0H_SHAD	间接数据存储器地址 0 高字节指针正常 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FEAh	FSR1L_SHAD	间接数据存储器地址 1 低字节指针正常 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FEBh	FSR1H_SHAD	间接数据存储器地址 1 高字节指针正常 (非 ICD) 影子寄存器								xxxx xxxx	uuuu uuuu	
FECh	—	未实现								—	—	
FEDh	STKPTR	—	—	—	当前堆栈指针					---1 1111	---1 1111	
FEEh	TOSL	栈顶低字节								xxxx xxxx	uuuu uuuu	
FEFh	TOSH	—	栈顶高字节								-xxx xxxx	-uuu uuuu

**图注:** x = 未知, u = 不变, q = 取值视条件而定, - = 未实现, 读为 0, r = 保留。  
阴影单元未实现, 读为 0。

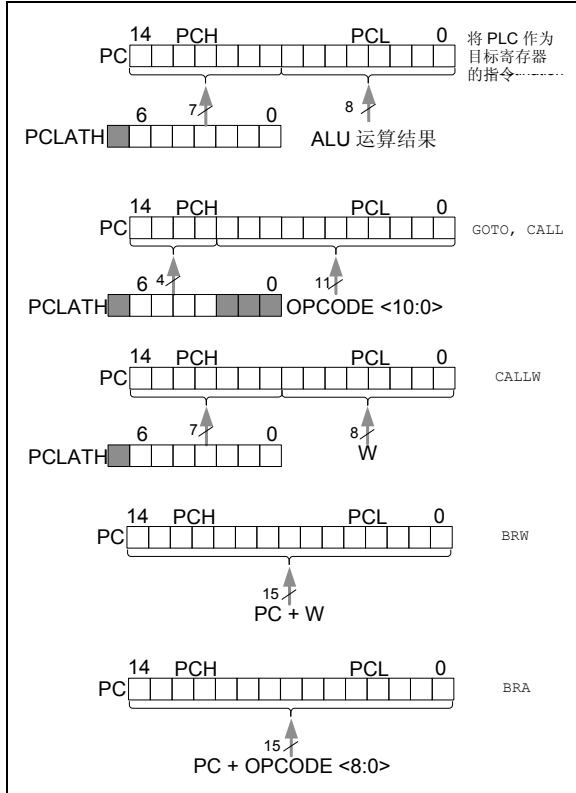
- 注**
- 1: 不能直接访问程序计数器的高字节。PCLATH 是 PC<14:8> 的保持寄存器, 其内容会被传送到程序计数器的高字节。
  - 2: 从任何存储区都可以寻址到这些寄存器。
  - 3: 未实现, 读为 1。

# PIC16(L)F1933

## 3.3 PCL 和 PCLATH

程序计数器 (Program Counter, PC) 为 15 位宽。它的低字节来自可读写的 PCL 寄存器。高字节 (PC<14:8>) 来自 PCLATH, 不可直接读写。任何复位都将清零 PC。图 3-3 给出了装载 PC 的 5 种情形。

图 3-3: 在不同情形下装载 PC



### 3.3.1 修改 PCL

执行任何将 PCL 寄存器作为目标寄存器的指令的同时会导致程序计数器 PC<14:8> 位 (PCH) 替换为 PCLATH 寄存器的内容。这使得程序计数器的整个内容可通过将所需的值写入 PCLATH 寄存器的高 7 位来进行更改。在写好 PCL 寄存器的低 8 位后, 程序计数器的所有 15 位的值将更改为 PCLATH 寄存器中的值以及那些写入到 PCL 寄存器的值。

### 3.3.2 计算 GOTO

计算 GOTO 指令是通过向程序计数器添加偏移量来实现的 (ADDWF PCL)。当使用计算 GOTO 方法执行表读操作时, 应该注意表地址是否会导致 PCL 的值超出存储边界 (每个存储区为 256 个字节)。请参见应用笔记 AN556“Implementing a Table Read” (DS00556)。

### 3.3.3 计算函数调用

计算函数 CALL 允许程序维护函数表并提供其他方法来执行状态机或查找表。当使用计算函数 CALL 执行表读操作时, 应该注意表地址是否会导致 PCL 的值超出存储边界 (每个存储区为 256 个字节)。

如果使用 CALL 指令, 则使用 CALL 指令的操作数来装载 PCH<2:0> 和 PCL 寄存器。使用 PCLATH<6:3> 来装载 PCH<6:3>。

CALLW 指令通过将 PCLATH 寄存器和 W 寄存器合并以形成目标地址来使能计算调用。计算 CALLW 可通过将所需地址装入 W 寄存器并执行 CALLW 指令来实现。将 W 寄存器的值装入 PCL 寄存器, 将 PCLATH 的值装入 PCH 寄存器。

### 3.3.4 跳转

跳转指令向 PC 添加偏移量, 从而允许重定位代码和代码跨页边界。有 2 种形式的跳转: BRW 和 BRA。在这两种情况下, PC 都会加 1 以取下一条指令。使用任一跳转指令都可以跨 PCL 存储边界。

如果使用 BRW, 则使用所需无符号地址装载 W 寄存器, 然后执行 BRW。整个 PC 会装入地址 PC + 1 + W。

如果使用 BRA, 则整个 PC 将装入 PC + 1 + (BRA 指令操作数的有符号值)。

## 3.4 堆栈

所有器件都具有一个 16 级深 x 15 位宽的硬件堆栈（见图 3-4 至图 3-7）。堆栈空间不属于程序存储空间或数据存储空间的一部分。当执行 CALL 或 CALLW 指令或由于中断导致程序跳转时，PC 的值会被压入堆栈。当执行 RETURN、RETLW 或 RETFIE 指令时，PC 值从堆栈弹出。PCLATH 的值不受压栈或出栈操作的影响。

如果 STVREN 位被编程为 0（配置字寄存器 2），则堆栈可作为循环缓冲区使用。这就是说在压栈 16 次后，第 17 次压栈操作会覆盖第 1 次压栈操作存储的值，而第 18 次压栈操作覆盖第 2 次压栈操作存储的值（以此类推）。不管是否使能复位，STKOVF 和 STKUNF 标志位在上溢 / 下溢时都会置 1。

**注 1:** 没有称为 PUSH 或 POP 的指令或助记符。这些操作都是在执行 CALL、CALLW、RETURN、RETLW 和 RETFIE 指令或跳转到中断地址向量时发生。

### 3.4.1 访问堆栈

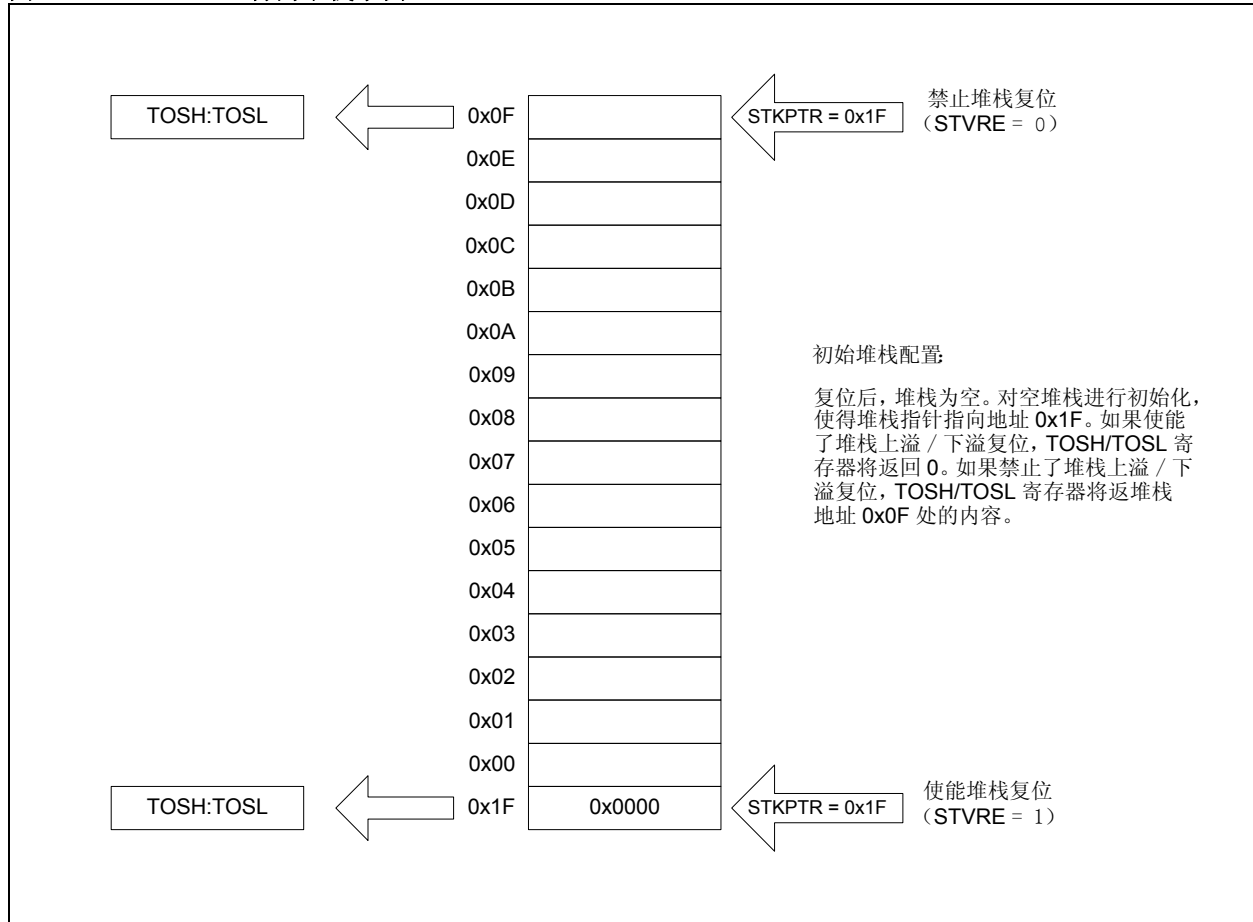
可通过 TOSH、TOSL 和 STKPTR 寄存器来使用堆栈。STKPTR 是堆栈指针的当前值。TOSH:TOSL 寄存器对指向栈顶。这两个寄存器都可读写。由于 PC 的大小为 15 位，故 TOS 划分为 TOSH 和 TOSL 两部分。要访问堆栈，可调整用来定位 TOSH:TOSL 的 STKPTR 值，然后对 TOSH:TOSL 执行读 / 写操作。STKPTR 为 5 位，允许检测上溢和下溢。

**注:** 在允许中断的情况下修改 STKPTR 时应谨慎。

在正常程序运行期间，CALL、CALLW 和中断会使 STKPTR 值递增 1，而 RETLW、RETURN 和 RETFIE 会使 STKPTR 值递减 1。任何时候都可以检查 STKPTR，以查看可用堆栈空间。STKPTR 总是指向堆栈中的当前使用单元。因此，CALL 或 CALLW 指令会先使 STKPTR 值递增 1，然后写 PC；而返回操作则会先卸载 PC，然后使 STKPTR 值递减 1。

访问堆栈的示例，请参见图 3-4 至图 3-7。

图 3-4: 访问堆栈示例 1



# PIC16(L)F1933

图 3-5: 访问堆栈示例 2

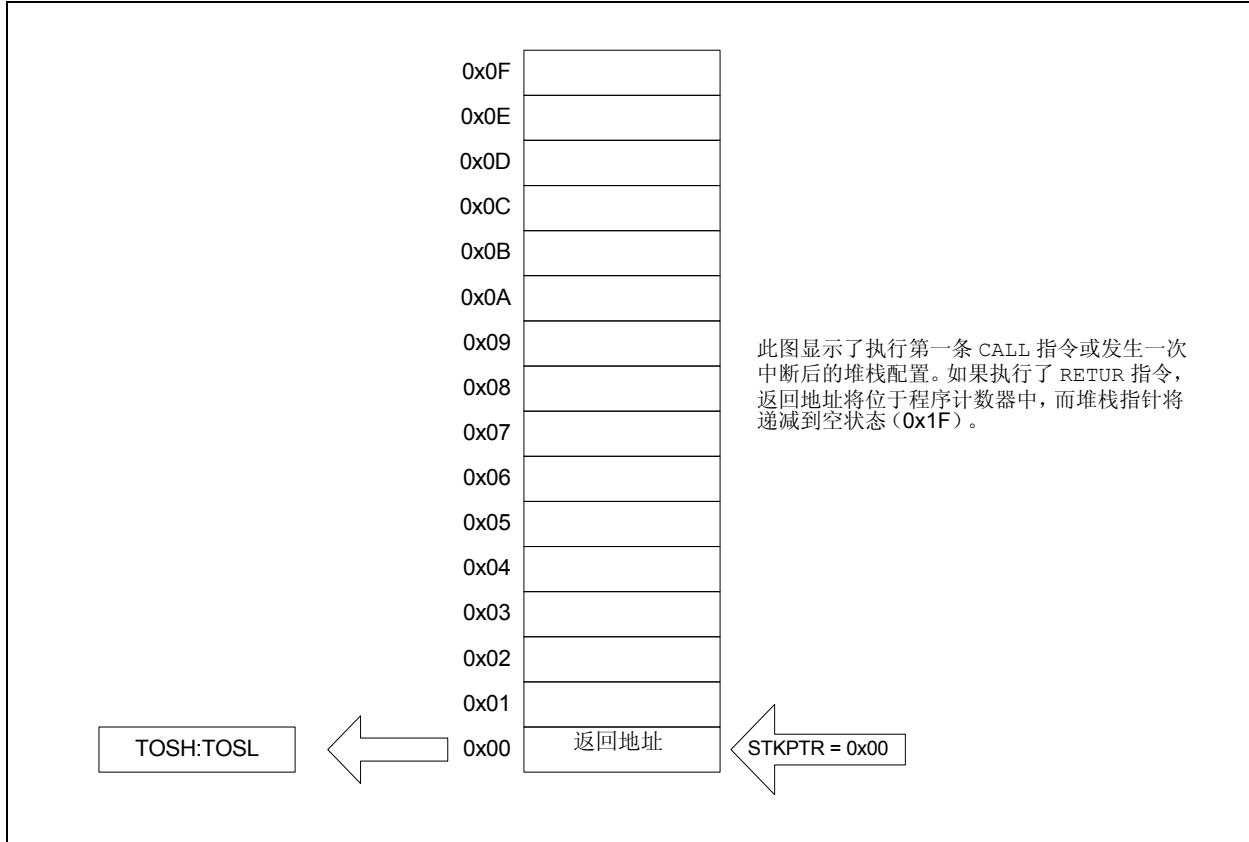


图 3-6: 访问堆栈示例 3

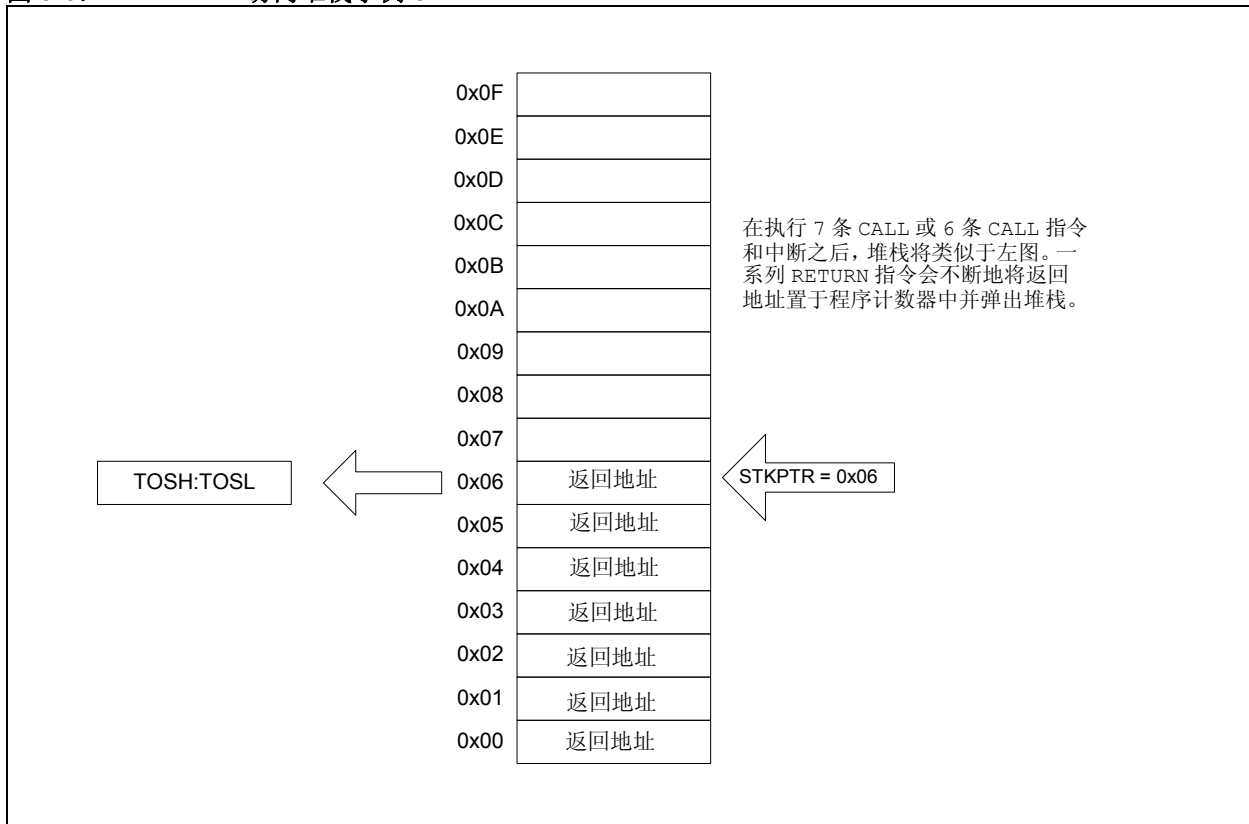
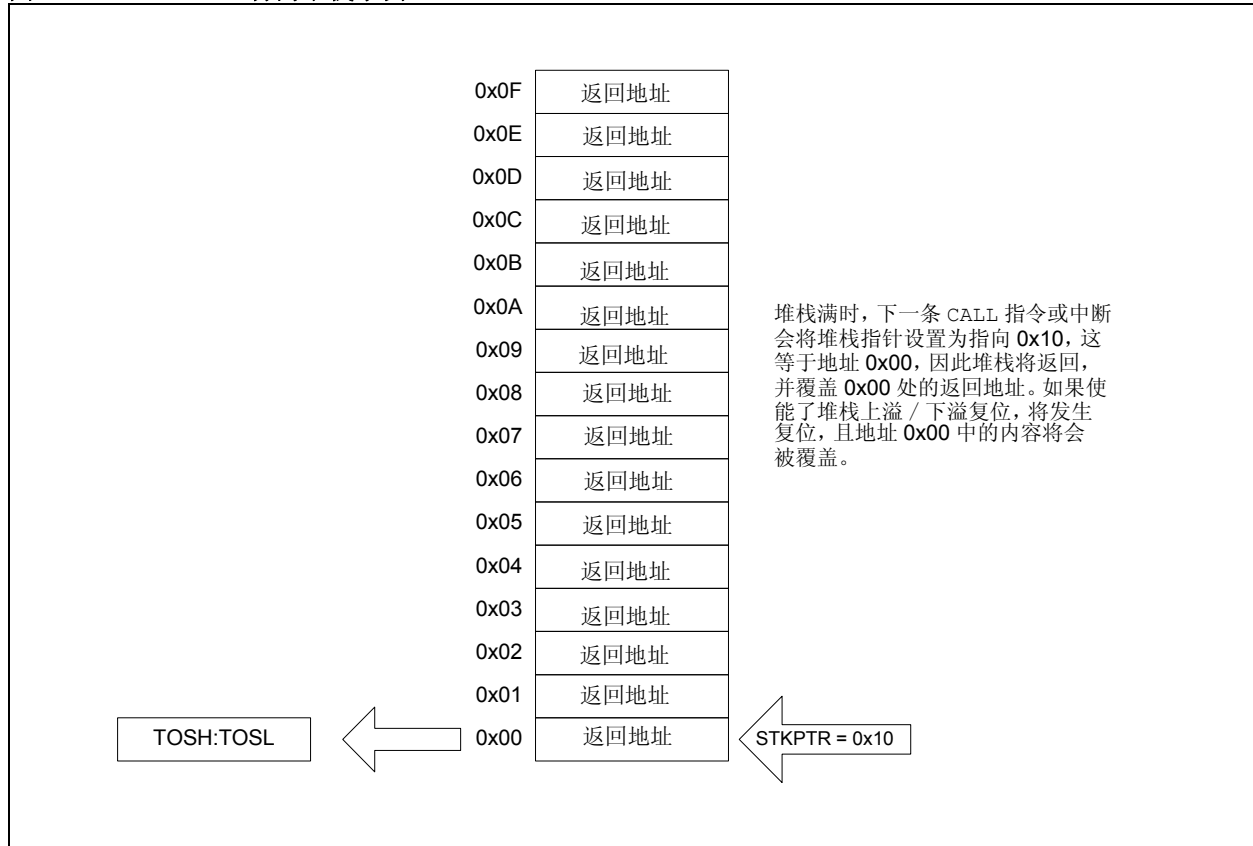


图 3-7: 访问堆栈示例 4



### 3.4.2 上溢 / 下溢复位

如果配置字寄存器 2 中的 STVREN 位编程为 1，则在压满 16 级后再执行压栈操作或弹出第 1 级后再执行出栈操作，PCON 寄存器中的相应位（分别为 STKOVF 或 STKUNF）会置 1，从而使器件复位。

### 3.5 间接寻址

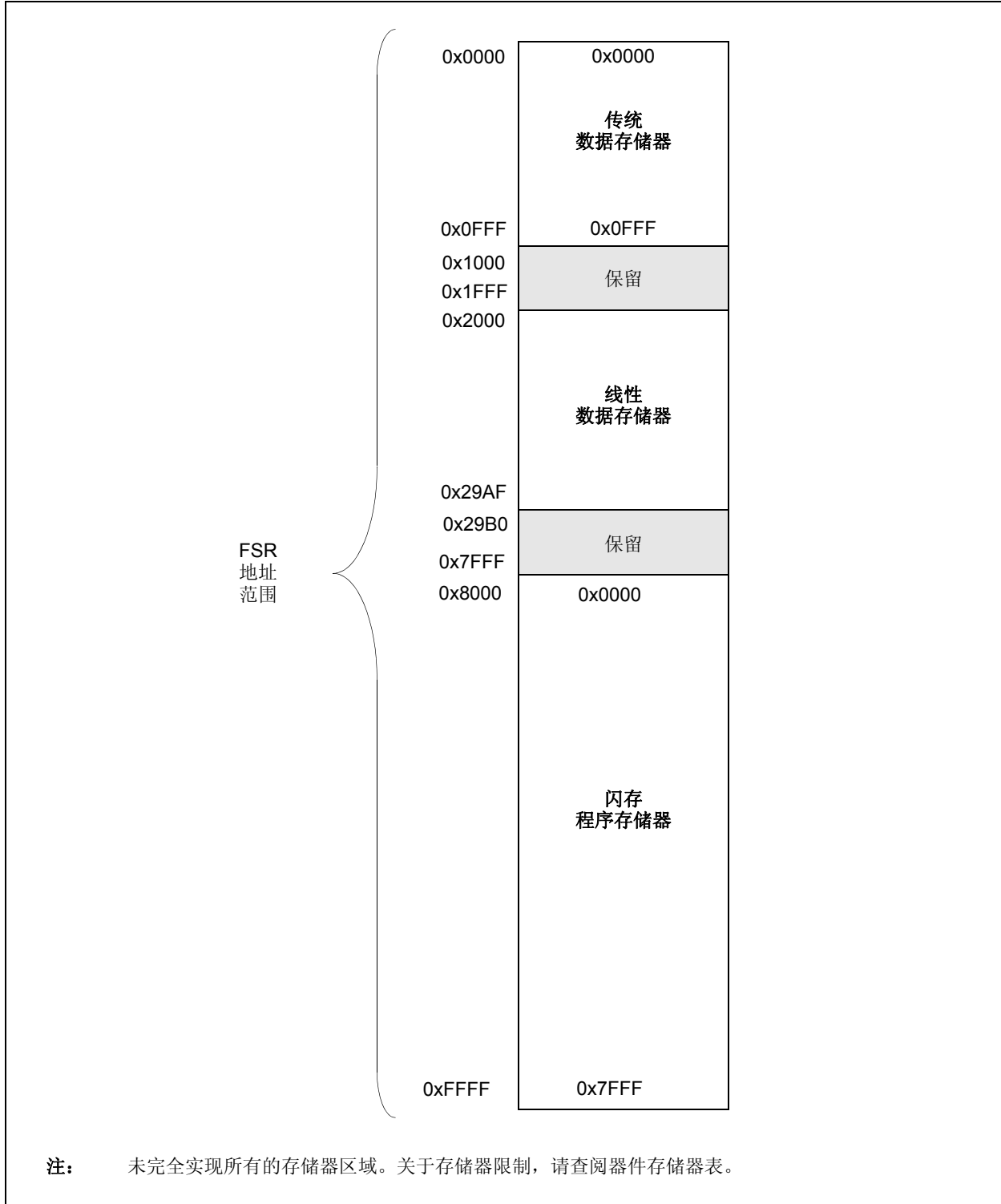
INDFn 寄存器不是物理寄存器。任何访问 INDFn 寄存器的指令，实际上都是访问由文件选择寄存器（FSR）指定的地址处的寄存器。如果 FSRn 地址指定了 2 个 INDFn 寄存器中的任何一个，执行读操作会返回 0，而写操作无法实现（尽管状态位会受影响）。可通过 FSRnH 和 FSRnL 对来创建 FSRn 寄存器值。

FSR 寄存器形成的 16 位地址允许对 65536 个地址单元的空间进行寻址。这些地址单元可划分为 3 个存储区：

- 传统数据存储器
- 线性数据存储器
- 闪存程序存储器

# PIC16(L)F1933

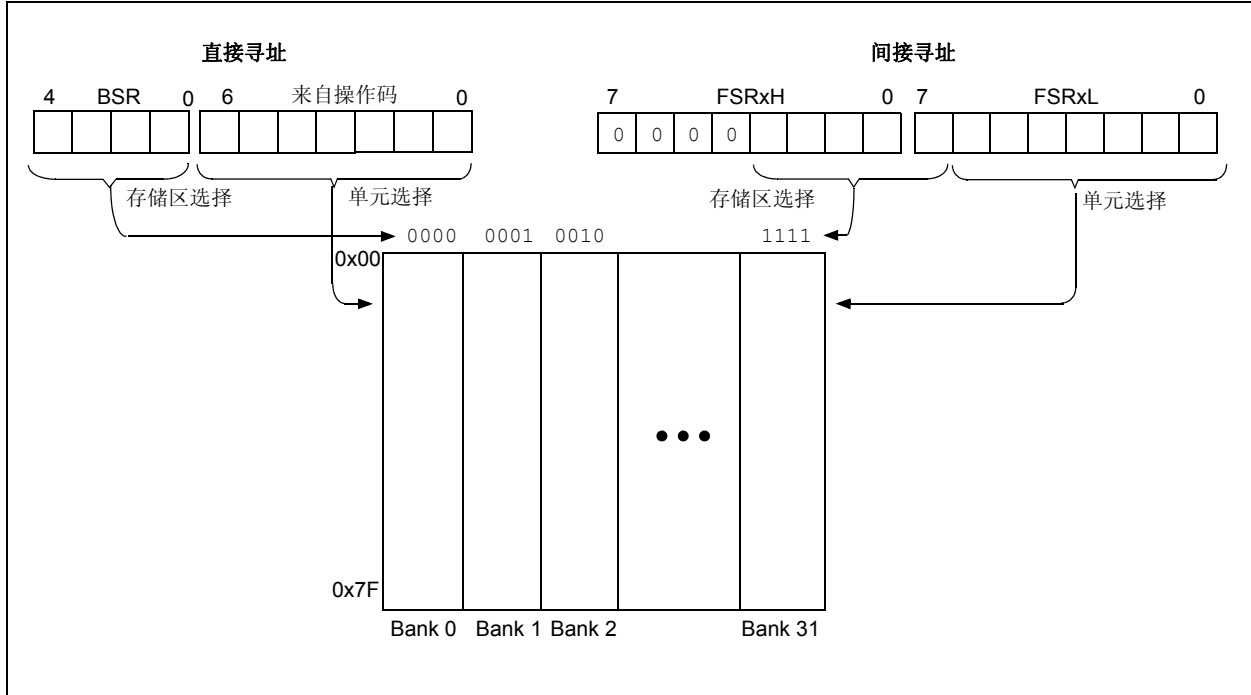
图 3-8: 间接寻址



## 3.5.1 传统数据存储器

传统数据存储器指的是从 FSR 地址 0x000 到 FSR 地址 0xFFF 的区域。此地址对应于所有 SFR、GPR 和公共寄存器的绝对地址。

**图 3-9: 传统数据存储器映射**



# PIC16(L)F1933

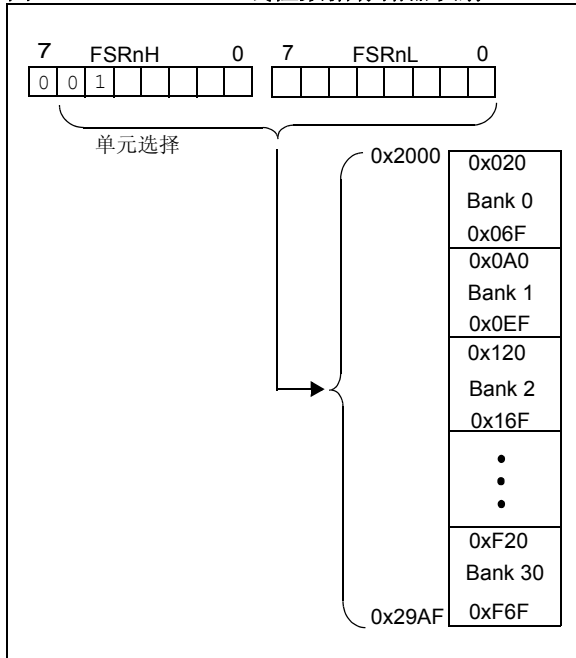
## 3.5.2 线性数据存储器

线性数据存储器指的是从 FSR 地址 0x2000 到 FSR 地址 0x29AF 的区域。该区域为虚拟区域，它指向所有存储区中 80 字节的 GPR 存储区块。

未实现的存储区读为 0x00。使用线性数据存储器区域允许缓冲区大于 80 字节，因为当 FSR 增大到超过一个存储区时，会直接转到下一个存储区的 GPR 存储器。

线性数据存储器区域不包含 16 字节的公共存储器。

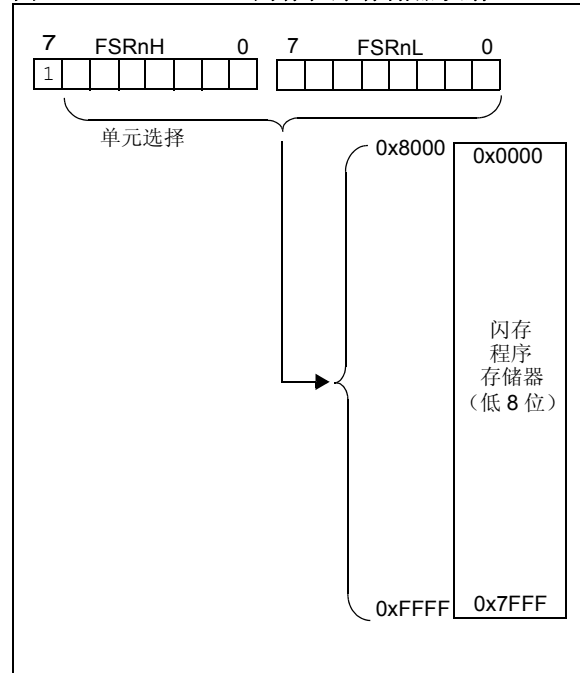
图 3-10: 线性数据存储器映射



## 3.5.3 闪存程序存储器

要使常数的访问更为容易，可将整个闪存程序存储器映射到 FSR 地址空间的高半部分。当 FSRnH 的 MSB 置 1 时，低 15 位就是可通过 INDF 进行访问的程序存储器的地址。只有每个存储单元的低 8 位可通过 INDF 进行访问。通过 FSR/INDF 接口无法对闪存程序存储器执行写操作。所有通过 FSR/INDF 接口对闪存程序存储器进行访问的指令都需要一个额外的指令周期才能完成。

图 3-11: 闪存程序存储器映射





## 4.0 器件配置

器件配置包括配置字寄存器 1 和配置字寄存器 2、代码保护以及器件 ID。

### 4.1 配置字

有几个配置字位可用于选择不同的振荡器和存储器保护选项，这些位实现为位于 8007h 处的配置字寄存器 1 和位于 8008h 处的配置字寄存器 2。

<b>注：</b>	配置字寄存器 2 中的 $\overline{\text{DEBUG}}$ 位由器件开发工具（包括调试器和编程器）自动管理。对于正常的器件操作，此位应保持为 1。
-----------	---

## 寄存器 4-1: 配置字寄存器 1

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
FCMEN	IESO	CLKOUTEN	BOREN1	BOREN0	CPD	CP
bit 13						bit 7

R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1	R/P-1/1
MCLRE	PWRTE	WDTE1	WDTE0	FOSC2	FOSC1	FOSC0
bit 6						bit 0

### 图注:

R = 可读位

P = 可编程位

U = 未实现位, 读为 1

0 = 清零

1 = 置 1

-n = 空白时或块擦除后的值

- bit 13 **FCMEN:** 故障保护时钟监视器使能位  
1 = 使能故障保护时钟监视器  
0 = 禁止故障保护时钟监视器
- bit 12 **IESO:** 内部 / 外部时钟切换位  
1 = 使能内部 / 外部时钟切换模式  
0 = 禁止内部 / 外部时钟切换模式
- bit 11 **CLKOUTEN:** 时钟输出使能位  
1 = 禁止 CLKOUT 功能, 即禁止 I/O 或 RA6/CLKOUT 引脚上的振荡器功能  
0 = 使能 RA6/CLKOUT 引脚上的 CLKOUT 功能
- bit 10-9 **BOREN<1:0>:** 欠压复位使能位 (1)  
11 = 使能 BOR  
10 = BOR 在工作时使能, 在休眠时禁止  
01 = BOR 由 BORCON 寄存器的 SBOREN 位控制  
00 = 禁止 BOR
- bit 8 **CPD:** 数据代码保护位 (2)  
1 = 禁止数据存储代码保护  
0 = 使能数据存储代码保护
- bit 7 **CP:** 代码保护位 (3)  
1 = 禁止程序存储器代码保护  
0 = 使能程序存储器代码保护
- bit 6 **MCLRE:** RE3/MCLR/VPP 引脚功能选择位  
如果 LVP 位 = 1:  
忽略此位。  
如果 LVP 位 = 0:  
1 = RE3/MCLR/VPP 引脚功能为 MCLR; 使能弱上拉功能。  
0 = RE3/MCLR/VPP 引脚功能为数字输入; 内部禁止 MCLR; 弱上拉功能由 WPUE3 位控制。
- bit 5 **PWRTE:** 上电延时定时器使能位 (1)  
1 = 禁止 PWRT  
0 = 使能 PWRT
- bit 4-3 **WDTE<1:0>:** 看门狗定时器使能位  
11 = 使能 WDT  
10 = WDT 在运行时使能, 休眠时禁止  
01 = WDT 由 WDTCON 寄存器中的 SWDTEN 位控制  
00 = 禁止 WDT

- 注 1: 使能欠压复位并不能自动使能上电延时定时器。  
2: 擦除操作期间关闭代码保护时, 将擦除整个数据 EEPROM 的内容。  
3: 当关闭代码保护时, 将擦除整个程序存储器的内容。

**寄存器 4-1: 配置字寄存器 1 (续)**

bit 2-0

**FOSC<2:0>**: 振荡器选择位

- 111 = ECH: 外部时钟, 高功耗模式: RA7/OSC1/CLKIN 引脚为 CLKIN 功能
- 110 = ECM: 外部时钟, 中等功耗模式: RA7/OSC1/CLKIN 引脚为 CLKIN 功能
- 101 = ECL: 外部时钟, 低功耗模式: RA7/OSC1/CLKIN 引脚为 CLKIN 功能
- 100 = INTOSC 振荡器: RA7/OSC1/CLKIN 引脚为 I/O 功能
- 011 = EXTRC 振荡器: RA7/OSC1/CLKIN 引脚为 RC 功能
- 010 = HS 振荡器: 高速晶振 / 谐振器连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚
- 001 = XT 振荡器: 晶振 / 谐振器连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚
- 000 = LP 振荡器: 低功耗晶振连接到 RA6/OSC2/CLKOUT 和 RA7/OSC1/CLKIN 引脚

- 注**
- 1: 使能欠压复位并不能自动使能上电延时定时器。
  - 2: 擦除操作期间关闭代码保护时, 将擦除整个数据 EEPROM 的内容。
  - 3: 当关闭代码保护时, 将擦除整个程序存储器的内容。

## 寄存器 4-2: 配置字寄存器 2

R/P-1/1	R/P-1/1	U-1	R/P-1/1	R/P-1/1	R/P-1/1	U-1
LVP <sup>(1)</sup>	DEBUG <sup>(3)</sup>	—	BORV	STVREN	PLLEN	—
bit 13						bit 7

U-1	R/P-1/1	R/P-1/1	U-1	U-1	R/P-1/1	R/P-1/1
—	VCAPEN<1:0> <sup>(2)</sup>		—	—	WRT1	WRT0
bit 6						bit 0

### 图注:

R = 可读位

P = 可编程位

U = 未实现位, 读为 1

0 = 清零

1 = 置 1

-n = 空白时或块擦除后的值

- bit 13     **LVP:** 低压编程使能位 <sup>(1)</sup>  
 1 = 使能低压编程  
 0 = 必须使用 MCLR/VPP 引脚上的高压进行编程
- bit 12     **DEBUG:** 在线调试器模式位 <sup>(3)</sup>  
 1 = 禁止在线调试器, RB6/ICSPCLK 和 RB7/ICSPDAT 是通用 I/O 引脚  
 0 = 使能在线调试器, RB6/ICSPCLK 和 RB7/ICSPDAT 专用于调试器
- bit 11     **未实现:** 读为 1
- bit 10     **BORV:** 欠压复位电压选择位  
 1 = 欠压复位电压设置为 1.9V  
 0 = 欠压复位电压设置为 2.5V
- bit 9      **STVREN:** 堆栈上溢 / 下溢复位使能位  
 1 = 堆栈上溢或下溢将导致复位  
 0 = 堆栈上溢或下溢不会导致复位
- bit 8      **PLLEN:** PLL 使能位  
 1 = 使能 4xPLL  
 0 = 禁止 4xPLL
- bit 7-6    **未实现:** 读为 1
- bit 5-4    **VCAPEN<1:0>:** 稳压器电容使能位 <sup>(2)</sup>  
 00 = 在 RA0 引脚上使能 VCAP 功能  
 01 = 在 RA5 引脚上使能 VCAP 功能  
 10 = 在 RA6 引脚上使能 VCAP 功能  
 11 = VCAP 引脚上无电容
- bit 3-2    **未实现:** 读为 1
- bit 1-0    **WRT<1:0>:** 闪存自写保护位  
**4 kW 闪存:**  
 11 = 写保护关闭  
 10 = 000h 至 1FFh 受写保护, 200h 至 FFFh 可以由 EECON 控制寄存器修改  
 01 = 000h 至 7FFh 受写保护, 800h 至 FFFh 可以由 EECON 控制寄存器修改  
 00 = 000h 至 FFFh 受写保护, 无可由 EECON 控制寄存器修改的地址

- 注    **1:** 当通过 LVP 进入编程模式时, LVP 位无法编程为 0。  
**2:** 仅在 PIC16LF193X 器件上读为 11。  
**3:** 配置字寄存器中的 DEBUG 位由器件开发工具 (包括调试器和编程器) 自动管理。对于正常的器件操作, 此位应保持为 1。

## 4.2 代码保护

代码保护用于保护器件不受未经授权的访问。程序存储器保护和数据 EEPROM 保护是独立控制的。任何代码保护设置不会影响对程序存储器和数据 EEPROM 的内部访问。

### 4.2.1 程序存储器保护

通过配置字寄存器 1 中的  $\overline{CP}$  位禁止对整个程序存储空间的外部读写操作。 $CP = 0$  时，禁止对程序存储器的外部读写操作，读操作将返回全 0。无论保护位的设置如何，CPU 仍可以继续读程序存储器。写程序存储器取决于写保护设置。更多信息，请参见第 4.3 节“写保护”。

### 4.2.2 数据 EEPROM 保护

通过  $\overline{CPD}$  位禁止对整个数据 EEPROM 的外部读写操作。 $CPD = 0$  时，禁止对数据 EEPROM 的外部读写操作。无论保护位的设置如何，CPU 仍可以继续读写数据 EEPROM。

## 4.3 写保护

写保护用于保护器件不受意外的自写访问。在允许修改程序存储器其他区域的同时可以保护应用程序，例如引导加载程序软件。

配置字寄存器 2 中的  $WRT<1:0>$  位定义了受保护的程序存储块的大小。

## 4.4 用户 ID

有 4 个存储单元（8000h-8003h）被指定为 ID 地址单元，供用户存储校验和或其他代码标识号。在正常执行过程中可以读写这些单元。更多有关访问这些存储单元的信息，请参见第 4.5 节“器件 ID 和版本 ID”。有关校验和计算的更多信息，请参见“PIC16F193X/LF193X/PIC16F194X/LF194X/PIC16LF190X Memory Programming Specification”（DS41397）。

# PIC16(L)F1933

## 4.5 器件 ID 和版本 ID

存储单元 8006h 用于存储器件 ID 和版本 ID。高 9 位保存器件 ID。低 5 位保存版本 ID。更多有关访问这些存储单元的信息，请参见第 11.5 节“用户 ID、器件 ID 和配置字访问”。

开发工具（例如器件编程器和调试器）也可以用来读取器件 ID 和版本 ID。

### 寄存器 4-3: DEVICEID: 器件 ID 寄存器 (1)

R	R	R	R	R	R	R
DEV8	DEV7	DEV6	DEV5	DEV4	DEV3	DEV2
bit 13						bit 7

R	R	R	R	R	R	R
DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 6						bit 0

<b>图注:</b>	U = 未实现位，读为 0
R = 可读位	W = 可写位
-n = 上电复位时的值	0 = 清零
	1 = 置 1
	x = 未知

bit 13-5      **DEV<8:0>**: 器件 ID 位  
100011001 = PIC16F1933  
100100001 = PIC16LF1933

bit 4-0      **REV<4:0>**: 版本 ID 位  
这些位用于标识版本。

注 1: 此单元不可写。

## 5.0 振荡器模块（带故障保护时钟监视器）

### 5.1 概述

该振荡器模块有很多种时钟源和选择功能，从而使其应用非常广泛，并可最大限度地提高性能和降低功耗。图 5-1 给出了振荡器模块框图。

时钟源可以由外部振荡器、石英晶振、陶瓷谐振器以及阻容（Resistor-Capacitor, RC）电路提供。此外，系统时钟源还可以由 2 个内部振荡器中的一个和 PLL 电路提供，并可通过软件选择时钟速度。其他时钟特性包括：

- 通过软件选择外部或内部系统时钟源。
- 双速启动模式，使外部振荡器起振到代码执行之间的延时达到最小。
- 故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM）旨在检测外部时钟源（LP、XT、HS、EC 或 RC 模式）的故障并自动切换到内部振荡器。
- 振荡器起振定时器（OST）可以确保晶振源的稳定性。

振荡器模块可配置为以下八种时钟模式之一。

1. ECL——外部时钟低功耗模式（0 MHz 至 0.5 MHz）
2. ECM ——外部时钟中等功耗模式（0.5 MHz 至 4 MHz）
3. ECH——外部时钟高功耗模式（4 MHz 至 32 MHz）
4. LP——32 kHz 低功耗晶振模式
5. XT——中等增益晶振或陶瓷谐振器振荡器模式（高达 4 MHz）
6. HS——高增益晶振或陶瓷谐振器模式（4 MHz 至 20 MHz）
7. RC——外部阻容（RC）
8. INTOSC——内部振荡器（31 kHz 至 32 MHz）

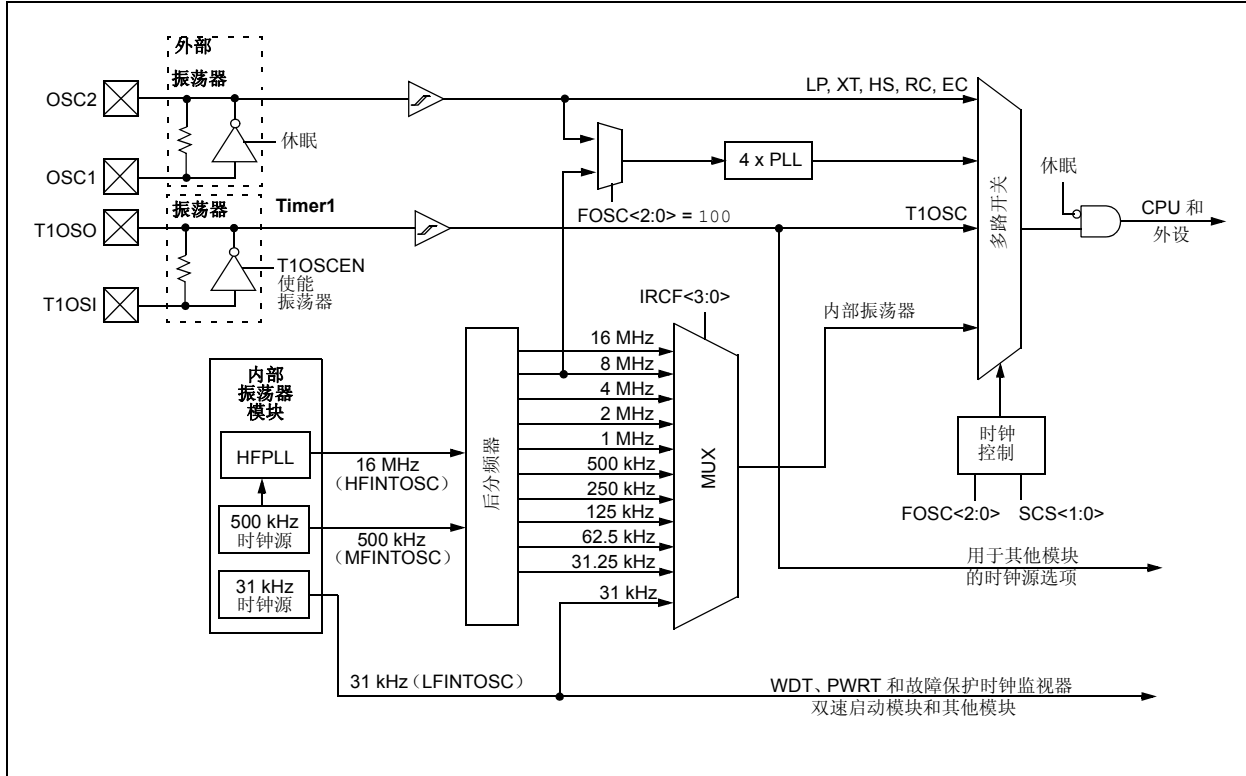
时钟源模式可通过配置字寄存器 1 中的 FOSC<2:0> 位选择。FOSC 位决定了器件首次上电时使用的振荡器类型。

EC 时钟模式依赖于外部逻辑电平信号作为器件时钟源。LP、XT 和 HS 时钟模式要求器件连接一个外部晶振或谐振器。针对不同的频率范围对每种模式进行了优化。RC 时钟模式要求使用外部阻容电路来设置振荡器频率。

INTOSC 内部振荡器电路可以产生低、中、高频时钟源，分别标记为 LFINTOSC、MFINTOSC 和 HFINTOSC。（请参见图 5-1 中的内部振荡器电路）。这三个时钟源可产生多种器件时钟频率供选择。

# PIC16(L)F1933

图 5-1: PIC® MCU 时钟源简化框图





## 5.2 时钟源类型

时钟源可以分为外部或内部两类。

外部时钟源依赖外部电路作为时钟源工作。示例有：振荡器模块（EC 模式）、石英晶振或陶瓷谐振器（LP、XT 和 HS 模式）以及阻容（RC）模式电路。

内部时钟源内置在振荡器模块中。内部振荡器模块有两个内部振荡器和一个专用的锁相环（HFPLL），用于产生三种内部系统时钟源：16 MHz 高频内部振荡器（HFINTOSC）、500 kHz 中频内部振荡器（MFINTOSC）和 31 kHz 低频内部振荡器（LFINTOSC）。

可以通过 OSCCON 寄存器中的系统时钟选择（SCS）位选择外部或内部时钟源作为系统时钟。其他信息请参见第 5.3 节“时钟切换”。

### 5.2.1 外部时钟源

通过执行以下某一操作，可将外部时钟源用作器件系统时钟：

- 编程配置字寄存器 1 中的 FOSC<2:0> 位，以选择器件复位时用作默认系统时钟的外部时钟源。
- 写入 OSCCON 寄存器中的 SCS<1:0> 位可以将系统时钟源切换为：
  - Timer1 振荡器（运行时），或
  - 外部时钟源（由 FOSC 位的值决定）。

更多信息，请参见第 5.3 节“时钟切换”。

#### 5.2.1.1 EC 模式

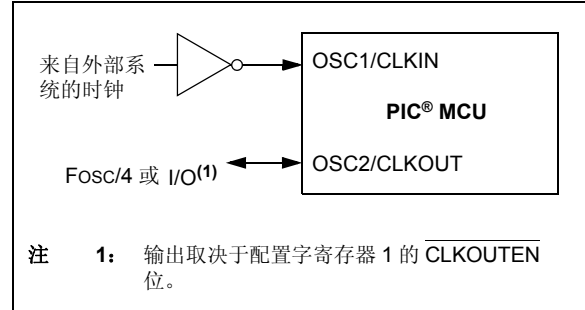
外部时钟（EC）模式将外部产生的逻辑电平信号作为系统时钟源。工作在此模式下时，外部时钟源应连接到 OSC1 输入引脚。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT 引脚。图 5-2 给出了 EC 模式的引脚连接。

EC 模式有 3 种功耗模式，可通过配置字寄存器 1 进行选择：

- 高功耗，4-32 MHz（FOSC = 111）
- 中等功耗，0.5-4 MHz（FOSC = 110）
- 低功耗，0-0.5 MHz（FOSC = 101）

当选择 EC 模式时，振荡器起振定时器（OST）被禁止。因此，在上电复位（POR）或从休眠模式唤醒后，不会有延时操作。由于 PIC® MCU 的设计是全静态的，停止外部时钟输入可以在停止器件的同时使所有的数据保持原样。外部时钟重新启动之后，器件将恢复工作就像没有时间流逝一样。

图 5-2: 外部时钟（EC）模式工作原理



#### 5.2.1.2 LP、XT 和 HS 模式

LP、XT 和 HS 模式支持在 OSC1 和 OSC2 引脚之间连接石英晶振或陶瓷谐振器的应用（图 5-3）。这三种模式可以选择内部反相放大器的低、中等或高增益设置以支持多种谐振器类型和速度。

**LP** 振荡器模式选择内部反相放大器的最低增益设置。LP 模式的电流消耗在这三种模式中最低。此模式只适合于驱动 32.768 kHz 的音叉型晶振（时钟晶体）。

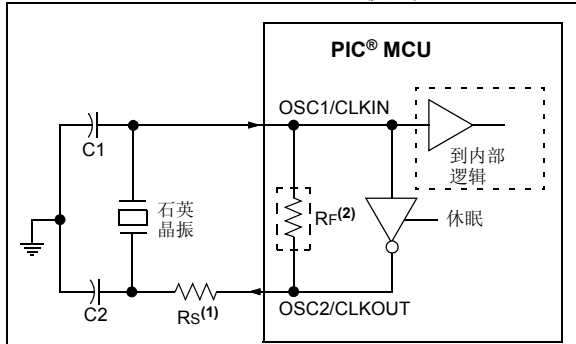
**XT** 振荡器模式选择内部反相放大器的中等增益设置。XT 模式的电流消耗在这三种模式中居中。此模式最适合于驱动具有中等驱动电平规范的谐振器。

**HS** 振荡器模式选择内部反相放大器的最高增益设置。HS 模式的电流消耗在这三种模式中最高。此模式最适合于要求高驱动电平设置的谐振器。

图 5-3 和图 5-4 分别显示了石英晶振和陶瓷谐振器的典型电路。

# PIC16(L)F1933

图 5-3: 石英晶振工作原理 (LP、XT 或 HS 模式)



- 注 1: 对于低驱动电平的石英晶振可能需要一个串联电阻 ( $R_s$ )。
- 2:  $R_f$  的值随选定的振荡器模式而变化 (通常介于  $2\text{ M}\Omega$  和  $10\text{ M}\Omega$  之间)。

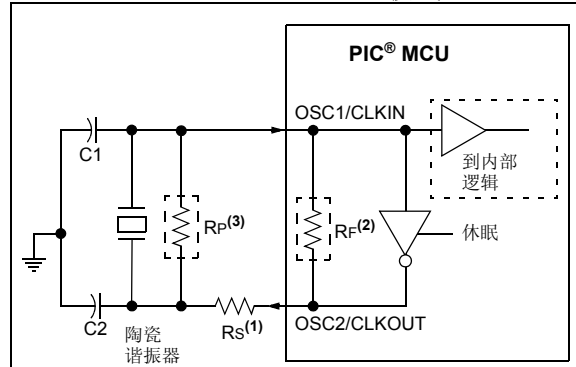
注 1: 石英晶振的特性取决于类型、封装以及制造商。用户应该查阅制造商的数据手册以获知规范和推荐的应用场合。

2: 请总是验证在应用期望的  $V_{DD}$  和温度范围下振荡器的性能。

3: 要获取振荡器设计帮助, 请参见以下 Microchip 应用笔记:

- AN826, “Crystal Oscillator Basics and Crystal Selection for rfPIC<sup>®</sup> and PIC<sup>®</sup> Devices” (DS00826)
- AN849, “Basic PIC<sup>®</sup> Oscillator Design” (DS00849)
- AN943, “Practical PIC<sup>®</sup> Oscillator Analysis and Design” (DS00943)
- AN949, “Making Your Oscillator Work” (DS00949)

图 5-4: 陶瓷谐振器工作原理 (XT 或 HS 模式)



- 注 1: 对于低驱动电平的陶瓷谐振器可能需要一个串联电阻 ( $R_s$ )。
- 2:  $R_f$  的值随选定的振荡器模式而变化 (通常介于  $2\text{ M}\Omega$  和  $10\text{ M}\Omega$  之间)。
- 3: 要让陶瓷谐振器正常工作可能还需要并联一个反馈电阻 ( $R_p$ )。

## 5.2.1.3 振荡器起振定时器 (OST)

如果振荡器模块配置为 LP、XT 或 HS 模式, 则振荡器起振定时器 (OST) 将对 OSC1 引脚的振荡计数 1024 次。这发生在上电复位 (POR) 且上电延时定时器 (PWRT) 延时结束 (如果配置了此延时) 时或从休眠状态唤醒时。在计数期间, 程序计数器不会递增, 程序执行暂停。OST 确保使用石英晶振或陶瓷谐振器的振荡器电路已起振并且为振荡器模块提供稳定的系统时钟。

为了让外部振荡器起振和代码执行之间的延时最短, 可以选择双速时钟启动模式 (见第 5.4 节“双速时钟启动模式”)。

## 5.2.1.4 4X PLL

振荡器模块包含一个 4X PLL, 用于与外部和内部时钟源一起使用, 以提供系统时钟源。4X PLL 的输入频率必须在规定的范围内。请参见应用的电气规范章节中的 PLL 时钟时序规范。

使用时, 可以通过以下两种方法之一使能 4X PLL:

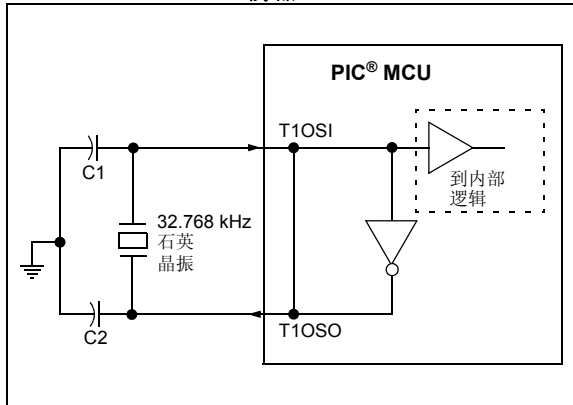
1. 将配置字寄存器 2 中的  $PLLEN$  位编程为 1。
2. 将 1 写入 OSCCON 寄存器中的  $SPLLEN$  位。如果配置字寄存器 2 中的  $PLLEN$  位编程为 1, 那么忽略  $SPLLEN$  的值。

## 5.2.1.5 TIMER1 振荡器

Timer1 振荡器是与 Timer1 外设关联的独立晶振。通过 T1OSO 和 T1OSI 器件引脚之间连接的 32.768 kHz 晶振，对 Timer1 振荡器的计时操作进行了优化。

Timer1 振荡器可以用作备用系统时钟源，在运行时可以通过时钟切换进行选择。更多信息，请参见第 5.3 节“时钟切换”。

图 5-5: 石英晶振操作 (TIMER1 振荡器)



- 注 1:** 石英晶振的特性取决于类型、封装以及制造商。用户应该查阅制造商的数据手册以获知规范和推荐的应用场合。
- 注 2:** 请总是验证在应用期望的 VDD 和温度范围内振荡器的性能。
- 注 3:** 要获取振荡器设计帮助，请参见以下 Microchip 应用笔记：
- AN826, “Crystal Oscillator Basics and Crystal Selection for rfPIC® and PIC® Devices” (DS00826)
  - AN849, “Basic PIC® Oscillator Design” (DS00849)
  - AN943, “Practical PIC® Oscillator Analysis and Design” (DS00943)
  - AN949, “Making Your Oscillator Work” (DS00949)
  - TB097, “Interfacing a Micro Crystal MS1V-T1K 32.768 kHz Tuning Fork Crystal to a PIC16F690/SS” (DS91097)
  - AN1288, “Design Practices for Low-Power External Oscillators” (DS01288)

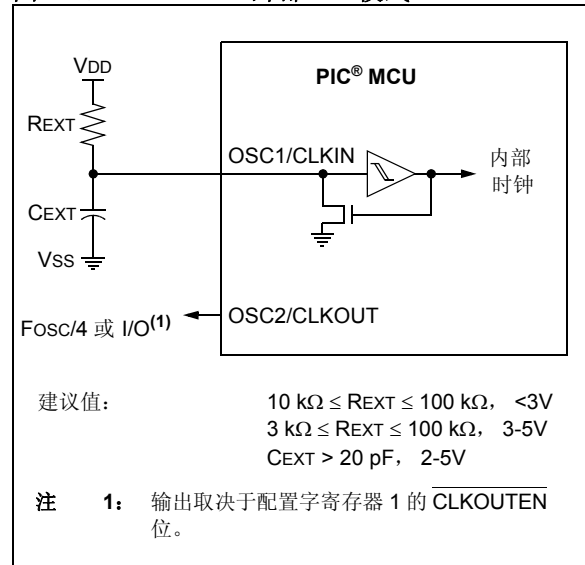
## 5.2.1.6 外部 RC 模式

外部阻容 (RC) 模式支持使用外部 RC 电路。当对时钟精度没有要求时，外部 RC 模式可以让设计人员在选择频率上有最大的灵活性，同时将成本保持在最低。

RC 电路与 OSC1 引脚连接。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT。OSC2/CLKOUT 引脚的功能由配置字寄存器 1 中的 CLKOUTEN 位的状态决定。

图 5-6 给出了外部 RC 模式连接。

图 5-6: 外部 RC 模式



RC 振荡器的频率是供电电压、电阻 (REXT)、电容 (CEXT) 值以及工作温度的函数。影响振荡器频率的因素还有：

- 阈值电压差异
- 元件公差
- 不同封装类型的电容差异

用户还需考虑由于所使用的外部 RC 元件的公差所引起的差异。

## 5.2.2 内部时钟源

通过执行以下某一操作，器件可以配置为使用内部振荡器模块作为系统时钟：

- 编程配置字寄存器 1 中的 FOSC<2:0> 位，以选择器件复位时用作默认系统时钟的 INTOSC 时钟源。
- 写入 OSCCON 寄存器中的 SCS<1:0> 位可以在运行时将系统时钟源切换为内部振荡器。更多信息，请参见第 5.3 节“时钟切换”。

在 INTOSC 模式下，OSC1/CLKIN 可用作通用 I/O 引脚。OSC2/CLKOUT 可用作通用 I/O 或 CLKOUT 引脚。

OSC2/CLKOUT 引脚的功能由配置字寄存器 1 中的 CLKOUTEN 位的状态决定。

内部振荡器模块具有两个独立的振荡器和一个专用的锁相环 HFPLL，用于产生三种内部系统时钟源中的一种。

1. **HFINTOSC**（高频内部振荡器）已经过厂家校准，工作频率为 16 MHz。HFINTOSC 时钟源由 500 kHz MFINTOSC 时钟源和专用锁相环 HFPLL 产生。用户可以通过软件使用 OSCTUNE 寄存器（寄存器 5-3）对 HFINTOSC 的频率进行调节。
2. **MFINTOSC**（中频内部振荡器）已经过厂家校准，工作频率为 500 kHz。用户可以通过软件使用 OSCTUNE 寄存器（寄存器 5-3）对 MFINTOSC 的频率进行调节。
3. **LFINTOSC**（低频内部振荡器）未经过厂家校准，工作频率为 31 kHz。

### 5.2.2.1 HFINTOSC

高频内部振荡器（HFINTOSC）是经过厂家校准、工作频率为 16 MHz 的内部时钟源。可以通过软件使用 OSCTUNE 寄存器（寄存器 5-3）对 HFINTOSC 的频率进行调节。

HFINTOSC 的输出与后分频器和多路开关相连（见图 5-1）。可以通过软件使用 OSCCON 寄存器的 IRCF<3:0> 位在 HFINTOSC 产生的九种频率中选择一种。更多信息，请参见第 5.2.2.7 节“内部振荡器时钟切换时序”。

HFINTOSC 可通过以下方式使能：

- 配置 OSCCON 寄存器的 IRCF<3:0> 位以选择所需的 HF 频率且
- FOSC<2:0> = 100，或
- 将 OSCCON 寄存器的系统时钟源（SCS）位设置为 1x

OSCSTAT 寄存器的高频内部振荡器就绪位（HFIOFR）指示 HFINTOSC 何时运行并可使用。

OSCSTAT 寄存器的高频内部振荡器状态锁定位（HFIOFL）指示 HFINTOSC 何时在距离其最终值 2% 的范围内运行。

OSCSTAT 寄存器的高频内部振荡器状态稳定位（HFIOFS）指示 HFINTOSC 何时在距离其最终值 0.5% 的范围内运行。

### 5.2.2.2 MFINTOSC

中频内部振荡器（MFINTOSC）是经过厂家校准、工作频率为 500 kHz 的内部时钟源。可以通过软件使用 OSCTUNE 寄存器（寄存器 5-3）对 MFINTOSC 的频率进行调节。

MFINTOSC 的输出与后分频器和多路开关相连（见图 5-1）。可以通过软件使用 OSCCON 寄存器的 IRCF<3:0> 位在 MFINTOSC 产生的九种频率中选择一种。更多信息，请参见第 5.2.2.7 节“内部振荡器时钟切换时序”。

MFINTOSC 可通过以下方式使能：

- 配置 OSCCON 寄存器的 IRCF<3:0> 位以选择所需的 HF 频率且
- FOSC<2:0> = 100，或
- 将 OSCCON 寄存器的系统时钟源（SCS）位设置为 1x

OSCSTAT 寄存器中的中频内部振荡器就绪位（MFIOFR）指示 MFINTOSC 何时运行并可使用。

## 5.2.2.3 内部振荡器频率调节

500 kHz 的内部振荡器经过厂家校准。通过写入 OSCTUNE 寄存器（寄存器 5-3），可以用软件调节此内部振荡器。由于 HFINTOSC 和 MFINTOSC 时钟源由 500 kHz 内部振荡器提供，因此对 OSCTUNE 寄存器值的更改还会应用到 HFINTOSC 和 MFINTOSC。

OSCTUNE 寄存器的默认值为 0。该值是一个 5 位的二进制补码。值 0Fh 将提供对最大频率的调节。值 10h 将提供对最小频率的调节。

当修改 OSCTUNE 寄存器时，振荡器频率将开始转变到新的频率。在频率转变期间代码继续执行。不会有任何迹象表明时钟发生了转变。

OSCTUNE 不影响 LFINTOSC 的频率。依赖 LFINTOSC 时钟源频率工作的部件，诸如上电延时定时器（PWRT）、看门狗定时器（WDT）、故障保护时钟监视器（FSCM）以及外设，它们的工作不受频率更改的影响。

## 5.2.2.4 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准、工作频率为 31 kHz 的内部时钟源。

LFINTOSC 的输出与后分频器和多路开关相连（见图 5-1）。可以通过软件使用 OSCCON 寄存器的 IRCF<3:0> 位选择 31 kHz 频率。更多信息，请参见第 5.2.2.7 节“内部振荡器时钟切换时序”。LFINTOSC 的频率也是上电延时定时器（PWRT）、看门狗定时器（WDT）和故障保护时钟监视器（FSCM）的时钟频率。

通过选择 31 kHz（OSCCON 寄存器的 IRCF<3:0> = 000）作为系统时钟源（OSCCON 寄存器的 SCS = 1x）或者使能以下任何一项都可以使能 LFINTOSC：

- 配置 OSCCON 寄存器的 IRCF<3:0> 位以选择所需的 LF 频率且
- FOSC<2:0> = 100，或
- 将 OSCCON 寄存器的系统时钟源（SCS）位设置为 1x

使用 LFINTOSC 的外设有：

- 上电延时定时器（PWRT）
- 看门狗定时器（WDT）
- 故障保护时钟监视器（FSCM）

OSCSTAT 寄存器的低频内部振荡器就绪位（LFIOFR）指示 LFINTOSC 何时运行并可使用。

## 5.2.2.5 内部振荡器频率选择

可以通过软件使用 OSCCON 寄存器的内部振荡器频率选择位 IRCF<3:0> 选择系统时钟速度。

16 MHz HFINTOSC 的输出和 31 kHz LFINTOSC 的输出与后分频器和多路开关相连（见图 5-1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<3:0> 选择内部振荡器的输出频率。通过软件可以选择以下频率中的一种：

- 32 MHz（需要 4X PLL）
- 16 MHz
- 8 MHz
- 4 MHz
- 2 MHz
- 1 MHz
- 500 kHz（复位后的默认频率）
- 250 kHz
- 125 kHz
- 62.5 kHz
- 31.25 kHz
- 31 kHz（LFINTOSC）

**注：** 发生任何复位后，OSCCON 寄存器的 IRCF<3:0> 位被设置为 0111 且频率选择被设置为 500 kHz。用户可以修改 IRCF 位以选择不同的频率。

OSCCON 的 IRCF<3:0> 位允许重复选择某些频率。这些重复选择使我们可以权衡系统设计。对于给定频率，更改振荡器源时可以获得更低的功耗。在使用同一振荡器源的两次频率更改之间可以获得更快的转换时间。

# PIC16(L)F1933

## 5.2.2.6 32 MHz 内部振荡器频率选择

内部振荡器模块可结合与外部振荡器模块关联的 4X PLL 一起使用，以产生 32 MHz 内部系统时钟源。使用 32 MHz 内部时钟源需要以下设置：

- 必须设置配置字寄存器 1 中的 FOSC 位，才能使用 INTOSC 源作为器件系统时钟（FOSC<2:0> = 100）。
- 要使用由配置字寄存器 1 中的 FOSC<2:0> 确定的时钟，必须清零 OSCCON 寄存器中的 SCS 位（SCS<1:0> = 00）。
- 必须将 OSCCON 寄存器中的 IRCF 位设置为使用 8 MHz HFINTOSC 设置（IRCF<3:0> = 1110）。
- OSCCON 寄存器中的 SPLLEN 位必须置 1 才能使能 4xPLL，或者配置字寄存器 2 的 PLEN 位必须编程为 1。

**注：** 当使用配置字寄存器 2 的 PLEN 位时，4xPLL 不能通过软件禁止，8 MHz HFINTOSC 选项不再可用。

当 OSCCON 寄存器中的 SCS 位设置为 1x 时，4xPLL 不能与内部振荡器一起使用。SCS 位必须设置为 00，4xPLL 才能与内部振荡器一起使用。

## 5.2.2.7 内部振荡器时钟切换时序

当在 HFINTOSC、MFINTOSC 和 LFINTOSC 之间切换时，新的振荡器可能已经被关闭以节省功耗（见图 5-7）。如果情况是这样，在 OSCCON 寄存器的 IRCF<3:0> 位被修改后到频率选择生效之前将有一个延时。OSCSTAT 寄存器可反映 HFINTOSC、MFINTOSC 和 LFINTOSC 振荡器的当前工作状态。频率选择的步骤如下：

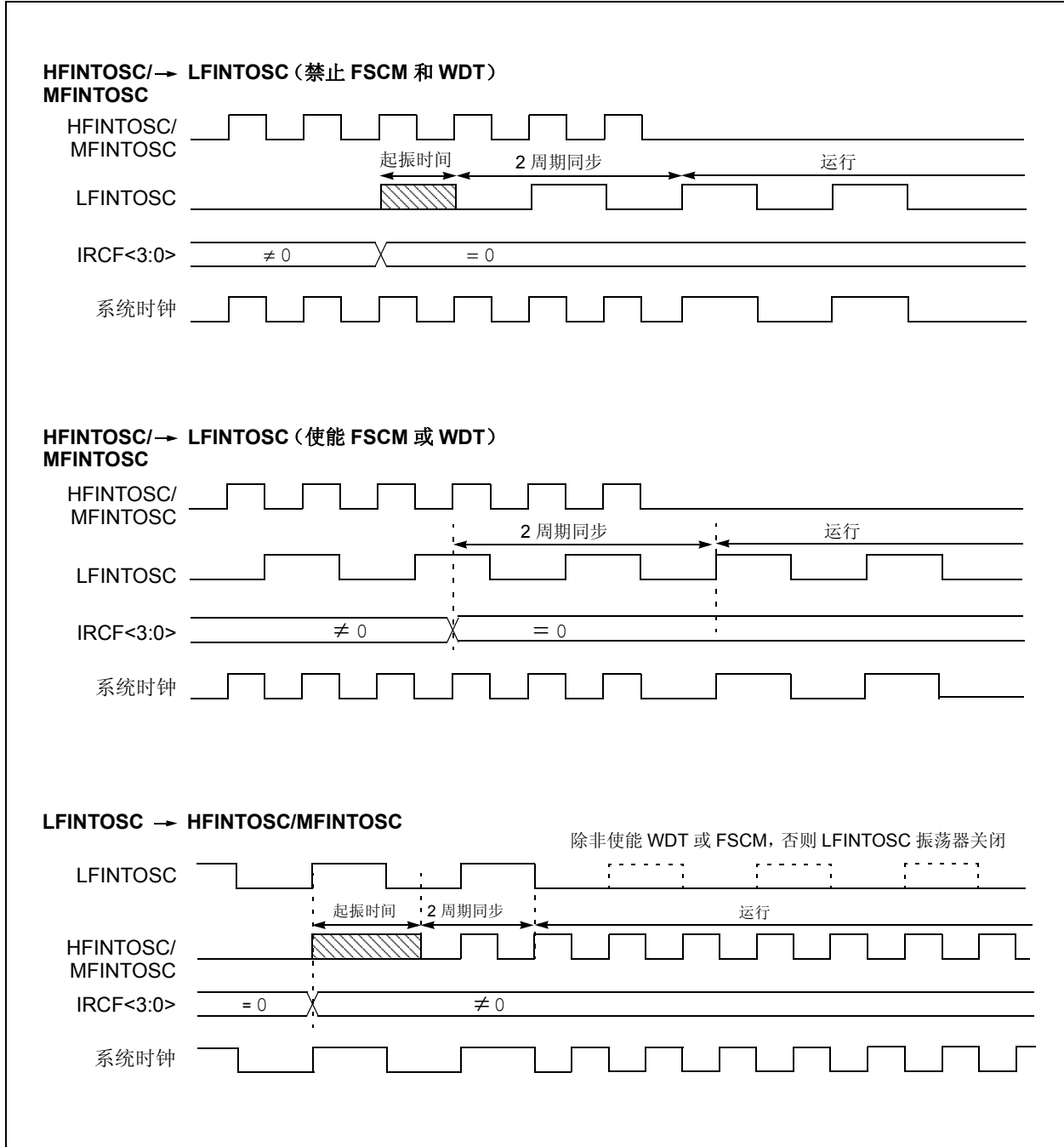
1. 修改 OSCCON 寄存器的 IRCF<3:0> 位。
2. 如果新的时钟已关闭，将开始一段时钟启动延时。
3. 时钟切换电路等待当前时钟的下降沿。
4. 当前时钟保持低电平，时钟切换电路等待新时钟的上升沿。
5. 新时钟立即有效。
6. 按照要求更新 OSCSTAT 寄存器。
7. 时钟切换完成。

更多详细信息，请参见图 5-7。

如果内部振荡器速度在相同时钟源的两个时钟之间切换，那么在选定新的频率之前将没有起振延时。时钟切换延时如表 5-1 所示。

应用的电气规范章节中的振荡器表给出了起振延时规范。

图 5-7: 内部振荡器切换时序



## 5.3 时钟切换

可通过软件使用 OSCCON 寄存器的系统时钟选择 (SCS) 位在外部和内部时钟源之间切换系统时钟源。使用 SCS 位可以选择以下时钟源：

- 默认系统振荡器由配置字寄存器 1 中的 FOSC 位决定
- Timer1 32 kHz 晶振
- 内部振荡器模块 (INTOSC)

### 5.3.1 系统时钟选择 (SCS) 位

OSCCON 寄存器的系统时钟选择 (SCS) 位选择供 CPU 和外设使用的系统时钟源。

- 当 OSCCON 寄存器的 SCS 位 = 00 时，系统时钟源由配置字寄存器 1 中的 FOSC<2:0> 位的值决定。
- 当 OSCCON 寄存器的 SCS 位 = 01 时，系统时钟源是 Timer1 振荡器。
- 当 OSCCON 寄存器的 SCS 位 = 1x 时，系统时钟源由 OSCCON 寄存器的 IRCF<3:0> 位选择的内部振荡器频率决定。复位后，OSCCON 寄存器的 SCS 位始终清零。

**注：** 可能由双速启动或故障保护时钟监视器引起的任何自动时钟切换将不会更新 OSCCON 寄存器的 SCS 位。用户可以监视 OSCSTAT 寄存器的 OST5 位来确定当前的系统时钟源。

在不同时钟源之间切换时，需要一个延时以使新的时钟稳定下来。表 5-1 中显示了这些振荡器延时。

### 5.3.2 振荡器起振延时状态 (OSTS) 位

OSCSTAT 寄存器的振荡器起振延时状态 (OSTS) 位表明系统时钟是来自于由配置字寄存器 1 中 FOSC<2:0> 位定义的外部时钟源还是来自于内部时钟源。特别是，当处于 LP、XT 或 HS 模式时，OSTS 表示振荡器起振定时器 (OST) 已经超时。OST 不反映 Timer1 振荡器的状态。

### 5.3.3 TIMER1 振荡器

Timer1 振荡器是与 Timer1 外设关联的独立晶振。通过 T1OSO 和 T1OSI 器件引脚之间连接的 32.768 kHz 晶振，对 Timer1 振荡器的计时操作进行了优化。

可使用 T1CON 寄存器中的 T1OSCEEN 控制位使能 Timer1 振荡器。更多关于 Timer1 外设的信息，请参见第 21.0 节“带门控的 Timer1 模块”。

### 5.3.4 TIMER1 振荡器就绪 (T1OSCR) 位

用户必须确保 Timer1 振荡器已就绪可供使用，然后才能将其选择为系统时钟源。OSCSTAT 寄存器的 Timer1 振荡器就绪 (T1OSCR) 位表示 Timer1 振荡器是否就绪，可以使用。在 T1OSCR 位置 1 时，可配置 SCS 位以选择 Timer1 振荡器。



## 5.4 双速时钟启动模式

双速时钟启动模式通过使外部振荡器起振到代码执行之间的延时达到最小而进一步降低功耗。在大量使用休眠模式的应用中，双速启动将使唤醒所花费的时间中不包含外部振荡器起振时间，并能降低器件的总功耗。此模式允许应用从休眠模式唤醒，使用 INTOSC 内部振荡器模块作为时钟源来执行一些指令，然后返回休眠模式，而无需等待外部振荡器稳定。

当振荡器模块配置为 LP、XT 或 HS 模式时，双速启动可提供许多优势。为这些模式使能振荡器起振定时器 (OST)，并在振荡器可用作系统时钟源之前必须计满 1024 次振荡。

如果振荡器模块被配置为除 LP、XT 或 HS 模式以外的任何其他模式，那么双速启动将被禁止。这是因为在上电复位后或从休眠模式退出后，外部时钟振荡器将不需要任何稳定时间。

如果在器件进入休眠模式之前，OST 计数已达到 1024，那么 OSCSTAT 寄存器的 OSTS 位置 1，并且程序执行切换到由外部振荡器充当时钟源。但是，如果唤醒所花的时间非常短，系统可能不再使用外部振荡器充当时钟源。

**注：** 执行 SLEEP 指令将中止振荡器起振延时并将使 OSCSTAT 寄存器的 OSTS 位保持清零。

### 5.4.1 双速启动模式配置

通过下列设置配置双速启动模式：

- IESO（配置字寄存器 1 中）= 1；内部 / 外部切换位（使能双速启动模式）。
- SCS（OSCCON 寄存器中）= 00。
- 将配置字寄存器 1 中的 FOSC<2:0> 位配置为 LP、XT 或 HS 模式。

发生下列事件后进入双速启动模式：

- 上电复位（POR）后、上电延时定时器（PWRT）延时结束（如果使能）后，或者
- 从休眠模式唤醒。

**表 5-1: 振荡器切换延时**

切换自	切换为	频率	振荡器延时
休眠 / 上电复位	LFINTOSC <sup>(1)</sup> MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31 kHz 31.25 kHz-500 kHz 31.25 kHz-16 MHz	振荡器预热延时 (TWARM)
休眠 / 上电复位	EC 和 RC <sup>(1)</sup>	DC – 32 MHz	2 个周期
LFINTOSC	EC 和 RC <sup>(1)</sup>	DC – 32 MHz	每种模式都为 1 个周期
休眠 / 上电复位	Timer1 振荡器 LP、XT 和 HS <sup>(1)</sup>	32 kHz-20 MHz	1024 个时钟周期 (OST)
任何时钟源	MFINTOSC <sup>(1)</sup> HFINTOSC <sup>(1)</sup>	31.25 kHz-500 kHz 31.25 kHz-16 MHz	2 μs (大约)
任何时钟源	LFINTOSC <sup>(1)</sup>	31 kHz	每种模式都为 1 个周期
任何时钟源	Timer1 振荡器	32 kHz	1024 个时钟周期 (OST)
PLL 无效	PLL 有效	16-32 MHz	2 ms (大约)

**注 1:** PLL 无效。

# PIC16(L)F1933

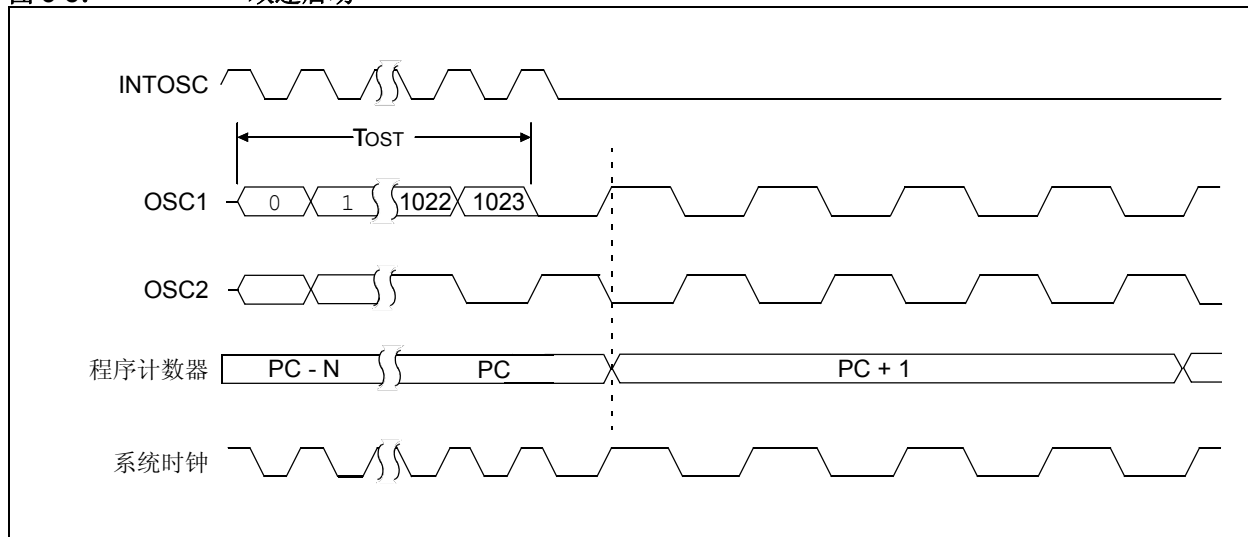
## 5.4.2 双速启动序列

1. 发生上电复位或从休眠模式唤醒。
2. 以内部振荡器作为时钟源（以 OSCCON 寄存器的 IRCF<3:0> 位设置的频率）开始执行指令。
3. 使能 OST 计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器的下降沿。
5. OSTS 置 1。
6. 系统时钟保持低电平直到新时钟的下一个下降沿（LP、XT 或 HS 模式）。
7. 系统时钟切换到外部时钟源。

## 5.4.3 检查双速时钟状态

检查 OSCSTAT 寄存器的 OSTS 位的状态将确定单片机是在使用由配置字寄存器 1 中的 FOSC<2:0> 位定义的外部时钟源还是内部振荡器。

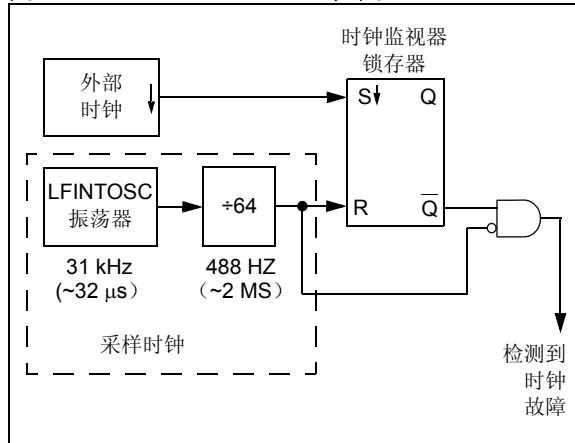
图 5-8: 双速启动



## 5.5 故障保护时钟监视器

故障保护时钟监视器（FSCM）旨在使器件能在外部振荡器发生故障时继续运行。FSCM 可以检测当振荡器起振定时器（OST）延时结束后的任何时刻发生的振荡器故障。通过将配置字寄存器 1 中的 FCMEN 位置 1 使能 FSCM。FSCM 适用于所有外部振荡器模式（LP、XT、HS、EC、Timer1 振荡器和 RC）。

图 5-9: FSCM 框图



### 5.5.1 故障保护检测

FSCM 模块通过将外部振荡器和 FSCM 采样时钟进行比较来检测有故障的振荡器。通过对 LFINTOSC 时钟进行 64 分频得到采样时钟。请参见图 5-9。故障检测电路内部有一个锁存器。在外部时钟的每个下降沿上将锁存器置 1。在采样时钟的每个上升沿将锁存器清零。如果采样时钟的一个完整半周期在外部时钟变为低电平之前结束，则将检测到故障。

### 5.5.2 故障保护工作原理

当外部时钟出现故障时，FSCM 会将器件时钟切换到内部时钟源并将 PIR2 寄存器的 OSFIF 标志位置 1，且如果 PIE2 寄存器的 OSFIE 位也置 1，将产生中断。器件固件可采取措施以减轻可能由故障时钟造成的问题。系统时钟将继续采用内部时钟源，直到器件固件成功地重启外部振荡器并使时钟重新切换到外部振荡器为止。

FSCM 选择的内部时钟源由 OSCCON 寄存器中的 IRCF<3:0> 位决定，从而允许在故障发生之前配置内部振荡器。

### 5.5.3 清除故障保护条件

在复位后、执行了 SLEEP 指令或者修改了 OSCCON 寄存器中的 SCS 位后将清除故障保护条件。如果修改 SCS 位，OST 将重新起振。OST 运行时，器件将继续使用在 OSCCON 中选择的 INTOSC 作为系统时钟。OST 超时后，故障保护条件会被清除，器件将使用外部时钟源作为系统时钟。必须先清除故障保护条件才能清零 OSFIF 标志位。

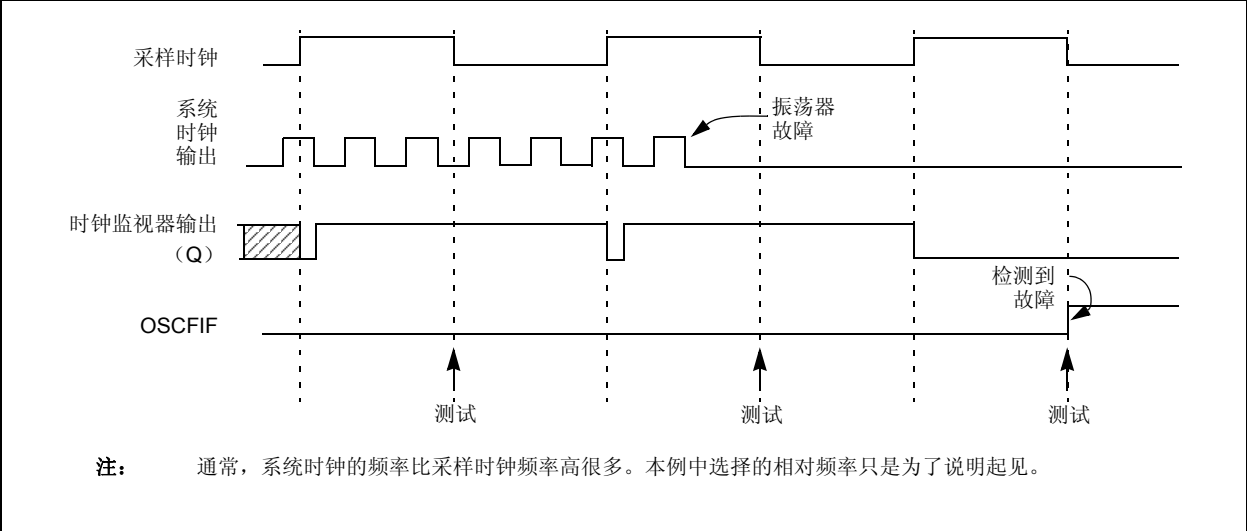
### 5.5.4 复位或从休眠模式唤醒

FSCM 设计为在振荡器起振定时器（OST）延时结束后检测振荡器故障。从休眠模式唤醒和任何类型的复位之后会启动 OST。OST 不与 EC 或 RC 时钟模式一起使用，因此在这两种模式下 FSCM 将在复位或唤醒后立即生效。使能 FSCM 也将使能双速启动。因此，在 OST 运行时，器件将始终执行代码。

**注：** 由于振荡器的起振时间范围很广，故障保护电路在振荡器起振期间（即，从复位或休眠模式退出后）不工作。在一段适当的时间后，用户应该检查 OSCSTAT 寄存器的状态位，以验证振荡器起振和系统时钟切换是否已经成功完成。

# PIC16(L)F1933

图 5-10: FSCM 时序图



## 5.6 振荡器控制寄存器

寄存器 5-1: **OSCCON: 振荡器控制寄存器**

R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1	R/W-1/1	U-0	R/W-0/0	R/W-0/0
SPLLEN	IRCF<3:0>			—	SCS<1:0>		
bit 7						bit 0	

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **SPLLEN:** 软件 PLL 使能位  
 如果配置字寄存器 1 中的 **PLLEN = 1:**  
 忽略 **SPLLEN** 位。始终使能 4x PLL (根据振荡器要求)  
 如果配置字寄存器 1 中的 **PLLEN = 0:**  
 1 = 使能 4x PLL  
 0 = 禁止 4x PLL
- bit 6-3    **IRCF<3:0>:** 内部振荡器频率选择位  
 000x = 31 kHz LF  
 0010 = 31.25 kHz MF  
 0011 = 31.25 kHz HF<sup>(1)</sup>  
 0100 = 62.5 kHz MF  
 0101 = 125 kHz MF  
 0110 = 250 kHz MF  
 0111 = 500 kHz MF (复位时的默认频率)  
 1000 = 125 kHz HF<sup>(1)</sup>  
 1001 = 250 kHz HF<sup>(1)</sup>  
 1010 = 500 kHz HF<sup>(1)</sup>  
 1011 = 1 MHz HF  
 1100 = 2 MHz HF  
 1101 = 4 MHz HF  
 1110 = 8 MHz 或 32 MHz HF (见第 5.2.2.1 节 “HFINTOSC”)  
 1111 = 16 MHz HF
- bit 2      未实现: 读为 0
- bit 1-0    **SCS<1:0>:** 系统时钟选择位  
 1x = 内部振荡器模块  
 01 = Timer1 振荡器  
 00 = 时钟由配置字寄存器 1 中的 **FOSC<2:0>** 决定。

注 1: 重复频率来自 HFINTOSC。

# PIC16(L)F1933

## 寄存器 5-2: OSCSTAT: 振荡器状态寄存器

R-1/q	R-0/q	R-q/q	R-0/q	R-0/q	R-q/q	R-0/0	R-0/q
T1OSCR	PLL	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 取值视条件而定

bit 7 **T1OSCR:** Timer1 振荡器就绪位

如果 T1OSCCN = 1:

1 = Timer1 振荡器就绪

0 = Timer1 振荡器未就绪

如果 T1OSCCN = 0:

1 = Timer1 时钟源始终就绪

bit 6 **PLL:** 4x PLL 就绪位

1 = 4x PLL 就绪

0 = 4x PLL 未就绪

bit 5 **OSTS:** 振荡器起振延时状态位

1 = 使用配置字寄存器 1 的 FOSC<2:0> 位定义的时钟运行

0 = 使用内部振荡器 (FOSC<2:0> = 100) 运行

bit 4 **HFIOFR:** 高频内部振荡器就绪位

1 = HFINTOSC 就绪

0 = HFINTOSC 未就绪

bit 3 **HFIOFL:** 高频内部振荡器锁定位

1 = HFINTOSC 的精度至少为 2%

0 = HFINTOSC 的精度不是 2%

bit 2 **MFIOFR:** 中频内部振荡器就绪位

1 = MFINTOSC 就绪

0 = MFINTOSC 未就绪

bit 1 **LFIOFR:** 低频内部振荡器就绪位

1 = LFINTOSC 就绪

0 = LFINTOSC 未就绪

bit 0 **HFIOFS:** 高频内部振荡器稳定位

1 = HFINTOSC 的精度至少为 0.5%

0 = HFINTOSC 的精度不是 0.5%

**寄存器 5-3: OSCTUNE: 振荡器调节寄存器**

U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	TUN<5:0>					
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

bit 7-6            **未实现:** 读为 0  
bit 5-0            **TUN<5:0>:** 频率调节位  
011111 = 最高频率  
011110 =  
•  
•  
•  
000001 =  
000000 = 振荡器模块以厂家校准后的频率运行  
111111 =  
•  
•  
•  
100000 = 最低频率

**表 5-2: 与时钟源相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69
OSCSTAT	T1OSCR	PLLRR	OSTS	HFIOFR	HFIOFL	MFIOFR	LFIOFR	HFIOFS	70
OSCTUNE	—	—	TUN<5:0>						71
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185

**图注:** — = 未实现单元, 读为 0。时钟源不使用阴影单元。

**表 5-3: 与时钟源相关的配置字寄存器汇总**

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在的页面
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		CPD	50
	7:0	CP	MCLRE	PWRTÉ	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	—	BORV	STVREN	PLLEN	52
	7:0	—	—	VCAPEN<1:0> <sup>(1)</sup>		—	—	WRT<1:0>		

**图注:** — = 未实现单元, 读为 0。时钟源不使用阴影单元。

**注 1:** 仅限 PIC16F1933 器件。

# PIC16(L)F1933

---

注:



## 6.0 复位

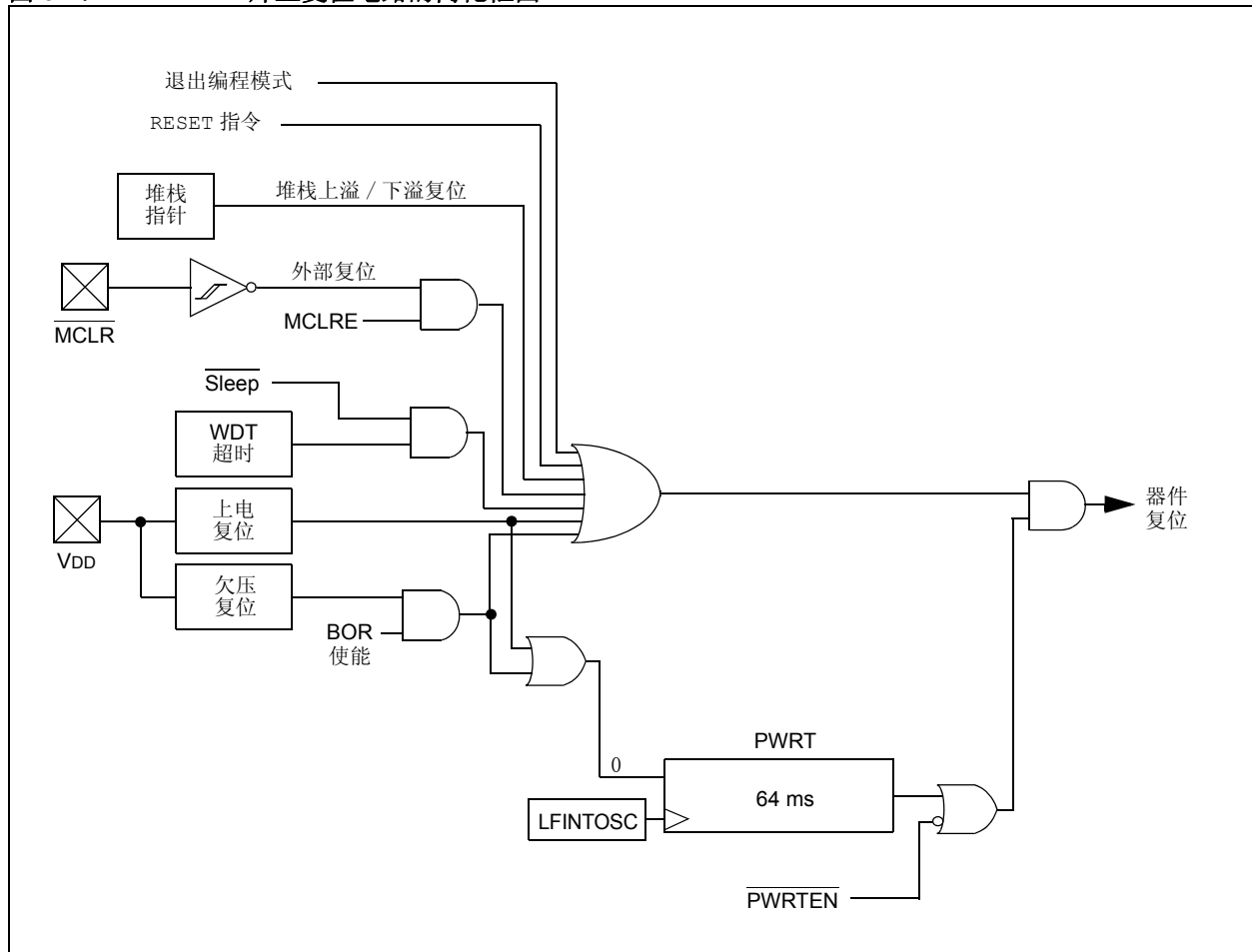
有多种方式可以复位器件：

- 上电复位（POR）
- 欠压复位（BOR）
- MCLR 复位
- WDT 复位
- RESET 指令
- 堆栈上溢
- 堆栈下溢
- 退出编程模式

要使 VDD 稳定，可以使能可选的上电复位定时器，以在 BOR 或 POR 事件后延长复位时间。

图 6-1 给出了片上复位电路的简化框图。

图 6-1: 片上复位电路的简化框图



# PIC16(L)F1933

## 6.1 上电复位 (POR)

POR 电路将器件保持在复位状态，直到 VDD 达到足以使器件正常工作的最低电平。VDD 上升缓慢、工作速度快或模拟性能可能要求工作电压大于最小 VDD。在满足所有器件操作条件之前，可以使用 PWRT、BOR 或 MCLR 功能延长启动周期。

### 6.1.1 上电延时定时器 (PWRT)

上电延时定时器在 POR 或欠压复位时提供一个标称值为 64 ms 的延时。

只要 PWRT 处于工作状态，器件就保持在复位状态。PWRT 延时为 VDD 上升到所需的电平提供额外的时间。通过清零配置字寄存器 1 中的 PWRTE 位可以使能上电延时定时器。

上电延时定时器在从 POR 和 BOR 恢复后启动。

更多信息，请参见应用笔记 AN607“Power-up Trouble Shooting” (DS00607)。

## 6.2 欠压复位 (BOR)

当 VDD 到达可选的最低电平时，BOR 电路将器件保持在复位状态。在 BOR 和 POR 之间的整个电压范围内，可实现执行保护功能。

欠压复位模块有 4 种工作模式，由配置字寄存器 1 中的 BOREN<1:0> 位控制。这 4 种模式是：

- BOR 始终使能
- BOR 在休眠时禁止
- BOR 由软件控制
- BOR 始终禁止

更多信息，请参见表 6-3。

通过配置配置字寄存器 2 中的 BORV 位，可以选择欠压复位电压级别。

VDD 噪声抑制滤波器可以防止在发生小事件时触发 BOR。如果 VDD 下降到 VBOR 以下，且持续时间大于参数 TBORDC，器件将复位。更多信息，请参见图 6-2。

表 6-1: BOR 的工作模式

BOREN<1:0>	SBOREN	器件模式	BOR 模式	从 POR 恢复时的器件操作	从休眠模式下唤醒时的器件操作
11	X	X	有效	等待 BOR 就绪 <sup>(1)</sup>	
10	X	唤醒	有效	等待 BOR 就绪	
		休眠	禁止		
01	1	X	有效	立即开始	
	0		禁止	立即开始	
00	X	X	禁止	立即开始	

注 1: 尽管这种情况下尤其需要等待 BOR 就绪，但 BOR 已经在运行，所以启动时没有延时。

### 6.2.1 BOR 始终使能

配置字寄存器 1 的 BOREN 位设置为 11 时，BOR 始终使能。器件启动延时直到 BOR 就绪并且 VDD 高于 BOR 阈值。

BOR 保护功能在休眠期间有效。BOR 不会延时从休眠唤醒。

### 6.2.2 BOR 在休眠时禁止

配置字寄存器 1 的 BOREN 位设置为 10 时，BOR 除了在休眠时之外，一直使能。器件启动延时直到 BOR 就绪并且 VDD 高于 BOR 阈值。

BOR 保护功能在休眠期间无效。器件唤醒延时直到 BOR 就绪。

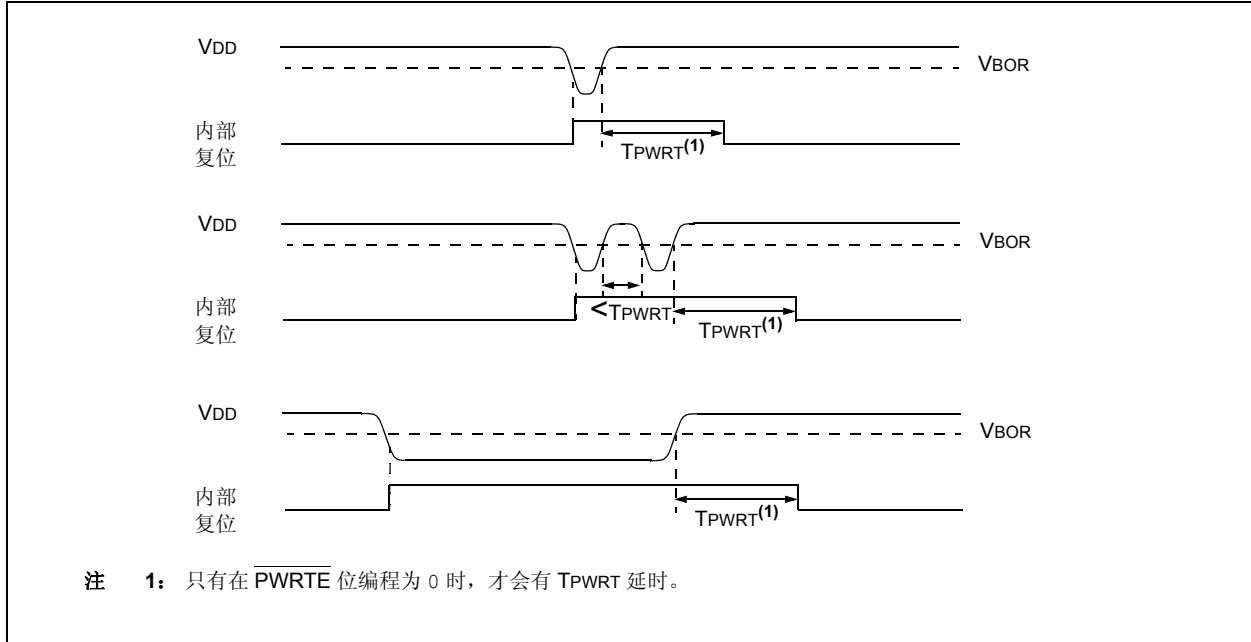
### 6.2.3 BOR 由软件控制

配置字寄存器 1 的 BOREN 位设置为 01 时，BOR 由 BORCON 寄存器的 SBOREN 位控制。器件启动不会被 BOR 就绪条件或 VDD 电平延时。

只要 BOR 电路就绪，就开始进行 BOR 保护。BOR 电路的状态通过 BORCON 寄存器的 BORRDY 位反映。

BOR 保护功能不受休眠影响。

图 6-2: 欠压情形



寄存器 6-1: **BORCON: 欠压复位控制寄存器**

R/W-1/u	U-0	U-0	U-0	U-0	U-0	U-0	R-q/u
SBOREN	—	—	—	—	—	—	BORRDY
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 取值视条件而定

bit 7

**SBOREN:** 软件欠压复位使能位

如果配置字寄存器 1 中的  $\text{BOREN} \langle 1:0 \rangle \neq 01$ :

SBOREN 位是可读写的, 但是对欠压复位没有影响。

如果配置字寄存器 1 中的  $\text{BOREN} \langle 1:0 \rangle = 01$ :

1 = 使能 BOR

0 = 禁止 BOR

bit 6-1

未实现: 读为 0

bit 0

**BORRDY:** 欠压复位电路就绪状态位

1 = 欠压复位电路有效

0 = 欠压复位电路无效

# PIC16(L)F1933

## 6.3 MCLR

MCLR 是可复位器件的可选外部输入。MCLR 功能由配置字寄存器 1 的 MCLRE 位和配置字寄存器 2 的 LVP 位控制（表 6-2）。

表 6-2: MCLR 配置

MCLRE	LVP	MCLR
0	0	禁止
1	0	使能
x	1	使能

### 6.3.1 MCLR 使能

MCLR 使能且引脚保持低电平时，器件保持在复位状态。MCLR 引脚通过内部弱上拉连接到 VDD。

器件在 MCLR 复位路径中有一个噪声滤波器。该滤波器能检测并滤除小脉冲。

**注：** 复位未驱动 MCLR 引脚为低电平。

### 6.3.2 MCLR 禁止

MCLR 禁止时，引脚用作通用输入，内部弱上拉由软件控制。更多信息，请参见第 12.5 节“PORTE 寄存器”。

## 6.4 看门狗定时器（WDT）复位

如果固件在超时周期内未发出 CLRWDWT 指令，看门狗定时器将产生复位。更改 STATUS 寄存器中的 TO 和 PD 位以指示 WDT 复位。更多信息，请参见第 10.0 节“看门狗定时器”。

## 6.5 RESET 指令

RESET 指令将导致器件复位。PCON 寄存器中的 RI 位将复位为 0。请参见表 6-4 以了解执行 RESET 指令后的默认条件。

## 6.6 堆栈上溢 / 下溢复位

堆栈上溢 / 下溢时，器件复位。PCON 寄存器的 STKOVF 或 STKUNF 位指示复位条件。通过将配置字寄存器 2 中的 STVREN 位置 1 可以使能这些复位。更多信息请参见第 3.4.2 节“上溢 / 下溢复位”。

## 6.7 退出编程模式

退出编程模式时，器件执行的操作与发生 POR 时一样。

## 6.8 上电延时定时器

上电延时定时器可以在 BOR 或 POR 事件后延迟器件执行。此定时器通常用于使 VDD 在器件开始运行之前稳定下来。

上电延时定时器由配置字寄存器 1 中的 PWRTE 位控制。

## 6.9 启动序列

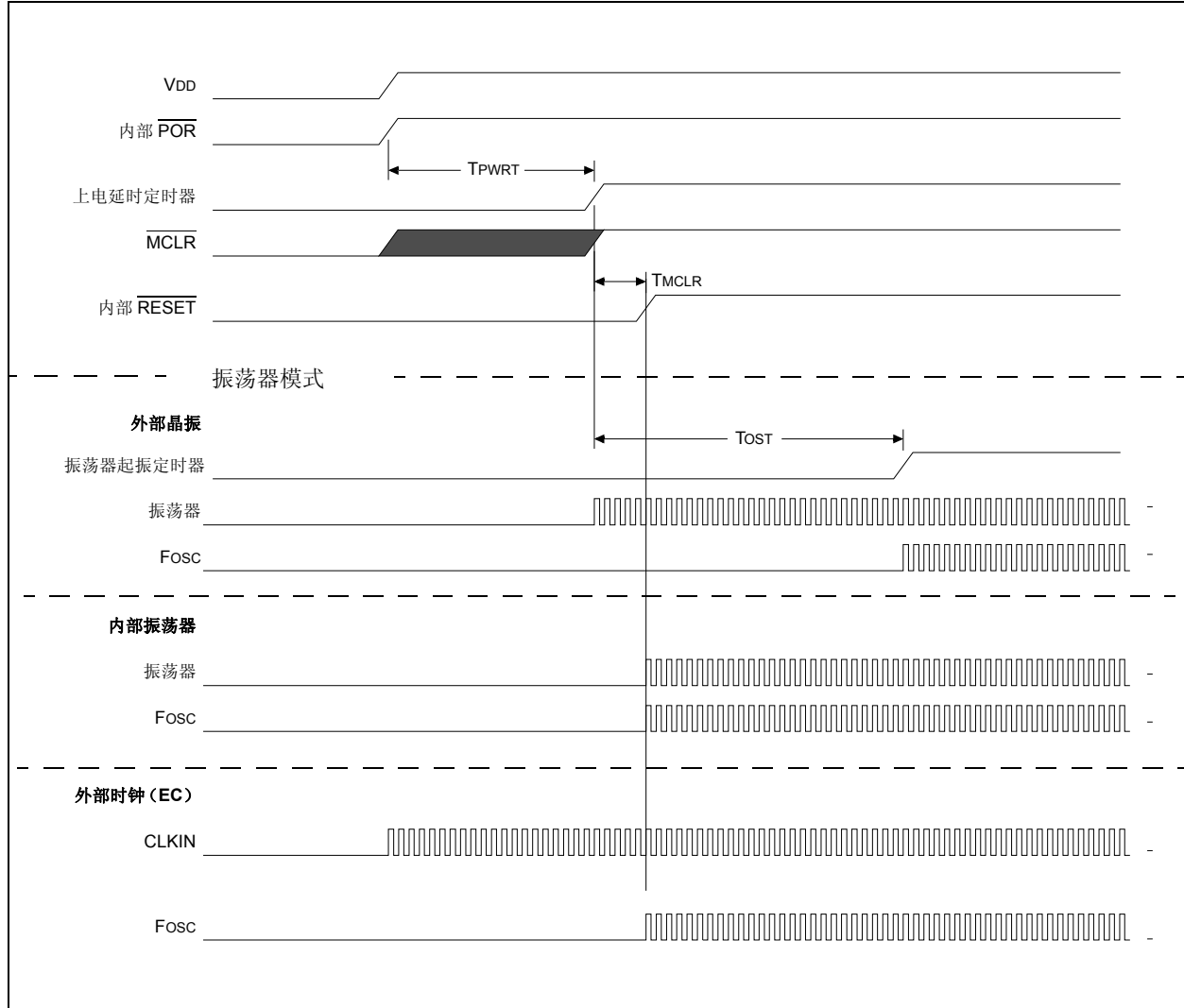
从 POR 或 BOR 恢复时，在器件开始执行之前必须发生以下事件：

1. 上电延时定时器运行结束（如果被使能）。
2. 振荡器起振定时器运行结束（如果振荡器源需要）。
3. 必须释放 MCLR（如果被使能）。

总超时时间将根据振荡器配置和上电延时定时器配置而变化。更多信息，请参见第 5.0 节“振荡器模块（带故障保护时钟监视器）”。

上电延时定时器和振荡器起振定时器独立于 MCLR 复位运行。如果 MCLR 保持足够长时间的低电平，上电延时定时器和振荡器起振定时器将超时。将 MCLR 电平拉高后器件将立即开始执行（见图 6-3）。这对于测试或同步多个并行工作的器件来说非常有用。

图 6-3: 复位启动时序



# PIC16(L)F1933

## 6.10 确定复位原因

发生任何复位时，STATUS 和 PCON 寄存器中的多个位都会更新以指示复位原因。表 6-3 和表 6-4 给出了这些寄存器的复位条件。

表 6-3: 复位状态位及其含义

STKOVF	STKUNF	RMCLR	RI	POR	BOR	TO	PD	条件
0	0	1	1	0	x	1	1	上电复位
0	0	1	1	0	x	0	x	非法， $\overline{TO}$ 在 $\overline{POR}$ 时置 1
0	0	1	1	0	x	x	0	非法， $\overline{PD}$ 在 $\overline{POR}$ 时置 1
0	0	1	1	u	0	1	1	欠压复位
u	u	u	u	u	u	0	u	WDT 复位
u	u	u	u	u	u	0	0	WDT 从休眠模式下唤醒
u	u	u	u	u	u	1	0	通过中断从休眠模式唤醒
u	u	0	u	u	u	u	u	正常工作期间的 $\overline{MCLR}$ 复位
u	u	0	u	u	u	1	0	休眠期间的 $\overline{MCLR}$ 复位
u	u	u	0	u	u	u	u	执行了 RESET 指令
1	u	u	u	u	u	u	u	堆栈上溢复位 (STVREN = 1)
u	1	u	u	u	u	u	u	堆栈下溢复位 (STVREN = 1)

表 6-4: 特殊寄存器的复位条件<sup>(2)</sup>

条件	程序计数器	STATUS 寄存器	PCON 寄存器
上电复位	0000h	---1 1000	00-- 110x
正常工作期间的 $\overline{MCLR}$ 复位	0000h	---u uuuu	uu-- 0uuu
休眠期间的 $\overline{MCLR}$ 复位	0000h	---1 0uuu	uu-- 0uuu
WDT 复位	0000h	---0 uuuu	uu-- uuuu
WDT 从休眠模式下唤醒	PC + 1	---0 0uuu	uu-- uuuu
欠压复位	0000h	---1 1uuu	00-- 11u0
通过中断从休眠模式唤醒	PC + 1 <sup>(1)</sup>	---1 0uuu	uu-- uuuu
执行了 RESET 指令	0000h	---u uuuu	uu-- u0uu
堆栈上溢复位 (STVREN = 1)	0000h	---u uuuu	1u-- uuuu
堆栈下溢复位 (STVREN = 1)	0000h	---u uuuu	u1-- uuuu

图注: u = 不变, x = 未知, - = 未实现位, 读为 0。

注 1: 如果器件被中断唤醒且全局中断允许位 (GIE) 置 1, 则执行 PC + 1 后, 返回地址被压入堆栈且 PC 装入中断向量 (0004h)。

注 2: 如果状态位未实现, 则该位读为 0。

## 6.11 电源控制 (PCON) 寄存器

电源控制 (PCON) 寄存器包含区分以下复位的标志位:

- 上电复位 ( $\overline{\text{POR}}$ )
- 欠压复位 ( $\overline{\text{BOR}}$ )
- RESET 指令复位 ( $\overline{\text{RI}}$ )
- 堆栈上溢复位 (STKOVF)
- 堆栈下溢复位 (STKUNF)
- MCLR 复位 ( $\overline{\text{RMCLR}}$ )

PCON 寄存器位如寄存器 6-2 所示。

寄存器 6-2: PCON: 电源控制寄存器

R/W/HS-0/q	R/W/HS-0/q	U-0	U-0	R/W/HC-1/q	R/W/HC-1/q	R/W/HC-q/u	R/W/HC-q/u
STKOVF	STKUNF	—	—	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

**图注:**

HC = 由硬件清零

HS = 由硬件置 1

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-m/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

q = 取值视条件而定

- bit 7      **STKOVF:** 堆栈上溢标志位  
 1 = 发生了堆栈上溢  
 0 = 未发生堆栈上溢, 或该位由固件设置为 0
- bit 6      **STKUNF:** 堆栈下溢标志位  
 1 = 发生了堆栈下溢  
 0 = 未发生堆栈下溢, 或该位由固件设置为 0
- bit 5-4    **未实现:** 读为 0
- bit 3       **$\overline{\text{RMCLR}}$ :**  $\overline{\text{MCLR}}$  复位标志位  
 1 = 未发生  $\overline{\text{MCLR}}$  复位, 或该位由固件设置为 1  
 0 = 发生了  $\overline{\text{MCLR}}$  复位 (当发生  $\overline{\text{MCLR}}$  复位时, 该位由硬件设置为 0)
- bit 2       **$\overline{\text{RI}}$ :** RESET 指令标志位  
 1 = 未执行 RESET 指令, 或该位由固件设置为 1  
 0 = 已执行 RESET 指令 (当执行 RESET 指令时, 该位由硬件设置为 0)
- bit 1      **POR:** 上电复位状态位  
 1 = 未发生上电复位  
 0 = 发生了上电复位 (在上电复位发生后必须用软件置 1)
- bit 0      **BOR:** 欠压复位状态位  
 1 = 未发生欠压复位  
 0 = 发生了欠压复位 (在上电复位或欠压复位发生后必须用软件置 1)

# PIC16(L)F1933

表 6-5: 与复位相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BORCON	SBOREN	—	—	—	—	—	—	BORRDY	75
PCON	STKOVF	STKUNF	—	—	$\overline{\text{RMCLR}}$	$\overline{\text{RI}}$	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	79
STATUS	—	—	—	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	22
WDTCON	—	—	WDTPS<4:0>					SWDTEN	101

图注: — = 未实现单元, 读为 0。复位不使用阴影单元。

注 1: 其他 (非上电) 复位包括正常工作时的  $\overline{\text{MCLR}}$  复位和看门狗定时器复位。



## 7.0 中断

中断功能允许某些事件的优先级高于正常的程序流程。固件用于判断中断源并采取相应的操作。一些中断还可配置为将 MCU 从休眠模式唤醒。

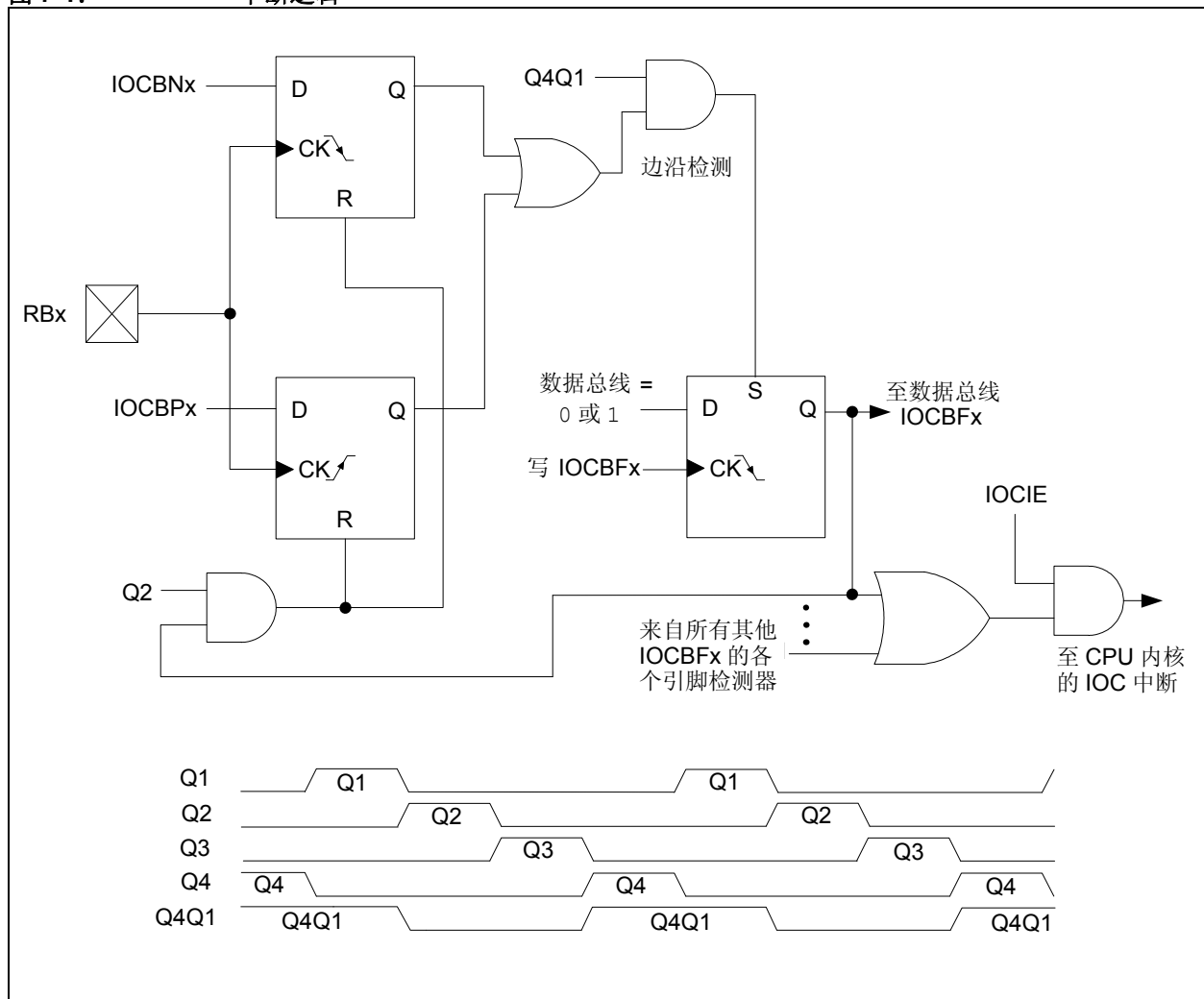
本章包含以下中断信息：

- 工作原理
- 中断响应延时
- 休眠期间的中断
- INT 引脚
- 自动现场保护

许多外设可以产生中断。详细信息请参见相应的章节。

图 7-1 给出了中断逻辑的框图。

图 7-1: 中断逻辑



# PIC16(L)F1933

## 7.1 工作原理

任何器件复位都将禁止中断。可通过将以下位置 1 来允许中断：

- INTCON 寄存器的 GIE 位
- 特殊中断事件的中断允许位
- INTCON 寄存器的 PEIE 位（如果中断事件的中断允许位包含在 PIE1、PIE2 和 PIE3 寄存器中）

INTCON、PIR1、PIR2 和 PIR3 寄存器通过中断标志位来记录各个中断。无论 GIE、PEIE 和各个中断允许位的状态如何，中断标志位都会在中断发生时置 1。

当 GIE 位置 1 时，中断事件的发生会引发以下事件：

- 清除当前预取的指令
- GIE 位清零
- 当前程序计数器（PC）值被压入堆栈
- 重要寄存器的内容自动保存到影子寄存器（见第 7.5 节“自动现场保护”）
- PC 装载中断向量 0004h

中断服务程序（Interrupt Service Routine, ISR）中的固件应该通过查询中断标志位来确定中断源。在退出 ISR 之前必须将中断标志位清零，以避免重复的中断。由于 GIE 位被清零，所以执行 ISR 期间发生的任何中断将会通过其中断标志进行记录，但是不会使处理器重定向到中断向量。

RETFIE 指令通过将先前地址从堆栈中弹出、从影子寄存器恢复保存的内容并将 GIE 位置 1 来退出 ISR。

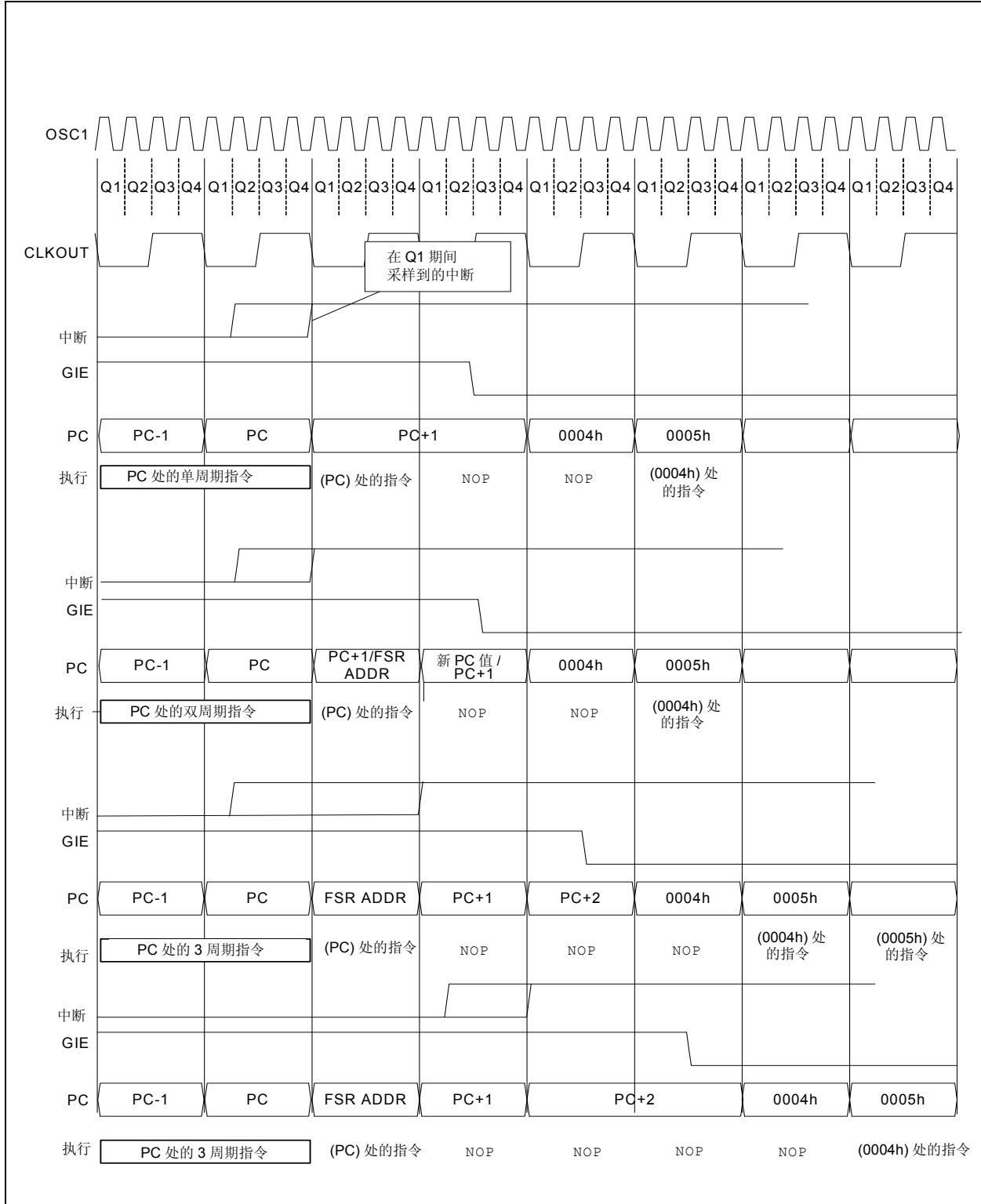
如需了解关于特殊中断操作的其他信息，请参见其相应的外设章节。

- |   |
|---|
| <p><b>注 1:</b> 无论任何其他中断允许位的状态如何，各个中断标志位都会在中断发生时置 1。</p> <p><b>2:</b> 当 GIE 位清零时，忽略所有中断。GIE 位清零时发生的任何中断在 GIE 位再次置 1 时得到处理。</p> |
|---|

## 7.2 中断响应延时

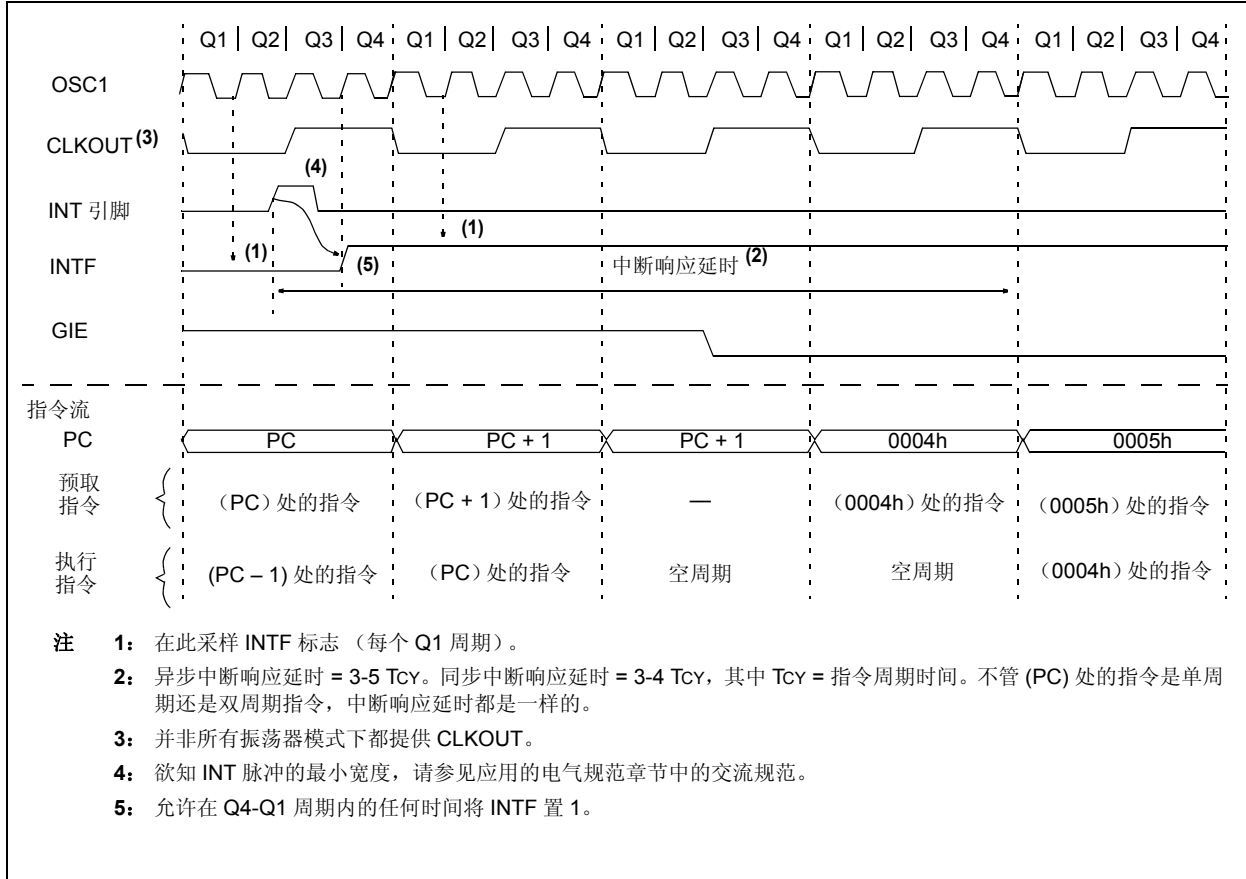
中断响应延时定义为从发生中断事件到开始执行中断向量处的代码所经过的时间。同步中断的响应延时为 3 或 4 个指令周期。对于异步中断，响应延时为 3 至 5 个指令周期，具体取决于中断发生的时间。更多信息，请参见图 7-2 和图 7-3。

图 7-2: 中断响应延时



# PIC16(L)F1933

图 7-3: INT 引脚中断时序



## 7.3 休眠期间的中断

有些中断可用于将器件从休眠模式唤醒。要使器件从休眠模式唤醒，相应外设必须能够在无系统时钟的条件下工作。进入休眠模式之前，必须将相应中断源的中断允许位置 1。

从休眠模式唤醒时，如果 GIE 位也被置 1，则处理器跳转到中断向量。否则，处理器将继续执行 SLEEP 指令后的指令。在跳转到 ISR 之前，始终会执行紧接着 SLEEP 指令的指令。更多详细信息，请参见第 9.0 节“掉电模式（休眠）”。

## 7.4 INT 引脚

INT 引脚用于产生异步边沿触发中断。将 INTCON 寄存器的 INTE 位置 1 可允许这种中断。OPTION\_REG 寄存器的 INTEDG 位可确定中断发生在哪个边沿。当 INTEDG 位置 1 时，上升沿触发中断。当 INTEDG 位清零时，下降沿触发中断。当 INT 引脚上出现有效边沿时，INTCON 寄存器的 INTF 位置 1。如果 GIE 和 INTE 位也置 1，则处理器将程序执行重定向到中断向量。

## 7.5 自动现场保护

进入中断时，将返回的 PC 地址保存在堆栈中。而且，以下寄存器自动保存在影子寄存器中：

- W 寄存器
- STATUS 寄存器（TO 和 PD 状态标志位除外）
- BSR 寄存器
- FSR 寄存器
- PCLATH 寄存器

退出中断服务程序时，这些寄存器将自动恢复。在 ISR 期间对这些寄存器所做的任何修改都将丢失。如果需要修改任何这些寄存器，应修改相应的影子寄存器，并在退出 ISR 时恢复此新值。影子寄存器在 Bank 31 中，可读写。根据用户应用程序的要求，可能还需要保存其他寄存器。

# PIC16(L)F1933

## 7.6 中断控制寄存器

### 7.6.1 INTCON 寄存器

INTCON 寄存器是可读写的寄存器，包含 TMR0 寄存器溢出中断、电平变化中断和外部 INT 引脚中断的各个允许和标志位。

**注：** 当有中断条件产生时，不管相应的中断允许位或全局中断允许位（INTCON 寄存器的 GIE）的状态如何，中断标志位都将置 1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 7-1: INTCON: 中断控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0
GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCF
bit 7							bit 0

**图注：**

R = 可读位                      W = 可写位                      U = 未实现位，读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                         0 = 清零

- bit 7        **GIE:** 全局中断允许位  
            1 = 允许所有有效中断  
            0 = 禁止所有中断
- bit 6        **PEIE:** 外设中断允许位  
            1 = 允许所有有效外设中断  
            0 = 禁止所有外设中断
- bit 5        **TMR0IE:** Timer0 溢出中断允许位  
            1 = 允许 Timer0 中断  
            0 = 禁止 Timer0 中断
- bit 4        **INTE:** INT 外部中断允许位  
            1 = 允许 INT 外部中断  
            0 = 禁止 INT 外部中断
- bit 3        **IOCFIE:** 电平变化中断允许位  
            1 = 允许电平变化中断  
            0 = 禁止电平变化中断
- bit 2        **TMR0IF:** Timer0 溢出中断标志位  
            1 = TMR0 寄存器已溢出  
            0 = TMR0 寄存器未溢出
- bit 1        **INTF:** INT 外部中断标志位  
            1 = 发生了 INT 外部中断  
            0 = 未发生 INT 外部中断
- bit 0        **IOCF:** 电平变化中断的中断标志位  
            1 = 至少一个电平变化中断引脚的状态发生改变  
            0 = 没有一个电平变化中断引脚的状态发生改变

**注 1:** IOCF 标志位为只读，将在 IOCBF 寄存器中的所有电平变化中断标志都由软件清零后，清零该位。

## 7.6.2 PIE1 寄存器

PIE1 寄存器包含中断允许位，如寄存器 7-2 所示。

**注：** 要允许任何一个外设中断，必须将 INTCON 寄存器的 PEIE 位置 1。

**寄存器 7-2: PIE1: 外设中断允许寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7							bit 0

**图注：**

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7      **TMR1GIE:** Timer1 门控中断允许位  
1 = 允许 Timer1 门控采集中断  
0 = 禁止 Timer1 门控采集中断
- bit 6      **ADIE:** A/D 转换器 (ADC) 中断允许位  
1 = 允许 ADC 中断  
0 = 禁止 ADC 中断
- bit 5      **RCIE:** USART 接收中断允许位  
1 = 允许 USART 接收中断  
0 = 禁止 USART 接收中断
- bit 4      **TXIE:** USART 发送中断允许位  
1 = 允许 USART 发送中断  
0 = 禁止 USART 发送中断
- bit 3      **SSPIE:** 同步串行端口 (MSSP) 中断允许位  
1 = 允许 MSSP 中断  
0 = 禁止 MSSP 中断
- bit 2      **CCP1IE:** CCP1 中断允许位  
1 = 允许 CCP1 中断  
0 = 禁止 CCP1 中断
- bit 1      **TMR2IE:** TMR2 与 PR2 匹配中断允许位  
1 = 允许 Timer2 与 PR2 匹配中断  
0 = 禁止 Timer2 与 PR2 匹配中断
- bit 0      **TMR1IE:** Timer1 上溢中断允许位  
1 = 允许 Timer1 上溢中断  
0 = 禁止 Timer1 上溢中断

