



MICROCHIP

# PIC24FJ128GA310 系列

## 采用 nanoWatt XLP 技术、带 LCD 控制器的 64/80/100 引脚通用 16 位闪存单片机

### 超低功耗特性:

- 多种功耗管理选项可将功耗降至超低水平:
  - V<sub>BAT</sub> 允许器件切换到备用电池, 将使用 RTCC 时的功耗降至最低
  - 深度休眠能够实现接近于完全掉电的运行状态, 同时能够在外部触发时唤醒
  - 休眠和空闲模式有选择地关闭外设和/或内核, 从而大幅降低功耗和实现快速唤醒
  - 打盹模式允许 CPU 以低于外设的时钟速度运行
- 备用时钟模式允许即时切换到低时钟速度以选择性降低功耗
- 深度休眠的超低功耗电流消耗:
  - WDT: 3.3V 时典型值为 270 nA
  - RTCC: 32 kHz、3.3V 时典型值为 400 nA
  - 深度休眠电流, 3.3V 时典型值为 40 nA

### 外设特性:

- LCD 显示控制器:
  - 最多 60 段, 8 个公共端
  - 内部电荷泵和低功耗内部电阻偏置
  - 休眠模式下工作
- 最多五个外部中断源
- 外设引脚选择 (Peripheral Pin Select, PPS): 允许对许多外设进行独立的 I/O 映射
- 五个带预分频器的 16 位定时器 / 计数器:
  - 可配对成 32 位定时器 / 计数器
- 六通道 DMA 支持所有外设模块
  - 最大程度减小 CPU 开销并提高数据吞吐量

### 外设特性 (续):

- 七个输入捕捉模块, 每个带有一个专用 16 位定时器
- 七个输出比较 /PWM 模块, 每个带有一个专用 16 位定时器
- 增强型并行主 / 从端口 (Enhanced Parallel Master/Slave Port, EPMP/EPSP)
- 硬件实时时钟 / 日历 (Real-Time Clock/Calendar, RTCC):
  - 在休眠和 V<sub>BAT</sub> 模式下运行
- 2 个 3 线 /4 线 SPI 模块 (支持 4 帧模式), 带 8 级 FIFO 缓冲区
- 2 个 I<sup>2</sup>C<sup>TM</sup> 模块, 支持多主器件 / 从模式和 7 位 / 10 位寻址
- 4 个 UART 模块:
  - 支持 RS-485、RS-232 和 LIN/J2602
  - IrDA<sup>®</sup> 用片上硬件编码器 / 解码器
  - 自动波特率检测时自动唤醒
  - 4 级深 FIFO 缓冲区
- 32 位可编程循环冗余校验 (Cyclic Redundancy Check, CRC) 发生器
- 数字信号调制器为数字信号流提供片上 FSK 和 PSK 调制
- 数字 I/O 引脚上的可配置漏级开路输出
- 所有 I/O 引脚均有高灌 / 拉电流 (18 mA / 18 mA)

### 模拟特性:

- 24 通道 10/12 位模数 (Analog-to-Digital, A/D) 转换器:
  - 500 ksps (10 位) 和 200 ksps (12 位) 的转换速率
  - 可在休眠和空闲模式下进行转换
- 带可编程输入 / 输出配置的三个轨到轨增强型模拟比较器
- 片上可编程参考电压
- 充电时间测量单元 (Charge Time Measurement Unit, CTMU):
  - 用于电容式触摸传感, 最多 24 通道
  - 时间测量分辨率最高达 1 ns
  - CTMU 温度检测

器件	引脚	存储器		可重映射的外设				I <sup>2</sup> C <sup>TM</sup>	10/12 位 A/D (通道)	比较器	CTMU (通道)	EPMP/EPSP	LCD (像素)	JTAG	采用 V <sub>BAT</sub> 的 深度休眠	
		闪存程序 (字节)	SRAM (字节)	16 位定时器	捕捉输入	比较 / PWM 输出	I <sup>2</sup> C <sup>TM</sup> 连接的 UART									
PIC24FJ128GA310	100	128K	8K	5	7	7	4	2	24	3	24	有	480	有	有	
PIC24FJ128GA308	80	128K	8K	5	7	7	4	2	2	16	3	16	有	368	有	有
PIC24FJ128GA306	64	128K	8K	5	7	7	4	2	2	16	3	16	有	240	有	有
PIC24FJ64GA310	100	64K	8K	5	7	7	4	2	2	24	3	24	有	480	有	有
PIC24FJ64GA308	80	64K	8K	5	7	7	4	2	2	16	3	16	有	368	有	有
PIC24FJ64GA306	64	64K	8K	5	7	7	4	2	2	16	3	16	有	240	有	有

# PIC24FJ128GA310 系列

---

---

## 高性能 CPU:

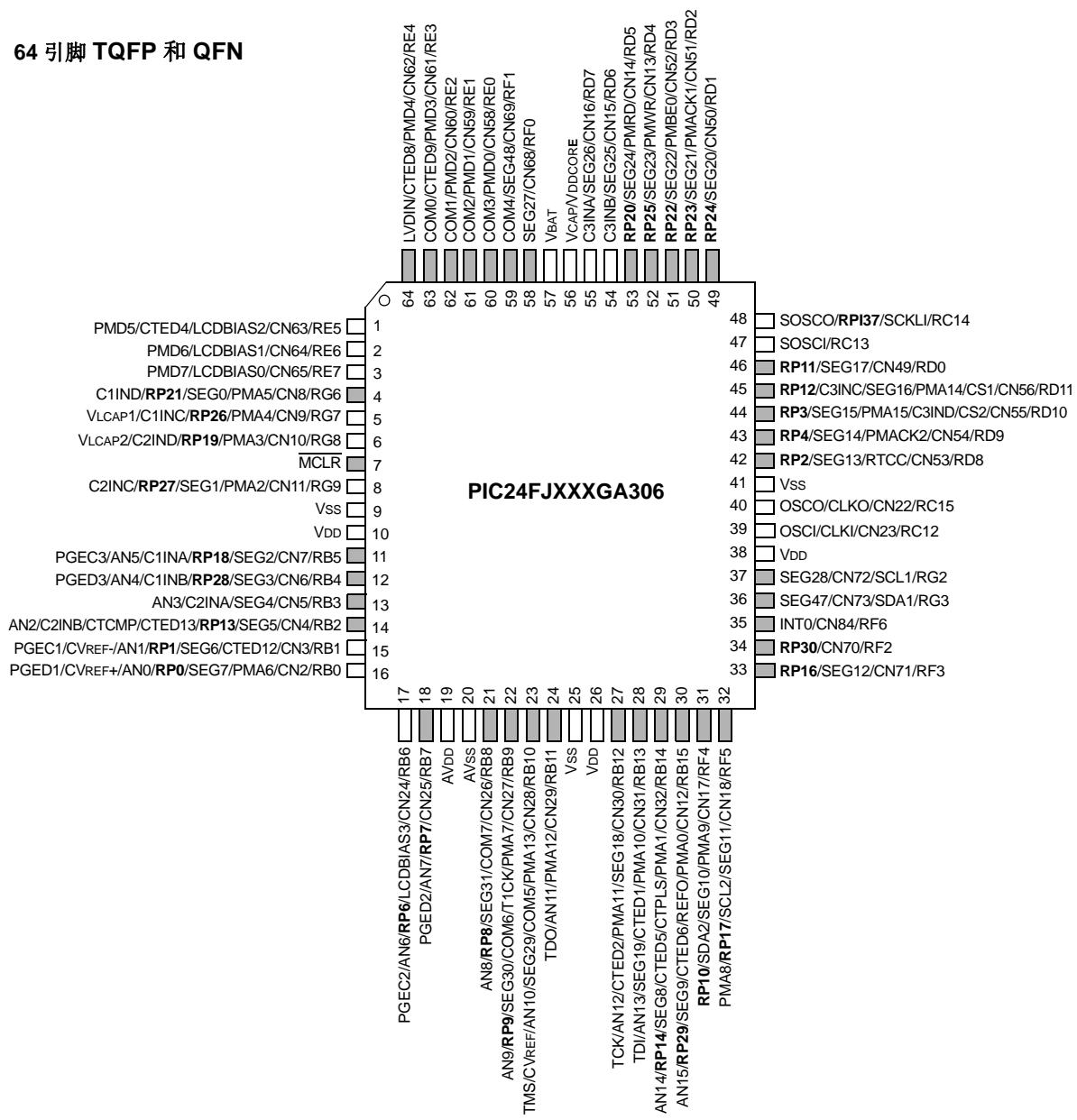
- 改进型哈佛架构
- 最高运行速度可达 16 MIPS (32 MHz 时)
- 8 MHz 内部振荡器:
  - 4x PLL 选项
  - 多个时钟分频选项
  - 快速起振
- 17 位 X 17 位单周期硬件小数 / 整数乘法器
- 32 位 /16 位硬件除法器
- 16 位 X 16 位工作寄存器阵列
- 优化的 C 编译器指令集架构
- 2 个地址发生单元，分别用于数据存储器的读 / 写寻址

## 单片机的特性:

- 工作电压范围为 2.0V 至 3.6V
- 两个片内稳压器 (1.8V 和 1.2V) 分别用于普通和超低功耗运行
- 可承受 20,000 次擦 / 写 (典型值) 的闪存程序存储器
- 闪存数据保持: 至少 20 年
- 可在软件控制下自编程
- 可编程参考时钟输出
- 通过 2 个引脚进行在线串行编程 (In-Circuit Serial Programming™, ICSPTM) 和在线仿真 (In-Circuit Emulation, ICE)
- 支持 JTAG 边界扫描
- 故障保护时钟监视器操作:
  - 检测时钟故障并切换至片上低功耗 RC 振荡器
- 上电复位 (Power-on Reset, POR)、上电延时定时器 (Power-up Timer, PWRT) 和振荡器起振定时器 (Oscillator Start-up Timer, OST)
- 运行电压低于 VBOR 时欠压复位 (Brown-out Reset, BOR)
- 低电压检测 (Low-Voltage Detect, LVD)
- 灵活的看门狗定时器 (Watchdog Timer, WDT) 自带 RC 振荡器以实现可靠运行
- 标准和超低功耗看门狗定时器 (WDT) 用于在标准和深度休眠模式下可靠运行

