第 5 章 闪存编程

目录

本章包括下列主题：

注： 本系列参考手册章节旨在作为器件数据手册的补充资料，并非适用于所有的 PIC32 器件，适用与否取决于具体的器件型号。

请查询具体器件数据手册中“**闪存编程**”章节开始处的注释，以查看本文档是否支持您当前使用的器件。

器件数据手册和系列参考手册的各章节均可从 Microchip 网站：http://www.microchip.com 下载。

5.1 简介

本章介绍闪存的编程技术。PIC32 器件包含用于执行用户代码的内部闪存。用户可使用三种方法对该存储器进行编程：

- 运行时自编程（Run-Time Self Programming， RTSP）—— 由用户软件执行
- 在线串行编程（In-Circuit Serial Programming™， ICSP™）—— 通过使用器件的串行数据连接执行，编程比 RTSP 快得多
- 增强型联合测试小组编程（Enhanced Joint Test Action Group， EJTAG）—— 使用器件的 EJTAG 端口，通过支持 EJTAG 的编程器执行

本章将介绍RTSP技术。PIC32编程规范文档中介绍了ICSP和EJTAG方法，该文档可从Microchip网站（www.microchip.com）下载。

5.2 控制寄存器

闪存编程和擦除操作由以下非易失性存储器（Nonvolatile Memory，NVM）的控制寄存器进行控制：

- **NVMCON：编程控制寄存器 (1,2,3)**
- **NVMKEY：编程解锁寄存器**
- **NVMADDR：闪存地址寄存器 (1,2,3)**
- **NVMDATA：闪存程序数据寄存器**
- **NVMSRCADDR：源数据地址寄存器**

表 5-1 简要汇总了与闪存编程相关的寄存器。该汇总表之后列出了相应的寄存器，并且每个寄存器均附有详细的说明。

表 5-1： 闪存控制器 SFR 汇总

名称		Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
NVMCON(1,2,3)	31:24	—	—	—	—	—	—	—	—
	23:16	—	—	—	—	—	—	—	—
	15:8	WR	WREN	WRERR	LVDERR	LVDSTAT	—	—	—
	7:0	—	—	—	—	NVMOP<3:0>			
NVMKEY	31:24	NVMKEY<31:24>							
	23:16	NVMKEY<23:16>							
	15:8	NVMKEY<15:8>							
	7:0	NVMKEY<7:0>							
NVMADDR(1,2,3)	31:24	NVMADDR<31:24>							
	23:16	NVMADDR<23:16>							
	15:8	NVMADDR<15:8>							
	7:0	NVMADDR<7:0>							
NVMDATA	31:24	NVMDATA<31:24>							
	23:16	NVMDATA<23:16>							
	15:8	NVMDATA<15:8>							
	7:0	NVMDATA<7:0>							
NVMSRCADDR	31:24	NVMSRCADDR<31:24>							
	23:16	NVMSRCADDR<23:16>							
	15:8	NVMSRCADDR<15:8>							
	7:0	NVMSRCADDR<7:0>							

图注： — = 未实现，读为 0。地址偏移值用十六进制表示。

注 1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR（例如，NVMCONCLR）。向清零寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位清零。将忽略对清零寄存器的读操作。

2： 该寄存器具有关联的置 1 寄存器，位于 0x8 字节偏移处。这些置 1 寄存器的命名方式是在关联寄存器的名称末尾附加 SET（例如，NVMCONSET）。向置 1 寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位置 1。将忽略对置 1 寄存器的读操作。

3： 该寄存器具有关联的取反寄存器，位于 0xC 字节偏移处。这些取反寄存器的命名方式是在关联寄存器的名称末尾附加 INV（例如，NVMCONINV）。向取反寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位取反。将忽略对取反寄存器的读操作。

寄存器 5-1： NVMCON：编程控制寄存器 (1,2,3)