# PIC16(L)F1933

## 7.6.3 PIE2 寄存器

PIE2 寄存器包含中断允许位，如寄存器 7-3 所示。

**注：** 要允许任何一个外设中断，必须将 INTCON 寄存器的 PEIE 位置 1。

**寄存器 7-3: PIE2: 外设中断允许寄存器 2**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE
bit 7							bit 0

**图注：**

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7      **OSFIE:** 振荡器故障中断允许位  
1 = 允许振荡器故障中断  
0 = 禁止振荡器故障中断
- bit 6      **C2IE:** 比较器 C2 中断允许位  
1 = 允许比较器 C2 中断  
0 = 禁止比较器 C2 中断
- bit 5      **C1IE:** 比较器 C1 中断允许位  
1 = 允许比较器 C1 中断  
0 = 禁止比较器 C1 中断
- bit 4      **EEIE:** EEPROM 写完成中断允许位  
1 = 允许 EEPROM 写完成中断  
0 = 禁止 EEPROM 写完成中断
- bit 3      **BCLIE:** MSSP 总线冲突中断允许位  
1 = 允许 MSSP 总线冲突中断  
0 = 禁止 MSSP 总线冲突中断
- bit 2      **LCDIE:** LCD 模块中断允许位  
1 = 允许 LCD 模块中断  
0 = 禁止 LCD 模块中断
- bit 1      **未实现:** 读为 0
- bit 0      **CCP2IE:** CCP2 中断允许位  
1 = 允许 CCP2 中断  
0 = 禁止 CCP2 中断



## 7.6.4 PIE3 寄存器

PIE3 寄存器包含中断允许位，如寄存器 7-4 所示。

**注：** 要允许任何一个外设中断，必须将 INTCON 寄存器的 PEIE 位置 1。

**寄存器 7-4: PIE3: 外设中断允许寄存器 3**

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	U-0
—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—
bit 7							bit 0

**图注：**

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7      **未实现：** 读为 0
- bit 6      **CCP5IE:** CCP5 中断允许位  
1 = 允许 CCP5 中断  
0 = 禁止 CCP5 中断
- bit 5      **CCP4IE:** CCP4 中断允许位  
1 = 允许 CCP4 中断  
0 = 禁止 CCP4 中断
- bit 4      **CCP3IE:** CCP3 中断允许位  
1 = 允许 CCP3 中断  
0 = 禁止 CCP3 中断
- bit 3      **TMR6IE:** TMR6 与 PR6 匹配中断允许位  
1 = 允许 TMR6 与 PR6 匹配中断  
0 = 禁止 TMR6 与 PR6 匹配中断
- bit 2      **未实现：** 读为 0
- bit 1      **TMR4IE:** TMR4 与 PR4 匹配中断允许位  
1 = 允许 TMR4 与 PR4 匹配中断  
0 = 禁止 TMR4 与 PR4 匹配中断
- bit 0      **未实现：** 读为 0

# PIC16(L)F1933

## 7.6.5 PIR1 寄存器

PIR1 寄存器包含中断标志位，如寄存器 7-5 所示。

**注：** 当有中断条件产生时，不管相应的中断允许位或全局中断允许位（INTCON 寄存器的 GIE）的状态如何，中断标志位都将置 1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 7-5: PIR1: 外设中断请求寄存器 1

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7							bit 0

**图注：**

R = 可读位                      W = 可写位                      U = 未实现位，读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                         0 = 清零

- bit 7            **TMR1GIF:** Timer1 门控中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 6            **ADIF:** A/D 转换器中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 5            **RCIF:** USART 接收中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 4            **TXIF:** USART 发送中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 3            **SSPIF:** 同步串行端口（MSSP）中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 2            **CCP1IF:** CCP1 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 1            **TMR2IF:** Timer2 与 PR2 匹配中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 0            **TMR1IF:** Timer1 上溢中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断

## 7.6.6 PIR2 寄存器

PIR2 寄存器包含中断标志位，如寄存器 7-6 所示。

**注：** 当有中断条件产生时，不管相应的中断允许位或全局中断允许位（INTCON 寄存器的 GIE）的状态如何，中断标志位都将置 1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

**寄存器 7-6: PIR2: 外设中断请求寄存器 2**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF
bit 7						bit 0	

**图注：**

R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **OSFIF:** 振荡器故障中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 6      **C2IF:** 比较器 C2 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 5      **C1IF:** 比较器 C1 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 4      **EEIF:** EEPROM 写完成中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 3      **BCLIF:** MSSP 总线冲突中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 2      **LCDIF:** LCD 模块中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 1      **未实现:** 读为 0
- bit 0      **CCP2IF:** CCP2 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断

# PIC16(L)F1933

## 7.6.7 PIR3 寄存器

PIR3 寄存器包含中断标志位，如寄存器 7-7 所示。

**注：** 当有中断条件产生时，不管相应的中断允许位或全局中断允许位（INTCON 寄存器的 GIE）的状态如何，中断标志位都将置 1。用户软件应该在允许中断之前确保将相应的中断标志位清零。

寄存器 7-7: PIR3: 外设中断请求寄存器 3

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—
bit 7							bit 0

**图注：**

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7      **未实现：** 读为 0
- bit 6      **CCP5IF:** CCP5 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 5      **CCP4IF:** CCP4 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 4      **CCP3IF:** CCP3 中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 3      **TMR6IF:** TMR6 与 PR6 匹配中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 2      **未实现：** 读为 0
- bit 1      **TMR4IF:** TMR4 与 PR4 匹配中断标志位  
1 = 中断等待响应  
0 = 没有等待响应的中断
- bit 0      **未实现：** 读为 0

表 7-1: 与中断相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFE	TMR0IF	INTF	IOCF	86
OPTION_REG	WPUEN	INTEDG	TMROCS	TMROSE	PSA	PS<2:0>			175
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92

图注: — = 未实现单元, 读为 0。中断不使用阴影单元。

# PIC16(L)F1933

---

注:

## 8.0 低压差 (LDO) 稳压器

PIC16F1933 具有内部低压差 (Low Dropout, LDO) 稳压器, 它可以提供 3.6V 以上的工作电压。LDO 可为内部器件逻辑提供稳压, 同时允许 VDD 和 I/O 引脚工作在较高的电压下。用户不能使能 / 禁止对 LDO 的控制, 它始终有效。PIC16LF193X 可以工作在最高 3.6V 的 VDD 下, 且不使用 LDO。

器件 I/O 引脚可以配置为 LDO 电压输出, 标记为 VCAP 引脚。尽管不需要, 但是外部低 ESR 电容可以连接到 VCAP 引脚, 以提供额外的稳压器稳定性。

配置字寄存器 2 的 VCAPEN<1:0> 位可以决定哪个引脚被指定为 VCAP 引脚。请参见表 8-1。

上电时, 外部电容将加载 LDO 稳压器。为防止错误操作, 恒流源对外部电容充电时, 器件保持在复位状态。电容完全充电后, 器件从复位状态释放。关于推荐电容值和恒流速率的更多信息, 请参见应用的电气规范章节中的 LDO 稳压器特性表。

表 8-1: VCAPEN<1:0> 选择位

VCAPEN<1:0>	引脚
00	RA0
01	RA5
10	RA6
11	无 Vcap

表 8-2: 与 LDO 相关的配置字寄存器汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在的页面
CONFIG2	13:8	—	—	LVP	DEBUG	—	BORV	STVREN	PLLEN	52
	7:0	—	—	VCAPEN1 <sup>(1)</sup>	VCAPEN0 <sup>(1)</sup>	—	—	WRT1	WRT0	

图注: — = 未实现单元, 读为 0。LDO 不使用阴影单元。

注 1: 仅适用于 PIC16F1933 器件。

# PIC16(L)F1933

---

注:



## 9.0 掉电模式（休眠）

器件通过执行 SLEEP 指令进入掉电模式。

进入休眠模式时，存在以下条件：

1. WDT 将清零但是保持运行（如果使能了在休眠期间工作）。
2. STATUS 寄存器的  $\overline{PD}$  位清零。
3. STATUS 寄存器的  $\overline{TO}$  位置 1。
4. 禁止 CPU 时钟。
5. 31 kHz LFINTOSC 不受影响，并且由 31 kHz LFINTOSC 提供时钟的外设可以在休眠模式下继续工作。
6. Timer1 振荡器不受影响，并且由 Timer1 振荡器提供时钟的外设可以在休眠模式下继续工作。
7. ADC 不受影响（如果选择了专用 FRC 时钟）。
8. 电容触摸传感振荡器不受影响。
9. I/O 端口保持执行 SLEEP 指令之前的状态（驱动为高电平、低电平或高阻态）。
10. WDT 之外的复位不受休眠模式影响。

关于外设是在休眠期间工作的更多详细信息，请参见各个章节。

要最大程度地降低电流消耗，应考虑以下条件：

- I/O 引脚不应悬空
- 外部电路从 I/O 引脚的灌电流
- 内部电路从 I/O 引脚的拉电流
- 电流汲取自带有内部弱上拉的引脚
- 模块使用 31 kHz LFINTOSC
- 模块使用 Timer1 振荡器

为了避免输入引脚悬空而引入开关电流，应在外部将高阻输入的 I/O 引脚拉为 VDD 或 VSS。

可以拉电流的内部电路示例包括 DAC 和 FVR 模块之类的模块。关于这些模块的更多信息，请参见第 17.0 节“数模转换器（DAC）模块”和第 14.0 节“固定参考电压（FVR）”。

## 9.1 从休眠模式唤醒

可以通过下列任一事件将器件从休眠状态唤醒：

1.  $\overline{MCLR}$  引脚上的外部复位输入（如果使能）
2. BOR 复位（如果使能）
3. POR 复位
4. 看门狗定时器（如果使能）
5. 任何外部中断
6. 能够在休眠期间运行的外设产生的中断（更多信息请参见各个外设）。

前 3 个事件会使器件复位。后 3 个事件认为是程序执行的延续。要确定是发生器件复位还是唤醒事件，请参见第 6.10 节“确定复位原因”。

当执行 SLEEP 指令时，下一条指令（PC + 1）被预先取出。如果希望通过中断事件唤醒器件，则必须允许相应的中断允许位。唤醒与 GIE 位的状态无关。如果 GIE 位被禁止，器件将继续执行 SLEEP 指令后的指令。如果 GIE 位被允许，器件先执行 SLEEP 指令后的指令，然后将调用中断服务程序。如果不想执行 SLEEP 指令后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

器件从休眠模式唤醒时，WDT 清零，与唤醒的原因无关。

# PIC16(L)F1933

## 9.1.1 使用中斷喚醒

当禁止全局中断（GIE 被清零）并且有任一中断源将其中断允许位和中断标志位置 1 时，将会发生下列某一事件：

- 如果在执行 SLEEP 指令之前发生中断
  - SLEEP 指令将作为 NOP 执行
  - WDT 和 WDT 预分频器不会清零
  - STATUS 寄存器的  $\overline{TO}$  位不会置 1
  - STATUS 寄存器的  $\overline{PD}$  位不会清零

- 如果在执行 SLEEP 指令期间或之后发生中断
  - SLEEP 指令将完全执行
  - 器件将立即从休眠模式唤醒
  - WDT 和 WDT 预分频器将清零
  - STATUS 寄存器的  $\overline{TO}$  位将置 1
  - STATUS 寄存器的  $\overline{PD}$  位将清零

即使在执行 SLEEP 指令之前，检查到标志位为 0，它也可能在 SLEEP 指令执行完毕之前被置 1。要确定是否执行了 SLEEP 指令，可以测试  $\overline{PD}$  位。如果  $\overline{PD}$  位置 1，则说明 SLEEP 指令被作为一条 NOP 指令执行了。

图 9-1: 通过中断将器件从休眠模式唤醒

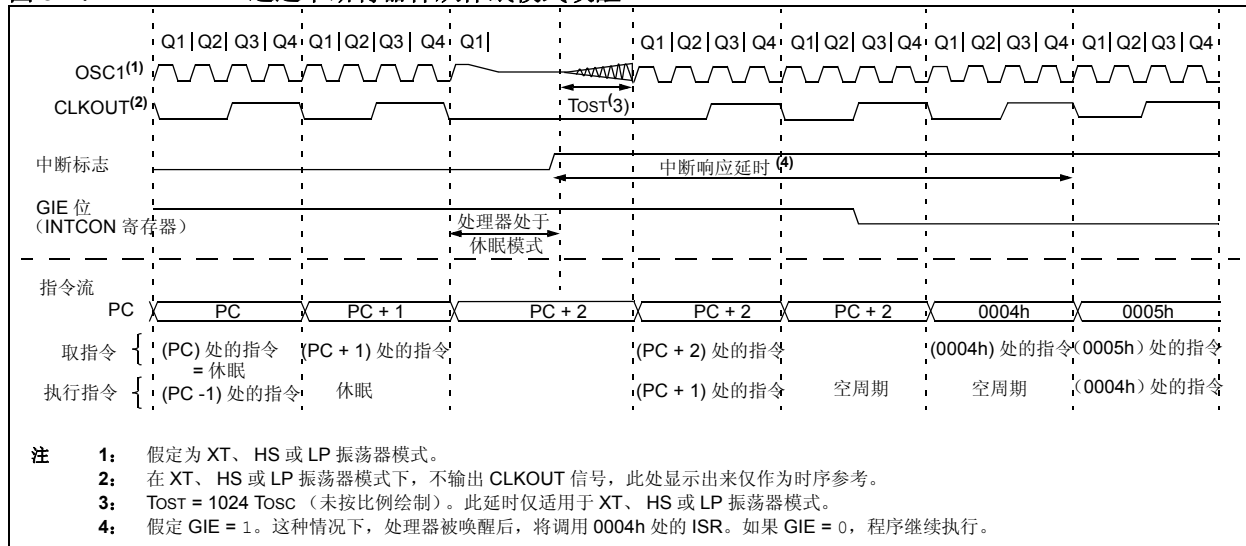


表 9-1: 与掉电模式相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	86
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	134
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	134
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	134
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92
STATUS	—	—	—	$\overline{TO}$	$\overline{PD}$	Z	DC	C	22
WDTCON	—	—	WDTPS<4:0>					SWDTEN	101

图注：— = 未实现单元，读为 0。掉电模式不使用阴影单元。

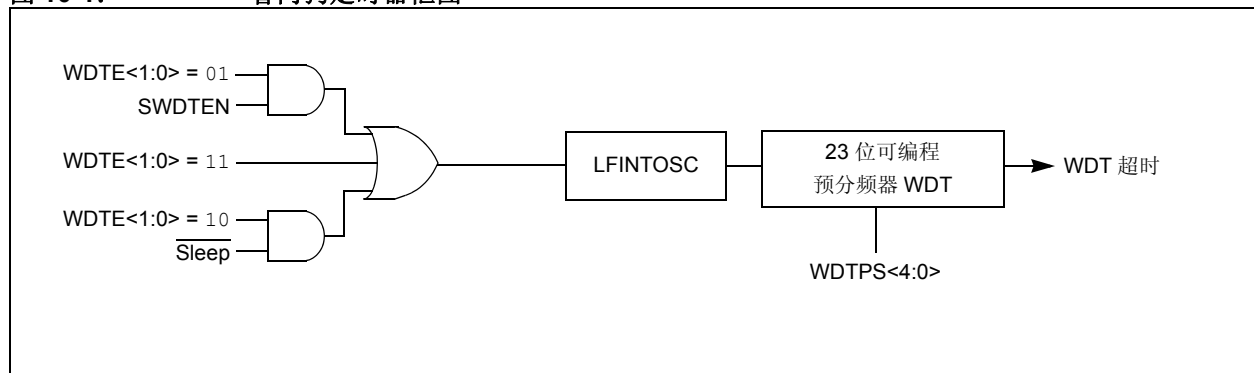
## 10.0 看门狗定时器

看门狗定时器是系统定时器，如果固件在超时周期内未发出 CLRWDT 指令，看门狗定时器将产生复位。看门狗定时器通常用于将系统从意外事件中恢复。

WDT 具有如下特性：

- 独立的时钟源
- 多种工作模式
  - WDT 始终使能
  - WDT 在休眠时禁止
  - WDT 由软件控制
  - WDT 始终禁止
- 可配置的典型超时周期为 1 ms 至 256 秒
- 多个复位条件
- 休眠期间的工作原理

图 10-1: 看门狗定时器框图



# PIC16(L)F1933

## 10.1 独立时钟源

WDT 以 31 kHz LFINTOSC 的内部振荡器作为其时基。本章中的时间间隔基于 1 ms 的标称时间间隔。LFINTOSC 公差，请参见电气规范章节。

## 10.2 WDT 工作模式

看门狗定时器模块有 4 种工作模式，由配置字寄存器 1 的 WDTE<1:0> 位控制。请参见表 10-1。

### 10.2.1 WDT 始终使能

配置字寄存器 1 的 WDTE 位设置为 11 时，WDT 始终使能。

WDT 保护功能在休眠期间有效。

### 10.2.2 WDT 在休眠时禁止

配置字寄存器 1 的 WDTE 位设置为 10 时，WDT 除了在休眠时之外，一直使能。

WDT 保护功能在休眠期间无效。

### 10.2.3 WDT 由软件控制

配置字寄存器 1 的 WDTE 位设置为 01 时，WDT 由 WDTCON 寄存器的 SWDTEN 位控制。

WDT 保护功能不受休眠影响。更多详细信息，请参见表 10-1。

表 10-1: WDT 工作模式

WDTE<1:0>	SWDTEN	器件模式	WDT 模式
11	X	X	有效
10	X	唤醒	有效
		休眠	禁止
01	1	X	有效
	0		禁止
00	X	X	禁止

表 10-2: WDT 清零条件

条件	WDT
WDTE<1:0> = 00	清零
WDTE<1:0> = 01 且 SWDTEN = 0	
WDTE<1:0> = 10 且进入休眠模式	
CLRWDT 命令	
检测到振荡器故障	
退出休眠 + 系统时钟 = T1OSC、EXTRC、INTOSC 或 EXTCLK	
退出休眠 + 系统时钟 = XT、HS 或 LP	清零直到 OST 延时结束
更改 INTOSC 分频比 (IRCF 位)	不受影响

## 10.3 超时周期

WDTCON 寄存器的 WDTPS 位可以设置从 1 ms 至 256 秒（标称值）的超时周期。复位后，默认超时周期为 2 秒。

## 10.4 清零 WDT

发生以下任何条件时，WDT 清零：

- 任何复位
- 执行 CLRWDT 指令
- 器件进入休眠模式
- 器件从休眠模式唤醒
- 振荡器故障事件
- WDT 被禁止
- 振荡器起振定时器 (OST) 正在运行

更多信息，请参见表 10-2。

## 10.5 休眠期间的工作原理

器件进入休眠模式时，WDT 清零。如果休眠期间使能了 WDT，WDT 将恢复计数。

器件退出休眠模式时，WDT 再次清零。WDT 保持清零，直到 OST（如果使能）结束。关于 OST 的更多信息，请参见第 5.0 节“振荡器模块（带故障保护时钟监视器）”。

如果在器件处于休眠模式时发生 WDT 超时，不会产生复位。而是，将器件唤醒并继续工作。更改 STATUS 寄存器中的 TO 和 PD 位以指示此事件。更多信息，请参见第 3.0 节“存储器构成”和 STATUS 寄存器（寄存器 3-1）。

## 10.6 看门狗控制寄存器

寄存器 10-1: **WDTCON: 看门狗定时器控制寄存器**

U-0	U-0	R/W-0/0	R/W-1/1	R/W-0/0	R/W-1/1	R/W-1/1	R/W-0/0
—	—	WDTPS<4:0>					SWDTEN
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-m/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6      **未实现:** 读为 0

bit 5-1      **WDTPS<4:0>:** 看门狗定时器周期选择位

位值 = 预分频比

- 00000 = 1:32 (典型间隔 1 ms)
- 00001 = 1:64 (典型间隔 2 ms)
- 00010 = 1:128 (典型间隔 4 ms)
- 00011 = 1:256 (典型间隔 8 ms)
- 00100 = 1:512 (典型间隔 16 ms)
- 00101 = 1:1024 (典型间隔 32 ms)
- 00110 = 1:2048 (典型间隔 64 ms)
- 00111 = 1:4096 (典型间隔 128 ms)
- 01000 = 1:8192 (典型间隔 256 ms)
- 01001 = 1:16384 (典型间隔 512 ms)
- 01010 = 1:32768 (典型间隔 1s)
- 01011 = 1:65536 (典型间隔 2s) (复位值)
- 01100 = 1:131072 ( $2^{17}$ ) (典型间隔 4s)
- 01101 = 1:262144 ( $2^{18}$ ) (典型间隔 8s)
- 01110 = 1:524288 ( $2^{19}$ ) (典型间隔 16s)
- 01111 = 1:1048576 ( $2^{20}$ ) (典型间隔 32s)
- 10000 = 1:2097152 ( $2^{21}$ ) (典型间隔 64s)
- 10001 = 1:4194304 ( $2^{22}$ ) (典型间隔 128s)
- 10010 = 1:8388608 ( $2^{23}$ ) (典型间隔 256s)

10011 = 保留。得到最小间隔 (1:32)

·  
·  
·

11111 = 保留。得到最小间隔 (1:32)

bit 0      **SWDTEN:** 看门狗定时器的软件使能 / 禁止位

如果 WDTE<1:0> = 00:

忽略此位。

如果 WDTE<1:0> = 01:

1 = 使能 WDT

0 = 禁止 WDT

如果 WDTE<1:0> = 1X:

忽略此位。

# PIC16(L)F1933

**表 10-3: 与看门狗定时器相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
OSCCON	SPLLEN	IRCF<3:0>				—	SCS<1:0>		69
STATUS	—	—	—	TO	PD	Z	DC	C	22
WDTCON	—	—	WDTPS<4:0>				—	SWDTEN	101

图注: x = 未知, u = 不变, — = 未实现单元, 读为 0。看门狗定时器不使用阴影单元。

**表 10-4: 与看门狗定时器的配置字寄存器汇总**

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在的页面
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>	—	CPD	50
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>	—	FOSC<2:0>	—	—	

图注: — = 未实现单元, 读为 0。看门狗定时器不使用阴影单元。

## 11.0 数据 EEPROM 和闪存程序存储器控制

数据 EEPROM 和闪存程序存储器在正常工作期间（整个 V<sub>DD</sub> 范围）是可读写的。这两种存储器没有直接映射到文件寄存器空间，而是通过特殊功能寄存器（SFR）间接寻址。有 6 个 SFR 用于访问这两种存储器：

- EECON1
- EECON2
- EEDATL
- EEDATH
- EEADRL
- EEADRH

当与数据存储器模块接口时，EEDATL 寄存器存放要读写的 8 位数据，EEADRL 寄存器存放被访问的 EEDATL 单元的地址。这些器件具有 256 字节的数据 EEPROM，地址范围从 0h 到 0FFh。

访问程序存储器模块时，EEDATH:EEDATL 寄存器对形成双字节字，存放要读 / 写的 14 位数据，而 EEADRL 和 EEADRH 寄存器形成双字节字，存放被读取的程序存储单元的 15 位地址。

EEPROM 数据存储器允许以字节为单位进行读写。EEPROM 字节写操作会自动擦除目标存储单元并写入新数据（在写入前擦除）。

写入时间由片上定时器控制。写入和擦除电压由片上电荷泵产生，此电荷泵能在器件电压范围内正常工作，用于字节或字操作。

器件能否对程序存储器的特定块执行写操作取决于配置字寄存器 2 的闪存程序存储器自写使能位 WRT<1:0> 的设置。然而，始终允许读程序存储器。

当器件被代码保护时，器件编程器将不再能访问数据或程序存储器。在代码保护时，CPU 仍可继续读写数据 EEPROM 存储器和闪存程序存储器。

## 11.1 EEADRL 和 EEADRH 寄存器

EEADRH:EEADRL 寄存器对可以寻址最大 256 字节的数据 EEPROM 或最大 32K 字的程序存储器。

当选择程序地址值时，地址的高字节写入 EEADRH 寄存器而低字节写入 EEADRL 寄存器。当选择 EEPROM 地址值时，只将地址的低字节写入 EEADRL 寄存器。

### 11.1.1 EECON1 和 EECON2 寄存器

EECON1 是访问 EE 存储器的控制寄存器。

控制位 EEPGD 决定访问的是程序存储器还是数据存储器。当清零时，任何后续操作都将针对 EEPROM 存储器进行。当置 1 时，任何后续操作都将针对程序存储器进行。复位后，默认选中 EEPROM。

控制位 RD 和 WR 分别启动读和写操作。用软件只能将这些位置 1 而无法清零。在读或写操作完成后，由硬件将它们清零。由于无法用软件将 WR 位清零，可避免因意外而过早地终止写操作。

当 WREN 位置 1 时，允许执行写操作。上电时，WREN 位被清零。在正常运行中当写操作被复位中断时，WRERR 位置 1。在这些情况下，复位后用户可以检查 WRERR 位并执行相应的错误处理程序。

当写操作完成时，PIR2 寄存器的中断标志位 EEIF 被置 1。该标志位必须用软件清零。

读 EECON2 得到的是全 0。EECON2 寄存器仅在数据 EEPROM 写过程中使用。要使能写操作，必须将特定模式写入 EECON2。

## 11.2 使用数据 EEPROM

数据 EEPROM 是高耐久性、可字节寻址的阵列，已将其优化以便存储频繁更改的信息（例如：程序变量或其他经常更新的数据）。当一个段中的变量频繁更改，而另一个段中的变量不发生变化时，就可能造成超过允许对 EEPROM 进行写入的总次数，而不是超出允许对某个字节进行写入的总次数。请参见应用的电气规范章节。如果是这种情况，那么必须执行阵列刷新。为此，应将不频繁更改的变量（例如常量、ID 和校准值等）存储在闪存程序存储器中。

### 11.2.1 读数据 EEPROM 存储器

要读取数据存储单元，用户必须把地址写入 EEDRL 寄存器，清零 EECON1 寄存器的 EEPGD 和 CFGS 控制位，然后置 1 控制位 RD。在紧接着的下一个周期，EEDATL 寄存器中就有了数据；因此，该数据可由下一条指令读取。EEDATL 将把此值保持至下一次读取或用户向该单元写入数据时（在写操作过程中）为止。

#### 例 11-1: 读数据 EEPROM

```
BANKSEL EEDRL      ;
MOVLW  DATA_EE_ADDR ;
MOVWF  EEDRL       ;Data Memory
                          ;Address to read
BCF    EECON1, CFGS ;Deselect Config space
BCF    EECON1, EEPGD;Point to DATA memory
BSF    EECON1, RD   ;EE Read
MOVF   EEDATL, W   ;W = EEDATL
```

**注：** 无论  $\overline{\text{CPD}}$  位的设置为何，都可以读取数据 EEPROM。

### 11.2.2 写数据 EEPROM 存储器

要写 EEPROM 数据存储单元，用户应首先将该单元的地址写入 EEADRL 寄存器并将数据写入 EEDATL 寄存器。然后用户必须按特定顺序开始写入每个字节。

如果未完全按照上述顺序（即，首先将 55h 写入 EECON2，随后将 AAh 写入 EECON2，最后将 WR 位置 1）逐字节写入，将不会启动写操作。在该代码段中应禁止中断。

此外，必须将 EECON1 中的 WREN 位置 1 以启用写操作。这种机制可防止由于代码执行错误（异常）（即程序失控）导致误写数据 EEPROM。除了更新 EEPROM 时以外，用户应始终保持 WREN 位清零。WREN 位不能由硬件清零。

一个写序列启动后，清零 WREN 位将不会影响此写周期。除非 WREN 位置 1，否则 WR 位将无法置 1。

写周期完成时，WR 位由硬件清零并且 EE 写完成中断标志位（EEIF）置 1。用户可以允许中断或查询此位。EEIF 必须用软件清零。

### 11.2.3 防止误写操作的保护措施

有些情况下，用户并不希望向数据 EEPROM 存储器写入数据。为了防止 EEPROM 误写操作，器件内建了各种保护机制。上电时，清零 WREN。同时，上电延时定时器（64 ms 的延时）也会阻止对 EEPROM 进行写操作。

写启动序列和 WREN 位共同防止在以下情况下发生意外写操作：

- 欠压
- 电源故障
- 软件故障

### 11.2.4 代码保护期间的数据 EEPROM 操作

将配置字寄存器 1（寄存器 5-1）中的  $\overline{\text{CPD}}$  位设置为 0，可以对数据存储器进行代码保护。

在数据存储器受代码保护时，只有 CPU 可对数据 EEPROM 进行读写操作。当数据存储器受代码保护时，建议对程序存储器也进行代码保护。这样便可阻止任何人使用访问数据 EEPROM 内容的程序来替代您的程序。



**例 11-2: 写数据 EEPROM**

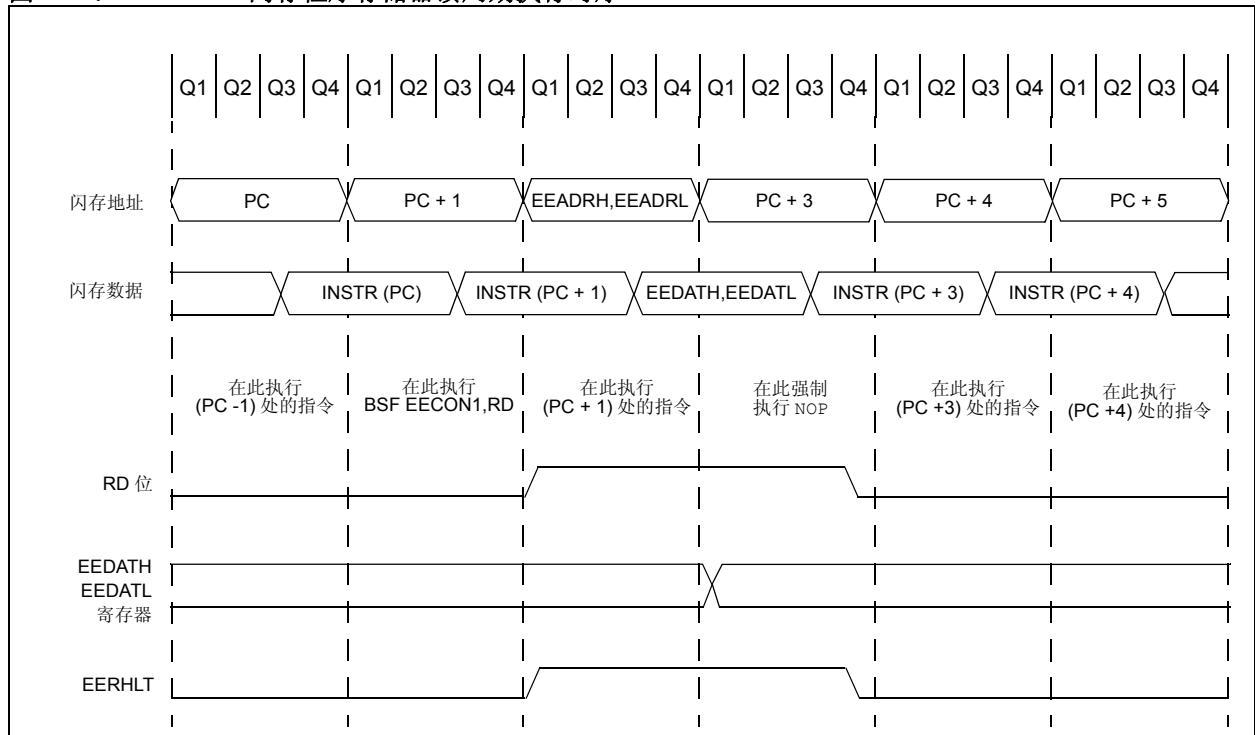
```

BANKSEL  EEADRL      ;
MOVLW   DATA_EE_ADDR ;
MOVWF   EEADRL      ;Data Memory Address to write
MOVLW   DATA_EE_DATA ;
MOVWF   EEDATL      ;Data Memory Value to write
BCF     EECON1, CFGS ;Deselect Configuration space
BCF     EECON1, EEPGD ;Point to DATA memory
BSF     EECON1, WREN ;Enable writes

BCF     INTCON, GIE  ;Disable INTs.
MOVLW   55h         ;
MOVWF   EECON2      ;Write 55h
MOVLW   0AAh        ;
MOVWF   EECON2      ;Write AAh
BSF     EECON1, WR   ;Set WR bit to begin write
BSF     INTCON, GIE  ;Enable Interrupts
BCF     EECON1, WREN ;Disable writes
BTFSC   EECON1, WR   ;Wait for write to complete
GOTO    $-2         ;Done
    
```

必需的  
序列

**图 11-1: 闪存程序存储器读周期执行时序**



# PIC16(L)F1933

## 11.3 闪存程序存储器概述

了解闪存程序存储器结构对于擦除和编程操作非常重要。闪存程序存储器按行排列。每行包括固定数量的 14 位程序存储器字。行是用户软件可擦除的最小块大小。

只有当目标地址位于未受写保护的存储器段内（由配置字寄存器 2 的 WRT<1:0> 位定义）时，才能对闪存程序存储器进行写或擦除操作。

擦除某行后，用户可以对该行的部分或全部进行重新编程。写入程序存储器行的数据将被写入 14 位宽的数据写锁存器中。用户不能直接访问这些写锁存器，但是可以通过对 EEDATH:EEDATL 寄存器对的连续写入来加载写锁存器的内容。

**注：** 如果用户只想修改部分以前编程的行，那么必须读取整行的内容并保存在 RAM 中，然后进行擦除。

数据写锁存器数可能不等于行单元数。编程时，用户软件需要填充该组写锁存器并多次启动编程操作，才能完全重新编程擦除的行。例如，具有 32 字的行大小和 8 个写锁存器的器件需要将数据装入写锁存器并启动编程操作四次。

程序存储器行的大小和程序存储器写锁存器数可能因器件而异。详情请参见表 11-1。

表 11-1: 闪存存储器构成（按器件）

器件	擦除块（行）大小 / 边界	写锁存器数 / 边界
PIC16(L)F1933	32 字, EEADRL<4:0> = 00000	8 字, EEADRL<2:0> = 000

### 11.3.1 读闪存程序存储器

要读程序存储单元，用户必须：

1. 将最低有效地址位和最高有效地址位写入 EEADRH:EEADRL 寄存器对。
2. 将 EECON1 寄存器的 CFGS 位清零。
3. 将 EECON1 寄存器的 EEPGD 控制位置 1。
4. 然后，将 EECON1 寄存器的控制位 RD 置 1。

一旦将读控制位置 1，闪存程序存储器控制器将使用第二个指令周期读取数据。这会导致紧跟 BSF EECON1, RD 指令后的第二条指令被忽略。在紧接着的下一个周期，EEDATH:EEDATL 寄存器对中就有数据了，因此可在随后的指令中将该数据读作两个字节。

EEDATH:EEDATL 寄存器对将把此值保存至下一次读操作或用户向该单元写入数据时为止。

- 注**
- 1: 要求程序存储器读操作后的两条指令为 NOP。这可以防止用户在 RD 位置 1 后的下一条指令执行双周期指令。
  - 2: 不管  $\overline{CP}$  位的设置如何，都可以读取闪存程序存储器。

## 例 11-3: 读闪存程序存储器

```
* This code block will read 1 word of program
* memory at the memory address:
  PROG_ADDR_HI: PROG_ADDR_LO
* data will be returned in the variables;
* PROG_DATA_HI, PROG_DATA_LO

BANKSEL  EEADRL          ; Select Bank for EEPROM registers
MOVLW    PROG_ADDR_LO   ;
MOVWF    EEADRL         ; Store LSB of address
MOVLW    PROG_ADDR_HI   ;
MOVWL    EEADRH         ; Store MSB of address

BCF      EECON1,CFGSR    ; Do not select Configuration Space
BSF      EECON1,EEPGD    ; Select Program Memory
BCF      INTCON,GIE      ; Disable interrupts
BSF      EECON1,RD       ; Initiate read
NOP      ; Executed (Figure 11-1)
NOP      ; Ignored (Figure 11-1)
BSF      INTCON,GIE      ; Restore interrupts

MOVWF    EEDATL,W       ; Get LSB of word
MOVWF    PROG_DATA_LO   ; Store in user location
MOVWF    EEDATH,W       ; Get MSB of word
MOVWF    PROG_DATA_HI   ; Store in user location
```

## 11.3.2 擦除闪存程序存储器

当执行代码时，程序存储器只能按行擦除。要擦除行：

1. 将要擦除的新行地址装入 EEADRH:EEADRL 寄存器对。
2. 将 EECON1 寄存器的 CFGS 位清零。
3. 将 EECON1 寄存器的 EEPGD、FREE 和 WREN 位置 1。
4. 依次将 55h 和 AAh 写入 EECON2（闪存编程解锁序列）。
5. 将 EECON1 寄存器的控制位 WR 置 1，以开始擦除操作。
6. 查询 EECON1 寄存器的 FREE 位，以确定行擦除何时结束。

请参见例 11-4。

在 BSF EECON1,WR 指令后，处理器需要两个周期来设置擦除操作。用户必须在 WR 位置 1 后执行两条 NOP 指令。处理器将暂停内部操作，通常为 2 ms 擦除时间。这不是休眠模式，因为时钟和外设将继续运行。擦除周期后，处理器从 EECON1 写指令后的第三条指令继续操作。

## 11.3.3 写闪存程序存储器

使用以下步骤编程程序存储器：

1. 装入要编程的字的起始地址。
2. 将数据装入写锁存器。
3. 启动编程操作。
4. 重复第 1 至 3 步，直到写入所有数据。

写入程序存储器之前，要写入的字必须已擦除或者以前未写入过。程序存储器一次只能擦除一行。启动写操作时不会自动擦除。

程序存储器可以一次写入一个或多个字。一次写入的最大字数等于写锁存器数。更多详细信息，请参见图 11-2（对带 8 个写锁存器的程序存储器的块写操作）。写锁存器与 EEADRL 定义的地址边界对齐，如表 11-1 所示。写操作不会跨这些边界。程序存储器写操作完成时，写锁存器将复位为包含 0x3FFF。

应完成以下步骤，以装载写锁存器和编程程序存储块。这些步骤可分为两个部分。首先，除最后一个程序存储单元外的所有写锁存器装入数据。然后，装载最后一个写锁存器，启动编程序列。需要特殊的解锁序列才能将数据装入写锁存器或启动闪存编程操作。不应中断此解锁序列。

1. 将 EECON1 寄存器的 EEPGD 和 WREN 位置 1。
2. 将 EECON1 寄存器的 CFGS 位清零。
3. 将 EECON1 寄存器的 LWLO 位置 1。当 EECON1 寄存器的 LWLO 位为 1 时，写序列只能装入写锁存器，不会启动对闪存程序存储器的写操作。
4. 将要写入的单元地址装入 EEADRH:EEADRL 寄存器对。
5. 将要写入的程序存储器数据装入 EEDATH:EEDATL 寄存器对。
6. 依次将 55h 和 AAh 写入 EECON2，然后将 EECON1 寄存器的 WR 位置 1（闪存编程解锁序列）。现在装入写锁存器。
7. 将 EEADRH:EEADRL 寄存器对的内容加 1 以指向下一个单元。
8. 重复第 5 至 7 步，直到除最后一个写锁存器以外的所有写锁存器都已装载完毕为止。
9. 将 EECON1 寄存器的 LWLO 位清零。当 EECON1 寄存器的 LWLO 位为 0 时，写序列将启动对闪存程序存储器的写操作。
10. 将要写入的程序存储器数据装入 EEDATH:EEDATL 寄存器对。
11. 依次将 55h 和 AAh 写入 EECON2，然后将 EECON1 寄存器的 WR 位置 1（闪存编程解锁序列）。现在将整个锁存器块写入了闪存程序存储器。

不必将用户程序数据装入整个写锁存器块。但是，会将整个写锁存器块写入程序存储器。

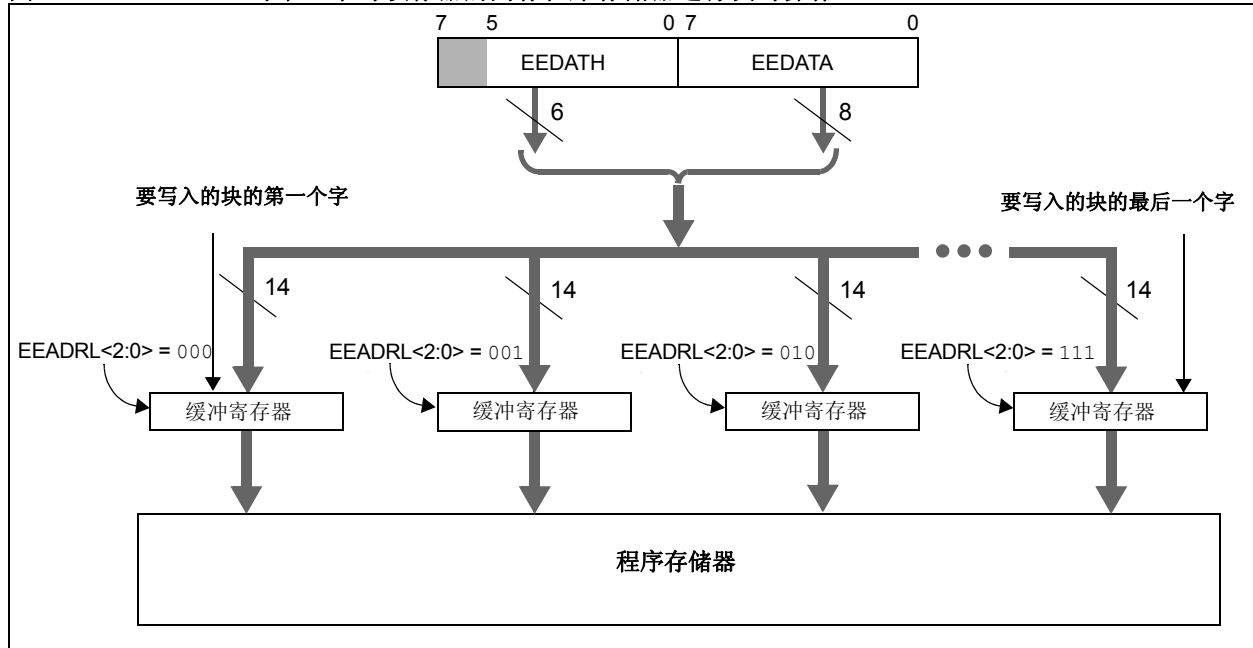
例 11-5 给出了完整的 8 字写序列示例。将初始地址装入 EEADRH:EEADRL 寄存器对，8 字数据通过间接寻址载入。

**注：** 例 11-5 中提供的代码序列必须重复多次，以完全编程擦除的程序存储器行。

在 `BSF EECON1, WR` 指令后，处理器需要两个周期来设置写操作。用户必须在 `WR` 位置 1 后执行两条 `NOP` 指令。处理器将暂停内部操作，通常为 2 ms，但仅在写操作发生（即，写入块的最后一个字）的周期内暂停。这不是

休眠模式，因为时钟和外设将继续运行。当 `LWLO = 1` 时，处理器不会停止，而是继续装入写锁存器。写周期后，处理器从 `EECON1` 写指令后的第三条指令继续操作。

图 11-2: 对带 8 个写锁存器的闪存程序存储器进行块写操作



# PIC16(L)F1933

## 例 11-4: 擦除程序存储器的一行

```
; This row erase routine assumes the following:
; 1. A valid address within the erase block is loaded in ADDRH:ADDRL
; 2. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)

        BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
        BANKSEL EEADRL
        MOVF     ADDRL,W         ; Load lower 8 bits of erase address boundary
        MOVWF   EEADRL
        MOVF     ADDRH,W        ; Load upper 6 bits of erase address boundary
        MOVWF   EEADRH
        BSF     EECON1,EEPGD     ; Point to program memory
        BCF     EECON1,CFGSR     ; Not configuration space
        BSF     EECON1,FREE      ; Specify an erase operation
        BSF     EECON1,WREN      ; Enable writes

        MOVLW   55h             ; Start of required sequence to initiate erase
        MOVWF   EECON2          ; Write 55h
        MOVLW   0AAh           ;
        MOVWF   EECON2          ; Write AAh
        BSF     EECON1,WR       ; Set WR bit to begin erase
        NOP                      ; Any instructions here are ignored as processor
        NOP                      ; halts to begin erase sequence
        NOP                      ; Processor will stop here and wait for erase complete.

        ; after erase processor continues with 3rd instruction

        BCF     EECON1,WREN      ; Disable writes
        BSF     INTCON,GIE      ; Enable interrupts
```

必需的  
序列

## 例 11-5: 写入闪存程序存储器

```

; This write routine assumes the following:
; 1. The 16 bytes of data are loaded, starting at the address in DATA_ADDR
; 2. Each word of data to be written is made up of two adjacent bytes in DATA_ADDR,
;    stored in little endian format
; 3. A valid starting address (the least significant bits = 000) is loaded in ADDRH:ADDRL
; 4. ADDRH and ADDRL are located in shared data memory 0x70 - 0x7F (common RAM)
;
    BCF      INTCON,GIE      ; Disable ints so required sequences will execute properly
    BANKSEL EEADRH          ; Bank 3
    MOVF    ADDRH,W         ; Load initial address
    MOVWF   EEADRH          ;
    MOVF    ADDRHL,W        ;
    MOVWF   EEADRL         ;
    MOVLW   LOW DATA_ADDR  ; Load initial data address
    MOVWF   FSR0L           ;
    MOVLW   HIGH DATA_ADDR ; Load initial data address
    MOVWF   FSR0H           ;
    BSF     EECON1,EEPGD    ; Point to program memory
    BCF     EECON1,CFGSR    ; Not configuration space
    BSF     EECON1,WREN     ; Enable writes
    BSF     EECON1,LWLO    ; Only Load Write Latches

LOOP
    MOVIW   FSR0++          ; Load first data byte into lower
    MOVWF   EEDATL         ;
    MOVIW   FSR0++          ; Load second data byte into upper
    MOVWF   EEDATH         ;

    MOVF    EEADRL,W        ; Check if lower bits of address are '000'
    XORLW   0x07           ; Check if we're on the last of 8 addresses
    ANDLW   0x07           ;
    BTFSC   STATUS,Z        ; Exit if last of eight words,
    GOTO    START_WRITE    ;

    MOVLW   55h             ; Start of required write sequence:
    MOVWF   EECON2          ; Write 55h
    MOVLW   0AAh           ;
    MOVWF   EECON2          ; Write AAh
    BSF     EECON1,WR       ; Set WR bit to begin write
    NOP     ; Any instructions here are ignored as processor
    NOP     ; halts to begin write sequence
    NOP     ; Processor will stop here and wait for write to complete.
    ; After write processor continues with 3rd instruction.

    INCF    EEADRL,F        ; Still loading latches Increment address
    GOTO    LOOP           ; Write next latches

START_WRITE
    BCF     EECON1,LWLO    ; No more loading latches - Actually start Flash program
    ; memory write

    MOVLW   55h             ; Start of required write sequence:
    MOVWF   EECON2          ; Write 55h
    MOVLW   0AAh           ;
    MOVWF   EECON2          ; Write AAh
    BSF     EECON1,WR       ; Set WR bit to begin write
    NOP     ; Any instructions here are ignored as processor
    NOP     ; halts to begin write sequence
    NOP     ; Processor will stop here and wait for write complete.
    ; after write processor continues with 3rd instruction

    BCF     EECON1,WREN    ; Disable writes
    BSF     INTCON,GIE     ; Enable interrupts

```

# PIC16(L)F1933

## 11.4 修改闪存程序存储器

修改程序存储器行中的现有数据，并且该行内的数据必须保留时，必须首先读取它并将其保存在RAM映像中。使用以下步骤修改程序存储器：

1. 装入要修改的行的起始地址。
2. 从行中读取现有数据并将其保存到RAM映像中。
3. 修改RAM映像以包含要写入到程序存储器的新数据。
4. 装入要重新写入的行的起始地址。
5. 擦除程序存储器行。
6. 将来自RAM映像的数据装入写锁存器。
7. 启动编程操作。
8. 根据需要重复第6至7步多次，以对擦除行进行重新编程。

## 11.5 用户ID、器件ID和配置字访问

当EECON1寄存器中的CFGs = 1时，不是访问程序存储器或EEPROM数据存储器，而是访问用户ID、器件ID/版本ID和配置字。这是PC<15> = 1时指向的区域，但不是所有的地址都可访问。对于读和写操作存在不同的访问权限。请参见表11-2。

在对表11-2列出的参数以外的地址进行读访问时，EEDATH:EEDATL寄存器对将被清零。

表 11-2: 用户ID、器件ID和配置字访问 (CFGs = 1)

地址	功能	读访问	写访问
8000h-8003h	用户ID	允许	允许
8006h	器件ID/版本ID	允许	禁止
8007h-8008h	配置字1和2	允许	禁止

例 11-3: 配置字和器件ID访问

```
* This code block will read 1 word of program memory at the memory address:
*   PROG_ADDR_LO (must be 00h-08h) data will be returned in the variables;
*   PROG_DATA_HI, PROG_DATA_LO

BANKSEL  EEADRL          ; Select correct Bank
MOVLW   PROG_ADDR_LO    ;
MOVWF   EEADRL          ; Store LSB of address
CLRF   EEADRH           ; Clear MSB of address

BSF     EECON1,CFGs     ; Select Configuration Space
BCF     INTCON,GIE      ; Disable interrupts
BSF     EECON1,RD       ; Initiate read
NOP     ; Executed (See Figure 11-1)
NOP     ; Ignored (See Figure 11-1)
BSF     INTCON,GIE      ; Restore interrupts

MOVF   EEDATL,W         ; Get LSB of word
MOVWF  PROG_DATA_LO    ; Store in user location
MOVF   EEDATH,W         ; Get MSB of word
MOVWF  PROG_DATA_HI    ; Store in user location
```



## 11.6 写校验

根据具体应用，将写入数据 EEPROM 或者程序存储器中的值对照期望写入的值进行校验（见例 11-6）是一个良好的编程习惯。例 11-6 显示了如何校验对 EEPROM 的写操作。

### 例 11-6: EEPROM 写校验

```
BANKSEL EEDATL      ;
MOVWF  EEDATL, W    ;EEDATL not changed
                    ;from previous write
BSF    EECON1, RD   ;YES, Read the
                    ;value written
XORWF  EEDATL, W    ;
BTFSS  STATUS, Z    ;Is data the same
GOTO   WRITE_ERR    ;No, handle error
:      ;Yes, continue
```

# PIC16(L)F1933

## 寄存器 11-1: EEDATL: EEPROM 数据低字节寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
EEDAT<7:0>							
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

bit 7-0                      **EEDAT<7:0>**: EEPROM 数据字节或程序存储器的低 8 位的读 / 写值

## 寄存器 11-2: EEDATH: EEPROM 数据高字节寄存器

U-0	U-0	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
EEDAT<13:8>							
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

bit 7-6                      **未实现**: 读为 0

bit 5-0                      **EEDAT<13:8>**: 程序存储器高 6 位的读 / 写值

## 寄存器 11-3: EEADRL: EEPROM 地址低字节寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EEADR<7:0>							
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

bit 7-0                      **EEADR<7:0>**: 指定程序存储器地址或 EEPROM 地址的低 8 位

## 寄存器 11-4: EEADRH: EEPROM 地址高字节寄存器

U-1	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
EEADR<14:8>							
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

bit 7                          **未实现**: 读为 1

bit 6-0                      **EEADR<14:8>**: 指定程序存储器地址或 EEPROM 地址的高 7 位

## 寄存器 11-5: EECON1: EEPROM 控制寄存器 1

R/W-0/0	R/W-0/0	R/W-0/0	R/W/HC-0/0	R/W-x/q	R/W-0/0	R/S/HC-0/0	R/S/HC-0/0
EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD
bit 7						bit 0	

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
S = 只可置 1 位	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 由硬件清零

- bit 7 **EEPGD:** 闪存程序存储器 / 数据 EEPROM 存储器选择位  
 1 = 访问程序空间闪存存储器  
 0 = 访问数据 EEPROM 存储器
- bit 6 **CFGS:** 闪存程序存储器 / 数据 EEPROM 或配置寄存器选择位  
 1 = 访问配置寄存器、用户 ID 寄存器和器件 ID 寄存器  
 0 = 访问闪存程序存储器或数据 EEPROM 存储器
- bit 5 **LWLO:** 仅装载写锁存器位  
如果 CFGS = 1 (配置空间) 或 CFGS = 0 且 EEPGD = 1 (闪存程序存储器):  
 1 = 下一条 WR 命令不会启动写操作; 仅更新程序存储器锁存器。  
 0 = 下一条 WR 命令将来自 EEDATH:EEDATL 的值写入程序存储器锁存器, 并启动对存储在程序存储器锁存器中的所有数据的写操作。  
如果 CFGS = 0 且 EEPGD = 0: (访问数据 EEPROM)  
 忽略 LWLO。下一条 WR 命令启动对数据 EEPROM 的写操作。
- bit 4 **FREE:** 程序闪存擦除使能位  
如果 CFGS = 1 (配置空间) 或 CFGS = 0 且 EEPGD = 1 (闪存程序存储器):  
 1 = 在下一条 WR 命令执行擦除操作 (擦除完成后由硬件清零)。  
 0 = 在下一条 WR 命令执行写操作。  
如果 EEPGD = 0 且 CFGS = 0: (访问数据 EEPROM)  
 忽略 FREE。下一条 WR 命令将启动一个擦除周期和一个写周期。
- bit 3 **WRERR:** EEPROM 错误标志位  
 1 = 此状态表示试图进行不当的编程或擦除序列或意外终止 (试图将 WR 位置 1 (写入 1) 时, 该位自动置 1)。  
 0 = 编程或擦除操作正常完成。
- bit 2 **WREN:** 编程 / 擦除使能位  
 1 = 允许执行编程 / 擦除操作  
 0 = 禁止编程 / 擦除闪存程序存储器和数据 EEPROM
- bit 1 **WR:** 写控制位  
 1 = 启动闪存程序存储器或数据 EEPROM 编程 / 擦除操作。  
 该操作是自定时的, 且该位在操作完成时由硬件清零。  
 用软件只能将 WR 位置 1, 但不能清零。  
 0 = 对闪存或数据 EEPROM 的编程 / 擦除操作已完成, 当前不在进行中。
- bit 0 **RD:** 读控制位  
 1 = 启动对闪存程序存储器或数据 EEPROM 的读操作。读操作只占用一个周期。RD 由硬件清零。用软件只能将 RD 位置 1, 但不能清零。  
 0 = 不启动闪存程序存储器或数据 EEPROM 读操作。

# PIC16(L)F1933

**寄存器 11-6: EECON2: EEPROM 控制寄存器 2**

W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0	W-0/0
EEPROM 控制寄存器 2							
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
S = 只可置 1 位	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **数据 EEPROM 解锁模式位**

要解锁写操作, 在置 1 EECON1 寄存器的 WR 位前, 必须先写 55h, 随后是 AAh。写入该寄存器的值用于解锁写操作。对这些写操作有特殊的时序要求。更多信息, 请参见第 11.2.2 节“写数据 EEPROM 存储器”。

**表 11-3: 与数据 EEPROM 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
EECON1	EEPGD	CFGS	LWLO	FREE	WRERR	WREN	WR	RD	115
EECON2	EEPROM 控制寄存器 2 (非物理寄存器)								103*
EEADRL	EEADRL<7:0>								114
EEADRH	—	EEADRH<6:0>							114
EEDATL	EEDATL<7:0>								114
EEDATH	—	—	EEDATH<5:0>						114
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91

**图注:** — = 未实现单元, 读为 0。数据 EEPROM 模块不使用阴影单元。

\* 该页提供寄存器信息。

## 12.0 I/O 端口

根据所选择的器件和所使能的外设，最多有 5 个端口可供使用。通常情况下，当某个端口上的外设使能时，其相关引脚可能不能用作通用输出引脚，但可读取该引脚。

每个端口有 3 个标准寄存器供其操作使用。这些寄存器包括：

- TRISx 寄存器（数据方向寄存器）
- PORTx 寄存器（用于读器件引脚上的电平）
- LATx 寄存器（输出锁存器）

某些端口可能有以下额外寄存器中的一个或几个。这些寄存器包括：

- ANSELx（模拟选择寄存器）
- WPUx（弱上拉寄存器）

表 12-1: 每个器件的端口可用性

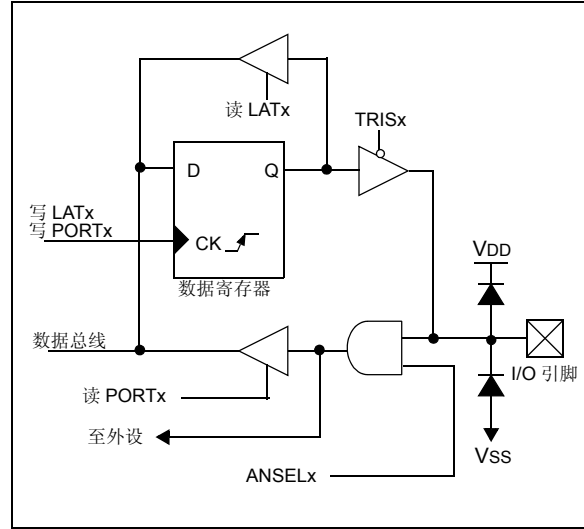
器件	PORTA	PORTB	PORTC	PORTE
PIC16(L)F1933	•	•	•	•

数据锁存器（LATx 寄存器）用于对 I/O 引脚所驱动的值进行读 - 修改 - 写操作。

对 LATx 寄存器的写操作与对相应 PORTx 寄存器的写操作具有相同的效果。读 LATx 寄存器将读取保存在 I/O 端口锁存器中的值，而读 PORTx 寄存器将读取实际的 I/O 引脚值。

支持模拟输入的端口具有相关的 ANSELx 寄存器。当 ANSEL 位置 1 时，禁止与该位相关的数字输入缓冲器。禁止输入缓冲器可防止处于逻辑高电平和逻辑低电平之间的引脚上的模拟信号电平造成逻辑输入电路中产生过电流。图 12-1 给出了一个通用 I/O 端口的简化模型，图中未显示与其他外设的接口。

图 12-1: 通用 I/O 端口操作



例 12-1: 初始化 PORTA

```

; This code example illustrates
; initializing the PORTA register. The
; other ports are initialized in the same
; manner.

BANKSEL PORTA      ;
CLRF PORTA         ;Init PORTA
BANKSEL LATA       ;Data Latch
CLRF LATA          ;
BANKSEL ANSELA    ;
CLRF ANSELA       ;digital I/O
BANKSEL TRISA     ;
MOVLW B'00111000' ;Set RA<5:3> as inputs
MOVWF TRISA       ;and set RA<2:0> as
                  ;outputs
    
```

# PIC16(L)F1933

## 12.1 备用引脚功能

备用引脚功能控制 (APFCON) 寄存器用于将特定外设输入和输出功能配置到不同的引脚上。APFCON 寄存器如寄存器 12-1 所示。对于此器件系列，以下功能可以配置到不同的引脚上。

- $\overline{SS}$  (从选择)
- P2B 输出
- CCP2/P2A 输出
- CCP3/P3A 输出
- Timer1 门控
- SR 锁存器 SRnQ 输出
- 比较器 C2 输出

这些位对任何 TRIS 寄存器的值都没有影响。PORT 和 TRIS 的更改可传递给正确的引脚。未选择的引脚不受影响。

寄存器 12-1: APFCON: 备用引脚功能控制寄存器

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CCP3SEL	T1GSEL	P2BSEL	SRnQSEL	C2OUTSEL	SSSEL	CCP2SEL
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7 **未实现:** 读为 0。

bit 6 **CCP3SEL:** CCP3 输入 / 输出引脚选择位  
0 = CCP3/P3A 功能位于 RC6/TX/CK/CCP3/P3A/SEG9 引脚上  
1 = CCP3/P3A 功能位于 RB5/AN13/CPS5/CCP3/P3A/T1G/COM1 引脚上

bit 5 **T1GSEL:** Timer1 门控输入引脚选择位  
0 = T1G 功能位于 RB5/AN13/CPS5/CCP3/P3A/T1G/COM1 引脚上  
1 = T1G 功能位于 RC4/SDI/SDA/T1G/SEG11 引脚上

bit 4 **P2BSEL:** CCP2 PWM B 输出引脚选择位  
0 = P2B 功能位于 RC0/T1OSO/T1CKI/P2B 引脚上  
1 = P2B 功能位于 RB5/AN13/P2B/CPS5/T1G/COM1 引脚上

bit 3 **SRnQSEL:** SR 锁存器 nQ 输出引脚选择位  
0 = SRnQ 功能位于 RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub> 引脚上  
1 = SRnQ 功能位于 RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub> 引脚上

bit 2 **C2OUTSEL:** 比较器 C2 输出引脚选择位  
0 = C2OUT 功能位于 RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub> 引脚上  
1 = C2OUT 功能位于 RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub> 引脚上

bit 1 **SSSEL:**  $\overline{SS}$  输入引脚选择位  
0 =  $\overline{SS}$  功能位于 RA5/AN4/C2OUT/SRnQ/ $\overline{SS}$ /CPS7/SEG5/V<sub>CAP</sub> 引脚上  
1 =  $\overline{SS}$  功能位于 RA0/AN0/C12IN0-/C2OUT/SRnQ/ $\overline{SS}$ /SEG12/V<sub>CAP</sub> 引脚上

bit 0 **CCP2SEL:** CCP2 输入 / 输出引脚选择位  
0 = CCP2/P2A 功能位于 RC1/T1OSI/CCP2/P2A 引脚上  
1 = CCP2/P2A 功能位于 RB3/AN9/C12IN2-/CPS3/CCP2/P2A/VLCD3 引脚上

## 12.2 PORTA 寄存器

PORTA 是 8 位宽的双向端口。其对应的数据方向寄存器是 TRISA（寄存器 12-3）。将 TRISA 某位置 1 (= 1) 可以让相应 PORTA 引脚作为输入引脚（即禁止输出驱动器）。将 TRISA 某位清零 (= 0) 可以让相应 PORTA 引脚作为输出引脚（即，使能输出驱动器并将输出锁存器的内容输出到所选择的引脚）。例 12-1 说明了如何初始化 I/O 端口。

读 PORTA 寄存器（寄存器 12-2）读的是引脚的状态，而写该寄存器将会写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器（LATA）。

即使在 PORTA 引脚用作模拟输入引脚的时候，TRISA 寄存器（寄存器 12-3）仍然控制 PORTA 引脚的输出驱动器。在将它们用作模拟输入引脚时，用户必须确保 TRISA 寄存器中的位保持为置 1 状态。配置为模拟输入引脚的 I/O 引脚总是读为 0。

### 12.2.1 ANSELA 寄存器

ANSELA 寄存器（寄存器 12-5）用于将 I/O 引脚的输入模式配置为模拟模式。将相应的 ANSELA 位设置为高电平将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正常工作。

ANSELA 位的状态不会影响数字输出功能。TRIS 清零且 ANSEL 置 1 的引脚将仍作为数字输出引脚工作，但输入模式将变为模拟输入模式。当在受影响的端口上执行读 - 修改 - 写指令时，这会引发意外操作。

**注：** 复位后，ANSELA 位默认为模拟模式。要将任一引脚用作数字通用或者外设输入，相应的 ANSEL 位必须由用户软件初始化为 0。

### 12.2.2 PORTA 功能和输出优先级

每个 PORTA 引脚均复用多个功能。表 12-2 给出了引脚及其组合功能和输出优先级。

当使能多个输出时，引脚实际上由以下列表中编号最小的外设控制。

优先级列表中未显示模拟输入功能（例如 ADC、比较器和电容触摸传感输入）。使用 ANSELx 寄存器将 I/O 引脚设置为模拟模式时，这些输入是有效的。当引脚处于模拟模式并具有表 12-2 所示的优先级时，数字输出功能可以控制该引脚。

表 12-2: PORTA 输出优先级

引脚名称	功能优先级 <sup>(1)</sup>
RA0	VCAP SEG12 (LCD) SRNQ (SR 锁存器) C2OUT (比较器) RA0
RA1	SEG7 (LCD) RA1
RA2	COM2 (LCD) AN2 (DAC) RA2
RA3	COM3 (LCD)，仅 28 引脚器件 SEG15 RA3
RA4	SEG4 (LCD) SRQ (SR 锁存器) C1OUT (比较器) CCP5，仅 28 引脚器件 RA4
RA5	VCAP (由配置字寄存器使能) SEG5 (LCD) SRNQ (SR 锁存器) C2OUT (比较器) RA5
RA6	VCAP (由配置字寄存器使能) OSC2 (由配置字寄存器使能) CLKOUT (由配置字寄存器使能) SEG1 (LCD) RA6
RA7	OSC1/CLKIN (由配置字寄存器使能) SEG2 (LCD) RA7

注 1: 按优先级从高至低排列。

# PIC16(L)F1933

## 寄存器 12-2: PORTA: PORTA 寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                        0 = 清零

bit 7-0            **RA<7:0>**: PORTA I/O 值位 <sup>(1)</sup>  
1 = 端口引脚电平 > V<sub>IH</sub>  
0 = 端口引脚电平 < V<sub>IL</sub>

注 1: 写 PORTA 实际上写的是相应的 LATA 寄存器。读 PORTA 寄存器返回的是实际的 I/O 引脚值。

## 寄存器 12-3: TRISA: PORTA 三态寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                        0 = 清零

bit 7-0            **TRISA<7:0>**: PORTA 三态控制位  
1 = PORTA 引脚配置为输入 (三态)  
0 = PORTA 引脚配置为输出

## 寄存器 12-4: LATA: PORTA 数据锁存寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0
bit 7							bit 0

### 图注:

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                        0 = 清零

bit 7-0            **LATA<7:0>**: PORTA 输出锁存器值位 <sup>(1)</sup>

注 1: 写 PORTA 实际上写的是相应的 LATA 寄存器。读 PORTA 寄存器返回的是实际的 I/O 引脚值。



## 寄存器 12-5: ANSELA: PORTA 模拟选择寄存器

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6      **未实现:** 读为 0

bit 5-0      **ANSA<5:0>:** 将 RA<5:0> 引脚选择为模拟或数字功能

0 = 数字 I/O。引脚被分配为端口或是数字特殊功能。

1 = 模拟输入。引脚被分配为模拟输入引脚<sup>(1)</sup>。禁止数字输入缓冲器。

**注 1:** 当引脚被设置为模拟输入引脚时, 必须设置相应的 TRIS 位以将对应的引脚配置为输入模式从而允许外部控制引脚电压。

# PIC16(L)F1933

表 12-3: 与 PORTA 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ADCON0	—	CHS<4:0>					GO/DONE	ADON	145
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		146
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	121
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1SP	C1HYS	C1SYNC	165
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	165
CM1CON1	C1NTP	C1INTN	C1PCH<1:0>		—	—	C1NCH<1:0>		166
CM2CON1	C2NTP	C2INTN	C2PCH<1:0>		—	—	C2NCH<1:0>		166
CPSCON0	CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	TOXCS	311
CPSCON1	—	—	—	—	—	CPSCH<2:0>			312
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	158
LATA	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	120
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		317
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	321
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	321
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			175
PORTA	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	120
SRCON0	SRLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR	171
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				269
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120

图注: x = 未知, u = 不变, — = 未实现单元, 读为 0。PORTA 不使用阴影单元。

表 12-4: 与 PORTA 相关的配置字寄存器汇总

名称	位	Bit -/7	Bit -/6	Bit 13/5	Bit 12/4	Bit 11/3	Bit 10/2	Bit 9/1	Bit 8/0	寄存器所在的页面
CONFIG1	13:8	—	—	FCMEN	IESO	CLKOUTEN	BOREN<1:0>		CPD	50
	7:0	CP	MCLRE	PWRTE	WDTE<1:0>		FOSC<2:0>			
CONFIG2	13:8	—	—	LVP	DEBUG	—	BORV	STVREN	PLLEN	52
	7:0	—	—	VCAPEN<1:0> <sup>(1)</sup>		—	—	WRT<1:0>		

图注: — = 未实现单元, 读为 0。PORTA 不使用阴影单元。

注 1: 仅限 PIC16F1933 器件。

## 12.3 PORTB 寄存器

PORTB 是 8 位宽的双向端口。其对应的数据方向寄存器是 TRISB（寄存器 12-7）。将 TRISB 某位置 1 (= 1) 可以让相应 PORTB 引脚作为输入引脚（即将相应的输出驱动器置于高阻模式）。将 TRISB 某位清零 (= 0) 可以让相应 PORTB 引脚作为输出引脚（即，使能输出驱动器并将输出锁存器的内容输出到所选择的引脚）。例 12-1 说明了如何初始化 I/O 端口。

读 PORTB 寄存器（寄存器 12-6）读的是引脚的状态，而写该寄存器将会写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器（LATB）。

即使在 PORTB 引脚用作模拟输入引脚的时候，TRISB 寄存器（寄存器 12-7）仍然控制 PORTB 引脚的输出驱动器。在将它们用作模拟输入引脚时，用户必须确保 TRISB 寄存器中的位保持为置 1 状态。配置为模拟输入引脚的 I/O 引脚总是读为 0。

### 12.3.1 弱上拉

每个 PORTB 引脚均具有可单独配置的内部弱上拉。控制位 WPUB<7:0> 可使能或禁止每个上拉（见寄存器 12-10）。当将端口引脚配置为输出时，其弱上拉电路会自动切断。在上电复位时通过 OPTION\_REG 寄存器的 WPUEN 位禁止所有上拉。

### 12.3.2 电平变化中断

所有 PORTB 引脚都可单独配置为电平变化中断引脚。控制位 IOCB<7:0> 可允许或禁止每个引脚的中断功能。电平变化中断功能在上电复位时禁止。更多信息，请参见第 13.0 节“电平变化中断”。

### 12.3.3 ANSELB 寄存器

ANSELB 寄存器（寄存器 12-9）用于将 I/O 引脚的输入模式配置为模拟模式。将相应的 ANSELB 位设置为高电平将使引脚上的所有数字读操作都读为 0，并允许引脚上的模拟功能正常工作。

ANSELB 位的状态不会影响数字输出功能。TRIS 清零且 ANSELB 置 1 的引脚将仍作为数字输出引脚工作，但输入模式将变为模拟输入模式。当在受影响的端口上执行读 - 修改 - 写指令时，得到的结果可能与预期不符。

**注：** 复位后，ANSELB 位默认为模拟模式。要将任一引脚用作数字通用或者外设输入，相应的 ANSEL 位必须由用户软件初始化为 0。

# PIC16(L)F1933

## 12.3.4 PORTB 功能和输出优先级

每个 PORTB 引脚均复用多个功能。表 12-5 给出了引脚及其组合功能和输出优先级。

当使能多个输出时，引脚实际上由以下列表中编号最小的外设控制。

以下列表中未包含模拟输入和某些数字输入功能。当引脚配置为输出时，这些输入功能仍保持有效。某些数字输入功能可覆盖其他端口功能，并包含在表 12-5 中。

**表 12-5: PORTB 输出优先级**

引脚名称	功能优先级 <sup>(1)</sup>
RB0	SEG0 (LCD) CCP4, 仅 28 引脚器件 RB0
RB1	P1C (ECCP1), 仅 28 引脚器件 RB1
RB2	P1B (ECCP1), 仅 28 引脚器件 RB2
RB3	CCP2/P2A RB3
RB4	COM0 P1D, 仅 28 引脚器件 RB4
RB5	COM1 P2B, 仅 28 引脚器件 CCP3/P3A RB5
RB6	ICSPCLK (编程) ICDCLK (由配置字寄存器使能) SEG14 (LCD) RB6
RB7	ICSPDAT (编程) ICDDAT (由配置字寄存器使能) SEG13 (LCD) RB7

注 1: 按优先级从高至低排列。

## 寄存器 12-6: PORTB: PORTB 寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **RB<7:0>**: PORTB I/O 引脚位  
 1 = 端口引脚电平 > V<sub>IH</sub>  
 0 = 端口引脚电平 < V<sub>IL</sub>

## 寄存器 12-7: TRISB: PORTB 三态寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **TRISB<7:0>**: PORTB 三态控制位  
 1 = PORTB 引脚配置为输入 (三态)  
 0 = PORTB 引脚配置为输出

## 寄存器 12-8: LATB: PORTB 数据锁存寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **LATB<7:0>**: PORTB 输出锁存器值位<sup>(1)</sup>

注 1: 写 PORTB 实际上写的是相应的 LATB 寄存器。读 PORTB 寄存器返回的是实际的 I/O 引脚值。

# PIC16(L)F1933

## 寄存器 12-9: ANSELB: PORTB 模拟选择寄存器

U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6 未实现: 读为 0

bit 5-0 **ANSB<5:0>**: 将 RB<5:0> 引脚选择为模拟或数字功能  
0 = 数字 I/O。引脚被分配为端口或是数字特殊功能。  
1 = 模拟输入。引脚被分配为模拟输入引脚<sup>(1)</sup>。禁止数字输入缓冲器。

注 1: 当引脚被设置为模拟输入引脚时, 必须设置相应的 TRIS 位以将对应的引脚配置为输入模式从而允许外部控制引脚电压。

## 寄存器 12-10: WPUB: 弱上拉 PORTB 寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0 **WPUB<7:0>**: 弱上拉寄存器位  
1 = 使能上拉  
0 = 禁止上拉

注 1: 必须将 OPTION\_REG 寄存器的全局  $\overline{\text{WPUEN}}$  位清零, 才能使能各个上拉。  
2: 如果引脚配置为输出模式, 则弱上拉器件将自动禁止。

**表 12-6: 与 PORTB 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ADCON0	—	CHS<4:0>					GO/DONE	ADON	145
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	126
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
CCPxCON	PxM<1:0>		DCxB<1:0>		CCPxM<3:0>				216
CPSCON0	CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	TOXCS	311
CPSCON1	—	—	—	—	—	CPSCH<2:0>			312
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOCF	86
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	134
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	134
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	134
LATB	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	125
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		317
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	321
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	321
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			175
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	125
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		186
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	WPUB3	WPUB2	WPUB1	WPUB0	126

**图注:** x = 未知, u = 不变, - = 未实现单元, 读为 0。PORTB 不使用阴影单元。

# PIC16(L)F1933

## 12.4 PORTC 寄存器

PORTC 是 8 位宽的双向端口。其对应的数据方向寄存器是 TRISC（寄存器 12-12）。将 TRISC 某位置 1 (= 1) 可以让相应 PORTC 引脚作为输入引脚（即将相应的输出驱动器置于高阻模式）。将 TRISC 某位清零 (= 0) 可以让相应 PORTC 引脚作为输出引脚（即，使能输出驱动器并将输出锁存器的内容输出到所选择的引脚）。例 12-1 说明了如何初始化 I/O 端口。

读 PORTC 寄存器（寄存器 12-11）读的是引脚的状态，而写该寄存器将会写入端口锁存器。所有写操作都是读 - 修改 - 写操作。因此，写一个端口就意味着读该端口的引脚电平，修改读到的值，然后再将改好的值写入端口数据锁存器（LATC）。

即使在 PORTC 引脚用作模拟输入引脚的时候，TRISC 寄存器（寄存器 12-12）仍然控制 PORTC 引脚的输出驱动器。在将它们用作模拟输入引脚时，用户必须确保 TRISC 寄存器中的位保持为置 1 状态。配置为模拟输入引脚的 I/O 引脚总是读为 0。

### 12.4.1 PORTC 功能和输出优先级

每个 PORTC 引脚均复用多个功能。表 12-7 给出了引脚及其组合功能和输出优先级。

当使能多个输出时，引脚实际上由以下列表中编号最小的外设控制。

以下列表中未包含模拟输入和某些数字输入功能。当引脚配置为输出时，这些输入功能仍保持有效。某些数字输入功能可覆盖其他端口功能，并包含在表 12-7 中。

表 12-7: PORTC 输出优先级

引脚名称	功能优先级 <sup>(1)</sup>
RC0	T1OSO (Timer1 振荡器) CCP2/P2B RC0
RC1	T1OSI (Timer1 振荡器) CCP2/P2A RC1
RC2	SEG3 (LCD) CCP1/P1A RC2
RC3	SEG6 (LCD) SCL (MSSP) SCK (MSSP) RC3
RC4	SEG11 (LCD) SDA (MSSP) RC4
RC5	SEG10 (LCD) SDO (MSSP) RC5
RC6	ISEG9 (LCD) TX (EUSART) CK (EUSART) CCP3/P3A, 仅 28 引脚器件 RC6
RC7	SEG8 (LCD) DT (EUSART) CCP3/P3B, 仅 28 引脚器件 RC7

注 1: 按优先级从高至低排列。



## 寄存器 12-11: PORTC: PORTC 寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **RC<7:0>**: PORTC 通用 I/O 引脚位  
 1 = 端口引脚电平 > V<sub>IH</sub>  
 0 = 端口引脚电平 < V<sub>IL</sub>

## 寄存器 12-12: TRISC: PORTC 三态寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **TRISC<7:0>**: PORTC 三态控制位  
 1 = PORTC 引脚配置为输入 (三态)  
 0 = PORTC 引脚配置为输出

## 寄存器 12-13: LATC: PORTC 数据锁存寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **LATC<7:0>**: PORTC 输出锁存器值位 <sup>(1)</sup>

注 1: 写 PORTC 实际上写的是相应的 LATC 寄存器。读 PORTC 寄存器返回的是实际的 I/O 引脚值。

# PIC16(L)F1933

表 12-8: 与 PORTC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
CCPxCON	PxM<1:0>		DCxB<1:0>		CCPxM<3:0>				216
LATC	LATC7	LATC6	LATC5	LATC4	LATC3	LATC2	LATC1	LATC0	129
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		317
LCDSE0	SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	321
LCDSE1	SE15	SE14	SE13	SE12	SE11	SE10	SE9	SE8	321
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	129
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				269
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	268
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	284
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129

图注: x = 未知, u = 不变, - = 未实现单元, 读为 0。PORTC 不使用阴影单元。

## 12.5 PORTE 寄存器

RE3 为仅输入引脚，也可用作  $\overline{\text{MCLR}}$ 。可通过配置熔丝位禁止 MCLR 功能。RE3 还提供编程电压。RE3 的 TRIS 位 (TRISE3) 始终读为 1。

### 12.5.1 PORTE 功能和输出优先级

RE3 为仅输入引脚，无输出优先级。

#### 寄存器 12-14: PORTE: PORTE 寄存器

U-0	U-0	U-0	U-0	R-x/u	U-0	U-0	U-0
—	—	—	—	RE3	—	—	—
bit 7							bit 0

#### 图注:

R = 可读位

W = 可写位

U = 未实现位，读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 未实现: 读为 0

bit 3 **RE3:** PORTE 输入引脚位  
1 = 端口引脚电平 > V<sub>IH</sub>  
0 = 端口引脚电平 < V<sub>IL</sub>

bit 2-0 未实现: 读为 0

#### 寄存器 12-15: TRISE: PORTE 三态寄存器

U-0	U-0	U-0	U-0	U-1 <sup>(1)</sup>	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

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-4 未实现: 读为 0

bit 3 未实现: 读为 1

bit 2-0 未实现: 读为 0

注 1: 未实现，读为 1。

# PIC16(L)F1933

**寄存器 12-16: WPUE: 弱上拉 PORTE 寄存器**

U-0	U-0	U-0	U-0	R/W-1/1	U-0	U-0	U-0
—	—	—	—	WPUE3	—	—	—
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7-4      **未实现:** 读为 0
- bit 3      **WPUE:** 弱上拉寄存器位  
1 = 使能上拉  
0 = 禁止上拉
- bit 2-0      **未实现:** 读为 0

- 注 1:** 必须将 OPTION\_REG 寄存器的全局  $\overline{\text{WPUEN}}$  位清零, 才能使能各个上拉。
- 注 2:** 如果引脚配置为输出, 则弱上拉器件将自动禁止。

**表 12-9: 与 PORTE 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ADCON0	—	CHS<4:0>					GO/DONE	ADON	145
CCPxCON	PxM<1:0>		DCxB<1:0>		CCPxM<3:0>				216
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		317
LCDSE2	SE23	SE22	SE21	SE20	SE19	SE18	SE17	SE16	321
PORTE	—	—	—	—	RE3	—	—	—	131
TRISE	—	—	—	—	— <sup>(3)</sup>	—	—	—	131
WPUE	—	—	—	—	WPUE3	—	—	—	132

**图注:** x = 未知, u = 不变, — = 未实现单元, 读为 0。PORTE 不使用阴影单元。

**注 1:** 未实现, 读为 1。

## 13.0 电平变化中断

PORTB 引脚可进行配置，以作为电平变化中断（Interrupt-On-Change, IOC）引脚使用。当检测到具有上升沿或下降沿的信号时产生中断。任何独立的 PORT IOC 引脚或 PORT IOC 引脚组合，都可进行配置以产生中断。电平变化中断模块具有以下功能：

- 允许电平变化中断（主开关）
- 独立的引脚配置
- 上升沿和下降沿检测
- 独立的引脚中断标志

图 13-1 给出了 IOC 模块的框图。

### 13.1 使能模块

要允许独立的 PORTB 引脚产生中断，必须将 INTCON 寄存器的 IOCFE 位置 1。如果 IOCFE 位被禁止，则仍然会对引脚上的信号进行边沿检测，但是不产生中断。

### 13.2 独立的引脚配置

每个 PORTB 引脚都带有一个上升沿检测器和一个下降沿检测器。要使某个引脚检测上升沿，必须将 IOCBP 寄存器中相应的 IOCBPx 位置 1。要使某个引脚检测下降沿，必须将 IOCBN 寄存器中相应的 IOCBNx 位置 1。

通过分别将 IOCBP 寄存器中相应的 IOCBPx 位和 IOCBN 寄存器中相应的 IOCBNx 位置 1，可将某个引脚配置为同时检测上升沿和下降沿。

## 13.3 中断标志

IOCBF 寄存器中的 IOCBFx 位是与 PORTB 的电平变化中断引脚对应的状态标志位。如果在相应的已使能引脚上检测到预期的边沿，则该引脚的状态标志位将置 1，且如果 IOCFE 位置 1 的话，还会产生中断。INTCON 寄存器的 IOCF 位反映所有 IOCBFx 位的状态。

### 13.4 清零中断标志

各个状态标志（IOCBFx 位）可通过将其复位为零进行清零。如果在清零操作期间又检测到一个边沿，那么在清零序列结束后相关状态标志位会置 1，而与实际写入的值无关。

为了确保清零标志时不丢失所检测到的边沿，只需执行用于屏蔽已知更改位的“逻辑与”操作。以下举例说明了应执行的序列。

#### 例 13-1:

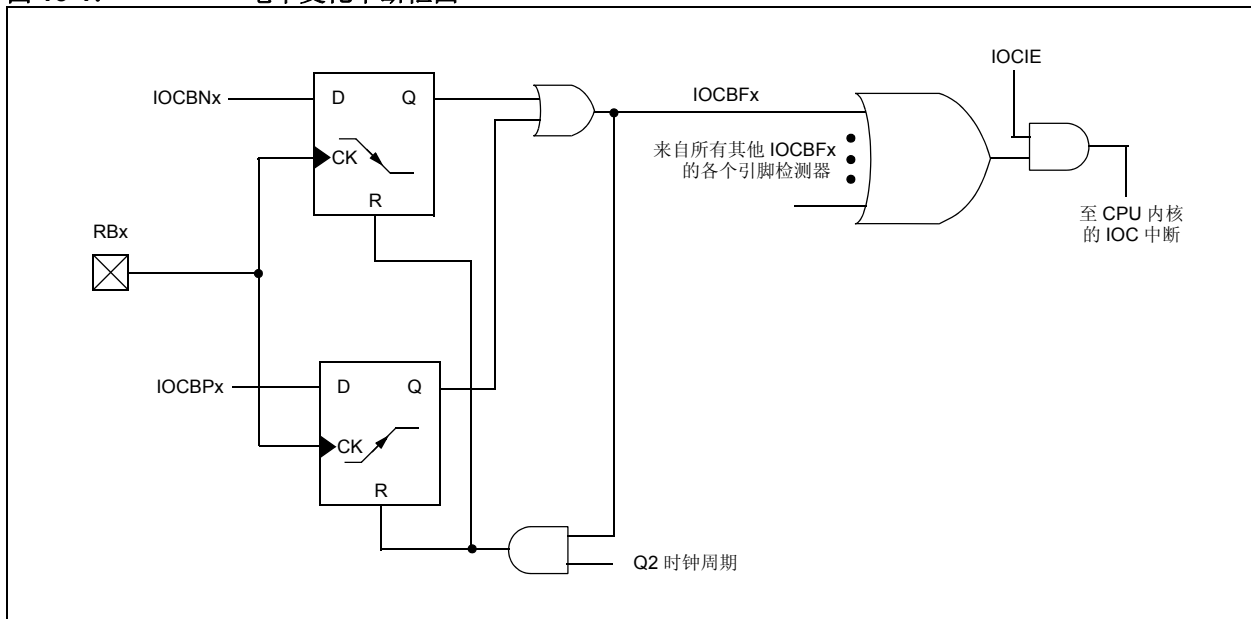
```
MOVLW 0xff
XORWF IOCBF, W
ANDWF IOCBF, F
```

### 13.5 休眠期间的工作原理

如果 IOCFE 位置 1 的话，电平变化中断序列会将器件从休眠模式唤醒。

如果在休眠模式下检测到边沿，则在退出休眠模式执行第一条指令之前，会更新 IOCBF 寄存器。

图 13-1: 电平变化中断框图



# PIC16(L)F193X

## 13.6 电平变化中断寄存器

寄存器 13-1: **IOCBP**: 正边沿电平变化中断寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                         0 = 清零

bit 7-0            **IOCBP<7:0>**: 正边沿电平变化中断允许位  
1 = 允许引脚上的正边沿电平变化中断。检测到边沿时, 将相关状态位和中断标志位置 1。  
0 = 禁止相关引脚的电平变化中断。

寄存器 13-2: **IOCBN**: 负边沿电平变化中断寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                         0 = 清零

bit 7-0            **IOCBN<7:0>**: 负边沿电平变化中断允许位  
1 = 允许引脚上的负边沿电平变化中断。检测到边沿时, 将相关状态位和中断标志位置 1。  
0 = 禁止相关引脚的电平变化中断。

寄存器 13-3: **IOCBF**: 电平变化中断标志寄存器

R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0	R/W/HS-0/0
IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                         0 = 清零                        HS = 由硬件清零

bit 7-0            **IOCBF<7:0>**: 电平变化中断标志位  
1 = 检测到相关引脚上发生允许的电平变化。  
    当IOCBPx = 1且检测到RBx引脚上有上升沿, 或IOCBNx = 1且检测到RBx引脚上有下降沿时置1。  
0 = 未检测到电平变化或用户清除了检测到的电平变化。

**表 13-1: 与电平变化中断相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	<a href="#">126</a>
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	<a href="#">86</a>
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0	<a href="#">134</a>
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0	<a href="#">134</a>
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0	<a href="#">134</a>
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	<a href="#">125</a>

**图注:** — = 未实现单元，读为 0。电平变化中断不使用阴影单元。

# PIC16(L)F193X

---

注:



## 14.0 固定参考电压 (FVR)

固定参考电压或 FVR 是稳定的参考电压，独立于 VDD，可选择 1.024V、2.048V 或 4.096V 输出级别。可配置 FVR 的输出来为以下各项提供参考电压：

- ADC 输入通道
- ADC 正参考电压
- 比较器正输入
- 数模转换器 (Digital-to-Analog Converter, DAC)
- 电容触摸传感 (Capacitive Sensing, CPS) 模块
- LCD 偏置电压发生器

通过将 FVRCON 寄存器的 FVREN 位置 1 可启用 FVR。

### 14.1 独立的增益放大器

FVR 输出可通过 2 个独立的可编程增益放大器连接到 ADC、比较器、DAC 和 CPS。每个放大器可配置为将

参考电压放大 1 倍、2 倍或 4 倍，从而提供三种可能的电压级别。

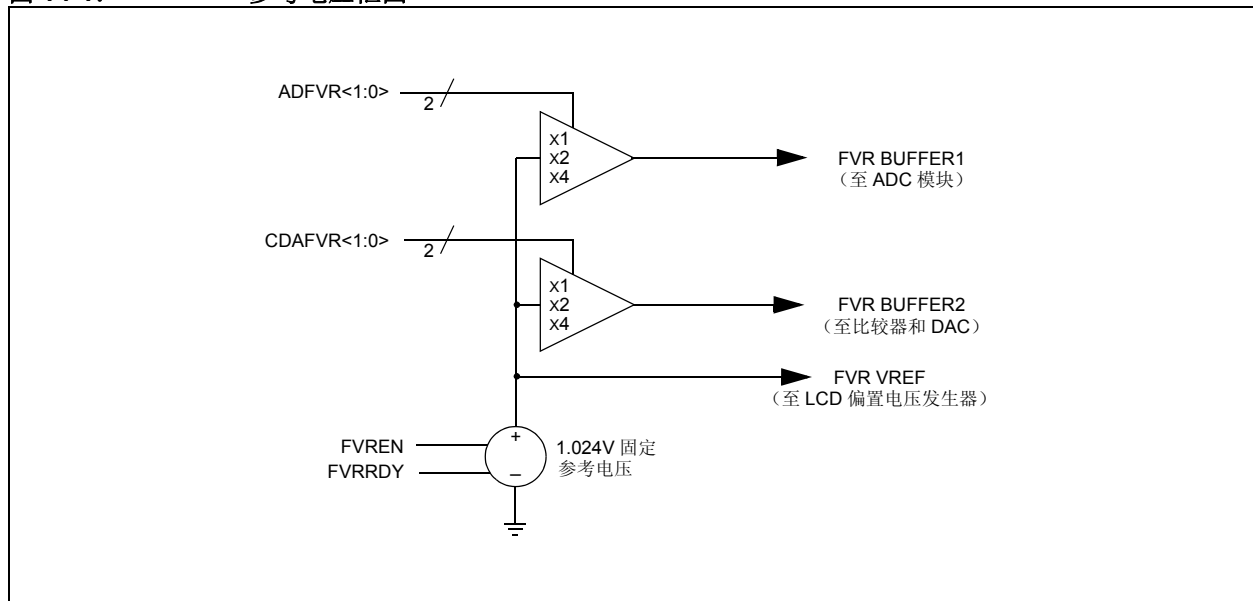
可使用 FVRCON 寄存器的 ADFVR<1:0> 位来为提供给 ADC 模块的参考电压使能和配置增益放大器设置。其他信息，请参见第 15.0 节“模数转换器 (ADC) 模块”。

可使用 FVRCON 寄存器的 CDAFVR<1:0> 位来为提供给 DAC、CPS 和比较器模块的参考电压使能和配置增益放大器设置。其他信息，请参见第 17.0 节“数模转换器 (DAC) 模块”、第 18.0 节“比较器模块”和第 26.0 节“电容触摸传感 (CPS) 模块”。

### 14.2 FVR 稳定时间

当使能固定参考电压模块时，参考电路和放大器电路需要一定的时间才能稳定。一旦电路稳定下来且就绪时，FVRCON 寄存器的 FVRRDY 位将置 1。请参见应用的电气规范章节中的最小延时要求。

图 14-1: 参考电压框图



# PIC16(L)F1933

## 14.3 FVR 控制寄存器

寄存器 14-1: **FVRCON: 固定参考电压控制寄存器**

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY <sup>(1)</sup>	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>	
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	q = 取值视条件而定

- bit 7      **FVREN:** 固定参考电压使能位  
 0 = 禁止固定参考电压  
 1 = 使能固定参考电压
- bit 6      **FVRRDY:** 固定参考电压就绪标志位 <sup>(1)</sup>  
 0 = 固定参考电压输出未就绪或未使能  
 1 = 固定参考电压输出已就绪
- bit 5      **TSEN:** 温度指示器使能位 <sup>(3)</sup>  
 0 = 禁止温度指示器  
 1 = 使能温度指示器
- bit 4      **TSRNG:** 温度指示器范围选择位 <sup>(3)</sup>  
 0 =  $V_{OUT} = V_{DD} - 2V_T$  (低电压范围)  
 1 =  $V_{OUT} = V_{DD} - 4V_T$  (高电压范围)
- bit 3-2    **CDAFVR<1:0>:** 比较器和 DAC 固定参考电压选择位  
 00 = 比较器和 DAC 固定参考电压外设输出关闭  
 01 = 比较器和 DAC 固定参考电压外设输出为 1x (1.024V)  
 10 = 比较器和 DAC 固定参考电压外设输出为 2x (2.048V) <sup>(2)</sup>  
 11 = 比较器和 DAC 固定参考电压外设输出为 4x (4.096V) <sup>(2)</sup>
- bit 1-0    **ADFVR<1:0>:** ADC 固定参考电压选择位  
 00 = ADC 固定参考电压外设输出关闭  
 01 = ADC 固定参考电压外设输出为 1x (1.024V)  
 10 = ADC 固定参考电压外设输出为 2x (2.048V) <sup>(2)</sup>  
 11 = ADC 固定参考电压外设输出为 4x (4.096V) <sup>(2)</sup>

- 注 1: 带 LDO 的器件 (PIC16F1933) 上的 FVRRDY 始终为 1。  
 2: 固定参考电压输出不能超过  $V_{DD}$ 。  
 3: 其他信息, 请参见第 16.0 节“温度指示器模块”。

表 14-1: 与固定参考电压相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		138

图注: 固定参考电压不使用阴影单元。

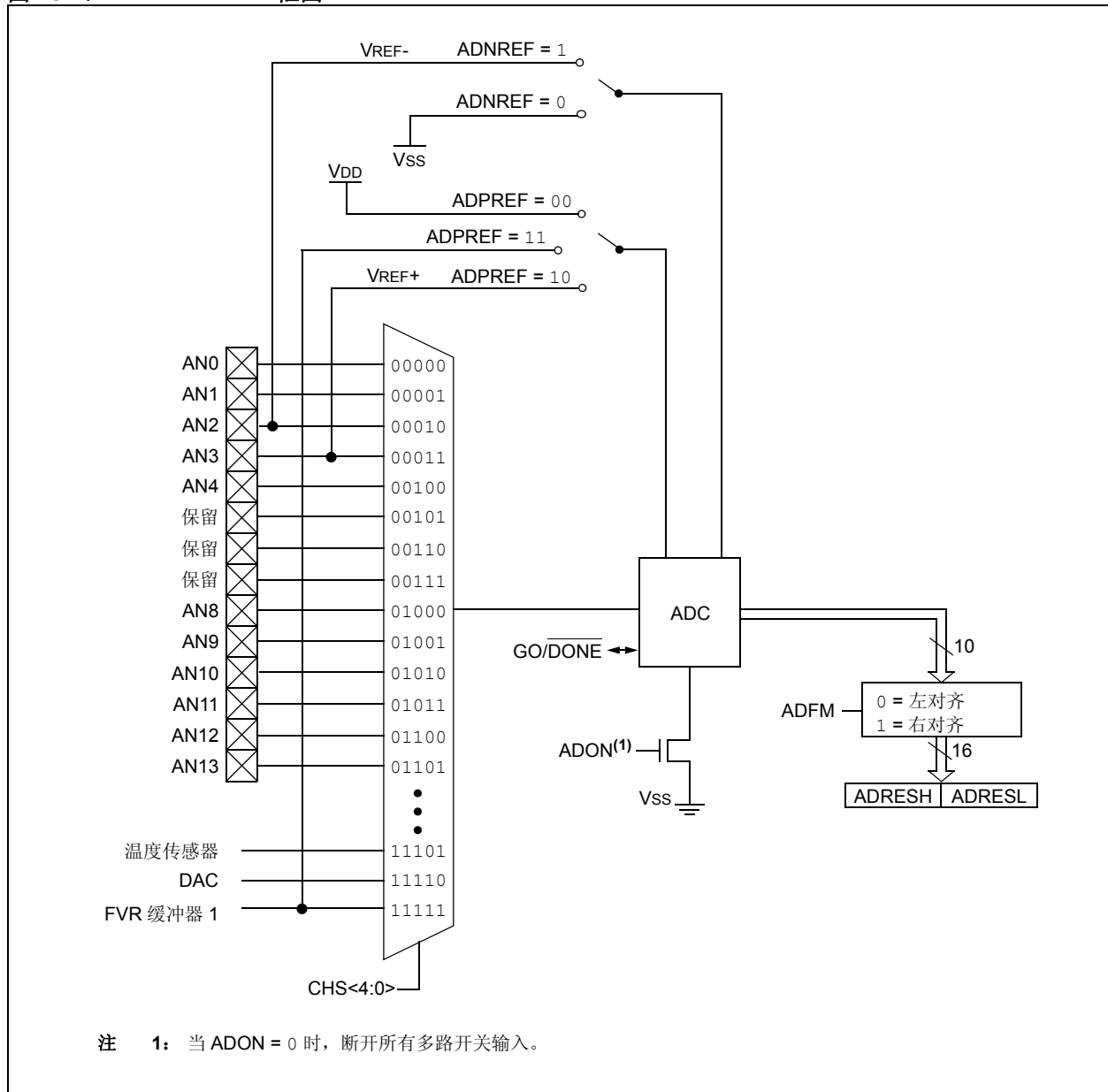
## 15.0 模数转换器 (ADC) 模块

模数转换器 (Analog-to-Digital Converter, ADC) 可以将模拟输入信号转换为该信号的 10 位二进制表示。该器件使用模拟输入, 这些输入通过多路开关连接到一个采样保持电路。采样保持电路的输出与转换器的输入相连。该转换器通过逐次逼近法生成 10 位二进制数并将转换结果存储到 ADC 结果寄存器 (ADRESH:ADRESL 寄存器对)。图 15-1 给出了 ADC 框图。

ADC 参考电压可通过软件选择为内部产生的参考电压或由外部提供。

ADC 可在转换完成时产生中断。此中断可用于将器件从休眠状态中唤醒。

图 15-1: ADC 框图



# PIC16(L)F1933

## 15.1 ADC 配置

在配置和使用 ADC 时，必须考虑以下功能：

- 端口配置
- 通道选择
- ADC 参考电压选择
- ADC 转换时钟源
- 中断控制
- 结果格式

### 15.1.1 端口配置

ADC 可用于转换模拟信号和数字信号。转换模拟信号时，应通过设置相关 TRIS 和 ANSEL 位来将 I/O 引脚配置为模拟引脚。更多信息，请参见第 12.0 节“[I/O 端口](#)”。

**注：** 对定义为数字输入的任何引脚施加模拟电压可能导致输入缓冲器出现过电流状态。

### 15.1.2 通道选择

提供了 14 个通道选择：

- AN<13:8, 4:0> 引脚
- 温度指示器
- DAC 输出
- FVR（固定参考电压）输出

更多关于这些通道选择的信息，请参见第 16.0 节“[温度指示器模块](#)”、第 17.0 节“[数模转换器（DAC）模块](#)”和第 14.0 节“[固定参考电压（FVR）](#)”。

ADCON0 寄存器的 CHS 位控制哪一路通道将与采样保持电路相连。

更换通道时需要一段时间的延时才能启动下一次转换。更多信息，请参见第 15.2 节“[ADC 工作原理](#)”。

### 15.1.3 ADC 参考电压

ADCON1 寄存器的 ADPREF 位控制正参考电压。正参考电压可以是：

- VREF+ 引脚
- VDD
- FVR 2.048V
- FVR 4.096V（LF 器件上未提供）

ADCON1 寄存器的 ADNREF 位控制负参考电压。负参考电压可以是：

- VREF- 引脚
- VSS

更多关于固定参考电压的信息，请参见第 14.0 节“[固定参考电压（FVR）](#)”。

### 15.1.4 转换时钟

转换时钟源可使用软件通过 ADCON1 寄存器的 ADCS 位进行选择。有以下 7 种时钟频率可供选择：

- FOSC/2
- FOSC/4
- FOSC/8
- FOSC/16
- FOSC/32
- FOSC/64
- FRC（专用内部振荡器）

转换 1 位所花的时间定义为 TAD。一次完整的 10 位转换需要 11.5 个 TAD 周期，如图 15-2 所示。

要实现正确的转换，必须满足适当的 TAD 规范。更多信息，请参见应用的电气规范章节中的 A/D 转换要求。表 15-1 给出了一个适当的 ADC 时钟选择的示例。

**注：** 除非使用 FRC，否则系统时钟频率的任何变化都会改变 ADC 时钟频率，从而对 ADC 结果产生不利影响。

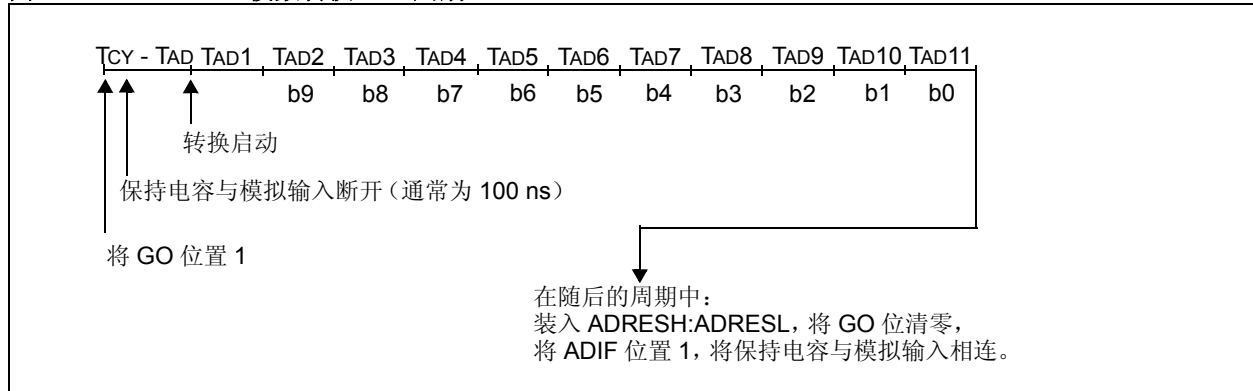
**表 15-1: ADC 时钟周期 (TAD) 与器件工作频率的关系**

ADC 时钟周期 (TAD)		器件频率 (Fosc)					
ADC 时钟源	ADCS<2:0>	32 MHz	20 MHz	16 MHz	8 MHz	4 MHz	1 MHz
Fosc/2	000	62.5ns <sup>(2)</sup>	100 ns <sup>(2)</sup>	125 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	2.0 μs
Fosc/4	100	125 ns <sup>(2)</sup>	200 ns <sup>(2)</sup>	250 ns <sup>(2)</sup>	500 ns <sup>(2)</sup>	1.0 μs	4.0 μs
Fosc/8	001	0.5 μs <sup>(2)</sup>	400 ns <sup>(2)</sup>	0.5 μs <sup>(2)</sup>	1.0 μs	2.0 μs	8.0 μs <sup>(3)</sup>
Fosc/16	101	800 ns	800 ns	1.0 μs	2.0 μs	4.0 μs	16.0 μs <sup>(3)</sup>
Fosc/32	010	1.0 μs	1.6 μs	2.0 μs	4.0 μs	8.0 μs <sup>(3)</sup>	32.0 μs <sup>(3)</sup>
Fosc/64	110	2.0 μs	3.2 μs	4.0 μs	8.0 μs <sup>(3)</sup>	16.0 μs <sup>(3)</sup>	64.0 μs <sup>(3)</sup>
FRC	x11	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>	1.0-6.0 μs <sup>(1,4)</sup>

图注: 阴影单元超出了建议范围。

- 注
- 1: 对于 VDD, FRC 源提供 1.6 μs 的典型 TAD 时间。
  - 2: 这些值均违反了最小 TAD 时间要求。
  - 3: 为了加快转换速度, 建议选用其他时钟源。
  - 4: 当 ADC 时钟由系统时钟 Fosc 提供时, 可以将 ADC 时钟周期 (TAD) 和总的 ADC 转换时间降至最低。但是, 如果在器件处于休眠模式下执行转换, 则必须使用 FRC 时钟源。

**图 15-2: 模数转换 TAD 周期**



# PIC16(L)F1933

## 15.1.5 中断

ADC模块具有在模数转换完成时产生中断的功能。ADC中断标志位是PIR1寄存器中的ADIF位。ADC中断允许位是PIE1寄存器中的ADIE位。必须用软件将ADIF位清零。

- 注 1:** 无论是否允许ADC中断，每次转换完成时都会将ADIF位置1。
- 2:** 仅当在选择了FRC振荡器时，ADC才能在休眠期间工作。

器件在工作期间或者休眠模式下都可以产生此中断。如果器件处于休眠模式，该中断可唤醒器件。从休眠模式唤醒时，总是执行SLEEP指令后的下一条指令。如果用户试图将器件从休眠模式下唤醒并继续执行主程序代码，则必须禁止INTCON寄存器的GIE和PEIE位。如果INTCON寄存器的GIE和PEIE位被使能，则转为执行中断服务程序。

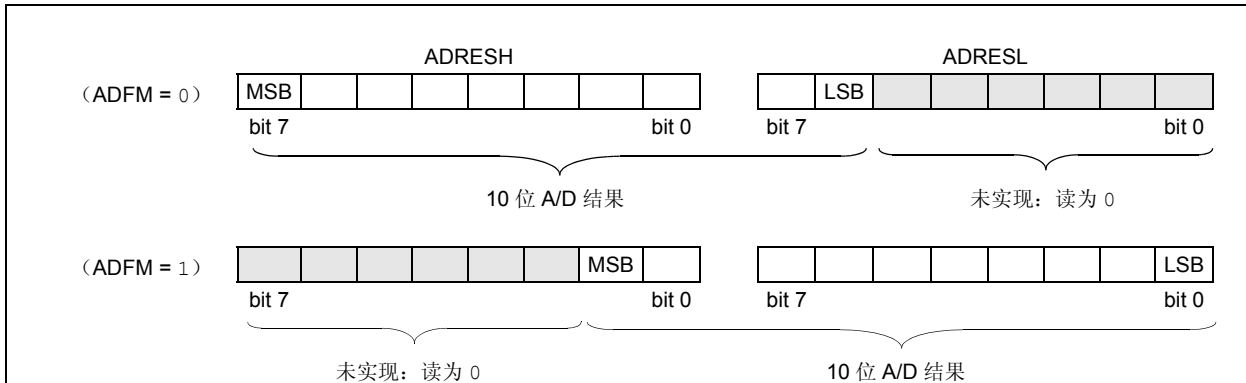
更多信息，请参见第15.1.5节“中断”。

## 15.1.6 结果格式

10位A/D转换的结果可采用两种格式：左对齐或右对齐。ADCON1寄存器的ADFM位控制输出格式。

图15-3给出了这两种输出格式。

图 15-3: 10 位 A/D 转换结果的格式



## 15.2 ADC 工作原理

### 15.2.1 启动转换

要使能 ADC 模块，必须将 ADCON0 寄存器的 ADON 位设置为 1。将 ADCON0 寄存器的 GO/DONE 位设置为 1 可启动模数转换。

**注：** 不应使用启动 ADC 的同一指令将 GO/DONE 位置 1。请参见第 15.2.6 节“A/D 转换步骤”。

### 15.2.2 完成转换

当转换完成时，ADC 模块将：

- 清零 GO/DONE 位
- 将 ADIF 中断标志位置 1
- 使用新的转换结果更新 ADRESH 和 ADRESL 寄存器

### 15.2.3 终止转换

如果必须要在转换完成前终止转换，则可用软件清零 GO/DONE 位。ADRESH 和 ADRESL 寄存器将使用部分完成的模数转换结果进行更新。未完成位将用最后转换的一位填充。

**注：** 器件复位强制所有寄存器进入复位状态。所以，会关闭 ADC 模块并终止任何待处理的转换。

### 15.2.4 休眠期间的 ADC 工作原理

ADC 模块可以工作在休眠模式下。这需要将 ADC 时钟源设置为 FRC 选项。当选择 FRC 时钟源时，ADC 会多等待一个指令周期再开始转换。这允许执行一条 SLEEP 指令，从而降低转换期间的系统噪声。如果允许了 ADC 中断，则转换完成时，器件将从休眠模式下唤醒。如果禁止了 ADC 中断，尽管 ADON 位仍保持置 1，但是在转换完成后 ADC 模块会关闭。

当 ADC 时钟源为 FRC 之外的其他时钟源时，尽管 ADON 位仍保持置 1，但是 SLEEP 指令会导致当前转换中止且 ADC 模块关闭。

### 15.2.5 特殊事件触发器

CCPx/ECCPx 模块的特殊事件触发器允许在没有软件介入的情况下，定期进行 ADC 转换。发生触发信号时，GO/DONE 位由硬件置 1，Timer1 计数器复位为零。

**表 15-2: 特殊事件触发器**

器件	CCPx/ECCPx
PIC16(L)F1933	CCP5

使用特殊事件触发器不能确保正确的 ADC 时序。用户有责任确保满足 ADC 时序要求。

更多信息，请参见第 23.0 节“捕捉/比较/PWM 模块”。

# PIC16(L)F1933

## 15.2.6 A/D 转换步骤

这是使用 ADC 执行模数转换的示例步骤：

- 配置端口：
  - 禁用引脚输出驱动器（见 TRIS 寄存器）
  - 将引脚配置为模拟引脚（见 ANSEL 寄存器）
- 配置 ADC 模块：
  - 选择 ADC 转换时钟
  - 配置参考电压
  - 选择 ADC 输入通道
  - 启动 ADC 模块
- 配置 ADC 中断（可选）：
  - 清零 ADC 中断标志
  - 允许 ADC 中断
  - 允许外设中断
  - 允许全局中断<sup>(1)</sup>
- 等待所需的采集时间<sup>(2)</sup>。
- 通过将  $\overline{GO/DONE}$  位置 1 来启动转换。
- 通过以下任一方式等待 ADC 转换完成：
  - 查询  $\overline{GO/DONE}$  位
  - 等待 ADC 中断（中断被允许）
- 读 ADC 结果。
- 清零 ADC 中断标志（如果中断被允许的话，必需进行此操作）。

- 注**
- 1: 如果用户尝试将器件从休眠模式下唤醒并继续执行主程序代码，则可以禁止全局中断。
  - 2: 请参见第 15.3 节“A/D 采集要求”。

## 例 15-1: A/D 转换

```
;This code block configures the ADC
;for polling, Vdd and Vss references, Frc
;clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL    ADCON1        ;
MOVLW     B'11110000'    ;Right justify, Frc
                                ;clock
MOVWF     ADCON1        ;Vdd and Vss Vref
BANKSEL    TRISA         ;
BSF       TRISA,0       ;Set RA0 to input
BANKSEL    ANSEL        ;
BSF       ANSEL,0       ;Set RA0 to analog
BANKSEL    ADCON0       ;
MOVLW     B'00000001'    ;Select channel AN0
MOVWF     ADCON0        ;Turn ADC On
CALL     SampleTime     ;Acquisition delay
BSF      ADCON0,ADGO    ;Start conversion
BTFSC    ADCON0,ADGO    ;Is conversion done?
GOTO     $-1            ;No, test again
BANKSEL    ADRESH       ;
MOVF     ADRESH,W       ;Read upper 2 bits
MOVWF    RESULTHI      ;store in GPR space
BANKSEL    ADRESL       ;
MOVF     ADRESL,W       ;Read lower 8 bits
MOVWF    RESULTLO      ;Store in GPR space
```



## 15.2.7 ADC 寄存器定义

下列寄存器用于控制 ADC 的工作。

### 寄存器 15-1: ADCON0: A/D 控制寄存器 0

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	CHS<4:0>					GO/DONE	ADON
bit 7							bit 0

#### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7      **未实现:** 读为 0

bit 6-2    **CHS<4:0>:** 模拟通道选择位

00000 = AN0

00001 = AN1

00010 = AN2

00011 = AN3

00100 = AN4

00101 = 保留。未连接通道。

00110 = 保留。未连接通道。

00111 = 保留。未连接通道。

01000 = AN8

01001 = AN9

01010 = AN10

01011 = AN11

01100 = AN12

01101 = AN13

01110 = 保留。未连接通道。

·

·

·

11100 = 保留。未连接通道。

11101 = 温度指示器<sup>(3)</sup>

11110 = DAC 输出<sup>(1)</sup>

11111 = FVR (固定参考电压) 缓冲器 1 输出<sup>(2)</sup>

bit 1      **GO/DONE:** A/D 转换状态位

1 = A/D 转换正在进行。将该位置 1 可启动 A/D 转换周期。

当 A/D 转换完成以后, 该位由硬件自动清零。

0 = A/D 转换完成 / 未进行

bit 0      **ADON:** ADC 使能位

1 = ADC 被使能

0 = ADC 被禁止且不消耗工作电流

注 1: 更多信息, 请参见第 17.0 节“数模转换器 (DAC) 模块”。

2: 更多信息, 请参见第 14.0 节“固定参考电压 (FVR)”。

3: 更多信息, 请参见第 16.0 节“温度指示器模块”。

# PIC16(L)F1933

寄存器 15-2: **ADCON1: A/D 控制寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>	
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

**ADFM:** A/D 结果格式选择位

1 = 右对齐。装入转换结果时, ADRESH 的高 6 位被设置为 0。

0 = 左对齐。装入转换结果时, ADRESL 的低 6 位被设置为 0。

bit 6-4

**ADCS<2:0>:** A/D 转换时钟选择位

000 = FOSC/2

001 = FOSC/8

010 = FOSC/32

011 = FRC (由专用 RC 振荡器提供时钟)

100 = FOSC/4

101 = FOSC/16

110 = FOSC/64

111 = FRC (由专用 RC 振荡器提供时钟)

bit 3

**未实现:** 读为 0

bit 2

**ADNREF:** A/D 负参考电压配置位

0 = VREF- 与 VSS 连接

1 = VREF- 与外部 VREF- 引脚连接 <sup>(1)</sup>

bit 1-0

**ADPREF<1:0>:** A/D 正参考电压配置位

00 = VREF+ 与 VDD 连接

01 = 保留

10 = VREF+ 与外部 VREF+ 引脚连接 <sup>(1)</sup>

11 = VREF+ 与内部固定参考电压 (FVR) 模块连接 <sup>(1)</sup>

**注 1:** 选择 FVR 或 VREF+ 引脚作为正参考电压源时, 注意存在一个最小电压规范值。详情请参见应用的电气规范章节。

**寄存器 15-3:      ADRESH: ADC 结果寄存器的高字节 (ADRESH), ADFM = 0**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<9:2>							
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-0      **ADRES<9:2>**: ADC 结果寄存器位  
10 位转换结果的高 8 位

**寄存器 15-4:      ADRESL: ADC 结果寄存器的低字节 (ADRESL), ADFM = 0**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<1:0>		—	—	—	—	—	—
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7-6      **ADRES<1:0>**: ADC 结果寄存器位  
10 位转换结果的低 2 位

bit 5-0      **保留**: 不用。

# PIC16(L)F1933

寄存器 15-5: **ADRESH: ADC 结果寄存器的高字节 (ADRESH), ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
—	—	—	—	—	—	ADRES<9:8>	
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-2 保留: 不用。

bit 1-0 **ADRES<9:8>**: ADC 结果寄存器位  
10 位转换结果的高 2 位

寄存器 15-6: **ADRESL: ADC 结果寄存器的低字节 (ADRESL), ADFM = 1**

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
ADRES<7:0>							
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0 **ADRES<7:0>**: ADC 结果寄存器位  
10 位转换结果的低 8 位

## 15.3 A/D 采集要求

为了使 ADC 达到规定精度，必须使充电保持电容（CHOLD）充满至输入通道的电平。图 15-4 显示了模拟输入模型。信号源阻抗（Rs）和内部采样开关（Rss）阻抗直接影响给电容 CHOLD 充电所需要的时间。采样开关（Rss）阻抗随器件电压（VDD）不同而有所不同，请参见图 15-4。模拟信号源的最大阻抗推荐值为 10 kΩ。

采集时间随着信号源阻抗的减少而减少。选择（或改变）模拟输入通道之后，A/D 采集必须在转换开始之前完成。可以使用公式 15-1 计算最小采集时间。该公式假定误差为 1/2 LSB（即 ADC 的 1,024 步）。1/2 LSB 的误差是 ADC 达到规定分辨率所能允许的最大误差。

### 公式 15-1: 采集时间示例

假设：温度 = 50°C，且外部阻抗为 10 kΩ，5.0V VDD

$$\begin{aligned} T_{ACQ} &= \text{放大器稳定时间} + \text{保持电容充电时间} + \text{温度系数} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2 \mu\text{s} + T_c + [(\text{温度} - 25^\circ\text{C}) (0.05 \mu\text{s}/^\circ\text{C})] \end{aligned}$$

用以下公式可近似求得  $T_C$  的值：

$$V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) = V_{CHOLD} \quad ; [1] \text{ 在 } 1/2 \text{ lsb 误差范围内对 } V_{CHOLD} \text{ 充电}$$

$$V_{APPLIED} \left( 1 - e^{\frac{-T_C}{RC}} \right) = V_{CHOLD} \quad ; [2] \text{ 依照 } V_{APPLIED} \text{ 对 } V_{CHOLD} \text{ 充电}$$

$$V_{APPLIED} \left( 1 - e^{\frac{-T_C}{RC}} \right) = V_{APPLIED} \left( 1 - \frac{1}{(2^{n+1}) - 1} \right) \quad ; \text{结合 } [1] \text{ 和 } [2]$$

注：其中  $n$  = ADC 的位数。

对  $T_C$  求值：

$$\begin{aligned} T_C &= -CHOLD(RIC + R_{SS} + R_S) \ln(1/511) \\ &= -10 \text{ pF}(1 \text{ k}\Omega + 7 \text{ k}\Omega + 10 \text{ k}\Omega) \ln(0.001957) \\ &= 1.12 \mu\text{s} \end{aligned}$$

因此：

$$\begin{aligned} T_{ACQ} &= 2 \mu\text{s} + 1.12 \mu\text{s} + [(50^\circ\text{C} - 25^\circ\text{C})(0.05 \mu\text{s}/^\circ\text{C})] \\ &= 4.37 \mu\text{s} \end{aligned}$$

- 注**
- 1: 参考电压（VREF）自行抵消，因此它对该公式没有影响。
  - 2: 在每次转换后，充电保持电容（CHOLD）并不放电。
  - 3: 模拟信号源的最大推荐阻抗为 10 kΩ。它必须符合引脚泄漏电流规范。

# PIC16(L)F1933

图 15-4: 模拟输入模型

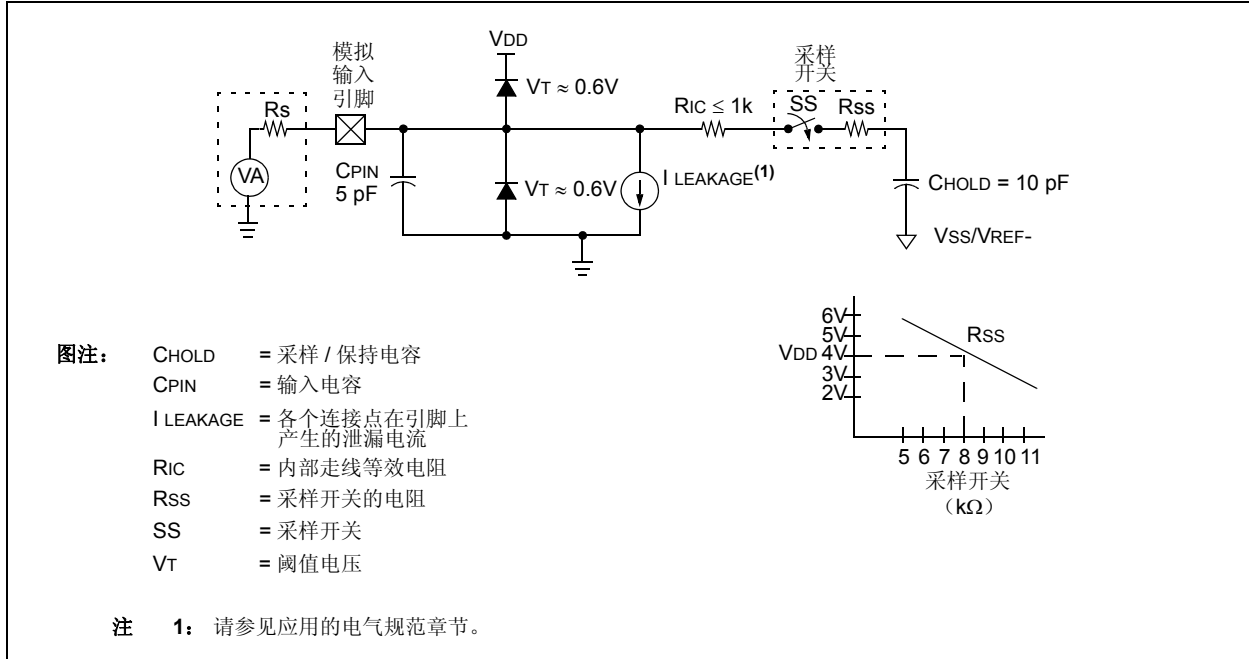
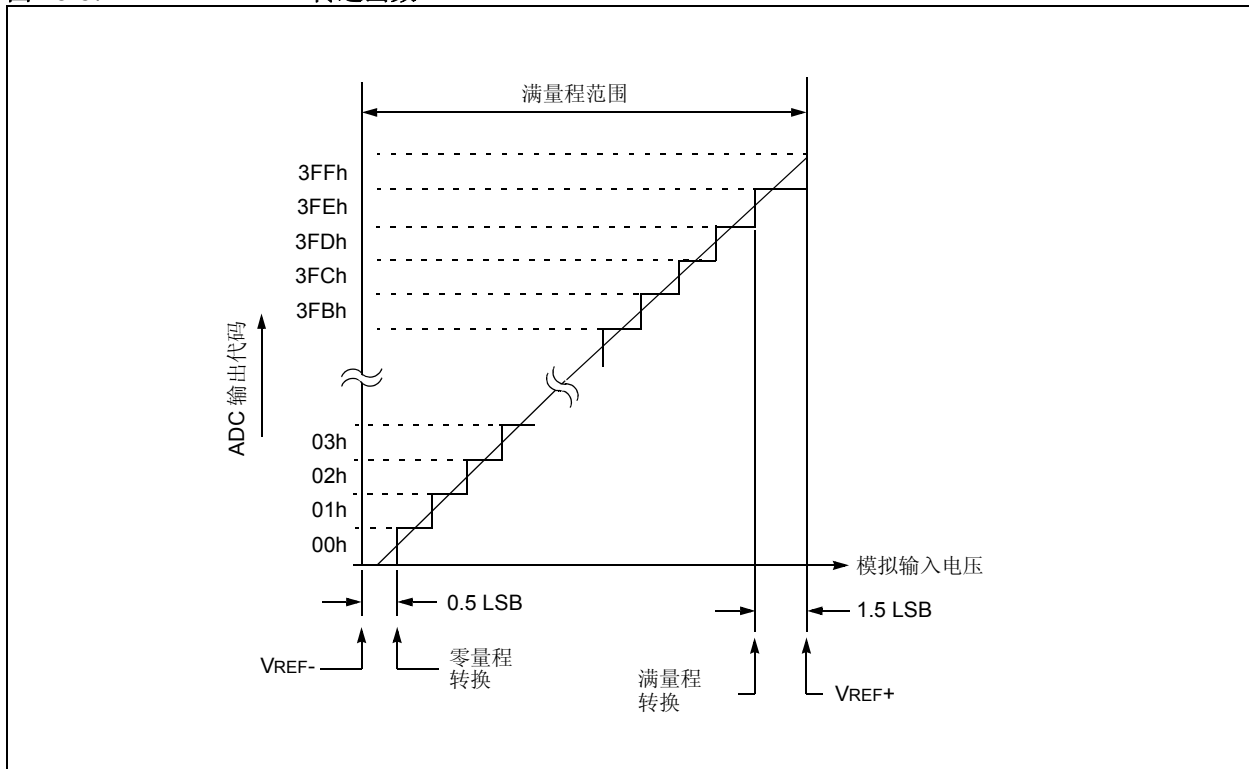


图 15-5: ADC 传递函数



**表 15-3: 与 ADC 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ADCON0	—	CHS<4:0>					GO/DONE	ADON	145
ADCON1	ADFM	ADCS<2:0>			—	ADNREF	ADPREF<1:0>		146
ADRESH	A/D 结果寄存器的高字节								147
ADRESL	A/D 结果寄存器的低字节								147
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	121
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	126
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				216
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		138
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	158
DACCON1	—	—	—	DACR<4:0>					158

**图注:** x = 未知, u = 不变, — = 未实现, 读为 0, q = 取值视条件而定。ADC 模块不使用阴影单元。

# PIC16(L)F1933

---

注:



## 16.0 温度指示器模块

本器件系列装配有一个温度电路，该电路旨在测量硅片的工作温度。电路的工作温度范围在  $-40^{\circ}\text{C}$  至  $+85^{\circ}\text{C}$  之间。输出电压与器件温度成比例。温度指示器的输出内部连接到器件 ADC。

该电路可用作温度阈值检测器或更精确的温度指示器，具体取决于执行的校准级别。单点校准使电路指示该点附近的温度。两点校准使电路能更精确地测量整个温度范围。有关校准过程的更多信息，请参见应用笔记《内部温度指示器的使用和校准》(DS01333A\_CN)。

### 16.1 电路工作原理

图 16-1 给出了该温度电路的简化框图。成比例的电压输出通过测量多个硅片接点上的正向压降实现。

公式 16-1 说明了温度指示器的输出特性。

#### 公式 16-1: $V_{\text{OUT}}$ 范围

高电压范围:  $V_{\text{OUT}} = V_{\text{DD}} - 4V_{\text{T}}$

低电压范围:  $V_{\text{OUT}} = V_{\text{DD}} - 2V_{\text{T}}$

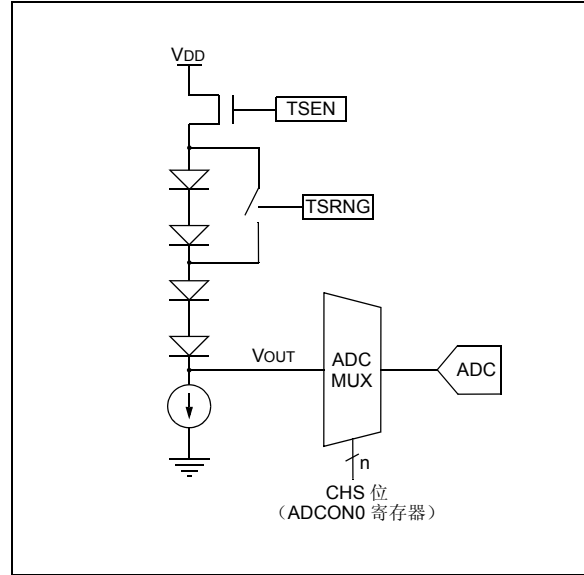
温度检测电路与固定参考电压 (FVR) 模块集成。更多信息，请参见第 14.0 节“固定参考电压 (FVR)”。

将 FVRCON 寄存器的 TSEN 位置 1 可启用该电路。禁止时，该电路不消耗电流。

该电路可在高或低电压范围下工作。高电压范围通过将 FVRCON 寄存器的 TSRNG 位置 1 选择，可提供更宽的输出电压。这为整个温度范围提供了更高的分辨率，但是各部件之间一致性较差。此范围需要更高的偏置电压才能工作，因此，需要更高的  $V_{\text{DD}}$ 。

低电压范围通过将 FVRCON 寄存器的 TSRNG 位清零来选择。低电压范围会产生较低的压降，因此，该电路需要较低的偏置电压才能工作。低电压范围为低电压工作而提供。

图 16-1: 温度电路图



### 16.2 最低工作电压 $V_{\text{DD}}$ 与最低检测温度

温度电路在低电压范围下工作时，器件可以在规范指定范围内的任何工作电压下工作。

温度电路在高电压范围下工作时，器件工作电压  $V_{\text{DD}}$  必须足够高才能确保温度电路正确偏置。

表 16-1 给出了建议的最低  $V_{\text{DD}}$  与范围设置关系。

表 16-1: 建议的  $V_{\text{DD}}$  与范围

最低 $V_{\text{DD}}$ , TSRNG = 1	最低 $V_{\text{DD}}$ , TSRNG = 0
3.6V	1.8V

### 16.3 温度输出

电路的输出使用内部模数转换器来测量。为温度电路输出预留了一个通道。详细信息，请参见第 15.0 节“模数转换器 (ADC) 模块”。

### 16.4 ADC 采集时间

为确保温度测量的准确性，用户在 ADC 输入多路开关连接到温度指示器输出后必须至少等待  $200\ \mu\text{s}$ ，才能执行转换。此外，用户必须在温度指示器输出的后续转换之间等待  $200\ \mu\text{s}$ 。

# PIC16(L)F1933

---

注:

## 17.0 数模转换器 (DAC) 模块

数模转换器可提供与输入源成比例的可变参考电压，共有 32 种输出选项可供使用。

DAC 的输入可以连接到：

- 外部 VREF 引脚
- VDD 电源电压
- FVR (固定参考电压)

可配置 DAC 的输出为以下各项提供参考电压：

- 比较器正输入
- ADC 输入通道
- DACOUT 引脚
- 电容触摸传感模块 (Capacitive Sensing Module, CSM)

可通过将 DACCON0 寄存器的 DACEN 位置 1 来使能数模转换器 (DAC)。

### 17.1 输出电压选择

DAC 有 32 种输出电压。这 32 种电压可使用 DACCON1 寄存器的 DACR<4:0> 位进行设置。

DAC 输出电压可通过以下公式确定：

#### 公式 17-1: DAC 输出电压

**如果  $DACEN = 1$**

$$V_{OUT} = \left( (V_{SOURCE+} - V_{SOURCE-}) \times \frac{DACR[4:0]}{2^5} \right) + V_{SOURCE-}$$

**如果  $DACEN = 0$  &  $DACLPS = 1$  &  $DACR[4:0] = 11111$**

$$V_{OUT} = V_{SOURCE+}$$

**如果  $DACEN = 0$  &  $DACLPS = 0$  &  $DACR[4:0] = 00000$**

$$V_{OUT} = V_{SOURCE-}$$

$$V_{SOURCE+} = V_{DD}, V_{REF} \text{ 或 } FVR \text{ BUFFER } 2$$

$$V_{SOURCE-} = V_{SS}$$

### 17.2 成比例输出电平

使用梯形电阻网络并将梯形电阻网络的每端与正负参考电压输入源连接，可以生成 DAC 输出值。如果任一输入源的电压波动，则会对 DAC 输出值造成相应的波动。

在应用的电气规范章节中可找到梯形电阻网络中各个电阻的值。

### 17.3 DAC 参考电压输出

通过将 DACCON0 寄存器的 DACOE 位设置为 1 可将 DAC 输出到 DACOUT 引脚。选择 DACOUT 引脚的 DAC 输出参考电压将自动改写该引脚的数字输出缓冲器和数字输入阈值检测器功能。在将 DACOUT 引脚配置为 DAC 参考电压输出后，对其执行读操作将始终返回 0。

由于电流驱动能力有限，因此必须在 DAC 参考电压输出引脚上使用缓冲器以将 DACOUT 供外部连接使用。图 17-2 举例说明了这一缓冲技术。

# PIC16(L)F1933

图 17-1: 数模转换器框图

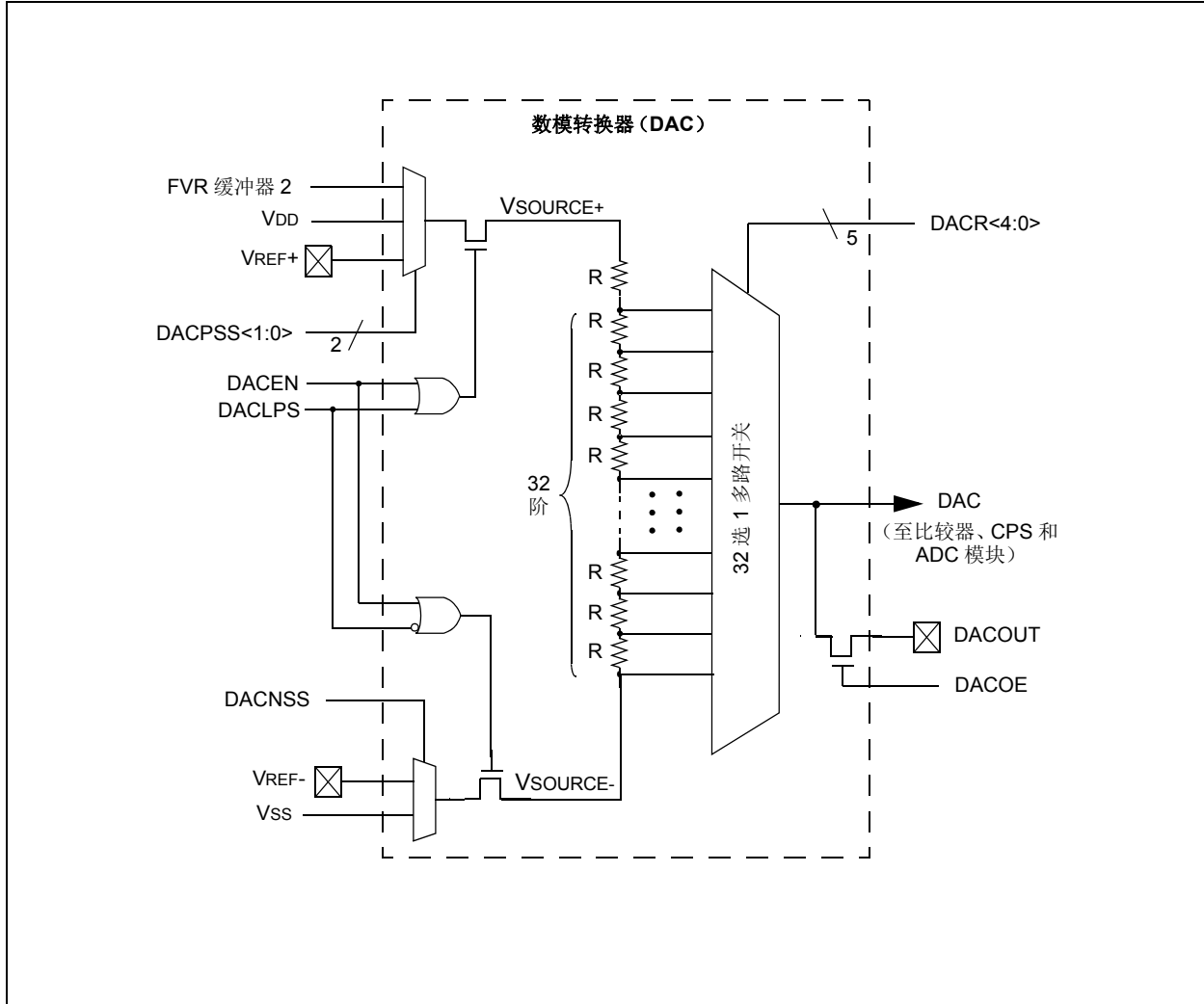
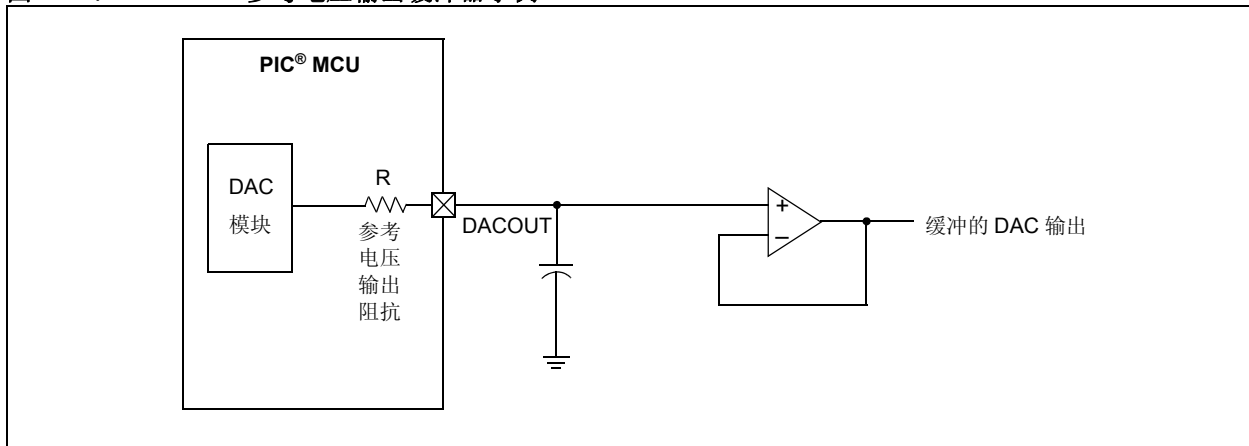


