ENROO 英锐恩

单片机集成方案全方位解决服务商

优质智能电子产品"芯"方案解决商

Microchip

PIC16F631/677/685/687/689/690产品用户参考手册

深圳市英锐恩科技有限公司
ENROO-TECH（SHENZHEN）CO.,LTD
中国·深圳市福田区福华路嘉汇新城汇商中心27楼2701

Enroo-Tech Technologies CO., Limited
香港新界荃灣青山道388號中國染廠大廈7樓P室

联系电话：86-755-82543411,83167411,83283911, 88845951
联系传真：86-755-82543511
全国热线：4007-888-234
联系邮件：enroo@enroo.com
公司网站：http://www.enroo.com　　http://www.enroo-tech.com

英锐恩为 MICROCHIP 授权 PARTNER；代理销售全系列 MICROCHIP 单片机及模拟器件、接口电路。我们时刻为您提供技术支持;提供开发应用协助;提供成熟方案;提供免费样品;提供详细资料。

RECOGNIZED THIRD-PARTY TOOL PROVIDER
MICROCHIP

MICROCHIP AUTHORIZED DESIGN PARTNER

PIC16F631/677/685/687/689/690

采用纳瓦技术的 20 引脚 8 位 CMOS 闪存单片机

高性能 RISC CPU：

- 仅需学习 35 条指令：
  - 除了跳转指令以外，所有指令都是单周期的
- 工作速度：
  - DC － 20 MHz 振荡器 / 时钟输入
  - DC － 200 ns 指令周期
- 中断能力
- 8 级深硬件堆栈
- 直接、间接和相对寻址模式

单片机特性：

- 精确的内部振荡器：
  - 出厂时精度已校准到 ±1%
  - 软件可选择频率范围：8 MHz 到 32 kHz
  - 可用软件调整
  - 双速启动模式
  - 用于关键应用的晶振故障检测
  - 工作时切换时钟模式以节电
- 节能休眠模式
- 宽工作电压范围 （2.0V-5.5V）
- 工业级和扩展级温度范围
- 上电复位 （Power-on Reset， POR）
- 上电延时定时器 （Power-up Timer， PWRTE）和振荡器起振定时器 （Oscillator Start-up Timer，OST）
- 带软件控制选择的欠压复位 （Brown-out Reset，BOR）
- 增强型低电流看门狗定时器 （Watchdog Timer，WDT），带有片上振荡器 （预分频器最大时，软件可选择的标称值为 268 秒），可用软件使能
- 复用式主复位 / 输入引脚
- 可编程代码保护
- 高耐久性的闪存 / EEPROM 存储单元：
  - 闪存耐写次数达 100,000 次
  - EEPROM 耐写次数达 1,000,000 次
  - 闪存 / 数据 EEPROM 的数据保持期：> 40 年
- 增强型 USART 模块：
  - 支持 RS-485、 RS-232 和 LIN 2.0
  - 自动波特率检测
  - 检测到启动位自动唤醒

低功耗特性：

- 待机电流：
  - 2.0V 时典型值为 50 nA
- 工作电流：
  - 32 kHz、 2.0V 时典型值为 11 μA
  - 4 MHz、 2.0V 时典型值为 220 μA
- 看门狗定时器电流：
  - 2.0V 时典型值为 <1 μA

外设特性：

- 17 个 I/O 引脚和 1 个只用作输入的引脚：
  - 高灌 / 拉电流能力，可直接驱动 LED
  - 电平变化中断引脚
  - 可单独编程的弱上拉
  - 超低功耗唤醒 （Ultra Low-Power Wake-up，ULPWU）
- 模拟比较器模块，具有：
  - 2 个模拟比较器
  - 片上可编程参考电压 （CVREF）模块 （VDD 的 %）
  - 可从外部访问比较器输入和输出
  - SR 锁存器模式
  - Timer1 门控同步锁存器
  - 固定 0.6V 的 VREF
- A/D 转换器：
  - 10 位分辨率和 12 路通道
- Timer0：带有 8 位可编程预分频器的 8 位定时器 / 计数器
- 增强型 Timer1：
  - 带有预分频器的 16 位定时器 / 计数器
  - 外部 Timer1 门控 （计数使能）
  - 如果选用 INTOSC 模式，可选择 OSC1 和 OSC2 之间的振荡器 （工作在 LP 模式下）作为 Timer1 的振荡器
- Timer2：带有 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
- 增强型捕捉、比较和 PWM+ 模块：
  - 16 位捕捉，最大精度 12.5 ns
  - 比较，最大精度 200 ns
  - 10 位 PWM，带有 1、 2 或 4 路输出通道及可编程 “死区”，最高频率 20 kHz
  - PWM 输出转向 （Steering）控制
- 同步串行口 （Synchronous Serial Port， SSP）：
  - SPI 模式 （主 / 从）
- I²C™ （主 / 从模式）：
  - I²C™ 地址屏蔽
- 通过两个引脚进行在线串行编程 （In-Circuit Serial Programming™， ICSP™）

器件	程序存储器	数据存储器		I/O	10 位 A/D（通道）	比较器	8/16 位定时器	SSP	ECCP+	EUSART
	闪存（字数）	SRAM（字节数）	EEPROM（字节数）							
PIC16F631	1024	64	128	18	—	2	1/1	无	无	无
PIC16F677	2048	128	256	18	12	2	1/1	有	无	无
PIC16F685	4096	256	256	18	12	2	2/1	无	有	无
PIC16F687	2048	128	256	18	12	2	1/1	有	无	有
PIC16F689	4096	256	256	18	12	2	1/1	有	无	有
PIC16F690	4096	256	256	18	12	2	2/1	有	有	有

PIC16F631 引脚图

20 引脚 PDIP、SOIC 和 SSOP

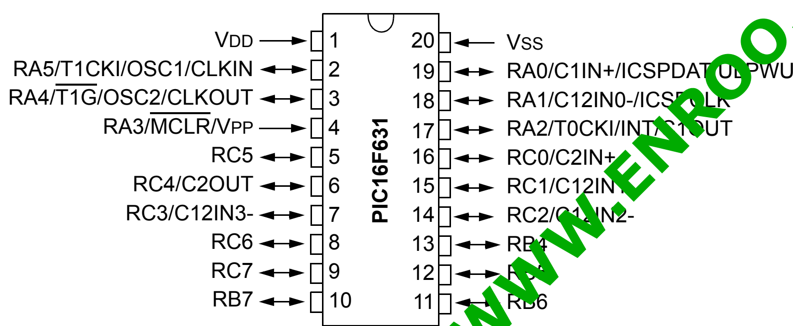


表 1： PIC16F631 引脚汇总

I/O	引脚	模拟	比较器	定时器	中断	上拉	基本
RA0	19	AN0/ULPWU	C1IN+	—	IOC	有	ICSPDAT
RA1	18	AN1	C12IN0-	—	IOC	有	ICSPCLK
RA2	17	—	C1OUT	T0CKI	IOC/INT	有	—
RA3	4	—	—	—	IOC	有 (1)	MCLR/VPP
RA4	3	—	—	T1G	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	IOC	有	OSC1/CLKIN
RB4	13	—	—	—	IOC	有	—
RB5	12	—	—	—	IOC	有	—
RB6	11	—	—	—	IOC	有	—
RB7	10	—	—	—	IOC	有	—
RC0	16	AN4	C2IN+	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—
RC2	14	AN6	C12IN2-	—	—	—	—
RC3	7	AN7	C12IN3-	—	—	—	—
RC4	6	—	C2OUT	—	—	—	—
RC5	5	—	—	—	—	—	—
RC6	8	—	—	—	—	—	—
RC7	9	—	—	—	—	—	—
—	1	—	—	—	—	—	VDD
—	20	—	—	—	—	—	VSS

注 1： 只有配置为外部 MCLR 时才使能上拉。

PIC16F677 引脚图

20 引脚 PDIP、SOIC 和 SSOP

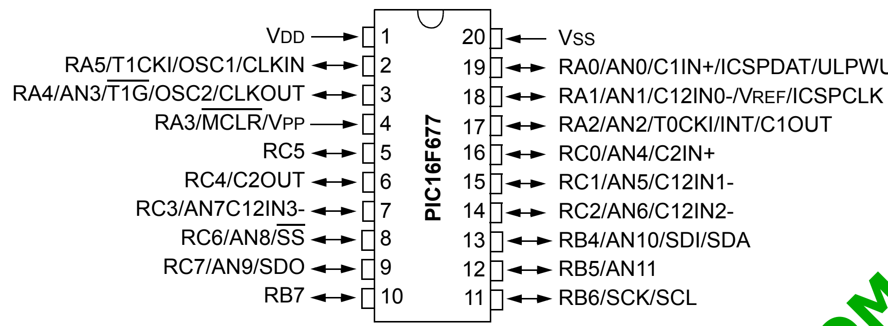


表 2: PIC16F677 引脚汇总

I/O	引脚	模拟	比较器	定时器	SSP	中断	上拉	基本
RA0	19	AN0/ULPWU	C1IN+	—	—	IOC	有	ICSPDAT
RA1	18	AN1/VREF	C12IN0-	—	—	IOC	有	ICSPCLK
RA2	17	AN2	C1OUT	T0CKI	—	IOC/INT	有	—
RA3	4	—	—	—	—	IOC	有 (1)	$\overline{\text{MCLR}}$/VPP
RA4	3	AN3	—	$\overline{\text{T1G}}$	—	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	IOC	有	OSC1/CLKIN
RB4	13	AN10	—	—	SDI/SDA	IOC	有	—
RB5	12	AN11	—	—	—	IOC	有	—
RB6	11	—	—	—	SCL/SCK	IOC	有	—
RB7	10	—	—	—	—	IOC	有	—
RC0	16	AN4	C2IN+	—	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—	—
RC2	14	AN6	C12IN2-	—	—	—	—	—
RC3	7	AN7	C12IN3-	—	—	—	—	—
RC4	6	—	C2OUT	—	—	—	—	—
RC5	5	—	—	—	—	—	—	—
RC6	8	AN8	—	—	$\overline{\text{SS}}$	—	—	—
RC7	9	AN9	—	—	SDO	—	—	—
—	1	—	—	—	—	—	—	VDD
—	20	—	—	—	—	—	—	VSS

注 1: 只有配置为外部 $\overline{\text{MCLR}}$ 时才激活上拉。

PIC16F685 引脚图

20 引脚 PDIP、SOIC 和 SSOP

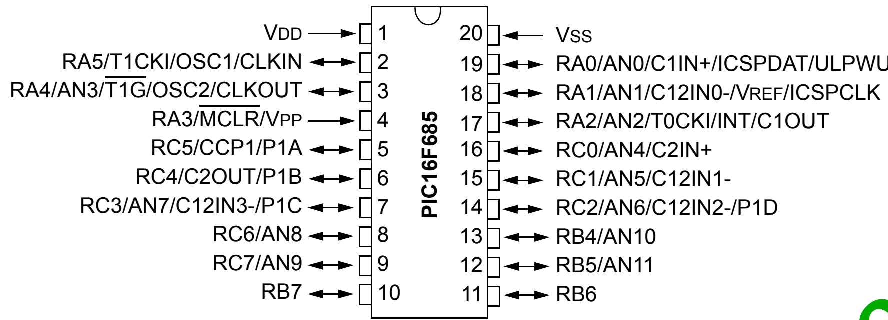


表 3： PIC16F685 引脚汇总

I/O	引脚	模拟	比较器	定时器	ECCP	中断	上拉	基本
RA0	19	AN0/ULPWU	C1IN+	—	—	IOC	有	ICSPDAT
RA1	18	AN1/VREF	C12IN0-	—	—	IOC	有	ICSPCLK
RA2	17	AN2	C1OUT	T0CKI	—	IOC/INT	有	—
RA3	4	—	—	—	—	IOC	有 [(1)]	MCLR/VPP
RA4	3	AN3	—	T1G	—	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	IOC	有	OSC1/CLKIN
RB4	13	AN10	—	—	—	IOC	有	—
RB5	12	AN11	—	—	—	IOC	有	—
RB6	11	—	—	—	—	IOC	有	—
RB7	10	—	—	—	—	IOC	有	—
RC0	16	AN4	C2IN+	—	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—	—
RC2	14	AN6	C12IN2-	—	P1D	—	—	—
RC3	7	AN7	C12IN3-	—	P1C	—	—	—
RC4	6	—	C2OUT	—	P1B	—	—	—
RC5	5	—	—	—	CCP1/P1A	—	—	—
RC6	8	AN8	—	—	—	—	—	—
RC7	9	AN9	—	—	—	—	—	—
—	1	—	—	—	—	—	—	VDD
—	20	—	—	—	—	—	—	VSS

注 1： 只有配置为外部 MCLR 时才激活上拉。

PIC16F687/689 引脚图

20 引脚 PDIP、SOIC 和 SSOP

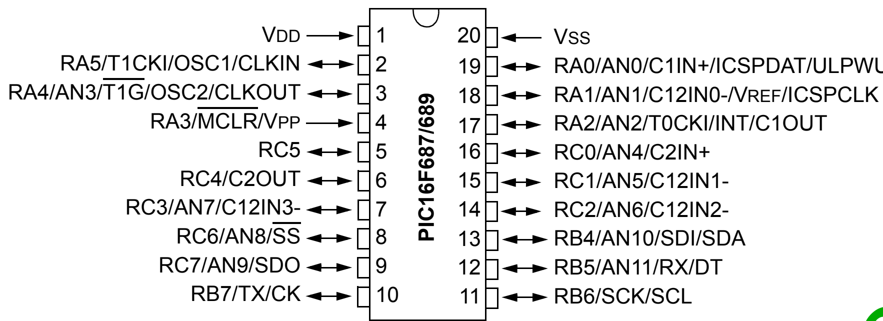


表 4： PIC16F687/689 引脚汇总

I/O	引脚	模拟	比较器	定时器	EUSART	SSP	中断	上拉	基本
RA0	19	AN0/ULPWU	C1IN+	—	—	—	IOC	有	ICSPDAT
RA1	18	AN1/VREF	C12IN0-		—	—	IOC	有	ICSPCLK
RA2	17	AN2	C1OUT	T0CKI	—	—	IOC/INT	有	—
RA3	4	—	—	—	—	—	IOC	有 (1)	$\overline{\text{MCLR}}$/VPP
RA4	3	AN3	—	$\overline{\text{T1G}}$	—	—	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	—	IOC	有	OSC1/CLKIN
RB4	13	AN10	—	—	—	SDI/SDA	IOC	有	—
RB5	12	AN11	—	—	RX/DT	—	IOC	有	—
RB6	11	—	—	—	—	SCL/SCK	IOC	有	—
RB7	10	—	—	—	TX/CK	—	IOC	有	—
RC0	16	AN4	C2IN+	—	—	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—	—	—
RC2	14	AN6	C12IN2-	—	—	—	—	—	—
RC3	7	AN7	C12IN3-	—	—	—	—	—	—
RC4	6	—	C2OUT	—	—	—	—	—	—
RC5	5	—	—	—	—	—	—	—	—
RC6	8	AN8	—	—	—	$\overline{\text{SS}}$	—	—	—
RC7	9	AN9	—	—	—	SDO	—	—	—
—	1	—	—	—	—	—	—	—	VDD
—	20	—	—	—	—	—	—	—	VSS

注 1： 只有配置为外部 $\overline{\text{MCLR}}$ 时才激活上拉。

PIC16F690 引脚图 （PDIP、 SOIC 和 SSOP）

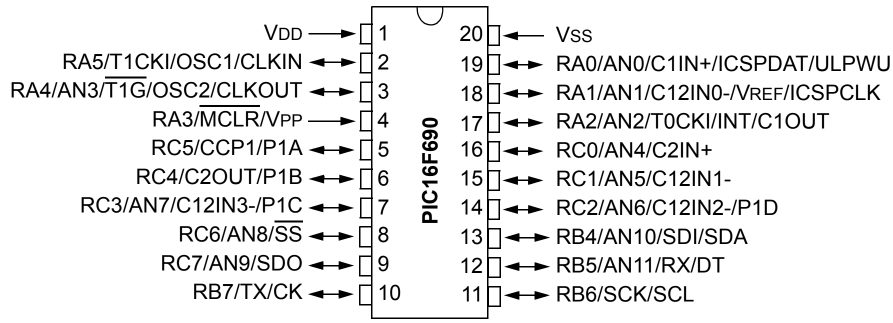


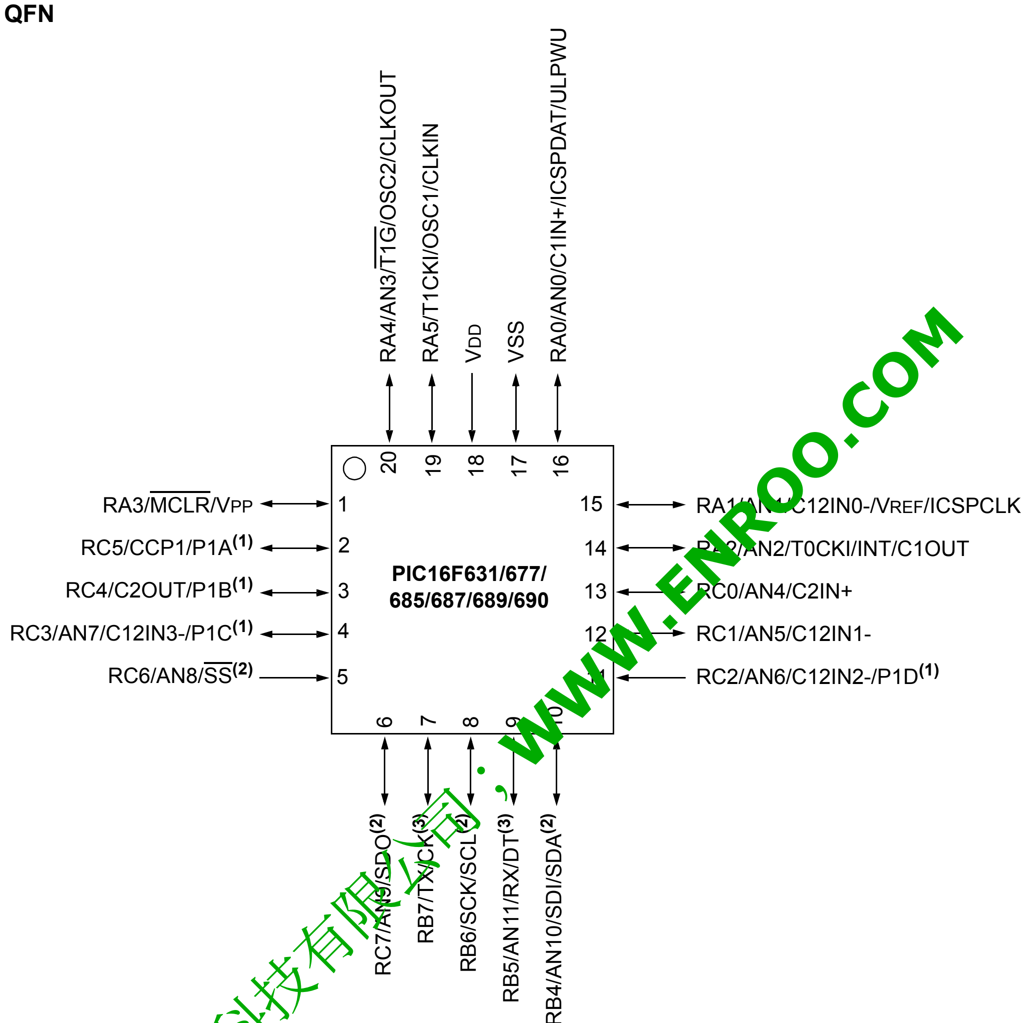
表 5： PIC16F690 引脚汇总

I/O	引脚	模拟	比较器	定时器	ECCP	EUSART	SSP	中断	上拉	基本
RA0	19	AN0/ULPWU	C1IN+	—	—	—	—	IOC	有	ICSPDAT
RA1	18	AN1/VREF	C12IN0-	—	—	—	—	IOC	有	ICSPCLK
RA2	17	AN2	C1OUT	T0CKI	—	—	—	IOC/INT	有	—
RA3	4	—	—	—	—	—	—	IOC	有 (1)	MCLR/VPP
RA4	3	AN3	—	T1G	—	—	—	IOC	有	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	—	—	IOC	有	OSC1/CLKIN
RB4	13	AN10	—	—	—	—	SDI/SDA	IOC	有	—
RB5	12	AN11	—	—	—	RX/DT	—	IOC	有	—
RB6	11	—	—	—	—	—	SCL/SCK	IOC	有	—
RB7	10	—	—	—	—	TX/CK	—	IOC	有	—
RC0	16	AN4	C2IN+	—	—	—	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—	—	—	—
RC2	14	AN6	C12IN2-	—	P1D	—	—	—	—	—
RC3	7	AN7	C12IN3-	—	P1C	—	—	—	—	—
RC4	6	—	C2OUT	—	P1B	—	—	—	—	—
RC5	5	—	—	—	CCP1/P1A	—	—	—	—	—
RC6	8	AN8	—	—	—	—	SS	—	—	—
RC7	9	AN9	—	—	—	—	SDO	—	—	—
—	1	—	—	—	—	—	—	—	—	VDD
—	20	—	—	—	—	—	—	—	—	VSS

注 1： 只有配置为外部 MCLR 时才激活上拉。

PIC16F631/677/685/687/689/690 引脚图 （QFN）

20 引脚 QFN



注 1： 只有 PIC16F685/PIC16F690 上有 CCP1/P1A、 P1B、 P1C 和 P1D。

2： 只有 PIC16F677/PIC16F687/PIC16F689/PIC16F690 上有 $\overline{\text{SS}}$、 SDO、 SDI/SDA 和 SCL/SCK。

3： 只有 PIC16F687/PIC16F689/PIC16F690 上有 RX/DT 和 TX/CK。

目录

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**，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** 上注册。

1.0 器件概述

本数据手册涵盖 PIC16F631/677/685/687/689/690 器件。其封装形式为 20 引脚 PDIP、 SOIC、 TSSOP 和 QFN 封装。

这些器件的框图和引脚说明如下：

- PIC16F631 （图 1-1，表 1-1）
- PIC16F677 （图 1-2，表 1-2）
- PIC16F685 （图 1-3，表 1-3）
- PIC16F687/PIC16F689 （图 1-4，表 1-4）
- PIC16F690 （图 1-5，表 1-5）

图 1-1： **PIC16F631 框图**

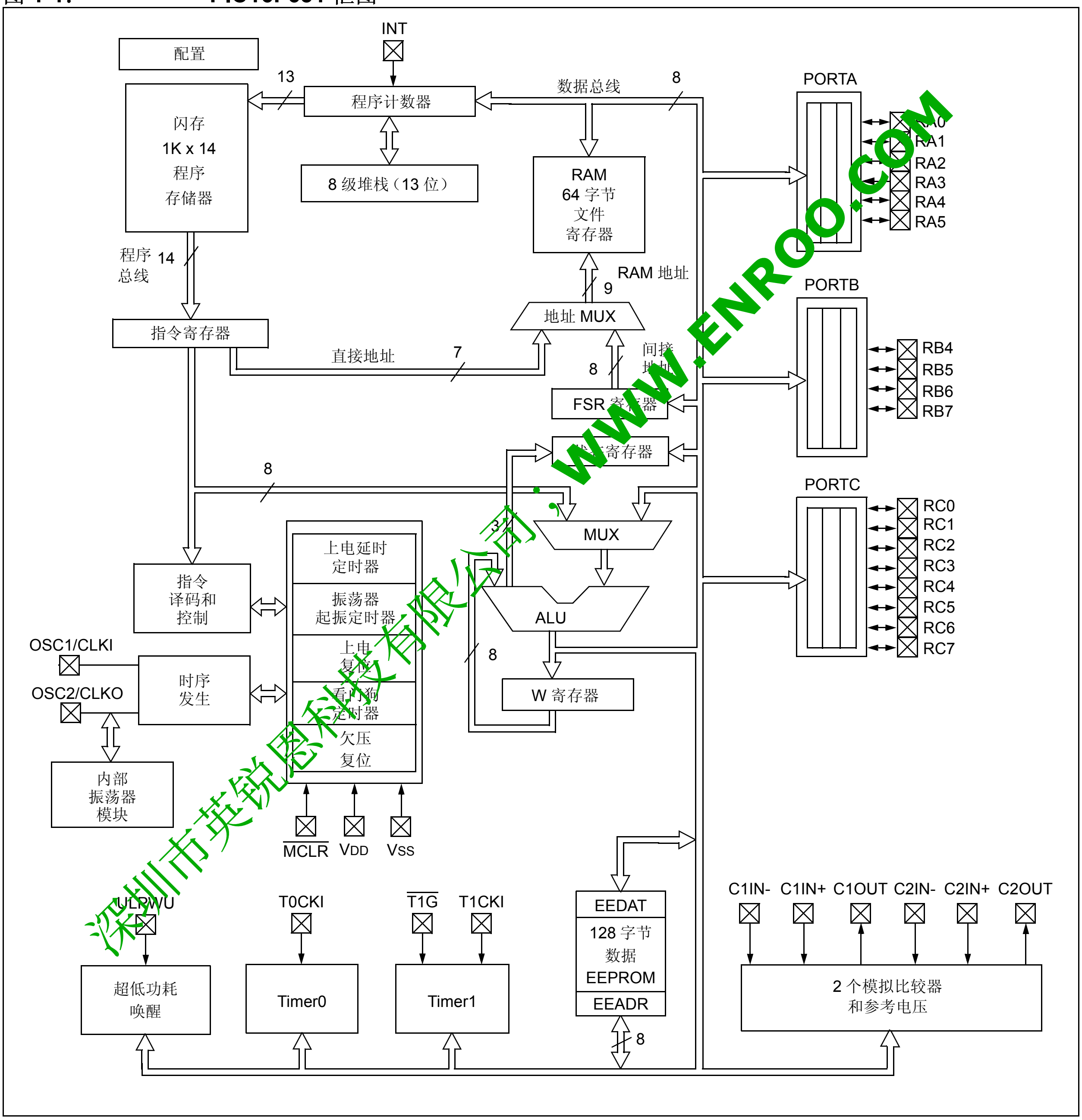


图 1-2： PIC16F677 框图

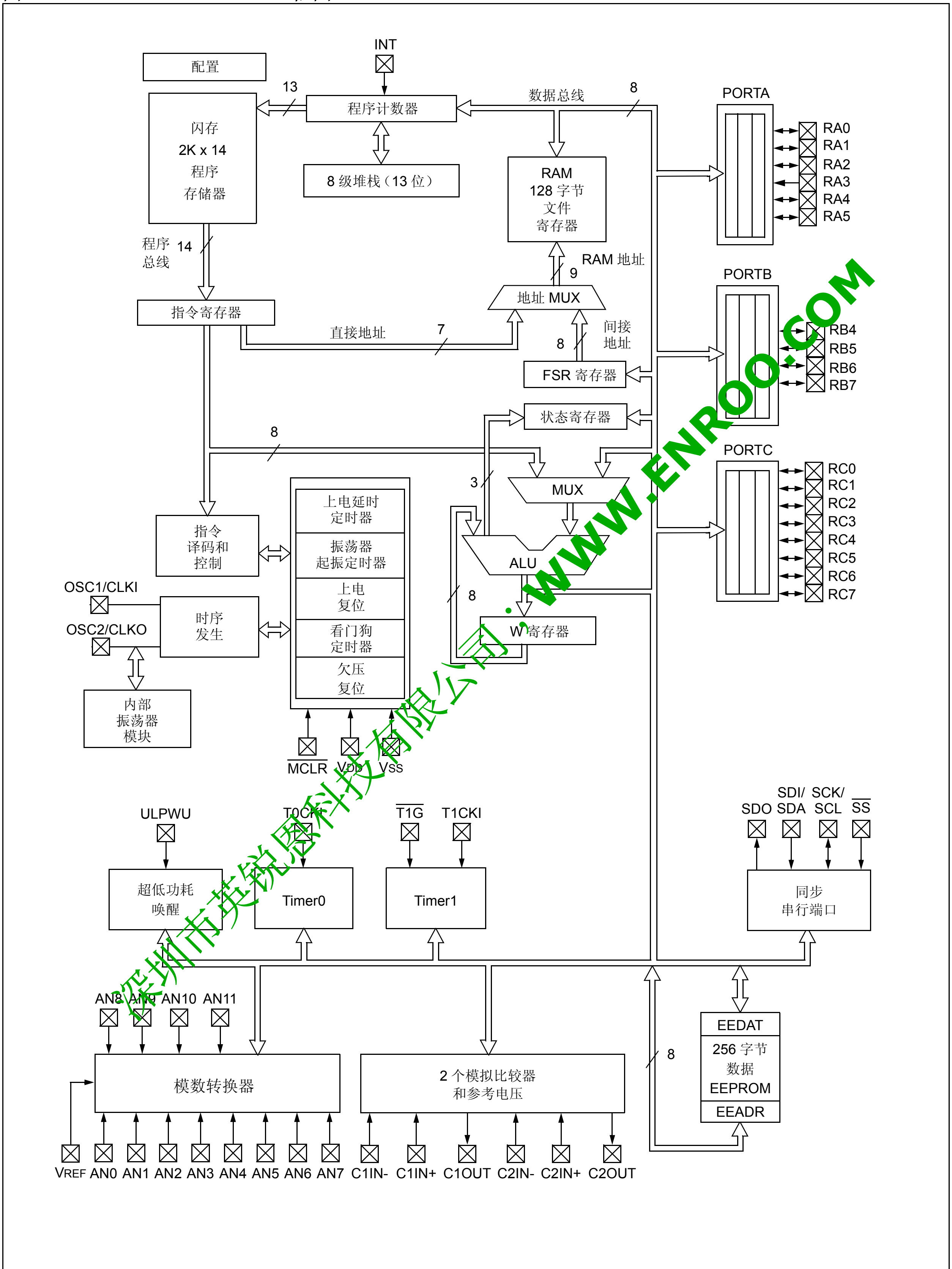


图 1-3： PIC16F685 框图

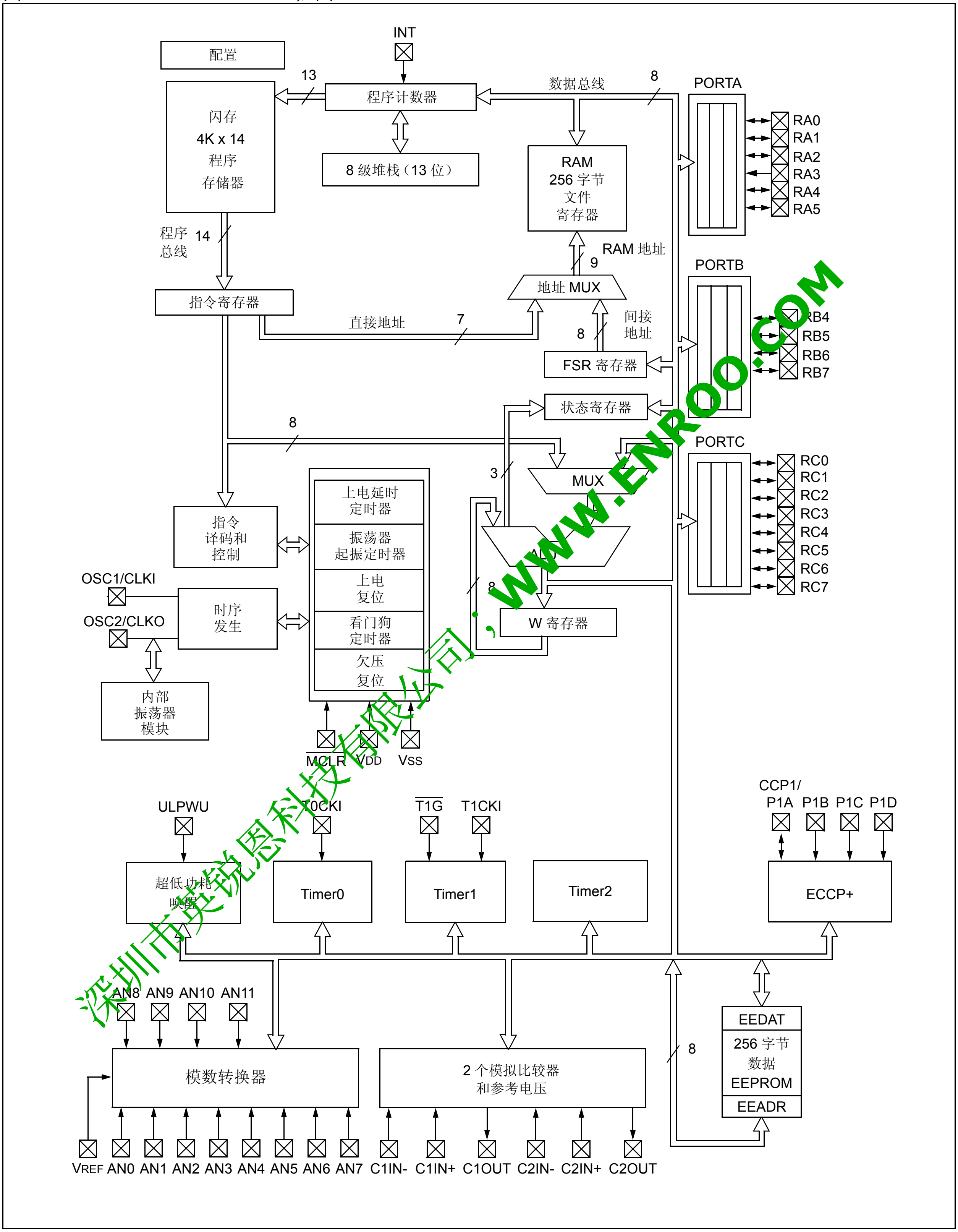
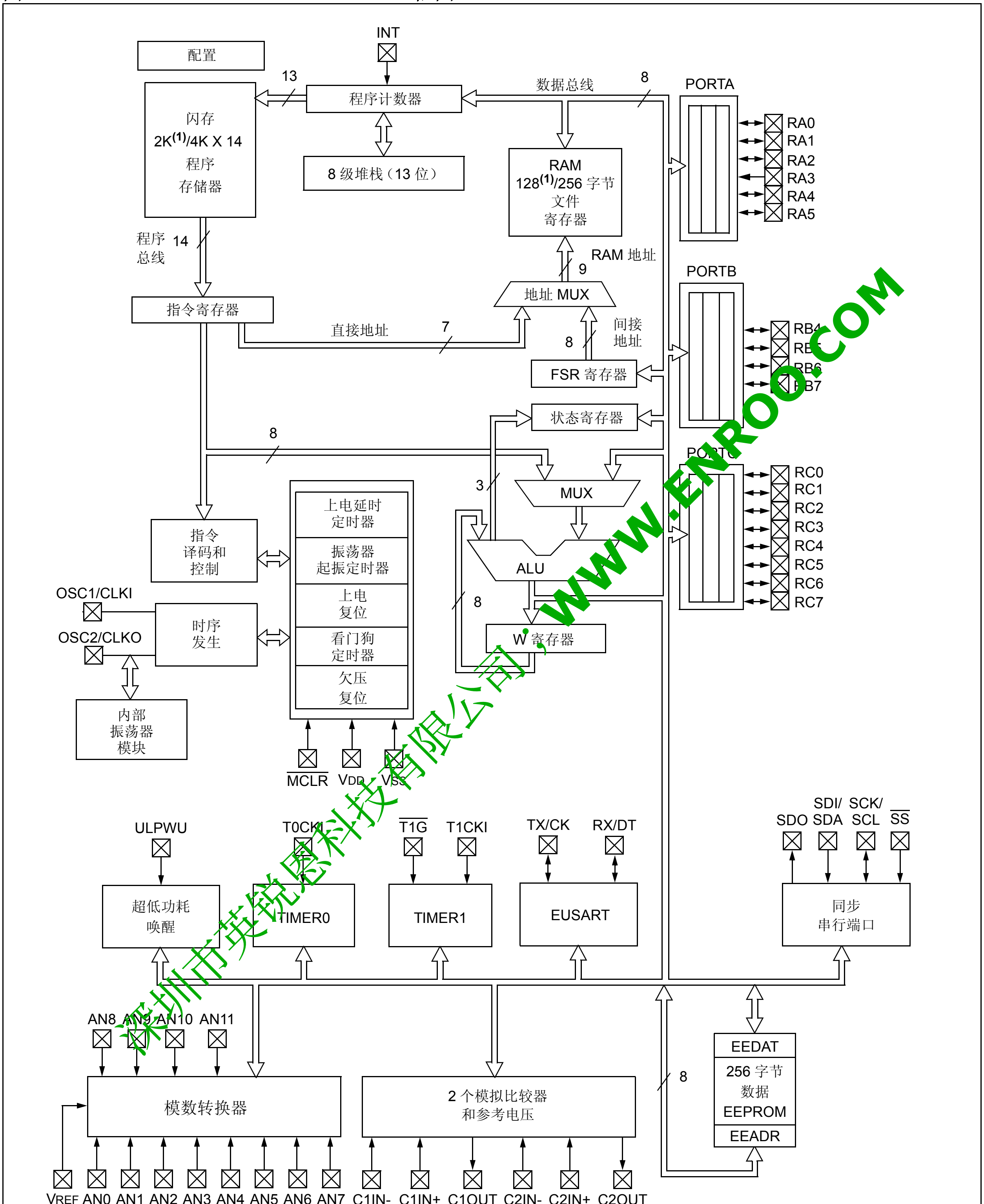


图 1-4： PIC16F687/PIC16F689 框图



注 1： 仅限 PIC16F687。

图 1-5： PIC16F690 框图

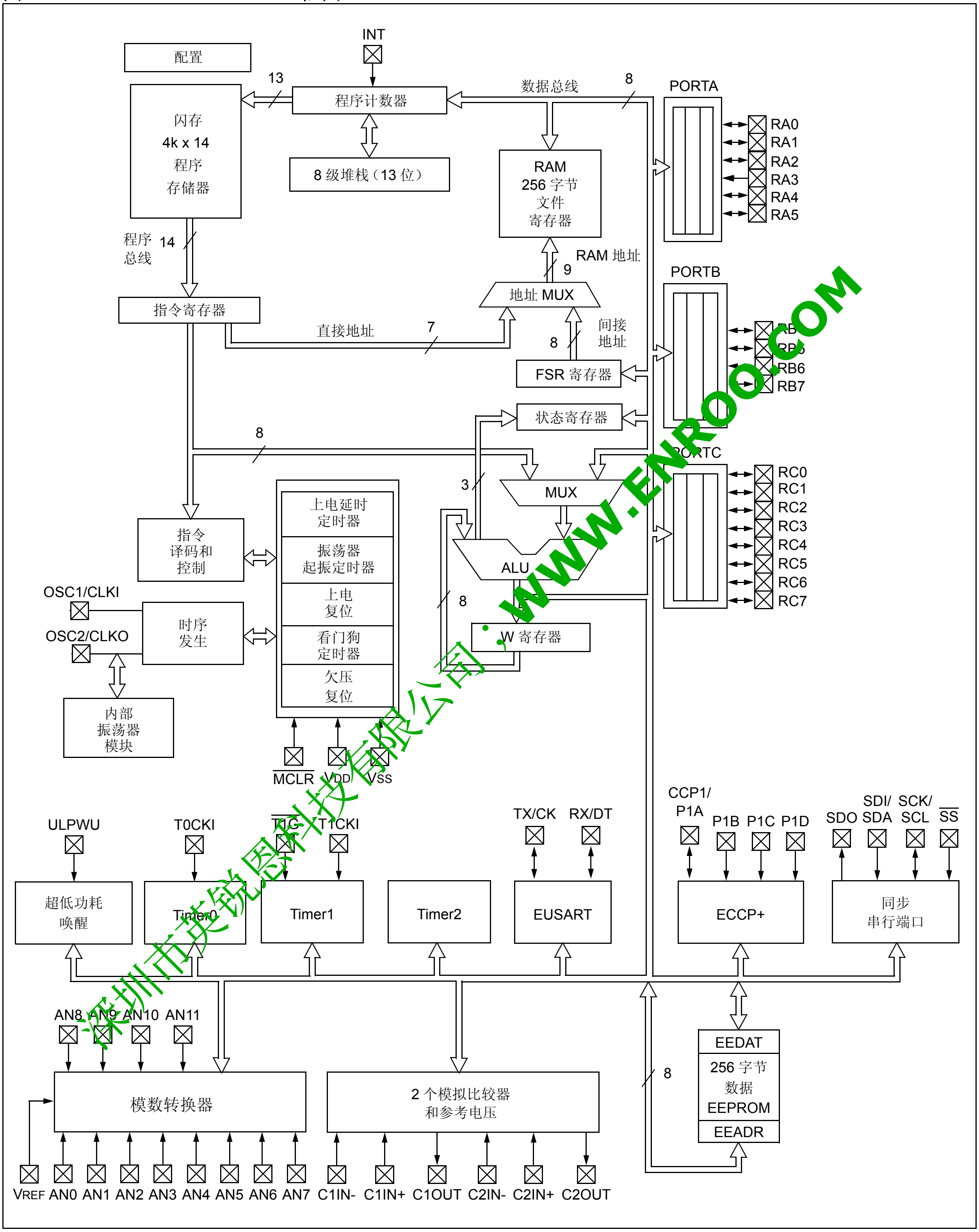


表 **1-1**： 引脚说明——**PIC16F631**

名称	功能	输入类型	输出类型	说明
RA0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	C1IN+	AN	—	比较器 C1 的同相输入。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
	ULPWU	AN	—	超低功耗唤醒输入。
RA1/C12IN0-/ICSPCLK	RA1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	C12IN0-	AN	—	比较器 C1 或 C2 的反相输入。
	ICSPCLK	ST	—	ICSP™ 时钟。
RA2/T0CKI/INT/C1OUT	RA2	ST	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T0CKI	ST	—	Timer0 的时钟输入。
	INT	ST	—	外部中断引脚。
	C1OUT	—	CMOS	比较器 C1 的输出。
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	通用输入。可单独控制电平变化中断。
	$\overline{\text{MCLR}}$	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/$\overline{\text{T1G}}$/OSC2/CLKOUT	RA4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	$\overline{\text{T1G}}$	ST	—	Timer1 门控输入。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	FOSC/4 输出。
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T1CKI	ST	—	Timer1 的时钟输入。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB4	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RB5	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RB6	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RB7	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RC0/C2IN+	RC0	ST	CMOS	通用 I/O。
	C2IN+	AN	—	比较器 C2 的同相输入。
RC1/C12IN1-	RC1	ST	CMOS	通用 I/O。
	C12IN1-	AN	—	比较器 C1 或 C2 的反相输入。
RC2/C12IN2-	RC2	ST	CMOS	通用 I/O。
	C12IN2-	AN	—	比较器 C1 或 C2 的反相输入。
RC3/C12IN3-	RC3	ST	CMOS	通用 I/O。
	C12IN3-	AN	—	比较器 C1 或 C2 的反相输入。
RC4/C2OUT	RC4	ST	CMOS	通用 I/O。
	C2OUT	—	CMOS	比较器 C2 的输出。
RC5	RC5	ST	CMOS	通用 I/O。

图注： AN = 模拟输入或输出　　CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入　　ST = CMOS 电平的施密特触发器输入
HV = 高压　　XTAL = 晶振

表 1-1： 引脚说明——PIC16F631 （续）

名称	功能	输入类型	输出类型	说明
RC6	RC6	ST	CMOS	通用 I/O。
RC7	RC7	ST	CMOS	通用 I/O。
VSS	VSS	电源	—	参考地。
VDD	VDD	电源	—	正电源。

图注： AN = 模拟输入或输出 CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入 ST = CMOS 电平的施密特触发器输入
HV = 高压 XTAL = 晶振

表 **1-2**： 引脚说明——**PIC16F677**

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN0	AN	—	A/D 通道 0 的输入。
	C1IN+	AN	—	比较器 C1 的同相输入。
	ICSPDAT	ST	CMOS	ICSP™ 数据 I/O。
	ULPWU	AN	—	超低功耗唤醒输入。
RA1/AN1/C12IN0-/VREF/ICSPCLK	RA1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN1	AN	—	A/D 通道 1 的输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的反相输入。
	VREF	AN	—	A/D 的外部参考电压。
	ICSPCLK	ST	—	ICSP™ 时钟。
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN2	AN	—	A/D 通道 2 的输入。
	T0CKI	ST	—	Timer0 的时钟输入。
	INT	ST	—	外部中断引脚。
	C1OUT	—	CMOS	比较器 C1 的输出。
RA3/$\overline{\text{MCLR}}$/VPP	RA3	TTL	—	通用输入。可单独控制电平变化中断。
	$\overline{\text{MCLR}}$	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT	RA4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN3	AN	—	A/D 通道 3 的输入。
	$\overline{\text{T1G}}$	ST	—	Timer1 门控输入。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	FOSC/4 输出。
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T1CKI	ST	—	Timer1 的时钟输入。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN10	AN	—	A/D 通道 10 的输入。
	SDI	ST	—	SPI 数据输入。
	SDA	ST	OD	I^2C™ 数据输入 / 输出。
RB5/AN11	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN11	AN	—	A/D 通道 11 的输入。
RB6/SCK/SCL	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	SCK	ST	CMOS	SPI 时钟。
	SCL	ST	OD	I^2C™ 时钟。

图注： AN = 模拟输入或输出　　CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入　　ST = CMOS 电平的施密特触发器输入
HV = 高压　　XTAL = 晶振

表 1-2： 引脚说明——PIC16F677 （续）

名称	功能	输入类型	输出类型	说明
RB7	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RC0/AN4/C2IN+	RC0	ST	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4 的输入。
	C2IN+	AN	—	比较器 C2 的同相输入。
RC1/AN5/C12IN1-	RC1	ST	CMOS	通用 I/O。
	AN5	AN	—	A/D 通道 5 的输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的反相输入。
RC2/AN6/C12IN2-	RC2	ST	CMOS	通用 I/O。
	AN6	AN	—	A/D 通道 6 的输入。
	C12IN2-	AN	—	比较器 C1 或 C2 的反相输入。
RC3/AN7/C12IN3-	RC3	ST	CMOS	通用 I/O。
	AN7	AN	—	A/D 通道 7 的输入。
	C12IN3-	AN	—	比较器 C1 或 C2 的反相输入。
RC4/C2OUT	RC4	ST	CMOS	通用 I/O。
	C2OUT	—	CMOS	比较器 C2 的输出。
RC5	RC5	ST	CMOS	通用 I/O。
RC6/AN8/$\overline{SS}$	RC6	ST	CMOS	通用 I/O。
	AN8	AN	—	A/D 通道 8 的输入。
	$\overline{SS}$	ST	—	从选择输入。
RC7/AN9/SDO	RC7	ST	CMOS	通用 I/O。
	AN9	AN	—	A/D 通道 9 的输入。
	SDO	—	CMOS	SPI 数据输出。
VSS	VSS	电源	—	参考地。
VDD	VDD	电源	—	正电源。

图注： AN = 模拟输入或输出　　CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入　　ST = CMOS 电平的施密特触发器输入
HV = 高压　　XTAL = 晶振

表 1-3： 引脚说明——PIC16F685

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN0	AN	—	A/D 通道 0 的输入。
	C1IN+	AN	—	比较器 C1 的同相输入。
	ICSPDAT	TTL	CMOS	ICSP™ 数据 I/O。
	ULPWU	AN	—	超低功耗唤醒输入。
RA1/AN1/C12IN0-/VREF/ICSPCLK	RA1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN1	AN	—	A/D 通道 1 的输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的反相输入。
	VREF	AN	—	A/D 的外部参考电压。
	ICSPCLK	ST	—	ICSP™ 时钟。
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN2	AN	—	A/D 通道 2 的输入。
	T0CKI	ST	—	Timer0 的时钟输入。
	INT	ST	—	外部中断引脚。
	C1OUT	—	CMOS	比较器 C1 的输出。
RA3/$\overline{MCLR}$/VPP	RA3	TTL	—	通用输入。可单独控制电平变化中断。
	$\overline{MCLR}$	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	RA4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN3	AN	—	A/D 通道 3 的输入。
	$\overline{T1G}$	ST	—	Timer1 门控输入。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	FOSC/4 输出。
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T1CKI	ST	—	Timer1 的时钟输入。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB4/AN10	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN10	AN	—	A/D 通道 10 的输入。
RB5/AN11	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN11	AN	—	A/D 通道 11 的输入。
RB6	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RB7	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
RC0/AN4/C2IN+	RC0	ST	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4 的输入。
	C2IN+	AN	—	比较器 C2 的同相输入。

图注： AN = 模拟输入或输出　CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入　ST = CMOS 电平的施密特触发器输入
HV = 高压　XTAL = 晶振

表 1-3： 引脚说明——PIC16F685 （续）

名称	功能	输入类型	输出类型	说明
RC1/AN5/C12IN1-	RC1	ST	CMOS	通用 I/O。
	AN5	AN	—	A/D 通道 5 的输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的反相输入。
RC2/AN6/C12IN2-/P1D	RC2	ST	CMOS	通用 I/O。
	AN6	AN	—	A/D 通道 6 的输入。
	C12IN2-	AN	—	比较器 C1 或 C2 的反相输入。
	P1D	—	CMOS	PWM 的输出。
RC3/AN7/C12IN3-/P1C	RC3	ST	CMOS	通用 I/O。
	AN7	AN	—	A/D 通道 7 的输入。
	C12IN3-	AN	—	比较器 C1 或 C2 的反相输入。
	P1C	—	CMOS	PWM 的输出。
RC4/C2OUT/P1B	RC4	ST	CMOS	通用 I/O。
	C2OUT	—	CMOS	比较器 C2 的输出。
	P1B	—	CMOS	PWM 的输出。
RC5/CCP1/P1A	RC5	ST	CMOS	通用 I/O。
	CCP1	ST	CMOS	捕捉 / 比较输入。
	P1A	ST	CMOS	PWM 的输出。
RC6/AN8	RC6	ST	CMOS	通用 I/O。
	AN8	AN	—	A/D 通道 8 的输入。
RC7/AN9	RC7	ST	CMOS	通用 I/O。
	AN9	AN	—	A/D 通道 9 的输入。
VSS	VSS	电源	—	参考地。
VDD	VDD	电源	—	正电源。

图注： AN = 模拟输入或输出　CMOS = CMOS 兼容输入或输出
TTL = TTL 兼容输入　ST = CMOS 电平的施密特触发器输入
HV = 高压　XTAL = 晶振

表 1-4： 引脚说明—— PIC16F687/PIC16F689

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN0	AN	—	A/D 通道 0 的输入。
	C1IN+	AN	—	比较器 C1 的同相输入。
	ICSPDAT	TTL	CMOS	ICSP™ 数据 I/O。
	ULPWU	AN	—	超低功耗唤醒输入。
RA1/AN1/C12IN0-/VREF/ICSPCLK	RA1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN1	AN	—	A/D 通道 1 的输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的反相输入。
	VREF	AN	—	A/D 的外部参考电压。
	ICSPCLK	ST	—	ICSP™ 时钟。
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN2	AN	—	A/D 通道 2 的输入。
	T0CKI	ST	—	Timer0 的时钟输入。
	INT	ST	—	外部中断。
	C1OUT	—	CMOS	比较器 C1 的输出。
RA3/$\overline{MCLR}$/VPP	RA3	TTL	—	通用输入。可单独控制电平变化中断。
	$\overline{MCLR}$	ST	—	带内部上拉的主复位。
	VPP	HV	—	编程电压。
RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	RA4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN3	AN	—	A/D 通道 3 的输入。
	$\overline{T1G}$	ST	—	Timer1 门控输入。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	FOSC/4 输出。
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T1CKI	ST	—	Timer1 的时钟输入。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN10	AN	—	A/D 通道 10 的输入。
	SDI	ST	—	SPI 数据输入。
	SDA	ST	OD	I^2C™ 数据输入 / 输出。
RB5/AN11/RX/DT	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN11	AN	—	A/D 通道 11 的输入。
	RX	ST	—	EUSART 异步输入。
	DT	ST	CMOS	EUSART 同步数据。

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 = CMOS 电平的施密特触发器输入
HV = 高压　XTAL = 晶振

表 1-4： 引脚说明—— PIC16F687/PIC16F689 （续）

名称	功能	输入类型	输出类型	说明
RB6/SCK/SCL	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	SCK	ST	CMOS	SPI 时钟。
	SCL	ST	OD	I²C™ 时钟。
RB7/TX/CK	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	TX	—	CMOS	EUSART 异步输出。
	CK	ST	CMOS	EUSART 同步时钟。
RC0/AN4/C2IN+	RC0	ST	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4 的输入。
	C2IN+	AN	—	比较器 C2 的同相输入。
RC1/AN5/C12IN1-	RC1	ST	CMOS	通用 I/O。
	AN5	AN	—	A/D 通道 5 的输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的反相输入。
RC2/AN6/C12IN2-	RC2	ST	CMOS	通用 I/O。
	AN6	AN	—	A/D 通道 6 的输入。
	C12IN2-	AN	—	比较器 C1 或 C2 的反相输入。
RC3/AN7/C12IN3-	RC3	ST	CMOS	通用 I/O。
	AN7	AN	—	A/D 通道 7 的输入。
	C12IN3-	AN	—	比较器 C1 或 C2 的反相输入。
RC4/C2OUT	RC4	ST	CMOS	通用 I/O。
	C2OUT	—	CMOS	比较器 C2 的输出。
RC5	RC5	ST	CMOS	通用 I/O。
RC6/AN8/$\overline{SS}$	RC6	ST	CMOS	通用 I/O。
	AN8	AN	—	A/D 通道 8 的输入。
	$\overline{SS}$	ST	—	从选择输入。
RC7/AN9/SDO	RC7	ST	CMOS	通用 I/O。
	AN9	AN	—	A/D 通道 9 的输入。
	SDO	—	CMOS	SPI 数据输出。
VSS	VSS	电源	—	参考地。
VDD	VDD	电源	—	正电源。

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 = CMOS 电平的施密特触发器输入
HV = 高压 XTAL = 晶振

表 1-5：　　　　引脚说明——PIC16F690

名称	功能	输入类型	输出类型	说明
RA0/AN0/C1IN+/ICSPDAT/ULPWU	RA0	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN0	AN	—	A/D 通道 0 的输入。
	C1IN+	AN	—	比较器 C1 的同相输入。
	ICSPDAT	TTL	CMOS	ICSP™ 数据 I/O。
	ULPWU	AN	—	超低功耗唤醒输入。
RA1/AN1/C12IN0-/V_{REF}/ICSPCLK	RA1	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN1	AN	—	A/D 通道 1 的输入。
	C12IN0-	AN	—	比较器 C1 或 C2 的反相输入。
	V_{REF}	AN	—	A/D 的外部参考电压。
	ICSPCLK	ST	—	ICSP™ 时钟。
RA2/AN2/T0CKI/INT/C1OUT	RA2	ST	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN2	AN	—	A/D 通道 2 的输入。
	T0CKI	ST	—	Timer0 的时钟输入。
	INT	ST	—	外部中断。
	C1OUT	—	CMOS	比较器 C1 的输出。
RA3/$\overline{MCLR}$/V_{PP}	RA3	TTL	—	通用输入。可单独控制电平变化中断。
	$\overline{MCLR}$	ST	—	带内部上拉的主复位。
	V_{PP}	HV	—	编程电压。
RA4/AN3/$\overline{T1G}$/OSC2/CLKOUT	RA4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN3	AN	—	A/D 通道 3 的输入。
	$\overline{T1G}$	ST	—	Timer1 门控输入。
	OSC2	—	XTAL	晶振 / 谐振器。
	CLKOUT	—	CMOS	F_{OSC}/4 输出。
RA5/T1CKI/OSC1/CLKIN	RA5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	T1CKI	ST	—	Timer1 的时钟输入。
	OSC1	XTAL	—	晶振 / 谐振器。
	CLKIN	ST	—	外部时钟输入 /RC 振荡器连接。
RB4/AN10/SDI/SDA	RB4	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN10	AN	—	A/D 通道 10 的输入。
	SDI	ST	—	SPI 数据输入。
	SDA	ST	OD	I^{2}C™ 数据输入 / 输出。
RB5/AN11/RX/DT	RB5	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	AN11	AN	—	A/D 通道 11 的输入。
	RX	ST	—	EUSART 异步输入。
	DT	ST	CMOS	EUSART 同步数据。

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 = CMOS 电平的施密特触发器输入
HV = 高压　　XTAL = 晶振

表 1-5： 引脚说明——PIC16F690 （续）

名称	功能	输入类型	输出类型	说明
RB6/SCK/SCL	RB6	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	SCK	ST	CMOS	SPI 时钟。
	SCL	ST	OD	I^{2}C™ 时钟。
RB7/TX/CK	RB7	TTL	CMOS	通用 I/O。可单独控制电平变化中断。可单独使能上拉。
	TX	—	CMOS	EUSART 异步输出。
	CK	ST	CMOS	EUSART 同步时钟。
RC0/AN4/C2IN+	RC0	ST	CMOS	通用 I/O。
	AN4	AN	—	A/D 通道 4 的输入。
	C2IN+	AN	—	比较器 C2 的同相输入。
RC1/AN5/C12IN1-	RC1	ST	CMOS	通用 I/O。
	AN5	AN	—	A/D 通道 5 的输入。
	C12IN1-	AN	—	比较器 C1 或 C2 的反相输入。
RC2/AN6/C12IN2-/P1D	RC2	ST	CMOS	通用 I/O。
	AN6	AN	—	A/D 通道 6 的输入。
	C12IN2-	AN	—	比较器 C1 或 C2 的反相输入。
	P1D	—	CMOS	PWM 的输出。
RC3/AN7/C12IN3-/P1C	RC3	ST	CMOS	通用 I/O。
	AN7	AN	—	A/D 通道 7 的输入。
	C12IN3-	AN	—	比较器 C1 或 C2 的反相输入。
	P1C	—	CMOS	PWM 的输出。
RC4/C2OUT/P1B	RC4	ST	CMOS	通用 I/O。
	C2OUT	—	CMOS	比较器 C2 的输出。
	P1B	—	CMOS	PWM 的输出。
RC5/CCP1/P1A	RC5	ST	CMOS	通用 I/O。
	CCP1	ST	CMOS	捕捉 / 比较输入。
	P1A	ST	CMOS	PWM 的输出。
RC6/AN8/$\overline{SS}$	RC6	ST	CMOS	通用 I/O。
	AN8	AN	—	A/D 通道 8 的输入。
	$\overline{SS}$	ST	—	从选择输入。
RC7/AN9/SDO	RC7	ST	CMOS	通用 I/O。
	AN9	AN	—	A/D 通道 9 的输入。
	SDO	—	CMOS	SPI 数据输出。
VSS	VSS	电源	—	参考地。
VDD	VDD	电源	—	正电源。

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 = CMOS 电平的施密特触发器输入
HV = 高压　XTAL = 晶振

2.0 存储器构成

2.1 程序存储器构成

PIC16F631/677/685/687/689/690器件具有一个13位程序计数器，能够寻址 8K x 14 的程序存储空间。只有PIC16F631 的前 1K x 14 存储单元（0000h-03FFh）、PIC16F677/PIC16F687 的前 2K x 14 存储单元（0000h-07FFh）和PIC16F685/PIC16F689/PIC16F690的前 4K x 14 存储单元（0000h-0FFFh）是物理实现的。访问超出上述边界的存储单元将回到原来空间。复位向量位于 0000h，而中断向量位于 0004h（见图 2-1 至 2-3）。

图 2-1： PIC16F631 的程序存储器映射和堆栈

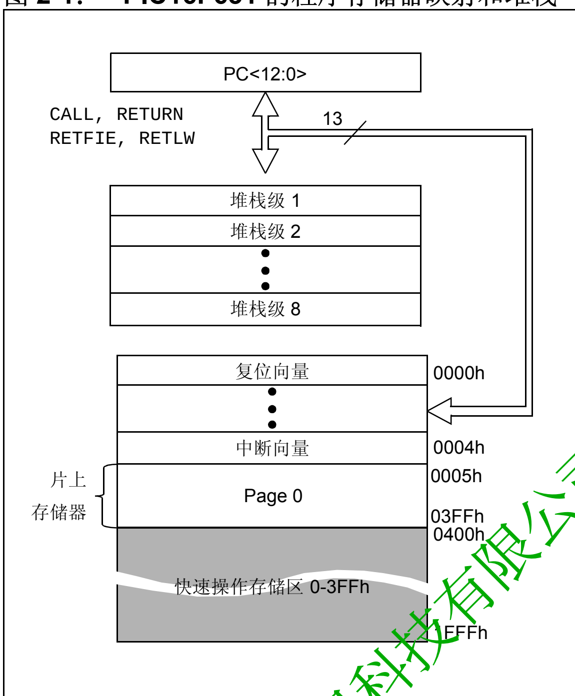


图 2-2： PIC16F685/689/690 的程序存储器映射和堆栈

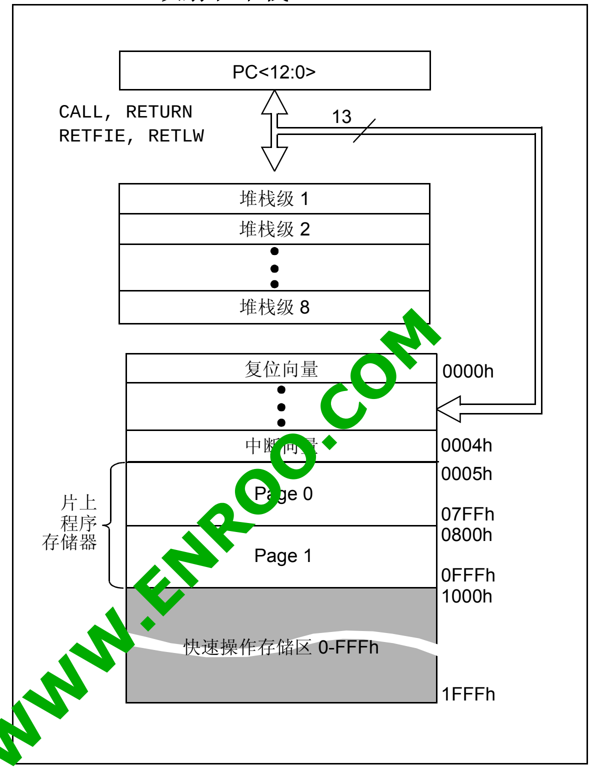
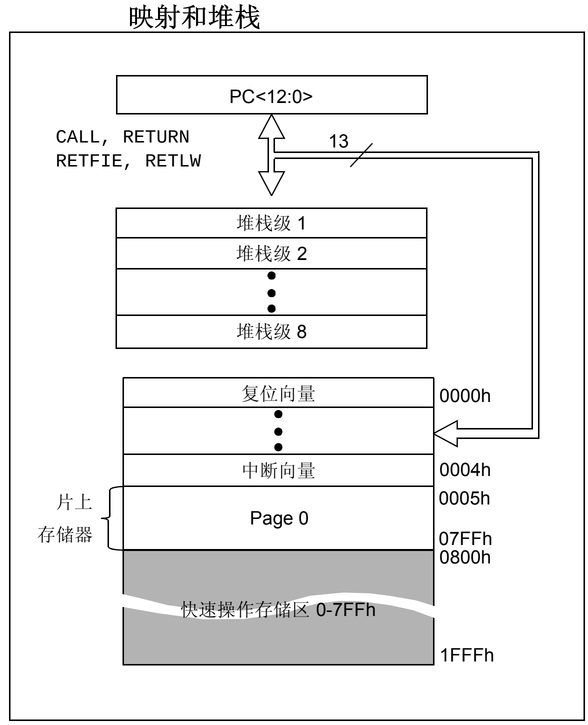


图 2-3： PIC16F677/PIC16F687 的程序存储器映射和堆栈



2.2 数据存储器构成

数据存储器（见图2-6至2-8）分为四个存储区（Bank），这四个存储区中包含通用寄存器（General Purpose Register，GPR）和特殊功能寄存器（Special Function Register，SFR）。特殊功能寄存器位于每个存储区的前32 个单元中。以静态 RAM 形式实现的通用寄存器位于每个存储区的后 96 个单元中。Bank 1 中的寄存器单元F0h-FFh、 Bank 2 中的 170h-17Fh 和 Bank 3 中的1F0h-1FFh 指向 Bank 0 中的地址 70h-7Fh。每个存储区中通用寄存器（GPR）的数量取决于具体器件。详细信息请参见图 2-4 至 2-8。所有其他 RAM 均未使用，读取时返回 0。状态寄存器的 RP<1:0> 是存储区选择位。

RP1	RP0		
0	0	→	选择 Bank 0
0	1	→	选择 Bank 1
1	0	→	选择 Bank 2
1	1	→	选择 Bank 3

2.2.1 通用寄存器文件

PIC16F687 的寄存器文件组织为 128 x 8，而PIC16F685/PIC16F689/PIC16F690 的寄存器文件则组织为 256 x 8。通过文件选择寄存器（File Select Register， FSR），可以直接或间接地访问每个寄存器（见第 **2.4** 节 **“间接寻址、 INDF 和 FSR 寄存器”**）。

2.2.2 特殊功能寄存器

特殊功能寄存器为CPU和外设模块用来对器件进行所需操作控制的寄存器（见表 2-1 至 2-4）。这些寄存器皆为静态 RAM。

特殊功能寄存器可分为两类，即：内核和外设。本节将介绍与“内核”相关的特殊功能寄存器。与外设模块相关的特殊功能寄存器将在相应的外设模块功能章节中介绍。

图 2-4： PIC16F631 特殊功能寄存器

	文件地址		文件地址		文件地址		文件地址
间接寻址 (1)	00h	间接寻址 (1)	80h	间接寻址 (1)	100h	间接寻址 (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	PORTA	105h	TRISA	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTC	107h	TRISC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh		10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
	11h		91h		111h		191h
	12h		92h		112h		192h
	13h		93h		113h		193h
	14h		94h		114h		194h
	15h	WPUA	95h	WPUB	115h		195h
	16h	IOCA	96h	IOCB	116h		196h
	17h	WDTCON	97h		117h		197h
	18h		98h	VRCON	118h		198h
	19h		99h	CM1CON0	119h		199h
	1Ah		9Ah	CM2CON0	11Ah		19Ah
	1Bh		9Bh	CM2CON1	11Bh		19Bh
	1Ch		9Ch		11Ch		19Ch
	1Dh		9Dh		11Dh		19Dh
	1Eh		9Eh	ANSEL	11Eh	SRCON	19Eh
	1Fh		9Fh		11Fh		19Fh
	20h		A0h		120h		1A0h
	3Fh						
通用寄存器 64 字节	40h						
	6Fh		EFh		16Fh		1EFh
	70h	访问 70h-7Fh	F0h	访问 70h-7Fh	170h	访问 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

▢ 未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

图 2-5： PIC16F677 特殊功能寄存器

Bank 0	文件地址	Bank 1	文件地址	Bank 2	文件地址	Bank 3	文件地址
间接寻址 (1)	00h	间接寻址 (1)	80h	间接寻址 (1)	100h	间接寻址 (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	PORTA	105h	TRISA	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTC	107h	TRISC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh		10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh		10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
	11h		91h		111h		191h
	12h		92h		112h		192h
SSPBUF	13h	SSPADD(2)	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
	15h	WPUA	95h	WPUB	115h		195h
	16h	IOCA	96h	IOCB	116h		196h
	17h	WDTCON	97h		117h		197h
	18h		98h	VRCON	118h		198h
	19h		99h	CM1CON0	119h		199h
	1Ah		9Ah	CM2CON0	11Ah		19Ah
	1Bh		9Bh	CM2CON1	11Bh		19Bh
	1Ch		9Ch		11Ch		19Ch
	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh	ANSEL	11Eh	SRCON	19Eh
ADCON0	1Fh	ADCON1	9Fh	ANSELH	11Fh		19Fh
通用寄存器 96 字节	20h	通用寄存器 32 字节	A0h		120h		1A0h
			BFh				
			C0h				
			EFh		16Fh		1EFh
		访问 70h-7Fh	F0h	访问 70h-7Fh	170h	访问 70h-7Fh	1F0h
	7Fh		FFh		17Fh		1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

注

■ 未实现的数据存储单元，读为 0。

1： 不是实际存在的寄存器。

2： 在某些情况下地址 93h 也可以访问 SSP 屏蔽 （SSPMSK）寄存器。更多详细信息，请参见寄存器 13-2 和 13-3。

图 2-6： PIC16F685 特殊功能寄存器

Bank 0	文件地址	Bank 1	文件地址	Bank 2	文件地址	Bank 3	文件地址
间接寻址 (1)	00h	间接寻址 (1)	80h	间接寻址 (1)	100h	间接寻址 (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	PORTA	105h	TRISA	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTC	107h	TRISC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h		91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
	13h		93h		113h		193h
	14h		94h		114h		194h
CCPR1L	15h	WPUA	95h	WPUB	115h		195h
CCPR1H	16h	IOCA	96h	IOCB	116h		196h
CCP1CON	17h	WDTCON	97h		117h		197h
	18h		98h	VRCON	118h		198h
	19h		99h	CM1CON0	119h		199h
	1Ah		9Ah	CM2CON0	11Ah		19Ah
	1Bh		9Bh	CM2CON1	11Bh		19Bh
PWM1CON	1Ch		9Ch		11Ch		19Ch
ECCPAS	1Dh		9Dh		11Dh	PSTRCON	19Dh
ADRESH	1Eh	ADRESL	9Eh	ANSEL	11Eh	SRCON	19Eh
ADCON0	1Fh	ADCON1	9Fh	ANSELH	11Fh		19Fh
通用寄存器 96 字节	20h–7Fh	通用寄存器 80 字节	A0h–EFh	通用寄存器 80 字节	120h–16Fh		1A0h–1EFh
		访问 70h-7Fh	F0h–FFh	访问 70h-7Fh	170h–17Fh	访问 70h-7Fh	1F0h–1FFh

■ 未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

图 2-7： PIC16F687/PIC16F689 特殊功能寄存器

Bank 0	文件地址	Bank 1	文件地址	Bank 2	文件地址	Bank 3	文件地址
间接寻址 (1)	00h	间接寻址 (1)	80h	间接寻址 (1)	100h	间接寻址 (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	PORTA	105h	TRISA	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTC	107h	TRISC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH(3)	10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH(3)	10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
	11h		91h		111h		191h
	12h		92h		112h		192h
SSPBUF	13h	SSPADD(2)	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
	15h	WPUA	95h	WPUB	115h		195h
	16h	IOCA	96h	IOCB	116h		196h
	17h	WDTCON	97h		117h		197h
RCSTA	18h	TXSTA	98h	VRCON	118h		198h
TXREG	19h	SPBRG	99h	CM1CON0	119h		199h
RCREG	1Ah	SPBRGH	9Ah	CM2CON0	11Ah		19Ah
	1Bh	BAUDCTL	9Bh	CM2CON1	11Bh		19Bh
	1Ch		9Ch		11Ch		19Ch
	1Dh		9Dh		11Dh		19Dh
ADRESH	1Eh	ADRESL	9Eh	ANSEL	11Eh	SRCON	19Eh
ADCON0	1Fh	ADCON1	9Fh	ANSELH	11Fh		19Fh
通用寄存器 96 字节	20h–7Fh	通用寄存器 32 字节	A0h–BFh	通用寄存器 80 字节（仅限 PIC16F689）	120h		1A0h
		48 字节（仅限 PIC16F689）	C0h–EFh				
		访问 70h-7Fh	F0h–FFh	访问 70h-7Fh	170h–17Fh	访问 70h-7Fh	1F0h–1FFh

■ 未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

2： 在某些情况下地址 93h 也可以访问 SSP 屏蔽 （SSPMSK）寄存器。更多详细信息，请参见寄存器 13-2 和 13-3。

3： 仅限 PIC16F689。

图 2-8： PIC16F690 特殊功能寄存器

	文件地址		文件地址		文件地址		文件地址
间接寻址 (1)	00h	间接寻址 (1)	80h	间接寻址 (1)	100h	间接寻址 (1)	180h
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h
PCL	02h	PCL	82h	PCL	102h	PCL	182h
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h
FSR	04h	FSR	84h	FSR	104h	FSR	184h
PORTA	05h	TRISA	85h	PORTA	105h	TRISA	185h
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h
PORTC	07h	TRISC	87h	PORTC	107h	TRISC	187h
	08h		88h		108h		188h
	09h		89h		109h		189h
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh
PIR1	0Ch	PIE1	8Ch	EEDAT	10Ch	EECON1	18Ch
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2(1)	18Dh
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh		18Eh
TMR1H	0Fh	OSCCON	8Fh	EEADRH	10Fh		18Fh
T1CON	10h	OSCTUNE	90h		110h		190h
TMR2	11h		91h		111h		191h
T2CON	12h	PR2	92h		112h		192h
SSPBUF	13h	SSPADD(2)	93h		113h		193h
SSPCON	14h	SSPSTAT	94h		114h		194h
CCPR1L	15h	WPUA	95h	WPUB	115h		195h
CCPR1H	16h	IOCA	96h	IOCB	116h		196h
CCP1CON	17h	WDTCON	97h		117h		197h
RCSTA	18h	TXSTA	98h	VRCON	118h		198h
TXREG	19h	SPBRG	99h	CM1CON0	119h		199h
RCREG	1Ah	SPBRGH	9Ah	CM2CON0	11Ah		19Ah
	1Bh	BAUDCTL	9Bh	CM2CON1	11Bh		19Bh
PWM1CON	1Ch		9Ch		11Ch		19Ch
ECCPAS	1Dh		9Dh		11Dh	PSTRCON	19Dh
ADRESH	1Eh	ADRESL	9Eh	ANSEL	11Eh	SRCON	19Eh
ADCON0	1Fh	ADCON1	9Fh	ANSELH	11Fh		19Fh
通用寄存器 96 字节	20h–7Fh	通用寄存器 80 字节	A0h–EFh	通用寄存器 80 字节	120h–16Fh		1A0h–1EFh
		访问 70h-7Fh	F0h–FFh	访问 70h-7Fh	170h–17Fh	访问 70h-7Fh	1F0h–1FFh
Bank 0		Bank 1		Bank 2		Bank 3	

■ 未实现的数据存储单元，读为 0。

注 1： 不是实际存在的寄存器。

2： 在某些情况下地址 93h 也可以访问 SSP 屏蔽 （SSPMSK）寄存器。更多详细信息，请参见寄存器 13-2 和 13-3。

表 2-1： PIC16F631/677/685/687/689/690 特殊功能寄存器汇总， BANK 0

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所在页
Bank 0											
00h	INDF	通过用 FSR 的内容寻址这个存储单元来寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	44,205
01h	TMR0	Timer0 模块寄存器								xxxx xxxx	81,205
02h	PCL	程序计数器 （PC）的低字节								0000 0000	44,205
03h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	36,205
04h	FSR	间接数据存储器地址指针								xxxx xxxx	44,205
05h	PORTA(7)	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	59,205
06h	PORTB(7)	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	69,205
07h	PORTC(7)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	76,205
08h	—	未实现								—	—
09h	—	未实现								—	—
0Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲器					---0 0000	44,205
0Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF(1)	0000 000x	38,205
0Ch	PIR1	—	ADIF(4)	RCIF(2)	TXIF(2)	SSPIF(5)	CCP1IF(3)	TMR2IF(3)	TMR1IF	-000 0000	41,205
0Dh	PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000 ----	42,205
0Eh	TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	86,205
0Fh	TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	86,205
10h	T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	88,205
11h	TMR2(3)	Timer2 模块寄存器								0000 0000	91,205
12h	T2CON(3)	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	92,205
13h	SSPBUF(5)	同步串行口接收缓冲器 / 发送寄存器								xxxx xxxx	182,205
14h	SSPCON(5, 6)	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	181,205
15h	CCPR1L(3)	捕捉 / 比较 /PWM 寄存器 1 （LSB）								xxxx xxxx	128,205
16h	CCPR1H(3)	捕捉 / 比较 /PWM 寄存器 1 （MSB）								xxxx xxxx	128,205
17h	CCP1CON(3)	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	127,205
18h	RCSTA(2)	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	161,205
19h	TXREG(2)	EUSART 发送数据寄存器								0000 0000	153
1Ah	RCREG(2)	EUSART 接收数据寄存器								0000 0000	156
1Bh	—	未实现								—	—
1Ch	PWM1CON(3)	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	145,205
1Dh	ECCPAS(3)	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	142,205
1Eh	ADRESH(4)	A/D 结果寄存器的高字节								xxxx xxxx	115,205
1Fh	ADCON0(4)	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	113,205

图注： – = 未实现单元 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件，阴影 = 未实现

注 1： $\overline{\text{MCLR}}$ 和 WDT 复位不影响数据锁存器原来的值。 RAIF 位在复位时将清零，但如果存在不匹配则会再次置 1。
2： 仅限 PIC16F687/PIC16F689/PIC16F690。
3： 仅限 PIC16F685/PIC16F690。
4： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。
5： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。
6： 当 SSPCON 寄存器位 SSPM<3:0> = 1001 时，任何对 SSPADD SFR 地址的读或写都必须通过 SSPMSK 寄存器来访问。更多详细信息，请参见寄存器 13-2 和 13-3。
7： 由 ANSEL 和 ANSELH 寄存器控制的具有模拟功能的端口引脚将在复位后立即读为 0，即使数据锁存器未定义 （POR）或不变 （其他复位）时也是如此。

表 2-2： PIC16F631/677/685/687/689/690 特殊功能寄存器汇总， BANK 1

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所在页
Bank 1											
80h	INDF	通过用 FSR 的内容寻址这个存储单元来寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	44,205
81h	OPTION_REG	$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	37,205
82h	PCL	程序计数器 （PC）的低字节								0000 0000	44,205
83h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	36,205
84h	FSR	间接数据存储器地址指针								xxxx xxxx	44,205
85h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	59,205
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	70,205
87h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	76,205
88h	—	未实现								—	—
89h	—	未实现								—	—
8Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲区					---0 0000	44,205
8Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF(1)	0000 000x	38,205
8Ch	PIE1	—	ADIE(4)	RCIE(2)	TXIE(2)	SSPIE(5)	CCP1IE(3)	TMR2IE(3)	TMR1IE	-000 0000	39,206
8Dh	PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	40,206
8Eh	PCON	—	—	ULPWUE	SBOREN	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$	--01 --qq	43,206
8Fh	OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 q000	48,206
90h	OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	52,206
91h	—	未实现								—	—
92h	PR2(3)	Timer2 周期寄存器								1111 1111	91,206
93h	SSPADD(5, 7)	同步串行口 （I²C 模式）地址寄存器								0000 0000	188,206
93h	SSPMSK(5, 7)	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	191,206
94h	SSPSTAT(5)	SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF	0000 0000	180,206
95h	WPUA(6)	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	62,206
96h	IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	62,206
97h	WDTCON	—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN	---0 1000	213,206
98h	TXSTA(2)	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	160,206
99h	SPBRG(2)	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	163,206
9Ah	SPBRGH(2)	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	163,206
9Bh	BAUDCTL(2)	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	162,206
9Ch	—	未实现								—	—
9Dh	—	未实现								—	—
9Eh	ADRESL(4)	A/D 结果寄存器的低字节								xxxx xxxx	115,206
9Fh	ADCON1(4)	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	114,206

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件，阴影 = 未实现

注 1： $\overline{\text{MCLR}}$ 和 WDT 复位不影响数据锁存器原来的值。 RAIF 位在复位时将清零，但如果存在不匹配则会再次置 1。
2： 仅限 PIC16F687/PIC16F689/PIC16F690。
3： 仅限 PIC16F685/PIC16F690。
4： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。
5： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。
6： 当引脚在配置字中配置为 $\overline{\text{MCLR}}$ 时， RA3 上拉使能。
7： 仅当 SSPCON 寄存器位 SSPM<3:0> = 1001 时才可访问。

表 2-3： PIC16F631/677/685/687/689/690 特殊功能寄存器汇总， BANK 2

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所在页
Bank 2											
100h	INDF	用 FSR 的内容寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	44,205
101h	TMR0	Timer0 模块寄存器								xxxx xxxx	81,205
102h	PCL	程序计数器 （PC）的低字节								0000 0000	44,205
103h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	36,205
104h	FSR	间接数据存储器地址指针								xxxx xxxx	44,205
105h	PORTA(4)	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	59,205
106h	PORTB(4)	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	69,205
107h	PORTC(4)	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	76,205
108h	—	未实现								—	—
109h	—	未实现								—	—
10Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲区					---0 0000	44,205
10Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF(1)	0000 000x	38,205
10Ch	EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	120,206
10Dh	EEADR	EEADR7(3)	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	120,206
10Eh	EEDATH(2)	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	120,206
10Fh	EEADRH(2)	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	120,206
110h	—	未实现								—	—
111h	—	未实现								—	—
112h	—	未实现								—	—
113h	—	未实现								—	—
114h	—	未实现								—	—
115h	WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	70,206
116h	IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	70,206
117h	—	未实现								—	—
118h	VRCON	C1VREN	C2VREN	VRR	VP6EN	VR3	VR2	VR1	VR0	0000 0000	106,206
119h	CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	98,206
11Ah	CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	99,206
11Bh	CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	00-- --10	101,206
11Ch	—	未实现								—	—
11Dh	—	未实现								—	—
11Eh	ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3(3)	ANS2(3)	ANS1	ANS0	1111 1111	61,206
11Fh	ANSELH(3)	—	—	—	—	ANS11	ANS10	ANS9	ANS8	---- 1111	115,206

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件， 阴影 = 未实现

注 1： $\overline{\text{MCLR}}$ 和 WDT 复位不影响数据锁存器原来的值。 RAIF 位在复位时将清零，但如果存在不匹配则会再次置 1。

2： 仅限 PIC16F685/PIC16F689/PIC16F690。

3： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

4： 由 ANSEL 和 ANSELH 寄存器控制的具有模拟功能的端口引脚将在复位后立即读为 0，即使数据锁存器未定义 （POR）或不变 （其他复位）时也是如此。

表 **2-4：** **PIC16F631/677/685/687/689/690 特殊功能寄存器汇总， BANK 3**

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所在页
Bank 3											
180h	INDF	通过用 FSR 的内容寻址这个存储单元来寻址数据存储器 （不是实际存在的寄存器）								xxxx xxxx	44,205
181h	OPTION_REG	$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	37,205
182h	PCL	程序计数器 （PC）的低字节								0000 0000	44,205
183h	STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C	0001 1xxx	36,205
184h	FSR	间接数据存储器地址指针								xxxx xxxx	44,205
185h	TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	59,205
186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	70,205
187h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	76,206
188h	—	未实现								—	—
189h	—	未实现								—	—
18Ah	PCLATH	—	—	—	程序计数器高 5 位的写缓冲区					---0 0000	44,205
18Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF[(1)]	0000 000x	38,205
18Ch	EECON1	EEPGD[(2)]	—	—	—	WRERR	WREN	WR	RD	x--- x000	121,206
18Dh	EECON2	EEPROM 控制寄存器 2 （不是实际存在的寄存器）								---- ----	119,206
18Eh	—	未实现								—	—
18Fh	—	未实现								—	—
190h	—	未实现								—	—
191h	—	未实现								—	—
192h	—	未实现								—	—
193h	—	未实现								—	—
194h	—	未实现								—	—
195h	—	未实现								—	—
196h	—	未实现								—	—
197h	—	未实现								—	—
198h	—	未实现								—	—
199h	—	未实现								—	—
19Ah	—	未实现								—	—
19Bh	—	未实现								—	—
19Ch	—	未实现								—	—
19Dh	PSTRCON[(2)]	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	146,206
19Eh	SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	—	0000 00--	103,206
19Fh	—	未实现								—	—

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知， q = 值取决于具体条件，阴影 = 未实现

注 1： $\overline{\text{MCLR}}$ 和 WDT 复位不影响数据锁存器原来的值。 RAIF 位在复位时将清零，但如果存在不匹配则会再次置 1。

2： 仅限 PIC16F685/PIC16F690。

2.2.2.1 状态寄存器

状态 （STATUS）寄存器如寄存器 2-1 所示，包括：

- ALU 的算术运算状态
- 复位状态
- 数据存储 （GPR 和 SFR）的存储区选择位

状态寄存器与任何其他寄存器一样，可作为任何指令的目标寄存器。如果一条影响 Z、 DC 或 C 位的指令以状态寄存器作为目标寄存器，那么对这三个位的写操作将被禁止。这些位根据器件逻辑被置 1 或清零。而且，$\overline{\text{TO}}$ 和 $\overline{\text{PD}}$ 位均为不可写位。因此，当执行一条将状态寄存器作为目标寄存器的指令时，运行结果可能会与预想的不同。

例如，指令 CLRF STATUS 将会清除状态寄存器中的高三位，并将 Z 位置 1。这将使状态寄存器中的值成为 000u u1uu （其中 u = 不变）。

因此，建议仅使用 BCF、 BSF、 SWAPF 和 MOVWF 指令来改变状态寄存器的值，因为这些指令不会影响任何状态位。关于其他不影响任何状态位的指令，请参见**第 15.0 节 “指令集汇总”**。

> 注 1： 在减法运算中， C 和 DC 位分别作为$\overline{\text{借位}}$和半$\overline{\text{借位}}$位。请参见 SUBLW 和 SUBWF 指令示例。

寄存器 2-1： STATUS：状态寄存器

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC (1)	C (1)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **IRP：**寄存器存储区选择位 （用于间接寻址）
1 = Bank 2 和 3 （100h — 1FFh）
0 = Bank 0 和 1 （00h — FFh）

bit 6-5 **RP<1:0>：**寄存器存储区选择位 （用于直接寻址）
00 = Bank 0 （00h — 7Fh）
01 = Bank 1 （80h — FFh）
10 = Bank 2 （100h — 17Fh）
11 = Bank 3 （180h — 1FFh）

bit 4 **$\overline{\text{TO}}$：**超时位
1 = 在上电、执行 CLRWDT 或 SLEEP 指令后
0 = 产生了 WDT 超时

bit 3 **$\overline{\text{PD}}$：**掉电位
1 = 在上电或执行 CLRWDT 指令后
0 = 执行 SLEEP 指令

bit 2 **Z：**全零位
1 = 算术运算或逻辑运算的结果为零
0 = 算术运算或逻辑运算的结果不为零

bit 1 **DC：**半进位 / $\overline{\text{借位}}$位 （ADDWF、 ADDLW、 SUBLW 和 SUBWF 指令） (1)
1 = 结果的第 4 个低位发生了进位
0 = 结果的第 4 个低位未发生进位

bit 0 **C：**进位 / $\overline{\text{借位}}$位 (1) （ADDWF、 ADDLW、 SUBLW 和 SUBWF 指令） (1)
1 = 结果中最高位发生了进位
0 = 结果中最高位未发生进位

注 1： 对于$\overline{\text{借位}}$，极性是相反的。减法是通过加上第二个操作数的二进制补码来执行的。对于移位指令 （RRF 和 RLF），此位装入源寄存器的最高位或最低位。

2.2.2.2 选项寄存器

选项（OPTION）寄存器（如寄存器 2-2 所示）是可读写寄存器，有各种控制位，用来配置以下各项：

- Timer0/WDT 预分频器
- 外部 RA2/INT 中断
- Timer0
- PORTA 的弱上拉 / 下拉

> 注： 要使 Timer0 获得 1:1 的预分频比，可将选项寄存器的 PSA 位置为 1，以将预分频器分配给 WDT。请参见第 **6.3 节 “Timer1 预分频器”**。

寄存器 2-2： OPTION_REG：选项寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **$\overline{\text{RABPU}}$**：PORTA/PORTB 上拉使能位
1 = 禁止 PORTA/PORTB 上拉
0 = PORTA/PORTB 上拉由各端口锁存值使能

bit 6 **INTEDG**：中断边沿选择位
1 = RA2/INT 引脚的上升沿触发中断
0 = RA2/INT 引脚的下降沿触发中断

bit 5 **T0CS**：Timer0 时钟源选择位
1 = RA2/T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟（FOSC/4）

bit 4 **T0SE**：Timer0 时钟源边沿选择位
1 = 在 RA2/T0CKI 引脚信号从高至低跳变时，递增计数
0 = 在 RA2/T0CKI 引脚信号从低至高跳变时，递增计数

bit 3 **PSA**：预分频器分配位
1 = 预分频器分配给 WDT
0 = 预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>**：预分频比选择位

位值	Timer0 分频比	WDT 分频比
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

2.2.2.3 INTCON 寄存器

中断控制（INTCON）寄存器（如寄存器 2-3 所示）是可读写的寄存器，它包含 TMR0 寄存器溢出、PORTA 电平变化和外部 RA2/AN2/T0CKI/INT/C1OUT 引脚中断等各种使能控制位和标志位。

> **注：** 当中断条件发生时，无论相应中断允许位或全局中断允许位 GIE（在 INTCON 寄存器中）的状态如何，中断标志位都将被置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 2-3：　INTCON：中断控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
GIE	PEIE	T0IE	INTE	RABIE(1,3)	T0IF(2)	INTF	RABIF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **GIE：**全局中断允许位
1 = 允许所有未被屏蔽的中断
0 = 禁止所有中断

bit 6 **PEIE：**外设中断允许位
1 = 允许所有未被屏蔽的外设中断
0 = 禁止所有外设中断

bit 5 **T0IE：**Timer0 溢出中断允许位
1 = 允许 Timer0 中断
0 = 禁止 Timer0 中断

bit 4 **INTE：**RA2/INT 外部中断允许位
1 = 允许 RA2/INT 外部中断
0 = 禁止 RA2/INT 外部中断

bit 3 **RABIE：**PORTA/PORTB 电平变化中断允许位 (1,3)
1 = 允许 PORTA/PORTB 电平变化中断
0 = 禁止 PORTA/PORTB 电平变化中断

bit 2 **T0IF：**Timer0 溢出中断标志位 (2)
1 = TMR0 寄存器溢出（必须用软件清零）
0 = TMR0 寄存器未溢出

bit 1 **INTF：**RA2/INT 外部中断标志位
1 = RA2/INT 外部中断发生（必须用软件清零）
0 = RA2/INT 外部中断没有发生

bit 0 **RABIF：**PORTA/PORTB 电平变化中断标志位
1 = 至少有一个 PORTA 或 PORTB 通用 I/O 引脚状态发生变化时（必须用软件清零）
0 = PORTA 或 PORTB 通用 I/O 引脚状态均未发生变化

注 **1：** IOCA 或 IOCB 寄存器也必须被使能。
2： 当 Timer0 计数出现计满返回时，T0IF 位将被置 1。Timer0 计数值在复位时不变，而且应在清零 T0IF 位之前对其进行初始化。
3： 包含 ULPWU 中断。

2.2.2.4 PIE1 寄存器

PIE1 寄存器包含中断允许位，如寄存器 2-4 所示。

注：	必须将 INTCON 寄存器的 PEIE 位置 1，以允许外设中断。

寄存器 2-4： PIE1：外设中断允许寄存器 1

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIE(5)	RCIE(3)	TXIE(3)	SSPIE(4)	CCP1IE(2)	TMR2IE(1)	TMR1IE
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 0

bit 6 **ADIE**：A/D 转换器 （ADC）中断允许位 (5)
1 = 允许 ADC 中断
0 = 禁止 ADC 中断

bit 5 **RCIE**：EUSART 接收中断允许位 (3)
1 = 允许 EUSART 接收中断
0 = 禁止 EUSART 接收中断

bit 4 **TXIE**：EUSART 发送中断允许位 (5)
1 = 允许 EUSART 发送中断
0 = 禁止 EUSART 发送中断

bit 3 **SSPIE**：同步串行口 （SSP）中断允许位 (4)
1 = 允许 SSP 中断
0 = 禁止 SSP 中断

bit 2 **CCP1IE**：CCP1 中断允许位 (2)
1 = 允许 CCP1 中断
0 = 禁止 CCP1 中断

bit 1 **TMR2IE**：Timer2 与 PR2 匹配中断允许位 (1)
1 = 允许 Timer2 与 PR2 匹配中断
0 = 禁止 Timer2 与 PR2 匹配中断

bit 0 **TMR1IE**：Timer1 溢出中断允许位
1 = 允许 Timer1 溢出中断
0 = 禁止 Timer1 溢出中断

注 **1**： 仅限 PIC16F685/PIC16F690。
2： 仅限 PIC16F685/PIC16F689/PIC16F690。
3： 仅限 PIC16F687/PIC16F689/PIC16F690。
4： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。
5： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

2.2.2.5 PIE2 寄存器

PIE2 寄存器包含中断允许位，如寄存器 2-5 所示。

注：	必须将 INTCON 寄存器的 PEIE 位置 1，以允许外设中断。

寄存器 2-5： PIE2：外设中断允许寄存器 2

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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：**EE 写操作中断允许位
1 = 允许写操作中断
0 = 禁止写操作中断

bit 3-0 未实现：读为 0

2.2.2.6 PIR1 寄存器

PIR1 寄存器包含中断标志位，如寄存器 2-6 所示。

注：	当中断条件发生时，无论相应中断允许位或全局中断允许位 GIE （在 INTCON 寄存器中）的状态如何，中断标志位都将被置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 2-6： PIR1：外设中断请求寄存器 1

U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADIF(5)	RCIF(3)	TXIF(3)	SSPIF(4)	CCP1IF(2)	TMR2IF(1)	TMR1IF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7　未实现：读为 0

bit 6　**ADIF：**A/D 转换器中断标志位 (5)
1 = A/D 转换完成 （必须用软件清零）
0 = A/D 转换未完成或尚未开始

bit 5　**RCIF：**EUSART 接收中断标志位 (3)
1 = EUSART 接收缓冲器满 （通过读取 RCREG 清零）
0 = EUSART 接收缓冲器未满

bit 4　**TXIF：**EUSART 发送中断标志位 (3)
1 = EUSART 发送缓冲器为空 （通过写入 TXREG 清零）
0 = EUSART 发送缓冲器满

bit 3　**SSPIF：**同步串行口 （SSP）中断标志位 (4)
1 = 发送 / 接收完成 （必须用软件清零）
0 = 等待发送 / 接收

bit 2　**CCP1IF：**CCP1 中断标志位 (2)
捕捉模式：
　1 = 发生 TMR1 寄存器捕捉 （必须用软件清零）
　0 = 未发生 TMR1 寄存器捕捉
比较模式：
　1 = 发生 TMR1 寄存器比较匹配 （必须用软件清零）
　0 = 未发生 TMR1 寄存器比较匹配
PWM 模式：
在此模式下不使用

bit 1　**TMR2IF：**Timer2 与 PR2 匹配中断标志位 (1)
1 = 发生 Timer2 与 PR2 匹配 （必须用软件清零）
0 = 未发生 Timer2 与 PR2 匹配

bit 0　**TMR1IF：**Timer1 溢出中断标志位
1 = TMR1 寄存器溢出 （必须用软件清零）
0 = TMR1 寄存器未溢出

注 **1：** 仅限 PIC16F685/PIC16F690。
2： 仅限 PIC16F685/PIC16F689/PIC16F690。
3： 仅限 PIC16F687/PIC16F689/PIC16F690。
4： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。
5： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

2.2.2.7 PIR2 寄存器

PIR2 寄存器包含中断标志位，如寄存器 2-7 所示。

注：	当中断条件发生时，无论相应中断允许位或全局中断允许位 GIE （在 INTCON 寄存器中）的状态如何，中断标志位都将被置 1。用户软件应在允许一个中断前，先将相应的中断标志位清零。

寄存器 2-7： PIR2：外设中断请求寄存器 2

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **OSFIF：**振荡器故障中断标志位
1 = 系统振荡器故障，时钟输入更改为 INTOSC （必须用软件清零）
0 = 系统时钟正常工作

bit 6 **C2IF：**比较器 C2 中断标志位
1 = 比较器输出 （C2OUT 位）发生变化 （必须用软件清零）
0 = 比较器输出 （C2OUT 位）不变

bit 5 **C1IF：**比较器 C1 中断标志位
1 = 比较器输出 （C1OUT 位）发生变化 （必须用软件清零）
0 = 比较器输出 （C1OUT 位）不变

bit 4 **EEIF：**EE 写操作中断标志位
1 = 写操作完成 （必须用软件清零）
0 = 写操作未完成或尚未开始

bit 3-0 未实现：读为 0

2.2.2.8 PCON 寄存器

电源控制（PCON）寄存器（见寄存器 2-8）包含区分以下各种复位的标志位：

- 上电复位（$\overline{\text{POR}}$）
- 欠压复位（$\overline{\text{BOR}}$）
- 看门狗定时器复位（WDT）
- 外部 $\overline{\text{MCLR}}$ 复位

PCON寄存器还控制超低功耗唤醒和$\overline{\text{BOR}}$的软件使能。

寄存器 2-8： PCON：电源控制寄存器

U-0	U-0	R/W-0	R/W-1	U-0	U-0	R/W-0	R/W-x
—	—	ULPWUE	SBOREN(1)	—	—	$\overline{\text{POR}}$	$\overline{\text{BOR}}$
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：**读为 0

bit 5 **ULPWUE：**超低功耗唤醒使能位
1 = 使能超低功耗唤醒
0 = 禁止超低功耗唤醒

bit 4 **SBOREN：**软件 BOR 使能位 (1)
1 = 使能 BOR
0 = 禁止 BOR

bit 3-2 **未实现：**读为 0

bit 1 **$\overline{\text{POR}}$：**上电复位状态位
1 = 未发生上电复位
0 = 发生了上电复位（发生上电复位后必须用软件置 1）

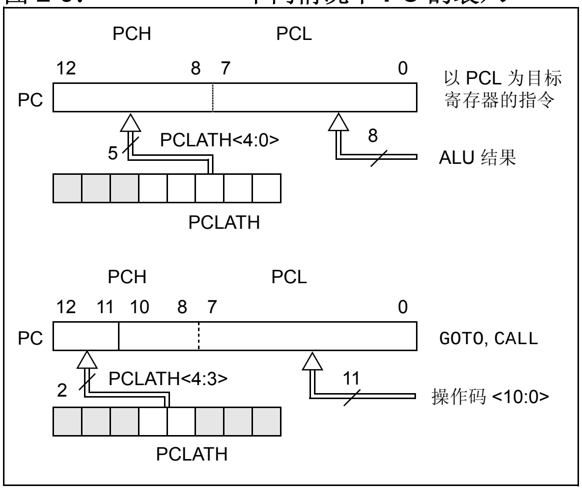
bit 0 **$\overline{\text{BOR}}$：**欠压复位状态位
1 = 未发生欠压复位
0 = 发生了欠压复位（发生欠压复位后必须用软件置 1）

注 1： 配置字寄存器中的 BOREN<1:0> = 01 时该位控制 $\overline{\text{BOR}}$。

2.3 PCL 和 PCLATH

程序计数器（PC）为 13 位宽。其低 8 位来自可读写的 PCL 寄存器，高 5 位（PC<12:8>）来自 PCLATH，不能直接读写。只要发生复位，PC 就将被清零。图 2-9 显示了装入 PC 值的两种情形。图 2-9 上方的例子说明在写 PCL（PCLATH<4:0> → PCH）时是如何装入 PC 的。图 2-9 下方的例子说明了在执行 CALL 或 GOTO 指令期间（PCLATH<4:3> → PCH），是如何装入 PC 的。

图 2-9： 不同情况下 **PC** 的装入



2.3.1 修改 PCL

在执行以 PCL 寄存器作为目标寄存器的任何指令的同时，也会使程序计数器的 PC<12:8> 位（PCH）被 PCLATH 寄存器的内容所代替。这使得可以通过将所需的高 5 位写入 PCLATH 寄存器来改变程序计数器的整个内容。当将低 8 位写入 PCL 寄存器时，程序计数器的所有 13 位都将变为 PCLATH 寄存器中和那些被写入 PCL 寄存器的值。

计算 GOTO 指令是通过向程序计数器加一个偏移量（ADDWF PCL）来实现的。当通过修改 PCL 寄存器跳转到查找表或程序转移表（计算 GOTO）时，应特别注意。假设 PCLATH 被设置为表起始地址，如果表长度大于 255 条指令，或如果在表中间存储器地址的低 8 位从 0xFF 计满返回到 0x00，那么 PCLATH 必须为在表起始和表中的目标存储单元之间发生计满返回的每个地址进行递增。

更多信息，请参见应用笔记 AN556，“*Implementing a Table Read*”（DS00556）。

2.3.2 堆栈

PIC16F631/677/685/687/689/690 器件具有 8 级 x 13 位宽的硬件堆栈（见图 2-2 和 2-3）。堆栈空间既不占用程序存储空间，也不占用数据存储空间，而且堆栈指针是不可读写的。当执行 CALL 指令或当中断导致程序跳转时，PC 值将被压入堆栈。而在执行 RETURN、RETLW 或 RETFIE 指令时，将从堆栈中弹出 PC 值。PCLATH 不受压栈或出栈操作的影响。

堆栈的工作原理犹如循环缓冲区。这意味着当压栈 8 次后，第 9 次压入堆栈的值将会覆盖第一次压栈时所保存的值，而第 10 次压入堆栈的值将覆盖第二次压栈时保存的值，依此类推。

> **注 1：** 不存在指明堆栈是否上溢或下溢的状态标志位。
>
> **2：** 不存在被称为 PUSH 或 POP 的指令 / 助记符。堆栈的压入或弹出是源于执行了 CALL、RETURN、RETLW 和 RETFIE 指令，或源于跳转到中断向量地址。

2.4 间接寻址，INDF 和 FSR 寄存器

INDF 寄存器不是实际存在的寄存器。对 INDF 寄存器进行寻址将产生间接寻址。

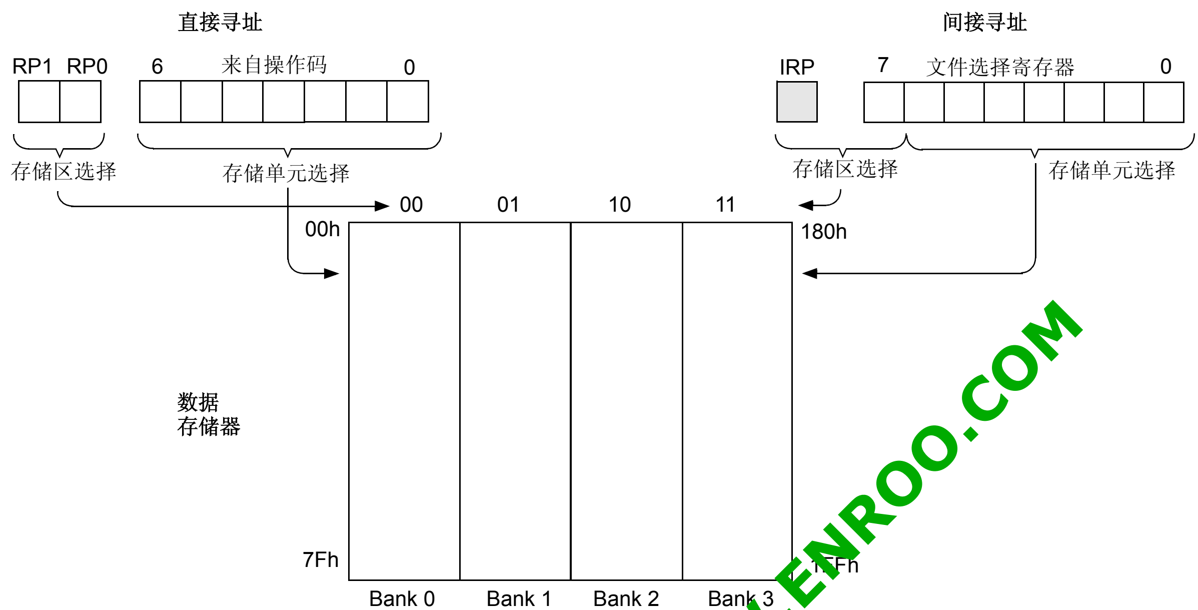
使用 INDF 寄存器可进行间接寻址。任何使用 INDF 寄存器的指令，实际上是对文件选择寄存器（FSR）所指向的数据进行存取。间接对 INDF 进行读操作将返回 00h。间接对 INDF 寄存器进行写操作将导致空操作（尽管可能会影响状态位）。通过将 8 位的 FSR 寄存器与状态寄存器的 IRP 位进行组合可得到一个有效的 9 位地址，如图 2-10 所示。

