

---

**具有高速ADC、运算放大器、比较器和  
高分辨率PWM的16位数字信号控制器**

---

**工作条件**

- 3.0V至3.6V: -40°C至+125°C, DC至100 MHz

**高性能16位DSP RISC CPU**

- 16位宽数据路径
- 高效代码型（C和汇编）架构
- 40位宽累加器
- 带双数据取操作的单周期（MAC/MPY）
- 单周期混合符号乘法：
  - 32位乘法支持
- 快速的6周期除法
- 零开销循环

**高速PWM**

- 4对PWM
- 最佳PWM分辨率为250 ps
- 上升沿和下降沿死区
- 死区补偿
- 用于高频操作的时钟斩波
- 为以下各种应用提供PWM支持：
  - 直流/直流、交流/直流、逆变器、PFC和照明
  - BLDC、PMSM、ACIM和SRM电机
- 故障输入和限流输入
- 用于ADC触发的灵活触发配置

**高速模数转换器**

- 12位分辨率
- 2个专用SAR ADC内核和1个共用SAR ADC内核
- 每个内核的转换速率最高为3.5 Msps
- 每个模拟通道都具有专用的结果缓冲区
- 灵活、独立的ADC触发源
- 4个数字比较器
- 4个过采样滤波器

**单片机特性**

- 28到48引脚的小引脚数封装，包括小至4x4 mm的UQFN
- 高I/O灌电流/拉电流
- I/O引脚上具有边沿或电平变化通知中断
- 外设引脚选择（Peripheral Pin Select, PPS）可重映射的引脚
- 最大64 KB的闪存：
  - 耐擦/写次数达10,000次
  - 数据保存时间至少20年
  - 可在软件控制下自编程
  - 可编程代码保护
  - 纠错码（Error Code Correction, ECC）
  - ICSP™写禁止
- 8 KB SRAM存储器：
  - SRAM存储器内置自检（Memory Built-In Self-Test, MBIST）
- 多个中断向量，带有独立的可编程优先级功能
- 用于快速中断处理的4组中断现场保护寄存器，包括累加器和状态寄存器
- 4个外部中断引脚
- 看门狗定时器（Watchdog Timer, WDT）
- 窗口程序监控定时器（Deadman Timer, DMT）
- 具有专用备用振荡器的故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM）
- 可选振荡器选项，包括：
  - 32 kHz低功耗RC（Low-Power RC, LPRC）振荡器
  - 高精度8 MHz内部快速RC（Fast RC, FRC）振荡器
  - 主高速晶振/谐振器或外部时钟
  - 主PLL，可由FRC或晶振提供时钟
  - 用于PWM和ADC的附属PLL（Auxiliary PLL, APLL）
- 低功耗管理模式（休眠和空闲）
- 上电复位和欠压复位
- 片内无电容稳压器
- 256字节的可一次性编程（One-Time-Programmable, OTP）存储器

# dsPIC33CK64MP105系列

---

## 外设特性

- 3个4线SPI模块（最高50 Mbps）：
  - 16字节FIFO
  - 可变宽度
  - I<sup>2</sup>S模式
- 2个I<sup>2</sup>C主器件和从器件，支持地址掩码和IPMI
- 3个协议UART，支持自动处理以下协议：
  - LIN 2.2
  - DMX
  - 智能卡（ISO 7816）
  - IrDA<sup>®</sup>
- 2个SENT模块
- 1个专用的16位定时器/计数器
- 4个单输出捕捉/比较/PWM/定时器（SCCP）模块：
  - 灵活配置为PWM、输入捕捉、输出比较或定时器
  - 每个模块中有2个16位定时器或1个32位定时器
  - PWM分辨率低至4 ns
  - 单个PWM输出
- 1个多输出捕捉/比较/PWM/定时器（MCCP）模块：
  - 灵活配置为PWM、输入捕捉、输出比较或定时器
  - 每个模块中有2个16位定时器或1个32位定时器
  - PWM分辨率低至4 ns
  - 最多6个PWM输出
  - 可编程死区
  - 自动关断
- 2个正交编码器接口（Quadrature Encoder Interface, QEI）：
  - 4个输入：A相、B相、归位和索引
- 参考时钟输出（REFCLKO）
- 4个可配置逻辑单元（Configurable Logic Cell, CLC），其内部连接到某些外设和PPS
- 4通道硬件DMA
- 32位CRC计算模块
- 外设触发信号发生器（Peripheral Trigger Generator, PTG）
  - 可为其他外设模块提供16个触发源
  - 基于与CPU无关的状态机的指令序列器

## 模拟特性

- 3个快速模拟比较器，带输入多路选择
- 3个运算放大器
- 3个12位PDM DAC及斜率补偿
- 1个输出DAC缓冲器

## 认证和B类支持

- AEC-Q100 REVG（1级：-40°C至+125°C）
- B类安全库，IEC 60730

## 调试功能

- 3个编程和调试接口：
  - 双线 ICSP<sup>™</sup> 接口，支持非侵入式访问和与应用的实时数据交换
- 3个复杂断点，5个简单断点
- 符合IEEE标准1149.2的（JTAG）边界扫描

dsPIC33CK64MP105产品系列

表1列出了每个器件的器件名称、引脚数、存储器大小和外设可用性。以下几页显示了相应的引脚分配图。

表1: dsPIC33CK64MP105系列

产品	引脚数	程序存储器	数据存储器	通用 I/O/PPS	高速 PWM (发生器)	12位 ADC (外部通道数)	可重映射的外设						运算放大器	比较器	12位 DAC	I <sup>2</sup> C	QEI	SENT	32位 CRC	DMA 通道	封装
							专用 16 位定时器	UART	MCCP <sup>(1)</sup>	SCCP <sup>(2)</sup>	CLC	SPI/I <sup>2</sup> S									
dsPIC33CK32MP102	28	32K	8K	21/16	4	12	1	3	1	4	4	3	2	3	3	2	2	2	1	4	SSOP/UQFN
dsPIC33CK32MP103	36	32K	8K	27/22	4	16	1	3	1	4	4	3	3	3	3	2	2	2	1	4	UQFN
dsPIC33CK32MP105	48	32K	8K	39/34	4	19	1	3	1	4	4	3	3	3	3	2	2	2	1	4	UQFN/TQFP
dsPIC33CK64MP102	28	64K	8K	21/16	4	12	1	3	1	4	4	3	2	3	3	2	2	2	1	4	SSOP/UQFN
dsPIC33CK64MP103	36	64K	8K	27/22	4	16	1	3	1	4	4	3	3	3	3	2	2	2	1	4	UQFN
dsPIC33CK64MP105	48	64K	8K	39/34	4	19	1	3	1	4	4	3	3	3	3	2	2	2	1	4	UQFN/TQFP

注 1: MCCP可配置为1个PWM（最多有6个输出）、输入捕捉、输出比较、2个16位定时器或1个32位定时器。

2: SCCP可配置为1个PWM（有1个输出）、输入捕捉、输出比较、2个16位定时器或1个32位定时器。

# dsPIC33CK64MP105系列

## 引脚分配图

28引脚SSOP<sup>(1)</sup>

RA1

RA2

RA3

RA4

AVDD

AVSS

VDD

VSS

RB0

RB1<sup>(2,4)</sup>

RB2

RB3

RB4

RB5

1

2

3

4

5

6

7

8

9

10

11

12

13

14

dsPIC33CKXXMP102

28

27

26

25

24

23

22

21

20

19

18

17

16

15

RA0

MCLR

RB15

RB14

RB13

RB12

RB11

RB10<sup>(3)</sup>

VDD

VSS

RB9

RB8

RB7

RB6

■ = 5V耐压

注 1:

有关引脚功能的完整说明, 请参见表2。

注 2:

引脚的电流驱动能力增强。有关详细信息, 请参见第31.0节“电气特性”。

注 3:

在编程期间或通过配置位使能JTAG时, 会将一个上拉电阻连接到该引脚。

注 4:

该引脚在编程期间进行翻转。

表2: 28引脚SSOP引脚功能完整说明

引脚 编号	功能 <sup>(1)</sup>	引脚 编号	功能 <sup>(1)</sup>
1	OA1IN-/ANA1/RA1	15	PGC3/RP38/SCL2/RB6
2	OA1IN+/AN9/RA2	16	TDO/AN2/CMP3A/RP39/RB7
3	DACOUT/AN3/CMP1C/RA3	17	PGD1/AN10/RP40/SCL1/RB8
4	AN4/CMP3B/IBIAS3/RA4	18	PGC1/AN11/RP41/SDA1/RB9
5	AVDD	19	Vss
6	AVSS	20	VDD
7	VDD	21	TMS/RP42/PWM3H/RB10 <sup>(3)</sup>
8	VSS	22	TCK/RP43/PWM3L/RB11
9	OSCI/CLKI/AN5/RP32/RB0	23	TDI/RP44/PWM2H/RB12
10	OSCO/CLKO/AN6/RP33/RB1 <sup>(2,4)</sup>	24	RP45/PWM2L/RB13
11	OA2OUT/AN1/AN7/ANA0/CMP1D/CMP2A/CMP3D/RP34/INT0/ RB2	25	RP46/PWM1H/RB14
12	PGD2/OA2IN-/AN8/RP35/RB3	26	RP47/PWM1L/RB15
13	PGC2/OA2IN+/RP36/RB4	27	MCLR
14	PGD3/RP37/SDA2/RB5	28	OA1OUT/AN0/CMP1A/IBIAS0/RA0

注 1: RPn表示可重映射外设功能。

注 2: 引脚的电流驱动能力增强。有关详细信息, 请参见第31.0节“电气特性”。

注 3: 在编程期间或通过配置位使能JTAG时, 会将一个上拉电阻连接到该引脚。

注 4: 该引脚在编程期间进行翻转。

DS70005363D\_CN 第4页

© 2019-2020 Microchip Technology Inc.

# dsPIC33CK64MP105系列

引脚分配图（续）

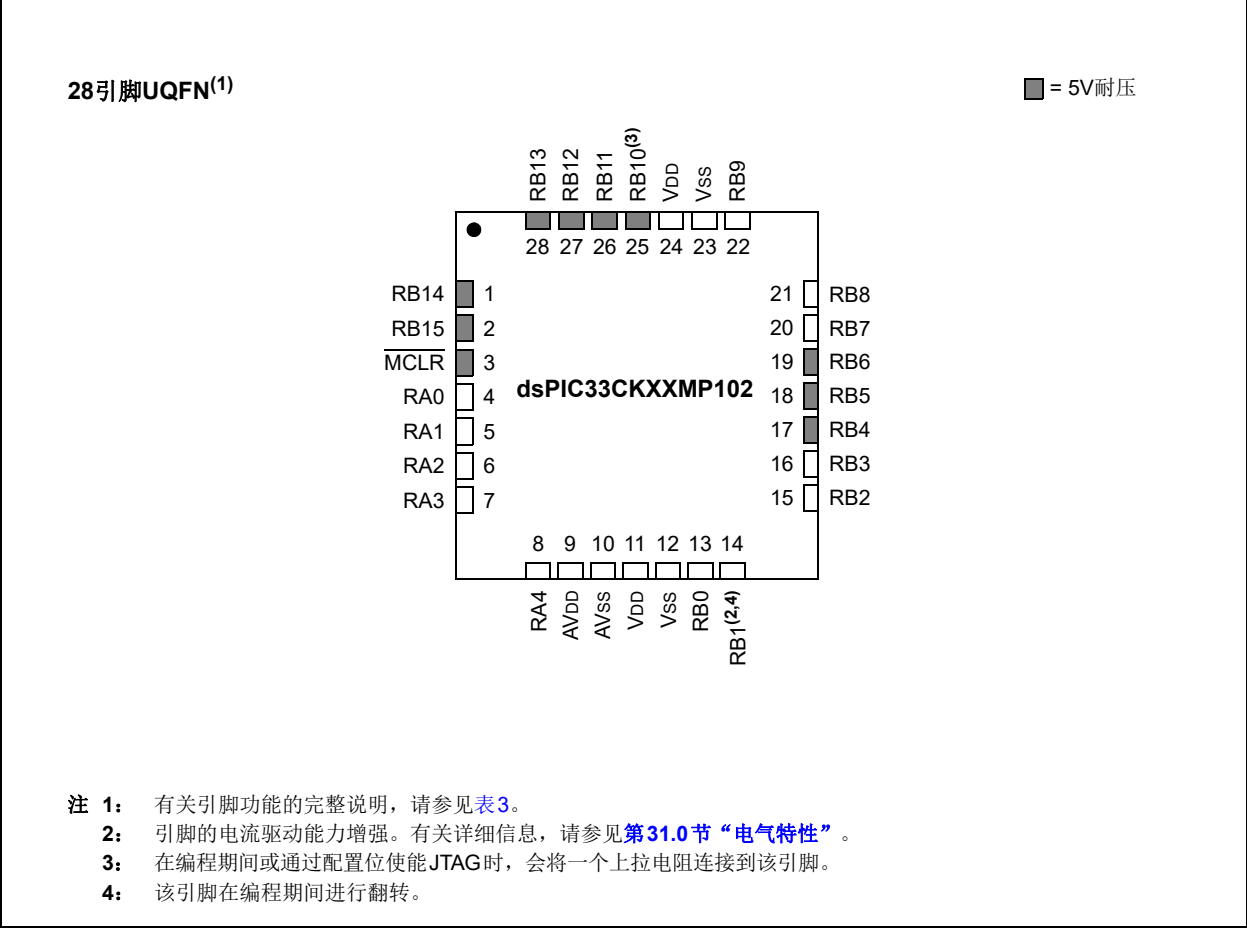


表3: 28引脚UQFN引脚功能完整说明

引脚 编号	功能 <sup>(1)</sup>	引脚 编号	功能 <sup>(1)</sup>
1	RP46/PWM1H/RB14	15	OA2OUT/AN1/AN7/ANA0/CMP1D/CMP2D/CMP3D/RP34/INT0/RB2
2	RP47/PWM1L/RB15	16	PGD2/OA2IN-/AN8/RP35/RB3
3	MCLR	17	PGC2/OA2IN+/RP36/RB4
4	OA1OUT/AN0/CMP1A/IBIAS0/RA0	18	PGD3/RP37/SDA2/RB5
5	OA1IN-/ANA1/RA1	19	PGC3/RP38/SCL2/RB6
6	OA1IN+/AN9/RA2	20	TDO/AN2/CMP3A/RP39/RB7
7	DACOUT/AN3/CMP1C/RA3	21	PGD1/AN10/RP40/SCL1/RB8
8	AN4/CMP3B/IBIAS3/RA4	22	PGC1/AN11/RP41/SDA1/RB9
9	AVDD	23	VSS
10	AVSS	24	VDD
11	VDD	25	TMS/RP42/PWM3H/RB10 <sup>(3)</sup>
12	VSS	26	TCK/RP43/PWM3L/RB11
13	OSCI/CLKI/AN5/RP32/RB0	27	TDI/RP44/PWM2H/RB12
14	OSCO/CLKO/AN6/RP33/RB1 <sup>(2,4)</sup>	28	RP45/PWM2L/RB13

- 注 1: RPn 表示可重映射外设功能。  
注 2: 引脚的电流驱动能力增强。有关详细信息, 请参见第31.0节“电气特性”。  
注 3: 在编程期间或通过配置位使能JTAG时, 会将一个上拉电阻连接到该引脚。  
注 4: 该引脚在编程期间进行翻转。

# dsPIC33CK64MP105系列

## 引脚分配图（续）

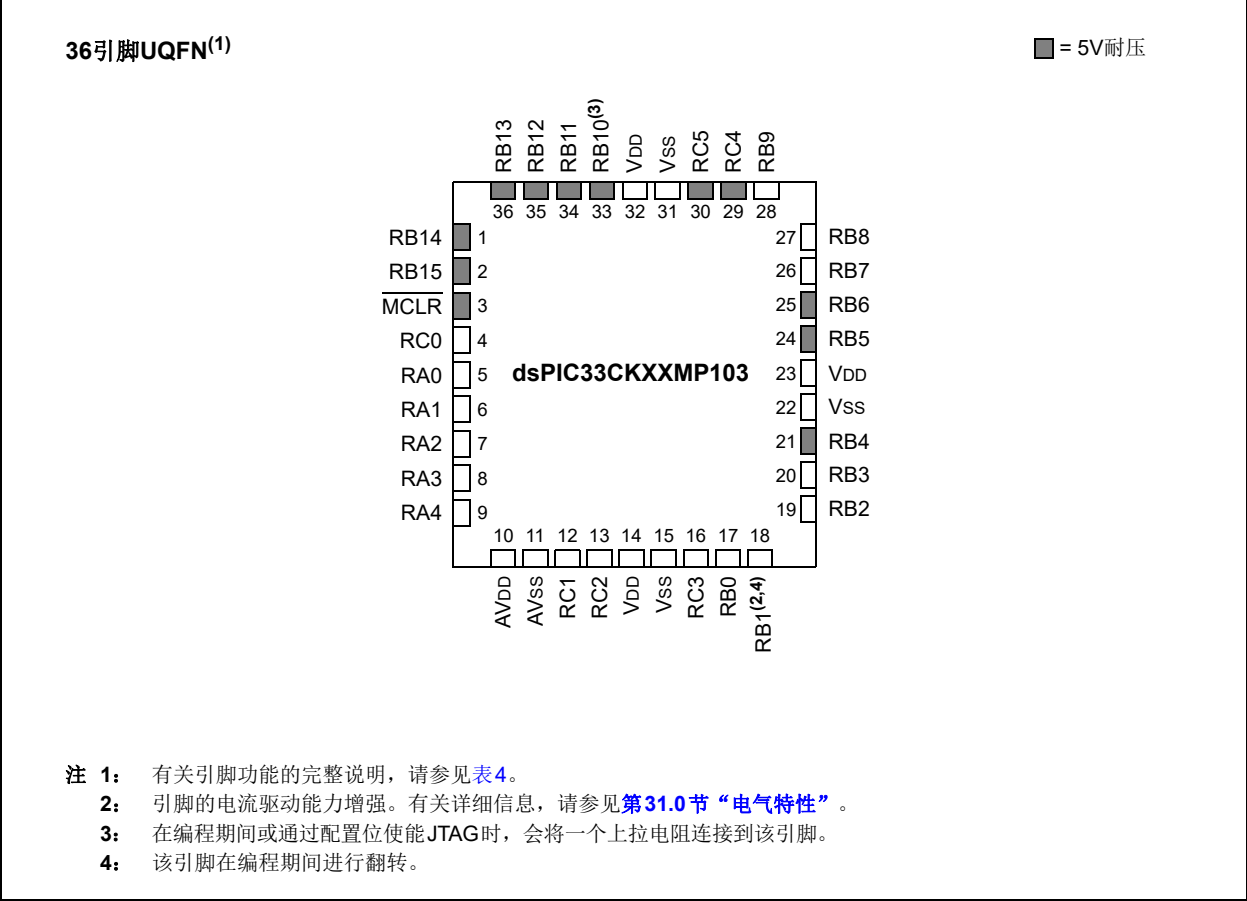
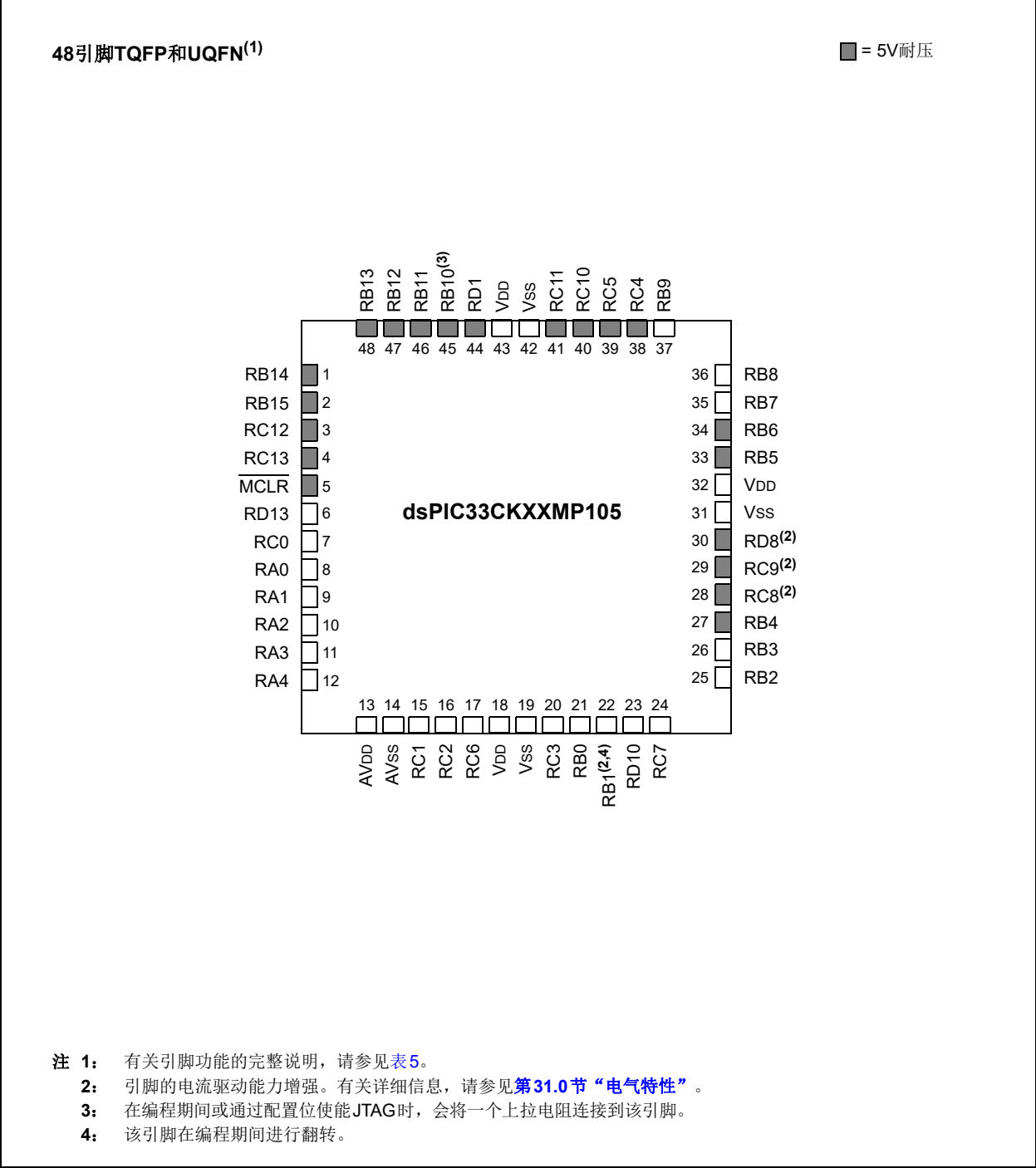


表4: 36引脚UQFN引脚功能完整说明

引脚 编号	功能 <sup>(1)</sup>	引脚 编号	功能 <sup>(1)</sup>
1	RP46/PWM1H/RB14	19	OA2OUT/AN1/AN7/ANA0/CMP1D/CMP2D/CMP3D/ RP34/INT0/RB2
2	RP47/PWM1L/RB15	20	PGD2/OA2IN-/AN8/ RP35/RB3
3	MCLR	21	PGC2/OA2IN+/ RP36/RB4
4	AN12/ANNO/ RP48/RC0	22	Vss
5	OA1OUT/AN0/CMP1A/IBIAS0/RA0	23	VDD
6	OA1IN-/ANA1/RA1	24	PGD3/ RP37/SDA2/RB5
7	OA1IN+/AN9/RA2	25	PGC3/ RP38/SCL2/RB6
8	DACOUT/AN3/CMP1C/RA3	26	TDO/AN2/CMP3A/ RP39/RB7
9	OA3OUT/AN4/CMP3B/IBIAS3/RA4	27	PGD1/AN10/ RP40/SCL1/RB8
10	AVDD	28	PGC1/AN11/ RP41/SDA1/RB9
11	AVSS	29	RP52/ASDA2/RC4
12	OA3IN-/AN13/CMP1B/ISRC0/ RP49/RC1	30	RP53/ASCL2/RC5
13	OA3IN+/AN14/CMP2B/ISRC1/ RP50/RC2	31	Vss
14	VDD	32	VDD
15	VSS	33	TMS/ RP42/PWM3H/RB10 <sup>(3)</sup>
16	AN15/CMP2A/IBIAS2/ RP51/RC3	34	TCK/ RP43/PWM3L/RB11
17	OSCI/CLKI/AN5/ RP32/RB0	35	TDI/ RP44/PWM2H/RB12
18	OSCO/CLKO/AN6/ RP33/RB1 <sup>(2,4)</sup>	36	RP45/PWM2L/RB13

- 注 1: RPn 表示可重映射外设功能。
- 2: 引脚的电流驱动能力增强。有关详细信息, 请参见第31.0节“电气特性”。
- 3: 在编程期间或通过配置位使能JTAG时, 会将一个上拉电阻连接到该引脚。
- 4: 该引脚在编程期间进行翻转。

引脚分配图（续）



# dsPIC33CK64MP105系列

表5: 48引脚TQFP和UQFN引脚功能完整说明

引脚 编号	功能 <sup>(1)</sup>	引脚 编号	功能 <sup>(1)</sup>
1	<b>RP46</b> /PWM1H/RB14	25	OA2OUT/AN1/AN7/ANA0/CMP1D/CMP2D/CMP3D/ <b>RP34</b> /INT0/RB2
2	<b>RP47</b> /PWM1L/RB15	26	PGD2/OA2IN-/AN8/ <b>RP35</b> /RB3
3	<b>RP60</b> /RC12	27	PGC2/OA2IN+/ <b>RP36</b> /RB4
4	<b>RP61</b> /RC13	28	<b>RP56</b> /ASDA1/SCK2/RC8 <sup>(2)</sup>
5	$\overline{\text{MCLR}}$	29	<b>RP57</b> /ASCL1/SDI2/RC9 <sup>(2)</sup>
6	ANN2/ <b>RP77</b> /RD13	30	<b>RP72</b> /SDO2/PC19/RD8 <sup>(2)</sup>
7	AN12/ANN0/ <b>RP48</b> /RC0	31	Vss
8	OA1OUT/AN0/CMP1A/IBIAS0/RA0	32	VDD
9	OA1IN-/ANA1/RA1	33	PGD3/ <b>RP37</b> /SDA2/RB5
10	OA1IN+/AN9/RA2	34	PGC3/ <b>RP38</b> /SCL2/RB6
11	DACOUT/AN3/CMP1C/RA3	35	TDO/AN2/CMP3A/ <b>RP39</b> /RB7
12	OA3OUT/AN4/CMP3B/IBIAS3/RA4	36	PGD1/AN10/ <b>RP40</b> /SCL1/RB8
13	AVDD	37	PGC1/AN11/ <b>RP41</b> /SDA1/RB9
14	AVss	38	<b>RP52</b> /ASDA2/RC4
15	OA3IN-/AN13/CMP1B/ISRC0/ <b>RP49</b> /RC1	39	<b>RP53</b> /ASCL2/RC5
16	OA3IN+/AN14/CMP2B/ISRC1/ <b>RP50</b> /RC2	40	<b>RP58</b> /RC10
17	AN17/ANN1/IBIAS1/ <b>RP54</b> /RC6	41	<b>RP59</b> /RC11
18	VDD	42	Vss
19	Vss	43	VDD
20	AN15/CMP2A/IBIAS2/ <b>RP51</b> /RC3	44	<b>RP65</b> /PWM4H/RD1
21	OSCI/CLKI/AN5/ <b>RP32</b> /RB0	45	TMS/ <b>RP42</b> /PWM3H/RB10 <sup>(3)</sup>
22	OSCO/CLKO/AN6/ <b>RP33</b> /RB1 <sup>(2,4)</sup>	46	TCK/ <b>RP43</b> /PWM3L/RB11
23	AN18/CMP3C/ISRC3/ <b>RP74</b> /RD10	47	TDI/ <b>RP44</b> /PWM2H/RB12
24	AN16/ISRC2/ <b>RP55</b> /RC7	48	<b>RP45</b> /PWM2L/RB13

- 注 1: **RPn** 表示可重映射外设功能。
- 2: 引脚的电流驱动能力增强。有关详细信息，请参见第31.0节“电气特性”。
- 3: 在编程期间或通过配置位使能JTAG时，会将一个上拉电阻连接到该引脚。
- 4: 该引脚在编程期间进行翻转。



## 目录

1.0	器件概述	13
2.0	16位数字信号控制器入门指南	17
3.0	CPU	23
4.0	存储器构成	33
5.0	闪存程序存储器	63
6.0	复位	77
7.0	中断控制器	81
8.0	I/O端口	101
9.0	带高频PLL的振荡器	155
10.0	直接存储器访问（DMA）控制器	179
11.0	支持精细边沿定位的高分辨率PWM	189
12.0	高速12位模数转换器（ADC）	223
13.0	带斜率补偿DAC的高速模拟比较器	253
14.0	正交编码器接口（QEI）	265
15.0	通用异步收发器（UART）	285
16.0	串行外设接口（SPI）	307
17.0	I <sup>2</sup> C	325
18.0	单边沿半字节传输（SENT）	335
19.0	Timer1	345
20.0	捕捉/比较/PWM/定时器模块（SCCP/MCCP）	349
21.0	可配置逻辑单元（CLC）	365
22.0	外设触发信号发生器（PTG）	377
23.0	偏流发生器（CBG）	393
24.0	运算放大器	397
25.0	程序监控定时器（DMT）	401
26.0	32位可编程循环冗余校验（CRC）发生器	409
27.0	节能特性	413
28.0	特殊功能	425
29.0	指令集汇总	451
30.0	开发支持	461
31.0	电气特性	463
32.0	高温电气特性	493
33.0	封装信息	501
附录A:	版本历史	519
索引		521
Microchip网站		529
变更通知客户服务		529
客户支持		529
产品标识体系		531

# dsPIC33CK64MP105系列

---

## 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需求。出版物的质量将随新文档及更新版本的推出而得到提升。

如果您对本出版物有任何问题和建议，请通过电子邮件联系我公司 TRC 经理，电子邮件地址为 [CTRC@microchip.com](mailto:CTRC@microchip.com)。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，请访问我公司网站：

<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。文献编号中紧跟数字串后的字母是版本号，例如：DS30000000A\_CN 是文档的 A 版本。

### 勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

### 客户通知系统

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

## 参考资料

本器件数据手册的内容基于《dsPIC33/PIC24 系列参考手册》中的以下各个章节。这些文档应作为某个特定模块或器件特性的操作的一般参考。

**注：** 要访问下列文档，请参见Microchip网站（[www.microchip.com](http://www.microchip.com)）上 dsPIC33CK64MP105 产品页面上的 Documents（文档）部分或从以下列表中选择系列参考手册章节。

除了参数、特性和其他文档，该页面还提供相关系列参考手册章节的链接。

- “简介”（[www.microchip.com/DS70573](http://www.microchip.com/DS70573)）
- “增强型 CPU”（[www.microchip.com/DS70005158](http://www.microchip.com/DS70005158)）
- “数据存储”（[www.microchip.com/DS70595](http://www.microchip.com/DS70595)）
- “dsPIC33/PIC24 程序存储器”（[www.microchip.com/DS70000613](http://www.microchip.com/DS70000613)）
- “复位”（[www.microchip.com/DS70602](http://www.microchip.com/DS70602)）
- “中断”（[www.microchip.com/DS70000600](http://www.microchip.com/DS70000600)）
- “带边沿检测的 I/O 端口”（[www.microchip.com/DS70005322](http://www.microchip.com/DS70005322)）
- “带高速 PLL 的振荡器模块”（[www.microchip.com/DS70005255](http://www.microchip.com/DS70005255)）
- “直接存储器访问（DMA）控制器”（[www.microchip.com/DS30009742](http://www.microchip.com/DS30009742)）
- “支持精细边沿定位的高分辨率 PWM”（[www.microchip.com/DS70005320](http://www.microchip.com/DS70005320)）
- “12 位高速多 SAR A/D 转换器（ADC）”（[www.microchip.com/DS70005213](http://www.microchip.com/DS70005213)）
- “高速模拟比较器模块”（[www.microchip.com/DS70005280](http://www.microchip.com/DS70005280)）
- “正交编码器接口（QE1）”（[www.microchip.com/DS70000601](http://www.microchip.com/DS70000601)）
- “多协议通用异步收发器（UART）模块”（[www.microchip.com/DS70005288](http://www.microchip.com/DS70005288)）
- “支持音频编解码器的串行外设接口（SPI）”（[www.microchip.com/DS70005136](http://www.microchip.com/DS70005136)）
- “I<sup>2</sup>C”（[www.microchip.com/DS70000195](http://www.microchip.com/DS70000195)）
- “单边沿半字节发送（SENT）模块”（[www.microchip.com/DS70005145](http://www.microchip.com/DS70005145)）
- “Timer1 模块”（[www.microchip.com/DS70005279](http://www.microchip.com/DS70005279)）
- “捕捉/比较/PWM/定时器（MCCP 和 SCCP）”（[www.microchip.com/DS30003035](http://www.microchip.com/DS30003035)）
- “可配置逻辑单元（CLC）”（[www.microchip.com/DS70005298](http://www.microchip.com/DS70005298)）
- “外设触发信号发生器（PTG）”（[www.microchip.com/DS70000669](http://www.microchip.com/DS70000669)）
- “偏流发生器（CBG）”（[www.microchip.com/DS70005253](http://www.microchip.com/DS70005253)）
- “程序监控定时器（DMT）”（[www.microchip.com/DS70005155](http://www.microchip.com/DS70005155)）
- “32 位可编程循环冗余校验（CRC）”（[www.microchip.com/DS30009729](http://www.microchip.com/DS30009729)）
- “双看门狗定时器”（[www.microchip.com/DS70005250](http://www.microchip.com/DS70005250)）
- “编程和诊断”（[www.microchip.com/DS70608](http://www.microchip.com/DS70608)）
- “CodeGuard™ 中等安全性”（[www.microchip.com/DS70005182](http://www.microchip.com/DS70005182)）
- “闪存编程”（[www.microchip.com/DS70000609](http://www.microchip.com/DS70000609)）

# dsPIC33CK64MP105系列

---

注:

## 1.0 器件概述

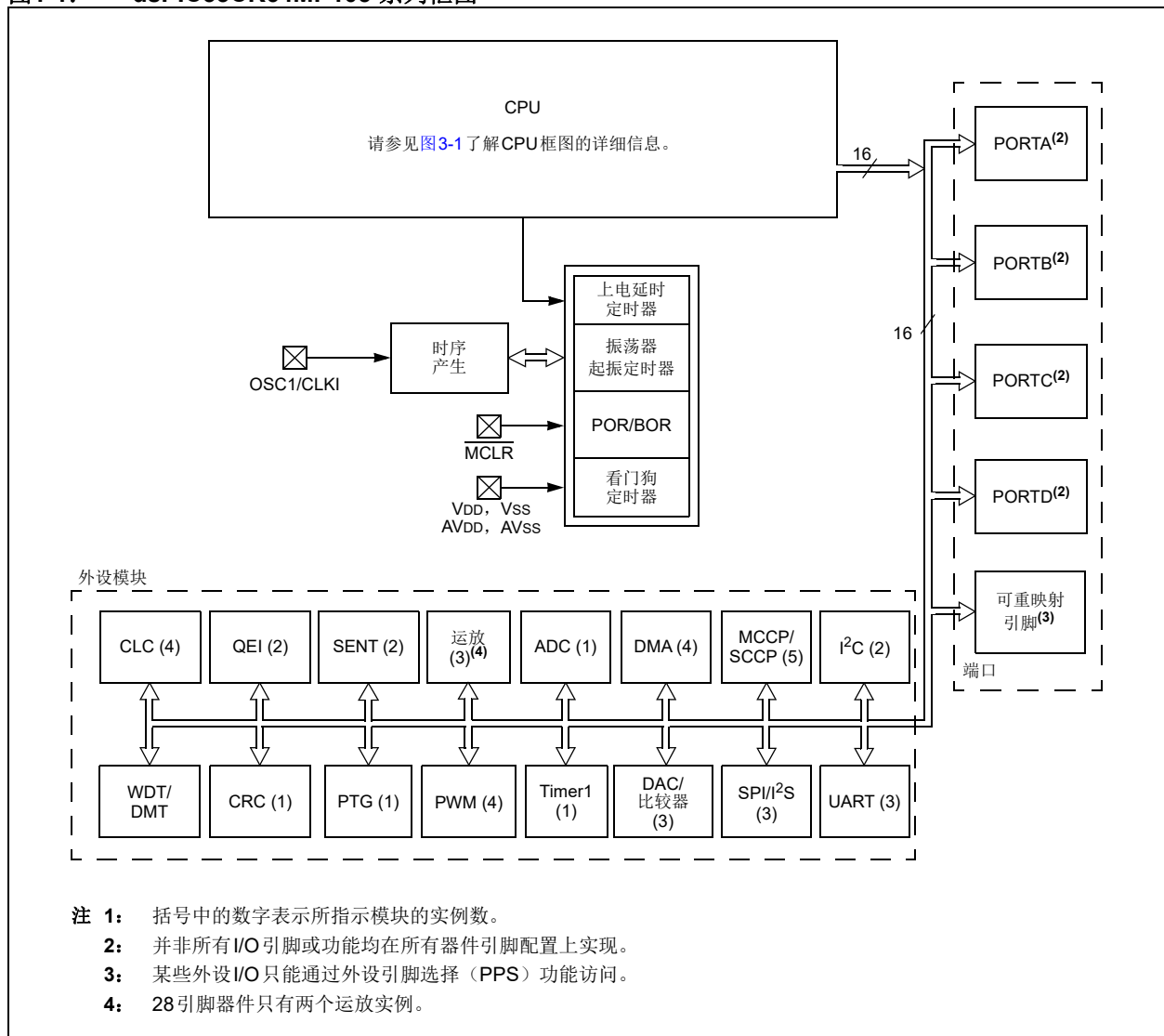
- 注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的相关章节,该文档可从Microchip网站([www.microchip.com](http://www.microchip.com))下载。
- 2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。关于具体器件的寄存器和位信息,请参见本数据手册中的**第4.0节“存储器构成”**。

本文档包含针对dsPIC33CK64MP105数字信号控制器(Digital Signal Controller, DSC)器件的具体信息。

dsPIC33CK64MP105器件在高性能16位MCU架构中,融合了丰富的数字信号处理器(Digital Signal Processor, DSP)功能。

图1-1给出了dsPIC33CK64MP105系列的内核和外设模块的一般框图。表1-1列出了引脚分配图中显示的各引脚的功能。

图1-1: dsPIC33CK64MP105 系列框图<sup>(1)</sup>



# dsPIC33CK64MP105系列

表1-1: 引脚I/O说明

引脚名称 <sup>(1)</sup>	引脚类型	缓冲器类型	支持PPS?	说明
AN0-AN18 ANA0-ANA1 ANN0-ANN1	I I I	模拟 模拟 模拟	否 否 否	模拟输入通道。 模拟备用输入。 模拟负输入。
CLKI	I	ST	否	外部时钟 (External Clock, EC) 源输入。总是与 OSC1 引脚功能相关联。
CLKO	O	—	否	可以通过配置位将其设置为输出 CPU 时钟。总是与 OSC0 引脚功能相关联。
OSC1 OSCO	I I/O	CMOS —	否 否	晶振输入。在晶振模式下, 该引脚与晶振或谐振器相连。 晶振输出。在晶振模式下, 该引脚与晶振或谐振器相连。
REFCLKI REFCLKO	I O	ST —	是 是	参考时钟输入。 参考时钟输出。
INT0 INT1 INT2 INT3	I I I I	ST ST ST ST	否 是 是 是	外部中断0。 外部中断1。 外部中断2。 外部中断3。
IOCA[4:0] IOCB[15:0] IOCC[13:0] IOCD1, IOCD8, IOCD10 IOCD13	I I I I	ST ST ST ST	否 否 否 否	PORTA 的电平变化中断输入。 PORTB 的电平变化中断输入。 PORTC 的电平变化中断输入。 PORTD 的电平变化中断输入。
QEIAx QEIBx QEINDXx QEIHOMx QEICMPx	I I I I O	ST ST ST ST —	是 是 是 是 是	QEIx 输入 A。 QEIx 输入 B。 QEIx 索引输入。 QEIx 归位输入。 QEIx 比较器输出。
RP32-RP61, RP65, RP72, RP74, RP77	I/O	ST	是	可重映射 I/O 端口。
RA0-RA4	I/O	ST	否	PORTA 是双向 I/O 端口。
RB0-RB15	I/O	ST	否	PORTB 是双向 I/O 端口。
RC0-RC13	I/O	ST	否	PORTC 是双向 I/O 端口。
RD1, RD8, RD10, RD13	I/O	ST	否	PORTD 是双向 I/O 端口。
T1CK	I	ST	是	Timer1 外部时钟输入。
U1CTS U1RTS U1RX U1TX U1DSR U1DTR	I O I O I O	ST — ST — ST —	是 是 是 是 是 是	UART1 允许发送。 UART1 请求发送。 UART1 接收。 UART1 发送。 UART1 数据设备就绪。 UART1 数据终端就绪。

图注: CMOS = CMOS 兼容输入或输出      模拟 = 模拟输入      P = 电源  
ST = CMOS 电平的施密特触发器输入      O = 输出      I = 输入  
PPS = 外设引脚选择

- 注 1: 并非所有引脚在所有封装类型的器件中都可用。关于引脚可用性, 请参见“引脚分配图”部分。
- 2: PWM4L 和 PWM4H 引脚支持 PPS。
- 3: 在 48 引脚器件上, SPI2 同时支持专用引脚和 PPS。

表1-1: 引脚I/O说明 (续)

引脚名称 <sup>(1)</sup>	引脚类型	缓冲器类型	支持PPS?	说明
U2CTS	I	ST	是	UART2 允许发送。
U2RTS	O	—	是	UART2 请求发送。
U2RX	I	ST	是	UART2 接收。
U2TX	O	—	是	UART2 发送。
U2DSR	I	ST	是	UART2 数据设备就绪。
U2DTR	O	—	是	UART2 数据终端就绪。
U3CTS	I	ST	是	UART3 允许发送。
U3RTS	O	—	是	UART3 请求发送。
U3RX	I	ST	是	UART3 接收。
U3TX	O	—	是	UART3 发送。
U3DSR	I	ST	是	UART3 数据设备就绪。
U3DTR	O	—	是	UART3 数据终端就绪。
SENT1	I	ST	是	SENT1 输入。
SENT1OUT	O	—	是	SENT1 输出。
SENT2	I	ST	是	SENT2 输入。
SENT2OUT	O	—	是	SENT2 输出。
PTGTRG24	O	—	是	PTG 触发信号输出 24。
PTGTRG25	O	—	是	PTG 触发信号输出 25。
TCKI1-TCKI5	I	ST	是	MCCP/SCCP 定时器输入。
ICM1-ICM5	I	ST	是	MCCP/SCCP 捕捉输入。
OCFA/OCFB	I	ST	是	MCCP/SCCP 故障输入。
OCM1x/OCM5x	O	—	是	MCCP/SCCP 比较输出。
SCK1	I/O	ST	是	SPI1 的同步串行时钟输入/输出。
SDI1	I	ST	是	SPI1 数据输入。
SDO1	O	—	是	SPI1 数据输出。
SS1	I/O	ST	是	SPI1 从同步或帧脉冲 I/O。
SCK2	I/O	ST	是 <sup>(3)</sup>	SPI2 的同步串行时钟输入/输出。
SDI2	I	ST	是 <sup>(3)</sup>	SPI2 数据输入。
SDO2	O	—	是 <sup>(3)</sup>	SPI2 数据输出。
SS2	I/O	ST	是 <sup>(3)</sup>	SPI2 从同步或帧脉冲 I/O。
SCK3	I/O	ST	是	SPI3 的同步串行时钟输入/输出。
SDI3	I	ST	是	SPI3 数据输入。
SDO3	O	—	是	SPI3 数据输出。
SS3	I/O	ST	是	SPI3 从同步或帧脉冲 I/O。
SCL1	I/O	ST	否	I2C1 的同步串行时钟输入/输出。
SDA1	I/O	ST	否	I2C1 的同步串行数据输入/输出。
ASCL1	I/O	ST	否	I2C1 的备用同步串行时钟输入/输出。
ASDA1	I/O	ST	否	I2C1 的备用同步串行数据输入/输出。
SCL2	I/O	ST	否	I2C2 的同步串行时钟输入/输出。
SDA2	I/O	ST	否	I2C2 的同步串行数据输入/输出。
ASCL2	I/O	ST	否	I2C2 的备用同步串行时钟输入/输出。
ASDA2	I/O	ST	否	I2C2 的备用同步串行数据输入/输出。

图注: CMOS = CMOS 兼容输入或输出      模拟 = 模拟输入      P = 电源  
 ST = CMOS 电平的施密特触发器输入      O = 输出      I = 输入  
 PPS = 外设引脚选择

注 1: 并非所有引脚在所有封装类型的器件中都可用。关于引脚可用性, 请参见“[引脚分配图](#)”部分。

2: PWM4L 和 PWM4H 引脚支持 PPS。

3: 在 48 引脚器件上, SPI2 同时支持专用引脚和 PPS。

# dsPIC33CK64MP105系列

表1-1: 引脚I/O说明 (续)

引脚名称 <sup>(1)</sup>	引脚类型	缓冲器类型	支持PPS?	说明
TMS	I	ST	否	JTAG 测试模式选择引脚。
TCK	I	ST	否	JTAG 测试时钟输入引脚。
TDI	I	ST	否	JTAG 测试数据输入引脚。
TDO	O	—	否	JTAG 测试数据输出引脚。
PC18-PC118	I	ST	是	PWM 输入 8 至 18。
PC119	I	ST	否	PWM 输入 19。
PWMEA-PWMED	O	—	是	PWM 事件输出 A 至 D。
PWM1L-PWM4L <sup>(2)</sup>	O	—	否	PWM 下桥臂输出 1 至 4。
PWM1H-PWM4H <sup>(2)</sup>	O	—	否	PWM 上桥臂输出 1 至 4。
CLCINA-CLCIND	I	ST	是	CLC 输入 A 至 D。
CLCxOUT	O	—	是	CLCx 输出。
CMP1A-CMP3A	I	模拟	否	比较器通道 1A 至 3A 的输入。
CMP1B-CMP3B	I	模拟	否	比较器通道 1B 至 3B 的输入。
CMP1C-CMP3C	I	模拟	否	比较器通道 1C 至 3C 的输入。
CMP1D-CMP3D	I	模拟	否	比较器通道 1D 至 3D 的输入。
DACOUT	O	—	否	DAC 输出电压。
IBIAS0-IBIAS3	O	模拟	否	50 $\mu$ A 恒流输出 0 至 3。
ISRC0-ISRC3	O	模拟	否	10 $\mu$ A 恒流输出 0 至 3。
OA1IN+	I	—	否	运放 1 同相输入。
OA1IN-	I	—	否	运放 1 反相输入。
OA1OUT	O	—	否	运放 1 输出。
OA2IN+	I	—	否	运放 2 同相输入。
OA2IN-	I	—	否	运放 2 反相输入。
OA2OUT	O	—	否	运放 2 输出。
OA3IN+	I	—	否	运放 3 同相输入。
OA3IN-	I	—	否	运放 3 反相输入。
OA3OUT	O	—	否	运放 3 输出。
ADTRG31	I	ST	否	外部 ADC 触发源。
PGD1	I/O	ST	否	编程/调试通信通道 1 的数据 I/O 引脚。
PGC1	I	ST	否	编程/调试通信通道 1 的时钟输入引脚。
PGD2	I/O	ST	否	编程/调试通信通道 2 的数据 I/O 引脚。
PGC2	I	ST	否	编程/调试通信通道 2 的时钟输入引脚。
PGD3	I/O	ST	否	编程/调试通信通道 3 的数据 I/O 引脚。
PGC3	I	ST	否	编程/调试通信通道 3 的时钟输入引脚。
MCLR	I/P	ST	否	主复位输入。此引脚为低电平有效的器件复位输入端。
AVDD	P	P	否	模拟模块的正电源。此引脚必须始终连接。
AVSS	P	P	否	模拟模块的参考地。此引脚必须始终连接。
VDD	P	P	否	外设逻辑和 I/O 引脚的正电源。
VSS	P	P	否	逻辑和 I/O 引脚的参考地。

图注: CMOS = CMOS 兼容输入或输出      模拟 = 模拟输入      P = 电源  
 ST = CMOS 电平的施密特触发器输入      O = 输出      I = 输入  
 PPS = 外设引脚选择

注 1: 并非所有引脚在所有封装类型的器件中都可用。关于引脚可用性, 请参见“引脚分配图”部分。

2: PWM4L 和 PWM4H 引脚支持 PPS。

3: 在 48 引脚器件上, SPI2 同时支持专用引脚和 PPS。



## 2.0 16位数字信号控制器入门指南

### 2.1 基本连接要求

在开始使用dsPIC33CK64MP105系列器件进行开发之前，需要注意最基本的器件引脚连接要求。下面列出了必须始终连接的引脚名称：

- 所有VDD和VSS引脚（见第2.2节“去耦电容”）
- 所有AVDD和AVSS引脚（无论是否使用ADC模块）（见第2.2节“去耦电容”）
- MCLR引脚（见第2.3节“主复位（MCLR）引脚”）
- 用于在线串行编程（In-Circuit Serial Programming™，ICSP™）和调试目的的PGCx/PGDx引脚（见第2.4节“ICSP引脚”）
- OSCI和OSCO引脚（使用外部振荡源时）（见第2.5节“外部振荡器引脚”）

### 2.2 去耦电容

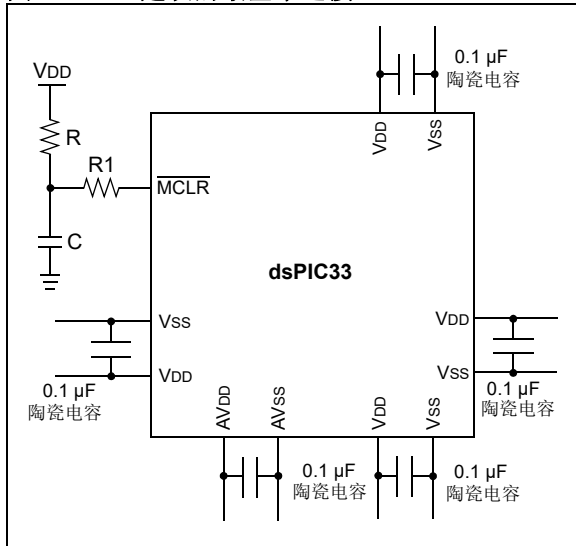
需要在每对电源引脚（例如，VDD/VSS和AVDD/AVSS）上使用去耦电容。

使用去耦电容时，需要考虑以下标准：

- **电容的类型和电容值：**建议使用参数为0.1  $\mu\text{F}$ （100 nF）、10-20V的电容。该电容应具有低ESR，谐振频率为20 MHz或更高。建议使用陶瓷电容。
- **在印刷电路板上的放置：**去耦电容应尽可能靠近引脚。建议将电容与器件放置在电路板的同一层。如果空间受限，可以使用过孔将电容放置在PCB的另一层，但请确保从引脚到电容的走线长度小于0.25英寸（6毫米）。
- **高频噪声处理：**如果电路板上存在高频噪声（频率高于数十MHz），则另外添加一个陶瓷电容，与上述去耦电容并联。第二个电容的电容值可以介于0.001  $\mu\text{F}$ 和0.01  $\mu\text{F}$ 之间。请将第二个电容放置在靠近主去耦电容的位置。在高速电路设计中，需要考虑尽可能靠近电源和接地引脚放置这一对电容。例如，0.1  $\mu\text{F}$ 电容与0.001  $\mu\text{F}$ 电容并联。
- **最大程度提高性能：**对于从电源电路开始的电路板布线，需要将电源和返回走线先连接到去耦电容，然后再与器件引脚连接。这可以确保去耦电容是电源链中的第一个元件。同等重要的是尽可能减小电容和电源引脚之间的走线长度，从而降低PCB走线电感。

# dsPIC33CK64MP105系列

图2-1: 建议的最基本连接



## 2.2.1 大电容

对于电源走线长度超出6英寸的电路板，建议对集成电路（包括DSC）使用大电容来提供本地电源。大电容的电容值应根据连接电源与器件的走线电阻和应用中器件消耗的最大电流确定。也就是说，选择的大电容需要满足器件的可接受电压骤降要求。典型值的范围为4.7  $\mu\text{F}$  至47  $\mu\text{F}$ 。

## 2.3 主复位 ( $\overline{\text{MCLR}}$ ) 引脚

$\overline{\text{MCLR}}$  引脚提供两种特定的器件功能：

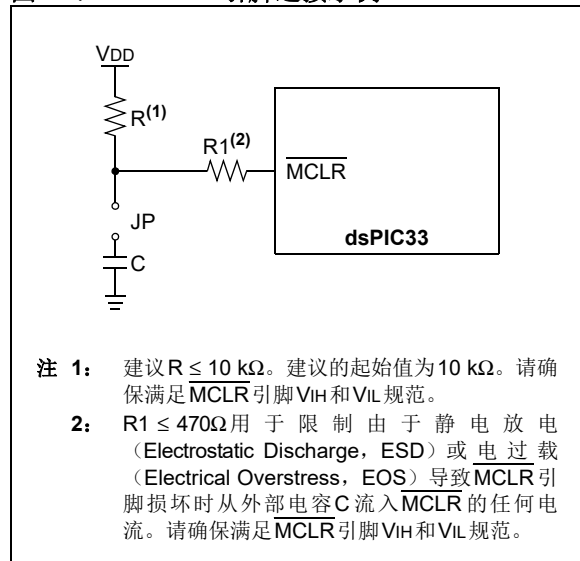
- 器件复位
- 器件编程和调试

在器件编程和调试过程中，必须考虑到引脚上可能会增加的电阻和电容。器件编程器和调试器会驱动 $\overline{\text{MCLR}}$ 引脚。因此，特定电压 ( $V_{IH}$  和  $V_{IL}$ ) 和快速信号跳变一定不能受到不利影响。所以，需要根据应用和PCB需求来调整R和C的具体值。

例如，如图2-2所示，在编程和调试操作期间，建议将电容C与 $\overline{\text{MCLR}}$ 引脚隔离。

将图2-2所示的元件放置在距离 $\overline{\text{MCLR}}$ 引脚0.25英寸（6毫米）范围内。

图2-2:  $\overline{\text{MCLR}}$  引脚连接示例



## 2.4 ICSP引脚

PGCx和PGDx引脚用于进行ICSP和调试。建议尽可能缩短ICSP连接器与器件ICSP引脚之间的走线长度。如果ICSP连接器会遇到ESD事件，则建议添加一个串联电阻，电阻值为几十欧姆，不要超出100Ω。

建议不要在PGCx和PGDx引脚上连接上拉电阻、串联二极管和电容，因为它们会影响编程器/调试器与器件之间的通信。如果应用需要此类分立元件，则在编程和调试期间应将它们从电路中去掉。或者，请参见相应器件闪存编程规范中的交流/直流特性与时序要求信息，了解关于容性负载限制、引脚输入高电压（V<sub>IH</sub>）和输入低电压（V<sub>IL</sub>）要求的信息。

请确保编程到器件中的“通信通道选择”（即PGCx/PGDx引脚）符合与MPLAB<sup>®</sup>调试器工具的ICSP物理连接。

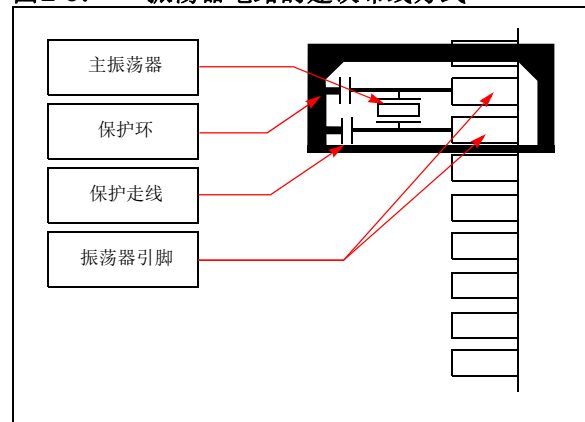
有关MPLAB编程器/调试器连接要求的更多信息，请参见Microchip网站。

## 2.5 外部振荡器引脚

许多DSC都有至少两个振荡器可供选择：高频主振荡器（Primary Oscillator, POSC）和低频辅助振荡器（Secondary Oscillator, SOS）。有关详细信息，请参见第9.4节“主振荡器（POSC）”。

振荡器电路与器件应放置在电路板的同一层。此外，请将振荡器电路放置在靠近相应振荡器引脚的位置，它们之间的距离不要超出0.5英寸（12毫米）。负载电容应靠近振荡器本身，位于电路板的同一层。请在振荡器电路周围使用接地覆铜区，以将其与周围电路隔离。接地覆铜区应与MCU地直接连接。不要在接地覆铜区内安排任何信号走线或电源走线。此外，如果使用双面电路板，请避免在电路板上晶振所在位置的背面有任何走线。图2-3给出了一个建议的布线图。

图2-3： 振荡器电路的建议布线方式



2.6 器件启动时的振荡器值条件

如果目标器件的PLL被使能且配置为器件启动时使用的振荡器，则振荡器源的最高频率必须限制为某一特定频率（见第9.0节“带高频PLL的振荡器”），以符合器件的PLL启动条件。这意味着，如果外部振荡器频率超出该范围，应用必须首先在FRC模式下启动。如果POR之后的默认PLL设置的振荡器频率超出该范围，将违反器件工作速度。

器件上电之后，应用固件可以将PLL SFR、CLKDIV和PLLFBF初始化为适当的值，然后执行时钟切换，切换为振荡器 + PLL时钟源。注意，必须通过器件配置字使能时钟切换。

2.7 未用I/O

未用I/O引脚应配置为输出，并驱动为逻辑低电平状态。

或者，在Vss和未用引脚之间连接一个1k至10k的电阻，并将输出驱动为逻辑低电平。

2.8 目标应用

- 功率因数校正（Power Factor Correction, PFC）：
  - 交错式PFC
  - 临界导通PFC
  - 无桥PFC
- 直流/直流转换器：
  - 降压、升压、正激、反激和推挽
  - 半桥/全桥
  - 相移全桥
  - 谐振转换器
- 直流/交流：
  - 半桥/全桥逆变器
  - 谐振逆变器
- 电机控制
  - BLDC
  - PMSM
  - SR
  - ACIM

图2-4至图2-6给出了典型应用连接的示例。

图2-4: 交错式PFC

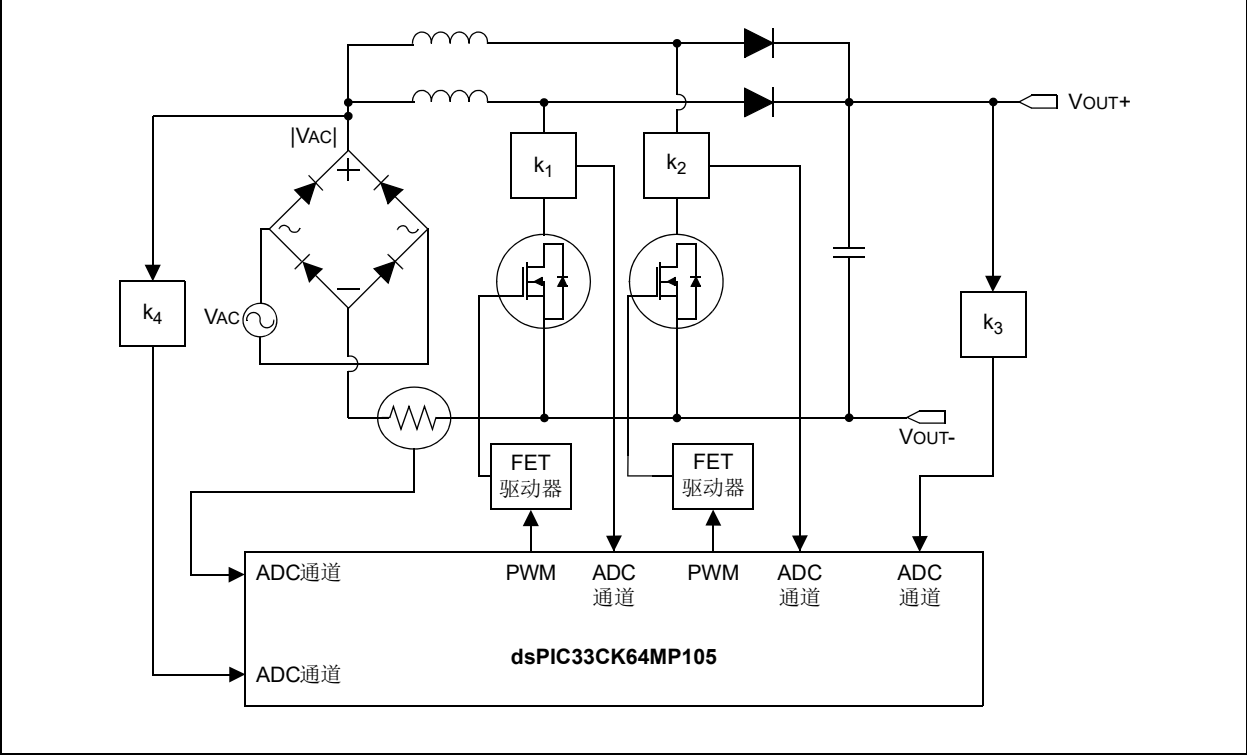


图2-5: 相移全桥转换器

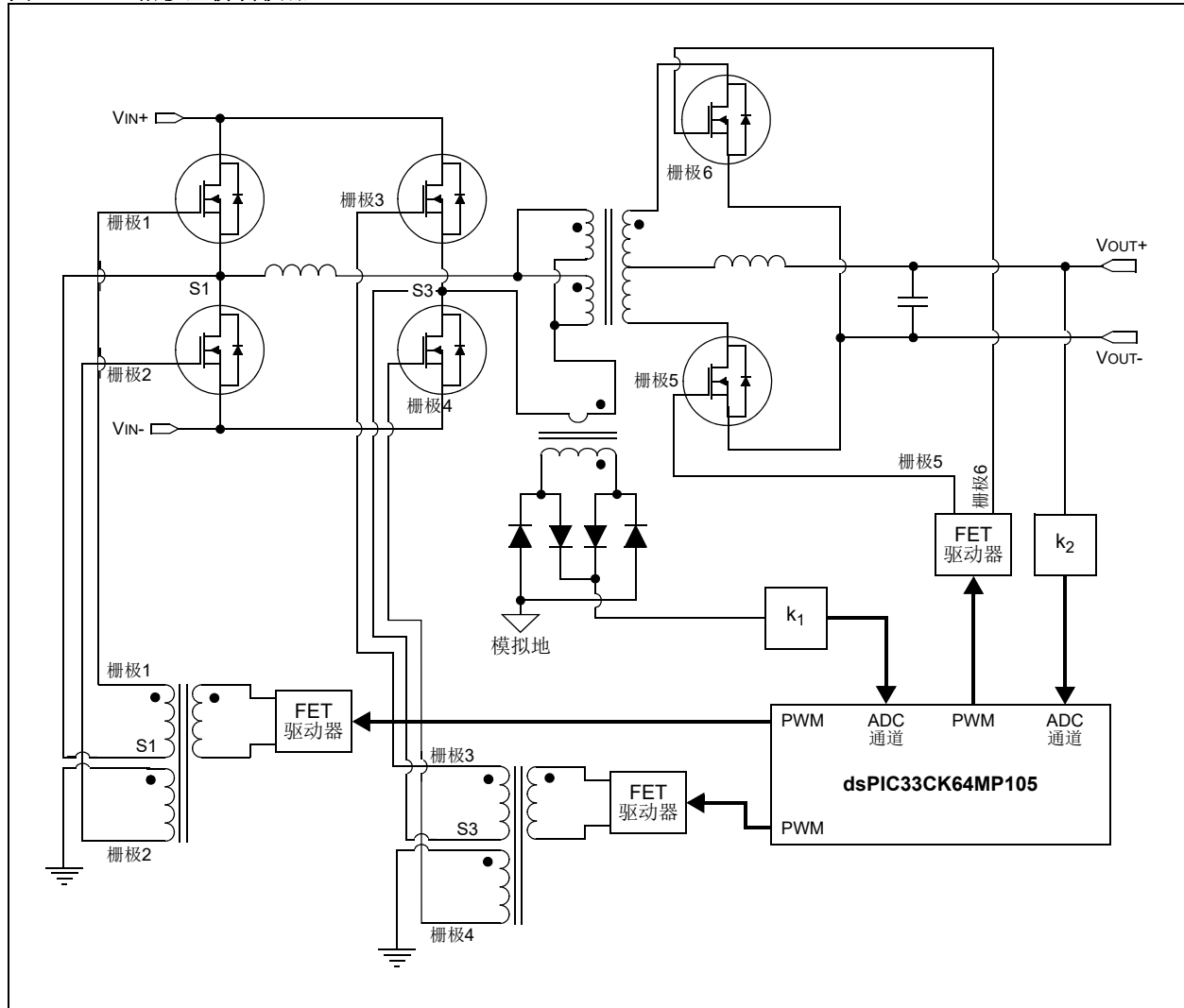
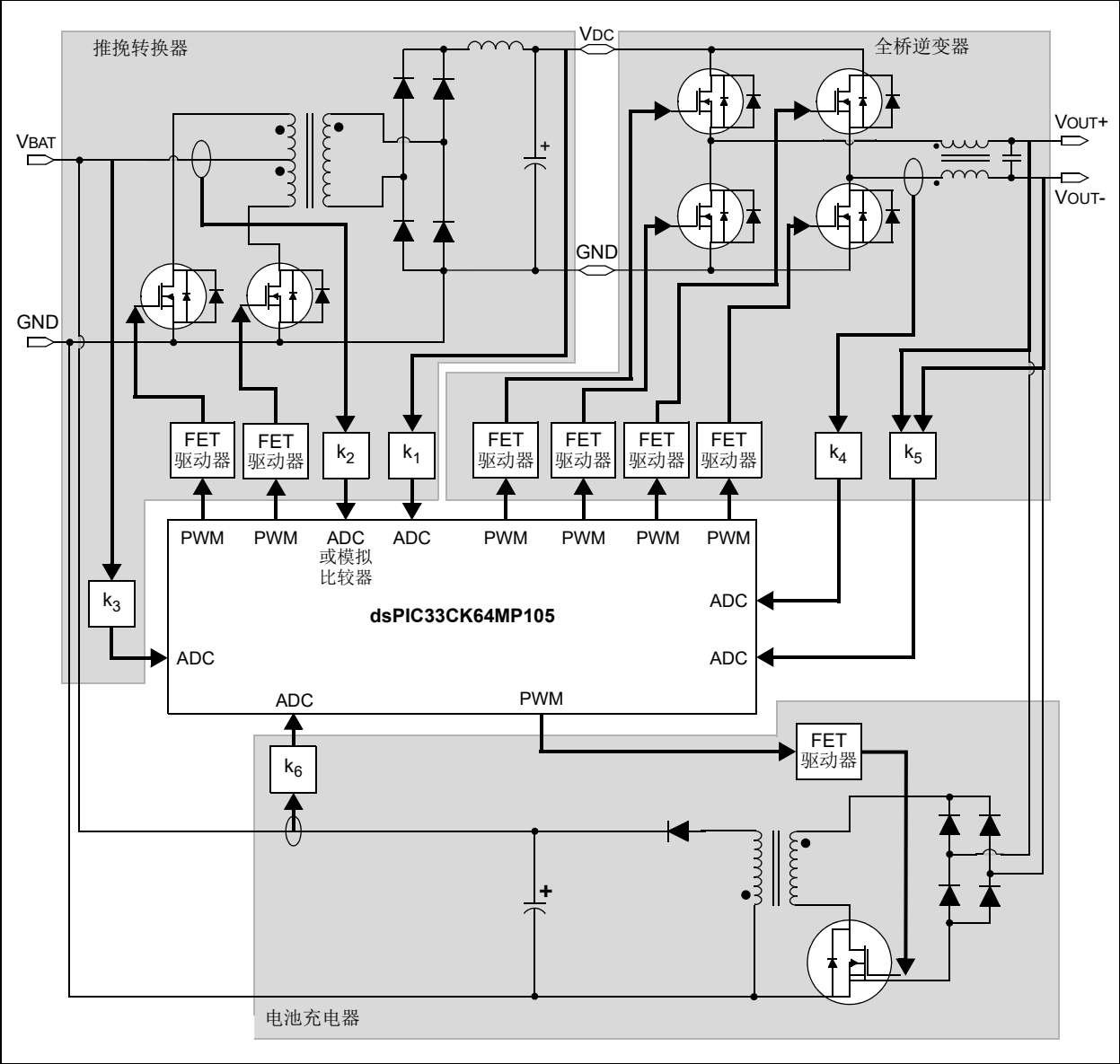


图2-6： 离线UPS



### 3.0 CPU

- 注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“dsPIC33E增强型CPU”(www.microchip.com/DS70005158)。
- 2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。关于具体器件的寄存器和位信息,请参见本数据手册中的第4.0节“存储器构成”。

dsPIC33CK64MP105系列CPU采用16位数据总线的改进型哈佛架构,具有增强指令集以及对数字信号处理(Digital Signal Processing, DSP)的重要支持。CPU具有24位指令字,指令字包含长度可变的操作码字段。程序计数器(PC)为23位宽,可以寻址最大4M x 24位的用户程序存储空间。

指令预取机制可帮助维持吞吐量,并使指令的执行具有可预测性。除了改变程序流的指令、双字传送(MOV.D)指令、PSV访问和表指令以外,大多数指令都以单个周期有效执行速率执行。使用DO和REPEAT指令支持无开销的程序循环结构,这两条指令在任何时间都可以被中断。

#### 3.1 寄存器

dsPIC33CK64MP105器件在编程模型中有16个16位工作寄存器。每个工作寄存器都可以充当数据、地址或地址偏移寄存器。第16个工作寄存器(W15)作为软件堆栈指针(Software Stack Pointer, SSP)工作,用于中断和调用。

另外,dsPIC33CK64MP105器件还包括4个备用工作寄存器组,由W0至W14组成。备用工作寄存器的内容是可持久保存的,有助于在中断服务程序(Interrupt Service Routine, ISR)执行期间减少寄存器内容的保存和恢复开销。可以通过配置FALTREG配置寄存器的CTXTx[2:0]位,为备用工作寄存器分配特定的中断优先级(IPL1到IPL6)。也可以使用CTXTSWP指令手动访问备用工作寄存器。CTXTSTAT寄存器中的CCTXI[2:0]和MCTXI[2:0]位可用于识别当前和最近手动选择的工作寄存器组。

### 3.2 指令集

dsPIC33CK64MP105器件的指令集包含两类指令:MCU类指令和DSP类指令。这两类指令无缝地集成到架构中,并从单个执行单元执行。指令集包含多种寻址模式,指令的设计可使C编译器的效率达到最优。

### 3.3 数据空间寻址

基本数据空间可寻址为最多4K字或8KB,该空间分成两块,分别称为X和Y数据存储区。每个存储块有各自独立的地址发生单元(Address Generation Unit, AGU)。MCU类指令只通过X存储区AGU进行操作,可将整个存储器映射作为一个线性数据空间访问。某些DSP指令通过X和Y的AGU进行操作以支持双操作数读操作,这样会将数据地址空间分成两个部分。X和Y数据空间的边界视具体器件而定。

可以选择将数据存储空间的高32KB映射到任何16K程序字范围的程序空间(Program Space, PS)。程序空间到数据空间的映射功能(称为程序空间可视性(Program Space Visibility, PSV))让任何指令都能像访问数据空间一样访问程序空间。有关PSV和表访问的更多详细信息,请参见《dsPIC33/PIC24系列参考手册》中的“数据存储器”(www.microchip.com/DS70595)。

在dsPIC33CK64MP105系列器件上,X和Y地址空间都支持无开销循环缓冲区(模寻址)。模寻址省去了DSP算法的软件边界检查开销。X AGU的循环寻址可以用于任何MCU类指令。X AGU还支持位反转寻址,这大幅简化了基2 FFT算法对输入或输出数据的重新排序。

### 3.4 寻址模式

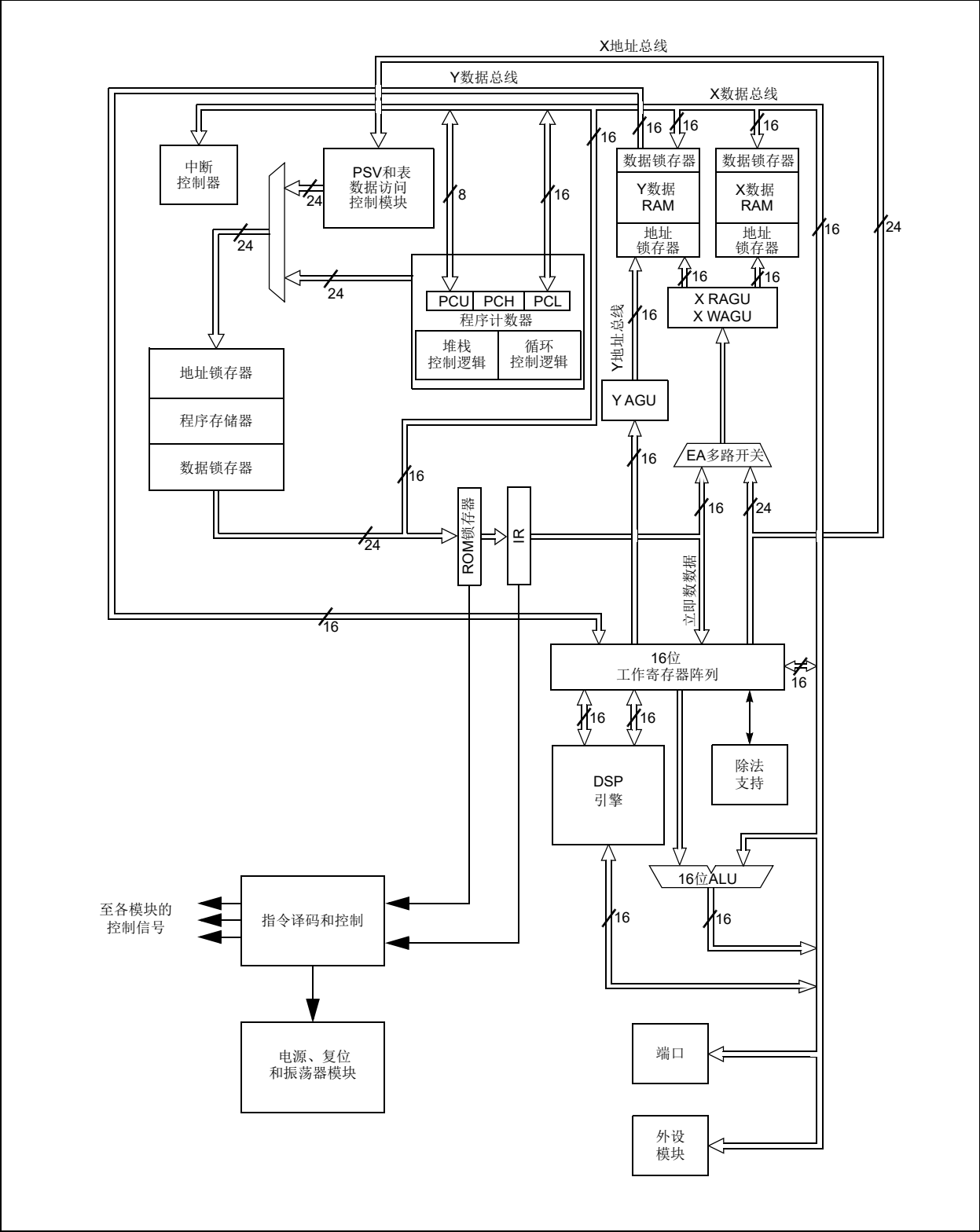
CPU支持以下寻址模式:

- 固有寻址(无操作数)
- 相对寻址
- 立即数寻址
- 存储器直接寻址
- 寄存器直接寻址
- 寄存器间接寻址

根据每条指令的功能要求,每条指令与预定义的寻址模式组相关联。每条指令最多支持六种寻址模式。

# dsPIC33CK64MP105系列

图3-1: dsPIC33CK64MP105系列CPU框图





## 3.4.1 编程模型

图3-2给出了dsPIC33CK64MP105系列的编程模型。编程模型中的所有寄存器都是存储器映射的，并且可以通过指令直接操作。表3-1列出了对每个寄存器的说明。

除了编程模型中包含的寄存器，dsPIC33CK64MP105器件还包含用于模寻址、位反转寻址和中断的控制寄存器。这些寄存器将在本文档后续章节中进行说明。

与编程模型相关的所有寄存器都是存储器映射的，如图3-2所示。

**表3-1： 编程模型的寄存器说明**

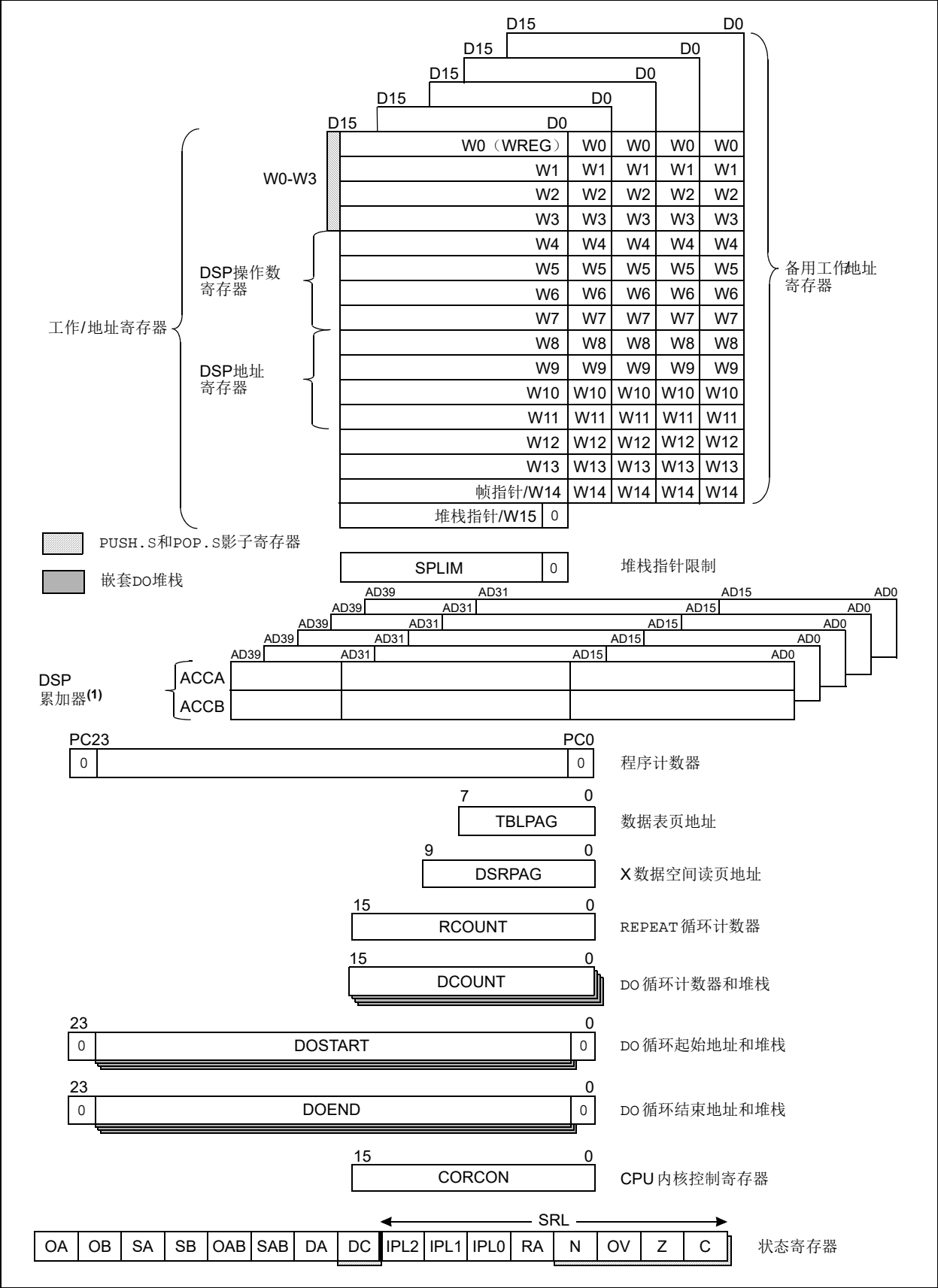
寄存器名称	说明
W0至W15 <sup>(1)</sup>	工作寄存器阵列
W0至W14 <sup>(1)</sup>	备用工作寄存器阵列1
W0至W14 <sup>(1)</sup>	备用工作寄存器阵列2
W0至W14 <sup>(1)</sup>	备用工作寄存器阵列3
W0至W14 <sup>(1)</sup>	备用工作寄存器阵列4
ACCA和ACCB	40位DSP累加器（额外4个备用累加器）
PC	23位程序计数器
SR	ALU和DSP引擎状态寄存器
SPLIM	堆栈指针限制值寄存器
TBLPAG	表存储器页地址寄存器
DSRPAG	扩展数据空间（Extended Data Space，EDS）读页寄存器
RCOUNT	REPEAT循环计数器寄存器
DCOUNT	DO循环计数器寄存器
DOSTARTH和DOSTARTL <sup>(2)</sup>	DO循环起始地址寄存器（高位字和低位字）
DOENDH和DOENDL	DO循环结束地址寄存器（高位字和低位字）
CORCON	包含DSP引擎、DO循环控制和陷阱状态位

**注 1：** 存储器映射的W0至W14表示当前激活的CPU现场中寄存器的值。

**2：** DOSTARTH和DOSTARTL寄存器是只读的。

# dsPIC33CK64MP105系列

图3-2: 编程模型



## 3.4.2 CPU资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 3.4.2.1 主要资源

- 《dsPIC33/PIC24系列参考手册》中的“**增强型CPU**” ([www.microchip.com/DS70005158](http://www.microchip.com/DS70005158))
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24系列参考手册》章节
- 开发工具

# dsPIC33CK64MP105系列

## 3.4.3 CPU控制寄存器

### 寄存器3-1: SR: CPU状态寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/C-0	R/C-0	R-0	R/W-0
OA	OB	SA <sup>(3)</sup>	SB <sup>(3)</sup>	OAB	SAB	DA	DC
bit 15							bit 8

R/W-0 <sup>(2)</sup>	R/W-0 <sup>(2)</sup>	R/W-0 <sup>(2)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(1)</sup>	IPL1 <sup>(1)</sup>	IPL0 <sup>(1)</sup>	RA	N	OV	Z	C
bit 7							bit 0

图注:	C = 可清零位		
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 15	<b>OA:</b> 累加器A溢出状态位 1 = 累加器A已溢出 0 = 累加器A未溢出
bit 14	<b>OB:</b> 累加器B溢出状态位 1 = 累加器B已溢出 0 = 累加器B未溢出
bit 13	<b>SA:</b> 累加器A饱和“粘住”状态位 <sup>(3)</sup> 1 = 累加器A饱和或在某时已经饱和 0 = 累加器A未饱和
bit 12	<b>SB:</b> 累加器B饱和“粘住”状态位 <sup>(3)</sup> 1 = 累加器B饱和或在某时已经饱和 0 = 累加器B未饱和
bit 11	<b>OAB:</b> OA和OB组合的累加器溢出状态位 1 = 累加器A或B已溢出 0 = 累加器A和累加器B都未溢出
bit 10	<b>SAB:</b> SA和SB组合的累加器“粘住”状态位 1 = 累加器A或B饱和或在某时已经饱和 0 = 累加器A和B都未饱和
bit 9	<b>DA:</b> DO循环活动位 1 = 正在进行DO循环 0 = 不在进行DO循环
bit 8	<b>DC:</b> MCU ALU半进位/借位标志位 1 = 结果的第4个低位（对于字节大小的数据）或第8个低位（对于字大小的数据）发生了进位 0 = 结果的第4个低位（对于字节大小的数据）或第8个低位（对于字大小的数据）未发生进位

- 注 1: IPL[2:0]位与IPL[3]位（CORCON[3]）组合形成CPU中断优先级。如果IPL[3] = 1，那么括号中的值表示IPL。当IPL[3] = 1时，禁止用户中断。
- 2: 当NSTDIS位（INTCON[15]）= 1时，IPL[2:0]状态位是只读的。
- 3: 对SR寄存器的数据写操作可以修改SA和SB位，方法是向SA和SB写入数据或清零SAB位。要避免可能出现的SA或SB位写竞争条件，不要使用位操作来修改SA和SB位。

## 寄存器3-1: SR: CPU状态寄存器 (续)

bit 7-5	<b>IPL[2:0]:</b> CPU中断优先级状态位 <sup>(1,2)</sup> 111 = CPU中断优先级为7 (15); 禁止用户中断 110 = CPU中断优先级为6 (14) 101 = CPU中断优先级为5 (13) 100 = CPU中断优先级为4 (12) 011 = CPU中断优先级为3 (11) 010 = CPU中断优先级为2 (10) 001 = CPU中断优先级为1 (9) 000 = CPU中断优先级为0 (8)
bit 4	<b>RA:</b> REPEAT循环活动位 1 = 正在进行REPEAT循环 0 = 不在进行REPEAT循环
bit 3	<b>N:</b> MCU ALU负标志位 1 = 结果为负 0 = 结果为非负 (零或正值)
bit 2	<b>OV:</b> MCU ALU溢出标志位 该位用于有符号的算术运算 (以二进制补码方式进行)。它表示量值上的溢出, 这种溢出将导致符号位改变状态。 1 = 有符号算术运算中发生溢出 (本次算术运算) 0 = 未发生溢出
bit 1	<b>Z:</b> MCU ALU全零标志位 1 = 影响Z位的任何运算在过去某时已将该位置1 0 = 影响Z位的最近一次运算已将该位清零 (即运算结果非零)
bit 0	<b>C:</b> MCU ALU进位/借位标志位 1 = 结果的最高有效位发生了进位 0 = 结果的最高有效位未发生进位

- 注 1:** IPL[2:0]位与IPL[3]位 (CORCON[3]) 组合形成CPU中断优先级。如果IPL[3] = 1, 那么括号中的值表示IPL。当IPL[3] = 1时, 禁止用户中断。
- 2:** 当NSTDIS位 (INTCON1[15]) = 1时, IPL[2:0]状态位是只读的。
- 3:** 对SR寄存器的数据写操作可以修改SA和SB位, 方法是向SA和SB写入数据或清零SAB位。要避免可能出现的SA或SB位写竞争条件, 不要使用位操作来修改SA和SB位。

# dsPIC33CK64MP105系列

## 寄存器3-2: CORCON: 内核控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
VAR	—	US1	US0	EDT <sup>(1)</sup>	DL2	DL1	DL0
bit 15							bit 8

R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3 <sup>(2)</sup>	SFA	RND	IF
bit 7							bit 0

图注:	C = 可清零位
R = 可读位	W = 可写位
-n = POR时的值	1 = 置1
	U = 未实现位, 读为0
	0 = 清零
	x = 未知

- bit 15 **VAR:** 可变异常处理延时控制位  
1 = 使能可变异常处理延时  
0 = 使能固定异常处理延时
- bit 14 **未实现:** 读为0
- bit 13-12 **US[1:0]:** DSP乘法无符号/有符号控制位  
11 = 保留  
10 = DSP引擎执行混合符号乘法运算  
01 = DSP引擎执行无符号乘法运算  
00 = DSP引擎执行有符号乘法运算
- bit 11 **EDT:** DO循环提前终止控制位<sup>(1)</sup>  
1 = 在当前循环迭代结束时终止执行DO循环  
0 = 无影响
- bit 10-8 **DL[2:0]:** DO循环嵌套层级状态位  
111 = 正在执行7层DO循环  
...  
001 = 正在执行1层DO循环  
000 = 不在执行DO循环
- bit 7 **SATA:** ACCA饱和和使能位  
1 = 使能累加器A饱和  
0 = 禁止累加器A饱和
- bit 6 **SATB:** ACCB饱和和使能位  
1 = 使能累加器B饱和  
0 = 禁止累加器B饱和
- bit 5 **SATDW:** DSP引擎的数据空间写饱和和使能位  
1 = 使能数据空间写饱和  
0 = 禁止数据空间写饱和
- bit 4 **ACCSAT:** 累加器饱和模式选择位  
1 = 9.31饱和 (超饱和)  
0 = 1.31饱和 (正常饱和)
- bit 3 **IPL3:** CPU中断优先级状态位3<sup>(2)</sup>  
1 = CPU中断优先级大于7  
0 = CPU中断优先级小于或等于7

- 注 1: 该位将总是读为0。
- 2: IPL3位与IPL[2:0]位 (SR[7:5]) 组合形成CPU中断优先级。

## 寄存器3-2: CORCON: 内核控制寄存器 (续)

bit 2	<b>SFA:</b> 堆栈帧有效状态位 1 = 堆栈帧有效; 无论 DSRPAG 如何, W14 和 W15 都寻址 0x0000 至 0xFFFF 0 = 堆栈帧无效; W14 和 W15 寻址基本数据空间
bit 1	<b>RND:</b> 舍入模式选择位 1 = 使能有偏 (常规) 舍入 0 = 使能无偏 (收敛) 舍入
bit 0	<b>IF:</b> 整数或小数乘法器模式选择位 1 = 使能整数模式的 DSP 乘法运算 0 = 使能小数模式的 DSP 乘法运算

注 1: 该位将总是读为 0。

2: IPL3 位与 IPL[2:0] 位 (SR[7:5]) 组合形成 CPU 中断优先级。

## 寄存器3-3: CTEXTSTAT: CPU W 寄存器现场状态寄存器

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	CCTXI2	CCTXI1	CCTXI0
bit 15					bit 8		

U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
—	—	—	—	—	MCTXI2	MCTXI1	MCTXI0
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 15-11 未实现: 读为 0

bit 10-8 **CCTXI[2:0]:** 当前 (W 寄存器) 现场标识符位

111 = 保留

...

100 = 当前正在使用备用工作寄存器组 4

011 = 当前正在使用备用工作寄存器组 3

010 = 当前正在使用备用工作寄存器组 2

001 = 当前正在使用备用工作寄存器组 1

000 = 当前正在使用默认工作寄存器组

bit 7-3 未实现: 读为 0

bit 2-0 **MCTXI[2:0]:** 手动 (W 寄存器) 现场标识符位

111 = 保留

...

100 = 最近手动选择了备用工作寄存器组 4

011 = 最近手动选择了备用工作寄存器组 3

010 = 最近手动选择了备用工作寄存器组 2

001 = 最近手动选择了备用工作寄存器组 1

000 = 最近手动选择了默认工作寄存器组

# dsPIC33CK64MP105系列

## 3.4.4 算术逻辑单元（ALU）

dsPIC33CK64MP105系列的ALU为16位宽，能进行加法、减法、移位和逻辑运算。除非另外说明，算术运算本质上采用二进制补码方式进行。根据不同的运算，ALU可能会影响SR寄存器中的进位标志位（C）、全零标志位（Z）、负标志位（N）、溢出标志位（OV）和半进位标志位（DC）的值。在减法运算中，C和DC状态位分别作为借位位和半借位位。

根据所使用的指令模式，ALU可执行8位或16位运算。根据指令的寻址模式，ALU运算的数据可以来自W寄存器阵列或数据存储寄存器。同样，ALU的输出数据可被写入W寄存器阵列或数据存储单元。

关于每条指令所影响的SR位的信息，请参见《16位MCU和DSC程序员参考手册》（[www.microchip.com/DS70000157](http://www.microchip.com/DS70000157)）。

内核CPU融入了对乘法和除法的硬件支持。这包括专门的硬件乘法器以及支持16位除数除法的硬件。

### 3.4.4.1 乘法器

通过使用高速17位x17位乘法器，ALU支持无符号、有符号或混合符号乘法运算的几种MCU乘法模式：

- 16位有符号x16位有符号
- 16位无符号x16位无符号
- 16位有符号x5位无符号（立即数）
- 16位有符号x16位无符号
- 16位无符号x5位无符号（立即数）
- 16位无符号x16位有符号
- 8位无符号x8位无符号

### 3.4.4.2 除法器

除法模块支持具有以下数据长度的32位/16位和16位/16位有符号和无符号整数除法运算：

- 32位有符号/16位有符号除法
- 32位无符号/16位无符号除法
- 16位有符号/16位有符号除法
- 16位无符号/16位无符号除法

16位有符号和无符号DIV指令可为16位除数指定任何W寄存器（Wn），也可为32位被除数指定任何（对齐的）W寄存器对（W(m+1):Wm）。除法运算中处理除数的每一位需要一个周期，因此32位/16位和16位/16位指令的执行周期数相同。还有其他指令：DIV2和DIVF2。除法指令将在6个周期内完成。

## 3.4.5 DSP引擎

DSP引擎由一个高速17位x17位乘法器、一个40位桶形移位寄存器和一个40位加法器/减法器（带两个目标累加器、舍入逻辑和饱和逻辑）组成。

DSP引擎还能执行固有、不需要其他数据的累加器-累加器操作。这些指令是ADD、SUB、NEG、MIN和MAX。

通过CPU内核控制寄存器（CORCON）中的各个位，可以对DSP引擎的操作进行多种选择，这些选择如下：

- 小数或整数DSP乘法（IF）
- 有符号、无符号或混合符号DSP乘法（USx）
- 常规或收敛舍入（RND）
- ACCA自动饱和和使能/禁止（SATA）
- ACCB自动饱和和使能/禁止（SATB）
- 对于写数据存储器的自动饱和和使能/禁止（SATDW）
- 累加器饱和模式选择（ACCSAT）

表3-2: DSP指令汇总

指令	代数运算	ACC回写
CLR	$A = 0$	是
ED	$A = (x - y)^2$	否
EDAC	$A = A + (x - y)^2$	否
MAC	$A = A + (x \cdot y)$	是
MAC	$A = A + x^2$	否
MOVSAC	A中内容将不发生改变	是
MPY	$A = x \cdot y$	否
MPY	$A = x^2$	否
MPY.N	$A = -x \cdot y$	否
MSC	$A = A - x \cdot y$	是



4.0 存储器构成

**注：** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“dsPIC33/PIC24程序存储器”（[www.microchip.com/DS70000613](http://www.microchip.com/DS70000613)）。

dsPIC33CK64MP105系列的架构具有独立的程序和数据存储空间以及总线。这一架构同时还允许在代码执行过程中从数据空间（Data Space，DS）直接访问程序存储器。

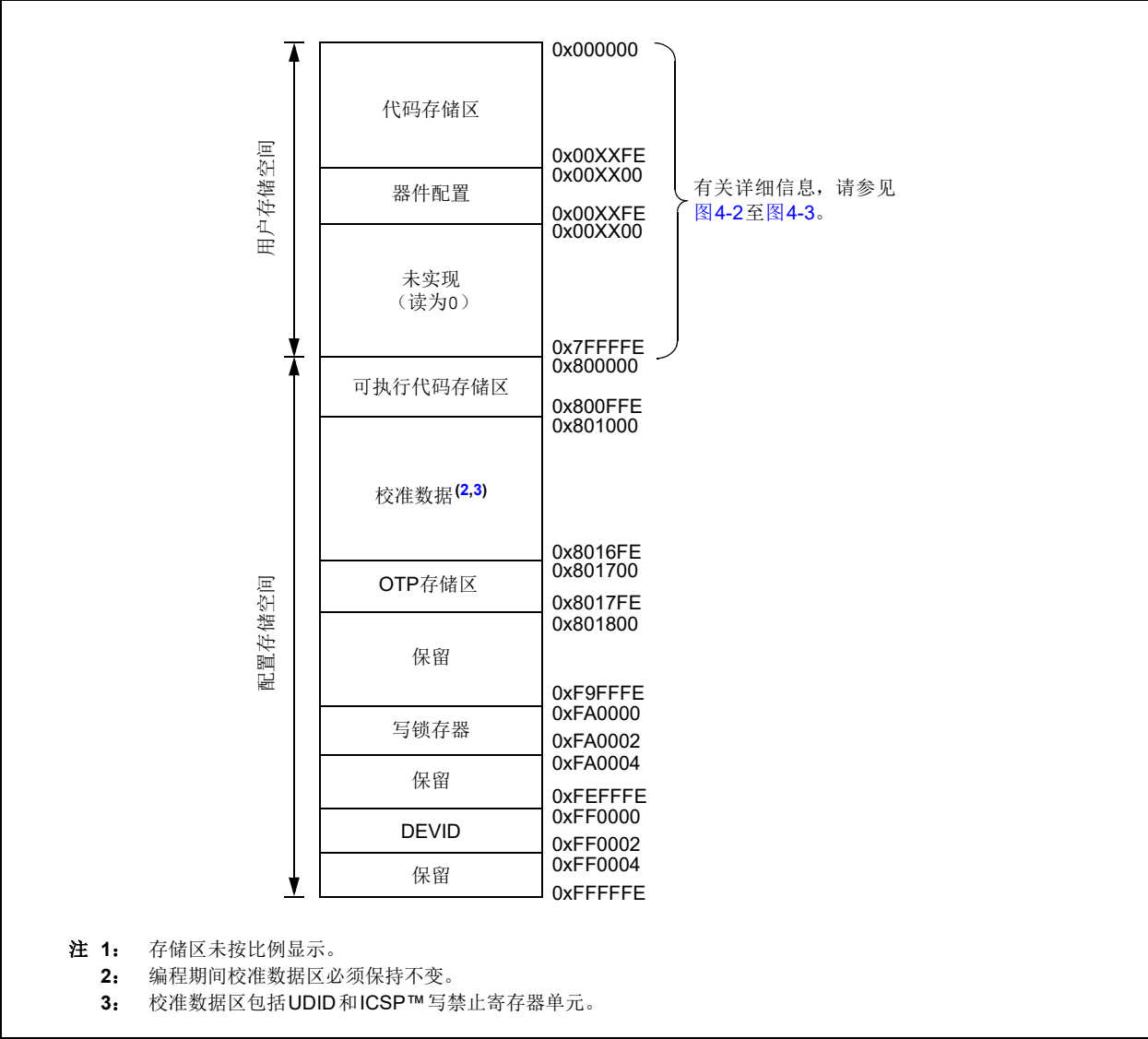
4.1 程序地址空间

dsPIC33CK64MP105系列器件的程序存储空间可存储4M条指令。可通过由程序执行过程中23位PC或者第4.4.5节“程序存储空间与数据存储空间的接口”中所述的表操作或数据空间重映射得到的24位值寻址这一空间。

用户应用只能访问程序存储空间的低半地址范围（0x000000至0x7FFFFFFF）。使用TBLRD指令时，情况有所不同，该指令使用TBLPAG[7]以允许访问配置存储空间中的校准数据和器件ID部分。

图4-1至图4-3给出了dsPIC33CK64MP105器件的程序存储器映射。

图4-1: dsPIC33CK32MP10X器件的程序存储器映射<sup>(1)</sup>



# dsPIC33CK64MP105系列

图4-2: dsPIC33CK64MP10X器件的代码存储区映射<sup>(1)</sup>

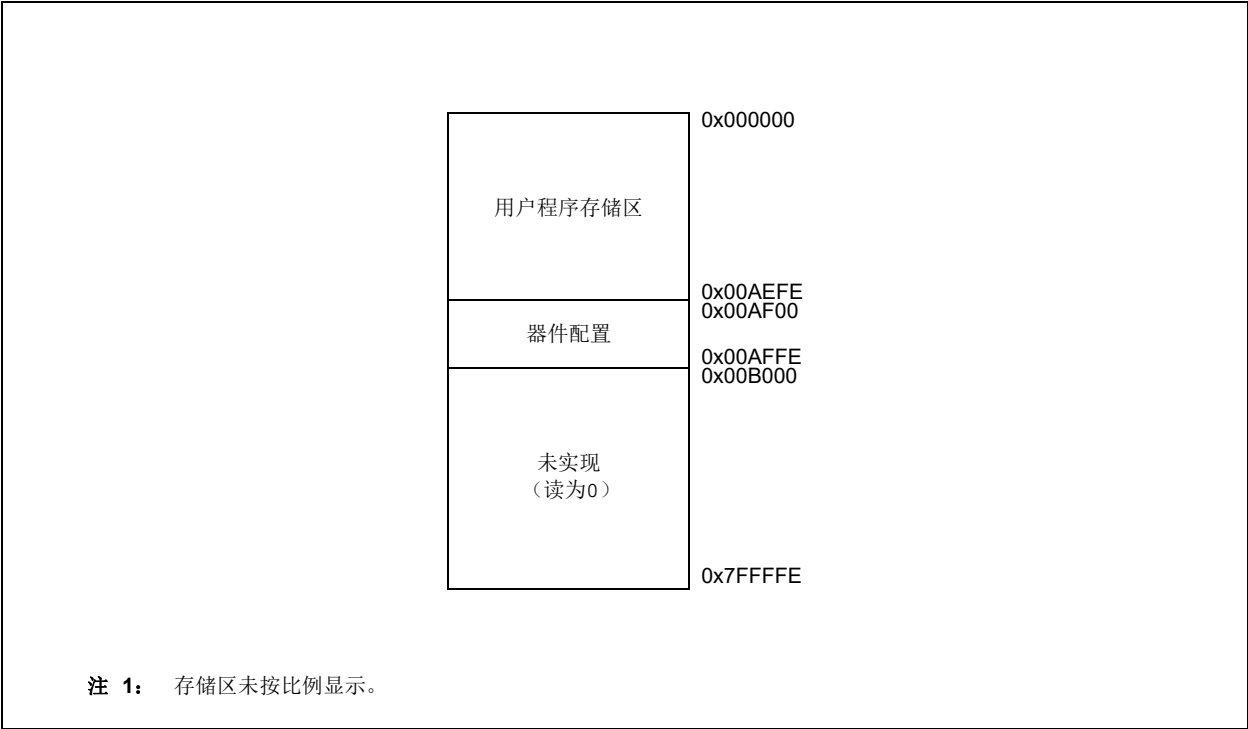
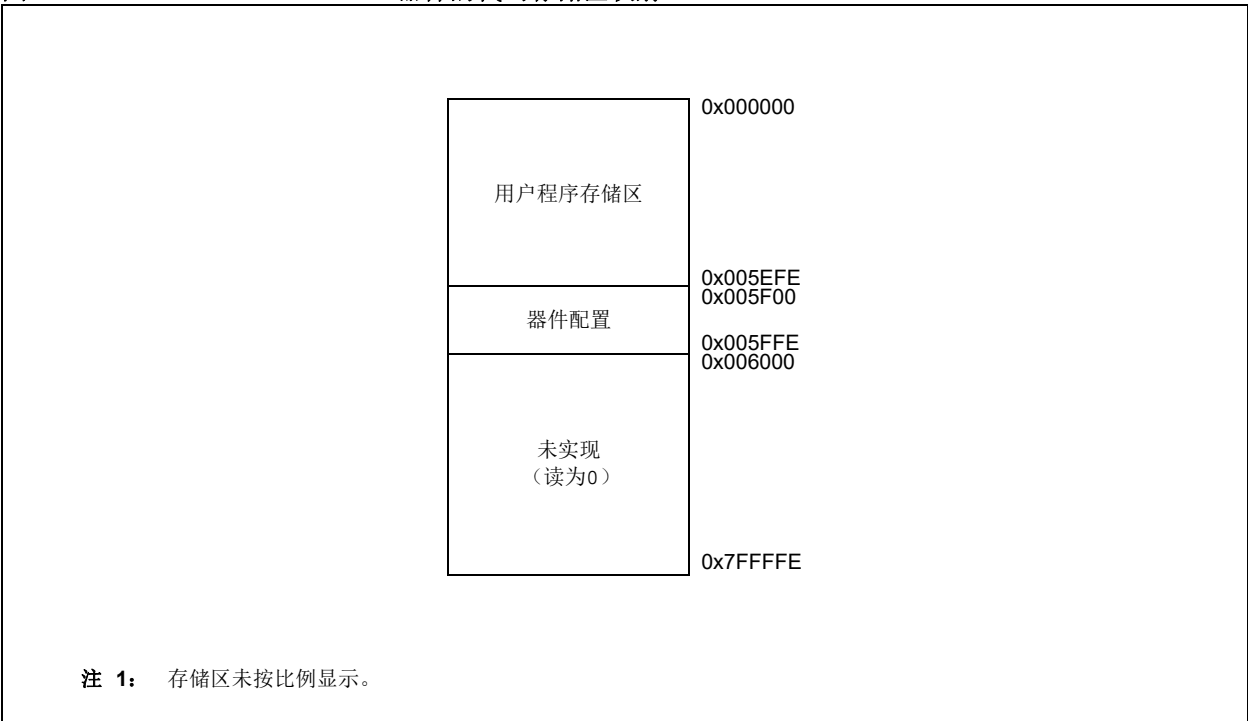


图4-3: dsPIC33CK32MP10X器件的代码存储区映射<sup>(1)</sup>



4.1.1 程序存储器构成

程序存储空间由可字寻址的块构成。虽然它被视为24位宽，但将程序存储器的每个地址单元视作一个低位字和一个高位字的组合更加合理，其中高位字的高字节未实现。低位字的地址始终为偶数，而高位字的地址为奇数（图4-4）。

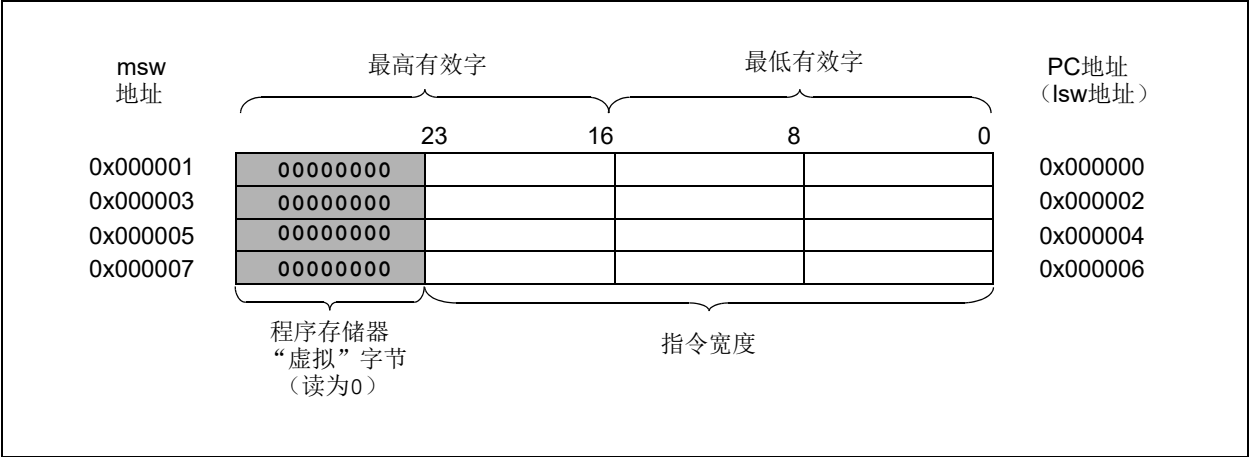
程序存储器地址始终在低位字处按字对齐，并且在代码执行过程中地址将递增或递减2。这种寻址模式也与数据存储空间寻址兼容，且为访问程序存储空间中的数据提供了可能。

4.1.2 中断和陷阱向量

所有dsPIC33CK64MP105系列器件中从0x000000至0x000200之间的地址空间都是保留的，用来存储硬编码的程序执行向量。提供了一个硬件复位向量将代码执行从器件复位时PC的默认值重定位到代码实际起始处。用户应用程序可在闪存地址0x000000处编写一条GOTO指令，以将代码的实际起始地址设置为闪存地址0x000002。

关于中断向量表（Interrupt Vector Table, IVT）更详细的讨论，请参见第7.0节“中断控制器”。

图4-4： 程序存储器构成



4.1.3 器件惟一标识符（UDID）

所有 dsPIC33CK64MP105 系列器件在最终制造期间都单独编码了器件惟一标识符（即UDID）。UDID不能通过批量擦除命令或任何其他用户可访问的方式擦除。凭借该功能，可以在需要时跟踪应用中Microchip 器件的制造信息。应用制造商也可将该功能用于任意个可能需要惟一标识的应用场合，例如：

- 跟踪器件
- 惟一序列号
- 惟一安全密钥

UDID 包含五个24 位程序字。这些程序字组合在一起构成一个120 位的惟一标识符。

UDID 存储在器件配置空间中0x801200 和0x801208 之间的五个只读单元中。表4-1 列出了标识符字的地址并显示了其中的内容。

表4-1： UDID地址

UDID	地址	说明
UDID1	0x801200	UDID 字 1
UDID2	0x801202	UDID 字 2
UDID3	0x801204	UDID 字 3
UDID4	0x801206	UDID 字 4
UDID5	0x801208	UDID 字 5

4.2 数据地址空间

dsPIC33CK64MP105 系列CPU 具有独立的16 位宽数据存储空间。使用独立的地址发生单元（AGU）对数据空间执行读写操作。数据存储器映射如图4-5 所示。

数据存储空间中的所有有效地址（Effective Address，EA）均为16 位宽，指向数据空间内的字节。这种构成方式使得基本数据空间地址范围为64 KB 或32K 字。

数据存储空间的低半地址部分（即当EA[15] = 0 时）用作实现的存储单元，而高半地址部分（EA[15] = 1）则保留为程序空间可视性（PSV）区域。

dsPIC33CK64MP105 系列器件实现了最大16 KB 的数据存储空间。如果EA 指向了该区域以外的存储单元，则将返回一个全零的字或字节。

4.2.1 数据空间宽度

数据存储空间组织为可字节寻址的16 位宽的块。在数据存储器和寄存器中的数据是以16 位字为单位对齐的，但所有数据空间EA 都将解析为字节。每个字的最低有效字节（Least Significant Byte，LSB）具有偶地址，而最高有效字节（Most Significant Byte，MSB）则具有奇地址。

4.2.2 数据存储器构成和对齐方式

为维持与PIC® MCU 器件的向后兼容性和提高数据存储空间的使用效率，dsPIC33CK64MP105 系列指令集支持字和字节操作。字节访问会在内部对按字对齐的存储空间的所有有效地址计算进行调整。例如，对于执行后修改寄存器间接寻址模式[Ws++]，字节操作时，内核将其识别为值Ws + 1；而字操作时，内核将其识别为值Ws + 2。

数据字节读取将读取包含字节的整个字，使用任何EA 的LSb 来确定要选取的字节。选定的字节被放在数据总线的LSB。这就是说，数据存储器和寄存器被组织为两个并行的字节宽的实体，它们共享（字）地址译码，但写入线相互独立。数据字节写操作只写入阵列或寄存器中与字节地址匹配的那一侧。

所有字访问必须按偶地址对齐。不支持不对齐的字数据读取操作，所以在混合字节和字操作时，或者从8 位MCU 代码移植时，必须要小心。如果试图进行不对齐的读或写操作，将产生地址错误陷阱。如果在读操作时产生错误，正在执行的指令将完成；而如果在写操作时产生错误，指令仍将执行，但不会进行写入。无论是哪种情况，都会执行陷阱，允许系统和/或用户应用程序检查地址错误发生之前的机器状态。

所有装入W 寄存器的字节都将被装入LSB，但MSB 不会更改。

提供了一条符号扩展（SE）指令，允许用户应用程序将8 位有符号数据转换为16 位有符号值。或者，对于16 位无符号数据，用户应用程序可以通过在适当地址处执行一条零扩展（ZE）指令清零任何W 寄存器的MSB。

## 4.2.3 SFR空间

Near数据空间的前4 KB存储单元（从0x0000至0x0FFF）主要被特殊功能寄存器（Special Function Register, SFR）占用。dsPIC33CK64MP105系列内核和外设模块使用这些寄存器来控制器件的工作。

SFR分布在受其控制的模块中，通常按模块分组。大部分SFR空间包含未用的地址单元；它们读为0。

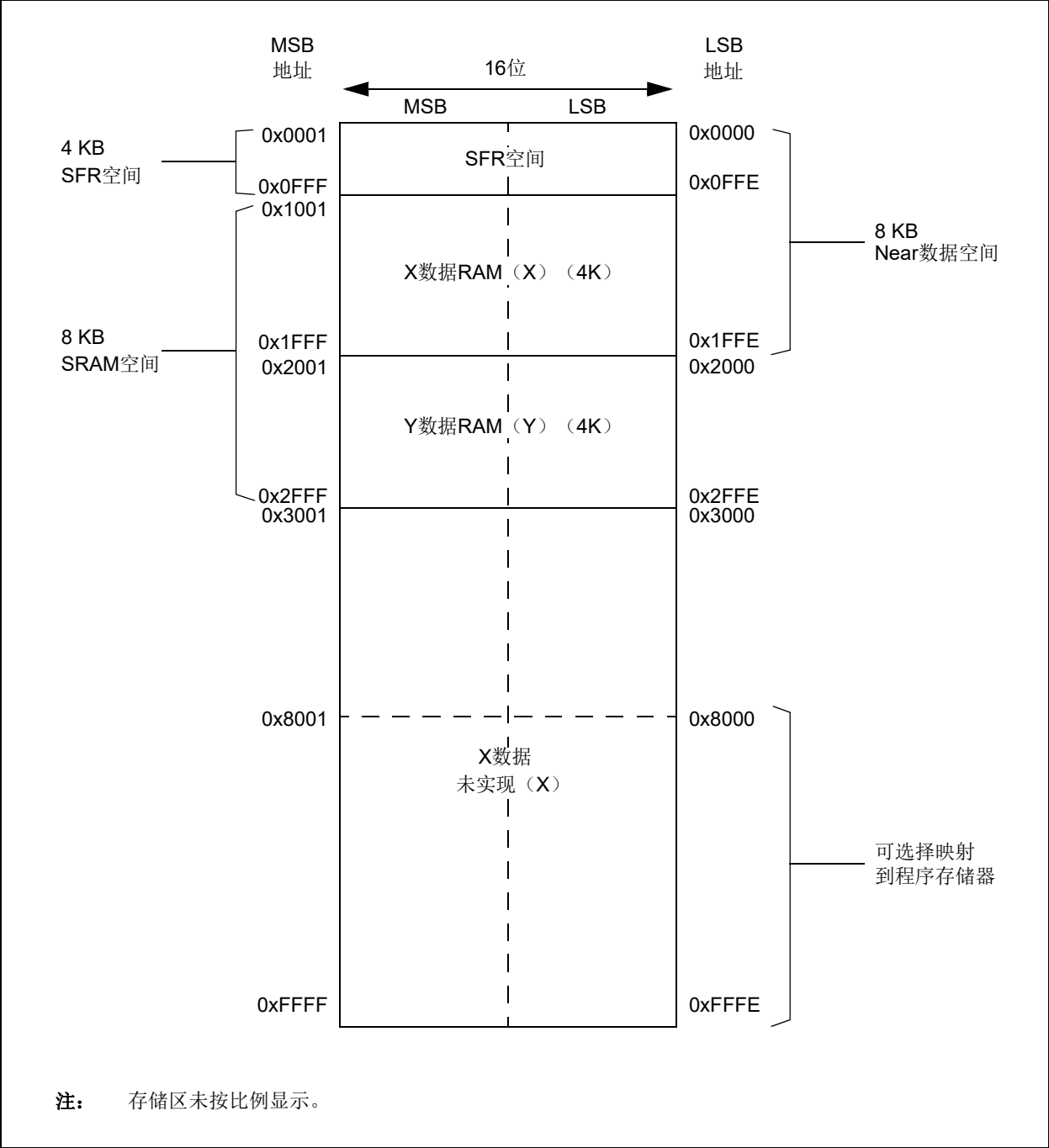
<p><b>注：</b> 不同器件的实际外设功能集和中断也各不相同。关于具体器件的信息，请参见相应器件的特性表和引脚分配图。</p>
--

## 4.2.4 NEAR数据空间

在0x0000和0x1FFF之间的8 KB的区域被称为Near数据空间。可以使用所有存储器直接寻址指令中的13位绝对地址字段直接寻址这一空间中的存储单元。此外，还可以通过支持使用16位地址字段的存储器直接寻址模式的MOV指令、或使用工作寄存器作为地址指针的间接寻址模式寻址整个数据空间。

# dsPIC33CK64MP105系列

图4-5: dsPIC33CK64MPX0X和dsPIC33CK32MPX0X器件的数据存储器映射



#### 4.2.5 X和Y数据空间

dsPIC33CK64MP105系列内核有两个数据空间X和Y。这两个数据空间可以看作是独立的（对于某些DSP指令），或者看作是一个统一的线性地址范围（对于MCU指令）。使用两个地址发生单元（AGU）和独立的数据总线来访问这两个数据空间。此特性允许某些指令同时从RAM中取两个字，因此提高了DSP算法的执行效率，如有限冲激响应（Finite Impulse Response, FIR）滤波器和快速傅立叶变换（Fast Fourier Transform, FFT）。

X数据空间可用于所有指令，并且支持所有寻址模式。X数据空间的读数据总线和写数据总线是独立的。所有将数据空间视为组合的X和Y地址空间的指令均将X读数据总线作为读数据路径。X读数据总线也是双操作数DSP指令（MAC类）的X数据预取路径。

MAC类指令（CLR、ED、EDAC、MAC、MOVSAC、MPY、MPY.N和MSC）将同时使用X数据空间和Y数据空间，从而提供两条同时进行数据读操作的路径。

X和Y数据空间都支持所有指令的模寻址模式，但要受到寻址模式的限制。位反转寻址模式只是在写X数据空间时才支持。

所有数据存储器写操作（包括DSP指令中的数据存储器写操作）均将数据空间视为组合的X和Y地址空间。X和Y数据空间的分界取决于具体的器件，且不能由用户编程。

#### 4.2.6 数据存储器测试（BIST）

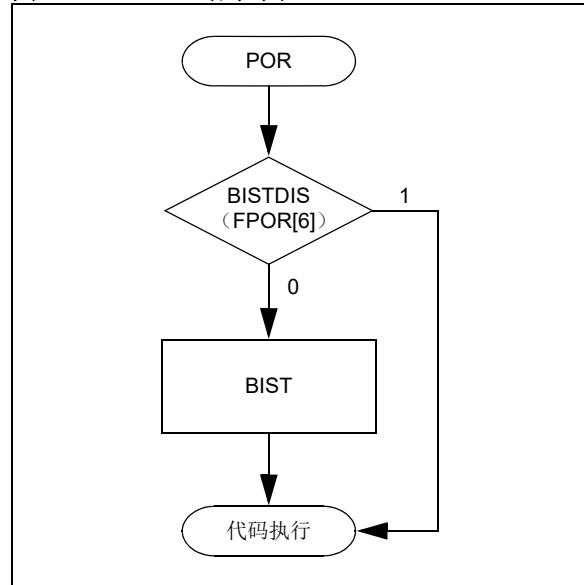
dsPIC33CK64MP105系列具有数据存储器内置自检（Built-In Self-Test, BIST）功能，可选择在启动或运行时运行该功能。存储器自检功能可检测所有存储单元是否正常工作，并提供RAM的通过/失败状态，软件可根据该状态在必要时采取措施。如果报告失败，不会标识测试失败的特定存储单元。

MBISTCON寄存器（寄存器4-1）包含BIST操作的控制和状态位。MBISTDONE位（MBISTCON[7]）指示自上次复位后BIST是否运行，MBISTSTAT位（MBISTCON[4]）提供通过/失败结果。

##### 4.2.6.1 启动时的BIST

BIST可配置为在POR型复位时自动运行，如图4-6所示。默认情况下，当BISTDIS（FPOR[6]）= 1时，BIST被禁止，不会在器件启动时运行。如果BISTDIS位在器件编程期间清零，则BIST将在装入所有配置寄存器之后、开始执行代码之前运行。BIST将始终基于FRC + PLL运行，采用的PLL设置会产生125 MHz的时钟速率。

图4-6: BIST流程图



##### 4.2.6.2 运行时的BIST

可请求BIST在后续器件复位时随时运行。

BIST将损坏所有RAM内容，包括堆栈指针，并且需要后续复位。在执行BIST之前，系统应做好复位准备。通过将MBISTEN位（MBISTCON[0]）置1并执行复位来调用BIST。可防止意外写入MBISTCON寄存器，在写入之前需要解锁序列。每个解锁序列只能将一个位置1。运行时BIST的步骤如下：

1. 通过将0x55和0xAA连续写入NVMKEY寄存器来执行解锁序列。
2. 将0x0001写入MBISTCON SFR。
3. 执行软件RESET命令。
4. 通过读取SWR（RCON[6]）验证是否发生了软件复位（可选）。
5. 验证MBISTDONE位是否置1。
6. 根据MBISTSTAT指示的测试结果采取操作。

# dsPIC33CK64MP105系列

## 4.2.6.3 故障模拟

可以使用一种机制来模拟BIST故障，以便测试故障处理软件。当FLTINJ位在运行时BIST期间置1时，无论测试结果如何，MBISTSTAT位都将置1。BIST故障模拟的步骤如下：

1. 通过将0x55和0xAA连续写入NVMKEY寄存器来执行解锁序列。
2. 将MBISTEN位（MBISTCON[0]）置1。
3. 通过将0x55和0xAA连续写入NVMKEY寄存器来执行第二个解锁序列。
4. 将FLTINJ位（MBISTCON[8]）置1。
5. 执行软件RESET命令。
6. 验证MBISTDONE、MBISTSTAT和FLTINJ位是否均置1。

寄存器4-1: MBISTCON: MBIST控制寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0 <sup>(1)</sup>
—	—	—	—	—	—	—	FLTINJ
bit 15							bit 8

R/W/HS-0 <sup>(1)</sup>	U-0	U-0	R-0	U-0	U-0	U-0	R/W/HC-0 <sup>(2)</sup>
MBISTDONE	—	—	MBISTSTAT	—	—	—	MBISTEN
bit 7							bit 0

图注:	HS = 硬件置1位	HC = 硬件清零位
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 15-9	未实现：读为0
bit 8	<b>FLTINJ</b> : MBIST故障注入控制位 <sup>(1)</sup> 1 = MBIST测试将完成并将MBISTSTAT置1，模拟SRAM测试失败的情况 0 = MBIST测试将正常执行
bit 7	<b>MBISTDONE</b> : MBIST完成状态位 <sup>(1)</sup> 1 = 执行了MBIST操作 0 = 上一个复位序列未发生任何MBIST操作
bit 6-5	未实现：读为0
bit 4	<b>MBISTSTAT</b> : MBIST状态位 1 = 上一次MBIST失败 0 = 上一次MBIST通过；可能尚未测试所有存储器
bit 3-1	未实现：读为0
bit 0	<b>MBISTEN</b> : MBIST使能位 <sup>(2)</sup> 1 = MBIST测试已就绪；MBIST测试将在下一次器件复位时执行 0 = MBIST测试未就绪

- 注 1: 硬件仅在真正的POR复位时复位。  
2: MBIST测试完成后，该位将自清零。



### 4.3 存储器资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

#### 4.3.1 主要资源

- 《dsPIC33/PIC24 系列参考手册》中的“dsPIC33E/PIC24E 程序存储器” ([www.microchip.com/DS70000613](http://www.microchip.com/DS70000613))
- 代码示例
- 应用笔记

- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24 系列参考手册》章节
- 开发工具

### 4.4 SFR映射

下面各表列出了dsPIC33CK64MP105系列SFR的名称、地址和复位值。这些表包含适用于dsPIC33CK64MP105系列的所有寄存器。并非所有器件型号均提供所有寄存器。有关外设可用性的信息，请参见表1和表2。表8-1详细列出了不同器件型号的端口可用性。

表4-2: SFR块000h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
内核			XMODSRT	048	xxxxxxxxxxxxxxxxx0	CRC		
WREG0	000	0000000000000000	XMODEND	04A	xxxxxxxxxxxxxxxxx1	CRCCONL	0B0	--000000010000--
WREG1	002	0000000000000000	YMODSRT	04C	xxxxxxxxxxxxxxxxx0	CRCCONH	0B2	---00000---00000
WREG2	004	0000000000000000	YMODEND	04E	xxxxxxxxxxxxxxxxx1	CRCXORL	0B4	0000000000000000
WREG3	006	0000000000000000	XBREV	050	xxxxxxxxxxxxxxxxxx	CRCXORH	0B6	0000000000000000
WREG4	008	0000000000000000	DISICNT	052	-xxxxxxxxxxxxxxxx00	CRCDATL	0B8	0000000000000000
WREG5	00A	0000000000000000	TBLPAG	054	-----00000000	CRCDATL	0BA	0000000000000000
WREG6	00C	0000000000000000	YPAG	056	-----00000001	CRCWDATL	0BC	0000000000000000
WREG7	00E	0000000000000000	MSTRPR	058	-----00-----0	CRCWDATH	0BE	0000000000000000
WREG8	010	0000000000000000	CTXTSTAT	05A	-----000-----000	CLC		
WREG9	012	0000000000000000	DMTCON	05C	-----000-----000	CLC1CONL	0C0	--0-00--000--000
WREG10	014	0000000000000000	DMTPRECLR	060	xxxxxxxx-----	CLC1CONH	0C2	-----0000
WREG11	016	0000000000000000	DMTCLR	064	-----xxxxxxxx	CLC1SEL	0C4	0000-000-000-000
WREG12	018	0000000000000000	DMTSTAT	068	-----xxx-----x	CLC1GLSL	0C8	0000000000000000
WREG13	01A	0000000000000000	DMTCNTL	06C	xxxxxxxxxxxxxxxxxx	CLC1GLSH	0CA	0000000000000000
WREG14	01C	0000000000000000	DMTCNTH	06E	xxxxxxxxxxxxxxxxxx	CLC2CONL	0CC	--0-00--000--000
WREG15	01E	0001000000000000	DMTHOLDREG	070	xxxxxxxxxxxxxxxxxx	CLC2CONH	0CE	-----0000
SPLIM	020	xxxxxxxxxxxxxxxxxx	DMTPSCNTL	074	xxxxxxxxxxxxxxxxxx	CLC2SELL	0D0	0000-000-000-000
ACCAL	022	xxxxxxxxxxxxxxxxxx	DMTPSCNTH	076	xxxxxxxxxxxxxxxxxx	CLC2GLSL	0D4	0000000000000000
ACCAH	024	xxxxxxxxxxxxxxxxxx	DMTPSINTVL	078	xxxxxxxxxxxxxxxxxx	CLC2GLSH	0D6	0000000000000000
ACCAU	026	xxxxxxxxxxxxxxxxxx	DMTPSINTVH	07A	xxxxxxxxxxxxxxxxxx	CLC3CONL	0D8	--0-00--000--000
ACCB	028	xxxxxxxxxxxxxxxxxx	SENT			CLC3CONH	0DA	-----0000
ACCBH	02A	xxxxxxxxxxxxxxxxxx	SENT1CON1	080	--0-000000-0-000	CLC3SELL	0DC	0000-000-000-000
ACCBU	02C	xxxxxxxxxxxxxxxxxx	SENT1CON2	084	0000000000000000	CLC3GLSL	0DE	0000000000000000
PCL	02E	0000000000000000	SENT1CON3	088	0000000000000000	CLC3GLSH	0E2	0000000000000000
PCH	030	-----00000000	SENT1STAT	08C	-----00000000	CLC4CONL	0E4	--0-00--000--000
DSRPAG	032	-----000000001	SENT1SYNC	090	0000000000000000	CLC4CONH	0E6	-----0000
DSWPAG	034	-----000000001	SENT1DATL	094	0000000000000000	CLC4SELL	0E8	0000-000-000-000
RCOUNT	036	xxxxxxxxxxxxxxxxxx	SENT1DATH	096	0000000000000000	CLC4GLSL	0EC	0000000000000000
DCOUNT	038	xxxxxxxxxxxxxxxxxx	SENT2CON1	098	--0-000000-0-000	CLC4GLSH	0EE	0000000000000000
DOSTARTL	03A	xxxxxxxxxxxxxxxxx0	SENT2CON2	09C	0000000000000000	ECC		
DOSTARTH	03C	-----xxxxxxxx	SENT2CON3	0A0	0000000000000000	ECCCONL	0F0	-----0
DOENDL	03E	xxxxxxxxxxxxxxxxx0	SENT2STAT	0A4	-----00000000	ECCCONH	0F2	0000000000000000
DOENDH	040	-----xxxxxxxx	SENT2SYNC	0A8	0000000000000000	ECCADDRH	0F4	0000000000000000
SR	042	0000000000000000	SENT2DATL	0AC	0000000000000000	ECCADDRH	0F6	0000000000000000
CORCON	044	--xx000000100000	SENT2DATH	0AE	0000000000000000	ECCSTATL	0F8	0000000000000000
MODCON	046	0--0000000000000				ECCSTATH	0FA	-----0000000000