图 17-2: 参考电压输出缓冲器示例



## 17.4 低功耗电压状态

为了使DAC模块消耗的功耗最低，必须断开梯形电阻网络的两个参考电压输入源中的一个。可以禁止正电压源（VSOURCE+）或负电压源（VSOURCE-）。

将DACCON0寄存器的DACLPS位置1可以禁止负电压源。将DACCON0寄存器的DACLPS位清零可以禁止正电压源。

### 17.4.1 输出电压钳位到正电压源

执行以下操作可以将DAC输出电压设置为VSOURCE+且功耗最低：

- 将DACCON0寄存器的DACEN位清零。
- 将DACCON0寄存器的DACLPS位置1。
- 配置DACPSS位以设置合适的正电源。
- 将DACCON1寄存器中的DACR<4:0>位配置为11111。

这也是用来将电压电平从FVR输出到输出引脚的方法。更多信息，请参见第17.5节“休眠期间的工作”。

输出电压钳位示例，请参见图17-3。

### 17.4.2 输出电压钳位到负电压源

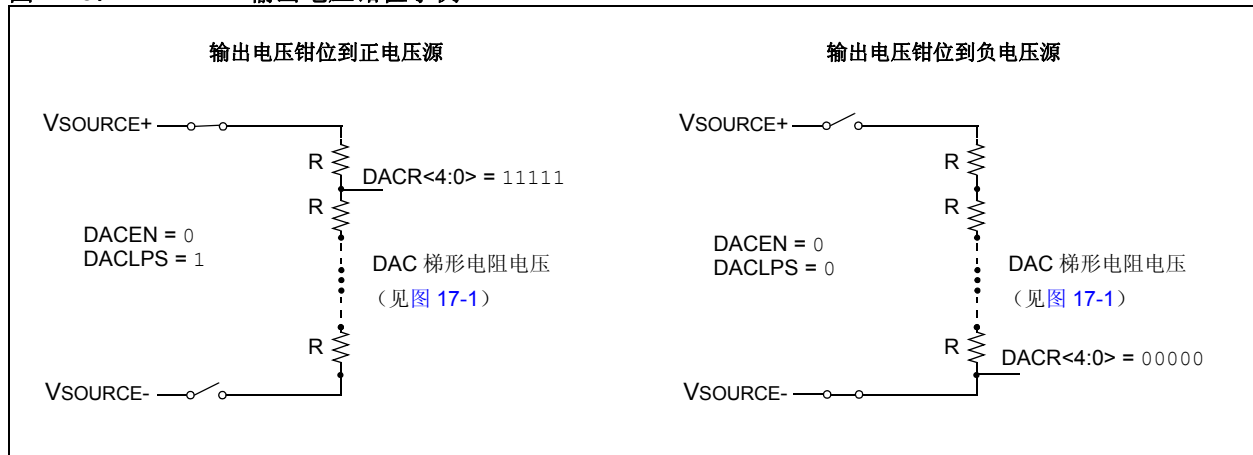
执行以下操作可以将DAC输出电压设置为VSOURCE-且功耗最低：

- 将DACCON0寄存器的DACEN位清零。
- 将DACCON0寄存器的DACLPS位清零。
- 配置DACNSS位以设置合适的负电源。
- 将DACCON1寄存器中的DACR<4:0>位配置为00000。

这使得比较器能够检测过零事件且不额外消耗流经DAC模块的电流。

输出电压钳位示例，请参见图17-3。

图 17-3: 输出电压钳位示例



## 17.5 休眠期间的工作

当器件通过中断或看门狗定时器超时从休眠模式唤醒时，DACCON0寄存器中的内容不受影响。要降低休眠模式下的电流消耗，应禁止参考电压。

## 17.6 复位的影响

器件复位影响以下各项：

- 禁止DAC。
- DAC输出电压断开与DACOUT引脚的连接。
- 将DACR<4:0>范围选择位清零。

# PIC16(L)F1933

**寄存器 17-1: DACCON0: 参考电压控制寄存器 0**

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0
DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
 1 = 置 1                         0 = 清零

bit 7            **DACEN:** DAC 使能位  
 1 = 使能 DAC  
 0 = 禁止 DAC

bit 6            **DACLPS:** DAC 低功耗电压状态选择位  
 1 = 选择 DAC 正参考电压源  
 0 = 选择 DAC 负参考电压源

bit 5            **DACOE:** DAC 电压输出使能位  
 1 = DAC 电平也从 DACOUT 引脚输出  
 0 = DAC 电平断开与 DACOUT 引脚的连接

bit 4            **未实现:** 读为 0

bit 3-2        **DACPSS<1:0>:** DAC 正电压源选择位  
 00 = VDD  
 01 = VREF+ 引脚  
 10 = FVR 缓冲器 2 输出  
 11 = 保留, 未使用

bit 1            **未实现:** 读为 0

bit 0            **DACNSS:** DAC 负电压源选择位  
 1 = VREF-  
 0 = VSS

**寄存器 17-2: DACCON1: 参考电压控制寄存器 1**

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	DACR<4:0>				
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
 u = 不变                        x = 未知                        -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
 1 = 置 1                         0 = 清零

bit 7-5        **未实现:** 读为 0

bit 4-0        **DACR<4:0>:** DAC 电压输出选择位

**表 17-1: 与 DAC 模块相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		138
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	158
DACCON1	—	—	—	DACR<4:0>					158

**图注:** — = 未实现单元, 读为 0。DAC 模块不使用阴影单元。

## 18.0 比较器模块

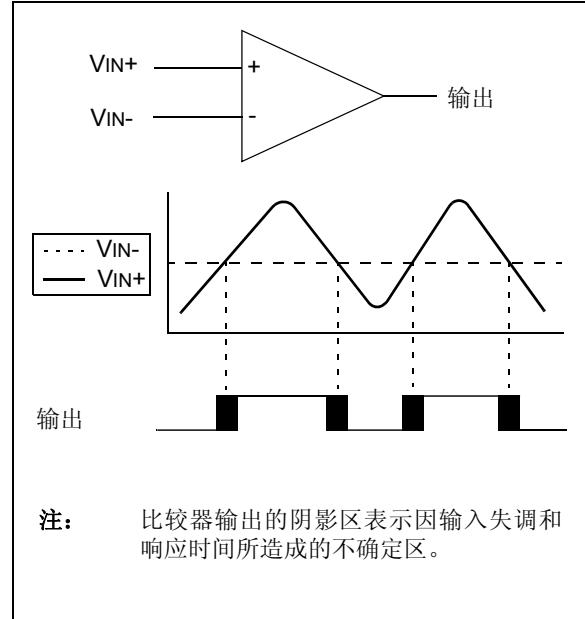
比较器用作模拟电路与数字电路的接口，通过比较两个模拟电压的大小并输出一个数字量以指示输入量的相对大小。比较器是非常有用的混合信号构建块，因为它提供了独立于程序执行的模拟功能。模拟比较器模块包含如下特性：

- 独立的比较器控制
- 可编程输入选择
- 可从内部 / 外部获取比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠模式唤醒
- 可编程速度 / 功耗优化
- PWM 关闭
- 可编程参考电压和固定参考电压

### 18.1 比较器概述

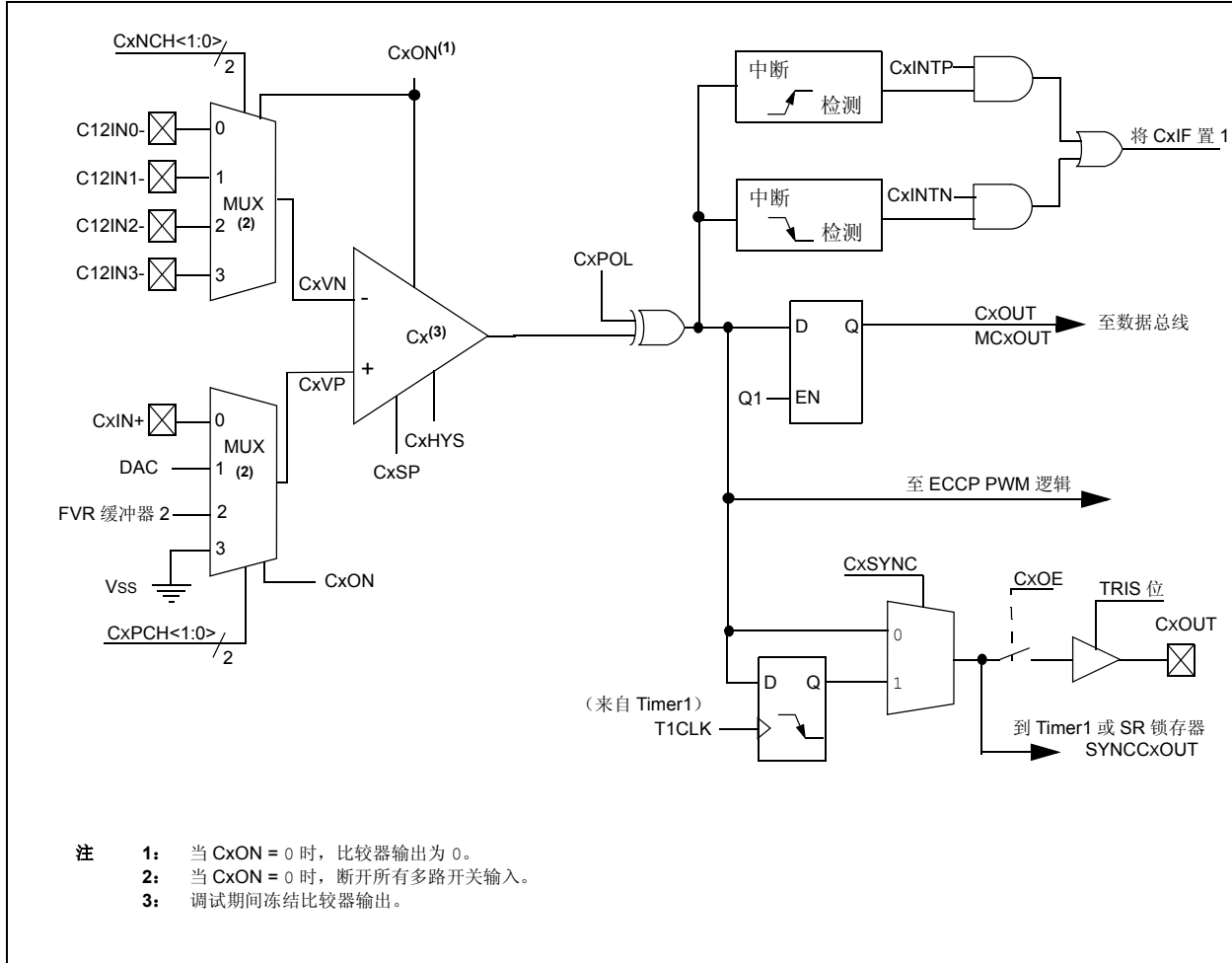
图18-1给出了单比较器及其模拟输入电平和数字输出之间的关系。如果  $V_{IN+}$  上的模拟电压低于  $V_{IN-}$  上的模拟电压时，比较器输出数字低电平。如果  $V_{IN+}$  上的模拟电压高于  $V_{IN-}$  上的模拟电压时，比较器输出数字高电平。

图 18-1: 单比较器



# PIC16(L)F1933

图 18-2: 比较器模块简化框图





## 18.2 比较器控制

每个比较器有 2 个控制寄存器：CMxCON0 和 CMxCON1。

CMxCON0 寄存器（见寄存器 18-1）包含具有以下功能的控制和状态位：

- 使能
- 输出选择
- 输出极性
- 速度 / 功耗选择
- 迟滞电压使能
- 输出同步

CMxCON1 寄存器（见寄存器 18-2）包含具有以下功能的控制位：

- 中断允许
- 中断边沿极性
- 正输入通道选择
- 负输入通道选择

### 18.2.1 比较器使能

将 CMxCON0 寄存器的 CxON 位置 1 可使能比较器工作。将 CxON 位清零可禁止比较器，从而使电流消耗降至最低。

### 18.2.2 比较器输出选择

可通过读 CMxCON0 寄存器的 CxOUT 位或 CMOUT 寄存器的 MCxOUT 位来监视比较器输出。为了使输出对外部连接可用，以下条件必须为真：

- CMxCON0 寄存器的 CxOE 位必须置 1
- 相应 TRIS 位必须清零
- CMxCON0 寄存器的 CxON 位必须置 1

**注 1:** CMxCON0 寄存器的 CxOE 位可改写端口数据锁存器。将 CMxCON0 寄存器的 CxON 置 1 不会影响端口改写。

**2:** 比较器的内部输出在每个指令周期都被锁存。除非另外指定，否则不锁存外部输出。

### 18.2.3 比较器输出极性

翻转比较器输出在功能上等同于交换比较器的两个输入端。可通过将 CMxCON0 寄存器的 CxPOL 位置 1 来翻转比较器输出的极性。清零 CxPOL 位时输出不反相。

表 18-1 给出了输出状态与输入条件（包括极性控制）的关系。

**表 18-1: 比较器输出状态与输入条件的关系**

输入条件	CxPOL	CxOUT
$CxVN > CxVP$	0	0
$CxVN < CxVP$	0	1
$CxVN > CxVP$	1	1
$CxVN < CxVP$	1	0

### 18.2.4 比较器速度 / 功耗选择

在程序执行期间，可使用 CxSP 控制位来最佳地权衡速度和功耗。CxSP 控制位的默认状态为 1，表示选择了正常速度模式。通过将 CxSP 位清为 0 可以优化器件的功耗，但代价是比较器传输延时变长。

## 18.3 比较器迟滞电压

可选择添加适量的跳变电压到每个比较器的输入引脚上，以使比较器具有迟滞功能。可通过将 CMxCON0 寄存器的 CxHYS 位置 1 来使能迟滞功能。

更多信息，请参见应用的电气规范章节。

## 18.4 Timer1 门控的工作原理

比较器的输出可用作 Timer1 的门控源。更多信息，请参见第 21.6 节“Timer1 门控”。此功能可用于对模拟事件的持续时间或间隔计时。

建议比较器输出与 Timer1 同步。这可确保比较器发生变化时 Timer1 不会递增计数。

### 18.4.1 比较器输出同步

可通过将 CMxCON0 寄存器的 CxSYNC 位置 1 来将任一比较器 C1 或 C2 的输出与 Timer1 同步。

当使能时，比较器的输出在 Timer1 时钟源的下降沿被锁存。如果 Timer1 使用预分频器，比较器的输出将在执行预分频操作后被锁存。为防止时钟冲突，比较器输出在 Timer1 时钟源的下降沿被锁存，而 Timer1 在其时钟源的上升沿递增计数。更多信息，请参见比较器框图（图 18-2）和 Timer1 框图（图 22-1）。

## 18.5 比较器中断

比较器输出值的变化会产生中断，这是因为每个比较器都有上升沿检测器和下降沿检测器。

当任一边沿检测器被触发且其相关中断允许位（CMxCON1 寄存器的 CxINTP 和 / 或 CxINTN 位）被置 1 时，相应中断标志位（PIR2 寄存器的 CxIF 位）也将置 1。

要允许中断，必须将以下各位置 1：

- CMxCON0 寄存器的 CxON、CxPOL 和 CxSP 位
- PIE2 寄存器的 CxIE 位
- CMxCON1 寄存器的 CxINTP 位（用于上升沿检测）
- CMxCON1 寄存器的 CxINTN 位（用于下降沿检测）
- INTCON 寄存器的 PEIE 和 GIE 位

PIR2 寄存器的相关中断标志位 CxIF 必须用软件清零。如果在该标志位清零期间又检测到一边沿，则在该序列结束时仍将标志位置 1。

**注：** 尽管比较器被禁止，但通过使用 CMxCON0 寄存器的 CxPOL 位更改比较器输出极性，或通过使用 CMxCON0 寄存器的 CxON 位打开或关闭比较器仍可产生中断。

## 18.6 比较器正输入选择

配置 CMxCON1 寄存器的 CxPCH<1:0> 位可将内部参考电压或模拟引脚连接到比较器的同相输入端：

- CxIN+ 模拟引脚
- DAC
- FVR（固定参考电压）
- Vss（地）

更多关于固定参考电压模块的信息，请参见第 14.0 节“固定参考电压（FVR）”。

更多关于 DAC 输入信号的信息，请参见第 17.0 节“数模转换器（DAC）模块”。

任何时候禁止比较器（CxON = 0）都将禁止所有比较器输入。

## 18.7 比较器负输入端选择

CMxCON0 寄存器的 CxNCH<1:0> 位可将 4 个模拟引脚中的 1 个引脚连接到比较器反相输入端。

**注：** 要将 CxIN+ 和 CxINx- 引脚用作模拟输入引脚，则 ANSEL 寄存器中与该引脚相关的位必须置 1，且相应的 TRIS 位也必须置 1 以禁止输出驱动器。

## 18.8 比较器的响应时间

在更改输入源或选用新的参考电压后的一段时间内，比较器输出是不确定的。这段时间就称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入更改的总响应时间时，这两个时间都必须考虑在内。更多详细信息，请参见应用的电气规范章节中的比较器和参考电压规范。

## 18.9 与 ECCP 逻辑的相互作用

C1 和 C2 比较器都可用作通用比较器。其输出可连接到 C1OUT 和 C2OUT 引脚。当 ECCP 自动关闭有效时，它可使用这两个比较器信号中的一个或全部。如果还使能了自动重启，则比较器可配置为 ECCP 的闭环模拟反馈电路，从而产生模拟控制的 PWM 信号。

**注：** 首次初始化比较器模块时，其输出状态未知。初始化后，用户应在使用结果前对比较器的输出状态进行验证，主要在使用的结果与其他外设功能（如 ECCP 自动关闭模式）相关时。

## 18.10 模拟输入连接注意事项

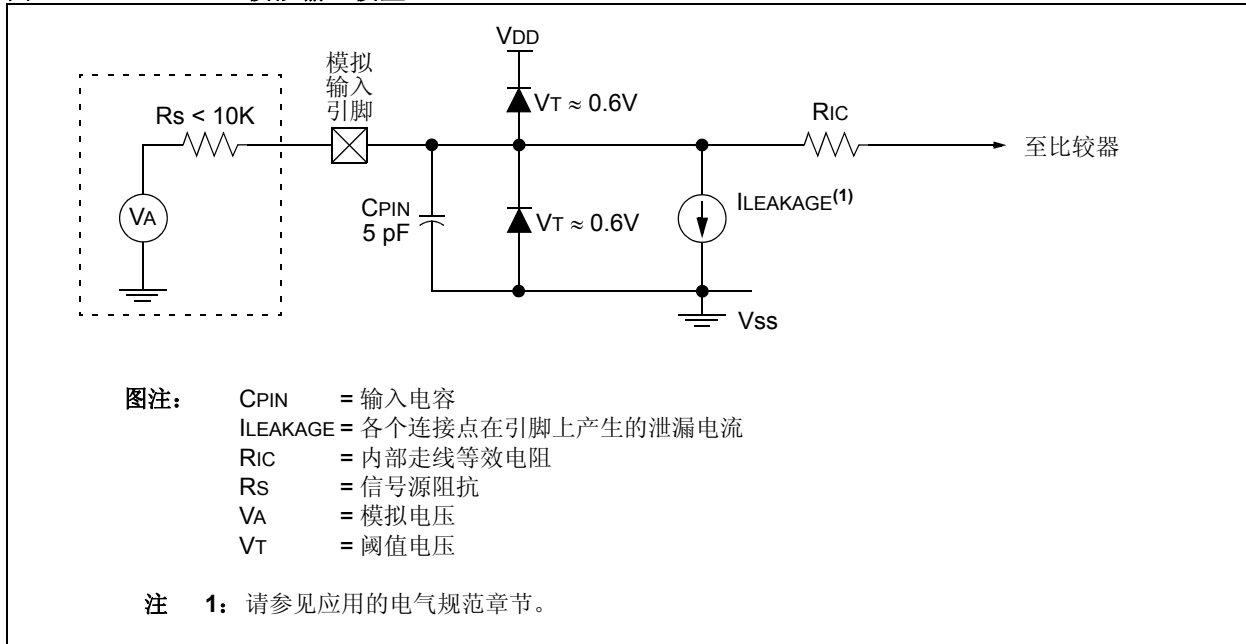
图 18-3 是一个简化的模拟输入电路。由于模拟输入引脚与数字输入端共用，因而它们与 VDD 和 VSS 之间加有反向偏置的 ESD 保护二极管，因此，模拟输入电压必须在 VSS 和 VDD 之间。一旦输入电压在任一方向上超出该范围 0.6V，其中一个二极管就会发生正向偏置，从而使输入电压被钳位。

模拟信号源的最大阻抗推荐值为 10 kΩ。连接到模拟输入引脚的任何外部元件（如电容或齐纳二极管），要保证其泄漏电流极小，从而使引入的误差最小。

- 注**
- 1: 读端口寄存器时，所有配置为模拟输入的引脚都将读为 0，而配置为数字输入的引脚将根据输入规范，对模拟输入信号进行相应的转换。
  - 2: 施加在数字输入引脚上的模拟电平会使其输入缓冲器消耗的电流超过规定值。

# PIC16(L)F1933

图 18-3: 模拟输入模型



## 寄存器 18-1: CMxCON0: 比较器 X 控制寄存器 0

R/W-0/0	R-0/0	R/W-0/0	R/W-0/0	U-0	R/W-1/1	R/W-0/0	R/W-0/0
CxON	CxOUT	CxOE	CxPOL	—	CxSP	CxHYS	CxSYNC
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7	<b>CxON:</b> 比较器使能位 1 = 比较器使能且不消耗有功功率 0 = 禁止比较器
bit 6	<b>CxOUT:</b> 比较器输出位 <u>如果 CxPOL = 1 (极性翻转):</u> 1 = CxVP < CxVN 0 = CxVP > CxVN <u>如果 CxPOL = 0 (极性不翻转):</u> 1 = CxVP > CxVN 0 = CxVP < CxVN
bit 5	<b>CxOE:</b> 比较器输出使能位 1 = CxOUT 通过 CxOUT 引脚输出。要求相关 TRIS 位清零以实际驱动该引脚。不受 CxON 位的影响。 0 = CxOUT 仅供内部使用
bit 4	<b>CxPOL:</b> 比较器输出极性选择位 1 = 比较器输出反相 0 = 比较器输出同相
bit 3	<b>未实现:</b> 读为 0
bit 2	<b>CxSP:</b> 比较器速度 / 功耗选择位 1 = 比较器工作在正常功耗、高速模式下 0 = 比较器工作在低功耗、低速模式下
bit 1	<b>CxHYS:</b> 比较器迟滞电压使能位 1 = 使能比较器迟滞电压 0 = 禁止比较器迟滞电压
bit 0	<b>CxSYNC:</b> 比较器输出同步模式位 1 = 比较器输出到 Timer1 且 I/O 引脚与 Timer1 时钟源上的变化同步。在 Timer1 时钟源的下降沿更新输出。 0 = 比较器输出到 Timer1 且 I/O 引脚异步。

# PIC16(L)F1933

**寄存器 18-2: CMxCON1: 比较器 CX 控制寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0
CxINTP	CxINTN	CxPCH<1:0>		—	—	CxNCH<1:0>	
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **CxINTP:** 比较器正边沿中断允许位  
 1 = CxIF 中断标志位在出现 CxOUT 位的正边沿时置 1  
 0 = 无中断标志位在出现 CxOUT 位的正边沿时置 1
- bit 6      **CxINTN:** 比较器负边沿中断允许位  
 1 = CxIF 中断标志位在出现 CxOUT 位的负边沿时置 1  
 0 = 无中断标志位在出现 CxOUT 位的负边沿时置 1
- bit 5-4    **CxPCH<1:0>:** 比较器正输入通道选择位  
 00 = CxVP 与 CxIN+ 引脚相连  
 01 = CxVP 与 DAC 参考电压相连  
 10 = CxVP 与 FVR 参考电压相连  
 11 = CxVP 与 Vss 相连
- bit 3-2    未实现: 读为 0
- bit 1-0    **CxNCH<1:0>:** 比较器负输入通道选择位  
 00 = CxVN 与 C12IN0- 引脚相连  
 01 = CxVN 与 C12IN1- 引脚相连  
 10 = CxVN 与 C12IN2- 引脚相连  
 11 = CxVN 与 C12IN3- 引脚相连

**寄存器 18-3: CMOUT: 比较器输出寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R-0/0	R-0/0
—	—	—	—	—	—	MC2OUT	MC1OUT
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7-2    未实现: 读为 0
- bit 1      **MC2OUT:** C2OUT 位的镜像副本
- bit 0      **MC1OUT:** C1OUT 位的镜像副本

**表 18-2: 与比较器模块相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
CM1CON0	C1ON	C1OUT	C1OE	C1POL	---	C1SP	C1HYS	C1SYNC	165
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2SP	C2HYS	C2SYNC	165
CM1CON1	C1NTP	C1INTN	C1PCH<1:0>		—	—	C1NCH<1:0>		166
CM2CON1	C2NTP	C2INTN	C2PCH<1:0>		—	—	C2NCH<1:0>		166
CMOUT	—	—	—	—	—	—	MC2OUT	MC1OUT	166
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFVR<1:0>		ADFVR<1:0>		138
DACCON0	DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS	158
DACCON1	—	—	—	DACR<4:0>					158
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	121
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	126

**图注:** — = 未实现单元，读为 0。比较器模块不使用阴影单元。

# PIC16(L)F1933

---

注:



## 19.0 SR 锁存器

本模块由一个 SR 锁存器组成，它具有多个置 1 和复位输入以及单独的锁存器输出。SR 锁存器模块包括以下功能：

- 可编程输入选择
- 可从外部获取 SR 锁存器输出
- 单独的 Q 和  $\bar{Q}$  输出
- 固件置 1 和复位

SR 锁存器用于各种模拟应用中，包括振荡器电路、单脉冲电路、迟滞控制器和模拟时序应用。

### 19.1 锁存器工作原理

该锁存器是一个不依赖时钟源的置 1- 复位锁存器。每个置 1 和复位输入都是高电平有效。锁存器可由以下项置 1 或复位：

- 软件控制（SRPS 和 SRPR 位）
- 比较器 C1 输出（SYNCC1OUT）
- 比较器 C2 输出（SYNCC2OUT）
- SRI 引脚
- 可编程时钟（SRCLK）

可以使用 SRCON0 寄存器的 SRPS 和 SRPR 位来分别置 1 或复位 SR 锁存器。锁存器是复位优先型的，因此，如果置 1 和复位输入同时为高电平，锁存器将进入复位状态。SRPS 和 SRPR 位都是自复位的，因此，要完成锁存器的置 1 或复位操作，仅需对这两位中的 1 位进行一次写操作即可。

比较器 C1 或 C2 的输出都可以用作 SR 锁存器的置 1 或复位输入。任一比较器的输出都可以与 Timer1 时钟源同步。更多信息，请参见第 18.0 节“比较器模块”和第 21.0 节“带门控的 Timer1 模块”。

SRI 引脚上的外部时钟源可以用作 SR 锁存器的置 1 或复位输入。

提供的内部时钟源也可以定期置 1 或复位 SR 锁存器。SRCON0 寄存器中的 SRCLK<2:0> 位用于选择时钟源周期。SRCON1 寄存器的 SRSCKE 和 SRRCKE 位可以使时钟源分别置 1 或复位 SR 锁存器。

**注：** 同时从任何一个时钟源使能置 1 和复位输入可能导致不确定操作，因为无法确保复位优先。

## 19.2 锁存器输出

SRCON0 寄存器的 SRQEN 和 SRNQEN 位可分别控制 Q 和  $\bar{Q}$  锁存器输出。这两种 SR 锁存器输出可同时直接输出到 I/O 引脚。可以使用 APFCON 寄存器的 SRNQSEL 位将  $\bar{Q}$  锁存器输出引脚功能移至备用引脚。

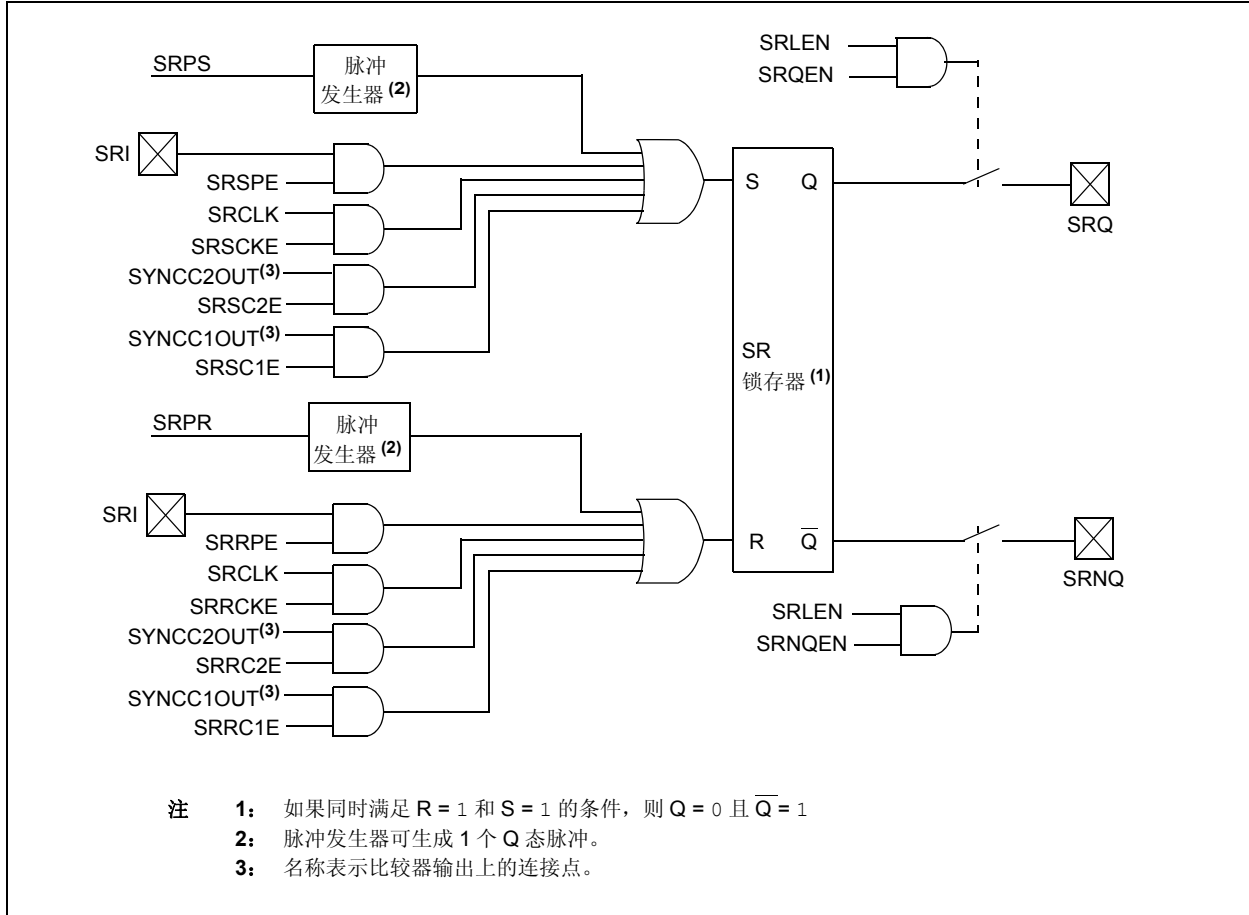
必须将相应端口对应的 TRIS 位清零，以使能该端口引脚的输出驱动器。

## 19.3 复位的影响

发生任何器件复位时，SR 锁存器输出都不会初始化为已知状态。用户固件负责初始化锁存器输出，然后使能输出引脚。

# PIC16(L)F1933

图 19-1: SR 锁存器简化框图



**表 19-1: SRCLK 频率表**

SRCLK	分频比	Fosc = 32 MHz	Fosc = 20 MHz	Fosc = 16 MHz	Fosc = 4 MHz	Fosc = 1 MHz
111	512	62.5 kHz	39.0 kHz	31.3 kHz	7.81 kHz	1.95 kHz
110	256	125 kHz	78.1 kHz	62.5 kHz	15.6 kHz	3.90 kHz
101	128	250 kHz	156 kHz	125 kHz	31.25 kHz	7.81 kHz
100	64	500 kHz	313 kHz	250 kHz	62.5 kHz	15.6 kHz
011	32	1 MHz	625 kHz	500 kHz	125 kHz	31.3 kHz
010	16	2 MHz	1.25 MHz	1 MHz	250 kHz	62.5 kHz
001	8	4 MHz	2.5 MHz	2 MHz	500 kHz	125 kHz
000	4	8 MHz	5 MHz	4 MHz	1 MHz	250 kHz

**寄存器 19-1: SRCON0: SR 锁存器控制寄存器 0**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/S-0/0	R/S-0/0
SRLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	S = 只可置 1 位

- bit 7      **SRLEN:** SR 锁存器使能位  
 1 = 使能 SR 锁存器  
 0 = 禁止 SR 锁存器
- bit 6-4    **SRCLK<2:0>:** SR 锁存器时钟分频比位  
 000 = 每 4 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 001 = 每 8 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 010 = 每 16 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 011 = 每 32 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 100 = 每 64 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 101 = 每 128 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 110 = 每 256 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲  
 111 = 每 512 个 Fosc 时钟周期生成 1 个宽度为 Fosc 的脉冲
- bit 3      **SRQEN:** SR 锁存器 Q 输出使能位  
 如果 **SRLEN = 1**:  
     1 = Q 出现在 SRQ 引脚上  
     0 = 禁止外部 Q 输出  
 如果 **SRLEN = 0**:  
     禁止 SR 锁存器
- bit 2      **SRNQEN:** SR 锁存器  $\bar{Q}$  输出使能位  
 如果 **SRLEN = 1**:  
     1 =  $\bar{Q}$  出现在 SRnQ 引脚上  
     0 = 禁止外部  $\bar{Q}$  输出  
 如果 **SRLEN = 0**:  
     禁止 SR 锁存器
- bit 1      **SRPS:** 使 SR 锁存器置 1 的脉冲输入位 <sup>(1)</sup>  
 1 = 1Q 时钟周期的置 1 脉冲输入  
 0 = 对置 1 输入无影响。
- bit 0      **SRPR:** 使 SR 锁存器复位的脉冲输入位 <sup>(1)</sup>  
 1 = 1Q 时钟周期的复位脉冲输入  
 0 = 对复位输入无影响。

注 1: 仅置 1, 始终读回 0。

# PIC16(L)F1933

**寄存器 19-2: SRCON1: SR 锁存器控制寄存器 1**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                            0 = 清零

- bit 7            **SRSPE:** SR 锁存器外设置 1 使能位  
1 = SR 锁存器在 SRI 引脚为高电平时置 1  
0 = SRI 引脚不影响 SR 锁存器的置 1 输入
- bit 6            **SRSCKE:** SR 锁存器置 1 时钟使能位  
1 = SRCLK 为 SR 锁存器的置 1 输入提供脉冲  
0 = SRCLK 不影响 SR 锁存器的置 1 输入
- bit 5            **SRSC2E:** SR 锁存器 C2 置 1 使能位  
1 = SR 锁存器在 C2 比较器输出为高电平时置 1  
0 = C2 比较器输出不影响 SR 锁存器的置 1 输入
- bit 4            **SRSC1E:** SR 锁存器 C1 置 1 使能位  
1 = SR 锁存器在 C1 比较器输出为高电平时置 1  
0 = C1 比较器输出不影响 SR 锁存器的置 1 输入
- bit 3            **SRRPE:** SR 锁存器外设复位使能位  
1 = SR 锁存器在 SRI 引脚为高电平时复位  
0 = SRI 引脚不影响 SR 锁存器的复位输入
- bit 2            **SRRCKE:** SR 锁存器复位时钟使能位  
1 = SRCLK 为 SR 锁存器的复位输入提供脉冲  
0 = SRCLK 不影响 SR 锁存器的复位输入
- bit 1            **SRRC2E:** SR 锁存器 C2 复位使能位  
1 = SR 锁存器在 C2 比较器输出为高电平时复位  
0 = C2 比较器输出不影响 SR 锁存器的复位输入
- bit 0            **SRRC1E:** SR 锁存器 C1 复位使能位  
1 = SR 锁存器在 C1 比较器输出为高电平时复位  
0 = C1 比较器输出不影响 SR 锁存器的复位输入

**表 19-2: 与 SR 锁存器模块相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	121
SRCON0	SRLLEN	SRCLK<2:0>			SRQEN	SRNQEN	SRPS	SRPR	171
SRCON1	SRSPE	SRSCKE	SRSC2E	SRSC1E	SRRPE	SRRCKE	SRRC2E	SRRC1E	172
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120

图注: — = 未实现单元, 读为 0。SR 锁存器模块不使用阴影单元。

## 20.0 TIMER0 模块

Timer0 模块是具有以下功能的 8 位定时器 / 计数器：

- 8 位定时器 / 计数器寄存器 (TMR0)
- 8 位预分频器 (独立于看门狗定时器)
- 可编程的内部或外部时钟源
- 可编程的外部时钟边沿选择
- 溢出时中断
- TMR0 可用于门控 Timer1

图 20-1 是 Timer0 模块的框图。

### 20.1 Timer0 工作原理

Timer0 模块可以用作 8 位定时器或 8 位计数器。

#### 20.1.1 8 位定时器模式

如果不使用预分频器，则 Timer0 模块在每指令周期递增 1。通过将 OPTION\_REG 寄存器的 TMR0CS 位清零可选择 8 位定时器模式。

如果对 TMR0 执行写操作，则在接下来的两个指令周期，它都不会递增 1。

**注：** 考虑到写入 TMR0 后两个指令周期的延时，可调整写入 TMR0 寄存器的值。

#### 20.1.2 8 位计数器模式

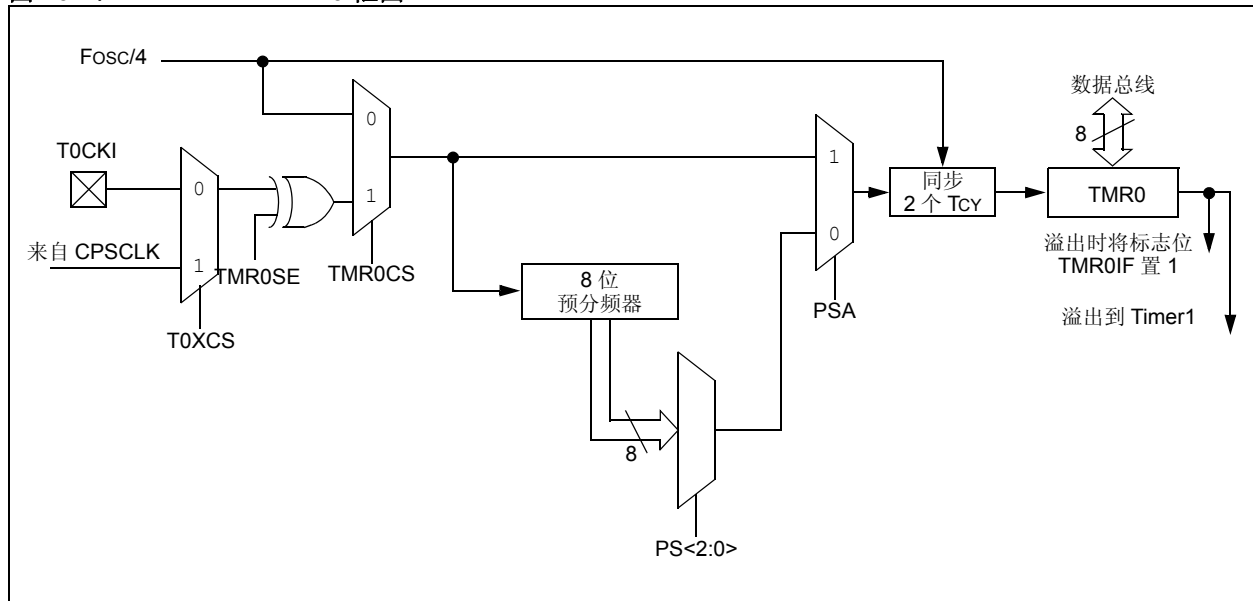
在 8 位计数器模式下，Timer0 模块将在 T0CKI 引脚信号或电容触摸传感振荡器 (CPSCCLK) 信号的每个上升沿或下降沿进行递增计数。

通过将 OPTION\_REG 寄存器中的 TMR0CS 位设置为 1 且将 CPSCON0 寄存器中的 T0XCS 位复位为 0 来选择使用 T0CKI 引脚的 8 位计数器模式。

通过将 OPTION\_REG 寄存器中的 TMR0CS 位设置为 1 且将 CPSCON0 寄存器中的 T0XCS 位设置为 1 来选择使用电容触摸传感振荡器 (CPSCCLK) 信号的 8 位计数器模式。

任一输入源的递增沿是上升沿还是下降沿由 OPTION\_REG 寄存器中的 TMR0SE 位确定。

图 20-1: TIMER0 框图



# PIC16(L)F1933

---

## 20.1.3 软件可编程预分频器

软件可编程预分频器专用于 Timer0。通过将 OPTION\_REG 寄存器的 PSA 位清零可启用预分频器。

**注：** 看门狗定时器（WDT）使用自己独立的预分频器。

Timer0 模块具有 8 种预分频比选择，范围为 1:2 至 1:256。可通过 OPTION\_REG 寄存器中的 PS<2:0> 位选择预分频值。要使 Timer0 模块具有 1:1 的预分频值，必须通过将 OPTION\_REG 寄存器的 PSA 位置 1 来禁止预分频器。

该预分频器不可读写。所有写 TMR0 寄存器的指令都将清零预分频器。

## 20.1.4 TIMER0 中断

当 TMR0 寄存器从 FFh 溢出至 00h 时，Timer0 会产生中断。每次 TMR0 寄存器溢出时，不管是否允许 Timer0 中断，INTCON 寄存器的 TMR0IF 中断标志位都将置 1。只能用软件将 TMR0IF 位清零。Timer0 中断允许位是 INTCON 寄存器的 TMR0IE 位。

**注：** 由于在休眠状态下定时器是关闭的，所以 Timer0 中断无法唤醒处理器。

## 20.1.5 8 位同步计数器模式

在 8 位计数器模式下，必须使 T0CKI 引脚信号的递增沿与指令时钟同步。同步可通过在指令时钟的 Q2 和 Q4 周期对预分频器输出进行采样实现。外部时钟源的高、低电平时间必须符合应用的电气规范章节中给出的时序要求。

## 20.1.6 休眠期间的工作原理

处理器处于休眠模式时，Timer0 无法工作。当处理器处于休眠模式时，TMR0 寄存器的内容保持不变。

## 20.2 选项和 Timer0 控制寄存器

寄存器 20-1: **OPTION\_REG: 选项寄存器**

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>		
bit 7							bit 0

**图注:**

R = 可读位                      W = 可写位                      U = 未实现位, 读为 0  
u = 不变                          x = 未知                          -n/n = POR 和 BOR 时的值 / 所有其他复位时的值  
1 = 置 1                          0 = 清零

- bit 7            **WPUEN:** 弱上拉使能位  
1 = 禁止所有弱上拉 (MCLR 除外, 如果它被使能的话)  
0 = 根据各 WPUx 锁存值使能弱上拉
- bit 6            **INTEDG:** 中断边沿选择位  
1 = INT 引脚信号的上升沿触发中断  
0 = INT 引脚信号的下降沿触发中断
- bit 5            **TMR0CS:** Timer0 时钟源选择位  
1 = T0CKI 引脚上信号的跳变  
0 = 内部指令周期时钟 (Fosc/4)
- bit 4            **TMR0SE:** Timer0 时钟源边沿选择位  
1 = 在 T0CKI 引脚电平发生由高到低的跳变时递增  
0 = 在 T0CKI 引脚电平发生由低到高的跳变时递增
- bit 3            **PSA:** 预分频器分配位  
1 = 不将预分频器分配给 Timer0 模块  
0 = 将预分频器分配给 Timer0 模块
- bit 2-0        **PS<2:0>:** 预分频器分频比选择位

位值	Timer0 分频比
000	1 : 2
001	1 : 4
010	1 : 8
011	1 : 16
100	1 : 32
101	1 : 64
110	1 : 128
111	1 : 256

表 20-1: 与 **TIMER0** 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
CPSCON0	CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	T0XCS	311
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
OPTION_REG	WPUEN	INTEDG	TMR0CS	TMR0SE	PSA	PS<2:0>			175
TMR0	Timer0 模块寄存器								173*
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120

图注: — = 未实现单元, 读为 0。Timer0 模块不使用阴影单元。

\* 该页提供寄存器信息。

# PIC16(L)F1933

---

注:



## 21.0 带门控的 TIMER1 模块

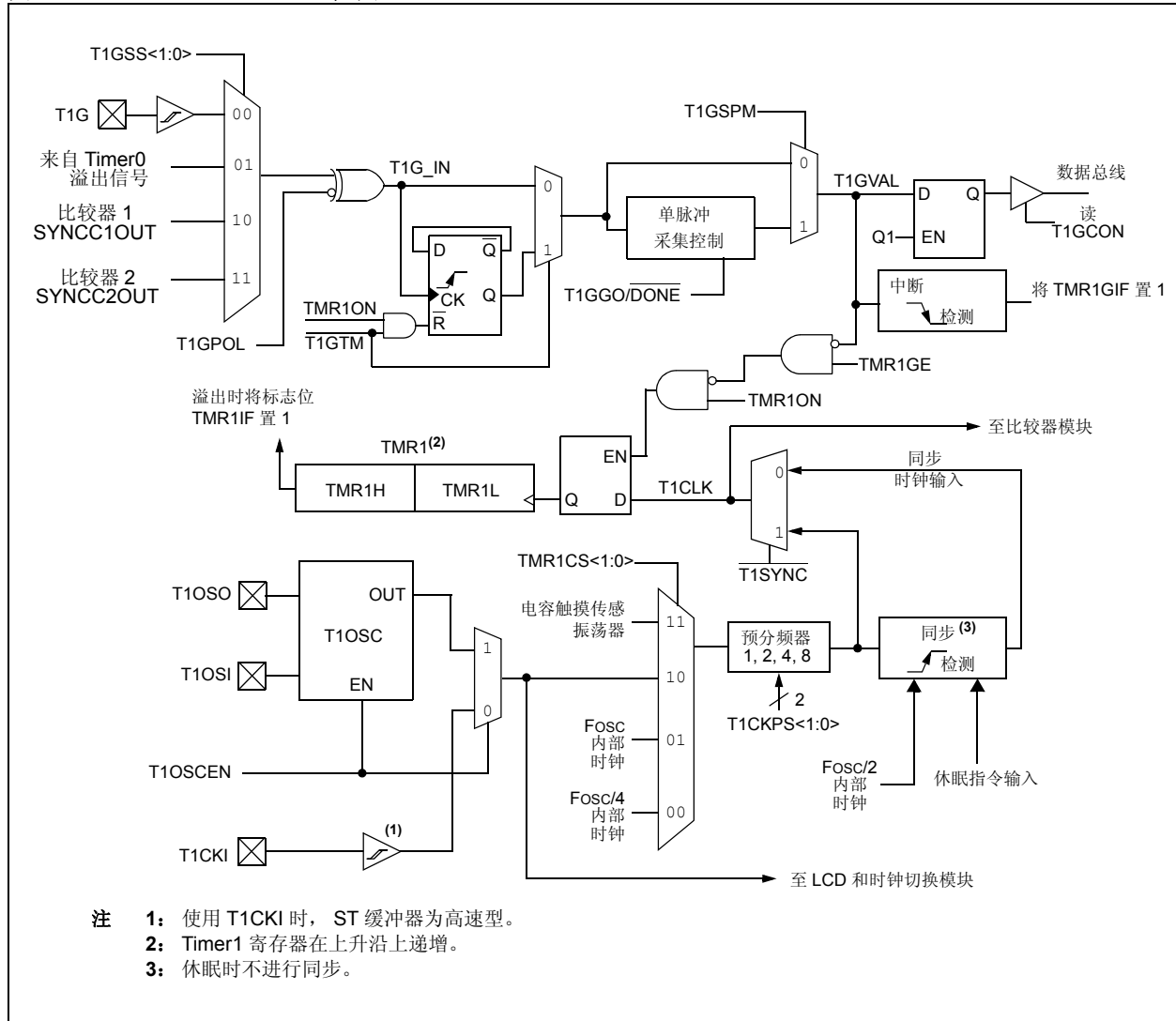
Timer1 模块是具有以下功能的 16 位定时器 / 计数器:

- 16 位定时器 / 计数器寄存器对 (TMR1H:TMR1L)
- 可编程的内部或外部时钟源
- 2 位预分频器
- 专用 32 kHz 振荡器电路
- 可选的同步比较器输出
- 多个 Timer1 门控 (计数使能) 信号源
- 溢出时中断
- 溢出时唤醒 (仅外部时钟异步模式)
- 用于捕捉 / 比较功能的时基
- 特殊事件触发器 (用于 CCP/ECCP)
- 可选门控信号源极性

- 门控交替计数模式
- 门控单脉冲模式
- 门控值状态
- 门控事件中断

图 21-1 给出了 Timer1 模块的框图。

图 21-1: TIMER1 框图



# PIC16(L)F1933

## 21.1 Timer1 工作原理

Timer1 模块是 16 位递增计数器，通过 TMR1H:TMR1L 寄存器对可对该模块进行访问。通过写 TMR1H 或 TMR1L 可直接更新该计数器。

当由内部时钟源提供时钟时，该模块为定时器，在每个指令周期递增 1。当由外部时钟源提供时钟时，该模块可用作定时器或计数器，并在外部时钟源的每个选定边沿递增 1。

分别配置 T1CON 寄存器中的 TMR1ON 位和 T1GCON 寄存器中的 TMR1GE 位可使能 Timer1。表 21-1 显示了 Timer1 的使能选择。

表 21-1: TIMER1 的使能选择

TMR1ON	TMR1GE	Timer1 工作状态
0	0	关闭
0	1	关闭
1	0	始终开启
1	1	使能计数

## 21.2 时钟源选择

T1CON 寄存器的 TMR1CS<1:0> 和 T1OSCEN 位用于选择 Timer1 的时钟源。表 21-2 显示了时钟源选择。

### 21.2.1 内部时钟源

选用内部时钟源时，TMR1H:TMR1L 寄存器对将在 Fosc 的整数倍（取决于 Timer1 预分频器）处递增。

选择 Fosc 内部时钟源时，Timer1 寄存器的值在每个指令时钟周期递增 4 个计数。由于此条件，读取 Timer1 值时的分辨率误差为 2 LSB。要利用 Timer1 的全分辨率，必须使用异步输入信号对 Timer1 时钟输入进行门控。

可以使用以下异步源：

- T1G 引脚上的异步事件对 Timer1 进行门控
- C1 或 C2 比较器输入对 Timer1 进行门控

### 21.2.2 外部时钟源

选择外部时钟源时，Timer1 模块可作为定时器或计数器。

使能计数模式时，Timer1 在外部时钟输入 T1CKI 信号或电容触摸传感振荡器信号的上升沿递增。这些外部时钟源都可与单片机系统时钟同步，也可以异步运行。

当用作带时钟振荡器的定时器时，可将外部 32.768 kHz 晶振与专用内部振荡器电路结合使用。

**注：** 在计数器模式下，在发生以下任何一个或多个条件时必须先经过一个下降沿，计数器才可以在上升沿进行第一次递增计数：

- 上电复位后使能 Timer1
- 写 TMR1H 或 TMR1L
- 禁止 Timer1
- 先将 T1CKI 拉为高电平来禁止 Timer1 (TMR1ON = 0)，然后将 T1CKI 拉为低电平来使能 Timer1 (TMR1ON = 1)

表 21-2: 时钟源选择

TMR1CS1	TMR1CS0	T1OSCEN	时钟源
0	0	x	指令时钟 (Fosc/4)
0	1	x	系统时钟 (Fosc)
1	0	0	T1CKI 引脚上的外部时钟
1	0	0	T1CKI 引脚上的外部时钟
1	1	x	电容触摸传感振荡器

## 21.3 Timer1 预分频器

Timer1 具有四种预分频比选择，允许对时钟输入进行 1、2、4 或 8 分频。T1CON 寄存器的 T1CKPS 位控制该预分频计数器。不能直接对预分频计数器进行读写操作；但是，通过写 TMR1H 或 TMR1L 可清零预分频计数器。

## 21.4 Timer1 振荡器

在引脚 T1OSI（输入）和 T1OSO（放大器输出）之间连接有一个内置的专用低功耗 32.768 kHz 振荡器电路。该内部电路要与外部 32.768 kHz 晶振结合使用。

可通过将 T1CON 寄存器的 T1OSCEN 位置 1 来使能该振荡器电路。该振荡器将在休眠期间继续运行。

**注：** 在使用振荡器之前需要一段起振和稳定时间。因此，应该将 T1OSCEN 置 1 并在经过一段合适的延时后才使能 Timer1。与 OST 延时类似的适当延时可以通过清零 TMR1IF 位并将 TMR1H:TMR1L 寄存器预置为 FC00h 由软件实现。当经过 1024 个时钟周期后，TMR1IF 标志位置 1，从而表示振荡器正在运行并相当稳定。

## 21.5 Timer1 在异步计数器模式下的工作原理

如果 T1CON 寄存器的控制位  $\overline{T1SYNC}$  被置 1，外部时钟输入就不同步。定时器的递增计数与内部相位时钟异步。如果选用了外部时钟源，则定时器将在休眠期间继续运行，并在溢出时产生中断，从而唤醒处理器。但是，在用软件对定时器进行读 / 写操作时应该特别小心（见第 21.5.1 节“异步计数器模式下对 Timer1 的读写操作”）。

**注：** 当从同步工作切换到异步工作模式时，有可能跳过一次递增。当从异步工作切换到同步工作模式时，有可能产生一次额外的递增。

### 21.5.1 异步计数器模式下对 TIMER1 的读写操作

当定时器采用外部异步时钟工作时，对 TMR1H 或 TMR1L 的读操作将确保有效（由硬件负责）。但是用户应注意，通过读取 2 个 8 位值本身来读取 16 位定时器会产生某些问题，因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户停止定时器后再写入需要的数值。当寄存器正在递增计数时，向定时器的寄存器写入数据可能会产生写争用，从而可能在 TMR1H:TMR1L 寄存器对中产生不可预测的值。

## 21.6 Timer1 门控

Timer1 可配置为自由计数，或使用 Timer1 门控电路来使能和禁止计数功能。这也称为 Timer1 门控使能。

Timer1 门控也可通过多个可选信号源来驱动。

### 21.6.1 TIMER1 门控使能

通过将 T1GCON 寄存器的 TMR1GE 位置 1 来使能 Timer1 门控使能模式。可使用 T1GCON 寄存器的 T1GPOL 位来配置 Timer1 门控使能模式的极性。

使能 Timer1 门控使能模式时，Timer1 在 Timer1 时钟源的上升沿递增计数。禁止 Timer1 门控使能模式时，不进行递增计数且 Timer1 保持当前计数。时序详情，请参见图 21-3。

表 21-3: TIMER1 门控使能选择

T1CLK	T1GPOL	T1G	Timer1 工作状态
↑	0	0	计数
↑	0	1	保持计数
↑	1	0	保持计数
↑	1	1	计数

# PIC16(L)F1933

## 21.6.2 TIMER1 门控信号源选择

可从 4 个门控信号源中选择 1 个作为 Timer1 门控信号源。门控信号源选择由 T1GCON 寄存器的 T1GSS 位控制。也可选择每个可用门控信号源的极性。门控信号源的极性选择由 T1GCON 寄存器的 T1GPOL 位控制。

表 21-4: TIMER1 门控信号源

T1GSS	Timer1 门控信号源
00	Timer1 门控引脚
01	Timer0 溢出 (TMR0 从 FFh 递增到 00h)
10	比较器 1 输出 SYNCC1OUT (可选的 Timer1 同步输出)
11	比较器 2 输出 SYNCC2OUT (可选的 Timer1 同步输出)

### 21.6.2.1 T1G 引脚用作门控源

T1G 引脚是 Timer1 门控信号源之一。可用于为 Timer1 门控电路提供外部门控信号源。

### 21.6.2.2 Timer0 溢出用作门控源

当 Timer0 从 FFh 递增到 00h 时，将自动生成一个由低到高的脉冲信号并内部提供给 Timer1 门控电路。

### 21.6.2.3 比较器 C1 用作门控源

可选择将比较器 1 的输出用作 Timer1 的门控信号源。比较器 1 的输出 (SYNCC1OUT) 可与 Timer1 时钟同步，也可异步运行。更多信息，请参见第 18.4.1 节“比较器输出同步”。

### 21.6.2.4 比较器 C2 用作门控源

可选择将比较器 2 的输出用作 Timer1 的门控信号源。比较器 2 的输出 (SYNCC2OUT) 可与 Timer1 时钟同步，也可异步运行。更多信息，请参见第 18.4.1 节“比较器输出同步”。

## 21.6.3 TIMER1 门控交替计数模式

当使能了 Timer1 门控交替计数模式时，可以测量 Timer1 门控信号的完整周期长度，而不是单电平脉冲信号的持续时间。

Timer1 门控信号源经由一个单稳态触发器输送到 Timer1，该单稳态触发器在信号的每个递增边沿改变状态。有关时序的详细信息，请参见图 21-4。

通过将 T1GCON 寄存器的 T1GTM 位置 1 来使能 Timer1 门控交替计数模式。当 T1GTM 位清零时，单稳态触发器也将清零并保持清零状态。该模式对于控制要计数的边沿是必需的。

**注：** 使能交替计数模式的同时更改门控信号的极性可能会导致操作不确定。

## 21.6.4 TIMER1 门控单脉冲模式

使能了 Timer1 门控单脉冲模式时，可以捕捉单脉冲门控事件。首先，通过将 T1GCON 寄存器中的 T1GSPM 位置 1 来使能 Timer1 门控单脉冲模式。其次，必须将 T1GCON 寄存器中的 T1GGO/DONE 位置 1。Timer1 将在下一个递增沿完全使能。在脉冲信号的后沿，T1GGO/DONE 位将自动清零。不允许其他门控事件使 Timer1 递增计数，直到用软件将 T1GGO/DONE 位再次置 1。时序详情，请参见图 21-5。

如果通过清零 T1GCON 寄存器中的 T1GSPM 位禁止门控单脉冲模式，T1GGO/DONE 位也会清零。

同时使能交替计数模式和单脉冲模式将允许这两种模式一起工作，从而允许测量 Timer1 门控信号源上的周期数。时序详情，请参见图 21-6。

## 21.6.5 TIMER1 门控值状态

采用 Timer1 门控值状态时，可以读取门控值的当前电平。该值存储在 T1GCON 寄存器中的 T1GVAL 位中。即使 Timer1 门控未使能 (TMR1GE 位清零)，T1GVAL 位也是有效的。

## 21.6.6 TIMER1 门控事件中断

允许 Timer1 门控事件中断时，可以在门控事件结束时产生中断。出现 T1GVAL 的下降沿时，PIR1 寄存器中的 TMR1GIF 标志位将置 1。如果 PIE1 寄存器中的 TMR1GIE 位置 1，则将响应中断。

即使 Timer1 门控未使能 (TMR1GE 位清零)，TMR1GIF 标志位也是有效的。

## 21.7 Timer1 中断

Timer1 寄存器对 (TMR1H:TMR1L) 递增计数到 FFFFh 并计满返回 0000h。当 Timer1 计满返回时, PIR1 寄存器的 Timer1 中断标志位置 1。要允许计满返回中断, 应将以下位置 1:

- T1CON 寄存器的 TMR1ON 位
- PIE1 寄存器的 TMR1IE 位
- INTCON 寄存器的 PEIE 位
- INTCON 寄存器的 GIE 位

在中断服务程序中将 TMR1IF 清零可以清除中断。

**注:** 允许中断前, 应将 TMR1H:TMR1L 寄存器对以及 TMR1IF 位清零。

## 21.8 休眠期间的 Timer1 工作原理

只有设置为异步计数器模式时, Timer1 才能在休眠期间工作。在该模式下, 可使用外部晶振或时钟源使计数器递增计数。通过如下步骤设置定时器以唤醒器件:

- 必须将 T1CON 寄存器的 TMR1ON 位置 1
- 必须将 PIE1 寄存器的 TMR1IE 位置 1
- 必须将 INTCON 寄存器的 PEIE 位置 1
- 必须将 T1CON 寄存器的 T1SYNC 位置 1
- 必须配置 T1CON 寄存器的 TMR1CS 位
- 必须配置 T1CON 寄存器的 T1OSCEN 位

器件将在溢出时被唤醒并执行下一条指令。如果将 INTCON 寄存器的 GIE 位置 1, 器件将调用中断服务程序。

不管 T1SYNC 位是否置 1, Timer1 振荡器都将在休眠模式下继续工作。

## 21.9 ECCP/CCP 捕捉 / 比较时基

工作在捕捉或比较模式下时, CCP 模块使用 TMR1H:TMR1L 寄存器对作为时基。

在捕捉模式下, 发生配置的事件时, TMR1H:TMR1L 寄存器对中的值被复制到 CCPR1H:CCPR1L 寄存器对中。

在比较模式下, 当 CCPR1H:CCPR1L 寄存器对中的值与 TMR1H:TMR1L 寄存器对中的值匹配时触发事件。该事件可以是特殊事件触发信号。

更多信息, 请参见第 12.0 节 “I/O 端口”。

## 21.10 ECCP/CCP 特殊事件触发信号

当将 CCP 配置为触发特殊事件时, 触发信号将清零 TMR1H:TMR1L 寄存器对。该特殊事件不会导致 Timer1 中断。CCP 模块仍可配置为产生 CCP 中断。

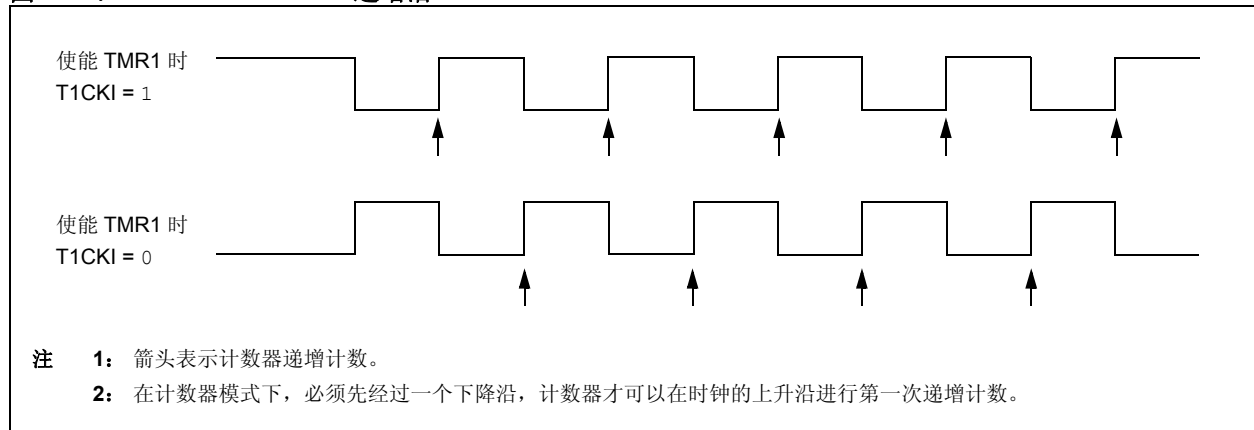
在这种工作模式下, CCPR1H:CCPR1L 寄存器对变成了 Timer1 的周期寄存器。

Timer1 应该同步, 并应该选择 Fosc/4 作为时钟源以利用特殊事件触发信号。Timer1 的异步运行会导致错过特殊事件触发信号。

如果对 TMR1H 或 TMR1L 的写操作与 CCP 的特殊事件触发信号同时发生, 则写操作具有优先权。

更多信息, 请参见第 15.2.5 节 “特殊事件触发器”。

图 21-2: TIMER1 递增沿



# PIC16(L)F1933

图 21-3: TIMER1 门控使能模式

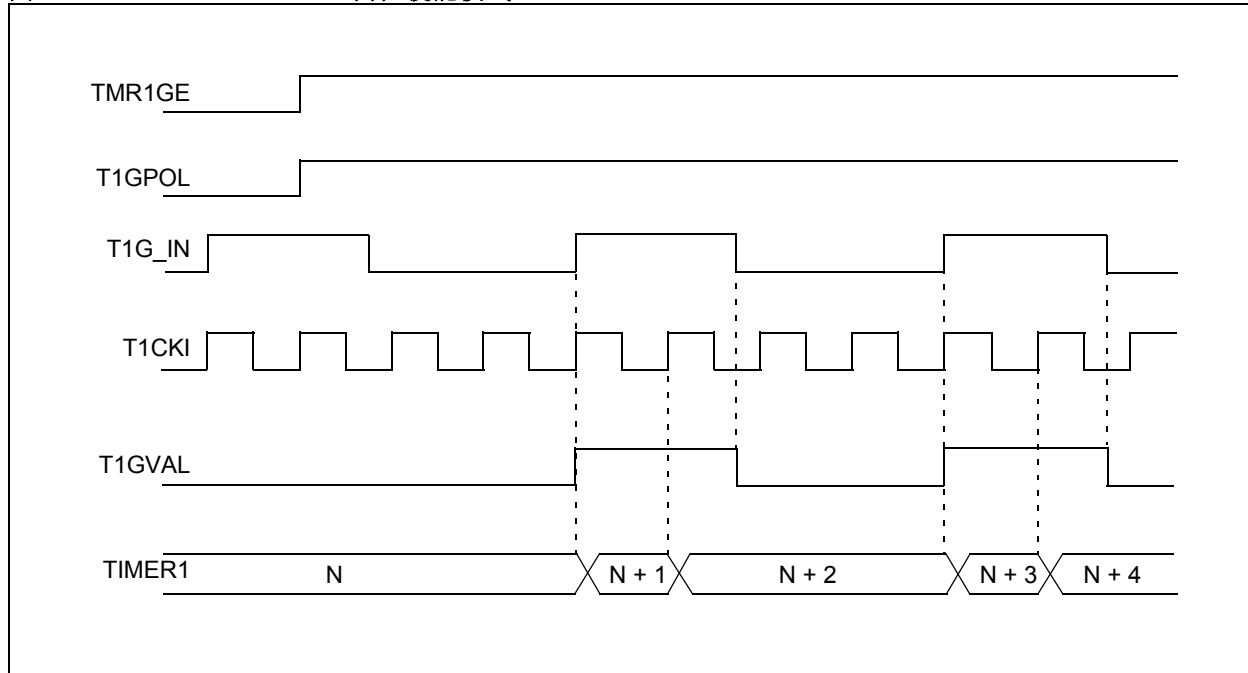


图 21-4: TIMER1 门控交替计数模式

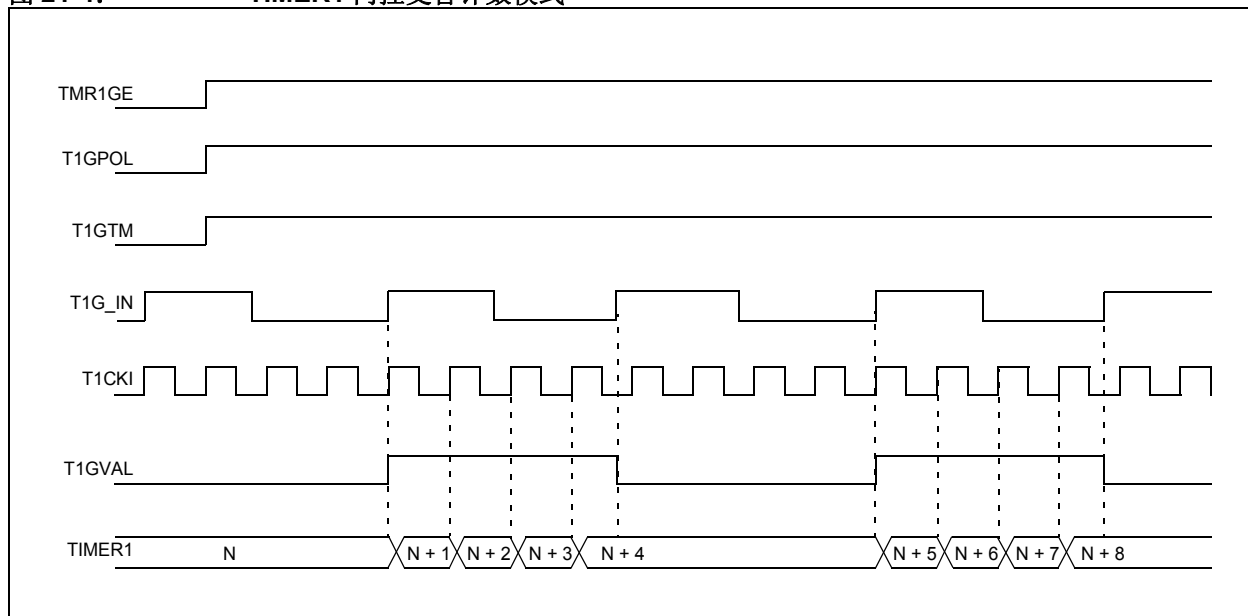
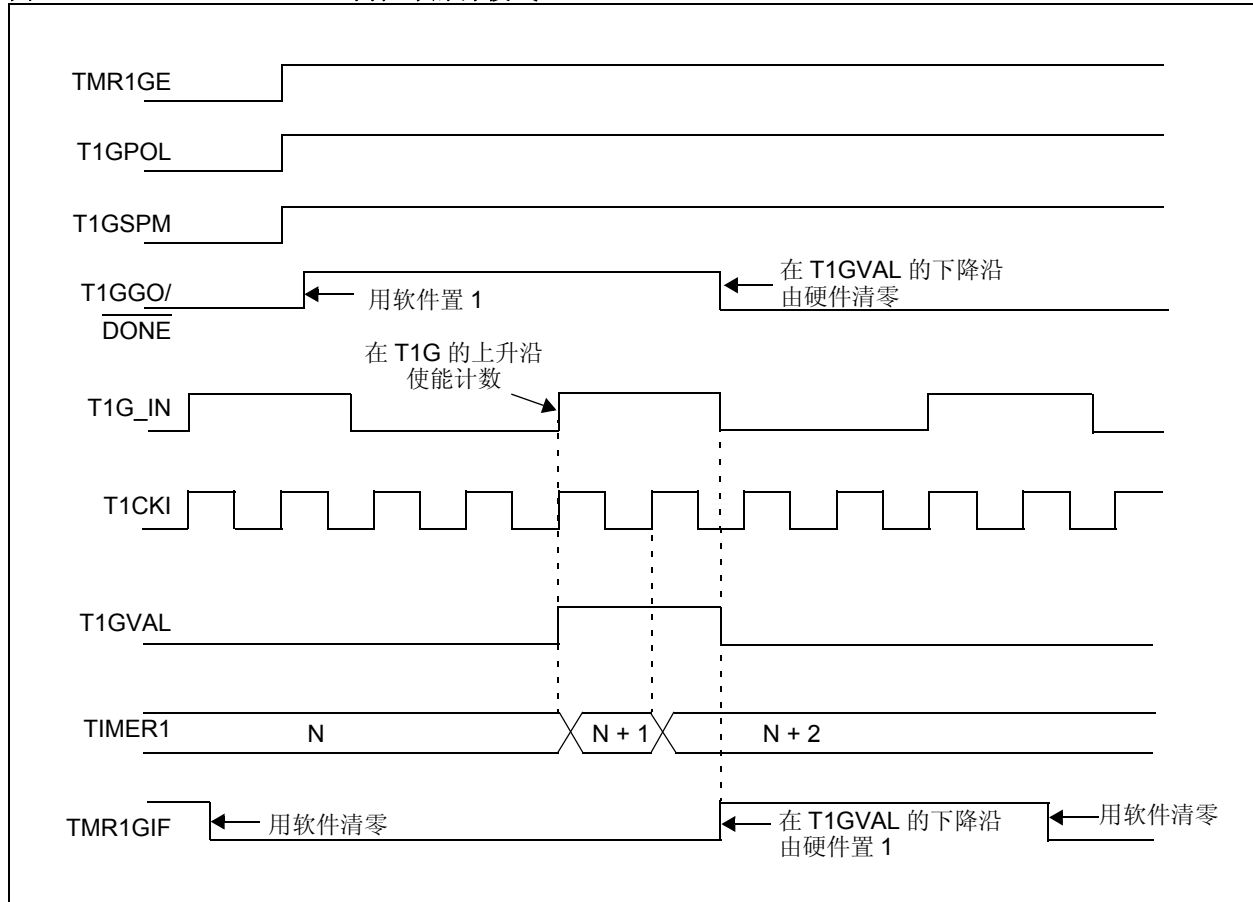
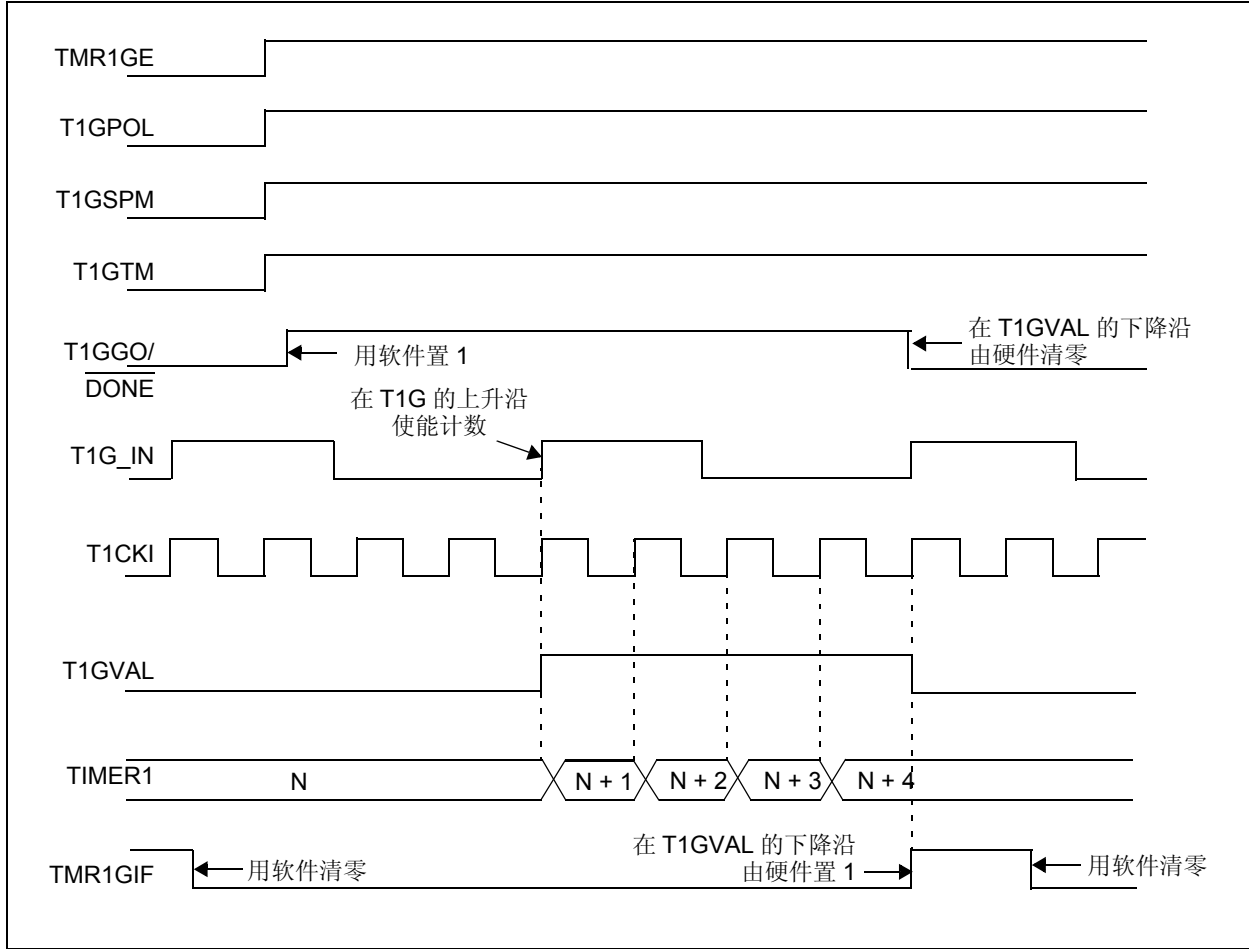


图 21-5: TIMER1 门控单脉冲模式



# PIC16(L)F1933

图 21-6: TIMER1 门控单脉冲和交替计数组合模式





## 21.11 Timer1 控制寄存器

Timer1 控制寄存器 (T1CON) 如寄存器 21-1 所示, 用于控制 Timer1 并选择 Timer1 模块的各种功能。

**寄存器 21-1: T1CON: TIMER1 控制寄存器**

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	U-0	R/W-0/u
TMR1CS<1:0>		T1CKPS<1:0>		T1OSCN	$\overline{T1SYNC}$	—	TMR1ON
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6

**TMR1CS<1:0>:** Timer1 时钟源选择位

11 = Timer1 时钟源是电容触摸传感振荡器 (CAPOSC)

10 = Timer1 时钟源是引脚或振荡器:

如果 T1OSCN = 0:

来自 T1CKI 引脚的外部时钟 (上升沿)

如果 T1OSCN = 1:

T1OSI/T1OSO 引脚上的晶振

01 = Timer1 时钟源为系统时钟 (Fosc)

00 = Timer1 时钟源为指令时钟 (Fosc/4)

bit 5-4

**T1CKPS<1:0>:** Timer1 输入时钟预分频比选择位

11 = 1:8 预分频比

10 = 1:4 预分频比

01 = 1:2 预分频比

00 = 1:1 预分频比

bit 3

**T1OSCN:** LP 振荡器使能控制位

1 = 使能专用的 Timer1 振荡器电路

0 = 禁止专用的 Timer1 振荡器电路

bit 2

**T1SYNC:** Timer1 外部时钟输入同步控制位

TMR1CS<1:0> = 1X

1 = 不与外部时钟输入同步

0 = 外部时钟输入与系统时钟 (Fosc) 同步

TMR1CS<1:0> = 0X

忽略此位。

bit 1

**未实现:** 读为 0

bit 0

**TMR1ON:** Timer1 启用位

1 = 使能 Timer1

0 = 停止 Timer1

清零 Timer1 门控单稳态触发器

# PIC16(L)F1933

## 21.12 Timer1 门控寄存器

Timer1 门控寄存器 (T1GCON) 如寄存器 21-2 所示, 用于控制 Timer1 门控。

寄存器 21-2: T1GCON: TIMER1 门控寄存器

R/W-0/u	R/W-0/u	R/W-0/u	R/W-0/u	R/W/HC-0/u	R-x/x	R/W-0/u	R/W-0/u
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>	
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 由硬件清零

- bit 7      **TMR1GE:** Timer1 门控使能位  
如果 TMR1ON = 0:  
忽略此位。  
如果 TMR1ON = 1:  
1 = Timer1 计数由 Timer1 门控功能控制  
0 = Timer1 计数与 Timer1 门控功能无关
- bit 6      **T1GPOL:** Timer1 门控信号极性位  
1 = Timer1 门控信号高电平有效 (门控信号为高电平时 Timer1 计数)  
0 = Timer1 门控信号低电平有效 (门控信号为低电平时 Timer1 计数)
- bit 5      **T1GTM:** Timer1 门控交替计数模式位  
1 = 使能 Timer1 门控交替计数模式  
0 = 禁止 Timer1 门控交替计数模式并清零交替计数单稳态触发器  
Timer1 门控单稳态触发器在每个上升沿改变输出状态。
- bit 4      **T1GSPM:** Timer1 门控单脉冲模式位  
1 = 使能 Timer1 门控单脉冲模式且正在控制 Timer1 门控信号  
0 = 禁止 Timer1 门控单脉冲模式
- bit 3      **T1GGO/DONE:** Timer1 门控单脉冲采集状态位  
1 = Timer1 门控单脉冲采集已就绪, 正在等待边沿  
0 = Timer1 门控单脉冲采集已完成或还未开始  
当 T1GSPM 位清零时, 该位自动清零。
- bit 2      **T1GVAL:** Timer1 门控当前状态位  
指示提供给 TMR1H:TMR1L 的 Timer1 门控信号的当前状态。  
不受 Timer1 门控使能位 (TMR1GE) 的影响。
- bit 1-0    **T1GSS<1:0>:** Timer1 门控信号源选择位  
00 = Timer1 门控引脚  
01 = Timer0 溢出输出  
10 = 比较器 1 可选同步输出 (SYNCC1OUT)  
11 = 比较器 2 可选同步输出 (SYNCC2OUT)

**表 21-5: 与 TIMER1 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	126
CCP1CON	P1M<1:0>		DC1B<1:0>		CCP1M<3:0>				216
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				216
INTCON	GIE	PEIE	TMR0IE	INTE	IOIE	TMR0IF	INTF	IOCF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								181*
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								181*
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/ DONE	T1GVAL	T1GSS<1:0>		186

**图注:** — = 未实现单元，读为 0。Timer1 模块不使用阴影单元。

\* 该页提供寄存器信息。

# PIC16(L)F1933

---

注:

## 22.0 TIMER2/4/6 模块

提供了多达 3 个相同的 Timer2 类型的模块。为了遵守之前的命名协定，分别称这些定时器为 Timer2、Timer4 和 Timer6（也可以称为 Timer2/4/6）。

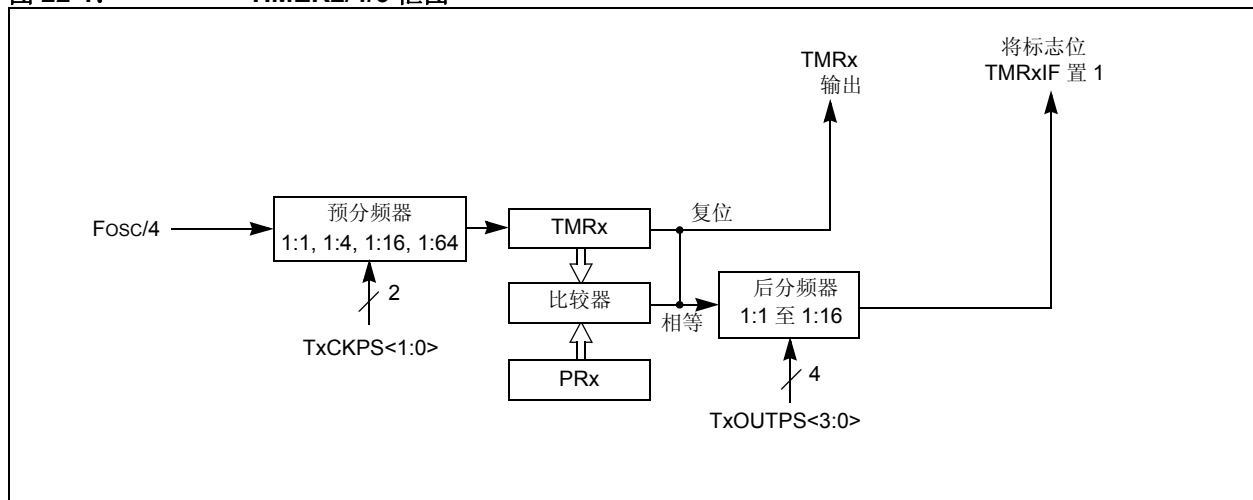
**注：** 本节中使用的“x”变量用于指示 Timer2、Timer4 或 Timer6。例如，TxCON 指 T2CON、T4CON或T6CON。PRx指PR2、PR4 或 PR6。

Timer2/4/6 模块具有以下功能：

- 8 位定时器和周期寄存器（分别称为 TMRx 和 PRx）
- 可读写（所有寄存器）
- 可软件编程的预分频比（1:1、1:4、1:16 和 1:64）
- 可软件编程的后分频比（1:1 至 1:16）
- TMRx 与相应的 PRx 分别匹配时中断
- 可选择作为 MSSP 模块的移位时钟（仅 Timer2）

请参见图 22-1 中的 Timer2/4/6 框图。

**图 22-1: TIMER2/4/6 框图**



# PIC16(L)F1933

## 22.1 Timer2/4/6 工作原理

Timer2/4/6 模块的时钟输入是系统指令时钟 (Fosc/4)。

TMRx 在每个时钟边沿递增计数，从 00h 开始。

时钟输入上的 4 位计数器 / 预分频器允许直接输入，提供 4 分频和 16 分频的预分频比选项。这些选项可通过预分频比控制位 (TxCON 寄存器的 TxCKPS<1:0>) 选择。在每个时钟周期将 TMRx 的值与周期寄存器 PRx 的值相比较。如果两者匹配，则比较器生成一个匹配信号作为定时器输出。该信号还可在下一个周期将 TMRx 的值复位为 00h，并驱动输出计数器 / 后分频器 (见第 22.2 节“Timer2/4/6 中断”)。

TMRx 和 PRx 寄存器都是可直接读写的。任何器件复位都将清零 TMRx 寄存器，同时 PRx 寄存器初始化为 FFh。发生以下事件时，预分频计数器和后分频计数器都将清零：

- 对 TMRx 寄存器进行写操作
- 对 TxCON 寄存器进行写操作
- 上电复位 (POR)
- 欠压复位 (BOR)
- MCLR 复位
- 看门狗定时器 (WDT) 复位
- 堆栈上溢复位
- 堆栈下溢复位
- RESET 指令

**注：** 写 TxCON 时 TMRx 不会清零。

## 22.2 Timer2/4/6 中断

Timer2/4/6 也可产生可选的器件中断。Timer2/4/6 输出信号 (TMRx 与 PRx 匹配) 是 4 位计数器 / 后分频器的输入。该计数器可使 PIRx 寄存器中的 TMRxIF 即 TMRx 匹配中断标志位置 1。将 TMRx 匹配中断允许位 (PIEx 寄存器的 TMRxIE) 置 1 来允许中断。

使用后分频比控制位 (TxCON 寄存器的 TxOUTPS<3:0>) 在 16 个后分频比选项 (1:1 至 1:16) 中进行选择。

## 22.3 Timer2/4/6 输出

TMRx 的不经分频的输出主要用于 CCP 模块，它用作 CCP 模块在 PWM 模式下工作时的时基。

可选择将 Timer2 作为 MSSP 模块在 SPI 模式下工作时的移位时钟源。其他信息，请参见第 24.0 节“主同步串行端口 (MSSP) 模块”。

## 22.4 休眠期间的 Timer2/4/6 工作原理

处理器处于休眠模式时，Timer2/4/6 定时器无法工作。此时，TMRx 和 PRx 寄存器的内容将保持不变。

## 22.5 Timer2/4/6 控制寄存器

寄存器 22-1: **TXCON: TIMER2/TIMER4/TIMER6 控制寄存器**

U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
—	TxOUTPS<3:0>			TMRxON	TxCKPS<1:0>		
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7

**未实现:** 读为 0

bit 6-3

**TxOUTPS<3:0>:** 定时器输出后分频比选择位

0000 = 1:1 后分频比

0001 = 1:2 后分频比

0010 = 1:3 后分频比

0011 = 1:4 后分频比

0100 = 1:5 后分频比

0101 = 1:6 后分频比

0110 = 1:7 后分频比

0111 = 1:8 后分频比

1000 = 1:9 后分频比

1001 = 1:10 后分频比

1010 = 1:11 后分频比

1011 = 1:12 后分频比

1100 = 1:13 后分频比

1101 = 1:14 后分频比

1110 = 1:15 后分频比

1111 = 1:16 后分频比

bit 2

**TMRxON:** Timerx 使能位

1 = 使能 Timerx

0 = 禁止 Timerx

bit 1-0

**TxCKPS<1:0>:** Timer2 类型的时钟预分频值选择位

00 = 预分频值为 1

01 = 预分频值为 4

10 = 预分频值为 16

11 = 预分频值为 64

# PIC16(L)F1933

表 22-1: 与 TIMER2/4/6 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
CCP2CON	P2M<1:0>		DC2B<1:0>		CCP2M<3:0>				216
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92
PR2	Timer2 模块周期寄存器								189*
PR4	Timer4 模块周期寄存器								189*
PR6	Timer6 模块周期寄存器								189*
T2CON	—	T2OUTPS<3:0>				TMR2ON	T2CKPS<1:0>		191
T4CON	—	T4OUTPS<3:0>				TMR4ON	T4CKPS<1:0>		191
T6CON	—	T6OUTPS<3:0>				TMR2ON	T6CKPS<1:0>		191
TMR2	8 位 TMR2 寄存器的保持寄存器								189*
TMR4	8 位 TMR4 寄存器的保持寄存器 <sup>(1)</sup>								189*
TMR6	8 位 TMR6 寄存器的保持寄存器 <sup>(1)</sup>								189*

图注: — = 未实现单元, 读为 0。Timer2 模块不使用阴影单元。

\* 该页提供寄存器信息。



## 23.0 捕捉 / 比较 / PWM 模块

捕捉 / 比较 / PWM 模块是允许用户定时和控制不同事件并产生脉宽调制 (Pulse-Width Modulation, PWM) 信号的外设。在捕捉模式下, 该外设能对事件的持续时间计时。比较模式允许用户在达到预先设定的定时时间后触发一个外部事件。PWM 模式可产生频率和占空比都可变化的脉宽调制信号。

本器件系列包含 3 个增强型捕捉 / 比较 / PWM 模块 (ECCP1、ECCP2 和 ECCP3) 和 2 个标准捕捉 / 比较 / PWM 模块 (CCP4 和 CCP5)。

所有五个 CCP 模块 (ECCP1、ECCP2、ECCP3、CCP4 和 CCP5) 的捕捉和比较功能都是相同的。CCP 模块之间的惟一区别是脉宽调制 (PWM) 功能。模块 CCP4 和 CCP5 的标准 PWM 功能是相同的。在 CCP 模块 ECCP1、ECCP2 和 ECCP3 中, 各个模块的增强型 PWM 功能稍有不同。全桥 ECCP 模块有四个可用 I/O 引脚, 而半桥 ECCP 模块只有两个可用 I/O 引脚。更多信息, 请参见表 23-1。

- 注 1:** 在带有不止一个 CCP 模块的器件中, 密切注意使用的寄存器名称非常重要。模块缩略词后的数字用于区别不同的模块。例如, CCP1CON 和 CCP2CON 控制两个完全不同的 CCP 模块的相同操作。
- 注 2:** 在本节中, 在其任何工作模式下对 CCP 模块的通称同样适用于 ECCP1、ECCP2、ECCP3、CCP4 和 CCP5。在需要时, 寄存器名称、模块信号、I/O 引脚和位名称也可以使用通用标识“x”来表示通过数字区别特定模块。

**表 23-1: PWM 资源**

器件名称	ECCP1	ECCP2	ECCP3	CCP4	CCP5
PIC16(L)F1933	增强型 全桥 PWM	增强型 半桥 PWM	增强型 半桥 PWM	标准 PWM	标准 PWM

## 23.1 捕捉模式

对于 CCP 模块 ECCP1、ECCP2、ECCP3、CCP4 和 CCP5，本节中描述的捕捉模式功能都可用，并且相同。

捕捉模式使用 16 位定时器 Timer1 资源。CCPx 引脚上发生事件时，16 位 CCPRxH:CCPRxL 寄存器对分别捕捉并存储 TMR1H:TMR1L 寄存器对的 16 位值。这些事件被定义为以下四种之一，并由 CCPxCON 寄存器中的 CCPxM<3:0> 位配置：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

当捕捉发生时，PIRx 寄存器的中断请求标志位 CCPxIF 置 1。必须用软件将该中断标志清零。如果在读取 CCPRxH/CCPRxL 寄存器对中的值之前发生了另一次捕捉，那么之前捕捉的值将会被新值覆盖。

图 23-1 给出了捕捉操作的简化框图。

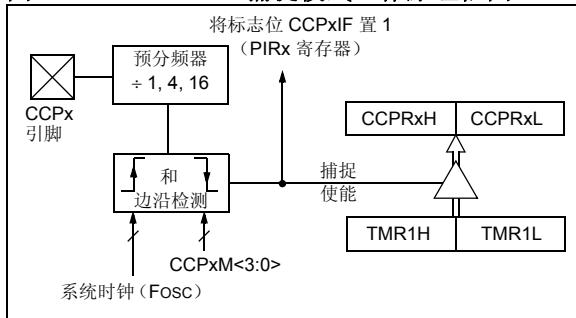
### 23.1.1 CCP 引脚配置

在捕捉模式下，应该通过将相应的 TRIS 控制位置 1 把 CCPx 引脚配置为输入引脚。

而且，可使用 APFCON 寄存器将 CCPx 引脚功能转移到备用引脚。更多详细信息，请参见第 12.1 节“备用引脚功能”。

**注：** 如果将 CCPx 引脚配置为输出引脚，对该端口的写操作可能引发一次捕捉事件。

图 23-1: 捕捉模式工作原理框图



### 23.1.2 TIMER1 模式资源

欲使 CCP 模块使用捕捉功能，Timer1 必须工作在定时器模式或同步计数器模式下。在异步计数器模式下，可能无法进行捕捉操作。

有关配置 Timer1 的更多信息，请参见第 21.0 节“带门控的 Timer1 模块”。

### 23.1.3 软件中断模式

当捕捉模式改变时，可能会产生误捕捉中断。用户应该保持 PIRx 寄存器中的 CCPxIE 中断允许位为零以避免产生误中断。而且应该在工作模式发生任何改变之后清零 PIRx 寄存器中的 CCPxIF 中断标志位。

**注：** 在捕捉模式下，不应使用系统时钟 (Fosc) 作为 Timer1 的时钟源。欲使捕捉模式能够识别 CCPx 引脚上的触发事件，Timer1 的时钟必须来自指令时钟 (Fosc/4) 或外部时钟源。

### 23.1.4 CCP 预分频器

有 4 种预分频器设置，由 CCPxCON 寄存器中的 CCPxM<3:0> 位指定。每当 CCP 模块被关闭或者没有处于捕捉模式时，预分频器计数器就会被清零。任何复位都将清零预分频器计数器。

从一个捕捉预分频比切换到另一个捕捉预分频比不会将预分频器清零但可能会产生误中断。要避免出现这种不期望的操作，应在改变预分频比之前通过清零 CCPxCON 寄存器来关闭模块。例 23-1 演示了用以执行此功能的代码。

例 23-1: 切换捕捉预分频比

```
BANKSEL CCPxCON    ;Set Bank bits to point
                    ;to CCPxCON
CLRWF  CCPxCON      ;Turn CCP module off
MOVLW  NEW_CAPT_PS ;Load the W reg with
                    ;the new prescaler
MOVWF  CCPxCON      ;move value and CCP ON
MOVLW  CCPxCON      ;Load CCPxCON with this
                    ;value
```

### 23.1.5 休眠期间的捕捉

捕捉模式依靠 Timer1 模块才能正确工作。可选用两种方式在捕捉模式下驱动 Timer1 模块：由指令时钟 (Fosc/4) 或由外部时钟源驱动。

Timer1 由 Fosc/4 提供时钟时，Timer1 在休眠期间不进行递增操作。当器件从休眠模式唤醒时，Timer1 将从其之前状态继续工作。Timer1 由外部时钟源提供时钟时，捕捉模式将在休眠期间继续工作。

## 23.1.6 备用引脚位置

本模块包括的 I/O 引脚可通过备用引脚功能寄存器 APFCON 转移到其他位置。要确定哪个引脚可以转移以及复位时其默认位置，请参见第 12.1 节“备用引脚功能”获取更多信息。

**表 23-2: 与捕捉相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
CCPxCON	PxM<1:0> <sup>(1)</sup>		DCxB<1:0>		CCPxM<3:0>				216
CCPRxL	捕捉 / 比较 / PWM 寄存器 x 的低字节 (LSB)								194
CCPRxH	捕捉 / 比较 / PWM 寄存器 x 的高字节 (MSB)								194
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		186
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								181
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								181
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129

图注: — = 未实现单元, 读为 0。捕捉模式不使用阴影单元。

注 1: 仅适用于 ECCP 模块。

# PIC16(L)F1933

## 23.2 比较模式

对于 CCP 模块 ECCP1、ECCP2、ECCP3、CCP4 和 CCP5，本节中描述的比较模式功能都可用，并且相同。

比较模式使用 16 位定时器 Timer1 资源。CCPRxH:CCPRxL 寄存器对的 16 位值不断与 TMR1H:TMR1L 寄存器对的 16 位值进行比较。发生匹配时，可能会发生以下某一事件：

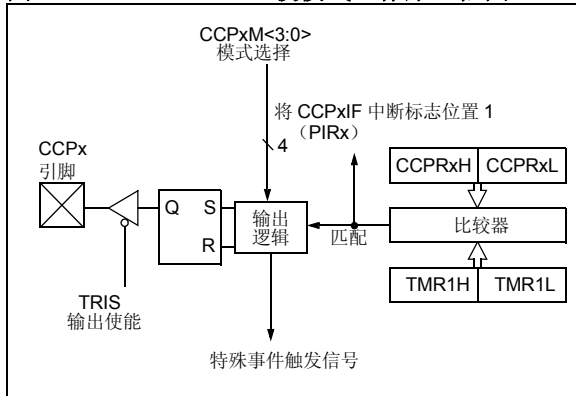
- CCPx 的输出电平翻转
- CCPx 输出高电平
- CCPx 输出低电平
- 产生特殊事件触发信号
- 产生软件中断

引脚的动作由 CCPxCON 寄存器的 CCPxM<3:0> 控制位的值决定。同时，中断标志位 CCPxIF 置 1。

所有比较模式都可以产生中断。

图 23-2 给出了比较操作的简化框图。

图 23-2: 比较模式工作原理框图



### 23.2.1 CCP 引脚配置

用户必须通过清零相应的 TRIS 位将 CCPx 引脚配置为输出引脚。

而且，可使用 APFCON 寄存器将 CCPx 引脚功能转移到备用引脚。更多详细信息，请参见第 12.1 节“备用引脚功能”。

**注：** 清零 CCPxCON 寄存器会将 CCPx 比较输出锁存器强制为默认的低电平状态。这不是端口 I/O 数据锁存器。

### 23.2.2 TIMER1 模式资源

在比较模式下，Timer1 必须运行在定时器模式或同步计数器模式下。在异步计数器模式下，可能无法进行比较操作。

有关配置 Timer1 的更多信息，请参见第 21.0 节“带门控的 Timer1 模块”。

**注：** 在比较模式下，不应使用系统时钟 (Fosc) 作为 Timer1 的时钟源。欲使比较模式能够识别 CCPx 引脚上的触发事件，Timer1 的时钟必须来自指令时钟 (Fosc/4) 或外部时钟源。

### 23.2.3 软件中断模式

当选择了产生软件中断模式 (CCPxM<3:0> = 1010) 时，CCPx 模块不会控制 CCPx 引脚 (见 CCPxCON 寄存器)。

### 23.2.4 特殊事件触发信号

当选择了特殊事件触发模式 (CCPxM<3:0> = 1011) 时，CCPx 模块将执行如下操作：

- 复位 Timer1
- 如果使能了 ADC，还将启动 ADC 转换

在此模式下，CCPx 模块不会控制 CCPx 引脚。

当 TMR1H/TMR1L 寄存器对与 CCPRxH/CCPRxL 寄存器对匹配时，CCP 会立即产生特殊事件触发信号输出。TMR1H/TMR1L 寄存器对将在 Timer1 时钟的下一个上升沿复位。如果使能了 A/D 模块，特殊事件触发信号输出将启动 A/D 转换。这使得 CCPRxH/CCPRxL 寄存器对实际上成为 Timer1 的 16 位可编程周期寄存器。

表 23-3: 特殊事件触发器

器件	CCPx/ECCPx
PIC16(L)F1933	CCP5

更多信息，请参见第 15.2.5 节“特殊事件触发器”。

- 注 1：** 来自 CCP 模块的特殊事件触发信号不会将 PIR1 寄存器的中断标志位 TMR1IF 置 1。
- 注 2：** 在产生特殊事件触发信号的时钟边沿和产生 Timer1 复位的时钟边沿之间改变 CCPRxH/CCPRxL 寄存器对的内容可清除匹配条件，从而阻止复位发生。

## 23.2.5 休眠期间的比较

比较模式依靠系统时钟（Fosc）才能正确工作。由于Fosc在休眠模式下被关闭，所以比较模式在休眠期间无法正确工作。

## 23.2.6 备用引脚位置

本模块包括的 I/O 引脚可通过备用引脚功能寄存器 APFCON 转移到其他位置。要确定哪个引脚可以转移以及复位时其默认位置，请参见第 12.1 节“备用引脚功能”获取更多信息。

**表 23-4: 与比较相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
CCPxCON	PxM<1:0> <sup>(1)</sup>		DCxB<1:0>		CCPxM<3:0>				216
CCPRxL	捕捉 / 比较 / PWM 寄存器 x 的低字节 (LSB)								194
CCPRxH	捕捉 / 比较 / PWM 寄存器 x 的高字节 (MSB)								194
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFE	TMR0IF	INTF	IOCFIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185
T1GCON	TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/DONE	T1GVAL	T1GSS<1:0>		186
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								181
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								181
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129

**图注:** — = 未实现单元，读为 0。比较模式不使用阴影单元。

**注 1:** 仅适用于 ECCP 模块。

## 23.3 PWM 概述

脉宽调制 (PWM) 是一种通过在完全导通和完全关断状态之间快速切换来为负载提供功率的机制。PWM 信号类似于方波, 信号的高电平部分被视为导通状态, 信号的低电平部分被视为关断状态。高电平部分也称为脉冲宽度, 以步长为单位, 随时间变化而变化。提供的步长数越多 (即增长脉冲宽度), 为负载提供的功率也就越大。提供的步长数越少 (即缩短脉冲宽度), 提供的功率也就越小。PWM 周期定义为一个完整周期的持续时间, 或导通和关断时间的总和。

PWM 分辨率定义了单个 PWM 周期中最多出现的步长数。分辨率越高, 对脉冲宽度时间的控制就越精确, 从而也能更加精确地控制提供给负载的功率。

术语占空比描述了导通时间与关断时间的比例, 以百分比表示, 0% 表示完全关断, 100% 表示完全导通。占空比越低, 提供的功率越小; 占空比越高, 提供的功率越大。

图 23-3 给出了 PWM 信号的典型波形。

### 23.3.1 标准 PWM 操作

对于 CCP 模块 ECCP1、ECCP2、ECCP3、CCP4 和 CCP5, 本节中描述的标准 PWM 功能都可用, 并且相同。

标准 PWM 模式会在 CCPx 引脚上产生高达 10 位分辨率的脉宽调制 (PWM) 信号。由下列寄存器控制周期、占空比和分辨率:

- PRx 寄存器
- TxCON 寄存器
- CCPRxL 寄存器
- CCPxCON 寄存器

图 23-4 给出了 PWM 操作的简化框图。

- 注 1:** 必须将 CCPx 引脚对应的 TRIS 位清零, 以使能该引脚上的 PWM 输出。
- 2:** 清零 CCPxCON 寄存器将会放弃对 CCPx 引脚的控制权。

图 23-3: CCP PWM 输出信号

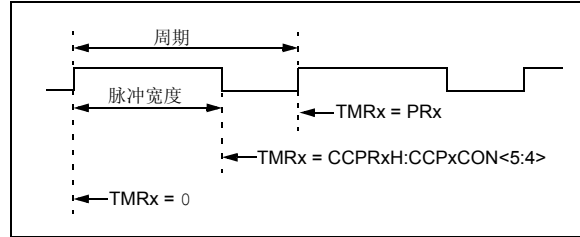
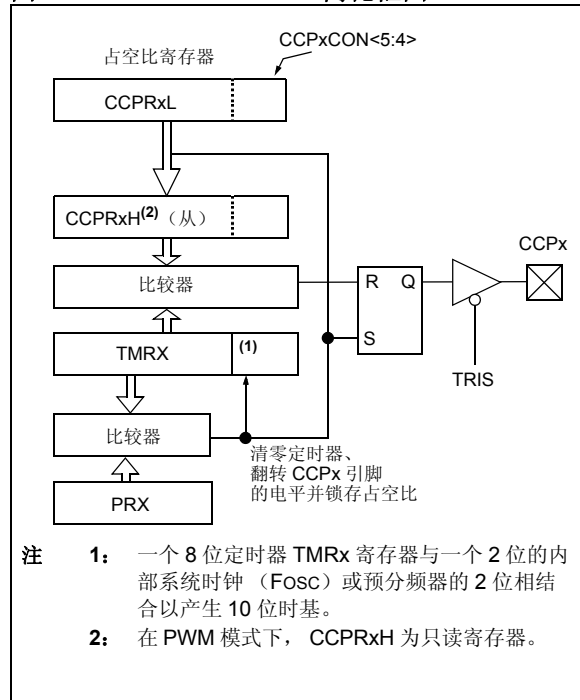


图 23-4: PWM 简化框图



## 23.3.2 设置 PWM 工作模式

当配置 CCP 模块使之工作于标准 PWM 模式时应遵循以下步骤：

1. 通过将相应的 TRIS 位置 1 禁止 CCPx 引脚输出驱动器。
2. 将 PWM 周期值装入 PRx 寄存器。
3. 通过将合适的值装入 CCPxCON 寄存器来配置 CCP 模块使之工作于 PWM 模式。
4. 将 PWM 占空比值装入 CCPRxL 寄存器和 CCPxCON 寄存器的 DCxBx 位。
5. 配置并启动 Timer2/4/6:
  - 通过设置 CCPTMRSx 寄存器中的 CxTSEL<1:0> 位，选择用于生成 PWM 的 Timer2/4/6 资源。
  - 清零 PIRx 寄存器的 TMRxIF 中断标志位。请参见下面的“注”。
  - 使用定时器预分频值配置 TxCON 寄存器的 TxCKPS 位。
  - 通过将 TxCON 寄存器的 TMRxON 位置 1 来使能定时器。
6. 使能 PWM 输出引脚:
  - 等待定时器溢出并且 PIRx 寄存器的 TMRxIF 位置 1。请参见下面的“注”。
  - 通过将相应的 TRIS 位清零使能 CCPx 引脚输出驱动器。

**注：** 为了在第一个 PWM 输出中发送完整的占空比和周期，设置过程必须包括上述步骤。如果不要要求在第一个输出中发送完整的 PWM 信号，那么可忽略第 6 步。

## 23.3.3 TIMER2/4/6 定时器资源

PWM 标准模式使用 8 位 Timer2/4/6 定时器资源之一指定 PWM 周期。

配置 CCPTMRSx 寄存器的 CxTSEL<1:0> 位可以选择要使用的 Timer2/4/6 定时器。

## 23.3.4 PWM 周期

PWM 周期由 Timer2/4/6 的 PRx 寄存器指定。使用公式 23-1 中的公式可以计算 PWM 周期。

### 公式 23-1: PWM 周期

$$PWM \text{ 周期} = [(PRx) + 1] \cdot 4 \cdot T_{OSC} \cdot (TMRx \text{ 预分频值})$$

**注 1:**  $T_{OSC} = 1/F_{OSC}$

当 TMRx 等于 PRx 时，在下一个递增计数周期中将发生以下三个事件：

- TMRx 清零。
- CCPx 引脚置 1。（例外情况：如果 PWM 占空比 = 0%，引脚将不会置 1。）
- PWM 占空比从 CCPRxL 锁存到 CCPRxH。

**注：** 定时器后分频器（见第 22.1 节“Timer2/4/6 工作原理”）不用于确定 PWM 频率。

## 23.3.5 PWM 占空比

可通过将一个 10 位值写入以下多个寄存器来指定 PWM 占空比：CCPRxL 寄存器和 CCPxCON 寄存器中的 DCxB<1:0> 位。CCPRxL 保存高 8 位，而 CCPxCON 寄存器中的 DCxB<1:0> 位保存低 2 位。可以在任何时候写入 CCPRxL 以及 CCPxCON 寄存器的 DCxB<1:0> 位。但直到周期结束（即，PRx 和 TMRx 寄存器之间发生匹配）后，占空比的值才被锁存到 CCPRxH 中。在 PWM 模式下，CCPRxH 是只读寄存器。

公式 23-2 用于计算 PWM 脉冲宽度。

公式 23-3 用于计算 PWM 占空比。

### 公式 23-2: 脉冲宽度

$$脉冲宽度 = (CCPRxL:CCPxCON<5:4>) \cdot T_{OSC} \cdot (TMRx \text{ 预分频值})$$

### 公式 23-3: 占空比

$$占空比 = \frac{(CCPRxL:CCPxCON<5:4>)}{4(PRx + 1)}$$

CCPRxH 寄存器和一个 2 位的内部锁存器用于为 PWM 占空比提供双重缓冲。这种双重缓冲极其重要，可以避免在 PWM 工作过程中产生毛刺。

一个 8 位定时器 TMRx 寄存器与一个 2 位的内部系统时钟 (Fosc) 或预分频器的 2 位相结合以产生 10 位时基。如果将 Timer2/4/6 预分频比设置为 1:1，则使用系统时钟。

当 10 位时基的值与 CCPRxH 和 2 位锁存器的值匹配时，会使 CCPx 引脚输出低电平（见图 23-4）。

# PIC16(L)F1933

## 23.3.6 PWM 分辨率

分辨率决定在给定周期内的可用占空比数。例如，10 位分辨率将产生 1024 个离散的占空比，而 8 位分辨率将产生 256 个离散的占空比。

当 PRx 为 255 时，PWM 的最大分辨率为 10 位。如公式 23-4 所示，分辨率是 PRx 寄存器值的函数。

### 公式 23-4: PWM 分辨率

$$\text{分辨率} = \frac{\log[4(PR_x + 1)]}{\log(2)} \text{ 位}$$

**注：** 如果脉宽值大于周期值，则指定的 PWM 引脚电平将保持不变。

表 23-5: PWM 频率与分辨率示例 (Fosc = 32 MHz)

PWM 频率	1.95 kHz	7.81 kHz	31.25 kHz	125 kHz	250 kHz	333.3 kHz
定时器预分频值 (1、4 或 16)	16	4	1	1	1	1
PRx 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最高分辨率 (位)	10	10	10	8	7	6.6

表 23-6: PWM 频率与分辨率示例 (Fosc = 20 MHz)

PWM 频率	1.22 kHz	4.88 kHz	19.53 kHz	78.12 kHz	156.3 kHz	208.3 kHz
定时器预分频值 (1、4 或 16)	16	4	1	1	1	1
PRx 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最高分辨率 (位)	10	10	10	8	7	6.6

表 23-7: PWM 频率与分辨率示例 (Fosc = 8 MHz)

PWM 频率	1.22 kHz	4.90 kHz	19.61 kHz	76.92 kHz	153.85 kHz	200.0 kHz
定时器预分频值 (1、4 或 16)	16	4	1	1	1	1
PRx 值	0x65	0x65	0x65	0x19	0x0C	0x09
最高分辨率 (位)	8	8	8	6	5	5



### 23.3.7 休眠模式下的工作

在休眠模式下，TMRx 寄存器将不会递增，并且模块的状态将保持不变。如果 CCPx 引脚有输出，它将继续保持该输出值不变。当器件被唤醒时，TMRx 将从之前的状态继续工作。

### 23.3.8 系统时钟频率的改变

PWM 频率是由系统时钟频率产生的。系统时钟频率发生任何变化都会使 PWM 频率发生变化。其他详细信息，请参见第 5.0 节“振荡器模块（带故障保护时钟监视器）”。

### 23.3.9 复位的影响

任何复位都会将所有端口强制为输入模式，并强制 CCP 寄存器进入复位状态。

### 23.3.10 备用引脚位置

本模块包括的 I/O 引脚可通过备用引脚功能寄存器 APFCON 转移到其他位置。要确定哪个引脚可以转移以及复位时其默认位置，请参见第 12.1 节“备用引脚功能”获取更多信息。

**表 23-8: 与标准 PWM 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面	
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118	
CCPxCON	PxM<1:0> <sup>(1)</sup>		DCxB<1:0>		CCPxM<3:0>					216
CCPTMRS0	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>			217
CCPTMRS1	—	—	—	—	—	—	C5TSEL<1:0>			217
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86	
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87	
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88	
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89	
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90	
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91	
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92	
PRx	Timer2/4/6 周期寄存器								189*	
TxCON	—	TxOUTPS<3:0>				TMRxON	TxCKPS<:0>1			191
TMRx	Timer2/4/6 模块寄存器								189	
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120	
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125	
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129	

图注： — = 未实现单元，读为 0。PWM 不使用阴影单元。

注 1: 仅适用于 ECCP 模块。

\* 该页提供寄存器信息。

# PIC16(L)F1933

## 23.4 增强型 PWM 模式

对于 CCP 模块 ECCP1、ECCP2 和 ECCP3，本节中描述的增强型 PWM 功能都可用，并且注明了模块之间的所有区别。

增强型 PWM 模式可在最多 4 个不同的输出引脚上产生分辨率最高为 10 位的脉宽调制 (PWM) 信号。由下列寄存器控制周期、占空比和分辨率：

- PRx 寄存器
- TxCON 寄存器
- CCPRxL 寄存器
- CCPxCON 寄存器

ECCP 模块具有以下额外的 PWM 寄存器，分别控制自动关闭、自动重启、死区延时和 PWM 脉冲转向模式：

- CCPxAS 寄存器
- PSTRxCON 寄存器
- PWMxCON 寄存器

增强型 PWM 模块可以产生以下五种 PWM 输出模式：

- 单 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式
- 具有 PWM 脉冲转向模式的单 PWM

要选择增强型 PWM 输出模式，必须相应地配置 CCPxCON 寄存器的 PxM 位。

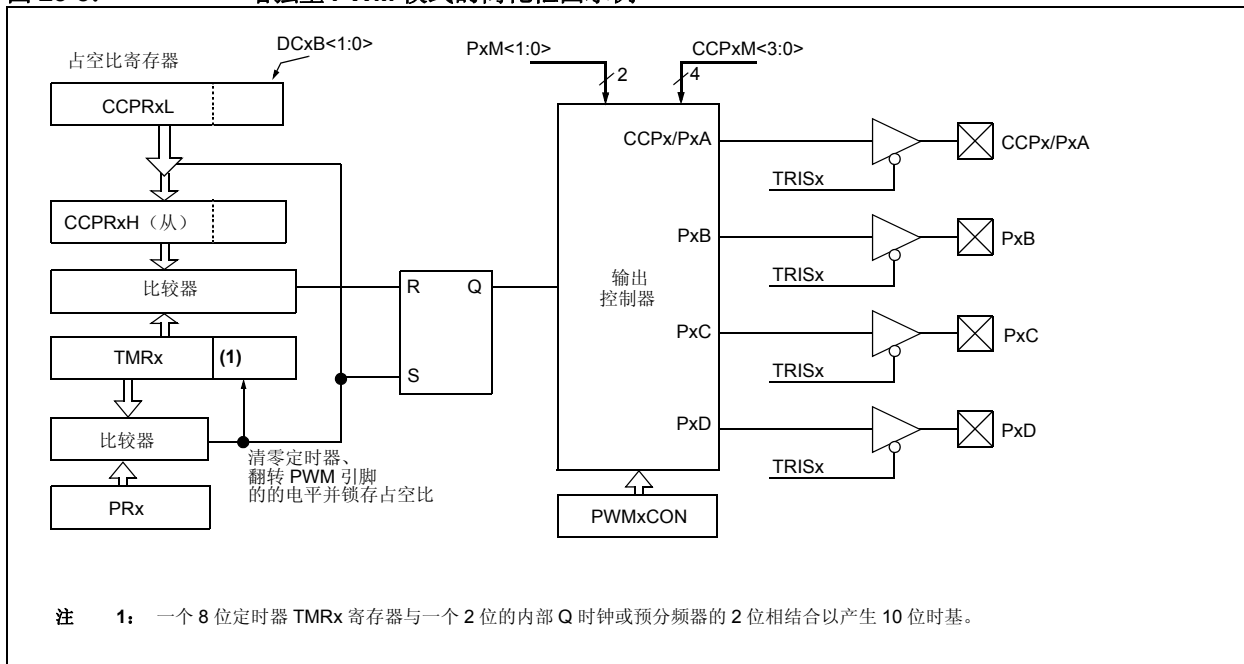
PWM 输出可与 I/O 引脚复用且标记为 PxA、PxB、PxC 和 PxD。PWM 引脚的极性是可配置的，可通过相应地设置 CCPxCON 寄存器中的 CCPxM 位来选择 PWM 引脚的极性。

图 23-5 给出了增强型 PWM 模块的简化框图示例。

表 23-9 说明了各种增强型 PWM 模式的引脚分配。

- 注**
- 1: 必须将 CCPx 引脚对应的 TRIS 位清零，以使能该引脚上的 PWM 输出。
  - 2: 清零 CCPxCON 寄存器将会放弃对 CCPx 引脚的控制权。
  - 3: 增强型 PWM 模式中未使用的任何引脚可提供备用引脚功能（如果适用）。
  - 4: 要防止 PWM 首次使能时产生不完整的波形，ECCP 模块在产生 PWM 信号之前须等待一段时间直到开始新的 PWM 周期。

图 23-5: 增强型 PWM 模式的简化框图示例

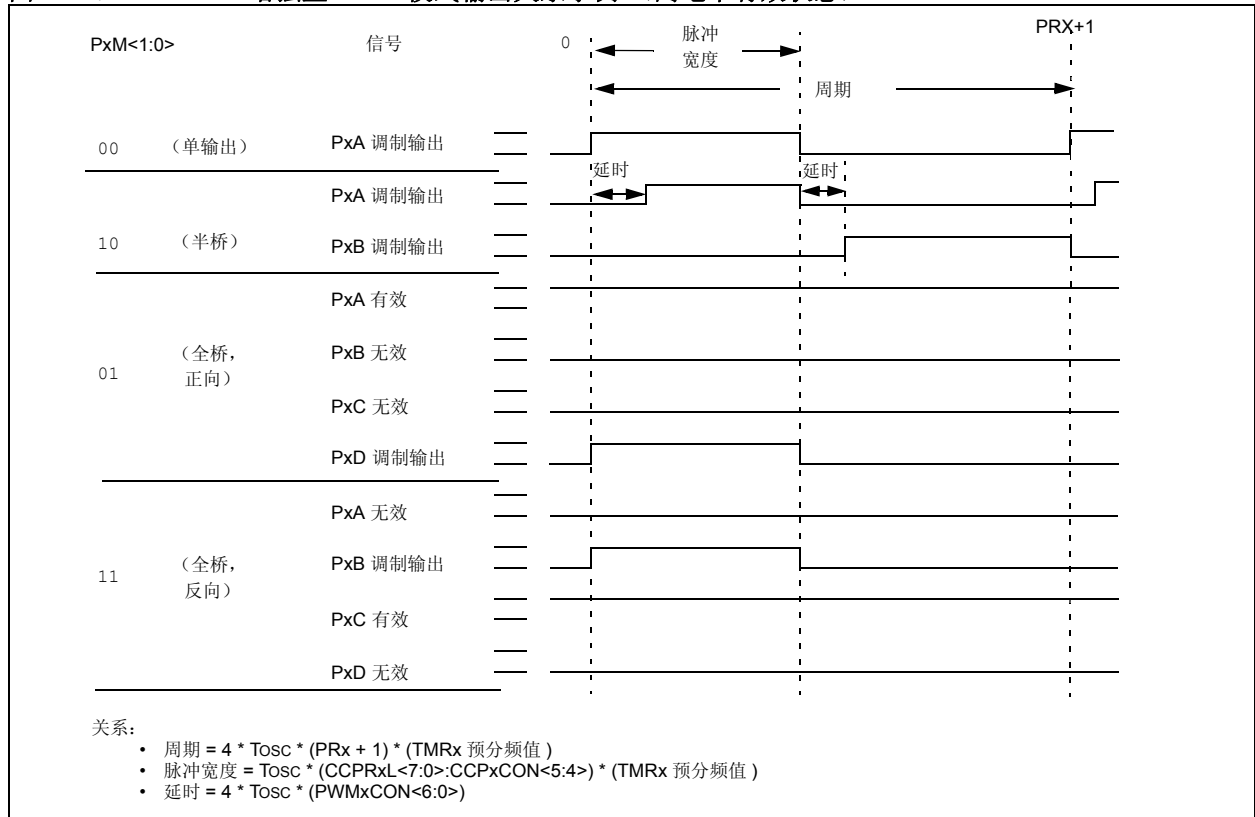


**表 23-9: 各种增强型 PWM 模式的引脚分配示例**

ECCP 模式	PxM<1:0>	CCPx/PxA	PxB	PxC	PxD
单模式	00	有 <sup>(1)</sup>	有 <sup>(1)</sup>	有 <sup>(1)</sup>	有 <sup>(1)</sup>
半桥模式	10	有	有	无	无
全桥模式, 正向	01	有	有	有	有
全桥模式, 反向	11	有	有	有	有

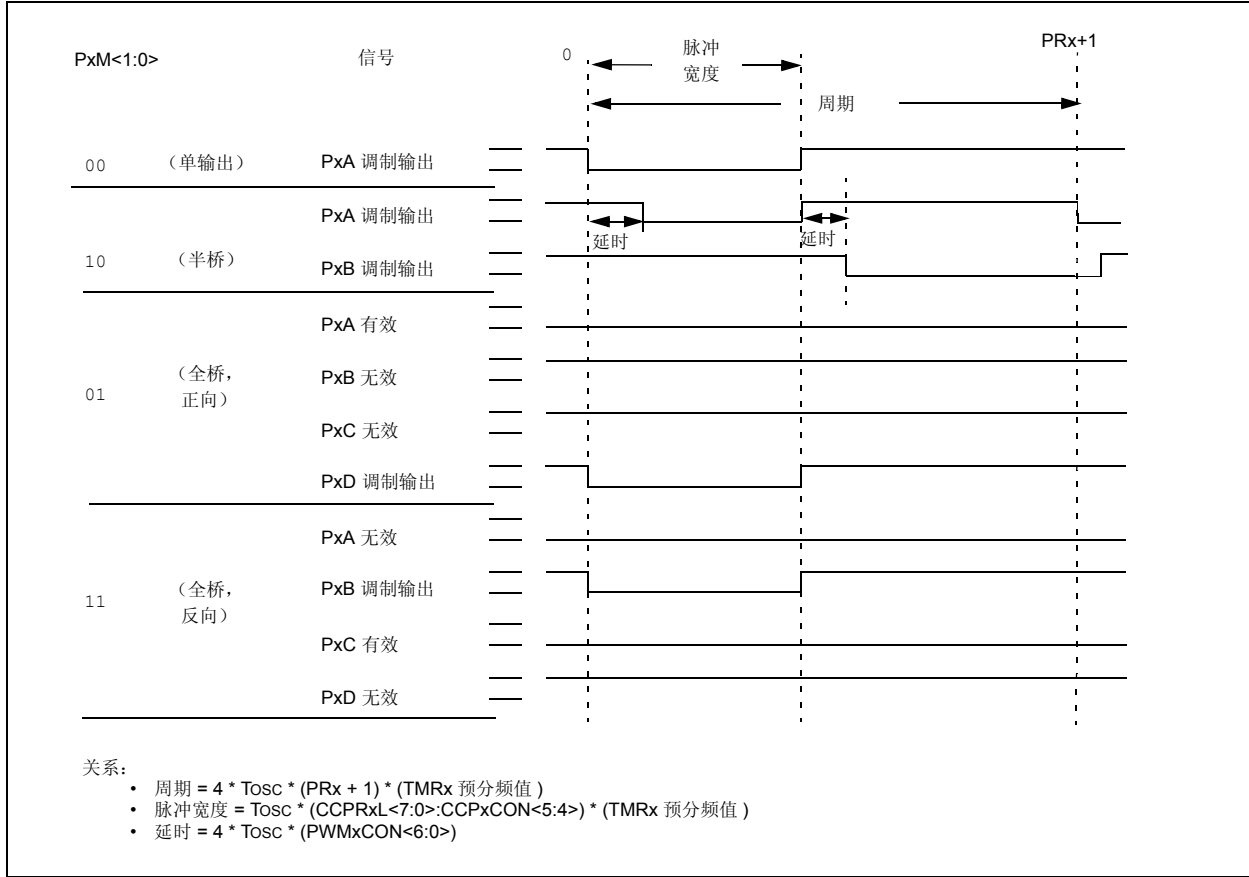
注 1: PWM 脉冲转向可使能单模式下的输出。

**图 23-6: 增强型 PWM 模式输出关系示例 (高电平有效状态)**



# PIC16(L)F1933

图 23-7: 增强型 PWM 模式输出关系示例 (低电平有效状态)



## 23.4.1 半桥模式

在半桥模式下，2 个引脚用作输出端来驱动推挽式负载。在 CCPx/PxA 引脚上输出 PWM 信号，而在 PxB 引脚上输出互补的 PWM 信号（见图 23-9）。该模式可用于半桥应用，如图 23-9 所示，也可用于使用 2 个 PWM 信号调制 4 个功率开关的全桥应用。

在半桥模式下，可使用可编程的死区延时来防止半桥功率器件中出现直通电流。PWMxCON 寄存器的 PDC<6:0> 位的值用来设置在输出被驱动为有效之前的指令周期数。如果该值大于占空比值，则相应的输出引脚在整个周期中都保持无效状态。更多关于死区延时操作的详细信息，请参见第 23.4.5 节“可编程死区延时模式”。

由于 PxA 和 PxB 输出与端口数据锁存器复用，所以必须将相应的 TRIS 位清零以将 PxA 和 PxB 配置为输出。

图 23-8: 半桥 PWM 输出示例

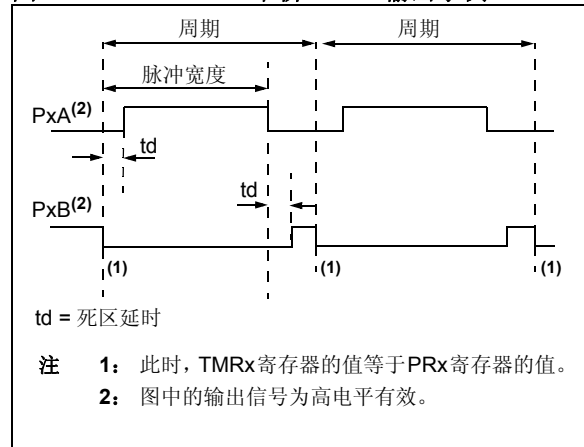
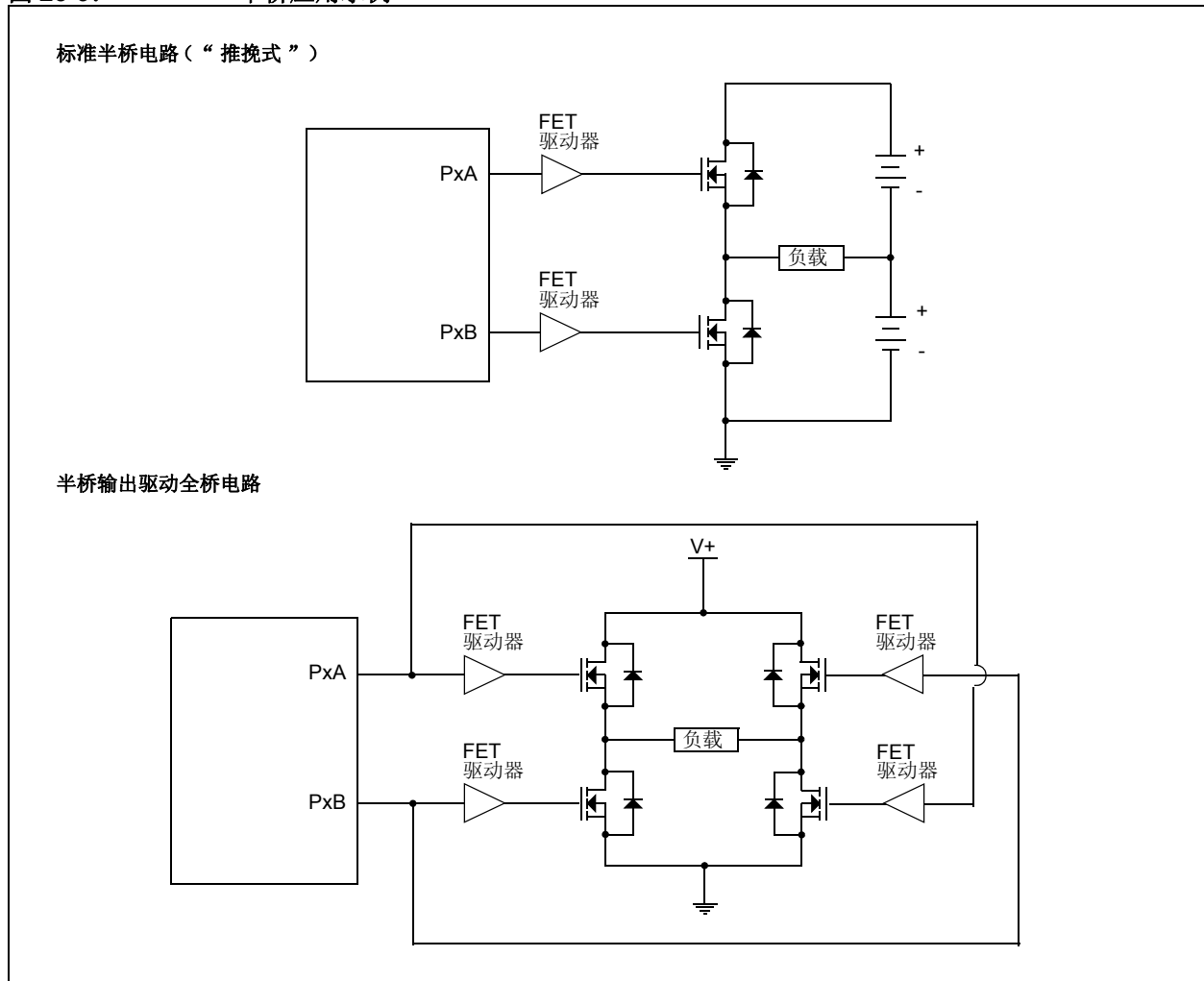