例 2-1 给出了一个使用间接寻址将 RAM 地址单元 20h-2Fh 清零的简单程序。

例 2-1： 间接寻址

```
         MOVLW   0x20     ;initialize pointer
         MOVWF   FSR      ;to RAM
NEXT     CLRF    INDF     ;clear INDF register
         INCF    FSR      ;inc pointer
         BTFSS   FSR,4    ;all done?
         GOTO    NEXT     ;no clear next
CONTINUE                  ;yes continue
```


图 2-10： PIC16F631/677/685/687/689/690 的直接 / 间接寻址



存储器映射的详细信息请参见图 2-6、 2-7 和 2-8。

3.0 振荡器模块（带故障保护时钟监视器）

3.1 概述

振荡器模块具有多种时钟源和选择特性，广泛使用于各种应用中，同时最大限度地发挥应用性能并降低功耗。图 3-1 为振荡器模块的框图。

时钟源可配置为来自外部振荡器、石英晶体谐振器、陶瓷谐振器以及阻容（Resistor-Capacitor，RC）电路。此外，系统时钟源可配置为两个内部振荡器之一，并通过软件来选择速度。其他时钟特性包括：

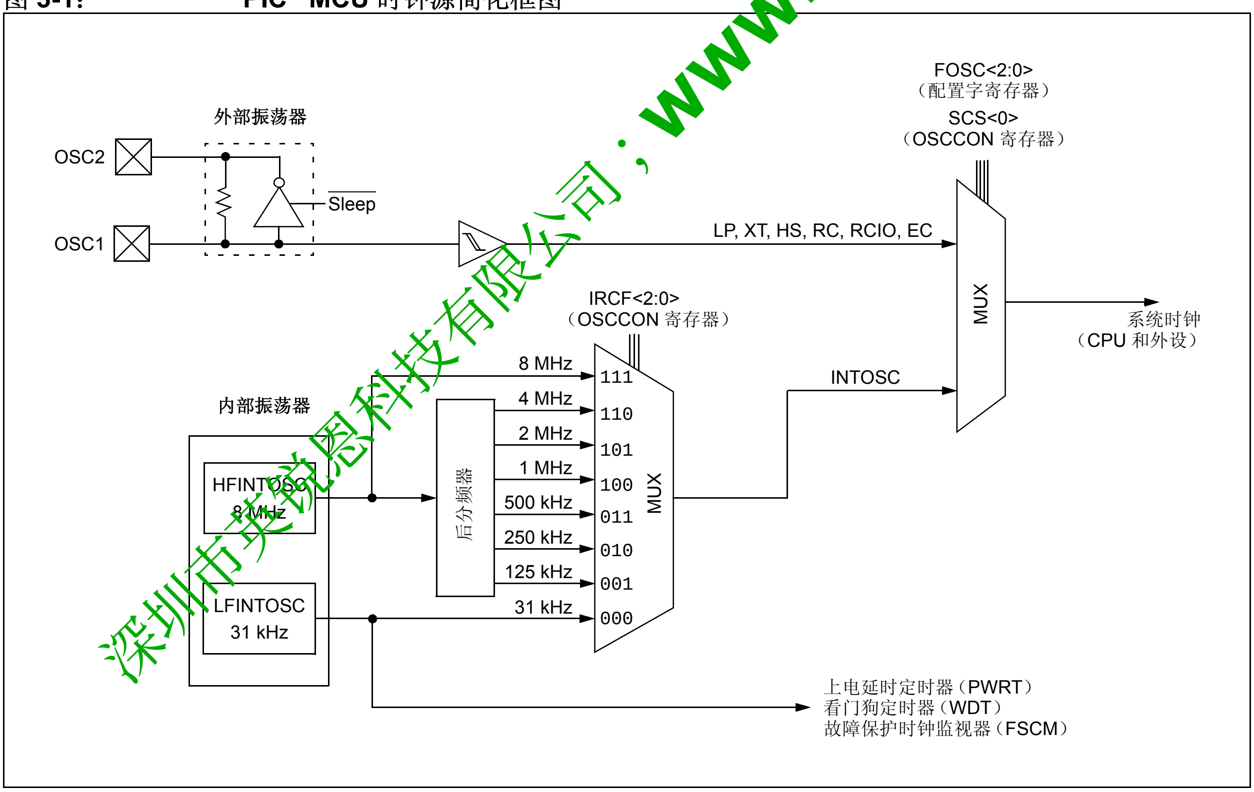
- 可通过软件选择外部或内部系统时钟源。
- 双速启动模式，最大限度地缩短外部振荡器起振与代码执行之间的延时。
- 故障保护时钟监视器（Fail-Safe Clock Monitor，FSCM），用来检测外部时钟源（LP、XT、HS、EC 或 RC 模式）故障以及自动切换到内部振荡器。

振荡器模块可配置为以下 8 种时钟模式之一。

1. EC——外部时钟，I/O 在 OSC2/CLKOUT 上。
2. LP——32 kHz 低功耗晶振模式。
3. XT——中等增益晶振或陶瓷谐振器模式。
4. HS——高增益晶振或陶瓷谐振器模式。
5. RC——外部阻容（RC），Fosc/4 输出到 OSC2/CLKOUT。
6. RCIO——外部阻容（RC），I/O 在 OSC2/CLKOUT 上。
7. INTOSC——内部振荡器，Fosc/4 输出到 OSC2 且 I/O 在 OSC1/CLKIN 上。
8. INTOSCIO——内部振荡器，I/O 在 OSC1/CLKIN 和 OSC2/CLKOUT 上。

通过配置字寄存器（CONFIG）中的 FOSC<2:0> 位来配置时钟源模式。内部时钟可用两个内部振荡器产生。HFINTOSC 是经过校准的高频振荡器。LFINTOSC 是未经校准的低频振荡器。

图 3-1： PIC® MCU 时钟源简化框图



3.2 振荡器控制

振荡器控制（OSCCON）寄存器（图 3-1）控制系统时钟和频率选择等选项。 OSCCON 寄存器包含以下位：

- 频率选择位 （IRCF）
- 频率状态位 （HTS 和 LTS）
- 系统时钟控制位 （OSTS 和 SCS）

寄存器 3-1： OSCCON：振荡器控制寄存器

U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

图注：		
R = 可读位	W = 可写位	U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未实现：**读为 0

bit 6-4 **IRCF<2:0>：**内部振荡器频率选择位
111 = 8 MHz
110 = 4 MHz （默认值）
101 = 2 MHz
100 = 1 MHz
011 = 500 kHz
010 = 250 kHz
001 = 125 kHz
000 = 31 kHz （LFINTOSC）

bit 3 **OSTS：**振荡器起振延时状态位 (1)
1 = 器件依靠 CONFIG 寄存器的 FOSC<2:0> 定义的时钟运行
0 = 器件依靠内部振荡器 （HFINTOSC 或 LFINTOSC）运行

bit 2 **HTS：**HFINTOSC 状态位 （高频——8 MHz 到 125 kHz）
1 = HFINTOSC 稳定
0 = HFINTOSC 不稳定

bit 1 **LTS：**LFINTOSC 状态位 （低频——31 kHz）
1 = LFINTOSC 稳定
0 = LFINTOSC 不稳定

bit 0 **SCS：**系统时钟选择位
1 = 内部振荡器用于系统时钟
0 = 时钟源由 CONFIG 寄存器的 FOSC<2:0> 决定

注 1： 双速启动且选取 LP、 XT 或 HS 为振荡器模式时，或者故障保护模式使能时，该位将复位为 0。

3.3 时钟源模式

时钟源模式可分为外部和内部模式。

- 外部时钟模式依靠外部电路提供时钟源。例子有：振荡器模块 （EC 模式）、石英晶体谐振器或陶瓷谐振器 （LP、 XT 和 HS 模式）以及阻容 （RC）模式电路。
- 内部时钟源内置于振荡器模块中。振荡器模块有两个内部振荡器：一个是 8 MHz 高频内部振荡器 （HFINTOSC），另一个是 31 kHz 低频内部振荡器 （LFINTOSC）。

可通过 OSCCON 寄存器的系统时钟选择 （SCS）位，在外部或内部时钟源之间选择系统时钟。更多信息，请参见**第 3.6 节 “时钟切换”**。

3.4 外部时钟模式

3.4.1 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）对来自 OSC1 的振荡计数 1024 次。这发生在上电复位 （POR）之后以及上电延时定时器（PWRT）延时结束 （如果配置了）时，或从休眠中唤醒后。在此期间，程序计数器不递增，程序执行暂停。OST 确保使用石英晶体谐振器或陶瓷谐振器的振荡器电路已经启动并向振荡器模块提供稳定的系统时钟信号。当在时钟源之间切换时，需要一定的延时以使新时钟稳定。表 3-1 给出了振荡器延时的例子。

为了使外部振荡器起振和代码执行之间的延时最小，可选择双速时钟启动模式（见**第 3.7 节 “双速时钟启动模式”**）。

表 3-1： 振荡器延时示例

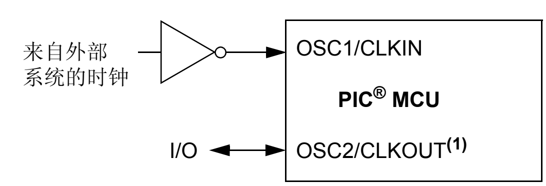
切换自	切换到	频率	振荡器延时
休眠 /POR	LFINTOSC HFINTOSC	31 kHz 125 kHz 到 8 MHz	振荡器预热延时 （TWARM）
休眠 /POR	EC 或 RC	DC － 20 MHz	双周期
LFINTOSC （31 kHz）	EC 或 RC	DC － 20 MHz	每次一周期
休眠 /POR	LP、 XT 或 HS	32 kHz 到 20 MHz	1024 个时钟周期 （OST）
LFINTOSC （31 kHz）	HFINTOSC	125 kHz 到 8 MHz	1 μs （近似值）

3.4.2 EC 模式

外部时钟 （EC）模式允许外部产生的逻辑电平作为系统时钟源。工作在此模式下时，外部时钟源连接到 OSC1 输入，OSC2 引脚可用作通用 I/O。图 3-2 给出了 EC 模式的引脚连接。

当选取 EC 模式时，振荡器起振定时器 （OST）被禁止。因此，上电复位（POR）后或者从休眠中唤醒后的操作不存在延时。因为 PIC® MCU 的设计是完全静态的，停止外部时钟输入将使器件暂停工作并保持所有数据完整。当再次启动外部时钟时，器件恢复工作，就好像没有停止过一样。

图 3-2： 外部时钟 （EC）模式的工作原理



注 1： 此引脚的其他功能列在**第 1.0 节 “器件概述”**中。

3.4.3 LP、 XT 和 HS 模式

LP、XT 和 HS 模式支持使用连接到 OSC1 和 OSC2 的石英晶体谐振器或陶瓷谐振器（图 3-3）。模式选择内部反相放大器的低、中或高增益设定，以支持各种谐振器类型及速度。

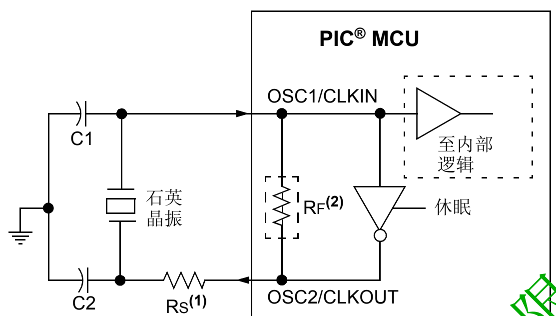
LP 振荡器模式选择内部反相放大器的最低增益设定。LP模式的电流消耗在三种模式中最小。该模式设计仅用于驱动 32.768 kHz 音叉（Tuning Fork）式晶振（钟表晶振）。

XT 振荡器模式选择内部反相放大器的中等增益设定。XT 模式的电流消耗在三种模式中居中。该模式最适用于驱动具备中等驱动电平规格要求的谐振器。

HS 振荡器模式选择内部反相放大器的最高增益设定。HS 模式的电流消耗在三种模式中最大。该模式最适用于驱动需要高驱动设定的谐振器。

图 3-3 和图 3-4 分别给出了石英晶体谐振器和陶瓷谐振器的典型电路。

图 3-3： 石英晶振的工作原理（**LP、 XT 或 HS 模式**）



注 1： 低驱动电平的石英晶振可能需要串联一个电阻（RS）。

2： RF 的值随所选振荡模式变化（典型值在 2 MΩ 到 10 MΩ 之间）。

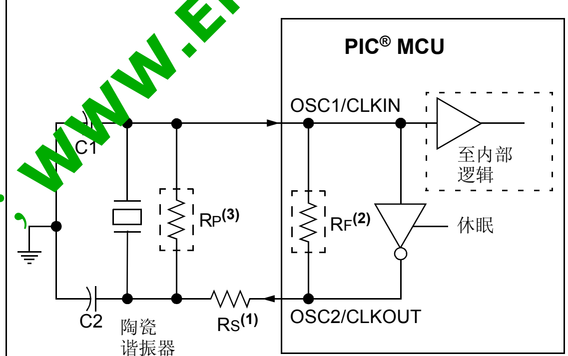
注 1： 石英晶振的特性随类型、封装和制造商而变化。要了解规格说明和推荐应用，应查阅制造商提供的数据手册。

2： 应始终验证振荡器在应用预期的 VDD 和温度范围内的性能。

3： 如需振荡器设计的帮助，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）

图 3-4： 陶瓷谐振器的工作原理（**XT 或 HS 模式**）



注 1： 低驱动电平的陶瓷谐振器可能需要串联一个电阻（RS）。

2： RF 的值随所选振荡模式变化（典型值在 2 MΩ 到 10 MΩ 之间）。

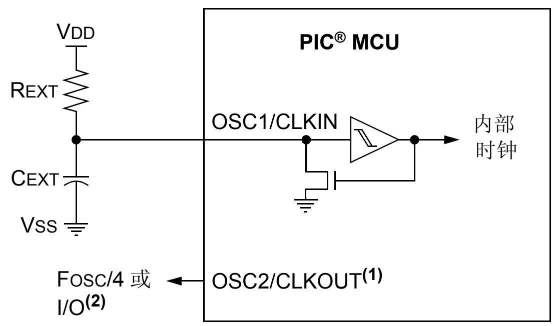
3： 要使陶瓷谐振器正常工作，可能需要并联一个反馈电阻（RP）。

3.4.4 外部 RC 模式

外部阻容（RC）模式支持使用外部 RC 电路。对时钟精度要求不高时，这使设计人员有了很大的频率选择空间，且保持成本最低。有 RC 和 RCIO 两种模式。

在 RC 模式下，RC 电路连接到 OSC1。OSC2/CLKOUT 输出 RC 振荡器频率的 4 分频。该信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。图 3-5 给出了外部 RC 模式的连接图。

图 3-5： 外部 RC 模式



建议值： 10 kΩ ≤ REXT ≤ 100 kΩ， <3V
3 kΩ ≤ REXT ≤ 100 kΩ， 3-5V
CEXT > 20 pF， 2-5V

注 1： 此引脚的其他功能列在**第 1.0 节 “器件概述”**中。
2： 输出取决于 RC 或 RCIO 时钟模式。

在 RCIO 模式下， RC 电路连接到 OSC1。 OSC2 成为额外的通用 I/O 引脚。

RC 振荡器频率与供电电压、电阻（REXT）和电容（CEXT）值以及工作温度有关。影响振荡器频率的其他因素有：

- 电压门限值变化
- 元件容差
- 不同封装的电容

用户还应考虑因所使用的外部 RC 元件的容差而导致的差异。

3.5 内部时钟模式

振荡器模块有两个独立的内部振荡器，可配置或选取为系统时钟源。

1. **HFINTOSC**（高频内部振荡器）出厂时已校准，工作频率为 8 MHz。使用 OSCTUNE 寄存器（寄存器 3-2），用户可通过软件调整 HFINTOSC 的频率。
2. **LFINTOSC**（低频内部振荡器）未经校准，工作频率为 31 kHz。

通过软件对 OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0> 进行操作，可选择系统时钟速度。

可通过 OSCCON 寄存器的系统时钟选择（SCS）位，在外部或内部时钟源之间选择系统时钟。更多信息，请参见**第 3.6 节 “时钟切换”**。

3.5.1 INTOSC 和 INTOSCIO 模式

当在配置字寄存器（CONFIG）中使用振荡器选择位 FOSC<2:0> 设置了相应值时，在 INTOSC 和 INTOSCIO 模式下将内部振荡器配置为系统时钟源。

在 **INTOSC** 模式下， OSC1/CLKIN 可用作通用 I/O。OSC2/CLKOUT 输出所选内部振荡器频率的 4 分频。CLKOUT 信号可用来为外部电路、同步、校准、测试或其他应用需求提供时钟。

在 **INTOSCIO** 模式下，OSC1/CLKIN 和 OSC2/CLKOUT 可用作通用 I/O。

3.5.2 HFINTOSC

高频内部振荡器（HFINTOSC）是出厂时已校准的8 MHz 内部时钟源。使用 OSCTUNE 寄存器（寄存器 3-2），可通过软件调整 HFINTOSC 的频率。

HFINTOSC 的输出连接到后分频器和多路开关（见图 3-1）。使用 OSCCON 寄存器的 IRCF<2:0> 位，可通过软件选择七个频率之一。更多信息，请参见**第 3.5.4 节 “频率选择位（IRCF）”**。

将 OSCCON 寄存器的 IRCF<2:0> 位设置为 ≠ `000` 选择 8 MHz到125 kHz之间的任一频率，可使能HFINTOSC。然后将 OSCCON 寄存器的系统时钟源（SCS）位设置为 `1`，或通过将配置字寄存器（CONFIG）中的 IESO 位设置为 `1` 使能双速启动。

OSCCON 寄存器的 HF 内部振荡器（HTS）位用于指示 HFINTOSC 是否稳定。

3.5.2.1 OSCTUNE 寄存器

HFINTOSC 在出厂时已校准，但可通过在软件中写入 OSCTUNE 寄存器 （寄存器 3-2）来进行调节。

OSCTUNE 寄存器的默认值为 0。该值是一个 5 位的二进制补码。

当 OSCTUNE 寄存器被修改时， HFINTOSC 频率将开始转变为新频率。转变期间，代码将继续执行。是否已发生频率转变并无明确的指示。

OSCTUNE 不影响 LFINTOSC 频率。依赖 LFINTOSC 时钟源工作的功能，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）以及外设等，其工作不受频率变化的影响。

寄存器 3-2： OSCTUNE：振荡器调节寄存器

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

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **未实现：**读为 0

bit 4-0 **TUN<4:0>：**频率调节位
01111 = 最高频率
01110 =
•
•
•
00001 =
00000 = 振荡器模块运行在出厂前校准的频率上
11111 =
•
•
•
10000 = 最低频率

3.5.3 LFINTOSC

低频内部振荡器（LFINTOSC）是未经校准的 31 kHz 内部时钟源。

LFINTOSC 的输出连接到后分频器和多路开关（见图 3-1）。通过软件对 OSCCON 寄存器的 IRCF<2:0> 位进行操作，选取 31 kHz。更多信息，请参见**第 3.5.4 节 “频率选择位（IRCF）”**。LFINTOSC 还是上电延时定时器（PWRT）、看门狗定时器（WDT）以及故障保护时钟监视器（FSCM）的时钟源。

选取 31 kHz（将 OSCCON 寄存器的 IRCF<2:0> 位设置为 `000`）为系统时钟源（OSCCON 寄存器的 SCS 位 = `1`），或者使能以下任一项时，LFINTOSC 将被使能：

- 双速启动（配置字寄存器的 IESO 位 = `1` 且 OSCCON 寄存器的 IRCF<2:0> 位 = `000`）
- 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）

OSCCON 寄存器的 LF 内部振荡器（LTS）位用于指示 HFINTOSC 是否稳定。

3.5.4 频率选择位（IRCF）

8 MHz HFINTOSC 和 31 kHz LFINTOSC 的输出连接到后分频器和多路开关（见图 3-1）。OSCCON 寄存器的内部振荡器频率选择位 IRCF<2:0> 用于选择内部振荡器的频率输出。可通过软件选择以下 8 种频率之一：

- 8 MHz
- 4 MHz（复位后的默认值）
- 2 MHz
- 1 MHz
- 500 kHz
- 250 kHz
- 125 kHz
- 31 kHz（LFINTOSC）

注：	任何复位后，OSCCON 寄存器的 IRCF<2:0> 位将被设置为 `110` 且频率选择设置为 4 MHz。用户可修改 IRCF 位来选择其他频率。

3.5.5 HFINTOSC 和 LFINTOSC 时钟切换时序

当在 LFINTOSC 和 HFINTOSC 之间切换时，新的振荡器可能为了省电已经关闭（见图 3-6）。在这种情况下，OSCCON 寄存器的 IRCF<2:0> 位被修改之后、频率选择生效之前，存在一个延时。OSCCON 寄存器的 LTS 和 HTS 位将反映 LFINTOSC 和 HFINTOSC 振荡器的当前活动状态。频率选择时序如下：

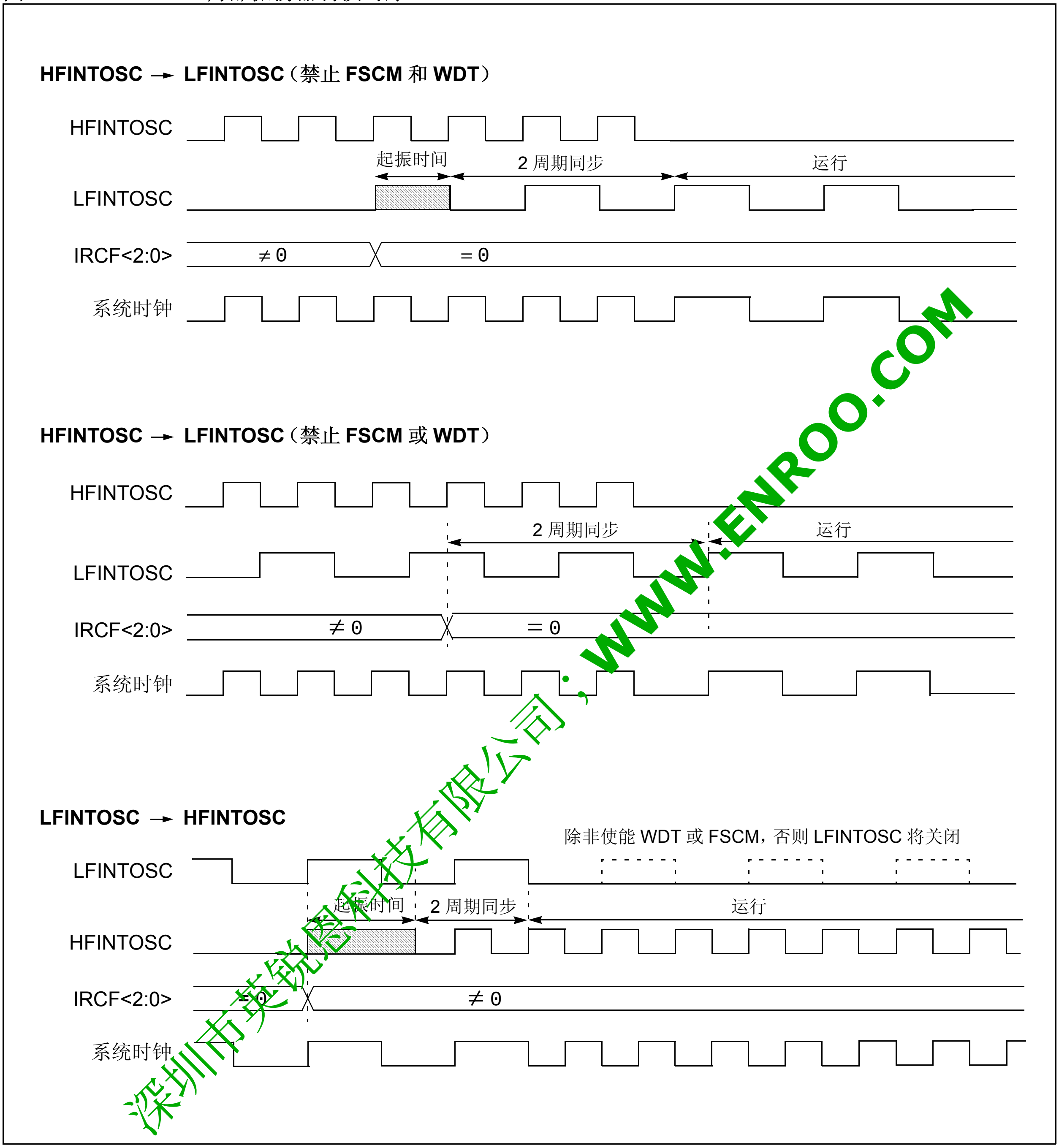
1. OSCCON 寄存器的 IRCF<2:0> 位被修改。
2. 如果新时钟是关闭的，开始一个时钟启动延时。
3. 时钟切换电路等待当前时钟下降沿的到来。
4. CLKOUT 保持为低电平，时钟切换电路等待新时钟上升沿的到来。
5. 现在 CLKOUT 连接到新时钟。OSCCON 寄存器的 LTS 和 HTS 位按要求被更新。
6. 时钟切换完成。

更多详细信息，请参见图 3-1。

如果选取的内部振荡器速度在 8 MHz 到 125 kHz 之间，选取新频率不存在启动延时。这是因为新旧频率都来自经过后分频器和多路开关的 HFINTOSC。

启动延时规范在**第 17.0 节 “电气规范”**中与振荡器相关的表格中。

图 3-6： 内部振荡器切换时序



3.6 时钟切换

通过软件对 OSCCON 寄存器的系统时钟选择 （SCS）位进行操作，可将系统时钟源在外部和内部时钟源之间切换。

3.6.1 系统时钟选择 （SCS）位

OSCCON 寄存器的系统时钟选择 （SCS）位选择用于 CPU 和外设的系统时钟源。

- OSCCON寄存器的SCS位 = `0`时，系统时钟源由配置字寄存器 （CONFIG）中的 FOSC<2:0> 位的配置决定。
- OSCCON 寄存器的 SCS 位 = `1` 时，根据 OSCCON 寄存器的 IRCF<2:0> 位所选的内部振荡器频率选取系统时钟源。复位后， OSCCON 寄存器的 SCS 位总是被清零。

> **注：** 任何自动时钟切换 （可能由于双速启动或故障保护时钟监视器）都不更新 OSCCON 寄存器的 SCS 位。用户可监控 OSCCON 寄存器的 OSTS 位以确定当前的系统时钟源。

3.6.2 振荡器起振超时状态 （OSTS）位

OSCCON 寄存器的振荡器起振超时状态 （OSTS）位用于指示系统时钟是来自外部时钟源（通过配置字寄存器 （CONFIG）中的 FOSC<2:0> 位定义），还是来自内部时钟源。OSTS 还特别指明在 LP、XT 或 HS 模式下，振荡器起振定时器 （OST）是否已超时。

3.7 双速时钟启动模式

双速启动模式通过最大限度地缩短外部振荡器起振与代码执行之间的延时，进一步节省了功耗。对于频繁使用休眠模式的应用，双速启动模式将从器件唤醒的时间中去除外部振荡器的起振时间，从而可降低器件的总体功耗。

该模式使得应用能够从休眠中唤醒，将 INTOSC 用作时钟源执行数条指令，然后再返回休眠状态而无需等待主振荡器的稳定。

> **注：** 执行 SLEEP 指令将中止振荡器起振时间，并使 OSCCON 寄存器的 OSTS 位保持清零。

当振荡器模块配置为 LP、 XT 或 HS 模式时，振荡器起振定时器 （OST）使能 （见第 **3.4.1 节 “振荡器起振定时器（OST）”**）。OST 将暂停程序执行，直到完成 1024 次振荡计数。双速启动模式在 OST 计数时使用内部振荡器进行工作，最大限度地缩短了代码执行的延时。当 OST 计数到 1024 且 OSCCON 寄存器的 OSTS 位置 1 时，程序执行切换至外部振荡器。

3.7.1 双速启动模式配置

通过以下设定来配置双速启动模式：

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

在下列操作之后，进入双速启动模式：

- 上电复位 （POR）且上电延时定时器 （PWRT）延时结束 （使能时）后，或者
- 从休眠状态唤醒。

如果外部时钟振荡器配置为除 LP、 XT 或 HS 模式以外的任一模式，那么双速启动将被禁止。这是因为 POR 后或从休眠状态退出时，外部时钟振荡器不需要稳定时间。

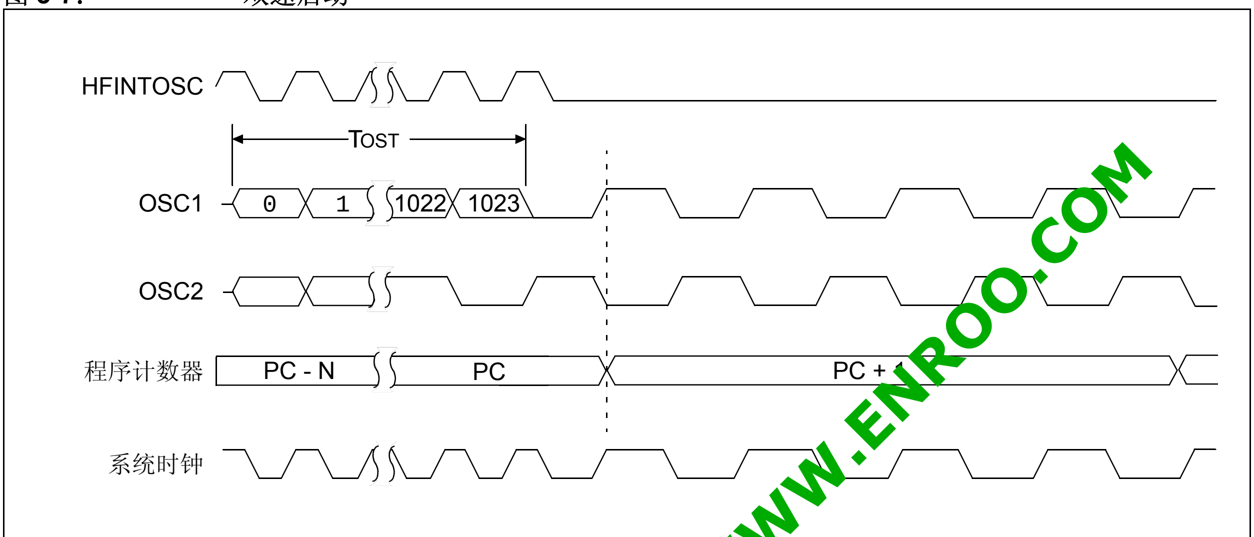
3.7.2 双速启动顺序

1. 从上电复位或休眠中唤醒。
2. 使用内部振荡器以 OSCCON 寄存器的 IRCF<2:0> 位设置的频率开始执行指令。
3. OST 使能，计数 1024 个时钟周期。
4. OST 超时，等待内部振荡器下降沿的到来。
5. OSTS 置 1。
6. 系统时钟保持为低电平，直到新时钟下一个下降沿的到来 （LP、 XT 或 HS 模式）。
7. 系统时钟切换到外部时钟源。

3.7.3 检查双速时钟状态

通过检查 OSCCON 寄存器的 OSTS 位的状态，可以确定单片机是否如配置字寄存器（CONFIG）中 FOSC<2:0> 位定义的那样依靠外部时钟源运行，抑或是依靠内部振荡器运行。

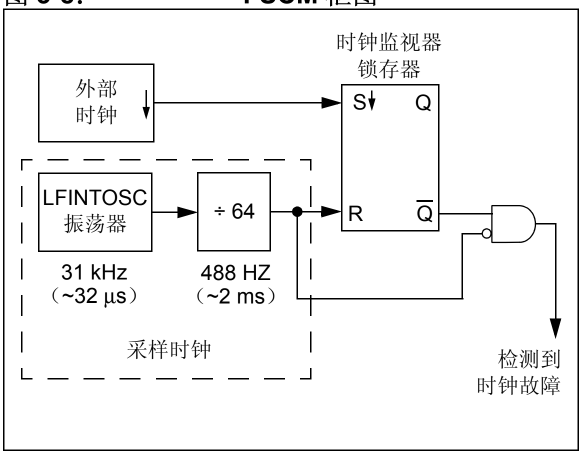
图 3-7： 双速启动



3.8 故障保护时钟监视器

故障保护时钟监视器（FSCM）使得器件在出现外部振荡器故障时仍能继续工作。FSCM 能在振荡器起振延时定时器（OST）延时结束后的任一时刻检测振荡器故障。FSCM 通过将配置字寄存器（CONFIG）中的 FCMEN 位置 1 来使能。FSCM 可用于所有外部振荡器模式（LP、XT、HS、EC、RC 和 RCIO）。

图 3-8： FSCM 框图



3.8.1 故障保护检测

FSCM 模块通过将外部振荡器与 FSCM 采样时钟比较来检测振荡器故障。LFINTOSC 除以 64，就产生了采样时钟。请参见图 3-8。故障检测器内部有一个锁存器。在外部时钟的每个下降沿，锁存器被置 1。在采样时钟的每个下降沿，锁存器被清零。如果在采样时钟的整个半周期后而主时钟依然未变为低电平，就检测到故障。

3.8.2 故障保护操作

当外部时钟出现故障时，FSCM 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。如果在 PIE2 寄存器的 OSFIE 位置 1 的同时将该标志位置 1，将产生中断。器件固件随后会采取措施减轻可能由故障时钟所产生的问题。系统时钟将继续来自内部时钟源，直到器件固件成功重启外部振荡器且切换回外部操作。

FSCM 所选的内部时钟源由 OSCCON 寄存器的 IRCF<2:0> 位决定。这使得可以在故障发生前配置内部振荡器。

3.8.3 故障保护条件清除

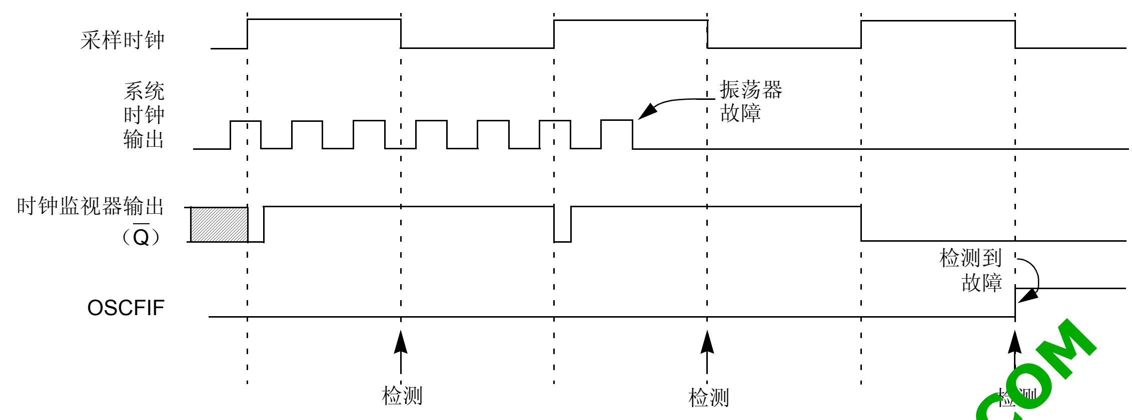
复位、执行 SLEEP 指令或翻转 OSCCON 寄存器的 SCS 位后，故障保护条件被清除。SCS 位被修改后，OST 将重新启动。OST 运行时，器件继续从 OSCCON 中选定的 INTOSC 进行操作。OST 超时后，故障保护条件被清除，器件将从外部时钟源进行操作。必须先清除故障保护条件，才能清零 OSFIF 标志位。

3.8.4 复位或从休眠中唤醒

FSCM 设计为能在振荡器起振延时定时器（OST）延时结束后的任一时刻检测振荡器故障。OST 用在从休眠状态唤醒后以及任何类型的复位后。OST 不能在 EC 或 RC 时钟模式下使用，所以一旦复位或唤醒完成，FSCM 就处于活动状态。当 FSCM 被使能时，双速启动也被使能。因此，当 OST 运行时，器件总是处于代码执行阶段。

> **注：** 由于振荡器起振时间的范围变化较大，在振荡器起振期间（即，从复位或休眠中退出时），故障保护电路不处于活动状态。经过一段适当的时间后，用户应检查 OSCCON 寄存器的 OSTS 位，以验证振荡器是否已成功起振以及系统时钟是否切换成功。

图 3-9：　　　　FSCM 时序图



注：　　系统时钟频率通常远高于采样时钟频率。本例中为便于说明选用相对频率。

表 3-2：　　　　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	POR/BOR 时的值	所有其他复位时的值 (1)
CONFIG(2)	CPD	CP	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0	—	—
OSCCON	—	IRCF2	IRCF1	IRCF0	OSTS	HTS	LTS	SCS	-110 x000	-110 x000
OSCTUNE	—	—	—	TUN4	TUN3	TUN2	TUN1	TUN0	---0 0000	---u uuuu
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000

图注：　x = 未知， u = 不变， – = 未实现单元 （读为 0）。振荡器不使用阴影单元。
注　1：　其他 （非上电）复位包括正常工作期间的 $\overline{MCLR}$ 复位和看门狗定时器复位。
　　2：　请参见配置字寄存器 （寄存器 14-1）了解所有寄存器位的操作。

4.0 I/O 端口

器件有多达 18 个可用的通用 I/O 引脚。根据外设的使能情况，部分甚至全部引脚可能不能用于通用 I/O。通常而言，当某个外设使能时，其相关引脚可能不能用作通用 I/O 引脚。

4.1 PORTA 和 TRISA 寄存器

PORTA 是一个 6 位宽的双向端口，对应的数据方向寄存器是 TRISA（寄存器 4-2）。将 TRISA 某位置 1（= `1`）时，会将 PORTA 的相应引脚设为输入（即，禁止输出驱动器）。将 TRISA 某位清零（= `0`）时，会将 PORTA 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选中引脚）。RA3 是个例外，仅可作为输入引脚，其 TRIS 位总是读为 `1`。例 4-1 显示了如何初始化 PORTA。

读 PORTA 寄存器（寄存器 4-1）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是 “读 - 修改 - 写” 操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器。当 MCLRE = `1` 时，RA3 读为 `0`。

TRISA 寄存器控制 PORTA 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保 TRISA 寄存器中的各位保持置 1。配置为模拟输入的 I/O 引脚总是读为 `0`。

> **注：** 必须对 ANSEL 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 `0`。

例 4-1： 初始化 PORTA

```
BCF    STATUS,RP0  ;Bank 0
BCF    STATUS,RP1  ;
CLRF   PORTA       ;Init PORTA
BSF    STATUS,RP1  ;Bank 2
CLRF   ANSEL       ;digital I/O
BSF    STATUS,RP0  ;Bank 1
BCF    STATUS,RP1  ;
MOVLW  0Ch         ;Set RA<3:2> as inputs
MOVWF  TRISA       ;and set RA<5:4,1:0>
                   ;as outputs
BCF    STATUS,RP0  ;Bank 0
```

寄存器 4-1： PORTA：PORTA 寄存器

U-0	U-0	R/W-x	R/W-x	R-x	R/W-x	R/W-x	R/W-x
—	—	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：** 读为 `0`

bit 5-0 **RA<5:0>：** PORTA I/O 引脚位
`1` = 端口引脚大于 V_{IH}
`0` = 端口引脚小于 V_{IL}

寄存器 4-2： TRISA：PORTA 三态寄存器

U-0	U-0	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-1
—	—	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现：** 读为 `0`

bit 5-0 **TRISA<5:0>：** 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

注 **1：** TRISA<3> 总是读为 `1`。
2： 在 XT、HS 和 LP 振荡器模式下，TRISA<5:4> 总是读为 `1`。

4.2 引脚的其他功能

本器件系列中每个 PORTA 引脚都具有电平变化中断功能和弱上拉功能。RA0 还有超低功耗唤醒功能。以下三节将对这些功能进行介绍。

4.2.1 ANSEL 和 ANSELH 寄存器

ANSEL 和 ANSELH 寄存器用于禁止 I/O 引脚的输入缓冲器，从而使模拟电压可施加在这些引脚上而不会引起电流过大。将相应引脚的 ANSx 位置 1 可使所有对该引脚的数字读操作返回 `0`，并使该引脚的模拟功能正常工作。

ANSx 位的状态对对应引脚的数字输出功能没有影响。TRISx 位清零而 ANSx 位置 1 的引脚将作为数字输出工作，同时具有该引脚的模拟输入功能。 ANSx 位置 1 的引脚总是读为 `0`，在执行读或写操作时这将导致意外的操作，这是因为所有读写操作均执行读 - 修改 - 写序列。

4.2.2 弱上拉

除 RA3 外的每个 PORTA 引脚都具有可单独配置的内部弱上拉功能。控制位 WPUAx 用于使能或禁止每个上拉功能。请参见寄存器 4-4。当端口引脚被配置为输出时，这些弱上拉会自动关闭。上电复位时，选项寄存器的 RABPU 位将禁止上拉功能。当配置为 MCLR 时， RA3 上的弱上拉自动使能；当 RA3 配置为 I/O 时则被禁止。没有对 MCLR 上拉的软件控制。

4.2.3 电平变化中断

每个 PORTA 引脚都可单独配置为电平变化中断引脚。控制位 IOCAx 用于允许或禁止各引脚的中断功能。请参见寄存器4-6。电平变化中断功能在上电复位时被禁止。

对于允许了电平变化中断功能的引脚，其值将与上次读取的 PORTA 的旧锁存值相比较。所有与上次读取值不匹配的输出进行或运算，运算结果用来设置 INTCON 寄存器 （寄存器 2-6）中的 PORTA 电平变化中断标志位（RABIF）。

该中断可将器件从休眠状态唤醒。用户在中断服务程序中可通过以下方式清除该中断：

a) 对 PORTA 的任何读或写操作。这将结束不匹配条件，然后：
b) 清零标志位 RABIF。

不匹配条件将继续把标志位 RABIF 置 1。读 PORTA 将结束不匹配条件并将标志位 RABIF 清零。保持上一次读取值的锁存器不受 MCLR 或 BOR 复位的影响。在这些复位之后，如果存在不匹配情况，RABIF 标志还将继续被置 1。

注：	当读操作正在执行时发生了I/O引脚电平变化（Q2 周期的起始时刻），则 RABIF 中断标志位可能不会被置 1。

寄存器 4-3： ANSEL：模拟选择寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 **ANS<7:0>**：模拟选择位
在 AN<7:0> 引脚上分别进行模拟或数字功能的模拟选择。
`1` = 模拟输入。引脚被配置为模拟输入 (1)。
`0` = 数字 I/O。引脚被配置为端口或特殊功能。

注 **1**： 将引脚设置为模拟输入将自动禁止数字输入电路、弱上拉和电平变化中断 （如果可用）。必须将相应的 TRIS 位设置为输入模式，以允许从外部控制引脚电压。

寄存器 4-4： ANSELH：模拟选择高位寄存器 (2)

U-0	U-0	U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1
—	—	—	—	ANS11	ANS10	ANS9	ANS8
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **未实现**：读为 `0`

bit 3-0 **ANS<11:8>**：模拟选择位
在 AN<7:0> 引脚上分别进行模拟或数字功能的模拟选择。
`1` = 模拟输入。引脚被配置为模拟输入 (1)。
`0` = 数字 I/O。引脚被配置为端口或特殊功能。

注 **1**： 将引脚设置为模拟输入将自动禁止数字输入电路、弱上拉和电平变化中断 （如果可用）。必须将相应的 TRIS 位设置为输入模式，以允许从外部控制引脚电压。
2： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

寄存器 4-5： WPUA：PORTA 寄存器

U-0	U-0	R/W-1	R/W-1	U-0	R/W-1	R/W-1	R/W-1
—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-4 **WPUA<5:4>：**弱上拉寄存器位
1 = 上拉使能
0 = 上拉禁止

bit 3 未实现：读为 0

bit 2-0 **WPUA<2:0>：**弱上拉寄存器位
1 = 上拉使能
0 = 上拉禁止

注 **1：** 必须使能选项寄存器的全局 $\overline{\text{RABPU}}$ 位以使能各个上拉。
2： 如果引脚处于输出模式 （TRISA = 0），则自动禁止弱上拉器件。
3： 当在配置字中引脚配置为 $\overline{\text{MCLR}}$ 时， RA3 上拉被使能，引脚配置为 I/O 时则被禁止。
4： 在 XT、 HS 和 LP 振荡器模式下， WPUA<5:4> 总是读为 1。

寄存器 4-6： IOCA：电平变化中断 PORTA 寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 未实现：读为 0

bit 5-0 **IOCA<5:0>：**电平变化中断 PORTA 控制位
1 = 允许电平变化中断
0 = 禁止电平变化中断

注 **1：** 要使各中断能够被识别，必须使能全局中断允许位 （GIE）。
2： 在 XT、 HS 和 LP 振荡器模式下， IOCA<5:4> 总是读为 1。

4.2.4 超低功耗唤醒

RA0 上的超低功耗唤醒（ULPWU）功能允许缓慢下降的电压能够在 RA0 上产生电平变化中断，同时不消耗很大的电流。将 PCON 寄存器的 ULPWUE 位置 1 将选取超低功耗唤醒模式。这将产生一个小的灌电流，可用来对 RA0 上电容器进行放电。

要使用此功能，请按以下步骤操作：

a) 将 RA0 引脚配置为输出（= 1）为 RA0 上的电容充电。
b) 将 RA0 配置为输入。
c) 允许 RA0 的电平变化中断。
d) 将 PCON 寄存器的 ULPWUE 位置 1 开始为电容放电。
e) 执行一条 SLEEP 指令。

当 RA0 上电压下降到低于 V_{IL} 时，将产生中断，唤醒器件并执行下一条指令。如果 INTCON 寄存器的 GIE 位置 1，器件随后将调用中断向量（0004h）。更多信息，请参见**第 4.4.2 节 “电平变化中断”**和**第 14.3.3 节 “PORTA/PORTB 中断”**。

该功能提供了一种周期性将器件从休眠中唤醒的低功耗方法。休眠时间取决于 RA0 上 RC 电路的放电时间。超低功耗唤醒模块的初始化，请参见例 4-2。

在 RA0 与外部电容之间有一个串联电阻，它为 RA0/AN0/C1IN+/ICSPDAT/ULPWU 引脚提供过流保护，同时允许使用软件校准延时（见图 4-1）。可使用一个定时器测量电容器的充放电时间。然后调节充电时间，以提供所需的中断延时。此方法将对温度、电压和元件精度的影响进行补偿。超低功耗唤醒外设还可配置为简单的可编程低压检测功能或温度传感器。

注：	更多信息，请参见应用笔记 AN879，*“Using the Microchip Ultra Low-Power Wake-up Module”*（DS00879）。

例 4-2： 超低功耗唤醒初始化

```
BCF     STATUS,RP0      ;Bank 0
BCF     STATUS,RP1      ;
BSF     PORTA,0         ;Set RA0 data latch
BSF     STATUS,RP1      ;Bank 2
BCF     ANSEL,0         ;RA0 to digital I/O
BSF     STATUS,RP0      ;Bank 1
BCF     STATUS,RP1      ;
BCF     TRISA,0         ;Output high to
CALL    CapDelay        ;charge capacitor
BSF     PCON,ULPWUE     ;Enable ULP Wake-up
BSF     IOCA,0          ;Select RA0 IOC
BSF     TRISA,0         ;RA0 to input
MOVLW   B'10001000'     ;Enable interrupt
MOVWF   INTCON          ;and clear flag
BCF     STATUS,RP0      ;Bank 0
SLEEP                   ;Wait for IOC
NOP                     ;
```

4.2.5 引脚说明和引脚原理图

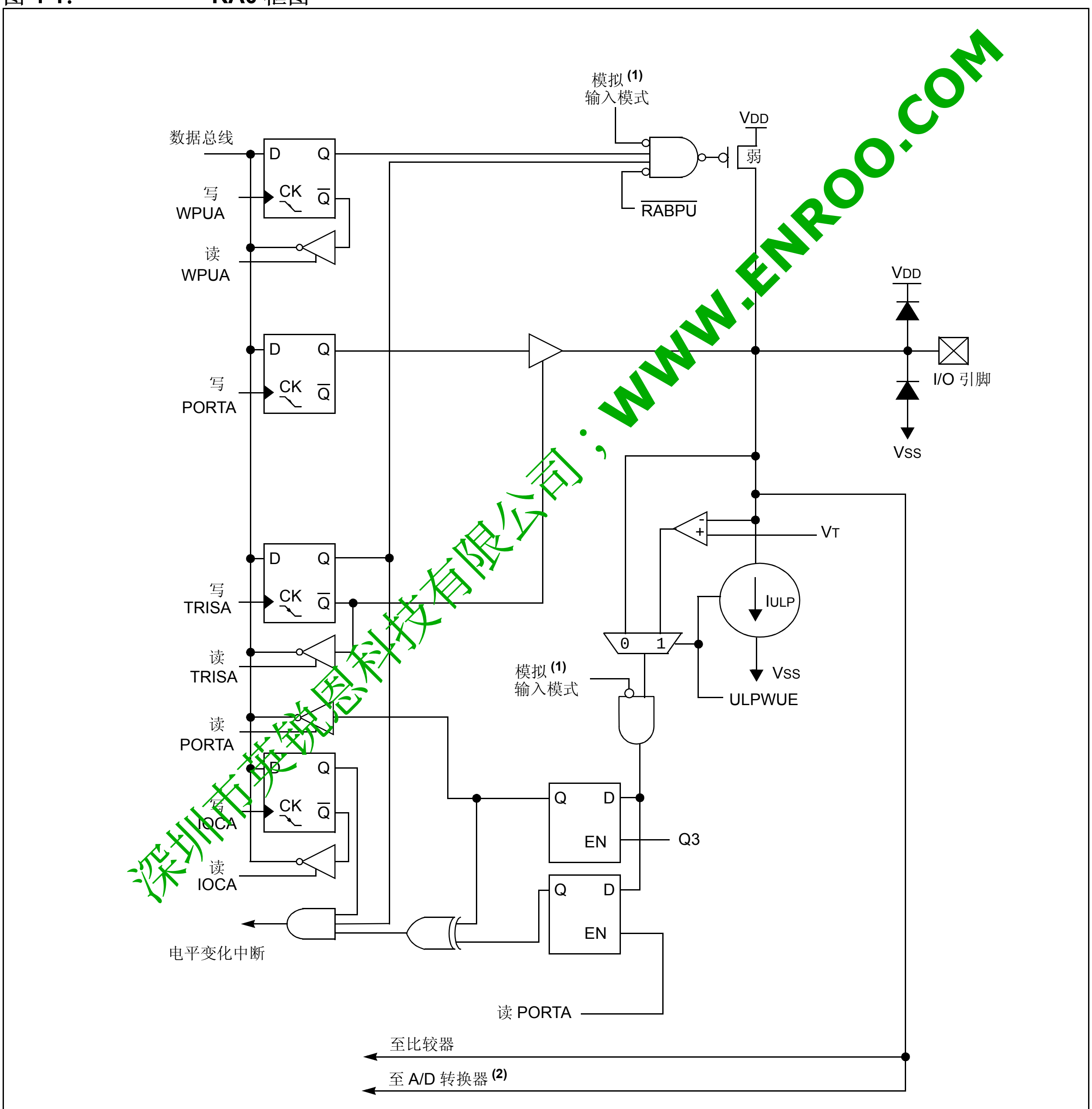
每个 PORTA 引脚都与其他功能复用。这里将简要说明引脚及其复用功能。各功能如比较器或 A/D 转换器（ADC）的具体信息，请参见本数据手册中的相关章节。

4.2.5.1 RA0/AN0/C1IN+/ICSPDAT/ULPWU

图4-2为此引脚的原理图。RA0/AN0/C1IN+/ICSPDAT/ULPWU 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 连接至比较器 C1 的模拟输入
- 在线串行编程数据
- 超低功耗唤醒的模拟输入

图 4-1： RA0 框图



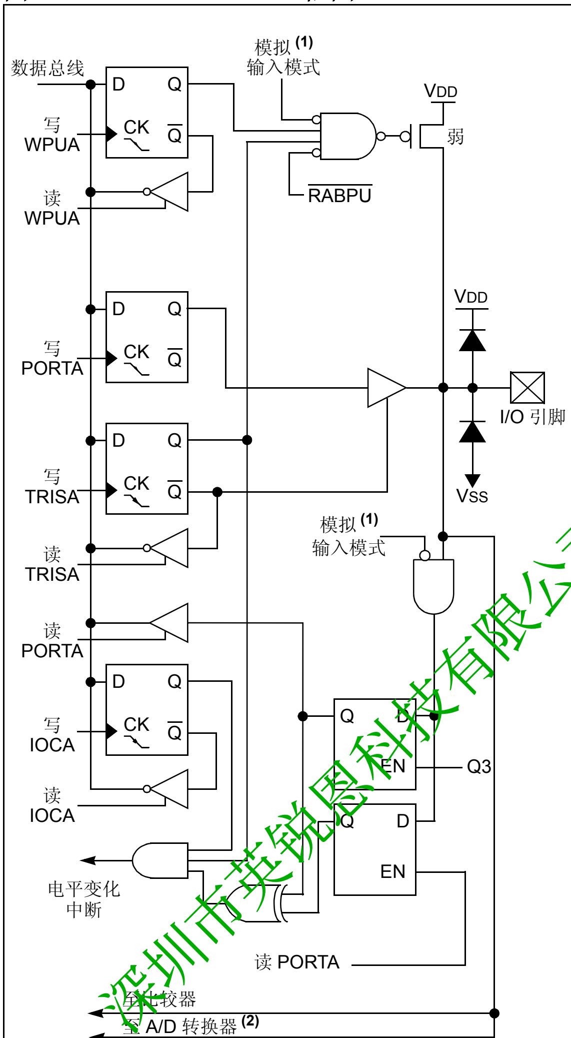
注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

4.2.5.2 RA1/AN1/C12IN0-/VREF/ICSPCLK

图4-2为此引脚的原理图。RA1/AN1/C12IN0-/VREF/ICSPCLK 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 连接至比较器 C1 或 C2 的模拟输入
- 连接至 ADC 的参考电压输入
- 在线串行编程时钟

图 4-2： RA1 框图



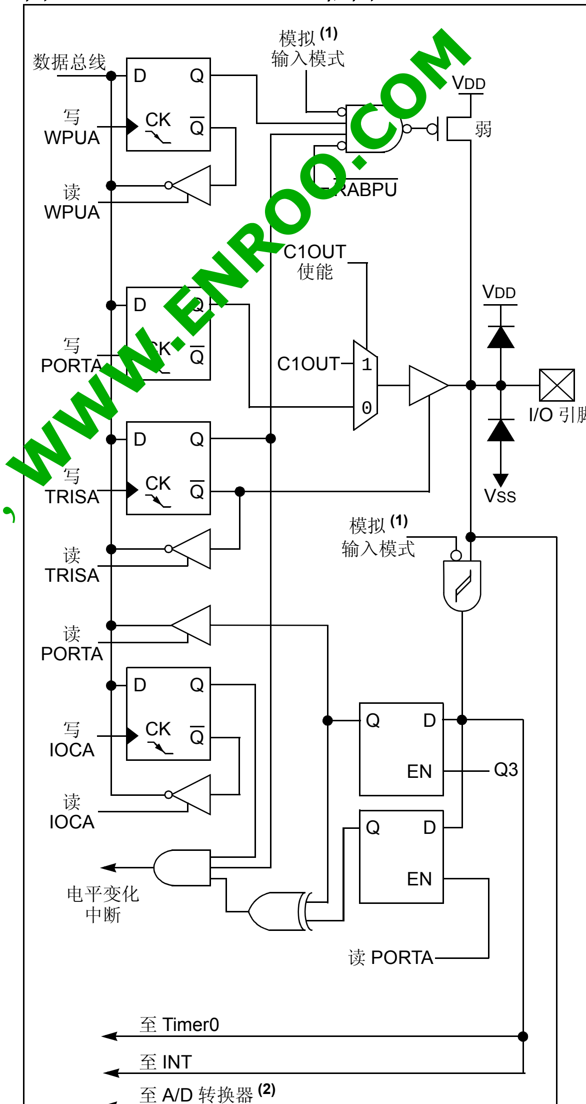
注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

4.2.5.3 RA2/AN2/T0CKI/INT/C1OUT

图 4-3 为此引脚的原理图。RA2/AN2/T0CKI/INT/C1OUT 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- Timer0 的时钟输入
- 外部边沿触发的中断
- 来自比较器 C1 的数字输出

图 4-3： RA2 框图



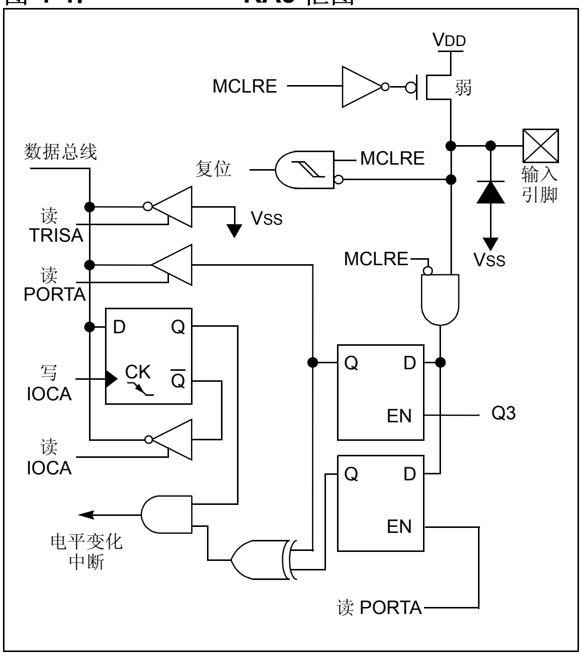
注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

4.2.5.4 RA3/$\overline{\text{MCLR}}$/VPP

图 4-4 为此引脚的原理图。RA3/$\overline{\text{MCLR}}$/VPP 引脚可配置为下列功能之一：

- 通用输入
- 带弱上拉的主复位

图 4-4： RA3 框图

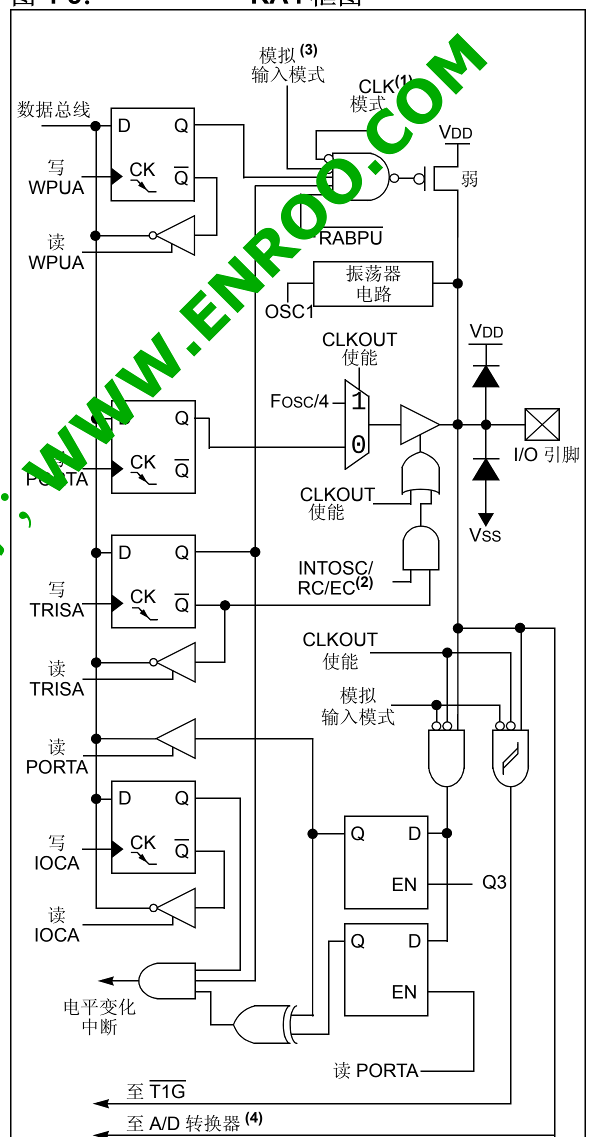


4.2.5.5 RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT

图4-5为此引脚的原理图。RA4/AN3/$\overline{\text{T1G}}$/OSC2/CLKOUT引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- Timer1 门控输入
- 晶振 / 谐振器连接
- 时钟输出

图 4-5： RA4 框图



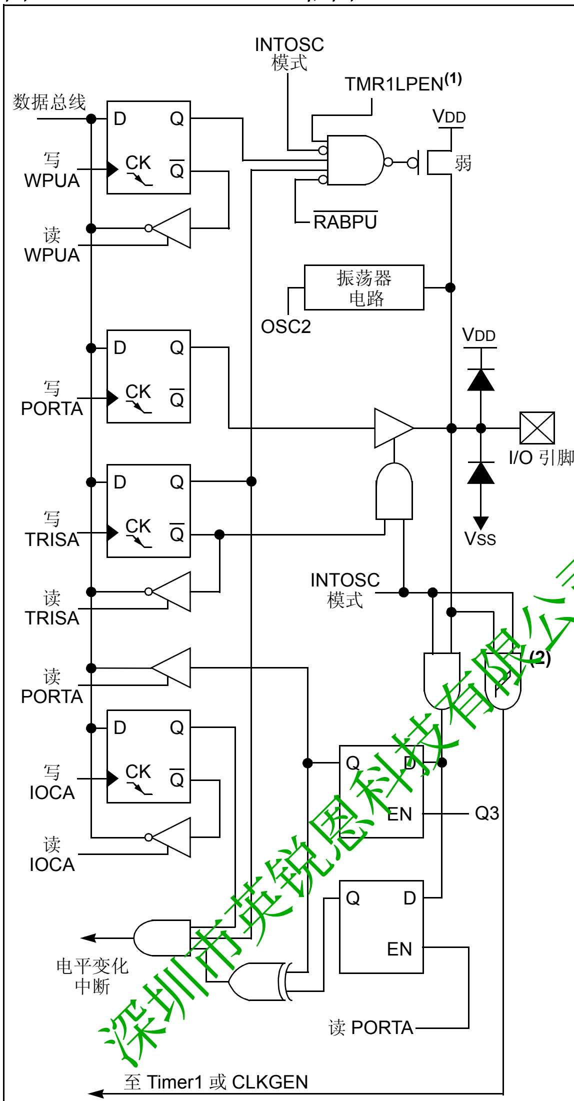
注 1：CLK 模式为 XT、HS、LP 和 LPTMR1，且 CLKOUT 使能。
2：具有 CLKOUT 选项。
3：ANSEL 决定模拟输入模式。
4：PIC16F631 上未实现。

4.2.5.6 RA5/T1CKI/OSC1/CLKIN

图 4-6 为此引脚的原理图。RA5/T1CKI/OSC1/CLKIN 引脚可配置为下列功能之一：

- 通用 I/O
- Timer1 时钟输入
- 晶振 / 谐振器连接
- 时钟输入

图 4-6： RA5 框图



注 1：Timer1 LP 振荡器使能。

2：在使用带 LP 振荡器的 Timer1 时，施密特触发器被旁路。

表 4-1： 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	POR/BOR 时的值	所有其他复位时的值
ADCON0	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	0000 0000
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
IOCA	—	—	IOCA5	IOCA4	IOCA3	IOCA2	IOCA1	IOCA0	--00 0000	--00 0000
OPTION_REG	$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
WPUA	—	—	WPUA5	WPUA4	—	WPUA2	WPUA1	WPUA0	--11 -111	--11 -111

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

4.3 PORTB 和 TRISB 寄存器

PORTB 是一个 4 位宽的双向端口，对应的数据方向寄存器是 TRISB（寄存器 4-6）。将 TRISB 某位置 1 （= 1）时，会将 PORTB 的相应引脚设为输入（即，使相应的输出驱动器呈高阻状态）。将 TRISB 某位清零 （= 0）时，会将 PORTB 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选中引脚）。例 4-3 显示了如何初始化 PORTB。读 PORTB 寄存器（寄存器 4-5）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是 “读 - 修改 - 写” 操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器。

TRISB 寄存器控制 PORTB 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保 TRISB 寄存器中的各位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

例 4-3： 初始化 PORTB

```
BCF     STATUS,RP0  ;Bank 0
BCF     STATUS,RP1  ;
CLRF    PORTB       ;Init PORTB
BSF     STATUS,RP0  ;Bank 1
MOVLW   FFh         ;Set RB<7:4> as inputs
MOVWF   TRISB       ;
BCF     STATUS,RP0  ;Bank 0
```

> **注：** 必须对 ANSELH 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚读为 0。

4.4 PORTB 引脚的其他功能

器件系列上的 PORTB 引脚 RB<7:4> 都具有电平变化中断功能和弱上拉功能。以下三节将对这些 PORTB 引脚功能进行介绍。

4.4.1 弱上拉

每个 PORTB 引脚都具有单独配置的内部弱上拉功能。控制位 WPUB<7:4> 用于使能或禁止每个上拉功能（见寄存器 4-9）。当端口引脚被配置为输出时，这些弱上拉会自动关闭。上电复位时，选项寄存器的 RABPU 位将禁止所有上拉功能。

4.4.2 电平变化中断

4 个 PORTB 引脚被单独配置为具有电平变化中断功能的引脚。控制位 IOCB<7:4> 用于允许或禁止各引脚的中断功能。请参见寄存器 4-10。电平变化中断功能在上电复位时被禁止。

对于允许了电平变化中断功能的引脚，当前值将与上次读取的 PORTB 的旧锁存值相比较，以确定哪个位发生了变化或与旧值不匹配。所有不匹配的输出进行或运算，运算结果用来设置 INTCON 寄存器 （寄存器 2-3）中的 PORTB 电平变化中断标志位 （RABIF）。

该中断可将器件从休眠状态唤醒。用户在中断服务程序中可通过以下方式清除该中断：

a) 对 PORTB 进行任何读或写操作。这将结束不匹配条件。

b) 清零标志位 RABIF。

不匹配条件将继续把标志位 RABIF 置 1。读或写 PORTB 将结束不匹配条件并将标志位 RABIF 清零。保持上一次读取值的锁存器不受 MCLR 或欠压复位的影响。在这些复位之后，如果存在不匹配情况， RABIF 标志还将继续被置 1。

> **注：** 当读操作正在执行时发生了 I/O 引脚电平变化 （Q2 周期的起始时刻），则 RABIF 中断标志位可能不会被置 1。此外，由于端口上的读或写操作会影响该端口的所有位，所以在电平变化中断模式下使用多个引脚时必须特别小心。一个引脚上的变化可能会在另一个引脚上反映出来。

寄存器 4-7： PORTB：PORTB 寄存器

R/W-x	R/W-x	R/W-x	R/W-x	U-0	U-0	U-0	U-0
RB7	RB6	RB5	RB4	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **RB<7:4>**：PORTB I/O 引脚位
1 = 端口引脚大于 V_{IH}
0 = 端口引脚小于 V_{IL}

bit 3-0 **未实现**：读为 0

寄存器 4-8： TRISB：PORTB 三态寄存器

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **TRISB<7:4>**：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

bit 3-0 **未实现**：读为 0

寄存器 4-9： WPUB：弱上拉 PORTB 寄存器

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **WPUB<7:4>**：弱上拉寄存器位
1 = 上拉使能
0 = 上拉禁止

bit 3-0 **未实现**：读为 0

注 1： 必须使能选项寄存器的全局 $\overline{\text{RABPU}}$ 位以使能各个上拉。
2： 如果引脚处于输出模式，则自动禁止弱上拉器件 （TRISB<7:4> = 0）。

寄存器 4-10： IOCB：电平变化中断 PORTB 寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **IOCB<7:4>**：电平变化中断 PORTB 控制位
1 = 允许电平变化中断
0 = 禁止电平变化中断

bit 3-0 **未实现**：读为 0

4.4.3 引脚说明和引脚原理图

每个 PORTB 引脚都与其他功能复用。这里将简要说明引脚及其复用功能。各功能的具体信息（如 SSP、I^2C™ 或中断），请参见本数据手册中的相关章节。

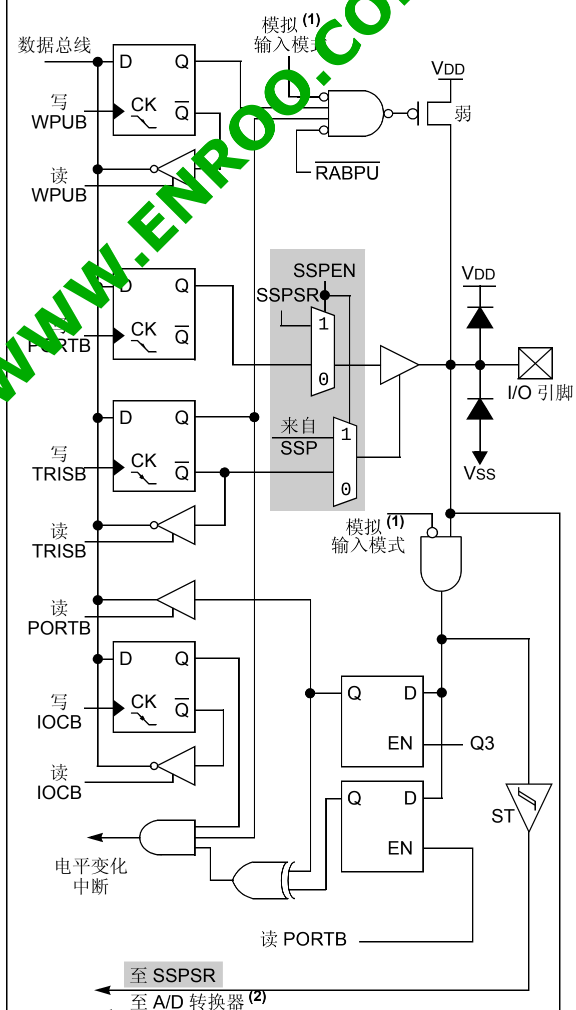
4.4.3.1 RB4/AN10/SDI/SDA

图 4-7 给出了此引脚的原理图。RB4/AN10/SDI/SDA[1] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- SPI 数据 I/O
- I^2C 数据 I/O

注 1：	SDI 和 SDA 仅在 PIC16F677/PIC16F687/PIC16F689/PIC16F690 上可用。

图 4-7： RB4 框图



仅在 PIC16F677/PIC16F687/PIC16F689/PIC16F690 上可用。

注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

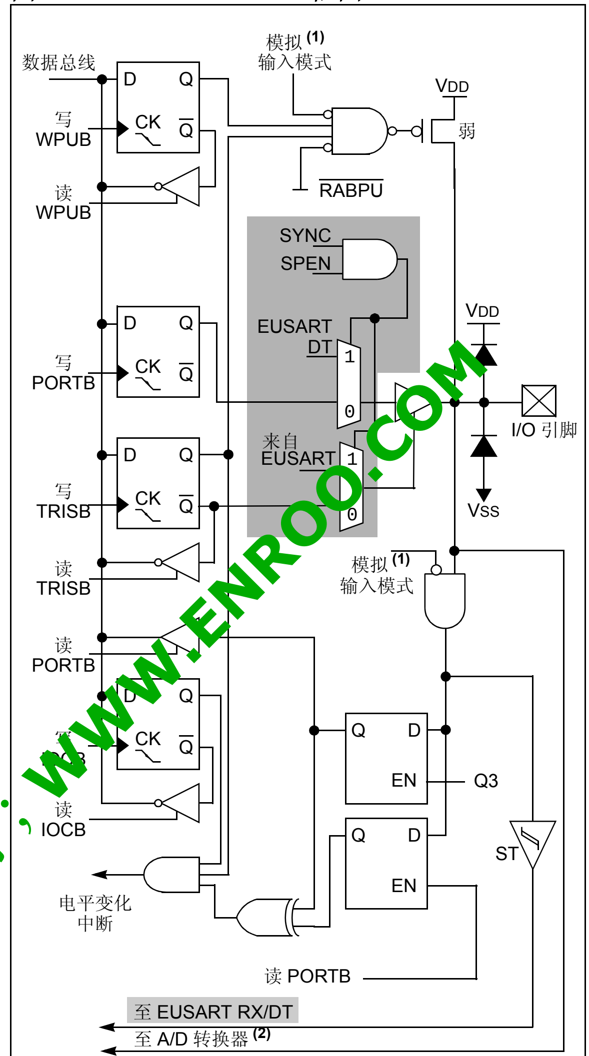
4.4.3.2 RB5/AN11/RX/DT

图 4-8 给出了此引脚的原理图。 RB5/AN11/RX/DT[1,2] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 异步串行输入
- 同步串行数据 I/O

注 1：	RX 和 DT 仅在 PIC16F687/PIC16F689/PIC16F690 上可用。
2：	AN11 在 PIC16F631 上未实现。

图 4-8： RB5 框图



仅在 PIC16F687/PIC16F689/PIC16F690 上可用。

注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

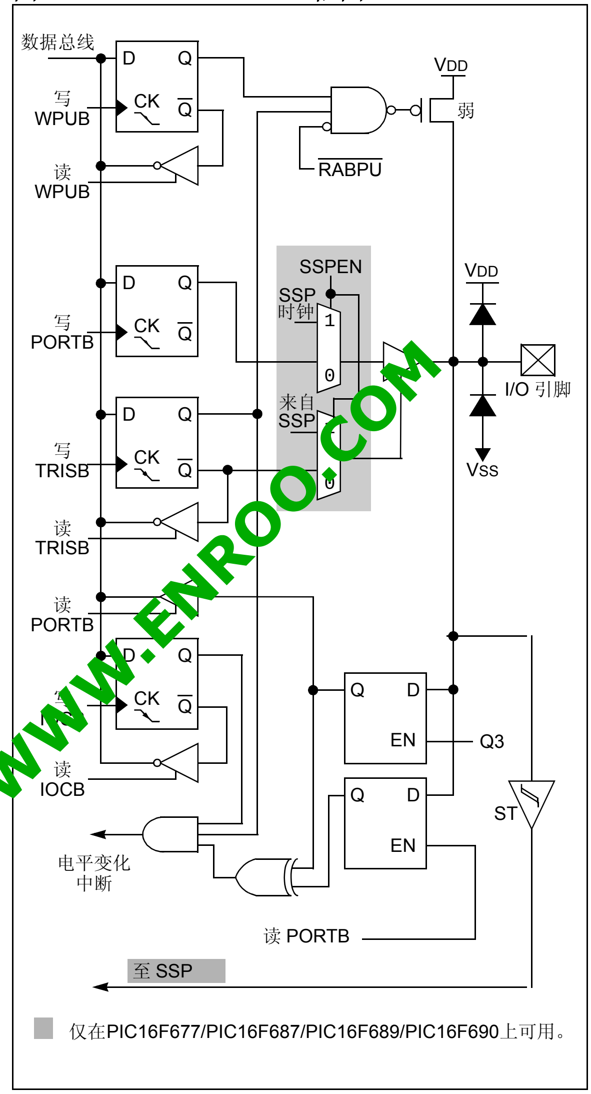
4.4.3.3 RB6/SCK/SCL

图 4-9 给出了此引脚的原理图。 RB6/SCK/SCL[(1)] 引脚可配置为下列功能之一：

- 通用 I/O
- SPI 时钟
- I^2C™ 时钟

注	**1：** SCK 和 SCL 仅在 PIC16F677/PIC16F687/PIC16F689/PIC16F690 上可用。

图 4-9： RB6 框图



4.4.3.4 RB7/TX/CK

图 4-10 给出了此引脚的原理图。RB7/TX/CK(1) 引脚可配置为下列功能之一：

- 通用 I/O
- 异步串行输出
- 同步时钟 I/O

> 注 1：TX 和 CK 仅在 PIC16F687/PIC16F689/PIC16F690 上可用。

图 4-10： RB7 框图

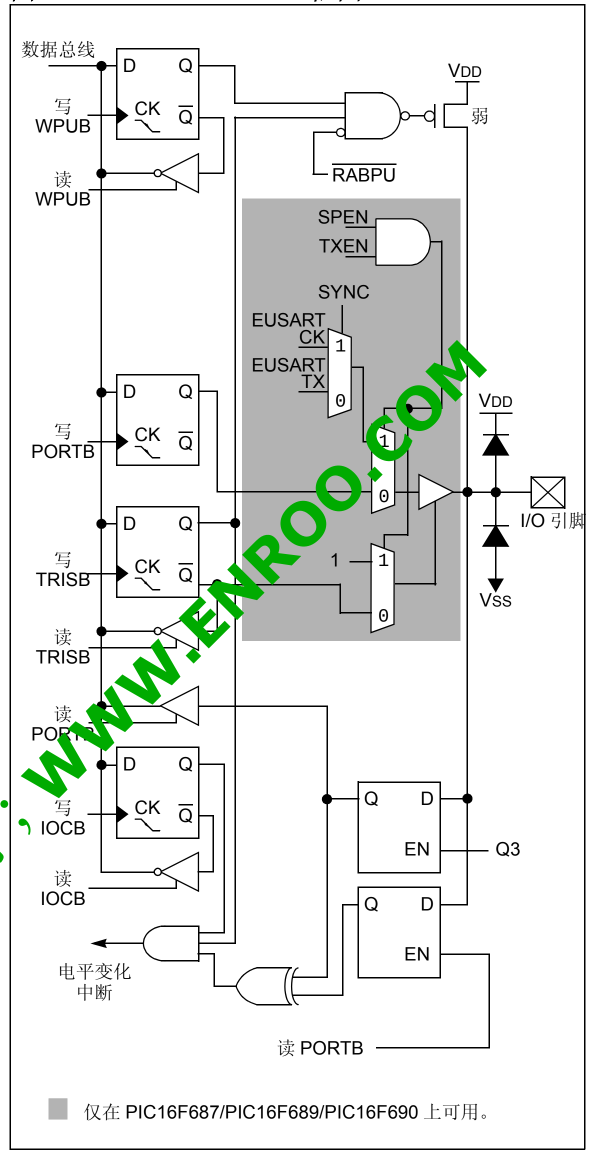


表 4-2： 与 PORTB 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
IOCB	IOCB7	IOCB6	IOCB5	IOCB4	—	—	—	—	0000 ----	0000 ----
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
WPUB	WPUB7	WPUB6	WPUB5	WPUB4	—	—	—	—	1111 ----	1111 ----

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

4.5 PORTC 和 TRISC 寄存器

PORTC 是一个 8 位宽的双向端口，对应的数据方向寄存器是 TRISC（寄存器 4-10）。将 TRISC 某位置 1（= 1）时，会将 PORTC 的相应引脚设为输入（即，使相应的输出驱动器呈高阻状态）。将 TRISC 某位清零 （= 0）时，会将 PORTC 的相应引脚设为输出（即，使能输出驱动器并将输出锁存器中的内容输出到选中引脚） 。例 4-4 显示了如何初始化 PORTC。读 PORTC 寄存器（寄存器 4-9）将读出相应引脚的状态，而对其进行写操作则是将数据写入端口锁存器。所有写操作都是“读 - 修改 - 写”操作。因此，对端口的写操作意味着总是先读端口引脚电平状态，然后修改这个值，最后再写入该端口的数据锁存器。

TRISC 寄存器控制 PORTC 引脚输出驱动器，即使它们被用作模拟输入。当引脚用于模拟输入时，用户应确保 TRISC 寄存器中的各位保持置 1。配置为模拟输入的 I/O 引脚总是读为 0。

> **注：** 必须对 ANSEL 和 ANSELH 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚将读为 0。

例 4-4： 初始化 PORTC

```
BCF     STATUS,RP0    ;Bank 0
BCF     STATUS,RP1    ;
CLRF    PORTC         ;Init PORTC
BSF     STATUS,RP1    ;Bank 2
CLRF    ANSEL         ;digital I/O
BSF     STATUS,RP0    ;Bank 1
BCF     STATUS,RP1    ;
MOVLW   0Ch           ;Set RC<3:2> as inputs
MOVWF   TRISC         ;and set RC<5:4,1:0>
                      ;as outputs
BCF     STATUS,RP0    ;Bank 0
```

寄存器 4-11： PORTC：PORTC 寄存器

R/W-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 **RC<7:0>**：PORTC 通用 I/O 引脚位
1 = 端口引脚大于 V_{IH}
0 = 端口引脚小于 V_{IL}

寄存器 4-12： TRISC：PORTC 三态寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R-1	R/W-1	R/W-1	R/W-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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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

4.5.1 RC0/AN4/C2IN+

RC0 引脚可配置为下列功能之一：

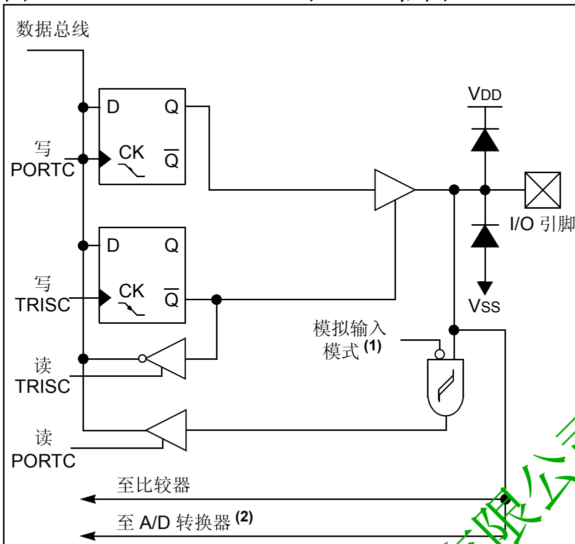
- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 连接至比较器 C2 的模拟输入

4.5.2 RC1/AN5/C12IN1-

RC1 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入
- 连接至比较器 C1 或 C2 的模拟输入

图 4-11： RC0 和 RC1 框图



注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

4.5.3 RC2/AN6/C12IN2-/P1D

RC2/AN6/P1D[(1)] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- PWM 输出
- 连接至比较器 C1 或 C2 的模拟输入

注	1： P1D 仅在 PIC16F685/PIC16F690 上可用。

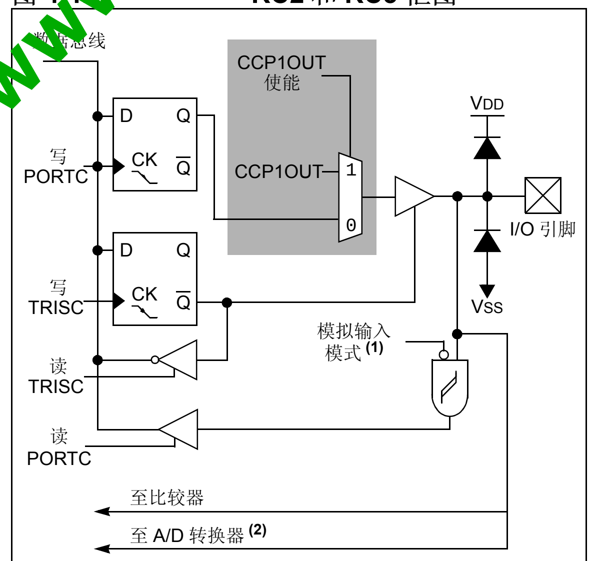
4.5.4 RC3/AN7/C12IN3-/P1C

RC3/AN7/P1C[(1)] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- PWM 输出
- PWM 输出
- 连接至比较器 C1 或 C2 的模拟输入

注	1： P1C 仅在 PIC16F685/PIC16F690 上可用。

图 4-12： RC2 和 RC3 框图



仅在 PIC16F685/PIC16F690 上可用。
注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

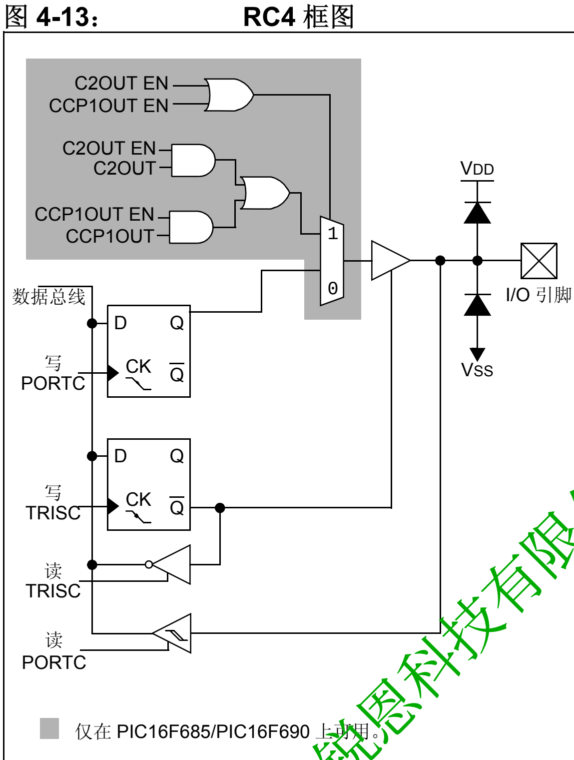
4.5.5 RC4/C2OUT/P1B

RC4/C2OUT/P1B[1, 2] 引脚可配置为下列功能之一：

- 通用 I/O
- 来自比较器 C2 的数字输出
- PWM 输出

> 注 1：同时使能 C2OUT 和 P1B 将在 RC4 上引起冲突并产生不可预知的结果。因此，如果 C2OUT 被使能，则 ECCP+ 不能用于半桥或全桥模式下，反之亦然。
>
> 2：P1B 仅在 PIC16F685/PIC16F690 上可用。

图 4-13： RC4 框图



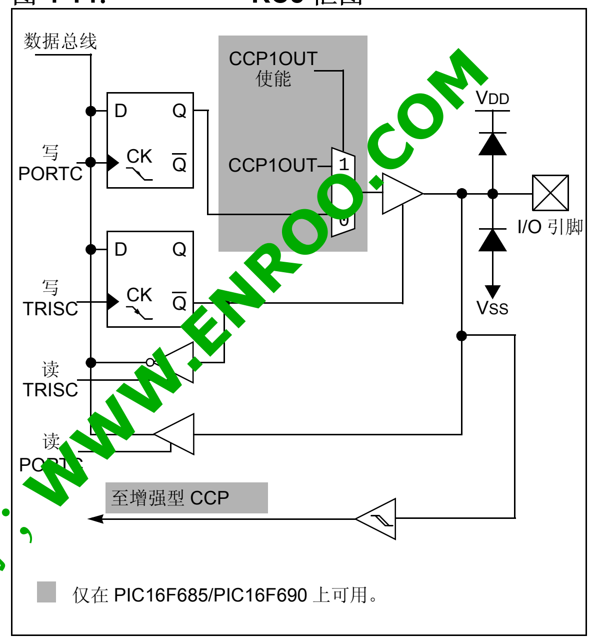
4.5.6 RC5/CCP1/P1A

RC5/CCP1/P1A[1] 引脚可配置为下列功能之一：

- 通用 I/O
- 增强型 CCP 的数字输入 / 输出
- PWM 输出

> 注 1：CCP1 和 P1A 仅在 PIC16F685/PIC16F690 上可用。

图 4-14： RC5 框图



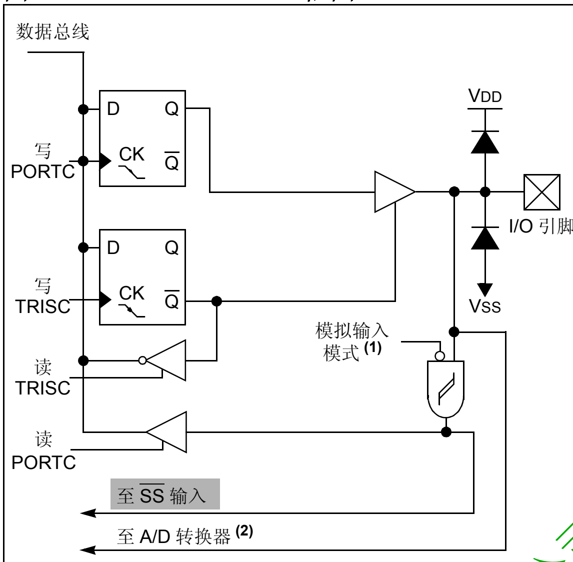
4.5.7 RC6/AN8/$\overline{SS}$

The RC6/AN8/$\overline{SS}$[1,2] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 从选择输入

注	**1：** $\overline{SS}$ 仅在 PIC16F687/PIC16F689/PIC16F690 上可用。
	2： AN8 在 PIC16F631 上未实现。

图 4-15： RC6 框图



仅在 PIC16F685/PIC16F690 上可用。

注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

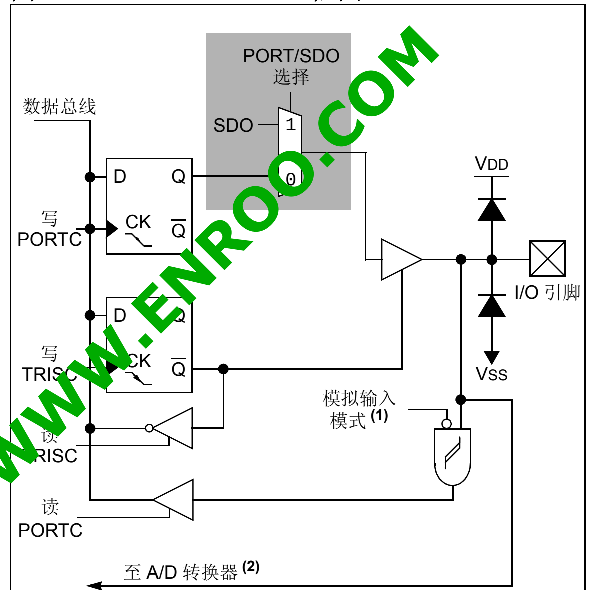
4.5.8 RC7/AN9/SDO

RC7/AN9/SDO[1,2] 引脚可配置为下列功能之一：

- 通用 I/O
- 连接至 ADC 的模拟输入 （PIC16F631 除外）
- 串行数据输出

注	**1：** SDO 仅在 PIC16F687/PIC16F689/PIC16F690 上可用。
	2： AN9 在 PIC16F631 上未实现。

图 4-16： RC7 框图



仅在 PIC16F685/PIC16F690 上可用。

注 1： ANSEL 决定模拟输入模式。
2： PIC16F631 上未实现。

表 4-3： 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	POR/BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSELH	—	—	—	—	ANS11	ANS10	ANS9	ANS8	---- 1111	---- 1111
CCP1CON(2)	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	00-- --10	00-- --10
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001
SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	—	0000 00--	0000 00--
SSPCON(1)	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
VRCON	C1VREN	C2VREN	VRR	VP6EN	VR3	VR2	VR1	VR0	0000 0000	0000 0000

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

注 1： 仅限 PIC16F687/PIC16F689/PIC16F690。

2： 仅限 PIC16F685/PIC16F690。

5.0 TIMER0 模块

Timer0 模块是 8 位定时器 / 计数器，具有以下特性：

- 8 位定时器 / 计数器寄存器 （TMR0）
- 8 位预分频器 （与看门狗定时器共用）
- 可编程内部或外部时钟源
- 可编程外部时钟边沿选择
- 溢出时产生中断

图 5-1 给出了 Timer0 模块的框图。

5.1 Timer0 工作原理

作为定时器使用时，Timer0 模块可用作 8 位定时器或 8 位计数器。

5.1.1 8 位定时器模式

作为定时器使用时，Timer0 模块将在每个指令周期递增（无预分频器）。将选项寄存器的 T0CS 位清零选择定时器模式。

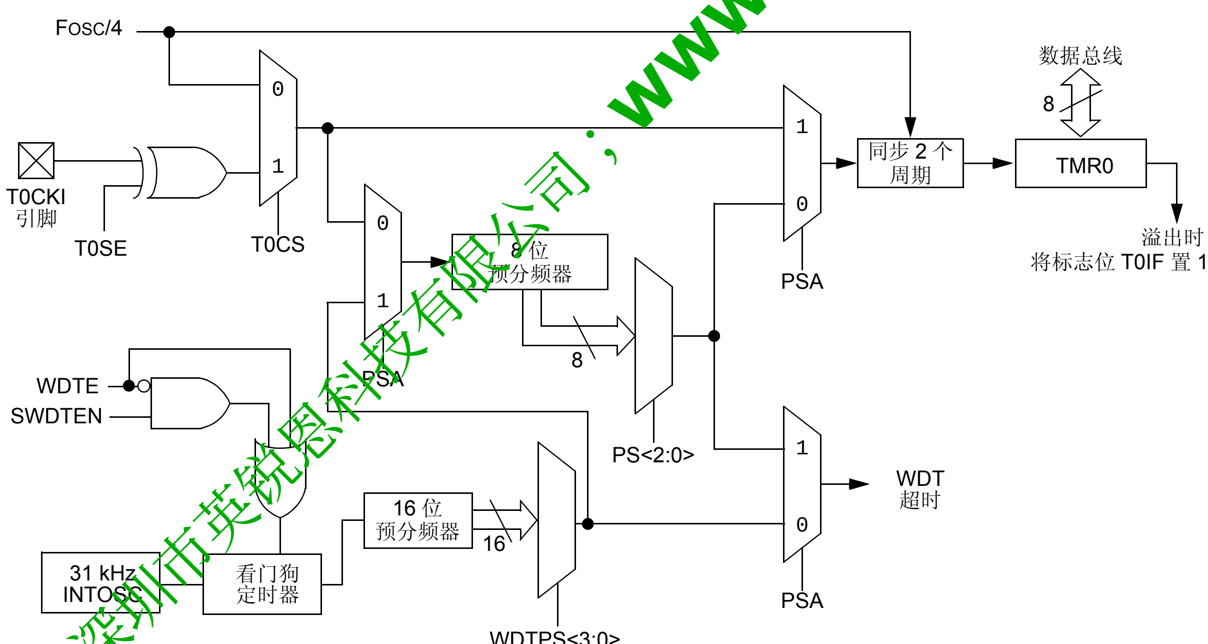
如果对 TMR0 执行写操作，则紧跟写操作之后的两个指令周期内 TMR0 禁止递增。

> **注：** 可以调整写入 TMR0 的值，以便在 TMR0 被写入时计入两个指令周期的延时。

5.1.2 8 位计数器模式

作为计数器使用时，Timer0 模块将在 T0CKI 引脚的每个上升或下降沿递增。递增边沿由选项寄存器的 T0SE 位决定。将选项寄存器的 T0CS 位设置为 1 选择计数器模式。

图 5-1： TIMER0/WDT 预分频器的框图



注 1： T0SE、 T0CS、 PSA 和 PS<2:0> 是选项寄存器中的位。
2： SWDTEN 和 WDTPS<3:0> 是 WDTCON 寄存器中的位。
3： WDTE 位在配置字寄存器中。

5.1.3 可软件编程的预分频器

有一个可软件编程的预分频器可供 Timer0 或看门狗定时器（WDT）使用，但不能同时使用。预分频器的分配由选项寄存器的 PSA 位控制。要将预分频器分配给 Timer0，必须将 PSA 位清零。

Timer0 模块有 8 个预分频比选项，范围从 1:2 到 1:256。预分频比可通过选项寄存器的 PS<2:0> 位进行选择。要使 Timer0 模块得到 1:1 的预分频比，必须将预分频器分配给 WDT 模块。

预分频器是不可读写的。当预分频器被分配给 Timer0 模块时，所有写入 TMR0 寄存器的指令都会将预分频器清零。

将预分频器被分配给 WDT 时， `CLRWDT` 指令会同时将预分频器和 WDT 清零。

5.1.3.1 在 Timer0 和 WDT 模块之间切换预分频器

由于预分频器可分配给 Timer0 或 WDT，因此在切换预分频比时可能会产生意外的器件复位。当将预分频器从 Timer0 分配给 WDT 模块时，必须执行如例 5-1 中所示的指令序列。

例 5-1： 改变预分频器的分配（**TIMER0 → WDT**）

```
BANKSEL  TMR0              ;
CLRWDT                     ;Clear WDT
CLRF     TMR0              ;Clear TMR0 and
                           ; prescaler
BANKSEL  OPTION_REG        ;
BSF      OPTION_REG,PSA    ;Select WDT
CLRWDT                     ;
                           ;
MOVLW    b'11111000'       ;Mask prescaler
ANDWF    OPTION_REG,W      ; bits
IORLW    b'00000101'       ;Set WDT prescaler
MOVWF    OPTION_REG        ; to 1:32
```

当将预分频器从 WDT 分配给 Timer0 模块时，必须执行以下指令序列（例 5-2）。

例 5-2： 改变预分频器的分配（**WDT → TIMER0**）

```
CLRWDT                    ;Clear WDT and
                          ;prescaler
BANKSEL OPTION_REG        ;
MOVLW   b'11110000'       ;Mask TMR0 select and
ANDWF   OPTION_REG,W      ; prescaler bits
IORLW   b'00000011'       ;Set prescale to 1:16
MOVWF   OPTION_REG        ;
```

5.1.4 TIMER0 中断

TMR0 寄存器从 FFh 溢出到 00h 时，Timer0 将产生中断。每次 TMR0 寄存器溢出时都会将 INTCON 寄存器的 T0IF 中断标志位置 1，与是否允许了 Timer0 中断无关。T0IF 位必须用软件清零。Timer0 中断允许位是 INTCON 寄存器的 T0IE 位。

注：	由于定时器在休眠状态下是冻结的，所以 Timer0 中断无法将处理器从休眠状态唤醒。

5.1.5 TIMER0 与外部时钟配合使用

当 Timer0 处于计数器模式下时，通过对内部相位时钟 Q2 和 Q4 周期的预分频输出进行采样，可实现 T0CKI 输入与 Timer0 寄存器的同步。因此，外部时钟源的高低电平周期必须满足第 **17.0** 节 **“电气规范”**中所示的时序要求。

寄存器 5-1： OPTION_REG：选项寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **$\overline{\text{RABPU}}$**：PORTA/PORTB 上拉使能位
1 = 禁止 PORTA/PORTB 上拉
0 = PORTA/PORTB 上拉由各 WPUAx 控制位使能

bit 6 **INTEDG**：中断边沿选择位
1 = 在 INT 引脚的上升沿触发中断
0 = 在 INT 引脚的下降沿触发中断

bit 5 **T0CS**：TMR0 时钟源选择位
1 = T0CKI 引脚上的电平跳变
0 = 内部指令周期时钟 （Fosc/4）

bit 4 **T0SE**：TMR0 时钟源边沿选择位
1 = 在 T0CKI 引脚信号从高至低跳变时，递增计数
0 = 在 T0CKI 引脚信号从低至高跳变时，递增计数

bit 3 **PSA**：预分频器分配位
1 = 预分频器分配给 WDT
0 = 预分频器分配给 Timer0 模块

bit 2-0 **PS<2:0>**：预分频比选择位

位值	TMR0 分频比	WDT 分频比
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

注 1： 有一个专用的 16 位 WDT 后分频器。更多信息，请参见**第 14.5 节 “看门狗定时器 （WDT）”**。

表 5-1： 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	POR/BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 0000	0000 0000
OPTION_REG	$\overline{\text{RABPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
TMR0	Timer0 模块寄存器								xxxx xxxx	uuuu uuuu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知。 Timer0 模块不使用阴影单元。

6.0 带门控的 TIMER1 模块

Timer1 模块是 16 位定时器 / 计数器，具有以下特性：

- 16 位定时器 / 计数器寄存器对 （TMR1H:TMR1L）
- 可编程内部或外部时钟源
- 3 位预分频器
- 可选的 LP 振荡器
- 同步或异步工作
- 通过比较器或 T1G 引脚的 Timer1 门控 （计数使能）
- 溢出时产生中断
- 溢出触发唤醒 （仅限外部时钟，异步模式）
- 捕捉 / 比较功能的时基 （仅限 PIC16F685/ PIC16F690）
- 特殊事件触发 （带 ECCP）（仅限 PIC16F685/ PIC16F690）
- 比较器输出与 Timer1 时钟同步

图 6-1 给出了 Timer1 模块的框图。

6.1 Timer1 工作原理

Timer1模块是16位递增计数器，可通过TMR1H:TMR1L 寄存器对访问。写 TMR1H 或 TMR1L 会直接更新计数器。

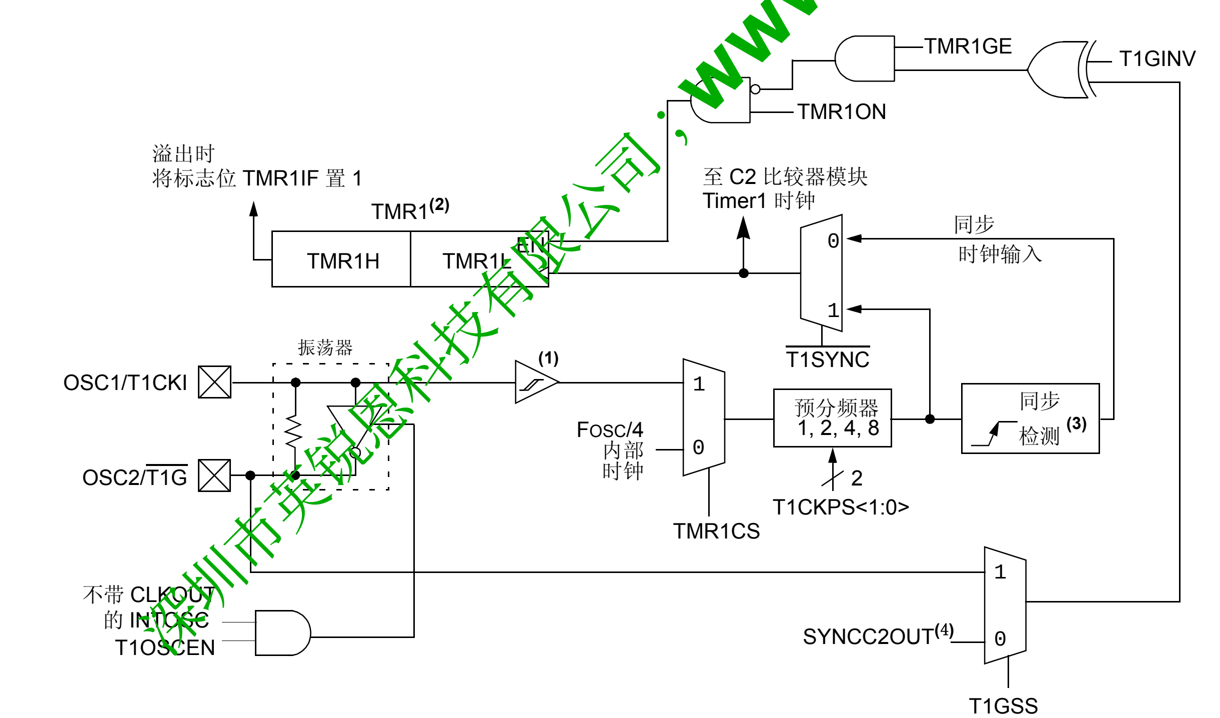
与内部时钟源配合使用时，该模块为定时器。与外部时钟源配合使用时，该模块可用作定时器或计数器。

6.2 时钟源选择

T1CON 寄存器的 TMR1CS 位用于选择时钟源。当 TMR1CS = 0 时，时钟源为 Fosc/4。当 TMR1CS = 1 时，时钟源由外部提供。

时钟源	T1OSCEN	FOSC 模式	TMR1CS
Fosc/4	x	xxx	0
T1CKI 引脚	0	xxx	1
T1LPOSC	1	LP 或 INTOSCIO	1

图 6-1： TIMER1 框图



注 1： ST 缓冲器在使用 LP 振荡器时为低功耗型，使用 T1CKI 时为高速型。
2： Timer1 寄存器在上升沿递增。
3： 处于休眠模式时同步不工作。
4： 当 CM2CON1 寄存器中的 C2SYNC 位置 1 时， SYNCC2OUT 同步。

