

简介

本编程规范介绍适用于 PIC16F13145 系列单片机的一种基于 SPI 的编程方法。[编程算法](#)一章介绍该方法中使用的编程命令、编程算法和电气规范。[附录 B](#) 包含该系列单片机的器件编号、器件标识、引脚分配和封装信息，以及配置字。

 **重要:** 要进入低电压编程 (Low-Voltage Programming, LVP) 模式，必须首先移入高 4 位的最高有效位 (Most Significant bit, MSb)。这一点与其他一些器件系列有所不同，后者采用输入密钥序列的方式。

目录

简介.....	1
1. 概述.....	3
1.1. 编程数据流.....	3
1.2. 使用的引脚.....	3
1.3. 硬件要求.....	3
1.4. 写和/或擦除操作选择.....	3
2. 存储器映射.....	5
2.1. 用户 ID 单元.....	5
2.2. 器件/版本 ID.....	5
2.3. 器件信息区 (DIA)	5
2.4. 器件配置信息 (DCI)	6
2.5. 配置字.....	7
2.6. 器件 ID.....	8
2.7. 版本 ID.....	9
3. 编程算法.....	10
3.1. 编程/校验模式.....	10
3.2. 编程/校验命令.....	12
3.3. 编程算法.....	17
3.4. 代码保护.....	23
3.5. Hex 文件用法.....	23
3.6. CRC 校验和计算.....	23
4. 电气规范.....	24
5. 附录 A: 版本历史.....	26
6. 附录 B: 引脚分配说明和配置字.....	27
6.1. CONFIG1.....	28
6.2. CONFIG2.....	30
6.3. CONFIG3.....	32
6.4. CONFIG4.....	34
6.5. CONFIG5.....	36
Microchip 信息.....	37
商标.....	37
法律声明.....	37
Microchip 器件代码保护功能.....	37

1. 概述

1.1. 编程数据流

非易失性存储器（Nonvolatile Memory, NVM）编程数据可通过高电压在线串行编程（In-Circuit Serial Programming™, ICSP™）接口或低电压 ICSP 接口来提供。该数据可编程到闪存程序存储器（Program Flash Memory, PFM）、专用用户 ID 单元和配置字中。

1.2. 使用的引脚

ICSP 编程需要使用 5 个引脚。表 1-1 列出了这些引脚。有关引脚位置和封装信息，请参见表 6-1。

表 1-1. 编程期间所用引脚的说明

引脚名称	编程期间		
	功能	引脚类型	引脚说明
ISCPCLK	ICSPCLK	I	时钟输入——施密特触发器输入
ISCPDAT	ICSPDAT	I/O	数据输入/输出——施密特触发器输入
MCLR/V _{PP}	编程/校验模式	I ⁽¹⁾	编程模式选择
V _{DD}	V _{DD}	P	电源
V _{SS}	V _{SS}	P	地

图注: I = 输入, O = 输出, P = 功率

注:

1. 用于编程的高电压在内部生成。要激活编程/校验模式，需要对 $\overline{\text{MCLR}}$ 输入施加高电压。由于 $\overline{\text{MCLR}}$ 用于提供电压源，因此 $\overline{\text{MCLR}}$ 不会消耗太大的电流。

1.3. 硬件要求

1.3.1. 高电压 ICSP 编程

在高电压 ICSP 模式下，器件需要两个可编程的电源：一个是 V_{DD} ，另一个由 $\overline{\text{MCLR}}/V_{PP}$ 引脚提供。

1.3.2. 低电压 ICSP 编程

在低电压 ICSP 模式下，可在器件工作范围内使用单个 V_{DD} 电压源对器件进行编程。 $\overline{\text{MCLR}}/V_{PP}$ 引脚无需达到编程电压，可以继续保留在正常工作电压。

1.3.2.1. 单电源 ICSP 编程

器件的 LVP 配置位用于使能单电源（低电压）ICSP 编程。LVP 位默认为 1（使能）。只能通过进入高电压 ICSP 模式将 LVP 位编程为 0，也就是将 $\overline{\text{MCLR}}/V_{PP}$ 引脚上的电压升至 V_{IHH} 。一旦将 LVP 位编程为 0，便只能使用高电压 ICSP 模式来编程器件。



重要:

- 无论 LVP 位的状态如何，只要将 V_{IHH} 施加到 $\overline{\text{MCLR}}/V_{PP}$ 引脚，始终可以使用高电压 ICSP 模式。
- 在低电压 ICSP 模式下，无论 MCLRE 位的状态如何，都始终使能 $\overline{\text{MCLR}}$ 。此外， $\overline{\text{MCLR}}$ 引脚无法再用作通用输入。

1.4. 写和/或擦除操作选择

擦除或写操作根据用于开始操作的命令（见表 3-1）来选择。表 1-2 定义了本文档中与擦除/写入程序存储器相关的术语。

表 1-2. 编程术语

术语	定义
已编程单元	值为逻辑 0 的存储单元
已擦除单元	值为逻辑 1 的存储单元
擦除	将存储单元从 0 改为 1
写	将存储单元从 1 改为 0
编程	通用擦除和/或写

1.4.1. 擦除存储器

闪存程序存储器 (PFM) 可按行擦除或全片擦除，配置字可按字擦除或全片擦除。“行”是指可以编程/擦除的最少字数，“批量”涵盖总存储空间的许多子集。擦除操作的持续时间始终在内部确定，并且与存储器的大小有关。所有批量 ICSP 擦除命令都有最小 V_{DD} 要求，并且高于行擦除和写命令的要求。



重要：自写操作不支持批量擦除。

1.4.2. 写入存储器

闪存程序存储器 (PFM) 一次写入一行。可以多次使用“为 NVM 装载数据”命令来填充 PFM 的行数据锁存器。写操作的持续时间在内部或外部都能确定。

1.4.3. 多字编程接口

PFM 分区包含 32 字 (一行) 编程接口。对行进行编程之前，必须先使用批量擦除或行擦除命令对其进行擦除。

2. 存储器映射

本章详细介绍该系列器件的存储器构成。

表 2-1. 程序存储器映射

地址	器件				
	PIC16F131x3	PIC16F131x4	PIC16F131x5		
0000h 至 07FFh	闪存程序存储器 (2 KW) ⁽¹⁾	不存在 ⁽²⁾	闪存程序存储器 (8 KW) ⁽¹⁾		
0800h 至 0FFFh	不存在 ⁽²⁾				
1000h 至 1FFFh					
4000h 至 7FFFh					
8000h 至 8003h	用户 ID (4 字) ⁽³⁾				
8004h	保留				
8005h	版本 ID (1 字) ⁽³⁾⁽⁴⁾⁽⁵⁾				
8006h	器件 ID (1 字) ⁽³⁾⁽⁴⁾⁽⁵⁾				
8007h 至 800Bh	配置字 ⁽³⁾				
800Ch 至 80FFh	保留				
8100h 至 813Fh	器件信息区 (DIA) ⁽³⁾⁽⁵⁾				
8140h 至 81FFh	保留				
8200h 至 82FFh	器件配置信息 ⁽³⁾⁽⁴⁾⁽⁵⁾				
8300h 至 FFFFh	保留				

注:

- 如果使能存储区闪存 (Storage Area Flash, SAF)，则以闪存程序存储器的最后 128 字来实现。
- 地址不会计满返回。该区域读为 0。使用 NVMCON 寄存器访问这些区域时，读/或写操作会将 NVMERR 位置 1。
- 不受代码保护。
- 在芯片中硬编码。
- 用户无法写入该区域，而且无法对其进行批量擦除。

2.1. 用户 ID 单元

用户可将标识信息 (用户 ID) 存储到四个存储单元中。用户 ID 单元映射到 8000h - 8003h。每个存储单元的长度为 14 位。代码保护对这些存储单元不起作用。无论是否使能代码保护，都可以读取这些存储单元。

2.2. 器件/版本 ID

14 位 **器件 ID** 寄存器位于 8006h，14 位 **版本 ID** 寄存器位于 8005h。这些单元是只读的，不能擦除或修改。

2.3. 器件信息区 (DIA)

器件信息区 (DIA) 是闪存程序存储器中的专用区域。数据映射范围为地址 8100h 到 813Fh。这些单元是只读的，不能擦除或修改。DIA 包含 Microchip 惟一标识符字以及固定参考电压 (Fixed Voltage Reference, FVR) 读数 (测量单位为 mV)。表 2-2 列出了 PIC16F13145 系列单片机的 DIA 信息。

表 2-2. 器件信息区

地址范围	区域名称	标准器件信息
8100h-8108h	MUI0	Microchip 唯一标识符 (9 个字)
	MUI1	
	MUI2	
	MUI3	
	MUI4	
	MUI5	
	MUI6	
	MUI7	
	MUI8	
8109h	MUI9	保留 (1 字)
810Ah-8111h	EUI0	可选外部唯一标识符 (8 字)
	EUI1	
	EUI2	
	EUI3	
	EUI4	
	EUI5	
	EUI6	
	EUI7	
8112h	TSLR1 ⁽¹⁾	增益 $= \frac{0.1C \times 256}{count}$ (低范围设置)
8113h	TSLR2 ⁽¹⁾	温度指示器 ADC 读数为 90°C (低范围设置)
8114h	TSLR3 ⁽¹⁾	偏移 (低范围设置)
8115h	TSHR1 ⁽²⁾	增益 $= \frac{0.1C \times 256}{count}$ (高范围设置)
8116h	TSHR2 ⁽²⁾	温度指示器 ADC 读数为 90°C (高范围设置)
8117h	TSHR3 ⁽²⁾	偏移 (高范围设置)
8118h	FVRA1X	1x 设置的 ADC FVR1 输出电压 (单位为 mV)
8119h	FVRA2X	2x 设置的 ADC FVR1 输出电压 (单位为 mV)
811Ah	FVRA4X	4x 设置的 ADC FVR1 输出电压 (单位为 mV)
811Bh	FVRC1X	1x 设置的比较器 FVR2 输出电压 (单位为 mV)
811Ch	FVRC2X	2x 设置的比较器 FVR2 输出电压 (单位为 mV)
811Dh	FVRC4X	4x 设置的比较器 FVR2 输出电压 (单位为 mV)
811Eh-811Fh	保留	保留 (2 字)