## 引脚图

### 64 引脚 TQFP 和 QFN

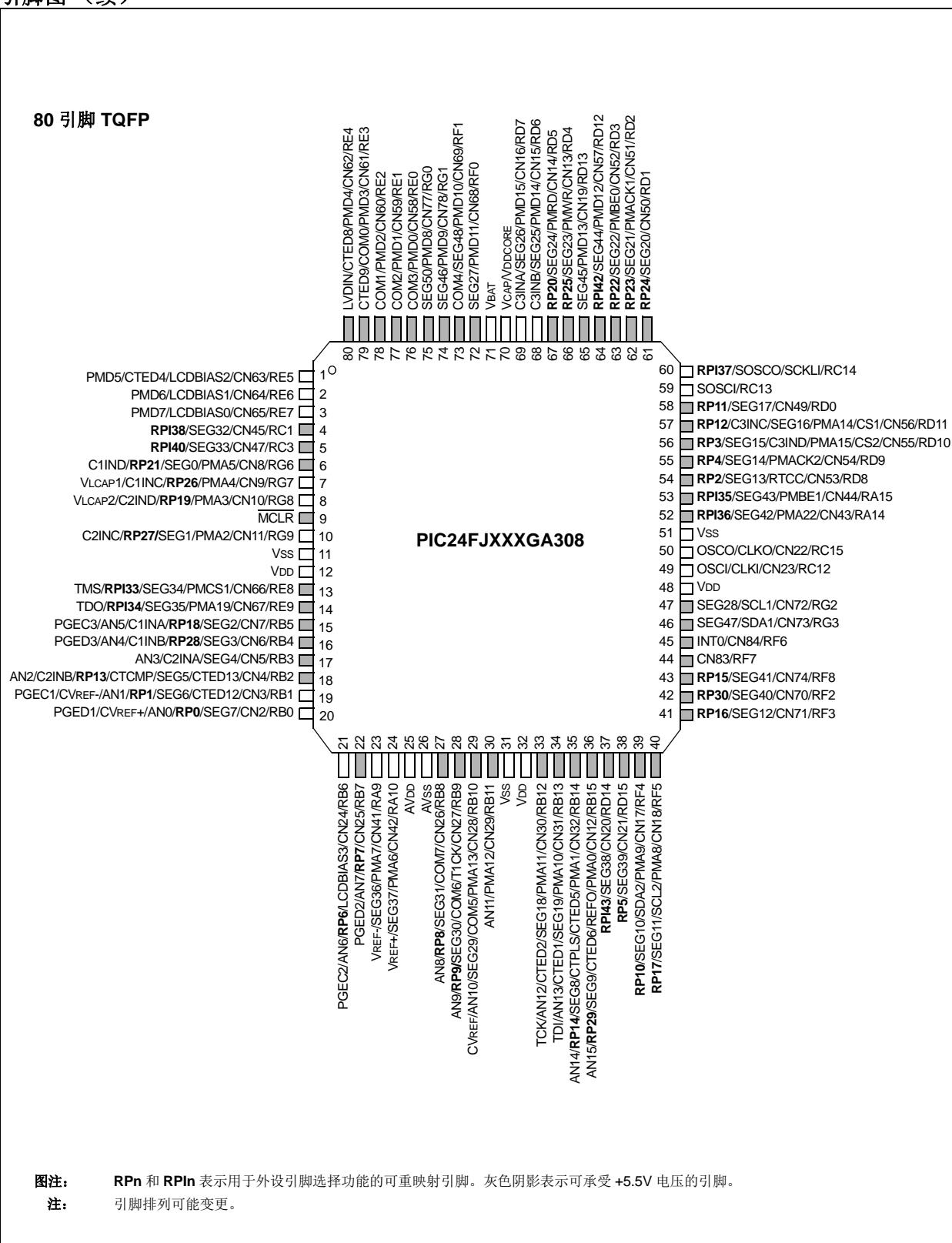


**图注：** RP<sub>n</sub> 和 RPIn 表示用于外设引脚选择功能的可重映射引脚。灰色阴影表示可承受 +5.5V 电压的引脚。  
**注：** 引脚排列可能变更。

# PIC24FJ128GA310 系列

## 引脚图 (续)

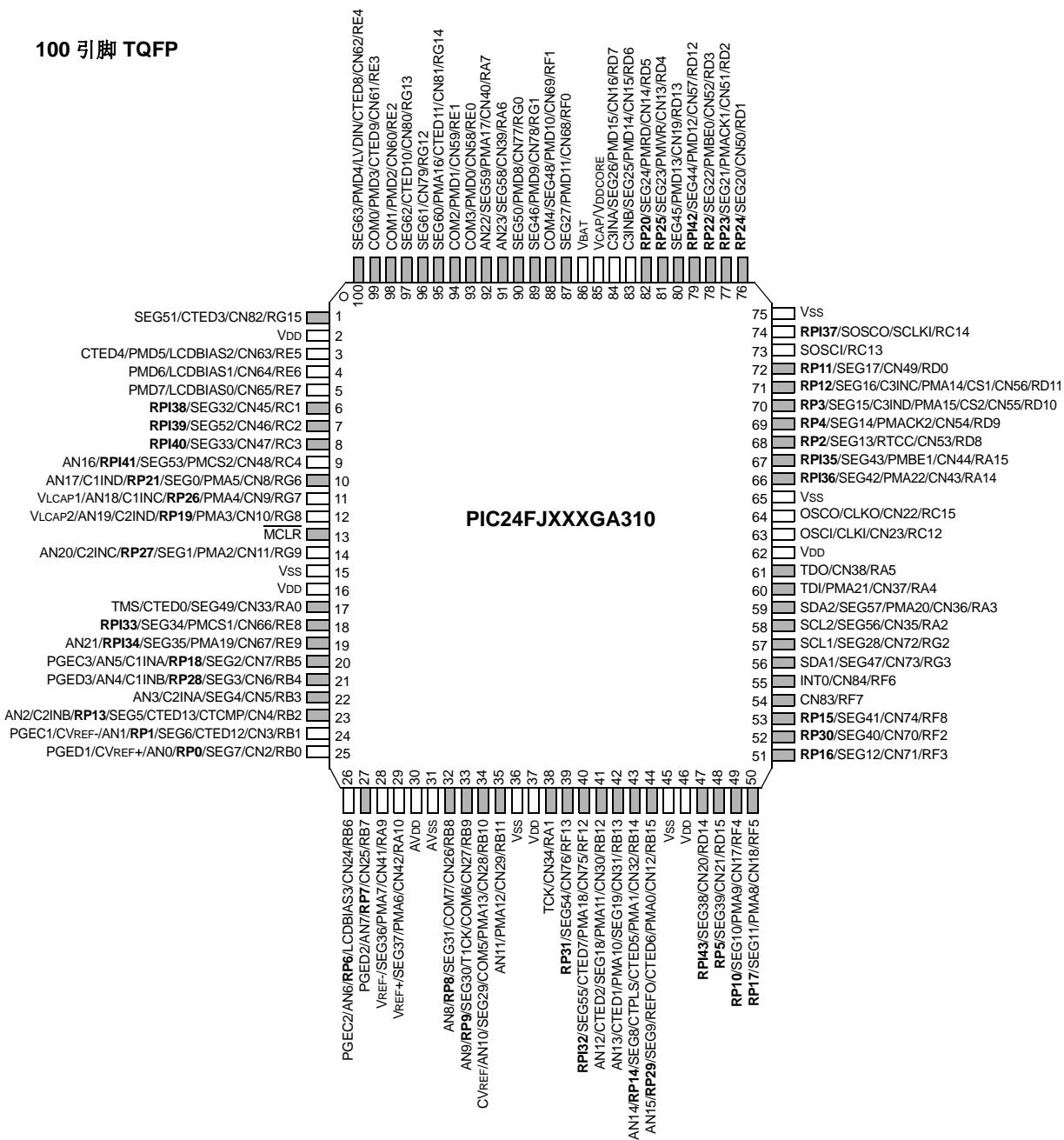
80 引脚 TQFP



**图注：** RPn 和 RPin 表示用于外设引脚选择功能的可重映射引脚。灰色阴影表示可承受 +5.5V 电压的引脚。  
**注：** 引脚排列可能变更。

## 引脚图 (续)

100 引脚 TQFP



**图注：** R<sub>Pn</sub> 和 R<sub>PIn</sub> 表示用于外设引脚选择功能的可重映射引脚。灰色阴影表示可承受 +5.5V 电压的引脚。  
**注：** 引脚排列可能变更。

# PIC24FJ128GA310 系列

## 引脚图 (续)

121 引脚 BGA (俯视图)

	1	2	3	4	5	6	7	8	9	10	11
A	RE4	RE3	RG13	RE0	RG0	RF1	VBAT	N/C	RD12	RD2	RD1
B	N/C	RG15	RE2	RE1	RA7	RF0	VCAP/ VDDCORE	RD5	RD3	Vss	RC14
C	RE6	VDD	RG12	RG14	RA6	N/C	RD7	RD4	N/C	RC13	RD11
D	RC1	RE7	RE5	N/C	N/C	N/C	RD6	RD13	RD0	N/C	RD10
E	RC4	RC3	RG6	RC2	N/C	RG1	N/C	RA15	RD8	RD9	RA14
F	MCLR	RG8	RG9	RG7	Vss	N/C	N/C	VDD	OSCI/ RC12	Vss	OSCO/ RC15
G	RE8	RE9	RA0	N/C	VDD	VSS	N/C	RA5	RA3	RA4	
H	RB5	RB4	N/C	N/C	N/C	VDD	N/C	RF7	RF6	RG2	RA2
J	RB3	RB2	RB7	AVDD	RB11	RA1	RB12	N/C	N/C	RF8	RG3
K	RB1	RB0	RA10	RB8	N/C	RF12	RB14	Vdd	RD15	RF3	RF2
L	RB6	RA9	AVss	RB9	RB10	RF13	RB13	RB15	RD14	RF4	RF5