6.2.1 内部时钟源

当选择内部时钟源时， TMR1H:TMR1L 寄存器对将在 Fosc 的整数倍 （取决于 Timer1 预分频器）处递增。

6.2.2 外部时钟源

当选择外部时钟源时，Timer1 模块可作为定时器或计数器工作。

计数时，Timer1 在外部时钟输入 T1CKI 的上升沿递增。此外，计数器模式时钟可以与单片机系统时钟同步，也可以异步工作。

如果需要使用外部时钟振荡器（并且单片机正在使用不带 CLKOUT 的 INTOSC），则 Timer1 可以将 LP 振荡器用作时钟源。

注：	在计数器模式下，发生以下任何一个或多个情况后，计数器在首个上升沿递增前，必须先经过一个下降沿： •POR 复位后使能 Timer1 •写入 TMR1H 或 TMR1L •Timer1 被禁止 •T1CKI 为高电平时 Timer1 被禁止 （TMR1ON=0），然后在 T1CKI 为低电平时 Timer1 被使能 （TMR1ON=1）。
注：	请参见图 6-2。

6.3 Timer1 预分频器

Timer1 有 4 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 位控制预分频计数器。对预分频计数器不能直接进行读写操作；但是，通过写入TMR1H或TMR1L可将预分频计数器清零。

6.4 Timer1 振荡器

在引脚 OSC1 （输入）和 OSC2 （放大器输出）之间接有一个内置低功耗 32.768 kHz 晶振。通过将 T1CON 寄存器的 T1OSCEN 控制位置 1 可使能该振荡器。在休眠期间，振荡器将继续工作。

Timer1 振荡器与系统 LP 振荡器共用。因此，只有当主系统时钟来自于内部振荡器或振荡器在 LP 模式下时，Timer1 才能采用该模式。用户必须提供软件延时以确保振荡器的正常起振。

当 Timer1 振荡器被使能时，TRISA5 和 TRISA4 位被置 1。RA5和RA4位读为0，而TRISA5和TRISA4位读为1。

注：	振荡器在使用之前需要一定的起振和稳定时间。因此，T1OSCEN 应置 1，且在使能 Timer1 之前确保有适当的延时。

6.5 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 $\overline{\text{T1SYNC}}$ 置 1，外部时钟输入将不同步。定时器异步于内部相位时钟进行递增计数。如果选择了外部时钟源，在休眠期间定时器将继续运行，并在溢出时产生中断以唤醒处理器。但是，用软件对定时器进行读 / 写操作时，要特别当心（见第 **6.5.1** 节 **“在异步计数器模式下读写 Timer1”**）。

注：	当从同步切换到异步操作时，可能会跳过一次递增。当从异步切换到同步操作时，可能会产生一次额外递增。

6.5.1 在异步计数器模式下读写 TIMER1

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（由硬件实现）。但是，应该注意的是，用两个 8 位值来读取 16 位定时器本身就会产生某些问题，这是因为定时器可能在两次读操作之间产生溢出。

对于写操作，建议用户直接停止计数器，然后写入需要的值。如果定时器寄存器正进行递增计数，对定时器寄存器进行写操作，可能会导致写入竞争，从而可能在 TMR1H:TMR1L 寄存器对中产生不可预测的值。

6.6 Timer1 门控

Timer1 门控源在使能时，允许 Timer1 在门控信号有效时进行计时。Timer1 门控源可用软件配置为 $\overline{\text{T1G}}$ 引脚或比较器 C2 的输出。这使得器件能够使用 $\overline{\text{T1G}}$ 直接对外部事件进行计时，或者使用比较器 C2 直接对模拟事件进行计时。Timer1 门控源的选择，请参见 CM2CON1 寄存器 （寄存器 8-3）。这个特性可以简化 Δ–Σ A/D 转换器和许多其他应用的程序。

注：	要将 $\overline{\text{T1G}}$ 或 C2OUT 用作 Timer1 门控源，必须将T1CON寄存器的TMR1GE位置1。选择 Timer1 门控源的更多信息，请参见 CM2CON1 寄存器 （寄存器 8-3）。

可使用 T1CON 寄存器的 T1GINV 位将 Timer1 门控反相，无论门控源来自 $\overline{\text{T1G}}$ 引脚还是比较器 C2 的输出。这样可将 Timer1 配置为测量事件之间的有效高电平或有效低电平时间。

6.7 Timer1 中断

Timer1 寄存器对 （TMR1H:TMR1L）递增到 FFFFh，然后返回到 0000h。当 Timer1 计满返回时，PIR1 寄存器的 Timer1 中断标志位将置 1。为允许计满返回时的中断，必须将以下位置 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 位清零将清除中断。

注：	在重新允许中断前，应将TMR1H:TTMR1L 寄存器对以及 TMR1IF 位清零。

6.8 Timer1 在休眠模式下的工作原理

只有在设置为异步计数器模式时，Timer1 才能在休眠模式下工作。在该模式下，可使用外部晶振或时钟源信号使计数器递增计数。要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 $\overline{\text{T1SYNC}}$ 位置 1
- 必须将 T1CON 寄存器的 TMR1CS 位置 1
- T1CON 寄存器的 T1OSCEN 位（可以被置 1）

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 （0004h）。

6.9 ECCP 捕捉 / 比较时基

当工作在捕捉或比较模式下时， ECCP 模块使用 TMR1H:TMR1L 寄存器对作为时基。

在捕捉模式下，当发生配置的事件时， 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寄存器对中的值相匹配时触发事件。该事件可以是特殊事件触发信号。

更多信息，请参见**第 11.0 节 “增强型捕捉 / 比较 /PWM 模块”**。

6.10 ECCP 特殊事件触发器

当将 ECCP 配置为触发特殊事件时，触发信号将清零 TMR1H:TMR1L寄存器对。该特殊事件不会引起Timer1 中断。 ECCP 模块仍可配置为产生 ECCP 中断。

在此工作模式下， CCPR1H:CCPR1L 寄存器对成为了 Timer1 的周期寄存器。

要利用特殊事件触发信号，应使 Timer1 与 Fosc 同步。Timer1 的异步操作会导致错过特殊事件触发信号。

如果对 TMR1H 或 TMR1L 的写操作和来自 ECCP 的特殊事件触发信号同时发生，则写操作优先。

更多信息，请参见**第 11.2.4 节 “特殊事件触发器”**。

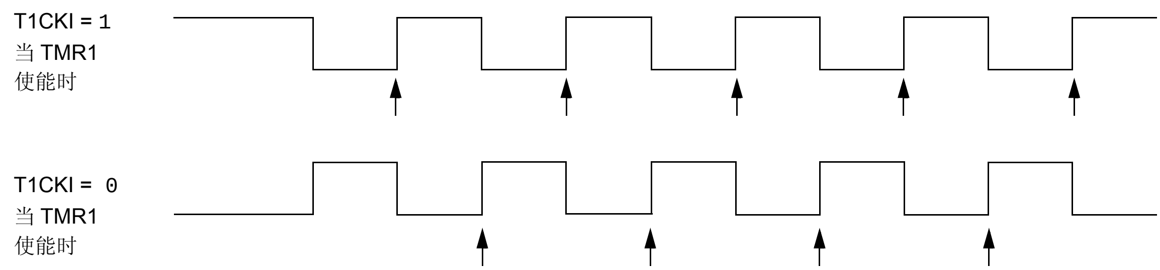
6.11 比较器同步

用于递增Timer1的时钟也可以用于同步比较器输出。比较器模块中使能了此功能。

当使用比较器作为 Timer1 门控时，应将比较器输出与 Timer1 保持同步。这可以确保 Timer1 在比较器发生变化时不会错过递增。

更多信息，请参见**第 8.8.2 节 “使比较器 C2 的输出与 Timer1 同步”**。

图 6-2： TIMER1 递增边沿



注 1： 箭头表明计数器递增。
2： 在计数器模式下，在时钟的首个递增上升沿之前必须先经过一个下降沿。

6.12 Timer1 控制寄存器

Timer1 控制寄存器（T1CON）用于控制 Timer1 以及选择 Timer1 模块的各种特性，如寄存器 6-1 中所示。

寄存器 6-1： T1CON：TIMER1 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1GINV[1]	TMR1GE[2]	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **T1GINV：** Timer1 门控反相位 [1]
1 = Timer1 门控高电平有效（当 Timer1 门控信号为高电平时 Timer1 计数）
0 = Timer1 门控低电平有效（当 Timer1 门控信号为低电平时 Timer1 计数）

bit 6 **TMR1GE：** Timer1 门控使能位 [2]
<u>如果 TMR1ON = 0：</u>
该位为无关位
<u>如果 TMR1ON = 1：</u>
1 = Timer1 计数与否，由 Timer1 门控功能控制
0 = Timer1 始终计数

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 **T1OSCEN：** LP 振荡器使能控制位
<u>如果不带 CLKOUT 的 INTOSC 处于有效状态：</u>
1 = LP 振荡器使能作为 Timer1 的时钟
0 = LP 振荡器关闭
<u>否则：</u>
该位为无关位

bit 2 **$\overline{\text{T1SYNC}}$：** Timer1 外部时钟输入同步控制位
<u>TMR1CS = 1：</u>
1 = 不同步外部时钟输入
0 = 同步外部时钟输入
<u>TMR1CS = 0：</u>
该位为无关位。 Timer1 使用内部时钟

bit 1 **TMR1CS：** Timer1 时钟源选择位
1 = 使用来自 T1CKI 引脚（在上升沿）的外部时钟
0 = 内部时钟（Fosc/4）

bit 0 **TMR1ON：** Timer1 使能位
1 = 使能 Timer1
0 = 停止 Timer1

注 1： T1GINV 位将 Timer1 门控逻辑反相，无论其来源为何。
2： 要将 $\overline{\text{T1G}}$ 引脚或 C2OUT（通过 CM2CON1 寄存器的 T1GSS 位在二者之间进行选择）用作 Timer1 门控源，必须将 TMR1GE 位置 1。

表 **6-1：** 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	POR/BOR 时的值	所有其他复位时的值
CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	---- --10	---- --10
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 0000	0000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。 Timer1 模块不使用阴影单元。

7.0 TIMER2 模块

Timer2 模块是 8 位定时器，具有以下特性：

- 8 位定时器寄存器 （TMR2）
- 8 位周期寄存器 （PR2）
- TMR2 与 PR2 匹配时产生中断
- 可软件编程的预分频器 （分频比为 1:1、 1:4 和 1:16）
- 可软件编程的后分频器 （分频比为 1:1 到 1:16）

Timer2 框图请参见图 7-1。

7.1 Timer2 工作原理

Timer2 模块的时钟输入是系统指令时钟（Fosc/4）。时钟送入 Timer2 预分频器， Timer2 预分频器提供 1:1、1:4 或 1:16 三种预分频比选择。然后预分频器的输出用于递增 TMR2 寄存器。

TMR2 的值与 PR2 的值不断进行比较以确定何时匹配。TMR2 将从 00h 开始递增，直到与 PR2 中的值匹配。当匹配发生时，会发生以下两件事：

- TMR2 在下一个递增周期复位为 00h。
- Timer2 后分频器递增

Timer2/PR2 比较器的匹配输出被送入 Timer2 后分频器。后分频器具有 1:1 到 1:16 的后分频比选择。Timer2 后分频器的输出用于将 PIR1 寄存器中的 TMR2IF 中断标志位置 1。

TMR2 和 PR2 寄存器均可完全读写。在任何复位时，TMR2寄存器被设置为00h，PR2寄存器被设置为FFh。

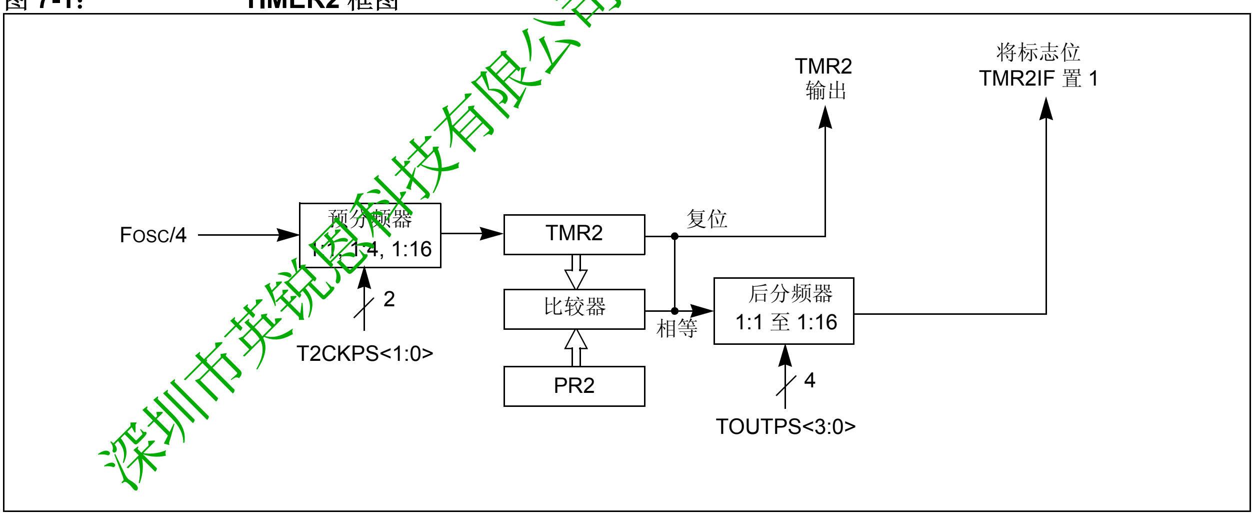
通过将 T2CON 寄存器中的 TMR2ON 位设置为 1 可使能Timer2。通过将TMR2ON位设置为0可关闭Timer2。

Timer2 预分频器由 T2CON 寄存器中的 T2CKPS 位控制。 Timer2 后分频器由 T2CON 寄存器中的 TOUTPS 位控制。预分频和后分频计数器均会在发生以下事件时清零：

- 对 TMR2 进行写操作。
- 对 T2CON 进行写操作。
- 发生任何器件复位 （上电复位、 $\overline{\text{MCLR}}$ 复位、看门狗定时器复位或欠压复位）。

注：	写 T2CON 时 TMR2 不会清零。

图 7-1： TIMER2 框图



寄存器 7-1： T2CON：TIMER2 控制寄存器 [1]

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **未实现：**读为 0

bit 6-3 **TOUTPS<3:0>：**Timer2 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 **TMR2ON：**Timer2 使能位
1 = 使能 Timer2
0 = 关闭 Timer2

bit 1-0 **T2CKPS<1:0>：**Timer2 时钟预分频比选择位
00 = 预分频比为 1
01 = 预分频比为 4
1x = 预分频比为 16

注 1： 仅限 PIC16F685/PIC16F690。

表 7-1： 与 TIMER2[1] 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	POR/BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PR2	Timer2 模块周期寄存器								1111 1111	1111 1111
TMR2	8 位 TMR2 寄存器的保持寄存器								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。 Timer2 模块不使用阴影单元。
注 1： 仅限 PIC16F685/PIC16F690。

8.0 比较器模块

比较器模块通过比较两个模拟电压并提供其相对幅值的数字表示，用于建立模拟电路与数字电路的接口。比较器是非常有用的混合信号构建模块，因为它提供了与程序执行相独立的模拟功能。模拟比较器模块包含以下特性：

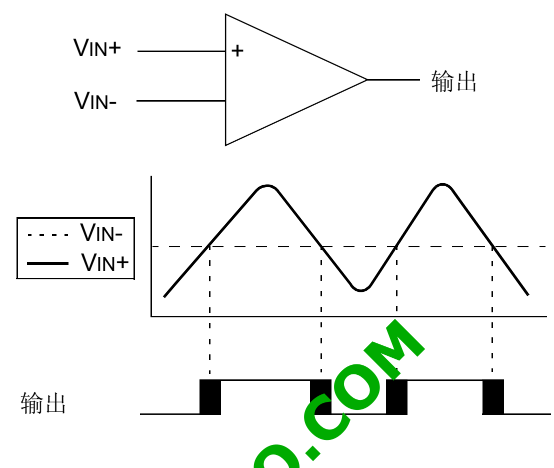
- 独立的比较器控制
- 可编程输入选择
- 有内部 / 外部比较器输出
- 可编程输出极性
- 电平变化中断
- 从休眠状态唤醒
- PWM 关闭
- Timer1 门控 （计数使能）
- 输出与 Timer1 时钟输入同步
- SR 锁存器
- 可编程和固定参考电压

注：	只有比较器 C2 可连接到 Timer1。

8.1 比较器概述

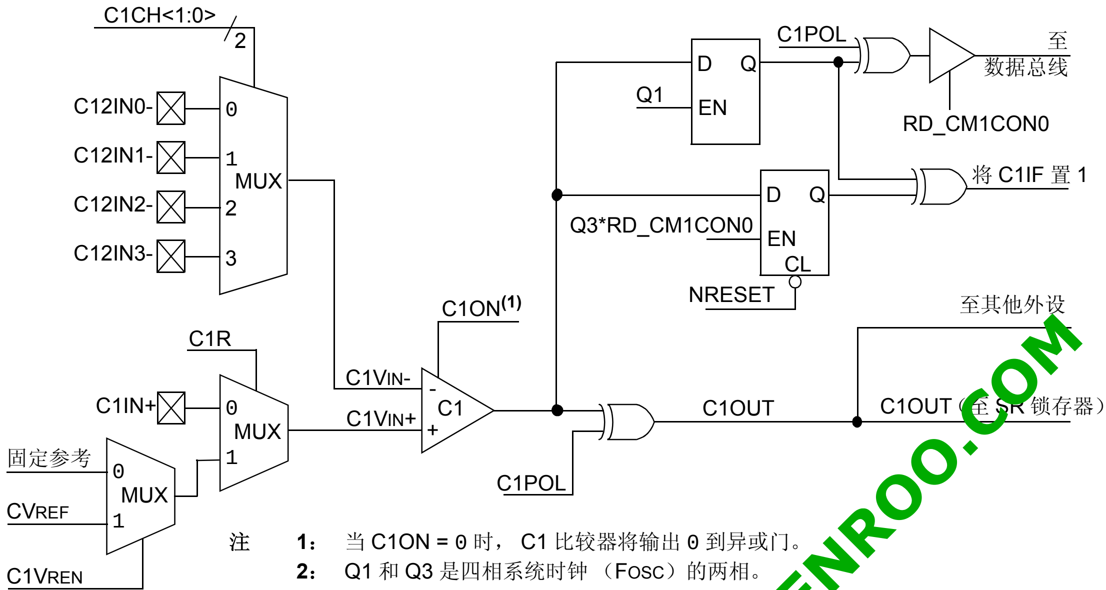
图 8-1 所示为单比较器以及模拟输入电平与数字输出之间的关系。当 VIN+ 上的模拟电压小于 VIN- 上的模拟电压值时，比较器输出为数字低电平。当 VIN+ 上的模拟电压大于 VIN- 上的模拟电压值时，比较器输出为数字高电平。

图 8-1： 单比较器



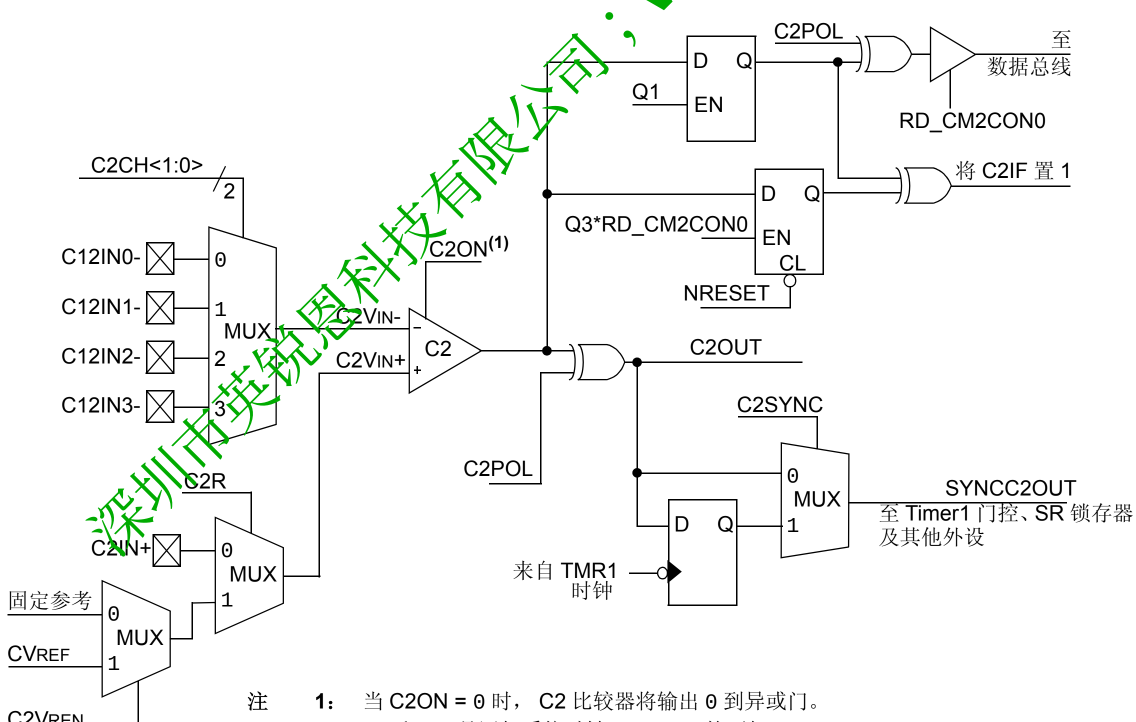
注： 比较器输出中的黑色区域表示因输入失调电压和响应时间所造成的输出不确定区域

图 8-2：　比较器 C1 的简化框图



注　**1：** 当 C1ON = 0 时， C1 比较器将输出 0 到异或门。
2： Q1 和 Q3 是四相系统时钟 （FOSC）的两相。
3： Q1 在休眠模式下保持为高电平。

图 8-3：　比较器 C2 的简化框图



注　**1：** 当 C2ON = 0 时， C2 比较器将输出 0 到异或门。
2： Q1 和 Q3 是四相系统时钟 （FOSC）的两相。
3： Q1 在休眠模式下保持为高电平。

8.2 比较器控制

每个比较器都有独立的控制和配置寄存器：比较器 C1 为 CM1CON0，比较器 C2 为 CM2CON0。此外，比较器 C2 还有另外一个控制寄存器 CM2CON1，用于控制与 Timer1 的交互和两个比较器输出的同时读取。

CM1CON0 和 CM2CON0 寄存器（分别见寄存器 8-1 和 8-2）包含以下控制和状态位：

- 使能
- 输入选择
- 参考电压选择
- 输出选择
- 输出极性

8.2.1 比较器使能

将 CMxCON0 寄存器的 CxON 位置 1 可以使能比较器操作。将 CxON 位清零可以禁止比较器，以使电流消耗降至最低。

8.2.2 比较器输入选择

CMxCON0 寄存器的 CxCH<1:0> 位指示将 4 个模拟输入引脚之一连接到比较器的反相输入端。

> **注：** 要将CxIN+和C12INx-引脚用作模拟输入，必须将 ANSEL 寄存器中的相应位置 1，同时也必须置 1 相应的 TRIS 位来禁止输出驱动器。

8.2.3 比较器参考电压选择

通过将 CMxCON0 寄存器的 CxR 位置 1，将内部参考电压或模拟输入引脚连接到比较器的不反相输入。关于内部参考电压模块的更多信息，请参见**第 8.9 节“比较器 SR 锁存器”**。

8.2.4 比较器输出选择

可以通过读 CMxCON0 寄存器的 CxOUT 位或 CM2CON1 寄存器的 MCxOUT 位监视比较器的输出。为了使输出可用于外部连接，必须满足以下条件：

- 必须将 CMxCON0 寄存器的 CxOE 位置 1
- 必须清零相应的 TRIS 位
- 必须将 CMxCON0 寄存器的 CxON 位置 1

> **注 1：** CxOE 位改写端口数据锁存器。将 CxON 置 1 对端口改写没有影响。
>
> **2：** 比较器的内部输出在每个指令周期被锁存。除非另外指定，否则不锁存外部输出。

8.2.5 比较器输出极性

反相比较器的输出在功能上等同于交换比较器输入。可以通过将 CMxCON0 寄存器的 CxPOL 位置 1 来使比较器输出的极性反相。清零 CxPOL 位导致不反相输出。

表 8-1 给出了输出状态与输入条件的关系（包括极性控制）。

表 8-1： 比较器输出状态与输入条件

输入条件	CxPOL	CxOUT
CxVIN- > CxVIN+	0	0
CxVIN- < CxVIN+	0	1
CxVIN- > CxVIN+	1	1
CxVIN- < CxVIN+	1	0

8.3 比较器响应时间

在改变输入源或选择新的参考电压后，比较器输出要经过一段不确定状态的时间。这段时间被称为响应时间。比较器的响应时间不同于参考电压的稳定时间。因此，在确定比较器输入改变的总响应时间时，必须考虑这两个时间。更多详细信息，请参见**第 17.0 节“电气规范”**中的比较器和参考电压规范。

8.4 比较器中断操作

只要比较器的输出值发生变化，相应的比较器中断标志位就会置 1。比较器输出值的变化由不匹配电路识别，该电路由两个锁存器和一个异或门组成（见图 8-2 和图 8-3）。当读 CMxCON0 寄存器时，一个锁存器更新为比较器的输出电平。该锁存器将保持该值，直到下次读 CMxCON0 寄存器，或者发生复位。不匹配电路的另一个锁存器在每个 Q1 系统时钟更新。当比较器的输出变化在 Q1 时钟周期输入到第二个锁存器时，即产生不匹配条件。这时两个不匹配锁存器具有相反的输出电平，由异或门检测并被送到中断电路。不匹配条件持续直到读 CMxCON0 寄存器或者比较器输出返回先前的状态。

> **注 1：** 对 CMxCON0 寄存器的写操作也将清除不匹配条件，因为所有写操作的写周期开始都包含一个读操作。
>
> **2：** 比较器中断的正常工作与 CxOE 的状态无关。

比较器中断是根据不匹配边沿而不是不匹配电平置 1 的。这意味着不需要额外的读取或写入 CMxCON0 寄存器来将不匹配寄存器清零这一步骤，即可将中断标志复位。当不匹配寄存器清零时，比较器输出返回前一个状态将发生中断，否则不会发生中断。

需要用软件保存比较器输出状态的信息（从 CMxCON0 寄存器或 CM2CON1 寄存器读取），以确定实际发生的变化。

PIR1 寄存器的 CxIF 位是比较器中断标志。必须用软件将该位清零以将其复位。由于可以对该寄存器写入 1，因此可产生中断。

必须将 PIE1 寄存器的 CxIE 位以及 INTCON 寄存器的 PEIE 和 GIE 位都置 1 以允许比较器中断。如果上述位中有任何一位被清零，则无法允许中断，尽管中断条件发生时仍会将 PIR1 寄存器的 CxIF 位置 1。

图 8-4：比较器中断时序（不带 CMxCON0 读取）

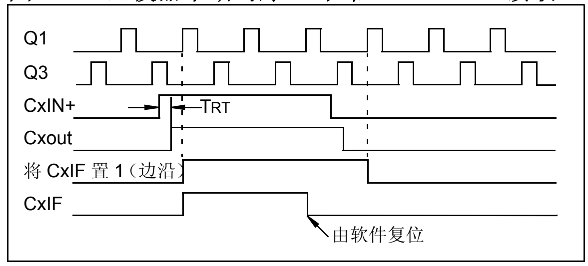
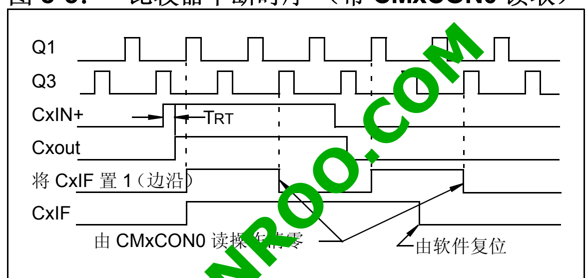


图 8-5： 比较器中断时序 （带 CMxCON0 读取）



> **注 1：** 在读操作执行过程中 （Q2 周期的起始时刻），如果 CMxCON0 寄存器（CxOUT）发生变化，那么 PIR1 寄存器的 CxIF 中断标志位可能不会被置 1。
>
> **2：** 当两个比较器之一先使能时，比较器模块中的偏置电路在稳定前可能产生比较器的无效输出。应允许偏置电路有 1 µs 的稳定时间，然后在允许比较器中断前将不匹配条件和中断标志清零。

8.5 休眠模式下的工作原理

如果在进入休眠模式之前比较器使能，则它在休眠期间仍将处于运行状态。**第 17.0 节“电气规范”**中单独给出了比较器消耗的额外电流。如果不使用比较器来唤醒器件，则在休眠模式下可通过关闭比较器使功耗降至最低。可以通过清零 CMxCON0 寄存器的 CxON 位来关闭每个比较器。

比较器输出变化可以将器件从休眠状态唤醒。要使比较器能将器件从休眠状态唤醒，必须将 PIE1 寄存器的 CxIE 位和 INTCON 寄存器的 PEIE 位置 1。紧跟 `Sleep` 指令后的指令总是在从休眠状态唤醒之后执行。如果也将 INTCON 寄存器的 GIE 位置 1，则器件将执行中断服务程序。

8.6 复位的影响

器件复位强制 CMxCON0 和 CM2CON1 寄存器为复位状态。这使比较器和参考电压被强制为 “关闭”状态。

寄存器 8-1： CM1CON0：比较器 C1 控制寄存器 0

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **C1ON**：比较器 C1 使能位
1 = 比较器 C1 使能
0 = 比较器 C1 禁止

bit 6 **C1OUT**：比较器 C1 输出位
如果 C1POL = 1 （极性反相）：
当 C1VIN+ > C1VIN- 时， C1OUT = 0
当 C1VIN+ < C1VIN- 时， C1OUT = 1
如果 C1POL = 0 （极性不反相）：
当 C1VIN+ > C1VIN- 时， C1OUT = 1
当 C1VIN+ < C1VIN- 时， C1OUT = 0

bit 5 **C1OE**：比较器 C1 输出使能位
1 = C1OUT 出现在 C1OUT 引脚上 (1)
0 = C1OUT 仅在内部有效

bit 4 **C1POL**：比较器 C1 输出极性选择位
1 = C1OUT 逻辑反相
0 = C1OUT 逻辑不反相

bit 3 **未实现**：读为 0

bit 2 **C1R**：比较器 C1 参考电压选择位（不反相输入）
1 = C1VIN+ 连接至 C1VREF 输出
0 = C1VIN+ 连接至 C1IN+ 引脚

bit 1-0 **C1CH<1:0>**：比较器 C1 通道选择位
00 = C1 的 C1VIN- 连接至 C12IN0- 引脚
01 = C1 的 C1VIN- 连接至 C12IN1- 引脚
10 = C1 的 C1VIN- 连接至 C12IN2- 引脚
11 = C1 的 C1VIN- 连接至 C12IN3- 引脚

注 1： 比较器输出需要以下 3 个条件：C1OE = 1， C1ON = 1 且相应的端口 TRIS 位 = 0。

寄存器 8-2： CM2CON0：比较器 C2 控制寄存器 0

R/W-0	R-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **C2ON：**比较器 C2 使能位
1 = 比较器 C2 使能
0 = 比较器 C2 禁止

bit 6 **C2OUT：**比较器 C2 输出位
如果 C2POL = 1 （极性反相）：
当 C2VIN+ > C2VIN- 时， C2OUT = 0
当 C2VIN+ < C2VIN- 时， C2OUT = 1
如果 C2POL = 0 （极性不反相）：
当 C2VIN+ > C2VIN- 时， C2OUT = 1
当 C2VIN+ < C2VIN- 时， C2OUT = 0

bit 5 **C2OE：**比较器 C2 输出使能位
1 = C2OUT 出现在 C2OUT 引脚上 (1)
0 = C2OUT 仅在内部有效

bit 4 **C2POL：**比较器 C2 输出极性选择位
1 = C2OUT 逻辑反相
0 = C2OUT 逻辑不反相

bit 3 **未实现：**读为 0

bit 2 **C2R：**比较器 C2 参考电压选择位 （不反相输入）
1 = C2VIN+ 连接至 C2VREF
0 = C2VIN+ 连接至 C2IN+ 引脚

bit 1-0 **C2CH<1:0>：**比较器 C2 通道选择位
00 = C2 的 C2VIN- 连接至 C12IN0- 引脚
01 = C2 的 C2VIN- 连接至 C12IN1- 引脚
10 = C2 的 C2VIN- 连接至 C12IN2- 引脚
11 = C2 的 C2VIN- 连接至 C12IN3- 引脚

注 1： 比较器输出需要以下 3 个条件：C2OE = 1， C2ON = 1 且相应的端口 TRIS 位 = 0。

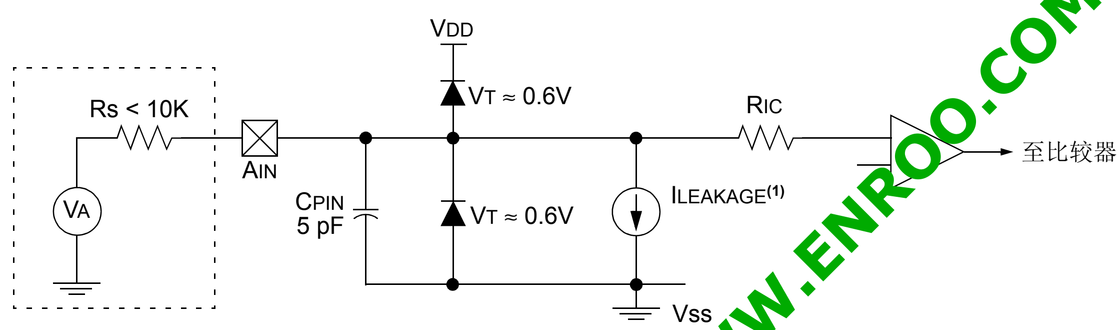
8.7 模拟输入连接注意事项

模拟输入的简化电路如图 8-6 所示。由于模拟输入引脚与数字输入共用连接，它们在 VDD 和 VSS 之间连有反向偏置的 ESD 保护二极管。因此，模拟输入必须在 VSS 和 VDD 之间。如果输入电压与这一范围偏离的绝对值超过 0.6V，就可能发生一个二极管正向导通，从而可能导致锁死发生。

模拟信号源的最大阻抗推荐值为 10 kΩ。任何连接到模拟输入引脚的外部元件 （如电容或齐纳二极管） ，应保证其泄漏电流极小以使引入的误差降至最低。

注	
1：	读端口寄存器时，所有配置为模拟输入的引脚均读为 0。配置为数字输入的引脚将根据输入规范转换为模拟输入。
2：	定义为数字输入引脚上的模拟电平可能会使输入缓冲器的电流消耗超过规定值。

图 8-6： 模拟输入模型



图注：
- CPIN = 输入电容
- ILEAKAGE = 由各连接点在引脚上产生的泄漏电流
- RIC = 片内走线等效电阻
- RS = 信号源阻抗
- VA = 模拟电压
- VT = 门限电压

注 1： 请参见第 17.0 节 “电气规范”。

8.8 其他比较器特性

共有 3 个额外的比较器特性：

- Timer1 计数使能 （门控）
- 使输出与 Timer1 同步
- 比较器输出的同时读取

8.8.1 由比较器 C2 提供 TIMER1 的门控源

这个特性可用来对模拟事件的持续时间或时间间隔计时。清零 CM2CON1 寄存器的 T1GSS 位将使 Timer1 根据比较器 C2 的输出递增。这需要已开启 Timer1 且已使能门控。详情请参见**第 6.0 节 “带门控的 Timer1 模块”**。

当比较器用作 Timer1 门控源时，建议通过将 C2SYNC 位置 1 使比较器与 Timer1 同步。这可以确保当比较器输出在递增期间改变时， Timer1 不会错过递增。

8.8.2 使比较器 C2 的输出与 TIMER1 同步

通过将 CM2CON1 寄存器的 C2SYNC 位置 1，可以使比较器 C2 的输出与 Timer1 保持同步。使能 C2 的输出时， C2 的输出在 Timer1 时钟源的下降沿被锁存。如果 Timer1 使用了预分频器，则比较器的输出在经过预分频后被锁存。为了防止发生竞争，比较器的输出在 Timer1 时钟源的下降沿被锁存，而 Timer1 在其时钟源的上升沿递增。更多信息，请参见比较器框图（图 8-3）和 Timer1 框图 （图 6-1）。

8.8.3 比较器输出的同时读取

CM2CON1 寄存器的 MC1OUT 和 MC2OUT 位是两个比较器输出的镜像副本。从同一个寄存器中同时读取两个输出的功能，可以消除从不同寄存器读取时存在的时序错位。

注	1： 通过读 CM2CON1 获取 C1OUT 或 C2OUT 的状态并不影响比较器中断不匹配寄存器。

寄存器 8-3： CM2CON1：比较器 C2 控制寄存器 1

R-0	R-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0
MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7	**MC1OUT**：C1OUT 的镜像副本位
bit 6	**MC2OUT**：C2OUT 的镜像副本位
bit 5-2	**未实现**：读为 0
bit 1	**T1GSS**：Timer1 门控源选择位 (1) 1 = Timer1 门控源为 $\overline{\text{T1G}}$ 0 = Timer1 门控源为 SYNCC2OUT
bit 0	**C2SYNC**：比较器 C2 输出同步位 (2) 1 = 输出与 Timer1 时钟的下降沿同步 0 = 输出异步

注 1：请参见**第 6.6 节 “Timer1 门控”**。
2：请参见图 8-3。

8.9 比较器 SR 锁存器

SR 锁存器模块为比较器输出提供了其他控制方法。该模块由一个 SR 锁存器和输出多路开关组成。 SR 锁存器可以由比较器输出置 1、复位或翻转。SR 锁存器也可以独立于比较器输出，通过 SRCON 控制寄存器中的控制位置 1 或复位。 SR 锁存器输出多路开关选择是将锁存器输出还是比较器输出送到 I/O 端口逻辑作为引脚的最终输出。

8.9.1 锁存器操作

锁存器是不依赖于时钟源的置 1- 复位锁存器。每个置 1 和复位输入均为高电平有效。每个锁存器输入都被连接到比较器输出和软件控制的脉冲发生器。可以通过 C1OUT 或 SRCON 寄存器的 PULSS 位将锁存器置 1。可以通过 C2OUT 或 SRCON 寄存器的 PULSR 位将锁存器复位。该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，则锁存器将进入复位状态。PULSS 和 PULSR 位都是自复位的，也就是说，对两个位中的任一个位执行一次写操作是完成锁存器置 1 或复位操作的必要条件。

8.9.2 锁存器输出

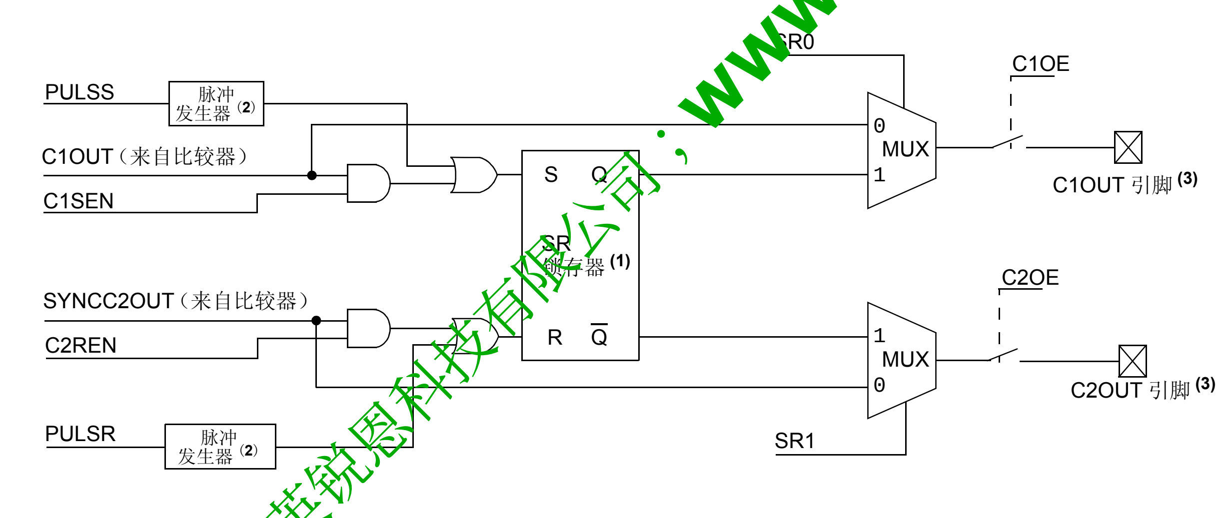
SRCON 寄存器的 SR<1:0> 位控制锁存器输出多路开关，并确定 4 种可能的输出配置。在这 4 种配置中，CxOUT I/O 端口逻辑连接到：

- C1OUT 和 C2OUT
- C1OUT 和 SR 锁存器 $\overline{Q}$
- C2OUT 和 SR 锁存器 Q
- SR 锁存器 Q 和 $\overline{Q}$

发生任何复位之后，默认的输出配置是未锁存的 C1OUT 和 C2OUT 模式。这维持了与不具有 SR 锁存器功能的器件的兼容性。

必须清零相应端口对应的 TRIS 位以使能端口引脚输出驱动器。此外，要使比较器或锁存器的输出出现在输出引脚上，必须将 CMxCON0 寄存器的 CxOE 比较器输出使能位置 1。锁存器配置使能状态与比较器的使能状态完全无关。

图 8-7： SR 锁存器的简化框图



注 1： 如果 R = 1 且 S = 1，则 Q = 0，$\overline{Q}$ = 1。
2： 脉冲发生器产生 1/2 Q 状态 （1 Tosc）的脉冲宽度。
3： 图中的输出仅供参考。更多详细信息，请参见 I/O 端口引脚框图。

寄存器 8-4： SRCON：SR 锁存器控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/S-0	R/S-0	U-0	U-0
SR1[(2)]	SR0[(2)]	C1SEN	C2REN	PULSS	PULSR	—	—
bit 7							bit 0

图注：	S = 只可置 1 位		
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **SR1**：SR 锁存器配置位[(2)]
1 = C2OUT 引脚为锁存器 $\overline{Q}$ 输出
0 = C2OUT 引脚为 C2 比较器输出

bit 6 **SR0**：SR 锁存器配置位[(2)]
1 = C1OUT 引脚为锁存器 Q 输出
0 = C1OUT 引脚为比较器 C1 输出

bit 5 **C1SEN**：C1 置 1 使能位
1 = C1 比较器输出将 SR 锁存器置 1
0 = C1 比较器输出对 SR 锁存器没有影响

bit 4 **C2REN**：C2 复位使能位
1 = C2 比较器输出将 SR 锁存器复位
0 = C2 比较器输出对 SR 锁存器没有影响

bit 3 **PULSS**：为 SR 锁存器提供置 1 输入脉冲位
1 = 触发脉冲发生器将 SR 锁存器置 1。该位立即由硬件复位。
0 = 不触发脉冲发生器

bit 2 **PULSR**：为 SR 锁存器提供复位输入脉冲位
1 = 触发脉冲发生器将 SR 锁存器复位。该位立即由硬件复位。
0 = 不触发脉冲发生器

bit 1-0 **未实现**：读为 0

注 1：无论 SR 锁存器的操作如何，CMxCON0 寄存器中的 CxOUT 位总是反映实际的比较器输出（不是引脚上的电平）。

2：要使 SR 锁存器输出出现在引脚上，相应的 CxOE 和 TRIS 位必须正确配置。

8.10 比较器参考电压

比较器参考电压模块为比较器提供了内部产生的参考电压。它具有以下特性：

- 与比较器操作独立
- 两个 16 级电压范围
- 输出钳位到 VSS
- 与 VDD 成比例
- 固定参考电压 （0.6）

VRCON 寄存器 （寄存器 8-5）控制参考电压模块，如图 8-8 所示。

8.10.1 独立操作

比较器参考电压独立于比较器配置。将 VRCON 寄存器的 VREN 位置 1 将使能参考电压。

8.10.2 输出电压选择

CVREF 参考电压有 2 个范围，每个范围有 16 个电平。范围的选择由 VRCON 寄存器的 VRR 位控制。16 个电平通过 VRCON 寄存器的 VR<3:0> 位进行设置。

CVREF 输出电压由下面的公式确定：

公式 8-1： CVREF 输出电压

VRR = 1（低电压范围）：

$$CV_{REF} = (VR<3:0>/24) \times V_{DD}$$

VRR = 0（高电压范围）：

$$CV_{REF} = (V_{DD}/4) + (VR<3:0> \times V_{DD}/32)$$

由于模块的构造所限，无法实现 VSS 到 VDD 的满量程。请参见图 8-8。

8.10.3 输出钳位到 VSS

可以将 CVREF 输出电压设为 VSS 且不产生功耗，方法是将 VRCON 寄存器的 VP6EN 位清零。

这使得比较器可以检测到过零点，且不消耗 CVREF 模块的电流。

8.10.4 输出与 VDD 成比例

比较器参考电压来自 VDD，因此， CVREF 输出会随着 VDD 的变化而变化。经过测试的比较器参考电压的绝对精度，请参见**第 17.0 节 “电气规范”**。

8.10.5 固定参考电压

固定参考电压独立于 VDD，输出电压标称值为 0.6V。可通过将 VRCON 寄存器的 VP6EN 位设为 1 使能该参考电压。当 HFINTOSC 振荡器有效时，该参考电压始终是使能的。

8.10.6 固定参考电压稳定周期

当固定参考电压模块使能时，需要一段时间使参考电压及其放大器电路达到稳定。用户程序必须包含允许模块稳定的延时子程序。关于最小延时要求，请参见电气规范章节。

8.10.7 参考电压选择

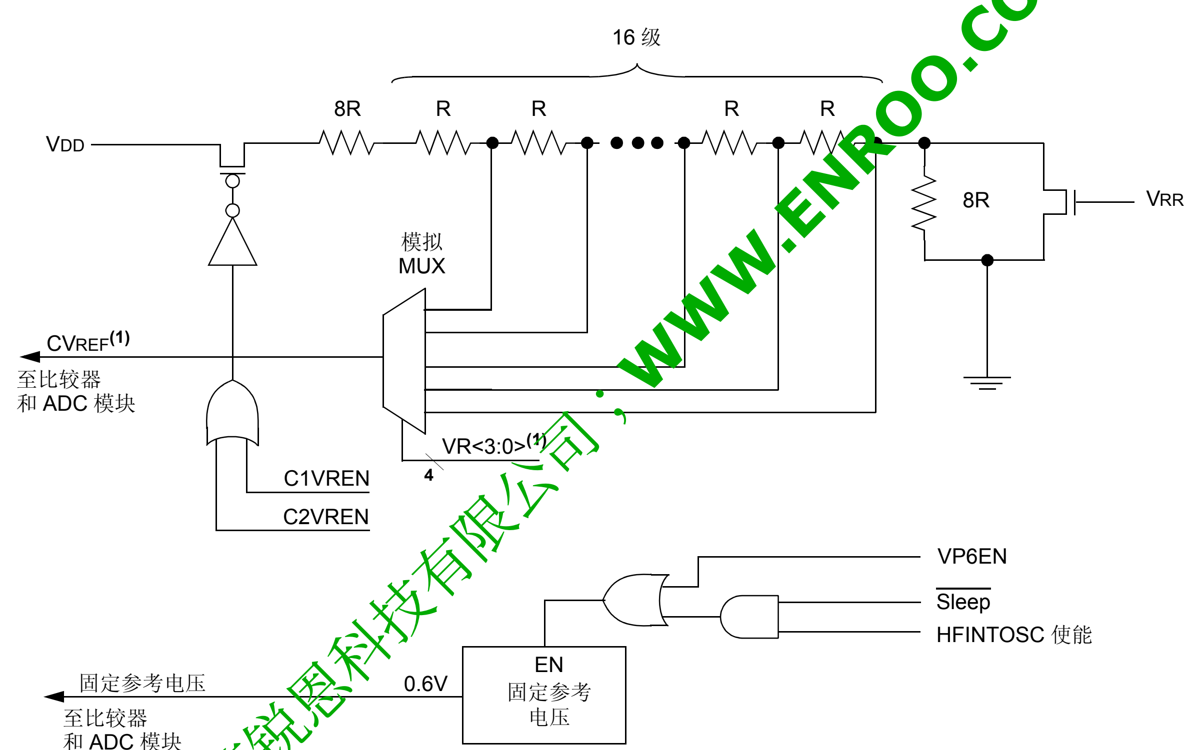
参考电压模块输出上的多路开关使比较器可以选择使用 CVREF 或固定参考电压。

将 VRCON 寄存器的 C1VREN 位置 1 使电流流入 CVREF 分压器，并使 C1 选择使用 CVREF 电压。清零 C1VREN 位使 C1 选择使用固定电压。

将 VRCON 寄存器的 C2VREN 位置 1 使电流流入 CVREF 分压器，并使 C2 选择使用 CVREF 电压。清零 C2VREN 位使 C2 选择使用固定电压。

当同时清零 C1VREN 和 C2VREN 位时，禁止电流流入 CVREF 分压器，从而将参考电压外设的功耗降至最低。

图 8-8： 比较器参考电压框图



注 1： 应注意确保 VREF 维持在比较器共模输入范围之内。更多详细信息，请参见第 **17.0 节 "电气规范"**。

寄存器 8-5： VRCON：参考电压控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
C1VREN	C2VREN	VRR	VP6EN	VR3	VR2	VR1	VR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **C1VREN：**比较器 1 参考电压使能位
`1` = CVREF 电路上电并送至比较器 C1 的 C1VREF 输入端
`0` = 0.6 伏恒定参考电压被送至比较器 C1 的 C1VREF 输入端

bit 6 **C2VREN：**比较器 2 参考电压使能位
`1` = CVREF 电路上电并送至比较器 C2 的 C2VREF 输入端
`0` = 0.6 伏恒定参考电压被送至比较器 C2 的 C2VREF 输入端

bit 5 **VRR：**CVREF 范围选择位
`1` = 低电压范围
`0` = 高电压范围

bit 4 **VP6EN：**0.6V 参考电压使能位
`1` = 使能
`0` = 禁止

bit 3-0 **VR<3:0>：**比较器参考电压 CVREF 值选择位 （0 ≤ VR<3:0> ≤ 15）
当 VRR = `1` 时：CVREF = (VR<3:0>/24) * VDD
当 VRR = `0` 时：CVREF = VDD/4 + (VR<3:0>/32) * VDD

表 8-2： 与比较器和参考电压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	POR/BOR 时的值	所有其他复位时的值
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 0000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	00-- --10	00-- --10
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	0000 ----
PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000----	0000----
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
SRCON	SR1	SR0	C1SEN	C2REN	PULSS	PULSR	—	—	0000 00--	0000 00--
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
VRCON	C1VREN	C2VREN	VRR	VP6EN	VR3	VR2	VR1	VR0	0000 0000	0000 0000

图注： x = 未知， u = 不变， – = 未实现 （读为 0）。比较器不使用阴影单元。

9.0 模数转换器 （ADC）模块

模数转换器（Analog-to-digital Converter，ADC）可将模拟输入信号转换为信号的 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同一个采样和保持电路。采样保持电路的输出与转换器的输入相连接。转换器通过逐次逼近法产生 10 位二进制结果，并将转换结果存储在 ADC 结果寄存器 （ADRESL 和 ADRESH）中。

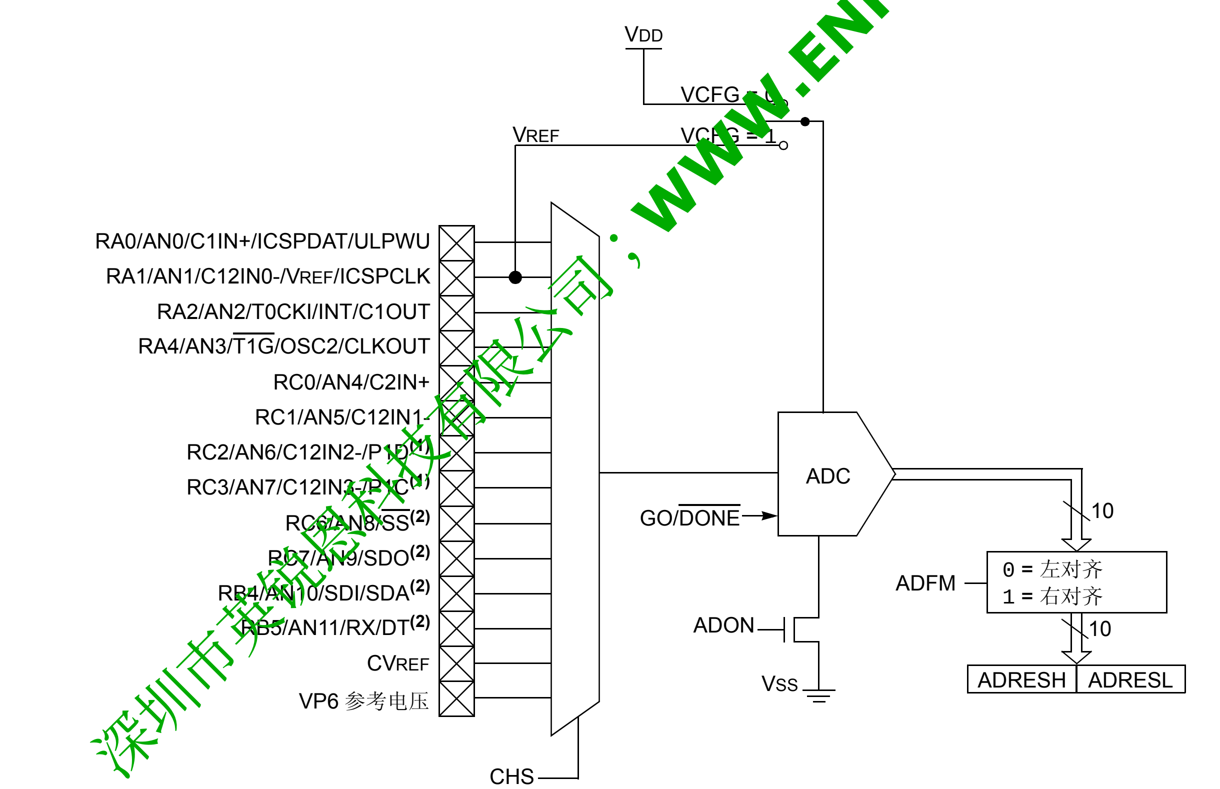
> **注：** 只有 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690 器件上有 ADC 模块。

可通过软件方式选择内部产生的电压或外部提供的电压作为 ADC 参考电压。

ADC 可在转换完成时产生中断。该中断可用于将器件从休眠状态唤醒。

图 9-1 给出了 ADC 的框图。

图 9-1： ADC 框图



注 1： P1C 和 P1D 仅在 PIC16F685/PIC16F690 上可用。
2： SS、 SDO、 SDA、 RX 和 DT 仅在 PIC16F677/PIC16F687/PIC16F689/PIC16F690 上可用。
3： 只有 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690 器件上有 ADC 模块。

9.1 ADC 配置

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

9.1.1 端口配置

ADC 可用于转换模拟和数字信号。转换模拟信号时，I/O 引脚应通过将相关的 TRIS 和 ANSEL 位置 1 配置为模拟。更多信息，请参见有关相应端口的章节。

注：	在任何定义为数字输入的引脚上施加模拟电压可能导致输入缓冲器消耗的电流增加。

9.1.2 通道选择

ADCON0 寄存器的 CHS 位决定与采样和保持电路相连接的通道。

改变通道时，在开始下一次转换前需要一段延时。更多信息，请参见第 **9.2 节 “ADC 工作原理”**。

9.1.3 ADC 参考电压

ADCON0 寄存器的 VCFG 位用于控制正参考电压。正参考电压可以是 V_{DD} 或外部电压源。负参考电压总是连接到参考地。

9.1.4 转换时钟

可通过软件方式设置 ADCON1 寄存器的 ADCS 位来选择转换时钟源。有以下 7 种时钟频率可供选择：

- F_{OSC}/2
- F_{OSC}/4
- F_{OSC}/8
- F_{OSC}/16
- F_{OSC}/32
- F_{OSC}/64
- F_{RC}（专用内部振荡器）

完成一个位转换所需的时间定义为 T_{AD}。一次完整的 10 位转换需要 11 个 T_{AD} 周期，如图 9-2 中所示。

为正确转换，必须满足适当的 T_{AD} 规范。更多信息，请参见第 **17.0 节 “电气规范”**中的 A/D 转换要求。表 9-1 给出了适当的 ADC 时钟选择的示例。

注：	除非使用 F_{RC}，否则系统时钟频率的任何改变都将改变 ADC 时钟频率，从而对 ADC 结果产生不利影响。

表 9-1： ADC 时钟周期 （TAD）与器件工作频率关系表 （VDD ≥ 3.0V， VREF ≥ 2.5V）

ADC 时钟周期 （TAD）		器件频率 （FOSC）			
ADC 时钟源	**ADCS<2:0>**	**20 MHz**	**8 MHz**	**4 MHz**	**1 MHz**
FOSC/2	000	100 ns(2)	250 ns(2)	500 ns(2)	2.0 μs
FOSC/4	100	200 ns(2)	500 ns(2)	1.0 μs(2)	4.0 μs
FOSC/8	001	400 ns(2)	1.0 μs(2)	2.0 μs	8.0 μs(3)
FOSC/16	101	800 ns(2)	2.0 μs	4.0 μs	16.0 μs(3)
FOSC/32	010	1.6 μs	4.0 μs	8.0 μs(3)	32.0 μs(3)
FOSC/64	110	3.2 μs	8.0 μs(3)	16.0 μs(3)	64.0 μs(3)
FRC	x11	2-6 μs(1,4)	2-6 μs(1,4)	2-6 μs(1,4)	2-6 μs(1,4)

图注： 阴影单元表示超出了建议范围。

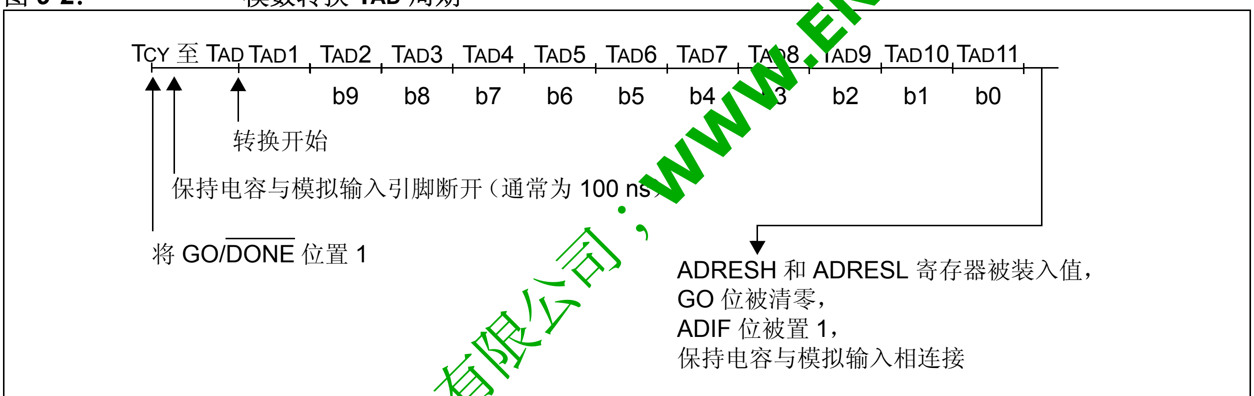
注 1： VDD > 3.0V 时， FRC 时钟源具有 4 μs 的典型 TAD 时间。

2： 这些值均违反了所需的最小 TAD 时间。

3： 为了加快转换速度，建议选用其他时钟源。

4： 当器件频率高于 1 MHz 时，仅当在休眠状态下进行转换时才推荐使用 FRC 时钟源。

图 9-2： 模数转换 TAD 周期



9.1.5 中断

ADC 模块可在模数转换完成时产生中断。 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。 ADIF 位必须用软件清零。

> 注： ADIF 位在每次转换完成时置 1，与是否允许了 ADC 中断无关。

器件运行或休眠时都可产生该中断。如果器件处于休眠状态，该中断会唤醒器件。从休眠状态唤醒时，总是执行 SLEEP 指令后紧跟的下一条指令。如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。如果允许了全局中断，执行将切换到中断服务程序。

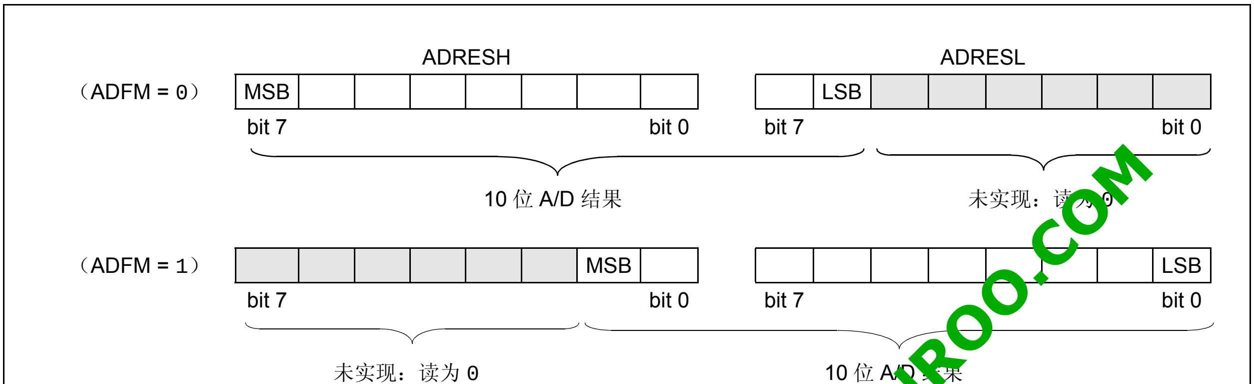
更多信息，请参见第 **9.1.5** 节 “中断”。

9.1.6 结果格式

10 位 A/D 转换结果可以两种格式提供：左对齐或右对齐。 ADCON0 寄存器的 ADFM 位控制输出格式。

图 9-3 给出了两种输出格式。

图 9-3： 10 位 A/D 转换结果格式



9.2 ADC 工作原理

9.2.1 启动转换

要使能 ADC 模块，ADCON0 寄存器的 ADON 位必须设置为 1。将 ADCON0 寄存器的 GO/$\overline{\text{DONE}}$ 位设置为 1 将启动模数转换。

注：	不应在启动 ADC 的同一条指令中将 GO/$\overline{\text{DONE}}$ 位置 1。请参见第 **9.2.6 节“A/D 转换步骤”**。

9.2.2 转换完成

转换完成时， ADC 模块将：

- 清零 GO/$\overline{\text{DONE}}$ 位
- 将 ADIF 标志位置 1
- 用新的转换结果更新 ADRESH:ADRESL 寄存器

9.2.3 终止转换

如果必须在转换完成前终止转换，可用软件将 GO/$\overline{\text{DONE}}$ 位清零。不会用部分完成的模数转换结果更新 ADRESH 和 ADRESL 寄存器， ADRESH:ADRESL 寄存器对仍然保留前一次转换的值。此外，必须经过 2 T_{AD} 的延时后才能开始下一次采集。延时结束后，将自动开始对选定通道的输入进行采集。

注：	器件复位将强制所有寄存器为复位状态。因此， ADC 模块被关闭，任何进行中的转换操作被终止。

9.2.4 休眠期间的 ADC 操作

ADC 模块可以在休眠模式下工作。这需要把 ADC 转换时钟设置为 FRC 选项。选择 FRC 时钟源后，ADC 需等待一个指令周期后才能启动转换操作。这使得可以执行 SLEEP 指令，以降低转换期间的系统噪声。如果允许了 ADC 中断，转换完成时器件将从休眠状态唤醒。如果禁止了 ADC 中断，尽管 ADON 位仍保持置 1，转换完成后 ADC 模块将关闭。

ADC 时钟源不是 FRC 时，尽管 ADON 位仍保持置 1， SLEEP 指令会导致当前转换中止， ADC 模块关闭。

9.2.5 特殊事件触发器

ECCP 特殊事件触发器允许定期测试 ADC 而无需软件干预。当出现触发信号后， GO/$\overline{\text{DONE}}$ 位由硬件置 1， Timer1 计数器复位为零。

使用特殊事件触发器不能确保正确的 ADC 时序。用户需负责确保 ADC 时序要求得到满足。

更多信息，请参见**第 11.0 节“增强型捕捉 / 比较 /PWM 模块”**。

9.2.6 A/D 转换步骤

以下是用 ADC 执行模数转换的示例步骤：

1. 配置端口：
   - 禁止引脚输出驱动器（见 TRIS 寄存器）
   - 将引脚配置为模拟
2. 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
   - 选择结果格式
   - 开启 ADC 模块
3. 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 (1)
4. 等待所需采集时间 (2)。
5. 通过将 GO/$\overline{\text{DONE}}$ 位置 1 启动转换。
6. 通过以下方式之一等待 ADC 转换完成：
   - 查询 GO/$\overline{\text{DONE}}$ 位
   - 等待 ADC 中断（已允许中断）
7. 读取 ADC 结果
8. 清零 ADC 中断标志（如果已允许中断则需要）。

注	
1：	如果用户试图从休眠状态唤醒器件并恢复主代码执行，必须禁止全局中断。
2：	请参见**第 9.3 节“A/D 采集要求”**。

例 9-1： A/D 转换

```
;This code block configures the ADC
;for polling, Vdd reference, Frc clock
;and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
BANKSEL   ADCON1       ;
MOVLW     B'01110000'  ;ADC Frc clock
MOVWF     ADCON1       ;
BANKSEL   TRISA        ;
BSF       TRISA,0      ;Set RA0 to input
BANKSEL   ANSEL        ;
BSF       ANSEL,0      ;Set RA0 to analog
BANKSEL   ADCON0       ;
MOVLW     B'10000001'  ;Right justify,
MOVWF     ADCON0       ; Vdd Vref, AN0, On
CALL      SampleTime   ;Acquisiton delay
BSF       ADCON0,GO    ;Start conversion
BTFSC     ADCON0,GO    ;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
```

9.2.7 ADC 寄存器定义

以下寄存器用于控制 ADC 的操作。

寄存器 9-1： ADCON0：A/D 控制寄存器 0

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ADFM：**A/D 转换结果格式选择位
1 = 右对齐
0 = 左对齐

bit 6 **VCFG：**参考电压位
1 = VREF 引脚
0 = VDD

bit 5-2 **CHS<3:0>：模拟通道选择位**
0000 = AN0
0001 = AN1
0010 = AN2
0011 = AN3
0100 = AN4
0101 = AN5
0110 = AN6
0111 = AN7
1000 = AN8
1001 = AN9
1010 = AN10
1011 = AN11
1100 = CVREF
1101 = 0.6V 固定参考电压
1110 = 保留。不要使用。
1111 = 保留。不要使用。

bit 1 **GO/$\overline{\text{DONE}}$：**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。
A/D 转换完成后，该位由硬件自动清零。
0 = A/D 转换已完成 / 未进行

bit 0 **ADON：**ADC 使能位
1 = ADC 使能
0 = ADC 禁止，不消耗工作电流

寄存器 9-2： ADCON1：A/D 控制寄存器 1

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	ADCS2	ADCS1	ADCS0	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 未实现：读为 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
x11 = FRC（由专用内部振荡器产生的时钟，其频率的最大值为 500 kHz）
100 = FOSC/4
101 = FOSC/16
110 = FOSC/64

bit 3-0 未实现：读为 0

寄存器 9-3： ADRESH：ADC 结果寄存器高字节 （ADRESH） ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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 9-4： ADRESL：ADC 结果寄存器低字节 （ADRESL） ADFM = 0

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES1	ADRES0	—	—	—	—	—	—
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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：**不要使用。

寄存器 9-5： ADRESH：ADC 结果寄存器高字节 （ADRESH） ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—	—	—	ADRES9	ADRES8
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-2 **保留：**不要使用。

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 9-6： ADRESL：ADC 结果寄存器低字节 （ADRESL） ADFM = 1

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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

9.3 A/D 采集要求

为了使 ADC 达到规定的精度，必须使充电保持电容（CHOLD）充满至输入通道的电平。模拟输入模型见图 9-4。模拟信号源阻抗（RS）和内部采样开关阻抗（RSS）直接影响电容 CHOLD 的充电时间。采样开关阻抗（RSS）随器件电压（VDD）的变化而变化，参见图 9-4。**建议模拟信号源的最大阻抗为 10 kΩ**。采集时间随着源阻抗的降低而缩短。在选择（或改变）模拟输入通道后，必须在开始转换前完成 A/D 采集。

可以使用公式 9-1 来计算最小采集时间。该公式假设误差为 1/2 LSb（ADC 转换需要 1024 步）。1/2 LSb 误差是 ADC 达到规定分辨率所允许的最大误差。

公式 9-1： 采集时间示例

假设： 温度 = 50°C，外部阻抗 10KΩ，5.0V V_{DD}

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

T_C 值可以用以下公式近似计算：

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

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

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

注：其中 n = ADC 的位数。

求解 T_C：

$$T_C = -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ \ln(1/2047)$$
$$= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ \ln(0.0004885)$$
$$= 1.37\mu s$$

因此：

$$T_{ACQ} = 2\mu s + 1.37\mu s + [(50°C - 25°C)(0.05\mu s/°C)]$$
$$= 4.67\mu s$$

注 1： 因为参考电压（VREF）自行抵消，因此它对该公式没有影响。

2： 充电保持电容（CHOLD）在每次转换后不会放电。

3： 建议模拟信号源的最大阻抗为 10 kΩ。此要求是为了符合引脚泄漏电流规范。

图 9-4： 模拟输入模型

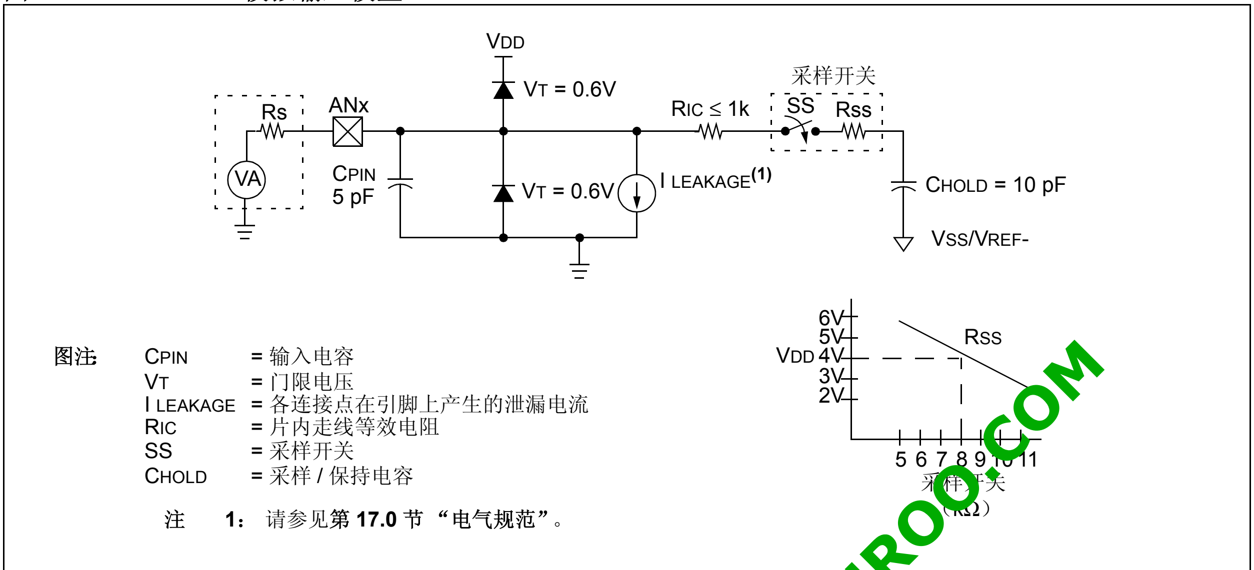


图 9-5： ADC 传递函数

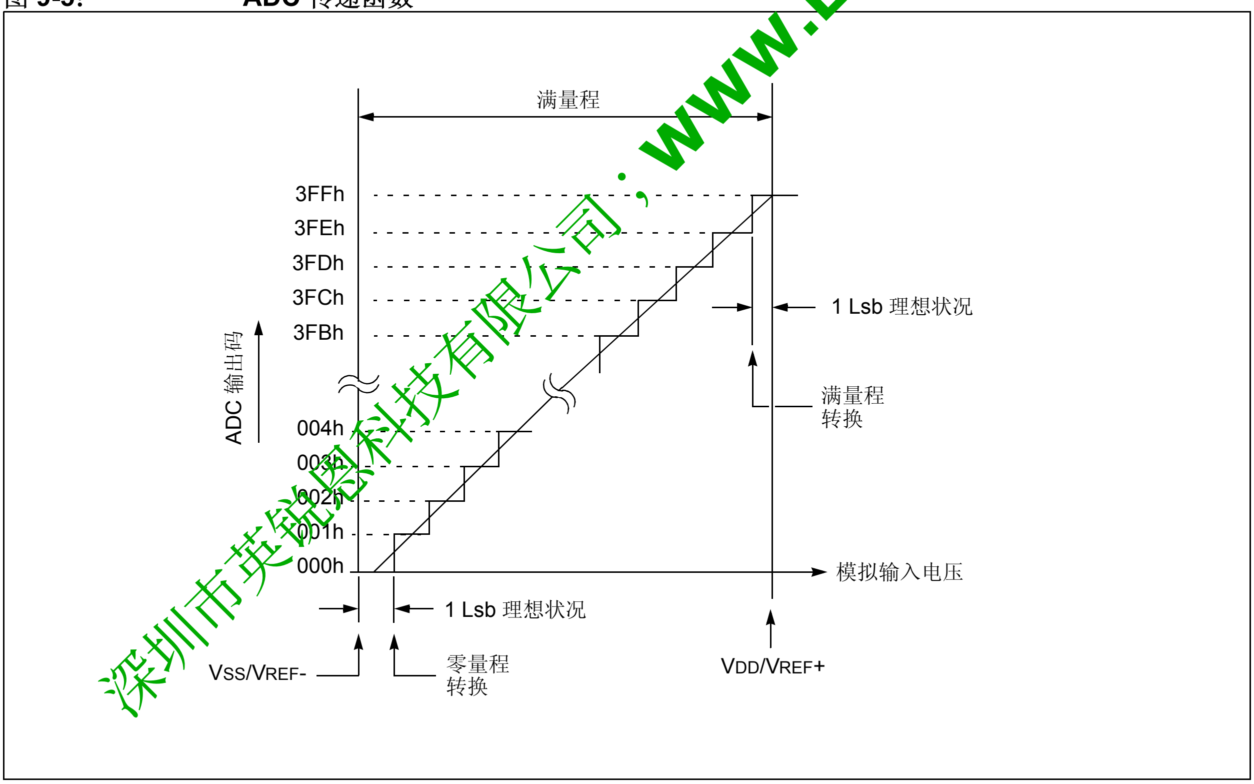


表 9-2： 与 ADC 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
ADCON0	ADFM	VCFG	CHS3	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	ADON	0000 0000	0000 0000
ADCON1	—	ADCS2	ADCS1	ADCS0	—	—	—	—	-000 ----	-000 ----
ANSEL	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0	1111 1111	1111 1111
ANSELH	—	—	—	—	ANS11	ANS10	ANS9	ANS8	---- 1111	---- 1111
ADRESH	A/D 结果寄存器的高字节								xxxx xxxx	uuuu uuuu
ADRESL	A/D 结果寄存器的低字节								xxxx xxxx	uuuu uuuu
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--xx xxxx	--uu uuuu
PORTB	RB7	RB6	RB5	RB4	—	—	—	—	xxxx ----	uuuu ----
PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
TRISA	—	—	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	--11 1111	--11 1111
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

图注： x = 未知， u = 不变， — = 未实现 （读为 0）。 ADC 模块不使用阴影单元。

10.0 数据 EEPROM 和闪存程序存储器控制

正常运行期间（整个 V_{DD} 范围内），数据 EEPROM 存储器（仅限 PIC16F685/PIC16F689/PIC16F690）是可读写的，而闪存程序存储器是可读的。这些存储器并不直接映射到寄存器文件空间，而是通过特殊功能寄存器（SFR）来间接寻址。有 6 个 SFR 用于访问这些存储器，如下：

- EECON1
- EECON2
- EEDAT
- EEDATH （仅限 PIC16F685/PIC16F689/PIC16F690）
- EEADR
- EEADRH （仅限 PIC16F685/PIC16F689/PIC16F690）

当与数据存储器模块接口时， EEDAT 存放 8 位读写数据，而 EEADR 存放被访问的 EEDAT 存储单元的地址。这些器件（PIC16F631 除外）有 256 字节的数据 EEPROM，地址范围从 0h 到 0FFh。PIC16F631 有 128 字节的数据 EEPROM，地址范围从 0h 到 07Fh。

寻址 PIC16F685/PIC16F689/PIC16F690 器件的程序存储器模块时，EEDAT 和 EEDATH 寄存器形成双字节字，保存 14 位读 / 写数据，而 EEADR 和 EEADRH 寄存器形成双字节字，保存 12 位被读取的 EEPROM 存储单元的地址。这些器件（PIC16F685/PIC16F689/PIC16F690）有 4K 字的程序 EEPROM，地址范围从 0h 到 0FFFh。程序存储器允许以字为单位读取。

EEPROM 数据存储器允许以字节为单位读写。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 / 擦除电压是由片上电荷泵产生的，此电荷泵在器件字节或字操作的电压范围内工作。

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

10.1 EEADR 和 EEADRH 寄存器

EEADR 和 EEADRH 寄存器能寻址最大 256 字节的数据 EEPROM 或最大 4K 字的程序 EEPROM。

当选择程序地址值时，地址的 MSB 被写入 EEADRH 寄存器，而 LSB 被写入 EEADR 寄存器。当选择数据地址值时，只将地址的 LSB 写入 EEADR 寄存器。

10.1.1 EECON1 和 EECON2 寄存器

EECON1 是访问 EE 存储器的控制寄存器。

控制位 EEPGD （PIC16F685/PIC16F689/PIC16F690）决定访问的是程序存储器还是数据存储器。当清零时，和在复位时一样，任何后续操作都将针对数据存储器进行。当置 1 时，任何后续操作都将针对程序存储器进行。程序存储器是只读的。

控制位 RD 和 WR 分别用于启动读和写操作。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，可避免写操作意外地过早终止。

当 WREN 位置 1 时，允许对数据 EEPROM 执行写操作。上电时，WREN 位被清零。当正常的写操作被 $\overline{MCLR}$ 复位或 WDT 超时复位中断时， WRERR 位会置 1。在这些情况下，复位后用户可以检查 WRERR 位并重写相应的存储单元。

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

EECON2 不是实际存在的寄存器。读 EECON2 将得到全 0。EECON2 寄存器仅在数据 EEPROM 的写过程中使用。

寄存器 10-1： EEDAT：EEPROM 数据寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 **EEDAT<7:0>**：写入数据 EEPROM 或从数据 EEPROM 或程序存储器读取的 8 个最低有效地址位

寄存器 10-2： EEADR：EEPROM 地址寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EEADR7[(1)]	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-0 **EEADR<7:0>**：EEPROM 读 / 写操作 [(1)] 或从程序存储器读取的 8 个最低有效地址位

注 1： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

寄存器 10-3： EEDATH：EEPROM 数据高字节寄存器 [(1)]

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **未实现**：读为 0

bit 5-0 **EEDATH<5:0>**：来自程序存储器的 6 个最高有效数据位

注 1： 仅限 PIC16F685/PIC16F689/PIC16F690。

寄存器 10-4： EEADRH：EEPROM 地址高字节寄存器 [(1)]

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-4 **未实现**：读为 0

bit 3-0 **EEADRH<3:0>**：指定 4 个最高有效地址位或读取程序存储器的高位

注 1： 仅限 PIC16F685/PIC16F689/PIC16F690。

寄存器 10-5： EECON1：EEPROM 控制寄存器

R/W-x	U-0	U-0	U-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD[1]	—	—	—	WRERR	WREN	WR	RD
bit 7							bit 0

图注：			
S = 该位只能被置 1			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **EEPGD**：程序 / 数据 EEPROM 选择位 [1]
1 = 访问程序存储器
0 = 访问数据存储器

bit 6-4 **未实现**：读为 0

bit 3 **WRERR**：EEPROM 错误标志位
1 = 写操作被过早终止 （正常操作中任何 $\overline{\text{MCLR}}$ 复位或任何 WDT 复位，或由于 BOR 复位）
0 = 写操作完成

bit 2 **WREN**：EEPROM 写操作使能位
1 = 允许写操作
0 = 禁止向数据 EEPROM 执行写操作

bit 1 **WR**：写操作控制位
<u>EEPGD = 1：</u>
该位为无关位
<u>EEPGD = 0：</u>
1 = 启动一次写操作 （写操作完成时此位由硬件清零。 WR 位只能用软件置 1，不能清零。）
0 = 向数据 EEPROM 进行的写操作完成

bit 0 **RD**：读操作控制位
1 = 启动一次存储器读操作 （RD 位由硬件清零，只能用软件置 1，不能清零。）
0 = 不启动对存储器的读操作

注 1： 仅限 PIC16F685/PIC16F689/PIC16F690。

10.1.2 读数据 EEPROM 存储器

要读取数据存储器单元，用户必须将地址写入 EEADR 寄存器，清零 EECON1 寄存器的 EEPGD 控制位，然后将控制位 RD 置 1。在紧接着的下一个周期， EEDAT 寄存器中就有数据了，因此该数据可由下一条指令读取。 EEDAT 将把此值保存至下一次用户向该单元读取或写入数据时 （在写操作过程中）为止。

例 10-1： 读数据 EEPROM

```
BANKSEL  EEADR          ;
MOVF     DATA_EE_ADDR,  W;
MOVWF    EEADR          ;Data Memory
                        ;Address to read
BANKSEL  EECON1         ;
BCF      EECON1, EEPGD  ;Point to DATA memory
BSF      EECON1, RD     ;EE Read
BANKSEL  EEDAT          ;
MOVF     EEDAT, W       ;W = EEDAT
BANKSEL  PORTA          ;Bank 0
```

10.1.3 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 EEADR 寄存器，并将数据写入 EEDAT 寄存器。然后用户必须按特定顺序开始写入每个字节。

如果没有完全按照这一特定的指令顺序（即首先将 55h 写入 EECON2，随后将 AAh 写入 EECON2，最后将 WR 位置 1）写每个字节，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 以使能写操作。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，用户应该始终保持 WREN 位清零。 WREN 位不能被硬件清零。

一个写过程启动后，将 WREN 位清零不会影响此写周期。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。用户可以允许此中断或查询此位。 EEIF 必须用软件清零。

例 10-2： 写数据 EEPROM

```
BANKSEL  EEADR            ;
MOVF     DATA_EE_ADDR,  W;
MOVWF    EEADR            ; Data Memory Address to write
MOVF     DATA_EE_DATA,  W;
MOVWF    EEDAT            ;Data Memory Value to write
BANKSEL  EECON1           ;
BCF      EECON1, EEPGD    ; Point to DATA memory
BSF      EECON1, WREN     ; Enable writes

BCF      INTCON, GIE      ;Disable INTs.
BTFSC    INTCON, GIE      ;SEE AN576
GOTO     $-2
MOVLW    55h              ;
MOVWF    EECON2           ;Write 55h
MOVLW    AAh              ;
MOVWF    EECON2           ;Write AAh
BSF      EECON1, WR       ;Set WR bit to begin write
BSF      INTCON, GIE      ;Enable INTs.

SLEEP                     ;Wait for interrupt to signal write complete (optional)
BCF      EECON1, WREN     ;Disable writes
BANKSEL  0x00             ;Bank 0
```

必需的序列

10.1.4 读闪存程序存储器 (PIC16F685/PIC16F689/PIC16F690)

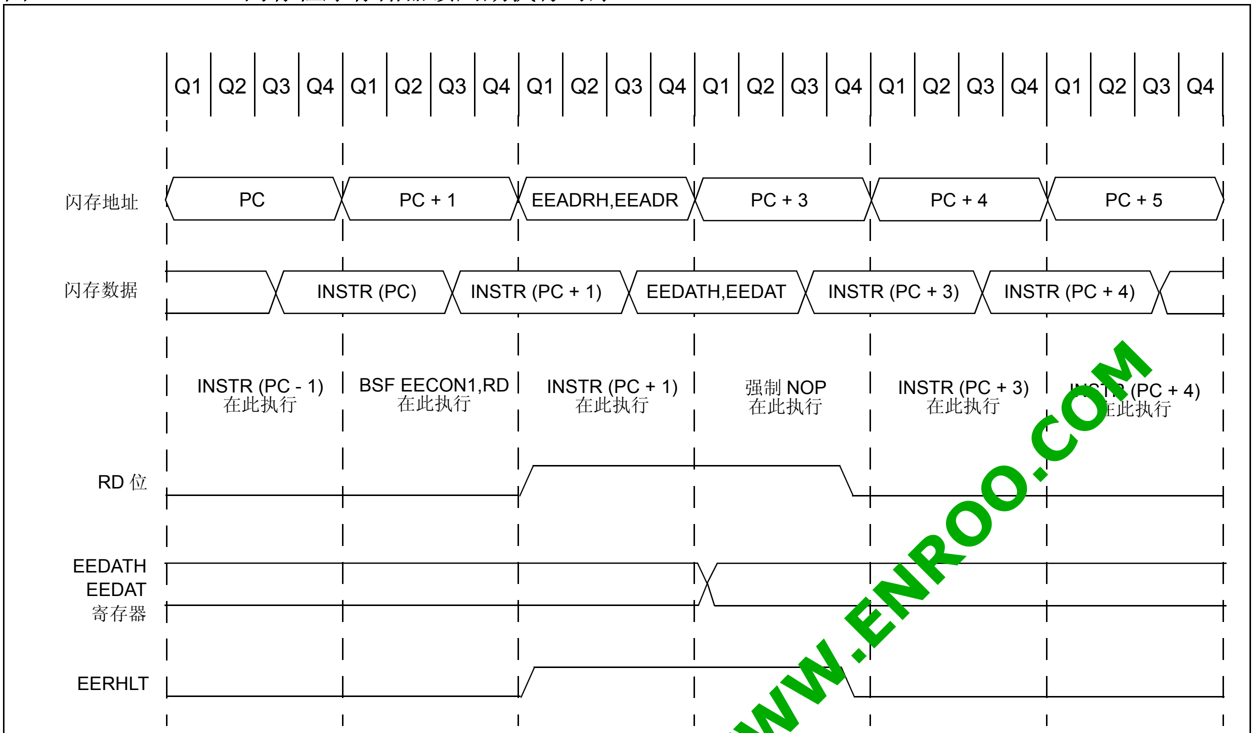
要读取程序存储器单元，用户必须将最低和最高有效地址位分别写入EEADR和EEADRH寄存器，将EECON1寄存器的 EEPGD 控制位置 1，然后将控制位 RD 置 1。一旦设置好读控制位，闪存程序存储器控制器将使用第二个指令周期来读数据。这会导致紧随 “BSF EECON1,RD” 指令的第二条指令被忽略。在紧接着的下一个周期，EEDAT 和 EEDATH 寄存器中就有数据了；因此在随后的指令中将该数据读作两个字节。EEDAT 和 EEDATH 寄存器将把此值保存至下一次用户向该单元读取或写入数据时为止。

> 注 1：程序存储器读操作后的两条指令必须为 NOP，从而阻止用户在 RD 位置 1 后的下一条指令执行双周期指令。
>
> 2：当 EEPGD = 1 时如果 WR 位置 1，它会立即复位为 0，而不执行任何操作。

例 10-3： 读闪存程序存储器

```
        BANKSEL EEADR              ;
        MOVF    MS_PROG_EE_ADDR, W ;
        MOVWF   EEADRH             ;MS Byte of Program Address to read
        MOVF    LS_PROG_EE_ADDR, W ;
        MOVWF   EEADR              ;LS Byte of Program Address to read
        BANKSEL EECON1             ;
   ┌    BSF     EECON1, EEPGD      ;Point to PROGRAM memory
必 │    BSF     EECON1, RD         ;EE Read
需 │
的 │    NOP                        ;First instruction after BSF EECON1,RD executes normally
序 │
列 └    NOP                        ;Any instructions here are ignored as program
                                   ;memory is read in second cycle after BSF EECON1,RD
                                   ;
        BANKSEL EEDAT              ;
        MOVF    EEDAT, W           ;W = LS Byte of Program Memory
        MOVWF   LOWPMBYTE          ;
        MOVF    EEDATH, W          ;W = MS Byte of Program EEDAT
        MOVWF   HIGHPMBYTE         ;
        BANKSEL 0x00               ;Bank 0
```

图 10-1：　闪存程序存储器读周期执行时序



10.2 写校验

根据具体应用，将写入数据 EEPROM 的值与期望写入的值相校验 （见例 10-4）是一个很好的编程习惯。

例 10-4： 写校验

```
BANKSEL EEDAT          ;
MOVF    EEDAT, W       ;EEDAT not changed
                       ; from previous write
BANKSEL EECON1         ;
BSF     EECON1, RD     ;YES, Read the
                       ;value written
BANKSEL EEDAT          ;
XORWF   EEDAT, W       ;
BTFSS   STATUS, Z      ;Is data the same
GOTO    WRITE_ERR      ;No, handle error
:                      ;Yes, continue
BANKSEL 0x00           ;Bank 0
```

10.2.1 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变动的信息（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变动，而另一个段中的变量不变时，就可能在不超出对一个字节的总写入次数时 （参数 D120 和 D120A），超出向 EEPROM 写入的总次数 （参数 D124）。如果是这种情况，就必须对队列进行更新。出于此原因，不会变化的量 （如常量、 ID 和校准值等）应存放在闪存程序存储器中。

10.3 防止误写操作的保护措施

有些情况下，用户并不希望写入数据EEPROM存储器。为防止EEPROM误写操作，芯片内嵌了各种保护机制。上电时，WREN 位被清零。而且，上电延时定时器（延时 64 ms）也会阻止对 EEPROM 进行写操作。

写操作的启动顺序以及 WREN 位将共同防止在以下情况下意外误写操作的发生：

- 欠压
- 电源毛刺
- 软件故障

10.4 代码保护期间数据 EEPROM 的操作

在配置字寄存器（寄存器 14-1）中将 CPD 位编程为 0，可对数据存储器进行代码保护。

数据存储器被代码保护时，仅 CPU 可对数据 EEPROM 进行读写操作。对数据存储器进行代码保护和采用 NOP 指令编程未使用的程序存储区时，建议用户也对程序存储器进行代码保护。

表 10-1： 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	POR/BOR 时的值	所有其他复位时的值
EECON1	EEPGD(1)	—	—	—	WRERR	WREN	WR	RD	x--- x000	0--- q000
EECON2	EEPROM 控制寄存器 2 （不是实际存在的寄存器）								---- ----	---- ----
EEADR	EEADR7(2)	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0	0000 0000	0000 0000
EEADRH(1)	—	—	—	—	EEADRH3	EEADRH2	EEADRH1	EEADRH0	---- 0000	---- 0000
EEDAT	EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0	0000 0000	0000 0000
EEDATH(1)	—	—	EEDATH5	EEDATH4	EEDATH3	EEDATH2	EEDATH1	EEDATH0	--00 0000	--00 0000
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 0000	0000 0000
PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	0000 ----
PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000 ----	0000 ----

图注： x = 未知， u = 不变， — = 未实现 （读为 0）， q = 值取决于具体条件。
数据 EEPROM 模块不使用阴影单元。

注 1： 仅限 PIC16F685/PIC16F689/PIC16F690。
2： 仅限 PIC16F677/PIC16F685/PIC16F687/PIC16F689/PIC16F690。

11.0 增强型捕捉 / 比较 /PWM 模块

增强型捕捉 / 比较 /PWM 模块是允许用户计时和控制不同事件的外设。在捕捉模式下，外设允许对事件的持续时间进行计时。当预先确定的时间超时后，比较模式允许用户触发一个外部事件。 PWM 模式可以产生不同频率和占空比的脉宽调制（Pulse-Width Modulate）信号。

表 11-1 给出了 ECCP 模块所需的定时器资源。

表 11-1： ECCP 模式——所需的定时器资源

ECCP 模式	定时器资源
捕捉	Timer1
比较	Timer1
PWM	Timer2

寄存器 11-1： CCP1CON：增强型 CCP1 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-6 **P1M<1:0>**：PWM 输出配置位
如果 CCP1M<3:2> = 00、 01 和 10：
xx = P1A 配置为捕捉 / 比较输入； P1B、 P1C 和 P1D 配置为端口引脚
如果 CCP1M<3:2> = 11：
00 = 单输出； P1A 调制； P1B、 P1C 和 P1D 配置为端口引脚
01 = 全桥正向输出； P1D 调制； P1A 有效； P1B 和 P1C 无效
10 = 半桥输出； P1A 和 P1B 调制，带有死区控制； P1C 和 P1D 配置为端口引脚
11 = 全桥反向输出； P1B 调制； P1C 有效； P1A 和 P1D 无效

bit 5-4 **DC1B<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 CCPR1L 中。

bit 3-0 **CCP1M<3:0>**：ECCP 模式选择位
0000 = 捕捉 / 比较 /PWM 关闭 （复位 ECCP 模块）
0001 = 未使用 （保留）
0010 = 比较模式，匹配时翻转输出 （CCP1IF 位置 1）
0011 = 未使用 （保留）
0100 = 捕捉模式，每个下降沿
0101 = 捕捉模式，每个上升沿
0110 = 捕捉模式，每 4 个上升沿
0111 = 捕捉模式，每 16 个上升沿
1000 = 比较模式，匹配时输出置 1 （CCP1IF 位置 1）
1001 = 比较模式，匹配时输出清零 （CCP1IF 位置 1）
1010 = 比较模式，匹配时产生软件中断 （CCP1IF 位置 1， CCP1 引脚不受影响）
1011 = 比较模式，触发特殊事件（CCP1IF 位置 1，CCP1 复位 TMR1 或 TMR2，且如果 ADC 模块被使能，启动一次 A/D 转换）
1100 = PWM 模式； P1A 和 P1C 高电平有效； P1B 和 P1D 高电平有效
1101 = PWM 模式； P1A 和 P1C 高电平有效； P1B 和 P1D 低电平有效
1110 = PWM 模式； P1A 和 P1C 低电平有效； P1B 和 P1D 高电平有效
1111 = PWM 模式； P1A 和 P1C 低电平有效； P1B 和 P1D 低电平有效

11.1 捕捉模式

在捕捉模式下，当在引脚 CCP1 上发生下列事件之一时， CCPR1H:CCPR1L 捕捉 TMR1 寄存器中的 16 位值。这些事件如下所示，可由 CCP1CON 寄存器的 CCP1M<3:0> 位进行配置：

- 每个下降沿
- 每个上升沿
- 每 4 个上升沿
- 每 16 个上升沿

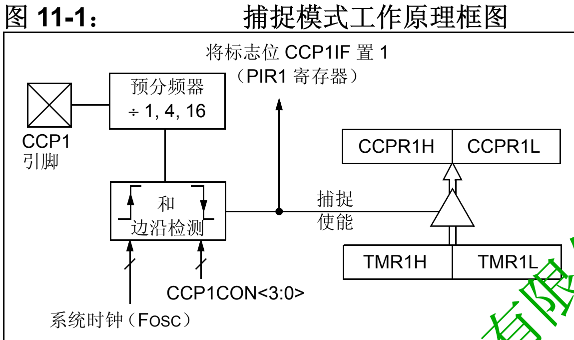
进行捕捉后，PIR1 寄存器的中断请求标志位 CCP1IF 被置 1。该中断标志位必须用软件清零。如果在 CCPR1H 和CCPR1L寄存器对中的值被读出之前又发生另一次捕捉，那么原来的捕捉值会被新捕捉值覆盖（见图 11-1）。

11.1.1 CCP1 引脚配置

在捕捉模式下，应该通过把相关的 TRIS 控制位置 1 将 CCP1 引脚配置为输入。

注：	如果 CCP1 引脚被配置为输出，则写端口将产生一次捕捉条件。

图 11-1： 捕捉模式工作原理框图



11.1.2 TIMER1 模式选择

为使 CCP 模块使用捕捉特性， Timer1 必须运行在定时器模式或同步计数器模式。在异步计数器模式下，捕捉操作可能无法进行。

11.1.3 软件中断

当捕捉模式改变时，可能会产生错误的捕捉中断。用户应该保持 PIE1 寄存器的 CCP1IE 中断允许位清零以避免错误中断。此外，用户应在工作模式的任何改变之后清零 PIR1 寄存器的 CCP1IF 中断标志位。

11.1.4 CCP 预分频器

通过对CCP1CON寄存器的CCP1M<3:0>位进行设置，可以指定 4 种不同的预分频比。每当关闭 CCP 模块，或者 CCP 模块不在捕捉模式下时，预分频计数器就会被清零。任何复位都会将预分频计数器清零。

从一个捕捉预分频比切换到另一个不会清零预分频器，而且可能产生一次错误中断。为避免此意外操作，可在改变预分频比前通过清零 CCP1CON 寄存器来关闭模块（见例 11-1）。

例 11-1： 改变捕捉预分频比

```
BANKSEL CCP1CON     ;Set Bank bits to point
                    ;to CCP1CON
CLRF    CCP1CON     ;Turn CCP module off
MOVLW   NEW_CAPT_PS ;Load the W reg with
                    ; the new prescaler
                    ; move value and CCP ON
MOVWF   CCP1CON     ;Load CCP1CON with this
                    ; value
```


11.2 比较模式

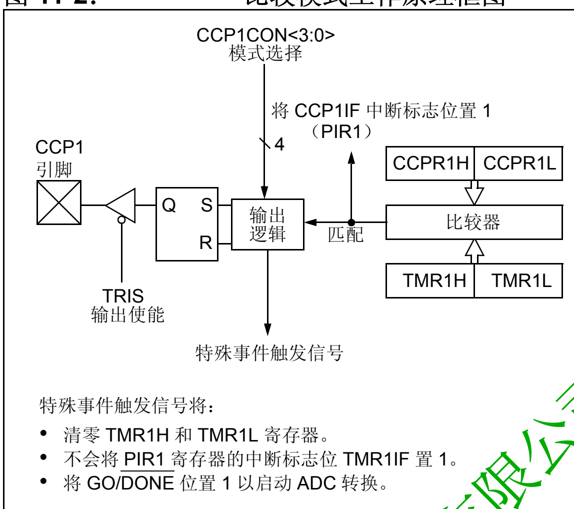
在比较模式下，16位CCPR1寄存器的值不断地与TMR1寄存器对的值作比较。如果二者匹配， CCP 模块可能会：

- 翻转 CCP1 输出
- 将 CCP1 输出置 1
- 将 CCP1 输出清零
- 产生特殊事件触发信号
- 产生软件中断

引脚的动作由 CCP1CON 寄存器的 CCP1M<3:0> 控制位的值决定。

所有比较模式都能产生中断。

图 11-2： 比较模式工作原理框图



11.2.1 CCP1 引脚配置

用户必须通过把相关的 TRIS 位清零将 CCP1 引脚配置为输出。

注：	清零 CCP1CON 寄存器将把 CCP1 比较输出锁存器强制设为默认的低电平。这不是端口 I/O 数据锁存器。

11.2.2 TIMER1 模式选择

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。在异步计数器模式下，比较操作可能无法进行。

11.2.3 软件中断模式

当选择“产生软件中断”模式（CCP1M<3:0> = 1010）时， CCP 模块不会对 CCP1 引脚进行控制（见 CCP1CON 寄存器）。

11.2.4 特殊事件触发器

当选择“特殊事件触发器”模式（CCP1M<3:0> = 1011）时， CCP 模块将进行以下操作：

- 复位 Timer1
- 如果 ADC 被使能，则启动 ADC 转换

在此模式下，CCP 模块不会对 CCP1 引脚进行控制（见 CCP1CON 寄存器）。

一旦TMR1H和TMR1L寄存器对与CCPR1H和CCPR1L寄存器对之间发生匹配，便会发生 CCP 的特殊事件触发输出。TMR1H 和 TMR1L 寄存器对在 Timer1 时钟的下一个上升沿到来之前不会复位。这使 CCPR1H 和 CCPR1L 寄存器对可作为 Timer1 的 16 位可编程周期寄存器。

注	
1：	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。
2：	通过在生成特殊事件触发信号的时钟边沿和使 Timer1 复位的时钟边沿之间更改 CCPR1H 和 CCPR1L 寄存器对的内容来移除匹配条件，可以避免复位发生。

11.3 PWM 模式

PWM 模式在 CCP1 引脚上产生脉宽调制信号。占空比、周期和分辨率由以下寄存器决定：

- PR2
- T2CON
- CCPR1L
- CCP1CON

在脉宽调制（PWM）模式下，CCP 模块会在 CCP1 引脚上产生最大 10 位分辨率的 PWM 输出信号。由于 CCP1 引脚与端口数据锁存器复用，该引脚的 TRIS 必须被清零以使能 CCP1 引脚输出驱动器。

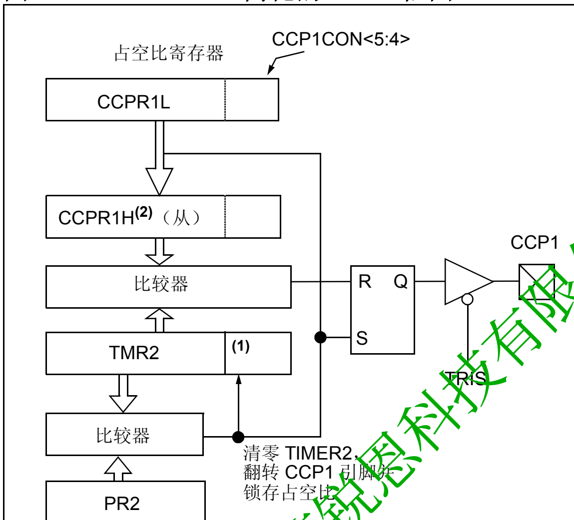
注：	清零 CCP1CON 寄存器将放弃对 CCP1 引脚的 CCP1 控制。

图 11-3 给出了 PWM 工作原理的简化框图。

图 11-4 给出了 PWM 信号的典型波形。

关于如何设置 CCP 模块使之工作于 PWM 模式的详细步骤，请参见**第 11.3.7 节 “设置 PWM 操作”**。

图 11-3： 简化的 PWM 框图

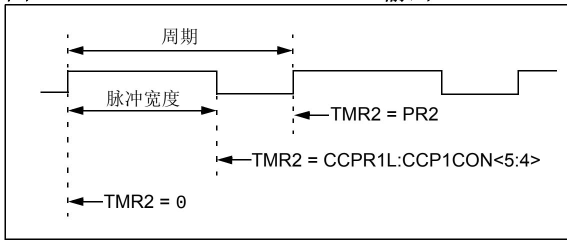


注 1： 8 位定时器 TMR2 寄存器与 2 位内部系统时钟（Fosc）或预分频器的 2 位一起构成 10 位时基。

2： 在 PWM 模式下， CCPR1H 是只读寄存器。

PWM 输出（图 11-4）有一个时基（周期）和一段输出保持为高电平的时间（占空比）。

图 11-4： CCP PWM 输出



11.3.1 PWM 周期

PWM周期可通过写Timer2的PR2寄存器来指定。PWM周期可由公式 11-1 计算。

公式 11-1： PWM 周期

$$PWM周期 = [(PR2) + 1] \bullet 4 \bullet Tosc \bullet (TMR2预分频值)$$

注： Tosc = 1/Fosc

当 TMR2 中的值与 PR2 中的值相等时，在下一个递增周期将发生以下 3 个事件：

- TMR2 被清零
- CCP1 引脚被置 1。（例外情况：如果 PWM 占空比 = 0%，引脚将不会被置 1。）
- PWM 占空比从 CCPR1L 锁存到 CCPR1H。

注：	在确定 PWM 频率时不会用到 Timer2 后分频比（见第 **7.1** 节 **“Timer2** 工作原理**”**）。

11.3.2 PWM 占空比

通过将 10 位值写入多个寄存器来指定 PWM 占空比：CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0> 位。CCPR1L 包含高 8 位而 CCP1CON 寄存器的 DC1B<1:0> 位包含低 2 位。可以在任何时候写入 CCPR1L 和 CCP1CON 寄存器的 DC1B<1:0> 位。在周期结束（即 PR2 和 TMR2 寄存器发生匹配）前占空比值不会被锁存到 CCPR1H 中。当使用 PWM 时，CCPR1H 寄存器是只读的。