注:

1. TSLR: 地址 8112h-8114h 存储 $V_{DD} = 3V$ 且 $V_{REF^+} = 2.048V$ (来自 FVR1) 时温度传感器低范围设置的测量值。
2. TSHR: 地址 8115h-8117h 存储 $V_{DD} = 3V$ 且 $V_{REF^+} = 2.048V$ (来自 FVR1) 时温度传感器高范围设置的测量值。

2.4. 器件配置信息 (DCI)

器件配置信息 (Device Configuration Information, DCI) 是存储器中的专用区域, 其中存储的器件相关信息对于编程和自举程序应用十分有用。该区域中存储的数据是只读的, 不能修改/擦除。有关完整的 DCI 表地址和说明, 请参见表 2-3。

表 2-3. 器件配置信息

地址	名称	说明	值			单位
			PIC16F131x3	PIC16F131x4	PIC16F131x5	
8200h	ERSIZ	擦除行大小		32		字

表 2-3. 器件配置信息 (续)

地址	名称	说明	值			单位
			PIC16F131x3	PIC16F131x4	PIC16F131x5	
8201h	WLSIZ	每行写锁存器数量		32		字
8202h	URSIZ	用户可擦除行数量	64	128	256	行
8203h	EESIZ	数据 EEPROM 存储器大小		0		字节
8204h	PCNT	引脚数		8/14/20		引脚

2.5. 配置字

器件包含5个配置字，从地址 8007h 开始。配置位用于使能或禁止特定功能，将这些控制置于正常软件过程之外，并在任何软件执行之前建立配置值。

从编程角度来看，应考虑以下重要的配置位：

1. LVP: 低电压编程使能位

- 1 = ON: 使能低电压编程。 $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ 引脚功能为 $\overline{\text{MCLR}}$ 。忽略 MCLRE 配置位。
- 0 = OFF: 必须使用 $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ 上的高电压进行编程。



重要: 通过 LVP 编程接口工作时，不能对 LVP 位进行写操作（写为 0）。该规则的目的是防止用户在通过 LVP 模式编程时退出 LVP 模式，或意外地从配置状态中删除 LVP 模式。更多信息，请参见**低电压编程 (LVP) 模式**。

2. MCLRE: 主复位 ($\overline{\text{MCLR}}$) 使能位

- 如果 LVP = 1: RA3 引脚功能为 $\overline{\text{MCLR}}$
- 如果 LVP = 0
 - 1 = $\overline{\text{MCLR}}$ 引脚为 $\overline{\text{MCLR}}$
 - 0 = $\overline{\text{MCLR}}$ 引脚功能为端口定义的功能

3. CP: 用户 NVM 程序存储器代码保护位

- 1 = OFF: 禁止用户闪存程序存储器代码保护
- 0 = ON: 使能用户闪存程序存储器代码保护

有关代码保护的更多信息，请参见[代码保护](#)。

2.6. 器件 ID

名称: DEVICEID
偏移量: 8006h

器件 ID 寄存器

位	15	14	13	12	11	10	9	8
DEV[11:8]								
访问			R	R	R	R	R	R
复位			1	1	q	q	q	q
位	7	6	5	4	3	2	1	0
DEV[7:0]								
访问	R	R	R	R	R	R	R	R
复位	q	q	q	q	q	q	q	q

Bit 13 - 保留 保留——读为 1

Bit 12 - 保留 保留——读为 1

Bit 11:0 - DEV[11:0] 器件 ID

器件	器件 ID
PIC16F13113	3121h
PIC16F13114	3124h
PIC16F13115	3127h
PIC16F13123	3122h
PIC16F13124	3125h
PIC16F13125	3128h
PIC16F13143	3123h
PIC16F13144	3126h
PIC16F13145	3129h

2.7. 版本 ID

名称: REVISIONID
偏移量: 8005h

版本 ID 寄存器

位	15	14	13	12	11	10	9	8
			保留	保留		MJRREV[5:2]		
访问			R	R	R	R	R	R
复位			1	0	q	q	q	q
位	7	6	5	4	3	2	1	0
	MJRREV[1:0]				MNRREV[5:0]			
访问	R	R	R	R	R	R	R	R
复位	q	q	q	q	q	q	q	q

Bit 13 - 保留 保留——读为 1

Bit 12 - 保留 保留——读为 0

Bit 11:6 - MJRREV[5:0] 主要版本 ID

这些位用于标识主要版本 (A0、B0 和 C0 等)。

Bit 5:0 - MNRREV[5:0] 次要版本 ID

这些位用于标识次要版本。

3. 编程算法

3.1. 编程/校验模式

在编程/校验模式下，可以采用串行方式访问程序存储器和配置存储器并对其进行编程。ICSPDAT 和 ICSPCLK 分别用于数据和时钟。所有命令和数据字都是先发送 MSb。数据在 ICSPCLK 的上升沿发生变化，在下降沿锁存。在编程/校验模式下，ICSPDAT 和 ICSPCLK 引脚都是施密特触发器输入。使器件进入编程/校验模式的序列会将所有其他逻辑都置于复位状态；所有 I/O 都自动配置为高阻抗输入，并且程序计数器（Program Counter, PC）清零。

3.1.1. 进入和退出高电压编程/校验模式

通过高电压进入编程/校验模式时有以下两种模式：

- V_{PP} 先升进入模式
- V_{DD} 先升进入模式

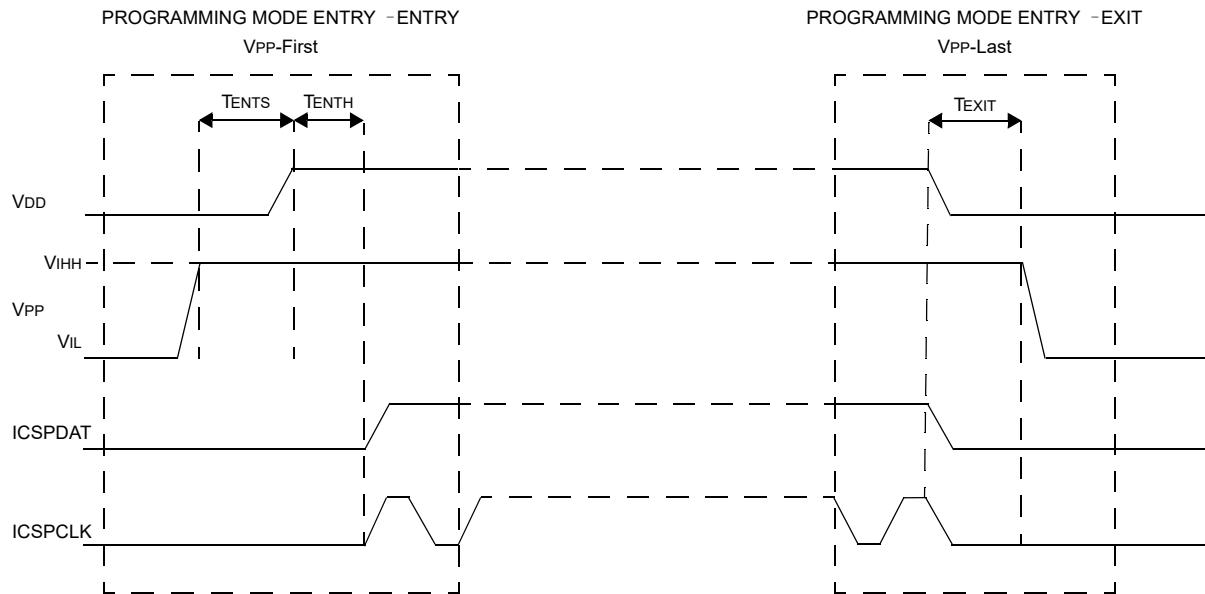
3.1.1.1. V_{PP} 先升进入模式

要通过 V_{PP} 先升模式进入编程/校验模式，必须遵循以下序列：

1. 使 ICSPCLK 和 ICSPDAT 保持低电平。
2. 将 \overline{MCLR} 上的电压从 0V 升至 V_{IHH} 。
3. 将 V_{DD} 上的电压从 0V 升至所需工作电压。

V_{PP} 先升进入模式会在进入编程/校验模式之前阻止器件执行代码。例如，当已通过编程配置字禁止 \overline{MCLR} ($MCLRE = 0$)、禁止上电延时定时器 ($PWRTE = 0$) 并选择内部振荡器后，器件将立即执行代码。强烈建议使用 V_{PP} 先升进入模式，因为它会阻止执行用户代码。相关时序图如图 3-1 所示。

图 3-1. 编程进入和退出模式—— V_{PP} 先升后降



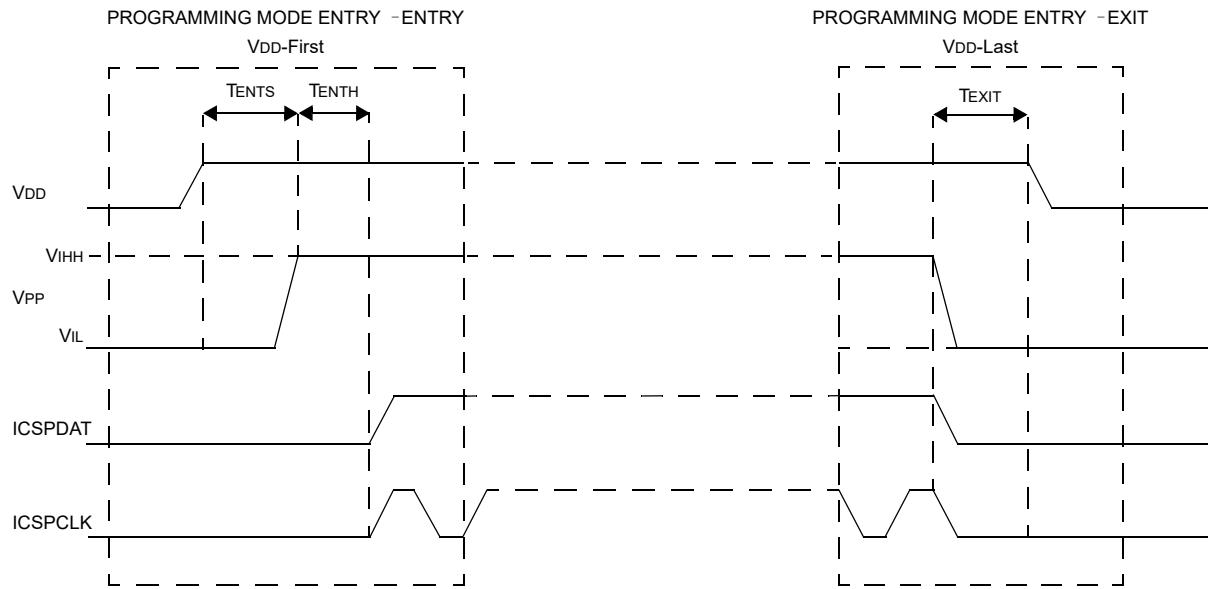
3.1.1.2. V_{DD} 先升进入模式

要通过 V_{DD} 先升模式进入编程/校验模式，必须遵循以下序列：

1. 使 ICSPCLK 和 ICSPDAT 保持低电平。
2. 将 V_{DD} 上的电压从 0V 升至所需工作电压。
3. 将 \overline{MCLR} 上的电压从 V_{DD} 或更低电压升至 V_{IHH} 。