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-3: SFR块100h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
定时器			INT1TMRH	15E	0000000000000000	POS2HLD	186	0000000000000000
T1CON	100	--0000000-00-00-	INT1HLDL	160	0000000000000000	VEL2CNT	188	0000000000000000
TMR1	104	0000000000000000	INT1HLDH	162	0000000000000000	VEL2CNTH	18A	0000000000000000
PR1	108	0000000000000000	INDX1CNTL	164	0000000000000000	VEL2HLD	18E	0000000000000000
QEI			INDX1CNTH	166	0000000000000000	INT2TMRL	190	0000000000000000
QEI1CON	140	--000000-0000000	INDX1HLD	16A	0000000000000000	INT2TMRH	192	0000000000000000
QEI1IOC	144	000000000000xxxx	QEI1GECL/ QEI1ICL	16C	0000000000000000	INT2HLDL	194	0000000000000000
QEI1IOCH	146	-----0	QEI1GECH/ QEI1ICH	16E	0000000000000000	INT2HLDH	196	0000000000000000
QEI1STAT	148	--000000000000000	QEI1LECL	170	0000000000000000	INDX2CNTL	198	0000000000000000
POS1CNTL	14C	0000000000000000	QEI1LECH	172	0000000000000000	INDX2CNTH	19A	0000000000000000
POS1CNTH	14E	0000000000000000	QEI2CON	174	--000000-0000000	INDX2HLD	19E	0000000000000000
POS1HLD	152	0000000000000000	QEI2IOC	178	000000000000xxxx	QEI2GECL/ QEI2ICL	1A0	0000000000000000
VEL1CNT	154	0000000000000000	QEI2IOCH	17A	-----0	QEI2GECH/ QEI2ICH	1A2	0000000000000000
VEL1CNTH	156	0000000000000000	QEI2STAT	17C	--000000000000000	QEI2LECL	1A4	0000000000000000
VEL1HLD	15A	0000000000000000	POS2CNTL	180	0000000000000000	QEI2LECH	1A6	0000000000000000
INT1TMRL	15C	0000000000000000	POS2CNTH	182	0000000000000000			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-4: SFR块200h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
<b>I2C1和I2C2</b>			U1SCCON	258	-----00000~	SPI1IMSKH	2C2	--0000000-000000
I2C1CONL	200	--01000000000000	U1SCINT	25A	--00-000--00-000	SPI1URDTL	2C4	0000000000000000
I2C1CONH	202	-----00000000	U1INT	25C	-----00---0--	SPI1URDTH	2C6	0000000000000000
I2C1STAT	204	000--00000000000	U2MODE	260	--000-0000000000	SPI2CON1L	2C8	--00000000000000
I2C1ADD	208	-----0000000000	U2MODEH	262	00---00000000000	SPI2CON1H	2CA	0000000000000000
I2C1MSK	20C	-----0000000000	U2STA	264	0000000010000000	SPI2CON2L	2CC	-----00000
I2C1BRG	210	0000000000000000	U2STAH	266	0000-00000101110	SPI2CON2H	2CE	-----
I2C1TRN	214	-----11111111	U2BRG	268	0000000000000000	SPI2STATL	2D0	--00--0001-1-00
I2C1RCV	218	-----00000000	U2BRGH	26A	-----0000	SPI2STATH	2D2	--000000--000000
I2C2CONL	21C	--01000000000000	U2RXREG	26C	-----xxxxxxxx	SPI2BUFL	2D4	0000000000000000
I2C2CONH	21E	-----00000000	U2TXREG	270	-----xxxxxxxx	SPI2BUFH	2D6	0000000000000000
I2C2STAT	220	000--00000000000	U2P1	274	-----0000000000	SPI2BRGL	2D8	--xxxxxxxxxxxxxxxx
I2C2ADD	224	-----0000000000	U2P2	276	-----0000000000	SPI2BRGH	2DA	-----
I2C2MSK	228	-----0000000000	U2P3	278	0000000000000000	SPI2IMSKL	2DC	--00--0000-0-00
I2C2BRG	22C	0000000000000000	U2P3H	27A	-----00000000	SPI2IMSKH	2DE	--0000000-000000
I2C2TRN	230	-----11111111	U2TXCHK	27C	-----00000000	SPI2URDTL	2E0	0000000000000000
I2C2RCV	234	-----00000000	U2RXCHK	27E	-----00000000	SPI2URDTH	2E2	0000000000000000
<b>UART1和UART2</b>			U2SCCON	280	-----00000~	SPI3CON1L	2E4	--00000000000000
U1MODE	238	--000-0000000000	U2SCINT	282	--00-000--00-000	SPI3CON1H	2E6	0000000000000000
U1MODEH	23A	00---00000000000	U2INT	284	-----00---0--	SPI3CON2L	2E8	-----00000
U1STA	23C	0000000010000000	<b>SPI</b>			SPI3CON2H	2EA	-----
U1STAH	23E	0000-00000101110	SPI1CON1L	2AC	--00000000000000	SPI3STATL	2EC	--00--0001-1-00
U1BRG	240	0000000000000000	SPI1CON1H	2AE	0000000000000000	SPI3STATH	2EE	--000000--000000
U1BRGH	242	-----0000	SPI1CON2L	2B0	-----00000	SPI3BUFL	2F0	0000000000000000
U1RXREG	244	-----xxxxxxxx	SPI1CON2H	2B2	-----	SPI3BUFH	2F2	0000000000000000
U1TXREG	248	-----xxxxxxxx	SPI1STATL	2B4	--00--0001-1-00	SPI3BRGL	2F4	--xxxxxxxxxxxxxxxx
U1P1	24C	-----0000000000	SPI1STATH	2B6	--000000--000000	SPI3BRGH	2F6	-----
U1P2	24E	-----0000000000	SPI1BUFL	2B8	0000000000000000	SPI3IMSKL	2F8	--00--0000-0-00
U1P3	250	0000000000000000	SPI1BUFH	2BA	0000000000000000	SPI3IMSKH	2FA	--0000000-000000
U1P3H	252	-----00000000	SPI1BRGL	2BC	--xxxxxxxxxxxxxxxx	SPI3URDTL	2FC	0000000000000000
U1TXCHK	254	-----00000000	SPI1BRGH	2BE	-----	SPI3URDTH	2F3	0000000000000000
U1RXCHK	256	-----00000000	SPI1IMSKL	2C0	--00--0000-0-00			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-5: SFR块300h-400h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
高速PWM			PG1TRIGB	356	0000000000000000	PG3FFPCIH	3AE	0000-000000000000
PCLKCON	300	00-----0--00--00	PG1TRIGC	358	0000000000000000	PG3SPCIL	3B0	0000000000000000
FSCL	302	0000000000000000	PG1DTL	35A	--0000000000000000	PG3SPCIH	3B2	0000-000000000000
FSMINPER	304	0000000000000000	PG1DTH	35C	--0000000000000000	PG3LEBL	3B4	0000000000000000
MPHASE	306	0000000000000000	PG1CAP	35E	0000000000000000	PG3LEBH	3B6	-----000-----0000
MDC	308	0000000000000000	PG2CONL	360	-----0000--000000	PG3PHASE	3B8	0000000000000000
MPER	30A	0000000000000000	PG2CONH	362	000-000000--0000	PG3DC	3BA	0000000000000000
LFISR	30C	0000000000000000	PG2STAT	364	0000000000000000	PG3DCA	3BC	-----00000000
CMBTRIGL	30E	-----00000000	PG2IOCONL	366	0000000000000000	PG3PER	3BE	0000000000000000
CMBTRIGH	310	-----00000000	PG2IOCONH	368	0000---0--000000	PG3TRIGA	3C0	0000000000000000
LOGCONA	312	000000000000-000	PG2EVTL	36A	00000000---00000	PG3TRIGB	3C2	0000000000000000
LOGCONB	314	000000000000-000	PG2EVTH	36C	0000--0000000000	PG3TRIGC	3C4	0000000000000000
LOGCONC	316	000000000000-000	PG2FPCIL	36E	0000000000000000	PG3DTL	3C6	--00000000000000
LOGCOND	318	000000000000-000	PG2FPCIH	370	0000-000000000000	PG3DTH	3C8	--00000000000000
LOGCONE	31A	000000000000-000	PG2CLPCIL	372	0000000000000000	PG3CAP	3CA	0000000000000000
LOGCONF	31C	000000000000-000	PG2CLPCIH	374	0000-000000000000	PG4CONL	3CC	-----0000--000000
PWMEVTA	31E	0000----0000-000	PG2FFPCIL	376	0000000000000000	PG4CONH	3CE	000-000000--0000
PWMEVTB	320	0000----0000-000	PG2FFPCIH	378	0000-000000000000	PG4STAT	3D0	0000000000000000
PWMEVTC	322	0000----0000-000	PG2SPCIL	37A	0000000000000000	PG4IOCONL	3D2	0000000000000000
PWMEVTD	324	0000----0000-000	PG2SPCIH	37C	0000-000000000000	PG4IOCONH	3D4	0000---0--000000
PWMEVTE	326	0000----0000-000	PG2LEBL	37E	0000000000000000	PG4EVTL	3D6	00000000---000000
PWMEVTF	328	0000----0000-000	PG2LEBH	380	-----000-----0000	PG4EVTH	3D8	0000--0000000000
PG1CONL	32A	-----0000--000000	PG2PHASE	382	0000000000000000	PG4FPCIL	3DA	0000000000000000
PG1CONH	32C	000-000000--0000	PG2DC	384	0000000000000000	PG4FPCIH	3DC	0000-000000000000
PG1STAT	32E	0000000000000000	PG2DCA	386	-----00000000	PG4CLPCIL	3DE	0000000000000000
PG1IOCONL	330	0000000000000000	PG2PER	388	0000000000000000	PG4CLPCIH	3E0	0000-000000000000
PG1IOCONH	332	0000---0--000000	PG2TRIGA	38A	0000000000000000	PG4FFPCIL	3E2	0000000000000000
PG1EVTL	334	00000000---000000	PG2TRIGB	38C	0000000000000000	PG4FFPCIH	3E4	0000-000000000000
PG1EVTH	336	0000--0000000000	PG2TRIGC	38E	0000000000000000	PG4SPCIL	3E6	0000000000000000
PG1FPCIL	338	0000000000000000	PG2DTL	390	--0000000000000000	PG4SPCIH	3E8	0000-000000000000
PG1FPCIH	33A	0000-000000000000	PG2DTH	392	--0000000000000000	PG4LEBL	3EA	0000000000000000
PG1CLPCIL	33C	0000000000000000	PG2CAP	394	0000000000000000	PG4LEBH	3EC	-----000-----0000
PG1CLPCIH	33E	0000-000000000000	PG3CONL	396	-----0000--000000	PG4PHASE	3EE	0000000000000000
PG1FFPCIL	340	0000000000000000	PG3CONH	398	000-000000--0000	PG4DC	3F0	0000000000000000
PG1FFPCIH	342	0000-000000000000	PG3STAT	39A	0000000000000000	PG4DCA	3F2	-----00000000
PG1SPCIL	344	0000000000000000	PG3IOCONL	39C	0000000000000000	PG4PER	3F4	0000000000000000
PG1SPCIH	346	0000-000000000000	PG3IOCONH	39E	0000---0--000000	PG4TRIGA	3F6	0000000000000000
PG1LEBL	348	0000000000000000	PG3EVTL	3A0	00000000---000000	PG4TRIGB	3F8	0000000000000000
PG1LEBH	34A	-----000-----0000	PG3EVTH	3A2	0000--0000000000	PG4TRIGC	3FA	0000000000000000
PG1PHASE	34C	0000000000000000	PG3FPCIL	3A4	0000000000000000	PG4DTL	3FC	--00000000000000
PG1DC	34E	0000000000000000	PG3FPCIH	3A6	0000-000000000000	PG4DTH	3FE	--00000000000000
PG1DCA	350	-----00000000	PG3CLPCIL	3A8	0000000000000000	PG4CAP	400	0000000000000000
PG1PER	352	0000000000000000	PG3CLPCIH	3AA	0000-000000000000			
PG1TRIGA	354	0000000000000000	PG3FFPCIL	3AC	0000000000000000			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-6: SFR块800h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
中断			IPC4	848	-100-100-100-100	IPC32	880	-----100
IFS0	800	0000000000-00000	IPC5	84A	-100-----100	IPC42	894	-100-100-100----
IFS1	802	-00000-00-000000	IPC6	84C	-100-100-----100	IPC43	896	-100-100-100-100
IFS2	804	--000-00-0000----	IPC7	84E	-----100-100-100	IPC44	898	-100-100-100-100
IFS3	806	0--00000-0--0000	IPC8	850	-100-----	IPC45	89A	-----100
IFS4	808	000-0----0000-00	IPC9	852	-----100-100-100	IPC47	89E	-100-100-100----
IFS5	80A	000000000000000-	IPC10	854	-100-----100-100	INTCON1	8C0	0000000000-0000-
IFS6	80C	0000000000000000	IPC11	856	-----100-100	INTCON2	8C2	000----0----0000
IFS7	80E	000000000000----	IPC12	858	-100-100-100-100	INTCON3	8C4	-----00---0---0
IFS8	810	-----0	IPC13	85A	-----100-----	INTCON4	8C6	-----00
IFS10	814	0000000-----	IPC14	85C	-100-100-100-100	INTTREG	8C8	000-0000-0000000
IFS11	816	000-----00000	IPC15	85E	-100-----100	闪存		
IEC0	820	0000000000-00000	IPC16	860	-100-----100-100	NVMCON	8D0	0000-000----0000
IEC1	822	-00000-00-000000	IPC17	862	-----100-100-100	NVMADR	8D2	0000000000000000
IEC2	824	--000-00-0000----	IPC18	864	-100-----	NVMADRU	8D4	-----00000000
IEC3	826	0--00000-0--0000	IPC19	866	-100-100-100----	NVMKEY	8D6	-----00000000
IEC4	828	000-0----0000-00	IPC20	868	-100-100-100----	NVMSRCADRL	8D8	0000000000000000
IEC5	82A	000000000000000-	IPC21	86A	-100-100-100-100	NVMSRCADRH	8DA	-----00000000
IEC6	82C	0000000000000000	IPC22	86C	-100-100-100-100	CBG		
IEC7	82E	000000000000----	IPC23	86E	-100-100-100-100	AMPCON1L	8DC	-----000
IEC8	830	-----0	IPC24	870	-100-100-100-100	AMPCON1H	8DE	-----000
IEC10	834	0000000-----	IPC25	872	-100-100-100-100	BIASCON	8F0	-----0000
IEC11	836	000-----00000	IPC26	874	-100-100-100-100	IBIASCONL	8F4	--000000--000000
IPC0	840	-100-100-100-100	IPC27	876	-100-100-100-100	IBIASCONH	8F6	--000000--000000
IPC1	842	-100-100-----100	IPC29	87A	-100-100-100-100			
IPC2	844	-100-100-100-100	IPC30	87C	-100-100-100-100			
IPC3	846	-100-100-100-100	IPC31	87E	-100-100-100-100			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-7: SFR块 900h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
<b>PTG</b>			CCP1CON3H	95A	0000-----0-00--	CCP3PRL	9AC	1111111111111111
PTGCST	900	--00-00000x---00	CCP1STATL	95C	-----0--00xx0000	CCP3PRH	9AE	1111111111111111
PTGCON	902	000000000000-000	CCP1STATH	95E	-----000000	CCP3RA	9B0	0000000000000000
PTGBTE	904	xxxxxxxxxxxxxxxxxx	CCP1TMRL	960	0000000000000000	CCP3RB	9B4	0000000000000000
PTGBTEH	906	0000000000000000	CCP1TMRH	962	0000000000000000	CCP3BUFL	9B8	0000000000000000
PTGHOLD	908	0000000000000000	CCP1PRL	964	1111111111111111	CCP3BUFH	9BA	0000000000000000
PTGTOLIM	90C	0000000000000000	CCP1PRH	966	1111111111111111	CCP4CON1L	9BC	--00000000000000
PTGT1LIM	910	0000000000000000	CCP1RA	968	0000000000000000	CCP4CON1H	9BE	00--000000000000
PTGSDLIM	914	0000000000000000	CCP1RB	96C	0000000000000000	CCP4CON2L	9C0	00-0----00000000
PTGC0LIM	918	0000000000000000	CCP1BUFL	970	0000000000000000	CCP4CON2H	9C2	-----100-000000
PTGC1LIM	91C	0000000000000000	CCP1BUFH	972	0000000000000000	CCP4CON3H	9C6	0000-----0-00--
PTGADJ	920	0000000000000000	CCP2CON1L	974	--00000000000000	CCP4STATL	9C8	-----0--00xx0000
PTGL0	924	0000000000000000	CCP2CON1H	976	00--000000000000	CCP4STATH	9CA	-----000000
PTGQPTR	928	-----000000	CCP2CON2L	978	00-0----00000000	CCP4TMRL	9CC	0000000000000000
PTGQUE0	930	xxxxxxxxxxxxxxxxxx	CCP2CON2H	97A	0-----100-000000	CCP4TMRH	9CE	0000000000000000
PTGQUE1	932	xxxxxxxxxxxxxxxxxx	CCP2CON3H	97E	0000-----0-00--	CCP4PRL	9D0	1111111111111111
PTGQUE2	934	xxxxxxxxxxxxxxxxxx	CCP2STATL	980	-----0--00xx0000	CCP4PRH	9D2	1111111111111111
PTGQUE3	936	xxxxxxxxxxxxxxxxxx	CCP2STATH	982	-----000000	CCP4RA	9D4	0000000000000000
PTGQUE4	938	xxxxxxxxxxxxxxxxxx	CCP2TMRL	984	0000000000000000	CCP4RB	9D8	0000000000000000
PTGQUE5	93A	xxxxxxxxxxxxxxxxxx	CCP2TMRH	986	0000000000000000	CCP4BUFL	9DC	0000000000000000
PTGQUE6	93C	xxxxxxxxxxxxxxxxxx	CCP2PRL	988	1111111111111111	CCP4BUFH	9DE	0000000000000000
PTGQUE7	93E	xxxxxxxxxxxxxxxxxx	CCP2PRH	98A	1111111111111111	CCP5CON1L	9E0	--00000000000000
PTGQUE8	940	xxxxxxxxxxxxxxxxxx	CCP2RA	98C	0000000000000000	CCP5CON1H	9E2	00--000000000000
PTGQUE9	942	xxxxxxxxxxxxxxxxxx	CCP2RB	990	0000000000000000	CCP5CON2L	9E4	00-0----00000000
PTGQUE10	944	xxxxxxxxxxxxxxxxxx	CCP2BUFL	994	0000000000000000	CCP5CON2H	9E6	-----100-000000
PTGQUE11	946	xxxxxxxxxxxxxxxxxx	CCP2BUFH	996	0000000000000000	CCP5CON3H	9EA	0000-----0-00--
PTGQUE12	948	xxxxxxxxxxxxxxxxxx	CCP3CON1L	998	--00000000000000	CCP5STATL	9EC	-----0--00xx0000
PTGQUE13	94A	xxxxxxxxxxxxxxxxxx	CCP3CON1H	99A	00--000000000000	CCP5STATH	9EE	-----000000
PTGQUE14	94C	xxxxxxxxxxxxxxxxxx	CCP3CON2L	99C	00-0----00000000	CCP5TMRL	9F0	0000000000000000
PTGQUE15	94E	xxxxxxxxxxxxxxxxxx	CCP3CON2H	99E	-----100-000000	CCP5TMRH	9F2	0000000000000000
<b>CCP</b>			CCP3CON3H	9A2	0000-----0-00--	CCP5PRL	9F4	1111111111111111
CCP1CON1L	950	--00000000000000	CCP3STATL	9A4	-----0--00xx0000	CCP5PRH	9F6	1111111111111111
CCP1CON1H	952	00--000000000000	CCP3STATH	9A6	-----000000	CCP5RA	9F8	0000000000000000
CCP1CON2L	954	00-0----00000000	CCP3TMRL	9A8	0000000000000000	CCP5RB	9FC	0000000000000000
CCP1CON2H	956	-----100-000000	CCP3TMRH	9AA	0000000000000000			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-8: SFR块A00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
<b>CCP (续)</b>			DMASRC0	AC8	0000000000000000	DMASRC2	ADC	0000000000000000
CCP5BUFL	A00	0000000000000000	DMADST0	ACA	0000000000000000	DMADST2	ADE	0000000000000000
CCP5BUFH	A02	0000000000000000	DMACNT0	ACC	0000000000000001	DMACNT2	AE0	0000000000000001
<b>DMA</b>			DMACH1	ACE	-----0000000000	DMACH3	AE2	-----0000000000
DMACON	ABC	--0-----0	DMAINT1	AD0	--0000000000--0	DMAINT3	AE4	--0000000000--0
DMABUF	ABE	0000000000000000	DMASRC1	AD2	0000000000000000	DMASRC3	AE6	0000000000000000
DMAL	AC0	0000000000000000	DMADST1	AD4	0000000000000000	DMADST3	AE8	0000000000000000
DMAH	AC2	0000000000000000	DMACNT1	AD6	0000000000000001	DMACNT3	AEA	0000000000000001
DMACH0	AC4	-----0000000000	DMACH2	AD8	-----0000000000			
DMAINT0	AC6	--0000000000--0	DMAINT2	ADA	--0000000000--0			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

表4-9: SFR块B00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
<b>ADC</b>			ADCMPL0	B44	0000000000000000	ADTRIG2H	B8A	0000000000000000
ADCON1L	B00	000-00000-000	ADCMPL1H	B46	0000000000000000	ADTRIG3L	B8C	0000000000000000
ADCON1H	B02	-----011-----	ADCMPL2ENL	B48	0000000000000000	ADTRIG3H	B8E	0000000000000000
ADCON2L	B04	00-00000000000000	ADCMPL2ENH	B4A	-----0000000000	ADTRIG4L	B90	0000000000000000
ADCON2H	B06	00-00000000000000	ADCMPL2LO	B4C	0000000000000000	ADTRIG4H	B92	0000000000000000
ADCON3L	B08	0000000000000000	ADCMPL2HI	B4E	0000000000000000	ADTRIG5L	B94	0000000000000000
ADCON3H	B0A	000000000-xxxx	ADCMPL3ENL	B50	0000000000000000	ADTRIG5H	B96	0000000000000000
ADCON4L	B0C	-----000-----xx	ADCMPL3ENH	B52	-----0000000000	ADTRIG6L	B98	0000000000000000
ADCON4H	B0E	00-----0000	ADCMPL3LO	B54	0000000000000000	ADCMPL0CON	BA0	0000000000000000
ADMOD0L	B10	0000000000000000	ADCMPL3HI	B56	0000000000000000	ADCMPL1CON	BA4	0000000000000000
ADMOD0H	B12	0000000000000000	ADFL0DAT	B68	0000000000000000	ADCMPL2CON	BA8	0000000000000000
ADMOD1L	B14	0000000000000000	ADFL0CON	B6A	xxx0000000000000	ADCMPL3CON	BAC	0000000000000000
ADMOD1H	B16	-----0000	ADFL1DAT	B6C	0000000000000000	ADLVLTRGL	BD0	0000000000000000
ADIEL	B20	xxxxxxxxxxxxxxxxxx	ADFL1CON	B6E	xxx0000000000000	ADLVLTRGH	BD2	-----xxxxxxxxxxxx
ADIEH	B22	-----xxxxxxxxxxxx	ADFL2DAT	B70	0000000000000000	ADCORE0L	BD4	0000000000000000
ADSTATL	B30	0000000000000000	ADFL2CON	B72	xxx0000000000000	ADCORE0H	BD6	0000001100000000
ADSTATH	B32	-----0000000000	ADFL3DAT	B74	0000000000000000	ADCORE1L	BD8	0000000000000000
ADCMPL0ENL	B38	0000000000000000	ADFL3CON	B76	xxx0000000000000	ADCORE1H	BDA	0000001100000000
ADCMPL0ENH	B3A	-----0000000000	ADTRIG0L	B80	0000000000000000	ADEIEL	BF0	xxxxxxxxxxxxxxxxxx
ADCMPL0LO	B3C	0000000000000000	ADTRIG0H	B82	0000000000000000	ADEIEH	BF2	-----xxxxxxxxxxxx
ADCMPL0HI	B3E	0000000000000000	ADTRIG1L	B84	0000000000000000	ADEISTATL	BF8	xxxxxxxxxxxxxxxxxx
ADCMPL1ENL	B40	0000000000000000	ADTRIG1H	B86	0000000000000000	ADEISTATH	BFA	-----xxxxxxxxxxxx
ADCMPL1ENH	B42	-----0000000000	ADTRIG2L	B88	0000000000000000			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-10: SFR块 C00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
ADC (续)			ADCBUF14	C28	0000000000000000	SLP1DAT	C94	0000000000000000
ADCON5L	C00	-----0-----	ADCBUF15	C2A	0000000000000000	DAC2CONL	C98	000--000x0000000
ADCON5H	C02	----xxxx0-----	ADCBUF16	C2C	0000000000000000	DAC2CONH	C9A	-----0000000000
ADCBUF0	C0C	0000000000000000	ADCBUF17	C2E	0000000000000000	DAC2DATL	C9C	0000000000000000
ADCBUF1	C0E	0000000000000000	ADCBUF18	C30	0000000000000000	DAC2DATH	C9E	0000000000000000
ADCBUF2	C10	0000000000000000	ADCBUF19	C32	0000000000000000	SLP2CONL	CA0	0000000000000000
ADCBUF3	C12	0000000000000000	ADCBUF20	C34	0000000000000000	SLP2CONH	CA2	----000-----
ADCBUF4	C14	0000000000000000	DAC			SLP2DAT	CA4	0000000000000000
ADCBUF5	C16	0000000000000000	DACCTRL1L	C80	--0-----0000-000	DAC3CONL	CA8	000--000x0000000
ADCBUF6	C18	0000000000000000	DACCTRL2L	C84	-----0001010101	DAC3CONH	CAA	-----0000000000
ADCBUF7	C1A	0000000000000000	DACCTRL2H	C86	-----0010001010	DAC3DATL	CAC	0000000000000000
ADCBUF8	C1C	0000000000000000	DAC1CONL	C88	000--000x0000000	DAC3DATH	CAE	0000000000000000
ADCBUF9	C1E	0000000000000000	DAC1CONH	C8A	-----0000000000	SLP3CONL	CB0	0000000000000000
ADCBUF10	C20	0000000000000000	DAC1DATL	C8C	0000000000000000	SLP3CONH	CB2	----000-----
ADCBUF11	C22	0000000000000000	DAC1DATH	C8E	0000000000000000	SLP3DAT	CB4	0000000000000000
ADCBUF12	C24	0000000000000000	SLP1CONL	C90	0000000000000000	VREGCON	CFC	0-----000000
ADCBUF13	C26	0000000000000000	SLP1CONH	C92	----000-----			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。



# dsPIC33CK64MP105系列

表4-11: SFR块 D00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
PPS			RPINR21	D2E	0000000000000000	RPOR4	D88	--000000--000000
RPCON	D00	----0-----	RPINR22	D30	0000000000000000	RPOR5	D8A	--000000--000000
RPINR0	D04	00000000-----	RPINR23	D32	-----00000000	RPOR6	D8C	--000000--000000
RPINR1	D06	0000000000000000	RPINR27	D3A	0000000000000000	RPOR7	D8E	--000000--000000
RPINR2	D08	00000000-----	RPINR29	D3E	0000000000000000	RPOR8	D90	--000000--000000
RPINR3	D0A	0000000000000000	RPINR30	D40	-----00000000	RPOR9	D92	--000000--000000
RPINR4	D0C	0000000000000000	RPINR37	D4E	0000000000000000	RPOR10	D94	--000000--000000
RPINR5	D0E	0000000000000000	RPINR38	D50	-----00000000	RPOR11	D96	--000000--000000
RPINR6	D10	0000000000000000	RPINR42	D58	0000000000000000	RPOR12	D98	--000000--000000
RPINR7	D12	0000000000000000	RPINR43	D5A	0000000000000000	RPOR13	D9A	--000000--000000
RPINR11	D1A	0000000000000000	RPINR44	D5C	0000000000000000	RPOR14	D9C	--000000--000000
RPINR12	D1C	0000000000000000	RPINR45	D5E	0000000000000000	RPOR16	DA0	--000000-----
RPINR13	D1E	0000000000000000	RPINR46	D60	0000000000000000	RPOR20	DA8	-----000000
RPINR14	D20	0000000000000000	RPINR47	D62	0000000000000000	RPOR21	DAA	-----000000
RPINR15	D22	0000000000000000	RPINR48	D64	0000000000000000	RPOR22	DAC	--000000-----
RPINR16	D24	0000000000000000	RPINR49	D66	0000000000000000	RPOR24	DB0	--000000--000000
RPINR17	D26	0000000000000000	RPOR0	D80	--000000--000000	RPOR25	DB2	--000000--000000
RPINR18	D28	0000000000000000	RPOR1	D82	--000000--000000	RPOR26	DB4	--000000--000000
RPINR19	D2A	0000000000000000	RPOR2	D84	--000000--000000			
RPINR20	D2C	0000000000000000	RPOR3	D86	--000000--000000			

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

表4-12: SFR块 E00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
I/O 端口			ODCB	E24	0000000000000000	CNSTATC	E4A	0000000000000000
ANSELA	E00	-----11111	CNPUB	E26	0000000000000000	CNEN1C	E4C	0000000000000000
TRISA	E02	-----11111	CNPDB	E28	0000000000000000	CNFC	E4E	0000000000000000
PORTA	E04	-----xxxxx	CNCONB	E2A	----0-----	ANSELD	E54	--1-11-----
LATA	E06	-----xxxxx	CNEN0B	E2C	0000000000000000	TRISD	E56	1111111111111111
ODCA	E08	-----00000	CNSTATB	E2E	0000000000000000	PORTD	E58	xxxxxxxxxxxxxxxx
CNPUA	E0A	-----00000	CNEN1B	E30	0000000000000000	LATD	E5A	xxxxxxxxxxxxxxxx
CNPDA	E0C	-----00000	CNFB	E32	0000000000000000	ODCD	E5C	0000000000000000
CNCONA	E0E	----0-----	ANSELC	E38	-----11--1111	CNPUD	E5E	0000000000000000
CNEN0A	E10	-----00000	TRISC	E3A	1111111111111111	CNPDD	E60	0000000000000000
CNSTATA	E12	-----00000	PORTC	E3C	xxxxxxxxxxxxxxxx	CNCOND	E62	----0-----
CNEN1A	E14	-----00000	LATC	E3E	xxxxxxxxxxxxxxxx	CNEN0D	E64	0000000000000000
CNFA	E16	-----00000	ODCC	E40	0000000000000000	CNSTATD	E66	0000000000000000
ANSELB	E1C	-----111--11111	CNPUC	E42	0000000000000000	CNEN1D	E68	0000000000000000
TRISB	E1E	1111111111111111	CNPDC	E44	0000000000000000	CNFD	E6A	0000000000000000
PORTB	E20	xxxxxxxxxxxxxxxx	CNCONC	E46	----0-----	存储器BIST		
LATB	E22	xxxxxxxxxxxxxxxx	CNEN0C	E48	0000000000000000	MBISTCON	EFC	-----00--0---1

图注: x = 未知或不确定的值; “-” = 未实现位。地址值用十六进制表示。复位值用二进制表示。

# dsPIC33CK64MP105系列

表4-13: SFR块 F00h

寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值	寄存器	地址	所有复位时的值
<b>UART3</b>			U3INT	F24	-----00---0--	PMD3	FA8	-----00-0-000-
U3MODE	F00	--000-0000000000	<b>复位和振荡器</b>			PMD4	FAA	-----0---
U3MODEH	F02	00---0000000000	RCON	F80	xx---x01x0xxxxx	PMD6	FAE	---0000-----
U3STA	F04	0000000010000000	OSCCON	F84	0000-yyy0-0-0--0	PMD7	FB0	-----000---0---
U3STAH	F06	0000-00000101110	CLKDIV	F86	00110000--000001	PMD8	FB2	--000-0-0-00000-
U3BRG	F08	0000000000000000	PLLFBD	F88	----000010010110	<b>WDT</b>		
U3BRGH	F0A	-----0000	PLLDIV	F8A	-----00-011-001	WDTCONL	FB4	---000000000000
U3RXREG	F0C	-----xxxxxxxx	OSCTUN	F8C	-----000000	WDTCONH	FB6	0000000000000000
U3TXREG	F10	-----xxxxxxxx	ACLKCON1	F8E	00-----0--000001	<b>参考时钟输出</b>		
U3P1	F14	-----00000000	APLLFBD1	F90	----000010010110	REFOCONL	FB8	--000-00---0000
U3P2	F16	-----00000000	APLLDIV1	F92	-----00-011-001	REFOCONH	FBA	0000000000000000
U3P3	F18	0000000000000000	CANCLKCON	F9A	---xxxx-xxxxxxx	REFOTRIMH	FBE	000000000-----
U3P3H	F1A	-----00000000	DCOTUN	F9C	--000000--000000	<b>编程器/调试器</b>		
U3TXCHK	F1C	-----00000000	DCOCON	F9E	--0-xxxx-----	VISI	FCC	xxxxxxxxxxxxxxxxxx
U3RXCHK	F1E	-----00000000	<b>PMD</b>			APPO	FD2	xxxxxxxxxxxxxxxxxx
U3SCCON	F20	-----00000-	PMD1	FA4	----000-00000-00	APPI	FD4	xxxxxxxxxxxxxxxxxx
U3SCINT	F22	--00-000--00-000	PMD2	FA6	-----00000000	APPS	FD6	-----xxxxx

图注: x = 未知或不确定的值; “-” = 未实现位; y = 由配置位设置的值。地址值用十六进制表示。  
复位值用二进制表示。

4.4.1 分页存储器方案

dsPIC33CK64MP105架构通过分页方案扩展可用数据空间，从而对于执行前修改或后修改有效地址（EA）支持使用MOV指令以线性方式访问可用数据空间。基本数据空间的高半地址部分与数据空间读页（DSRPAG）寄存器配合使用，构成程序空间可视性（PSV）地址。

数据空间读页（DSRPAG）寄存器位于SFR空间中。  
图4-7给出了PSV地址的构成。当DSRPAG[9] = 1且基址位EA[15] = 1时，DSRPAG[8:0]位与EA[14:0]相连构成24位PSV读地址。

通过分页存储器方案可以访问PSV存储区中的多个32 KB窗口。数据空间读页（DSRPAG）寄存器与数据空间高半地址部分组合使用时，最多可以提供8 MB的PSV地址空间。分页数据存储空间如图4-8所示。

DSRPAG为0x200或更大时，可以访问程序空间（PS）。仅支持使用DSRPAG对PS执行读操作。

图4-7： 程序空间可视性（PSV）的读地址生成方式

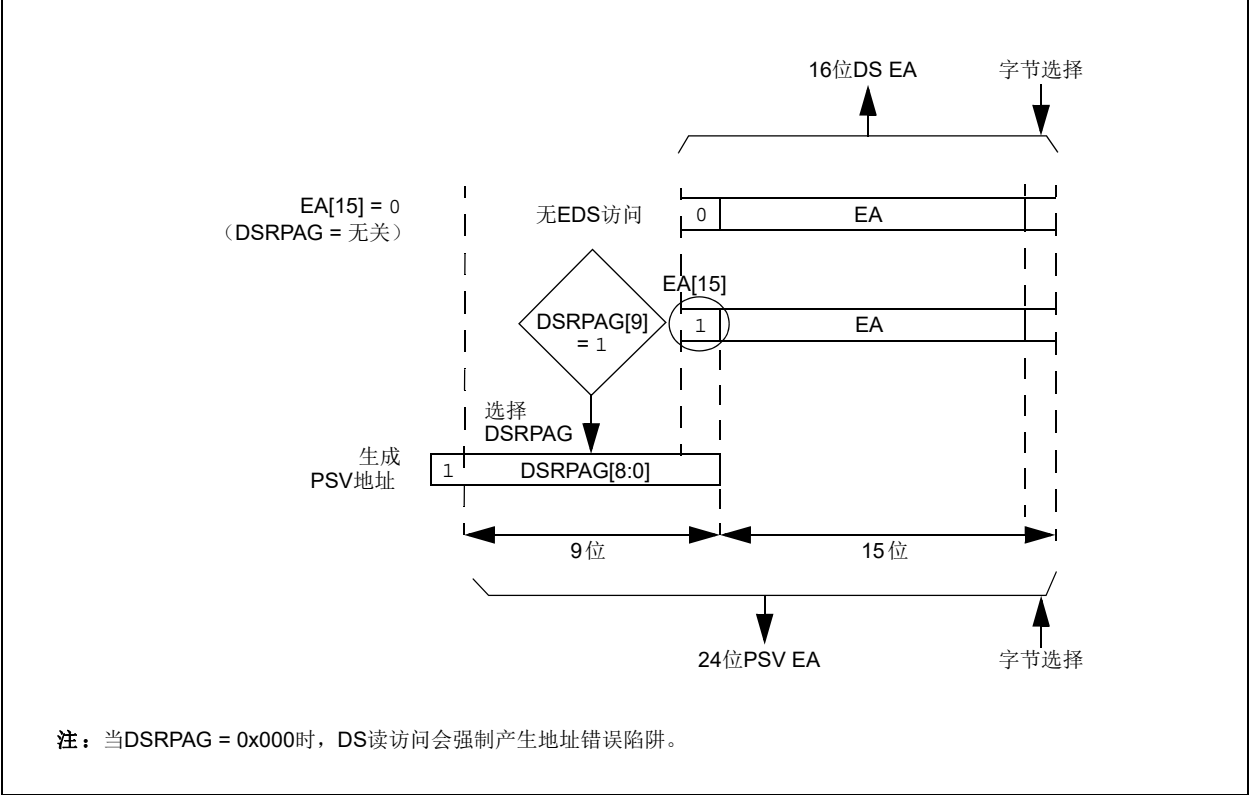
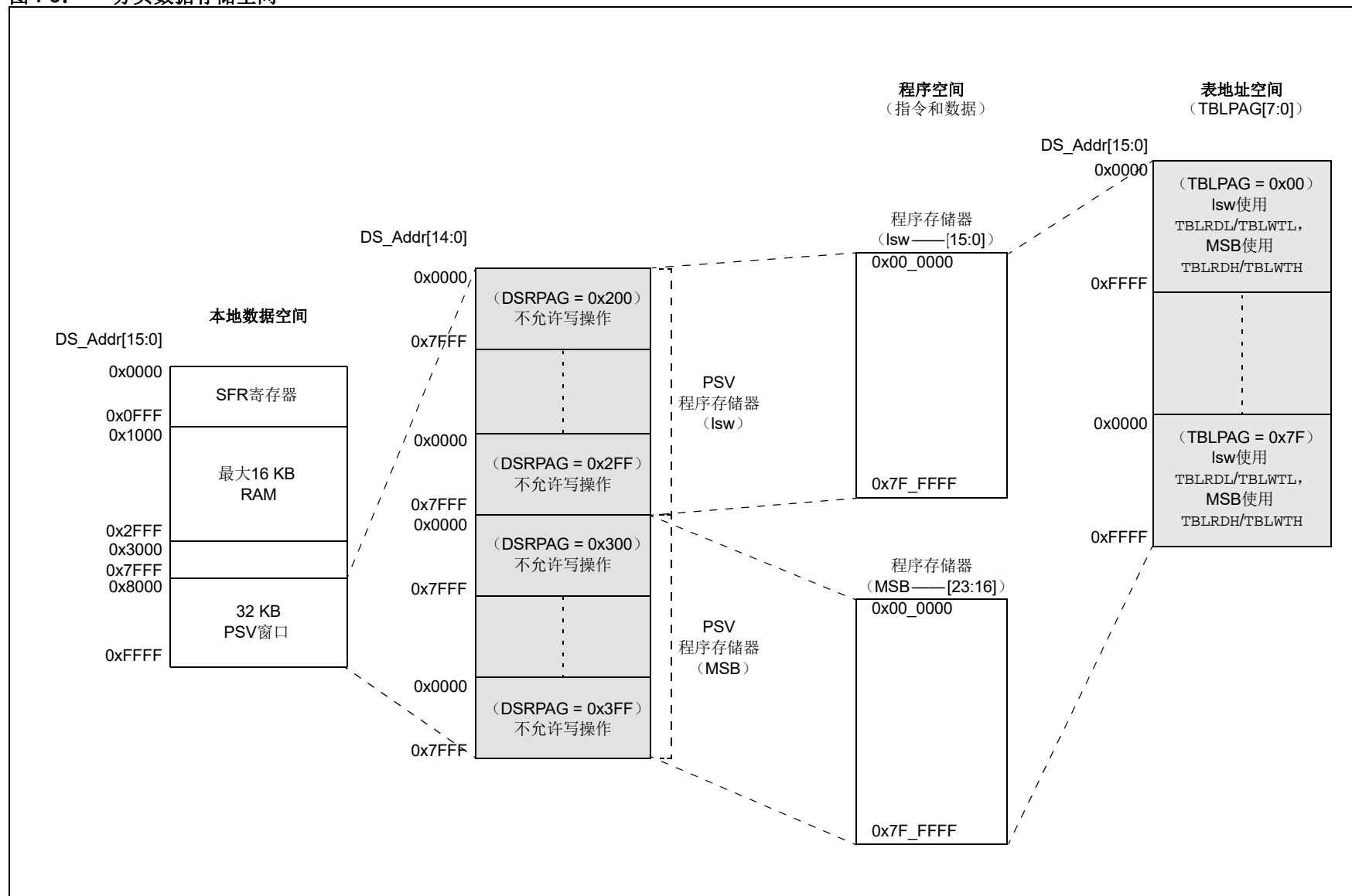


图4-8: 分页数据存储空间



发生PSV页上溢或下溢时，EA[15]会由于寄存器间接EA计算而清零。在以下情况下，PSV页中的EA会在页边界处发生上溢或下溢：

- 初始地址（在修改之前）寻址PSV页
- EA计算使用执行前修改或执行后修改的寄存器间接寻址模式；但是，这不包括寄存器偏移量寻址

通常在检测到上溢时，DSRPAG寄存器会递增，EA[15]位会置1，以使基址保持在PSV窗口内。当检测到下溢时，DSRPAG寄存器会递减，EA[15]位会置1，以使基

址保持在PSV窗口内。这可以产生线性的PSV地址空间，但仅限使用寄存器间接寻址模式时。

在进入和退出Page 0和PSV空间的边界时，会出现上述操作的例外情形。表4-14列出了不同边界处上溢和下溢情形的影响。

在以下情形中，发生上溢或下溢时，EA[15]位会置1，DSRPAG不会发生修改；因此，EA会折回到当前页的起始位置：

- 带寄存器偏移量的寄存器间接寻址
- 模寻址
- 位反转寻址

表4-14: PAGE 0和PSV空间边界处的上溢和下溢情形<sup>(2,3,4)</sup>

O/U, R/W	操作	之前			之后		
		DSRPAG	DS EA[15]	页说明	DSRPAG	DS EA[15]	页说明
O, 读	[++Wn] 或 [Wn++]	DSRPAG = 0x2FF	1	PSV: 最后一个 lsw 页	DSRPAG = 0x300	1	PSV: 第一个 MSB 页
O, 读		DSRPAG = 0x3FF	1	PSV: 最后一个 MSB 页	DSRPAG = 0x3FF	0	见注1
U, 读	[--Wn] 或 [Wn--]	DSRPAG = 0x001	1	PSV 页	DSRPAG = 0x001	0	见注1
U, 读		DSRPAG = 0x200	1	PSV: 第一个 lsw 页	DSRPAG = 0x200	0	见注1
U, 读		DSRPAG = 0x300	1	PSV: 第一个 MSB 页	DSRPAG = 0x2FF	1	PSV: 最后一个 lsw 页

图注: O = 上溢, U = 下溢, R = 读, W = 写

注 1: 寄存器间接寻址现在寻址基本数据空间（0x0000-0x8000）中的某个存储单元。

- 2: DSRPAG = 0x000时的EDS访问会产生地址错误陷阱。
- 3: 仅支持使用DSRPAG对PS执行读操作。
- 4: 偏移量很大时，不支持伪线性寻址。

4.4.1.1 扩展X数据空间

无论数据空间读页寄存器的内容如何，总是可以访问介于0x0000和0x7FFF之间的基本地址空间的低地址部分。它可以通过寄存器间接寻址指令进行间接寻址。它可以视为位于默认的EDS Page 0中（即，EDS地址范围0x000000至0x007FFF，对于该地址范围，基址位EA[15] = 0）。但是，Page 0不能通过基本数据空间高32 KB（0x8000至0xFFFF）结合DSRPAG = 0x000来进行访问。因此在复位时，DSRPAG 会被初始化为0x001。

- 注 1:** DSRPAG不应用于访问Page 0。DSRPAG设置为0x000时进行EDS访问会产生地址错误陷阱。

**2:** 用软件清零DSRPAG将不起作用。

其余PSV页只能使用DSRPAG寄存器结合基本地址空间高32 KB（0x8000至0xFFFF）进行访问，此时基址位EA[15] = 1。

4.4.1.2 软件堆栈

W15寄存器用作专用软件堆栈指针（SSP），由异常处理、子程序调用和返回自动修改；但是，W15可以被任何指令以与所有其他W寄存器相同的方式引用。这样就简化了堆栈指针的读、写和操作（例如，创建堆栈帧）。

- 注:** 为了防止不对齐的堆栈访问，W15[0]被硬件固定设置为0。

所有复位均将W15初始化为0x1000。该地址确保在所有dsPIC33CK64MP105器件中SSP将指向有效的RAM，并允许不可屏蔽陷阱异常使用堆栈。这些可能在SSP被用户软件初始化之前发生。在初始化期间，可以将SSP再编程以指向数据空间内的任何存储单元。

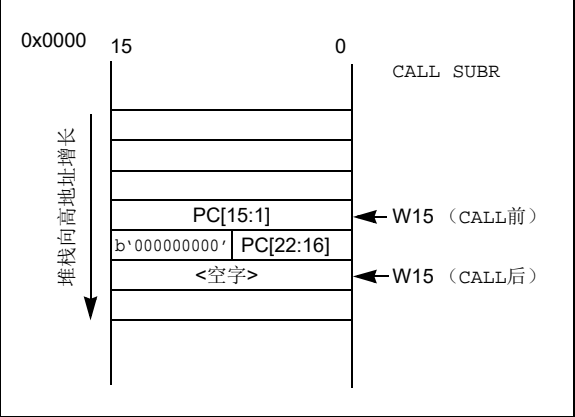
软件堆栈指针总是指向第一个可用的空字并从低地址到高地址填充软件堆栈。图4-9显示了它如何在弹出堆栈（读）时预递减，以及在压入堆栈（写）时后递增。

当PC压入堆栈时，PC[15:0]被压入第一个可用的堆栈字，然后PC[22:16]被压入第二个可用的堆栈单元。对于任何CALL指令执行期间的PC压栈，压栈前PC的MSB被零扩展，如图4-9所示。异常处理期间，PC的MSB与CPU状态寄存器SR的低8位组合在一起。这样就能在中断处理期间自动保存SRL的内容。

- 注 1:** 为了维持系统堆栈指针（W15）一致性，W15不会受（EDS）分页影响，因此被限制在地址范围0x0000至0xFFFF。当W14用作堆栈帧指针（SFA = 1）时，这同样适用于W14。

**2:** 由于堆栈可置于X和Y空间中并能访问这两个空间，因此在使用堆栈、特别是涉及到C语言开发环境中的局部自动变量时必须小心。

图4-9: CALL堆栈帧



4.4.2 指令寻址模式

表4-15给出了基本的寻址模式，这些寻址模式经过优化可以支持各指令的具体功能。MAC类指令中提供的寻址模式与其他指令类型中的寻址模式有所不同。

4.4.2.1 文件寄存器指令

大多数文件寄存器指令使用一个13位地址字段（f）来直接寻址数据存储器中的前8192字节（Near数据空间）。大多数文件寄存器指令使用工作寄存器W0，W0在这些指令中表示为WREG。目标寄存器通常是同一个文件寄存器或WREG（MUL指令除外，它将结果写入寄存器或寄存器对）。使用MOV指令能够获得更大的灵活性，并可以访问整个数据空间。

4.4.2.2 MCU指令

三操作数MCU指令的形式是：

操作数3 = 操作数1<功能>操作数2

其中，操作数1始终是工作寄存器（即，寻址模式只能是寄存器直接寻址），称为Wb。操作数2可以是一个W寄存器，取自数据存储器，或为一个5位立即数。结果可以保存在W寄存器或数据存储单元中。MCU指令支持以下寻址模式：

- 寄存器直接寻址
- 寄存器间接寻址
- 执行后修改的寄存器间接寻址
- 执行前修改的寄存器间接寻址
- 5位或10位立即数寻址

注：并非所有指令都支持上述所有的寻址模式。各条指令可能支持这些寻址模式中的部分模式。

表4-15： 支持的基本寻址模式

寻址模式	说明
文件寄存器直接寻址	明确指定文件寄存器的地址。
寄存器直接寻址	直接访问寄存器的内容。
寄存器间接寻址	Wn的内容形成有效地址（EA）。
执行后修改的寄存器间接寻址	Wn的内容形成EA。然后用一个常量值来修改Wn（递增或递减）。
执行前修改的寄存器间接寻址	先用一个有符号常量值修改Wn（递增或递减），再由此时的Wn内容形成EA。
带寄存器偏移量的寄存器间接寻址（寄存器变址寻址）	Wn和Wb的和形成EA。
带立即数偏移量的寄存器间接寻址	Wn和立即数的和形成EA。

### 4.4.2.3 传送指令和累加器指令

与其他指令相比，传送指令和DSP累加器类指令提供了更为灵活的寻址。除了大多数MCU指令支持的寻址模式以外，传送指令和累加器指令还支持带寄存器偏移量的寄存器间接寻址模式，这也称为寄存器变址寻址模式。

**注：** 对于MOV指令，指令中指定的寻址模式对于源和目标EA，可以是不同的。然而，4位Wb（寄存器偏移量）字段为源和目标所共用（但通常只由其中之一使用）。

概括地说，传送指令和累加器指令支持以下寻址模式：

- 寄存器直接寻址
- 寄存器间接寻址
- 执行后修改的寄存器间接寻址
- 执行前修改的寄存器间接寻址
- 带寄存器偏移量的寄存器间接寻址（变址寻址）
- 带立即数偏移量的寄存器间接寻址
- 8位立即数寻址
- 16位立即数寻址

**注：** 并非所有指令都支持上述所有的寻址模式。各条指令可能支持这些寻址模式中的部分模式。

### 4.4.2.4 MAC指令

双源操作数DSP指令（CLR、ED、EDAC、MAC、MPY、MPY.N、MOVSAC和MSC），也称为MAC指令，它们使用一组简化的寻址模式，允许用户应用程序通过寄存器间接寻址表有效地对数据指针进行操作。

双源操作数预取寄存器必须是集合{W8, W9, W10, W11}的成员。对于数据读取操作，W8和W9始终用于X RAGU，而W10和W11始终用于Y AGU。从而，产生的有效地址（无论是在修改之前还是之后），对于W8和W9必须是X数据空间中的有效地址，对于W10和W11则必须是Y数据空间中的有效地址。

**注：** 带寄存器偏移量的寄存器间接寻址模式仅可用于W9（在X空间中）和W11（在Y空间中）。

概括地说，MAC类指令支持以下寻址模式：

- 寄存器间接寻址
- 执行后修改（修改量为2）的寄存器间接寻址
- 执行后修改（修改量为4）的寄存器间接寻址
- 执行后修改（修改量为6）的寄存器间接寻址
- 带寄存器偏移量的寄存器间接寻址（变址寻址）

### 4.4.2.5 其他指令

除了上述的各种寻址模式之外，一些指令使用各种长度的立即数常量。例如，BRA（跳转）指令使用16位有符号立即数来直接指定跳转的目标，而DISI指令则使用一个14位无符号立即数字段。在一些指令中，例如ULNK，操作数或结果的存储位置已经暗含在操作码中。某些操作，例如NOP，没有任何操作数。



4.4.3 模寻址

模寻址模式是一种使用硬件来自动支持循环数据缓冲区的方法。目的是在执行紧密循环代码时（这在许多DSP算法中很典型），不需要用软件来执行数据地址边界检查。

可以在数据空间或程序空间中进行模寻址（因为这两种空间的数据指针机制本质上是相同的）。X数据空间（也提供指向程序空间的指针）和Y数据空间中都可支持一个循环缓冲区。模寻址可以对任何W寄存器指针进行操作。然而，最好不要将W14或W15用于模寻址，因为这两个寄存器分别用作堆栈帧指针和堆栈指针。

一般来说，任何特定的循环缓冲区只能配置为单向操作，因为根据缓冲区的方向，对缓冲区起始地址（对于递增缓冲区）或结束地址（对于递减缓冲区）有一定限制。

使用限制的惟一例外是那些长度为2的幂的缓冲区。由于这些缓冲区满足起始和结束地址标准，它们可以双向操作（即，在低地址边界和高地址边界上都将进行地址边界检查）。

4.4.3.1 起始地址和结束地址

模寻址机制要求指定起始地址和结束地址，并将它们装入16位模缓冲区地址寄存器：XMODSRT、XMODEND、YMODSRT和YMODEND（见表4-2）。

**注：** Y空间模寻址的EA计算使用字长度的数据（每个EA的LSb始终清零）。

循环缓冲区的长度不直接指定。它由相应的起始地址和结束地址之差决定。循环缓冲区的最大可能长度为32K字（64 KB）。

4.4.3.2 W地址寄存器选择

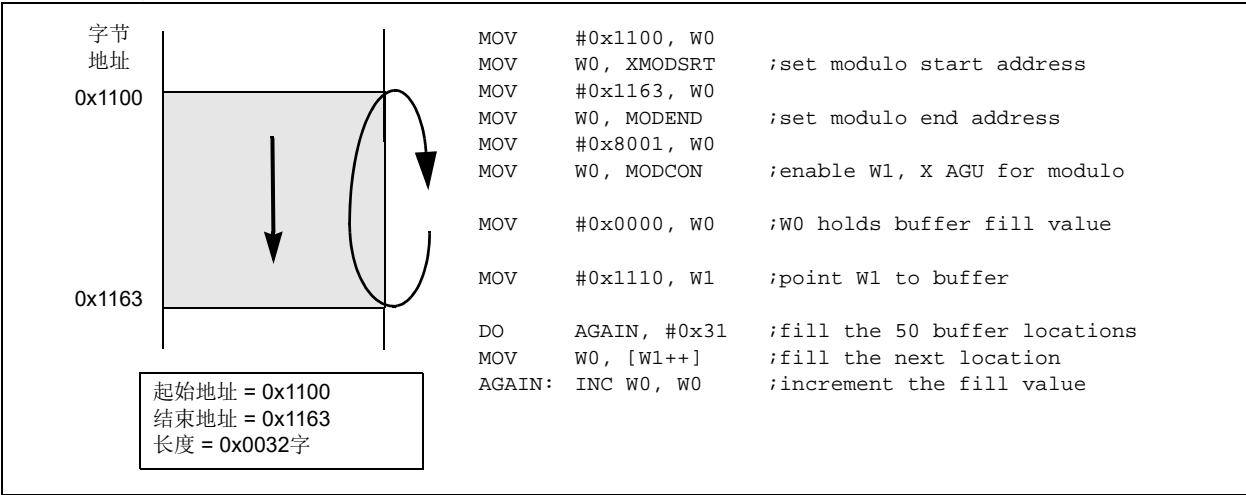
模寻址和位反转寻址控制寄存器MODCON[15:0]包含使能标志以及指定W地址寄存器的W寄存器字段。XWM和YWM字段选择对哪些寄存器进行模寻址：

- 如果XWM = 1111，则禁止X RAGU和X WAGU模寻址
- 如果YWM = 1111，则禁止Y AGU模寻址

要进行模寻址的X地址空间指针W（XWM）寄存器存储在MODCON[3:0]中（见表4-2）。当XWM设置为除1111之外的任何值且XMODEN位（MODCON[15]）置1时，X数据空间的模寻址被使能。

要进行模寻址的Y地址空间指针W（YWM）寄存器存储在MODCON[7:4]中。当YWM设置为除1111之外的任何值且YMODEN位（MODCON[14]）置1时，Y数据空间的模寻址被使能。

图4-10： 模寻址操作示例



4.4.3.3 模寻址的应用

模寻址可以应用于与任何W寄存器相关的有效地址(EA)计算中。地址边界检查功能检查地址是否等于：

- 上边界地址 (对于递增缓冲区)
- 下边界地址 (对于递减缓冲区)

重要的是要意识到，地址边界检查功能不仅会检查地址是否正好在地址边界上，而且会检查地址是否大于上边界地址 (对于递增缓冲区)、或小于下边界地址 (对于递减缓冲区)。因此，地址变化可能会越过边界，但仍然可以正确调整。

**注：** 只有在使用执行前修改或执行后修改寻址模式来计算有效地址时，模修正有效地址才被写回寄存器。如果使用了地址偏移量 (例如，[W7 + W2])，会进行模地址修正，但寄存器的内容保持不变。

4.4.4 位反转寻址

位反转寻址模式用来简化基2 FFT算法的数据重新排序。它为X AGU所支持，仅限于数据写入。

地址修改量可以是常量或寄存器的内容，可视为将其位顺序反转。源地址和目标地址仍然是正常的顺序。于是，惟一需要反转的操作数就是地址修改量。

4.4.4.1 位反转寻址的实现

在以下任一条件下，均可使能位反转寻址模式：

- MODCON寄存器中BWMx位 (W寄存器选择) 的值是除1111以外的任何值 (不能使用位反转寻址访问堆栈)
- XBREV寄存器中的BREN位置1
- 使用的寻址模式是预递增或后递增的寄存器间接寻址模式

如果位反转缓冲区的长度为M = 2<sup>N</sup>字节，则数据缓冲区起始地址的最后“N”位必须为零。

XB[14:0]是位反转地址修改量或“轴心点”(pivot point)，通常是一个常量。对于FFT计算，其值等于FFT数据缓冲区长度的一半。

**注：** 所有位反转EA的计算都使用字长度数据 (每个EA的LSb始终清零)。为了产生兼容 (字节) 地址，要相应地调整XB的值。

使能位反转寻址时，仅对预递增或后递增的寄存器间接寻址、且仅对字长度数据写，才会进行位反转寻址。对于任何其他寻址模式或对于字节长度数据，不会进行位反转寻址，而是生成正常的地址。在进行位反转寻址时，W地址指针将始终加上地址修改量 (XB)，与寄存器间接寻址模式相关的偏移量将被忽略。此外，由于要求是字长度数据，EA的LSb被忽略 (且始终清零)。

**注：** 可以使用同一个W寄存器同时使能模寻址和位反转寻址，但当同时使能时，对于数据写操作位反转寻址总是优先。

如果已通过将BREN位 (XBREV[15]) 置1使能了位反转寻址，那么，在写XBREV寄存器之后，不应立即进行要使用被指定为位反转指针的W寄存器的间接读操作。

图4-11： 位反转寻址示例

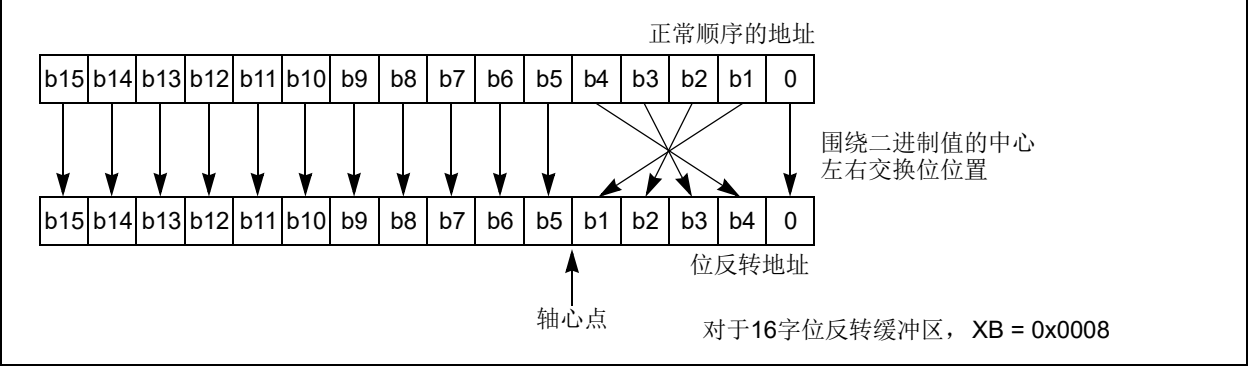


表4-16： 位反转寻址序列（16项）

正常地址					位反转地址				
A3	A2	A1	A0	十进制	A3	A2	A1	A0	十进制
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	8
0	0	1	0	2	0	1	0	0	4
0	0	1	1	3	1	1	0	0	12
0	1	0	0	4	0	0	1	0	2
0	1	0	1	5	1	0	1	0	10
0	1	1	0	6	0	1	1	0	6
0	1	1	1	7	1	1	1	0	14
1	0	0	0	8	0	0	0	1	1
1	0	0	1	9	1	0	0	1	9
1	0	1	0	10	0	1	0	1	5
1	0	1	1	11	1	1	0	1	13
1	1	0	0	12	0	0	1	1	3
1	1	0	1	13	1	0	1	1	11
1	1	1	0	14	0	1	1	1	7
1	1	1	1	15	1	1	1	1	15

# dsPIC33CK64MP105系列

## 4.4.5 程序存储空间与数据存储空间的接口

dsPIC33CK64MP105系列架构采用24位宽的程序空间（PS）和16位宽的数据空间（DS）。该架构也是一种改进型哈佛结构，这意味着数据也能存放在程序空间内。要成功使用这些数据，在访问数据时必须确保这两种存储空间中的信息是对齐的。

除了正常执行外，dsPIC33CK64MP105系列器件的架构还提供了两种可在操作过程中访问程序空间的方法：

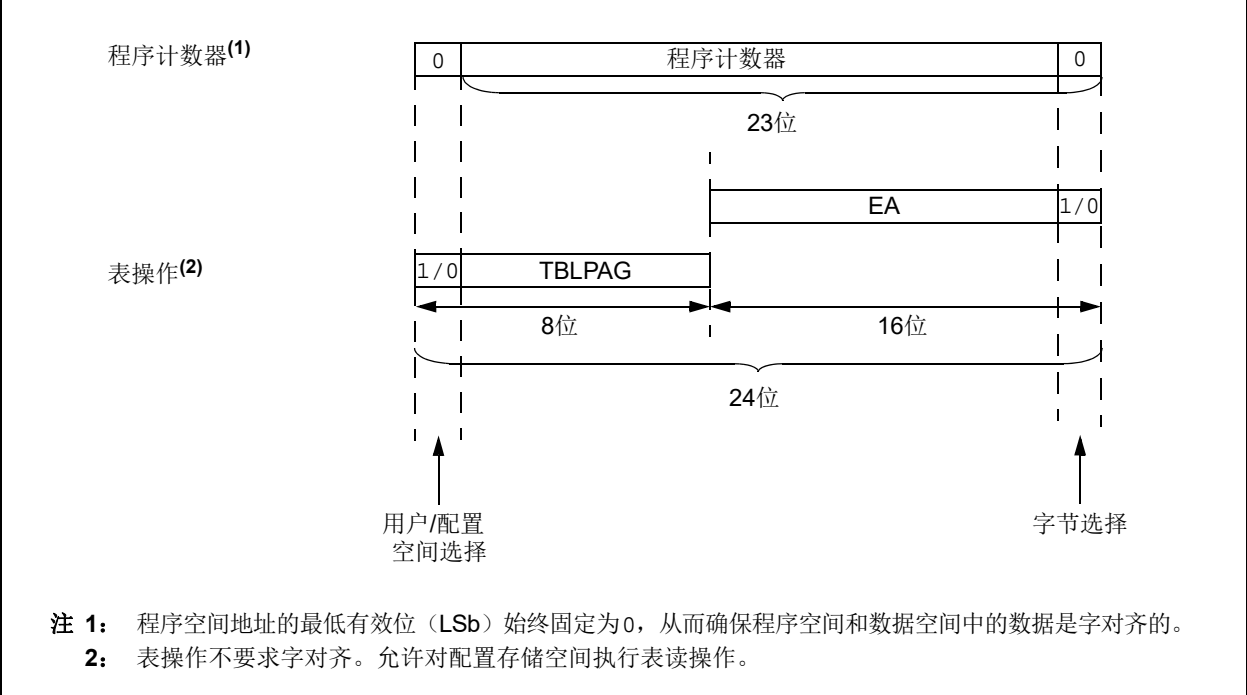
- 使用表指令访问程序空间中任意位置的各个字节或字
- 将程序空间的一部分重映射到数据空间（程序空间可视性）

表指令允许应用程序读取程序存储器的小块区域。这一功能对于访问需要周期性更新的数据表来说非常理想。也可通过该功能访问一个程序字的所有字节。重映射方法允许应用程序访问一大块数据，但只限于读操作，它非常适合于在一个大的静态数据表中进行查找。通过这一方法应用程序只能访问程序字的低位字。

表4-17： 程序空间地址构成

访问类型	访问空间	程序空间地址				
		[23]	[22:16]	[15]	[14:1]	[0]
指令访问（代码执行）	用户	0	PC[22:1]			0
		0xxx xxxx xxxx xxxx xxxx xxx0				
TBLRD（字节/字读）	用户	TBLPAG[7:0]		数据EA[15:0]		
		0xxx xxxx		xxxx xxxx xxxx xxxx		
	配置	TBLPAG[7:0]		数据EA[15:0]		
		1xxx xxxx		xxxx xxxx xxxx xxxx		

图4-12： 访问程序空间内数据的地址生成方式



4.4.5.1 使用表指令访问程序存储器中的数据

TBLRD<sub>L</sub>指令提供了直接读程序空间内任何地址的低位字的方法，无需通过数据空间。TBLRD<sub>H</sub>指令是可以将一个程序空间字的最高8位作为数据读取的惟一方法。

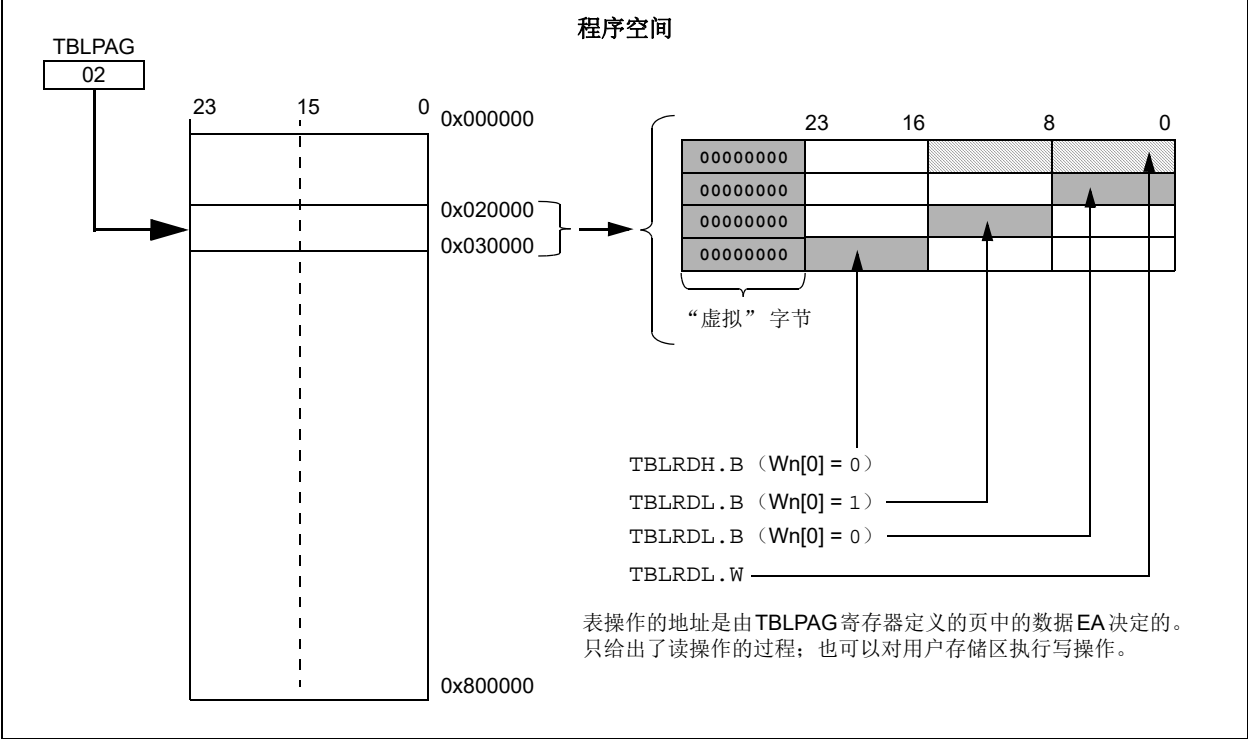
这使得程序存储器地址能够直接映射到数据空间地址。于是，程序存储器可以看作是两个16位宽字的地址空间，它们并排放置，具有相同的地址范围。TBLRD<sub>L</sub>访问存有最低有效数据字的空间，而TBLRD<sub>H</sub>则访问存有最高数据字节的空间。

提供了两条表指令来对程序空间执行字节或字（16位）长度的读操作。读和写都可以采用字节或字操作的形式。

- TBLRD<sub>L</sub>（表读低位字）：
  - 在字模式下，该指令将程序空间地址中的低位字（P[15:0]）映射到数据地址（D[15:0]）中
  - 在字节模式下，低位程序字的高字节或低字节被映射到数据地址的低字节中。当字节选择位为1时映射高字节；当字节选择位为0时映射低字节。

- TBLRD<sub>H</sub>（表读高位字）：
  - 在字模式下，该指令将程序地址中的整个高位字（P[23:16]）映射到数据地址中。“虚拟”字节（D[15:8]）始终为全0。
  - 在字节模式下，该指令将程序字的高字节或低字节映射到数据地址的D[7:0]中，就如同TBLRD<sub>L</sub>指令。当选择最高“虚拟”字节（字节选择位 = 1）时，数据将始终为全0。

图4-13： 使用表指令访问程序存储器



注:

5.0 闪存程序存储器

**注 1：** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“闪存编程”([www.microchip.com/DS70000609](http://www.microchip.com/DS70000609))。

dsPIC33CK64MP105系列器件包含用于存储和执行应用程序代码的内部闪存程序存储器。在整个VDD范围内，正常操作期间，该存储器都是可读写、可擦除的。

可采用三种方式对闪存进行编程：

- 在线串行编程（ICSP）功能
- 增强型在线串行编程（增强型ICSP）
- 运行时自编程（Run-Time Self-Programming, RTSP）

ICSP允许在最终的应用电路中对dsPIC33CK64MP105系列器件进行串行编程。只需要使用5根线就可以完成编程，它们分别是编程时钟线（PGCx）、编程数据线（PGDx）、电源线（VDD）、地线（VSS）和主复位线（MCLR）。这允许用户在生产电路板时使用未编程器件，而仅在产品交付之前才对器件进行编程，从而可以使用最新版本的固件或者定制固件进行编程。

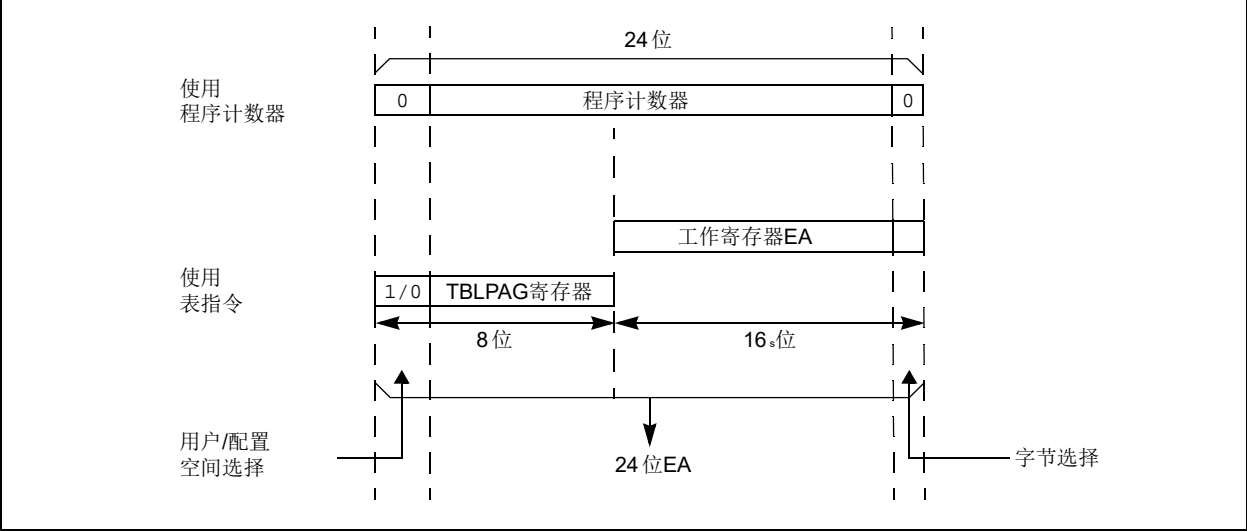
增强型在线串行编程使用称为编程执行程序的片上自举程序来管理编程过程。编程执行程序可以使用SPI数据帧格式擦除、编程和校验程序存储器。有关增强型ICSP的更多信息，请参见器件编程规范。

使用TBLRD（表读）和TBLWT（表写）指令来实现RTSP。使用RTSP，用户应用程序可以通过两个程序存储字或128条指令（256个可寻址字节）的块（“行”）写入程序存储器数据。RTSP擦除程序存储器时可以一次擦除1024条指令（2048个可寻址字节）的块或“页”。

5.1 表指令和闪存编程

闪存的所有编程都是通过表读和表写指令完成的，与使用的编程方法无关。这些指令允许器件在正常工作模式下从数据存储器直接读写程序存储空间。程序存储器中24位目标地址由TBLPAG寄存器的bit[7:0]和表指令中指定W寄存器中的有效地址（EA）组成，如图5-1所示。TBLRDL和TBLWTL指令用于读或写程序存储器的bit[15:0]。TBLRDL和TBLWTL能以字或字节模式访问程序存储器。TBLRDH和TBLWTH指令用于读或写程序存储器的bit[23:16]。TBLRDH和TBLWTH同样能以字或字节模式访问程序存储器。

图5-1： 表寄存器的寻址



5.2 RTSP工作原理

dsPIC33CK64MP105 系列闪存程序存储器阵列是由 128 条指令或 384 字节的行组成的。RTSP 允许用户应用程序一次擦除由 8 行（1024 条指令）组成的一个存储器页，以及一次编程一行。也可以一次编程两条指令。

页擦除和单行写入块都是边沿对齐的，从程序存储器起始地址开始，分别到 3072 字节边界和 384 字节边界。

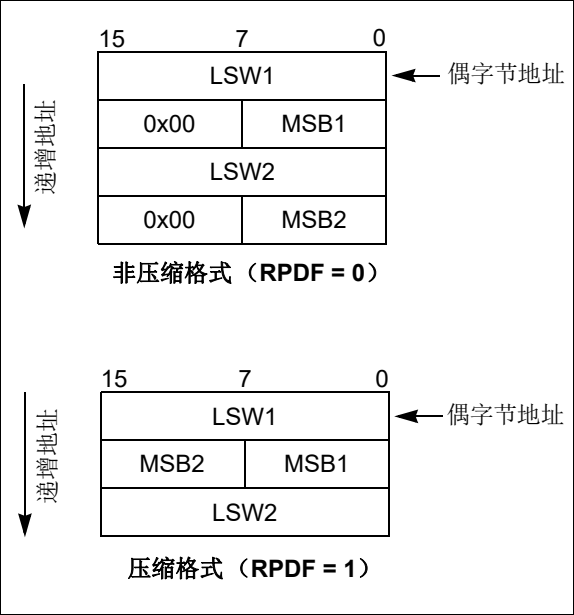
**第 31.0 节“电气特性”**中的表 31-18 列出了典型的擦除和编程时间。要写入闪存，需要先擦除包含用户希望更改存储单元的地址的页。

通过将 384 个字节装入数据存储器，然后将该行第一个字节的地址装入 NVMSRCADRL/H 寄存器对来执行行编程。写操作启动后，器件将自动装入写锁存器并递增 NVMSRCADRL/H 和 NVMA DR/U 寄存器，直到编程完所有字节。RPDF 位（NVMCON[9]）用于选择 RAM 中存储数据的格式为压缩还是非压缩。关于数据格式，请参见图 5-2。压缩数据会将第二个字的高字节用于第二条指令的 MSB，这样有助于减少所需 RAM 的量。

RTSP 字编程的基本过程是使用 TBLWTL 和 TBLWTH 指令将两条 24 位指令装入位于配置存储器空间的写锁存器。关于写锁存器的地址，请参见图 4-1 至图 4-3。通过解锁和设置 NVMCON 寄存器中的控制位来执行编程。

所有擦除和编程操作均可选择使用 NVM 中断来发出已成功完成操作的信号。

图 5-2: 非压缩/压缩格式



在 RTSP 模式下，对内部闪存进行编程或擦除需要执行完整的编程序列。处理器暂停（等待）直到编程操作完成。将 WR 位（NVMCON[15]）置 1 启动操作，当操作完成时 WR 位会自动清零。WR 位受到保护，可防止意外写入。要将该位置 1，必须将 0x55 和 0xAA 值按顺序写入 NVMKEY 寄存器。执行编程命令（WR 位 = 1）后，用户应用程序必须等待编程完成（WR 位 = 0）。编程序列启动后紧跟的两条指令应该为 NOP 指令。

**注：** MPLAB® XC16 提供了一个内置 C 语言函数，包括用于将 NVMCON 寄存器中的 WR 位置 1 的解锁序列：  
\_\_builtin\_write\_NVM()



## 5.3 闪存程序存储器控制寄存器

有 6 个 SFR 用于写入和擦除闪存程序存储器：NVMCON、NVMKEY、NVMADRU 和 NVMSRCADRL/H。

NVMCON 寄存器（[寄存器5-1](#)）用于选择要执行的操作（页擦除、字/行编程和非活动分区擦除）和启动编程或擦除周期。

NVMKEY（[寄存器5-4](#)）是一个只写寄存器，用于写保护。要启动编程或擦除序列，用户应用程序必须将 0x55 和 0xAA 连续写入 NVMKEY 寄存器。

有两个 NVM 地址寄存器：NVMADRU 和 NVMADR。这两个寄存器组合在一起时，构成要进行编程操作的选定字/行的 24 位有效地址（EA），或者要进行擦除操作的选定页的 24 位有效地址（EA）。NVMADRU 寄存器用于保存 EA 的高 8 位，而 NVMADR 寄存器用于保存 EA 的低 16 位。

对于行编程操作，要写入闪存程序存储器的数据将写入位于由 NVMSRCADRL/H 寄存器对定义的地址（行编程数据中第一个元素的位置）处的数据存储空间（RAM）。

# dsPIC33CK64MP105系列

## 寄存器5-1: NVMCON: 非易失性存储器 (NVM) 控制寄存器

R/SO-0 <sup>(1,6)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0	U-0	U-0	R/W-0	R/C-0
WR	WREN	WRERR	NVMSIDL <sup>(2)</sup>	—	—	RPDF	URERR
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>
—	—	—	—	NVMOP3 <sup>(3,4)</sup>	NVMOP2 <sup>(3,4)</sup>	NVMOP1 <sup>(3,4)</sup>	NVMOP0 <sup>(3,4)</sup>
bit 7							bit 0

图注:	C = 可清零位	SO = 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 15 **WR:** 写控制位<sup>(1,6)</sup>  
 1 = 启动闪存编程或擦除操作; 该操作是自定时的, 一旦操作完成, 该位即由硬件清零  
 0 = 编程或擦除操作完成, 并处于停止状态
- bit 14 **WREN:** 写使能位<sup>(1)</sup>  
 1 = 使能闪存编程/擦除操作  
 0 = 禁止闪存编程/擦除操作
- bit 13 **WRERR:** 写序列错误标志位<sup>(1)</sup>  
 1 = 试图执行不合法的编程或擦除序列, 或者发生终止 (试图将WR位置1时自动置1该位)  
 0 = 编程或擦除操作正常完成
- bit 12 **NVMSIDL:** NVM空闲模式停止控制位<sup>(2)</sup>  
 1 = 在空闲模式下闪存稳压器进入待机模式  
 0 = 在空闲模式下闪存稳压器继续工作
- bit 11-10 **未实现:** 读为0
- bit 9 **RPDF:** 行编程数据格式位  
 1 = 要存储在RAM中的行数据采用压缩格式  
 0 = 要存储在RAM中的行数据采用非压缩格式
- bit 8 **URERR:** 行编程数据不足错误位  
 1 = 指示行编程操作已终止  
 0 = 未检测到数据不足错误
- bit 7-4 **未实现:** 读为0

- 注 1: 这些位只能在POR时复位。
- 2: 如果该位置1, 功耗 (IDLE) 将达到最低, 并且当退出空闲模式后、闪存开始工作之前会有一段延时 (TVREG)。
- 3: NVMOP[3:0]的所有其他组合均未实现。
- 4: 在进行任意NVM操作时, PWRSAV指令的执行会被忽略。
- 5: 在执行该操作期间, 会对4字边界上两个相邻的字编程。
- 6: 写入该位需要解锁序列 (见第5.2节“RTSP工作原理”)。

## 寄存器5-1: NVMCON: 非易失性存储器 (NVM) 控制寄存器 (续)

bit 3-0 NVMOP[3:0]: NVM操作选择位 (1,3,4)

1111 = 保留  
1110 = 用户存储器批量擦除操作  
1101 = 保留  
1100 = 保留  
1011 = 保留  
1010 = 保留  
1001 = 保留  
1000 = 保留  
0111 = 保留  
0101 = 保留  
0100 = 保留  
0011 = 存储器页擦除操作  
0010 = 存储器行编程操作  
0001 = 存储器双字操作 (5)  
0000 = 保留

注 1: 这些位只能在POR时复位。

2: 如果该位置1, 功耗 (IDLE) 将达到最低, 并且当退出空闲模式后、闪存开始工作之前会有一段延时 (TVREG)。

3: NVMOP[3:0]的所有其他组合均未实现。

4: 在进行任意NVM操作时, PWRSAV指令的执行会被忽略。

5: 在执行该操作期间, 会对4字边界上两个相邻的字编程。

6: 写入该位需要解锁序列 (见第5.2节 “RTSP工作原理”)。

# dsPIC33CK64MP105系列

## 寄存器5-2: NVMADR: 非易失性存储器低位字地址寄存器

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADR[15:8]							
bit 15				bit 8			

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADR[7:0]							
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 15-0 **NVMADR[15:0]:** 非易失性存储器低位字写地址位

选择闪存程序存储器中要进行编程或擦除的存储单元的低16位。用户应用程序可以读写该寄存器。

## 寄存器5-3: NVMADRU: 非易失性存储器高位字地址寄存器

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

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
NVMADRU[23:16]							
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 15-8 **未实现:** 读为0

bit 7-0 **NVMADRU[23:16]:** 非易失性存储器最高字节写地址位

选择闪存程序存储器中要进行编程或擦除的存储单元的高8位。用户应用程序可以读写该寄存器。

寄存器5-4: **NVMKEY**: 非易失性存储器密钥寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-8      未实现: 读为0

bit 7-0      **NVMKEY[7:0]**: NVM密钥寄存器位 (只写)

# dsPIC33CK64MP105系列

## 寄存器5-5: NVMSRCADRL: NVM源数据地址低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NVMSRCADR[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
NVMSRCADR[7:0]							
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 15-0

**NVMSRCADR[15:0]:** NVM源数据地址位

待编程到闪存中的数据的RAM地址（将NVMOP[3:0]位设置为行编程时）。

## 寄存器5-6: NVMSRCADRH: NVM源数据地址高位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
NVMSRCADR[23:16]							
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 15-8

**未实现:** 读为0

bit 7-0

**NVMSRCADR[23:16]:** NVM源数据地址位

待编程到闪存中的数据的RAM地址（将NVMOP[3:0]位设置为行编程时）。

## 5.4 纠错码（ECC）

为了提高程序存储器的性能和耐用性，器件中还包括纠错码（**Error Correcting Code, ECC**）功能，作为闪存存储器控制器的一个组成部分。ECC可以确定程序数据中是否存在单个位错误（包括哪个位有错误），并且能够在没有用户干预的情况下自动纠正数据。无法禁止ECC。

当数据写入程序存储器时，ECC为每个双（24位）指令字生成一个7位汉明码奇偶校验值。数据存储在包含48个数据位和7个奇偶校验位的块中；奇偶校验数据不是存储器映射的，不可访问。当回读数据时，ECC会计算数据的奇偶校验值，并将其与先前存储的奇偶校验值进行比较。如果出现奇偶校验不匹配的情况，则有两种可能的结果：

- 发生了单个位错误，且已在回读时自动纠正。
- 发生了双位错误，且读取的数据没有改变。

可通过ECCSBEIF（IFS0[13]）位的状态来识别是否发生了单个位错误。当相应的中断允许位ECCSBEIE（IEC0[13]）置1时，可产生中断。ECCSTATL寄存器包含单个位错误的奇偶校验信息。SECOUT[7:0]位域包含预期的SEC奇偶校验计算值，SECIN[7:0]位包含闪存读操作的实际值。SECSYNDx位（ECCSTATH[7:0]）指示48位指令字对中单个位错误的位位置。不存在错误时，SECINx等于SECOUTx，SECSYNDx为零。

双位错误会导致通用硬陷阱。ECCDBE位（INTCON4[1]）将置1以识别硬陷阱源。如果没有为硬陷阱实现中断服务程序，则还会发生器件复位。ECCSTATH寄存器包含双位错误状态信息。DEDOUT位是预期的DED奇偶校验计算值，DEDIN是闪存读操作的实际值。不存在错误时，DEDIN等于DEDOUT。

### 5.4.1 ECC错误注入

为了测试故障处理情况，可能会产生EEC错误。读取和写入数据路径中都可能产生单个位错误和双位错误。读取路径错误注入会先读取闪存数据，然后在进入ECC逻辑之前对其进行修改。写入路径错误注入会在实际数据写入目标闪存之前对其进行修改，并会在后续闪存读取时导致EEC错误。以下步骤用于注入错误：

1. 将闪存目标地址装入ECCADDR寄存器。
2. 选择由FLT1PTRx（ECCCONH[7:0]）确定的第一个错误位。将目标位翻转，以产生错误。
3. 如果需要双位错误，请选择由FLT2PTRx（ECCCONH[15:8]）确定的第二个错误位，否则设置为全1。
4. 写入NVMKEY解锁序列（见第5.3节“闪存程序存储器控制寄存器”）。
5. 将FLTINJ位（ECCCONL[0]）置1，使能ECC错误注入逻辑。
6. 对闪存目标地址执行读或写操作。

# dsPIC33CK64MP105系列

## 5.4.2 ECC控制寄存器

### 寄存器5-7: ECCCONL: ECC错误注入配置低位字寄存器

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

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	FLTINJ
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 15-1

未实现: 读为0

bit 0

**FLTINJ:** 错误注入序列使能位

1 = 使能

0 = 禁止

### 寄存器5-8: ECCCONH: ECC错误注入配置高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FLT2PTR[7:0]							
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
FLT1PTR[7:0]							
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 15-8

**FLT2PTR[7:0]:** ECC错误注入位指针2位

11111111-00111000 = 未发生错误注入

00110111 = ECC位顺序的bit 55发生错误注入 (位翻转)

•

•

•

00000001 = ECC位顺序的bit 1发生错误注入 (位翻转)

00000000 = ECC位顺序的bit 0发生错误注入 (位翻转)

bit 7-0

**FLT1PTR[7:0]:** ECC错误注入位指针1位

11111111-00111000 = 未发生错误注入

00110111 = ECC位顺序的bit 55发生错误注入

•

•

•

00000001 = ECC位顺序的bit 1发生错误注入

00000000 = ECC位顺序的bit 0发生错误注入



## 寄存器5-9: ECCADDRL: ECC错误注入地址比较低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ECCADDR[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
ECCADDR[7:0]							
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 15-0 **ECCADDR[15:0]:** ECC错误注入NVM地址匹配比较位

## 寄存器5-10: ECCADDRH: ECC错误注入地址比较高位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
ECCADDR[23:16]							
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 15-8 **未实现:** 读为0

bit 7-0 **ECCADDR[23:16]:** ECC错误注入NVM地址匹配比较位

# dsPIC33CK64MP105系列

## 寄存器5-11: ECCSTATL: ECC系统状态显示低位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECOUT[7:0]							
bit 15							
bit 8							

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECIN[7:0]							
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 15-8                **SECOUT[7:0]:** 计算得到的单个位错误纠正奇偶校验值位

bit 7-0                **SECIN[7:0]:** 读取的单个位错误纠正奇偶校验值位  
 SECIN[7:0]位是闪存读取操作的实际奇偶校验值。

## 寄存器5-12: ECCSTATH: ECC系统状态显示高位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	DEDOUT	DEDIN
bit 15							
bit 8							

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SECSYND[7:0]							
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 15-10              **未实现:** 读为0

bit 9                    **DEDOUT:** 计算得到的双位错误检测奇偶校验位

bit 8                    **DEDIN:** 读取的双位错误检测奇偶校验位  
 DEDIN是闪存读取操作的实际奇偶校验值。

bit 7-0                **SECSYND[7:0]:** 计算得到的ECC校正子值位  
 表示包含错误的位位置。

5.5 ICSP写禁止

ICSP 写禁止是一种访问限制功能，激活后会限制所有闪存。一旦激活，ICSP 写禁止将永久阻止ICSP 闪存编程和擦除操作，并且无法取消激活。此功能旨在防止更改闪存内容，其行为类似于可一次性编程（OTP）器件。

在激活ICSP写禁止时，RTSP（包括擦除和编程操作）不受限制；不过，在激活ICSP写禁止之前，必须将执行这些操作的代码编程到器件中。这允许自举程序类型的应用程序在激活ICSP写禁止的情况下更改闪存内容。

进入ICSP和增强型ICSP模式不受ICSP写禁止的影响。在这些模式下，可继续读取配置存储空间和任何未受代码保护的用户存储空间区域。禁止ICSP写操作时，如果尝试将WR（NVMCON[15]）置1，则WR仍保持为0，而WRERR（NVMCON[13]）会置1。所有增强型ICSP擦除和编程命令都不起作用，自检查编程命令会返回FAIL响应操作码（如果目标已经与所请求的编程数据完全匹配，则返回PASS）。

激活ICSP写禁止后，在调试模式下执行的器件无法擦除/写入闪存，调试工具也无法将器件切换到生产模式。因此，只能在进行生产编程的器件上激活ICSP写禁止。

JTAG端口使能后，可用于将ICSP信号映射到JTAG I/O引脚。因此，在激活ICSP写禁止后，也会阻止通过JTAG端口启动的所有闪存擦除/编程操作。

5.5.1 激活ICSP写禁止

**小心：** 无法取消激活ICSP写禁止。

激活ICSP写禁止的方式是执行一对NVMCON双字编程命令，以将两个16位激活值保存在配置存储空间中。[表5-1](#)列出了激活所需的目标NVM地址和值。一旦两个地址都包含其激活值，ICSP写禁止将在下一次器件复位后产生永久性影响。在任一地址成功编程后，即使另一个地址尚未编程，也不能通过任何方式来复位、擦除或修改该地址内容。

仅评估存储在激活地址中的低16位数据：通过双字编程（NVMOP[3:0]）写入的高8位和第二个24位字应写为0。地址可以按任何顺序进行编程，也可以在单独的ICSP/增强型ICSP/RTSP会话期间进行编程，但编程不正确的16位值或使用行编程操作编程值的任何尝试都将被中止，而不会改变现有数据。

表5-1: ICSP写禁止激活地址和数据

	配置存储器地址	ICSP 写禁止激活值
写锁定1	0x801030	0x006D63
写锁定2	0x801034	0x006870

注:

## 6.0 复位

- 注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“复位”（[www.microchip.com/DS70602](http://www.microchip.com/DS70602)）。
- 2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。

复位模块结合了所有复位源并控制器件的主复位信号SYSRST。下面列出了器件的复位源：

- POR：上电复位
- BOR：欠压复位
- MCLR：主复位引脚复位
- SWR：RESET指令
- WDTO：看门狗定时器超时复位
- CM：配置不匹配复位
- TRAPR：陷阱冲突复位
- IOPUWR：非法条件器件复位
  - 非法操作码复位
  - 未初始化的W寄存器复位
  - 安全性复位

图6-1给出了复位模块的简化框图。

任何有效的复位源都将使SYSRST信号有效。在系统复位时，一些与CPU和外设相关的寄存器被强制为已知的复位状态，而有一些寄存器不受影响。

**注：** 如需了解寄存器复位状态的信息，请参见本手册中的特定外设章节或第4.0页“存储器构成”。

任何类型的器件复位都会将RCON寄存器中相应的状态位置1，以指示复位类型（见寄存器6-1）。

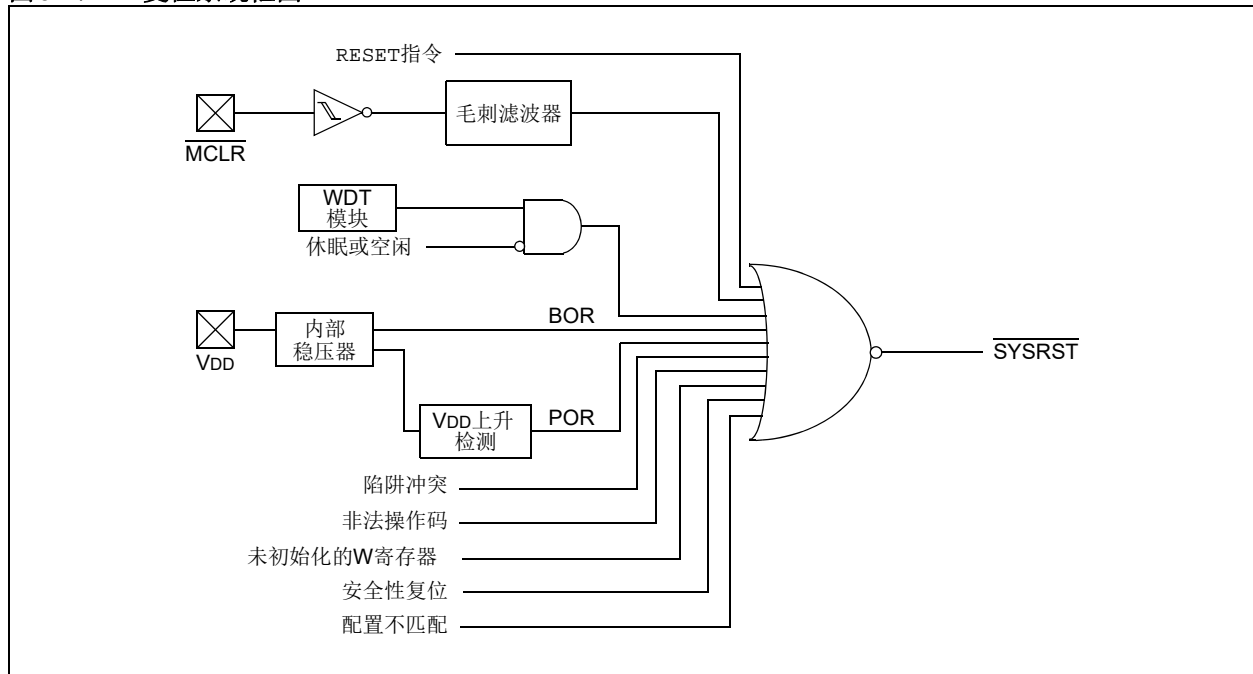
POR将清零除BOR和POR位（RCON[1:0]）之外的所有位，BOR和POR位在POR时被置1。用户应用程序可在代码执行过程中的任何时间置1或清零RCON寄存器中的任何位。RCON寄存器中的位仅用作状态位。用软件将某个复位状态位置1不会导致器件发生复位。

RCON寄存器还包含与看门狗定时器和器件节能状态相关的其他位。本手册的其他章节中将讨论这些位的功能。

**注：** RCON寄存器中的状态位应该在被读取后清零，这样在器件复位后RCON寄存器的下一个值才有意义。

在所有类型的复位中，默认时钟源由FOSCSEL配置寄存器中的FNOSC[2:0]位决定。复位时，FNOSC<sub>x</sub>位的值装入NOSC[2:0]（OSCCON[10:8]）位，从而初始化系统时钟。

图6-1： 复位系统框图



## 6.1 复位资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 6.1.1 主要资源

- 《dsPIC33/PIC24系列参考手册》中的“复位” ([www.microchip.com/DS70602](http://www.microchip.com/DS70602))
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24系列参考手册》章节
- 开发工具

## 寄存器 6-1: RCON: 复位控制寄存器<sup>(1)</sup>

R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
TRAPR	IOPUWR	—	—	—	—	CM	VREGS
bit 15						bit 8	

R/W-1	R/W-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1	R/W-1
EXTR	SWR	—	WDTO	SLEEP	IDLE	BOR	POR
bit 7						bit 0	

图注: r = 保留位  
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 15 **TRAPR:** 陷阱复位标志位  
1 = 发生了陷阱冲突复位  
0 = 未发生陷阱冲突复位
- bit 14 **IOPUWR:** 非法操作码或访问未初始化的W寄存器复位标志位  
1 = 检测到非法操作码、非法地址模式或将未初始化的W寄存器用作地址指针而导致复位  
0 = 未发生非法操作码或未初始化的W寄存器复位
- bit 13-10 **未实现:** 读为0
- bit 9 **CM:** 配置不匹配标志位  
1 = 发生了配置不匹配复位  
0 = 未发生配置不匹配复位
- bit 8 **VREGS:** 休眠模式下稳压器待机位  
1 = 在休眠模式下稳压器继续工作  
0 = 在休眠模式下稳压器进入待机模式
- bit 7 **EXTR:** 外部复位 ( $\overline{\text{MCLR}}$ ) 引脚位  
1 = 发生了主复位 (引脚) 复位  
0 = 未发生主复位 (引脚) 复位
- bit 6 **SWR:** 软件RESET (指令) 标志位  
1 = 执行了RESET指令  
0 = 未执行RESET指令
- bit 5 **保留:** 读为0
- bit 4 **WDTO:** 看门狗定时器超时标志位  
1 = 发生了WDT超时  
0 = 未发生WDT超时
- bit 3 **SLEEP:** 从休眠模式唤醒标志位  
1 = 器件处于休眠模式  
0 = 器件不处于休眠模式
- bit 2 **IDLE:** 从空闲模式唤醒标志位  
1 = 器件处于空闲模式  
0 = 器件不处于空闲模式
- bit 1 **BOR:** 欠压复位标志位  
1 = 发生了欠压复位  
0 = 未发生欠压复位
- bit 0 **POR:** 上电复位标志位  
1 = 发生了上电复位  
0 = 未发生上电复位

**注 1:** 所有复位状态位都可以用软件置1或清零。用软件将这些位中的某一位置1不会导致器件复位。

注:



## 7.0 中断控制器

- 注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“中断”([www.microchip.com/DS70000600](http://www.microchip.com/DS70000600))。
- 2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。

dsPIC33CK64MP105系列中断控制器将诸多外设中断请求信号缩减为一个送往dsPIC33CK64MP105系列CPU的中断请求信号。

中断控制器具有以下特性:

- 6个处理器异常和软件陷阱
- 7个可由用户选择的优先级
- 中断向量表 (IVT), 其中每个中断或异常源对应惟一的向量
- 在指定的用户优先级内具有固定的优先级
- 固定的中断进入和返回延时
- 用于支持调试功能的备用中断向量表 (Alternate Interrupt Vector Table, AIVT)

### 7.1 中断向量表

dsPIC33CK64MP105系列的中断向量表 (IVT) (如图7-1所示) 位于程序存储器中, 起始单元地址是000004h。IVT包含6个不可屏蔽陷阱向量和最多246个中断源。一般来说, 每个中断源都有自己的中断向量。每个中断向量都包含一个24位宽的地址。每个中断向量单元中编程的值是其相关的中断服务程序 (ISR) 的起始地址。

中断向量根据其自然优先级区分优先顺序。自然优先级与中断向量在向量表中的位置有关。一般来说, 较低地址的中断向量具有较高的自然优先级。例如, 与向量0相关的中断比任何其他向量地址的中断具有更高的自然优先级。

#### 7.1.1 备用中断向量表

只有定义了引导段 (Boot Segment, BS) 并已使能备用中断向量表 (AIVT) 时, AIVT (如图7-2所示) 才可用。要使能备用中断向量表, 必须编程FSEC寄存器中的配置位BSEN和AIVTDIS并将AIVTEN位置1 (INTCON2[8] = 1)。使能AIVT时, 所有中断和异常处理都将使用备用向量, 而非默认向量。AIVT从引导段最后一页 (由BSLIM[12:0]定义) 的起始处开始。该页的另一半不再为可用的空间。引导段必须至少具有2页以使能AIVT。

**注:** 尽管必须使能引导段才能使能AIVT, 但应用程序代码无需位于引导段内部。AIVT (和IVT) 将继承引导段代码保护。

AIVT通过提供一种不需要将中断向量再编程就可以在应用程序和支持环境之间切换的方法, 来支持调试功能。此特性也支持运行时在不同应用程序之间切换以便评估各种不同的软件算法。

### 7.2 复位序列

器件复位不是真正的异常, 因为复位过程中并不涉及到中断控制器。作为对复位的响应, dsPIC33CK64MP105系列器件清零其寄存器, 强制PC为零。然后器件从地址0x000000处开始执行程序。用户可以在复位地址处编程一条GOTO指令, 将程序执行重定向到相应的启动程序。

**注:** 应使用包含RESET指令的默认中断处理程序的地址编程IVT中所有未实现或未使用的向量单元。

# dsPIC33CK64MP105系列

图7-1: dsPIC33CK64MP105 系列中断向量表

自然顺序优先级降序排列	复位——GOTO指令	0x000000	有关陷阱向量的详细信息， 请参见表7-1
	复位——GOTO地址	0x000002	
	振荡器故障陷阱向量	0x000004	
	地址错误陷阱向量	0x000006	
	通用硬陷阱向量	0x000008	
	堆栈错误陷阱向量	0x00000A	
	数学错误陷阱向量	0x00000C	
	保留	0x00000E	
	通用软陷阱向量	0x000010	
	保留	0x000012	
	中断向量0	0x000014	有关中断向量的详细信息， 请参见表7-2
	中断向量1	0x000016	
	:	:	
	:	:	
	:	:	
	中断向量52	0x00007C	
	中断向量53	0x00007E	
	中断向量54	0x000080	
	:	:	
	:	:	
	中断向量116	0x0000FC	有关中断向量的详细信息， 请参见表7-2
	中断向量117	0x0000FE	
	中断向量118	0x000100	
	中断向量119	0x000102	
	中断向量120	0x000104	
	:	:	
	:	:	
	:	:	
	中断向量244	0x0001FC	
	中断向量245	0x0001FE	
	代码起始单元	0x000200	

图7-2: dsPIC33CK64MP105 备用中断向量表

自然顺序优先级降序排列	保留	BSLIM[12:0] <sup>(1)</sup> + 0x000000	有关陷阱向量的详细信息， 请参见表7-1
	保留	BSLIM[12:0] <sup>(1)</sup> + 0x000002	
	振荡器故障陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x000004	
	地址错误陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x000006	
	通用硬陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x000008	
	堆栈错误陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x00000A	
	数学错误陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x00000C	
	保留	BSLIM[12:0] <sup>(1)</sup> + 0x00000E	
	通用软陷阱向量	BSLIM[12:0] <sup>(1)</sup> + 0x000010	
	保留	BSLIM[12:0] <sup>(1)</sup> + 0x000012	
	中断向量0	BSLIM[12:0] <sup>(1)</sup> + 0x000014	有关中断向量的详细信息， 请参见表7-2
	中断向量1	BSLIM[12:0] <sup>(1)</sup> + 0x000016	
	:	:	
	:	:	
	:	:	
	中断向量52	BSLIM[12:0] <sup>(1)</sup> + 0x00007C	
	中断向量53	BSLIM[12:0] <sup>(1)</sup> + 0x00007E	
	中断向量54	BSLIM[12:0] <sup>(1)</sup> + 0x000080	
	:	:	
	:	:	
	:	:	
	中断向量116	BSLIM[12:0] <sup>(1)</sup> + 0x0000FC	
	中断向量117	BSLIM[12:0] <sup>(1)</sup> + 0x0000FE	
	中断向量118	BSLIM[12:0] <sup>(1)</sup> + 0x000100	
	中断向量119	BSLIM[12:0] <sup>(1)</sup> + 0x000102	
	中断向量120	BSLIM[12:0] <sup>(1)</sup> + 0x000104	
	:	:	
	:	:	
	:	:	
	中断向量244	BSLIM[12:0] <sup>(1)</sup> + 0x0001FC	
	中断向量245	BSLIM[12:0] <sup>(1)</sup> + 0x0001FE	

注 1: 地址取决于BSLIM[12:0]定义的引导段的大小:  $[(BSLIM[12:0] - 1) \times 0x800] + \text{偏移量}$ 。

# dsPIC33CK64MP105系列

表7-1: 陷阱向量详细信息

陷阱说明	MPLAB® XC16 陷阱ISR名称	IVT 地址	陷阱位的位置			优先级
			中断标志	类型	允许位	
振荡器故障	_OscillatorFail	0x000004	INTCON1[1]	—	—	15
地址错误	_AddressError	0x000006	INTCON1[3]	—	—	14
ECC 双位错误	_HardTrapError	0x000008	INTCON4[1]	—	—	13
软件生成的陷阱	_HardTrapError	0x000008	INTCON4[0]	—	INTCON2[13]	13
堆栈错误	_StackError	0x00000A	INTCON1[2]	—	—	12
累加器A溢出	_MathError	0x00000C	INTCON1[4]	INTCON1[14]	INTCON1[10]	11
累加器B溢出	_MathError	0x00000C	INTCON1[4]	INTCON1[13]	INTCON1[9]	11
累加器A灾难性溢出	_MathError	0x00000C	INTCON1[4]	INTCON1[12]	INTCON1[8]	11
累加器B灾难性溢出	_MathError	0x00000C	INTCON1[4]	INTCON1[11]	INTCON1[8]	11
累加器移位错误	_MathError	0x00000C	INTCON1[4]	INTCON1[7]	INTCON1[8]	11
被零除错误	_MathError	0x00000C	INTCON1[4]	INTCON1[6]	INTCON1[8]	11
保留	保留	0x00000E	—	—	—	—
NVM 地址错误	_SoftTrapError	0x000010	INTCON3[8]	—	—	9
DMA 地址错误	_SoftTrapError	0x000010	INTCON3[5]	—	—	9
DO 堆栈溢出	_SoftTrapError	0x000010	INTCON3[4]	—	—	9
APLL 失锁	_SoftTrapError	0x000010	INTCON3[0]	—	—	9
保留	保留	0x000012	—	—	—	—

表7-2: 中断向量详细信息

中断源	MPLAB® XC16 ISR 名称	向量 编号	IRQ 编号	IVT 地址	中断位的位置		
					标志位	允许位	优先级
外部中断0	_INT0Interrupt	8	0	0x000014	IFS0[0]	IEC0[0]	IPC0[2:0]
Timer1	_T1Interrupt	9	1	0x000016	IFS0[1]	IEC0[1]	IPC0[6:4]
电平变化通知中断A	_CNAInterrupt	10	2	0x000018	IFS0[2]	IEC0[2]	IPC0[10:8]
电平变化通知中断B	_CNBInterrupt	11	3	0x00001A	IFS0[3]	IEC0[3]	IPC0[14:12]
DMA通道0	_DMA0Interrupt	12	4	0x00001C	IFS0[4]	IEC0[4]	IPC1[2:0]
保留	保留	13	5	0x00001E	—	—	—
输入捕捉/输出比较1	_CCP1Interrupt	14	6	0x000020	IFS0[6]	IEC0[6]	IPC1[10:8]
CCP1定时器	_CCT1Interrupt	15	7	0x000022	IFS0[7]	IEC0[7]	IPC1[14:12]
DMA通道1	_DMA1Interrupt	16	8	0x000024	IFS0[8]	IEC0[8]	IPC2[2:0]
SPI1接收器	_SPI1RXInterrupt	17	9	0x000026	IFS0[9]	IEC0[9]	IPC2[6:4]
SPI1发送器	_SPI1TXInterrupt	18	10	0x000028	IFS0[10]	IEC0[10]	IPC2[10:8]
UART1接收器	_U1RXInterrupt	19	11	0x00002A	IFS0[11]	IEC0[11]	IPC2[14:12]
UART1发送器	_U1TXInterrupt	20	12	0x00002C	IFS0[12]	IEC0[12]	IPC3[2:0]
ECC单个位错误	_ECCSBEInterrupt	21	13	0x00002E	IFS0[13]	IEC0[13]	IPC3[6:4]
NVM写完成	_NVMInterrupt	22	14	0x000030	IFS0[14]	IEC0[14]	IPC3[10:8]
外部中断1	_INT1Interrupt	23	15	0x000032	IFS0[15]	IEC0[15]	IPC3[14:12]
I2C1从事件	_SI2C1Interrupt	24	16	0x000034	IFS1[0]	IEC1[0]	IPC4[2:0]
I2C1主事件	_MI2C1Interrupt	25	17	0x000036	IFS1[1]	IEC1[1]	IPC4[6:4]
DMA通道2	_DMA2Interrupt	26	18	0x000038	IFS1[2]	IEC1[2]	IPC4[10:8]
电平变化通知中断C	_CNCInterrupt	27	19	0x00003A	IFS1[3]	IEC1[3]	IPC4[14:12]
外部中断2	_INT2Interrupt	28	20	0x00003C	IFS1[4]	IEC1[4]	IPC5[2:0]
DMA通道3	_DMA3Interrupt	29	21	0x00003E	IFS1[5]	IEC1[5]	IPC5[6:4]
保留	保留	30	22	0x000040	—	—	—
输入捕捉/输出比较2	_CCP2Interrupt	31	23	0x000042	IFS1[7]	IEC1[7]	IPC5[14:12]
CCP2定时器	_CCT2Interrupt	32	24	0x000044	IFS1[8]	IEC1[8]	IPC6[2:0]
保留	保留	33	25	0x000046	—	—	—
外部中断3	_INT3Interrupt	34	26	0x000048	IFS1[10]	IEC1[10]	IPC6[10:8]
U2RX——UART2接收器	_U2RXInterrupt	35	27	0x00004A	IFS1[11]	IEC1[11]	IPC6[14:12]
U2TX——UART2发送器	_U2TXInterrupt	36	28	0x00004C	IFS1[12]	IEC1[12]	IPC7[2:0]
SPI2接收器	_SPI2RXInterrupt	37	29	0x00004E	IFS1[13]	IEC1[13]	IPC7[6:4]
SPI2发送器	_SPI2TXInterrupt	38	30	0x000050	IFS1[14]	IEC1[14]	IPC7[10:8]
保留	保留	39-42	31-34	0x000052-0x000058	—	—	—
输入捕捉/输出比较3	_CCP3Interrupt	43	35	0x00005A	IFS2[3]	IEC2[3]	IPC8[14:12]
CCP3定时器	_CCT3Interrupt	44	36	0x00005C	IFS2[4]	IEC2[4]	IPC9[2:0]
I2C2从事件	_SI2C2Interrupt	45	37	0x00005E	IFS2[5]	IEC2[5]	IPC9[6:4]
I2C2主事件	_MI2C2Interrupt	46	38	0x000060	IFS2[6]	IEC2[6]	IPC9[10:8]
保留	保留	47	39	0x000062	—	—	—
输入捕捉/输出比较4	_CCP4Interrupt	48	40	0x000064	IFS2[8]	IEC2[8]	IPC10[2:0]
CCP4定时器	_CCT4Interrupt	49	41	0x000066	IFS2[9]	IEC2[9]	IPC10[6:4]
保留	保留	50	42	0x000068	—	—	—
输入捕捉/输出比较5	_CCP5Interrupt	51	43	0x00006A	IFS2[11]	IEC2[11]	IPC10[14:12]
CCP5定时器	_CCT5Interrupt	52	44	0x00006C	IFS2[12]	IEC2[12]	IPC11[2:0]
程序监控定时器	_DMTInterrupt	53	45	0x00006E	IFS2[13]	IEC2[13]	IPC11[6:4]
保留	保留	54-55	46-47	0x000070-0x000072	—	—	—

# dsPIC33CK64MP105系列

表7-2: 中断向量详细信息 (续)

中断源	MPLAB® XC16 ISR 名称	向量 编号	IRQ 编号	IVT 地址	中断位的位置		
					标志位	允许位	优先级
QE1 位置计数器比较	_QE1Interrupt	56	48	0x000074	IFS3[0]	IEC3[0]	IPC12[2:0]
UART1 错误	_U1EInterrupt	57	49	0x000076	IFS3[1]	IEC3[1]	IPC12[6:4]
UART2 错误	_U2EInterrupt	58	50	0x000078	IFS3[2]	IEC3[2]	IPC12[10:8]
CRC 发生器	_CRCInterrupt	59	51	0x00007A	IFS3[3]	IEC3[3]	IPC12[14:12]
保留	保留	60-61	52-53	0x00007C-0x00007E	—	—	—
QE1 位置计数器比较	_QE12Interrupt	62	54	0x000080	IFS3[6]	IEC3[6]	IPC13[10:8]
保留	保留	63	55	0x000082	—	—	—
UART3 错误	_U3EInterrupt	64	56	0x000084	IFS3[8]	IEC3[8]	IPC14[2:0]
UART3 接收器	_U3RXInterrupt	65	57	0x000086	IFS3[9]	IEC3[9]	IPC14[6:4]
UART3 发送器	_U3TXInterrupt	66	58	0x000088	IFS3[10]	IEC3[10]	IPC14[10:8]
SPI3 接收器	_SPI3RXInterrupt	67	59	0x00008A	IFS3[11]	IEC3[11]	IPC14[14:12]
SPI3 发送器	_SPI3TXInterrupt	68	60	0x00008C	IFS3[12]	IEC3[12]	IPC15[2:0]
保留	保留	69-70	61-62	0x00008E-0x000090	—	—	—
PTG 步阶	_PTGSTEPInterrupt	71	63	0x000092	IFS3[15]	IEC3[15]	IPC15[14:12]
I2C1 总线冲突	_I2C1BCInterrupt	72	64	0x000094	IFS4[0]	IEC4[0]	IPC16[2:0]
I2C2 总线冲突	_I2C2BCInterrupt	73	65	0x000096	IFS4[1]	IEC4[1]	IPC16[6:4]
保留	保留	74	66	0x000098	—	—	—
PWM 发生器 1	_PWM1Interrupt	75	67	0x00009A	IFS4[3]	IEC4[3]	IPC16[14:12]
PWM 发生器 2	_PWM2Interrupt	76	68	0x00009C	IFS4[4]	IEC4[4]	IPC17[2:0]
PWM 发生器 3	_PWM3Interrupt	77	69	0x00009E	IFS4[5]	IEC4[5]	IPC17[6:4]
PWM 发生器 4	_PWM4Interrupt	78	70	0x0000A0	IFS4[6]	IEC4[6]	IPC17[10:8]
保留	保留	79-82	71-74	0x0000A2-0x0000A8	—	—	—
电平变化通知 D	_CNDInterrupt	83	75	0x0000AA	IFS4[11]	IEC4[11]	IPC18[14:12]
保留	保留	84	76	0x0000AC	—	—	—
比较器 1	_CMP1Interrupt	85	77	0x0000AE	IFS4[13]	IEC4[13]	IPC19[6:4]
比较器 2	_CMP2Interrupt	86	78	0x0000B0	IFS4[14]	IEC4[14]	IPC19[10:8]
比较器 3	_CMP3Interrupt	87	79	0x0000B2	IFS4[15]	IEC4[15]	IPC19[14:12]
保留	保留	88	80	0x0000B4	—	—	—
PTG 看门狗定时器超时	_PTGWDTInterrupt	89	81	0x0000B6	IFS5[1]	IEC5[1]	IPC20[6:4]
PTG 触发信号 0	_PTG0Interrupt	90	82	0x0000B8	IFS5[2]	IEC5[2]	IPC20[10:8]
PTG 触发信号 1	_PTG1Interrupt	91	83	0x0000BA	IFS5[3]	IEC5[3]	IPC20[14:12]
PTG 触发信号 2	_PTG2Interrupt	92	84	0x0000BC	IFS5[4]	IEC5[4]	IPC21[2:0]
PTG 触发信号 3	_PTG3Interrupt	93	85	0x0000BE	IFS5[5]	IEC5[6]	IPC21[6:4]
SENT1 TX/RX	_SENT1Interrupt	94	86	0x0000C0	IFS5[6]	IEC5[6]	IPC21[10:8]
SENT1 错误	_SENT1EInterrupt	95	87	0x0000C2	IFS5[7]	IEC5[7]	IPC21[14:12]
SENT2 TX/RX	_SENT2Interrupt	96	88	0x0000C4	IFS5[8]	IEC5[8]	IPC22[2:0]
SENT2 错误	_SENT2EInterrupt	97	89	0x0000C6	IFS5[9]	IEC5[9]	IPC22[6:4]
ADC 全局中断	_ADCInterrupt	98	90	0x0000C8	IFS5[10]	IEC5[10]	IPC22[10:8]

# dsPIC33CK64MP105系列

表7-2: 中断向量详细信息 (续)

中断源	MPLAB® XC16 ISR 名称	向量 编号	IRQ 编号	IVT 地址	中断位的位置		
					标志位	允许位	优先级
ADC AN0 中断	_ADCAN0Interrupt	99	91	0x0000CA	IFS5[11]	IEC5[11]	IPC22[14:12]
ADC AN1 中断	_ADCAN1Interrupt	100	92	0x0000CC	IFS5[12]	IEC5[12]	IPC23[2:0]
ADC AN2 中断	_ADCAN2Interrupt	101	93	0x0000CE	IFS5[13]	IEC5[13]	IPC23[6:4]
ADC AN3 中断	_ADCAN3Interrupt	102	94	0x0000D0	IFS5[14]	IEC5[14]	IPC23[10:8]
ADC AN4 中断	_ADCAN4Interrupt	103	95	0x0000D2	IFS5[15]	IEC5[15]	IPC23[14:12]
ADC AN5 中断	_ADCAN5Interrupt	104	96	0x0000D4	IFS6[0]	IEC6[0]	IPC24[2:0]
ADC AN6 中断	_ADCAN6Interrupt	105	97	0x0000D6	IFS6[1]	IEC6[1]	IPC24[6:4]
ADC AN7 中断	_ADCAN7Interrupt	106	98	0x0000D8	IFS6[2]	IEC6[2]	IPC24[10:8]
ADC AN8 中断	_ADCAN8Interrupt	107	99	0x0000DA	IFS6[3]	IEC6[3]	IPC24[14:12]
ADC AN9 中断	_ADCAN9Interrupt	108	100	0x0000DC	IFS6[4]	IEC6[4]	IPC25[2:0]
ADC AN10 中断	_ADCAN10Interrupt	109	101	0x0000DE	IFS6[5]	IEC6[5]	IPC25[6:4]
ADC AN11 中断	_ADCAN11Interrupt	110	102	0x0000E0	IFS6[6]	IEC6[6]	IPC25[10:8]
ADC AN12 中断	_ADCAN12Interrupt	111	103	0x0000E2	IFS6[7]	IEC6[7]	IPC25[14:12]
ADC AN13 中断	_ADCAN13Interrupt	112	104	0x0000E4	IFS6[8]	IEC6[8]	IPC26[2:0]
ADC AN14 中断	_ADCAN14Interrupt	113	105	0x0000E6	IFS6[9]	IEC6[9]	IPC26[6:4]
ADC AN15 中断	_ADCAN15Interrupt	114	106	0x0000E8	IFS6[10]	IEC6[10]	IPC26[10:8]
ADC AN16 中断	_ADCAN16Interrupt	115	107	0x0000EA	IFS6[11]	IEC6[11]	IPC26[14:12]
ADC AN17 中断	_ADCAN17Interrupt	116	108	0x0000EC	IFS6[12]	IEC6[12]	IPC27[2:0]
ADC AN18 中断	_ADCAN18Interrupt	117	109	0x0000EE	IFS6[13]	IEC6[13]	IPC27[6:4]
ADC AN19 中断	_ADCAN19Interrupt	118	110	0x0000F0	IFS6[14]	IEC6[14]	IPC27[10:8]
ADC AN20 中断	_ADCAN20Interrupt	119	111	0x0000F2	IFS6[15]	IEC6[15]	IPC27[14:12]
保留	保留	120-123	112-115	0x0000F4-0x0000FA	—	—	—
ADC 数字比较器0	_ADCMPOInterrupt	124	116	0x0000FC	IFS7[4]	IEC7[4]	IPC29[2:0]
ADC 数字比较器1	_ADCMPIInterrupt	125	117	0x0000FE	IFS7[5]	IEC7[5]	IPC29[6:4]
ADC 数字比较器2	_ADCMPIInterrupt	126	118	0x000100	IFS7[6]	IEC7[6]	IPC29[10:8]
ADC 数字比较器3	_ADCMPIInterrupt	127	119	0x000102	IFS7[7]	IEC7[7]	IPC29[14:12]
ADC 过采样滤波器0	_ADFLTR0Interrupt	128	120	0x000104	IFS7[8]	IEC7[8]	IPC30[2:0]
ADC 过采样滤波器1	_ADFLTR1Interrupt	129	121	0x000106	IFS7[9]	IEC7[9]	IPC30[6:4]
ADC 过采样滤波器2	_ADFLTR2Interrupt	130	122	0x000108	IFS7[10]	IEC7[10]	IPC30[10:8]
ADC 过采样滤波器3	_ADFLTR3Interrupt	131	123	0x00010A	IFS7[11]	IEC7[11]	IPC30[14:12]
CLC1 正边沿	_CLC1PInterrupt	132	124	0x00010C	IFS7[12]	IEC7[12]	IPC31[2:0]
CLC2 正边沿	_CLC2PInterrupt	133	125	0x00010E	IFS7[13]	IEC7[13]	IPC31[6:4]
SPI1 错误	_SPI1Interrupt	134	126	0x000110	IFS7[14]	IEC7[14]	IPC31[10:8]
SPI2 错误	_SPI2Interrupt	135	127	0x000112	IFS7[15]	IEC7[15]	IPC31[14:12]
SPI3 错误	_SPI3Interrupt	136	128	0x000114	IFS8[0]	IEC8[0]	IPC32[2:0]
保留	保留	137-176	129-168	0x000116-0x000164	—	—	—
PEVTA——PWM 事件A	_PEVTAInterrupt	177	169	0x000166	IFS10[9]	IEC10[9]	IPC42[6:4]
PEVTB——PWM 事件B	_PEVTBInterrupt	178	170	0x000168	IFS10[10]	IEC10[10]	IPC42[10:8]
PEVTC——PWM 事件C	_PEVTCInterrupt	179	171	0x00016A	IFS10[11]	IEC10[11]	IPC42[14:12]
PEVTD——PWM 事件D	_PEVTDInterrupt	180	172	0x00016C	IFS10[12]	IEC10[12]	IPC43[2:0]
PEVTE——PWM 事件E	_PEVTEInterrupt	181	173	0x00016E	IFS10[13]	IEC10[13]	IPC43[6:4]
PEVTF——PWM 事件F	_PEVTFInterrupt	182	174	0x000170	IFS10[14]	IEC10[14]	IPC43[10:8]

# dsPIC33CK64MP105系列

表7-2: 中断向量详细信息 (续)

中断源	MPLAB® XC16 ISR 名称	向量 编号	IRQ 编号	IVT 地址	中断位的位置		
					标志位	允许位	优先级
CLC3 正边沿	_CLC3PInterrupt	183	175	0x000172	IFS10[15]	IEC10[15]	IPC43[14:12]
CLC4 正边沿	_CLC4PInterrupt	184	176	0x000174	IFS11[0]	IEC11[0]	IPC44[2:0]
CLC1 负边沿	_CLC1NInterrupt	185	177	0x000176	IFS11[1]	IEC11[1]	IPC44[6:4]
CLC2 负边沿	_CLC2NInterrupt	186	178	0x000178	IFS11[2]	IEC11[2]	IPC44[10:8]
CLC3 负边沿	_CLC3NInterrupt	187	179	0x00017A	IFS11[3]	IEC11[3]	IPC44[14:12]
CLC4 负边沿	_CLC4NInterrupt	188	180	0x00017C	IFS11[4]	IEC11[4]	IPC45[2:0]
保留	保留	189-196	181-188	0x0017E-0x0018C	—	—	—
UART1 事件	_U1EVTInterrupt	197	189	0x00018E	IFS11[13]	IF2C11[13]	IPC47[6:4]
UART2 事件	_U2EVTInterrupt	198	190	0x000190	IFS11[14]	IF2C11[14]	IPC47[12:8]
UART3 事件	_U3EVTInterrupt	199	191	0x000192	IFS11[15]	IF2C11[15]	IPC47[14:12]
保留	保留	200-255	192-247	0x000194-0x0001FE	—	—	—



**表 7-3: 中断标志寄存器**

寄存器	地址	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IFS0	800h	INT1IF	NVMIF	ECCSBEIF	U1TXIF	U1RXIF	SPI1TXIF	SPI1RXIF	DMA1IF	CCT1IF	CCP1IF	—	DMA0IF	CNBIF	CNAIF	T1IF	INT0IF
IFS1	802h	—	SPI2TXIF	SPI2RXIF	U2TXIF	U2RXIF	INT3IF	—	CCT2IF	CCP2IF	—	DMA3IF	INT2IF	CNCIF	DMA2IF	MI2C1IF	SI2C1IF
IFS2	804h	—	—	DMTIF	CCT5IF	CCP5IF	—	CCT4IF	CCP4IF	—	MI2C2IF	SI2C2IF	CCT3IF	CCP3IF	—	—	—
IFS3	806h	PTGSTIEIF	—	—	SPI3TXIF	SPI3RXIF	U3TXIF	U3RXIF	U3EIF	—	QE12IF	—	—	CRCIF	U2EIF	U1EIF	QE11IF
IFS4	808h	CMP3IF	CMP2IF	CMP1IF	—	CNDIF	—	—	—	—	PWM4IF	PWM3IF	PWM2IF	PWM1IF	—	I2C2BCIF	I2C1BCIF
IFS5	80Ah	ADCAN4IF	ADCAN3IF	ADCAN2IF	ADCAN1IF	ADCAN0IF	ADCIF	SENT2EIF	SENT2IF	SENT1EIF	SENT1IF	PTG3IF	PTG2IF	PTG1IF	PTG0IF	PTGWDTIF	—
IFS6	80Ch	ADCAN20IF	ADCAN19IF	ADCAN18IF	ADCAN17IF	ADCAN16IF	ADCAN15IF	ADCAN14IF	ADCAN13IF	ADCAN12IF	ADCAN11IF	ADCAN10IF	ADCAN9IF	ADCAN8IF	ADCAN7IF	ADCAN6IF	ADCAN5IF
IFS7	80Eh	SPI2GIF	SPI1GIF	CLC2PIF	CLC1PIF	ADFLTR3IF	ADFLTR2IF	ADFLTR1IF	ADFLTR0IF	ADCOMP3IF	ADCOMP2IF	ADCOMP1IF	ADCOMP0IF	—	—	—	—
IFS8	810h	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3GIF
IFS10	814h	CLC3PIF	PEVTFIF	PEVTEIF	PEVTDIF	PEVTCIF	PEVTBIF	PEVTAIF	—	—	—	—	—	—	—	—	—
IFS11	816h	U3EVTIF	U2EVTIF	U1EVTIF	—	—	—	—	—	—	—	—	—	CLC4NIF	CLC3NIF	CLC2NIF	CLC1NIF

图注: — = 未实现。

**表 7-4: 中断允许寄存器**

寄存器	地址	Bit 15	Bit14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEC0	820h	INT1IE	NVMIE	ECCSBEIE	U1TXIE	U1RXIE	SPI1TXIE	SPI1RXIE	DMA1IE	CCT1IE	CCP1IE	—	DMA0IE	CNBIE	CNAIE	T1IE	INT0IE
IEC1	822h	—	SPI2TXIE	SPI2RXIE	U2TXIE	U2RXIE	INT3IE	—	CCT2IE	CCP2IE	—	DMA3IE	INT2IE	CNCIE	DMA2IE	MI2C1IE	SI2C1IE
IEC2	824h	—	—	DMTIE	CCT5IE	CCP5IE	—	CCT4IE	CCP4IE	—	MI2C2IE	SI2C2IE	CCT3IE	CCP3IE	—	—	—
IEC3	826h	PTGSTIE	—	—	SPI3TXIE	SPI3RXIE	U3TXIE	U3RXIE	U3EIE	—	QE12IE	—	—	CRCIE	U2EIE	U1EIE	QE11IE
IEC4	828h	CMP3IE	CMP2IE	CMP1IE	—	CNDIE	—	—	—	—	PWM4IE	PWM3IE	PWM2IE	PWM1IE	—	I2C2BCIE	I2C1BCIE
IEC5	82Ah	ADCAN4IE	ADCAN3IE	ADCAN2IE	ADCAN1IE	ADCAN0IE	ADCIE	SENT2EIE	SENT2IE	SENT1EIE	SENT1IE	PTG3IE	PTG2IE	PTG1IE	PTG0IE	PTGWDTIE	—
IEC6	82Ch	ADCAN20IE	ADCAN19IE	ADCAN18IE	ADCAN17IE	ADCAN16IE	ADCAN15IE	ADCAN14IE	ADCAN13IE	ADCAN12IE	ADCAN11IE	ADCAN10IE	ADCAN9IE	ADCAN8IE	ADCAN7IE	ADCAN6IE	ADCAN5IE
IEC7	82Eh	SPI2GIE	SPI1GIE	CLC2PIE	CLC1PIE	ADFLTR3IE	ADFLTR2IE	ADFLTR1IE	ADFLTR0IE	ADCOMP3IE	ADCOMP2IE	ADCOMP1IE	ADCOMP0IE	—	—	—	—
IEC8	830h	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3GIE
IEC10	834h	CLC3PIE	PEVTFIE	PEVTEIE	PEVTDIE	PEVTCIE	PEVTBIE	PEVTAIE	—	—	—	—	—	—	—	—	—
IEC11	836h	U3EVTIE	U2EVTIE	U1EVTIE	—	—	—	—	—	—	—	—	—	CLC4NIE	CLC3NIE	CLC2NIE	CLC1NIE

图注: — = 未实现。

表7-5: 中断优先级寄存器

寄存器	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPC0	840h	—	CNBIP2	CNBIP1	CNBIP0	—	CNAIP2	CNAIP1	CNAIP0	—	T1IP2	T1IP1	T1IP0	—	INT0IP2	INT0IP1	INT0IP0
IPC1	842h	—	CCT1IP2	CCT1IP1	CCT1IP0	—	CCP1IP2	CCP1IP1	CCP1IP0	—	—	—	—	—	DMA0IP2	DMA0IP1	DMA0IP0
IPC2	844h	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1TXIP2	SPI1TXIP1	SPI1TXIP0	—	SPI1RXIP2	SPI1RXIP1	SPI1RXIP0	—	DMA1IP2	DMA1IP1	DMA1IP0
IPC3	846h	—	INT1IP2	INT1IP1	INT1IP0	—	NVMIP2	NVMIP1	NVMIP0	—	ECCSBEIP2	ECCSBEIP1	ECCSBEIP0	—	U1TXIP2	U1TXIP1	U1TXIP0
IPC4	848h	—	CNCIP2	CNCIP1	CNCIP0	—	DMA2IP2	DMA2IP1	DMA2IP0	—	M12C1IP2	M12C1IP1	M12C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
IPC5	84Ah	—	CCP2IP2	CCP2IP1	CCP2IP0	—	—	—	—	—	DMA3IP2	DMA3IP1	DMA3IP0	—	INT2IP2	INT2IP1	INT2IP0
IPC6	84Ch	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	—	CCT2IP2	CCT2IP1	CCT2IP0
IPC7	84Eh	—	—	—	—	—	SPI2TXIP2	SPI2TXIP1	SPI2TXIP0	—	SPI2RXIP2	SPI2RXIP1	SPI2RXIP0	—	U2TXIP2	U2TXIP1	U2TXIP0
IPC8	850h	—	CCP3IP2	CCP3IP1	CCP3IP0	—	—	—	—	—	—	—	—	—	—	—	—
IPC9	852h	—	—	—	—	—	M12C2IP2	M12C2IP1	M12C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	CCT3IP2	CCT3IP1	CCT3IP0
IPC10	854h	—	CCP5IP2	CCP5IP1	CCP5IP0	—	—	—	—	—	CCT4IP2	CCT4IP1	CCT4IP0	—	CCP4IP2	CCP4IP1	CCP4IP0
IPC11	856h	—	—	—	—	—	—	—	—	—	DMTIP2	DMTIP1	DMTIP0	—	CCT5IP2	CCT5IP1	CCT5IP0
IPC12	858h	—	CRCIP2	CRCIP1	CRCIP0	—	U2EIP2	U2EIP1	U2EIP0	—	U1EIP2	U1EIP1	U1EIP0	—	QE1IP2	QE1IP1	QE1IP0
IPC13	85Ah	—	—	—	—	—	QE12IP2	QE12IP1	QE12IP0	—	—	—	—	—	—	—	—
IPC14	85Ch	—	SPI3RXIP2	SPI3RXIP1	SPI3RXIP0	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3EIP2	U3EIP1	U3EIP0
IPC15	85Eh	—	PTGSTEIP2	PTGSTEIP1	PTGSTEIP0	—	—	—	—	—	—	—	—	—	SPI3TXIP2	SPI3TXIP1	SPI3TXIP0
IPC16	860h	—	PWM1IP2	PWM1IP1	PWM1IP0	—	—	—	—	—	I2C2BCIP2	I2C2BCIP1	I2C2BCIP0	—	I2C1BCIP2	I2C1BCIP1	I2C1BCIP0
IPC17	862h	—	—	—	—	—	PWM4IP2	PWM4IP1	PWM4IP0	—	PWM3IP2	PWM3IP1	PWM3IP0	—	PWM2IP2	PWM2IP1	PWM2IP0
IPC18	864h	—	CNDIP2	CNDIP1	CNDIP0	—	—	—	—	—	—	—	—	—	—	—	—
IPC19	866h	—	CMP3IP2	CMP3IP1	CMP3IP0	—	CMP2IP2	CMP2IP1	CMP2IP0	—	CMP1IP2	CMP1IP1	CMP1IP0	—	—	—	—
IPC20	868h	—	PTG1IP2	PTG1IP1	PTG1IP0	—	PTG0IP2	PTG0IP1	PTG0IP0	—	PTGWDIP2	PTGWDIP1	PTGWDIP0	—	—	—	—
IPC21	86Ah	—	SENT1EIP2	SENT1EIP1	SENT1EIP0	—	SENT1IP2	SENT1IP1	SENT1IP0	—	PTG3IP2	PTG3IP1	PTG3IP0	—	PTG2IP2	PTG2IP1	PTG2IP0
IPC22	86Ch	—	ADCAN0IP2	ADCAN0IP1	ADCAN0IP0	—	ADCIP2	ADCIP1	ADCIP0	—	SENT2EIP2	SENT2EIP1	SENT2EIP0	—	SENT2IP2	SENT2IP1	SENT2IP0
IPC23	86Eh	—	ADCAN4IP2	ADCAN4IP1	ADCAN4IP0	—	ADCAN3IP2	ADCAN3IP1	ADCAN3IP0	—	ADCAN2IP2	ADCAN2IP1	ADCAN2IP0	—	ADCAN1IP2	ADCAN1IP1	ADCAN1IP0
IPC24	870h	—	ADCAN8IP2	ADCAN8IP1	ADCAN8IP0	—	ADCAN7IP2	ADCAN7IP1	ADCAN7IP0	—	ADCAN6IP2	ADCAN6IP1	ADCAN6IP0	—	ADCAN5IP2	ADCAN5IP1	ADCAN5IP0
IPC25	872h	—	ADCAN12IP2	ADCAN12IP1	ADCAN12IP0	—	ADCAN11IP2	ADCAN11IP1	ADCAN11IP0	—	ADCAN10IP2	ADCAN10IP1	ADCAN10IP0	—	ADCAN9IP2	ADCAN9IP1	ADCAN9IP0
IPC26	874h	—	ADCAN16IP2	ADCAN16IP1	ADCAN16IP0	—	ADCAN15IP2	ADCAN15IP1	ADCAN15IP0	—	ADCAN14IP2	ADCAN14IP1	ADCAN14IP0	—	ADCAN13IP2	ADCAN13IP1	ADCAN13IP0
IPC27	876h	—	ADCAN20IP2	ADCAN20IP1	ADCAN20IP0	—	ADCAN19IP2	ADCAN19IP1	ADCAN19IP0	—	ADCAN18IP2	ADCAN18IP1	ADCAN18IP0	—	ADCAN17IP2	ADCAN17IP1	ADCAN17IP0
IPC29	87Ah	—	ADCMIP2	ADCMIP1	ADCMIP0	—	ADCMIP2	ADCMIP1	ADCMIP0	—	ADCMIP2	ADCMIP1	ADCMIP0	—	ADCMIP2	ADCMIP1	ADCMIP0
IPC30	87Ch	—	ADFLTR3IP2	ADFLTR3IP1	ADFLTR3IP0	—	ADFLTR2IP2	ADFLTR2IP1	ADFLTR2IP0	—	ADFLTR1IP2	ADFLTR1IP1	ADFLTR1IP0	—	ADFLTR0IP2	ADFLTR0IP1	ADFLTR0IP0
IPC31	87Eh	—	SPI2GIP2	SPI2GIP1	SPI2GIP0	—	SPI1GIP2	SPI1GIP1	SPI1GIP0	—	CLC2IP2	CLC2IP1	CLC2IP0	—	CLC1IP2	CLC1IP1	CLC1IP0
IPC32	880h	—	—	—	—	—	—	—	—	—	—	—	—	—	SPI3GIP2	SPI3GIP1	SPI3GIP0
IPC42	894h	—	PEVTCIP2	PEVTCIP1	PEVTCIP0	—	PEVTBIP2	PEVTBIP1	PEVTBIP0	—	PEVTAIP2	PEVTAIP1	PEVTAIP0	—	—	—	—
IPC43	896h	—	CLC3PI2	CLC3PI1	CLC3PI0	—	PEVTFIP2	PEVTFIP1	PEVTFIP0	—	PEVTEIP2	PEVTEIP1	PEVTEIP0	—	PEVTDIP2	PEVTDIP1	PEVTDIP0
IPC44	898h	—	CLC3NIP2	CLC3NIP1	CLC3NIP0	—	CLC2NIP2	CLC2NIP1	CLC2NIP0	—	CLC1NIP2	CLC1NIP1	CLC1NIP0	—	CLC4IP2	CLC4IP1	CLC4IP0
IPC45	89Ah	—	—	—	—	—	—	—	—	—	—	—	—	—	CLC4NIP2	CLC4NIP1	CLC4NIP0
IPC47	89Eh	—	U3EVTIP2	U3EVTIP1	U3EVTIP0	—	U2EVTIP2	U2EVTIP1	U2EVTIP0	—	U1EVTIP2	U1EVTIP1	U1EVTIP0	—	—	—	—

图注: — = 未实现。

## 7.3 中断资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 7.3.1 主要资源

- 《dsPIC33/PIC24系列参考手册》中的“中断” ([www.microchip.com/DS70000600](http://www.microchip.com/DS70000600))
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24系列参考手册》章节
- 开发工具

## 7.4 中断控制和状态寄存器

dsPIC33CK64MP105系列器件实现了以下用于中断控制器的寄存器：

- INTCON1
- INTCON2
- INTCON3
- INTCON4
- INTTREG

### 7.4.1 INTCON1至INTCON4

INTCON1、INTCON2、INTCON3和INTCON4控制全局中断控制功能。

INTCON1包含中断嵌套禁止位（NSTDIS）以及处理器陷阱源的控制和状态标志。

INTCON2寄存器控制外部中断请求信号行为，包含全局中断允许（Global Interrupt Enable, GIE）位和备用中断向量表使能（Alternate Interrupt Vector Table Enable, AIVTEN）位。

INTCON3包含附属PLL和DO堆栈溢出状态陷阱源的状态标志。

INTCON4寄存器包含软件生成的硬陷阱（Software Generated Hard Trap, SGHT）状态位。

### 7.4.2 IFSx

IFSx寄存器维护所有中断请求标志。每个中断源都具有一个中断标志状态位，该状态位由相应的外设中断或外部中断信号置1，通过软件进行清零。

### 7.4.3 IECx

IECx寄存器维护所有中断允许位。这些控制位用于单独允许外设中断或外部中断信号。

### 7.4.4 IPCx

IPCx寄存器用于设置每个中断源的中断优先级（IPL）。可以为每个用户中断源分配7个优先级之一。

### 7.4.5 INTTREG

INTTREG寄存器包含相关的中断向量编号和新的CPU中断优先级，分别锁存在INTTREG寄存器中的向量编号（VECNUM[7:0]）和中断优先级（ILR[3:0]）位域中。新的中断优先级是等待处理中断的优先级。

中断源按表7-2中的顺序分配给IFSx、IECx和IPCx寄存器。例如，INT0（外部中断0）向量编号为8，自然顺序优先级为0。所以，INT0IF位在IFS0[0]中，INT0IE位在IEC0[0]中，INT0IP[2:0]位在IPC0的第一个位置（IPC0[2:0]）中。

### 7.4.6 状态/控制寄存器

尽管这些寄存器不是中断控制硬件的特定组成部分，但其中两个CPU控制寄存器包含控制中断功能的位。关于这些寄存器的详细信息，请参见《dsPIC33/PIC24系列参考手册》中的“增强型CPU”（[www.microchip.com/DS70005158](http://www.microchip.com/DS70005158)）。

- CPU状态寄存器SR包含IPL[2:0]位（SR[7:5]）。这些位指示当前CPU中断优先级。用户软件可以通过写IPLx位来更改当前CPU中断优先级。
- CORCON寄存器包含IPL3位，这个位与IPL[2:0]位一起指示当前CPU优先级。IPL3是只读位，所以用户软件不能屏蔽陷阱事件。

下面各页中的寄存器7-3至寄存器7-7介绍了所有中断寄存器。

# dsPIC33CK64MP105系列

寄存器 7-1:        **SR: CPU状态寄存器<sup>(1)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/C-0	R/C-0	R-0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8
R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R/W-0 <sup>(3)</sup>	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2)</sup>	IPL1 <sup>(2)</sup>	IPL0 <sup>(2)</sup>	RA	N	OV	Z	C
bit 7							bit 0

图注:	C = 可清零位
R = 可读位	W = 可写位
-n = POR时的值	1 = 置1
	U = 未实现位, 读为0
	0 = 清零
	x = 未知

bit 7-5        **IPL[2:0]: CPU中断优先级状态位<sup>(2,3)</sup>**

111 = CPU中断优先级为7（15）；禁止用户中断

110 = CPU中断优先级为6（14）

101 = CPU中断优先级为5（13）

100 = CPU中断优先级为4（12）

011 = CPU中断优先级为3（11）

010 = CPU中断优先级为2（10）

001 = CPU中断优先级为1（9）

000 = CPU中断优先级为0（8）

- 注 1: 如需了解整个寄存器的详细信息，请参见[寄存器 3-1](#)。
- 2: IPL[2:0]位与IPL[3]位（CORCON[3]）组合形成CPU中断优先级。如果IPL[3] = 1，那么括号中的值表示IPL。当IPL[3] = 1时，禁止用户中断。
- 3: 当NSTDIS位（INTCON1[15]）= 1时，IPL[2:0]状态位是只读的。

寄存器 7-2: **CORCON: 内核控制寄存器**<sup>(1)</sup>

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
VAR	—	US1	US0	EDT	DL2	DL1	DL0
bit 15							bit 8
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R-0	R/W-0	R/W-0
SATA	SATB	SATDW	ACCSAT	IPL3 <sup>(2)</sup>	SFA	RND	IF
bit 7							bit 0

图注: C = 可清零位  
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 15 **VAR:** 可变异常处理延时控制位  
1 = 使能可变异常处理延时  
0 = 使能固定异常处理延时  
bit 3 **IPL3:** CPU中断优先级状态位<sup>3(2)</sup>  
1 = CPU中断优先级大于7  
0 = CPU中断优先级小于或等于7

注 1: 如需了解整个寄存器的详细信息, 请参见[寄存器 3-2](#)。  
2: IPL3位与IPL[2:0]位 (SR[7:5]) 组合形成CPU中断优先级。

# dsPIC33CK64MP105系列

## 寄存器 7-3: INTCON1: 中断控制寄存器 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
NSTDIS	OVAERR	OVBERR	COVAERR	COVBERR	OVATE	OVBTE	COVTE
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	U-0
SFTACERR	DIV0ERR	DMACERR	MATHERR	ADDRERR	STKERR	OSCFAIL	—
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 15 **NSTDIS:** 中断嵌套禁止位  
1 = 禁止中断嵌套  
0 = 使能中断嵌套
- bit 14 **OVAERR:** 累加器A溢出陷阱标志位  
1 = 陷阱由累加器A溢出引起  
0 = 陷阱不是由累加器A溢出引起
- bit 13 **OVBERR:** 累加器B溢出陷阱标志位  
1 = 陷阱由累加器B溢出引起  
0 = 陷阱不是由累加器B溢出引起
- bit 12 **COVAERR:** 累加器A灾难性溢出陷阱标志位  
1 = 陷阱由累加器A灾难性溢出引起  
0 = 陷阱不是由累加器A灾难性溢出引起
- bit 11 **COVBERR:** 累加器B灾难性溢出陷阱标志位  
1 = 陷阱由累加器B灾难性溢出引起  
0 = 陷阱不是由累加器B灾难性溢出引起
- bit 10 **OVATE:** 累加器A溢出陷阱允许位  
1 = 允许累加器A溢出陷阱  
0 = 禁止陷阱
- bit 9 **OVBTE:** 累加器B溢出陷阱允许位  
1 = 允许累加器B溢出陷阱  
0 = 禁止陷阱
- bit 8 **COVTE:** 灾难性溢出陷阱允许位  
1 = 允许累加器A或B的灾难性溢出陷阱  
0 = 禁止陷阱
- bit 7 **SFTACERR:** 累加器移位错误状态位  
1 = 数学错误陷阱由非法累加器移位引起  
0 = 数学错误陷阱不是由非法累加器移位引起
- bit 6 **DIV0ERR:** 被零除错误状态位  
1 = 数学错误陷阱由被零除引起  
0 = 数学错误陷阱不是由被零除引起
- bit 5 **DMACERR:** DMA控制器陷阱状态位  
1 = 发生了DMAC错误陷阱  
0 = 未发生DMAC错误陷阱
- bit 4 **MATHERR:** 数学错误状态位  
1 = 发生了数学错误陷阱  
0 = 未发生数学错误陷阱

## 寄存器 7-3:       INTCON1: 中断控制寄存器1 (续)

bit 3	<b>ADDRERR:</b> 地址错误陷阱状态位 1 = 发生了地址错误陷阱 0 = 未发生地址错误陷阱
bit 2	<b>STKERR:</b> 堆栈错误陷阱状态位 1 = 发生了堆栈错误陷阱 0 = 未发生堆栈错误陷阱
bit 1	<b>OSCFAIL:</b> 振荡器故障陷阱状态位 1 = 发生了振荡器故障陷阱 0 = 未发生振荡器故障陷阱
bit 0	<b>未实现:</b> 读为 0

# dsPIC33CK64MP105系列

寄存器 7-4: INTCON2: 中断控制寄存器2

R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
GIE	DISI	SWTRAP	—	—	—	—	AIVTEN
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	INT3EP	INT2EP	INT1EP	INT0EP
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 15      **GIE:** 全局中断允许位  
1 = 允许中断, 将相关的IE位置1  
0 = 禁止中断, 但仍然允许陷阱
- bit 14      **DISI:** DISI 指令状态位  
1 = DISI 指令有效  
0 = DISI 指令无效
- bit 13      **SWTRAP:** 软件陷阱状态位  
1 = 允许软件陷阱  
0 = 禁止软件陷阱
- bit 12-9    **未实现:** 读为0
- bit 8       **AIVTEN:** 备用中断向量表使能位  
1 = 使用备用中断向量表  
0 = 使用标准中断向量表
- bit 7-4     **未实现:** 读为0
- bit 3       **INT3EP:** 外部中断3边沿检测极性选择位  
1 = 负边沿中断  
0 = 正边沿中断
- bit 2       **INT2EP:** 外部中断2边沿检测极性选择位  
1 = 负边沿中断  
0 = 正边沿中断
- bit 1       **INT1EP:** 外部中断1边沿检测极性选择位  
1 = 负边沿中断  
0 = 正边沿中断
- bit 0       **INT0EP:** 外部中断0边沿检测极性选择位  
1 = 负边沿中断  
0 = 正边沿中断



**寄存器 7-5: INTCON3: 中断控制寄存器3**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	NAE
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
—	—	DAE	DOOVR	—	—	—	APLL
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 15-9 **未实现:** 读为0

bit 8 **NAE:** NVM地址错误软陷阱状态位

1 = 发生了NVM地址错误软陷阱

0 = 未发生NVM地址错误软陷阱

bit 7-6 **未实现:** 读为0

bit 5 **DAE:** DMA地址错误软陷阱状态位

1 = 发生了DMA地址错误软陷阱

0 = 未发生DMA地址错误软陷阱

bit 4 **DOOVR:** DO堆栈溢出软陷阱状态位

1 = 发生了DO堆栈溢出软陷阱

0 = 未发生DO堆栈溢出软陷阱

bit 3-1 **未实现:** 读为0

bit 0 **APLL:** 附属PLL失锁软陷阱状态位

1 = 发生了APLL锁定软陷阱

0 = 未发生APLL锁定软陷阱

# dsPIC33CK64MP105系列

寄存器 7-6: INTCON4: 中断控制寄存器 4

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	ECCDBE	SGHT
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 15-2 未实现: 读为0

bit 1 **ECCDBE**: ECC 双位错误陷阱位

1 = 发生了ECC 双位错误陷阱

0 = 未发生ECC 双位错误陷阱

bit 0 **SGHT**: 软件生成的硬陷阱状态位

1 = 发生了软件生成的硬陷阱

0 = 未发生软件生成的硬陷阱

## 寄存器 7-7: INTTREG: 中断控制和状态寄存器

U-0	U-0	R-0	U-0	R-0	R-0	R-0	R-0
—	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
VECNUM7	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
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 15-14 **未实现:** 读为0

bit 13 **VHOLD:** 向量编号捕捉使能位

1 = VECNUM[7:0]位读取向量编号编码树的当前值 (即最高优先级待处理中断)  
0 = 向量编号在中断应答时锁存到VECNUM[7:0], 一直保持到下一次中断应答

bit 12 **未实现:** 读为0

bit 11-8 **ILR[3:0]:** 新的CPU中断优先级位

1111 = CPU中断优先级为15

...