图 23-9: 半桥应用示例



# PIC16(L)F1933

## 23.4.2 全桥模式

在全桥模式下，所有4个引脚都用作输出引脚。图23-10给出了全桥应用的示例。

在正向模式下，引脚CCPx/PxA处于有效状态，引脚PxD调制输出，而PxB和PxC处于无效状态，如图23-11所示。

在反向模式下，PxC处于有效状态，引脚PxB调制输出，而PxA和PxD处于无效状态，如图23-11所示。

PxA、PxB、PxC和PxD输出与端口数据锁存器复用。必须清零相应的TRIS位以配置PxA、PxB、PxC和PxD为输出引脚。

图 23-10: 全桥应用示例

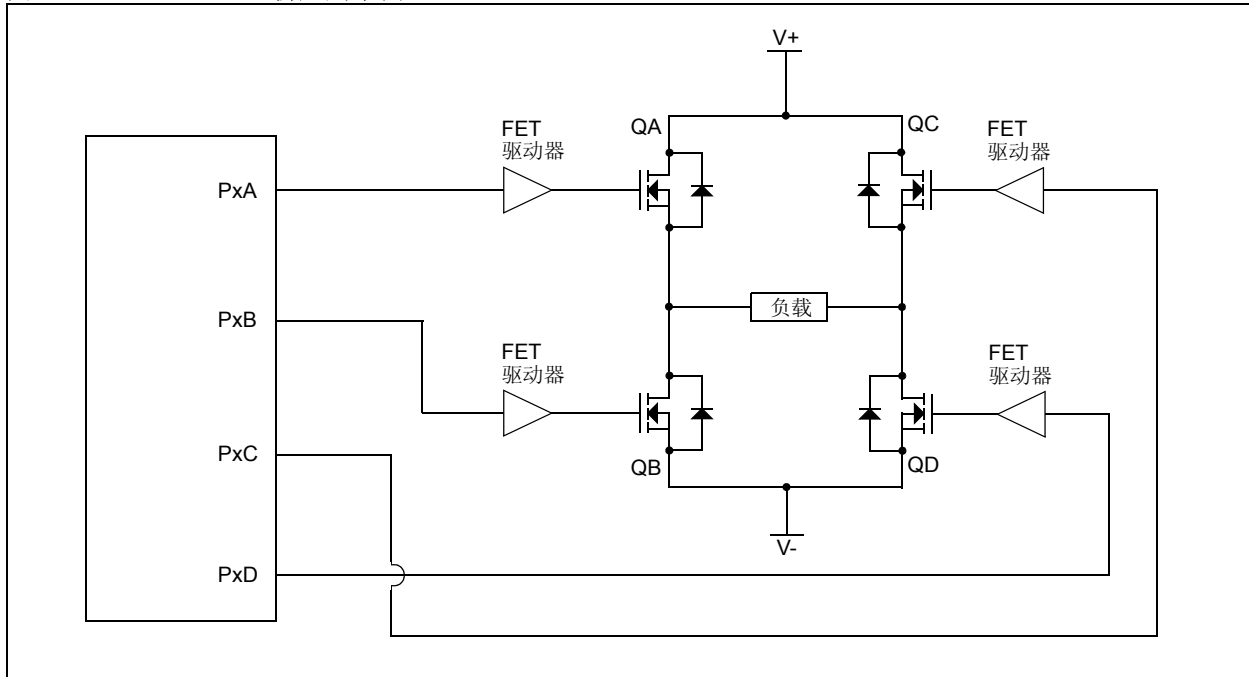
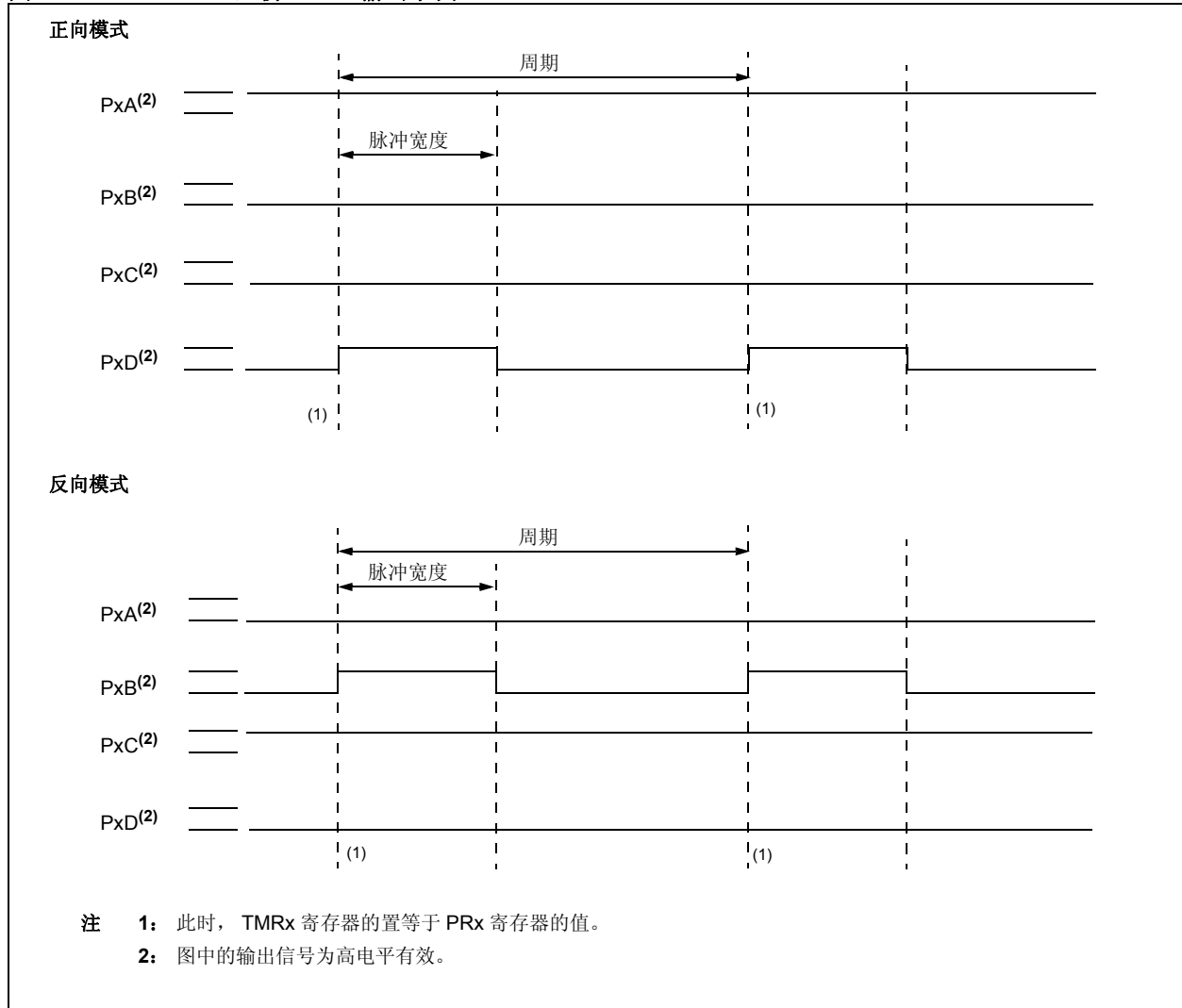


图 23-11: 全桥 PWM 输出示例



# PIC16(L)F1933

## 23.4.2.1 全桥模式下的方向更改

在全桥模式下，CCPxCON 寄存器的 PxM1 位允许用户控制方向的正反。当应用程序固件改变了这个方向控制位时，模块将在下一个 PWM 周期更改为新的方向。

可用软件通过改变 CCPxCON 寄存器的 PxM1 位来实现方向更改。在当前 PWM 周期结束之前，以下序列发生的时间为 4 个定时器周期：

- 将调制后的输出（PxB 和 PxD）置于无效状态。
- 将相应的未经调制的输出（PxA 和 PxC）切换为相反的方向。
- 在下一个周期开始时继续 PWM 调制。

关于此序列的说明，请参见图 23-12。

全桥模式不提供死区延时。由于一次只能调制一个输出，所以通常情况下不需要死区延时。然而也有一种需要死区延时的情况，这种情况出现在以下两个条件都为真时：

1. 当输出的占空比为 100% 或接近 100% 时，PWM 输出的方向改变。
2. 功率开关（包括功率器件和驱动器电路）的关断时间大于导通时间。

图 23-13 给出了占空比接近 100% 时，PWM 方向从正向变为反向的示例。在此示例中，在 t1 时刻输出 PxA 和 PxD 变为无效，而输出 PxC 变为有效。由于功率器件的关断时间大于导通时间，直通电流将流经功率器件 QC 和 QD（见图 23-10）长达“t”时间。当 PWM 方向从反向变为正向时，功率器件 QA 和 QB 上也会发生同样的现象。

如果应用需要在高占空比时更改 PWM 方向，则有两种方法可以避免出现直通电流：

1. 在更改方向之前的那个 PWM 周期缩小 PWM 占空比。
2. 使用可使开关元件的关断速度比导通速度快的开关驱动器。

可能还存在其他防止直通电流的方法。

图 23-12: PWM 方向更改示例

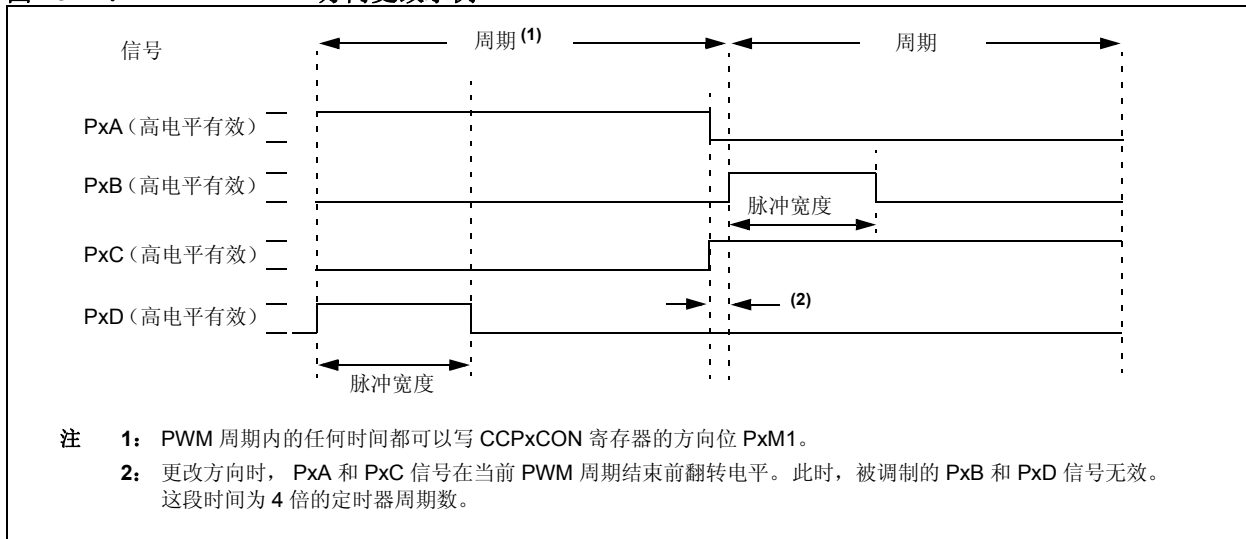
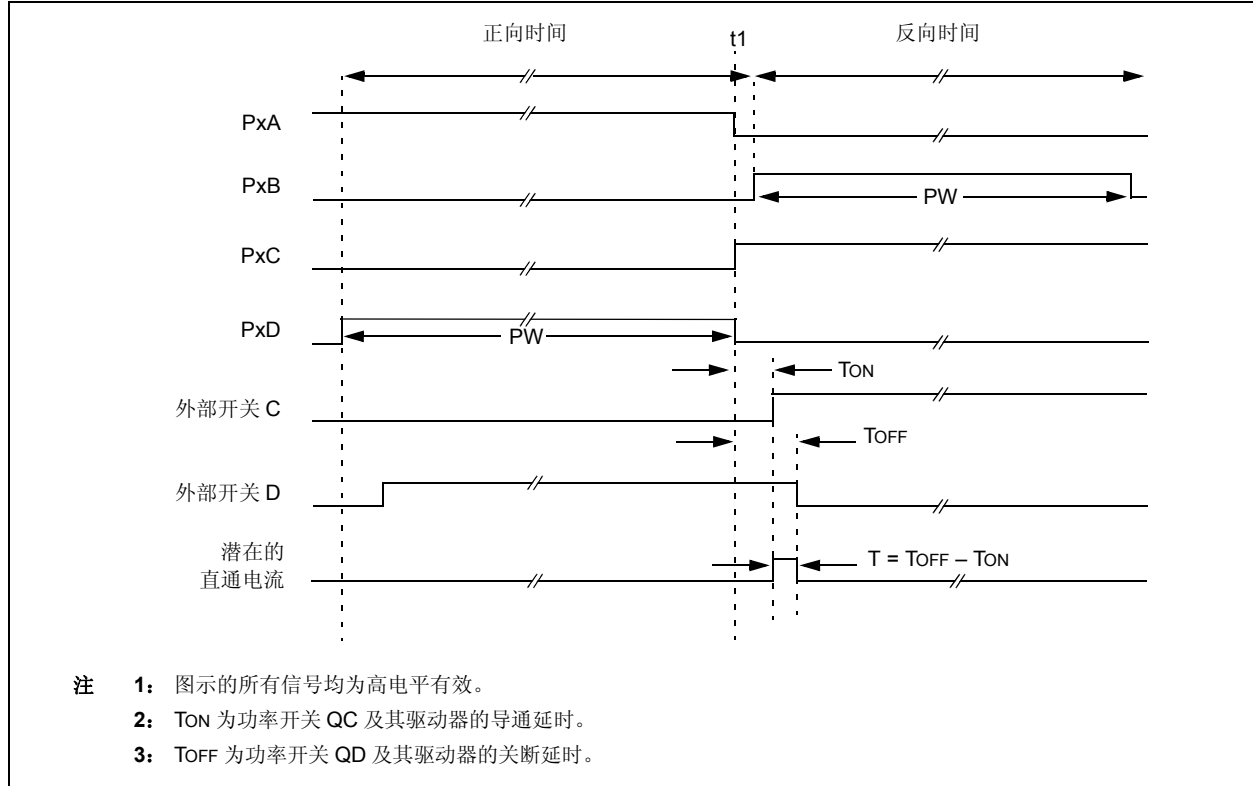




图 23-13: 占空比接近 100% 时 PWM 方向更改的示例



# PIC16(L)F1933

## 23.4.3 增强型 PWM 自动关闭模式

PWM 模式支持自动关闭模式，可在发生外部关闭事件时禁止 PWM 输出。自动关闭模式将 PWM 输出引脚置于预先设定的状态。此模式有助于防止不合适的 PWM 信号损坏应用。

使用 CCPxAS 寄存器的 CCPxAS<2:0> 位来选择自动关闭源。可由以下条件产生关闭事件：

- INT 引脚上的逻辑 0
- 比较器 (Cx) 输出上的逻辑 1

由 CCPxAS 寄存器的 CCPxASE (自动关闭事件状态) 位指示关闭状态。如果该位为 0，则 PWM 引脚正常工作。如果该位为 1，则 PWM 输出处于关闭状态。

当关闭事件发生时，会发生以下两种情况：

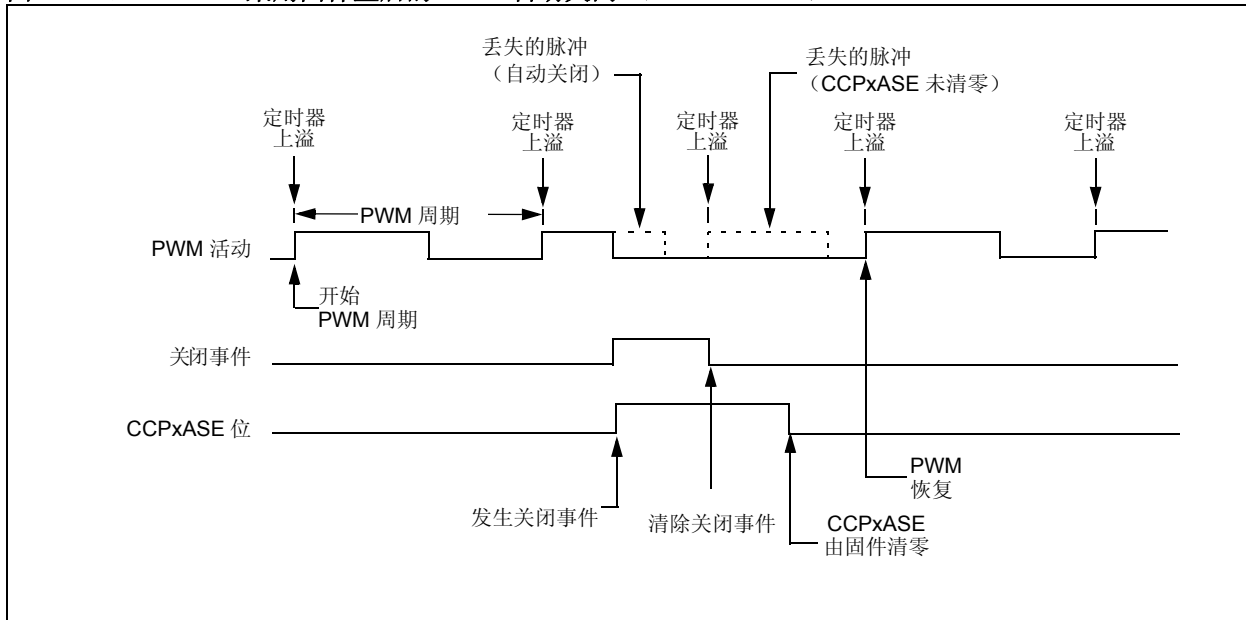
CCPxASE 位被置为 1。CCPxASE 保持置 1 状态直到由固件清零或发生自动重启为止 (见第 23.4.4 节“自动重启模式”)。

使能的 PWM 引脚会陆续被置于关闭状态。PWM 输出引脚被编组为引脚对 [PxA/PxC] 和 [PxB/PxD]。两个引脚对的状态分别由 CCPxAS 寄存器的 PSSxAC 和 PSSxBD 位确定。每个引脚对可以置于以下三种状态之一：

- 驱动为逻辑 1
- 驱动为逻辑 0
- 三态 (高阻态)

- 注 1:** 自动关闭条件是一个基于电平的信号，而不是基于边沿的信号。只要有电平，自动关闭条件就保持。
- 注 2:** 自动关闭条件保持时，禁止写 CCPxASE 位。
- 注 3:** 一旦清除了自动关闭条件且重新启动了 PWM (由固件重启或自动重启)，PWM 信号始终在下一个 PWM 周期开始时重启。
- 注 4:** 在由比较器输出引起的自动关闭事件或 INT 引脚事件之前，可以通过在固件中将 CCPxAS 寄存器的 CCPxASE 位设置为 1 来触发软件关闭。自动重启功能仅追踪由比较器输出引起的关闭或 INT 引脚事件的有效状态。如果此时启用，将立即清零此位并在下一个 PWM 周期开始时重新启动 ECCP 模块。

图 23-14: 采用固件重启的 PWM 自动关闭 (PXRSEN = 0)

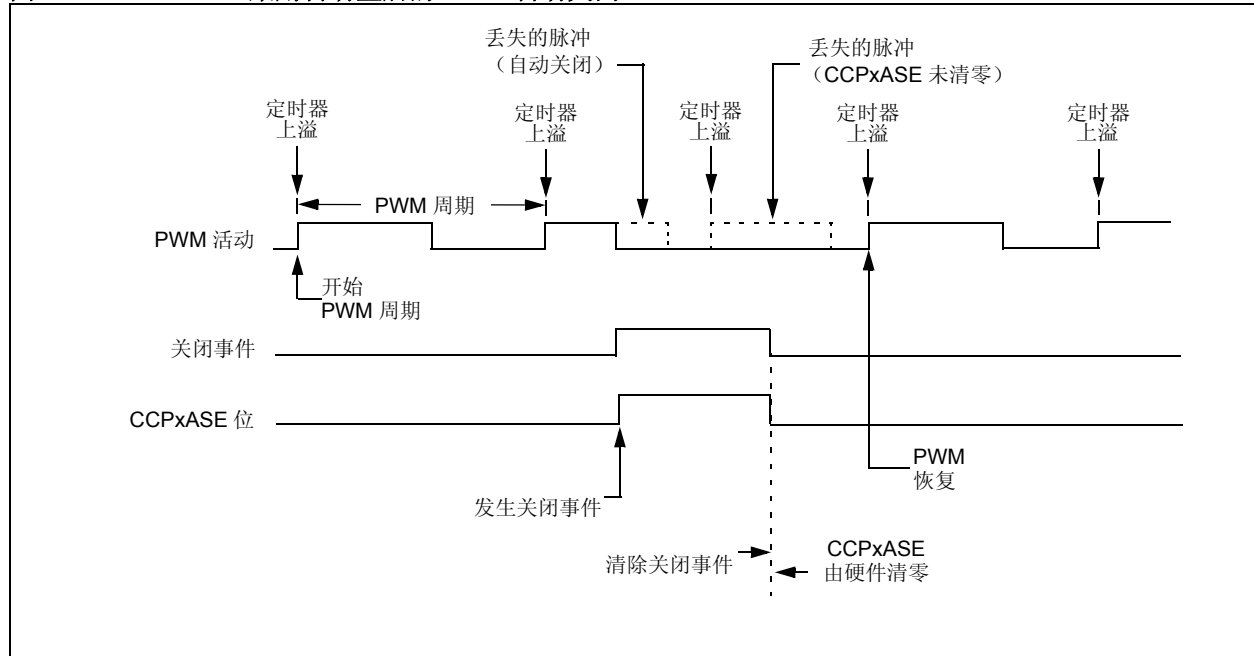


## 23.4.4 自动重启模式

增强型 PWM 可配置为在清除自动关闭条件后自动重启 PWM 信号。将 PWMxCON 寄存器中的 PxRSEN 位置 1 可启用自动重启。

如果使能了自动重启，只要自动关闭条件有效，CCPxASE 位就保持置 1 状态。清除了自动关闭条件后，CCPxASE 位由硬件清零并恢复正常工作。

**图 23-15: 采用自动重启的 PWM 自动关闭 (PXRSEN = 1)**



# PIC16(L)F1933

## 23.4.5 可编程死区延时模式

在所有功率开关都被 PWM 频率调制的半桥应用中，关断功率开关通常比导通它需要更长的时间。如果同时切换上下两个功率开关（一个导通，另一个关断），则两个开关可能在一段很短的时间内都处于导通状态，直到一个开关完全关断。在这个短暂的间隔内，将会有个非常高的电流（**直通电流**）流过这两个功率开关，使得该桥式供电电路短路。要避免切换期间出现这种具有潜在破坏性的直通电流，一般延时一个功率开关的导通以允许另一开关完全关断。

在半桥模式下，使用可数字编程的死区延时来避免直通电流损坏桥式功率开关。该延时在信号从无效状态转变为有效状态时发生。请参见图 23-16 了解详细信息。相关 PWMxCON 寄存器（寄存器 23-5）的低 7 位以单片机指令周期（ $T_{CY}$  或  $4 T_{OSC}$ ）为计数单位设置延时时间。

图 23-16: 半桥 PWM 输出示例

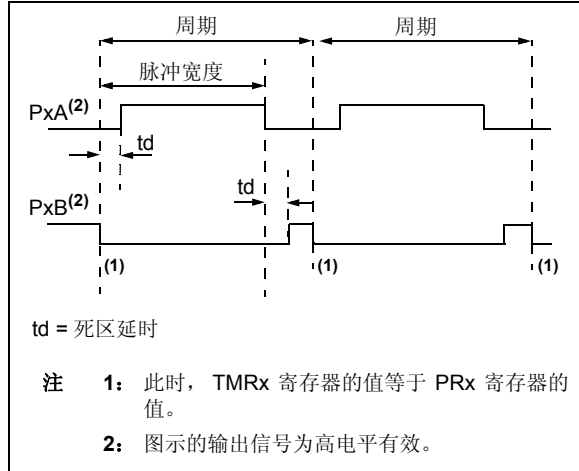
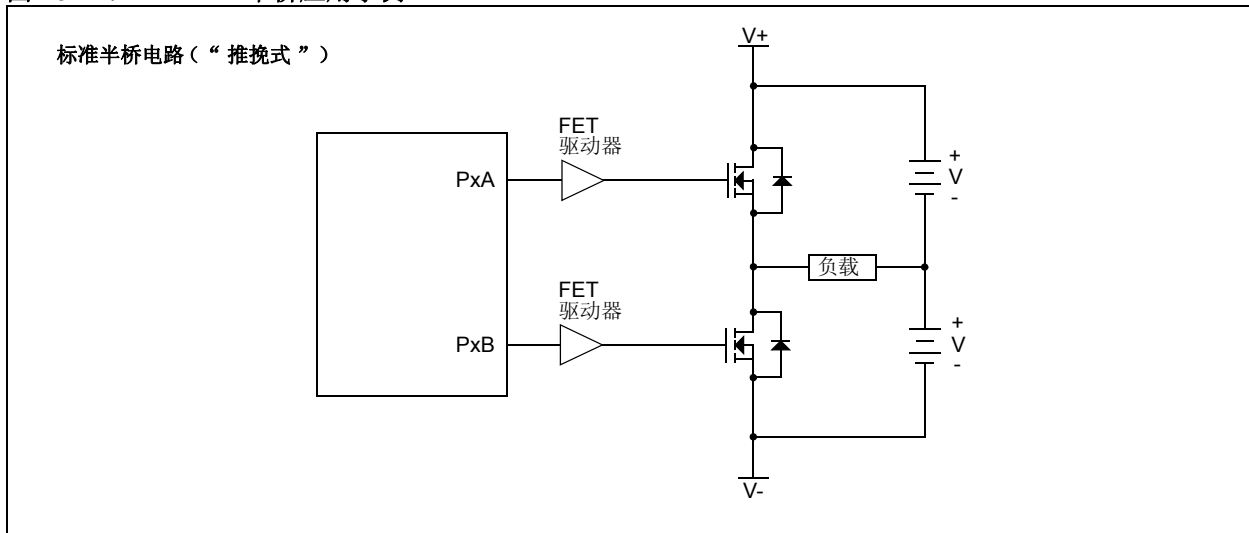


图 23-17: 半桥应用示例



## 23.4.6 PWM 脉冲转向模式

在单输出模式下，PWM 脉冲转向允许任何 PWM 引脚输出调制后的信号。而且，可在多个引脚上同时输出同一 PWM 信号。

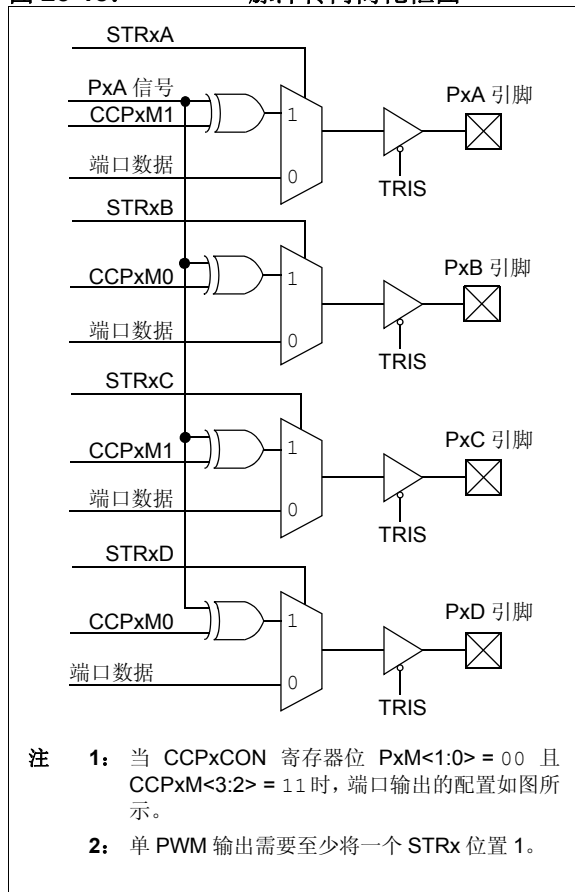
选择了单输出模式（CCPxCON 寄存器的 CCPxM<3:2> = 11 和 PxM<1:0> = 00）后，用户固件通过设置 PSTRxCON 寄存器的相应 STRx<D:A> 位将同一 PWM 信号通过 1 个、2 个、3 个或 4 个输出引脚输出，如表 23-9 所示。

**注：** 为了在某个引脚上输出 PWM 信号，必须将该引脚对应的 TRIS 位设置为输出模式（即设置为 0）以使能该引脚的输出驱动器。

当 PWM 脉冲转向模式有效时，CCPxCON 寄存器的 CCPxM<1:0> 位选择 Px<D:A> 引脚的 PWM 输出信号的极性。

PWM 自动关闭操作也适用于 PWM 脉冲转向模式，如第 23.4.3 节“增强型 PWM 自动关闭模式”所述。自动关闭事件仅影响使能了 PWM 输出的引脚。

**图 23-18: 脉冲转向简化框图**



# PIC16(L)F1933

## 23.4.6.1 脉冲转向同步

PSTRxCON 寄存器的 STRxSYNC 位提供给用户设定转向事件发生时间的两种选择。当 STRxSYNC 位为 0 时，脉冲转向事件发生在写入 PSTRxCON 寄存器的指令结束时。这种情况下，Px<D:A> 引脚上的输出信号可能是不完整的 PWM 波形。当用户固件需要立即从该引脚除去 PWM 信号时，此操作非常有用。

当 STRxSYNC 位为 1 时，实际的脉冲转向更新将发生在下一个 PWM 周期开始时。这种情况下，开/关 PWM 输出将始终产生完整的 PWM 波形。

图 23-19 和 23-20 说明了基于 STRxSYNC 的设置进行 PWM 脉冲转向的时序图。

## 23.4.7 启动注意事项

使用了任何 PWM 模式后，应用硬件必须在 PWM 输出引脚上使用适当的外部上拉和/或下拉电阻。

CCPxCON 寄存器的 CCPxM<1:0> 位允许用户选择每个 PWM 输出引脚对 (PxA/PxC 和 PxB/PxD) 上的 PWM 输出信号为高电平有效还是低电平有效。在使能

PWM 引脚输出驱动器之前，必须选择 PWM 输出信号的极性。不推荐在使能了 PWM 引脚输出驱动器之后更改极性配置，因为这可能会损坏应用电路。

当 PWM 模块初始化时，PxA、PxB、PxC 和 PxD 输出锁存器可能处于不正确的状态。使能 PWM 引脚输出驱动器的同时使能增强型 PWM 模式可能会损坏应用电路。必须将增强型 PWM 模式使能为正确的输出模式，并在使能 PWM 引脚输出驱动器之前完成一个完整的 PWM 周期。在第二个 PWM 周期开始时 PIRx 寄存器的 TMRxIF 位会置 1，从而表明经过了一个完整的 PWM 周期。

**注：** 当单片机从复位状态释放时，所有 I/O 引脚均处于高阻态。外部电路必须保持功率开关器件处于关断状态，直到单片机用适当的信号电平驱动 I/O 引脚或激活 PWM 输出为止。

图 23-19: 指令结束时的脉冲转向事件示例 (STRxSYNC = 0)

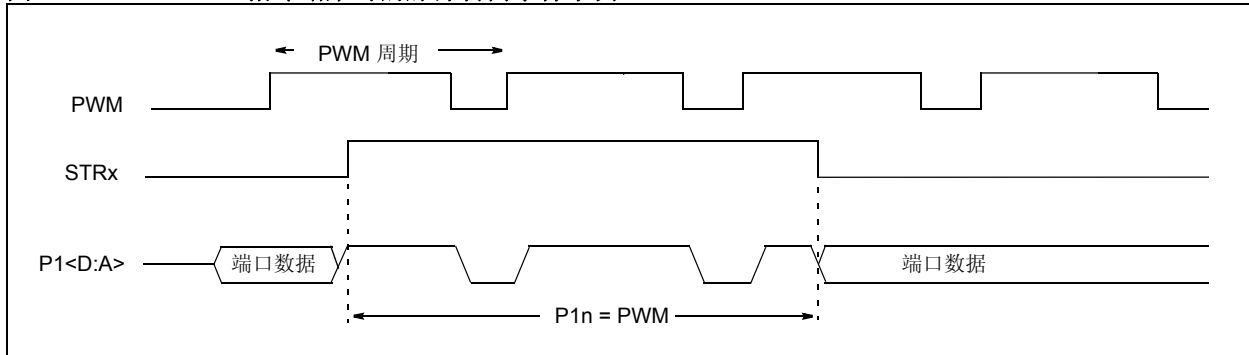
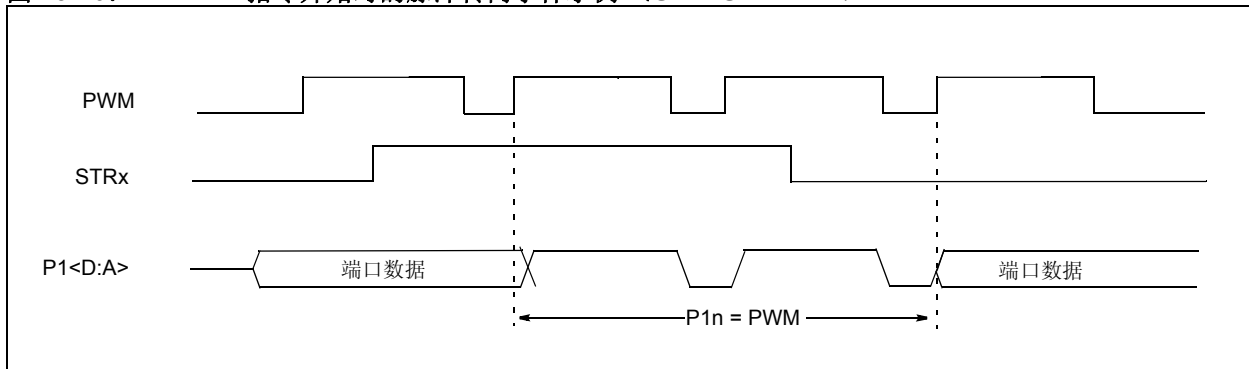


图 23-20: 指令开始时的脉冲转向事件示例 (STRxSYNC = 1)



**表 23-10: 与增强型 PWM 相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面	
CCPxCON	PxM<1:0> <sup>(1)</sup>		DCxB<1:0>		CCPxM<3:0>					216
CCPxAS	CCPxASE	CCPxAS<2:0>			PSSxAC<1:0>		PSSxBD<1:0>			218
CCPTMRS0	C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>			217
CCPTMRS1	—	—	—	—	—	—	C5TSEL<1:0>			217
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86	
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87	
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88	
PIE3	—	CCP5IE	CCP4IE	CCP3IE	TMR6IE	—	TMR4IE	—	89	
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90	
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91	
PIR3	—	CCP5IF	CCP4IF	CCP3IF	TMR6IF	—	TMR4IF	—	92	
PRx	Timer2/4/6 周期寄存器								189*	
PSTRxCON	—	—	—	STRxSYNC	STRxD	STRxC	STRxB	STRxA	220	
PWMxCON	PxRSEN	PxDC<6:0>							219	
TxCON	—	TxOUTPS<3:0>				TMRxON	TxCKPS<:0>1			191
TMRx	Timer2/4/6 模块寄存器								189	
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120	
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	125	
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129	

**图注:** — = 未实现单元, 读为 0。PWM 不使用阴影单元。

**注 1:** 仅适用于 ECCP 模块。

\* 该页提供寄存器信息。

# PIC16(L)F1933

## 23.5 CCP 控制寄存器

寄存器 23-1: CCPxCON: CCPx 控制寄存器

R/W-00	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxM<1:0> <sup>(1)</sup>		DCxB<1:0>		CCPxM<3:0>			
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 **PxM<1:0>**: 增强型 PWM 输出配置位 <sup>(1)</sup>

捕捉模式:

未使用

比较模式:

未使用

如果 CCPxM<3:2> = 00、01 和 10:

xx = PxA 被指定为捕捉 / 比较输入引脚; PxB、PxC 和 PxD 被指定为端口引脚

如果 CCPxM<3:2> = 11:

00 = 单输出; PxA 调制输出; PxB、PxC 和 PxD 被指定为端口引脚

01 = 全桥正向输出; PxD 调制输出; PxA 有效; PxB 和 PxC 无效

10 = 半桥输出; PxA 和 PxB 为带有死区控制的调制输出; PxC 和 PxD 被指定为端口引脚

11 = 全桥反向输出; PxB 调制输出; PxC 有效; PxA 和 PxD 无效

bit 5-4 **DCxB<1:0>**: PWM 占空比最低有效位

捕捉模式:

未使用

比较模式:

未使用

PWM 模式:

这些位是 PWM 占空比的低 2 位。高 8 位在 CCPRxL 中。

bit 3-0 **CCPxM<3:0>**: ECCPx 模式选择位

0000 = 关闭捕捉 / 比较 / PWM (复位 ECCPx 模块)

0001 = 保留

0010 = 比较模式: 匹配时输出电平翻转

0011 = 保留

0100 = 捕捉模式: 每个下降沿捕捉一次

0101 = 捕捉模式: 每个上升沿捕捉一次

0110 = 捕捉模式: 每 4 个上升沿捕捉一次

0111 = 捕捉模式: 每 16 个上升沿捕捉一次

1000 = 比较模式: 初始化 ECCPx 引脚为低电平; 当所比较的值匹配时将输出置为高电平 (将 CCPxIF 置 1)

1001 = 比较模式: 初始化 ECCPx 引脚为高电平; 当所比较的值匹配时将输出置为低电平 (将 CCPxIF 置 1)

1010 = 比较模式: 仅产生软件中断; ECCPx 引脚恢复为 I/O 状态

1011 = 比较模式: 触发特殊事件 (ECCPx 复位定时器, 将 CCPxIF 位置 1, 如果使能了 A/D 模块, 还启动 A/D 转换) <sup>(1)</sup>

仅 CCP4/CCP5:

11xx = PWM 模式

仅 ECCP1/ECCP2/ECCP3:

1100 = PWM 模式: PxA、PxB、PxC 和 PxD 均为高电平有效

1101 = PWM 模式: PxA 和 PxC 高电平有效; PxB 和 PxD 低电平有效

1110 = PWM 模式: PxA 和 PxC 低电平有效; PxB 和 PxD 高电平有效

1111 = PWM 模式: PxA、PxB、PxC 和 PxD 均为低电平有效

注 1: 在 CCP4 和 CCP5 上未实现这些位。



## 寄存器 23-2: CCPTMRS0: PWM 定时器选择控制寄存器 0

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
C4TSEL<1:0>		C3TSEL<1:0>		C2TSEL<1:0>		C1TSEL<1:0>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 **C4TSEL<1:0>**: CCP4 定时器选择位  
 00 = 在 PWM 模式下 CCP4 以 Timer2 作为定时器  
 01 = 在 PWM 模式下 CCP4 以 Timer4 作为定时器  
 10 = 在 PWM 模式下 CCP4 以 Timer6 作为定时器  
 11 = 保留

bit 5-4 **C3TSEL<1:0>**: CCP3 定时器选择位  
 00 = 在 PWM 模式下 CCP3 以 Timer2 作为定时器  
 01 = 在 PWM 模式下 CCP3 以 Timer4 作为定时器  
 10 = 在 PWM 模式下 CCP3 以 Timer6 作为定时器  
 11 = 保留

bit 3-2 **C2TSEL<1:0>**: CCP2 定时器选择位  
 00 = 在 PWM 模式下 CCP2 以 Timer2 作为定时器  
 01 = 在 PWM 模式下 CCP2 以 Timer4 作为定时器  
 10 = 在 PWM 模式下 CCP2 以 Timer6 作为定时器  
 11 = 保留

bit 1-0 **C1TSEL<1:0>**: CCP1 定时器选择位  
 00 = 在 PWM 模式下 CCP1 以 Timer2 作为定时器  
 01 = 在 PWM 模式下 CCP1 以 Timer4 作为定时器  
 10 = 在 PWM 模式下 CCP1 以 Timer6 作为定时器  
 11 = 保留

## 寄存器 23-3: CCPTMRS1: PWM 定时器选择控制寄存器 1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0
—	—	—	—	—	—	C5TSEL<1:0>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-2 **未实现**: 读为 0

bit 1-0 **C5TSEL<1:0>**: CCP5 定时器选择位  
 00 = 在 PWM 模式下 CCP5 以 Timer2 作为定时器  
 01 = 在 PWM 模式下 CCP5 以 Timer4 作为定时器  
 10 = 在 PWM 模式下 CCP5 以 Timer6 作为定时器  
 11 = 保留

# PIC16(L)F1933

## 寄存器 23-4: CCPxAS: CCPX 自动关闭控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
CCPxASE	CCPxAS<2:0>			PSSxAC<1:0>		PSSxBD<1:0>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

- bit 7      **CCPxASE:** CCPx 自动关闭事件状态位  
1 = 发生关闭事件: CCPx 输出处于关闭状态  
0 = CCPx 输出正在工作
- bit 6-4    **CCPxAS<2:0>:** CCPx 自动关闭源选择位  
000 = 禁止自动关闭  
001 = 比较器 C1 输出高电平 <sup>(1)</sup>  
010 = 比较器 C2 输出高电平 <sup>(1)</sup>  
011 = 比较器 C1 或 C2 输出高电平 <sup>(1)</sup>  
100 = INT 引脚上的 V<sub>IL</sub>  
101 = INT 引脚上的 V<sub>IL</sub> 或比较器 C1 输出高电平 <sup>(1)</sup>  
110 = INT 引脚上的 V<sub>IL</sub> 或比较器 C2 输出高电平 <sup>(1)</sup>  
111 = INT 引脚上的 V<sub>IL</sub> 或比较器 C1 或 C2 输出高电平 <sup>(1)</sup>
- bit 3-2    **PSSxAC<1:0>:** 引脚 PxA 和 PxC 关闭状态控制位  
00 = 驱动引脚 PxA 和 PxC 为 0  
01 = 驱动引脚 PxA 和 PxC 为 1  
1x = 引脚 PxA 和 PxC 为三态
- bit 1-0    **PSSxBD<1:0>:** 引脚 PxB 和 PxD 关闭状态控制位  
00 = 驱动引脚 PxB 和 PxD 为 0  
01 = 驱动引脚 PxB 和 PxD 为 1  
1x = 引脚 PxB 和 PxD 为三态

注 1: 如果使能了 CxSYNC, 则关闭延时由 Timer1 确定。

**寄存器 23-5: PWMxCON: 增强型 PWM 控制寄存器**

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
PxRSEN	PxDC<6:0>						
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

bit 7      **PxRSEN:** PWM 重启使能位  
 1 = 自动关闭时, 清除关闭事件后 CCPxASE 位自动清零; PWM 自动重启  
 0 = 自动关闭时, 必须用软件清零 CCPxASE 以重启 PWM

bit 6-0      **PxDC<6:0>:** PWM 延时计数位  
 PxDCx = PWM 信号应转变为有效状态的预定时间和实际时间之差, 为 Fosc/4 (4 \* T<sub>osc</sub>) 的倍数

**注 1:** 选择 LP、XT 或 HS 为振荡器模式或使能故障保护模式时, 发生双速启动会使该位复位为 0。

# PIC16(L)F1933

寄存器 23-6: PSTRxCON: PWM 脉冲转向控制寄存器<sup>(1)</sup>

U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-1/1
—	—	—	STRxSYNC	STRxD	STRxC	STRxB	STRxA
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-5      **未实现:** 读为 0

bit 4      **STRxSYNC:** 脉冲转向同步位

1 = 在下一个 PWM 周期更新输出转向

0 = 在指令周期边界开始处更新输出转向

bit 3      **STRxD:** 脉冲转向使能位 D

1 = PxD 引脚输出由 CCPxM<1:0> 控制极性的 PWM 波形

0 = PxD 引脚被指定为端口引脚

bit 2      **STRxC:** 脉冲转向使能位 C

1 = PxC 引脚输出由 CCPxM<1:0> 控制极性的 PWM 波形

0 = PxC 引脚被指定为端口引脚

bit 1      **STRxB:** 脉冲转向使能位 B

1 = PxB 引脚输出由 CCPxM<1:0> 控制极性的 PWM 波形

0 = PxB 引脚被指定为端口引脚

bit 0      **STRxA:** 脉冲转向使能位 A

1 = PxA 引脚输出由 CCPxM<1:0> 控制极性的 PWM 波形

0 = PxA 引脚被指定为端口引脚

注 1: 仅当 CCPxCON 寄存器位 CCPxM<3:2> = 11 和 Pxm<1:0> = 00 时, PWM 脉冲转向模式才可用。

## 24.0 主同步串行端口（MSSP）模块

### 24.1 主 SSP（MSSP）模块概述

主同步串行端口（Master Synchronous Serial Port, MSSP）模块是用于同其他外设或单片机进行通信的串行接口。这些外设可能是串行 EEPROM、移位寄存器、显示驱动器以及 A/D 转换器等。MSSP 模块有下列两种工作模式：

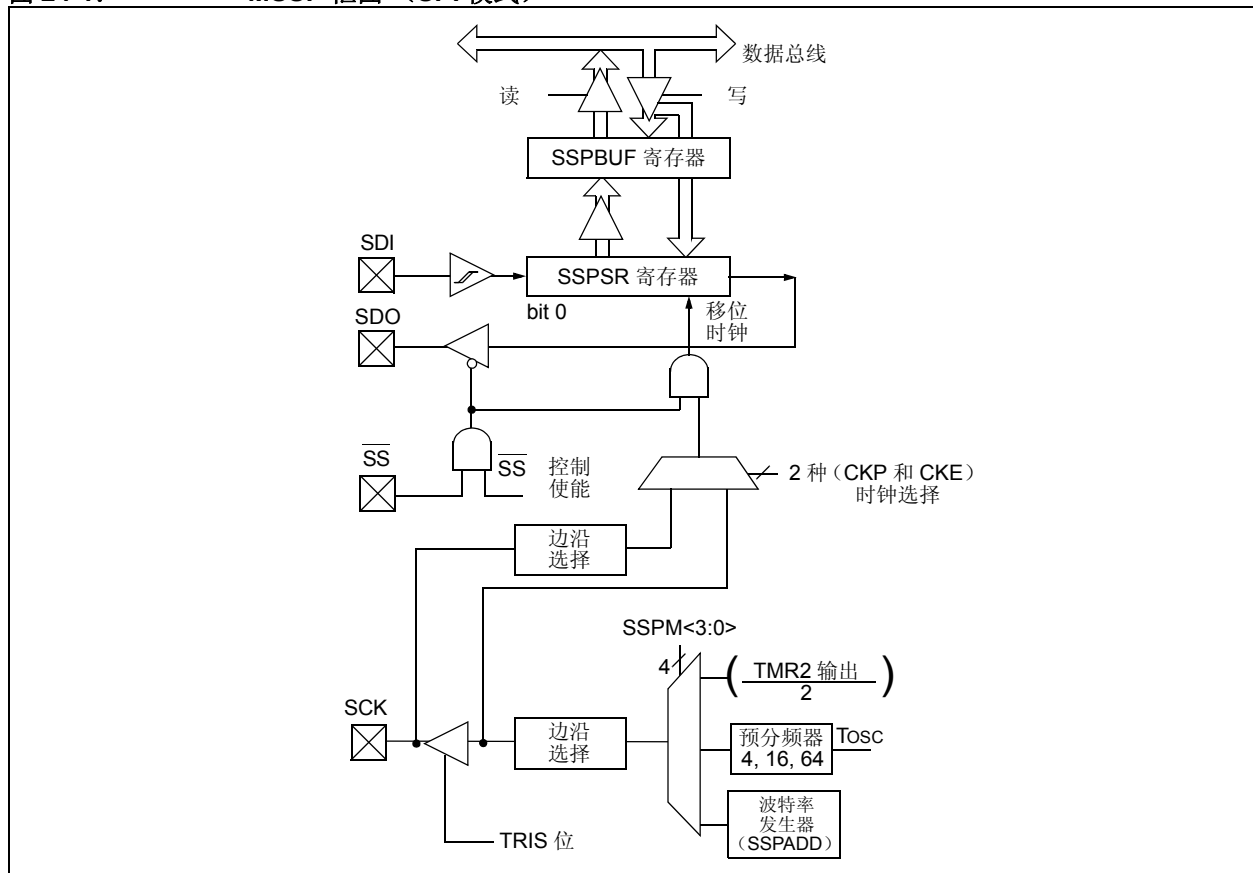
- 串行外设接口（SPI）
- I<sup>2</sup>C™

SPI 接口支持以下模式和功能：

- 主模式
- 从模式
- 时钟极性
- 从选择同步（仅用于从模式）
- 从器件的菊花链连接

图 24-1 给出了 SPI 接口模块的框图。

图 24-1: MSSP 框图（SPI 模式）



# PIC16(L)F1933

I<sup>2</sup>C 接口支持以下模式和功能：

- 主模式
- 从模式
- 不应答字节（从模式）
- 有限多主器件支持
- 7 位和 10 位寻址模式
- 启动和停止中断
- 中断屏蔽
- 时钟延长
- 总线冲突检测
- 广播呼叫地址匹配
- 地址掩码
- 地址保持和数据保持模式
- 可选的 SDA 保持时间

图 24-2 给出了主模式下 I<sup>2</sup>C 接口模块的框图。图 24-3 给出了从模式下 I<sup>2</sup>C 接口模块的框图。

图 24-2: MSSP 框图 (I<sup>2</sup>C™ 主模式)

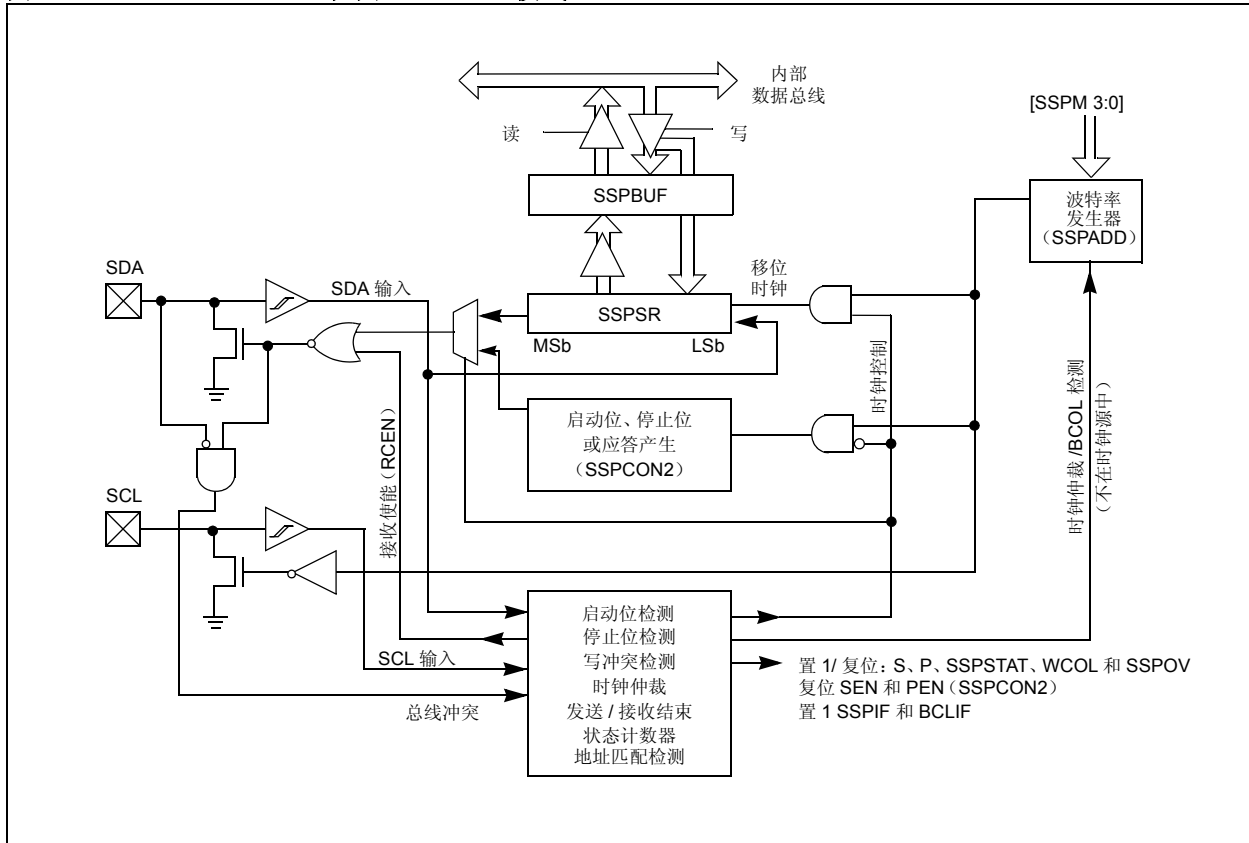
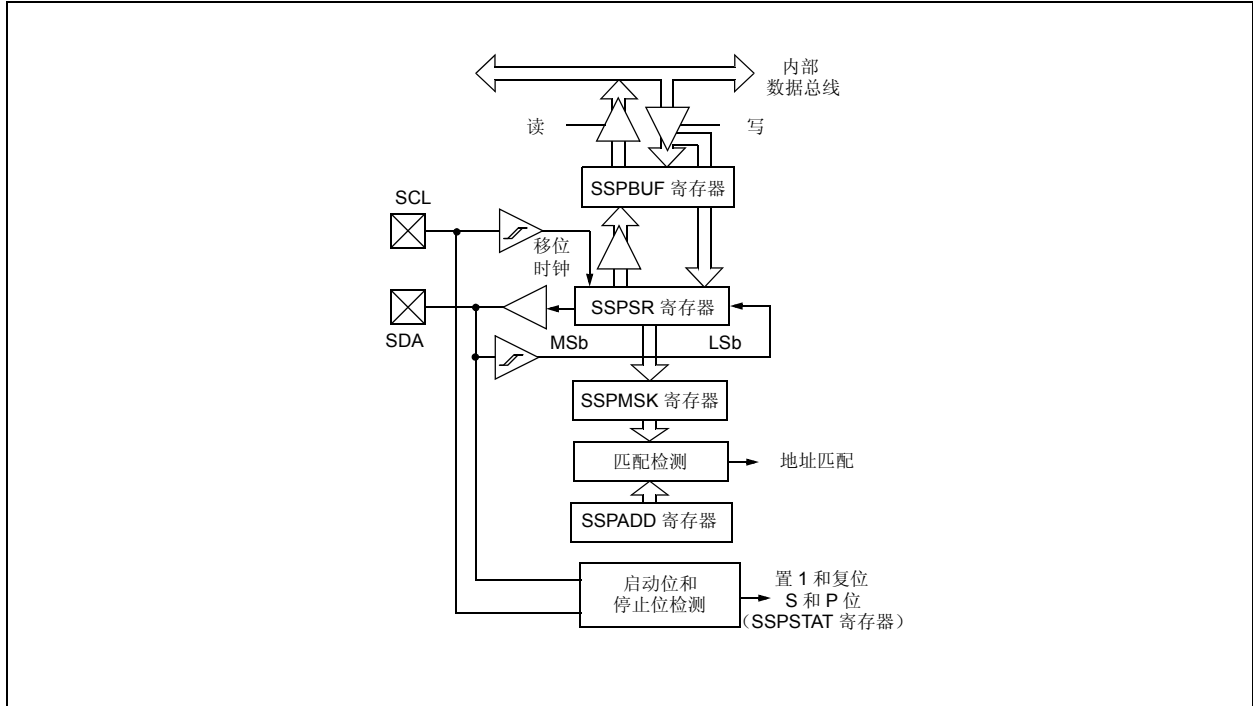


图 24-3: MSSP 框图 (I<sup>2</sup>C™ 从模式)



## 24.2 SPI 模式概述

串行外设接口（Serial Peripheral Interface, SPI）总线是工作在全双工模式下的同步串行数据通信总线。器件在主/从环境下通信，在该环境下，由主器件发起通信。可通过称为从选择的片选信号控制从器件。

SPI 总线指定 4 个信号连接：

- 串行时钟（SCK）
- 串行数据输出（SDO）
- 串行数据输入（SDI）
- 从选择（SS）

图 24-1 给出了工作在 SPI 模式下时的 MSSP 模块框图。

SPI 总线可以允许有一个主器件和一个或多个从器件同时工作。使用多个从器件时，需要从主器件到每个从器件的独立从选择连接。

图 24-4 显示了主器件和多个从器件之间的典型连接。

主器件一次只能选择一个从器件。大多数从器件具有三态输出，因此未选择时其输出信号看起来好像从总线断开一样。

发送需要两个移位寄存器（8 位大小），一个在主器件中，一个在从器件中。在主器件或从器件上，数据始终一次移出一位，首先移出最高有效位（MSb）。同时，新的最低有效位（LSb）移入同一寄存器。

图 24-5 显示了配置为主器件和从器件的两个处理器之间的典型连接。

数据在编程设定的时钟边沿移出两个移位寄存器，并在相反的时钟边沿被锁存。

主器件在与从器件的 SDI 输入引脚连接的 SDO 输出引脚上发出信息，并由从器件 SDI 输入引脚接收该信息。从器件在与主器件的 SDI 输入引脚连接的 SDO 输出引脚上发出信息，并由主器件 SDI 输入引脚接收该信息。

要开始通信，主器件首先发出时钟信号。主器件和从器件应配置为相同的时钟极性。

主器件通过从其移位寄存器发出最高有效位启动发送。从器件从同一线路上读取此位，并将其保存到它的移位寄存器的最低有效位中。

在每个 SPI 时钟周期期间，发生全双工数据发送。这意味着当主器件从其移位寄存器（在其 SDO 引脚上）发出最高有效位且从器件读取该位并将其保存为它的移位

寄存器的 LSb，从器件也从其移位寄存器（在其 SDO 引脚上）发出 MSb 且主器件读取该位并将其保存为它的移位寄存器的 LSb。

移出 8 位数据后，主器件和从器件交换了寄存器值。

如果有更多数据要交换，移位寄存器装入新数据并重复此过程。

数据是有意义还是无意义（无效数据），取决于应用软件。这就导致以下三种数据发送情形：

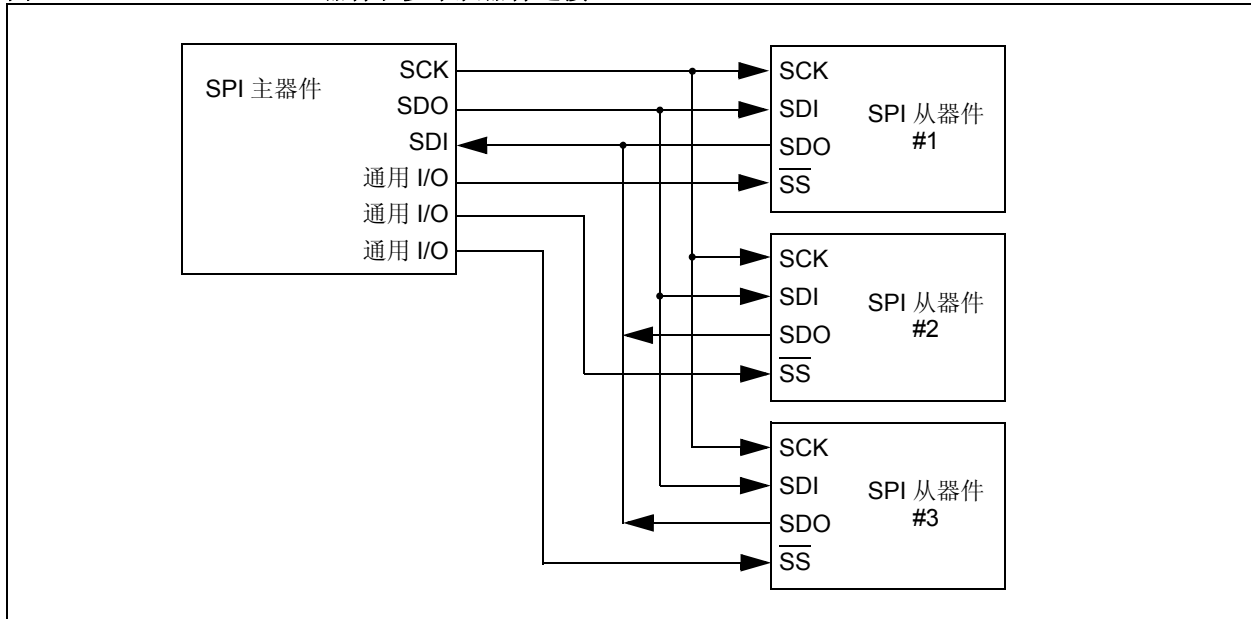
- 主器件发送有用的数据，从器件发送无效数据。
- 主器件发送有用的数据，从器件发送有用的数据。
- 主器件发送无效数据，从器件发送有用的数据。

发送可以占用任何数量的时钟周期。没有更多要发送的数据时，主器件停止发送时钟信号，并且取消选择从器件。

每个与总线连接的从器件，如果尚未通过其从选择线路被选定，则必须丢弃时钟和发送信号，且不得发出任何自己的数据。



图 24-4: SPI 主器件和多个从器件连接



## 24.2.1 SPI 模式寄存器

MSSP 模块提供了 6 个寄存器用于 SPI 工作模式。它们是：

- MSSP 状态寄存器 (SSPSTAT)
- MSSP 控制寄存器 1 (SSPCON1)
- MSSP 控制寄存器 3 (SSPCON3)
- MSSP 数据缓冲寄存器 (SSPBUF)
- MSSP 地址寄存器 (SSPADD)
- MSSP 移位寄存器 (SSPSR) (不可直接访问)

SSPCON1 和 SSPSTAT 是在 SPI 模式下工作的控制寄存器和状态寄存器。SSPCON1 寄存器是可读写的。SSPSTAT 的低 6 位是只读位。SSPSTAT 的高 2 位是可读写的。

在一种 SPI 主模式下，SSPADD 可装入波特率发生器中使用的值。第 24.7 节“波特率发生器”给出了关于波特率发生器的更多信息。

SSPSR 是用来移入 / 移出数据的移位寄存器。SSPBUF 提供了对 SSPSR 寄存器的间接访问。SSPBUF 是用于写 / 读出数据字节的缓冲寄存器。

在接收操作中，SSPSR 和 SSPBUF 一起组成了缓冲接收器。在 SSPSR 接收到一个完整的字节后，该字节被传送到 SSPBUF 且 SSPIF 中断标志位置 1。

发送期间，SSPBUF 不能进行缓冲。对 SSPBUF 的写操作相当于同时写入 SSPBUF 和 SSPSR。

# PIC16(L)F1933

## 24.2.2 SPI 模式工作原理

当初始化 SPI 时，需要指定几个选项。这是通过对相应的控制位（SSPCON1<5:0> 和 SSPSTAT<7:6>）编程来完成的。这些控制位允许指定以下项：

- 主模式（SCK 作为时钟输出）
- 从模式（SCK 作为时钟输入）
- 时钟极性（SCK 的空闲状态）
- 输入数据的采样阶段（数据输出时间的中间或末端）
- 时钟边沿（在 SCK 的上升沿 / 下降沿输出数据）
- 时钟速率（仅用于主模式）
- 从选择模式（仅用于从模式）

要启用串行端口，SSPCON1 寄存器的 SSP 使能位 SSPEN 必须置 1。要复位或重新配置 SPI 模式，应先将 SSPEN 位清零，重新初始化 SSPCONx 寄存器，然后将 SSPEN 位置 1。这将把 SDI、SDO、SCK 和 SS 引脚配置为串行端口引脚。要让上述引脚用于串行端口功能，应对它们的数据方向位（在 TRIS 寄存器中）做如下设置：

- 对于 SDI，必须将相应的 TRIS 位置 1
- 对于 SDO，必须将相应的 TRIS 位清零
- 对于 SCK（主模式），必须将相应的 TRIS 位清零
- 对于 SCK（从模式），必须将相应的 TRIS 位置 1
- 对于 SS，必须将相应的 TRIS 位置 1

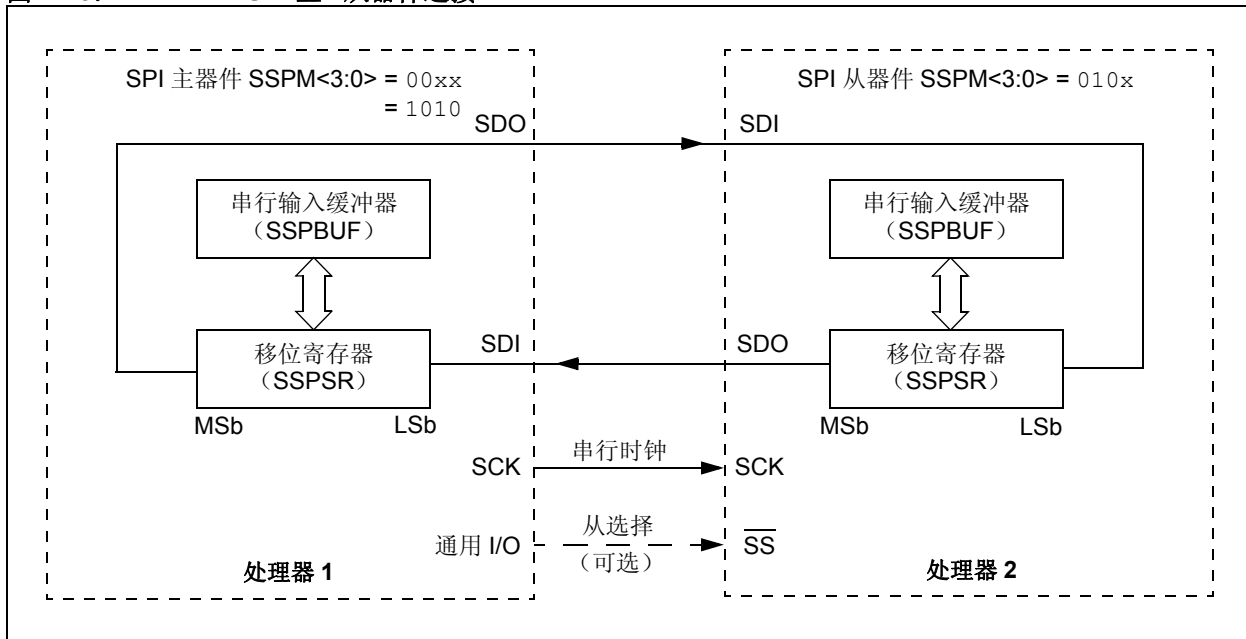
对于不需要的任何串行端口功能，可通过将对应的数据方向（TRIS）寄存器设置为相反值来屏蔽。

MSSP 由一个发送 / 接收移位寄存器（SSPSR）和一个缓冲寄存器（SSPBUF）组成。SSPSR 将数据移入 / 移出器件，最高有效位在前。在数据接收完毕前，SSPBUF 保存上次写入 SSPSR 的数据。一旦 8 位数据接收完毕，该字节就被移入 SSPBUF 寄存器。然后，缓冲器满检测位 BF（在 SSPSTAT 寄存器中）和中断标志位 SSPIF 置 1。这种接收数据的双重缓冲方式（采用 SSPBUF），允许在读取刚接收的数据之前，就开始接收下一个字节。在数据发送/接收期间，任何写 SSPBUF 寄存器的操作都被忽略，并且会将 SSPCON1 寄存器的写冲突检测位 WCOL 置 1。用户软件必须清零 WCOL 位，才能使随后的写 SSPBUF 寄存器的操作成功完成。

为确保应用软件能有效地接收数据，应该在要传送的下一数据字节写入 SSPBUF 之前，读取 SSPBUF 中现有的数据。SSPSTAT 寄存器的缓冲器满检测位 BF 表示将接收到的数据装入 SSPBUF（发送完成）的时间。当 SSPBUF 中的数据被读取后，BF 位即被清零。如果 SPI 仅仅作为一个发送器，则不必理会接收的数据。通常，可用 MSSP 中断来判断发送 / 接收完成的时间。如果不打算使用中断方法，使用软件查询同样可确保不会发生写冲突。

不能直接读写 SSPSR，只能通过寻址 SSPBUF 寄存器来访问。此外，SSPSTAT 寄存器指示各种状态条件。

图 24-5: SPI 主 / 从器件连接



## 24.2.3 SPI 主模式

因为由主器件控制 SCK 线路，所以它可以在任意时刻启动数据传送。主器件根据软件协议确定从器件（图 24-5 中的处理器 2）应在何时广播数据。

在主模式下，数据一旦写入 SSPBUF 寄存器就开始发送/接收。如果只打算将 SPI 作为接收器，则可以禁止 SDO 输出（将其设置为输入）。SSPSR 寄存器按设置的时钟速率连续移入 SDI 引脚上的信号。每收到一个字节，就将其装入 SSPBUF 寄存器，就像接收到普通字节一样（中断和状态位相应置 1）。

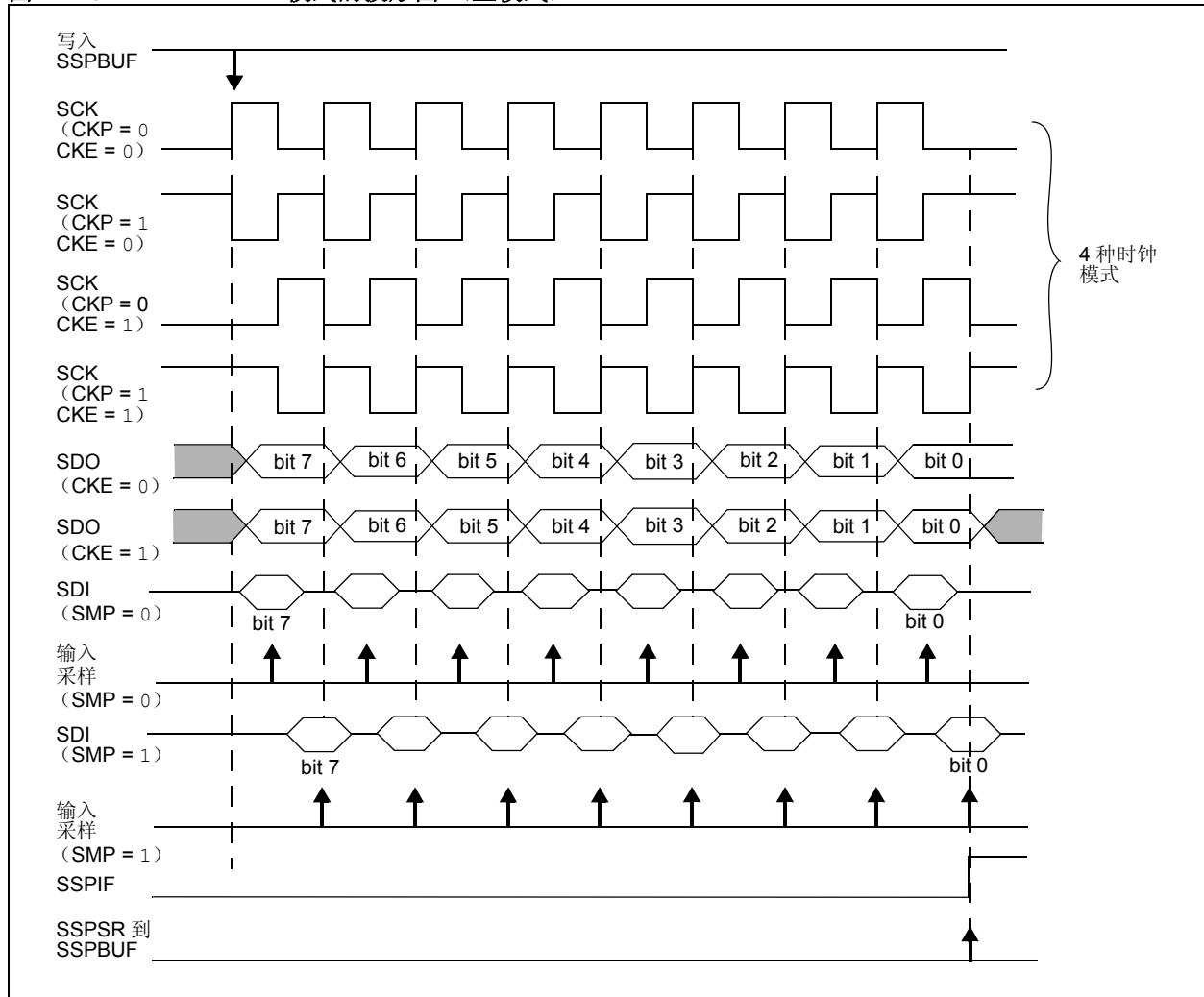
通过对 SSPCON1 寄存器的 CKP 位和 SSPSTAT 寄存器的 CKE 位进行适当的设置来选择时钟极性。图 24-6、图 24-8 和图 24-9 给出了 SPI 通信的波形图，其中首先发送的是最高有效字节。在主模式下，SPI 时钟速率（比特率）可由用户编程设定为下面几种之一：

- $F_{osc}/4$ （或  $T_{CY}$ ）
- $F_{osc}/16$ （或  $4 * T_{CY}$ ）
- $F_{osc}/64$ （或  $16 * T_{CY}$ ）
- Timer2 输出 /2
- $F_{osc}/(4 * (SSPAD + 1))$

图 24-6 给出了主模式的波形图。

当 CKE 位置 1 时，SDO 数据在 SCK 出现时钟边沿前一直有效。图中给出了 SMP 位处于不同状态时，对输入信号进行采样的时间也不同。图中给出了将接收到的数据装入 SSPBUF 的时间。

图 24-6: SPI 模式的波形图（主模式）



## 24.2.4 SPI 从模式

在从模式下，当 SCK 引脚上出现外部时钟脉冲时发送和接收数据。当最后一位数据被锁存后，SSPIF 中断标志位置 1。

在使能该模块工作在 SPI 从模式下之前，时钟线必须处于正确的空闲状态。通过读 SCK 引脚可以查看时钟线的状态。具体的空闲状态由 SSPCON1 寄存器的 CKP 位决定。

在从模式下，外部时钟由 SCK 引脚上的外部时钟源提供。外部时钟必须满足电气规范中规定的高电平和低电平的最短时间要求。

在休眠模式下，从器件仍可发送 / 接收数据。移位寄存器的时钟来自 SCK 引脚的输入信号，当接收到一个字节时，器件会产生一个中断。如果允许该中断，则器件将从休眠模式唤醒。

### 24.2.4.1 菊花链配置

SPI 总线有时采用菊花链配置进行连接。第一个从器件的输出连接到第二个从器件的输入，第二个从器件的输出连接到第三个从器件的输入，依此类推。最后一个从器件的输出连接到主器件的输入。第二组时钟脉冲期间，每个从器件发出在第一组时钟脉冲期间收到的数据的确切副本。整个链路充当一个大的通信移位寄存器。菊花链功能只需要有一个来自主器件的从选择线路。

图 24-7 给出了工作在 SPI 模式下的典型菊花链连接的框图。

在菊花链配置中，从器件只需要总线上的最新字节。将 SSPCON3 寄存器的 BOEN 位置 1 将使能对 SSPBUF 寄存器的写操作，即使前一个字节未被读取时也是如此。这允许软件忽略可能不是最新的数据。

## 24.2.5 从选择同步

从选择还可用于同步通信。在主器件准备好通信之前，从选择线路保持高电平。从选择线路拉低时，从器件就会知道正在开始新的发送。

如果从器件未正确接收通信，它将在发送结束时复位，此时从选择线路返回高电平状态。然后从器件准备好在从选择线路再次拉低时接收新的发送。如果未使用从选择线路，将存在从器件最终不与主器件同步的风险。如果从器件错过一位，它将在后续的发送中始终错过一位。使用从选择线路允许主器件和从器件在每次发送开始时同步。

$\overline{SS}$  引脚允许器件工作在同步从模式下。SPI 必须处于从模式，并使能  $\overline{SS}$  引脚控制 (SSPCON1<3:0> = 0100)。

当  $\overline{SS}$  引脚为低电平时，使能数据的发送和接收，同时驱动 SDO 引脚。

当  $\overline{SS}$  引脚变为高电平时，即使是在字节的发送过程中，SDO 引脚也不再被驱动，而是变成悬空输出状态。根据应用的需要，可在 SDO 引脚上外接上拉 / 下拉电阻。

- |  |
|--|
| <p><b>注 1:</b> 当 SPI 处于从模式并使能了 <math>\overline{SS}</math> 引脚控制 (SSPCON1&lt;3:0&gt; = 0100) 时，如果 <math>\overline{SS}</math> 引脚设置为 VDD，那么 SPI 模块将复位。</p> <p><b>2:</b> 当 SPI 工作在从模式下并且 CKE 位置 1 时，用户必须使能 <math>\overline{SS}</math> 引脚控制。</p> <p><b>3:</b> 当工作在 SPI 从模式下时，SSPSTAT 寄存器的 SMP 位必须保持清零状态。</p> |
|--|

当 SPI 模块复位时，位计数器被强制为 0。这可以通过强制  $\overline{SS}$  引脚为高电平或清零 SSPEN 位来实现。

图 24-7: SPI 菊花链连接

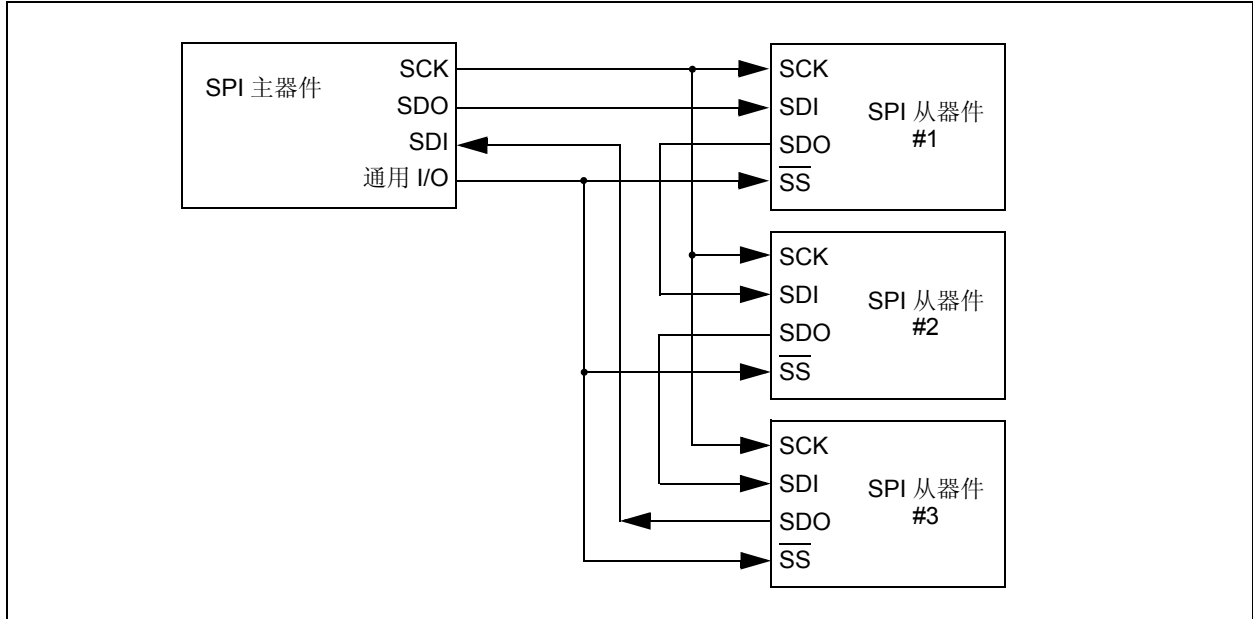
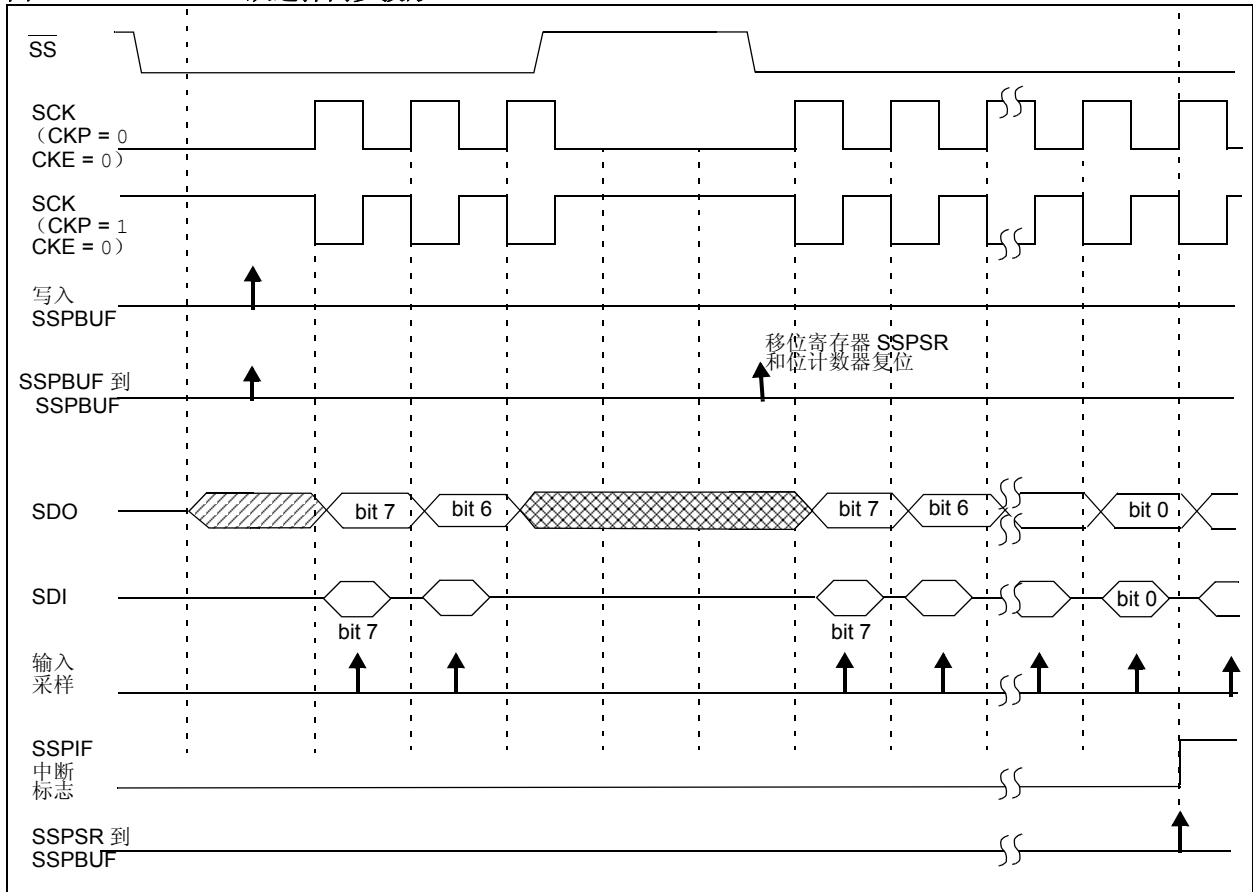


图 24-8: 从选择同步波形



# PIC16(L)F1933

图 24-9: SPI 模式波形 (从模式且  $CKE = 0$ )

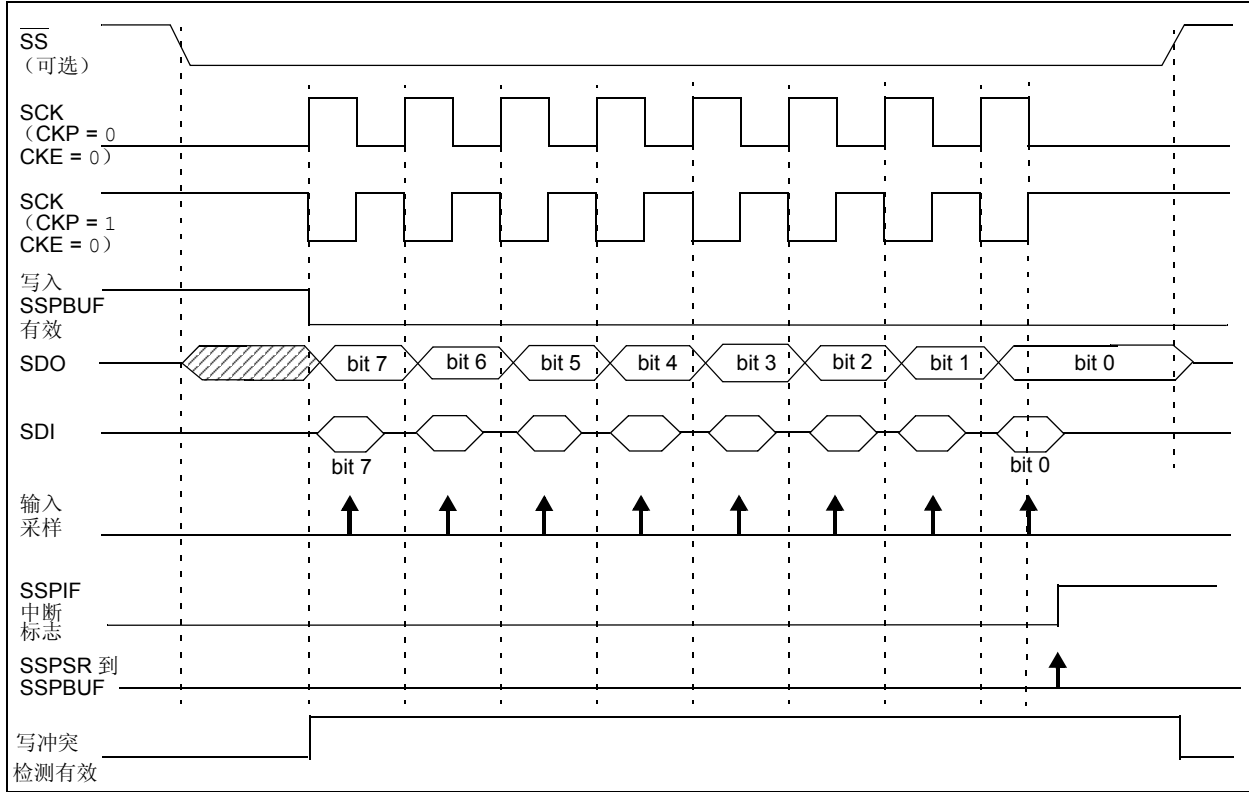
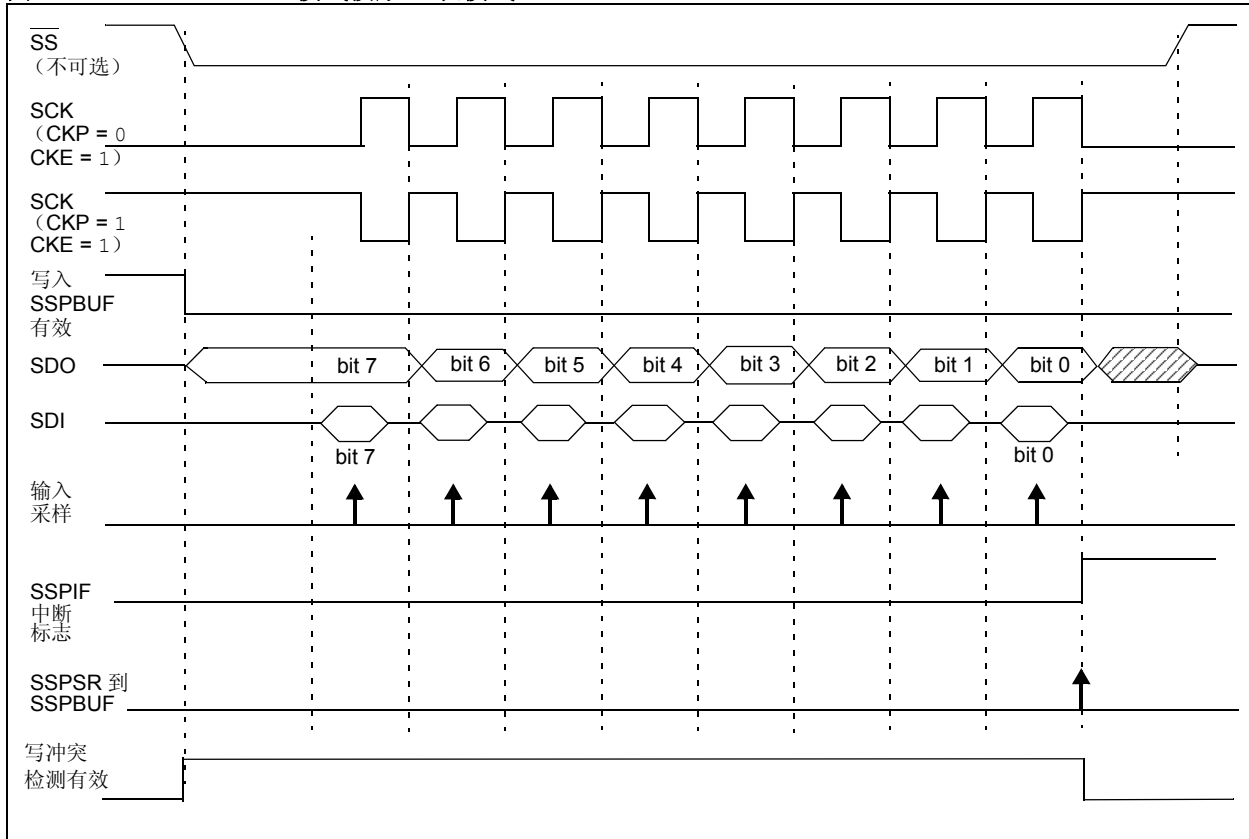


图 24-10: SPI 模式波形 (从模式且  $CKE = 1$ )



## 24.2.6 SPI 在休眠模式下的工作原理

在 SPI 主模式下，模块时钟以不同于全功耗模式下的速度运行；在休眠模式下，所有时钟都停止。

当 MSSP 时钟比系统时钟快许多时，用户必须特别小心。

在从模式下，当允许了 MSSP 中断时，在主器件发送数据完毕后，MSSP 中断将控制器从休眠唤醒。

如果不希望退出休眠模式，应禁止 MSSP 中断。

在 SPI 主模式下，选择了休眠模式后所有模块的时钟都停止，并且在器件唤醒前，发送 / 接收也将保持原先的状态。在器件返回运行模式后，模块将继续发送和接收数据。

在 SPI 从模式下，SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件置于休眠模式下，且数据被移入 SPI 发送 / 接收移位寄存器。在接收完所有 8 位数据后，MSSP 中断标志位将置 1，而且如果此时允许该中断，还将唤醒器件。

**表 24-1: 与 SPI 操作相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	121
APFCON	—	CCP3SEL	T1GSEL	P2BSEL	SRNQSEL	C2OUTSEL	SSSEL	CCP2SEL	118
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSP1IE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSP1IF	CCP1IF	TMR2IF	TMR1IF	90
SSPBUF	同步串行端口接收缓冲器 / 发送寄存器								225*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				269
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	272
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	268
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	120
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISB2	TRISC1	TRISC0	129

**图注:** — = 未实现单元，读为 0。SPI 模式下的 MSSP 不使用阴影单元。

\* 该页提供寄存器信息。

# PIC16(L)F1933

## 24.3 I<sup>2</sup>C 模式概述

内部集成电路总线（I<sup>2</sup>C）是多主器件串行数据通信总线。器件可以在主 / 从环境下通信，在该环境下，由主器件发起通信。可通过寻址控制从器件。

I<sup>2</sup>C 总线指定两种信号连接：

- 串行时钟（SCL）
- 串行数据（SDA）

图24-11给出了工作在I<sup>2</sup>C模式下时的MSSP模块框图。

SCL 和 SDA 连接都是双向漏极开路线路，每个都需要通过上拉电阻连接到电源电压。将线路接地可视为逻辑 0，将线路悬空可视为逻辑 1。

图 24-11 显示了配置为主器件和从器件的两个处理器之间的典型连接。

I<sup>2</sup>C 总线可以允许有一个或多个主器件和一个或多个从器件同时工作。

对于给定器件，有四种可能的工作模式：

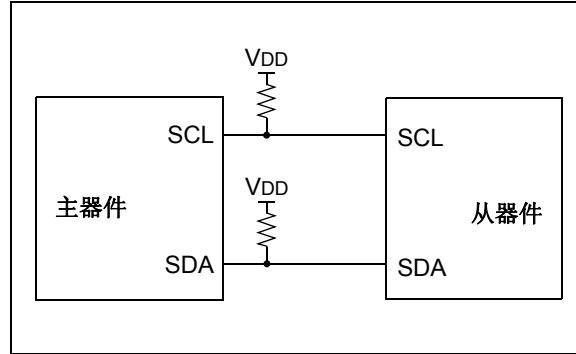
- 主发送模式  
（主器件向从器件发送数据）
- 主接收模式  
（主器件接收来自从器件的数据）
- 从发送模式  
（从器件向主器件发送数据）
- 从接收模式  
（从器件接收来自主器件的数据）

要开始通信，主器件开始应处于主发送模式。主器件发送启动位，后跟它想与之通信的从器件的地址字节。随后是单个读 / 写位，用于确定主器件是想发送数据还是想从从器件接收数据。

如果总线上存在请求的从器件，那么它将以应答位（也称为 ACK）作为响应。然后，主器件继续处于发送模式或接收模式，从器件继续处于与其对应的模式，即处于接收模式或发送模式。

启动位定义为 SCL 线保持高电平时 SDA 线从高电平到低电平的变化。发出地址和数据字节时，首先发送最高有效位（MSb）。当主器件想从从器件读取数据时，读 / 写位作为逻辑 1 发出，当主器件想写入数据到从器件时，读 / 写位作为逻辑 0 发出。

图 24-11: I<sup>2</sup>C 主 / 从连接



应答位（ACK）是低电平有效的信号，它将 SDA 线保持为低电平以告知发送器从器件已接收到发送的数据并准备接收更多数据。

SCL 线保持低电平时，始终执行数据位的转换。SCL 线路保持高电平时进行的转换用于指示启动和停止位。

如果主器件想写入数据到从器件，那么它将重复发出一字节数据，同时从器件在每个字节后以 ACK 位作出响应。在此示例中，主器件处于主发送模式，从器件处于从接收模式。

如果主器件想从从器件读取数据，那么它将重复接收来自从器件的一字节数据，同时每个字节后以 ACK 位作出响应。在此示例中，主器件处于主接收模式，从器件处于从发送模式。

在传输数据的最后一个字节时，主器件可以通过发送停止位结束发送。如果主器件处于接收模式，它将发送停止位代替最后的 ACK 位。停止位定义为 SCL 线保持高电平时 SDA 线从低电平到电平的变化。

在某些情况下，主器件可能想保持对总线的控制权，重新发起另一次发送。如果这样，主器件可以发送另一个启动位代替停止位或最后的 ACK 位（处于接收模式时）。

I<sup>2</sup>C 总线指定三种报文协议：

- 主器件向从器件写入数据的单个报文。
- 主器件从从器件读取数据的单个报文。
- 组合报文，其中主器件至少向一个或多个从器件发起两次写入操作或两次读取操作或一次读写操作组合。



一个器件发送逻辑 1 或将线路悬空，并且第二个器件发送逻辑 0 或保持线路为低电平时，第一个器件可以检测到线路上不是逻辑 1。这种检测用于 SCL 线路时，被称为时钟延长。时钟延长为从器件提供了一种控制数据流的机制。这种检测用于 SDA 线路时，被称为仲裁。仲裁可以确保任一时间内只有一个主器件在传输数据。

### 24.3.1 时钟延长

从器件尚未处理完数据时，它可以通过时钟延长过程延迟传输其他数据。寻址到的从器件可能在接收或发送一位后保持 SCL 时钟线为低电平，这表示它尚未准备好继续。与该从器件通信的主器件可以尝试拉高 SCL 线以传输下一位，但是将检测到时钟线尚未被释放。由于 SCL 连接是漏极开路，从器件可以在准备好继续通信之前保持该线为低电平。

时钟延长允许无法与发送器同步的接收器控制传入数据流。

### 24.3.2 仲裁

每个主器件必须监视总线上的启动和停止位。如果器件检测到总线繁忙，它将无法开始新的报文，直到总线返回到空闲状态为止。

但是，两个主器件可能同时或几乎同时尝试启动发送。发生这种情况时，开始仲裁过程。每个发送器都检查 SDA 数据线的电平，并将其与它希望找到的电平比较。第一个发现两个电平不匹配的发送器将仲裁失败，并且必须停止 SDA 线路上的发送。

例如，如果一个发送器将 SDA 线保持为逻辑 1（将其悬空），第二个发送器将 SDA 线保持为逻辑 0（将其拉低），结果 SDA 线将为低电平。第一个发送器将发现线路电平与期望的电平不同，并得出结论由另一个发送器进行通信。

注意到电平不同的第一个发送器是仲裁失败并且必须停止驱动 SDA 线的发送器。如果此发送器还是主器件，它也必须停止驱动 SCL 线。然后它将监视线路上的停止条件，然后尝试再次发起发送。同时，未注意到期望电平与 SDA 线上的实际电平之间有任何不同的另一器件将继续其初始传输。它可以毫无困难地完成操作，因为此时，没有其他发送器干扰，报文发送如期进行。

当主器件寻址到多个从器件时，也可以对从发送模式进行仲裁，但是这不常见。

如果两个主器件发送同一个报文给两个地址段的两个不同从器件，发送较低从器件地址的主器件将始终仲裁成功。两个主器件向同一从器件地址发送报文，并且地址可能有时指向多个从器件时，仲裁过程必须前进到数据段。

仲裁通常极少发生，但是它是正确支持多主器件的必需过程。

## 24.4 I<sup>2</sup>C™ 模式工作原理

所有 MSSP I<sup>2</sup>C 通信都是面向字节的，且首先移出最高有效位。该模块提供了 6 个 SFR 寄存器和 2 个中断标志来与 PIC® 单片机和用户软件接口。该模块还采用了 2 个引脚（SDA 和 SCL）与其他外部 I<sup>2</sup>C 器件通信。

### 24.4.1 字节格式

I<sup>2</sup>C 模式下的所有通信都采用 9 位数据段。字节可从主器件发送到从器件（或反向发送），随后返回一个应答位。在 SCL 线的第 8 个下降沿之后，器件将数据输出到 SDA 引脚，将该引脚变为输入引脚，然后在下一个时钟脉冲时读取应答值。

由主器件提供时钟信号 SCL。当 SCL 信号为低电平时，更改数据有效，新数据将在时钟信号的上升沿被采样。SCL 线为高电平时 SDA 线上的更改被定义为总线上的特殊条件，说明见下文。

### 24.4.2 I<sup>2</sup>C 术语定义

I<sup>2</sup>C 通信描述中的语言和术语提供了具体的 I<sup>2</sup>C 定义。下面定义了所使用的术语，在本文档的后续部分中可能会用到这些术语，而不另作说明。此表摘自 Philips I<sup>2</sup>C 规范。

### 24.4.3 SDA 和 SCL 引脚

选择任何 I<sup>2</sup>C 模式并将 SSPEN 位置 1，可强制 SCL 和 SDA 引脚为漏极开路。用户应通过将相应的 TRIS 位置 1 来把这些引脚设置为输入引脚。

**注：** 使能了某种 I<sup>2</sup>C 模式后，将输出全零数据。

### 24.4.4 SDA 保持时间

通过 SSPCON3 寄存器的 SDAHT 位可以选择 SDA 引脚的保持时间。保持时间是 SCL 下降沿后 SDA 保持有效的的时间。将 SDAHT 位置 1 可以选择至少长达 300 ns 的保持时间，有助于具有大电容的总线。

**表 24-2: I<sup>2</sup>C 总线术语**

术语	描述
发送器	将数据移出到总线的器件。
接收器	从总线移入数据的器件。
主器件	发起传送、生成时钟信号和终止传送的器件。
从器件	被主器件寻址的器件。
多主器件	总线上不止一个器件可以发起数据传送。
仲裁	确保一个时刻只有一个主器件控制总线的过程。赢得仲裁可确保报文不被破坏。
同步	同步总线上两个或两个以上器件的时钟的过程。
空闲	没有主器件控制总线，且 SDA 和 SCL 线都为高电平。
工作	任何时间都有一个或一个以上的主器件正在控制总线。
寻址到的从器件	接收到匹配地址并使用由主器件提供的时钟的从器件。
匹配地址	从器件接收到的地址字节与存储在 SSPADD 中的值相匹配。
写请求	从器件接收到 R/W 位清零的匹配地址，且准备随时钟移入数据。
读请求	主器件发送 R/W 位置 1 的地址字节，表示要求从器件随时钟将数据移出。从器件在接收到该地址字节后会立即移出所有数据字节直到发生重复启动或停止条件。
时钟延长	总线上的器件保持 SCL 低电平以暂停通信的时间。
总线冲突	当模块采样到 SDA 线为低电平，但当 SDA 线正在输出数据而期望值为高电平的任何情况。

## 24.4.5 启动条件

I<sup>2</sup>C 规范将启动条件定义为 SCL 线为高电平时 SDA 从高电平到低电平的变化。启动条件始终由主器件产生，指示总线从空闲状态到工作状态的变化。图 24-10 给出了启动条件和停止条件的波形图。

如果模块在将 SDA 线拉为低电平之前采样到它已经是低电平状态，则表示启动条件期间发生总线冲突。这不符合 I<sup>2</sup>C 规范，该规范规定启动时不能发生总线冲突。

## 24.4.6 停止条件

停止条件指 SCL 线为高电平时 SDA 线从低电平到高电平的变化。

**注：** 在停止条件有效之前必须至少出现一个 SCL 低电平时间，因此如果在 SCL 线处于高电平状态时，SDA 线上再次发生从低电平到高电平的变化，那么只能检测到启动条件。

## 24.4.7 重复启动条件

停止条件有效时，重复启动条件也有效。如果主器件希望在终止当前传送后仍占用总线，可发出一个重复启动条件。重复启动条件与启动条件对从器件来说具有相同的作用，都会复位所有的从器件逻辑并准备随着时钟移入地址。主器件可能想寻址同一从器件或其他从器件。

在 10 位寻址从模式下，主器件需要使用重复启动条件来将数据随着时钟移出寻址到的从器件。从器件被完全寻址到（即高低地址字节都匹配）后，主器件发出重复启动条件以及 R/W 位置 1 的高地址字节。然后，从器件逻辑采用该时钟并准备随着时钟移出数据。

在 10 位模式下与 R/W 清零的地址字节完全匹配后，前一次匹配标志置 1 并保持置 1 状态不变。在停止条件之前，R/W 清零的高地址或高地址匹配失败。

## 24.4.8 启动 / 停止条件中断屏蔽

SSPCON3 寄存器的 SCIE 和 PCIE 位可允许在从模式下产生中断（该模式通常不支持此功能）。在已允许启动和停止条件检测中断的从模式下，这两位没有影响。

图 24-12: I<sup>2</sup>C 启动和停止条件

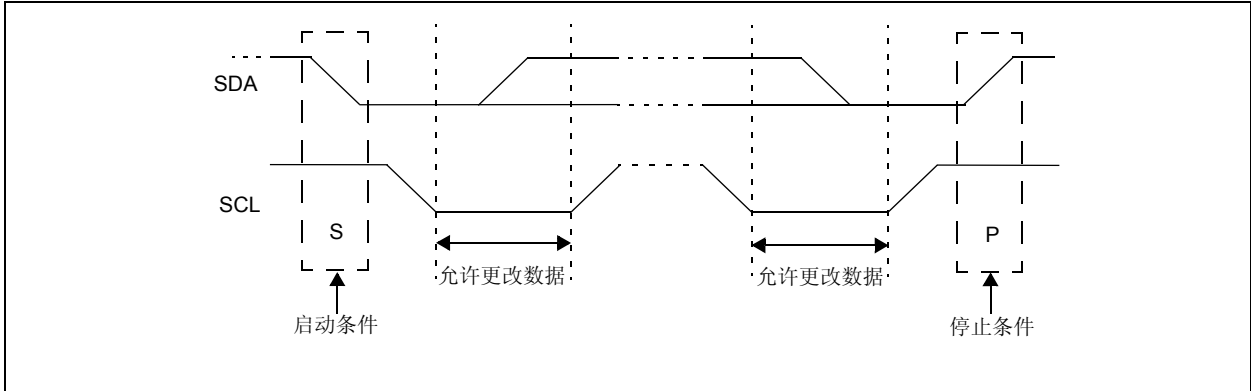
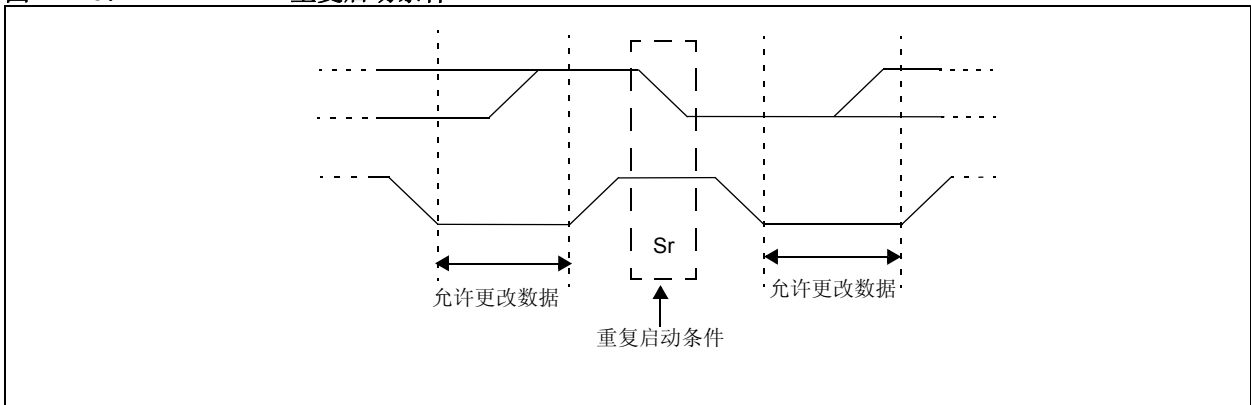


图 24-13: I<sup>2</sup>C 重复启动条件



## 24.4.9 应答序列

I<sup>2</sup>C 模式下传送的任何字节的第 9 个 SCL 脉冲专用作应答信号。它允许接收器件通过将 SDA 线拉为低电平来响应发送器。发送器在该时间内必须释放对 SDA 线的控制权以移入响应信号。应答 (ACK) 信号是低电平有效的信号，将 SDA 线拉为低电平是为了告知发送器该器件已接收到发送的数据并准备接收更多数据。

ACK 信号的结果保存在 SSPCON2 寄存器的 ACKSTAT 位中。

当 AHEN 和 DHEN 位置 1 时，从软件允许用户设置返回给发送器的 ACK 值。SSPCON2 寄存器的 ACKDT 位被置 1/ 清零，以决定响应信号。

如果 SSPCON3 寄存器的 AHEN 和 DHEN 位清零，则从硬件产生 ACK 响应信号。

在某些情况下，从器件不发送 ACK 信号。例如，如果接收到字节时 SSPSTAT 寄存器的 BF 位或 SSPCON1 寄存器的 SSPOV 位置 1，从器件将不发送 ACK 信号。

寻址到模块后，在总线上 SCL 的第 8 个下降沿后，将 SSPCON3 寄存器的 ACKTIM 位置 1。ACKTIM 位指示活动总线的应答时间。ACKTIM 状态位仅在使能 AHEN 位或 DHEN 位时有效。

## 24.5 I<sup>2</sup>C 从模式工作原理

通过 SSPCON1 寄存器中的 SSPM 位，可在 4 种 MSSP 从模式中选择一种作为工作模式。工作模式还可划分为 7 位寻址模式和 10 位寻址模式。10 位寻址模式与 7 位寻址模式的工作原理相同，只是在处理较大地址时需要增加系统开销。

配置了启动位和停止位中断的模式与其他模式的工作原理相同，只是 SSPIF 在检测到启动条件、重复启动条件或停止条件时会置 1。

### 24.5.1 从模式地址

SSPADD 寄存器 (寄存器 24-6) 包含了从模式地址。启动或重复启动条件后收到的第一个字节与存储在该寄存器中的值相比较。如果该字节匹配，则将该值装入 SSPBUF 寄存器并产生中断。如果该值不匹配，则模块进入空闲模式，不会向软件指示发生了任何事情。

SSP 掩码寄存器 (寄存器 24-5) 影响地址匹配过程。更多信息，请参见第 24.5.9 节“SSP 掩码寄存器”。

#### 24.5.1.1 I<sup>2</sup>C 从模式 7 位寻址

在 7 位寻址模式下，当确定是否发生地址匹配时忽略已接收数据字节的最低有效位。

#### 24.5.1.2 I<sup>2</sup>C 从模式 10 位寻址

在 10 位寻址模式下，接收到的第一个字节与二进制值 1 1 1 1 0 A9 A8 0 相比较。A9 和 A8 是 10 位地址的高 2 位，分别存储在 SSPADD 寄存器的 bit 2 和 bit 1 中。

在应答高字节之后，用户使用低地址字节更新 SSPADD 之前，UA 位置 1 且 SCL 保持低电平。低地址字节随着时钟移入，且所有 8 位都与 SSPADD 中的低地址值相比较。即使未发生地址匹配，在 SSPADD 更新以再次接收高字节之前，SSPIF 和 UA 位置 1 且 SCL 保持低电平。SSPADD 更新时，UA 位清零。这确保了模块可在下次通信时接收高地址字节。

在所有 10 位寻址通信开始时，需要以写请求的方式发送高低地址字节以判断地址是否匹配。在寻址到从器件后，通过发出重复启动条件并随着时钟移入 R/W 位置 1 的高地址字节来启动发送。然后，从硬件应答读请求并准备随着时钟移出数据。这只有在从器件接收到与之匹配的完整的高低地址字节后才有效。

## 24.5.2 从接收

当接收到的匹配地址字节的  $\overline{R/W}$  位清零时，SSPSTAT 寄存器的 R/W 位也清零。接收到的地址被装入 SSPBUF 寄存器并产生应答信号。

当接收到的地址存在溢出条件时，不提供应答信号。溢出条件定义为 SSPSTAT 寄存器的 BF 位置 1 或 SSPCON1 寄存器的 SSPOV 位置 1。SSPCON3 寄存器的 BOEN 位可修改此操作。更多信息，请参见寄存器 24-4。

传送的每个数据字节都会产生 MSSP 中断。必须用软件将标志位 SSPIF 清零。

当 SSPCON2 寄存器的 SEN 位置 1 时，每接收到一个字节 SCL 都会保持低电平（时钟延长）状态。必须通过将 SSPCON1 寄存器的 CKP 位置 1 来释放时钟，10 位模式下的特殊情况除外。更多详细信息，请参见第 24.2.3 节“SPI 主模式”。

### 24.5.2.1 7 位寻址接收

本节描述了 MSSP 模块配置为工作在 7 位寻址模式下的 I<sup>2</sup>C 从器件时的标准事件序列。还描述了所有由硬件或软件所作的决定及其对接收的影响。图 24-13 和图 24-14 用直观的方式对此作了说明。

下面是完成 I<sup>2</sup>C 通信通常必需执行的详细步骤。

1. 启动位检测。
2. SSPSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPIF 位也置 1。
3. 接收  $\overline{R/W}$  位清零的匹配地址。
4. 从器件将 SDA 线拉为低电平，向主器件发送  $\overline{ACK}$  信号，并将 SSPIF 位置 1。
5. 用软件清零 SSPIF 位。
6. 软件从 SSPBUF 读接收到的地址，清零 BF 标志位。
7. 如果 SEN = 1，从软件将 CKP 位置 1 以释放 SCL 线。
8. 主器件随着时钟移出数据字节。
9. 从器件将 SDA 线驱动为低电平，向主器件发送 ACK 信号，并将 SSPIF 位置 1。
10. 软件清零 SSPIF 位。
11. 软件从 SSPBUF 读接收到的字节，清零 BF 位。
12. 对于所有从主器件接收的字节，重复第 8-12 步。
13. 主器件发送停止条件，将 SSPSTAT 的 P 位置 1，总线进入空闲状态。

### 24.5.2.2 配置了 AHEN 和 DHEN 位的 7 位接收

AHEN 和 DHEN 位置 1 的从器件接收，与未配置这两个选项的从器件接收的工作原理相同，但在 SCL 信号的第 8 个下降沿之后添加了额外中断和时钟延长。这些额外的中断允许从软件（而不是硬件）决定它是否想要对接收地址或数据字节产生 ACK 信号。此功能增加了对本模块以前版本中没有的 PMBus™ 的支持。

以下列表描述了从软件使用这些选项进行 I<sup>2</sup>C 通信必须采取的步骤。图 24-15 给出了使用地址和数据保持功能的模块。图 24-16 包括了 SSPCON2 寄存器的 SEN 位置 1 条件下的操作。

1. SSPSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPIF 位也置 1。
2. R/W 位清零的匹配地址随着时钟移入。在 SCL 信号的第 8 个下降沿之后 SSPIF 位置 1 且 CKP 位清零。
3. 从器件清零 SSPIF 位。
4. 从器件可查看 SSPCON3 寄存器的 ACKTIM 位，以决定 SSPIF 位是在 ACK 信号之前还是之后置 1。
5. 从器件从 SSPBUF 读地址值，清零 BF 标志位。
6. 从器件通过设置 ACKDT 位来设置随着时钟移出到主器件的 ACK 值。
7. 从器件通过将 CKP 位置 1 来释放时钟。
8. SSPIF 位在  $\overline{ACK}$  信号（而不是 NACK 信号）之后置 1。
9. 如果 SEN = 1，则从硬件在  $\overline{ACK}$  信号之后延长时钟。
10. 从器件清零 SSPIF。

**注：** 即使在有时钟延长且 BF 位清零的情况下，SSPIF 位在 SCL 信号的第 9 个下降沿之后仍然置 1。仅在向主器件发送了 NACK 信号的情况下，SSPIF 位才不会置 1。

11. 对于接收到的数据字节，在 SCL 信号的第 8 个下降沿之后 SSPIF 位置 1 且 CKP 位清零。
12. 从器件查看 SSPCON3 的 ACKTIM 位以决定中断源。
13. 从器件从 SSPBUF 读接收到的数据，清零 BF 位。
14. 对于每个接收到的数据字节，重复第 7-14 步。
15. 从器件发送  $\overline{ACK} = 1$ ，或主器件发送停止条件可结束通信。如果发送了停止条件且禁止在检测到停止条件时中断，则从器件只能通过查询 SSTSTAT 寄存器的 P 位来了解通信是否结束。

# PIC16(L)F1933

图 24-14: I<sup>2</sup>C 从模式接收时序 (SEN = 0, AHEN = 0, DHEN = 0, 7 位地址)

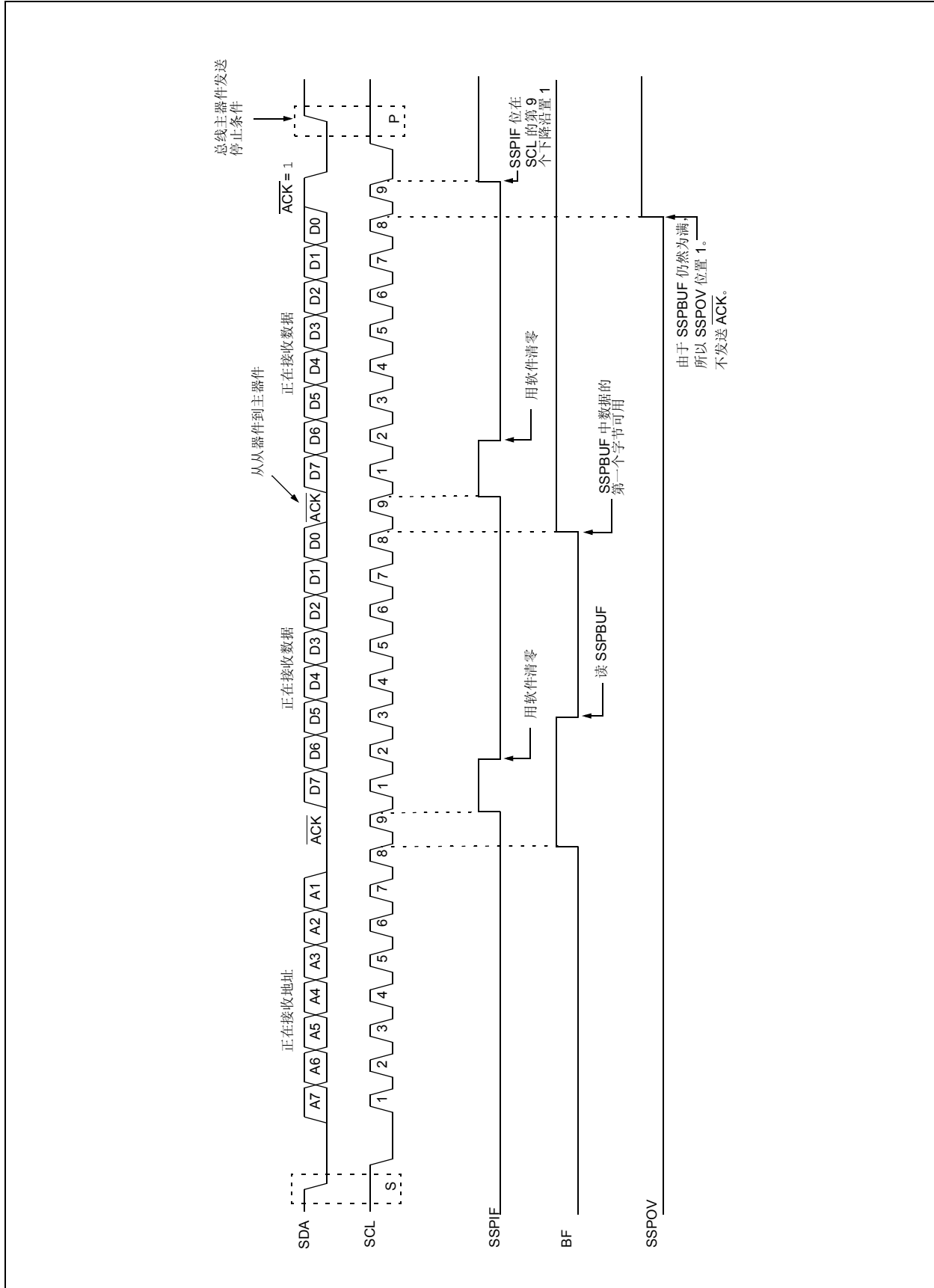
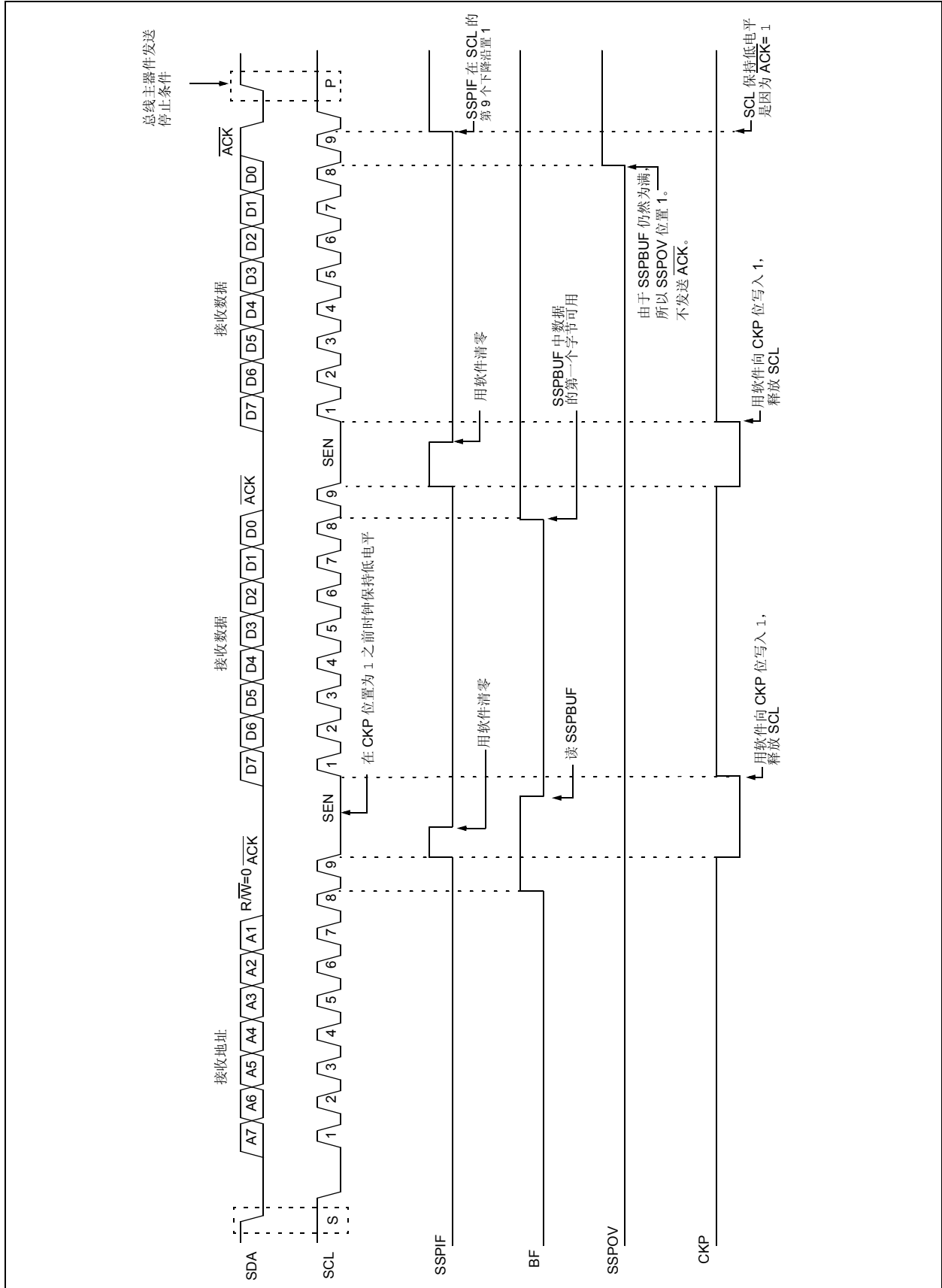


图 24-15: I<sup>2</sup>C 从模式接收时序 (SEN = 1, AHEN = 0, DHEN = 0, 7 位地址)



# PIC16(L)F1933

图 24-16: I<sup>2</sup>C 从模式接收时序 (SEN = 0, AHEN = 1, DHEN = 1, 7 位地址)

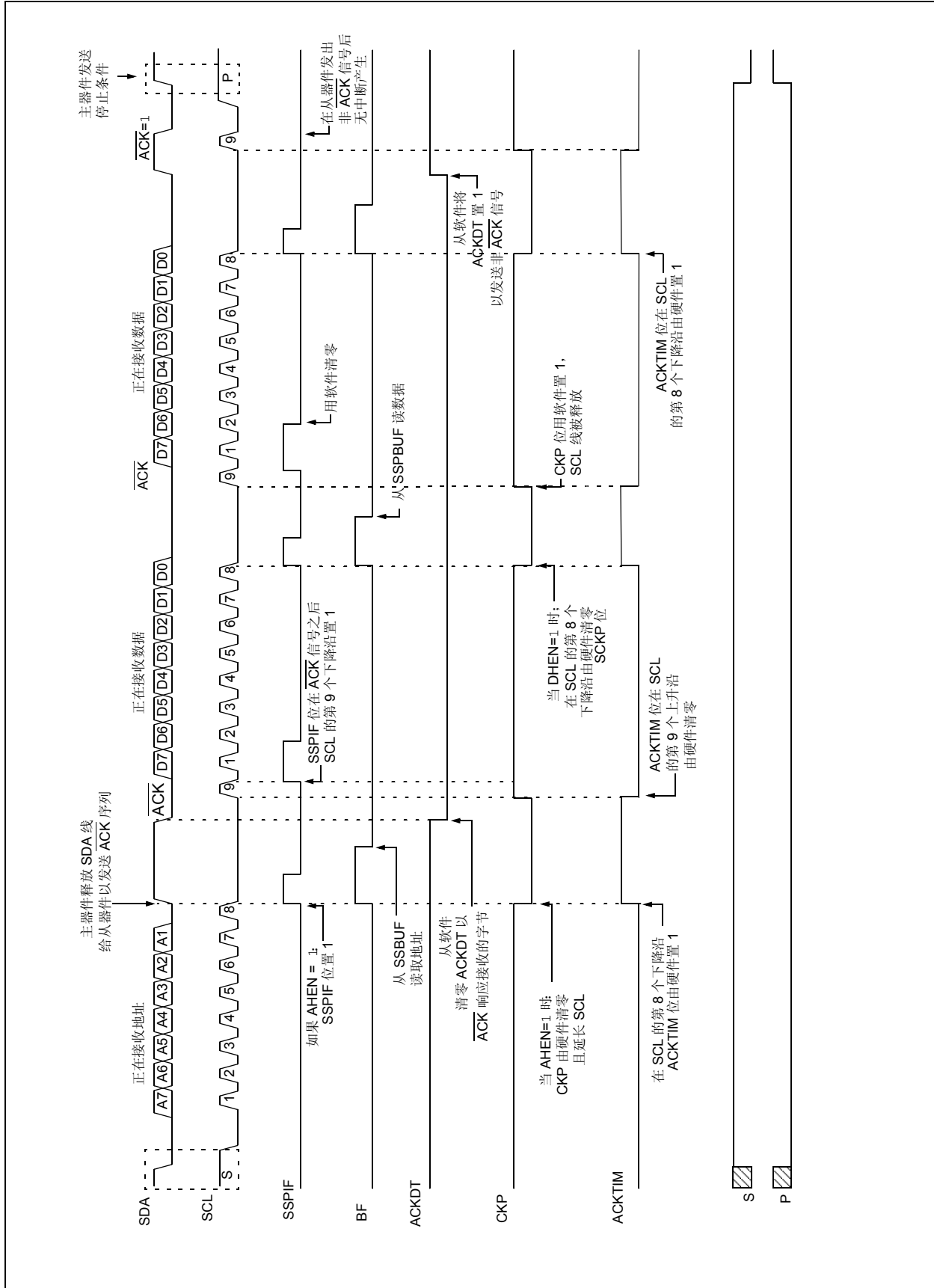
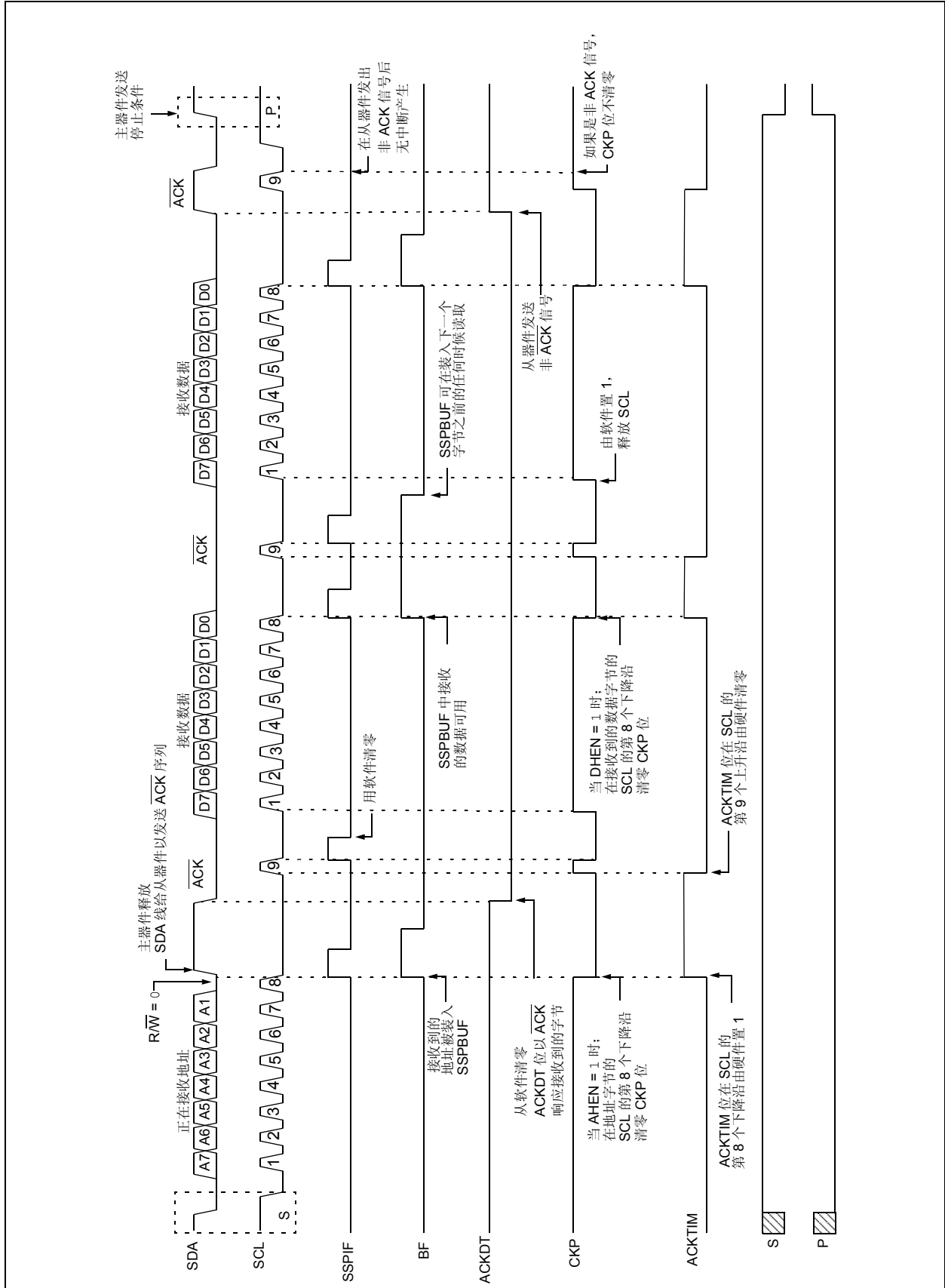




图 24-17: I<sup>2</sup>C 从模式接收时序 (SEN = 1, AHEN = 1, DHEN = 1, 7 位地址)



## 24.5.3 从发送

当传入的地址字节的  $\overline{R/W}$  位置 1 并发生地址匹配时，SSPSTAT 寄存器的  $\overline{R/W}$  位置 1。接收的地址装入 SSPBUF 寄存器，且在第 9 位由从器件发送一个  $\overline{ACK}$  脉冲信号。

在  $\overline{ACK}$  信号之后，从硬件清零 CKP 位且 SCL 引脚保持低电平（更多详细信息，请参见第 24.5.6 节“时钟延长”）。通过延长时钟，主器件在从器件准备发送数据之前不能发送其他时钟脉冲信号。

发送数据必须装入 SSPBUF 寄存器，同时也装入 SSPSR 寄存器。然后，应通过将 SSPCON1 寄存器的 CKP 位置 1 来释放 SCL 引脚。8 个数据位在 SCL 输入信号的下降沿被移出。这可以确保在 SCL 为高电平期间 SDA 信号是有效的。

主接收器的  $\overline{ACK}$  脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。此  $\overline{ACK}$  值将复制到 SSPCON2 寄存器的 ACKSTAT 位。如果 ACKSTAT 位置 1（非  $\overline{ACK}$ ），则数据传送完成。这种情况下，当从器件锁存了非  $\overline{ACK}$  值时，从器件进入空闲模式，等待出现下一个启动位。如果 SDA 线为低电平（ $\overline{ACK}$ ），则将下一个发送数据装入 SSPBUF 寄存器。同样，必须再次将 CKP 位置 1 来释放 SCL 引脚。

传送的每个数据字节都会产生 MSSP 中断。SSPIF 位必须用软件清零，SSPSTAT 寄存器用于确定字节的状态。SSPIF 位在第 9 个时钟脉冲的下降沿被置 1。

### 24.5.3.1 从模式总线冲突

从器件接收到读请求并开始从 SDA 线上移出数据。如果检测到总线冲突并且 SSPCON3 寄存器的 SBCDE 位置 1，那么 PIR 寄存器的 BCLIF 位置 1。检测到总线冲突时，从器件进入空闲状态，等待再次被寻址。用户软件可以使用 BCLIF 位处理从器件总线冲突。

### 24.5.3.2 7 位发送

主器件可以向从器件发送读请求，然后使数据随着时钟移出从器件。下面的列表罗列了从器件完成标准发送所需的软件。图 24-17 可用于辅助说明此列表。

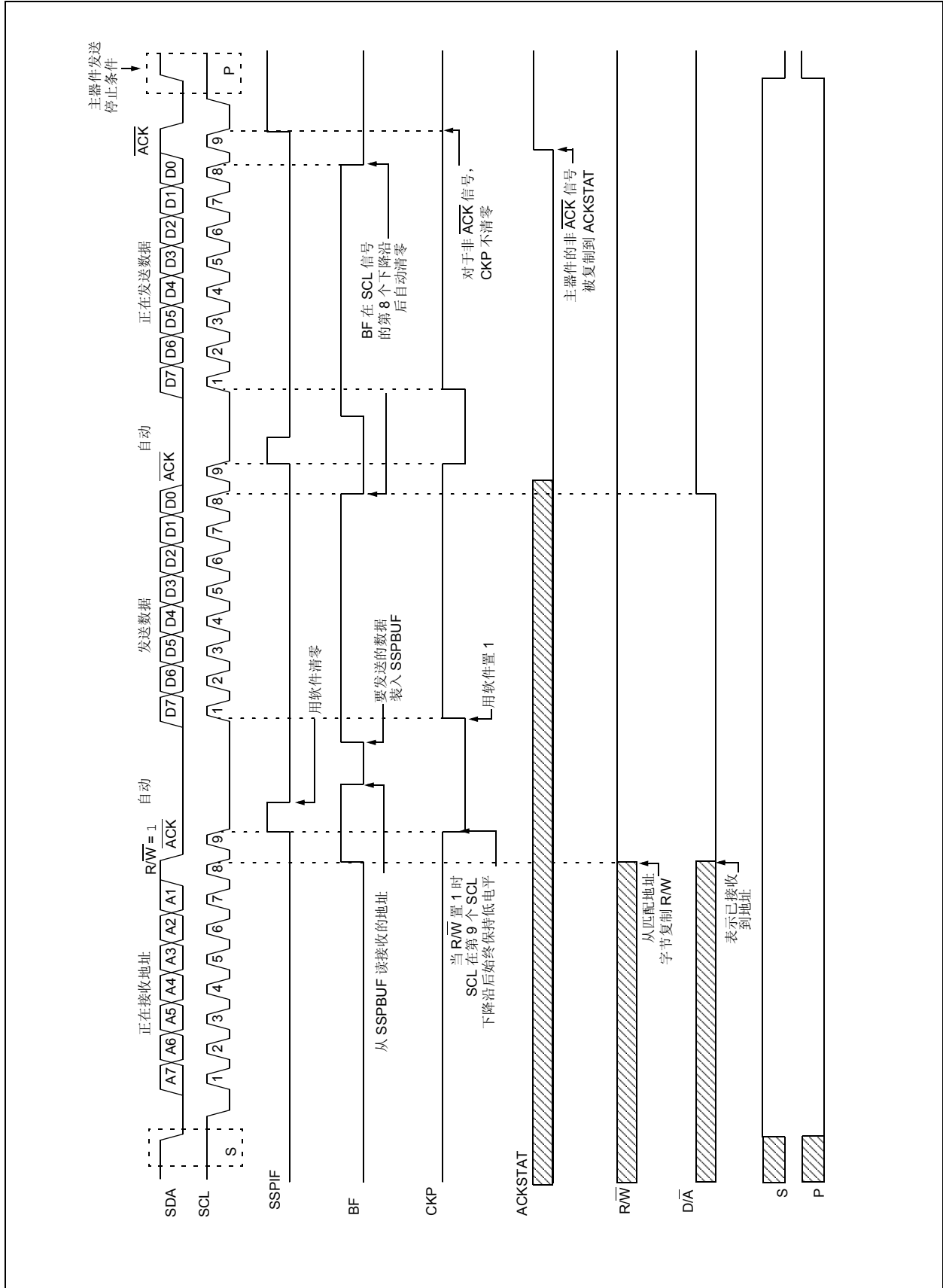
1. 主器件通过 SDA 和 SCL 线发送启动条件。
2. SSPSTAT 的 S 位置 1；如果允许在检测到启动条件时产生中断，则 SSPIF 位也置 1。
3. 从器件接收  $\overline{R/W}$  位置 1 的匹配地址，以将 SSPIF 位置 1。
4. 从硬件产生  $\overline{ACK}$  信号并将 SSPIF 位置 1。
5. SSPIF 位由用户清零。
6. 软件从 SSPBUF 读取接收到的地址，清零 BF 位。
7.  $\overline{R/W}$  位置 1，因此 CKP 位在  $\overline{ACK}$  之后自动清零。
8. 从软件将发送数据装入 SSPBUF。
9. CKP 位置 1，释放 SCL 线，从而允许主器件将数据随着时钟移出从器件。
10. SSPIF 位在来自主器件的  $\overline{ACK}$  响应装入 ACKSTAT 寄存器之后置 1。
11. SSPIF 位清零。
12. 从软件检查 ACKSTAT 位以查看主器件是否想随着时钟移出更多的数据。

**注 1:** 主器件发送了  $\overline{ACK}$  信号时，时钟被延长。

**2:** ACKSTAT 是在 SCL 信号的第 9 个上升沿（而不是下降沿）上惟一被更新的位。

13. 对于每个发送的字节，重复第 9-13 步。
14. 如果主器件发送了非  $\overline{ACK}$  信号，则时钟不被延长，但是 SSPIF 位仍然置 1。
15. 主器件发送重复启动条件或停止条件。
16. 从器件不再被寻址。