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 31							bit 24

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R-0	R-0	R-0	U-0	U-0	U-0
WR	WREN	WRERR(4)	LVDERR(4)	LVDSTAT(4)	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	NVMOP<3:0>			
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-16 **保留：**写入 0；忽略读操作

bit 15 **WR：**写控制位
当 WREN = 1，并随后执行解锁序列时，该位为可写
1 = 启动闪存操作。当操作完成时，由硬件清零该位。
0 = 闪存操作完成或无效

bit 14 **WREN：**写使能位
1 = 使能对 WR 位的写操作，并使能 LVD 电路
0 = 禁止对 WR 位的写操作，并禁止 LVD 电路

注：器件复位时，该寄存器中只有该位会复位。

bit 13 **WRERR：**写错误位 (4)
该位是只读位，由硬件自动置 1
1 = 编程或擦除序列未成功完成
0 = 编程或擦除序列正常完成

bit 12 **LVDERR：**低电压检测错误位 （LVD 电路必须使能）(4)
该位是只读位，由硬件自动置 1
1 = 检测到低电压 （如果 WRERR 置 1，则数据可能损坏）
0 = 电压在编程可接受的范围内

注 1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR （例如， NVMCONCLR）。向清零寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位清零。将忽略对清零寄存器的读操作。

2： 该寄存器具有关联的置 1 寄存器，位于 0x8 字节偏移处。这些置 1 寄存器的命名方式是在关联寄存器的名称末尾附加 SET （例如， NVMCONSET）。向置 1 寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位置 1。将忽略对置 1 寄存器的读操作。

3： 该寄存器具有关联的取反寄存器，位于 0xC 字节偏移处。这些取反寄存器的命名方式是在关联寄存器的名称末尾附加 INV（例如， NVMCONINV）。向取反寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位取反。将忽略对取反寄存器的读操作。

4： 清零方式为：设置 NVMOP == 0000b，并启动闪存操作 （即， WR）。

寄存器 5-1：　NVMCON：编程控制寄存器 (1,2,3) （续）

bit 11　**LVDSTAT**：低电压检测状态位 （LVD 电路必须使能）(4)
该位是只读位，由硬件自动置 1 和清零
1 = 低电压事件有效
0 = 低电压事件无效

bit 10-4　**保留**：写入 0；忽略读操作

bit 3-0　**NVMOP<3:0>**：NVM 操作位
当 WREN = 0 时，这些位可写。
1111 = 保留
•
•
•
0111 = 保留
0110 = 无操作
0101 = 程序闪存 （PFM）擦除操作：如果所有页均无写保护，则擦除 PFM
0100 = 页擦除操作：擦除通过 NVMADDR 选择的页 （如果无写保护）
0011 = 行编程操作：对通过 NVMADDR 选择的行进行编程 （如果无写保护）
0010 = 无操作
0001 = 字编程操作：对通过 NVMADDR 选择的字进行编程 （如果无写保护）
0000 = 无操作

注　1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR （例如， NVMCONCLR）。向清零寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位清零。将忽略对清零寄存器的读操作。

2： 该寄存器具有关联的置 1 寄存器，位于 0x8 字节偏移处。这些置 1 寄存器的命名方式是在关联寄存器的名称末尾附加 SET （例如， NVMCONSET）。向置 1 寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位置 1。将忽略对置 1 寄存器的读操作。

3： 该寄存器具有关联的取反寄存器，位于 0xC 字节偏移处。这些取反寄存器的命名方式是在关联寄存器的名称末尾附加 INV （例如， NVMCONINV）。向取反寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位取反。将忽略对取反寄存器的读操作。

4： 清零方式为：设置 NVMOP == 0000b，并启动闪存操作 （即， WR）。

寄存器 5-2： NVMKEY：编程解锁寄存器

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<31:24>							
bit 31							bit 24

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<23:16>							
bit 23							bit 16

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<15:8>							
bit 15							bit 8

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
NVMKEY<7:0>							
bit 7							bit 0

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-0 **NVMKEY<31:0>：**解锁寄存器位
这些位是只写位，在读取时读为 0

注： 该寄存器用作解锁序列的一部分，以防止对 PFM 的意外写操作。

寄存器 5-3： NVMADDR：闪存地址寄存器 (1,2,3)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMADDR<31:24>							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMADDR<23:16>							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMADDR<15:8>							
bit 15							bit 8

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-0 **NVMADDR<31:0>**：闪存地址位
批量 / 芯片 /PFM 擦除：地址会被忽略
页擦除：地址指示要擦除的页
行编程：地址指示要编程的行
字编程：地址指示要编程的字