0001 = CPU中断优先级为1

0000 = CPU中断优先级为0

bit 7-0 **VECNUM[7:0]:** 待处理中断向量编号位

11111111 = 255, 保留; 不要使用

...

00001001 = 9, T1——Timer1中断

00001000 = 8, INT0——外部中断0

00000111 = 7, 保留; 不要使用

00000110 = 6, 通用软陷阱

00000101 = 5, 保留; 不要使用

00000100 = 4, 数学错误陷阱

00000011 = 3, 堆栈错误陷阱

00000010 = 2, 通用硬陷阱

00000001 = 1, 地址错误陷阱

00000000 = 0, 振荡器故障陷阱

注:

## 8.0 I/O端口

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“带边沿检测的I/O端口”(www.microchip.com/DS70005322)。

**2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。

许多器件引脚由外设和并行I/O端口所共用。所有I/O输入端口都为施密特触发器输入,提高了抗噪声能力。端口寄存器位于SFR中。

以下是I/O端口的一些主要特性:

- 可单独使能/禁止输出引脚的漏极开路
- 可单独使能/禁止输入引脚的弱上拉和下拉
- 监视选择的输入并在检测到引脚电平状态发生变化时产生中断
- 可在休眠和空闲模式下继续工作

## 8.1 并行I/O (PIO) 端口

所有端口引脚都有12个寄存器与其作为数字I/O时的操作直接相关。数据方向寄存器(TRISx)决定引脚是输入还是输出。如果数据方向位为1,则引脚为输入。

复位后,所有端口引脚均定义为输入。读锁存器(LATx)时,读到的是锁存器中的值;写锁存器时,写入的是锁存器。读端口(PORTx)时,读到的是端口引脚的值;而写端口引脚时,写入的是锁存器。对于特定器件无效的任何位及其相关的数据和控制寄存器都将被禁止。这意味着对应的LATx和TRISx寄存器以及端口引脚都将读为零。

当端口引脚与定义为仅输入的另一个外设或功能共用时,它将被视为专用端口,因为没有任何其他的竞争输出源。表8-1列出了引脚的可用性。表8-2列出了该系列器件上的5V输入耐压引脚。

# dsPIC33CK64MP105系列

表8-1: 引脚和ANSELx的可用性

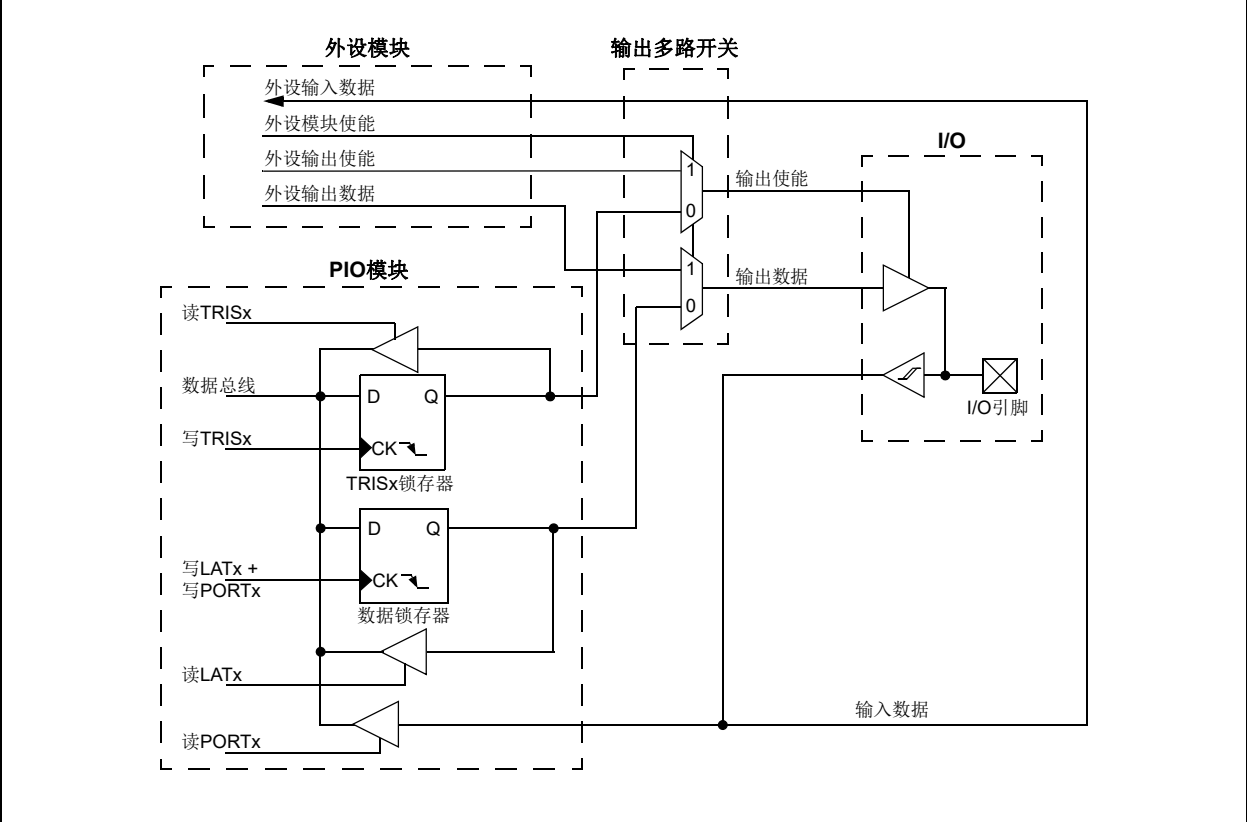
器件	Rx15	Rx14	Rx13	Rx12	Rx11	Rx10	Rx9	Rx8	Rx7	Rx6	Rx5	Rx4	Rx3	Rx2	Rx1	Rx0
<b>PORTA</b>																
dsPIC33CKXXMP105	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
dsPIC33CKXXMP103	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
dsPIC33CKXXMP102	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
ANSELA	—	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X
<b>PORTB</b>																
dsPIC33CKXXMP105	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
dsPIC33CKXXMP103	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
dsPIC33CKXXMP102	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
ANSELB	—	—	—	—	—	—	X	X	X	—	—	X	X	X	X	X
<b>PORTC</b>																
dsPIC33CKXXMP105	—	—	X	X	X	X	X	X	X	X	X	X	X	X	X	X
dsPIC33CKXXMP103	—	—	—	—	—	—	—	—	—	—	X	X	X	X	X	X
dsPIC33CKXXMP102	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ANSEL C	—	—	—	—	—	—	—	—	X	X	—	—	X	X	X	X
<b>PORTD</b>																
dsPIC33CKXXMP105	—	—	X	—	—	X	—	X	—	—	—	—	—	—	X	—
dsPIC33CKXXMP103	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
dsPIC33CKXXMP102	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ANSEL D	—	—	X	—	—	X	—	—	—	—	—	—	—	—	—	—

表8-2: 5V耐压输入端口

PORTA	—	—	—	—	—	—	—	—	—	—	—	RA4	RA3	RA2	RA1	RA0
PORTB	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
PORTC	—	—	RC13	RC12	RC11	RC10	RC9	RC8	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0
PORTD	—	—	RD13	—	—	RD10	—	RD8	—	—	—	—	—	—	RD1	—

图注: 阴影引脚最高可承受5.5 VDC的输入电压。

图8-1: 典型共用端口结构框图



# dsPIC33CK64MP105系列

## 8.1.1 漏极开路配置

除PORTx、LATx和TRISx寄存器用于数据控制外，端口引脚也可被单独地配置为数字输出或漏极开路输出。这是由与每个端口相关的PORTx漏极开路使能寄存器ODCx控制的。将其中的任何位置1即可将相应的引脚配置为漏极开路输出。

这种漏极开路特性允许通过使用外部上拉电阻，产生不同于VDD的输出。任何引脚上允许的最大漏极开路电压与该特定引脚的最大VIH规范相同。

## 8.2 配置模拟和数字端口引脚

ANSELx寄存器用于控制模拟端口引脚的操作。如果要将端口引脚用作模拟输入或输出，则相应的ANSELx和TRISx位必须置1。要将端口引脚用于数字模块（例如，定时器和UART等）的I/O功能，相应的ANSELx位必须清零。

ANSELx寄存器具有默认值0xFFFF；因此，在默认情况下，所有共用模拟功能的引脚都是模拟（而非数字）引脚。

引脚I/O说明（见表1-1）中列出了模拟功能受ANSELx寄存器影响的引脚，其缓冲器类型为模拟。

如果TRISx位清零（输出），而ANSELx位置1，则会通过一个模拟外设（例如，ADC模块或比较器模块）转换数字输出电平（VOH或VOL）。

当读取PORTx寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

配置为数字输入的引脚将不会对模拟输入进行转换。对于任何定义为数字输入的引脚（包括ANx引脚），加在引脚上的模拟电压可能导致输入缓冲器消耗的电流超出器件规范。

## 8.2.1 I/O端口写/读时序

端口方向改变或端口写操作与同一端口的读操作之间需要间隔一个指令周期。通常此指令将是一条NOP指令。

## 8.3 控制寄存器

端口模块中有以下寄存器：

- 寄存器8-1: ANSELx（每端口1个）
- 寄存器8-2: TRISx（每端口1个）
- 寄存器8-3: PORTx（每端口1个）
- 寄存器8-4: LATx（每端口1个）
- 寄存器8-5: ODCx（每端口1个）
- 寄存器8-6: CNPUx（每端口1个）
- 寄存器8-7: CNPDx（每端口1个）
- 寄存器8-8: CNCONx（每端口1个——可选）
- 寄存器8-9: CNEN0x（每端口1个）
- 寄存器8-10: CNSTATx（每端口1个——可选）
- 寄存器8-11: CNEN1x（每端口1个）
- 寄存器8-12: CNFxx（每端口1个）

寄存器8-1: ANSELx: PORTx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
ANSELx[15:8]							
bit 15				bit 8			

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSELx[7:0]							
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 15-0 ANSELx[15:0]: PORTx模拟选择位  
1 = 在PORTx[n]引脚上使能模拟输入，禁止数字输入  
0 = 在PORTx[n]引脚上禁止模拟输入，使能数字输入



## 寄存器8-2: TRISx: PORTx输出使能寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx[15:8]							
bit 15				bit 8			

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TRISx[7:0]							
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 15-0

**TRISx[15:0]:** PORTx输出使能位

1 = 不在PORTx[n]引脚上输出LATx[n]

0 = 在PORTx[n]引脚上输出LATx[n]

## 寄存器8-3: PORTx: PORTx输入数据寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx[15:8]							
bit 15				bit 8			

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PORTx[7:0]							
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 15-0

**PORTx[15:0]:** PORTx数据输入值位

# dsPIC33CK64MP105系列

## 寄存器8-4: LATx: PORTx输出数据寄存器

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx[15:8]							
bit 15				bit 8			

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
LATx[7:0]							
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 15-0 LATx[15:0]: PORTx数据输出值位

## 寄存器8-5: ODCx: PORTx漏极开路使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ODCx[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
ODCx[7:0]							
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 15-0 ODCx[15:0]: PORTx漏极开路使能位

1 = 在PORTx引脚上使能漏极开路

0 = 在PORTx引脚上禁止漏极开路

## 寄存器8-6: CNPUx: PORTx电平变化通知上拉使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPUx[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
CNPUx[7:0]							
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 15-0                  **CNPUx[15:0]:** PORTx电平变化通知上拉使能位  
1 = 使能PORTx[n]的上拉——优先于下拉选择  
0 = 禁止PORTx[n]的上拉

## 寄存器8-7: CNPDx: PORTx电平变化通知下拉使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNPDx[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
CNPDx[7:0]							
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 15-0                  **CNPDx[15:0]:** PORTx电平变化通知下拉使能位  
1 = 使能PORTx[n]的下拉 (未使能PORTx[n]的上拉时)  
0 = 禁止PORTx[n]的下拉

# dsPIC33CK64MP105系列

**寄存器8-8: CNCONx: PORTx电平变化通知控制寄存器**

R/W-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
ON	—	—	—	CNSTYLE	—	—	—
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7				bit 0			

**图注:**

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

- bit 15            **ON:** PORTx电平变化通知 (Change Notification, CN) 控制使能位  
                   1 = 使能 CN  
                   0 = 禁止 CN
- bit 14-12        **未实现:** 读为0
- bit 11            **CNSTYLE:** 电平变化通知类型选择位  
                   1 = 边沿方式 (检测边沿跳变, CNFx[15:0]位用于指示电平变化通知事件)  
                   0 = 不匹配方式 (检测相对于上次端口读操作的变化, CNSTATx[15:0]位用于指示电平变化通知事件)
- bit 10-0        **未实现:** 读为0

**寄存器8-9: CNEN0x: PORTx电平变化通知中断允许寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN0x[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
CNEN0x[7:0]							
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 15-0        **CNEN0x[15:0]:** PORTx电平变化通知中断允许位  
                   1 = 允许PORTx[n]的电平变化 (相对于上次读取的值) 中断  
                   0 = 禁止PORTx[n]的电平变化中断

## 寄存器8-10: CNSTATx: PORTx电平变化通知中断状态寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CNSTATx[15:8]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CNSTATx[7:0]							
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 15-0 **CNSTATx[15:0]:** PORTx电平变化通知中断状态位

当CNSTYLE (CNCONx[11]) = 0时:

1 = 自上次读取PORTx[n]以来PORTx[n]发生了变化

0 = 自上次读取PORTx[n]以来PORTx[n]未发生变化

## 寄存器8-11: CNEN1x: PORTx电平变化通知中断边沿选择寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNEN1x[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
CNEN1x[7:0]							
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 15-0 **CNEN1x[15:0]:** PORTx电平变化通知中断边沿选择位

# dsPIC33CK64MP105系列

寄存器8-12: CNF<sub>x</sub>: PORT<sub>x</sub>电平变化通知中断标志寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CNF <sub>x</sub> [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
CNF <sub>x</sub> [7:0]							
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 15 CNF<sub>x</sub>[15:0]: PORT<sub>x</sub>电平变化通知中断标志位

当CNSTYLE (CNCON<sub>x</sub>[11]) = 1时:

1 = 在PORT<sub>x</sub>[n]引脚上发生了使能的边沿事件。

0 = 在PORT<sub>x</sub>[n]引脚上未发生使能的边沿事件。

## 8.4 输入电平变化通知 (ICN)

I/O端口的输入电平变化通知功能允许dsPIC33CK64MP105系列器件在所选输入引脚的状态变化(Change-Of-State, COS)时,向处理器发出中断请求。该特性可检测输入状态变化,即使在休眠模式下、禁止时钟时。每个I/O端口引脚都可以选择(使能)为在发生状态变化时产生中断请求。5个控制寄存器与每个I/O端口的电平变化通知(CN)功能相关。要启用端口的电平变化通知功能,必须将ON位(CNCONx[15])置1。

CNEN0x和CNEN1x寄存器包含每个输入引脚的CN中断允许控制位。将这些位置1将允许相应引脚的CN中断。此外,这些位与CNSTYLE位(CNCONx[11])相结合,可定义产生中断时的电平跳变类型。表8-3列出了可能的CN事件选项。

表8-3: 电平变化通知事件选项

CNSTYLE位 (CNCONx[11])	CNEN1x位	CNEN0x位	电平变化通知事件说明
0	无关	0	禁止
0	无关	1	检测上次读取的引脚状态与当前的引脚状态是否不匹配
1	0	0	禁止
1	0	1	仅检测正跳变(从0到1)
1	1	0	仅检测负跳变(从1到0)
1	1	1	同时检测正跳变和负跳变

CNSTATx寄存器指示自上次读取PORTx位以来对应引脚上的电平是否发生了变化。除了CNSTATx寄存器之外,还为每个端口实现了CNFxF寄存器。此寄存器包含电平变化通知事件的标志。如果检测到在CNEN0x和CNEN1x寄存器中选择的有效跳变边沿,则这些标志将置1。CNFxF存储是否发生电平变化中断事件的标志。必须用软件清零CNFxF中的位,才能获得下一个电平变化通知中断。仅为配置为输入的I/O产生CN中断(相应的TRISx位必须置1)。

**注:** 当端口引脚被配置为数字输出时,输入电平变化通知引脚上的上拉和下拉应始终被禁止。

## 8.5 外设引脚选择 (PPS)

通用器件的主要挑战是提供尽可能多的外设功能,同时将其与I/O引脚功能的冲突降到最小。在低引脚数器件上,这一挑战更为严峻。在需要多个外设复用同一个引脚的应用中,要在应用程序代码中进行变通比较困难,彻底重新设计可能是惟一的选择。

外设引脚选择配置提供了这些选择的替代方法,使得用户可以从大量的I/O引脚中选择用于外设功能的引脚。通过增加特定器件上可用的引脚分配选项,用户可以让器件更适合他们的整个应用,而不必通过修改应用来适应器件。

外设引脚选择配置功能对固定的一部分数字I/O引脚进行操作。用户可以将大多数数字外设的输入和/或输出独立地映射到这些I/O引脚中的任何一个。一旦建立外设映射,就同时包含了硬件保护,以防止对外设映射的意外或错误更改。

### 8.5.1 可用的引脚

可用引脚的数目取决于特定器件及其引脚数。支持外设引脚选择功能的引脚在它们的引脚全称中包含标记“RPn”,其中“n”是可重映射的引脚编号。“RP”用于表示支持可重映射输入和输出功能的引脚。

### 8.5.2 可用的外设

外设引脚选择管理的外设都是仅数字功能的外设。这些外设包括一般串行通信(UART和SPI)、通用定时器时钟输入、与定时器相关的外设(输入捕捉和输出比较)以及电平变化中断输入。

相比较而言,一些仅数字功能的外设模块不能使用外设引脚选择功能。这是因为此类外设功能需要在特定端口上具有特殊I/O电路,不容易连接到多个引脚。其中一个示例是I<sup>2</sup>C模块。类似的要求排除了所有带模拟输入的模块,例如A/D转换器(ADC)。

可重映射和不可重映射外设之间的主要差异在于可重映射外设与默认的I/O引脚无关。在使用外设前必须始终将外设分配给特定的I/O引脚。相反,不可重映射外设始终可使用默认引脚(假设该外设有效且与其他外设没有冲突)。

# dsPIC33CK64MP105系列

当某个可重映射外设给定I/O引脚上有效时，它的优先级高于与该引脚相关的所有其他数字I/O以及数字通信外设。优先级与被映射外设的类型无关。可重映射外设的优先级不会高于与该引脚相关的任何模拟功能。

### 8.5.3 控制配置更改

由于可以在运行时更改外设映射，因此需要对外设重映射进行一些限制以防止意外更改配置。dsPIC33CK64MP105器件实现了控制寄存器锁定序列。

复位后，允许写入RPINRx和RPORx寄存器，但可以通过将IOLOCK位（RPCON[11]）置1来禁止这些写入操作。IOLOCK位置1时尝试的写入操作看似正常执行，但实际上寄存器的内容保持不变。将IOLOCK置1可阻止写入控制寄存器；将IOLOCK清零则允许写操作。要将IOLOCK置1或清零，必须执行NVMKEY解锁序列：

- 1. 将0x55写入NVMKEY。
- 2. 将0xAA写入NVMKEY。
- 3. 执行单个操作，将IOLOCK清零（或置1）。

**注：** XC16编译器提供了一个用于解锁和修改RPCON寄存器的内置C语言函数：  
\_\_builtin\_write\_RPCON(value);  
更多信息，请参见XC16编译器帮助文件。

### 8.5.4 输入映射

外设引脚选择选项的输入在外设基础上进行映射。即，与外设相关的控制寄存器指示要映射到的引脚。RPINRx寄存器用于配置外设输入映射。每个寄存器包含8位域组，每组都与可重映射外设之一相关。使用适当的8位索引值编程给定外设的位域，会将具有对应值的RPN引脚或内部信号映射到该外设。请参见表8-4了解可用输入的列表。

例如，图8-2显示了U1RX输入的可重映射引脚选择。

图8-2: U1RX的可重映射输入

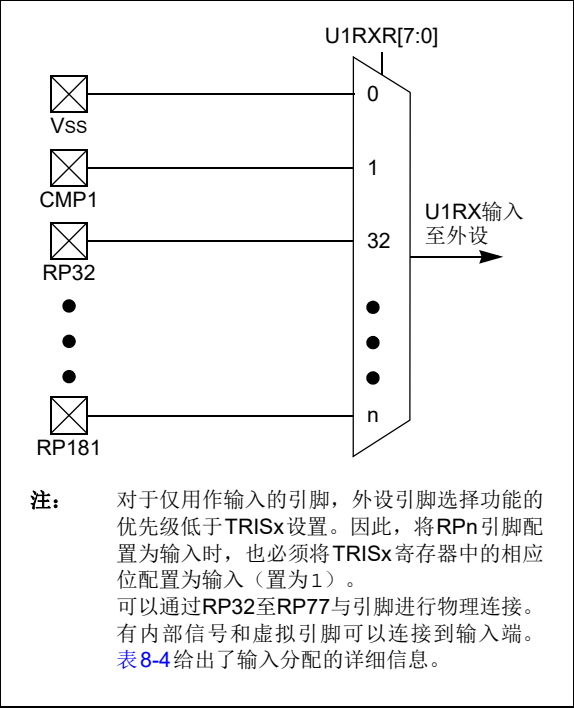




表8-4: 可重映射引脚输入

RPINRx[15:8]或 RPINRx[7:0]	功能	位置
0	Vss	内部
1	比较器 1	内部
2	比较器 2	内部
3	比较器 3	内部
4-5	RP4-RP5	保留
6	PTG 触发信号 26	内部
7	PTG 触发信号 27	内部
8-10	RP8-RP10	保留
11	PWM 事件输出 C	内部
12	PWM 事件输出 D	内部
13	PWM 事件输出 E	内部
14-31	RP14-RP31	保留
32	RP32	端口引脚 RB0
33	RP33	端口引脚 RB1
34	RP34	端口引脚 RB2
35	RP35	端口引脚 RB3
36	RP36	端口引脚 RB4
37	RP37	端口引脚 RB5
38	RP38	端口引脚 RB6
39	RP39	端口引脚 RB7
40	RP40	端口引脚 RB8
41	RP41	端口引脚 RB9
42	RP42	端口引脚 RB10
43	RP43	端口引脚 RB11
44	RP44	端口引脚 RB12
45	RP45	端口引脚 RB13
46	RP46	端口引脚 RB14
47	RP47	端口引脚 RB15
48	RP48	端口引脚 RC0
49	RP49	端口引脚 RC1
50	RP50	端口引脚 RC2
51	RP51	端口引脚 RC3
52	RP52	端口引脚 RC4
53	RP53	端口引脚 RC5
54	RP54	端口引脚 RC6
55	RP55	端口引脚 RC7
56	RP56	端口引脚 RC8
57	RP57	端口引脚 RC9
58	RP58	端口引脚 RC10
59	RP59	端口引脚 RC11
60	RP60	端口引脚 RC12
61	RP61	端口引脚 RC13

# dsPIC33CK64MP105系列

表8-4: 可重映射引脚输入（续）

RPINRx[15:8]或 RPINRx[7:0]	功能	位置
62-64	RP62-RP64	保留
65	RP65	端口引脚RD1
66-71	RP66-RP71	保留
72	RP72	端口引脚RD8
73	RP73	保留
74	RP74	端口引脚RD10
75-76	RP75-RP76	保留
77	RP77	端口引脚RD13
78-175	RP78-RP175	保留
176	RP176	虚拟RPV0
177	RP177	虚拟RPV1
178	RP178	虚拟RPV2
179	RP179	虚拟RPV3
180	RP180	虚拟RPV4
181	RP181	虚拟RPV5

## 8.5.5 虚拟连接

dsPIC33CK64MP105 器件支持6个虚拟RPn引脚（RP176-RP181），它们与所有其他RPn引脚的功能相同，但没有对应的物理位置。这6个引脚位于器件内部，它们不与物理器件引脚连接。

这些引脚提供了一种简单方式来进行外设间的连接，而无需使用物理引脚。例如，模拟比较器的输出可以连接到RP176，同时可以将PWM故障输入配置到RP176。该配置使模拟比较器无需使用器件上的实际物理引脚就可以触发PWM故障。

表8-5: 可选的输入源（将输入映射到功能）

输入名称 <sup>(1)</sup>	功能名称	寄存器	寄存器位
外部中断1	INT1	RPINR0	INT1R[7:0]
外部中断2	INT2	RPINR1	INT2R[7:0]
外部中断3	INT3	RPINR1	INT3R[7:0]
Timer1 外部时钟	T1CK	RPINR2	T1CK[7:0]
SCCP Timer1	TCKI1	RPINR3	TCKI1R[7:0]
SCCP 捕捉1	ICM1	RPINR3	ICM1R[7:0]
SCCP Timer2	TCKI2	RPINR4	TCKI2R[7:0]
SCCP 捕捉2	ICM2	RPINR4	ICM2R[7:0]
SCCP Timer3	TCKI3	RPINR5	TCKI3R[7:0]
SCCP 捕捉3	ICM3	RPINR5	ICM3R[7:0]
SCCP Timer4	TCKI4	RPINR6	TCKI4R[7:0]
SCCP 捕捉4	ICM4	RPINR6	ICM4R[7:0]
MCCP Timer5	TCKI5	RPINR7	TCKI5R[7:0]
MCCP 捕捉5	ICM5	RPINR7	ICM5R[7:0]
xCCP 故障A	OCFA	RPINR11	OCFAR[7:0]
xCCP 故障B	OCFB	RPINR11	OCFBR[7:0]
PWM PCI输入8	PCI8	RPINR12	PCI8R[7:0]
PWM PCI输入9	PCI9	RPINR12	PCI9R[7:0]
PWM PCI输入10	PCI10	RPINR13	PCI10R[7:0]
PWM PCI输入11	PCI11	RPINR13	PCI11R[7:0]
QE1输入A	QEIA1	RPINR14	QEIA1R[7:0]
QE1输入B	QEIB1	RPINR14	QEIB1R[7:0]
QE1索引1输入	QEINDX1	RPINR15	QEINDX1R[7:0]
QE1归位1输入	QEIHOM1	RPINR15	QEIHOM1R[7:0]
QE2输入A	QEIA2	RPINR16	QEIA2R[7:0]
QE2输入B	QEIB2	RPINR16	QEIB2R[7:0]
QE2索引1输入	QEINDX2	RPINR17	QEINDX2R[7:0]
QE2归位1输入	QEIHOM2	RPINR17	QEIHOM2R[7:0]
UART1接收	U1RX	RPINR18	U1RXR[7:0]
UART1数据设备就绪	U1DSR	RPINR18	U1DSRR[7:0]
UART2接收	U2RX	RPINR19	U2RXR[7:0]
UART2数据设备就绪	U2DSR	RPINR19	U2DSRR[7:0]
SPI1数据输入	SDI1	RPINR20	SDI1R[7:0]
SPI1时钟输入	SCK1IN	RPINR20	SCK1R[7:0]
SPI1从选择	SS1	RPINR21	SS1R[7:0]
参考时钟输入	REFCLKI	RPINR21	REFOIR[7:0]
SPI2数据输入	SDI2	RPINR22	SDI2R[7:0]
SPI2时钟输入	SCK2IN	RPINR22	SCK2R[7:0]
SPI2从选择	SS2	RPINR23	SS2R[7:0]
UART3接收	U3RX	RPINR27	U3RXR[7:0]
UART3数据设备就绪	U3DSR	RPINR27	U3DSRR[7:0]

注 1: 除非另外声明, 否则所有输入均使用施密特触发器输入缓冲器。

# dsPIC33CK64MP105系列

表8-5: 可选择的输入源（将输入映射到功能）（续）

输入名称 <sup>(1)</sup>	功能名称	寄存器	寄存器位
SPI3 数据输入	SDI3	RPINR29	SDI3R[7:0]
SPI3 时钟输入	SCK3IN	RPINR29	SCK3R[7:0]
SPI3 从选择	SS3	RPINR30	SS3R[7:0]
xCCP 故障C	OCFC	RPINR37	OCFCR[7:0]
PWM PCI输入 17	PCI17	RPINR37	PCI17R[7:0]
PWM PCI输入 18	PCI18	RPINR38	PCI18R[7:0]
PWM PCI输入 12	PCI12	RPINR42	PCI12R[7:0]
PWM PCI输入 13	PCI13	RPINR42	PCI13R[7:0]
PWM PCI输入 14	PCI14	RPINR43	PCI14R[7:0]
PWM PCI输入 15	PCI15	RPINR43	PCI15R[7:0]
PWM PCI输入 16	PCI16	RPINR44	PCI16R[7:0]
SENT1 输入	SENT1	RPINR44	SENT1R[7:0]
SENT2 输入	SENT2	RPINR45	SENT2R[7:0]
CLC 输入 A	CLCINA	RPINR45	CLCINAR[7:0]
CLC 输入 B	CLCINB	RPINR46	CLCINBR[7:0]
CLC 输入 C	CLCINC	RPINR46	CLCINCR[7:0]
CLC 输入 D	CLCIND	RPINR47	CLCINDR[7:0]
ADC 触发信号输入（ADTRIG31）	ADCTRG	RPINR47	ADCTRGR[7:0]
xCCP 故障D	OCFD	RPINR48	OCFDR[7:0]
UART1 允许发送	$\overline{U1CTS}$	RPINR48	U1CTSR[7:0]
UART2 允许发送	$\overline{U2CTS}$	RPINR49	U2CTSR[7:0]
UART3 允许发送	$\overline{U3CTS}$	RPINR49	U3CTSR[7:0]

注 1: 除非另外声明，否则所有输入均使用施密特触发器输入缓冲器。

### 8.5.6 输出映射

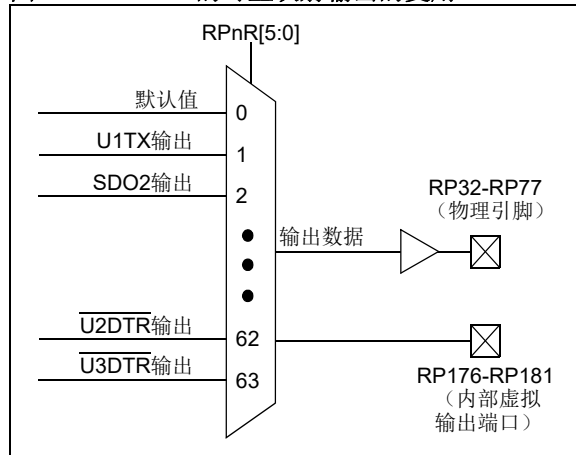
与输入不同，外设引脚选择选项的输出在引脚基础上进行映射。在这种情况下，与特定引脚相关的控制寄存器指示要被映射的外设输出。**RPORx** 寄存器用于控制输出映射。每个寄存器包含6位位域组，每组都与一个 **RPn** 引脚相关（见[寄存器8-48至寄存器8-67](#)）。位域的值与外设之一相对应，并且该外设的输出被映射到该引脚（见[表8-7](#)和[图8-3](#)）。

空输出与输出寄存器复位值0关联。这样做可确保在默认情况下，可重映射输出保持与所有输出引脚之间的断开状态。

### 8.5.7 映射限制

外设选择引脚的控制机制不局限于固定外设置的小范围内。在任何外设映射**SFR**之间没有互锁或硬件强制的锁定。也就是说，可以跨任何或所有 **RPn** 引脚实现外设映射的任何组合，这包括外设输入和输出到引脚的多对一或一对多映射。从配置角度来看，这种映射在技术上是可行的，但从电气角度来看可能不受支持（见[表8-6](#)）。

**图8-3: RPn的可重映射输出的复用**



**注 1:** 有6个虚拟输出端口（**RP176-RP181**）未连接任何I/O端口。这些虚拟端口可通过 **RPOR17**、**RPOR18**和**RPOR19**访问。

# dsPIC33CK64MP105系列

表8-6: 可重映射输出引脚寄存器

寄存器	RP 引脚	I/O 端口
RPOR0[5:0]	RP32	端口引脚 RB0
RPOR0[13:8]	RP33	端口引脚 RB1
RPOR1[5:0]	RP34	端口引脚 RB2
RPOR1[13:8]	RP35	端口引脚 RB3
RPOR2[5:0]	RP36	端口引脚 RB4
RPOR2[13:8]	RP37	端口引脚 RB5
RPOR3[5:0]	RP38	端口引脚 RB6
RPOR3[13:8]	RP39	端口引脚 RB7
RPOR4[5:0]	RP40	端口引脚 RB8
RPOR4[13:8]	RP41	端口引脚 RB9
RPOR5[5:0]	RP42	端口引脚 RB10
RPOR5[13:8]	RP43	端口引脚 RB11
RPOR6[5:0]	RP44	端口引脚 RB12
RPOR6[13:8]	RP45	端口引脚 RB13
RPOR7[5:0]	RP46	端口引脚 RB14
RPOR7[13:8]	RP47	端口引脚 RB15
RPOR8[5:0]	RP48	端口引脚 RC0
RPOR8[13:8]	RP49	端口引脚 RC1
RPOR9[5:0]	RP50	端口引脚 RC2
RPOR9[13:8]	RP51	端口引脚 RC3
RPOR10[5:0]	RP52	端口引脚 RC4
RPOR10[13:8]	RP53	端口引脚 RC5
RPOR11[5:0]	RP54	端口引脚 RC6
RPOR11[13:8]	RP55	端口引脚 RC7
RPOR12[5:0]	RP56	端口引脚 RC8
RPOR12[13:8]	RP57	端口引脚 RC9
RPOR13[5:0]	RP58	端口引脚 RC10
RPOR13[13:8]	RP59	端口引脚 RC11
RPOR14[5:0]	RP60	端口引脚 RC12
RPOR14[13:8]	RP61	端口引脚 RC13
RPOR15[5:0]	RP65	端口引脚 RD1
RPOR15[13:8]	RP72	端口引脚 RD8
RPOR16[5:0]	RP74	端口引脚 D10
RPOR16[13:8]	RP77	端口引脚 RD13
RPOR17[5:0]	RP176	虚拟引脚 RPV0
RPOR17[13:8]	RP177	虚拟引脚 RPV1
RPOR18[5:0]	RP178	虚拟引脚 RPV2
RPOR18[13:8]	RP179	虚拟引脚 RPV3
RPOR19[5:0]	RP180	虚拟引脚 RPV4
RPOR19[13:8]	RP181	虚拟引脚 RPV5

表8-7: 可重映射引脚 (RPn) 的输出选择

功能	RPnR[5:0]	输出名称
未连接	0	未连接
U1TX	1	RPn连接到UART1发送
U1RTS	2	RPn连接到UART1请求发送
U2TX	3	RPn连接到UART2发送
U2RTS	4	RPn连接到UART2请求发送
SDO1	5	RPn连接到SPI1数据输出
SCK1	6	RPn连接到SPI1时钟输出
SS1	7	RPn连接到SPI1从选择
SDO2	8	RPn连接到SPI2数据输出
SCK2	9	RPn连接到SPI2时钟输出
SS2	10	RPn连接到SPI2从选择
SDO3	11	RPn连接到SPI3数据输出
SCK3	12	RPn连接到SPI3时钟输出
SS3	13	RPn连接到SPI3从选择
REFCLKO	14	RPn连接到参考时钟输出
OCM1A	15	RPn连接到SCCP1输出
OCM2A	16	RPn连接到SCCP2输出
OCM3A	17	RPn连接到SCCP3输出
OCM4A	18	RPn连接到SCCP4输出
CMP1	23	RPn连接到比较器1输出
CMP2	24	RPn连接到比较器2输出
CMP3	25	RPn连接到比较器3输出
U3TX	27	RPn连接到UART3发送
U3RTS	28	RPn连接到UART3请求发送
PWM4H	34	RPn连接到PWM4H输出
PWM4L	35	RPn连接到PWM4L输出
PWMEA	36	RPn连接到PWM事件A输出
PWMEB	37	RPn连接到PWM事件B输出
QEICMP1	38	RPn连接到QEI1比较器输出
QEICMP2	39	RPn连接到QEI2比较器输出
CLC1OUT	40	RPn连接到CLC1输出
CLC2OUT	41	RPn连接到CLC2输出
PWMEC	44	RPn连接到PWM事件C输出
PWMEC	45	RPn连接到PWM事件D输出
PTGTRG24	46	PTG触发信号输出24
PTGTRG25	47	PTG触发信号输出25
SENT1OUT	48	RPn连接到SENT1输出
SENT2OUT	49	RPn连接到SENT2输出
OCM5A	50	RPn连接到MCCP5输出A
OCM5B	51	RPn连接到MCCP5输出B
OCM5C	52	RPn连接到MCCP5输出C
OCM5D	53	RPn连接到MCCP5输出D

# dsPIC33CK64MP105系列

---

表8-7: 可重映射引脚 (RPn) 的输出选择 (续)

功能	RPnR[5:0]	输出名称
OCM5E	54	RPn连接到MCCP5输出E
OCM5F	55	RPn连接到MCCP5输出F
CLC3OUT	59	RPn连接到CLC4输出
CLC4OUT	60	RPn连接到CLC4输出
U1DTR	61	RPn连接到UART1 DTR
U2DTR	62	RPn连接到UART2 DTR
U3DTR	63	RPn连接到UART3 DTR



## 8.5.8 I/O有用技巧

1. 在某些情况下，如表31-15的“注入电流”下所定义，一些引脚与VDD和VSS之间存在内部保护二极管。术语“注入电流”也称为“钳位电流”。在指定引脚上，在用户采取了足够外部限流措施的情况下，允许I/O引脚输入电压大于或小于数据手册中的绝对最大值（与电源VSS和VDD有关）。请注意，当用户应用对高端或低端内部输入钳位二极管进行正向偏置时，所产生的电流会注入内部钳位到VDD和VSS电源轨的器件，该电流可能影响ADC精度（4至6个计数）。
2. 在发生任何复位之后，与任意模拟输入引脚（即ANx）复用的I/O引脚默认情况下总是模拟引脚。因此，将某个引脚配置为模拟输入引脚时，将会自动禁止数字输入引脚缓冲器，如果尝试通过读取PORTx或LATx来读取数字输入电平，无论引脚上的数字逻辑电平为何，将总是返回0。要将与ANx引脚复用的引脚作为数字I/O引脚，用户应用程序需要配置I/O端口模块中的PORTx模拟选择寄存器（即ANSELx），将对应于该I/O端口的相应位设置为0。

**注：** 虽然在使能模拟功能时无法使用数字输入引脚，但在使能模拟功能的情况下可以使用数字I/O输出功能（TRISx = 0x0）。但建议不要这样做，特别是如果模拟输入连接到外部模拟电压源，这会在模拟信号和输出引脚驱动器之间产生信号争用。

3. 大多数I/O引脚具有多种功能。请参见本数据手册中的器件引脚分配图，分配给任意引脚的功能的优先级按照由从左到右阅读引脚名称的方式来指示。在命名约定中，最左侧功能名称的优先级高于其右侧的所有功能。例如：AN16/T2CK/T7CK/RC1；这表明在该示例中AN16的优先级最高，将优先于列表中其右侧的所有其他功能。即使已使其右侧的那些其他功能，只要其左侧的任何功能已使能，右侧功能就不会起作用。该规则适用于对于给定引脚列出的所有功能。
4. 每个引脚都具有内部弱上拉电阻和弱下拉电阻，可分别使用CNPUx和CNPDx寄存器进行配置。由于具有这些电阻，在一些应用中可以不需要外部电阻。内部上拉可将电压最高拉至 $\sim(V_{DD} - 0.8)$ ，而不是VDD。该值仍然高于CMOS和TTL器件的最小VIH。
5. 直接驱动LED时，I/O引脚的拉电流或灌电流可以高于VOH/IOH和VOL/IOL直流特性规范中规定的值。相应的IOH和IOL电流额定值只是为了使相应输出保持大于等于VOH和小于等于VOL电压。但对于LED，不同于外部连接器件的数字输入，它们不受相同的最小VIH/VIL电压限制。I/O引脚输出可以安全地灌入或拉出小于本数据手册第31.0节“电气特性”中的绝对最大值部分所列值的任何电流。例如：  
 $IOH = -8\text{ mA}$  且  $V_{DD} = 3.3\text{V}$  时  $VOH = 2.4\text{V}$   
 任意8 mA I/O引脚的最大输出拉电流 = 12 mA。  
 从技术上说，允许用于LED的拉电流 < 12 mA。

## 6. 外设引脚选择（PPS）引脚映射规则如下：

- a) 在任意时刻，给定引脚上只能有一个“输出”功能处于活动状态，无论它是专用还是可重映射功能（一个引脚，一个输出）。
- b) 可以将某个“可重映射输出”功能分配给多个引脚，并在外部将它们短接或连接在一起，以提高电流驱动能力。
- c) 如果在某个引脚上使能了任何“专用输出”功能，它将优先于任何可重映射“输出”功能。
- d) 如果在某个引脚上使能了任何“专用数字”（输入或输出）功能，则可以将任意数量的“输入”可重映射功能映射到同一引脚。
- e) 如果在某个给定引脚上使能了任何“专用模拟”功能，则将禁止任何一种“数字输入”，但用户可以审慎地使能单个“数字输出”并使之处于活动状态，前提是它不会与外部模拟输入信号发生信号争用。例如，可以使用ADC转换数字输出逻辑电平，或翻转比较器的数字输出或ADC输入，前提是没有诸如用于内置自检的外部模拟输入。
- f) 可以同时将任意数量的“输入”可重映射功能映射到相同引脚，包括映射到具有来自专用或可重映射“输出”的单个输出的任意引脚。
- g) TRISx寄存器仅控制数字I/O输出缓冲器。任何其他专用或可重映射的活动“输出”将自动改写TRISx设置。TRISx寄存器不控制数字逻辑“输入”缓冲器。可重映射数字“输入”不会自动改写TRISx设置，这意味着对于仅分配有可重映射输入功能的引脚，必须将TRISx位设置为输入。
- h) 发生任意复位之后，默认情况下将使能所有模拟引脚，并且引脚上相应的数字输入缓冲器会被禁止。只有PORTx模拟选择（ANSELx）寄存器会控制数字输入缓冲器，TRISx寄存器不会。为了使用某个引脚上的任何“数字输入”，用户必须使用PORTx模拟选择寄存器禁止相应引脚上的模拟功能，没有例外。

## 8.5.9 I/O端口资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 8.5.9.1 主要资源

- 《dsPIC33/PIC24 系列参考手册》中的“带边沿检测的I/O端口”（[www.microchip.com/DS70005322](http://www.microchip.com/DS70005322)）
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24 系列参考手册》章节
- 开发工具

表8-8: PORTA寄存器汇总

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELA	—	—	—	—	—	—	—	—	—	—	—	ANSELA[4:0]				
TRISA	—	—	—	—	—	—	—	—	—	—	—	TRISA[4:0]				
PORTA	—	—	—	—	—	—	—	—	—	—	—	RA[4:0]				
LATA	—	—	—	—	—	—	—	—	—	—	—	LATA[4:0]				
ODCA	—	—	—	—	—	—	—	—	—	—	—	ODCA[4:0]				
CNPUA	—	—	—	—	—	—	—	—	—	—	—	CNPUA[4:0]				
CNPDA	—	—	—	—	—	—	—	—	—	—	—	CNPDA[4:0]				
CNCONA	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0A	—	—	—	—	—	—	—	—	—	—	—	CNEN0A[4:0]				
CNSTATA	—	—	—	—	—	—	—	—	—	—	—	CNSTATATA[4:0]				
CNEN1A	—	—	—	—	—	—	—	—	—	—	—	CNEN1A[4:0]				
CNFA	—	—	—	—	—	—	—	—	—	—	—	CNFA[4:0]				

表8-9: PORTB寄存器汇总

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELB	—	—	—	—	—	—	ANSELB[9:7]			—	—	ANSELB[4:0]				
TRISB	TRISB[15:0]															
PORTB	RB[15:0]															
LATB	LATB[15:0]															
ODCB	ODCB[15:0]															
CNPUB	CNPUB[15:0]															
CNPDB	CNPDB[15:0]															
CNCONB	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0B	CNEN0[15:0]															
CNSTATB	CNSTATB[15:0]															
CNEN1B	CNEN1B[15:0]															
CNFB	CNFB[15:0]															

表8-10: PORTC寄存器汇总

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELC	—	—	—	—	—	—	—	—	ANSELC[7:6]		—	—	ANSELC[3:0]			
TRISC	—	—	TRISC[13:0]													
PORTC	—	—	RC[13:0]													
LATC	—	—	LATC[13:0]													
ODCC	—	—	ODCC[13:0]													
CNPUC	—	—	CNPUC[13:0]													
CNPDC	—	—	CNPDC[13:0]													
CNCONC	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0C	—	—	CNEN0C[13:0]													
CNSTATC	—	—	CNSTATC[13:0]													
CNEN1C	—	—	CNEN1C[13:0]													
CNFC	—	—	CNFC[13:0]													

表8-11: PORTD寄存器汇总

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ANSELD	—	—	ANSELD13	—	—	ANSELD10	—	—	—	—	—	—	—	—	—	—
TRISD	—	—	TRISD13	—	—	TRISD10	—	TRISD8	—	—	—	—	—	—	TRISD1	—
PORTD	—	—	RD13	—	—	RD10	—	RD8	—	—	—	—	—	—	RD1	—
LATD	—	—	LATD13	—	—	LATD10	—	LATD8	—	—	—	—	—	—	LATD1	—
ODCD	—	—	ODCD13	—	—	ODCD10	—	ODCD8	—	—	—	—	—	—	ODCD1	—
CNPUD	—	—	CNPUD13	—	—	CNPUD10	—	CNPUD8	—	—	—	—	—	—	CNPUD1	—
CNPDD	—	—	CNPDD13	—	—	CNPDD10	—	CNPDD8	—	—	—	—	—	—	CNPDD1	—
CNCOND	ON	—	—	—	CNSTYLE	—	—	—	—	—	—	—	—	—	—	—
CNEN0D	—	—	CNEN0D13	—	—	CNEN0D10	—	CNEN0D8	—	—	—	—	—	—	CNEN0D1	—
CNSTATD	—	—	CNSTATD13	—	—	CNSTATD10	—	CNSTATD8	—	—	—	—	—	—	CNSTATD1	—
CNEN1D	—	—	CNEN1D13	—	—	CNEN1D10	—	CNEN1D8	—	—	—	—	—	—	CNEN1D1	—
CNFD	—	—	CNFD13	—	—	CNFD10	—	CNFD8							CNFD1	—

## 8.5.10 外设引脚选择寄存器

### 寄存器8-13: **RPCON**: 外设重映射配置寄存器<sup>(1)</sup>

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	IOLOCK	—	—	—
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7				bit 0			

#### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-12 **未实现**: 读为0

bit 11 **IOLOCK**: 外设重映射寄存器锁定位  
1 = 所有外设重映射寄存器均锁定, 无法写入  
0 = 所有外设重映射寄存器均未锁定, 可写入

bit 10-0 **未实现**: 读为0

**注 1:** 对该寄存器进行写操作需要解锁序列。

### 寄存器8-14: **RPINR0**: 外设引脚选择输入寄存器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
INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7				bit 0			

#### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-8 **INT1R[7:0]**: 将外部中断1 (INT1) 分配给对应RPN引脚的位  
请参见表8-4。

bit 7-0 **未实现**: 读为0

# dsPIC33CK64MP105系列

## 寄存器8-15: RPINR1: 外设引脚选择输入寄存器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
INT3R7	INT3R6	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
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
INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
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 15-8                      **INT3R[7:0]:** 将外部中断3 (INT3) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0                      **INT2R[7:0]:** 将外部中断2 (INT2) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-16: RPINR2: 外设引脚选择输入寄存器2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
T1CKR7	T1CKR6	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

### 图注:

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

bit 15-8                      **T1CKR[7:0]:** 将Timer1 外部时钟 (T1CK) 分配给对应RPn引脚的位  
请参见表8-4。

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

## 寄存器8-17: **RPINR3: 外设引脚选择输入寄存器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
ICM1R7	ICM1R6	ICM1R5	ICM1R4	ICM1R3	ICM1R2	ICM1R1	ICM1R0
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
TCKI1R7	TCKI1R6	TCKI1R5	TCKI1R4	TCKI1R3	TCKI1R2	TCKI1R1	TCKI1R0
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 15-8 **ICM1R[7:0]:** 将SCCP捕捉1 (ICM1) 输入分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **TCKI1[7:0]:** 将SCCP Timer1 (TCKI1) 输入分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-18: **RPINR4: 外设引脚选择输入寄存器4**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM2R7	ICM2R6	ICM2R5	ICM2R4	ICM2R3	ICM2R2	ICM2R1	ICM2R0
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
TCKI2R7	TCKI2R6	TCKI2R5	TCKI2R4	TCKI2R3	TCKI2R2	TCKI2R1	TCKI2R0
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 15-8 **ICM2R[7:0]:** 将SCCP捕捉2 (ICM2) 输入分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **TCKI2R[7:0]:** 将SCCP Timer2 (TCKI2) 输入分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-19: RPINR5: 外设引脚选择输入寄存器5

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM3R7	ICM3R6	ICM3R5	ICM3R4	ICM3R3	ICM3R2	ICM3R1	ICM3R0
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
TCKI3R7	TCKI3R6	TCKI3R5	TCKI3R4	TCKI3R3	TCKI3R2	TCKI3R1	TCKI3R0
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 15-8 **ICM3R[7:0]:** 将SCCP捕捉3 (ICM3) 输入分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **TCKI3R[7:0]:** 将SCCP Timer3 (TCKI3) 输入分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-20: RPINR6: 外设引脚选择输入寄存器6

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM4R7	ICM4R6	ICM4R5	ICM4R4	ICM4R3	ICM4R2	ICM4R1	ICM4R0
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
TCKI4R7	TCKI4R6	TCKI4R5	TCKI4R4	TCKI4R3	TCKI4R2	TCKI4R1	TCKI4R0
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 15-8 **ICM4R[7:0]:** 将SCCP捕捉4 (ICM4) 输入分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **TCKI4R[7:0]:** 将SCCP Timer4 (TCKI4) 输入分配给对应RPn引脚的位  
请参见表8-4。



## 寄存器8-21: RPINR7: 外设引脚选择输入寄存器7

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICM5R7	ICM5R6	ICM5R5	ICM5R4	ICM5R3	ICM5R2	ICM5R1	ICM5R0
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
TCKI5R7	TCKI5R6	TCKI5R5	TCKI5R4	TCKI5R3	TCKI5R2	TCKI5R1	TCKI5R0
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 15-8 **ICM5R[7:0]:** 将SCCP捕捉5 (ICM5) 输入分配给对应RPn引脚的位

请参见表8-4。

bit 7-0 **TCKI5R[7:0]:** 将SCCP Timer5 (TCKI5) 输入分配给对应RPn引脚的位

请参见表8-4。

## 寄存器8-22: RPINR11: 外设引脚选择输入寄存器11

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OCFBR7	OCFBR6	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
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
OCFAR7	OCFAR6	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
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 15-8 **OCFBR[7:0]:** 将SCCP故障B (OCFB) 输入分配给对应RPn引脚的位

请参见表8-4。

bit 7-0 **OCFAR[7:0]:** 将SCCP故障A (OCFA) 输入分配给对应RPn引脚的位

请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-23: RPINR12: 外设引脚选择输入寄存器12

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI9R7	PCI9R6	PCI9R5	PCI9R4	PCI9R3	PCI9R2	PCI9R1	PCI9R0
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
PCI8R7	PCI8R6	PCI8R5	PCI8R4	PCI8R3	PCI8R2	PCI8R1	PCI8R0
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 15-8 **PCI9R[7:0]:** 将PWM输入9 (PCI9) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **PCI8R[7:0]:** 将PWM输入8 (PCI8) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-24: RPINR13: 外设引脚选择输入寄存器13

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI11R7	PCI11R6	PCI11R5	PCI11R4	PCI11R3	PCI11R2	PCI11R1	PCI11R0
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
PCI10R7	PCI10R6	PCI10R5	PCI10R4	PCI10R3	PCI10R2	PCI10R1	PCI10R0
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 15-8 **PCI11R[7:0]:** 将PWM输入11 (PCI11) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **PCI10R[7:0]:** 将PWM输入10 (PCI10) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-25: RPINR14: 外设引脚选择输入寄存器14

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIB1R7	QEIB1R6	QEIB1R5	QEIB1R4	QEIB1R3	QEIB1R2	QEIB1R1	QEIB1R0
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
QEIA1R7	QEIA1R6	QEIA1R5	QEIA1R4	QEIA1R3	QEIA1R2	QEIA1R1	QEIA1R0
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 15-8 **QEIB1R[7:0]:** 将QE1输入B (QEIB1) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **QEIA1R[7:0]:** 将QE1输入A (QEIA1) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-26: RPINR15: 外设引脚选择输入寄存器15

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIHOM1R7	QEIHOM1R6	QEIHOM1R5	QEIHOM1R4	QEIHOM1R3	QEIHOM1R2	QEIHOM1R1	QEIHOM1R0
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
QEINDX1R7	QEINDX1R6	QEINDX1R5	QEINDX1R4	QEINDX1R3	QEINDX1R2	QEINDX1R1	QEINDX1R0
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 15-8 **QEIHOM1R[7:0]:** 将QE1归位1输入 (QEIHOM1) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **QEINDX1R[7:0]:** 将QE1索引1输入 (QEINDX1) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

**寄存器8-27: RPINR16: 外设引脚选择输入寄存器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
QEIB2R7	QEIB2R6	QEIB2R5	QEIB2R4	QEIB2R3	QEIB2R2	QEIB2R1	QEIB2R0
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
QEIA2R7	QEIA2R6	QEIA2R5	QEIA2R4	QEIA2R3	QEIA2R2	QEIA2R1	QEIA2R0
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 15-8 **QEIB2R[7:0]:** 将QEI2输入B (QEIB2) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **QEIA2R[7:0]:** 将QEI2输入A (QEIA2) 分配给对应RPn引脚的位  
请参见表8-4。

**寄存器8-28: RPINR17: 外设引脚选择输入寄存器17**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIHOM2R7	QEIHOM2R6	QEIHOM2R5	QEIHOM2R4	QEIHOM2R3	QEIHOM2R2	QEIHOM2R1	QEIHOM2R0
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
QEINDX2R7	QEINDX2R6	QEINDX2R5	QEINDX2R4	QEINDX2R3	QEINDX2R2	QEINDX2R1	QEINDX2R0
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 15-8 **QEIHOM2R[7:0]:** 将QEI2归位1输入 (QEIHOM2) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **QEINDX2R[7:0]:** 将QEI2索引1输入 (QEINDX2) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-29: RPINR18: 外设引脚选择输入寄存器18

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1DSRR7	U1DSRR6	U1DSRR5	U1DSRR4	U1DSRR3	U1DSRR2	U1DSRR1	U1DSRR0
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
U1RXR7	U1RXR6	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
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 15-8 **U1DSRR[7:0]:** 将UART1数据设备就绪 ( $\overline{U1DSR}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **U1RXR[7:0]:** 将UART1接收 (U1RX) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-30: RPINR19: 外设引脚选择输入寄存器19

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U2DSRR7	U2DSRR6	U2DSRR5	U2DSRR4	U2DSRR3	U2DSRR2	U2DSRR1	U2DSRR0
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
U2RXR7	U2RXR6	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
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 15-8 **U2DSRR[7:0]:** 将UART2数据设备就绪 ( $\overline{U2DSR}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **U2RXR[7:0]:** 将UART2接收 (U2RX) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-31: RPINR20: 外设引脚选择输入寄存器20

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK1R7	SCK1R6	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
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
SDI1R7	SDI1R6	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
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 15-8      **SCK1R[7:0]:** 将SPI1时钟输入 (SCK1IN) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0      **SDI1R[7:0]:** 将SPI1数据输入 (SDI1) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-32: RPINR21: 外设引脚选择输入寄存器21

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
REFOIR7	REFOIR6	REFOIR5	REFOIR4	REFOIR3	REFOIR2	REFOIR1	REFOIR0
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
SS1R7	SS1R6	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
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 15-8      **REFOIR[7:0]:** 将参考时钟输入 (REFCLKI) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0      **SS1R[7:0]:** 将SPI1从选择 ( $\overline{SS1}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-33: RPINR22: 外设引脚选择输入寄存器22

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK2R7	SCK2R6	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0
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
SDI2R7	SDI2R6	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
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 15-8 **SCK2R[7:0]:** 将SPI2时钟输入 (SCK2IN) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **SDI2R[7:0]:** 将SPI2数据输入 (SDI2) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-34: RPINR23: 外设引脚选择输入寄存器23

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
SS2R7	SS2R6	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
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 15-8 **未实现:** 读为0

bit 7-0 **SS2R[7:0]:** 将SPI2从选择 (SS2) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-35: RPINR27: 外设引脚选择输入寄存器27

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3DSRR7	U3DSRR6	U3DSRR5	U3DSRR4	U3DSRR3	U3DSRR2	U3DSRR1	U3DSRR0
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
U3RXR7	U3RXR6	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
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 15-8 **U3DSRR[7:0]:** 将UART3数据设备就绪 ( $\overline{\text{U3DSR}}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **U3RXR[7:0]:** 将UART3接收 (U3RX) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-36: RPINR29: 外设引脚选择输入寄存器29

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SCK3R7	SCK3R6	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0
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
SDI3R7	SDI3R6	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
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 15-8 **SCK3R[7:0]:** 将SPI3时钟输入 (SCK3IN) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **SDI3R[7:0]:** 将SPI3数据输入 (SDI3) 分配给对应RPn引脚的位  
请参见表8-4。



## 寄存器8-37: RPINR30: 外设引脚选择输入寄存器30

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
SS3R7	SS3R6	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
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 15-8 未实现: 读为0

bit 7-0 **SS3R[7:0]**: 将SPI3从选择 ( $\overline{SS3}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-38: RPINR37: 外设引脚选择输入寄存器37

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI17R7	PCI17R6	PCI17R5	PCI17R4	PCI17R3	PCI17R2	PCI17R1	PCI17R0
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
OCFCR7	OCFCR6	OCFCR5	OCFCR4	OCFCR3	OCFCR2	OCFCR1	OCFCR0
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 15-8 **PCI17R[7:0]**: 将PWM输入17 (PCI17) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **OCFCR[7:0]**: 将xCCP故障C (OCFC) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-39: RPINR38: 外设引脚选择输入寄存器38

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
PCI18R7	PCI18R6	PCI18R5	PCI18R4	PCI18R3	PCI18R2	PCI18R1	PCI18R0
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 15-8 未实现: 读为0

bit 7-0 **PCI18R[7:0]**: 将PWM输入18 (PCI18) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-40: RPINR42: 外设引脚选择输入寄存器42

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI13R7	PCI13R6	PCI13R5	PCI13R4	PCI13R3	PCI13R2	PCI13R1	PCI13R0
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
PCI12R7	PCI12R6	PCI12R5	PCI12R4	PCI12R3	PCI12R2	PCI12R1	PCI12R0
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 15-8 **PCI13R[7:0]**: 将PWM输入13 (PCI13) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **PCI12R[7:0]**: 将PWM输入12 (PCI12) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-41: RPINR43: 外设引脚选择输入寄存器43

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCI15R7	PCI15R6	PCI15R5	PCI15R4	PCI15R3	PCI15R2	PCI15R1	PCI15R0
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
PCI14R7	PCI14R6	PCI14R5	PCI14R4	PCI14R3	PCI14R2	PCI14R1	PCI14R0
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 15-8 **PCI15R[7:0]:** 将PWM输入 15 (PCI15) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **PCI14R[7:0]:** 将PWM输入 14 (PCI14) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-42: RPINR44: 外设引脚选择输入寄存器44

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SENT1R7	SENT1R6	SENT1R5	SENT1R4	SENT1R3	SENT1R2	SENT1R1	SENT1R0
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
PCI16R7	PCI16R6	PCI16R5	PCI16R4	PCI16R3	PCI16R2	PCI16R1	PCI16R0
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 15-8 **SENT1R[7:0]:** 将SENT1输入 (SENT1) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **PCI16R[7:0]:** 将PWM输入 16 (PCI16) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

## 寄存器8-43: RPINR45: 外设引脚选择输入寄存器45

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINAR7	CLCINAR6	CLCINAR5	CLCINAR4	CLCINAR3	CLCINAR2	CLCINAR1	CLCINAR0
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
SENT2R7	SENT2R6	SENT2R5	SENT2R4	SENT2R3	SENT2R2	SENT2R1	SENT2R0
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 15-8 **CLCINAR[7:0]:** 将CLC输入A (CLCINA) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **SENT2R[7:0]:** 将SENT2输入 (SENT2) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-44: RPINR46: 外设引脚选择输入寄存器46

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLCINCR7	CLCINCR6	CLCINCR5	CLCINCR4	CLCINCR3	CLCINCR2	CLCINCR1	CLCINCR0
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
CLCINBR7	CLCINBR6	CLCINBR5	CLCINBR4	CLCINBR3	CLCINBR2	CLCINBR1	CLCINBR0
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 15-8 **CLCINCR[7:0]:** 将CLC输入C (CLCINC) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **CLCINBR[7:0]:** 将CLC输入B (CLCINB) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-45: RPINR47: 外设引脚选择输入寄存器47

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADCTRGR7	ADCTRGR6	ADCTRGR5	ADCTRGR4	ADCTRGR3	ADCTRGR2	ADCTRGR1	ADCTRGR0
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
CLCINDR7	CLCINDR6	CLCINDR5	CLCINDR4	CLCINDR3	CLCINDR2	CLCINDR1	CLCINDR0
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 15-8 **ADCTRGR[7:0]:** 将ADC触发信号输入 (ADCTRG) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **CLCINDR[7:0]:** 将CLC输入D (CLCIND) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-46: RPINR48: 外设引脚选择输入寄存器48

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U1CTSR7	U1CTSR6	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
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
OCFDR7	OCFDR6	OCFDR5	OCFDR4	OCFDR3	OCFDR2	OCFDR1	OCFDR0
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 15-8 **U1CTSR[7:0]:** 将UART1允许发送 ( $\overline{U1CTS}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **OCFDR[7:0]:** 将xCCP故障D (OCFD) 分配给对应RPn引脚的位  
请参见表8-4。

# dsPIC33CK64MP105系列

寄存器8-47: **RPINR49: 外设引脚选择输入寄存器49**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
U3CTSR7	U3CTSR6	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
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
U2CTSR7	U2CTSR6	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
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 15-8 **U3CTSR[7:0]:** 将UART3允许发送 ( $\overline{\text{U3CTS}}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

bit 7-0 **U2CTSR[7:0]:** 将UART2允许发送 ( $\overline{\text{U2CTS}}$ ) 分配给对应RPn引脚的位  
请参见表8-4。

## 寄存器8-48: RPOR0: 外设引脚选择输出寄存器0

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP33R5	RP33R4	RP33R3	RP33R2	RP33R1	RP33R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP32R5	RP32R4	RP32R3	RP32R2	RP32R1	RP32R0
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 15-14 未实现: 读为0

bit 13-8 **RP33R[5:0]**: 将外设输出功能分配给RP33输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP32R[5:0]**: 将外设输出功能分配给RP32输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-49: RPOR1: 外设引脚选择输出寄存器1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP35R5	RP35R4	RP35R3	RP35R2	RP35R1	RP35R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP34R5	RP34R4	RP34R3	RP34R2	RP34R1	RP34R0
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 15-14 未实现: 读为0

bit 13-8 **RP35R[5:0]**: 将外设输出功能分配给RP35输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP34R[5:0]**: 将外设输出功能分配给RP34输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

# dsPIC33CK64MP105系列

寄存器8-50: RPOR2: 外设引脚选择输出寄存器2

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0
bit 15						bit 8	

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
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 15-14 未实现: 读为0

bit 13-8 **RP37R[5:0]**: 将外设输出功能分配给 RP37 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP36R[5:0]**: 将外设输出功能分配给 RP36 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

寄存器8-51: RPOR3: 外设引脚选择输出寄存器3

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0
bit 15						bit 8	

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP38R5	RP38R5	RP38R5	RP38R5	RP38R5	RP38R5
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 15-14 未实现: 读为0

bit 13-8 **RP39R[5:0]**: 将外设输出功能分配给 RP39 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP38R[5:0]**: 将外设输出功能分配给 RP38 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))



**寄存器8-52: RPOR4: 外设引脚选择输出寄存器4**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0
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 15-14      **未实现:** 读为0

bit 13-8      **RP41R[5:0]:** 将外设输出功能分配给RP41输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6      **未实现:** 读为0

bit 5-0      **RP40R[5:0]:** 将外设输出功能分配给RP40输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

**寄存器8-53: RPOR5: 外设引脚选择输出寄存器5**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0
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 15-14      **未实现:** 读为0

bit 13-8      **RP43R[5:0]:** 将外设输出功能分配给RP43输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6      **未实现:** 读为0

bit 5-0      **RP42R[5:0]:** 将外设输出功能分配给RP42输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

# dsPIC33CK64MP105系列

寄存器8-54: RPOR6: 外设引脚选择输出寄存器6

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
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 15-14 未实现: 读为0

bit 13-8 **RP45R[5:0]**: 将外设输出功能分配给RP45输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP44R[5:0]**: 将外设输出功能分配给RP44输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

寄存器8-55: RPOR7: 外设引脚选择输出寄存器7

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
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 15-14 未实现: 读为0

bit 13-8 **RP47R[5:0]**: 将外设输出功能分配给RP47输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP46R[5:0]**: 将外设输出功能分配给RP46输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-56: RPOR8: 外设引脚选择输出寄存器8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP49R5	RP49R4	RP49R3	RP49R2	RP49R1	RP49R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP48R5	RP48R4	RP48R3	RP48R2	RP48R1	RP48R0
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 15-14 未实现: 读为0

bit 13-8 **RP49R[5:0]:** 将外设输出功能分配给RP49输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP48R[5:0]:** 将外设输出功能分配给RP48输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-57: RPOR9: 外设引脚选择输出寄存器9

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP51R5	RP51R4	RP51R3	RP51R2	RP51R1	RP51R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP50R5	RP50R4	RP50R3	RP50R2	RP50R1	RP50R0
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 15-14 未实现: 读为0

bit 13-8 **RP51R[5:0]:** 将外设输出功能分配给RP51输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP50R[5:0]:** 将外设输出功能分配给RP50输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

# dsPIC33CK64MP105系列

**寄存器8-58: RPOR10: 外设引脚选择输出寄存器10**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP53R5	RP53R4	RP53R3	RP53R2	RP53R1	RP53R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP52R5	RP52R4	RP52R3	RP52R2	RP52R1	RP52R0
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 15-14 未实现: 读为0

bit 13-8 **RP53[5:0]:** 将外设输出功能分配给 RP53 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP52R[5:0]:** 将外设输出功能分配给 RP52 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

**寄存器8-59: RPOR11: 外设引脚选择输出寄存器11**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0
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 15-14 未实现: 读为0

bit 13-8 **RP55R[5:0]:** 将外设输出功能分配给 RP55 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP54R[5:0]:** 将外设输出功能分配给 RP54 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

## 寄存器8-60: RPOR12: 外设引脚选择输出寄存器12

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0
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 15-14 未实现: 读为0

bit 13-8 **RP57R[5:0]:** 将外设输出功能分配给RP57输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP56R[5:0]:** 将外设输出功能分配给RP56输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-61: RPOR13: 外设引脚选择输出寄存器13

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP59R5	RP59R4	RP59R3	RP59R2	RP59R1	RP59R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP58R5	RP58R4	RP58R3	RP58R2	RP58R1	RP58R0
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 15-14 未实现: 读为0

bit 13-8 **RP59R[5:0]:** 将外设输出功能分配给RP59输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP58R[5:0]:** 将外设输出功能分配给RP58输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

# dsPIC33CK64MP105系列

## 寄存器8-62: RPOR14: 外设引脚选择输出寄存器14

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP61R5	RP61R4	RP61R3	RP61R2	RP61R1	RP61R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP60R5	RP60R4	RP60R3	RP60R2	RP60R1	RP60R0
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 15-14 未实现: 读为0

bit 13-8 **RP61R[5:0]**: 将外设输出功能分配给RP61输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP60R[5:0]**: 将外设输出功能分配给RP60输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-63: RPOR15: 外设引脚选择输出寄存器15

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP72R5	RP72R4	RP72R3	RP72R2	RP72R1	RP72R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP65R5	RP65R4	RP65R3	RP65R2	RP65R1	RP65R0
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 15-14 未实现: 读为0

bit 13-8 **RP72R[5:0]**: 将外设输出功能分配给RP72输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP65R[5:0]**: 将外设输出功能分配给RP65输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-64: RPOR16: 外设引脚选择输出寄存器16

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP77R5	RP77R4	RP77R3	RP77R2	RP77R1	RP77R0
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP74R5	RP74R4	RP74R3	RP74R2	RP74R1	RP74R0
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 15-14 未实现: 读为0

bit 13-8 **RP77R[5:0]**: 将外设输出功能分配给 RP77 输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP74R[5:0]**: 将外设输出功能分配给 RP74 输出引脚的位  
(有关外设功能编号的信息, 见[表8-7](#))

## 寄存器8-65: RPOR17: 外设引脚选择输出寄存器17

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP177R5 <sup>(1)</sup>	RP177R4 <sup>(1)</sup>	RP177R3 <sup>(1)</sup>	RP177R2 <sup>(1)</sup>	RP177R1 <sup>(1)</sup>	RP177R0 <sup>(1)</sup>
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP176R5 <sup>(1)</sup>	RP176R4 <sup>(1)</sup>	RP176R3 <sup>(1)</sup>	RP176R2 <sup>(1)</sup>	RP176R1 <sup>(1)</sup>	RP176R0 <sup>(1)</sup>
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-14 未实现: 读为0

bit 13-8 **RP177R[5:0]**: 将外设输出功能分配给 RP177 输出引脚的位<sup>(1)</sup>  
(有关外设功能编号的信息, 见[表8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP176R[5:0]**: 将外设输出功能分配给 RP176 输出引脚的位<sup>(1)</sup>  
(有关外设功能编号的信息, 见[表8-7](#))

注 1: 这些是虚拟输出端口。

# dsPIC33CK64MP105系列

**寄存器8-66: RPOR18: 外设引脚选择输出寄存器18**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP179R5 <sup>(1)</sup>	RP179R4 <sup>(1)</sup>	RP179R3 <sup>(1)</sup>	RP179R2 <sup>(1)</sup>	RP179R1 <sup>(1)</sup>	RP179R0 <sup>(1)</sup>
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP178R5 <sup>(1)</sup>	RP178R4 <sup>(1)</sup>	RP178R3 <sup>(1)</sup>	RP178R2 <sup>(1)</sup>	RP178R1 <sup>(1)</sup>	RP178R0 <sup>(1)</sup>
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-14 未实现: 读为0

bit 13-8 **RP179R[5:0]**: 将外设输出功能分配给 RP179 输出引脚的位<sup>(1)</sup>  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP178R[5:0]**: 将外设输出功能分配给 RP178 输出引脚的位<sup>(1)</sup>  
(有关外设功能编号的信息, 见[表 8-7](#))

注 1: 这些是虚拟输出端口。

**寄存器8-67: RPOR19: 外设引脚选择输出寄存器19**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP181R5 <sup>(1)</sup>	RP181R4 <sup>(1)</sup>	RP181R3 <sup>(1)</sup>	RP181R2 <sup>(1)</sup>	RP181R1 <sup>(1)</sup>	RP181R0 <sup>(1)</sup>
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	RP180R5 <sup>(1)</sup>	RP180R4 <sup>(1)</sup>	RP180R3 <sup>(1)</sup>	RP180R2 <sup>(1)</sup>	RP180R1 <sup>(1)</sup>	RP180R0 <sup>(1)</sup>
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-14 未实现: 读为0

bit 13-8 **RP181R[5:0]**: 将外设输出功能分配给 RP181 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

bit 7-6 未实现: 读为0

bit 5-0 **RP180R[5:0]**: 将外设输出功能分配给 RP180 输出引脚的位  
(有关外设功能编号的信息, 见[表 8-7](#))

注 1: 这些是虚拟输出端口。



表8-12: PPS输入控制寄存器

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RPCON	—	—	—	—	IOLOCK	—	—	—	—	—	—	—	—	—	—	—
RPINR0	INT1R7	INT1R6	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—
RPINR1	INT3R7	INT3R6	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	INT2R7	INT2R6	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0
RPINR2	T1CKR7	T1CKR6	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0	—	—	—	—	—	—	—	—
RPINR3	ICM1R7	ICM1R6	ICM1R5	ICM1R4	ICM1R3	ICM1R2	ICM1R1	ICM1R0	TCKI1R7	TCKI1R6	TCKI1R5	TCKI1R4	TCKI1R3	TCKI1R2	TCKI1R1	TCKI1R0
RPINR4	ICM2R7	ICM2R6	ICM2R5	ICM2R4	ICM2R3	ICM2R2	ICM2R1	ICM2R0	TCKI2R7	TCKI2R6	TCKI2R5	TCKI2R4	TCKI2R3	TCKI2R2	TCKI2R1	TCKI2R0
RPINR5	ICM3R7	ICM3R6	ICM3R5	ICM3R4	ICM3R3	ICM3R2	ICM3R1	ICM3R0	TCKI3R7	TCKI3R6	TCKI3R5	TCKI3R4	TCKI3R3	TCKI3R2	TCKI3R1	TCKI3R0
RPINR6	ICM4R7	ICM4R6	ICM4R5	ICM4R4	ICM4R3	ICM4R2	ICM4R1	ICM4R0	TCKI4R7	TCKI4R6	TCKI4R5	TCKI4R4	TCKI4R3	TCKI4R2	TCKI4R1	TCKI4R0
RPINR7	ICM5R7	ICM5R6	ICM5R5	ICM5R4	ICM5R3	ICM5R2	ICM5R1	ICM5R0	TCKI5R7	TCKI5R6	TCKI5R5	TCKI5R4	TCKI5R3	TCKI5R2	TCKI5R1	TCKI5R0
RPINR11	OCFBR7	OCFBR6	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0	OCFAR7	OCFAR6	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0
RPINR12	PCI9R7	PCI9R6	PCI9R5	PCI9R4	PCI9R3	PCI9R2	PCI9R1	PCI9R0	PCI8R7	PCI8R6	PCI8R5	PCI8R4	PCI8R3	PCI8R2	PCI8R1	PCI8R0
RPINR13	PCI11R7	PCI11R6	PCI11R5	PCI11R4	PCI11R3	PCI11R2	PCI11R1	PCI11R0	PCI10R7	PCI10R6	PCI10R5	PCI10R4	PCI10R3	PCI10R2	PCI10R1	PCI10R0
RPINR14	QEIB1R7	QEIB1R6	QEIB1R5	QEIB1R4	QEIB1R3	QEIB1R2	QEIB1R1	QEIB1R0	QEIA1R7	QEIA1R6	QEIA1R5	QEIA1R4	QEIA1R3	QEIA1R2	QEIA1R1	QEIA1R0
RPINR15	QEIHM1R7	QEIHM1R6	QEIHM1R5	QEIHM1R4	QEIHM1R3	QEIHM1R2	QEIHM1R1	QEIHM1R0	QEINDX1R7	QEINDX1R6	QEINDX1R5	QEINDX1R4	QEINDX1R3	QEINDX1R2	QEINDX1R1	QEINDX1R0
RPINR16	QEIB2R7	QEIB2R6	QEIB2R5	QEIB2R4	QEIB2R3	QEIB2R2	QEIB2R1	QEIB2R0	QEIA2R7	QEIA2R6	QEIA2R5	QEIA2R4	QEIA2R3	QEIA2R2	QEIA2R1	QEIA2R0
RPINR17	QEIHM2R7	QEIHM2R6	QEIHM2R5	QEIHM2R4	QEIHM2R3	QEIHM2R2	QEIHM2R1	QEIHM2R0	QEINDX2R7	QEINDX2R6	QEINDX2R5	QEINDX2R4	QEINDX2R3	QEINDX2R2	QEINDX2R1	QEINDX2R0
RPINR18	U1DSRR7	U1DSRR6	U1DSRR5	U1DSRR4	U1DSRR3	U1DSRR2	U1DSRR1	U1DSRR0	U1RXR7	U1RXR6	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0
RPINR19	U2DSRR7	U2DSRR6	U2DSRR5	U2DSRR4	U2DSRR3	U2DSRR2	U2DSRR1	U2DSRR0	U2RXR7	U2RXR6	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0
RPINR20	SCK1R7	SCK1R6	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	SDI1R7	SDI1R6	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0
RPINR21	REFOIR7	REFOIR6	REFOIR5	REFOIR4	REFOIR3	REFOIR2	REFOIR1	REFOIR0	SS1R7	SS1R6	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0
RPINR22	SCK2R7	SCK2R6	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	SDI2R7	SDI2R6	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0
RPINR23	—	—	—	—	—	—	—	—	SS2R7	SS2R6	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0
RPINR27	U3DSRR7	U3DSRR6	U3DSRR5	U3DSRR4	U3DSRR3	U3DSRR2	U3DSRR1	U3DSRR0	U3RXR7	U3RXR6	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
RPINR29	SCK3R7	SCK3R6	SCK3R5	SCK3R4	SCK3R3	SCK3R2	SCK3R1	SCK3R0	SDI3R7	SDI3R6	SDI3R5	SDI3R4	SDI3R3	SDI3R2	SDI3R1	SDI3R0
RPINR30	—	—	—	—	—	—	—	—	SS3R7	SS3R6	SS3R5	SS3R4	SS3R3	SS3R2	SS3R1	SS3R0
RPINR37	PCI17R7	PCI17R6	PCI17R5	PCI17R4	PCI17R3	PCI17R2	PCI17R1	PCI17R0	OCFCR7	OCFCR6	OCFCR5	OCFCR4	OCFCR3	OCFCR2	OCFCR1	OCFCR0
RPINR38	—	—	—	—	—	—	—	—	PCI18R7	PCI18R6	PCI18R5	PCI18R4	PCI18R3	PCI18R2	PCI18R1	PCI18R0
RPINR42	PCI13R7	PCI13R6	PCI13R5	PCI13R4	PCI13R3	PCI13R2	PCI13R1	PCI13R0	PCI12R7	PCI12R6	PCI12R5	PCI12R4	PCI12R3	PCI12R2	PCI12R1	PCI12R0
RPINR43	PCI15R7	PCI15R6	PCI15R5	PCI15R4	PCI15R3	PCI15R2	PCI15R1	PCI15R0	PCI14R7	PCI14R6	PCI14R5	PCI14R4	PCI14R3	PCI14R2	PCI14R1	PCI14R0
RPINR44	SENT1R7	SENT1R6	SENT1R5	SENT1R4	SENT1R3	SENT1R2	SENT1R1	SENT1R0	PCI16R7	PCI16R6	PCI16R5	PCI16R4	PCI16R3	PCI16R2	PCI16R1	PCI16R0
RPINR45	CLCINAR7	CLCINAR6	CLCINAR5	CLCINAR4	CLCINAR3	CLCINAR2	CLCINAR1	CLCINAR0	SENT2R7	SENT2R6	SENT2R5	SENT2R4	SENT2R3	SENT2R2	SENT2R1	SENT2R0
RPINR46	CLCINCR7	CLCINCR6	CLCINCR5	CLCINCR4	CLCINCR3	CLCINCR2	CLCINCR1	CLCINCR0	CLCINBR7	CLCINBR6	CLCINBR5	CLCINBR4	CLCINBR3	CLCINBR2	CLCINBR1	CLCINBR0
RPINR47	ADCTRGR7	ADCTRGR6	ADCTRGR5	ADCTRGR4	ADCTRGR3	ADCTRGR2	ADCTRGR1	ADCTRGR0	CLCINDR7	CLCINDR6	CLCINDR5	CLCINDR4	CLCINDR3	CLCINDR2	CLCINDR1	CLCINDR0
RPINR48	U1CTSR7	U1CTSR6	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	OCFDR7	OCFDR6	OCFDR5	OCFDR4	OCFDR3	OCFDR2	OCFDR1	OCFDR0
RPINR49	U3CTSR7	U3CTSR6	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	U2CTSR7	U2CTSR6	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0

表8-13: PPS输出控制寄存器

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RPOR0	—	—	RP33R5	RP33R4	RP33R3	RP33R2	RP33R1	RP33R0	—	—	RP32R5	RP32R4	RP32R3	RP32R2	RP32R1	RP32R0
RPOR1	—	—	RP35R5	RP35R4	RP35R3	RP35R2	RP35R1	RP35R0	—	—	RP34R5	RP34R4	RP34R3	RP34R2	RP34R1	RP34R0
RPOR2	—	—	RP37R5	RP37R4	RP37R3	RP37R2	RP37R1	RP37R0	—	—	RP36R5	RP36R4	RP36R3	RP36R2	RP36R1	RP36R0
RPOR3	—	—	RP39R5	RP39R4	RP39R3	RP39R2	RP39R1	RP39R0	—	—	RP38R5	RP38R4	RP38R3	RP38R2	RP38R1	RP38R0
RPOR4	—	—	RP41R5	RP41R4	RP41R3	RP41R2	RP41R1	RP41R0	—	—	RP40R5	RP40R4	RP40R3	RP40R2	RP40R1	RP40R0
RPOR5	—	—	RP43R5	RP43R4	RP43R3	RP43R2	RP43R1	RP43R0	—	—	RP42R5	RP42R4	RP42R3	RP42R2	RP42R1	RP42R0
RPOR6	—	—	RP45R5	RP45R4	RP45R3	RP45R2	RP45R1	RP45R0	—	—	RP44R5	RP44R4	RP44R3	RP44R2	RP44R1	RP44R0
RPOR7	—	—	RP47R5	RP47R4	RP47R3	RP47R2	RP47R1	RP47R0	—	—	RP46R5	RP46R4	RP46R3	RP46R2	RP46R1	RP46R0
RPOR8	—	—	RP49R5	RP49R4	RP49R3	RP49R2	RP49R1	RP49R0	—	—	RP48R5	RP48R4	RP48R3	RP48R2	RP48R1	RP48R0
RPOR9	—	—	RP51R5	RP51R4	RP51R3	RP51R2	RP51R1	RP51R0	—	—	RP50R5	RP50R4	RP50R3	RP50R2	RP50R1	RP50R0
RPOR10	—	—	RP53R5	RP53R4	RP53R3	RP53R2	RP53R1	RP53R0	—	—	RP52R5	RP52R4	RP52R3	RP52R2	RP52R1	RP52R0
RPOR11	—	—	RP55R5	RP55R4	RP55R3	RP55R2	RP55R1	RP55R0	—	—	RP54R5	RP54R4	RP54R3	RP54R2	RP54R1	RP54R0
RPOR12	—	—	RP57R5	RP57R4	RP57R3	RP57R2	RP57R1	RP57R0	—	—	RP56R5	RP56R4	RP56R3	RP56R2	RP56R1	RP56R0
RPOR13	—	—	RP59R5	RP59R4	RP59R3	RP59R2	RP59R1	RP59R0	—	—	RP58R5	RP58R4	RP58R3	RP58R2	RP58R1	RP58R0
RPOR14	—	—	RP61R5	RP61R4	RP61R3	RP61R2	RP61R1	RP61R0	—	—	RP60R5	RP60R4	RP60R3	RP60R2	RP60R1	RP60R0
RPOR15	—	—	RP72R5	RP72R4	RP72R3	RP72R2	RP72R1	RP72R0	—	—	RP65R5	RP65R4	RP65R3	RP65R2	RP65R1	RP65R0
RPOR16	—	—	RP77R5	RP77R4	RP77R3	RP77R2	RP77R1	RP77R0	—	—	RP74R5	RP74R4	RP74R3	RP74R2	RP74R1	RP74R0
RPOR17	—	—	RP177R5	RP177R4	RP177R3	RP177R2	RP177R1	RP177R0	—	—	RP176R5	RP176R4	RP176R3	RP176R2	RP176R1	RP176R0
RPOR18	—	—	RP179R5	RP179R4	RP179R3	RP179R2	RP179R1	RP179R0	—	—	RP178R5	RP178R4	RP178R3	RP178R2	RP178R1	RP178R0
RPOR19	—	—	RP181R5	RP181R4	RP181R3	RP181R2	RP181R1	RP181R0	—	—	RP180R5	RP180R4	RP180R3	RP180R2	RP180R1	RP180R0

9.0 带高频PLL的振荡器

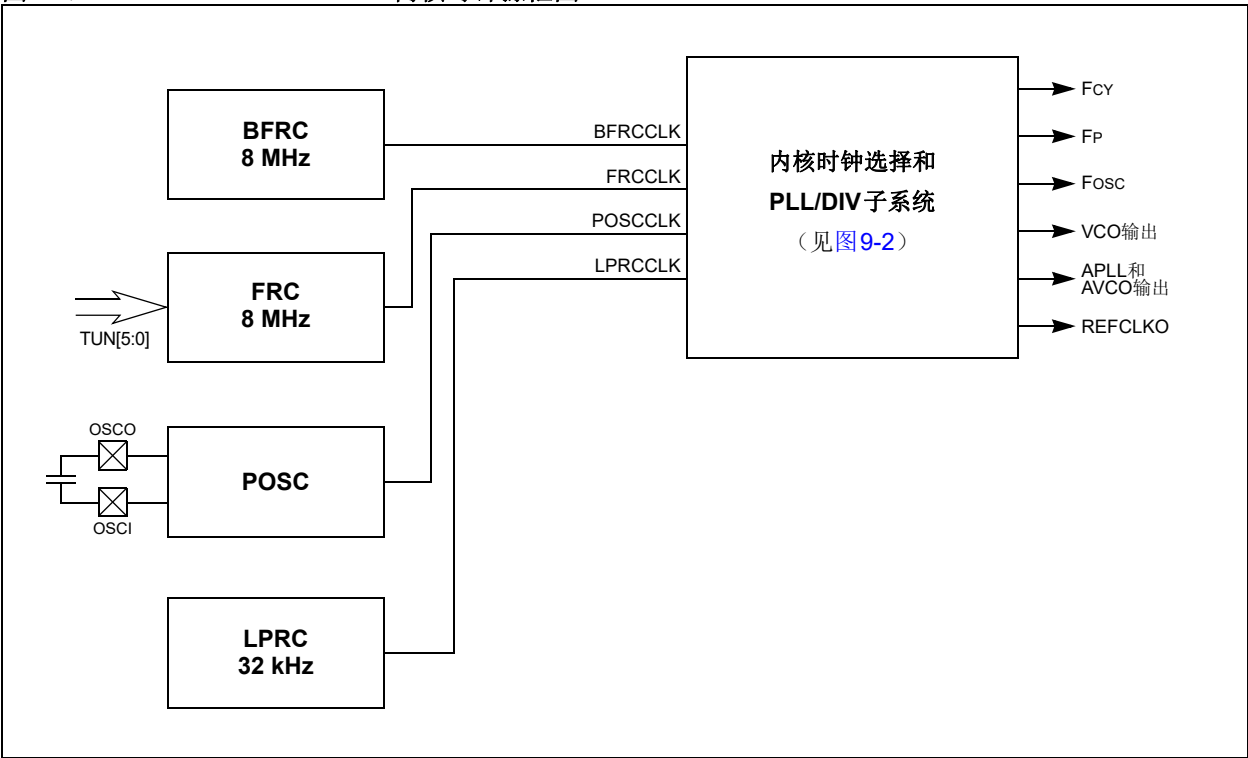
**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“带高速PLL的振荡器模块”（[www.microchip.com/DS70005255](http://www.microchip.com/DS70005255)）。

dsPIC33CK64MP105 系列带高频PLL 的振荡器包括以下特性：

- 片上锁相环（Phase-Locked Loop，PLL），可基于所选的内部和外部振荡器源升高内部工作频率
- 附属PLL（Auxiliary PLL，APLL）时钟发生器，可提升外设的工作频率
- 用于节省系统功耗的打盹模式
- 可扩展的参考时钟输出（Reference Clock Output，REFCLKO）
- 各种时钟源之间的动态时钟切换
- 故障保护时钟监视（FSCM），可检测时钟故障并允许安全地恢复或关闭应用

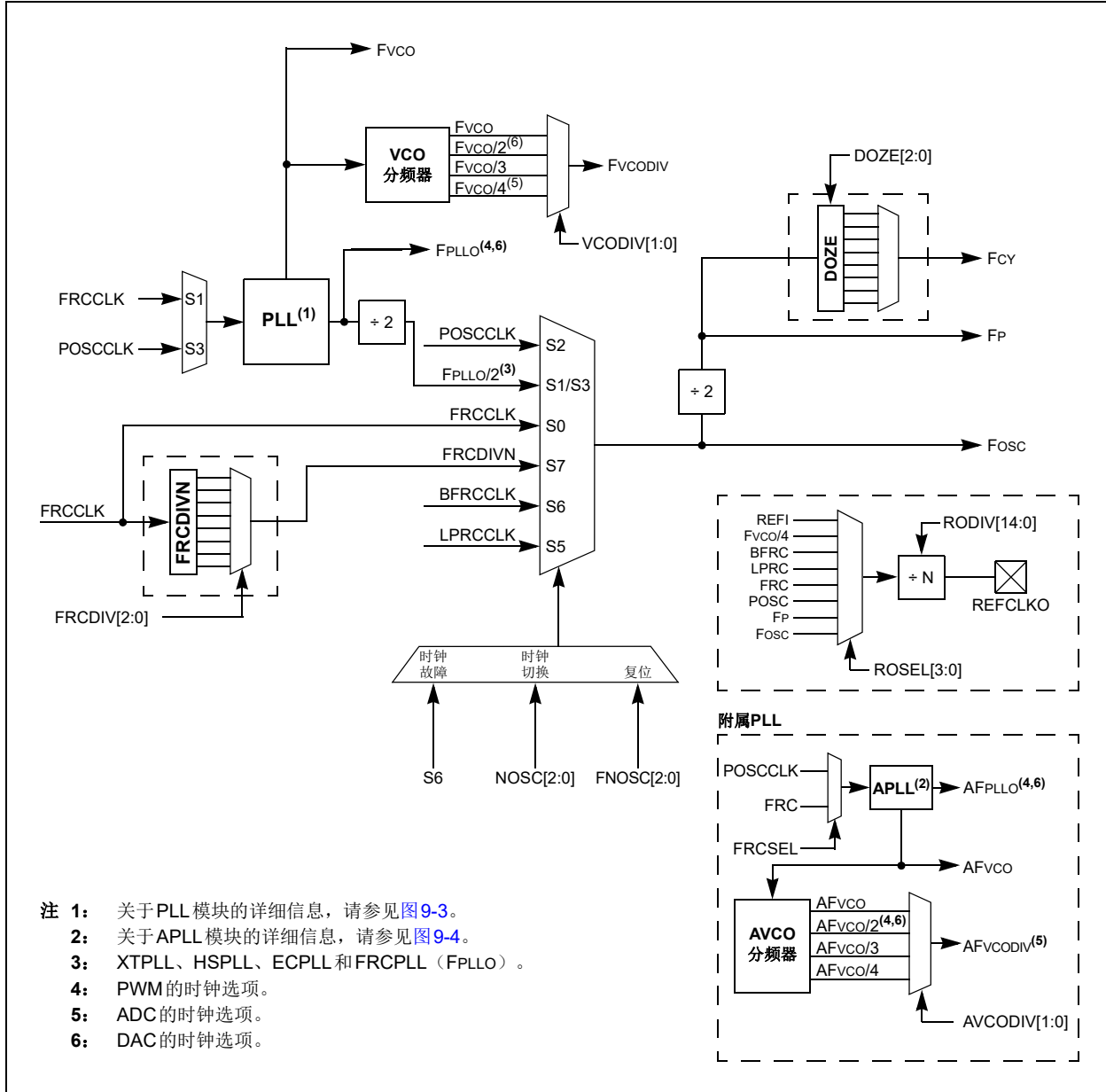
图9-1 给出了 dsPIC33CK64MP105 振荡器系统的框图。

图9-1: dsPIC33CK64MP105内核时钟源框图



# dsPIC33CK64MP105系列

图9-2: dsPIC33CK64MP105内核振荡器子系统



9.1 主PLL

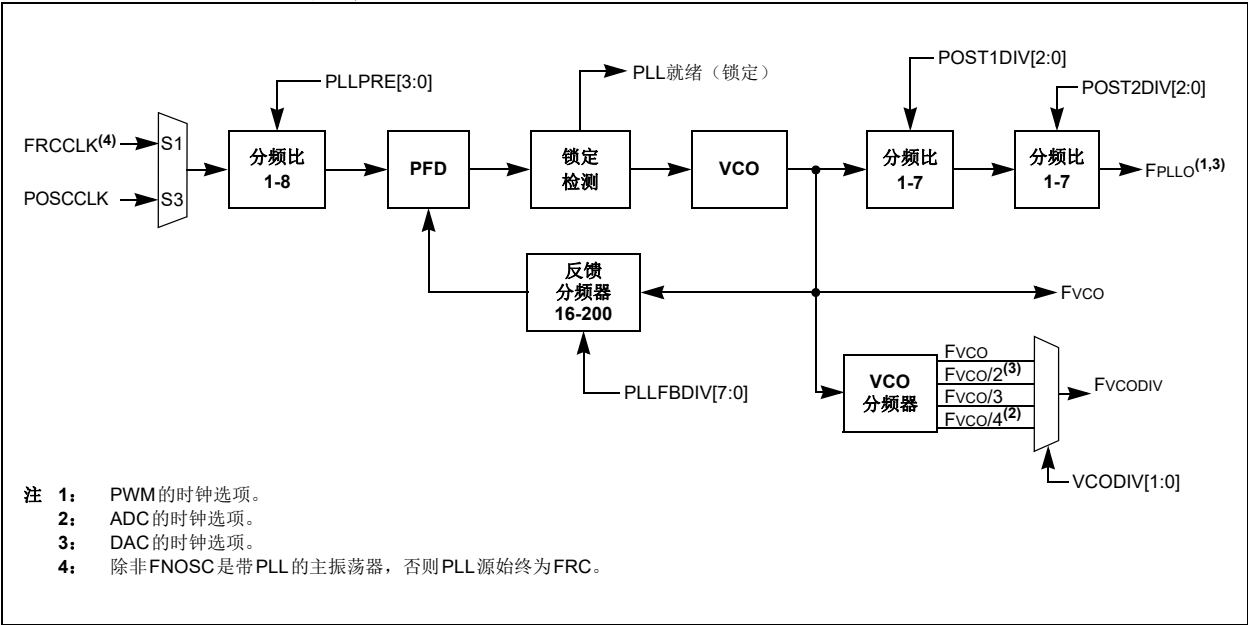
主振荡器和内部FRC振荡器源可选择使用片上PLL来获取更高的工作速度。图9-3给出了PLL模块的框图。

为了让PLL工作，必须始终毫无例外地满足以下要求：

- PLL输入频率（FPLLI）必须介于8 MHz到64 MHz之间
- PFD输入频率（FPFD）必须介于8 MHz到（Fvco/16）MHz之间

VCO输出频率（Fvco）必须介于400 MHz到1600 MHz之间

图9-3: PLL和VCO的详细信息



# dsPIC33CK64MP105系列

公式9-1给出了PLL输入频率（F<sub>PLLI</sub>）和VCO输出频率（F<sub>VCO</sub>）之间的关系。

**公式9-1: F<sub>VCO</sub>计算**

$$F_{VCO} = F_{PLLI} \times \left( \frac{M}{N1} \right) = F_{PLLI} \times \left( \frac{PLLFBDIV[7:0]}{PLLPRE[3:0]} \right)$$

公式9-2给出了PLL输入频率（F<sub>PLLI</sub>）和PLL输出频率（F<sub>PLLO</sub>）之间的关系。

**公式9-2: F<sub>PLLO</sub>计算**

$$F_{PLLO} = F_{PLLI} \times \left( \frac{M}{N1 \times N2 \times N3} \right) = F_{PLLI} \times \left( \frac{PLLFBDIV[7:0]}{PLLPRE[3:0] \times POST1DIV[2:0] \times POST2DIV[2:0]} \right)$$

其中：

$M = PLLFBDIV[7:0]$   
 $N1 = PLLPRE[3:0]$   
 $N2 = POST1DIV[2:0]$   
 $N3 = POST2DIV[2:0]$

**注：** 在PLL模式下工作时，不应更改PLL相位检测器输入分频比选择（PLLPRE<sub>x</sub>）位和PLL反馈分频比（PLLFBDIV<sub>x</sub>）位。因此，用户必须处于非PLL模式或者将时钟切换到非PLL模式（例如，内部FRC振荡器），做出任何必要的更改后，再将时钟切换到所需的PLL模式。

不允许将时钟从一个PLL时钟源直接切换到另一个PLL时钟源。用户需要通过将时钟切换到非PLL时钟源来进行PLL时钟源之间的切换。

例9-1给出了PLL（50 MIPS）与主振荡器配合使用的代码。

## 例9-1: PLL（50 MIPS）与主振荡器（POSC）配合使用的代码示例

```
//code example for 50 MIPS system clock using POSC with 10 MHz external crystal

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// Enable Clock Switching and Configure POSC in XT mode
_FOSC(FCKSM_CSECMD & POSCMD_XT);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 100;      // M = 100
    PLLDIVbits.POST1DIV = 5;        // N2=5
    PLLDIVbits.POST2DIV = 1;        // N3=1

    // Initiate Clock Switch to Primary Oscillator with PLL (NOSC=0b011)
    __builtin_write_OSCCONH(0x03);
    __builtin_write_OSCCONL(OSCCON | 0x01);

    // Wait for Clock switch to occur
    while (OSCCONbits.OSWEN != 0);

    // Wait for PLL to lock
    while (OSCCONbits.LOCK != 1);
}
```

例9-2给出了PLL与8 MHz内部FRC配合使用的代码。

## 例9-2: PLL（50 MIPS）与8 MHz内部FRC配合使用的代码示例

```
//code example for 50 MIPS system clock using 8MHz FRC

// Select Internal FRC at POR
_FOSCSEL(FNOSC_FRC & IESO_OFF);

// Enable Clock Switching
_FOSC(FCKSM_CSECMD);

int main()
{
    // Configure PLL prescaler, both PLL postscalers, and PLL feedback divider
    CLKDIVbits.PLLPRE = 1;          // N1=1
    PLLFBDbits.PLLFBDIV = 125;      // M = 125
    PLLDIVbits.POST1DIV = 5;        // N2=5
    PLLDIVbits.POST2DIV = 1;        // N3=1

    // Initiate Clock Switch to FRC with PLL (NOSC=0b001)
    __builtin_write_OSCCONH(0x01);
    __builtin_write_OSCCONL(OSCCON | 0x01);

    // Wait for Clock switch to occur
    while (OSCCONbits.OSWEN != 0);

    // Wait for PLL to lock
    while (OSCCONbits.LOCK != 1);
}
```

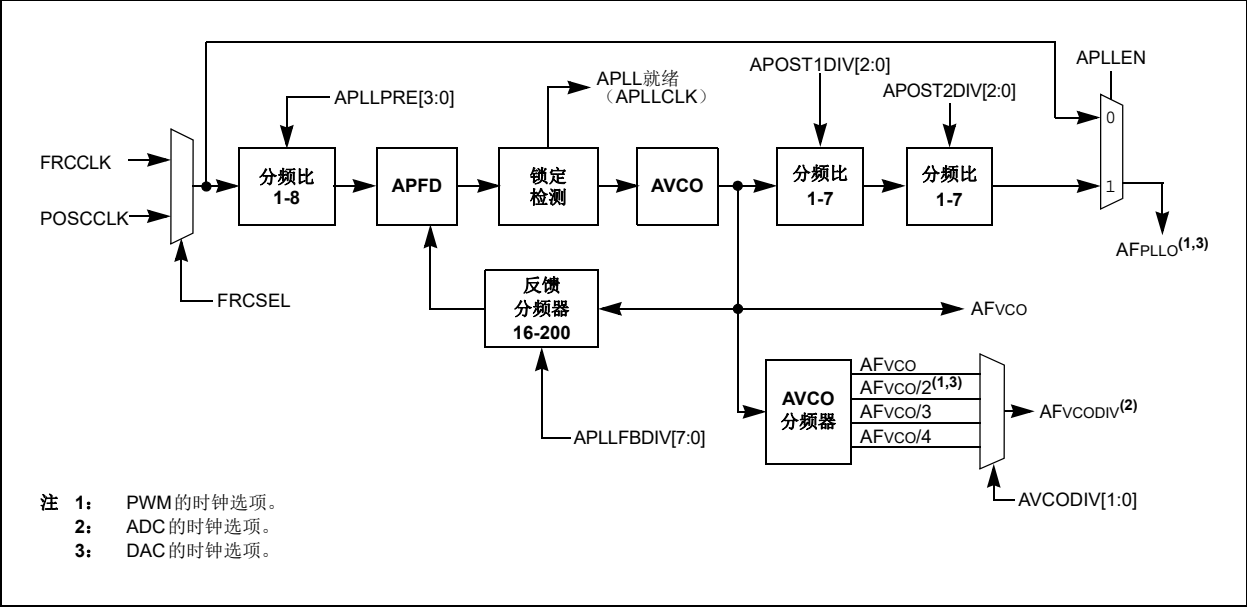
9.2 附属PLL

dsPIC33CK64MP105 器件系列实现了一个附属PLL（APLL）模块，用来生成独立于系统时钟的各种外设时钟源。图9-4给出了APLL模块的框图。

为了让APLL工作，必须始终毫无例外地满足以下要求：

- APLL输入频率（AFPLLI）必须介于8 MHz到64 MHz之间
- APFD输入频率（AFPFDD）必须介于8 MHz到（AFVCO/16）MHz之间
- AVCO输出频率（AFVCO）必须介于400 MHz到1600 MHz之间

图9-4: APLL和VCO的详细信息





公式9-3给出了APLL输入频率（AFPLLI）和AVCO输出频率（AFVCO）之间的关系。

## 公式9-3: AFVCO计算

$$AFVCO = AFPLLI \times \left( \frac{M}{N1} \right) = AFPLLI \times \left( \frac{APLLFBDIV[7:0]}{APLLPRE[3:0]} \right)$$

公式9-4给出了APLL输入频率（AFPLLI）和APLL输出频率（AFPLLO）之间的关系。

## 公式9-4: AFPLLO计算

$$AFPLLO = AFPLLI \times \left( \frac{M}{N1 \times N2 \times N3} \right) = AFPLLI \times \left( \frac{APLLFBDIV[7:0]}{APLLPRE[3:0] \times APOST1DIV[2:0] \times APOST2DIV[2:0]} \right)$$

其中:

$$M = APLLFBDIV[7:0]$$

$$N1 = APLLPRE[3:0]$$

$$N2 = APOST1DIV[2:0]$$

$$N3 = APOST2DIV[2:0]$$

## 例9-3: 附属PLL与内部FRC振荡器配合使用的代码示例

```
//code example for AFVCO = 1 GHz and AFPLLO = 500 MHz using 8 MHz internal FRC
// Configure the source clock for the APLL
ACLKCON1bits.FRCSEL = 1;           // Select internal FRC as the clock source
// Configure the APLL prescaler, APLL feedback divider, and both APLL postscalers.
ACLKCON1bits.APLLPRE = 1;           // N1 = 1
APLLFBD1bits.APLLFBDIV = 125;       // M = 125
APLLDIV1bits.APOST1DIV = 2;         // N2 = 2
APLLDIV1bits.APOST2DIV = 1;         // N3 = 1
// Enable APLL
ACLKCON1bits.APLEN = 1;
```

**注:** 即使APLEN位已置1，还必须有一个外设产生时钟请求，然后APLL才能启动。

## 9.3 CPU时钟

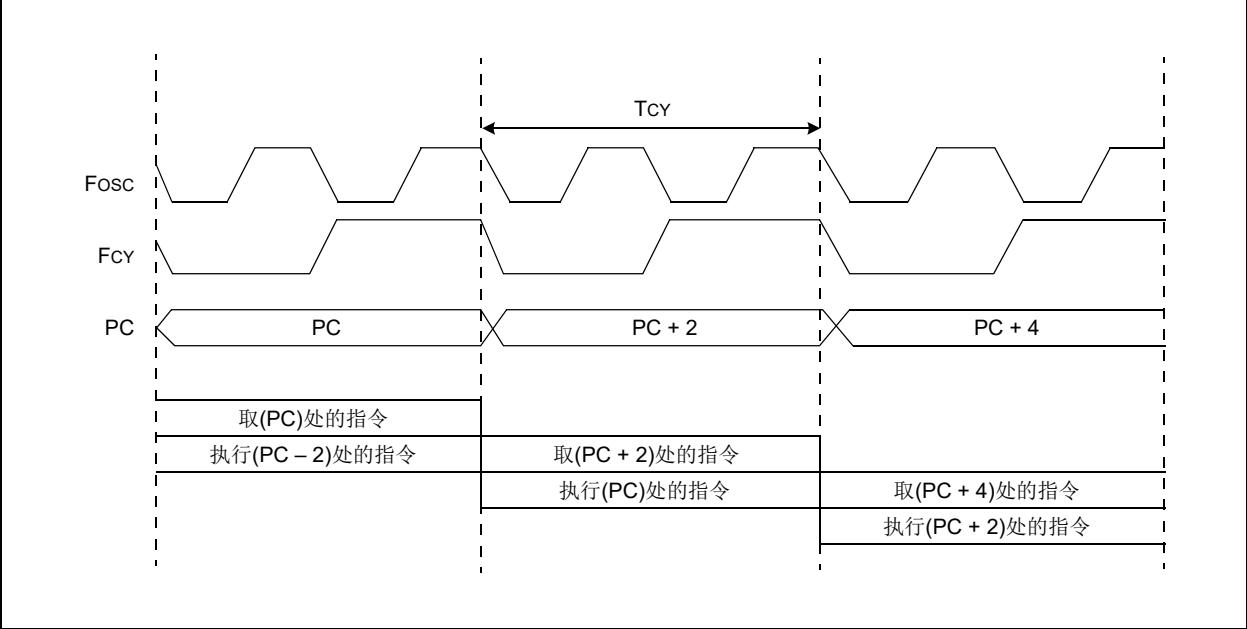
dsPIC33CK64MP105 器件可以配置为使用以下任意时钟配置：

- OSCI和OSCO引脚上的主振荡器（POSC）
- 可选时钟分频比的内部快速RC（FRC）振荡器
- 内部低功耗RC（LPRC）振荡器
- 带PLL的主振荡器（ECPLL、HSPLL和XTPLL）
- 带PLL的内部快速RC（FRCPLL）振荡器
- 内部备用快速RC（Backup Fast RC，BFRC）振荡器

系统时钟源进行2分频，以产生内部指令周期时钟。在本文档中，指令周期时钟用Fcy表示。图9-5中的时序图说明了系统时钟（Fosc）、指令周期时钟（Fcy）和程序计数器（PC）之间的关系。

如果主振荡器模式（POSCMD[1:0]）未配置为HS/XT，则可以在OSCO I/O 引脚上输出内部指令周期时钟（Fcy）。更多信息，请参见第9.0节“带高频PLL的振荡器”。

图9-5： 时钟和指令周期时序



## 9.4 主振荡器（POSC）

dsPIC33CK64MP105系列器件具有主振荡器（POSC），该振荡器连接在OSCI和OSCO引脚之间。此连接使能外部晶振（或陶瓷谐振器）向器件提供时钟。主振荡器提供三种工作模式：

- 中速振荡器（XT模式）：  
XT模式是用于以3.5 MHz到10 MHz的晶振频率进行工作的中等增益中频模式。
- 高速振荡器（HS模式）：  
HS模式是用于以10 MHz到32 MHz的晶振频率进行工作的高增益高频模式。
- 外部时钟源工作（EC模式）：  
如果未使用片上振荡器，EC模式将允许旁路内部振荡器。器件时钟从外部源（0 MHz到最高64 MHz）产生，并在OSCI引脚上输入。

## 9.5 内部快速RC（FRC）振荡器

dsPIC33CK64MP105系列器件包含一个内部快速RC（FRC）振荡器实例，可提供标称8 MHz时钟，不需要外部晶振或陶瓷谐振器，因此将可以为不需要精确时钟参考的应用节省系统成本。

应用程序软件可以使用FRC振荡器调节寄存器中的FRC振荡器调节位（TUN[5:0]）（OSCTUN[5:0]）来调节振荡器的频率。

## 9.6 低功耗RC（LPRC）振荡器

dsPIC33CK64MP105系列器件包含一个低功耗RC（LPRC）振荡器实例，可提供32 kHz的标称时钟频率，是时钟子系统上的上电延时定时器（Power-up Timer, PWRT）、看门狗定时器（WDT）和故障保护时钟监视器（FSCM）电路的时钟源。

LPRC振荡器是PWRT、WDT和FSCM的时钟源。LPRC振荡器在上电时使能。

LPRC振荡器在以下任一条件下保持使能：

- FSCM使能
- WDT使能
- LPRC振荡器已选择为系统时钟

如果上述条件没有一个为true，LPRC振荡器将在PWRT延时结束后关闭。LPRC振荡器在休眠模式下将关闭。

## 9.7 内部备用快速RC（BFRC）振荡器

振荡器模块为故障保护时钟监视器（FSCM）提供稳定的参考时钟源。当通过FCKSM[1:0]配置位（FOSC[7:6]）使能FSCM时，FSCM会根据来自8 MHz内部备用快速RC（BFRC）振荡器的参考信号持续监视主时钟源。如果发生时钟故障，故障保护时钟监视器会将时钟切换到BFRC振荡器，从而允许持续低速操作或安全关闭应用。

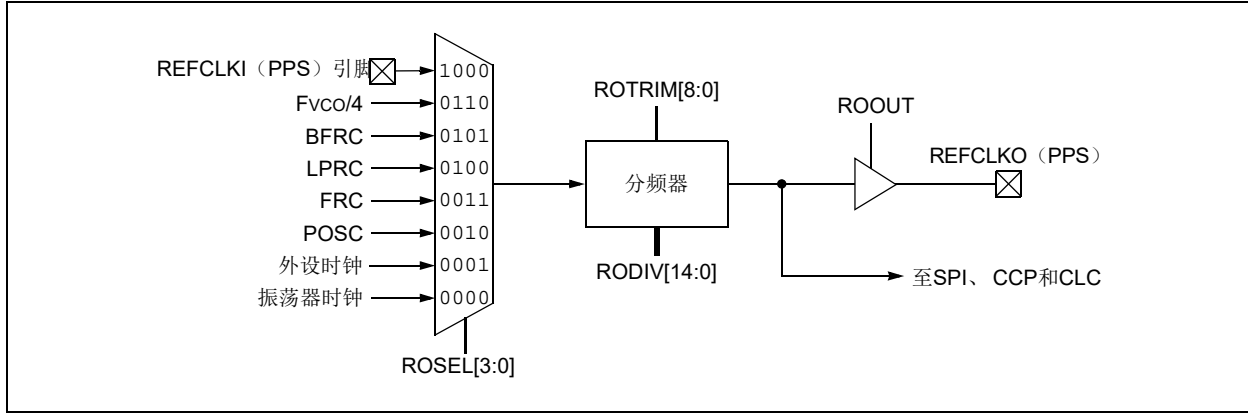
# dsPIC33CK64MP105系列

## 9.8 参考时钟输出

除了CLKO输出（Fosc/2）外，dsPIC33CK64MP105系列器件还可配置为向端口引脚提供参考时钟输出信号。此功能适用于所有振荡器配置，允许用户在更大范围内选择时钟因数来驱动应用中的外部器件。CLKO由配置

位OSCIOFCN使能，且独立于REFCLKO参考时钟。REFCLKO可映射到任何具有映射输出功能的I/O引脚。更多信息，请参见表8-7。图9-6给出了参考时钟输出模块的框图。

图9-6： 参考时钟发生器



该参考时钟输出由REFOCONL和REFOCONH寄存器控制。将ROEN位（REFOCONL[15]）置1可在REFCLKO引脚上输出时钟信号。RODM[14:0]位（REFOTRIMH[15:7]）和ROTRIM[8:0]位（REFOTRIMH[15:7]）用于使能选择不同的时钟分频比选项。公式9-5用于确定最终频率输出。ROSWEN位（REFOCONL[9]）为0指示已成功切换时钟分频比。为了切换REFCLKO分频比，用户应确保该位读为0。将更新后的分频比值写入RODIV[14:0]或ROTRIM[8:0]位，将ROSWEN位置1，然后等待其清零，然后才能认为REFCLKO时钟有效。

公式9-5： 计算频率输出

$$F_{REFOUT} = \frac{F_{REFIN}}{2 \cdot (RODIV[14:0] + ROTRIM[8:0]/512)}$$

其中：  $F_{REFOUT}$  = 输出频率  
 $F_{REFIN}$  = 输入频率  
当RODIV[14:0] = 0时，输出时钟与输入时钟相同。

ROSEL[3:0]位（REFOCONL[3:0]）确定哪个时钟源用于参考时钟输出。ROSLP位（REFOCONL[11]）确定当器件处于休眠模式时，参考时钟源是否可用于REFCLKO。

要在休眠模式下使用参考时钟输出，必须将ROSLP位置1，并且如果可能，必须使能ROSEL[3:0]位选择的时钟在休眠模式下工作。在任何时钟切换期间，清零ROSEL[3:0]位允许参考时钟输出频率随系统时钟的变化而变化。ROOUT位使能/禁止REFCLKO引脚上的参考时钟输出。

ROACTIV位（REFOCONL[8]）指示模块是否处于活动状态；可以通过禁止模块（将ROEN设置为0）将其清零。在ROACTIV位指示模块处于活动状态时，用户不得更改参考时钟源，也不能调整分频比。为避免故障，用户应仅在ROACTIV位为1时禁止模块。

9.9 振荡器配置

振荡器系统具有配置寄存器和SFR，用于配置、控制和监视系统。FOSCSEL和FOSC配置寄存器（分别对应于寄存器28-4和寄存器28-5）用于初始设置。

表9-1列出了在上电复位（Power-on Reset，POR）时选择器件振荡器源和工作模式的配置设置。

表9-1： 用于时钟选择的配置位值

振荡器源	振荡器模式	FNOSC[2:0] 值	POSCMD[1:0] 值
S0	快速RC振荡器（FRC）	000	xx
S1	带PLL的快速RC振荡器（FRCPLL）	001	xx
S2	主振荡器（EC）	010	00
S2	主振荡器（XT）	010	01
S2	主振荡器（HS）	010	10
S3	带PLL的主振荡器（ECPLL）	011	00
S3	带PLL的主振荡器（XTPLL）	011	01
S3	带PLL的主振荡器（HSPLL）	011	10
S4	保留	100	xx
S5	低功耗RC振荡器（LPRC）	101	xx
S6	备用FRC（BFRC）	110	xx
S7	N分频快速RC振荡器（FRCDIVN）	111	xx

9.10 OSCCON解锁序列

通过锁定机制保护OSCCON寄存器免受意外写入。OSCCON的高字节和低字节都有自己的解锁序列，写入寄存器的高低字节时必须使用对应的解锁序列。在写入OSCCON之前，必须使用以下解锁序列：

1. 执行OSCCON高字节的解锁序列。  
使用两条连续指令：
  - 将0x78写入OSCCON[15:8]
  - 将0x9A写入OSCCON[15:8]
2. 在紧接解锁序列之后的指令中，可以修改OSCCON[15:8]位。

3. 执行OSCCON低字节的解锁序列。  
使用两条连续指令：
  - 将0x46写入OSCCON[7:0]
  - 将0x57写入OSCCON[7:0]
4. 在紧接解锁序列之后的指令中，可以修改OSCCON[7:0]位。

注： MPLAB® XC16提供了一个内置C语言函数，包括用于修改OSCCON寄存器中的高字节和低字节的解锁序列：  
\_\_builtin\_write\_OSCCONH(value)  
\_\_builtin\_write\_OSCCONL(value)

# dsPIC33CK64MP105系列

## 9.11 振荡器控制寄存器

寄存器9-1: **OSCCON**: 振荡器控制寄存器<sup>(1)</sup>

U-0	R-0	R-0	R-0	U-0	R/W-y	R/W-y	R/W-y
—	COSC2	COSC1	COSC0	—	NOSC2 <sup>(2)</sup>	NOSC1 <sup>(2)</sup>	NOSC0 <sup>(2)</sup>
bit 15				bit 8			
R/W-0	U-0	R-0	U-0	R/W-0	U-0	U-0	R/W-0
CLKLOCK	—	LOCK	—	CF <sup>(3)</sup>	—	—	OSWEN
bit 7				bit 0			

图注: y = 在POR时由配置位设置的值  
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 15 未实现: 读为0

bit 14-12 **COSC[2:0]**: 当前振荡器选择位 (只读)  
111 = n分频快速RC振荡器 (FRC) (FRCDIVN)  
110 = 备用FRC (BFRC)  
101 = 低功耗RC振荡器 (LPRC)  
100 = 保留——默认为FRC  
011 = 带PLL的主振荡器 (XT、HS和EC) (XTPLL、HSPLL和ECPLL)  
010 = 主振荡器 (XT、HS和EC)  
001 = 带PLL的快速RC振荡器 (FRC) (FRCPLL)  
000 = 快速RC振荡器 (FRC)

bit 11 未实现: 读为0

bit 10-8 **NOSC[2:0]**: 新振荡器选择位<sup>(2)</sup>  
111 = n分频快速RC振荡器 (FRC) (FRCDIVN)  
110 = 备用FRC (BFRC)  
101 = 低功耗RC振荡器 (LPRC)  
100 = 保留——默认为FRC  
011 = 带PLL的主振荡器 (XT、HS和EC) (XTPLL、HSPLL和ECPLL)  
010 = 主振荡器 (XT、HS和EC)  
001 = 带PLL的快速RC振荡器 (FRC) (FRCPLL)  
000 = 快速RC振荡器 (FRC)

bit 7 **CLKLOCK**: 时钟锁定使能位  
1 = 如果FCKSM0 = 1, 则锁定时钟和PLL配置; 如果FCKSM0 = 0, 则可以修改时钟和PLL配置  
0 = 不锁定时钟和PPL选择, 可以修改配置

bit 6 未实现: 读为0

bit 5 **LOCK**: PLL锁定状态位 (只读)  
1 = 指示PLL处于锁定状态, 或PLL起振定时器延时结束  
0 = 指示PLL处于失锁状态, 起振定时器在进行延时或PLL被禁止

bit 4 未实现: 读为0

- 注 1: 对该寄存器进行写操作需要解锁序列 (见第9.10节“OSCCON解锁序列”)。
- 2: 不允许直接在使能PLL的任何主振荡器模式和FRCPLL模式之间进行时钟切换。(这指两者之间任意方向的时钟切换)。在这些情况下, 应用必须首先切换到FRC模式将其作为两个PLL模式之间的过渡时钟源。
- 3: 该位只能用软件清零。在软件中将该位置1 (= 1) 的效果与实际振荡器故障相同, 将触发振荡器故障陷阱。

## 寄存器 9-1: OSCCON: 振荡器控制寄存器<sup>(1)</sup> (续)

bit 3	<b>CF:</b> 时钟故障检测位 <sup>(3)</sup> 1 = FSCM 已检测到时钟故障 0 = FSCM 未检测到时钟故障
bit 2-1	<b>未实现:</b> 读为 0
bit 0	<b>OSWEN:</b> 振荡器切换使能位 1 = 请求振荡器切换为由 NOSC[2:0] 位指定的振荡器 0 = 已完成振荡器切换

- 注 1:** 对该寄存器进行写操作需要解锁序列 (见第 9.10 节 “OSCCON 解锁序列”)。
- 2:** 不允许直接在使能 PLL 的任何主振荡器模式和 FRCPLL 模式之间进行时钟切换。(这指两者之间任意方向的时钟切换)。在这些情况下, 应用必须首先切换到 FRC 模式将其作为两个 PLL 模式之间的过渡时钟源。
- 3:** 该位只能用软件清零。在软件中将该位置 1 (= 1) 的效果与实际振荡器故障相同, 将触发振荡器故障陷阱。

# dsPIC33CK64MP105系列

**寄存器 9-2: CLKDIV: 时钟分频比寄存器**

R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ROI	DOZE2 <sup>(1)</sup>	DOZE1 <sup>(1)</sup>	DOZE0 <sup>(1)</sup>	DOZEN <sup>(2,3)</sup>	FRCDIV2	FRCDIV1	FRCDIV0
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	—	PLLPRE3 <sup>(4)</sup>	PLLPRE2 <sup>(4)</sup>	PLLPRE1 <sup>(4)</sup>	PLLPRE0 <sup>(4)</sup>
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 15 ROI: 中断恢复位**  
 1 = 中断将清零 DOZEN 位, 并且处理器时钟与外设时钟之间的频率比被设置为 1:1  
 0 = 中断对 DOZEN 位没有影响

**bit 14-12 DOZE[2:0]: 处理器时钟分频比选择位<sup>(1)</sup>**  
 111 = FP 被 128 分频  
 110 = FP 被 64 分频  
 101 = FP 被 32 分频  
 100 = FP 被 16 分频  
 011 = FP 被 8 分频 (默认)  
 010 = FP 被 4 分频  
 001 = FP 被 2 分频  
 000 = FP 被 1 分频

**bit 11 DOZEN: 打盹模式使能位<sup>(2,3)</sup>**  
 1 = DOZE[2:0] 位域指定外设时钟与处理器时钟之间的频率比  
 0 = 处理器时钟与外设时钟之间的频率比被强制为 1:1

**bit 10-8 FRCDIV[2:0]: 内部快速 RC 振荡器后分频比位**  
 111 = FRC 被 256 分频  
 110 = FRC 被 64 分频  
 101 = FRC 被 32 分频  
 100 = FRC 被 16 分频  
 011 = FRC 被 8 分频  
 010 = FRC 被 4 分频  
 001 = FRC 被 2 分频  
 000 = FRC 被 1 分频 (默认)

**bit 7-6 未实现: 读为 0**

**bit 5-4 保留: 读为 0**

- 注 1:** 只有 DOZEN 位清零时, 才能写入 DOZE[2:0] 位。如果 DOZEN = 1, 则对 DOZE[2:0] 的任何写操作都会被忽略。  
**2:** 该位在 ROI 位置 1 且产生中断时清零。  
**3:** 如果 DOZE[2:0] = 000, 则 DOZEN 位不能置 1。如果 DOZE[2:0] = 000, 则用户软件将 DOZEN 位置 1 的任何尝试都会被忽略。  
**4:** 在 PLL 工作时, 可以更新 PLLPRE[3:0], 但 VCO 可能会出现过冲。



## 寄存器 9-2: CLKDIV: 时钟分频比寄存器 (续)

bit 3-0      **PLLPRE[3:0]:** PLL 相位检测器输入分频比选择位 (也表示为 “N1”, PLL 预分频比) <sup>(4)</sup>

11111 = 保留  
...  
1001 = 保留  
1000 = 输入被 8 分频  
0111 = 输入被 7 分频  
0110 = 输入被 6 分频  
0101 = 输入被 5 分频  
0100 = 输入被 4 分频  
0011 = 输入被 3 分频  
0010 = 输入被 2 分频  
0001 = 输入被 1 分频 (上电默认选择)  
0000 = 保留

- 注 1:** 只有DOZEN位清零时,才能写入DOZE[2:0]位。如果DOZEN = 1,则对DOZE[2:0]的任何写操作都会被忽略。
- 2:** 该位在ROI位置1且产生中断时清零。
- 3:** 如果DOZE[2:0] = 000,则DOZEN位不能置1。如果DOZE[2:0] = 000,则用户软件将DOZEN位置1的任何尝试都会被忽略。
- 4:** 在PLL工作时,可以更新PLLPRE[3:0],但VCO可能会出现过冲。

# dsPIC33CK64MP105系列

寄存器9-3: PLLFBD: PLL反馈分频比寄存器

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15				bit 8			
R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0
PLLFBDIV[7:0]							
bit 7				bit 0			

图注:	r = 保留位		
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 15-12      未实现: 读为0

bit 11-8      保留: 保持为0

bit 7-0      **PLLFBDIV[7:0]:** PLL 反馈分频比位 (也表示为 “M” , PLL 倍频比)

11111111 = 保留

...

11001000 = 200 (最大值) <sup>(1)</sup>

...

10010110 = 150 (默认值)

...

00010000 = 16 (最小值) <sup>(1)</sup>

...

00000010 = 保留

00000001 = 保留

00000000 = 保留

注 1: 允许的范围为16-200 (十进制)。其余为保留值, 应避免使用。上电时默认采用8 MHz FRC输入时钟, 默认反馈分频比为150 (十进制)。VCO频率为1.2 GHz。

**寄存器 9-4: OSCTUN: FRC振荡器调节寄存器**

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	TUN[5:0]					
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-6      **未实现:** 读为0

bit 5-0      **TUN[5:0]: FRC振荡器调节位**  
 011111 = 最大频率偏差为+1.45%  
 011110 = 中心频率+1.40%  
 ...  
 000001 = 中心频率+0.047%  
 000000 = 中心频率 (标称值 8.00 MHz)  
 111111 = 中心频率-0.047%  
 ...  
 100001 = 中心频率-1.45%  
 100000 = 最小频率偏差为-1.50%

# dsPIC33CK64MP105系列

寄存器9-5: PLLDIV: PLL输出分频比寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	VCODIV1	VCODIV0
bit 15						bit 8	

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	POST1DIV2 <sup>(1,2)</sup>	POST1DIV1 <sup>(1,2)</sup>	POST1DIV0 <sup>(1,2)</sup>	—	POST2DIV2 <sup>(1,2)</sup>	POST2DIV1 <sup>(1,2)</sup>	POST2DIV0 <sup>(1,2)</sup>
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 15-10 **未实现:** 读为0

bit 9-8 **VCODIV[1:0]:** PLL VCO输出分频比选择位

11 = Fvco  
10 = Fvco/2  
01 = Fvco/3  
00 = Fvco/4

bit 7 **未实现:** 读为0

bit 6-4 **POST1DIV[2:0]:** PLL输出分频比1位<sup>(1,2)</sup>

POST1DIV[2:0]的有效值为1到7 (POST1DIVx值应大于或等于POST2DIVx值)。POST1DIVx分频比被设计为以比POST2DIVx分频比更高的时钟速率工作。

bit 3 **未实现:** 读为0

bit 2-0 **POST2DIV[2:0]:** PLL输出分频比2位<sup>(1,2)</sup>

POST2DIV[2:0]的有效值为1到7 (POST2DIVx值应小于或等于POST1DIVx值)。POST1DIVx分频比被设计为以比POST2DIVx分频比更高的时钟速率工作。

注 1: 在PLL工作时, 不能更改POST1DIVx和POST2DIVx分频比值。

2: POST1DIVx和POST2DIVx的默认值分别为4和1, 产生150 MHz系统时钟源。

## 寄存器 9-6: ACLKCON1: 附属时钟控制寄存器

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
APLLEN <sup>(1)</sup>	APLLCK	—	—	—	—	—	FRCSEL
bit 15							bit 8

U-0	U-0	r-0	r-0	R/W-0	R/W-0	R/W-0	R/W-1
—	—	—	—	APLLPRE3	APLLPRE2	APLLPRE1	APLLPRE0
bit 7							bit 0

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

- bit 15 **APLLEN**: 附属PLL使能/旁路选择位<sup>(1)</sup>  
 1 = AFPLLO连接到APLL后分频器输出 (禁止旁路)  
 0 = AFPLLO连接到APLL输入时钟 (使能旁路)
- bit 14 **APLLCK**: APLL锁相状态位  
 1 = 附属PLL处于锁定状态  
 0 = 附属PLL不处于锁定状态
- bit 13-9 **未实现**: 读为0
- bit 8 **FRCSEL**: FRC时钟源选择位  
 1 = FRC是APLL的时钟源  
 0 = 主振荡器是APLL的时钟源
- bit 7-6 **未实现**: 读为0
- bit 5-4 **保留**: 保持为0
- bit 3-0 **APLLPRE[3:0]**: 附属PLL相位检测器输入分频比位  
 1111 = 保留  
 ...  
 1001 = 保留  
 1000 = 输入被8分频  
 0111 = 输入被7分频  
 0110 = 输入被6分频  
 0101 = 输入被5分频  
 0100 = 输入被4分频  
 0011 = 输入被3分频  
 0010 = 输入被2分频  
 0001 = 输入被1分频 (上电默认选择)  
 0000 = 保留

**注 1:** 即使APLLEN位已置1, 也必须有一个外设产生时钟请求, 然后APLL才能启动。

# dsPIC33CK64MP105系列

寄存器9-7:        **APLLFBD1: APLL反馈分频比寄存器**

U-0	U-0	U-0	U-0	r-0	r-0	r-0	r-0
—	—	—	—	—	—	—	—
bit 15				bit 8			
R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-1	R/W-0
APLLFBDIV[7:0]							
bit 7				bit 0			

图注:	r = 保留位		
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 15-12        **未实现:** 读为0

bit 11-8        **保留:** 保持为0

bit 7-0        **APLLFBDIV[7:0]: APLL 反馈分频比位**  
11111111 = 保留  
...  
11001000 = 200 (最大值) <sup>(1)</sup>  
...  
10010110 = 150 (默认值)  
...  
00010000 = 16 (最小值) <sup>(1)</sup>  
...  
00000010 = 保留  
00000001 = 保留  
00000000 = 保留

注 1: 允许的范围为16-200 (十进制)。其余为保留值, 应避免使用。上电时默认采用8 MHz FRC输入时钟, 默认反馈分频比为150 (十进制)。VCO频率为1.2 GHz。

## 寄存器 9-8: APLLDIV1: APLL 输出分频比寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	AVCODIV[1:0]	
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-1
—	APOST1DIV[2:0] <sup>(1,2)</sup>			—	APOST2DIV[2:0] <sup>(1,2)</sup>		
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 15-10 未实现: 读为0

bit 9-8 **AVCODIV[1:0]:** APLL VCO输出分频比选择位

11 = AFVCO

10 = AFVCO/2

01 = AFVCO/3

00 = AFVCO/4

bit 7 未实现: 读为0

bit 6-4 **APOST1DIV[2:0]:** APLL 输出分频比1位<sup>(1,2)</sup>

APOST1DIV[2:0]的有效值为1到7 (APOST1DIVx值应大于或等于APOST2DIVx值)。APOST1DIVx分频比被设计为以比APOST2DIVx分频比更高的时钟速率工作。

bit 3 未实现: 读为0

bit 2-0 **APOST2DIV[2:0]:** APLL 输出分频比2位<sup>(1,2)</sup>

APOST2DIV[2:0]的有效值为1到7 (APOST2DIVx值应小于或等于APOST1DIVx值)。APOST1DIVx分频比被设计为以比APOST2DIVx分频比更高的时钟速率工作。

注 1: 在PLL工作时, 不能更改APOST1DIVx和APOST2DIVx值。

2: APOST1DIVx和APOST2DIVx的默认值分别为4和1, 产生150 MHz系统时钟源。

# dsPIC33CK64MP105系列

## 寄存器 9-9: REFOCONL: 参考时钟控制低位字寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	HC/R/W-0	HSC/R-0
ROEN	—	ROSIDL	ROOUT	ROSLP	—	ROSWEN	ROACTIV
bit 15						bit 8	

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	ROSEL3	ROSEL2	ROSEL1	ROSEL0
bit 7							bit 0

图注:	HC = 硬件清零位	HSC = 硬件置 1/清零位
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 15 **ROEN:** 参考时钟使能位  
1 = 在 REFCLKO 引脚上使能参考振荡器  
0 = 禁止参考振荡器
- bit 14 **未实现:** 读为 0
- bit 13 **ROSIDL:** 空闲模式参考时钟停止位  
1 = 空闲模式下参考振荡器继续运行  
0 = 空闲模式下禁止参考振荡器
- bit 12 **ROOUT:** 参考时钟输出使能位  
1 = 使能参考时钟外部输出, 将其输出到 REFCLKO 引脚  
0 = 禁止参考时钟外部输出
- bit 11 **ROSLP:** 休眠模式参考时钟停止位  
1 = 休眠模式下参考振荡器继续运行  
0 = 休眠模式下禁止参考振荡器
- bit 10 **未实现:** 读为 0
- bit 9 **ROSWEN:** 参考时钟输出状态位  
1 = 已请求 (通过更改 RODIVx 来请求) 或正在进行时钟分频比更改 (由软件置 1, 完成时由硬件清零)  
0 = 时钟分频比更改已完成或不处于待处理状态
- bit 8 **ROACTIV:** 参考时钟状态位  
1 = 参考时钟在运行; 不要更改时钟源  
0 = 参考时钟已停止; 可以安全更改时钟源和配置
- bit 7-4 **未实现:** 读为 0
- bit 3-0 **ROSEL[3:0]:** 参考时钟源选择位  
1111 = 保留  
... = 保留  
1000 = 保留  
0111 = REFI 引脚  
0110 = Fvco/4  
0101 = BFRC  
0100 = LPRC  
0011 = FRC  
0010 = 主振荡器  
0001 = 外设时钟 (FP)  
0000 = 系统时钟 (Fosc)



## 寄存器 9-10: REFOCONH: 参考时钟控制高位字寄存器

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	RODIV[14:8]						
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
RODIV[7:0]							
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 15 未实现: 读为0

bit 14-0 **RODIV[14:0]:** 参考时钟整数分频比选择位

选定输入时钟源的分频比是所选值的两倍。

111 1111 1111 1111 = 基本时钟值被65,534分频 (2 \* 7FFFh)

111 1111 1111 1110 = 基本时钟值被65,532分频 (2 \* 7FFEh)

111 1111 1111 1101 = 基本时钟值被65,530分频 (2 \* 7FFDh)

...