图注：灰色阴影表示可承受 +5.5V 电压的引脚。

注：有关引脚排列的详细说明，请参见表 1。引脚排列可能变更。

表 1： 121 引脚器件的完整引脚功能描述

引脚	功能	引脚	功能
A1	SEG63/PMD4/LVDIN/CTED8/CN62/RE4	E1	AN16/RPI41/SEG53/PMCS2/CN48/RC4
A2	COM0/PMD3/CTED9/CN61/RE3	E2	RPI40/SEG33/CN47/RC3
A3	SEG62/CTED10/CN80/RG13	E3	AN17/C1IND/RP21/SEG0/PMA5/CN8/RG6
A4	COM3/PMD0/CN58/RE0	E4	RPI39/SEG52/CN46/RC2
A5	SEG50/PMD8/CN77/RG0	E5	N/C
A6	SEG48/COM4/PMD10/CN69/RF1	E6	SEG46/PMD9/CN78/RG1
A7	VBAT	E7	N/C
A8	N/C	E8	RPI35/SEG43/PMBE1/CN44/RA15
A9	<b>RPI42</b> /SEG44/PMD12/CN57/RD12	E9	RP2/SEG13/RTCC/CN53/RD8
A10	<b>RP23</b> /SEG21/PMACK1/CN51/RD2	E10	RP4/SEG14/PMACK2/CN54/RD9
A11	<b>RP24</b> /SEG20/CN50/RD1	E11	RPI36/SEG42/PMA22/CN43/RA14
B1	N/C	F1	MCLR
B2	SEG51/CTED3/CN82/RG15	F2	VLCAP2/AN19/C2IND/RP19/PMA3/CN10/RG8
B3	COM1/PMD2/CN60/RE2	F3	AN20/C2INC/RP27/SEG1/PMA2/CN11/RG9
B4	COM2/PMD1/CN59/RE1	F4	VLCAP1/AN18/C1INC/RP26/PMA4/CN9/RG7
B5	AN22/SEG59/PMA17/CN40/RA7	F5	VSS
B6	SEG27/PMD11/CN68/RF0	F6	N/C
B7	VCAP	F7	N/C
B8	<b>RP20</b> /SEG24/PMRD/CN14/RD5	F8	VDD
B9	<b>RP22</b> /SEG22/PMBE0/CN52/RD3	F9	OSCI/CLK1/CN23/RC12
B10	Vss	F10	VSS
B11	RPI37/SOSCO/SCLKI/RC14	F11	OSCO/CLKO/CN22/RC15
C1	PMD6/LCDBIAS1/CN64/RE6	G1	RPI33/SEG34/PMCS1/CN66/RE8
C2	VDD	G2	AN21/RPI34/SEG35/PMPA19/CN67/RE9
C3	SEG61/CN79/RG12	G3	TMS/SEG49/CTED0/CN33/RA0
C4	SEG60/PMA16/CTED11/CN81/RG14	G4	N/C
C5	AN23/SEG58/CN39/RA6	G5	VDD
C6	N/C	G6	VSS
C7	C3INA/SEG26/PMD15/CN16/RD7	G7	VSS
C8	<b>RP25</b> /SEG23/PMWR/CN13/RD4	G8	N/C
C9	N/C	G9	TDO/CN38/RA5
C10	SOSCI/RC13	G10	SDA2/SEG57/PMA20/CN36/RA3
C11	<b>RP12</b> /SEG16/C3INC/PMA14/CS1/CN56/RD11	G11	TDI/PMA21/CN37/RA4
D1	<b>RPI38</b> /SEG32/CN45/RC1	H1	PGEC3/AN5/C1INA/ <b>RP18</b> /SEG2/CN7/RB5
D2	PMD7/LCDBIAS0/CN65/RE7	H2	PGED3/AN4/C1INB/ <b>RP28</b> /SEG3/CN6/RB4
D3	PMD5/CTED4/LCDBIAS2/CN63/RE5	H3	N/C
D4	N/C	H4	N/C
D5	N/C	H5	N/C
D6	N/C	H6	VDD
D7	C3INB/SEG25/PMD14/CN15/RD6	H7	N/C
D8	SEG45/PMD13/CN19/RD13	H8	CN83/RF7
D9	<b>RP11</b> /SEG17/CN49/RD0	H9	INT0/CN84/RF6
D10	N/C	H10	SCL1/SEG28/CN72/RG2
D11	<b>RP3</b> /SEG15/C3IND/PMA15/CS2/CN55/RD10	H11	SCL2/SEG56/CN35/RA2

图注： **R<sub>Pn</sub>** 和 **R<sub>Pln</sub>** 表示用于外设引脚选择功能的可重映射引脚。

注： 引脚排列可能变更。

# PIC24FJ128GA310 系列

表 1：121 引脚器件的完整引脚功能描述（续）

引脚	功能	引脚	功能
J1	AN3/C2INA/SEG4/CN5/RB3	K7	AN14/ <b>RP14</b> /SEG8/CTPLS/CTED5/PMA1/CN32/RB14
J2	AN2/C2INB/ <b>RP13</b> /SEG5/CTCMP/CTED13/CN4/RB2	K8	V <sub>DD</sub>
J3	PGED2/AN7/ <b>RP7</b> /CN25/RB7	K9	<b>RP5</b> /SEG39/CN21/RD15
J4	AV <sub>DD</sub>	K10	<b>RP16</b> /SEG12/CN71/RF3
J5	AN11/PMA12/CN29/RB11	K11	<b>RP30</b> /SEG40/CN70/RF2
J6	TCK/CN34/RA1	L1	PGEC2/AN6/ <b>RP6</b> /LCDBIAS3/CN24/RB6
J7	AN12/SEG18/CTED2/PMA11/CN30/RB12	L2	V <sub>REF-</sub> /SEG36/PMA7/CN41/RA9
J8	N/C	L3	AV <sub>SS</sub>
J9	N/C	L4	AN9/ <b>RP9</b> /COM6/SEG30/T1CK/CN27/RB9
J10	<b>RP15</b> /SEG41/CN74/RF8	L5	CV <sub>REF</sub> /AN10/COM5/SEG29/PMA13/CN28/RB10
J11	SDA1/SEG47/CN73/RG3	L6	<b>RP31</b> /SEG54/CN76/RF13
K1	PGEC1/CV <sub>REF</sub> -/AN1/RP1/SEG6/CTED12/CN3/RB1	L7	AN13/SEG19/CTED1/PMA10/CN31/RB13
K2	PGD1/CV <sub>REF</sub> +/AN0/ <b>RP0</b> /SEG7/CN2/RB0	L8	AN15/ <b>RP29</b> /SEG9/CTED6/REFO/PMA0/CN12/RB15
K3	V <sub>REF</sub> +/SEG37/PMA6/CN42/RA10	L9	<b>RP143</b> /SEG38/CN20/RD14
K4	AN8/ <b>RP8</b> /COM7/SEG31/CN26/RB8	L10	<b>RP10</b> /SEG10/PMA9/CN17/RF4
K5	N/C	L11	<b>RP17</b> /SEG11/PMA8/CN18/RF5
K6	<b>RP132</b> /SEG55/CTED7/PMA18/CN75/RF12		

图注：**RPn** 和 **RPin** 表示用于外设引脚选择功能的可重映射引脚。

注：引脚排列可能变更。

## 目录

1.0	器件概述	11
2.0	16 位单片机入门指南	29
3.0	CPU	35
4.0	存储器构成	41
5.0	直接存储器访问控制器（DMA）	75
6.0	闪存程序存储器	83
7.0	复位	89
8.0	中断控制器	95
9.0	振荡器配置	145
10.0	节能特性	155
11.0	I/O 端口	167
12.0	Timer1	197
13.0	Timer2/3 和 Timer4/5	199
14.0	带专用定时器的输入捕捉	205
15.0	带专用定时器的输出比较	211
16.0	串行外设接口（SPI）	221
17.0	$\text{I}^2\text{C}^{\text{TM}}$	233
18.0	通用异步收发器（UART）	241
19.0	数据信号调制器	249
20.0	增强型并行主端口（EPMP）	253
21.0	液晶显示（LCD）控制器	265
22.0	实时时钟和日历（RTCC）	275
23.0	32 位可编程循环冗余校验（CRC）发生器	289
24.0	带阈值扫描功能的 12 位 A/D 转换器	295
25.0	三比较器模块	315
26.0	比较器参考电压	321
27.0	充电时间测量单元（CTMU）	323
28.0	高 / 低压检测（HLVD）	331
29.0	特殊功能	333
30.0	开发支持	347
31.0	指令集汇总	351
32.0	电气特性	359
33.0	封装信息	377
	附录 A: 版本历史	393
	索引	395
	Microchip 网站	401
	变更通知客户服务	401
	客户支持	401
	读者反馈表	402
	产品标识体系	403

## 致 客 户

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

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

### 最新数据手册

欲获得本数据手册的最新版本，请查询我公司的网站：

<http://www.microchip.com>

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

### 勘误表

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

欲了解某一器件是否存在勘误表，请通过以下方式之一查询：

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

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

### 客户通知系统

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

## 1.0 器件概述

本文档包含以下器件的具体信息：

- PIC24FJ64GA306      • PIC24FJ128GA306
- PIC24FJ64GA308      • PIC24FJ128GA308
- PIC24FJ64GA310      • PIC24FJ128GA310

PIC24FJ128GA310 系列为 Microchip 的 16 位单片机增加了许多新特性，包括新的超低功耗特性、针对外设的直接存储器访问（Direct Memory Access, DMA），以及内置 LCD 控制器和驱动器。这些特性结合在经济节能的封装内，提供各种强大的功能。

## 1.1 内核特性

### 1.1.1 16 位架构

所有 PIC24F 器件的内核都采用了 16 位改进型哈佛架构，该架构最初是在 Microchip 的 dsPIC® 数字信号控制器（Digital Signal Controller, DSC）中引入的。PIC24F CPU 内核提供了大量增强功能，例如：

- 16 位数据路径和 24 位地址路径，可在数据空间和程序空间之间传送信息
- 线性寻址能力最高可达 12 MB（程序空间）和 32 KB（数据空间）
- 利用内建软件堆栈支持 16 元工作寄存器阵列
- 支持整数算术运算的 17 位 x 17 位硬件乘法器
- 支持 32 位 /16 位除法运算的硬件
- 支持多种寻址模式并针对高级语言（如 C 语言）进行优化的指令集
- 工作性能最高可达 16 MIPS

### 1.1.2 nanoWatt XLP 节能技术

PIC24FJ128GA310 系列器件引入了大量节能工作模式，以实现最高程度的节能。新增模式包括：

- 保持休眠模式，由单独的低压稳压器为基本电路供电
- 深度休眠模式（RTCC 不工作），在软件控制下实现尽可能低的功耗
- VBAT 模式（RTCC 工作或不工作），在 VDD 移除后通过备用电池继续有限工作

这些新的低功耗模式有许多还支持低功耗片上实时时钟 / 日历（RTCC）的连续工作，使应用程序可在器件处于休眠状态时保持时间。

除这些新特性外，PIC24FJ128GA310 系列器件还包括以前 PIC24F 单片机的所有传统节能特性，例如：

- 动态时钟切换，允许在运行时选择低功耗时钟
- 打盹工作模式，在降低 CPU 时钟速度的同时保持外设时钟速度
- 基于指令的节能模式，快速调用空闲模式和多种休眠模式。

### 1.1.3 振荡器选项和特性

PIC24FJ128GA310 系列中的所有器件均提供五种不同的振荡器选项，使用户在开发应用硬件时有很大的选择范围。这些选项包括：

- 两种晶振模式
- 两种外部时钟模式
- 一个锁相环（Phase Lock Loop, PLL）倍频器，最高可使时钟速度达到 32 MHz
- 快速内部振荡器（Fast Internal Oscillator, FRC）（标称 8 MHz 输出），具有多个分频器选项
- 独立低功耗内部 RC 振荡器（Low-Power Internal RC Oscillator, LPRC）（31 kHz 标称值），用于对时序要求不高的低功耗应用。