注 1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR（例如， NVMCONCLR）。向清零寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位清零。将忽略对清零寄存器的读操作。

2： 该寄存器具有关联的置 1 寄存器，位于 0x8 字节偏移处。这些置 1 寄存器的命名方式是在关联寄存器的名称末尾附加 SET（例如， NVMCONSET）。向置 1 寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位置 1。将忽略对置 1 寄存器的读操作。

3： 该寄存器具有关联的取反寄存器，位于 0xC 字节偏移处。这些取反寄存器的命名方式是在关联寄存器的名称末尾附加 INV（例如， NVMCONINV）。向取反寄存器的任意位位置写入 1 时，会将关联寄存器中的有效位取反。将忽略对取反寄存器的读操作。

寄存器 5-4： NVMDATA：闪存程序数据寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMDATA<31:24>							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMDATA<23:16>							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMDATA<15:8>							
bit 15							bit 8

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-0 **NVMDATA<31:0>：**闪存编程数据位

注： 这些位只能通过上电复位 （Power-on Reset， POR）进行复位。

寄存器 5-5： NVMSRCADDR：源数据地址寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADDR<31:24>							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADDR<23:16>							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADDR<15:8>							
bit 15							bit 8

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

图注：			
R = 可读位	W = 可写位	P = 可编程位	r = 保留位
U = 未实现位	-n = POR 时的值：（0， 1， x = 未知）		

bit 31-0 **NVMSRCADDR<31:0>**：源数据地址位

当NVMOP<3:0>位（NVMCON<3:0>）设置为执行行编程时，会将数据的系统物理地址编程到闪存中。

5.2.1 NVMCON 寄存器

NVMCON 寄存器是闪存编程 / 擦除操作的控制寄存器。该寄存器选择将执行擦除还是编程操作，并用于启动编程或擦除周期。

寄存器 5-1 所示为 NVMCON 寄存器。NVMCON 的低字节用于配置将执行的 NVM 操作的类型。表 5-2 汇总了各种编程和擦除操作的 NVMCON 设置值。

表 5-2： NVMCON 寄存器值

操作	NVMCON 值
页擦除	0x8004
字编程	0x8001
行编程	0x8003
NOP	0x8000

5.2.2 NVMADDR 寄存器

NVM 地址寄存器用于选择闪存写操作的行、字写操作的地址单元，以及闪存擦除操作的页地址。

注： NVM 地址寄存器必须装载闪存存储器的物理地址，而不是虚拟地址。

5.2.3 NVMKEY 寄存器

NVMKEY 是一个只写寄存器，用于防止闪存或 EEPROM 存储器的误写 / 误擦除操作。要启动编程或擦除序列，必须严格按如下顺序执行步骤：

1. 将 0xAA996655 写入 NVMKEY。
2. 将 0x556699AA 写入 NVMKEY。

执行该序列之后，仅允许外设总线上的下一个事务写 NVMCON 寄存器。在多数情况下，用户只需将NVMCON寄存器中的WR位置1，就可以启动编程或擦除周期。在解锁序列期间应禁止中断。

5.2.4 NVMSRCADDR 寄存器

NVM 源地址寄存器用于在 SRAM 中选择用于执行行编程操作的源数据缓冲区地址。

注： 地址必须字对齐。

5.3 运行时自编程 （RTSP）工作原理

运行时自编程 （RTSP）允许用户代码修改闪存程序存储器的内容。器件闪存分为两个逻辑闪存分区：程序闪存存储器（PFM）和引导闪存存储器（BFM）。引导闪存存储器的最后页包含调试页，保留供调试器工具在进行调试时使用。

PIC32 器件的程序闪存阵列由一系列的行构成。一行包含 128 个 32 位指令字或 512 字节。一组 8 行组成一页；因此，它包含 8 × 512 = 4096 字节或 1024 个指令字。闪存页是在单次中可擦除的最小存储器单元。程序闪存阵列可以使用两种方式进行编程：