公式 11-2 用于计算 PWM 脉冲宽度。

公式 11-3 用于计算 PWM 占空比。

公式 11-2： 脉冲宽度

$$脉冲宽度 = (CCPR1L:CCP1CON<5:4>) \bullet Tosc \bullet (TMR2预分频值)$$

公式 11-3： 占空比

$$占空比 = \frac{(CCPR1L:CCP1CON<5:4>)}{4(PR2 + 1)}$$

CCPR1H 寄存器和一个 2 位的内部锁存器用于给 PWM 占空比提供双重缓冲。这种双重缓冲结构非常重要，它可以避免在 PWM 操作中产生毛刺。

8 位定时器 TMR2 寄存器与 2 位内部系统时钟（Fosc）或预分频器的 2 位一起构成 10 位时基。如果 Timer2 预分频比设置为 1:1，则使用系统时钟。

当 10 位时基与 CCPR1H 和 2 位锁存值匹配时，CCP1 引脚被清零（见图 11-3）。

11.3.3 PWM 分辨率

分辨率决定给定周期的可用占空比数。例如，10 位分辨率将得到 1024 个不连续的占空比，而 8 位分辨率将可得到 256 个不连续的占空比。

当 PR2 为 255 时，PWM 最大分辨率为 10 位。分辨率是 PR2 寄存器值的函数，如公式 11-4 所示。

公式 11-4： PWM 分辨率

$$分辨率 = \frac{\log[4(PR2 + 1)]}{\log(2)} 位$$

注：	如果脉冲宽度值比周期长，则指定的 PWM 引脚将保持不变。

表 11-2： PWM 频率和分辨率示例（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 和 16）	16	4	1	1	1	1
PR2 值	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
最大分辨率（位）	10	10	10	8	7	6.6

表 11-3： PWM 频率和分辨率示例（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 和 16）	16	4	1	1	1	1
PR2 值	0x65	0x65	0x65	0x19	0x0C	0x09
最大分辨率（位）	8	8	8	6	5	5

11.3.4 休眠模式下的操作

在休眠模式下，TMR2 寄存器将不会递增，模块状态也不会改变。如果 CCP1 引脚正在驱动一个值，则会继续驱动该值。当器件被唤醒时，TMR2将从先前状态继续。

11.3.5 改变系统时钟频率

PWM 频率来自于系统时钟频率。系统时钟频率的任何改变将导致 PWM 频率的改变。更多详细信息，请参见**第 3.0 节 “振荡器模块 （带故障保护时钟监视器）”**。

11.3.6 复位的影响

任何复位都将强制所有端口为输入模式，并强制CCP寄存器为其复位状态。

11.3.7 设置 PWM 操作

当配置 CCP 模块的 PWM 操作时，可采用以下步骤：

1. 通过将相关的 TRIS 位置 1，禁止 PWM 引脚（CCP1）输出驱动器。
2. 通过装入 PR2 寄存器设置 PWM 周期。
3. 通过装入恰当的值到 CCP1CON 寄存器将 CCP 模块配置为 PWM 模式。
4. 通过装入 CCPR1L 寄存器和 CCP1CON 寄存器的 DC1B<1:0> 位设置 PWM 占空比。
5. 配置和启动 Timer2：
   - 清零 PIR1 寄存器的 TMR2IF 中断标志位。
   - 通过装入 T2CON 寄存器的 T2CKPS 位设置 Timer2 预分频值。
   - 通过将T2CON寄存器的TMR2ON位置1使能 Timer2。
6. 在新的 PWM 周期开始后使能 PWM 输出：
   - 等待直到 Timer2 溢出 （PIR1 寄存器的 TMR2IF 位置 1）。
   - 通过清零相关的TRIS位使能CCP1 引脚输出驱动器。

11.4 PWM （增强型模式）

增强型 PWM 模式可以在最多 4 个输出引脚上产生 PWM 信号，最高可达 10 位分辨率。可通过 4 种 PWM 输出模式实现：

- 单 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

要选择增强型 PWM 模式，CCP1CON 寄存器的 P1M 位必须适当置 1。

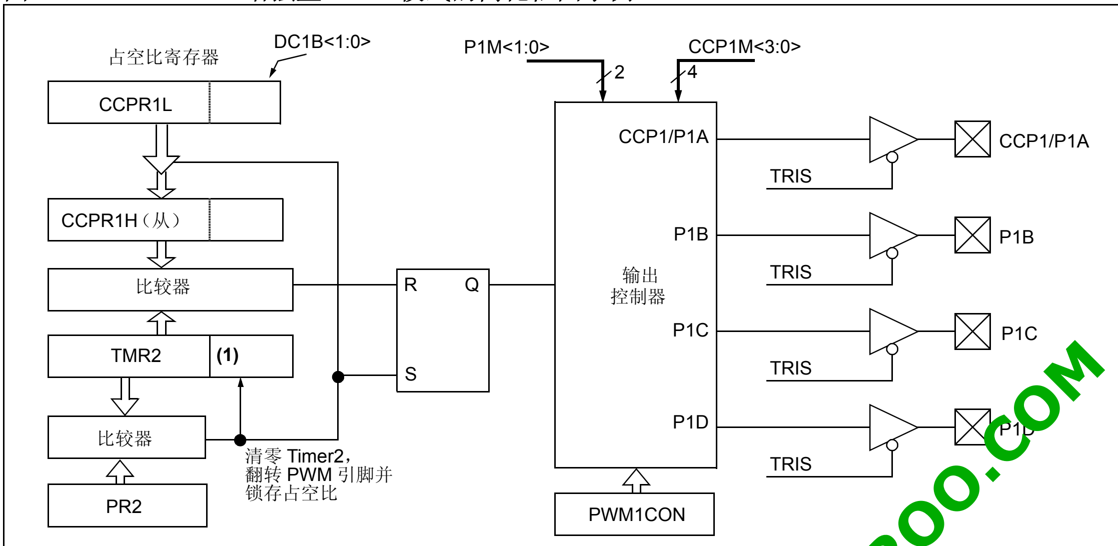
PWM 输出与 I/O 引脚复用，指定为 P1A、P1B、P1C 和 P1D。PWM 引脚的极性是可配置的，通过将 CCP1CON 寄存器中的 CCP1M 位适当置 1 来选择。

表 11-4 给出了每种增强型 PWM 模式的引脚分配。

图 11-5 给出了增强型 PWM 模块的简化框图的示例。

注：	为防止在第一次使能PWM时产生不完整的波形，ECCP模块在产生PWM信号前会等待直到新的PWM 周期开始。

图 11-5： 增强型 PWM 模式的简化框图示例



注 1： 8 位定时器 TMR2 寄存器与 2 位内部 Q 时钟或预分频器中的 2 位一起构成 10 位时基。

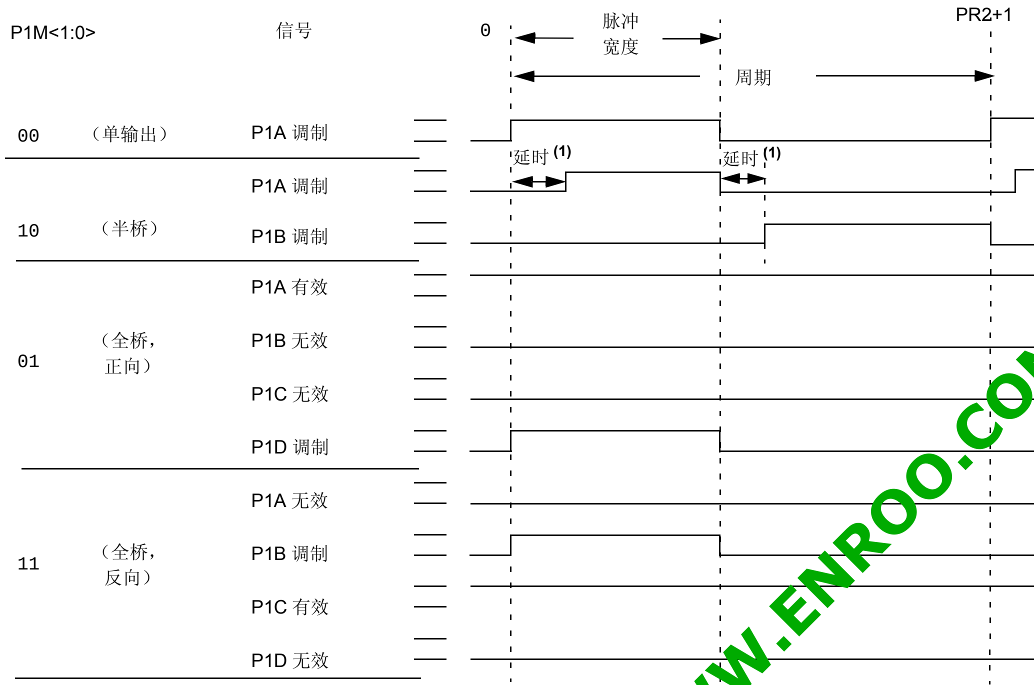
> 注 1： 每个 PWM 输出的 TRIS 寄存器值必须进行适当配置。
> 2： 清零 CCP1CON 寄存器将放弃对所有 PWM 输出引脚的 ECCP 控制。
> 3： 增强型 PWM 模式没有使用的任何引脚均可用于备用引脚功能。

表 11-4： 各种 PWM 增强型模式的引脚分配示例

ECCP 模式	P1M<1:0>	CCP1/P1A	P1B	P1C	P1D
单 PWM 模式	00	是 (1)	是 (1)	是 (1)	是 (1)
半桥	10	是	是	否	否
全桥，正向	01	是	是	是	是
全桥，反向	11	是	是	是	是

注 1： 脉冲转向使能单 PWM 模式下的输出。

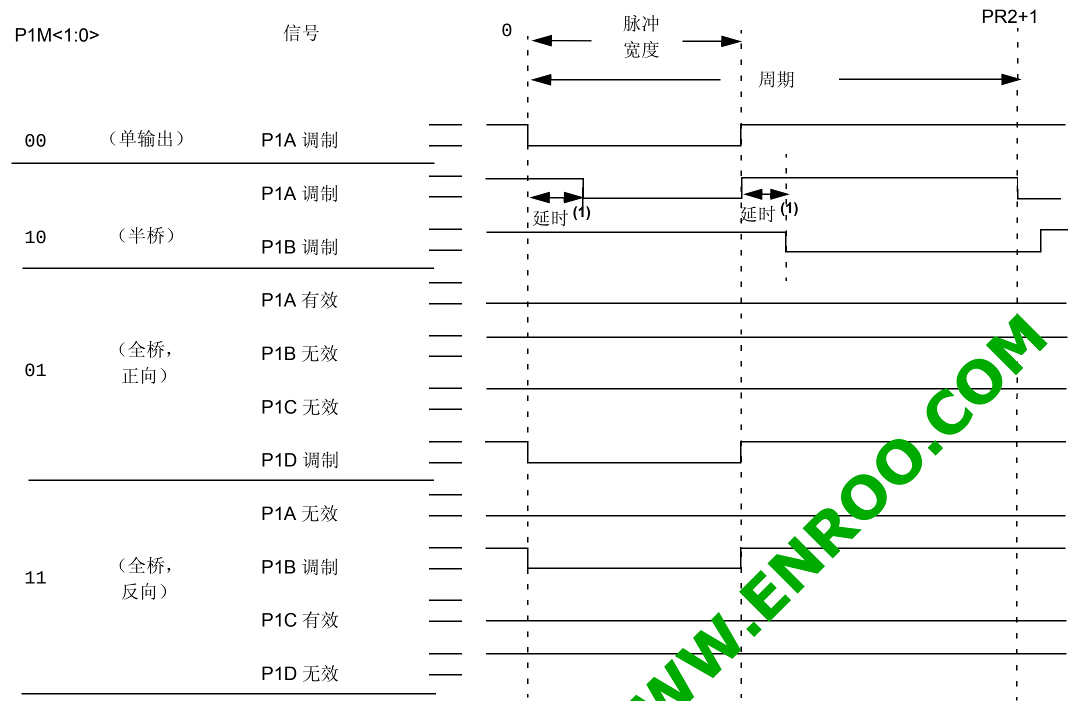
图 11-6： **PWM（增强型模式）输出关系示例（高电平有效状态）**



关系：
- 周期 = 4 * Tosc * (PR2 + 1) * (TMR2 预分频值)
- 脉冲宽度 = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 预分频值)
- 延时 = 4 * Tosc * (PWM1CON<6:0>)

注 **1：** 使用 PWM1CON 寄存器对死区延时编程（**第 11.4.6 节“可编程死区延时模式”**）。

图 11-7： 增强型 **PWM** 输出关系示例 （低电平有效状态）



关系：
- 周期 = 4 * Tosc * (PR2 + 1) * (TMR2 预分频值)
- 脉冲宽度 = Tosc * (CCPR1L<7:0>:CCP1CON<5:4>) * (TMR2 预分频值)
- 延时 = 4 * Tosc * (PWM1CON<6:0>)

注 1： 使用 PWM1CON 寄存器对死区延时编程（第 **11.4.6** 节 **“可编程死区延时模式”**）。

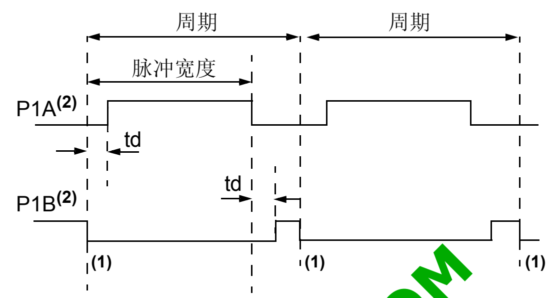
11.4.1 半桥模式

在半桥模式下，有两个引脚用作输出驱动推挽式负载。CCP1/P1A 引脚输出 PWM 输出信号，P1B 引脚输出互补的 PWM 输出信号（见图 11-6）。这种模式可用于半桥应用（如图 11-9 所示），或者用于全桥应用——这种情况下使用两个 PWM 信号调制 4 个功率开关。

在半桥模式下，可编程死区延时可用来防止半桥功率器件中流过直通（Shoot-through）电流。PWM1CON 寄存器的 PDC<6:0> 位的值设置在输出被驱动为有效之前的指令周期数。如果这个值比占空比大，则在整个周期中相应的输出保持为无效。关于死区延时操作的更多详细信息，请参见**第 11.4.6 节 “可编程死区延时模式”**。

由于 P1A 和 P1B 输出与端口数据锁存器是复用的，相关的TRIS位必须清零，从而将P1A和P1B配置为输出。

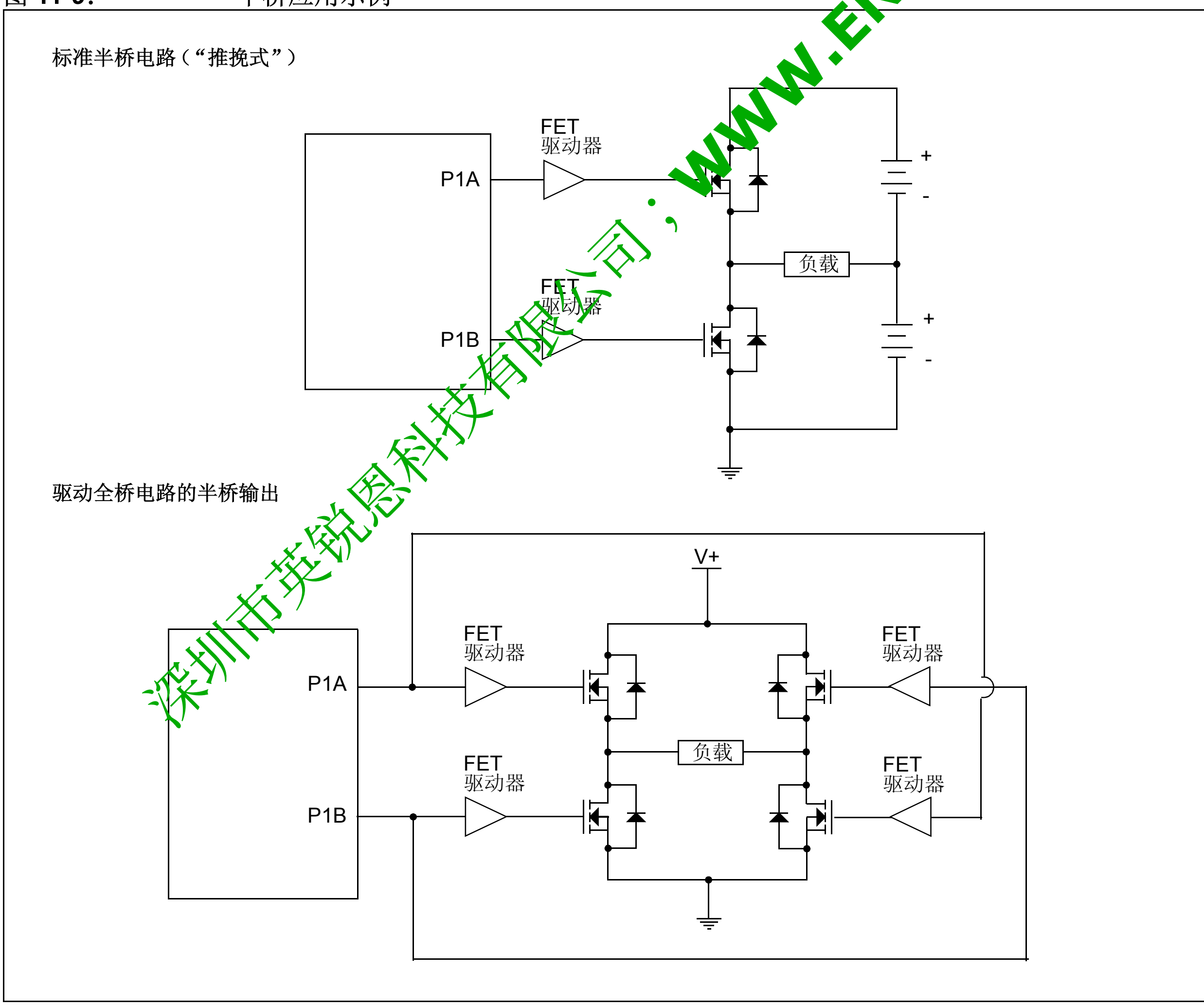
图 11-8： 半桥 **PWM** 输出示例



td = 死区延时

注 **1：** 此时 TMR2 寄存器和 PR2 寄存器的值相等。
2： 图中输出信号为高电平有效。

图 11-9： 半桥应用示例



11.4.2 全桥模式

在全桥模式下， 4 个引脚都用作输出。全桥应用的示例如图 11-10 所示。

在正向模式下，引脚 CCP1/P1A 被驱动为有效状态，引脚 P1D 被调制，而 P1B 和 P1C 将被驱动为无效状态，如图 11-11 所示。

在反向模式下，P1C 被驱动为有效状态，引脚 P1B 被调制，而 P1A 和 P1D 将被驱动为无效状态，如 图 11-11 所示。

P1A、P1B、P1C 和 P1D 输出与端口数据锁存器复用。相关的 TRIS 位必须清零，从而将 P1A、P1B、P1C 和 P1D 引脚配置为输出。

图 11-10： 全桥应用示例

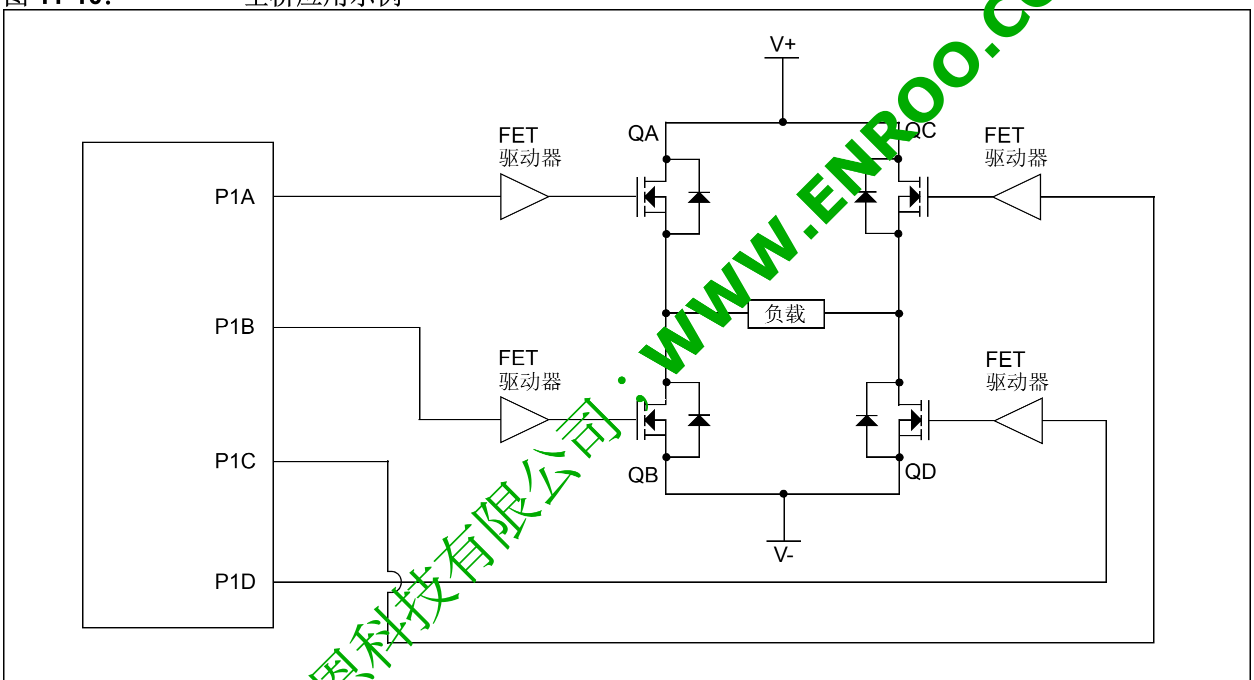
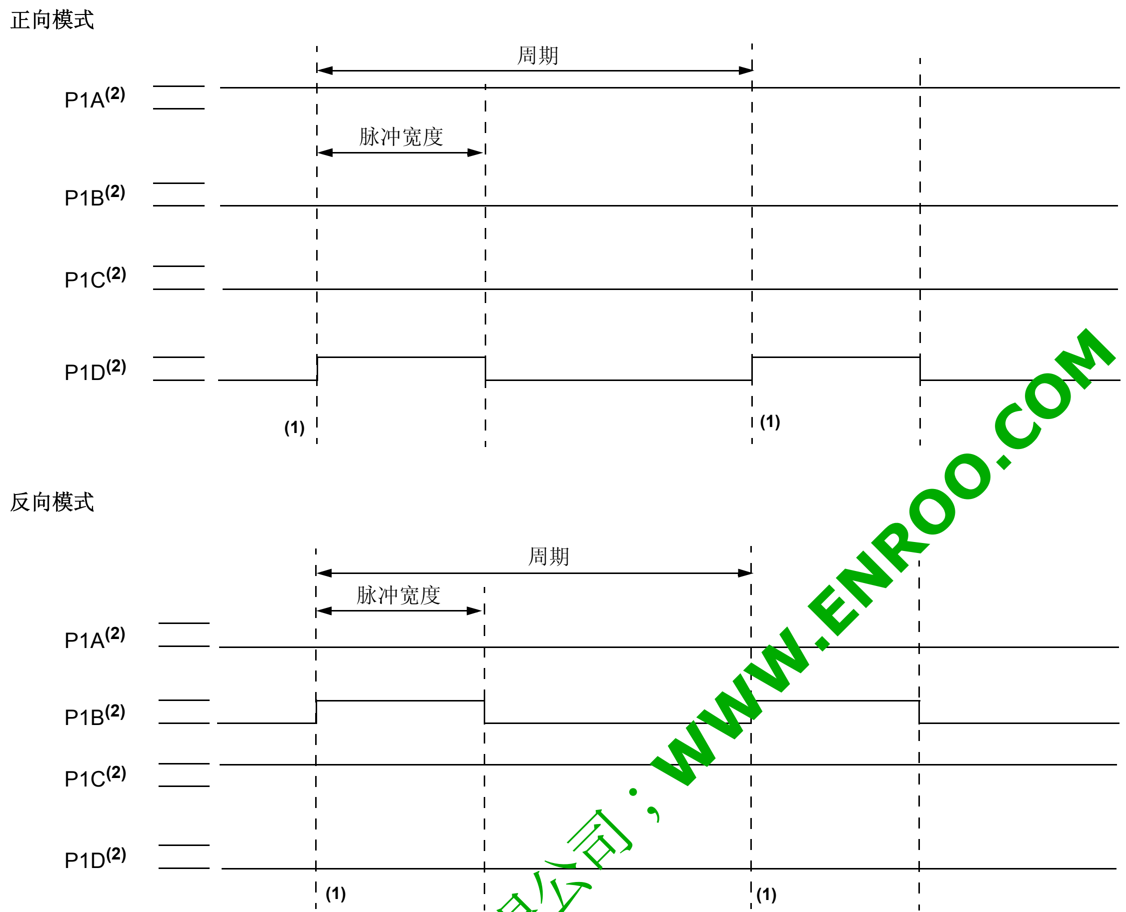


图 11-11： 全桥 PWM 输出示例



注 1： 此时 TMR2 寄存器和 PR2 寄存器的值相等。
2： 图中输出信号为高电平有效。

11.4.2.1 全桥模式中的方向改变

在全桥模式下，CCP1CON 寄存器中的 P1M1 位允许用户控制正 / 反方向。当应用软件改变这个方向控制位时，模块将在下一个 PWM 周期改用新的方向。

通过改变 CCP1CON 寄存器的 P1M1 位，可以用软件启动方向改变。以下序列在当前 PWM 周期结束前发生：

- 调制输出 （P1B 和 P1D）进入无效状态。
- 相关的未调制输出 （P1A 和 P1C）被切换到以相反的方向驱动。
- PWM 调制在下一个周期开始继续。

关于该序列的说明，请参见图 11-12。

全桥模式不提供死区延时。因为始终只有一个输出被调制，所以通常不需要死区延时。有一种情况需要死区延时。这一情况发生在以下两个条件同时满足时：

1. 当输出的占空比达到或者接近 100%，PWM 输出方向改变。
2. 功率开关（包括功率器件和驱动电路）的关断时间比导通时间要长。

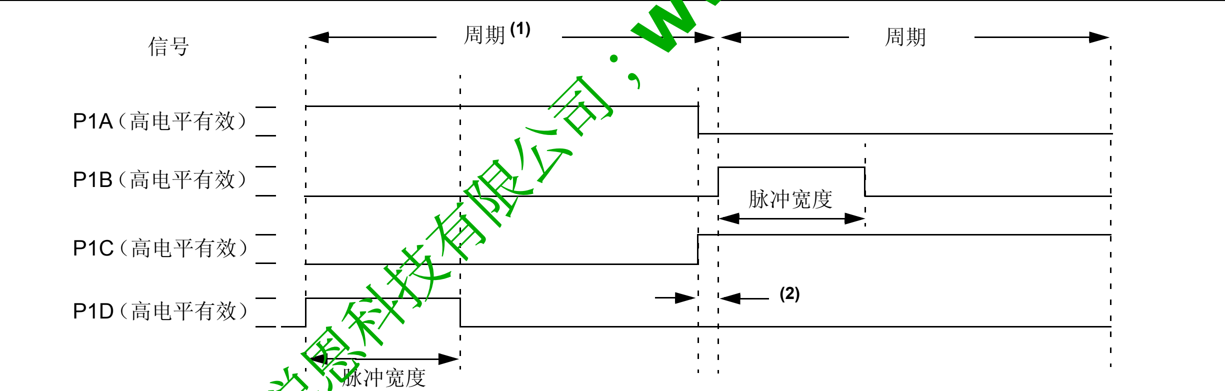
在图 11-13 所示的例子中，在占空比接近 100% 时，PWM 方向从正向改变到反向。在这个示例中，在时间 t1，输出 P1A 和 P1D 变为无效，而输出 P1C 变为有效。因为功率器件的关断时间比导通时间要长，在 “t” 时间内，功率器件 QC 和 QD 中可能流过直通电流（shoot-through current）（见图 11-13）。当 PWM 方向从反向改变到正向时，功率器件 QA 和 QB 也将出现相同的现象。

如果应用中需要在高占空比时改变 PWM 方向，避免直通电流的两种可能方法是：

1. 在改变方向之前的一个 PWM 周期降低 PWM 占空比。
2. 使用开关驱动电路，使开关管的关断时间比导通时间短。

也可能存在其他避免直通电流的方案。

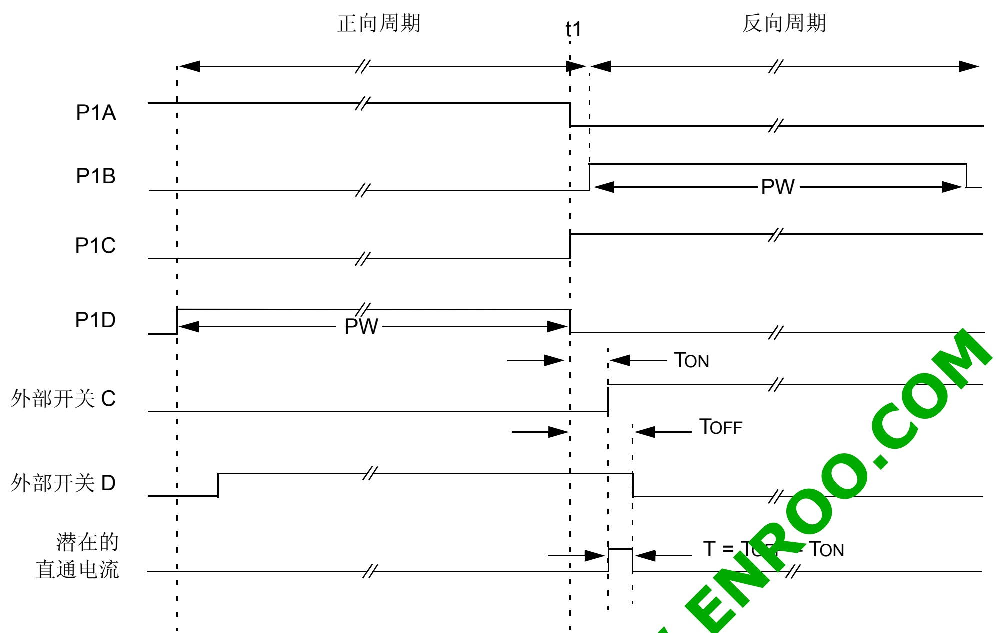
图 11-12： PWM 方向改变的示例



注 1： 可在 PWM 周期的任意时刻，写 CCP1CON 寄存器的方向位 P1M1。

2： 当改变方向时， P1A 和 P1C 信号在当前 PWM 周期结束之前切换。此时被调制的 P1B 和 P1D 信号是无效的。该时间长度为 (1/Fosc) • TMR2 预分频值。

图 11-13： 在占空比接近 100% 时改变 PWM 方向的示例



注 1： 图中所有信号均为高电平有效。
2： TON 是功率开关 QC 及其驱动电路的导通延时。
3： TOFF 是功率开关 QD 及其驱动电路的关断延时。

11.4.3 启动注意事项

当使用 PWM 模式时，应用硬件必须在 PWM 输出引脚上外接适当的上拉和 / 或下拉电阻。

> **注：** 当单片机退出复位状态时，所有 I/O 引脚呈高阻态。外部电路必须保持功率开关器件处于关断状态，直到单片机将 I/O 引脚驱动为适当的信号电平，或者激活 PWM 输出为止。

CCP1CON 寄存器的 CCP1M<1:0> 位允许用户为每一对 PWM 输出引脚 （P1A/P1C 和 P1B/P1D）选择 PWM 输出信号为高电平有效还是低电平有效。PWM 输出极性必须在使能 PWM 引脚输出驱动器之前选择。建议不要在使能 PWM 引脚输出驱动器时改变极性配置，因为这可能导致应用电路损坏。

当 PWM 模块初始化时，P1A、P1B、P1C 和 P1D 输出锁存器可能不在正确的状态。这样在配置为增强型 PWM 模式的同时使能 PWM 引脚输出驱动器，可能损坏应用电路。应首先将增强型 PWM 模式配置为正确的输出模式并经过一个完整的 PWM 周期之后，再使能 PWM 引脚输出驱动器。当第二个 PWM 周期开始时，PIR1 寄存器的 TMR2IF 位置 1 表明一个完整的 PWM 周期结束了。

11.4.4 增强型 PWM 自动关闭模式

PWM模式支持自动关闭模式，当外部关闭事件发生时将禁止 PWM 输出。自动关闭模式将 PWM 输出引脚置于预先确定的状态。该模式用于防止 PWM 损坏应用。

通过使用 ECCPAS 寄存器的 ECCPASx 位来选择自动关闭源。关闭事件由以下条件产生：

- INT 引脚上的逻辑 0
- 比较器 C1
- 比较器 C2
- 用固件将 ECCPASE 位置 1

关闭条件由 ECCPAS 寄存器的 ECCPASE （自动关闭事件状态）位指定。如果该位为 0， PWM 引脚正常工作。如果该位为 1， PWM 输出处于关闭状态。

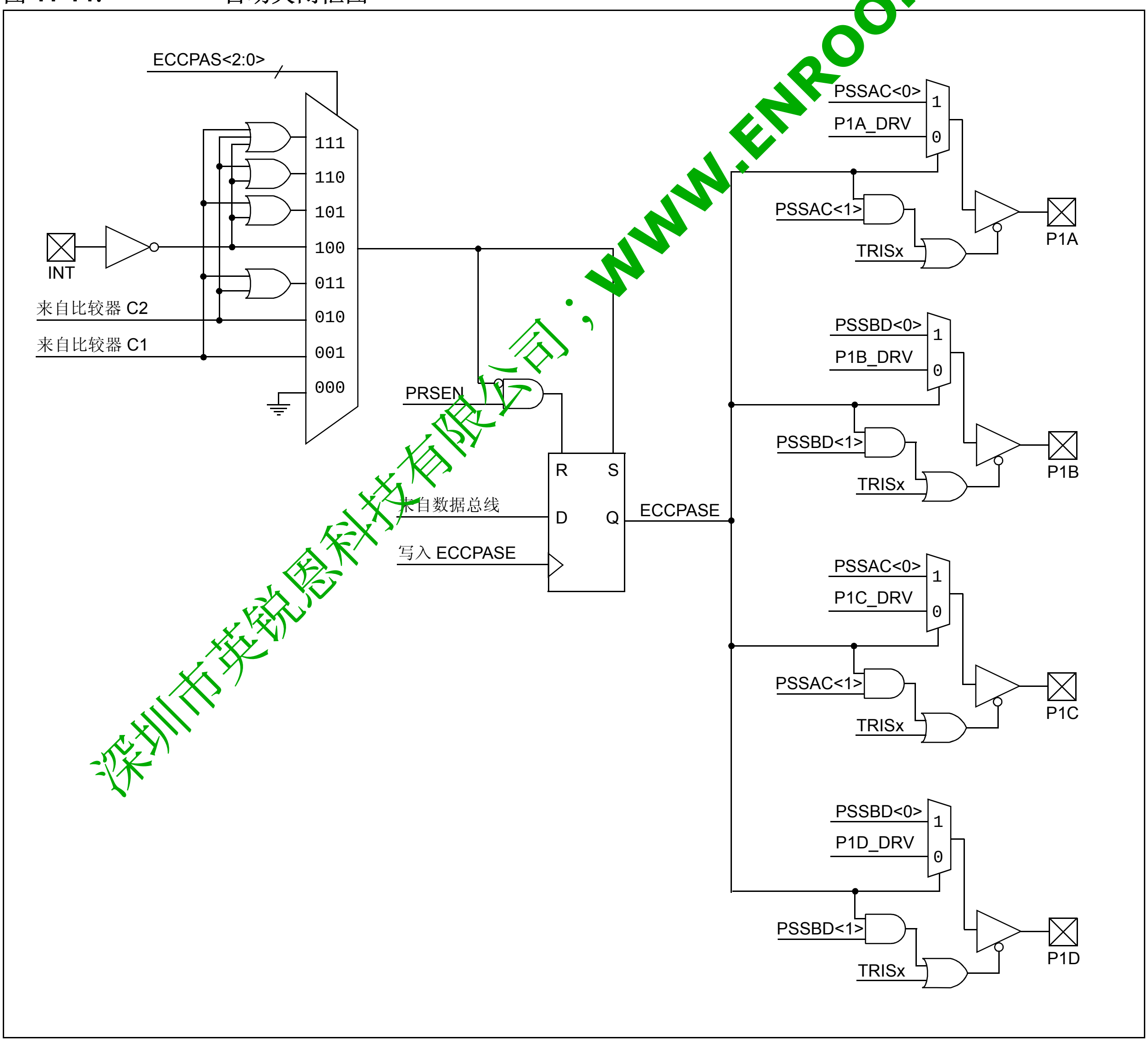
当关闭事件发生时，会发生以下两件事：

ECCPASE 位被设置为 1。 ECCPASE 将保持置 1 直到由固件清零或发生自动重启（见第 **11.4.5** 节 **“自动重启模式”**）。

使能的 PWM 引脚被异步置为其关闭状态。PWM 输出引脚被分组为 [P1A/P1C] 和 [P1B/P1D] 对。每对引脚的状态由 ECCPAS 寄存器的 PSSAC 和 PSSBD 位决定。每对引脚可以设置为以下 3 种状态之一：

- 驱动逻辑 1
- 驱动逻辑 0
- 三态 （高阻态）

图 11-14： 自动关闭框图



寄存器 11-2： ECCPAS：增强型捕捉 / 比较 /PWM 自动关闭控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **ECCPASE：**ECCP 自动关闭事件状态位
1 = 发生了关闭事件； ECCP 输出为关闭状态
0 = ECCP 输出正常工作

bit 6-4 **ECCPAS<2:0>：**ECCP 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
010 = 比较器 C2 输出高电平 (1)
011 = 比较器 C1 或 C2 输出高电平
100 = INT 引脚电压为 V_{IL}
101 = INT 引脚电压为 V_{IL} 或比较器 C1 输出高电平
110 = INT 引脚电压为 V_{IL} 或比较器 C2 输出高电平
111 = INT 引脚电压为 V_{IL} 或比较器 C1 或 C2 输出高电平

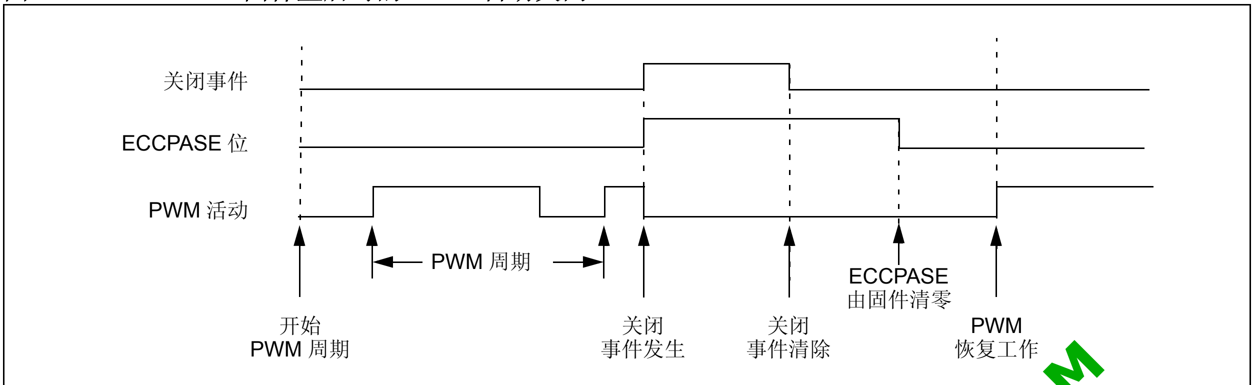
bit 3-2 **PSSACn：**引脚 P1A 和 P1C 关闭状态控制位
00 = 驱动引脚 P1A 和 P1C 为 0
01 = 驱动引脚 P1A 和 P1C 为 1
1x = 引脚 P1A 和 P1C 为三态

bit 1-0 **PSSBDn：**引脚 P1B 和 P1D 关闭状态控制位
00 = 驱动引脚 P1B 和 P1D 为 0
01 = 驱动引脚 P1B 和 P1D 为 1
1x = 引脚 P1B 和 P1D 为三态

注 1： 如果 C2SYNC 使能， Timer1 将使关闭延时。

> 注 1： 自动关闭条件是基于电平的信号，而不是基于边沿的信号。只要电平存在，自动关闭就将持续。
> 2： 当自动关闭条件持续时，禁止写 ECCPASE 位。
> 3： 一旦自动关闭条件被移除并且 PWM 重启（通过固件或自动重启），PWM 信号将总是在下一个 PWM 周期重启。

图 11-15：　　固件重启时的 **PWM** 自动关闭 （**PRSEN =** `0`）

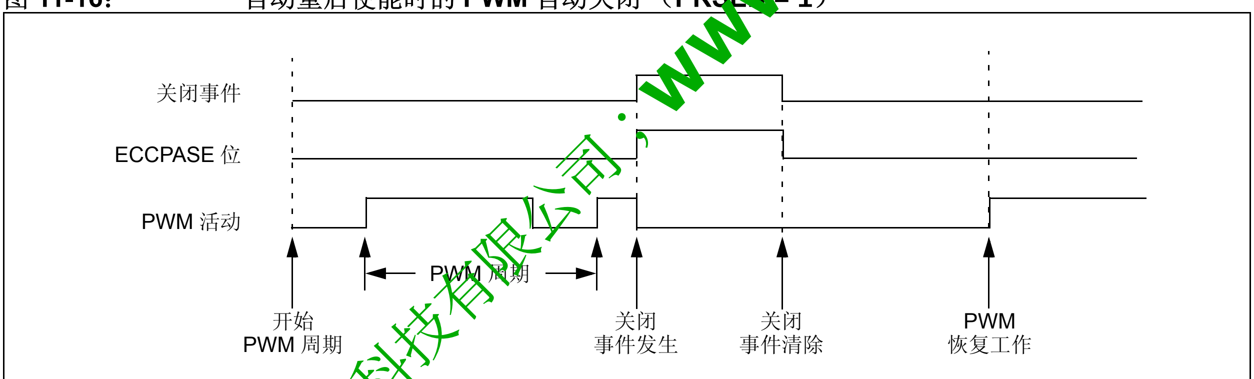


11.4.5　　自动重启模式

一旦自动关闭条件被移除，增强型 PWM 可被配置为自动重启 PWM 信号。通过将 PWM1CON 寄存器中的 PRSEN 位置 1 使能自动重启。

如果使能了自动重启，只要自动关闭条件有效，ECCPASE 位将保持置 1。当自动关闭条件被移除时，ECCPASE 位将由硬件清零并继续正常工作。

图 11-16：　　自动重启使能时的 **PWM** 自动关闭 （**PRSEN =** `1`）

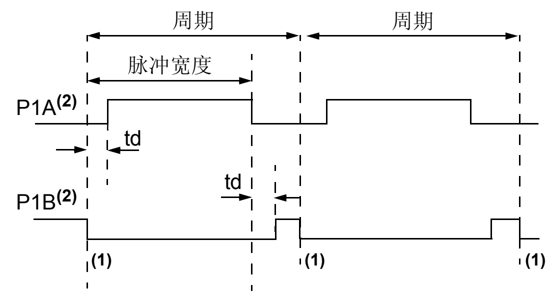


11.4.6 可编程死区延时模式

在所有功率开关管都以 PWM 频率调制的半桥应用中，功率开关管关断通常比导通需要更多的时间。如果上下两个功率开关管在同一时间开关（一个导通，另一个关断），那么在一段很短的时间里，两个开关管可能同时导通，直到一个开关管完全关断为止。在这短暂的时间中，两个功率开关管中可能流过较高的电流（*直通电流*），将逆变桥的电源与地短路。为避免开关过程中可能会出现的破坏性直通电流，通常需要延迟功率开关管的导通，保证在另一个开关管完全关断之后，再导通相应的功率开关管。

在半桥模式下，可采用数字可编程死区延时来避免出现损坏逆变桥功率开关管的直通电流。在信号从无效状态切换到有效状态时增加延时。请参见图 11-8。PWM1CON 寄存器（寄存器 11-3）的低 7 位以单片机指令周期（TCY 或 4 TOSC）为单位设置延时。

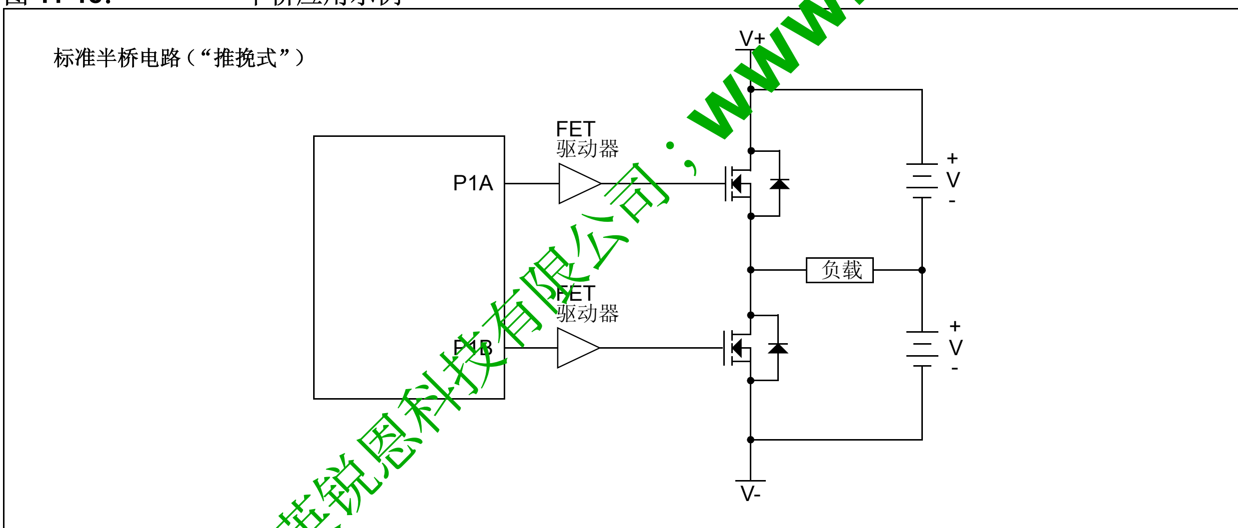
图 11-17： 半桥 PWM 输出示例



td = 死区延时

注 1：此时 TMR2 寄存器和 PR2 寄存器的值相等。
2：图中输出信号为高电平有效。

图 11-18： 半桥应用示例



寄存器 11-3： PWM1CON：增强型 PWM 控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **PRSEN：** PWM 重启使能位

1 = 自动关闭时，一旦关闭事件消失， ECCPASE 位自动清零； PWM 自动重启

0 = 自动关闭时， ECCPASE 必须用软件清零以重启 PWM

bit 6-0 **PDC<6:0>：** PWM 延时计数位

PDCn = 在 PWM 信号应该转换为有效的预定时间和转换为有效的实际时间之间的 Fosc/4 (4 * Tosc) 周期数

11.4.7 脉冲转向模式

在单输出模式下，脉冲转向使任一 PWM 引脚均可输出调制信号。此外，同一 PWM 信号可在多个引脚上同时输出。

一旦选择了单输出模式（CCP1CON 寄存器的 CCP1M<3:2> = 11 且 P1M<1:0> = 00），用户固件即可通过将 PSTRCON 寄存器的相应 STR<D:A> 位置 1，将同一 PWM 信号输出到 1、2、3 或 4 个输出引脚，如图 11-19 所示。

注：	必须将相关的 TRIS 位设置为输出（0）使能引脚输出驱动器，才能将 PWM 信号输出到引脚上。

当 PWM 转向模式有效时，CCP1CON 寄存器的 CCP1M<1:0> 位为 P1<D:A> 引脚选择 PWM 输出极性。

PWM 自动关闭操作也适用于**第 11.4.4 节"增强型 PWM 自动关闭模式"**中所述的 PWM 转向模式。自动关闭事件只会影响使能 PWM 输出的引脚。

寄存器 11-4： PSTRCON：脉冲转向控制寄存器 [(1)]

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	STRSYNC	STRD	STRC	STRB	STRA
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 **未实现：**读为 0

bit 4 **STRSYNC：**转向同步位
1 = 在下一个 PWM 周期发生输出转向更新
0 = 在指令周期边界的开始发生输出转向更新

bit 3 **STRD：**转向使能位 D
1 = P1D 引脚输出 PWM 波形，其极性由 CCP1M<1:0> 控制
0 = P1D 引脚被分配为端口引脚

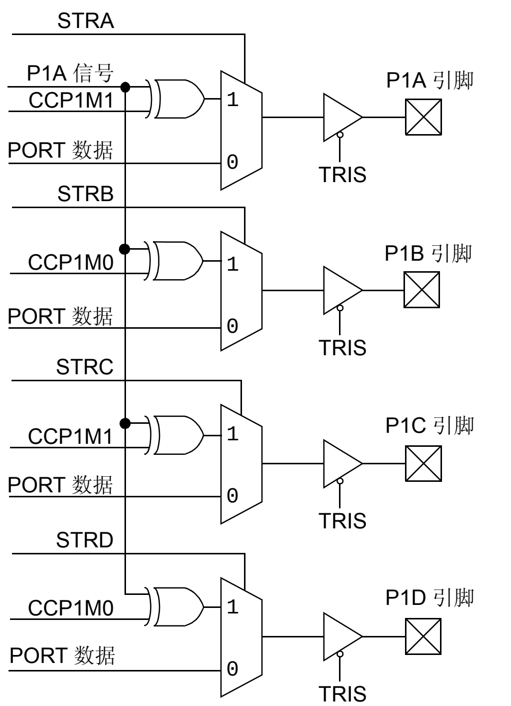
bit 2 **STRC：**转向使能位 C
1 = P1C 引脚输出 PWM 波形，其极性由 CCP1M<1:0> 控制
0 = P1C 引脚被分配为端口引脚

bit 1 **STRB：**转向使能位 B
1 = P1B 引脚输出 PWM 波形，其极性由 CCP1M<1:0> 控制
0 = P1B 引脚被分配为端口引脚

bit 0 **STRA：**转向使能位 A
1 = P1A 引脚输出 PWM 波形，其极性由 CCP1M<1:0> 控制
0 = P1A 引脚被分配为端口引脚

注 **1：** 只有在 CCP1CON 寄存器位 CCP1M<3:2> = 11 且 P1M<1:0> = 00 时 PWM 转向模式才可用。

图 11-19： 转向的简化框图



注 1： CCP1CON 寄存器位 P1M<1:0> = 00 且 CCP1M<3:2> = 11 时，端口输出配置如上所示。

2： 单 PWM 输出要求至少将一个 STRx 位置 1。

11.4.7.1 转向同步

PSTRCON 寄存器的 STRSYNC 位为用户提供了转向事件何时发生的两种选择。当 STRSYNC 位为 0 时，转向事件在写入 PSTRCON 寄存器的指令结束时发生。此时， P1<D:A> 引脚的输出信号可能是一个不完整的 PWM 波形。用户固件需要立即从引脚移除 PWM 信号时，该操作非常有用。

当 STRSYNC 位为 1 时，在下一个 PWM 周期的开始将发生有效的转向更新。此时，开 / 关转向 PWM 输出将始终产生一个完整的 PWM 波形。

图 11-20 和 11-21 显示了根据 STRSYNC 设置进行 PWM 转向的时序图。

图 11-20： 指令结束时发生的转向事件的示例 （STRSYNC = 0）

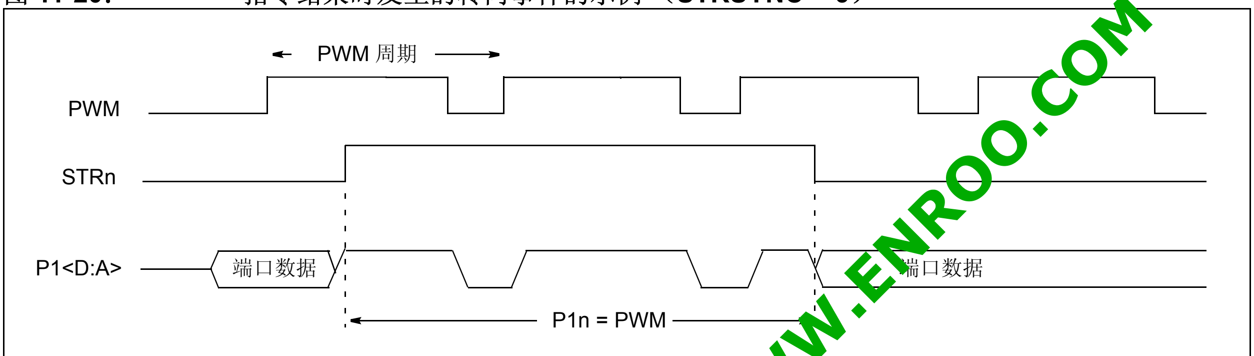


图 11-21： 指令开始时发生的转向事件的示例 （STRSYNC = 1）

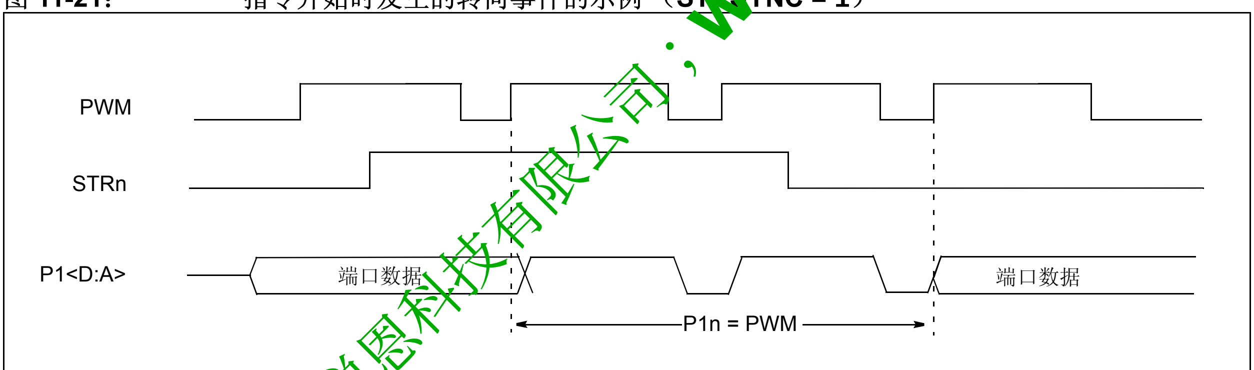


表 11-5： 与捕捉、比较和 PWM 相关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
CCP1CON	P1M1	P1M0	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	0000 0000	0000 0000
CM1CON0	C1ON	C1OUT	C1OE	C1POL	—	C1R	C1CH1	C1CH0	0000 -000	0000 -000
CM2CON0	C2ON	C2OUT	C2OE	C2POL	—	C2R	C2CH1	C2CH0	0000 -000	0000 -000
CM2CON1	MC1OUT	MC2OUT	—	—	—	—	T1GSS	C2SYNC	00-- --10	00-- --10
CCPR1L	捕捉 / 比较 /PWM 寄存器 1 的低字节								xxxx xxxx	uuuu uuuu
CCPR1H	捕捉 / 比较 /PWM 寄存器 1 的高字节								xxxx xxxx	uuuu uuuu
ECCPAS	ECCPASE	ECCPAS2	ECCPAS1	ECCPAS0	PSSAC1	PSSAC0	PSSBD1	PSSBD0	0000 0000	0000 0000
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 0000	0000 0000
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PSTRCON	—	—	—	STRSYNC	STRD	STRC	STRB	STRA	---0 0001	---0 0001
PWM1CON	PRSEN	PDC6	PDC5	PDC4	PDC3	PDC2	PDC1	PDC0	0000 0000	0000 0000
T1CON	T1GINV	TMR1GE	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0000 0000	uuuu uuuu
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
TMR1L	16 位 TMR1 寄存器低字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR1H	16 位 TMR1 寄存器高字节的保持寄存器								xxxx xxxx	uuuu uuuu
TMR2	Timer2 模块寄存器								0000 0000	0000 0000
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知。捕捉、比较和 PWM 不使用阴影单元。

12.0 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（Enhanced Universal Synchronous Asynchronous Receiver Transmitter，EUSART）模块是一种串行 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等。EUSART 也是一种串行通信接口（Serial Communications Interface，SCI），可配置为全双工异步系统或半双工同步系统。全双工模式可用来与外设通信，如 CRT 终端和个人电脑。半双工同步模式用于与外设器件通信，如 A/D 或 D/A 集成电路、串行 EEPROM 或其他单片机。这些器件通常不具备用以产生波特率的内部时钟，并需要由主同步器件提供外部时钟信号。

EUSART 模块具备以下功能：

- 全双工异步收发
- 双字符输入缓冲区
- 单字符输出缓冲区
- 可编程 8 位或 9 位字符长度
- 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
- 半双工同步从
- 同步模式下的可编程时钟极性
- 休眠工作

EUSART 模块还具备以下特性，使其成为局部互联网（Local Interconnect Network，LIN）总线系统的理想选择：

- 自动检测和波特率校准
- 接收间隔（Break）字符时唤醒
- 13 位间隔字符发送

EUSART 收发器的框图如图 12-1 和图 12-2 所示。

图 12-1： EUSART 发送框图

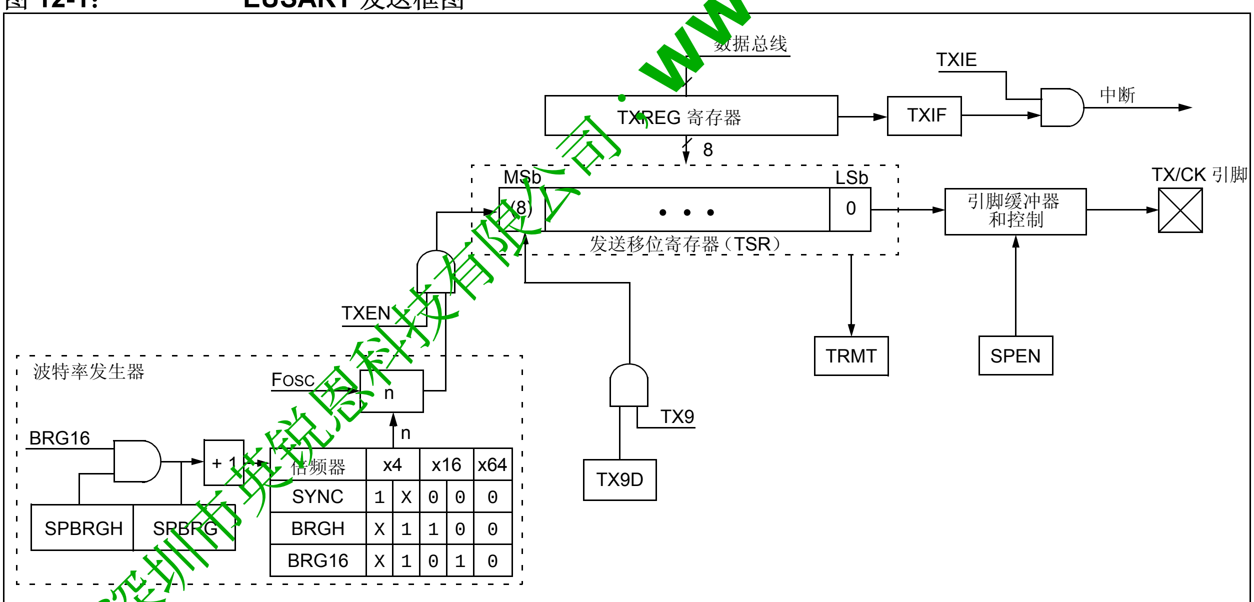
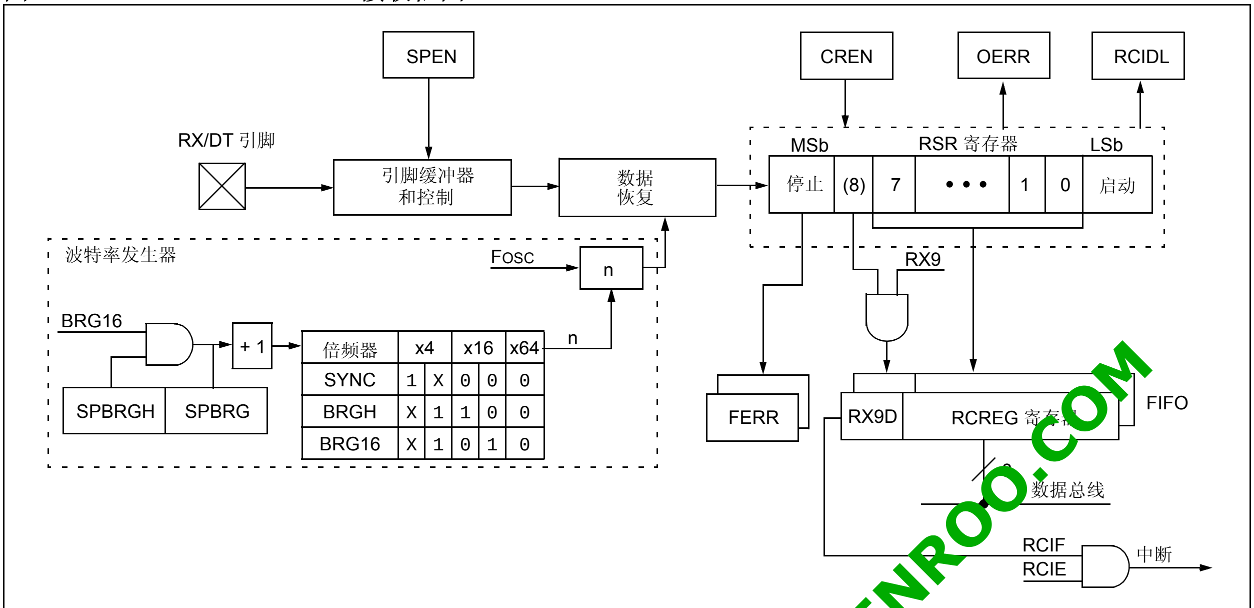


图 12-2： EUSART 接收框图



EUSART 模块的操作由以下 3 个寄存器控制：

- 发送状态和控制 （TXSTA）
- 接收状态和控制 （RCSTA）
- 波特率控制 （BAUDCTL）

这些寄存器的详细信息请分别参见寄存器 12-1、寄存器 12-2 和寄存器 12-3。

12.1 EUSART 异步模式

EUSART 采用标准不归零（non-return-to-zero，NRZ）格式发送和接收数据。NRZ 实现为两种电平：表示数据位 1 的 V_{OH} 标记状态（mark state），以及表示数据位 0 的 V_{OL} 空格状态（space state）。NRZ 指的是当具有相同值的连续数据位被发送时，它们保持在该位的输出电平不变，而不会在每个位发送之间回到中性电平。NRZ 发送端口在标记状态空闲。每个字符发送包含 1 个启动位及随后的 8 个或 9 个数据位，并始终由 1 个或多个停止位结束。启动位始终是一个空格，停止位始终是标记。最常见的数据格式为 8 位。每个发送位保持 1/(波特率) 个周期。使用片上专用 8 位 /16 位波特率发生器（Baud Rate Generator，BRG）从系统振荡器产生标准波特率频率。波特率配置示例请参见表 12-5。

EUSART 先发送和接收 LSb。EUSART 的发送器和接收器在功能上是相互独立的，但它们的数据格式和波特率相同。硬件不支持奇偶校验，但可通过软件实现并作为第 9 位存储。

12.1.1 EUSART 异步发送器

图 12-1 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），该寄存器不可用软件直接访问。TSR 从发送缓冲区 TXREG 取得数据。

12.1.1.1 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 使能 EUSART 的发送器电路。清零 TXSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 使能 EUSART 并自动将 TX/CK I/O 引脚配置为输出。如果 TX/CK 引脚与模拟外设共用，那么必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

注	
1：	当 SPEN 位置 1 时，RX/DT I/O 引脚被自动配置为输入，无论相应 TRIS 位的状态如何以及 EUSART 接收器是否被使能。通过正常的 PORT 读取操作可读取 RX/DT 引脚的数据，但不包括 PORT 锁存器的数据输出值。
2：	TXEN 允许位置 1 时，TXIF 发送器中断标志置 1。

12.1.1.2 发送数据

向 TXREG 寄存器写入一个字符时启动发送。如果这是首字符，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 中仍保存前一个字符的全部或部分，则新字符被保存在 TXREG 中，直到前一个字符的停止位被发送。之后，在 TXREG 中等待的字符在停止位发送后 1 个 T_{CY} 后被传送到 TSR 中。TXREG 中的数据被传送到 TSR 后，启动位、数据位和停止位的发送序列立即开始。

12.1.1.3 发送中断标志

只要 EUSART 发送器被使能，而且 TXREG 中没有等待发送的字符，PIR1 寄存器的 TXIF 中断标志位就被置 1。换句话说，只有在 TSR 中有字符，并且 TXREG 中还有在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。执行写操作后的第二个指令周期 TXIF 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，TXIF 标志位就被置 1，无论 TXIE 允许位的状态如何。

要在发送数据时使用中断，应只在没有数据可发送时才将 TXIE 位置 1。在将发送的最后一个字符写入 TXREG 后应清零 TXIE 中断允许位。

12.1.1.4 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，而当一个字符从 TXREG 寄存器传送到 TSR 寄存器中时，该位清零。TRMT 位将保持清零，直到所有位移出 TSR 寄存器。该位不与任何中断逻辑相连，因此用户必须查询该位以确定 TSR 的状态。

注：	TSR 寄存器不映射到数据存储器中，因此用户无法使用。

12.1.1.5 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， TX9D 数据位必须先于 8 个最低有效位写入 TXREG。写入 TXREG 后，所有 9 个位将被立即传送到 TSR 移位寄存器中。

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 12.1.2.7 节“地址检测”**。

12.1.1.6 异步发送设置：

1. 初始化 SPBRGH、SPBRG 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见**第 12.3 节“EUSART 波特率发生器（BRG）”**）。
2. 清零 SYNC 位并将 SPEN 位置 1，使能异步串行口。
3. 如果需要 9 位发送，将 TX9 控制位置 1。如果第 9 个数据位置 1，则表示发送器置于地址检测时，8 个最低有效位为地址。
4. 将 TXEN 控制位置 1 使能发送。这将导致 TXIF 中断位置 1。
5. 如果需要中断，将 PIE1 寄存器中的 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。
6. 如果选择了 9 位发送，应将第 9 位装入 TX9D 数据位。
7. 将 8 位数据装入 TXREG 寄存器。这将启动发送。

图 12-3： 异步发送

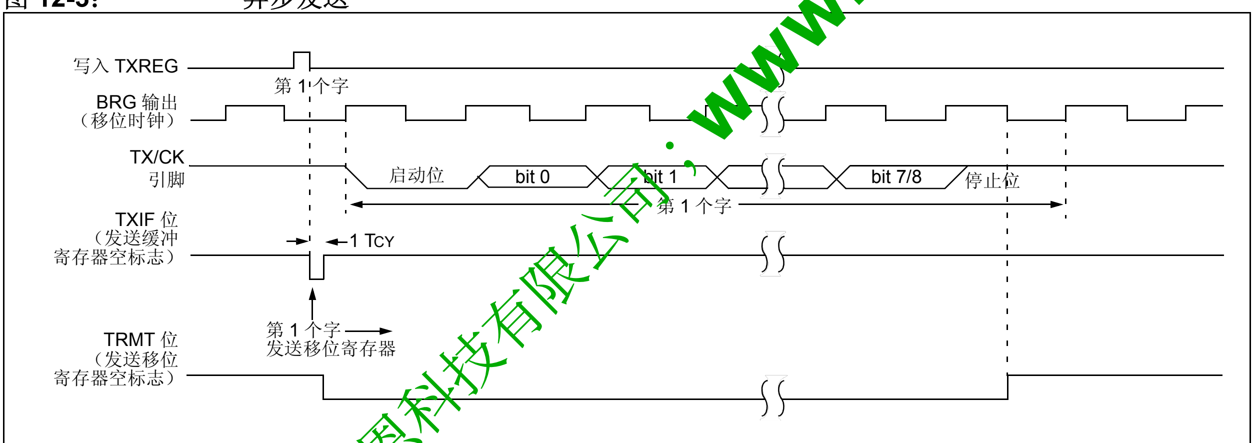
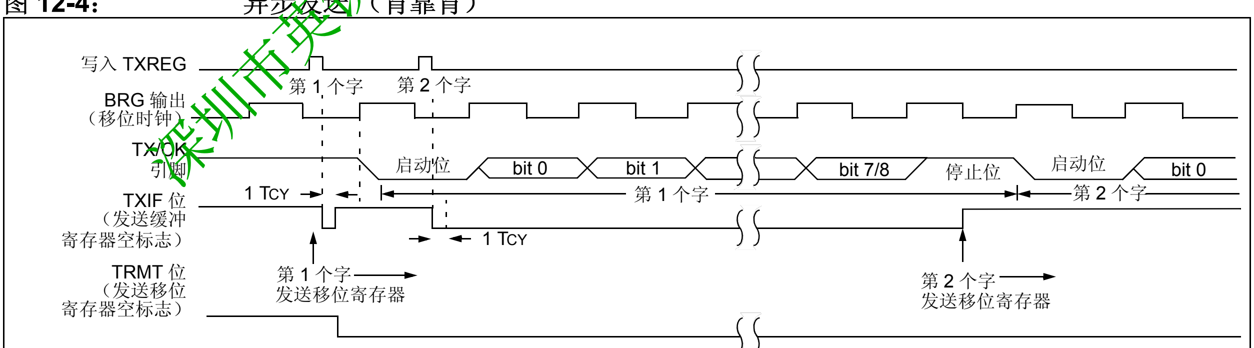


图 12-4： 异步发送（背靠背）



注： 此时序图表示的是两次连续的发送。

表 **12-1**： 与异步发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， - = 未实现 （读为 0）。异步发送不使用阴影单元。

12.1.2 EUSART 异步接收器

异步模式通常用于 RS-232 系统中。图 12-2 给出了接收器框图。数据在 RX/DT 引脚上接收并驱动数据恢复模块。数据恢复模块实际上是一个高速移位器，工作频率为 16 倍波特率，而串行接收移位器（Receive Shift Register， RSR）的工作频率为比特率。所有 8 位或 9 位字符移入后被立即传送到双字符的先进先出（First-In-First-Out， FIFO）存储器中。FIFO 缓冲区允许先接收两个完整字符和第三个字符的开始部分后，再启动软件服务 EUSART 接收器。 FIFO 和 RSR 寄存器不能直接用软件访问。通过 RCREG 寄存器访问接收数据。

12.1.2.1 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 使能 EUSART 的接收器电路。清零 TXSTA 寄存器的 SYNC 位将 EUSART 配置为异步操作。将 RCSTA 寄存器的 SPEN 位置 1 使能 EUSART 并自动将 RX/DT I/O 引脚配置为输入。如果 RX/DT 引脚与模拟外设共用，那么必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

注：	当 SPEN 位置 1 时， TX/CK I/O 引脚被自动配置为输出，无论相应 TRIS 位的状态如何以及 EUSART 发送器是否被使能。PORT 锁存器与输出驱动器断开，因此不可能将 TX/CK 引脚用作通用输出。

12.1.2.2 接收数据

接收器的数据恢复电路在第一位的下降沿启动字符接收。第一位也称启动（Start）位，始终为零。数据恢复电路计数一个半位时间至启动位的中点并验证该位是否仍为零。如果该位非零则数据恢复电路中止字符接收，不产生错误，并恢复寻找启动位的下降沿。如果启动位被验证为零，则数据恢复电路计数一整个位时间至下一个位的中点。该位被一个择多检测电路（majority detect circuit）采样，其结果（0 或 1）被移入 RSR。重复此过程直到所有数据位均被采样并移入RSR。最后一个位时间被测量且其电平被采样。此为停止（Stop）位，始终为 1。如果数据恢复电路在停止位处采样到 0，则产生此字符的帧错误，否则此字符的帧错误被清零。关于帧错误的更多信息，请参见**第 12.1.2.4 节 "接收帧错误"**。

所有数据位和停止位被接收后，RSR 中的字符就被立即传送到 EUSART 接收 FIFO，且 PIR1 寄存器的 RCIF 中断标志位被置 1。读取 RCREG 寄存器时，FIFO 中顶部的字符被送出 FIFO。

注：	如果接收 FIFO 溢出，在溢出条件被清除前不会接收更多字符。关于溢出错误的更多信息，请参见**第 12.1.2.5 节 "接收溢出错误"**。

12.1.2.3 接收中断

只要 EUSART 接收器被使能且接收 FIFO 中存在未被读取的字符， PIR1 寄存器的 RCIF 中断标志位就会被置 1。RCIF 中断标志位是只读位，不能用软件置 1 或清零。

将以下所有位置 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 中存在未被读取的字符时，无论中断允许位的状态如何， RCIF 中断标志位均会被置 1。

12.1.2.4 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见到停止位。通过 RCSTA 寄存器的 FERR 位可访问帧错误状态。 FERR 位表示接收 FIFO 中顶部的未读字符的状态。因此，在读取 RCREG 前必须读出 FERR 位。

FERR 位是只读位，只用于接收 FIFO 中顶部的未读字符。帧错误 （FERR = 1）不排除接收额外字符。此时不必将 FERR 位清零。从 FIFO 缓冲区读出下一个字符将使 FIFO 进入下一个字符和下一个相应的帧错误。

将 RCSTA 寄存器的 SPEN 位清零可复位 EUSART，这样就可将FERR位强制清零。将RCSTA寄存器的CREN位清零不影响 FERR 位。自身产生的帧错误不会产生中断。

注：	如果接收FIFO中的所有接收字符均有帧错误，反复读取 RCREG 不会将 FERR 位清零。

12.1.2.5 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在访问 FIFO 前接收到完整的第三个字符时会产生溢出错误。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，但溢出错误被清除前不能再接收其他字符。将 RCSTA 寄存器的 CREN 位清零或通过将 RCSTA 寄存器的 SPEN 位清零复位 EUSART，可清除该错误。

12.1.2.6 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位。从接收 FIFO 缓冲区读取 9 位数据时，在读取 RCREG 的 8 个最低有效位前必须先读取 RX9D 数据位。

12.1.2.7 地址检测

当多个接收器共用同一条发送线时，如在 RS-485 系统中，有一个特殊的地址检测模式可供使用。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 个数据位置 1 的字符会被传送到接收 FIFO 缓冲区，并将 RCIF 中断位置 1。所有其他字符均被忽略。

接收到地址字符后，用户软件可决定地址是否与自身匹配。地址匹配时，发生下一个停止位前，用户软件必须通过清零 ADDEN 位禁止地址检测。当用户软件根据所使用的报文协议检测到报文的末尾时，软件将 ADDEN 位置 1，将接收器重新置于地址检测模式。

12.1.2.8 异步接收设置：

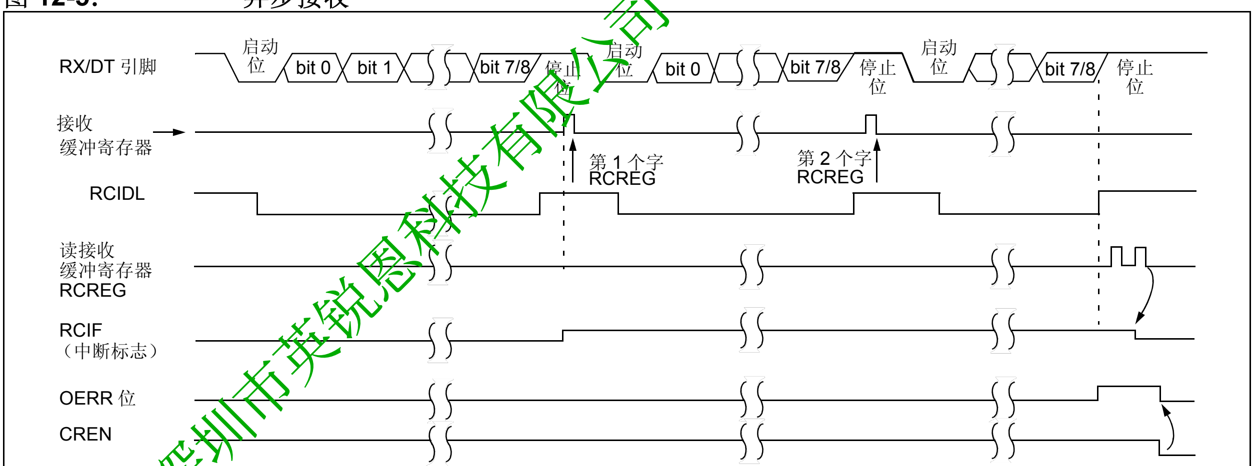
1. 初始化SPBRGH和SPBRG寄存器对以及BRGH和 BRG16 位，获得所需的波特率（见第 **12.3** 节 **"EUSART 波特率发生器（BRG）"**）。
2. 将 SPEN 位置 1 使能串行口。SYNC 位必须清零才能进行异步操作。
3. 如果需要中断，将 PIE1 寄存器中的 RCIE 位置 1，并将 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 如果需要接收 9 位，将 RX9 位置 1。
5. 将 CREN 位置 1 使能接收。
6. 当字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
7. 读取 RCSTA 寄存器取得错误标志，以及第 9 个数据位（9 位数据接收使能时）。
8. 读取 RCREG 寄存器从接收缓冲区取得接收的 8 个最低有效位。
9. 发生溢出时，通过清零 CREN 接收器使能位清零 OERR 标志位。

12.1.2.9 9 位地址检测模式设置

该模式一般用于 RS-485 系统中。要设置使能地址检测的异步接收：

1. 初始化SPBRGH和SPBRG寄存器对以及BRGH和 BRG16 位，获得所需的波特率（见第 **12.3** 节 **"EUSART 波特率发生器（BRG）"**）。
2. 将 SPEN 位置 1 使能串行口。SYNC 位必须清零才能进行异步操作。
3. 如果需要中断，将 PIE1 寄存器中的 RCIE 位置 1，并将 INTCON 寄存器的 GIE 和 PEIE 位置 1。
4. 将 RX9 位置 1 使能 9 位接收。
5. 将 ADDEN 位置 1 使能地址检测。
6. 将 CREN 位置 1 使能接收。
7. 当第 9 位置 1 的字符从 RSR 被移入接收缓冲区时，RCIF 中断标志位将被置 1。如果 RCIE 中断允许位也置 1，则产生中断。
8. 读取 RCSTA 寄存器取得错误标志。第 9 个数据位将始终置 1。
9. 读取 RCREG 寄存器从接收缓冲区取得接收的 8 个最低有效位。软件将决定此地址是否是器件地址。
10. 发生溢出时，通过清零 CREN 接收器使能位清零 OERR 标志位。
11. 如果器件被寻址，将 ADDEN 位清零以允许所有接收到的数据被送入接收缓冲区并产生中断。

图 12-5： 异步接收



注： 此时序图显示了在 RX 输入引脚上顺序接收的 3 个字。在第 3 个字后读 RCREG（接收缓冲器），这会使 OERR（溢出）位置 1。

表 12-2： 与异步接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	`01-0 0-00`	`01-0 0-00`
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	`0000 000x`	`0000 000x`
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	`-000 0000`	`-000 0000`
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	`-000 0000`	`-000 0000`
RCREG	EUSART 接收数据寄存器								`0000 0000`	`0000 0000`
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	`0000 000x`	`0000 000x`
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	`0000 0000`	`0000 0000`
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	`0000 0000`	`0000 0000`
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	`1111 ----`	`1111 ----`
TXREG	EUSART 发送数据寄存器								`0000 0000`	`0000 0000`
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	`0000 0010`	`0000 0010`

图注： `x` = 未知， `-` = 未实现 （读为 `0`）。异步接收不使用阴影单元。

12.2 异步操作的时钟精确性

内部振荡器模块输出 （INTOSC）在出厂时做了校准。但是，VDD 或温度变化时 INTOSC 频率有可能漂移，这将直接影响异步波特率。有两种方法可用来调整波特率时钟，但它们都需要某种参考时钟源。

第一种（建议）方法使用OSCTUNE寄存器调整INTOSC输出。调整 OSCTUNE 寄存器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 3.5 节 “内部时钟模式”**。

另一种方法调整波特率发生器的值。自动波特率检测可自动完成这种调整 （见**第 12.3.1 节 “自动波特率检测”**）。调整波特率发生器以补偿外设时钟频率的逐渐变化时，可能分辨率精度不够。

寄存器 12-1： TXSTA：发送状态和控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN(1)	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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **CSRC：**时钟源选择位
异步模式：
无关位
同步模式：
1 = 主模式 （由 BRG 内部产生时钟）
0 = 从模式 （时钟源来自外部）

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 (1)
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 （完成后由硬件清零）
0 = 同步间隔字符发送完成
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：
在此模式下不使用

bit 1 **TRMT：**发送移位寄存器状态位
1 = TSR 空
0 = TSR 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： 在同步模式下， SREN/CREN 可改写 TXEN。

寄存器 12-2： RCSTA：接收状态和控制寄存器 [1]

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-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	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **SPEN**：串行口使能位
1 = 使能串行口 （将 RX/DT 和 TX/CK 引脚配置为串行口引脚）
0 = 禁止串行口 （保持为复位状态）

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
<u>异步模式</u>：
无关位
<u>同步模式——主</u>：
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。
<u>同步模式——从</u>：
无关位

bit 4 **CREN**：连续接收使能位
<u>异步模式</u>：
1 = 使能接收器
0 = 禁止接收器
<u>同步模式</u>：
1 = 使能连续接收，直到使能位 CREN 清零 （CREN 改写 SREN）
0 = 禁止连续接收

bit 3 **ADDEN**：地址检测使能位
<u>9 位异步模式 （RX9 = 1）</u>：
1 = 使能地址检测、允许中断，当 RSR<8> 置 1 时装入接收缓冲区
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
<u>8 位异步模式 （RX9 = 0）</u>：
无关位

bit 2 **FERR**：帧错误位
1 = 帧错误 （可通过读 RCREG 寄存器刷新该位，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。

寄存器 12-3： BAUDCTL：波特率控制寄存器

R-0	R-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

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 = 自动波特率定时器没有溢出
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 = 已接收到启动位且接收器正在接收
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 = 传送反相数据到 RB7/TX/CK 引脚
0 = 传送非反相数据到 RB7/TX/CK 引脚
同步模式：
1 = 数据在时钟上升沿同步
0 = 数据在时钟下降沿同步

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。不会接收字符。 RCIF 将被置 1。 RCIF 置 1 后 WUE 将被自动清零。
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 = 使能自动波特率模式（完成自动波特率后清零）
0 = 禁止自动波特率模式
同步模式：
无关位

12.3 EUSART 波特率发生器 （BRG）

波特率发生器 （BRG）是 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 BAUDCTL 寄存器的 BRG16 位置 1 可选择 16 位模式。

SPBRGH 和 SPBRG 寄存器对决定自由运行波特率定时器的周期。在异步模式下，波特率周期的乘数由 TXSTA 寄存器的 BRGH 位和 BAUDCTL 寄存器的 BRG16 位决定。在同步模式下， BRGH 位被忽略。

表 12-3 提供了确定波特率的公式。例 12-1 是确定波特率和波特率误差的计算示例。

为便于您使用，各种异步模式的典型波特率和误差值已经计算出来，如表 12-3 所示。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 模式用于在高速振荡器频率下取得较缓慢的波特率。

将新值写入 SPBRGH 和 SPBRG 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过程中更改系统时钟，可能会导致接收错误或数据丢失。为避免此问题，应检查 RCIDL 位的状态，以确保在更改系统时钟前接收操作处于空闲状态。

例 12-1： 计算波特率误差

器件工作在 Fosc = 16 MHz，目标波特率 = 9600，异步模式， 8 位 BRG：

$$\text{目标波特率} = \frac{F_{OSC}}{64([SPBRGH:SPBRG] + 1)}$$

求解 SPBRGH:SPBRG：

$$X = \frac{\frac{F_{OSC}}{\text{目标波特率}}}{64} - 1$$

$$= \frac{\frac{16000000}{9600}}{64} - 1$$

$$= [25.042] = 25 \quad \text{十进制整数}$$

$$\text{计算波特率} = \frac{16000000}{64(25 + 1)}$$

$$= 9615$$

$$\text{误差} = \frac{\text{计算波特率} - \text{目标波特率}}{\text{目标波特率}}$$

$$= \frac{(9615 - 9600)}{9600} = 0.16\%$$

表 12-3： 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	Fosc/[64 (n+1)]
0	0	1	8 位 / 异步	Fosc/[16 (n+1)]
0	1	0	16 位 / 异步	Fosc/[16 (n+1)]
0	1	1	16 位 / 异步	Fosc/[4 (n+1)]
1	0	x	8 位 / 同步	Fosc/[4 (n+1)]
1	1	x	16 位 / 同步	Fosc/[4 (n+1)]

图注： x = 无关位， n = SPBRGH 和 SPBRG 寄存器对的值

表 12-4： 与波特率发生器相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， – = 实现 （读为 0）。波特率发生器不使用阴影单元。

表 **12-5:** 异步模式下的波特率

波特率	SYNC = 0，BRGH = 0，BRG16 = 0											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	1221	1.73	255	1200	0.00	239	1200	0.00	143	1202	0.16	103
2400	2404	0.16	129	2400	0.00	119	2400	0.00	71	2404	0.16	51
9600	9470	-1.36	32	9600	0.00	29	9600	0.00	17	9615	0.16	12
10417	10417	0.00	29	10286	-1.26	27	10165	-2.42	16	10417	0.00	11
19.2k	19.53k	1.73	15	19.20k	0.00	14	19.20k	0.00	8	—	—	—
57.6k	—	—	—	57.60k	0.00	7	57.60k	0.00	2	—	—	—
115.2k	—	—	—	—	—	—	—	—	—	—	—	—

波特率	SYNC = 0，BRGH = 0，BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	103	300	0.16	51
1200	1202	0.16	51	1200	0.00	47	1202	0.16	25	1202	0.16	12
2400	2404	0.16	25	2400	0.00	23	2404	0.16	12	—	—	—
9600	—	—	—	9600	0.00	5	—	—	—	—	—	—
10417	10417	0.00	5	—	—	—	10417	0.00	2	—	—	—
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 = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	—	—	—	—	—	—	—	—	—	—	—	—
1200	—	—	—	—	—	—	—	—	—	—	—	—
2400	—	—	—	—	—	—	—	—	—	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19231	0.16	25
57.6k	56.82k	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.64k	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

表 12-5： 异步模式下的波特率 （续）

波特率	SYNC = 0， BRGH = 1， BRG16 = 0											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.2k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

波特率	SYNC = 0， BRGH = 0， BRG16 = 1											
	Fosc = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	-0.01	4166	300.0	0.00	3839	300.0	0.00	2303	299.9	-0.02	1666
1200	1200	-0.03	1041	1200	0.00	959	1200	0.00	575	1199	-0.08	416
2400	2399	-0.03	520	2400	0.00	479	2400	0.00	287	2404	0.16	207
9600	9615	0.16	129	9600	0.00	119	9600	0.00	71	9615	0.16	51
10417	10417	0.00	119	10378	-0.37	110	10473	0.53	65	10417	0.00	47
19.2k	19.23k	0.16	64	19.20k	0.00	59	19.20k	0.00	35	19.23k	0.16	25
57.6k	56.818	-1.36	21	57.60k	0.00	19	57.60k	0.00	11	55556	-3.55	8
115.2k	113.63	-1.36	10	115.2k	0.00	9	115.2k	0.00	5	—	—	—

波特率	SYNC = 0， BRGH = 0， BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	300.1	0.04	832	300.0	0.00	767	299.8	-0.108	416	300.5	0.16	207
1200	1202	0.16	207	1200	0.00	191	1202	0.16	103	1202	0.16	51
2400	2404	0.16	103	2400	0.00	95	2404	0.16	51	2404	0.16	25
9600	9615	0.16	25	9600	0.00	23	9615	0.16	12	—	—	—
10417	10417	0.00	23	10473	0.53	21	10417	0.00	11	10417	0.00	5
19.2k	19.23k	0.16	12	19.20k	0.00	11	—	—	—	—	—	—
57.6k	—	—	—	57.60k	0.00	3	—	—	—	—	—	—
115.2k	—	—	—	115.2k	0.00	1	—	—	—	—	—	—

表 12-5： 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 = 20.000 MHz			Fosc = 18.432 MHz			Fosc = 11.0592 MHz			Fosc = 8.000 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	16665	300.0	0.00	15359	300.0	0.00	9215	300.0	0.00	6666
1200	1200	-0.01	4166	1200	0.00	3839	1200	0.00	2303	1200	-0.02	1666
2400	2400	0.02	2082	2400	0.00	1919	2400	0.00	1151	2401	0.04	832
9600	9597	-0.03	520	9600	0.00	479	9600	0.00	287	9615	0.16	207
10417	10417	0.00	479	10425	0.08	441	10433	0.16	264	10417	0	191
19.2k	19.23k	0.16	259	19.20k	0.00	239	19.20k	0.00	143	19.23k	0.16	103
57.6k	57.47k	-0.22	86	57.60k	0.00	79	57.60k	0.00	47	57.14k	-0.79	34
115.2k	116.3k	0.94	42	115.2k	0.00	39	115.2k	0.00	23	117.6k	2.12	16

波特率	SYNC = 0， BRGH = 1， BRG16 = 1 或 SYNC = 1， BRG16 = 1											
	Fosc = 4.000 MHz			Fosc = 3.6864 MHz			Fosc = 2.000 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	0.01	3332	300.0	0.00	3071	299.9	-0.02	1666	300.1	0.04	832
1200	1200	0.04	832	1200	0.00	767	1199	-0.08	416	1202	0.16	207
2400	2398	0.08	416	2400	0.00	383	2404	0.16	207	2404	0.16	103
9600	9615	0.16	103	9600	0.00	95	9615	0.16	51	9615	0.16	25
10417	10417	0.00	95	10473	0.53	87	10417	0.00	47	10417	0.00	23
19.2k	19.23k	0.16	51	19.20k	0.00	47	19.23k	0.16	25	19.23k	0.16	12
57.6k	58.82k	2.12	16	57.60k	0.00	15	55.56k	-3.55	8	—	—	—
115.2k	111.1k	-3.55	8	115.2k	0.00	7	—	—	—	—	—	—

12.3.1 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 Rate Detect，ABD）模式下， BRG 的时钟信号反向。 BRG 并不为进入的 RX 信号提供时钟信号，而是相反由 RX 信号为 BRG 定时。波特率发生器用于为接收的 55h （“U”的 ASCII 码）定时， 55h 是 LIN 总线的同步字符。此字符的特殊之处在于它具有包括停止位边沿在内的 5 个上升沿。

将 BAUDCTL 寄存器的 ABDEN 位置 1 将启动自动波特率校验序列（图 12-6）。当发生 ABD 序列时，EUSART 状态机保持在空闲状态。在接收线的第一个上升沿（启动位之后）， SPBRG 使用 BRG 计数器时钟递增计数，如表 12-6 所示。在第 8 位周期的末尾将在 RX 引脚上出现第 5 个上升沿。此时，对正确的 BRG 周期的累计值被留在 SPBRGH 和 SPBRG 寄存器对中，ABDEN 位被自动清零而 RCIF 中断标志被置 1。要清除 RCIF 中断，需要读取RCREG中的值。RCREG的内容应该被丢弃。校准不使用 SPBRGH 寄存器的模式时，用户可通过查询 SPBRGH 寄存器中的 00h 验证 SPBRG 寄存器是否未溢出。

BRG 自动波特率时钟由 BRG16 和 BRGH 位决定，如表 12-6 所示。在 ABD 期间， SPBRGH 和 SPBRG 寄存器共同用作 16 位计数器，这与 BRG16 位的设置无关。在校准波特率周期时，SPBRGH 和 SPBRG 寄存器的定时频率为 BRG 基时钟频率的 1/8。得到的字节测量结果为全速下的平均位时间。

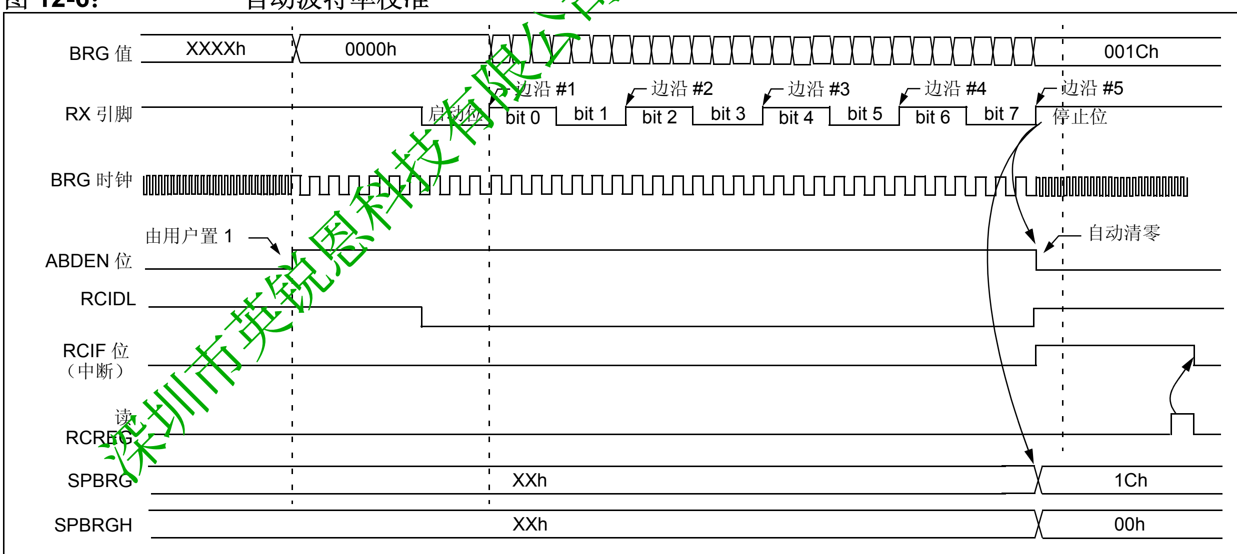
注	
1：	如果 WUE 位和 ABDEN 位都置 1，自动波特率检测将发生在间隔字符之后的字节处（见第 **12.3.2 节 “接收到间隔字符时自动唤醒”**）。
2：	需要由用户来判断输入字符的波特率是否处于所选 BRG 时钟源范围内。某些振荡器频率和 EUSART 波特率组合不可能实现。
3：	在自动波特率过程中，自动波特率计数器从 1 开始计数。自动波特率序列完成后，为了得到最准确的结果，应从 SPBRGH:SPBRG 寄存器对的值中减去 1。

表 12-6： BRG 计数器时钟速率

BRG16	BRGH	BRG 基时钟	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： 在 ABD 序列期间，SPBRG 和 SPBRGH 寄存器都用作 16 位计数器，与 BRG16 的设置无关。

图 12-6： 自动波特率校准



注 1： ABD 序列要求 **EUSART** 模块配置为异步模式。

12.3.2 接收到间隔字符时自动唤醒

在休眠模式下，EUSART 的所有时钟都会暂停。因此，波特率发生器处于不工作状态，不能正常进行字符接收。自动唤醒功能使控制器可被 RX/DT 线上的活动唤醒。该功能只在异步模式下可用。

自动唤醒功能可通过将 BAUDCTL 寄存器的 WUE 位置 1 来使能。一旦置 1， RX/DT 上的正常接收序列就被禁止， EUSART 保持在空闲状态，监视与 CPU 模式无关的唤醒事件。唤醒事件包含 RX/DT 线上由高至低的跳变。（这正好也是同步间隔字符的开始或 LIN 协议的唤醒信号字符。）

EUSART 模块生成的 RCIF 中断与唤醒事件巧合。在正常 CPU 工作模式下，中断产生与 Q 时钟同步（图 12-7），而器件处于休眠模式时则异步发生 （图 12-8）。通过读 RCREG 寄存器可清除中断条件。

在间隔字符末尾 RX 线由低至高的跳变将自动清零 WUE 位。这向用户表明间隔事件结束。此时，EUSART 模块处于空闲模式，等待接收下一个字符。

12.3.2.1 特殊注意事项

间隔字符

在发生唤醒事件期间为了避免字符错误或字符碎片，唤醒字符必须为全零。

唤醒被使能时，其工作与数据流的低电平时间无关。如果 WUE 位置 1 并接收到了有效的非零字符，则从启动位至第一个上升沿的低电平时间将被解读为唤醒事件。字符的其余位将作为碎片字符接收，后续字符有可能产生碎片或出现溢出错误。

因此，发送的首字符必须为全 0。这必须持续 10 个或更长的位时间，建议 LIN 总线持续 13 个位时间，而标准 RS-232 器件可为任意个位时间。

振荡器起振时间

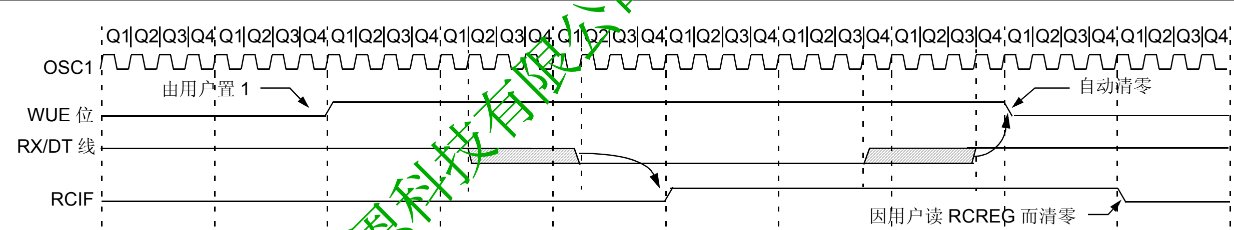
必须考虑振荡器起振时间，特别在使用具有较长起振时间的振荡器（即，LP、XT 或 HS/PLL 模式）的应用中。同步间隔 （或唤醒信号）字符必须足够长，并随后有一个足够长的时间，以使所选的振荡器有足够的时间起振并向 EUSART 提供适当的初始化。

WUE 位

唤醒事件会通过将 RCIF 位置 1 产生一个接收中断。WUE 位在 RX/DT 的上升沿由硬件清零。之后软件通过读取 RCREG 寄存器并丢弃其内容将中断条件清除。

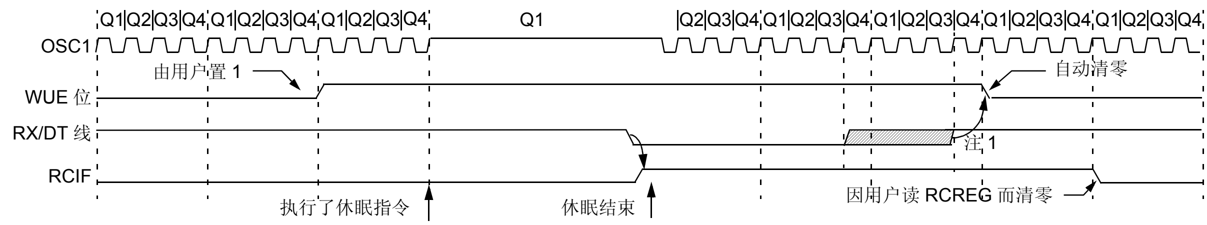
要确保不丢失实际数据，应在将 WUE 位置 1 前检查 RCIDL 位，验证没有接收操作在进行。如果未发生接收操作，可在进入休眠模式前将 WUE 位置 1。

图 12-7： 正常操作时的自动唤醒位 （WUE） 时序



注 1： WUE 位置 1 时， EUSART 保持在空闲状态。

图 12-8： 休眠时的自动唤醒位 （WUE）时序



注 1： 如果唤醒事件需要要较长的振荡器预热时间，在 stposc 信号仍然有效时可能发生 WUE 位自动清零。该时序与 Q 时钟出现与否无关。

2： WUE 位置 1 时， **EUSART** 保持在空闲状态。

12.3.3 间隔字符序列

EUSART 模块能够发送符合 LIN 总线标准的特殊间隔字符序列。间隔字符包含 1 个启动位，随后的 12 个 0 位和 1 个停止位。

要发送间隔字符，应将 TXSTA 寄存器的 SENDB 和 TXEN 位置 1。随后对 TXREG 执行写操作可启动间隔字符发送。写入 TXREG 的数据值会被忽略并发送全 0。

在发送了相应的停止位后，硬件会自动将 SENDB 位复位。这样用户可以在间隔字符（在 LIN 规范中通常是同步字符）后将下一个要发送字节预先装入发送 FIFO。

TXSTA 寄存器的 TRMT 位表明发送操作何时处于激活或空闲状态，这与正常发送时相同。关于发送间隔字符的时序，请参见图 12-9。

12.3.3.1 间隔和同步发送序列

以下序列将启动报文帧头（header），它由间隔字符和其后的自动波特率同步字节组成。这是 LIN 总线主器件的典型序列。

1. 将 EUSART 配置为所需的模式。
2. 将 TXEN 和 SENDB 位置 1 使能间隔序列。
3. 将无效字符装入 TXREG，启动发送（该值会被忽略）。
4. 将 55h 写入 TXREG，以将同步字符装入发送 FIFO 缓冲区。
5. 发送间隔字符后，SENDB 位被硬件复位，同步字符随后被发送。

当 TXREG 为空时（由 TXIF 指出），下一个数据字节会写入 TXREG。

12.3.4 接收间隔字符

增强型 EUSART 模块接收间隔字符有两种方法。

第一种检测间隔字符的方法采用 RCSTA 寄存器的 FERR 位和如 RCREG 所指示的接收数据。假定波特率发生器已初始化为预期的波特率。

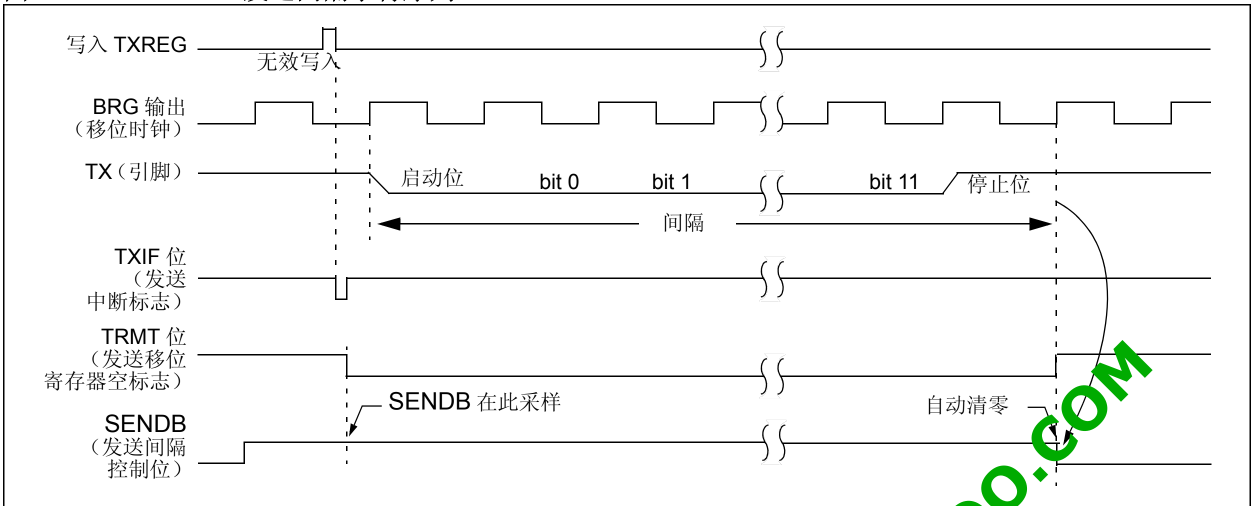
以下成立时，表明接收到间隔字符：

- RCIF 位被置 1
- FERR 位被置 1
- RCREG = 00h

第二种方法采用**第 12.3.2 节 “接收到间隔字符时自动唤醒”**中所述的自动唤醒功能。通过使能此功能，EUSART 将采样 RX/DT 上的下两次跳变，产生 RCIF 中断，并接收下一个数据字节并再产生一次中断。