图 24-18: I<sup>2</sup>C 从模式发送时序 (AHEN = 0, 7 位地址)



# PIC16(L)F1933

## 24.5.3.3 使能了地址保持的 7 位发送

将 SSPCON3 寄存器的 AHEN 位置 1 可以在接收的匹配地址的第 8 个下降沿后增加时钟延长并产生中断。随着时钟移入匹配地址后，清零 CKP 并将 SSPIF 中断标志位置 1。

图 24-18 给出了使能了 AHEN 时 7 位地址从发送的标准波形。

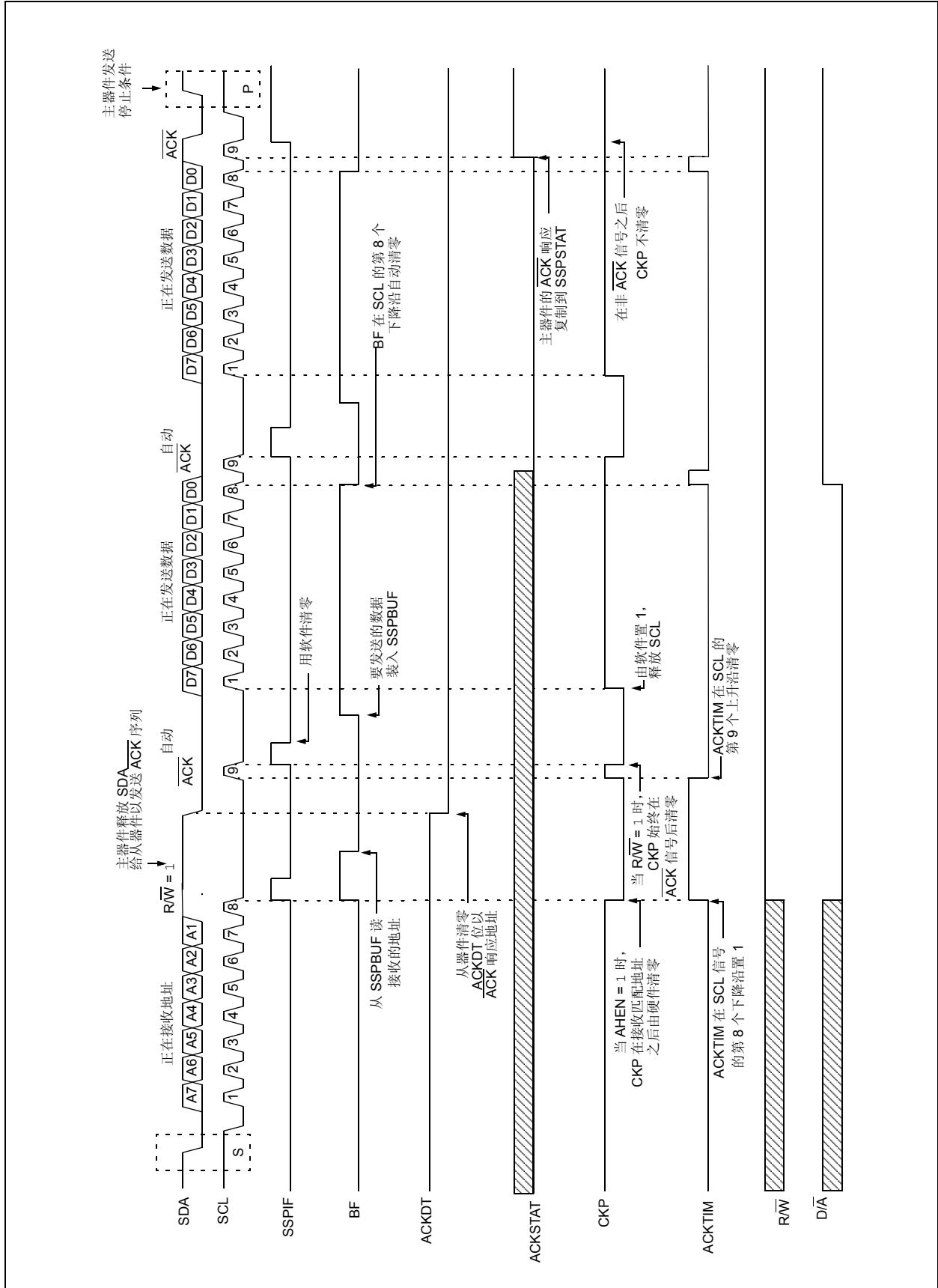
1. 总线启动空闲模式。
2. 主器件发送启动条件； SSPSTAT 的 S 位置 1；如果允许在检测到启动条件时中断，则 SSPIF 位也置 1。
3. 主器件发送  $\overline{R/W}$  位置 1 的匹配地址。在 SCL 线的第 8 个下降沿后，CKP 位清零，并产生 SSPIF 中断。
4. 从软件清零 SSPIF。
5. 从软件读取 SSPCON3 寄存器的 ACKTIM 位以及 SSPSTAT 寄存器的 R/W 和 D/A 位，以确定中断源。
6. 从器件从 SSPBUF 寄存器读地址值，清零 BF 位。
7. 从软件根据此信息决定是发送  $\overline{ACK}$  信号还是非  $\overline{ACK}$  信号，并相应地设置 SSPCON2 寄存器的 ACKDT 位。
8. 从器件将 CKP 位置 1，以释放 SCL。
9. 主器件随着时钟移入来自从器件的  $\overline{ACK}$  值。
10. 如果  $\overline{R/W}$  位置 1，则从硬件在  $\overline{ACK}$  之后自动清零 CKP 位并将 SSPIF 置 1。
11. 从软件清零 SSPIF。
12. 从器件将要发送给主器件的值装入 SSPBUF，从而将 BF 位置 1。

**注：** SSPBUF 在接收到  $\overline{ACK}$  之后才能装入数据。

13. 从器件将 CKP 位置 1 以释放时钟。
14. 主器件随着时钟将数据移出从器件，并在第 9 个 SCL 脉冲发送  $\overline{ACK}$  值。
15. 从硬件将  $\overline{ACK}$  值复制到 SSPCON2 寄存器的 ACKSTAT 位。
16. 对于每个从从器件发送到主器件的字节，重复第 10-15 步。
17. 如果主器件发送非  $\overline{ACK}$  值，从器件将释放总线以允许主器件发送停止位并结束通信。

**注：** 主器件必须在最后一个字节发送非  $\overline{ACK}$  值，以确保从器件释放 SCL 线以接收停止位。

图 24-19: I<sup>2</sup>C 从模式发送时序 (AHEN = 1, 7 位地址)



# PIC16(L)F1933

## 24.5.4 从模式 10 位地址接收

本节描述了 MSSP 模块配置为工作在 10 位寻址模式下的 I<sup>2</sup>C 从器件时的标准事件序列。

图 24-19 用直观的方式对此配置作了说明。

这是从软件完成 I<sup>2</sup>C 通信所必需的详细步骤。

1. 总线启动空闲模式。
2. 主器件发送启动条件；SSPSTAT 的 S 位置 1；如果允许在检测到启动条件时中断，则 SSPIF 位也置 1。
3. 主器件发送  $\overline{R/W}$  位清零的匹配高地址，SSPSTAT 寄存器的 UA 位置 1。
4. 从器件发送  $\overline{ACK}$ ，SSPIF 置 1。
5. 用软件清零 SSPIF 位。
6. 软件从 SSPBUF 读接收到的地址，清零 BF 标志位。
7. 从器件将低地址字节装入 SSPADD，释放 SCL。
8. 主器件将匹配的低地址字节发送给从器件，UA 位置 1。

**注：** 只有在  $\overline{ACK}$  序列后，才允许更新 SSPADD 寄存器。

9. 从器件发送  $\overline{ACK}$ ，SSPIF 置 1。

**注：** 如果低地址不匹配，SSPIF 和 UA 仍然置 1，以便从软件可以将 SSPADD 设置回高地址。BF 不置 1，因为没有发生匹配。CKP 不受影响。

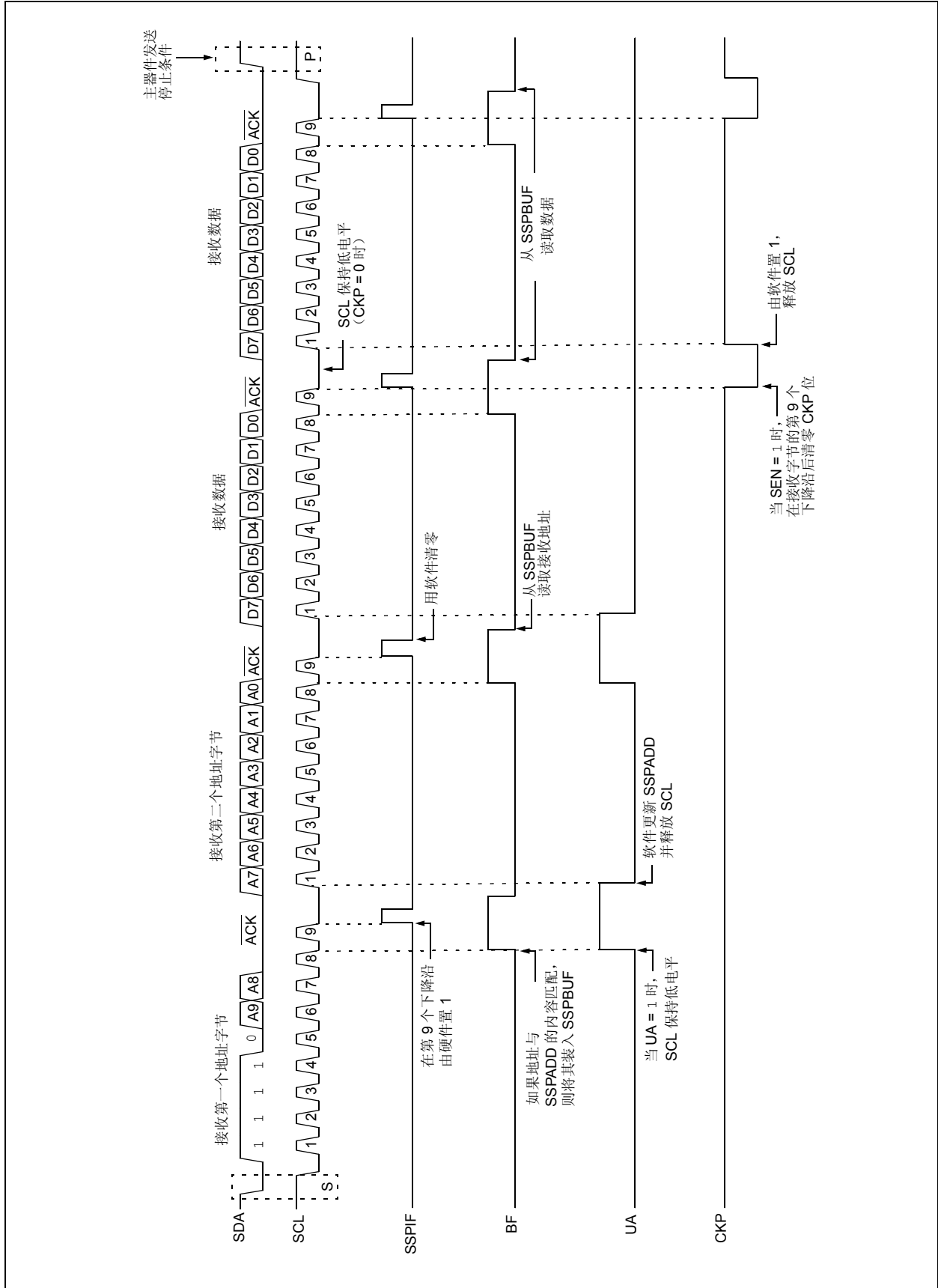
10. 从器件清零 SSPIF。
11. 从器件从 SSPBUF 读接收到的匹配地址，清零 BF 位。
12. 从器件将高地址字节装入 SSPADD。
13. 主器件随着时钟将数据字节移入从器件，并在第 9 个 SCL 脉冲随着时钟将  $\overline{ACK}$  移出从器件；SSPIF 置 1。
14. 如果 SSPCON2 的 SEN 位置 1，那么 CKP 由硬件清零且时钟被延长。
15. 从器件清零 SSPIF。
16. 从器件从 SSPBUF 读接收到的字节，清零 BF 位。
17. 如果 SEN 置 1，从器件会将 CKP 置 1 以释放 SCL。
18. 对于每个接收到的字节，重复第 13-17 步。
19. 主器件发送停止位以结束发送。

## 24.5.5 带地址或数据保持的 10 位寻址

使用 AHEN 或 DHEN 位置 1 的 10 位寻址的接收操作与 7 位模式下的接收相同，惟一区别是需要使用 UA 位更新 SSPADD 寄存器。所有功能，特别在 CKP 位清零且 SCL 线保持低电平时都是相同的。图 24-20 可辅助说明 AHEN 位置 1 时从器件在 10 位寻址模式下的工作。

图 24-21 提供了 10 位寻址模式下从发送器的标准波形。

图 24-20: I<sup>2</sup>C 从模式接收时序 (SEN = 1, AHEN = 0, DHEN = 0, 10 位地址)



# PIC16(L)F1933

图 24-21: I<sup>2</sup>C 从模式接收时序 (SEN = 0, AHEN = 1, DHEN = 0, 10 位地址)

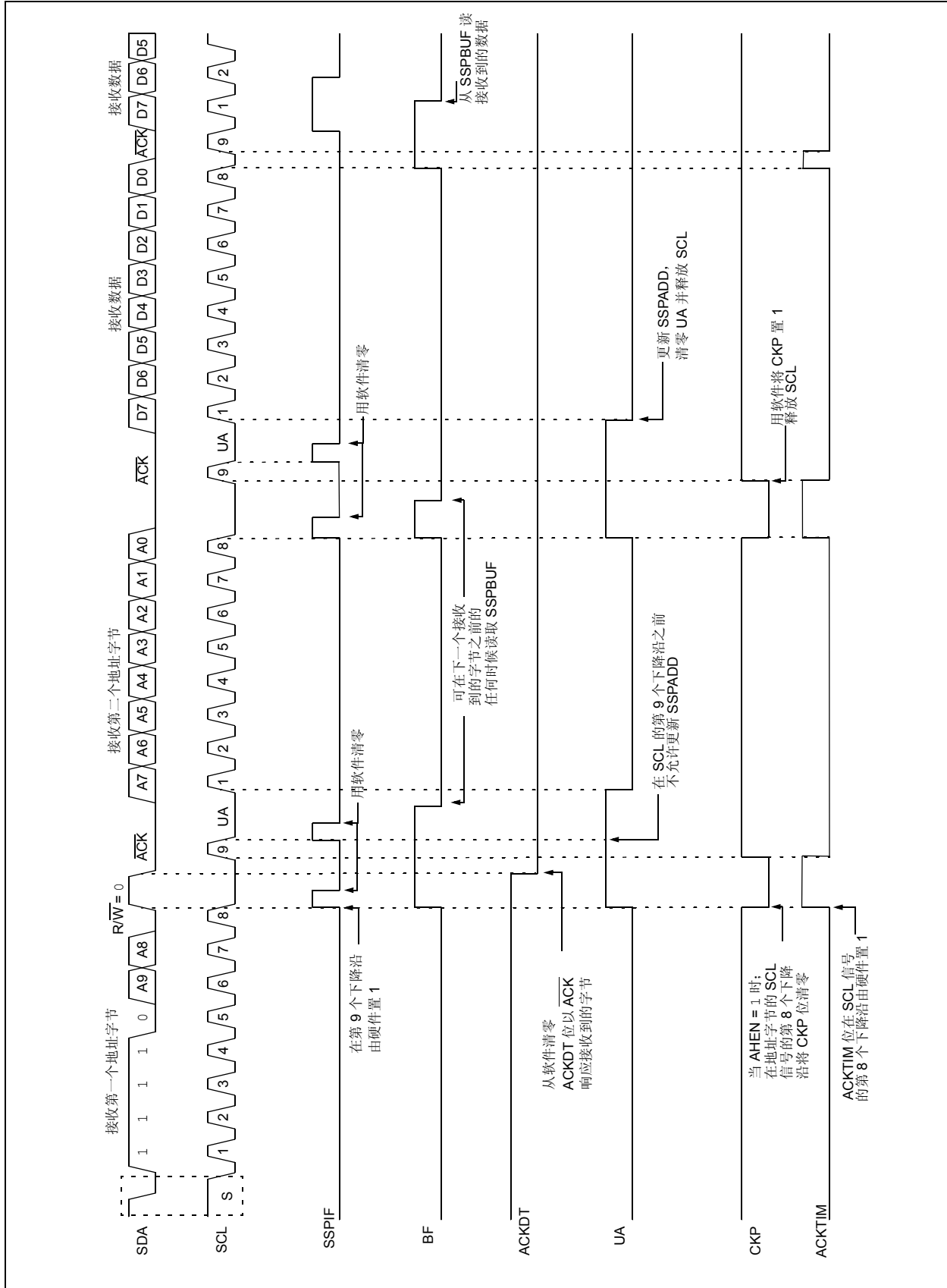
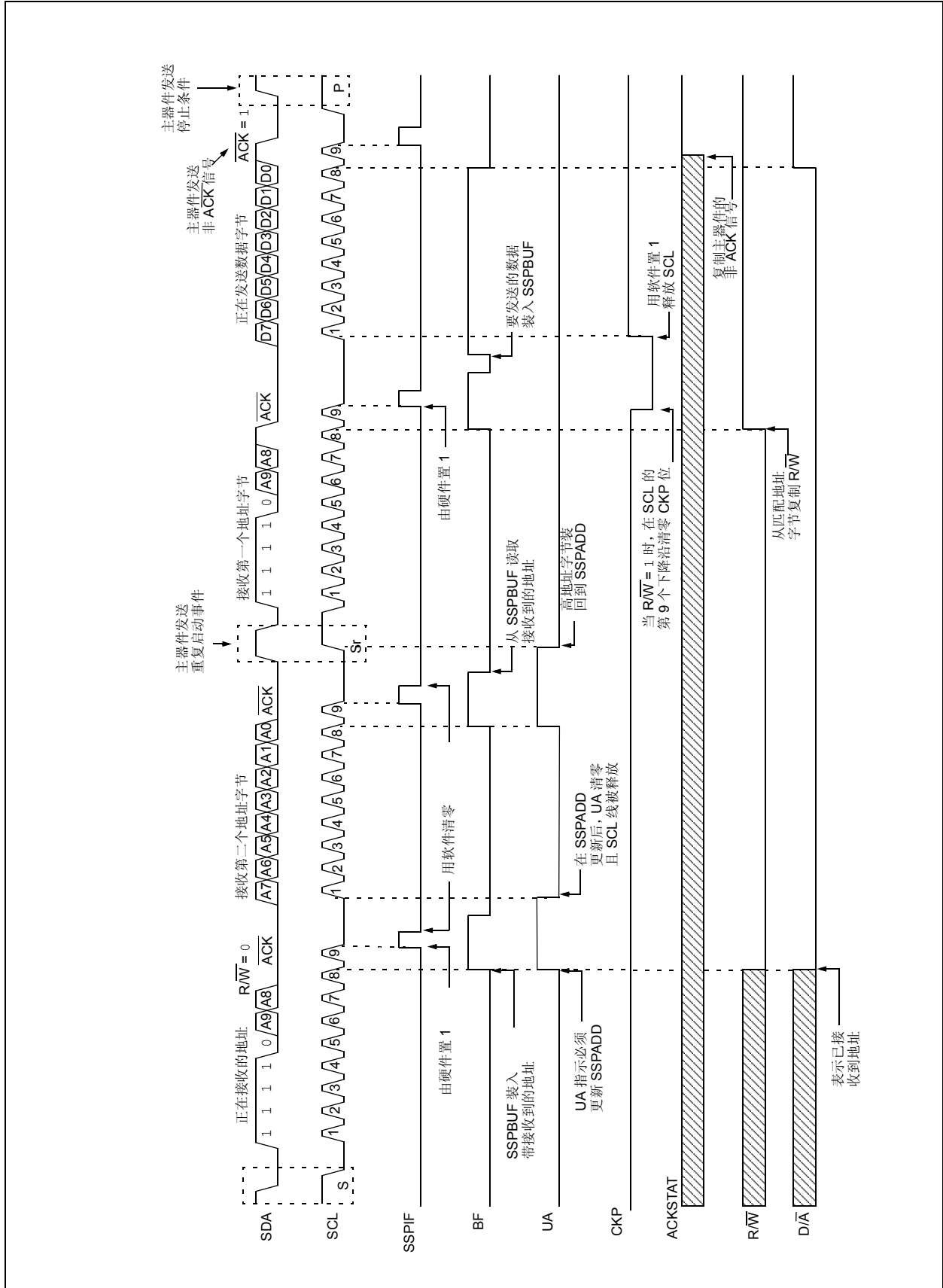




图 24-22: I<sup>2</sup>C 从模式发送时序 (SEN = 0, AHEN = 0, DHEN = 0, 10 位地址)



# PIC16(L)F1933

## 24.5.6 时钟延长

当总线上的器件保持 SCL 线为低电平以有效暂停通信时，发生时钟延长。从器件必须延长时钟以允许有更多的时间来处理数据或准备响应主器件。时钟延长时不关心主器件的工作，任何时候只需总线上主器件处于活动状态但是不传输数据就可以被认为是时钟延长。从器件所执行的任何延长对于主软件都是不可见的，且由产生 SCL 信号的硬件进行处理。

SSPCON1 寄存器的 CKP 位用于用软件控制延长。任何时候 CKP 位清零都会使模块等待 SCL 线变为低电平，然后保持低电平状态不变。将 CKP 置 1 将释放 SCL 并允许更多通信。

### 24.5.6.1 正常的时钟延长

如果 SSPSTAT 的 R/W 位置 1（表示读请求），那么在 ACK 之后，从硬件将清零 CKP。这使得从器件有时间使用传送到主器件的数据更新 SSPBUF。如果 SSPCON2 的 SEN 位置 1，从硬件将在 ACK 序列后始终延长时钟。从器件就绪后，CKP 由软件置 1，通信继续。

- 注 1:** BF 位对于是否延长时钟没有影响。这与模块之前的版本不同，如果在 SCL 的第 9 个下降沿之前读取 SSPBUF，模块之前的版本不会延长时钟、清零 CKP。
- 2:** 对于发送操作，如果在 SCL 的第 9 个下降沿之前装入 SSPBUF，模块之前的版本不会延长时钟。对于读请求，CKP 此时将始终清零。

### 24.5.6.2 10 位寻址模式

在 10 位寻址模式下，UA 位置 1 时，时钟始终延长。只有在这时才会在 CKP 未清零时延长 SCL。在写入 SSPADD 后立即释放 SCL。

**注:** 如果第二个地址字节不匹配，模块之前的版本不会延长时钟。

### 24.5.6.3 不应答字节

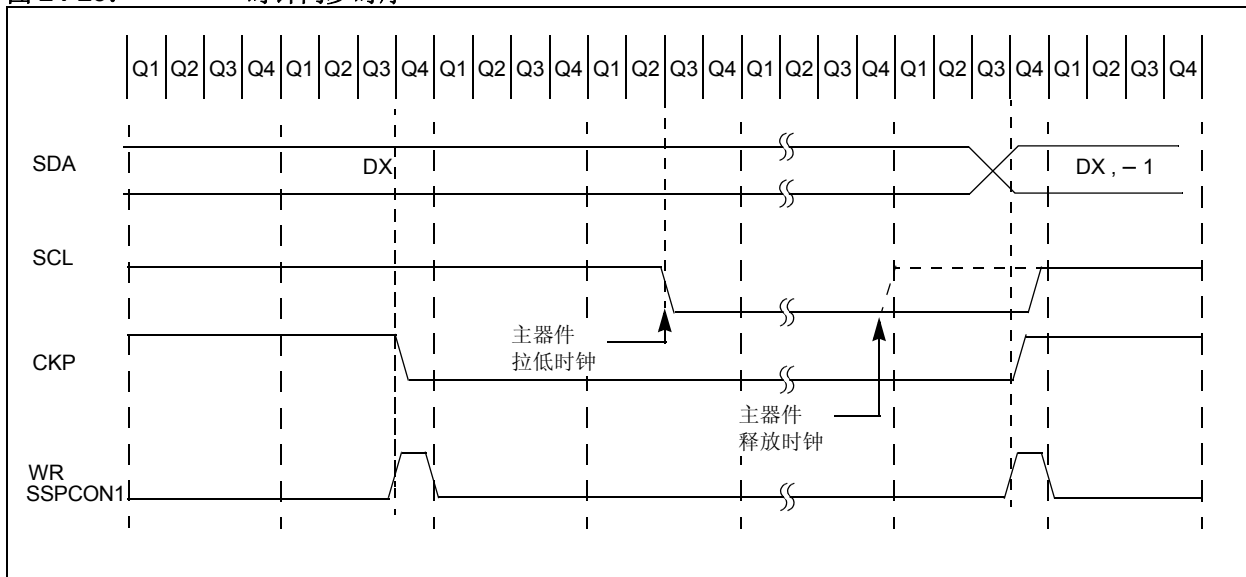
SSPCON3 的 AHEN 位置 1 时，在接收到的匹配地址字节的 SCL 信号的第 8 个下降沿之后 CKP 位由硬件清零。SSPCON3 的 DHEN 位置 1 时，在接收到的数据的 SCL 信号的第 8 个下降沿之后清零 CKP 位。

在 SCL 信号的第 8 个下降沿之后延长时钟，可以使从器件能够查看接收到的地址或数据，并决定它是否应答接收的数据。

### 24.5.7 时钟同步和 CKP 位

任何时候 CKP 位清零都会使模块等待 SCL 线变为低电平，然后保持低电平状态不变。但是，清零 CKP 位不会在 SCL 输出已采样到低电平之前将 SCL 输出信号拉低。因此，CKP 位不会在外部 I<sup>2</sup>C 主器件已拉低 SCL 线之前拉低 SCL 线。SCL 输出在 CKP 位置 1 并且 I<sup>2</sup>C 总线上的所有其他器件已释放 SCL 之前将保持低电平。这可以确保对 CKP 位的写操作不会违反 SCL 的最小高电平时间要求（见图 24-22）。

图 24-23: 时钟同步时序



## 24.5.8 广播呼叫地址支持

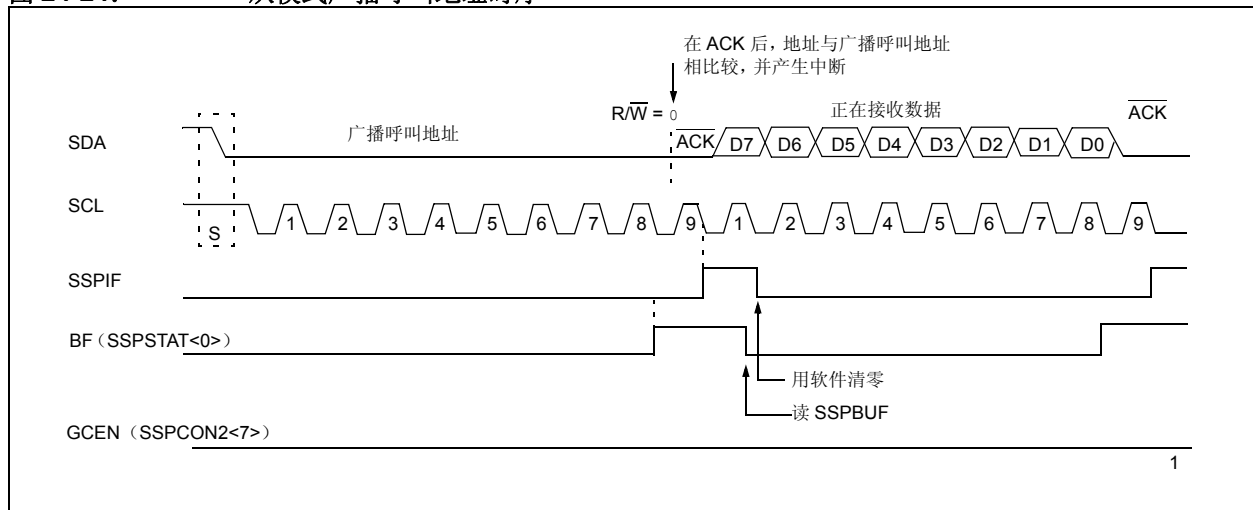
在 I<sup>2</sup>C 总线的寻址过程中，启动条件后的第一个字节通常确定了哪个器件是主器件寻址的从器件。但可寻址所有器件的广播呼叫地址除外。使用此地址时，理论上所有器件都应该给出一个应答信号作为响应。

在 I<sup>2</sup>C 协议中，广播呼叫地址是保留地址，定义为地址 0x00。SSPCON2 寄存器的 GCEN 位置 1 时，不管存储在 SSPADD 中的值如何，从模块将自动以 ACK 信号响应接收到此地址。在从器件随着时钟移入 R/W 位清零的全零地址后，将产生中断，且从软件可以读取 SSPBUF 并做出响应。图 24-23 给出了广播呼叫接收序列。

在 10 位地址模式下，接收到广播呼叫地址时，UA 位不会置 1。从器件将准备接收作为数据的第二个字节，就像在 7 位模式下一样。

如果 SSPCON3 寄存器的 AHEN 位置 1，从硬件会接收任何其他地址一样，在 SCL 信号的第 8 个下降沿之后延长时钟。然后，从器件必须将其 ACKDT 位置 1 并释放时钟，通信过程恢复正常。

图 24-24: 从模式广播呼叫地址时序



## 24.5.9 SSP 掩码寄存器

SSP 掩码 (SSPMSK) 寄存器 (寄存器 24-5) 在 I<sup>2</sup>C 从模式下可用，用于在地址比较操作期间屏蔽 SSPSR 寄存器中保存的值。SSPMSK 寄存器中的零 (0) 位可使接收地址的相应位变为“无关位”。

任何复位条件都可将此寄存器复位为全 1，因此在写入掩码值之前对标准 SSP 操作没有影响。

SSP 掩码寄存器在以下模式下有效：

- 7 位地址模式：用于比较地址的 A<7:1>。
- 10 位地址模式：仅用于比较地址的 A<7:0>。在接收地址的第一个（高）字节期间，SSP 掩码寄存器没有影响。

## 24.6 I<sup>2</sup>C 主模式

主模式可通过以下方法使能：将 SSPCON1 寄存器中的相应 SSPM 位置 1 和清零并将 SSPEN 位置 1。在主模式下，SDA 和 SCK 线必须配置为输入。如果需要将该引脚驱动为低电平，则 MSSP 外设硬件将覆写由 TRIS 控制的输出驱动器。

主操作模式支持在检测到启动和停止条件时产生中断。复位或禁止 MSSP 模块时可以清零停止 (P) 和启动 (S) 位。在 P 位置 1 时可获得 I<sup>2</sup>C 总线的控制权，否则总线空闲。

在固件控制的主模式下，用户代码根据启动和停止位条件检测进行所有 I<sup>2</sup>C 总线操作。启动和停止条件检测是此模式下惟一工作的电路。可通过用户软件直接操作 SDA 和 SCL 线来进行所有其他通信。

以下事件将导致 SSP 中断标志位 SSPIF 置 1（如果允许 SSP 中断，则产生中断）：

- 检测到启动条件
- 检测到停止条件
- 发送 / 接收到数据传送字节
- 发送 / 接收到应答
- 产生了重复启动条件

**注 1:** MSSP 模块配置为工作在 I<sup>2</sup>C 主模式下时，不允许事件排队。例如，在启动条件完成之前不允许用户发出启动条件并立即写 SSPBUF 寄存器以启动发送。在这种情况下，不会写入 SSPBUF 且 WCOL 位将置 1，表示未发生对 SSPBUF 的写操作。

**2:** 在主模式下时，在 SEN/PEN 位清零时，会屏蔽启动 / 停止检测并将产生中断。

### 24.6.1 I<sup>2</sup>C 主模式工作原理

主器件产生所有串行时钟脉冲以及启动条件和停止条件。传输以停止条件或重复启动条件结束。由于重复启动条件也是下一次串行传输的开始，所以不会释放 I<sup>2</sup>C 总线。

在主器件发送模式下，串行数据通过 SDA 输出，而 SCL 输出串行时钟。发送的第一个字节包含接收器件的从地址（7 位）和读 / 写 (R/W) 位。在这种情况下，R/W 位为逻辑 0。每次发送 8 位串行数据。发送完每个字节后，将接收到应答位。输出启动和停止条件以指示串行传输的开始和结束。

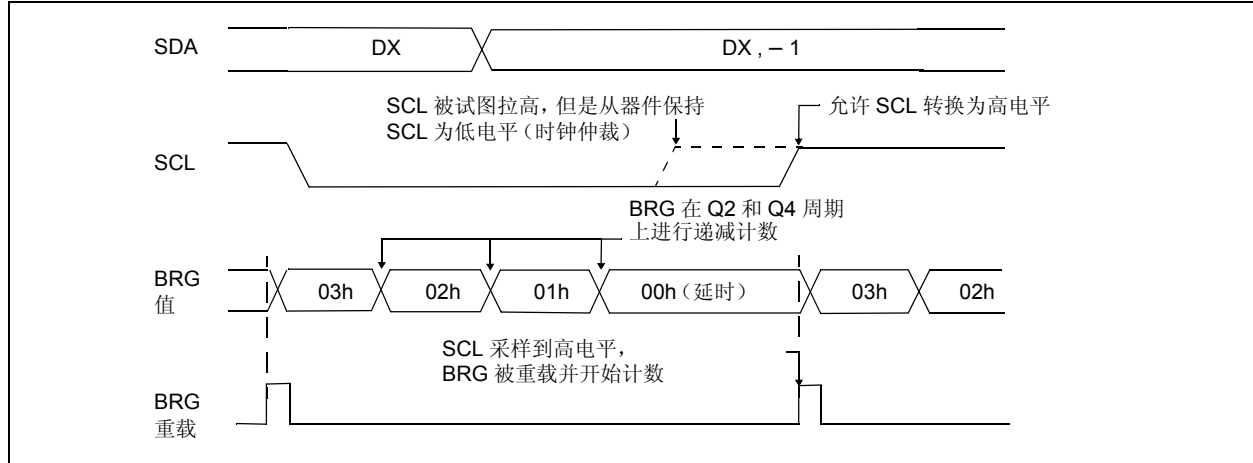
在主器件接收模式下，第一个发送的字节包含发送器件的从地址（7 位）和 R/W 位。在这种情况下，R/W 位为逻辑 1。因此，第一个发送的字节是 7 位从地址以及随后用于指示接收位的 1。通过 SDA 接收串行数据，而 SCL 输出串行时钟。每次接收 8 位串行数据。接收完每个字节后，将发送应答位。启动和停止条件表示发送的开始和结束。

波特率发生器用于设置 SCL 上输出的时钟频率。更多详细信息，请参见第 24.7 节“波特率发生器”。

## 24.6.2 时钟仲裁

如果在任何接收、发送或重复启动 / 停止条件期间，主器件释放了 SCL 引脚（SCL 允许悬空为高电平）时发生时钟仲裁。SCL 引脚允许悬空为高电平时，在 SCL 引脚上实际采样到高电平之前，波特率发生器（Baud Rate Generator, BRG）暂停计数。在 SCL 引脚上采样到高电平时，波特率发生器重新装入 SSPADD<7:0> 的内容并开始计数。这可以确保当外部器件将时钟保持为低电平时，始终保持 SCL 高电平时间至少为一个 BRG 计满返回计数周期（图 24-25）。

**图 24-25: 时钟仲裁时的波特率发生器时序**



## 24.6.3 WCOL 状态标志

如果用户在启动、重复启动、停止、接收或发送序列过程中时写入 SSPBUF，那么 WCOL 将置 1，且缓冲器的内容保持不变（未发生写操作）。任何时候 WCOL 位置 1 都表示在模块未处于空闲时尝试对 SSPBUF 执行某种动作。

**注：** 由于不允许事件排队，在启动条件完成之前，禁止写入 SSPCON2 的低 5 位。

# PIC16(L)F1933

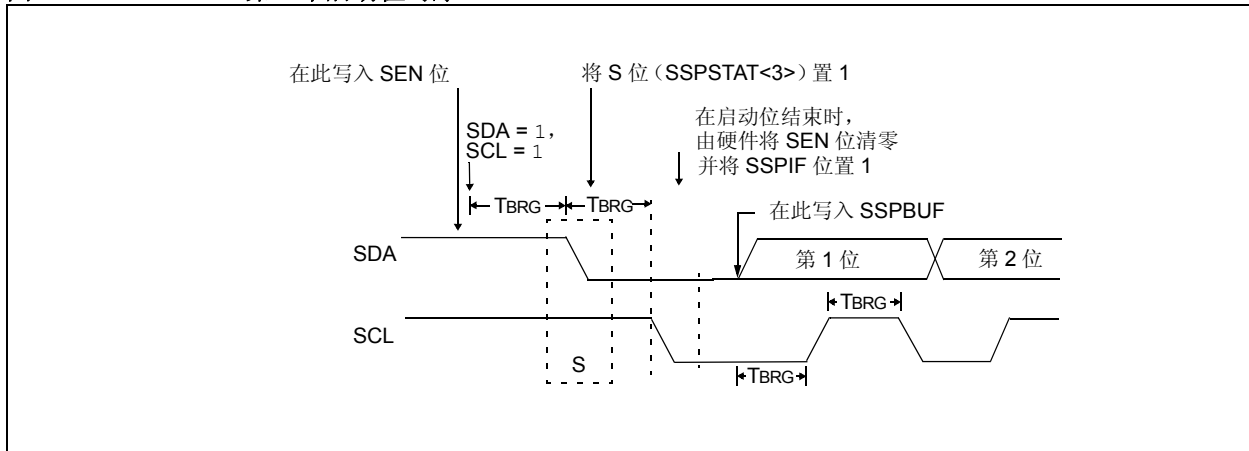
## 24.6.4 I<sup>2</sup>C 主模式启动条件时序

要发起启动条件，用户要将 SSPCON2 寄存器的启动使能位 SEN 置 1。如果采样到 SDA 和 SCL 引脚为高电平，则波特率发生器重新装入 SSPADD<7:0> 的内容并开始计数。如果在波特率发生器超时 (TBRG) 时采样到 SDA 和 SCL 引脚都为高电平，那么 SDA 引脚将驱动为低电平。当 SCL 为高电平时驱动 SDA 为低电平的操作就是启动条件，将导致 SSPSTAT1 寄存器的 S 位置 1。然后，波特率发生器将重新装入 SSPADD<7:0>

的内容并继续计数。波特率发生器超时 (TBRG) 时，SSPCON2 寄存器的 SEN 位将由硬件自动清零；波特率发生器暂停，SDA 线保持低电平且启动条件完成。

- 注 1:** 如果在启动条件开始时，已采样到 SDA 和 SCL 引脚为低电平，或者如果在启动条件期间，在 SDA 线驱动为低电平之前采样到 SCL 线为低电平，则发生总线冲突，总线冲突中断标志 BCLIF 置 1，启动条件终止，且 I<sup>2</sup>C 模块复位为空闲状态。
- 2:** Philips I<sup>2</sup>C 规范规定启动时不能发生总线冲突。

图 24-26: 第一个启动位时序



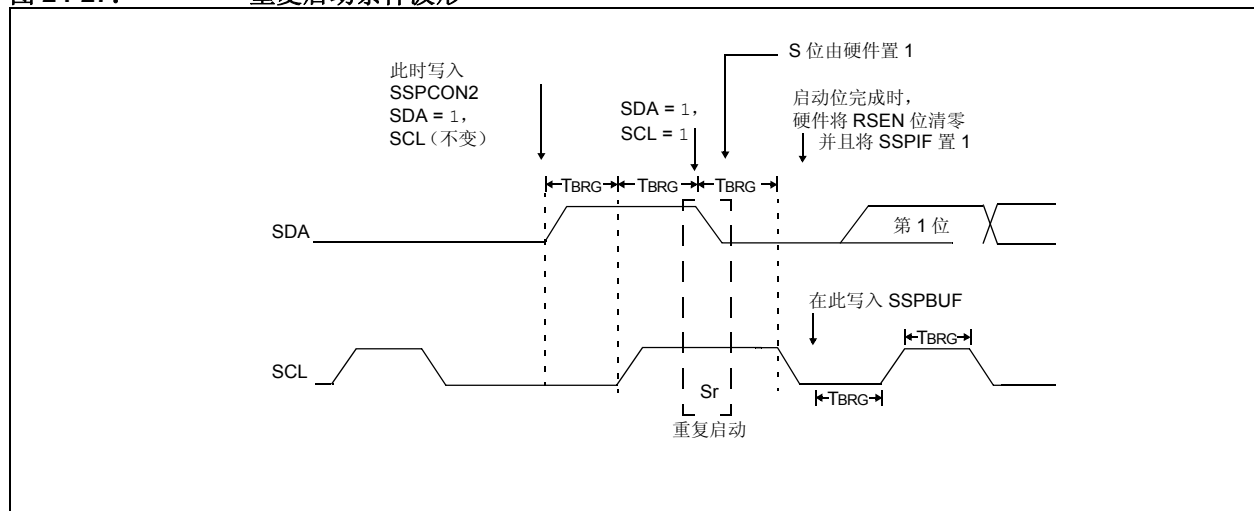
## 24.6.5 I<sup>2</sup>C 主模式重复启动条件时序

在 SSPCON2 寄存器的 RSEN 位编程为高电平并且主状态机不再有效时，产生重复启动条件。RSEN 位置 1 时，SCL 引脚拉为低电平。在采样到 SCL 引脚为低电平时，波特率发生器装入数据并开始计数。SDA 引脚在一个波特率发生器计数周期 (TBRG) 内被释放 (拉为高电平)。波特率发生器超时，如果采样到 SDA 为高电平，SCL 引脚将拉为高电平。采样到 SCL 为高电平时，波特率发生器重新装入数据并开始计数。SDA 和 SCL 必须在一个 TBRG 采样到高电平。随后在 SCL 引脚保持为高电平时，将 SDA 引脚拉为低电平 (SDA = 0) 一个 TBRG 周期。接下来 SCL 引脚被拉为低电平。随后，SSPCON2 寄存器的 RSEN 位将自动清零，波特率

发生器不会重新装入数据，保持 SDA 引脚为低电平。只要在 SDA 和 SCL 引脚上检测到启动条件，SSPSTAT 寄存器的 S 位就会置 1。在波特率发生器超时之前，SSPIF 位不会置 1。

- 注 1:** 如果在任何其他事件正在进行时编程 RSEN，此操作将不起作用。
- 注 2:** 在重复启动条件期间如果发生以下情况，将发生总线冲突：
- SCL 从低电平变为高电平时，采样到 SDA 为低电平。
  - 在 SDA 被拉低之前 SCL 变为低电平。这可能表示另一个主器件正在尝试发送数据 1。

图 24-27: 重复启动条件波形



## 24.6.6 I<sup>2</sup>C 主模式发送

数据字节、7 位地址或 10 位地址的另一半的发送都是通过简单的写入值到 SSPBUF 寄存器来完成。此操作会将缓冲器满标志位 BF 置 1，允许波特率发生器开始计数并开始下一次发送。地址 / 数据的每一位都在 SCL 信号的下降沿之后移出到 SDA 引脚。SCL 保持低电平的时间为一个波特率发生器计满返回计数周期 (TBRG)。数据应在 SCL 释放为高电平之前有效。当 SCL 引脚被释放为高电平时，它将在一个 TBRG 时间内保持不变。在此期间及 SCL 信号的下一个下降沿之后的一段时间内，SDA 引脚上的数据必须保持稳定。在第 8 位被移出 (第 8 个时钟的下降沿) 后，BF 标志清零，主器件释放 SDA。如果发生地址匹配或数据被正确接收，则被寻址的从器件在第 9 个位时间以一个 ACK 位作为响应。ACK 的状态将在第 9 个时钟的上升沿写入 ACKSTAT 位。如果主器件接收到应答，则应答状态位 ACKSTAT 将清零。否则，该位置 1。在第 9 个时钟后，SSPIF 位置 1 且主时钟 (波特率发生器) 暂停，之后在下一个数据字节装入 SSPBUF，保持 SCL 为低电平且 SDA 不变 (图 24-27)。

在写入 SSPBUF 之后，地址的每一位将在 SCL 的下降沿移出，直到 7 位地址和 R/W 位都被移出为止。在第 8 个时钟的下降沿，主器件将释放 SDA 引脚，允许从器件发出应答信号作为响应。在第 9 个时钟的下降沿，主器件将采样 SDA 引脚，以查看地址是否被从器件识别。ACK 位的状态将装入到 SSPCON2 寄存器的 ACKSTAT 状态位。在地址发送时钟的第 9 位的下降沿之后，SSPIF 置 1，BF 标志清零，并且波特率发生器关闭直到下一次写 SSPBUF，保持 SCL 为低电平并允许 SDA 悬空。

### 24.6.6.1 BF 状态标志

在发送模式下，SSPSTAT 寄存器的 BF 位在 CPU 写 SSPBUF 时置 1，在移出所有 8 位时清零。

### 24.6.6.2 WCOL 状态标志

如果用户在发送过程中 (即 SSPSR 仍在移出数据字节) 写 SSPBUF，则 WCOL 将置 1，且缓冲器的内容不变 (未发生写操作)。

在下次发送之前，WCOL 必须由软件清零。

### 24.6.6.3 ACKSTAT 状态标志

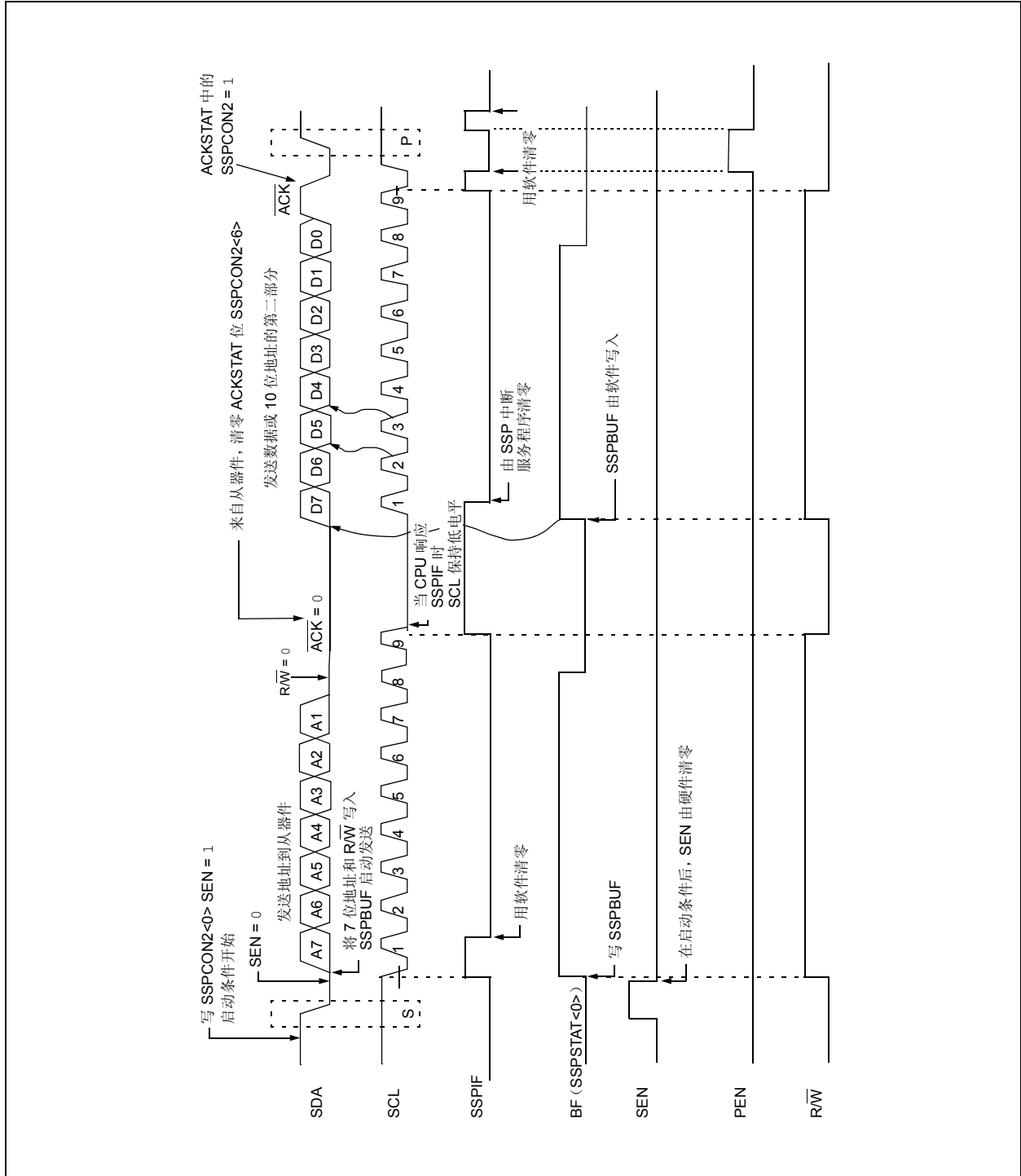
在发送模式下，当从器件已发送应答 ( $\overline{\text{ACK}} = 0$ ) 时清零 SSPCON2 寄存器的 ACKSTAT 位，当从器件未应答 ( $\overline{\text{ACK}} = 1$ ) 时置 1。从器件在已确认其地址 (包括广播呼叫) 或正确接收到数据时发送应答。

### 24.6.6.4 典型发送序列:

1. 用户通过将 SSPCON2 寄存器的 SEN 位置 1 来产生启动条件。
2. 启动条件完成时，SSPIF 由硬件置 1。
3. SSPIF 由软件清零。
4. 在发生任何其他操作之前，MSSP 模块将等待所需的启动时间。
5. 用户将要发送的从地址装入 SSPBUF。
6. 地址将移出 SDA 引脚，直到所有 8 位发送完毕。写入 SSPBUF 时便开始发送。
7. MSSP 模块移入来自从器件的  $\overline{\text{ACK}}$  位，并将其值写入 SSPCON2 寄存器的 ACKSTAT 位。
8. 在第 9 个时钟周期结束时，MSSP 模块通过将 SSPIF 位置 1 产生中断。
9. 用户将 8 位数据装入 SSPBUF。
10. 数据将被移出 SDA 引脚，直到所有 8 位发送完毕。
11. MSSP 模块移入来自从器件的  $\overline{\text{ACK}}$  位，并将其值写入 SSPCON2 寄存器的 ACKSTAT 位。
12. 对于所有发送的数据字节，重复第 8-11 步。
13. 用户通过将 SSPCON2 寄存器的 PEN 或 RSEN 位置 1 来产生停止或重复启动条件。停止 / 重复启动条件完成时产生中断。



图 24-28: I<sup>2</sup>C 主模式波形 (发送, 7 位或 10 位地址)



## 24.6.7 I<sup>2</sup>C 主模式接收

通过对 SSPCON2 寄存器的接收使能位 RCEN 位编程可以使能主模式接收。

**注：** MSSP 模块必须在 RCEN 位置 1 之前处于空闲状态，否则将忽略 RCEN 位。

波特率发生器开始计数，并且在每次计满返回时，SCL 引脚的状态将发生改变（从高到低 / 从低到高），并将数据移入 SSPSR。在第 8 个时钟的下降沿之后，接收使能标志自动清零，SSPSR 的内容被装入 SSPBUF、BF 标志位置 1、SSPIF 标志位置 1，且波特率发生器暂停计数，保持 SCL 为低电平。MSSP 现在处于空闲状态，等待下一条命令。CPU 读取缓冲器时，BF 标志位自动清零。然后，通过将 SSPCON2 寄存器的应答序列使能位 ACKEN 置 1，允许用户在接收结束时发送应答位。

### 24.6.7.1 BF 状态标志

在接收操作中，当地址或数据字节从 SSPSR 装入到 SSPBUF 中时，BF 位置 1。读取 SSPBUF 寄存器时，该位清零。

### 24.6.7.2 SSPOV 状态标志

在接收操作中，当 SSPSR 接收到 8 位数据并且 BF 标志位已经在上一次接收时置 1 时，SSPOV 位置 1。

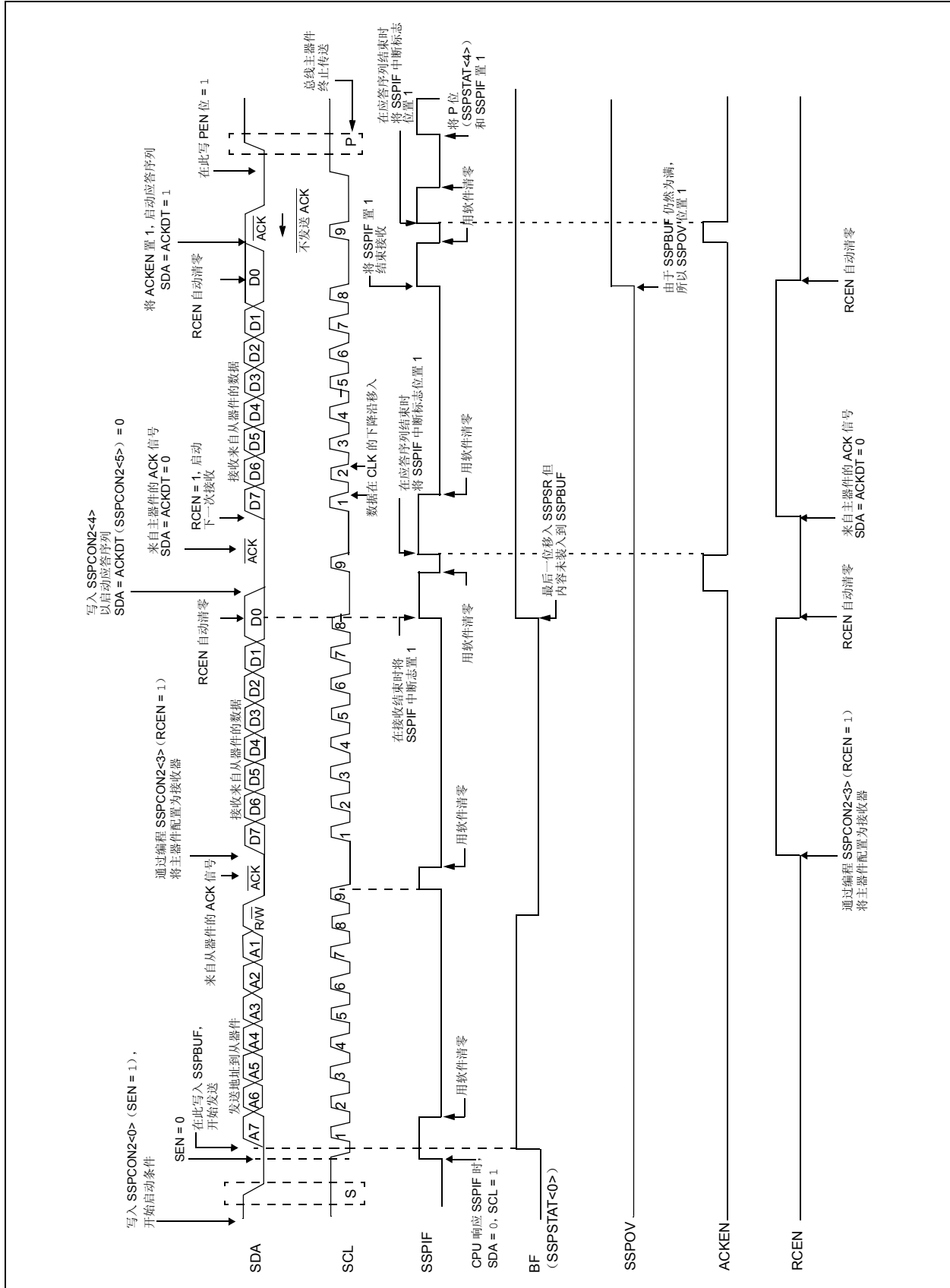
### 24.6.7.3 WCOL 状态标志

如果用户在接收过程中（即 SSPSR 仍在移入数据字节）写 SSPBUF，WCOL 将置 1，且缓冲器的内容不变（未发生写操作）。

### 24.6.7.4 典型接收序列：

1. 用户通过将 SSPCON2 寄存器的 SEN 位置 1 来产生启动条件。
2. 启动条件完成时，SSPIF 由硬件置 1。
3. SSPIF 由软件清零。
4. 用户将要发送的从地址写入 SSPBUF 且 R/W 位置 1。
5. 地址将移出 SDA 引脚，直到所有 8 位发送完毕。写 SSPBUF 便开始发送。
6. MSSP 模块移入来自从器件的  $\overline{\text{ACK}}$  位，并将其值写入 SSPCON2 寄存器的 ACKSTAT 位。
7. 在第 9 个时钟周期结束时，MSSP 模块通过将 SSPIF 位置 1 产生中断。
8. 用户将 SSPCON2 寄存器的 RCEN 位置 1，主器件随着时钟移入来自从器件的字节。
9. 在 SCL 的第 8 个下降沿后，SSPIF 和 BF 置 1。
10. 主器件清零 SSPIF，并从 SSPBUF 中读取接收到的字节，清零 BF。
11. 主器件在 SSPCON2 寄存器的 ACKDT 位中设置将发送给从器件的  $\overline{\text{ACK}}$  值，并通过将 ACKEN 位置 1 发送 ACK。
12. 主器件随着时钟将  $\overline{\text{ACK}}$  移出到从器件，SSPIF 置 1。
13. 用户清零 SSPIF。
14. 对于每个从从器件接收的字节，重复第 8-13 步。
15. 主器件发送非  $\overline{\text{ACK}}$  或停止位以结束通信。

图 24-29: I<sup>2</sup>C 主模式波形 (接收, 7 位地址)



# PIC16(L)F1933

## 24.6.8 应答序列时序

通过将 SSPCON2 寄存器的应答序列使能位 ACKEN 位置 1，使能应答序列。该位置 1 时，SCL 引脚被拉低，且应答数据位的内容输出到 SDA 引脚。如果用户希望产生应答，那么 ACKDT 位应清零。否则，用户应该在启动应答序列之前将 ACKDT 位置 1。然后，波特率发生器计数一个计满返回周期 (TBRG)，SCL 引脚拉为高电平。当采样到 SCL 引脚为高电平 (时钟仲裁) 时，波特率发生器计数一个 TBRG。然后 SCL 引脚被拉为低电平。随后，ACKEN 位自动清零，波特率发生器关闭，MSSP 模块进入空闲模式 (图 24-29)。

### 24.6.8.1 WCOL 状态标志

如果用户在应答序列进行过程中写 SSPBUF，那么 WCOL 将置 1，且缓冲器的内容不变 (未发生写操作)。

## 24.6.9 停止条件时序

将 SSPCON2 寄存器的停止序列使能位 PEN 置 1，可在接收 / 发送结束后在 SDA 引脚上产生停止位。在接收 / 发送结束时，SCL 线在第 9 个时钟的下降沿之后保持低电平。PEN 位置 1 时，主器件将 SDA 线拉为低电平。采样到 SDA 线为低电平时，波特率发生器重新装入数据并递减计数到 0。当波特率发生器超时，SCL 引脚将被拉为高电平，并且在一个 TBRG (波特率发生器计满返回计数) 后，SDA 引脚将被拉为高电平。如果在 SCL 为高电平时采样到 SDA 引脚为高电平，那么 SSPSTAT 寄存器的 P 位置 1。一个 TBRG 后，PEN 位清零，且 SSPIF 位置 1 (图 24-30)。

### 24.6.9.1 WCOL 状态标志

如果用户在停止序列进行过程中写 SSPBUF，那么 WCOL 位将置 1，且缓冲器的内容不变 (未发生写操作)。

图 24-30: 应答序列波形

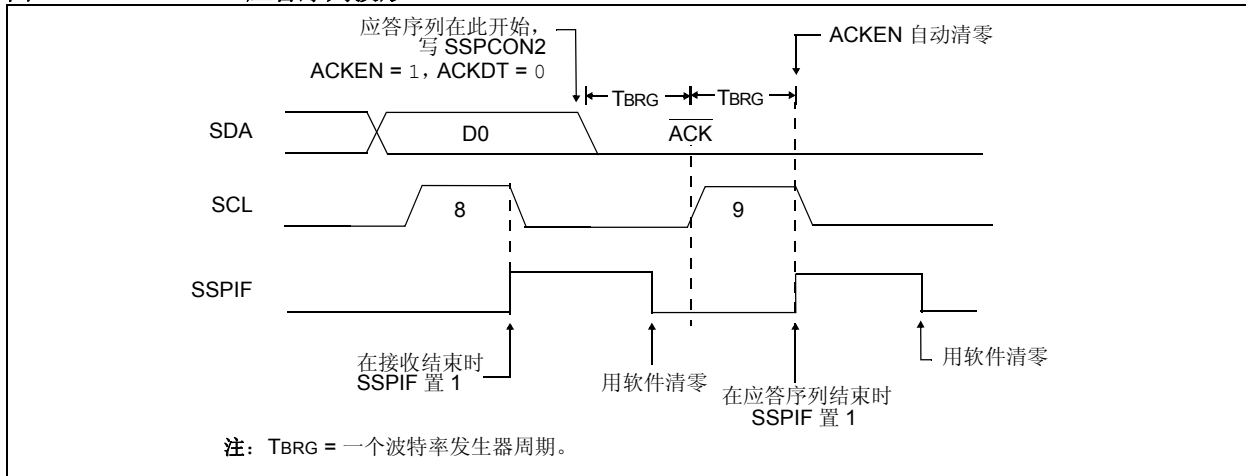
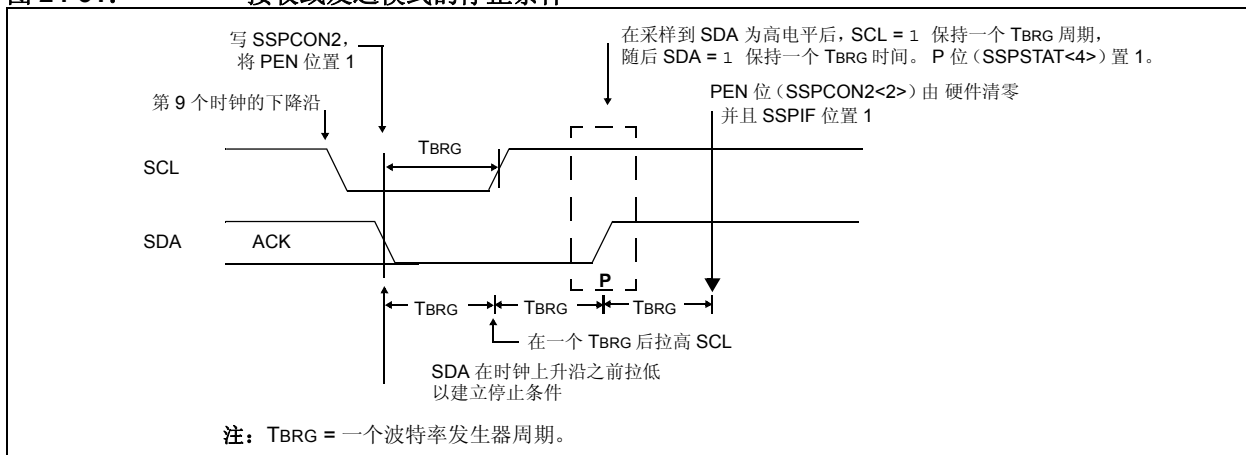


图 24-31: 接收或发送模式的停止条件



## 24.6.10 休眠模式下的工作原理

在休眠模式下，I<sup>2</sup>C 从模块可以接收地址或数据，并在发生地址匹配或完成字节传输时，将处理器从休眠模式唤醒（如果允许了 MSSP 中断）。

## 24.6.11 复位的影响

复位会禁止 MSSP 模块并终止当前的数据传输。

## 24.6.12 多主器件模式

在多主器件模式下，在检测到启动条件和停止条件时产生的中断可用于判断总线何时空闲。停止（P）和启动（S）位在复位或禁止 MSSP 模块时被清零。当 SSPSTAT 寄存器的 P 位置 1 时，可以获得 I<sup>2</sup>C 总线的控制权；否则，P 位和 S 位都清零，总线处于空闲状态。当总线处于忙状态且允许 SSP 中断时，一旦发生停止条件便产生中断。

在多主器件操作中，必须监视 SDA 线来进行仲裁，以确认信号电平是否为期望的输出电平。此检查由硬件执行，并将结果保存在 BCLIF 位中。

可能导致仲裁失败的状态为：

- 地址传输
- 数据传输
- 启动条件
- 重复启动条件
- 应答条件

## 24.6.13 多主器件通信、总线冲突和总线仲裁

通过总线仲裁可支持多主器件工作模式。当主器件将地址 / 数据位输出到 SDA 引脚上时，如果主器件在 SDA 引脚上输出 1（SDA 悬空为高电平）并且另一个主器件输出 0，就会发生仲裁。当 SCL 引脚悬空为高电平时，数据应稳定。如果在 SDA 上期望的数据为 1，而在 SDA 引脚上采样到的数据为 0，那么将发生总线冲突。器件会将总线冲突中断标志位 BCLIF 置 1，并将 I<sup>2</sup>C 端口复位为空闲状态（图 24-31）。

如果在发送数据时发生总线冲突，那么发送将停止，BF 标志位清零，SDA 和 SCL 线被拉高，并将数据写入 SSPBUF。当执行总线冲突中断服务程序时，如果 I<sup>2</sup>C 总线空闲，则用户可以通过发出启动条件来继续通信。

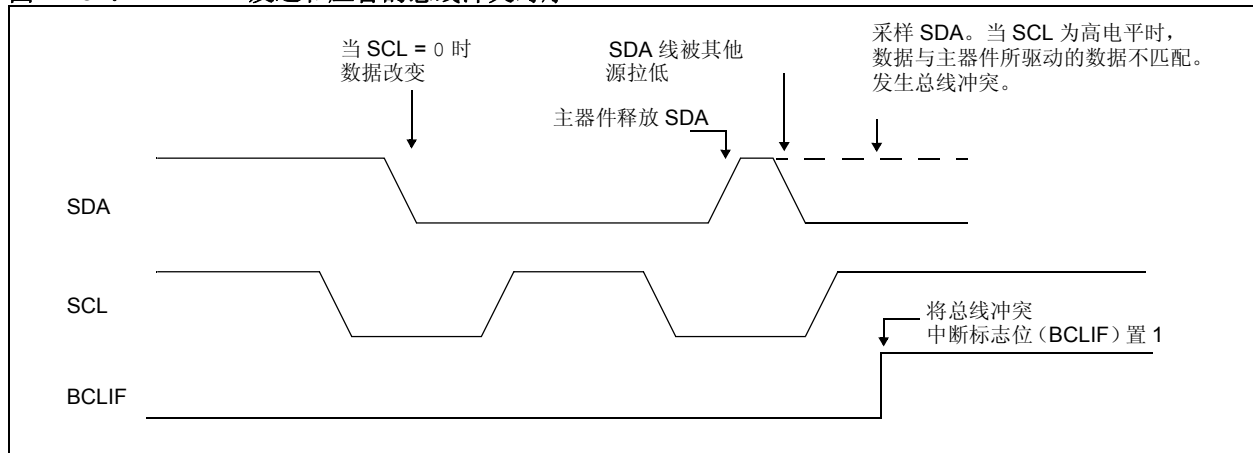
如果在执行启动、重复启动、停止或应答条件时发生总线冲突，则这种条件被中止，SDA 和 SCL 线被拉高，且 SSPCON2 寄存器中的相应控制位清零。当执行总线冲突中断服务程序时，如果 I<sup>2</sup>C 总线空闲，则用户可以通过发出启动条件来继续通信。

主器件将继续监视 SDA 和 SCL 引脚。如果出现停止条件，SSPIF 位将置 1。

写 SSPBUF 将从第一个数据位开始数据发送，而与发生总线冲突时发送器的进度无关。

在多主器件模式下，在检测到启动条件和停止条件时产生的中断可用于判断总线何时空闲。当 SSPSTAT 寄存器的 P 位置 1，可以获得 I<sup>2</sup>C 总线的控制权；否则，S 位和 P 位都清零，总线处于空闲状态。

图 24-32: 发送和应答的总线冲突时序



# PIC16(L)F1933

## 24.6.13.1 启动条件期间的总线冲突

启动条件期间，出现以下情况时发生总线冲突：

- 在启动条件开始时采样到 SDA 或 SCL 为低电平（图 24-32）。
- 在 SDA 引脚被拉为低电平之前采样到 SCL 引脚为低电平（图 24-33）。

启动条件期间，监视 SDA 和 SCL 引脚。

如果 SDA 引脚已为低电平，或者 SCL 引脚已为低电平，那么发生所有以下事件：

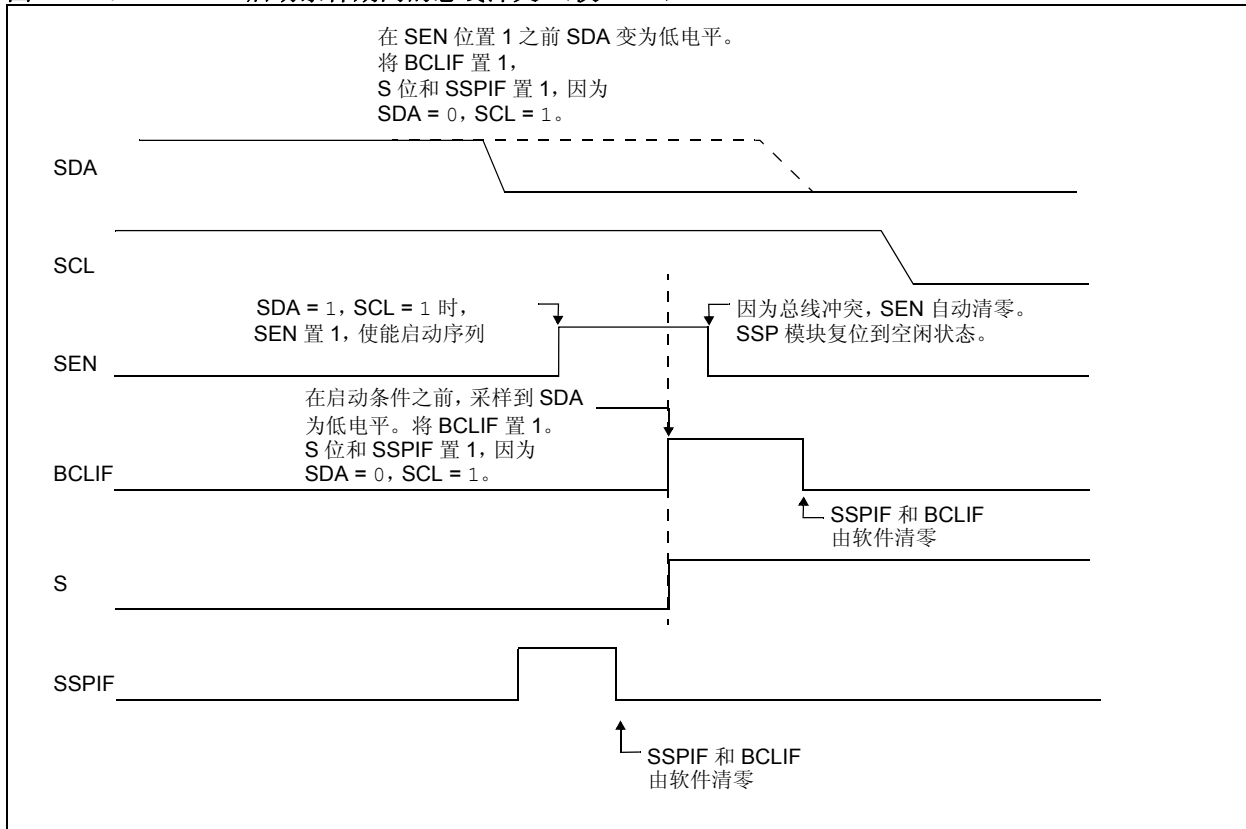
- 中止启动条件，
- BCLIF 标志置 1 并且
- MSSP 模块复位为空闲状态（图 24-32）。

启动条件从 SDA 和 SCL 引脚被拉高开始。当采样到 SDA 引脚为高电平时，波特率发生器装入数据并递减计数。如果在 SDA 为高电平时采样到 SCL 引脚为低电平，则发生总线冲突，这是因为它认为有另一个主器件在启动条件期间尝试驱动数据 1。

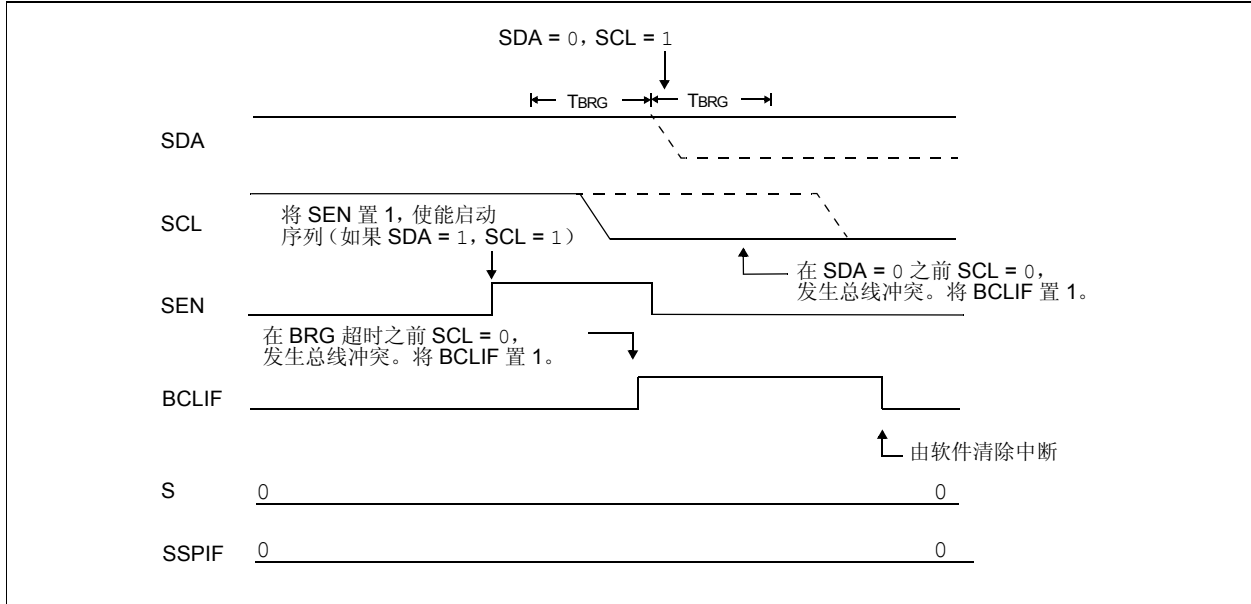
如果在此计数期间采样到 SDA 引脚为低电平，那么 BRG 复位，且 SDA 线被提前拉为高电平（图 24-34）。但是，如果在 SDA 引脚上采样到 1，那么在 BRG 计数结束时，将 SDA 引脚拉为低电平。然后，波特率发生器重新装入数据并递减计数到零；如果在此时间内采样到 SCL 引脚为 0，则不会发生总线冲突。在 BRG 计数结束时，SCL 引脚拉为低电平。

**注：** 在启动条件期间不会发生总线冲突，这是因为不会有两个总线主器件正好同时发出启动条件。因此，一个主器件总是先于另一个主器件将 SDA 拉低。此条件不会导致总线冲突，因为两个主器件必须对启动条件后的第一个地址进行仲裁。如果地址相同，则必须继续对数据部分、重复启动或停止条件进行仲裁。

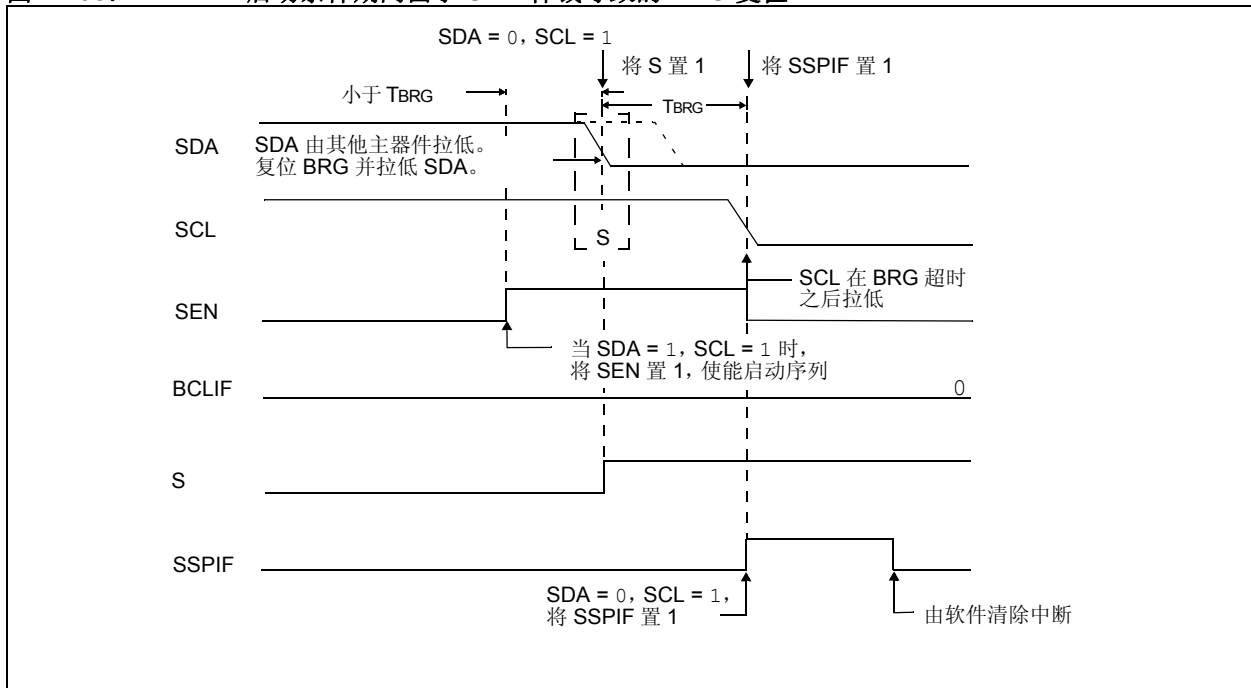
图 24-33: 启动条件期间的总线冲突（仅 SDA）



**图 24-34: 启动条件期间的总线冲突 (SCL = 0)**



**图 24-35: 启动条件期间由于 SDA 仲裁导致的 BRG 复位**



# PIC16(L)F1933

## 24.6.13.2 重复启动条件期间的总线冲突

在重复启动条件期间，出现以下情况时发生总线冲突：

- 当 SCL 从低电平变为高电平时，采样到 SDA 引脚为低电平。
- 在 SDA 被拉低之前 SCL 变为低电平，这表示另一个主器件正在尝试发送数据 1。

用户释放 SDA 并且允许该引脚悬空为高电平时，BRG 装入 SSPADD 的内容并递减计数到零。然后 SCL 引脚被拉高，并且在采样到 SCL 引脚为高电平时，采样 SDA 引脚。

如果 SDA 为低电平，则发生总线冲突（即另一个主器件正在尝试发送数据 0，图 24-35）。如果采样到 SDA 为高电平，则 BRG 重新装入数据并开始计数。如果在 BRG 超时之前，SDA 从高电平变为低电平，那么不会发生总线冲突，因为不会有两个主器件正好同时拉低 SDA。

如果在 BRG 超时之前，SCL 从高电平变为低电平，而 SDA 尚未拉低，则发生总线冲突。这种情况下，表示另一个主器件在重复启动条件期间尝试发送数据 1，请参见图 24-36。

如果在 BRG 超时结束时，SCL 和 SDA 仍为高电平，那么 SDA 引脚将驱动为低电平，BRG 重新装入数据并开始计数。在计数结束时，不管 SCL 引脚的状态如何，SCL 引脚被驱动为低电平，重复启动条件完成。

图 24-36: 重复启动条件期间的总线冲突（情形 1）

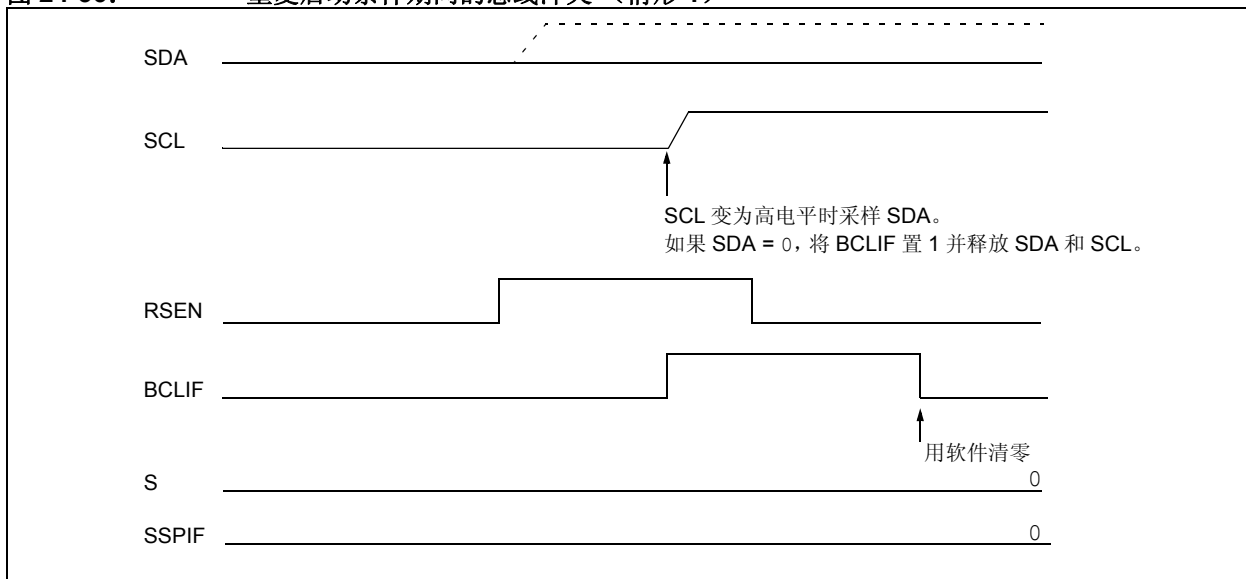
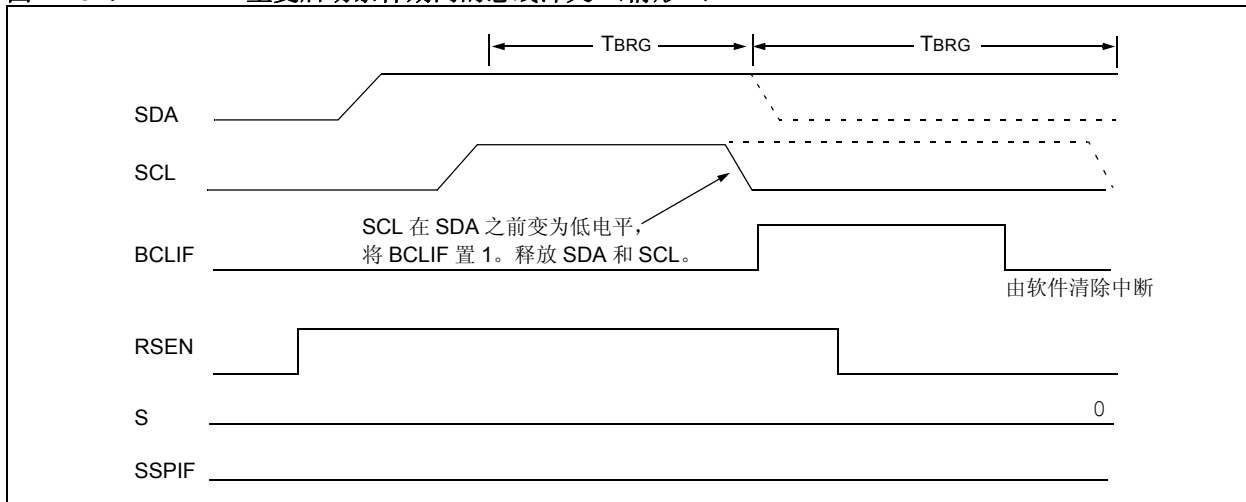


图 24-37: 重复启动条件期间的总线冲突（情形 2）





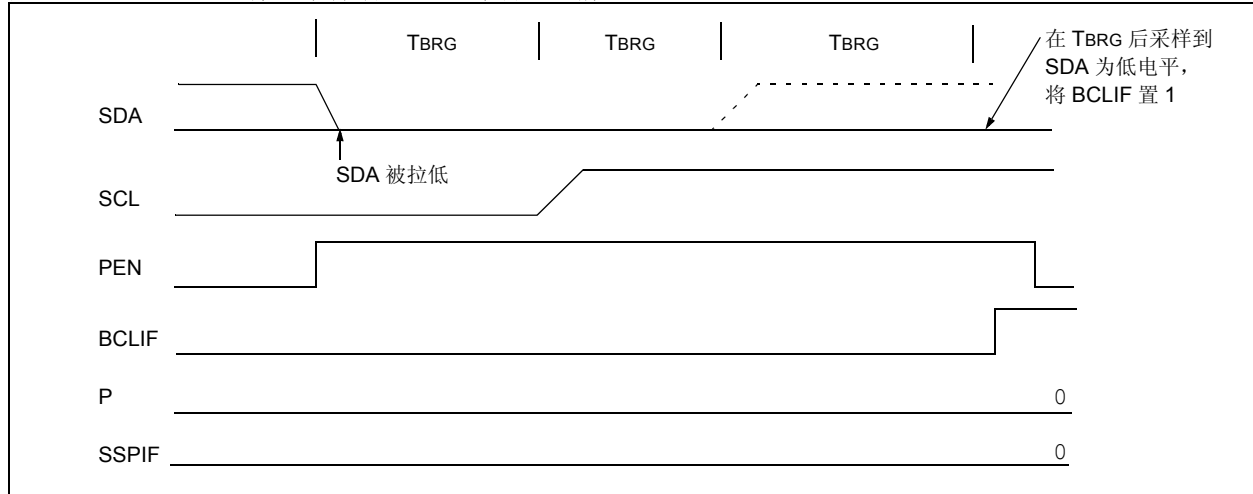
### 24.6.13.3 停止条件期间的总线冲突

在停止条件期间出现以下情况时发生总线冲突：

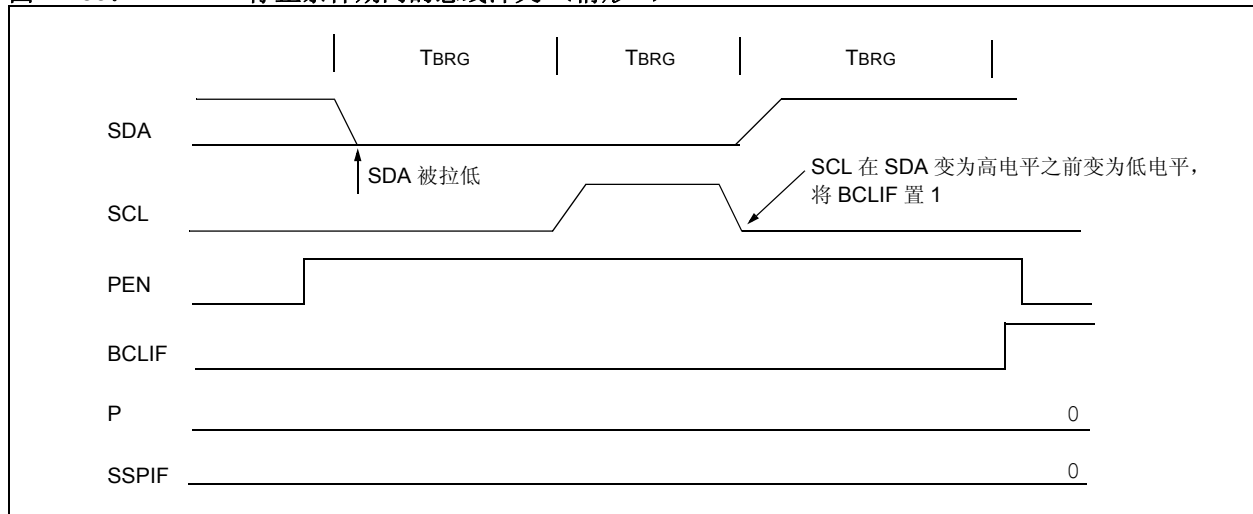
- 在 SDA 引脚已拉高并允许悬空为高电平之后，在 BRG 超时后采样到 SDA 引脚为低电平。
- 在 SCL 引脚被拉高后，在 SDA 变为高电平之前采样到 SCL 引脚为低电平。

停止条件从 SDA 被拉为低电平开始。当采样到 SDA 为低电平时，允许 SCL 引脚悬空。当引脚上采样到高电平（时钟仲裁）时，波特率发生器装入 SSPADD 的内容并递减计数到 0。在 BRG 超时后，采样 SDA。如果采样到 SDA 为低电平，则会发生总线冲突。这是因为另一个主器件正在尝试驱动数据 0（图 24-37）。如果在 SDA 允许悬空为高电平之前采样到 SCL 引脚为低电平，则发生总线冲突。这是有另一个主器件正在尝试驱动数据 0 的另一种情况（图 24-38）。

**图 24-38:** 停止条件期间的总线冲突（情形 1）



**图 24-39:** 停止条件期间的总线冲突（情形 2）



# PIC16(L)F1933

表 24-3: 与 I<sup>2</sup>C™ 操作相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	复位值所在的页面
INTCON	GIE	PEIE	TMR0IE	INTE	IOCFIE	TMR0IF	INTF	IOCFIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
SSPADD	ADD<7:0>								273
SSPBUF	MSSP 接收缓冲器 / 发送寄存器								225*
SSPCON1	WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>				269
SSPCON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	271
SSPCON3	ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN	272
SSPMSK	MSK<7:0>								273
SSPSTAT	SMP	CKE	D/A	P	S	R/W	UA	BF	268

图注: — = 未实现单元, 读为 0。I<sup>2</sup>C™ 模式下的 MSSP 模块不使用阴影单元。

\* 该页提供寄存器信息。

## 24.7 波特率发生器

MSSP 模块具有波特率发生器，用于在 I<sup>2</sup>C 和 SPI 主模式下生成时钟。波特率发生器 (BRG) 的重载值保存在 SSPADD 寄存器 (寄存器 24-6) 中。写 SSPBUF 时，波特率发生器将自动开始递减计数。

完成给定操作时，内部时钟将自动停止计数，时钟引脚将保持最新的状态。

图 24-39 中的内部“重载”信号可使 SSPADD 的值装入到 BRG 计数器中。此操作在模块时钟信号的每个振

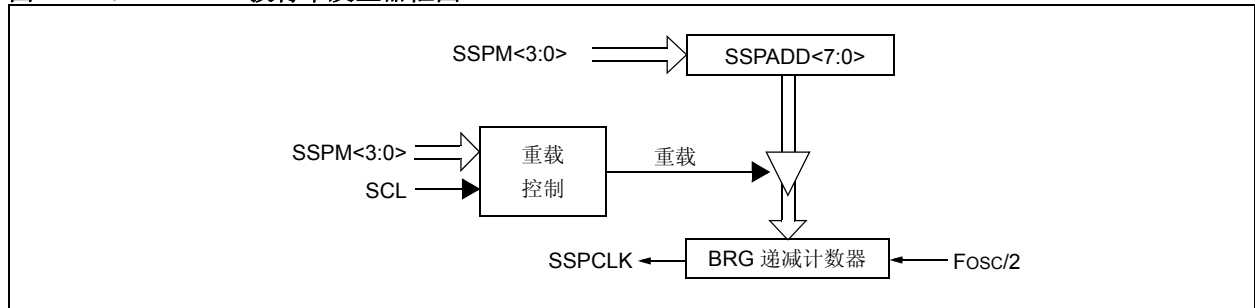
荡周期内发生两次。指示“重载”信号何时有效的逻辑取决于 MSSP 正在使用的工作模式。

表 24-4 演示了不同指令周期下的时钟速率以及装入 SSPADD 的 BRG 值。

公式 24-1:

$$F_{CLOCK} = \frac{F_{OSC}}{(SSPxADD + 1)(4)}$$

图 24-40: 波特率发生器框图



**注:** 当用作 I<sup>2</sup>C 模式下的波特率发生器时，SSPADD 的值 0x00、0x01 和 0x02 无效。这是固有的限制。

表 24-4: 不同 BRG 值下的 MSSP 时钟速率

Fosc	Fcy	BRG 值	F <sub>CLOCK</sub> (2 次 BRG 计满返回)
32 MHz	8 MHz	13h	400 kHz <sup>(1)</sup>
32 MHz	8 MHz	19h	308 kHz
32 MHz	8 MHz	4Fh	100 kHz
16 MHz	4 MHz	09h	400 kHz <sup>(1)</sup>
16 MHz	4 MHz	0Ch	308 kHz
16 MHz	4 MHz	27h	100 kHz
4 MHz	1 MHz	09h	100 kHz

**注 1:** I<sup>2</sup>C 接口并非在各个方面都符合 400 kHz I<sup>2</sup>C 规范 (该规范适用于速率大于 100 kHz 的情况)，在需要较高速率的应用场合应慎重使用。

# PIC16(L)F1933

## 寄存器 24-1: SSPSTAT: SSP 状态寄存器

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0	R-0/0
SMP	CKE	D/A	P	S	R/W	UA	BF
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7 **SMP:** SPI 数据输入采样位  
SPI 主模式:  
1 = 在数据输出时间结束时采样输入数据  
0 = 在数据输出时间的中间采样输入数据  
SPI 从模式:  
当 SPI 用于从模式时, 必须将 SMP 清零  
在 I<sup>2</sup>C 主模式或从模式下:  
1 = 标准速度模式 (100 kHz 和 1 MHz) 下禁止斜率控制  
0 = 高速模式 (400 kHz) 下使能斜率控制
- bit 6 **CKE:** SPI 时钟边沿选择位 (仅 SPI 模式)  
在 SPI 主或从模式下:  
1 = 在从有效时钟状态转变为空闲时钟状态时进行发送  
0 = 在从空闲时钟状态转变为有效时钟状态时进行发送  
仅在 I<sup>2</sup>C™ 模式下:  
1 = 使能输入逻辑, 以便阈值符合 SMBus 规范  
0 = 禁止 SMBus 特定输入
- bit 5 **D/A:** 数据 / 地址位 (仅 I<sup>2</sup>C 模式)  
1 = 表示上次接收或发送的字节是数据  
0 = 表示上次接收或发送的字节是地址
- bit 4 **P:** 停止位  
(仅 I<sup>2</sup>C 模式。当禁止 MSSP 模块且 SSPEN 清零时该位清零。)  
1 = 表示上次检测到了停止位 (此位在复位时为 0)  
0 = 表示上次没有检测到停止位
- bit 3 **S:** 启动位  
(仅 I<sup>2</sup>C 模式。当禁止 MSSP 模块且 SSPEN 清零时该位清零。)  
1 = 表示上次检测到了启动位 (此位在复位时为 0)  
0 = 表示上次没有检测到启动位
- bit 2 **R/W:** 读 / 写信息位 (仅 I<sup>2</sup>C 模式)  
该位用来保存在上次地址匹配后的 R/W 位信息。此位仅在地址匹配与遇到下一个启动位、停止位或非 ACK 位之间有效。  
在 I<sup>2</sup>C 从模式下:  
1 = 读  
0 = 写  
在 I<sup>2</sup>C 主模式下:  
1 = 发送正在进行  
0 = 发送未进行  
该位与 SEN、RSEN、PEN、RCEN 或 ACKEN 的或运算结果指示 MSSP 是否处于空闲状态。
- bit 1 **UA:** 更新地址位 (仅 10 位 I<sup>2</sup>C 模式)  
1 = 表示用户需要更新 SSPADD 寄存器中的地址  
0 = 表示不需要更新地址
- bit 0 **BF:** 缓冲器满状态位  
接收 (SPI 和 I<sup>2</sup>C 模式):  
1 = 接收完成, SSPBUF 满  
0 = 接收未完成, SSPBUF 空  
发送 (仅 I<sup>2</sup>C 模式):  
1 = 数据发送正在进行 (不包括 ACK 和停止位), SSPBUF 满  
0 = 数据发送完成 (不包括 ACK 和停止位), SSPBUF 空

**寄存器 24-2: SSPCON1: SSP 控制寄存器 1**

R/C/HS-0/0	R/C/HS-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
WCOL	SSPOV	SSPEN	CKP	SSPM<3:0>			
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HS = 由硬件置 1                      C = 用户清零

bit 7            **WCOL:** 写冲突检测位

主模式:

1 = 启动发送所需的 I<sup>2</sup>C 条件无效时尝试写入 SSPBUF 寄存器

0 = 无冲突

从模式:

1 = 正在发送前一个字时, 又有数据写入 SSPBUF 寄存器 (必须用软件清零)

0 = 无冲突

bit 6            **SSPOV:** 接收溢出指示位<sup>(1)</sup>

在 SPI 模式下:

1 = SSPBUF 寄存器仍持有前一数据时, 又接收到一个新的字节。如果溢出, SSPSR 中的数据会丢失。溢出只在从模式下发生。在从模式下, 即使只是发送数据, 用户也必须读 SSPBUF, 以避免溢出标志位置 1。在主模式下, 溢出位不会置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSPBUF 寄存器启动 (必须用软件清零)。

0 = 无溢出

在 I<sup>2</sup>C 模式下:

1 = SSPBUF 寄存器仍持有前一字节时, 又接收到一个字节。SSPOV 在发送模式下为无关位 (必须用软件清零)。

0 = 无溢出

bit 5            **SSPEN:** 同步串行端口使能位

在两种模式下, 当使能时, 这些引脚必须正确配置为输入或输出引脚

在 SPI 模式下:

1 = 使能串行端口并将 SCK、SDO、SDI 和 SS 配置为串行端口引脚<sup>(2)</sup>

0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚

在 I<sup>2</sup>C 模式下:

1 = 使能串行端口并将 SDA 和 SCL 配置为串行端口引脚<sup>(3)</sup>

0 = 禁止串行端口并将这些引脚配置为 I/O 端口引脚

bit 4            **CKP:** 时钟极性选择位

在 SPI 模式下:

1 = 时钟为高电平时处于空闲状态

0 = 时钟为低电平时处于空闲状态

在 I<sup>2</sup>C 从模式下:

**SCK 释放控制**

1 = 使能时钟

0 = 保持时钟为低电平 (时钟延长)。(用于确保数据建立时间。)

在 I<sup>2</sup>C 主模式下:

在此模式下未使用

# PIC16(L)F1933

---

## 寄存器 24-2: SSPCON1: SSP 控制寄存器 1 (续)

bit 3-0	<b>SSPM&lt;3:0&gt;</b> : 同步串行端口模式选择位
	0000 = SPI 主模式, 时钟 = Fosc/4
	0001 = SPI 主模式, 时钟 = Fosc/16
	0010 = SPI 主模式, 时钟 = Fosc/64
	0011 = SPI 主模式, 时钟 = TMR2 输出 /2
	0100 = SPI 从模式, 时钟 = SCK 引脚, 使能 $\overline{SS}$ 引脚控制
	0101 = SPI 从模式, 时钟 = SCK 引脚, 禁止 $\overline{SS}$ 引脚控制, $\overline{SS}$ 可用作 I/O 引脚
	0110 = I <sup>2</sup> C 从模式, 7 位地址
	0111 = I <sup>2</sup> C 从模式, 10 位地址
	1000 = I <sup>2</sup> C 主模式, 时钟 = Fosc / (4 * (SSPADD+1)) <sup>(4)</sup>
	1001 = 保留
	1010 = SPI 主模式, 时钟 = Fosc/(4 * (SSPADD+1)) <sup>(5)</sup>
	1011 = I <sup>2</sup> C 固件控制主模式 (从器件空闲模式)
	1100 = 保留
	1101 = 保留
	1110 = I <sup>2</sup> C 从模式, 7 位地址, 并允许启动位和停止位中断
	1111 = I <sup>2</sup> C 从模式, 10 位地址, 并允许启动位和停止位中断

- 注
- 1: 在主模式下, 溢出位不会置 1, 因为每次接收 (和发送) 新数据都是通过写入 SSPBUF 寄存器启动。
  - 2: 当使能时, 这些引脚必须正确配置为输入或输出引脚。
  - 3: 当使能时, SDA 和 SCL 引脚必须配置为输入引脚。
  - 4: I<sup>2</sup>C 模式不支持 SSPADD 值为 0、1 或 2 的情况。
  - 5: 不支持 SSPADD 值为 0。而使用 SSPM = 0000。

## 寄存器 24-3: SSPCON2: SSP 控制寄存器 2

R/W-0/0	R-0/0	R/W-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/S/HS-0/0	R/W/HS-0/0
GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	HC = 由硬件清零      S = 用户置 1

- bit 7      **GCEN:** 广播呼叫使能位 (仅在 I<sup>2</sup>C 从模式下)  
 1 = 允许在 SSPSR 中接收到广播呼叫地址 (0x00 或 00h) 时中断  
 0 = 禁止广播呼叫地址
- bit 6      **ACKSTAT:** 应答状态位 (仅在 I<sup>2</sup>C 模式下)  
 1 = 未收到应答  
 0 = 已收到应答
- bit 5      **ACKDT:** 应答数据位 (仅在 I<sup>2</sup>C 模式下)  
在接收模式下:  
 当用户在接收结束后启动应答序列时发送的值  
 1 = 无应答  
 0 = 应答
- bit 4      **ACKEN:** 应答序列使能位 (仅在 I<sup>2</sup>C 主模式下)  
在主接收模式下:  
 1 = 在 SDA 和 SCL 引脚上发起应答序列并发送 ACKDT 数据位。由硬件自动清零。  
 0 = 应答序列空闲
- bit 3      **RCEN:** 接收使能位 (仅在 I<sup>2</sup>C 主模式下)  
 1 = 使能 I<sup>2</sup>C 接收模式  
 0 = 接收空闲
- bit 2      **PEN:** 停止条件使能位 (仅在 I<sup>2</sup>C 主模式下)  
SCKMSSP 释放控制:  
 1 = 在 SDA 和 SCL 引脚上发起停止条件。由硬件自动清零。  
 0 = 停止条件空闲
- bit 1      **RSEN:** 重复启动条件使能位 (仅在 I<sup>2</sup>C 主模式下)  
 1 = 在 SDA 和 SCL 引脚上发起重复启动条件。由硬件自动清零。  
 0 = 重复启动条件空闲
- bit 0      **SEN:** 启动条件使能位 (仅在 I<sup>2</sup>C 主模式下)  
在主模式下:  
 1 = 在 SDA 和 SCL 引脚上发起启动条件。由硬件自动清零。  
 0 = 启动条件空闲  
在从模式下:  
 1 = 在从发送和从接收模式下都使能时钟延长 (使能了延长功能)  
 0 = 禁止时钟延长

**注 1:** 对于 ACKEN、RCEN、PEN、RSEN 和 SEN 位: 如果 I<sup>2</sup>C 模块不处于空闲模式, 这些位可能不会置 1 (不支持并行操作) 并且也可能不会写 SSPBUF (或禁止写入 SSPBUF)。

# PIC16(L)F1933

寄存器 24-4: SSPCON3: SSP 控制寄存器 3

R-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ACKTIM	PCIE	SCIE	BOEN	SDAHT	SBCDE	AHEN	DHEN
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7 **ACKTIM:** 应答时间状态位 (仅 I<sup>2</sup>C 模式) (3)  
1 = 表示 I<sup>2</sup>C 总线上有应答序列, 在 SCL 时钟的第 8 个下降沿置 1  
0 = 无应答序列, 在 SCL 时钟的第 9<sup>↑</sup>上升沿清零
- bit 6 **PCIE:** 停止条件中断允许位 (仅 I<sup>2</sup>C 模式)  
1 = 允许在检测到停止条件时中断  
0 = 禁止在检测到停止条件时中断 (2)
- bit 5 **SCIE:** 启动条件中断允许位 (仅 I<sup>2</sup>C 模式)  
1 = 允许在检测到启动或重复启动条件时中断  
0 = 禁止在检测到启动条件时中断 (2)
- bit 4 **BOEN:** 缓冲器改写使能位  
在 SPI 从模式下: (1)  
1 = 每次移入一个新的数据字节, 便会更新 SSPBUF, 与 BF 位无关  
0 = 如果在 SSPSTAT 寄存器的 BF 位已置 1 的条件下接收到新字节, 则 SSPCON1 寄存器的 SSPOV 位置 1 且缓冲器不更新  
在 I<sup>2</sup>C 主模式和 SPI 主模式下:  
忽略此位。  
在 I<sup>2</sup>C 从模式下:  
1 = 仅当 BF 位 = 0 时, 在接收到地址/数据字节时, 更新 SSPBUF 并生成 ACK 信号, 忽略 SSPOV 位的状态  
0 = 仅当 SSPOV 位清零时更新 SSPBUF
- bit 3 **SDAHT:** SDA 保持时间选择位 (仅 I<sup>2</sup>C 模式)  
1 = 在 SCL 的下降沿之后在 SDA 上至少保持 300 ns 的时间  
0 = 在 SCL 的下降沿之后在 SDA 上至少保持 100 ns 的时间
- bit 2 **SBCDE:** 从模式总线冲突检测允许位 (仅 I<sup>2</sup>C 从模式)  
如果在 SCL 的上升沿, 模块输出为高电平状态时采样到 SDA 为低电平, 则 PIR2 寄存器的 BCLIF 位置 1 且总线进入空闲状态  
1 = 允许从总线冲突中断  
0 = 禁止从总线冲突中断
- bit 1 **AHEN:** 地址保持使能位 (仅 I<sup>2</sup>C 从模式)  
1 = 在接收匹配地址字节的 SCL 的第 8 个下降沿之后, SSPCON1 寄存器的 CKP 位清零且 SCL 保持低电平。  
0 = 禁止地址保持
- bit 0 **DHEN:** 数据保持使能位 (仅 I<sup>2</sup>C 从模式)  
1 = 在接收数据字节的 SCL 的第 8 个下降沿之后, 从硬件清零 SSPCON1 寄存器的 CKP 位且 SCL 保持低电平。  
0 = 禁止数据保持

- 注 1: 用于菊花链结构的 SPI 操作; 允许用户忽略除最后一个接收到的字节之外的所有字节。当接收到新字节且 BF = 1 时, SSPOV 位仍然置 1, 但是硬件继续将最新字节写入 SSPBUF。
- 2: 在从模式下, 由于启动条件检测和停止条件检测明确显示为使能状态, 所以该位不起作用。
- 3: ACKTIM 状态位仅在 AHEN 位或 DHEN 位置 1 时有效。



## 寄存器 24-5: SSPMSK: SSP 掩码寄存器

R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1
MSK<7:0>							
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7-1      **MSK<7:1>**: 掩码位  
 1 = 接收到的地址位 bit n 与 SSPADD<n> 相比较以检测 I<sup>2</sup>C 地址匹配  
 0 = 接收到的地址位 bit n 不用于检测 I<sup>2</sup>C 地址匹配
- bit 0      **MSK<0>**: 在 I<sup>2</sup>C 从模式, 10 位地址条件下的掩码位  
 I<sup>2</sup>C 从模式, 10 位地址 (SSPM<3:0> = 0111 或 1111):  
 1 = 接收到的地址位 bit 0 与 SSPADD<0> 相比较以检测 I<sup>2</sup>C 地址匹配  
 0 = 接收到的地址位 bit 0 不用于检测 I<sup>2</sup>C 地址匹配  
 在 I<sup>2</sup>C 从模式, 7 位地址条件下, 忽略该位

## 寄存器 24-6: SSPADD: MSSP 地址和波特率寄存器 (I<sup>2</sup>C 模式)

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
ADD<7:0>							
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

### 主模式:

- bit 7-0      **ADD<7:0>**: 波特率时钟分频比位  
 SCL 引脚的时钟周期 = ((ADD<7:0> + 1) \* 4) / Fosc

### 10 位从模式 —— 高地址字节:

- bit 7-3      **未使用**: 高地址字节未使用这 5 位。寄存器中的这些位为无关位。主器件发送的位模式由 I<sup>2</sup>C 规范指定且必须等于 11110。但是, 这些位由硬件进行比较且不受该寄存器中的值的影响。
- bit 2-1      **ADD<2:1>**: 10 位地址的高 2 位
- bit 0      **未使用**: 在此模式下未使用。该位为无关位。

### 10 位从模式 —— 低地址字节:

- bit 7-0      **ADD<7:0>**: 10 位地址的低 8 位

### 7 位从模式:

- bit 7-1      **ADD<7:1>**: 7 位地址
- bit 0      **未使用**: 在此模式下未使用。该位为无关位。

# PIC16(L)F1933

---

注:

## 25.0 增强型通用同步 / 异步收发器 (EUSART)

增强型通用同步 / 异步收发器 (EUSART) 模块是串行 I/O 通信外设。它包含用来完成与器件程序执行无关的输入或输出串行数据传输所需的所有时钟发生器、移位寄存器和数据缓冲区。EUSART 也称为串行通信接口 (Serial Communications Interface, SCI)，可配置为全双工异步系统或半双工同步系统。全双工模式可用于与 CRT 终端和个人计算机等外部系统通信。半双工同步模式用于与 A/D 或 D/A 集成电路、串行 EEPROM 或其他单片机等外部器件通信。这些器件通常不具备用以生成波特率的内部时钟，并需要由主同步器件提供外部时钟信号。

EUSART 模块包含以下功能：

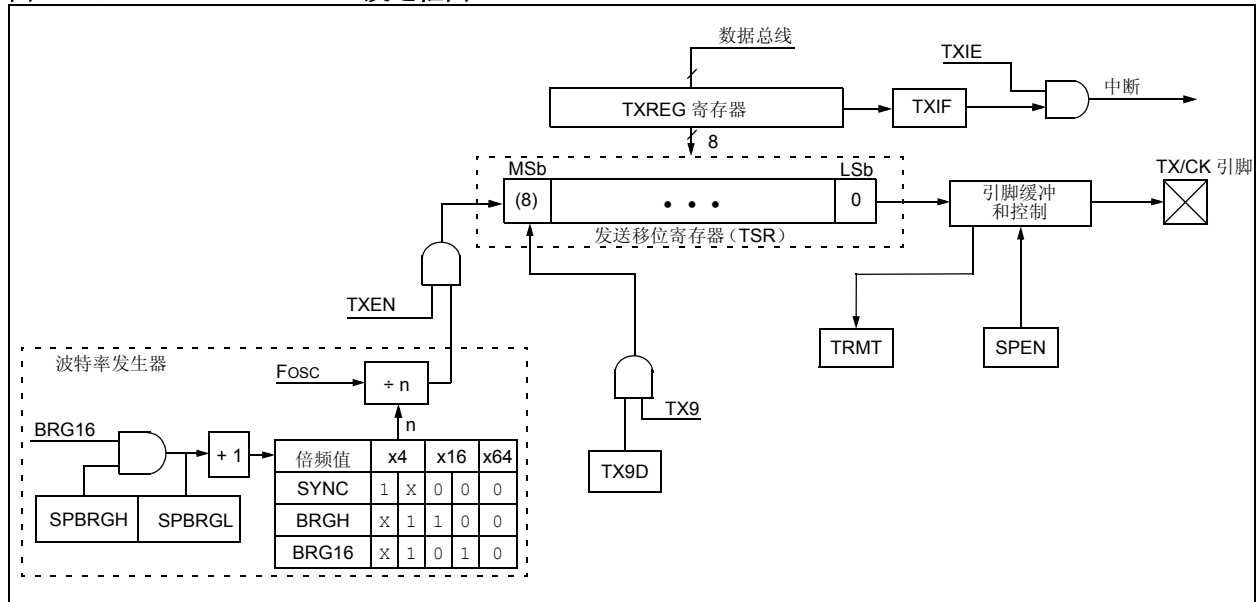
- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 字符长度可编程为 8 位或 9 位
- 9 位模式下的地址检测
- 输入缓冲区溢出错误检测
- 接收字符帧错误检测
- 半双工同步主模式
- 半双工同步从模式
- 同步模式下的可编程时钟极性
- 休眠模式下工作

EUSART 模块还实现了以下功能，使其成为局域网 (Local Interconnect Network, LIN) 总线系统的理想选择：

- 波特率的自动检测和校准
- 接收到间隔字符时唤醒
- 13 位间隔字符发送

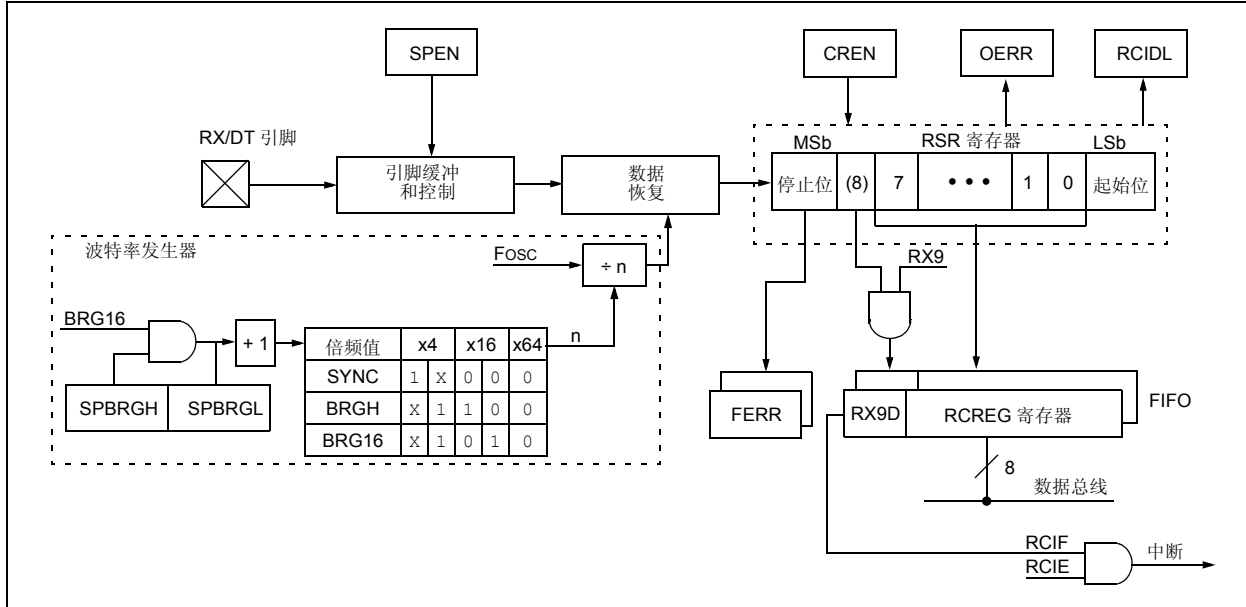
图 25-1 和图 25-2 给出了 EUSART 收发器的框图。

图 25-1: EUSART 发送框图



# PIC16(L)F1933

图 25-2: EUSART 接收框图



EUSART 模块的操作通过以下三个寄存器控制：

- 发送状态和控制寄存器 (TXSTA)
- 接收状态和控制寄存器 (RCSTA)
- 波特率控制寄存器 (BAUDCON)

寄存器 25-1、寄存器 25-2 和寄存器 25-3 分别给出了这些寄存器的详细说明。

未使能接收器或发送器部分时，相应的 RX 或 TX 引脚可以用作通用输入和输出引脚。

## 25.1 EUSART 异步模式

EUSART 采用标准非归零 (non-return-to-zero, NRZ) 格式发送和接收数据。NRZ 实现为两种电平: V<sub>OH</sub> 标记状态 (mark state) 代表 1 数据位, 而 V<sub>OL</sub> 空格状态 (space state) 代表 0 数据位。NRZ 指的是连续发送相同值的数据位时, 将保持该位的输出电平不变, 而不会在发送完每个位后返回到中间电平。NRZ 发送端口在标记状态空闲。每个字符发送包含 1 个起始位及随后的 8 个或 9 个数据位, 并始终由 1 个或多个停止位终止。起始位始终处于空格状态, 停止位始终处于标记状态。最常用的数据格式为 8 位。每个发送位持续时间为 1/(波特率)。使用片上专用 8 位/16 位波特率发生器从系统振荡器产生标准波特率频率。请参见表 25-5 中的波特率配置示例。

EUSART 首先发送和接收 LSb。EUSART 发送器和接收器在功能上是相互独立的, 但采用相同的数据格式和波特率。硬件不支持奇偶校验, 但可以用软件实现并作为第 9 个数据位存储。

### 25.1.1 EUSART 异步发送器

图 25-1 给出了 EUSART 发送器框图。发送器的核心是串行发送移位寄存器 (Transmit Shift Register, TSR), 该寄存器不能用软件直接访问。TSR 从发送缓冲区 (即 TXREG 寄存器) 获取数据。

#### 25.1.1.1 使能发送器

EUSART 发送器可通过配置以下 3 个控制位使能为异步操作:

- TXEN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 TXSTA 寄存器的 TXEN 位置 1 可使能 EUSART 的发送器电路。清零 TXSTA 寄存器的 SYNC 位可将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 可使能 EUSART 并自动将 TX/CK I/O 引脚配置为输出。如果 EUSART 与模拟外设共享 TX/CK 引脚, 必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

**注 1:** TXEN 使能位置 1 时, TXIF 发送器中断标志位也将置 1。

#### 25.1.1.2 发送数据

通过向 TXREG 寄存器写入一个字符来启动发送。如果是第一个字符, 或前一个字符被完全从 TSR 中送出, TXREG 中的数据就立即被传送到 TSR 寄存器。如果 TSR 仍包含前一个字符的全部或部分, 则新字符数据保存在 TXREG 中, 直到前一个字符的停止位发送完毕。然后, 在停止位发送后经过 1 个 T<sub>cy</sub>, 立即将 TXREG 中待处理的字符传送到 TSR。TXREG 中的数据被传送到 TSR 后, 立即开始起始位、数据位和停止位序列的发送。

#### 25.1.1.3 发送数据极性

发送数据的极性可通过 BAUDCON 寄存器的 SCKP 位控制。该位的默认状态为 0, 它选择高电平发送空闲和数据位。将 SCKP 位设置为 1 将翻转发送数据, 产生低电平空闲和数据位。SCKP 位仅在异步模式下控制发送数据极性。在同步模式下, SCKP 位具有不同的功能。请参见第 25.4.1.2 节“时钟极性”。

#### 25.1.1.4 发送中断标志

只要 EUSART 发送器被使能且 TXREG 中没有等待发送的字符, PIR1 寄存器的 TXIF 中断标志位就会被置 1。也就是说, 只有在 TSR 忙于处理字符且 TXREG 中还有一个排队等待发送的新字符时, TXIF 位才被清零。写入 TXREG 后并不立即清零 TXIF 标志位。TXIF 在执行写操作后的第 2 个指令周期才有效。写入 TXREG 后立即查询 TXIF 位将返回无效结果。TXIF 位是只读的, 不能用软件置 1 或清零。

将 PIE1 寄存器的 TXIE 中断允许位置 1 可允许 TXIF 中断。但是, 只要 TXREG 为空, 不管 TXIE 允许位的状态如何, TXIF 标志位都将被置 1。

要在发送数据时使用中断, 应只在仍有数据要发送时才将 TXIE 位置 1。将发送的最后一个字符写入 TXREG 后清零 TXIE 中断允许位。

# PIC16(L)F1933

## 25.1.1.5 TSR 状态

TXSTA 寄存器的 TRMT 位指示 TSR 寄存器的状态。该位是只读位。TSR 寄存器为空时，TRMT 位置 1，当有一个字符从 TXREG 传送到 TSR 寄存器时，该位清零。TRMT 位保持清零状态，直到所有位移出 TSR 寄存器为止。该位不与任何中断逻辑有关，因此用户必须查询该位以确定 TSR 的状态。

**注：** TSR 寄存器并未映射到数据存储中，因此用户不能访问它。

## 25.1.1.6 发送 9 位字符

EUSART 支持 9 位字符发送。当 TXSTA 寄存器的 TX9 位置 1 时，EUSART 将在发送每个字符时移出 9 位。TXSTA 寄存器的 TX9D 位是第 9 个数据位，也是最高有效位。发送 9 位数据时，必须在将低 8 位写入 TXREG 之前先写入 TX9D 数据位。写入 TXREG 后，所有 9 个数据位将被立即传送到 TSR 移位寄存器。

有多个接收器时，可使用一种特殊的 9 位地址模式。关于地址模式的更多信息，请参见第 25.1.2.7 节“地址检测”。

## 25.1.1.7 异步发送设置：

1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，以获得所需的波特率（见第 25.3 节“EUSART 波特率发生器（BRG）”）。
2. 通过清零 SYNC 位并将 SPEN 位置 1，使能异步串口。
3. 如果需要发送 9 位数据，将 TX9 控制位置 1。接收器置于地址检测模式时，第 9 个数据位置 1 表示低 8 个数据位为地址。
4. 如果需要翻转发送数据，将 SCKP 位置 1。
5. 将 TXEN 控制位置 1 来使能发送。这将导致 TXIF 中断标志位置 1。
6. 如果需要中断，将 PIE1 寄存器的 TXIE 中断允许位置 1。如果 INTCON 寄存器的 GIE 和 PEIE 位也置 1，则立即产生中断。
7. 如果选择了发送 9 位数据，应将第 9 位装入 TX9D 数据位。
8. 将 8 位数据装入 TXREG 寄存器。这将启动发送。

图 25-3: 异步发送

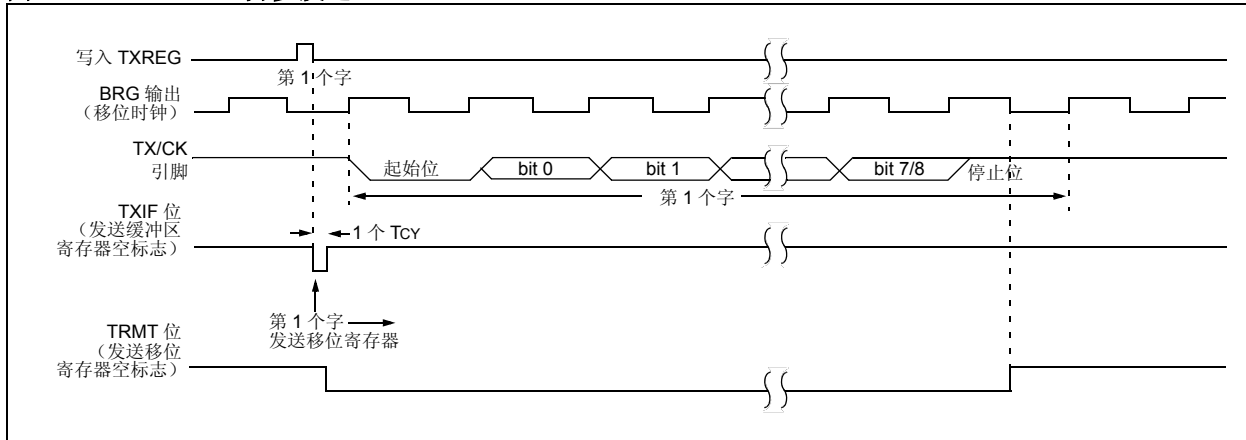
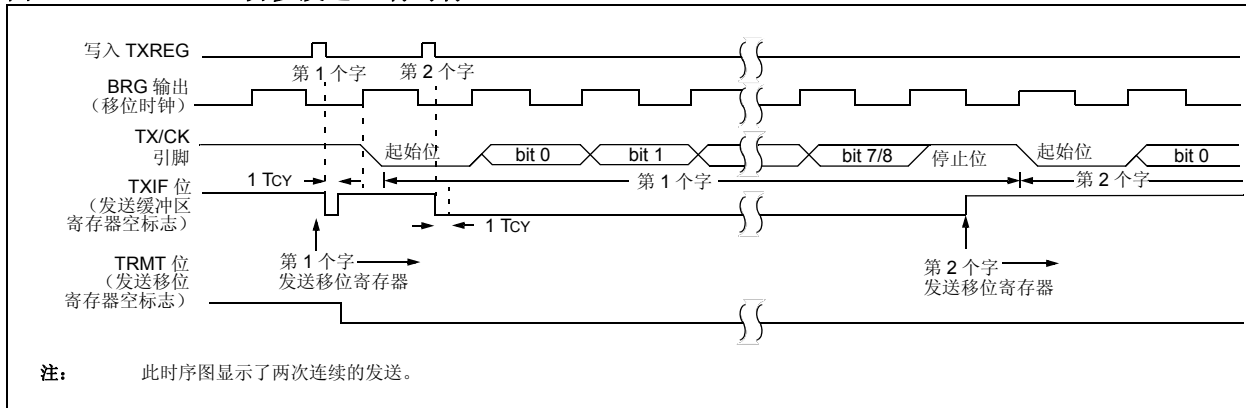


图 25-4: 异步发送（背对背）



**注：** 此时序图显示了两次连续的发送。

**表 25-1: 与异步发送相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	286
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
SPBRGL	BRG<7:0>								287*
SPBRGH	BRG<15:8>								287*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
TXREG	EUSART 发送数据寄存器								277*
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	284

**图注:** — = 未实现单元, 读为 0。异步发送不使用阴影单元。

\* 该页提供寄存器信息。

## 25.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统中。图 25-2 给出了接收器框图。数据通过 RX/DT 引脚接收并驱动数据恢复模块。数据恢复模块实际上是以 16 倍波特率工作的高速移位器，而串行接收移位寄存器（Receive Shift Register, RSR）工作在比特率下。字符的所有 8 位或 9 位移入后被立即传送到双字符的先进先出（First-In-First-Out, FIFO）存储区中。FIFO 缓冲区允许先接收 2 个完整字符和第 3 个字符的起始位后，再由软件将数据提供给 EUSART 接收器。FIFO 和 RSR 寄存器不能直接用软件访问。可通过 RCREG 寄存器访问所接收的数据。

### 25.1.2.1 使能接收器

EUSART 接收器可通过配置以下 3 个控制位使能为异步操作：

- CREN = 1
- SYNC = 0
- SPEN = 1

假定所有其他 EUSART 控制位均处于其默认状态。

将 RCSTA 寄存器的 CREN 位置 1 来使能 EUSART 的接收器电路。清零 TXSTA 寄存器的 SYNC 位可将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 可使能 EUSART。编程器必须将相应的 TRIS 位置 1 以将 RX/DT I/O 引脚配置为输入。

**注 1：** 如果 RX/DT 功能在模拟引脚上，那么必须清零相应的 ANSEL 位以使接收器正常工作。

### 25.1.2.2 接收数据

接收器的数据恢复电路在第 1 位的下降沿启动字符接收。第 1 位也称起始位，始终为零。数据恢复电路计数半个位时间至起始位的中点，并检验该位是否仍为零。如果该位不为零，则数据恢复电路中止字符接收，而不会产生错误，并继续查找起始位的下降沿。如果起始位零校验通过，则数据恢复电路计数 1 个完整的位时间至下一个位的中点。然后，由择多检测电路对该位采样并将结果（0 或 1）移入 RSR。重复此过程直到所有数据位均被采样并移入 RSR 为止。测量最后一个位时间并采样其电平。此位为停止位，始终为 1。如果数据恢复电路在停止位处采样到 0，则将此字符的帧错误标志置 1，否则清零。更多关于帧错误的信息，请参见第 25.1.2.4 节“接收帧错误”。

接收到所有数据位和停止位之后，立即将 RSR 中的字符传送到 EUSART 接收 FIFO，且 PIR1 寄存器的 RCIF 中断标志位置 1。通过读 RCREG 寄存器，将 FIFO 顶部的字符送出 FIFO。

**注：** 如果接收 FIFO 溢出，在溢出条件清除前不会接收更多字符。更多关于溢出错误的信息，请参见第 25.1.2.5 节“接收溢出错误”。

### 25.1.2.3 接收中断

只要 EUSART 接收器使能且接收 FIFO 中存在未读字符，PIR1 寄存器的 RCIF 中断标志位就会置 1。RCIF 中断标志位是只读位，不能用软件置 1 或清零。

将以下位置 1 可允许 RCIF 中断：

- PIE1 寄存器的 RCIE 中断允许位
- INTCON 寄存器的 PEIE 外设中断允许位
- INTCON 寄存器的 GIE 全局中断允许位

当 FIFO 中存在未读字符时，不管中断允许位的状态如何，RCIF 中断标志位都会置 1。



## 25.1.2.4 接收帧错误

接收 FIFO 缓冲区中的每个字符都有相应的帧错误状态位。帧错误表明在预期时间内未接收到停止位。通过 RCSTA 寄存器的 FERR 位可访问帧错误状态。FERR 位表示接收 FIFO 顶部未读字符的状态。因此，在读 RCREG 之前必须先读 FERR 位。

FERR 位是只读位，只用于接收 FIFO 顶部的未读字符。帧错误 (FERR = 1) 不会阻止接收其他字符。此时不必将 FERR 位清零。从 FIFO 缓冲区读下一个字符将前进至 FIFO 的下一个字符和下一个对应的帧错误状态。

将 RCSTA 寄存器的 SPEN 位清零可复位 EUSART，从而将 FERR 位强制清零。将 RCSTA 寄存器的 CREN 位清零不影响 FERR 位。帧错误本身不会产生中断。

**注：** 如果接收FIFO中的所有接收到的字符均有帧错误，反复读 RCREG 不会将 FERR 位清零。

## 25.1.2.5 接收溢出错误

接收 FIFO 缓冲区可保存两个字符。如果在访问 FIFO 之前接收到完整的第三个字符，则会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 缓冲区中已有的字符可被读出，但溢出错误清除之前不会接收其他字符。必须通过清零 RCSTA 寄存器的 CREN 位或通过清零 RCSTA 寄存器的 SPEN 位使 EUSART 复位来清除该错误。

## 25.1.2.6 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将接收到的每个字符的 9 个位移入 RSR。RCSTA 寄存器的 RX9D 位是第 9 位，也是接收 FIFO 顶部未读字符的最高有效数据位。从接收 FIFO 缓冲区读 9 位数据时，在读 RCREG 的低 8 位之前必须先读 RX9D 数据位。

## 25.1.2.7 地址检测

当多个接收器共用同一条传输线时（如在 RS-485 系统中），可使用特殊地址检测模式。将 RCSTA 寄存器的 ADDEN 位置 1 可使能地址检测。

地址检测要求接收 9 位字符。使能地址检测时，只有第 9 数据位置 1 的字符会被传送到接收 FIFO 缓冲区，从而将 RCIF 中断标志位置 1。忽略所有其他字符。

接收到地址字符后，用户软件判断此地址是否与其匹配。地址匹配时，用户软件必须在出现下一个停止位之前通过清零 ADDEN 位禁止地址检测。当用户软件检测到报文的末尾（由所使用的报文协议确定）时，软件通过将 ADDEN 位置 1 把接收器重新置于地址检测模式。

# PIC16(L)F1933

## 25.1.2.8 异步接收设置:

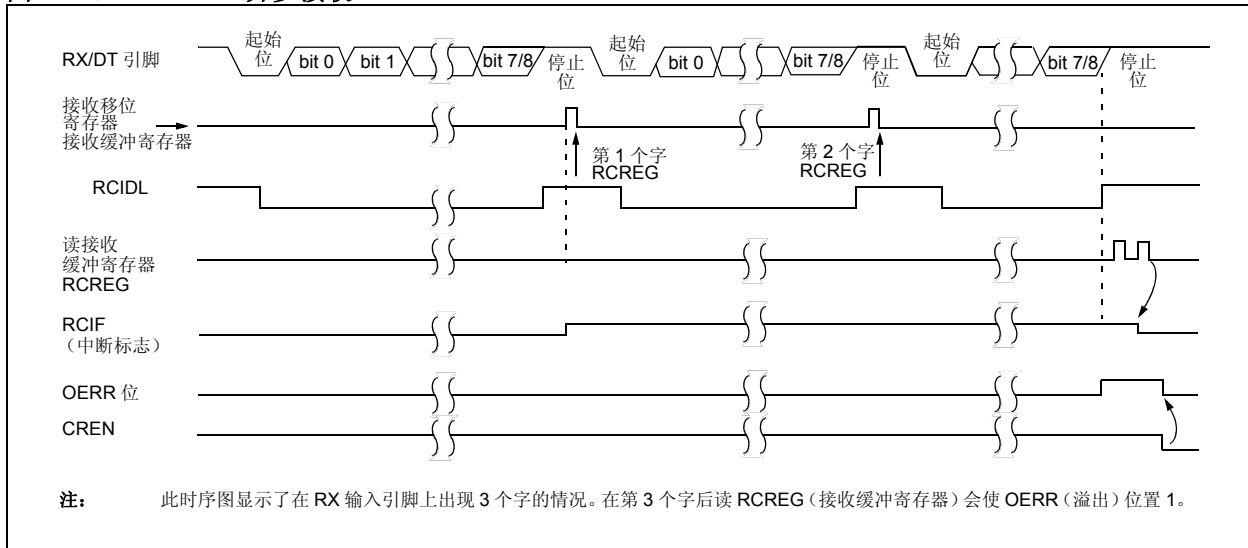
1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位, 以获得所需的波特率 (见第 25.3 节“EUSART 波特率发生器 (BRG)”)。
2. 将 RX 引脚的 ANSEL 位清零 (如果适用)。
3. 通过将 SPEN 位置 1, 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断, 将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要接收 9 位数据, 将 RX9 位置 1。
6. 通过将 CREN 位置 1, 使能接收。
7. 当字符从 RSR 传送到接收缓冲区时, RCIF 中断标志位置 1。如果 RCIE 中断允许位也置 1, 则产生中断。
8. 读 RCSTA 寄存器以获取错误标志和第 9 个数据位 (如果使能了 9 位数据接收)。
9. 通过读 RCREG 寄存器从接收缓冲区获取所接收数据的低 8 位。
10. 如果发生溢出, 则通过清零 CREN 接收器使能位来清零 OERR 标志。

## 25.1.2.9 9 位地址检测模式设置

此模式通常用于 RS-485 系统中。要设置使能了地址检测的异步接收:

1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位, 以获得所需的波特率 (见第 25.3 节“EUSART 波特率发生器 (BRG)”)。
2. 将 RX 引脚的 ANSEL 位清零 (如果适用)。
3. 通过将 SPEN 位置 1, 使能串口。SYNC 位必须清零才能进行异步操作。
4. 如果需要中断, 将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 通过将 RX9 位置 1, 使能 9 位接收。
6. 通过将 ADDEN 位置 1, 使能地址检测。
7. 通过将 CREN 位置 1, 使能接收。
8. 当第 9 位置 1 的字符从 RSR 传送到接收缓冲区时, RCIF 中断标志位置 1。如果 RCIE 中断允许位也置 1, 则产生中断。
9. 读 RCSTA 寄存器以获取错误标志。第 9 个数据位始终置 1。
10. 通过读 RCREG 寄存器从接收缓冲区获取所接收数据的低 8 位。软件判断此地址是否为器件地址。
11. 如果发生溢出, 则通过清零 CREN 接收器使能位来清零 OERR 标志。
12. 如果器件被寻址, 将 ADDEN 位清零以允许所有接收到的数据送入接收缓冲区并产生中断。

图 25-5: 异步接收



**表 25-2: 与异步接收相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	286
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								280*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
SPBRGL	BRG<7:0>								287*
SPBRGH	BRG<15:8>								287*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	284

**图注:** — = 未实现单元, 读为 0。异步接收不使用阴影单元。

\* 该页提供寄存器信息。

# PIC16(L)F1933

## 25.2 异步工作的时钟精度

厂家会校准内部振荡器模块的输出（INTOSC）。但在 VDD 或温度变化时，INTOSC 频率可能会发生漂移，这直接影响了异步工作的波特率。可使用两种方法来调节波特率时钟，但是这两种方法都需要某种类型的参考时钟源。

第一种（首选）方法使用 OSCTUNE 寄存器来调节 INTOSC 输出。调节 OSCTUNE 寄存器中的值可对系统时钟源的分辨率进行微调。更多信息，请参见第 5.2.2 节“内部时钟源”。

另一种方法是调节波特率发生器中的值。这可以使用自动波特率检测功能来自动完成（见第 25.3.1 节“自动波特率检测”）。调节波特率发生器来补偿外设时钟频率的逐步变化可能无法足够细微地调节分辨率。

寄存器 25-1: TXSTA: 发送状态和控制寄存器

R/W-0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-1/1	R/W-0/0	
CSRC	TX9	TXEN <sup>(1)</sup>	SYNC	SENDB	BRGH	TRMT	TX9D	
bit 7								bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位，读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **CSRC:** 时钟源选择位  
异步模式:  
    无关位  
同步模式:  
    1 = 主模式（时钟来自内部 BRG）  
    0 = 从模式（时钟来自外部时钟源）
- bit 6      **TX9:** 9 位发送使能位  
    1 = 选择 9 位发送  
    0 = 选择 8 位发送
- bit 5      **TXEN:** 发送使能位 <sup>(1)</sup>  
    1 = 使能发送  
    0 = 禁止发送
- bit 4      **SYNC:** EUSART 模式选择位  
    1 = 同步模式  
    0 = 异步模式
- bit 3      **SENDB:** 发送间隔字符位  
异步模式:  
    1 = 在下一次发送时发送同步间隔字符（完成时由硬件清零）  
    0 = 同步间隔字符发送完毕  
同步模式:  
    无关位
- bit 2      **BRGH:** 高波特率选择位  
异步模式:  
    1 = 高速  
    0 = 低速  
同步模式:  
    在此模式下未使用
- bit 1      **TRMT:** 发送移位寄存器状态位  
    1 = TSR 为空  
    0 = TSR 已满
- bit 0      **TX9D:** 发送数据的第 9 位  
    可以是地址 / 数据位或奇偶校验位。

注 1: 同步模式下 SREN/CREN 的优先级高于 TXEN。

## 寄存器 25-2: RCSTA: 接收状态和控制寄存器 <sup>(1)</sup>

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R-0/0	R-0/0	R-x/x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **SPEN:** 串口使能位  
 1 = 使能串口 (将 RX/DT 和 TX/CK 引脚配置为串口引脚)  
 0 = 禁止串口 (保持在复位状态)
- bit 6      **RX9:** 9 位接收使能位  
 1 = 选择 9 位接收  
 0 = 选择 8 位接收
- bit 5      **SREN:** 单字节接收使能位  
异步模式:  
 无关位  
同步主模式:  
 1 = 使能单字节接收  
 0 = 禁止单字节接收  
 此位在接收完成后清零。  
同步从模式:  
 无关位
- bit 4      **CREN:** 连续接收使能位  
异步模式:  
 1 = 使能接收器  
 0 = 禁止接收器  
同步模式:  
 1 = 使能连续接收, 直到使能位 CREN 清零 (CREN 的优先级高于 SREN)  
 0 = 禁止连续接收
- bit 3      **ADDEN:** 地址检测使能位  
9 位异步模式 (RX9 = 1):  
 1 = 当 RSR<8> 置 1 时, 使能地址检测、允许中断和装入接收缓冲区  
 0 = 禁止地址检测、接收所有字节并且第 9 位可作为奇偶校验位  
8 位异步模式 (RX9 = 0):  
 无关位
- bit 2      **FERR:** 帧错误位  
 1 = 帧错误 (可以通过读 RCREG 寄存器更新该位并接收下一个有效字节)  
 0 = 无帧错误
- bit 1      **OERR:** 溢出错误位  
 1 = 溢出错误 (可以通过清零 CREN 位来清零该位)  
 0 = 无溢出错误
- bit 0      **RX9D:** 接收数据的第 9 位  
 该位可以是地址 / 数据位或奇偶校验位, 并且必须由用户固件计算得到。

# PIC16(L)F1933

## 寄存器 25-3: BAUDCON: 波特率控制寄存器

R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

### 图注:

R = 可读位

u = 不变

1 = 置 1

W = 可写位

x = 未知

0 = 清零

U = 未实现位, 读为 0

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

bit 7 **ABDOVF:** 自动波特率检测溢出位

#### 异步模式:

1 = 自动波特率定时器溢出

0 = 自动波特率定时器未溢出

#### 同步模式:

无关位

bit 6 **RCIDL:** 接收空闲标志位

#### 异步模式:

1 = 接收器空闲

0 = 已接收到起始位且接收器正在接收数据

#### 同步模式:

无关位

bit 5 **未实现:** 读为 0

bit 4 **SCKP:** 同步时钟极性选择位

#### 异步模式:

1 = 将数据电平取反后发送到 TX/CK 引脚

0 = 直接将数据发送到 TX/CK 引脚

#### 同步模式:

1 = 在时钟的上升沿传送数据

0 = 在时钟的下降沿传送数据

bit 3 **BRG16:** 16 位波特率发生器位

1 = 使用 16 位波特率发生器

0 = 使用 8 位波特率发生器

bit 2 **未实现:** 读为 0

bit 1 **WUE:** 唤醒使能位

#### 异步模式:

1 = 接收器正在等待下降沿。不接收任何字符, 字节 RCIF 置 1。RCIF 置 1 后, WUE 自动清零。

0 = 接收器正常工作

#### 同步模式:

无关位

bit 0 **ABDEN:** 自动波特率检测使能位

#### 异步模式:

1 = 使能自动波特率检测模式 (自动波特率检测完成后清零)

0 = 禁止自动波特率检测模式

#### 同步模式:

无关位

## 25.3 EUSART 波特率发生器 (BRG)

波特率发生器 (BRG) 是一个 8 位或 16 位定时器, 专用于支持异步和同步 EUSART 操作。默认情况下, BRG 工作在 8 位模式下。将 BAUDCON 寄存器的 BRG16 位置 1 可选择 16 位模式。

SPBRGH 和 SPBRGL 寄存器对决定自由运行的波特率定时器的周期。在异步模式下, 波特率周期的倍频值由 TXSTA 寄存器的 BRGH 位和 BAUDCON 寄存器的 BRG16 位决定。在同步模式下, BRGH 位被忽略。

表 25-3 提供了确定波特率的公式。例 25-1 提供了确定波特率和波特率误差的计算示例。

表 25-3 中给出了已计算好的各种异步模式下的典型波特率和误差值, 以供您使用。使用高波特率 (BRGH = 1) 或 16 位 BRG (BRG16 = 1) 有助于降低波特率误差。16 位 BRG 模式用于在快速振荡器频率条件下实现低波特率。

将新值写入 SPBRGH 和 SPBRGL 寄存器对将导致 BRG 定时器复位 (或清零)。这可以确保 BRG 无需等待定时器溢出就可以输出新的波特率。

如果在有效接收操作期间更改了系统时钟, 则可能导致接收错误或数据丢失。要避免这个问题, 请检查 RCIDL 位的状态以确保在更改系统时钟之前接收操作处于空闲状态。

### 例 25-1: 计算波特率误差

针对工作在异步模式下, Fosc 为 16 MHz, 目标波特率为 9600, 采用 8 位 BRG 的器件:

$$\text{目标波特率} = \frac{F_{osc}}{64([\text{SPBRGH}:\text{SPBRG}] + 1)}$$

求解 SPBRGH:SPBRGL:

$$X = \frac{F_{osc}}{\text{目标波特率}} - 1$$

$$= \frac{16000000}{9600} - 1$$

$$= [25.042] = 25$$

$$\text{计算得到的波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{误差} = \frac{\text{计算得到的波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

# PIC16(L)F1933

表 25-3: 波特率公式

配置位			BRG/EUSART 模式	波特率公式
SYNC	BRG16	BRGH		
0	0	0	8 位 / 异步	$F_{osc}/[64 (n+1)]$
0	0	1	8 位 / 异步	$F_{osc}/[16 (n+1)]$
0	1	0	16 位 / 异步	
0	1	1	16 位 / 异步	$F_{osc}/[4 (n+1)]$
1	0	x	8 位 / 同步	
1	1	x	16 位 / 同步	

图注: x = 任意值, n = SPBRGH 和 SPBRGL 寄存器对的值

表 25-4: 与波特率发生器相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	<a href="#">286</a>
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	<a href="#">285</a>
SPBRGL	BRG<7:0>								<a href="#">287*</a>
SPBRGH	BRG<15:8>								<a href="#">287*</a>
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	<a href="#">284</a>

图注: — = 未实现, 读为 0。波特率发生器不使用阴影单元。

\* 该页提供寄存器信息。



表 25-5: 异步模式下的波特率

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20,000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	1221	1.73	255	1200	0.00	239	1200	0.00	143
2400	2404	0.16	207	2404	0.16	129	2400	0.00	119	2400	0.00	71
9600	9615	0.16	51	9470	-1.36	32	9600	0.00	29	9600	0.00	17
10417	10417	0.00	47	10417	0.00	29	10286	-1.26	27	10165	-2.42	16
19.2k	19.23k	0.16	25	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8
57.6k	55.55k	-3.55	3	—	—	—	57.60k	0.00	7	57.60k	0.00	2
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 0											
	Fosc = 8,000 MHz			Fosc = 4,000 MHz			Fosc = 3.6864 MHz			Fosc = 1,000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	300	0.16	207	300	0.00	191	300	0.16	51
1200	1202	0.16	103	1202	0.16	51	1200	0.00	47	1202	0.16	12
2400	2404	0.16	51	2404	0.16	25	2400	0.00	23	—	—	—
9600	9615	0.16	12	—	—	—	9600	0.00	5	—	—	—
10417	10417	0.00	11	10417	0.00	5	—	—	—	—	—	—
19.2k	—	—	—	—	—	—	19.20k	0.00	2	—	—	—
57.6k	—	—	—	—	—	—	57.60k	0.00	0	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 32.000 MHz			Fosc = 20,000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	—	—	—
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.64k	2.12	16	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

# PIC16(L)F1933

表 25-5: 异步模式下的波特率 (续)

波特率	SYNC = 0, BRGH = 1, BRG16 = 0											
	Fosc = 8,000 MHz			Fosc = 4,000 MHz			Fosc = 3.6864 MHz			Fosc = 1,000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	—	—	—	—	—	—	—	—	—	300	0.16	207
1200	—	—	—	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19231	0.16	25	19.23k	0.16	12	19.2k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 32,000 MHz			Fosc = 20,000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303
1200	1200	-0.02	3332	1200	-0.03	1041	1200	0.00	959	1200	0.00	575
2400	2401	-0.04	832	2399	-0.03	520	2400	0.00	479	2400	0.00	287
9600	9615	0.16	207	9615	0.16	129	9600	0.00	119	9600	0.00	71
10417	10417	0.00	191	10417	0.00	119	10378	-0.37	110	10473	0.53	65
19.2k	19.23k	0.16	103	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35
57.6k	57.14k	-0.79	34	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11
115.2k	117.6k	2.12	16	113.636	-1.36	10	115.2k	0.00	9	115.2k	0.00	5

波特率	SYNC = 0, BRGH = 0, BRG16 = 1											
	Fosc = 8,000 MHz			Fosc = 4,000 MHz			Fosc = 3.6864 MHz			Fosc = 1,000 MHz		
	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)	实际波特率	误差 %	SPBRG 值 (十进制)
300	299.9	-0.02	1666	300.1	0.04	832	300.0	0.00	767	300.5	0.16	207
1200	1199	-0.08	416	1202	0.16	207	1200	0.00	191	1202	0.16	51
2400	2404	0.16	207	2404	0.16	103	2400	0.00	95	2404	0.16	25
9600	9615	0.16	51	9615	0.16	25	9600	0.00	23	—	—	—
10417	10417	0.00	47	10417	0.00	23	10473	0.53	21	10417	0.00	5
19.2k	19.23k	0.16	25	19.23k	0.16	12	19.20k	0.00	11	—	—	—
57.6k	55556	-3.55	8	—	—	—	57.60k	0.00	3	—	—	—
115.2k	—	—	—	—	—	—	115.2k	0.00	1	—	—	—

表 25-5: 异步模式下的波特率 (续)

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 32,000 MHz			Fosc = 20,000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	300.0	0.00	26666	300.0	0.00	16665	300.0	0.00	15359	300.0	0.00	9215
1200	1200	0.00	6666	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303
2400	2400	0.01	3332	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151
9600	9604	0.04	832	9597	-0.03	520	9600	0.00	479	9600	0.00	287
10417	10417	0.00	767	10417	0.00	479	10425	0.08	441	10433	0.16	264
19.2k	19.18k	-0.08	416	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143
57.6k	57.55k	-0.08	138	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47
115.2k	115.9k	0.64	68	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23

波特率	SYNC = 0, BRGH = 1, BRG16 = 1 或 SYNC = 1, BRG16 = 1											
	Fosc = 8,000 MHz			Fosc = 4,000 MHz			Fosc = 3.6864 MHz			Fosc = 1,000 MHz		
	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)	实际 波特率	误差 %	SPBRG 值 (十进制)
300	300.0	0.00	6666	300.0	0.01	3332	300.0	0.00	3071	300.1	0.04	832
1200	1200	-0.02	1666	1200	0.04	832	1200	0.00	767	1202	0.16	207
2400	2401	0.04	832	2398	0.08	416	2400	0.00	383	2404	0.16	103
9600	9615	0.16	207	9615	0.16	103	9600	0.00	95	9615	0.16	25
10417	10417	0	191	10417	0.00	95	10473	0.53	87	10417	0.00	23
19.2k	19.23k	0.16	103	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	12
57.6k	57.14k	-0.79	34	58.82k	2.12	16	57.60k	0.00	15	—	—	—
115.2k	117.6k	2.12	16	111.1k	-3.55	8	115.2k	0.00	7	—	—	—

# PIC16(L)F1933

## 25.3.1 自动波特率检测

EUSART 模块支持波特率自动检测和校准。

在自动波特率检测 (Auto-Baud Detect, ABD) 模式下, 提供给 BRG 的时钟是反向的。不是由 BRG 向传入的 RX 信号提供时钟, 而是由 RX 信号为 BRG 定时。波特率发生器用来测量接收 55h 字符的周期 (ASCII “U”, 也是 LIN 总线的同步字符)。该字符的惟一特性是它具有 5 个上升沿 (包括停止位边沿在内)。

将 BAUDCON 寄存器的 ABDEN 位置 1 可启动自动波特率校准序列 (图 25-6)。执行 ABD 序列时, EUSART 状态机保持空闲状态。在起始位之后, SPBRG 使用 BRG 计数器时钟 (如表 25-6 所示) 在接收信号的第一个上升沿开始计数。在第 8 个位周期结束时, 在 RX 引脚上出现第 5 个上升沿。此时, 会将相应 BRG 周期内累计所得的值保存到相应的 SPBRGH 和 SPBRGL 寄存器对中, ABDEN 位自动清零, 并将 RCIF 中断标志位置 1。要清零 RCIF 中断标志位, 需要读 RCREG 中的值。应丢弃 RCREG 的内容。当在不使用 SPBRGH 寄存器的模式下进行校准时, 用户可以通过检查 SPBRGH 寄存器中的值是否为 00h 来验证 SPBRGL 寄存器是否未溢出。

BRG 自动波特率时钟由表 25-6 中的 BRG16 和 BRGH 位决定。自动波特率检测期间, SPBRGH 和 SPBRGL 寄存器都可用作 16 位计数器, 与 BRG16 位的设置无关。校准波特率周期时, SPBRGH 和 SPBRGL 寄存器

以 BRG 基本时钟速率的 1/8 作为时钟。此时测量一个字的时间就相当于全速时钟模式下传输一位所需的平均时间。

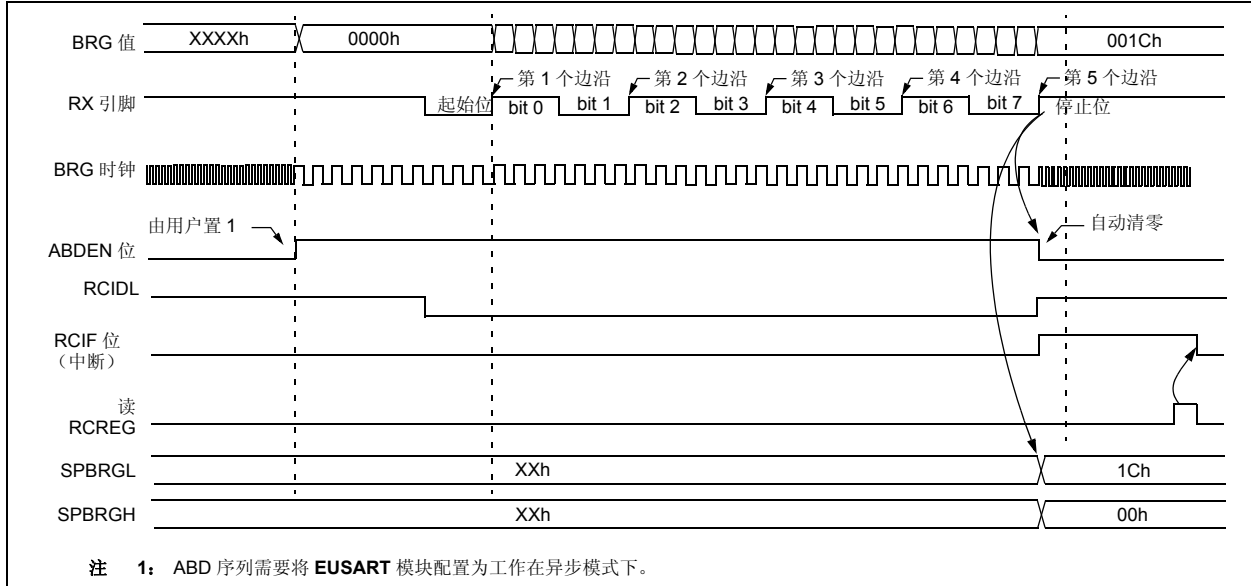
- 注 1:** 如果 WUE 位与 ABDEN 位都置 1, 则在间隔字符随后的字节上发生自动波特率检测 (见第 25.3.3 节 “接收到间隔字符时自动唤醒”)。
- 注 2:** 由用户判断传入字符的波特率是否在选定 BRG 时钟源的范围。可能无法实现振荡器频率和 EUSART 波特率的某些组合。
- 注 3:** 在自动波特率检测过程中, 自动波特率计数器从 1 开始计数。自动波特率序列完成后, 为了达到最大精度, 应将 SPBRGH:SPBRGL 寄存器对减 1。

表 25-6: BRG 计数器时钟速率

BRG16	BRGH	BRG 基本时钟	BRG ABD 时钟
0	0	Fosc/64	Fosc/512
0	1	Fosc/16	Fosc/128
1	0	Fosc/16	Fosc/128
1	1	Fosc/4	Fosc/32

**注:** ABD 序列执行期间, SPBRGL 和 SPBRGH 寄存器都可用作 16 位计数器, 与 BRG16 的设置无关。

图 25-6: 自动波特率校准



## 25.3.2 自动波特率溢出

在自动波特率检测过程中，如果在 RX 引脚上检测到第 5 个上升沿之前，波特率计数器溢出，则 BAUDCON 寄存器的 ABDOVF 位置 1。ABDOVF 位指示计数器超出了 SPBRGH:SPBRGL 寄存器对的 16 位所能允许的最大计数。在 ABDOVF 置 1 后，计数器将一直计数，直到在 RX 引脚上检测到第 5 个上升沿为止。检测到第 5 个 RX 边沿时，硬件将置 1 RCIF 中断标志位并清零 BAUDCON 寄存器的 ABDEN 位。随后通过读 RCREG 寄存器，RCIF 中断标志位也将清零。BAUDCON 寄存器的 ABDOVF 中断标志位可用软件直接清零。

要在 RCIF 中断标志位置 1 之前终止自动波特率检测过程，可依次将 BAUDCON 寄存器的 ABDEN 位和 ABDOVF 位清零。如果没有先将 ABDEN 位清零，则 ABDOVF 位保持置 1 状态。

## 25.3.3 接收到间隔字符时自动唤醒

休眠模式下，所有提供给 EUSART 的时钟都将暂停。从而使得波特率发生器无效且无法执行相应的字符接收操作。自动唤醒功能允许控 RX/DT 线路上的活动唤醒控制器。此功能仅在异步模式下可用。

通过将 BAUDCON 寄存器的 WUE 位置 1 可使能自动唤醒功能。置 1 后，禁止 RX/DT 引脚上的正常接收序列，且 EUSART 保持空闲状态，监视独立于 CPU 模式的唤醒事件。唤醒事件包括 RX/DT 线上从高到低的电平跳变。（这与同步间隔字符或 LIN 协议唤醒信号字符的启动条件一致。）

EUSART 模块在发生唤醒事件时产生 RCIF 中断。在正常 CPU 工作模式下，中断与 Q 时钟同步产生（图 25-7），如果器件处于休眠模式，则异步产生（图 25-8）。通过读 RCREG 寄存器可清除中断条件。

RX 线上间隔字符末尾处发生的从低到高的电平跳变可使 WUE 位自动清零。这将告诉用户间隔字符传输事件已经结束。此时，EUSART 模块处于空闲模式，等待接收下一个字符。

## 25.3.3.1 特殊注意事项

### 间隔字符

为了避免唤醒事件期间的字符错误或字符分割，唤醒字符必须全部为零。

使能唤醒时，该工作状况与数据流的低电平时间无关。如果 WUE 位置 1 且接收到有效的非零字符，则从起始位到第一个上升沿之间的低电平时间解释为唤醒事件。字符中的剩余位将被作为分割后的字符接收，而后续的字符会导致帧错误或溢出错误。

因此，发送的初始字符必须全部为 0，且其持续时间必须为 10 或更多的位时间，对于 LIN 总线，推荐采用 13 个位时间，或对于标准 RS-232 器件为任意数量的位时间。

### 振荡器起振时间

必须考虑振荡器起振时间，特别是在使用起振时间较长的振荡器（即 LP、XT 或 HS/PLL 模式）的应用中更是如此。同步间隔（或唤醒信号）字符必须足够长，并且跟有足够长的时间间隔，以便选中的振荡器有足够的时间起振并使 EUSART 正确初始化。

### WUE 位

唤醒事件可通过将 RCIF 位置 1 来产生接收中断。在 RX/DT 引脚信号的上升沿，WUE 位由硬件清零。然后通过读 RCREG 寄存器并丢弃其内容，用软件清除中断条件。

要确保没有丢失实际数据，应在将 WUE 位置 1 之前检查 RCIDL 位以验证是否正在进行接收。如果未在进行接收，则可将 WUE 位置 1，然后使器件进入休眠模式。

# PIC16(L)F1933

图 25-7: 正常工作期间的自动唤醒位 (WUE) 时序

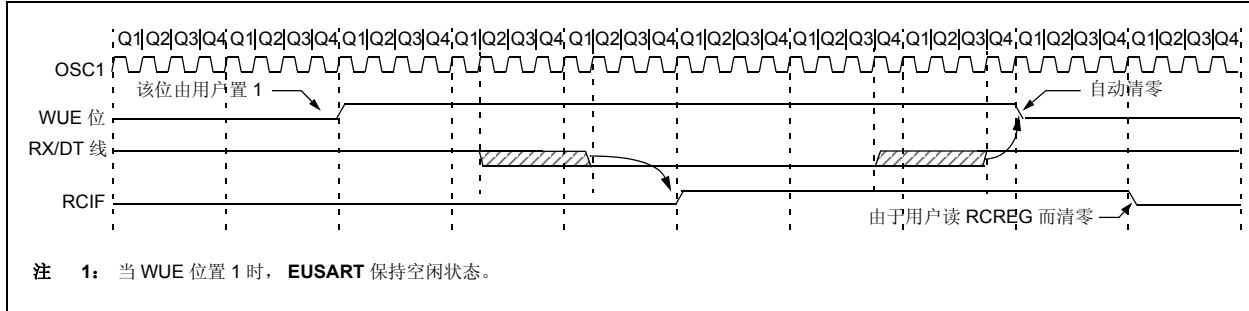
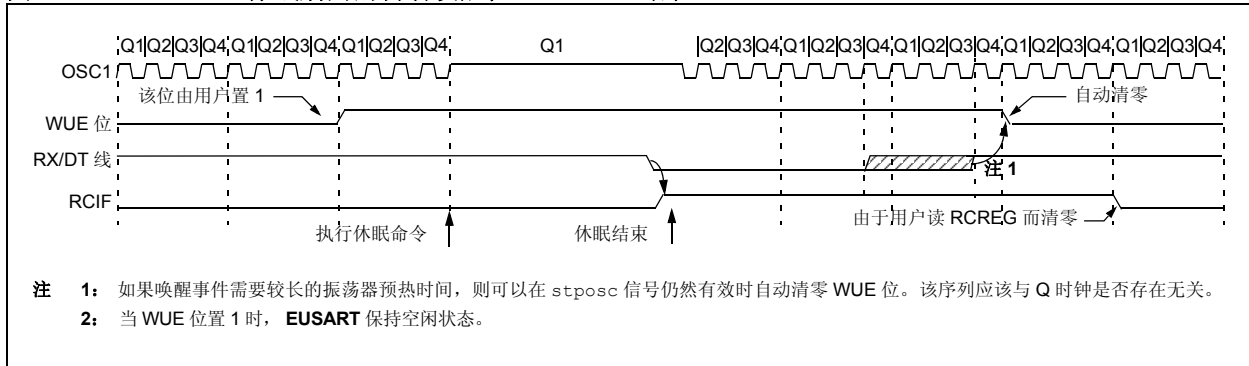


图 25-8: 休眠期间的自动唤醒位 (WUE) 时序



## 25.3.4 间隔字符序列

EUSART 模块可以发送符合 LIN 总线标准的特殊间隔字符序列。间隔字符包括一个起始位以及随后的 12 个 0 位和一个停止位。

要发送间隔字符，应将 TXSTA 寄存器的 SENDB 和 TXEN 位置 1。然后通过写 TXREG 来启动间隔字符的发送。写入 TXREG 的数据值会被忽略，而发送全 0。

在发送了相应的停止位之后，硬件会自动将 SENDB 位复位。这允许用户使用间隔字符（在 LIN 规范中通常指同步字符）后下一个要发送的字节预装载发送 FIFO。

正如其在正常发送操作中一样，TXSTA 寄存器的 TRMT 位指示发送操作处于有效状态还是空闲状态。请参见图 25-9 中间隔字符序列的时序。

### 25.3.4.1 间隔和同步字符发送序列

以下序列将启动由间隔字符及随后的自动波特率同步字节组成的报文帧头。该序列是 LIN 总线主器件的典型序列。

1. 将 EUSART 配置为需要的模式。
2. 将 TXEN 和 SENDB 位置 1，以启用间隔字符序列。
3. 使用无效字符（该值被忽略）装载 TXREG，以启动发送。
4. 将 55h 写入 TXREG，以将同步字符装入发送 FIFO 缓冲区。

5. 发送了间隔字符之后，由硬件将 SENDB 位复位，然后发送同步字符。

当 TXREG 为空（由 TXIF 位所示）时，将下一个数据字节写入 TXREG。

### 25.3.5 接收间隔字符

增强型 EUSART 模块可采用两种方式接收间隔字符。

检测间隔字符的第一种方法是使用 RCSTA 寄存器的 FERR 位和 RCREG 指示的已接收数据。假定波特率发生器已初始化为期望的波特率。

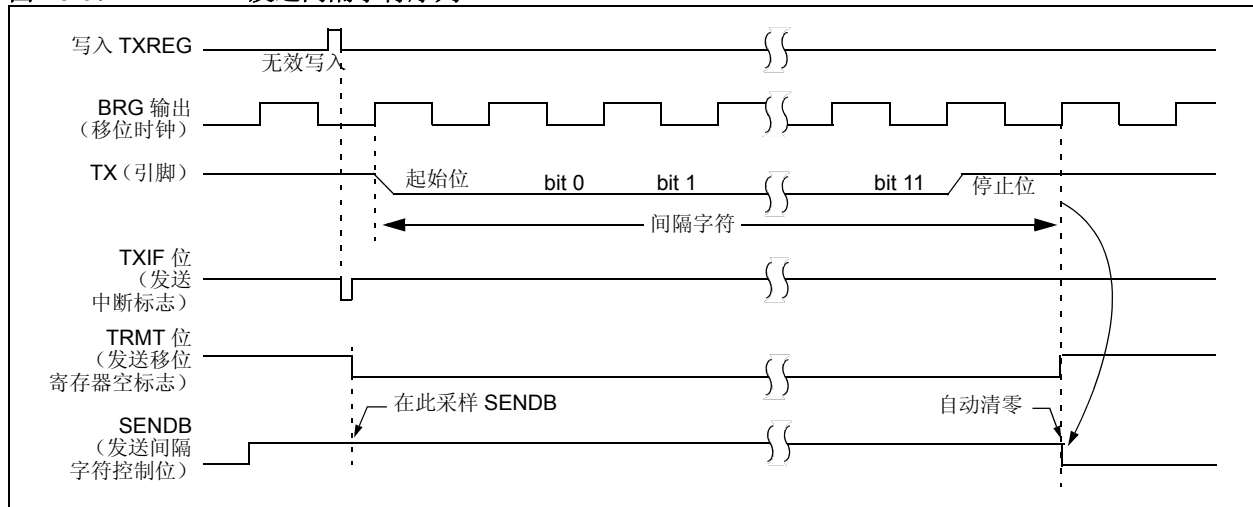
发生以下情况时，表示接收到间隔字符：

- RCIF 位置 1
- FERR 位置 1
- RCREG = 00h

第二种方法使用第 25.3.3 节“接收到间隔字符时自动唤醒”中描述的自动唤醒功能。通过使能此功能，EUSART 将在 RX/DT 引脚上采样下两次电平跳变，产生 RCIF 中断，接收下一个数据字节，之后再产生一次中断。

注意在间隔字符后，用户通常想要使能自动波特率检测功能。对于这两种方法，用户都可以在 EUSART 进入休眠模式之前，将 BAUDCON 寄存器的 ABDEN 位置 1。

图 25-9: 发送间隔字符序列



## 25.4 EUSART 同步模式

同步串行通信通常用于具有一个主器件和一个或多个从器件的系统中。主器件包含生成波特率所需的电路，并为系统中的所有器件提供时钟。从器件可使用主器件时钟，从而无需内部时钟发生电路。

同步模式下有两条信号线：一根双向数据线和一根时钟线。从器件使用主器件提供的外部时钟将串行数据移入或移出相应的接收和发送移位寄存器。由于数据线是双向的，所以同步操作只能是半双工的。半双工指主从器件都能够接收和发送数据，但不能同时进行。EUSART 可作为主器件，也可作为从器件。

同步发送时不使用起始位和停止位。

### 25.4.1 同步主模式

使用以下位将 EUSART 配置为同步主操作：

- SYNC = 1
- CSRC = 1
- SREN = 0（用于发送）； SREN = 1（用于接收）
- CREN = 0（用于发送）； CREN = 1（用于接收）
- SPEN = 1

将 TXSTA 寄存器的 SYNC 位置 1 可将器件配置为同步操作。将 TXSTA 寄存器的 CSRC 位置 1 可将器件配置为主器件。将 RCSTA 寄存器的 SREN 和 CREN 位清零可确保器件处于发送模式，否则器件将配置为接收模式。将 RCSTA 寄存器的 SPEN 位置 1 可启用 EUSART。

#### 25.4.1.1 主时钟

同步数据传输使用独立于数据线但与数据线同步的时钟线。配置为主器件的器件将时钟信号发送到 TX/CK 线上。EUSART 配置为同步发送或接收操作时，自动使能 TX/CK 引脚的输出驱动器。串行数据位在每个时钟的上升沿改变，以确保其在时钟的下降沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个时钟周期。

#### 25.4.1.2 时钟极性

提供了与 Microwire 兼容的时钟极性选项。时钟极性通过 BAUDCON 寄存器的 SCKP 位选择。将 SCKP 位置 1 可将时钟空闲状态设置为高电平。当 SCKP 位置 1 时，

数据在每个时钟的下降沿改变。将 SCKP 位清零可将空闲状态设置为低电平。当 SCKP 位清零时，数据在每个时钟的上升沿改变。

#### 25.4.1.3 同步主发送

从器件的 RX/DT 引脚输出数据。EUSART 配置为同步主发送操作时，自动使能 RX/DT 和 TX/CK 引脚的输出驱动器。

通过向 TXREG 寄存器写入一个字符来启动发送。如果 TSR 仍包含前一个字符的全部或部分，则新字符数据保存在 TXREG 中，直到前一个字符的最后一位发送完毕。如果这是第一个字符，或前一个字符被完全从 TSR 中送出，TXREG 中的数据就立即被传送到 TSR。字符发送在数据从 TXREG 送入 TSR 后立即开始。

每个数据位在主时钟的时钟上升沿改变，并在下一个时钟上升沿到来前保持有效。

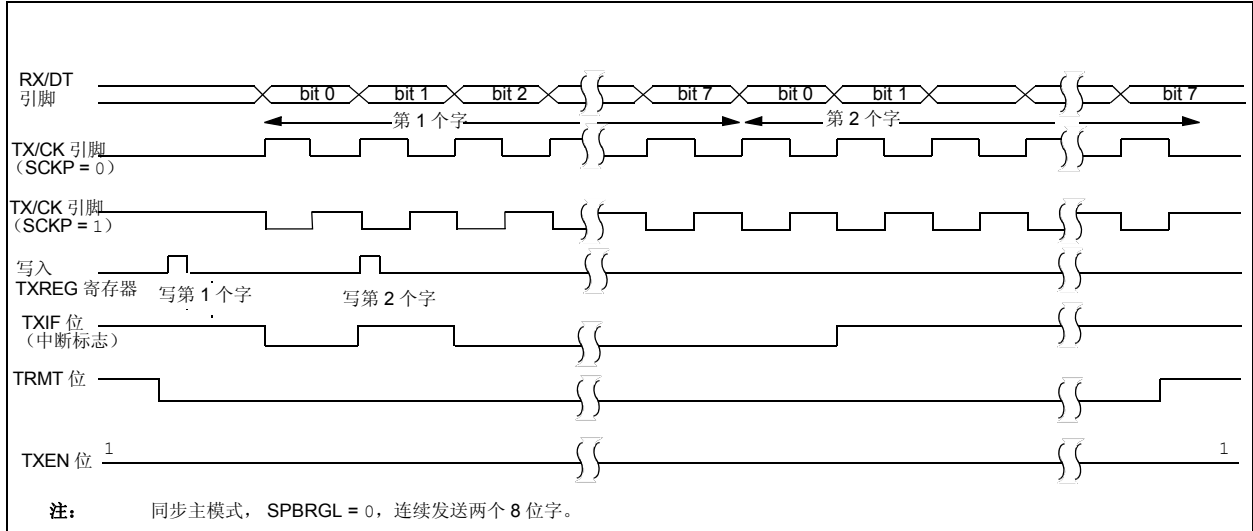
**注：** TSR 寄存器并未映射到数据存储寄存器中，因此用户不能访问它。

#### 25.4.1.4 同步主发送设置：

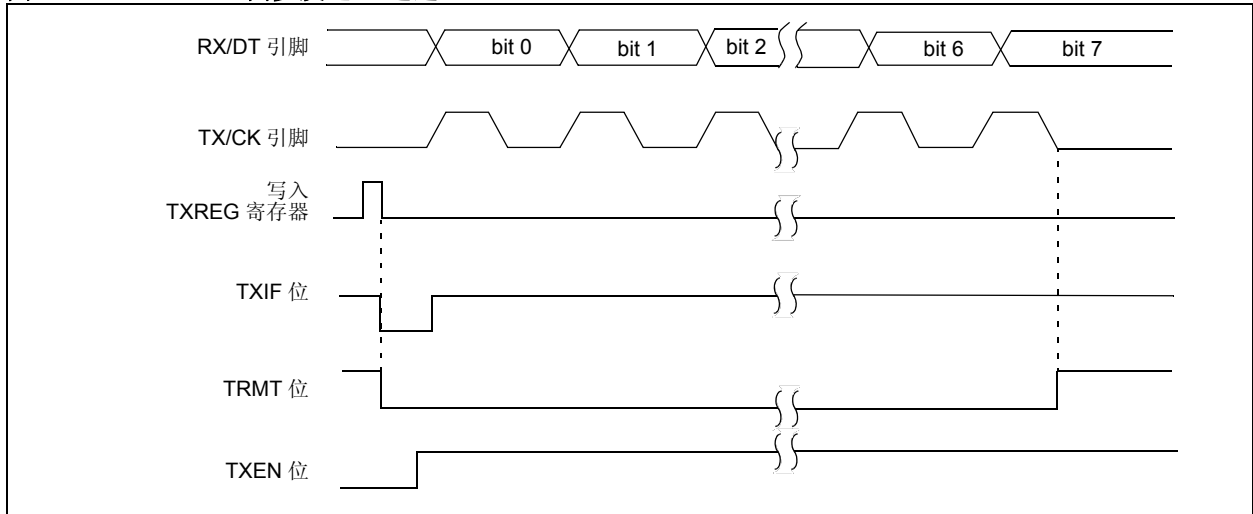
1. 初始化 SPBRGH 和 SPBRGL 寄存器对以及 BRGH 和 BRG16 位，以获得所需的波特率（见第 25.3 节“EUSART 波特率发生器 (BRG)”）。
2. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主串行端口。
3. 通过将 SREN 和 CREN 位清零，禁止接收模式。
4. 通过将 TXEN 位置 1，使能发送模式。
5. 如果需要发送 9 位数据，将 TX9 位置 1。
6. 如果需要中断，将 PIE1 寄存器的 TXIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
7. 如果选择发送 9 位数据，将第 9 位数据装入 TX9D 位。
8. 将数据装入 TXREG 寄存器，启动发送。



**图 25-10: 同步发送**



**图 25-11: 同步发送 (通过 TXEN)**



# PIC16(L)F1933

表 25-7: 与同步主发送相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	<a href="#">286</a>
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	<a href="#">86</a>
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	<a href="#">87</a>
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	<a href="#">90</a>
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	<a href="#">285</a>
SPBRGL	BRG<7:0>								<a href="#">287*</a>
SPBRGH	BRG<15:8>								<a href="#">287*</a>
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	<a href="#">129</a>
TXREG	EUSART 发送数据寄存器								<a href="#">277*</a>
TXSTA	CSRC	TX9	TXEN	SYNC	SEnDB	BRGH	TRMT	TX9D	<a href="#">284</a>

图注: — = 未实现单元, 读为 0。同步主发送不使用阴影单元。

\* 该页提供寄存器信息。

## 25.4.1.5 同步主接收

在 RX/DT 引脚上接收数据。EUSART 配置为同步主接收操作时，自动禁止 RX/DT 引脚的输出驱动器。

在同步模式下，可通过将单字节接收使能位（RCSTA 寄存器的 SREN）或连续接收使能位（RCSTA 寄存器的 CREN）置 1 来使能接收。

SREN 置 1 且 CREN 清零时，一个字符中有多少数据位就产生多少个时钟周期。一个字符接收完成后 SREN 位自动清零。CREN 置 1 时，将连续产生时钟直到 CREN 清零。如果 CREN 在字符接收过程中清零，则 CK 时钟立即停止，字符中收到的那部分将被丢弃。如果 SREN 和 CREN 同时置 1，则第一个字符接收完成时 SREN 清零，CREN 的优先级更高。

要启动接收，将 SREN 或 CREN 置 1。在 TX/CK 时钟引脚的下降沿对 RX/DT 引脚上的数据进行采样，并将采样结果移入接收移位寄存器（RSR）。当 RSR 接收到一个完整的字符后，RCIF 位置 1 且该字符自动传送到两个字符的接收 FIFO。接收 FIFO 中顶部字符的低 8 位在 RCREG 中。只要接收 FIFO 中有未读字符，RCIF 位就保持置 1 状态。

## 25.4.1.6 从时钟

同步数据传输使用独立于数据线但与数据线同步的时钟线。配置为从器件的器件在 TX/CK 线上接收时钟信号。器件配置为同步从发送或接收操作时，自动禁止 TX/CK 引脚的输出驱动器。串行数据位在每个时钟的上升沿改变，以确保其在时钟的下降沿有效。每个时钟周期传送一个数据位。数据位有多少，就产生多少个接收时钟周期。

## 25.4.1.7 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。如果在读 RCREG 以访问 FIFO 之前接收到完整的第三个字符，则会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 中的前一个数据不会被覆盖。FIFO 缓冲区中的两个字符可被读出，但溢出错误清除之前不会接收其他字符。只能通过清除溢出条件来清零 OERR 位。如果 SREN 位置 1 且 CREN 位清零时发生溢出错误，则可

通过读 RCREG 来清除该错误。如果 CREN 位置 1 时发生溢出，则可通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位（复位 EUSART）来清除错误条件。

## 25.4.1.8 接收 9 位字符

EUSART 支持 9 位字符接收。当 RCSTA 寄存器的 RX9 位置 1 时，EUSART 将在接收每个字符时将 9 个位移入 RSR。RCSTA 寄存器的 RX9D 位是第 9 位，也是接收 FIFO 顶部未读字符的最高有效数据位。从接收 FIFO 缓冲区读 9 位数据时，在读 RCREG 的低 8 位之前必须先读 RX9D 数据位。

## 25.4.1.9 同步主接收设置：

1. 初始化 SPBRGH 和 SPBRGL 寄存器对，以设置合适的波特率。根据需要 BRGH 和 BRG16 位置 1 或清零，以获得所需的波特率。
2. 将 RX 引脚的 ANSEL 位清零（如果适用）。
3. 通过将 SYNC、SPEN 和 CSRC 位置 1，使能同步主串行端口。
4. 确保 CREN 和 SREN 位清零。
5. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
6. 若需要接收 9 位数据，将 RX9 位置 1。
7. 将 SREN 位置 1 启动接收，或将 CREN 位置 1 启动连续接收。
8. 字符接收完成时，中断标志位 RCIF 置 1。如果中断允许位 RCIE 也置 1，则产生中断。
9. 读 RCSTA 寄存器以获取第 9 个数据位（如果已使能），并判断在接收过程中是否发生了任何错误。
10. 通过读 RCREG 寄存器来读取接收到的 8 位数据。
11. 如果发生了溢出错误，则可通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位使 EUSART 复位来清除该错误。

# PIC16(L)F1933

图 25-12: 同步接收 (主模式, SREN)

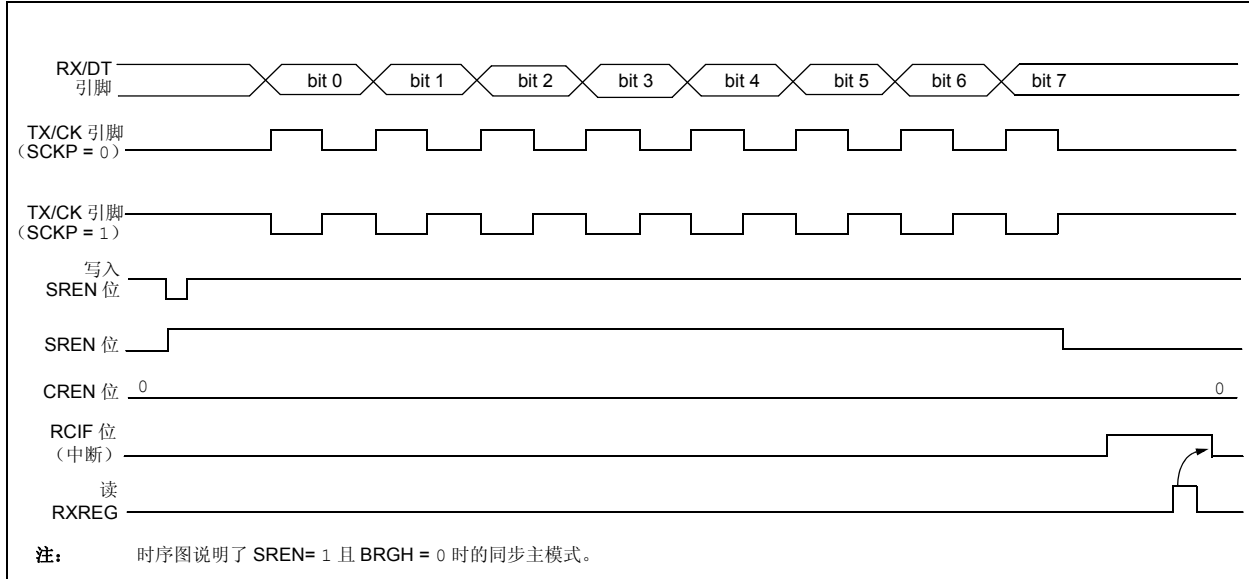


表 25-8: 与同步主接收相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	286
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								280*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
SPBRGL	BRG<7:0>								287*
SPBRGH	BRG<15:8>								287*
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	284

图注: — = 未实现单元, 读为 0。同步主接收不使用阴影单元。

\* 该页提供寄存器信息。

## 25.4.2 同步从模式

使用以下位将 EUSART 配置为同步从操作：

- SYNC = 1
- CSRC = 0
- SREN = 0（用于发送）； SREN = 1（用于接收）
- CREN = 0（用于发送）； CREN = 1（用于接收）
- SPEN = 1

将 TXSTA 寄存器的 SYNC 位置 1 可将器件配置为同步操作。将 TXSTA 寄存器的 CSRC 位清零可将器件配置为从器件。将 RCSTA 寄存器的 SREN 和 CREN 位清零可确保器件处于发送模式，否则器件将配置为接收模式。将 RCSTA 寄存器的 SPEN 位置 1 可启用 EUSART。

### 25.4.2.1 EUSART 同步从发送

除了休眠模式以外，同步主发送模式和同步从发送模式的工作原理是相同的（见第 25.4.1.3 节“同步主发送”）。

如果向 TXREG 写入两个字，然后执行 SLEEP 指令，则会发生以下事件：

1. 第一个字符将立即传送到 TSR 寄存器并发送。
2. 第二个字保留在 TXREG 寄存器中。
3. TXIF 位不会置 1。
4. 第一个字符移出 TSR 后，TXREG 寄存器将第二个字符传送到 TSR，此时 TXIF 位置 1。
5. 如果 PEIE 和 TXIE 位均置 1，则中断会将器件从休眠状态唤醒并执行下一条指令。如果 GIE 位也置 1，程序将调用中断服务程序。

### 25.4.2.2 同步从发送设置：

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 将 CK 引脚的 ANSEL 位清零（如果适用）。
3. 将 CREN 和 SREN 位清零。
4. 如果需要中断，将 PIE1 寄存器的 TXIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
5. 如果需要发送 9 位数据，将 TX9 位置 1。
6. 通过将 TXEN 位置 1，使能发送。
7. 如果选择发送 9 位数据，则将最高有效位插入 TX9D 位。
8. 通过将低 8 位写入 TXREG 寄存器来启动发送。

**表 25-9: 与同步从发送相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	286
INTCON	GIE	PEIE	TMR0IE	INTE	IOCIE	TMR0IF	INTF	IOCIF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
TXREG	EUSART 发送数据寄存器								277*
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	284

**图注：** — = 未实现单元，读为 0。同步从发送不使用阴影单元。

\* 该页提供寄存器信息。

# PIC16(L)F1933

## 25.4.2.3 EUSART 同步从接收

除下列各项外，同步主接收模式和同步从接收模式的工作原理是相同的（第 25.4.1.5 节“同步主接收”）：

- 休眠
- CREN 位始终置 1，因此接收器从不空闲
- SREN 位在从模式下为无关位

在进入休眠模式之前，通过将 CREN 位置 1 可在休眠模式下接收一个字符。接收到该字后，RSR 寄存器将数据发送到 RCREG 寄存器。如果 RCIE 中断允许位置 1，则发生的中断会将器件从休眠状态唤醒并执行下一条指令。如果 GIE 位也置 1，程序将跳转到中断向量。

## 25.4.2.4 同步从接收设置：

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 将 CK 和 DT 引脚的 ANSEL 位清零（如果适用）。
3. 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 如果需要接收 9 位数据，将 RX9 位置 1。
5. 将 CREN 位置 1，以启用接收。
6. 接收完成后，RCIF 位置 1。如果 RCIE 位也置 1，则产生中断。
7. 如果使能了 9 位模式，从 RCSTA 寄存器的 RX9D 位取出最高有效位。
8. 通过读 RCREG 寄存器从接收 FIFO 取出低 8 位。
9. 如果发生了溢出错误，则可通过清零 RCSTA 寄存器的 CREN 位或清零 SPEN 位（复位 EUSART）来清除该错误。

表 25-10: 与同步从接收相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
BAUDCON	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	286
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	86
PIE1	TMR1GIE	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	87
PIR1	TMR1GIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	90
RCREG	EUSART 接收数据寄存器								280*
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	285
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	129
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	284

图注： — = 未实现单元，读为 0。同步从接收不使用阴影单元。

\* 该页提供寄存器信息。

## 25.5 休眠期间的 EUSART 工作原理

EUSART 只有在同步从模式下，才会在休眠期间保持有效状态。所有其他模式都需要系统时钟，因此在休眠期间无法产生运行发送移位寄存器或接收移位寄存器所需的信号。

同步从模式使用外部产生的时钟来运行发送移位寄存器和接收移位寄存器。

### 25.5.1 休眠期间的同步接收

要在休眠模式下接收，在进入休眠模式之前必须满足以下所有条件：

- RCSTA 和 TXSTA 控制寄存器必须配置为同步从接收模式（见第 25.4.2.4 节“同步从接收设置：”）。
- 如果需要中断，将 PIE1 寄存器的 RCIE 位以及 INTCON 寄存器的 GIE 和 PEIE 位置 1。
- 必须通过读 RCREG 来清零 RCIF 中断标志位，以卸载接收缓冲区中等待处理的任何字符。

进入休眠模式时，器件已准备好接收分别在 RX/DT 和 TX/CK 引脚上的数据和时钟信号。在数据字随着由外部器件提供的时钟完全移入后，PIR1 寄存器的 RCIF 中断标志位将置 1。从而将处理器从休眠模式唤醒。

从休眠模式唤醒时，将执行 SLEEP 指令后紧跟的指令。如果 INTCON 寄存器的全局中断允许位 GIE 也置 1，将调用地址 004h 处的中断服务程序。

### 25.5.2 休眠期间的同步发送

要在休眠模式下进行发送，在进入休眠模式之前必须满足以下所有条件：

- RCSTA 和 TXSTA 控制寄存器必须配置为同步从发送模式（见第 25.4.2.2 节“同步从发送设置：”）。
- 必须通过将输出数据写入 TXREG 来清零 TXIF 中断标志位，从而填充 TSR 和发送缓冲区。
- 如果需要中断，将 PIE1 寄存器的 TXIE 位以及 INTCON 寄存器的 PEIE 位置 1。
- PIE1 寄存器的中断允许位 TXIE 以及 INTCON 寄存器的 PEIE 位必须置 1。

进入休眠模式时，器件已准备好在 TX/CK 引脚上接收时钟信号，在 RX/DT 引脚上发送数据。TSR 中的数据字随着由外部器件提供的时钟完全移出后，TXREG 中等待处理的字节将传送到 TSR，TXIF 中断标志位置 1。从而将处理器从休眠模式唤醒。此时，TXREG 可接收其他字符以进行发送，此操作将清零 TXIF 中断标志位。

从休眠模式唤醒时，将执行 SLEEP 指令后紧跟的指令。如果全局中断允许位（GIE）也置 1，将调用地址 0004h 处的中断服务程序。

# PIC16(L)F1933

---

注:

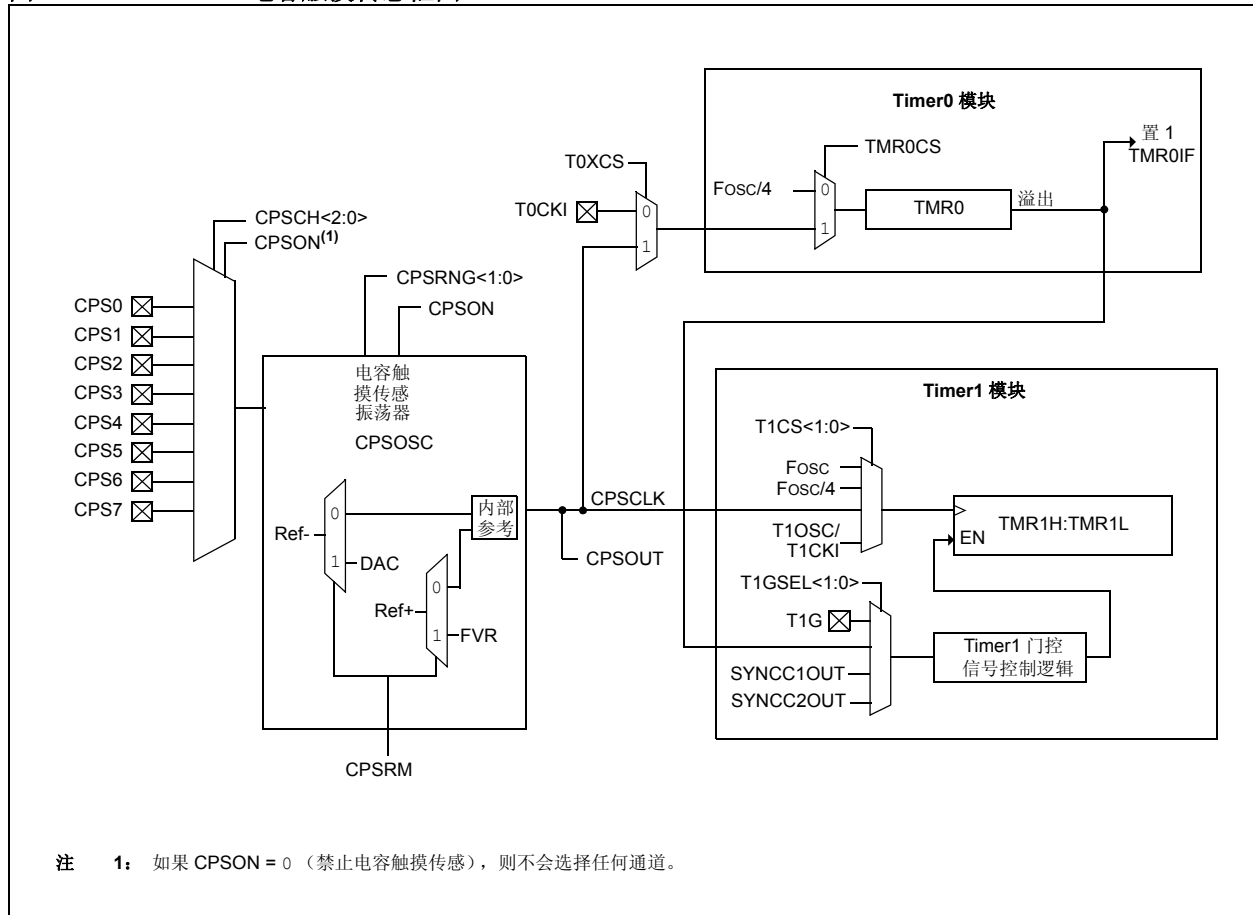


## 26.0 电容触摸传感（CPS）模块

电容触摸传感（CPS）模块允许不通过机械接口便可与最终用户交互。在典型应用中，CPS 模块连接至印刷电路板（Printed Circuit Board, PCB）上的一个焊盘，后者与最终用户电气隔离。当最终用户将手指置于 PCB 焊盘上时，就加入了一个容性负载，导致 CPS 模块的频率漂移。CPS 模块需要软件和至少一个定时器资源来确定频率变化。模块的主要功能包括：

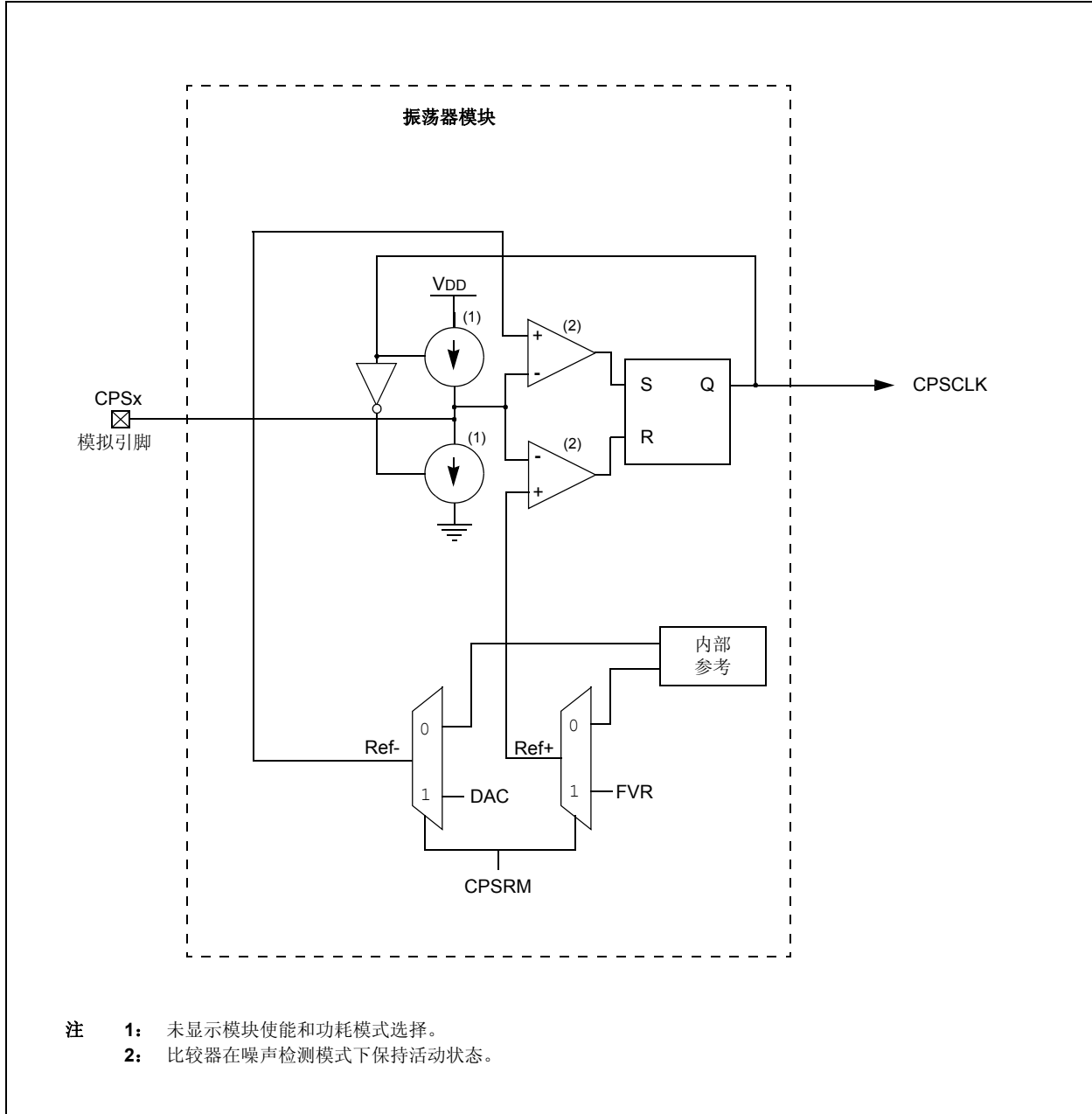
- 监视多个输入的模拟多路开关
- 电容触摸传感振荡器
- 多种功耗模式
- 具有可变参考电压的高功耗范围
- 多个定时器资源
- 软件控制
- 在休眠期间工作

图 26-1: 电容触摸传感框图



# PIC16(L)F1933

图 26-2: 电容触摸传感振荡器框图



## 26.1 模拟多路开关

CPS 模块最多监视 16 路输入。电容触摸传感输入定义为 CPS<15:0>。要确定是否发生了频率变化，用户必须：

- 通过设置 CPSCON1 寄存器的 CPSCH<2:0> 位来选择合适的 CPS 引脚。
- 将相应的 ANSEL 位置 1。
- 将相应的 TRIS 位置 1。
- 运行软件算法。

在模块使能时选择 CPSx 引脚会导致电容触摸传感振荡信号出现在 CPSx 引脚上。如果未将相应的 ANSEL 和 TRIS 位置 1，会导致电容触摸传感振荡器停止，最终导致读取到的频率无效。

## 26.2 电容触摸传感振荡器

电容触摸传感振荡器的输出由恒定的拉电流和恒定的灌电流组成，以产生三角波。CPSCON0 寄存器的 CPSOUT 位显示电容触摸传感振荡器的状态，即其是灌电流还是拉电流。振荡器用来驱动容性负载（单个 PCB 焊盘），同时作为 Timer0 或 Timer1 的时钟源。振荡器有三种不同的电流设置，由 CPSCON0 寄存器的 CPSRNG<1:0> 定义。振荡器的不同电流设置有两种用途：

- 最大化固定时基下的定时器的计数个数。
- 在频率变化时，最大化定时器的计数差值。

## 26.3 参考电压

电容触摸传感振荡器使用参考电压为振荡提供两个电压阈值。上电压阈值称为 Ref+，下电压阈值称为 Ref-。

用户可以选择使用固定参考电压（在电容触摸传感振荡器内部）或可变参考电压（由固定参考电压（FVR）模块和数模转换器（DAC）模块提供）。

使用固定参考电压时，Vss 电压决定下阈值电平（Ref-），VDD 电压决定上阈值电平（Ref+）。

使用可变参考电压时，DAC 电压决定下阈值电平（Ref-），FVR 电压决定上阈值电平（Ref+）。使用这些参考源的优点是振荡器频率在 VDD 变化时保持恒定。

通过使用这些可变参考电压可以获得不同的振荡频率。高参考电压电平降的越多，且低参考电压电平升的越多，电容触摸传感振荡器频率就会越高。

参考电压的选择由 CPSCON0 寄存器的 CPSRM 位控制。此位置 1 表示选择可变参考电压，此位清零表示选择固定参考电压。

更多关于配置可变电压电平的信息，请参见第 14.0 节“固定参考电压（FVR）”和第 17.0 节“数模转换器（DAC）模块”。

# PIC16(L)F1933

## 26.4 功耗模式

电容触摸传感振荡器可以在七种不同的功耗模式之一下工作。功耗模式划分为两个范围：低功耗范围和高功耗范围。

选择了振荡器的低功耗范围时，使用电容触摸传感振荡器的固定内部参考电压。选择了振荡器的高功耗范围时，使用由 FVR 和 DAC 模块提供的可变参考电压。参考电压的选择由 CPSCON0 寄存器的 CPSRM 位控制。更多信息，请参见第 26.3 节“参考电压”。

在每个范围内，都有三种不同的功耗模式：低、中和高。电流消耗取决于选择的范围和模式。可通过配置 CPSCON0 寄存器的 CPSRNG <1:0> 位来选择每个范围内的功耗模式。正确的功耗模式选择请参见表 26-1。

还有一种模式是噪声检测模式，位于高功耗范围内。噪声检测模式是惟一的，它禁止了模拟引脚上的灌电流和拉电流，但是保持振荡器电路的其余部分仍处于活动状态。这可以将模拟引脚上的振荡频率减至零，还大大减少了振荡器模块消耗的电流。

当有噪声进入引脚时，振荡器以此噪声决定的频率驱动。这会在比较器输出上产生可检测的信号，指示引脚上有活动。

图 26-2 显示了拉电流以及与振荡器相关的比较器的详细图。

表 26-1: 功耗模式选择

CPSRM <sup>(2)</sup>	范围	CPSRNG<1:0>	模式	标称电流 <sup>(1)</sup>
0	低	00	关闭	0.0 $\mu$ A
		01	低功耗	0.1 $\mu$ A
		10	中等功耗	1.2 $\mu$ A
		11	高功耗	18 $\mu$ A
1	高	00	噪声检测	0.0 $\mu$ A
		01	低功耗	9 $\mu$ A
		10	中等功耗	30 $\mu$ A
		11	高功耗	100 $\mu$ A

注 1: 更多信息，请参见应用的电气规范章节。

## 26.5 定时器资源

要测量电容触摸传感振荡器的频率变化，需要固定时基。在固定时基周期内，可使用电容触摸传感振荡器为 Timer0 或 Timer1 提供时钟。电容触摸传感振荡器的频率等于定时器计数值除以固定时基周期。

## 26.6 固定时基

要测量电容触摸传感振荡器的频率，需要固定时基。任何定时器资源或软件循环都能用来建立固定时基。可由最终用户决定以何种方式产生固定时基。

**注：** 固定时基不能由将电容触摸传感振荡器用作时钟源的定时器资源产生。

### 26.6.1 TIMER0

要选择 Timer0 作为 CPS 模块的定时器资源：

- 将 CPSCON0 寄存器的 T0XCS 位置 1。
- 将 OPTION\_REG 寄存器的 TMR0CS 位清零。

当选择 Timer0 作为定时器资源时，电容触摸传感振荡器将成为 Timer0 的时钟源。其他信息，请参见第 20.0 节“Timer0 模块”。

### 26.6.2 TIMER1

要选择 Timer1 作为 CPS 模块的定时器资源，请将 T1CON 寄存器的 TMR1CS<1:0> 设置为 11。选择 Timer1 作为定时器资源时，电容触摸传感振荡器将成为 Timer1 的时钟源。因为 Timer1 模块具有门控功能，使用 Timer0 溢出标志可以简化用于频率测量的时基的构建。

建议将 Timer0 溢出标志与 Timer1 门控的交替计数模式结合使用，产生 CPS 模块的软件部分所需的固定时基。其他信息，请参见第 21.12 节“Timer1 门控寄存器”。

**表 26-2: TIMER1 使能功能**

TMR1ON	TMR1GE	Timer1 工作状态
0	0	关闭
0	1	关闭
1	0	开启
1	1	通过输入使能计数

## 26.7 软件控制

要确定电容触摸传感振荡器的频率变化，需要 CPS 模块的软件部分。这可以通过以下步骤实现：

- 设置固定时基来获取 Timer0 或 Timer1 上的计数。
- 确定电容触摸传感振荡器的标称频率。
- 确定由于额外容性负载造成电容触摸传感振荡器频率降低的量。
- 设置频率阈值。

### 26.7.1 标称频率（无容性负载）

要确定电容触摸传感振荡器的标称频率：

- 移除选定 CPSx 引脚上的额外容性负载。
- 在固定时基开始时，清零定时器资源。
- 在固定时基结束时，保存定时器资源中的值。

定时器资源的值等于电容触摸传感振荡器在给定时基内的振荡次数。电容触摸传感振荡器的频率等于定时器计数值除以固定时基周期。

### 26.7.2 降低的频率（额外的容性负载）

额外的容性负载会导致电容触摸传感振荡器的频率降低。要确定电容触摸传感振荡器频率的降低量：

- 在选定 CPSx 引脚上添加典型容性负载。
- 使用测量标称频率时的固定时基。
- 在固定时基开始时，清零定时器资源。
- 在固定时基结束时，保存定时器资源中的值。

定时器资源的值等于有额外容性负载的电容触摸传感振荡器的振荡次数。电容触摸传感振荡器的频率等于定时器计数值除以固定时基周期。此频率应该小于在标称频率测量时获得的值。

# PIC16(L)F1933

---

## 26.7.3 频率阈值

频率阈值应置于电容触摸传感振荡器的标称频率和降低后的频率之间。更多关于 CPS 模块所需的软件的详细信息，请参见应用笔记 AN1103 《电容触摸传感的软件处理》(DS01103A\_CN)。

**注：** 更多关于常规电容触摸传感的信息，请参见应用笔记：

- AN1101 《电容触摸传感简介》(DS01101A\_CN)
- AN1102 《电容触摸传感布板和物理设计指南》(DS01102A\_CN)

## 26.8 休眠期间的工作

只要模块使能，电容触摸传感振荡器便可持续运行，即使器件处于休眠状态也是如此。为了让软件能够判断是否发生了频率变化，器件必须被唤醒。然而，在定时器资源采集计数时，不必唤醒器件。

**注：** Timer0 在休眠期间不能工作，因此在器件休眠时不能用于电容触摸传感测量。

## 寄存器 26-1: CPSCON0: 电容触摸传感控制寄存器 0

R/W-0/0	R/W-0/0	U-0	U-0	R/W-0/0	R/W-0/0	R-0/0	R/W-0/0
CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	T0XCS
bit 7						bit 0	

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	

- bit 7      **CPSON:** CPS 模块使能位  
 1 = 使能 CPS 模块  
 0 = 禁止 CPS 模块
- bit 6      **CPSRM:** 电容触摸传感参考电压模式位  
 1 = CPS 模块处于高功耗范围。DAC 和 FVR 提供振荡器参考电压。  
 0 = CPS 模块处于低功耗范围。使用内部振荡器参考电压。
- bit 5-4    **未实现:** 读为 0
- bit 3-2    **CPSRNG<1:0>:** 电容触摸传感电流范围位  
如果 CPSRM = 0 (低功耗范围):  
 00 = 振荡器关闭  
 01 = 振荡器处于低量程。充电 / 放电电流标称值为 0.1  $\mu$ A  
 10 = 振荡器处于中量程。充电 / 放电电流标称值为 1.2  $\mu$ A  
 11 = 振荡器处于高量程。充电 / 放电电流标称值为 18  $\mu$ A
- 如果 CPSRM = 1 (高功耗范围):  
 00 = 振荡器开启。噪声检测模式。不提供充电 / 放电电流。  
 01 = 振荡器处于低量程。充电 / 放电电流标称值为 9  $\mu$ A  
 10 = 振荡器处于中量程。充电 / 放电电流标称值为 30  $\mu$ A  
 11 = 振荡器处于高量程。充电 / 放电电流标称值为 100  $\mu$ A
- bit 1      **CPSOUT:** 电容触摸传感振荡器状态位  
 1 = 振荡器在拉电流 (流出引脚的电流)  
 0 = 振荡器在灌电流 (流入引脚的电流)
- bit 0      **T0XCS:** Timer0 外部时钟源选择位  
如果 TMR0CS = 1:  
 T0XCS 位控制哪个位于内核 /Timer0 模块外部的时钟作为 Timer0 的时钟源:  
 1 = Timer0 时钟源是电容触摸传感振荡器  
 0 = Timer0 时钟源是 T0CKI 引脚  
如果 TMR0CS = 0:  
 Timer0 时钟源由内核 /Timer0 模块控制, 为 Fosc/4

# PIC16(L)F1933

## 寄存器 26-2: CPSCON1: 电容触摸传感控制寄存器 1

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	CPSCH<2:0>		
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-3

未实现: 读为 0

bit 2-0

**CPSCH<2:0>**: 电容触摸传感通道选择位

如果 **CPSON = 0**:

忽略这些位。没有通道被选中。

如果 **CPSON = 1**:

000 = 通道 0 (CPS0)

001 = 通道 1 (CPS1)

010 = 通道 2 (CPS2)

011 = 通道 3 (CPS3)

100 = 通道 4 (CPS4)

101 = 通道 5 (CPS4)

110 = 通道 6 (CPS4)

111 = 通道 7 (CPS4)



**表 26-3: 与电容触摸传感相关的寄存器汇总**

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
ANSELA	—	—	ANSA5	ANSA4	ANSA3	ANSA2	ANSA1	ANSA0	<a href="#">121</a>
ANSELB	—	—	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0	<a href="#">126</a>
CPSCON0	CPSON	CPSRM	—	—	CPSRNG<1:0>		CPSOUT	T0XCS	<a href="#">311</a>
CPSCON1	—	—	—	—	—	CPSCH<2:0>			<a href="#">312</a>
OPTION_REG	$\overline{\text{WPUEN}}$	INTEDG	TMR0CS	TMR0SE	PSA	PS2	PS1	PS0	<a href="#">175</a>
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	$\overline{\text{T1SYNC}}$	—	TMR1ON	<a href="#">185</a>
TRISA	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	<a href="#">120</a>
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	<a href="#">125</a>

图注： — = 未实现单元，读为 0。CPS 模块不使用阴影单元。

# PIC16(L)F1933

---

注:

## 27.0 液晶显示 (LCD) 驱动模块

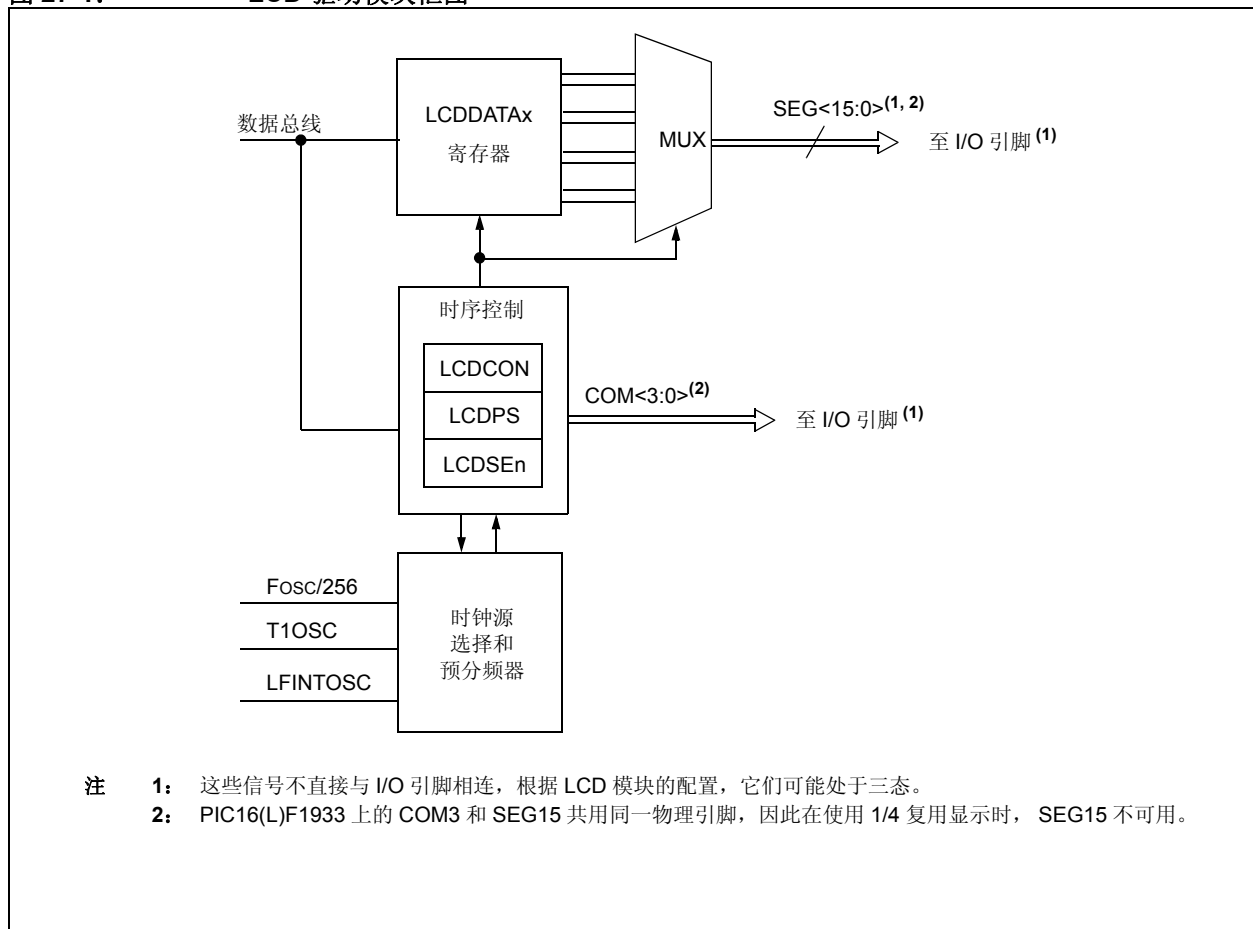
液晶显示 (Liquid Crystal Display, LCD) 驱动模块对静态或复用的 LCD 面板进行时序控制。在 PIC16(L)F1933 器件中, 该模块最多能驱动 4 个公共端和 24 段的面板。该 LCD 模块同时还可以控制 LCD 像素数据。

**注:** PIC16(L)F1933 上的 COM3 和 SEG15 共用同一物理引脚, 因此在使用 1/4 复用显示时, SEG15 不可用。

LCD 驱动模块支持:

- 直接驱动 LCD 面板
- 3 个预分频比可选的 LCD 时钟源
- 最多 4 个公共端:
  - 静态 (1 个公共端)
  - 1/2 复用 (2 个公共端)
  - 1/3 复用 (3 个公共端)
  - 1/4 复用 (4 个公共端)
- 段引脚数最多:
  - 16 (PIC16(L)F1933)
- 静态、1/2 或 1/3 LCD 偏置

图 27-1: LCD 驱动模块框图



# PIC16(L)F1933

## 27.1 LCD 寄存器

LCD 模块包含以下寄存器：

- LCD 控制寄存器 (LCDCON)
- LCD 相位寄存器 (LCDPS)
- LCD 参考梯形电阻网络寄存器 (LCDRL)
- LCD 对比度控制寄存器 (LCDGST)
- LCD 参考电压控制寄存器 (LCDREF)
- 最多 3 个 LCD 段使能寄存器 (LCDSEn)
- 最多 12 个 LCD 数据寄存器 (LCDDATAn)

表 27-1: LCD 段和数据寄存器

器件	LCD 寄存器数	
	段使能	数据
PIC16(L)F1933	2	8

LCDCON 寄存器 (寄存器 27-1) 控制 LCD 驱动模块的操作。LCDPS 寄存器 (寄存器 27-2) 配置 LCD 时钟源预分频器和波形类型：A 型或 B 型。LCDSEn 寄存器 (寄存器 27-5) 配置端口引脚的功能。

以下 LCDSEn 寄存器可用：

- LCDSE0 SE<7:0>
- LCDSE1 SE<15:8>

一旦为 LCD 面板初始化了模块，就可将 LCDDATAn 寄存器的各个位清零 / 置 1 以分别代表透明 / 不透明像素：

- LCDDATA0 SEG<7:0>COM0
- LCDDATA1 SEG<15:8>COM0
- LCDDATA3 SEG<7:0>COM1
- LCDDATA4 SEG<15:8>COM1
- LCDDATA6 SEG<7:0>COM2
- LCDDATA7 SEG<15:8>COM2
- LCDDATA9 SEG<7:0>COM3
- LCDDATA10 SEG<15:8>COM3

作为示例，在寄存器 27-6 中详细说明了 LCDDATAn。

一旦配置了模块，LCDCON 寄存器的 LCDEN 位就可用来使能或禁止 LCD 模块。通过清零 LCDCON 寄存器的 SLPEN 位可使 LCD 面板在休眠模式下仍能继续工作。

## 寄存器 27-1: LCDCON: 液晶显示 (LCD) 控制寄存器

R/W-0/0	R/W-0/0	R/C-0/0	U-0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1
LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>	
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

C = 只可清零位

bit 7

**LCDEN:** LCD 驱动使能位

1 = 使能 LCD 驱动模块

0 = 禁止 LCD 驱动模块

bit 6

**SLPEN:** 休眠模式下 LCD 驱动使能位

1 = 休眠模式下禁止 LCD 驱动模块

0 = 休眠模式下使能 LCD 驱动模块

bit 5

**WERR:** LCD 写失败错误位

1 = LCDPS 寄存器的 WA 位 = 0 时写入 LCDDATAN 寄存器 (必须用软件清零)

0 = 无 LCD 写错误

bit 4

**未实现:** 读为 0

bit 3-2

**CS<1:0>:** 时钟源选择位

00 = Fosc/256

01 = T1OSC (Timer1)

1x = LFINTOSC (31 kHz)

bit 1-0

**LMUX<1:0>:** 公共端选择位

LMUX<1:0>	复用	最大像素数	偏置
		PIC16(L)F1933	
00	静态 (COM0)	16	静态
01	1/2 (COM<1:0>)	32	1/2 或 1/3
10	1/3 (COM<2:0>)	48	1/2 或 1/3
11	1/4 (COM<3:0>)	60 <sup>(1)</sup>	1/3

**注 1:** 在这些器件上, COM3 和 SEG15 共用同一个引脚, 从而限制器件无法驱动 64 个像素。

# PIC16(L)F1933

寄存器 27-2: LCDPS: LCD 相位寄存器

R/W-0/0	R/W-0/0	R-0/0	R-0/0	R/W-0/0	R/W-0/0	R/W-1/1	R/W-1/1
WFT	BIASMD	LCDA	WA	LP<3:0>			
bit 7							bit 0

**图注:**

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	C = 只可清零位

- bit 7      **WFT:** 波形类型位  
 1 = B 型波形 (在每一帧边界改变相位)  
 0 = A 型波形 (在每种公共端复用类型的一帧内改变相位)
- bit 6      **BIASMD:** 偏置模式选择位  
当 LMUX<1:0> = 00 时:  
 0 = 静态偏置模式 (不要将该位置为 1)  
当 LMUX<1:0> = 01 时:  
 1 = 1/2 偏置模式  
 0 = 1/3 偏置模式  
当 LMUX<1:0> = 10 时:  
 1 = 1/2 偏置模式  
 0 = 1/3 偏置模式  
当 LMUX<1:0> = 11 时:  
 0 = 1/3 偏置模式 (不要将该位置为 1)
- bit 5      **LCDA:** LCD 工作状态位  
 1 = LCD 驱动模块正在工作  
 0 = LCD 驱动模块不在工作
- bit 4      **WA:** LCD 写允许状态位  
 1 = 允许写入 LCDDATAn 寄存器  
 0 = 禁止写入 LCDDATAn 寄存器
- bit 3-0    **LP<3:0>:** LCD 预分频比选择位  
 1111 = 1:16  
 1110 = 1:15  
 1101 = 1:14  
 1100 = 1:13  
 1011 = 1:12  
 1010 = 1:11  
 1001 = 1:10  
 1000 = 1:9  
 0111 = 1:8  
 0110 = 1:7  
 0101 = 1:6  
 0100 = 1:5  
 0011 = 1:4  
 0010 = 1:3  
 0001 = 1:2  
 0000 = 1:1

## 寄存器 27-3: LCDREF: LCD 参考电压控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0	U-0
LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—
bit 7							bit 0

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
u = 不变	x = 未知	-n/n = POR 和 BOR 时的值 / 所有其他复位时的值
1 = 置 1	0 = 清零	C = 只可清零位

- bit 7      **LCDIRE:** LCD 内部参考电压使能位  
 1 = 使能内部 LCD 参考电压, 并将其连接到内部对比度控制电路  
 0 = 禁止内部 LCD 参考电压
- bit 6      **LCDIRS:** LCD 内部参考电压源位  
如果 LCDIRE = 1:  
 0 = 内部 LCD 对比度控制由 VDD 提供电源  
 1 = 内部 LCD 对比度控制由 FVR 的 3.072V 输出电压提供电源  
如果 LCDIRE = 0:  
 未连接内部 LCD 对比度控制电路。禁止 LCD 带隙缓冲区。
- bit 5      **LCDIRI:** LCD 内部参考梯形电阻网络空闲使能位  
 当 LCD 参考梯形电阻网络处于功耗模式 B 时, 允许关闭内部 FVR 缓冲区  
 1 = 当 LCD 参考梯形电阻网络处于功耗模式 B 时, 禁止 LCD 内部 FVR 缓冲区  
 0 = LCD 内部 FVR 缓冲区忽略 LCD 参考梯形电阻网络功耗模式
- bit 4      **未实现:** 读为 0
- bit 3      **VLCD3PE:** VLCD3 引脚使能位  
 1 = VLCD3 引脚连接至内部偏置电压 LCDBIAS3<sup>(1)</sup>  
 0 = 未连接 VLCD3 引脚
- bit 2      **VLCD2PE:** VLCD2 引脚使能位  
 1 = VLCD2 引脚连接至内部偏置电压 LCDBIAS2<sup>(1)</sup>  
 0 = 未连接 VLCD2 引脚
- bit 1      **VLCD1PE:** VLCD1 引脚使能位  
 1 = VLCD1 引脚连接至内部偏置电压 LCDBIAS1<sup>(1)</sup>  
 0 = 未连接 VLCD1 引脚
- bit 0      **未实现:** 读为 0

注 1: TRISx 和 ANSELx 的正常引脚控制不受影响。

# PIC16(L)F1933

寄存器 27-4: LCDCST: LCD 对比度控制寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
—	—	—	—	—	LCDCST<2:0>		
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

C = 只可清零位

bit 7-3

未实现: 读为 0

bit 2-0

**LCDCST<2:0>:** LCD 对比度控制位

选择 LCD 对比度控制梯形电阻网络的阻值

位值 = 梯形电阻网络

000 = 最小阻值 (最大对比度)。梯形电阻网络被短路。

001 = 梯形电阻网络为最大阻值的 1/7

010 = 梯形电阻网络为最大阻值的 2/7

011 = 梯形电阻网络为最大阻值的 3/7

100 = 梯形电阻网络为最大阻值的 4/7

101 = 梯形电阻网络为最大阻值的 5/7

110 = 梯形电阻网络为最大阻值的 6/7

111 = 梯形电阻网络为最大阻值 (最小对比度)。



## 寄存器 27-5: LCDSEn: LCD 段使能寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
SEn	SEn	SEn	SEn	SEn	SEn	SEn	SEn
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

**SEn:** 段使能位

1 = 使能引脚的段功能

0 = 使能引脚的 I/O 功能

## 寄存器 27-6: LCDDATAn: LCD 数据寄存器

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy	SEGx-COMy
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-0

**SEGx-COMy:** 像素点亮位

1 = 点亮像素 (不透明)

0 = 不点亮像素 (透明)

# PIC16(L)F1933

## 27.2 LCD 时钟源选择

LCD 模块有 3 种可用的时钟源：

- $F_{osc}/256$
- T1OSC
- LFINTOSC

第一个时钟源是系统时钟的 256 分频 ( $F_{osc}/256$ )。当系统时钟为 8 MHz 时，选择该分频比将提供约 1 kHz 的输出。分频比不可编程。而是使用 LCDPS 寄存器的 LCD 预分频比位 LP<3:0> 设置 LCD 帧时钟速率。

第二个时钟源是 T1OSC。Timer1 振荡器使用 32.768 kHz 晶振时，它也能提供约 1 kHz 的输出。要使用 Timer1 作为时钟源，应将 T1CON 寄存器的 T1OSCN 位置 1。

第三个时钟源是 31 kHz LFINTOSC，它可提供约 1 kHz 的输出。

当处理器处于休眠模式时，第二个和第三个时钟源用于维持运行 LCD。

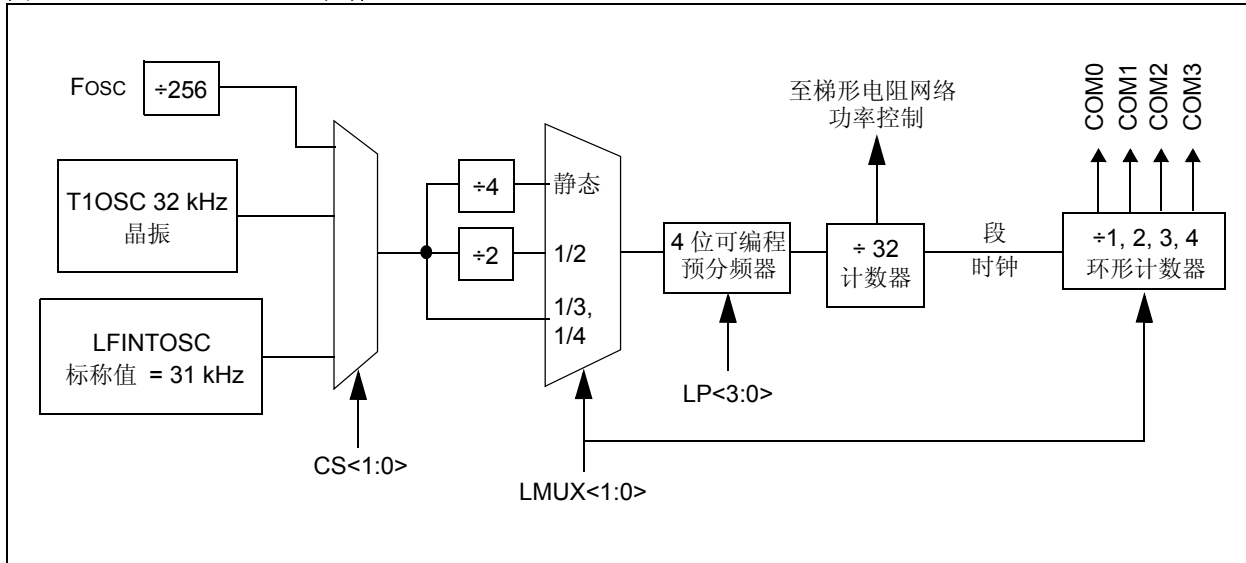
使用 LCDCON 寄存器的 CS<1:0> 位可以选择任何这些时钟源。

### 27.2.1 LCD 预分频器

4 位计数器可以用作 LCD 时钟的预分频器。不可直接读写该预分频器；其值由 LCDPS 寄存器的 LP<3:0> 位设置，该值决定了预分频器的分配和预分频比。

预分频值的可选范围是 1:1 至 1:16。

图 27-2: LCD 时钟产生



## 27.3 LCD 偏置电压生成

LCD 模块可配置为以下三种偏置类型之一：

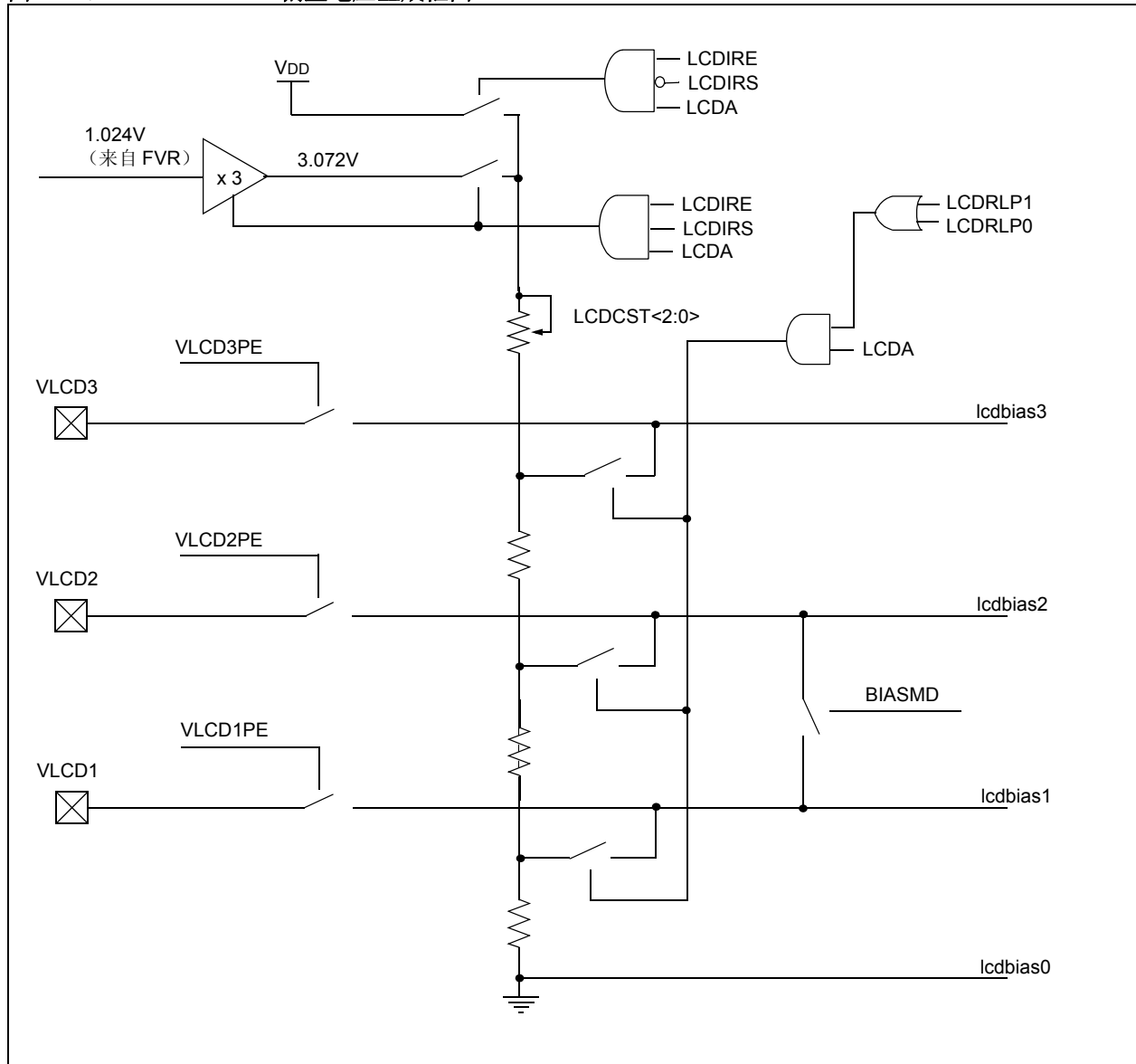
- 静态偏置（2 个电压级别：Vss 和 VLCD）
- 1/2 偏置（3 个电压级别：Vss、1/2 VLCD 和 VLCD）
- 1/3 偏置（4 个电压级别：Vss、1/3 VLCD、2/3 VLCD 和 VLCD）

表 27-2: LCD 偏置电压

	静态偏置	1/2 偏置	1/3 偏置
LCD 偏置 0	Vss	Vss	Vss
LCD 偏置 1	—	1/2 VDD	1/3 VDD
LCD 偏置 2	—	1/2 VDD	2/3 VDD
LCD 偏置 3	VLCD3	VLCD3	VLCD3

因此不要求用户放置外部组件并最多使用 3 个引脚来生成偏置电压，PIC16(L)F1933 内部提供内部对比度控制和内部参考梯形电阻网络。这两种功能可以与外部 VLCD<3:1> 引脚结合使用，以提供最大的灵活性。请参见图 27-3。

图 27-3: LCD 偏置电压生成框图



## 27.4 LCD 偏置内部参考梯形电阻网络

可以使用内部参考梯形电阻网络将LCD偏置电压二或三等分，并将电压提供给LCD段引脚。要创建这种电压，参考梯形电阻网络应由3个阻值匹配的电阻组成。请参见图27-3。

### 27.4.1 偏置模式下的相互影响

在1/2偏置模式（BIASMD = 1）下时，梯形的中间电阻应短路，以便只产生两个电压。由于去掉了一个电阻，所以此模式下的梯形电阻网络的电流消耗较高。

**表 27-3: LCD 内部梯形电阻功耗模式 (1/3 偏置)**

功耗模式	整个梯形电阻网络的标称阻值	I <sub>DD</sub> 标称值
低	3 MΩ	1 μA
中	300 kΩ	10 μA
高	30 kΩ	100 μA

### 27.4.2 功耗模式

内部参考梯形电阻网络可以工作在三种功耗模式下，从而允许用户在特定应用中权衡LCD对比度和功耗。LCD显示屏越大，物理LCD段上的电容也就越高，因此需要更大的电流来维持相同对比度。

3种可用的功耗模式为：LP、MP和HP。针对想要使用外部梯形电阻网络或者最小化功耗的应用，也可关闭内部参考梯形电阻网络。禁用内部参考梯形电阻网络会导致所有梯形电阻断开，以允许由外部电压来提供电源。

LCD模块停止工作（LCDA = 0）时，内部参考梯形电阻网络将关闭。

## 27.4.3 自动功耗模式切换

因为 LCD 段从电气角度来说只是一个电容，故仅在电压改变时才消耗电流。为了降低器件总电流，LCD 内部参考梯形电阻网络在电压变化的过程中可工作在不同模式下，具体由 LCDRL 寄存器（寄存器 27-7）控制。

LCDRL 寄存器允许在被指定为 A 和 B 的两种功耗模式之间切换。A 功耗模式的有效时间可编程设定，从 LCD 段转换时开始。B 功耗模式在段或者公共端再次变化前的剩余时间内有效。LRLAT<2:0> 位选择 A 功耗模式有效时间（如果有）。请参见图 27-4。

要实现此操作，可以使用该 5 位预分频器选择功耗模式，用于将 32 kHz 时钟分频以得到 LCD 控制器 1 kHz 基准频率。

图 27-4: LCD 内部参考梯形电阻网络功耗模式切换时序图——A 型

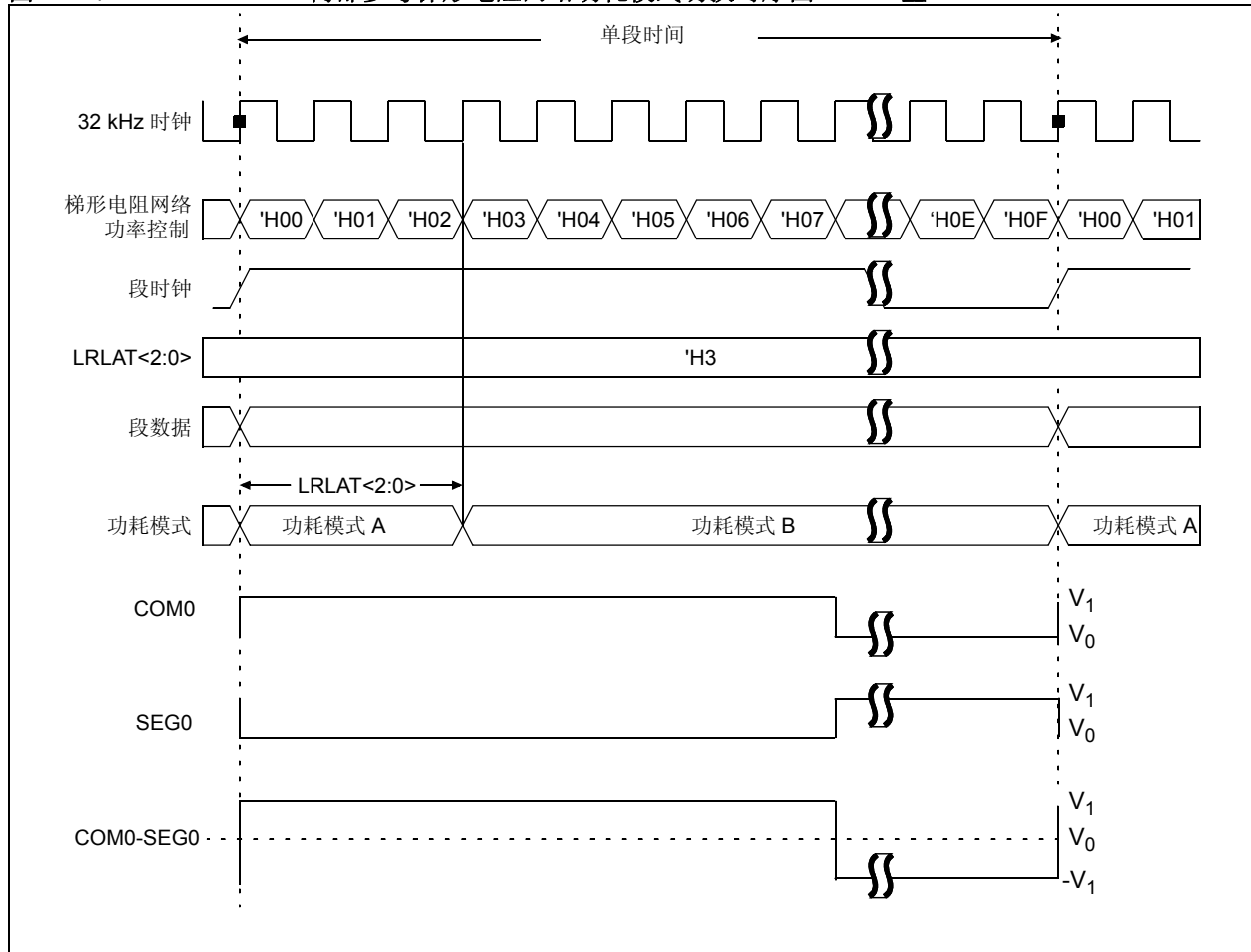


图 27-5: LCD 内部参考梯形电阻网络功耗模式切换时序图——A 型波形 (1/2 复用, 1/2 偏置驱动)

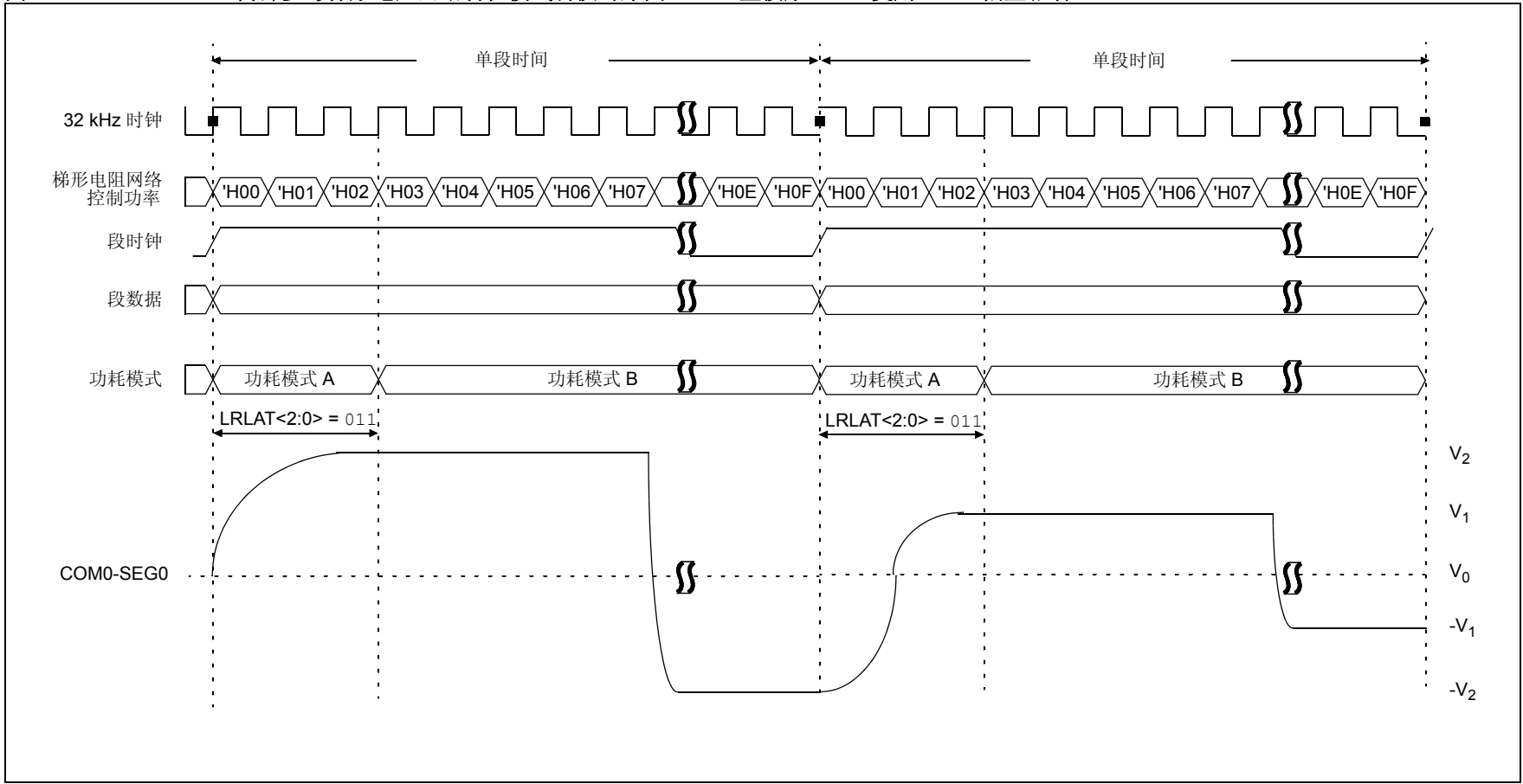
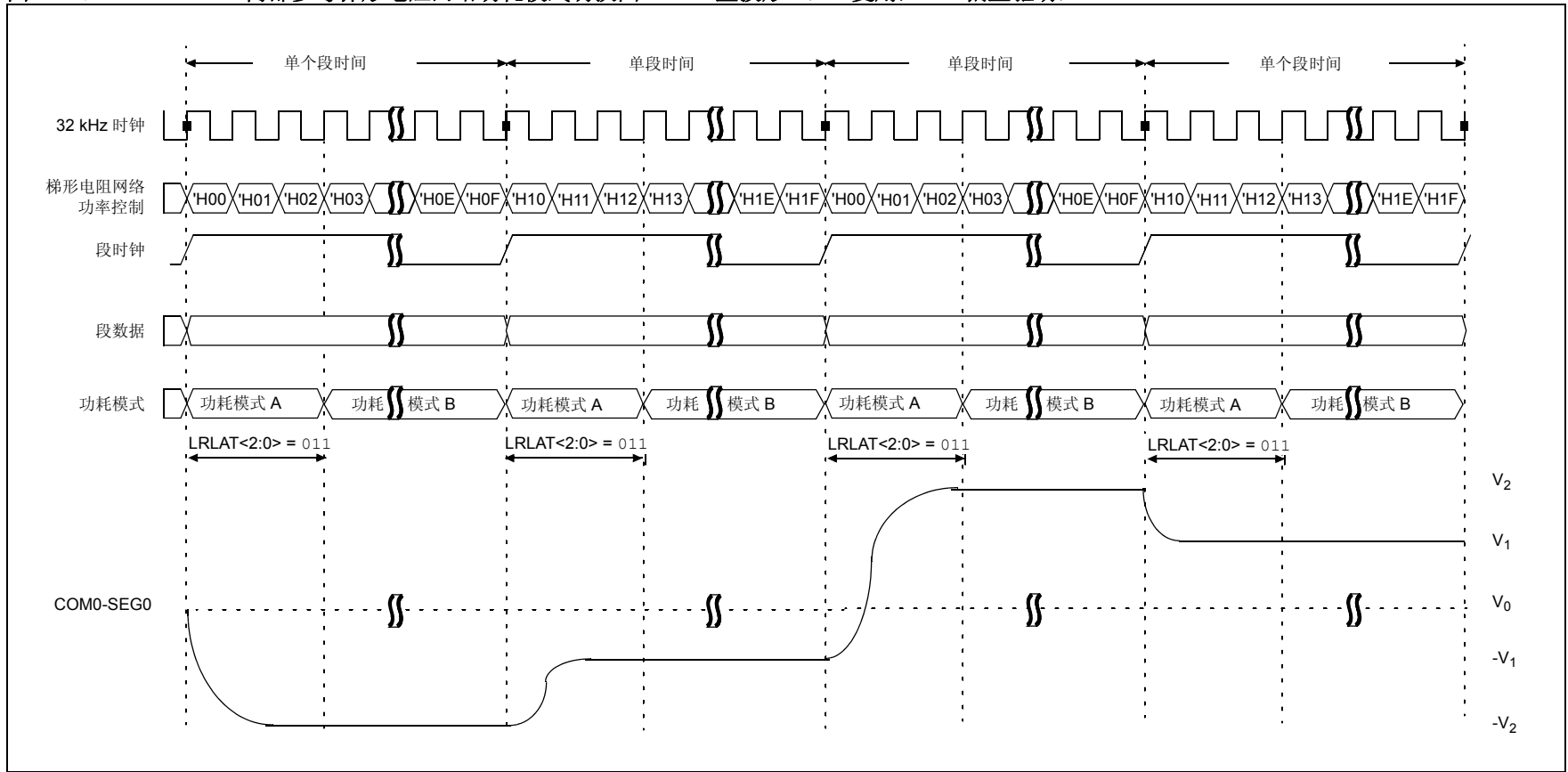


图 27-6: LCD 内部参考梯形电阻网络功耗模式切换图——B 型波形 (1/2 复用, 1/2 偏置驱动)



# PIC16(L)F1933

## 寄存器 27-7: LCDRL: LCD 参考梯形电阻网络控制寄存器

R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0	R/W-0/0
LRLAP<1:0>		LRLBP<1:0>		—	LRLAT<2:0>		
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

u = 不变

x = 未知

-n/n = POR 和 BOR 时的值 / 所有其他复位时的值

1 = 置 1

0 = 清零

bit 7-6 **LRLAP<1:0>**: LCD 参考梯形电阻网络 A 时间功率控制位  
时间间隔 A 期间 (见图 27-4):

- 00 = 内部 LCD 参考梯形电阻网络掉电并断开连接
- 01 = 内部 LCD 参考梯形电阻网络工作在低功耗模式下
- 10 = 内部 LCD 参考梯形电阻网络工作在中等功耗模式下
- 11 = 内部 LCD 参考梯形电阻网络工作在高功耗模式下

bit 5-4 **LRLBP<1:0>**: LCD 参考梯形电阻网络 B 时间功率控制位  
时间间隔 B 期间 (见图 27-4):

- 00 = 内部 LCD 参考梯形电阻网络掉电并断开连接
- 01 = 内部 LCD 参考梯形电阻网络工作在低功耗模式下
- 10 = 内部 LCD 参考梯形电阻网络工作在中等功耗模式下
- 11 = 内部 LCD 参考梯形电阻网络工作在高功耗模式下

bit 3 **未实现**: 读为 0

bit 2-0 **LRLAT<2:0>**: LCD 参考梯形电阻网络 A 时间间隔控制位  
设置 32 kHz 时钟的脉冲数, 表示 A 时间间隔功耗模式有效的的时间

对于 A 型波形 (WFT = 0):

- 000 = 内部 LCD 参考梯形电阻网络始终处于 B 功耗模式
- 001 = 内部 LCD 参考梯形电阻网络在 1 个时钟周期内处于 A 功耗模式而在 15 个时钟周期内处于 B 功耗模式
- 010 = 内部 LCD 参考梯形电阻网络在 2 个时钟周期内处于 A 功耗模式而在 14 个时钟周期内处于 B 功耗模式
- 011 = 内部 LCD 参考梯形电阻网络在 3 个时钟周期内处于 A 功耗模式而在 13 个时钟周期内处于 B 功耗模式
- 100 = 内部 LCD 参考梯形电阻网络在 4 个时钟周期内处于 A 功耗模式而在 12 个时钟周期内处于 B 功耗模式
- 101 = 内部 LCD 参考梯形电阻网络在 5 个时钟周期内处于 A 功耗模式而在 11 个时钟周期内处于 B 功耗模式
- 110 = 内部 LCD 参考梯形电阻网络在 6 个时钟周期内处于 A 功耗模式而在 10 个时钟周期内处于 B 功耗模式
- 111 = 内部 LCD 参考梯形电阻网络在 7 个时钟周期内处于 A 功耗模式而在 9 个时钟周期内处于 B 功耗模式

对于 B 型波形 (WFT = 1):

- 000 = 内部 LCD 参考梯形电阻网络始终处于 B 功耗模式
- 001 = 内部 LCD 参考梯形电阻网络在 1 个时钟周期内处于 A 功耗模式而在 31 个时钟周期内处于 B 功耗模式
- 010 = 内部 LCD 参考梯形电阻网络在 2 个时钟周期内处于 A 功耗模式而在 30 个时钟周期内处于 B 功耗模式
- 011 = 内部 LCD 参考梯形电阻网络在 3 个时钟周期内处于 A 功耗模式而在 29 个时钟周期内处于 B 功耗模式
- 100 = 内部 LCD 参考梯形电阻网络在 4 个时钟周期内处于 A 功耗模式而在 28 个时钟周期内处于 B 功耗模式
- 101 = 内部 LCD 参考梯形电阻网络在 5 个时钟周期内处于 A 功耗模式而在 27 个时钟周期内处于 B 功耗模式
- 110 = 内部 LCD 参考梯形电阻网络在 6 个时钟周期内处于 A 功耗模式而在 26 个时钟周期内处于 B 功耗模式
- 111 = 内部 LCD 参考梯形电阻网络在 7 个时钟周期内处于 A 功耗模式而在 25 个时钟周期内处于 B 功耗模式



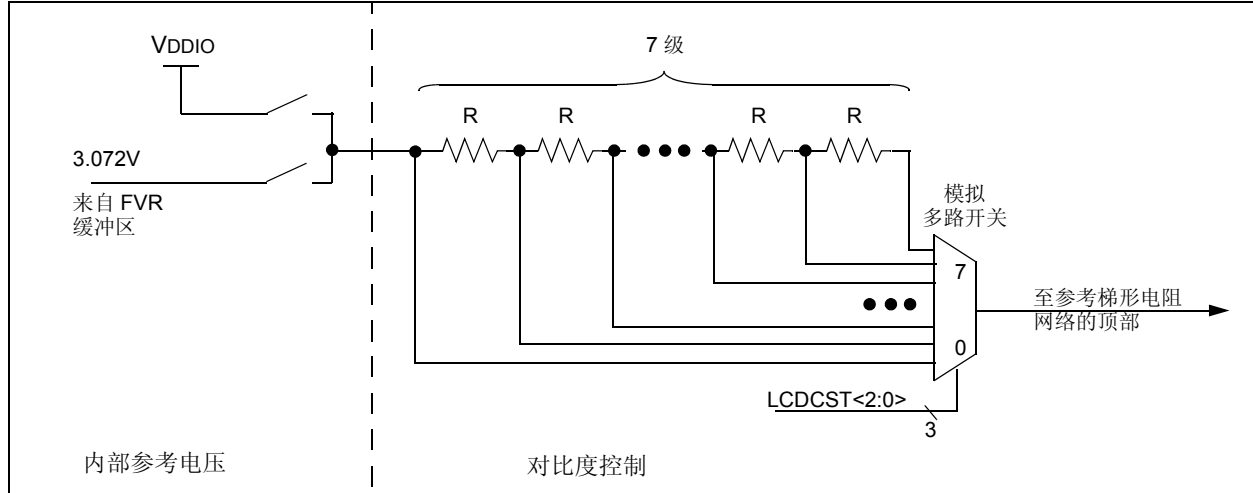
## 27.4.4 对比度控制

LCD 对比度控制电路由一个 7 抽头的梯形电阻组成，由 LCDCST 位控制。请参见图 27-7。

LCDCST = 111 时，对比度控制电路用于将单信号源的输出电压减少约 10%。

LCD 模块停止工作 (LCDA = 0) 时，对比度控制梯形电阻网络将关闭 (开路)。

图 27-7: 内部参考电压和对比度控制框图



## 27.4.5 内部参考电压

在固件控制下，可以使能 LCD 偏置电压的内部参考电压。使能时，此电压源可以是 VDDIO 或主固定参考电压的 3 倍 (3.072V)。未选择内部参考电压时，LCD 对比度控制电路被禁止，LCD 偏置电压必须由外部提供。

LCD 模块停止工作 (LCDA = 0) 时，内部参考电压将被关闭。

内部参考电压使能并选择了固定参考电压时，LCDIRI 位与 LCD 参考梯形电阻网络自动功耗模式切换相结合可用来降低功耗。LCDIRI = 1 且 LCD 参考梯形电阻网络处于功耗模式 B 时禁止 LCD 内部 FVR 缓冲区。

## 27.4.6 VLCD<3:1> 引脚

VLCD<3:1> 引脚提供使用外部 LCD 偏置网络的能力，而不依靠内部梯形电阻网络。使用 VLCD<3:1> 引脚并不阻碍内部梯形电阻网络的使用。每个 VLCD 引脚在 LCDREF 寄存器 (寄存器 27-3) 中具有独立的控制，能允许接入任一或所有 LCD 偏置信号。此架构为各种不同的应用提供了最大灵活性。

例如，可以使用 VLCD<3:1> 引脚向内部参考梯形电阻网络添加电容，以增强驱动能力。

对于内部对比度控制能力不足的应用，该固件可选择仅使能 VLCD3 引脚，从而允许外部对比度控制电路使用内部参考电压分压器。

**注：** 需要时，LCD 模块会自动启动固定参考电压。

# PIC16(L)F1933

## 27.5 LCD 复用类型

LCD 驱动模块可以配置为以下四种复用类型之一：

- 静态（仅使用 COM0）
- 1/2 复用（使用 COM<1:0>）
- 1/3 复用（使用 COM<2:0>）
- 1/4 复用（使用 COM<3:0>）

LCDCON 寄存器的 LMUX<1:0> 位的位置决定了要使用哪个 LCD 公共端引脚（详情请参见表 27-4）。

如果该引脚为数字 I/O，那么相应的 TRIS 位控制数据方向。如果该引脚为 COM 驱动，那么该引脚的 TRIS 设置将不起作用。

表 27-4: 公共端引脚使用情况

复用	LMUX <1:0>	COM3	COM2	COM1
静态	00	未使用	未使用	未使用
1/2	01	未使用	未使用	有效
1/3	10	未使用	有效	有效
1/4	11	有效	有效	有效

## 27.6 段使能

LCDSEn 寄存器用于选择每个段引脚的引脚功能。引脚功能选择可使每个引脚作为 LCD 段驱动或者负责提供引脚的备用功能。要把引脚配置为段引脚，LCDSEn 寄存器中的相应位必须设置为 1。

如果该引脚为数字 I/O，那么相应的 TRIS 位控制数据方向。LCDSEn 寄存器的任何位置 1 都会覆盖相应 TRIS 寄存器的相应位的设置。

**注：** 上电复位时，这些引脚被配置为常规 I/O，而不是 LCD 引脚。

## 27.7 像素控制

LCDDATAx 寄存器中的位用于定义每个像素的状态。每一位只定义一个像素。

寄存器 27-6 显示了 LCDDATAx 寄存器中的每个位与对应公共端和段信号的相互关系。

没有用于显示的 LCD 像素地址单元可被用作通用 RAM。

## 27.8 LCD 帧频率

COM 和 SEG 输出改变的速率称为 LCD 帧频率。

表 27-5: 帧频率公式

复用	帧频率 =
静态	时钟源 / (4 x 1 x (LPD 预分频比) x 32)
1/2	时钟源 / (2 x 2 x (LPD 预分频比) x 32)
1/3	时钟源 / (1 x 3 x (LPD 预分频比) x 32)
1/4	时钟源 / (1 x 4 x (LPD 预分频比) x 32)

**注：** 时钟源为 Fosc/256、T1OSC 或 LFINTOSC。

表 27-6: 使用 8 MHz 的 Fosc、工作在 32.768 kHz 的 TIMER1 或 LFINTOSC 时的近似帧频率（单位为 Hz）

LP<3:0>	静态	1/2	1/3	1/4
2	122	122	162	122
3	81	81	108	81
4	61	61	81	61
5	49	49	65	49
6	41	41	54	41
7	35	35	47	35

表 27-7: LCD 段映射工作表

LCD 功能	COM0		COM1		COM2		COM3	
	LCDDATAx 地址	LCD 段	LCDDATAx 地址	LCD 段	LCDDATAx 地址	LCD 段	LCDDATAx 地址	LCD 段
SEG0	LCDDATA0, 0		LCDDATA3, 0		LCDDATA6, 0		LCDDATA9, 0	
SEG1	LCDDATA0, 1		LCDDATA3, 1		LCDDATA6, 1		LCDDATA9, 1	
SEG2	LCDDATA0, 2		LCDDATA3, 2		LCDDATA6, 2		LCDDATA9, 2	
SEG3	LCDDATA0, 3		LCDDATA3, 3		LCDDATA6, 3		LCDDATA9, 3	
SEG4	LCDDATA0, 4		LCDDATA3, 4		LCDDATA6, 4		LCDDATA9, 4	
SEG5	LCDDATA0, 5		LCDDATA3, 5		LCDDATA6, 5		LCDDATA9, 5	
SEG6	LCDDATA0, 6		LCDDATA3, 6		LCDDATA6, 6		LCDDATA9, 6	
SEG7	LCDDATA0, 7		LCDDATA3, 7		LCDDATA6, 7		LCDDATA9, 7	
SEG8	LCDDATA1, 0		LCDDATA4, 0		LCDDATA7, 0		LCDDATA10, 0	
SEG9	LCDDATA1, 1		LCDDATA4, 1		LCDDATA7, 1		LCDDATA10, 1	
SEG10	LCDDATA1, 2		LCDDATA4, 2		LCDDATA7, 2		LCDDATA10, 2	
SEG11	LCDDATA1, 3		LCDDATA4, 3		LCDDATA7, 3		LCDDATA10, 3	
SEG12	LCDDATA1, 4		LCDDATA4, 4		LCDDATA7, 4		LCDDATA10, 4	
SEG13	LCDDATA1, 5		LCDDATA4, 5		LCDDATA7, 5		LCDDATA10, 5	
SEG14	LCDDATA1, 6		LCDDATA4, 6		LCDDATA7, 6		LCDDATA10, 6	
SEG15	LCDDATA1, 7		LCDDATA4, 7		LCDDATA7, 7		LCDDATA10, 7	
SEG16	LCDDATA2, 0		LCDDATA5, 0		LCDDATA8, 0		LCDDATA11, 0	
SEG17	LCDDATA2, 1		LCDDATA5, 1		LCDDATA8, 1		LCDDATA11, 1	
SEG18	LCDDATA2, 2		LCDDATA5, 2		LCDDATA8, 2		LCDDATA11, 2	
SEG19	LCDDATA2, 3		LCDDATA5, 3		LCDDATA8, 3		LCDDATA11, 3	
SEG20	LCDDATA2, 4		LCDDATA5, 4		LCDDATA8, 4		LCDDATA11, 4	
SEG21	LCDDATA2, 5		LCDDATA5, 5		LCDDATA8, 5		LCDDATA11, 5	
SEG22	LCDDATA2, 6		LCDDATA5, 6		LCDDATA8, 6		LCDDATA11, 6	
SEG23	LCDDATA2, 7		LCDDATA5, 7		LCDDATA8, 7		LCDDATA11, 7	

# PIC16(L)F1933

## 27.9 LCD 波形产生

产生的LCD波形应使得不透明像素上的净交流电压为最大值，而透明像素上的净交流电压为最小值。任何像素上的净直流电压应该为零。

COM 信号表示每个公共端的时间片，而 SEG 中包含像素数据。

像素信号 (COM-SEG) 中将不包含直流分量，并且只可取两个 RMS 值中的一个。高 RMS 值会产生不透明像素，而低 RMS 值会产生透明像素。

随着公共端数量的增加，两个 RMS 值间的判别比逐渐减小。这个判别比表示显示器可具有的最大对比度。

可以用两种波形驱动 LCD: A 型和 B 型。在 A 型波形中，对于每种公共端类型相位在一帧的中间改变，然而在 B 型波形中，相位在每个帧边界上改变。这样，A 型波形在单帧内的净电压为 0 Vdc，而 B 型波形则需要两个帧才能保证净电压为零。

- 注 1:** 如果要通过禁止 LCD 休眠模式 (LCDCON<SLPEN> 为 1) 进入休眠状态，则必须格外小心，因为只有当所有像素上的 Vdc 为 0 时才可执行休眠。
- 2:** 当 LCD 时钟源为 Fosc/256，如果执行休眠，不管 LCDCON<SLPEN> 设置如何，LCD 都将立即进入休眠状态。因此，当执行休眠时应注意查看所有像素上的 Vdc 是否为 0。

图 27-8 至图 27-18 所示为 A 型和 B 型波形在静态、1/2 复用、1/3 复用和 1/4 复用驱动时的波形图。

图 27-8: 静态驱动时的 A/B 型波形

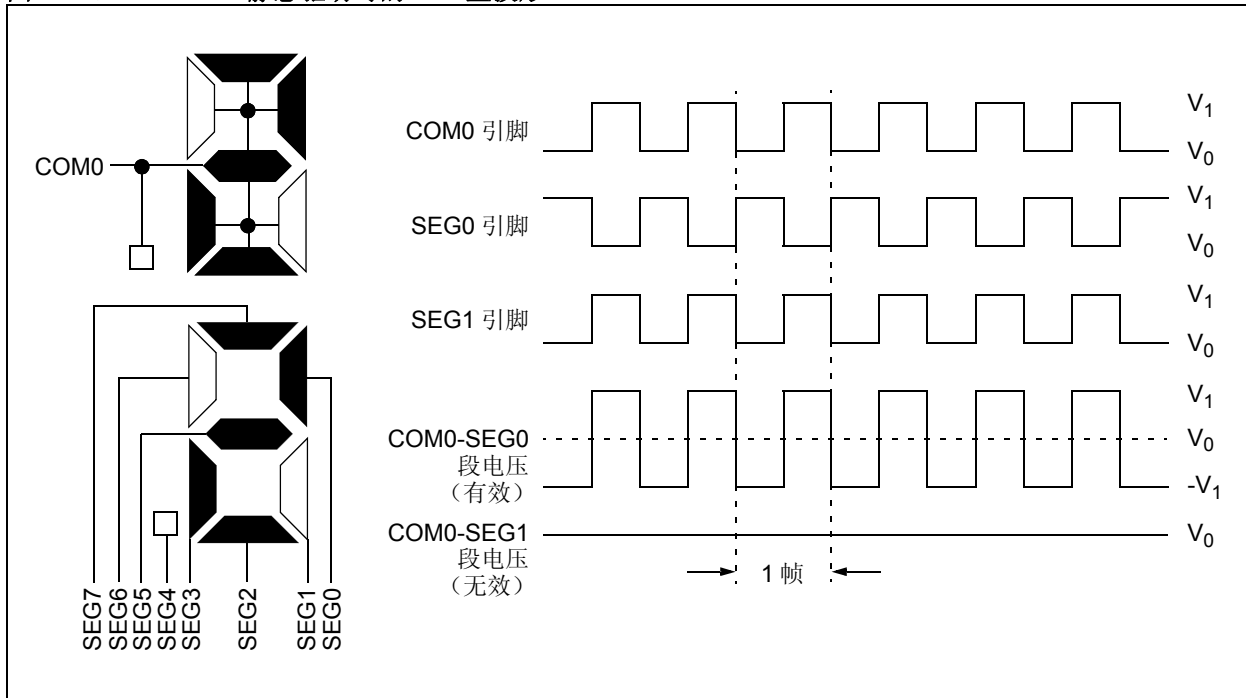
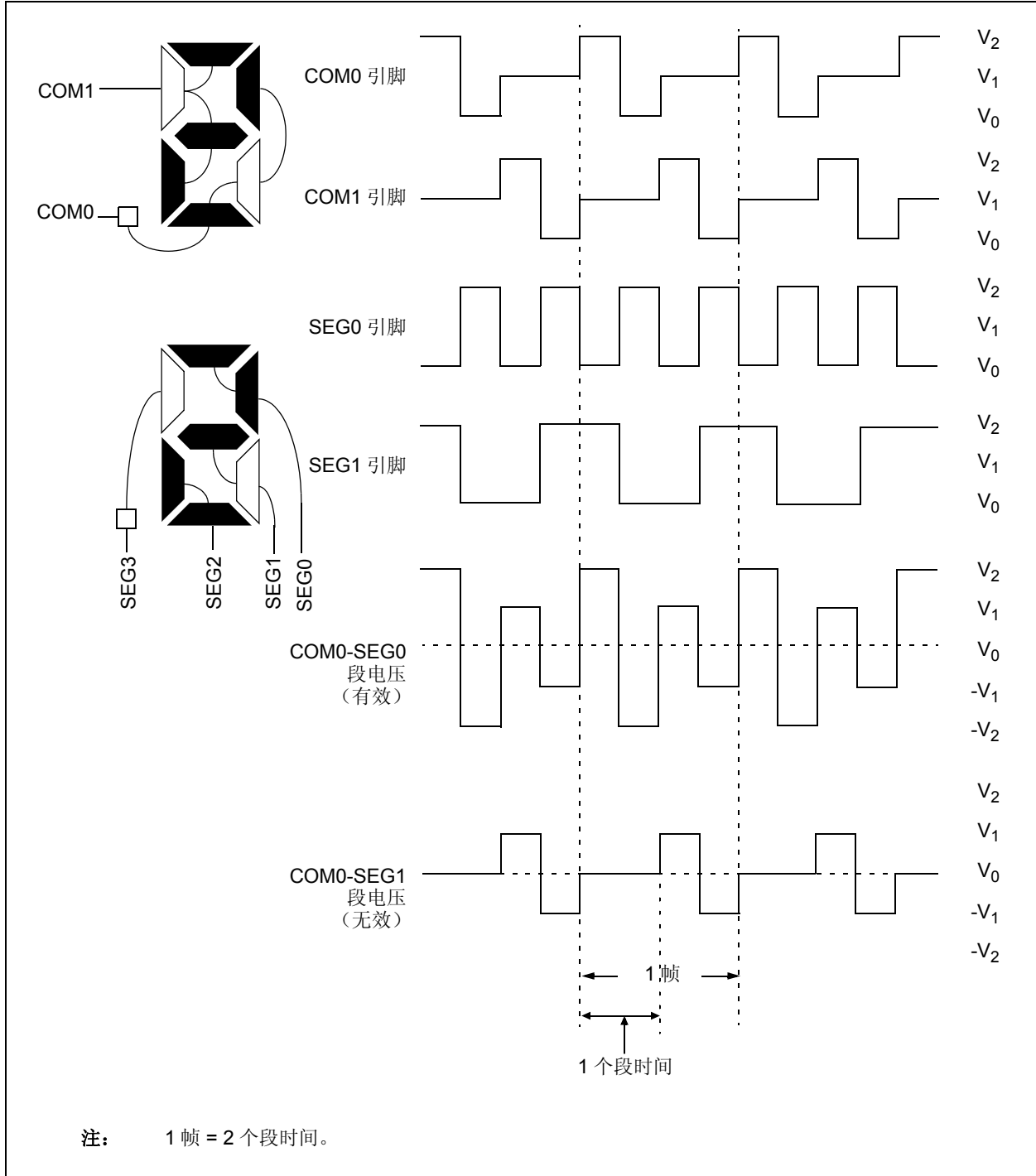


图 27-9: 1/2 复用、1/2 偏置驱动时的 A 型波形



# PIC16(L)F1933

图 27-10: 1/2 复用、1/2 偏置驱动时的 B 型波形

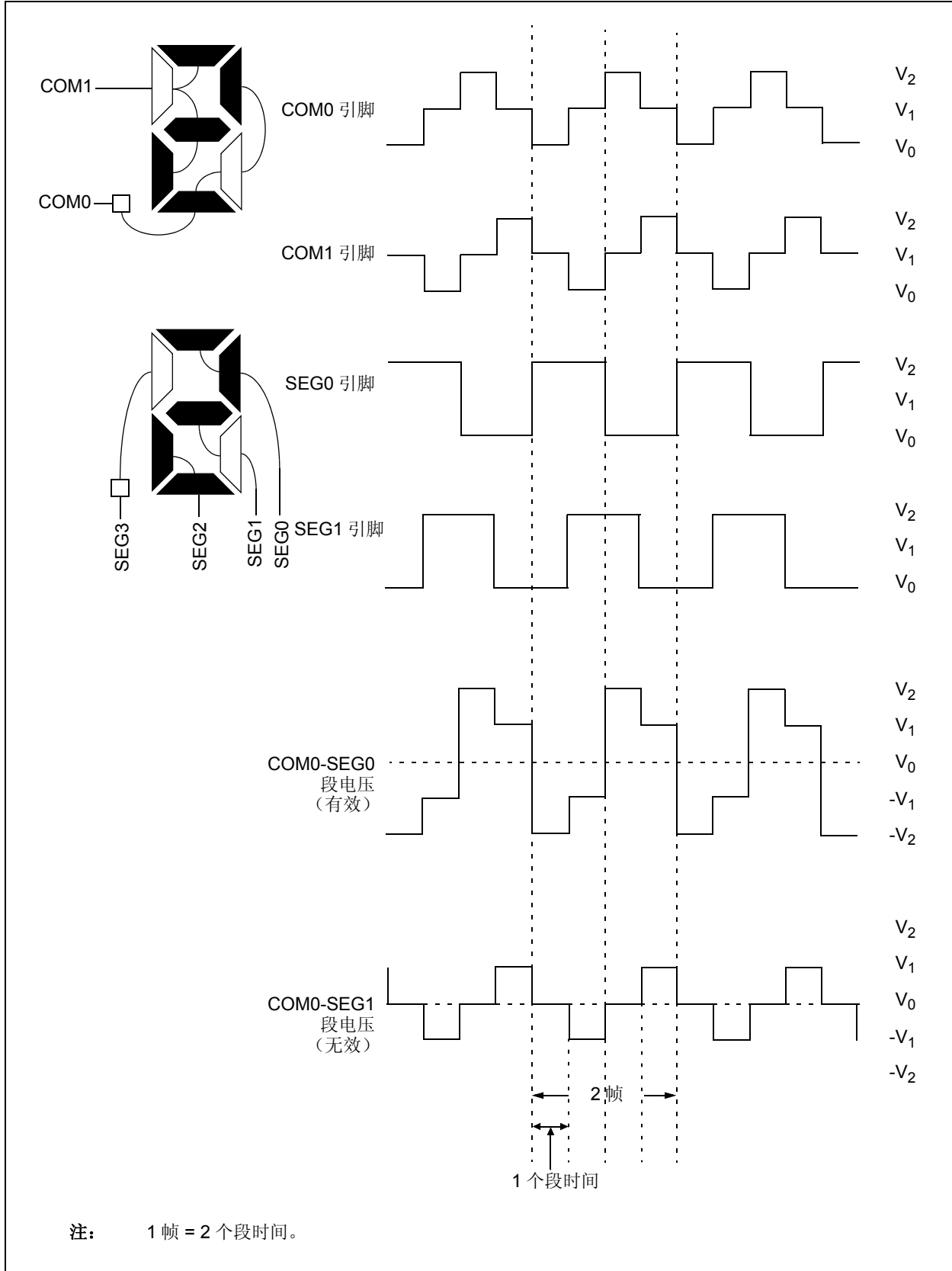
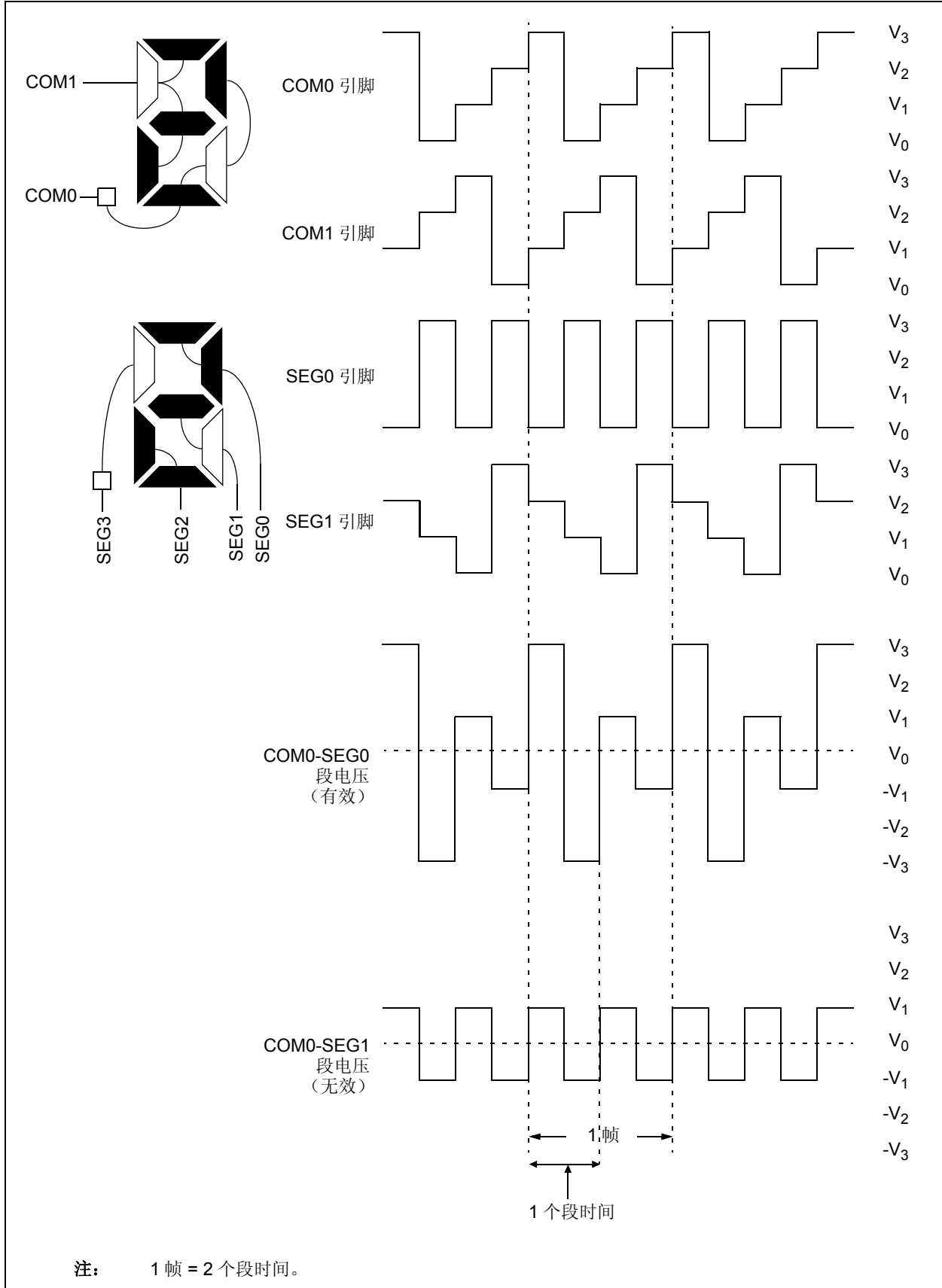