内部振荡器模块还为故障保护时钟监视器（Fail-Safe Clock Monitor, FSCM）提供了一个稳定的参考源。故障保护时钟监视器不断地监视主时钟源，将之与内部振荡器提供的参考信号作比较。一旦发生时钟故障，允许控制器将时钟源切换到内部振荡器，继续保持低速工作或安全地关闭应用。

### 1.1.4 易于移植

无论存储器容量如何，所有器件均共享同一组丰富外设，使得应用程序在扩展和升级时移植变得很方便。整个系列使用相同的引脚配置方案也有助于从一个器件向下一代更大型器件移植，或甚至从 64 引脚到 100 引脚器件移植。

PIC24F 系列器件的引脚同 dsPIC33 系列器件的引脚是兼容的，并与 PIC18 和 dsPIC30 器件的引脚配置方案部分兼容。这样仍然可以使用 Microchip 器件而将相对简单的应用顺利移植为强大而复杂的应用。

# PIC24FJ128GA310 系列

## 1.2 DMA 控制器

PIC24FJ128GA310 系列器件还为 PIC24 架构引入了新的直接存储器访问 (DMA) 控制器。此模块的工作与 CPU 互不冲突，允许数据在数据存储器与外设之间移动而无需 CPU 干预，从而提高数据吞吐量并缩短执行时间。六条独立可编程通道几乎可同时为多个外设提供服务，每个通道外设都执行不同的操作。支持多种类型的数据传输操作。

## 1.3 LCD 控制器

通过 PIC24FJ128GA310 系列器件，Microchip 将其多功能液晶显示 (Liquid Crystal Display, LCD) 控制器和驱动器引入到 PIC24F 系列。片上 LCD 驱动器包括多种特性，可以更方便地在低功耗应用中集成显示功能。这些特性包括带电荷泵的集成稳压器，以及允许用软件进行对比度控制和在电压高于器件 V<sub>DD</sub> 时进行显示操作的集成内部梯形电阻网络。

## 1.4 其他特殊功能

- 外设引脚选择：**外设引脚选择 (PPS) 功能允许大部分的数字外设通过一组固定的数字 I/O 引脚进行映射。用户可将许多数字外设之一的输入和 / 或输出独立地映射到其中的任一 I/O 引脚。
- 通信：**PIC24FJ128GA310 系列器件集成了许多串行通信外设以处理各种应用需求。提供了两个同时支持主从工作模式的独立 I<sup>2</sup>C<sup>TM</sup> 模块。通过 PPS 功能，器件还具有四个包含内置 IrDA<sup>®</sup> 编码器 / 解码器的独立 UART 以及两个 SPI 模块。
- 模拟特性：**PIC24FJ128GA310 系列的所有器件都包含一个新的 12 位 A/D 转换器 (A/D) 模块和一个三比较器模块。A/D 模块包含一系列新功能，允许转换器对传入数据进行访问和判断，从而降低例行 A/D 转换的 CPU 开销。比较器模块包含三个模拟比较器，它们可针对多种操作进行配置。
- CTMU 接口：**除了其他模拟功能外，PIC24FJ128GA310 系列器件包含 CTMU 接口模块。此模块为精确时间测量和脉冲生成提供了一个便利的方法，并可用作容性传感器的接口。

- 增强型并行主 / 并行从端口：**此模块允许快速直观地访问单片机数据总线，并允许 CPU 直接对外部数据存储器进行寻址。并行端口可在主模式或从模式下工作，接受 4、8 或 16 位的数据宽度，可在主模式下进行最高 23 位寻址。
- 实时时钟和日历 (RTCC)：**此模块通过硬件实现带有闹钟功能的全功能时钟和日历，从而释放了定时器资源和程序存储空间供核心应用程序使用。
- 数据信号调制器 (DSM)：**数据信号调制器 (Data Signal Modulator, DSM) 允许用户将电子数据流 (“调制器信号”) 与载波信号混合来产生调制输出。

## 1.5 系列中各器件的详细说明

PIC24FJ128GA310 系列中的器件具有 64 引脚、80 引脚和 100 引脚三类封装形式。[图 1-1](#) 给出了所有器件的一般框图。

这些器件在以下六个方面存在差异：

- 闪存程序存储器 (PIC24FJ64GA3XX 器件 64 KB； PIC24FJ128GA3XX 器件 128 KB)。
- 可用的 I/O 引脚和端口 (64 引脚器件有 6 个端口 53 个引脚，80 引脚器件有 7 个端口 69 个引脚，100 引脚器件有 7 个端口 85 个引脚)。
- 可用的电平变化中断通知 (Interrupt-on-Change Notification, ICN) 输入 (对于 64 引脚器件为 52，对于 80 引脚器件为 66，对于 100 引脚器件为 82)。
- 可用的重映射引脚 (64 引脚器件有 29 个引脚，80 引脚器件有 40 个引脚，100 引脚器件有 44 个引脚)。
- 可驱动的 LCD 像素的最大数 (64 引脚器件为 272，80 引脚器件为 368，100 引脚器件为 480)。
- 模拟输入通道 (64 引脚和 80 引脚器件有 16 通道，100 引脚器件有 24 通道)。

本系列器件的所有其他特性都是相同的。[表 1-1](#)、[表 1-2](#) 和 [表 1-3](#) 汇总了这些特性。

[表 1-4](#) 给出了 PIC24FJ128GA310 系列器件上可用的引脚功能的列表，按功能排序。注意，此表只显示了各个外设功能所使用的引脚的位置，而没有显示同一引脚上的多种功能的复用方式。在本数据手册开始部分的引脚图中提供了有关引脚复用的信息。复用的功能按功能的优先级排列，最前面的是优先级最高的外设功能。

# PIC24FJ128GA310 系列

表 1-1： PIC24FJ128GA310 系列器件的特性：64 引脚

特性	PIC24FJ64GA306	PIC24FJ128GA306
工作频率	DC – 32 MHz	
程序存储器（字节数）	64K	128K
程序存储器（指令数）	22,016	44,032
数据存储器（字节数）	8K	
中断源（软向量数 /NMI 陷阱数）	65 (61/4)	
I/O 端口	端口 B、C、D、E、F 和 G	
I/O 引脚总数	53	
可重映射的引脚数	30 (29 个 I/O, 仅 1 个输入)	
定时器：		
总数 (16 位)	5 <sup>(1)</sup>	
32 位 (由一对 16 位定时器组成)	2	
输入捕捉通道	7 <sup>(1)</sup>	
输出比较 /PWM 通道	7 <sup>(1)</sup>	
输入电平变化通知中断	52	
串行通信：		
UART	4 <sup>(1)</sup>	
SPI (3 线 /4 线)	2 <sup>(1)</sup>	
I <sup>2</sup> C <sup>TM</sup>	2	
数据信号调制器	有	
并行通信 (EPMP/PSP)	有	
JTAG 边界扫描	有	
12/10 位模数转换器 (A/D) 模块 (输入通道)	16	
模拟比较器	3	
CTMU 接口	有	
LCD 控制器 (可用像素)	240 (30 SEG x 8 COM)	
复位 (和延时)	内核 POR、VDD POR、VBAT POR、BOR、RESET 指令、 MCLR、WDT；非法操作码、REPEAT 指令、 硬件陷阱、配置字失配 (OST 和 PLL 锁定)	
指令集	76 条基本指令和多种寻址模式	
封装	64 引脚 TQFP 和 QFN	

注 1： 外设可通过可重映射的引脚访问。

# PIC24FJ128GA310 系列

表 1-2： PIC24FJ128GA310 系列器件的特性： 80 引脚

特性	PIC24FJ64GA308	PIC24FJ128GA308
工作频率	DC – 32 MHz	
程序存储器（字节数）	64K	128K
程序存储器（指令数）	22,016	44,032
数据存储器（字节数）	8K	
中断源（软向量数/NMI 陷阱数）	65 (61/4)	
I/O 端口	端口 A、B、C、D、E、F 和 G	
I/O 引脚总数	69	
可重映射的引脚数	40 (31 个 I/O, 仅 9 个输入)	
定时器：		
总数 (16 位)	5 <sup>(1)</sup>	
32 位 (由一对 16 位定时器组成)	2	
输入捕捉通道	7 <sup>(1)</sup>	
输出比较 /PWM 通道	7 <sup>(1)</sup>	
输入电平变化通知中断	66	
串行通信：		
UART	4 <sup>(1)</sup>	
SPI (3 线 /4 线)	2 <sup>(1)</sup>	
I <sup>2</sup> C <sup>TM</sup>	2	
数据信号调制器	有	
并行通信 (EPMP/PSP)	有	
JTAG 边界扫描	有	
12/10 位模数转换器 (A/D) 模块 (输入通道)	16	
模拟比较器	3	
CTMU 接口	有	
LCD 控制器 (可用像素)	368 (46 SEG x 8 COM)	
复位 (和延时)	内核 POR、V <sub>DD</sub> POR、V <sub>BAT</sub> POR、BOR、RESET 指令、 MCLR、WDT；非法操作码、REPEAT 指令、 硬件陷阱、配置字失配 (OST 和 PLL 锁定)	
指令集	76 条基本指令和多种寻址模式	
封装	80 引脚 TQFP 和 QFN	