请注意在间隔字符后，用户通常希望使能自动波特率检测功能。采用这两种方法时，用户均可在 EUSART 进入休眠模式前将 BAUDCTL 寄存器的 ABDEN 位置 1。

图 12-9： 发送间隔字符序列



12.4 EUSART 同步模式

系统使用的同步串行通信通常带有一个主器件和一个或多个从器件。主器件包含波特率发生所需的电路，可将时钟提供给系统中的所有器件。从器件使用主时钟，可不再需要内部时钟发生电路。

同步模式下有两条信号线：双向数据线和时钟线。从器件使用由主器件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，同步操作只能是半双工的。半双工指主从器件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。

同步发送不使用启动位和停止位。

12.4.1 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 （发送）； SREN = 1 （接收）
- CREN = 0 （发送）； CREN = 1 （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，否则器件将被配置为接收。将 RCSTA 寄存器的 SPEN 位置 1 可使能 EUSART。如果 RX/DT 或 TX/CK 引脚与模拟外设共用，那么必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

12.4.1.1 主时钟

同步数据传送使用独立的时钟线，它与数据同步。配置为主器件的器件将时钟发送到 TX/CK 线上。EUSART 配置为同步发送或接收操作时，TX/CK 引脚的输出驱动器被自动使能。串行数据位在时钟前沿改变，以确保其在时钟的后续边沿有效。每个数据位都产生一个时钟周期。数据位有多少，就产生多少时钟周期。

12.4.1.2 时钟极性

提供了时钟极性选项以与 Microwire 兼容。时钟极性通过 BAUDCTL 寄存器的 SCKP 位选择。将 SCKP 位置 1 将时钟空闲状态设置为高电平。 SCKP 位置 1 时，数据在每个时钟的下降沿改变。将 SCKP 位清零将时钟空闲状态设置为低电平。 SCKP 位清零时，数据在每个时钟的上升沿改变。

12.4.1.3 同步主发送

数据从器件被传送到 RX/DT 引脚上。 EUSART 配置为同步主发送操作时， RX/DT 和 TX/CK 引脚的输出驱动器被自动使能。

向 TXREG 寄存器写入一个字符时启动发送。如果 TSR 中仍保存前一个字符的全部或部分，则新字符被保存在 TXREG 中，直到前一个字符的最后一位被发送。如果这是首字符，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。

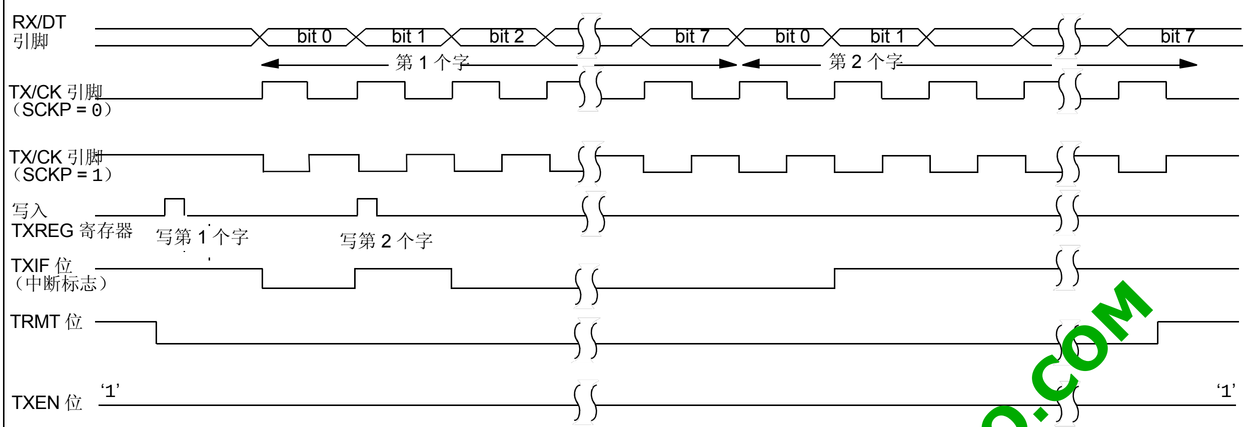
每个数据位在主时钟的时钟前沿改变，并在下一个时钟前沿到来前保持有效。

> **注：** TSR 寄存器不映射到数据存储器中，因此用户无法使用。

12.4.1.4 同步主发送设置：

1. 初始化 SPBRGH 和 SPBRG 寄存器对以及 BRGH 和 BRG16 位，获得所需的波特率（见第 **12.3 节 “EUSART 波特率发生器 （BRG）”**）。
2. 将 SYNC、SPEN 和 CSRC 位置 1 使能同步主串行口。
3. 将 SREN 和 CREN 位清零禁止接收模式。
4. 将 TXEN 位置 1 使能发送模式。
5. 如果需要 9 位发送，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. 如果选择了 9 位发送，应将第 9 位装入 TX9D 位。
8. 将数据装入 TXREG 寄存器，启动发送。

图 12-10：　　同步发送



注：　同步主模式， SPBRG = 0，连续发送两个 8 位字。

图 12-11：　　同步发送 （由 TXEN 位控制）

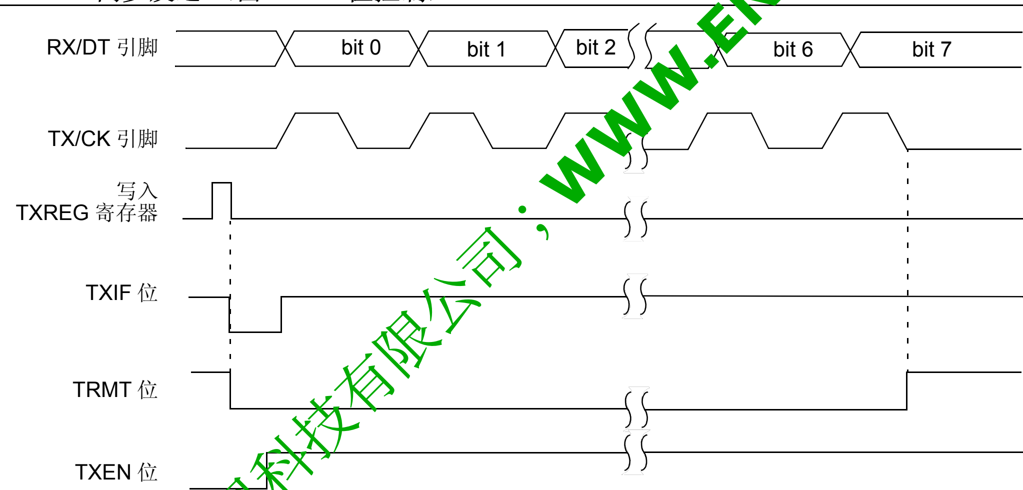


表 12-7：　　与同步主发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注：　x = 未知， — = 未实现 （读为 0）。同步主发送不使用阴影单元。

12.4.1.5 同步主接收

数据在 RX/DT 引脚上接收。 EUSART 配置为同步主接收操作时， RX/DT 引脚的输出驱动器被自动禁止。

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 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时钟周期。一个字符接收完成后 SREN 位被自动清零。CREN 置 1 时，将连续产生时钟直到 CREN 被清零。如果 CREN 在字符接收过程中被清零，则 CK 时钟立即停止，接收到的部分字符被丢弃。如果 SREN 和 CREN 同时置 1，则首字符接收完成时 SREN 被清零， CREN 优先。

要启动接收，将 SREN 或 CREN 置 1。在 TX/CK 时钟引脚的后续边沿对 RX/DT 引脚上的数据进行采样，并移入接收移位寄存器 （RSR） 。当完整的字符被接收进 RSR 后，RCIF 位置 1 且该字符被自动送入两个字符的接收 FIFO。接收 FIFO 中顶部字符的 8 个最低有效位在 RCREG 中。只要接收 FIFO 中有未读字符，RCIF 位就保持置 1。

12.4.1.6 从时钟

同步数据传送使用独立的时钟线，时钟与数据同步。配置为从器件的器件在 TX/CK 线上接收时钟信号。将器件配置为同步从发送或接收操作时，TX/CK 引脚输出驱动器被自动进制。串行数据位在时钟前沿改变，以确保其在时钟的后续边沿有效。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。

12.4.1.7 接收溢出错误

接收 FIFO 缓冲区可容纳两个字符。在 RCREG 被读取以访问 FIFO 前，接收到完整的第三个字符时会产生溢出错误。此时，RCSTA 寄存器的 OERR 位置 1。FIFO 中的前一个数据将不会被改写。 FIFO 缓冲区中的两个字符可被读出，但错误被清除前不能再接收其他字符。只有清除了溢出条件才可将 OERR 位清零。如果 SREN 位置1且CREN位清零时发生溢出错误，则读取RCREG 可清除错误。如果 CREN 位置 1 时发生溢出，则可通过清零 RCSTA 寄存器的 CREN 位或清零可将 EUSART 复位的 SPEN 位清除错误条件。

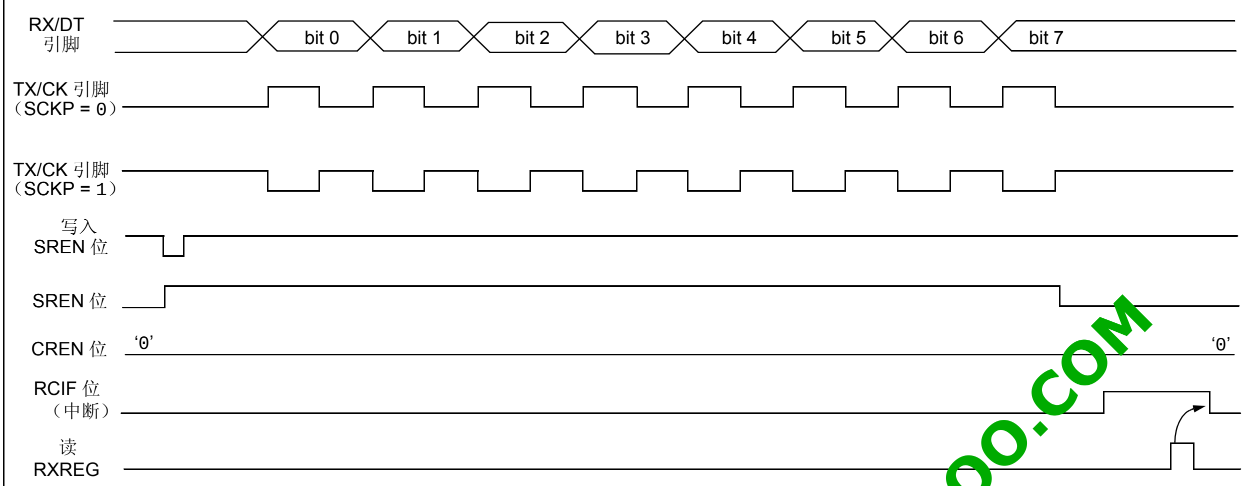
12.4.1.8 接收 9 位字符

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位。从接收 FIFO 缓冲区读取 9 位数据时，在读取 RCREG 的 8 个最低有效位前必须先读取 RX9D 数据位。

12.4.1.9 同步主接收设置：

1. 初始化 SPBRGH 和 SPBRG 寄存器对，获得所需的波特率。按需要将 BRGH 和 BRG16 位置 1 或清零，获得所需的波特率。
2. 将 SYNC、SPEN 和 CSRC 位置 1 使能同步主串行口。
3. 确保将 CREN 和 SREN 位清零。
4. 如果需要中断，将 INTCON 寄存器的 GIE 和 PEIE 位置 1，并将 PIE1 寄存器的 RCIE 位也置 1。
5. 如果需要接收 9 位数据，将 RX9 位置 1。
6. 将 SREN 位置 1，或连续接收时将 CREN 位置 1，启动接收。
7. 字符接收完成时中断标志位 RCIF 将被置 1。如果允许位 RCIE 已置 1，则产生中断。
8. 读取 RCSTA 寄存器取得第 9 位（如果已使能），并确定接收时是否发生了错误。
9. 通过读取 RCREG 寄存器来读取接收到的 8 位数据。
10. 如果发生了溢出错误，可通过清零 RCSTA 寄存器的CREN位或清零可将EUSART复位的SPEN 位清除错误。

图 12-12：　　同步接收（主模式，SREN）



注：　　时序图说明 SREN = 1 且 BRGH = 0 时的同步主模式。

表 12-8：　　与同步主接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注：　x = 未知， – = 未实现（读为 0）。同步主接收不使用阴影单元。

12.4.2 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 （发送）； SREN = 1 （接收）
- CREN = 0 （发送）； CREN = 1 （接收）
- SPEN = 1

将 TXSTA 寄存器的 SYNC 位置 1 将器件配置为同步操作。将 TXSTA 寄存器的 CSRC 位清零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，否则器件将被配置为接收。将 RCSTA 寄存器的 SPEN 位置 1 可使能 EUSART。如果 RX/DT 或 TX/CK 引脚与模拟外设共用，那么必须通过清零相应的 ANSEL 位禁止模拟 I/O 功能。

12.4.2.1 EUSART 同步从发送

除了休眠模式以外，同步主模式和从模式的工作原理是相同的 （见 第 **12.4.1.3** 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. 第二个字将保留在 TXREG 寄存器中。
3. TXIF 位不会被置 1。
4. 第一个字符移出 TSR 后，TXREG 寄存器会将第二个字符传送到 TSR，此时 TXIF 位将置 1。
5. 如果 PEIE 和 TXIE 位均置 1，则发生中断将器件从休眠唤醒，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。

12.4.2.2 同步从发送设置：

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 清零 CREN 和 SREN 位。
3. 如果需要中断，应确保将 INTCON 寄存器的 GIE 和PEIE位置1，并将PIE1寄存器的TXIE位置1。
4. 如果需要 9 位发送，将 TX9 位置 1。
5. 将 TXEN 位置 1 使能发送。
6. 如果选择了 9 位发送，将最高有效位插入 TX9D 位。
7. 将 8 个最低有效位写入 TXREG 寄存器，启动发送。

表 12-9： 与同步从发送相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， - = 未实现 （读为 0）。同步从发送不使用阴影单元。

12.4.2.3 EUSART 同步从接收

除下列各项外，同步主模式和从模式的工作原理是相同的（第 **12.4.1.5** 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”

进入休眠前将 CRENA 位置 1，可在休眠模式下接收一个字符。接收到该字后， RSR 寄存器将把数据发送到 RCREG 寄存器。如果 RCIE 允许位置 1，产生的中断会将器件从休眠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。

12.4.2.4 同步从接收设置：

1. 将 SYNC 和 SPEN 位置 1 并清零 CSRC 位。
2. 如果需要中断，应确保将 INTCON 寄存器的 GIE 和 PEIE 位置 1，并将 PIE1 寄存器的 RCIE 位也置 1。
3. 如果需要接收 9 位，将 RX9 位置 1。
4. 将 CREN 位置 1 使能接收。
5. 接收完成时 RCIF 位将被置 1。如果 RCIE 位已置 1，则产生中断。
6. 如果使能了 9 位模式，从 RCSTA 寄存器的 RX9D 位取出最高有效位。
7. 读取 RCREG 寄存器，从接收 FIFO 中取出 8 个最低有效位。
8. 如果发生了溢出错误，可通过清零 RCSTA 寄存器的CREN位或清零可将EUSART复位的SPEN 位清除错误。

表 12-10： 与同步从接收相关的寄存器

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART 接收数据寄存器								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
TXREG	EUSART 发送数据寄存器								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	0000 0010	0000 0010

图注： x = 未知， - = 未实现 （读为 0）。同步从接收不使用阴影单元。

12.5 EUSART 在休眠期间的工作

只有在同步从模式下，EUSART 在休眠期间才可继续工作。所有其他模式都需要系统时钟，因此不能产生在休眠模式下运行发送或接收移位寄存器所必需的信号。

同步从模式使用外部产生的时钟对发送和接收移位寄存器进行操作。

12.5.1 休眠期间的同步接收

要在休眠期间执行接收操作，在进入休眠模式之前必须满足下列所有条件：

- RCSTA 和 TXSTA 控制寄存器必须配置为同步从接收（见第 **12.4.2.4 节 "同步从接收设置："**）。
- 如果需要中断，将 PIE1 寄存器的 RCIE 位和 INTCON 寄存器的 GIE 和 PEIE 位置 1。
- 必须通过读取 RCREG 寄存器将 RCIF 中断标志位清零，以卸载接收缓冲器中所有待处理的字符。

进入休眠模式后，器件已准备好分别接受 RX/DT 和 TX/CK 引脚上的数据和时钟信号。当外部器件将数据字完全移入后，将 PIR1 寄存器的 RCIF 中断标志位置 1。从而，将处理器从休眠模式唤醒。

从休眠模式唤醒后，器件将执行 SLEEP 指令后面的指令。如果 INTCON 寄存器的 GIE 全局中断允许位也置 1，则调用地址 004h 处的中断服务程序。

12.5.2 休眠期间的同步发送

要在休眠期间执行发送操作，在进入休眠模式之前必须满足下列所有条件：

- RCSTA 和 TXSTA 控制寄存器必须配置为同步从发送模式（见第 **12.4.2.2 节 "同步从发送设置："**）。
- 将输出数据写入 TXREG 来清零 TXIF 中断标志位，从而填充 TSR 和发送缓冲器。
- 如果需要中断，将 PIE1 寄存器的 TXIE 位和 INTCON 寄存器的 PEIE 位置 1。

进入休眠模式后，器件已准备好接受 TX/CK 上的时钟信号和发送 RX/DT 引脚上的数据。当外部器件将 TSR 中数据字全部移出后， TXREG 中的待处理字节将被发送到 TSR，且 TXIF 标志位置 1。从而将处理器从休眠模式唤醒。此时， TXREG 可用于接收其他发送字符，此操作将清零 TXIF 标志位。

从休眠模式唤醒后，器件将执行 SLEEP 指令后面的指令。如果GIE全局中断允许位也置1，则调用地址0004h处的中断服务程序。

13.0 SSP 模块概述

同步串行口（SSP）模块是用于同其他外设或单片机器件进行通信的串行接口。这些外设器件可能是串行 EEPROM、移位寄存器、显示驱动器或 A/D 转换器等。SSP 模块有下列两种工作模式：

- 串行外设接口 （SPI）
- I^2C™

请参见应用笔记 AN578，*“Use of the SSP Module in the Multi-Master Environment”*（DS00578）。

13.1 SPI 模式

本节包含寄存器定义和 SPI 模块的工作特性。

SPI 模式允许同时同步发送和接收 8 位数据。通常使用以下 3 个引脚来实现通信：

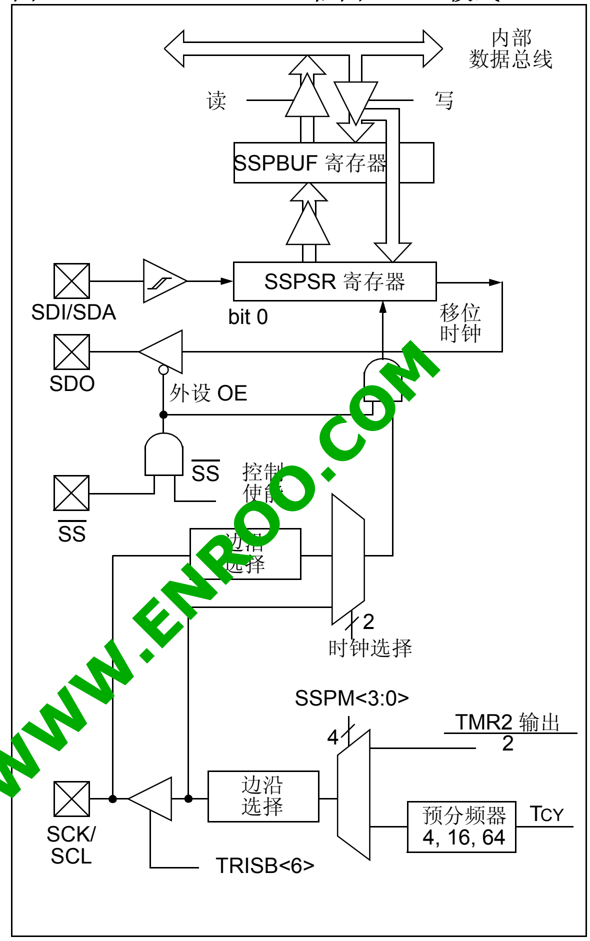
- 串行数据输出 （Serial Data Out， SDO）
- 串行数据输入 （Serial Data In， SDI）
- 串行时钟 （Serial Clock， SCK）

此外，当处于从工作模式时要使用第 4 个引脚：

- 从选择 （Slave Select， $\overline{SS}$）

注	
1：	当 SPI 处于从模式且 $\overline{SS}$ 引脚控制使能（SSPCON 寄存器的 SSPM<3:0> 位 = 0100）时，如果 $\overline{SS}$ 引脚设置为 VDD，SPI 模块将会复位。
2：	如果 SPI 用于从模式且 CKE = 1，则必须使能 $\overline{SS}$ 引脚控制。
3：	当 SPI 处于从模式且 $\overline{SS}$ 引脚控制使能（SSPCON 寄存器的 SSPM<3:0> 位 = 0100）时， $\overline{SS}$ 引脚的状态可能会影响从 TRISC<4> 位读取的状态。从 SSP 模块向 PORTC 发送的外设 OE 信号控制从 TRISC<4> 位读取的状态（关于 PORTC 的信息，请参见**第 17.0 节“电气规范”**）。如果对 TRISC 寄存器执行读 - 写 - 修改指令（例如 BSF），而 $\overline{SS}$ 引脚处于高电平，这将使TRISC<7>位被置1，从而禁止SDO输出。

图 13-1： SSP 框图 （SPI 模式）



寄存器 13-1： SSPSTAT：同步串行口状态寄存器 (1)

R/W-0	R/W-0	R-0	R-0	R-0	R-0	R-0	R-0
SMP	CKE	D/$\overline{\text{A}}$	P	S	R/$\overline{\text{W}}$	UA	BF
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **SMP**：SPI 数据输入采样相位位
SPI 主模式：
1 = 在数据输出时间的末端采样输入数据
0 = 在数据输出中间时采样输入数据 （Microwire）
SPI 从模式：
当 SPI 用于从模式时，必须将 SMP 清零
I²C™ 模式：
该位必须保持清零

bit 6 **CKE**：SPI 时钟边沿选择位
SPI 模式， CKP = 0：
1 = 在 SCK 的上升沿发送数据 （Microwire 备选）
0 = 在 SCK 的下降沿发送数据
SPI 模式， CKP = 1：
1 = 在 SCK 的下降沿发送数据 （Microwire 默认）
0 = 在 SCK 的上升沿发送数据
I²C 模式：
该位必须保持清零

bit 5 **D/$\overline{\text{A}}$**：数据 / $\overline{\text{地址}}$位 （仅限 I²C 模式） (2)
1 = 表示接收或发送的最后字节是数据
0 = 表示接收或发送的最后字节是地址

bit 4 **P**：停止位 （仅限 I²C 模式）
当 SSP 模块禁止或最后检测到启动位时，该位会清零。 SSPEN 清零。
1 = 表示最后检测到了停止位 （该位在复位时为 0）
0 = 最后没有检测到停止位

bit 3 **S**：启动位 （仅限 I²C 模式）
当 SSP 模块禁止或最后检测到停止位时，该位会清零。 SSPEN 清零。
1 = 表示最后检测到了启动位 （该位在复位时为 0）
0 = 最后没有检测到启动位

bit 2 **R/$\overline{\text{W}}$**：读 / $\overline{\text{写}}$位信息 （仅限 I²C 模式）
该位保存最后一次地址匹配后的 R/$\overline{\text{W}}$ 位信息。该位仅在地址匹配到下一个启动位、停止位或 $\overline{\text{ACK}}$ 位之间有效。
1 = 读
0 = 写

bit 1 **UA**：更新地址位 （仅限 10 位 I²C 模式）
1 = 表示用户需要更新 SSPADD 寄存器中的地址
0 = 不需要更新地址

bit 0 **BF**：缓冲区满状态位
接收 （SPI 和 I²C 模式）：
1 = 接收完成， SSPBUF 满
0 = 接收未完成， SSPBUF 空
发送 （仅限 I²C 模式）：
1 = 正在发送， SSPBUF 满
0 = 发送完成， SSPBUF 空

注 1： 仅限 PIC16F687/PIC16F689/PIC16F690。
2： 如果接收操作被忽略，则该位的值不会更新。

寄存器 13-2： SSPCON：同步串行口控制寄存器 (1)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WCOL	SSPOV	SSPEN	CKP	SSPM3(2)	SSPM2(2)	SSPM1(2)	SSPM0(2)
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7 **WCOL**：写冲突检测位
`1` = 正在发送前一个字时对 SSPBUF 寄存器进行写操作 （必须用软件清零）
`0` = 无冲突

bit 6 **SSPOV**：接收溢出指示位
在 SPI 模式下：
`1` = SSPBUF 寄存器中仍保存原先的数据时，接收到新字节。如果发生溢出，SSPSR 中的数据会丢失。溢出只会在从模式下发生。即使只是发送数据，用户也必须读 SSPBUF，以避免将溢出位置 1。在主模式下，溢出位未置 1，这是因为每个新接收 （和发送）的字节都是通过写 SSPBUF 寄存器开始的。
`0` = 无溢出
在 I²C™ 模式下：
`1` = SSPBUF 寄存器中仍保存原先的字节时，接收到新字节。在发送模式下 SSPOV 为无关位。两种模式下 SSPOV 都必须用软件清零。
`0` = 无溢出

bit 5 **SSPEN**：同步串行口使能位
在 SPI 模式下：
`1` = 使能串行口，并将 SCK、 SDO 和 SDI 配置为串行口引脚
`0` = 禁止串行口，并将这些引脚配置为 I/O 端口引脚
在 I²C 模式下：
`1` = 使能串行口，并将 SDA 和 SCL 引脚配置为串行口引脚
`0` = 禁止串行口，并将这些引脚配置为 I/O 端口引脚
在两种模式下，当使能时，这些引脚必须被正确配置为输入或输出引脚。

bit 4 **CKP**：时钟极性选择位
在 SPI 模式下：
`1` = 高电平为时钟的空闲状态 （Microwire 默认）
`0` = 低电平为时钟的空闲状态 （Microwire 备选）
在 I²C 模式下：
SCK 释放控制
`1` = 使能时钟
`0` = 保持时钟低电平 （时钟延长）。（用于确保数据建立时间。）

bit 3-0 **SSPM<3:0>**：同步串行口模式选择位
`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` = I²C 从模式， 7 位地址
`0111` = I²C 从模式， 10 位地址
`1000` = 保留
`1001` = 在 SSPADD SFR 地址处装入 SSPMSK 寄存器 (2)
`1010` = 保留
`1011` = I²C 固件控制主模式 （从单元空闲）
`1100` = 保留
`1101` = 保留
`1110` = I²C 从模式， 7 位地址，并允许启动位和停止位中断
`1111` = I²C 从模式， 10 位地址，并允许启动位和停止位中断

注 **1：** 仅限 PIC16F687/PIC16F689/PIC16F690。
2： 选择该模式后，对 SSPADD SFR 地址进行的任何读写操作实际上是访问 SSPMSK 寄存器。

13.2 工作原理

初始化 SPI 时需要指定几个选项。这可以通过对相应的控制位（SSPCON<5:0> 和 SSPSTAT<7:6>）编程来实现。这些控制位用于指定以下选项：

- 主模式 （SCK 是时钟输出）
- 从模式 （SCK 是时钟输入）
- 时钟极性 （SCK 的空闲状态）
- 数据输入采样相位 （数据输出时间的中间或末尾）
- 时钟边沿 （在 SCK 的上升沿 / 下降沿输出数据）
- 时钟速率 （仅限主模式）
- 从选择模式 （仅限从模式）

SSP 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，先移位 MSb。在新数据接收完毕前，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，SSPSTAT 寄存器的缓冲区满状态位 BF 和中断标志位 SSPIF 被置 1。对接收数据进行双重缓冲 （SSPBUF）允许在前一个数据被读取前开始接收下一个字节。当 SSPBUF 寄存器正在发送 / 接收数据时，对它写入的任何数据都将被忽略，同时 SSPCON 寄存器的写冲突检测位 WCOL 被置 1。用户必须将 WCOL 位清零才能判断 SSPBUF 寄存器随后的写入是否成功完成。

为确保应用软件能有效地接收数据，在下一个要发送的数据字节写入 SSPBUF 之前，读取 SSPBUF 中现有的数据。 SSPSTAT 寄存器的缓冲区满位 BF 用于指示 SSPBUF 何时接收完毕（发送完成）。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，则不必理会该数据。一般来说， SSP 中断用于检测发送 / 接收何时完毕。 SSPBUF 必须被读出和 / 或写入。如果不使用中断方式，可以使用软件查询方式确保不发生写冲突。例 13-1 显示了数据发送时如何装入 SSPBUF （SSPSR）。

不能直接读写 SSPSR 寄存器，只能通过寻址 SSPBUF 寄存器来访问。此外， SSP 状态寄存器 （SSPSTAT）用于指示各种状态。

例 13-1： 装入 SSPBUF （SSPSR）寄存器

```
        BSF     STATUS,RP0      ;Bank 1
        BCF     STATUS,RP1      ;
LOOP    BTFSS   SSPSTAT, BF     ;Has data been received(transmit complete)?
        GOTO    LOOP            ;No
        BCF     STATUS,RP0      ;Bank 0
        MOVF    SSPBUF, W       ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA          ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W       ;W reg = contents of TXDATA
        MOVWF   SSPBUF          ;New data to xmit
```

13.3 使能 SPI I/O

要使能串行口，SSPCON 寄存器的 SSP 使能位 SSPEN 必须置 1。要复位或重新配置 SPI 模式，先将 SSPEN 位清零，重新初始化 SSPCON 寄存器，然后再将 SSPEN 位置 1。这将把 SDI、SDO、SCK 和 $\overline{SS}$ 引脚配置为串行口引脚。要使引脚实现串行口功能，有些引脚的数据方向位（在 TRISB 和 TRISC 寄存器中）必须进行正确编程如下：

- SDI 由 SPI 模块自动控制
- SDO 必须将 TRISC<7> 位清零
- SCK （主模式）必须将 TRISB<6> 位清零
- SCK （从模式）必须将 TRISB<6> 位置 1
- $\overline{SS}$ 必须将 TRISC<6> 位置 1

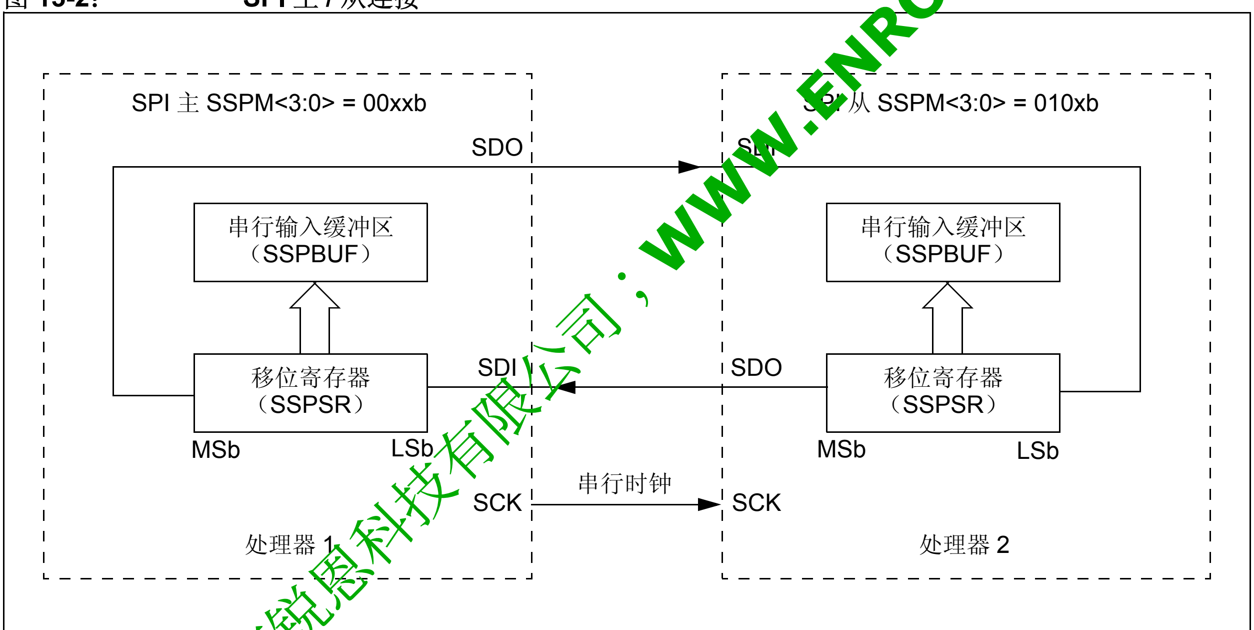
任何不需要的串行口功能可通过将相应引脚数据方向位（TRISB 和 TRISC）寄存器用相反值进行改写。

13.4 典型连接

图 13-2 给出了两个单片机之间的典型连接。主器件（处理器 1）通过发送 SCK 信号来启动数据传输。数据在编程设定的时钟边沿被移出两个处理器的移位寄存器，并在相反的时钟边沿被锁存。必须将两个处理器的时钟极性（CKP）设置为相同，这样就可以同时收发数据。数据是否有意义（或无效数据），取决于应用程序软件。数据发送有 3 种情况：

- 主器件发送数据——从器件发送无效数据
- 主器件发送数据——从器件发送数据
- 主器件发送无效数据——从器件发送数据

图 13-2： SPI 主 / 从连接



13.5 主模式

因为由主器件控制 SCK 信号，所以它可以在任意时刻启动数据传输。主器件根据软件协议确定从器件（图 13-2 中的处理器 2）应在何时广播数据。

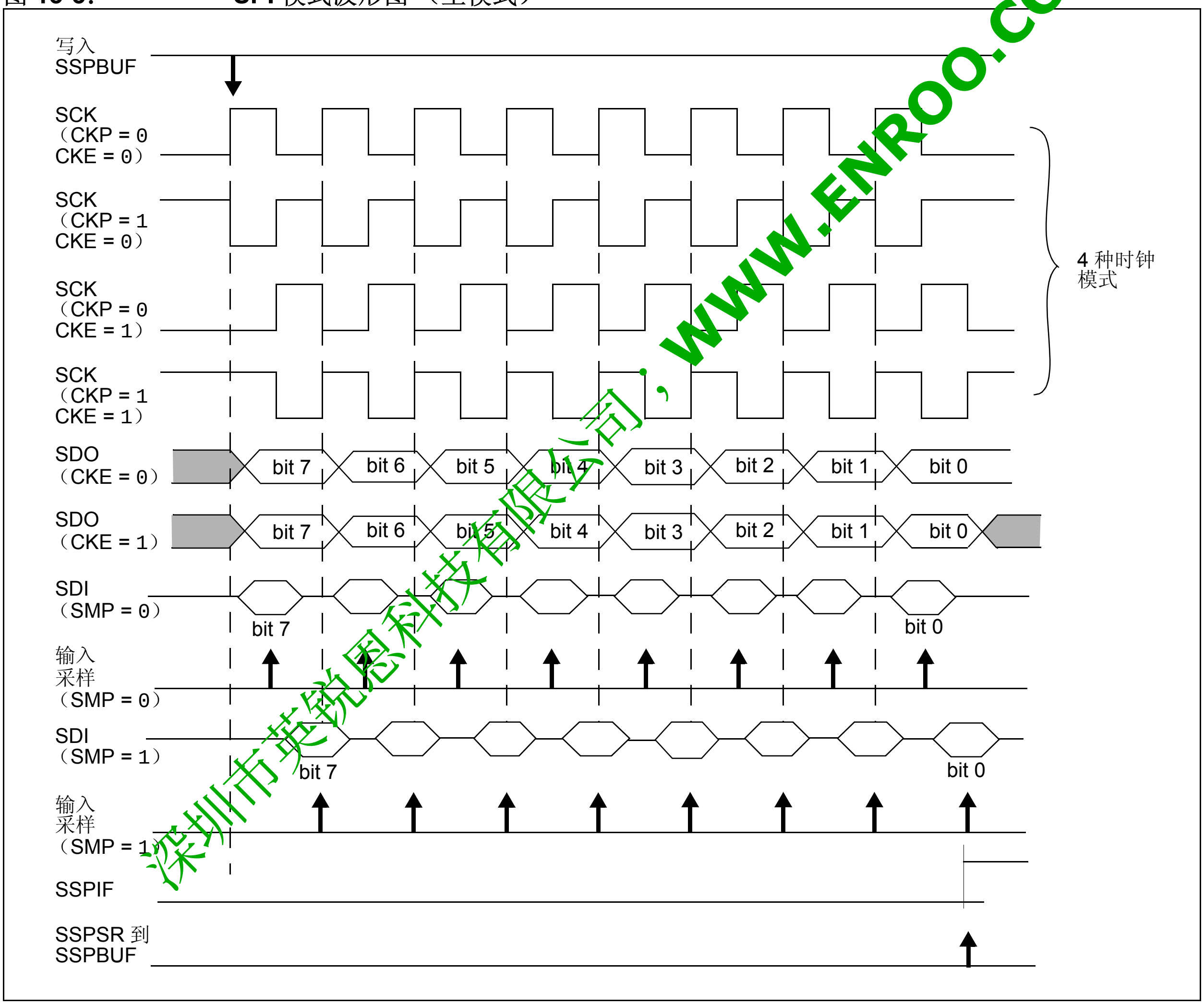
在主模式下，数据一旦写入 SSPBUF 寄存器就开始发送或接收。如果只打算将 SPI 作为接收器，则可以禁止 SDO 输出（将其编程设置为输入）。SSPSR 寄存器按设置的时钟速率，连续移入 SDI 引脚上的信号。每接收一个字节，就将它装入 SSPBUF 寄存器，就像普通的接收字节一样（相应中断和状态位置 1）。这在以“线路主动监控”（Line Activity Monitor）模式工作的接收器应用中很有用。

通过正确编程 SSPCON 寄存器的 CKP 位可以选择时钟极性。图 13-3、图 13-5 和图 13-6 将给出 SPI 通信的波形图，其中 MSb 最先发送。在主模式下，SPI 时钟速率（位速率）可由用户编程为以下之一：

- FOSC/4 （或 TCY）
- FOSC/16 （或 4 • TCY）
- FOSC/64 （或 16 • TCY）
- Timer2 输出 /2（无 SSP 模式，仅限 PIC16F690）

图 13-3 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。图中所示的输入采样的变化由 SMP 位的状态反映。图中给出了将接收到的数据装入 SSPBUF 的时间。

图 13-3： SPI 模式波形图 （主模式）



13.6 从模式

在从模式下，当 SCK 引脚上有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，中断标志位 SSPIF 置 1。

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。当接收到一个字节时，器件从休眠状态中唤醒。

13.7 从选择同步

$\overline{SS}$ 引脚允许同步从模式。要将 SPI 设置成从模式，需要使能 $\overline{SS}$ 引脚控制（SSPCON<3:0> = 04h）。要使 $\overline{SS}$ 引脚作为输入，不得将该引脚驱动为低电平。数据锁存器必须为高电平。当 $\overline{SS}$ 引脚为低电平时，使能数据的发送和接收，同时 SDO 引脚被驱动。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。根据应用需要，可外接上拉 / 下拉电阻。

> **注 1：** 当 SPI 处于从模式且 $\overline{SS}$ 引脚控制使能（SSPCON<3:0> = 0100）时，如果 $\overline{SS}$ 引脚设置为 VDD， SPI 模块将会复位。
>
> **2：** 如果 SPI 用于从模式且 CKE 置 1，则必须使能 $\overline{SS}$ 引脚控制。

SPI 模块复位时，位计数器被强制为 0。这可通过强制将 $\overline{SS}$ 引脚拉为高电平或将 SSPEN 位清零来实现。

将 SDO 引脚和 SDI 引脚相连，可以仿真二线制通信。当 SPI 需要作为接收器工作时， SDO 引脚可以被配置为输入。这样就禁止了从 SDO 发送数据。因为 SDI 不会引起总线冲突，所以可以一直将其保持为输入（SDI 功能）。

图 13-4： 从动同步波形图

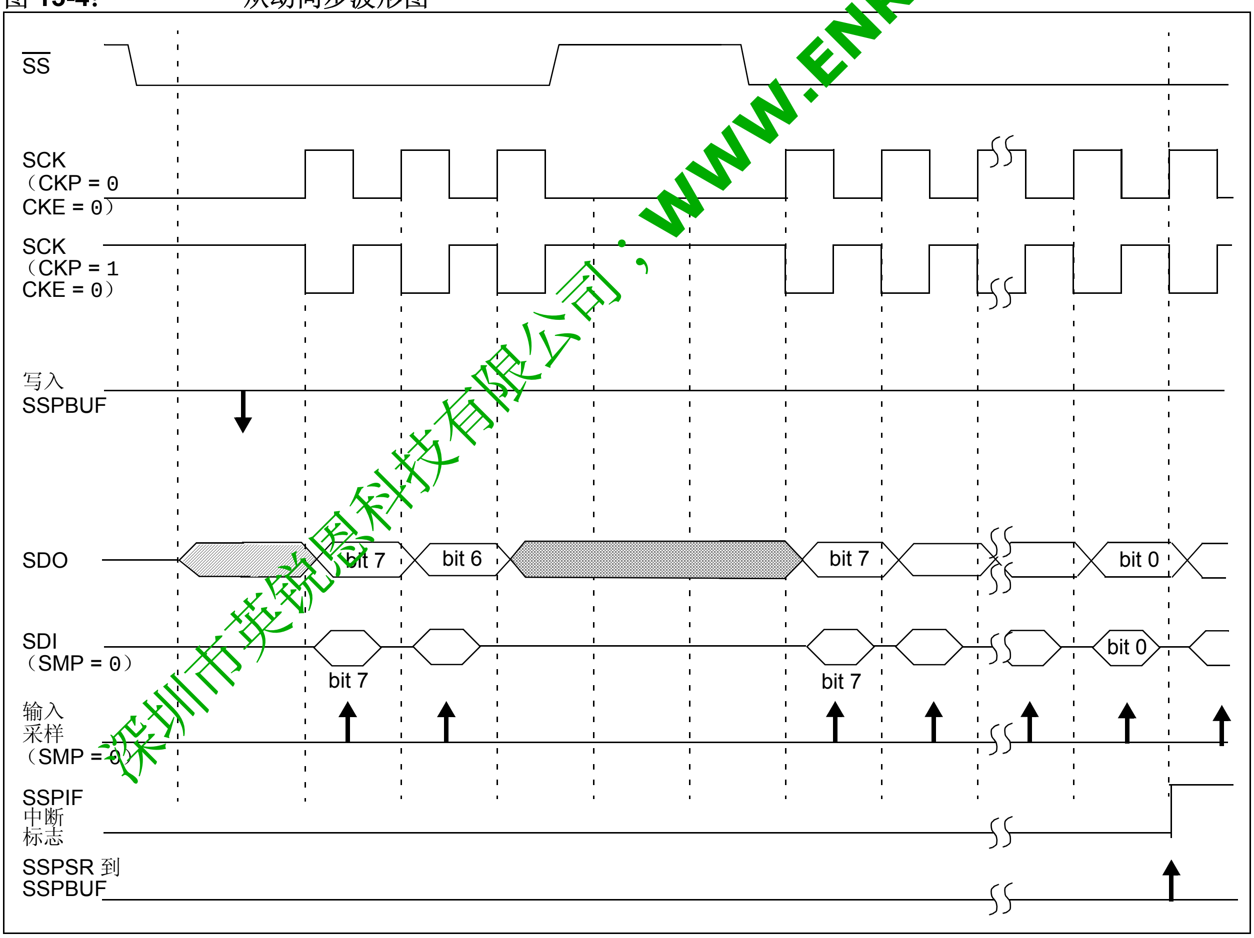


图 13-5: SPI 模式波形图（从模式， CKE = 0）

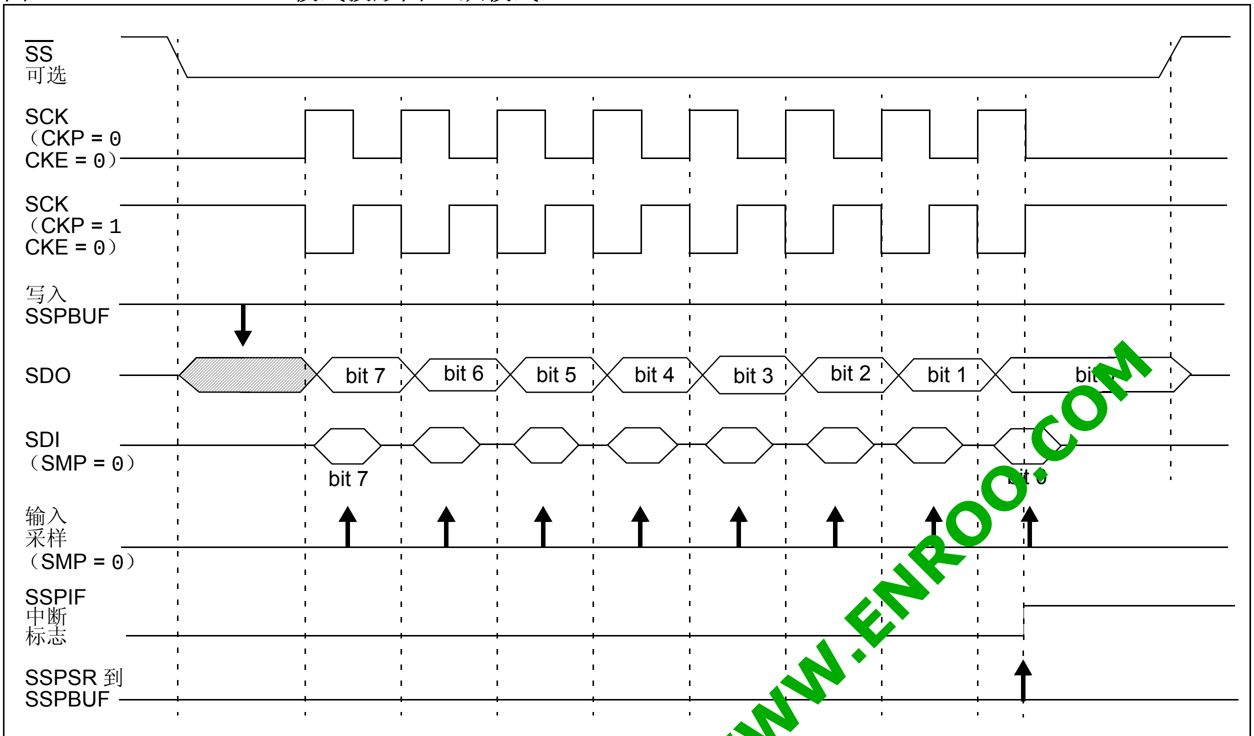
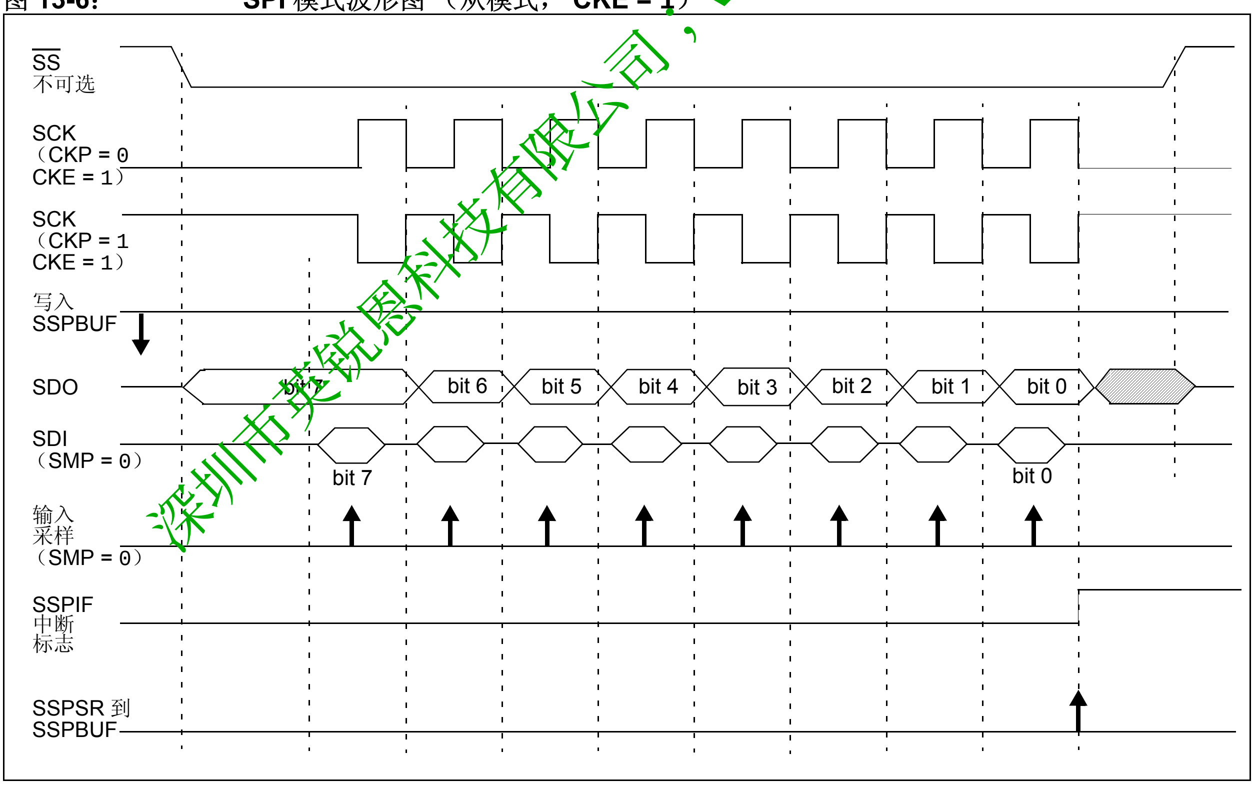


图 13-6: SPI 模式波形图（从模式， CKE = 1）



13.8 休眠模式下的操作

在主模式下，进入休眠模式后所有模块的时钟都停止，在器件被唤醒前，发送 / 接收也将保持原先的状态。器件恢复正常工作模式后，模块将继续发送 / 接收数据。

在从模式下， SPI 发送 / 接收移位寄存器与器件异步工作。这可以使器件在休眠状态时，仍可使数据被移入 SPI 发送 / 接收移位寄存器。当接收完 8 位数据后，SSP 中断标志位将置 1，如果此时该中断被允许，将从休眠状态唤醒器件。

13.9 复位的影响

复位会禁止 SSP 模块并终止当前的数据传输。

13.10 总线模式兼容性

表 13-1 显示了标准 SPI 模式与 CKP 和 CKE 控制位状态之间的兼容性。

表 13-1： SPI 总线模式

标准 SPI 模式术语	控制位状态	
	CKP	CKE
0, 0	0	1
0, 1	0	0
1, 0	1	1
1, 1	1	0

此外还有 SMP 位用于控制采样时间。

表 13-2： 与 SPI 操作相关的寄存器 (1)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
0Bh/8Bh/10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
13h	SSPBUF	同步串行口接收缓冲区 / 发送寄存器								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
86h/186h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
87h/187h	TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	1111 1111	1111 1111
8Ch	PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
94h	SSPSTAT	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000

图注： x = 未知， u = 不变， – = 未实现 （读为 0）。在 SPI 模式下， SSP 不使用阴影单元。

注 1： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。

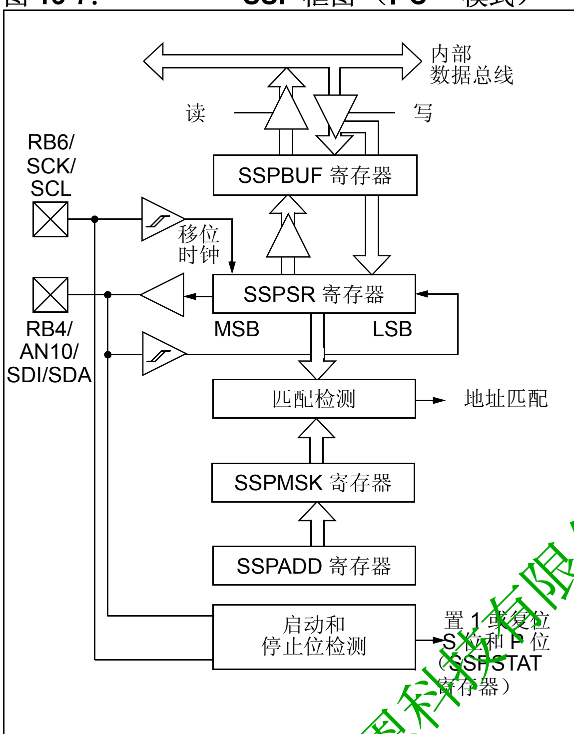
13.11 SSP I²C™ 工作原理

除了全局呼叫支持外，I²C 模式的 SSP 模块几乎完全实现了所有的从功能，并提供了启动位和停止位的硬件中断，帮助固件实现主功能。SSP 模块实现标准模式规范及 7 位和 10 位寻址。

数据传输使用了两个引脚。 RB6/SCK/SCL 引脚作为时钟线（SCL），而 RB4/AN10/SDI/SDA 引脚作为数据线（SDA）。

通过将 SSP 使能位 SSPEN （SSPCON<5>）置 1，可以使能 SSP 模块的功能。

图 13-7： SSP 框图 （I²C™ 模式）



SSP 模块有 6 个寄存器用于 I²C 操作。它们是：

- SSP 控制寄存器 （SSPCON）
- SSP 状态寄存器 （SSPSTAT）
- 串行接收 / 发送缓冲区 （SSPBUF）
- SSP 移位寄存器 （SSPSR）——不可直接访问
- SSP 地址寄存器 （SSPADD）
- SSP 屏蔽寄存器 （SSPMSK）

通过 SSPCON 寄存器可以控制 I²C 的工作状况。通过 4 个模式选择位 （SSPCON<3:0>）可以选择以下 I²C 模式之一：

- I²C 从模式 （7 位地址）
- I²C 从模式 （10 位地址）
- I²C 从模式 （7 位地址），允许启动和停止位中断，支持固件主模式
- I²C 从模式 （10 位地址），允许启动和停止位中断，支持固件主模式
- 允许 I²C 启动和停止位中断，支持固件主模式；从模式空闲

在 SSPEN 位置 1 时选择任何 I²C 模式都将强制 SCL 和 SDA 引脚为漏极开路，前提是将相应的 TRISB 位置 1，使这些引脚编程为输入引脚。 SCL 和 SDA 引脚上必须外接上拉电阻，才能使 I²C 模块正常工作。

13.12 从模式

在从模式下，必须将 SCL 和 SDA 引脚配置为输入（TRISB<6,4> 置 1）。必要时 SSP 模块将用输出数据改写输入状态 （从发送器）。

当地址匹配或接收到地址匹配后发送的数据时，硬件会自动产生一个应答 （$\overline{\text{ACK}}$）脉冲，并把当时 SSPSR 寄存器接收到的值装入 SSPBUF 寄存器。

某些条件会使 SSP 模块不发出此 $\overline{\text{ACK}}$ 脉冲。这些条件包括 （之一或两者）：

a) SSPSTAT 寄存器的缓冲区满位 BF 在接收到传输数据前置 1。
b) SSPCON 寄存器的溢出位 SSPOV 在接收到传输数据前置 1。

在这种情况下，SSPSR寄存器的值不会装入SSPBUF，但 PIR1 寄存器的 SSPIF 位会置 1。表 13-3 给出了根据 BF 和 SSPOV 位的状态，接收到数据传输字节时产生的结果。阴影单元显示了当用户软件没有正确将溢出状态清除时的情况。当 SSPOV 位用软件清零时，通过读 SSPBUF 寄存器可以将标志位 BF 清零。

为确保正常工作，SCL 时钟输入必须满足最小高电平和最小低电平时间要求。关于 I²C 规范的高低电平时间和对 SSP 模块的要求，请参见**第 17.0 节 “电气规范”**。

13.12.1 寻址

一旦 SSP 模块被使能，它就会等待启动条件发生。启动条件发生后， 8 位数据被移入 SSPSR 寄存器。在时钟（SCL）线的上升沿采样所有的输入位。寄存器 SSPSR<7:1> 的值会和 SSPADD 寄存器的值比较，该比较是在第 8 个时钟（SCL）脉冲下降沿进行的。如果地址匹配，并且 BF 和 SSPOV 位都被清零，会发生以下事件：

a) SSPSR 寄存器的值被装入 SSPBUF 寄存器。
b) 缓冲区满标志位 BF 被置 1。
c) 产生 $\overline{\text{ACK}}$ 脉冲。
d) 在第 9 个 SCL 脉冲的下降沿， PIR1 寄存器的 SSP 中断标志位 SSPIF 被置 1（如果允许中断，则产生中断）。

在 10 位地址模式下，从单元需要接收两个地址字节（图 13-8）。第一个地址字节的高 5 位（MSb）将指定这是否是一个 10 位地址。R/$\overline{\text{W}}$ 位（SSPSTAT<2>）必须指定写操作，这样从器件才能接收到第二个地址字节。对于 10 位地址，第一个字节等于 “`1111 0 A9 A8 0`”，其中 A9 和 A8 是该地址的两个 MSb。

10 位地址的事件顺序如下，其中 7-9 步是针对从发送器而言的：

1. 接收地址的第一个（高）字节（SSPIF 位、BF 位和 UA 位（SSPSTAT<1>）置 1）。
2. 用地址的第二个（低）字节更新 SSPADD 寄存器（UA 位清零并释放 SCL 线）。
3. 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。
4. 接收地址的第二个（低）字节（SSPIF 位、BF 位和 UA 位置 1）。
5. 用地址的第一个（高）字节更新 SSPADD 寄存器。如果匹配，则释放 SCL 线，这将清零 UA 位。
6. 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。
7. 接收重复启动条件。
8. 接收地址的第一个（高）字节（SSPIF 位和 BF 位置 1）。
9. 读 SSPBUF 寄存器（BF 位清零）并将标志位 SSPIF 清零。

表 13-3： 数据传输中接收到字节的动作

接收到传输数据时的状态位		SSPSR → SSPBUF	产生 $\overline{\text{ACK}}$ 脉冲	SSPIF 位置 1（如果允许，还将产生 SSP 中断）
BF	**SSPOV**			
0	0	是	是	是
1	0	否	否	是
1	1	否	否	是
0	1	否	否	是

注： 阴影单元显示了当用户软件没有正确将溢出状态清除时的情况。

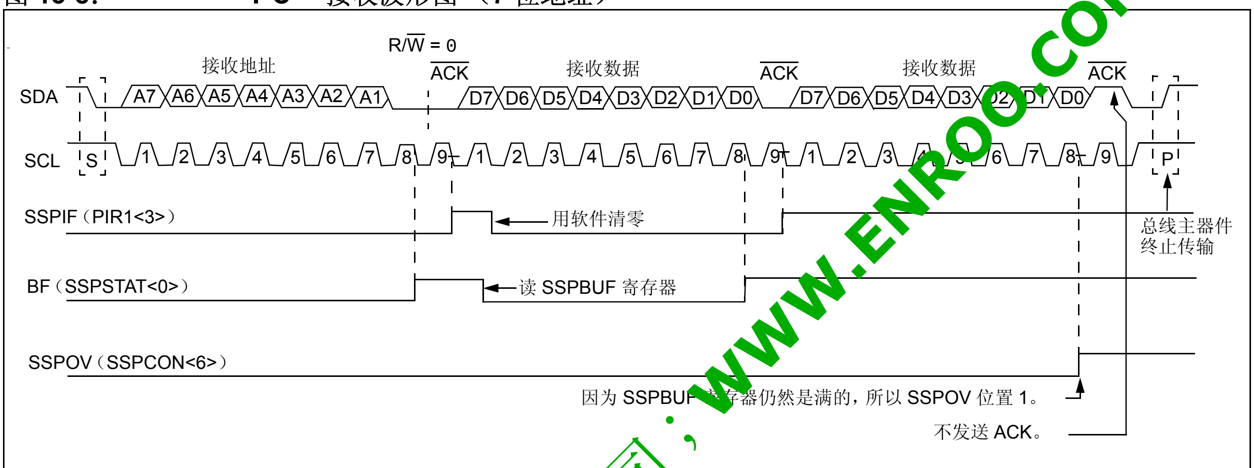
13.12.2 接收

当地址字节的 R/$\overline{W}$ 位清零并发生地址匹配时，SSPSTAT 寄存器中的 R/$\overline{W}$ 位清零。接收到的地址被装入 SSPBUF 寄存器。

当发生地址字节溢出时，则不会产生应答（$\overline{ACK}$）脉冲。溢出条件是指 SSPSTAT 寄存器的 BF 位置 1 或 SSPCON 寄存器的 SSPOV 位置 1。这是一个由于用户固件导致的错误状况。

每个数据传输字节都会产生 SSP 中断。PIR1 寄存器的标志位 SSPIF 必须用软件清零。通过 SSPSTAT 寄存器可以确定该字节的状态。

图 13-8： I²C™ 接收波形图（7 位地址）



13.12.3 SSP 屏蔽寄存器

SSP 屏蔽（SSPMSK）寄存器在 I^{2}C 从模式下可用，用作地址比较操作期间 SSPSR 寄存器中保存的值的屏蔽。SSPMSK 寄存器中的零 （0）位会影响对 SSPSR 寄存器中相应位的忽略。

任何复位后，该寄存器都会复位到全 1 状态，因此，在写入屏蔽值之前对标准 SSP 操作没有影响。

在将 SSPM<3:0> 位置 1 前必须初始化该寄存器，以便选择 I^{2}C 从模式 （7 位或 10 位地址）。

只有通过某些位 （SSPCON 的 SSPM<3:0>）选择了适当模式后，才能访问该寄存器。

SSP 屏蔽寄存器在以下期间保持活动状态：

- 7 位地址模式：A<7:1> 的地址比较。
- 10 位地址模式：仅针对 A<7:0> 的地址比较。在接收地址的第一个 （高）字节期间， SSP 屏蔽没有影响。

寄存器 13-3： SSPMSK：SSP 屏蔽寄存器 [1]

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0[2]
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

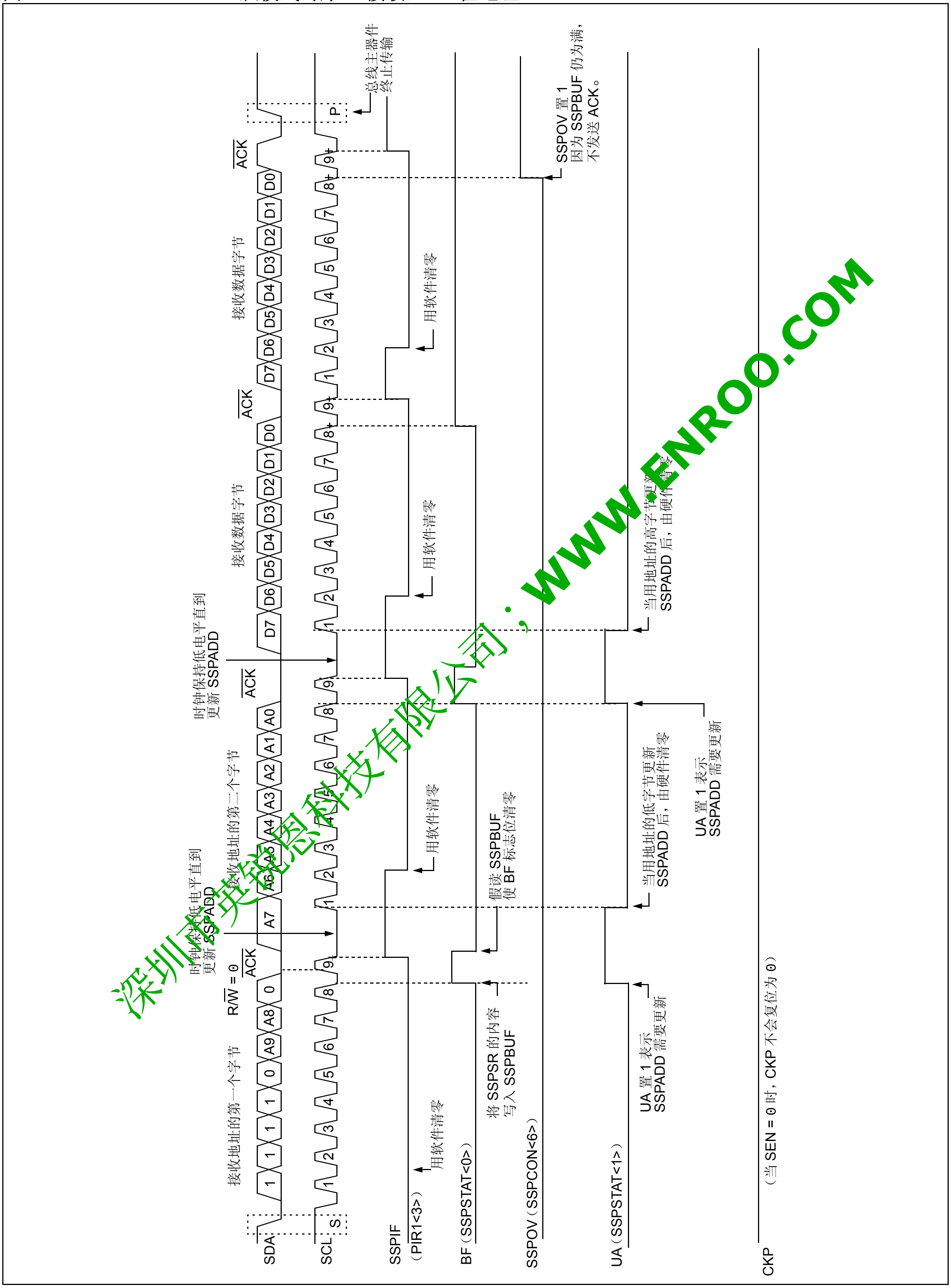
bit 7-1 **MSK<7:1>**：屏蔽位
1 = 接收到的地址 bit n 与 SSPADD<n> 相比较来检测 I^{2}C 地址匹配
0 = 接收到的地址 bit n 不用于检测 I^{2}C 地址匹配

bit 0 **MSK<0>**：屏蔽位用于 I^{2}C 从模式， 10 位地址 [2]
I^{2}C 从模式， 10 位地址 （SSPM<3:0> = 0111）：
1 = 接收到的地址 bit 0 与 SSPADD<0> 相比较来检测 I^{2}C 地址匹配
0 = 接收到的地址 bit 0 不用于检测 I^{2}C 地址匹配

注 1： 当 SSPCON 位 SSPM<3:0> = 1001 时，任何对 SSPADD SFR 地址的读或写都必须通过 SSPMSK 寄存器来访问。访问 SSPMSK 寄存器时， SSPCON 寄存器的 SSPEN 位应为零。

2： 在所有其他 SSP 模式下，该位不起作用。

图 13-9： I^2C™ 从模式时序 （接收， 10 位地址）



13.12.4 发送

当进入地址字节的 R/$\overline{W}$ 位置 1 并发生地址匹配时，SSPSTAT 寄存器的 R/$\overline{W}$ 位被置 1。接收到的地址被装入 SSPBUF 寄存器。$\overline{ACK}$ 脉冲在第 9 位上发送，RB6/SCK/SCL 引脚保持低电平。发送数据必须被装入 SSPBUF 寄存器，同时也装入 SSPSR 寄存器。然后，应该通过将 CKP 位（SSPCON<4>）位置 1 来使能 RB6/SCK/SCL 引脚。主器件必须在发出另一个时钟脉冲前监视 SCL 引脚。从器件可以通过延时时钟低电平时间不与主器件同步。8 个数据位在 SCL 输入的下降沿被移出。这可以确保在 SCL 为高电平期间 SDA 信号是有效的（图 13-10）。

每个数据传输字节都会产生 SSP 中断。标志位 SSPIF 必须用软件清零，SSPSTAT 寄存器用于确定该字节的状态。标志位SSPIF在第9个时钟脉冲的下降沿被置1。

对于从发送器，来自主接收器的 $\overline{ACK}$ 脉冲将在第 9 个 SCL 输入脉冲的上升沿被锁存。如果 SDA 线为高电平（无 $\overline{ACK}$），则表示数据传输已完成。如果从控制器锁存了 ACK，将复位从逻辑（复位 SSPSTAT 寄存器），同时从控制器监控下一个启动位的出现。如果 SDA 线为低电平（ACK），则必须将要发送的数据装入 SSPBUF 寄存器，同时也装入 SSPSR 寄存器。然后，通过将 CKP 位置 1 使能 RB6/SCK/SCL 引脚。

图 13-10：　　I²C™ 发送波形图（7 位地址）

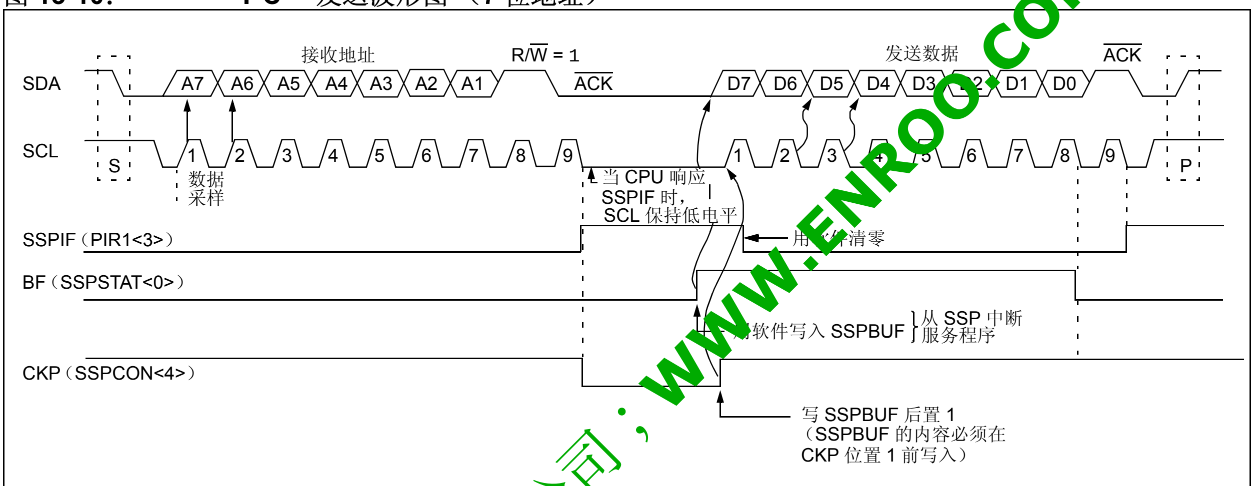
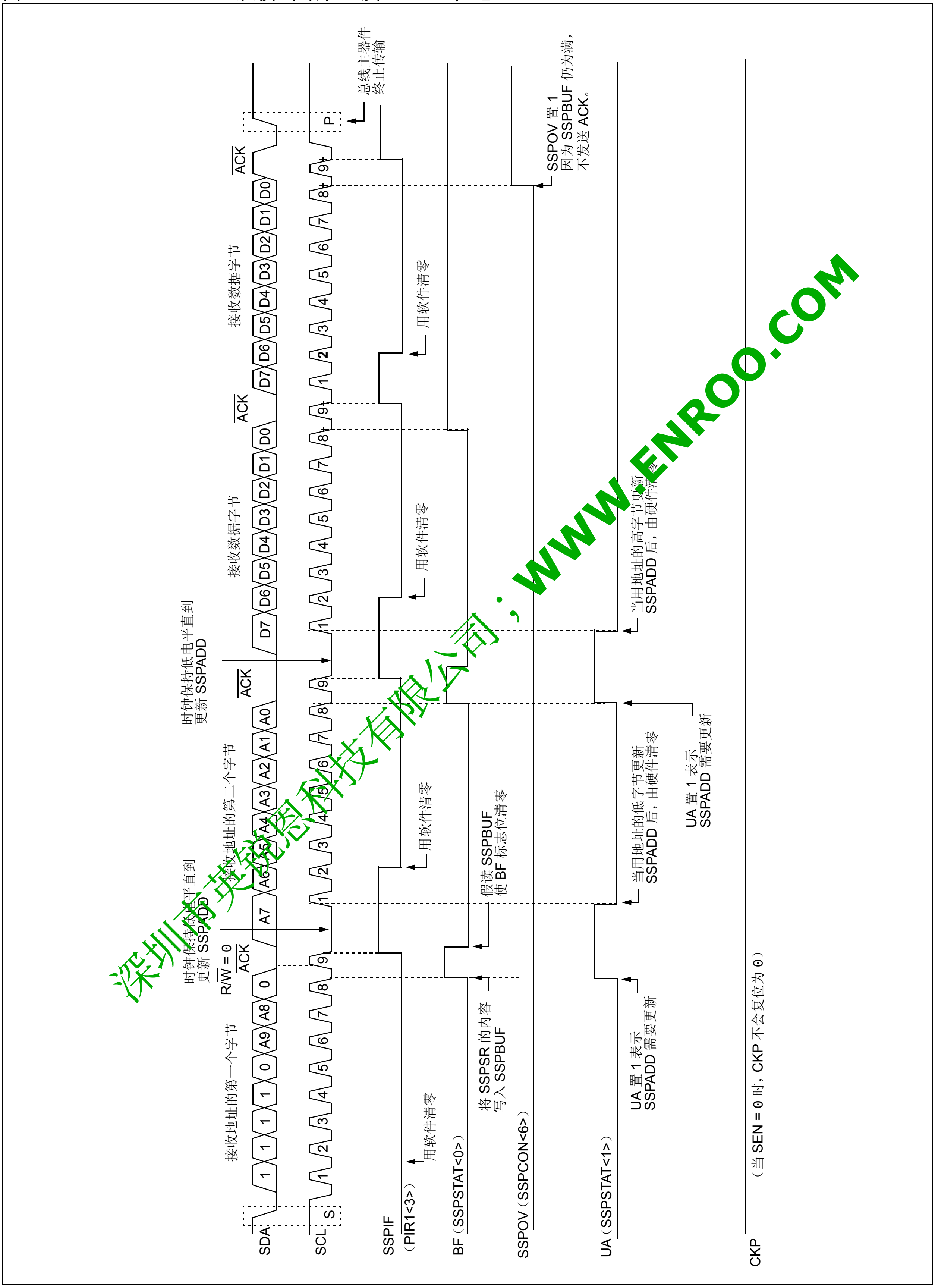


图 13-11： I^2C™ 从模式时序 （发送， 10 位地址）



13.13 主模式

主模式通过固件在检测到启动和停止条件时产生中断来工作。停止（P）位和启动（S）位在复位时或禁止 SSP 模块时清零。停止 （P）位和启动 （S）位会根据启动和停止条件切换。当 P 位置 1 时，可以获得 I^2C 总线的控制权；否则， S 和 P 位都清零，总线处于空闲状态。

在主模式下， SCL 和 SDA 线通过清零相应的 TRISB<6,4> 位来控制。输出电平始终为低电平，而与 PORTB<6,4> 的值无关。因此当发送数据时，必须用 `1` 数据位将 TRISB<4> 位置 1 （输入），并且用 `0` 数据位将 TRISB<4> 位清零 （输出）。对于 SCL 线，也可使用 TRISB<6> 位采用同样的方法进行控制。SCL 和 SDA 引脚上必须外接上拉电阻，才能使 I^2C 模块正常工作。

以下事件会使 SSP 中断标志位 SSPIF 置 1 （如果允许 SSP 中断，则产生中断）：

- 启动条件
- 停止条件
- 数据字节发送 / 接收

可用从模式空闲 （SSPM<3:0> = `1011`）或从模式活动完成主模式操作。当同时使能主模式和从模式时，需要使用软件区分中断源。

13.14 多主模式

在多主模式下，在检测到启动和停止条件时产生的中断可用于判断总线是否空闲。停止 （P）位和启动 （S）位在复位时或禁止 SSP 模块时清零。停止 （P）位和启动 （S）位会根据启动和停止条件切换。当 P 位 （SSPSTAT<4>）置 1 时，可以获得 I^2C 总线的控制权；否则， S 和 P 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。

在多主操作中，必须监视 SDA 线以确定信号电平是否为所需的输出电平。此检查仅需在输出为高电平时进行。如果期望输出高电平，但检测到的是低电平，器件就需要释放 SDA 和 SCL 线（TRISB<6,4> 位置 1）。此仲裁在以下两个阶段可能会失败：

- 地址传输
- 数据传输

当使能从逻辑时，从控制器将继续接收数据。如果在地址传输阶段仲裁失败，可能表示与该器件的通信正在进行中。如果寻址到器件，则会产生一个 ACK 脉冲。如果在数据传输阶段仲裁失败，则器件需要在以后重新传输数据。

13.14.1 时钟同步与 CKP 位

当 CKP 位清零时将迫使 SCL 输出为 `0`；但是，在 SCL 输出采样值为低之前将 CKP 位置 1 并不会将 SCL 输出拉低。因此， CKP 位不会将 SCL 信号拉为低电平，除非外部 I^2C 主器件将 SCL 线拉低。 SCL 输出将保持低电平，直到 CKP 位置 1 且 I^2C 总线上的其他器件将 SCL 电平拉高为止。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 13-12）。

图 13-12： 时钟同步时序

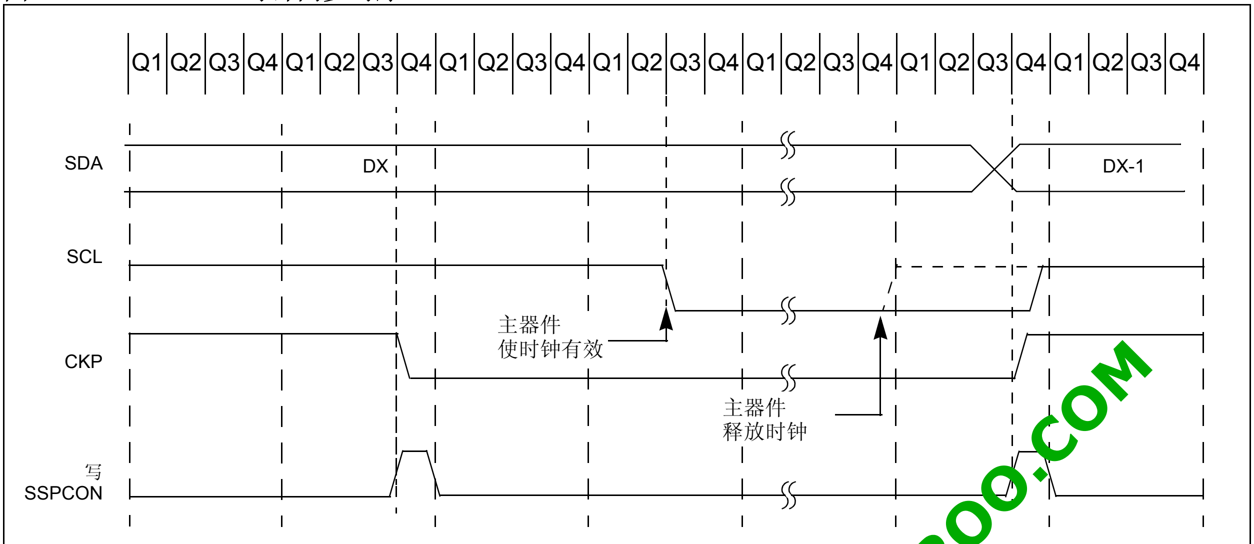


表 13-4： 与 I²C™ 操作相关的寄存器 (1)

地址	名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
0Bh/8Bh/10Bh/18Bh	INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
13h	SSPBUF	同步串行口接收缓冲区 / 发送寄存器								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
93h	SSPMSK[2]	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111
94h	SSPSTAT	SMP[3]	CKE[3]	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	0000 0000	0000 0000
8Ch	PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IF	TMR1IF	-000 0000	-000 0000

图注： — = 未实现单元 （读为 0）， u = 不变， x = 未知。 SSP 模块不使用阴影单元。

注 1： 仅限 PIC16F677/PIC16F687/PIC16F689/PIC16F690。

2： SSPMSK 寄存器 （寄存器 13-3）可以在 SSPM 位 <3:0> = 1001 时读或写 SSPADD 寄存器来进行访问。更多详细信息，请参见寄存器 13-2 和 13-3。

3： 保持这些位清零。

14.0 CPU 的特殊功能

PIC16F631/677/685/687/689/690 有许多功能，旨在最大限度地提高系统可靠性，通过减少外部元件将成本降至最低，并提供省电工作模式和代码保护功能。

这些功能包括：

- 复位
  - 上电复位 （POR）
  - 上电延时定时器 （PWRT）
  - 振荡器起振定时器 （OST）
  - 欠压复位 （BOR）
- 中断
- 看门狗定时器 （WDT）
- 振荡器选择
- 休眠
- 代码保护
- ID 存储单元
- 在线串行编程

PIC16F631/677/685/687/689/690 有两个定时器提供必要的上电延时。一个是振荡器起振定时器 （OST），旨在确保芯片在晶振达到稳定之前始终处于复位状态。另一个是上电延时定时器 （PWRT），仅在上电时提供 64 ms （标称值）的固定延时，用来确保器件在供电电压稳定之前处于复位状态。还有当器件发生欠压时使器件复位的电路，该电路可使用上电延时定时器，提供至少 64 ms 的复位延时。有了这三种片上功能，绝大多数应用就无需再外接复位电路了。

休眠模式的设计是为了提供了电流极低的掉电模式。用户可通过以下方法将器件从休眠模式唤醒：

- 外部复位
- 看门狗定时器唤醒
- 中断

有几种振荡器模式可供选择，以使器件适应各种应用。选择 INTOSC 可节约系统成本，而选择 LP 晶振可以节能。通过配置位的设定可选择不同选项（见寄存器 14-2）。

14.1 配置位

可以通过对配置位编程 （读为 0）或不编程 （读为 1）来选择不同的器件配置，如寄存器 14-2 所示。这些位映射到程序存储单元 2007h 中。

注：	地址2007h超出了用户程序存储空间范围。它属于特殊配置存储空间（2000h-3FFFh），只能在编程时对其进行访问。更多信息，请参见“*PIC12F6XX/16F6XX Memory Programming Specification*”（DS41204）。

寄存器 14-1： CONFIG：配置字寄存器

保留	保留	FCMEN	IESO	BOREN1(1)	BOREN0(1)	$\overline{CPD}$(2)
bit 13						bit 7

$\overline{CP}$(3)	MCLRE(4)	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0
bit 6						bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	U = 未实现位，读为 0
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 13-12 **保留**：保留位。不得使用。

bit 11 **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 10 **IESO**：内 / 外部切换位
1 = 使能内 / 外部切换模式
0 = 禁止内 / 外部切换模式

bit 9-8 **BOREN<1:0>**：欠压复位选择位 (1)
11 = 使能 BOR
10 = 正常工作时使能 BOR，休眠时禁止 BOR
01 = BOR 由 PCON 寄存器的 SBOREN 位控制
00 = 禁止 BOR

bit 7 **$\overline{CPD}$**：数据代码保护位 (2)
1 = 禁止数据存储器代码保护
0 = 使能数据存储器代码保护

bit 6 **$\overline{CP}$**：代码保护位 (2)
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 5 **MCLRE**：$\overline{MCLR}$ 引脚功能选择位 (4)
1 = $\overline{MCLR}$ 引脚功能为 $\overline{MCLR}$
0 = MCLR 引脚功能为数字输入，$\overline{MCLR}$ 内部连接到 VDD

bit 4 **$\overline{PWRTE}$**：上电延时定时器使能位
1 = 禁止 PWRT
0 = 使能 PWRT

bit 3 **WDTE**：看门狗定时器使能位
1 = 使能 WDT
0 = 禁止 WDT

bit 2-0 **FOSC<2:0>**：振荡器选择位
111 = RC 振荡器：RA4/OSC2/CLKOUT 引脚为 CLKOUT 功能，RA5/OSC1/CLKIN 引脚上连接 RC
110 = RCIO 振荡器：RA4/OSC2/CLKOUT 引脚为 I/O 功能，RA5/OSC1/CLKIN 引脚上连接 RC
101 = INTOSC 振荡器：RA4/OSC2/CLKOUT 引脚为 CLKOUT 功能，RA5/OSC1/CLKIN 引脚为 I/O 功能
100 = INTOSCIO 振荡器：RA4/OSC2/CLKOUT 引脚为 I/O 功能，RA5/OSC1/CLKIN 引脚为 I/O 功能
011 = EC：RA4/OSC2/CLKOUT 引脚为 I/O 功能，RA5/OSC1/CLKIN 引脚上连接 CLKIN
010 = HS 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接高速晶振 / 谐振器
001 = XT 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接晶振 / 谐振器
000 = LP 振荡器：RA4/OSC2/CLKOUT 和 RA5/OSC1/CLKIN 引脚上连接低功耗晶振

注 1： 使能欠压复位并不能自动使能上电延时定时器。
2： 当关闭代码保护时，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。
4： 当 $\overline{MCLR}$ 在 INTOSC 或 RC 模式下被拉为低电平时，将禁止内部时钟振荡器。

14.2 复位

PIC16F631/677/685/687/689/690 有以下几种不同类型的复位：

a) 上电复位（POR）
b) 正常工作期间的 WDT 复位
c) 休眠期间的 WDT 复位
d) 正常工作期间的 MCLR 复位
e) 休眠期间的 MCLR 复位
f) 欠压复位（BOR）

有些寄存器不受任何复位的影响；在上电复位时它们的状态未知，而在其他复位时状态不变。大多数寄存器在以下复位时会复位到各自的“复位状态”：

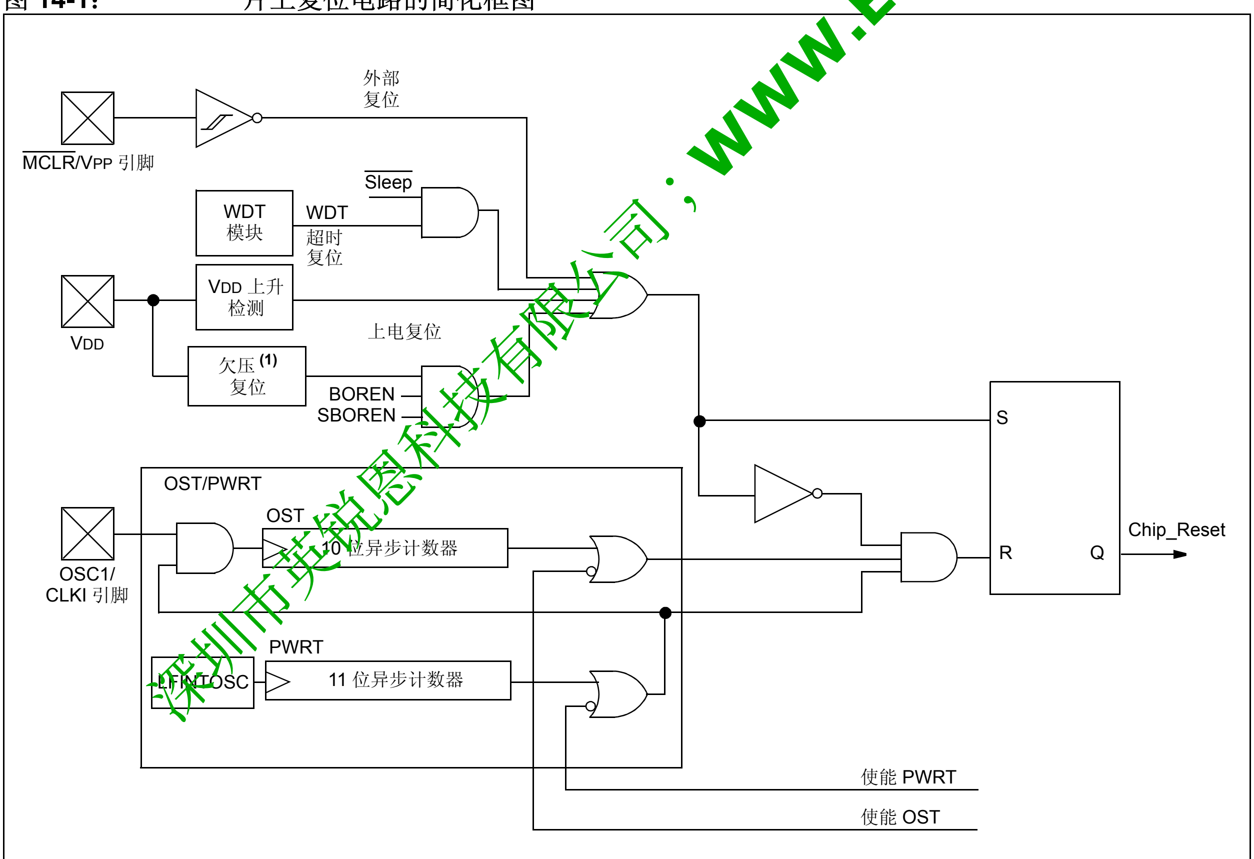
- 上电复位
- MCLR 复位
- 休眠期间的 MCLR 复位
- WDT 复位
- 欠压复位（BOR）

它们不受 WDT 唤醒的影响，因为这被视为恢复正常工作。如表 14-2 所示，TO 和 PD 位在不同的复位情形下会分别被置 1 或清零。软件可以使用这些位来判断复位的性质。关于所有寄存器的复位状态的完整说明，请参见表 14-4。

图 14-1 给出了片上复位电路的简化框图。

MCLR 复位路径上有一个噪声滤波器，用来检测并滤除小脉冲。关于脉冲宽度规范，请参见**第 17.0 节“电气规范”**。

图 14-1： 片上复位电路的简化框图



注 1：请参见配置字寄存器（寄存器 14-1）。

14.2.1 上电复位（POR）

在 VDD 达到足以使器件正常工作的电平之前，片上上电复位电路将使器件保持在复位状态。需要一个最大上升时间才能达到 VDD。详情请参见**第 17.0 节“电气规范”**。如果使能了欠压复位，那么该最大上升时间规范将不再适用。BOR 电路将使器件保持在复位状态，直到 VDD 达到 VBOR（见**第 14.2.4 节“欠压复位（BOR）”**）。

> **注：** 当 VDD 降低时，POR 电路不会产生内部复位。要重新使能 POR，VDD 必须至少保持 100 µs 的 Vss 电压。

当器件开始正常工作（退出复位状态）时，器件的工作参数（即电压、频率和温度等）必须得到满足，以确保其正常工作。如果不满足这些条件，那么器件必须保持在复位状态，直到满足工作条件为止。

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

14.2.2 $\overline{\text{MCLR}}$

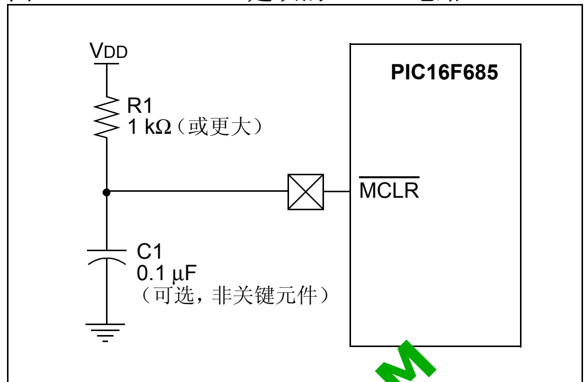
PIC16F631/677/685/687/689/690 在 $\overline{\text{MCLR}}$ 复位路径中有一个噪声滤波器。该滤波器检测并滤除小脉冲。

应该注意，WDT 复位不会将 $\overline{\text{MCLR}}$ 引脚驱动为低电平。

$\overline{\text{MCLR}}$ 引脚上 ESD 保护的工作原理与该系列早期器件有所不同。施加在该引脚上的电压超过规范值将导致 $\overline{\text{MCLR}}$ 复位，并且在 ESD 事件中产生的电流也将超过器件的规范值。因此，Microchip 建议不要把 $\overline{\text{MCLR}}$ 引脚直接连接到 VDD。建议使用图 14-2 给出的 RC 网络。

通过清零配置字寄存器中的 MCLRE 位，可使能内部 $\overline{\text{MCLR}}$ 选项。当 MCLRE = 0 时，在内部产生芯片的复位信号。当 MCLRE = 1 时，RA3/$\overline{\text{MCLR}}$ 引脚变成外部复位输入。在这种模式下，RA3/$\overline{\text{MCLR}}$ 引脚具有到 VDD 的弱上拉。但是，为了能在噪声环境下可靠工作，仍推荐采用图 14-2 所示的电路。

图 14-2： 建议的 $\overline{\text{MCLR}}$ 电路



14.2.3 上电延时定时器（PWRT）

上电延时定时器仅在上电时（上电复位或欠压复位）提供一个 64 ms（标称值）的固定延时。上电延时定时器采用 LFINTOSC 振荡器作为时钟源，工作频率为 31 kHz。更多信息，请参见**第 3.5 节“内部时钟模式”**。只要 PWRT 处于活动状态，芯片就保持在复位状态。PWRT 延时使 VDD 有足够的时间上升到所需的电平。配置位 $\overline{\text{PWRTE}}$ 可以禁止（如果置 1）或使能（如果清零或被编程）上电延时定时器。虽然不是必需的，但是在使能欠压复位时也应使能上电延时定时器。

由于以下原因不同芯片的上电延时定时器的延时也各不相同：

- VDD 差异
- 温度差异
- 制造工艺差异

详情请参见直流参数（**第 17.0 节“电气规范”**）。

14.2.4 欠压复位 （BOR）

配置字寄存器中的 BOREN0 和 BOREN1 位用于选择 4 种欠压复位模式中的一种。其中添加了两种允许使用软件或硬件对 BOR 的使能进行控制的模式。当 BOREN<1:0> = `01`时，可由SBOREN位（PCON<4>）使能 / 禁止 BOR，从而能用软件对其进行控制。通过选择 BOREN<1:0>，可使欠压复位在休眠时被自动禁止，从而节约功耗；而在唤醒后被重新使能。在此模式下， SBOREN 位被禁止。关于配置字的定义，请参见寄存器 14-2。

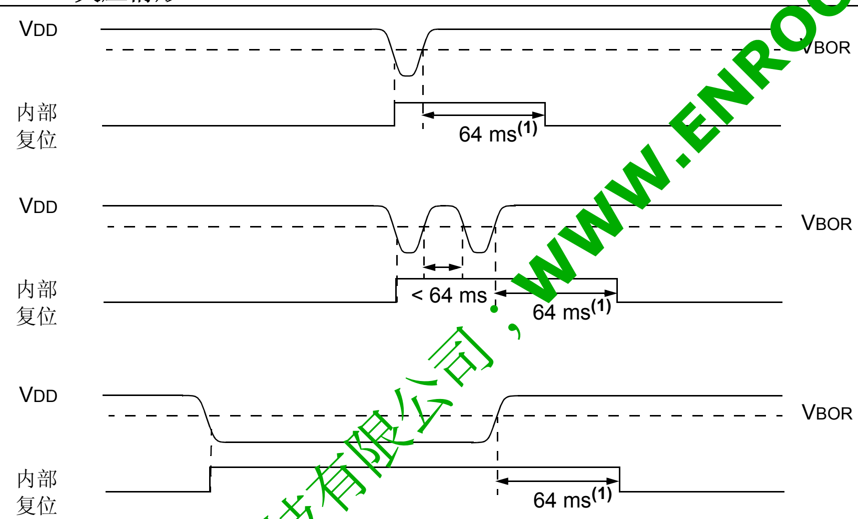
如果 VDD 下降到 VBOR 以下，且持续时间超过参数值（TBOR）（见**第 17.0 节 “电气规范”**），欠压状况将使器件复位。不管 VDD 的变化速率如何，上述情况都会发生。如果 VDD 低于 VBOR 的时间少于参数值 （TBOR），则不一定会发生复位。

任何复位 （上电复位、欠压复位或看门狗定时器复位等）都会使器件保持复位状态，直到 VDD 上升到 V 以上（见图 14-3）。如果使能了上电延时定时器，此时它将启动，并且会使器件保持复位状态的时间延长 64 ms。

> **注：** 配置字寄存器中的 $\overline{\text{PWRTE}}$ 位用于使能上电延时定时器。

如果在上电延时定时器运行过程中， VDD 降低到 VBOR 以下，芯片将重新回到欠压复位状态并且上电延时定时器会恢复为初始状态。一旦 VDD 上升到 VBOR 以上，上电延时定时器将执行一段 64 ms 的复位。

图 14-3： 欠压情形



注 1：仅在 $\overline{\text{PWRTE}}$ 位被编程为 0 时，才增加 64 ms 延时。

14.2.5 延时时序

上电延时时序如下：首先，在 POR 延时到期后，施加一段 PWRT 延时，随后振荡器起振。总延时时间取决于振荡器配置和 $\overline{\mathrm{PWRTE}}$ 位的状态。例如，在 EC 模式且 $\overline{\mathrm{PWRTE}}$ 位被擦除（PWRT 禁止）的情况下，根本不会出现延时。图 14-4、14-5 和 14-6 分别描绘各种情形下的延时时序。当振荡器起振后，通过使能双速启动或故障保护监视器，器件将以 INTOSC 作为时钟源来执行代码（见第 **3.7.2** 节 **“双速启动顺序”** 和第 **3.8** 节 **“故障保护时钟监视器”**）。

由于延时是由 POR 脉冲触发的，因此如果 $\overline{\mathrm{MCLR}}$ 保持足够长时间的低电平，所有延时都将结束。将 $\overline{\mathrm{MCLR}}$ 电平拉高后，器件将立即开始执行代码（见图 14-5）。这对于测试或同步多个并行工作的 PIC16F631/677/685/687/689/690 器件来说是非常有用的。

表 14-5 给出了一些特殊寄存器的复位条件，而表 14-4 给出了所有寄存器的复位条件。

14.2.6 电源控制 （PCON）寄存器

电源控制寄存器 PCON（地址 8Eh）有两个状态位，用于指示上次发生的复位的类型。

bit 0 是 $\overline{\mathrm{BOR}}$（欠压复位）标志位。$\overline{\mathrm{BOR}}$ 在上电复位时未知。然后，用户必须将该位置 1，并在随后的复位发生时检查中 $\overline{\mathrm{BOR}}$ 是否为 0，如果是，则表示已发生欠压复位。当禁止欠压复位电路（配置字寄存器中的 BOREN<1:0> = 00）时，$\overline{\mathrm{BOR}}$ 状态位为“无关位”并且不一定预测得到。

bit 1 是 $\overline{\mathrm{POR}}$（上电复位）标志位。在上电复位时它的值为 0，其他情况下不受影响。上电复位后，用户必须对该位写 1。发生后续复位后，如果 $\overline{\mathrm{POR}}$ 为 0，则表示发生了上电复位（即 VDD 可能已经变为了低电平）。

更多信息，请参见第 **4.2.4** 节 **“超低功耗唤醒”** 和第 **14.2.4** 节 **“欠压复位（BOR）”**。

表 14-1： 各种情形下的延时

振荡器配置	上电		欠压复位		从休眠状态唤醒
	$\overline{\mathrm{PWRTE}}$ = 0	$\overline{\mathrm{PWRTE}}$ = 1	$\overline{\mathrm{PWRTE}}$ = 0	$\overline{\mathrm{PWRTE}}$ = 1	
XT、 HS 和 LP	TPWRT + 1024 • TOSC	1024 • TOSC	TPWRT + 1024 • TOSC	1024 • TOSC	1024 • TOSC
LP， T1OSCIN = 1	TPWRT	—	TPWRT	—	—
RC、 EC 和 INTOSC	TPWRT	—	TPWRT	—	—

表 14-2： **STATUS/PCON** 位及其含义

$\overline{\mathrm{POR}}$	$\overline{\mathrm{BOR}}$	$\overline{\mathrm{TO}}$	$\overline{\mathrm{PD}}$	条件
0	x	1	1	上电复位
u	0	1	1	欠压复位
u	u	0	u	WDT 复位
u	u	0	0	WDT 唤醒
u	u	u	u	正常工作期间的 $\overline{\mathrm{MCLR}}$ 复位
u	u	1	0	休眠期间的 $\overline{\mathrm{MCLR}}$ 复位

图注： u = 不变， x = 未知

表 14-3： 与欠压有关的寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
PCON	—	—	ULPWUE	SBOREN	—	—	$\overline{\mathrm{POR}}$	$\overline{\mathrm{BOR}}$	--01 --qq	--0u --uu
STATUS	IRP	RP1	RPO	$\overline{\mathrm{TO}}$	$\overline{\mathrm{PD}}$	Z	DC	C	0001 1xxx	000q quuu

图注： u = 不变， x = 未知， - = 未实现位 （读为 0）， q = 值取决于具体条件。 BOR 不使用阴影单元。
注 1： 其他 （非上电）复位包括在正常操作期间的 $\overline{\mathrm{MCLR}}$ 复位和看门狗定时器复位。