- 行编程，每次 128 个指令字。
- 字编程，每次 1 个指令字。

当从程序闪存存储器执行指令（取指）时进行 RTSP 操作， CPU 会暂停 （等待）直到编程操作完成为止。这段时间内，CPU 不会执行任何指令，也不会响应任何中断。如果编程周期内发生了任何中断，中断将保持在等待处理状态，直到编程周期完成为止。

当从 RAM 存储器执行指令（取指）时进行 RTSP 操作，CPU 在编程操作期间仍可继续执行指令并响应中断。请注意，任何计划在 RTSP 操作期间执行的可执行代码必须存放于 RAM 存储器中。这包括相关中断向量和中断服务程序指令。

注：	对于闪存擦除和写操作，需要达到它的最低 VDD 要求。更多详细信息，请参见具体器件数据手册。

5.4 锁定特性

5.4.1 NVMWREN

器件中存在许多用于确保不会对程序闪存执行意外写操作的机制。除非软件要对程序闪存执行写操作，否则 WREN 位 （NVMCON<14>）应为 0。当 WREN = 1 时，闪存写控制位 WR （NVMCON<15>）可写，并且闪存 LVD 电路会使能。

5.4.2 NVMKEY

除了通过 WREN 位提供的写保护之外，还需要先执行解锁序列，然后 WR（NVMCOM<15>）位才能置 1。如果 WR 位未在下一个外设总线事务 （读操作或写操作）中置 1，则 WR 会被锁定，必须重新启动解锁序列。

5.4.3 解锁序列

要解锁闪存操作，必须严格按照顺序执行下面的步骤 4 至步骤 9：如果未严格按照该序列执行，则 WR 不会置 1。

1. 暂停或禁止可访问外设总线和中断解锁序列的所有发起器，例如 DMA 和中断。
2. 将 WREN 位 （NVMCON<14>）置 1，以允许写入 WR，并使用单条存储指令将 NVMOP<3:0> 位 （NVMCON<3:0>）设置为所需的操作。
3. 等待 LVD 启动。
4. 向 CPU 寄存器 X 中装入 0xAA996655。
5. 向 CPU 寄存器 Y 中装入 0x556699AA。
6. 向 CPU 寄存器 Z 中装入 0x00008000。
7. 将 CPU 寄存器 X 存储到 NVMKEY 中。
8. 将 CPU 寄存器 Y 存储到 NVMKEY 中。
9. 将 CPU 寄存器 Z 存储到 NVMCONSET 中。
10. 等待 WR 位 （NVMCON<15>）清零。
11. 清零 WREN 位 （NVMCON<14>）。
12. 检查 WRERR（NVMCON<13>）和 LVDERR（NVMCON<12>）位，以确保编程 / 擦除序列成功完成。

当 WR 位置 1 时，编程 / 擦除序列会启动，在该序列期间， CPU 无法从闪存中执行。

例 5-1：　　解锁示例

```
unsigned int NVMUnlock (unsigned int nvmop)
{
    unsigned int status;

    // Suspend or Disable all Interrupts
    asm volatile ("di %0" : "=r" (status));

    // Enable Flash Write/Erase Operations and Select
    // Flash operation to perform
    NVMCON = nvmop;

    // Write Keys
    NVMKEY = 0xAA996655;
    NVMKEY = 0x556699AA;

    // Start the operation using the Set Register
    NVMCONSET = 0x8000;

    // Wait for operation to complete
    while (NVMCON & 0x8000);

    // Restore Interrupts
    if (status & 0x00000001)
      asm volatile ("ei");
    else
      asm volatile ("di");


    // Disable NVM write enable
    NVMCONCLR = 0x0004000;

    // Return WRERR and LVDERR Error Status Bits
    return (NVMCON & 0x3000);
}
```

5.5 字编程序列

在单次操作中可以编程的最小数据块为 32 位字。在启动编程序列之前，必须先将要编程的数据写入 NVMDATA 寄存器，并且必须将字的地址装入 NVMADDR 寄存器。然后，位于 NVMADDR 所指向单元中的指令字会被编程。

编程序列包含以下步骤：

1. 将要编程的 32 位数据写入 NVMDATA 寄存器。
2. 在 NVMADDR 寄存器中装入要编程的地址。
3. 使用字编程命令运行解锁序列 （见**第 5.4.3 节 “ 解锁序列 ”**）。