在已施加 V_{DD} 的情况下编程器件时， V_{DD} 先升进入模式会很有用。无需将 V_{DD} 断开连接即可进入编程/校验模式。相关时序图如图 3-2 所示。

图 3-2. 编程进入和退出模式—— V_{DD} 先升后降



3.1.1.3. 退出编程/校验模式

要退出编程/校验模式，应将 \overline{MCLR} 从 V_{IHH} 降至 V_{IL} 。 V_{PP} 先升进入模式将与 V_{PP} 后降退出模式搭配使用（见图 3-1）。 V_{DD} 先升进入模式将与 V_{DD} 后降退出模式搭配使用（见图 3-2）。

3.1.2. 低电压编程 (LVP) 模式

低电压编程模式允许器件在没有高电压的情况下仅使用 V_{DD} 进行编程。当配置字 4 寄存器中的 LVP 位设置为 1 时，将会使能低电压 ICSP 编程进入模式。要禁止低电压 ICSP 模式，必须将 LVP 位设置为 0。这只能在高电压进入模式下进行。

进入低电压 ICSP 编程/校验模式需要以下步骤：

1. \overline{MCLR} 拉至 V_{IL} 。
2. 在 ICSPDAT 引脚上输出 32 位密钥序列，并通过 ICSPCLK 引脚提供时钟。该模式的最低有效位 (Least Significant bit, LSb) 是“任意值 x”。编程/校验模式进入模式检测硬件只校验该序列的前 31 位，在模式检测生效之前需要最后 1 个时钟位。

该密钥序列为一个特定的 32 位形式 32' h4d434850（以其 ASCII 格式 MCHP 记忆更为简便）。只有在密钥序列有效时，器件才能进入编程/校验模式。首先必须移入最高有效字节 (Most Significant Byte, MSB) 的 MSb。在密钥序列完成后，只要保持在编程/校验模式下， \overline{MCLR} 就必须保持在 V_{IL} 电平。关于低电压编程时序，请参见图 3-3 和图 3-4。

图 3-3. LVP 进入 (已供电)

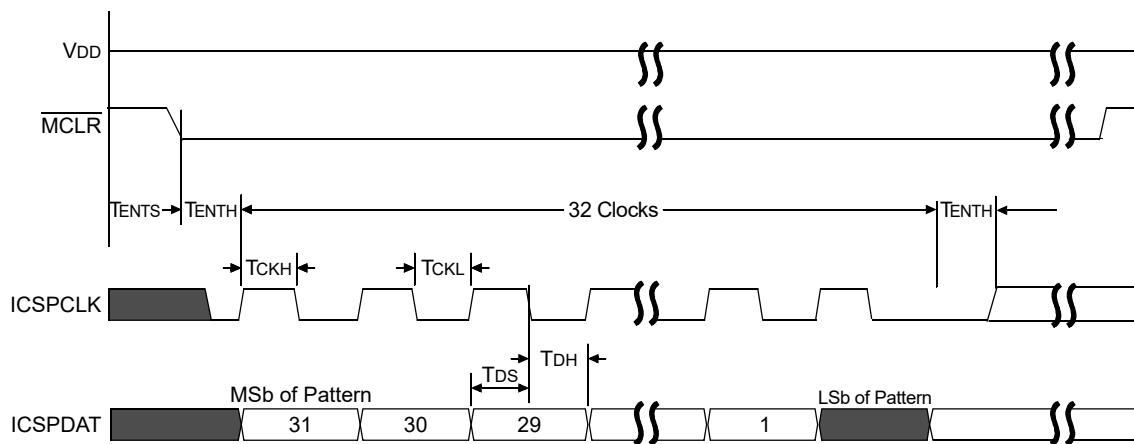
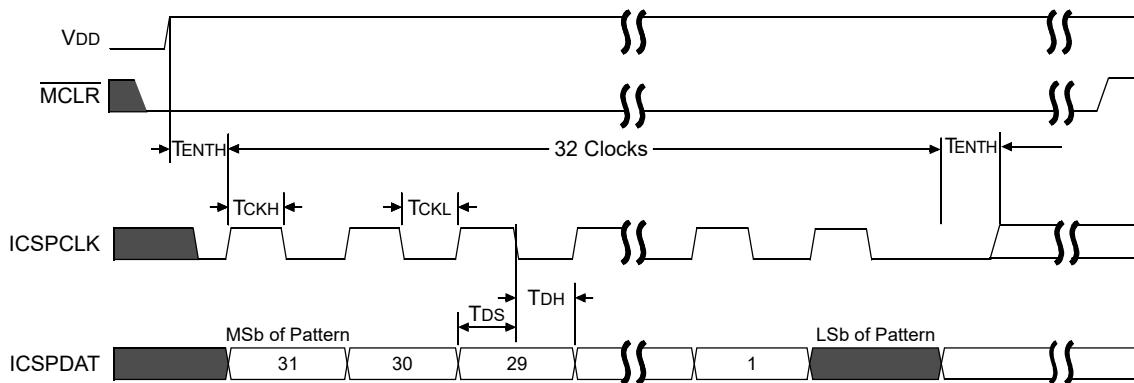


图 3-4. LVP 进入 (正在上电)



Exiting the Program/Verify mode is done by raising **MCLR** from below **VIL** to **VIH** level (or higher, up to **VDD**).



重要:

要进入 LVP 模式，首先必须移入高 4 位的 **MSb**。这一点与其他一些器件系列有所不同，后者采用输入密钥序列的方式。

3.2. 编程/校验命令

当器件进入 ICSP 编程/校验模式（使用高电压或 LVP 进入）后，编程主器件可以向单片机发出命令，每条命令的长度为 8 位。表 3-1 汇总了用于根据程序计数器（PC）的位置来擦除或编程器件的命令。

一些 8 位命令还有关联的数据有效负载（如“装载 PC 地址”和“从 NVM 读取数据”命令）。

如果主器件发出一个具有关联数据有效负载的 8 位命令字节，则主器件负责发送额外的 24 个时钟脉冲（例如，三个 8 位字节），用以发送或接收与该命令关联的有效负载数据。

有效负载字段大小与许多基于 SPI 的 8 位系统兼容。在每个 24 位有效负载字段内，发送的第一个位总是起始位，然后是数量不等的填充位，接着是有用的数据有效负载位，最后是一个停止位。有用的数据有效负载位始终先发送 **MSb**。

当编程器件发出的命令涉及从主器件到单片机的有效负载（如“装载 PC 地址”命令）时，起始位、停止位和填充位必须均由编程器驱动，如表 3-1 中的数据列所定义。当编程主器件发出的命令涉及从单片机到主

器件的有效负载数据（如“从 NVM 读取数据”命令）时，必须将起始位、停止位和填充位视为“无关”位，主器件可以忽略这些位的值。

当编程主器件向单片机发出一个 8 位命令字节时，主器件必须先等待一段指定的最短延时（特定于命令），然后再发送任何额外的时钟脉冲（与 24 位数据有效负载字段或下一个命令字节相关联）。

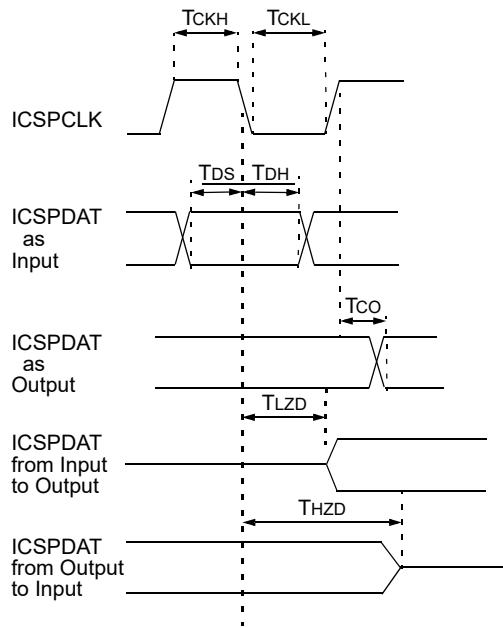
表 3-1. ICSP™ 命令集汇总

命令名称	命令值		预期的有效负载	发出命令后的延时	数据/注释
	二进制（从 MSb 到 LSb）	十六进制			
装载 PC 地址	1000 0000	80	有	T_{DLY}	有效负载值 = PC
批量擦除程序存储器	0001 1000	18	有	T_{ERAB}	有效负载用于标识需要批量擦除的区域。
行擦除程序存储器	1111 0000	F0	无	T_{ERAS}	擦除由 PC 的 MSb 寻址的行；忽略 LSb。
为 NVM 装载数据	0000 00J0	00/02	有——数据输入	T_{DLY}	将数据装载到由 PC 的 LSb 寻址的数据锁存器；忽略 MSb。 J = 0: PC 保持不变 J = 1: 写入后, PC = PC + 1
从 NVM 读取数据	1111 11J0	FC/FE	有——数据输出	T_{DLY}	如果使能代码保护，数据输出为 0。 J = 0: PC 保持不变 J = 1: 读取后, PC = PC + 1
递增地址	1111 1000	F8	无	T_{DLY}	$PC = PC + 1$
开始内部计时编程	1110 0000	E0	无	T_{PINT}	将锁存的数据提交到 NVM（自计时）。
开始外部计时编程	1100 0000	C0	无	T_{PEXT}	将锁存的数据提交到 NVM（外部计时）。经过 T_{PEXT} 之后，必须使用“结束外部计时编程”命令。
结束外部计时编程	1000 0010	82	无	T_{DIS}	必须在发出“开始外部计时编程”命令后要求的延时 (T_{PEXT}) 内发出。