000 0000 0000 0010 = 基本时钟值被4分频 (2 \* 2)

000 0000 0000 0001 = 基本时钟值被2分频 (2 \* 1)

000 0000 0000 0000 = 基本时钟值

## 寄存器 9-11: REFOTRIMH: 参考振荡器微调寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROTRIM[8:1]							
bit 15							bit 8
R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ROTRIM0	—	—	—	—	—	—	—
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 15-7 **ROTRIM[8:0]:** REFO微调位

这些位选择与1/2 REFO时钟周期对应的RODIV[14:0]值相加的小数分频比。

000000000 = 0/512 (将0.0分频比与RODIV[14:0]值相加)

000000001 = 1/512 (将0.001953125分频比与RODIV[14:0]值相加)

000000010 = 2/512 (将0.00390625分频比与RODIV[14:0]值相加)

...

100000000 = 256/512 (将0.5000分频比与RODIV[14:0]值相加)

...

111111110 = 510/512 (将0.99609375分频比与RODIV[14:0]值相加)

111111111 = 511/512 (将0.998046875分频比与RODIV[14:0]值相加)

bit 6-0 未实现: 读为0

注:

## 10.0 直接存储器访问（DMA）控制器

**注 1：** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“直接存储器访问控制器（DMA）”（[www.microchip.com/DS30009742](http://www.microchip.com/DS30009742)）。

直接存储器访问（DMA）控制器旨在为工作在SFR总线上的高数据吞吐量外设提供服务，使它们可以直接访问数据存储，并减少对CPU密集型管理的需求。通过使这些数据密集型外设共享自己的数据路径，也可降低主数据总线负载，从而进一步节省功耗。

DMA控制器用作CPU的外设和直接扩展。它位于单片机数据总线上，处于CPU与支持DMA的外设之间，可以直接访问SRAM。这将SFR总线划分为两条总线，使得DMA控制器可以访问新DMA SFR总线上支持DMA的外设。该控制器用作DMA SFR总线上的主器件，控制来自支持DMA的外设的数据流。

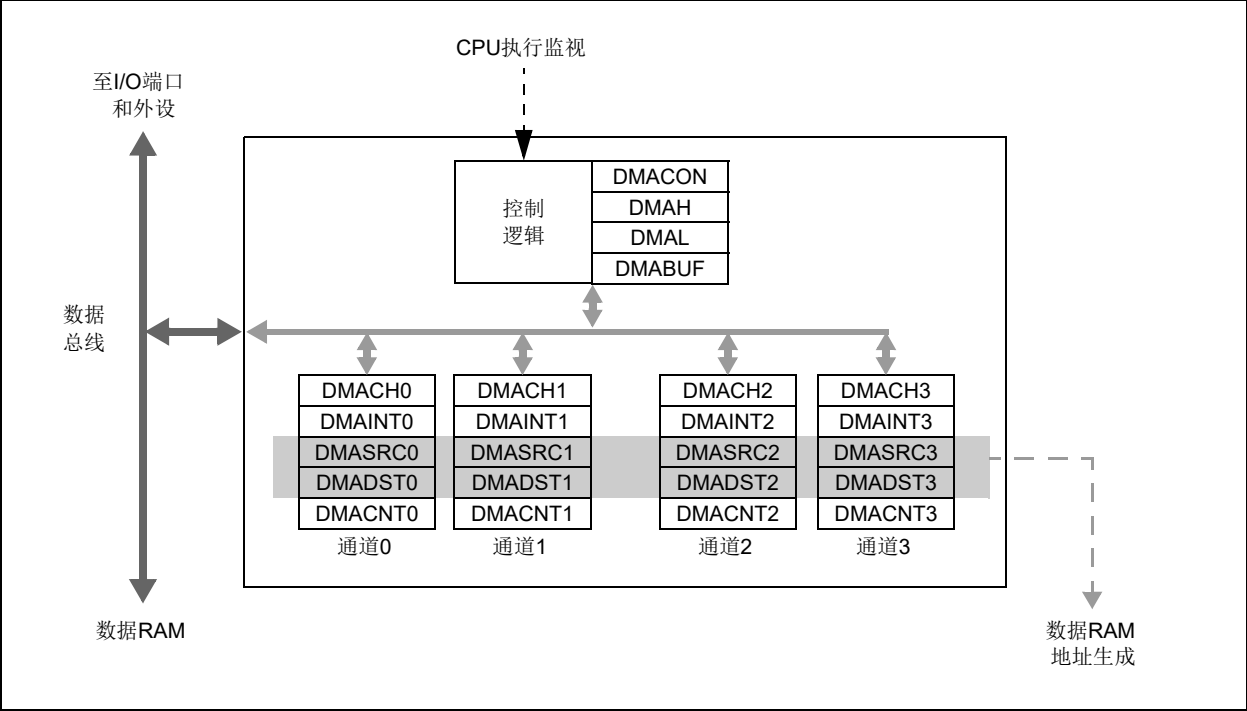
控制器还会直接监视CPU指令处理，借此了解CPU何时需要访问DMA总线上的外设，并根据需要自动将控制权让给CPU。这增大了用于处理数据的有效带宽，而DMA操作也不会造成处理器停止工作。因此，该控制器对于用户基本上是透明的。

DMA控制器具有以下特性：

- 四个独立可编程通道
- 与CPU并行工作（DMA不会引入等待状态）
- DMA总线仲裁
- 5种可编程地址模式
- 4种可编程传输模式
- 4种灵活的内部数据传输模式
- 支持按字节或字传输数据
- 每个通道均具有16位源地址和目标地址寄存器，可动态更新和重载
- 16位事务计数寄存器，可动态更新和重载
- 地址上限和地址下限寄存器
- 计数器半满中断
- 软件触发的传输
- 用于对称缓冲区操作的空写模式

图 10-1 给出了DMA控制器的简化框图。

图10-1: DMA功能框图



## 10.1 DMA操作概述

DMA控制器可根据多个不同参数在地址之间传送数据。对于任何事务，都可以单独配置其中的每个参数。此外，任何或所有DMA通道都可以在同一时间独立执行不同的事务。事务按以下参数分类：

- 源和目标（SFR和数据RAM）
- 数据大小（字节或字）
- 触发源
- 传输模式（单次、重复或连续）
- 寻址模式（固定地址或地址块，进行或不进行地址递增/递减）

此外，DMA控制器为所有通道都提供了通道优先级仲裁。

### 10.1.1 源和目标

使用DMA控制器，可以在数据空间中的任意两个地址之间传送数据。SFR空间（0000h至0FFFh）或数据RAM空间（1000h至2FFFh）可以用作源或目标。既可以在这些区域之间沿任一方向传送数据，也可以在任一区域的地址之间传送数据。图10-2给出了4种不同组合。

如果需要保护数据RAM区域，用户可通过DMA控制器为SFR空间之上的数据空间操作设置地址上限/下限边界。边界通过DMAH和DMAL限制寄存器设置。如果DMA通道尝试在地址边界之外执行操作，则会终止事务并产生中断。

### 10.1.2 数据大小

DMA控制器能够处理8位和16位事务。用户可以使用SIZE位（DMACHn[1]）来选择大小。默认情况下，每个通道都配置为字大小事务。当选择字节大小事务时，源和/或目标地址的LSB决定数据是表示数据RAM存储单元的高字节还是低字节。

### 10.1.3 触发源

DMA控制器可以使用器件的82个中断源来启动事务。DMA触发源的发生顺序与其自然中断优先级相反，如表10-1所示。

任何事务的源地址和目标地址都可以独立于触发源进行编程，因此DMA控制器可以使用任何触发信号对任何外设执行操作。这样一来，还可以级联多个DMA通道以执行更复杂的传输操作。

### 10.1.4 传输模式

基于每次触发所传输的数据量，DMA控制器支持4种数据传输类型。

- 单次：每次触发执行单个事务。
- 连续：每次触发执行一系列连续事务；事务的数量由DMACNTn事务计数器决定。
- 重复单次：重复执行单个事务，每次触发执行一次，直到DMA通道被禁止。
- 重复连续：重复执行一系列事务，每次触发执行一次，直到DMA通道被禁止。

所有传输模式都支持在事务完成后自动重载源地址和目标地址以及计数器值的选项。

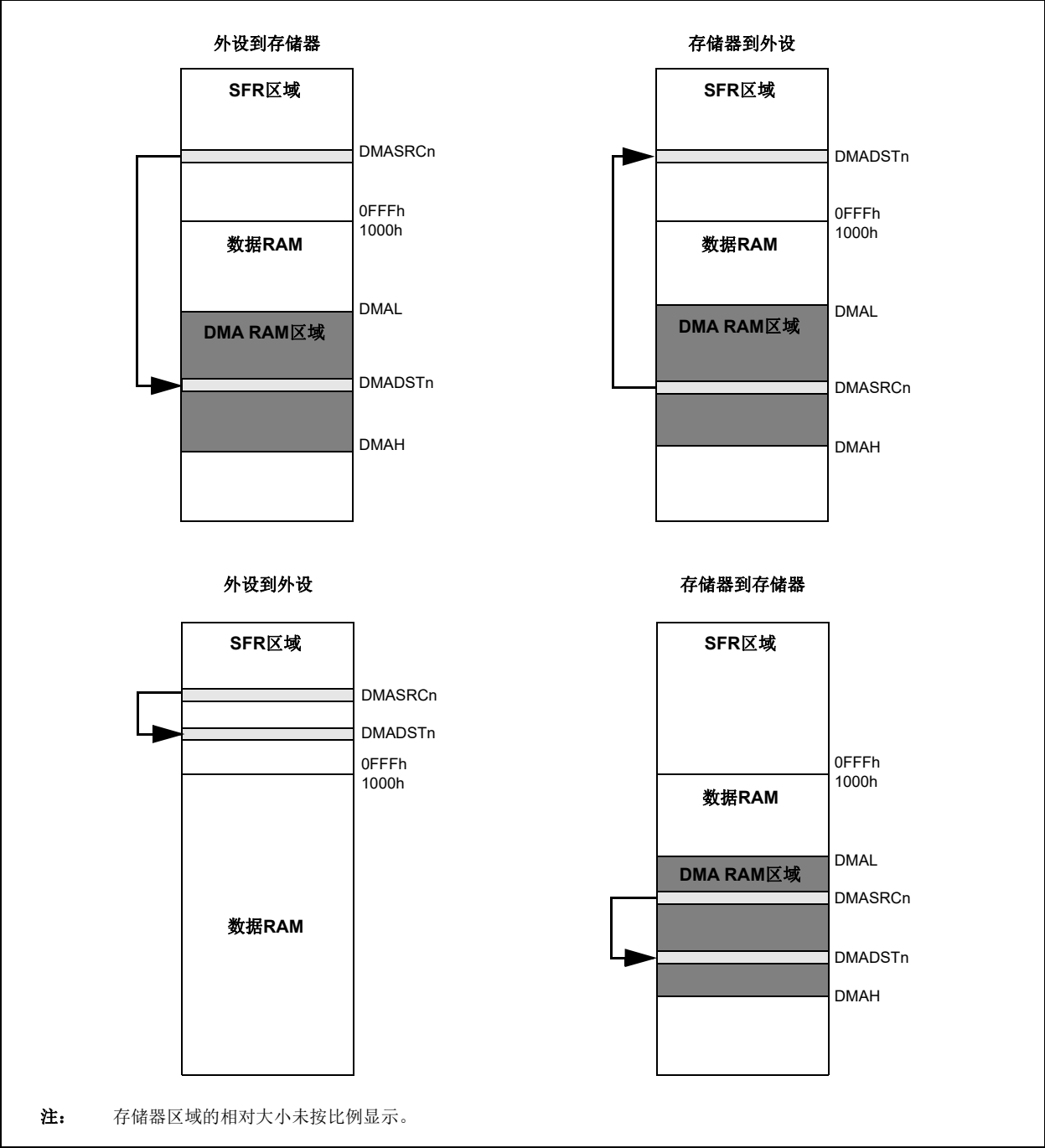
### 10.1.5 寻址模式

DMA控制器也支持单个地址或地址范围之间的传输。有以下4种基本模式可供选择：

- 固定地址到固定地址：2个固定地址之间
- 固定地址到地址块：从固定源地址到某一范围目标地址
- 地址块到固定地址：从某一范围源地址到单个固定目标地址
- 地址块到地址块：从某一范围源地址到另一范围目标地址

在块寻址模式下，可以选择自动递增或递减源地址和/或目标地址。

图10-2: DMA数据传输类型



## 10.1.6 通道优先级

各个DMA通道彼此之间独立工作，但在访问数据和DMA总线时会存在竞争。发生访问冲突时，DMA控制器会按照用户选择的优先级方案在通道之间进行仲裁。有以下两种方案可供选择：

- 轮转优先级：当有两个或多个通道第一次发生冲突时，编号最低的通道会获得优先权。在后续发生的冲突中，编号较高的通道会基于其通道编号依次获得优先权。
- 固定优先级：当有两个或多个通道发生冲突时，无论之前是否发生过冲突，编号最低的通道始终会获得优先权；但正在处理的通道均将无法立即重新触发。如果优先级较高的通道不断请求服务，则会将它安排在下一个待处理的较低优先级通道之后处理。

## 10.2 典型设置

设置DMA通道以进行基本数据传输：

1. 使能DMA控制器（DMAEN = 1），并通过置1或清零PRSEL来选择适当的通道优先级方案。
2. 对DMAH和DMAL进行编程，为数据RAM操作设置适当的地址上限/下限边界。
3. 选择要使用的DMA通道并禁止其操作（CHEN = 0）。
4. 将事务的适当源地址和目标地址编程到通道的DMASRCn和DMADSTn寄存器中。
5. 对DMACNTn寄存器进行编程，以设定每次传输的触发次数（单次或连续模式）或要传输的字（字节）数（重复模式）。
6. 将SIZE位置1或清零以选择数据大小。
7. 对TRMODE[1:0]位进行编程，以选择数据传输模式。
8. 对SAMODE[1:0]和DAMODE[1:0]位进行编程，以选择寻址模式。
9. 通过置1CHEN使能DMA通道。
10. 允许触发源中断。

## 10.3 外设模块禁止

可以使用外设模块禁止（PMD）寄存器为DMA控制器的各个通道单独掉电。

## 10.4 寄存器

DMA控制器使用多个寄存器来控制其操作。寄存器的数量取决于为具体器件实现的通道数量。

通常有4个模块级寄存器（1个控制寄存器和3个缓冲/地址寄存器）：

- DMACON：DMA引擎控制寄存器（[寄存器10-1](#)）
- DMAH和DMAL：DMA地址上限/下限寄存器
- DMABUF：DMA传输数据缓冲区

每个DMA通道实现5个寄存器（2个控制寄存器和3个缓冲/地址寄存器）：

- DMACHn：DMA通道n控制寄存器（[寄存器10-2](#)）
- DMAINTn：DMA通道n中断寄存器（[寄存器10-3](#)）
- DMASRCn：通道n的DMA数据源地址指针寄存器
- DMADSTn：通道n的DMA数据目标地址指针寄存器
- DMACNTn：通道n的DMA事务计数器寄存器

dsPIC33CK64MP105器件共有34个寄存器。

# dsPIC33CK64MP105系列

寄存器10-1: **DMACON: DMA引擎控制寄存器**

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
DMAEN	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PRSSEL
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 15 **DMAEN:** DMA模块使能位

1 = 使能模块

0 = 禁止模块并终止所有正在进行的DMA操作

bit 14-1 **未实现:** 读为0

bit 0 **PRSSEL:** 通道优先级方案选择位

1 = 轮转优先级方案

0 = 固定优先级方案



## 寄存器10-2: DMACHn: DMA通道n控制寄存器

U-0	U-0	U-0	r-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NULLW	RELOAD <sup>(1)</sup>	CHREQ <sup>(3)</sup>
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
SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	SIZE	CHEN
bit 7						bit 0	

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

bit 15-13	未实现: 读为0
bit 12	保留: 保持为0
bit 11	未实现: 读为0
bit 10	<b>NULLW:</b> 空写模式位 1 = 每次写入DMADSTn时, 启动对DMASRCn的虚拟写入 0 = 不启动虚拟写入
bit 9	<b>RELOAD:</b> 地址和计数重载位 <sup>(1)</sup> 1 = 在开始下一个操作时, DMASRCn、DMADSTn和DMACNTn寄存器重载它们先前的值 0 = 在开始下一个操作时, DMASRCn、DMADSTn和DMACNTn不重载 <sup>(2)</sup>
bit 8	<b>CHREQ:</b> DMA通道软件请求位 <sup>(3)</sup> 1 = 已由软件启动DMA请求; 在DMA传输结束时自动清除 0 = 没有DMA请求处于待处理状态
bit 7-6	<b>SAMODE[1:0]:</b> 源地址模式选择位 11 = 保留 10 = 传输完成后, DMASRCn根据SIZE位递减 01 = 传输完成后, DMASRCn根据SIZE位递增 00 = 传输完成后, DMASRCn保持不变
bit 5-4	<b>DAMODE[1:0]:</b> 目标地址模式选择位 11 = 保留 10 = 传输完成后, DMADSTn根据SIZE位递减 01 = 传输完成后, DMADSTn根据SIZE位递增 00 = 传输完成后, DMADSTn保持不变
bit 3-2	<b>TRMODE[1:0]:</b> 传输模式选择位 11 = 重复连续 10 = 连续 01 = 重复单次 00 = 单次
bit 1	<b>SIZE:</b> 数据大小选择位 1 = 字节 (8位) 0 = 字 (16位)
bit 0	<b>CHEN:</b> DMA通道使能位 1 = 使能相应通道 0 = 禁止相应通道

注 1: 只需存储DMACNTn的原始值即可恢复DMASRCn和DMADSTn的原始值。

2: DMACNTn在重复传输模式下始终会重载, 而与RELOAD位的状态无关。

3: CHREQ置1时执行的传输次数取决于TRMODE[1:0]的配置。

# dsPIC33CK64MP105系列

**寄存器10-3: DMAINTn: DMA通道n中断寄存器**

R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DBUFWF <sup>(1)</sup>	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
HIGHIF <sup>(1,2)</sup>	LOWIF <sup>(1,2)</sup>	DONEIF <sup>(1)</sup>	HALFIF <sup>(1)</sup>	OVRUNIF <sup>(1)</sup>	—	—	HALFEN
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 15      **DBUFWF:** DMA缓冲数据写标志位<sup>(1)</sup>  
1 = 在空写模式下, DMA缓冲区的内容尚未写入DMADSTn或DMASRCn中指定的地址单元  
0 = 在空写模式下, DMA缓冲区的内容已写入DMADSTn或DMASRCn中指定的地址单元
- bit 14-8    **CHSEL[6:0]:** DMA通道触发选择位  
有关完整列表, 请参见表10-1。
- bit 7      **HIGHIF:** DMA地址上限中断标志位<sup>(1,2)</sup>  
1 = DMA通道尝试访问高于DMAH (或数据RAM空间的上限) 的地址  
0 = DMA通道未调用地址上限中断
- bit 6      **LOWIF:** DMA地址下限中断标志位<sup>(1,2)</sup>  
1 = DMA通道尝试访问低于DMAL但高于SFR范围 (07FFh) 的DMA SFR地址  
0 = DMA通道未调用地址下限中断
- bit 5      **DONEIF:** DMA完成操作中中断标志位<sup>(1)</sup>  
如果CHEN = 1:  
1 = 前一个DMA会话已结束并完成传输  
0 = 当前DMA会话尚未完成传输  
如果CHEN = 0:  
1 = 前一个DMA会话已结束并完成传输  
0 = 前一个DMA会话已结束但未完成传输
- bit 4      **HALFIF:** DMA 50% 水印等级中断标志位<sup>(1)</sup>  
1 = DMACNTn已到达距离0000h的中点位置  
0 = DMACNTn未到达中点位置
- bit 3      **OVRUNIF:** DMA通道溢出标志位<sup>(1)</sup>  
1 = DMA通道仍在完成基于上一次触发的操作时被触发  
0 = 未发生溢出条件
- bit 2-1    **未实现:** 读为0
- bit 0      **HALFEN:** 完成一半水印位  
1 = 在DMACNTn到达其中点位置和传输完成时调用中断  
0 = 仅在传输完成时调用中断

**注 1:** 使用软件将这些标志置1不会产生中断。

**2:** 在实际访问之前未完成地址限制违反 (DMASRCn/DMADSTn大于DMAH或小于DMAL) 测试。

表10-1: DMA通道触发源

CHSEL[6:0]		触发信号（中断）	CHSEL[6:0]		触发信号（中断）	CHSEL[6:0]		触发信号（中断）
0	00h	INT0——外部中断0	33	21h	（保留，不要使用）	66	42h	AD1FLTR3——过采样滤波器3
1	01h	SCCP1 中断	34	22h		67	43h	AD1FLTR4——过采样滤波器4
2	02h	SPI1 接收器	35	23h		68	44h	CLC1 正边沿中断
3	03h	SPI1 发送器	36	24h	PWM 事件 C	69	45h	CLC2 正边沿中断
4	04h	UART1 接收器	37	25h	SENT1 TX/RX	70	46h	SPI1——故障中断
5	05h	UART1 发送器	38	26h	SENT2 TX/RX	71	47h	SPI2——故障中断
6	06h	ECC 单个位错误	39	27h	ADC 通用中断	72	48h	（保留，不要使用）
7	07h	NVM 写完成	40	28h	ADC 完成 AN0	...	...	
8	08h	INT1——外部中断1	41	29h	ADC 完成 AN1	86	56h	
9	09h	SI2C1——I2C1 从事件	42	2Ah	ADC 完成 AN2	87	57h	PWM 事件 D
10	0Ah	MI2C1——I2C1 主事件	43	2Bh	ADC 完成 AN3	88	58h	PWM 事件 E
11	0Bh	INT2——外部中断2	44	2Ch	ADC 完成 AN4	89	59h	PWM 事件 F
12	0Ch	SCCP2 中断	45	2Dh	ADC 完成 AN5	90	5Ah	（保留，不要使用）
13	0Dh	INT3——外部中断3	46	2Eh	ADC 完成 AN6	91	5Bh	
14	0Eh	UART2 接收器	47	2Fh	ADC 完成 AN7	92	5Ch	
15	0Fh	UART2 发送器	48	30h	ADC 完成 AN8	93	5Dh	
16	10h	SPI2 接收器	49	31h	ADC 完成 AN9	94	5Eh	
17	11h	SPI2 发送器	50	32h	ADC 完成 AN10	95	5Fh	
18	12h	SCCP3 中断	51	33h	ADC 完成 AN11	96	60h	CLC3 正边沿中断
19	13h	SI2C2——I2C2 从事件	52	34h	ADC 完成 AN12	97	61h	CLC4 正边沿中断
20	14h	MI2C2——I2C2 主事件	53	35h	ADC 完成 AN13	98	62h	SPI3 接收器
21	15h	SCCP4 中断	54	36h	ADC 完成 AN14	99	63h	SPI3 发送器
22	16h	MCCP5 中断	55	37h	ADC 完成 AN15	100	64h	SI2C3——I2C3 从事件
23	17h	（保留，不要使用）	56	38h	ADC 完成 AN16	101	65h	MI2C3——I2C3 主事件
24	18h	CRC 发生器中断	57	39h	ADC 完成 AN17	102	66h	SPI3 故障
25	19h	PWM 事件 A	58	3Ah	ADC 完成 AN18	103	67h	MCCP9
26	1Ah	（保留，不要使用）	59	3Bh	ADC 完成 AN19	104	68h	UART3 接收器
27	1Bh	PWM 事件 B	60	3Ch	ADC 完成 AN20	105	69h	UART3 发送器
28	1Ch	PWM 发生器 1	61	3Dh	（保留，不要使用）	106	6Ah	（保留，不要使用）
29	1Dh	PWM 发生器 2	62	3Eh		...	...	
30	1Eh	PWM 发生器 3	63	3Fh		127	7Fh	
31	1Fh	PWM 发生器 4	64	40h	AD1FLTR1——过采样滤波器 1			
32	20h	（保留，不要使用）	65	41h	AD1FLTR2——过采样滤波器 2			

注:

## 11.0 支持精细边沿定位的高分辨率PWM

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“支持精细边沿定位的高分辨率PWM”(www.microchip.com/DS70005320)。

高速PWM (High-Speed PWM, HSPWM) 模块是一个脉宽调制 (Pulse-Width Modulated, PWM) 模块, 支持电机控制和电源应用。这款灵活的模块功能丰富, 支持许多类型的电机控制 (Motor Control, MC) 和电源控制 (Power Control, PC) 应用, 其中包括:

- 交流-直流转换器
- 直流-直流转换器
- 交流和直流电机: BLDC、PMSM、ACIM和SRM等
- 逆变器
- 电池充电器
- 数字照明
- 功率因数校正 (PFC)

## 11.1 特性

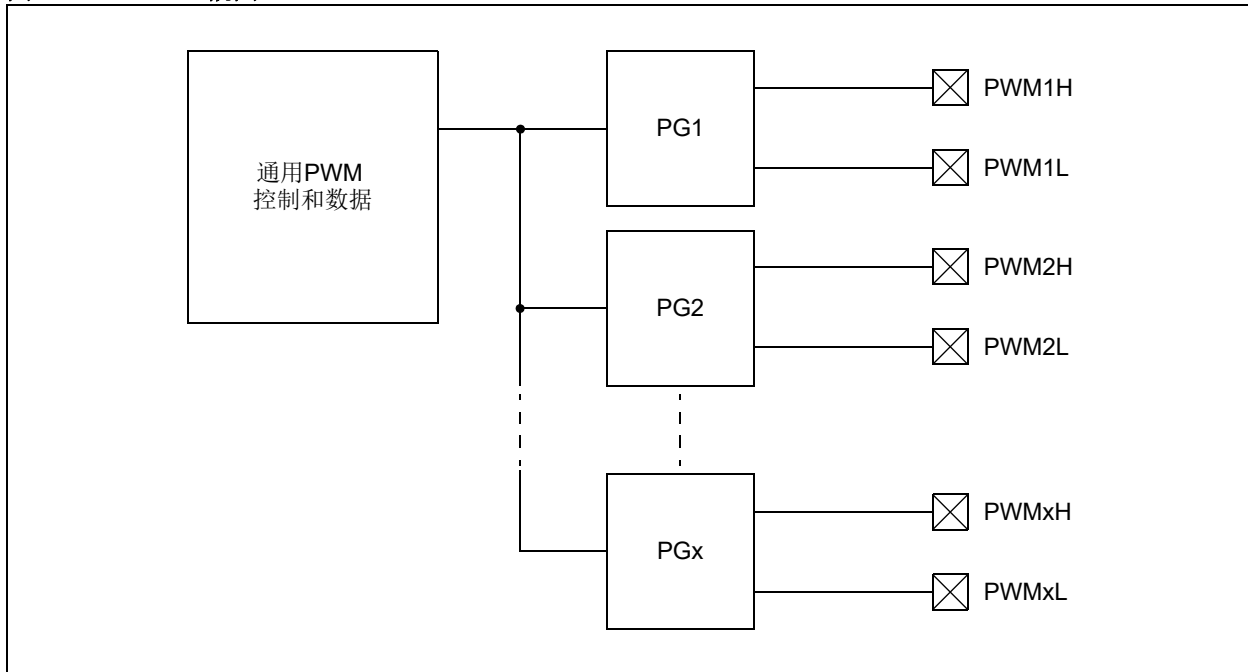
- 4个独立的PWM发生器, 每个发生器均配有双输出
- 工作模式:
  - 独立边沿模式
  - 可变相位PWM模式
  - 中心对齐模式
  - 双重更新中心对齐模式
  - 双边沿中心对齐模式
  - 双PWM模式
- 输出模式:
  - 互补
  - 独立
  - 推挽
- 死区发生器
- 前沿消隐 (Leading-Edge Blanking, LEB)
- 用于故障处理的输出改写
- 灵活的周期/占空比更新选项
- 可编程控制输入 (Programmable Control Input, PCI)
- 高级触发选项
- 六个组合逻辑输出
- 六个PWM事件输出

## 11.2 架构概述

PWM模块包括一组通用控制和功能，以及PWM发生器（PWM Generator, PG）的多个实例。可以独立配置

每个PWM发生器，或者将多个PWM发生器组合使用以实现复杂的多相系统。此外，PWM发生器还可用于实现复杂的触发、保护和逻辑功能。图11-1给出了概图。

图11-1: PWM概图



## 11.3 基于PPS的PWM4H输出

所有器件都能通过PPS将PWM4H信号输出到任何“RPn”引脚。此功能适用于PWM4H没有专用引脚的低引脚数器件。在具有固定RP65/PWM4H/RD1引脚的48引脚器件上使用PWM4H PPS输出功能，则输出信号将同时出现在专用引脚和“RPn”引脚上。PWM4L/H输出端口使能位PENH和PENL（PG4IOCONH[3:2]）控制专用引脚和PPS引脚；无法禁止专用引脚而仅使用PPS。

鉴于“RPn”功能的自然优先级高于PWM，可以使用专用RP65/PWM4H/RD1引脚的PPS输出功能，而将PWM4H信号通过PPS连接到其他引脚。

## 11.4 写限制

LOCK位（PCLKCON[8]）可通过软件置1，以阻止对某些寄存器的写操作。更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“支持精细边沿定位的高分辨率PWM”（[www.microchip.com/DS70005320](http://www.microchip.com/DS70005320)）。

要将LOCK位置1或清零，需要以下锁定/解锁序列。

1. 将0x55写入NVMKEY。
2. 将0xAA写入NVMKEY。
3. 通过单个操作将LOCK位（PCLKCON[8]）清零（或置1）。

通常，不应在模块运行时（由ON位（PGxCONL[15]）置1指示）对配置控制进行修改。

## 11.5 控制寄存器

可使用以下两种类型特殊功能寄存器（SFR）控制 PWM 模块的操作：

- 通用类型，由所有 PWM 发生器共用
- 针对特定 PWM 发生器的类型

寄存器名称中的“x”表示 PWM 发生器的实例。

寄存器名称中的“y”表示通用功能的实例。

**寄存器 11-1: PCLKCON: PWM 时钟控制寄存器**

R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
HRRDY	HRERR	—	—	—	—	—	LOCK <sup>(1)</sup>
bit 15							bit 8
U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	DIVSEL1	DIVSEL0	—	—	MCLKSEL1 <sup>(2,3)</sup>	MCLKSEL0 <sup>(2,3)</sup>
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 15 **HRRDY**: 高分辨率就绪位  
1 = 高分辨率电路已就绪  
0 = 高分辨率电路未就绪
- bit 14 **HRERR**: 高分辨率错误位  
1 = 发生了错误: PWM 信号的分辨率将受限  
0 = 未发生错误: HRRDY = 1 时 PWM 信号将具有全分辨率
- bit 13-9 **未实现**: 读为 0
- bit 8 **LOCK**: 锁定位<sup>(1)</sup>  
1 = 锁定写保护寄存器和位  
0 = 解锁写保护寄存器和位
- bit 7-6 **未实现**: 读为 0
- bit 5-4 **DIVSEL[1:0]**: PWM 时钟分频比选择位  
11 = 分频比为 1:16  
10 = 分频比为 1:8  
01 = 分频比为 1:4  
00 = 分频比为 1:2
- bit 3-2 **未实现**: 读为 0
- bit 1-0 **MCLKSEL[1:0]**: PWM 主时钟选择位<sup>(2,3)</sup>  
11 = AFPLLO——附属 PLL 后分频器输出  
10 = FPLLO——主 PLL 后分频器输出  
01 = AFVCO/2——附属 VCO/2  
00 = FOSC

**注 1:** LOCK 位受到保护，可防止意外写入。要将该位置 1，必须将 0x55 和 0xAA 值按顺序写入 NVMKEY 寄存器（见第 11.4 节“写限制”）。

**2:** 不建议在 ON (PGxCONL[15]) = 1 时更改 MCLKSEL[1:0] 位。

**3:** 在标准分辨率模式下，MCLKSEL[1:0] 位选择的 PWM 输入时钟频率不得超过 500 MHz，在高分辨率模式下则必须为 500 MHz。

# dsPIC33CK64MP105系列

## 寄存器 11-2: FSC: 频率调节寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSC[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
FSC[7:0]							
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 15-0 **FSC[15:0]: 频率调节寄存器位**

在每个PWM时钟将该寄存器中的值加到频率调节累加器中。当累加值超过FSMINPER值时, 会产生一个时钟脉冲。

## 寄存器 11-3: FSMINPER: 频率调节最小周期寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSMINPER[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
FSMINPER[7:0]							
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 15-0 **FSMINPER[15:0]: 频率调节最小周期寄存器位**

该寄存器保存频率调节电路可产生的最小时钟周期（最大时钟频率）。



## 寄存器 11-4: MPHASE: 主相位寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPHASE[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
MPHASE[7:0]							
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 15-0

**MPHASE[15:0]:** 主相位寄存器位

该寄存器保存可由多个PWM发生器共用的相位偏移值。

## 寄存器 11-5: MDC: 主占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MDC[15:8] <sup>(1)</sup>							
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
MDC[7:0] <sup>(1)</sup>							
bit 7				bit 0			

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-0

**MDC[15:0]:** 主占空比寄存器位<sup>(1)</sup>

该寄存器保存可由多个PWM发生器共用的占空比值。

**注 1:** 不得使用小于0x0008的占空比值（高分辨率模式下为0x0020）。

# dsPIC33CK64MP105系列

寄存器 11-6:      **MPER: 主周期寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
MPER[15:8] <sup>(1)</sup>							
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
MPER[7:0] <sup>(1)</sup>							
bit 7							bit 0

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

bit 15-0      **MPER[15:0]: 主周期寄存器位<sup>(1)</sup>**  
该寄存器保存可由多个PWM发生器共用的周期值。

注 1:    不得使用小于 0x0010 的周期值 (高分辨率模式下为 0x0080)。

**寄存器 11-7: CMBTRIGL: 组合触发低位字寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CTA4EN	CTA3EN	CTA2EN	CTA1EN
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 15-4      **未实现:** 读为0

bit 3      **CTA4EN:** 使能来自PWM发生器4的触发输出作为组合触发A的源位  
1 = 使能指定的触发信号作为组合触发A信号的逻辑或输入  
0 = 禁止

bit 2      **CTA3EN:** 使能来自PWM发生器3的触发输出作为组合触发A的源位  
1 = 使能指定的触发信号作为组合触发A信号的逻辑或输入  
0 = 禁止

bit 1      **CTA2EN:** 使能来自PWM发生器2的触发输出作为组合触发A的源位  
1 = 使能指定的触发信号作为组合触发A信号的逻辑或输入  
0 = 禁止

bit 0      **CTA1EN:** 使能来自PWM发生器1的触发输出作为组合触发A的源位  
1 = 使能指定的触发信号作为组合触发A信号的逻辑或输入  
0 = 禁止

# dsPIC33CK64MP105系列

寄存器 11-8: CMBTRIGH: 组合触发高位字寄存器

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

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	CTB4EN	CTB3EN	CTB2EN	CTB1EN
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 15-4

未实现: 读为0
- bit 3

**CTB4EN:** 使能来自PWM发生器4的触发输出作为组合触发B的源位  
1 = 使能指定的触发信号作为组合触发B信号的逻辑或输入  
0 = 禁止
- bit 2

**CTB3EN:** 使能来自PWM发生器3的触发输出作为组合触发B的源位  
1 = 使能指定的触发信号作为组合触发B信号的逻辑或输入  
0 = 禁止
- bit 1

**CTB2EN:** 使能来自PWM发生器2的触发输出作为组合触发B的源位  
1 = 使能指定的触发信号作为组合触发B信号的逻辑或输入  
0 = 禁止
- bit 0

**CTB1EN:** 使能来自PWM发生器1的触发输出作为组合触发B的源位  
1 = 使能指定的触发信号作为组合触发B信号的逻辑或输入  
0 = 禁止

**寄存器11-9: LOGCONy: 组合PWM逻辑控制寄存器<sup>(2)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PWMS1y3 <sup>(1)</sup>	PWMS1y2 <sup>(1)</sup>	PWMS1y1 <sup>(1)</sup>	PWMS1y0 <sup>(1)</sup>	PWMS2y3 <sup>(1)</sup>	PWMS2y2 <sup>(1)</sup>	PWMS2y1 <sup>(1)</sup>	PWMS2y0 <sup>(1)</sup>
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
S1yPOL	S2yPOL	PWMLFy1	PWMLFy0	—	PWMLFyD2 <sup>(3)</sup>	PWMLFyD1 <sup>(3)</sup>	PWMLFyD0 <sup>(3)</sup>
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 15-12 **PWMS1y[3:0]:** 组合PWM逻辑源1选择位<sup>(1)</sup>

1111-1000 = 保留  
0111 = PWM4L  
0110 = PWM4H  
0101 = PWM3L  
0100 = PWM3H  
0011 = PWM2L  
0010 = PWM2H  
0001 = PWM1L  
0000 = PWM1H

bit 11-8 **PWMS2y[3:0]:** 组合PWM逻辑源2选择位<sup>(1)</sup>

1111-1000 = 保留  
0111 = PWM4L  
0110 = PWM4H  
0101 = PWM3L  
0100 = PWM3H  
0011 = PWM2L  
0010 = PWM2H  
0001 = PWM1L  
0000 = PWM1H

bit 7 **S1yPOL:** 组合PWM逻辑源1极性位

1 = 输入反相  
0 = 输入为正逻辑

bit 6 **S2yPOL:** 组合PWM逻辑源2极性位

1 = 输入反相  
0 = 输入为正逻辑

bit 5-4 **PWMLFy[1:0]:** 组合PWM逻辑功能选择位

11 = 保留  
10 = PWMS1y ^ PWMS2y (异或)  
01 = PWMS1y & PWMS2y (与)  
00 = PWMS1y | PWMS2y (或)

bit 3 **未实现:** 读为0

**注 1:** 如果所选PWM通道不存在, 则逻辑功能输入将连接到0。

**2:** “y”表示通用实例(A-F)。

**3:** 当LOGCONy的y = A、C或E时, 将逻辑功能输出分配给PWMxH引脚。当LOGCONy的y = B、D或F时, 将逻辑功能分配给PWMxL引脚。

# dsPIC33CK64MP105系列

---

寄存器11-9: LOGCONy: 组合PWM逻辑控制寄存器y<sup>(2)</sup> (续)

bit 2-0 PWMLFyD[2:0]: 组合PWM逻辑目标选择位<sup>(3)</sup>  
111-100 = 保留  
011 = 逻辑功能分配给PWM4H或PWM4L引脚  
010 = 逻辑功能分配给PWM3H或PWM3L引脚  
001 = 逻辑功能分配给PWM2H或PWM2L引脚  
000 = 不分配, 禁止组合PWM逻辑功能

注 1: 如果所选PWM通道不存在, 则逻辑功能输入将连接到0。

2: “y”表示通用实例(A-F)。

3: 当LOGCONy的y = A、C或E时, 将逻辑功能输出分配给PWMxH引脚。当LOGCONy的y = B、D或F时, 将逻辑功能分配给PWMxL引脚。

寄存器11-10: **PWMEVTy: PWM事件输出控制寄存器<sup>(5)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
EVTyOEN	EVTyPOL	EVTySTRD	EVTySYNC	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
EVTySEL3	EVTySEL2	EVTySEL1	EVTySEL0	—	EVTyPGS2 <sup>(2)</sup>	EVTyPGS1 <sup>(2)</sup>	EVTyPGS0 <sup>(2)</sup>
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 15 **EVTyOEN:** PWM事件输出使能位  
1 = 事件输出信号输出到PWMEy引脚  
0 = 事件输出信号仅限内部使用
- bit 14 **EVTyPOL:** PWM事件输出极性位  
1 = 事件输出信号为低电平有效  
0 = 事件输出信号为高电平有效
- bit 13 **EVTySTRD:** PWM事件输出信号延长禁止位  
1 = 不延长事件输出信号的脉冲宽度  
0 = 将事件输出信号延长到至少8个PWM时钟周期<sup>(1)</sup>
- bit 12 **EVTySYNC:** PWM事件输出同步位  
1 = 事件输出信号与系统时钟同步  
0 = 事件输出信号与系统时钟不同步  
该位置1且EVTySTRD = 1时, 事件输出信号脉冲将为两个系统时钟。
- bit 11-8 **未实现:** 读为0
- bit 7-4 **EVTySEL[3:0]:** PWM事件选择位  
1111 = 高分辨率错误事件信号  
1110-1010 = 保留  
1001 = ADC触发2信号  
1000 = ADC触发1信号  
0111 = STEER信号 (仅适用于推挽输出模式)<sup>(4)</sup>  
0110 = CAHALF信号 (仅适用于中心对齐模式)<sup>(4)</sup>  
0101 = PCI故障有效输出信号  
0100 = PCI限流有效输出信号  
0011 = PCI前馈有效输出信号  
0010 = PCI同步有效输出信号  
0001 = PWM发生器输出信号<sup>(3)</sup>  
0000 = 通过PGTRGSEL[2:0]位选择的源
- bit 3 **未实现:** 读为0

注 1: 使用peripheral\_clk延长事件信号, 因为不同的PWM发生器可能使用不同的时钟源进行工作。

2: 如果所选PWM发生器不存在, 则不会产生任何事件。

3: 此信号为进入输出模式逻辑和任何输出改写逻辑之前的PWM发生器输出信号。

4: 此信号应为同步到系统时钟域之前的PGx\_clk域信号。

5: “y”表示通用实例(A-F)。

# dsPIC33CK64MP105系列

---

寄存器11-10: **PWMEVTy**: PWM事件输出控制寄存器<sup>(5)</sup> (续)

bit 2-0      **EVTyPGS[2:0]**: PWM事件源选择位<sup>(2)</sup>  
111-100 = 保留  
011 = PWM发生器4  
...  
000 = PWM发生器1

- 注 1:** 使用 `peripheral_clk` 延长事件信号，因为不同的PWM发生器可能使用不同的时钟源进行工作。
- 2:** 如果所选PWM发生器不存在，则不会产生任何事件。
- 3:** 此信号为进入输出模式逻辑和任何输出改写逻辑之前的PWM发生器输出信号。
- 4:** 此信号应为同步到系统时钟域之前的PGx\_clk域信号。
- 5:** “y”表示通用实例(A-F)。



寄存器 11-11:     **LFSR: 线性反馈移位寄存器**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	LFSR[14:8]						
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
LFSR[7:0]							
bit 7							bit 0

<b>图注:</b>			
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 15     **未实现:** 读为0

bit 14-0   **LFSR[14:0]:** 线性反馈移位寄存器位  
            读取该寄存器将得到一个**15**位伪随机值。

# dsPIC33CK64MP105系列

寄存器11-12: PGxCONL: PWM发生器x控制低位字寄存器

R/W-0	r-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ON	—	—	—	—	TRGCNT2	TRGCNT1	TRGCNT0
bit 15						bit 8	

R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HREN <sup>(2)</sup>	—	—	CLKSEL1	CLKSEL0	MODSEL2	MODSEL1	MODSEL0
bit 7							bit 0

图注: r = 保留位  
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 15 **ON:** 使能位  
1 = 使能PWM发生器  
0 = 禁止PWM发生器
- bit 14 **保留:** 保持为0
- bit 13-11 **未实现:** 读为0
- bit 10-8 **TRGCNT[2:0]:** 触发计数选择位  
111 = PWM发生器在触发后产生8个PWM周期  
110 = PWM发生器在触发后产生7个PWM周期  
101 = PWM发生器在触发后产生6个PWM周期  
100 = PWM发生器在触发后产生5个PWM周期  
011 = PWM发生器在触发后产生4个PWM周期  
010 = PWM发生器在触发后产生3个PWM周期  
001 = PWM发生器在触发后产生2个PWM周期  
000 = PWM发生器在触发后产生1个PWM周期
- bit 7 **HREN:** PWM发生器x高分辨率使能位<sup>(2)</sup>  
1 = PWM发生器x工作在高分辨率模式下  
0 = PWM发生器x工作在标准分辨率模式下
- bit 6-5 **未实现:** 读为0
- bit 4-3 **CLKSEL[1:0]:** 时钟选择位  
11 = PWM发生器使用主时钟经频率调节电路调节得到的时钟<sup>(1)</sup>  
10 = PWM发生器使用主时钟经时钟分频电路分频得到的时钟<sup>(1)</sup>  
01 = PWM发生器使用由MCLKSEL[1:0] (PCLKCON[1:0]) 控制位选择的主时钟  
00 = 未选择时钟, PWM发生器处于最低功耗状态 (默认)
- bit 2-0 **MODSEL[2:0]:** 模式选择位  
111 = 双边沿中心对齐PWM模式 (每周产生中断/寄存器更新两次)  
110 = 双边沿中心对齐PWM模式 (每周产生中断/寄存器更新一次)  
101 = 双重更新中心对齐PWM模式  
100 = 中心对齐PWM模式  
011 = 保留  
010 = 独立边沿PWM模式, 双输出  
001 = 可变相位PWM模式  
000 = 独立边沿PWM模式

注 1: PWM发生器时基采用频率调节电路时钟, 可有效地对PWM发生器输出的占空比和周期进行调节。

2: 高分辨率模式必须使用500 MHz的输入频率。

## 寄存器11-13: PGxCONH: PWM发生器x控制高位字寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
MDCSEL	MPERSEL	MPHSEL	—	MSTEN	UPDMOD2	UPDMOD1	UPDMOD0
bit 15						bit 8	

r-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TRGMOD	—	—	SOCS3 <sup>(1,2,3)</sup>	SOCS2 <sup>(1,2,3)</sup>	SOCS1 <sup>(1,2,3)</sup>	SOCS0 <sup>(1,2,3)</sup>
bit 7						bit 0	

图注:	r = 保留位
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 15 **MDCSEL:** 主占空比寄存器选择位

- 1 = PWM发生器使用MDC寄存器
- 0 = PWM发生器使用PGxDC寄存器

bit 14 **MPERSEL:** 主周期寄存器选择位

- 1 = PWM发生器使用MPER寄存器
- 0 = PWM发生器使用PGxPER寄存器

bit 13 **MPHSEL:** 主相位寄存器选择位

- 1 = PWM发生器使用MPHASE寄存器
- 0 = PWM发生器使用PGxPHASE寄存器

bit 12 **未实现:** 读为0

bit 11 **MSTEN:** 主更新使能位

- 1 = PWM发生器向其他PWM发生器广播UPDREQ状态位的状态（通过软件置1/清零）和EOC信号
- 0 = PWM发生器不广播UPDREQ状态位的状态和EOC信号

bit 10-8 **UPDMOD[2:0]:** PWM缓冲区更新模式选择位

011 = 应请求立即更新

当接收到主更新请求时, 会立即或尽快更新数据寄存器。对于发出请求的PWM发生器, 如果MSTEN = 1且UPDREQ = 1, 则会发送主更新请求。

010 = 应请求SOC更新

当接收到主更新请求时, 会在下一个周期开始时更新数据寄存器。对于发出请求的PWM发生器, 如果MSTEN = 1且UPDREQ = 1, 则会发送主更新请求。

001 = 立即更新

如果UPDREQ = 1, 会立即或尽快更新数据寄存器。发生更新后, UPDATE状态位将自动清零。

000 = SOC更新

如果UPDREQ = 1, 在下一个PWM周期开始时更新数据寄存器。发生更新后, UPDATE状态位将自动清零。<sup>(1)</sup>

bit 7 **保留:** 保持为0

注 1: 如果使能了PCI同步功能, 则PCI选择的同步信号始终可与SOCS[3:0]位选择的SOC信号进行逻辑或。

2: SOCS[3:0]位选择的源必须与本地PWM发生器使用相同的时钟源。否则, 源必须经由PCI同步逻辑, 以便触发信号能够同步到PWM发生器时钟域。

3: PWM发生器四个为一组: PG1-PG4和PG5-PG8（如果可用）。一组四个中的任一发生器都可用来触发同一组中的另一个发生器。

# dsPIC33CK64MP105系列

---

## 寄存器11-13: PGxCONH: PWM发生器x控制高位字寄存器 (续)

bit 6	<b>TRGMOD:</b> PWM发生器触发模式选择位 1 = PWM发生器工作在可重触发模式下 0 = PWM发生器工作在单触发模式下
bit 5-4	<b>未实现:</b> 读为0
bit 3-0	<b>SOCS[3:0]:</b> 周期开始选择位 <b>(1,2,3)</b> 1111 = 仅 TRIG 位或 PCI 同步功能 (不选择硬件触发源) 1110-0101 = 保留 0100 = 通过 PG4 PGTRGSEL[2:0] 位 (PGxEVTL[2:0]) 选择的触发输出 0011 = 通过 PG3 PGTRGSEL[2:0] 位 (PGxEVTL[2:0]) 选择的触发输出 0010 = 通过 PG2 PGTRGSEL[2:0] 位 (PGxEVTL[2:0]) 选择的触发输出 0001 = 通过 PG1 PGTRGSEL[2:0] 位 (PGxEVTL[2:0]) 选择的触发输出 0000 = 本地 EOC——PWM发生器自触发

- 注 1:** 如果使能了 PCI 同步功能, 则 PCI 选择的同步信号始终可与 SOCS[3:0] 位选择的 SOC 信号进行逻辑或。
- 2:** SOCS[3:0] 位选择的源必须与本地 PWM 发生器使用相同的时钟源。否则, 源必须经由 PCI 同步逻辑, 以便触发信号能够同步到 PWM 发生器时钟域。
- 3:** PWM 发生器四个为一组: PG1-PG4 和 PG5-PG8 (如果可用)。一组四个中的任一发生器都可用来触发同一组中的另一个发生器。

## 寄存器11-14: PGxSTAT: PWM发生器x状态寄存器

HS/C-0	HS/C-0	HS/C-0	HS/C-0	R-0	R-0	R-0	R-0
SEVT	FLTEVT	CLEVT	FFEVT	SACT	FLTACT	CLACT	FFACT
bit 15				bit 8			

W-0	W-0	HS/R/W-0	R-0	W-0	R-0	R-0	R-0
TRSET	TRCLR	CAP <sup>(1)</sup>	UPDATE	UPDREQ	STEER	CAHALF	TRIG
bit 7				bit 0			

<b>图注:</b>	C = 可清零位	HS = 硬件置1位	
R = 可读位	W = 可写位	0 = 清零	x = 未知
-n = POR时的值	1 = 置1	U = 未实现位, 读为0	

- bit 15 **SEVT:** PCI同步事件位  
1 = 发生了PCI同步事件 (使能模块时, PCI同步输出上出现上升沿, 或者PCI同步输出为高电平)  
0 = 未发生PCI同步事件
- bit 14 **FLTEVT:** PCI故障有效状态位  
1 = 发生了故障事件 (使能模块时, PCI故障输出上出现上升沿, 或者PCI故障输出为高电平)  
0 = 未发生故障事件
- bit 13 **CLEVT:** PCI限流状态位  
1 = 发生了PCI限流事件 (使能模块时, PCI限流输出上出现上升沿, 或者PCI限流输出为高电平)  
0 = 未发生PCI限流事件
- bit 12 **FFEVT:** PCI前馈有效状态位  
1 = 发生了PCI前馈事件 (使能模块时, PCI前馈输出上出现上升沿, 或者PCI前馈输出为高电平)  
0 = 未发生PCI前馈事件
- bit 11 **SACT:** PCI同步状态位  
1 = PCI同步输出有效  
0 = PCI同步输出无效
- bit 10 **FLTACT:** PCI故障有效状态位  
1 = PCI故障输出有效  
0 = PCI故障输出无效
- bit 9 **CLACT:** PCI限流状态位  
1 = PCI限流输出有效  
0 = PCI限流输出无效
- bit 8 **FFACT:** PCI前馈有效状态位  
1 = PCI前馈输出有效  
0 = PCI前馈输出无效
- bit 7 **TRSET:** PWM发生器软件触发置1位  
用户软件向此位写1可触发PWM发生器周期。此位始终读为0。触发PWM发生器时, TRIG位将指示1。
- bit 6 **TRCLR:** PWM发生器软件触发清零位  
用户软件向此位写1可停止PWM发生器周期。此位始终读为0。未触发PWM发生器时, TRIG位将指示0。

**注 1:** 用户软件可向CAP写1以请求启动软件捕捉。发生捕捉事件时, CAP状态位将置1。在CAP由软件清零之前, 不会发生后续捕捉。

# dsPIC33CK64MP105系列

---

## 寄存器11-14: PGxSTAT: PWM发生器x状态寄存器 (续)

bit 5	<b>CAP:</b> 捕捉状态位 <sup>(1)</sup> 1 = 已在PGxCAP中捕捉到PWM发生器时基值 0 = 未发生捕捉
bit 4	<b>UPDATE:</b> PWM数据寄存器更新状态位 1 = PWM数据寄存器更新处于待处理状态, 用户数据寄存器不可写 0 = 没有PWM数据寄存器更新处于待处理状态
bit 3	<b>UPDREQ:</b> PWM数据寄存器更新请求位 用户软件向此位写1可请求一次PWM数据寄存器更新。此位始终读为0。当更新处于待处理状态时, UPDATE状态位将指示1。
bit 2	<b>STEER:</b> 输出转向状态位 (仅限推挽输出模式) 1 = PWM发生器处于推挽模式的第2周期 0 = PWM发生器处于推挽模式的第1周期
bit 1	<b>CAHALF:</b> 半周期状态位 (仅限中心对齐模式) 1 = PWM发生器处于时基周期的下半周期 0 = PWM发生器处于时基周期的上半周期
bit 0	<b>TRIG:</b> PWM触发状态位 1 = PWM发生器已触发, PWM周期正在进行 0 = PWM周期不在进行

**注 1:** 用户软件可向CAP写1以请求启动软件捕捉。发生捕捉事件时, CAP状态位将置1。在CAP由软件清零之前, 不会发生后续捕捉。

## 寄存器11-15: PGxIOCONL: PWM发生器x I/O控制低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CLMOD	SWAP	OVRENH	OVRENL	OVRDAT1	OVRDAT0	OSYNC1	OSYNC0
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
FLTDAT1	FLTDAT0	CLDAT1	CLDAT0	FFDAT1	FFDAT0	DBDAT1	DBDAT0
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 15 **CLMOD**: 限流模式选择位

1 = 如果PCI限流处于有效状态, 则PWMxH和PWMxL输出信号反相(位翻转), 不使用CLDAT[1:0]位  
0 = 如果PCI限流处于有效状态, 则CLDAT[1:0]位定义PWM输出电平

bit 14 **SWAP**: 交换PWMxH和PWMxL器件引脚的PWM信号位

1 = PWMxH信号连接至PWMxL引脚, PWMxL信号连接至PWMxH引脚  
0 = PWMxH/L信号映射到它们各自对应的引脚

bit 13 **OVRENH**: PWMxH引脚的用户改写使能位

1 = OVRDAT1为PWMxH引脚的输出提供数据  
0 = PWM发生器为PWMxH引脚提供数据

bit 12 **OVRENL**: PWMxL引脚的用户改写使能位

1 = OVRDAT0为PWMxL引脚的输出提供数据  
0 = PWM发生器为PWMxL引脚提供数据

bit 11-10 **OVRDAT[1:0]**: 使能改写时PWMxH/PWMxL引脚数据位

如果OVRENH = 1, 则OVRDAT1为PWMxH提供数据。  
如果OVRENL = 1, 则OVRDAT0为PWMxL提供数据。

bit 9-8 **OSYNC[1:0]**: 用户输出改写同步控制位

11 = 保留  
10 = 何时通过OVRENH/L和OVRDAT[1:0]位进行用户输出改写由PGxCONH寄存器中的UPDMOD[2:0]位指定  
01 = 立即(尽快)通过OVRENH/L和OVRDAT[1:0]位进行用户输出改写  
00 = 通过OVRENH/L和OVRDAT[1:0]位进行的用户输出改写与本地PWM时基同步(下一周期开始时进行)

bit 7-6 **FLTDAT[1:0]**: 故障事件有效时PWMxH/PWMxL引脚数据位

如果故障有效, 则FLTDAT1为PWMxH提供数据。  
如果故障有效, 则FLTDAT0为PWMxL提供数据。

bit 5-4 **CLDAT[1:0]**: 限流事件有效时PWMxH/PWMxL引脚数据位

如果限流有效, 则CLDAT1为PWMxH提供数据。  
如果限流有效, 则CLDAT0为PWMxL提供数据。

bit 3-2 **FFDAT[1:0]**: 前馈事件有效时PWMxH/PWMxL引脚数据位

如果前馈有效, 则FFDAT1为PWMxH提供数据。  
如果前馈有效, 则FFDAT0为PWMxL提供数据。

bit 1-0 **DBDAT[1:0]**: 调试模式有效时PWMxH/PWMxL引脚数据位

如果调试模式有效且器件暂停, 则DBDAT1为PWMxH提供数据。  
如果调试模式有效且器件暂停, 则DBDAT0为PWMxL提供数据。

# dsPIC33CK64MP105系列

寄存器11-16: PGxIOCONH: PWM发生器x I/O控制高位字寄存器

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
—	CAPSRC2 <sup>(1)</sup>	CAPSRC1 <sup>(1)</sup>	CAPSRC0 <sup>(1)</sup>	—	—	—	DTCMPSEL
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	PMOD1	PMOD0	PENH	PENL	POLH	POLL
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 15 未实现: 读为0

bit 14-12 **CAPSRC[2:0]:** 时基捕捉源选择位<sup>(1)</sup>

111 = 保留

110 = 保留

101 = 保留

100 = 将所选PCI故障信号置为有效时捕捉时基值

011 = 将所选PCI限流信号置为有效时捕捉时基值

010 = 将所选PCI前馈信号置为有效时捕捉时基值

001 = 将所选PCI同步信号置为有效时捕捉时基值

000 = 没有为时基捕捉选择硬件源——仅使用软件

bit 11-9 未实现: 读为0

bit 8 **DTCMPSEL:** 死区补偿选择位

1 = 死区补偿由PCI前馈限制逻辑控制

0 = 死区补偿由PCI同步逻辑控制

bit 7-6 未实现: 读为0

bit 5-4 **PMOD[1:0]:** PWM发生器输出模式选择位

11 = 保留

10 = PWM发生器输出工作在推挽模式下

01 = PWM发生器输出工作在独立模式下

00 = PWM发生器输出工作在互补模式下

bit 3 **PENH:** PWMxH输出端口使能位

1 = PWM发生器控制PWMxH输出引脚

0 = PWM发生器不控制PWMxH输出引脚

bit 2 **PENL:** PWMxL输出端口使能位

1 = PWM发生器控制PWMxL输出引脚

0 = PWM发生器不控制PWMxL输出引脚

bit 1 **POLH:** PWMxH输出极性位

1 = 输出引脚为低电平有效

0 = 输出引脚为高电平有效

bit 0 **POLL:** PWMxL输出极性位

1 = 输出引脚为低电平有效

0 = 输出引脚为高电平有效

注 1: 随时可通过软件向CAP (PGxSTAT[5]) 写入1来启动捕捉。



## 寄存器11-17: PGxEVTL: PWM发生器x事件低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADTR1PS4	ADTR1PS3	ADTR1PS2	ADTR1PS1	ADTR1PS0	ADTR1EN3	ADTR1EN2	ADTR1EN1
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	UPDTRG1	UPDTRG0	PGTRGSEL2 <sup>(1)</sup>	PGTRGSEL1 <sup>(1)</sup>	PGTRGSEL0 <sup>(1)</sup>
bit 7							bit 0

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-11 **ADTR1PS[4:0]:** ADC触发1后分频比选择位

11111 = 1:32

...

00010 = 1:3

00001 = 1:2

00000 = 1:1

bit 10 **ADTR1EN3:** ADC触发1源为PGxTRIGC比较事件使能位

1 = 使能PGxTRIGC寄存器比较事件作为ADC触发1的触发源

0 = 禁止PGxTRIGC寄存器比较事件作为ADC触发1的触发源

bit 9 **ADTR1EN2:** ADC触发1源为PGxTRIGB比较事件使能位

1 = 使能PGxTRIGB寄存器比较事件作为ADC触发1的触发源

0 = 禁止PGxTRIGB寄存器比较事件作为ADC触发1的触发源

bit 8 **ADTR1EN1:** ADC触发1源为PGxTRIGA比较事件使能位

1 = 使能PGxTRIGA寄存器比较事件作为ADC触发1的触发源

0 = 禁止PGxTRIGA寄存器比较事件作为ADC触发1的触发源

bit 7-5 **未实现:** 读为0bit 4-3 **UPDTRG[1:0]:** 更新触发源选择位

11 = 对PGxTRIGA寄存器进行写操作会自动将UPDATE位置1

10 = 对PGxPHASE寄存器进行写操作会自动将UPDATE位置1

01 = 对PGxDC寄存器进行写操作会自动将UPDATE位置1

00 = 用户必须手动将UPDREQ位(PGxSTAT[3])置1

bit 2-0 **PGTRGSEL[2:0]:** PWM发生器触发输出选择位<sup>(1)</sup>

111 = 保留

110 = 保留

101 = 保留

100 = 保留

011 = PGxTRIGC比较事件为PWM发生器触发输出

010 = PGxTRIGB比较事件为PWM发生器触发输出

001 = PGxTRIGA比较事件为PWM发生器触发输出

000 = EOC事件为PWM发生器触发输出

**注 1:** 这些事件源自内部PWM发生器时基比较事件。

# dsPIC33CK64MP105系列

## 寄存器 11-18: PGxEVTH: PWM发生器x事件高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
FLTIE <sup>(1)</sup>	CLIE <sup>(2)</sup>	FFIE <sup>(3)</sup>	SIEN <sup>(4)</sup>	—	—	IEVTSEL1	IEVTSEL0
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
ADTR2EN3	ADTR2EN2	ADTR2EN1	ADTR1OFS4	ADTR1OFS3	ADTR1OFS2	ADTR1OFS1	ADTR1OFS0
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 15      **FLTIE**: PCI故障中断允许位<sup>(1)</sup>  
             1 = 允许故障中断  
             0 = 禁止故障中断
- bit 14      **CLIE**: PCI限流中断允许位<sup>(2)</sup>  
             1 = 允许限流中断  
             0 = 禁止限流中断
- bit 13      **FFIE**: PCI前馈中断允许位<sup>(3)</sup>  
             1 = 允许前馈中断  
             0 = 禁止前馈中断
- bit 12      **SIEN**: PCI同步中断允许位<sup>(4)</sup>  
             1 = 允许同步中断  
             0 = 禁止同步中断
- bit 11-10   **未实现**: 读为0
- bit 9-8      **IEVTSEL[1:0]**: 中断事件选择位  
             11 = 禁止时基中断 (同步、故障、限流和前馈事件可单独使能)  
             10 = 在发生ADC触发1事件时中断CPU  
             01 = 在发生TRIGA比较事件时中断CPU  
             00 = 在EOC时中断CPU
- bit 7      **ADTR2EN3**: ADC触发2源为PGxTRIGC比较事件使能位  
             1 = 使能PGxTRIGC寄存器比较事件作为ADC触发2的触发源  
             0 = 禁止PGxTRIGC寄存器比较事件作为ADC触发2的触发源
- bit 6      **ADTR2EN2**: ADC触发2源为PGxTRIGB比较事件使能位  
             1 = 使能PGxTRIGB寄存器比较事件作为ADC触发2的触发源  
             0 = 禁止PGxTRIGB寄存器比较事件作为ADC触发2的触发源
- bit 5      **ADTR2EN1**: ADC触发2源为PGxTRIGA比较事件使能位  
             1 = 使能PGxTRIGA寄存器比较事件作为ADC触发2的触发源  
             0 = 禁止PGxTRIGA寄存器比较事件作为ADC触发2的触发源
- bit 4-0      **ADTR1OFS[4:0]**: ADC触发1偏移选择位  
             11111 = 偏移31个触发事件  
             ...  
             00010 = 偏移2个触发事件  
             00001 = 偏移1个触发事件  
             00000 = 无偏移

- 注 1: 仅在PCI故障有效信号的上升沿产生中断。  
 2: 仅在PCI限流有效信号的上升沿产生中断。  
 3: 仅在PCI前馈有效信号的上升沿产生中断。  
 4: 仅在PCI同步有效信号的上升沿产生中断。

**寄存器 11-19: PGxyPCIL: PWM发生器xy PCI低位字寄存器 (x = PWM发生器编号; y = F、CL、FF或S)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TSYNCDIS	TERM2	TERM1	TERM0	AQPS	AQSS2	AQSS1	AQSS0
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
SWTERM	PSYNC	PPS	PSS4	PSS3	PSS2	PSS1	PSS0
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 15 TSYNCDIS:** 终止同步禁止位

1 = 立即终止锁存的PCI

0 = 在PWM EOC时终止锁存的PCI

**bit 14-12 TERM[2:0]:** 终止事件选择位

111 = 选择PCI源9

110 = 选择PCI源8

101 = 选择PCI源1 (PWM发生器输出由PWMPCI[2:0]位选择)

100 = PGxTRIGC触发事件

011 = PGxTRIGB触发事件

010 = PGxTRIGA触发事件

001 = 自动终止: PCI源从有效变为无效时终止

000 = 手动终止: 向SWTERM位写入1时终止

**bit 11 AQPS:** 接收限定符极性选择位

1 = 反相

0 = 不反相

**bit 10-8 AQSS[2:0]:** 接收限定符源选择位

111 = 仅SWPCI控制位 (限定符强制为0)

110 = 选择PCI源9

101 = 选择PCI源8

100 = 选择PCI源1 (PWM发生器输出由PWMPCI[2:0]位选择)

011 = 触发PWM发生器

010 = LEB有效

001 = 占空比有效 (基本PWM发生器信号)

000 = 不使用接收限定符 (限定符强制为1)

**bit 7 SWTERM:** PCI软件终止位

向此位写入1将产生终止事件。此位始终读为0。

**bit 6 PSYNC:** PCI同步控制位

1 = PCI源与PWM EOC同步

0 = PCI源不与PWM EOC同步

**bit 5 PPS:** PCI极性选择位

1 = 反相

0 = 不反相

# dsPIC33CK64MP105系列

---

**寄存器 11-19: PGxyPCIL: PWM发生器xy PCI低位字寄存器 (x = PWM发生器编号; y = F、CL、FF或S) (续)**

bit 4-0      **PSS[4:0]:** PCI源选择位

- 11111 = CLC1
- 11110 = 保留
- 11101 = 比较器3输出
- 11100 = 比较器2输出
- 11011 = 比较器1输出
- 11010 = PWM事件D
- 11001 = PWM事件C
- 11000 = PWM事件B
- 10111 = PWM事件A
- 10110 = 器件引脚, PCI[22]
- 10101 = 器件引脚, PCI[21]
- 10100 = 器件引脚, PCI[20]
- 10011 = 器件引脚, PCI[19]
- 10010 = RPn输入, PCI18R
- 10001 = RPn输入, PCI17R
- 10000 = RPn输入, PCI16R
- 01111 = RPn输入, PCI15R
- 01110 = RPn输入, PCI14R
- 01101 = RPn输入, PCI13R
- 01100 = RPn输入, PCI12R
- 01011 = RPn输入, PCI11R
- 01010 = RPn输入, PCI10R
- 01001 = RPn输入, PCI9R
- 01000 = RPn输入, PCI8R
- 00111 = 保留
- 00110 = 保留
- 00101 = 保留
- 00100 = 保留
- 00011 = 在内部连接到组合触发B
- 00010 = 在内部连接到组合触发A
- 00001 = 在内部连接到PWMPCI[2:0]多路开关的输出
- 00000 = 连接到0

**寄存器 11-20: PGxyPCIH: PWM发生器xy PCI高位字寄存器 (x = PWM发生器编号; y = F、CL、FF或S)**

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
BPEN	BPSEL2 <sup>(1)</sup>	BPSEL1 <sup>(1)</sup>	BPSEL0 <sup>(1)</sup>	—	ACP2	ACP1	ACP0
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
SWPCI	SWPCIM1	SWPCIM0	LATMOD	TQPS	TQSS2	TQSS1	TQSS0
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 15 **BPEN:** PCI旁路使能位

1 = 使能PCI功能, 旁路本地PCI逻辑; PWM发生器将由BPSEL[2:0]位选择的PWM发生器中的PCI功能控制

0 = 不旁路PCI功能

bit 14-12 **BPSEL[2:0]:** PCI旁路源选择位<sup>(1)</sup>

111-100 = 保留

011 = BPEN = 1时由PWM发生器4 PCI逻辑进行PCI控制

010 = BPEN = 1时由PWM发生器3 PCI逻辑进行PCI控制

001 = BPEN = 1时由PWM发生器2 PCI逻辑进行PCI控制

000 = BPEN = 1时由PWM发生器1 PCI逻辑进行PCI控制

bit 11 **未实现:** 读为0

bit 10-8 **ACP[2:0]:** PCI接收标准选择位

111 = 保留

110 = 保留

101 = 任意边沿锁存

100 = 上升沿锁存

011 = 锁存

010 = 任意边沿

001 = 上升沿

000 = 电平敏感

bit 7 **SWPCI:** 软件PCI控制位

1 = 使用1来驱动由SWPCIM[1:0]控制位分配的PCI逻辑

0 = 使用0来驱动由SWPCIM[1:0]控制位分配的PCI逻辑

bit 6-5 **SWPCIM[1:0]:** 软件PCI控制模式位

11 = 保留

10 = 将SWPCI位分配至终止限定符逻辑

01 = 将SWPCI位分配至接收限定符逻辑

00 = 将SWPCI位分配至PCI接收逻辑

bit 4 **LATMOD:** PCI SR锁存器模式位

1 = SR锁存器在锁存接收模式下是复位优先型的

0 = SR锁存器在锁存接收模式下是置1优先型的

bit 3 **TQPS:** 终止限定符极性选择位

1 = 反相

0 = 不反相

**注 1:** 如果所选PWM发生器不存在, 则选择0。

# dsPIC33CK64MP105系列

---

寄存器 11-20: **PGxyPCIH: PWM发生器xy PCI高位字寄存器** (x = PWM发生器编号; y = F、CL、FF或S) (续)

bit 2-0      **TQSS[2:0]:** 终止限定符源选择位

- 111 = 仅SWPCI控制位 (限定符强制为0)
- 110 = 选择PCI源9
- 101 = 选择PCI源8
- 100 = 选择PCI源1 (由PWMPCI[2:0]位选择的PWM发生器输出)
- 011 = 触发PWM发生器
- 010 = LEB有效
- 001 = 占空比有效 (基本PWM发生器信号)
- 000 = 不使用终止限定符 (限定符强制为1)

**注 1:** 如果所选PWM发生器不存在, 则选择0。

寄存器11-21: PGxLEBL: PWM发生器x前沿消隐低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LEB[15:8]							
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-0
LEB[7:0] <sup>(1)</sup>							
bit 7				bit 0			

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

bit 15-0      **LEB[15:0]:** 前沿消隐周期位<sup>(1)</sup>  
前沿消隐周期。不使用消隐时间的低3位, 提供8个时钟周期的消隐分辨率。最小消隐周期为8个时钟周期(当LEB[15:3] = 0时)。

注 1: Bit[2:0]是只读的, 始终保持为0。

# dsPIC33CK64MP105系列

寄存器11-22: PGxLEBH: PWM发生器x前沿消隐高位字寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	PWMPCI2 <sup>(1)</sup>	PWMPCI1 <sup>(1)</sup>	PWMPCI0 <sup>(1)</sup>
bit 15					bit 8		

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PHR	PHF	PLR	PLF
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 15-11 未实现: 读为0

bit 10-8 **PWMPCI[2:0]**: PCI的PWM源选择位<sup>(1)</sup>

111-100 = 保留

011 = 将PWM发生器4输出用于PCI逻辑

010 = 将PWM发生器3输出用于PCI逻辑

001 = 将PWM发生器2输出用于PCI逻辑

000 = 将PWM发生器1输出用于PCI逻辑

bit 7-4 未实现: 读为0

bit 3 **PHR**: PWMxH上升沿触发使能位

1 = PWMxH的上升沿将触发LEB持续时间计数器

0 = LEB忽略PWMxH的上升沿

bit 2 **PHF**: PWMxH下降沿触发使能位

1 = PWMxH的下降沿将触发LEB持续时间计数器

0 = LEB忽略PWMxH的下降沿

bit 1 **PLR**: PWMxL上升沿触发使能位

1 = PWMxL的上升沿将触发LEB持续时间计数器

0 = LEB忽略PWMxL的上升沿

bit 0 **PLF**: PWMxL下降沿触发使能位

1 = PWMxL的下降沿将触发LEB持续时间计数器

0 = LEB忽略PWMxL的下降沿

**注 1:** 所选PWM发生器源不影响LEB计数器。该源可选择用作PCI输入、PCI限定符、PCI终止符或PCI终止符限定符（更多信息, 请参见[寄存器 11-19](#)和[寄存器 11-20](#)中的说明）。



## 寄存器11-23: PGxPHASE: PWM发生器x相位寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPHASE[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
PGxPHASE[7:0]							
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 15-0 PGxPHASE[15:0]: PWM发生器x相位寄存器位

## 寄存器11-24: PGxDC: PWM发生器x占空比寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxDC[15:8] <sup>(1)</sup>							
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
PGxDC[7:0] <sup>(1)</sup>							
bit 7				bit 0			

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-0 PGxDC[15:0]: PWM发生器x占空比寄存器位<sup>(1)</sup>

注 1: 不得使用小于0x0008的占空比值（高分辨率模式下为0x0020）。

# dsPIC33CK64MP105系列

## 寄存器11-25: PGxDCA: PWM发生器x占空比调整寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
PGxDCA[7:0]							
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 15-8      **未实现:** 读为0

bit 7-0      **PGxDCA[7:0]:** PWM发生器x占空比调整值位  
 根据所选PCI源的状态, 将PGxDC寄存器中的值与PGxDCA值相加, 以形成有效占空比。当PCI源有效时, 与PGxDCA相加。

## 寄存器11-26: PGxPER: PWM发生器x周期寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PGxPER[15:8] <sup>(1)</sup>							
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
PGxPER[7:0] <sup>(1)</sup>							
bit 7							bit 0

### 图注:

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

bit 15-0      **PGxPER[15:0]:** PWM发生器x周期寄存器位<sup>(1)</sup>

**注 1:** 不得使用小于0x0010的周期值 (高分辨率模式下为0x0080)。

# dsPIC33CK64MP105系列

## 寄存器11-27: PGxTRIGA: PWM发生器x触发A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
PGxTRIGA[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
PGxTRIGA[7:0]							
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 15-0 PGxTRIGA[15:0]: PWM发生器x触发A寄存器位

## 寄存器11-28: PGxTRIGB: PWM发生器x触发B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
PGxTRIGB[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
PGxTRIGB[7:0]							
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 15-0 PGxTRIGB[15:0]: PWM发生器x触发B寄存器位

## 寄存器11-29: PGxTRIGC: PWM发生器x触发C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
PGxTRIGC[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
PGxTRIGC[7:0]							
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 15-0 PGxTRIGC[15:0]: PWM发生器x触发C寄存器位

# dsPIC33CK64MP105系列

寄存器11-30: PGxDTL: PWM发生器x死区低位字寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTL[13:8] <sup>(1)</sup>					
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
DTL[7:0]							
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 15-14    未实现: 读为0

bit 13-0    DTL[13:0]: PWMxL死区延时位<sup>(1)</sup>

注 1: 当HREN (PGxCONL[7]) = 0时, DTL[13:11]位不可用。

寄存器11-31: PGxDTH: PWM发生器x死区高位字寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DTH[13:8] <sup>(1)</sup>					
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
DTH[7:0]							
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 15-14    未实现: 读为0

bit 13-0    DTH[13:0]: PWMxH死区延时位<sup>(1)</sup>

注 1: 当HREN (PGxCONL[7]) = 0时, DTH[13:11]位不可用。

寄存器11-32: PGxCAP: PWM发生器x捕捉寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PGxCAP[15:8]							
bit 15							
bit 8							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PGxCAP[7:0] <sup>(1)</sup>							
bit 7							
bit 0							

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

bit 15-0 PGxCAP[15:0]: PGx时基捕捉位<sup>(1)</sup>

注 1: 在标准分辨率模式下, PGxCAP[1:0]读为0。在高分辨率模式下, PGxCAP[4:0]读为0。

注:

## 12.0 高速12位模数转换器（ADC）

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“12位高速多SAR A/D转换器（ADC）”（[www.microchip.com/DS70005213](http://www.microchip.com/DS70005213)）。

**2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。关于具体器件的寄存器和位信息，请参见本数据手册中的第4.0节“存储器构成”。

dsPIC33CK64MP105器件具有一个高速12位模数转换器（Analog-to-Digital Converter, ADC），该模数转换器具有低转换延时、高分辨率和过采样功能，可提高在交流/直流和直流/直流电源转换器应用中的性能。器件实现了带三个SAR内核（两个专用内核和一个共用内核）的ADC。

### 12.1 ADC特性概述

高速12位多SAR模数转换器（ADC）包括以下特性：

- 三个ADC内核：两个专用内核和一个共用内核
- 对于每个内核，用户可配置的分辨率最高可达12位
- 12位分辨率时每个通道的转换速率最高为3.5 Msps
- 低延时转换
- 最多21个模拟输入通道，每个输入通道具有独立的16位转换结果寄存器
- 对于所有通道中的每个通道，转换结果可以采用无符号或有符号数据格式

- 可同时采样最多3个模拟输入
- 通道扫描功能
- 每个内核支持多个转换触发选项，包括：
  - 来自CPU内核的PWM触发信号
  - M CCP/SCCP 模块触发信号
  - CLC 模块触发信号
  - 外部引脚触发事件（ADTRG31）
  - 软件触发信号
- 四个具有专用中断的集成数字比较器：
  - 多个比较选项
  - 可分配给特定的模拟输入
- 四个具有专用中断的过采样滤波器：
  - 提供更高的分辨率
  - 可分配给特定的模拟输入

模块包含3个独立的SAR ADC内核。图12-1和图12-2给出了12位多SAR ADC的简化框图。

模拟输入（通道）通过多路开关和开关连接到每个ADC内核的采样保持（Sample-and-Hold, S&H）电路。内核使用通道信息（输出格式、测量模式和输入编号）来处理模拟采样。转换完成时，结果将存储到特定模拟输入的结果缓冲区中，并在数字滤波器和数字比较器配置为使用此特定通道的数据时传递给二者。

ADC模块每次最多可采样5个输入（4个输入来自专用SAR内核，1个输入来自共用SAR内核）。如果共用内核上有多个ADC输入请求转换，模块将从编号最低的输入开始按顺序进行转换。

ADC使每个模拟输入能够指定其自身的触发源。该功能使ADC可以采样和转换与依靠独立时基工作的PWM发生器关联的模拟输入。

图12-1： ADC模块框图

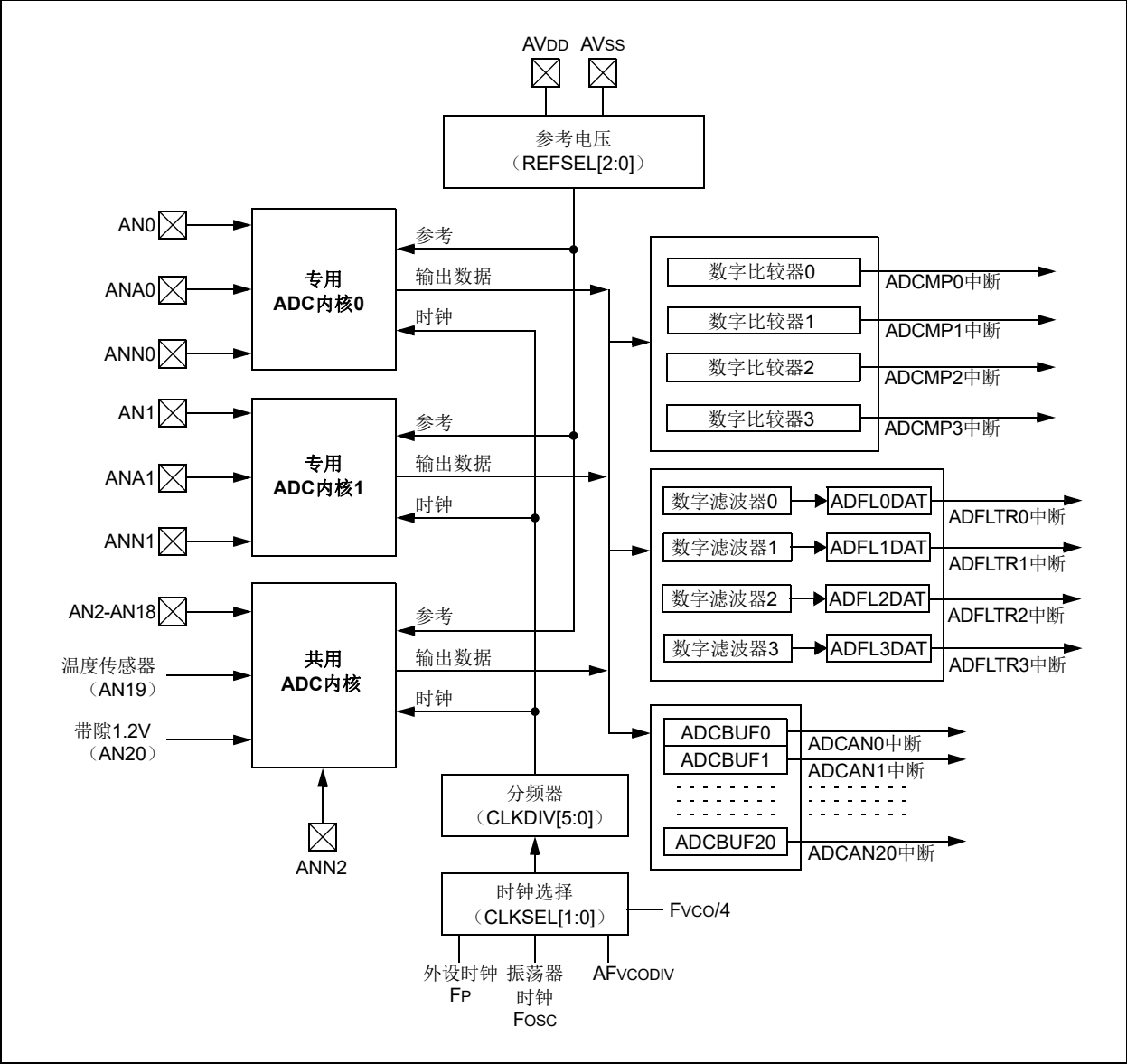




图12-2: ADC共用内核框图

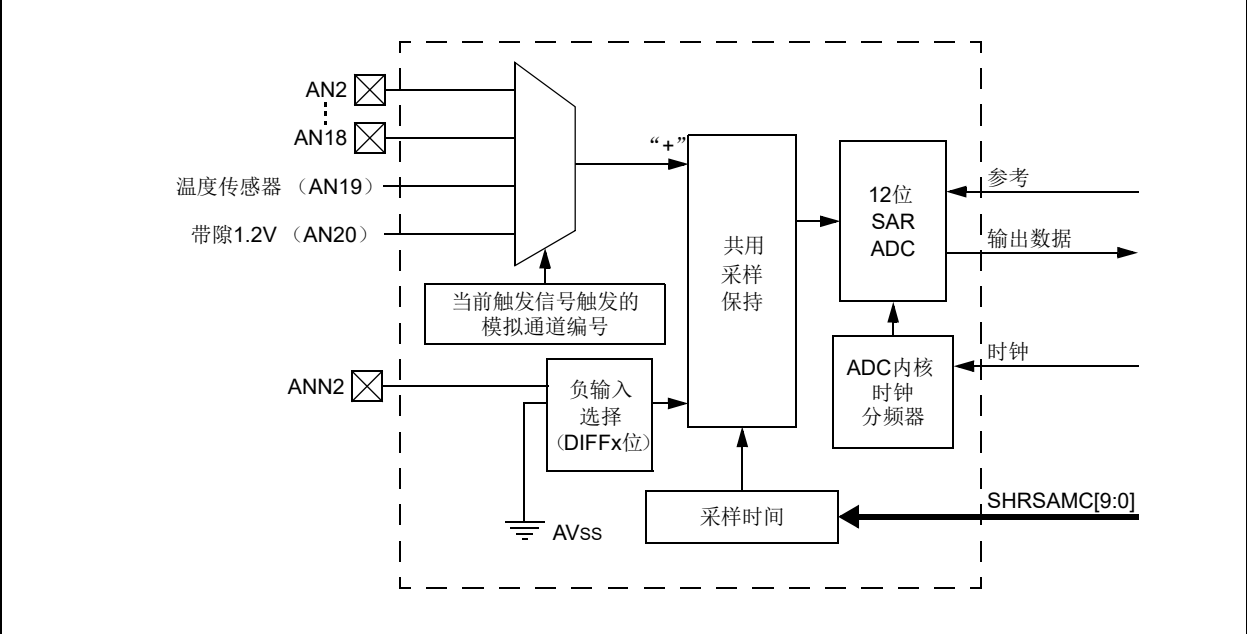
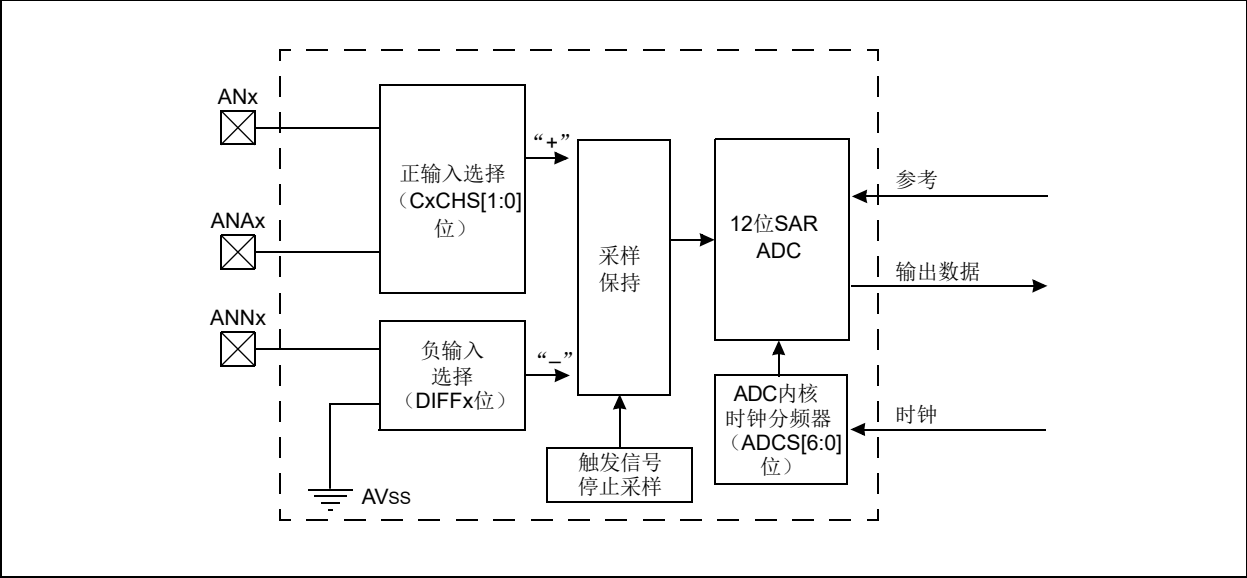


图12-3: 专用ADC内核



## 12.2 温度传感器

ADC通道AN19连接到正向偏置二极管。它可用于测量芯片温度。该二极管提供温度系数约为-1.5 mV/°C的输出，输出可由ADC监视。要获得精确的增益和偏移值，建议使用两个温度点校准。

## 12.3 模数转换器资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 12.3.1 主要资源

- 《dsPIC33/PIC24 系列参考手册》中的“**12位高速多SAR A/D转换器 (ADC)**” ([www.microchip.com/DS70005213](http://www.microchip.com/DS70005213))
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24 系列参考手册》章节
- 开发工具

## 12.4 ADC控制/状态寄存器

寄存器12-1: **ADCON1L: ADC控制低位字寄存器1**

R/W-0	U-0	R/W-0	U-0	r-0	U-0	U-0	U-0
ADON <sup>(1)</sup>	—	ADSIDL	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 15	<b>ADON:</b> ADC使能位 <sup>(1)</sup> 1 = 使能ADC模块 0 = 禁止ADC模块
bit 14	<b>未实现:</b> 读为0
bit 13	<b>ADSIDL:</b> ADC空闲模式停止位 1 = 当器件进入空闲模式时, 模块停止工作 0 = 在空闲模式下模块继续工作
bit 12	<b>未实现:</b> 读为0
bit 11	<b>保留:</b> 保持为0
bit 10-0	<b>未实现:</b> 读为0

**注 1:** 仅应在配置完ADC模块后将ADON位置1。在ADON = 1时更改ADC配置位将导致不可预测的行为。

# dsPIC33CK64MP105系列

寄存器12-2:      **ADCON1H: ADC控制高位字寄存器1**

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

R/W-0	R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0
FORM	SHRRES1	SHRRES0	—	—	—	—	—
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 15-8      **未实现:** 读为0

bit 7      **FORM:** 小数数据输出格式位

1 = 小数

0 = 整数

bit 6-5      **SHRRES[1:0]:** 共用ADC内核分辨率选择位

11 = 12位分辨率

10 = 10位分辨率

01 = 8位分辨率

00 = 6位分辨率

bit 4-0      **未实现:** 读为0

寄存器12-3: ADON2L: ADC控制低位字寄存器2

R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
REFCIE	REFERCIE	—	EIEN	PTGEN <sup>(3)</sup>	SHREISEL2 <sup>(1)</sup>	SHREISEL1 <sup>(1)</sup>	SHREISEL0 <sup>(1)</sup>
bit 15				bit 8			
U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SHRADCS[6:0] <sup>(2)</sup>						
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 15 **REFCIE:** 带隙和参考电压就绪通用中断允许位

1 = 带隙就绪时产生通用中断

0 = 禁止在发生带隙就绪事件时产生通用中断

bit 14 **REFERCIE:** 带隙或参考电压错误通用中断允许位

1 = 检测到带隙或参考电压错误时产生通用中断

0 = 禁止在发生带隙和参考电压错误事件时产生通用中断

bit 13 **未实现:** 读为0bit 12 **EIEN:** 提前中断允许位

1 = 为输入通道中断允许提前中断功能 (EISTATx标志置1时)

0 = 转换完成时 (ANxRDY标志置1时) 产生各个中断

bit 11 **PTGEN:** PTG转换请求接口位<sup>(3)</sup>

1 = 使能PTG触发信号

0 = 禁止PTG触发信号

bit 10-8 **SHREISEL[2:0]:** 共用内核提前中断时间选择位<sup>(1)</sup>

111 = 设置提前中断, 产生中断的时间比数据就绪的时间提前8个TADCORE时钟

110 = 设置提前中断, 产生中断的时间比数据就绪的时间提前7个TADCORE时钟

101 = 设置提前中断, 产生中断的时间比数据就绪的时间提前6个TADCORE时钟

100 = 设置提前中断, 产生中断的时间比数据就绪的时间提前5个TADCORE时钟

011 = 设置提前中断, 产生中断的时间比数据就绪的时间提前4个TADCORE时钟

010 = 设置提前中断, 产生中断的时间比数据就绪的时间提前3个TADCORE时钟

001 = 设置提前中断, 产生中断的时间比数据就绪的时间提前2个TADCORE时钟

000 = 设置提前中断, 产生中断的时间比数据就绪的时间提前1个TADCORE时钟

bit 7 **未实现:** 读为0bit 6-0 **SHRADCS[6:0]:** 共用ADC内核输入时钟分频位<sup>(2)</sup>

这些位用于确定一个共用TADCORE (内核时钟周期) 中的TCORESRC (源时钟周期) 数。

1111111 = 254个源时钟周期

...

0000011 = 6个源时钟周期

0000010 = 4个源时钟周期

0000001 = 2个源时钟周期

0000000 = 2个源时钟周期

**注 1:** 对于6位共用ADC内核分辨率 (SHRRES[1:0] = 00), SHREISEL[2:0]设置100至111无效, 请勿使用。对于8位共用ADC内核分辨率 (SHRRES[1:0] = 01), SHREISEL[2:0]设置110和111无效, 请勿使用。

**2:** 由SHRADCS[6:0]位选择的ADC时钟频率不得超过70 MHz。

**3:** 如果使能了PTG触发信号, 则无法使用其他ADC触发源。

# dsPIC33CK64MP105系列

**寄存器12-4: ADON2H: ADC控制高位字寄存器2**

HSC/R-0	HSC/R-0	U-0	r-0	r-0	r-0	R/W-0	R/W-0
REFRDY	REFERR	—	—	—	—	SHRSAMC9	SHRSAMC8
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
SHRSAMC7	SHRSAMC6	SHRSAMC5	SHRSAMC4	SHRSAMC3	SHRSAMC2	SHRSAMC1	SHRSAMC0
bit 7						bit 0	

<b>图注:</b>	r = 保留位	U = 未实现位, 读为0
R = 可读位	W = 可写位	HSC = 硬件置1/清零位
-n = POR时的值	1 = 置1	0 = 清零
		x = 未知

- bit 15      **REFRDY:** 带隙和参考电压就绪标志位  
1 = 带隙就绪  
0 = 带隙未就绪
- bit 14      **REFERR:** 带隙或参考电压错误标志位  
1 = ADC模块使能 (ADON = 1) 后带隙被移除  
0 = 未检测到带隙错误
- bit 13      **未实现:** 读为0
- bit 12-10   **保留:** 保持为0
- bit 9-0      **SHRSAMC[9:0]:** 共用ADC内核采样时间选择位  
这些位用于指定共用ADC内核采样时间内的共用ADC内核时钟周期 (TADCORE) 数。  
1111111111 = 1025个TADCORE  
...  
0000000001 = 3个TADCORE  
0000000000 = 2个TADCORE

寄存器12-5: **ADCON3L: ADC控制低位字寄存器3**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	HSC/R-0	R/W-0	HSC/R-0
REFSEL2	REFSEL1	REFSEL0	SUSPEND	SUSPCIE	SUSPRDY	SHRSAMP	CNVRTCH
bit 15							bit 8

R/W-0	HSC/R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SWLCTRG	SWCTRG	CNVCHSEL5	CNVCHSEL4	CNVCHSEL3	CNVCHSEL2	CNVCHSEL1	CNVCHSEL0
bit 7							bit 0

图注:	U = 未实现位, 读为0
R = 可读位	W = 可写位
-n = POR时的值	1 = 置1
	HSC = 硬件置1/清零位
	0 = 清零
	x = 未知

bit 15-13 **REFSEL[2:0]: ADC参考电压选择位**

值	VREFH	VREFL
000	AVDD	AVSS

001-111 = **未实现**: 不要使用bit 12 **SUSPEND**: 所有ADC内核触发禁止位

1 = 禁止所有ADC内核的所有新触发事件

0 = 可以触发所有ADC内核

bit 11 **SUSPCIE**: 所有ADC内核暂停通用中断允许位1 = ADC内核触发暂停 (**SUSPEND**位 = 1) 且之前的所有转换完成 (**SUSPRDY**位置1) 时产生通用中断

0 = 发生ADC内核暂停事件时不产生通用中断

bit 10 **SUSPRDY**: 所有ADC内核暂停标志位1 = 所有ADC内核暂停 (**SUSPEND**位 = 1) 且未在进行任何转换

0 = ADC内核正在进行之前的转换

bit 9 **SHRSAMP**: 共用ADC内核采样直接控制位该位应与 **CNVRTCH** 位控制的单独通道转换触发信号配合使用。它将 **CNVCHSEL[5:0]** 位指定的模拟输入与共用ADC内核相连, 并允许延长采样时间。该位不由硬件控制, 必须在转换开始前清零 (将 **CNVRTCH** 置1)。1 = 共用ADC内核采样 **CNVCHSEL[5:0]** 位指定的模拟输入

0 = 采样由共用ADC内核硬件控制

bit 8 **CNVRTCH**: 软件单独通道转换触发位1 = 为 **CNVCHSEL[5:0]** 位指定的模拟输入产生单个触发信号; 该位置1后, 将在下一个指令周期由硬件自动清零

0 = 可产生下一个单独通道转换触发信号

bit 7 **SWLCTRG**: 软件电平敏感通用触发位1 = 通过软件连续为所有通道产生触发信号, 此电平敏感通用触发信号选作 **ADTRIGNL** 和 **ADTRIGNH** 寄存器中的触发源

0 = 不产生软件电平敏感通用触发信号

bit 6 **SWCTRG**: 软件通用触发位1 = 通过软件为所有通道产生单个触发信号, 此通用触发信号选作 **ADTRIGNL** 和 **ADTRIGNH** 寄存器中的触发源; 该位置1后, 将在下一个指令周期由硬件自动清零

0 = 准备好产生下一个软件通用触发信号

bit 5-0 **CNVCHSEL [5:0]: 软件单独通道转换触发的通道编号选择位**这些位用于定义 **CNVRTCH** 位置1时要转换的通道。

# dsPIC33CK64MP105系列

寄存器12-6: **ADCON3H: ADC控制高位字寄存器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
CLKSEL1 <sup>(1)</sup>	CLKSEL0 <sup>(1)</sup>	CLKDIV5 <sup>(2)</sup>	CLKDIV4 <sup>(2)</sup>	CLKDIV3 <sup>(2)</sup>	CLKDIV2 <sup>(2)</sup>	CLKDIV1 <sup>(2)</sup>	CLKDIV0 <sup>(2)</sup>
bit 15							bit 8

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
SHREN	—	—	—	—	—	C1EN	C0EN
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 15-14 **CLKSEL[1:0]: ADC模块时钟源选择位<sup>(1)</sup>**

11 = Fvco/4

10 = AFVcodiv

01 = Fosc

00 = Fp (外设时钟)

bit 13-8 **CLKDIV[5:0]: ADC模块时钟源分频比位<sup>(2)</sup>**

该分频比根据CLKSEL[1:0]位选择的Tsrc ADC模块时钟源形成所有(共用和专用)ADC内核使用的Tcoresrc时钟。随后, 每个ADC内核通过ADCORExH寄存器中的ADCS[6:0]位或ADCON2L寄存器中的SHRADCS[6:0]位单独对Tcoresrc时钟进行分频, 以获得内核特定的Tadc core时钟。

111111 = 64个源时钟周期

...

000011 = 4个源时钟周期

000010 = 3个源时钟周期

000001 = 2个源时钟周期

000000 = 1个源时钟周期

bit 7 **SHREN: 共用ADC内核使能位**

1 = 使能共用ADC内核

0 = 禁止共用ADC内核

bit 6-2 **未实现: 读为0**

bit 1 **C1EN: 专用ADC内核1使能位**

1 = 使能专用ADC内核1

0 = 禁止专用ADC内核1

bit 0 **C0EN: 专用ADC内核0使能位**

1 = 使能专用ADC内核0

0 = 禁止专用ADC内核0

**注 1:** 由CLKSEL[1:0]位选择的ADC输入时钟频率不得超过560 MHz。

**2:** 经CLKDIV[5:0]位选择的第一个分频比分频后的ADC时钟频率不得超过280 MHz。



**寄存器12-7: ADCON4L: ADC控制低位字寄存器4**

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

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SAMC1EN	SAMC0EN
bit 7						bit 0	

<b>图注:</b>	r = 保留位		
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 15-10      **未实现:** 读为0

bit 9-8        **保留:** 必须写为0

bit 7-2        **未实现:** 读为0

bit 1          **SAMC1EN:** 专用ADC内核1转换延时使能位

1 = 触发后, 转换将延时, ADC内核将在ADCCORE1L寄存器中的SAMC[9:0]位指定的时间内继续采样  
0 = 触发后, 采样立即停止, 转换将在下一个内核时钟周期开始

bit 0          **SAMC0EN:** 专用ADC内核0转换延时使能位

1 = 触发后, 转换将延时, ADC内核将在ADCCORE0L寄存器中的SAMC[9:0]位指定的时间内继续采样  
0 = 触发后, 采样立即停止, 转换将在下一个内核时钟周期开始

# dsPIC33CK64MP105系列

寄存器12-8:      **ADCON4H: ADC控制高位字寄存器4**

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

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	C1CHS1	C1CHS0	C0CHS1	C0CHS0
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 15-4      **未实现:** 读为0

bit 3-2      **C1CHS[1:0]:** 专用ADC内核1输入通道选择位

11 = 保留

10 = 保留

01 = ANA1

00 = AN1

bit 1-0      **C0CHS[1:0]:** 专用ADC内核0输入通道选择位

11 = 保留

10 = 保留

01 = ANA0

00 = AN0

## 寄存器12-9: ADCON5L: ADC控制低位字寄存器5

HSC/R-0	U-0	U-0	U-0	U-0	U-0	HSC/R-0	HSC/R-0
SHRRDY	—	—	—	—	—	C1RDY	C0RDY
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
SHRPWR	—	—	—	—	—	C1PWR	C0PWR
bit 7						bit 0	

<b>图注:</b>	U = 未实现位, 读为0		
R = 可读位	W = 可写位	HSC = 硬件置1/清零位	
-n = POR时的值	1 = 置1	0 = 清零	x = 未知

- bit 15      **SHRRDY:** 共用ADC内核就绪标志位  
1 = 共用ADC内核上电并准备好进行工作  
0 = 共用ADC内核未准备好进行工作
- bit 14-10      **未实现:** 读为0
- bit 9      **C1RDY:** 专用ADC内核1就绪标志位  
1 = ADC内核1上电并准备好进行工作  
0 = ADC内核1未准备好进行工作
- bit 8      **C0RDY:** 专用ADC内核0就绪标志位  
1 = ADC内核0上电并准备好进行工作  
0 = ADC内核0未准备好进行工作
- bit 7      **SHRPWR:** 共用ADC内核电源使能位  
1 = ADC内核上电  
0 = ADC内核关闭
- bit 6-2      **未实现:** 读为0
- bit 1      **C1PWR:** 专用ADC内核1电源使能位  
1 = ADC内核1上电  
0 = ADC内核1掉电
- bit 0      **C0PWR:** 专用ADC内核0电源使能位  
1 = ADC内核0上电  
0 = ADC内核0关闭

# dsPIC33CK64MP105系列

寄存器12-10: **ADCON5H: ADC控制高位字寄存器5**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WARMTIME3	WARMTIME2	WARMTIME1	WARMTIME0
bit 15				bit 8			

R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
SHRCIE	—	—	—	—	—	C1CIE	C0CIE
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 15-12 **未实现:** 读为0

bit 11-8 **WARMTIME[3:0]:** ADC专用内核x上电延时位

这些位用于确定所有ADC内核的上电延时(用内核源时钟周期(TCRESRC)数表示)。

1111 = 32768个源时钟周期

1110 = 16384个源时钟周期

1101 = 8192个源时钟周期

1100 = 4096个源时钟周期

1011 = 2048个源时钟周期

1010 = 1024个源时钟周期

1001 = 512个源时钟周期

1000 = 256个源时钟周期

0111 = 128个源时钟周期

0110 = 64个源时钟周期

0101 = 32个源时钟周期

0100 = 16个源时钟周期

00xx = 16个源时钟周期

bit 7 **SHRCIE:** 共用ADC内核就绪通用中断允许位

1 = 当共用ADC内核上电且准备好进行工作时产生通用中断

0 = 禁止在发生共用ADC内核就绪事件时产生通用中断

bit 6-2 **未实现:** 读为0

bit 1 **C1CIE:** 专用ADC内核1就绪通用中断允许位

1 = 当ADC内核1上电且准备好进行工作时产生通用中断

0 = 禁止在发生ADC内核1就绪事件时产生通用中断

bit 0 **C0CIE:** 专用ADC内核0就绪通用中断允许位

1 = 当ADC内核0上电且准备好进行工作时产生通用中断

0 = 禁止在发生ADC内核0就绪事件时产生通用中断

**寄存器12-11:     ADCORExL: 专用ADC内核x控制低位字寄存器 (x = 0和1)**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SAMC[9:8]	
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
SAMC[7:0]							
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 15-10     **未实现:** 读为0

bit 9-0     **SAMC[9:0]:** 专用ADC内核x转换延时选择位

这些位用于确定从触发事件到转换开始的时间（用内核时钟周期（TADCORE）数表示）。在这段时间内，ADC内核x将继续采样。该功能通过ADCON4L寄存器中的SAMCxEN位使能。

1111111111 = 1025个TADCORE

...

0000000001 = 3个TADCORE

0000000000 = 2个TADCORE

# dsPIC33CK64MP105系列

寄存器12-12: **ADCORExH**: 专用ADC内核x控制高位字寄存器 (x = 0和1)

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EISEL2	EISEL1	EISEL0	RES1	RES2
bit 15						bit 8	

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	ADCS6 <sup>(2)</sup>	ADCS5 <sup>(2)</sup>	ADCS4 <sup>(2)</sup>	ADCS3 <sup>(2)</sup>	ADCS2 <sup>(2)</sup>	ADCS1 <sup>(2)</sup>	ADCS0 <sup>(2)</sup>
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 15-13 **未实现**: 读为0

bit 12-10 **EISEL[2:0]**: ADC 内核x提前中断时间选择位

111 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 8 个 TADCORE 时钟  
 110 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 7 个 TADCORE 时钟  
 101 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 6 个 TADCORE 时钟  
 100 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 5 个 TADCORE 时钟  
 011 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 4 个 TADCORE 时钟  
 010 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 3 个 TADCORE 时钟  
 001 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 2 个 TADCORE 时钟  
 000 = 设置提前中断, 产生中断的时间比数据就绪的时间提前 1 个 TADCORE 时钟

bit 9-8 **RES[1:0]**: ADC 内核x分辨率选择位

11 = 12 位分辨率  
 10 = 10 位分辨率  
 01 = 8 位分辨率<sup>(1)</sup>  
 00 = 6 位分辨率<sup>(1)</sup>

bit 7 **未实现**: 读为0

bit 6-0 **ADCS[6:0]**: ADC 内核x输入时钟分频比位<sup>(2)</sup>

这些位用于确定一个内核时钟周期 (TADCORE) 中的源时钟周期 (TCORESRC) 数。  
 1111111 = 254 个源时钟周期  
 ...  
 0000011 = 6 个源时钟周期  
 0000010 = 4 个源时钟周期  
 0000001 = 2 个源时钟周期  
 0000000 = 2 个源时钟周期

**注 1:** 对于6位ADC内核分辨率 (RES[1:0] = 00), EISEL[2:0]位设置100至111无效, 请勿使用。对于8位ADC内核分辨率 (RES[1:0] = 01), EISEL[2:0]位设置110和111无效, 请勿使用。

**2:** 由ADCS[6:0]位选择的ADC时钟频率不得超过 70 MHz。

## 寄存器12-13: ADLVLTRGL: ADC电平敏感触发控制低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LVLEN[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
LVLEN[7:0]							
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 15-0 **LVLEN[15:0]:** 相应模拟输入的电平触发使能位

1 = 输入触发为电平敏感

0 = 输入触发为边沿敏感

## 寄存器12-14: ADLVLTRGH: ADC电平敏感触发控制高位字寄存器

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	LVLEN[20:16]				
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 15-5 **未实现:** 读为0

bit 4-0 **LVLEN[20:16]:** 相应模拟输入的电平触发使能位

1 = 输入触发为电平敏感

0 = 输入触发为边沿敏感

# dsPIC33CK64MP105系列

寄存器12-15:     **ADEIEL: ADC提前中断允许低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EIEN[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
EIEN[7:0]							
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 15-0     **EIEN[15:0]:** 相应模拟输入的提前中断允许位  
1 = 允许相应通道的提前中断  
0 = 禁止相应通道的提前中断

寄存器12-16:     **ADEIEH: ADC提前中断允许高位字寄存器**

U-0		U-0		U-0		U-0		U-0		U-0		U-0	
—		—		—		—		—		—		—	
bit 15												bit 8	
U-0		U-0		U-0		R/W-0		R/W-0		R/W-0		R/W-0	
—		—		—		EIEN[20:16]							
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 15-5     **未实现:** 读为0  
bit 4-0     **EIEN[20:16]:** 相应模拟输入的提前中断允许位  
1 = 允许相应通道的提前中断  
0 = 禁止相应通道的提前中断



## 寄存器12-17: ADEISTATL: ADC提前中断状态低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EISTAT[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
EISTAT[7:0]							
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 15-0 **EISTAT[15:0]:** 相应模拟输入的提前中断状态位

1 = 产生了提前中断

0 = 自上一次读取ADCBUFx后未产生提前中断

## 寄存器12-18: ADEISTATH: ADC提前中断状态高位字寄存器

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	EISTAT[20:16]				
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 15-5 **未实现:** 读为0

bit 4-0 **EISTAT[20:16]:** 相应模拟输入的提前中断状态位

1 = 产生了提前中断

0 = 自上一次读取ADCBUFx后未产生提前中断

# dsPIC33CK64MP105系列

## 寄存器12-19: ADMOD0L: ADC输入模式控制低位字寄存器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
DIFF7	SIGN7	DIFF6	SIGN6	DIFF5	SIGN5	DIFF4	SIGN4
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
DIFF3	SIGN3	DIFF2	SIGN2	DIFF1	SIGN1	DIFF0	SIGN0
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 15至bit 1 **DIFF[7:0]:** 相应模拟输入的差分模式位

(奇数)

1 = 通道为差分模式

0 = 通道为单端模式

bit 14至bit 0 **SIGN[7:0]:** 相应模拟输入的输出数据符号位

(偶数)

1 = 通道输出数据有符号

0 = 通道输出数据无符号

## 寄存器12-20: ADMOD0H: ADC输入模式控制高位字寄存器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
DIFF15	SIGN15	DIFF14	SIGN14	DIFF13	SIGN13	DIFF12	SIGN12
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
DIFF11	SIGN11	DIFF10	SIGN10	DIFF9	SIGN9	DIFF8	SIGN8
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 15至bit 1 **DIFF[15:8]:** 相应模拟输入的差分模式位

(奇数)

1 = 通道为差分模式

0 = 通道为单端模式

bit 14至bit 0 **SIGN[15:8]:** 相应模拟输入的输出数据符号位

(偶数)

1 = 通道输出数据有符号

0 = 通道输出数据无符号

## 寄存器12-21: ADMOD1L: ADC输入模式控制低位字寄存器1

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	DIFF20	SIGN20
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
DIFF19	SIGN19	DIFF18	SIGN18	DIFF17	SIGN17	DIFF16	SIGN16
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 15至bit 1 **DIFF[20:16]:** 相应模拟输入的差分模式位

(奇数)

1 = 通道为差分模式

0 = 通道为单端模式

bit 14至bit 0 **SIGN[20:16]:** 相应模拟输入的输出数据符号位

(偶数)

1 = 通道输出数据有符号

0 = 通道输出数据无符号

# dsPIC33CK64MP105系列

**寄存器12-22: ADIEL: ADC中断允许低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IE[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
IE[7:0]							
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 15-0

**IE[15:0]:** 通用中断允许位

1 = 允许相应通道的通用中断和单独中断

0 = 禁止相应通道的通用中断和单独中断

**寄存器12-23: ADIEH: ADC中断允许高位字寄存器**

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	IE[20:16]				
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 15-5

**未实现:** 读为0

bit 4-0

**IE[20:16]:** 通用中断允许位

1 = 允许相应通道的通用中断和单独中断

0 = 禁止相应通道的通用中断和单独中断

## 寄存器12-24: ADSTATL: ADC数据就绪状态低位字寄存器

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN[15:8]RDY							
bit 15							
bit 8							

HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
AN[7:0]RDY							
bit 7							
bit 0							

图注: U = 未实现位, 读为0  
R = 可读位 W = 可写位 HSC = 硬件置1/清零位  
-n = POR时的值 1 = 置1 0 = 清零 x = 未知

bit 15-0 **AN[15:0]RDY:** 相应模拟输入的通用中断允许位  
1 = 相应ADCBUFx寄存器中的通道转换结果已就绪  
0 = 通道转换结果未就绪

## 寄存器12-25: ADSTATH: ADC数据就绪状态高位字寄存器

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

U-0	U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	—	AN[20:16]RDY				
bit 7			bit 0				

图注: U = 未实现位, 读为0  
R = 可读位 W = 可写位 HSC = 硬件置1/清零位  
-n = POR时的值 1 = 置1 0 = 清零 x = 未知

bit 15-5 **未实现:** 读为0  
bit 4-0 **AN[20:16]RDY:** 相应模拟输入的通用中断允许位  
1 = 相应ADCBUFx寄存器中的通道转换结果已就绪  
0 = 通道转换结果未就绪

# dsPIC33CK64MP105系列

**寄存器 12-26: ADTRIGnL/ADTRIGnH: ADC通道触发n(x)选择低位字和高位字寄存器 (x = 0至20; n = 0至6)**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TRGSRC(x+1)4	TRGSRC(x+1)3	TRGSRC(x+1)2	TRGSRC(x+1)1	TRGSRC(x+1)0
bit 15						bit 8	

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	TRGSRCx4	TRGSRCx3	TRGSRCx2	TRGSRCx1	TRGSRCx0
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 15-13 **未实现:** 读为0

bit 12-8 **TRGSRC(x+1)[4:0]:** 相应模拟输入的触发源选择位 (TRGSRC1至TRGSRC19——奇编号)

11111 = ADTRG31 (PPS输入)

11110 = PTG12

11101 = CLC2

11100 = CLC1

11011 = 保留

11010 = 保留

11001 = 保留

11000 = MCCP5 CCP中断

10111 = SCCP4 CCP中断

10110 = SCCP3 CCP中断

10101 = SCCP2 CCP中断

10100 = SCCP1 CCP中断

10011 = 保留

10010 = CLC4输出

10001 = CLC3输出

10000 = MCCP5触发信号

01111 = SCCP4触发信号

01110 = SCCP3触发信号

01101 = SCCP2触发信号

01100 = SCCP1触发信号

01011 = PWM4触发信号2

01010 = PWM4触发信号1

01001 = PWM3触发信号2

01000 = PWM3触发信号1

00111 = PWM2触发信号2

00110 = PWM2触发信号1

00101 = PWM1触发信号2

00100 = PWM1触发信号1

00011 = 保留

00010 = 电平软件触发

00001 = 通用软件触发

00000 = 不使能触发

bit 7-5 **未实现:** 读为0

**寄存器 12-26: ADTRIGnL/ADTRIGnH: ADC通道触发n(x)选择低位字和高位字寄存器 (x = 0至20; n = 0至6) (续)**

bit 4-0 **TRGSRCx[4:0]:** 相应模拟输入的通用中断允许位 (TRGSRC0至TRGSRC20——偶编号)

11111 = ADTRG31 (PPS输入)

11110 = PTG12

11101 = CLC2

11100 = CLC1

11011 = 保留

11010 = 保留

11001 = 保留

11000 = MCCP5 CCP中断

10111 = SCCP4 CCP中断

10110 = SCCP3 CCP中断

10101 = SCCP2 CCP中断

10100 = SCCP1 CCP中断

10011 = 保留

10010 = CLC4输出

10001 = CLC3输出

10000 = MCCP5触发信号

01111 = SCCP4触发信号

01110 = SCCP3触发信号

01101 = SCCP2触发信号

01100 = SCCP1触发信号

01011 = PWM4触发信号2

01010 = PWM4触发信号1

01001 = PWM3触发信号2

01000 = PWM3触发信号1

00111 = PWM2触发信号2

00110 = PWM2触发信号1

00101 = PWM1触发信号2

00100 = PWM1触发信号1

00011 = 保留

00010 = 电平软件触发

00001 = 通用软件触发

00000 = 不使能触发

# dsPIC33CK64MP105系列

**寄存器12-27: ADCMPxCON: ADC数字比较器x控制寄存器 (x = 0、1、2和3)**

U-0	U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	—	CHNL4	CHNL3	CHNL2	CHNL1	CHNL0
bit 15			bit 8				

R/W-0	R/W-0	HC/HS/R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CMPEN	IE	STAT	BTWN	HIHI	HILO	LOHI	LOLO
bit 7			bit 0				

<b>图注:</b>	HC = 硬件清零位	U = 未实现位, 读为0
R = 可读位	W = 可写位	HSC = 硬件置1/清零位
-n = POR时的值	1 = 置1	0 = 清零
		HS = 硬件置1位

- bit 15-13     **未实现:** 读为0
- bit 12-8     **CHNL[4:0]:** 输入通道编号位  
如果比较器检测到某个通道发生事件, 则该通道编号将写入这些位。  
11111 = 保留  
...  
10101 = 保留  
10100 = 带隙, 1.2V (AN20)  
10011 = 温度传感器 (AN19)  
10010 = AN18  
...  
00011 = AN3  
00010 = AN2  
00001 = AN1  
00000 = AN0
- bit 7     **CMPEN:** 比较器使能位  
1 = 使能比较器  
0 = 禁止比较器并清零STAT状态位
- bit 6     **IE:** 比较器通用ADC中断允许位  
1 = 如果比较器检测到比较事件, 将产生通用ADC中断  
0 = 比较器将不产生通用ADC中断
- bit 5     **STAT:** 比较器事件状态位  
当从CHNL[4:0]位读取通道编号时, 该位由硬件清零  
1 = 上一次读取CHNL[4:0]位后已检测到比较事件  
0 = 上一次读取CHNL[4:0]位后尚未检测到比较事件
- bit 4     **BTWN:** 介于下限/上限间的比较器事件位  
1 = 当 $ADCMPxLO \leq ADCBUFx < ADCMPxHI$ 时, 产生数字比较器事件  
0 = 当 $ADCMPxLO \leq ADCBUFx < ADCMPxHI$ 时, 不产生数字比较器事件
- bit 3     **HIHI:** 高于上限的比较器事件位  
1 = 当 $ADCBUFx \geq ADCMPxHI$ 时, 产生数字比较器事件  
0 = 当 $ADCBUFx \geq ADCMPxHI$ 时, 不产生数字比较器事件
- bit 2     **HILO:** 低于上限的比较器事件位  
1 = 当 $ADCBUFx < ADCMPxHI$ 时, 产生数字比较器事件  
0 = 当 $ADCBUFx < ADCMPxHI$ 时, 不产生数字比较器事件
- bit 1     **LOHI:** 高于下限的比较器事件位  
1 = 当 $ADCBUFx \geq ADCMPxLO$ 时, 产生数字比较器事件  
0 = 当 $ADCBUFx \geq ADCMPxLO$ 时, 不产生数字比较器事件
- bit 0     **LOLO:** 低于下限的比较器事件位  
1 = 当 $ADCBUFx < ADCMPxLO$ 时, 产生数字比较器事件  
0 = 当 $ADCBUFx < ADCMPxLO$ 时, 不产生数字比较器事件



**寄存器12-28: ADCMPxENL: ADC数字比较器x通道使能低位字寄存器 (x = 0或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
CMPEN[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
CMPEN[7:0]							
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 15-0 **CMPEN[15:0]:** 相应输入通道的比较器使能位

1 = 比较器使用相应通道的转换结果

0 = 比较器不使用相应通道的转换结果

**寄存器12-29: ADCMPxENH: ADC数字比较器x通道使能高位字寄存器 (x = 0或3)**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	—
bit 15				bit 8			

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CMPEN[20:16]				
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 15-5 **未实现:** 读为0

bit 4-0 **CMPEN[20:16]:** 相应输入通道的比较器使能位

1 = 比较器使用相应通道的转换结果

0 = 比较器不使用相应通道的转换结果

# dsPIC33CK64MP105系列

**寄存器12-30: ADFLxCON: ADC数字滤波器x控制寄存器 (x = 0或3)**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	HSC/R-0
FLEN	MODE1	MODE0	OVRSAM2	OVRSAM1	OVRSAM0	IE	RDY
bit 15							bit 8

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	FLCHSEL4	FLCHSEL3	FLCHSEL2	FLCHSEL1	FLCHSEL0
bit 7							bit 0

**图注:** U = 未实现位, 读为0  
R = 可读位 W = 可写位 HSC = 硬件置1/清零位  
-n = POR时的值 1 = 置1 0 = 清零 x = 未知

- bit 15 **FLEN:** 滤波器使能位  
1 = 使能滤波器  
0 = 禁止滤波器并清零RDY位
- bit 14-13 **MODE[1:0]:** 滤波器模式位  
11 = 平均模式  
10 = 保留  
01 = 保留  
00 = 过采样模式
- bit 12-10 **OVRSAM[2:0]:** 滤波器平均/过采样率位  
如果MODE[1:0] = 00:  
111 = 128x (ADFLxDAT 寄存器中的 16 位结果采用 12.4 格式)  
110 = 32x (ADFLxDAT 寄存器中的 15 位结果采用 12.3 格式)  
101 = 8x (ADFLxDAT 寄存器中的 14 位结果采用 12.2 格式)  
100 = 2x (ADFLxDAT 寄存器中的 13 位结果采用 12.1 格式)  
011 = 256x (ADFLxDAT 寄存器中的 16 位结果采用 12.4 格式)  
010 = 64x (ADFLxDAT 寄存器中的 15 位结果采用 12.3 格式)  
001 = 16x (ADFLxDAT 寄存器中的 14 位结果采用 12.2 格式)  
000 = 4x (ADFLxDAT 寄存器中的 13 位结果采用 12.1 格式)  
如果MODE[1:0] = 11 (所有情况下 ADFLxDAT 寄存器中的 12 位结果):  
111 = 256x  
110 = 128x  
101 = 64x  
100 = 32x  
011 = 16x  
110 = 8x  
001 = 4x  
000 = 2x
- bit 9 **IE:** 滤波器通用ADC中断允许位  
1 = 当滤波器结果就绪时, 产生通用ADC中断  
0 = 滤波器将不产生通用ADC中断
- bit 8 **RDY:** 过采样滤波器数据就绪标志位  
当从ADFLxDAT 寄存器读取结果时, 该位由硬件清零。  
1 = ADFLxDAT 寄存器中的数据已就绪  
0 = ADFLxDAT 寄存器已被读取, ADFLxDAT 寄存器中的新数据未就绪
- bit 7-5 **未实现:** 读为0

寄存器12-30:     **ADFLxCON: ADC数字滤波器x控制寄存器 (x = 0或3) (续)**

bit 4-0     **FLCHSEL[4:0]:** 过采样滤波器输入通道选择位

11111 = 保留

...

10101 = 保留

10100 = 带隙, 1.2V (AN20)

10011 = 温度传感器 (AN19)

10010 = AN18

...