图 14-4: 上电时的延时时序（$\overline{\text{MCLR}}$ 延时）：情形 1

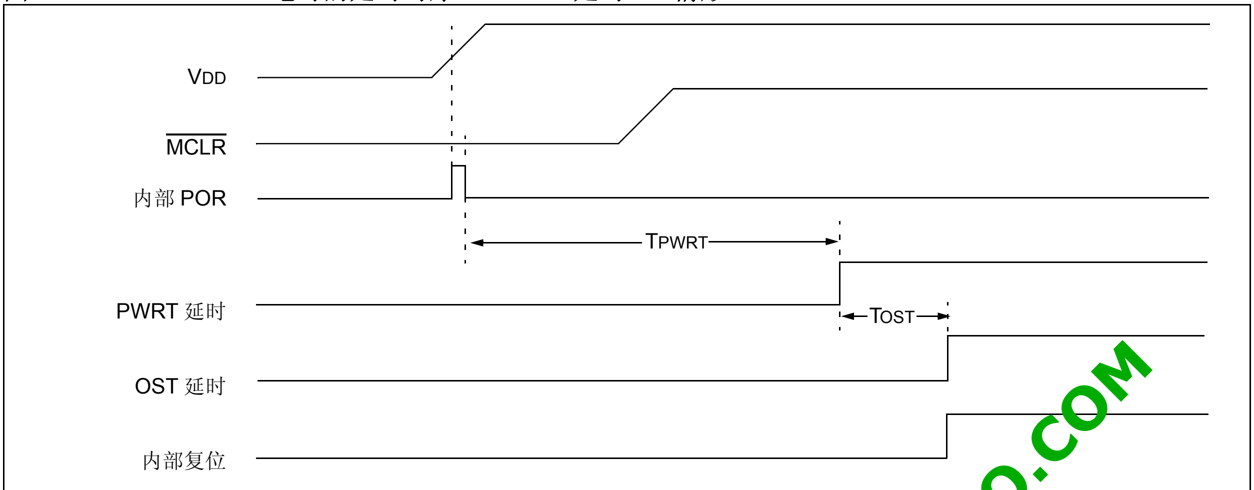


图 14-5: 上电时的延时时序（$\overline{\text{MCLR}}$ 延时）：情形 2

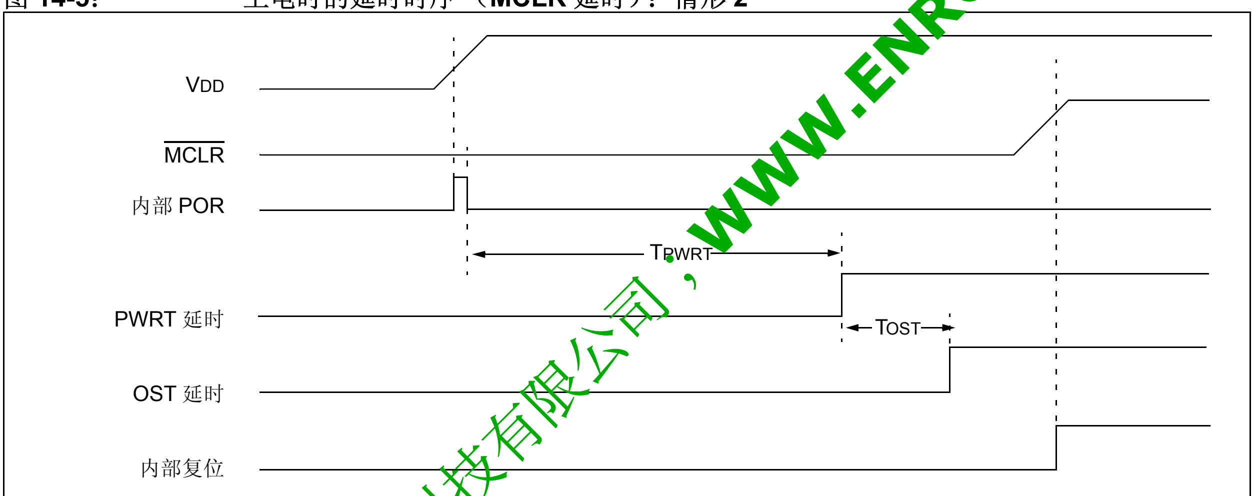


图 14-6: 上电时的延时时序（带有 VDD 的 $\overline{\text{MCLR}}$）

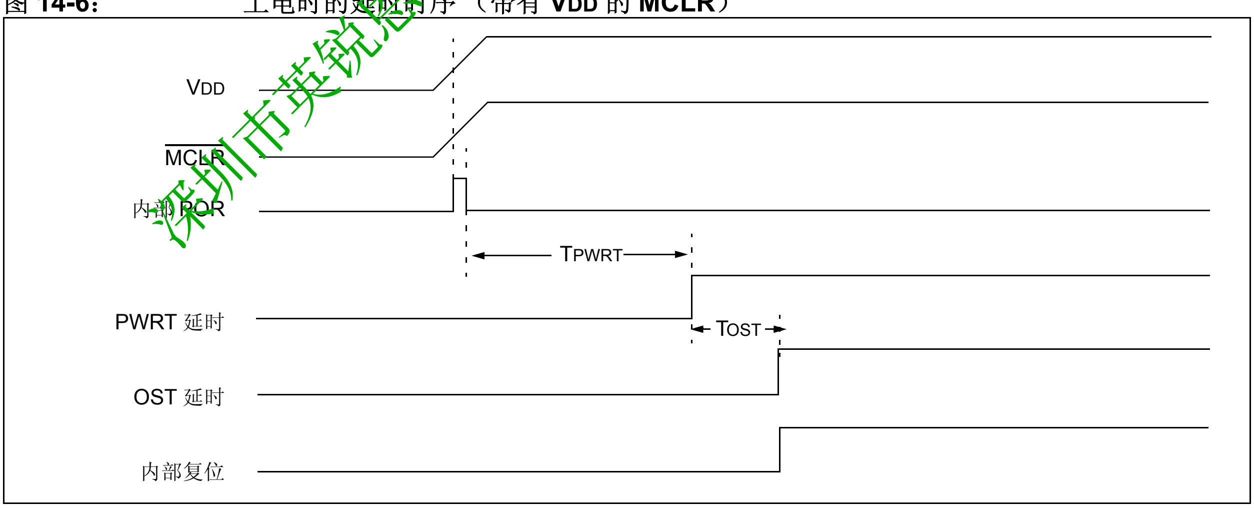


表 **14-4**： 寄存器的初始化状态

寄存器	地址	上电复位	$\overline{\text{MCLR}}$ 复位 **WDT** 复位 欠压复位 (1)	通过中断将器件从休眠状态唤醒 通过 **WDT** 超时从休眠状态唤醒
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h/80h/ 100h/180h	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0	01h/101h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h/82h/ 102h/182h	0000 0000	0000 0000	PC + 1(3)
STATUS	03h/83h/ 103h/183h	0001 1xxx	000q quuu(4)	uuuq quuu(4)
FSR	04h/84h/ 104h184h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h/105h	--xx xxxx	--uu uuuu	--uu uuuu
PORTB	06h/106h	xxxx ----	uuuu ----	uuuu ----
PORTC	07h/107h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah/8Ah/ 10Ah/18Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh/8Bh/ 10Bh/18Bh	0000 000x	0000 000u	uuuu uuuu(2)
PIR1	0Ch	-000 0000	-000 0000	-uuu uuuu(2)
PIR2	0Dh	0000 ----	0000 ----	uuuu ----(2)
TMR1L	0Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
TMR1H	0Fh	xxxx xxxx	uuuu uuuu	uuuu uuuu
T1CON	10h	0000 0000	uuuu uuuu	uuuu uuuu
TMR2	11h	0000 0000	0000 0000	uuuu uuuu
T2CON	12h	-000 0000	-000 0000	-uuu uuuu
SSPBUF	13h	xxxx xxxx	uuuu uuuu	uuuu uuuu
SSPCON	14h	0000 0000	0000 0000	uuuu uuuu
CCPR1L	15h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCPR1H	16h	xxxx xxxx	uuuu uuuu	uuuu uuuu
CCP1CON	17h	0000 0000	0000 0000	uuuu uuuu
RCSTA	18h	0000 000x	0000 000x	uuuu uuuu
TXREG	19h	0000 0000	0000 0000	uuuu uuuu
RCREG	1Ah	0000 0000	0000 0000	uuuu uuuu
PWM1CON	1Ch	0000 0000	0000 0000	uuuu uuuu
ECCPAS	1Dh	0000 0000	0000 0000	uuuu uuuu
ADRESH	1Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON0	1Fh	0000 0000	0000 0000	uuuu uuuu
OPTION_REG	81h/181h	1111 1111	1111 1111	uuuu uuuu
TRISA	85h/185h	--11 1111	--11 1111	--uu uuuu
TRISB	86h/186h	1111 ----	1111 ----	uuuu ----

图注： u = 不变， x = 未知， – = 未实现位 （读为 0）， q = 值取决于具体条件。

注 1： 如果 Vdd 过低，将激活上电复位，寄存器将受到不同的影响。
2： INTCON 和 / 或 PIR1 中的一位或多位会受到影响 （引起唤醒）。
3： 当器件被中断唤醒且 GIE 位置 1 时， PC 装入中断向量 （0004h）。
4： 具体条件下的复位值，请参见表 14-5。
5： 如果复位是由于欠压引起的，则 bit 0 = 0。所有其他复位将导致 bit 0 = u。
6： 仅在 SSPM<3:0> = 1001 时可以访问。

表 14-4： 寄存器的初始化状态 （续）

寄存器	地址	上电复位	MCLR 复位 WDT 复位 （续） 欠压复位 [1]	通过中断将器件从休眠状态唤醒 通过 WDT 超时从休眠状态唤醒
TRISC	87h/187h	1111 1111	1111 1111	uuuu uuuu
PIE1	8Ch	-000 0000	-000 0000	-uuu uuuu
PIE2	8Dh	0000 ----	0000 ----	uuuu uuuu
PCON	8Eh	--01 --0x	--0u --uq[1, 5]	--uu --uu
OSCCON	8Fh	-110 q000	-110 q000	-uuu uuuu
OSCTUNE	90h	---0 0000	---u uuuu	---u uuuu
PR2	92h	1111 1111	1111 1111	uuuu uuuu
SSPADD	93h	0000 0000	1111 1111	uuuu uuuu
SSPMSK[6]	93h	---- ----	1111 1111	uuuu uuuu
SSPSTAT	94h	0000 0000	1111 1111	uuuu uuuu
WPUA	95h	--11 -111	--11 -111	uuuu uuuu
IOCA	96h	--00 0000	--00 0000	--uu uuuu
WDTCON	97h	---0 1000	---0 1000	---u uuuu
TXSTA	98h	0000 0010	0000 0010	uuuu uuuu
SPBRG	99h	0000 0000	0000 0000	uuuu uuuu
SPBRGH	9Ah	0000 0000	0000 0000	uuuu uuuu
BAUDCTL	9Bh	01-0 0-00	01-0 0-00	uu-u u-uu
ADRESL	9Eh	xxxx xxxx	uuuu uuuu	uuuu uuuu
ADCON1	9Fh	-000 ----	-000 ----	-uuu ----
EEDAT	10Ch	0000 0000	0000 0000	uuuu uuuu
EEADR	10Dh	0000 0000	0000 0000	uuuu uuuu
EEDATH	10Eh	--00 0000	--00 0000	--uu uuuu
EEADRH	10Fh	---- 0000	---- 0000	---- uuuu
WPUB	115h	1111 ----	1111 ----	uuuu ----
IOCB	116h	0000 ----	0000 ----	uuuu ----
VRCON	118h	0000 0000	0000 0000	uuuu uuuu
CM1CON0	119h	0000 -000	0000 -000	uuuu -uuu
CM2CON0	11Ah	0000 -000	0000 -000	uuuu -uuu
CM2CON1	11Bh	00-- --00	00-- --10	uu-- --uu
ANSEL	11Eh	1111 1111	1111 1111	uuuu uuuu
ANSELH	11Fh	---- 1111	---- 1111	---- uuuu
EECON1	18Ch	x--- x000	u--- q000	---- uuuu
EECON2	18Dh	---- ----	---- ----	---- ----
PSTRCON	19Dh	---0 0001	---0 0001	---u uuuu
SRCON	19EH	0000 00--	0000 00--	uuuu uu--

图注： u = 不变， x = 未知， – = 未实现位 （读为 0）， q = 值取决于具体条件。

注 1： 如果 Vdd 过低，将激活上电复位，寄存器将受到不同的影响。

2： INTCON 和 / 或 PIR1 中的一位或多位会受到影响 （引起唤醒）。

3： 当器件被中断唤醒且 GIE 位置 1 时， PC 装入中断向量 （0004h）。

4： 具体条件下的复位值，请参见表 14-5。

5： 如果复位是由于欠压引起的，则 bit 0 = 0。所有其他复位将导致 bit 0 = u。

6： 仅在 SSPM<3:0> = 1001 时可以访问。

表 14-5：　特殊寄存器的初始化状态

条件	程序计数器	状态寄存器	PCON 寄存器
上电复位	000h	0001 1xxx	--01 --0x
正常工作期间的 $\overline{\text{MCLR}}$ 复位	000h	000u uuuu	--0u --uu
休眠期间的 $\overline{\text{MCLR}}$ 复位	000h	0001 0uuu	--0u --uu
WDT 复位	000h	0000 uuuu	--0u --uu
WDT 唤醒	PC + 1	uuu0 0uuu	--uu --uu
欠压复位	000h	0001 1uuu	--01 --u0
被中断从休眠状态唤醒	PC + 1[(1)]	uuu1 0uuu	--uu --uu

图注：　u = 不变， x = 未知， — = 未实现位 （读为 0）。

注　**1：**　当器件被中断唤醒且全局中断允许位 GIE 位置 1 时，执行 PC + 1 后， PC 装入中断向量 （0004h）。

14.3 中断

PIC16F631/677/685/687/689/690 有以下多种中断源：

- 外部中断 RA2/INT
- TMR0 溢出中断
- PORTA/PORTB 电平变化中断
- 两个比较器中断
- A/D 中断 （PIC16F631 除外）
- Timer1 溢出中断
- Timer2 匹配中断 （仅限 PIC16F685/PIC16F690）
- EEPROM 数据写中断
- 故障保护时钟监视器中断
- 增强型 CCP 中断（仅限 PIC16F685/PIC16F690）
- EUSART 接收和发送中断 （仅限 PIC16F687/ PIC16F689/PIC16F690）

中断控制寄存器 （INTCON）和外设中断请求寄存器 1（PIR1）在各自的标志位中记录各种中断请求。INTCON 寄存器还包括各个中断允许位和全局中断允许位。

全局中断允许位 GIE （INTCON<7>）在置 1 时允许所有未屏蔽的中断，而在清零时禁止所有中断。可以通过在 INTCON、PIE1 和 PIE2 寄存器中相应的使能位来禁止各个中断。复位时 GIE 被清零。

执行 “从中断返回” 指令 RETFIE 退出中断服务程序并将 GIE 位置 1，从而重新允许未屏蔽的中断。

INTCON 寄存器包含以下中断标志位：

- INT 引脚中断
- PORTA/PORTB 电平变化中断
- TMR0 溢出中断

外设中断标志位在 PIR1 和 PIR2 寄存器中。相应的中断允许位在 PIE1 和 PIE2 寄存器中。

PIR1 寄存器包含以下中断标志位：

- A/D 中断
- EUSART 接收和发送中断
- Timer1 溢出中断
- 同步串行口 （SSP）中断
- 增强型 CCP1 中断
- Timer1 溢出中断
- Timer2 匹配中断

PIR2 寄存器包含以下中断标志位：

- 故障保护时钟监视器中断
- 两个比较器中断
- EEPROM 数据写中断

当响应一个中断时：

- 将 GIE 位清零以禁止其他中断。
- 将返回地址压入堆栈。
- PC 中装入 0004h。

对于外部中断事件，例如 INT 引脚或 PORTA/PORTB 电平变化中断，中断响应延时将会是 3 到 4 个指令周期。确切的延时时间取决于发生中断事件的时间（见图 14-8）。对于单周期或双周期指令，中断响应延时完全相同。进入中断服务程序之后，就可以通过查询中断标志位来确定中断源。在重新允许中断前，必须用软件将中断标志位清零，以避免重复响应该中断。

> **注 1：** 各中断标志位的置 1 不受相应的中断屏蔽位或 GIE 位状态的影响。
>
> **2：** 当执行一条清零 GIE 位的指令后，任何在一条等待在下一周期执行的中断都将被忽略。当 GIE 位被重新置 1 后，被忽略的中断仍会继续等待处理。

关于 Timer1、Timer2、比较器、A/D、数据 EEPROM、EUSART、SSP 或增强型 CCP 模块的更多信息，请参见相应的外设章节。

14.3.1 RA2/INT 中断

RA2/INT 引脚上的外部中断是边沿触发的；当 INTEDG 位 （OPTION_REG<6>）被置 1 时在上升沿触发，而当 INTEDG 位被清零时在下降沿触发。当 RA2/INT 引脚上出现有效边沿时， INTF 位 （INTCON<1>）置 1。可以通过将 INTE 控制位 （INTCON<4>）清零来禁止该中断。在重新允许该中断前，必须在中断服务程序中先用软件将 INTF 位清零。如果 INTE 位在进入休眠状态前被置 1，则 RA2/INT 中断能将处理器从休眠状态唤醒。GIE 位的状态决定处理器在被唤醒后是否会跳转到中断向量（0004h）处执行代码。关于休眠的详细信息，请参见**第 14.6 节 “掉电模式（休眠）”**；关于 RA2/INT 中断将处理器从休眠状态唤醒的时序，请参见图 14-10。

> **注：** 必须对 ANSEL 和 CM2CON0 寄存器进行初始化，以将模拟通道配置为数字输入。配置为模拟输入的引脚读为 0。

14.3.2 TIMER0 中断

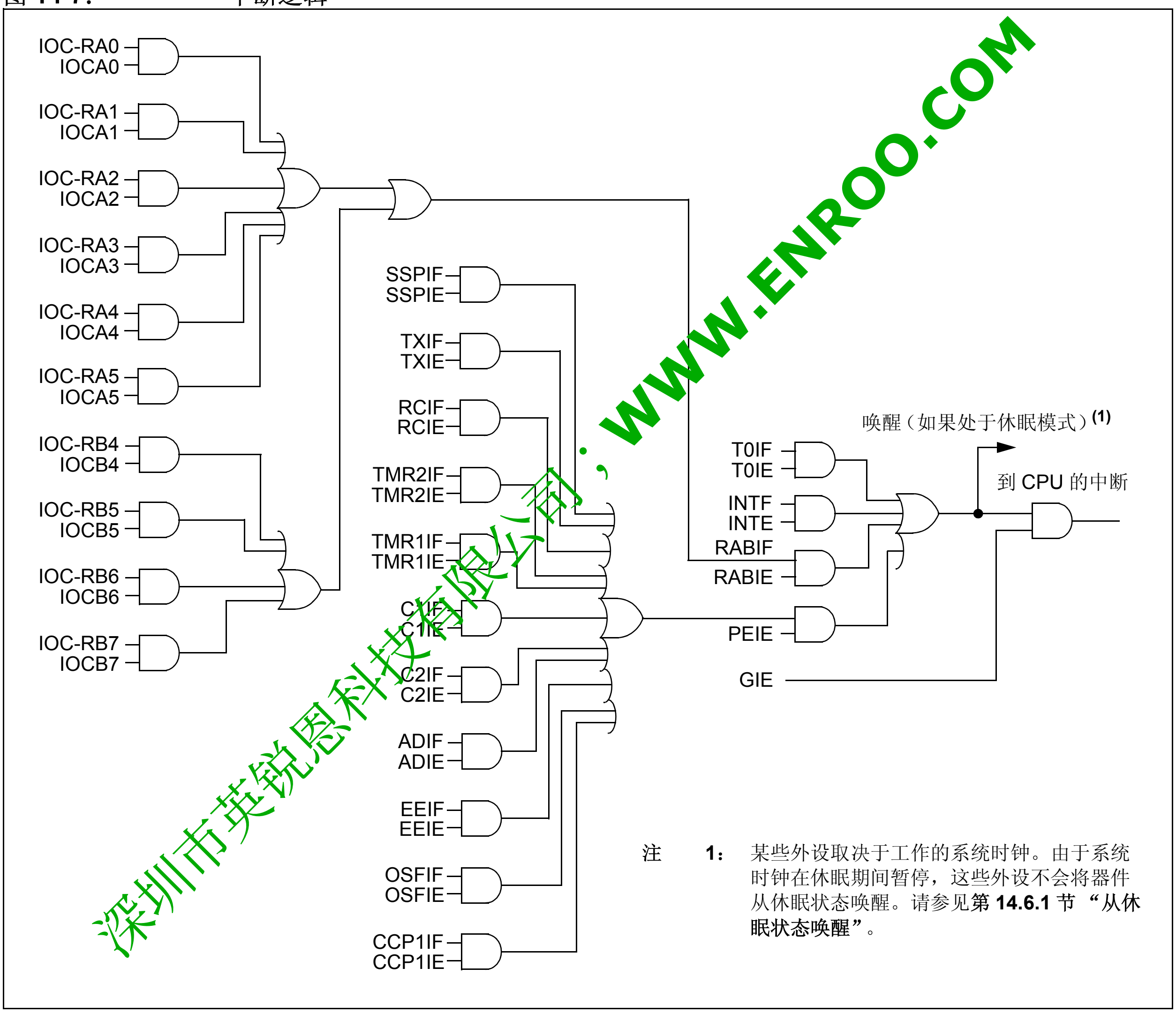
TMR0 寄存器溢出（FFh → 00h）会将 T0IF（INTCON<2>）位置 1。可以通过置 1/ 清零 T0IE（INTCON<5>）位来允许 / 禁止该中断。关于 Timer0 模块的操作，请参见第 **5.0** 节 **“Timer0 模块”**。

14.3.3 PORTA/PORTB 中断

PORTA 或 PORTB 输入电平的变化会使 RABIF（INTCON<0>）位置 1。可以通过置 1/ 清零 RABIE（INTCON<3>）位来使能 / 禁止该中断。此外，可通过 IOCA 或 IOCB 寄存器对该端口的各个引脚进行配置。

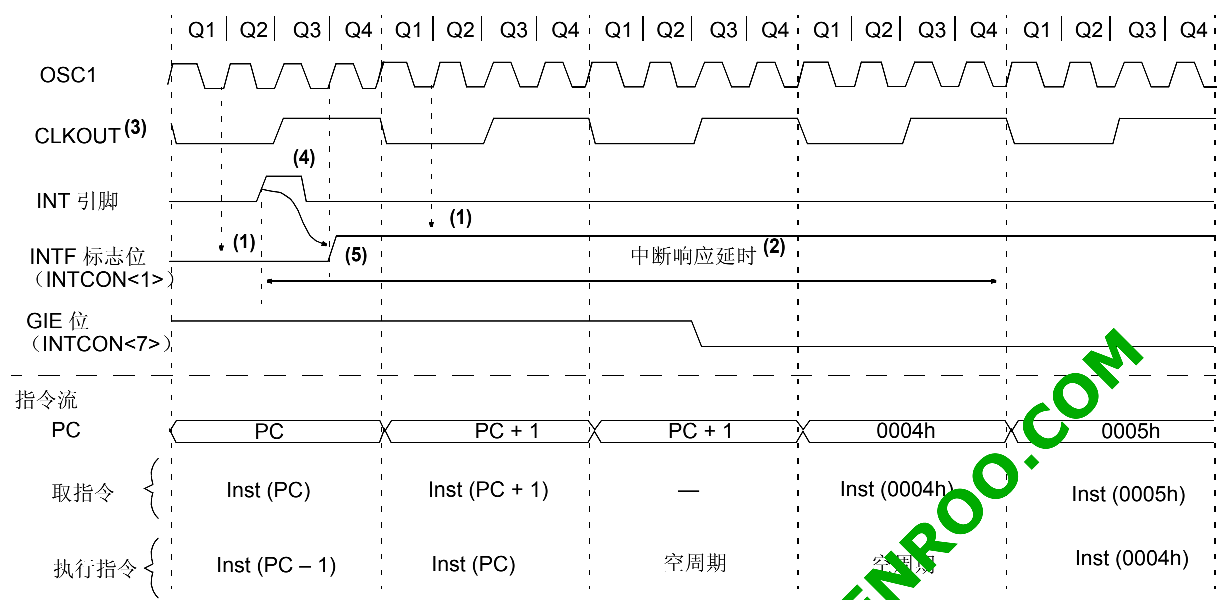
> **注：** 当读操作正在执行时发生了 I/O 引脚电平变化（Q2 周期的起始时刻），则 RABIF 中断标志位可能不会被置 1。更多信息，请参见第 **4.2.3** 节 **“电平变化中断”**。

图 14-7： 中断逻辑



注 1： 某些外设取决于工作的系统时钟。由于系统时钟在休眠期间暂停，这些外设不会将器件从休眠状态唤醒。请参见第 **14.6.1** 节 **“从休眠状态唤醒”**。

图 14-8： INT 引脚中断时序



注 1： 在此时采样 INTF 标志（每个 Q1 周期）。
2： 异步中断响应延时为 3-4 个 TCY。同步中断响应延时为 3 个 TCY，其中 TCY 为一个指令周期。不管 Inst（PC）是单周期还是双周期指令，中断响应延时都是一样的。
3： 只有在 INTOSC 和 RC 振荡器模式下 CLKOUT 才有效。
4： 关于 INT 脉冲的最小宽度，请参见**第 17.0 节 "电气规范"**中的交流规范。
5： 允许在 Q4-Q1 周期内的任何时间将 INTF 置 1。

表 14-6： 中断寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	0000 ----
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000 ----	0000 ----

图注： x = 未知， u = 不变， — = 未实现（读为 0）， q = 值取决于具体条件。
中断模块不使用阴影单元。

14.4 中断的现场保护

在中断期间，仅将返回 PC 值压入堆栈。通常情况下，用户可能希望在中断期间保存关键寄存器（例如，W 寄存器和状态寄存器）。这必须用软件实现。

由于在 PIC16F631/677/685/687/689/690 中，每个存储区的低 16 个字节都是公用的（见图 2-2 和 2-3），临时保存寄存器 W_TEMP 和 STATUS_TEMP 都应该放在这里。这 16 个存储单元不需要分区，因此便于现场保护和恢复。与例 14-1 中相同的代码可被用于：

- 保存 W 寄存器
- 保存状态寄存器
- 执行 ISR 代码
- 恢复状态寄存器 （和存储区选择位寄存器）
- 恢复 W 寄存器

注：	PIC16F631/677/685/687/689/690 通常不需要保存 PCLATH。但是，如果要在 ISR 和主程序中使用计算 GOTO，就必须在 ISR 中保存和恢复 PCLATH。

例 14-1： 在 RAM 中保存状态寄存器和 W 寄存器

```
MOVWF  W_TEMP          ;Copy W to TEMP register
SWAPF  STATUS,W        ;Swap status to be saved into W
CLRF   STATUS          ;bank 0, regardless of current bank, Clears IRP,RP1,RP0
MOVWF  STATUS_TEMP     ;Save status to bank zero STATUS_TEMP register
:
:(ISR)                 ;Insert user code here
:
SWAPF  STATUS_TEMP,W   ;Swap STATUS_TEMP register into W
                       ;(sets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
```

14.5 看门狗定时器 （WDT）

WDT 具有以下特性：

- 工作于 LFINTOSC （31 kHz）
- 包含 16 位预分频器
- 与 Timer0 共用 8 位预分频器
- 超时周期从 1 ms 到 268 秒
- 配置位和使用软件控制

在表 14-7 中说明的特定条件下， WDT 会被清零。

14.5.1 WDT 振荡器

WDT 以 31 kHz LFINTOSC 振荡器作为其工作的时基。OSCCON 寄存器的 LTS 位并不反映 LFINTOSC 是否被使能。

在所有复位时 WDTCON 的值均为 “`---0 1000`”。这样标称时基为 17 ms。

注：	当振荡器起振定时器 （OST）被启动时， WDT 保持复位状态，因为 OST 使用 WDT 异步计数器来执行振荡器延时计数。 OST 计数到期后， WDT 将开始计数 （如果使能）。

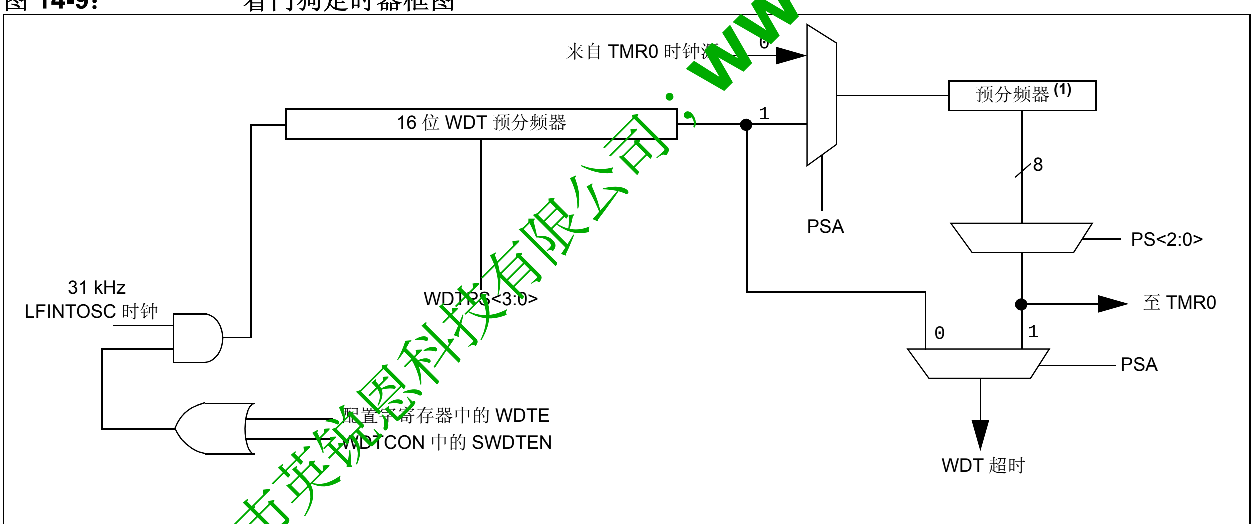
14.5.2 WDT 控制

WDTE 位位于配置字寄存器中。当其置 1 时， WDT 持续运行。

如果配置字寄存器中的 WDTE 位置 1，则 WDTCON 寄存器的 SWDTEN 位不起作用。如果 WDTE 清零，便可以使用 SWDTEN 位使能和禁止 WDT。将该位置 1 将使能 WDT，清零则禁止 WDT。

选项寄存器的 PSA 和 PS<2:0> 位功能与 PIC16F631/677/685/687/689/690 系列单片机的早期版本相同。更多信息，请参见**第 5.0 节 “Timer0 模块”**。

图 14-9： 看门狗定时器框图



注 1： 这是 Timer0 与 WDT 共用的预分频器。更多信息，请参见**第 5.4 节 “预分频器”**。

表 14-7： **WDT 状态**

条件	**WDT**
WDTE = `0`	清零
`CLRWDT` 指令	
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 结束

寄存器 14-2： WDTCON：看门狗定时器控制寄存器

U-0	U-0	U-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0
—	—	—	WDTPS3	WDTPS2	WDTPS1	WDTPS0	SWDTEN[(1)]
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	U = 未实现位，读为 0	
-n = POR 时的值	1 = 置 1	0 = 清零	x = 未知

bit 7-5 未实现：读为 0

bit 4-1 **WDTPS<3:0>**：看门狗定时器周期选择位

位值 = 预分频比
0000 = 1:32
0001 = 1:64
0010 = 1:128
0011 = 1:256
0100 = 1:512 （复位值）
0101 = 1:1024
0110 = 1:2048
0111 = 1:4096
1000 = 1:8192
1001 = 1:16384
1010 = 1:32768
1011 = 1:65536
1100 = 保留
1101 = 保留
1110 = 保留
1111 = 保留

bit 0 **SWDTEN**：软件使能或禁止看门狗定时器位[(1)]
1 = WDT 开启
0 = WDT 关闭 （复位值）

注 1：如果 WDTE 配置位 = 1，则 WDT 始终被使能，而与该控制位的状态无关。如果 WDTE 配置位 = 0，则可以使用该控制位开启 / 关闭 WDT。

表 14-8： 看门狗定时器寄存器汇总

名称	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	POR/BOR 时的值	所有其他复位时的值
CONFIG[(1)]	$\overline{CPD}$	$\overline{CP}$	MCLRE	$\overline{PWRTE}$	WDTE	FOSC2	FOSC1	FOSC0	—	—
OPTION_REG	$\overline{RABPU}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111
WDTCON	—	—	—	WDTPS3	WDTPS2	WSTPS1	WDTPS0	SWDTEN	---0 1000	---0 1000

图注： 看门狗定时器不使用阴影单元。
注 1： 关于配置字寄存器中所有位的操作，请参见寄存器 14-1。

14.6 掉电模式（休眠）

通过执行 SLEEP 指令可进入掉电模式。

如果使能看门狗定时器：

- WDT 将被清零并保持运行。
- 状态寄存器中的 PD 位被清零。
- TO 位被置 1。
- 关闭振荡器驱动器。
- 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）。

为使这种模式下的电流消耗降至最低，所有 I/O 引脚都应保持为 VDD 或 VSS，以确保没有外部电路从 I/O 引脚消耗电流，同时应禁止比较器和 CVREF。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引脚拉为高电平或低电平。为使电流消耗降至最低，T0CKI 输入也应保持为 VDD 或 VSS。还应考虑 PORTA 片上上拉的影响。

MCLR 引脚必须处于逻辑高电平。

> **注：** 应该注意到WDT超时导致的复位并不会将 MCLR 引脚驱动为低电平。

14.6.1 从休眠状态唤醒

可以通过以下任一事件将器件从休眠状态唤醒：

1. MCLR 引脚上的外部复位输入。
2. 看门狗定时器唤醒（如果 WDT 使能）。
3. RA2/INT 引脚中断、 PORTA 电平变化中断或外设中断。

第一种事件会导致器件复位。后两种事件被认为是程序执行的继续。状态寄存器中的 TO 和 PD 位用于确定器件复位的原因。PD 位在上电时被置 1，而在执行 SLEEP 指令时被清零。 TO 位在发生 WDT 唤醒时被清零。

下列外设中断可以将器件从休眠状态唤醒：

1. TMR1 中断。 Timer1 必须用作异步计数器。
2. ECCP 捕捉模式中断。
3. A/D 转换（当 A/D 时钟源为 FRC 时）。
4. EEPROM 写操作完成。
5. 比较器输出状态变化。
6. 电平变化中断。
7. 来自 INT 引脚的外部中断。
8. EUSART 间隔检测， I^2C 从。

由于在休眠期间没有片上时钟处于工作状态，因此其他外设不能产生中断。

当执行 SLEEP 指令时，下一条指令（PC + 1）将预先取出。如果希望通过中断事件唤醒器件，则必须将相应的中断允许位置 1（允许）。发生唤醒与 GIE 位的状态无关。如果 GIE 位被清零（禁止），器件将继续执行 SLEEP 指令之后的指令。如果 GIE 位被置 1（允许），器件执行 SLEEP 指令之后的指令，然后跳转到中断地址（0004h）处执行代码。如果不希望执行 SLEEP 指令之后的指令，用户应该在 SLEEP 指令后面放置一条 NOP 指令。

> **注：** 如果禁止了全局中断（GIE 被清零），但有任一中断源将其中断允许位以及相应的中断标志位置 1，器件将立即从休眠状态唤醒。 SLEEP 指令执行完成。

器件从休眠状态唤醒时， WDT 都将被清零，而与唤醒原因无关。

14.6.2 使用中断唤醒

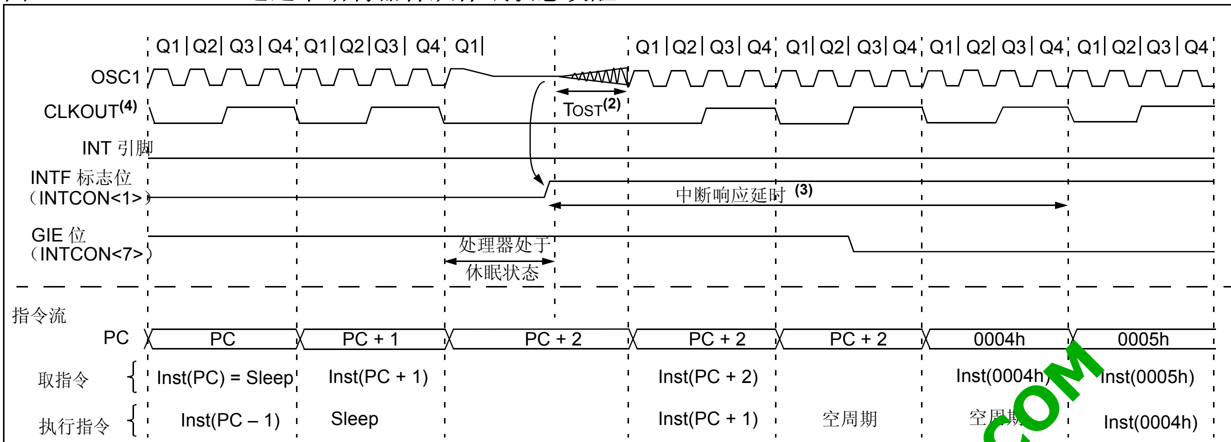
当禁止全局中断（GIE 被清零）时，并且有任一中断源的中断允许位和中断标志位都置 1，将会发生下列事件之一：

- 如果在执行 SLEEP 指令**之前**产生了中断，那么 SLEEP 指令将被作为一条 NOP 指令执行。因此，WDT 及其预分频器和后分频器（如果使能）将不会被清零，并且 TO 位将不会被置 1，同时 PD 位也不会被清零。
- 如果在执行 SLEEP 指令**期间或之后**产生了中断，那么器件将被立即从休眠状态唤醒。 SLEEP 指令将在唤醒之前执行完毕。因此， WDT 及其预分频器和后分频器（如果使能）将被清零，并且 TO 位将被置 1，同时 PD 位也将被清零。

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，可测试 PD 位。如果 PD 位置 1，则说明 SLEEP 指令被当作一条 NOP 指令执行了。

在执行 SLEEP 指令之前，必须先执行一条 CLRWDT 指令，来确保将 WDT 清零。

图 14-10： 通过中断将器件从休眠状态唤醒



注 1： 假设为 XT、HS 或 LP 振荡器模式。

2： TOST = 1024 TOSC （图中未按比例绘制）。该延时不适用于 EC 和 RC 振荡器模式。

3： 假设 GIE = 1。在这种情形下，处理器被唤醒后，将跳转到 0004h 处执行代码。如果 GIE = 0，程序将继续执行。

4： 在 XT、 HS、 LP 或 EC 振荡器模式下，不输出 CLKOUT 信号，在此处仅作为时序参考。

14.7 代码保护

如果代码保护位未编程，可以通过使用ICSP™读片上程序存储器来进行校验。

> **注：** 当关闭代码保护功能时，将擦除整个数据 EEPROM 和闪存程序存储器的内容。更多信息，请参见 *“PIC12F6XX/16F6XX Memory Programming Specification”*（DS41204）。

14.8 ID 存储单元

有 4 个存储器单元 （2000h-2003h）被指定为 ID 存储单元，供用户存储校验和或其他代码标识号。在正常执行过程中不能访问这些单元，但可在编程 / 校验模式下对它们进行读写。只使用了 ID 存储单元的低 7 位。

14.9 在线串行编程

可在最终应用电路中对 PIC16F631/677/685/687/689/690 单片机进行串行编程。编程可以简单地通过一根时钟线、一根数据线和另外三根线 （如下）完成：

- 电源线
- 接地线
- 编程电压线

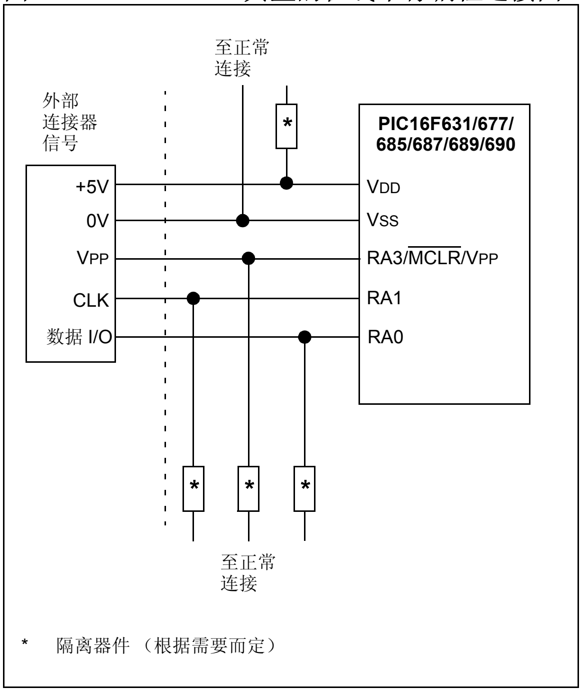
这允许用户在生产电路板时使用未编程器件，仅在产品交付之前才对单片机进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

通过将 RA0/AN0/C1IN+/ICSPDAT/ULPWU 和 RA1/AN1/C12IN-/VREF/ICSPCLK 引脚保持为低电平，并同时将 MCLR （VPP）引脚电平从 VIL 升到 VIHH，可将器件置于编程 / 校验模式。更多信息，请参见 *“PIC12F6XX/16F6XX Memory Programming Specification”*（DS41204）。RA0 变成编程数据引脚，而 RA1 变成编程时钟引脚。在此模式下， RA0 和 RA1 均为施密特触发器输入引脚。

复位后，为将器件置于编程 / 校验模式，程序计数器（PC）指向地址单元 00h。然后向器件发送一条 6 位命令。根据具体命令是执行装载还是读取操作，可向器件提供一个 14 位的程序数据或是从器件取一个 14 位的程序数据。关于串行编程的完整细节，请参见 *“PIC12F6XX/16F6XX Memory Programming Specification”*（DS41204）。

图 14-11 为典型的在线串行编程连接图。

图 14-11： 典型的在线串行编程连接图



15.0 指令集汇总

PIC16F690 指令集具有高度正交性，由以下三种基本类型的指令组成：

- **字节**操作类指令
- **位**操作类指令
- **立即数和控制**操作类指令

每条 PIC16 指令都是一个 14 位字，由**操作码**（指定指令类型）和一个或多个**操作数**（进一步指定指令操作）组成。图 15-1 给出了上述各种类型的指令格式，表 15-1 给出了不同操作码字段的说明。

表 15-2 列出了 MPASM™ 汇编器可识别的指令。

对于**字节**操作类指令，“f”表示文件寄存器标识符，“d”表示目标寄存器标识符。文件寄存器标识符指定了指令将会使用哪一个文件寄存器。

目标标识符指定了操作结果的存放位置。如果“d”为 `0`，操作结果存入 W 寄存器。如果“d”为 `1`，操作结果存入指令指定的文件寄存器中。

对于**位**操作类指令，“b”表示位域标识符，它可选择受到操作影响的位，而“f”表示该位所处的文件寄存器地址。

对于**立即数和控制**操作类指令，“k”表示一个 8 位或 11 位的常数或立即数。

每个指令周期由 4 个振荡器周期组成。因此，如果振荡器频率为 4 MHz，正常的指令执行时间为 1 µs。除非条件测试结果为真或指令执行改变了程序计数器的值，否则执行所有的指令都只需要一个指令周期。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 `NOP` 指令。

所有指令示例均使用格式“`0xhh`”来表示一个十六进制数，其中“h”表示一个十六进制数字。

15.1 读 - 修改 - 写操作

任何一条指定文件寄存器作为指令一部分的指令都进行读 - 修改 - 写（RMW）操作。读寄存器、修改数据并根据指令或目标标识符“d”存储结果。即使指令写入该寄存器，还是会发生对寄存器的读操作。

例如，`CLRF PORTA` 指令将读取 PORTA，清除所有数据位，然后把结果写回到 PORTA。这个示例会得到一个意外结果，即清除将 RAIF 标志位置 1 的条件。

表 15-1： 操作码字段说明

字段	说明
f	文件寄存器地址（0x00 至 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	无关位（= `0` 或 `1`）。 汇编器将产生 x = `0` 的代码。为了与所有的 Microchip 软件工具兼容，建议使用这种形式。
d	目标寄存器选择； d = `0`：结果存入 W， d = `1`：结果存入文件寄存器 f。 默认值 d = `1`。
PC	程序计数器
$\overline{\text{TO}}$	超时位
C	进位位
DC	半进位位
Z	全零位
$\overline{\text{PD}}$	掉电位

图 15-1： 指令的通用格式

针对字节的文件寄存器操作

13 — 8	7	6 — 0
操作码	d	f（寄存器地址）

d = 0，结果存入 W
d = 1，结果存入 f
f = 7 位文件寄存器地址

针对位的文件寄存器操作

13 — 10	9 — 7	6 — 0
操作码	b（位号）	f（寄存器地址）

b = 3 位位地址
f = 7 位文件寄存器地址

立即数和控制操作

一般格式

13 — 8	7 — 0
操作码	k（立即数）

k = 8 位立即数

仅限 `CALL` 和 `GOTO` 指令

13 — 11	10 — 0
操作码	k（立即数）

k = 11 位立即数

表 15-2： PIC16F684 指令集

助记符，操作数		说明	周期	14 位操作码 MSb　LSb	受影响的状态位	注
针对字节的文件寄存器操作						
ADDWF	f, d	W 与 f 相加	1	00 0111 dfff ffff	C, DC, Z	1, 2
ANDWF	f, d	W 和 f 作逻辑与运算	1	00 0101 dfff ffff	Z	1, 2
CLRF	f	将 f 清零	1	00 0001 lfff ffff	Z	2
CLRW	–	将 W 清零	1	00 0001 0xxx xxxx	Z	
COMF	f, d	对 f 取反	1	00 1001 dfff ffff	Z	1, 2
DECF	f, d	f 递减 1	1	00 0011 dfff ffff	Z	1, 2
DECFSZ	f, d	f 递减 1，为 0 则跳过	1(2)	00 1011 dfff ffff		1, 2, 3
INCF	f, d	f 递增 1	1	00 1010 dfff ffff	Z	1, 2
INCFSZ	f, d	f 递增 1，为 0 则跳过	1(2)	00 1111 dfff ffff		1, 2, 3
IORWF	f, d	W 和 f 作逻辑或运算	1	00 0100 dfff ffff	Z	1, 2
MOVF	f, d	传送 f	1	00 1000 dfff ffff	Z	1, 2
MOVWF	f	将 W 内容传送到 f	1	00 0000 lfff ffff		
NOP	–	空操作	1	00 0000 0xx0 0000		
RLF	f, d	f 带进位循环左移	1	00 1101 dfff ffff	C	1, 2
RRF	f, d	f 带进位循环右移	1	00 1100 dfff ffff	C	1, 2
SUBWF	f, d	f 减去 W	1	00 0010 dfff ffff	C, DC, Z	1, 2
SWAPF	f, d	将 f 中的两个半字节进行交换	1	00 1110 dfff ffff		1, 2
XORWF	f, d	W 和 f 作逻辑异或运算	1	00 0110 dfff ffff	Z	1, 2
针对位的文件寄存器操作						
BCF	f, b	将 f 寄存器中的某位清零	1	01 00bb bfff ffff		1, 2
BSF	f, b	将 f 寄存器中的某位置 1	1	01 01bb bfff ffff		1, 2
BTFSC	f, b	测试 f 中的某位，为 0 则跳过	1(2)	01 10bb bfff ffff		3
BTFSS	f, b	测试 f 中的某位，为 1 则跳过	1(2)	01 11bb bfff ffff		3
立即数和控制操作						
ADDLW	k	立即数与 W 相加	1	11 111x kkkk kkkk	C, DC, Z	
ANDLW	k	立即数和 W 作逻辑与运算	1	11 1001 kkkk kkkk	Z	
CALL	k	调用子程序	2	10 0kkk kkkk kkkk		
CLRWDT	–	将看门狗定时器清零	1	00 0000 0110 0100	$\overline{TO}$, $\overline{PD}$	
GOTO	k	转移到地址	2	10 1kkk kkkk kkkk		
IORLW	k	立即数和 W 作逻辑或运算	1	11 1000 kkkk kkkk	Z	
MOVLW	k	将立即数传送到 W	1	11 00xx kkkk kkkk		
RETFIE	–	从中断返回	2	00 0000 0000 1001		
RETLW	k	返回并将立即数送入 W	2	11 01xx kkkk kkkk		
RETURN	–	从子程序返回	2	00 0000 0000 1000		
SLEEP	–	进入待机模式	1	00 0000 0110 0011	$\overline{TO}$, $\overline{PD}$	
SUBLW	k	立即数减去 W	1	11 110x kkkk kkkk	C, DC, Z	
XORLW	k	立即数和 W 作逻辑异或运算	1	11 1010 kkkk kkkk	Z	

注 1：当 I/O 寄存器修改自身时（例如，`MOVF GPIO, 1`），所使用的值是引脚上的当前值。例如，如果将一引脚配置为输入，且对应数据锁存器中的值为 `1`，但此时若有外部器件将该引脚驱动为低电平，则被写回数据锁存器的数据值将是 `0`。

2：当对 TMR0 寄存器执行该指令（并且 d = 1）时，如果已为 Timer0 模块分配了预分频器，则将该预分频器清零。

3：如果程序计数器（PC）被修改或条件测试结果为真，则该指令需要两个周期。第二个周期执行一条 `NOP` 指令。

15.2 指令说明

ADDLW 立即数与 W 相加

语法：	[*标号*] ADDLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) + k → (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，结果存储在 W 寄存器中。

ADDWF W 与 f 相加

语法：	[*标号*] ADDWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) + (f) → （目标寄存器）
受影响的状态位：	C、 DC 和 Z
说明：	将 W 寄存器的内容与寄存器 f 的内容相加。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

ANDLW 立即数和 W 作逻辑与运算

语法：	[*标号*] ANDLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) .AND.(k) → (W)
受影响的状态位：	Z
说明：	将 W 寄存器中的内容与 8 位立即数 k 进行逻辑与运算。结果存入 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 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .AND.(f) → （目标寄存器）
受影响的状态位：	Z
说明：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑与运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

BCF 将 f 寄存器中的某位清零

语法：	[*标号*] BCF f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	0 → (f<b>)
受影响的状态位：	无
说明：	将寄存器 f 中的位 b 清零。

BSF 将 f 寄存器中的某位置 1

语法：	[*标号*] BSF f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	1 → (f<b>)
受影响的状态位：	无
说明：	将寄存器 f 的位 b 置 1。

BTFSC 测试 f 中的某位，为 0 则跳过

语法：	[*标号*] BTFSC f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b ≤ 7
操作：	如果 (f<b>) = 0，则跳过
受影响的状态位：	无
说明：	如果寄存器 f 的位 b 为 1，则执行下一条指令。 如果寄存器 f 的位 b 为 0，则丢弃下一条指令转而执行一条 NOP 指令，使之成为一条双周期指令。

BTFSS　测试 f 中的某位，为 1 则跳过

语法：	[*标号*] BTFSS f,b
操作数：	0 ≤ f ≤ 127 0 ≤ b < 7
操作：	如果 (f<b>) = 1，则跳过
受影响的状态位：	无
说明：	如果寄存器 f 的位 b 为 0，则执行下一条指令。 如果位 b 为 1，则丢弃下一条指令转而执行一条 NOP 指令，使之成为一条双周期指令。

CALL　调用子程序

语法：	[*标号*] CALL k
操作数：	0 ≤ k ≤ 2047
操作：	(PC)+ 1→ TOS， k → PC<10:0>， (PCLATH<4:3>) → PC<12:11>
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。 11 位直接地址值被装入 PC 的 <10:0> 位。PC 的高位值从 PCLATH 装入。CALL 是一条双周期指令。

CLRF　将 f 清零

语法：	[*标号*] CLRF f
操作数：	0 ≤ f ≤ 127
操作：	00h → (f) 1 → Z
受影响的状态位：	Z
说明：	寄存器 f 的内容被清零，并且 Z 位被置 1。

CLRW　将 W 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。全零位（Z）被置 1。

CLRWDT　将看门狗定时器清零

语法：	[*标号*] CLRWDT
操作数：	无
操作：	00h → WDT 0 → WDT 预分频器， 1 → $\overline{TO}$ 1 → $\overline{PD}$
受影响的状态位：	$\overline{TO}$ 和 $\overline{PD}$
说明：	CLRWDT 指令复位看门狗定时器及其预分频器。 状态位 $\overline{TO}$ 和 $\overline{PD}$ 均被置 1。

COMF　对 f 取反

语法：	[*标号*] COMF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	$(\bar{f})$ → （目标寄存器）
受影响的状态位：	Z
说明：	将寄存器 f 的内容取反。如果 d 为 0，结果存入 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 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - 1 → （目标寄存器）
受影响的状态位：	Z
说明：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

DECFSZ	f 递减 1，为 0 则跳过
语法：	[*标号*] DECFSZ f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - 1 → （目标寄存器）； 如果结果 = 0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递减 1。如果 d 为 0，结果存入 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，代之执行一条 NOP 指令，使之成为一条双周期指令。

GOTO	无条件跳转
语法：	[*标号*] GOTO k
操作数：	0 ≤ k ≤ 2047
操作：	k → PC<10:0> PCLATH<4:3> → PC<12:11>
受影响的状态位：	无
说明：	GOTO 是一条无条件转移指令。 11 位立即数被装入 PC 的 <10:0> 位。 PC 的高位从 PCLATH<4:3> 装入。 GOTO 是一条双周期指令。

INCF	f 递增 1
语法：	[*标号*] INCF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) + 1 → （目标寄存器）
受影响的状态位：	Z
说明：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

INCFSZ	f 递增 1，为 0 则跳过
语法：	[*标号*] INCFSZ f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) + 1 → （目标寄存器）， 如果结果 = 0 则跳过
受影响的状态位：	无
说明：	将寄存器 f 的内容递增 1。如果 d 为 0，结果存入 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，代之执行一条 NOP 指令，使之成为一条双周期指令。

IORLW	立即数和 W 作逻辑或运算
语法：	[*标号*] IORLW k
操作数：	0 ≤ k ≤ 255
操作：	(W) .OR. 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 进行逻辑或运算。结果存入 W 寄存器。

IORWF	W 和 f 作逻辑或运算
语法：	[*标号*] IORWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .OR.(f) → （目标寄存器）
受影响的状态位：	Z
说明：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑或运算。如果 d 为 0，结果存入 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：	$0 \le f \le 127$ $d \in [0,1]$
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，目标寄存器为文件寄存器 f 本身。由于状态标志位 Z 要受影响，可用 d = 1 对文件寄存器进行检测。
指令字数：	1
指令周期数：	1
示例：	`MOVF FSR, 0` 执行指令后 W = FSR寄存器的值 Z = 1

MOVLW　将立即数传送到 W

语法：	[*标号*]　MOVLW　k
操作数：	$0 \le k \le 255$
操作：	k → (W)
受影响的状态位：	无
说明：	将 8 位立即数 k 装入 W 寄存器。其余无关位均汇编为 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：	$0 \le f \le 127$
操作：	(W) → (f)
受影响的状态位：	无
说明：	将 W 寄存器的数据传送到寄存器 f。
指令字数：	1
指令周期数：	1
示例：	`MOVW OPTION F` 执行指令前 OPTION = 0xFF W = 0x4F 执行指令后 OPTION = 0x4F W = 0x4F

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：	不执行任何操作。
指令字数：	1
指令周期数：	1
示例：	`NOP`

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（Top-of-Stack， 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 返回并将立即数送入 W

语法:	[*标号*] RETLW k
操作数:	0 ≤ k ≤ 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
TABLE •           ;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　　f 带进位循环左移

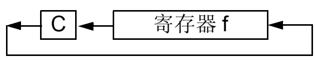
语法：　[*标号*]　RLF　f,d

操作数：　$0 \le f \le 127$
$d \in [0,1]$

操作：　参见如下说明

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，结果存入 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

<u>示例：</u>
```
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
```

SLEEP　　进入休眠模式

语法：　[*标号*]　SLEEP

操作数：　无

操作：　00h → WDT，
0 → WDT 预分频器，
1 → $\overline{TO}$，
0 → $\overline{PD}$

受影响的状态位：　$\overline{TO}$ 和 $\overline{PD}$

说明：　掉电状态位 $\overline{PD}$ 被清零。超时状态位 $\overline{TO}$ 被置 1。看门狗定时器及其预分频器被清零。振荡器停振，处理器进入休眠模式。

RRF　　f 带进位循环右移

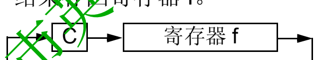
语法：　[*标号*]　RRF　f,d

操作数：　$0 \le f \le 127$
$d \in [0,1]$

操作：　参见如下说明

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，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。



SUBLW　　立即数减去 W

语法：　[*标号*]　SUBLW　k

操作数：　$0 \le k \le 255$

操作：　k - (W) → (W)

受影响的状态位：　C、 DC 和 Z

说明：　用 8 位立即数 k 减去 W 寄存器的内容（通过二进制补码方式进行运算）。结果存入 W 寄存器。

C = 0	W > k
C = 1	W ≤ k
DC = 0	W<3:0> > k<3:0>
DC = 1	W<3:0> ≤ k<3:0>

SUBWF	f 减去 W
语法：	[*标号*] SUBWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(f) - (W) → （目标寄存器）
受影响的状态位：	C、 DC 和 Z
说明：	用寄存器 f 的内容减去 W 寄存器的内容 （通过二进制补码方式进行运算）。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

C = 0	W > f
C = 1	W ≤ f
DC = 0	W<3:0> > f<3:0>
DC = 1	W<3:0> ≤ f<3:0>

SWAPF	将 f 的高半字节和低半字节交换
语法：	[*标号*] SWAPF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [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：	寄存器 f 的高半字节和低半字节相互交换。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

XORLW	立即数和 W 作逻辑异或运算
语法：	[*标号*] XORLW k
操作数：	0 ≤ k ≤ 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 进行逻辑异或运算。结果存入 W 寄存器。

XORWF	W 和 f 作逻辑异或运算
语法：	[*标号*] XORWF f,d
操作数：	0 ≤ f ≤ 127 d ∈ [0,1]
操作：	(W) .XOR.(f) → （目标寄存器）
受影响的状态位：	Z
说明：	将 W 寄存器的内容与寄存器 f 的内容进行逻辑异或运算。如果 d 为 0，结果存入 W 寄存器。如果 d 为 1，结果存回寄存器 f。

16.0 开发支持

一系列硬件及软件开发工具对 PIC® 单片机提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 汇编器 / 编译器 / 链接器
  - MPASM™ 汇编器
  - MPLAB C18 和 MPLAB C30 C 编译器
  - MPLINK™ 目标链接器 / MPLIB™ 目标库管理器
  - MPLAB ASM30 汇编器 / 链接器 / 库
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB ICE 2000 在线仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 2
- 器件编程器
  - PICSTART® Plus 开发编程器
  - MPLAB PM3 器件编程器
  - PICkit™ 2 开发编程器
- 低成本演示和开发板及评估工具包

16.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16 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）
  - 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
- 可视化器件初始化程序，便于进行寄存器的初始化
- 鼠标停留在变量上进行查看的功能
- 通过拖放把变量从源代码窗口拉到观察窗口
- 丰富的在线帮助
- 集成了可选的第三方工具，如 HI-TECH 软件 C 编译器和 IAR C 编译器

MPLAB IDE 可以让您：

- 编辑源文件 （汇编语言或 C 语言）
- 单击一次即可完成汇编 （或编译）并将代码下载到 PIC MCU 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 （汇编语言或 C 语言）
  - 混合汇编语言和 C 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更强大的工具时的学习时间。

16.2 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于所有的 PIC 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 汇编器具有如下特征：

- 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

16.3 MPLAB C18 和 MPLAB C30 C 编译器

MPLAB C18 和 MPLAB C30 代码开发系统是完全的 ANSI C 编译器，分别适用于 Microchip 的 PIC18 和 PIC24 系列单片机及 dsPIC30F 和 dsPIC33 系列数字信号控制器。这些编译器可提供其他编译器并不具备的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，编译器提供了针对 MPLAB IDE 调试器的优化符号信息。

16.4 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中。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征：

- 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来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

16.5 MPLAB ASM30 汇编器、链接器和库管理器

MPLAB ASM30 汇编器为 dsPIC30F 器件提供转换自符号汇编语言的可重定位机器码。 MPLAB C30 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征：

- 支持整个 dsPIC30F 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

16.6 MPLAB SIM 软件模拟器

MPLAB SIM软件模拟器在指令级对PIC MCU和dsPIC® DSC 进行模拟，使得用户可以在 PC 主机的环境下进行代码开发。对于任何给定的指令，用户均可对数据区进行检查或修改，并通过各种触发机制来产生激励。可以将各寄存器的情况记录在文件中，以便进行进一步地运行时分析。跟踪缓冲器和逻辑分析器的显示使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的状况。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C18 和 MPLAB C30 C 编译器以及MPASM和MAPLAB ASM30 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。

16.7 MPLAB ICE 2000 高性能在线仿真器

MPLAB ICE 2000 在线仿真器旨在为产品开发工程师提供一整套用于 PIC 单片机的设计工具。MPLAB ICE 2000 在线仿真器的软件控制由 MPLAB 集成开发环境平台提供，它允许在单一环境下进行编辑、编译、下载以及源代码调试。

MPLAB ICE 2000 是全功能仿真器系统，它具有增强的跟踪、触发和数据监控功能。处理器模块可插拔，使系统可轻松进行重新配置以适应各种不同处理器的仿真需要。MPLAB ICE 2000 在线仿真器的架构允许对其进行扩展以支持新的 PIC 单片机。

MPLAB ICE 2000 在线仿真器系统设计为一款实时仿真系统，该仿真系统具备通常只有昂贵的开发工具中才有的高级功能。选择 PC 平台和 Microsoft® Windows® 32 位操作系统可使这些功能在一个简单而统一的应用中得到很好的利用。

16.8 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® 闪存 MCU 和 dsPIC® DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

MPLAB REAL ICE 探针通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与常用 MPLAB ICD 2 系统兼容的连接器（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 MPLAB REAL ICE 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，如软件断点和汇编代码跟踪等。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、高速仿真、实时变量监视、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。

16.9 MPLAB ICD 2 在线调试器

Microchip 的在线调试器 MPLAB ICD 2 是一款功能强大而成本低廉的运行时开发工具，通过 RS-232 或高速 USB 接口与 PC 主机相连。该工具基于闪存 PIC MCU，可用于开发本系列及其他 PIC MCU 和 dsPIC DSC。MPLAB ICD 2 使用了闪存器件中内建的在线调试功能。该功能结合 Microchip 的在线串行编程（In-Circuit Serial Programming™，ICSP™）协议，可在 MPLAB 集成开发环境的图形用户界面上提供成本效益很高的在线闪存调试。这使设计人员可通过设置断点、单步运行以及对变量、CPU 状态以及外设寄存器进行监视的方法实现源代码的开发和调试。其全速运行特性可对硬件和应用进行实时测试。MPLAB ICD 2 还可用作某些 PIC 器件的开发编程器。

16.10 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款通用的、符合 CE 规范的器件编程器，其可编程电压设置在 VDDMIN 和 VDDMAX 之间时可靠性最高。它有一个用来显示菜单和错误信息的大 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 器件编程器无需与 PC 相连即可对 PIC 器件进行读取、验证和编程。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，可对存储器很大的器件进行快速编程，它还采用 SD/MMC 卡用作文件存储及数据安全应用。

16.11 PICSTART Plus 开发编程器

PICSTART Plus开发编程器是一款易于使用而成本低廉的原型编程器。它通过 COM （RS-232）端口与 PC 相连。 MPLAB 集成开发环境软件使得该编程器的使用简便、高效。PICSTART Plus 开发编程器支持采用 DIP 封装的大部分 PIC 器件，其引脚数最多可达 40 个。引脚数更多的器件，如 PIC16C92X 和 PIC17C76X，可通过连接一个转接插槽来获得支持。PICSTART Plus 开发编程器符合 CE 规范。

16.12 PICkit 2 开发编程器

PICkit™ 2 开发编程器是一个低成本编程器；对于某些选定闪存器件，它也是一个调试器，通过其易于使用的接口可对众多 Microchip 的低档、中档和 PIC18F 系列闪存单片机进行编程。 PICkit 2 入门工具包中包含一个有实验布线区的开发板、十二堂系列课程、软件和 HI-TECH 的 PICC™ Lite C 编译器，有助于用户快速掌握 PIC® 单片机的使用。这一工具包为使用 Microchip 功能强大的中档闪存系列单片机进行编程、评估和应用开发，提供了所需的一切。

16.13 演示、开发和评估板

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，用于测试和修改。

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。

有关演示、开发和评估工具包的完整列表，请查阅 Microchip 公司网页 （www.microchip.com）。

17.0 电气规范

绝对最大额定值 (†)

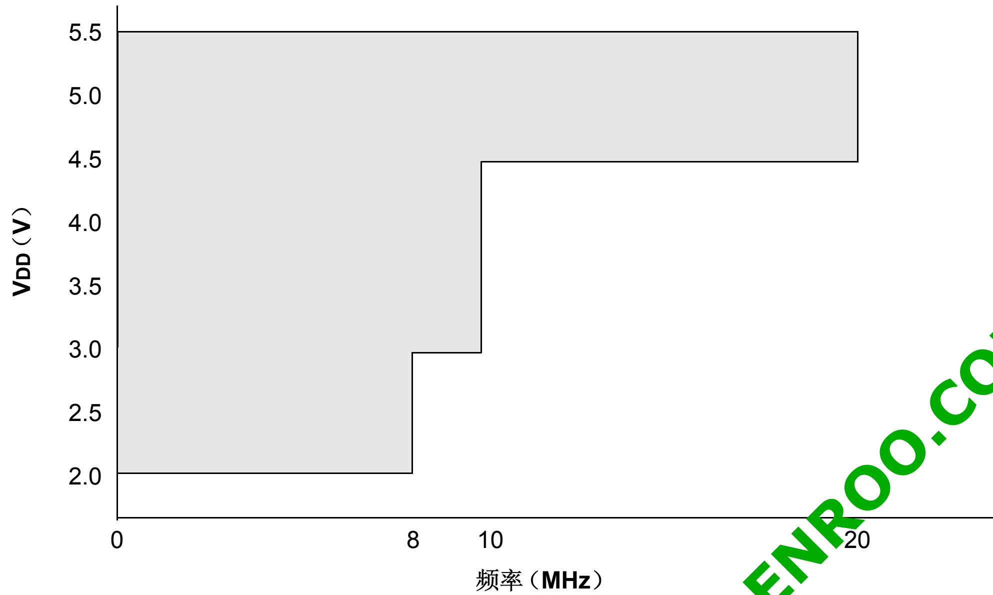
环境温度	-40°C 至 +125°C
储存温度	-65°C 至 +150°C
V_{DD} 引脚相对于 V_{SS} 的电压	-0.3V 至 +6.5V
$\overline{MCLR}$ 引脚相对于 V_{SS} 的电压	-0.3V 至 +13.5V
所有其他引脚相对于 V_{SS} 的电压	-0.3V 至 (V_{DD} + 0.3V)
总功耗 [(1)]	800 mW
流出 V_{SS} 引脚的最大电流	300 mA
流入 V_{DD} 引脚的最大电流	250 mA
输入钳位电流， I_{IK} （$V_I < 0$ 或 $V_I > V_{DD}$）	±20 mA
输出钳位电流， I_{OK} （$V_O < 0$ 或 $V_O > V_{DD}$）	±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
PORTA、 PORTB 和 PORTC （联合）最大灌电流	200 mA
PORTA、 PORTB 和 PORTC （联合）最大拉电流	200 mA

注 1： 功耗按如下公式计算：$P_{DIS} = V_{DD} \times \{I_{DD} - \sum I_{OH}\} + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum(V_{OL} \times I_{OL})$。

> † 注：如果器件工作条件超过上述 “绝对最大额定值”，可能引起器件永久性损坏。这仅是极限参数，我们不建议器件工作在极限值甚至超过上述极限值。器件长时间工作在额定最大值条件下，其稳定性可能受到影响。

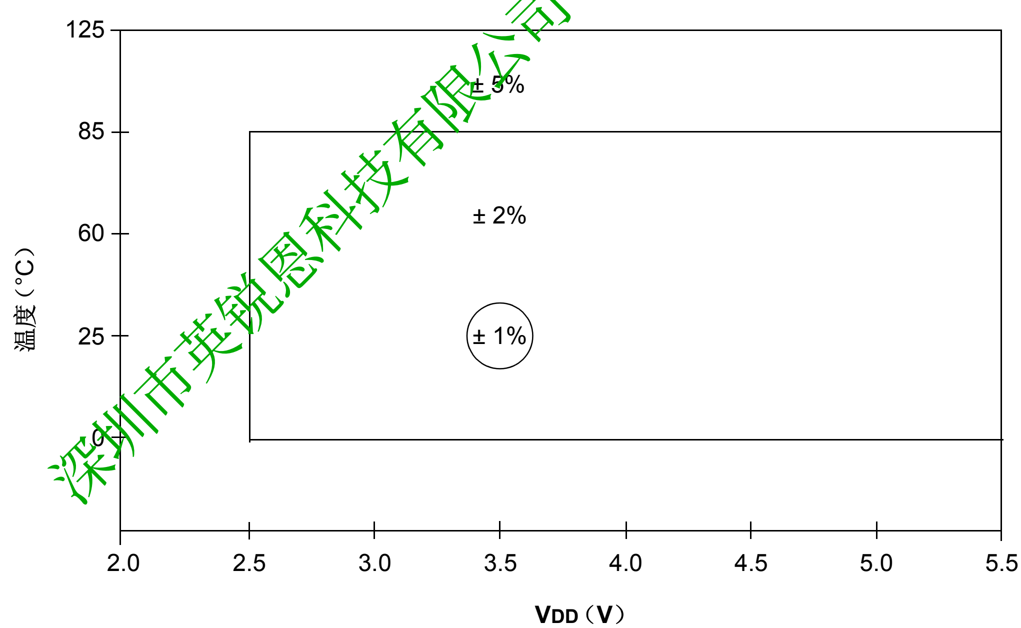
> 注： $\overline{MCLR}$ 引脚上若出现低于 V_{SS} 的尖峰电压，感应电流超过 80 mA，可能导致锁死。因此，在 $\overline{MCLR}$ 引脚上施加低电平时，应使用一个 50-100 Ω 的串联电阻，而不是将该引脚直接与 V_{SS} 连接。

图 17-1： PIC16F631/677/685/687/689/690 电压—频率关系图， -40°C ≤ TA ≤ +125°C



注 1：阴影区域表示允许的电压频率组合。

图 17-2： 器件 VDD 和温度范围内 HFINTOSC 的频率精度



17.1 直流特性： PIC16F631/677/685/687/689/690-I （工业级） PIC16F631/677/685/687/689/690-E （扩展级）

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
	VDD	供电电压	2.0	—	5.5	V	FOSC < = 8 MHz：HFINTOSC 和 EC
D001			2.0	—	5.5	V	FOSC < = 4 MHz
D001C			3.0	—	5.5	V	FOSC < = 10 MHz
D001D			4.5	—	5.5	V	FOSC < = 20 MHz
D002*	VDR	**RAM 数据保持电压 (1)**	1.5	—	—	V	器件处于休眠模式
D003	VPOR	**VDD 启动电压** （确保内部上电复位信号）	—	VSS	—	V	详情请参见第 **14.2.1** 节 “上电复位（**POR**）”。
D004*	SVDD	**VDD 上升率** （确保内部上电复位信号）	0.05	—	—	V/ms	详情请参见第 **14.2.1** 节 “上电复位（**POR**）”。

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 这是在不丢失 RAM 数据的前提下，休眠模式下 VDD 所能降到的最小电压值。

17.2 直流特性：PIC16F631/677/685/687/689/690-I （工业级） PIC16F631/677/685/687/689/690-E （扩展级）

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	器件特性	最小值	典型值†	最大值	单位	条件	
						VDD	注
D010	供电电流 （IDD） (1, 2)	—	13	19	μA	2.0	FOSC = 32 kHz LP 振荡器模式
		—	22	30	μA	3.0	
		—	33	60	μA	5.0	
D011*		—	140	240	μA	2.0	FOSC = 1 MHz XT 振荡器模式
		—	220	380	μA	3.0	
		—	380	550	μA	5.0	
D012		—	260	360	μA	2.0	FOSC = 4 MHz XT 振荡器模式
		—	420	650	μA	3.0	
		—	0.8	1.1	mA	5.0	
D013*		—	130	220	μA	2.0	FOSC = 1 MHz EC 振荡器模式
		—	215	360	μA	3.0	
		—	360	520	μA	5.0	
D014		—	220	340	μA	2.0	FOSC = 4 MHz EC 振荡器模式
		—	375	550	μA	3.0	
		—	0.65	1.0	mA	5.0	
D015		—	8	20	μA	2.0	FOSC = 31 kHz LFINTOSC 模式
		—	16	40	μA	3.0	
		—	31	65	μA	5.0	
D016*		—	340	450	μA	2.0	FOSC = 4 MHz HFINTOSC 模式
		—	500	700	μA	3.0	
		—	0.8	1.2	mA	5.0	
D017		—	410	650	μA	2.0	FOSC = 8 MHz HFINTOSC 模式
		—	700	950	μA	3.0	
		—	1.30	1.65	mA	5.0	
D018		—	230	400	μA	2.0	FOSC = 4 MHz EXTRC 模式 (3)
		—	400	680	μA	3.0	
		—	0.63	1.1	mA	5.0	
D019		—	3.8	5.0	mA	4.5	FOSC = 20 MHz HS 振荡器模式
		—	4.0	5.45	mA	5.0	

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 有效工作模式下，所有 IDD 测量值的测试条件为：OSC1 = 外部方波，轨到轨满幅；所有 I/O 引脚均为三态，上拉至 VDD ； $\overline{MCLR}$ = VDD ； WDT 禁止。

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 IR = VDD/2REXT （mA）来估算，其中 REXT 的单位是 kΩ。

4： 外设电流为基本 IDD 或 IPD 与该外设使能时所额外消耗的电流之和。可通过从该极限值中减去基本 IDD 或 IPD 电流，以确定外设 Δ 电流。在计算总电流消耗时应使用最大值。

5： 在休眠模式下，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、所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

17.2 直流特性：PIC16F631/677/685/687/689/690-I （工业级） PIC16F631/677/685/687/689/690-E （扩展级） （续）

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	器件特性	最小值	典型值 †	最大值	单位	条件 VDD	条件 注
D020	掉电基本电流 （IPD） [2]	—	0.05	1.2	µA	2.0	WDT、BOR、比较器、VREF 和 T1OSC 均被禁止
		—	0.15	1.5	µA	3.0	
		—	0.35	1.8	µA	5.0	
		—	90	500	nA	3.0	-40°C ≤ TA ≤ +25°C
D021		—	1.0	2.2	µA	2.0	WDT 电流 [1]
		—	2.0	4.0	µA	3.0	
		—	3.0	7.0	µA	5.0	
D022		—	42	60	µA	3.0	BOR 电流 [1]
		—	85	122	µA	5.0	
D023		—	32	45	µA	2.0	比较器电流 [1]，两个比较器均被使能
		—	60	78	µA	3.0	
		—	120	160	µA	5.0	
D024		—	30	36	µA	2.0	CVREF 电流 [1] （高电压范围）
		—	45	55	µA	3.0	
		—	75	95	µA	5.0	
D024a*		—	39	47	µA	2.0	CVREF 电流 [1] （低电压范围）
		—	59	72	µA	3.0	
		—	98	124	µA	5.0	
D025		—	2.0	5.0	µA	2.0	T1OSC 电流， 32.768 kHz
		—	2.5	5.5	µA	3.0	
		—	3.0	7.0	µA	5.0	
D026		—	0.30	1.6	µA	3.0	A/D 电流 [1]，无转换进行
		—	0.36	1.9	µA	5.0	
D027		—	90	125	µA	3.0	VP6 电流
		—	125	162	µA	5.0	

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 有效工作模式下，所有 IDD 测量值的测试条件为：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； WDT 禁止。

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 IR = VDD/2REXT （mA）来估算，其中 REXT 的单位是 kΩ。

4： 外设电流为基本 IDD 或 IPD 与该外设使能时所额外消耗的电流之和。可通过从该极限值中减去基本 IDD 或 IPD 电流，以确定外设 Δ 电流。在计算总电流消耗时应使用最大值。

5： 在休眠模式下，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、所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

17.3 直流特性：PIC16F631/677/685/687/689/690-E （扩展级）

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	**器件特性**	**最小值**	**典型值 †**	**最大值**	**单位**	**条件 VDD**	**条件 注**
D020E	掉电基本电流 （IPD） (2)	—	0.05	9	μA	2.0	WDT、 BOR、比较器、 VREF 和 T1OSC 均被禁止
		—	0.15	11	μA	3.0	
		—	0.35	15	μA	5.0	
		—	90	500	nA	3.0	-40C ≤ TA ≤ +25°C
D021E		—	1.0	17.5	μA	2.0	WDT 电流 (1)
		—	2.0	19	μA	3.0	
		—	3.0	22	μA	5.0	
D022E		—	42	65	μA	3.0	BOR 电流 (1)
		—	85	127	μA	5.0	
D023E		—	32	45	μA	2.0	比较器电流 (1)，两个比较器均被使能
		—	60	78	μA	3.0	
		—	120	160	μA	5.0	
D024E		—	30	70	μA	2.0	CVREF 电流 (1) （高电压范围）
		—	45	90	μA	3.0	
		—	75	120	μA	5.0	
D024AE*		—	39	91	μA	2.0	CVREF 电流 (1) （低电压范围）
		—	59	117	μA	3.0	
		—	98	156	μA	5.0	
D025E		—	2.0	18	μA	2.0	T1OSC 电流
		—	2.5	21	μA	3.0	
		—	3.0	24	μA	5.0	
D026E		—	0.30	12	μA	3.0	A/D 电流 (1)，无转换进行
		—	0.36	16	μA	5.0	
D027E		—	90	130	μA	3.0	VP6 电流
		—	125	170	μA	5.0	

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1：有效工作模式下，所有 IDD 测量值的测试条件为：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 ； WDT 禁止。

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 IR = VDD/2REXT （mA）来估算，其中 REXT 的单位是 kΩ。

4：外设电流为基本 IDD 或 IPD 与该外设使能时所额外消耗的电流之和。可通过从该极限值中减去基本 IDD 或 IPD 电流，以确定外设 Δ 电流。在计算总电流消耗时应使用最大值。

5：在休眠模式下，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、所有 I/O 引脚处于高阻态并且连接到 VDD 时测得的。

17.4 直流特性： PIC16F631/677/685/687/689/690-I （工业级） PIC16F631/677/685/687/689/690-E （扩展级）

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
	VIL	输入低电压					
		I/O 端口：					
D030		带 TTL 缓冲器	Vss	—	0.8	V	4.5V ≤ VDD ≤ 5.5V
D030A			Vss	—	0.15 VDD	V	2.0V ≤ VDD ≤ 4.5V
D031		带施密特触发缓冲器	Vss	—	0.2 VDD	V	2.0V ≤ VDD ≤ 5.5V
D032		MCLR 和 OSC1 （RC 模式） (1)	Vss	—	0.2 VDD	V	
D033		OSC1 （XT 和 LP 模式）	Vss	—	0.3	V	
D033A		OSC1 （HS 模式）	Vss	—	0.3 VDD	V	
	VIH	输入高电压					
		I/O 端口：					
D040		带 TTL 缓冲器	2.0	—	VDD	V	4.5V ≤ VDD ≤ 5.5V
D040A			0.25 VDD + 0.8	—	VDD	V	2.0V ≤ VDD ≤ 4.5V
D041		带施密特触发缓冲器	0.8 VDD	—	VDD	V	2.0V ≤ VDD ≤ 5.5V
D042		MCLR	0.8 VDD	—	VDD	V	
D043		OSC1 （XT 和 LP 模式）	1.6	—	VDD	V	
D043A		OSC1 （HS 模式）	0.7 VDD	—	VDD	V	
D043B		OSC1 （RC 模式）	0.9 VDD	—	VDD	V	（注 1）
	IIL	输入泄漏电流 (2)					
D060		I/O 端口	—	±0.1	±1	μA	VSS ≤ VPIN ≤ VDD，引脚处于高阻态
D061		MCLR(3)	—	±0.1	±5	μA	VSS ≤ VPIN ≤ VDD
D063		OSC1	—	±0.1	±5	μA	VSS ≤ VPIN ≤ VDD，XT、 HS 和 LP 振荡器配置
D070*	IPUR	PORTA 弱上拉电流	50	250	400	μA	VDD = 5.0V， VPIN = VSS
	VOL	输出低电压 (5)					
D080		I/O 端口	—	—	0.6	V	IOL = 8.5 mA， VDD = 4.5V （工业级）
	VOH	输出高电压 (5)					
D090		I/O 端口	VDD – 0.7	—	—	V	IOH = -3.0 mA， VDD = 4.5V （工业级）

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，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。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测得更高的泄漏电流。

4： 更多信息，请参见**第 10.2.1 节 “使用数据 EEPROM”**。

5： 在 CLKOUT 模式下包括 OSC2。

17.4 直流特性： PIC16F631/677/685/687/689/690-I （工业级） PIC16F631/677/685/687/689/690-E （扩展级） （续）

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
D100	IULP	**超低功耗唤醒电流**	—	200	—	nA	请参见应用笔记 AN879，“*Using the Microchip Ultra Low-Power Wake-up Module*”（DS00879）
		输出引脚上的容性负载规范					
D101*	COSC2	OSC2 引脚	—	—	15	pF	当外部时钟用于驱动 OSC1 时处于 XT、 HS 和 LP 模式下
D101A *	CIO	所有 I/O 引脚	—	—	50	pF	
		数据 EEPROM 存储器					
D120	ED	字节耐擦写能力	100K	1M	—	E/W	-40°C ≤ TA ≤ +85°C
D120A	ED	字节耐擦写能力	10K	100K	—	E/W	+85°C ≤ TA ≤ +125°C
D121	VDRW	读 / 写操作时的 VDD	VMIN	—	5.5	V	用 EECON1 来读 / 写 VMIN = 最小工作电压
D122	TDEW	擦 / 写周期时间	—	5	6	ms	
D123	TRETD	特性保持时间	40	—	—	年	假设没有违反其他规范
D124	TREF	刷新前的总擦除 / 写周期数 (4)	1M	10M	—	E/W	-40°C ≤ TA ≤ +85°C
		闪存程序存储器					
D130	EP	单元耐擦写能力	10K	100K	—	E/W	-40°C ≤ TA ≤ +85°C
D130A	ED	单元耐擦写能力	1K	10K	—	E/W	+85°C ≤ TA ≤ +125°C
D131	VPR	读操作时的 VDD	VMIN	—	5.5	V	VMIN = 最小工作电压
D132	VPEW	擦除 / 写操作时的 VDD	4.5	—	5.5	V	
D133	TPEW	擦除 / 写周期时间	—	2	2.5	ms	
D134	TRETD	特性保持时间	40	—	—	年	假设没有违反其他规范

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值”栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，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。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测得更高的泄漏电流。
4： 更多信息，请参见**第 10.2.1 节 “使用数据 EEPROM”**。
5： 在 CLKOUT 模式下包括 OSC2。

17.5 热阻注意事项

标准工作条件（除非另外声明） 工作温度 -40°C ≤ T_A ≤ +125°C					
参数编号	符号	特性	典型值	单位	条件
TH01	θ_{JA}	热阻（结到环境）	62.4	C/W	20 引脚 PDIP 封装
			85.2	C/W	20 引脚 SOIC 封装
			108.1	C/W	20 引脚 SSOP 封装
			40	C/W	20 引脚 QFN 4x4 mm 封装
TH02	θ_{JC}	热阻（结到管壳）	28.1	C/W	20 引脚 PDIP 封装
			24.2	C/W	20 引脚 SOIC 封装
			32.2	C/W	20 引脚 SSOP 封装
			2.5	C/W	20 引脚 QFN 4x4 mm 封装
TH03	T_{DIE}	管芯温度	150	C	用于计算降额功耗
TH04	PD	功耗	—	W	$P_D = P_{INTERNAL} + P_{I/O}$
TH05	$P_{INTERNAL}$	内部功耗	—	W	$P_{INTERNAL} = I_{DD} \times V_{DD}$（注 1）
TH06	$P_{I/O}$	I/O 功耗	—	W	$P_{I/O} = \Sigma (I_{OL} * V_{OL}) + \Sigma (I_{OH} * (V_{DD} - V_{OH}))$
TH07	P_{DER}	降额功耗	—	W	$P_{DER} = P_{DMAX} (T_{DIE} - T_A)/\theta_{JA}$（注 **2** 和 **3**）

注 **1**：I_{DD} 为输出引脚上不连接任何负载时芯片本身运行的电流。
2：T_A = 环境温度。
3：最大允许功耗是绝对最大总功耗和降额功耗两者中较低的值。

17.6 时序参数符号体系

时序参数符号采用以下格式之一进行创建：

1. TppS2ppS
2. TppS

T			
F	频率	T	时间

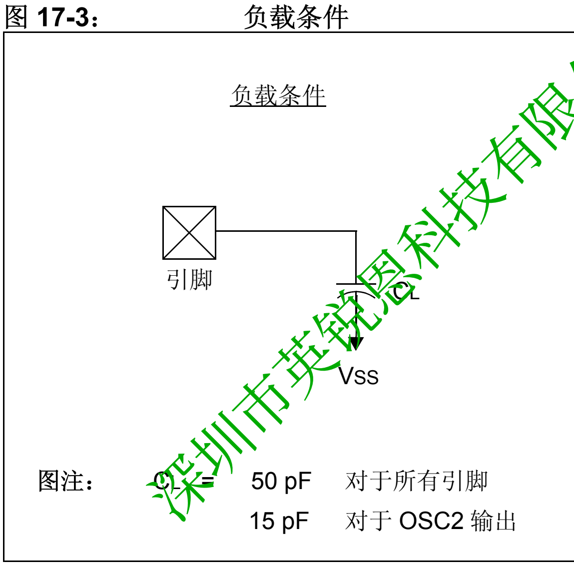
小写字母 （pp）及其含义：

pp			
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：

S			
F	下降	P	周期
H	高	R	上升
I	无效 （高阻）	V	有效
L	低	Z	高阻

图 17-3： 负载条件



17.7 交流特性：PIC16F631/677/685/687/689/690 （工业级，扩展级）

图 17-4： 时钟时序

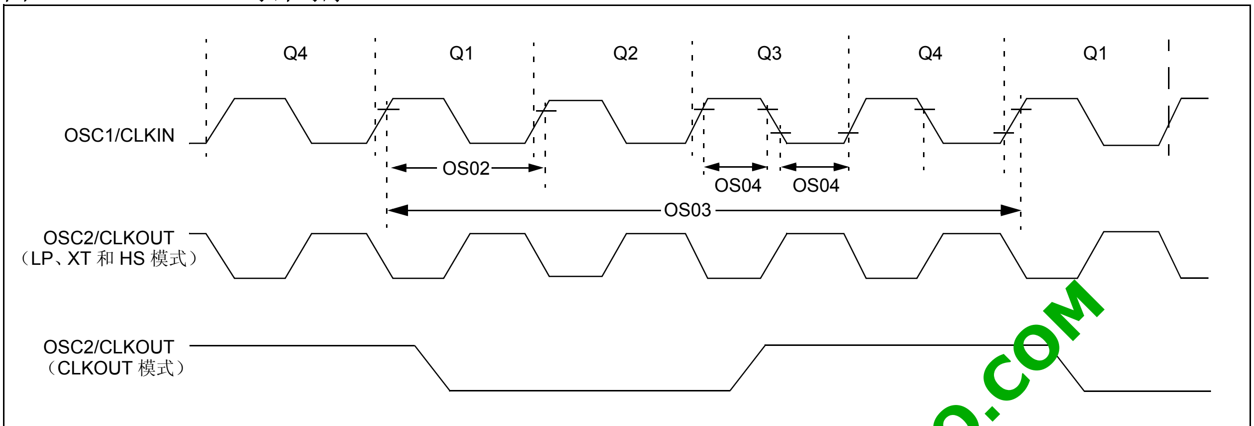


表 17-1： 时钟振荡器时序要求

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°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	—	37	kHz	LP 振荡器模式
			DC	—	4	MHz	XT 振荡器模式
			DC	—	20	MHz	HS 振荡器模式
			DC	—	20	MHz	EC 振荡器模式
		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	—	20	MHz	HS 振荡器模式
			DC	—	4	MHz	RC 振荡器模式
OS02	TOSC	外部 CLKIN 周期 (1)	27	—	∞	μs	LP 振荡器模式
			250	—	∞	ns	XT 振荡器模式
			50	—	∞	ns	HS 振荡器模式
			50	—	∞	ns	EC 振荡器模式
		振荡器周期 (1)		30.5	—	μs	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	指令周期时间 (1)	200	TCY	DC	ns	TCY = 4/FOSC
OS04*	TOSH，TOSL	外部 CLKIN 高电平时间，外部 CLKIN 低电平时间	2	—	—	μs	LP 振荡器
			100	—	—	ns	XT 振荡器
			20	—	—	ns	HS 振荡器
OS05*	TOSR，TOSF	外部 CLKIN 上升时间，外部 CLKIN 下降时间	0	—	∞	ns	LP 振荡器
			0	—	∞	ns	XT 振荡器
			0	—	∞	ns	HS 振荡器

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 指令周期时间 （TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大” 周期时间限制为 “DC”（无时钟）。

表 17-2： 振荡器参数

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°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
OS06	T_{WARM}	运行时内部振荡器开关时间 (3)	—	—	—	2	T_{OSC}	最慢的时钟
OS07	T_{SC}	故障保护采样时钟周期 (1)	—	—	21	—	ms	LFINTOSC/64
OS08	HF_{OSC}	内部已校准的 HFINTOSC 频率 (2)	±1%	7.92	8.0	8.08	MHz	V_{DD} = 3.5V， 25°C
			±2%	7.84	8.0	8.16	MHz	2.5V ≤ V_{DD} ≤ 5.5V， 0°C ≤ T_A ≤ +85°C
			±5%	7.60	8.0	8.40	MHz	2.0V ≤ V_{DD} ≤ 5.5V， -40°C ≤ T_A ≤ +85°C（工业级）， -40°C ≤ T_A ≤ +125°C（扩展级）
OS09*	LF_{OSC}	内部未校准的 LFINTOSC 频率	—	15	31	45	kHz	
OS10*	$T_{IOSC\ ST}$	HFINTOSC 振荡器从休眠模式唤醒后的起振时间	—	5.5	12	24	μs	V_{DD} = 2.0V， -40°C 至 +85°C
			—	3.5	7	14	μs	V_{DD} = 3.0V， -40°C 至 +85°C
			—	3	6	11	μs	V_{DD} = 5.0V， -40°C 至 +85°C

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 指令周期时间 （T_{CY}）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大” 周期时间限制为 “DC”（无时钟）。

2： 为了确保振荡器频率容差，必须尽可能靠近器件，在 V_{DD} 和 V_{SS} 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

3： 根据设计。

图 17-5： CLKOUT 和 I/O 时序

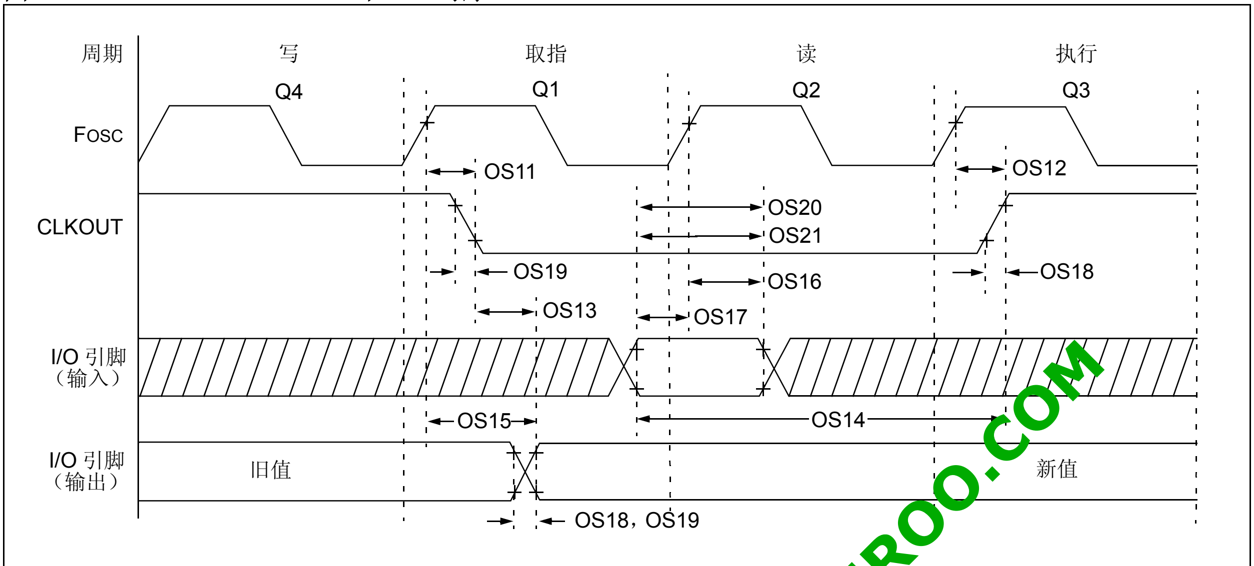


表 17-3： CLKOUT 和 I/O 时序参数

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°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↑ 到 CLKOUT↓ 的时间 (1)	—	—	70	ns	VDD = 5.0V
OS12	TosH2ckH	Fosc↑ 到 CLKOUT↑ 的时间 (1)	—	—	72	ns	VDD = 5.0V
OS13	TckL2ioV	CLKOUT↓ 到端口输出有效的时间 (1)	—	—	20	ns	
OS14	TioV2ckH	在 CLKOUT↑ 之前端口输入有效的时间 (1)	Tosc + 200 ns	—	—	ns	
OS15	TosH2ioV	Fosc↑ （Q1 周期）到端口输出有效的时间	—	50	70*	ns	VDD = 5.0V
OS16	TosH2ioI	Fosc↑ （Q2 周期）到端口输入无效的时间（I/O 输入保持时间）	50	—	—	ns	VDD = 5.0V
OS17	TioV2osH	端口输入有效到 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 (2)	— —	15 40	72 32	ns	VDD = 2.0V VDD = 5.0V
OS19	TioF	端口输出下降时间 (2)	— —	28 15	55 30	ns	VDD = 2.0V VDD = 5.0V
OS20*	Tinp	INT 引脚输入高电平或低电平时间	25	—	—	ns	
OS21*	Trap	PORTA 电平变化中断新输入电平时间	Tcy	—	—	ns	

* 这些参数为特性值，未经测试。
† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 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 x Tosc。
2： 在 CLKOUT 模式下包括 OSC2。

图 17-6： 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

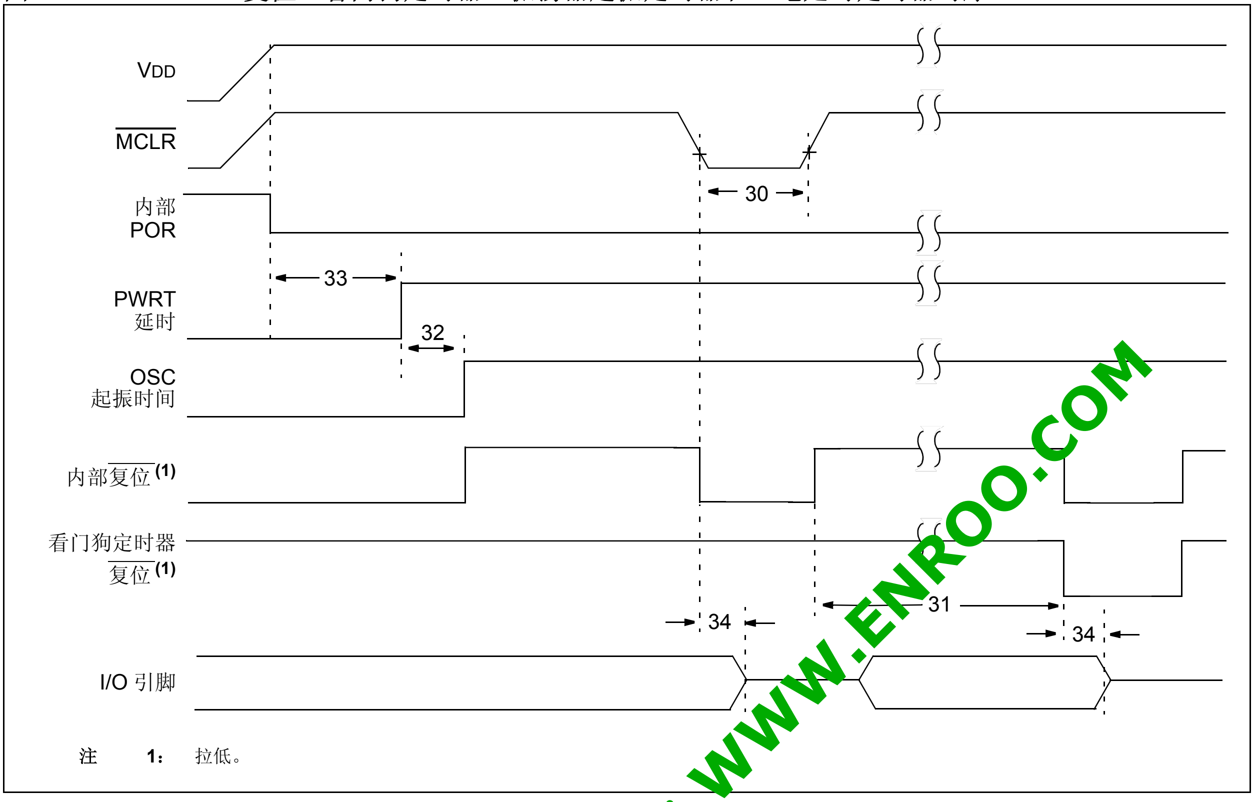


图 17-7： 欠压复位时序和特性

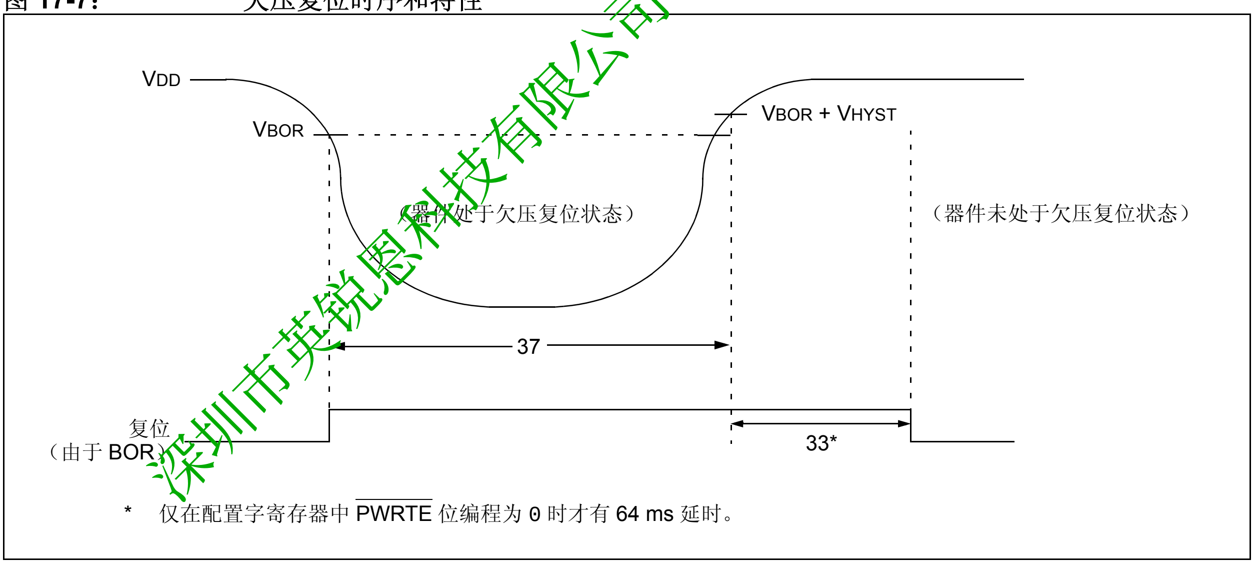


表 **17-4**： 复位、看门狗定时器、振荡器起振定时器、上电延时定时器和欠压复位参数

标准工作条件（除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	条件
30	TMCL	$\overline{\text{MCLR}}$ 脉冲宽度（低电平）	2 5	— —	— —	μs μs	VDD = 5V，-40°C 至 +85°C VDD = 5V
31	TWDT	看门狗定时器超时周期（无预分频器）	10 10	17 17	25 30	ms ms	VDD = 5V，-40°C 至 +85°C VDD = 5V
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	40	65	140	ms	
34*	TIOZ	自 $\overline{\text{MCLR}}$ 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	μs	
35	VBOR	欠压复位电压	2.0	—	2.2	V	（注 **4**）
36*	VHYST	欠压复位滞后	—	50	—	mV	
37*	TBOR	欠压复位最小检测周期	100	—	—	μs	VDD ≤ VBOR

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 **1**： 指令周期时间（TCY）等于输入振荡器时基周期的四倍。所有规定值均为基于针对特定振荡器类型，器件在标准工作条件下执行代码时的特性数据。超出这些规定的极限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 引脚连接了外部时钟。当使用了外部时钟输入时，所有器件的 “最大” 周期时间限制为 “DC”（无时钟）。

2： 根据设计。

3： 较慢时钟的周期。

4： 为了确保这些电压容差，必须尽可能靠近器件，在 VDD 和 VSS 之间接去耦电容。建议并联 0.1 μF 和 0.01 μF 的电容。

图 17-8: TIMER0 和 TIMER1 外部时钟时序

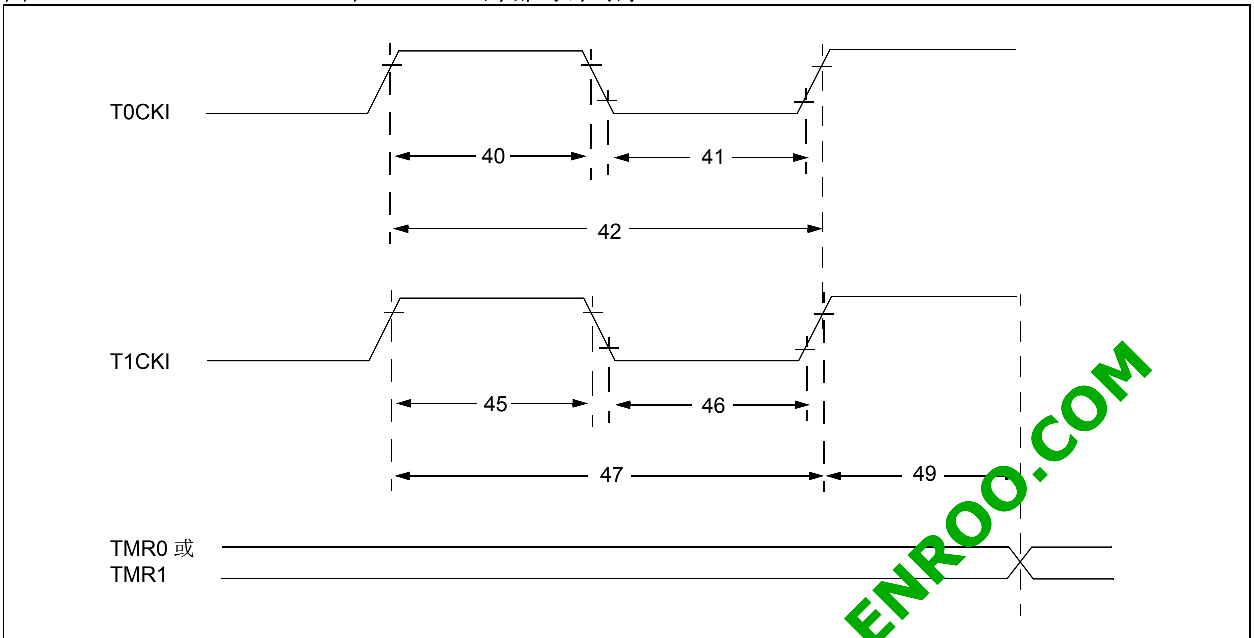


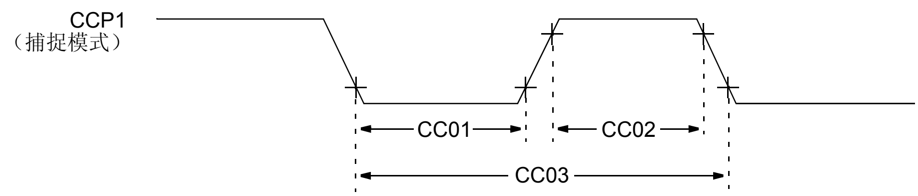
表 17-5: TIMER0 和 TIMER1 外部时钟要求

标准工作条件（除非另外声明）
工作温度 -40°C ≤ T_A ≤ +125°C

参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
40*	T_{T0H}	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*	T_{T0L}	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*	T_{T0P}	T0CKI 周期		取如下二者中较大值：20 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值（2, 4, ..., 256）
45*	T_{T1H}	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*	T_{T1L}	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*	T_{T1P}	T1CKI 输入周期	同步	取如下二者中较大值：30 或 $\frac{T_{CY} + 40}{N}$	—	—	ns	N = 预分频值（1, 2, 4, 8）
			异步	60	—	—	ns	
48	F_{T1}	Timer1 振荡器输入频率范围（通过将 T1OSCEN 位置 1，使能振荡器）		—	32.768	—	kHz	
49*	$T_{CKEZTMR1}$	从外部时钟边沿到定时器递增的延时		2 T_{OSC}	—	7 T_{OSC}	—	定时器处于同步模式

* 这些参数为特性值，未经测试。
† 除非另外声明，否则“典型值”栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

图 17-9： 捕捉 / 比较 /PWM 时序 （ECCP）



注： 负载条件请参见图 17-3。

表 17-6： 捕捉 / 比较 /PWM 要求 （ECCP）

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°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	0.5TCY + 20	—	—	ns	
			有预分频器	20	—	—	ns	
CC02*	TccH	CCP1 输入高电平时间	无预分频器	0.5TCY + 20	—	—	ns	
			有预分频器	20	—	—	ns	
CC03*	TccP	CCP1 输入周期		$\frac{3T_{CY} + 40}{N}$	—	—	ns	N = 预分频值（1、 4 或 16）

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

表 17-7： 比较器规范

比较器规范				标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C				
参数编号	符号	特性		最小值	典型值	最大值	单位	备注
CM01	VOS	输入失调电压		—	±5.0	±10	mV	
CM02	VCM	输入共模电压		0	—	VDD - 1.5	V	
CM03*	CMRR	共模抑制比		+55	—	—	db	
CM04*	TRT	响应时间	下降	—	150	600	ns	（注 1）
			上升	—	200	1000	ns	
CM05*	TMC2COV	比较器模式改变到输出有效的时间		—	—	10	µs	