重要: 8 位命令和 24 位有效负载字段的所有时钟脉冲都由编程主器件生成。单片机不驱动 ICSPCLK 线。ICSPDAT 信号为双向数据线。对于所有的命令和有效负载字段（“从 NVM 读取数据”有效负载除外），编程主器件持续驱动 ICSPDAT 线。主编程器和单片机都必须在 ICSPCLK 线的下降沿锁存接收到的 ICSPDAT 值。当单片机从主编程器接收到 ICSPDAT 线值时，ICSPDAT 值在 ICSPCLK 下降沿之前的有效时间必须至少为 T_{DS} ，而在 ICSPDAT 下降沿之后的有效时间必须至少为 T_{DH} 。请参见图 3-5。

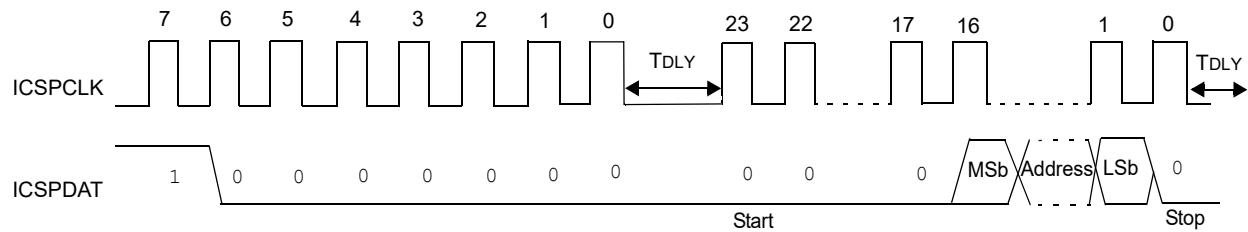
图 3-5. 时钟与数据时序



3.2.1. 装载 PC 地址

PC 值使用提供的数据来设置。地址指示要访问的存储器分区（见图 3-6）。

图 3-6. 装载 PC 地址



3.2.2. 批量擦除

批量擦除命令用于完全擦除不同的存储器区域。区域选择由有效负载中的位域实现。

通过设置有效负载的以下位，可对相应的存储器区域进行批量擦除。允许同时设置多个位。

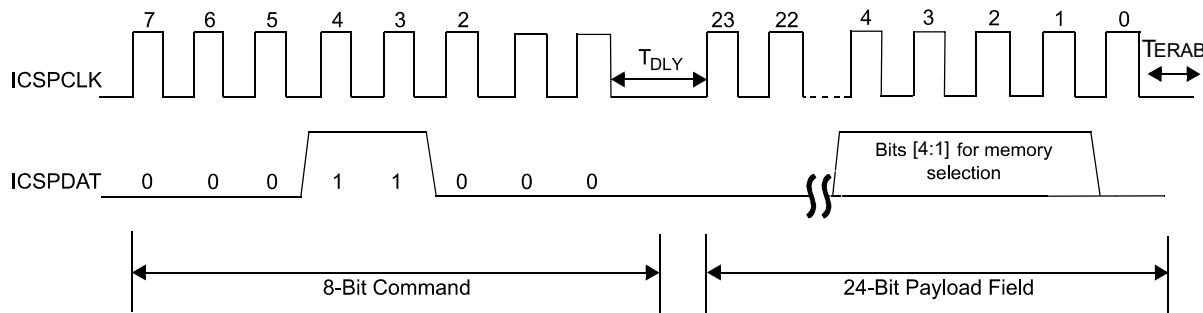
1. Bit 0: 数据 EEPROM (如果可用)
2. Bit 1: 闪存
3. Bit 2: 用户 ID 存储器
4. Bit 3: 配置存储器



重要: 如果器件受到代码保护，当发出针对配置存储器的批量擦除命令时，所有其他区域也将一并批量擦除。

收到批量擦除命令后，将在时间间隔 T_{ERAB} 后完成擦除操作。有关批量擦除命令的结构，请参见图 3-7。

图 3-7. 批量擦除存储器

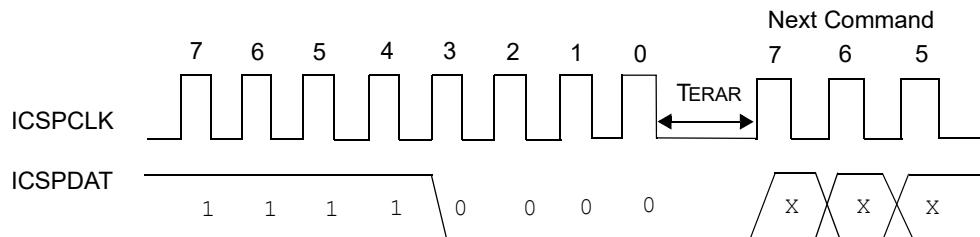


3.2.3. 行擦除程序存储器

“行擦除程序存储器”命令将擦除单个行。如果程序存储器受代码保护，则将忽略“行擦除程序存储器”命令。当地址为 8000h-8004h 时，“行擦除程序存储器”命令将只擦除用户 ID 单元，而与 \overline{CP} 配置位的设置无关。按行执行写操作和擦除操作时，擦除操作的行大小（14 位字的数量）为 32，写操作的行大小（14 位锁存器的数量）也为 32。

将擦除的闪存行由当前 PC 定义。用户必须等待 T_{ERAR} 才能完成擦除操作（见图 3-8）。

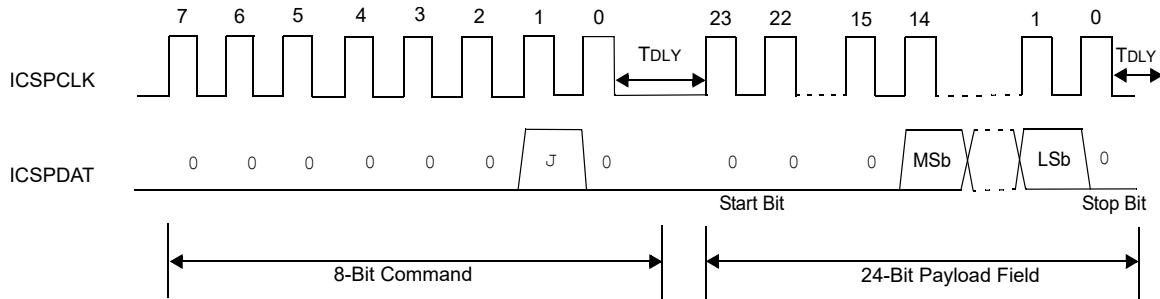
图 3-8. 行擦除程序存储器



3.2.4. 为 NVM 装载数据

“为 NVM 装载数据”命令用于装载一个编程数据锁存器（例如，为程序存储器/配置存储器/用户 ID 存储器装载一个 14 位指令字）。“为 NVM 装载数据”命令可用于为闪存程序存储器装载数据（见图 3-9）。在“开始内部计时编程”或“开始外部计时编程”命令后对程序存储器进行的字写操作将写入数据锁存器的一整行，而非仅仅写入一个字。地址的低 5 位是有意义的，其他位将被忽略。程序计数器（PC）是否递增取决于命令 bit 1 的值（见表 3-1）。

图 3-9. 为 NVM 装载数据

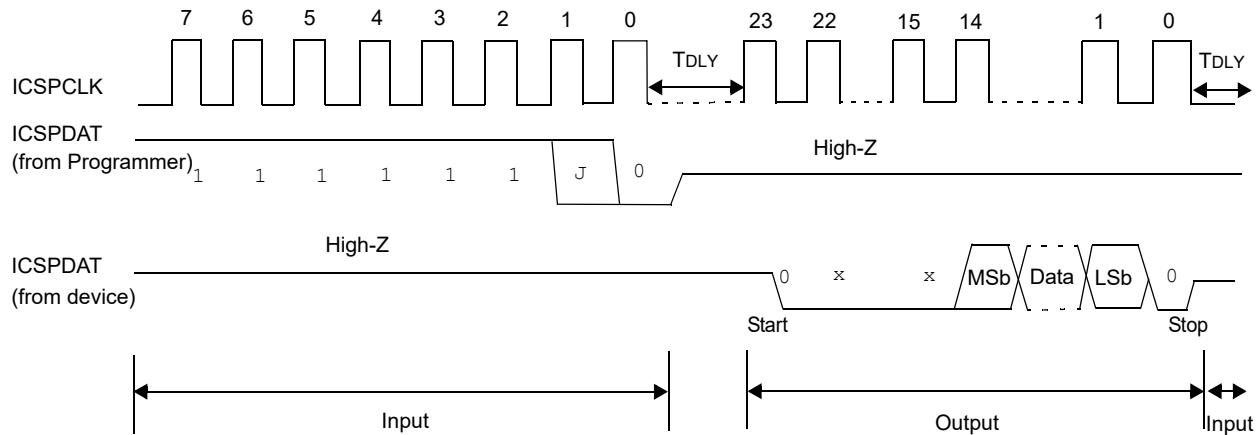


3.2.5. 从 NVM 读取数据

“从 NVM 读取数据”命令将发送当前 PC 地址中的数据位。ICSPDAT 引脚将在 ICSPCLK 的第一个下降沿进入输出模式，然后在该时钟的第 24 个下降沿之后恢复输入模式（高阻抗）。起始位和停止位的宽度只有位时间的一半，必须被主编程器忽略（因为锁存的值可能是不确定的）。此外，主编程器必须只将 MSb 到