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

注:

## 13.0 带斜率补偿DAC的高速模拟比较器

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“高速模拟比较器模块”(www.microchip.com/DS70005280)。

高速模拟比较器模块提供了一种方法来监视电源转换应用中的电压、电流和其他关键信号,这些信号可能因为速度太快而无法被CPU和ADC捕捉到。总共有3个比较器模块。模拟比较器模块可用于实现峰值电流模式控制、临界导通模式(变频)和滞后控制模式。

### 13.1 概述

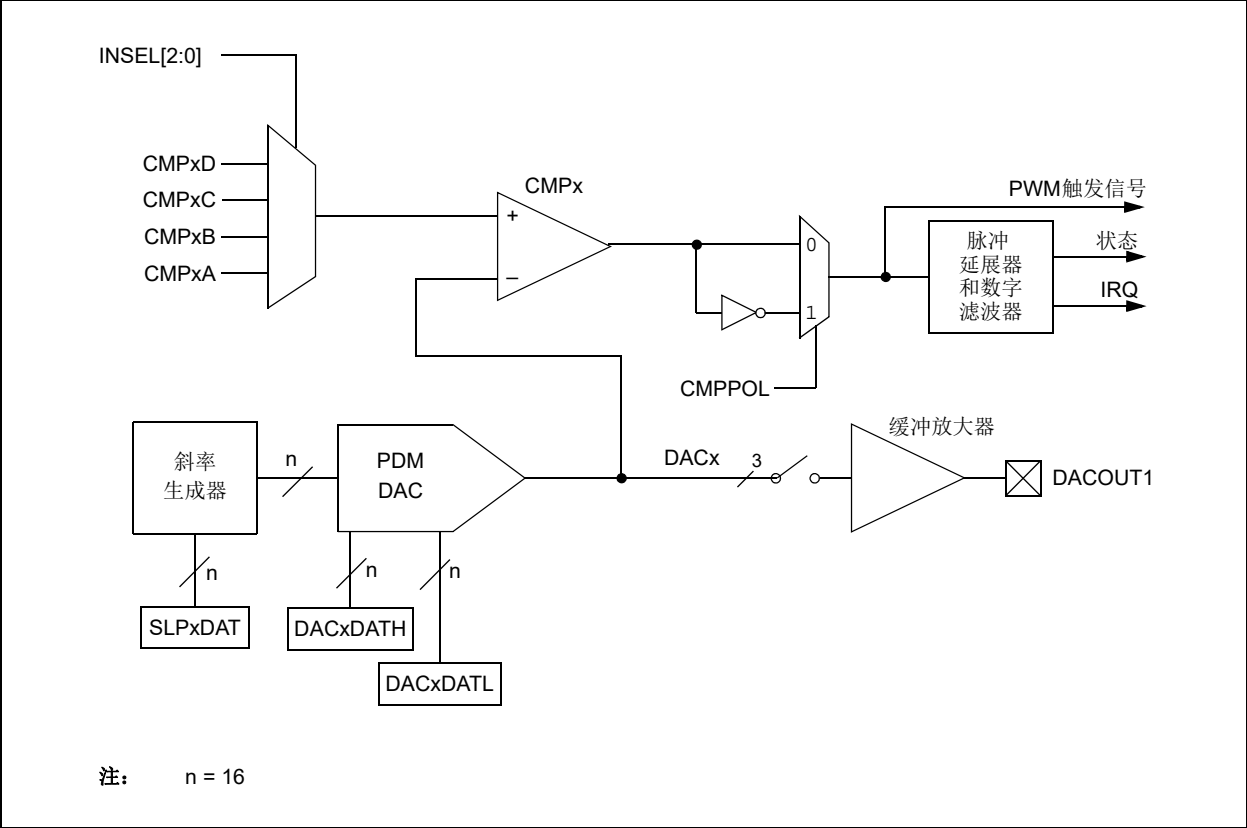
高速模拟比较器模块包括高速比较器、脉冲密度调制(Pulse Density Modulation, PDM) DAC和斜率补偿单元。斜率补偿单元提供了可用于改变DAC输出的用户自定义斜率。在峰值电流模式控制等需要通过斜率补偿维持电源稳定性的应用中,此功能非常有用。用户仅需指定斜率补偿的方向和变化率,并相应地修改DAC的输出。

DAC包含一个PDM单元,PDM单元后跟一个数字控制的多相RC滤波器。PDM单元使用相位累加器电路来生成输出脉冲流。相对于累加器位宽所支持的最大值,脉冲流的密度与输入数据值成比例。输出脉冲密度代表所需的输出电压。脉冲流经RC滤波器滤波后产生模拟电压。DAC的输出与比较器的反相输入连接。比较器的同相输入可以使用多路开关(MUX)从输入引脚中选择。比较器提供高速操作,典型延时为15 ns。

比较器的输出通过脉冲延展器和数字滤波器模块处理,从而防止比较器响应输入中的意外快速瞬变。图13-1给出了高速模拟比较器模块的框图。DAC模块可在三种模式下工作:斜率生成模式、滞后模式和三角波模式。每种模式均可用于各种电源应用。

**注:** 在任意给定时刻,DACOUT1引脚只能与单个DAC输出相关联。如果多个DACOEN位置1,则DACOUT1引脚将是多个信号的组合。

图 13-1： 高速模拟比较器模块框图



## 13.2 特性概述

- 3个轨到轨模拟比较器
- 每个比较器具有最多4个可选输入源
- 可编程比较器滞后
- 可编程输出极性
- 可产生中断
- 每个模拟比较器均配有专用的脉冲密度调制

### DAC:

- PDM单元后跟一个数字控制多模式多极点RC滤波器
- 多模式多极点RC输出滤波器:
  - 切换模式: 提供最快响应
  - 快速模式: 用于追踪DAC斜率
  - 稳态模式: 提供12位分辨率
- 斜率补偿与每个DAC配合使用, 具有以下工作模式:
  - 斜率生成模式
  - 滞后控制模式
  - 三角波模式
- 高速PWM模块支持的功能包括:
  - PWM占空比控制
  - PWM周期控制
  - PWM故障检测

## 13.3 控制寄存器

DACCTRL1L和DACCTRL2H/L寄存器为DAC模块的共用配置寄存器。

DACxCON、DACxDAT、SLPxCON和SLPxDAT寄存器用于指定各个模块的操作。

# dsPIC33CK64MP105系列

寄存器13-1: DACCTRL1L: DAC控制1低位字寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
DACON	—	DACSIDL	—	—	—	—	—
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CLKSEL1 <sup>(1,3)</sup>	CLKSEL0 <sup>(1,3)</sup>	CLKDIV1 <sup>(1,3)</sup>	CLKDIV0 <sup>(1,3)</sup>	—	FCLKDIV2 <sup>(2)</sup>	FCLKDIV1 <sup>(2)</sup>	FCLKDIV0 <sup>(2)</sup>
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

- bit 15        **DACON:** DAC模块全局使能位  
 1 = 使能所有DAC模块  
 0 = 禁止所有DAC模块并禁止FSCM时钟以降低功耗; 清除任何待处理的斜率模式和/或下溢条件
- bit 14        **未实现:** 读为0
- bit 13        **DACSIDL:** DAC空闲模式停止位  
 1 = 当器件进入空闲模式时, 模块停止工作  
 0 = 在空闲模式下模块继续工作
- bit 12-8     **未实现:** 读为0
- bit 7-6        **CLKSEL[1:0]:** DAC时钟源选择位<sup>(1,3)</sup>  
 11 = FPLLO  
 10 = AFPLLO  
 01 = FVCO/2  
 00 = AFVCO/2
- bit 5-4        **CLKDIV[1:0]:** DAC时钟分频比位<sup>(1,3)</sup>  
 11 = 4分频  
 10 = 3分频 (非均匀占空比)  
 01 = 2分频  
 00 = 不分频
- bit 3        **未实现:** 读为0
- bit 2-0        **FCLKDIV[2:0]:** 比较器滤波器时钟分频比位<sup>(2)</sup>  
 111 = 8分频  
 110 = 7分频  
 101 = 6分频  
 100 = 5分频  
 011 = 4分频  
 010 = 3分频  
 001 = 2分频  
 000 = 不分频

- 注 1:** 为避免出现不可预测的行为, 这些位只能在DACON = 0时更改。  
**2:** 该分频器的输入时钟为由CLKSEL[1:0]位所选的时钟输入再进行二分频。  
**3:** 时钟源和分频器应产生500 MHz的有效DAC时钟输入。



## 寄存器13-2: DACCTRL2H: DAC控制2高位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SSTIME[9:8] <sup>(1)</sup>	
bit 15							bit 8
R/W-1	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0
SSTIME[7:0] <sup>(1)</sup>							
bit 7							bit 0

### 图注:

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

bit 15-10      未实现: 读为0

bit 9-0        **SSTIME[9:0]**: 从切换模式启动到使能稳态滤波器的时间<sup>(1)</sup>

注 1: SSTIME[9:0]的值应大于 TMODTIME[9:0]的值。

## 寄存器13-3: DACCTRL2L: DAC控制2低位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	TMODTIME[9:8] <sup>(1)</sup>	
bit 15							bit 8
R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1	R/W-0	R/W-1
TMODTIME[7:0] <sup>(1)</sup>							
bit 7							bit 0

### 图注:

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

bit 15-10      未实现: 读为0

bit 9-0        **TMODTIME[9:0]**: 切换模式持续时间位<sup>(1)</sup>

注 1: TMODTIME[9:0]的值应小于 SSTIME[9:0]的值。

# dsPIC33CK64MP105系列

**寄存器13-4: DACxCONH: DACx控制高位字寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	TMCB[9:8]	
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
TMCB[7:0]							
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

bit 15-10      **未实现:** 读为0

bit 9-0      **TMCB[9:0]: DACx前沿消隐位**

这些寄存器位指定通过[寄存器13-9](#)中的HCFSEL[3:0]位选择的输入信号发生状态变化(COS)期间DAC输出变化后比较器的消隐周期。

**寄存器13-5: DACxCONL: DACx控制低位字寄存器**

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
DACEN	IRQM1 <sup>(1,2)</sup>	IRQM0 <sup>(1,2)</sup>	—	—	CBE	DACOEN	FLTREN
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
CMPSTAT	CMPPOL	INSEL2	INSEL1	INSEL0	HYSPOL	HYSSEL1	HYSSEL0
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

bit 15      **DACEN:** DACx模块单独使能位

1 = 使能DACx模块  
0 = 禁止DACx模块以降低功耗; 清除任何待处理的斜率模式和/或下溢条件

bit 14-13      **IRQM[1:0]: 中断模式选择位<sup>(1,2)</sup>**

11 = 在检测到上升沿或下降沿时产生中断  
10 = 在检测到下降沿时产生中断  
01 = 在检测到上升沿时产生中断  
00 = 禁止中断

bit 12-11      **未实现:** 读为0

**注 1:** 在操作期间更改这些位可能会产生虚假中断。

**2:** 边沿选择由通过CMPPOL位选择的极性决定。

## 寄存器13-5: DACxCONL: DACx控制低位字寄存器 (续)

bit 10	<b>CBE:</b> 比较器消隐使能位 1 = 在斜率操作完成后的恢复切换期间, 使模拟比较器输出处于消隐状态 (断开) 0 = 禁止模拟比较器的消隐信号; 因此, 模拟比较器输出始终有效
bit 9	<b>DACOEN:</b> DACx输出缓冲器使能位 1 = DACx 模拟电压连接到DACOUT 引脚 0 = DACx 模拟电压未连接到DACOUT 引脚
bit 8	<b>FLTREN:</b> 比较器数字滤波器使能位 1 = 使能数字滤波器 0 = 禁止数字滤波器
bit 7	<b>CMPSTAT:</b> 比较器状态位 比较器输出的当前状态 (包括CMPPOL 选择)。
bit 6	<b>CMPPOL:</b> 比较器输出极性控制位 1 = 输出反相 0 = 输出不反相
bit 5-3	<b>INSEL[2:0]:</b> 比较器输入源选择位 111 = 保留 110 = 保留 101 = 保留 100 = 保留 011 = CMPxD 输入引脚 010 = CMPxC 输入引脚 001 = CMPxB 输入引脚 000 = CMPxA 输入引脚
bit 2	<b>HYPOL:</b> 比较器滞后极性选择位 1 = 在比较器输出的下降沿应用滞后 0 = 在比较器输出的上升沿应用滞后
bit 1-0	<b>HYSSEL[1:0]:</b> 比较器滞后选择位 11 = 45 mV 滞后 10 = 30 mV 滞后 01 = 15 mV 滞后 00 = 未选择任何滞后

- 注 1:** 在操作期间更改这些位可能会产生虚假中断。  
**2:** 边沿选择由通过CMPPOL 位选择的极性决定。

# dsPIC33CK64MP105系列

**寄存器13-6: DACxDATH: DACx数据上限值寄存器**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DACDATH[11:8]			
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
DACDATH[7:0]							
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

bit 15-12      **未实现:** 读为0  
 bit 11-0      **DACDATH[11:0]:** DACx数据上限值位  
                 该寄存器指定DACx数据上限值。有效值为205至3890。

**寄存器13-7: DACxDATL: DACx数据下限值寄存器**

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DACDATL[11:8]			
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
DACDATL[7:0]							
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

bit 15-12      **未实现:** 读为0  
 bit 11-0      **DACDATL[11:0]:** DACx数据下限值位  
                 在滞后模式、斜率生成器模式和三角波模式下, 该寄存器指定DACx模块的数据下限值。有效值为205至3890。

**寄存器13-8: SLPxCONH: DACx斜率控制高位字寄存器**

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
SLOPEN	—	—	—	HME <sup>(1)</sup>	TWME <sup>(2)</sup>	PSE	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

- bit 15      **SLOPEN:** 斜率功能使能/开启位  
1 = 使能斜率功能  
0 = 禁止斜率功能; 禁止斜率累加器以降低功耗
- bit 14-12    **未实现:** 读为0
- bit 11      **HME:** 滞后模式使能位<sup>(1)</sup>  
1 = 使能DACx滞后模式  
0 = 禁止DACx滞后模式
- bit 10      **TWME:** 三角波模式使能位<sup>(2)</sup>  
1 = 使能DACx三角波模式  
0 = 禁止DACx三角波模式
- bit 9        **PSE:** 正斜率模式使能位  
1 = 正斜率模式 (递增)  
0 = 负斜率模式 (递减)
- bit 8-0     **未实现:** 读为0

**注 1:** HME 模式需要用户禁止斜率功能 (SLOPEN = 0)。

**2:** TWME 模式需要用户使能斜率功能 (SLOPEN = 1)。

# dsPIC33CK64MP105系列

**寄存器13-9: SLPxCONL: DACx斜率控制低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
HCFSEL3	HCFSEL2	HCFSEL1	HCFSEL0	SLPSTOPA3	SLPSTOPA2	SLPSTOPA1	SLPSTOPA0
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
SLPSTOPB3	SLPSTOPB2	SLPSTOPB1	SLPSTOPB0	SLPSTRT3	SLPSTRT2	SLPSTRT1	SLPSTRT0
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

**bit 15-12 HCFSEL[3:0]:** 比较器滞后功能输入选择位  
选择的输入信号控制作为PDM DAC数据源的DACx上限 (DACxDATH) 和DACx下限 (DACxDATL) 之间的切换。它修改比较器的极性, 上升沿和下降沿启动LEB计数器开始计数 (寄存器13-4中的TMCB[9:0]位)。

输入选择	源
0101-1111	1
0100	PWM4H
0011	PWM3H
0010	PWM2H
0001	PWM1H
0000	0

**bit 11-8 SLPSTOPA[3:0]:** 斜率停止A信号选择位  
选定的斜率停止A信号与选定的斜率停止B信号进行逻辑或运算, 以终止斜率功能。

斜率停止A信号选择	源
0101-1111	1
0100	PWM4触发信号2
0011	PWM3触发信号2
0010	PWM2触发信号2
0001	PWM1触发信号2
0000	0

**寄存器13-9: SLPxCONL: DACx斜率控制低位字寄存器 (续)**

bit 7-4

**SLPSTOPB[3:0]:** 斜率停止B信号选择位

选定的斜率停止B信号与选定的斜率停止A信号进行逻辑或运算，以终止斜率功能。

斜率停止B信号选择	源
0100-1111	1
0011	CMP3输出
0010	CMP2输出
0001	CMP1输出
0000	0

bit 3-0

**SLPSTRT[3:0]:** 斜率开始信号选择位

斜率开始信号选择	源
0101-1111	1
0100	PWM4触发信号1
0011	PWM3触发信号1
0010	PWM2触发信号1
0001	PWM1触发信号1
0000	0

**寄存器13-10: SLPxDAT: DACx斜率数据寄存器<sup>(1)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPDAT[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
SLPDAT[7:0]							
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

bit 15-0

**SLPDAT[15:0]:** 斜率值位

SLPDATx值采用12.4格式。

**注 1:** 寄存器数据左对齐。

注:



14.0 正交编码器接口 (QEI)

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。更多信息, 请参见《dsPIC33/PIC24 系列参考手册》中的“正交编码器接口 (QEI)” ([www.microchip.com/DS70000601](http://www.microchip.com/DS70000601))。

正交编码器接口 (QEI) 模块提供了与用于获得机械位置数据的增量式编码器的接口。dsPIC33CK64MP105 系列实现了两个 QEI 实例。正交编码器 (也称为增量式编码器或光电编码器) 用于检测旋转运动系统的位置和速度。正交编码器可用于实现多种电机控制应用的闭环控制, 例如开关磁阻 (Switched Reluctance, SR) 电机和交流感应电机 (AC Induction Motor, ACIM)。

典型的正交编码器包括一个装在电机轴上的开槽的轮子和一个用于检测轮子上槽口的发射器/检测器模块。通常, 有三个输出通道, A 相 (QEAx)、B 相 (QEBx)

和索引 (INDXx), 提供有关电机轴运动的信息, 包括距离和方向。

A 相 (QEAx) 和 B 相 (QEBx) 这两个通道通常相互存在 90 度的相位差。A 相和 B 相这两个通道之间的关系是惟一的。如果 A 相超前于 B 相, 那么电机的旋转方向被认为是正向的。如果 A 相滞后于 B 相, 那么电机的旋转方向则被认为是反向的。每机械旋转一圈产生一个索引脉冲, 作为参考用来确定绝对位置。图 14-1 所示为正交编码器接口信号。

编码器产生的正交信号可以有 4 种各不相同的状态 (01、00、10 和 11), 这些状态反映了 QEAx 和 QEBx 之间的关系。图 14-1 给出了一个计数周期中的这些状态。当旋转的方向改变时, 这些状态的顺序与此相反。

每次状态改变 (COS) 时, 正交解码器都会使 32 位递增/递减位置 x 计数器 (POSxCNTH/L) 寄存器递增或递减。当 QEAx 超前于 QEBx 时, 计数器递增; 当 QEBx 超前于 QEAx 时, 计数器递减。

图 14-1: 正交编码器接口信号

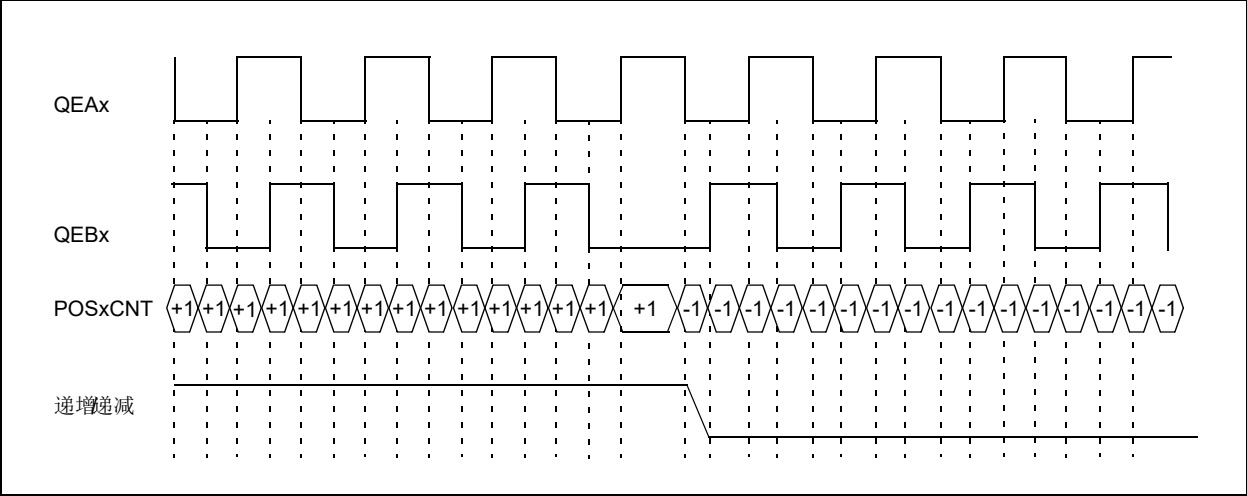


表 14-1 给出了说明正交信号如何解码的真值表。

表 14-1: 正交编码器的真值表

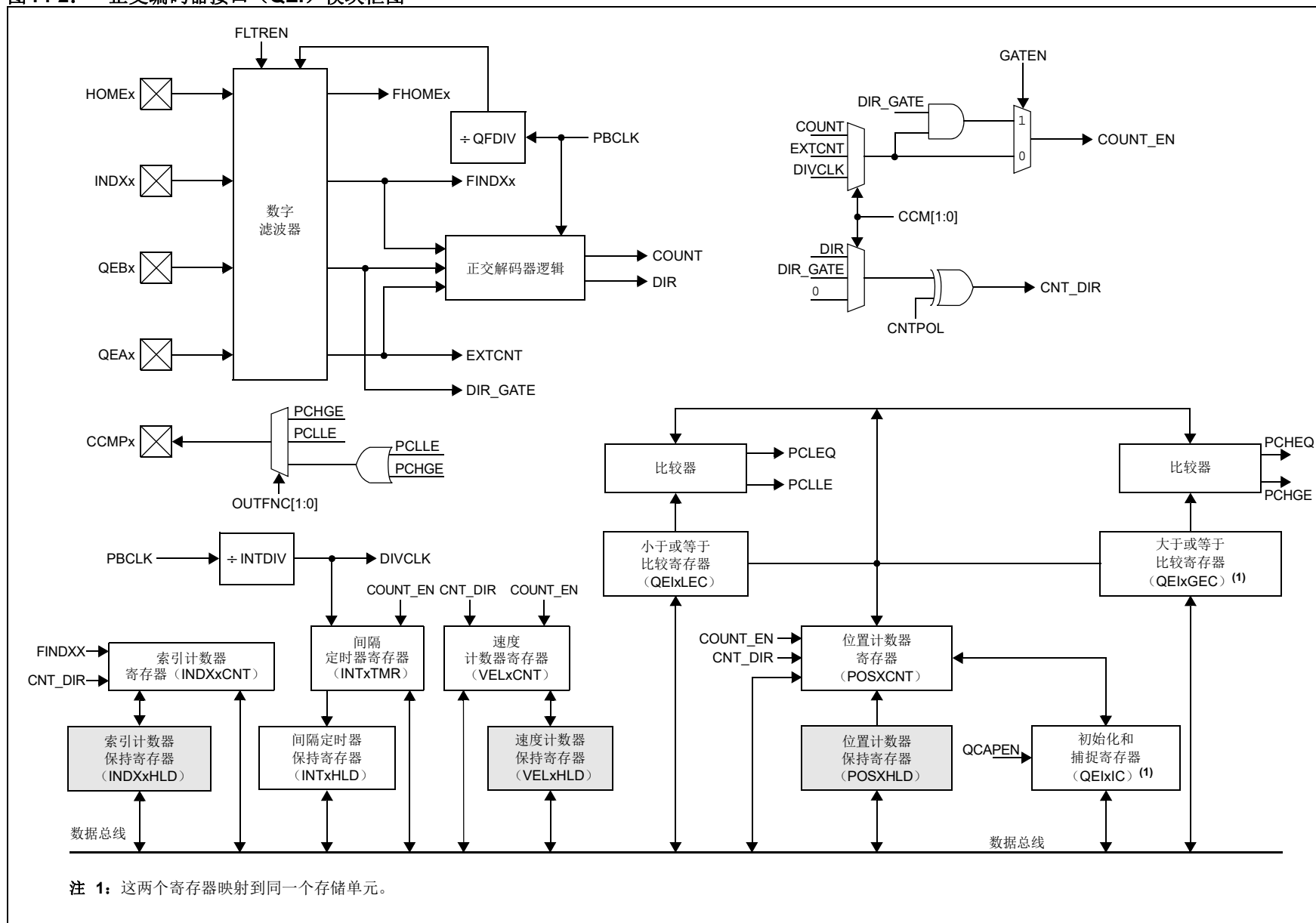
当前正交状态		前一个正交状态		操作
QEA	QEB	QEA	QEB	
1	1	1	1	计数或方向无变化
1	1	1	0	递增计数
1	1	0	1	递减计数
1	1	0	0	无效的状态变化；忽略
1	0	1	1	递减计数
1	0	1	0	计数或方向无变化
1	0	0	1	无效的状态变化；忽略
1	0	0	0	递增计数
0	1	1	1	递增计数
0	1	1	0	无效的状态变化；忽略
0	1	0	1	计数或方向无变化
0	1	0	0	递减计数
0	0	1	1	无效的状态变化；忽略
0	0	1	0	递减计数
0	0	0	1	递增计数
0	0	0	0	计数或方向无变化

图 14-2 给出了 QEI 模块的简化框图。QEI 模块由用于解析 A 相（QEAx）和 B 相（QEBx）信号的解码器逻辑以及用于累加计数值的递增/递减计数器组成。计数器脉冲在正交状态发生改变时产生。计数方向信息必须保存在一个寄存器中，直到检测到方向变化为止。模块还包含了一些数字噪声滤波器，用于对输入信号进行调理。

QEI 模块包括以下主要特性：

- 四个输入引脚：两个相信号、一个索引脉冲和一个归位脉冲
- 输入端的可编程数字噪声滤波器
- 正交解码器，提供计数器脉冲和计数方向
- 计数方向状态
- 4x 计数分辨率
- 索引（INDXx）脉冲，用于复位位置计数器
- 通用 32 位定时器/计数器模式
- 由 QEI 或计数器事件产生的中断
- 32 位速度计数器
- 32 位位置计数器
- 32 位索引脉冲计数器
- 32 位间隔定时器
- 32 位位置初始化/捕捉寄存器
- 32 位小于和大于比较寄存器
- 外部递增/递减计数模式
- 外部门控计数模式
- 外部门控定时器模式
- 间隔定时器模式

图14-2: 正交编码器接口 (QE1) 模块框图



# dsPIC33CK64MP105系列

## 14.1 QEI控制/状态寄存器

寄存器14-1: QEIXCON: QEIX控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIEN	—	QEISIDL	PIMOD2 <sup>(1,5)</sup>	PIMOD1 <sup>(1,5)</sup>	PIMOD0 <sup>(1,5)</sup>	IMV1 <sup>(2)</sup>	IMV0 <sup>(2)</sup>
bit 15							bit 8

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	INTDIV2 <sup>(3)</sup>	INTDIV1 <sup>(3)</sup>	INTDIV0 <sup>(3)</sup>	CNTPOL	GATEN	CCM1	CCM0
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 15        **QEIEN:** 正交编码器接口模块使能位  
              1 = 使能模块计数器  
              0 = 禁止模块计数器, 但可读取或写入 SFR
- bit 14        **未实现:** 读为0
- bit 13        **QEISIDL:** QEI空闲模式停止位  
              1 = 当器件进入空闲模式时, 模块停止工作  
              0 = 在空闲模式下模块继续工作
- bit 12-10     **PIMOD[2:0]:** 位置计数器初始化模式选择位<sup>(1,5)</sup>  
              111 = 为位置计数器选择模计数模式, 每个索引事件都会复位位置计数器<sup>(4)</sup>  
              110 = 为位置计数器选择模计数模式  
              101 = 位置计数器等于 QEIXGEC 寄存器时, 复位位置计数器  
              100 = 归位事件后的第二个索引事件用 QEIXIC 寄存器的内容初始化位置计数器  
              011 = 归位事件后的第一个索引事件用 QEIXIC 寄存器的内容初始化位置计数器  
              010 = 下一个索引输入事件用 QEIXIC 寄存器的内容初始化位置计数器  
              001 = 每个索引输入事件都会复位位置计数器  
              000 = 索引输入事件不影响位置计数器
- bit 9-8       **IMV[1:0]:** 索引匹配值位<sup>(2)</sup>  
              11 = QEBx = 1 且 QEAx = 1 时发生索引匹配  
              10 = QEBx = 1 且 QEAx = 0 时发生索引匹配  
              01 = QEBx = 0 且 QEAx = 1 时发生索引匹配  
              00 = QEBx = 0 且 QEAx = 0 时发生索引匹配
- bit 7        **未实现:** 读为0

- 注 1: 当 CCMx = 10 或 CCMx = 11 时, 所有 QEI 计数器都将用作定时器, PIMOD[2:0] 位将被忽略。
- 2: 当 CCMx = 00 且 QEAx 和 QEBx 值与索引匹配值 (Index Match Value, IMV) 相匹配时, POSxCNT 和 POSxCNTL 寄存器将复位。
- 3: 所选时钟速率应至少为预期最大正交计数速率的 2 倍。
- 4: 并非所有器件均支持该模式。
- 5: 在 PIMODx 模式 2 到 7 期间, 必须清零 QCAPEN 和 HCAPEN 位, 以确保正常工作。并非所有器件均支持 HCAPEN。

## 寄存器14-1: QEIXCON: QEIX控制寄存器 (续)

- bit 6-4     **INTDIV[2:0]:** 定时器输入时钟预分频比选择位<sup>(3)</sup>  
(间隔定时器、主定时器 (位置计数器)、速度计数器和索引计数器内部时钟分频比选择)
- 111 = 1:256 预分频值  
110 = 1:64 预分频值  
101 = 1:32 预分频值  
100 = 1:16 预分频值  
011 = 1:8 预分频值  
010 = 1:4 预分频值  
001 = 1:2 预分频值  
000 = 1:1 预分频值
- bit 3     **CNTPOL:** 位置和索引计数器/定时器方向选择位  
1 = 除非通过外部递增/递减信号修改, 否则计数器方向为负  
0 = 除非通过外部递增/递减信号修改, 否则计数器方向为正
- bit 2     **GATEN:** 外部计数门控使能位  
1 = 外部门控信号控制位置计数器的操作  
0 = 外部门控信号不影响位置计数器的操作
- bit 1-0     **CCM[1:0]:** 计数器控制模式选择位  
11 = 内部定时器模式  
10 = 采用外部门控的外部时钟计数模式  
01 = 采用外部递增/递减的外部时钟计数模式  
00 = 正交编码器模式

- 注 1:** 当CCMx = 10或CCMx = 11时, 所有QEI计数器都将用作定时器, PIMOD[2:0]位将被忽略。
- 2:** 当CCMx = 00且QEAx和QEBx值与索引匹配值 (Index Match Value, IMV) 相匹配时, POSxCNTH和POSxCNTL寄存器将复位。
- 3:** 所选时钟速率应至少为预期最大正交计数速率的2倍。
- 4:** 并非所有器件均支持该模式。
- 5:** 在PIMODx模式2到7期间, 必须清零QCAPEN和HCAPEN位, 以确保正常工作。并非所有器件均支持HCAPEN。

# dsPIC33CK64MP105系列

**寄存器14-2: QEIXIOC: QEIX I/O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
QCAPEN	FLTREN	QFDIV2	QFDIV1	QFDIV0	OUTFNC1	OUTFNC0	SWPAB
bit 15						bit 8	

R/W-0	R/W-0	R/W-0	R/W-0	R-x	R-x	R-x	R-x
HOMPOL	IDXPOL	QEBPOL	QEAPOL	HOME	INDEX	QEB	QEA
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 15      **QCAPEN:** QEIX位置计数器输入捕捉使能位  
 1 = HOME<sub>x</sub>输入事件（正边沿）触发位置捕捉事件（HCAPEN必须清零）  
 0 = HOME<sub>x</sub>输入事件（正边沿）不触发位置捕捉事件
- bit 14      **FLTREN:** QEAX/QEBX/INDX<sub>x</sub>/HOME<sub>x</sub>数字滤波器使能位  
 1 = 使能输入引脚数字滤波器  
 0 = 禁止（旁路）输入引脚数字滤波器
- bit 13-11   **QFDIV[2:0]:** QEAX/QEBX/INDX<sub>x</sub>/HOME<sub>x</sub>数字输入滤波器时钟分频比选择位  
 111 = 1:256时钟分频比  
 110 = 1:64时钟分频比  
 101 = 1:32时钟分频比  
 100 = 1:16时钟分频比  
 011 = 1:8时钟分频比  
 010 = 1:4时钟分频比  
 001 = 1:2时钟分频比  
 000 = 1:1时钟分频比
- bit 10-9   **OUTFNC[1:0]:** QEIX模块输出功能模式选择位  
 11 = 当POSxCNT ≤ QEIXLEC或POSxCNT ≥ QEIXGEC时, QEICMP<sub>x</sub>引脚变为高电平  
 10 = 当POSxCNT ≤ QEIXLEC时, QEICMP<sub>x</sub>引脚变为高电平  
 01 = 当POSxCNT ≥ QEIXGEC时, QEICMP<sub>x</sub>引脚变为高电平  
 00 = 禁止输出
- bit 8      **SWPAB:** QEAX和QEBX输入交换位  
 1 = 在正交解码器逻辑前交换QEAX和QEBX  
 0 = 不交换QEAX和QEBX
- bit 7      **HOMPOL:** HOME<sub>x</sub>输入极性选择位  
 1 = 输入反相  
 0 = 输入不反相
- bit 6      **IDXPOL:** INDX<sub>x</sub>输入极性选择位  
 1 = 输入反相  
 0 = 输入不反相
- bit 5      **QEBPOL:** QEBX输入极性选择位  
 1 = 输入反相  
 0 = 输入不反相
- bit 4      **QEAPOL:** QEAX输入极性选择位  
 1 = 输入反相  
 0 = 输入不反相
- bit 3      **HOME:** 极性控制后HOME<sub>x</sub>输入引脚的状态位（只读）  
 1 = 如果HOMPOL位设置为0, 则引脚为逻辑1; 如果HOMPOL位设置为1, 则引脚为逻辑0  
 0 = 如果HOMPOL位设置为0, 则引脚为逻辑0; 如果HOMPOL位设置为1, 则引脚为逻辑1

**寄存器14-2: QEIXIOC: QEIX I/O控制寄存器 (续)**

- bit 2 **INDEX:** 极性控制后 INDXX 输入引脚的状态位 (只读)  
 1 = 如果 IDXPOL 位设置为 0, 则引脚为逻辑 1; 如果 IDXPOL 位设置为 1, 则引脚为逻辑 0  
 0 = 如果 IDXPOL 位设置为 0, 则引脚为逻辑 0; 如果 IDXPOL 位设置为 1, 则引脚为逻辑 1
- bit 1 **QEB:** 极性控制和 SWPAB 引脚交换后 QEBx 输入引脚的状态位 (只读)  
 1 = 如果 QEBPOL 位设置为 0 且 SWPAB 位设置为 0, 则物理引脚 QEBx 为逻辑 1;  
 如果 QEBPOL 位设置为 1 且 SWPAB 位设置为 0, 则物理引脚 QEBx 为逻辑 0;  
 如果 QEBPOL 位设置为 0 且 SWPAB 位设置为 1, 则物理引脚 QEAx 为逻辑 1;  
 如果 QEBPOL 位设置为 1 且 SWPAB 位设置为 1, 则物理引脚 QEAx 为逻辑 0  
 0 = 如果 QEBPOL 位设置为 0 且 SWPAB 位设置为 0, 则物理引脚 QEBx 为逻辑 0;  
 如果 QEBPOL 位设置为 1 且 SWPAB 位设置为 0, 则物理引脚 QEBx 为逻辑 1;  
 如果 QEBPOL 位设置为 0 且 SWPAB 位设置为 1, 则物理引脚 QEAx 为逻辑 0;  
 如果 QEBPOL 位设置为 1 且 SWPAB 位设置为 1, 则物理引脚 QEAx 为逻辑 1
- bit 0 **QEA:** 极性控制和 SWPAB 引脚交换后 QEAx 输入引脚的状态位 (只读)  
 1 = 如果 QEAPOL 位设置为 0 且 SWPAB 位设置为 0, 则物理引脚 QEAx 为逻辑 1;  
 如果 QEAPOL 位设置为 1 且 SWPAB 位设置为 0, 则物理引脚 QEAx 为逻辑 0;  
 如果 QEAPOL 位设置为 0 且 SWPAB 位设置为 1, 则物理引脚 QEBx 为逻辑 1;  
 如果 QEAPOL 位设置为 1 且 SWPAB 位设置为 1, 则物理引脚 QEBx 为逻辑 0  
 0 = 如果 QEAPOL 位设置为 0 且 SWPAB 位设置为 0, 则物理引脚 QEAx 为逻辑 0;  
 如果 QEAPOL 位设置为 1 且 SWPAB 位设置为 0, 则物理引脚 QEAx 为逻辑 1;  
 如果 QEAPOL 位设置为 0 且 SWPAB 位设置为 1, 则物理引脚 QEBx 为逻辑 0;  
 如果 QEAPOL 位设置为 1 且 SWPAB 位设置为 1, 则物理引脚 QEBx 为逻辑 1

**寄存器14-3: QEIXIOCH: QEIX I/O控制高位字寄存器**

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

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	HCAPEN
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 15-1 **未实现:** 读为 0

bit 0 **HCAPEN:** 通过归位事件触发位置计数器输入捕捉使能位  
 1 = HOME<sub>x</sub> 输入事件 (正边沿) 触发位置捕捉事件  
 0 = HOME<sub>x</sub> 输入事件 (正边沿) 不触发位置捕捉事件

# dsPIC33CK64MP105系列

**寄存器14-4: QEIXSTAT: QEIX状态寄存器**

U-0	U-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0
—	—	PCHEQIRQ	PCHEQIEN	PCLEQIRQ	PCLEQIEN	POSOVIRQ	POSOVIEN
bit 15						bit 8	

HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0	HS/R/C-0	R/W-0
PCIIIRQ <sup>(1)</sup>	PCIIEN	VELOVIRQ	VELOVIEN	HOMIRQ	HOMIEN	IDXIRQ	IDXIEN
bit 7						bit 0	

<b>图注:</b>	C = 可清零位	HS = 硬件置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 15-14     **未实现:** 读为0
- bit 13     **PCHEQIRQ:** 位置计数器大于比较状态位  
1 = POSxCNT ≥ QEIXGEC  
0 = POSxCNT < QEIXGEC
- bit 12     **PCHEQIEN:** 位置计数器大于比较中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 11     **PCLEQIRQ:** 位置计数器小于比较状态位  
1 = POSxCNT ≤ QEIXLEC  
0 = POSxCNT > QEIXLEC
- bit 10     **PCLEQIEN:** 位置计数器小于比较中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 9     **POSOVIRQ:** 位置计数器溢出状态位  
1 = 发生了溢出  
0 = 未发生溢出
- bit 8     **POSOVIEN:** 位置计数器溢出中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 7     **PCIIIRQ:** 位置计数器（归位）初始化过程完成状态位<sup>(1)</sup>  
1 = POSxCNT已重新初始化  
0 = POSxCNT未重新初始化
- bit 6     **PCIIEN:** 位置计数器（归位）初始化过程完成中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 5     **VELOVIRQ:** 速度计数器溢出状态位  
1 = 发生了溢出  
0 = 未发生溢出
- bit 4     **VELOVIEN:** 速度计数器溢出中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 3     **HOMIRQ:** 归位事件状态的状态标志位  
1 = 发生了归位事件  
0 = 未发生归位事件

**注 1:** 该状态位仅适用于PIMOD[2:0]模式011和100。



## 寄存器14-4: QElxSTAT: QEIx状态寄存器 (续)

bit 2	<b>HOMIEN:</b> 归位输入事件中断允许位 1 = 允许中断 0 = 禁止中断
bit 1	<b>IDXIRQ:</b> 索引事件状态的状态标志位 1 = 发生了索引事件 0 = 未发生索引事件
bit 0	<b>IDXIEN:</b> 索引输入事件中断允许位 1 = 允许中断 0 = 禁止中断

**注 1:** 该状态位仅适用于PIMOD[2:0]模式011和100。

# dsPIC33CK64MP105系列

寄存器14-5:      **POSxCNTL: 位置x计数器低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[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
POSCNT[7:0]							
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 15-0      **POSCNT[15:0]:** 用于构成32位位置计数器寄存器 (POSxCNT) 的低位字的位

寄存器14-6:      **POSxCNTH: 位置x计数器高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSCNT[31:24]							
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
POSCNT[23:16]							
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 15-0      **POSCNT[31:16]:** 用于构成32位位置计数器寄存器 (POSxCNT) 的高位字的位

## 寄存器14-7: POSxHLD: 位置x计数器保持寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POSHLD[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
POSHLD[7:0]							
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 15-0     **POSHLD[15:0]:** 用于读/写位置x计数器高位字寄存器 (POSxCNTH) 的保持寄存器位

# dsPIC33CK64MP105系列

## 寄存器14-8: VELxCNT: 速度x计数器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[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
VELCNT[7:0]							
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 15-0 VELCNT[15:0]: 速度计数器位

## 寄存器14-9: VELxCNTH: 速度x计数器高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELCNT[31:24]							
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
VELCNT[23:16]							
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 15-0 VELCNT[31:16]: 速度计数器位

**寄存器14-10: VELxHLD: 速度x计数器保持寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
VELHLD[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
VELHLD[7:0]							
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 15-0 **VELHLD[15:0]:** 用于读/写速度计数器寄存器 (VELxCNT) 的保持位

# dsPIC33CK64MP105系列

寄存器14-11: INTxTMRL: 间隔x定时器低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[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
INTTMR[7:0]							
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 15-0 INTTMR[15:0]: 用于构成32位间隔定时器寄存器 (INTxTMR) 的低位字的位

寄存器14-12: INTxTMRH: 间隔x定时器高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTTMR[31:24]							
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
INTTMR[23:16]							
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 15-0 INTTMR[31:16]: 用于构成32位间隔定时器寄存器 (INTxTMR) 的高位字的位

## 寄存器14-13: INTXxHLDL: 间隔x定时器保持低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[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
INTHLD[7:0]							
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 15-0 **INTHLD[15:0]:** 用于构成32位间隔定时器保持寄存器 (INTxHLD) 的低位字的位

## 寄存器14-14: INTXxHLDH: 间隔x定时器保持高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTHLD[31:24]							
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
INTHLD[23:16]							
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 15-0 **INTHLD[31:16]:** 用于构成32位间隔定时器保持寄存器 (INTxHLD) 的高位字的位

# dsPIC33CK64MP105系列

## 寄存器14-15: INDXXCNTL: 索引x计数器低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[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
INDXCNT[7:0]							
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 15-0 **INDXCNT[15:0]**: 用于构成32位索引x计数器寄存器 (INDXXCNT) 的低位字的位

## 寄存器14-16: INDXXCNTH: 索引x计数器高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXCNT[31:24]							
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
INDXCNT[23:16]							
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 15-0 **INDXCNT[31:16]**: 用于构成32位索引x计数器寄存器 (INDXXCNT) 的高位字的位



寄存器14-17: **INDXxHLD**: 索引x计数器保持寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INDXHLD[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
INDXHLD[7:0]							
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 15-0 **INDXHLD[15:0]**: 用于读/写索引x计数器高位字寄存器 (INDXxCNTH) 的保持寄存器位

# dsPIC33CK64MP105系列

## 寄存器14-18: QEIXICL: QEIX初始化/捕捉低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIIC[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
QEIIIC[7:0]							
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 15-0 **QEIIIC[15:0]:** 用于构成32位初始化/捕捉寄存器 (QEIXIC) 的低位字的位

## 寄存器14-19: QEIXICH: QEIX初始化/捕捉高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIIIC[31:24]							
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
QEIIIC[23:16]							
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 15-0 **QEIIIC[31:16]:** 用于构成32位初始化/捕捉寄存器 (QEIXIC) 的高位字的位

## 寄存器14-20: QEIxLECL: QEIx小于或等于比较低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[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
QEILEC[7:0]							
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 15-0 **QEILEC[15:0]:** 用于构成32位小于或等于比较寄存器 (QEILEC) 的低位字的位

## 寄存器14-21: QEIxLECH: QEIx小于或等于比较高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEILEC[31:24]							
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
QEILEC[23:16]							
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 15-0 **QEILEC[31:16]:** 用于构成32位小于或等于比较寄存器 (QEILEC) 的高位字的位

# dsPIC33CK64MP105系列

寄存器14-22: **QEIxGECL: QEIx大于或等于比较低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[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
QEIGEC[7:0]							
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 15-0 **QEIGEC[15:0]:** 用于构成32位大于或等于比较寄存器 (QEIGEC) 的低位字的位

寄存器14-23: **QEIxGECH: QEIx大于或等于比较高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
QEIGEC[31:24]							
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
QEIGEC[23:16]							
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 15-0 **QEIGEC[31:16]:** 用于构成32位大于或等于比较寄存器 (QEIGEC) 的高位字的位

## 15.0 通用异步收发器（UART）

**注 1：** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“多协议通用异步收发器（UART）模块”（[www.microchip.com/DS70005288](http://www.microchip.com/DS70005288)）。

通用异步收发器（Universal Asynchronous Receiver Transmitter, UART）是一种灵活的串行通信外设，用于将 dsPIC<sup>®</sup> 器件与其他设备（包括计算机和外设）相连接。UART 是一个全双工异步通信通道，可用于实现 RS-232 和 RS-485 等协议。UART 还支持以下硬件扩展：

- LIN/J2602
- IrDA<sup>®</sup>
- 直接矩阵架构（Direct Matrix Architecture, DMX）
- 智能卡

UART 的主要特性有：

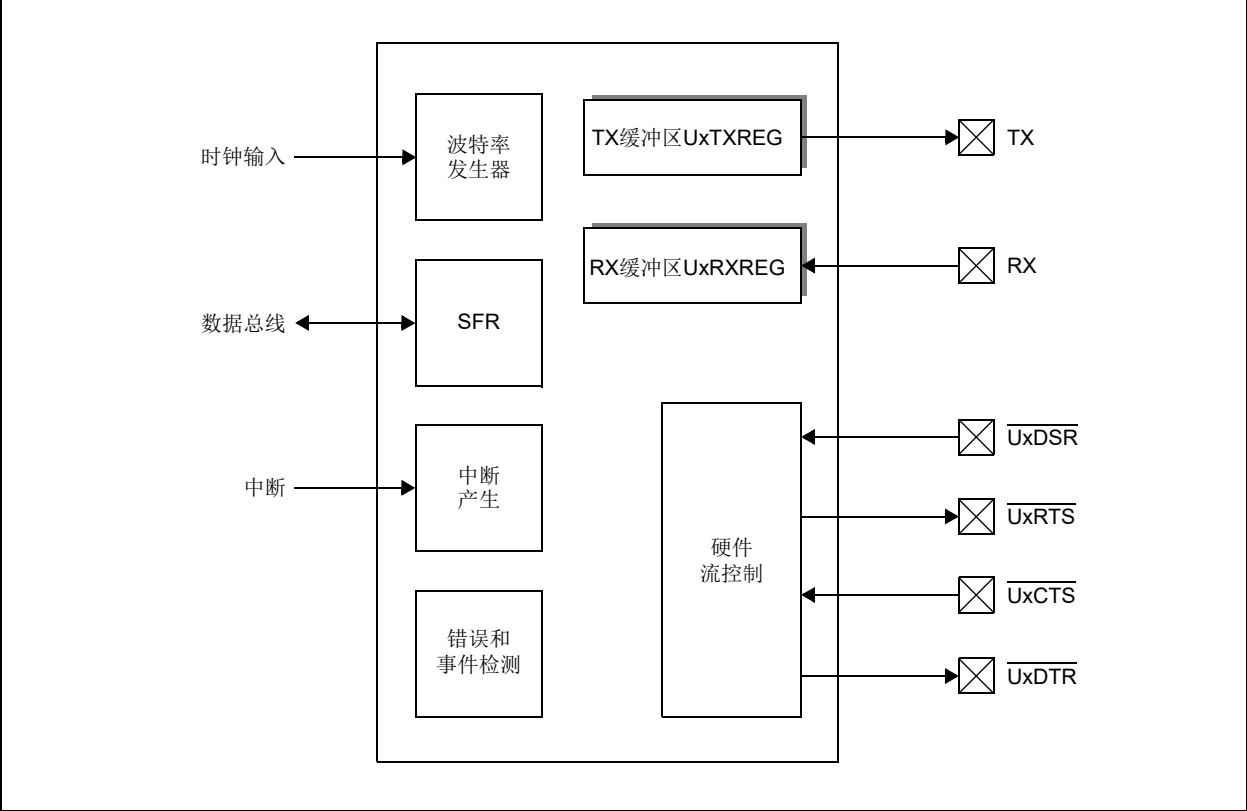
- 全双工或半双工操作
- 最大 8 字节深的 TX 和 RX 先进先出（First-In, First-Out, FIFO）缓冲区
- 8 位或 9 位数据宽度
- 可配置停止位长度
- 流控制
- 自动波特率校准
- 奇偶校验错误、帧错误和缓冲区溢出错误检测
- 地址检测
- 暂停（break）字符发送
- 发送和接收极性控制
- 曼彻斯特编码器/解码器
- 在休眠模式下工作
- 接收同步暂停字符中断将器件从休眠模式唤醒

15.1 架构概述

UART使用最大8字节深的先进先出（FIFO）缓冲区与器件引脚之间传输数据字节。用户软件可通过特殊功能

寄存器（SFR）获取缓冲区状态以及数据。UART实现了多个中断通道来处理发送、接收和错误事件。图15-1给出了UART的简化框图。

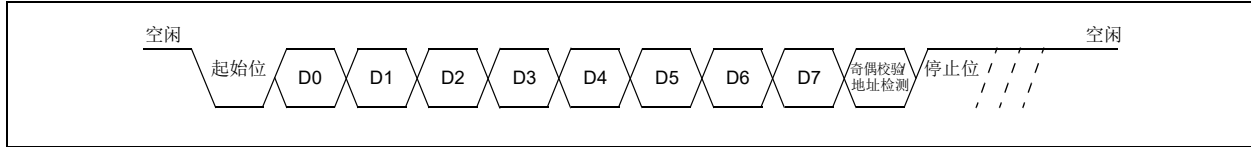
图15-1: UARTx简化框图



## 15.2 字符帧

图15-2给出了典型的UART字符帧。空闲状态为高电平，“起始”条件由下降沿指示。起始位后依次是数据位、奇偶校验/地址检测和停止位，具体由所选的MOD[3:0]（UxMODE[3:0]）位定义。

图15-2: UART字符帧



## 15.3 数据缓冲区

发送和接收功能均使用缓冲区来存储移入/移出引脚的数据。这些缓冲区为FIFO，分别通过读取SFR、UxTXREG和UxRXREG进行访问。每个数据缓冲区有多个与其操作相关的标志，以供软件读取状态。此外，还可基于缓冲区中可用的空间来配置中断。可清空发送和接收缓冲区，并使用相关的TX/RX发送缓冲区空状态位UTXBE（UxSTAH[5]）和URXBE（UxSTAH[1]）复位其指针。

## 15.4 协议扩展

UART为LIN/J2602、IrDA®、DMX和智能卡协议扩展提供硬件支持，从而降低软件开销。协议扩展通过向MOD[3:0]（UxMODE[3:0]）选择位中写入相应的值来使能，并且可使用UARTx时序参数寄存器UxP1（寄存器15-9）、UxP2（寄存器15-10）、UxP3（寄存器15-11）和UxP3H（寄存器15-12）进一步配置。相应的章节中将讨论有关操作和使用的详细信息。

# dsPIC33CK64MP105系列

## 15.5 UART控制/状态寄存器

寄存器15-1: **UxMODE: UARTx配置寄存器**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	HC/R/W-0 <sup>(1)</sup>
UARTEN	—	USIDL	WAKE	RXBIMD	—	BRKOVr	UTXBRK
bit 15							bit 8

R/W-0	HC/R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRGH	ABAUd	UTXEN	URXEN	MOD3	MOD2	MOD1	MOD0
bit 7							bit 0

图注:	HC = 硬件清零位		
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 15 **UARTEN:** UART 使能位  
1 = UART 已准备好进行发送和接收  
0 = UART 状态机、FIFO 缓冲区指针和计数器均复位; 寄存器可读写
- bit 14 **未实现:** 读为0
- bit 13 **USIDL:** UART 空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12 **WAKE:** 唤醒使能位  
1 = 模块将继续采样RX引脚——在下降沿产生中断, 在下一个上升沿由硬件将该位清零; 如果ABAUd置1, 则自动波特率检测 (Auto-Baud Detection, ABD) 将立即开始  
0 = 不监视Rx引脚, 也不检测上升沿
- bit 11 **RXBIMD:** 接收暂停字符中断模式位  
1 = 检测到至少23 (DMX) /11 (异步或LIN/J2602) 个低电平周期时RXBKIF标志置1  
0 = 暂停字符在至少持续23/11个位周期的低电平阶段后发生低电平到高电平跳变时RXBKIF标志置1
- bit 10 **未实现:** 读为0
- bit 9 **BRKOVr:** 发送暂停字符软件改写位  
改写TX数据线:  
1 = 激活TX线 (UTXINV = 0时输出0, UTXINV = 1时输出1)  
0 = TX线由移位寄存器驱动
- bit 8 **UTXBRK:** UART 发送暂停字符位<sup>(1)</sup>  
1 = 在下次发送时发送同步暂停字符; 完成时由硬件清零  
0 = 禁止或已完成同步暂停字符的发送
- bit 7 **BRGH:** 高波特率选择位  
1 = 高速: 波特率为baudclk/4  
0 = 低速: 波特率为baudclk/16
- bit 6 **ABAUd:** 自动波特率检测使能位 (MOD[3:0] = 1xxx时为只读位)  
1 = 使能对下一个字符的波特率测量——需要接收同步字段 (55h); 完成时由硬件清零  
0 = 禁止或已完成波特率测量

注 1: DMX和LIN模式下为R/HS/HC。



## 寄存器15-1: UxMODE: UARTx配置寄存器 (续)

bit 5	<b>UTXEN:</b> UART发送使能位 1 = 使能发送——自动波特率检测期间除外 0 = 禁止发送——所有发送计数器、指针和状态机均复位; TX缓冲区未清空, 状态位未复位
bit 4	<b>URXEN:</b> UART接收使能位 1 = 使能接收——自动波特率检测期间除外 0 = 禁止接收——所有接收计数器、指针和状态机均复位; RX缓冲区未清空, 状态位未复位
bit 3-0	<b>MOD[3:0]:</b> UART模式位 Other = 保留 1111 = 智能卡 1110 = IrDA® 1101 = 保留 1100 = LIN主/从模式 1011 = 仅LIN从模式 1010 = DMX 1001 = 保留 1000 = 保留 0111 = 保留 0110 = 保留 0101 = 保留 0100 = 异步9位UART, 带地址检测, 第9位 = 1 (表示地址) 0011 = 异步8位UART, 不带地址检测, 第9位用作偶校验位 0010 = 异步8位UART, 不带地址检测, 第9位用作奇校验位 0001 = 异步7位UART 0000 = 异步8位UART

注 1: DMX和LIN模式下为R/HS/HC。

# dsPIC33CK64MP105系列

寄存器15-2: UxMODEH: UARTx配置高位字寄存器

R/W-0	R-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SLPEN	ACTIVE	—	—	BCLKMOD	BCLKSEL1	BCLKSEL0	HALFDPLX
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
RUNOVF	URXINV	STSEL1	STSEL0	C0EN	UTXINV	FLO1	FLO0
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 15      **SLPEN:** 休眠期间运行使能位  
 1 = UART BRG 时钟在休眠模式下运行  
 0 = UART BRG 时钟在休眠模式下关闭
- bit 14      **ACTIVE:** UART 运行状态位  
 1 = UART 时钟请求处于活动状态 (用户无法更新 UxMODE/UxMODEH 寄存器)  
 0 = UART 时钟请求未处于活动状态 (用户可更新 UxMODE/UxMODEH 寄存器)
- bit 13-12   **未实现:** 读为0
- bit 11      **BCLKMOD:** 波特率时钟生成模式选择位  
 1 = 使用小数波特率生成模式  
 0 = 使用传统 x 分频计数器来生成波特率时钟 (x = 4 或 16, 具体取决于 BRGH 位)
- bit 10-9    **BCLKSEL[1:0]:** 波特率时钟源选择位  
 11 = AFVCO/3  
 10 = Fosc  
 01 = 保留  
 00 = Fosc/2 (FP)
- bit 8       **HALFDPLX:** UART 半双工选择模式位  
 1 = 半双工模式: UxTX 发送时驱动为输出, 空闲时驱动为三态  
 0 = 全双工模式: 当 UARTEN 和 UTXEN 均置1时, UxTX 始终驱动为输出
- bit 7       **RUNOVF:** 溢出条件期间运行模式位  
 1 = 当检测到溢出错误 (OERR) 条件时, RX 移位寄存器继续运行以保持与传入 RX 数据同步; 如果 UxRXREG 已满, 则不会向其传送任何数据 (即, 不会改写任何 UxRXREG 数据)  
 0 = 当检测到溢出错误 (OERR) 条件时, RX 移位寄存器停止接受新数据 (传统模式)
- bit 6       **URXINV:** UART 接收极性位  
 1 = RX 极性翻转; 空闲状态为低电平  
 0 = 输入不反相; 空闲状态为高电平
- bit 5-4     **STSEL[1:0]:** 停止位数选择位  
 11 = 发送2个停止位, 接收时检查1个停止位  
 10 = 发送2个停止位, 接收时检查2个停止位  
 01 = 发送1.5个停止位, 接收时检查1.5个停止位  
 00 = 发送1个停止位, 接收时检查1个停止位
- bit 3       **C0EN:** 使能传统校验和 (C0) 发送和接收位  
 1 = 校验和模式1 (在 LIN 模式下使用增强型 LIN 校验和; 在所有其他模式下将所有 TX/RX 字相加)  
 0 = 校验和模式0 (在 LIN 模式下使用传统 LIN 校验和; 在所有其他模式下不使用)

## 寄存器15-2: UxMODEH: UARTx配置高位字寄存器 (续)

- bit 2      **UTXINV:** UART 发送极性位  
1 = TX 极性翻转; TX 在空闲状态下为低电平  
0 = 输出数据不反相; TX 输出在空闲状态下为高电平
- bit 1-0    **FLO[1:0]:** 流控制使能位 (仅在MOD[3:0] = 0xxx时有效)  
11 = 保留  
10 =  $\overline{\text{RTS-DSR}}$  (对于TX侧) /  $\overline{\text{CTS-DTR}}$  (对于RX侧) 硬件流控制  
01 = XON/XOFF 软件流控制  
00 = 流控制关闭

# dsPIC33CK64MP105系列

## 寄存器15-3: UxSTA: UARTx状态寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXMTIE	PERIE	ABDOVE	CERIE	FERIE	RXBKIE	OERIE	TXCIE
bit 15						bit 8	

R-1	R-0	HS/R/W-0	HS/R/W-0	R-0	HS/R/W-0	HS/R/W-0	R/W-0
TRMT	PERR	ABDOVF	CERIF	FERR	RXBKIF	OERR	TXCIF
bit 7						bit 0	

<b>图注:</b>		HS = 硬件置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 15      **TXMTIE:** 发送移位寄存器空中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 14      **PERIE:** 奇偶校验错误中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 13      **ABDOVE:** 自动波特率采集中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 12      **CERIE:** 校验和错误中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 11      **FERIE:** 帧错误中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 10      **RXBKIE:** 接收暂停字符中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 9        **OERIE:** 接收缓冲区溢出中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 8        **TXCIE:** 发送冲突中断允许位  
1 = 允许中断  
0 = 禁止中断
- bit 7        **TRMT:** 发送移位寄存器空中断标志位 (只读)  
1 = 发送移位寄存器 (Transmit Shift Register, TSR) 为空 (如果STPMD = 1, 则在最后一个停止位结束时触发中断; 如果STPMD = 0, 则在第一个停止位的中间位置触发中断)  
0 = 发送移位寄存器不为空
- bit 6        **PERR:** 奇偶校验错误/接收到地址/转发帧中断标志位  
LIN和奇偶校验模式:  
1 = 检测到奇偶校验错误  
0 = 未检测到奇偶校验错误  
地址模式:  
1 = 接收到地址  
0 = 未检测到地址  
所有其他模式:  
不使用。

## 寄存器15-3: UxSTA: UARTx状态寄存器 (续)

bit 5	<b>ABDOVF:</b> 自动波特率采集中断标志位 (必须由软件清零) 1 = BRG 在自动波特率采集序列期间计满返回 (必须由软件清零) 0 = BRG 尚未在自动波特率采集序列期间计满返回
bit 4	<b>CERIF:</b> 校验和错误中断标志位 (必须由软件清零) 1 = 校验和错误 0 = 无校验和错误
bit 3	<b>FERR:</b> 帧错误中断标志位 1 = 帧错误: 与缓冲区中最顶端字符对应的停止位的电平与期望值相反; 通过含有所接收字符的缓冲区传播 0 = 无帧错误
bit 2	<b>RXBKIF:</b> 接收暂停字符中断标志位 (必须由软件清零) 1 = 接收到暂停字符 0 = 未检测到暂停字符
bit 1	<b>OERR:</b> 接收缓冲区溢出中断标志位 (必须由软件清零) 1 = 接收缓冲区溢出 0 = 接收缓冲区未溢出
bit 0	<b>TXCIF:</b> 发送冲突中断标志位 (必须由软件清零) 1 = 发送的字与接收的字不相等 0 = 发送的字与接收的字相等

# dsPIC33CK64MP105系列

**寄存器15-4: UxSTAH: UARTx状态高位字寄存器**

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	UTXISEL2	UTXISEL1	UTXISEL0	—	URXISEL2 <sup>(1)</sup>	URXISEL1 <sup>(1)</sup>	URXISEL0 <sup>(1)</sup>
bit 15				bit 8			

HS/R/W-0	R/W-0	R/S-1	R-0	R-1	R-1	R/S-1	R-0
TXWRE	STPMD	UTXBE	UTXBF	RIDLE	XON	URXBE	URXBF
bit 7				bit 0			

<b>图注:</b>	HS = 硬件置1位	S = 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 15 **未实现:** 读为0

bit 14-12 **UTXISEL[2:0]:** UART发送中断选择位  
 111 = 在缓冲区中剩余1个空字节时设置发送中断  
 ...  
 010 = 在缓冲区中有6个或更多空字节时设置发送中断  
 001 = 在缓冲区中有7个或更多空字节时设置发送中断  
 000 = 在缓冲区中有8个空字节时设置发送中断; TX缓冲区为空

bit 11 **未实现:** 读为0

bit 10-8 **URXISEL[2:0]:** UART接收中断选择位<sup>(1)</sup>  
 111 = 在缓冲区中有8个字节时触发接收中断; RX缓冲区已满  
 ...  
 001 = 在缓冲区中有2个或更多字节时触发接收中断  
 000 = 在缓冲区中有1个或更多字节时触发接收中断

bit 7 **TXWRE:** TX写发送错误状态位  
**LIN和奇偶校验模式:**  
 1 = 当缓冲区已满或P2[8:0] = 0时写入新字节 (必须由软件清零)  
 0 = 无错误  
**地址检测模式:**  
 1 = 当缓冲区已满时写入新字节, 或者当P1x已满时写入P1[8:0] (必须由软件清零)  
 0 = 无错误  
**其他模式:**  
 1 = 当缓冲区已满时写入新字节 (必须由软件清零)  
 0 = 无错误

bit 6 **STPMD:** 停止位检测模式位  
 1 = 在最后一个停止位结束时触发RXIF  
 0 = 在第一个 (或第二个, 具体取决于STSEL[1:0]设置) 停止位的中间位置触发RXIF

bit 5 **UTXBE:** UART TX缓冲区空状态位  
 1 = 发送缓冲区为空; 在UTXEN = 0时写入1将复位TX FIFO指针和计数器  
 0 = 发送缓冲区不为空

bit 4 **UTXBF:** UART TX缓冲区满状态位  
 1 = 发送缓冲区已满  
 0 = 发送缓冲区未滿

bit 3 **RIDLE:** 接收空闲位  
 1 = UART RX线处于空闲状态  
 0 = UART RX线处于接收状态

**注 1:** 如果PERR或FERR置1且相应的IE位置1, 则不会设置接收水印中断。

## 寄存器15-4: UxSTAH: UARTx状态高位字寄存器 (续)

bit 2	<b>XON:</b> UART 处于 XON 模式位 仅当 FLO[1:0] 控制位设置为 XON/XOFF 模式时有效。 1 = UART 已接收到 XON 0 = UART 尚未接收到 XON 或已接收到 XOFF
bit 1	<b>URXBE:</b> UART RX 缓冲区空状态位 1 = 接收缓冲区为空; 在 URXEN = 0 时写入 1 将复位 RX FIFO 指针和计数器 0 = 接收缓冲区不为空
bit 0	<b>URXBF:</b> UART RX 缓冲区满状态位 1 = 接收缓冲区已满 0 = 接收缓冲区未满

**注 1:** 如果 PERR 或 FERR 置 1 且相应的 IE 位置 1, 则不会设置接收水印中断。

# dsPIC33CK64MP105系列

寄存器15-5: UxBRG: UARTx波特率寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG[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
BRG[7:0]							
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 15-0 **BRG[15:0]:** 波特率分频比位

寄存器15-6: UxBRGH: UARTx波特率高位字寄存器

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

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	BRG[19:16]			
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 15-4 **未实现:** 读为0

bit 3-0 **BRG[19:16]:** 波特率分频比位



## 寄存器15-7: UxRXREG: UARTx接收缓冲区寄存器

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

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x
RXREG[7:0]							
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 15-8 未实现: 读为0

bit 7-0 **RXREG[7:0]**: 接收字符数据 bit 7-0

## 寄存器15-8: UxTXREG: UARTx发送缓冲区寄存器

W-x	U-0	U-0	U-0	U-0	U-0	U-0	U-0
LAST	—	—	—	—	—	—	—
bit 15							bit 8

W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
TXREG[7:0]							
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 15 **LAST**: 用于支持智能卡的最后一个字节指示符位

bit 14-8 未实现: 读为0

bit 7-0 **TXREG[7:0]**: 发送字符数据 bit 7-0

如果缓冲区已满, 则对缓冲区的后续写操作将被忽略。

# dsPIC33CK64MP105系列

寄存器15-9: UxP1: UARTx时序参数1寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	P1[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
P1[7:0]							
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 15-9 未实现: 读为0

bit 8-0 **P1[8:0]:** 参数1位

DMX TX:

待发送的字节数 - 1 (不包括起始码)。

LIN主TX:

待发送的PID (bit[5:0])。

带地址检测的异步TX:

待发送的地址。自动向bit 9插入1 (bit[7:0])。

智能卡模式:

保护时间计数器位。该计数器基于周期始终等于1个ETU的位时钟工作 (bit[8:0])。

其他模式:

不使用。

## 寄存器15-10: UxP2: UARTx时序参数2寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	P2[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
P2[7:0]							
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 15-9 未实现: 读为0

bit 8-0 **P2[8:0]:** 参数2位

#### DMX RX:

待接收的第一个字节的编号 – 1, 不包括起始码 (bit[8:0])。

#### LIN从TX:

待发送的字节数 (bit[7:0])。

#### 带地址检测的异步RX:

要进行匹配的地址 (bit[7:0])。

#### 智能卡模式:

块时间计数器位。该计数器基于周期始终等于1个ETU的位时钟工作 (bit[8:0])。

#### 其他模式:

不使用。

# dsPIC33CK64MP105系列

寄存器15-11: UxP3: UARTx时序参数3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
P3[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
P3[7:0]							
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 15-0

**P3[15:0]:** 参数3位

DMX RX:

待接收的最后一个字节的编号 - 1, 不包括起始码 (bit[8:0])。

LIN从RX:

待接收的字节数 (bit[7:0])。

异步RX:

用于屏蔽UxP2地址位; 1 = 使用P2地址位, 0 = 屏蔽P2地址位 (bit[7:0])。

智能卡模式:

等待时间计数器位 (bit[15:0])。

其他模式:

不使用。

寄存器15-12: UxP3H: UARTx时序参数3高位字寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
P3[23:16]							
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 15-8

**未实现:** 读为0

bit 7-0

**P3[23:16]:** 参数3高字节位:

智能卡模式:

等待时间计数器位 (bit[23:16])。

其他模式:

不使用。

## 寄存器15-13: UxTXCHK: UARTx发送校验和寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
TXCHK[7:0]							
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 15-8 未实现: 读为0

bit 7-0 **TXCHK[7:0]:** 发送校验和位 (针对TX字计算)

#### LIN模式:

C0EN = 1: 发送的所有数据 + 加法进位之和, 包括PID。

C0EN = 0: 发送的所有数据 + 加法进位之和, 不包括PID。

#### LIN从模式:

检测到暂停字符时清零。

#### LIN主/从模式:

检测到暂停字符时清零。

#### 其他模式:

C0EN = 1: 发送的每个字节 + 加法进位之和。

C0EN = 0: 值保持不变。

# dsPIC33CK64MP105系列

寄存器15-14: UxRXCHK: UARTx接收校验和寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
RXCHK[7:0]							
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 15-8 未实现: 读为0

bit 7-0 **RXCHK[7:0]:** 接收校验和位 (针对RX字计算)

LIN模式:

C0EN = 1: 接收的所有数据 + 加法进位之和, 包括PID。

C0EN = 0: 接收的所有数据 + 加法进位之和, 不包括PID。

LIN从模式:

检测到暂停字符时清零。

LIN主/从模式:

检测到暂停字符时清零。

其他模式:

C0EN = 1: 接收的每个字节 + 加法进位之和。

C0EN = 0: 值保持不变。

## 寄存器15-15: UxSCCON: UARTx智能卡配置寄存器

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	TXRPT1	TXRPT0	CONV	T0PD	PRTCL	—
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 15-6 未实现: 读为0

bit 5-4 **TXRPT[1:0]**: 重复发送选择位

11 = 将错误字节重复发送4次

10 = 将错误字节重复发送3次

01 = 将错误字节重复发送2次

00 = 将错误字节重复发送1次

bit 3 **CONV**: 逻辑约定选择位

1 = 反向逻辑约定

0 = 正向逻辑约定

bit 2 **T0PD**: T = 0 错误处理时的下拉持续时间位

1 = 2个ETU

0 = 1个ETU

bit 1 **PRTCL**: 智能卡协议选择位

1 = T = 1

0 = T = 0

bit 0 未实现: 读为0

# dsPIC33CK64MP105系列

## 寄存器15-16: UxSCINT: UARTx智能卡中断寄存器

U-0	U-0	HS/R/W-0	HS/R/W-0	U-0	HS/R/W-0	HS/R/W-0	HS/R/W-0
—	—	RXRPTIF	TXRPTIF	—	BTCIF	WTCIF	GTCIF
bit 15				bit 8			

U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	—	RXRPTIE	TXRPTIE	—	BTCIE	WTCIE	GTCIE
bit 7				bit 0			

<b>图注:</b>	HS = 硬件置1位
R = 可读位	W = 可写位
-n = POR时的值	1 = 置1
	U = 未实现位, 读为0
	0 = 清零
	x = 未知

- bit 15-14 **未实现:** 读为0
- bit 13 **RXRPTIF:** 重复接收中断标志位  
1 = 在接收到同一字符5次（重复发送4次）后，奇偶校验错误仍存在  
0 = 标志清零
- bit 12 **TXRPTIF:** 重复发送中断标志位  
1 = 在通过TXRPT[1:0]配置的最后一次重新发送后检测到线路错误  
0 = 标志清零
- bit 11 **未实现:** 读为0
- bit 10 **BTCIF:** 块时间计数器中断标志位  
1 = 块时间计数器达到0  
0 = 块时间计数器未达到0
- bit 9 **WTCIF:** 等待时间计数器中断标志位  
1 = 等待时间计数器达到0  
0 = 等待时间计数器未达到0
- bit 8 **GTCIF:** 保护时间计数器中断标志位  
1 = 保护时间计数器达到0  
0 = 保护时间计数器未达到0
- bit 7-6 **未实现:** 读为0
- bit 5 **RXRPTIE:** 重复接收中断允许位  
1 = 在接收到同一字符5次（重复发送4次）后，如果奇偶校验错误仍存在，则调用中断  
0 = 禁止中断
- bit 4 **TXRPTIE:** 重复发送中断允许位  
1 = 如果在通过TXRPT[1:0]配置的最后一次重新发送完成后检测到线路错误，则调用中断  
0 = 禁止中断
- bit 3 **未实现:** 读为0
- bit 2 **BTCIE:** 块时间计数器中断允许位  
1 = 允许块时间计数器中断  
0 = 禁止块时间计数器中断
- bit 1 **WTCIE:** 等待时间计数器中断允许位  
1 = 允许等待时间计数器中断  
0 = 禁止等待时间计数器中断
- bit 0 **GTCIE:** 保护时间计数器中断允许位  
1 = 允许保护时间计数器中断  
0 = 禁止保护时间计数器中断



## 寄存器15-17: UxINT: UARTx中断寄存器

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

R/W-0, HS	R/W-0, HS	U-0	U-0	U-0	R/W-0	U-0	U-0
WUIF	ABDIF	—	—	—	ABDIE	—	—
bit 7							bit 0

<b>图注:</b>	HS = 硬件置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 15-8      **未实现:** 读为0
- bit 7          **WUIF:** 唤醒中断标志位  
1 = 当WAKE = 1且RX从1跳变为0时置1; 触发事件中断 (必须由软件清零)  
0 = WAKE未使能; 或已使能, 但未发生唤醒事件
- bit 6          **ABDIF:** 自动波特率完成中断标志位  
1 = 在ABD序列实现最终的1到0跳变时置1; 触发事件中断 (必须由软件清零)  
0 = ABAUD未使能; 或已使能, 但未完成自动波特率
- bit 5-3      **未实现:** 读为0
- bit 2          **ABDIE:** 自动波特率完成中断允许位  
1 = 允许ABDIF设置事件中断  
0 = ABDIF不设置事件中断
- bit 1-0      **未实现:** 读为0

注:

## 16.0 串行外设接口（SPI）

**注 1：** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“支持音频编解码器的串行外设接口（SPI）”（[www.microchip.com/DS70005136](http://www.microchip.com/DS70005136)）。

串行外设接口（Serial Peripheral Interface, SPI）模块是用于同其他外设或单片机器件进行通信的同步串行接口。这些外设可以是串行EEPROM、移位寄存器、显示驱动器和A/D转换器等。SPI模块与Motorola®的SPI和SIOP接口兼容。dsPIC33CK64MP105系列的所有器件均包含三个SPI模块。在48引脚器件中，将SPI实例SPI2选作非PPS引脚时，工作速度最高可达50 MHz。使用SPI2PIN位（FDEVOPT[13]）进行选择。如果SPI2PIN位为1，则将使用PPS引脚。当SPI2PIN为0时，SPI信号将连接到专用引脚。

该模块可在两种缓冲区模式下工作。在标准缓冲区模式下，通过单个串行缓冲区移动数据。在增强型缓冲区模式下，通过FIFO缓冲区移动数据。FIFO深度取决于配置的模式。

**注：** 该器件的FIFO深度为4（在8位数据模式下）。

可以发送和接收的可变数据长度介于2至32位之间。

**注：** 无论是在标准还是增强型缓冲区模式下，都不要对SPIxBUF寄存器执行读-修改-写操作（例如针对位的指令）。

无论是在主模式还是从模式下工作，该模块都支持基本帧SPI协议。该模块总共支持四种帧SPI配置。

该模块还支持音频模式。下面列出了四种可供选择的音频模式。

- I<sup>2</sup>S 模式
- 左对齐模式
- 右对齐模式
- PCM/DSP 模式

在每种模式下，串行时钟都自由运行并且始终传输音频数据。

如果在两个器件之间传输音频协议数据，则通常其中一个器件为主器件，而另一个器件为从器件。不过，两个从器件之间也可传输音频数据。由于音频协议需要自由运行的时钟，因此主器件可以是第三方控制器。在任何一种情况下，主器件都会生成两个自由运行的时钟：SCKx和LRC（左、右通道时钟/SSx/FSYNC）。

SPI串行接口由4个引脚组成：

- SDIx：串行数据输入
- SDOx：串行数据输出
- SCKx：移位时钟输入或输出
- SSx：低电平有效从选择或帧同步I/O脉冲

可将SPI模块配置为使用2、3或4个引脚进行工作。在3引脚模式下，不使用SSx。在2引脚模式下，不使用SDOx和SSx。

# dsPIC33CK64MP105系列

SPI 模块可产生三种中断，以反映数据通信期间发生的事件。下面列出了这三种中断：

1. 接收中断，由 SPIxRXIF 指示。在以下情况下会发生该事件：

- RX 水印中断
- SPIROV = 1
- SPIRBF = 1
- SPIRBE = 1

前提是通过 SPIxIMSKL/H 使能了相应的屏蔽位。

2. 发送中断，由 SPIxTXIF 指示。在以下情况下会发生该事件：

- TX 水印中断
- SPITUR = 1
- SPITBF = 1
- SPITBE = 1

前提是通过 SPIxIMSKL/H 使能了相应的屏蔽位。

3. 一般中断，由 SPIxGIF 指示。在以下情况下会发生该事件：

- FRMERR = 1
- SPIBUSY = 1
- SRMT = 1

前提是通过 SPIxIMSKL/H 使能了相应的屏蔽位。

图 16-1 和图 16-2 分别给出了标准和增强型缓冲区模式下的模块框图。

**注：** 在本节中，SPI 模块统称为 SPIx，或分别称为 SPI1、SPI2 或 SPI3。特殊功能寄存器也使用类似的表示法。例如，SPIxCON1 和 SPIxCON2 表示三个 SPI 模块中任一模块的控制寄存器。

要将 SPIx 模块设置为标准主工作模式：

1. 如果使用中断：
  - a) 将相应 IFSx 寄存器中的中断标志位清零。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 写入相应 IPCx 寄存器中的 SPIxIP 位，设置中断优先级。
2. 将所需设置写入 SPIxCON1L 和 SPIxCON1H 寄存器，同时使 MSTEN 位 (SPIxCON1L[5]) = 1。
3. 清零 SPIROV 位 (SPIxSTATL[6])。
4. 通过将 SPIEN 位 (SPIxCON1L[15]) 置 1 使能 SPIx 操作。
5. 将待发送的数据写入 SPIxBUFL 和 SPIxBUFH 寄存器。一旦数据写入 SPIxBUFL 和 SPIxBUFH 寄存器，便会立即开始发送（和接收）。

要将 SPIx 模块设置为标准从工作模式：

1. 清零 SPIxBUF 寄存器。
2. 如果使用中断：
  - a) 清零 SPIxBUFL 和 SPIxBUFH 寄存器。
  - b) 将相应 IECx 寄存器中的中断允许位置 1。
  - c) 写入相应 IPCx 寄存器中的 SPIxIP 位，设置中断优先级。
3. 将所需设置写入 SPIxCON1L、SPIxCON1H 和 SPIxCON2L 寄存器，同时使 MSTEN 位 (SPIxCON1L[5]) = 0。
4. 清零 SMP 位。
5. 如果 CKE 位 (SPIxCON1L[8]) 置 1，则必须将 SSx 位 (SPIxCON1L[7]) 置 1 以使能 SSx 引脚。
6. 清零 SPIROV 位 (SPIxSTATL[6])。
7. 通过将 SPIEN 位 (SPIxCON1L[15]) 置 1 使能 SPIx 操作。



# dsPIC33CK64MP105系列

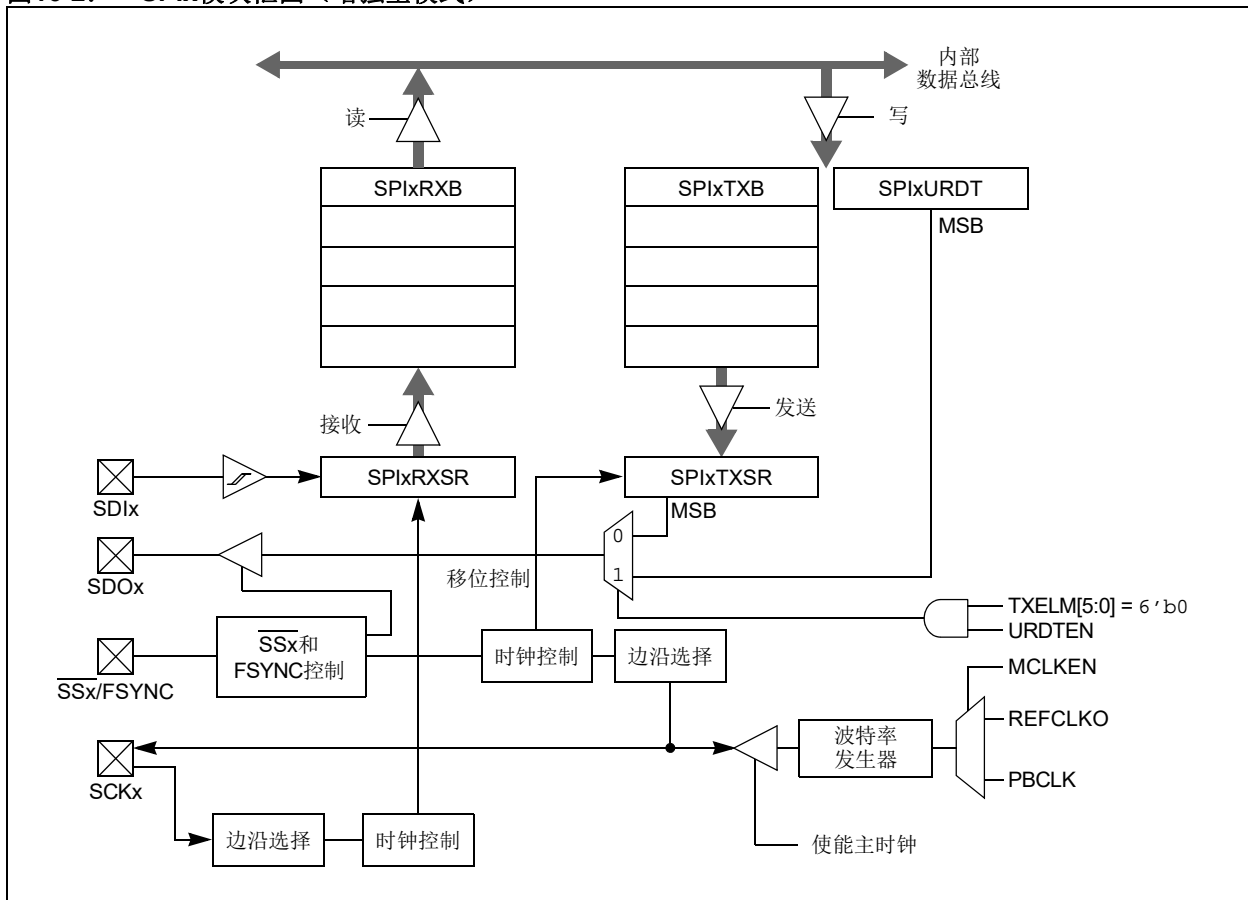
要将SPIx模块设置为增强型缓冲区主工作模式：

1. 如果使用中断：
  - a) 将相应IFSx寄存器中的中断标志位清零。
  - b) 将相应IECx寄存器中的中断允许位置1。
  - c) 写入相应IPCx寄存器中的SPIxIP位。
2. 将所需设置写入SPIxCON1L、SPIxCON1H和SPIxCON2L寄存器，同时使MSTEN (SPIxCON1L[5]) = 1。
3. 清零SPIOV位 (SPIxSTATL[6])。
4. 通过将ENHBUF位 (SPIxCON1L[0]) 置1选择增强型缓冲区模式。
5. 通过将SPIEN位 (SPIxCON1L[15]) 置1使能SPIx操作。
6. 将待发送的数据写入SPIxBUFL和SPIxBUFH寄存器。一旦数据写入SPIxBUFL和SPIxBUFH寄存器，便会立即开始发送（和接收）。

要将SPIx模块设置为增强型缓冲区从工作模式：

1. 清零SPIxBUFL和SPIxBUFH寄存器。
2. 如果使用中断：
  - a) 将相应IFSx寄存器中的中断标志位清零。
  - b) 将相应IECx寄存器中的中断允许位置1。
  - c) 写入相应IPCx寄存器中的SPIxIP位，设置中断优先级。
3. 将所需设置写入SPIxCON1L、SPIxCON1H和SPIxCON2L寄存器，同时使MSTEN位 (SPIxCON1L[5]) = 0。
4. 清零SMP位。
5. 如果CKE位置1，则必须将SSEN位置1以使能SSx引脚。
6. 清零SPIOV位 (SPIxSTATL[6])。
7. 通过将ENHBUF位 (SPIxCON1L[0]) 置1选择增强型缓冲区模式。
8. 通过将SPIEN位 (SPIxCON1L[15]) 置1使能SPIx操作。

图16-2: SPIx模块框图（增强型模式）



要将SPIx模块设置为音频模式：

1. 清零SPIxBUFL和SPIxBUFH寄存器。
2. 如果使用中断：
  - a) 将相应IFSx寄存器中的中断标志位清零。
  - b) 将相应IECx寄存器中的中断允许位置1。
  - c) 写入相应IPCx寄存器中的SPIxIP位，设置中断优先级。
3. 将所需设置写入SPIxCON1L、SPIxCON1H和SPIxCON2L寄存器，同时使AUDEN (SPIxCON1H[15]) = 1。
4. 清零SPIOV位 (SPIxSTATL[6])。
5. 通过将SPIEN位 (SPIxCON1L[15]) 置1使能SPIx操作。
6. 将待发送的数据写入SPIxBUFL和SPIxBUFH寄存器。一旦数据写入SPIxBUFL和SPIxBUFH寄存器，便会立即开始发送（和接收）。

# dsPIC33CK64MP105系列

## 16.1 SPI控制/状态寄存器

寄存器16-1: **SPIxCON1L: SPIx控制低位字寄存器1**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIEN	—	SPISIDL	DISSDO	MODE32 <sup>(1,4)</sup>	MODE16 <sup>(1,4)</sup>	SMP	CKE <sup>(1)</sup>
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
SSEN <sup>(2)</sup>	CKP	MSTEN	DISSDI	DISSCK	MCLKEN <sup>(3)</sup>	SPIFE	ENHBUF
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 15      **SPIEN:** SPIx使能位  
 1 = 使能模块  
 0 = 关闭并复位模块、禁止时钟、禁止中断事件产生并允许修改SFR
- bit 14      **未实现:** 读为0
- bit 13      **SPISIDL:** SPIx空闲模式停止位  
 1 = 在CPU空闲模式下停止工作  
 0 = 在CPU空闲模式下继续工作
- bit 12      **DISSDO:** 禁止SDOx输出端口位  
 1 = 模块不使用SDOx引脚; 引脚由端口功能控制  
 0 = SDOx引脚由模块控制
- bit 11-10   **MODE32和MODE16:** 串行字长度选择位<sup>(1,4)</sup>

MODE32	MODE16	AUDEN	通信
1	x	0	32位
0	1		16位
0	0		8位
1	1	1	24位数据, 32位FIFO, 32位通道/64位帧
1	0		32位数据, 32位FIFO, 32位通道/64位帧
0	1		16位数据, 16位FIFO, 32位通道/64位帧
0	0		16位FIFO, 16位通道/32位帧

- bit 9      **SMP:** SPIx数据输入采样阶段位  
主模式:  
 1 = 在数据输出时间的末端采样输入数据  
 0 = 在数据输出时间的中间采样输入数据  
从模式:  
 无论SMP设置如何, 始终在数据输出时间的中间采样输入数据
- bit 8      **CKE:** SPIx时钟边沿选择位<sup>(1)</sup>  
 1 = 时钟状态从有效转换到空闲时发送  
 0 = 时钟状态从空闲转换到有效时发送

- 注 1: 当AUDEN (SPIxCON1H[15]) = 1时, 该模块按照CKE = 0的情况运行, 而不管该位的实际值为何。
- 2: 当FRMEN = 1时, 不使用SSEN。
- 3: 仅当SPIEN位 = 0时才能写MCLKEN位。
- 4: 该通道对于DSP/PCM模式而言没有意义, 因为LRC遵循FRMSYPW。



## 寄存器16-1: SPIxCON1L: SPIx控制低位字寄存器1 (续)

bit 7	<b>SSEN:</b> 从选择使能位 (从模式) <sup>(2)</sup> 1 = 宏在从模式下使用 $\overline{SSx}$ 引脚; $\overline{SSx}$ 引脚用作从选择输入 0 = 宏不使用 $\overline{SSx}$ 引脚 ( $\overline{SSx}$ 引脚将由端口 I/O 控制)
bit 6	<b>CKP:</b> 时钟极性选择位 1 = 时钟信号空闲状态为高电平; 有效状态为低电平 0 = 时钟信号空闲状态为低电平; 有效状态为高电平
bit 5	<b>MSTEN:</b> 主模式使能位 1 = 主模式 0 = 从模式
bit 4	<b>DISSDI:</b> 禁止 SDIx 输入端口位 1 = 模块不使用 SDIx 引脚; 引脚由端口功能控制 0 = SDIx 引脚由模块控制
bit 3	<b>DISSCK:</b> 禁止 SCKx 输出端口位 1 = 模块不使用 SCKx 引脚; 引脚由端口功能控制 0 = SCKx 引脚由模块控制
bit 2	<b>MCLKEN:</b> 主时钟使能位 <sup>(3)</sup> 1 = BRG 使用参考时钟 (REFCLKO) 0 = BRG 使用外设时钟 (FP = Fosc/2)
bit 1	<b>SPIFE:</b> 帧同步脉冲边沿选择位 1 = 帧同步脉冲 (空闲到有效边沿) 与第一个位时钟一致 0 = 帧同步脉冲 (空闲到有效边沿) 超前于第一个位时钟
bit 0	<b>ENHBUF:</b> 增强型缓冲区使能位 1 = 使能增强型缓冲区模式 0 = 禁止增强型缓冲区模式

注 1: 当 AUDEN (SPIxCON1H[15]) = 1 时, 该模块按照 CKE = 0 的情况运行, 而不管该位的实际值为何。

2: 当 FRMEN = 1 时, 不使用 SSEN。

3: 仅当 SPIEN 位 = 0 时才能写 MCLKEN 位。

4: 该通道对于 DSP/PCM 模式而言没有意义, 因为 LRC 遵循 FRMSYPW。

# dsPIC33CK64MP105系列

**寄存器16-2: SPIxCON1H: SPIx控制高位字寄存器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
AUDEN <sup>(1)</sup>	SPIGNEXT	IGNROV	IGNTUR	AUDMONO <sup>(2)</sup>	URDTEN <sup>(3)</sup>	AUDMOD1 <sup>(4)</sup>	AUDMOD0 <sup>(4)</sup>
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
FRMEN	FRMSYNC	FRMPOL	MSEN	FRMSYPW	FRMCNT2	FRMCNT1	FRMCNT0
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 15      **AUDEN:** 音频编解码器支持使能位<sup>(1)</sup>  
 1 = 使能音频协议; MSTEN控制SCKx和帧(又称为LRC)的方向并且该模块按照FRMEN = 1、FRMSYNC = MSTEN、FRMCNT[2:0] = 001且SMP = 0的情况运行, 而不管它们的实际值为何  
 0 = 禁止音频协议
- bit 14      **SPIGNEXT:** SPIx符号扩展RX FIFO读数据使能位  
 1 = 对RX FIFO中的数据进行符号扩展  
 0 = 不对RX FIFO中的数据进行符号扩展
- bit 13      **IGNROV:** 忽略接收溢出位  
 1 = 接收溢出(Receive Overflow, ROV)不是关键错误; 在ROV期间, FIFO中的数据不会被接收数据覆盖  
 0 = ROV是关键错误, 会停止SPI操作
- bit 12      **IGNTUR:** 忽略发送数据不足位  
 1 = 发送数据不足(Transmit Underrun, TUR)不是关键错误, 在SPIxTXB不为空之前将一直发送URDTEN指示的数据  
 0 = TUR是关键错误, 会停止SPI操作
- bit 11      **AUDMONO:** 发送音频数据格式位<sup>(2)</sup>  
 1 = 音频数据为单声道(即, 每个数据字在左右通道同时发送)  
 0 = 音频数据为立体声
- bit 10      **URDTEN:** 发送数据不足使能位<sup>(3)</sup>  
 1 = 出现发送数据不足的情况时发送SPIxURDT寄存器中的数据  
 0 = 出现发送数据不足的情况时发送上一次接收到的数据
- bit 9-8     **AUDMOD[1:0]:** 音频协议模式选择位<sup>(4)</sup>  
 11 = PCM/DSP模式  
 10 = 右对齐模式: 该模块按照SPIFE = 1的情况运行, 而不管该位的实际值为何  
 01 = 左对齐模式: 该模块按照SPIFE = 1的情况运行, 而不管该位的实际值为何  
 00 = I<sup>2</sup>S模式: 该模块按照SPIFE = 0的情况运行, 而不管该位的实际值为何
- bit 7      **FRMEN:** 帧SPIx支持位  
 1 = 使能帧SPIx支持(SSx引脚用作FSYNC输入/输出)  
 0 = 禁止帧SPIx支持

- 注 1:** 仅当SPIEN位 = 0时才能写AUDEN位。  
**2:** 仅当SPIEN位 = 0时才能写AUDMONO, 并且AUDMONO仅在AUDEN = 1时有效。  
**3:** URDTEN仅在IGNTUR = 1时有效。  
**4:** 仅当SPIEN位 = 0时才能写AUDMOD[1:0], 并且AUDMOD[1:0]仅在AUDEN = 1时有效。当未处于PCM/DSP模式时, 该模块按照FRMSYPW = 1的情况运行, 而不管该位的实际值为何。

## 寄存器16-2: SPIxCON1H: SPIx控制高位字寄存器1 (续)

bit 6	<b>FRMSYNC:</b> 帧同步脉冲方向控制位 1 = 帧同步脉冲输入 (从器件) 0 = 帧同步脉冲输出 (主器件)
bit 5	<b>FRMPOL:</b> 帧同步/从选择极性位 1 = 帧同步脉冲/从选择为高电平有效 0 = 帧同步脉冲/从选择为低电平有效
bit 4	<b>MSEN:</b> 主模式从选择使能位 1 = 使能 SPIx 从选择支持, 极性由 FRMPOL 决定 (在主模式下, 发送期间会自动驱动 $\overline{SSx}$ 引脚) 0 = 禁止 SPIx 从选择支持 ( $\overline{SSx}$ 引脚将由端口 I/O 控制)
bit 3	<b>FRMSYPW:</b> 帧同步脉冲宽度位 1 = 帧同步脉冲宽度为 1 个串行字长 (如 MODE[32,16]/WLENGTH[4:0] 所定义) 0 = 帧同步脉冲宽度为一个时钟 (SCKx)
bit 2-0	<b>FRMCNT[2:0]:</b> 帧同步脉冲计数器位 控制每个同步脉冲发送的串行字数。 111 = 保留 110 = 保留 101 = 每 32 个串行字产生一个帧同步脉冲 100 = 每 16 个串行字产生一个帧同步脉冲 011 = 每 8 个串行字产生一个帧同步脉冲 010 = 每 4 个串行字产生一个帧同步脉冲 001 = 每 2 个串行字产生一个帧同步脉冲 (音频协议使用的值) 000 = 每个串行字产生一个帧同步脉冲

- 注 1: 仅当 SPIEN 位 = 0 时才能写 AUDEN 位。
- 2: 仅当 SPIEN 位 = 0 时才能写 AUDMONO, 并且 AUDMONO 仅在 AUDEN = 1 时有效。
- 3: URDTEN 仅在 IGNTUR = 1 时有效。
- 4: 仅当 SPIEN 位 = 0 时才能写 AUDMOD[1:0], 并且 AUDMOD[1:0] 仅在 AUDEN = 1 时有效。当未处于 PCM/DSP 模式时, 该模块按照 FRMSYPW = 1 的情况运行, 而不管该位的实际值为何。

# dsPIC33CK64MP105系列

寄存器16-3: SPIxCON2L: SPIx控制低位字寄存器2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	WLENGTH[4:0] <sup>(1,2)</sup>				
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 15-5 未实现: 读为0

bit 4-0 **WLENGTH[4:0]:** 可变字长度位<sup>(1,2)</sup>

11111 = 32位数据  
11110 = 31位数据  
11101 = 30位数据  
11100 = 29位数据  
11011 = 28位数据  
11010 = 27位数据  
11001 = 26位数据  
11000 = 25位数据  
10111 = 24位数据  
10110 = 23位数据  
10101 = 22位数据  
10100 = 21位数据  
10011 = 20位数据  
10010 = 19位数据  
10001 = 18位数据  
10000 = 17位数据  
01111 = 16位数据  
01110 = 15位数据  
01101 = 14位数据  
01100 = 13位数据  
01011 = 12位数据  
01010 = 11位数据  
01001 = 10位数据  
01000 = 9位数据  
00111 = 8位数据  
00110 = 7位数据  
00101 = 6位数据  
00100 = 5位数据  
00011 = 4位数据  
00010 = 3位数据  
00001 = 2位数据  
00000 = 请参见SPIxCON1L[11:10]中的MODE[32,16]位

注 1: 这些位仅在AUDEN = 0时有效。

2: 通过更改这些位来改变长度不会影响TX/RX FIFO的深度。

寄存器 16-4: SPIxSTATL: SPIx状态低位字寄存器

U-0	U-0	U-0	HS/R/C-0	HSC/R-0	U-0	U-0	HSC/R-0
—	—	—	FRMERR	SPIBUSY	—	—	SPITUR <sup>(1)</sup>
bit 15							bit 8

HSC/R-0	HS/R/C-0	HSC/R-1	U-0	HSC/R-1	U-0	HSC/R-0	HSC/R-0
SRMT	SPIROV	SPIRBE	—	SPITBE	—	SPITBF	SPIRBF
bit 7							bit 0

图注:	C = 可清零位	U = 未实现, 读为0
R = 可读位	W = 可写位	HSC = 硬件置1/清零位
-n = POR时的值	1 = 置1	0 = 清零
		HS = 硬件置1位

bit 15-13 未实现: 读为0

bit 12 **FRMERR**: SPIx帧错误状态位

1 = 检测到帧错误  
0 = 未检测到帧错误

bit 11 **SPIBUSY**: SPIx活动状态位

1 = 模块当前正忙于处理一些事务  
0 = 当前未处理任何事务 (读取时)

bit 10-9 未实现: 读为0

bit 8 **SPITUR**: SPIx发送数据不足状态位<sup>(1)</sup>

1 = 发送缓冲区遇到发送数据不足条件  
0 = 发送缓冲区未遇到发送数据不足条件

bit 7 **SRMT**: 移位寄存器空状态位

1 = 当前未处理任何事务且无事务等待处理 (即, SPIxTXB和SPIxTXSR均不包含要发送的数据)  
0 = 当前正在处理事务或存在等待处理的事务

bit 6 **SPIROV**: SPIx接收溢出状态位

1 = SPIxRXB已满时又完整接收了一个新字节/半字/字  
0 = 未发生溢出

bit 5 **SPIRBE**: SPIx RX缓冲区空状态位

1 = RX缓冲区为空  
0 = RX缓冲区不为空

标准缓冲区模式:

当读SPIxBUF来读SPIxRXB时, 该位由硬件自动置1。当SPIx将数据从SPIxRXSR传输到SPIxRXB时, 该位由硬件自动清零。

增强型缓冲区模式:

指示RXELM[5:0] = 000000。

bit 4 未实现: 读为0

注 1: SPIEN = 0时, SPITUR清零。当IGNTUR = 1时, SPITUR提供发送数据不足条件的动态状态, 但不会停止RX/TX操作, 并且也不需要软件清零。

# dsPIC33CK64MP105系列

---

## 寄存器 16-4: SPIxSTATL: SPIx状态低位寄存器 (续)

bit 3	<p><b>SPITBE:</b> SPIx发送缓冲区空状态位</p> <p>1 = SPIxTXB 为空 0 = SPIxTXB 非空</p> <p><u>标准缓冲区模式:</u> 当SPIx将数据从SPIxTXB传输到SPIxTXSR时, 该位由硬件自动置1。当写SPIxBUF来装载SPIxTXB时, 该位由硬件自动清零。</p> <p><u>增强型缓冲区模式:</u> 指示TXELM[5:0] = 000000。</p>
bit 2	<p><b>未实现:</b> 读为0</p>
bit 1	<p><b>SPITBF:</b> SPIx发送缓冲区满状态位</p> <p>1 = SPIxTXB 已满 0 = SPIxTXB 未满</p> <p><u>标准缓冲区模式:</u> 当写SPIxBUF来装载SPIxTXB时, 该位由硬件自动置1。当SPIx将数据从SPIxTXB传输到SPIxTXSR时, 该位由硬件自动清零。</p> <p><u>增强型缓冲区模式:</u> 指示TXELM[5:0] = 111111。</p>
bit 0	<p><b>SPIRBF:</b> SPIx接收缓冲区满状态位</p> <p>1 = SPIxRXB 已满 0 = SPIxRXB 未满</p> <p><u>标准缓冲区模式:</u> 当SPIx将数据从SPIxRXSR传输到SPIxRXB时, 该位由硬件自动置1。当读SPIxBUF来读SPIxRXB时, 该位由硬件自动清零。</p> <p><u>增强型缓冲区模式:</u> 指示RXELM[5:0] = 111111。</p>

**注 1:** SPIEN = 0时, SPITUR清零。当IGNTUR = 1时, SPITUR提供发送数据不足条件的动态状态, 但不会停止RX/TX操作, 并且也不需要通过软件清零。

## 寄存器16-5: SPIxSTATH: SPIx状态高位字寄存器

U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	RXELM5 <sup>(3)</sup>	RXELM4 <sup>(2)</sup>	RXELM3 <sup>(1)</sup>	RXELM2	RXELM1	RXELM0
bit 15							bit 8

U-0	U-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
—	—	TXELM5 <sup>(3)</sup>	TXELM4 <sup>(2)</sup>	TXELM3 <sup>(1)</sup>	TXELM2	TXELM1	TXELM0
bit 7							bit 0

<b>图注:</b>	HSC = 硬件置1/清零位						
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 15-14 未实现: 读为0

bit 13-8 **RXELM[5:0]:** 接收缓冲区元素计数位 (在增强型缓冲区模式下有效) <sup>(1,2,3)</sup>

bit 7-6 未实现: 读为0

bit 5-0 **TXELM[5:0]:** 发送缓冲区元素计数位 (在增强型缓冲区模式下有效) <sup>(1,2,3)</sup>

**注 1:** 只有FIFODEPTH = 8或更高值时, 才存在RXELM3和TXELM3位。

**2:** 只有FIFODEPTH = 16或更高值时, 才存在RXELM4和TXELM4位。

**3:** 只有FIFODEPTH = 32时, 才存在RXELM5和TXELM5位。

# dsPIC33CK64MP105系列

寄存器16-6: SPIxIMSKL: SPIx中断屏蔽低位字寄存器

U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	—	FRMERREN	BUSYEN	—	—	SPITUREN
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0
SRMTEN	SPIROVEN	SPIRBEN	—	SPITBEN	—	SPITBFEN	SPIRBFEN
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 15-13      **未实现:** 读为0
- bit 12      **FRMERREN:** 允许通过FRMERR产生中断事件位  
1 = 帧错误产生中断事件  
0 = 帧错误不产生中断事件
- bit 11      **BUSYEN:** 允许通过SPIBUSY产生中断事件位  
1 = SPIBUSY产生中断事件  
0 = SPIBUSY不产生中断事件
- bit 10-9      **未实现:** 读为0
- bit 8      **SPITUREN:** 允许通过SPITUR产生中断事件位  
1 = 发送数据不足 (TUR) 产生中断事件  
0 = 发送数据不足不产生中断事件
- bit 7      **SRMTEN:** 允许通过SRMT产生中断事件位  
1 = 移位寄存器为空 (Shift Register Empty, SRMT) 产生中断事件  
0 = 移位寄存器为空不产生中断事件
- bit 6      **SPIROVEN:** 允许通过SPIOV产生中断事件位  
1 = SPIx接收溢出 (ROV) 产生中断事件  
0 = SPIx接收溢出不产生中断事件
- bit 5      **SPIRBEN:** 允许通过SPIRBE产生中断事件位  
1 = SPIx RX缓冲区为空产生中断事件  
0 = SPIx RX缓冲区为空不产生中断事件
- bit 4      **未实现:** 读为0
- bit 3      **SPITBEN:** 允许通过SPITBE产生中断事件位  
1 = SPIx发送缓冲区为空产生中断事件  
0 = SPIx发送缓冲区为空不产生中断事件
- bit 2      **未实现:** 读为0
- bit 1      **SPITBFEN:** 允许通过SPITBF产生中断事件位  
1 = SPIx发送缓冲区已满产生中断事件  
0 = SPIx发送缓冲区已满不产生中断事件
- bit 0      **SPIRBFEN:** 允许通过SPIRBF产生中断事件位  
1 = SPIx接收缓冲区已满产生中断事件  
0 = SPIx接收缓冲区已满不产生中断事件



## 寄存器16-7: SPIxIMSKH: SPIx中断屏蔽高位字寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
RXWIEN	—	RXMSK5 <sup>(1)</sup>	RXMSK4 <sup>(1,4)</sup>	RXMSK3 <sup>(1,3)</sup>	RXMSK2 <sup>(1,2)</sup>	RXMSK1 <sup>(1)</sup>	RXMSK0 <sup>(1)</sup>
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TXWIEN	—	TXMSK5 <sup>(1)</sup>	TXMSK4 <sup>(1,4)</sup>	TXMSK3 <sup>(1,3)</sup>	TXMSK2 <sup>(1,2)</sup>	TXMSK1 <sup>(1)</sup>	TXMSK0 <sup>(1)</sup>
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

- bit 15 **RXWIEN**: 接收水印中断允许位  
1 = 当RXMSK[5:0] ≤ RXELM[5:0]时, 触发接收缓冲区元素水印中断  
0 = 禁止接收缓冲区元素水印中断
- bit 14 **未实现**: 读为0
- bit 13-8 **RXMSK[5:0]**: RX缓冲区屏蔽位<sup>(1,2,3,4)</sup>  
RX屏蔽位, 与RXWIEN位配合使用。
- bit 7 **TXWIEN**: 发送水印中断允许位  
1 = 当TXMSK[5:0] = TXELM[5:0]时, 触发发送缓冲区元素水印中断  
0 = 禁止发送缓冲区元素水印中断
- bit 6 **未实现**: 读为0
- bit 5-0 **TXMSK[5:0]**: TX缓冲区屏蔽位<sup>(1,2,3,4)</sup>  
TX屏蔽位, 与TXWIEN位配合使用。

注 1: 高于FIFODEPTH的屏蔽值无效。在这种情况下, 模块不会为任何值触发匹配。

2: 只有FIFODEPTH = 8或更高值时, 才存在RXMSK2和TXMSK2位。

3: 只有FIFODEPTH = 16或更高值时, 才存在RXMSK3和TXMSK3位。

4: 只有FIFODEPTH = 32时, 才存在RXMSK4和TXMSK4位。

图16-3: SPIx主/从连接（标准模式）

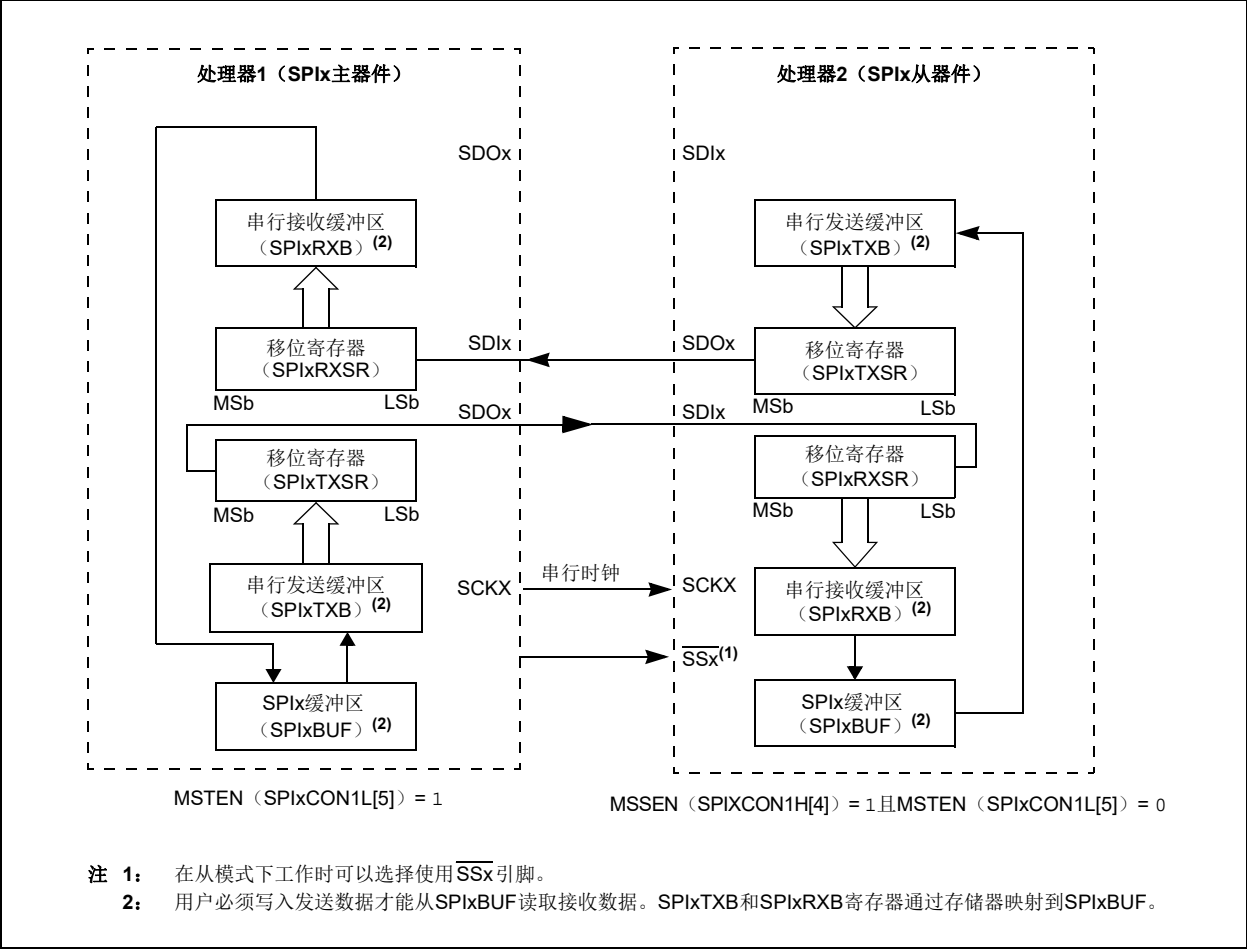
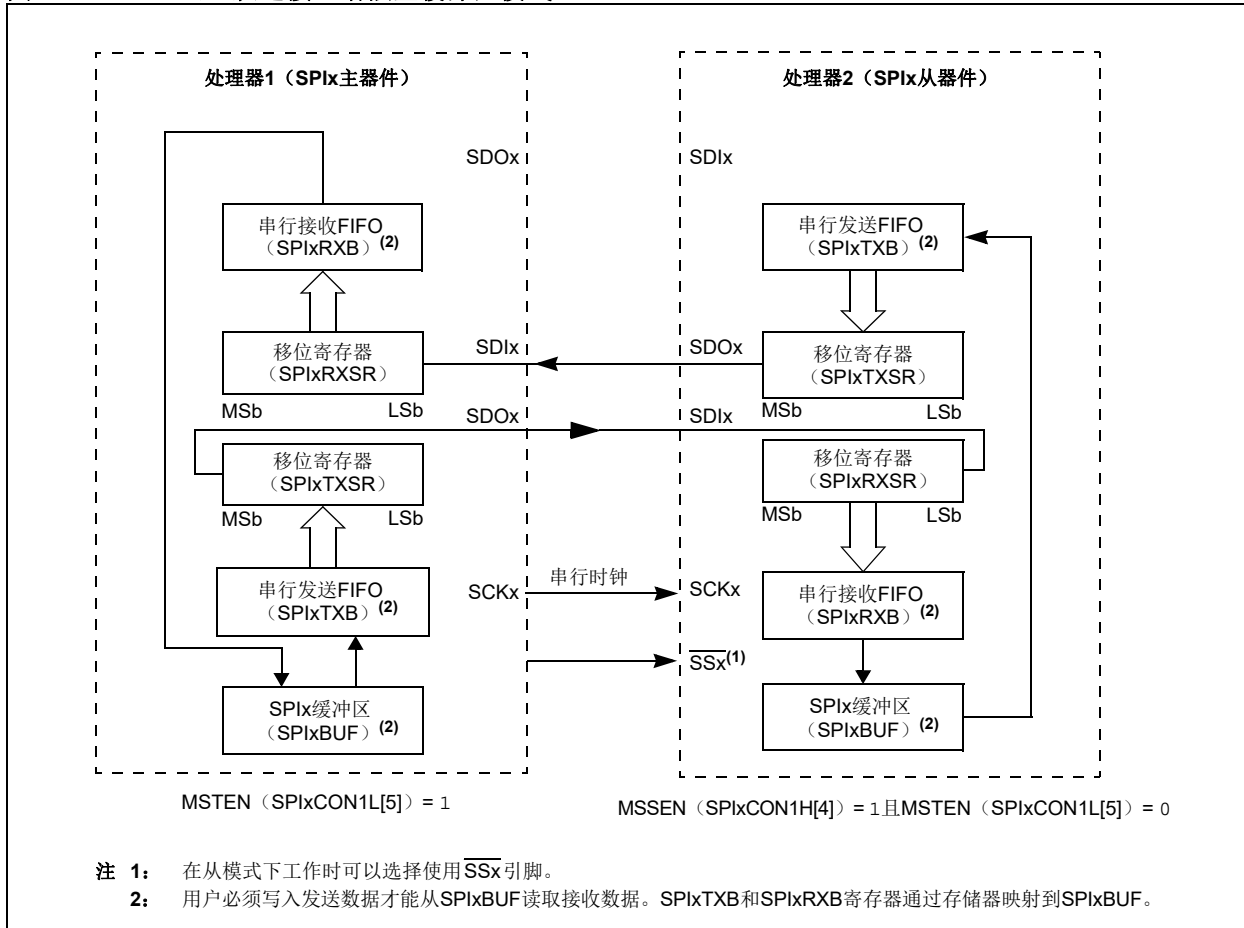
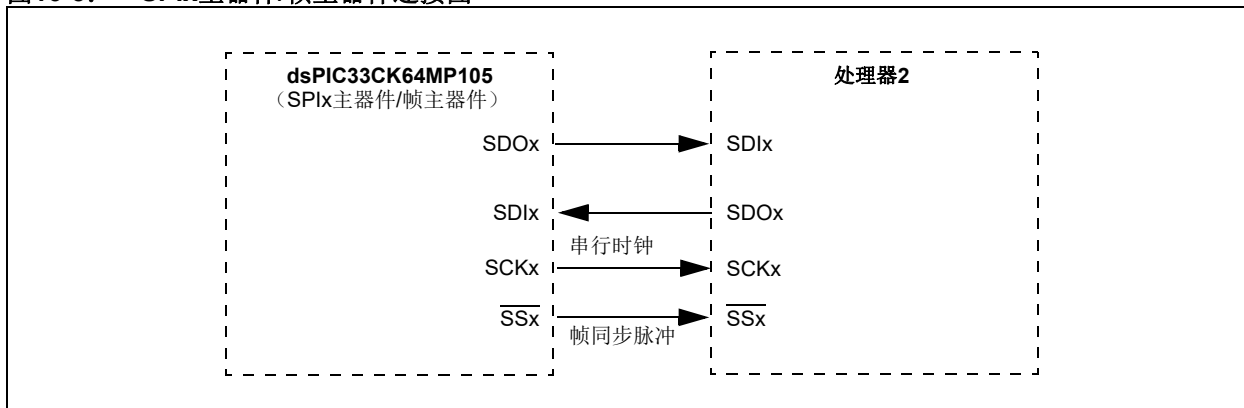


图16-4: SPIx主/从连接（增强型缓冲区模式）



**图16-5: SPIx主器件/帧主器件连接图**



# dsPIC33CK64MP105系列

图16-6: SPIx主器件/帧从器件连接图

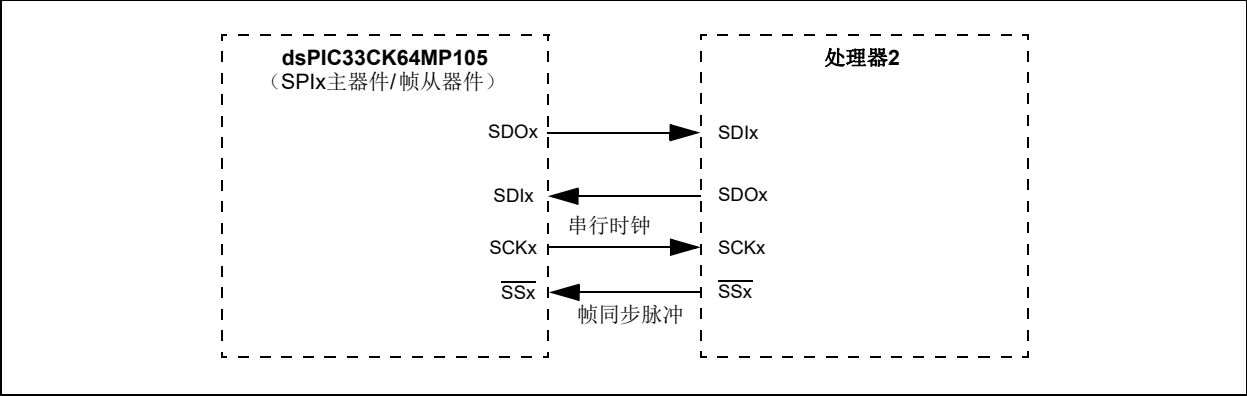


图16-7: SPIx从器件/帧主器件连接图

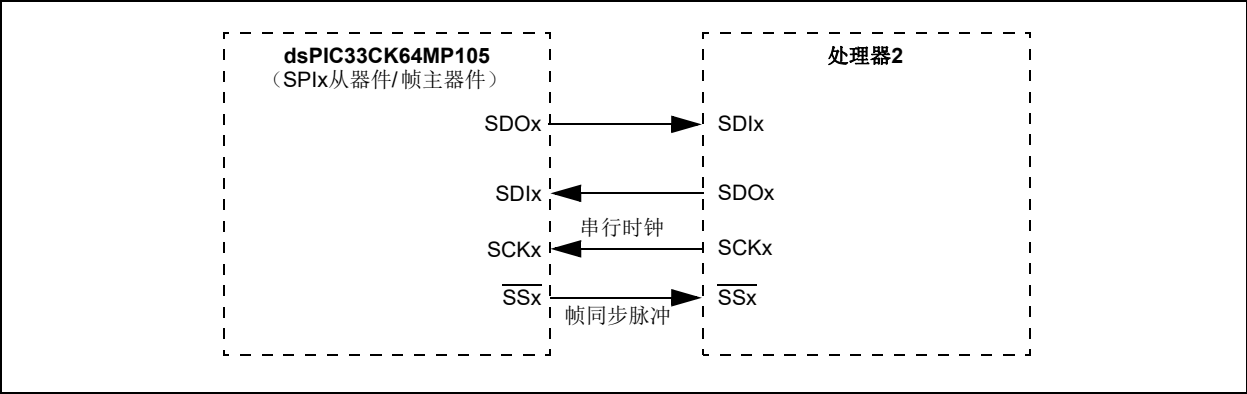
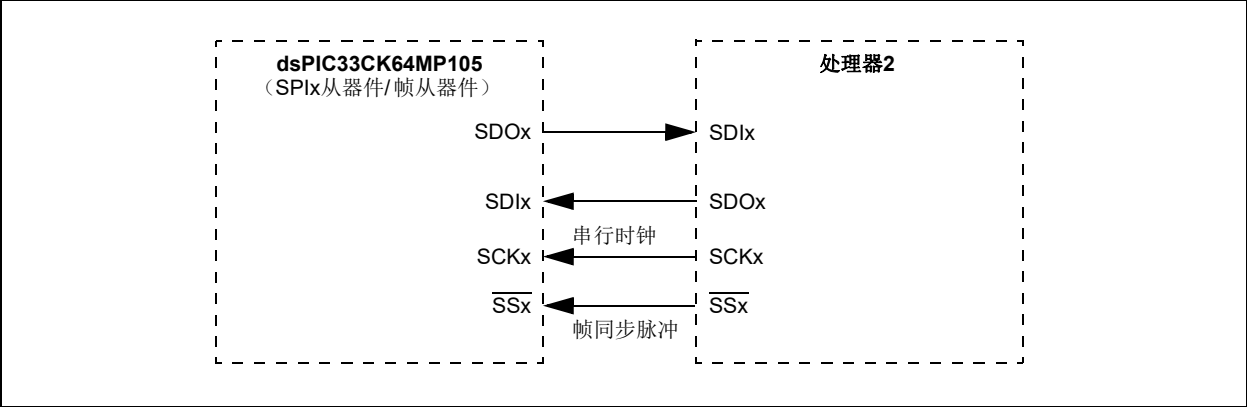


图16-8: SPIx从器件/帧从器件连接图



公式16-1: 器件速度与SPIx时钟速度之间的关系

$$\text{波特率} = \frac{FP}{(2 * (SPIxBRG + 1))}$$

其中:  
FP为外设总线时钟频率。

## 17.0 I<sup>2</sup>C

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。更多信息，请参见《dsPIC33/PIC24 系列数据手册》中的“**I<sup>2</sup>C**”([www.microchip.com/DS70000195](http://www.microchip.com/DS70000195))。

I<sup>2</sup>C 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。这些外设器件可以是串行 EEPROM、显示驱动器和 A/D 转换器等。

I<sup>2</sup>C 模块支持以下特性：

- 主器件和从器件逻辑相互独立
- 7 位和 10 位器件地址
- I<sup>2</sup>C 协议中定义的广播呼叫地址
- 时钟延长，为处理器提供延时以响应从器件的数据请求
- 支持 100 kHz 和 400 kHz 两种总线规范
- 可配置地址掩码
- 多主器件模式，防止仲裁时丢失报文
- 总线转发器模式，允许模块作为从器件接收所有报文，与地址无关
- 自动 SCL

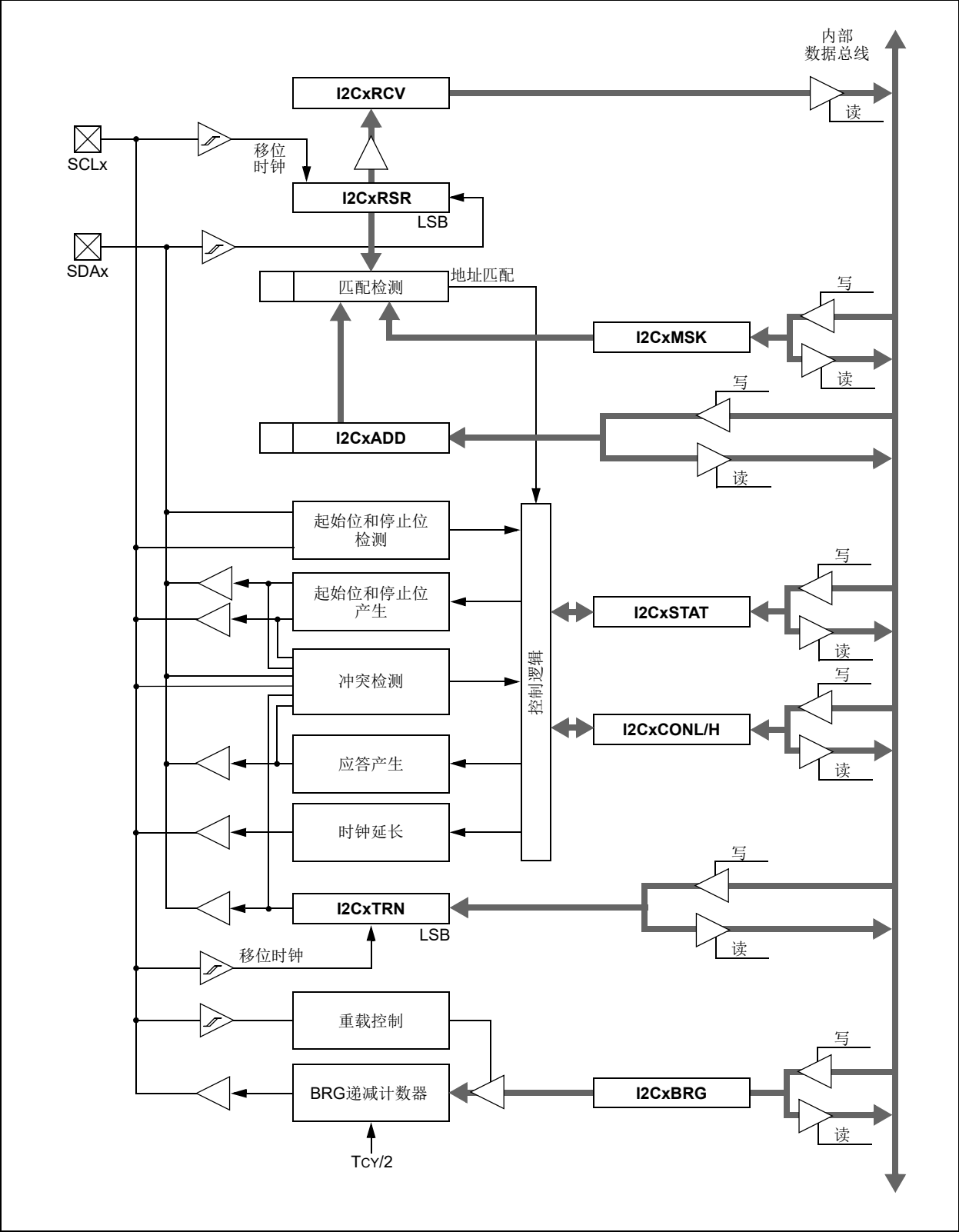
图 17-1 给出了模块的框图。

## 17.1 作为主器件在单主器件环境中通信

在主模式下发送报文的细节取决于作为通信对象的器件所采用的通信协议。通常，事件序列如下：

1. 在 SDAx 和 SCLx 上发出起始条件。
2. 将 I<sup>2</sup>C 器件地址字节发送到从器件，并指示要进行写操作。
3. 等待并验证来自从器件的应答。
4. 将第一个数据字节（有时称为命令）发送到从器件。
5. 等待并验证来自从器件的应答。
6. 将串行存储器地址低字节发送到从器件。
7. 重复步骤 4 和 5，直到发送完所有数据字节。
8. 在 SDAx 和 SCLx 上发出重复起始条件。
9. 将器件地址字节发送到从器件，并指示要进行读操作。
10. 等待并验证来自从器件的应答。
11. 使能主器件接收以接收串行存储器数据。
12. 在接收的数据字节末尾，产生 ACK 或 NACK 条件。
13. 在 SDAx 和 SCLx 上产生停止条件。

图17-1: I2Cx框图



## 17.2 设置用作总线主器件时的波特率

要计算波特率发生器的重载值，请使用[公式 17-1](#)。

**公式 17-1: 计算波特率重载值<sup>(1,2,3,4)</sup>**

$$I2CxBRG = ((1/F_{SCL} - \text{延时}) \cdot F_P/2) - 2$$

- 注 1:** 基于  $F_P = F_{OSC}/2$ 。
- 2:** 这些时钟速率值仅供参考。实际的时钟速率可能会受各种系统级参数的影响。应在目标应用中测量实际的时钟速率。
- 3:** 延时的典型值介于 110 ns 和 150 ns 之间。
- 4:** 明确禁止为 I2CxBRG 值使用 0 到 3。用户始终不应使用值 0x0、0x1、0x2 或 0x3 对 I2CxBRG 进行编程，否则会导致不确定的结果。

## 17.3 从器件地址掩码

I2CxMSK 寄存器（[寄存器 17-4](#)）用于指定在 7 位和 10 位寻址模式下为“无关位”的地址位。当 I2CxMSK 寄存器中的特定位置 1 (=1) 时，无论相应地址位的值是 0 还是 1，从模块都会作出响应。例如，当 I2CxMSK 设置为 0010000000 时，从模块将检测 0000000000 和 0010000000 这两个地址。

要 使 能 地 址 掩 码 ， 必 须 通 过 将 STRICT 位 (I2CxCONL[11]) 清零来禁止智能外设管理接口 (Intelligent Peripheral Management Interface, IPMI)。

**注:** 由于修改了 I<sup>2</sup>C 协议，[表 17-2](#) 中的地址得以保留，并且在从模式下不会获得应答。这包括涉及任何这些地址的所有地址掩码设置。

**表 17-1: I2Cx 时钟速率<sup>(1,2)</sup>**

Fcy	FSCL	I2CxBRG 值	
		十进制	十六进制
100 MHz	1 MHz	41	29
100 MHz	400 kHz	116	74
100 MHz	100 kHz	491	1EB
80 MHz	1 MHz	32	20
80 MHz	400 kHz	92	5C
80 MHz	100 kHz	392	188
60 MHz	1 MHz	24	18
60 MHz	400 kHz	69	45
60 MHz	100 kHz	294	126
40 MHz	1 MHz	15	0F
40 MHz	400 kHz	45	2D
40 MHz	100 kHz	195	C3
20 MHz	1 MHz	7	7
20 MHz	400 kHz	22	16
20 MHz	100 kHz	97	61

**注 1:** 基于  $F_P = F_{OSC}/2$ 。

- 2:** 这些时钟速率值仅供参考。实际的时钟速率可能会受各种系统级参数的影响。应在目标应用中测量实际的时钟速率。

# dsPIC33CK64MP105系列

表17-2: I2Cx保留地址<sup>(1)</sup>

从器件地址	R/W位	说明
0000 000	0	广播呼叫地址 <sup>(2)</sup>
0000 000	1	起始字节
0000 001	x	Cbus地址
0000 01x	x	保留
0000 1xx	x	HS 模式主器件代码
1111 0xx	x	10位从器件高字节 <sup>(3)</sup>
1111 1xx	x	保留

- 注 1: 无论地址掩码设置如何, 上述地址位都不会导致地址匹配。
- 2: 仅当GCEN = 1时才会应答该地址。
- 3: 只有10位寻址模式下的高字节才会与该地址发生匹配。



17.4 I<sup>2</sup>C控制/状态寄存器

寄存器17-1: I2CxCONL: I2Cx控制低位字寄存器

R/W-0	U-0	HC/R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL <sup>(1)</sup>	STRICT	A10M	DISSLW	SMEN <sup>(3)</sup>
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0	HC/R/W-0
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7				bit 0			

图注:	HC = 硬件清零位		
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 15 **I2CEN:** I2Cx使能位 (仅可通过软件写入)

- 1 = 使能I2Cx模块, 并将SDAx和SCLx引脚配置为串口引脚  
0 = 禁止I2Cx模块, 所有I<sup>2</sup>C引脚均由端口功能控制

bit 14 **未实现:** 读为0

bit 13 **I2CSIDL:** I2Cx空闲模式停止位

- 1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作

bit 12 **SCLREL:** SCLx释放控制位 (仅限I<sup>2</sup>C从模式)<sup>(1)</sup>

- 1 = 释放SCLx时钟  
0 = 保持SCLx时钟为低电平 (时钟延长)

如果STREN = 1: <sup>(2)</sup>

用户软件可以写0来启动时钟延长, 写1来释放时钟。在每个从器件数据字节发送开始时由硬件清零。在每个从器件地址字节接收结束时由硬件清零。在每个从器件数据字节接收结束时由硬件清零。

如果STREN = 0:

用户软件只能写1来释放时钟。在每个从器件数据字节发送开始时由硬件清零。在每个从器件地址字节接收结束时由硬件清零。

bit 11 **STRICT:** I2Cx严格保留地址规则使能位

- 1 = 强制执行严格保留寻址; 关于保留地址, 请参见表17-2。  
(在从模式下) —— 器件不对保留地址空间进行响应, 对于属于该类别的地址产生NACK。  
(在主模式下) —— 允许器件产生保留地址空间内的地址。  
0 = 应答保留寻址。  
(在从模式下) —— 器件会对属于保留地址空间的地址进行响应。当与任何保留地址发生匹配时, 器件将产生ACK。  
(在主模式下) —— 保留。

bit 10 **A10M:** 10位从器件地址标志位

- 1 = I2CxADD为10位从器件地址  
0 = I2CxADD为7位从器件地址

注 1: 在从器件发送开始时自动清零; 在从器件接收结束时自动清零。

2: 在从器件发送开始时自动清零。

3: SMB3EN配置位 (FDEVOP[10]) 在正常电平和SMBus 3.0电平之间进行选择。

# dsPIC33CK64MP105系列

## 寄存器17-1: I2CxCONL: I2Cx控制低位字寄存器 (续)

bit 9	<b>DISSLW:</b> 压摆率控制禁止位 1 = 标准速度模式 (100 kHz) 下禁止压摆率控制 (1 MHz 模式下同样禁止) 0 = 高速模式 (400 kHz) 下使能压摆率控制
bit 8	<b>SMEN:</b> SMBus输入电平使能位 <sup>(3)</sup> 1 = 使能输入逻辑以使阈值符合 SMBus 规范 0 = 禁止 SMBus 特定输入
bit 7	<b>GCEN:</b> 广播呼叫使能位 (仅限 I <sup>2</sup> C 从模式) 1 = 允许在 I2CxRSR 中接收到广播呼叫地址时产生中断; 已使能模块接收 0 = 禁止广播呼叫地址
bit 6	<b>STREN:</b> SCLx 时钟延长使能位 仅适用于 I <sup>2</sup> C 从模式; 与 SCKREL 位配合使用。 1 = 使能时钟延长 0 = 禁止时钟延长
bit 5	<b>ACKDT:</b> 应答数据位 在 I <sup>2</sup> C 主模式下, 在主器件接收模式期间。当用户在接收结束后发出应答序列时将发送的值。 在 I <sup>2</sup> C 从模式下, 当 AHEN = 1 或 DHEN = 1 时。当从器件在地址或数据接收结束后发出应答序列时将发送的值。 1 = 发送 NACK 0 = 发送 ACK
bit 4	<b>ACKEN:</b> 应答序列使能位 仅在 I <sup>2</sup> C 主模式下; 适用于主器件接收模式。 1 = 在 SDAx 和 SCLx 引脚上发出应答序列, 并发送 ACKDT 数据位 0 = 应答序列空闲
bit 3	<b>RCEN:</b> 接收使能位 (仅限 I <sup>2</sup> C 主模式) 1 = 使能 I <sup>2</sup> C 接收模式; 在接收到 8 位数据字节后由硬件自动清零 0 = 接收序列不在进行中
bit 2	<b>PEN:</b> 停止条件使能位 (仅限 I <sup>2</sup> C 主模式) 1 = 在 SDAx 和 SCLx 引脚上发出停止条件 0 = 停止条件空闲
bit 1	<b>RSEN:</b> 重复起始条件使能位 (仅限 I <sup>2</sup> C 主模式) 1 = 在 SDAx 和 SCLx 引脚上发出重复起始条件 0 = 重复起始条件空闲
bit 0	<b>SEN:</b> 启动条件使能位 (仅限 I <sup>2</sup> C 主模式) 1 = 在 SDAx 和 SCLx 引脚上发出起始条件 0 = 起始条件空闲

注 1: 在从器件发送开始时自动清零; 在从器件接收结束时自动清零。

2: 在从器件发送开始时自动清零。

3: SMB3EN 配置位 (FDEVOPT[10]) 在正常电平和 SMBus 3.0 电平之间进行选择。

## 寄存器17-2: I2CxCONH: I2Cx控制高位字寄存器

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

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
—	PCIE	SCIE	BOEN	SDAHT	—	AHEN	DHEN
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 15-7 未实现: 读为0

bit 6 **PCIE**: 停止条件中断允许位 (仅限 I<sup>2</sup>C 从模式)。

1 = 允许在检测到停止条件时产生中断

0 = 禁止停止条件检测中断

bit 5 **SCIE**: 起始条件中断允许位 (仅限 I<sup>2</sup>C 从模式)

1 = 允许在检测到起始或重复起始条件时产生中断

0 = 禁止起始条件检测中断

bit 4 **BOEN**: 缓冲区改写使能位 (仅限 I<sup>2</sup>C 从模式)

1 = 仅当 RBF 位 = 0 时, 在接收到地址/数据字节时, 更新 I2CxRCV 并产生 ACK 信号, 并忽略 I2COV 位的状态

0 = 仅当 I2COV 位清零时更新 I2CxRCV

bit 3 **SDAHT**: SDAx 保持时间选择位

1 = 在 SCLx 的下降沿之后 SDAx 至少保持 300 ns 的时间

0 = 在 SCLx 的下降沿之后 SDAx 至少保持 100 ns 的时间

bit 2 未实现: 读为0

bit 1 **AHEN**: 地址保持使能位 (仅限 I<sup>2</sup>C 从模式)

1 = 在匹配接收地址字节的 SCLx 的第 8 个下降沿后, SCLREL 位 (I2CxCONL[12]) 清零, SCLx 保持低电平

0 = 禁止地址保持

bit 0 **DHEN**: 数据保持使能位 (仅限 I<sup>2</sup>C 从模式)

1 = 在接收数据字节的 SCLx 的第 8 个下降沿之后, 从器件硬件清零 SCLREL 位 (I2CxCONL[12]) 且 SCLx 保持低电平

0 = 禁止数据保持

# dsPIC33CK64MP105系列

寄存器17-3: I2CxSTAT: I2Cx状态寄存器

HSC/R-0	HSC/R-0	HSC/R-0	U-0	U-0	HSC/R/C-0	HSC/R-0	HSC/R-0
ACKSTAT	TRSTAT	ACKTIM	—	—	BCL	GCSTAT	ADD10
bit 15						bit 8	

HS/R/C-0	HS/R/C-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
IWCOL	I2COV	D/ $\bar{A}$	P	S	R/ $\bar{W}$	RBF	TBF
bit 7							bit 0

图注:	C = 可清零位	HSC = 硬件置1/清零位
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
		HS = 硬件置1位

- bit 15 **ACKSTAT:** 应答状态位 (在所有主模式和从模式下更新)  
1 = 未收到来自从器件的应答  
0 = 收到来自从器件的应答
- bit 14 **TRSTAT:** 发送状态位 (作为I<sup>2</sup>C主器件工作时; 适用于主器件发送操作)  
1 = 主器件正在进行发送 (8位 + ACK)  
0 = 主器件不在进行发送
- bit 13 **ACKTIM:** 应答时间状态位 (仅在I<sup>2</sup>C从模式下有效)  
1 = 指示I<sup>2</sup>C总线处于应答序列中, 在SCLx时钟的第8个下降沿置1  
0 = 不处于应答序列中, 在SCLx时钟的第9个上升沿清零
- bit 12-11 **未实现:** 读为0
- bit 10 **BCL:** 总线冲突检测位 (禁止I<sup>2</sup>C模块 (I2CEN = 0) 时清零)  
1 = 发送操作期间检测到了总线冲突  
0 = 未检测到总线冲突
- bit 9 **GCSTAT:** 广播呼叫状态位 (在检测到停止条件后清零)  
1 = 接收到广播呼叫地址  
0 = 未接收到广播呼叫地址
- bit 8 **ADD10:** 10位地址状态位 (在检测到停止条件后清零)  
1 = 10位地址匹配  
0 = 10位地址不匹配
- bit 7 **IWCOL:** I2Cx写冲突检测位  
1 = 因为I<sup>2</sup>C模块忙, 尝试写I2CxTRN寄存器失败; 必须用软件清零  
0 = 未发生冲突
- bit 6 **I2COV:** I2Cx接收溢出标志位  
1 = 当I2CXRCV寄存器仍存有前一字节时接收到了新字节; 在发送模式下, I2COV为“无关位”, 必须用软件清零  
0 = 无溢出
- bit 5 **D/ $\bar{A}$ :** 数据/地址位 (作为I<sup>2</sup>C从器件工作时)  
1 = 指示上次接收的字节为数据  
0 = 表示上次接收或发送的字节是地址
- bit 4 **P:** I2Cx停止位  
当检测到起始、复位或停止条件时更新; 禁止I<sup>2</sup>C模块 (I2CEN = 0) 时清零。  
1 = 指示上次检测到停止位  
0 = 上次未检测到停止位

## 寄存器17-3: I2CxSTAT: I2Cx状态寄存器 (续)

bit 3	<b>S:</b> I2Cx起始位 当检测到起始、复位或停止条件时更新; 禁止I <sup>2</sup> C模块 (I2CEN = 0) 时清零。 1 = 指示上次检测到起始 (或重复起始) 位 0 = 上次未检测到起始位
bit 2	<b>R/W:</b> 读/写信息位 (作为I <sup>2</sup> C从器件工作时) 1 = 读: 表示数据自从器件输出 0 = 写: 表示数据输入到从器件
bit 1	<b>RBF:</b> 接收缓冲区满状态位 1 = 接收完成, I2CxRCV已满 0 = 接收未完成, I2CxRCV为空
bit 0	<b>TBF:</b> 发送缓冲区满状态位 1 = 发送正在进行, I2CxTRN已满 (8位数据) 0 = 发送完成, I2CxTRN为空

## 寄存器17-4: I2CxMSK: I2Cx从模式地址掩码寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	MSK[9:8]	
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
MSK[7:0]							
bit 7						bit 0	

### 图注:

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

bit 15-10	未实现: 读为0
bit 9-0	<b>MSK[9:0]:</b> I2Cx地址中bit x的掩码选择位 1 = 使能输入报文的地址中bit x的掩码; 在此位置上不需要位匹配 0 = 禁止bit x的掩码; 在此位置上需要位匹配

注:

## 18.0 单边沿半字节传输 (SENT)

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息,请参见《dsPIC33/PIC24系列参考手册》中的“单边沿半字节发送 (SENT) 模块” ([www.microchip.com/DS70005145](http://www.microchip.com/DS70005145))。

单边沿半字节传输 (Single-Edge Nibble Transmission, SENT) 模块基于 SAE J2716, “SENT – Single-Edge Nibble Transmission for Automotive Applications”。SENT 协议是一种基于连续下降沿的单向单线时间调制型串行通信协议。该协议旨在用于需要从传感器向引擎控制单元 (Engine Control Unit, ECU) 传送高分辨率传感器数据的应用。

SENTx 模块具有以下主要特性:

- 可选发送或接收模式
- 同步或异步发送模式
- 自动数据速率同步
- (可选) 接收模式下自动检测 CRC 错误
- (可选) 发送模式下通过硬件计算 CRC
- 支持可选的暂停脉冲周期
- 1 个报文帧的数据缓冲
- 发送/接收数据长度可选: 1 至 6 个半字节
- 自动检测帧错误

SENT 协议的时序基于一个预先确定的时间单位  $T_{\text{Tick}}$ 。必须预先配置发送器和接收器的  $T_{\text{Tick}}$ , 取值范围为 3 至 90  $\mu\text{s}$ 。SENT 报文帧以同步脉冲开始。同步脉冲的

作用是使接收器能够计算发送器所编码的报文的数据速率。SENT 规范允许  $T_{\text{Tick}}$  最高具有 20% 的偏差, 在此范围内报文均可通过验证。这使发送器和接收器可以使用可能不精确的、会随时间和温度而漂移的不同时钟运行。数据半字节的长度为 4 位, 其编码为数据值 + 12 个节拍。因此产生的 0 值为 12 个节拍, 产生的最大值 0xF 为 27 个节拍。

SENT 报文包含以下各项:

- 同步/校准周期: 56 个节拍时间
- 状态半字节: 12-27 个节拍时间
- 最多 6 个数据半字节: 12-27 个节拍时间
- CRC 半字节: 12-27 个节拍时间
- 可选的暂停脉冲周期: 12-768 个节拍时间

图 18-1 给出了 SENTx 模块的框图。

图 18-2 显示了典型 6 个半字节数据帧的构成, 数字代表每个部分的最小或最大节拍时间数量。

图 18-1: SENTx模块框图

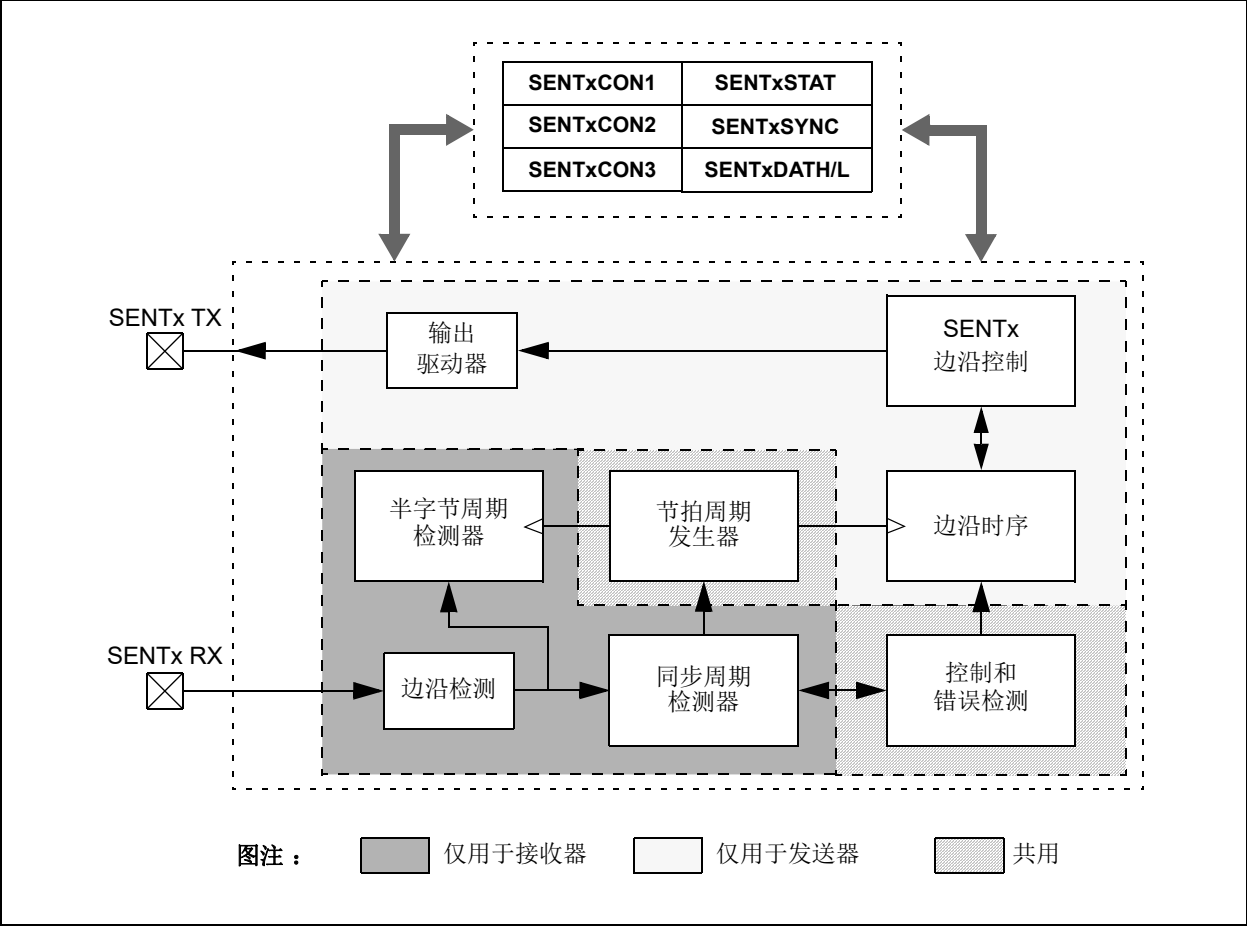
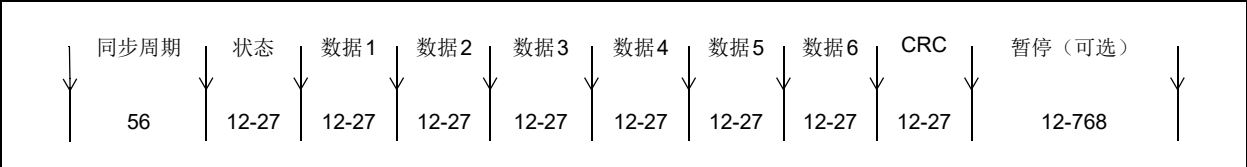


图 18-2: SENTx协议数据帧





## 18.1 发送模式

默认情况下，SENTx模块配置为发送操作。该模块可配置为连续异步报文帧发送，也可配置为软件触发的同步模式。使能后，发送器将发送同步脉冲，接着是适当数量的数据半字节、可选的CRC和可选的暂停脉冲。SENTx发送器使用的节拍周期通过向TICKTIME[15:0]（SENTxCON2[15:0]）位写入值进行设置。公式18-1给出了节拍周期的计算方式。

### 公式18-1： 节拍周期计算

$$TICKTIME[15:0] = \frac{T_{TICK}}{T_{CLK}} - 1$$

异步模式下可使用暂停脉冲来提供固定的报文帧时间周期。SENTx发送器使用的帧周期通过向FRAMETIME[15:0]（SENTxCON3[15:0]）位写入值进行设置。公式18-2给出了用于计算帧时间值的公式。