* 这些参数为特性值，未经测试。

注 1： 响应时间是在比较器一个输入端从 (VDD - 1.5)/2 -100 mV 变化到 (VDD - 1.5)/2 + 20 mV 的过程中测量的。

表 17-8： 比较器参考电压 （CVREF）规范

标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
参数编号	符号	特性	最小值	典型值 †	最大值	单位	备注
CV01*	CLSB	步长 (2)	— —	VDD/24 VDD/32	— —	V V	低电压范围 （VRR = 1） 高电压范围 （VRR = 0）
CV02*	CACC	绝对精度	— —	— —	±1/2 ±1/2	LSb LSb	低电压范围 （VRR = 1） 高电压范围 （VRR = 0）
CV03*	CR	单位电阻值 （R）	—	2k	—	Ω	
CV04*	CST	稳定时间 (1)	—	—	10	µs	

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： 稳定时间是在 VRR = 1 并且 VR<3:0> 从 0000 跳变到 1111 时测得的。

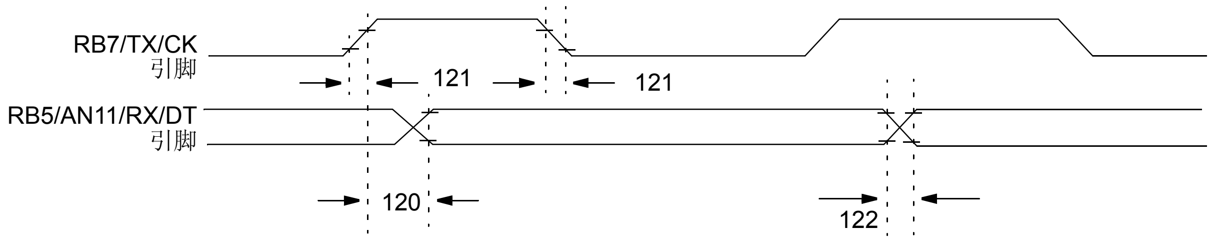
2： 更多信息，请参见第 8.10 节 “比较器参考电压”。

表 17-9： 参考电压 （VR）规范

VR 参考电压规范			标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
VR01	VROUT	VR 电压输出	0.5	0.6	0.7	V	
VR02*	TSTABLE	稳定时间	—	10	100*	µs	

* 这些参数为特性值，未经测试。

图 **17-10：** **EUSART** 同步发送 （主 / 从）时序

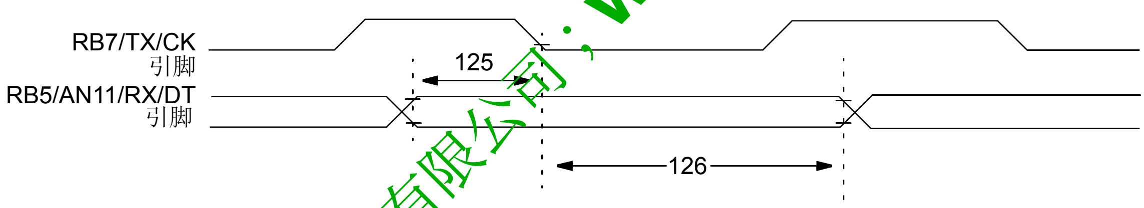


注： 负载条件请参见图 17-3。

表 **17-10：** **EUSART** 同步发送要求

标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C						
参数编号	符号	特性	最小值	最大值	单位	条件
120	TCKH2DTV	同步发送 （主 / 从） 时钟高电平到数据输出有效的时间	—	40	ns	
121	TCKRF	时钟输出上升时间和下降时间 （主模式）		20	ns	
122	TDTRF	数据输出上升时间和下降时间	—	20	ns	

图 **17-11：** **EUSART** 同步接收 （主 / 从）时序

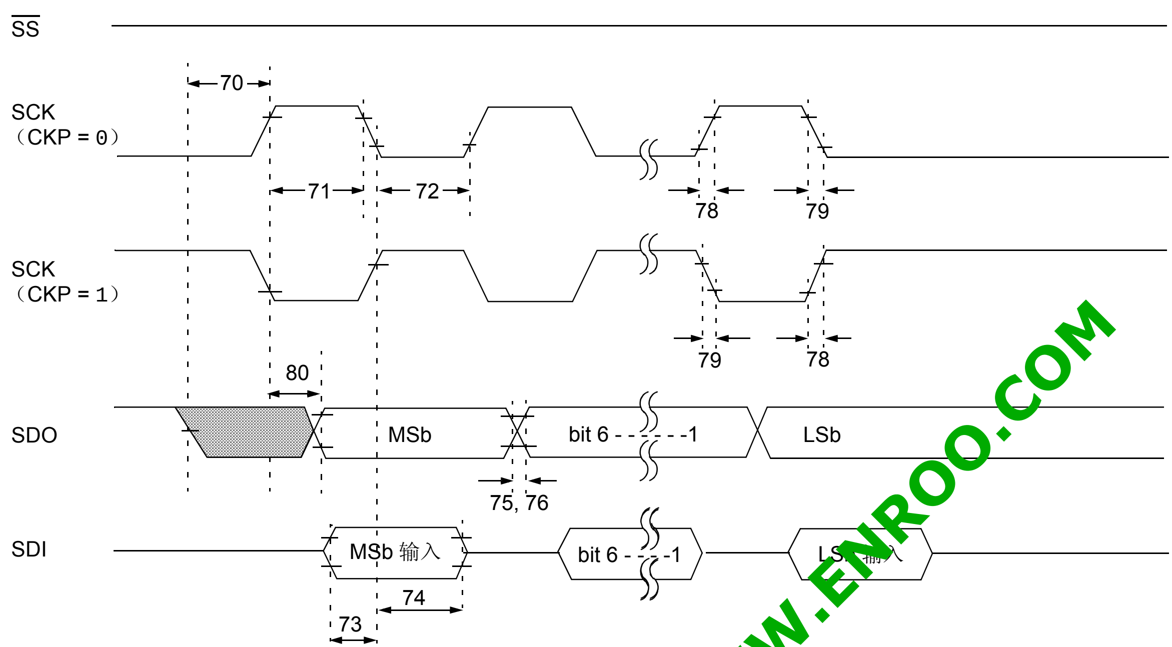


注：负载条件请参见图 17-3。

表 **17-11：** **EUSART** 同步接收要求

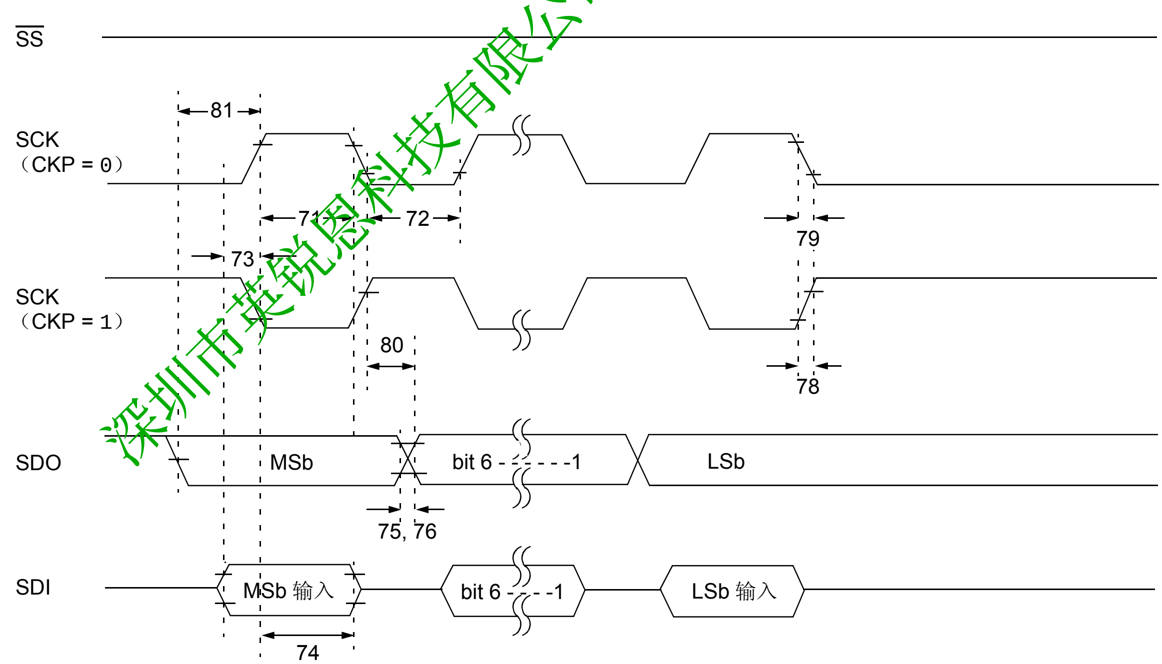
标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C						
参数编号	符号	特性	最小值	最大值	单位	条件
125	TDTV2CKL	同步接收 （主 / 从） CK ↓ 前的数据保持时间 （DT 保持时间）	10	—	ns	
126	TCKL2DTL	CK ↓ 后的数据保持时间 （DT 保持时间）	15	—	ns	

图 17-12： SPI 主模式时序 （CKE = 0， SMP = 0）



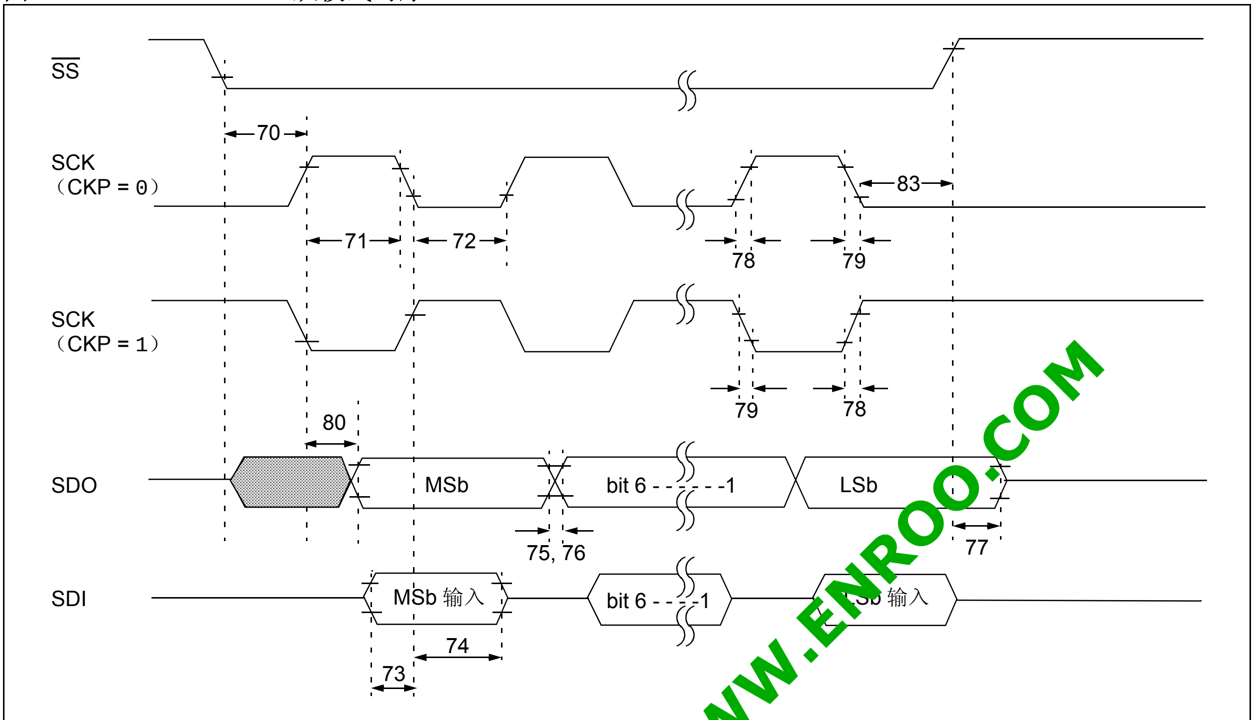
注：负载条件请参见图 17-3。

图 17-13： SPI 主模式时序 （CKE = 1， SMP = 1）



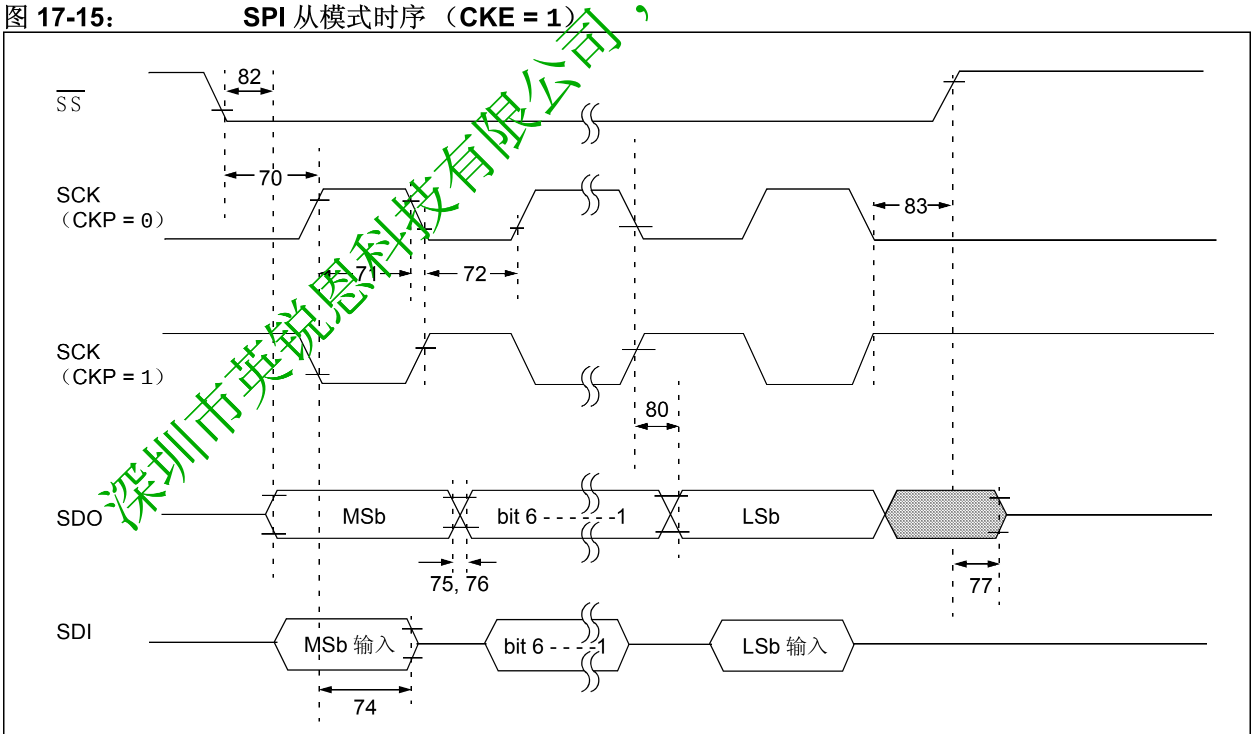
注：负载条件请参见图 17-3。

图 17-14： SPI 从模式时序 （CKE = 0）



注：负载条件请参见图 17-3。

图 17-15： SPI 从模式时序 （CKE = 1）



注：负载条件请参见图 17-3。

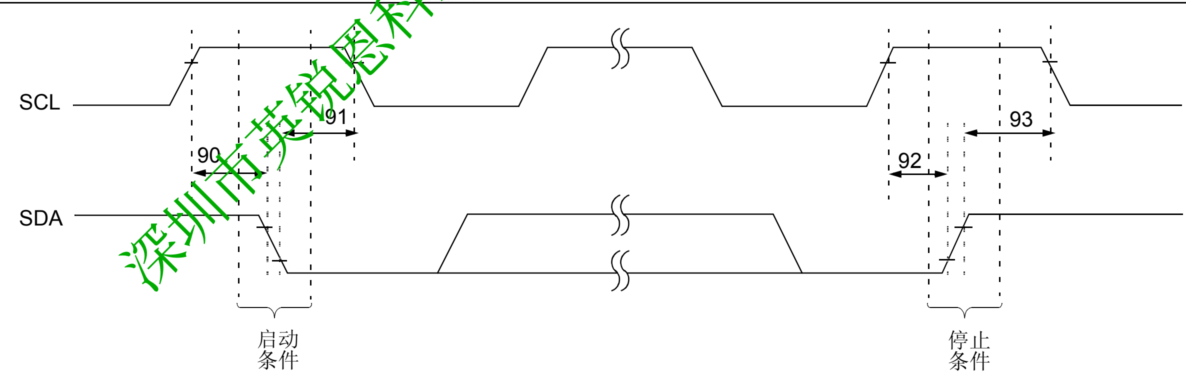
表 17-12：　　SPI 模式要求

参数编号	符号	特性		最小值	典型值 †	最大值	单位	条件
70*	TssL2scH，TssL2scL	SS↓ 到 SCK↓ 或 SCK↑ 输入的时间		TCY	—	—	ns	
71*	TscH	SCK 输入高电平时间 （从模式）		TCY + 20	—	—	ns	
72*	TscL	SCK 输入低电平时间 （从模式）		TCY + 20	—	—	ns	
73*	TDIV2scH，TDIV2scL	SDI 数据输入到 SCK 边沿的建立时间		100	—	—	ns	
74*	TscH2DIL，TscL2DIL	SDI 数据输入到 SCK 边沿的保持时间		100	—	—	ns	
75*	TDOR	SDO 数据输出上升时间	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
76*	TDOF	SDO 数据输出下降时间		—	10	25	ns	
77*	TssH2DOZ	SS↑ 到 SDO 输出高阻抗的时间		10	—	50	ns	
78*	TscR	SCK 输出上升时间（主模式）	3.0-5.5V	—	10	25	ns	
			2.0-5.5V	—	25	50	ns	
79*	TscF	SCK 输出下降时间 （主模式）		—	10	25	ns	
80*	TscH2DOV，TscL2DOV	SCK 边沿后 SDO 数据输出有效的时间	3.0-5.5V	—	—	50	ns	
			2.0-5.5V	—	—	145	ns	
81*	TDOV2scH，TDOV2scL	SDO 数据输出建立到 SCK 边沿的时间		TCY	—	—	ns	
82*	TssL2DOV	SS↓ 边沿后 SDO 数据输出有效的时间		—	—	50	ns	
83*	TscH2ssH，TscL2ssH	SCK 边沿后出现 SS ↑ 的时间		1.5TCY + 40	—	—	ns	

*　这些参数为特性值，未经测试。

†　除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

图 17-16：　　I^2C™ 总线启动 / 停止位时序



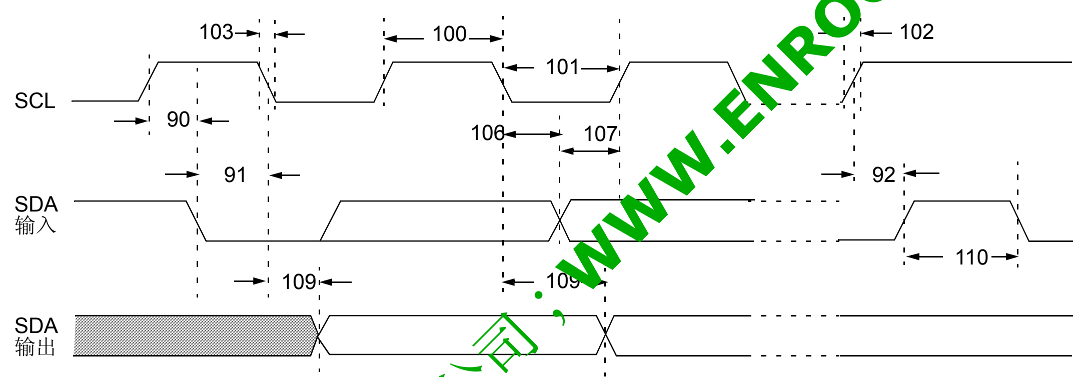
注：负载条件请参见图 17-3。

表 **17-13**： **I²C™ 总线启动 / 停止位要求**

参数编号	符号	特性		最小值	典型值	最大值	单位	条件
90*	$T_{SU:STA}$	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	—	—		
91*	$T_{HD:STA}$	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	—	—		
92*	$T_{SU:STO}$	停止条件建立时间	100 kHz 模式	4700	—	—	ns	
			400 kHz 模式	600	—	—		
93	$T_{HD:STO}$	停止条件保持时间	100 kHz 模式	4000	—	—	ns	
			400 kHz 模式	600	—	—		

* 这些参数为特性值，未经测试。

图 **17-17**： **I²C™ 总线数据时序**



注：负载条件请参见图 17-3。

表 **17-14：** **I²C™** 总线数据要求

参数编号	符号	特性		最小值	最大值	单位	条件
100*	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	1.5TCY	—		
101*	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	1.5TCY	—		
102*	TR	SDA 和 SCL 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	20 + 0.1CB	300	ns	CB 值规定在 10-400 pF 之间
103*	TF	SDA 和 SCL 下降时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	20 + 0.1CB	300	ns	CB 值规定在 10-400 pF 之间
90*	TSU:STA	启动条件建立时间	100 kHz 模式	4.7	—	μs	仅与重复启动条件相关
			400 kHz 模式	0.6	—	μs	
91*	THD:STA	启动条件保持时间	100 kHz 模式	4.0	—	μs	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	μs	
106*	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	μs	
107*	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	（注 **2**）
			400 kHz 模式	100	—	ns	
92*	TSU:STO	停止条件建立时间	100 kHz 模式	4.7	—	μs	
			400 kHz 模式	0.6	—	μs	
109*	TAA	时钟输出有效时间	100 kHz 模式	—	3500	ns	（注 **1**）
			400 kHz 模式	—	—	ns	
110*	TBUF	总线空闲时间	100 kHz 模式	4.7	—	μs	在启动一个新的传输前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	μs	
	CB	总线容性负载		—	400	pF	

* 这些参数为特性值，但未经测试。

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这个内部最小延时以覆盖 SCL 下降沿的未定义区域（最小值 300 ns）。

2： 快速模式（400 kHz）的 I²C 总线器件也可在标准模式（100 kHz）的 I²C 总线系统中使用，但必须满足 TSU:DAT ≥ 250 ns 的要求。如果快速模式器件没有延长 SCL 信号的低电平周期，则必然满足此条件。如果该器件延长了 SCL 信号的低电平周期，其下一个数据位必须输出到 SDA 线。 SCL 线被释放前，根据标准模式 I²C 总线规范， TR max. + TSU:DAT = 1000 + 250 = 1250 ns。

表 17-15： A/D 转换器 （ADC）特性

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°C

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	位	
AD02	EIL	积分误差	—	—	±1	LSb	VREF = 5.12V
AD03	EDL	微分误差	—	—	±1	LSb	10 位不会丢失编码 VREF = 5.12V
AD04	EOFF	失调误差	—	—	±1	LSb	VREF = 5.12V
AD04A			—	+1.5	+3.0	LSb	（仅限 PIC16F677）
AD07	EGN	增益误差	—	—	±1	LSb	VREF = 5.12V
AD06 AD06A	VREF	参考电压 (3)	2.2 2.5	—	— VDD	V	 绝对最小值以确保 1 LSb 精度
AD07	VAIN	满量程范围	VSS	—	VREF	V	
AD08	ZAIN	模拟电压源的推荐阻抗	—	—	10	kΩ	
AD09*	IREF	VREF 输入电流 (3)	10	—	1000	μA	在采集 VAIN 期间。 基于 VHOLD 对 VAIN 的微分。
			—	—	50	μA	在 A/D 转换周期期间

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

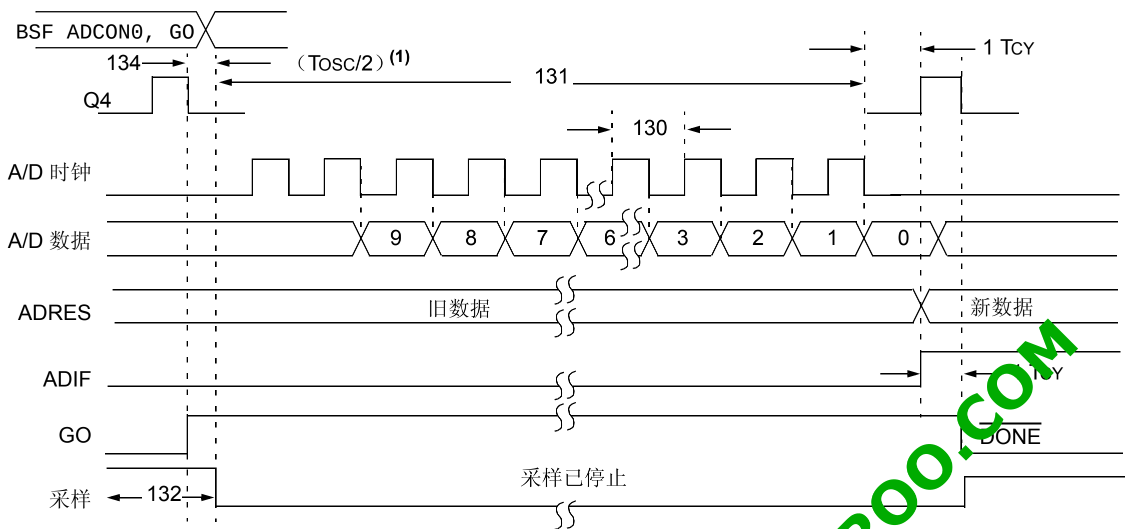
注 1： 总的绝对误差包括积分、微分、失调和增益误差。

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 来自选择作为参考输入的外部 VREF 或 VDD 引脚。

4： 当 ADC 关闭时，它除了消耗很少的泄漏电流外，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。

图 17-18： A/D 转换时序 （正常模式）



注 1： 如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

表 17-16： A/D 转换要求

标准工作条件 （除非另外声明）
工作温度 -40°C ≤ TA ≤ +125°C

参数编号	符号	特性	最小值	典型值†	最大值	单位	条件
130*	TAD	A/D 时钟周期	1.5	—	—	µs	基于 TOSC， VREF ≥ 2.5V
			3.0*	—	—	µs	基于 TOSC， VREF 满量程
		A/D 内部 RC 振荡器周期					ADCS<1:0> = 11 （RC 模式）
			3.0*	6.0	9.0*	µs	VDD = 2.5V 时
			2.0*	4.0	6.0*	µs	VDD = 5.0V 时
131	TCNV	转换时间 （不包括采集时间） (1)	—	11	—	TAD	将 GO 位设定为 A/D 结果寄存器中的新数据
132*	TACQ	采集时间	(2)	11.5	—	µs	
			5*	—	—	µs	最小时间为放大器稳定时间。如果 “新的” 输入电压相对于上次采样电压 （保存在 CHOLD 中）的变化不超过 1 LSb （即 4.096V 时为 4.1 mV），便可采用此时间。
134	TGO	Q4 到 A/D 时钟启动	—	TOSC/2	—	—	如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

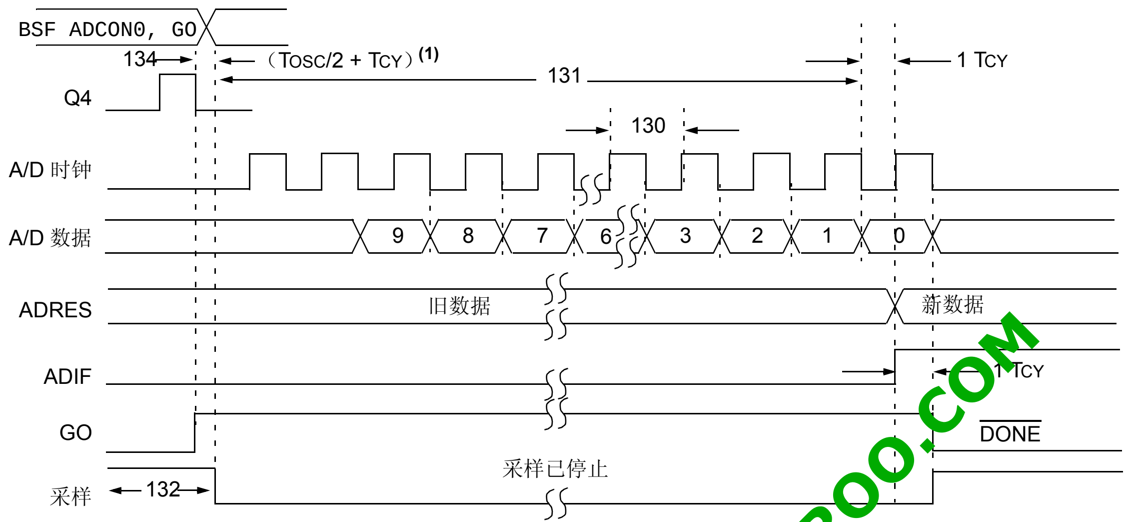
* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： ADRESH 和 ADRESL 寄存器可在下一个 TCY 周期被读取。

2： 最小条件请参见表 9-1。

图 **17-19：** **A/D** 转换时序 （休眠模式）



注 **1：** 如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

表 17-17： A/D 转换要求 （休眠模式）

标准工作条件 （除非另外声明） 工作温度 -40°C ≤ TA ≤ +125°C							
参数编号	**符号**	**特性**	**最小值**	**典型值 †**	**最大值**	**单位**	**条件**
130*	TAD	A/D 内部 RC 振荡器周期	 3.0* 2.0*	 6.0 4.0	 9.0* 6.0*	 μs μs	ADCS<1:0> = 11 （RC 模式） VDD = 2.5V 时 VDD = 5.0V 时
131	TCNV	转换时间 （不包括采集时间） (1)	—	11	—	TAD	
132*	TACQ	采集时间	(2) 5*	11.5 —	— —	μs μs	 最小时间为放大器稳定时间。如果 “新的” 输入电压相对于上次采样电压 （保存在 CHOLD 中）的变化不超过 1 LSb （即 4.096V 时为 4.1 mV），便可采用此时间。
134	TGO	Q4 到 A/D 时钟启动	—	TOSC/2 + TCY	—	—	如果将 A/D 时钟源选择为 RC，在 A/D 时钟启动前要加上一个 TCY 时间，用以执行 SLEEP 指令。

* 这些参数为特性值，未经测试。

† 除非另外声明，否则 “典型值” 栏中的数据均为 5.0V 和 25°C 条件下的值。这些参数仅供设计参考，未经测试。

注 1： ADRES 寄存器可在下一个 TCY 周期被读取。

2： 最小条件请参见表 9-1。

18.0 直流和交流特性图表

本节提供的图表仅供**设计参考**并且**未经测试**。

有些图表中的数据**超出了规定的工作范围** （例如，超出了规定的 VDD 范围）。这些图表**仅供参考**，器件只有工作在规定范围内才可确保正常工作。

注：	本注释以下所提供的图表均为基于有限数量样本的统计汇总，仅供参考。此处列出的特性未经测试，不作保证。有些图表中的数据超出了规定的工作范围 （例如，超出了规定的电源范围），因此不在担保范围内。