LSb 的有效负载位视为有效，并且必须忽略填充位的值。如果程序存储器受代码保护 ($\overline{CP} = 0$)，则数据将读为零（见图 3-10）。程序计数器（PC）是否递增取决于命令 bit 1 的值（见表 3-1）。

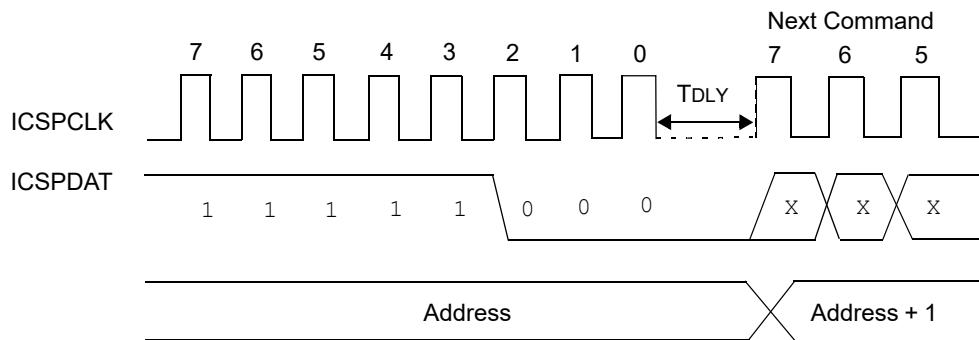
图 3-10. 从 NVM 读取数据



3.2.6. 递增地址

收到该命令时，PC 递增 1。地址无法递减。要复位该计数器，用户必须使用“装载 PC 地址”命令。该命令执行的操作与装载/读命令中的 J 位相同。请参见图 3-11。

图 3-11. 递增地址

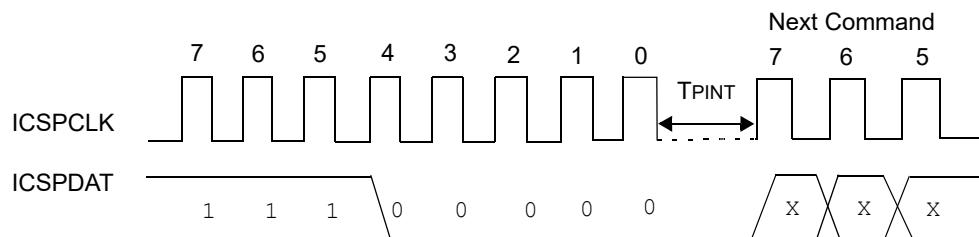


3.2.7. 开始内部计时编程

发出“开始内部计时编程命令”之前，必须已经使用“为 NVM 装载数据”命令装载了写编程锁存器。收到该命令后，将开始编程寻址的存储器行。地址的低位将被忽略。内部计时机制执行写操作。用户必须留出擦除/写周期时间 (T_{PINT}) 来完成编程，之后才能发出下一条命令（见图 3-12）。

编程周期结束后，所有数据锁存器都复位为 1。

图 3-12. 开始内部计时编程

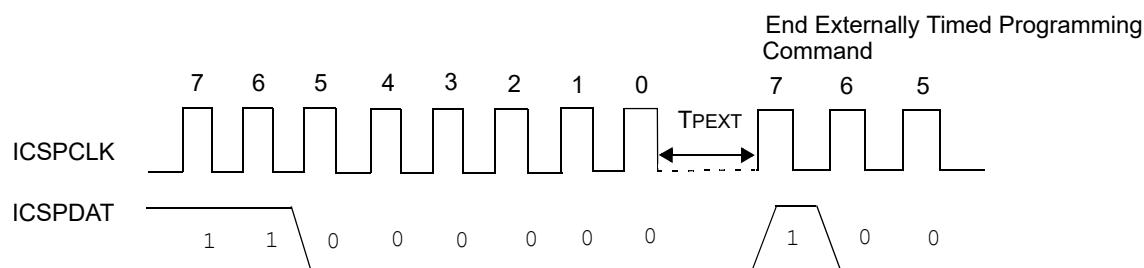


3.2.8. 开始外部计时编程

每次发出“开始外部计时编程”命令之前，必须先使用“为 NVM 装载数据”命令装载要编程的数据。若要结束编程，必须在由 T_{PEXT} 定义的指定时间窗口内发送“开始外部计时编程”命令（见图 3-13）。地址的低位将被忽略。

配置字不支持外部计时写操作。针对配置字的任何外部计时写操作都不会对目标字产生影响。

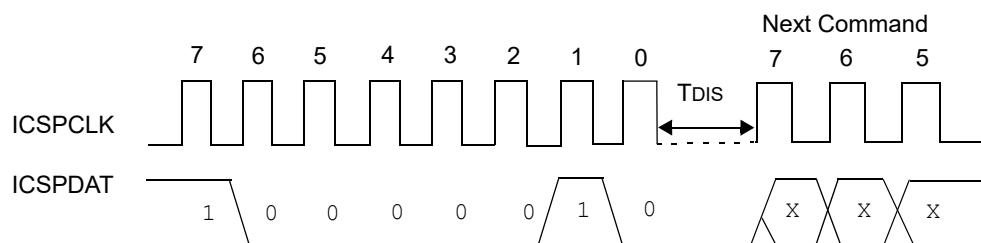
图 3-13. 开始外部计时编程



3.2.9. 结束外部计时编程

发出“开始外部计时编程”命令后，需要使用“结束外部计时编程”命令才能终止编程序列。如果未在执行编程命令或者编程周期为内部计时，则该命令将作为一个空操作（NOP）来执行（见图 3-14）。

图 3-14. 结束外部计时编程



3.3. 编程算法

器件使用内部锁存器来临时存储用于编程的 14 位字。数据锁允许用户使用一条“开始内部计时编程”或“开始外部计时编程”命令实现整行编程。“为 NVM 装载数据”命令用于装载单个数据锁存器。在发出“开始内部计时编程”或“开始外部计时编程”命令之前，数据锁存器将一直保存数据。

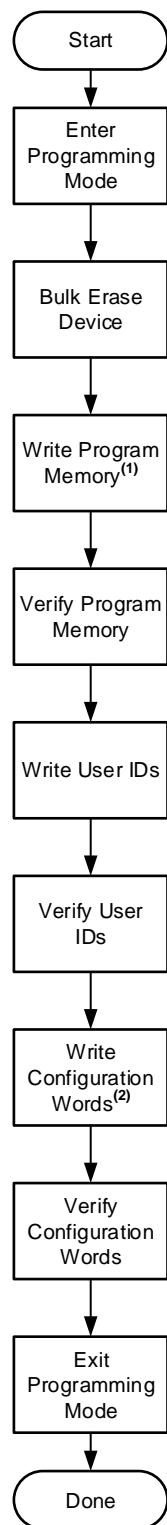
数据锁存器与地址的 Lsb 对齐。发出“开始内部计时编程”或“开始外部计时编程”命令时所在的地址将决定写入的存储器行。写操作不能跨越物理行边界。例如，在具有 32 个锁存器的器件中尝试写入地址 0002h-0021h 时，数据最终将写入 0020h-003Fh。

如果对锁存器执行写操作时超出了锁存器的最大数量，并且没有使用“开始内部计时编程”或“开始外部计时编程”命令，则数据锁存器中的数据将被覆盖。下面给出了推荐的编程流程图。



重要：闪存程序存储器区域每次编程一行（32 字）（图 3-18），而配置字每次编程一个字（图 3-17）。发出“开始内部计时编程”或“开始外部计时编程”命令时的 PC 值决定将编程闪存程序存储器的哪一行或配置字的哪一个字。

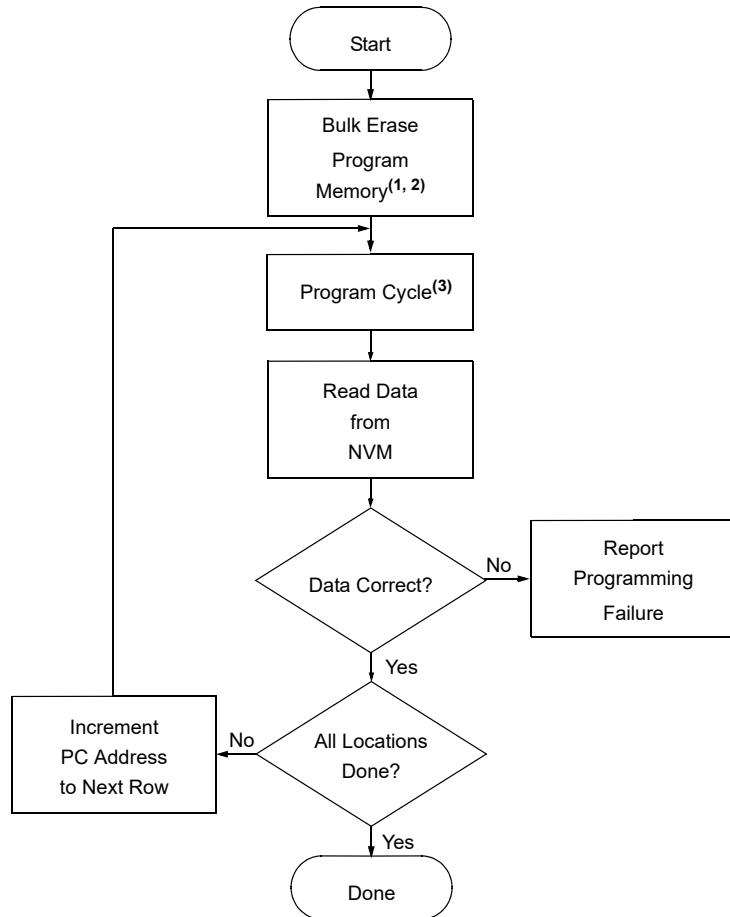
图 3-15. 器件编程/校验流程图



注：

1. 请参见[装载 PC 地址](#)一节。
 2. 请参见[行擦除程序存储器](#)一节。

图 3-16. 程序存储器流程图



注：

1. 如果器件已完成擦除或之前尚未编程，则可以选择跳过该步骤。
 2. 如果器件受代码保护或必须完全擦除，则按照图 3-20 对器件进行批量擦除。
 3. 请参见**批量擦除**一节。