### 公式18-2： 帧时间计算

$$FRAMETIME[15:0] = T_{TICK}/T_{FRAME}$$

$$FRAMETIME[15:0] \geq 122 + 27N$$

$$FRAMETIME[15:0] \geq 848 + 12N$$

其中：

$T_{FRAME}$  = 报文总时间，单位为ms

$N$  = 报文中的数据半字节数，值为1-6

**注：** 无论FRAMETIME[15:0]值为何，模块都不会产生时间少于12个节拍的暂停周期。FRAMETIME[15:0]值超过2047时将对数据帧长度不起任何作用。

## 18.1.1 发送模式配置

### 18.1.1.1 初始化SENTx模块

要初始化模块，请执行以下步骤：

1. 写入RCVEN（SENTxCON1[11]）= 0以配置为发送模式。
2. 写入TXM（SENTxCON1[10]）= 0以配置为异步发送模式，或写入TXM = 1以配置为同步模式。
3. 向NIBCNT[2:0]（SENTxCON1[2:0]）写入相应的值以配置所需数据帧长度。
4. 写入CRCEN（SENTxCON1[8]）以配置为硬件或软件CRC计算。
5. 写入PPP（SENTxCON1[7]）以配置可选暂停脉冲。
6. 如果PPP = 1，则将TFRAME写入SENTxCON3。
7. 向SENTxCON2写入相应的值以配置所需节拍周期。
8. 允许中断并设置中断优先级。
9. 将初始状态和数据值写入SENTxDATH/L。
10. 如果CRCEN = 0，计算CRC并将得到的值写入CRC[3:0]（SENTxDATL[3:0]）。
11. 将SENTEN（SENTxCON1[15]）位置1以使能模块。

用户软件对SENTxDATH/L的更新必须在CRC完成之后、下一个报文帧的状态半字节之前执行。建议的方法是使用报文帧完成中断来触发数据写入。

18.2 接收模式

可通过将RCVEN (SENTxCON1[11]) 位置1来将该模块配置为接收操作。各下降沿之间的时间将与SYNCMIN[15:0] (SENTxCON3[15:0]) 和SYNCMAX[15:0] (SENTxCON2[15:0]) 进行比较, 如果测量时间在上下限之间, 则模块开始接收数据。经验证的同步时间在SENTxSYNC寄存器中捕捉, 然后计算节拍时间。后续下降沿经验证属于有效数据宽度范围之后, 将数据存储于SENTxDATL/H寄存器中。报文完成时将产生中断事件, 用户软件应在接收下一个半字节之前读取SENTx数据寄存器。公式18-3给出了SYNCMIN[15:0]和SYNCMAX[15:0]的计算公式。

公式18-3: SYNCMIN[15:0]和SYNCMAX[15:0]计算

$$TTICK = TCLK \cdot (TICKTIME[15:0] + 1)$$
$$FRAMETIME[15:0] = TTICK / TFRAME$$
$$SyncCount = 8 \times FRCV \times TTICK$$
$$SYNCMIN[15:0] = 0.8 \times SyncCount$$
$$SYNCMAX[15:0] = 1.2 \times SyncCount$$
$$FRAMETIME[15:0] \geq 122 + 27N$$
$$FRAMETIME[15:0] \geq 848 + 12N$$

其中:  
 $TFRAME$  = 报文总时间, 单位为ms  
 $N$  = 报文中的数据半字节数, 值为1-6  
 $FRCV = F_{CY} \times$  预分频比  
 $TCLK = F_{CY} /$  预分频比

TTICK = 3.0 μs且FCLK = 4 MHz时, SYNCMIN[15:0] = 76。

注: 为确保可以识别同步周期, 写入SYNCMIN[15:0]的值必须小于写入SYNCMAX[15:0]的值。

18.2.1 接收模式配置

18.2.1.1 初始化SENTx模块

要初始化模块, 请执行以下步骤:

1. 写入RCVEN (SENTxCON1[11]) = 1以配置为接收模式。
2. 向NIBCNT[2:0] (SENTxCON1[2:0]) 写入相应的值以配置所需数据帧长度。
3. 写入CRCEN (SENTxCON1[8]) 以配置为硬件或软件CRC验证。
4. 如果存在暂停脉冲, 则写入PPP (SENTxCON1[7]) = 1。
5. 将SYNCMAXx的值 (标称同步周期 + 20%) 写入SENTxCON2。
6. 将SYNCMINx的值 (标称同步周期 - 20%) 写入SENTxCON3。
7. 允许中断并设置中断优先级。
8. 将SNTEN (SENTxCON1[15]) 位置1以使能模块。

应在CRC完成之后、下一个报文帧的状态半字节之前从SENTxDATL/H寄存器中读取数据。建议的方法是使用报文帧完成中断触发。

## 18.3 SENT 控制/状态寄存器

寄存器18-1: SENTxCON1: SENTx控制寄存器1

R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SNTEN	—	SNTSIDL	—	RCVEN	TXM <sup>(1)</sup>	TXPOL <sup>(1)</sup>	CRCEN
bit 15							bit 8
R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PPP	SPCEN <sup>(2)</sup>	—	PS	—	NIBCNT2	NIBCNT1	NIBCNT0
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 15      **SNTEN**: SENTx使能位  
1 = 使能 SENTx  
0 = 禁止 SENTx
- bit 14      **未实现**: 读为0
- bit 13      **SNTSIDL**: SENTx空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12      **未实现**: 读为0
- bit 11      **RCVEN**: SENTx接收使能位  
1 = SENTx 作为接收器工作  
0 = SENTx 作为发送器 (传感器) 工作
- bit 10      **TXM**: SENTx发送模式位<sup>(1)</sup>  
1 = 只有在使用 SYNCTXEN 状态位触发时 SENTx 才发送数据帧  
0 = SNTEN = 1 时 SENTx 连续发送数据帧
- bit 9      **TXPOL**: SENTx发送极性位<sup>(1)</sup>  
1 = SENTx 数据输出引脚在空闲状态下为低电平  
0 = SENTx 数据输出引脚在空闲状态下为高电平
- bit 8      **CRCEN**: CRC使能位  
模块处于接收模式 (RCVEN = 1) 时:  
1 = SENTx使用首选的J2716方法对接收的数据执行CRC校验  
0 = SENTx不对接收的数据执行CRC校验  
模块处于发送模式 (RCVEN = 0) 时:  
1 = SENTx使用首选的J2716方法自动计算CRC  
0 = SENTx不计算CRC
- bit 7      **PPP**: 暂停脉冲存在位  
1 = SENTx 配置为发送 / 接收带有暂停脉冲的 SENT 报文  
0 = SENTx 配置为发送 / 接收不带暂停脉冲的 SENT 报文
- bit 6      **SPCEN**: 短PWM代码使能位<sup>(2)</sup>  
1 = 使能外部源进行 SPC 控制  
0 = 禁止外部源进行 SPC控制
- bit 5      **未实现**: 读为0

注 1: 该位在接收模式 (RCVEN = 1) 下不起任何作用。

注 2: 该位在发送模式 (RCVEN = 0) 下不起任何作用。

# dsPIC33CK64MP105系列

---

## 寄存器18-1: SENTxCON1: SENTx控制寄存器1 (续)

bit 4	<b>PS:</b> SENTx模块时钟预分频比位 1 = 4 分频 0 = 1分频
bit 3	<b>未实现:</b> 读为0
bit 2-0	<b>NIBCNT[2:0]:</b> 半字节计数控制位 111 = 保留; 不要使用 110 = 单个SENT数据包中, 模块发送/接收6个数据半字节 101 = 单个SENT数据包中, 模块发送/接收5个数据半字节 100 = 单个SENT数据包中, 模块发送/接收4个数据半字节 011 = 单个SENT数据包中, 模块发送/接收3个数据半字节 010 = 单个SENT数据包中, 模块发送/接收2个数据半字节 001 = 单个SENT数据包中, 模块发送/接收1个数据半字节 000 = 保留; 不要使用

**注 1:** 该位在接收模式 (RCVEN = 1) 下不起任何作用。

**2:** 该位在发送模式 (RCVEN = 0) 下不起任何作用。

寄存器18-2: SENTxSTAT: SENTx状态寄存器

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

R-0	R-0	R-0	R-0	R/C-0	R/C-0	R-0	HC/R/W-0
PAUSE	NIB2	NIB1	NIB0	CRCERR	FRMERR	RXIDLE	SYNCTXEN <sup>(1)</sup>
bit 7						bit 0	

图注:	C = 可清零位	HC = 硬件清零位
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 15-8      **未实现:** 读为0

bit 7      **PAUSE:** 暂停周期状态位  
 1 = 模块在发送 / 接收暂停周期  
 0 = 模块未在发送/接收暂停周期

bit 6-4      **NIB[2:0]:** 半字节状态位  
模块处于发送模式 (RCVEN = 0) 时:  
 111 = 模块正在发送 CRC 半字节  
 110 = 模块正在发送数据半字节6  
 101 = 模块正在发送数据半字节5  
 100 = 模块正在发送数据半字节4  
 011 = 模块正在发送数据半字节3  
 010 = 模块正在发送数据半字节2  
 001 = 模块正在发送数据半字节1  
 000 = 模块正在发送状态半字节或在暂停周期, 或未在发送

模块处于接收模式 (RCVEN = 1) 时:  
 111 = 模块正在接收 CRC 半字节或之前在接收该半字节时发生错误  
 110 = 模块正在接收数据半字节6或之前在接收该半字节时发生错误  
 101 = 模块正在接收数据半字节5或之前在接收该半字节时发生错误  
 100 = 模块正在接收数据半字节4或之前在接收该半字节时发生错误  
 011 = 模块正在接收数据半字节3或之前在接收该半字节时发生错误  
 010 = 模块正在接收数据半字节2或之前在接收该半字节时发生错误  
 001 = 模块正在接收数据半字节1或之前在接收该半字节时发生错误  
 000 = 模块正在接收状态半字节或等待同步周期

bit 3      **CRCERR:** CRC 状态位 (仅限接收模式)  
 1 = 对于 SENTxDATL/H 中的 1-6 数据半字节, 发生了 CRC 错误  
 0 = 未发生 CRC 错误

bit 2      **FRMERR:** 帧错误状态位 (仅限接收模式)  
 1 = 接收到的数据半字节少于 12 个节拍周期或大于 27 个节拍周期  
 0 = 未发生帧错误

bit 1      **RXIDLE:** SENTx接收器空闲状态位 (仅限接收模式)  
 1 = SENTx 数据总线处于空闲状态 (高电平) 的时间大于或等于 SYNCMAX[15:0]  
 0 = SENTx 数据总线未处于空闲状态

**注 1:** 在接收模式下 (RCVEN = 1), SYNCTXEN位是只读位。

## 寄存器18-2: SENTxSTAT: SENTx状态寄存器 (续)

bit 0 **SYNCTXEN**: SENTx同步周期状态/发送使能位<sup>(1)</sup>

模块处于接收模式 (RCVEN = 1) 时:

1 = 已检测到有效的同步周期; 模块正在接收半字节数据

0 = 未检测到同步周期; 模块未在接收半字节数据

模块处于异步发送模式 (RCVEN = 0, TXM = 0) 时:

使能模块后该位始终读为 1, 指示模块连续发送 SENTx 数据帧。禁止模块后该位读为 0。

模块处于同步发送模式 (RCVEN = 0, TXM = 1) 时:

1 = 模块正在发送 SENTx 数据帧

0 = 模块未在发送数据帧, 用户软件可以通过将 SYNCTXEN 置 1 来启动另一次数据帧发送

**注 1:** 在接收模式下 (RCVEN = 1), SYNCTXEN 位是只读位。

## 寄存器 18-3: SENTxDATL: SENTx接收数据低位字寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DATA4[3:0]				DATA5[3:0]			
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
DATA6[3:0]				CRC[3:0]			
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 15-12 **DATA4[3:0]:** 数据半字节4数据位

bit 11-8 **DATA5[3:0]:** 数据半字节5数据位

bit 7-4 **DATA6[3:0]:** 数据半字节6数据位

bit 3-0 **CRC[3:0]:** CRC半字节数据位

**注 1:** 在接收模式下 (RCVEN = 1), 这些寄存器位为只读位。在发送模式下, 如果使能了自动CRC计算 (RCVEN = 0, CRCEN = 1), 则CRC[3:0]位为只读位。

## 寄存器 18-4: SENTxDATH: SENTx接收数据高位字寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STAT[3:0]				DATA1[3:0]			
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
DATA2[3:0]				DATA3[3:0]			
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 15-12 **STAT[3:0]:** 状态半字节数据位

bit 11-8 **DATA1[3:0]:** 数据半字节1数据位

bit 7-4 **DATA2[3:0]:** 数据半字节2数据位

bit 3-0 **DATA3[3:0]:** 数据半字节3数据位

**注 1:** 在接收模式下 (RCVEN = 1), 这些寄存器位为只读位。在发送模式下, 如果使能了自动CRC计算 (RCVEN = 0, CRCEN = 1), 则CRC[3:0]位为只读位。

注:



## 19.0 TIMER1

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“Timer1 模块”（[www.microchip.com/DS70005279](http://www.microchip.com/DS70005279)）。

Timer1 模块是一个 16 位定时器，可作为自由运行的间隔定时器/计数器。

Timer1 模块具有不同于其他定时器的独特特性，如下：

- 可以异步计数器模式工作
- 异步定时器
- 可在 CPU 休眠模式下工作
- 可通过软件选择预分频比（1:1、1:8、1:64 和 1:256）
- 外部时钟选择控制
- 可选择将 Timer1 外部时钟输入（T1CK）与内部器件时钟同步，时钟同步在预分频之后执行

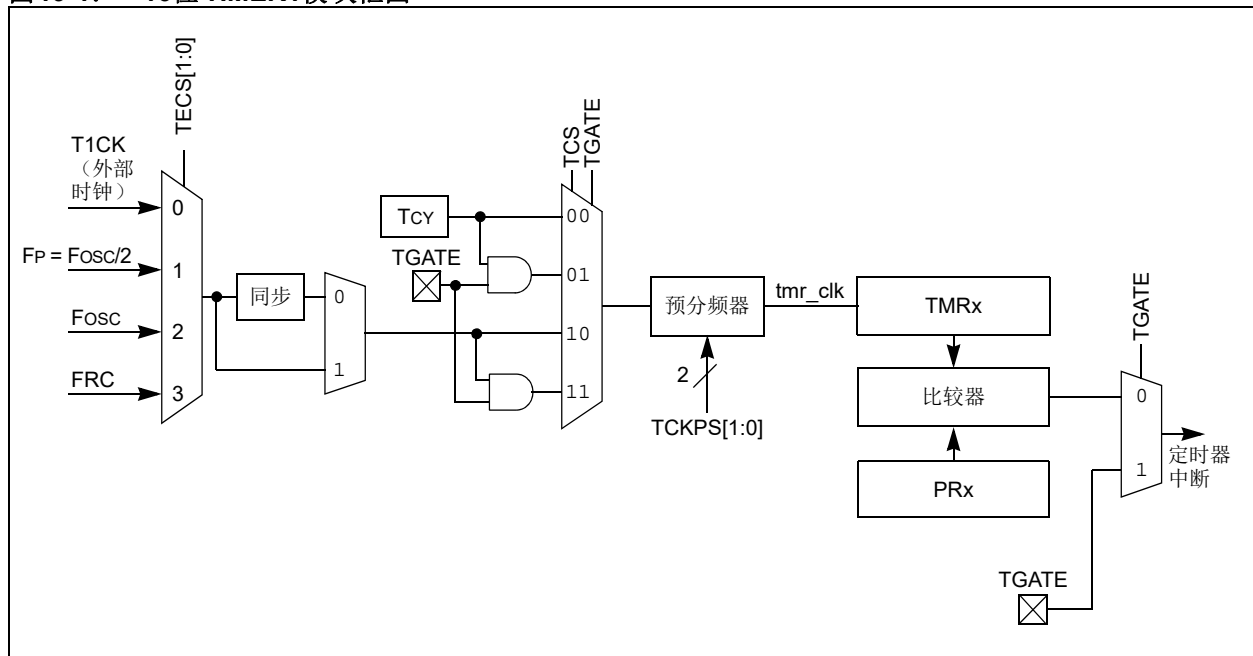
如果将 Timer1 用于 SCCP，则定时器应在同步模式下工作。

Timer1 模块可以工作于以下模式之一：

- 定时器模式
- 门控定时器模式
- 同步计数器模式
- 异步计数器模式

图 19-1 给出了 Timer1 的框图。

图 19-1: 16 位 TIMER1 模块框图



# dsPIC33CK64MP105系列

## 19.1 Timer1控制寄存器

寄存器19-1: T1CON: TIMER1控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0
TON <sup>(1)</sup>	—	SIDL	TMWDIS	TMWIP	PRWIP	TECS1	TECS0
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
TGATE	—	TCKPS1	TCKPS0	—	TSYNC <sup>(1)</sup>	TCS <sup>(1)</sup>	—
bit 7							bit 0

图注:

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

- bit 15      **TON:** Timer1使能位<sup>(1)</sup>  
1 = 启动16位Timer1  
0 = 停止16位Timer1
- bit 14      **未实现:** 读为0
- bit 13      **SIDL:** Timer1空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12      **TMWDIS:** 异步Timer1写禁止位  
1 = 当对TMR1或PR1的无响应写操作 (posted write) 与异步时钟域同步时, 忽略对定时器的写操作  
0 = 在异步模式下使能连续写操作
- bit 11      **TMWIP:** 异步Timer1写进行位  
1 = 在异步模式下对定时器进行的写操作为待处理状态  
0 = 在异步模式下对定时器进行的写操作已完成
- bit 10      **PRWIP:** 异步周期写进行位  
1 = 在异步模式下对周期寄存器进行的写操作为待处理状态  
0 = 在异步模式下对周期寄存器进行的写操作已完成
- bit 9-8      **TECS[1:0]:** Timer1扩展时钟选择位  
11 = FRC时钟  
10 = FOSC振荡器时钟  
01 = Fp = FOSC/2外设时钟  
00 = 来自T1CK引脚的外部时钟
- bit 7      **TGATE:** Timer1门控时间累加使能位  
当TCS = 1时:  
忽略该位。  
当TCS = 0时:  
1 = 使能门控时间累加  
0 = 禁止门控时间累加
- bit 6      **未实现:** 读为0

**注 1:** 将Timer1使能为外部同步计数器模式 (TCS = 1, TSYNC = 1, TON = 1) 时, 用户软件试图对TMR1寄存器进行的任何写操作都会被忽略。

## 寄存器19-1: T1CON: TIMER1控制寄存器 (续)

bit 5-4	<b>TCKPS[1:0]:</b> Timer1输入时钟预分频比选择位 11 = 1:256 10 = 1:64 01 = 1:8 00 = 1:1
bit 3	未实现: 读为0
bit 2	<b>TSYNC:</b> Timer1外部时钟输入同步选择位 <sup>(1)</sup> <u>当TCS = 1时:</u> 1 = 同步外部时钟输入 0 = 不同步外部时钟输入 <u>当TCS = 0时:</u> 忽略该位。
bit 1	<b>TCS:</b> Timer1时钟源选择位 <sup>(1)</sup> 1 = 外部时钟源由TECS[1:0]选择 0 = 内部外设时钟 (FP)
bit 0	未实现: 读为0

**注 1:** 将Timer1使能为外部同步计数器模式 (TCS = 1, TSYNC = 1, TON = 1) 时, 用户软件试图对TMR1寄存器进行的任何写操作都会被忽略。

注:

## 20.0 捕捉/比较/PWM/定时器模块 (SCCP/MCCP)

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。有关 MCCP/SCCP 模块的更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“捕捉/比较/PWM/定时器 (MCCP 和 SCCP)” ([www.microchip.com/DS30003035](http://www.microchip.com/DS30003035))。

dsPIC33CK64MP105 系列器件包含四个 SCCP 和一个 MCCP 捕捉/比较/PWM/定时器基本模块，提供了早期 PIC24F 器件的三种不同外设的功能。这类模块可在以下三种主要模式之一下运行：

- 通用定时器
- 输入捕捉
- 输出比较/PWM

该模块以两种不同的形式提供，区别在于模块可生成的 PWM 输出的数量。单捕捉/比较/PWM (Single Capture/Compare/PWM, SCCP) 输出模块只提供一个 PWM 输出。

多捕捉/比较/PWM (Multiple Capture/Compare/PWM, MCCP) 输出模块可根据特定器件的引脚数量提供最多六个输出和丰富的电源控制功能。这两种模块的所有其他功能都是相同的。

在任何时刻，SCCPx 和 MCCPx 模块都只能在三种主要模式之一下运行。除非将模块重新配置为新模式，否则其他模式不可用。

图20-1给出了模块的概念框图。所有三种模式共用一个时基发生器和一个通用定时器寄存器对 (CCPxTMRH/L)；其他共用硬件组件根据特定模式的需要添加。

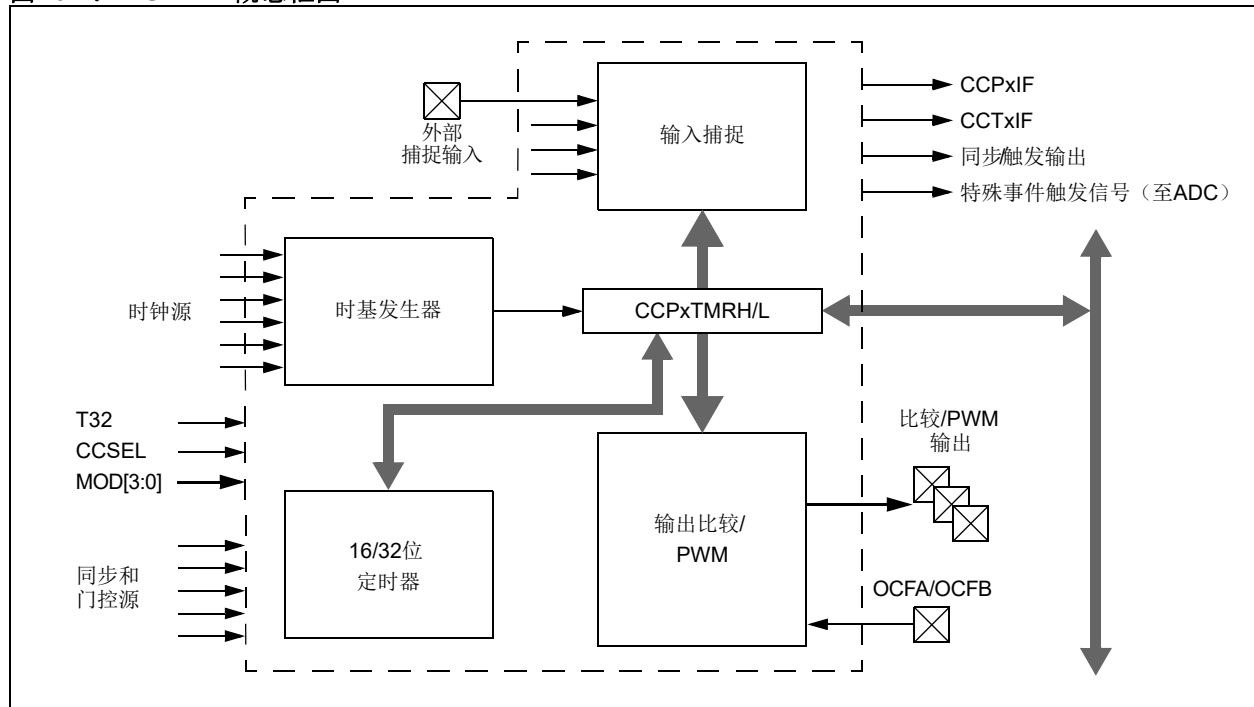
每个模块共有6个控制和状态寄存器：

- CCPxCON1L (寄存器 20-1)
- CCPxCON1H (寄存器 20-2)
- CCPxCON2L (寄存器 20-3)
- CCPxCON2H (寄存器 20-4)
- CCPxCON3H (寄存器 20-6)
- CCPxSTATL (寄存器 20-7)

每个模块还包含8个缓冲/计数器寄存器，这些寄存器用作定时器值寄存器或数据保持缓冲区：

- CCPxTMRH/CCPxTMRL (CCPx 定时器高位字/低位字计数器)
- CCPxPRH/CCPxPRL (CCPx 定时器周期高位字/低位字)
- CCPxRA (CCPx 主输出比较数据缓冲区)
- CCPxRB (CCPx 辅助输出比较数据缓冲区)
- CCPxBUFH/CCPxBUFL (CCPx 输入捕捉高位字/低位字缓冲区)

图20-1: SCCPx 概念框图

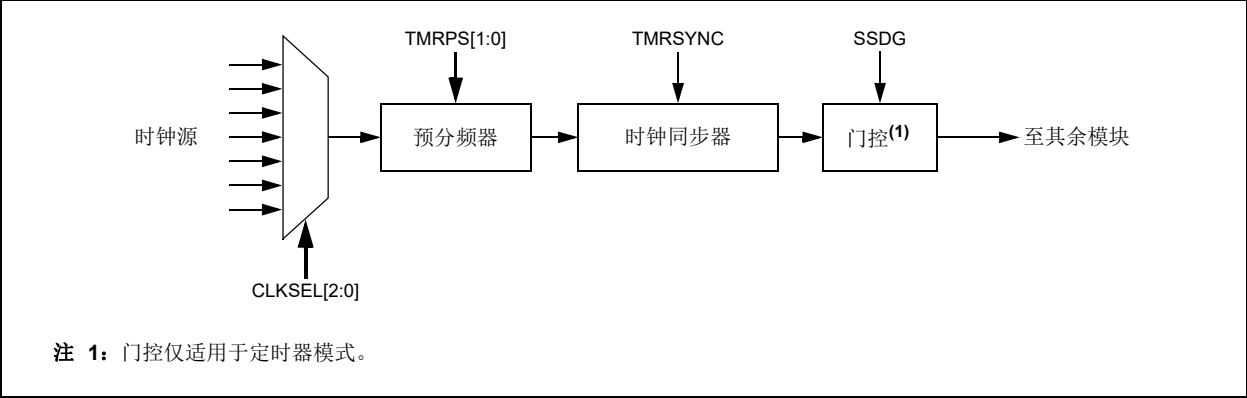


20.1 时基发生器

定时器时钟发生器（Timer Clock Generator, TCG）使用单片机上已有的时钟信号之一为模块的内部时基生成时钟。该内部时基用作模块在三种主要模式下的时间参考。图20-2给出了内部时基。

有8个输入可供时钟发生器使用，这些输入使用CLKSEL[2:0]位（CCPxCON1L[10:8]）进行选择。可用时钟源包括FRC和LPRC、辅助振荡器以及TCLKI外部时钟输入。系统时钟为默认时钟源（CLKSEL[2:0] = 000）。

图20-2： 定时器时钟发生器



20.2 通用定时器

当CCSEL = 0且MOD[3:0] = 0000时，选择定时器模式。定时器可用作一个32位定时器或双16位定时器，具体取决于T32位的设置（表20-1）。

表20-1： 定时器工作模式

T32 (CCPxCON1L[5])	工作模式
0	双定时器模式（16位）
1	定时器模式（32位）

双16位定时器模式提供简单的定时器功能，具有两个独立的16位定时器/计数器。主定时器使用CCPxTMRL和CCPxPRL。仅主定时器可与器件上的其他模块交互。它会生成SCCPx同步输出信号以供其他SCCP模块使用。它还可使用其他模块生成的SYNC[4:0]位信号。

辅助定时器使用CCPxTMRH和CCPxPRH。它仅用作调度CPU事件的周期性中断源，不会像主时基那样生成输出同步/触发信号。在双定时器模式下，CCPx辅助定时器周期寄存器CCPxPRH生成供器件上许多其他模块使用的SCCP比较事件（CCPxIF）。

32位定时器模式将CCPxTMRL和CCPxTMRH寄存器一起用作一个32位定时器。当CCPxTMRL溢出时，CCPxTMRH将递增1。在需要跟踪一段较长时间时，该模式可提供简单的定时器功能。请注意，在对CCPxTMRL或CCPxPRH寄存器进行写操作以初始化32位定时器之前，应先将T32位（CCPxCON1L[5]）置1。

20.2.1 同步和触发操作

在16位和32位模式下，定时器还可用于同步或触发操作。这两种操作均使用SYNC[4:0]位（CCPxCON1H[4:0]）来确定输入信号源，区别在于该信号对定时器的作用。

对于同步操作，当SYNC[4:0]选择的输入置为有效时，定时器复位或清零。定时器将立即从零开始重新计数，出于其他原因而需要保持计数的情况除外。只要TRIGEN位（CCPxCON1H[7]）清零，便会使用同步操作。SYNC[4:0]可以是除11111外的任一值。

对于触发操作，定时器将保持在复位状态，直到SYNC[4:0]选择的输入置为有效；置为有效后，定时器开始计数。只要TRIGEN位置1，便会使用触发操作。在触发模式下，只要CCPTRIG位（CCPxSTATL[7]）置1，定时器就会在触发事件后继续运行。要清零CCPTRIG，必须将TRCLR位（CCPxSTATL[5]）置1以清除触发事件、复位定时器并将其保持为零，直到发生另一个触发事件。对于dsPIC33CK64MP105系列器件，触发操作只能用于系统时钟为时基源的情况（CLKSEL[2:0] = 000）。

图20-3： 双16位定时器模式

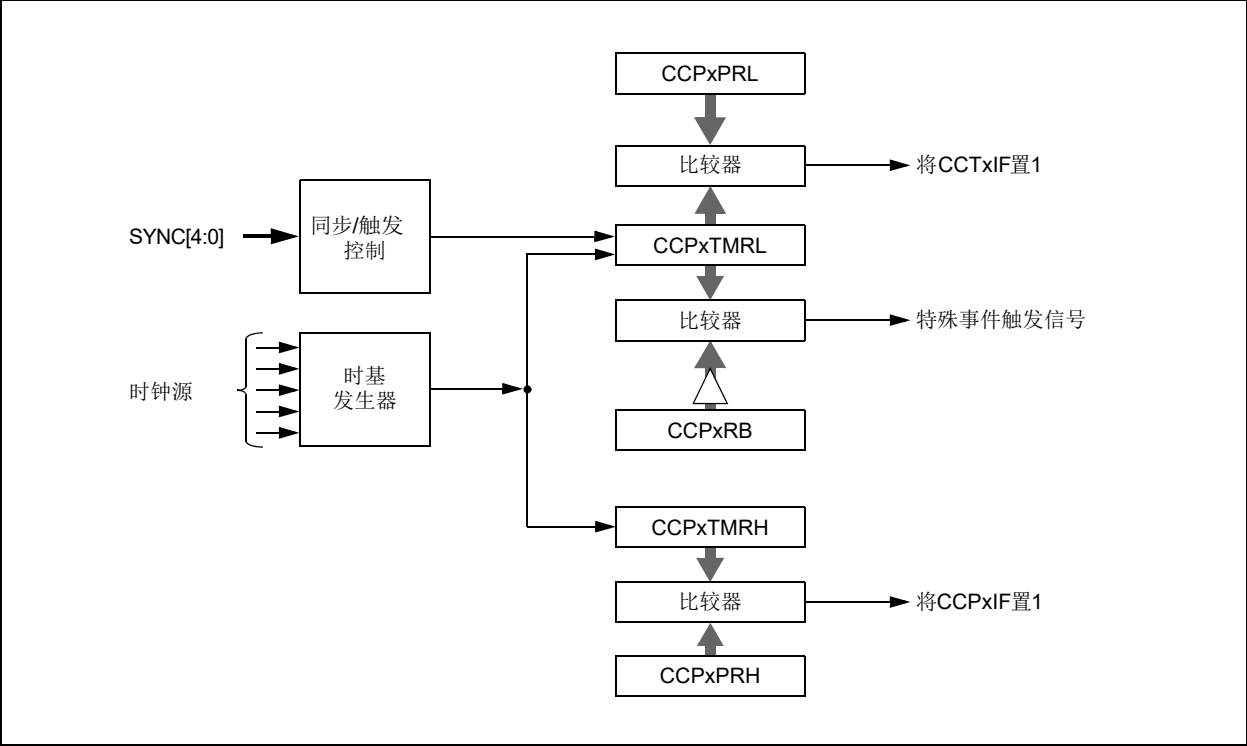
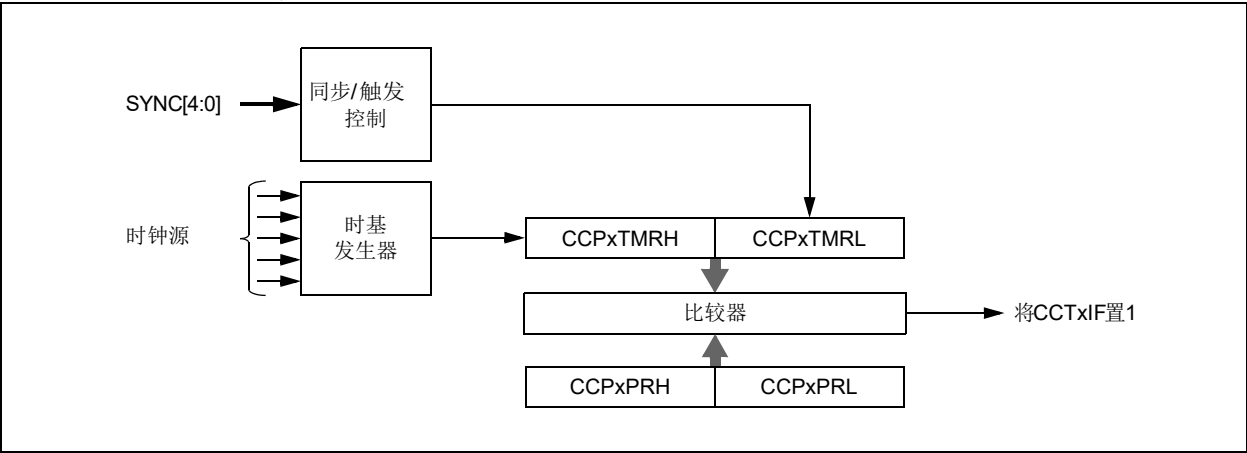


图20-4： 32位定时器模式





20.3 输出比较模式

输出比较模式将定时器寄存器的值与一个或两个比较寄存器的值进行比较，具体取决于其工作模式。在发生比较匹配事件时，输出比较x模块能够生成单个输出跳变

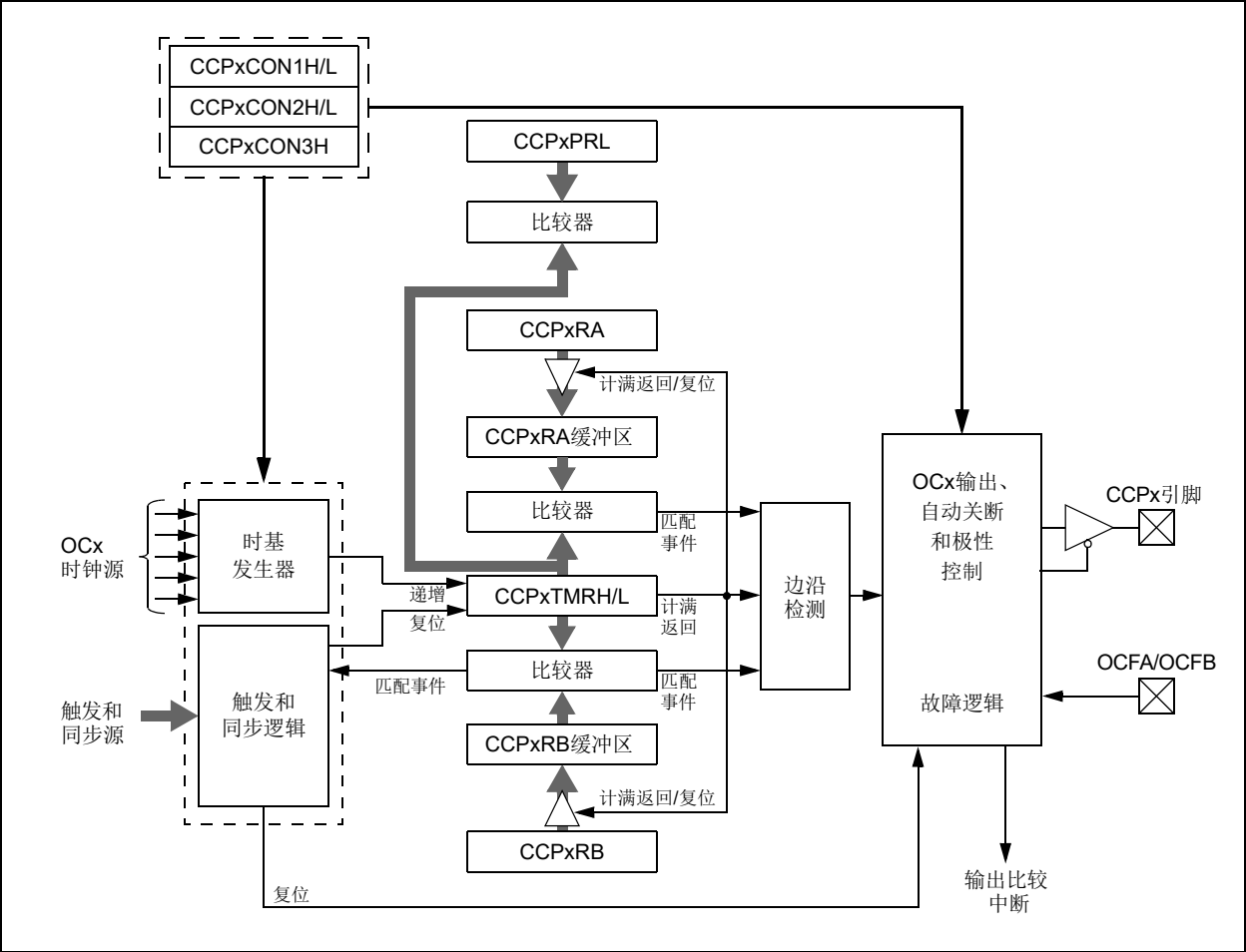
或一系列输出脉冲。与大多数PIC MCU外设一样，输出比较x模块还可在发生比较匹配事件时产生中断。

表20-2给出了在输出比较模式下可用的各种模式。

表20-2： 输出比较x/PWMx模式

MOD[3:0] (CCPxCON1L[3:0])	T32 (CCPxCON1L[5])	工作模式	
0001	0	比较匹配时输出高电平（16位）	单边沿模式
0001	1	比较匹配时输出高电平（32位）	
0010	0	比较匹配时输出低电平（16位）	
0010	1	比较匹配时输出低电平（32位）	
0011	0	比较匹配时输出翻转（16位）	
0011	1	比较匹配时输出翻转（32位）	
0100	0	双边沿比较（16位）	双边沿模式
0101	0	双边沿比较（16位缓冲）	PWM模式

图20-5： 输出比较x框图



20.4 输入捕捉模式

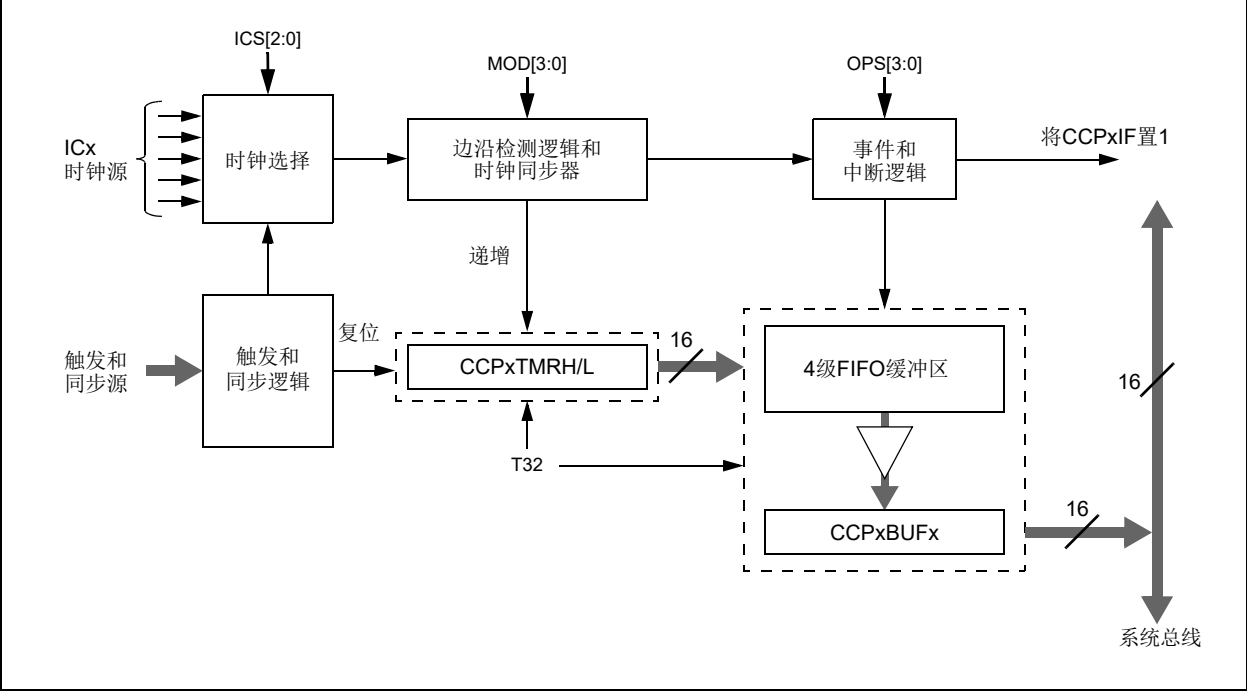
输入捕捉模式用于在输入引脚或其他内部触发源上有事件发生时，捕捉来自独立时基的定时器值。输入捕捉功能在需要频率（时间周期）和脉冲测量的应用中很有用。图20-6给出了输入捕捉模式的简化框图。

输入捕捉模式使用专用的16/32位同步递增计数定时器来实现捕捉功能。当发生捕捉事件时，定时器值将写入FIFO。可使用CCPxTMRH/L寄存器读取内部值（存在同步延时）。  
要使用输入捕捉模式，必须将CCSEL位（CCPxCON1L[4]）置1。T32和MOD[3:0]位用于选择合适的捕捉模式，如表20-3所示。

表20-3： 输入捕捉x模式

MOD[3:0] (CCPxCON1L[3:0])	T32 (CCPxCON1L[5])	工作模式
0000	0	边沿检测（16位捕捉）
0000	1	边沿检测（32位捕捉）
0001	0	每个上升沿（16位捕捉）
0001	1	每个上升沿（32位捕捉）
0010	0	每个下降沿（16位捕捉）
0010	1	每个下降沿（32位捕捉）
0011	0	每个上升沿/下降沿（16位捕捉）
0011	1	每个上升沿/下降沿（32位捕捉）
0100	0	每4个上升沿（16位捕捉）
0100	1	每4个上升沿（32位捕捉）
0101	0	每16个上升沿（16位捕捉）
0101	1	每16个上升沿（32位捕捉）

图20-6： 输入捕捉x框图



## 20.5 辅助输出

SCCPx模块具有一个辅助输出，方便其他外设访问模块内部信号。辅助输出旨在连接其他SCCP模块或其他数字外设，从而提供以下类型的功能：

- 时基同步
- 外设触发和时钟输入
- 信号门控

使用AUXOUT[1:0]控制位（CCPxCON2H[4:3]）选择输出信号的类型。输出信号的类型还取决于模块的工作模式。

表20-4： 辅助输出

AUXOUT[1:0]	CCSEL	MOD[3:0]	备注	信号说明
00	x	xxxx	禁止辅助输出	无输出
01	0	0000	时基模式	时基周期复位或计满返回
10				特殊事件触发输出
11				无输出
01	0	0001 至 1111	输出比较模式	时基周期复位或计满返回
10				输出比较事件信号
11				输出比较信号
01	1	xxxx	输入捕捉模式	时基周期复位或计满返回
10				反映ICDIS位的值
11				输入捕捉事件信号

# dsPIC33CK64MP105系列

## 20.6 SCCP/MCCP控制/状态寄存器

寄存器 20-1: **CCPxCON1L: CCPx控制1低位字寄存器**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CCPON	—	CCPSIDL	CCPSLP	TMRSYNC	CLKSEL2	CLKSEL1	CLKSEL0
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
TMRPS1	TMRPS0	T32	CCSEL	MOD3	MOD2	MOD1	MOD0
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 15        **CCPON:** CCPx模块使能位  
               1 = 使能模块, 并由MOD[3:0]控制位指定工作模式  
               0 = 禁止模块
- bit 14        **未实现:** 读为0
- bit 13        **CCPSIDL:** CCPx空闲模式停止位  
               1 = 当器件进入空闲模式时, 模块停止工作  
               0 = 在空闲模式下模块继续工作
- bit 12        **CCPSLP:** CCPx休眠模式使能位  
               1 = 休眠模式下模块继续工作  
               0 = 休眠模式下模块不工作
- bit 11        **TMRSYNC:** 时基时钟同步位  
               1 = 选择异步模块时基时钟, 并且该时钟与内部系统时钟同步 (CLKSEL[2:0] ≠ 000)  
               0 = 选择同步模块时基时钟, 并且该时钟不需要同步 (CLKSEL[2:0] = 000)
- bit 10-8      **CLKSEL[2:0]** CCPx时基时钟选择位  
               111 = PPS TxCK输入  
               110 = CLC4  
               101 = CLC3  
               100 = CLC2  
               011 = CLC1  
               010 = 保留  
               001 = 参考时钟 (REFCLKO)  
               000 = 外设时钟 (Fp = Fosc/2)
- bit 7-6      **TMRPS[1:0]:** 时基预分频比选择位  
               11 = 1:64 预分频比  
               10 = 1:16 预分频比  
               01 = 1:4 预分频比  
               00 = 1:1 预分频比
- bit 5        **T32:** 32位时基选择位  
               1 = 将32位时基用于定时器、单边沿输出比较或输入捕捉功能  
               0 = 将16位时基用于定时器、单边沿输出比较或输入捕捉功能
- bit 4        **CCSEL:** 捕捉/比较模式选择位  
               1 = 输入捕捉外设  
               0 = 输出比较/PWM/定时器外设 (确切的功能由MOD[3:0]位选择)

## 寄存器 20-1: CCPxCON1L: CCPx控制1低位字寄存器 (续)

bit 3-0

**MOD[3:0]:** CCPx模式选择位

CCSEL = 1 (输入捕捉模式) 时:

1xxx = 保留

011x = 保留

0101 = 每 16 个上升沿捕捉一次

0100 = 每 4 个上升沿捕捉一次

0011 = 每个上升沿和下降沿捕捉一次

0010 = 每个下降沿捕捉一次

0001 = 每个上升沿捕捉一次

0000 = 每个上升沿和下降沿捕捉一次 (边沿检测模式)

CCSEL = 0 (输出比较/定时器模式) 时:

1111 = 外部输入模式: 禁止脉冲发生器, 由 ICS[2:0] 选择源

1110 = 保留

110x = 保留

10xx = 保留

0111 = 保留

0110 = 保留

0101 = 双边沿缓冲比较模式

0100 = 双边沿比较模式

0011 = 16 位/32 位单边沿模式, 在比较匹配时翻转输出

0010 = 16 位/32 位单边沿模式, 在比较匹配时将输出驱动为低电平

0001 = 16 位/32 位单边沿模式, 在比较匹配时将输出驱动为高电平

0000 = 16 位/32 位定时器模式, 禁止输出功能

# dsPIC33CK64MP105系列

寄存器 20-2: CCPxCON1H: CCPx控制1高位字寄存器

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
OPSSRC <sup>(1)</sup>	RTRGEN <sup>(2)</sup>	—	—	OPS3 <sup>(3)</sup>	OPS2 <sup>(3)</sup>	OPS1 <sup>(3)</sup>	OPS0 <sup>(3)</sup>
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
TRIGEN	ONESHOT	ALTSYNC	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
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 15      **OPSSRC:** 输出后分频器源选择位<sup>(1)</sup>  
             1 = 输出后分频器调整模块触发输出事件发生的频率  
             0 = 输出后分频器调整时基中断事件发生的频率
- bit 14      **RTRGEN:** 重新触发使能位<sup>(2)</sup>  
             1 = TRIGEN位 = 1时可重新触发时基  
             0 = TRIGEN位 = 1时不可重新触发时基
- bit 13-12   **未实现:** 读为0
- bit 11-8    **OPS3[3:0]:** CCPx中断输出后分频比选择位<sup>(3)</sup>  
             1111 = 每发生16次时基周期匹配产生一次中断  
             1110 = 每发生15次时基周期匹配产生一次中断  
             ...  
             0100 = 每发生5次时基周期匹配产生一次中断  
             0011 = 每发生4次时基周期匹配或4次输入捕捉事件产生一次中断  
             0010 = 每发生3次时基周期匹配或3次输入捕捉事件产生一次中断  
             0001 = 每发生2次时基周期匹配或2次输入捕捉事件产生一次中断  
             0000 = 每次发生时基周期匹配或输入捕捉事件产生一次中断
- bit 7       **TRIGEN:** CCPx触发使能位  
             1 = 使能时基的触发操作  
             0 = 禁止时基的触发操作
- bit 6       **ONESHOT:** 单触发模式使能位  
             1 = 使能单触发模式; 触发持续时间由OSCNT[2:0]设置  
             0 = 禁止单触发模式
- bit 5       **ALTSYNC:** CCPx备用同步输出信号选择位  
             1 = 将备用信号用作模块同步输出信号  
             0 = 模块同步输出信号为时基复位/计满返回事件
- bit 4-0     **SYNC[4:0]:** CCPx同步源选择位  
             有关输入的定义, 请参见表20-5。

- 注 1:** 该控制位在输入捕捉模式下不起作用。  
**注 2:** 该控制位在 TRIGEN = 0时不起作用。  
**注 3:** 在输入捕捉模式下, 输出后分频比设置为1:5到1:16 (0100-1111) 时将导致FIFO缓冲区溢出。

表 20-5: 同步源

SYNC[4:0]	同步源
00000	无: 定时器在 CCPxPR 匹配或达到 FFFFh 时计满返回
00001	同步输出 SCCP1
00010	同步输出 SCCP2
00011	同步输出 SCCP3
00100	同步输出 SCCP4
00101-01000	保留
01001	INT0
01010	INT1
01011	INT2
01100	UART1 RX 边沿检测
01101	UART1 TX 边沿检测
01110	UART2 RX 边沿检测
01111	UART2 TX 边沿检测
10000	CLC1 输出
10001	CLC2 输出
10010	CLC3 输出
10011	CLC4 输出
10100	UART3 RX 边沿检测
10101	UART3 TX 边沿检测
10110	同步输出 MCCP5
10111	比较器 1 输出
11000	比较器 2 输出
11001	比较器 3 输出
11010-11110	保留
11111	无: 定时器自动计满返回 (FFFFh → 0000h)

# dsPIC33CK64MP105系列

寄存器 20-3: CCPxCON2L: CCPx控制2低位字寄存器

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
PWMRSEN	ASDGM	—	SSDG	—	—	—	—
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
ASDG7	ASDG6	ASDG5	ASDG4	ASDG3	ASDG2	ASDG1	ASDG0
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 15      **PWMRSEN:** CCPx PWM重启使能位  
1 = ASEVT位在关断输入结束后的下一个PWM周期开始时自动清零  
0 = ASEVT位必须由软件清零以恢复输出引脚上的PWM活动
- bit 14      **ASDGM:** CCPx自动关断门控模式使能位  
1 = 等待下一次时基复位或计满返回时进行关断  
0 = 立即发生关断事件
- bit 13      **未实现:** 读为0
- bit 12      **SSDG:** CCPx软件关断/门控位  
1 = 手动强制触发自动关断、定时器时钟门控或输入捕捉信号门控事件 (ASDGM位的设置仍适用)  
0 = 模块正常工作
- bit 11-8    **未实现:** 读为0
- bit 7-0     **ASDG[7:0]:** CCPx自动关断/门控源使能位  
1 = 使能ASDGx源n (有关自动关断/门控源, 请参见表20-6)  
0 = 禁止ASDGx源n

表20-6: 自动关断和门控源

ASDG[x]位	自动关断/门控源				
	SCCP1	SCCP2	SCCP3	SCCP4	MCCP5
0	比较器1输出				
1	比较器2输出				
2	OCFC				
3	OCFD				
4	ICM1 <sup>(1)</sup>	ICM2 <sup>(1)</sup>	ICM3 <sup>(1)</sup>	ICM4 <sup>(1)</sup>	ICM5 <sup>(1)</sup>
5	CLC1 <sup>(1)</sup>				
6	OCFA <sup>(1)</sup>				
7	OCFB <sup>(1)</sup>				

注 1: 通过外设引脚选择 (PPS) 进行选择。



**寄存器 20-4: CCPxCON2H: CCPx控制2高位字寄存器**

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
OENSYNC	—	OCFEN <sup>(1)</sup>	OCEEN <sup>(1)</sup>	OCDEN <sup>(1)</sup>	OCCEN <sup>(1)</sup>	OCBEN <sup>(1)</sup>	OCAEN
bit 15							bit 8
R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ICGSM1	ICGSM0	—	AUXOUT1	AUXOUT0	ICS2	ICS1	ICS0
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 15 **OENSYNC**: 输出使能同步位  
1 = 在下一时基复位或计满返回时通过输出使能位更新  
0 = 立即通过输出使能位更新
- bit 14 **未实现**: 读为0
- bit 13-8 **OC[F:A]EN**: 输出使能/转向控制位<sup>(1)</sup>  
1 = OCMx引脚由CCPx模块控制, 并且会产生输出比较或PWM信号  
0 = OCMx引脚不受CCPx模块控制; 该引脚可用于端口逻辑或引脚上复用的另一个外设
- bit 7-6 **ICGSM[1:0]**: 输入捕捉门控源模式控制位  
11 = 保留  
10 = 单触发模式: 门控源的下降沿会禁止未来的捕捉事件 (ICDIS = 1)  
01 = 单触发模式: 门控源的上升沿会使能未来的捕捉事件 (ICDIS = 0)  
00 = 电平敏感模式: 门控源的高电平将使能未来的捕捉事件; 低电平将禁止未来的捕捉事件
- bit 5 **未实现**: 读为0
- bit 4-3 **AUXOUT[1:0]**: 辅助输出信号事件选择位  
11 = 输入捕捉或输出比较事件; 定时器模式下无信号  
10 = 信号输出由模块工作模式定义 (见表20-4)  
01 = 时基计满返回事件 (所有模式)  
00 = 禁止
- bit 2-0 **ICS[2:0]**: 输入捕捉源选择位  
111 = CLC4输出  
110 = CLC3输出  
101 = CLC2输出  
100 = CLC1输出  
011 = 比较器3输出  
010 = 比较器2输出  
001 = 比较器1输出  
000 = 输入捕捉ICMx引脚 (PPS)

**注 1:** OCFEN至OCBEN (bit[13:9]) 仅在MCCP5模块中实现。

# dsPIC33CK64MP105系列

寄存器 20-5: CCPxCON3L: CCPx控制3低位字寄存器<sup>(1)</sup>

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

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	DT[5:0]					
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 15-6      未实现: 读为0

bit 5-0      **DT[5:0]:** CCPx死区选择位

111111 = 在互补输出信号间插入63个死区延时周期

111110 = 在互补输出信号间插入62个死区延时周期

...

000010 = 在互补输出信号间插入2个死区延时周期

000001 = 在互补输出信号间插入1个死区延时周期

000000 = 禁止死区逻辑

注 1: 该寄存器仅在MCCP9模块中实现。

寄存器 20-6: CCPxCON3H: CCPx控制3高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
OETRIG	OSCNT2	OSCNT1	OSCNT0	—	OUTM2 <sup>(1)</sup>	OUTM1 <sup>(1)</sup>	OUTM0 <sup>(1)</sup>
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	POLACE	POLBDF <sup>(1)</sup>	PSSACE1	PSSACE0	PSSBDF1 <sup>(1)</sup>	PSSBDF0 <sup>(1)</sup>
bit 7							bit 0

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15 **OETRIG**: CCPx死区选择位

1 = 对于触发模式 (TRIGEN = 1): 在触发之前, 模块不会驱动使能的输出引脚  
 0 = 输出引脚正常工作

bit 14-12 **OSCNT[2:0]**: 单触发事件计数位

111 = 将单触发事件延长7个时基周期 (总共8个时基周期)  
 110 = 将单触发事件延长6个时基周期 (总共7个时基周期)  
 101 = 将单触发事件延长5个时基周期 (总共6个时基周期)  
 100 = 将单触发事件延长4个时基周期 (总共5个时基周期)  
 011 = 将单触发事件延长3个时基周期 (总共4个时基周期)  
 010 = 将单触发事件延长2个时基周期 (总共3个时基周期)  
 001 = 将单触发事件延长1个时基周期 (总共2个时基周期)  
 000 = 不延长单触发事件

bit 11 **未实现**: 读为0bit 10-8 **OUTM[2:0]**: PWMx输出模式控制位<sup>(1)</sup>

111 = 保留  
 110 = 输出扫描模式  
 101 = 有刷直流输出模式 (正向)  
 100 = 有刷直流输出模式 (反向)  
 011 = 保留  
 010 = 半桥输出模式  
 001 = 推挽输出模式  
 000 = 可转向单输出模式

bit 7-6 **未实现**: 读为0bit 5 **POLACE**: CCPx输出引脚OCMxA、OCMxC和OCMxE极性控制位

1 = 输出引脚极性为低电平有效  
 0 = 输出引脚极性为高电平有效

bit 4 **POLBDF**: CCPx输出引脚OCMxB、OCMxD和OCMxF极性控制位<sup>(1)</sup>

1 = 输出引脚极性为低电平有效  
 0 = 输出引脚极性为高电平有效

bit 3-2 **PSSACE[1:0]**: PWMx输出引脚OCMxA、OCMxC和OCMxE关断状态控制位

11 = 当发生关断事件时, 将引脚驱动为有效状态  
 10 = 当发生关断事件时, 将引脚驱动为无效状态  
 0x = 当发生关断事件时, 引脚处于三态

bit 1-0 **PSSBDF[1:0]**: PWMx输出引脚OCMxB、OCMxD和OCMxF关断状态控制位<sup>(1)</sup>

11 = 当发生关断事件时, 将引脚驱动为有效状态  
 10 = 当发生关断事件时, 将引脚驱动为无效状态  
 0x = 当发生关断事件时, 引脚处于高阻态

注 1: 这些位仅在MCCP9模块中实现。

# dsPIC33CK64MP105系列

## 寄存器 20-7: CCPxSTATL: CCPx状态寄存器

U-0	U-0	U-0	U-0	U-0	W1-0	U-0	U-0
—	—	—	—	—	ICGARM	—	—
bit 15						bit 8	

R-0	W1-0	W1-0	R/C-0	R/C-0	R/C-0	R/C-0	R/C-0
CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE
bit 7							bit 0

图注:	C = 可清零位	U = 未实现位, 读为0
R = 可读位	W1 = 仅写1位	0 = 清零
-n = POR时的值	1 = 置1	x = 未知

- bit 15-11 **未实现:** 读为0
- bit 10 **ICGARM:** 输入捕捉门控就绪位  
向该位写入1将使输入捕捉门控逻辑为ICGSM[1:0] = 01或10时的单触发门控事件准备就绪。此位始终读为0。
- bit 9-8 **未实现:** 读为0
- bit 7 **CCPTRIG:** CCPx触发状态位  
1 = 定时器已触发并正在运行  
0 = 定时器未触发并保持复位
- bit 6 **TRSET:** CCPx触发设置请求位  
向该位写入1可在TRIGEN = 1时触发定时器（位始终读为0）。
- bit 5 **TRCLR:** CCPx触发清除请求位  
向此位写入1可在TRIGEN = 1时取消定时器触发（位始终读为0）。
- bit 4 **ASEVT:** CCPx自动关断事件状态/控制位  
1 = 关断事件正在进行; CCPx输出处于关断状态  
0 = CCPx输出正常工作
- bit 3 **SCEVT:** 单边沿比较事件状态位  
1 = 发生了单边沿比较事件  
0 = 未发生单边沿比较事件
- bit 2 **ICDIS:** 输入捕捉x禁止位  
1 = 输入捕捉x引脚（ICx）上发生事件时不会产生捕捉事件  
0 = 输入捕捉x引脚上发生事件时会产生捕捉事件
- bit 1 **ICOV:** 输入捕捉x缓冲区溢出状态位  
1 = 输入捕捉x FIFO缓冲区已溢出  
0 = 输入捕捉x FIFO缓冲区未溢出
- bit 0 **ICBNE:** 输入捕捉x缓冲区状态位  
1 = 输入捕捉x缓冲区中有数据  
0 = 输入捕捉x缓冲区为空

## 21.0 可配置逻辑单元 (CLC)

**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。更多信息, 请参见《dsPIC33/PIC24 系列参考手册》中的“可配置逻辑单元 (CLC)” ([www.microchip.com/DS70005298](http://www.microchip.com/DS70005298))。本数据手册中的信息取代FRM中的信息。

可配置逻辑单元 (CLC) 模块允许用户将信号的组合指定为逻辑功能的输入, 并使用逻辑输出来控制其他外设或 I/O 引脚。另外, CLC 模块不受软件执行限制并支持大量的输出设计, 这为嵌入式设计提供了更大的灵活性与更多的可能性。

选定的逻辑功能具有4个输入门。这4个输入门使用4个数据源选择多路开关从最多32个信号的信号池中选择输入信号。图21-1给出了该模块的概览。

图21-3给出了数据源多路开关的详细信息, 图21-2显示了逻辑输入门连接。

图21-1: CLCx模块

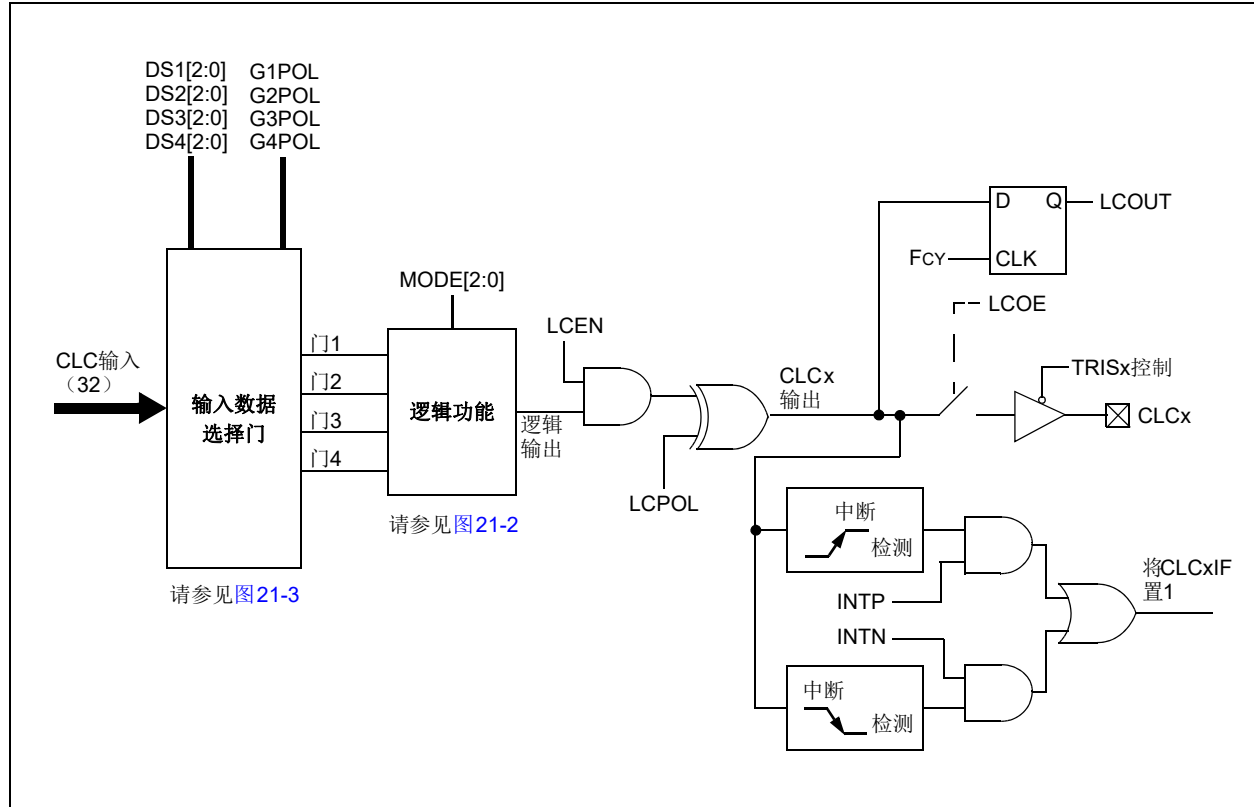


图21-2: CLCx逻辑功能组合选项

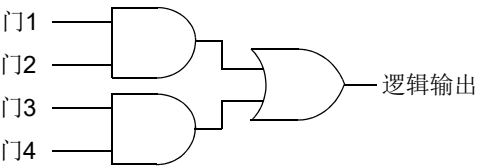
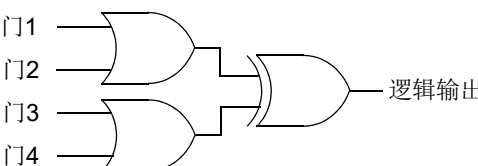
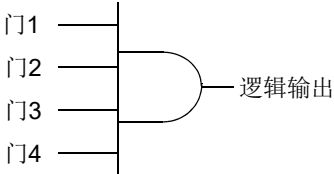
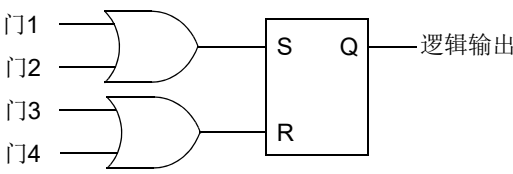
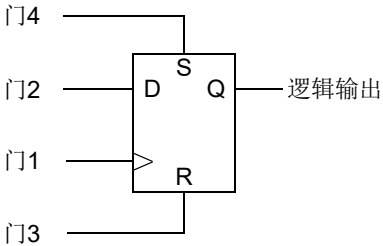
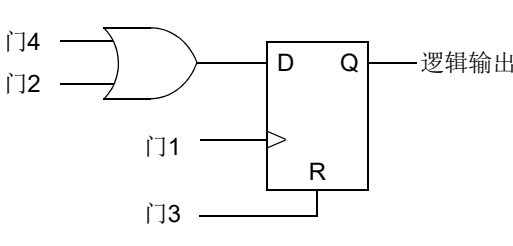
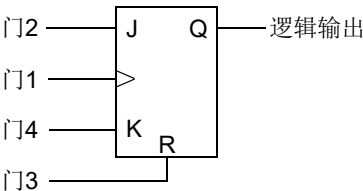
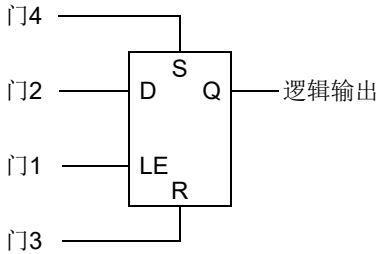
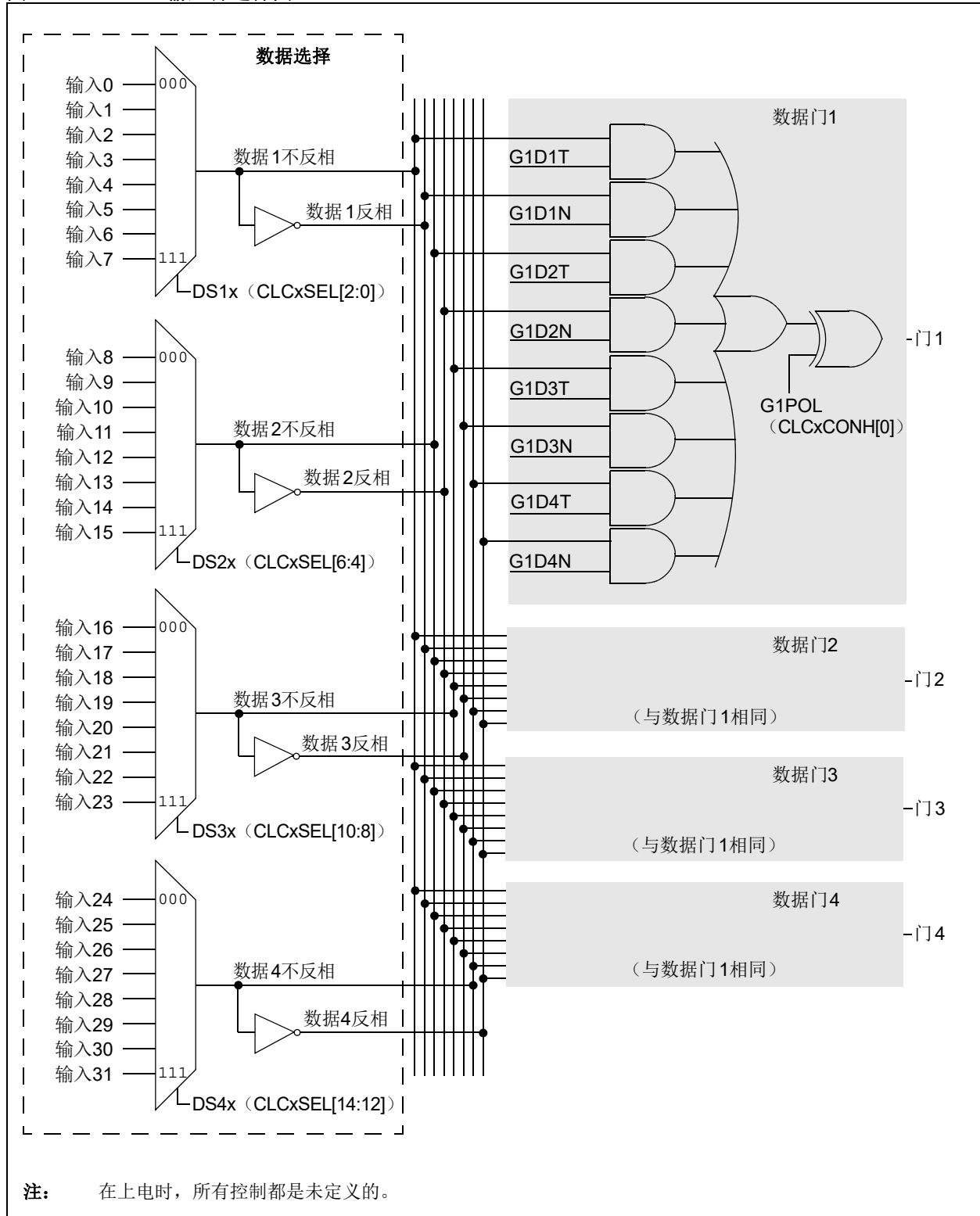
<p><b>AND - OR</b></p>  <p>MODE[2:0] = 000</p>	<p><b>OR - XOR</b></p>  <p>MODE[2:0] = 001</p>
<p><b>4输入AND</b></p>  <p>MODE[2:0] = 010</p>	<p><b>S-R锁存器</b></p>  <p>MODE[2:0] = 011</p>
<p><b>带置1和复位功能的1输入D触发器</b></p>  <p>MODE[2:0] = 100</p>	<p><b>带复位功能的2输入D触发器</b></p>  <p>MODE[2:0] = 101</p>
<p><b>带复位功能的J-K触发器</b></p>  <p>MODE[2:0] = 110</p>	<p><b>带置1和复位功能的1输入透明锁存器</b></p>  <p>MODE[2:0] = 111</p>

图21-3: CLCx输入源选择图



# dsPIC33CK64MP105系列

## 21.1 控制寄存器

CLCx模块由以下寄存器控制：

- CLCxCONL
- CLCxCONH
- CLCxSEL
- CLCxGLSL
- CLCxGLSH

CLCx控制寄存器（CLCxCONL和CLCxCONH）用于使能模块和允许中断、控制输出使能位、选择输出极性和选择逻辑功能。此外，CLCx控制寄存器不仅允许用户控制单元输出的逻辑极性，而且允许控制一些中间变量的逻辑极性。

CLCx输入多路开关选择寄存器（CLCxSEL）允许用户使用4个数据输入选择多路开关选择最多4个数据输入源。每个多路开关都有8个数据源可供选择。

CLCx门逻辑输入选择寄存器（CLCxGLSL和CLCxGLSH）允许用户选择将每个选择多路开关的哪些输出用作逻辑单元输入门的输入。每个数据源多路开关都会输出其输出的真值和取反形式。所有这8个信号均被使能，通过逻辑单元输入门进行逻辑或运算。如果未选择任何输入（CLCxGLS = 0x00），则输出将为0或1，具体取决于GxPOL位。

### 寄存器 21-1: CLCxCONL: CLCx控制寄存器（低位字）

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
LCEN	—	—	—	INTP	INTN	—	—
bit 15				bit 8			

R-0	R-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
LCOE	LCOUT	LCPOL	—	—	MODE2	MODE1	MODE0
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 15	<b>LCEN:</b> CLCx使能位 1 = 使能CLCx并混合输入信号 0 = 禁止CLCx并输出逻辑0
bit 14-12	<b>未实现:</b> 读为0
bit 11	<b>INTP:</b> CLCx正边沿中断允许位 1 = 当LCOUT上出现上升沿时，将产生中断 0 = 不产生中断
bit 10	<b>INTN:</b> CLCx负边沿中断允许位 1 = 当LCOUT上出现下降沿时，将产生中断 0 = 不产生中断
bit 9-8	<b>未实现:</b> 读为0
bit 7	<b>LCOE:</b> CLCx端口使能位 1 = 使能CLCx端口引脚输出 0 = 禁止CLCx端口引脚输出
bit 6	<b>LCOUT:</b> CLCx数据输出状态位 1 = CLCx输出高电平 0 = CLCx输出低电平
bit 5	<b>LCPOL:</b> CLCx输出极性控制位 1 = 模块的输出反相 0 = 模块的输出不反相
bit 4-3	<b>未实现:</b> 读为0



## 寄存器 21-1: CLCxCONL: CLCx控制寄存器 (低位字) (续)

bit 2-0 **MODE[2:0]:** CLCx模式位

- 111 = 带置1和复位功能的1输入透明锁存器
- 110 = 带复位功能的J-K触发器
- 101 = 带复位功能的2输入D触发器
- 100 = 带置1和复位功能的1输入D触发器
- 011 = SR锁存器
- 010 = 4输入AND门
- 001 = 4输入OR-XOR门
- 000 = 4输入AND-OR门

## 寄存器 21-2: CLCxCONH: CLCx控制寄存器 (高位字)

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

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	G4POL	G3POL	G2POL	G1POL
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 15-4 **未实现:** 读为0

bit 3 **G4POL:** 门4极性控制位

- 1 = 通道4逻辑输出在施加到逻辑单元时反相
- 0 = 通道4逻辑输出不反相

bit 2 **G3POL:** 门3极性控制位

- 1 = 通道3逻辑输出在施加到逻辑单元时反相
- 0 = 通道3逻辑输出不反相

bit 1 **G2POL:** 门2极性控制位

- 1 = 通道2逻辑输出在施加到逻辑单元时反相
- 0 = 通道2逻辑输出不反相

bit 0 **G1POL:** 门1极性控制位

- 1 = 通道1逻辑输出在施加到逻辑单元时反相
- 0 = 通道1逻辑输出不反相

# dsPIC33CK64MP105系列

寄存器 21-3: CLCxSEL: CLCx输入多路开关选择寄存器

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS4[2:0]			—	DS3[2:0]		
bit 15				bit 8			

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
—	DS2[2:0]			—	DS1[2:0]		
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 15 未实现: 读为0

bit 14-12 **DS4[2:0]:** 数据选择多路开关4信号选择位

111 = SCCP3辅助输出

110 = SCCP1辅助输出

101 = CLCIND引脚

100 = 保留

011 = SPI1输入 (SDIx) <sup>(1)</sup>

010 = 比较器3输出

001 = CLC2输出

000 = PWM事件A

bit 11 未实现: 读为0

bit 10-8 **DS3[2:0]:** 数据选择多路开关3信号选择位

111 = SCCP4比较事件标志 (CCP4IF)

110 = SCCP3比较事件标志 (CCP3IF)

101 = CLC4输出

100 = 对应于CLCx模块的UART1 RX输出

011 = 对应于CLCx模块的SPI1输出 (SDOx) <sup>(1)</sup>

010 = 比较器2输出

001 = CLC1输出

000 = CLCINC I/O引脚

bit 7 未实现: 读为0

bit 6-4 **DS2[2:0]:** 数据选择多路开关2信号选择位

111 = SCCP2 OC (CCP2IF) 输出

110 = SCCP1 OC (CCP1IF) 输出

101 = 保留

100 = 保留

011 = 对应于CLCx模块的UART1 TX输入

010 = 比较器1输出

001 = 保留

000 = CLCINB I/O引脚

bit 3 未实现: 读为0

**注 1:** 仅在PPS引脚上使用SPI时有效。

## 寄存器 21-3: CLCxSEL: CLCx输入多路开关选择寄存器 (续)

bit 2-0      **DS1[2:0]:** 数据选择多路开关1信号选择位

111 = SCCP4 辅助输出

110 = SCCP2 辅助输出

101 = 保留

100 = REFCLKO 输出

011 = INTRC/LPRC 时钟源

010 = CLC3 输出

001 = 系统时钟 (FcY)

000 = CLCINA I/O 引脚

**注 1:** 仅在PPS引脚上使用SPI时有效。

# dsPIC33CK64MP105系列

**寄存器 21-4: CLCxGLSL: CLCx门逻辑输入选择低位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G2D4T	G2D4N	G2D3T	G2D3N	G2D2T	G2D2N	G2D1T	G2D1N
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
G1D4T	G1D4N	G1D3T	G1D3N	G1D2T	G1D2N	G1D1T	G1D1N
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 15      **G2D4T:** 门2数据源4真值使能位  
1 = 对门2使能数据源4信号  
0 = 对门2禁止数据源4信号

bit 14      **G2D4N:** 门2数据源4取反使能位  
1 = 对门2使能数据源4反相信号  
0 = 对门2禁止数据源4反相信号

bit 13      **G2D3T:** 门2数据源3真值使能位  
1 = 对门2使能数据源3信号  
0 = 对门2禁止数据源3信号

bit 12      **G2D3N:** 门2数据源3取反使能位  
1 = 对门2使能数据源3反相信号  
0 = 对门2禁止数据源3反相信号

bit 11      **G2D2T:** 门2数据源2真值使能位  
1 = 对门2使能数据源2信号  
0 = 对门2禁止数据源2信号

bit 10      **G2D2N:** 门2数据源2取反使能位  
1 = 对门2使能数据源2反相信号  
0 = 对门2禁止数据源2反相信号

bit 9        **G2D1T:** 门2数据源1真值使能位  
1 = 对门2使能数据源1信号  
0 = 对门2禁止数据源1信号

bit 8        **G2D1N:** 门2数据源1取反使能位  
1 = 对门2使能数据源1反相信号  
0 = 对门2禁止数据源1反相信号

bit 7        **G1D4T:** 门1数据源4真值使能位  
1 = 对门1使能数据源4信号  
0 = 对门1禁止数据源4信号

bit 6        **G1D4N:** 门1数据源4取反使能位  
1 = 对门1使能数据源4反相信号  
0 = 对门1禁止数据源4反相信号

bit 5        **G1D3T:** 门1数据源3真值使能位  
1 = 对门1使能数据源3信号  
0 = 对门1禁止数据源3信号

bit 4        **G1D3N:** 门1数据源3取反使能位  
1 = 对门1使能数据源3反相信号  
0 = 对门1禁止数据源3反相信号

**注 1:** 如果不选择输入 (CLCxGLS = 0x00), 输出将为 0 或 1, 具体取决于 GxPOL 位。

## 寄存器 21-4: CLCxGLSL: CLCx门逻辑输入选择低位字寄存器 (续)

bit 3	<b>G1D2T:</b> 门1数据源2真值使能位 1 = 对门1使能数据源2信号 0 = 对门1禁止数据源2信号
bit 2	<b>G1D2N:</b> 门1数据源2取反使能位 1 = 对门1使能数据源2反相信号 0 = 对门1禁止数据源2反相信号
bit 1	<b>G1D1T:</b> 门1数据源1真值使能位 1 = 对门1使能数据源1信号 0 = 对门1禁止数据源1信号
bit 0	<b>G1D1N:</b> 门1数据源1取反使能位 1 = 对门1使能数据源1反相信号 0 = 对门1禁止数据源1反相信号

**注 1:** 如果不选择输入 (CLCxGLS = 0x00), 输出将为 0 或 1, 具体取决于 GxPOL 位。

# dsPIC33CK64MP105系列

**寄存器 21-5: CLCxGLSH: CLCx门逻辑输入选择高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
G4D4T	G4D4N	G4D3T	G4D3N	G4D2T	G4D2N	G4D1T	G4D1N
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
G3D4T	G3D4N	G3D3T	G3D3N	G3D2T	G3D2N	G3D1T	G3D1N
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 15 **G4D4T:** 门4数据源4真值使能位

1 = 对门4使能数据源4信号  
0 = 对门4禁止数据源4信号

bit 14 **G4D4N:** 门4数据源4取反使能位

1 = 对门4使能数据源4反相信号  
0 = 对门4禁止数据源4反相信号

bit 13 **G4D3T:** 门4数据源3真值使能位

1 = 对门4使能数据源3信号  
0 = 对门4禁止数据源3信号

bit 12 **G4D3N:** 门4数据源3取反使能位

1 = 对门4使能数据源3反相信号  
0 = 对门4禁止数据源3反相信号

bit 11 **G4D2T:** 门4数据源2真值使能位

1 = 对门4使能数据源2信号  
0 = 对门4禁止数据源2信号

bit 10 **G4D2N:** 门4数据源2取反使能位

1 = 对门4使能数据源2反相信号  
0 = 对门4禁止数据源2反相信号

bit 9 **G4D1T:** 门4数据源1真值使能位

1 = 对门4使能数据源1信号  
0 = 对门4禁止数据源1信号

bit 8 **G4D1N:** 门4数据源1取反使能位

1 = 对门4使能数据源1反相信号  
0 = 对门4禁止数据源1反相信号

bit 7 **G3D4T:** 门3数据源4真值使能位

1 = 对门3使能数据源4信号  
0 = 对门3禁止数据源4信号

bit 6 **G3D4N:** 门3数据源4取反使能位

1 = 对门3使能数据源4反相信号  
0 = 对门3禁止数据源4反相信号

bit 5 **G3D3T:** 门3数据源3真值使能位

1 = 对门3使能数据源3信号  
0 = 对门3禁止数据源3信号

bit 4 **G3D3N:** 门3数据源3取反使能位

1 = 对门3使能数据源3反相信号  
0 = 对门3禁止数据源3反相信号

**注 1:** 如果不选择输入 (CLCxGLS = 0x00), 输出将为0或1, 具体取决于 GxPOL 位。

## 寄存器 21-5: CLCxGLSH: CLCx门逻辑输入选择高位字寄存器 (续)

bit 3	<b>G3D2T:</b> 门3数据源2真值使能位 1 = 对门3使能数据源2信号 0 = 对门3禁止数据源2信号
bit 2	<b>G3D2N:</b> 门3数据源2取反使能位 1 = 对门3使能数据源2反相信号 0 = 对门3禁止数据源2反相信号
bit 1	<b>G3D1T:</b> 门3数据源1真值使能位 1 = 对门3使能数据源1信号 0 = 对门3禁止数据源1信号
bit 0	<b>G3D1N:</b> 门3数据源1取反使能位 1 = 对门3使能数据源1反相信号 0 = 对门3禁止数据源1反相信号

**注 1:** 如果不选择输入 (CLCxGLS = 0x00), 输出将为 0 或 1, 具体取决于 GxPOL 位。

注:



## 22.0 外设触发信号发生器（PTG）

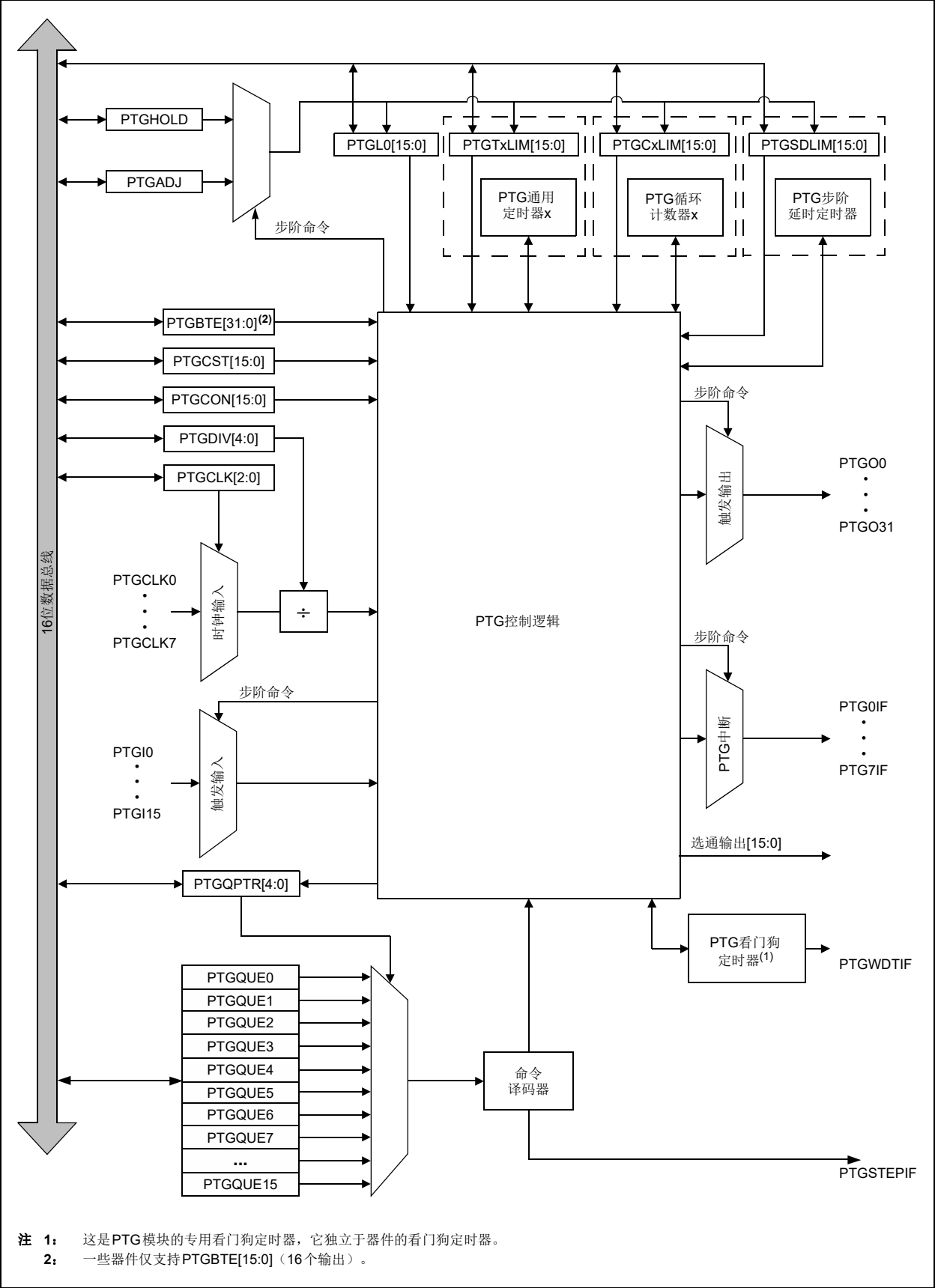
**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“**外设触发信号发生器（PTG）**”（[www.microchip.com/DS70000669](http://www.microchip.com/DS70000669)）。

dsPIC33CK64MP105系列外设触发信号发生器（PTG）模块是一个用户可编程的序列发生器，它能够生成复杂的触发信号序列来协调其他外设的操作。PTG模块设计为可以与诸如模数转换器（ADC）、输出比较和PWM模块、定时器以及中断控制器等模块接口。

## 22.1 特性

- 行为通过步阶命令来驱动：
  - 步阶命令的宽度为8位
- 命令存储在一个步阶队列中：
  - 队列深度最多32
  - 可编程步阶执行时间（步阶延时）
- 支持命令序列循环：
  - 可以嵌套1级深
  - 有条件或无条件循环
  - 两个16位循环计数器
- 15个硬件输入触发信号
  - 正边沿或负边沿敏感，或者高电平或低电平敏感
- 1个软件输入触发信号
- 最多可生成32个不同的输出触发信号
- 可生成两种类型的触发输出：
  - 独立
  - 广播
- 最多可生成10个不同的中断信号
- 两个16位通用定时器
- 灵活的独立看门狗定时器（WDT），用于设置触发等待时间的上限
- 调试模式下支持单步命令
- 可选时钟（系统、脉宽调制器（PWM）或ADC）
- 可编程时钟分频比

图22-1: PTG框图



## 22.2 PTG控制/状态寄存器

寄存器 22-1: PTGCST: PTG控制/状态低位字寄存器

R/W-0	U-0	R/W-0	R/W-0	U-0	HC/R/W-0	R/W-0	R/W-0
PTGEN	—	PTGSIDL	PTGTOGL	—	PTGSWT <sup>(2)</sup>	PTGSSEN <sup>(3)</sup>	PTGIVIS
bit 15				bit 8			

HC/R/W-0	HS/R/W-0	HS/HC/R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
PTGSTRT	PTGWDTO	PTGBUSY	—	—	—	PTGITM1 <sup>(1)</sup>	PTGITM0 <sup>(1)</sup>
bit 7				bit 0			

图注:	HC = 硬件清零位	HS = 硬件置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 15 **PTGEN:** PTG使能位

1 = 使能PTG

0 = 禁止PTG

bit 14 **未实现:** 读为0

bit 13 **PTGSIDL:** PTG调试模式冻结位

1 = 器件空闲时, PTG暂停工作

0 = 器件空闲时, PTG继续工作

bit 12 **PTGTOGL:** PTG触发输出翻转位

1 = 每次执行PTGTRIG后翻转触发输出的状态

0 = 每次执行PTGTRIG后生成一个触发脉冲

bit 11 **未实现:** 读为0

bit 10 **PTGSWT:** PTG软件触发位<sup>(2)</sup>

1 = 每次执行PTGTRIG后翻转触发输出的状态

0 = 每次执行PTGTRIG后生成一个触发脉冲

bit 9 **PTGSSEN:** PTG单步命令位<sup>(3)</sup>

1 = 在调试模式下使能单步

0 = 禁止单步

bit 8 **PTGIVIS:** PTG计数器/定时器可视性位

1 = 读取PTGSDLIM、PTGCxLIM或PTGTxLIM寄存器时返回它们相应的计数器/定时器寄存器 (PTGSDLIM、PTGCxLIM和PTGTxLIM) 的当前值

0 = 读取PTGSDLIM、PTGCxLIM或PTGTxLIM寄存器时返回这些限制寄存器的值

bit 7 **PTGSTRT:** 启动PTG序列发生器位

1 = 开始按顺序执行命令 (连续模式)

0 = 停止执行命令

bit 6 **PTGWDTO:** PTG看门狗定时器超时状态位

1 = PTG看门狗定时器已超时

0 = PTG看门狗定时器未超时

bit 5 **PTGBUSY:** PTG状态机忙位

1 = PTG基于所选的时钟源运行; 不允许对PTGCLK[2:0]和PTGDIV[4:0]执行SFR写操作

0 = PTG状态机未运行

注 1: 这些位仅适用于PTGWHI和PTGWLO命令。

2: 该位仅用于PTGCTRL步阶命令软件触发选项。

3: 只能在调试模式下写入PTGSSEN位。

# dsPIC33CK64MP105系列

---

## 寄存器 22-1: PTCST: PTG控制/状态低位字寄存器 (续)

bit 4-2 未实现: 读为0

bit 1-0 **PTGITM[1:0]**: PTG输入触发操作选择位<sup>(1)</sup>

11 =在退出命令时不执行带步阶延时的单电平检测 (无论PTGCTRL命令如何) (模式3)

10 =在退出命令时执行带步阶延时的单电平检测 (模式2)

01 =在退出命令时不执行带步阶延时的连续边沿检测 (无论PTGCTRL命令如何) (模式1)

00 =在退出命令时执行带步阶延时的连续边沿检测 (模式0)

**注 1:** 这些位仅适用于PTGWHI和PTGWLO命令。

**2:** 该位仅用于PTGCTRL步阶命令软件触发选项。

**3:** 只能在调试模式下写入PTGSSEN位。

**寄存器 22-2: PTGCON: PTG控制/状态高位字寄存器**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGCLK2	PTGCLK1	PTGCLK0	PTGDIV4	PTGDIV3	PTGDIV2	PTGDIV1	PTGDIV0
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTGPWD3	PTGPWD2	PTGPWD1	PTGPWD0	—	PTGWDT2	PTGWDT1	PTGWDT0
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 15-13 **PTGCLK[2:0]:** PTG模块时钟源选择位  
 111 = CLC1  
 110 = PLL VCO 4分频输出  
 101 = 保留  
 100 = 保留  
 011 = 来自Timer1时钟引脚T1CK的输入  
 010 = PTG模块时钟源将为ADC时钟  
 001 = PTG模块时钟源将为Fosc  
 000 = PTG模块时钟源将为Fosc/2 (Fp)

bit 12-8 **PTGDIV[4:0]:** PTG模块时钟预分频比位  
 11111 = 32分频  
 11110 = 31分频  
 ...  
 00001 = 2分频  
 00000 = 1分频

bit 7-4 **PTGPWD[3:0]:** PTG触发输出脉冲宽度(用PTG时钟周期数表示)位  
 1111 = 所有触发输出的宽度为16个PTG时钟周期  
 1110 = 所有触发输出的宽度为15个PTG时钟周期  
 ...  
 0001 = 所有触发输出的宽度为2个PTG时钟周期  
 0000 = 所有触发输出的宽度为1个PTG时钟周期

bit 3 **未实现:** 读为0

bit 2-0 **PTGWDT[2:0]:** PTG看门狗定时器超时选择位  
 111 = 看门狗定时器将在512个PTG时钟后超时  
 110 = 看门狗定时器将在256个PTG时钟后超时  
 101 = 看门狗定时器将在128个PTG时钟后超时  
 100 = 看门狗定时器将在64个PTG时钟后超时  
 011 = 看门狗定时器将在32个PTG时钟后超时  
 010 = 看门狗定时器将在16个PTG时钟后超时  
 001 = 看门狗定时器将在8个PTG时钟后超时  
 000 = 禁止看门狗定时器

# dsPIC33CK64MP105系列

寄存器 22-3: PTGBTE: PTG广播触发使能低位字寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[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
PTGBTE[7:0]							
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 15-0 **PTGBTE[15:0]:** PTG广播触发使能位

1 = 执行广播命令时生成触发信号

0 = 执行广播命令时不生成触发信号

注 1: 当模块执行步阶命令时, 这些位是只读的。

寄存器 22-4: PTGBTEH: PTG广播触发使能高位字寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGBTE[31:24]							
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
PTGBTE[23:16]							
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 15-0 **PTGBTE[31:16]:** PTG广播触发使能位

1 = 执行广播命令时生成触发信号

0 = 执行广播命令时不生成触发信号

注 1: 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-5: PTGHOLD: PTG保持寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGHOLD[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
PTGHOLD[7:0]							
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 15-0 **PTGHOLD[15:0]:** PTG通用保持寄存器位

该寄存器用于保存用户提供的、要使用PTGCOPY命令复制到PTGTxLIM、PTGCxLIM、PTGSDLIM或PTGL0寄存器的数据。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-6: PTGT0LIM: PTG TIMER0限制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT0LIM[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
PTGT0LIM[7:0]							
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 15-0 **PTGT0LIM[15:0]:** PTG Timer0限制寄存器位

通用Timer0限制寄存器。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

# dsPIC33CK64MP105系列

## 寄存器 22-7: PTGT1LIM: PTG TIMER1限制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGT1LIM[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
PTGT1LIM[7:0]							
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 15-0 **PTGT1LIM[15:0]:** PTG Timer1 限制寄存器位  
通用Timer1限制寄存器。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-8: PTGSDLIM: PTG步阶延时限制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGSDLIM[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
PTGSDLIM[7:0]							
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 15-0 **PTGSDLIM[15:0]:** PTG步阶延时限制寄存器位  
该寄存器用于保存一个PTG步阶延时值, 该值表示从步阶命令启动到步阶命令完成之间的额外PTG时钟数。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。



## 寄存器 22-9: PTGC0LIM: PTG计数器0限制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC0LIM[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
PTGC0LIM[7:0]							
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 15-0 **PTGC0LIM[15:0]:** PTG计数器0限制寄存器位

该寄存器用于指定PTGJMPC0步阶命令的循环计数, 或用作通用计数器0的限制寄存器。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-10: PTGC1LIM: PTG计数器1限制寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGC1LIM[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
PTGC1LIM[7:0]							
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 15-0 **PTGC1LIM[15:0]:** PTG计数器1限制寄存器位

该寄存器用于指定PTGJMPC1步阶命令的循环计数, 或用作通用计数器1的限制寄存器。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

# dsPIC33CK64MP105系列

## 寄存器 22-11: PTGADJ: PTG调节寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGADJ[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
PTGADJ[7:0]							
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 15-0 **PTGADJ[15:0]: PTG调节寄存器位**

该寄存器用于保存用户提供的、要使用PTGADD命令加到PTGTxLIM、PTGCxLIM、PTGSDLIM或PTGL0寄存器的数据。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-12: PTGL0: PTG立即数0寄存器<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTGL0[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
PTGL0[7:0]							
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 15-0 **PTGL0[15:0]: PTG立即数0寄存器位**

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-13: PTGQPTR: PTG步阶队列指针寄存器<sup>(1)</sup>

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PTGQPTR[4:0]				
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 15-5      **未实现:** 读为0

bit 4-0      **PTGQPTR[4:0]:** PTG 步阶队列指针寄存器位  
该寄存器指向步阶队列中当前处于活动状态的步阶命令。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

## 寄存器 22-14: PTGQUEn: PTG步阶队列n指针寄存器 (n = 0-15)<sup>(1)</sup>

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP2n+1[7:0] <sup>(2)</sup>							
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
STEP2n[7:0] <sup>(2)</sup>							
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 15-8      **STEP2n+1[7:0]:** PTG 命令 4n+1 位<sup>(2)</sup>  
用于存储STEP2n+1命令字节的队列位置, 其中“n”来自PTGQUEn。

bit 7-0      **STEP2n[7:0]:** PTG 命令 4n+2 位<sup>(2)</sup>  
用于存储STEP2n命令字节的队列位置, 其中“n”是奇编号步阶队列指针。

**注 1:** 当模块执行步阶命令时, 这些位是只读的。

**2:** 有关步阶命令编码的信息, 请参见表 22-1。

# dsPIC33CK64MP105系列

表 22-1: P TG步阶命令格式和说明

步阶命令字节			
STEPx[7:0]			
CMD[3:0]		OPTION[3:0]	
bit 7	bit 4	bit 3	bit 0

bit 7-4	步阶命令	CMD[3:0]	命令说明
	PTGCTRL	0000	根据 OPTION[3:0] 位的描述执行控制命令。
	PTGADD	0001	根据 OPTION[3:0] 位的描述，将 PTGADJ 寄存器的内容加到目标寄存器中。
	PTGCOPY		根据 OPTION[3:0] 位的描述，将 PTGHOLD 寄存器的内容复制到目标寄存器中。
	PTGSTRB	001x	该命令启动 CMD[0] 和 OPTION[3:0] 位中指定通道的 ADC 转换。
	PTGWHI	0100	根据 OPTION[3:0] 位的描述，等待来自选定 P TG 触发输入的从低电平变为高电平的边沿输入。
	PTGWLO	0101	根据 OPTION[3:0] 位的描述，等待来自选定 P TG 触发输入的从高电平变为低电平的边沿输入。
	—	0110	保留；不要使用。(1)
	PTGIRQ	0111	根据 OPTION[3:0] 位的描述，产生独立的中断请求。
	PTGTRIG	100x	根据 CMD[0]:OPTION[3:0] 位的描述，产生独立的触发输出。
	PTGJMP	101x	将 CMD[0]:OPTION[3:0] 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到相应的步阶队列。
	PTGJMPC0	110x	PTGC0 = PTGC0LIM: 递增 PTGQPTR 寄存器。
			PTGC0 ≠ PTGC0LIM: 递增计数器 0 (PTGC0) 并将 CMD[0]:OPTION[3:0] 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到相应的步阶队列。
	PTGJMPC1	111x	PTGC1 = PTGC1LIM: 递增 PTGQPTR 寄存器。
			PTGC1 ≠ PTGC1LIM: 递增计数器 1 (PTGC1) 并将 CMD[0]:OPTION[3:0] 位中包含的值复制到 PTGQPTR 寄存器，然后跳转到相应的步阶队列。

注 1: 所有保留的命令或选项都将执行，但它们不起任何作用（即，作为一条 NOP 指令执行）。

表22-2: PTG命令选项

bit 3-0	步阶命令	OPTION[3:0]	命令说明
	PTGCTRL <sup>(1)</sup>	0000	NOP。
		0001	保留；不要使用。
		0010	禁止步阶延时定时器（PTGSD）。
		0011	保留；不要使用。
		0100	保留；不要使用。
		0101	保留；不要使用。
		0110	使能步阶延时定时器（PTGSD）。
		0111	保留；不要使用。
		1000	启动并等待PTG Timer0与PTGT0LIM寄存器匹配。
		1001	启动并等待PTG Timer1与PTGT1LIM寄存器匹配。
		1010	等待软件触发（电平，PTGSWT = 1）。
		1011	等待软件触发（正边沿，PTGSWT = 0至1）。
		1100	将PTGC0LIM寄存器的内容复制到选通输出。
		1101	将PTGC1LIM寄存器的内容复制到选通输出。
		1110	保留；不要使用。
		1111	生成PTGBTE寄存器中指示的触发信号。
	PTGADD <sup>(1)</sup>	0000	将PTGADJ寄存器的内容加到PTGC0LIM寄存器。
		0001	将PTGADJ寄存器的内容加到PTGC1LIM寄存器。
		0010	将PTGADJ寄存器的内容加到PTGT0LIM寄存器。
		0011	将PTGADJ寄存器的内容加到PTGT1LIM寄存器。
		0100	将PTGADJ寄存器的内容加到PTGSDLIM寄存器。
		0101	将PTGADJ寄存器的内容加到PTGL0寄存器。
		0110	保留；不要使用。
		0111	保留；不要使用。
	PTGCOPY <sup>(1)</sup>	1000	将PTGHOLD寄存器的内容复制到PTGC0LIM寄存器。
		1001	将PTGHOLD寄存器的内容复制到PTGC1LIM寄存器。
		1010	将PTGHOLD寄存器的内容复制到PTGT0LIM寄存器。
		1011	将PTGHOLD寄存器的内容复制到PTGT1LIM寄存器。
		1100	将PTGHOLD寄存器的内容复制到PTGSDLIM寄存器。
		1101	将PTGHOLD寄存器的内容复制到PTGL0寄存器。
		1110	保留；不要使用。
		1111	保留；不要使用。

注 1: 所有保留的命令或选项都将执行，但它们不起任何作用（即，作为一条NOP指令执行）。

# dsPIC33CK64MP105系列

表22-2: PTG命令选项 (续)

bit 3-0	步阶命令	OPTION[3:0]	选项说明
	PTGWHI <sup>(1)</sup> 或 PTGWLO <sup>(1)</sup>	0000	PTGI0 (关于输入分配, 见表22-3)。
		•	•
		•	•
		•	•
	PTGIRQ <sup>(1)</sup>	1111	PTGI15 (关于输入分配, 见表22-3)。
		0000	产生PTG中断0。
		•	•
		•	•
		•	•
		0111	产生PTG中断7。
		1000	保留; 不要使用。
		•	•
		•	•
		•	•
	PTGTRIG	1111	保留; 不要使用。
		00000	PTGO0 (关于输出分配, 见表22-4)。
		00001	PTGO1 (关于输出分配, 见表22-4)。
		•	•
		•	•
	PTGWHI <sup>(1)</sup> 或 PTGWLO <sup>(1)</sup>	•	•
		•	•
		•	•
	PTGIRQ <sup>(1)</sup>	1111	PTGI15 (关于输入分配, 见表22-3)。
		0000	产生PTG中断0。
		•	•
		•	•
		•	•
		0111	产生PTG中断7。
		1000	保留; 不要使用。
		•	•
		•	•
		•	•
	PTGTRIG	1111	保留; 不要使用。
		00000	PTGO0 (关于输出分配, 见表22-4)。
		00001	PTGO1 (关于输出分配, 见表22-4)。

注 1: 所有保留的命令或选项都将执行, 但它们不起任何作用 (即, 作为一条NOP指令执行)。

表22-3: PTG输入说明

PTG输入编号	PTG输入说明
PTG触发输入0	来自PWM通道1的触发输入
PTG触发输入1	来自PWM通道2的触发输入
PTG触发输入2	来自PWM通道3的触发输入
PTG触发输入3	来自PWM通道4的触发输入
PTG触发输入4	保留
PTG触发输入5	保留
PTG触发输入6	保留
PTG触发输入7	来自SCCP4的触发输入
PTG触发输入8	来自MCCP5的触发输入
PTG触发输入9	来自比较器1的触发输入
PTG触发输入10	来自比较器2的触发输入
PTG触发输入11	来自比较器3的触发输入
PTG触发输入12	来自CLC1的触发输入
PTG触发输入13	来自ADC通用中断的触发输入
PTG触发输入14	保留
PTG触发输入15	来自INT2 PPS的触发输入

表22-4: PTG输出说明

PTG输出编号	PTG输出说明
PTGO0至PTGO11	保留
PTGO12	ADC TRGSRC[30]
PTGO13至PTGO23	保留
PTGO24	PPS输出RP46
PTGO25	PPS输出RP47
PTGO26	PPS输入RP6
PTGO27	PPS输入RP7
PTGO28至PTGO31	保留

注:



## 23.0 偏流发生器 (CBG)

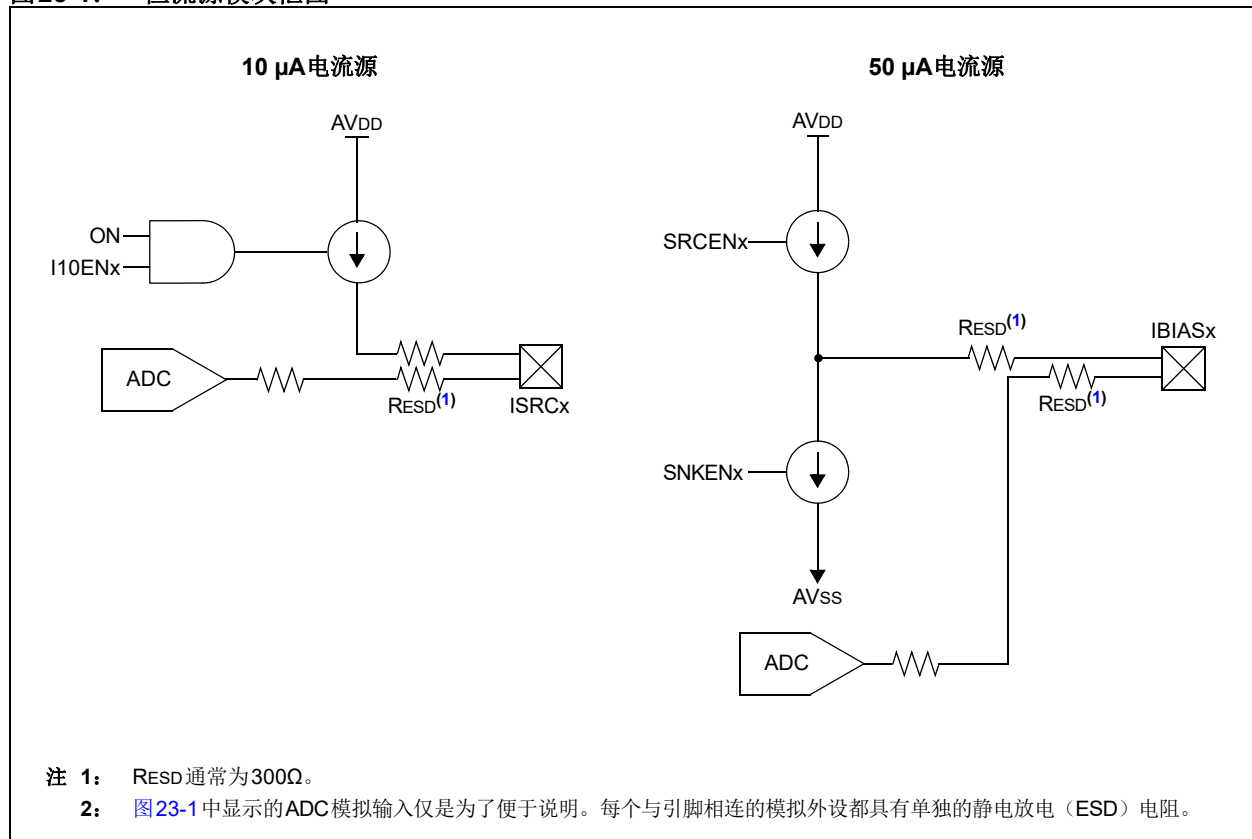
- 注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的“偏流发生器 (CBG)” ([www.microchip.com/DS70005253](http://www.microchip.com/DS70005253))。
- 2:** 本章中描述的一些寄存器及其相关的位并非在所有器件上都可用。关于具体器件的寄存器和位信息，请参见本数据手册中的第4.0节“存储器构成”。

偏流发生器 (Current Bias Generator, CBG) 由两类电流源组成：10  $\mu\text{A}$  和 50  $\mu\text{A}$  电流源。这两类电流源的主要特性分别如下：

- 10  $\mu\text{A}$  电流源：
  - 仅拉电流
  - 最多4个独立电流源
- 50  $\mu\text{A}$  电流源：
  - 可选拉电流或灌电流
  - 可选拉灌电流镜像

图23-1 给出了CBG模块的简化框图。

图23-1: 恒流源模块框图<sup>(2)</sup>



# dsPIC33CK64MP105系列

## 23.1 偏流发生器控制寄存器

寄存器 23-1: **BIASCON**: 偏流发生器控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ON	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	I10EN3	I10EN2	I10EN1	I10EN0
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 15 **ON**: 偏流模块使能位

1 = 使能模块

0 = 禁止模块

bit 14-4 **未实现**: 读为0

bit 3 **I10EN3**: 10  $\mu$ A 输出3使能位

1 = 使能 10  $\mu$ A 输出

0 = 禁止 10  $\mu$ A 输出

bit 2 **I10EN2**: 10  $\mu$ A 输出2使能位

1 = 使能 10  $\mu$ A 输出

0 = 禁止 10  $\mu$ A 输出

bit 1 **I10EN1**: 10  $\mu$ A 输出1使能位

1 = 使能 10  $\mu$ A 输出

0 = 禁止 10  $\mu$ A 输出

bit 0 **I10EN0**: 10  $\mu$ A 输出0使能位

1 = 使能 10  $\mu$ A 输出

0 = 禁止 10  $\mu$ A 输出

## 寄存器 23-2: IBIASCONH: 偏流发生器 50 $\mu$ A 电流源控制高位字寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN3	SHRSNKEN3	GENSRCEN3	GENSNKEN3	SRCEN3	SNKEN3
bit 15						bit 8	

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN2	SHRSNKEN2	GENSRCEN2	GENSNKEN2	SRCEN2	SNKEN2
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 15-14 未实现: 读为0

bit 13 **SHRSRCEN3**: 输出3共用拉电流使能位  
1 = 使能拉电流镜像模式 (使用来自其他源的参考)  
0 = 禁止拉电流镜像模式

bit 12 **SHRSNKEN3**: 输出3共用灌电流使能位  
1 = 使能灌电流镜像模式 (使用来自其他源的参考)  
0 = 禁止灌电流镜像模式

bit 11 **GENSRCEN3**: 输出3拉电流产生使能位  
1 = 电流源产生拉电流镜像参考  
0 = 电流源不产生拉电流镜像参考

bit 10 **GENSNKEN3**: 输出3灌电流产生使能位  
1 = 电流源产生灌电流镜像参考  
0 = 电流源不产生灌电流镜像参考

bit 9 **SRCEN3**: 输出3拉电流使能位  
1 = 使能拉电流  
0 = 禁止拉电流

bit 8 **SNKEN3**: 输出3灌电流使能位  
1 = 使能灌电流  
0 = 禁止灌电流

bit 7-6 未实现: 读为0

bit 5 **SHRSRCEN2**: 输出2共用拉电流使能位  
1 = 使能拉电流镜像模式 (使用来自其他源的参考)  
0 = 禁止拉电流镜像模式

bit 4 **SHRSNKEN2**: 输出2共用灌电流使能位  
1 = 使能灌电流镜像模式 (使用来自其他源的参考)  
0 = 禁止灌电流镜像模式

bit 3 **GENSRCEN2**: 输出2拉电流产生使能位  
1 = 电流源产生拉电流镜像参考  
0 = 电流源不产生拉电流镜像参考

bit 2 **GENSNKEN2**: 输出2灌电流产生使能位  
1 = 电流源产生灌电流镜像参考  
0 = 电流源不产生灌电流镜像参考

bit 1 **SRCEN2**: 输出2拉电流使能位  
1 = 使能拉电流  
0 = 禁止拉电流

bit 0 **SNKEN2**: 输出2灌电流使能位  
1 = 使能灌电流  
0 = 禁止灌电流

# dsPIC33CK64MP105系列

**寄存器 23-3: IBIASCONL: 偏流发生器 50  $\mu$ A 电流源控制低位字寄存器**

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN1	SHRSNKEN1	GENSRCEN1	GENSNKEN1	SRCEN1	SNKEN1
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	SHRSRCEN0	SHRSNKEN0	GENSRCEN0	GENSNKEN0	SRCEN0	SNKEN0
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 15-14 **未实现:** 读为0

bit 13 **SHRSRCEN1:** 输出1共用拉电流使能位  
1 = 使能拉电流镜像模式 (使用来自其他源的参考)  
0 = 禁止拉电流镜像模式

bit 12 **SHRSNKEN1:** 输出1共用灌电流使能位  
1 = 使能灌电流镜像模式 (使用来自其他源的参考)  
0 = 禁止灌电流镜像模式

bit 11 **GENSRCEN1:** 输出1拉电流产生使能位  
1 = 电流源产生拉电流镜像参考  
0 = 电流源不产生拉电流镜像参考

bit 10 **GENSNKEN1:** 输出1灌电流产生使能位  
1 = 电流源产生灌电流镜像参考  
0 = 电流源不产生灌电流镜像参考

bit 9 **SRCEN1:** 输出1拉电流使能位  
1 = 使能拉电流  
0 = 禁止拉电流

bit 8 **SNKEN1:** 输出1灌电流使能位  
1 = 使能灌电流  
0 = 禁止灌电流

bit 7-6 **未实现:** 读为0

bit 5 **SHRSRCEN0:** 输出0共用拉电流使能位  
1 = 使能拉电流镜像模式 (使用来自其他源的参考)  
0 = 禁止拉电流镜像模式

bit 4 **SHRSNKEN0:** 输出0共用灌电流使能位  
1 = 使能灌电流镜像模式 (使用来自其他源的参考)  
0 = 禁止灌电流镜像模式

bit 3 **GENSRCEN0:** 输出0拉电流产生使能位  
1 = 电流源产生拉电流镜像参考  
0 = 电流源不产生拉电流镜像参考

bit 2 **GENSNKEN0:** 输出0灌电流产生使能位  
1 = 电流源产生灌电流镜像参考  
0 = 电流源不产生灌电流镜像参考

bit 1 **SRCEN0:** 输出0拉电流使能位  
1 = 使能拉电流  
0 = 禁止拉电流

bit 0 **SNKEN0:** 输出0灌电流使能位  
1 = 使能灌电流  
0 = 禁止灌电流

## 24.0 运算放大器

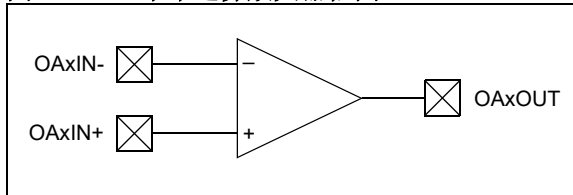
**注：** 28引脚器件仅支持两个运放实例。有关可用性的信息，请参见表1和表2。

dsPIC33CK64MP105系列实现了三个运算放大器（运放）的实例。运放可用于各种用途，包括信号调理和滤波。这三个运放的功能相同。单个放大器的框图如图24-1所示。

运放由两个SFR寄存器控制：AMPCON1L和AMPCON1H。这两个寄存器保持低功耗状态，直到AMPON位置1。之后，可以通过将相应的AMPENx位（x = 1、2和3）置1来独立使能每个运放。

NCHDISx位在输入范围与积分非线性（INL）之间提供了一些灵活性。当NCHDISx = 0（默认）时，运放的输入电压范围较宽（见第31.0节“电气特性”中的表31-39）。当NCHDISx = 1时，将通过缩小输入范围来提高INL性能（降低INL）。

图24-1： 单个运算放大器框图



# dsPIC33CK64MP105系列

## 24.1 运算放大器控制寄存器

寄存器 24-1: **AMPCON1L**: 运放控制低位字寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
AMPON	—	—	—	—	—	—	—
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	AMPEN3 <sup>(1)</sup>	AMPEN2	AMPEN1
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 15

**AMPON:** 运放使能位  
1 = 如果运放模块对应的AMPENx位也置为有效, 则使能运放模块  
0 = 禁止所有运放模块
- bit 14-3

**未实现:** 读为0
- bit 2

**AMPEN3:** 运放3使能位<sup>(1)</sup>  
1 = 如果AMPON位也置为有效, 则使能运放3  
0 = 禁止运放3
- bit 1

**AMPEN2:** 运放2使能位  
1 = 如果AMPON位也置为有效, 则使能运放2  
0 = 禁止运放2
- bit 0

**AMPEN1:** 运放1使能位  
1 = 如果AMPON位也置为有效, 则使能运放1  
0 = 禁止运放1

注 1: 28引脚器件不提供此位。

## 寄存器 24-2: AMPCON1H: 运放控制高位字寄存器

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

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	NCHDIS3 <sup>(1)</sup>	NCHDIS2	NCHDIS1
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 15-3      **未实现:** 读为0

bit 2      **NCHDIS3:** 运放3的N沟道禁止位<sup>(1)</sup>

1 = 禁止运放3的N沟道输入级; 减小了INL, 但降低了输入电压范围

0 = 运放3具有宽输入范围

bit 1      **NCHDIS2:** 运放2的N沟道禁止位

1 = 禁止运放2的N沟道输入级; 减小了INL, 但降低了输入电压范围

0 = 运放2具有宽输入范围

bit 0      **NCHDIS1:** 运放1的N沟道禁止位

1 = 禁止运放1的N沟道输入级; 减小了INL, 但降低了输入电压范围

0 = 运放1具有宽输入范围

**注 1:** 28引脚器件不提供此位。

注:



25.0 程序监控定时器（DMT）

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“程序监控定时器（DMT）”（[www.microchip.com/DS70005155](http://www.microchip.com/DS70005155)）。

程序监控定时器（DMT）的主要功能是在发生软件故障时使处理器中断。DMT 是一个基于系统时钟自由运

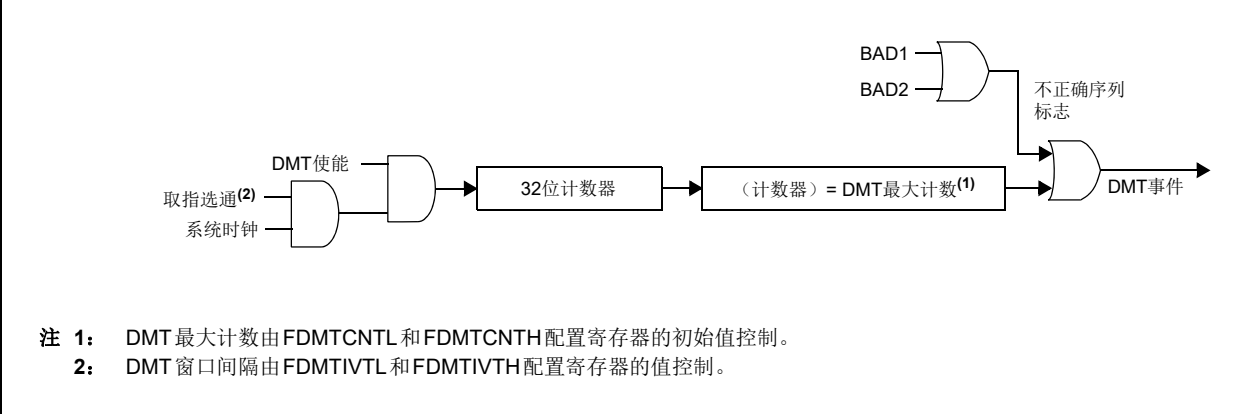
行的取指操作定时器，它在每次发生取指操作时产生计数，直到计数匹配为止。当处理器处于休眠模式时，不会产生取指操作。

可以在配置熔丝中或通过软件将 DMTCON 寄存器中的 ON 位置 1 使能 DMT。DMT 包含一个 32 位计数器，其超时计数匹配值由两个 16 位配置熔丝寄存器指定：FDMTCNTL 和 FDMTCNTH。

DMT 通常用于任务关键型和安全关键型应用，在此类情况下必须检测软件功能和时序方面的每一个故障。

图 25-1 给出了程序监控定时器模块的框图。

图 25-1： 程序监控定时器框图



# dsPIC33CK64MP105系列

## 25.1 程序监控定时器控制/状态寄存器

寄存器 25-1: **DMTCON**: 程序监控定时器控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
ON <sup>(1)</sup>	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15 **ON**: DMT 模块使能位<sup>(1)</sup>

1 = 使能程序监控定时器模块

0 = 不使能程序监控定时器模块

bit 14-0 **未实现**: 读为0

**注 1:** 只有当FDMT 寄存器中的DMTDIS = 0时, 该位才具有控制权。

寄存器 25-2: **DMTPRECLR**: 程序监控定时器预清零寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
STEP1[7:0]							
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为0

-n = POR时的值

1 = 置1

0 = 清零

x = 未知

bit 15-8 **STEP1[7:0]**: DMT 预清零使能位

01000000=使能程序监控定时器预清零 (步骤1)

所有其他

写模式 = 将BAD1标志置1; 发生DMT复位事件时, 这些位将被清零。如果按正确序列向STEP2[7:0]位装入正确值, STEP1[7:0]也将被清零。

bit 7-0 **未实现**: 读为0

## 寄存器 25-3: DMTCLR: 程序监控定时器清零寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
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
STEP2[7:0]							
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 15-8      **未实现:** 读为0

bit 7-0      **STEP2[7:0]:** DMT 清零定时器位

00001000 = 只有按正确序列正确装载 STEP1[7:0] 位后, 才能清零 STEP1[7:0] 位、STEP2[7:0] 位和程序监控定时器。通过读取 DMTCNTL/H 寄存器并观察计数器发生复位, 可以校验对这些位的写操作。

所有其他

写模式 =

将 BAD2 位置 1; STEP1[7:0] 的值保持不变, 并捕捉写入 STEP2[7:0] 的新值。发生 DMT 复位事件时, 这些位将被清零。

# dsPIC33CK64MP105系列

寄存器 25-4: DMTSTAT: 程序监控定时器状态寄存器

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

HC/R-0	HC/R-0	HC/R-0	U-0	U-0	U-0	U-0	R-0
BAD1	BAD2	DMTEVENT	—	—	—	—	WINOPN
bit 7							bit 0

图注:	HC = 硬件清零位		
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 15-8      **未实现:** 读为 0
- bit 7      **BAD1:** 程序监控定时器错误 STEP1[7:0] 值检测位  
1 = 检测到不正确的 STEP1[7:0] 值  
0 = 未检测到不正确的 STEP1[7:0] 值
- bit 6      **BAD2:** 程序监控定时器错误 STEP2[7:0] 值检测位  
1 = 检测到不正确的 STEP2[7:0] 值  
0 = 未检测到不正确的 STEP2[7:0] 值
- bit 5      **DMTEVENT:** 程序监控定时器事件位  
1 = 检测到程序监控定时器事件 (计数器计满或在计数器递增前输入错误的 STEP1[7:0] 或 STEP2[7:0] 值)  
0 = 未检测到程序监控定时器事件
- bit 4-1      **未实现:** 读为 0
- bit 0      **WINOPN:** 程序监控定时器清零窗口位  
1 = 已打开程序监控定时器清零窗口  
0 = 未打开程序监控定时器清零窗口

## 寄存器 25-5: DMTCNTL: 程序监控定时器计数低位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[15:8]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[7:0]							
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 15-0 **COUNTER[15:0]:** 读取 DMT 计数器低位字的当前内容

## 寄存器 25-6: DMTCNTH: 程序监控定时器计数高位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[31:24]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
COUNTER[23:16]							
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 15-0 **COUNTER[31:16]:** 读取 DMT 计数器高位字的当前内容

# dsPIC33CK64MP105系列

寄存器 25-7: DMT PSCNTL: DMT 配置后计数状态低位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[15:8]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[7:0]							
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 15-0 **PSCNT[15:0]:** DMT 指令计数值配置状态低位字  
它总是为 FDMTCNTL 配置寄存器的值。

寄存器 25-8: DMT PSCNTH: DMT 配置后计数状态高位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[31:24]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSCNT[23:16]							
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 15-0 **PSCNT[31:16]:** DMT 指令计数值配置状态高位字  
它总是为 FDMTCNTH 配置寄存器的值。

## 寄存器 25-9: DMTPSINTVL: DMT配置后间隔状态低位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[15:8]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[7:0]							
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 15-0 **PSINTV[15:0]: DMT窗口间隔配置状态低位字**

它总是为FDMTIVTL配置寄存器的值。

## 寄存器 25-10: DMTPSINTVH: DMT配置后间隔状态高位字寄存器

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[31:24]							
bit 15				bit 8			

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
PSINTV[23:16]							
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 15-0 **PSINTV[31:16]: DMT窗口间隔配置状态高位字**

它总是为FDMTIVTH配置寄存器的值。

# dsPIC33CK64MP105系列

寄存器 25-11: DMTHOLDREG: DMT保持寄存器<sup>(1)</sup>

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
UPRCNT[15:8]							
bit 15							
bit 8							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
UPRCNT[7:0]							
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 15-0      **UPRCNT[15:0]:** 上次读取DMTCNTL和DMTCNTH时的DMTCNTH寄存器值位

注 1: DMTHOLDREG 寄存器在复位时初始化为0, 并且仅在读取DMTCNTL和DMTCNTH寄存器时装载。



## 26.0 32位可编程循环冗余校验（CRC）发生器

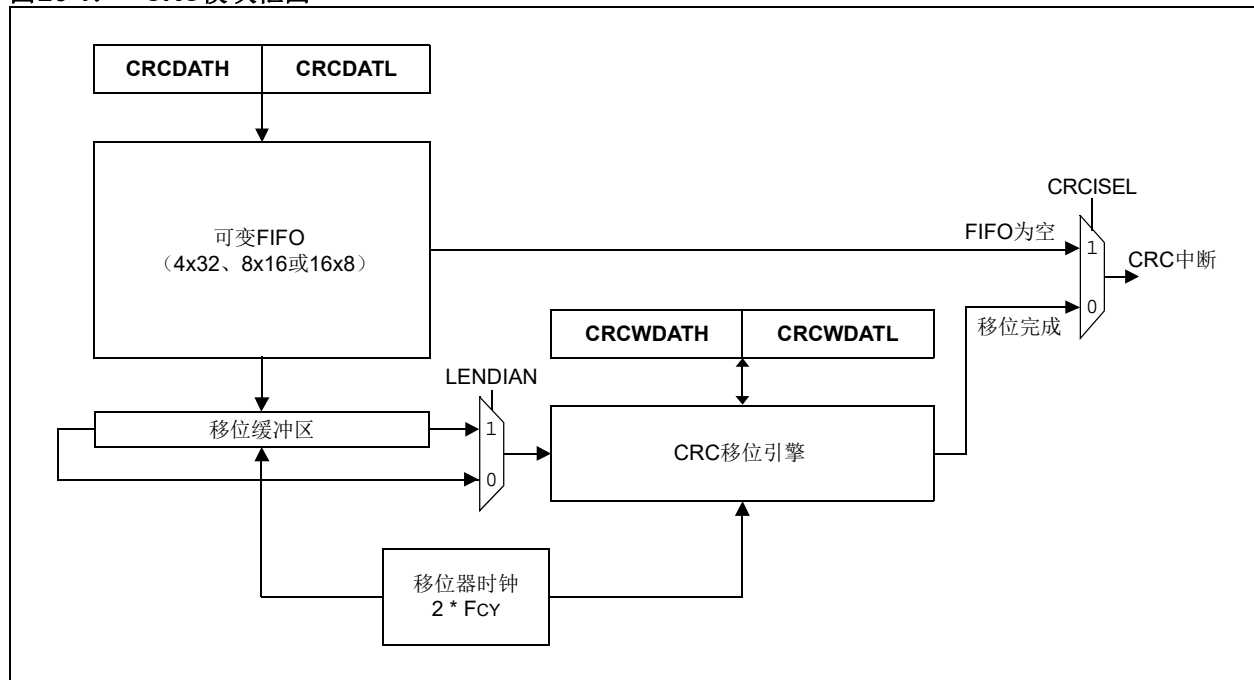
**注 1:** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。更多信息，请参见《dsPIC33/PIC24 系列参考手册》中的“32位可编程循环冗余校验（CRC）”（[www.microchip.com/DS30009729](http://www.microchip.com/DS30009729)）。

32位可编程CRC发生器提供了一种由硬件实现的方法，为各种网络和安全应用快速生成校验和。它具有以下特性：

- 用户可编程的CRC多项式，最多32位
- 可编程移位方向（小尾数法或大尾数法）
- 独立的数据和多项式长度
- 可配置的中断输出
- 数据FIFO

图26-1给出了该模块的概览。

图26-1: CRC模块框图



# dsPIC33CK64MP105系列

## 26.1 CRC控制寄存器

寄存器 26-1: **CRCCONL: CRC控制低位字寄存器**

R/W-0	U-0	R/W-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0	HSC/R-0
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15							bit 8

HSC/R-0	HSC/R-1	R/W-0	HC/R/W-0	R/W-0	R/W-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	MOD	—	—
bit 7							bit 0

图注:	HC = 硬件清零位	HSC = 硬件置 1/清零位
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 15 **CRCEN**: CRC 使能位  
1 = 使能模块  
0 = 禁止模块
- bit 14 **未实现**: 读为 0
- bit 13 **CSIDL**: CRC 空闲模式停止位  
1 = 当器件进入空闲模式时, 模块停止工作  
0 = 在空闲模式下模块继续工作
- bit 12-8 **VWORD[4:0]**: 指针值位  
表示 FIFO 中的有效字数。当  $PLEN[4:0] \geq 7$  时, 最大值为 8; 当  $PLEN[4:0] \leq 7$  时, 最大值为 16。
- bit 7 **CRCFUL**: CRC FIFO 满位  
1 = FIFO 已满  
0 = FIFO 未满
- bit 6 **CRCMPT**: CRC FIFO 空位  
1 = FIFO 为空  
0 = FIFO 非空
- bit 5 **CRCISEL**: CRC 中断选择位  
1 = 在 FIFO 为空时产生中断; 最后一个数据字仍然在通过 CRC 移位  
0 = 在移位完成且结果就绪时产生中断
- bit 4 **CRCGO**: CRC 启动位  
1 = 启动 CRC 串行移位器  
0 = 关闭 CRC 串行移位器
- bit 3 **LENDIAN**: 数据移位方向选择位  
1 = 数据字从 LSb 开始移入 FIFO (小尾数法)  
0 = 数据字从 MSb 开始移入 FIFO (大尾数法)
- bit 2 **MOD**: CRC 计算模式位  
1 = 备用模式  
0 = 传统模式
- bit 1-0 **未实现**: 读为 0

**寄存器 26-2: CRCCONH: CRC控制高位字寄存器**

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0
bit 15							
			bit 8				
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0
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 15-13      **未实现:** 读为0

bit 12-8      **DWIDTH[4:0]:** 数据字宽度配置位  
配置数据字的宽度 (数据字宽度 - 1)。

bit 7-5      **未实现:** 读为0

bit 4-0      **PLEN[4:0]:** 多项式长度配置位  
配置多项式的长度 (多项式长度 - 1)。

# dsPIC33CK64MP105系列

寄存器 26-3: CRCXORL: CRC XOR多项式低位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X[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	U-0
X[7:1]							—
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 15-1 **X[15:1]**: 多项式项 $x^n$ 的XOR使能位

bit 0 未实现: 读为0

寄存器 26-4: CRCXORH: CRC XOR多项式高位字寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X[31:24]							
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
X[23:16]							
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 15-0 **X[31:16]**: 多项式项 $x^n$ 的XOR使能位

27.0 节能特性

**注 1:** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“看门狗定时器和节能模式”（[www.microchip.com/DS70615](http://www.microchip.com/DS70615)）。

dsPIC33CK64MP105 系列器件提供了管理功耗的功能，该功能是通过有选择地管理 CPU 和外设的时钟来实现的。一般来说，较低的时钟频率和减少时钟所驱动外设的数目可降低功耗。

dsPIC33CK64MP105 系列器件可通过以下 4 种方式来管理功耗：

- 时钟频率
- 基于指令的休眠模式和空闲模式
- 软件控制的打盹模式
- 用软件有选择地进行外设控制

可以组合使用这些方法，从而在保证关键应用特性（如对于时序敏感的通信）的情况下有选择地调节应用的功耗。

例 27-1: PWRSAV 指令语法

```
PWRSAV #0          ; Put the device into Sleep mode
PWRSAV #1          ; Put the device into Idle mode
```

27.1 时钟频率和时钟切换

dsPIC33CK64MP105 系列器件的时钟频率范围较宽，用户可根据应用需要进行选择。如果未锁定系统时钟配置，用户只需更改 NOSC<sub>x</sub> 位（OSCCON[10:8]）即可选择低功耗或高精度振荡器。在工作期间更改系统时钟的过程以及相应的限制，已在第 9.0 节“带高频 PLL 的振荡器”中进行了详细的讨论。

27.2 基于指令的节能模式

dsPIC33CK64MP105 系列器件有两种特殊的节能模式，通过执行特殊的 PWRSAV 指令可以进入这两种模式。休眠模式下时钟停止工作并暂停所有代码执行。空闲模式下 CPU 暂停工作并暂停代码执行，但是允许外设模块继续工作。PWRSAV 指令的汇编语法如例 27-1 所示。

在被允许的中断产生、WDT 超时或器件复位时，器件会退出休眠和空闲模式。器件退出这两种模式的过程称为“唤醒”。

27.2.1 休眠模式

休眠模式具有以下特征：

- 系统时钟源关闭。如果使用了片上振荡器，则它也会关闭。
- 如果没有 I/O 引脚拉出电流，则器件电流消耗将降至最低。
- 因为系统时钟源被禁止，所以故障保护时钟监视器在休眠模式下不工作。
- 如果 WDT 被使能，则 LPRC 时钟将在休眠模式下继续运行。
- 如果 WDT 被使能，则它在进入休眠模式之前被自动清零。
- 有些器件功能或外设能在休眠模式下继续工作，包括 I/O 端口上的输入电平变化通知功能或使用外部时钟输入的外设等。
- 任何需要使用系统时钟源工作的外设在休眠模式下将被禁止。

当发生以下任何事件时，器件将被从休眠模式唤醒：

- 产生任何已被单独允许的中断
- 任何形式的器件复位
- WDT 超时

从休眠模式唤醒时，处理器将使用在进入休眠模式时处于工作状态的时钟源重新开始工作。

为了实现最佳的节能效果，可以通过清零 VREGS (RCON[8]) 位（默认配置），将稳压器配置为在休眠模式下进入待机状态。

如果应用需要更快的唤醒速度，并且可以接受更高的电流要求，可以将 VREGS (RCON[8]) 位置 1，使稳压器在休眠模式下保持工作状态。表 27-1 给出了可用的低功耗休眠模式。第 28.4 节“片上稳压器”提供了其他稳压器信息。

表 27-1： 低功耗休眠模式

相对功耗	LPWREN	VREGS	模式
最高	0	1	全功耗，工作
—	0	0	全功耗，待机
—	1 <sup>(1)</sup>	1	低功耗，工作
最低	1 <sup>(1)</sup>	0	低功耗，待机

注 1： 低功耗模式，当 LPWREN = 1 时，只能用于工业温度范围。

27.2.2 空闲模式

空闲模式具有以下特征：

- CPU 停止执行指令。
- WDT 被自动清零。
- 系统时钟源保持工作状态。默认情况下，所有外设模块将继续使用系统时钟源正常工作，也可以有选择地禁止它们（见第 27.4 节“外设模块禁止”）。
- 如果 WDT 或 FSCM 被使能，则 LPRC 也将保持工作状态。