注 1： 外设可通过可重映射的引脚访问。

# PIC24FJ128GA310 系列

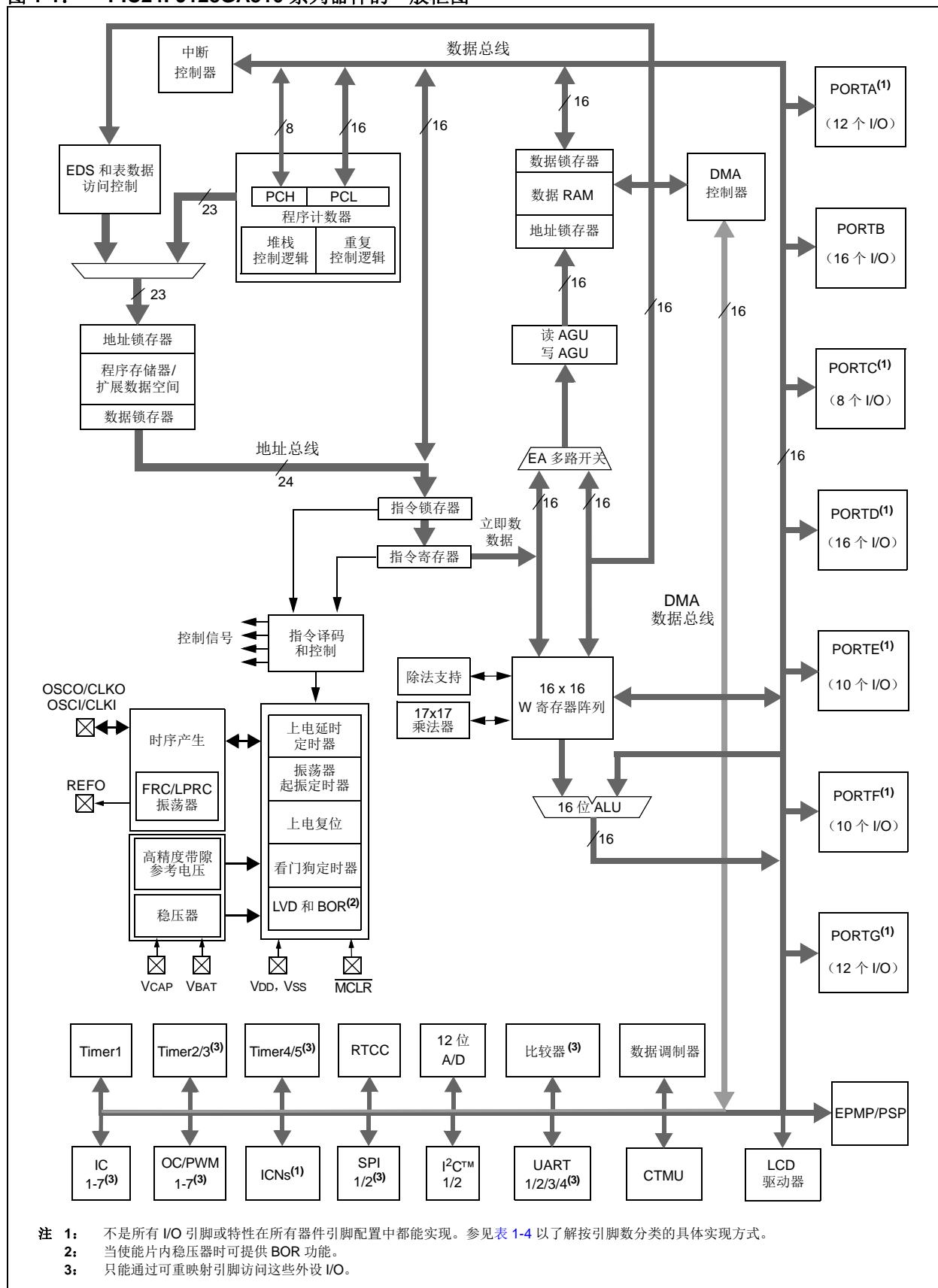
表 1-3： PIC24FJ128GA310 系列器件的特性：100 引脚器件

特性	PIC24FJ64GA310	PIC24FJ128GA310
工作频率	DC – 32 MHz	
程序存储器（字节数）	64K	128K
程序存储器（指令数）	22,016	44,032
数据存储器（字节数）	8K	
中断源（软向量 /NMI 陷阱）	66 (62/4)	
I/O 端口	端口 A、B、C、D、E、F 和 G	
I/O 引脚总数	85	
可重映射的引脚数	44 (32 个 I/O, 仅 12 个输入)	
定时器：		
总数 (16 位)	5 <sup>(1)</sup>	
32 位 (由一对 16 位定时器组成)	2	
输入捕捉通道	7 <sup>(1)</sup>	
输出比较 /PWM 通道	7 <sup>(1)</sup>	
输入电平变化通知中断	82	
串行通信：		
UART	4 <sup>(1)</sup>	
SPI (3 线 /4 线)	2 <sup>(1)</sup>	
I <sup>2</sup> C <sup>TM</sup>	2	
数据信号调制器	有	
并行通信 (EPMP/PSP)	有	
JTAG 边界扫描	有	
12/10 位模数转换器 (A/D) 模块 (输入通道)	24	
模拟比较器	3	
CTMU 接口	有	
LCD 控制器 (可用像素)	480 (60 SEG x 8 COM)	
复位 (和延时)	内核 POR、VDD POR、VBAT POR、BOR、RESET 指令、 MCLR、WDT；非法操作码、REPEAT 指令、 硬件陷阱、配置字失配 (OST 和 PLL 锁定)	
指令集	76 条基本指令和多种寻址模式	
封装	100 引脚 TQFP 和 121 引脚 BGA	

注 1： 外设可通过可重映射的引脚访问。

# PIC24FJ128GA310 系列

图 1-1：PIC24FJ128GA310 系列器件的一般框图



注 1：不是所有 I/O 引脚或特性在所有器件引脚配置中都能实现。参见表 1-4 以了解按引脚数分类的具体实现方式。

2：当使能片内稳压器时可提供 BOR 功能。

3：只能通过可重映射引脚访问这些外设 I/O。

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
AN0	16	20	25	K2	I	ANA	A/D 模拟输入。
AN1	15	19	24	K1	I	ANA	
AN1-	15	19	24	K1	I	ANA	
AN2	14	18	23	J2	I	ANA	A/D 模拟输入。
AN3	13	17	22	J1	I	ANA	
AN4	12	16	21	H2	I	ANA	
AN5	11	15	20	H1	I	ANA	
AN6	17	21	26	L1	I	ANA	
AN7	18	22	27	J3	I	ANA	
AN8	21	27	32	K4	I	ANA	
AN9	22	28	33	L4	I	ANA	
AN10	23	29	34	L5	I	ANA	
AN11	24	30	35	J5	I	ANA	
AN12	27	33	41	J7	I	ANA	
AN13	28	34	42	L7	I	ANA	
AN14	29	35	43	K7	I	ANA	
AN15	30	36	44	L8	I	ANA	
AN16	—	—	9	E1	I	ANA	
AN17	—	—	10	E3	I	ANA	
AN18	—	—	11	F4	I	ANA	
AN19	—	—	12	F2	I	ANA	
AN20	—	—	14	F3	I	ANA	
AN21	—	—	19	G2	I	ANA	
AN22	—	—	92	B5	I	ANA	
AN23	—	—	91	C5	I	ANA	
AVDD	19	25	30	J4	P	—	模拟模块的正电源。
AVSS	20	26	31	L3	P	—	模拟模块的参考地。
C1INA	11	15	20	H1	I	ANA	比较器 1 的输入 A。
C1INB	12	16	21	H2	I	ANA	比较器 1 的输入 B。
C1INC	5	7	11	F4	I	ANA	比较器 1 的输入 C。
C1IND	4	6	10	E3	I	ANA	比较器 1 的输入 D。
C2INA	13	17	22	J1	I	ANA	比较器 2 的输入 A。
C2INB	14	18	23	J2	I	ANA	比较器 2 的输入 B。
C2INC	8	10	14	F3	I	ANA	比较器 2 的输入 C。
C2IND	6	8	12	F2	I	ANA	比较器 2 的输入 D。
C3INA	55	69	84	C7	I	ANA	比较器 3 的输入 A。
C3INB	54	68	83	D7	I	ANA	比较器 3 的输入 B。
C3INC	45	57	71	C11	I	ANA	比较器 3 的输入 C。
C3IND	44	56	70	D11	I	ANA	比较器 3 的输入 D。
CLKI	39	49	63	F9	I	ANA	主时钟输入连接。
CLKO	40	50	64	F11	O	—	系统时钟输出。

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C$ ™ =  $I^2C/SMBus$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
CN2	16	20	25	K2	I	ST	电平变化中断输入。
CN3	15	19	24	K1	I	ST	
CN4	14	18	23	J2	I	ST	
CN5	13	17	22	J1	I	ST	
CN6	12	16	21	H2	I	ST	
CN7	11	15	20	H1	I	ST	
CN8	4	6	10	E3	I	ST	
CN9	5	7	11	F4	I	ST	
CN10	6	8	12	F2	I	ST	
CN11	8	10	14	F3	I	ST	
CN12	30	36	44	L8	I	ST	
CN13	52	66	81	C8	I	ST	
CN14	53	67	82	B8	I	ST	
CN15	54	68	83	D7	I	ST	
CN16	55	69	84	C7	I	ST	
CN17	31	39	49	L10	I	ST	
CN18	32	40	50	L11	I	ST	
CN19	—	65	80	D8	I	ST	
CN20	—	37	47	L9	I	ST	
CN21	—	38	48	K9	I	ST	
CN22	40	50	64	F11	I	ST	
CN23	39	49	63	F9	I	ST	
CN24	17	21	26	L1	I	ST	
CN25	18	22	27	J3	I	ST	
CN26	21	27	32	K4	I	ST	
CN27	22	28	33	L4	I	ST	
CN28	23	29	34	L5	I	ST	
CN29	24	30	35	J5	I	ST	
CN30	27	33	41	J7	I	ST	
CN31	28	34	42	L7	I	ST	
CN32	29	35	43	K7	I	ST	
CN33	—	—	17	G3	I	ST	
CN34	—	—	38	J6	I	ST	
CN35	—	—	58	H11	I	ST	
CN36	—	—	59	G10	I	ST	
CN37	—	—	60	G11	I	ST	
CN38	—	—	61	G9	I	ST	
CN39	—	—	91	C5	I	ST	
CN40	—	—	92	B5	I	ST	
CN41	—	23	28	L2	I	ST	
CN42	—	24	29	K3	I	ST	
CN43	—	52	66	E11	I	ST	

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C^{\text{TM}}$  =  $I^2C$ /SMBus 输入缓冲器

表 1-4: PIC24FJ128GA310 系列引脚排列说明 (续)

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
CN44	—	53	67	E8	I	ST	电平变化中断输入。
CN45	—	4	6	D1	I	ST	
CN46	—	—	7	E4	I	ST	
CN47	—	5	8	E2	I	ST	
CN48	—	—	9	E1	I	ST	
CN49	46	58	72	D9	I	ST	
CN50	49	61	76	A11	I	ST	
CN51	50	62	77	A10	I	ST	
CN52	51	63	78	B9	I	ST	
CN53	42	54	68	E9	I	ST	
CN54	43	55	69	E10	I	ST	
CN55	44	56	70	D11	I	ST	
CN56	45	57	71	C11	I	ST	
CN57	—	64	79	A9	I	ST	
CN58	60	76	93	A4	I	ST	
CN59	61	77	94	B4	I	ST	
CN60	62	78	98	119	I	ST	
CN61	63	79	99	A2	I	ST	
CN62	64	80	100	A1	I	ST	
CN63	1	1	3	D3	I	ST	
CN64	2	2	4	C1	I	ST	
CN65	3	3	5	D2	I	ST	
CN66	—	13	18	G1	I	ST	
CN67	—	14	19	G2	I	ST	
CN68	58	72	87	B6	I	ST	
CN69	59	73	88	A6	I	ST	
CN70	34	42	52	K11	I	ST	
CN71	33	41	51	K10	I	ST	
CN72	37	47	57	H10	I	ST	
CN73	36	46	56	J11	I	ST	
CN74	—	43	53	J10	I	ST	
CN75	—	—	40	K6	I	ST	
CN76	—	—	39	L6	I	ST	
CN77	—	75	90	A5	I	ST	
CN78	—	74	89	E6	I	ST	
CN79	—	—	96	C3	I	ST	
CN80	—	—	97	A3	I	ST	
CN81	—	—	95	C4	I	ST	
CN82	—	—	1	B2	I	ST	
CN83	—	44	54	H8	I	ST	
CN84	35	45	55	H9	I	ST	