编程序列完成， WR 位 （NVMCON<15>）由硬件清零。

例 5-2： 字编程示例

```
unsigned int NVMWriteWord (void* address, unsigned int data)
{
    unsigned int res;

    // Load data into NVMDATA register
    NVMDATA = data;

    // Load address to program into NVMADDR register
    NVMADDR = (unsigned int) address;

    // Unlock and Write Word
    res = NVMUnlock (0x4001);

    // Return Result
    return res;

}
```

5.6 行编程序列

可编程的最大数据块为 1 行，等于 512 字节的数据。数据行必须先装入 SRAM 的缓冲区中。然后， NVMADDR 寄存器指向闪存控制器将开始编程的数据行的起始地址。

注：	控制器会忽略行以下的地址位，总是从行起始位置处开始编程。

行编程序列包含以下步骤：

1. 将要编程的整行数据写入系统 SRAM。源地址必须字对齐。
2. 使用要编程的闪存行的起始地址设置 NVMADDR 寄存器。
3. 使用来自步骤 1 的物理源地址设置 NVMSRCADDR 寄存器。
4. 使用行编程命令运行解锁序列 （见**第 5.4.3 节 “ 解锁序列 ”**）。
5. 编程序列完成， WR 位 （NVMCON<15>）由硬件清零。

例 5-3： 行编程示例

```
unsigned int NVMWriteRow (void* address, void* data)
{
    unsigned int res;

    // Set NVMADDR to Start Address of row to program
    NVMADDR = (unsigned int) address;

    // Set NVMSRCADDR to the SRAM data buffer Address
    NVMSRCADDR = (unsigned int) data;

    // Unlock and Write Row
    res = NVMUnlock(0x4003);

    // Return Result
    return res;

}
```

5.7 页擦除序列

页擦除对 PFM 或 BFM 的单个页（等于 4096 字节）执行擦除操作。要擦除的页使用 NVMADDR 寄存器进行选择。

注： 在页选择中，地址的低位会被忽略。

只有关联的页写保护未使能时，才能擦除闪存页。

- 所有 BFM 页都会受引导写保护配置位影响。
- PFM 页会受程序闪存写保护配置位影响。

如果处于任务模式，禁止应用程序从擦除页中执行代码。

页擦除序列包含以下步骤：

1. 使用要擦除页的地址设置 NVMADDR 寄存器。
2. 使用所需的擦除命令运行解锁序列 （见**第 5.4.3 节 “ 解锁序列 ”**）。
3. 擦除序列完成， WR 位 （NVMCON<15>）由硬件清零。

例 5-4： 页擦除示例

```
unsigned int NVMErasePage(void* address)
{
    unsigned int res;

    // Set NVMADDR to the Start Address of page to erase
    NVMADDR = (unsigned int) address;

    // Unlock and Erase Page
    res = NVMUnlock(0x4004);

    // Return Result
    return res;

}
```

5.8 程序闪存存储器擦除序列

可以擦除整个 PFM 区域。该模式会将引导闪存保持原样，旨在供可现场升级的器件使用。

如果程序闪存中的所有页均无写保护，则可以擦除闪存。

注： 禁止应用程序从 PFM 地址范围中执行代码。

PFM 擦除序列包含以下步骤：

1. 使用程序闪存存储器擦除命令运行解锁序列 （见**第 5.4.3 节 “ 解锁序列 ”**）。
2. 擦除序列完成， WR 位 （NVMCON<15>）由硬件清零。

例 5-5： 程序闪存擦除示例

```
unsigned int NVMErasePFM(void)
{
    unsigned int res;

    // Unlock and Erase Program Flash
    res = NVMUnlock(0x4005);

    // Return Result
    return res;

}
```

5.9 节能和调试模式下的操作

5.9.1 休眠模式下的操作

当 PIC32 器件进入休眠模式时，系统时钟被禁止。闪存控制器在休眠模式下不工作。如果在 NVM 操作正在进行时要进入休眠模式，那么只有 NVM 操作完成之后，器件才会进入休眠模式。