当发生以下任何事件时，器件将被从空闲模式唤醒：

- 产生任何已被单独允许的中断
- 任何器件复位
- WDT 超时

从空闲模式唤醒时，重新为 CPU 提供时钟并开始执行指令（2-4 个时钟周期后），从 PWRSAV 指令之后的下一条指令或 ISR 中的第一条指令开始执行。

所有外设还都具有在进入空闲模式时停止工作的选项，以提高节能效果。该选项可在每个外设的控制寄存器中进行选择；例如，Timer1 控制寄存器中的 SIDL 位 (T1CON[13])。

27.2.3 在节能指令执行期间的中断

在执行 PWRSAV 指令时产生的任何中断都将延迟到进入休眠或空闲模式后才起作用，并导致器件从休眠或空闲模式唤醒。

## 27.3 打盹模式

更改时钟速度和使用某种节能模式是降低功耗的首选策略。在某些情况下可能不可行。例如，某些应用可能必须保持不间断的同步通信，即便在它不执行任何其他操作时也不例外。降低系统时钟速度可能会带来通信错误，而使用节能模式可能会完全停止通信。

打盹模式是另一种简单有效的节能方法，它可以在器件仍然执行代码的情况下降低功耗。在此模式下，系统时钟以相同的时钟源和相同的速度继续工作。外设模块时钟速度保持不变，但CPU时钟速度降低了。保持这两个时钟域同步，可以允许外设访问SFR，同时CPU以较慢的速度执行代码。

通过将DOZEN位（CLKDIV[11]）置1使能打盹模式。外设与内核的时钟速度之比是由DOZE[2:0]位（CLKDIV[14:12]）决定的。有8种可能的配置，从1:1至1:128，其中1:1是默认设置。

在事件驱动的应用中，程序可以使用打盹模式有选择地降低功耗。这样就可以实现不间断地运行对时钟敏感的功能（如同步通信），而CPU保持空闲等待事件调用中断服务程序。通过将ROI位（CLKDIV[15]）置1，可以使器件在产生中断时自动返回到全速CPU工作模式。默认情况下，中断事件对打盹模式工作没有影响。

## 27.4 外设模块禁止

外设模块禁止（PMD）寄存器通过停止提供给外设模块的所有时钟源提供一种禁止外设模块的方法。当通过相应的PMD控制位禁止外设时，外设就进入了功耗最低的状态。与外设相关的控制和状态寄存器也会被禁止，因此写入这些寄存器不起作用，且读取值无效。

只有在PMD寄存器中的相应位被清零且特定的dsPIC DSC器件支持外设时，才会使能相应的外设模块。如果外设存在于器件中，则默认情况下，它是通过PMD寄存器使能的。

**注 1:** 如果PMD位置1，则对应的模块将在一个指令周期的延时后被禁止。类似地，如果PMD位清零，则对应的模块将在一个指令周期的延时后被使能（假设已将模块控制寄存器配置为使能模块的工作）。

## 27.5 节能资源

对于本数据手册中列出的器件，Microchip网站的主要产品页面中提供了许多有用的资源。该产品页面包含最新更新和附加信息。

### 27.5.1 主要资源

- 《dsPIC33/PIC24系列参考手册》中的“看门狗定时器和节能模式”（[www.microchip.com/DS70615](http://www.microchip.com/DS70615)）
- 代码示例
- 应用笔记
- 软件库
- 网上研讨会
- 所有相关的《dsPIC33/PIC24系列参考手册》章节
- 开发工具

# dsPIC33CK64MP105系列

## 27.6 PMD控制寄存器

寄存器 27-1: PMD1: 外设模块禁止1控制寄存器

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	T1MD	QE11MD	PWMMD	—
bit 15				bit 8			

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD
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 15-12      **未实现:** 读为0
- bit 11        **T1MD:** Timer1 模块禁止位  
               1 = 禁止Timer1 模块  
               0 = 使能Timer1 模块
- bit 10        **QE11MD:** QE11 模块禁止位  
               1 = 禁止QE11 模块  
               0 = 使能QE11 模块
- bit 9         **PWMMD:** PWM 模块禁止位  
               1 = 禁止PWM 模块  
               0 = 使能PWM 模块
- bit 8         **未实现:** 读为0
- bit 7         **I2C1MD:** I2C1 模块禁止位  
               1 = 禁止I2C1 模块  
               0 = 使能I2C1 模块
- bit 6         **U2MD:** UART2 模块禁止位  
               1 = 禁止UART2 模块  
               0 = 使能UART2 模块
- bit 5         **U1MD:** UART1 模块禁止位  
               1 = 禁止UART1 模块  
               0 = 使能UART1 模块
- bit 4         **SPI2MD:** SPI2 模块禁止位  
               1 = 禁止SPI2 模块  
               0 = 使能SPI2 模块
- bit 3         **SPI1MD:** SPI1 模块禁止位  
               1 = 禁止SPI1 模块  
               0 = 使能SPI1 模块
- bit 2-1       **未实现:** 读为0
- bit 0         **ADC1MD:** ADC 模块禁止位  
               1 = 禁止ADC 模块  
               0 = 使能ADC 模块



**寄存器 27-2: PMD2: 外设模块禁止2控制寄存器**

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
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 15-5 **未实现:** 读为0

bit 4 **CCP5MD:** SCCP5模块禁止位

1 = 禁止SCCP5模块

0 = 使能SCCP5模块

bit 3 **CCP4MD:** SCCP4模块禁止位

1 = 禁止SCCP4模块

0 = 使能SCCP4模块

bit 2 **CCP3MD:** SCCP3模块禁止位

1 = 禁止SCCP3模块

0 = 使能SCCP3模块

bit 1 **CCP2MD:** SCCP2模块禁止位

1 = 禁止SCCP2模块

0 = 使能SCCP2模块

bit 0 **CCP1MD:** SCCP1模块禁止位

1 = 禁止SCCP1模块

0 = 使能SCCP1模块

# dsPIC33CK64MP105系列

寄存器27-3: **PMD3: 外设模块禁止3控制寄存器**

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

R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0
CRCMD	—	QE12MD	—	U3MD	—	I2C2MD	—
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 15-8      **未实现:** 读为0
- bit 7        **CRCMD:** CRC模块禁止位  
1 = 禁止CRC模块  
0 = 使能CRC模块
- bit 6        **未实现:** 读为0
- bit 5        **QE12MD:** QE12模块禁止位  
1 = 禁止QE12模块  
0 = 使能QE12模块
- bit 4        **未实现:** 读为0
- bit 3        **U3MD:** UART3模块禁止位  
1 = 禁止UART3模块  
0 = 使能UART3模块
- bit 2        **未实现:** 读为0
- bit 1        **I2C2MD:** I2C2模块禁止位  
1 = 禁止I2C2模块  
0 = 使能I2C2模块
- bit 0        **未实现:** 读为0

**寄存器 27-4: PMD4: 外设模块禁止4控制寄存器**

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

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	REFOMD	—	—	—
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 15-4      **未实现:** 读为0

bit 3      **REFOMD:** 参考时钟模块禁止位

1 = 禁止参考时钟模块

0 = 使能参考时钟模块

bit 2-0      **未实现:** 读为0

# dsPIC33CK64MP105系列

寄存器 27-5: PMD6: 外设模块禁止6控制寄存器

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	DMA3MD	DMA2MD	DMA1MD	DMA0MD
bit 15				bit 8			

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	SPI3MD
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 15-12 未实现: 读为0

bit 11 **DMA3MD:** DMA3 模块禁止位

1 = 禁止DMA3 模块

0 = 使能DMA3 模块

bit 10 **DMA2MD:** DMA2 模块禁止位

1 = 禁止DMA2 模块

0 = 使能DMA2 模块

bit 9 **DMA1MD:** DMA1 模块禁止位

1 = 禁止DMA1 模块

0 = 使能DMA1 模块

bit 8 **DMA0MD:** DMA0 模块禁止位

1 = 禁止DMA0 模块

0 = 使能DMA0 模块

bit 7-1 未实现: 读为0

bit 0 **SPI3MD:** SPI3 模块禁止位

1 = 禁止SPI3 模块

0 = 使能SPI3 模块

**寄存器 27-6: PMD7: 外设模块禁止7控制寄存器**

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CMP3MD	CMP2MD	CMP1MD
bit 15					bit 8		

U-0	U-0	U-0	U-0	R/W-0	U-0	U-0	U-0
—	—	—	—	PTGMD	—	—	—
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 15-11     **未实现:** 读为0
- bit 10        **CMP3MD:** 比较器3模块禁止位  
1 = 禁止比较器3模块  
0 = 使能比较器3模块
- bit 9         **CMP2MD:** 比较器2模块禁止位  
1 = 禁止比较器2模块  
0 = 使能比较器2模块
- bit 8         **CMP1MD:** 比较器1模块禁止位  
1 = 禁止比较器1模块  
0 = 使能比较器1模块
- bit 7-4       **未实现:** 读为0
- bit 3         **PTGMD:** PTG模块禁止位  
1 = 禁止PTG模块  
0 = 使能PTG模块
- bit 2-0       **未实现:** 读为0

# dsPIC33CK64MP105系列

## 寄存器 27-7: PMD8: 外设模块禁止8控制寄存器

U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
—	—	OPAMPMD	SENT2MD	SENT1MD	—	—	DMTMD
bit 15						bit 8	

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	BIASMD	—
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 15-14     **未实现:** 读为0
- bit 13       **OPAMPMD:** 运放模块禁止位  
1 = 禁止运放模块  
0 = 使能运放模块
- bit 12       **SENT2MD:** SENT2模块禁止位  
1 = 禁止SENT2模块  
0 = 使能SENT2模块
- bit 11       **SENT1MD:** SENT1模块禁止位  
1 = 禁止SENT1模块  
0 = 使能SENT1模块
- bit 10-9     **未实现:** 读为0
- bit 8        **DMTMD:** 程序监控定时器模块禁止位  
1 = 禁止DMT模块  
0 = 使能DMT模块
- bit 7-6      **未实现:** 读为0
- bit 5        **CLC4MD:** CLC4模块禁止位  
1 = 禁止CLC4模块  
0 = 使能CLC4模块
- bit 4        **CLC3MD:** CLC3模块禁止位  
1 = 禁止CLC3模块  
0 = 使能CLC3模块
- bit 3        **CLC2MD:** CLC2模块禁止位  
1 = 禁止CLC2模块  
0 = 使能CLC2模块
- bit 2        **CLC1MD:** CLC1模块禁止位  
1 = 禁止CLC1模块  
0 = 使能CLC1模块
- bit 1        **BIASMD:** 恒流源模块禁止位  
1 = 禁止恒流源模块  
0 = 使能恒流源模块
- bit 0        **未实现:** 读为0

表27-2: PMD寄存器

寄存器	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PMD1	—	—	—	—	T1MD	QEIMD	PWMMD	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD
PMD2	—	—	—	—	—	—	—	—	—	—	—	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD
PMD3	—	—	—	—	—	—	—	—	CRCMD	—	QEI2MD	—	U3MD	—	I2C2MD	—
PMD4	—	—	—	—	—	—	—	—	—	—	—	—	REFOMD	—	—	—
PMD6	—	—	—	—	DMA3MD	DMA2MD	DMA1MD	DMA0MD	—	—	—	—	—	—	—	SPI3MD
PMD7	—	—	—	—	—	CMP3MD	CMP2MD	CMP1MD	—	—	—	—	PTGMD	—	—	—
PMD8	—	—	OPAMPMD	SENT2MD	SENT1MD	—	—	DMTMD	—	—	CLC4MD	CLC3MD	CLC2MD	CLC1MD	BIASMD	—

注:



## 28.0 特殊功能

**注：** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24系列参考手册》中的相关章节，该文档可从 Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 下载。

dsPIC33CK64MP105 系列器件的一些特殊功能旨在最大限度地提高应用的灵活性和可靠性，并通过减少外部元件将成本降至最低。提供的特殊功能包括：

- 灵活的配置
- 看门狗定时器 (WDT)
- 代码保护和 CodeGuard™ 安全性
- JTAG 边界扫描接口
- 在线串行编程 (ICSP)
- 在线仿真
- 欠压复位 (Brown-out Reset, BOR)

## 28.1 配置位

在 dsPIC33CK64MP105 系列器件中，配置字以易失性存储方式实现。这就意味着每次器件上电时都要将配置数据从闪存配置字装入易失性存储器。配置数据存储在片上程序存储空间的末尾，称为闪存配置字。表 28-1 列出了这些配置字的具体地址。器件复位期间，配置数据会自动从闪存配置字装入到相应的配置影子寄存器中。

**注：** 所有类型的器件复位都会重新装入配置数据。

在为这些器件创建应用程序时，用户应总是在其代码中告知编译器特别为配置数据分配闪存配置字地址，以确保当编译代码时程序代码不会存储到该地址。从配置空间执行程序代码会导致器件复位。

**注：** 对程序存储器的最后一页执行页擦除操作会清除闪存配置字。

表 28-1: dsPIC33CKXXMPX0X 配置地址

寄存器名称	64k	32k
FSEC	0x00AF00	0x005F00
FBSLIM	0x00AF10	0x005F10
FSIGN	0x00AF14	0x005F14
FOSCSEL	0x00AF18	0x005F18
FOSC	0x00AF1C	0x005F1C
FWDT	0x00AF20	0x005F20
FPOR	0x00AF24	0x005F24
FICD	0x00AF28	0x005F28
FDMTIVTL	0x00AF2C	0x005F2C
FDMTIVTH	0x00AF30	0x005F30
FDMTCNTL	0x00AF34	0x005F34
FDMTCNTH	0x00AF38	0x005F38
FDMT	0x00AF3C	0x005F3C
FDEVOPT	0x00AF40	0x005F40
FALTREG	0x00AF44	0x005F44

表28-2: 配置寄存器映射

寄存器名称	Bit 23-16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FSEC	—	AIVTDIS	—	—	—	CSS[2:0]			CWRP	GSS[1:0]		GWRP	—	BSEN	BSS[1:0]		BWRP
FBSLIM	—	—	—	—	BSLIM[12:0]												
FSIGN	—	r <sup>(2)</sup>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
FOSCSSEL	—	—	—	—	—	—	—	—	—	IESO	—	—	—	—	FNOSC[2:0]		
FOSC	—	—	—	—	XTBST	XTCFG[1:0]		—	PLLKEN	FCKSM[1:0]		—	—	—	OSCIOFCN	POSCMD[1:0]	
FWDT	—	FWDTEN	SWDTPS[4:0]					WDTWIN[1:0]		WINDIS	RCLKSEL[1:0]		RWDTPS[4:0]				
FPOR	—	—	—	—	—	—	r <sup>(1)</sup>	—	—	—	BISTDIS	r <sup>(1)</sup>	r <sup>(1)</sup>	—	—	—	—
FICD	—	—	—	—	—	—	—	—	—	r <sup>(1)</sup>	—	JTAGEN	—	—	—	ICS[1:0]	
FDMTIVTL	—	DMTIVT[15:0]															
FDMTIVTH	—	DMTIVT[31:16]															
FDMTCNTL	—	DMTCNT[15:0]															
FDMTCNTH	—	DMTCNT[31:16]															
FDMT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMTDIS
FDEVOPT	—	—	—	SPI2PIN	—	—	SMB3EN	r <sup>(2)</sup>	r <sup>(2)</sup>	r <sup>(1)</sup>	—	—	ALT12C2	ALT12C1	r <sup>(1)</sup>	—	—
FALTREG	—	—	CTXT4[2:0]			—	CTXT3[2:0]			—	CTXT2[2:0]			—	CTXT1[2:0]		

图注: — = 未实现位, 读为1; r = 保留位。

注 1: 位被保留, 保持为1。

2: 位被保留, 保持为0。

## 寄存器 28-1: FSEC 配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
AIVTDIS	—	—	—	CSS2	CSS1	CSS0	CWRP
bit 15						bit 8	

R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
GSS1	GSS0	GWRP	—	BSEN	BSS1	BSS0	BWRP
bit 7						bit 0	

<b>图注:</b>	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

bit 23-16	<b>未实现:</b> 读为1
bit 15	<b>AIVTDIS:</b> 备用中断向量表禁止位 1 = 禁止AIVT 0 = 使能AIVT
bit 14-12	<b>未实现:</b> 读为1
bit 11-9	<b>CSS[2:0]:</b> 配置段代码闪存保护级别位 111 = 无保护 (CWRP写保护除外) 110 = 标准安全性 10x = 增强安全性 0xx = 高安全性
bit 8	<b>CWRP:</b> 配置段写保护位 1 = 配置段不被写保护 0 = 配置段被写保护
bit 7-6	<b>GSS[1:0]:</b> 通用段代码闪存保护级别位 11 = 无保护 (GWRP写保护除外) 10 = 标准安全性 0x = 高安全性
bit 5	<b>GWRP:</b> 通用段写保护位 1 = 用户程序存储区不被写保护 0 = 用户程序存储区被写保护
bit 4	<b>未实现:</b> 读为1
bit 3	<b>BSEN:</b> 引导段控制位 1 = 无引导段 0 = 引导段大小由BSLIM[12:0]决定
bit 2-1	<b>BSS[1:0]:</b> 引导段代码闪存保护级别位 11 = 无保护 (BWRP写保护除外) 10 = 标准安全性 0x = 高安全性
bit 0	<b>BWRP:</b> 引导段写保护位 1 = 用户程序存储区不被写保护 0 = 用户程序存储区被写保护

# dsPIC33CK64MP105系列

**寄存器 28-2: FBSLIM配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
—	—	—	BSLIM[12:8] <sup>(1)</sup>				
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
BSLIM[7:0] <sup>(1)</sup>							
bit 7							bit 0

<b>图注:</b>	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

bit 23-13 **未实现:** 读为1

bit 12-0 **BSLIM[12:0]:** 引导段代码闪存页地址限制位<sup>(1)</sup>  
包含第一个有效的通用段页的页地址。要编程的值为页地址的补码, 这样编程额外的0只会增加引导段的大小。

**注 1:** BSLIMx位是“写入一次”元素。在复位序列之后, 如果没有擦除这些位(全1), 则将禁止编程FBSLIM位。尝试编程不会将WR位(NVMCON[15])置1, 因此没有影响。

**寄存器 28-3: FSIGN配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-0	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 7							bit 0

<b>图注:</b>	r = 保留位	PO = 一次编程位	
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

bit 23-16 **未实现:** 读为1

bit 15 **保留:** 保持为0

bit 14-0 **未实现:** 读为1

**寄存器 28-4: FOSCSEL配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15						bit 8	

R/PO-1	U-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
IESO	—	—	—	—	FNOSC2	FNOSC1	FNOSC0
bit 7						bit 0	

<b>图注:</b>	PO = 一次编程位						
R = 可读位	W = 可写位			U = 未实现位, 读为0			
-n = 擦除值	1 = 置1			0 = 清零		x = 未知	

bit 23-8 **未实现:** 读为1

bit 7 **IESO:** 内/外部时钟源切换位  
 1 = 使能内/外部时钟源切换模式 (使能双速启动)  
 0 = 禁止内/外部时钟源切换模式 (禁止双速启动)

bit 6-3 **未实现:** 读为1

bit 2-0 **FNOSC[2:0]:** 初始振荡器源选择位  
 111 = 带后分频器的内部快速RC (FRC) 振荡器  
 110 = 备用快速RC (BFRC)  
 101 = LPRC振荡器  
 100 = 保留  
 011 = 带PLL的主振荡器 (XTPLL、HSPLL和ECPLL)  
 010 = 主振荡器 (XT、HS和EC)  
 001 = 带PLL的内部快速RC振荡器 (FRCPLL)  
 000 = 快速RC (FRC) 振荡器

# dsPIC33CK64MP105系列

寄存器 28-5: FOSC配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1
—	—	—	XTBST	XTCFG1	XTCFG0	—	PLLKEN <sup>(1)</sup>
bit 15						bit 8	

R/PO-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	—	—	—	OSCIOFNC	POSCMD1	POSCMD0
bit 7						bit 0	

图注:	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

- bit 23-13      **未实现:** 读为1
- bit 12      **XTBST:** 振荡器启动编程位  
1 = 加快启动  
0 = 默认启动
- bit 11-10      **XTCFG[1:0]:** 晶振驱动选择位  
振荡器的电流增益编程 (输出驱动)。  
11 = 增益3 (用于 24-32 MHz 晶振)  
10 = 增益2 (用于 16-24 MHz 晶振)  
01 = 增益1 (用于 8-16 MHz 晶振)  
00 = 增益0 (用于 4-8 MHz 晶振)
- bit 9      **未实现:** 读为1
- bit 8      **PLLKEN:** PLL 锁定使能位<sup>(1)</sup>  
1 = 如果锁定丢失, 将禁止PLL时钟输出  
0 = 如果锁定丢失, 将不会禁止PLL时钟输出
- bit 7-6      **FCKSM[1:0]:** 时钟切换模式位  
1x = 禁止时钟切换, 禁止故障保护时钟监视器  
01 = 使能时钟切换, 禁止故障保护时钟监视器  
00 = 使能时钟切换, 使能故障保护时钟监视器
- bit 5-3      **未实现:** 读为1
- bit 2      **OSCIOFNC:** OSCO 引脚功能位 (XT 和 HS 模式下除外)  
1 = OSCO 为时钟输出  
0 = OSCO 为通用数字 I/O 引脚
- bit 1-0      **POSCMD[1:0]:** 主振荡器模式选择位  
11 = 禁止主振荡器  
10 = HS 晶振模式 (10 MHz-32 MHz)  
01 = XT 晶振模式 (3.5 MHz-10 MHz)  
00 = EC (外部时钟) 模式

注 1: 当系统时钟切换逻辑请求PLL时钟源且PLL尚未使能时, 将发生超时周期。

**寄存器 28-6: FWDT配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23				bit 16			

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
FWDTEN	SWDTPS4	SWDTPS3	SWDTPS2	SWDTPS1	SWDTPS0	WDTWIN1	WDTWIN0
bit 15				bit 8			

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
WINDIS	RCLKSEL1	RCLKSEL0	RWDTPS4	RWDTPS3	RWDTPS2	RWDTPS1	RWDTPS0
bit 7				bit 0			

<b>图注:</b>	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

bit 23-16 **未实现:** 读为1

bit 15 **FWDTEN:** 看门狗定时器使能位  
1 = 在硬件中使能WDT  
0 = WDT由ON位 (WDTCONL[15]) 控制

bit 14-10 **SWDTPS[4:0]:** 休眠模式看门狗定时器周期选择位  
111111 =  $2^{31}$  (= 2,147,483,648) 分频  
111110 =  $2^{30}$  (= 1,073,741,824) 分频  
...  
00001 =  $2^1$  (= 2) 分频  
00000 =  $2^0$  (= 1) 分频

bit 9-8 **WDTWIN[1:0]:** 看门狗定时器窗口选择位  
11 = WDT窗口为WDT周期的25%  
10 = WDT窗口为WDT周期的37.5%  
01 = WDT窗口为WDT周期的50%  
00 = WDT窗口为WDT周期的75%

bit 7 **WINDIS:** 看门狗定时器窗口使能位  
1 = 看门狗定时器处于非窗口模式  
0 = 看门狗定时器处于窗口模式

bit 6-5 **RCLKSEL[1:0]:** 看门狗定时器时钟选择位  
11 = LPRC时钟  
10 = 当WINDIS = 0、系统时钟不是INTOSC/LPRC且器件未处于休眠模式时, 使用FRC; 否则, 使用INTOSC/LPRC  
01 = 当系统时钟不是INTOSC/LPRC且器件未处于休眠模式时, 使用外设时钟; 否则, 使用INTOSC/LPRC  
00 = 保留

bit 4-0 **RWDTPS[4:0]:** 运行模式看门狗定时器周期选择位  
111111 =  $2^{31}$  (= 2,147,483,648) 分频  
111110 =  $2^{30}$  (= 1,073,741,824) 分频  
...  
00001 =  $2^1$  (= 2) 分频  
00000 =  $2^0$  (= 1) 分频

# dsPIC33CK64MP105系列

寄存器 28-7: FPOR配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	
U-1	U-1	U-1	U-1	U-1	r-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15						bit 8	
U-1	R/PO-1 <sup>(1)</sup>	r-1	r-1	U-1	U-1	U-1	U-1
—	BISTDIS	—	—	—	—	—	—
bit 7						bit 0	

图注:	PO = 一次编程位	r = 保留位
R = 可读位	W = 可写位	U = 未实现位，读为0
-n = 擦除值	1 = 置1	0 = 清零
		x = 未知

- bit 23-11 未实现：读为1
- bit 10 保留：保持为1
- bit 9-7 未实现：读为1
- bit 6 **BISTDIS**: 存储器BIST功能禁止位<sup>(1)</sup>
  - 1 = 禁止复位时MBIST功能
  - 0 = 使能复位时MBIST功能
- bit 5-4 保留：保持为0b11
- bit 3-0 未实现：读为1

注 1: 仅适用于上电复位（POR）。



**寄存器 28-8: FICD配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15						bit 8	

r-1	U-1	R/PO-1	U-1	U-1	U-1	R/PO-1	R/PO-1
—	—	JTAGEN	—	—	—	ICS1	ICS0
bit 7						bit 0	

<b>图注:</b>	PO = 一次编程位	r = 保留位
R = 可读位	W = 可写位	U = 未实现位, 读为0
-n = 擦除值	1 = 置1	0 = 清零
		x = 未知

- bit 23-8      **未实现:** 读为1
- bit 7        **保留:** 保持为1
- bit 6        **未实现:** 读为1
- bit 5        **JTAGEN:** JTAG 使能位
  - 1 = 使能JTAG 端口
  - 0 = 禁止JTAG 端口
- bit 4-2      **未实现:** 读为1
- bit 1-0      **ICS[1:0]:** ICD 通信通道选择位
  - 11 = 通过PGC1和PGD1进行通信
  - 10 = 通过PGC2和PGD2进行通信
  - 01 = 通过PGC3和PGD3进行通信
  - 00 = 保留; 不要使用

# dsPIC33CK64MP105系列

寄存器 28-9: FDMTIVTL配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[15:8]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[7:0]							
bit 7							bit 0

图注: PO = 一次编程位  
R = 可读位 W = 可写位 U = 未实现位, 读为0  
-n = 擦除值 1 = 置1 0 = 清零 x = 未知

bit 23-16 未实现: 读为1  
bit 15-0 DMTIVT[15:0]: DMT窗口间隔低16位

寄存器 28-10: FDMTIVTH配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[31:24]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTIVT[23:16]							
bit 7							bit 0

图注: PO = 一次编程位  
R = 可读位 W = 可写位 U = 未实现位, 读为0  
-n = 擦除值 1 = 置1 0 = 清零 x = 未知

bit 23-16 未实现: 读为1  
bit 15-0 DMTIVT[31:16]: DMT窗口间隔高16位

## 寄存器 28-11: FDMTCNTL配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[15:8]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[7:0]							
bit 7							bit 0

<b>图注:</b>	PO = 一次编程位						
R = 可读位	W = 可写位			U = 未实现位, 读为0			
-n = 擦除值	1 = 置1			0 = 清零		x = 未知	

bit 23-16 未实现: 读为1  
bit 15-0 **DMTCNT[15:0]:** DMT 指令计数超时值低 16 位

## 寄存器 28-12: FDMTCNTH配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[31:24]							
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
DMTCNT[23:16]							
bit 7							bit 0

<b>图注:</b>	PO = 一次编程位						
R = 可读位	W = 可写位			U = 未实现位, 读为0			
-n = 擦除值	1 = 置1			0 = 清零		x = 未知	

bit 23-16 未实现: 读为1  
bit 15-0 **DMTCNT[31:16]:** DMT 指令计数超时值高 16 位

# dsPIC33CK64MP105系列

寄存器 28-13: FDMT配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 15							bit 8

U-1	U-1	U-1	U-1	U-1	U-1	U-1	R/PO-1
—	—	—	—	—	—	—	DMTDIS
bit 7							bit 0

图注: PO = 一次编程位  
R = 可读位 W = 可写位 U = 未实现位, 读为0  
-n = 擦除值 1 = 置1 0 = 清零 x = 未知

bit 23-1 未实现: 读为1  
bit 0 **DMTDIS**: DMT禁止位  
1 = 禁止DMT  
0 = 使能DMT

**寄存器 28-14: FDEVOPT配置寄存器**

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	
U-1	U-1	R/PO-1	U-1	U-1	R/PO-1	r-0	r-0
—	—	SPI2PIN <sup>(1)</sup>	—	—	SMB3EN <sup>(2)</sup>	—	—
bit 15						bit 8	
r-1	U-1	U-1	R/PO-1	R/PO-1	r-1	U-1	U-1
—	—	—	ALT12C2	ALT12C1	—	—	—
bit 7						bit 0	

<b>图注:</b>	PO = 一次编程位	r = 保留位
R = 可读位	W = 可写位	U = 未实现位, 读为0
-n = 擦除值	1 = 置1	0 = 清零
		x = 未知

- bit 23-14 **未实现:** 读为1
- bit 13 **SPI2PIN:** 主SPI2快速I/O引脚禁止位<sup>(1)</sup>  
 1 = 主SPI2使用PPS (I/O重映射) 与器件引脚建立连接  
 0 = 主SPI2直接连接指定的器件引脚
- bit 12-11 **未实现:** 读为1
- bit 10 **SMB3EN:** SMBus 3.0电平使能位<sup>(2)</sup>  
 1 = SMBus 3.0输入电平  
 0 = 标准SMBus输入电平
- bit 9-8 **保留:** 保持为0
- bit 7 **保留:** 保持为1
- bit 6-5 **未实现:** 读为1
- bit 4 **ALT12C2:** 备用I2C2引脚映射位  
 1 = SCL2/SDA2引脚的默认位置  
 0 = SCL2/SDA2引脚的备用位置 (ASCL2/ASDA2)
- bit 3 **ALT12C1:** 备用I2C1引脚映射位  
 1 = SCL1/SDA1引脚的默认位置  
 0 = SCL1/SDA1引脚的备用位置 (ASCL1/ASDA1)
- bit 2 **保留:** 保持为1
- bit 1-0 **未实现:** 读为1

- 注 1:** 固定引脚选项仅适用于48引脚封装。
- 2:** 通过SMEN位 (I2CxCONL[8]) 使能SMBus模式。

# dsPIC33CK64MP105系列

寄存器 28-15: FALTREG 配置寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23						bit 16	

U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
—	CTXT4[2:0]			—	CTXT3[2:0]		
bit 15							bit 8

U-1	R/PO-1	R/PO-1	R/PO-1	U-1	R/PO-1	R/PO-1	R/PO-1
—	CTXT2[2:0]			—	CTXT1[2:0]		
bit 7				bit 0			

图注:	PO = 一次编程位		
R = 可读位	W = 可写位	U = 未实现位, 读为0	
-n = 擦除值	1 = 置1	0 = 清零	x = 未知

- bit 23-15      **未实现:** 读为1
- bit 14-12      **CTXT4[2:0]:** 指定备用工作寄存器组4的中断优先级 (IPL) 的位  
 111 = 未分配  
 110 = 为备用寄存器组4分配 IPL 7  
 101 = 为备用寄存器组4分配 IPL 6  
 100 = 为备用寄存器组4分配 IPL 5  
 011 = 为备用寄存器组4分配 IPL 4  
 010 = 为备用寄存器组4分配 IPL 3  
 001 = 为备用寄存器组4分配 IPL 2  
 000 = 为备用寄存器组4分配 IPL 1
- bit 11      **未实现:** 读为1
- bit 10-8      **CTXT3[2:0]:** 指定备用工作寄存器组3的中断优先级 (IPL) 的位  
 111 = 未分配  
 110 = 为备用寄存器组3分配 IPL 7  
 101 = 为备用寄存器组3分配 IPL 6  
 100 = 为备用寄存器组3分配 IPL 5  
 011 = 为备用寄存器组3分配 IPL 4  
 010 = 为备用寄存器组3分配 IPL 3  
 001 = 为备用寄存器组3分配 IPL 2  
 000 = 为备用寄存器组3分配 IPL 1
- bit 7      **未实现:** 读为1
- bit 6-4      **CTXT2[2:0]:** 指定备用工作寄存器组2的中断优先级 (IPL) 的位  
 111 = 未分配  
 110 = 为备用寄存器组2分配 IPL 7  
 101 = 为备用寄存器组2分配 IPL 6  
 100 = 为备用寄存器组2分配 IPL 5  
 011 = 为备用寄存器组2分配 IPL 4  
 010 = 为备用寄存器组2分配 IPL 3  
 001 = 为备用寄存器组2分配 IPL 2  
 000 = 为备用寄存器组2分配 IPL 1
- bit 3      **未实现:** 读为1

## 寄存器 28-15: FALTREG 配置寄存器 (续)

bit 2-0      **CTXT1[2:0]:** 指定备用工作寄存器组 1 的中断优先级 (IPL) 的位

111 = 未分配

110 = 为备用寄存器组 1 分配 IPL 7

101 = 为备用寄存器组 1 分配 IPL 6

100 = 为备用寄存器组 1 分配 IPL 5

011 = 为备用寄存器组 1 分配 IPL 4

010 = 为备用寄存器组 1 分配 IPL 3

001 = 为备用寄存器组 1 分配 IPL 2

000 = 为备用寄存器组 1 分配 IPL 1

# dsPIC33CK64MP105系列

## 28.2 器件标识

dsPIC33CK64MP105 器件在配置存储空间末尾处附近有两个标识寄存器，分别存储器件ID（DEVID）和器

件版本（DEVREV）。这两个寄存器用于确定器件的掩码、型号和生产信息。这两个寄存器为只读寄存器，如寄存器 28-16 和寄存器 28-17 所示。

寄存器 28-16: DEVREV: 器件版本寄存器

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23				bit 16			
U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 15				bit 8			
U-0	U-0	U-0	U-0	R	R	R	R
—	—	—	—	DEVREV[3:0]			
bit 7				bit 0			

图注:							
R = 只读位		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 23-4      未实现：读为0  
bit 3-0      DEVREV[3:0]: 器件版本位



# dsPIC33CK64MP105系列

**寄存器 28-17: DEVID: 器件ID寄存器**

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
—	—	—	—	—	—	—	—
bit 23							bit 16
R-1	R-0	R-0	R-0	R-1	R-1	R-1	R-0
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8
R	R	R	R	R	R	R	R
DEV7 <sup>(1)</sup>	DEV6 <sup>(1)</sup>	DEV5 <sup>(1)</sup>	DEV4 <sup>(1)</sup>	DEV3 <sup>(1)</sup>	DEV2 <sup>(1)</sup>	DEV1 <sup>(1)</sup>	DEV0 <sup>(1)</sup>
bit 7							bit 0

**图注:**

R = 只读位

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 23-16      **未实现:** 读为0

bit 15-8      **FAMID[7:0]:** 器件系列标识符位  
1000 1110 = dsPIC33CK64MP105系列

bit 7-0      **DEV[7:0]:** 单个器件标识符位<sup>(1)</sup>

**注 1:** 关于器件标识符位列表, 请参见表28-3。

**表 28-3: dsPIC33CK64MP105系列的器件ID**

器件	DEVID
dsPIC33CK64MP105	0x8E12
dsPIC33CK64MP103	0x8E11
dsPIC33CK64MP102	0x8E10
dsPIC33CK32MP105	0x8E02
dsPIC33CK32MP103	0x8E01
dsPIC33CK32MP102	0x8E00

# dsPIC33CK64MP105系列

## 28.3 用户OTP存储区

dsPIC33CK64MP105系列器件包含64个可一次性编程（OTP）双字，地址范围为801700h至8017FEh。每个48位OTP双字都只能写入一次。OTP字可用于存储校验和、代码版本、制造日期、制造批号和任何其他特定于应用的信息。

任何擦除命令都不会清除OTP存储区。该存储区只能写入一次。

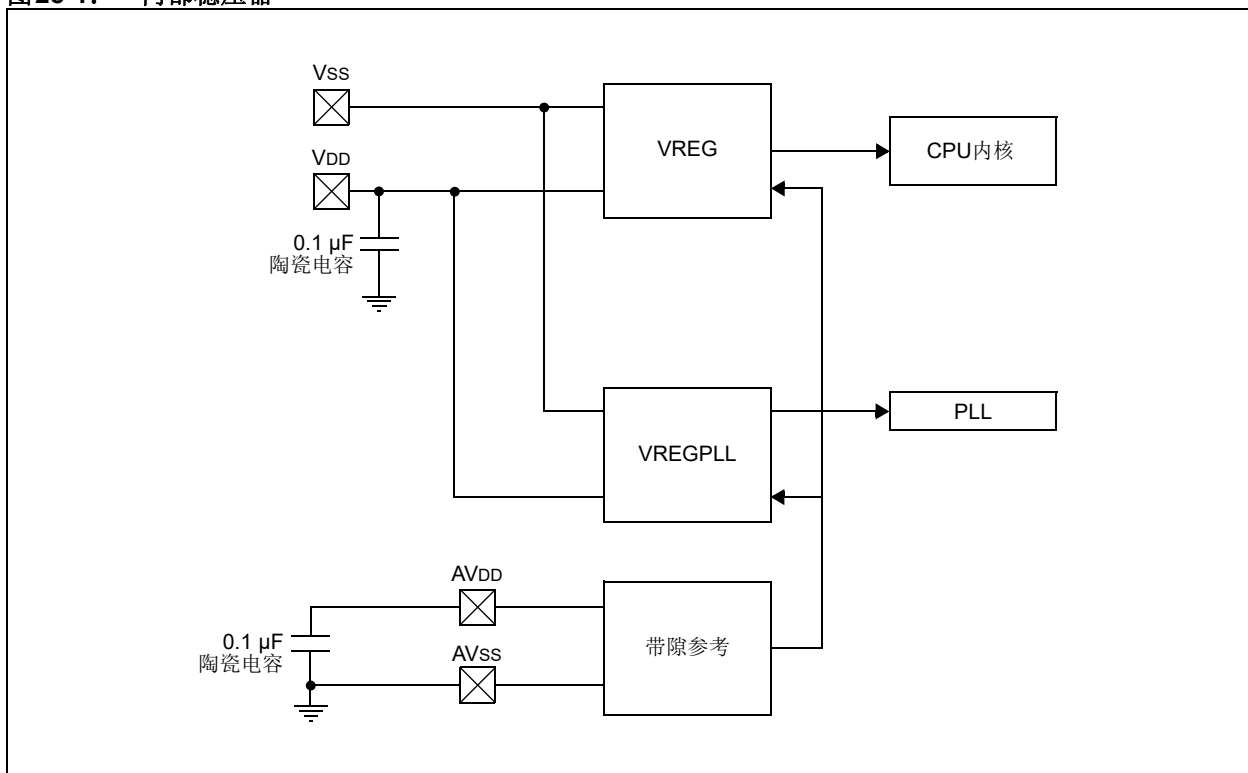
## 28.4 片上稳压器

dsPIC33CK64MP105系列器件具有一个无电容的内部稳压器，用于为内核供电（典型值为1.2V）。稳压器VREG为内核供电。PLL使用单独的稳压器VREGPLL供电，如图28-1所示。稳压器具有低功耗模式和待机模式，可在休眠模式下使用。有关休眠的更多信息，请参见第27.2.1节“休眠模式”。

当稳压器处于低功耗模式（LPWREN = 1）时，内核的可用功率受到限制。

在LPWREN位置1之前，应通过禁止外设和降低CPU频率（例如，使用不带PLL的8 MHz FRC）将器件置于低功耗状态。两个稳压器的输出电压均可由用户单独控制，从而能够在休眠模式下进一步节省功耗。

图28-1： 内部稳压器



## 寄存器 28-18: VREGCON: 稳压器控制寄存器

R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
LPWREN <sup>(1)</sup>	—	—	—	—	—	—	—
bit 15							bit 8

U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
—	—	VREG3OV1	VREG3OV0	—	—	VREG1OV1	VREG1OV0
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 15      **LPWREN:** 低功耗模式使能位<sup>(1)</sup>  
             1 = 稳压器处于低功耗模式  
             0 = 稳压器处于全功耗模式
- bit 14-6    **未实现:** 读为0
- bit 5-4      **VREG3OV[1:0]:** VREGPLL 电压控制位  
             11/00 =  $V_{OUT} = 1.5 * V_{BG} = 1.2V$   
             10 =  $V_{OUT} = 1.25 * V_{BG} = 1.0V$   
             01 =  $V_{OUT} = V_{BG} = 0.8V$
- bit 3-2      **未实现:** 读为0
- bit 1-0      **VREG1OV[1:0]:** VREG 电压控制位  
             11/00 =  $V_{OUT} = 1.5 * V_{BG} = 1.2V$   
             10 =  $V_{OUT} = 1.25 * V_{BG} = 1.0V$   
             01 =  $V_{OUT} = V_{BG} = 0.8V$

**注 1:** 低功耗模式只能在工业温度范围内使用。将该位置1前, CPU应当在低速(8 MHz或更低)下运行。

## 28.5 欠压复位（BOR）

欠压复位（BOR）模块基于内部参考电压电路，该电路监视经过稳压的电源电压。BOR模块的主要用途是在发生欠压条件时产生器件复位。欠压条件通常由交流电源线上的毛刺（例如，由于电力传输线路不良造成的交流周期波形部分丢失，或者由于接入大感性负载时电流消耗过大造成电压骤降）产生。

BOR将产生用于复位器件的复位脉冲。BOR会根据器件配置位的选择来选择时钟源。

如果选择了振荡器模式，BOR将激活振荡器起振定时器（Oscillator Start-up Timer, OST）。系统时钟将保持到OST超时。如果使用了PLL，则时钟将被保持到LOCK位（OSCCON[5]）为1。

同时，将在内部复位释放前应用PWRT延时（TPWRT）。如果TPWRT = 0且使用了晶振，那么会应用TFSCM的标称延时。这种情况下总的延时为TFSCM。关于具体的TFSCM值，请参见第31.0节“电气特性”的表31-26中的参数SY35。

BOR状态位（RCON[1]）置1，指示发生了BOR。BOR电路在休眠或空闲模式下继续工作，当VDD下降到BOR阈值电压以下时将复位器件。

## 28.6 双看门狗定时器（WDT）

**注 1：** 本数据手册总结了 dsPIC33CK64MP105 系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《dsPIC33/PIC24 系列参考手册》中的“双看门狗定时器”（[www.microchip.com/DS70005250](http://www.microchip.com/DS70005250)）。

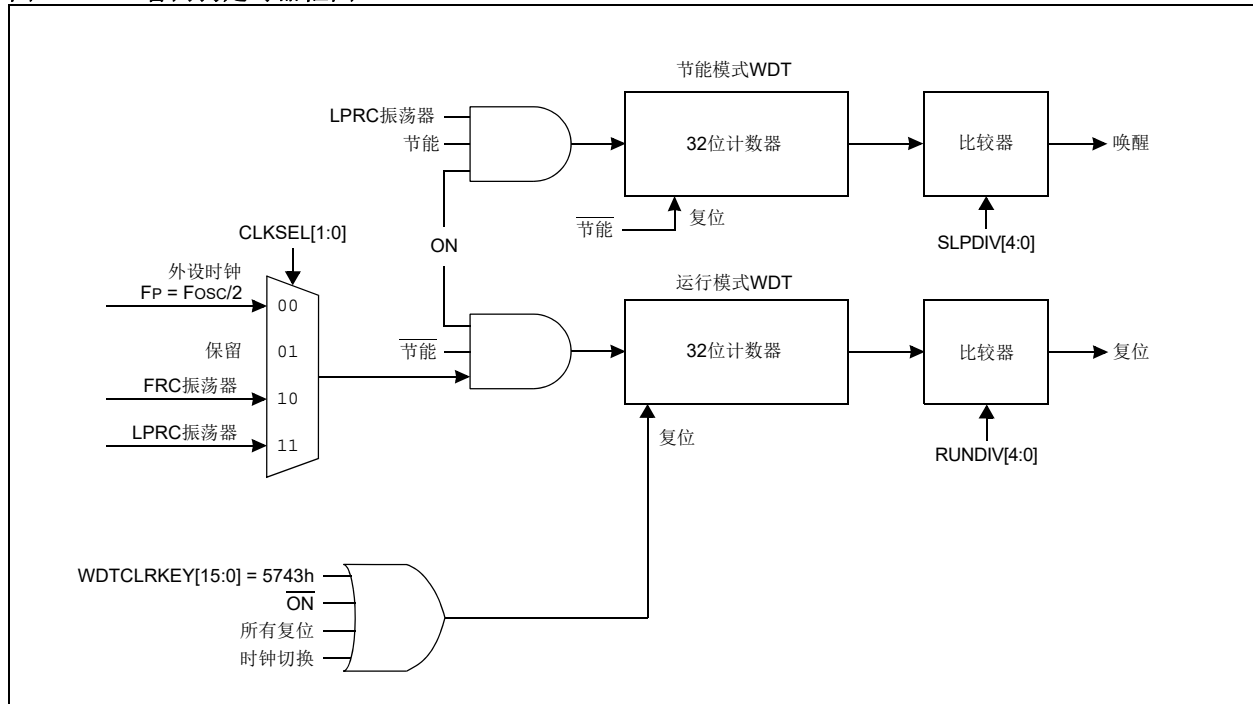
本章将介绍 dsPIC33 双看门狗定时器（WDT）。有关 WDT 框图，请参见图 28-2。

WDT 使能后，将依靠内部低功耗 RC（LPRC）振荡器时钟源或可选的运行模式时钟源工作。如果软件未定期清零 WDT，则可使用 WDT 通过复位器件来检测系统软件故障。WDT 可配置为窗口模式或非窗口模式。可使用 WDT 后分频器选择各种 WDT 超时周期。WDT 还可用于将器件从休眠或空闲模式（节能模式）唤醒。如果 WDT 超时并发出器件复位信号，则 RCON 的 WDTO 位（寄存器 6-1）将置 1。

以下是 WDT 模块的一些主要特性：

- 可配置或由软件控制
- 运行和休眠/空闲模式具有独立的、用户可配置超时周期
- 可将器件从休眠或空闲模式唤醒
- 用户可选择运行模式时钟源
- 休眠/空闲模式下依靠 LPRC 工作

图 28-2： 看门狗定时器框图



# dsPIC33CK64MP105系列

寄存器 28-19: WDTCONL: 看门狗定时器控制低位字寄存器

R/W-0	U-0	U-0	R-y	R-y	R-y	R-y	R-y
ON <sup>(1,2)</sup>	—	—	RUNDIV4 <sup>(3)</sup>	RUNDIV3 <sup>(3)</sup>	RUNDIV2 <sup>(3)</sup>	RUNDIV1 <sup>(3)</sup>	RUNDIV0 <sup>(3)</sup>
bit 15							bit 8

R	R	R-y	R-y	R-y	R-y	R-y	HS/R/W-0
CLKSEL1 <sup>(3,5)</sup>	CLKSEL0 <sup>(3,5)</sup>	SLPDIV4 <sup>(3)</sup>	SLPDIV3 <sup>(3)</sup>	SLPDIV2 <sup>(3)</sup>	SLPDIV1 <sup>(3)</sup>	SLPDIV0 <sup>(3)</sup>	WDTWINEN <sup>(4)</sup>
bit 7							bit 0

图注:	HS = 硬件置1位	y = 在POR时由配置位设置的值
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 15 **ON:** 看门狗定时器使能位<sup>(1,2)</sup>  
 1 = 如果器件配置未使能看门狗定时器, 则将其使能  
 0 = 如果在软件中使能了看门狗定时器, 则将其禁止
- bit 14-13 **未实现:** 读为0
- bit 12-8 **RUNDIV[4:0]:** 休眠和空闲模式WDT后分频比状态位<sup>(3)</sup>  
 11111 =  $2^{31}$  (= 2,147,483,648) 分频  
 11110 =  $2^{30}$  (= 1,073,741,824) 分频  
 ...  
 00001 =  $2^1$  (= 2) 分频  
 00000 =  $2^0$  (= 1) 分频
- bit 7-6 **CLKSEL[1:0]WDT** 运行模式时钟选择状态位<sup>(3,5)</sup>  
 11 = LPRC振荡器  
 10 = FRC振荡器  
 01 = 保留  
 00 = SYSCLK
- bit 5-1 **SLPDIV[4:0]:** 休眠和空闲模式WDT后分频比状态位<sup>(3)</sup>  
 11111 =  $2^{31}$  (= 2,147,483,648) 分频  
 11110 =  $2^{30}$  (= 1,073,741,824) 分频  
 ...  
 00001 =  $2^1$  (= 2) 分频  
 00000 =  $2^0$  (= 1) 分频
- bit 0 **WDTWINEN:** 看门狗定时器窗口使能位<sup>(4)</sup>  
 1 = 使能窗口模式  
 0 = 禁止窗口模式

- 注 1: 如果WDT由器件配置或软件使能, 则读取此位将得到1。
- 2: 在清零模块ON位的指令之后, 用户软件不应立即读/写外设的SFR。
- 3: 这些位反映配置位的值。
- 4: 配置位置1时, WDTWINEN位将反映配置位的状态。配置位清零时, 该位的值由软件控制。
- 5: 可用时钟源取决于器件。

寄存器 28-20: **WDTCONH**: 看门狗定时器控制高位字寄存器

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
WDTCLRKEY[15:8]							
bit 15				bit 8			

W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
WDTCLRKEY[7:0]							
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 15-0 **WDTCLRKEY[15:0]**: 看门狗定时器清零密钥位

要清零看门狗定时器以防止发生超时, 软件必须使用单个16位写操作向该存储单元写入值0x5743。

# dsPIC33CK64MP105系列

---

## 28.7 JTAG 接口

dsPIC33CK64MP105系列器件实现了一个JTAG接口，该接口支持边界扫描器件测试。关于该接口的详细信息将会在本文档以后的版本中提供。

**注：** 关于JTAG接口使用、配置和操作方面的更多信息，请参见《dsPIC33/PIC24系列参考手册》中的“编程和诊断”（[www.microchip.com/DS70608](http://www.microchip.com/DS70608)）。

## 28.8 在线串行编程（ICSP）

可以在最终的应用电路中对dsPIC33CK64MP105系列器件进行串行编程。只需要5根线即可实现这一操作，其中时钟线、数据线各一根，其余3根分别是电源线、地线和编程序列线。串行编程允许用户在生产电路板时使用未编程器件，而仅在产品交付之前才对器件进行编程，从而可以使用最新版本的固件或者定制固件进行编程。关于在线串行编程（ICSP）的详细信息，请参见“*dsPIC33CK64MP105 Family Flash Programming Specification*”（DS70005352）。

可使用3对编程时钟/数据引脚中的任意一对：

- PGC1 和 PGD1
- PGC2 和 PGD2
- PGC3 和 PGD3

## 28.9 在线调试器

当选择MPLAB工具作为调试器时，将使能在线调试功能。该功能允许与MPLAB IDE配合使用来进行简单的调试。通过PGCx（仿真/调试时钟）和PGDx（仿真/调试数据）引脚功能控制调试功能。

可使用3对调试时钟/数据引脚中的任意一对：

- PGC1 和 PGD1
- PGC2 和 PGD2
- PGC3 和 PGD3

要使用器件的在线调试功能，就必须在设计中对MCLR、VDD、VSS和PGCx/PGDx引脚对进行ICSP连接。此外，当使能该功能时，某些资源就不能用于一般用途了。这些资源包括数据RAM的前80字节和两个I/O引脚（PGCx和PGDx）。



28.10 代码保护和CodeGuard安全性

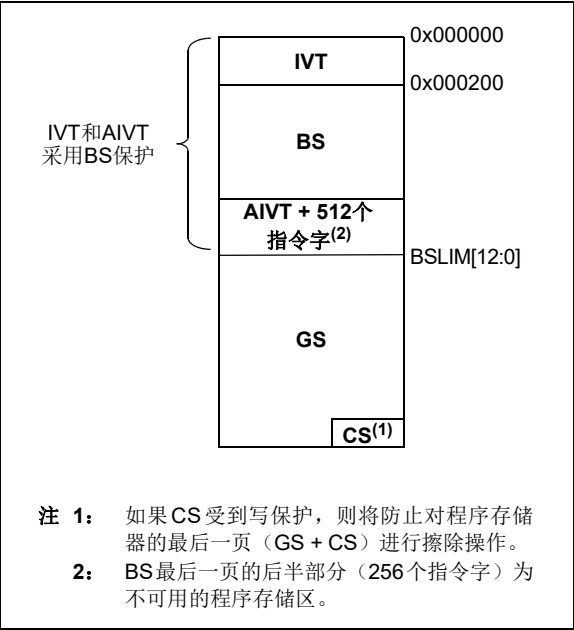
dsPIC33CK64MP105系列器件提供了多个安全级别，用于保护各方的知识产权。闪存程序存储器根据保护级别可分为三个段：引导段（BS）、通用段（General Segment, GS）和配置段（Configuration Segment, CS）。引导段的安全权限最高，可被认为在访问其他段时的限制是有限的。通用段的安全权限最低，用于最终用户的系统代码。配置段仅包含器件的用户配置数据，位于程序存储空间的末尾。

代码保护功能由配置寄存器FSEC和FBSLIM控制。FSEC寄存器用于控制各个段的代码保护级别以及段是否被写保护。BS和GS的大小将取决于BSLIM[12:0]位设置以及是否使能了备用中断向量表（AIVT）。BSLIM[12:0]位用于定义BS的页数，其中每页包含1024个指令字。最小的BS大小为一页，其中包含中断向量表（IVT）和代码保护的512个指令字。

如果使能了AIVT，则BS的最后一页将包含AIVT，而不包含任何BS代码。使能AIVT时，最小的BS大小为两页（2048个指令字），其中一页用于IVT和BS代码，另一页用于AIVT。BS的写保护不包含AIVT。BS的最后一页始终可被BS代码编程或擦除。通用段将从下一页开始，占用程序闪存中除闪存配置字以外的剩余部分。只有在未使能BS时，IVT才将采用GS安全性。当BS和GS中任一安全段已使能写保护时，将防止对IVT进行编程或页擦除操作。

图28-3给出了器件的各个安全段。图中显示了全部三个段，但并非都是必需的。如果只需要基本代码保护，则可以根据需要单独使能GS或者组合使能GS与CS。

图28-3：安全段示例



注:

## 29.0 指令集汇总

**注：** 本数据手册总结了dsPIC33CK64MP105系列器件的特性。但是不应把本手册当作无所不包的参考资料来使用。如需了解本数据手册的补充信息，请参见《16位MCU和DSC程序员参考手册》（[www.microchip.com/DS70000157](http://www.microchip.com/DS70000157)），该文档可从Microchip网站（[www.microchip.com](http://www.microchip.com)）下载。

dsPIC33CK64MP105系列的指令集与dsPIC30F和dsPIC33F系列器件的指令集几乎完全相同。

大部分指令的长度为一个程序存储字（24位）。只有三条指令需要两个程序存储单元。

每一条单字指令的长度为24位，分为一个指定指令类型的8位操作码和进一步指定指令操作的一个或多个操作数。

指令集是高度正交的，分为5个基本类别：

- 针对字或字节的操作
- 针对位的操作
- 立即数操作
- DSP操作
- 控制操作

表29-1列出了在说明指令时使用的通用符号。

表29-2是dsPIC33指令集的汇总，列出了所有指令以及每条指令影响的状态标志。

大多数针对字或字节的W寄存器指令（包括桶形移位指令）有三个操作数：

- 第一个源操作数，通常是寄存器Wb，不带任何地址修改量
- 第二个源操作数，通常是寄存器Ws，带或不带地址修改量
- 保存结果的目标寄存器，通常是寄存器Wd，带或不带地址修改量

不过，针对字或字节的文件寄存器指令只有两个操作数：

- 文件寄存器，由f值指定
- 目标寄存器，可以是文件寄存器f或W0寄存器（用WREG表示）

大多数位操作类指令（包括简单的循环/移位指令）有两个操作数：

- W寄存器（带或不带地址修改量）或文件寄存器（由Ws或f的值指定）
- W寄存器或文件寄存器中的位（由一个立即数值指定，或者由寄存器Wb的内容间接指定）

涉及数据传送的立即数指令，可以使用下列操作数：

- 要被装入到W寄存器或文件寄存器中的立即数（由k指定）
- 要装入立即数的W寄存器或文件寄存器（由Wb或f指定）

然而，涉及算术或逻辑运算的立即数指令，使用如下的操作数：

- 第一个源操作数是寄存器Wb，不带任何地址修改量
- 第二个源操作数是立即数
- 操作结果的目标寄存器（仅在与第一个源操作数不同时）通常是寄存器Wd（带或不带地址修改量）

MAC类DSP指令可使用下列操作数：

- 要使用的累加器（A或B）（必需的操作数）
- 要用作两个操作数的W寄存器
- X和Y地址空间预取操作
- X和Y地址空间预取目标寄存器
- 累加器回写目标寄存器

与乘法无关的其他DSP指令使用的操作数可能包括：

- 要使用的累加器（必需）
- 源操作数或目标操作数（分别由Wso或Wdo指定），带或不带地址修改量
- 移位位数，由W寄存器Wn或立即数指定

控制指令可以使用下列操作数：

- 程序存储器地址
- 表读和表写指令的模式

# dsPIC33CK64MP105系列

大多数指令都是单字指令。设计了一些双字指令，用48位来提供全部所需信息。在第二个字中，高8位全为0。如果将第二个字单独作为一条指令来执行的话，它将作为一条NOP指令来执行。

双字指令执行需要两个指令周期。

大多数单字长指令都在一个指令周期内执行，除非条件测试为真、指令执行结果改变了程序计数器或者执行了PSV或表读操作。在这些情况下，指令执行需要多个指令周期，在额外的指令周期中执行NOP指令。某些可能

涉及到跳过后续指令的指令，如果要执行跳过的话，可能需要两个或三个周期，这取决于被跳过的指令是单字还是双字指令。此外，双字传送需要两个周期。

**注：** 与dsPIC30F、dsPIC33F、PIC24F和PIC24H器件相比，在dsPIC33CK64MP105器件中，对非CPU特殊功能寄存器执行读操作和读-修改-写操作时需要一个额外的周期。

**注：** 关于指令集的更多详细信息，请参见《16位MCU和DSC程序员参考手册》（[www.microchip.com/DS70000157](http://www.microchip.com/DS70000157)）。

表29-1： 操作码说明中使用的符号

字段	说明
#text	表示由text定义的立即数
(text)	表示text的内容
[text]	表示由text寻址的存储单元
{ }	可选字段或操作
$a \in \{b, c, d\}$	a从值b、c和d的集合中选择
[n:m]	寄存器位域
.b	字节模式选择
.d	双字模式选择
.S	影子寄存器选择
.w	字模式选择（默认）
Acc	两个累加器{A, B}之一
AWB	累加器回写目标地址寄存器 $\in \{W13, [W13]+2\}$
bit4	4位选择字段（用于字寻址指令） $\in \{0...15\}$
C, DC, N, OV, Z	MCU状态位：进位、半进位、负、溢出和全零标志位
Expr	绝对地址、标号或表达式（由链接器解析）
f	文件寄存器地址 $\in \{0x0000...0x1FFF\}$
lit1	1位无符号立即数 $\in \{0,1\}$
lit4	4位无符号立即数 $\in \{0...15\}$
lit5	5位无符号立即数 $\in \{0...31\}$
lit8	8位无符号立即数 $\in \{0...255\}$
lit10	10位无符号立即数 $\in \{0...255\}$ （对于字节模式）或 $\{0:1023\}$ （对于字模式）
lit14	14位无符号立即数 $\in \{0...16384\}$
lit16	16位无符号立即数 $\in \{0...65535\}$
lit23	23位无符号立即数 $\in \{0...8388608\}$ ；LSb必须为0
无	字段无需内容，可为空
OA, OB, SA, SB	DSP状态位：ACCA溢出、ACCB溢出、ACCA饱和和ACCB饱和
PC	程序计数器
Slit10	10位有符号立即数 $\in \{-512...511\}$
Slit16	16位有符号立即数 $\in \{-32768...32767\}$
Slit6	6位有符号立即数 $\in \{-16...16\}$
Wb	基本W寄存器 $\in \{W0...W15\}$
Wd	目标W寄存器 $\in \{Wd, [Wd], [Wd++] , [Wd--], [++Wd], [--Wd]\}$
Wdo	目标W寄存器 $\in \{Wnd, [Wnd], [Wnd++] , [Wnd--], [++Wnd], [--Wnd], [Wnd+Wb]\}$
Wm, Wn	被除数和除数工作寄存器对（直接寻址）

表29-1: 操作码说明中使用的符号(续)

字段	说明
Wm*Wm	用于平方指令的被乘数和乘数工作寄存器对 $\in \{W4 * W4, W5 * W5, W6 * W6, W7 * W7\}$
Wm*Wn	用于DSP指令的被乘数和乘数工作寄存器对 $\in \{W4 * W5, W4 * W6, W4 * W7, W5 * W6, W5 * W7, W6 * W7\}$
Wn	16个工作寄存器之一 $\in \{W0...W15\}$
Wnd	16个目标工作寄存器之一 $\in \{W0...W15\}$
Wns	16个源工作寄存器之一 $\in \{W0...W15\}$
WREG	W0 (文件寄存器指令中使用的工作寄存器)
Ws	源W寄存器 $\in \{Ws, [Ws], [Ws++] , [Ws--], [++Ws], [--Ws]\}$
Wso	源W寄存器 $\in \{Wns, [Wns], [Wns++] , [Wns--], [++Wns], [--Wns], [Wns+Wb]\}$
Wx	用于DSP指令的X数据空间预取地址寄存器 $\in \{[W8] + 6, [W8] + 4, [W8] + 2, [W8], [W8] - 6, [W8] - 4, [W8] - 2, [W9] + 6, [W9] + 4, [W9] + 2, [W9], [W9] - 6, [W9] - 4, [W9] - 2, [W9 + W12], \text{无}\}$
Wxd	用于DSP指令的X数据空间预取目标寄存器 $\in \{W4...W7\}$
Wy	用于DSP指令的Y数据空间预取地址寄存器 $\in \{[W10] + 6, [W10] + 4, [W10] + 2, [W10], [W10] - 6, [W10] - 4, [W10] - 2, [W11] + 6, [W11] + 4, [W11] + 2, [W11], [W11] - 6, [W11] - 4, [W11] - 2, [W11 + W12], \text{无}\}$
Wyd	用于DSP指令的Y数据空间预取目标寄存器 $\in \{W4...W7\}$

# dsPIC33CK64MP105系列

表 29-2: 指令集概述

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
1	ADD	ADD Acc	累加器相加	1	1	OA,OB,SA,SB
		ADD f	$f = f + WREG$	1	1	C,DC,N,OV,Z
		ADD f, WREG	$WREG = f + WREG$	1	1	C,DC,N,OV,Z
		ADD #lit10, Wn	$Wd = lit10 + Wd$	1	1	C,DC,N,OV,Z
		ADD Wb, Ws, Wd	$Wd = Wb + Ws$	1	1	C,DC,N,OV,Z
		ADD Wb, #lit5, Wd	$Wd = Wb + lit5$	1	1	C,DC,N,OV,Z
		ADD Wso, #Slit4, Acc	将 16 位有符号立即数加到累加器	1	1	OA,OB,SA,SB
2	ADDC	ADDC f	$f = f + WREG + (C)$	1	1	C,DC,N,OV,Z
		ADDC f, WREG	$WREG = f + WREG + (C)$	1	1	C,DC,N,OV,Z
		ADDC #lit10, Wn	$Wd = lit10 + Wd + (C)$	1	1	C,DC,N,OV,Z
		ADDC Wb, Ws, Wd	$Wd = Wb + Ws + (C)$	1	1	C,DC,N,OV,Z
		ADDC Wb, #lit5, Wd	$Wd = Wb + lit5 + (C)$	1	1	C,DC,N,OV,Z
3	AND	AND f	$f = f .AND. WREG$	1	1	N,Z
		AND f, WREG	$WREG = f .AND. WREG$	1	1	N,Z
		AND #lit10, Wn	$Wd = lit10 .AND. Wd$	1	1	N,Z
		AND Wb, Ws, Wd	$Wd = Wb .AND. Ws$	1	1	N,Z
		AND Wb, #lit5, Wd	$Wd = Wb .AND. lit5$	1	1	N,Z
4	ASR	ASR f	$f = \text{算术右移 } f$	1	1	C,N,OV,Z
		ASR f, WREG	$WREG = \text{算术右移 } f$	1	1	C,N,OV,Z
		ASR Ws, Wd	$Wd = \text{算术右移 } Ws$	1	1	C,N,OV,Z
		ASR Wb, Wns, Wnd	$Wnd = \text{将 } Wb \text{ 算术右移 } Wns \text{ 位}$	1	1	N,Z
		ASR Wb, #lit5, Wnd	$Wnd = \text{将 } Wb \text{ 算术右移 } lit5 \text{ 位}$	1	1	N,Z
5	BCLR	BCLR f, #bit4	将 f 中的指定位清零	1	1	无
		BCLR Ws, #bit4	将 Ws 中的指定位清零	1	1	无
6	BFEXT	BFEXT bit4, wid5, Ws, Wb	从 Ws 中提取位域, 写入 Wb	2	2	无
		BFEXT bit4, wid5, f, Wb	从 f 中提取位域, 写入 Wb	2	2	无
7	BFINS	BFINS bit4, wid5, Wb, Ws	将 Wb 中的位域插入 Ws	2	2	无
		BFINS bit4, wid5, Wb, f	将 Wb 中的位域插入 f	2	2	无
		BFINS bit4, wid5, lit8, Ws	将 #lit8 中的位域插入 Ws	2	2	无

注 1: 对非 CPU SFR 执行读操作和读-修改-写操作 (例如, 位操作和逻辑运算) 时需要一个额外的指令周期。

2: 除法指令之前必须有一条 “REPEAT #5” 指令, 以便能够连续执行 6 次。

表 29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
9	BRA	BRA C, Expr	如果有进位则跳转	1	1 (4)	无
		BRA GE, Expr	如果大于或等于则跳转	1	1 (4)	无
		BRA GEU, Expr	如果无符号大于或等于则跳转	1	1 (4)	无
		BRA GT, Expr	如果大于则跳转	1	1 (4)	无
		BRA GTU, Expr	如果无符号大于则跳转	1	1 (4)	无
		BRA LE, Expr	如果小于或等于则跳转	1	1 (4)	无
		BRA LEU, Expr	如果无符号小于或等于则跳转	1	1 (4)	无
		BRA LT, Expr	如果小于则跳转	1	1 (4)	无
		BRA LTU, Expr	如果无符号小于则跳转	1	1 (4)	无
		BRA N, Expr	如果为负则跳转	1	1 (4)	无
		BRA NC, Expr	如果没有进位则跳转	1	1 (4)	无
		BRA NN, Expr	如果不为负则跳转	1	1 (4)	无
		BRA NOV, Expr	如果未溢出则跳转	1	1 (4)	无
		BRA NZ, Expr	如果不为零则跳转	1	1 (4)	无
		BRA OA, Expr	如果累加器A溢出则跳转	1	1 (4)	无
		BRA OB, Expr	如果累加器B溢出则跳转	1	1 (4)	无
		BRA OV, Expr	如果溢出则跳转	1	1 (4)	无
		BRA SA, Expr	如果累加器A饱和则跳转	1	1 (4)	无
		BRA SB, Expr	如果累加器B饱和则跳转	1	1 (4)	无
		BRA Expr	无条件跳转	1	4	无
		BRA Z, Expr	如果为零则跳转	1	1 (4)	无
		BRA Wn	计算跳转	1	4	无
10	BREAK	BREAK	停止用户代码执行	1	1	无
11	BSET	BSET f, #bit4	将f中的指定位置1	1	1	无
		BSET Ws, #bit4	将Ws中的指定位置1	1	1	无
12	BSW	BSW.C Ws, Wb	将C位内容写入Ws[Wb]	1	1	无
		BSW.Z Ws, Wb	将Z位内容取反写入Ws[Wb]	1	1	无
13	BTG	BTG f, #bit4	将f中的指定位取反	1	1	无
		BTG Ws, #bit4	将Ws中的指定位取反	1	1	无
14	BTSC	BTSC f, #bit4	对f中的指定位进行测试, 如果为零则跳过	1	1 (2或3)	无
		BTSC Ws, #bit4	对Ws中的指定位进行测试, 如果为零则跳过	1	1 (2或3)	无
15	BTSS	BTSS f, #bit4	对f中的指定位进行测试, 如果为1则跳过	1	1 (2或3)	无
		BTSS Ws, #bit4	对Ws中的指定位进行测试, 如果为1则跳过	1	1 (2或3)	无
16	BTST	BTST f, #bit4	对f中的指定位进行测试	1	1	Z
		BTST.C Ws, #bit4	对Ws中的指定位进行测试, 并将其值存储到C	1	1	C
		BTST.Z Ws, #bit4	对Ws中的指定位进行测试, 并将其反码存储到Z	1	1	Z
		BTST.C Ws, Wb	对Ws[Wb]位进行测试, 并将其值存储到C	1	1	C
		BTST.Z Ws, Wb	对Ws[Wb]位进行测试, 并将其反码存储到Z	1	1	Z
17	BTSTS	BTSTS f, #bit4	对f中的指定位进行测试, 并将f中的该位置1	1	1	Z
		BTSTS.C Ws, #bit4	对Ws中的指定位进行测试, 并将其值存储到C, 然后将Ws中的该位置1	1	1	C
		BTSTS.Z Ws, #bit4	对Ws中的指定位进行测试, 并将其反码存储到Z, 然后将Ws中的该位置1	1	1	Z
18	CALL	CALL lit23	调用子程序	2	4	SFA
		CALL Wn	间接调用子程序	1	4	SFA
		CALL.L Wn	间接调用子程序 (长地址)	1	4	SFA
19	CLR	CLR f	f = 0x0000	1	1	无
		CLR WREG	WREG = 0x0000	1	1	无
		CLR Ws	Ws = 0x0000	1	1	无
		CLR Acc, Wx, Wxd, Wy, Wyd, AWB	将累加器清零	1	1	OA, OB, SA, SB

注 1: 对非CPU SFR执行读操作和读-修改-写操作 (例如, 位操作和逻辑运算) 时需要一个额外的指令周期。

2: 除法指令之前必须有一条“REPEAT #5”指令, 以便能够连续执行6次。

# dsPIC33CK64MP105系列

表29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
20	CLRWDT	CLRWDT	将看门狗定时器清零	1	1	WDTO, Sleep
21	COM	COM $f$	$f = \bar{f}$	1	1	N, Z
		COM $f, WREG$	$WREG = \bar{f}$	1	1	N, Z
		COM $Ws, Wd$	$Wd = \bar{Ws}$	1	1	N, Z
22	CP	CP $f$	比较 $f$ 和 WREG	1	1	C, DC, N, OV, Z
		CP $Wb, \#lit8$	比较 $Wb$ 和 $lit8$	1	1	C, DC, N, OV, Z
		CP $Wb, Ws$	比较 $Wb$ 和 $Ws$ ( $Wb - Ws$ )	1	1	C, DC, N, OV, Z
23	CP0	CP0 $f$	比较 $f$ 和 0x0000	1	1	C, DC, N, OV, Z
		CP0 $Ws$	比较 $Ws$ 和 0x0000	1	1	C, DC, N, OV, Z
24	CPB	CPB $f$	带借位比较 $f$ 和 WREG	1	1	C, DC, N, OV, Z
		CPB $Wb, \#lit8$	带借位比较 $Wb$ 和 $lit8$	1	1	C, DC, N, OV, Z
		CPB $Wb, Ws$	带借位比较 $Wb$ 和 $Ws$ ( $Wb - Ws - C$ )	1	1	C, DC, N, OV, Z
25	CPSEQ	CPSEQ $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果相等则跳过	1	1 (2或3)	无
	CPBEQ	CPBEQ $Wb, Wn, Expr$	比较 $Wb$ 和 $Wn$ , 如果相等则跳转	1	1 (5)	无
26	CPSGT	CPSGT $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果大于则跳过	1	1 (2或3)	无
	CPBGT	CPBGT $Wb, Wn, Expr$	比较 $Wb$ 和 $Wn$ , 如果大于则跳转	1	1 (5)	无
27	CPSLT	CPSLT $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果小于则跳过	1	1 (2或3)	无
		CPBLT $Wb, Wn, Expr$	比较 $Wb$ 和 $Wn$ , 如果小于则跳转	1	1 (5)	无
28	CPSNE	CPSNE $Wb, Wn$	比较 $Wb$ 和 $Wn$ , 如果 $\neq$ 则跳过	1	1 (2或3)	无
		CPBNE $Wb, Wn, Expr$	比较 $Wb$ 和 $Wn$ , 如果 $\neq$ 则跳转	1	1 (5)	无
29	CTXTSWP	CTXTSWP $\#lit3$	将 CPU 寄存器现场切换至 $lit3$ 定义的现场	1	2	无
30	CTXTSWP	CTXTSWP $Wn$	将 CPU 寄存器现场切换至 $Wn$ 定义的现场	1	2	无
31	DAW.B	DAW.B $Wn$	$Wn =$ 十进制调整 $Wn$	1	1	C
32	DEC	DEC $f$	$f = f - 1$	1	1	C, DC, N, OV, Z
		DEC $f, WREG$	$WREG = f - 1$	1	1	C, DC, N, OV, Z
		DEC $Ws, Wd$	$Wd = Ws - 1$	1	1	C, DC, N, OV, Z
33	DEC2	DEC2 $f$	$f = f - 2$	1	1	C, DC, N, OV, Z
		DEC2 $f, WREG$	$WREG = f - 2$	1	1	C, DC, N, OV, Z
		DEC2 $Ws, Wd$	$Wd = Ws - 2$	1	1	C, DC, N, OV, Z
34	DISI	DISI $\#lit14$	在 $k$ 个指令周期内禁止中断	1	1	无
35	DIVF	DIVF $Wm, Wn$	有符号 16/16 位小数除法	1	18	N, Z, C, OV
36	DIV.S <sup>(2)</sup>	DIV.S $Wm, Wn$	有符号 16/16 位整数除法	1	18	N, Z, C, OV
		DIV.SD $Wm, Wn$	有符号 32/16 位整数除法	1	18	N, Z, C, OV
37	DIV.U <sup>(2)</sup>	DIV.U $Wm, Wn$	无符号 16/16 位整数除法	1	18	N, Z, C, OV
		DIV.UD $Wm, Wn$	无符号 32/16 位整数除法	1	18	N, Z, C, OV
38	DIVF2 <sup>(2)</sup>	DIVF2 $Wm, Wn$	有符号 16/16 位小数除法 (W1:W0 保留)	1	6	N, Z, C, OV
39	DIV2.S <sup>(2)</sup>	DIV2.S $Wm, Wn$	有符号 16/16 位整数除法 (W1:W0 保留)	1	6	N, Z, C, OV
		DIV2.SD $Wm, Wn$	有符号 32/16 位整数除法 (W1:W0 保留)	1	6	N, Z, C, OV
40	DIV2.U <sup>(2)</sup>	DIV2.U $Wm, Wn$	无符号 16/16 位整数除法 (W1:W0 保留)	1	6	N, Z, C, OV
		DIV2.UD $Wm, Wn$	无符号 32/16 位整数除法 (W1:W0 保留)	1	6	N, Z, C, OV
41	DO	DO $\#lit15, Expr$	执行 DO 循环代码到 $PC + Expr$ , 执行次数为 $lit15 + 1$ 次	2	2	无
		DO $Wn, Expr$	执行 DO 循环代码到 $PC + Expr$ , 执行次数为 $(Wn) + 1$ 次	2	2	无

注 1: 对非 CPU SFR 执行读操作和读-修改-写操作 (例如, 位操作和逻辑运算) 时需要一个额外的指令周期。

2: 除法指令之前必须有一条 “REPEAT #5” 指令, 以便能够连续执行 6 次。



表 29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
42	ED	ED $Wm*Wm, Acc, Wx, Wy, Wxd$	欧几里德距离 (无累加)	1	1	OA,OB,OAB, SA,SB,SAB
43	EDAC	EDAC $Wm*Wm, Acc, Wx, Wy, Wxd$	欧几里德距离	1	1	OA,OB,OAB, SA,SB,SAB
44	EXCH	EXCH $Wns, Wnd$	交换 $Wns$ 和 $Wnd$ 的内容	1	1	无
46	FBCL	FBCL $Ws, Wnd$	从左边 (MSb) 开始查找第一个位变化	1	1	C
47	FF1L	FF1L $Ws, Wnd$	从左边 (MSb) 开始查找第一个 1	1	1	C
48	FF1R	FF1R $Ws, Wnd$	从右边 (LSb) 开始查找第一个 1	1	1	C
49	FLIM	FLIM $Wb, Ws$	强制执行数据范围限制 (上限和下限), 不保存超限结果	1	1	N,Z,OV
		FLIM.V $Wb, Ws, Wd$	强制执行数据范围限制 (上限和下限), 保存超限结果	1	1	N,Z,OV
50	GOTO	GOTO $Expr$	跳转到地址	2	4	无
		GOTO $Wn$	间接跳转到地址	1	4	无
		GOTO.L $Wn$	间接跳转到地址 (长地址)	1	4	无
51	INC	INC $f$	$f = f + 1$	1	1	C,DC,N,OV,Z
		INC $f, WREG$	$WREG = f + 1$	1	1	C,DC,N,OV,Z
		INC $Ws, Wd$	$Wd = Ws + 1$	1	1	C,DC,N,OV,Z
52	INC2	INC2 $f$	$f = f + 2$	1	1	C,DC,N,OV,Z
		INC2 $f, WREG$	$WREG = f + 2$	1	1	C,DC,N,OV,Z
		INC2 $Ws, Wd$	$Wd = Ws + 2$	1	1	C,DC,N,OV,Z
53	IOR	IOR $f$	$f = f .IOR. WREG$	1	1	N,Z
		IOR $f, WREG$	$WREG = f .IOR. WREG$	1	1	N,Z
		IOR $\#lit10, Wn$	$Wd = lit10 .IOR. Wd$	1	1	N,Z
		IOR $Wb, Ws, Wd$	$Wd = Wb .IOR. Ws$	1	1	N,Z
		IOR $Wb, \#lit5, Wd$	$Wd = Wb .IOR. lit5$	1	1	N,Z
54	LAC	LAC $Wso, \#Slit4, Acc$	装载累加器	1	1	OA,OB,OAB, SA,SB,SAB
		LAC.D $Wso, \#Slit4, Acc$	装载累加器 (双字)	1	2	OA,SA,OB,SB
56	LNK	LNK $\#lit14$	分配堆栈帧	1	1	SFA
57	LSR	LSR $f$	$f =$ 逻辑右移 $f$	1	1	C,N,OV,Z
		LSR $f, WREG$	$WREG =$ 逻辑右移 $f$	1	1	C,N,OV,Z
		LSR $Ws, Wd$	$Wd =$ 逻辑右移 $Ws$	1	1	C,N,OV,Z
		LSR $Wb, Wns, Wnd$	$Wnd =$ 将 $Wb$ 逻辑右移 $Wns$ 位	1	1	N,Z
		LSR $Wb, \#lit5, Wnd$	$Wnd =$ 将 $Wb$ 逻辑右移 $lit5$ 位	1	1	N,Z
58	MAC	MAC $Wm*Wn, Acc, Wx, Wxd, Wy, Wyd, AWB$	相乘并累加	1	1	OA,OB,OAB, SA,SB,SAB
		MAC $Wm*Wm, Acc, Wx, Wxd, Wy, Wyd$	平方并累加	1	1	OA,OB,OAB, SA,SB,SAB
59	MAX	MAX $Acc$	强制执行数据范围上限	1	1	N,OV,Z
		MAX.V $Acc, Wnd$	强制执行数据范围上限, 保存超限结果	1	1	N,OV,Z
60	MIN	MIN $Acc$	如果累加器A的内容小于累加器B, 则用累加器B的内容装载累加器A, 反之亦然	1	1	N,OV,Z
		MIN.V $Acc, Wd$	如果累加器A的内容小于累加器B, 则强制执行累加器数据范围下限, 保存超限结果	1	1	N,OV,Z
		MINZ $Acc$	强制执行累加器数据范围下限	1	1	N,OV,Z
		MINZ.V $Acc, Wd$	强制执行累加器数据范围下限, 保存超限结果	1	1	N,OV,Z

注 1: 对非CPU SFR执行读操作和读-修改-写操作 (例如, 位操作和逻辑运算) 时需要一个额外的指令周期。

2: 除法指令之前必须有一条 "REPEAT #5" 指令, 以便能够连续执行6次。

# dsPIC33CK64MP105系列

表29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
61	MOV	MOV <i>f</i> , <i>Wn</i>	将 <i>f</i> 中的内容送入 <i>Wn</i>	1	1	无
		MOV <i>f</i>	将 <i>f</i> 中的内容送入 <i>f</i>	1	1	无
		MOV <i>f</i> , WREG	将 <i>f</i> 中的内容送入WREG	1	1	无
		MOV #lit16, <i>Wn</i>	将16位立即数送入 <i>Wn</i>	1	1	无
		MOV.b #lit8, <i>Wn</i>	将8位立即数送入 <i>Wn</i>	1	1	无
		MOV <i>Wn</i> , <i>f</i>	将 <i>Wn</i> 中的内容送入 <i>f</i>	1	1	无
		MOV <i>Wso</i> , <i>Wdo</i>	将 <i>Ws</i> 中的内容送入 <i>Wd</i>	1	1	无
		MOV WREG, <i>f</i>	将WREG中的内容送入 <i>f</i>	1	1	无
		MOV.D <i>Wns</i> , <i>Wd</i>	将W(ns):W(ns + 1)中的双字内容送入 <i>Wd</i>	1	2	无
		MOV.D <i>Ws</i> , <i>Wnd</i>	将 <i>Ws</i> 中的双字内容送入W(nd + 1):W(nd)	1	2	无
62	MOVPAG	MOVPAG #lit10, DSRPAG	将10位立即数送入DSRPAG	1	1	无
		MOVPAG #lit8, TBLPAG	将8位立即数送入TBLPAG	1	1	无
		MOVPAG <i>Ws</i> , DSRPAG	将 <i>Ws</i> [9:0]中的内容送入DSRPAG	1	1	无
		MOVPAG <i>Ws</i> , TBLPAG	将 <i>Ws</i> [7:0]中的内容送入TBLPAG	1	1	无
64	MOVSAC	MOVSAC <i>Acc</i> , <i>Wx</i> , <i>Wxd</i> , <i>Wy</i> , <i>Wyd</i> , AWB	预取操作数并保存累加器	1	1	无
65	MPY	MPY <i>Wm</i> * <i>Wn</i> , <i>Acc</i> , <i>Wx</i> , <i>Wxd</i> , <i>Wy</i> , <i>Wyd</i>	<i>Wm</i> 与 <i>Wn</i> 相乘, 结果存入累加器	1	1	OA,OB,OAB, SA,SB,SAB
		MPY <i>Wm</i> * <i>Wm</i> , <i>Acc</i> , <i>Wx</i> , <i>Wxd</i> , <i>Wy</i> , <i>Wyd</i>	<i>Wm</i> 平方, 结果存入累加器	1	1	OA,OB,OAB, SA,SB,SAB
66	MPY.N	MPY.N <i>Wm</i> * <i>Wn</i> , <i>Acc</i> , <i>Wx</i> , <i>Wxd</i> , <i>Wy</i> , <i>Wyd</i>	<i>Wm</i> 与 <i>Wn</i> 相乘并取反, 结果存入累加器	1	1	无
67	MSC	MSC <i>Wm</i> * <i>Wm</i> , <i>Acc</i> , <i>Wx</i> , <i>Wxd</i> , <i>Wy</i> , <i>Wyd</i> , AWB	相乘再从累加器中减去	1	1	OA,OB,OAB, SA,SB,SAB
68	MUL	MUL.SS <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 有符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MUL.SS <i>Wb</i> , <i>Ws</i> , <i>Acc</i>	累加器 = 有符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MUL.SU <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 有符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MUL.SU <i>Wb</i> , <i>Ws</i> , <i>Acc</i>	累加器 = 有符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MUL.SU <i>Wb</i> , #lit5, <i>Acc</i>	累加器 = 有符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL.US <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 无符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MUL.US <i>Wb</i> , <i>Ws</i> , <i>Acc</i>	累加器 = 无符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MUL.UU <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 无符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MUL.UU <i>Wb</i> , #lit5, <i>Acc</i>	累加器 = 无符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL.UU <i>Wb</i> , <i>Ws</i> , <i>Acc</i>	累加器 = 无符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MULW.SS <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 有符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MULW.SU <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 有符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MULW.US <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 无符号( <i>Wb</i> ) * 有符号( <i>Ws</i> )	1	1	无
		MULW.UU <i>Wb</i> , <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 无符号( <i>Wb</i> ) * 无符号( <i>Ws</i> )	1	1	无
		MUL.SU <i>Wb</i> , #lit5, <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 有符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL.SU <i>Wb</i> , #lit5, <i>Wnd</i>	<i>Wnd</i> = 有符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL.UU <i>Wb</i> , #lit5, <i>Wnd</i>	{ <i>Wnd</i> + 1, <i>Wnd</i> } = 无符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL.UU <i>Wb</i> , #lit5, <i>Wnd</i>	<i>Wnd</i> = 无符号( <i>Wb</i> ) * 无符号(lit5)	1	1	无
		MUL <i>f</i>	W3:W2 = <i>f</i> * WREG	1	1	无

注 1: 对非CPU SFR执行读操作和读-修改-写操作(例如, 位操作和逻辑运算)时需要一个额外的指令周期。

2: 除法指令之前必须有一条“REPEAT #5”指令, 以便能够连续执行6次。

表 29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
69	NEG	NEG <i>Acc</i>	将累加器内容求补	1	1	OA,OB,OAB, SA,SB,SAB
		NEG <i>f</i>	$f = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG <i>f</i> , WREG	$WREG = \bar{f} + 1$	1	1	C,DC,N,OV,Z
		NEG <i>Ws</i> , <i>Wd</i>	$Wd = \overline{Ws} + 1$	1	1	C,DC,N,OV,Z
70	NOP	NOP	空操作	1	1	无
		NOPR	空操作	1	1	无
71	NORM	NORM <i>Acc</i> , <i>Wd</i>	归一化累加器	1	1	N,OV,Z
72	POP	POP <i>f</i>	将栈顶 (Top-of-Stack, TOS) 的内容弹出到 <i>f</i>	1	1	无
		POP <i>Wdo</i>	将栈顶 (TOS) 的内容弹出到 <i>Wdo</i>	1	1	无
		POP.D <i>Wnd</i>	将栈顶 (TOS) 的内容弹出到 <i>W(nd):W(nd + 1)</i>	1	2	无
		POP.S	将影子寄存器的内容弹出到主寄存器	1	1	全部
73	PUSH	PUSH <i>f</i>	将 <i>f</i> 的内容压入栈顶 (TOS)	1	1	无
		PUSH <i>Wso</i>	将 <i>Wso</i> 的内容压入栈顶 (TOS)	1	1	无
		PUSH.D <i>Wns</i>	将 <i>W(ns):W(ns + 1)</i> 的内容压入栈顶 (TOS)	1	2	无
		PUSH.S	将主寄存器中的内容压入影子寄存器	1	1	无
74	PWRSV	PWRSV #lit1	进入休眠或空闲模式	1	1	WDTO,Sleep
75	RCALL	RCALL <i>Expr</i>	相对调用	1	4	SFA
		RCALL <i>Wn</i>	计算调用	1	4	SFA
76	REPEAT	REPEAT #lit15	将下一条指令重复执行 lit15 + 1 次	1	1	无
		REPEAT <i>Wn</i>	将下一条指令重复执行 ( <i>Wn</i> ) + 1 次	1	1	无
77	RESET	RESET	软件器件复位	1	1	无
78	RETFIE	RETFIE	从中断返回	1	6 (5)	SFA
79	RETLW	RETLW #lit10, <i>Wn</i>	返回并将立即数存入 <i>Wn</i>	1	6 (5)	SFA
80	RETURN	RETURN	从子程序返回	1	6 (5)	SFA
81	RLC	RLC <i>f</i>	$f = \text{对 } f \text{ 执行带进位的循环左移}$	1	1	C,N,Z
		RLC <i>f</i> , WREG	$WREG = \text{对 } f \text{ 执行带进位的循环左移}$	1	1	C,N,Z
		RLC <i>Ws</i> , <i>Wd</i>	$Wd = \text{对 } Ws \text{ 执行带进位的循环左移}$	1	1	C,N,Z
82	RLNC	RLNC <i>f</i>	$f = \text{循环左移 } f \text{ (不带进位)}$	1	1	N,Z
		RLNC <i>f</i> , WREG	$WREG = \text{循环左移 } f \text{ (不带进位)}$	1	1	N,Z
		RLNC <i>Ws</i> , <i>Wd</i>	$Wd = \text{循环左移 } Ws \text{ (不带进位)}$	1	1	N,Z
83	RRC	RRC <i>f</i>	$f = \text{对 } f \text{ 执行带进位的循环右移}$	1	1	C,N,Z
		RRC <i>f</i> , WREG	$WREG = \text{对 } f \text{ 执行带进位的循环右移}$	1	1	C,N,Z
		RRC <i>Ws</i> , <i>Wd</i>	$Wd = \text{对 } Ws \text{ 执行带进位的循环右移}$	1	1	C,N,Z
84	RRNC	RRNC <i>f</i>	$f = \text{循环右移 } f \text{ (不带进位)}$	1	1	N,Z
		RRNC <i>f</i> , WREG	$WREG = \text{循环右移 } f \text{ (不带进位)}$	1	1	N,Z
		RRNC <i>Ws</i> , <i>Wd</i>	$Wd = \text{循环右移 } Ws \text{ (不带进位)}$	1	1	N,Z
85	SAC	SAC <i>Acc</i> , #Slit4, <i>Wdo</i>	保存累加器内容	1	1	无
		SAC.R <i>Acc</i> , #Slit4, <i>Wdo</i>	保存舍入后的累加器内容	1	1	无
86	SE	SE <i>Ws</i> , <i>Wnd</i>	<i>Wnd</i> = 符号扩展后的 <i>Ws</i>	1	1	C,N,Z
87	SETM	SETM <i>f</i>	$f = 0xFFFF$	1	1	无
		SETM WREG	$WREG = 0xFFFF$	1	1	无
		SETM <i>Ws</i>	$Ws = 0xFFFF$	1	1	无
88	SFTAC	SFTAC <i>Acc</i> , <i>Wn</i>	对累加器算术移位 ( <i>Wn</i> ) 次	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC <i>Acc</i> , #Slit6	对累加器算术移位 Slit6 次	1	1	OA,OB,OAB, SA,SB,SAB

注 1: 对非CPU SFR执行读操作和读-修改-写操作 (例如, 位操作和逻辑运算) 时需要一个额外的指令周期。

2: 除法指令之前必须有一条 “REPEAT #5” 指令, 以便能够连续执行 6 次。

# dsPIC33CK64MP105系列

表 29-2: 指令集概述 (续)

基本指令 编号	汇编 助记符	汇编语法	说明	字数	周期数 <sup>(1)</sup>	影响的 状态标志
89	SL	SL f	f = 左移f	1	1	C,N,OV,Z
		SL f, WREG	WREG = 左移f	1	1	C,N,OV,Z
		SL Ws, Wd	Wd = 左移Ws	1	1	C,N,OV,Z
		SL Wb, Wns, Wnd	Wnd = 将Wb左移Wns位	1	1	N,Z
		SL Wb, #lit5, Wnd	Wnd = 将Wb左移lit5位	1	1	N,Z
91	SUB	SUB Acc	从累加器减去	1	1	OA,OB,OAB, SA,SB,SAB
		SUB f	f = f - WREG	1	1	C,DC,N,OV,Z
		SUB f, WREG	WREG = f - WREG	1	1	C,DC,N,OV,Z
		SUB #lit10, Wn	Wn = Wn - lit10	1	1	C,DC,N,OV,Z
		SUB Wb, Ws, Wd	Wd = Wb - Ws	1	1	C,DC,N,OV,Z
		SUB Wb, #lit5, Wd	Wd = Wb - lit5	1	1	C,DC,N,OV,Z
92	SUBB	SUBB f	f = f - WREG - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBB f, WREG	WREG = f - WREG - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBB #lit10, Wn	Wn = Wn - lit10 - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBB Wb, Ws, Wd	Wd = Wb - Ws - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBB Wb, #lit5, Wd	Wd = Wb - lit5 - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
93	SUBR	SUBR f	f = WREG - f	1	1	C,DC,N,OV,Z
		SUBR f, WREG	WREG = WREG - f	1	1	C,DC,N,OV,Z
		SUBR Wb, Ws, Wd	Wd = Ws - Wb	1	1	C,DC,N,OV,Z
		SUBR Wb, #lit5, Wd	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
94	SUBBR	SUBBR f	f = WREG - f - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBBR f, WREG	WREG = WREG - f - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBBR Wb, Ws, Wd	Wd = Ws - Wb - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
		SUBBR Wb, #lit5, Wd	Wd = lit5 - Wb - ( $\overline{C}$ )	1	1	C,DC,N,OV,Z
95	SWAP	SWAP.b Wn	Wn = 半字节交换Wn内容	1	1	无
		SWAP Wn	Wn = 将Wn的两个字节相交换	1	1	无
96	TBLRDH	TBLRDH Ws, Wd	将程序存储器中某个单元的bit[23:16]读入Wd[7:0]	1	5	无
97	TBLRDL	TBLRDL Ws, Wd	将程序存储器中某个单元的bit[15:0]读入Wd	1	5	无
98	TBLWTH	TBLWTH Ws, Wd	将Ws[7:0]写入程序存储器中某个单元的bit[23:16]	1	2	无
99	TBLWTL	TBLWTL Ws, Wd	将Ws写入程序存储器中某个单元的bit[15:0]	1	2	无
101	ULNK	ULNK	释放堆栈帧	1	1	SFA
104	XOR	XOR f	f = f .XOR.WREG	1	1	N,Z
		XOR f, WREG	WREG = f .XOR.WREG	1	1	N,Z
		XOR #lit10, Wn	Wd = lit10 .XOR.Wd	1	1	N,Z
		XOR Wb, Ws, Wd	Wd = Wb .XOR.Ws	1	1	N,Z
		XOR Wb, #lit5, Wd	Wd = Wb .XOR. lit5	1	1	N,Z
105	ZE	ZE Ws, Wnd	Wnd = 零扩展后的Ws	1	1	C,Z,N

注 1: 对非CPU SFR执行读操作和读-修改-写操作(例如, 位操作和逻辑运算)时需要一个额外的指令周期。

2: 除法指令之前必须有一条“REPEAT #5”指令, 以便能够连续执行6次。

## 30.0 开发支持

利用Microchip屡获奖项的开发工具，快速将设计从概念转化为生产。通过我们免费的MPLAB® X和Atmel Studio集成开发环境（Integrated Development Environment, IDE）以及代码生成工具的简单易用的图形用户界面（Graphical User Interfaces, GUI），Microchip的各种开发工具协同工作，为您的任何项目提供高级调试功能。Microchip的一系列编程器、调试器和仿真器与我们的软件工具无缝配合工作，为您提供极致的易用体验。Microchip的开发板有助于评估适用于应用的最佳芯片。Microchip的所有这些工具与我们的第三方工具系列一起构成了完善的开发工具解决方案。

Microchip的MPLAB X和Atmel Studio生态系统为您提供各种嵌入式设计工具，这些工具支持多款器件，包括PIC® MCU、AVR® MCU、SAM MCU和dsPIC® DSC。MPLAB X工具兼容Windows®、Linux®和Mac®操作系统，Atmel Studio工具与Windows兼容。

更多信息与详情请访问如下网站：

<https://www.microchip.com/development-tools/>

注:

31.0 电气特性

本章将对dsPIC33CK64MP105系列的电气特性进行概括介绍。其余信息在本文档的将来版本中提供。

下面列出了dsPIC33CK64MP105系列的绝对最大值。器件长时间工作在最大值条件下，其可靠性可能受到影响。未表明器件处于或超过本规范指定的极限值条件下仍可正常工作。

绝对最大值<sup>(1)</sup>

偏置时的环境温度 .....	-40°C 至 +125°C
储存温度 .....	-65°C 至 +150°C
VDD 引脚相对于 Vss 的电压 .....	-0.3V 至 +4.0V
任一非5V耐压引脚相对于 Vss 的电压 <sup>(3)</sup> .....	-0.3V 至 (VDD + 0.3V)
任一5V耐压引脚相对于 Vss 的电压 <sup>(3)</sup> .....	-0.3V 至 +5.5V
流出 Vss 引脚的最大电流 .....	300 mA
流入 VDD 引脚的最大电流 <sup>(2)</sup> .....	300 mA
任一常规 I/O 引脚的最大灌电流/拉电流 .....	15 mA
电流驱动能力增强的 I/O 引脚 (RB1、RC8、RC9 和 RD8) 的最大灌电流/拉电流 .....	25 mA
两个 Vss 引脚之间的一组 I/O 的最大灌电流 <sup>(4)</sup> .....	75 mA
两个 VDD 引脚之间的一组 I/O 的最大拉电流 <sup>(4)</sup> .....	75 mA
所有 I/O 的最大灌电流 <sup>(2,5)</sup> .....	200 mA
所有 I/O 的最大拉电流 <sup>(2,5)</sup> .....	200 mA

- 注 1:** 如果器件的工作条件超过上述“绝对最大值”，可能对器件造成永久性损坏。上述值仅代表本规范规定的极限工作条件，未表明器件在上述极限值或超出极限值的情况下仍可正常工作。器件长时间工作在最大值条件下，其可靠性可能受到影响。

**2:** 允许的最大电流由器件最大功耗决定（见表31-2）。

**3:** 关于5V耐压的引脚，请参见“引脚分配图”一节。

**4:** 不适用于 AVDD 和 AVss 引脚。

**5:** 对于28引脚封装，所有 I/O 的最大灌电流/拉电流限制为 150 mA。

# dsPIC33CK64MP105系列

## 31.1 直流特性

表31-1: dsPIC33CK64MP105系列工作条件

VDD 范围	温度范围	最大CPU时钟频率
3.0V 至 3.6V	-40°C 至 +125°C	100 MHz

表31-2: 热工作条件

参数	符号	最小值	最大值	单位
工业级温度器件				
工作结温范围	TJ	-40	+125	°C
工作环境温度范围	TA	-40	+85	°C
扩展级温度器件				
工作结温范围	TJ	-40	+140	°C
工作环境温度范围	TA	-40	+125	°C
功耗: 芯片内部功耗: $P_{INT} = V_{DD} \times (I_{DD} - \Sigma I_{OH})$ I/O 引脚功耗: $I/O = \Sigma (\{V_{DD} - V_{OH}\} \times I_{OH}) + \Sigma (V_{OL} \times I_{OL})$	PD	PINT + PI/O		W
最大允许功耗	PDMAX	(TJ - TA)/θJA		W

表31-3: 封装热阻<sup>(1)</sup>

封装	符号	典型值	单位
48 引脚 TQFP 7x7 mm	θJA	62.76	°C/W
48 引脚 UQFN 6x6 mm	θJA	27.6	°C/W
36 引脚 UQFN 5x5 mm	θJA	29.2	°C/W
28 引脚 UQFN 6x6 mm	θJA	22.41	°C/W
28 引脚 SSOP 5.30 mm	θJA	52.84	°C/W

注 1: 通过封装模拟获得结与环境的热阻 (θJA) 值。

表31-4: 工作电压规范

工作条件（除非另外声明）： -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	最大值	单位	条件
DC10	VDD	电源电压	3.0	3.6	V	
DC16	VPOR	确保内部上电复位信号的VDD启动电压	—	VSS	V	
DC17	SVDD	确保内部上电复位信号的VDD上升速率	0.03	—	V/ms	0V-3V/100 ms
BO10	VBOR <sup>(1)</sup>	VDD跳变（高电压到低电压）时的BOR事件	2.68	2.99	V	

注 1: 器件可在 VBORMIN < VDD < VDDMIN 条件下工作，但此时模拟模块（ADC 和比较器）的性能可能下降。VBOR 参数仅供设计参考，未经生产测试。



表31-5: 工作电流 (IDD) (2)

参数编号	典型值 <sup>(1)</sup>	最大值	单位	条件		
DC20	5.5	6.7	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, FVCO = 400 MHz, FPLLO = 40 MHz)
	5.6	6.9	mA	+25°C		
	6.3	9.5	mA	+85°C		
	8.5	18.0	mA	+125°C		
DC21	7.5	11.0	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 280 MHz)
	7.6	9.1	mA	+25°C		
	8.3	11.7	mA	+85°C		
	10.5	20.2	mA	+125°C		
DC22	10.7	15.8	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, FVCO = 480 MHz, FPLLO = 160 MHz)
	10.8	12.7	mA	+25°C		
	11.6	15.3	mA	+85°C		
	13.9	23.8	mA	+125°C		
DC23	16.6	25.8	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	16.9	19.4	mA	+25°C		
	17.7	22.0	mA	+85°C		
	20.0	30.4	mA	+125°C		
DC24	21.1	32.7	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, FVCO = 720 MHz, FPLLO = 360 MHz)
	21.4	24.5	mA	+25°C		
	22.1	27.0	mA	+85°C		
	23.9	34.5	mA	+125°C		
DC25	20.7	33.9	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, FVCO = 400 MHz, FPLLO = 400 MHz)
	21.0	24.1	mA	+25°C		
	21.4	26.2	mA	+85°C		
	23.7	35.0	mA	+125°C		

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本运行电流 (IDD) 的测量条件如下:

- 通过软件将振荡器切换为EC+PLL 模式
- OSC1 引脚通过电压为0.3V至VDD - 0.3V的外部8 MHz方波驱动
- OSC2通过配置字配置为I/O (OSCIOFCN (FOSC[2]) = 0)
- 禁止FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWDTEN (FWDIT[15]) = 0)
- 所有I/O引脚 (OSC1除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的PMDx位均置1)
- 禁止JTAG (JTAGEN (FICD[5]) = 0)
- 执行NOP指令

# dsPIC33CK64MP105系列

表31-6: 空闲电流 (I<sub>IDLE</sub>) (2)

参数编号	典型值 <sup>(1)</sup>	最大值	单位	条件		
DC30	4.5	6.5	mA	-40°C	3.3V	10 MIPS (N = 1, N2 = 5, N3 = 2, M = 50, Fvco = 400 MHz, FPLLO = 40 MHz)
	4.5	5.8	mA	+25°C		
	5.3	8.7	mA	+85°C		
	7.5	17.6	mA	+125°C		
DC31	5.1	7.8	mA	-40°C	3.3V	20 MIPS (N = 1, N2 = 5, N3 = 1, M = 50, Fvco = 400 MHz, FPLLO = 80 MHz)
	5.2	6.5	mA	+25°C		
	5.9	9.7	mA	+85°C		
	8.1	18.3	mA	+125°C		
DC32	6.7	9.2	mA	-40°C	3.3V	40 MIPS (N = 1, N2 = 3, N3 = 1, M = 60, Fvco = 480 MHz, FPLLO = 160 MHz)
	6.8	8.1	mA	+25°C		
	7.4	12.5	mA	+85°C		
	9.7	19.8	mA	+125°C		
DC33	8.9	12.5	mA	-40°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, Fvco = 560 MHz, FPLLO = 280 MHz)
	9.0	10.5	mA	+25°C		
	9.6	16.0	mA	+85°C		
	11.8	23.3	mA	+125°C		
DC34	10.6	16.6	mA	-40°C	3.3V	90 MIPS (N = 1, N2 = 2, N3 = 1, M = 90, Fvco = 720 MHz, FPLLO = 360 MHz)
	10.8	12.5	mA	+25°C		
	11.4	18.4	mA	+85°C		
	13.7	26.1	mA	+125°C		
DC35	10.2	15.3	mA	-40°C	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, Fvco = 400 MHz, FPLLO = 400 MHz)
	10.3	12.0	mA	+25°C		
	10.9	17.5	mA	+85°C		
	13.2	25.2	mA	+125°C		

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本空闲电流 (I<sub>IDLE</sub>) 的测量条件如下:

- 通过软件将振荡器切换为EC+PLL 模式
- OSC1 引脚通过电压为0.3V至V<sub>DD</sub> - 0.3V的外部8 MHz方波驱动
- OSC2通过配置字配置为I/O (OSCIOFCN (FOSC[2]) = 0)
- 禁止FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWDTEN (FWD[15]) = 0)
- 所有I/O引脚 (OSC1除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的PMDx位均置1)
- 禁止JTAG (JTAGEN (FICD[5]) = 0)
- 执行NOP指令

表31-7: 掉电电流 (IPD) <sup>(2)</sup>

参数编号	典型值 <sup>(1)</sup>	最大值	单位	条件		
DC40 <sup>(3)</sup>	0.3	0.7	mA	-40°C	3.3V	VREGS 位 (RCON[8]) = 0
	0.5	1.3	mA	+25°C		
	1.5	4.7	mA	+85°C		
DC41	0.9	—	mA	-40°C	3.3V	VREGS 位 (RCON[8]) = 1
	1.1	—	mA	+25°C		
	2.3	—	mA	+85°C		
	4.7	13.9	mA	+125°C		

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本休眠电流 (IPD) 的测量条件如下:

- OSC1 引脚通过电压为 0.3V 至  $V_{DD} - 0.3V$  的外部 8 MHz 方波驱动
- OSC2 通过配置字配置为 I/O (OSCIOFCN (FOSC[2]) = 0)
- 使能稳压器的低功耗模式 (LPWREN (VREGCON[15]) = 1)
- 禁止 FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWDTEN (FWDTC[15]) = 0)
- 所有 I/O 引脚 (OSC1 除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的 PMDx 位均置 1)
- 禁止 JTAG (JTAGEN (FICD[5]) = 0)

3: 稳压器待机模式 (当 VREGS 位 = 0 时) 仅在工业温度范围内有效:  $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ 。

表31-8: 打盹电流 (ID0ZE)

参数编号	典型值 <sup>(1)</sup>	打盹模式时钟分频比	单位	条件		
DC70	13.4	1:2	mA	-40℃	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, Fvco = 560 MHz, FPLLO = 280 MHz)
	9.1	1:128	mA			
	13.6	1:2	mA	+25℃		
	9.2	1:128	mA			
	14.1	1:2	mA	+85℃		
	9.9	1:128	mA			
	16.4	1:2	mA	+125℃		
	12.1	1:128	mA			
DC71	16.6	1:2	mA	-40℃	3.3V	100 MIPS (N = 1, N2 = 1, N3 = 1, M = 50, Fvco = 400 MHz, FPLLO = 400 MHz)
	10.5	1:128	mA			
	16.9	1:2	mA	+25℃		
	10.6	1:128	mA			
	17.2	1:2	mA	+85℃		
	11.3	1:128	mA			
	19.5	1:2	mA	+125℃		
	13.5	1:128	mA			

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

# dsPIC33CK64MP105系列

表31-9: 看门狗定时器增量电流 ( $\Delta I_{WDT}$ ) <sup>(1)</sup>

参数编号	典型值	单位	条件	
DC61	1	$\mu A$	-40°C	3.3V
	2	$\mu A$	+25°C	
	4	$\mu A$	+85°C	
	11	$\mu A$	+125°C	

注 1:  $\Delta I_{WDT}$  电流为当模块使能时额外消耗的电流。此电流应被加到基本  $I_{PD}$  电流。所有参数仅供设计参考，未经测试。

表31-10: PWM增量电流 <sup>(1)</sup>

参数编号	典型值	最大值	单位	条件		
DC100	5.96	6.6	mA	-40°C	3.3V	PWM输出频率 = 500 kHz, PWM输入 (AFPLLO = 500 MHz) (AVCO = 1000 MHz, PLLFBD = 125, APLLDIV1 = 2)
	5.99	6.7	mA	+25°C		
	5.92	6.9	mA	+85°C		
	5.47	7	mA	+125°C		
DC101	4.89	5.4	mA	-40°C	3.3V	PWM输出频率 = 500 kHz, PWM输入 (AFPLLO = 400 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 1)
	4.91	5.5	mA	+25°C		
	4.85	5.7	mA	+85°C		
	4.42	5.7	mA	+125°C		
DC102	2.77	3.7	mA	-40°C	3.3V	PWM输出频率 = 500 kHz, PWM输入 (AFPLLO = 200 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 2)
	2.75	3.7	mA	+25°C		
	2.7	3.7	mA	+85°C		
	2.26	3.7	mA	+125°C		
DC103	1.67	2	mA	-40°C	3.3V	PWM输出频率 = 500 kHz, PWM输入 (AFPLLO = 100 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 4)
	1.66	2.2	mA	+25°C		
	1.63	2.3	mA	+85°C		
	1.17	2.3	mA	+125°C		

注 1: 不包括 APLL 电流。如果多个 PWM 在运行，则 APLL 电流将相同。当 HREN (PGxCONL[7]) = 0 时，列出的增量电流仅针对一个 PWM 实例。所有参数均为特性值，但未经生产测试。

表31-11: APLL增量电流

参数编号	典型值	最大值	单位	条件 <sup>(1)</sup>		
DC110	5.93	6.6	mA	-40°C	3.3V	AF <sub>P</sub> LLO = 500 MHz (AVCO = 1000 MHz, PLLFBD = 125, APLLDIV1 = 2)
	5.95	7	mA	+25°C		
	6.15	7.6	mA	+85°C		
	7.15	9	mA	+125°C		
DC111	2.72	3.3	mA	-40°C	3.3V	AF <sub>P</sub> LLO = 400 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 1)
	2.74	3.7	mA	+25°C		
	2.92	4.3	mA	+85°C		
	3.87	5.6	mA	+125°C		
DC112	1.39	2.7	mA	-40°C	3.3V	AF <sub>P</sub> LLO = 200 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 2)
	1.49	2.7	mA	+25°C		
	1.65	3	mA	+85°C		
	2.6	4.4	mA	+125°C		
DC113	0.79	1.1	mA	-40°C	3.3V	AF <sub>P</sub> LLO = 100 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 4)
	0.84	1.4	mA	+25°C		
	0.96	2.3	mA	+85°C		
	1.93	3.6	mA	+125°C		

注 1: 如果多个PWM或DAC基于APLL时钟运行, 则APLL电流将相同。所有参数均为特性值, 但未经生产测试。

表31-12: ADC增量电流<sup>(1)</sup>

参数编号	典型值	最大值	单位	条件		
DC120	3.61	4	mA	-40°C	3.3V	T <sub>AD</sub> = 14.3 ns (3.5 Msps转换速率)
	3.68	4.1	mA	+25°C		
	3.69	4.2	mA	+85°C		
	3.89	4.6	mA	+125°C		

注 1: 共用内核连续转换。T<sub>AD</sub> = 14.3 ns (3.5 Msps转换速率)。列出的增量电流仅针对一个ADC内核。所有参数均为特性值, 但未经生产测试。

# dsPIC33CK64MP105系列

表31-13： 比较器 + DAC增量电流

参数编号	典型值	最大值	单位	条件		
DC130	1.2	1.35	mA	-40°C	3.3V	AFPLLO @ 500 MHz <sup>(1)</sup>
	1.23	1.65	mA	+25°C		
	1.23	1.65	mA	+85°C		
	1.24	1.65	mA	+125°C		

注 1： 不包括APLL 电流。列出的增量电流仅针对一个比较器 + DAC实例。所有参数均为特性值，但未经生产测试。

表31-14： 运放增量电流<sup>(1)</sup>

参数编号	典型值	最大值	单位	条件	
DC140	0.25	1	mA	-40°C	3.3V
	0.27	1.1	mA	+25°C	
	0.32	1.4	mA	+85°C	
	0.46	1.7	mA	+125°C	

注 1： 列出的增量电流仅针对一个运放实例。所有参数均为特性值，但未经生产测试。

表31-15: I/O引脚输入规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）						
参数编号	符号	特性	最小值	最大值	单位	条件
DI10	$V_{IL}$	低电平输入电压				
		所有 I/O 引脚和 $\overline{MCLR}$	$V_{SS}$	$0.2 V_{DD}$	V	
		与 SDAx 和 SCLx 复用的 I/O 引脚	$V_{SS}$	$0.3 V_{DD}$	V	禁止 SMBus
		与 SDAx 和 SCLx 复用的 I/O 引脚	$V_{SS}$	0.8	V	使能 SMBus
DI20	$V_{IH}$	与 SDAx 和 SCLx 复用的 I/O 引脚	$V_{SS}$	0.8	V	使能 SMBus 3.0
		高电平输入电压 <sup>(1)</sup>				
		非 5V 耐压的 I/O 引脚	$0.8 V_{DD}$	$V_{DD}$	V	
		5V 耐压的 I/O 引脚和 $\overline{MCLR}$	$0.8 V_{DD}$	5.5	V	
		与 SDAx 和 SCLx 复用的 5V 耐压 I/O 引脚	$0.8 V_{DD}$	5.5	V	禁止 SMBus
		与 SDAx 和 SCLx 复用的 5V 耐压 I/O 引脚	2.1	5.5	V	使能 SMBus
		与 SDAx 和 SCLx 复用的 5V 耐压 I/O 引脚	1.35	$V_{DD}$	V	使能 SMBus 3.0
		与 SDAx 和 SCLx 复用的非 5V 耐压 I/O 引脚	$0.8 V_{DD}$	$V_{DD}$	V	禁止 SMBus
DI30	ICNPU	使能上拉电阻时的输入电流 <sup>(2)</sup>	175	545	$\mu A$	$V_{DD} = 3.3V$ , $V_{PIN} = V_{SS}$
		使能下拉电阻时的输入电流 <sup>(2)</sup>	65	360	$\mu A$	$V_{DD} = 3.3V$ , $V_{PIN} = V_{DD}$
DI50	IIL	输入泄漏电流	-1	—	$\mu A$	$V_{PIN} = V_{SS}$
		I/O 引脚和 $\overline{MCLR}$ 引脚	—	1	$\mu A$	$V_{PIN} = V_{DD}$

注 1: 关于 5V 耐压的 I/O 引脚，请参见“引脚分配图”部分。

2: 特性值，但未经测试。

# dsPIC33CK64MP105系列

**表31-16: I/O引脚输入注入电流规范**

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）						
参数编号	符号	特性	最小值	最大值	单位	条件
DI60a	I <sub>ICL</sub>	输入低注入电流	0	-5 <sup>(1,4)</sup>	mA	该参数适用于所有引脚
DI60b	I <sub>ICH</sub>	输入高注入电流	0	+5 <sup>(2,3,4)</sup>	mA	该参数适用于除5V耐压引脚和SOSCI外的所有引脚
DI60c	ΣI <sub>ICT</sub>	总输入注入电流（所有I/O和控制引脚的和）	-20 <sup>(5)</sup>	+20 <sup>(5)</sup>	mA	来自所有I/O引脚的所有±输入注入电流的绝对瞬时值的和 $\Sigma( I_{ICL}  +  I_{ICH} ) \leq \Sigma I_{ICT}$

注 1: V<sub>IL</sub>源 < (V<sub>SS</sub> - 0.3)。

2: V<sub>IH</sub>源 > (V<sub>DD</sub> + 0.3) 仅适用于非5V耐压的引脚。

3: 5V耐压引脚没有连接到V<sub>DD</sub>的内部上端二极管，因此不能承受任何“正”输入注入电流。

4: 注入电流会影响ADC结果。

5: I<sub>ICL</sub>或I<sub>ICH</sub>条件下未排除的任何数量的I/O引脚和/或其组合均允许包含在总和和中。

**表31-17: I/O引脚输出规范**

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）					
参数	符号	特性	典型值 <sup>(1)</sup>	单位	条件
DO10	VOL	灌电流驱动电压	0.2	V	I <sub>SINK</sub> = 3.0 mA, V <sub>DD</sub> = 3.3V
			0.4	V	I <sub>SINK</sub> = 6.0 mA, V <sub>DD</sub> = 3.3V
			0.6	V	I <sub>SINK</sub> = 9.0 mA, V <sub>DD</sub> = 3.3V
		灌电流驱动电压 对于RB1、RC8、RC9和RD8引脚	0.25	V	I <sub>SINK</sub> = 6.0 mA, V <sub>DD</sub> = 3.3V
			0.5	V	I <sub>SINK</sub> = 12.0 mA, V <sub>DD</sub> = 3.3V
			0.75	V	I <sub>SINK</sub> = 18.0 mA, V <sub>DD</sub> = 3.3V
DO20	VOH	拉电流驱动电压	3.1	V	I <sub>SOURCE</sub> = 3.0 mA, V <sub>DD</sub> = 3.3V
			2.9	V	I <sub>SOURCE</sub> = 6.0 mA, V <sub>DD</sub> = 3.3V
			2.7	V	I <sub>SOURCE</sub> = 9.0 mA, V <sub>DD</sub> = 3.3V
		拉电流驱动电压 对于RB1、RC8、RC9和RD8引脚	3.1	V	I <sub>SOURCE</sub> = 6.0 mA, V <sub>DD</sub> = 3.3V
			2.8	V	I <sub>SOURCE</sub> = 12.0 mA, V <sub>DD</sub> = 3.3V
			2.6	V	I <sub>SOURCE</sub> = 18.0 mA, V <sub>DD</sub> = 3.3V

注 1: 除非另外声明，否则“典型值”列中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。



表31-18: 闪存程序存储器规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}\text{C} \leq T_A \leq +85^{\circ}\text{C}$ （工业级） $-40^{\circ}\text{C} \leq T_A \leq +125^{\circ}\text{C}$ （扩展级）						
参数 编号	符号	特性	最小值	最大值	单位	条件
		<b>闪存程序存储器</b>				
D130	EP	单元耐擦写能力	10,000	—	E/W	
D134	TRETD	特性保持时间	20	—	年	
D137a	TPE	自定时页擦除时间	—	20	ms	
D137b	TCE	自定时芯片擦除时间	—	20	ms	
D138a	TWW	自定时双字写周期时间	—	20	$\mu\text{s}$	6字节，数据不全是1
D138b	TRW	自定时行写周期时间	—	1.28	ms	384字节，数据不全是1

## 31.2 交流特性和时序参数

图31-1: I/O负载条件规范

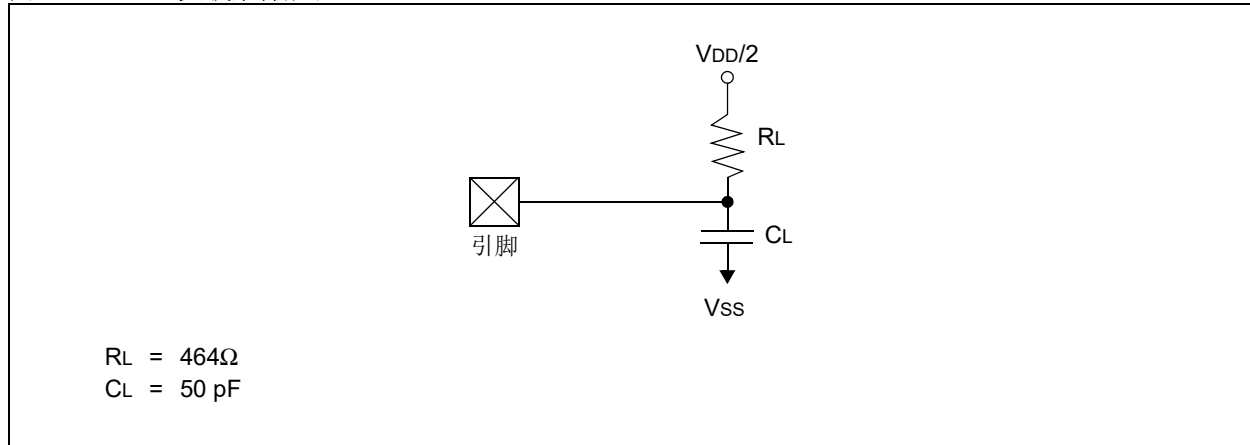


图31-2: I/O时序特性

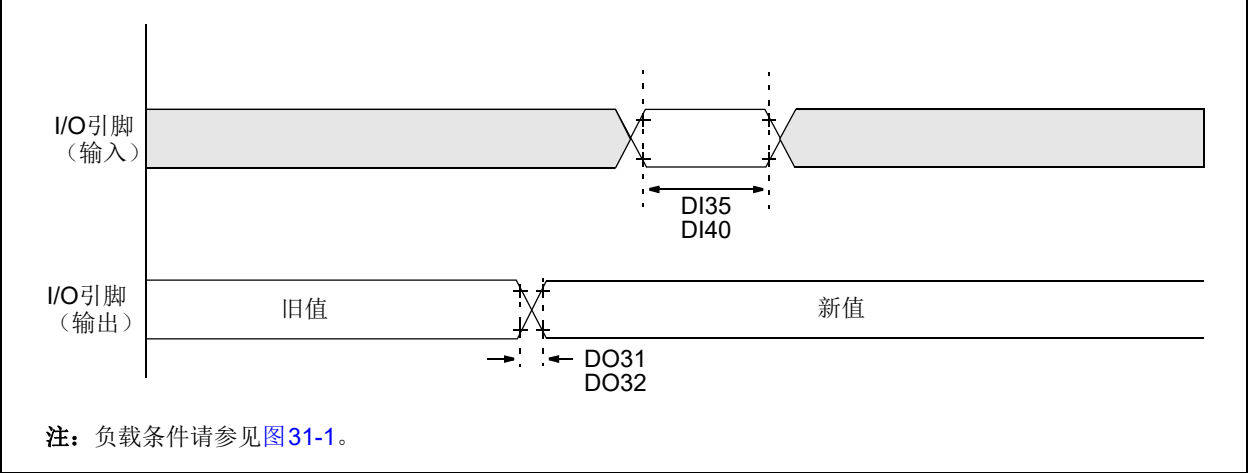


表31-19: I/O时序要求

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	最大值	单位
DO31	TioR	端口输出上升时间 <sup>(1)</sup>	—	10	ns
DO32	TioF	端口输出下降时间 <sup>(1)</sup>	—	10	ns
DI35	TINP	INTx输入引脚高电平或低电平时间	20	—	ns
DI40	TRBP	I/O和CNx输入高电平或低电平时间	2	—	Tcy

注 1： 该参数为特性值，但未经生产测试。

# dsPIC33CK64MP105系列

图31-3： 外部时钟时序

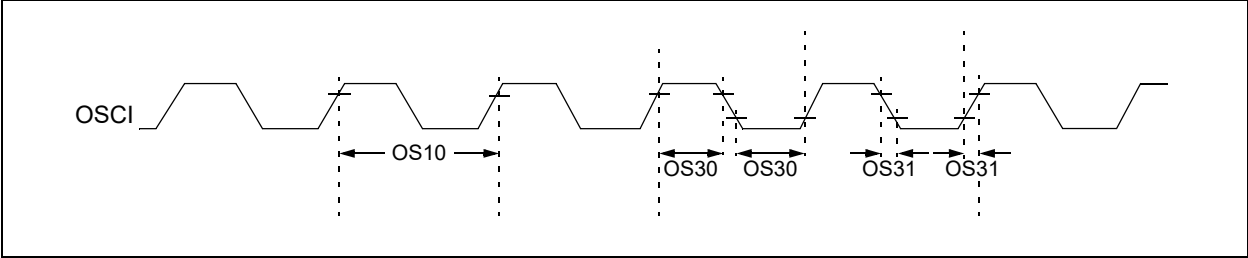


表31-20： 外部时钟时序要求

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	最大值	单位	条件
OS10	FIN	外部CLKI频率	DC	64	MHz	EC
		晶振频率	3.5	10	MHz	XT
			10	32	MHz	HS
OS30	TosL, TosH	外部时钟输入（OSCI）高电平或低电平时间	0.45 x OS10	0.55 x OS10	ns	EC
OS31	TosR, TosF	外部时钟输入（OSCI）上升或下降时间 <sup>(1)</sup>	—	10	ns	EC

注 1： 该参数为特性值，但未经生产测试。

表31-21: PLL时钟时序规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）					
参数编号	符号	特性	最小值	最大值	单位
OS50	FPLLI	PLL输入频率范围	8	64	MHz
OS51	FPPD	相频检测器输入频率（在第一个分频器后）	8	Fvco/16	MHz
OS52	FVCO	VCO输出频率	400	1600	MHz
OS53	TLOCK	PLL的锁定时间 <sup>(1)</sup>	—	250	μs

注 1: 该参数为特性值，但未经生产测试。

表31-22: 附属PLL时钟时序规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）					
参数编号	符号	特性	最小值	最大值	单位
OS60	FPLLI	APLL输入频率范围	8	64	MHz
OS61	FPPD	相频检测器输入频率（在第一个分频器后）	8	Fvco/16	MHz
OS62	FVCO	VCO输出频率	400	1600	MHz
OS63	TLOCK	APLL的锁定时间 <sup>(1)</sup>	—	250	μs

注 1: 该参数为特性值，但未经生产测试。

# dsPIC33CK64MP105系列

表31-23: FRC振荡器规范

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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 <sup>(2)</sup>	最大值	单位	条件
F20	AFRC	FRC精度 @ 8 MHz <sup>(1)</sup>	-3.0	—	3.0	%	-40°C ≤ TA ≤ 0°C
			-1.5	—	1.5	%	0°C ≤ TA ≤ 85°C
			-2.0	—	2.0	%	+85°C ≤ TA ≤ +125°C
F21	TFRC	FRC振荡器起振延时 <sup>(3)</sup>	—	—	15	μs	
F22	STUNE	OSCTUN步长	—	0.05	—	%/位	

注 1: 为了达到这种精度，必须使施加到单片机封装的物理应力（例如，通过弯曲PCB）尽可能小。

2: 除非另外声明，否则“典型值”列中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

3: 该参数为特性值，但未经生产测试。

表31-24: LPRC振荡器规范

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	最大值	单位
F30	ALPRC	频率为32 kHz时的LPRC精度	-30	30	%
F31	TLPRC	LPRC振荡器起振时间 <sup>(1)</sup>	—	50	μs

注 1: 该参数为特性值，但未经生产测试。

表31-25: BFRC振荡器规范

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	最大值	单位
F40	ABFRC	频率为8 MHz时的BFRC精度	-17	17	%

图31-4: BOR和主复位时序特性

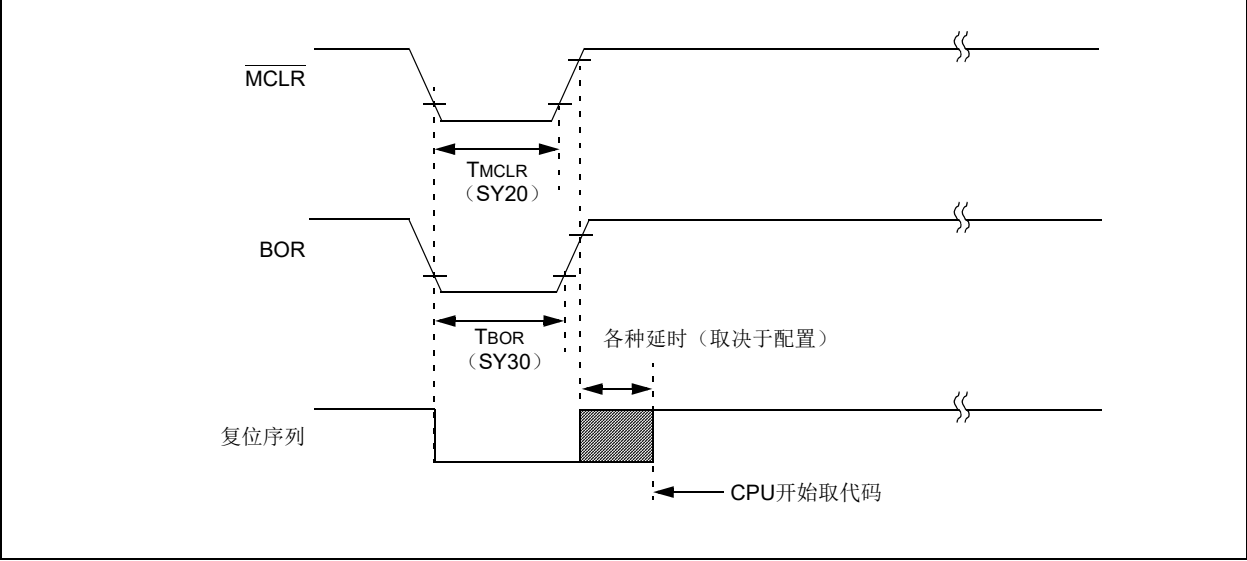


表31-26: 复位和欠压复位要求

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -40°C ≤ TA ≤ +85°C（工业级） -40°C ≤ TA ≤ +125°C（扩展级）						
参数编号	符号	特性 <sup>(1)</sup>	最小值	典型值 <sup>(2)</sup>	最大值	单位
SY13	Tioz	自MCLR低电平或看门狗定时器复位起I/O处于高阻态的时间	—	1.5	—	μs
SY20	TMCLR	MCLR脉冲宽度（低电平）	2	—	—	μs
SY30	TBOR	BOR脉冲宽度（低电平）	1	—	—	μs
SY35	TFSCM	故障保护时钟监视器延时	—	—	40	μs

注 1: 这些参数为特性值，但未经生产测试。  
2: 除非另外声明，否则“典型值”列中的数据均为3.3V和+25°C条件下的值。

# dsPIC33CK64MP105系列

图31-5： 高速PWMx模块时序特性

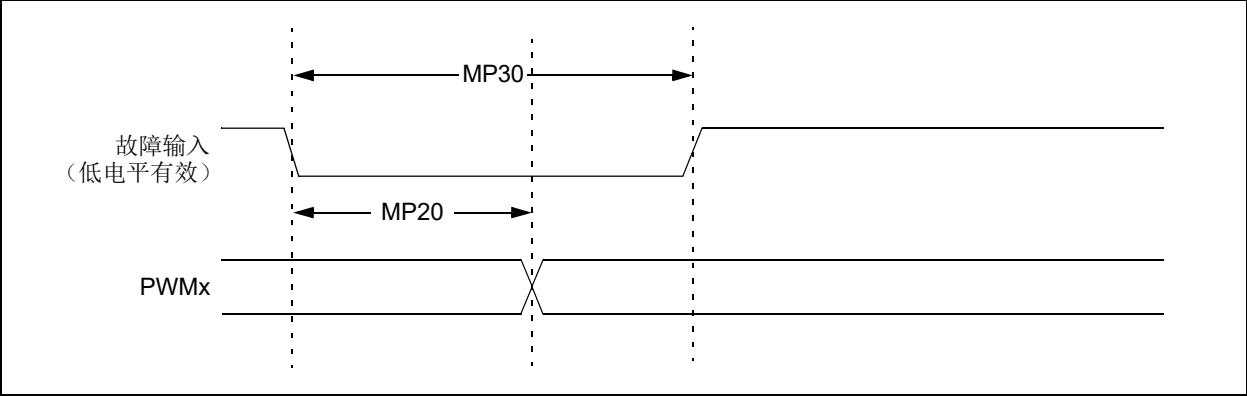


表31-27： 高速PWMx模块时序要求

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -40°C ≤ TA ≤ +85°C（工业级） -40°C ≤ TA ≤ +125°C（扩展级）					
参数编号	符号	特性 <sup>(1)</sup>	最小值	最大值	单位
MP10	FIN	PWM输入频率 <sup>(2)</sup>	—	500	MHz
MP20	TFD	故障输入↓到PWMx I/O发生变化的时间	—	26	ns
MP30	TFH	故障输入脉冲宽度	8	—	ns

注 1： 这些参数为特性值，但未经生产测试。  
2： 高分辨率模式必须使用 500 MHz 的输入频率。



图31-6: SPIx模块主模式 (CKE = 0) 时序特性

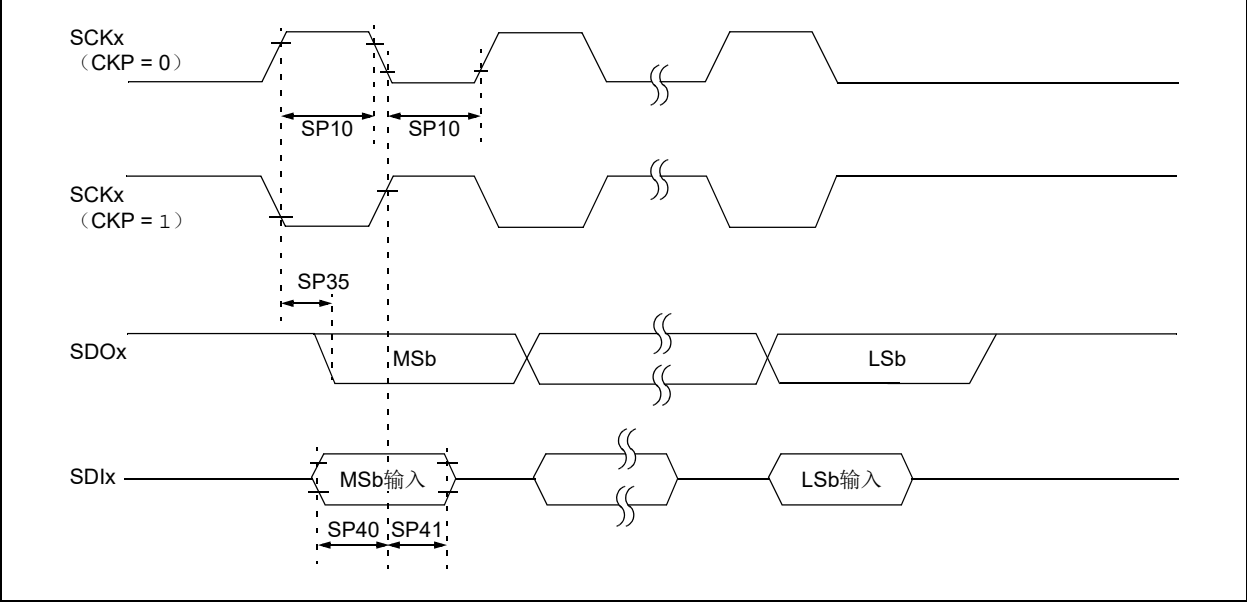
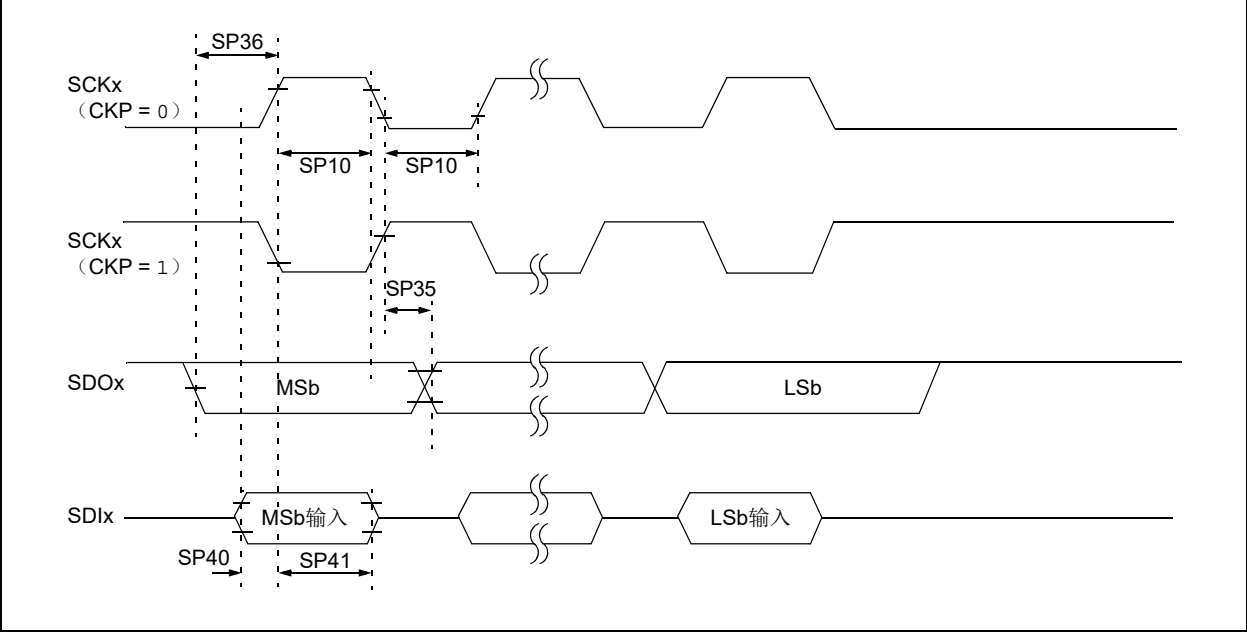


图31-7: SPIx模块主模式 (CKE = 1) 时序特性



# dsPIC33CK64MP105系列

表31-28： SPIx模块主模式时序要求

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -40°C ≤ TA ≤ +85°C（工业级） -40°C ≤ TA ≤ +125°C（扩展级）					
参数编号	符号	特性 <sup>(1)</sup>	最小值	最大值	单位
SP10	TscL, TscH	SCKx 输出低电平或高电平时间	15	—	ns
SP35	Tsch2doV, TscL2doV	SCKx 边沿之后 SDOx 数据输出有效的的时间	—	20	ns
SP36	TdoV2sc, TdoV2scL	SDOx 数据输出建立到第一个 SCKx 边沿的时间	3	—	ns
SP40	TdiV2sch, TdiV2scL	SDIx 数据输入到 SCKx 边沿的建立时间	10	—	ns
SP41	Tsch2diL, TscL2diL	SDIx 数据输入到 SCKx 边沿的保持时间	15	—	ns