图注: TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C$ ™ =  $I^2C/SMBus$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
COM0	63	79	99	A2	O	—	LCD 驱动器公共端输出。
COM1	62	78	98	B3	O	—	
COM2	61	77	94	B4	O	—	
COM3	60	76	93	A4	O	—	
COM4	59	73	88	A6	O	—	
COM5	23	29	34	L5	O	—	
COM6	22	28	33	L4	O	—	
COM7	21	27	32	K4	O	—	
CS1	45	57	71	C11	I/O	ST/TTL	并行主端口片选选通 1 (与 PMA14 共享)
CS2	44	56	70	D11	O	—	并行主端口片选选通 2 (与 PMA15 共享)
CTCMP	14	18	23	J2	I	ANA	CTMU 比较器 2 输入 (脉冲模式)。
CTED0	—	—	17	G3	I	ANA	CTMU 外部边沿输入。
CTED1	28	34	42	L7	I	ANA	
CTED2	27	33	41	J7	I	ANA	
CTED3	—	—	1	B2	I	ANA	
CTED4	1	1	3	D3	I	ANA	
CTED5	29	35	43	K7	I	ANA	
CTED6	30	36	44	L8	I	ANA	
CTED7	—	—	40	47	I	ANA	
CTED8	64	80	100	A1	I	ANA	
CTED9	63	79	99	A2	I	ANA	
CTED10	—	—	97	A3	I	ANA	
CTED11	—	—	95	C4	I	ANA	
CTED12	15	19	24	K1	I	ANA	
CTED13	14	18	23	J2	I	ANA	
CTPLS	29	35	43	K7	O	—	CTMU 脉冲输出
CVREF	23	29	34	L5	O	—	比较器参考电压输出。
CVREF+	16	20	25	K2	I	ANA	比较器 /A/D 参考电压 (低) 输入。
CVREF-	15	19	24	K1	I	ANA	比较器 /A/D 参考电压 (高) 输入。
INT0	35	45	55	H9	I	ST	外部中断输入 0。
LCDBIAS0	3	3	5	D2	I	ANA	LCD 驱动器电荷泵的偏置输入。
LCDBIAS1	2	2	4	C1	I	ANA	
LCDBIAS2	1	1	3	D3	I	ANA	
LCDBIAS3	17	21	26	L1	I	ANA	
LV DIN	64	80	100	A1	I	ANA	低压检测输入。
MCLR	7	9	13	F1	I	ST	主复位 (器件复位) 输入。将此引脚拉为低电平可导致器件复位。
OSCI	39	49	63	F9	I	ANA	主振荡器输入连接。
OSCO	40	50	64	F11	O	—	主振荡器输出连接。

图注：  
TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C^{\text{TM}}$  = I<sup>2</sup>C/SMBus 输入缓冲器

# PIC24FJ128GA310 系列

表 1-4: PIC24FJ128GA310 系列引脚排列说明 (续)

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
PGEC1	15	19	24	K1	I/O	ST	在线调试器 / 仿真器 /ICSP™ 编程时钟。
PGED1	16	20	25	K2	I/O	ST	在线调试器 / 仿真器 /ICSP 编程数据。
PGEC2	17	21	26	L1	I/O	ST	在线调试器 / 仿真器 /ICSP 编程时钟。
PGED2	18	22	27	J3	I/O	ST	在线调试器 / 仿真器 /ICSP 编程数据。
PGEC3	11	15	20	H1	I/O	ST	在线调试器 / 仿真器 /ICSP 编程时钟。
PGED3	12	16	21	H2	I/O	ST	在线调试器 / 仿真器 /ICSP 编程数据。
PMA0	30	36	44	L8	I/O	ST	并行主端口地址 bit 0 输入 (缓冲从模式) 和输出 (主模式)。
PMA1	29	35	43	K7	I/O	ST	并行主端口地址 bit 1 输入 (缓冲从模式) 和输出 (主模式)。
PMA2	8	10	14	F3	O	—	并行主端口地址 (bit <22:2>)。
PMA3	6	8	12	F2	O	—	
PMA4	5	7	11	F4	O	—	
PMA5	4	6	10	E3	O	—	
PMA6	16	24	29	K3	O	—	
PMA7	22	23	28	L2	O	—	
PMA8	32	40	50	L11	O	—	
PMA9	31	39	49	L10	O	—	
PMA10	28	34	42	L7	O	—	
PMA11	27	33	41	J7	O	—	
PMA12	24	30	35	J5	O	—	
PMA13	23	29	34	L5	O	—	
PMA14	45	57	71	C11	O	—	
PMA15	44	56	70	D11	O	—	
PMA16	—	—	95	C4	O	—	
PMA17	—	—	92	B5	O	—	
PMA18	—	—	40	K6	O	—	
PMA19	—	14	19	G2	O	—	
PMA20	—	—	59	G10	O	—	
PMA21	—	—	60	G11	O	—	
PMA22	—	52	66	E11	O	—	
PMACK1	50	62	77	A10	I	ST/TTL	并行主端口应答输入 1。
PMACK2	43	55	69	E10	I	ST/TTL	并行主端口应答输入 2。
PMBE0	51	63	78	B9	O	—	并行主端口字节使能选通 0。
PMBE1	—	53	67	E8	O	—	并行主端口字节使能选通 1。
PMCS1	—	13	18	G1	I/O	ST/TTL	并行主端口片选选通 1。
PMCS2	—	—	9	E1	O	—	并行主端口片选选通 2。

图注: TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $\text{I}^2\text{C}^\text{TM}$  =  $\text{I}^2\text{C}/\text{SMBus}$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
PMD0	60	76	93	A4	I/O	ST/TTL	并行主端口数据（非复用的主模式）或地址 / 数据（复用的主模式）。
PMD1	61	77	94	B4	I/O	ST/TTL	
PMD2	62	78	98	B3	I/O	ST/TTL	
PMD3	63	79	99	A2	I/O	ST/TTL	
PMD4	64	80	100	A1	I/O	ST/TTL	
PMD5	1	1	3	D3	I/O	ST/TTL	
PMD6	2	2	4	C1	I/O	ST/TTL	
PMD7	3	3	5	D2	I/O	ST/TTL	
PMD8	—	75	90	A5	I/O	ST/TTL	
PMD9	—	74	89	E6	I/O	ST/TTL	
PMD10	—	73	88	A6	I/O	ST/TTL	
PMD11	—	72	87	B6	I/O	ST/TTL	
PMD12	—	64	79	A9	I/O	ST/TTL	
PMD13	—	65	80	D8	I/O	ST/TTL	
PMD14	—	68	83	D7	I/O	ST/TTL	
PMD15	—	69	84	C7	I/O	ST/TTL	
PMRD	53	67	82	B8	O	—	并行主端口读选通。
PMWR	52	66	81	C8	O	—	并行主端口写选通。
RA0	—	—	17	G3	I/O	ST	PORTA 数字 I/O。
RA1	—	—	38	J6	I/O	ST	
RA2	—	—	58	H11	I/O	ST	
RA3	—	—	59	G10	I/O	ST	
RA4	—	—	60	G11	I/O	ST	
RA5	—	—	61	G9	I/O	ST	
RA6	—	—	91	C5	I/O	ST	
RA7	—	—	92	B5	I/O	ST	
RA9	—	23	28	L2	I/O	ST	
RA10	—	24	29	K3	I/O	ST	
RA14	—	52	66	E11	I/O	ST	
RA15	—	53	67	E8	I/O	ST	

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C^{\text{TM}}$  =  $I^2C/SMBus$  输入缓冲器

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
RB0	16	20	25	K2	I/O	ST	PORTB 数字 I/O。
RB1	15	19	24	K1	I/O	ST	
RB2	14	18	23	J2	I/O	ST	
RB3	13	17	22	J1	I/O	ST	
RB4	12	16	21	H2	I/O	ST	
RB5	11	15	20	H1	I/O	ST	
RB6	17	21	26	L1	I/O	ST	
RB7	18	22	27	J3	I/O	ST	
RB8	21	27	32	K4	I/O	ST	
RB9	22	28	33	L4	I/O	ST	
RB10	23	29	34	L5	I/O	ST	
RB11	24	30	35	J5	I/O	ST	
RB12	27	33	41	J7	I/O	ST	
RB13	28	34	42	L7	I/O	ST	
RB14	29	35	43	K7	I/O	ST	
RB15	30	36	44	L8	I/O	ST	
RC1	—	4	6	D1	I/O	ST	PORTC 数字 I/O。
RC2	—	—	7	E4	I/O	ST	
RC3	—	5	8	E2	I/O	ST	
RC4	—	—	9	E1	I/O	ST	
RC12	39	49	63	F9	I/O	ST	
RC13	47	59	73	C10	I	ST	
RC14	48	60	74	B11	I	ST	
RC15	40	50	64	F11	I/O	ST	
RD0	46	58	72	D9	I/O	ST	PORTD 数字 I/O。
RD1	49	61	76	A11	I/O	ST	
RD2	50	62	77	A10	I/O	ST	
RD3	51	63	78	B9	I/O	ST	
RD4	52	66	81	C8	I/O	ST	
RD5	53	67	82	B8	I/O	ST	
RD6	54	68	83	D7	I/O	ST	
RD7	55	69	84	C7	I/O	ST	
RD8	42	54	68	E9	I/O	ST	
RD9	43	55	69	E10	I/O	ST	
RD10	44	56	70	D11	I/O	ST	
RD11	45	57	71	C11	I/O	ST	
RD12	—	64	79	A9	I/O	ST	
RD13	—	65	80	D8	I/O	ST	
RD14	—	37	47	L9	I/O	ST	
RD15	—	38	48	K9	I/O	ST	