“典型值”表示 25°C 时的平均分布。“最大值”或“最小值”分别表示（平均值 + 3σ）或（平均值 – 3σ），其中 σ 为各温度范围内的标准公差。

图 18-1：　不同 VDD 的典型 IDD － FOSC 曲线 （EC 模式）

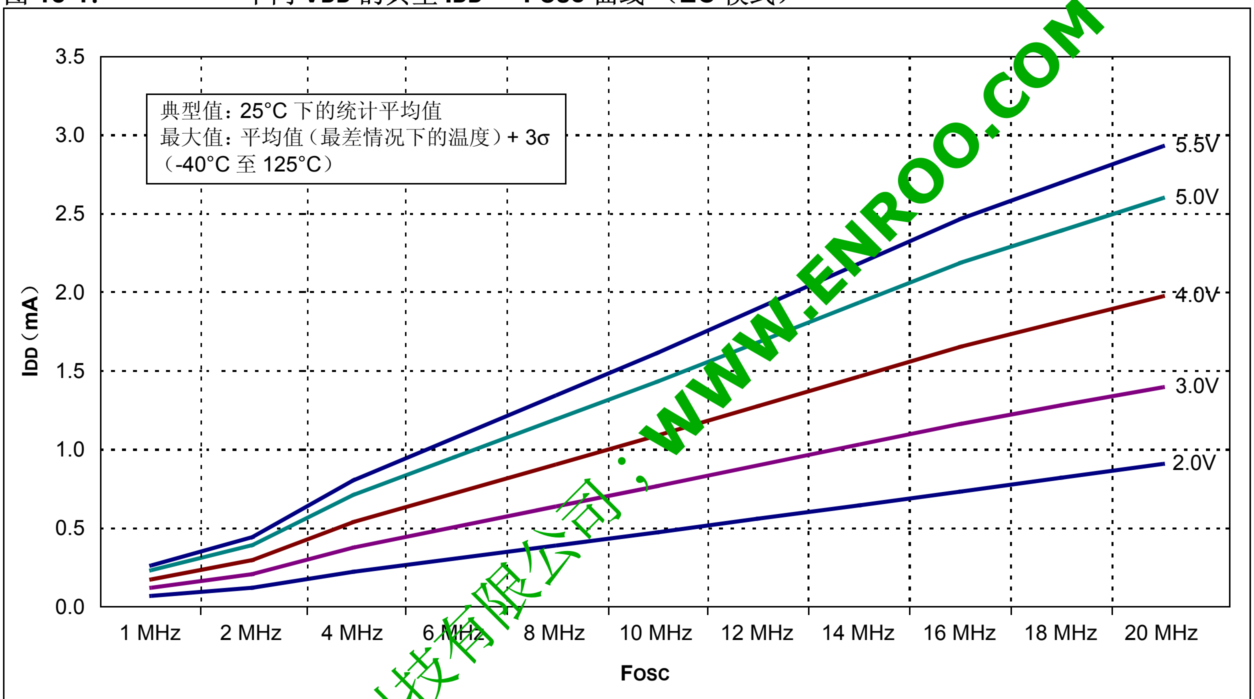


图 18-2： 不同 VDD 的最大 IDD — FOSC 曲线 （EC 模式）

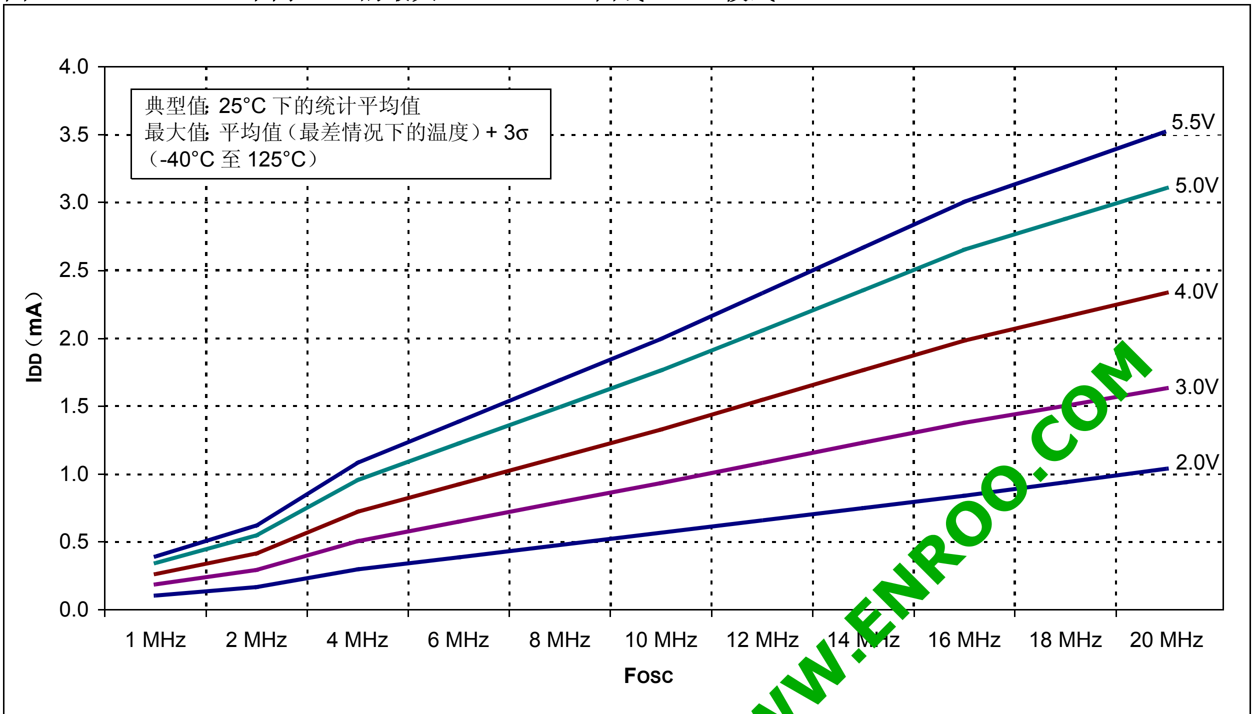


图 18-3： 不同 VDD 的典型 IDD — FOSC 曲线 （HS 模式）

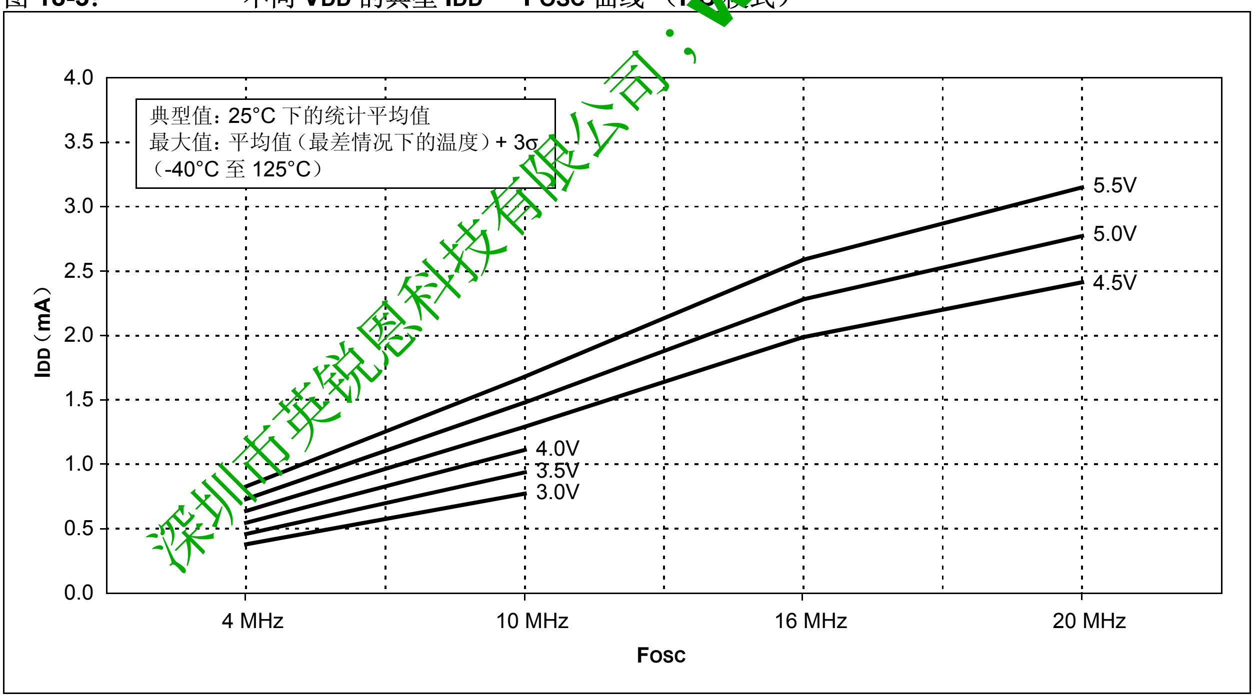


图 18-4： 不同 VDD 的最大 IDD — FOSC 曲线 （HS 模式）

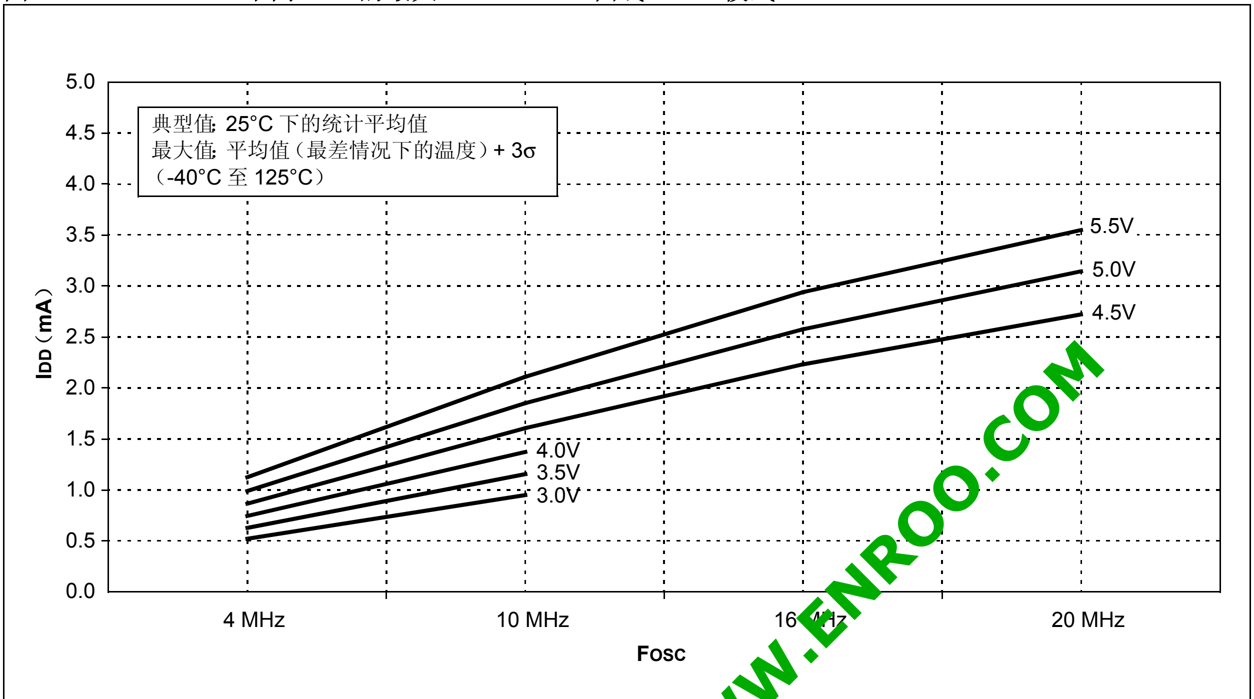


图 18-5： 不同 FOSC 的典型 IDD — VDD 曲线 （XT 模式）

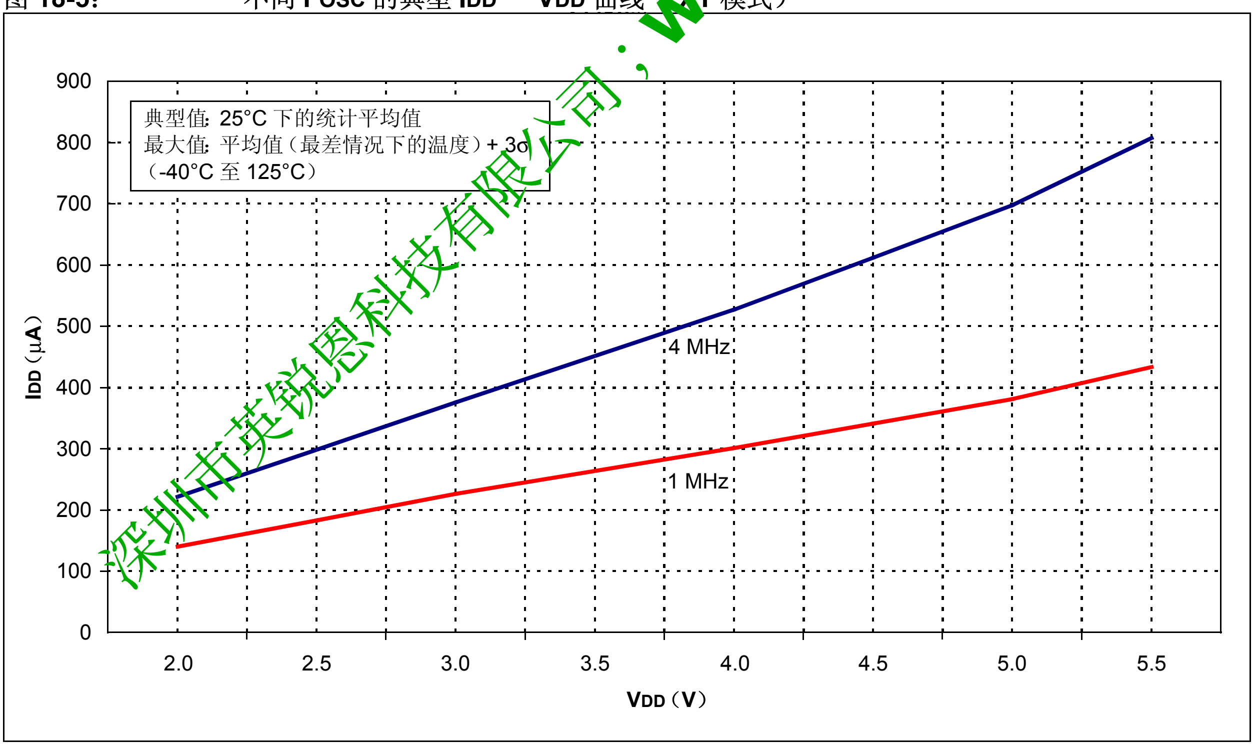


图 18-6： 不同 FOSC 的最大 IDD — VDD 曲线 （XT 模式）

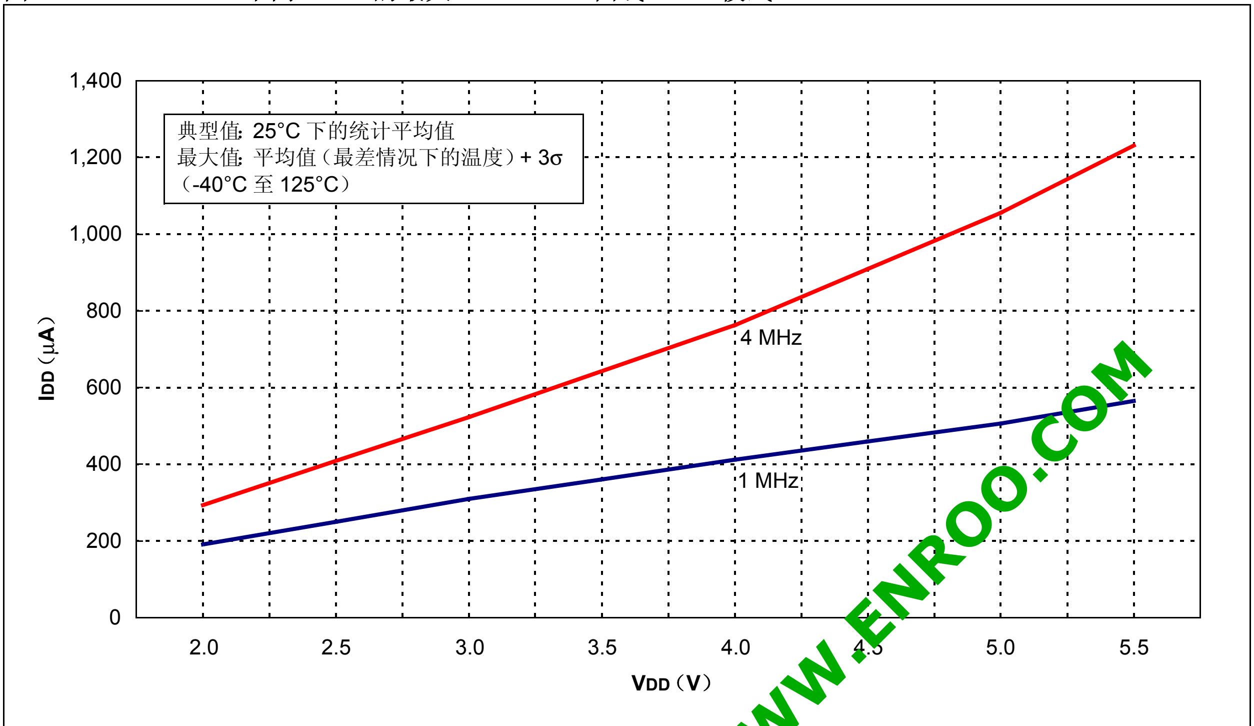


图 18-7： IDD — VDD 曲线 （LP 模式）

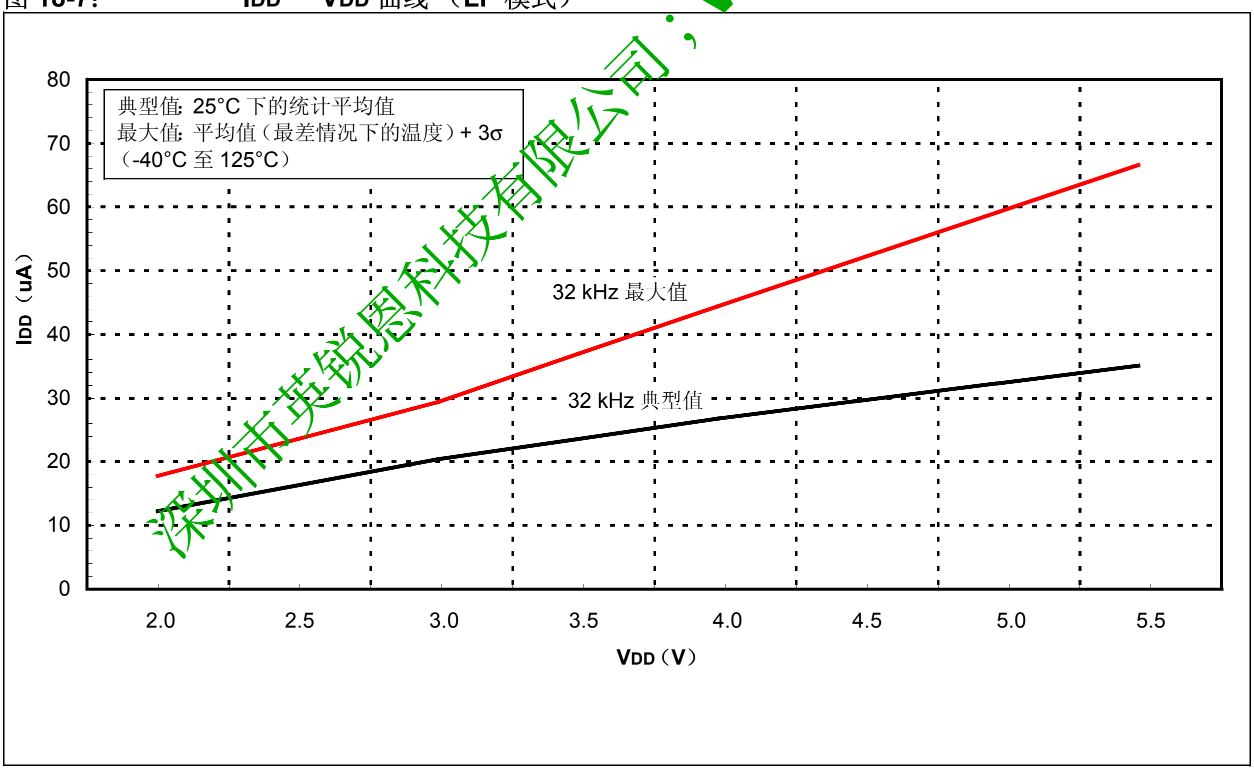


图 18-8： 不同 FOSC 的典型 IDD — VDD 曲线 （EXTRC 模式）

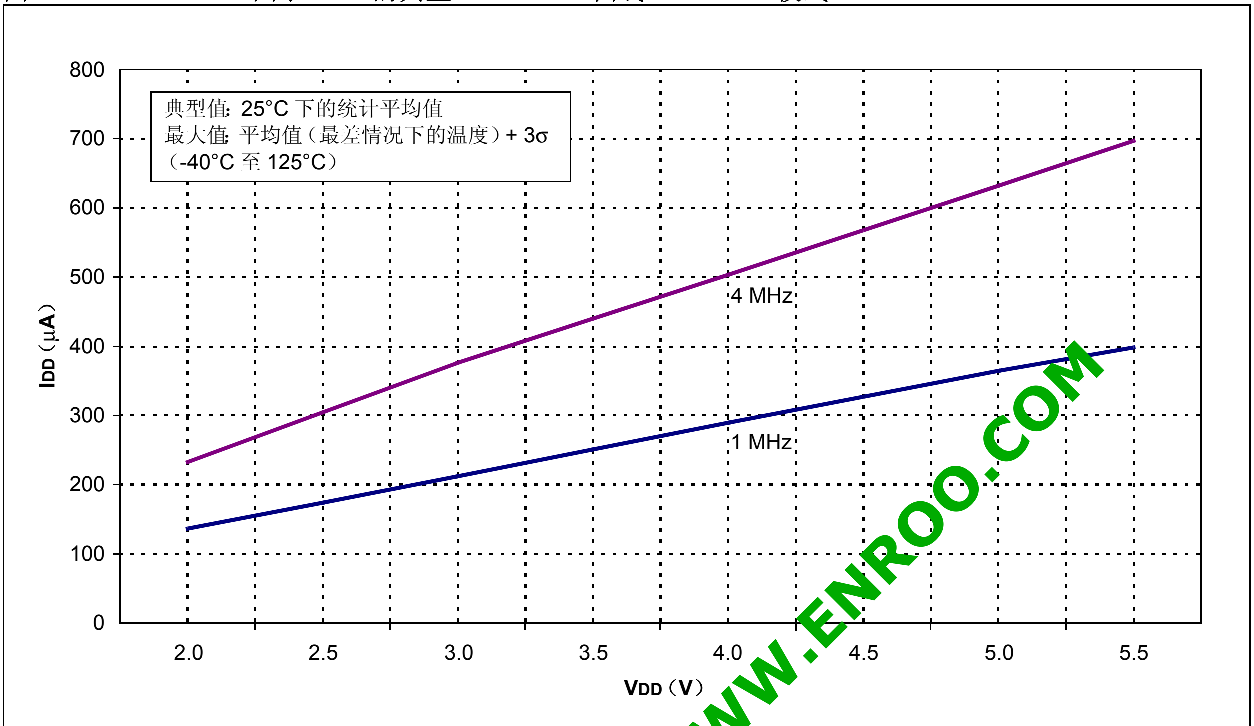


图 18-9： 不同 FOSC 的最大 IDD — VDD 曲线 （EXTRC 模式）

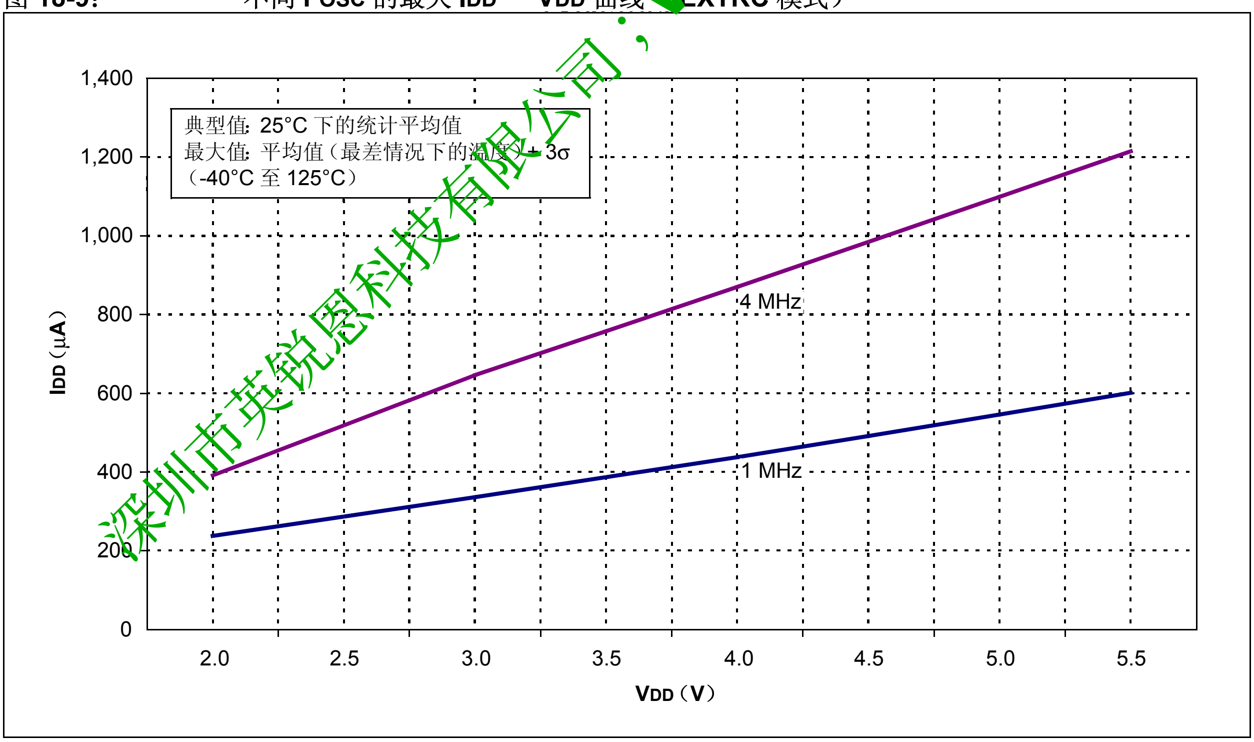


图 18-10： 不同 FOSC 的 IDD — VDD 曲线 （LFINTOSC 模式， 31 kHz）

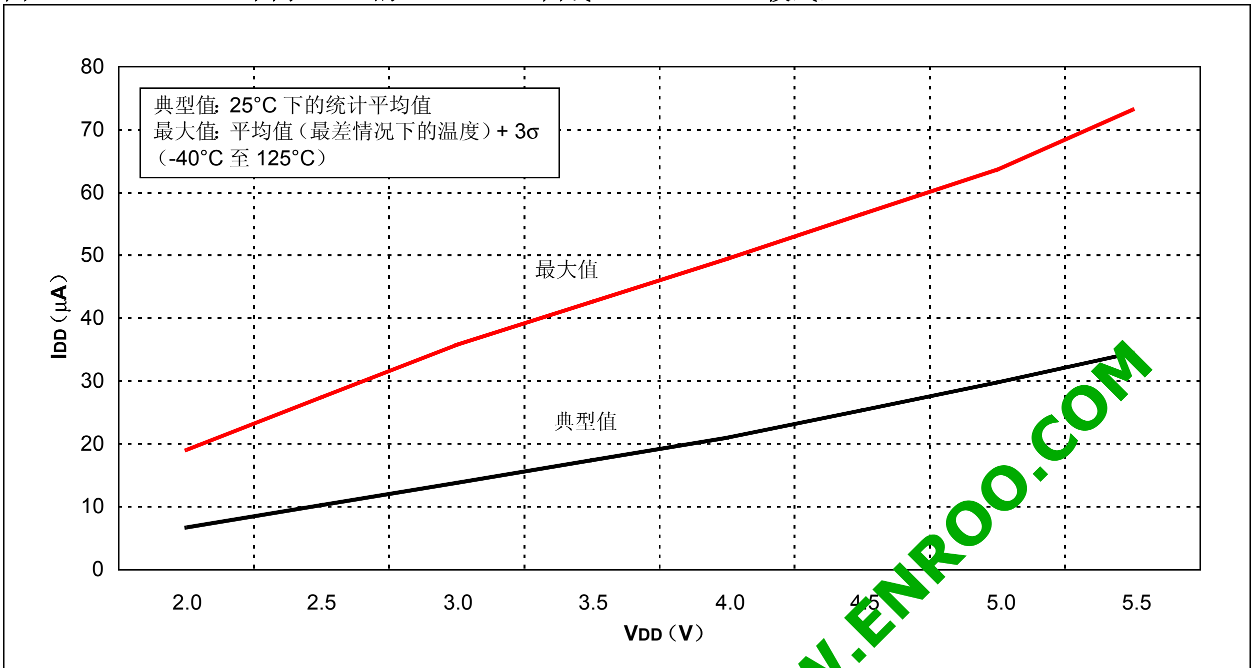


图 18-11： 不同 VDD 的典型 IDD — FOSC 曲线 （HFINTOSC 模式）

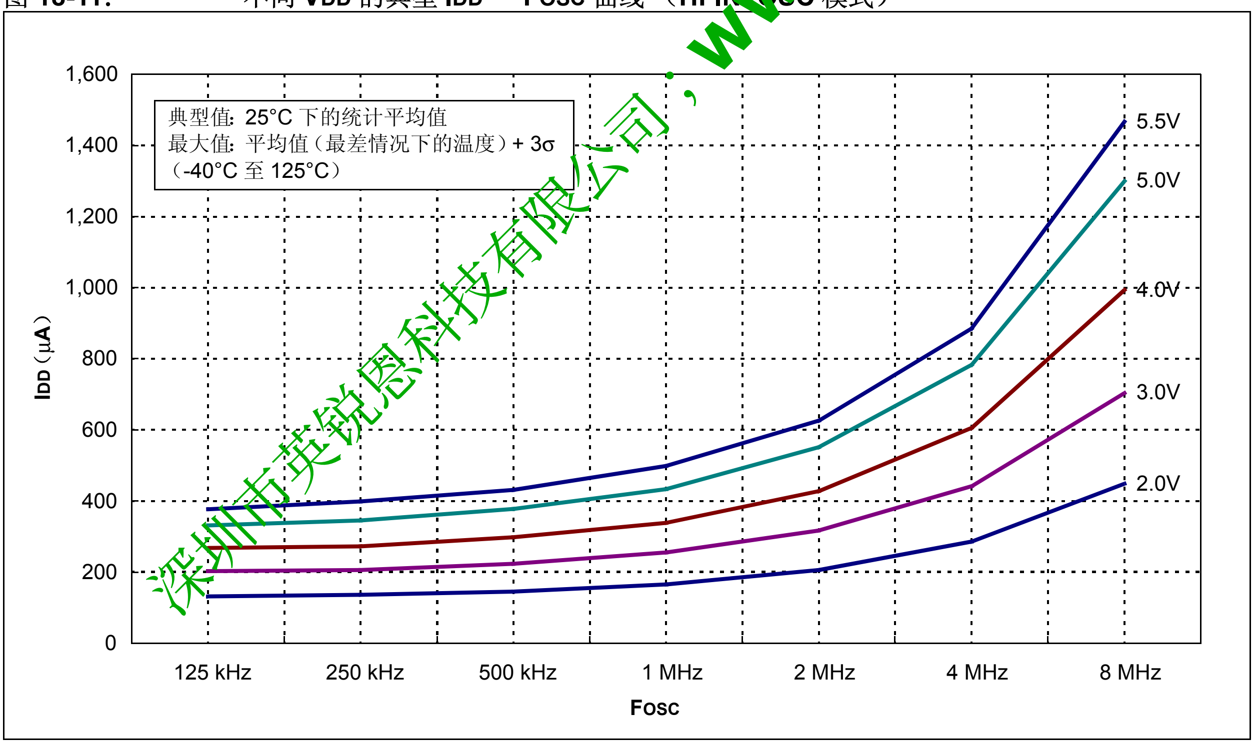


图 18-12：　不同 VDD 的最大 IDD — FOSC 曲线 （HFINTOSC 模式）

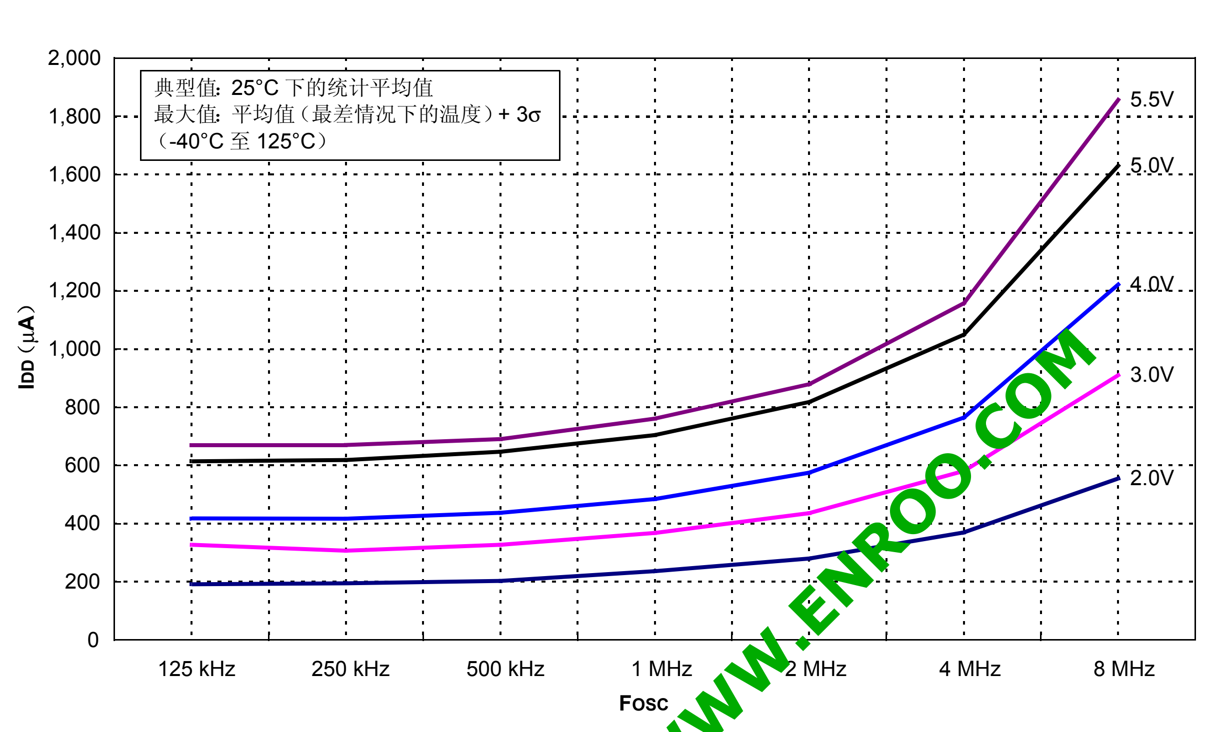


图 18-13：　典型 IPD — VDD 曲线 （休眠模式，禁止所有外设）

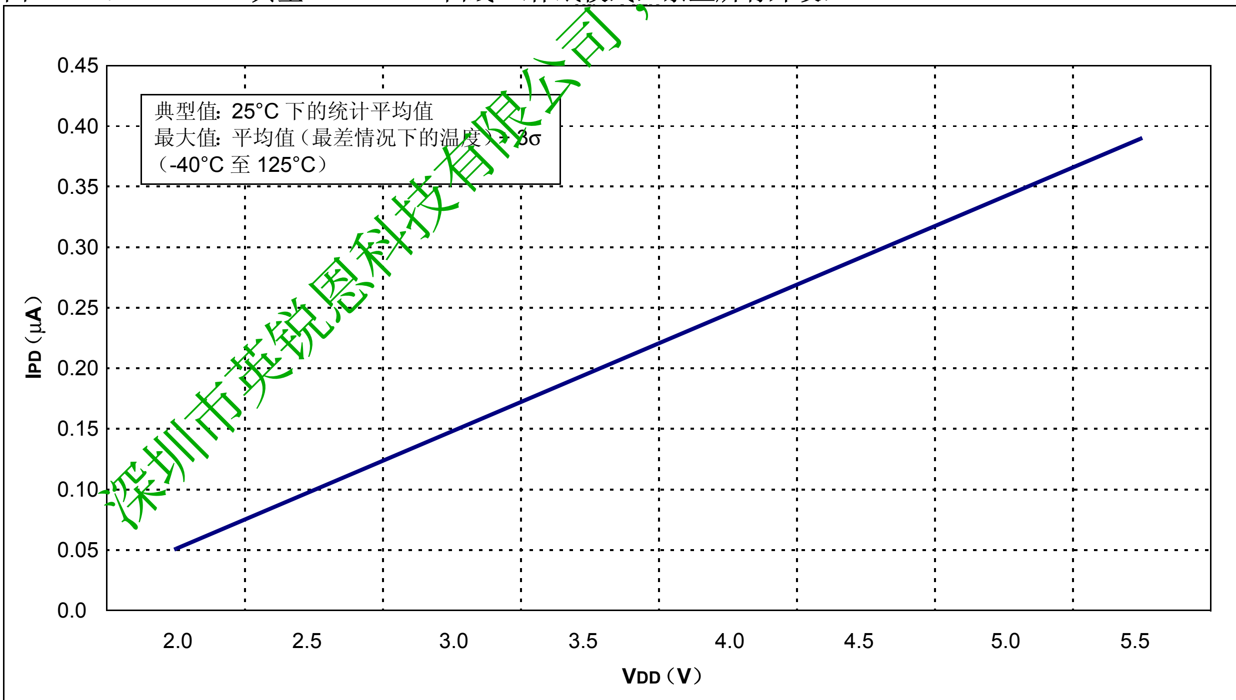


图 18-14： 最大 IPD — VDD 曲线 （休眠模式，禁止所有外设）

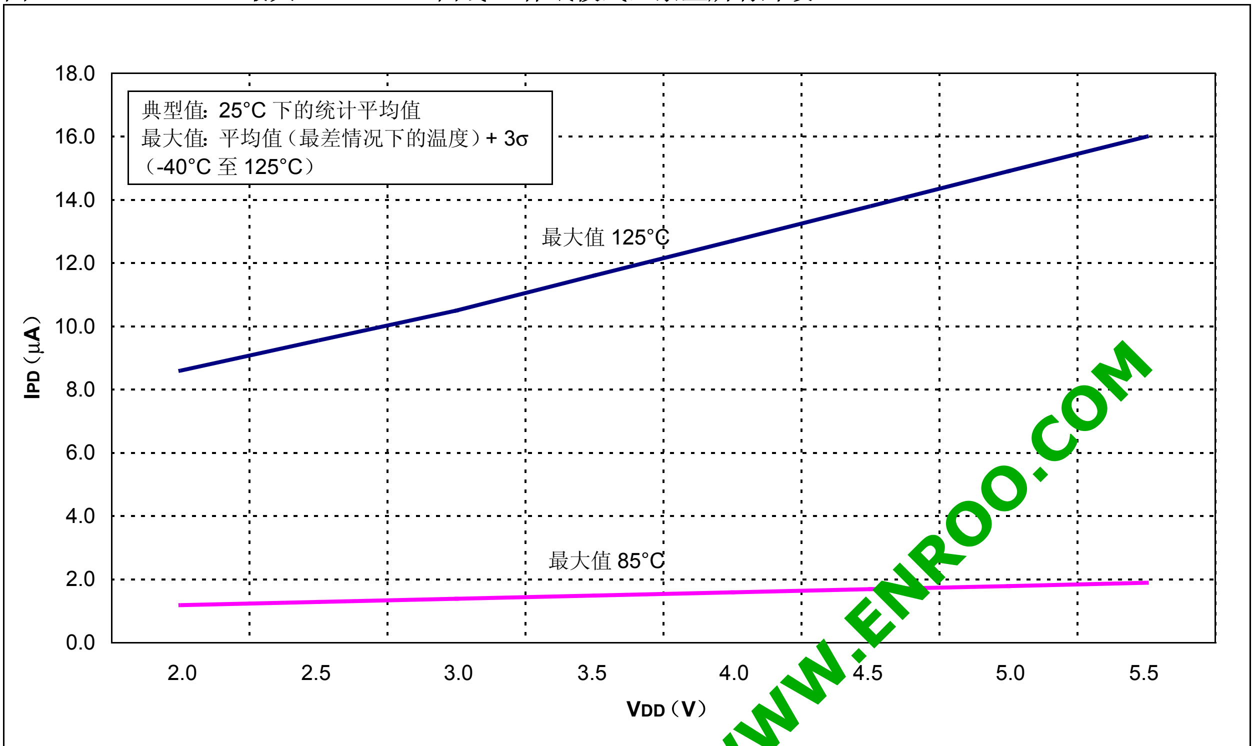


图 18-15： 比较器 IPD — VDD 曲线 （两个比较器均使能）

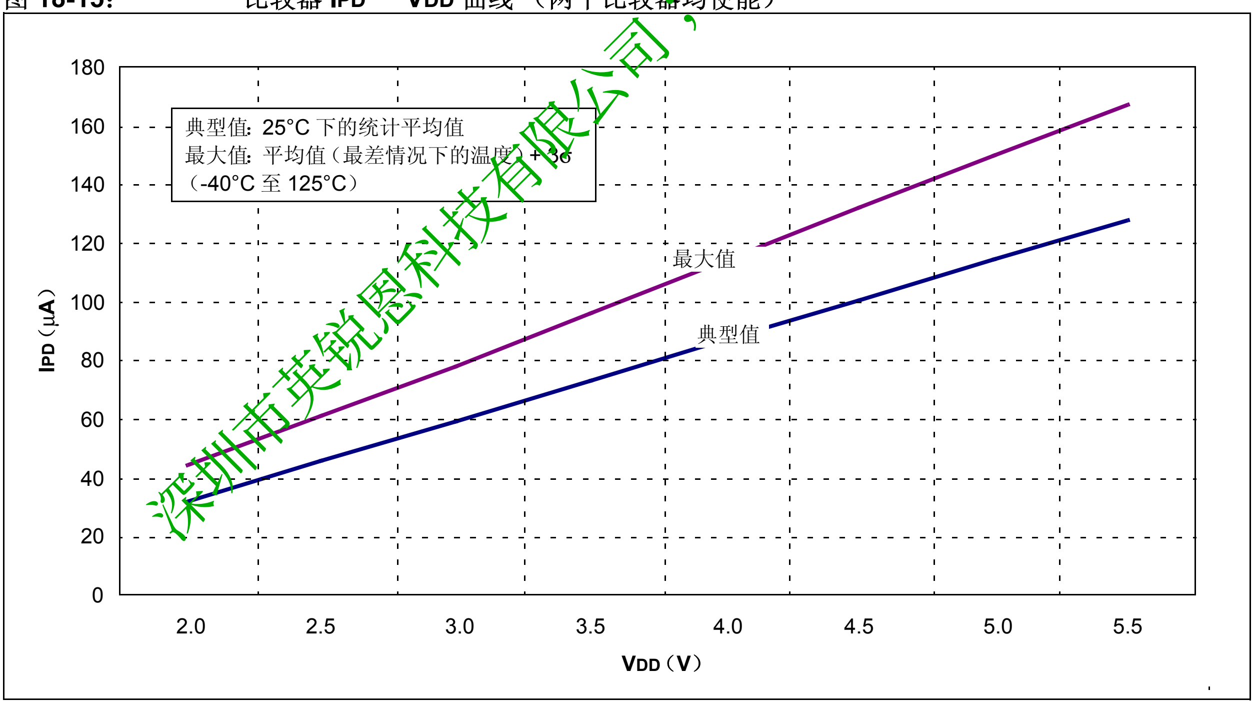


图 18-16： 不同温度的 BOR IPD — VDD 曲线

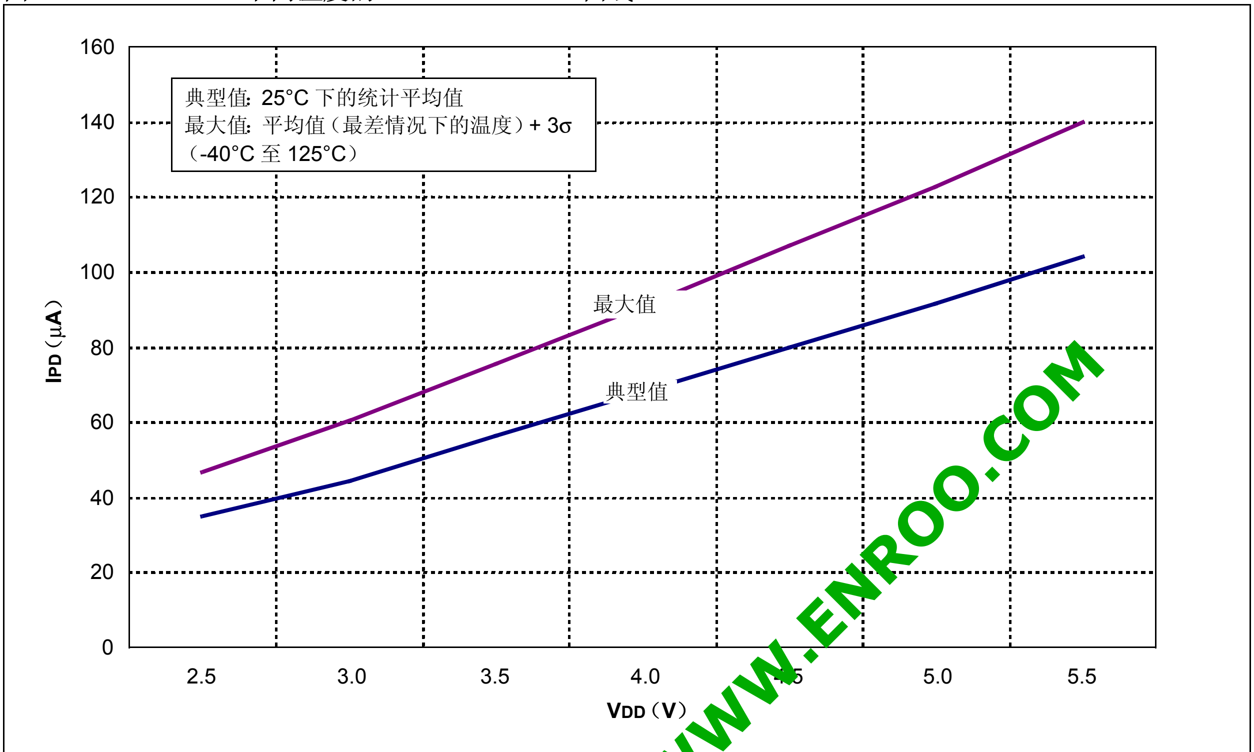


图 18-17： 不同温度的典型 WDT IPD — VDD 曲线

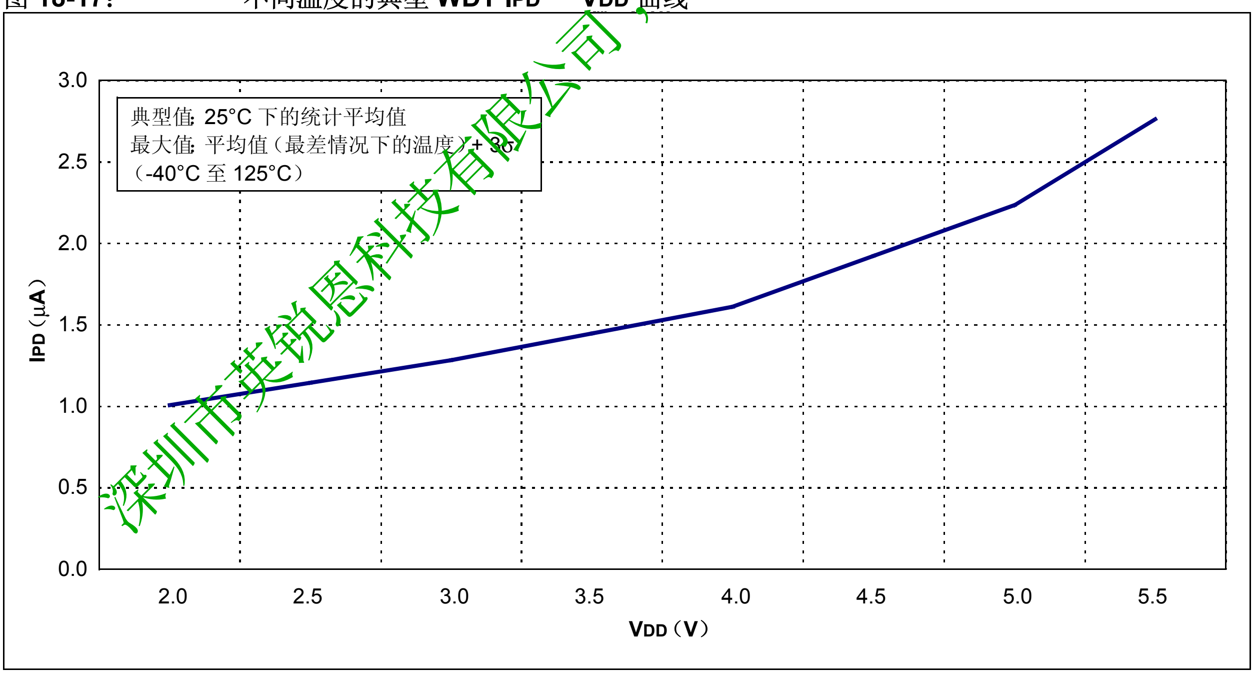


图 18-18： 不同温度的最大 WDT IPD — VDD 曲线

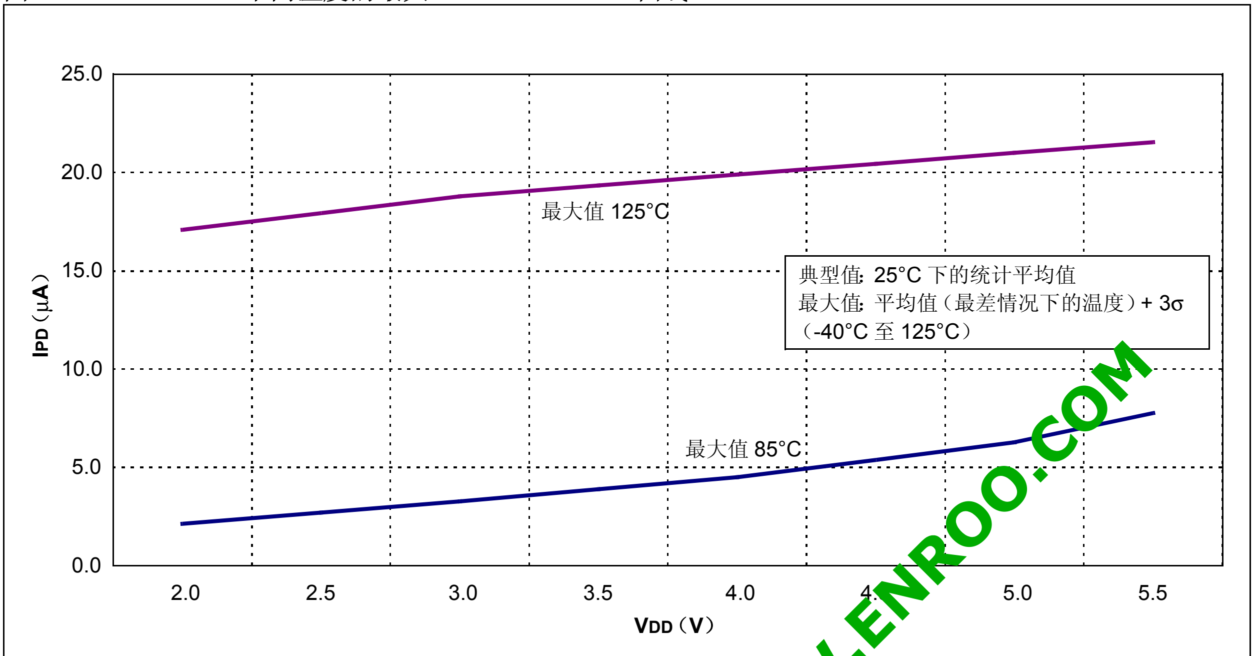


图 18-19： 不同温度的 WDT 周期 — VDD 曲线

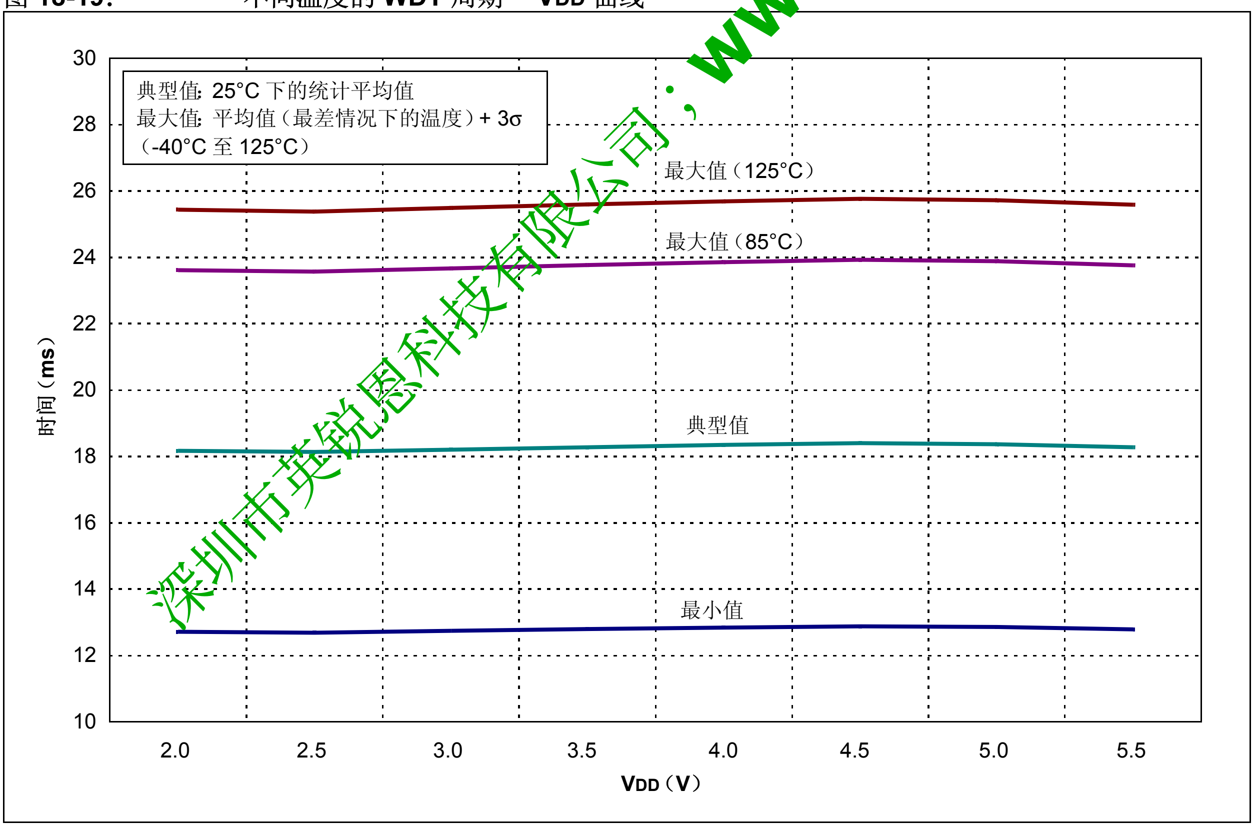


图 18-20： 不同 VDD 的 WDT 周期—温度曲线 （5.0V）

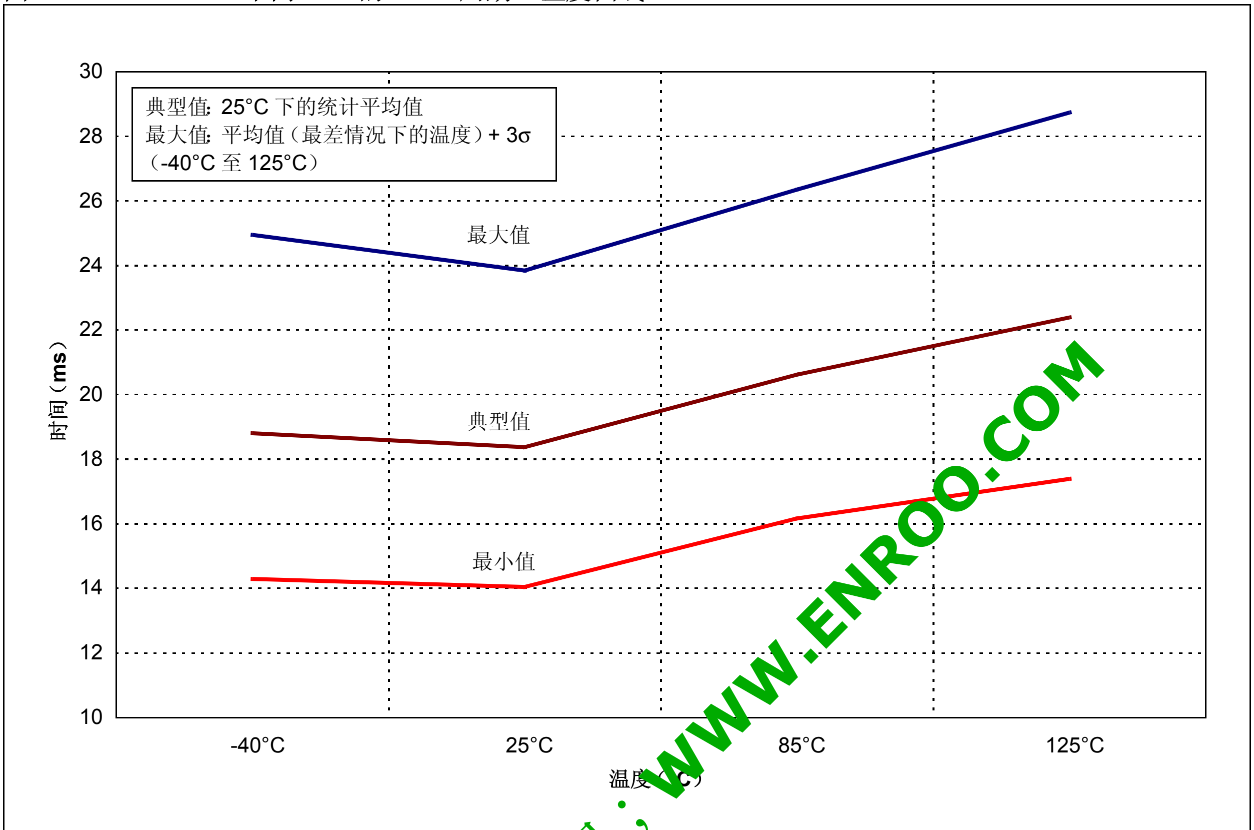


图 18-21： 不同温度的 CVREF IPD — VDD 曲线 （高电压范围）

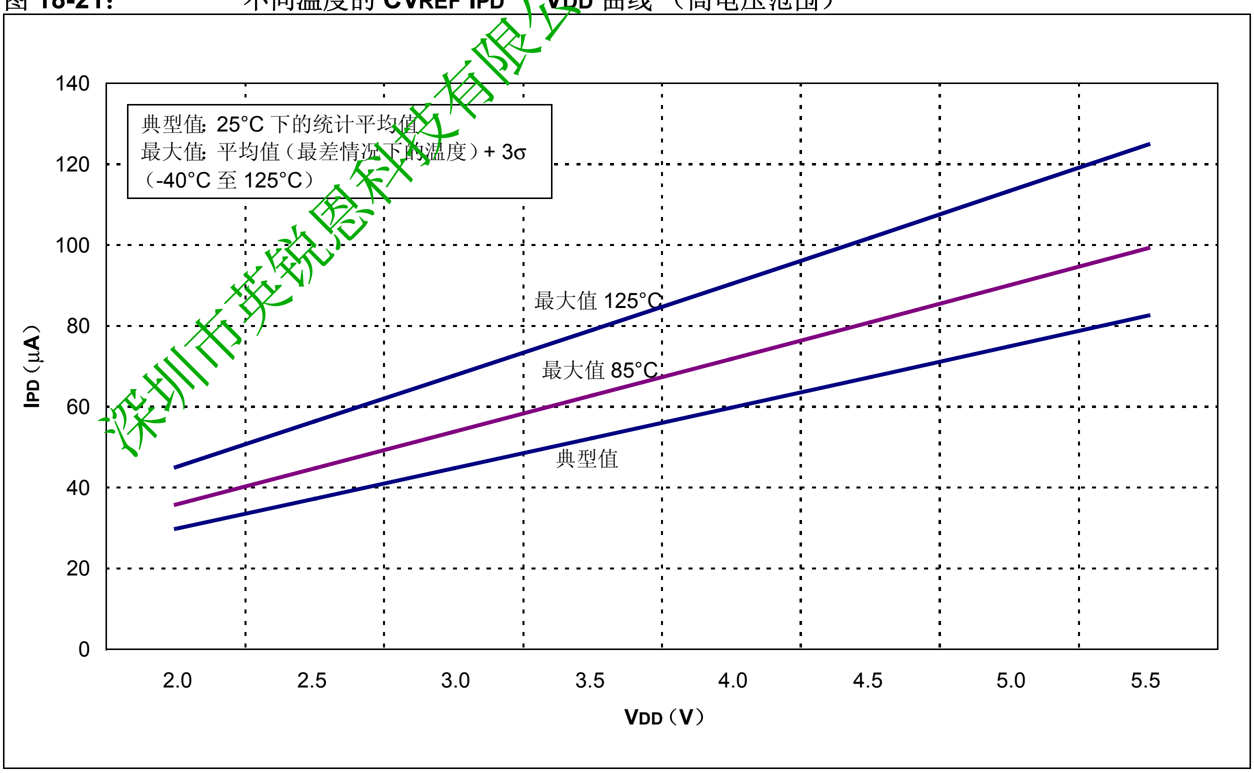


图 18-22: 不同温度的 CVREF IPD — VDD 曲线 （低电压范围）

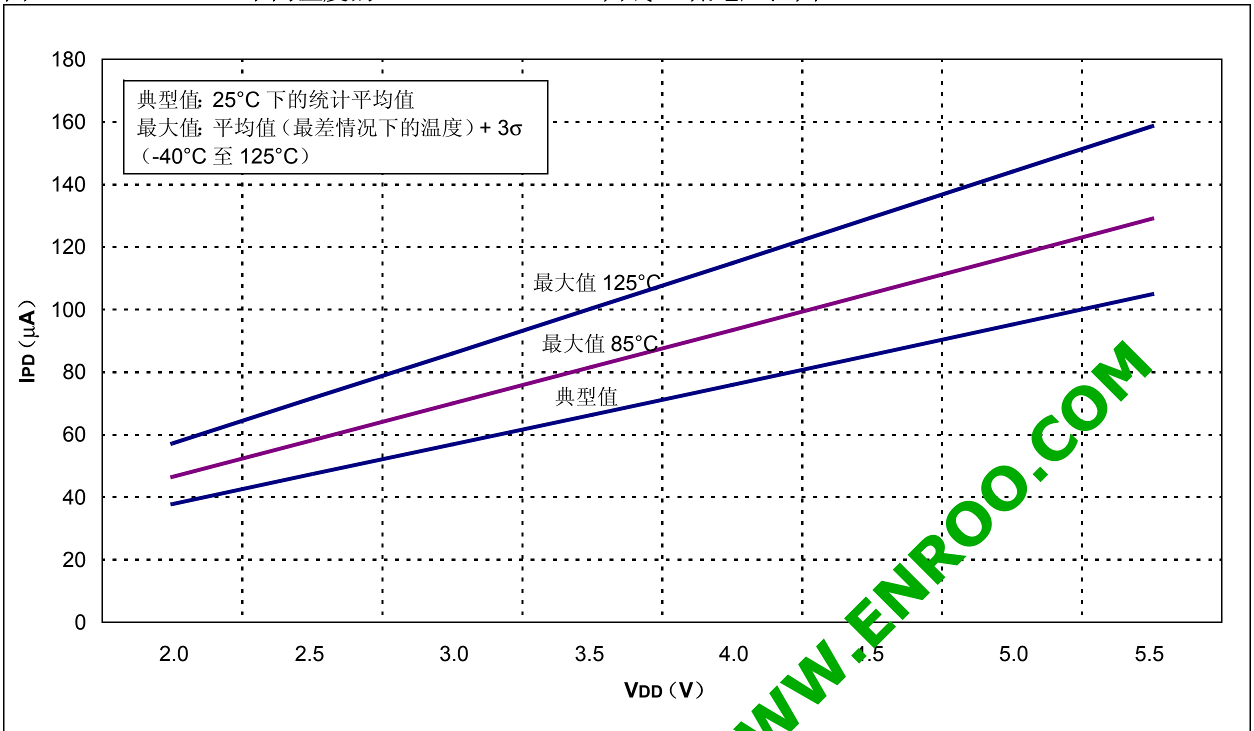


图 18-23: 典型 VP6 参考 IPD — VDD 曲线 （25°C）

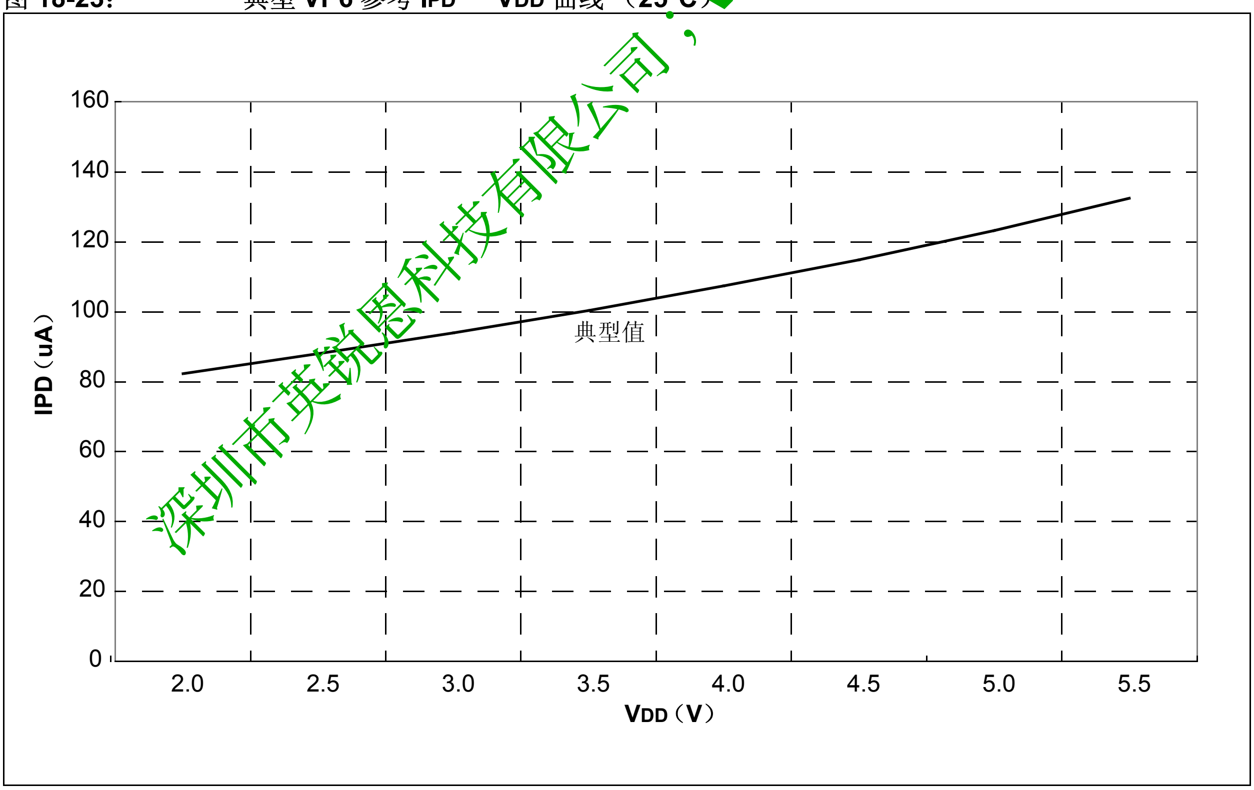


图 18-24：　不同温度的最大 VP6 参考 IPD － VDD 曲线

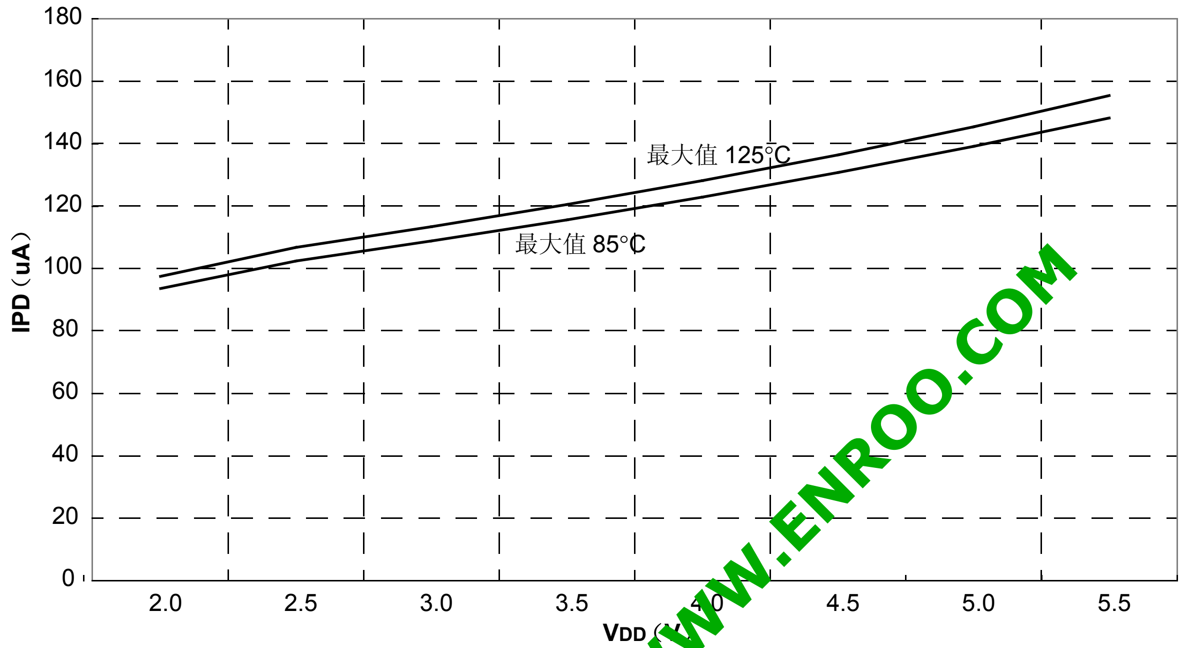


图 18-25：　不同温度的 T1OSC IPD － VDD 曲线 （32 kHz）

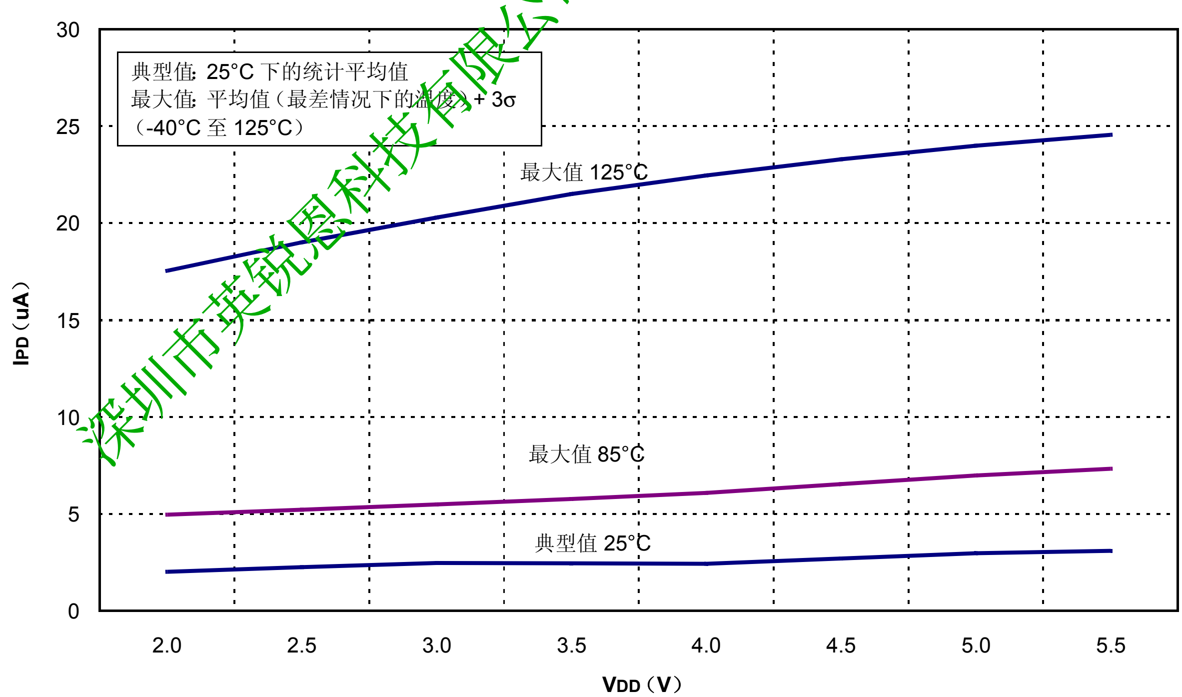


图 18-26： 不同温度的 VOL — IOL 曲线 （VDD = 3.0V）

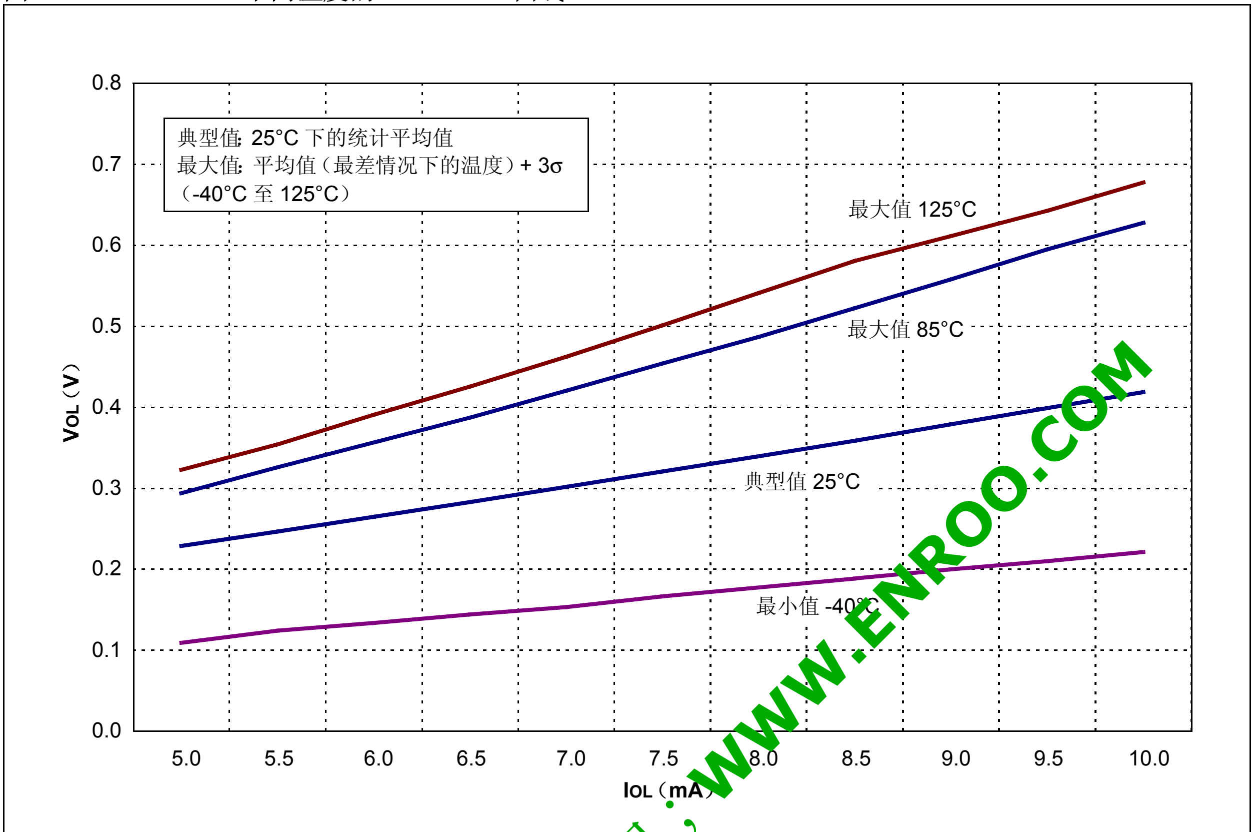


图 18-27： 不同温度的 VOL — IOL 曲线 （VDD = 5.0V）

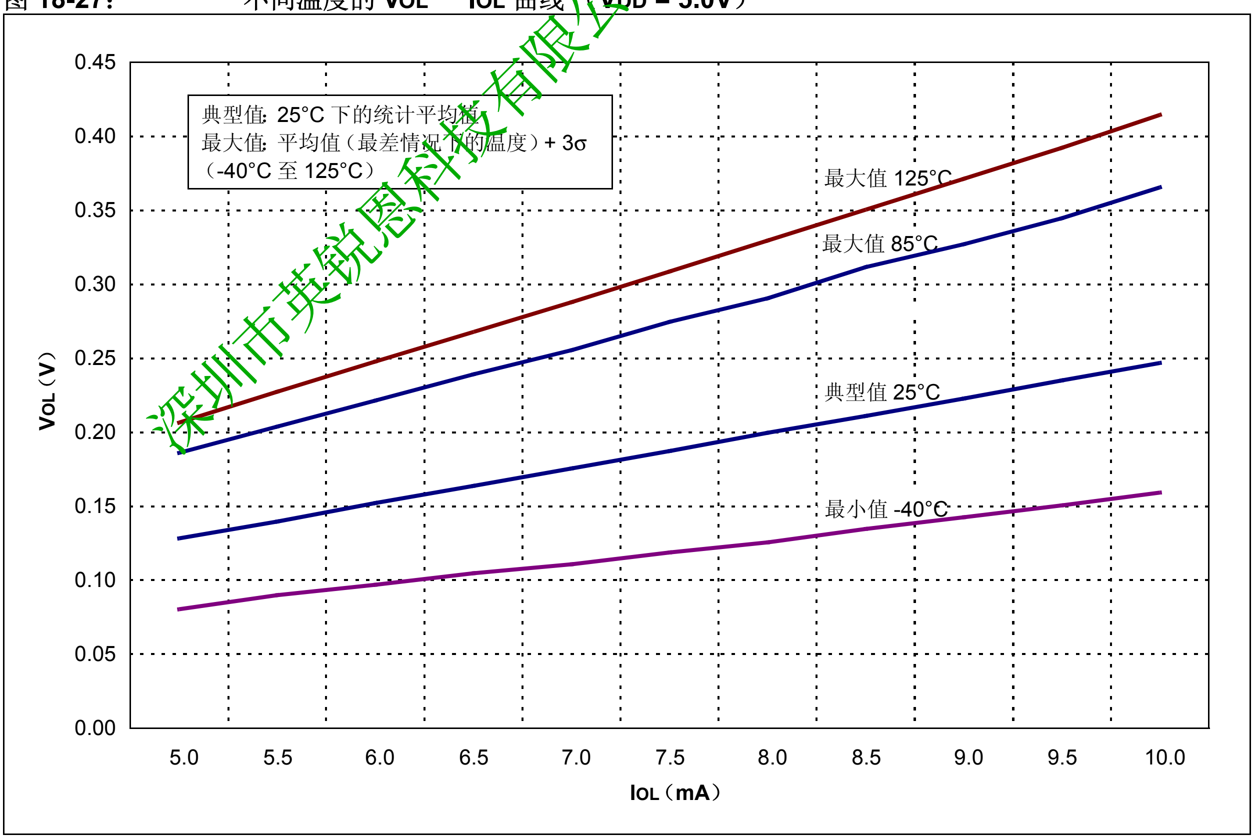


图 18-28： 不同温度的 VOH — IOH 曲线 （VDD = 3.0V）

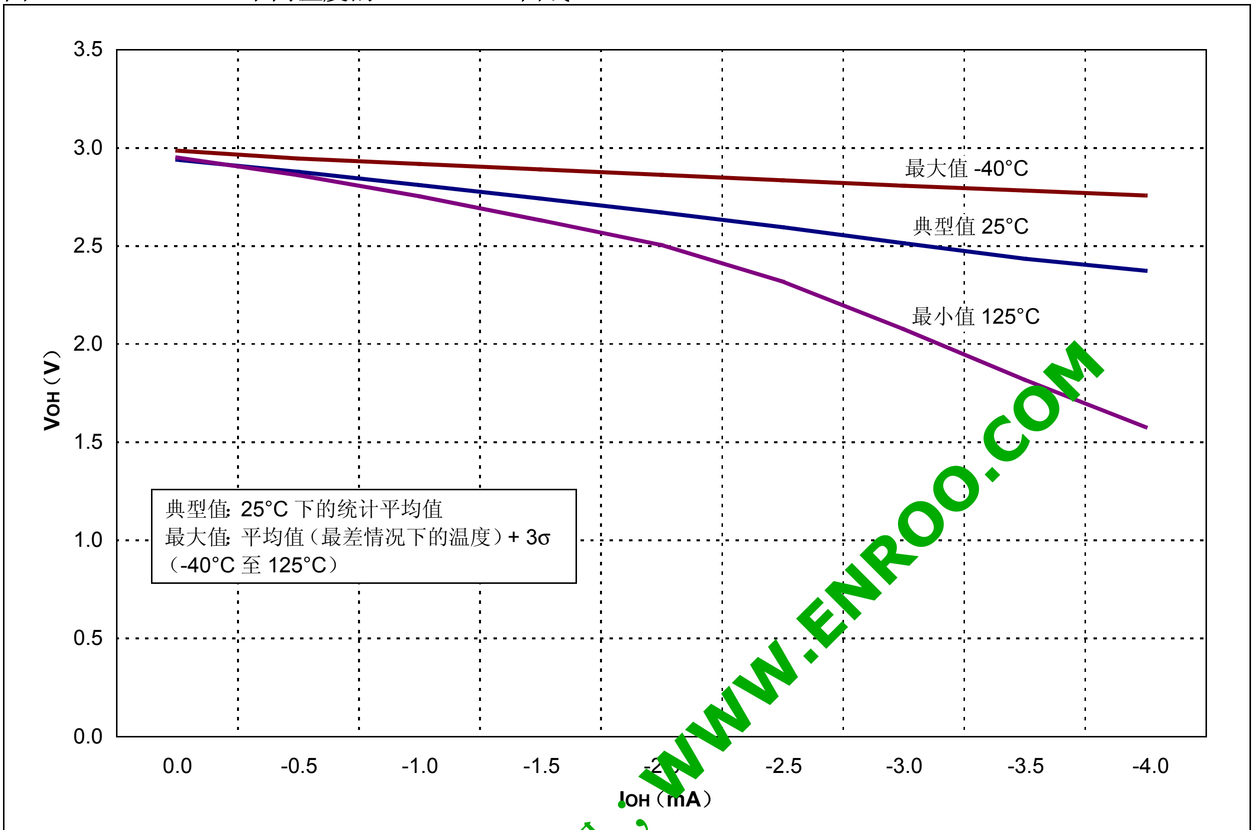


图 18-29： 不同温度的 VOH — IOH 曲线 （VDD = 5.0V）

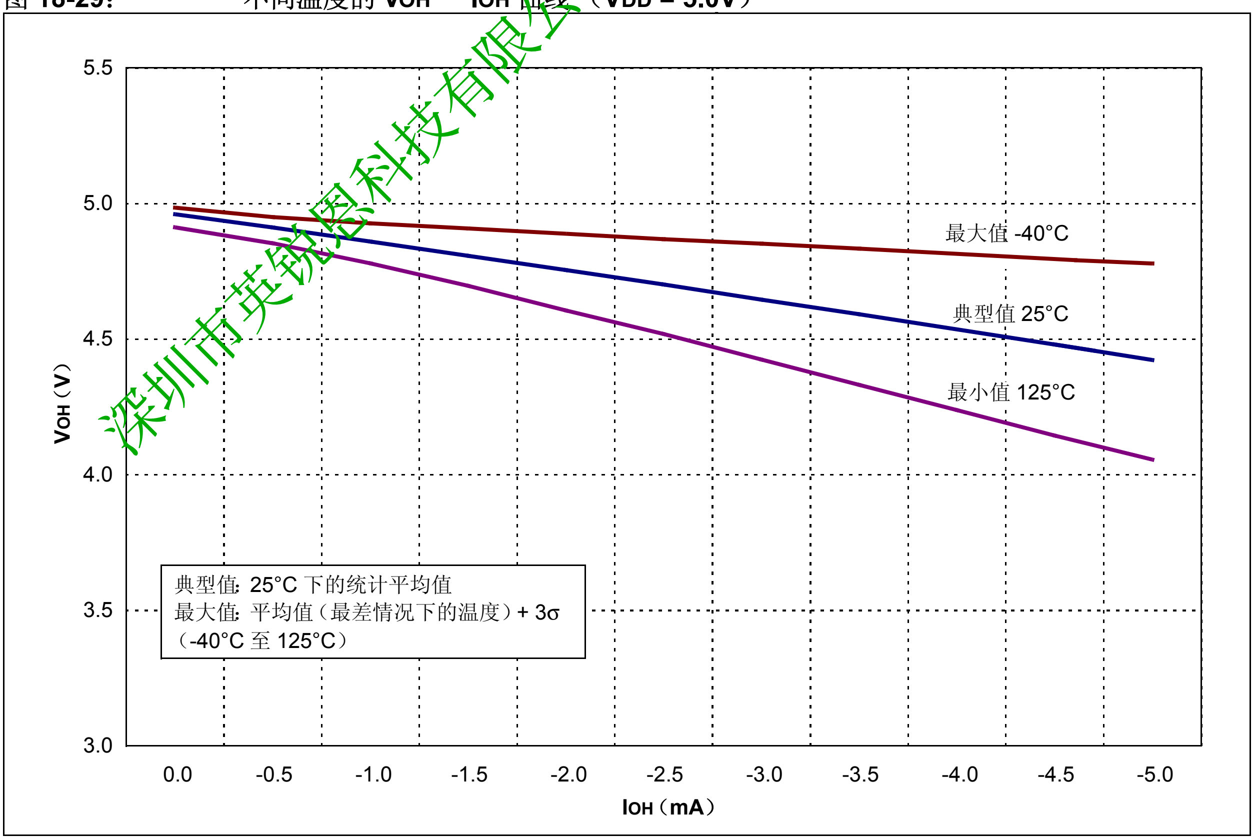


图 18-30： 不同温度的 TTL 输入门限 VIN — VDD 曲线

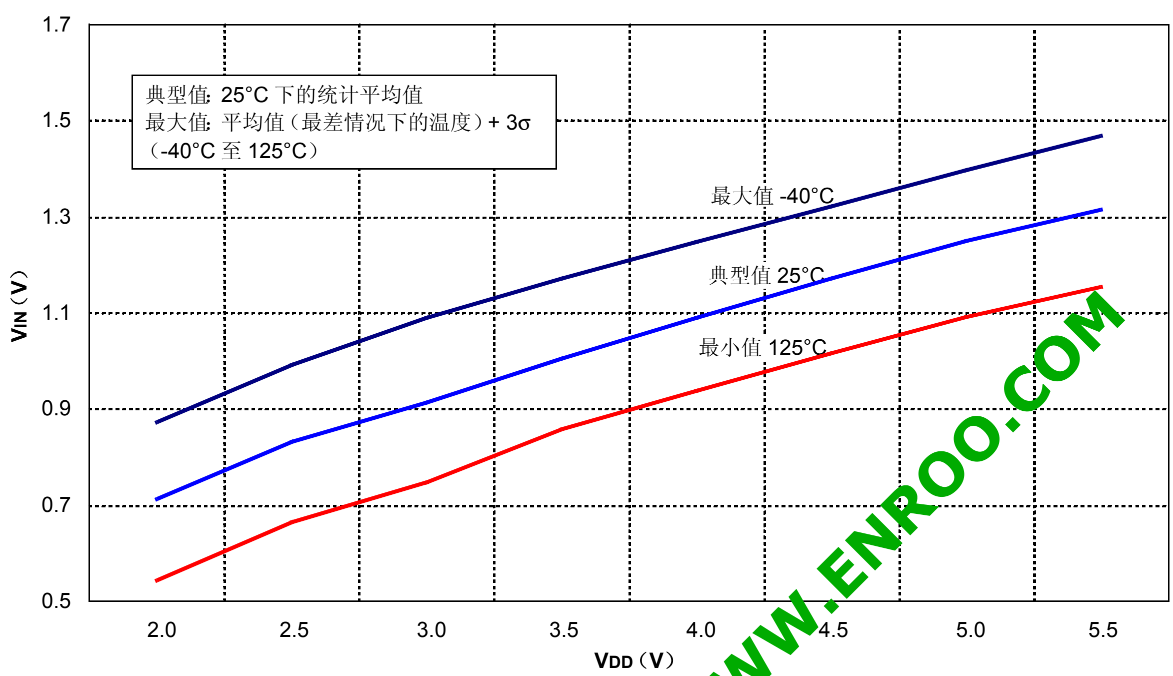


图 18-31： 不同温度的施密特触发器输入门限 VIN — VDD 曲线

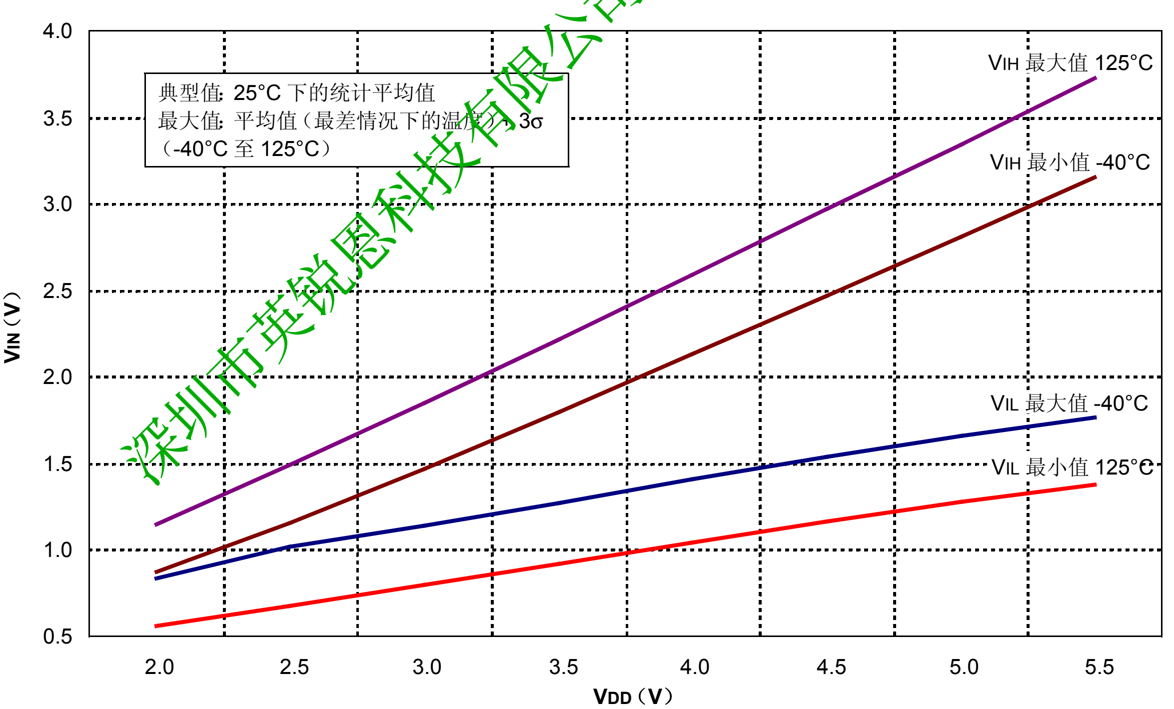


图 18-32： 比较器响应时间曲线 （上升沿）

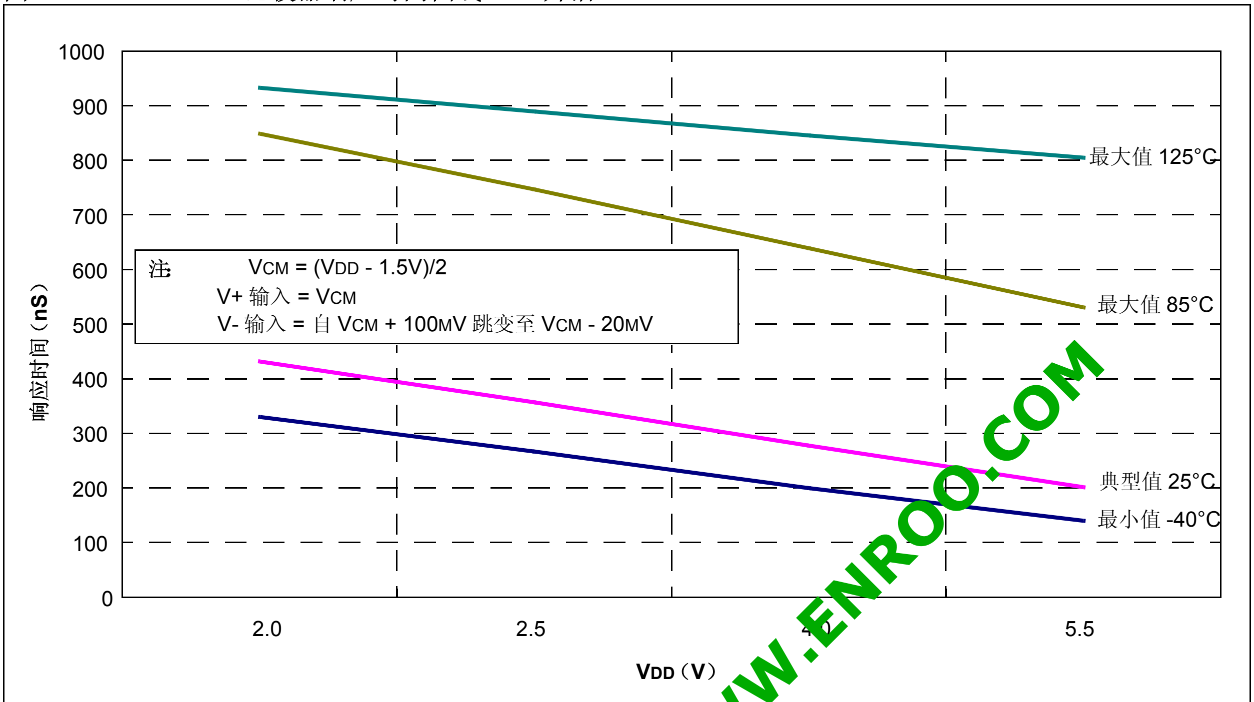


图 18-33： 比较器响应时间曲线 （下降沿）

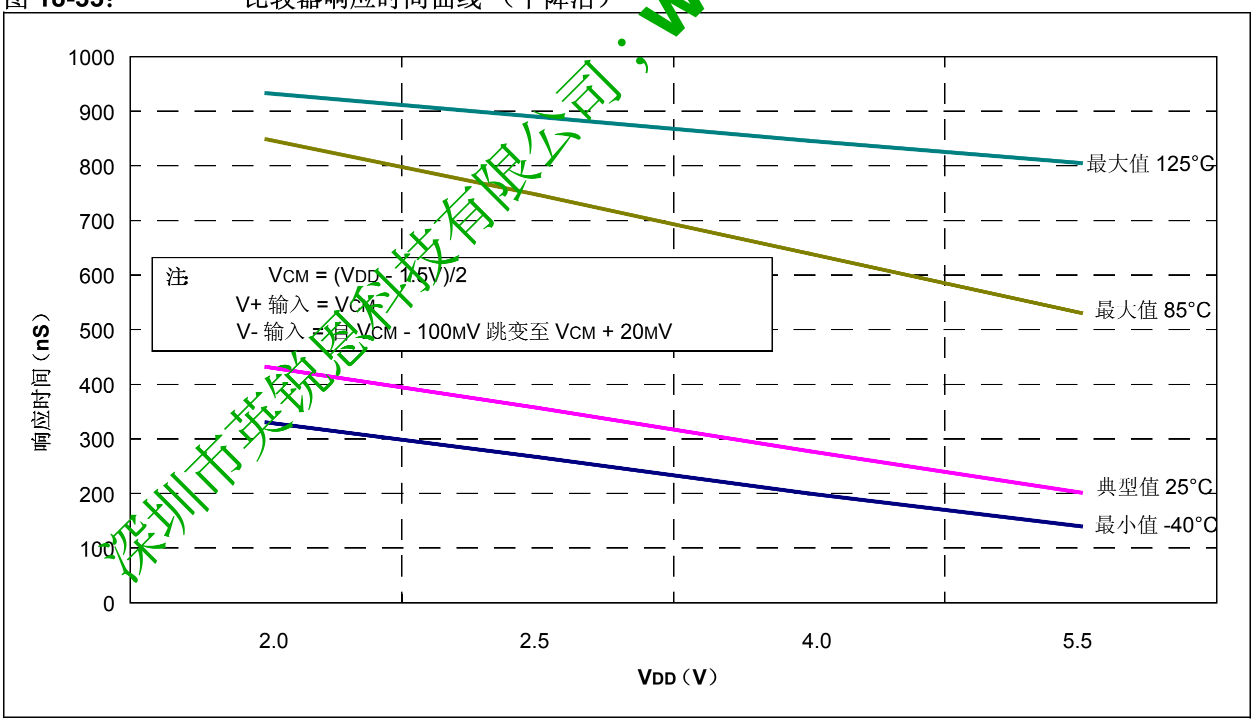


图 18-34： 不同温度的 LFINTOSC 频率— VDD 曲线 （31 kHz）

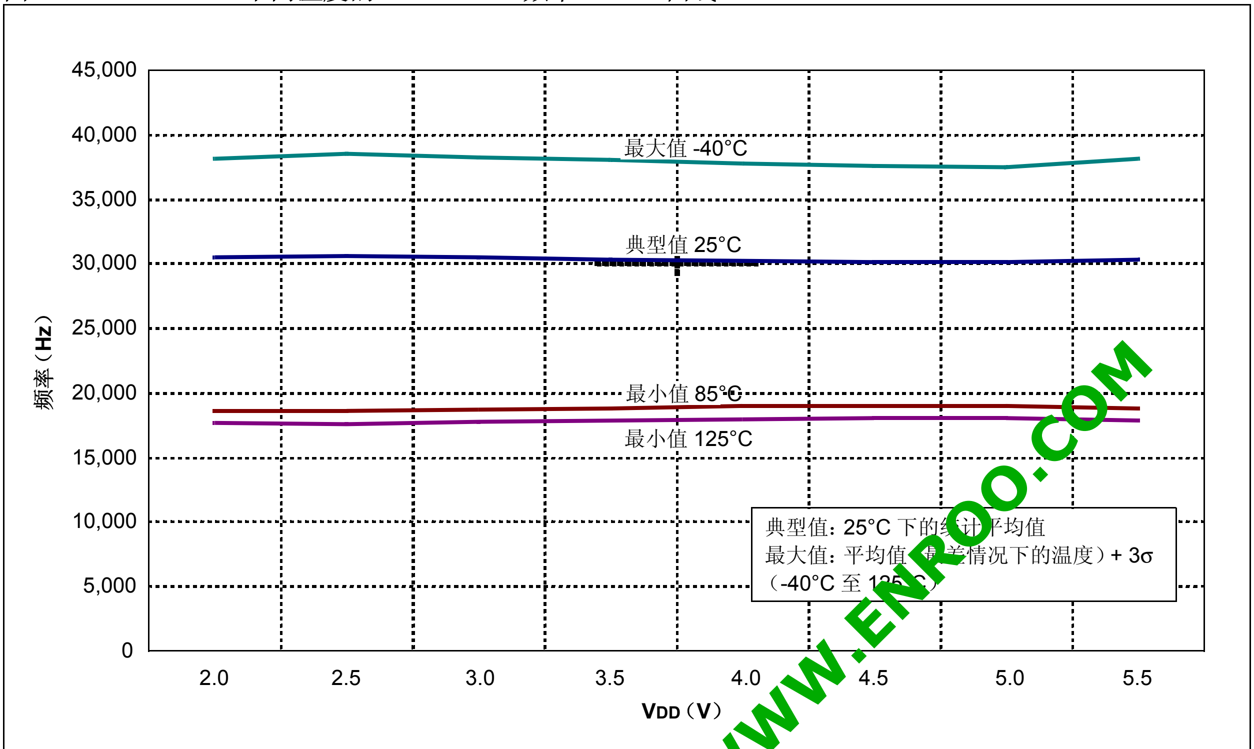


图 18-35： 不同温度的 ADC 时钟周期— VDD 曲线

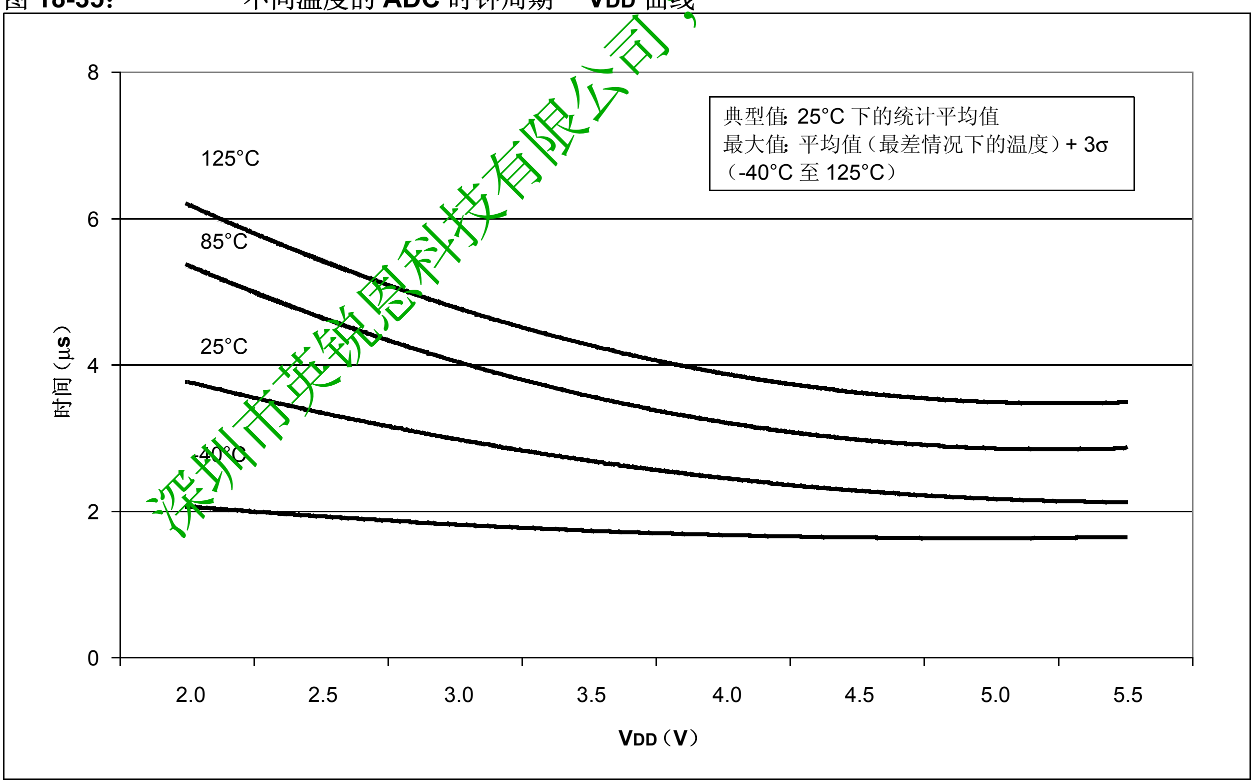


图 18-36： 不同温度的典型 HFINTOSC 起振时间— VDD 曲线

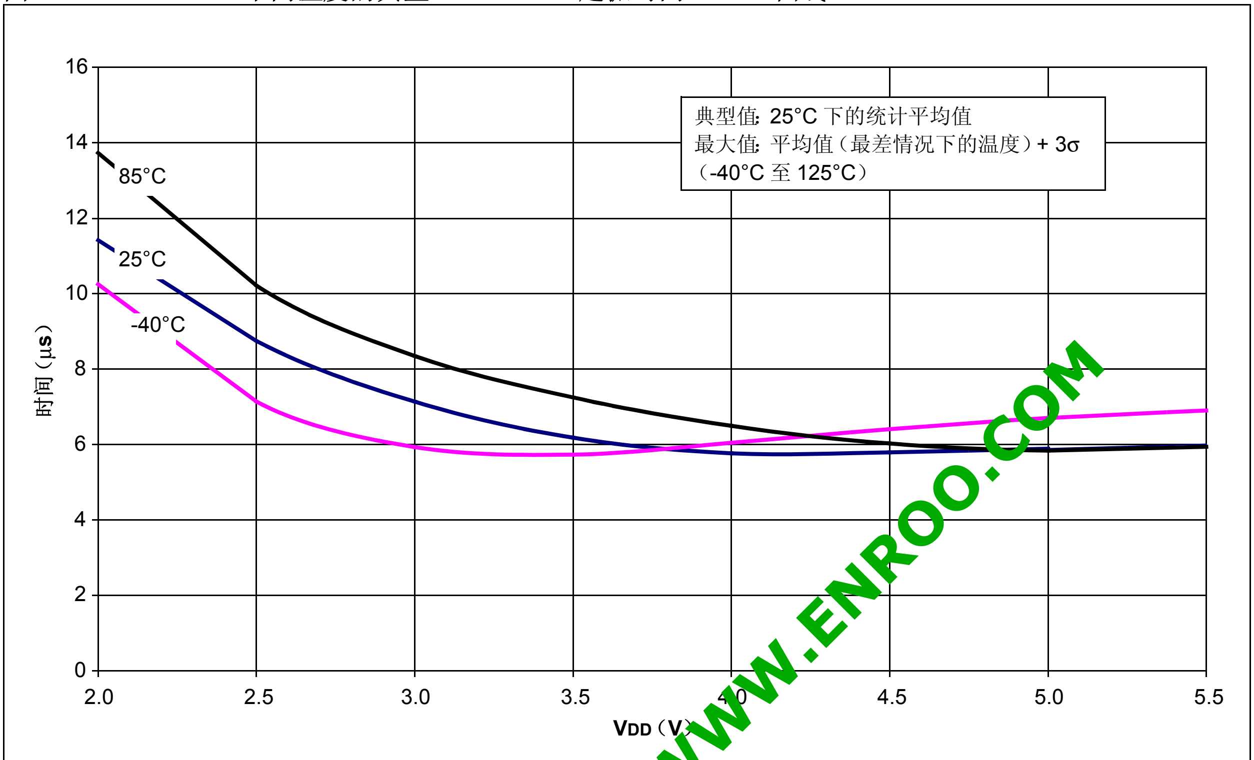


图 18-37： 不同温度的最大 HFINTOSC 起振时间— VDD 曲线

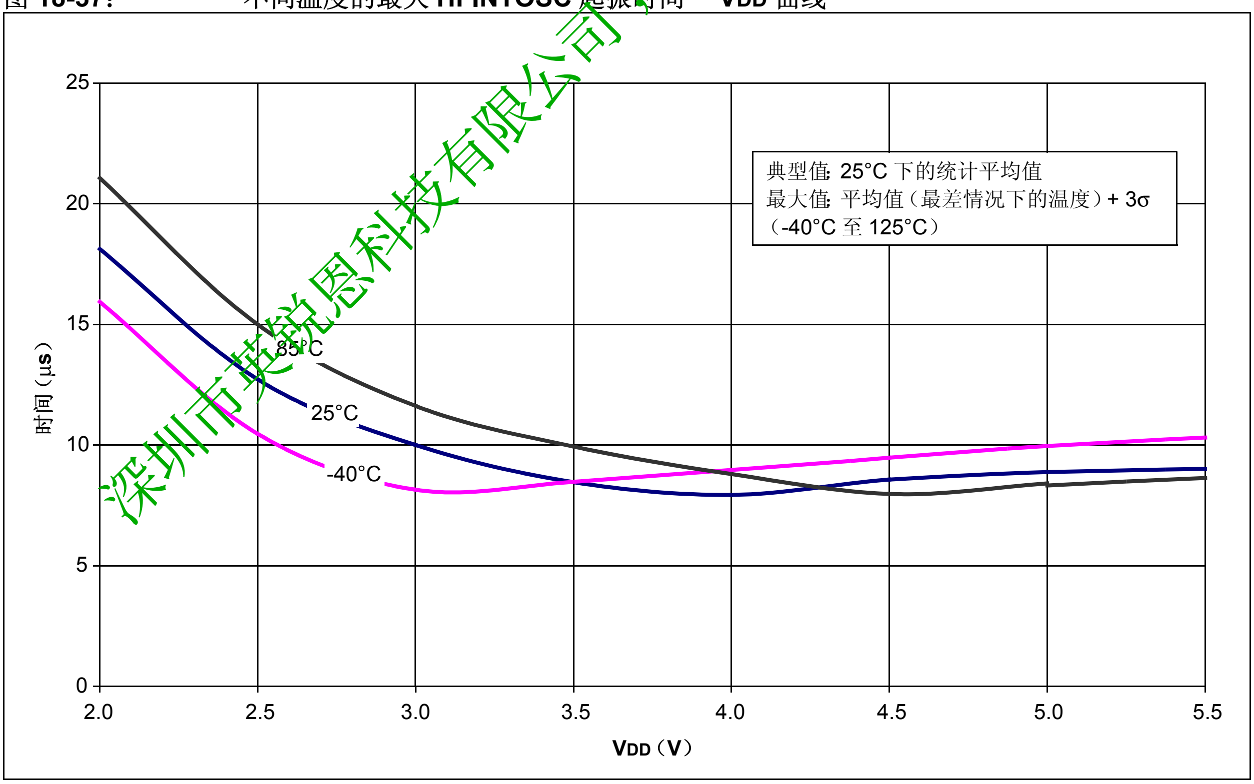


图 18-38： 不同温度的最小 HFINTOSC 起振时间— VDD 曲线

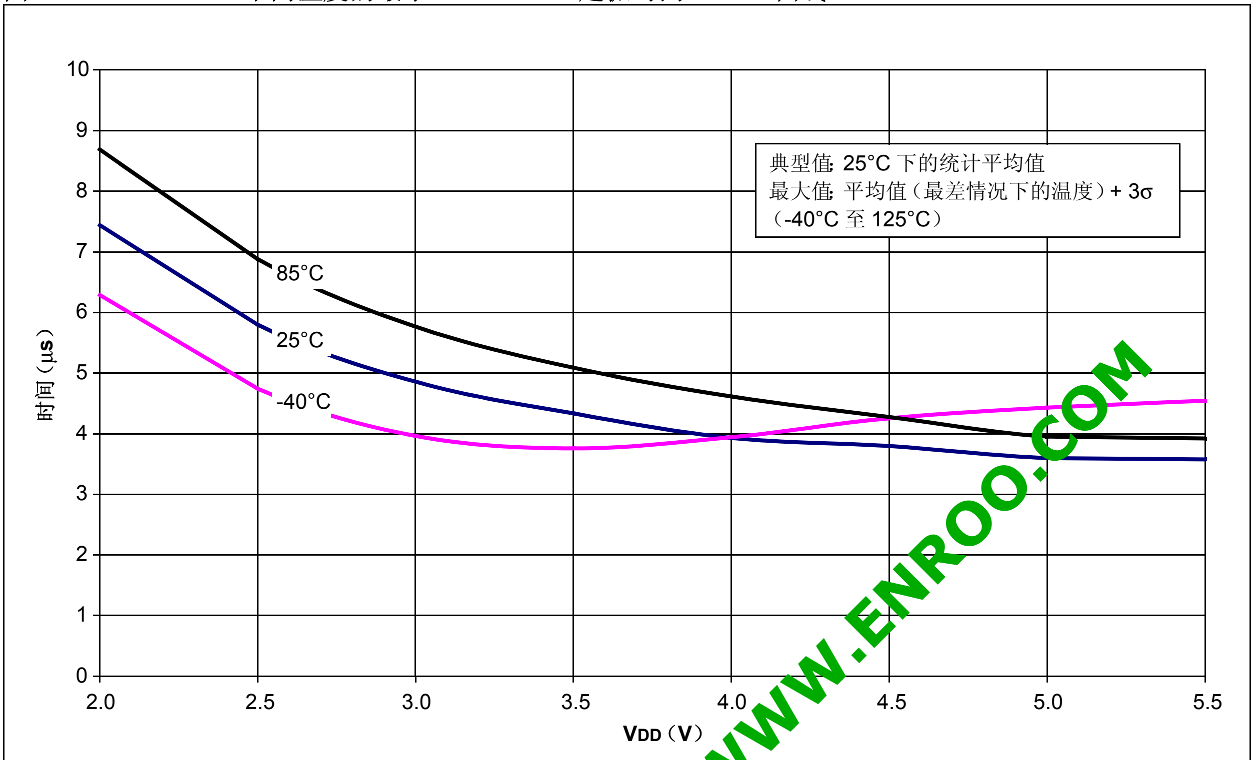


图 18-39： 典型 HFINTOSC 频率变化— VDD 曲线（25°C）

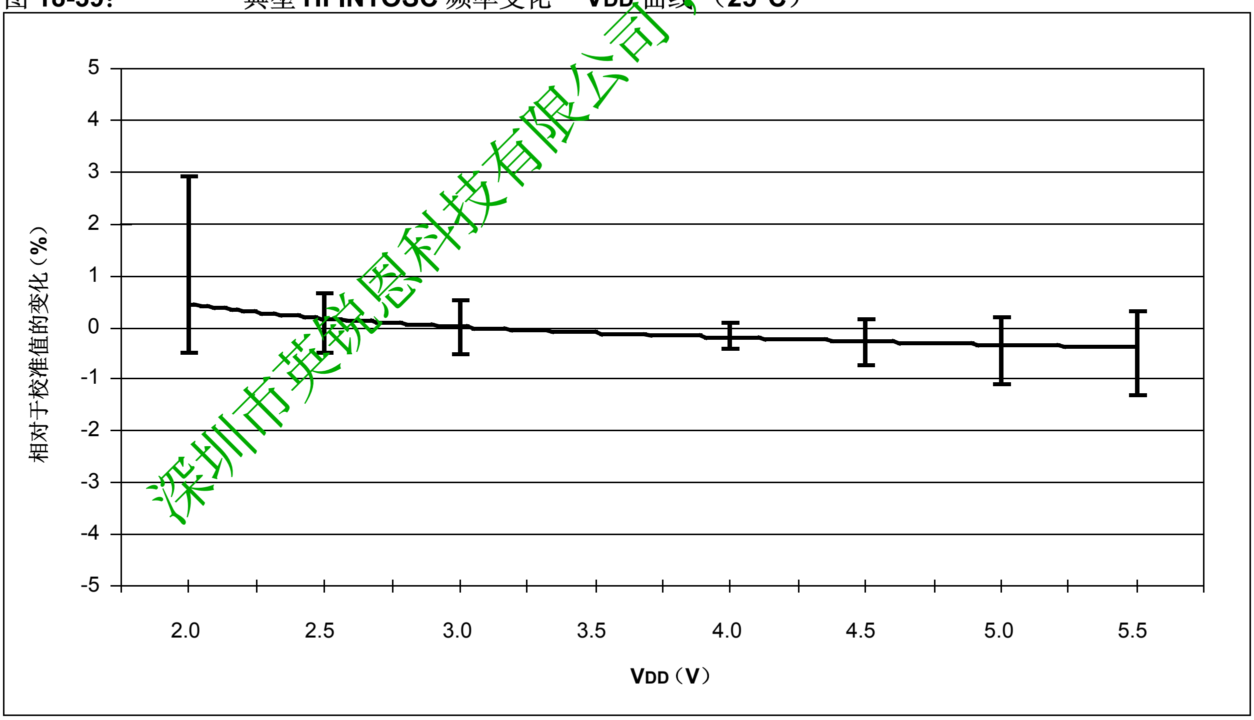


图 18-40： 典型 HFINTOSC 频率变化— VDD 曲线 （85°C）

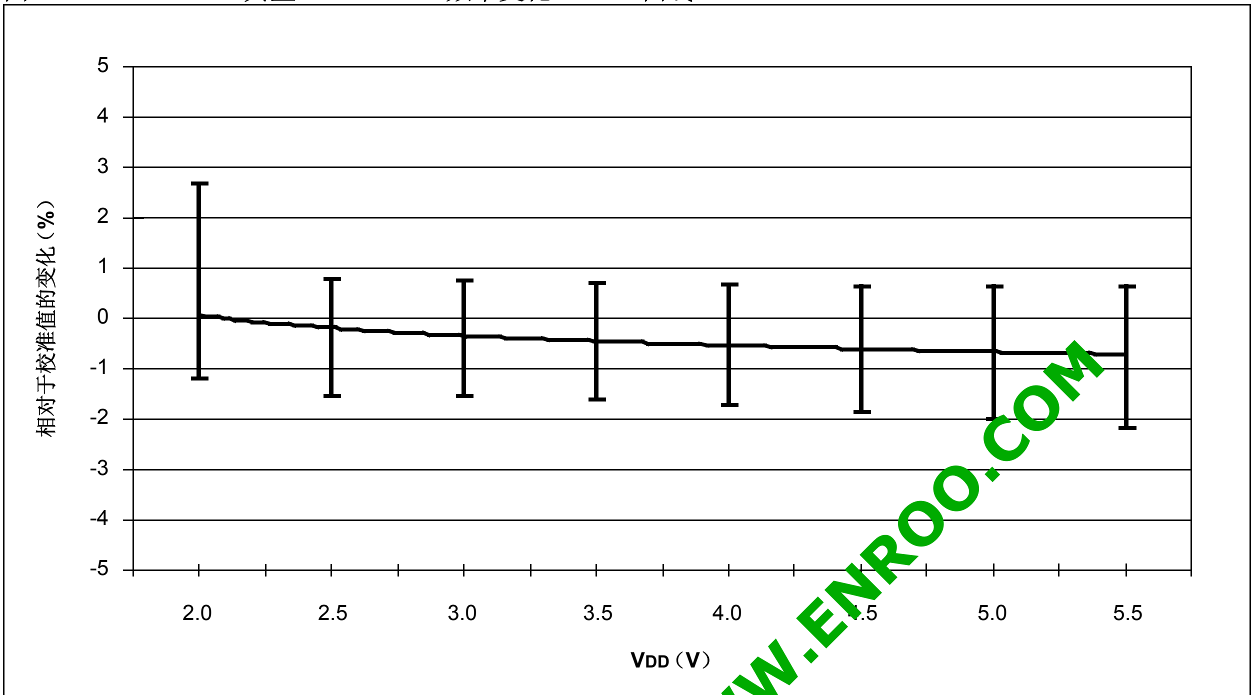


图 18-41： 典型 HFINTOSC 频率变化— VDD 曲线 （125°C）

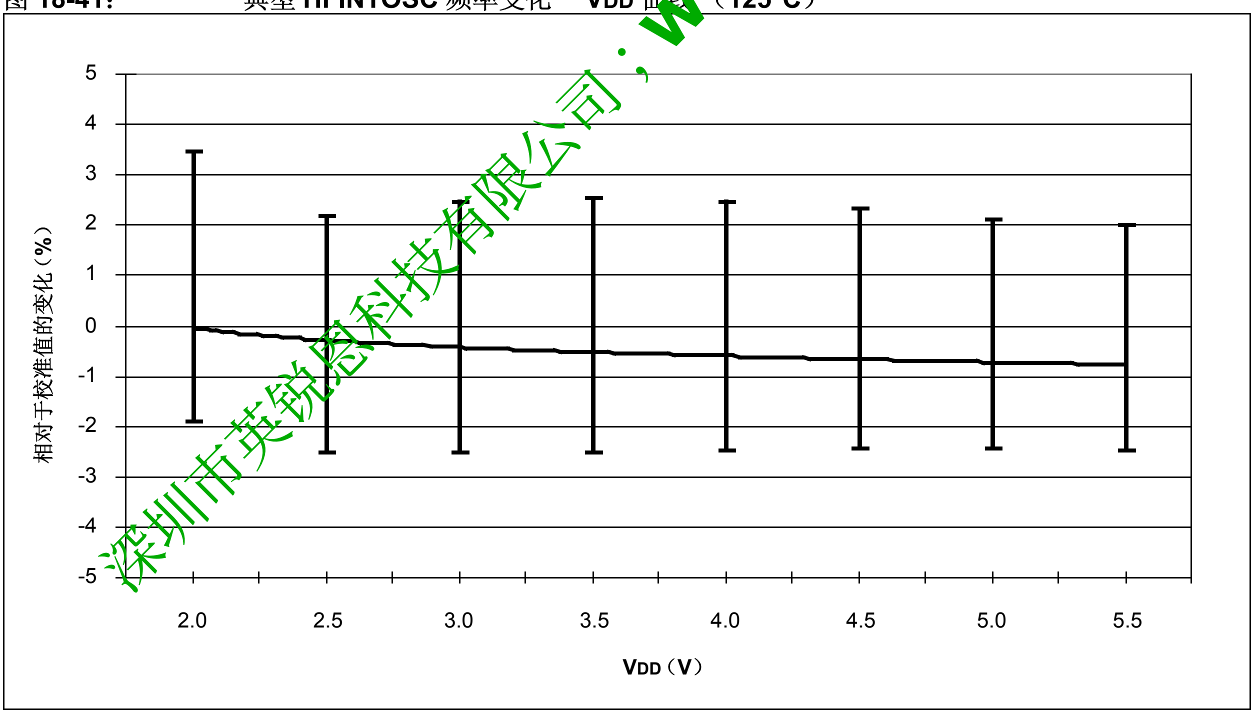


图 18-42： 典型 HFINTOSC 频率变化－ VDD 曲线 （-40°C）

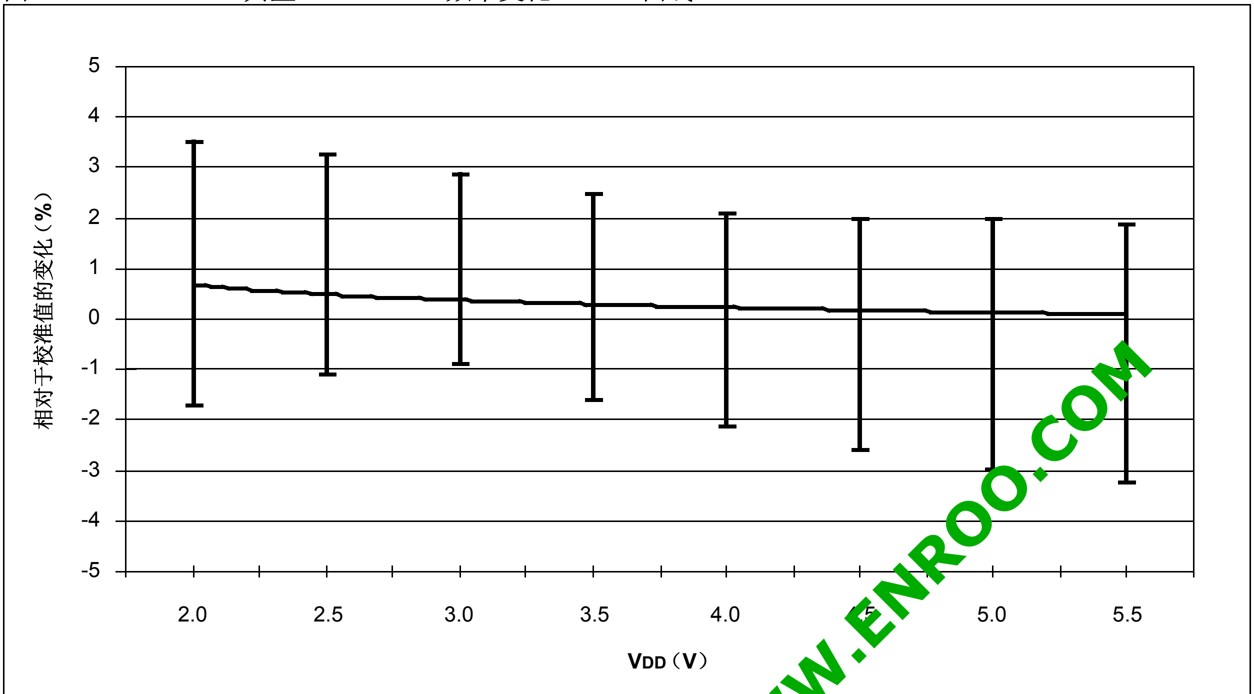


图 18-43： 典型 VP6 参考电压－ VDD 曲线 （25°C）

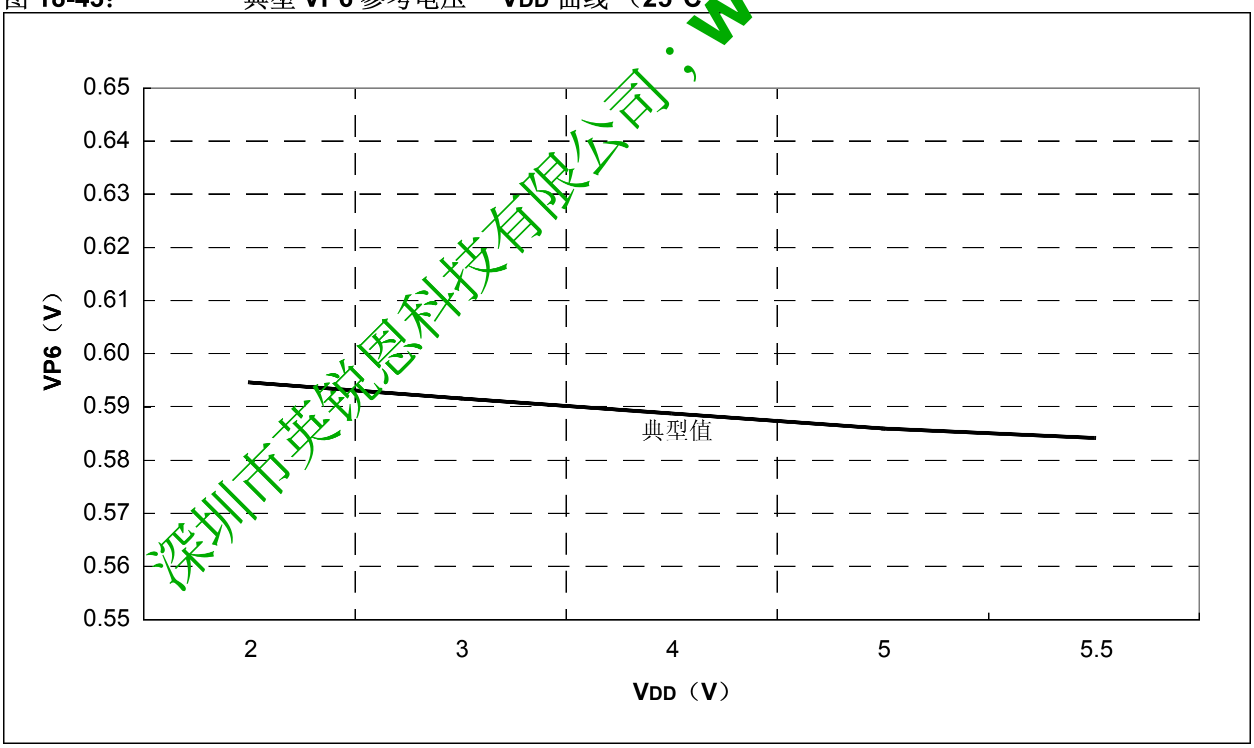


图 18-44： 不同温度的典型 VP6 参考电压 （3V）

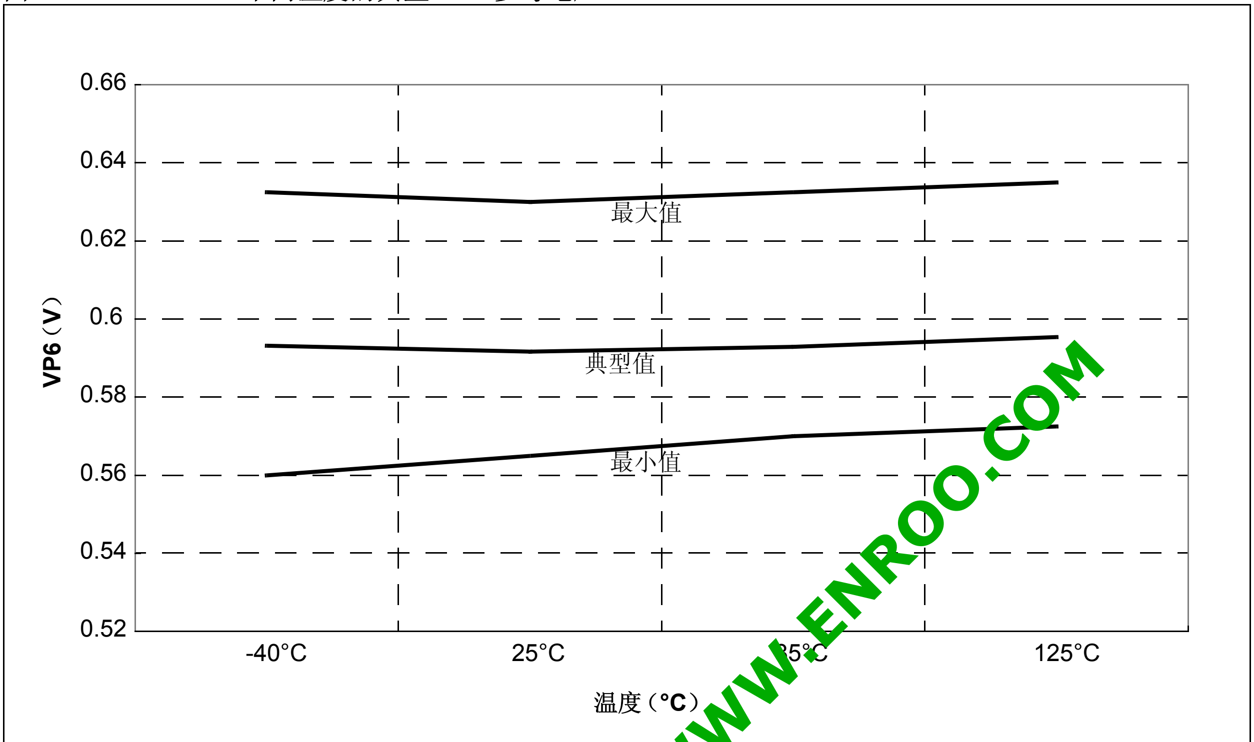


图 18-45： 不同温度的典型 VP6 参考电压 （5V）

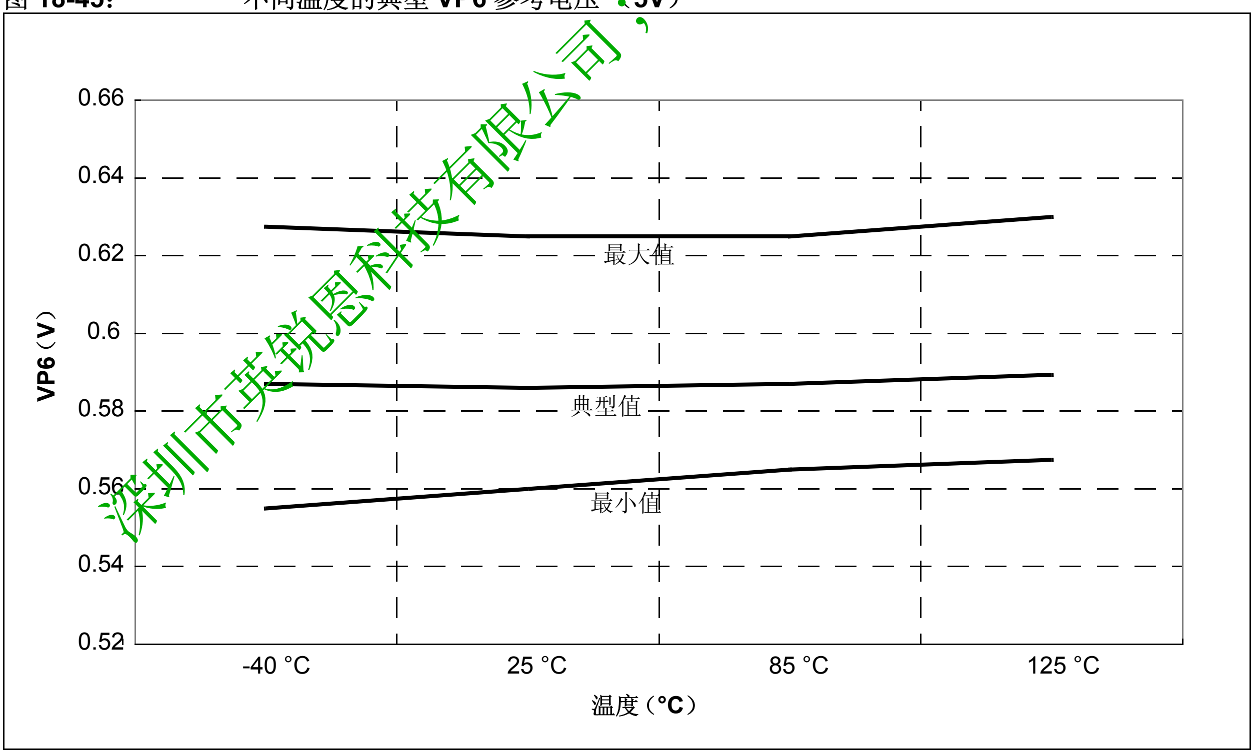


图 18-46： 典型 VP6 参考电压分布 （3V， 25°C）

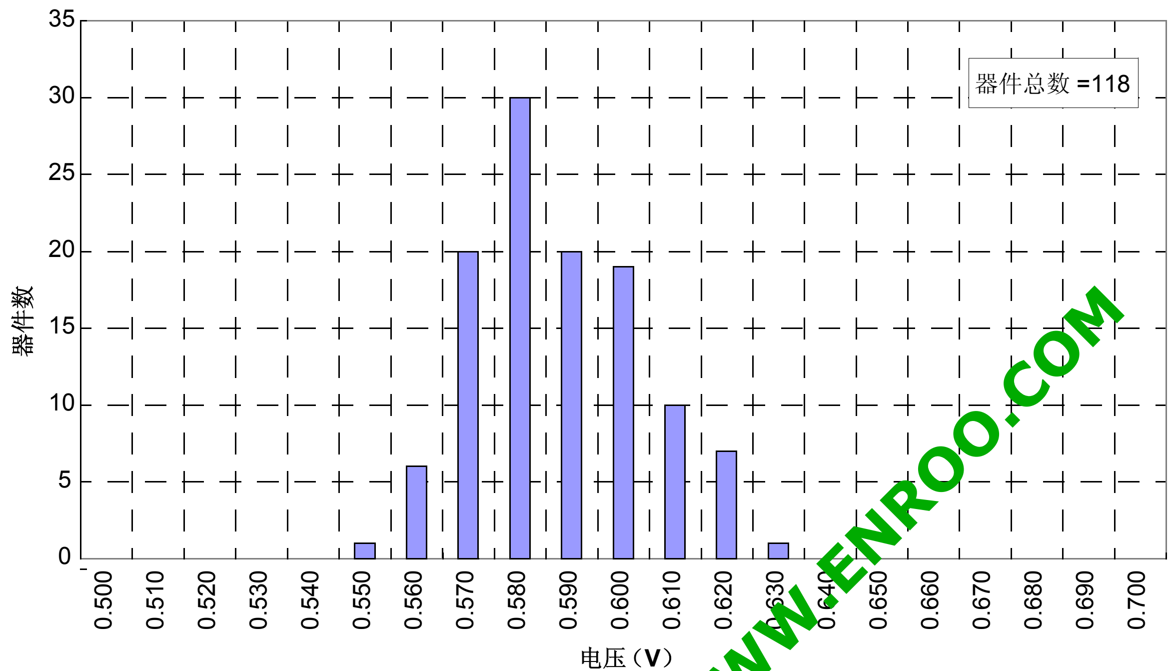


图 18-47： 典型 VP6 参考电压分布 （3V，85°C）

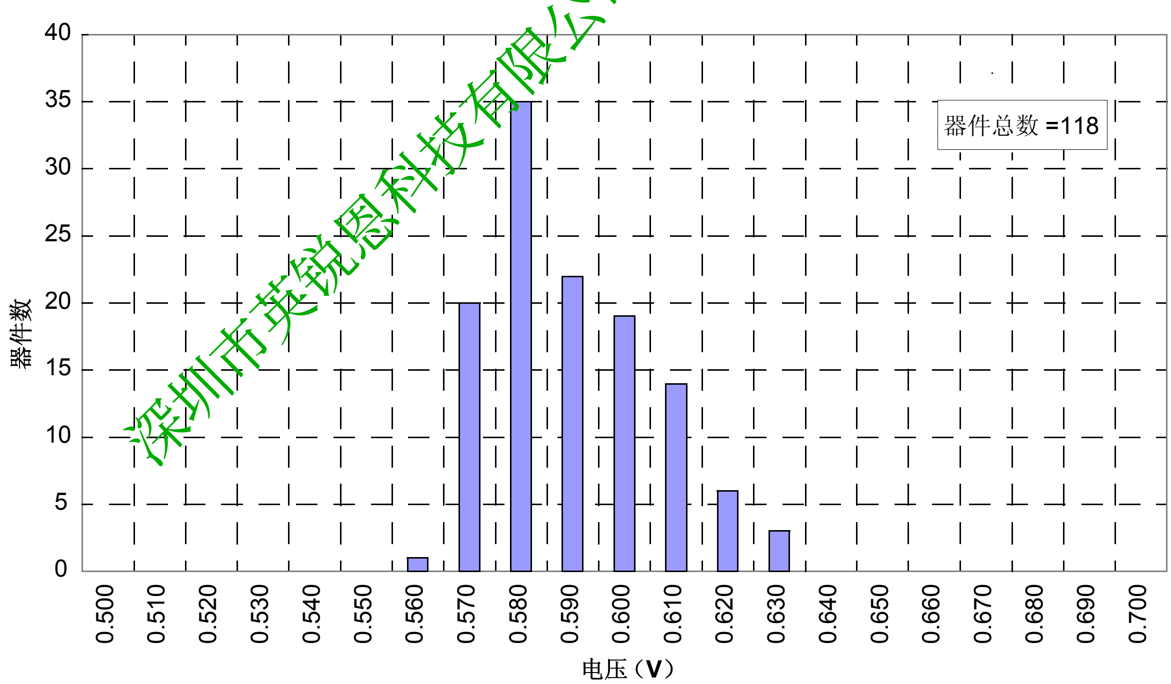


图 18-48： 典型 VP6 参考电压分布 （3V， 125°C）

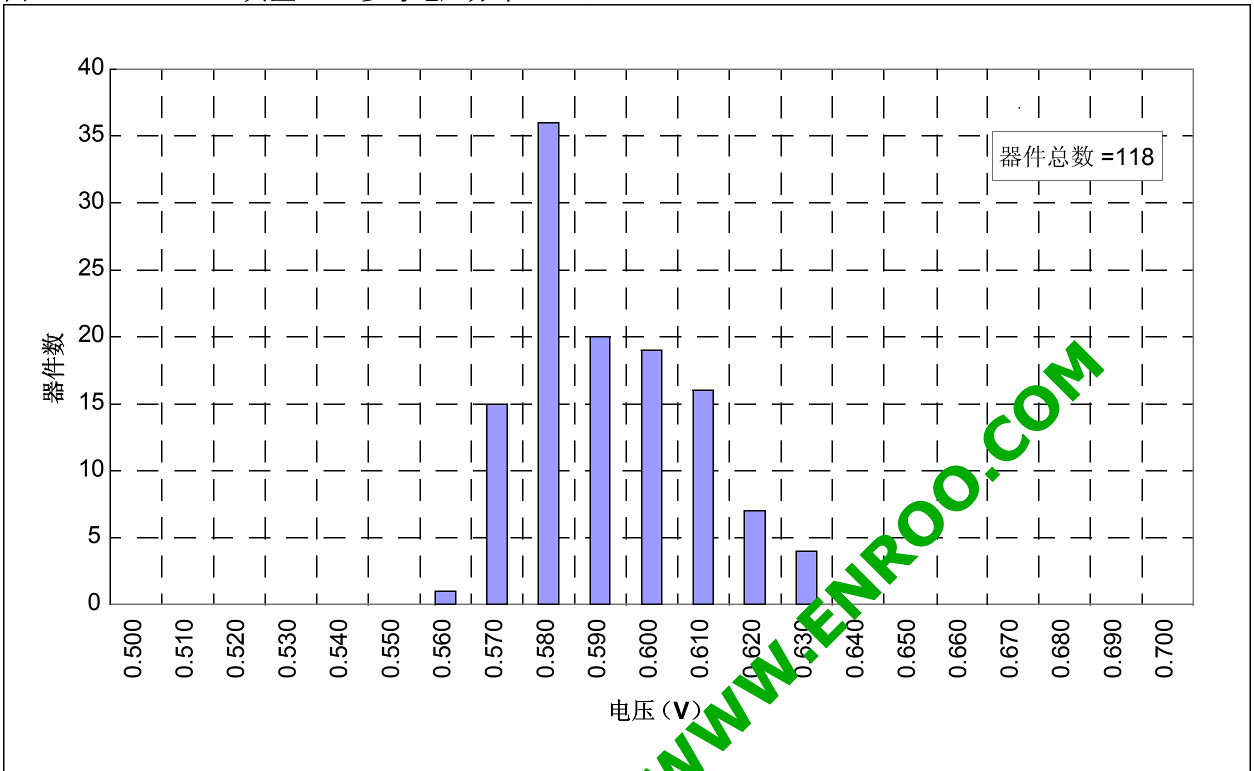


图 18-49： 典型 VP6 参考电压分布 （3V， -40°C）

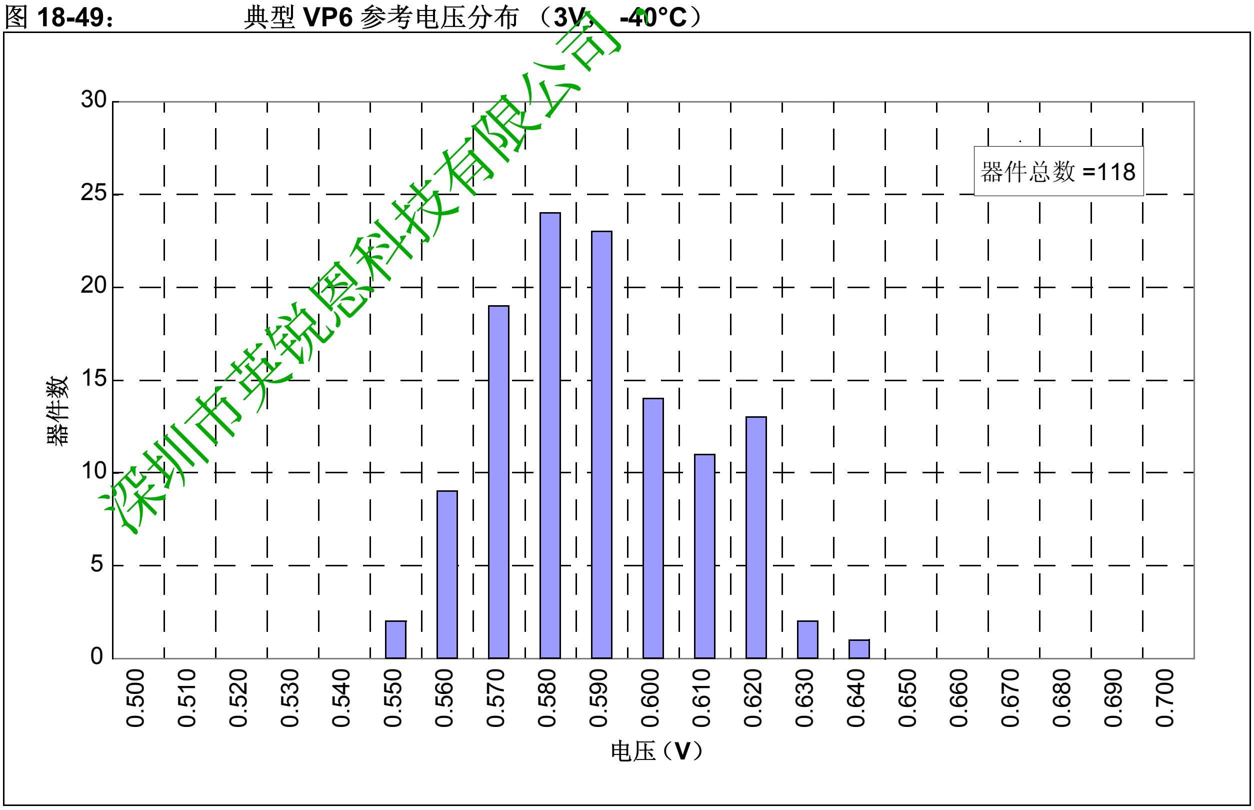


图 18-50: 典型 VP6 参考电压分布 （5V， 25°C）

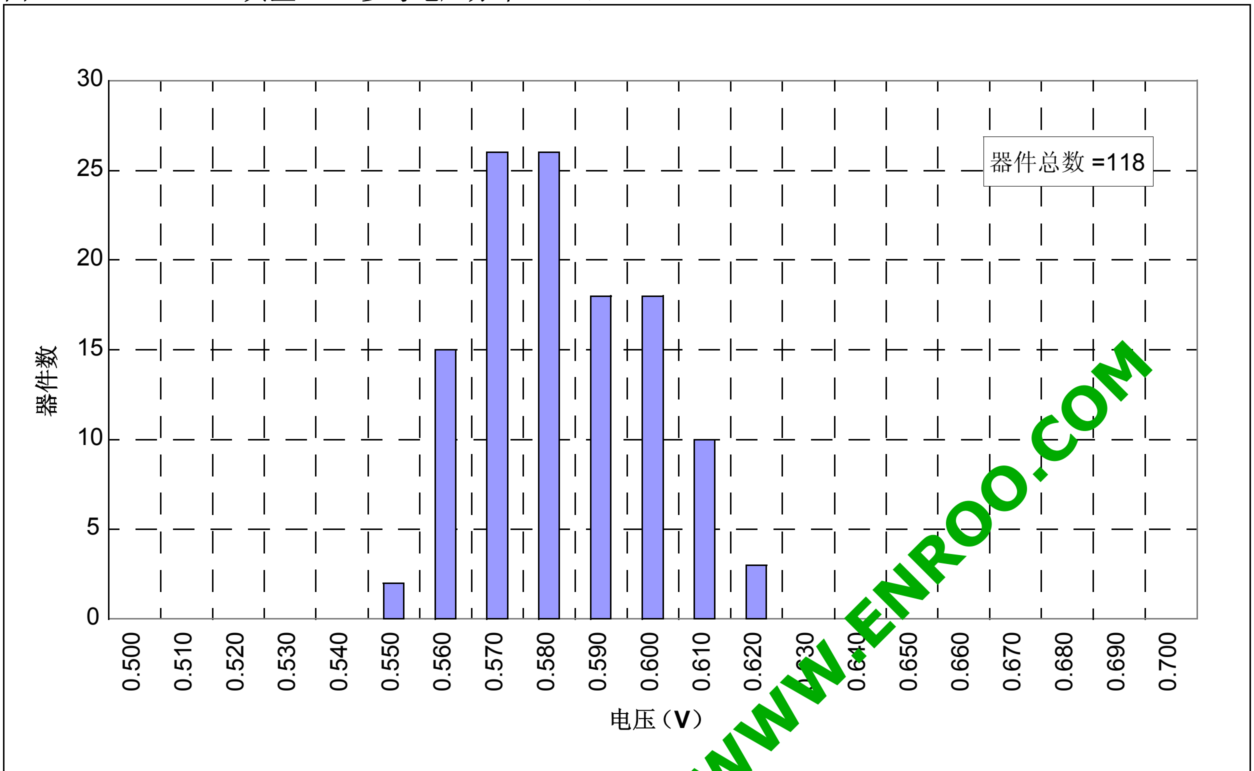


图 18-51: 典型 VP6 参考电压分布 （5V，85°C）

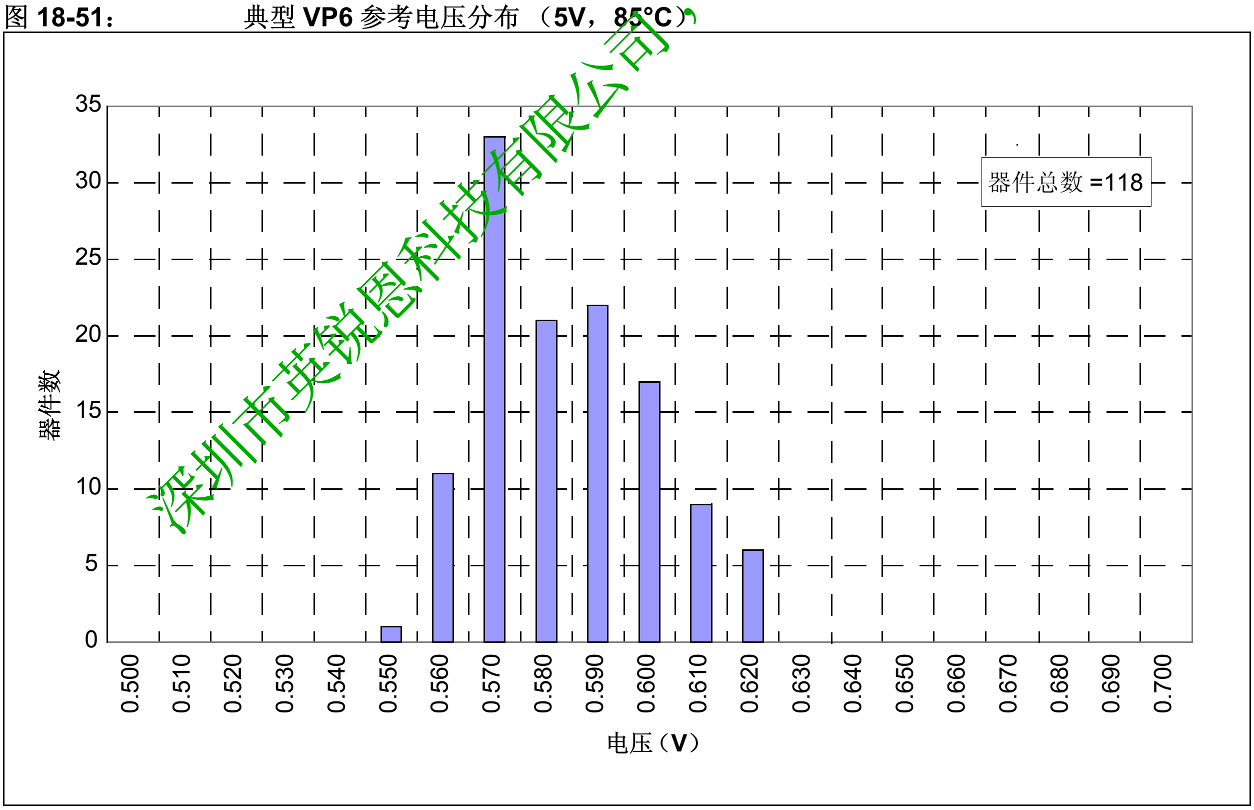


图 18-52： 典型 VP6 参考电压分布 （5V， 125°C）

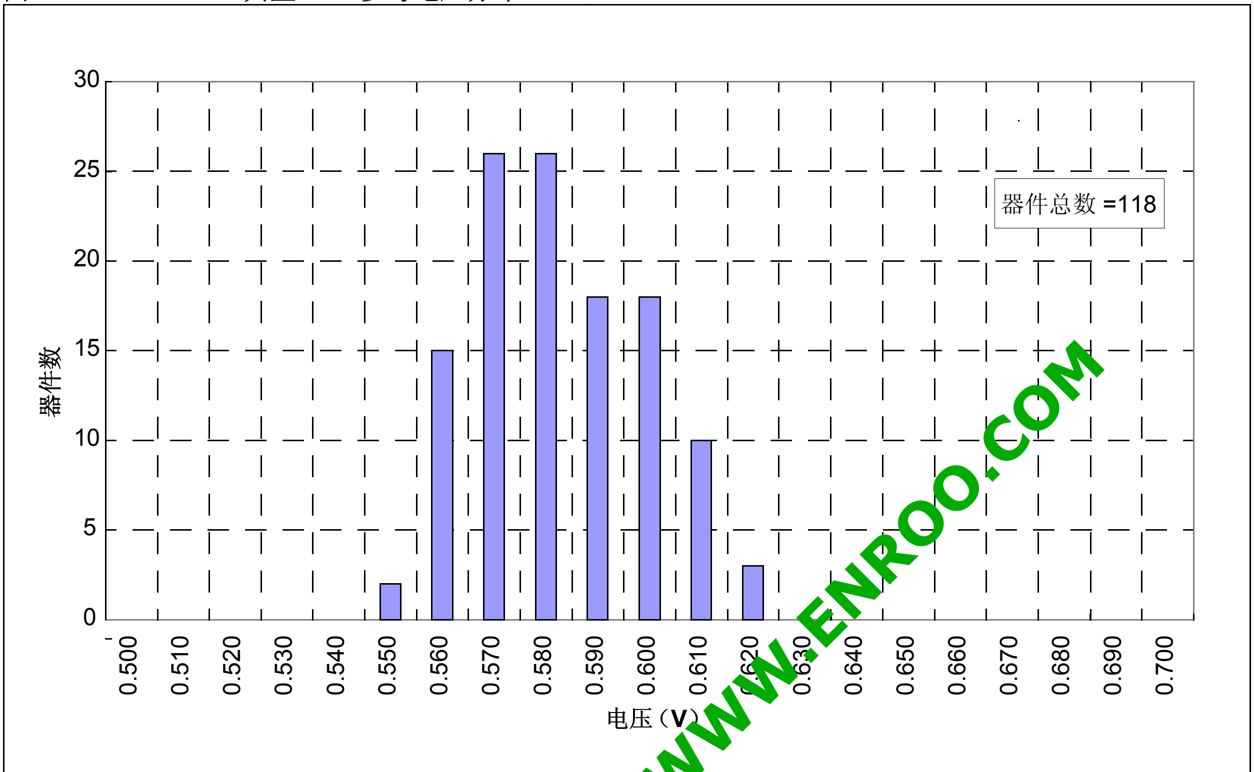
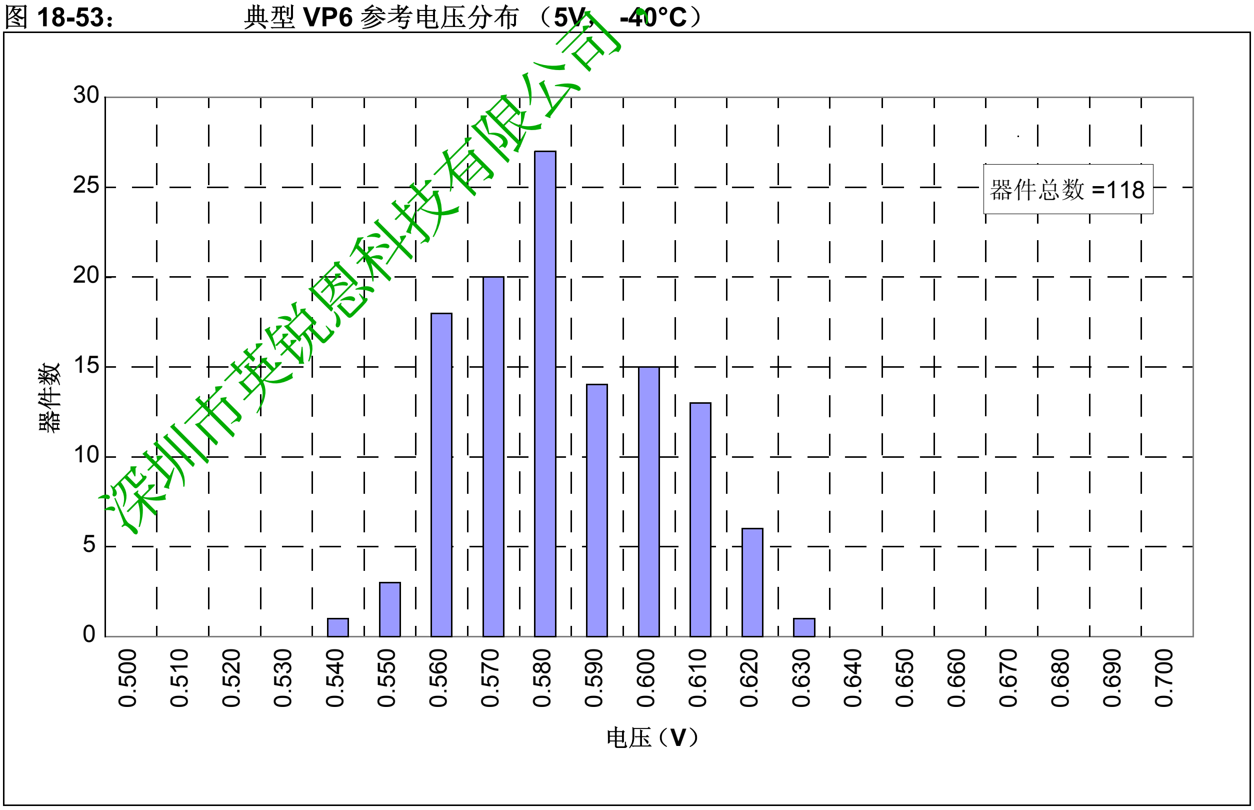


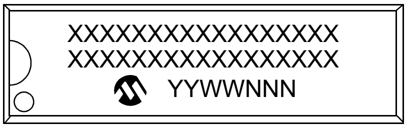
图 18-53： 典型 VP6 参考电压分布 （5V， -40°C）



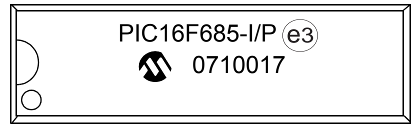
19.0 封装信息

19.1 封装标识信息

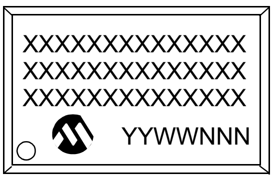
20 引脚 PDIP



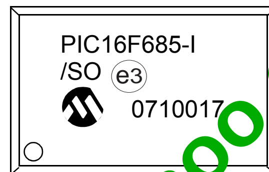
示例



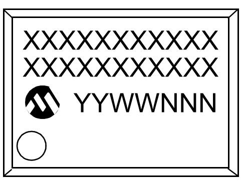
20 引脚 SOIC（7.50 mm）



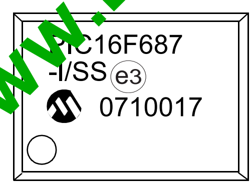
示例



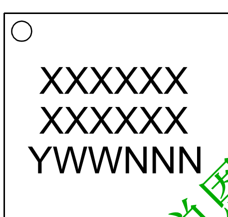
20 引脚 SSOP



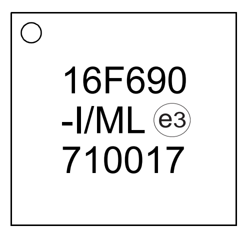
示例



20 引脚 QFN



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（1 月 1 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
	*	此为无铅封装。 JEDEC 无铅标志（e3）标示于此种封装的外包装上。

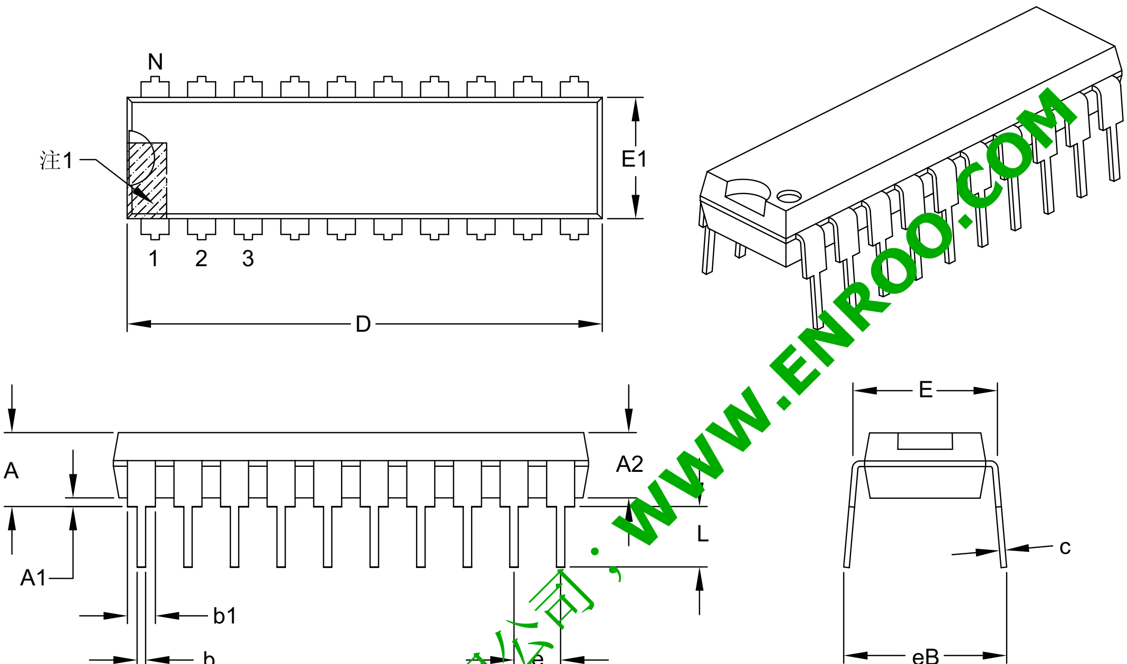
注： Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

19.2 封装详细信息

以下部分将介绍各种封装的技术细节。

20引脚塑封双列直插式封装（P）——300 mil主体 [PDIP]

注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



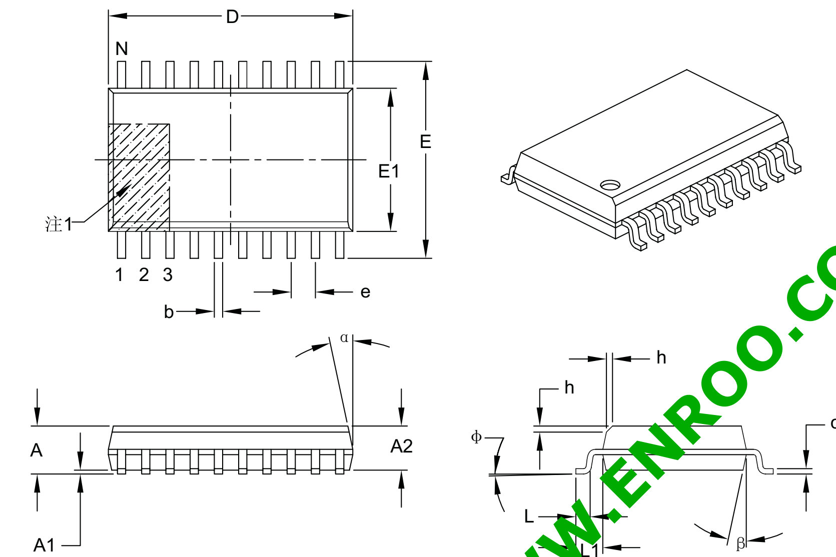
	单位	英寸		
	尺寸范围	最小	正常	最大
引脚数	N	20		
引脚间距	e	.100 BSC		
顶端到固定面高度	A	–	–	.210
塑模封装厚度	A2	.115	.130	.195
塑模底面到固定面高度	A1	.015	–	–
肩到肩宽度	E	.300	.310	.325
塑模封装宽度	E1	.240	.250	.280
总长度	D	.980	1.030	1.060
引脚尖到固定面高度	L	.115	.130	.150
引脚厚度	c	.008	.010	.015
引脚上部宽度	b1	.045	.060	.070
引脚下部宽度	b	.014	.018	.022
总排列间距 §	eB	–	–	.430

注：
1. 引脚1的可见定位标记可不同，但必须位于阴影区域内。
2. § 重要特性。
3. 尺寸D和E1不包括塑模毛边或突起。 塑模每侧的毛边或突起不得超过0.010英寸。
4. 尺寸和公差遵循ASME Y14.5M。
   BSC：基本尺寸。显示的是没有公差的理论精确值。

Microchip Technology 图号C04-019B

20引脚塑封小外形封装（SO）——宽条，7.50 mm主体 [SOIC]

注：　最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



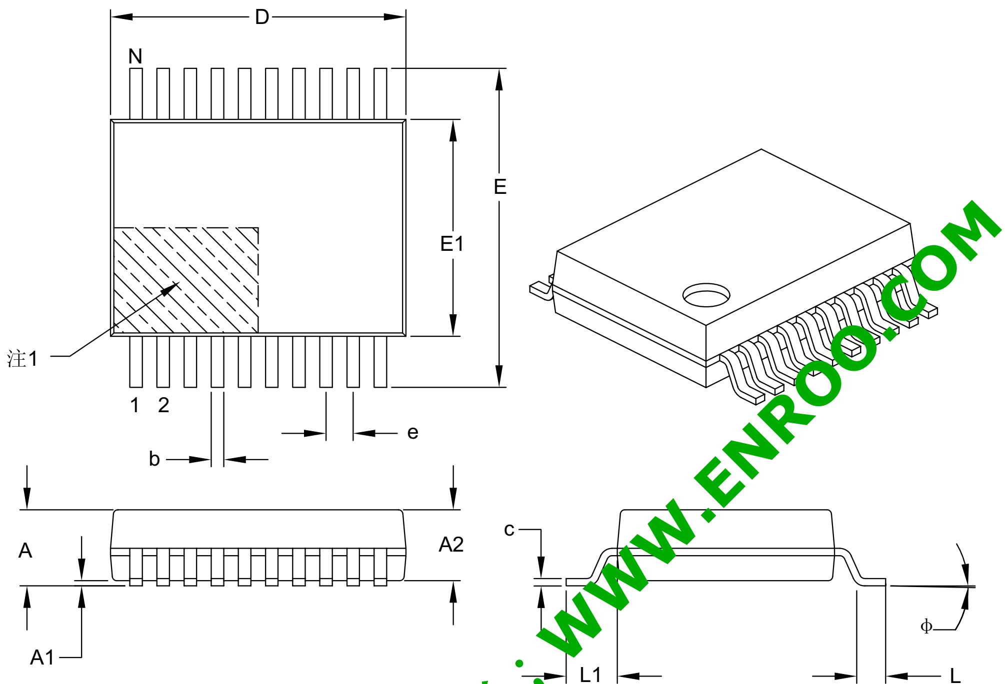
单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		20	
引脚间距	e		1.27 BSC	
总高度	A	–	–	2.65
塑模封装厚度	A2	2.05	–	–
悬空间隙 §	A1	0.10	–	0.30
总宽度	E		10.30 BSC	
塑模封装宽度	E1		7.50 BSC	
总长度	D		12.80 BSC	
斜面（可选）	h	0.25	–	0.75
底脚长度	L	0.40	–	1.27
引脚投影长度	L1		1.40 REF	
底脚倾斜角	ϕ	0°	–	8°
引脚厚度	c	0.20	–	0.33
引脚宽度	b	0.31	–	0.51
塑模顶部锥度	α	5°	–	15°
塑模底部锥度	β	5°	–	15°

注：
1. 引脚1的可见定位标记可能不同，但必须位于阴影区域内。
2. § 重要特性。
3. 尺寸D和E1不包括塑模毛边或突起。 塑模每侧的毛边或突起不得超过0.15毫米。
4. 尺寸和公差遵循ASME Y14.5M。
   BSC：基本尺寸。显示的是没有公差的理论精确值。
   REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-094B

20引脚塑封缩小型小外形封装（**SS**）**——5.30 mm**主体 **[SSOP]**

注：　最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



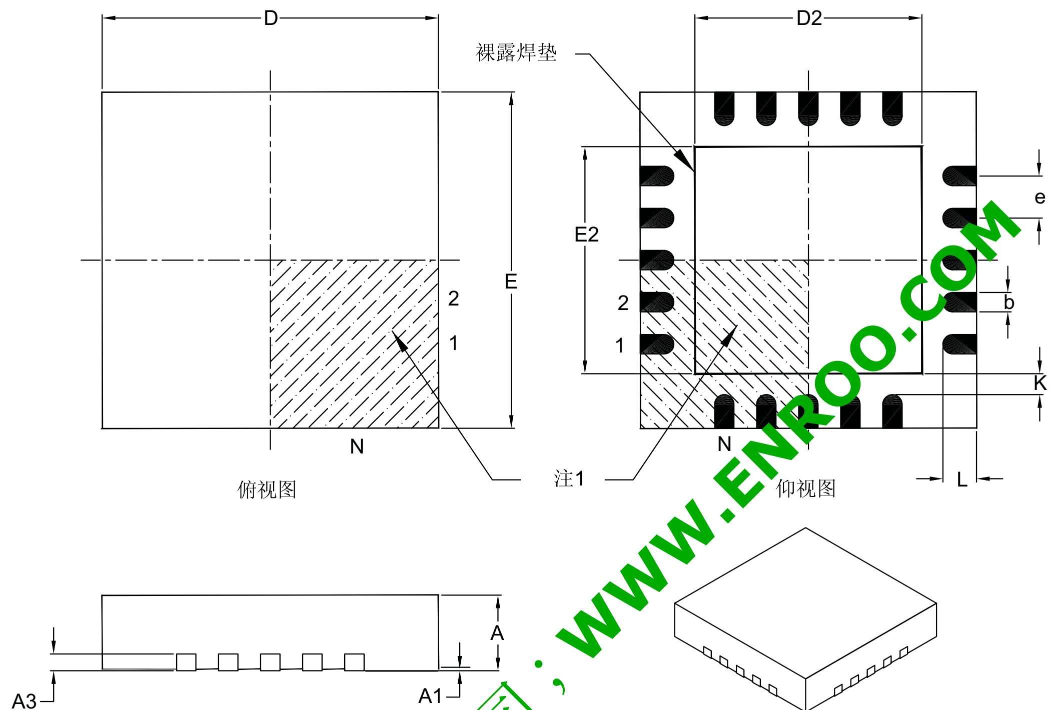
单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		20	
引脚间距	e		0.65 BSC	
总高度	A	–	–	2.00
塑模封装厚度	A2	1.65	1.75	1.85
悬空间隙	A1	0.05	–	–
总宽度	E	7.40	7.80	8.20
塑模封装宽度	E1	5.00	5.30	5.60
总长度	D	6.90	7.20	7.50
底脚长度	L	0.55	0.75	0.95
引脚投影长度	L1		1.25 REF	
引脚厚度	c	0.09	–	0.25
底脚倾斜角	φ	0°	4°	8°
引脚宽度	b	0.22	–	0.38

注：
1. 引脚1的可见定位标记可能不同，但必须位于阴影区域内。
2. 尺寸D和E1不包括塑模毛边或突起。 塑模每侧的毛边或突起不得超过0.20毫米。
3. 尺寸和公差请遵循ASME Y14.5M。
　BSC：基本尺寸。显示的是没有公差的理论精确值。
　REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-072B

20引脚塑封正方扁平无脚封装（ML）——4x4x0.9 mm主体 [QFN]

注：	最新封装图请至http://www.microchip.com/packaging查看Microchip封装规范。



单位		毫米		
尺寸范围		最小	正常	最大
引脚数	N		20	
引脚间距	e		0.50 BSC	
总高度	A	0.80	0.90	1.00
悬空间隙	A1	0.00	0.02	0.05
触点厚度	A3		0.20 REF	
总宽度	E		4.00 BSC	
裸露金属焊垫宽度	E2	2.60	2.70	2.80
总长度	D		4.00 BSC	
裸露金属焊垫长度	D2	2.60	2.70	2.80
触点宽度	b	0.18	0.25	0.30
触点长度	L	0.30	0.40	0.50
触点到裸露金属焊垫的距离	K	0.20	–	–

注：
1. 引脚1的可见定位标记可能变化，但必须位于阴影区域内。
2. 封装为切割分离。
3. 尺寸和公差遵循ASME Y14.5M。

   BSC：基本尺寸。显示的是没有公差的理论精确值。

   REF：参考尺寸，通常无公差，仅供参考。

Microchip Technology 图号C04-126B

附录 A： 数据手册版本历史

版本 A （2005 年 3 月）

这是新的数据手册。

版本 B （2006 年 5 月）

增加了 631/677 器件编号；在引脚图后面增加了引脚汇总表；使用了固定章节。

版本 C （2006 年 7 月）

修改了第 4.2.1 节， ANSEL 和 ANSELH 寄存器；寄存器4-3，ANSEL 模拟选择；增加了寄存器4-4，ANSELH 模拟选择高位；第 11.3.2 节，将 CCP1<1:0> 修改为 DC1B<1:0> ；第 11.3.7 节，第 4 项——将 CCP1 修改为 DC1B ；图 11-5，将 CCP1 修改为 DC1B ；表 11-4，将 P1M 修改为 P1M<1:0> ；第 12.3.1 节，修改了第 3 段；修改了注 2 ；修改了图 12-6 的标题。

版本 D （2007 年 2 月）

删除了初稿状态；将 PICmicro 更改为 PIC ；更换了开发工具章节；更换了封装图。

版本 E （2008 年 3 月）

添加了特性数据图；更新了 EUSART 一章；更新了电气规范部分，并根据需要更新了封装图。

附录 B： 从其他 PIC® 器件移植

本节讨论从其他 PIC 器件移植到 PIC16F6XX 系列器件的某些问题。

B.1 PIC16F676 到 PIC16F685

表 B-1： 功能比较

功能	PIC16F676	PIC16F685
最大工作速度	20 MHz	20 MHz
最大程序存储器（字数）	1024	4096
SRAM （字节数）	64	128
A/D 分辨率	10 位	10 位
数据 EEPROM（字节数）	128	256
定时器 （8/16 位）	1/1	2/1
振荡器模式	8	8
欠压复位	有	有
内部上拉	RA0/1/2/4/5	RA0/1/2/4/5， MCLR
电平变化中断	RA0/1/2/3/4/5	RA0/1/2/3/4/5
比较器	1	2
ECCP+	无	有
超低功耗唤醒	无	有
扩展型 WDT	无	有
WDT/BOR 的软件控制选项	无	有
INTOSC 频率	4 MHz	31 kHz-8 MHz
时钟切换	无	有

> 注： 该器件设计为按照数据手册上的参数来工作。它已通过电气规范测试，该规范用于确定器件是否符合这些参数。由于存在器件生产工艺上的差异，其工作特性可能与其早期版本的产品有所不同。这些不同可能导致该器件与其早期版本在应用中的性能差异。

索引

Y

Z

MICROCHIP 网站

Microchip 网站 （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、应用笔记和示例程序、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 的客户通知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，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://support.microchip.com获得网上技术支持。

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： PIC16F631/677/685/687/689/690　　　　文献编号： DS41262E_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。

器件编号	X	/XX	XXX
器件	温度范围	封装	模式

器件：	PIC16F631[1]、 PIC16F677[1]、 PIC16F685[1]、 PIC16F687[1]、 PIC16F689[1]、 PIC16F690[1]； V_{DD} 范围为 2.0V 至 5.5V
温度范围：	I = -40°C 至 +85°C （工业级） E = -40°C 至 +125°C （扩展级）
封装：	ML = QFN （正方扁平，无引脚） P = PDIP SO = SOIC SS = SSOP
模式：	QTP、 SQTP、代码或特殊要求 （其他情况空白）

示例：

a) PIC16F685 - I/ML 301 = 工业级温度， QFN 封装， QTP 模板 #301。

b) PIC16F689 - I/SO = 工业级温度，SOIC 封装。

c) PIC16F690T - E/SS = 扩展级温度， SSOP 封装。

注 1： T = 卷带式，仅限 SSOP、 SOIC 和 QFN 封装。

ENROO 英锐恩

单片机集成方案全方位解决服务商

优质智能电子产品"芯"方案解决商

联系信息：

深圳市英锐恩科技有限公司
ENROO-TECH（SHENZHEN）CO.,LTD
中国·深圳市福田区福华路嘉汇新城汇商中心27楼2701室

Enroo-Tech Technologies CO., Limited
香港荃灣青山道 388 號中國染廠大廈 17 樓 A 室

联系电话：86-755-82543411,83167411,83283911,88845951
联系传真：86-755-82543511
全国热线：4007-888-234
联系邮件：enroo@enroo.com
公司网站：http://www.enroo.com　　http://www.enroo-tech.com