注 1： 这些参数为特性值，但未经生产测试。

图31-8： SPIx模块从模式（CKE = 0）时序特性

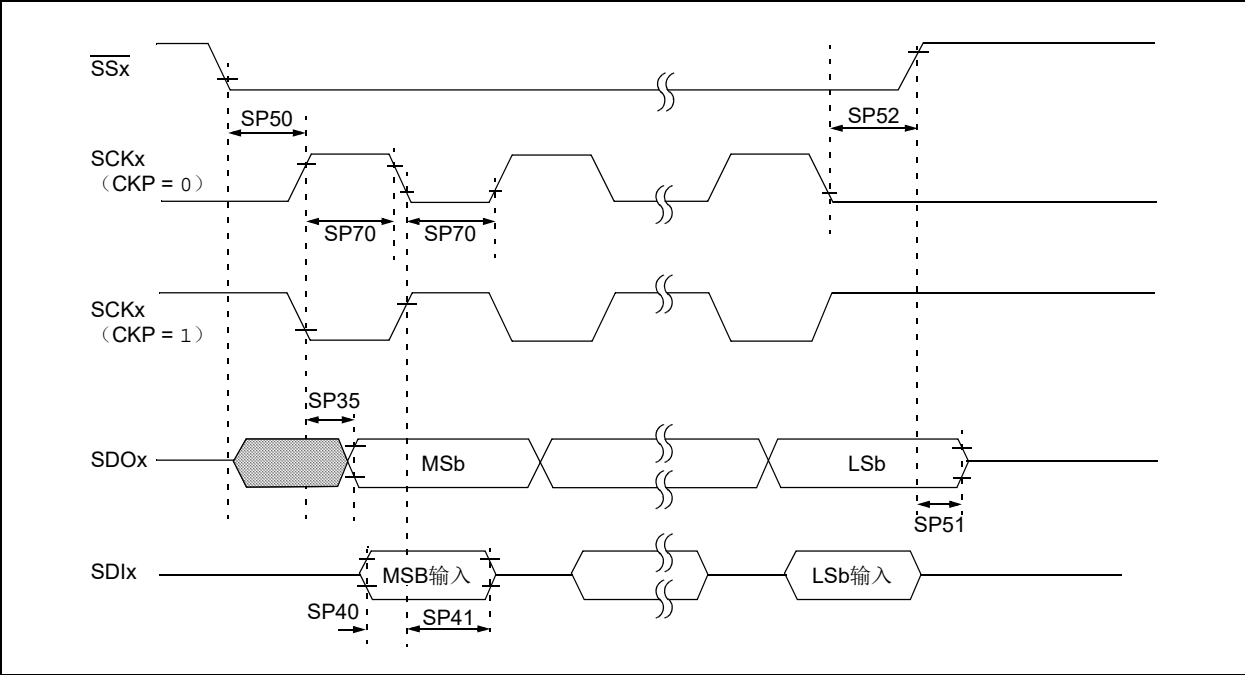


图31-9: SPIx模块从模式 (CKE = 1) 时序特性

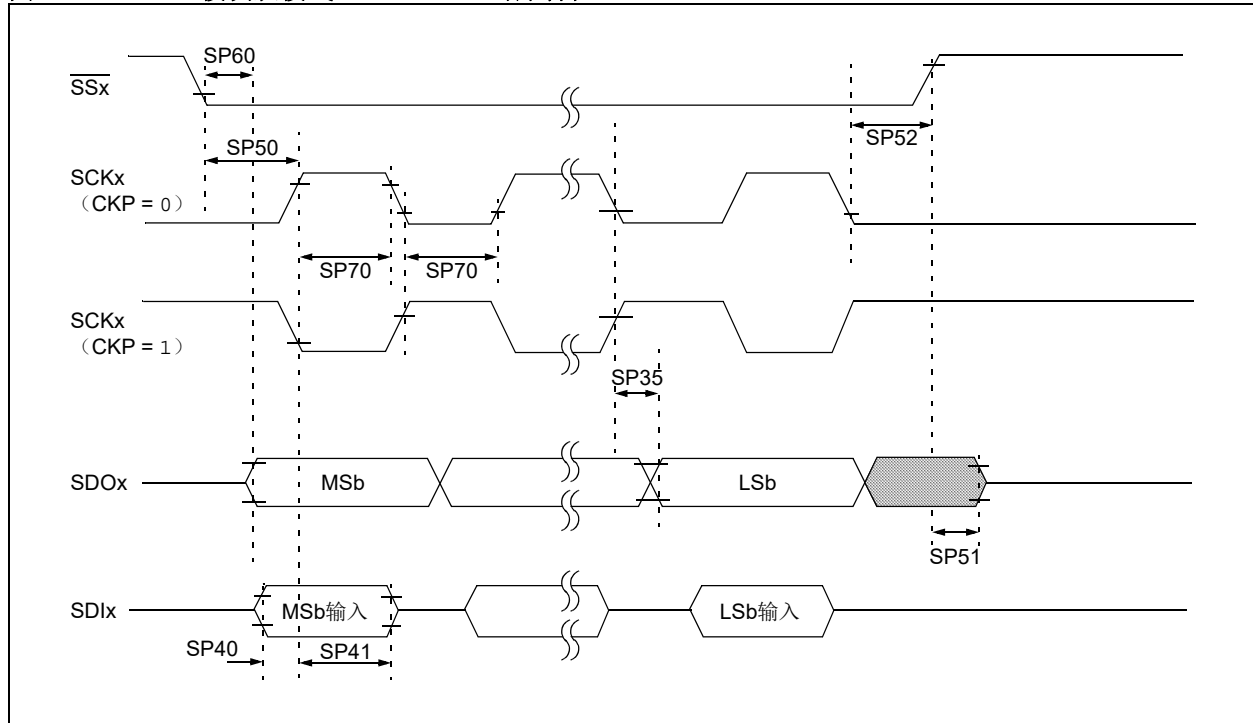


表31-29: SPIx模块从模式时序要求

工作条件 (除非另外声明): $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (工业级) $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ (扩展级)					
参数编号	符号	特性 <sup>(1)</sup>	最小值	最大值	单位
SP70	TscL, TschH	SCKx输入低电平时间或高电平时间	15	—	ns
SP35	Tsch2boV, TscL2boV	SCKx边沿之后 SDOx数据输出有效的的时间	—	20	ns
SP40	TdiV2sch, TdiV2scL	SDIx数据输入到 SCKx边沿的建立时间	10	—	ns
SP41	Tsch2diL, TscL2diL	SDIx数据输入到 SCKx边沿的保持时间	15	—	ns
SP50	Tssl2sch, Tssl2scL	$\overline{SSx}$ ↓ 到 SCKx ↓ 或 SCKx ↑ 输入的时间	120	—	ns
SP51	Tssh2boZ	$\overline{SSx}$ ↑ 到 SDOx输出高阻抗的时间	8	50	ns
SP52	Tsch2ssH, TscL2ssH	SCKx边沿之后 $\overline{SSx}$ ↑ 的时间	$1.5 T_{CY} + 40$	—	ns
SP60	Tssl2doV	$\overline{SSx}$ 边沿之后 SDOx数据输出有效的的时间	—	50	ns

注 1: 这些参数为特性值, 但未经生产测试。

图31-10： I2Cx总线起始位/停止位时序特性（主模式）

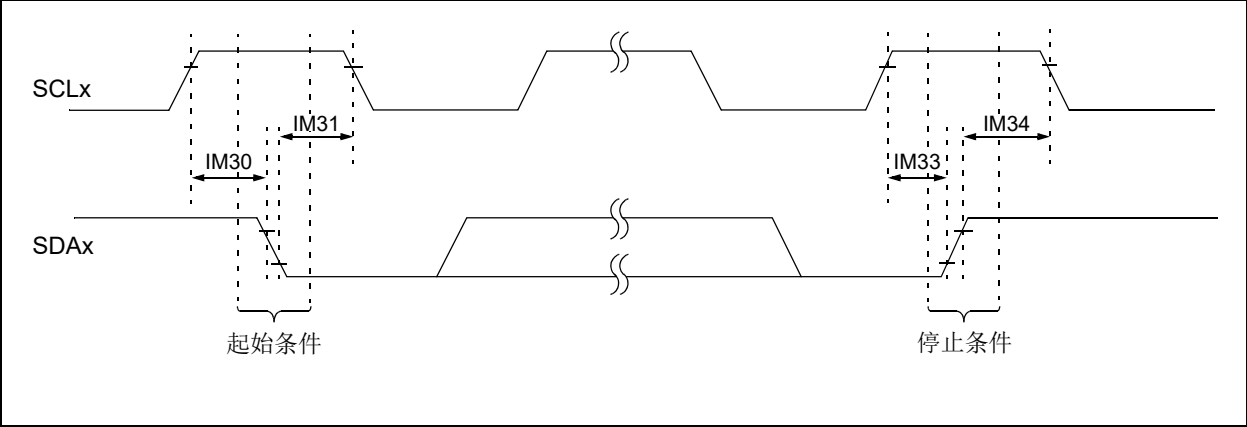


图31-11： I2Cx总线数据时序特性（主模式）

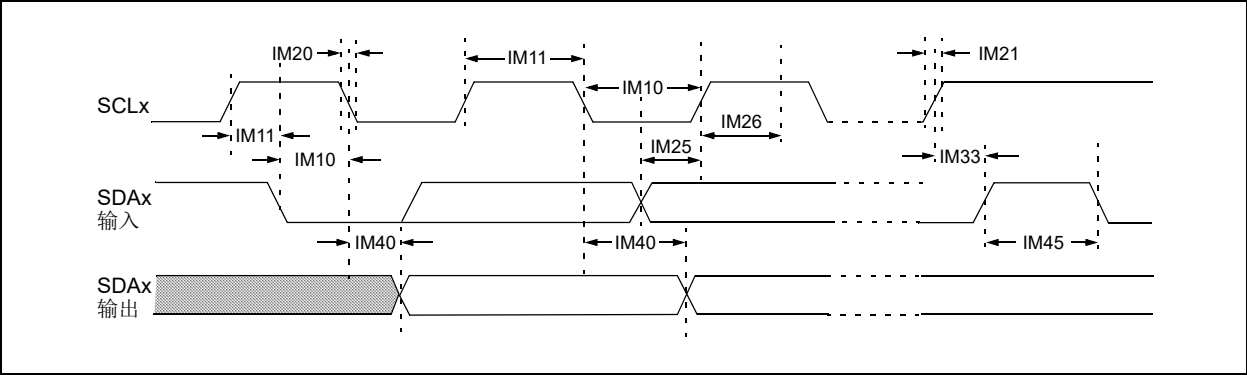


表31-30: I2Cx总线数据时序要求 (主模式)

工作条件 (除非另外声明): $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (工业级) $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ (扩展级)							
参数编号	符号	特性		最小值 <sup>(1)</sup>	最大值	单位	条件
IM10	TLO:SCL	时钟低电平时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
IM11	THI:SCL	时钟高电平时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
IM20	TF:SCL	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	$20 \times (V_{DD}/5.5V)$	300	ns	
			1 MHz 模式	$20 \times (V_{DD}/5.5V)$	120	ns	
IM21	TR:SCL	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	$20 + 0.1 C_B$	300	ns	
			1 MHz 模式	—	120	ns	
IM25	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	
			400 kHz 模式	100	—	ns	
			1 MHz 模式	50	—	ns	
IM26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	$\mu s$	
			400 kHz 模式	0	0.9	$\mu s$	
			1 MHz 模式	0	0.3	$\mu s$	
IM30	TSU:STA	起始条件建立时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	仅与重复起始条件相关
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
IM31	THD:STA	起始条件保持时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	这个周期后产生第一个时钟脉冲
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
IM33	TSU:STO	停止条件建立时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	$\mu s$	
IM34	THD:STO	停止条件保持时间	100 kHz 模式	$T_{CY} * (BRG + 1)$	—	ns	
			400 kHz 模式	$T_{CY} * (BRG + 1)$	—	ns	
			1 MHz 模式	$T_{CY} * (BRG + 1)$	—	ns	
IM40	TAA:SCL	自时钟边沿到输出有效的时间	100 kHz 模式	—	3450	ns	
			400 kHz 模式	—	900	ns	
			1 MHz 模式	—	450	ns	
IM45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	$\mu s$	在新的传输启动之前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	$\mu s$	
			1 MHz 模式	0.5	—	$\mu s$	
IM50	C <sub>B</sub>	总线容性负载	100 kHz 模式	—	400	pF	
			400 kHz 模式	—	400	pF	
			1 MHz 模式	—	10	pF	
IM51	TPGD	脉冲干扰抑制电路延时		65	390	ns	

注 1: BRG 为 I<sup>2</sup>C 波特率发生器的值。

图31-12: I2Cx总线起始位/停止位时序特性（从模式）

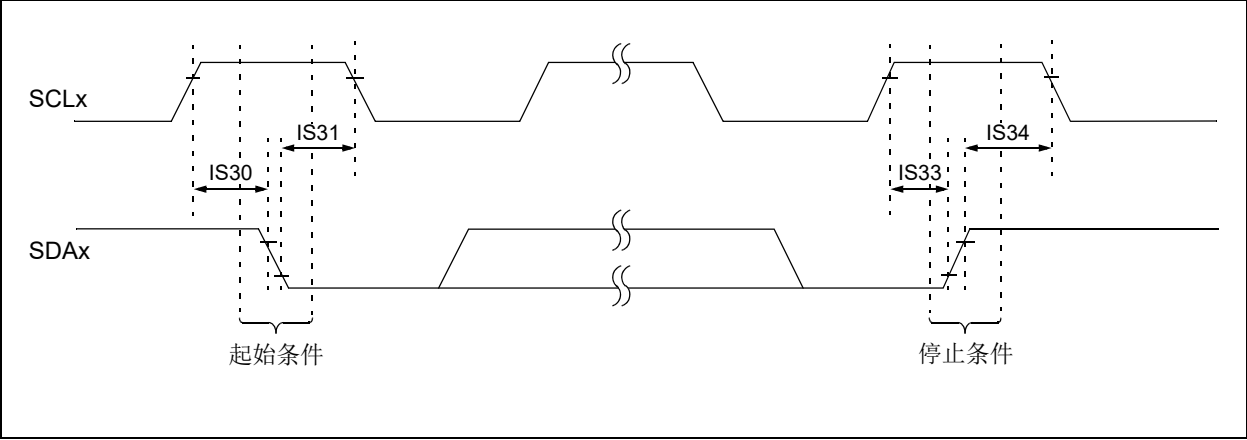


图31-13: I2Cx总线数据时序特性（从模式）

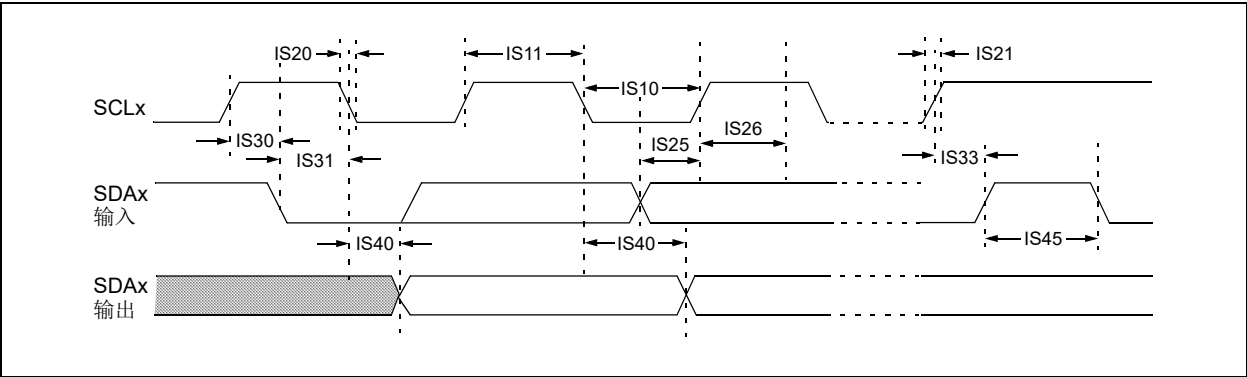


表31-31: I2Cx总线数据时序要求 (从模式)

工作条件 (除非另外声明): $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ (工业级) $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ (扩展级)							
参数编号	符号	特性	最小值	最大值	单位	条件	
IS10	TLO:SCL	时钟低电平时间	100 kHz 模式	4.7	—	$\mu s$	CPU 时钟必须至少为 800 kHz
			400 kHz 模式	1.3	—	$\mu s$	CPU 时钟必须至少为 3.2 MHz
			1 MHz 模式	0.5	—	$\mu s$	
IS11	THI:SCL	时钟高电平时间	100 kHz 模式	4.0	—	$\mu s$	CPU 时钟必须至少为 800 kHz
			400 kHz 模式	0.6	—	$\mu s$	CPU 时钟必须至少为 3.2 MHz
			1 MHz 模式	0.26	—	$\mu s$	
IS20	TF:SCL	SDAx 和 SCLx 下降时间	100 kHz 模式	—	300	ns	
			400 kHz 模式	$20 \times (V_{DD}/5.5V)$	300	ns	
			1 MHz 模式	$20 \times (V_{DD}/5.5V)$	120	ns	
IS21	TR:SCL	SDAx 和 SCLx 上升时间	100 kHz 模式	—	1000	ns	
			400 kHz 模式	$20 + 0.1 C_B$	300	ns	
			1 MHz 模式	—	120	ns	
IS25	TSU:DAT	数据输入建立时间	100 kHz 模式	250	—	ns	
			400 kHz 模式	100	—	ns	
			1 MHz 模式	50	—	ns	
IS26	THD:DAT	数据输入保持时间	100 kHz 模式	0	—	ns	
			400 kHz 模式	0	0.9	$\mu s$	
			1 MHz 模式	0	0.3	$\mu s$	
IS30	TSU:STA	起始条件建立时间	100 kHz 模式	4.7	—	$\mu s$	仅与重复起始条件相关
			400 kHz 模式	0.6	—	$\mu s$	
			1 MHz 模式	0.26	—	$\mu s$	
IS31	THD:STA	起始条件保持时间	100 kHz 模式	4.0	—	$\mu s$	这个周期后产生第一个时钟脉冲
			400 kHz 模式	0.6	—	$\mu s$	
			1 MHz 模式	0.26	—	$\mu s$	
IS33	TSU:STO	停止条件建立时间	100 kHz 模式	4.0	—	$\mu s$	
			400 kHz 模式	0.6	—	$\mu s$	
			1 MHz 模式	0.26	—	$\mu s$	
IS34	THD:STO	停止条件保持时间	100 kHz 模式	$> 0$	—	$\mu s$	
			400 kHz 模式	$> 0$	—	$\mu s$	
			1 MHz 模式	$> 0$	—	$\mu s$	
IS40	TAA:SCL	自时钟边沿到输出有效的 时间	100 kHz 模式	0	3.45	$\mu s$	
			400 kHz 模式	0	0.9	$\mu s$	
			1 MHz 模式	0	0.45	$\mu s$	
IS45	TBF:SDA	总线空闲时间	100 kHz 模式	4.7	—	$\mu s$	在新的传输启动之前总线必须保持空闲的时间
			400 kHz 模式	1.3	—	$\mu s$	
			1 MHz 模式	0.5	—	$\mu s$	
IS50	CB	总线容性负载	100 kHz 模式	—	400	pF	
			400 kHz 模式	—	400	pF	
			1 MHz 模式	—	10	pF	

# dsPIC33CK64MP105系列

图31-14: UARTx模块时序特性

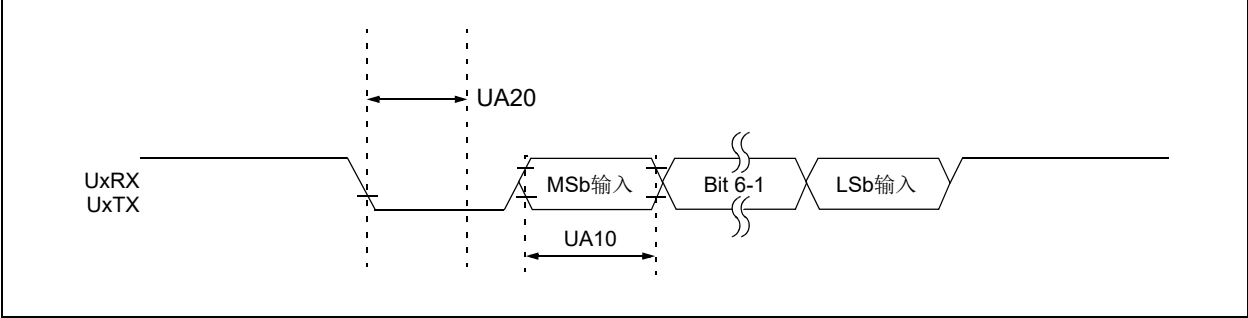


表31-32: UARTx模块时序要求

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）					
参数编号	符号	特性 <sup>(1)</sup>	最小值	最大值	单位
UA10	TUABAUD	UARTx波特率时间	40	—	ns
UA11	FBAUD	UARTx波特率	—	25	Mbps
UA20	TcWF	触发UARTx唤醒的起始位脉冲宽度	50	—	ns

注 1: 这些参数为特性值，但未经生产测试。



表31-33: ADC模块规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）							
参数编号	符号	特性	最小值	典型值 <sup>(2)</sup>	最大值	单位	条件
模拟输入							
AD12	V <sub>INH-V<sub>INL</sub></sub>	满量程输入范围	AV <sub>SS</sub>	—	AV <sub>DD</sub>	V	
AD14	V <sub>IN</sub>	绝对输入电压	AV <sub>SS</sub> - 0.3	—	AV <sub>DD</sub> + 0.3	V	
AD17	R <sub>IN</sub>	模拟电压源的推荐阻抗	—	100	—	Ω	对于最小采样时间
AD66	V <sub>BG</sub>	内部带隙输入电压	—	1.2	—	V	
ADC精度							
AD20c	N <sub>r</sub>	分辨率	12个数据位			位	
AD21c	INL	积分非线性	> -11.3	—	< 11.3	LSb	AV <sub>SS</sub> = 0V, AV <sub>DD</sub> = 3.3V
AD22c	DNL	微分非线性	> -1.5	—	< 11.5	LSb	AV <sub>SS</sub> = 0V, AV <sub>DD</sub> = 3.3V
AD23c	GERR	增益误差	> -12	—	< 12	LSb	AV <sub>SS</sub> = 0V, AV <sub>DD</sub> = 3.3V
AD24c	E <sub>OFF</sub>	失调误差	> -7.5	—	< 7.5	LSb	AV <sub>SS</sub> = 0V, AV <sub>DD</sub> = 3.3V
AD25c	—	单调性	—	—	—	—	保证
动态性能							
AD31b	SINAD <sup>(1)</sup>	信噪比和失真	56	—	70	dB	
AD34b	ENOB <sup>(1)</sup>	有效位数	9.0	—	11.4	位	

注 1: 这些参数为特性值，但未经生产测试；采用 1 kHz 正弦波信号确定特性值。

2: 除非另外声明，否则“典型值”列中的数据均为 3.3V 和 +25°C 条件下的值。这些参数仅供设计参考，未经测试。

表31-34: 模数转换时序要求

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ ， $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）					
参数编号	符号	特性	最小值	最大值	单位
AD50	T <sub>AD</sub>	ADC 时钟周期	14.28	—	ns
AD51	F <sub>TP</sub>	ADC 吞吐率（对于所有通道）	—	3.5	Msp/s

# dsPIC33CK64MP105系列

表31-35: 高速模拟比较器模块规范

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	备注
CM09	FIN	输入频率	400	500	550	MHz	
CM10	VI <sub>OFF</sub>	输入失调电压	-20	—	20	mV	
CM11	V <sub>ICM</sub>	输入共模电压范围 <sup>(1)</sup>	AV <sub>SS</sub>	—	AV <sub>DD</sub>	V	
CM13	CMRR	共模抑制比 <sup>(1)</sup>	65	—	—	dB	
CM14	T <sub>RESP</sub>	大信号响应	—	15	—	ns	V-输入保持为AV <sub>DD</sub> /2时，V+输入步长为100 mV
CM15	V <sub>HYST</sub>	输入滞后	15	—	45	mV	取决于 HYSSEL[1:0]

注 1: 这些参数仅供设计参考，未经生产测试。

表31-36: DAC模块规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）							
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	备注
DA02	CVRES	分辨率	12			位	
DA03	INL	积分非线性误差	-38	—	0	LSb	
DA04	DNL	微分非线性误差	-5	—	5	LSb	
DA05	EOFF	失调误差	-3.5	—	21.5	LSb	
DA06	EG	增益误差	0	—	41	%	
DA07	TSET	稳定时间	—	750	—	ns	对于10-90%或90-10%的阶跃输入，输出电压容差达到稳定在所需输出电压的2%范围内的时间
DA08	VOUT	输出电压范围	0.165	—	3.135	V	$V_{DD} = 3.3V$

注 1: 除非另外声明，否则“典型值”列中的数据均为3.3V和+25°C条件下的值。这些参数仅供设计参考，未经测试。

表31-37: DAC输出（DACOUT引脚）规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ , $-40^{\circ}C \leq T_A \leq +85^{\circ}C$ （工业级） $-40^{\circ}C \leq T_A \leq +125^{\circ}C$ （扩展级）							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
DA11	RLOAD	阻性输出负载阻抗	10K	—	—	$\Omega$	
DA11a	CLOAD	输出负载电容	—	—	35	pF	包括输出引脚电容
DA12	IOUT	输出电流驱动能力	—	3	—	mA	灌电流和拉电流

# dsPIC33CK64MP105系列

表31-38： 偏流发生器规范<sup>(1)</sup>

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	最大值	单位
CC03	I10SRC	10 μA 拉电流	8.8	11.2	μA
CC04	I50SRC	50 μA 拉电流	44	56	μA
CC05	I50SNK	50 μA 灌电流	-44	-56	μA

注 1： 参数为特性值，但未经生产测试。

表31-39： 运算放大器规范<sup>(1)</sup>

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V， -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	备注
OAMP1	GBWP	增益带宽积	—	20	—	MHz	
OAMP2	SR	压摆率	—	40	—	V/μs	
OAMP3	VIOFF	输入失调电压	-20	—	20	mV	
OAMP4	VICM	共模输入电压范围	AVSS	—	AVDD	V	NCHDISx = 0
			AVSS	—	2.8	V	NCHDISx = 1
OAMP5	CMRR	共模抑制比	—	68	—	db	
OAMP6	PSRR	电源抑制比	—	74	—	dB	
OAMP7	VOR	输出电压范围	AVSS	—	AVDD	mV	0.5V输入过驱动， 没有输出负载

注 1： 这些参数仅供设计参考，未经生产测试。

## 32.0 高温电气特性

本章将对 dsPIC33CK64MP105 系列器件在 -40°C 至 +150°C 环境温度范围内工作的电气特性进行概括介绍。

除本章中列出的参数外，-40°C 至 +150°C 范围内的规范与第 31.0 节“电气特性”中给出的在 -40°C 至 +125°C 范围内工作的规范相同。

本章中的参数都以 H 开头，H 表示高温。例如，第 31.0 节“电气特性”中的参数 DC20 是 HDC20 在工业级和扩展级温度时的等效参数。

下面列出了 dsPIC33CK64MP105 系列高温器件的绝对最大值。器件长时间工作在最大值条件下，其可靠性可能受到影响。未表明器件处于或超过本规范指定的极限值条件下仍可正常工作。

### 绝对最大值<sup>(1)</sup>

偏置时的环境温度 .....	-40°C 至 +150°C
储存温度 .....	-65°C 至 +150°C
VDD 引脚相对于 VSS 的电压 .....	-0.3V 至 +4.0V
任一非 5V 耐压引脚相对于 VSS 的电压 <sup>(3)</sup> .....	-0.3V 至 (VDD + 0.3V)
任一 5V 耐压引脚相对于 VSS 的电压 (VDD ≥ 3.0V) <sup>(3)</sup> .....	-0.3V 至 +5.5V
任一 5V 耐压引脚相对于 VSS 的电压 (VDD < 3.0V) <sup>(3)</sup> .....	-0.3V 至 +3.6V
流出 VSS 引脚的最大电流 .....	300 mA
流入 VDD 引脚的最大电流 <sup>(2)</sup> .....	300 mA
任一常规 I/O 引脚的最大灌电流/拉电流 .....	15 mA
电流驱动能力增强的 I/O 引脚 (RB1、RC8、RC9 和 RD8) 的最大灌电流/拉电流 .....	25 mA
两个 VSS 引脚之间的一组 I/O 的最大灌电流 <sup>(4)</sup> .....	75 mA
两个 VDD 引脚之间的一组 I/O 的最大拉电流 <sup>(4)</sup> .....	75 mA
所有 I/O 的最大灌电流 <sup>(2,5)</sup> .....	200 mA
所有 I/O 的最大拉电流 <sup>(2,5)</sup> .....	200 mA

**注 1:** 1 如果器件的工作条件超过上述“绝对最大值”，可能对器件造成永久性损坏。上述值仅代表本规范规定的极限工作条件，未表明器件在上述极限值或超出极限值的情况下仍可正常工作。器件长时间工作在最大值条件下，其可靠性可能受到影响。

**2:** 允许的最大电流由器件最大功耗决定（见表 32-2）。

**3:** 关于 5V 耐压的引脚，请参见“引脚分配图”一节。

**4:** 不适用于 AVDD 和 AVSS 引脚。

**5:** 对于 28 引脚封装，所有 I/O 的最大灌电流/拉电流限制为 150 mA。

# dsPIC33CK64MP105系列

## 32.1 直流特性

表32-1: 工作MIPS与电压

V <sub>DD</sub> 范围	温度范围	最大CPU时钟频率
3.0V至3.6V	-40°C至+150°C	70

表32-2: 热工作条件

参数	符号	最小值	最大值	单位
高温器件				
工作结温范围	T <sub>J</sub>	-40	+165	°C
工作环境温度范围	T <sub>A</sub>	-40	+150	°C
功耗: 芯片内部功耗: $P_{INT} = V_{DD} \times (I_{DD} - \sum I_{OH})$ I/O 引脚功耗: $I/O = \sum (\{V_{DD} - V_{OH}\} \times I_{OH}) + \sum (V_{OL} \times I_{OL})$	P <sub>D</sub>	P <sub>INT</sub> + P <sub>I/O</sub>		W
最大允许功耗	P <sub>DMAX</sub>	(T <sub>J</sub> - T <sub>A</sub> )/θ <sub>JA</sub>		W

表32-3: 热封装特性<sup>(1)</sup>

封装	符号	典型值	单位
48引脚TQFP 7x7 mm	θ <sub>JA</sub>	62.76	°C/W
48引脚UQFN 6x6 mm	θ <sub>JA</sub>	27.6	°C/W
36引脚UQFN 5x5 mm	θ <sub>JA</sub>	29.2	°C/W
28引脚UQFN 6x6 mm	θ <sub>JA</sub>	22.41	°C/W
28引脚SSOP 5.30 mm	θ <sub>JA</sub>	52.84	°C/W

注 1: 通过封装模拟获得结与环境的热阻 (θ<sub>JA</sub>) 值。

表32-4: 工作电压规范

工作条件 (除非另外声明): -40°C ≤ T <sub>A</sub> ≤ +150°C (高温)						
参数编号	符号	特性	最小值	最大值	单位	条件
HDC10	V <sub>DD</sub>	电源电压	3.0	3.6	V	
HDC16	V <sub>POR</sub>	确保内部上电复位信号的V <sub>DD</sub> 启动电压	—	V <sub>SS</sub>	V	
HDC17	S <sub>VDD</sub>	确保内部上电复位信号的V <sub>DD</sub> 上升速率	0.03	—	V/ms	0V-3V/100 ms
HBO10	V <sub>BOR</sub> <sup>(1)</sup>	V <sub>DD</sub> 跳变 (高电压到低电压) 时的BOR事件	2.68	2.99	V	

注 1: 器件可在V<sub>BORMIN</sub> < V<sub>DD</sub> < V<sub>DDMIN</sub>条件下工作, 但此时模拟模块 (ADC和比较器) 的性能可能下降。V<sub>BOR</sub>参数仅供设计参考, 未经生产测试。

表 32-5: 工作电流 (I<sub>DD</sub>) (2)

参数编号	典型值 (1)	最大值	单位	条件		
HDC20	10.2	23.4	mA	+150°C	3.3V	10 MIPS (N1 = 1, N2 = 5, N3 = 2, M = 50, Fvco = 400 MHz, FPLLO = 40 MHz)
HDC21	12.2	25.5	mA	+150°C	3.3V	20 MIPS (N1 = 1, N2 = 5, N3 = 1, M = 50, Fvco = 400 MHz, FPLLO = 80 MHz)
HDC22	15.5	30.1	mA	+150°C	3.3V	40 MIPS (N1 = 1, N2 = 3, N3 = 1, M = 60, Fvco = 480 MHz, FPLLO = 160 MHz)
HDC23	21.2	41.34	mA	+150°C	3.3V	70 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 70, Fvco = 560 MHz, FPLLO = 280 MHz)

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本运行电流 (I<sub>DD</sub>) 的测量条件如下:

- 通过软件将振荡器切换为EC+PLL 模式
- OSC1 引脚通过电压为0.3V至V<sub>DD</sub> - 0.3V的外部8 MHz方波驱动
- OSC2通过配置字配置为I/O (OSCIOFCN (FOSC[2]) = 0)
- 禁止FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWDTC[15] = 0 且 WDTCONL[15] = 0)
- 所有I/O 引脚 (OSC1 除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的PMDx位均置1)
- 禁止JTAG (JTAGEN (FICD[5]) = 0)
- 在while(1) 循环中执行NOP指令

# dsPIC33CK64MP105系列

表 32-6: 空闲电流 (I<sub>IDLE</sub>) <sup>(2)</sup>

参数编号	典型值 <sup>(1)</sup>	最大值	单位	条件		
HDC40	9.0	23.22	mA	+150°C	3.3V	10 MIPS (N1 = 1, N2 = 5, N3 = 2, M = 50, F <sub>VCO</sub> = 400 MHz, F <sub>PLLO</sub> = 40 MHz)
HDC41	9.7	23.4	mA	+150°C	3.3V	20 MIPS (N1 = 1, N2 = 5, N3 = 1, M = 50, F <sub>VCO</sub> = 400 MHz, F <sub>PLLO</sub> = 80 MHz)
HDC42	11.2	25.6	mA	+150°C	3.3V	40 MIPS (N1 = 1, N2 = 3, N3 = 1, M = 60, F <sub>VCO</sub> = 480 MHz, F <sub>PLLO</sub> = 160 MHz)
HDC43	13.4	27.9	mA	+150°C	3.3V	70 MIPS (N1 = 1, N2 = 2, N3 = 1, M = 70, F <sub>VCO</sub> = 560 MHz, F <sub>PLLO</sub> = 280 MHz)

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本空闲电流 (I<sub>IDLE</sub>) 的测量条件如下:

- 通过软件将振荡器切换为EC+PLL 模式
- OSC1 引脚通过电压为0.3V至V<sub>DD</sub> - 0.3V的外部8 MHz方波驱动
- OSC2通过配置字配置为I/O (OSCIOFCN (FOSC[2]) = 0)
- 禁止FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWD<sub>TD</sub>[15] = 0且WDTCONL[15] = 0)
- 所有I/O引脚 (OSC1除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的PMDx位均置1)
- 禁止JTAG (JTAGEN (FICD[5]) = 0)
- 闪存稳压器处于待机模式 (NVMSIDL (NVMCON[12]) = 1)



表 32-7: 掉电电流 (IPD) <sup>(2)</sup>

参数编号	特性	典型值 <sup>(1)</sup>	最大值	单位	条件	
HDC60	基本掉电电流	6.3	19.8	mA	+150°C	3.3V

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

2: 基本休眠电流 (IPD) 的测量条件如下:

- OSC1 引脚通过电压为 0.3V 至  $V_{DD} - 0.3V$  的外部 8 MHz 方波驱动
- OSC2 通过配置字配置为 I/O (OSCIOFCN (FOSC[2]) = 0)
- 禁止 FSCM (FCKSM[1:0] (FOSC[7:6]) = 01)
- 禁止看门狗定时器 (FWDTC[15] = 0 且 WDTCONL[15] = 0)
- 所有 I/O 引脚 (OSC1 除外) 均配置为输出并驱动为低电平
- 所有外设模块都不工作, 也不为其提供时钟 (所有定义的 PMDx 位均置 1)
- 禁止 JTAG (JTAGEN (FICD[5]) = 0)
- 稳压器处于待机模式 (VREGS (RCON[8]) = 0)
- 稳压器处于低功耗模式 (LPWREN (VREGCON[15]) = 1)

表 32-8: 打盹电流 (IDoZE)

参数编号	典型值 <sup>(1)</sup>	打盹模式 时钟分频比	单位	条件			
HDC70	17.9	35.6	1:2	mA	+150°C	3.3V	70 MIPS (N = 1, N2 = 2, N3 = 1, M = 70, FVCO = 560 MHz, FPLLO = 280 MHz)
	13.6	29.1	1:128	mA			

注 1: “典型值”列中的数据仅供设计参考, 未经测试。

表 32-9: 看门狗定时器增量电流 ( $\Delta I_{WDT}$ ) <sup>(1)</sup>

参数编号	典型值	单位	条件		
HDC61	24	120	$\mu A$	+150°C	3.3V

注 1:  $\Delta I_{WDT}$  电流为当模块使能时额外消耗的电流。此电流应被加到基本 IPD 电流。所有参数均为特性值, 但未经生产测试。

# dsPIC33CK64MP105系列

表 32-10: PWM增量电流 <sup>(1)</sup>

参数编号	典型值	最大值	单位	条件		
HDC100	5.48	7.0	mA	+150°C	3.3V	PWM 输出频率 = 500 kHz, PWM输入 (AFPLLO = 500 MHz) (AVCO = 1000 MHz, PLLFBD = 125, APLLDIV1 = 2)
HDC101	4.44	5.7	mA	+150°C	3.3V	PWM 输出频率 = 500 kHz, PWM输入 (AFPLLO = 400 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 1)
HDC102	2.31	3.7	mA	+150°C	3.3V	PWM 输出频率 = 500 kHz, PWM输入 (AFPLLO = 200 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 2)
HDC103	1.22	2.3	mA	+150°C	3.3V	PWM 输出频率 = 500 kHz, PWM输入 (AFPLLO = 100 MHz), (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 4)

注 1: 不包括APLL 电流。如果多个PWM在运行, 则APLL 电流将相同。当HREN (PGxCONL[7]) = 0时, 列出的增量电流仅针对一个PWM实例。所有参数均为特性值, 但未经生产测试。

表 32-11: APLL增量电流

参数编号	典型值	最大值	单位	条件 <sup>(1)</sup>		
HDC110	7.04	9.3	mA	+150°C	3.3V	AFPLLO = 500 MHz (AVCO = 1000 MHz, PLLFBD = 125, APLLDIV1 = 2)
HDC111	3.78	5.8	mA	+150°C	3.3V	AFPLLO = 400 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 1)
HDC112	2.49	4.5	mA	+150°C	3.3V	AFPLLO = 200 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 2)
HDC113	1.83	3.6	mA	+150°C	3.3V	AFPLLO = 100 MHz (AVCO = 400 MHz, PLLFBD = 50, APLLDIV1 = 4)

注 1: 如果多个PWM或DAC基于APLL时钟运行, 则APLL 电流将相同。所有参数均为特性值, 但未经生产测试。

表 32-12: ADC增量电流 <sup>(1)</sup>

参数编号	典型值	最大值	单位	条件		
HDC120	3.76	6.8	mA	+150°C	3.3V	TAD = 14.3 ns (3.5 Msps 转换速率)

注 1: 共用内核连续转换。TAD = 14.3 ns (3.5 Msps 转换速率)。列出的增量电流仅针对一个ADC 内核。所有参数均为特性值, 但未经生产测试。

**表32-13: 比较器 + DAC增量电流**

参数编号	典型值	最大值	单位	条件		
HDC130	1.25	1.65	mA	+150°C	3.3V	AFPLLO @ 500 MHz <sup>(1)</sup>

注 1: 不包括APLL 电流。列出的增量电流仅针对一个比较器 + DAC实例。所有参数均为特性值，但未经生产测试。

**表32-14: 运放增量电流<sup>(1)</sup>**

参数编号	典型值	最大值	单位	条件	
HDC140	0.58	2.3	mA	+150°C	3.3V

注 1: 列出的增量电流仅针对一个运放实例。所有参数均为特性值，但未经生产测试。

**表32-15: I/O引脚输入规范**

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V -40°C < TA < +150°C（高温）					
参数编号	符号	特性	最小值 <sup>(3)</sup>	最大值 <sup>(4)</sup>	单位
HDI50	IIL	输入泄漏电流 <sup>(1)</sup>			
		5V 耐压的 I/O 引脚 <sup>(2)</sup>	-1	1	μA
		非5V 耐压的 I/O 引脚 <sup>(2)</sup>	-1	1	μA
		MCLR	-1	1	μA
		OSCI	-1	1	μA

注 1: 负电流定义为引脚的拉电流。

2: 关于5V 耐压的 I/O 引脚，请参见“引脚分配图”部分。

3: VPIN = VSS。

4: VPIN = VDD。

**表32-16: 内部 FRC精度**

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V -40°C < TA < +150°C（高温）				
参数编号	特性	最小值	最大值	单位
HF20a	FRC @ 8 MHz <sup>(1)</sup>	-3	+3	%

注 1: 频率在+25°C和3.3V下校准。

**表32-17: 内部 LPRC精度**

工作条件（除非另外声明）： 3.0V ≤ VDD ≤ 3.6V -40°C < TA < +150°C（高温）				
参数编号	特性	最小值	最大值	单位
HF21	LPRC @ 32 kHz	-27	+27	%

# dsPIC33CK64MP105系列

表32-18: ADC模块精度<sup>(1)</sup>

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ $-40^{\circ}C < T_A < +150^{\circ}C$ （高温）						
参数编号	符号	特性	最小值	最大值	单位	条件
HAD23c	GERR	增益误差	$> -17.5$	$< 17.5$	LSb	$AV_{SS} = 0V, AV_{DD} = 3.3V$
HAD24c	EOFF	失调误差	$> -15$	$< 15$	LSb	$AV_{SS} = 0V, AV_{DD} = 3.3V$

注 1: ADC模块可在 $V_{BORMIN} < V_{DD} < V_{DDMIN}$ 条件下工作，但其性能将下降。除非另外声明，否则模块功能可确保，但未确定特性值。

表32-19: DAC模块规范

工作条件（除非另外声明）： $3.0V \leq V_{DD} \leq 3.6V$ $-40^{\circ}C < T_A < +150^{\circ}C$ （高温）						
参数编号	符号	特性	最小值	最大值	单位	备注
HDA03	INL	积分非线性误差	-45	0	LSb	

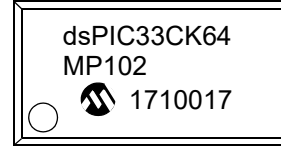
## 33.0 封装信息

### 33.1 封装标识信息

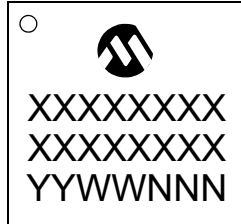
28引脚SSOP（5.30 mm）



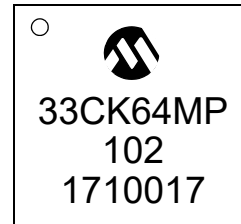
示例



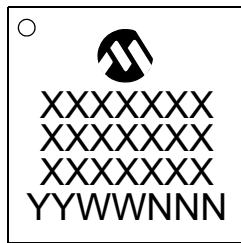
28引脚UQFN（6x6 mm）



示例



36引脚UQFN（5x5 mm）



示例



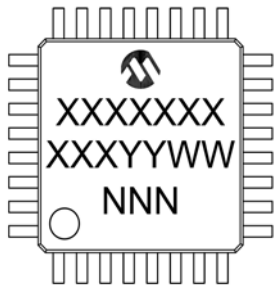
**图注:**    XX...X    客户指定信息  
              Y        年份代码（日历年的最后一位数字）  
              YY        年份代码（日历年的最后两位数字）  
              WW        星期代码（一月一日的星期代码为“01”）  
              NNN       由字母数字组成的追踪代码

**注:**        Microchip 部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指定信息的字符数。

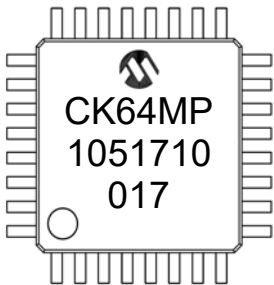
# dsPIC33CK64MP105系列

## 33.1 封装标识信息（续）

48引脚TQFP（7x7 mm）



示例



48引脚UQFN（6x6 mm）



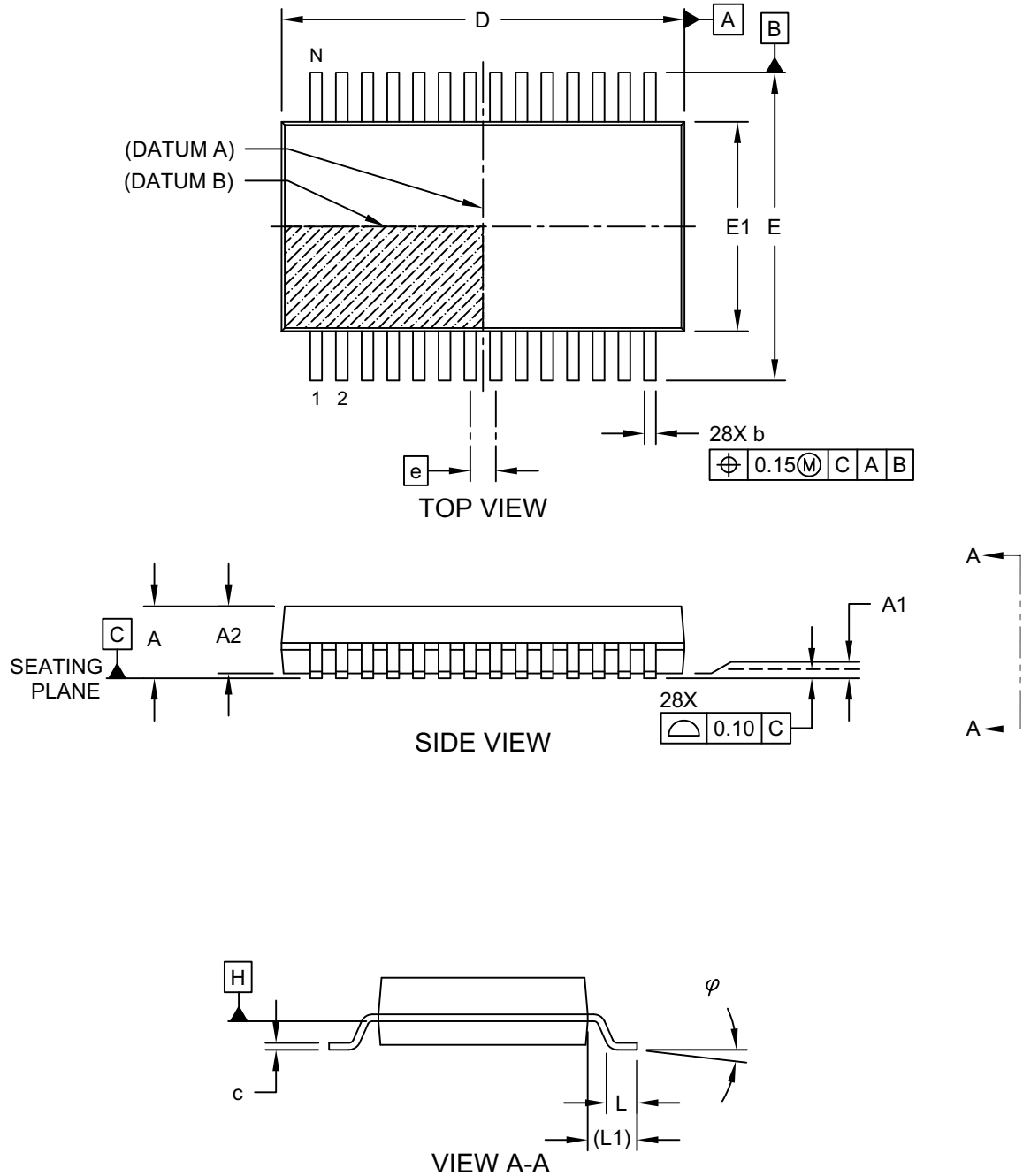
示例



## 33.2 封装详细信息

### 28引脚塑封紧缩小外形封装（SS）——主体5.30 mm [SSOP]

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

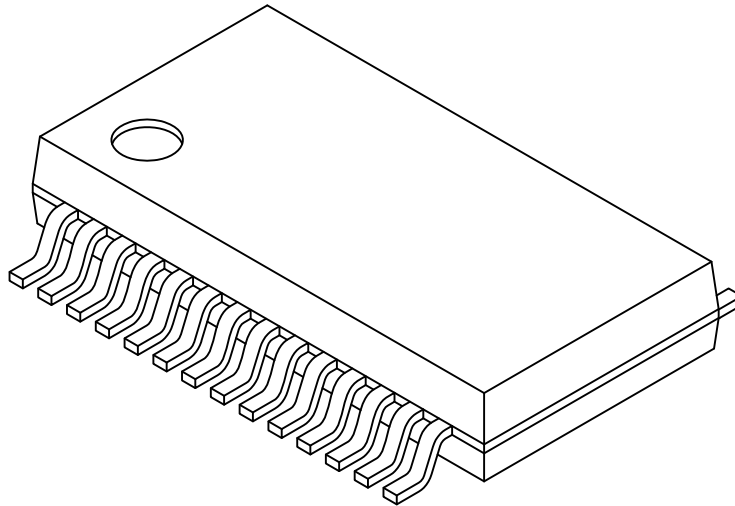


Microchip Technology Drawing C04-073 Rev C Sheet 1 of 2

# dsPIC33CK64MP105系列

## 28引脚塑封紧缩小外形封装（SS）——主体5.30 mm [SSOP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	-	-	2.00
Molded Package Thickness	A2	1.65	1.75	1.85
Standoff	A1	0.05	-	-
Overall Width	E	7.40	7.80	8.20
Molded Package Width	E1	5.00	5.30	5.60
Overall Length	D	9.90	10.20	10.50
Foot Length	L	0.55	0.75	0.95
Footprint	L1	1.25 REF		
Lead Thickness	c	0.09	-	0.25
Foot Angle	$\varphi$	0°	4°	8°
Lead Width	b	0.22	-	0.38

### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.20mm per side.

3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

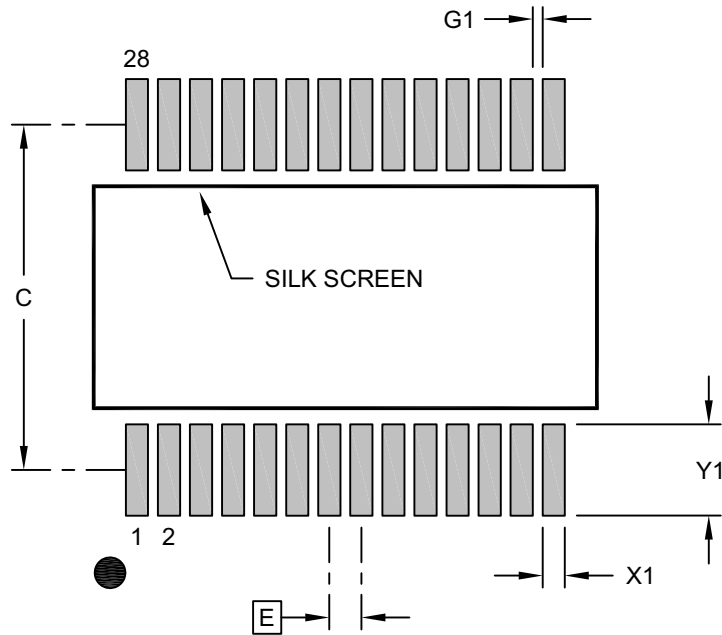
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-073 Rev C Sheet 2 of 2



## 28引脚塑封紧缩小外形封装（SS）——主体5.30 mm [SSOP]

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



### RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Contact Pad Spacing	C		7.00	
Contact Pad Width (X28)	X1			0.45
Contact Pad Length (X28)	Y1			1.85
Contact Pad to Center Pad (X26)	G1	0.20		

#### Notes:

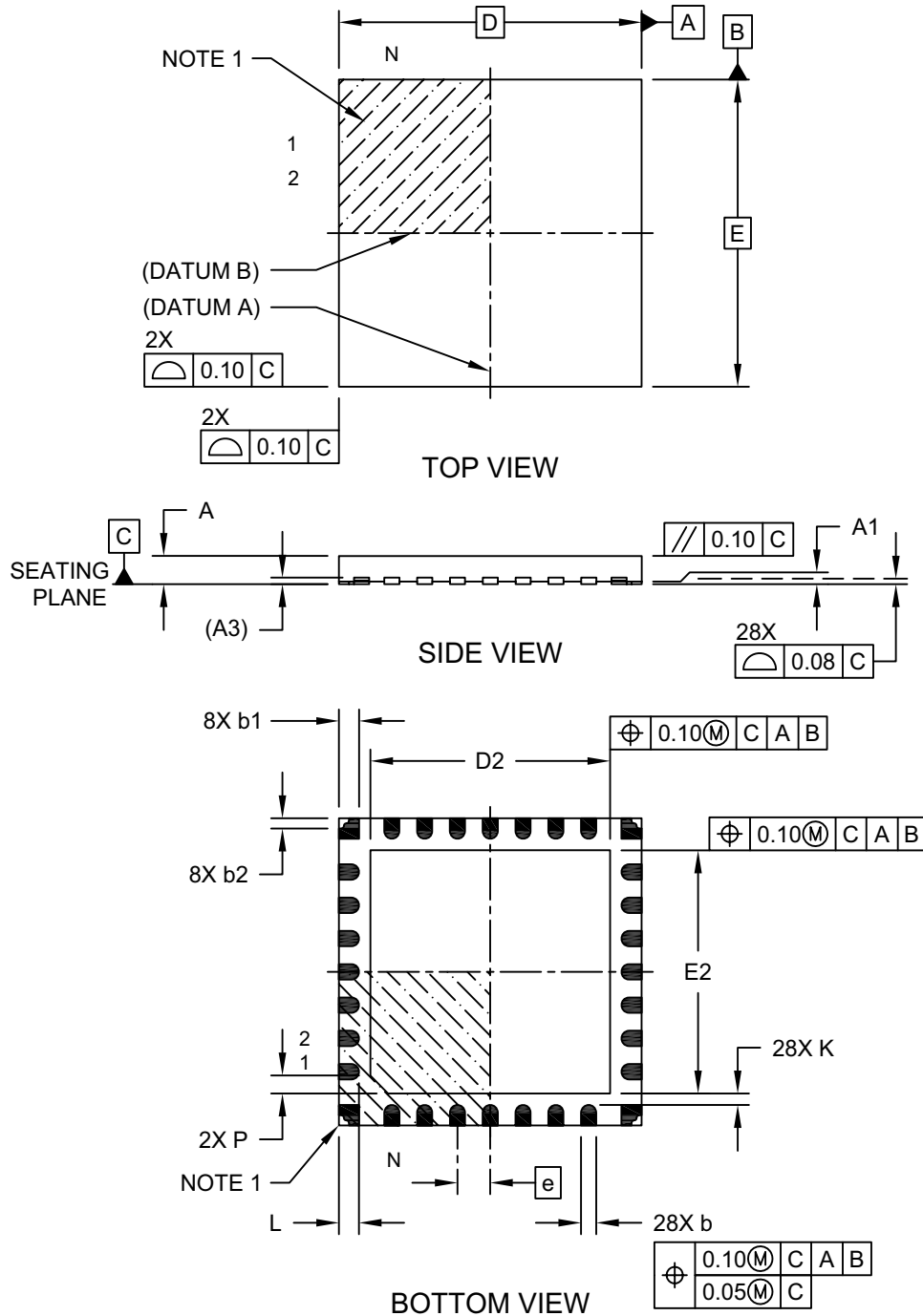
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2073 Rev B

# dsPIC33CK64MP105系列

**28引脚塑封超薄正方扁平无引线封装（2N）——主体6x6x0.55 mm [UQFN]，带4.65x4.65 mm外露焊盘和角锚点**

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

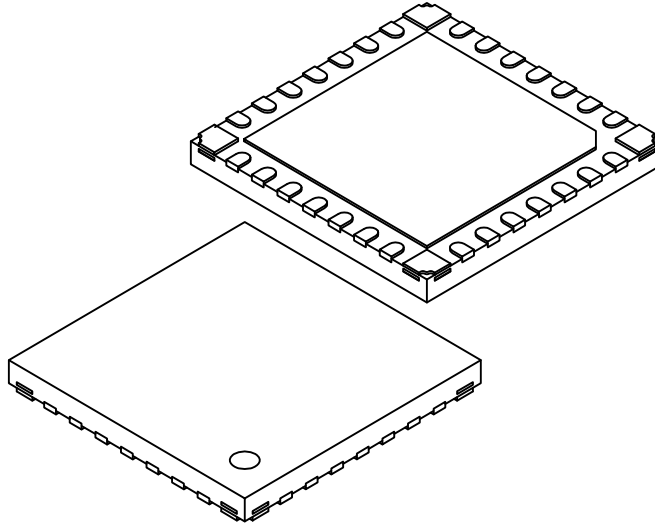


Microchip Technology Drawing C04-385 Rev C Sheet 1 of 2

# dsPIC33CK64MP105系列

28引脚塑封超薄正方扁平无引线封装（2N）——主体6x6x0.55 mm [UQFN]，带4.65x4.65 mm外露焊盘和角锚点

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N	28		
Pitch	e	0.65 BSC		
Overall Height	A	0.45	0.50	0.55
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.127 REF		
Overall Width	E	6.00 BSC		
Exposed Pad Width	E2	4.55	4.65	4.75
Overall Length	D	6.00 BSC		
Exposed Pad Length	D2	4.55	4.65	4.75
Exposed Pad Corner Chamfer	P	-	0.35	-
Terminal Width	b	0.25	0.30	0.35
Corner Anchor Pad	b1	0.35	0.40	0.43
Corner Pad, Metal Free Zone	b2	0.15	0.20	0.25
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K	0.20	-	-

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

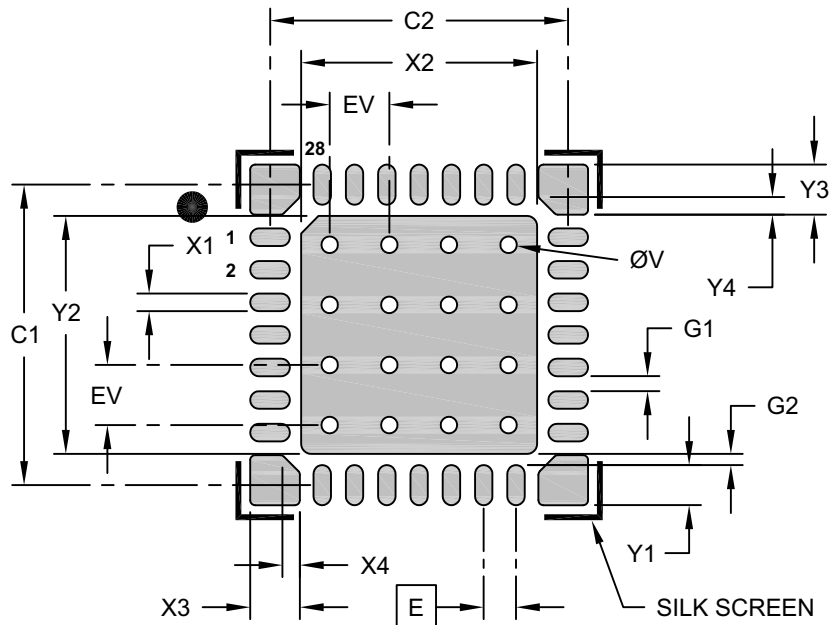
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-385 Rev C Sheet 2 of 2

# dsPIC33CK64MP105系列

**28引脚塑封超薄正方扁平无引线封装（2N）——主体6x6x0.55 mm [UQFN]，带4.65x4.65 mm外露焊盘和角锚点**

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



RECOMMENDED LAND PATTERN

Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E	0.65 BSC		
Optional Center Pad Width	X2			4.75
Optional Center Pad Length	Y2			4.75
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X28)	X1			0.35
Contact Pad Length (X28)	Y1			0.80
Corner Anchor (X4)	X3			1.00
Corner Anchor (X4)	Y3			1.00
Corner Anchor Chamfer (X4)	X4			0.35
Corner Anchor Chamfer (X4)	Y4			0.35
Contact Pad to Pad (X28)	G1	0.20		
Contact Pad to Center Pad (X28)	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

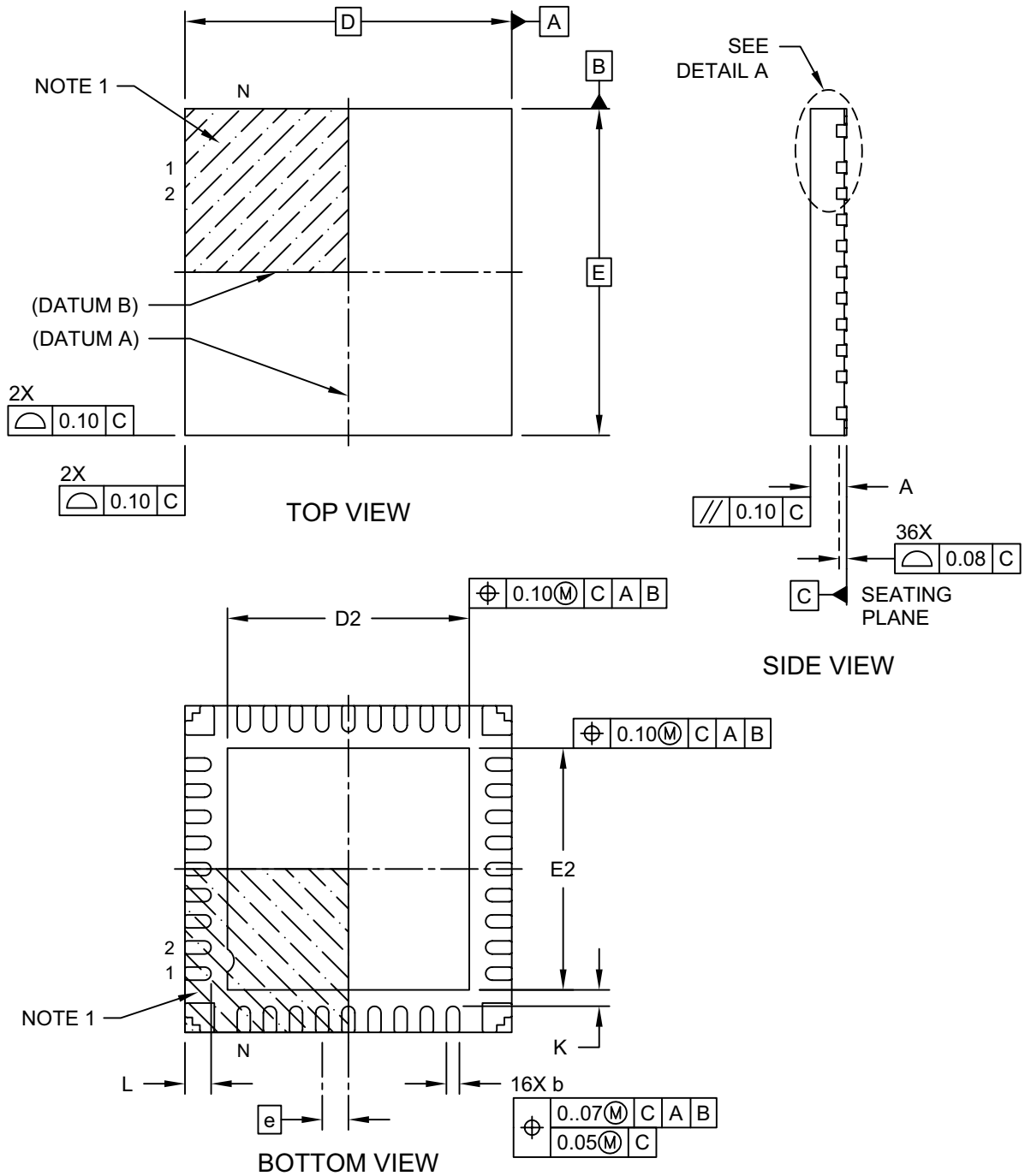
Notes:

1. Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2385B

## 36引脚塑封超薄正方扁平无引线封装（M5）——主体5x5 mm [UQFN]，带角锚点

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

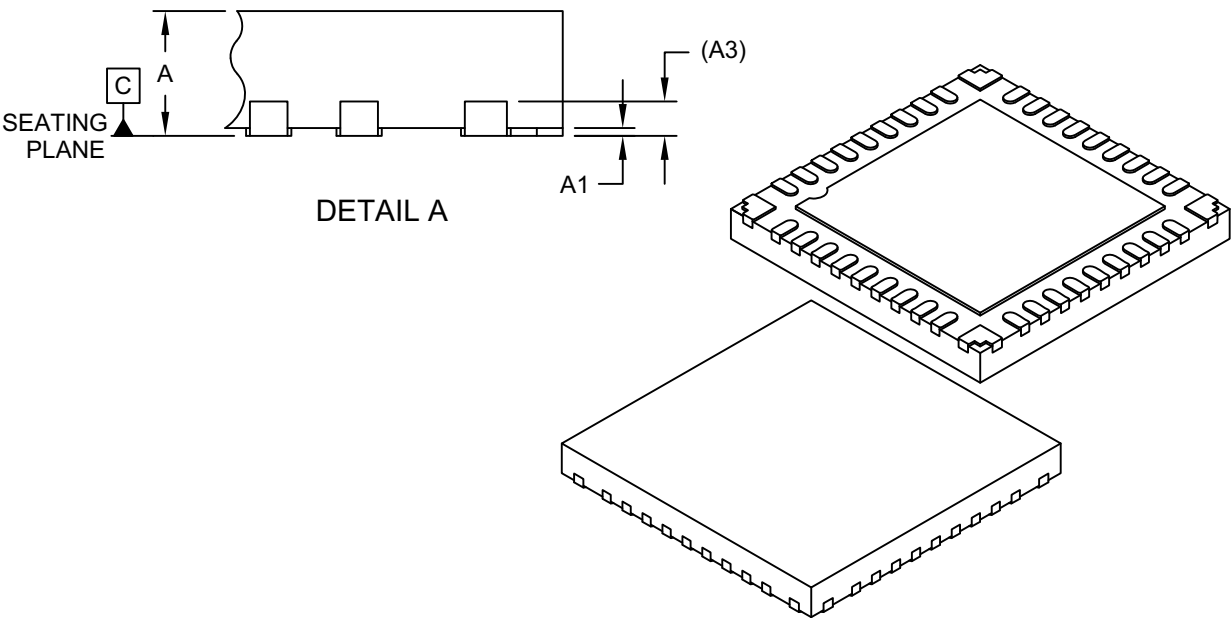


Microchip Technology Drawing C04-436-M5 Rev B Sheet 1 of 2

# dsPIC33CK64MP105系列

## 36 引脚塑封超薄正方扁平无引线封装（M5）——主体 5x5 mm [UQFN]，带角锚点

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Terminals	N	36		
Pitch	e	0.40 BSC		
Overall Height	A	0.50	0.55	0.60
Standoff	A1	0.00	0.02	0.05
Terminal Thickness	A3	0.152 REF		
Overall Length	D	5.00 BSC		
Exposed Pad Length	D2	3.60	3.70	3.80
Overall Width	E	5.00 BSC		
Exposed Pad Width	E2	3.60	3.70	3.80
Terminal Width	b	0.15	0.20	0.25
Terminal Length	L	0.30	0.40	0.50
Terminal-to-Exposed-Pad	K	0.25 REF		

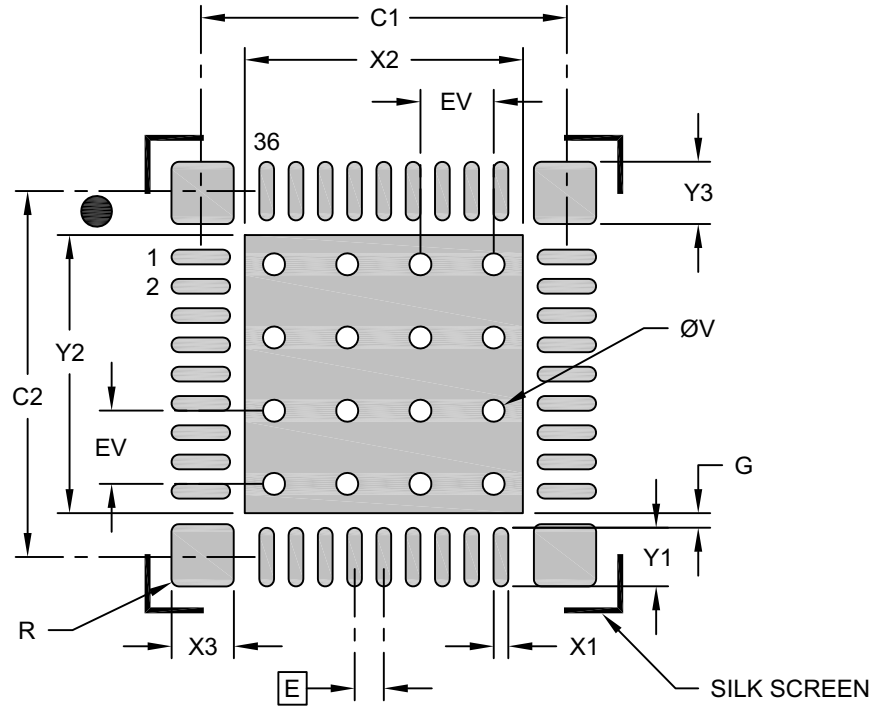
Notes:

- 1. Pin 1 visual index feature may vary, but must be located within the hatched area.
- 2. Package is saw singulated
- 3. Dimensioning and tolerancing per ASME Y14.5M
  - BSC: Basic Dimension. Theoretically exact value shown without tolerances.
  - REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-436-M5 Rev B Sheet 2 of 2

## 36引脚塑封超薄正方扁平无引线封装（M5）——主体5x5 mm [UQFN]，带角锚点

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



### RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Center Pad Width	X2			3.80
Center Pad Length	Y2			3.80
Contact Pad Spacing	C1		5.00	
Contact Pad Spacing	C2		5.00	
Contact Pad Width (X36)	X1			0.20
Contact Pad Length (X36)	Y1			0.80
Corner Pad Width (X4)	X3			0.85
Corner Pad Length (X4)	Y3			0.85
Corner Pad Radius	R		0.10	
Contact Pad to Center Pad (X36)	G	0.20		
Thermal Via Diameter	V		0.30	
Thermal Via Pitch	EV		1.00	

#### Notes:

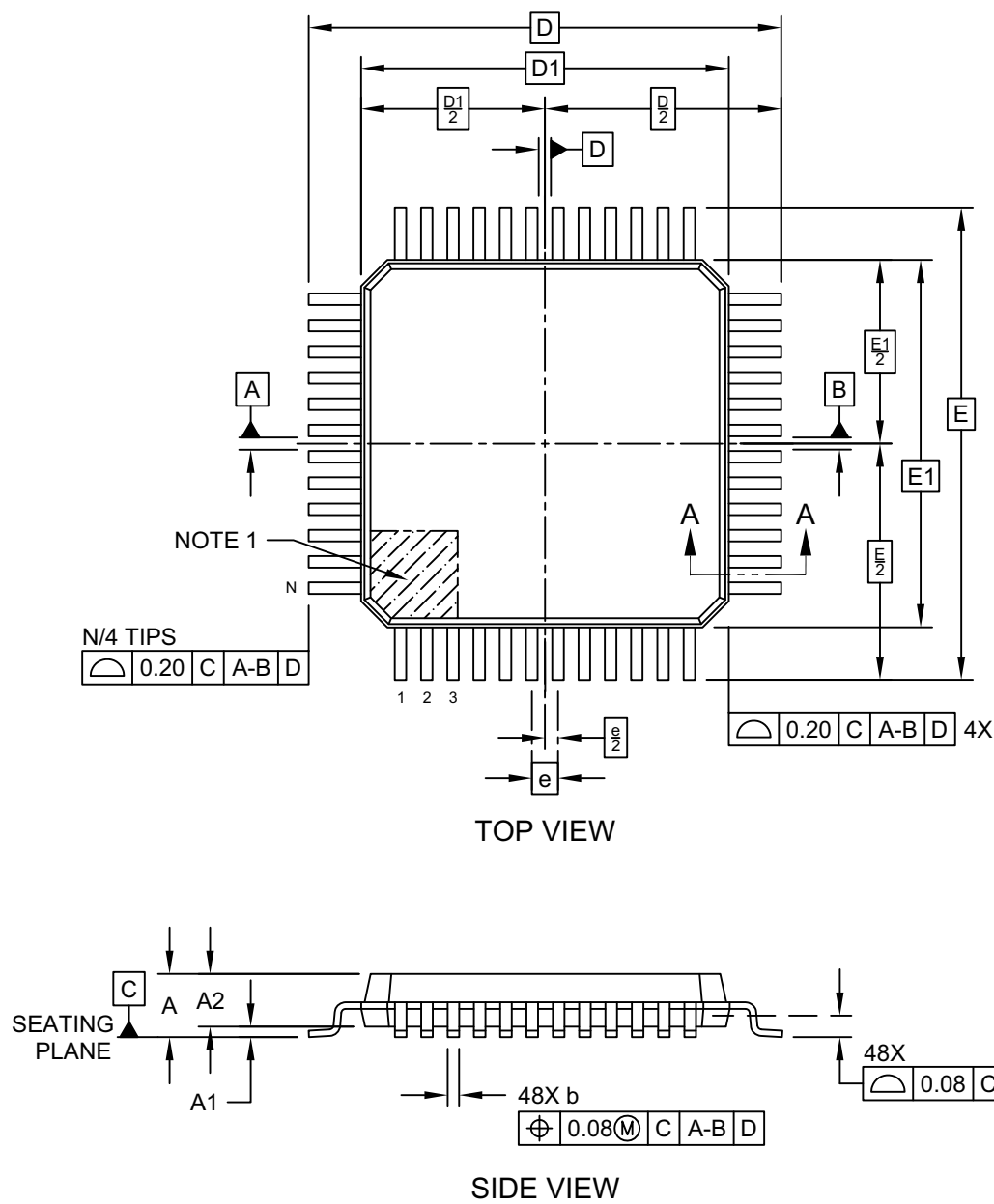
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2436-M5 Rev B

# dsPIC33CK64MP105系列

## 48引脚薄型正方扁平封装（PT）——主体7x7x1.0 mm [TQFP]

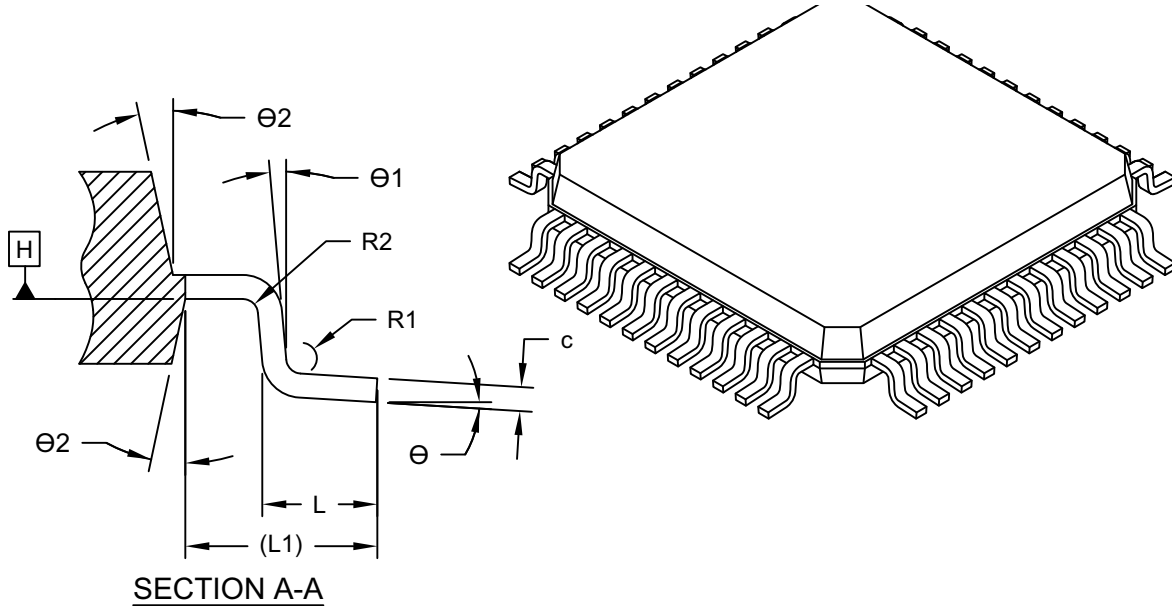
注： 最新封装图请至<http://www.microchip.com/packaging>查看Microchip封装规范。





## 48引脚薄型正方扁平封装（PT）——主体7x7x1.0 mm [TQFP]

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



Dimension	Units	MILLIMETERS		
		MIN	NOM	MAX
Number of Terminals	N	48		
Pitch	e	0.50 BSC		
Overall Height	A	-	-	1.20
Standoff	A1	0.05	-	0.15
Molded Package Thickness	A2	0.95	1.00	1.05
Overall Length	D	9.00 BSC		
Molded Package Length	D1	7.00 BSC		
Overall Width	E	9.00 BSC		
Molded Package Width	E1	7.00 BSC		
Terminal Width	b	0.17	0.22	0.27
Terminal Thickness	c	0.09	-	0.16
Terminal Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Lead Bend Radius	R1	0.08	-	-
Lead Bend Radius	R2	0.08	-	0.20
Foot Angle	Θ	0°	3.5°	7°
Lead Angle	Θ1	0°	-	-
Mold Draft Angle	Θ2	11°	12°	13°

### Notes:

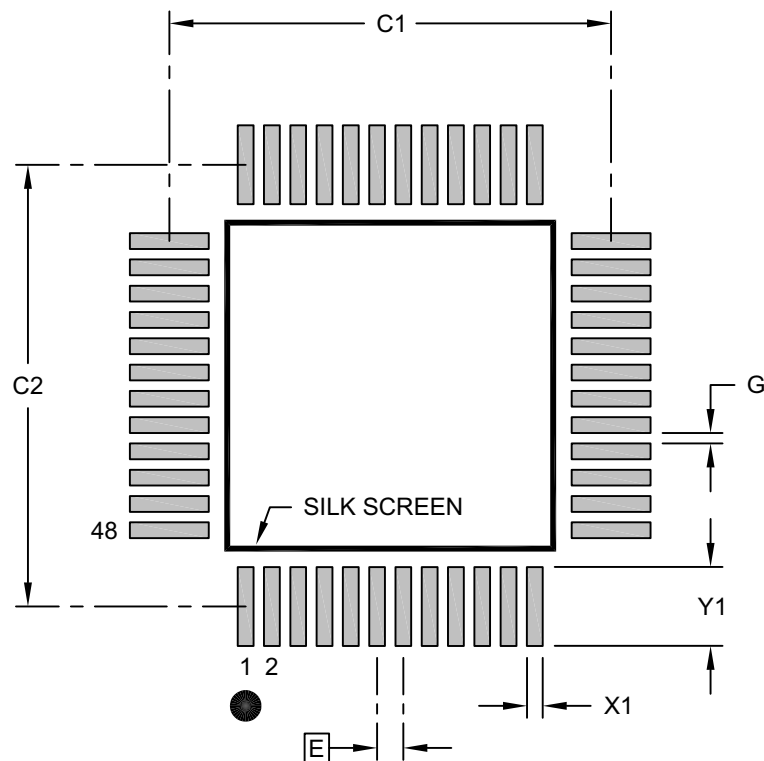
- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.  
REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-300-PT Rev C Sheet 2 of 2

# dsPIC33CK64MP105系列

## 48引脚薄型正方扁平封装（PT）——主体7x7x1.0 mm [TQFP]

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



RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.50 BSC		
Contact Pad Spacing	C1		8.40	
Contact Pad Spacing	C2		8.40	
Contact Pad Width (X48)	X1			0.30
Contact Pad Length (X48)	Y1			1.50
Distance Between Pads	G	0.20		

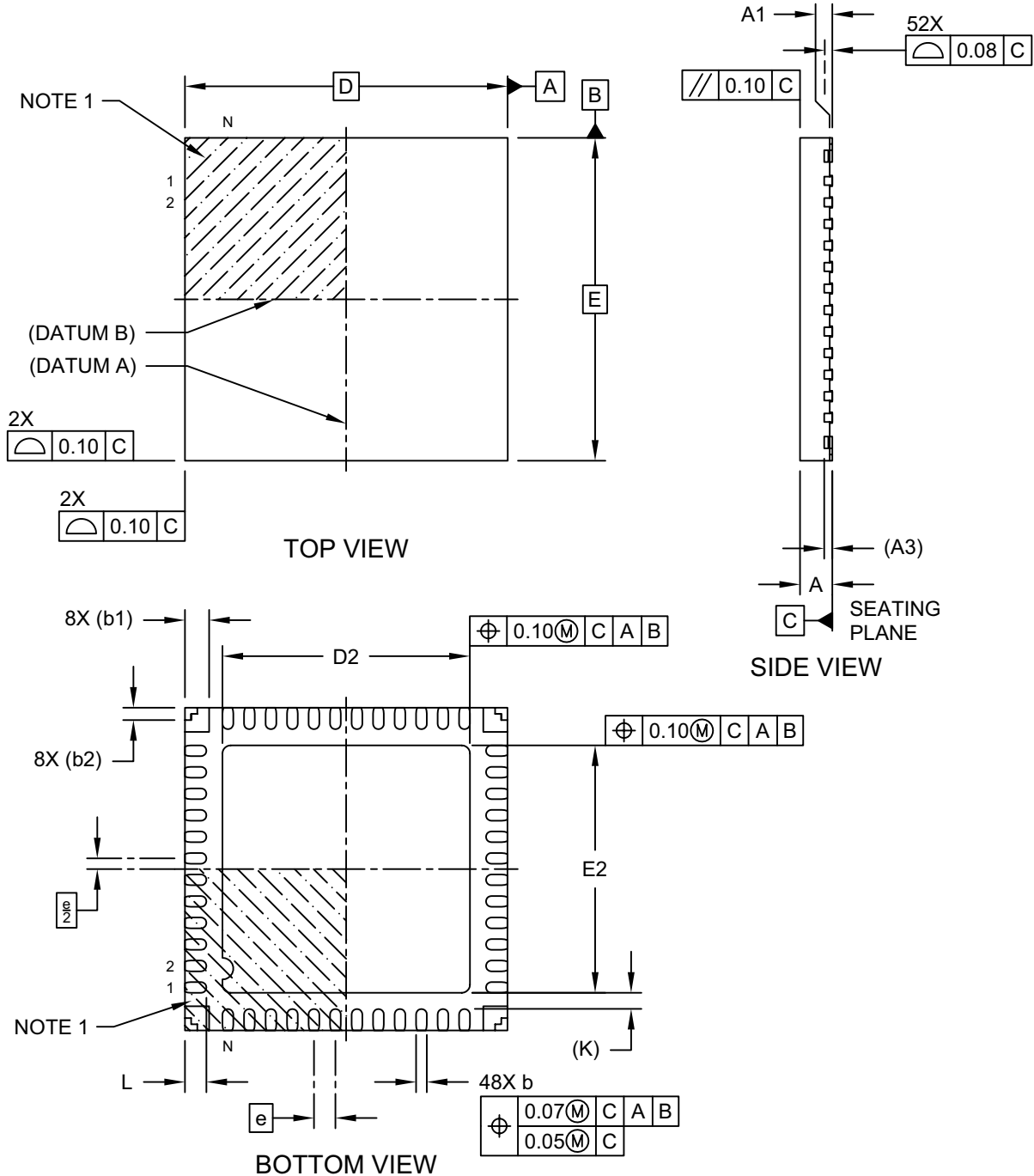
Notes:

- 1. Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- 2. For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2300-PT Rev C

**48引脚塑封超薄正方扁平无引线封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘**

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

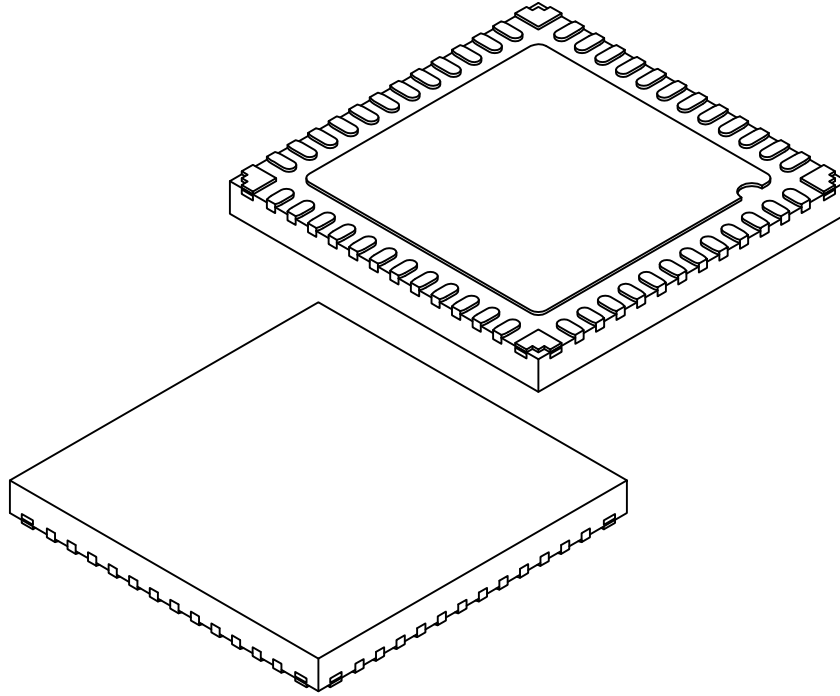


Microchip Technology Drawing C04-442A-M4 Sheet 1 of 2

# dsPIC33CK64MP105系列

**48引脚塑封超薄正方扁平无引线封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘**

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Number of Terminals	N		48		
Pitch	e		0.40 BSC		
Overall Height	A		0.50	0.55	0.60
Standoff	A1		0.00	0.02	0.05
Terminal Thickness	A3		0.15 REF		
Overall Length	D		6.00 BSC		
Exposed Pad Length	D2		4.50	4.60	4.70
Overall Width	E		6.00 BSC		
Exposed Pad Width	E2		4.50	4.60	4.70
Terminal Width	b		0.15	0.20	0.25
Corner Anchor Pad	b1		0.45 REF		
Corner Anchor Pad, Metal-free Zone	b2		0.23 REF		
Terminal Length	L		0.35	0.40	0.45
Terminal-to-Exposed-Pad	K		0.30 REF		

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated
3. Dimensioning and tolerancing per ASME Y14.5M

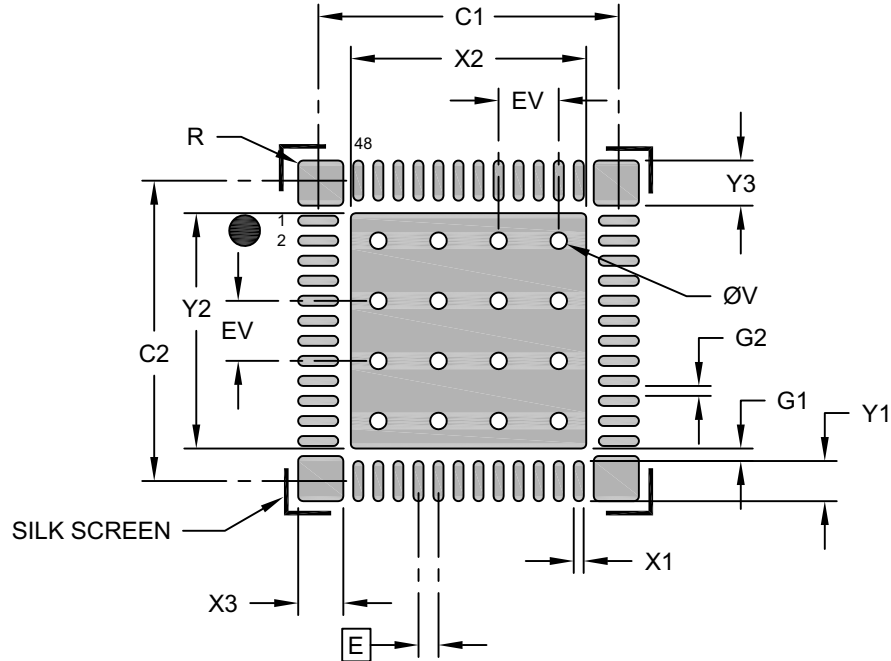
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-442A-M4 Sheet 2 of 2

## 48引脚塑封超薄正方扁平无引线封装（M4）——主体6x6 mm [UQFN]，带角锚点和4.6x4.6 mm外露焊盘

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



### RECOMMENDED LAND PATTERN

Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E	0.40 BSC		
Center Pad Width	X2			4.70
Center Pad Length	Y2			4.70
Contact Pad Spacing	C1		6.00	
Contact Pad Spacing	C2		6.00	
Contact Pad Width (X48)	X1			0.20
Contact Pad Length (X48)	Y1			0.80
Corner Anchor Pad Width (X4)	X3			0.90
Corner Anchor Pad Length (X4)	Y3			0.90
Pad Corner Radius (X 20)	R			0.10
Contact Pad to Center Pad (X48)	G1	0.25		
Contact Pad to Contact Pad	G2	0.20		
Thermal Via Diameter	V		0.33	
Thermal Via Pitch	EV		1.20	

#### Notes:

- Dimensioning and tolerancing per ASME Y14.5M  
BSC: Basic Dimension. Theoretically exact value shown without tolerances.
- For best soldering results, thermal vias, if used, should be filled or tented to avoid solder loss during reflow process

Microchip Technology Drawing C04-2442A-M4

注:

## 附录A： 版本历史

### 版本A（2018年5月）

本文档的初始版本。

### 版本B（2019年1月）

此版本包括以下更新：

- 章节：
  - 更新了“单片机特性”、“认证和B类支持”、[第8.5.6节“输出映射”](#)、[第5.0节“闪存程序存储器”](#)、[第5.2节“RTSP工作原理”](#)和[第31.0节“电气特性”](#)。
  - 删除了[第5.3.1节“闪存程序存储器的编程算法”](#)。
  - 将[第8.5.3节“控制外设引脚选择”](#)替换为[第8.5.3节“控制配置更改”](#)。
  - 增加了[第11.3节“基于PPS的PWM4H输出”](#)和[第12.2节“温度传感器”](#)。
- 表：
  - 更新了表5、表1-1、表4-9、表4-11、表7-3、表8-4、表8-5、表8-6、表8-13、表10-1、表22-1、表28-2、表31-3、表31-15、表31-17、表31-19、表31-20、表31-21、表31-22、表31-23、表31-24、表31-26、表31-27、表31-28、表31-29、表31-30、表31-32、表31-34、表31-35、表31-36和表31-37。
  - 增加了表31-10、表31-11、表31-12、表31-13和表31-14。
- 图：
  - 更新了图1-1、图4-5、图8-2、图8-3和图31-3。

- 寄存器：
  - 更新了寄存器4-1、寄存器5-1、寄存器8-63、寄存器8-64、寄存器8-65（原来为寄存器8-67）、寄存器8-66（原来为寄存器8-68）、寄存器8-67（原来为寄存器8-69）、寄存器9-4、寄存器11-1、寄存器11-2、寄存器11-21、寄存器12-3、寄存器12-6、寄存器12-12、寄存器15-3、寄存器17-1、寄存器17-2、寄存器17-3、寄存器18-2、寄存器25-5、寄存器25-6、寄存器25-7、寄存器25-8、寄存器25-9、寄存器25-10、寄存器25-11、寄存器28-1、寄存器28-2、寄存器28-3、寄存器28-4、寄存器28-5、寄存器28-6、寄存器28-7、寄存器28-8、寄存器28-9、寄存器28-10、寄存器28-11、寄存器28-12、寄存器28-13、寄存器28-14、寄存器28-15、寄存器28-16和寄存器28-17。
  - 删除了寄存器8-65和寄存器8-66。
- 封装信息
  - 在[第33.0节“封装信息”](#)中增加了28引脚UQFN（M6）封装图。

### 版本C（2019年2月）

此版本包括以下更新：

- 表：
  - 更新了表31-4、表31-24、表31-35和表31-38。

### 版本D（2019年7月）

此版本包括以下更新：

- 章节：
  - 在“参考资料”部分中，为《dsPIC33/PIC24系列参考手册》中的“CodeGuard™中等安全性”添加了正确的网络链接。
  - 在“单片机特性”部分中，删除了“可编程高/低电压检测（High/Low-Voltage Detect, HLVD）”。
  - 更新了[第9.8节“参考时钟输出”](#)和[第33.0节“封装信息”](#)。
  - 增加了[第32.0节“高温电气特性”](#)。
- 表：
  - 更新了表4-13、表7-1和表7-5。
- 寄存器：
  - 更新了寄存器9-11、寄存器11-13、寄存器11-14、寄存器11-17、寄存器21-4和寄存器21-5。

注:



## 索引

## 数字

32位可编程循环冗余校验（CRC）发生器 .....	409
5V耐压输入端口 .....	102

## B

## BIST

启动时 .....	39
运行时的BIST .....	39
版本历史 .....	519
备用中断向量表 .....	83

## 比较器/DAC

控制寄存器 .....	255
特性概述 .....	255

## 编程模型

编程模型的	
寄存器说明 .....	25
变更通知客户服务 .....	529

## 捕捉/比较/PWM/定时器

辅助输出 .....	355
控制/状态寄存器 .....	356
时基发生器 .....	350
输出比较模式 .....	353
输入捕捉模式 .....	354
同步源 .....	359
通用定时器 .....	351
自动关断和门控源 .....	360
捕捉/比较/PWM/定时器（SCCP/MCCP） .....	349

## C

## CBG

## CLC

控制寄存器 .....	368
-------------	-----

CodeGuard 安全性 .....	425, 449
---------------------	----------

## CPU

寄存器 .....	23
控制寄存器 .....	28
数据空间寻址 .....	23
寻址模式 .....	23
指令集 .....	23
资源 .....	27

CPU的特殊功能 .....	425
----------------	-----

## CRC

控制寄存器 .....	410
-------------	-----

## 参考资料

程序存储空间 .....	11
--------------	----

## 与数据存储空间的接口

与数据存储空间的接口 .....	60
------------------	----

## 程序存储器

地址空间 .....	33
表读低位字指令（TBLRDL） .....	61
表读高位字指令（TBLRDH） .....	61
构成 .....	60
使用表指令访问程序存储器中的数据 .....	61
复位向量 .....	35
构成 .....	35

## 程序地址空间

dsPIC33CK32MP10X器件的存储器映射 .....	33
--------------------------------	----

程序监控定时器。请参见DMT。 .....	401
-----------------------	-----

程序监控定时器（DMT） .....	401
--------------------	-----

控制/状态寄存器 .....	402
----------------	-----

## 串行外设接口。请参见SPI。

串行外设接口（SPI） .....	307
-------------------	-----

存储器构成 .....	33
-------------	----

资源 .....	41
----------	----

## D

打盹模式 .....	415
------------	-----

## DMA

操作概述 .....	181
典型设置 .....	183
控制寄存器 .....	183
数据传输类型 .....	182
通道触发源 .....	187
外设模块禁止（PMD） .....	183

dsPIC33CK32MP10X器件的代码存储器映射 .....	34
----------------------------------	----

dsPIC33CK64MP10X器件的代码存储器映射 .....	34
----------------------------------	----

DSP引擎 .....	32
-------------	----

带高频PLL的振荡器 .....	155
------------------	-----

代码保护 .....	425, 449
------------	----------

## 代码示例

附属PLL与内部FRC振荡器配合使用 .....	161
PLL（50 MIPS）与POSC配合使用 .....	159
PLL与8 MHz内部FRC配合使用 .....	159
PWRSV指令语法 .....	413

带斜率补偿DAC的高速模拟比较器 .....	253
------------------------	-----

## 单边沿半字节传输。请参见SENT。

单边沿半字节传输（SENT） .....	335
----------------------	-----

发送模式 .....	337
------------	-----

配置 .....	337
----------	-----

接收模式 .....	338
------------	-----

配置 .....	338
----------	-----

控制/状态寄存器 .....	339
----------------	-----

低功耗休眠模式（表） .....	414
------------------	-----

## 电气特性

ADC规范 .....	489
-------------	-----

ADC增量电流 .....	469
---------------	-----

APLL增量电流 .....	469
----------------	-----

BFRC振荡器规范 .....	478
-----------------	-----

比较器 + DAC增量电流 .....	470
---------------------	-----

DAC规范 .....	491
-------------	-----

DAC输出（DACOUT引脚）规范 .....	491
-------------------------	-----

打盹电流（ID0ZE） .....	467
-------------------	-----

掉电电流（IPD） .....	467
-----------------	-----

FRC振荡器规范 .....	478
----------------	-----

封装热阻 .....	464
------------	-----

附属PLL时钟时序规范 .....	477
-------------------	-----

复位和欠压复位要求 .....	479
-----------------	-----

高速模拟比较器规范 .....	490
-----------------	-----

高速PWMx时序要求 .....	480
------------------	-----

工作电流（IDD） .....	465
-----------------	-----

工作电压规范 .....	464
--------------	-----

I/O负载条件规范 .....	474
-----------------	-----

I/O时序要求 .....	475
---------------	-----

I/O引脚输出规范 .....	472
-----------------	-----

I/O引脚输入规范 .....	471
-----------------	-----

I/O引脚输入注入电流规范 .....	472
---------------------	-----

I2Cx总线数据时序要求（从模式） .....	487
-------------------------	-----

I2Cx总线数据要求（主模式） .....	485
-----------------------	-----

交流 .....	474
----------	-----

看门狗定时器增量电流（ΔIWDt） .....	468
-------------------------	-----

空闲电流（Iidle） .....	466
-------------------	-----

LPRC振荡器规范 .....	478
-----------------	-----

PLL时钟时序规范 .....	477
-----------------	-----

PWM增量电流 .....	468
---------------	-----

偏流发生器规范 .....	492
---------------	-----

热工作条件 .....	464
-------------	-----

SPIx从模式时序要求 .....	483
-------------------	-----

SPIx主模式要求 .....	482
-----------------	-----

# dsPIC33CK64MP105系列

闪存程序存储器规范.....	473
UARTx时序要求.....	488
外部时钟时序要求.....	476
运放增量电流.....	470
运算放大器规范.....	492
读者反馈表.....	530

## E

### ECC

控制寄存器.....	72
------------	----

## F

封装.....	501
标识.....	501
详细信息.....	503
复位.....	77
非法条件复位（IOPUWR）.....	77
安全性.....	77
非法操作码.....	77
未初始化的W寄存器.....	77
看门狗定时器超时复位（WDTO）.....	77
配置不匹配复位（CM）.....	77
欠压复位（BOR）.....	77
RESET指令（SWR）.....	77
上电复位（POR）.....	77
陷阱冲突复位（TRAPR）.....	77
主复位（MCLR）引脚复位.....	77
资源.....	78

## G

高速12位模数转换器（ADC）.....	223
控制/状态寄存器.....	227
特性概述.....	223
温度传感器.....	226
资源.....	226
高温电气特性.....	493
高温直流特性.....	
ADC模块精度.....	500
ADC增量电流.....	498
APLL增量电流.....	498
比较器 + DAC增量电流.....	499
DAC模块规范.....	500
打盹电流（ID0ZE）.....	497
掉电电流（IPD）.....	497
工作电流（IDD）.....	495
工作电压规范.....	494
工作MIPS与电压.....	494
I/O引脚输入规范.....	499
看门狗定时器增量电流（Symbol>DIWDT）.....	497
空闲电流（IIDL）.....	496
内部FRC精度.....	499
内部LPRC精度.....	499
PWM增量电流.....	498
热封装特性.....	494
热工作条件.....	494
运放增量电流.....	499
公式.....	
AFPLLO计算.....	161
AFVCO计算.....	161
FPLLO计算.....	158
FVCO计算.....	158
I <sup>2</sup> C波特率重载值计算.....	327
计算频率输出.....	164
节拍周期计算.....	337
器件速度与SPIx时钟速度之间的关系.....	324
SYNCMINx和SYNCMAXx计算.....	338

帧时间计算.....	337
------------	-----

## H

### HSPWM

基于PPS的PWM4H输出.....	190
架构.....	190
控制寄存器.....	191
写限制.....	190

## I

### I/O 端口

并行I/O（PIO）.....	101
控制寄存器.....	104
配置模拟/数字端口引脚.....	104
写/读时序.....	104
有用技巧.....	121
资源.....	122

### I<sup>2</sup>C

保留地址.....	328
从器件地址掩码.....	327
控制/状态寄存器.....	329
设置用作总线主器件时的波特率.....	327
时钟速率.....	327
作为主器件在单主器件环境中通信.....	325

### I2C 请参见I<sup>2</sup>C。

ICSP写禁止.....	75
激活.....	75

## J

### 寄存器

ACLKCON1（附属时钟控制）.....	173
ADCMPCON（ADC数字比较器x控制）.....	248
ADCMPCONENH（ADC数字比较器x通道使能高位字）.....	249
ADCMPCONENL（ADC数字比较器x通道使能低位字）.....	249
ADCON1H（ADC控制1高位字）.....	228
ADCON1L（ADC控制1低位字）.....	227
ADCON2H（ADC控制2高位字）.....	230
ADCON2L（ADC控制2低位字）.....	229
ADCON3H（ADC控制3高位字）.....	232
ADCON3L（ADC控制3低位字）.....	231
ADCON4H（ADC控制4高位字）.....	234
ADCON4L（ADC控制4低位字）.....	233
ADCON5H（ADC控制5高位字）.....	236
ADCON5L（ADC控制5低位字）.....	235
ADCORExH（专用ADC内核x控制高位字）.....	238
ADCORExL（专用ADC内核x控制低位字）.....	237
ADEIEH（ADC提前中断允许高位字）.....	240
ADEIEL（ADC提前中断允许低位字）.....	240
ADEISTATH（ADC提前中断状态高位字）.....	241
ADEISTATL（ADC提前中断状态低位字）.....	241
ADFLxCON（ADC数字滤波器x控制）.....	250
ADIEH（ADC中断允许高位字）.....	244
ADIEL（ADC中断允许低位字）.....	244
ADLVLTRGH（ADC电平敏感触发控制高位字）.....	239
ADLVLTRGL（ADC电平敏感触发控制低位字）.....	239
ADMOD0H（ADC输入模式控制0高位字）.....	242
ADMOD0L（ADC输入模式控制0低位字）.....	242
ADMOD1L（ADC输入模式控制1低位字）.....	243
ADSTATH（ADC数据就绪状态高位字）.....	245
ADSTATL（ADC数据就绪状态低位字）.....	245
ADTRIGnL/ADTRIGnH（ADC通道触发n(x)选择低位字/高位字）.....	246
AMPCON1H（运放控制高位字）.....	399
AMPCON1L（运放控制低位字）.....	398

ANSELx (PORTx模拟选择) .....	104	FDMTCNTL配置 .....	435
APLLDIV1 (APLL输出分频比) .....	175	FDMTIVTH配置 .....	434
APLLFBD1 (APLL反馈分频比) .....	174	FDMTIVTL配置 .....	434
BIASCON (偏流发生器控制) .....	394	FDMT配置 .....	436
CCPxCON1H (CCPx控制1高位字) .....	358	FICD配置 .....	433
CCPxCON1L (CCPx控制1低位字) .....	356	FOSC配置 .....	430
CCPxCON2H (CCPx控制2高位字) .....	361	FOSCSEL配置 .....	429
CCPxCON2L (CCPx控制2低位字) .....	360	FPOR配置 .....	432
CCPxCON3H (CCPx控制3高位字) .....	363	FSLC (频率调节) .....	192
CCPxCON3L (CCPx控制3低位字) .....	362	FSEC配置 .....	427
CCPxSTATL (CCPx状态) .....	364	FSIGN配置 .....	428
CLCxCONH (CLCx控制高位字) .....	369	FSMINPER (频率调节最小周期) .....	192
CLCxCONL (CLCx控制低位字) .....	368	FWDT配置 .....	431
CLCxGLSH (CLCx门逻辑输入选择高位字) .....	374	I2CxCONH (I2Cx控制高位字) .....	331
CLCxGLSL (CLCx门逻辑输入选择低位字) .....	372	I2CxCONL (I2Cx控制低位字) .....	329
CLCxSEL (CLCx输入多路开关选择) .....	370	I2CxMSK (I2Cx从模式地址掩码) .....	333
CLKDIV (时钟分频比) .....	168	I2CxSTAT (I2Cx状态) .....	332
CMBTRIGH (组合触发高位字) .....	196	IBIASCONH (偏流发生器电流源控制高位字) .....	395
CMBTRIGL (组合触发低位字) .....	195	IBIASCONL (偏流发生器电流源控制低位字) .....	396
CNCONx (PORTx电平变化通知控制) .....	108	INDXxCNTH (索引x计数器高位字) .....	280
CNEN0x (PORTx电平变化通知中断允许) .....	108	INDXxCNTL (索引x计数器低位字) .....	280
CNEN1x (PORTx电平变化通知中断边沿选择) .....	109	INDXxHLD (索引x计数器保持) .....	281
CNFX (PORTx电平变化通知中断标志) .....	110	INTCON1 (中断控制1) .....	94
CNPDX (PORTx电平变化通知下拉使能) .....	107	INTCON2 (中断控制2) .....	96
CNPUx (PORTx电平变化通知上拉使能) .....	107	INTCON3 (中断控制3) .....	97
CNSTATx (PORTx电平变化通知中断状态) .....	109	INTCON4 (中断控制4) .....	98
CORCON (内核控制) .....	30, 93	INTTREG (中断控制和状态) .....	99
CRCCONH (CRC控制高位字) .....	411	INTxTMRH (间隔x定时器高位字) .....	278
CRCCONL (CRC控制低位字) .....	410	INTxTMRL (间隔x定时器低位字) .....	278
CRCXORH (CRC XOR多项式, 高位字) .....	412	INTXxHLDH (间隔x定时器保持高位字) .....	279
CRCXORL (CRC XOR多项式, 低位字) .....	412	INTXxHLDL (间隔x定时器保持低位字) .....	279
CTXTSTAT (CPU W寄存器现场状态) .....	31	LATx (PORTx输出数据) .....	106
DACCTRL1L (DAC控制1低位字) .....	256	LFSR (线性反馈移位) .....	201
DACCTRL2H (DAC控制2高位字) .....	257	LOGCONy (组合PWM逻辑控制y) .....	197
DACCTRL2L (DAC控制2低位字) .....	257	MBISTCON (MBIST控制) .....	40
DACxCONH (DACx控制高位字) .....	258	MDC (主占空比) .....	193
DACxCONL (DACx控制低位字) .....	258	MPER (主周期) .....	194
DACxDATH (DACx数据高位字) .....	260	MPHASE (主相位) .....	193
DACxDATL (DACx数据低位字) .....	260	NVMADR (非易失性存储器低位字地址) .....	68
DEVID (器件ID) .....	441	NVMADRU (非易失性存储器高位字地址) .....	68
DEVREV (器件版本) .....	440	NVMCON (非易失性存储器(NVM)控制) .....	66
DMACHn (DMA通道n控制) .....	185	NVMKEY (非易失性存储器密钥) .....	69
DMACON (DMA引擎控制) .....	184	NVMSRCADRH (NVM源数据地址高位字) .....	70
DMAINTn (DMA通道n中断) .....	186	NVMSRCADRL (NVM源数据地址低位字) .....	70
DMTCLR (程序监控定时器清零) .....	403	ODCx (PORTx漏极开路使能) .....	106
DMTCNTH (程序监控定时器计数高位字) .....	405	OSCCON (振荡器控制) .....	166
DMTCNTL (程序监控定时器计数低位字) .....	405	OSCTUN (FRC振荡器调节) .....	171
DMTCON (程序监控定时器控制) .....	402	PCLKCON (PWM时钟控制) .....	191
DMTHOLDREG (DMT保持) .....	408	PGxCAP (PWM发生器x捕捉) .....	221
DMTPRECLR (程序监控定时器预清零) .....	402	PGxCONH (PWM发生器x控制高位字) .....	203
DMTPSCNTH (DMT配置后计数状态高位字) .....	406	PGxCONL (PWM发生器x控制低位字) .....	202
DMTPSCNTL (DMT配置后计数状态低位字) .....	406	PGxDC (PWM发生器x占空比) .....	217
DMTPSINTVH (DMT配置后间隔状态高位字) .....	407	PGxDCA (PWM发生器x占空比调整) .....	218
DMTPSINTVL (DMT配置后间隔状态低位字) .....	407	PGxDTH (PWM发生器x死区高位字) .....	220
DMTSTAT (程序监控定时器状态) .....	404	PGxDTL (PWM发生器x死区低位字) .....	220
ECCADDRH (ECC错误注入地址比较高位字) .....	73	PGxEVTH (PWM发生器x事件高位字) .....	210
ECCADDRL (ECC错误注入地址比较低位字) .....	73	PGxEVTL (PWM发生器x事件低位字) .....	209
ECCCONH (ECC错误注入配置高位字) .....	72	PGxIOCONH (PWM发生器x I/O控制高位字) .....	208
ECCCONL (ECC错误注入配置低位字) .....	72	PGxIOCONL (PWM发生器x I/O控制低位字) .....	207
ECCSTATH (ECC系统状态显示高位字) .....	74	PGxLEBH (PWM发生器x前沿消隐高位字) .....	216
ECCSTATL (ECC系统状态显示低位字) .....	74	PGxLEBL (PWM发生器x前沿消隐低位字) .....	215
FALTREG配置 .....	438	PGxPER (PWM发生器x周期) .....	218
FBSLIM配置 .....	428	PGxPHASE (PWM发生器x相位) .....	217
FDEVOPT配置 .....	437	PGxSTAT (PWM发生器x状态) .....	205
FDMTCNTH配置 .....	435	PGxTRIGA (PWM发生器x触发A) .....	219

# dsPIC33CK64MP105系列

PGxTRIGB (PWM发生器x触发B) .....	219	RPINR3 (外设引脚选择输入3) .....	127
PGxTRIGC (PWM发生器x触发C) .....	219	RPINR30 (外设引脚选择输入30) .....	137
PGxyPCIH (PWM发生器xy PCI高位字) .....	213	RPINR37 (外设引脚选择输入37) .....	137
PGxyPCIL (PWM发生器xy PCI低位字) .....	211	RPINR38 (外设引脚选择输入38) .....	138
PLLDIV (PLL输出分频比) .....	172	RPINR4 (外设引脚选择输入4) .....	127
PLLFBD (PLL反馈分频比) .....	170	RPINR42 (外设引脚选择输入42) .....	138
PMD1 (PMD1控制) .....	416	RPINR43 (外设引脚选择输入43) .....	139
PMD2 (PMD2控制) .....	417	RPINR44 (外设引脚选择输入44) .....	139
PMD3 (PMD3控制) .....	418	RPINR45 (外设引脚选择输入45) .....	140
PMD4 (PMD4控制) .....	419	RPINR46 (外设引脚选择输入46) .....	140
PMD6 (PMD6控制) .....	420	RPINR47 (外设引脚选择输入47) .....	141
PMD7 (PMD7控制) .....	421	RPINR48 (外设引脚选择输入48) .....	141
PMD8 (PMD8控制) .....	422	RPINR49 (外设引脚选择输入49) .....	142
PORTx (PORTx输入数据) .....	105	RPINR5 (外设引脚选择输入5) .....	128
POSxCNTH (位置x计数器高位字) .....	274	RPINR6 (外设引脚选择输入6) .....	128
POSxCNTL (位置x计数器低位字) .....	274	RPINR7 (外设引脚选择输入7) .....	129
POSxHLD (位置x计数器保持) .....	275	RPOR0 (外设引脚选择输出0) .....	143
PTGADJ (PTG调节) .....	386	RPOR1 (外设引脚选择输出1) .....	143
PTGBTE (PTG广播触发使能低位字) .....	382	RPOR10 (外设引脚选择输出10) .....	148
PTGBTEH (PTG广播触发使能高位字) .....	382	RPOR11 (外设引脚选择输出11) .....	148
PTGC0LIM (PTG计数器0限制) .....	385	RPOR12 (外设引脚选择输出12) .....	149
PTGC1LIM (PTG计数器1限制) .....	385	RPOR13 (外设引脚选择输出13) .....	149
PTGCON (PTG控制/状态高位字) .....	381	RPOR14 (外设引脚选择输出14) .....	150
PTGCST (PTG控制/状态低位字) .....	379	RPOR15 (外设引脚选择输出15) .....	150
PTGHOLD (PTG保持) .....	383	RPOR16 (外设引脚选择输出16) .....	151
PTGL0 (PTG立即数0) .....	386	RPOR17 (外设引脚选择输出17) .....	151
PTGQPTR (PTG步阶队列指针) .....	387	RPOR18 (外设引脚选择输出18) .....	152
PTGQUEn (PTG步阶队列n指针) .....	387	RPOR19 (外设引脚选择输出19) .....	152
PTGSDLIM (PTG步阶延时限制) .....	384	RPOR2 (外设引脚选择输出2) .....	144
PTGT0LIM (PTG Timer0限制) .....	383	RPOR3 (外设引脚选择输出3) .....	144
PTGT1LIM (PTG Timer1限制) .....	384	RPOR4 (外设引脚选择输出4) .....	145
PWMEVty (PWM事件输出控制y) .....	199	RPOR5 (外设引脚选择输出5) .....	145
QEIXCON (QEIX控制) .....	268	RPOR6 (外设引脚选择输出6) .....	146
QEIXGECH (QEIX大于或等于比较高位字) .....	284	RPOR7 (外设引脚选择输出7) .....	146
QEIXGECL (QEIX大于或等于比较低位字) .....	284	RPOR8 (外设引脚选择输出8) .....	147
QEIXICH (QEIX初始化/捕捉高位字) .....	282	RPOR9 (外设引脚选择输出9) .....	147
QEIXICL (QEIX初始化/捕捉低位字) .....	282	SENTxCON1 (SENTx控制1) .....	339
QEIXIOC (QEIX I/O控制) .....	270	SENTxDATH (SENTx接收数据高位字) .....	343
QEIXIOCH (QEIX I/O控制高位字) .....	271	SENTxDATL (SENTx接收数据低位字) .....	343
QEIXLECH (QEIX小于或等于比较高位字) .....	283	SENTxSTAT (SENTx状态) .....	341
QEIXLECL (QEIX小于或等于比较低位字) .....	283	SLPxCONH (DACx斜率控制高位字) .....	261
QEIXSTAT (QEIX状态) .....	272	SLPxCONL (DACx斜率控制低位字) .....	262
RCON (复位控制) .....	79	SLPxDAT (DACx斜率数据) .....	263
REFOCONH (参考时钟控制高位字) .....	177	SPIxCON1H (SPIx控制1高位字) .....	314
REFOCONL (参考时钟控制低位字) .....	176	SPIxCON1L (SPIx控制1低位字) .....	312
REFOTRIM (参考振荡器微调) .....	177	SPIxCON2L (SPIx控制2低位字) .....	316
RPCON (外设重映射配置) .....	125	SPIxIMSKH (SPIx中断屏蔽高位字) .....	321
RPINR0 (外设引脚选择输入0) .....	125	SPIxIMSKL (SPIx中断屏蔽低位字) .....	320
RPINR1 (外设引脚选择输入1) .....	126	SPIxSTATH (SPIx状态高位字) .....	319
RPINR11 (外设引脚选择输入11) .....	129	SPIxSTATL (SPIx状态低位字) .....	317
RPINR12 (外设引脚选择输入12) .....	130	SR (CPU状态) .....	28, 92
RPINR13 (外设引脚选择输入13) .....	130	T1CON (Timer1控制) .....	346
RPINR14 (外设引脚选择输入14) .....	131	TRISx (PORTx输出使能) .....	105
RPINR15 (外设引脚选择输入15) .....	131	WDTCONH (看门狗定时器控制高位字) .....	447
RPINR16 (外设引脚选择输入16) .....	132	WDTCONL (看门狗定时器控制低位字) .....	446
RPINR17 (外设引脚选择输入17) .....	132	VELxCNT (速度x计数器) .....	276
RPINR18 (外设引脚选择输入18) .....	133	VELxCNTH (速度x计数器高位字) .....	276
RPINR19 (外设引脚选择输入19) .....	133	VELxHLD (速度x计数器保持) .....	277
RPINR2 (外设引脚选择输入2) .....	126	VREGCON (稳压器控制) .....	443
RPINR20 (外设引脚选择输入20) .....	134	UxBRG (UARTx波特率) .....	296
RPINR21 (外设引脚选择输入21) .....	134	UxBRGH (UARTx波特率高位字) .....	296
RPINR22 (外设引脚选择输入22) .....	135	UxINT (UARTx中断) .....	305
RPINR23 (外设引脚选择输入23) .....	135	UxMODE (UARTx配置) .....	288
RPINR27 (外设引脚选择输入27) .....	136	UxMODEH (UARTx配置高位字) .....	290
RPINR29 (外设引脚选择输入29) .....	136	UxP1 (UARTx时序参数1) .....	298

UxP2 (UARTx时序参数2) .....	299	定时器时钟发生器 .....	350
UxP3 (UARTx时序参数3) .....	300	访问程序空间内数据的地址生成方式 .....	60
UxP3H (UARTx时序参数3高位字) .....	300	概念 SCCPx 模块 .....	349
UxRXCHK (UARTx接收校验和) .....	302	高速模拟比较器 .....	254
UxRXREG (UARTx接收缓冲区) .....	297	共用端口结构 .....	103
UxSCCON (UARTx智能卡配置) .....	303	恒流源 .....	393
UxSCINT (UARTx智能卡中断) .....	304	I2Cx 模块 .....	326
UxSTA (UARTx状态) .....	292	简化 UARTx .....	286
UxSTAH (UARTx状态高位字) .....	294	建议的振荡器电路布局 .....	19
UxTXCHK (UARTx发送校验和) .....	301	建议的最基本连接 .....	18
UxTXREG (UARTx发送缓冲区) .....	297	交错式 PFC .....	20
寄存器映射		看门狗定时器 (WDT) .....	445
PMD .....	423	离线 UPS .....	22
PORTA .....	123	MCLR 引脚连接 .....	18
PORTB .....	123	内部稳压器 .....	442
PORTC .....	124	PLL 和 VCO .....	157
PORTD .....	124	PSV 的读地址生成方式 .....	51
PPS 输出控制 .....	154	PTG 模块 .....	378
PPS 输入控制 .....	153	PWM 高级模块 .....	190
配置 .....	426	QEI 模块 .....	267
中断标志 .....	89	RPN 的可重映射输出的复用 .....	117
中断优先级 .....	90	SENTx 模块 .....	336
中断允许 .....	89	SPIx 从器件/帧从器件连接 .....	324
JTAG 边界扫描接口 .....	425	SPIx 从器件/帧主器件连接 .....	324
JTAG 接口 .....	448	SPIx 模块 (标准模式) .....	309
基于指令的节能模式 .....	413	SPIx 模块 (增强型模式) .....	310
空闲模式 .....	414	SPIx 主/从连接 (标准模式) .....	322
休眠模式 .....	414	SPIx 主/从连接 (增强型缓冲区模式) .....	323
交流特性 .....	474	SPIx 主器件/帧从器件连接 .....	324
模数转换要求 .....	489	SPIx 主器件/帧主器件连接 .....	323
节能特性 .....	413	输出比较 x 模块 .....	353
时钟频率和切换 .....	413	输入捕捉 x 模块 .....	354
资源 .....	415	双 16 位定时器模式 .....	352
纠错码 (ECC) .....	71	U1RX 的可重映射输入 .....	112
错误注入 .....	71	相移全桥转换器 .....	21
绝对最大值 .....	463, 493	直接存储器访问 (DMA) .....	180
<b>K</b>		主内核复位系统 .....	77
开发支持 .....	461	专用 ADC 内核 .....	225
看门狗定时器 (WDT) .....	425	<b>L</b>	
勘误表 .....	10	灵活的配置 .....	425
客户通知服务 .....	529	<b>M</b>	
客户支持 .....	529	Microchip 因特网网站 .....	529
可配置逻辑单元。请参见 CLC。		模数转换器。请参见 ADC。	
可配置逻辑单元 (CLC) .....	365	模寻址 .....	57
框图		操作示例 .....	57
16 位 Timer1 模块 .....	345	的应用 .....	58
32 位定时器模式 .....	352	起始地址和结束地址 .....	57
ADC 共用内核 .....	225	W 地址寄存器选择 .....	57
ADC 模块 .....	224	<b>N</b>	
APLL 和 VCO 的详细信息 .....	160	内置自检。请参见 BIST。	
安全段 .....	449	<b>P</b>	
编程模型 .....	26	PTG	
表寄存器的寻址 .....	63	步阶命令格式/说明 .....	388
CALL 堆栈帧 .....	54	控制/状态寄存器 .....	379
CLCx 逻辑功能组合选项 .....	366	命令选项 .....	389
CLCx 模块 .....	365	输出说明 .....	391
CLCx 输入源选择 .....	367	输入说明 .....	391
CPU 内核 .....	24	特性 .....	377
CRC 模块 .....	409	配置位 .....	425
参考时钟发生器 .....	164	用于时钟选择的位值 .....	165
程序监控定时器模块 .....	401	偏流发生器	
dsPIC33CK64MP105 内核时钟源 .....	155	控制寄存器 .....	394
dsPIC33CK64MP105 内核振荡器子系统 .....	156		
dsPIC33CK64MP105 系列 .....	13		
单运算放大器 .....	397		

# dsPIC33CK64MP105系列

偏流发生器。请参见CBG。

偏流发生器（CBG）..... 393

## Q

### QE1

控制/状态寄存器..... 268

真值表..... 266

器件标识..... 440

器件概述..... 13

器件唯一标识符。请参见UDID。

器件唯一标识符（UDID）..... 36

欠压复位（BOR）..... 425, 444

## R

入门指南..... 17

ICSP引脚..... 19

连接要求..... 17

目标应用..... 20

启动时的振荡器值条件..... 20

去耦电容..... 17

外部振荡器引脚..... 19

未用I/O..... 20

主复位（MCLR）引脚..... 18

软件模拟器（MPLAB SIM）..... 461

## S

### SENT

SENTx协议数据帧..... 336

### SFR块

000h..... 41

100h..... 42

200h..... 43

300h-400h..... 44

800h..... 45

900h..... 46

A00h..... 47

B00h..... 47

C00h..... 48

D00h..... 49

E00h..... 49

F00h..... 50

Single-Edge Nibble Transmission for Automotive

Applications..... 335

### SPI

控制/状态寄存器..... 312

闪存程序存储器

和表指令..... 63

控制寄存器..... 65

RTSP操作..... 64

时序图

BOR和主复位特性..... 479

高速PWMx特性..... 480

I/O特性..... 475

I2Cx总线启动位/停止位（从模式）..... 486

I2Cx总线启动位/停止位（主模式）..... 484

I2Cx总线数据（从模式）..... 486

I2Cx总线数据（主模式）..... 484

QE1接口信号..... 265

SPIx从模式（CKE = 0）特性..... 482

SPIx从模式（CKE = 1）特性..... 483

SPIx主模式（CKE = 0）特性..... 481

SPIx主模式（CKE = 1）特性..... 481

时钟/指令周期..... 162

UARTx特性..... 488

外部时钟..... 476

数据存储测试（BIST）..... 39

流程图..... 39

数据地址空间..... 36

dsPIC33CK64MPX0X和dsPIC33CK32MPX0X器件

的寄存器映射..... 38

构成，对齐方式..... 36

宽度..... 36

Near数据空间..... 37

SFR空间..... 37

X和Y数据空间..... 39

数据空间

分页存储器方案..... 51

分页数据存储空间（图）..... 52

扩展X..... 54

输入电平变化通知（ICN）..... 111

双看门狗定时器（双WDT）..... 445

算术逻辑单元（ALU）..... 32

## T

Timer1..... 345

控制寄存器..... 346

通用异步收发器（UART）..... 285

通用异步收发器请参见UART。

## U

### UART

架构概述..... 286

控制/状态寄存器..... 288

数据缓冲区..... 287

协议扩展..... 287

字符帧..... 287

## W

WWW地址..... 529

WWW在线技术支持..... 10

外设触发信号发生器。请参见PTG。

外设触发信号发生器（PTG）..... 377

外设模块禁止（PMD）..... 415

控制寄存器..... 416

外设引脚选择（PPS）..... 111

可选输入源..... 115

可用的外设..... 111

可用的引脚..... 111

可重映射输出引脚寄存器..... 118

可重映射引脚的输出选择..... 119

可重映射引脚输入..... 113

控制寄存器..... 125

配置更改控制..... 112

输出映射..... 117

输入映射..... 112

位反转寻址..... 58

的实现..... 58

示例..... 59

序列表（16项）..... 59

稳压器（片上）..... 442

## X

循环冗余校验。请参见CRC。

## Y

引脚和ANSELx的可用性..... 102

引脚I/O说明（表）..... 14

因特网地址..... 529

用户OTP存储区..... 442

运算放大器..... 397

控制寄存器..... 398

## Z

在节能指令执行期间的中断 .....	414
在线串行编程 (ICSP) .....	425, 448
在线调试器 .....	448
在线仿真 .....	425
振荡器	
CPU时钟 .....	162
参考时钟输出 .....	164
低功耗RC (LPRC) .....	163
附属PLL .....	160
控制寄存器 .....	166
内部备用快速RC (BFRC) 振荡器 .....	163
内部快速RC (FRC) .....	163
OSCCON解锁序列 .....	165
配置 .....	165
主PLL .....	157
主振荡器 (POSC) .....	163
正交编码器接口。请参见QE1。	
正交编码器接口 (QE1) .....	265
支持精细边沿定位的高分辨率PWM (HSPWM) .....	189
直接存储器访问控制器。请参见DMA。	
指令集汇总 .....	451
操作码说明中使用的符号 .....	452
概述 .....	454
指令寻址模式 .....	55
MAC指令 .....	56
MCU指令 .....	55
其他指令 .....	56
文件寄存器指令 .....	55
支持的基本模式 .....	55
传送指令和累加器指令 .....	56
直流特性	
工作条件 .....	464
中断控制器	
备用中断向量表 (AIVT) .....	81
复位序列 .....	81
控制和状态寄存器 .....	91
INTCON1 .....	91
INTCON2 .....	91
INTCON3 .....	91
INTCON4 .....	91
INTTREG .....	91
陷阱向量详细信息 .....	84
中断向量表 (IVT) .....	81
中断向量详细信息 .....	85
资源 .....	91
中断向量表 .....	82

注:



## MICROCHIP网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。我们的网站提供以下内容:

- **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归档软件
- **一般技术支持**——常见问题解答 (FAQ)、技术支持请求、在线讨论组以及 Microchip 设计伙伴计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时, 收到电子邮件通知。

欲注册, 请访问 [www.microchip.com/pcn](http://www.microchip.com/pcn), 然后按照注册说明进行操作。

## 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 (ESE)
- 技术支持

客户应联系其代理商、代表或 ESE 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过<http://microchip.com/support>获得网上技术支持。

注:

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。

dsPIC 33 CK 64 MP 105 T-I / PT XXX

Microchip的商标

架构

闪存系列

程序存储器容量 (KB)

产品组

引脚数

卷带标志 (如果适用)

温度范围

封装

定制编号

架构:

产品组:

引脚数:

温度范围:

封装:

33 = 16位数字信号控制器

MP = 电机控制/电源

02 = 28引脚  
03 = 36引脚  
05 = 48引脚

I = -40°C至+85°C (工业级)  
E = -40°C至+125°C (扩展级)  
H = -40°C至+150°C (高温)

SS = 塑封紧缩小外形封装——(28引脚)主体5.30 mm (SSOP)  
2N = 塑封超薄正方扁平无引线封装——(28引脚)主体6x6 mm (UQFN)  
M5 = 塑封超薄正方扁平无引线封装——(36引脚)主体5x5 mm (UQFN)  
PT = 薄型正方扁平封装——(48引脚)主体7x7 mm (TQFP)  
M4 = 塑封超薄正方扁平无引线封装——(48引脚)主体6x6 mm (UQFN)

示例:

dsPIC33CK64MP105-I/PT:  
dsPIC33, 64 KB程序存储器,  
电机控制/电源系列, 48 引脚,  
工业级温度, TQFP封装。

注:

---

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。

---

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。

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

有关 **Microchip** 质量管理体系的更多信息，请访问  
[www.microchip.com/quality](http://www.microchip.com/quality)。

## 商标

**Microchip** 的名称和徽标组合、**Microchip** 徽标、**Adaptec**、**AnyRate**、**AVR**、**AVR** 徽标、**AVR Freaks**、**BesTime**、**BitCloud**、**chipKIT**、**chipKIT** 徽标、**CryptoMemory**、**CryptoRF**、**dsPIC**、**FlashFlex**、**flexPWR**、**HELDO**、**IGLOO**、**JukeBlox**、**KeeLoq**、**Kleer**、**LANCheck**、**LinkMD**、**maxStylus**、**maxTouch**、**MediaLB**、**megaAVR**、**Microsemi**、**Microsemi** 徽标、**MOST**、**MOST** 徽标、**MPLAB**、**OptoLyzer**、**PackTime**、**PIC**、**picoPower**、**PICSTART**、**PIC32** 徽标、**PolarFire**、**Prochip Designer**、**QTouch**、**SAM-BA**、**SenGenuity**、**SpyNIC**、**SST**、**SST** 徽标、**SuperFlash**、**Symmetricon**、**SyncServer**、**Tachyon**、**TempTrackr**、**TimeSource**、**tinyAVR**、**UNI/O**、**Vectron** 及 **XMEGA** 均为 **Microchip Technology Incorporated** 在美国和其他国家或地区的注册商标。

**APT**、**ClockWorks**、**The Embedded Control Solutions Company**、**EtherSynch**、**FlashTec**、**Hyper Speed Control**、**HyperLight Load**、**IntelliMOS**、**Libero**、**motorBench**、**mTouch**、**Powermite** 3、**Precision Edge**、**ProASIC**、**ProASIC Plus**、**ProASIC Plus** 徽标、**Quiet-Wire**、**SmartFusion**、**SyncWorld**、**Temux**、**TimeCesium**、**TimeHub**、**TimePictra**、**TimeProvider**、**Vite**、**WinPath** 和 **ZL** 均为 **Microchip Technology Incorporated** 在美国的注册商标。

**Adjacent Key Suppression**、**AKS**、**Analog-for-the-Digital Age**、**Any Capacitor**、**AnyIn**、**AnyOut**、**BlueSky**、**BodyCom**、**CodeGuard**、**CryptoAuthentication**、**CryptoAutomotive**、**CryptoCompanion**、**CryptoController**、**dsPICDEM**、**dsPICDEM.net**、**Dynamic Average Matching**、**DAM**、**ECAN**、**EtherGREEN**、**In-Circuit Serial Programming**、**ICSP**、**INICNet**、**Inter-Chip Connectivity**、**JitterBlocker**、**KleerNet**、**KleerNet** 徽标、**memBrain**、**Mindi**、**MiWi**、**MPASM**、**MPF**、**MPLAB Certified** 徽标、**MPLIB**、**MPLINK**、**MultiTRAK**、**NetDetach**、**Omniscient Code Generation**、**PICDEM**、**PICDEM.net**、**PICkit**、**PICtail**、**PowerSmart**、**PureSilicon**、**QMatrix**、**REAL ICE**、**Ripple Blocker**、**SAM-ICE**、**Serial Quad I/O**、**SMART-I.S.**、**SQL**、**SuperSwitcher**、**SuperSwitcher II**、**Total Endurance**、**TSHARC**、**USBCheck**、**VariSense**、**ViewSpan**、**WiperLock**、**Wireless DNA** 和 **ZENA** 均为 **Microchip Technology Incorporated** 在美国和其他国家或地区的商标。

**SQTP** 为 **Microchip Technology Incorporated** 在美国的服务标记。

**Adaptec** 徽标、**Frequency on Demand**、**Silicon Storage Technology** 和 **Symmcom** 均为 **Microchip Technology Inc.** 在除美国外的国家或地区的注册商标。

**GestIC** 为 **Microchip Technology Inc.** 的子公司 **Microchip Technology Germany II GmbH & Co. KG** 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2019-2020, **Microchip Technology Incorporated** 版权所有。

ISBN: 978-1-5224-5827-2

## 全球销售及服务中心

### 美洲

公司总部 **Corporate Office**  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:  
<http://www.microchip.com/support>

网址: [www.microchip.com](http://www.microchip.com)

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

**奥斯汀 Austin, TX**  
Tel: 1-512-257-3370

**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

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

**底特律 Detroit**  
Novi, MI  
Tel: 1-248-848-4000

**休斯敦 Houston, TX**  
Tel: 1-281-894-5983

**印第安纳波利斯 Indianapolis**  
Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453  
Tel: 1-317-536-2380

**洛杉矶 Los Angeles**  
Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608  
Tel: 1-951-273-7800

**罗利 Raleigh, NC**  
Tel: 1-919-844-7510

**纽约 New York, NY**  
Tel: 1-631-435-6000

**圣何塞 San Jose, CA**  
Tel: 1-408-735-9110  
Tel: 1-408-436-4270

**加拿大多伦多 Toronto**  
Tel: 1-905-695-1980  
Fax: 1-905-695-2078

### 亚太地区

**中国 - 北京**  
Tel: 86-10-8569-7000

**中国 - 成都**  
Tel: 86-28-8665-5511

**中国 - 重庆**  
Tel: 86-23-8980-9588

**中国 - 东莞**  
Tel: 86-769-8702-9880

**中国 - 广州**  
Tel: 86-20-8755-8029

**中国 - 杭州**  
Tel: 86-571-8792-8115

**中国 - 南京**  
Tel: 86-25-8473-2460

**中国 - 青岛**  
Tel: 86-532-8502-7355

**中国 - 上海**  
Tel: 86-21-3326-8000

**中国 - 沈阳**  
Tel: 86-24-2334-2829

**中国 - 深圳**  
Tel: 86-755-8864-2200

**中国 - 苏州**  
Tel: 86-186-6233-1526

**中国 - 武汉**  
Tel: 86-27-5980-5300

**中国 - 西安**  
Tel: 86-29-8833-7252

**中国 - 厦门**  
Tel: 86-592-238-8138

**中国 - 香港特别行政区**  
Tel: 852-2943-5100

**中国 - 珠海**  
Tel: 86-756-321-0040

**台湾地区 - 高雄**  
Tel: 886-7-213-7830

**台湾地区 - 台北**  
Tel: 886-2-2508-8600

**台湾地区 - 新竹**  
Tel: 886-3-577-8366

### 亚太地区

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

**印度 India - Bangalore**  
Tel: 91-80-3090-4444

**印度 India - New Delhi**  
Tel: 91-11-4160-8631

**印度 India - Pune**  
Tel: 91-20-4121-0141

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770

**韩国 Korea - Daegu**  
Tel: 82-53-744-4301

**韩国 Korea - Seoul**  
Tel: 82-2-554-7200

**马来西亚 Malaysia - Kuala Lumpur**  
Tel: 60-3-7651-7906

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

**菲律宾 Philippines - Manila**  
Tel: 63-2-634-9065

**新加坡 Singapore**  
Tel: 65-6334-8870

**泰国 Thailand - Bangkok**  
Tel: 66-2-694-1351

**越南 Vietnam - Ho Chi Minh**  
Tel: 84-28-5448-2100

### 欧洲

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

**丹麦 Denmark - Copenhagen**  
Tel: 45-4485-5910  
Fax: 45-4485-2829

**芬兰 Finland - Espoo**  
Tel: 358-9-4520-820

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

**德国 Germany - Garching**  
Tel: 49-8931-9700

**德国 Germany - Haan**  
Tel: 49-2129-3766400

**德国 Germany - Heilbronn**  
Tel: 49-7131-72400

**德国 Germany - Karlsruhe**  
Tel: 49-721-625370

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

**德国 Germany - Rosenheim**  
Tel: 49-8031-354-560

**以色列 Israel - Ra'anana**  
Tel: 972-9-744-7705

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

**意大利 Italy - Padova**  
Tel: 39-049-7625286

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

**挪威 Norway - Trondheim**  
Tel: 47-7288-4388

**波兰 Poland - Warsaw**  
Tel: 48-22-3325737

**罗马尼亚 Romania - Bucharest**  
Tel: 40-21-407-87-50

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

**瑞典 Sweden - Gothenburg**  
Tel: 46-31-704-60-40

**瑞典 Sweden - Stockholm**  
Tel: 46-8-5090-4654

**英国 UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820