图 3-17. 单字编程周期

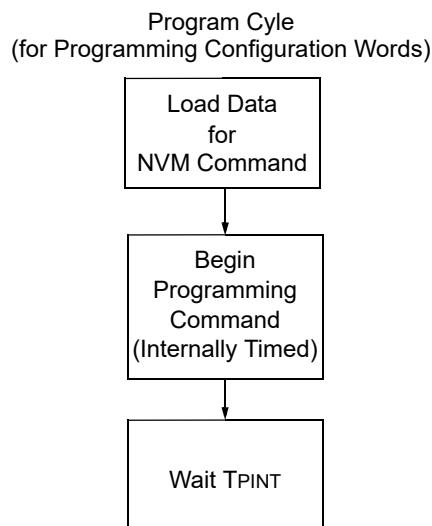


图 3-18. 多字编程周期

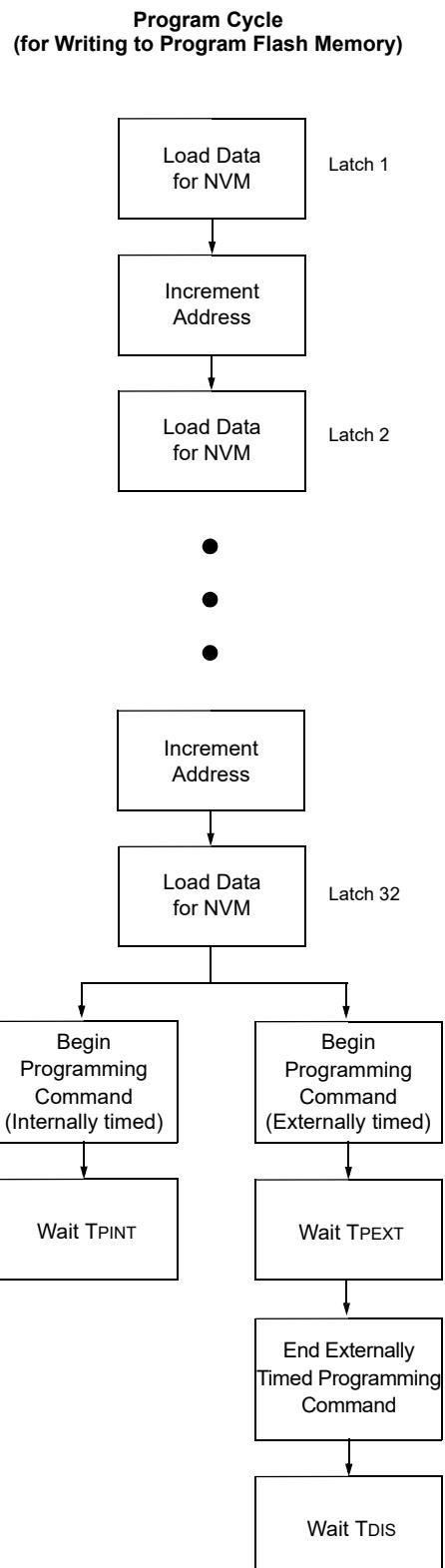
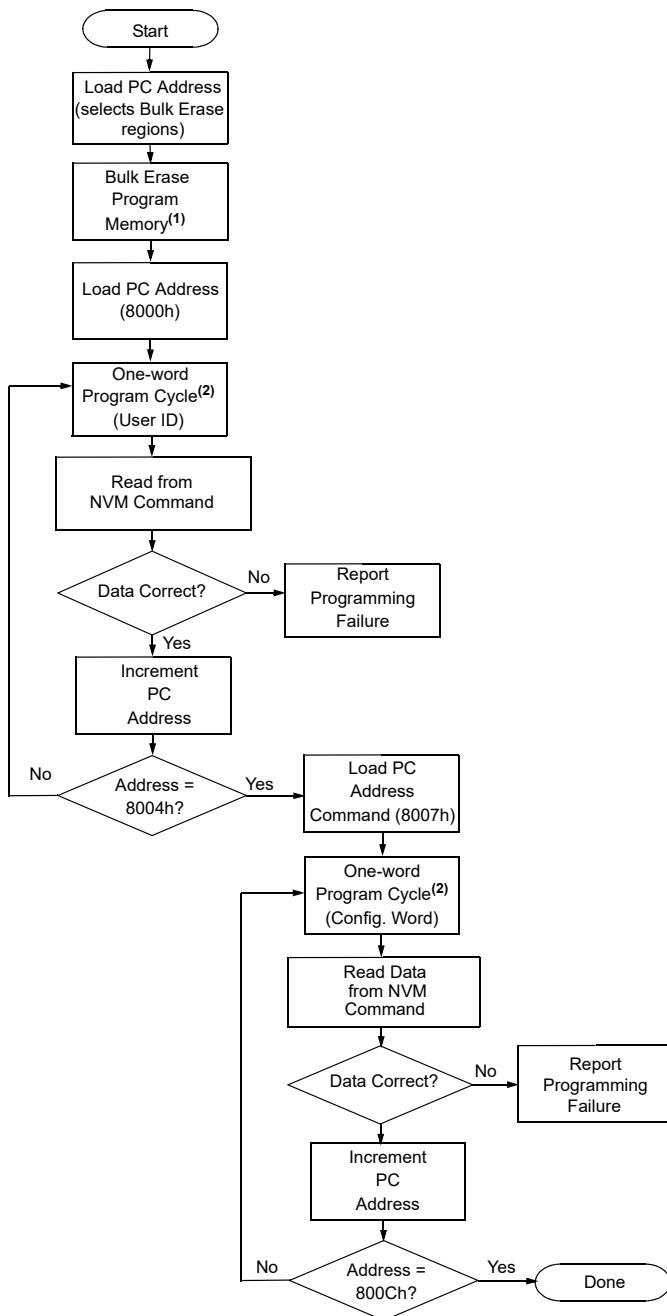


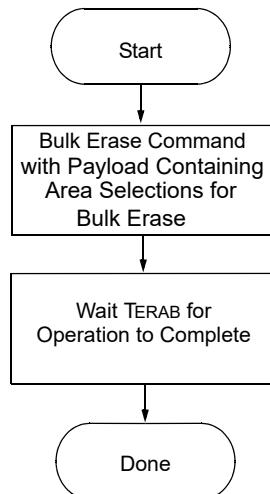
图 3-19. 配置存储器编程流程图



注：

1. 如果器件已擦除或之前未编程，则可以选择跳过该步骤。
2. 请参见开始内部计时编程一节。

图 3-20. 批量擦除流程图



3.4. 代码保护

代码保护使用 \overline{CP} 位进行控制。使能代码保护时，所有程序存储器单元均读为 0。禁止对程序存储器进行进一步的编程，除非对配置存储器区域执行批量擦除操作。在程序执行过程中，仍然可以对程序存储器进行编程和读取操作。

无论是否设置代码保护，都可以对用户 ID 存储单元和配置字进行编程或读取操作。

要禁止代码保护，惟一的方法是使用“批量擦除程序存储器”命令并将有效负载的 bit 4 设置为 1。这将禁止代码保护并擦除所有存储单元。

3.5. Hex 文件用法

在 hex 文件中，每个程序字有两个字节以 Intel® INHX32 hex 格式存储。数据按照先 LSB 后 MSB 的顺序存储。由于每个字有两个字节，因此 hex 文件中的地址数量是程序存储器中的地址数量的二倍。例如，如果配置字 1 存储在 8007h 处，则在 hex 文件中将以 1000Eh-1000Fh 来引用。

3.5.1. 配置字

为了实现代码的可移植性，强烈建议允许编程器从 hex 文件中读取配置字和用户 ID 单元。如果 hex 文件中没有配置字信息，可能会发出一条简单的警告消息。同理，在保存 hex 文件时，需要包含配置字和用户 ID 信息。



重要：Microchip 认为这一功能可以使得最终客户极大受益。

3.5.2. 器件 ID

如果器件 ID 位于 hex 文件中的 1000Ch-1000Dh（器件上的 8006h）处，则编程器必须对照从器件上读出的值来校验器件 ID。如果不匹配，编程器必须生成一条警告消息。

3.6. CRC 校验和计算

与旧款 PIC® 器件不同，Microchip 工具链对整个 hex 文件运行 32 位 CRC 计算，以计算其校验和。校验和采用标准 CRC-32 算法及多项式 0x4C11DB7