图 27-11: 1/2 复用、1/3 偏置驱动时的 A 型波形



# PIC16(L)F1933

图 27-12: 1/2 复用、1/3 偏置驱动时的 B 型波形

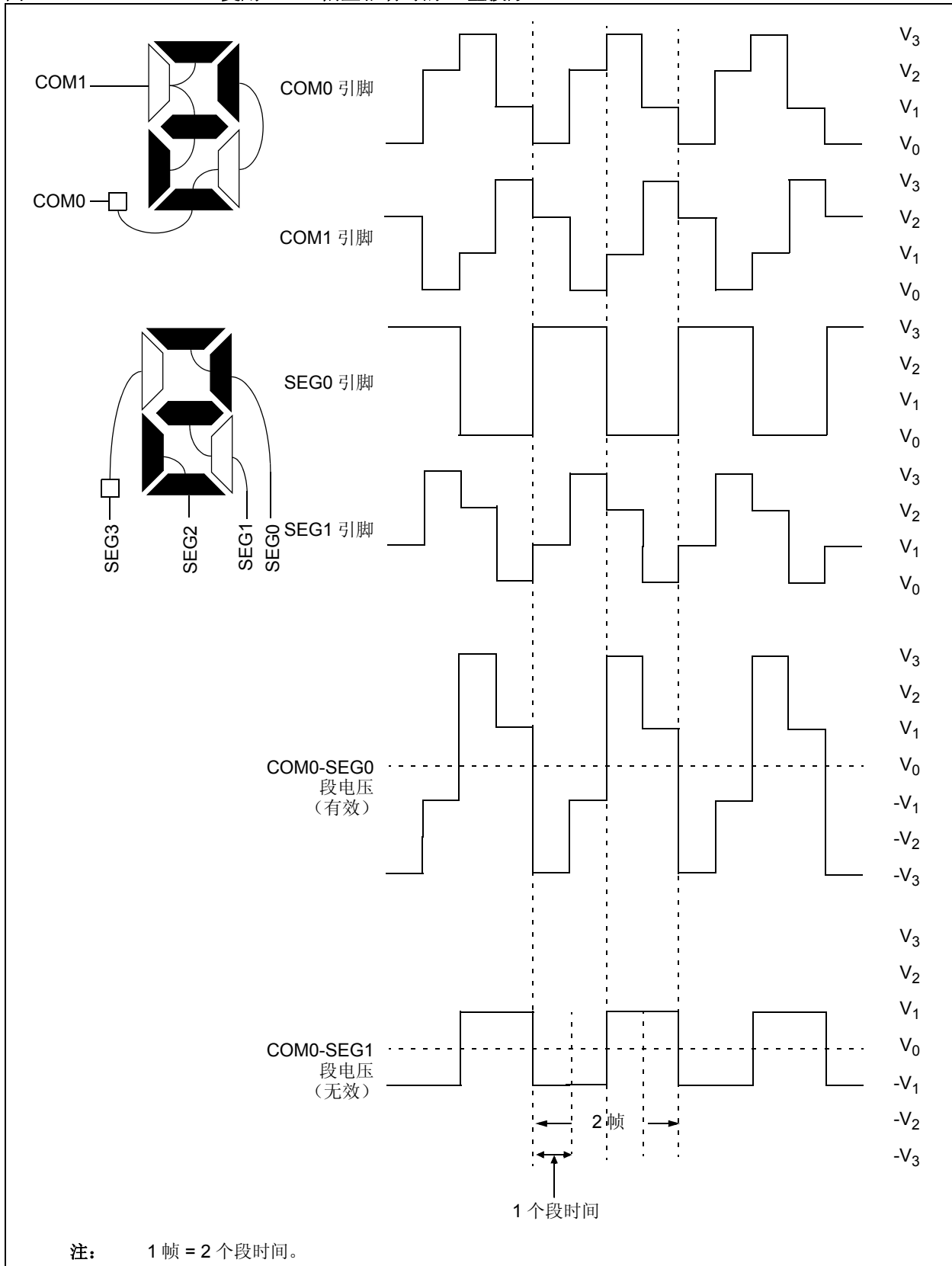
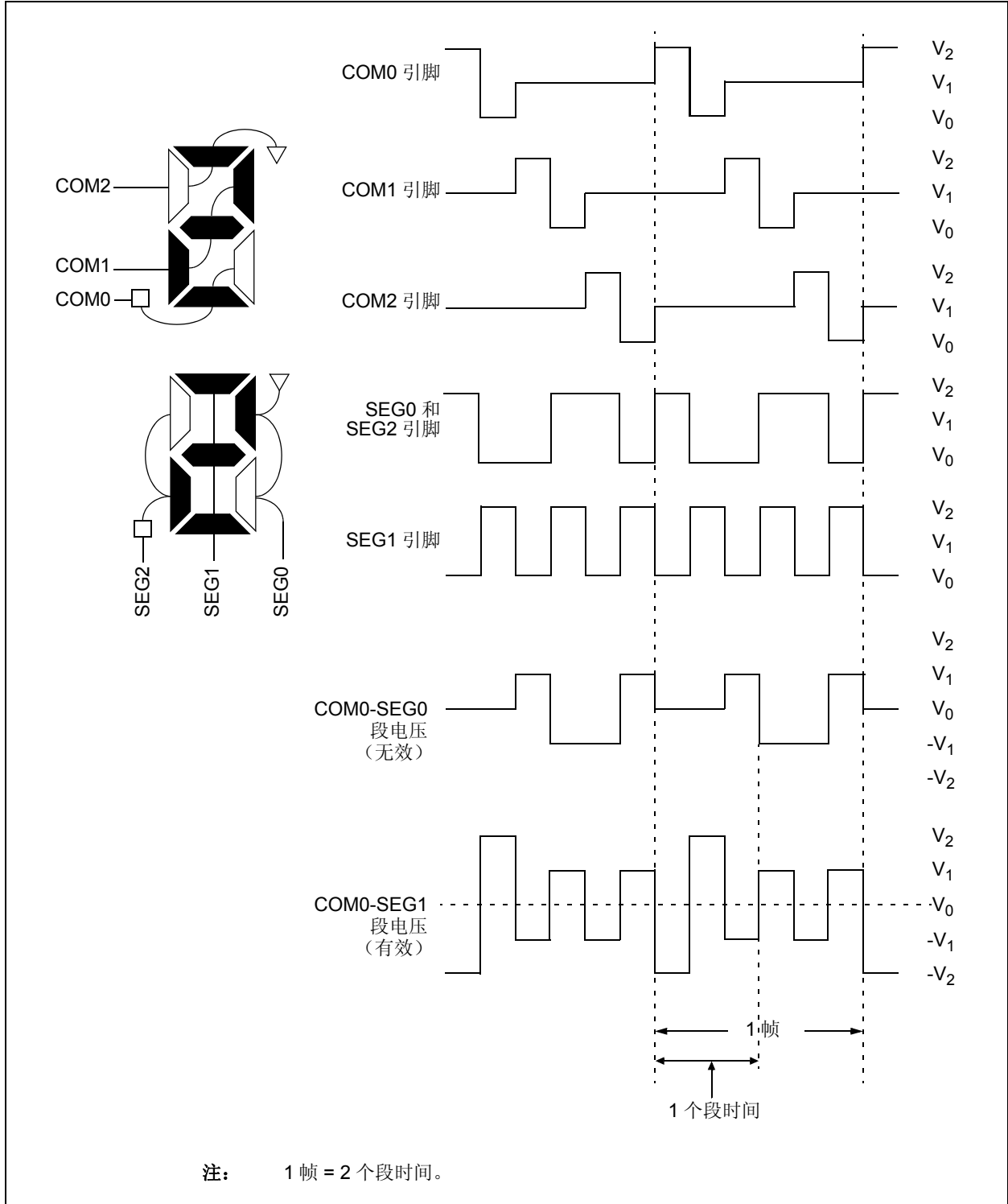




图 27-13: 1/3 复用、1/2 偏置驱动时的 A 型波形



# PIC16(L)F1933

图 27-14: 1/3 复用、1/2 偏置驱动时的 B 型波形

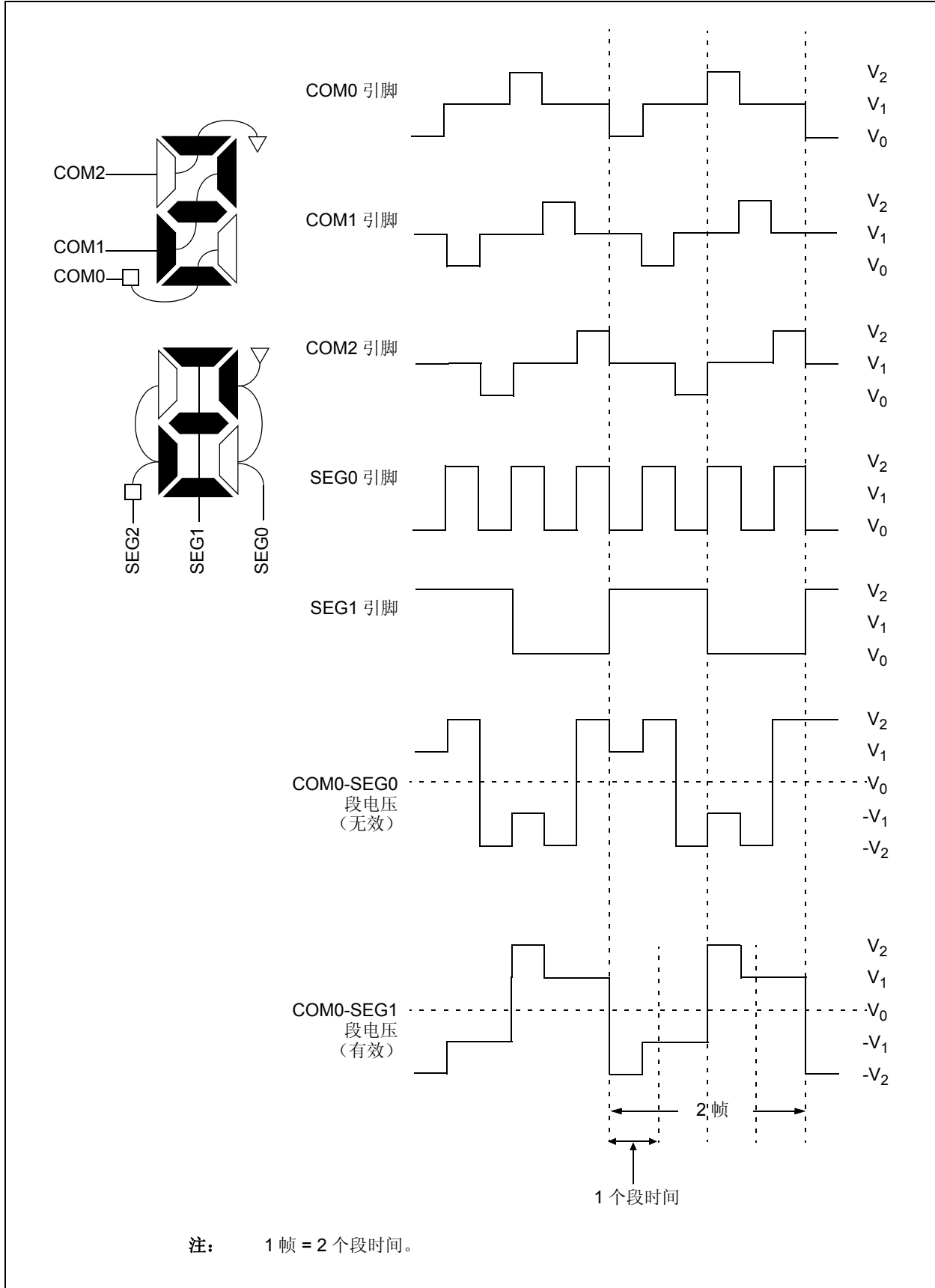
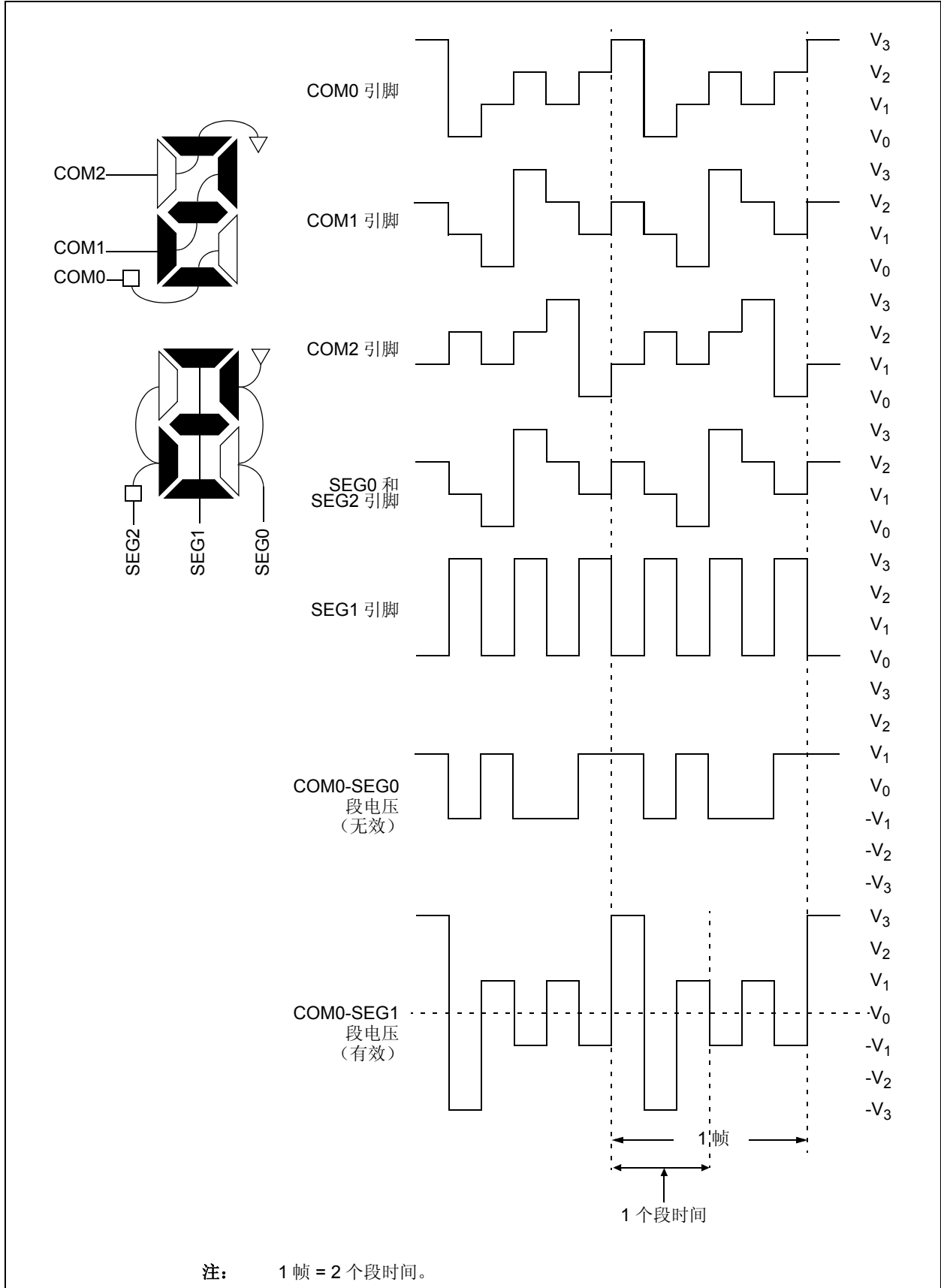


图 27-15: 1/3 复用、1/3 偏置驱动时的 A 型波形



# PIC16(L)F1933

图 27-16: 1/3 复用、1/3 偏置驱动时的 B 型波形

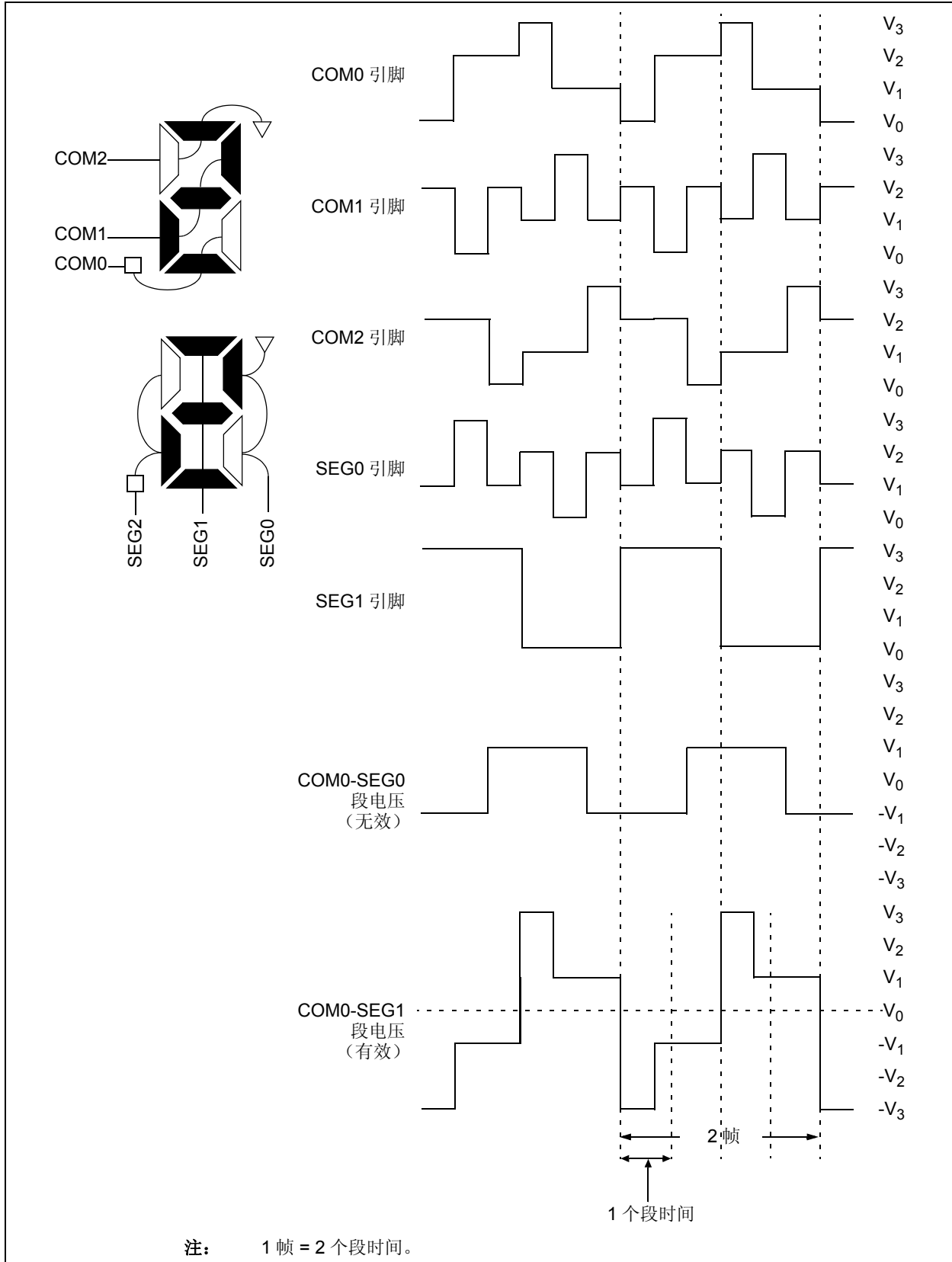
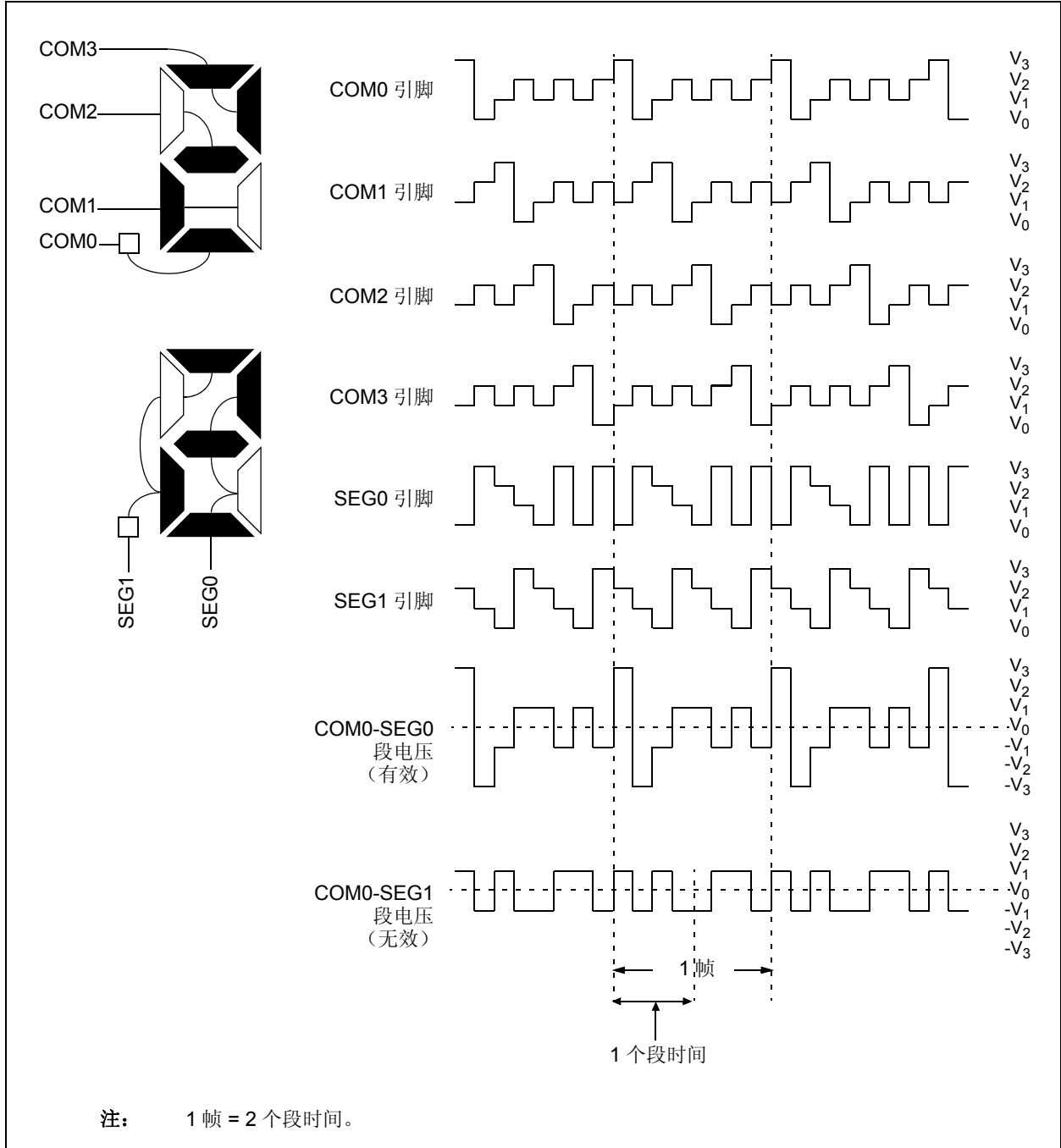
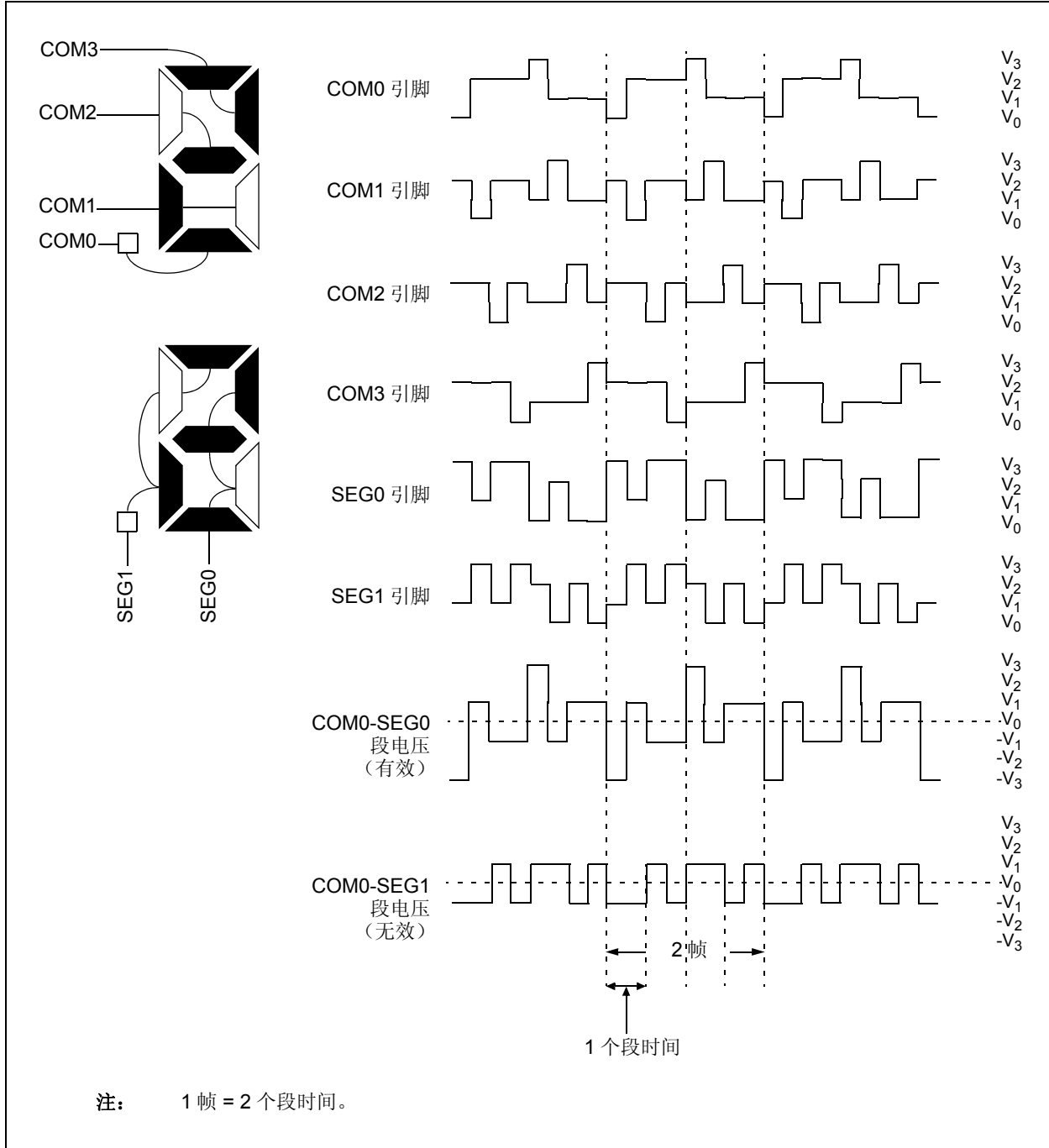


图 27-17: 1/4 复用、1/3 偏置驱动时的 A 型波形



# PIC16(L)F1933

图 27-18: 1/4 复用、1/3 偏置驱动时的 B 型波形



## 27.10 LCD 中断

LCD 模块提供两种中断方式。当 LCD 控制器从有效变为无效时将产生中断。中断也为 B 型波形提供无帧边界。LCD 时序发生提供了一个定义 LCD 帧时序的中断。

### 27.10.1 LCD 在模块关闭时产生中断

模块完成关闭（LCDA 从 1 变为 0）时产生 LCD 中断。

### 27.10.2 LCD 帧中断

一个新帧开始于 COM0 公共端信号的起始边沿。在 LCD 控制器完成对帧所需的所有像素数据的访问后，将立即产生中断。中断发生在帧边界前的某一固定时间（TFINT），如图 27-19 所示。在中断发生的 TFWR 时间后，LCD 控制器将开始访问下一帧数据。新数据必须在 TFWR 内写入，因为在此间隔后 LCD 控制器将开始访问下一帧数据。

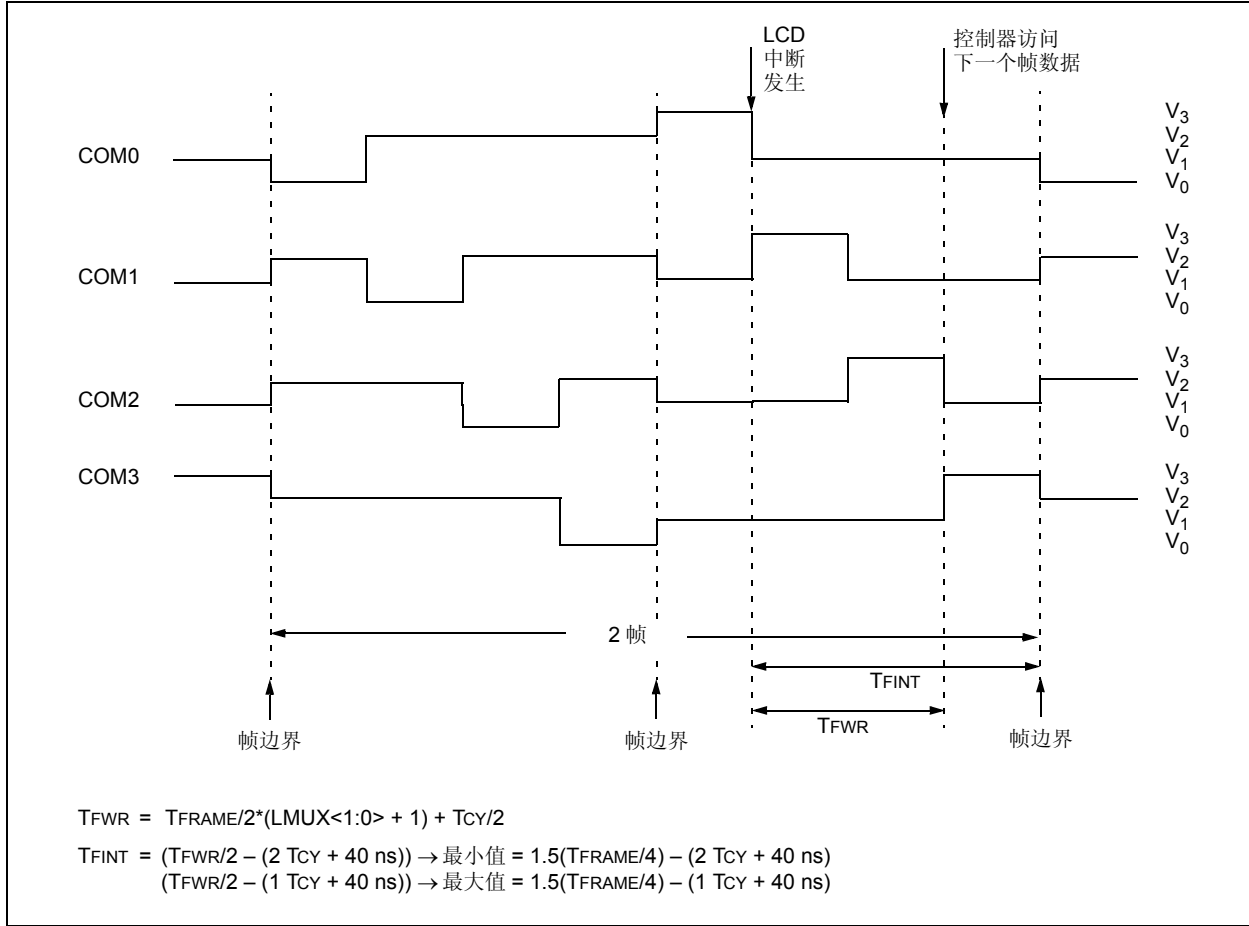
当 LCD 驱动器输出 B 型波形且 LMUX<1:0> 位不等于 00（静态驱动）时，必须处理一些其他问题。由于需要两帧来维持像素上的直流电压为零，因此在此期间像素数据要保持不变。一旦像素数据发生改变，奇数帧波形和偶数帧波形就不再互补，在面板中会引入一个直流分量。因此，当使用 B 型波形时，用户必须确保 LCD 像素更新发生在帧中断后的下一帧中。

在 B 型波形时要使写入序列正确，中断将只能发生在完整的相位间隔内。当禁止写操作时，一旦用户试图进行写操作，那么 LCDCON 寄存器的 WERR 位将被置 1，且不会发生写操作。

<b>注：</b> 当选择 A 型波形和选择不复用（静态）的 B 型波形时不会产生 LCD 帧中断。
--

# PIC16(L)F1933

图 27-19: 1/4 占空比驱动时的波形和中断时序 (示例——B 型, 非静态)





## 27.11 休眠期间的工作原理

LCD模块可以工作在休眠模式下。模式选择由LCDCON寄存器的SLPEN位控制。将SLPEN位置1允许LCD模块进入休眠模式。将SLPEN位清零允许该模块在休眠期间继续工作。

如果执行了SLEEP指令且SLPEN = 1，那么LCD模块将停止所有功能并进入电流消耗极低的模式。该模块将立即停止操作，并在段和公共端上输出最小LCD驱动电压。图27-20说明了此操作。

LCD模块可以配置为在休眠模式下运行。模式选择由LCDCON寄存器的SLPEN位控制。清零SLPEN并正确配置LCD模块时钟将允许LCD模块在休眠模式下运行。将SLPEN置1并正确执行LCD模块关闭操作将禁止LCD模块在休眠模式下运行从而节省功耗。

如果执行了SLEEP指令且SLPEN = 1，那么LCD模块将立即停止所有功能、将输出驱动为V<sub>SS</sub>并进入电流消耗极低的模式。在LCD模块被禁止并且当前周期完成后，确保在显示屏上没有直流电压，才可执行SLEEP指令。要禁止LCD模块，应清零LCDEN位。LCD模块将在当前帧后完成禁止过程、清零LCDA位并有选择地产生中断。

要在LCD被禁止的情况下，使器件正确地进入休眠模式需要以下步骤：

- 清零LCDEN
- 等待LCDA = 0（通过查询或中断）
- 执行SLEEP

在LCD模块时钟源为Fosc/4时，如果SLPEN = 0且执行了SLEEP，那么LCD模块将暂停工作，保持引脚驱动最近的LCD电压模式。过长的施加固定的LCD电压将损坏LCD显示屏。要防止损坏LCD显示屏，可以在休眠前正确地关闭LCD模块，或者改变LCD模块时钟以允许LCD模块继续在休眠期间运行。

如果执行了SLEEP指令、SLPEN = 0且LCD模块时钟为T1OSC或LFINTOSC，那么模块将继续显示LCDDATA寄存器的当前内容。在休眠模式下，不能更改LCD数据。如果LCDIE位置1，器件将在下一个LCD帧边界从休眠状态唤醒。LCD模块的电流消耗在此模式下不会降低；然而，器件的整体功耗将因CPU和其他外设的关闭而降低。

表27-8显示了LCD模块在休眠模式下使用三个可用时钟源中每一个时钟源时的状态。

**注：** LCDEN位清零时，LCD模块将在帧结束时禁止。此时，端口引脚将转换为数字功能。要降低由于数字输入引脚悬空而产生的功耗，应当使用端口和TRIS寄存器将LCD引脚驱动为低电平。

如果执行了SLEEP指令且SLPEN = 0，那么模块将继续显示LCDDATA寄存器中的当前内容。要使模块在休眠模式下继续工作，时钟源必须为LFINTOSC或T1OSC外部振荡器。在休眠模式下，不能更改LCD数据。LCD模块的电流消耗在此模式下不会降低；然而，器件的整体功耗将因内核和其他外设功能的关闭而降低。

表27-8显示了LCD模块在休眠模式下使用三个可用时钟源中每一个时钟源时的状态：

**表 27-8: 休眠期间 LCD 模块的状态**

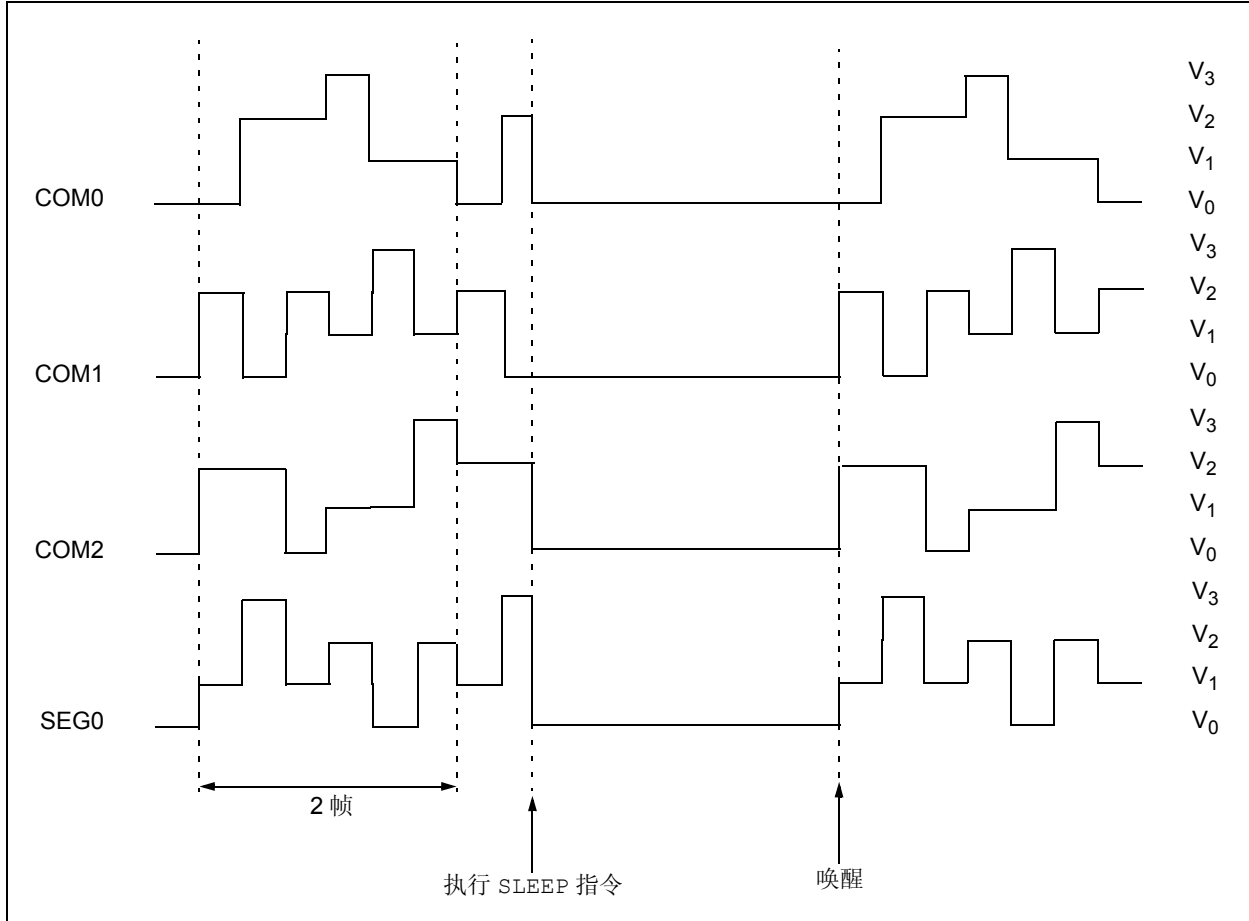
时钟源	SLPEN	休眠期间的工作
T1OSC	0	是
	1	否
LFINTOSC	0	是
	1	否
Fosc/4	0	否
	1	否

**注：** 必须使用LFINTOSC或外部T1OSC振荡器才能使LCD模块在休眠期间继续工作。

如果产生了LCD中断（非静态复用模式的B型波形）且LCDIE = 1，那么器件将在下一个帧边界从休眠模式唤醒。

# PIC16(L)F1933

图 27-20: 进入 / 退出休眠 (SLPEN = 1 时)



## 27.12 配置 LCD 模块

下面是配置 LCD 模块的步骤顺序。

1. 使用 LCDPS 寄存器的 LP<3:0> 位可以选择帧时钟预分频比。
2. 使用 LCDSEn 寄存器可以将相应的引脚配置为段驱动引脚。
3. 使用 LCDCON 寄存器对 LCD 模块作如下配置：
  - 使用 LMUX<1:0> 位配置复用和偏置模式
  - 使用 CS<1:0> 位配置时钟源
  - 使用 SLPEN 位配置休眠模式
4. 将初始值写入像素数据寄存器 LCDDATA0 至 LCDDATA11。
5. 清零 PIR2 寄存器的 LCD 中断标志位 LCDIF, 如果需要, 通过将 PIE2 寄存器的 LCDIE 位置 1 来允许中断。
6. 根据需要设置 LCDRL、LCDREF 和相关的 ANSELx 寄存器来配置偏置电压。
7. 通过将 LCDCON 寄存器的 LCDEN 位置 1 使能 LCD 模块。

## 27.13 禁止 LCD 模块

要禁止 LCD 模块, 应向 LCDCON 寄存器写入全 0。

## 27.14 LCD 电流消耗

使用 LCD 模块时, 电流消耗由以下三个因素决定:

- 振荡器选择
- LCD 偏置源
- LCD 段的电容

与这些因素相比, LCD 模块的电流消耗可以忽略。

### 27.14.1 振荡器选择

使用 LCD 模块时必须考虑所选时钟源消耗的电流。振荡器电流消耗信息请参见应用的电气规范章节。

### 27.14.2 LCD 偏置源

LCD 偏置源 (内部或外部) 对于电流消耗影响巨大。在保证对比度不变的情况下尽可能使用大电阻以使电流最小。

### 27.14.3 LCD 段的电容

可以把 LCD 段看作电容, 必须在每一帧对其进行充放电。LCD 段的大小和其采用的技术决定它的电容大小。

# PIC16(L)F1933

表 27-9: 与 LCD 操作相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	寄存器所在的页面
INTCON	GIE	PEIE	TMR0IE	INTE	IOCF	TMR0IF	INTF	IOCF	86
LCDCON	LCDEN	SLPEN	WERR	—	CS<1:0>		LMUX<1:0>		317
LCDCST	—	—	—	—	—	LCDCST<2:0>			320
LCDDATA0	SEG7 COM0	SEG6 COM0	SEG5 COM0	SEG4 COM0	SEG3 COM0	SEG2 COM0	SEG1 COM0	SEG0 COM0	321
LCDDATA1	SEG15 COM0	SEG14 COM0	SEG13 COM0	SEG12 COM0	SEG11 COM0	SEG10 COM0	SEG9 COM0	SEG8 COM0	321
LCDDATA3	SEG7 COM1	SEG6 COM1	SEG5 COM1	SEG4 COM1	SEG3 COM1	SEG2 COM1	SEG1 COM1	SEG0 COM1	321
LCDDATA4	SEG15 COM1	SEG14 COM1	SEG13 COM1	SEG12 COM1	SEG11 COM1	SEG10 COM1	SEG9 COM1	SEG8 COM1	321
LCDDATA6	SEG7 COM2	SEG6 COM2	SEG5 COM2	SEG4 COM2	SEG3 COM2	SEG2 COM2	SEG1 COM2	SEG0 COM2	321
LCDDATA7	SEG15 COM2	SEG14 COM2	SEG13 COM2	SEG12 COM2	SEG11 COM2	SEG10 COM2	SEG9 COM2	SEG8 COM2	321
LCDDATA9	SEG7 COM3	SEG6 COM3	SEG5 COM3	SEG4 COM3	SEG3 COM3	SEG2 COM3	SEG1 COM3	SEG0 COM3	321
LCDDATA10	SEG15 COM3	SEG14 COM3	SEG13 COM3	SEG12 COM3	SEG11 COM3	SEG10 COM3	SEG9 COM3	SEG8 COM3	321
LCDPS	WFT	BIASMD	LCDA	WA	LP<3:0>				318
LCDREF	LCDIRE	LCDIRS	LCDIRI	—	VLCD3PE	VLCD2PE	VLCD1PE	—	319
LCDRL	LRLAP<1:0>		LRLBP<1:0>		—	LRLAT<2:0>			328
LCDSE0	SE<7:0>								321
LCDSE1	SE<15:8>								321
PIE2	OSFIE	C2IE	C1IE	EEIE	BCLIE	LCDIE	—	CCP2IE	88
PIR2	OSFIF	C2IF	C1IF	EEIF	BCLIF	LCDIF	—	CCP2IF	91
T1CON	TMR1CS<1:0>		T1CKPS<1:0>		T1OSCEN	T1SYNC	—	TMR1ON	185

图注: — = 未实现单元, 读为 0。LCD 模块不使用阴影单元。

## 28.0 在线串行编程 (ICSP™)

ICSP™ 编程允许客户在生产电路板时使用未编程器件。编程可以在装配流程之后完成，从而可以使用最新版本的固件或者定制固件对器件编程。ICSP™ 编程需要 5 个引脚：

- ICSPCLK
- ICSPDAT
- MCLR/VPP
- VDD
- VSS

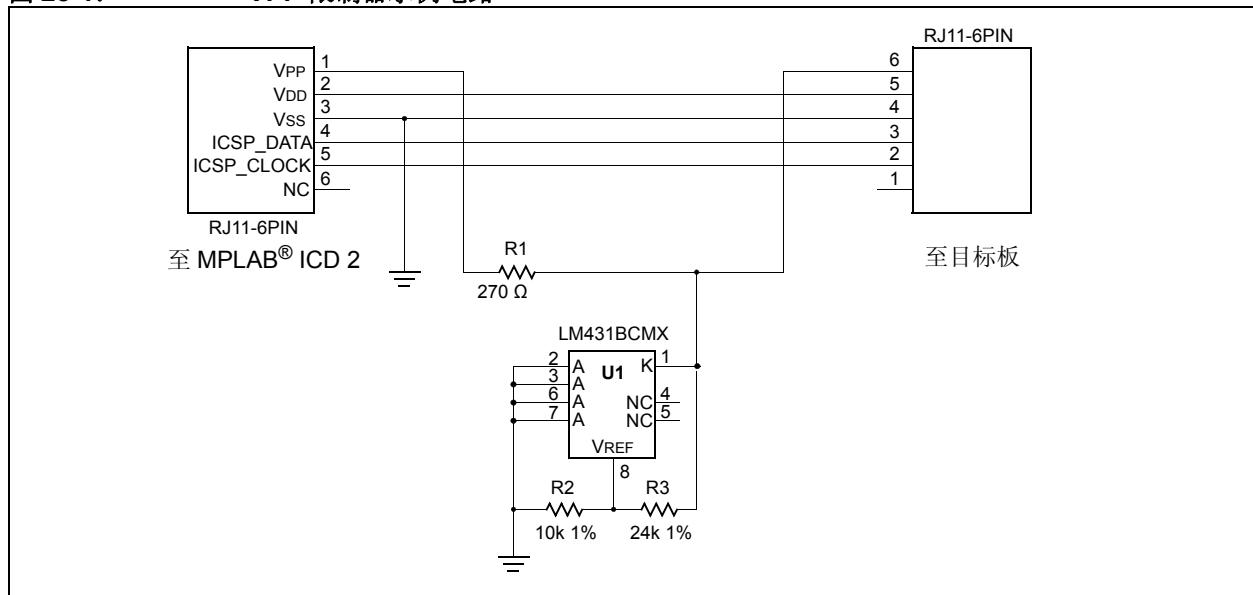
在编程 / 校验模式下，通过串行通信对程序存储器、用户 ID 和配置字进行编程。ICSPDAT 引脚是用于传输串行数据的双向 I/O，ICSPCLK 引脚是时钟输入引脚。关于 ICSP™ 的更多信息，请参见“PIC16193X/PIC16LF193X Memory Programming Specification” (DS41360A)。

## 28.1 高压编程进入模式

通过将 ICSPCLK 和 ICSPDAT 引脚保持为低电平然后将 MCLR/VPP 引脚上的电压升至  $V_{IH}$ ，可将器件置于高压编程进入模式。

某些编程器可产生大于  $V_{IH}$  (9.0V) 的  $V_{PP}$ ，从而需要一个外部电路来限制  $V_{PP}$  电压。示例电路，请参见图 28-1。

图 28-1: VPP 限制器示例电路



**注：** MPLAB® ICD 2 产生的  $V_{PP}$  电压大于 PIC16(L)F1933 的最大  $V_{PP}$  规范。

# PIC16(L)F1933

## 28.2 低压编程进入模式

低压编程进入模式允许PIC16(L)F1933器件在没有高压的情况下仅使用VDD进行编程。当配置字寄存器2的LVP位设置为1时，便使能了低压ICSP编程进入模式。要禁止低压ICSP模式，必须将LVP位设置为0。

进入低压编程进入模式需要以下步骤：

1. 将MCLR拉至VIL。
2. 在ICSPDAT引脚上输出32位密钥序列，而在ICSPCLK引脚上输出时钟。

在密钥序列完成后，只要保持在编程/校验模式下，MCLR就必须保持在VIL电平。

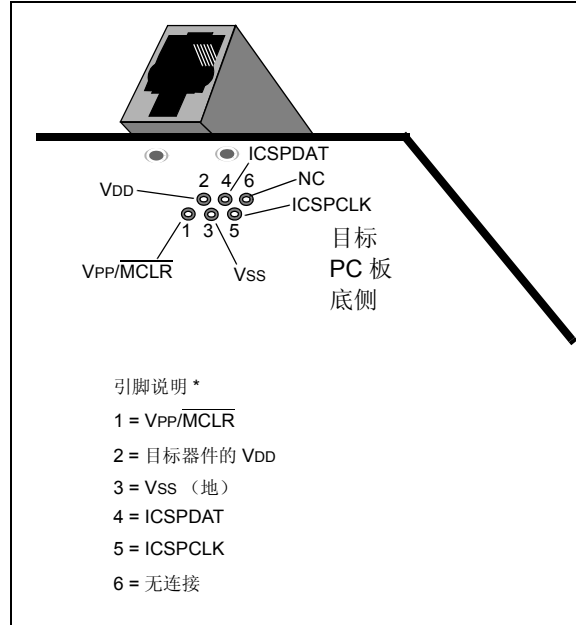
如果使能了低压编程（LVP = 1），将自动使能MCLR复位功能，且无法禁止。更多信息，请参见第6.3节“MCLR”。

只能通过使用高压编程模式将LVP位重新编程为0。

## 28.3 通用编程接口

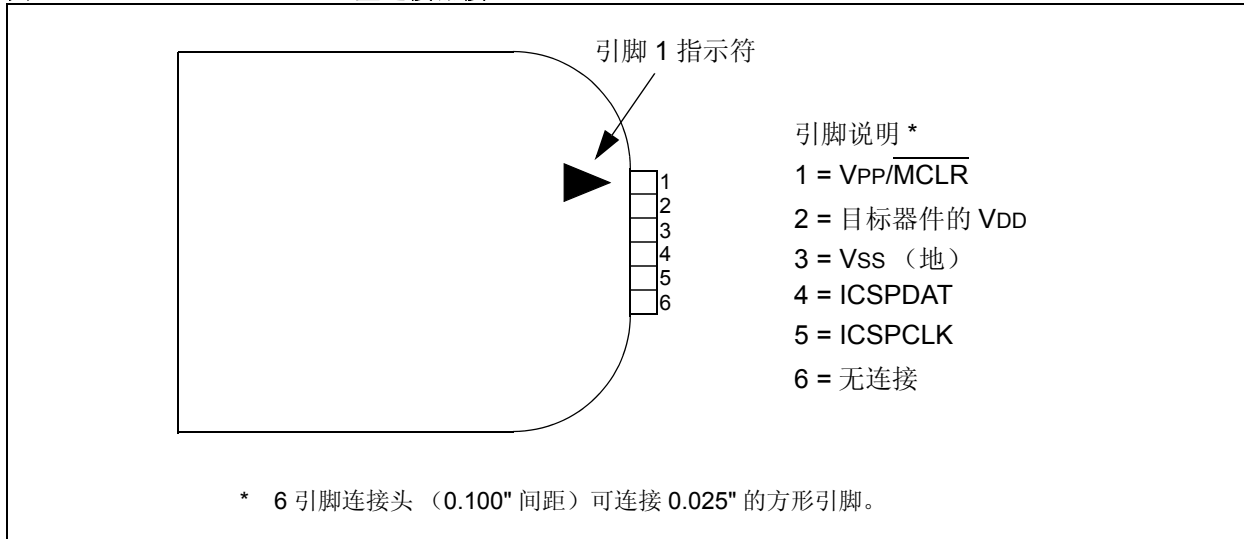
通常，与目标器件的连接通过一个ICSP™连接头完成。开发工具上常见的连接器是采用6P6C（6个引脚，6个连接器）配置的RJ-11。请参见图28-2。

图 28-2: ICD RJ-11 型连接器接口



另一个经常与PICkit™编程器一起使用的连接器是间距为0.1英寸的标准6引脚连接头。请参见图28-3。

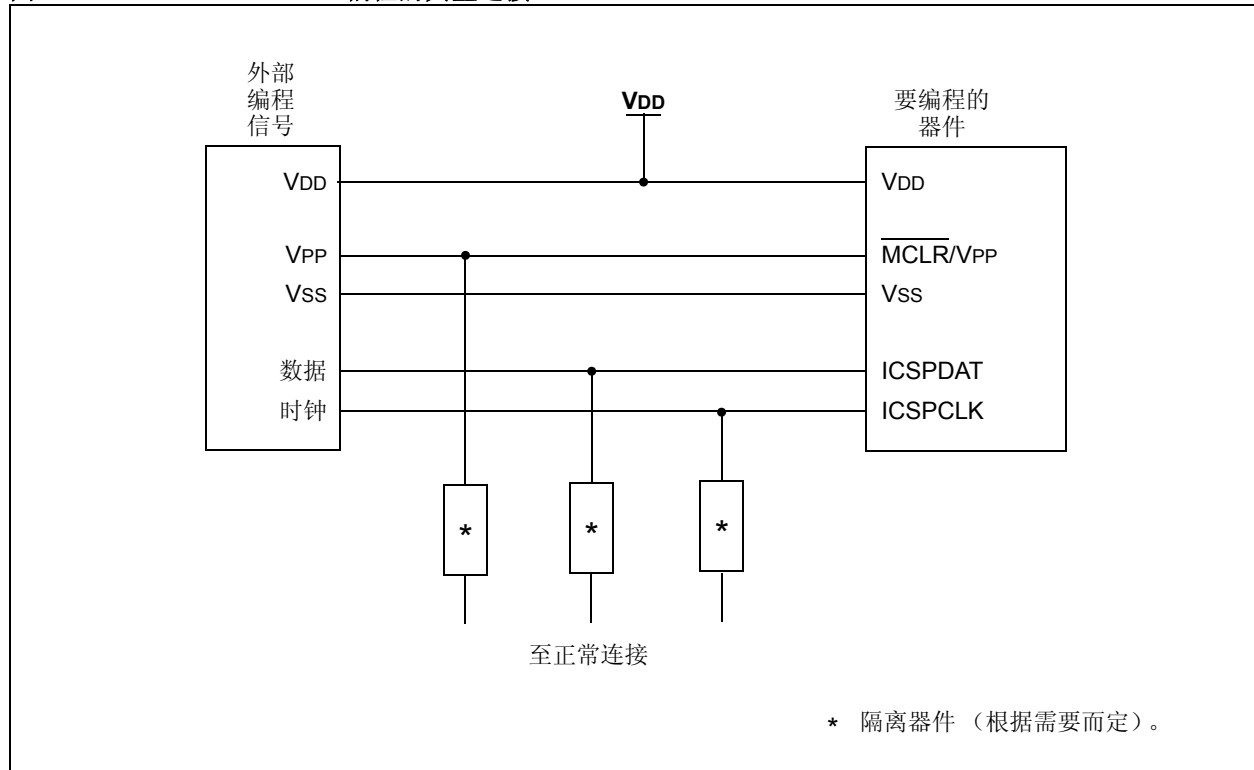
图 28-3: PICkit™ 型连接器接口



有关其他推荐接口，在进行 PCB 设计之前，请参见具体的器件编程器手册。

建议使用隔离器件将编程引脚与其他电路隔离。隔离类型主要取决于具体应用，可能包括电阻、二极管甚至跳线之类的器件。更多信息，请参见图 28-4。

图 28-4: ICSP™ 编程的典型连接



# PIC16(L)F1933

---

注:



## 29.0 指令集汇总

每条 PIC16 指令都是一个 14 位字，包含操作码和所有必需的操作数。操作码可分为以下三大类。

- 面向字节操作类指令
- 面向位操作类指令
- 立即数和控制操作类指令

立即数和控制操作类指令字的格式最为丰富。

表 29-3 列出了可被 MPASM™ 汇编器识别的指令。

所有指令都可在单指令周期内执行完毕，但也有例外，以下指令需要两或三个周期：

- 子程序调用需要两个周期（CALL 和 CALLW）
- 从中断或子程序返回需要两个周期（RETURN、RETLW 和 RETFIE）
- 程序跳转需要两个周期（GOTO、BRA、BRW、BTFSS、BTFSC、DECFSZ 和 INCSFZ）
- 当任何指令引用间接文件寄存器且相应的文件选择寄存器指向程序存储器时，还需要多用一个指令周期。

一个指令周期由 4 个振荡周期组成；如果振荡器频率为 4 MHz，则可提供 1 MHz 的标称指令执行速率。

所有指令示例均使用 0xhh 格式表示一个十六进制数，其中 h 表示一位十六进制数字。

## 29.1 读 - 修改 - 写操作

所有需要使用文件寄存器的指令都会执行读 - 修改 - 写（R-M-W）操作。根据指令或目标寄存器指示符 d 读寄存器、修改数据和保存结果。即使是有写寄存器的指令也将先对该寄存器进行读操作。

表 29-1: 操作码字段说明

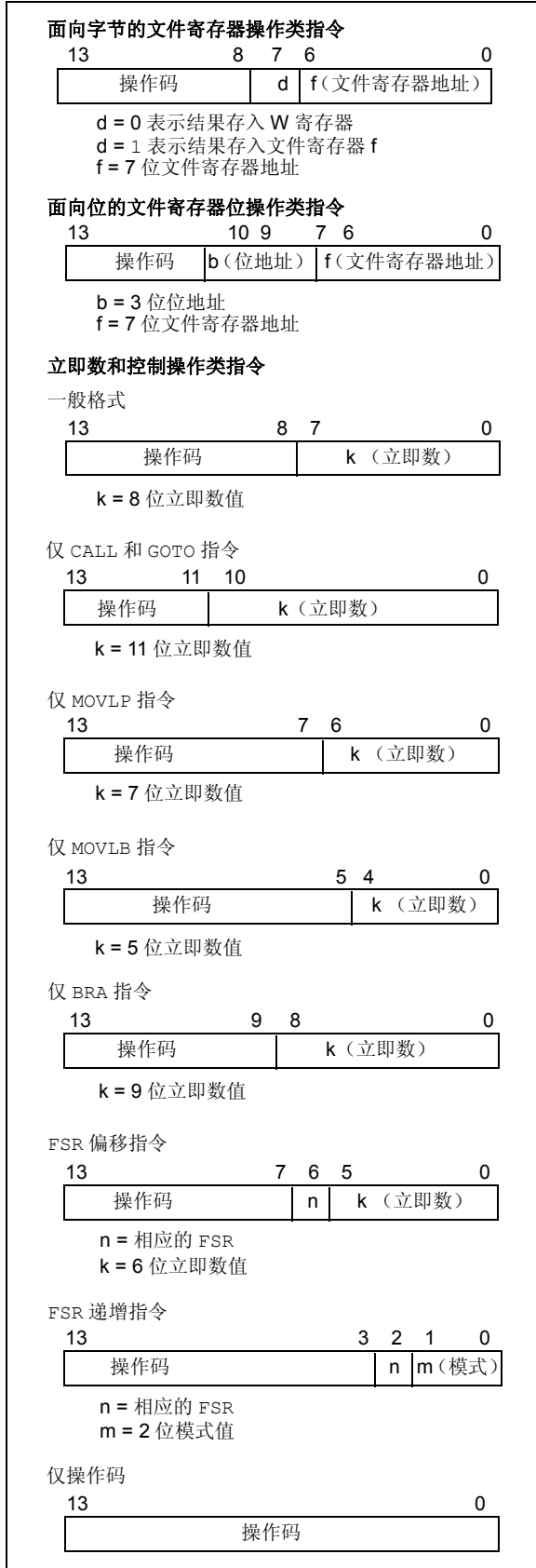
字段	说明
f	文件寄存器地址（0x00 到 0x7F）
W	工作寄存器（累加器）
b	8 位文件寄存器内的位地址
k	立即数字段、常数或标号
x	忽略（= 0 或 1）。 汇编器将生成 x = 0 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种格式。
d	目标寄存器选择；d = 0：结果保存至 W， d = 1：结果保存至文件寄存器 f。 默认值为 d = 1。
n	FSR 或 INDF 编号（0-1）
mm	预 / 后增 / 减模式选择

表 29-2: 缩写说明

字段	说明
PC	程序计数器
TO	超时位
C	进位位
DC	半进位位
Z	全零标志位
$\overline{PD}$	掉电位

# PIC16(L)F1933

图 29-1: 指令的一般格式



**表 29-3: PIC16(L)F1933 增强型指令集**

助记符, 操作数	说明	周期数	14 位操作码				受影响的状态位	注	
			MSb		LSb				
<b>面向字节的文件寄存器操作</b>									
ADDWF	f, d	W 和 f 相加	1	00	0111	dfff	ffff	C、DC 和 Z	2
ADDWFC	f, d	W 和 f 进行带进位的相加	1	11	1101	dfff	ffff	C、DC 和 Z	2
ANDWF	f, d	W 和 f 作逻辑与运算	1	00	0101	dfff	ffff	Z	2
ASRF	f, d	算术右移	1	11	0111	dfff	ffff	C 和 Z	2
LSLF	f, d	逻辑左移	1	11	0101	dfff	ffff	C 和 Z	2
LSRF	f, d	逻辑右移	1	11	0110	dfff	ffff	C 和 Z	2
CLRF	f	将 f 清零	1	00	0001	1fff	ffff	Z	2
CLRW	—	将 W 寄存器清零	1	00	0001	0000	00xx	Z	2
COMF	f, d	f 取反	1	00	1001	dfff	ffff	Z	2
DECf	f, d	f 递减 1	1	00	0011	dfff	ffff	Z	2
INCF	f, d	f 递增 1	1	00	1010	dfff	ffff	Z	2
IORWF	f, d	W 与 f 作逻辑或运算	1	00	0100	dfff	ffff	Z	2
MOVF	f, d	传送 f	1	00	1000	dfff	ffff	Z	2
MOVWF	f	将 W 的内容送到 f	1	00	0000	1fff	ffff		2
RLF	f, d	对 f 执行带进位的循环左移	1	00	1101	dfff	ffff	C	2
RRF	f, d	对 f 执行带进位的循环右移	1	00	1100	dfff	ffff	C	2
SUBWF	f, d	f 减去 W	1	00	0010	dfff	ffff	C、DC 和 Z	2
SUBWFB	f, d	f 减去 W (带借位)	1	11	1011	dfff	ffff	C、DC 和 Z	2
SWAPF	f, d	将 f 中的两个半字节进行交换	1	00	1110	dfff	ffff		2
XORWF	f, d	W 与 f 作逻辑异或运算	1	00	0110	dfff	ffff	Z	2
<b>面向字节的跳过操作</b>									
DECFSZ	f, d	f 递减 1, 为 0 则跳过	1(2)	00	1011	dfff	ffff		1, 2
INCFSZ	f, d	f 递增 1, 为 0 则跳过	1(2)	00	1111	dfff	ffff		1, 2
<b>面向位的文件寄存器操作</b>									
BCF	f, b	将 f 中的某位清零	1	01	00bb	bfff	ffff		2
BSF	f, b	将 f 中的某位置 1	1	01	01bb	bfff	ffff		2
<b>面向位的跳过操作</b>									
BTFSC	f, b	检测 f 中的某位, 为 0 则跳过	1 (2)	01	10bb	bfff	ffff		1, 2
BTFSS	f, b	检测 f 中的某位, 为 1 则跳过	1 (2)	01	11bb	bfff	ffff		1, 2
<b>立即数操作</b>									
ADDLW	k	立即数和 W 相加	1	11	1110	kkkk	kkkk	C、DC 和 Z	
ANDLW	k	立即数与 W 作逻辑与运算	1	11	1001	kkkk	kkkk	Z	
IORLW	k	立即数与 W 作逻辑或运算	1	11	1000	kkkk	kkkk	Z	
MOVLB	k	将立即数传送到 BSR	1	00	0000	001k	kkkk		
MOVLW	k	将立即数传送到 PCLATH	1	11	0001	1kkk	kkkk		
MOVLW	k	将立即数传送到 W	1	11	0000	kkkk	kkkk		
SUBLW	k	从立即数中减去 W 的内容	1	11	1100	kkkk	kkkk	C、DC 和 Z	
XORLW	k	立即数与 W 作逻辑异或运算	1	11	1010	kkkk	kkkk	Z	

注 1: 如果程序计数器 (PC) 被修改或条件测试为真, 则执行该指令需要两个周期。第二个周期执行一条 NOP 指令。

注 2: 如果该指令寻址的是 INDF 寄存器且相应 FSR 的 MSb 置 1, 则该指令需要一个额外的指令周期。

# PIC16(L)F1933

表 29-3: PIC16(L)F1933 增强型指令集 (续)

助记符, 操作数	说明	周期数	14 位操作码				受影响的状态位	注:	
			MSb			LSb			
<b>控制类操作</b>									
BRA	k	相对跳转	2	11	001k	kkkk	kkkk		
BRW	-	将 W 寄存器的内容作为偏移量进行相对跳转	2	00	0000	0000	1011		
CALL	k	调用子程序	2	10	0kkk	kkkk	kkkk		
CALLW	-	调用地址由 W 寄存器指定的子程序	2	00	0000	0000	1010		
GOTO	k	跳转到地址	2	10	1kkk	kkkk	kkkk		
RETFIE	k	从中断返回	2	00	0000	0000	1001		
RETLW	k	返回并将立即数传送到 W	2	11	0100	kkkk	kkkk		
RETURN	-	从子程序返回	2	00	0000	0000	1000		
<b>固有操作</b>									
CLRWDT	-	清零看门狗定时器	1	00	0000	0110	0100	$\overline{TO}$ 和 $\overline{PD}$	
NOP	-	空操作	1	00	0000	0000	0000		
OPTION	-	将 W 寄存器的内容装入 OPTION_REG 寄存器	1	00	0000	0110	0010		
RESET	-	软件器件复位	1	00	0000	0000	0001		
SLEEP	-	进入待机模式	1	00	0000	0110	0011	$\overline{TO}$ 和 $\overline{PD}$	
TRIS	f	将 W 寄存器的内容装入 TRIS 寄存器	1	00	0000	0110	0fff		
<b>C 编译器优化后的操作</b>									
ADDFSR	n, k	立即数 k 与 FSRn 相加	1	11	0001	0nkk	kkkk		
MOVIW	n mm	将间接 FSRn 的内容传送到 W 寄存器, 带预 / 后增 / 减修改符, mm	1	00	0000	0001	0nmm	Z	<b>2, 3</b>
MOVWI	k[n]	将 INDFn 的内容传送到 W 寄存器, 采用变址间接寻址模式	1	11	1111	0nkk	kkkk	Z	<b>2</b>
	n mm	将 W 寄存器的内容传送到间接 FSRn, 带预 / 后增 / 减修改符, mm	1	00	0000	0001	1nmm		<b>2, 3</b>
	k[n]	将 W 寄存器的内容传送到 INDFn, 采用变址间接寻址模式	1	11	1111	1nkk	kkkk		<b>2</b>

- 注 1: 如果程序计数器 (PC) 被修改或条件测试为真, 则执行该指令需要两个周期。第二个周期执行一条 NOP 指令。  
 2: 如果该指令寻址的是 INDF 寄存器且相应 FSR 的 MSb 置 1, 则该指令需要一个额外的指令周期。  
 3: 请参见 MOVIW 和 MOVWI 指令说明中的表。

## 29.2 指令说明

ADDFSR	立即数与 FSRn 相加
语法:	[ 标号 ] ADDFSR FSRn, k
操作数:	$-32 \leq k \leq 31$ $n \in [0, 1]$
操作:	$FSR(n) + k \rightarrow FSR(n)$
受影响的状态位:	无
说明:	有符号的 6 位立即数 k 与 FSRnH:FSRnL 寄存器对的内容相加。  FSRn 限制在范围 0000h - FFFFh 内。传送超出这些边界将导致 FSR 返回。

ADDLW	立即数和 W 相加
语法:	[ 标号 ] ADDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) + k \rightarrow (W)$
受影响的状态位:	C、DC 和 Z
说明:	将 W 寄存器的内容与 8 位立即数 k 相加，结果存放到 W 寄存器。

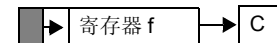
ADDWF	W 和 f 相加
语法:	[ 标号 ] ADDWF f, d
操作数:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) + (f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	C、DC 和 Z
说明:	将 W 寄存器的内容与 f 寄存器的内容相加。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ADDWFC	W 和 f 进行带进位的相加
语法:	[ 标号 ] ADDWFC f {,d}
操作数:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) + (f) + (C) \rightarrow \text{目标寄存器}$
受影响的状态位:	C、DC 和 Z
说明:	将 W 寄存器的内容、进位标志位和数据存储单元 f 的内容相加。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存放到数据存储单元 f。

ANDLW	立即数与 W 作逻辑与运算
语法:	[ 标号 ] ANDLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .AND. (k) \rightarrow (W)$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 作逻辑与运算。结果存放到 W 寄存器。

ANDWF	W 和 f 作逻辑与运算
语法:	[ 标号 ] ANDWF f, d
操作数:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(W) .AND. (f) \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	W 寄存器与 f 寄存器作逻辑与运算。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ASRF	算术右移
语法:	[ 标号 ] ASRF f {,d}
操作数:	$0 \leq f \leq 127$ $d \in [0, 1]$
操作:	$(f < 7 >) \rightarrow \text{目标寄存器 } < 7 >$ $(f < 7:1 >) \rightarrow \text{目标寄存器 } < 6:0 >$ $(f < 0 >) \rightarrow C$
受影响的状态位:	C 和 Z
说明:	将寄存器 f 的内容连同进位标志位一起右移 1 位。MSb 保持不变。如果 d 为 0，结果存放到 W 寄存器。如果 d 为 1，结果存回寄存器 f。



# PIC16(L)F1933

## BCF 将 f 寄存器中的某位清零

语法: [ 标号 ] BCF f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
操作:  $0 \rightarrow (f<b>)$   
受影响的状态位: 无  
说明: 将寄存器 f 中的位 b 清零。

## BRA 相对跳转

语法: [ 标号 ] BRA 标号  
[ 标号 ] BRA \$+k  
操作数:  $-256 \leq \text{标号} - \text{PC} + 1 \leq 255$   
 $-256 \leq k \leq 255$   
操作:  $(\text{PC}) + 1 + k \rightarrow \text{PC}$   
受影响的状态位: 无  
说明: 有符号的 9 位立即数 k 与 PC 相加。由于 PC 将递增 1 以取下一条指令，所以新地址为  $\text{PC} + 1 + k$ 。该指令为双周期指令。此跳转有范围限制。

## BRW 将 W 寄存器的内容作为偏移量进行相对跳转

语法: [ 标号 ] BRW  
操作数: 无  
操作:  $(\text{PC}) + (\text{W}) \rightarrow \text{PC}$   
受影响的状态位: 无  
说明: W 的内容（无符号）与 PC 相加。由于 PC 将递增 1 以取下一条指令，所以新地址为  $\text{PC} + 1 + (\text{W})$ 。该指令为双周期指令。

## BSF 将 f 中的某位置 1

语法: [ 标号 ] BSF f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
操作:  $1 \rightarrow (f<b>)$   
受影响的状态位: 无  
说明: 将寄存器 f 中的位 b 置 1。

## BTFSF 检测 f 中的某位，为 0 则跳过

语法: [ 标号 ] BTFSF f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b \leq 7$   
操作: 如果  $(f<b>) = 0$  则跳过  
受影响的状态位: 无  
说明: 如果 f 寄存器中的位 b 为 1，则执行下一条指令。  
如果 f 寄存器中的位 b 为 0，则丢弃下一条指令，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

## BTFSF 检测 f 中的某位，为 1 则跳过

语法: [ 标号 ] BTFSF f,b  
操作数:  $0 \leq f \leq 127$   
 $0 \leq b < 7$   
操作: 如果  $(f<b>) = 1$  则跳过  
受影响的状态位: 无  
说明: 如果 f 寄存器中的位 b 为 0，则执行下一条指令。  
如果位 b 为 1，那么下一条指令将丢弃，转而执行一条 NOP 指令，从而使该指令成为双周期指令。

CALL	调用子程序
语法:	[ 标号 ] CALL k
操作数:	$0 \leq k \leq 2047$
操作:	(PC)+ 1 → TOS, k → PC<10:0>, (PCLATH<6:3>) → PC<14:11>
受影响的状态位:	无
说明:	调用子程序。首先, 将返回地址 (PC+1) 压入堆栈。将 11 位立即数地址装入 PC 位 <10:0>。将 PCLATH 的内容装入 PC 的高位。CALL 是双周期指令。

CALLW	调用地址由 W 寄存器指定的子程序
语法:	[ 标号 ] CALLW
操作数:	无
操作:	(PC) + 1 → TOS, (W) → PC<7:0>, (PCLATH<6:0>) → PC<14:8>
受影响的状态位:	无
说明:	调用地址由 W 寄存器指定的子程序。首先, 将返回地址 (PC + 1) 压入返回堆栈。然后, 将 W 寄存器的内容装入 PC<7:0>, 将 PCLATH 的内容装入 PC<14:8>。CALLW 是双周期指令。

CLRF	将 f 清零
语法:	[ 标号 ] CLRF f
操作数:	$0 \leq f \leq 127$
操作:	00h → (f) 1 → Z
受影响的状态位:	Z
说明:	f 寄存器的内容被清零, Z 位置 1。

CLRW	将 W 寄存器清零
语法:	[ 标号 ] CLRW
操作数:	无
操作:	00h → (W) 1 → Z
受影响的状态位:	Z
说明:	W 寄存器被清零。全零标志位 (Z) 置 1。

CLRWDT	清零看门狗定时器
语法:	[ 标号 ] CLRWDT
操作数:	无
操作:	00h → WDT 0 → $\overline{\text{WDT}}$ 预分频器 1 → $\overline{\text{TO}}$ 1 → $\overline{\text{PD}}$
受影响的状态位:	$\overline{\text{TO}}$ 和 $\overline{\text{PD}}$
说明:	CLRWDT 指令复位看门狗定时器。还将复位 WDT 的预分频器。状态位 $\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 置 1。

COMF	f 取反
语法:	[ 标号 ] COMF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$\overline{(f)} \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将 f 寄存器的内容取反。如果 d 为 0, 结果存放在 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

DECF	f 递减 1
语法:	[ 标号 ] DECF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) - 1 \rightarrow (\text{目标寄存器})$
受影响的状态位:	Z
说明:	将 f 寄存器的内容减 1。如果 d 为 0, 结果存放在 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

# PIC16(L)F1933

## DECFSZ      f 递减 1, 为 0 则跳过

语法:	[ 标号] DECFSZ f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) - 1 \rightarrow$ (目标寄存器); 结果 = 0 则跳过
受影响的状态位:	无
说明:	将 f 寄存器的内容减 1。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。 如果结果为 1, 则执行下一条指令。如果结果为 0, 则转而执行一条 NOP 指令, 从而使该指令成为双周期指令。

## INCFSZ      f 递增 1, 为 0 则跳过

语法:	[ 标号] INCFSZ f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) + 1 \rightarrow$ (目标寄存器), 结果 = 0 则跳过
受影响的状态位:	无
说明:	将 f 寄存器的内容加 1。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。 如果结果为 1, 则执行下一条指令。如果结果为 0, 则转而执行 NOP 指令, 从而使该指令成为双周期指令。

## GOTO      无条件跳转

语法:	[ 标号] GOTO k
操作数:	$0 \leq k \leq 2047$
操作:	$k \rightarrow PC<10:0>$ $PCLATH<6:3> \rightarrow PC<14:11>$
受影响的状态位:	无
说明:	GOTO 是无条件跳转指令。将 11 位立即数值装入 PC 位 <10:0>。将 PCLATH<4:3> 的内容装入 PC 的高位。GOTO 是双周期指令。

## IORLW      立即数与 W 作逻辑或运算

语法:	[ 标号] IORLW k
操作数:	$0 \leq k \leq 255$
操作:	$(W) .OR. k \rightarrow (W)$
受影响的状态位:	Z
说明:	将 W 寄存器的内容与 8 位立即数 k 作逻辑或运算。结果存放到 W 寄存器。

## INCF      f 递增 1

语法:	[ 标号] INCF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(f) + 1 \rightarrow$ (目标寄存器)
受影响的状态位:	Z
说明:	将 f 寄存器的内容加 1。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

## IORWF      W 与 f 作逻辑或运算

语法:	[ 标号] IORWF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	$(W) .OR. (f) \rightarrow$ (目标寄存器)
受影响的状态位:	Z
说明:	W 寄存器与 f 寄存器作逻辑或运算。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。



**LSLF**                    **逻辑左移**

---

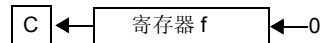
语法:                    [ 标号] LSLF f {,d}

操作数:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

操作:                    (f<7>) → C  
(f<6:0>) → 目标寄存器 <7:1>  
0 → 目标寄存器 <0>

受影响的状态位:    C 和 Z

说明:                    将 f 寄存器的内容连同进位标志位一起左移 1 位。0 移入 LSB。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。



**LSRF**                    **逻辑右移**

---

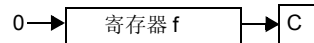
语法:                    [ 标号] LSRF f {,d}

操作数:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

操作:                    0 → 目标寄存器 <7>  
(f<7:1>) → 目标寄存器 <6:0>,  
(f<0>) → C,

受影响的状态位:    C 和 Z

说明:                    将寄存器 f 的内容连同进位标志位一起右移 1 位。0 移入 MSb。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。



**MOVF**                    **传送 f**

---

语法:                    [ 标号] MOVF f,d

操作数:                 $0 \leq f \leq 127$   
 $d \in [0,1]$

操作:                    (f) → ( 目标寄存器 )

受影响的状态位:    Z

说明:                    根据 d 的状态, 将寄存器 f 的内容传送到目标寄存器。如果 d = 0, 目标寄存器为 W 寄存器。如果 d = 1, 则目标寄存器为文件寄存器 f。由于状态标志位 Z 会受到指令结果的影响, 可用 d = 1 检测文件寄存器。

指令字数:                1

指令周期数:            1

示例:                    MOVF     FSR, 0

执行指令后

W = FSR 寄存器中的值

Z = 1

# PIC16(L)F1933

## MOVIW 将 INDFn 的内容传送到 W 寄存器

语法: [ 标号 ] MOVIW ++FSRn  
 [ 标号 ] MOVIW --FSRn  
 [ 标号 ] MOVIW FSRn++  
 [ 标号 ] MOVIW FSRn--  
 [ 标号 ] MOVIW k[FSRn]

操作数:  $n \in [0,1]$   
 $mm \in [00,01,10,11]$   
 $-32 \leq k \leq 31$

操作: INDFn  $\rightarrow$  W  
 有效的地址由以下项决定

- FSR + 1 ( 预增 1 )
- FSR - 1 ( 预减 1 )
- FSR + k ( 相对偏移 )

执行传送指令后, FSR 值为以下任一项:

- FSR + 1 ( 所有值都加 1 )
- FSR - 1 ( 所有值都减 1 )
- 不变

受影响的状态位: Z

模式	语法	mm
预增	++FSRn	00
预减	--FSRn	01
后增	FSRn++	10
后减	FSRn--	11

说明: 该指令用于在 W 寄存器和任一间接寄存器 (INDFn) 之间传送数据。执行该传送指令之前 / 之后, 可通过对其进行预 / 后增 / 减来更新指针 (FSRn)。

注: INDFn 寄存器不是物理寄存器。任何访问 INDFn 寄存器的指令实际上访问的是由 FSRn 指定的地址处的寄存器。

FSRn 限制在范围 0000h - FFFFh 内。递增 / 递减使其超出边界会导致其返回。

## MOVLB 将立即数传送到 BSR

语法: [ 标号 ] MOVLB k

操作数:  $0 \leq k \leq 15$

操作:  $k \rightarrow$  BSR

受影响的状态位: 无

说明: 将 5 位立即数 k 装入存储区选择寄存器 (BSR)。

## MOVLP 将立即数传送到 PCLATH

语法: [ 标号 ] MOVLP k

操作数:  $0 \leq k \leq 127$

操作:  $k \rightarrow$  PCLATH

受影响的状态位: 无

说明: 将 7 位立即数 k 装入 PCLATH 寄存器。

## MOVLW 将立即数传送到 W

语法: [ 标号 ] MOVLW k

操作数:  $0 \leq k \leq 255$

操作:  $k \rightarrow$  (W)

受影响的状态位: 无

说明: 将 8 位立即数 k 装入 W 寄存器。“无关位”被汇编为 0。

指令字数: 1

指令周期数: 1

示例: `MOVLW 0x5A`

执行指令后  
 $W = 0x5A$

## MOVWF 将 W 的内容传送到 f

语法: [ 标号 ] MOVWF f

操作数:  $0 \leq f \leq 127$

操作: (W)  $\rightarrow$  (f)

受影响的状态位: 无

说明: 将 W 寄存器中的数据传送到寄存器 f。

指令字数: 1

指令周期数: 1

示例: `MOVWF OPTION_REG`

执行指令前  
 $OPTION\_REG = 0xFF$   
 $W = 0x4F$

执行指令后  
 $OPTION\_REG = 0x4F$   
 $W = 0x4F$

## MOVWI 将 W 寄存器的内容传送到 INDFn

语法: [ 标号 ] MOVWI ++FSRn  
 [ 标号 ] MOVWI --FSRn  
 [ 标号 ] MOVWI FSRn++  
 [ 标号 ] MOVWI FSRn--  
 [ 标号 ] MOVWI k[FSRn]

操作数:  $n \in [0,1]$   
 $mm \in [00,01, 10, 11]$   
 $-32 \leq k \leq 31$

操作:  $W \rightarrow \text{INDFn}$   
 有效的地址由以下项决定

- FSR + 1 ( 预增 1 )
- FSR -1 ( 预减 1 )
- FSR + k ( 相对偏移 )

执行传送指令后, FSR 值为以下任一项:

- FSR + 1 ( 所有值都加 1 )
- FSR -1 ( 所有值都减 1 )

不变

受影响的状态位: 无

模式	语法	mm
预增	++FSRn	00
预减	--FSRn	01
后增	FSRn++	10
后减	FSRn--	11

说明: 该指令用于在 W 寄存器和任一间接寄存器 (INDFn) 之间传送数据。执行该传送指令之前 / 之后, 可通过对其进行预 / 后增 / 减来更新指针 (FSRn)。

注: INDFn 寄存器不是物理寄存器。任何访问 INDFn 寄存器的指令实际上访问的是由 FSRn 指定的地址处的寄存器。

FSRn 限制在范围 0000h - FFFFh 内。递增 / 递减使其超出边界会导致其返回。

对 FSRn 的递增 / 递减操作不会影响任何状态位。

## NOP 空操作

语法: [ 标号 ] NOP

操作数: 无

操作: 空操作

受影响的状态位: 无

说明: 空操作。

指令字数: 1

指令周期数: 1

示例: NOP

## OPTION 将 W 寄存器的内容装入 OPTION\_REG 寄存器

语法: [ 标号 ] OPTION

操作数: 无

操作:  $(W) \rightarrow \text{OPTION\_REG}$

受影响的状态位: 无

说明: 将 W 寄存器中的数据传送到 OPTION\_REG 寄存器。

指令字数: 1

指令周期数: 1

示例: OPTION

执行指令前  
 OPTION\_REG = 0xFF  
 W = 0x4F

执行指令后  
 OPTION\_REG = 0x4F  
 W = 0x4F

## RESET 软件复位

语法: [ 标号 ] RESET

操作数: 无

操作: 执行器件复位。复位 PCON 寄存器的 RI 标志位。

受影响的状态位: 无

说明: 该指令为用软件执行硬件复位提供了一种方法。

# PIC16(L)F1933

## RETFIE 从中断返回

语法:	[ 标号] RETFIE
操作数:	无
操作:	TOS → PC, 1 → GIE
受影响的状态位:	无
说明:	从中断返回。执行出栈操作, 将栈顶 (TOS) 的内容装入 PC。通过将全局中断允许位 GIE (INTCON<7>) 置 1 来允许中断。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	RETFIE 中断后 PC = TOS GIE = 1

## RETLW 返回并将立即数传送到 W

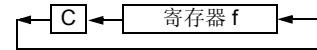
语法:	[ 标号] RETLW k
操作数:	$0 \leq k \leq 255$
操作:	k → (W); TOS → PC
受影响的状态位:	无
说明:	将 8 位立即数 k 装入 W 寄存器。将栈顶内容 (返回地址) 装入程序计数器。这是一条双周期指令。
指令字数:	1
指令周期数:	2
示例:	CALL TABLE;W contains table ;offset value ;W now has table value . . ADDWF PC ;W = offset RETLW k1 ;Begin table RETLW k2 ; . . RETLW kn ; End of table
执行指令前	W = 0x07
执行指令后	W = k8 的值

## RETURN 从子程序返回

语法:	[ 标号] RETURN
操作数:	无
操作:	TOS → PC
受影响的状态位:	无
说明:	从子程序返回。执行出栈操作, 将栈顶 (TOS) 内容装入程序计数器。这是一条双周期指令。

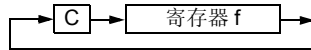
## RLF 对 f 执行带进位的循环左移

语法:	[ 标号] RLF f,d
操作数:	$0 \leq f \leq 127$ $d \in [0,1]$
操作:	参见下面的说明
受影响的状态位:	C
说明:	将寄存器 f 的内容连同进位标志位一起循环左移 1 位。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。
指令字数:	1
指令周期数:	1
示例:	RLF REG1,0 执行指令前 REG1 = 1110 0110 C = 0 执行指令后 REG1 = 1110 0110 W = 1100 1100 C = 1



## RRF 对 f 执行带进位的循环右移

语法: [ 标号 ] RRF f,d  
 操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 操作: 参见下面的说明  
 受影响的状态位: C  
 说明: 将寄存器 f 的内容连同进位标志位一起循环右移 1 位。如果 d 为 0, 结果存放在 W 寄存器。如果 d 为 1, 结果放回寄存器 f。



## SLEEP 进入休眠模式

语法: [ 标号 ] SLEEP  
 操作数: 无  
 操作:  $00h \rightarrow WDT$ ,  
 $0 \rightarrow \overline{WDT}$  预分频器,  
 $1 \rightarrow \overline{TO}$ ,  
 $0 \rightarrow \overline{PD}$   
 受影响的状态位:  $\overline{TO}$  和  $\overline{PD}$   
 说明: 掉电状态位  $\overline{PD}$  清零。超时状态位  $\overline{TO}$  置 1。看门狗定时器及其预分频器清零。振荡器停振, 处理器进入休眠模式。

## SUBLW 从立即数中减去 W 的内容

语法: [ 标号 ] SUBLW k  
 操作数:  $0 \leq k \leq 255$   
 操作:  $k - (W) \rightarrow (W)$   
 受影响的状态位: C、DC 和 Z  
 说明: 从 8 位立即数 k 中减去 W 寄存器的内容 (采用二进制补码方法进行运算)。结果存放在 W 寄存器。

C = 0	$W > k$
C = 1	$W \leq k$
DC = 0	$W\langle 3:0 \rangle > k\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq k\langle 3:0 \rangle$

## SUBWF f 减去 W

语法: [ 标号 ] SUBWF f,d  
 操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 操作:  $(f) - (W) \rightarrow (\text{目标寄存器})$   
 受影响的状态位: C、DC 和 Z  
 说明: 从寄存器 f 中减去 W 寄存器的内容 (采用二进制补码方法进行运算)。如果 d 为 0, 结果存放在 W 寄存器。如果 d 为 1, 结果放回寄存器 f。

C = 0	$W > f$
C = 1	$W \leq f$
DC = 0	$W\langle 3:0 \rangle > f\langle 3:0 \rangle$
DC = 1	$W\langle 3:0 \rangle \leq f\langle 3:0 \rangle$

## SUBWFB f 减去 W (带借位)

语法: SUBWFB f {,d}  
 操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
 操作:  $(f) - (W) - (\overline{B}) \rightarrow \text{目标寄存器}$   
 受影响的状态位: C、DC 和 Z  
 说明: 从寄存器 f 中减去 W 寄存器的内容以及借位 (进位) 标志位 (采用二进制补码方法进行运算)。如果 d 为 0, 结果存放在 W 寄存器。如果 d 为 1, 结果放回寄存器 f。

# PIC16(L)F1933

---

## SWAPF 将 f 中的两个半字节进行交换

---

语法: [ 标号 ] SWAPF f,d  
操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
操作: (f<3:0>) → (目标寄存器 <7:4>),  
(f<7:4>) → (目标寄存器 <3:0>)  
受影响的状态位: 无  
说明: 将寄存器 f 的高半字节和低半字节交换。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存放到寄存器 f。

## TRIS 将 W 寄存器的内容装入 TRIS 寄存器

---

语法: [ 标号 ] TRIS f  
操作数:  $5 \leq f \leq 7$   
操作: (W) → TRIS 寄存器 f  
受影响的状态位: 无  
说明: 将 W 寄存器中的数据传送到 TRIS 寄存器。  
当 f = 5 时, 数据装入 TRISA。  
当 f = 6 时, 数据装入 TRISB。  
当 f = 7 时, 数据装入 TRISC。

## XORLW 立即数与 W 作逻辑异或运算

---

语法: [ 标号 ] XORLW k  
操作数:  $0 \leq k \leq 255$   
操作: (W) .XOR. k → (W)  
受影响的状态位: Z  
说明: 将 W 寄存器的内容与 8 位立即数 k 作逻辑异或运算。结果存放到 W 寄存器。

## XORWF W 与 f 作逻辑异或运算

---

语法: [ 标号 ] XORWF f,d  
操作数:  $0 \leq f \leq 127$   
 $d \in [0,1]$   
操作: (W) .XOR. (f) → (目标寄存器)  
受影响的状态位: Z  
说明: 将 W 寄存器的内容与 f 寄存器的内容作逻辑异或运算。如果 d 为 0, 结果存放到 W 寄存器。如果 d 为 1, 结果存回寄存器 f。

## 30.0 电气规范 (PIC16(L)F1933)

### 绝对最大值 (†)

偏置电压下的环境温度 .....	-40°C 至 +125°C
储存温度 .....	-65°C 至 +150°C
V <sub>DD</sub> 相对于 V <sub>SS</sub> 的电压, PIC16F1933 .....	-0.3V 至 +6.5V
V <sub>CAP</sub> 引脚相对于 V <sub>SS</sub> 的电压, PIC16F1933 .....	-0.3V 至 +4.0V
V <sub>DD</sub> 相对于 V <sub>SS</sub> 的电压, PIC16LF1933 .....	-0.3V 至 +4.0V
MCLR 相对于 V <sub>SS</sub> 的电压 .....	-0.3V 至 +9.0V
所有其他引脚相对于 V <sub>SS</sub> 的电压 .....	-0.3V 至 (V <sub>DD</sub> + 0.3V)
总功耗 (†) .....	800 mW
V <sub>SS</sub> 引脚上的最大输出电流, -40°C ≤ T <sub>A</sub> ≤ +85°C (工业级) .....	255 mA
V <sub>SS</sub> 引脚上的最大输出电流, -40°C ≤ T <sub>A</sub> ≤ +125°C (扩展级) .....	105 mA
V <sub>DD</sub> 引脚上的最大输入电流, -40°C ≤ T <sub>A</sub> ≤ +85°C (工业级) .....	170 mA
V <sub>DD</sub> 引脚上的最大输入电流, -40°C ≤ T <sub>A</sub> ≤ +125°C (扩展级) .....	70 mA
输入钳位电流 I <sub>K</sub> (V <sub>PIN</sub> < 0 或 V <sub>PIN</sub> > V <sub>DD</sub> ) .....	± 20 mA
任一 I/O 引脚的最大输出灌电流 .....	25 mA
任一 I/O 引脚的最大输出拉电流 .....	25 mA

**注 1:** 功耗用以下公式计算:  $P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$ 。

†注: 如果器件工作条件超过上述“绝对最大值”, 就可能会对器件造成永久性损坏。上述值仅为运行条件极大值, 我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下, 其稳定性会受到影响。

# PIC16(L)F1933

图 30-1: PIC16(L)F1933 电压 — 频率关系图,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

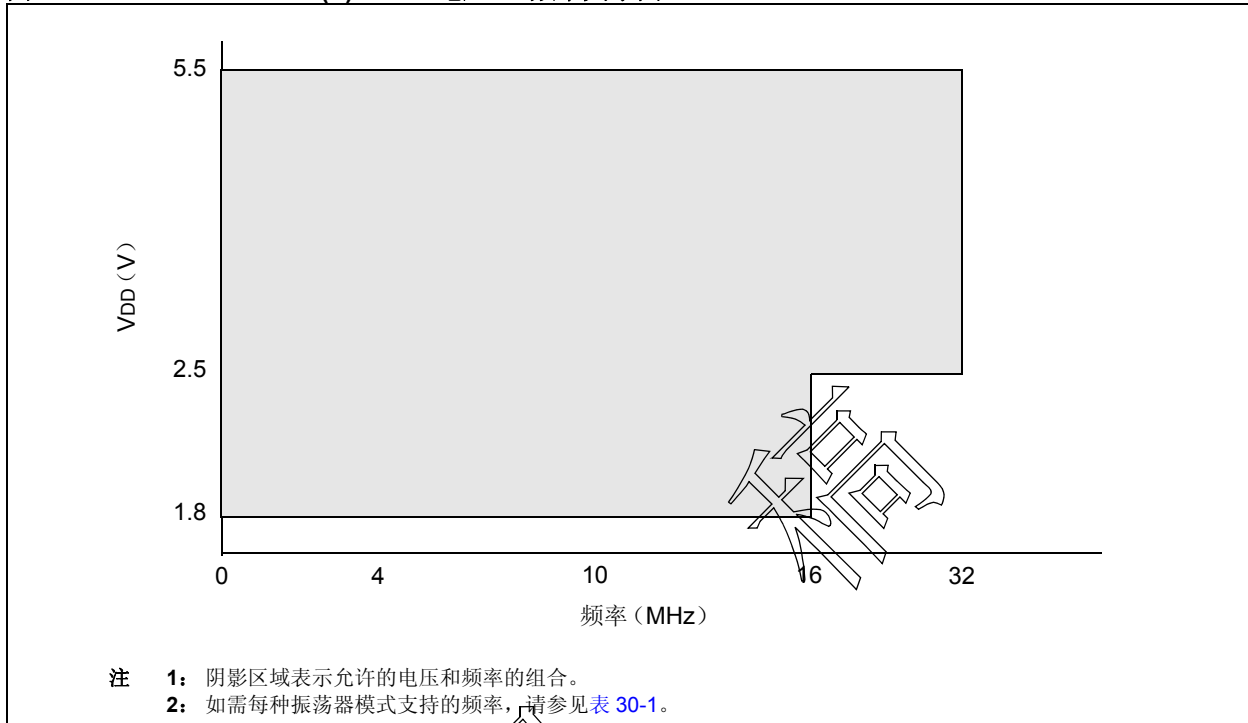


图 30-2: PIC16(L)F1933 电压 — 频率关系图,  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

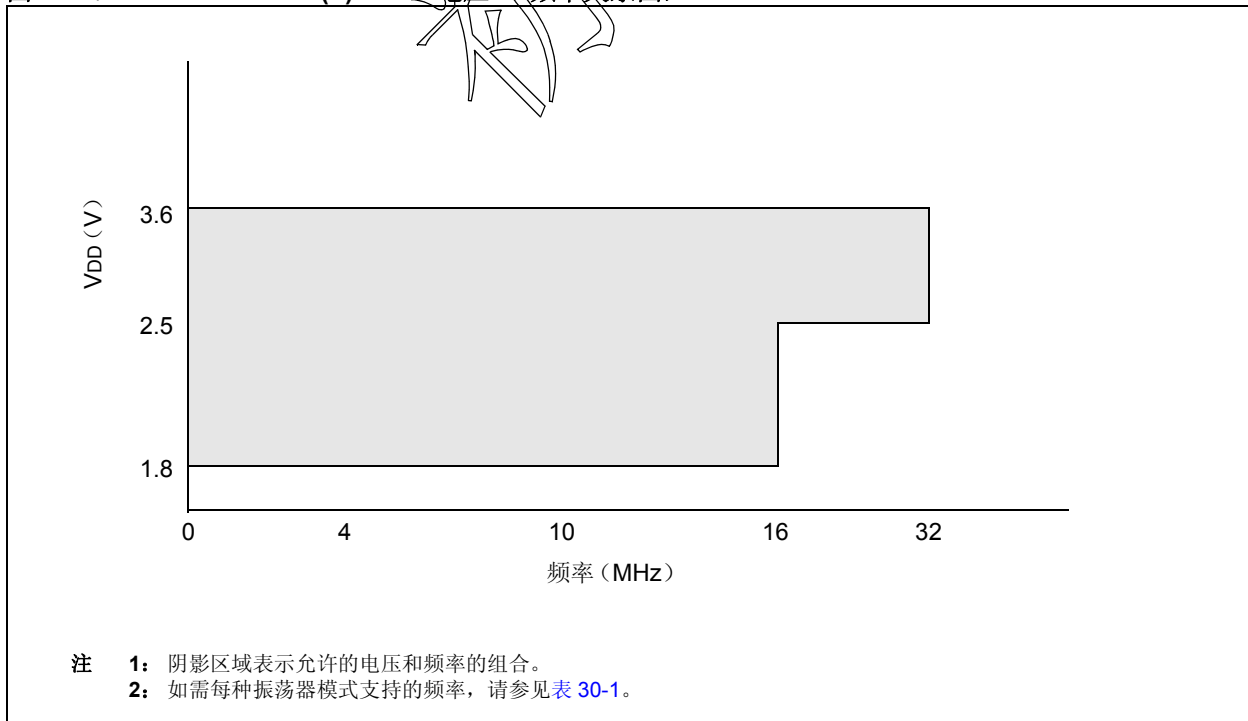
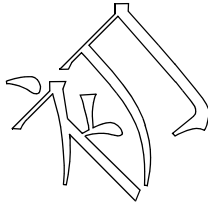
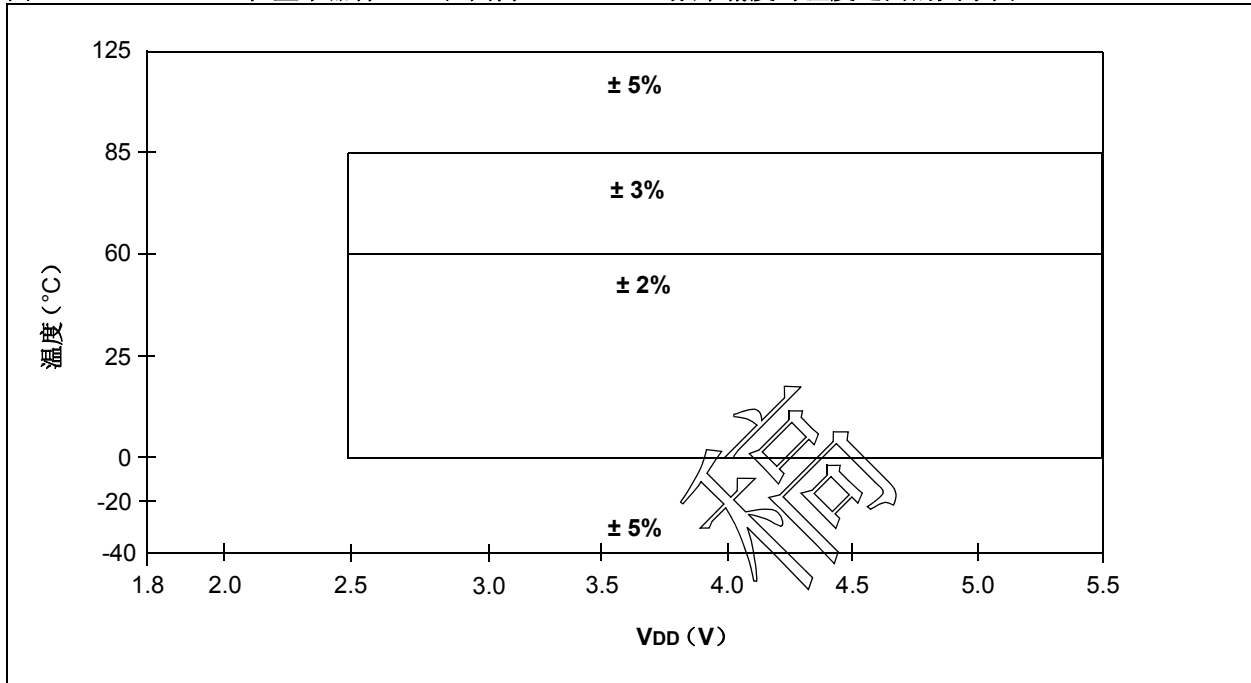




图 30-3: 在整个器件 VDD 范围内 HFINTOSC 频率精度与温度之间的关系图



# PIC16(L)F1933

## 30.1 直流特性: PIC16(L)F1933-I/E (工业级, 扩展级)

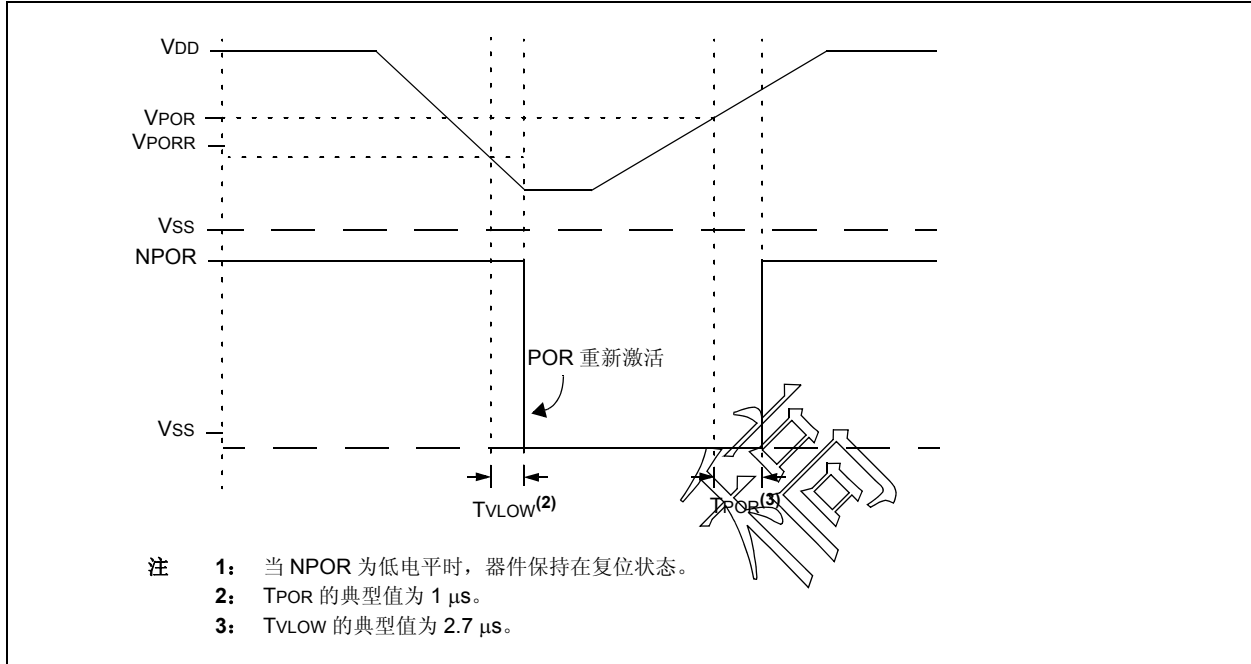
PIC16LF1933		标准工作条件 (除非另外声明)					
		工作温度					
		-40°C ≤ TA ≤ +85°C (工业级)					
		-40°C ≤ TA ≤ +125°C (扩展级)					
PIC16F1933		标准工作条件 (除非另外声明)					
		工作温度					
		-40°C ≤ TA ≤ +85°C (工业级)					
		-40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D001	VDD	电源电压					
		PIC16LF1933	1.8	—	3.6	V	Fosc ≤ 16 MHz
			2.3	—	3.6	V	Fosc ≤ 32 MHz (注 2)
D001		PIC16F1933	1.8	—	5.5	V	Fosc ≤ 16 MHz
			2.3	—	5.5	V	Fosc ≤ 32 MHz (注 2)
D002*	VDR	RAM 数据保持电压 <sup>(1)</sup>					
		PIC16LF1933	1.5	—	—	V	器件工作在休眠模式下
D002*		PIC16F1933	1.7	—	—	V	器件工作在休眠模式下
	VPOR*	上电复位释放电压	—	1.6	—	V	
	VPORR*	上电复位重新激活电压					
		PIC16LF1933	—	0.8	—	V	器件工作在休眠模式下
		PIC16F1933	—	1.7	—	V	器件工作在休眠模式下
D003	VADFVR	ADC 的固定参考电压	-8	—	6	%	1.024V, VDD ≥ 2.5V 2.048V, VDD ≥ 2.5V 4.096V, VDD ≥ 4.75V
D003A	VCDAFVR	比较器和 DAC 的固定参考电压	-11	—	7	%	1.024V, VDD ≥ 2.5V 2.048V, VDD ≥ 2.5V 4.096V, VDD ≥ 4.75V
D003B	VLCDFVR	LCD 偏置的固定参考电压	-11	—	10	%	3.072V, VDD ≥ 3.6V, 85°C
D004*	SVDD	VDD 上升速率确保能够产生内部上电复位信号	0.05	—	—	V/ms	详情请参见第 6.1 节“上电复位 (POR)”。

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.3V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

- 注
- 1: 该电压是休眠模式下保证不丢失 RAM 数据的最小 VDD。
  - 2: PLL 用于 32 MHz 操作。
  - 3: 为确保正常工作, ADC 正参考电压的最小值必须大于等于 1.8V。选择 FVR 或 VREF+ 引脚作为 ADC 正参考电压源时, 注意电压必须大于等于 1.8V。

图 30-4: VDD 缓慢上升时的 POR 和 POR 重新激活



## 30.2 直流特性: PIC16(L)F1933/4/E (工业级, 扩展级)

PIC16(L)F1933/4/E		标准工作条件 (除非另外声明)		工作温度			
PIC16LF1933		-40°C ≤ TA ≤ +85°C (工业级)		-40°C ≤ TA ≤ +125°C (扩展级)			
PIC16F1933		标准工作条件 (除非另外声明)		工作温度			
PIC16F1933		-40°C ≤ TA ≤ +85°C (工业级)		-40°C ≤ TA ≤ +125°C (扩展级)			
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
供电电流 (IDD) (1, 2)							
D009	LDO 稳压器	—	350	—	μA	—	HS、EC 或 INTOSC/INTOSCIO (8-16 MHz) 时钟模式 (禁止所有 VCAP 引脚)
		—	50	—	μA	—	禁止所有 VCAP 引脚
		—	30	—	μA	—	在 RA0、RA5 或 RA6 上使能 VCAP
		—	5	—	μA	—	LP 时钟模式和休眠模式 (需要禁止 FVR 和 BOR)
D010		—	8.0	14	μA	1.8	Fosc = 32 kHz
		—	12	18	μA	3.0	LP 振荡器模式 (注 4) -40°C ≤ TA ≤ +85°C

图注: TBD = 待定

- 注 1: 有效工作模式下, 所有 IDD 测量的测试条件为: OSC1 = 外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 上拉至 VDD; MCLR = VDD; 禁止 WDT。
- 注 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。
- 注 3: 对于 RC 振荡器配置, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式  $I_R = V_{DD}/2R_{EXT}$  (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 注 4: 禁止 FVR 和 BOR。
- 注 5: VCAP (RA0) 上的电容为 0.1 μF。
- 注 6: 使能了具有 4x PLL 的 8 MHz 晶振。

# PIC16(L)F1933

## 30.2 直流特性: PIC16(L)F1933-I/E (工业级, 扩展级) (续)

PIC16LF1933		标准工作条件 (除非另外声明)						
		工作温度					-40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
PIC16F1933		标准工作条件 (除非另外声明)						
		工作温度					-40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件		
						VDD	注	
D010		—	23	63	μA	1.8	Fosc = 32 kHz LP 振荡器模式 (注 4 和注 5) -40°C ≤ TA ≤ +85°C	
		—	28	74	μA	3.0		
		—	33	79	μA	5.0		
D010A		—	10	18	μA	1.8	Fosc = 32 kHz LP 振荡器模式 (注 4) 40°C ≤ TA ≤ +125°C	
		—	15	20	μA	3.0		
D010A		—	24	79	μA	1.8	Fosc = 32 kHz LP 振荡器模式 (注 4 和注 5) -40°C ≤ TA ≤ +125°C	
		—	30	93	μA	3.0		
		—	35	99	μA	5.0		
D011		—	140	180	μA	1.8	Fosc = 1 MHz XT 振荡器模式	
		—	200	300	μA	3.0		
D011		—	140	200	μA	1.8	Fosc = 1 MHz XT 振荡器模式 (注 5)	
		—	210	350	μA	3.0		
		—	350	470	μA	5.0		
D012		—	300	400	μA	1.8	Fosc = 4 MHz XT 振荡器模式	
		—	500	700	μA	3.0		
D012		—	320	550	μA	1.8	Fosc = 4 MHz XT 振荡器模式 (注 5)	
		—	570	800	μA	3.0		
		—	700	960	μA	5.0		

图注: TBD = 待定

- 注 1: 有效工作模式下, 所有 IDD 测量的测试条件为: OSC1 = 外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 上拉至 VDD; MCLR = VDD; 禁止 WDT。
- 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。
- 3: 对于 RC 振荡器配置, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式  $I_R = V_{DD}/2R_{EXT}$  (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 禁止 FVR 和 BOR。
- 5: VCAP (RA0) 上的电容为 0.1 μF。
- 6: 使能了具有 4x PLL 的 8 MHz 晶振。

## 30.2 直流特性: PIC16(L)F1933-I/E (工业级, 扩展级) (续)

PIC16LF1933		标准工作条件 (除非另外声明)					
		工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
PIC16F1933		标准工作条件 (除非另外声明)					
		工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件	
						VDD	注
D013	供电电流 (IDD) (1, 2)	—	140	—	μA	1.8	Fosc = 500 kHz
		—	320	—	μA	3.0	EC 振荡器低功耗模式
D013		—	160	—	μA	1.8	Fosc = 500 kHz
		—	340	—	μA	3.0	EC 振荡器低功耗模式 (注 5)
		—	390	—	μA	5.0	
D014		—	230	380	μA	1.8	Fosc = 4 MHz
		—	480	650	μA	3.0	EC 振荡器模式 中等功耗模式
D014		—	250	430	μA	1.8	Fosc = 4 MHz
		—	500	730	μA	3.0	EC 振荡器模式 (注 5)
		—	600	830	μA	5.0	中等功耗模式
D015		—	3.4	—	mA	3.0	Fosc = 32 MHz
		—	4.1	—	mA	3.6	EC 振荡器高功耗模式
D015		—	3.6	—	mA	3.0	Fosc = 32 MHz
		—	3.9	—	mA	5.0	EC 振荡器高功耗模式 (注 5)
D016		—	7.0	10	μA	1.8	Fosc = 32 kHz
		—	10	13	μA	3.0	LFINTOSC 模式, 85°C
D016		—	21	61	μA	1.8	Fosc = 32 kHz
		—	27	74	μA	3.0	LFINTOSC 模式, 85°C (注 5)
		—	28	76	μA	5.0	

图注: TBD = 待定

- 注 1: 有效工作模式下, 所有 IDD 测量的测试条件为: OSC1 = 外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 上拉至 VDD; MCLR = VDD; 禁止 WDT。
- 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。
- 3: 对于 RC 振荡器配置, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式  $I_R = V_{DD}/2R_{EXT}$  (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 禁止 FVR 和 BOR。
- 5: VCAP (RA0) 上的电容为 0.1 μF。
- 6: 使能了具有 4x PLL 的 8 MHz 晶振。

# PIC16(L)F1933

## 30.2 直流特性: PIC16(L)F1933-I/E (工业级, 扩展级) (续)

PIC16LF1933		标准工作条件 (除非另外声明)						
		工作温度					-40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
PIC16F1933		标准工作条件 (除非另外声明)						
		工作温度					-40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)	
参数编号	器件特性	最小值	典型值 †	最大值	单位	条件		
						VDD	注	
D017	供电电流 (IDD) (1, 2)	—	130	170	μA	1.8	Fosc = 500 kHz	
		—	190	220	μA	3.0	MFINTOSC 模式	
D017		—	150	200	μA	1.8	Fosc = 500 kHz	
		—	210	270	μA	3.0	MFINTOSC 模式 (注 5)	
		—	270	330	μA	5.0		
D018		—	0.8	1.1	mA	1.8	Fosc = 8 MHz	
		—	1.3	1.7	mA	3.0	HFINTOSC 模式	
D018		—	0.9	1.2	mA	1.8	Fosc = 8 MHz	
		—	1.4	1.9	mA	3.0	HFINTOSC 模式 (注 5)	
		—	1.6	2.2	mA	5.0		
D019		—	1.3	1.8	mA	1.8	Fosc = 16 MHz	
		—	2.0	2.5	mA	3.0	HFINTOSC 模式	
D019		—	1.4	2.0	mA	1.8	Fosc = 16 MHz	
		—	2.2	2.8	mA	3.0	HFINTOSC 模式 (注 5)	
		—	2.4	3.0	mA	5.0		
D020		—	TBD	TBD	mA	3.0	Fosc = 32 MHz	
		—	TBD	TBD	mA	3.6	HFINTOSC 模式	
D020		—	TBD	TBD	mA	3.0	Fosc = 32 MHz	
		—	TBD	TBD	mA	5.0	HFINTOSC 模式	
D021		—	300	450	μA	1.8	Fosc = 4 MHz	
		—	500	700	μA	3.0	EXTRC 模式 (注 3)	
D021		—	350	600	μA	1.8	Fosc = 4 MHz	
		—	550	900	μA	3.0	EXTRC 模式 (注 3 和注 5)	
		—	620	1000	μA	5.0		
D022		—	3.4	—	mA	3.0	Fosc = 32 MHz	
		—	4.2	—	mA	3.6	HS 振荡器模式 (注 6)	
D022		—	3.6	—	mA	3.0	Fosc = 32 MHz	
		—	3.9	—	mA	5.0	HS 振荡器模式 (注 5 和注 6)	

图注: TBD = 待定

- 注 1: 有效工作模式下, 所有 IDD 测量的测试条件为: OSC1 = 外部方波, 轨到轨满幅; 所有 I/O 引脚均为三态, 上拉至 VDD; MCLR = VDD; 禁止 WDT。
- 2: 供电电流主要受工作电压和频率的影响。其他因素如 I/O 引脚负载和开关速率、振荡器类型、内部代码执行模式以及温度也对电流消耗有影响。
- 3: 对于 RC 振荡器配置, 该电流不包括流经 REXT 的电流。流经该电阻的电流可以由公式  $I_R = V_{DD}/2R_{EXT}$  (mA) 来估算, 其中 REXT 的单位是 kΩ。
- 4: 禁止 FVR 和 BOR。
- 5: VCAP (RA0) 上的电容为 0.1 μF。
- 6: 使能了具有 4x PLL 的 8 MHz 晶振。

## 30.3 直流特性: PIC16(L)F1933-I/E (掉电)

PIC16LF1933		标准工作条件 (除非另外声明)						
		工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)						
PIC16F1933		标准工作条件 (除非另外声明)						
		工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)						
参数编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件	
							VDD	注
基本掉电电流 (IPD) (2)								
D023		—	0.06	1	7	μA	1.8	禁止 WDT、BOR、FVR 和 T1OSC, 所有外设不工作
		—	0.08	2	8	μA	3.0	
D023		—	15	55	69	μA	1.8	禁止 WDT、BOR、FVR 和 T1OSC, 所有外设不工作
		—	18	58	72	μA	3.0	
		—	19	60	75	μA	5.0	
D024		—	0.5	6	8	μA	1.8	LPWDT 电流 (注 1)
		—	0.8	7	9	μA	3.0	
D024		—	16	56	70	μA	1.8	LPWDT 电流 (注 1)
		—	19	59	73	μA	3.0	
		—	20	61	76	μA	5.0	
D025		—	8.5	20	20	μA	1.8	FVR 电流
		—	8.5	23	23	μA	3.0	
D025		—	32	96	120	μA	1.8	FVR 电流 (注 4)
		—	39	110	140	μA	3.0	
		—	70	140	170	μA	5.0	
D026		—	7.5	17	17	μA	3.0	BOR 电流 (注 1)
D026		—	34	96	120	μA	3.0	BOR 电流 (注 1 和注 4)
		—	67	130	170	μA	5.0	
D027		—	0.6	5	5	μA	1.8	T1OSC 电流 (注 1)
		—	1.8	9	9.5	μA	3.0	
D027		—	16	57	71	μA	1.8	T1OSC 电流 (注 1)
		—	21	62	78	μA	3.0	
		—	25	66	83	μA	5.0	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

图注: TBD = 待定

- 注 1: 外设电流是基本 IDD 或 IPD 电流以及该外设使能时额外消耗的电流之和。外设 Δ 电流可以从该外设电流中减去基本 IDD 或 IPD 电流而得出。在计算总电流消耗时应使用最大值。
- 2: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式时, 所有 I/O 引脚处于高阻态并连接到 VDD 时测得的。
- 3: A/D 振荡器的时钟源为 FRC。
- 4: VCAP (RA0) 上的电容为 0.1 μF。

# PIC16(L)F1933

## 30.3 直流特性: PIC16(L)F1933-I/E (掉电) (续)

PIC16LF1933		标准工作条件 (除非另外声明)							
		工作温度							
		-40°C ≤ TA ≤ +85°C (工业级)							
		-40°C ≤ TA ≤ +125°C (扩展级)							
PIC16F1933		标准工作条件 (除非另外声明)							
		工作温度							
		-40°C ≤ TA ≤ +85°C (工业级)							
		-40°C ≤ TA ≤ +125°C (扩展级)							
参数编号	器件特性	最小值	典型值 †	最大值 +85°C	最大值 +125°C	单位	条件		
							VDD	注	
基本掉电电流 (IPD) (2)									
D028		—	0.1	1	7	μA	1.8	A/D 电流 (注 1 和注 3), 转换未进行	
		—	0.1	2	8	μA	3.0		
D028		—	16	56	70	μA	1.8	A/D 电流 (注 1 和注 3), 转换未进行	
		—	21	58	73	μA	3.0		
		—	25	61	76	μA	5.0		
D029		—	250	—	—	μA	1.8	A/D 电流 (注 1 和注 3), 转换正在进行	
		—	250	—	—	μA	3.0		
D029		—	280	—	—	μA	1.8	A/D 电流 (注 1、注 3 和注 4), 转换正在进行	
		—	280	—	—	μA	3.0		
		—	280	—	—	μA	5.0		
D030		—	3.5	6	6	μA	1.8	电容触摸传感, 低功耗模式	
		—	7	9	9	μA	3.0		
D030		—	17	61	76	μA	1.8	电容触摸传感, 低功耗模式	
		—	21	64	79	μA	3.0		
		—	22	66	83	μA	5.0		
D031		—	1	—	—	μA	3.0	LCD 偏置梯形电阻, 低功耗	
		—	10	—	—	μA	3.0		
		—	75	—	—	μA	3.0		
D031		—	1	—	—	μA	5.0	LCD 偏置梯形电阻, 低功耗	
		—	10	—	—	μA	5.0		
		—	75	—	—	μA	5.0		
D032		—	7.6	18	18	μA	1.8	比较器, 低功耗模式	
		—	8.0	20	20	μA	3.0		
D032		—	24	65	81	μA	1.8	比较器, 低功耗模式	
		—	26	75	94	μA	3.0		
		—	28	76	95	μA	5.0		

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

图注: TBD = 待定

- 注 1: 外设电流是基本 IDD 或 IPD 电流以及该外设使能时额外消耗的电流之和。外设 Δ 电流可以从该外设电流中减去基本 IDD 或 IPD 电流而得出。在计算总电流消耗时应使用最大值。
- 2: 休眠模式下的掉电电流与振荡器类型无关。掉电电流是在器件处于休眠模式时, 所有 I/O 引脚处于高阻态并连接到 VDD 时测得的。
- 3: A/D 振荡器的时钟源为 FRC。
- 4: VCAP (RA0) 上的电容为 0.1 μF。



## 30.4 直流特性: PIC16(L)F1933-I/E

直流特性		标准工作条件 (除非另外声明)					
		工作温度 -40°C ≤ TA ≤ +85°C (工业级) -40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
D032 D032A D033 D034 D034A	V <sub>IL</sub>	<b>输入低电压</b>					
		I/O 端口:					
		带 TTL 缓冲器	—	—	0.8	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V
		带施密特触发缓冲器	—	—	0.15 V <sub>DD</sub>	V	1.8V ≤ V <sub>DD</sub> ≤ 4.5V
		带 I <sup>2</sup> C™ 电平	—	—	0.2 V <sub>DD</sub>	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V
		带 SMBus 电平	—	—	0.8	V	2.7V ≤ V <sub>DD</sub> ≤ 5.5V
		MCLR 和 OSC1 (RC 模式) (1)	—	—	0.2 V <sub>DD</sub>	V	
D040 D040A D041 D042 D043A D043B	V <sub>IH</sub>	<b>输入高电压</b>					
I/O 端口:							
带 TTL 缓冲器		2.0	—	—	V	4.5V ≤ V <sub>DD</sub> ≤ 5.5V	
带施密特触发缓冲器		0.25 V <sub>DD</sub> + 0.8	—	—	V	1.8V ≤ V <sub>DD</sub> ≤ 4.5V	
带 I <sup>2</sup> C™ 电平		0.8 V <sub>DD</sub>	—	—	V	2.0V ≤ V <sub>DD</sub> ≤ 5.5V	
带 SMBus 电平		0.7 V <sub>DD</sub>	—	—	V		
MCLR		2.1	—	—	V	2.7V ≤ V <sub>DD</sub> ≤ 5.5V	
OSC1 (HS 模式)	0.8 V <sub>DD</sub>	—	—	V			
OSC1 (RC 模式)	0.7 V <sub>DD</sub>	—	—	V			
D060 D061	I <sub>IL</sub>	<b>输入泄漏电流 (2)</b>					
I/O 端口		—	± 5	± 125	nA	V <sub>SS</sub> ≤ V <sub>PIN</sub> ≤ V <sub>DD</sub> , 引脚处于高阻态 (85°C 时)	
D061	MCLR (3)	—	± 5	± 1000	nA	125°C	
D070*	I <sub>PUR</sub>	<b>弱上拉电流</b>					
		25 25	100 140	200 300	μA	V <sub>DD</sub> = 3.3V, V <sub>PIN</sub> = V <sub>SS</sub> V <sub>DD</sub> = 5.0V, V <sub>PIN</sub> = V <sub>SS</sub>	
D080	V <sub>OL</sub>	<b>输出低电压 (4)</b>					
I/O 端口		—	—	0.6	V	I <sub>OL</sub> = 8mA, V <sub>DD</sub> = 5V I <sub>OL</sub> = 6mA, V <sub>DD</sub> = 3.3V I <sub>OL</sub> = 1.8mA, V <sub>DD</sub> = 1.8V	
D090	V <sub>OH</sub>	<b>输出高电压 (4)</b>					
I/O 端口		V <sub>DD</sub> - 0.7	—	—	V	I <sub>OH</sub> = 3.5mA, V <sub>DD</sub> = 5V I <sub>OH</sub> = 3mA, V <sub>DD</sub> = 3.3V I <sub>OH</sub> = 1mA, V <sub>DD</sub> = 1.8V	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

注 1: 在 RC 振荡器配置中, OSC1/CLKIN 引脚被配置为施密特触发器输入引脚。在 RC 模式下, 建议不要使用外部时钟。

2: 负电流定义为引脚的拉电流。

3: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。

4: 包括 CLKOUT 模式下的 OSC2 引脚。

# PIC16(L)F1933

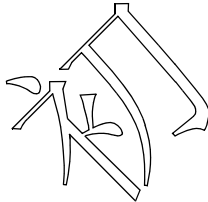
## 30.4 直流特性: PIC16(L)F1933-I/E (续)

直流特性		标准工作条件 (除非另外声明)					条件
		工作温度					
		-40°C ≤ TA ≤ +85°C (工业级)					
		-40°C ≤ TA ≤ +125°C (扩展级)					
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
<b>输出引脚上的容性负载规范</b>							
D101*	COSC2	OSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时处于 XT、HS 和 LP 模式
D101A*	Cio	所有 I/O 引脚	—	—	50	pF	
<b>Vcap 电容充电</b>							
D102		充电电流	—	200	—	μA	
D102A		充电结束时的灌拉能力	—	0.0	—	mA	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

- 注
- 1: 在 RC 振荡器配置中, OSC1/CLKIN 引脚被配置为施密特触发器输入引脚。在 RC 模式下, 建议不要使用外部时钟。
  - 2: 负电流定义为引脚的拉电流。
  - 3: MCLR 引脚上的泄漏电流主要取决于所施加的电压。规定电压为正常工作条件下的电压。在不同的输入电压下可能测得更高的泄漏电流。
  - 4: 包括 CLKOUT 模式下的 OSC2 引脚。



## 30.5 存储器编程要求

直流特性		标准工作条件（除非另外声明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$					
参数编号	符号	特性	最小值	典型值 <sup>†</sup>	最大值	单位	条件
		<b>程序存储器编程规范</b>					
D110	VIHH	MCLR/VPP/RE3 引脚上的电压	8.0	—	9.0	V	(注 3 和注 4)
D111	IDDP	编程期间的供电电流	—	—	10	mA	
D112		用于块擦除的 VDD	2.7	—	VDD 最大值	V	
D113	VPEW	用于写或行擦除的 VDD	VDD 最小值	—	VDD 最大值	V	
D114	IPPPGM	擦 / 写期间 MCLR/VPP 上的电流	—	—	10	mA	
D115	IDDPGM	擦 / 写期间 VDD 上的电流	—	—	5.0	mA	
		<b>数据 EEPROM 存储器</b>					
D116	ED	字节耐用性	100K	—	—	E/W	-40°C 至 +85°C
D117	VDRW	用于读 / 写操作的 VDD	VDD 最小值	—	VDD 最大值	V	
D118	TDEW	擦 / 写周期	—	4.0	5.0	ms	
D119	TRETD	特性保持时间	—	40	—	年	假设没有违反其他规范
D120	TREF	刷新前的总擦 / 写周期数 <sup>(2)</sup>	1M	10M	—	E/W	-40°C 至 +85°C
		<b>闪存程序存储器</b>					
D121	EP	电池耐用性	10K	—	—	E/W	-40°C 至 +85°C (注 1)
D122	VPR	用于读操作的 VDD	VDD 最小值	—	VDD 最大值	V	
D123	TIW	自定时的写周期	—	2	2.5	ms	
D124	TRETD	特性保持时间	—	40	—	年	假设没有违反其他规范

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考，未经测试。

注 1: 自写和块擦除。

2: 关于数据 EEPROM 耐用性的更多详细讨论，请参见第 11.2 节“使用数据 EEPROM”。

3: 仅在禁止单电源编程的情况下需要。

4: MPLAB ICD 2 不支持变化的 VPP 输出。当使用 ICD 2 进行编程或调试时，用来限制 ICD 2 VPP 电压的电路必须放在 ICD 2 和目标系统之间。

# PIC16(L)F1933

## 30.6 散热考虑

标准工作条件（除非另外声明）

工作温度  $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$

参数编号	符号	特性	典型值	单位	条件
TH01	$\theta_{JA}$	结点到环境的热阻	60	$^{\circ}\text{C}/\text{W}$	28 引脚 SPDIP 封装
			80	$^{\circ}\text{C}/\text{W}$	28 引脚 SOIC 封装
			90	$^{\circ}\text{C}/\text{W}$	28 引脚 SSOP 封装
			27.5	$^{\circ}\text{C}/\text{W}$	28 引脚 UQFN 4x4 mm 封装
			27.5	$^{\circ}\text{C}/\text{W}$	28 引脚 QFN 6x6 mm 封装
			47.2	$^{\circ}\text{C}/\text{W}$	40 引脚 PDIP 封装
			46	$^{\circ}\text{C}/\text{W}$	44 引脚 TQFP 封装
			24.4	$^{\circ}\text{C}/\text{W}$	44 引脚 QFN 8x8 mm 封装
TH02	$\theta_{JC}$	结点到外壳的热阻	31.4	$^{\circ}\text{C}/\text{W}$	28 引脚 SPDIP 封装
			24	$^{\circ}\text{C}/\text{W}$	28 引脚 SOIC 封装
			24	$^{\circ}\text{C}/\text{W}$	28 引脚 SSOP 封装
			24	$^{\circ}\text{C}/\text{W}$	28 引脚 UQFN 4x4 mm 封装
			24	$^{\circ}\text{C}/\text{W}$	28 引脚 QFN 6x6 mm 封装
			24.7	$^{\circ}\text{C}/\text{W}$	40 引脚 PDIP 封装
			14.5	$^{\circ}\text{C}/\text{W}$	44 引脚 TQFP 封装
			20	$^{\circ}\text{C}/\text{W}$	44 引脚 QFN 8x8 mm 封装
TH03	$T_{JMAX}$	最高结温	150	$^{\circ}\text{C}$	
TH04	PD	功耗	—	W	$PD = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}^{(1)}$
TH06	$P_{I/O}$	I/O 功耗	—	W	$P_{I/O} = \sum (I_{OL} \times V_{OL}) + \sum (I_{OH} \times (V_{DD} - V_{OH}))$
TH07	$P_{DER}$	降额功耗	—	W	$P_{DER} = PD_{MAX} (T_J - T_A) / \theta_{JA}^{(2)}$