5.9.2 空闲模式下的操作

在编程操作正在进行时，空闲模式对于闪存控制器模块没有任何作用。CPU 会继续被占用，直到编程操作完成。

5.9.3 调试模式下的操作

闪存控制器未提供调试冻结功能，因此在编程操作正在进行时，它对于闪存控制器模块没有任何作用。CPU 会继续被占用，直到编程操作完成。中断正常的编程序列可能会导致器件锁死。该情形的唯一例外是 NVMKEY 解锁序列，该序列可在调试模式下暂停，让用户可以单步执行解锁序列。

5.10 各种复位的影响

5.10.1 器件复位

在器件复位时，只有 WREN 和 LVDSTAT 的 NVMCON 位会复位。所有其他 SFR 位只能通过 POR 复位。但是， NVMKEY 的状态则通过器件复位进行复位。

5.10.2 上电复位

在发生 POR 时，所有闪存控制器寄存器会被强制设为它们的复位状态。

5.10.3 看门狗定时器复位

在发生看门狗定时器复位时，所有闪存控制器寄存器不变。

5.11 中断

闪存控制器可以产生反映在编程操作期间所发生事件的中断。它可以产生以下中断：

- 闪存控制事件中断 FCEIF （IFS1<24>）

中断标志必须用软件清零。闪存控制器可以通过以下相应的闪存控制器中断允许位使能为中断源：

- FCEIE （IE1<24>）

此外，还必须配置中断优先级位和中断子优先级位：

- FCEIP （IPC11<2:0>）和 FCEIS （IPC11<1:0>）

详情请参见 《PIC32 系列参考手册》中的**第 8 章 “ 中断 ”**（DS61108）。

5.11.1 中断配置

闪存控制器模块具有一个专用中断标志位 FCEIF 和一个相应的中断允许 / 屏蔽位 FCEIE。

这两个位用于决定中断源和使能 / 禁止各个中断源。请注意，特定闪存控制器模块的所有中断源仅共用一个中断向量。

此外， FCEIF 位是否置 1 与相应允许位的状态无关，如果需要，可以通过软件查询 FCEIF 位。

FCEIE 位用于定义在相应 FCEIF 位置 1 时，向量中断控制器（Vector Interrupt Controller，VIC）的行为。当相应的 FCEIE 位清零时， VIC 模块不会为事件产生 CPU 中断。如果 FCEIE 位置 1，则 VIC 模块会在相应的 FCEIF 位置 1 时向 CPU 产生中断（受以下段落中概述的优先级和子优先级制约）。

处理特定中断的用户软件程序需要负责在服务程序完成之前清零相应的中断标志位。

闪存控制器模块的优先级可以使用 FCEIP<2:0> 位独立设置。该优先级定义为中断源分配的优先级组。优先级组的值的范围为从 7 （最高优先级）到 0 （不产生中断）。较高优先级组中的中断会抢占正在处理、但优先级较低的中断。

子优先级位用于设置中断源在优先级组中的优先级。子优先级 FCEIS<1:0> 值的范围为从 3 （最高优先级）到 0 （最低优先级）。处于相同优先级组，但具有更高子优先级值的中断不会抢占子优先级较低、但正在进行的中断。

优先级组和子优先级位让多个中断源可以共用相同的优先级和子优先级。如果在该配置下同时发生若干个中断，则中断源在优先级 / 子优先级组对中的自然顺序将决定所产生的中断。自然优先级基于中断源的向量编号。向量编号越小，中断的自然优先级就越高。在当前中断的中断标志清零之后，所有不按照自然顺序执行的中断会基于优先级、子优先级和自然顺序产生相应的中断。

产生允许的中断之后，CPU 会跳转到为该中断分配的向量处。该中断的向量编号与自然顺序编号相同。然后，CPU 会在向量地址处开始执行代码。该向量地址处的用户代码应执行特定于应用程序的操作、清零 FCEIF 中断标志，然后退出。更多关于中断和向量地址表的详细信息，请参见《PIC32 系列参考手册》中的**第 8 章 “ 中断 ”**（DS61108）和具体器件数据手册中的**“ 中断控制器 ”** 章节。