$$(x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1)。$$

4. 电气规范

有关绝对最大值的信息，请参见具体器件数据手册。

表 4-1. 编程/校验模式的交流/直流特性时序要求

交流/直流特性		标准工作条件 (在+25°C 下进行生产测试)				
符号	特性	最小值	典型值	最大值	单位	条件/备注
编程电源电压和电流						
V_{DD}	电源电压 (V_{DDMIN} 和 V_{DDMAX})	1.8	—	5.5	V	(注 1)
V_{PEW}	读/写和行擦除操作	V_{DDMIN}	—	V_{DDMAX}	V	
V_{BE}	批量擦除操作	V_{BORMAX}	—	V_{DDMAX}	V	(注 2)
I_{DDI}	V_{DD} 空闲电流	—	—	1	mA	
I_{DDP}	V_{DD} 编程电流	—	—	10	mA	
V_{PP}						
I_{PP}	$MCLR/V_{PP}$ 上的电流	—	—	600	μ A	
V_{IHH}	进入编程/校验模式所需的 $MCLR/V_{PP}$ 高电压	7.9	—	9	V	
T_{VHHR}	$MCLR$ 上升时间 (V_{IL} 到 V_{IHH})，之后可进入编程/校验模式	—	—	1	μ s	
I/O 引脚						
V_{IH}	(ICSPCLK、ICSPDAT 和 $MCLR/V_{PP}$) 高电平输入电压	0.8 V_{DD}	—	V_{DD}	V	
V_{IL}	(ICSPCLK、ICSPDAT 和 $MCLR/V_{PP}$) 低电平输入电压	V_{SS}	—	0.2 V_{DD}	V	
V_{OH}	ICSPDAT 高电平输出电压	$V_{DD}-0.7$	—	—	V	$I_{OH} = 3$ mA, $V_{DD} = 3.0V$
V_{OL}	ICSPDAT 低电平输出电压	—	—	$V_{SS} + 0.6$	V	$I_{OL} = 6$ mA, $V_{DD} = 3.0V$
编程模式进入和退出						
T_{EANTS}	编程模式进入建立时间: V_{DD} 或 $MCLR \uparrow$ 前的 ICSPCLK 和 ICSPDAT 建立时间	100	—	—	ns	
T_{EANTH}	编程模式进入保持时间: V_{DD} 或 $MCLR \uparrow$ 前的 ICSPCLK 和 ICSPDAT 保持时间	250	—	—	μ s	
串行编程/校验						
T_{CKL}	时钟低电平脉冲宽度	100	—	—	ns	
T_{CKH}	时钟高电平脉冲宽度	100	—	—	ns	
T_{DS}	时钟 \downarrow 前的数据输入建立时间	100	—	—	ns	
T_{DH}	时钟 \downarrow 后的数据输入保持时间	100	—	—	ns	
T_{CO}	时钟 \uparrow 到数据输出有效的时间 (在“从 NVM 读取数据”命令期间)	0	—	80	ns	
T_{LZD}	时钟 \downarrow 到数据低阻抗的时间 (在“从 NVM 读取数据”命令期间)	0	—	80	ns	
T_{HZD}	时钟 \downarrow 到数据高阻态的时间 (在“从 NVM 读取数据”命令期间)	0	—	80	ns	
T_{DLY}	数据输入未驱动到下一个时钟输入的时间 (命令/数据或命令/命令之间所需的延时)	1	—	—	μ s	
T_{ERAB}	批量擦除周期时间	—	—	20	ms	编程、配置和 ID
T_{ERAR}	行擦除周期时间	—	—	9	ms	
T_{PINT}	内部计时编程操作时间	—	—	7	ms	程序存储器
		—	—	12	ms	配置字
T_{PEXT}	开始外部计时编程命令到结束外部计时编程命令所需的延时	1	—	2.1	ms	(注 3)
T_{DIS}	结束外部计时编程命令后所需的延时	300	—	—	μ s	
T_{EXIT}	退出编程/校验模式时的延时	1	—	—	μ s	

表 4-1. 编程/校验模式的交流/直流特性时序要求 (续)

符号	交流/直流特性	标准工作条件 (在+25°C 下进行生产测试)					条件/备注
		最小值	典型值	最大值	单位		
注:							
1.	批量擦除器件默认使能欠压复位, 其中 $BORV = 1$ (低电平跳变点)。对批量擦除器件执行低电压编程时, V_{DDMIN} 作为 V_{BOR} 阈值 ($BORV = 1$) 以确保器件不会保持欠压复位状态。						
2.	硬件要求 V_{DD} 高于 BOR 阈值, 以执行批量擦除操作。该阈值与 $BORV$ 配置位设置无关。有关 V_{BOR} 电平的最小值/典型值/最大值限值, 请参见单片机数据手册规范。						
3.	配置字不支持外部计时写操作。						

5. 附录 A: 版本历史

文档版本	日期	备注
B	2023 年 9 月	更正了批量擦除和行擦除周期时间以及内部计时编程操作时间。
A	2023 年 5 月	文档初始版本。

6. 附录 B: 引脚分配说明和配置字

表 6-1. 不同封装类型的编程引脚位置

器件	封装	封装代码	V _{DD}	V _{SS}	MCLR		ICSPCLK		ICSPDAT	
			引脚	引脚	引脚	端口	引脚	端口	引脚	端口
PIC16F13113 PIC16F13114 PIC16F13115	8 引脚 PDIP	P	1	8	4	RA3	6	RA1	7	RA0
	8 引脚 SOIC	SN	1	8	4	RA3	6	RA1	7	RA0
	8 引脚 DFN	MF	1	8	4	RA3	6	RA1	7	RA0
PIC16F13123 PIC16F13124 PIC16F13125	14 引脚 PDIP	P	1	14	4	RA3	12	RA1	13	RA0
	14 引脚 SOIC	SL	1	14	4	RA3	12	RA1	13	RA0
	14 引脚 TSSOP	ST	1	14	4	RA3	12	RA1	13	RA0
	16 引脚 VQFN	MG	16	13	3	RA3	11	RA1	12	RA0
PIC16F13143 PIC16F13144 PIC16F13145	20 引脚 PDIP	P	1	20	4	RA3	18	RA1	19	RA0
	20 引脚 SOIC	SO	1	20	4	RA3	18	RA1	19	RA0
	20 引脚 SSOP	SS	1	20	4	RA3	18	RA1	19	RA0
	20 引脚 VQFN	REB	18	17	1	RA3	15	RA1	16	RA0

注: 最新封装图请至 <http://www.microchip.com/packaging> 查看 Microchip 封装规范 DS00000049。

6.1. CONFIG1

名称: CONFIG1
偏移量: 0x8007

配置字 1

位	15	14	13	12	11	10	9	8
访问			FCMEN	VDDAR	CSWEN			CLKOUTEN
复位				R/W	R/W	R/W		R/W
				1	1	1		1
位	7	6	5	4	3	2	1	0
访问			RSTOSC[2:0]				FEXTOSC[2:0]	
复位		R/W	R/W	R/W		R/W	R/W	R/W
	1	1	1	1		1	1	1

Bit 13 – FCMEN 故障保护时钟监视器使能位

值	说明
1	使能故障保护时钟监视器
0	禁止故障保护时钟监视器

Bit 12 – VDDAR V_{DD} 模拟范围校准选择

值	说明
1	校准内部模拟系统, 以便在 V _{DD} = 2.3V - 5.5V 范围内工作
0	校准内部模拟系统, 以便在 V _{DD} = 1.8V - 3.6V 范围内工作

Bit 11 – CSWEN 时钟切换使能

值	说明
1	允许写入 NOSC 和 NDIV
0	用户软件无法更改 NOSC 和 NDIV 位域

Bit 8 – CLKOUTEN 时钟输出使能

值	说明
1	禁止 CLKOUT 功能; CLKOUT 引脚为 I/O 功能
0	使能 CLKOUT 功能; CLKOUT 引脚为 F _{osc} /4 时钟

Bit 6:4 – RSTOSC[2:0] NOSC/COSC 的上电默认值位

选择用户软件使用的振荡器源。

值	说明
111	EXTOSC, 工作模式取决于 FEXTOSC 位
110	HFINTOSC = 1 MHz (FRQ = 4 MHz, CDIV = 4:1)
101	LFINTOSC
100	保留
011	保留
010	采用 4x PLL 的 EXTOSC, EXTOSC 根据 FEXTOSC 位操作
001	HFINTOSC = 16 MHz (FRQ = 16 MHz, CDIV = 1:1)
000	HFINTOSC = 32 MHz (FRQ = 32 MHz, CDIV = 1:1)

Bit 2:0 – FEXTOSC[2:0] 外部振荡器模式选择

值	说明
111	ECH (16 MHz 及更高值)
110	保留
101	ECL (低于 16 MHz)
100	振荡器未使能
011	保留
010	HS (晶振), 高于 4 MHz
001	XT (晶振), 100 kHz 至 4 MHz
000	LP (晶振), 32 kHz

6.2. CONFIG2

名称: CONFIG2
偏移量: 0x8008

配置字 2

位	15	14	13	12	11	10	9	8
			DEBUG	STVREN	PPS1WAY		BORV	DACAU TOEN
访问			R/W	R/W	R/W		R/W	R/W
复位			1	1	1		1	1
位	7	6	5	4	3	2	1	0
	BOREN[1:0]		LPBOREN			PWRTS[1:0]		MCLRE
访问	R/W	R/W	R/W			R/W	R/W	R/W
复位	1	1	1			1	1	1

Bit 13 – DEBUG 调试器使能⁽¹⁾

值	说明
1	禁止后台调试器
0	使能后台调试器

Bit 12 – STVREN 堆栈上溢/下溢复位使能

值	说明
1	堆栈上溢或下溢将导致复位
0	堆栈上溢或下溢不会导致复位

Bit 11 – PPS1WAY PPSLOCKED 一次置 1 使能

值	说明
1	PPSLOCKED 位只能在执行解锁序列之后置 1 一次；将 PPSLOCKED 置 1 后，可防止将来对 PPS 寄存器执行任何更改
0	PPSLOCKED 位可根据需要置 1 和清零（需要解锁序列）

Bit 9 – BORV 欠压复位 (BOR) 电压选择⁽²⁾

值	说明
1	欠压复位电压 (V_{BOR}) 设为 1.9V
0	欠压复位电压 (V_{BOR}) 设为 2.65V

Bit 8 – DACAU TOEN DAC 缓冲区自动范围选择使能

值	说明
1	DAC 缓冲区参考范围由 DACxCON 的 REFRNG 位确定
0	DAC 缓冲参考范围由模块硬件自动确定

Bit 7:6 – BOREN[1:0] 欠压复位 (BOR) 使能⁽³⁾

值	说明
11	使能欠压复位，忽略 SBOREN 位
10	运行时使能欠压复位，休眠时禁止；忽略 SBOREN 位
01	按照 SBOREN 使能欠压复位
00	禁止欠压复位

Bit 5 – LPBOREN 低功耗 BOR 使能

值	说明
1	禁止低功耗 BOR
0	使能低功耗 BOR

Bit 2:1 – PWRTS[1:0] 上电延时定时器 (PWRT) 选择

值	说明
11	禁止 PWRT
10	PWRT 设为 64 ms
01	PWRT 设为 16 ms
00	PWRT 设为 1 ms

Bit 0 – MCLRE 主复位 ($\overline{\text{MCLR}}$) 使能

值	条件	说明
x	如果 LVP = 1	$\overline{\text{MCLR}}$ 引脚为 $\overline{\text{MCLR}}$
1	如果 LVP = 0	$\overline{\text{MCLR}}$ 引脚为 $\overline{\text{MCLR}}$
0	如果 LVP = 0	$\overline{\text{MCLR}}$ 引脚功能为端口定义的功能

注:

1. $\overline{\text{DEBUG}}$ 位由器件开发工具 (包括调试器和编程器) 自动管理。对于正常的器件操作, 此位需保持为1。
2. 对于工作在 16 MHz 或高于 16 MHz 的情况, 建议选择更高电压。
3. 使能后, 欠压复位电压 (V_{BOR}) 通过 BORV 位置 1。

6.3. CONFIG3

名称: CONFIG3
偏移量: 0x8009

配置字 3

注: 该寄存器保留。

位	15	14	13	12	11	10	9	8
			WDTCCS[2:0]			WDTCWS[2:0]		
访问			R/W	R/W	R/W	R/W	R/W	R/W
复位			1	1	1	1	1	1
位	7	6	5	4	3	2	1	0
		WDTE[1:0]		WDTCPS[4:0]				
访问	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
复位	1	1	1	1	1	1	1	1

Bit 13:11 – WDTCCS[2:0] WDT 输入时钟选择器

值	条件	说明
x	WDTE = 00	这些位不起任何作用
111	WDTE ≠ 00	软件控制
110 至 010	WDTE ≠ 00	保留
001	WDTE ≠ 00	WDT 参考时钟为 31.25 kHz MFINTOSC
000	WDTE ≠ 00	WDT 参考时钟为 31.0 kHz LFINTOSC

Bit 10:8 – WDTCWS[2:0] WDT 窗口选择

WDTCWS	POR 时的 WDTCON1[WINDOW]			对窗口进行软件控制?	需要进行密钥访问?
	值	窗口延时时间百分比	窗口打开时间百分比		
111	111	n/a	100	是	否
110	110	n/a	100		
101	101	25	75		
100	100	37.5	62.5		
011	011	50	50		
010	010	62.5	37.5		
001	001	75	25		
000	000	87.5	12.5		

Bit 6:5 – WDTE[1:0] WDT 工作模式

值	说明
11	WDT 使能 (无论是否处于休眠状态); 忽略 WDTCON0 中的 SEN 位
10	WDT 在 Sleep = 0 时使能, 在 Sleep = 1 时暂停; 忽略 WDTCON0 中的 SEN 位
01	WDT 由 WDTCON0 中的 SEN 位使能/禁止
00	WDT 禁止, 忽略 WDTCON0 中的 SEN 位

Bit 4:0 – WDTCPS[4:0] WDT 周期选择

WDTCPS	POR 时的 WDTCON0[WDTPS]				对 WDTPS 进行软件控制?
	值	分频比	典型超时 ($F_{IN} = 31 \text{ kHz}$)		
11111	01011	1:65536	2^{16}	2s	是
11110 至 10011	11110 至 10011	1:32	2^5	1 ms	否

WDTCPS (续)

WDTCPS	POR 时的 WDTCON0[WDTPS]				对 WDTPS 进行软件控制?
	值	分频比		典型超时 ($F_{IN} = 31 \text{ kHz}$)	
10010	10010	1:8388608	2^{23}	256s	否
10001	10001	1:4194304	2^{22}	128s	否
10000	10000	1:2097152	2^{21}	64s	否
01111	01111	1:1048576	2^{20}	32s	否
01110	01110	1:524288	2^{19}	16s	否
01101	01101	1:262144	2^{18}	8s	否
01100	01100	1:131072	2^{17}	4s	否
01011	01011	1:65536	2^{16}	2s	否
01010	01010	1:32768	2^{15}	1s	否
01001	01001	1:16384	2^{14}	512 ms	否
01000	01000	1:8192	2^{13}	256 ms	否
00111	00111	1:4096	2^{12}	128 ms	否
00110	00110	1:2048	2^{11}	64 ms	否
00101	00101	1:1024	2^{10}	32 ms	否
00100	00100	1:512	2^9	16 ms	否
00011	00011	1:256	2^8	8 ms	否
00010	00010	1:128	2^7	4 ms	否
00001	00001	1:64	2^6	2 ms	否
00000	00000	1:32	2^5	1 ms	否

6.4. CONFIG4

名称: CONFIG4
偏移量: 0x800A

配置字 4

位	15	14	13	12	11	10	9	8
访问			LVP		WRTSAF		WRTC	WRTB
复位				R/W		R/W	R/W	R/W
位	7	6	5	4	3	2	1	0
访问	WRTAPP			SAFEN	BBEN		BBSIZE[2:0]	
复位	R/W			R/W	R/W	R/W	R/W	R/W
	1			1	1	1	1	1

Bit 13 – LVP 低电压编程使能⁽¹⁾

值	说明
1	使能低电压编程。 $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ 引脚功能是 $\overline{\text{MCLR}}$ 。MCLRE 位被忽略。
0	必须使用 $\overline{\text{MCLR}}/\text{V}_{\text{PP}}$ 的高电压 (HV) 进行编程

Bit 11 – WRTSAF 存储区闪存 (SAF) 写保护^(2,3)

值	说明
1	SAF 不受写保护
0	SAF 受写保护

Bit 9 – WRTC 配置寄存器写保护⁽²⁾

值	说明
1	配置寄存器不受写保护
0	配置寄存器受写保护

Bit 8 – WRTB 引导块写保护^(2,4)

值	说明
1	引导块不受写保护
0	引导块受写保护

Bit 7 – WRTAPP 应用程序块写保护⁽²⁾

值	说明
1	应用程序块不受写保护
0	应用程序块受写保护

Bit 4 – SAFEN 存储区闪存 (SAF) 使能⁽²⁾

值	说明
1	禁止 SAF
0	使能 SAF

Bit 3 – BBEN 引导块使能⁽²⁾

值	说明
1	禁止引导块
0	使能引导块

Bit 2:0 – BBSIZE[2:0] 引导块大小选择^(5,6)

表 6-2. 引导块大小

BBEN	BBSIZE	引导块结束地址	引导块大小 (字)		
			PIC16F131x3	PIC16F131x4	PIC16F131x5
1	xxx	-	-	-	-
0	111	01FFh	-	512	-
0	110	03FFh	-	1024	-
0	101	07FFh	- (6)	-	2048
0	100	0FFFh	- (6)	-	4096
0	011	1FFFh	-	- (6)	-
0	010	3FFFh	-	- (6)	-
0	001	3FFFh	-	- (6)	-
0	000	3FFFh	-	- (6)	-

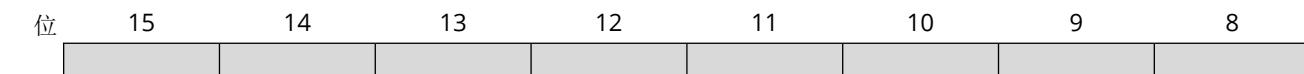
注:

1. 通过 LVP 编程接口工作时, 不能对 LVP 位进行写操作 (写为 0)。该规则的目的是防止用户在通过 LVP 模式编程时退出 LVP 模式, 或意外地从配置状态中删除 LVP 模式。
2. 保护功能通过 ICSP 或自写操作使能后, 只能通过批量擦除操作复位。
3. 仅在 SAFEN = 0 时适用。
4. 仅在 BBEN = 0 时适用。
5. BBSIZE[2:0]位只能在 BBEN = 1 时更改。当 BBEN = 0 后, BBSIZE[2:0]只能通过批量擦除操作进行更改。
6. 最大引导块大小是用户程序存储器大小的一半。如果选择的引导块大小超过器件程序存储器大小的一半, 则会默认设为最大引导块大小, 即 PFM 的一半。

6.5. CONFIG5

名称: CONFIG5
偏移量: 0x800B

配置字 5⁽¹⁾



访问
复位



访问 R/W
复位 1

Bit 0 - CP 用户闪存程序存储器 (PFM) 代码保护⁽²⁾

值	说明
1	禁止用户 PFM 代码保护
0	使能用户 PFM 代码保护

注:

1. 由于器件代码保护会立即生效, 因此该配置字可最后写入。
2. 代码保护使能后, 只能通过批量擦除操作取消。

Microchip 信息

商标

“Microchip”的名称和徽标组合、“M”徽标及其他名称、徽标和品牌均为 Microchip Technology Incorporated 或其关联公司和/或子公司在美国和/或其他国家或地区的注册商标或商标（“Microchip 商标”）。有关 Microchip 商标的信息，可访问 <https://www.microchip.com/en-us/about/legal-information/microchip-trademarks>。

ISBN: 979-8-3371-2458-2

法律声明

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

本出版物及其提供的信息仅适用于 Microchip 产品，包括设计、测试以及将 Microchip 产品集成到您的应用中。以其他任何方式使用这些信息都将被视为违反条款。本出版物中的器件应用信息仅为您提供便利，将来可能会发生更新。您须自行确保应用符合您的规范。如需额外的支持，请联系当地的 Microchip 销售办事处，或访问 www.microchip.com/en-us/support/design-help/client-support-services。

Microchip “按原样” 提供这些信息。Microchip 对这些信息不作任何明示或暗示、书面或口头、法定或其他形式的声明或担保，包括但不限于针对非侵权性、适销性和特定用途的适用性的暗示担保，或针对其使用情况、质量或性能的担保。

在任何情况下，对于因这些信息或使用这些信息而产生的任何间接的、特殊的、惩罚性的、偶然的或附带的损失、损害或任何类型的开销，Microchip 概不承担任何责任，即使 Microchip 已被告知可能发生损害或损害可以预见。在法律允许的最大范围内，对于因这些信息或使用这些信息而产生的所有索赔，Microchip 在任何情况下所承担的全部责任均不超出您为获得这些信息向 Microchip 直接支付的金额（如有）。如果将 Microchip 器件用于生命维持和/或生命安全应用，一切风险由买方自负。买方同意在由此引发任何一切损害、索赔、诉讼或费用时，会维护和保障 Microchip 免于承担法律责任。除非另外声明，在 Microchip 知识产权保护下，不得暗中或以其他方式转让任何许可证。

Microchip 器件代码保护功能

请注意以下有关 Microchip 产品代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术规范。
- Microchip 确信：在正常使用且符合工作规范的情况下，Microchip 系列产品非常安全。
- Microchip 注重并积极保护其知识产权。严禁任何试图破坏 Microchip 产品代码保护功能的行为，这种行为可能会违反《数字千年版权法案》（Digital Millennium Copyright Act）。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是“牢不可破”的。代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。