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C^{\text{TM}}$  =  $I^2C/\text{SMBus}$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
RE0	60	76	93	A4	I/O	ST	PORTE 数字 I/O。
RE1	61	77	94	B4	I/O	ST	
RE2	62	78	98	B3	I/O	ST	
RE3	63	79	99	A2	I/O	ST	
RE4	64	80	100	A1	I/O	ST	
RE5	1	1	3	D3	I/O	ST	
RE6	2	2	4	C1	I/O	ST	
RE7	3	3	5	D2	I/O	ST	
RE8	—	13	18	G1	I/O	ST	
RE9	—	14	19	G2	I/O	ST	
REF0	30	36	44	L8	O	—	参考时钟输出。
RF0	58	72	87	B6	I/O	ST	PORTF 数字 I/O。
RF1	59	73	88	A6	I/O	ST	
RF2	34	42	52	K11	I/O	ST	
RF3	33	41	51	K10	I/O	ST	
RF4	31	39	49	L10	I/O	ST	
RF5	32	40	50	L11	I/O	ST	
RF6	35	45	55	H9	I/O	ST	
RF7	—	44	54	H8	I/O	ST	
RF8	—	43	53	J10	I/O	ST	
RF12	—	—	40	K6	I/O	ST	PORTG 数字 I/O。
RF13	—	—	39	L6	I/O	ST	
RG0	—	75	90	A5	I/O	ST	
RG1	—	74	89	E6	I/O	ST	
RG2	37	47	57	H10	I/O	ST	
RG3	36	46	56	J11	I/O	ST	
RG6	4	6	10	E3	I/O	ST	
RG7	5	7	11	F4	I/O	ST	
RG8	6	8	12	F2	I/O	ST	
RG9	8	10	14	F3	I/O	ST	
RG12	—	—	96	C3	I/O	ST	
RG13	—	—	97	A3	I/O	ST	
RG14	—	—	95	C4	I/O	ST	
RG15	—	—	1	B2	I/O	ST	

图注：  
TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $\text{I}^2\text{C}^{\text{TM}}$  =  $\text{I}^2\text{C}/\text{SMBus}$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
RP0	16	20	25	K2	I/O	ST	可重映射外设（输入或输出）。
RP1	15	19	24	K1	I/O	ST	
RP2	42	54	68	E9	I/O	ST	
RP3	44	56	70	D11	I/O	ST	
RP4	43	55	69	E10	I/O	ST	
RP5	—	38	48	K9	I/O	ST	
RP6	17	21	26	L1	I/O	ST	
RP7	18	22	27	J3	I/O	ST	
RP8	21	27	32	K4	I/O	ST	
RP9	22	28	33	L4	I/O	ST	
RP10	31	39	49	L10	I/O	ST	
RP11	46	58	72	D9	I/O	ST	
RP12	45	57	71	C11	I/O	ST	
RP13	14	18	23	J2	I/O	ST	
RP14	29	35	43	K7	I/O	ST	
RP15	—	43	53	J10	I/O	ST	
RP16	33	41	51	K10	I/O	ST	
RP17	32	40	50	L11	I/O	ST	
RP18	11	15	20	H1	I/O	ST	
RP19	6	8	12	F2	I/O	ST	
RP20	53	67	82	B8	I/O	ST	
RP21	4	6	10	E3	I/O	ST	
RP22	51	63	78	B9	I/O	ST	
RP23	50	62	77	A10	I/O	ST	
RP24	49	61	76	A11	I/O	ST	
RP25	52	66	81	C8	I/O	ST	
RP26	5	7	11	F4	I/O	ST	
RP27	8	10	14	F3	I/O	ST	
RP28	12	16	21	H2	I/O	ST	
RP29	30	36	44	L8	I/O	ST	
RP30	34	42	52	K11	I/O	ST	
RP31	—	—	39	L6	I/O	ST	
RPI32	—	—	40	K6	I	ST	可重映射外设（仅输入）。
RPI33	—	13	18	G1	I	ST	
RPI34	—	14	19	G2	I	ST	
RPI35	—	53	67	E8	I	ST	
RPI36	—	52	66	E11	I	ST	
RPI37	48	60	74	B11	I	ST	
RPI38	—	4	6	D1	I	ST	
RPI39	—	—	7	E4	I	ST	
RPI40	—	5	8	E2	I	ST	
RPI41	—	—	9	E1	I	ST	
RPI42	—	64	79	A9	I	ST	
RPI43	—	37	47	L9	I	ST	

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $\text{I}^2\text{C}^{\text{TM}}$  =  $\text{I}^2\text{C}/\text{SMBus}$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
RTCC	42	54	68	E9	O	—	实时时钟闹钟 / 秒脉冲输出。
SCL1	37	47	57	H10	I/O	I <sup>2</sup> C	I <sup>2</sup> C1 同步串行时钟输入 / 输出。
SCL2	32	40	58	H11	I/O	I <sup>2</sup> C	I <sup>2</sup> C2 同步串行时钟输入 / 输出。
SCLKI	48	60	74	B11			
SDA1	36	46	56	J11	I/O	I <sup>2</sup> C	I <sup>2</sup> C1 数据输入 / 输出。
SDA2	31	39	59	G10	I/O	I <sup>2</sup> C	I <sup>2</sup> C2 数据输入 / 输出。
SEG0	4	6	10	E3	O	—	LCD 驱动器段输出。
SEG1	8	10	14	F3	O	—	
SEG2	11	15	20	H1	O	—	
SEG3	12	16	21	H2	O	—	
SEG4	13	17	22	J1	O	—	
SEG5	14	18	23	J2	O	—	
SEG6	15	19	24	K1	O	—	
SEG7	16	20	25	K2	O	—	
SEG8	29	35	43	K7	O	—	
SEG9	30	36	44	L8	O	—	
SEG10	31	39	49	L10	O	—	
SEG11	32	40	50	L11	O	—	
SEG12	33	41	51	K10	O	—	
SEG13	42	54	68	E9	O	—	
SEG14	43	55	69	E10	O	—	
SEG15	44	56	70	D11	O	—	
SEG16	45	57	71	C11	O	—	
SEG17	46	58	72	D9	O	—	
SEG18	27	33	41	J7	O	—	
SEG19	28	34	42	L7	O	—	
SEG20	49	61	76	A11	O	—	
SEG21	50	62	77	A10	O	—	
SEG22	51	63	78	B9	O	—	
SEG23	52	66	81	C8	O	—	
SEG24	53	67	82	B8	O	—	
SEG25	54	68	83	D7	O	—	
SEG26	55	69	84	C7	O	—	
SEG27	58	72	87	B6	O	—	
SEG28	37	47	57	H10	O	—	
SEG29	23	29	34	L5	O	—	
SEG30	22	28	33	L4	O	—	
SEG31	21	27	32	K4	O	—	
SEG32	—	4	6	D1	O	—	
SEG33	—	5	8	E2	O	—	
SEG34	—	13	18	G1	O	—	

图注：  
TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
I<sup>2</sup>C<sup>TM</sup> = I<sup>2</sup>C/SMBus 输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
SEG35	—	14	19	G2	O	—	LCD 驱动器段输出。
SEG36	—	23	28	L2	O	—	
SEG37	—	24	29	K3	O	—	
SEG38	—	37	47	L9	O	—	
SEG39	—	38	48	K9	O	—	
SEG40	—	42	52	K11	O	—	
SEG41	—	43	53	J10	O	—	
SEG42	—	52	66	E11	O	—	
SEG43	—	53	67	E8	O	—	
SEG44	—	64	79	A9	O	—	
SEG45	—	65	80	D8	O	—	
SEG46	—	74	89	E6	O	—	
SEG47	36	46	56	J11	O	—	
SEG48	59	73	88	A6	O	—	
SEG49	—	—	17	G3	O	—	
SEG50	—	75	90	A5	O	—	
SEG51	—	—	1	B2	O	—	
SEG52	—	—	7	E4	O	—	
SEG53	—	—	9	E1	O	—	
SEG54	—	—	39	L6	O	—	
SEG55	—	—	40	K6	O	—	
SEG56	—	—	58	H11	O	—	
SEG57	—	—	59	G10	O	—	
SEG58	—	—	91	C5	O	—	
SEG59	—	—	92	B5	O	—	
SEG60	—	—	95	C4	O	—	
SEG61	—	—	96	C3	O	—	
SEG62	—	—	97	A3	O	—	
SEG63	—	—	100	A1	O	—	
SOSCI	47	59	73	C10	I	ANA	辅助振荡器 /Timer1 时钟输入。
SOSCO	48	60	74	B11	O	ANA	辅助振荡器 /Timer1 时钟输出。
T1CK	22	28	33	L4	I	ST	Timer1 时钟。
TCK	27	33	38	J6	I	ST	JTAG 测试时钟 / 编程时钟输入。
TDI	28	34	60	G11	I	ST	JTAG 测试数据 / 编程数据输入。
TDO	24	14	61	G9	O	—	JTAG 测试数据输出。
TMS	23	13	17	G3	I	ST	JTAG 测试模式选择输入。

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $I^2C^{\text{TM}}$  =  $I^2C/\text{SMBus}$  输入缓冲器

# PIC24FJ128GA310 系列

表 1-4： PIC24FJ128GA310 系列引脚排列说明（续）

引脚功能	引脚数/网格定位符				I/O	输入缓冲器	说明
	64 引脚 TQFP	80 引脚 TQFP	100 引脚 TQFP	121 引脚 BGA			
VBAT	57	71	86	A7	P	—	备用电池。
VCAP	56	70	85	B7	P	—	外部滤波电容连接（使能稳压器）。
VDD	10, 26, 38	12, 32, 48	2, 16, 37, 46, 62	C2, F8, G5, H6, K8	P	—	外设数字逻辑和 I/O 引脚的正电源。
VLCAP1	5	7	11	F4	I	ANA	LCD 驱动器电荷泵电容输入。
VLCAP2	6	8	12	F2	I	ANA	
VREF+	—	24	29	K3	I	ANA	比较器 /A/D 参考电压（低）输入（默认）。
VREF-	—	23	28	L2	I	ANA	比较器 /A/D 参考电压（高）输入（默认）。
Vss	9, 25, 41	11, 31, 51	15, 36, 45, 65, 75	B10, F5, F10, G6, G7	P	—	逻辑和 I/O 引脚的参考地。

图注： TTL = TTL 输入缓冲器  
ANA = 模拟电平输入/输出

ST = 施密特触发器输入缓冲器  
 $\text{I}^2\text{C}^{\text{TM}}$  =  $\text{I}^2\text{C}/\text{SMBus}$  输入缓冲器