5.12 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与闪存编程相关的应用笔记包括：

标题	应用笔记编号
目前没有相关的应用笔记。	N/A

注： 如需获取更多 PIC32 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

5.13 版本历史

版本 A （2007 年 9 月）

这是本文档的初始版本。

版本 B （2007 年 10 月）

更新了文档 （删除了 “ 机密 ” 状态）。

版本 C （2008 年 4 月）

将状态修改为 “ 初稿 ” ；将 U-0 修改为 r-x。

版本 D （2008 年 6 月）

修改了寄存器 5-1 的 bit 14 NVMWREN ；在寄存器 5-12 至 5-14 中增加了脚注 1 ；在第 5.3 节中增加了注释；修改了第 5.4.1 节；修改了例 5-1 ；将保留位的 “ 保持为 ” 更改为 “ 写入 ”。

版本 E （2010 年 12 月）

此版本包括以下更新：

- 对全文的内容和格式做了少量更新
- 添加注 1、 2 和 3，用以描述下列清零、置 1 和取反寄存器：
  - 表 5-1：闪存控制器 SFR 汇总
  - 寄存器 5-1：NVMCON：编程控制寄存器 (1,2,3)
  - 寄存器 5-3：NVMADDR：闪存地址寄存器 (1,2,3)
- 移除所有清零、置 1 和取反寄存器说明
- 移除所有中断寄存器说明
- 在全文中重命名以下 NVMCON 寄存器位名称：
  - NVMWR 重命名为 WR
  - NVMWREN 重命名为 WREN
  - NVMERR 重命名为 WRERR
- 更新**第 5.3 节 “ 运行时自编程 （RTSP）工作原理 ”** 中的第三节，并新增第四节。
- 更新解锁闪存操作序列，添加步骤 3 （见**第 5.4.3 节 “ 解锁序列 ”**）
- 更新了解锁示例中的代码 （见例 5-1）
- 移除表 5-3

请注意以下有关 Microchip 器件代码保护功能的要点：

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

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

ISBN: 978-1-60932-848-1

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

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham 的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2002 认证。公司在 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品方面的质量体系流程均符合 ISO/TS-16949:2002。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技术支持：
http://support.microchip.com
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax:1-678-957-1455

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

克里夫兰 Cleveland
Independence, OH
Tel: 1-216-447-0464
Fax: 1-216-447-0643

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Farmington Hills, MI
Tel: 1-248-538-2250
Fax: 1-248-538-2260

科科莫 Kokomo
Kokomo, IN
Tel: 1-765-864-8360
Fax: 1-765-864-8387

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608

圣克拉拉 Santa Clara
Santa Clara, CA
Tel: 1-408-961-6444
Fax: 1-408-961-6445

加拿大多伦多 Toronto
Mississauga, Ontario,
Canada
Tel: 1-905-673-0699
Fax: 1-905-673-6509

亚太地区

亚太总部 Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon
Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 北京
Tel: 86-10-8528-2100
Fax: 86-10-8528-2104

中国 – 成都
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

中国 – 重庆
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

中国 – 香港特别行政区
Tel: 852-2401-1200
Fax: 852-2401-3431

中国 – 南京
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

中国 – 青岛
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

中国 – 上海
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

中国 – 沈阳
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

中国 – 深圳
Tel: 86-755-8203-2660
Fax: 86-755-8203-1760

中国 – 武汉
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

中国 – 西安
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

中国 – 厦门
Tel: 86-592-238-8138
Fax: 86-592-238-8130

中国 – 珠海
Tel: 86-756-321-0040
Fax: 86-756-321-0049

台湾地区 – 高雄
Tel: 886-7-213-7830
Fax: 886-7-330-9305

台湾地区 – 台北
Tel: 886-2-2500-6610
Fax: 886-2-2508-0102

亚太地区

台湾地区 – 新竹
Tel: 886-3-6578-300
Fax: 886-3-6578-370

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

印度 India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

印度 India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

印度 India - Pune
Tel: 91-20-2566-1512
Fax: 91-20-2566-1513

日本 Japan - Yokohama
Tel: 81-45-471- 6166
Fax: 81-45-471-6122

韩国 Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

韩国 Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 或
82-2-558-5934

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

菲律宾 Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

新加坡 Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

泰国 Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

欧洲

奥地利 Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark-Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

意大利 Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

西班牙 Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

英国 UK - Wokingham
Tel: 44-118-921-5869
Fax: 44-118-921-5820

08/04/10