注 1:  $I_{DD}$  为单独运行芯片而不驱动输出引脚上的任何负载时的电流。

2:  $T_A$  = 环境温度

3:  $T_J$  = 结温

## 30.7 时序参数符号

可根据以下任一格式来创建时序参数符号：

1. TppS2ppS
2. TppS

<b>T</b>		<b>T</b>	时间
<b>F</b>	频率		

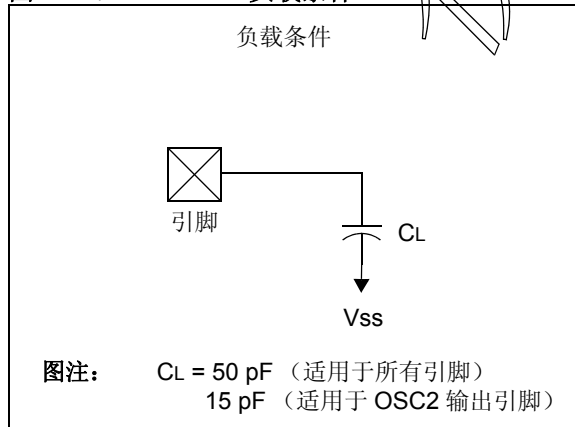
小写字母 (pp) 及其含义：

<b>pp</b>			
cc	CCP1	osc	OSC1
ck	CLKOUT	rd	$\overline{RD}$
cs	$\overline{CS}$	rw	$\overline{RD}$ 或 $\overline{WR}$
di	SDI	sc	SCK
do	SDO	ss	$\overline{SS}$
dt	数据输入	t0	T0CKI
io	I/O 端口	t1	T1CKI
mc	$\overline{MCLR}$	wr	$\overline{WR}$

大写字母及其含义：

<b>S</b>		<b>P</b>	周期
<b>F</b>	下降	<b>R</b>	上升
<b>H</b>	高	<b>V</b>	有效
<b>I</b>	无效 (高阻态)	<b>Z</b>	高阻态
<b>L</b>	低		

图 30-5: 负载条件



# PIC16(L)F1933

## 30.8 交流特性: PIC16(L)F1933-I/E

图 30-6: 时钟时序

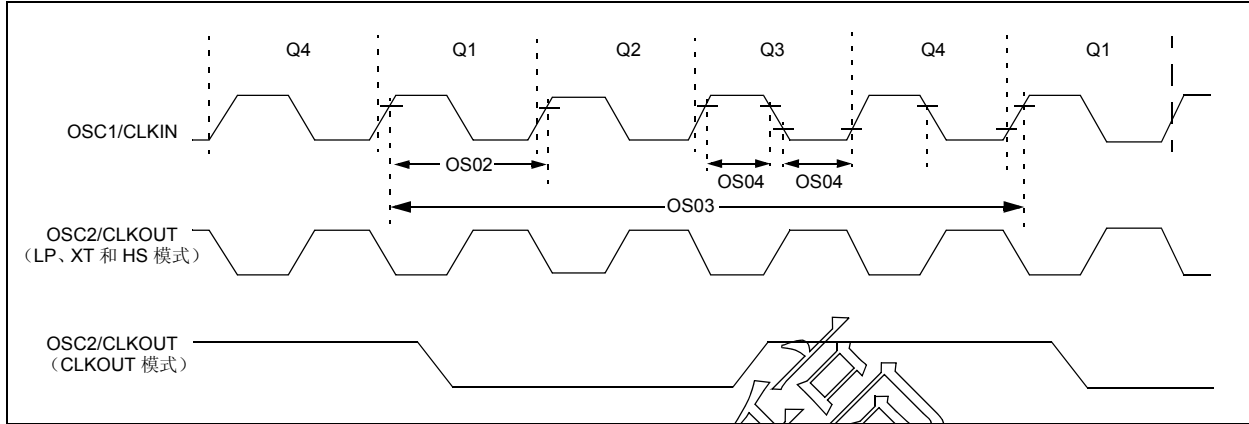


表 30-1: 时钟振荡器时序要求

标准工作条件 (除非另外声明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS01	Fosc	外部 CLKIN 频率 (1)	DC	—	0.5	MHz	EC 振荡器模式 (低)
			DC	—	4	MHz	EC 振荡器模式 (中等)
			DC	—	20	MHz	EC 振荡器模式 (高)
	振荡器频率 (1)	—	32.768	—	—	kHz	LP 振荡器模式
		0.1	—	4	—	MHz	XT 振荡器模式
		1	—	4	—	MHz	HS 振荡器模式
1		—	20	—	MHz	HS 振荡器模式, $V_{DD} > 2.7\text{V}$	
OS02	Tosc	外部 CLKIN 周期 (1)	27	—	$\infty$	$\mu\text{s}$	LP 振荡器模式
			250	—	$\infty$	ns	XT 振荡器模式
			50	—	$\infty$	ns	HS 振荡器模式
			50	—	$\infty$	ns	EC 振荡器模式
	振荡器周期 (1)	—	30.5	—	—	$\mu\text{s}$	LP 振荡器模式
250		—	10,000	—	ns	XT 振荡器模式	
50		—	1,000	—	ns	HS 振荡器模式	
250		—	—	—	ns	RC 振荡器模式	
OS03	Tcy	指令周期 (1)	200	Tcy	DC	ns	$T_{CY} = 4/F_{osc}$
OS04*	TosH, TosL	外部 CLKIN 高电平时间, 外部 CLKIN 低电平时间	2	—	—	$\mu\text{s}$	LP 振荡器模式
			100	—	—	ns	XT 振荡器模式
			20	—	—	ns	HS 振荡器模式
OS05*	TosR, TosF	外部 CLKIN 上升时间, 外部 CLKIN 下降时间	0	—	$\infty$	ns	LP 振荡器模式
			0	—	$\infty$	ns	XT 振荡器模式
			0	—	$\infty$	ns	HS 振荡器模式

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

注 1: 指令周期 (Tcy) 等于输入振荡器时钟周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码时对应特定振荡器类型的特征数据。超出这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试“最小值”时, 都在 OSC1 引脚上连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC” (没有时钟)。

**表 30-2: 振荡器参数**

标准工作条件 (除非另外声明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性	频率容差	最小值	典型值 †	最大值	单位	条件
OS08	HFosc	经过校准的内部 HFINTOSC 频率 (2)	$\pm 2\%$	—	16.0	—	MHz	$0^{\circ}\text{C} \leq T_A \leq +60^{\circ}\text{C}, V_{DD} \geq 2.5\text{V}$ $60^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}, V_{DD} \geq 2.5\text{V}$ $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
			$\pm 3\%$	—	16.0	—	MHz	
			$\pm 5\%$	—	16.0	—	MHz	
OS08A	MFosc	经过校准的内部 MFINTOSC 频率 (2)	$\pm 2\%$	—	500	—	kHz	$0^{\circ}\text{C} \leq T_A \leq +60^{\circ}\text{C}, V_{DD} \geq 2.5\text{V}$ $60^{\circ}\text{C} \leq T_A \leq 85^{\circ}\text{C}, V_{DD} \geq 2.5\text{V}$ $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
			$\pm 3\%$	—	500	—	kHz	
			$\pm 5\%$	—	500	—	kHz	
OS09	LFosc	内部 LFINTOSC 频率	—	—	31	—	kHz	$-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$
OS10*	Tiosc ST	HFINTOSC 从休眠模式唤醒的起振时间	—	—	3.2	8	$\mu\text{s}$	
		MFINTOSC 从休眠模式唤醒的起振时间	—	—	24	85	$\mu\text{s}$	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为  $3.0\text{V}$ 、 $25^{\circ}\text{C}$  条件下的值。这些参数仅作为设计参考, 未经测试。

- 注 1:** 指令周期 ( $T_{cy}$ ) 等于输入振荡器时基周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码时对应特定振荡器类型的特征数据。超出这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试“最小值”时, 都在 OSC1 引脚上连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC” (没有时钟)。
- 2:** 为了确保振荡器频率的容差, 必须尽量靠近器件在  $V_{DD}$  和  $V_{SS}$  之间连接去耦电容。建议并联一个  $0.1 \mu\text{F}$  和  $0.01 \mu\text{F}$  的电容。
- 3:** 通过设计确保。

**表 30-3: PLL 时钟时序规范 ( $V_{DD} = 2.7\text{V}$  至  $5.5\text{V}$ )**

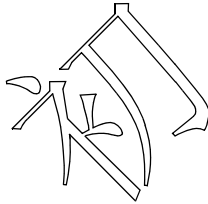
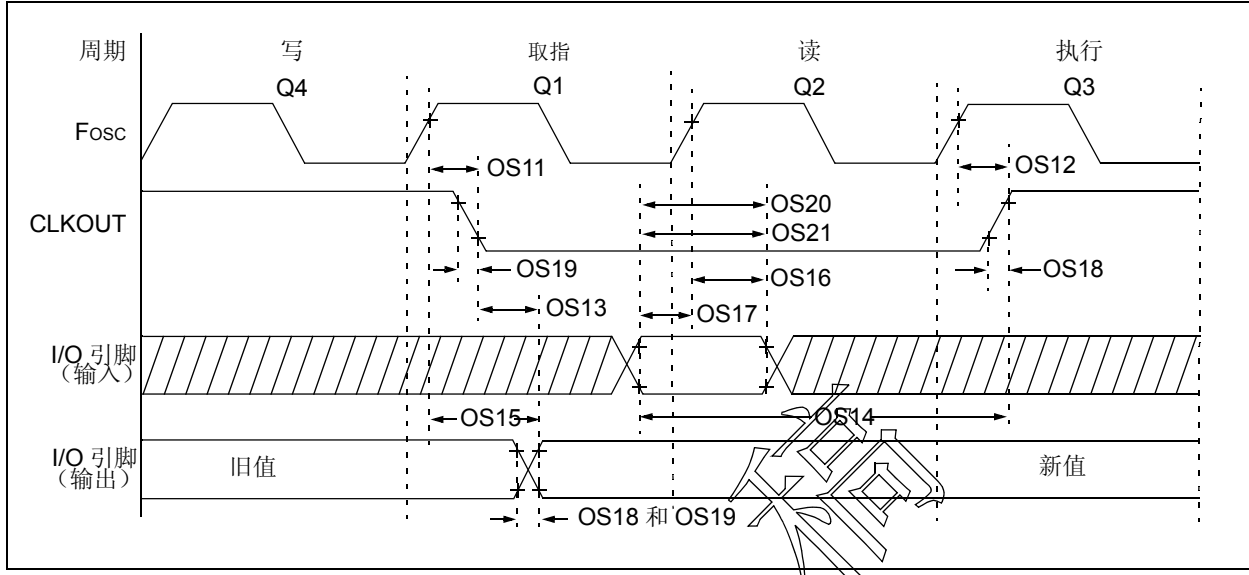
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
F10	FOSC	振荡器频率范围	4	—	8	MHz	
F11	FSYS	片内 VCO 系统频率	16	—	32	MHz	
F12	TRC	PLL 起振时间 (锁定时间)	—	—	2	ms	
F13*	$\Delta\text{CLK}$	CLKOUT 稳定性 (抗抖动性能)	-0.25%	—	+0.25%	%	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为  $5\text{V}$ 、 $25^{\circ}\text{C}$  条件下的值。这些参数仅作为设计参考, 未经测试。

# PIC16(L)F1933

图 30-7: CLKOUT 和 I/O 时序





**表 30-4: CLKOUT 和 I/O 时序参数**

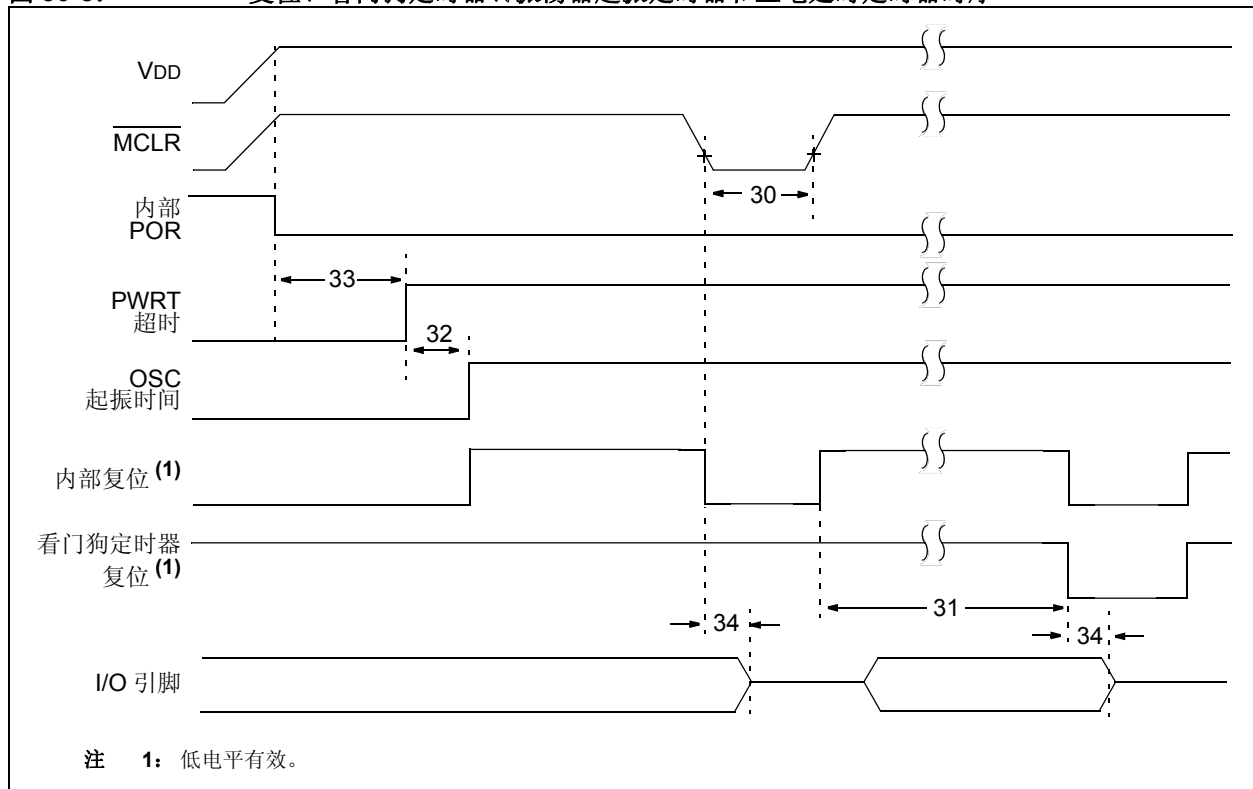
标准工作条件 (除非另外声明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
OS11	TosH2ckL	Fosc↑ 至 CLKOUT↓ 的时间 (1)	—	—	70	ns	VDD = 3.3-5.0V
OS12	TosH2ckH	Fosc↑ 至 CLKOUT↑ 的时间 (1)	—	—	72	ns	VDD = 3.3-5.0V
OS13	TckL2ioV	CLKOUT↓ 至端口输出有效的的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	CLKOUT↑ 前端口输入有效的的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ (Q1 周期) 至端口输出有效的的时间	—	50	70*	ns	VDD = 3.3-5.0V
OS16	TosH2ioI	Fosc↑ (Q2 周期) 至端口输入无效的时间 (I/O 输入保持时间)	50	—	—	ns	VDD = 3.3-5.0V
OS17	TioV2osH	端口输入有效至 Fosc↑ (Q2 周期) 的时间 (I/O 输入建立时间)	20	—	—	ns	
OS18	TioR	端口输出上升时间	— —	40 15	72 32	ns	VDD = 1.8V VDD = 3.3-5.0V
OS19	TioF	端口输出下降时间	— —	28 15	55 30	ns	VDD = 1.8V VDD = 3.3-5.0V
OS20*	Tinp	INT 引脚输入高电平时间或低电平时间	25	—	—	ns	
OS21*	Tioc	电平变化中断信号的新输入电平时间	25	—	—	ns	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。

注 1: 测量是在 RC 模式下进行的, 其中 CLKOUT 输出为  $4 \times T_{osc}$ 。

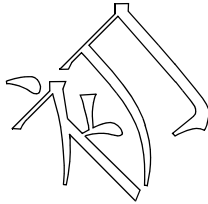
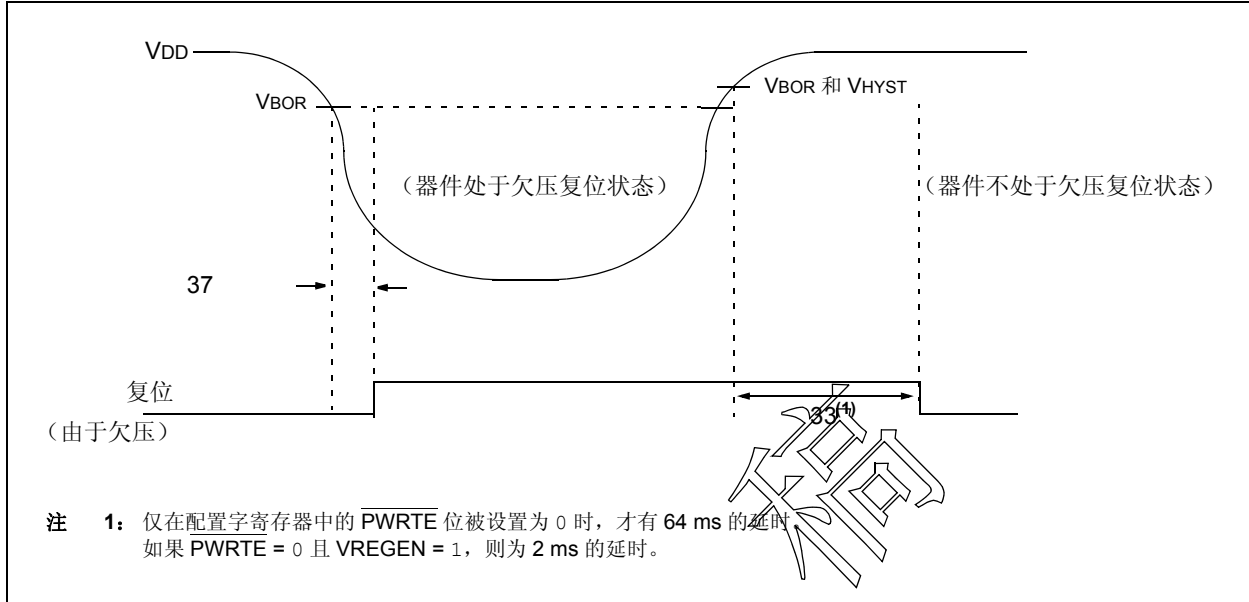
**图 30-8: 复位、看门狗定时器、振荡器起振定时器和上电延时定时器时序**



注 1: 低电平有效。

# PIC16(L)F1933

图 30-9: 欠压复位时序和特性



**表 30-5: 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数**

标准工作条件 (除非另外声明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	TMCL	MCLR 脉宽 (低电平)	2	—	—	$\mu\text{s}$	
31	TWDTLP	低功耗看门狗定时器超时周期	10	16	27	ms	$V_{DD} = 3.3\text{V}-5\text{V}$ 使用 1:16 预分频比
32	TOST	振荡器起振定时器周期 (1), (2)	—	1024	—	Tosc	(注 3)
33*	TPWRT	上电延时定时器周期, $\overline{\text{PWRTE}} = 0$	40	65	140	ms	
34*	TIOZ	自 MCLR 低电平或看门狗定时器复位起 I/O 处于高阻态的时间	—	—	2.0	$\mu\text{s}$	
35	VBOR	欠压复位电压	2.38 1.80	2.5 1.9	2.73 2.11	V	BORV=2.5V BORV=1.9V
36*	VHYST	欠压复位迟滞	0	25	60	mV	$-40^{\circ}\text{C}$ 至 $+85^{\circ}\text{C}$
37*	TBORDC	欠压复位直流响应时间	1	3	35	$\mu\text{s}$	$V_{DD} \leq V_{BOR}$

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

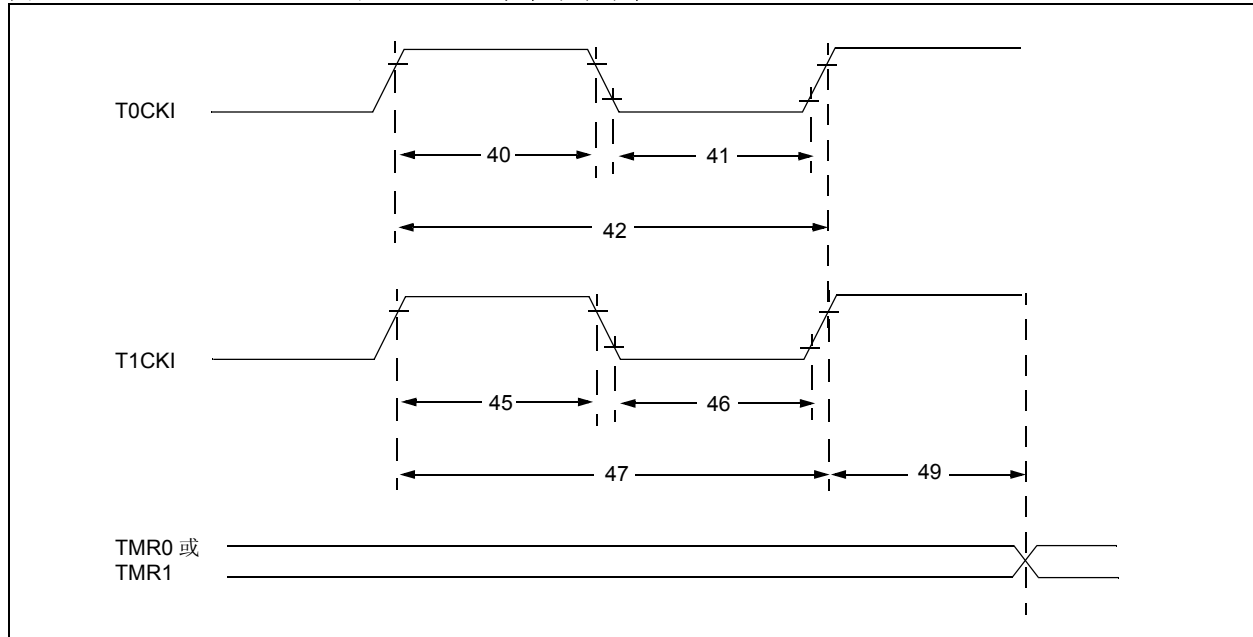
**注 1:** 指令周期 (Tcy) 等于输入振荡器时钟周期的 4 倍。所有规范值均基于器件在标准工作条件下执行代码时对应特定振荡器类型的特征数据。超出这些规范值可能导致振荡器运行不稳定和 / 或电流消耗超出预期值。所有器件在测试“最小值”时, 都在 OSC1 引脚上连接了外部时钟。当使用了外部时钟输入时, 所有器件的“最大值”周期时限为“DC” (没有时钟)。

**2:** 通过设计确保。

**3:** 较慢时钟的周期。

**4:** 为了确保这些电压的容差, 必须尽量靠近器件在 VDD 和 VSS 之间连接去耦电容。建议并联一个 0.1  $\mu\text{F}$  和 0.01  $\mu\text{F}$  的电容。

**图 30-10: TIMER0 和 TIMER1 外部时钟时序**



# PIC16(L)F1933

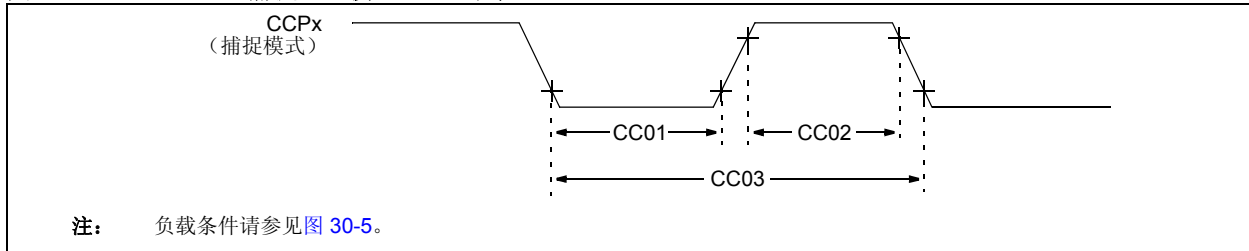
**表 30-6: TIMER0 和 TIMER1 外部时钟要求**

标准工作条件 (除非另外声明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
40*	Tt0H	T0CKI 高电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	10	—	—	ns	
41*	Tt0L	T0CKI 低电平脉冲宽度	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	10	—	—	ns	
42*	Tt0P	T0CKI 周期		取以下两者中的较大值: $20$ 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值 (2, 4, ..., 256)
45*	Tt1H	T1CKI 高电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
46*	Tt1L	T1CKI 低电平时间	同步, 无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			同步, 有预分频器	15	—	—	ns	
			异步	30	—	—	ns	
47*	Tt1P	T1CKI 输入周期	同步	取以下两者中的较大值: $30$ 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值 (1, 2, 4, 8)
			异步	60	—	—	ns	
48	Ft1	Timer1 振荡器输入频率范围 (将 T1OSCEN 位置 1 使能振荡器)		32.4	32.768	33.1	kHz	
49*	TCKEZTMR1	从外部时钟边沿到定时器递增的延时		2 T <sub>osc</sub>	—	7 T <sub>osc</sub>	—	同步模式下的定时器

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

**图 30-11: 捕捉 / 比较 / PWM 时序 (CCP)**



**表 30-7: 捕捉 / 比较 / PWM 要求 (CCP)**

标准工作条件 (除非另外声明)								
工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$								
参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
CC01*	TccL	CCP1 输入低电平时间	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	20	—	—	ns	
CC02*	TccH	CCPx 输入高电平时间	无预分频器	$0.5 T_{CY} + 20$	—	—	ns	
			有预分频器	20	—	—	ns	
CC03*	TccP	CCPx 输入周期		$\frac{3 T_{CY} + 40}{N}$	—	—	ns	N = 预分频值 (1, 4 或 16)

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

**表 30-8: PIC16(L)F1933 A/D 转换器 (ADC) 特性**

标准工作条件 (除非另外声明)							
工作温度 在 25°C 下测试							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD01	NR	分辨率	—	—	10	位	
AD02	EIL	积分误差	—	—	±1.7	LSb	VREF = 3.0V
AD03	EDL	微分误差	—	—	±1	LSb	不丢失编码 VREF = 3.0V
AD04	E0FF	失调误差	—	—	±2.5	LSb	VREF = 3.0V
AD05	EGN	增益误差	—	—	±2.0	LSb	VREF = 3.0V
AD06	VREF	参考电压 <sup>(3)</sup>	1.8	—	VDD	V	VREF = (VREF+ 和 VREF- 的较小值) (注 5)
AD07	VAIN	满量程范围	VSS	—	VREF	V	
AD08	ZAIN	模拟信号源推荐阻抗	—	—	10	kΩ	如果输入引脚上有外部 0.01μF 电容, 则该值可以更高。

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

注 1: 绝对误差包括积分误差、微分误差、失调误差和增益误差。

2: A/D 转换结果不会因输入电压的增加而减小, 并且不会丢失编码。

3: ADC VREF 来自选为参考输入引脚的外部 VREF、VDD 引脚或 FVR。

4: 当 ADC 关闭时, 它除了消耗泄漏电流外, 不消耗任何其他电流。掉电电流规范包括 ADC 模块消耗的任何泄漏电流。

5: 选择的 FVR 电压必须是 2.048V 或 4.096V。

**表 30-9: PIC16(L)F1933 A/D 转换要求**

标准工作条件 (除非另外声明)							
工作温度 $-40^{\circ}\text{C} \leq T_A \leq 125^{\circ}\text{C}$							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
AD130*	TAD	A/D 时钟周期	1.0	—	9.0	μs	基于 TOSC
		A/D 内部 RC 振荡器周期	1.0	2.5	6.0	μs	ADCS<1:0> = 11 (ADRC 模式)
AD131	TCNV	转换时间 (不包括采集时间) <sup>(1)</sup>	—	11	—	TAD	将 GO/DONE 位置 1 以完成转换
AD132*	TACQ	采集时间	—	5.0	—	μs	

\* 这些参数仅为特征值, 未经测试。

† 除非另外声明, 否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考, 未经测试。

注 1: ADRES 寄存器可在下一个 Tcy 周期被读取。

# PIC16(L)F1933

图 30-12: PIC16(L)F1933 A/D 转换时序 (正常模式)

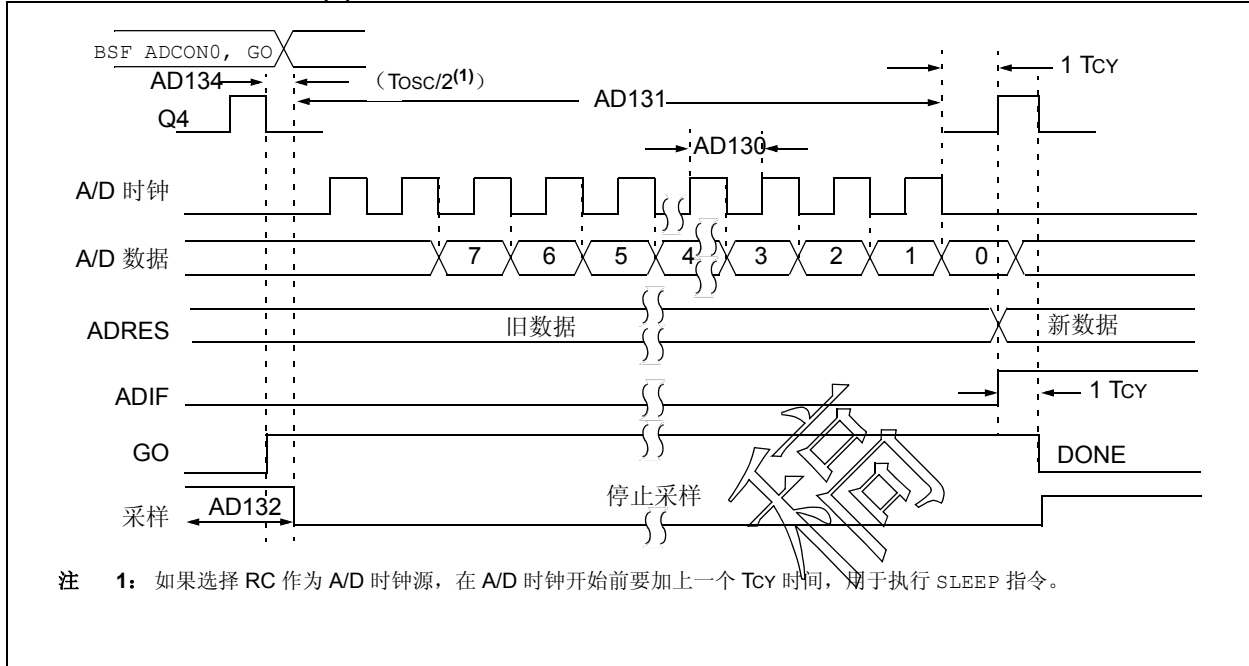
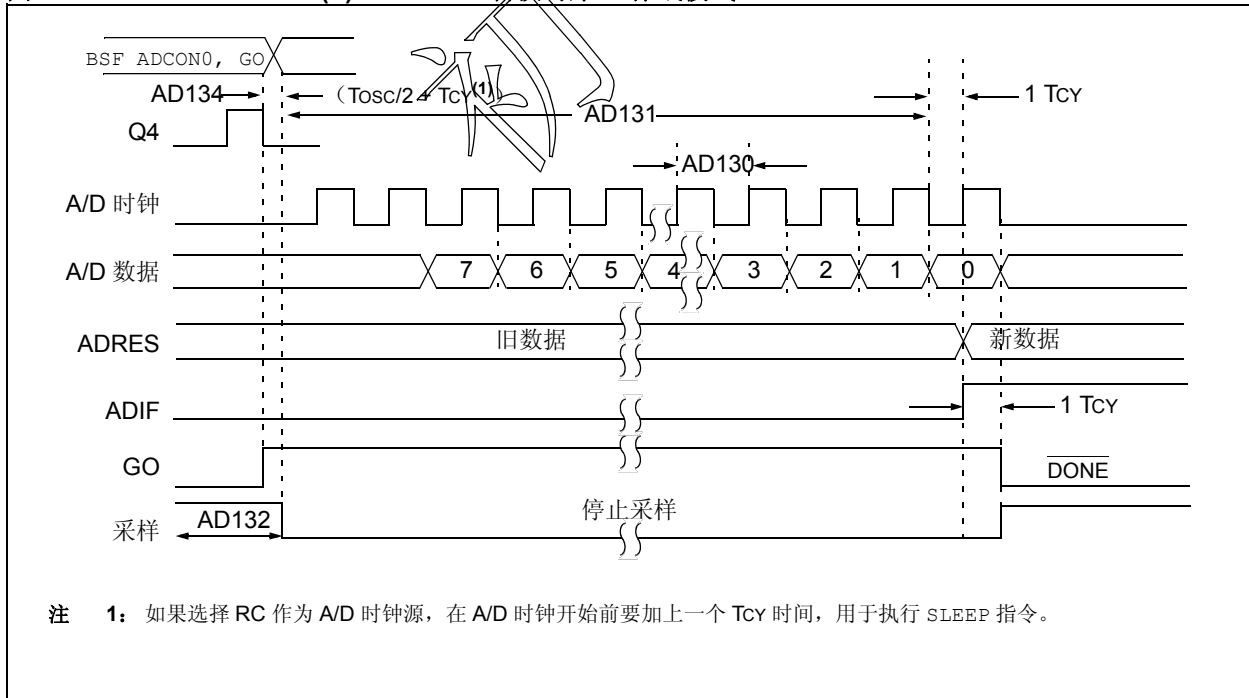


图 30-13: PIC16(L)F1933 A/D 转换时序 (休眠模式)



**表 30-10: 比较器规范**

工作条件:  $1.8V < V_{DD} < 5.5V$ ,  $-40^{\circ}C < T_A < +125^{\circ}C$  (除非另外声明)。

参数编号	符号	特性	最小值	典型值	最大值	单位	备注
CM01	$V_{IOFF}$	输入失调电压	—	$\pm 7.5$	$\pm 60$	mV	高功耗模式
CM02	$V_{ICM}$	输入共模电压	0	—	$V_{DD}$	V	
CM03	CMRR	共模抑制比	—	50	—	dB	
CM04A	TRESP	响应时间上升沿	—	400	800	ns	高功耗模式
CM04B		响应时间下降沿	—	200	400	ns	高功耗模式
CM04C		响应时间上升沿	—	1200	—	ns	低功耗模式
CM04D		响应时间下降沿	—	550	—	ns	低功耗模式
CM05	TMC2OV	比较器模式改变到输出有效的时间*	—	—	10	$\mu s$	
CM06	CHYSTER	比较器迟滞	—	45	—	mV	迟滞开启

\* 这些参数仅为特征值, 未经测试。

- 注 1: 响应时间是在比较器的一个输入端的电压为  $V_{DD}/2$ , 而另一个输入端的电压从  $V_{SS}$  变化到  $V_{DD}$  时测得的。  
 注 2: 使能  $CMxCON0$  寄存器的  $CxHYS$  位时, 将提供比较器迟滞。

**表 30-11: 数模转换器 (DAC) 规范**

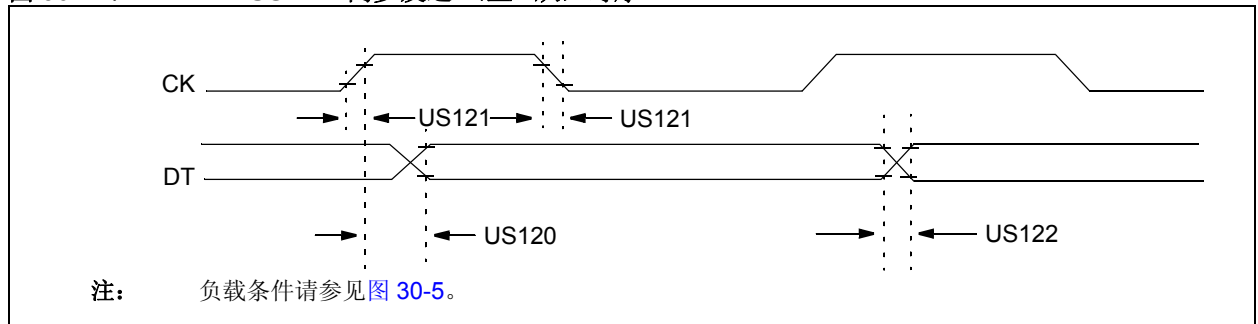
工作条件:  $2.5V < V_{DD} < 5.5V$ ,  $-40^{\circ}C < T_A < +125^{\circ}C$  (除非另外声明)。

参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DAC01*	CLSB	步长 (2)	—	$V_{DD}/32$	—	V	
DAC02*	CACC	绝对精度	—	—	$\pm 1/2$	LSb	
DAC03*	CR	单位电阻值 (R)	—	5000	—	$\Omega$	
DAC04*	CST	稳定时间 (1)	—	—	10	$\mu s$	

\* 这些参数仅为特征值, 未经测试。

- 注 1: 稳定时间是在  $DACR<4:0>$  从 0000 变化到 1111 时测得的。

**图 30-14: USART 同步发送 (主/从) 时序**

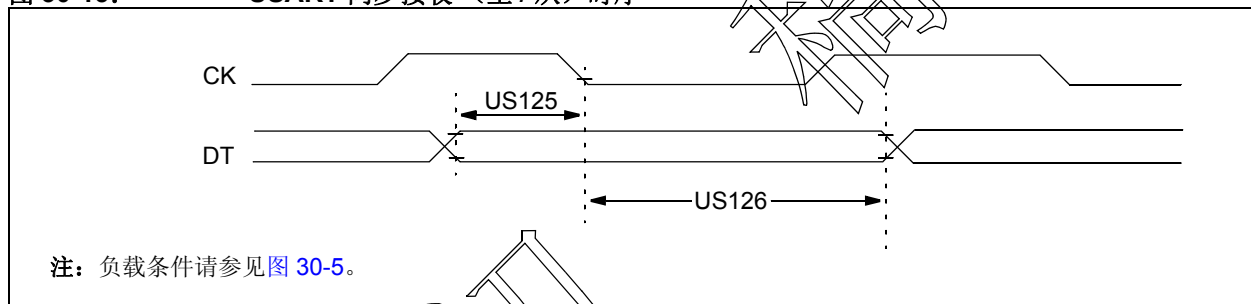


# PIC16(L)F1933

**表 30-12: USART 同步发送要求**

标准工作条件（除非另外声明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$						
参数编号	符号	特性	最小值	最大值	单位	条件
US120	TckH2DTV	同步发送（主从模式） 时钟高电平到数据输出有效的的时间	3.0-5.5V	—	80	ns
			1.8-5.5V	—	100	ns
US121	TckRF	时钟输出上升和下降时间 （主模式）	3.0-5.5V	—	45	ns
			1.8-5.5V	—	50	ns
US122	TdTRF	数据输出上升和下降时间	3.0-5.5V	—	45	ns
			1.8-5.5V	—	50	ns

**图 30-15: USART 同步接收（主/从）时序**



**表 30-13: USART 同步接收要求**

标准工作条件（除非另外声明） 工作温度 $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$						
参数编号	符号	特性	最小值	最大值	单位	条件
US125	TdtV2ckL	同步接收（主从模式） CK ↓ 前的数据保持时间（数据保持时间）	10	—	ns	
		US126	TckL2DTL	CK ↓ 后的数据保持时间（数据保持时间）	15	—



图 30-16: SPI 主模式时序 (CKE = 0, SMP = 0)

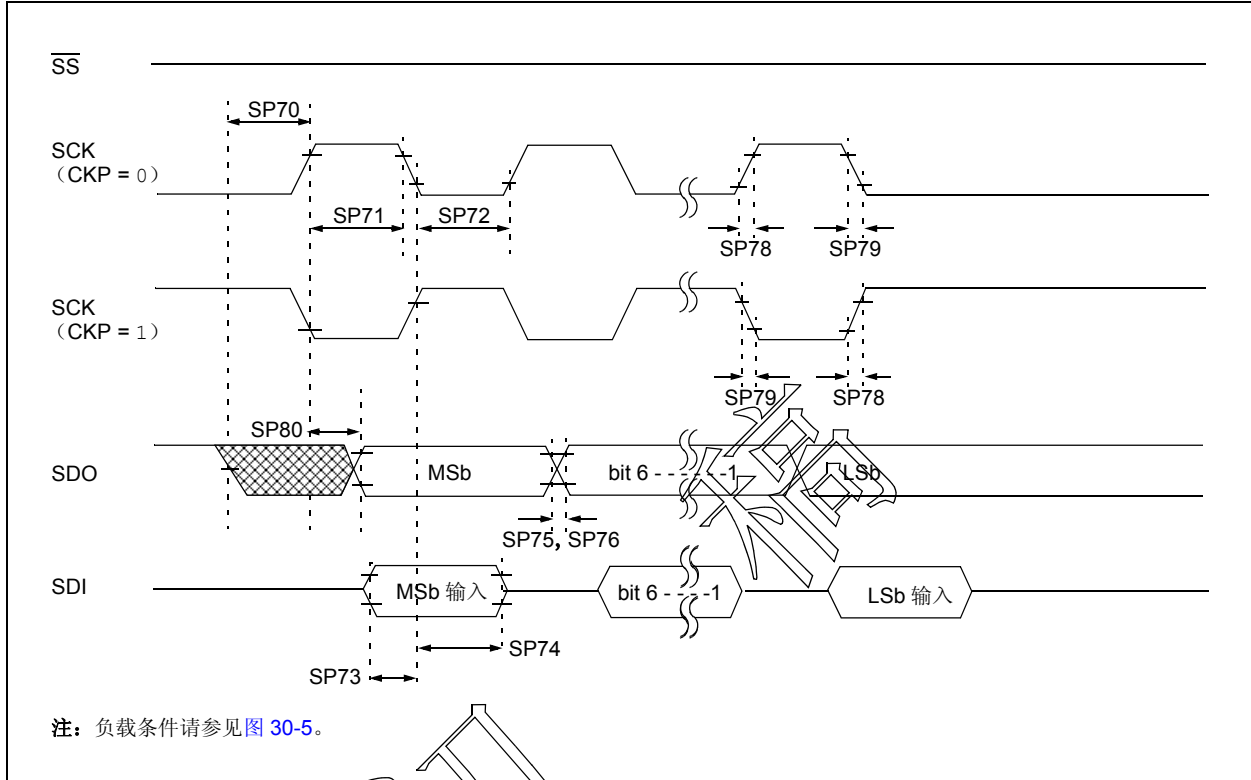
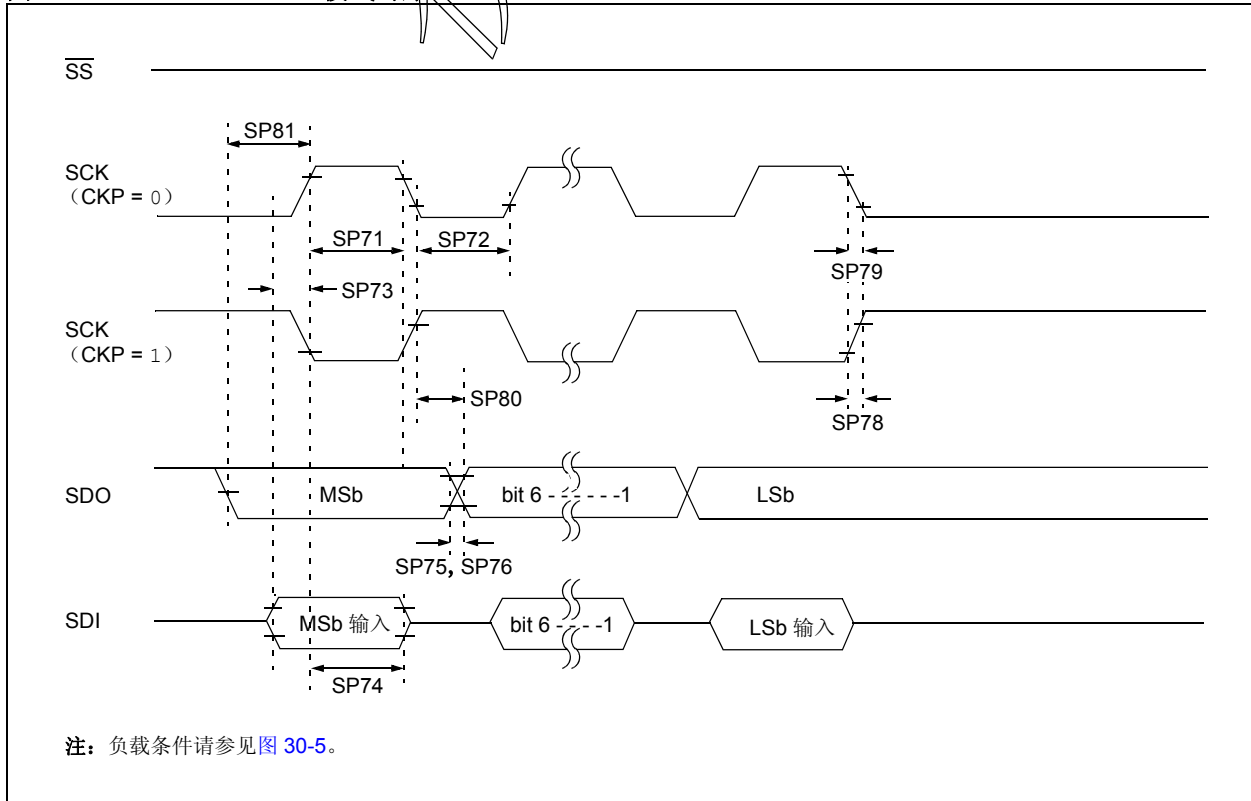


图 30-17: SPI 主模式时序 (CKE = 1, SMP = 1)



# PIC16(L)F1933

图 30-18: SPI 从模式时序 (CKE = 0)

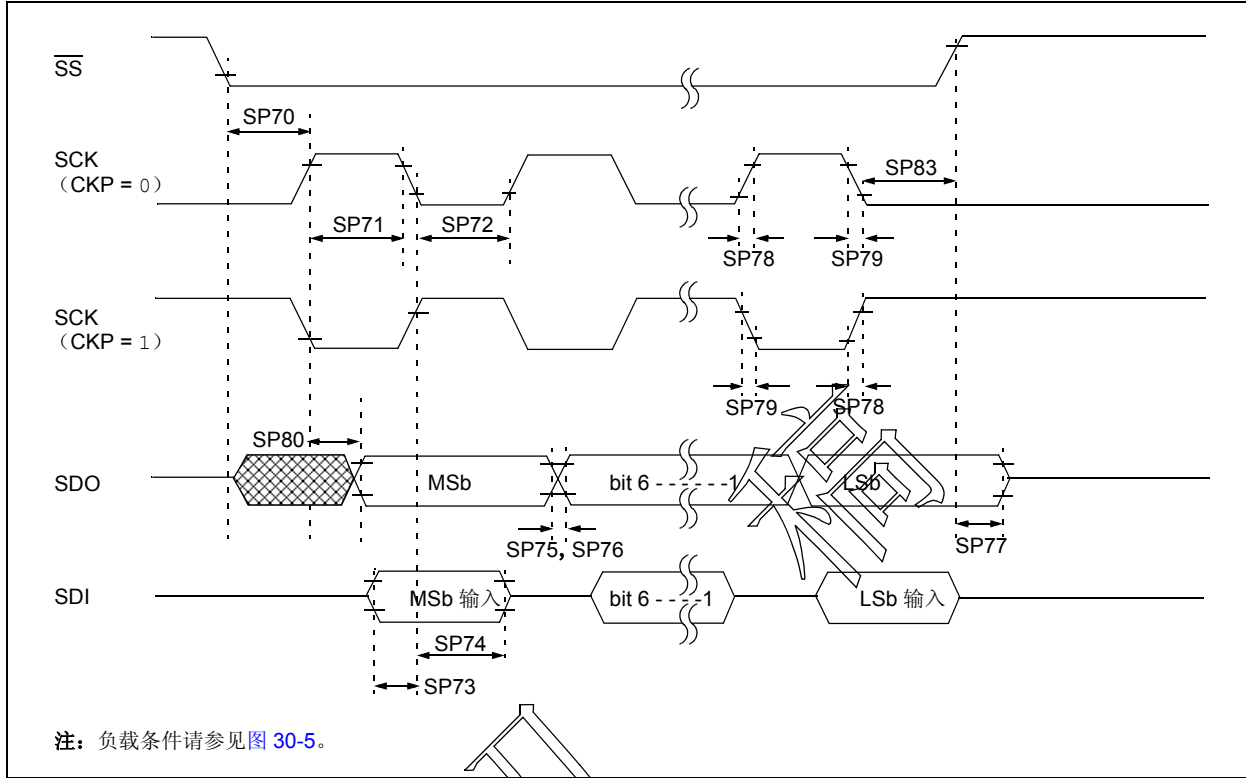
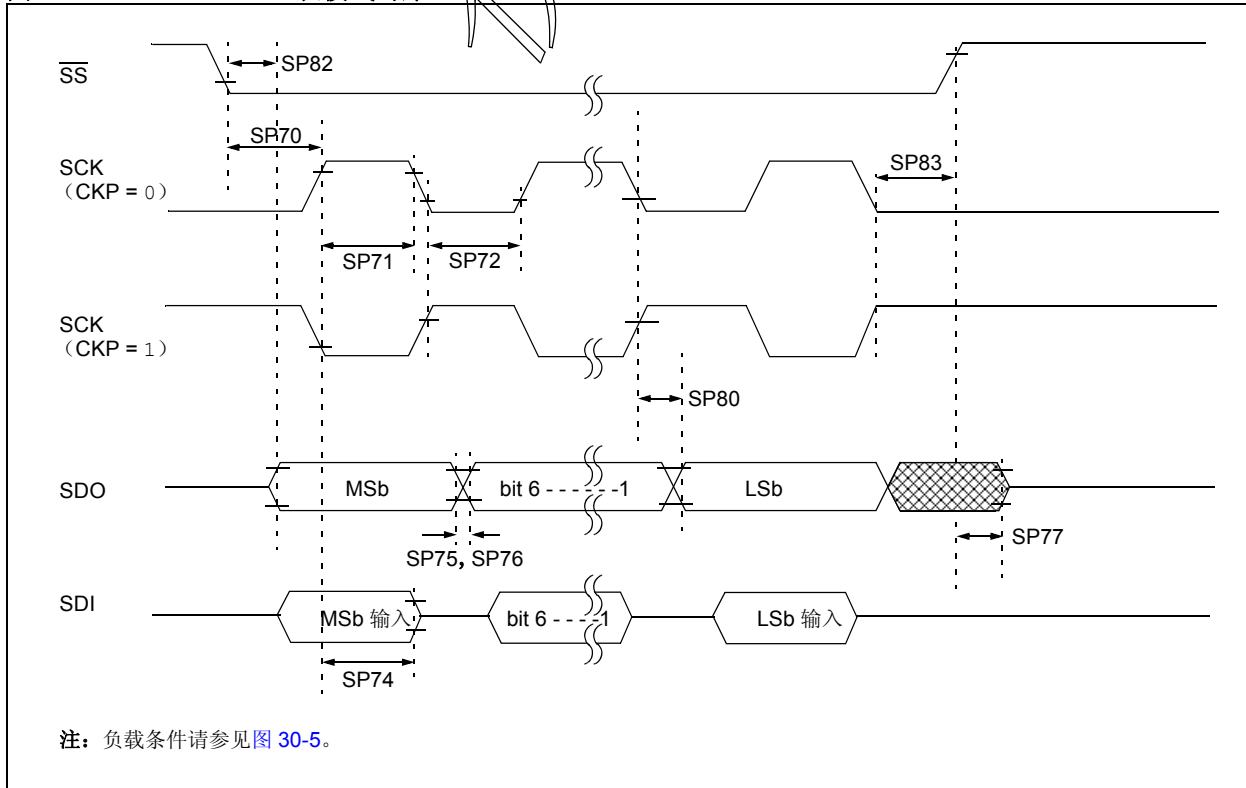


图 30-19: SPI 从模式时序 (CKE = 1)



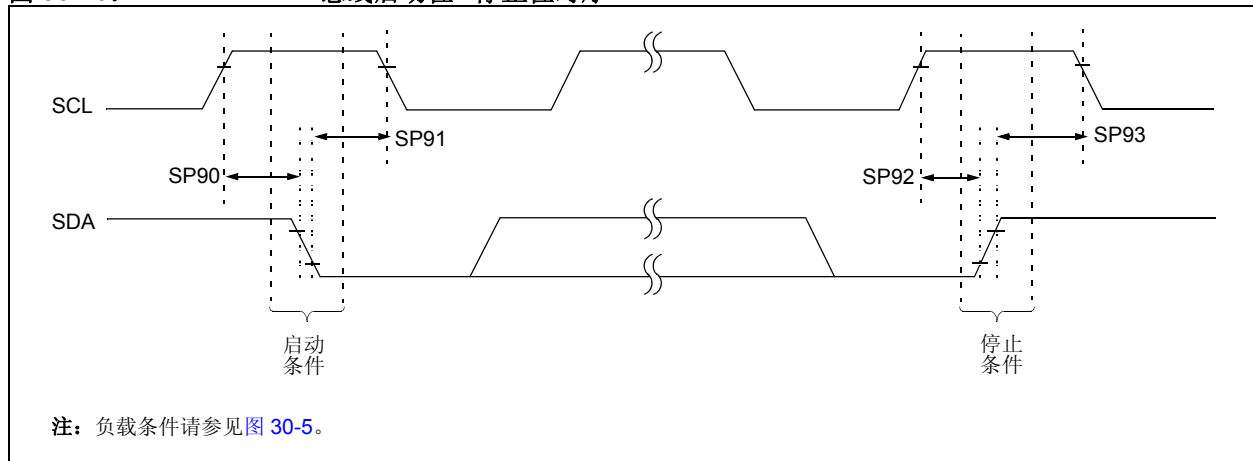
**表 30-14: SPI 模式要求**

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
SP70*	TssL2sCH, TssL2sCL	$\overline{SS}\downarrow$ 至 SCK $\downarrow$ 或 SCK $\uparrow$ 输入的时间	Tcy	—	—	ns	
SP71*	Tsch	SCK 输入高电平时间 (从模式)	Tcy + 20	—	—	ns	
SP72*	Tscl	SCK 输入低电平时间 (从模式)	Tcy + 20	—	—	ns	
SP73*	TdIV2sCH, TdIV2sCL	SDI 数据输入至 SCK 边沿的建立时间	100	—	—	ns	
SP74*	Tsch2DiL, Tscl2DiL	SDI 数据输入至 SCK 边沿的保持时间	100	—	—	ns	
SP75*	TdoR	SDO 数据输出上升时间	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP76*	TdoF	SDO 数据输出下降时间	—	10	25	ns	
SP77*	TssH2doZ	$\overline{SS}\uparrow$ 至 SDO 输出高阻态的时间	—	—	50	ns	
SP78*	TscR	SCK 输出上升时间 (主模式)	3.0-5.5V	—	10	25	ns
			1.8-5.5V	—	25	50	ns
SP79*	TscF	SCK 输出下降时间 (主模式)	—	10	25	ns	
SP80*	Tsch2doV, Tscl2doV	SCK 边沿后 SDO 数据输出有效的时间	3.0-5.5V	—	—	50	ns
			1.8-5.5V	—	—	145	ns
SP81*	TdoV2sCH, TdoV2sCL	SDO 数据输出建立至 SCK 边沿的时间	Tcy	—	—	ns	
SP82*	TssL2doV	$\overline{SS}\downarrow$ 边沿后 SDO 数据输出有效的时间	—	—	50	ns	
SP83*	Tsch2ssH, Tscl2ssH	SCK 边沿至 $\overline{SS}\uparrow$ 的时间	1.5Tcy + 40	—	—	ns	

\* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考未经测试。

**图 30-20: I<sup>2</sup>C™ 总线启动位 / 停止位时序**



# PIC16(L)F1933

表 30-15: I<sup>2</sup>C™ 总线启动位 / 停止位时序要求

参数编号	符号	特性	最小值	典型值	最大值	单位	条件	
SP90*	TSU:STA	启动条件建立时间	100 kHz 模式	4700	—	—	ns	仅与重复启动条件相关
			400 kHz 模式	600	—	—		
SP91*	THD:STA	启动条件保持时间	100 kHz 模式	4000	—	—	ns	这个周期后产生第一个时钟脉冲
			400 kHz 模式	600	—	—		
SP92*	TSU:STO	停止条件建立时间	100 kHz 模式	4700	—	—	ns	
			400 kHz 模式	600	—	—		
SP93	THD:STO	停止条件保持时间	100 kHz 模式	4000	—	—	ns	
			400 kHz 模式	600	—	—		

\* 这些参数仅为特征值，未经测试。

图 30-21: I<sup>2</sup>C™ 总线数据时序

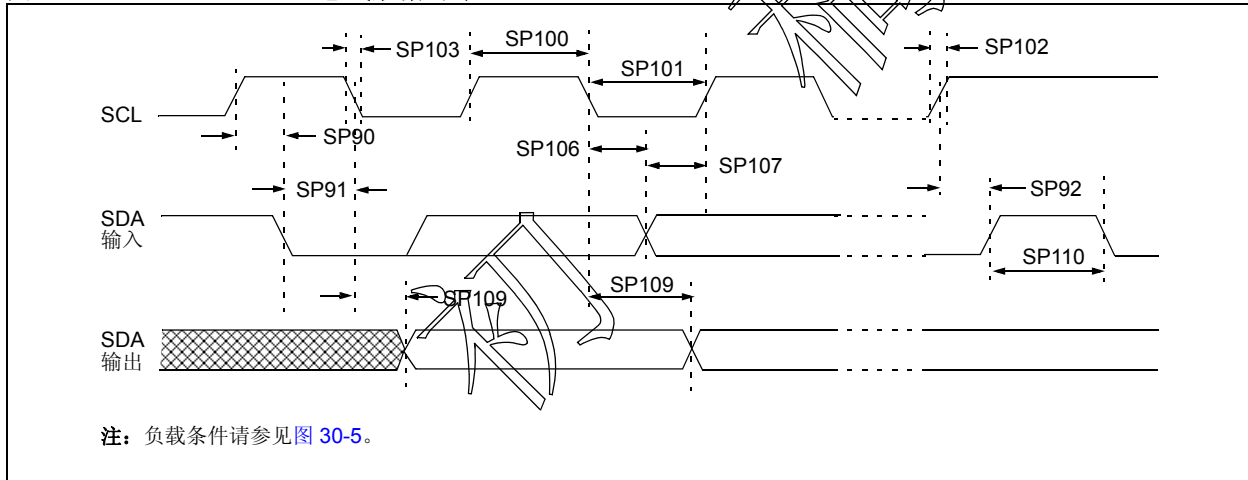


表 30-16: I<sup>2</sup>C™ 总线数据要求

参数编号	符号	特性	最小值	最大值	单位	条件	
SP100*	THIGH	时钟高电平时间	100 kHz 模式	4.0	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	0.6	—	μs	器件工作频率不得低于 10 MHz
			SSP 模块	1.5T <sub>CY</sub>	—		
SP101*	TLOW	时钟低电平时间	100 kHz 模式	4.7	—	μs	器件工作频率不得低于 1.5 MHz
			400 kHz 模式	1.3	—	μs	器件工作频率不得低于 10 MHz
			SSP 模块	1.5T <sub>CY</sub>	—		
SP102*	TR	SDA 和 SCL 上升时间	100 kHz 模式	1000	—	ns	C <sub>B</sub> 值规定在 10-400 pF 之间
			400 kHz 模式	20 + 0.1C <sub>B</sub>	300	ns	
SP103*	TF	SDA 和 SCL 下降时间	100 kHz 模式	—	250	ns	C <sub>B</sub> 值规定在 10-400 pF 之间
			400 kHz 模式	20 + 0.1C <sub>B</sub>	250	ns	
SP106*	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
SP107*	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	(注 2)
			400 kHz 模式	100	—	ns	
SP109*	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns	(注 1)
			400 kHz 模式	—	—	ns	
SP110*	TBUF	总线空闲时间	100 kHz 模式	4.7	—	μs	在新的传输开始前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
SP111	CB	总线容性负载	—	400	pF		

\* 这些参数仅为特征值，未经测试。

- 注 1: 为避免产生意外的启动或停止条件，作为发送器的器件必须提供此内部最小延时以补偿 SCL 下降沿的未定义区域（最小值 300 ns）。
- 注 2: 快速模式（400 kHz）的 I<sup>2</sup>C™ 总线器件也可在标准模式（100 kHz）的 I<sup>2</sup>C 总线系统中使用，但必须满足 TSU:DAT ≥ 250 ns 的要求。如果该器件没有延长 SCL 信号的低电平时间，则自动满足此条件。如果该器件延长了 SCL 信号的低电平时间，其下一个数据位必须输出到 SDA 线。在 SCL 线被释放前，根据标准模式 I<sup>2</sup>C 总线规范，TR max. + TSU:DAT = 1000 + 250 = 1250 ns。

# PIC16(L)F1933

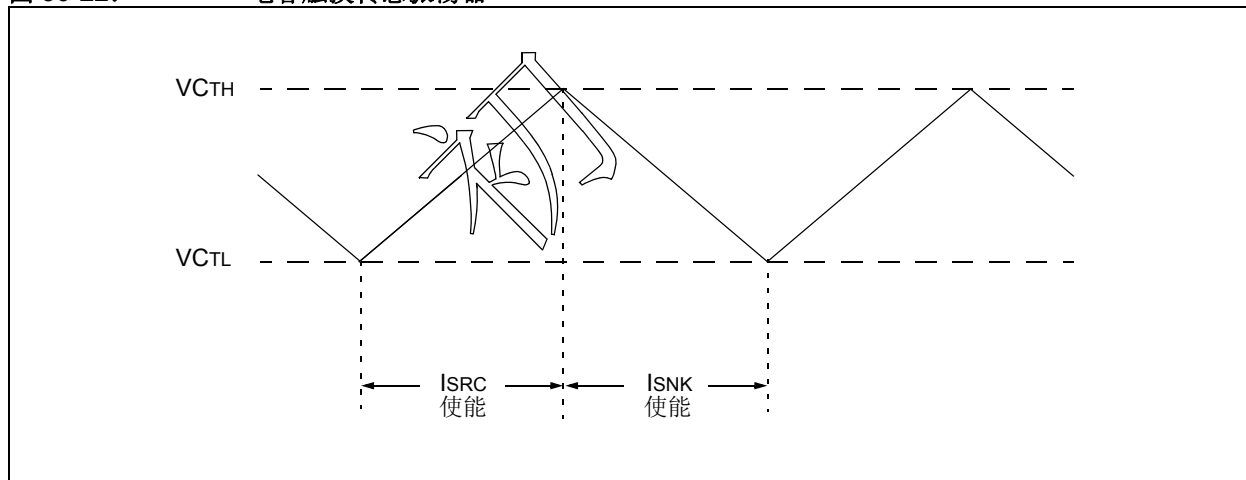
表 30-17: 电容触摸传感振荡器规范

参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
CS01*	ISRC	拉电流	高	—	-8	—	μA
			中	—	-1.5	—	μA
			低	—	-0.3	—	μA
CS02*	ISNK	灌电流	高	—	7.5	—	μA
			中	—	1.5	—	μA
			低	—	0.25	—	μA
CS03*	VCTH	电容阈值	—	0.8	—	mV	
CS04*	VCTL	电容阈值	—	0.4	—	mV	
CS05*	VCHYST	电容迟滞 (VCTH-VCTL)	高	—	525	—	mV
			中	—	375	—	mV
			低	—	300	—	mV

\* 这些参数仅为特征值，未经测试。

† 除非另外声明，否则“典型值”栏中的数据均为 3.0V、25°C 条件下的值。这些参数仅作为设计参考未经测试。

图 30-22: 电容触摸传感振荡器



## 31.0 直流和交流特性图表

当前没有可用图表。

# PIC16(L)F1933

---

注:



## 32.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机和 dsPIC® 数字信号控制器提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 编译器 / 汇编器 / 链接器
  - 适用于各种器件系列的 MPLAB C 编译器
  - 适用于各种器件系列的 HI-TECH C 编译器
  - MPASM™ 汇编器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - 适用于各种器件系列的 MPLAB 汇编器/链接器/库管理器
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- 器件编程器
  - PICkit™ 2 编程器
  - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包

## 32.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16/32 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
  - 模拟器
  - 编程器（单独销售）
  - 在线仿真器（单独销售）
  - 在线调试器（单独销售）
- 具有彩色上下文代码显示的全功能编辑器
- 多项目管理器
- 内容可直接编辑的可定制式数据窗口
- 高级源代码调试
- 鼠标停留在变量上进行查看的功能
- 将变量从源代码窗口拖放到 Watch（观察）窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件（C 语言或汇编语言）
- 点击一次即可完成编译或汇编，并将代码下载到仿真器和模拟器工具中（自动更新所有项目信息）
- 可使用如下各项进行调试：
  - 源文件（C 语言或汇编语言）
  - 混合 C 语言和汇编语言
  - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能强大的工具时的学习时间。

## 32.2 适用于各种器件系列的 MPLAB C 编译器

MPLAB C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC18、PIC24 和 PIC32 系列单片机及 dsPIC30 和 dsPIC33 系列数字信号控制器。这些编译器提供强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

## 32.3 适用于各种器件系列的 HI-TECH C 编译器

HI-TECH C 编译器代码开发系统是完整的 ANSI C 编译器，适用于 Microchip 的 PIC 系列单片机及 dsPIC 系列数字信号控制器。这些编译器提供强大的集成功能和全知代码生成能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

编译器包括一个宏汇编器、链接器、预处理程序和单步驱动程序，可以在多种平台上运行。

## 32.4 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

## 32.5 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

## 32.6 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24、PIC32 和 dsPIC 器件从符号汇编语言生成可重定位机器码。MPLAB C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

## 32.7 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC<sup>®</sup> DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

## 32.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境 (IDE) 所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC<sup>®</sup> 闪存 MCU 和 dsPIC<sup>®</sup> 闪存 DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器和 (RJ11) 或新型抗噪声、高速低压差分信号 (LVDS) 互连电缆 (CAT5) 与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对该仿真器进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、全速仿真、运行时变量查看、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

## 32.9 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 闪存数字信号控制器 (DSC) 和单片机 (MCU) 器件。结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大但易于使用的图形用户界面，该调试器可对 PIC<sup>®</sup> 闪存单片机和 dsPIC<sup>®</sup> DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器和 (RJ-11) 与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接器。

## 32.10 PICkit 3 在线调试器 / 编程器及 PICkit 3 Debug Express

结合 MPLAB 集成开发环境 (IDE) 所具有的功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC<sup>®</sup> 闪存单片机和 dsPIC<sup>®</sup> 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试 (RJ-11) 连接器 (与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容) 与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘 (内含用户指南、课程、教程、编译器和 MPLAB IDE 软件)。

## 32.11 PICkit 2 开发编程器 / 调试器及 PICkit 2 Debug Express

PICkit™ 2 开发编程器 / 调试器是一款低成本开发工具，具有易于使用的界面，适用于对 Microchip 的闪存系列单片机进行编程和调试。这一全功能的 Windows® 编程界面支持低档（PIC10F、PIC12F5xx 和 PIC16F5xx）、中档（PIC12F6xx 和 PIC16F）、PIC18F、PIC24、dsPIC30、dsPIC33 和 PIC32 系列的 8 位、16 位及 32 位单片机，以及许多 Microchip 串行 EEPROM 产品。结合 Microchip 功能强大的 MPLAB 集成开发环境（IDE），PICkit 2 可对大多数 PIC® 单片机进行在线调试。即使 PIC 单片机已嵌入应用，在线调试功能仍可以运行、暂停和单步执行程序。在断点处暂停时，可以检查和修改文件寄存器。

PICkit 2 Debug Express 包括 PICkit 2、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器及 MPLAB IDE 软件）。

## 32.12 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器（128 x 64），以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSP™ 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

## 32.13 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVAL® 评估系统、Σ-Δ ADC、流速传感器，等等。

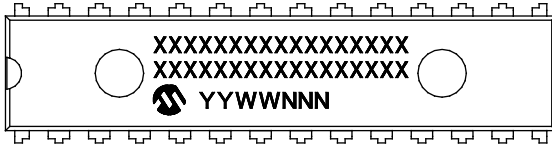
同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站（[www.microchip.com](http://www.microchip.com)）。

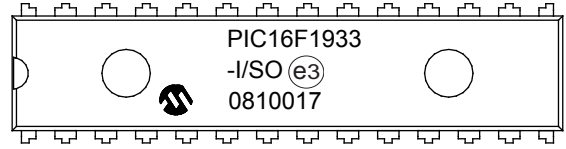
## 33.0 封装信息

### 33.1 封装标识信息

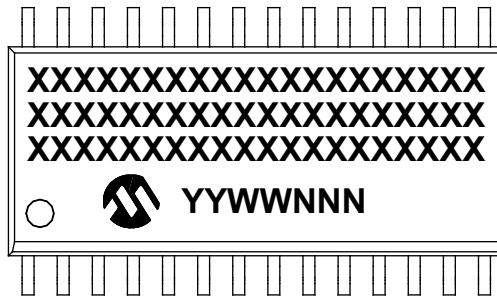
28 引脚 SPDIP (300")



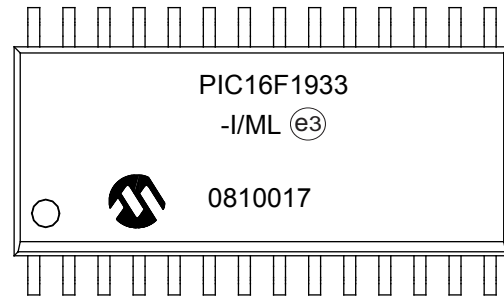
示例



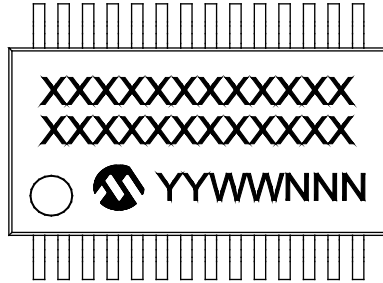
28 引脚 SOIC (7.50 mm)



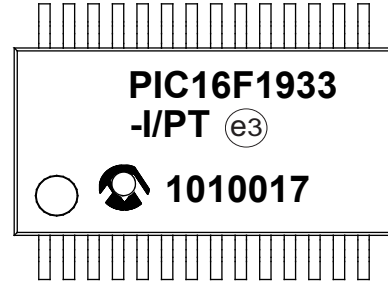
示例



28 引脚 SOIC (5.30 mm)



示例



**图注:** XX...X 客户信息  
 Y 年份代码 (日历年的最后一位数字)  
 YY 年份代码 (日历年的最后两位数字)  
 WW 星期代码 (一月一日的星期代码为“01”)  
 NNN 以字母数字排序的追踪代码  
 (e3) 雾锡 (Matte Tin, Sn) 的 JEDEC 无铅标志  
 \* 表示无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。

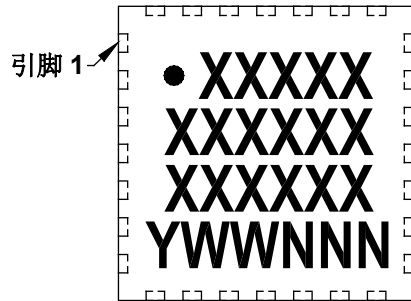
**注:** Microchip 元器件编号如果无法在同一行内完整标注, 将换行标出, 因此会限制表示客户信息的字符数。

\* 标准 PICmicro<sup>®</sup> 器件标识由 Microchip 元器件编号、年份代码、星期代码和追踪代码组成。若 PICmicro 器件标识超出上述内容, 需支付一定的附加费用。请向当地的 Microchip 销售办事处了解确认。对于 QTP 器件, 任何特殊标记的费用都已包含在 QTP 价格中。

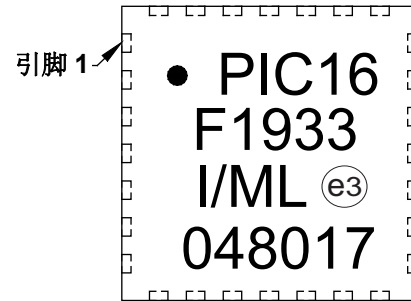
# PIC16(L)F1933

## 封装标识信息（续）

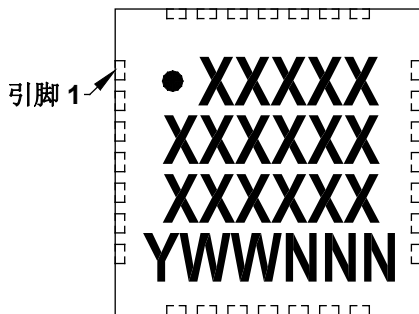
28 引脚 QFN (6x6x0.9 mm)



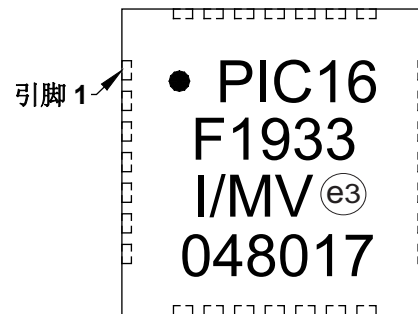
示例



28 引脚 UQFN (4x4x0.5 mm)



示例



<b>图注:</b>	XX...X 客户信息 Y 年份代码（日历年的最后一位数字） YY 年份代码（日历年的最后两位数字） WW 星期代码（一月一日的星期代码为“01”） NNN 以字母数字排序的追踪代码 (e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志 * 表示无铅封装。JEDEC 无铅标志 (e3) 标示于此种封装的外包装上。
<b>注:</b>	Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

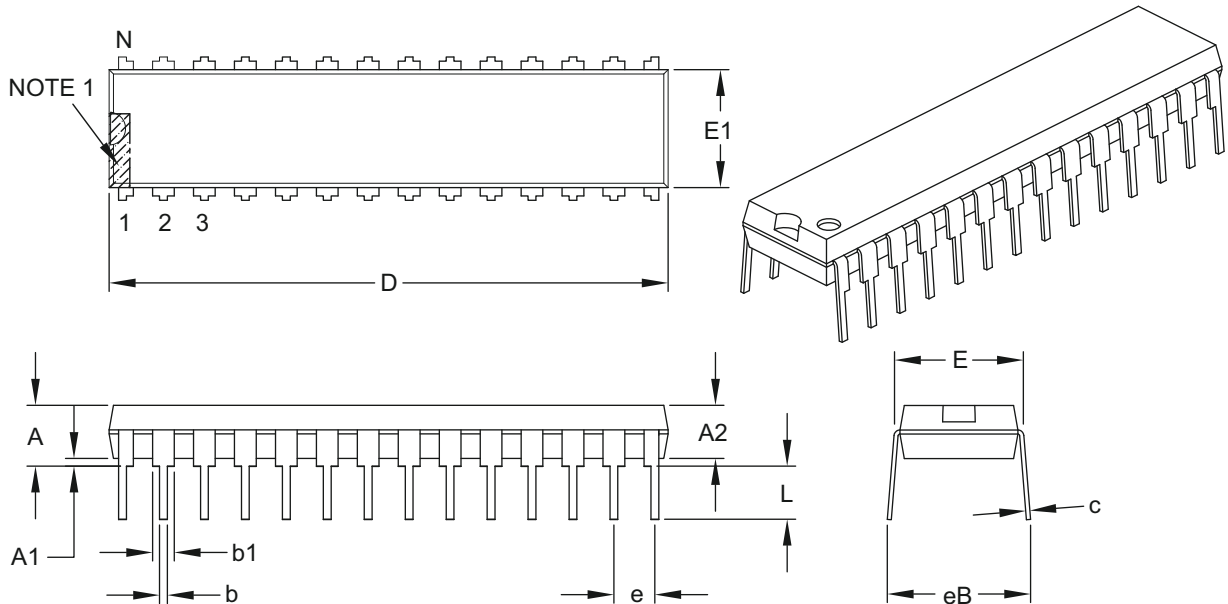
\* 标准 PICmicro<sup>®</sup> 器件标识由 Microchip 元器件编号、年份代码、星期代码和追踪代码组成。若 PICmicro 器件标识超出上述内容，需支付一定的附加费用。请向当地的 Microchip 销售办事处了解确认。对于 QTP 器件，任何特殊标记的费用都已包含在 QTP 价格中。

## 33.2 封装详细信息

以下部分将介绍各种封装的技术细节。

### 28 引脚窄型塑封双列直插式封装 (SP) —— 主体 300 mil [SPDIP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	INCHES		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	.100 BSC		
Top to Seating Plane	A	–	–	.200
Molded Package Thickness	A2	.120	.135	.150
Base to Seating Plane	A1	.015	–	–
Shoulder to Shoulder Width	E	.290	.310	.335
Molded Package Width	E1	.240	.285	.295
Overall Length	D	1.345	1.365	1.400
Tip to Seating Plane	L	.110	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.040	.050	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing §	eB	–	–	.430

#### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. § Significant Characteristic.
3. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

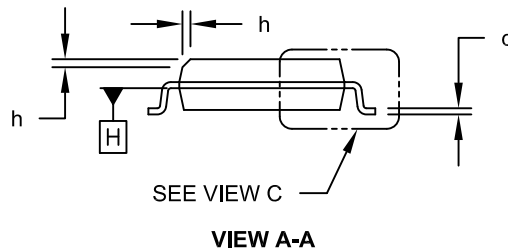
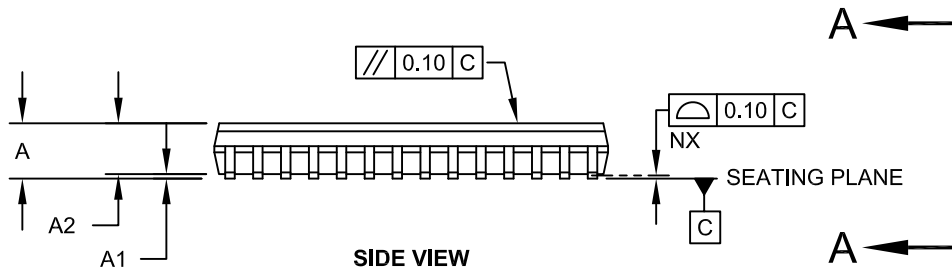
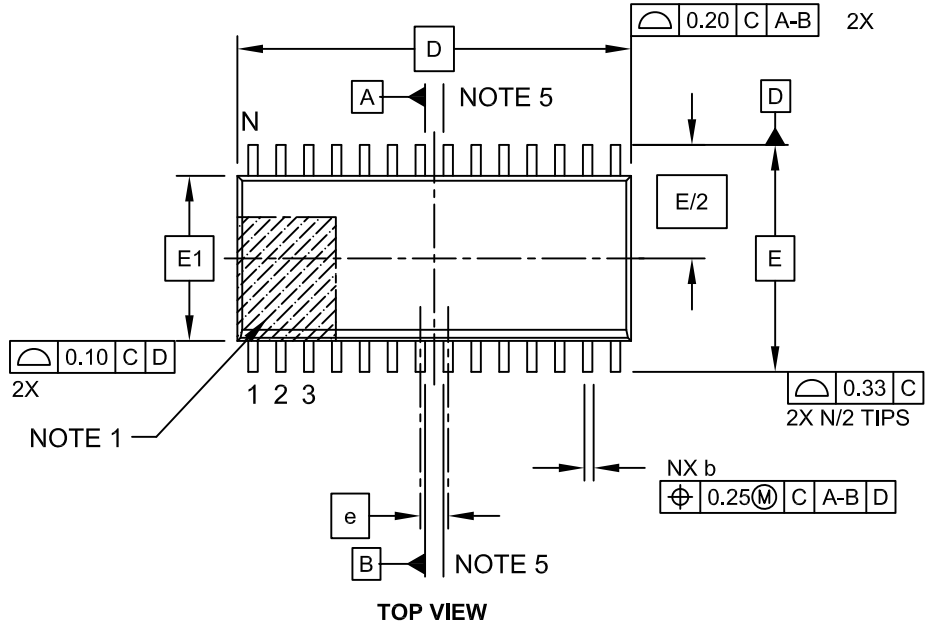
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing C04-070B

# PIC16(L)F1933

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



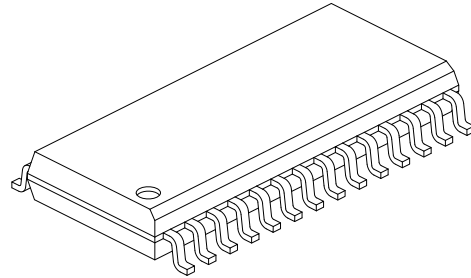
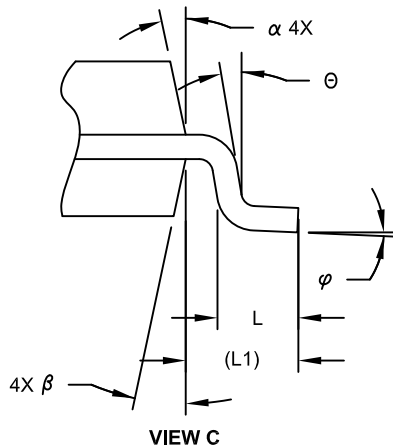
Microchip Technology Drawing C04-052C Sheet 1 of 2



# PIC16(L)F1933

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Pins	N		28		
Pitch	e		1.27 BSC		
Overall Height	A	-	-	-	2.65
Molded Package Thickness	A2	2.05	-	-	-
Standoff §	A1	0.10	-	-	0.30
Overall Width	E		10.30 BSC		
Molded Package Width	E1		7.50 BSC		
Overall Length	D		17.90 BSC		
Chamfer (Optional)	h	0.25	-	-	0.75
Foot Length	L	0.40	-	-	1.27
Footprint	L1		1.40 REF		
Lead Angle	$\theta$	0°	-	-	-
Foot Angle	$\varphi$	0°	-	-	8°
Lead Thickness	c	0.18	-	-	0.33
Lead Width	b	0.31	-	-	0.51
Mold Draft Angle Top	$\alpha$	5°	-	-	15°
Mold Draft Angle Bottom	$\beta$	5°	-	-	15°

### Notes:

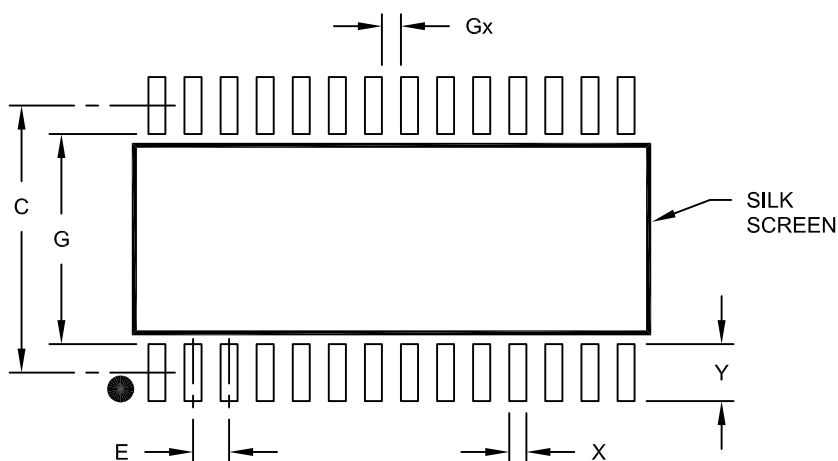
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- § Significant Characteristic
- Dimension D does not include mold flash, protrusions or gate burrs, which shall not exceed 0.15 mm per end. Dimension E1 does not include interlead flash or protrusion, which shall not exceed 0.25 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.
- Datums A & B to be determined at Datum H.

Microchip Technology Drawing C04-052C Sheet 2 of 2

# PIC16(L)F1933

## 28 引脚塑封宽条小外形封装 (SO) —— 主体 7.50 mm [SOIC]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



### RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	1.27 BSC		
Contact Pad Spacing	C		9.40	
Contact Pad Width (X28)	X			0.60
Contact Pad Length (X28)	Y			2.00
Distance Between Pads	Gx	0.67		
Distance Between Pads	G	7.40		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

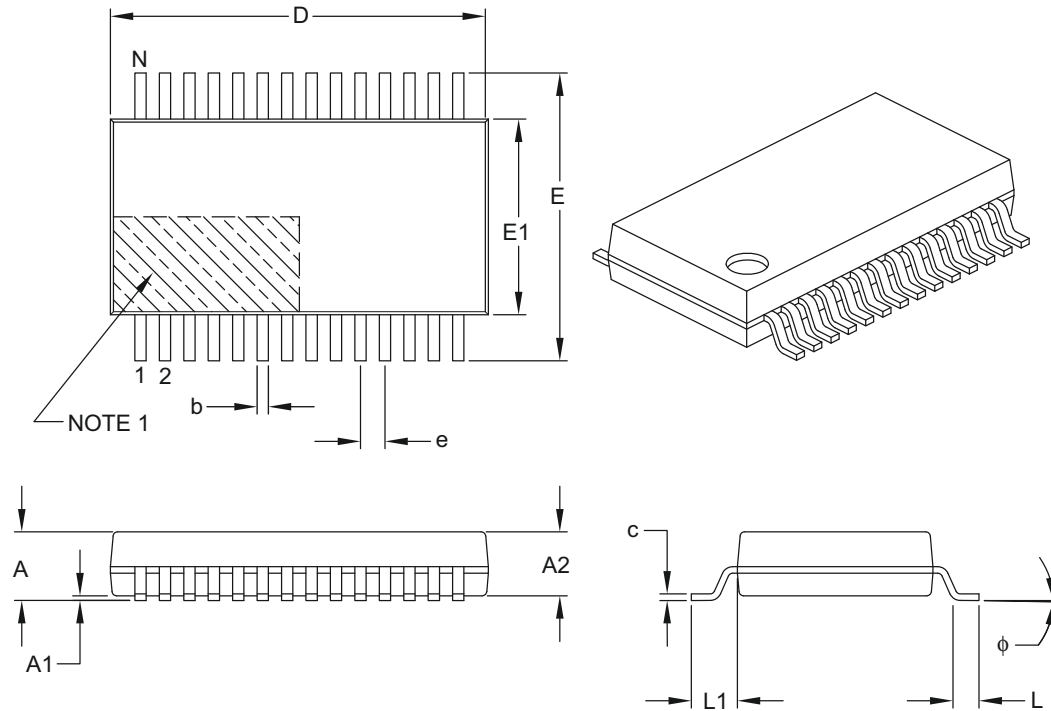
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2052A

# PIC16(L)F1933

## 28 引脚塑封缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	–	–	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	–	–
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	–	0.25
Foot Angle	$\phi$	0°	4°	8°
Lead Width	b	0.22	–	0.38

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20 mm per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

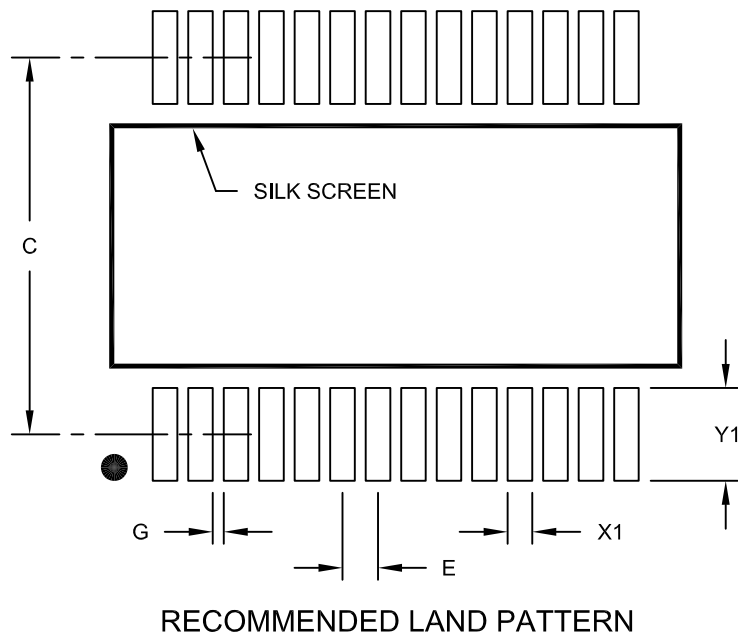
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073B

# PIC16(L)F1933

## 28 引脚塑封缩小外形封装 (SS) —— 主体 5.30 mm [SSOP]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.20	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.75
Distance Between Pads	G	0.20		

**Notes:**

1. Dimensioning and tolerancing per ASME Y14.5M

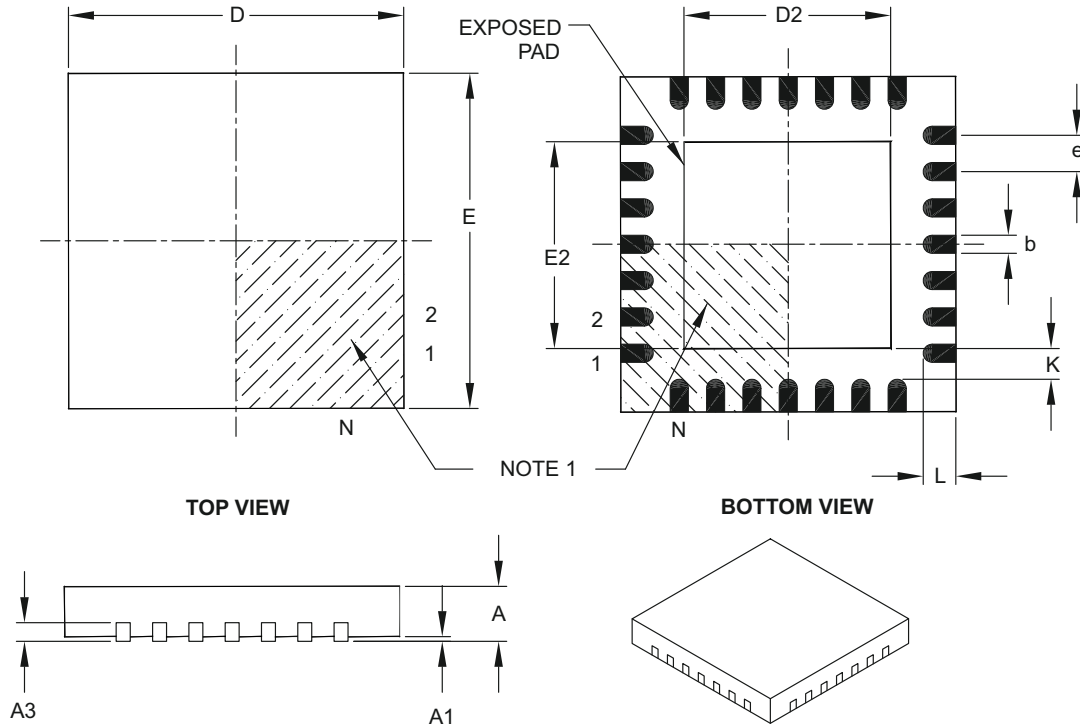
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2073A

# PIC16(L)F1933

28 引脚塑封四方扁平无脚封装 (ML) —— 主体 6x6 mm, 底脚占位长度 0.55 mm [QFN]

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	3.65	3.70	4.20
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	3.65	3.70	4.20
Contact Width	b	0.23	0.30	0.35
Contact Length	L	0.50	0.55	0.70
Contact-to-Exposed Pad	K	0.20	-	-

**Notes:**

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Package is saw singulated.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

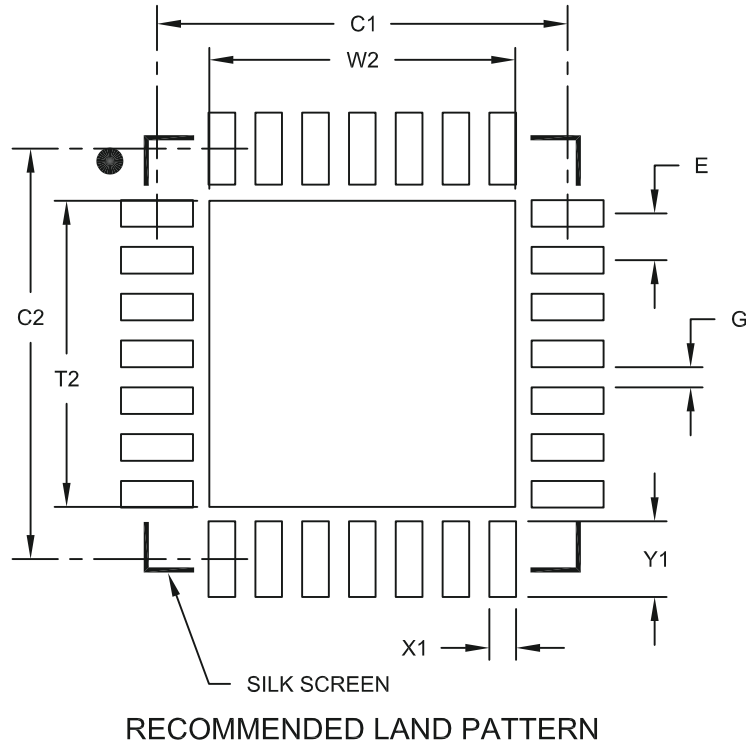
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-105B

# PIC16(L)F1933

28 引脚塑封四方扁平无脚封装 (ML) —— 主体 6x6 mm，底脚占位长度 0.55 mm [QFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	W2			4.25
Optional Center Pad Length	T2			4.25
Contact Pad Spacing	C1		5.70	
Contact Pad Spacing	C2		5.70	
Contact Pad Width (X28)	X1			0.37
Contact Pad Length (X28)	Y1			1.00
Distance Between Pads	G	0.20		

Notes:

1. Dimensioning and tolerancing per ASME Y14.5M

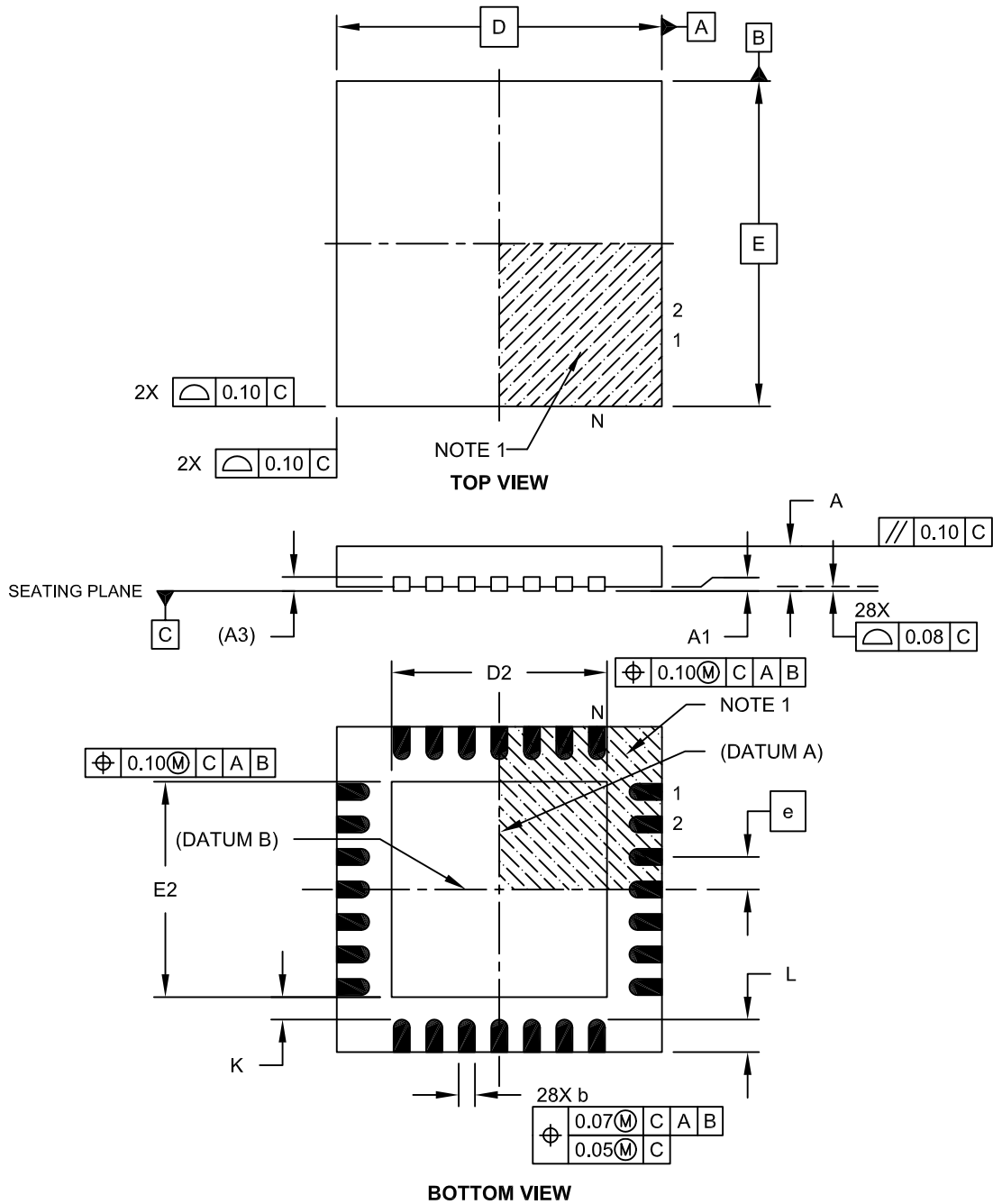
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2105A

# PIC16(L)F1933

## 28 引脚塑封超薄型四方扁平无脚封装 (MV) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。

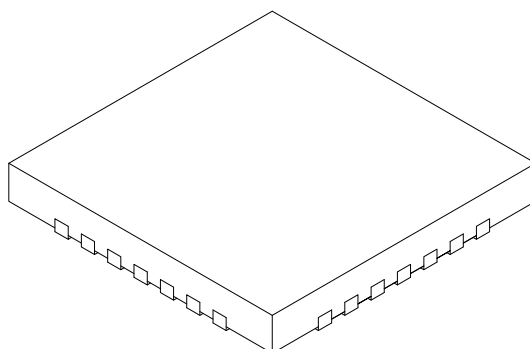


Microchip Technology Drawing C04-152A Sheet 1 of 2

# PIC16(L)F1933

## 28 引脚塑封超薄型四方扁平无脚封装 (MV) —— 主体 4x4x0.5 mm [UQFN]

注：最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范。



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.40 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.127 REF		
Overall Width	E	4.00 BSC		
Exposed Pad Width	E2	2.55	2.65	2.75
Overall Length	D	4.00 BSC		
Exposed Pad Length	D2	2.55	2.65	2.75
Contact Width	b	0.15	0.20	0.25
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-152A Sheet 2 of 2



## 附录 A: 数据手册版本历史

### 版本 A

本文档的初始版本（2011 年 5 月）。

## 附录 B: 从其他 PIC® 器件移植

本节将讨论从其他 PIC® 器件移植到 PIC16(L)F1933 系列器件的一些问题。

### B.1 PIC16F917 到 PIC16(L)F1933

表 B-1: 功能比较

功能	PIC16F917	低档器件
最大工作频率	20 MHz	32 MHz
程序存储器最大容量 (字)	8K	8K
最大 SRAM (字节)	368	512
A/D 转换的分辨率	10 位	10 位
定时器 (8/16 位)	2/1	4/1
振荡器模式	4	8
欠压复位	有	有
内部上拉	RB<7:0>	RB<7:0>
电平变化中断	RB<7:4>	RB<7:0>
比较器	2	2
AUSART/EUSART	1/0	0/1
扩展的 WDT	有	有
WDT/BOR 的软件控制选项	无	有
INTOSC 频率	30 kHz-8 MHz	500 kHz-32 MHz
时钟切换	有	有
电容触摸传感	无	有
CCP/ECCP	2/0	2/3
增强型 PIC16 CPU	无	有
MSSP/SSP	0/1	1/0
LCD	有	有

# PIC16(L)F1933

---

注:

## 索引

### A

A/D	
规范	389
ACKSTAT	256
ACKSTAT 状态标志	256
ADC	139
采集要求	149
参考电压 (VREF)	140
端口配置	140
工作原理	143
计算采集时间	149
框图	139
内部采样开关阻抗 (R <sub>ss</sub> )	149
配置	140
配置中断	144
启动 A/D 转换	142
特殊事件触发器	143
通道选择	140
相关的寄存器	151
休眠期间的工作原理	143
源阻抗	149
中断	142
转换过程	144
转换时钟	140
ADCON0 寄存器	30, 145
ADCON1 寄存器	30, 146
ADDFSR	357
ADDWFC	357
ADRESH 寄存器	30
ADRESH (ADFM = 1)	148
ADRESH (ADFM = 0)	147
ADRESL (ADFM = 1)	148
ADRESL (ADFM = 0)	147
ANSELA 寄存器	121
ANSELB 寄存器	126
APFCON 寄存器	118

### B

BAUDCON 寄存器	286
BF	256, 258
BF 状态标志	256, 258
BORCON 寄存器	75
BRA	358
版本历史	417
备用引脚功能	118
比较模块。参见增强型捕捉 / 比较 / PWM (ECCP)	
比较器	
C2OUT 作为 T1 的门控信号	179
工作原理	159
相关的寄存器	167, 168
比较器规范	391
比较器模块	159
Cx 输出状态 —— 输入条件	161
编程, 器件指令	353
变更通知客户服务	427
捕捉 / 比较 / PWM	193
捕捉 / 比较 / PWM (CCP)	
比较模式	196
CCPx 引脚配置	196
软件中断模式	194, 196
Timer1 模式资源	194, 196
特殊事件触发器	196

捕捉模式	194
CCPx 引脚配置	194
PWM 操作	198
PWM 概述	198
PWM 模式	
分辨率	200
复位的影响	201
示例 PWM 频率与分辨率, 20 MHz	200
示例 PWM 频率与分辨率, 32 MHz	200
示例 PWM 频率与分辨率, 8 MHz	200
系统时钟频率的改变	201
休眠模式下的工作原理	201
占空比	199
PWM 设置	199
PWM 周期	199
与比较相关的寄存器	197
与捕捉相关的寄存器	195
预分频器	194
与 PWM 相关的寄存器	201, 215
捕捉模块。参见增强型捕捉 / 比较 / PWM (ECCP)	

### C

CALL	359
CALLW	359
C 编译器	
MPLAB C18	402
CCP1CON 寄存器	34, 35
CCPR1H 寄存器	34, 35
CCPR1L 寄存器	34, 35
CCPTMRS0 寄存器	217
CCPTMRS1 寄存器	217
CCPxAS 寄存器	218
CCPxCON (ECCPx) 寄存器	216
CMOUT 寄存器	166
CMxCON0 寄存器	165
CMxCON1 寄存器	166
CONFIG1 寄存器	50
CONFIG2 寄存器	52
CPSCON0 寄存器	311
CPSCON1 寄存器	312
操作码字段说明	353
程序存储器	19
映射和堆栈 (PIC16F1933/LF1933)	19, 24
从其他 PIC 单片机移植	417
存储器构成	
程序	19
数据	21, 24

### D

DACCON0 (数模转换器控制 0) 寄存器	158
DACCON1 (数模转换器控制 1) 寄存器	158
代码示例	
A/D 转换	144
初始化 PORTA	117
切换捕捉预分频比	194
写入闪存程序存储器	111
写校验	113
电平变化中断	133
相关的寄存器	135
电气规范 (PIC16F/LF1933)	367
电容触摸传感	305
规范	398
与电容触摸传感相关的寄存器	313

# PIC16(L)F1933

掉电模式 (休眠) .....	97	PDIP 详细信息 .....	407
相关的寄存器 .....	98	复位 .....	73
定时器 .....		相关的寄存器 .....	80
Timer2/4/6 .....		复位的影响 .....	
TXCON .....	191	PWM 模式 .....	201
Timer1 .....		负载条件 .....	381
T1CON .....	185	<b>G</b>	
T1GCON .....	186	高精度内部振荡器参数 .....	383
读 - 修改 - 写操作 .....	353	固定参考电压 (FVR) .....	
读者反馈表 .....	428	相关的寄存 .....	138
堆栈 .....	43	固件指令 .....	353
复位 .....	45	故障保护时钟监视器 .....	67
快速操作 .....	43	复位或从休眠模式唤醒 .....	67
堆栈上溢 / 下溢 .....	76	故障保护工作原理 .....	67
<b>E</b>		故障保护检测 .....	67
ECCP/CCP。参见增强型捕捉 / 比较 / PWM		清除故障保护条件 .....	67
EEADRH 寄存器 .....	103	<b>H</b>	
EEADR 寄存器 .....	103	汇编器 .....	
EEADRL 寄存器 .....	103, 114	MPASM 汇编器 .....	402
EECON1 寄存器 .....	103, 115	<b>I</b>	
EECON2 寄存器 .....	103, 116	I <sup>2</sup> C 模式 (MSSP) .....	
EEDATH 寄存器 .....	114	从模式 .....	
EEDATL 寄存器 .....	114	发送 .....	242
EEPROM 数据存储		带 BRG 的 I <sup>2</sup> C 时钟速率 .....	267
避免意外写入 .....	104	读写位信息 (R/W 位) .....	237
写校验 .....	113	多主模式 .....	261
EUSART .....	275	多主器件通信、总线冲突和仲裁 .....	261
波特率发生器 (BRG) .....		复位的影响 .....	261
波特率, 异步模式 .....	289	停止条件时序 .....	260
波特率误差, 计算 .....	287	休眠模式下工作 .....	261
高波特率选择位 (BRGH) .....	287	应答序列时序 .....	260
公式 .....	288	主模式 .....	
自动波特率检测 .....	292	发送 .....	256
同步从模式 .....		工作原理 .....	252
发送 .....	301	接收 .....	258
接收 .....	302	启动条件时序 .....	254, 255
相关的寄存器 .....		总线冲突 .....	
发送 .....	301	停止条件期间 .....	265
接收 .....	302	重复启动条件期间 .....	264
同步主模式 .....	296, 301	INDF 寄存器 .....	29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41
发送 .....	296	INTCON 寄存器 .....	86
接收 .....	299	INTOSC 规范 .....	383
相关的寄存器 .....		IOCBF 寄存器 .....	134
发送 .....	298	IOCBN 寄存器 .....	134
接收 .....	300	IOCBP 寄存器 .....	134
相关的寄存器 .....		<b>J</b>	
波特率发生器 .....	288	寄存器 .....	
异步模式 .....	277	ADCON0 (ADC 控制 0) .....	145
12 位间隔字符的发送和接收 .....	295	ADCON1 (ADC 控制 1) .....	146
波特率发生器 (BRG) .....	287	ADRESH (ADC 结果寄存器	
发送器 .....	277	的高字节, ADFM = 1) .....	148
接收到间隔字符时自动唤醒 .....	293	ADRESH (ADC 结果寄存器	
接收器 .....	280	的高字节, ADFM = 0) .....	147
设置带有地址检测功能的 9 位模式 .....	282	ADRESL (ADC 结果寄存器	
时钟精度 .....	284	的低字节, ADFM = 1) .....	148
相关的寄存器 .....		ADRESL (ADC 结果寄存器	
发送 .....	279	的低字节, ADFM = 0) .....	147
接收 .....	283	ANSELA (PORTA 模拟选择) .....	121
<b>F</b>		ANSELB (PORTB 模拟选择) .....	126
FSR 寄存器 .....	29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41	APFCON (备用引脚功能控制) .....	118
FVRCON (固定参考电压控制) 寄存器 .....	138	BAUDCON (波特率控制) .....	286
封装 .....	405		
标识 .....	405, 406		

BORCON (欠压复位控制) .....	75	TRISC (三态 PORTC) .....	129
CCPTMRS0 (PWM 定时器选择控制 0) .....	217	TRISE (三态 PORTE) .....	131
CCPTMRS1 (PWM 定时器选择控制 1) .....	217	TXCON .....	191
CCPxAS (CCPx 自动关闭控制) .....	218	TXSTA (发送状态和控制寄存器) .....	284
CCPxCON (ECCPx 控制) .....	216	特殊功能汇总 .....	29
CMOUT (比较器输出) .....	166	WDTCON (看门狗定时器控制) .....	101
CMxCON0 (Cx 控制) .....	165	WPUB (弱上拉 PORTB) .....	126
CMxCON1 (Cx 控制 1) .....	166	间隔字符 (12 位) 的发送和接收 .....	295
CPSCON0 (电容触摸传感控制寄存器 0) .....	311	间接寻址 .....	45
CPSCON1 (电容触摸传感控制寄存器 1) .....	312	交流特性	
DACCON0 .....	158	负载条件 .....	381
DACCON1 .....	158	工业级和扩展级 (PIC16F/LF1933) .....	382
EEADRL (EEPROM 地址) .....	114	接收到间隔字符时唤醒 .....	293
EECON1 (EEPROM 控制 1) .....	115	绝对最大值 (PIC16F/LF1933) .....	367
EECON2 (EEPROM 控制 2) .....	116	<b>K</b>	
EEDATH (EEPROM 数据) .....	114	开发支持 .....	401
EEDATL (EEPROM 数据) .....	114	看门狗定时器 (WDT) .....	76
FVRCON .....	138	带看门狗定时器的配置字 .....	102
INTCON (中断控制) .....	86	规范 .....	387
IOCBF (电平变化中断标志) .....	134	模式 .....	100
IOCBN (电平变化中断负边沿) .....	134	相关的寄存器 .....	102
IOCBP (电平变化中断正边沿) .....	134	勘误表 .....	9
LATA (PORTA 数据锁存) .....	120	客户通知服务 .....	427
LATB (PORTB 数据锁存) .....	125	客户支持 .....	427
LATC (PORTC 数据锁存) .....	129	框图	
LCDCON (LCD 控制) .....	317	(CCP) 捕捉模式工作原理 .....	194
LCDCST (LCD 对比控制) .....	320	ADC .....	139
LCDDATAx (LCD 数据) .....	321, 328	ADC 传递函数 .....	150
LCDPS (LCD 相位) .....	318	比较 .....	196
LCDREF (LCD 参考电压控制) .....	319	比较器 .....	160
LCDRL (LCD 参考电压控制) .....	328	CCP PWM .....	198
LCDSEn (LCD 段使能) .....	321	参考电压 .....	137
OPTION_REG (OPTION) .....	175	参考电压输出缓冲器示例 .....	156
OSCSTAT (振荡器状态) .....	70	电容触摸传感 .....	305, 306
OSCTUNE (振荡器调节) .....	71	EUSART 发送 .....	275
PCON (电源控制) .....	79	EUSART 接收 .....	276
PCON (电源控制寄存器) .....	79	故障保护时钟监视器 (FSCM) .....	67
PIE1( 外设中断允许 1) .....	87	晶振工作原理 .....	58, 59
PIE2 (外设中断允许 2) .....	88	LCD 偏置电压发生 .....	323
PIE3 (外设中断允许 3) .....	89	LCD 时钟产生 .....	322
PIR1 (外设中断寄存器 1) .....	90	模拟输入模型 .....	150, 164
PIR2 (外设中断请求 2) .....	91	PIC16F193X/LF193X .....	12, 18
PIR3 (外设中断请求 3) .....	92	片上复位电路 .....	73
PORTA .....	120	时钟源 .....	56
PORTB .....	125	数模转换器 (DAC) .....	156
PORTE .....	131	Timer0 .....	173
PSTRxCON (PWM 控制的控制) .....	220	Timer2/4/6 .....	189
PWMxCON (增强型 PWM 控制) .....	219	Timer1 .....	177
配置字 1 .....	50	Timer1 门控 .....	182, 183, 184
配置字 2 .....	52	通用 I/O 端口 .....	117
RCREG 寄存器 .....	292	外部 RC 模式 .....	59
RCSTA (接收状态和控制寄存器) .....	285	谐振器工作原理 .....	58
SRCON0 (SR 锁存器控制 0) .....	171	增强型 PWM .....	202
SRCON1 (SR 锁存器控制 1) .....	172	中断逻辑 .....	81
SSPADD (MSSP 地址和波特率, I <sup>2</sup> C 模式) .....	273	扩展型指令集	
SSPCON1 (MSSP 控制 1) .....	269	ADDFSR .....	357
SSPCON2( SSP 控制 2) .....	271	<b>L</b>	
SSPCON3( SSP 控制 3) .....	272	LATA 寄存器 .....	120
SSPMSK (SSP 屏蔽) .....	273	LATB 寄存器 .....	125
SSPSTAT (SSP 状态) .....	268	LATC 寄存器 .....	129
STATUS .....	22	LCD	
T1CON (Timer1 控制) .....	185	波形发生 .....	332
T1GCON (Timer1 门控) .....	186	段使能 .....	330
TRISA (三态 PORTA) .....	120	复用类型 .....	330
TRISB (三态 PORTB) .....	125		

# PIC16(L)F1933

禁止模块 .....	347	PIE3 寄存器 .....	89
LCDCON 寄存器 .....	316	PIR1 寄存器 .....	29, 90
LCDPS 寄存器 .....	316	PIR2 寄存器 .....	29, 91
配置模块 .....	347	PIR3 寄存器 .....	92
偏置电压发生 .....	323, 324	PORTA .....	119
时钟源选择 .....	322	ANSELA 寄存器 .....	119
相关的寄存器 .....	348	带 PORTA 的配置字 .....	122
像素控制 .....	330	规范 .....	385
休眠期间的工作原理 .....	345	PORTA 寄存器 .....	29, 31
预分频器 .....	322	相关的寄存器 .....	122
帧频率 .....	330	PORTA 寄存器 .....	120
中断 .....	343	PORTB .....	123
LCDCON 寄存器 .....	316, 317	ANSELB 寄存器 .....	123
LCDCST 寄存器 .....	320	电平变化中断 .....	123
LCDDATAx 寄存器 .....	321, 328	P1B/P1C/P1D。见增强型捕捉 /	
LCDPS 寄存器 .....	316, 318	比较 /PWM+ (ECCP+) .....	123
LP 位 .....	322	PORTB 寄存器 .....	29, 31
LCDREF 寄存器 .....	319	相关的寄存器 .....	127
LCDRL 寄存器 .....	328	引脚其他功能	
LCDSEn 寄存器 .....	321	弱上拉 .....	124
LSLF .....	361	引脚说明和框图 .....	124
LSRF .....	361	PORTB 寄存器 .....	125
<b>M</b>		PORTC .....	128
MCLR .....	76	规范 .....	385
内部 .....	76	P1A。见增强型捕捉 / 比较 /PWM+ (ECCP+) .....	128
MICROCHIP 网站 .....	427	PORTC 寄存器 .....	29, 31
MOVIW .....	362	相关的寄存器 .....	130
MOVLB .....	362	引脚说明和框图 .....	128
MOVWI .....	363	PORTC 寄存器 .....	129
MPLAB 编译器、链接器和库管理器 .....	402	PORTE .....	131
MPLAB PM3 器件编程器 .....	404	PORTE 寄存器 .....	29
MPLAB REAL ICE 在线仿真器系统 .....	403	相关的寄存器 .....	132
MPLAB 集成开发环境软件 .....	401	引脚说明和框图 .....	131
MPLINK 目标链接器 /MPLIB 目标库管理器 .....	402	PORTE 寄存器 .....	131
MSSP .....	221	PR2 寄存器 .....	29, 37
SPI 模式 .....	224	PSTRxCON 寄存器 .....	220
SSPBUF 寄存器 .....	227	PWM (ECCP 模块)	
SSPSR 寄存器 .....	227	脉冲转向同步 .....	214
模数转换器。见 ADC		PWM 脉冲转向 .....	213
<b>N</b>		PWM 脉冲转向 .....	213
内部采样开关阻抗 (R <sub>ss</sub> ) .....	149	PWM 模式。参见增强型捕捉 / 比较 /PWM .....	202
内部振荡电路		PWMxCON 寄存器 .....	219
INTOSC		<b>Q</b>	
规范 .....	383	器件概述 .....	11, 99
内核寄存器 .....	29	器件配置 .....	49
<b>O</b>		代码保护 .....	53
OPTION .....	363	配置字 .....	49
OPTION 寄存器 .....	175	用户 ID .....	53, 54
OSCCON 寄存器 .....	69	欠压复位 (BOR) .....	75
OSCSTAT 寄存器 .....	70	规范 .....	387
OSCTUNE 寄存器 .....	71	时序和特性 .....	386
<b>P</b>		<b>R</b>	
P1A/P1B/P1C/P1D。参见增强型		RCREG .....	282
捕捉 / 比较 /PWM (ECCP) .....	202	RCREG 寄存器 .....	32
PCLATH 寄存器		RCSTA 寄存器 .....	32, 285
..... 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		RESET .....	363
PCL 和 PCLATH .....	18	RESET 指令 .....	76
PCL 寄存器		软件模拟器 (MPLAB SIM) .....	403
..... 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41		<b>S</b>	
PCON 寄存器 .....	30, 79	SPBRG 寄存器 .....	31
PIE1 寄存器 .....	30, 87	SPBRGH .....	287
PIE2 寄存器 .....	30, 88	SPBRG 寄存器 .....	32
		SPBRGL .....	287

SPI 模式 (MSSP)		启动条件期间的总线冲突 (仅 SDA) .....	262
SPI 时钟 .....	227	启动条件期间的总线冲突 (SCL = 0) .....	263
相关的寄存器 .....	231	启动条件期间由于 SDA 仲裁导致的 BRG 复位 .....	263
SRCON0 寄存器 .....	171	欠压复位 (BOR) .....	386
SRCON1 寄存器 .....	172	欠压复位情形 .....	75
SR 锁存器 .....	169	全桥 PWM 输出 .....	207
与 SR 锁存器模块关联的寄存器 .....	172	SPI 从模式 (CKE = 0) .....	394
SSPADD 寄存器 .....	33, 273	SPI 从模式 (CKE = 1) .....	394
SSPBUF 寄存器 .....	33	SPI 模式 (主模式) .....	227
SSPCON1 寄存器 .....	269	SPI 主模式 (CKE = 1, SMP = 1) .....	393
SSPCON2 寄存器 .....	271	时钟时序 .....	382
SSPCON3 寄存器 .....	272	时钟同步 .....	250
SSPCON 寄存器 .....	33	双速启动 .....	66
SSPMSK 寄存器 .....	273	四个占空比驱动时的 LCD 中断时序 .....	344
SSPOV .....	258	Timer0 和 Timer1 外部时钟 .....	387
SSPOV 状态标志 .....	258	Timer1 递增沿 .....	181
SSPSTAT 寄存器 .....	33, 268	停止条件期间的总线冲突 (情形 1) .....	265
R/W 位 .....	237	停止条件期间的总线冲突 (情形 2) .....	265
STATUS 寄存器 .....	22	同步发送 .....	297
SUBWFB .....	365	同步发送 (通过 TXEN) .....	297
散热考虑 (PIC16F/LF1933) .....	380	同步接收 (主模式, SREN) .....	300
闪存程序存储器 .....	103	USART 同步发送 (主/从) .....	391
擦除 .....	108	USART 同步接收 (主/从) .....	392
写 .....	108	休眠期间的自动唤醒位 (WUE) .....	294
修改 .....	112	异步发送 .....	278
上电复位 .....	74	异步发送 (背对背) .....	278
上电延时定时器 (PWRT) .....	74	异步接收 .....	282
规范 .....	387	应答序列 .....	260
时序参数符号 .....	381	增强型捕捉 / 比较 / PWM (ECCP) .....	388
时序图		占空比接近 100% 时 PWM 方向改变 .....	209
1/2 复用、1/2 偏置驱动时的 A 型波形 .....	333	正常工作期间的自动唤醒位 (WUE) .....	294
1/2 复用、1/2 偏置驱动时的 B 型波形 .....	334	中断唤醒 .....	98
1/2 复用、1/3 偏置驱动时的 A 型波形 .....	335	重复启动条件 .....	255
1/2 复用、1/3 偏置驱动时的 B 型波形 .....	336	重复启动条件期间的总线冲突 (情形 1) .....	264
1/3 复用、1/2 偏置驱动时的 A 型波形 .....	337	重复启动条件期间的总线冲突 (情形 2) .....	264
1/3 复用、1/2 偏置驱动时的 B 型波形 .....	338	自动波特率校准 .....	292
1/3 复用、1/3 偏置驱动时的 A 型波形 .....	339	时序图和规范	
1/3 复用、1/3 偏置驱动时的 B 型波形 .....	340	PLL 时钟 .....	383
1/4 复用、1/3 偏置驱动时的 A 型波形 .....	341	时序要求	
1/4 复用、1/3 偏置驱动时的 B 型波形 .....	342	I <sup>2</sup> C 总线启动位 / 停止位 .....	396, 397
A/D 转换 .....	390	SPI 模式 .....	395
A/D 转换 (休眠模式) .....	390	使用中断唤醒 .....	98
半桥 PWM 输出 .....	205, 212	时钟切换 .....	64
比较器输出 .....	159	时钟源	
CLKOUT 和 I/O .....	384	内部模式 .....	60
带时钟仲裁的波特率发生器 .....	253	HFINTOSC .....	60
第一个启动位时序 .....	254	LFINTOSC .....	61
发送和应答的总线冲突 .....	261	MFINTOSC .....	60
发送间隔字符序列 .....	295	内部振荡器时钟切换时序 .....	62
复位、看门狗定时器、振荡器上电延时定时器 .....	385	外部模式 .....	57
复位启动时序 .....	77	EC .....	57
故障保护时钟监视器 (FSCM) .....	68	HS .....	57
I <sup>2</sup> C 停止条件接收或发送模式 .....	260	LP .....	57
I <sup>2</sup> C 主模式 (7 位接收) .....	257, 259	OST .....	58
I <sup>2</sup> C 总线启动 / 停止位 .....	395	RC .....	59
I <sup>2</sup> C 总线数据 .....	396	XT .....	57
INT 引脚中断 .....	84	数据存储器 .....	21, 24
静态驱动下的类型 A/ 类型 B .....	332	数据 EEPROM 存储器 .....	103
LCD 休眠进入 / 退出 (SLPEN = 1 或 CS = 00) .....	346	代码保护 .....	104
内部振荡器切换时序 .....	63	读 .....	104
PWM 方向更改 .....	208	相关的寄存器 .....	116
PWM 输出 (低电平有效) .....	204	写 .....	104
PWM 输出 (高电平有效) .....	203	数模转换器 (DAC) .....	155
PWM 自动关闭 .....	211	复位的影响 .....	156
固件重启 .....	210	规范 .....	391

# PIC16(L)F1933

相关的寄存器 .....	158	WPUB 寄存器 .....	126
双速时钟启动模式 .....	65	WWW 地址 .....	427
<b>T</b>		WWW 在线支持 .....	9
T1CON 寄存器 .....	29, 185	温度指示器模块 .....	153
T1GCON 寄存器 .....	186	<b>X</b>	
T2CON 寄存器 .....	29, 37	写保护 .....	53
Timer0 .....	173	<b>Y</b>	
工作原理 .....	173	液晶显示驱动器 .....	315
规范 .....	388	异步操作的时钟精度 .....	284
相关的寄存器 .....	175	引脚排列说明	
Timer2 .....		PIC16F193X/PIC16LF193X .....	13
相关的寄存器 .....	192	引脚图	
Timer2/4/6 .....	189	PIC16F1933、PIC16LF1933 和 28 引脚 QFN .....	6
相关的寄存器 .....	192	PIC16F1933、PIC16LF1933 和 28 引脚	
Timer1 .....	177	PDIP/SOIC/SSOP .....	5
工作原理 .....	178	因特网地址 .....	427
规范 .....	388	<b>Z</b>	
时钟源选择 .....	178	增强型捕捉 / 比较 / PWM (ECCP) .....	193
Timer1 门控		规范 .....	388
选择信号源 .....	179	增强型 PWM 模式 .....	202
TMR1H 寄存器 .....	177	半桥模式 .....	205
TMR1L 寄存器 .....	177	半桥应用 .....	205
相关的寄存器 .....	187	半桥应用示例 .....	212
休眠期间的工作原理 .....	181	可编程死区延时 .....	212
异步计数器模式 .....	179	启动注意事项 .....	214
读写 .....	179	全桥模式 .....	206
预分频器 .....	179	全桥输出模式下的方向改变 .....	208
振荡器 .....	179	全桥应用 .....	206
中断 .....	181	输出关系 (高电平有效和低电平有效) .....	203
TMR0 寄存器 .....	29	输出关系图 .....	204
TMR1H 寄存器 .....	29	自动关闭 .....	210
TMR1L 寄存器 .....	29	自动重启 .....	211
TMR2 寄存器 .....	29, 37	直通电流 .....	212
TRIS .....	366	增强型中档 CPU .....	17
TRISA 寄存器 .....	30	增强型通用同步 / 异步收发器 (EUSART) .....	275
TRISB .....	123	振荡器	
TRISB 寄存器 .....	30	相关的寄存器 .....	71
TRISC .....	128	振荡器参数 .....	383
TRISC 寄存器 .....	30	振荡器规范 .....	382
TRISE .....	131	振荡器模块 .....	55
TRISE 寄存器 .....	30, 131	ECH .....	55
TXCON (Timer2/4/6) 寄存器 .....	191	ECL .....	55
TXREG .....	277	ECM .....	55
TXREG 寄存器 .....	32	HS .....	55
TXSTA 寄存器 .....	32, 284	INTOSC .....	55
BRGH 位 .....	287	LP .....	55
特殊功能寄存器 (SFR) .....	29	RC .....	55
特殊事件触发器 .....	143	XT .....	55
退出编程模式 .....	76	振荡器起振定时器 (OST)	
<b>U</b>		规范 .....	387
USART		振荡器切换	
同步主模式		故障保护时钟监视器 .....	67
时序图, 同步发送 .....	391	双速时钟启动 .....	65
时序图, 同步接收 .....	392	直通电流 .....	212
要求, 同步发送 .....	392	指令格式 .....	354
要求, 同步接收 .....	392	指令集 .....	353
<b>V</b>		ADDLW .....	357
VREF。见 ADC 参考电压		ADDWF .....	357
<b>W</b>		ADDWFC .....	357
WCOL .....	253, 256, 258, 260	ANDLW .....	357
WCOL 状态标志 .....	253, 256, 258, 260	ANDWF .....	357
WDTCON 寄存器 .....	101	BCF .....	358
		BRA .....	358



BSF .....	358
BTFSC .....	358
BTFSS .....	358
CALL .....	359
CALLW .....	359
CLRF .....	359
CLRWF .....	359
CLRWDI .....	359
COMF .....	359
DECF .....	359
DECFSZ .....	360
GOTO .....	360
INCF .....	360
INCFSZ .....	360
IORLW .....	360
IORWF .....	360
LSLF .....	361
LSRF .....	361
MOVF .....	361
MOVIW .....	362
MOVLB .....	362
MOVLW .....	362
MOVWF .....	362
MOVWI .....	363
NOP .....	363
OPTION .....	363
RESET .....	363
RETFIE .....	364
RETLW .....	364
RETURN .....	364
RLF .....	364
RRF .....	365
SLEEP .....	365
SWAPF .....	366
SUBLW .....	365
SUBWF .....	365
SUBWFB .....	365
TRIS .....	366
XORLW .....	366
XORWF .....	366
直流和交流特性 .....	399
直流特性	
工业级和扩展级 (PIC16F/LF1933) .....	370
扩展级和工业级 (PIC16F/LF1933) .....	377
中断 .....	81
ADC .....	144
TMR1 .....	181
与 LDO 关联的配置字 .....	95
与时钟源关联的配置字 .....	71
与中断相关的寄存器 .....	93
主同步串行端口。参见 MSSP	

# PIC16(L)F1933

---

注:

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的因特网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和样本程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及存档软件
- **一般技术支持**——常见问题 (FAQ)、技术支持请求、在线讨论组以及 Microchip 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com)。在“支持” (Support) 下，点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

# PIC16(L)F1933

---

## 读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。请填写以下信息，并从下面各方面提出您对本文档的意见。

致: TRC 经理 总页数 \_\_\_\_\_  
关于: 读者反馈  
发自: 姓名 \_\_\_\_\_  
公司 \_\_\_\_\_  
地址 \_\_\_\_\_  
国家 / 省份 / 城市 / 邮编 \_\_\_\_\_  
电话 (\_\_\_\_\_) \_\_\_\_\_ 传真 (\_\_\_\_\_) \_\_\_\_\_

应用 (选填):

您希望收到回复吗? 是\_\_\_\_ 否\_\_\_\_

器件: PIC16(L)F1933 文献编号: DS41575A\_CN

问题

1. 本文档中哪些部分最有特色?

---

---

2. 本文档是否满足了您的软硬件开发要求? 如何满足的?

---

---

3. 您认为本文档的组织结构便于理解吗? 如果不便于理解, 那么问题何在?

---

---

4. 您认为本文档应该添加哪些内容以改善其结构和主题?

---

---

5. 您认为本文档中可以删减哪些内容, 而又不会影响整体使用效果?

---

---

6. 本文档中是否存在错误或误导信息? 如果存在, 请指出是什么信息及其具体页数。

---

---

7. 您认为本文档还有哪些方面有待改进?

---

---

## 产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

部件编号	[X] <sup>(1)</sup>	-	X	/XX	XXX	示例:
器件	卷带式选项		温度范围	封装	编程模式	
<b>器件:</b>	PIC16F1933 和 PIC16LF1933					
<b>卷带式选项:</b>	空白	=	标准封装 (管状)			a) PIC16LF1933 - I/P = 工业级温度, 塑封 DIP 封装, 低压 VDD 限制。
	T	=	卷带式 <sup>(1)</sup>			b) PIC16F1933 - E/ML = 扩展级温度, QFN 封装, 标准 VDD 限制。
<b>温度范围:</b>	I	=	-40°C 至 +85°C			
	E	=	-40°C 至 +125°C			
<b>封装:</b>	ML	=	微型引线框 (QFN)			
	MV	=	微型引线框 (UQFN)			
	SO	=	SOIC			
	SP	=	窄型塑封 DIP			
	SS	=	SSOP			
<b>编程模式:</b>	3 位数字表示 QTP 代码编程模式 (空白为其他情况)					<b>注 1:</b> 卷带式标识符仅出现在目录元部件编号说明中, 此标识符用于订购用途, 不会印刷于器件封装上。请与您的销售办事处联系, 确认卷带式选项是否有货。



# MICROCHIP

## 全球销售及及服务网点

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

#### 亚特兰大 Atlanta

Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

#### 波士顿 Boston

Westborough, MA  
Tel: 1-774-760-0087  
Fax: 1-774-760-0088

#### 芝加哥 Chicago

Itasca, IL  
Tel: 1-630-285-0071  
Fax: 1-630-285-0075

#### 克里夫兰 Cleveland

Independence, OH  
Tel: 1-216-447-0464  
Fax: 1-216-447-0643

#### 达拉斯 Dallas

Addison, TX  
Tel: 1-972-818-7423  
Fax: 1-972-818-2924

#### 底特律 Detroit

Farmington Hills, MI  
Tel: 1-248-538-2250  
Fax: 1-248-538-2260

#### 印第安纳波利斯 Indianapolis

Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453

#### 洛杉矶 Los Angeles

Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

#### 圣克拉拉 Santa Clara

Santa Clara, CA  
Tel: 1-408-961-6444  
Fax: 1-408-961-6445

#### 加拿大多伦多 Toronto

Mississauga, Ontario,  
Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

#### 亚太总部 Asia Pacific Office

Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
Fax: 852-2401-3431

#### 中国 - 北京

Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

#### 中国 - 成都

Tel: 86-28-8665-5511  
Fax: 86-28-8665-7889

#### 中国 - 重庆

Tel: 86-23-8980-9588  
Fax: 86-23-8980-9500

#### 中国 - 杭州

Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

#### 中国 - 香港特别行政区

Tel: 852-2401-1200  
Fax: 852-2401-3431

#### 中国 - 南京

Tel: 86-25-8473-2460  
Fax: 86-25-8473-2470

#### 中国 - 青岛

Tel: 86-532-8502-7355  
Fax: 86-532-8502-7205

#### 中国 - 上海

Tel: 86-21-5407-5533  
Fax: 86-21-5407-5066

#### 中国 - 沈阳

Tel: 86-24-2334-2829  
Fax: 86-24-2334-2393

#### 中国 - 深圳

Tel: 86-755-8203-2660  
Fax: 86-755-8203-1760

#### 中国 - 武汉

Tel: 86-27-5980-5300  
Fax: 86-27-5980-5118

#### 中国 - 西安

Tel: 86-29-8833-7252  
Fax: 86-29-8833-7256

#### 中国 - 厦门

Tel: 86-592-238-8138  
Fax: 86-592-238-8130

#### 中国 - 珠海

Tel: 86-756-321-0040  
Fax: 86-756-321-0049

### 亚太地区

#### 台湾地区 - 高雄

Tel: 886-7-536-4818  
Fax: 886-7-330-9305

#### 台湾地区 - 台北

Tel: 886-2-2500-6610  
Fax: 886-2-2508-0102

#### 台湾地区 - 新竹

Tel: 886-3-5778-3666  
Fax: 886-3-5770-9555

#### 澳大利亚 Australia - Sydney

Tel: 61-2-9868-6733  
Fax: 61-2-9868-6755

#### 印度 India - Bangalore

Tel: 91-80-3090-4444  
Fax: 91-80-3090-4123

#### 印度 India - New Delhi

Tel: 91-11-4160-8631  
Fax: 91-11-4160-8632

#### 印度 India - Pune

Tel: 91-20-2566-1512  
Fax: 91-20-2566-1513

#### 日本 Japan - Yokohama

Tel: 81-45-471-6166  
Fax: 81-45-471-6122

#### 韩国 Korea - Daegu

Tel: 82-53-744-4301  
Fax: 82-53-744-4302

#### 韩国 Korea - Seoul

Tel: 82-2-554-7200  
Fax: 82-2-558-5932 或  
82-2-558-5934

#### 马来西亚 Malaysia - Kuala Lumpur

Tel: 60-3-6201-9857  
Fax: 60-3-6201-9859

#### 马来西亚 Malaysia - Penang

Tel: 60-4-227-8870  
Fax: 60-4-227-4068

#### 菲律宾 Philippines - Manila

Tel: 63-2-634-9065  
Fax: 63-2-634-9069

#### 新加坡 Singapore

Tel: 65-6334-8870  
Fax: 65-6334-8850

#### 泰国 Thailand - Bangkok

Tel: 66-2-694-1351  
Fax: 66-2-694-1350

### 欧洲

#### 奥地利 Austria - Wels

Tel: 43-7242-2244-39  
Fax: 43-7242-2244-393

#### 丹麦 Denmark - Copenhagen

Tel: 45-4450-2828  
Fax: 45-4485-2829

#### 法国 France - Paris

Tel: 33-1-69-53-63-20  
Fax: 33-1-69-30-90-79

#### 德国 Germany - Munich

Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

#### 意大利 Italy - Milan

Tel: 39-0331-742611  
Fax: 39-0331-466781

#### 荷兰 Netherlands - Drunen

Tel: 31-416-690399  
Fax: 31-416-690340

#### 西班牙 Spain - Madrid

Tel: 34-91-708-08-90  
Fax: 34-91-708-08-91

#### 英国 UK - Wokingham

Tel: 44-118-921-5869  
Fax: 44-118-921-5820

08/02/11