## 2.0 16 位单片机入门指南

### 2.1 基本连接要求

在开始使用 PIC24FJ128GA310 系列 16 位单片机进行开发之前，需要注意最低限度的器件引脚连接要求。

必须始终连接以下引脚：

- 所有 VDD 和 VSS 引脚（见第 2.2 节“电源引脚”）
- 所有 AVDD 和 AVSS 引脚，不论是否使用模拟器件功能（见第 2.2 节“电源引脚”）
- MCLR 引脚（见第 2.3 节“主复位（MCLR）引脚”）
- VCAP 引脚（见第 2.4 节“稳压器引脚（VCAP）”）

如果在最终应用中使用了以下引脚，则也必须连接它们：

- 用于在线串行编程（ICSP™）和调试的 PGECx/PGEDx 引脚（见第 2.5 节“ICSP 引脚”）
- OSCI 和 OSCO 引脚，使用外部振荡器源时（见第 2.6 节“外部振荡器引脚”）

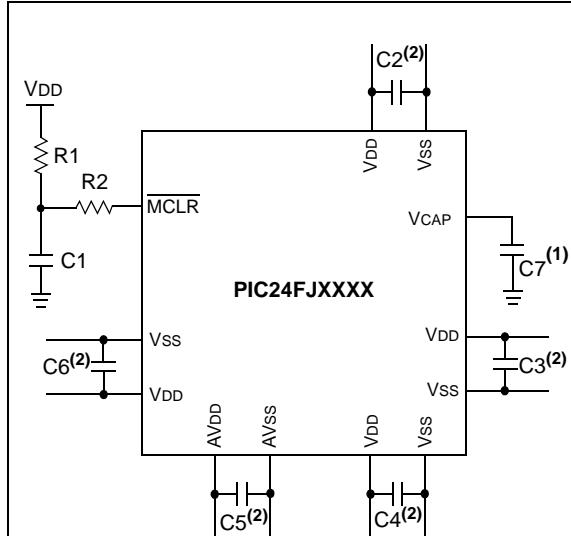
此外，可能还需要连接以下引脚：

- 实现模拟模块的外部参考电压时使用的 VREF+/VREF- 引脚

**注：**不论是否使用任何模拟模块，都必须始终连接 AVDD 和 AVSS 引脚。

图 2-1 中显示了最低限度的连接要求。

图 2-1：建议的最低限度连接



关键（所有值均为建议值）：

C1 至 C6: 0.1  $\mu$ F, 20V 陶瓷电容

C7: 10  $\mu$ F、6.3V 或更大的钽电容或陶瓷电容

R1: 10 k $\Omega$

R2: 100 $\Omega$  到 470 $\Omega$

**注 1：**有关为 Vcap 选择适当电容的详细信息，请参见第 2.4 节“稳压器引脚（VCAP）”。

**2:** 此处所示示例针对的是具有 5 个 VDD/VSS 和 AVDD/AVSS 引脚对的 PIC24F 器件。其他器件的引脚对可能增多或减少，请相应地调整去耦电容的数量。

## 2.2 电源引脚

### 2.2.1 去耦电容

需要在每对电源引脚（如VDD与VSS以及AVDD与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  $0.1 \mu\text{F}$  ( $100 \text{nF}$ )、 $10\text{-}20\text{V}$  的电容。应使用谐振频率在  $200 \text{MHz}$  及更高范围内的低 ESR 电容。建议使用陶瓷电容。
- 在印制电路板上的放置：**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。如果空间受到限制，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不超出  $0.25$  英寸 ( $6$  毫米)。
- 高频噪声处理：**如果电路板遇到高频噪声（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  $0.001 \mu\text{F}$  至  $0.01 \mu\text{F}$  之间。请将第二个电容放置在靠近每个主去耦电容的位置。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两个数量级相差数十倍的电容（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。

### 2.2.2 槽路电容

在电源走线长度大于  $6$  英寸的电路板上，建议在包含单片机的集成电路中使用槽路电容，以提供本地电源。槽路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的器件的最大电流确定。也就是说，选择的槽路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  $47 \mu\text{F}$ 。

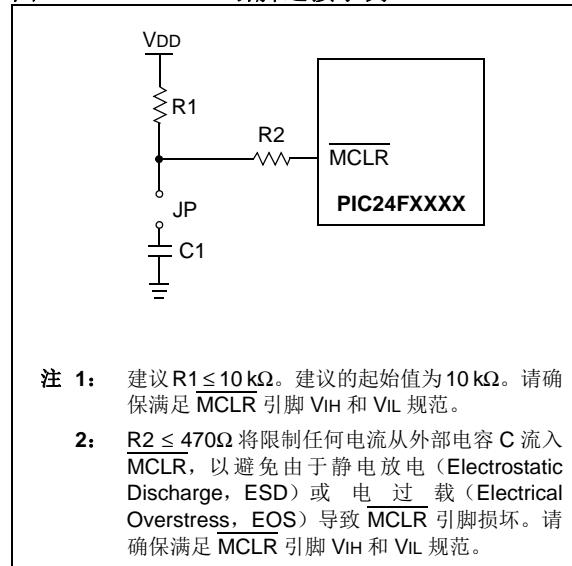
## 2.3 主复位 (MCLR) 引脚

MCLR 引脚提供两种特定的器件功能：器件复位及器件编程与调试。如果最终应用中不需要进行编程和调试，则只需直接连接 VDD 即可。添加其他元件有助于提高应用抵抗由于电压骤降导致意外复位的能力。图 2-1 给出了一种典型配置。根据应用的需求，还可以实现其他电路设计。

在编程和调试过程中，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 MCLR 引脚。因此，特定电平 (VIH 和 Vil) 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 R1 和 C1 的具体值。例如，在编程和调试操作期间，建议通过使用跳线将电容 C1 与 MCLR 引脚隔离（图 2-2）。对于正常的运行时操作，可以将跳线放回原处。

与 MCLR 引脚关联的所有元件都应放置在距离该引脚  $0.25$  英寸 ( $6$  毫米) 的范围内。

图 2-2: MCLR 引脚连接示例



## 2.4 稳压器引脚 (VCAP)

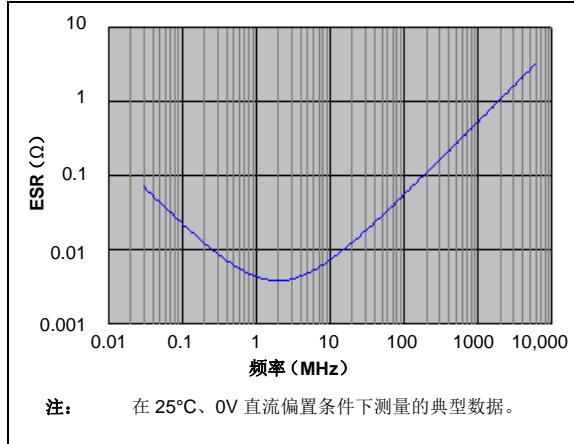
VCAP 引脚需要一个低 ESR ( $< 5\Omega$ ) 电容来稳定片内稳压器的输出电压。VCAP 引脚一定不能与 VDD 连接，并且必须使用  $10 \mu\text{F}$  的电容接地。电容类型可以是陶瓷电容或者钽电容。[表 2-1](#) 列出了一些适用电容的示例。同等规格的电容都可以使用。

此电容应靠近 VCAP 放置。建议走线长度不要超出 0.25 英寸（6 毫米）。更多信息，请参见[第 32.0 节“电气特性”](#)。

设计人员可以使用[图 2-3](#) 来计算备选电容的 ESR。

关于连接和使用片内稳压器的详细信息，请参见[第 29.2 节“片内稳压器”](#)。

**图 2-3：建议的 VCAP 的频率与 ESR 性能关系**



**表 2-1：适用的等效电容**

制造商	部件编号	标称电容	基本容差	额定电压	温度范围
TDK	C3216X7R1C106K	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $125^\circ\text{C}$
TDK	C3216X5R1C106K	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $85^\circ\text{C}$
Panasonic	ECJ-3YX1C106K	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $125^\circ\text{C}$
Panasonic	ECJ-4YB1C106K	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $85^\circ\text{C}$
Murata	GRM32DR71C106KA01L	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $125^\circ\text{C}$
Murata	GRM31CR61C106KC31L	$10 \mu\text{F}$	$\pm 10\%$	16V	-55 至 $85^\circ\text{C}$

## 2.4.1 关于陶瓷电容的注意事项

近年来，大容量、低电压的表面贴装式陶瓷电容极具成本效益，其容值最高可达数十微法。体积小、低 ESR 及其他一些属性，使得陶瓷电容对于许多类型的应用颇具吸引力。

陶瓷电容适用于本系列单片机的内部稳压器。但在选择具体电容时，仍需注意确保电容在应用的目标工作范围内，始终能保持足够的电容值。

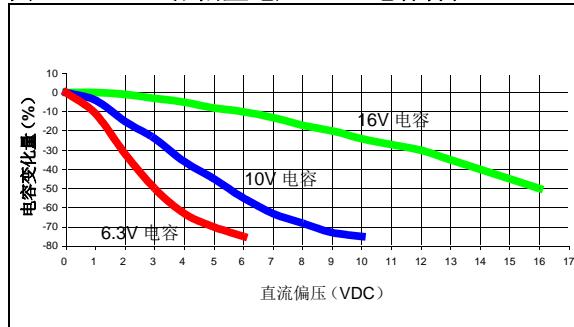
典型的低成本  $10\ \mu F$  陶瓷电容有 X5R、X7R 和 Y5V 等介电特性（也存在其他型号，不过不太常见）。此类电容的初始公差通常指定为  $\pm 10\%$  至  $\pm 20\%$ （X5R 和 X7R），或  $-20\% + 80\%$ （Y5V）。然而，这些电容在应用中提供的有效电容还会根据其他因素产生变化，比如施加的直流偏压和温度。因此，电容在电路中的总公差会比初始公差规范指定的范围宽很多。

X5R 和 X7R 电容通常能表现良好的温度稳定性（例如，在很宽的温度范围内，电容值变化量在  $\pm 15\%$  之间。具体的参数请参见制造商提供的数据手册）。但是，Y5V 电容在温度变化时的电容公差通常很糟糕，为  $+22\% - 82\%$ 。因此，标称值为  $10\ \mu F$  的 Y5V 型电容可能无法提供足够的总电容来满足内部稳压器稳定和瞬态响应的最低要求。所以，当应用必须在较宽温度范围内工作时，建议不要将 Y5V 电容用于内部稳压器。

除了温度的影响，大值陶瓷电容的有效电容值根据施加给电容的直流电压，会有很大的变化。直流偏压的效果非常明显，但却常常被人们忽视，或常常没有在文档中说明。

X7R 型电容的典型直流偏压和电容的关系图见图 2-4。

图 2-4： 直流偏置电压 —— 电容特性



在选择要用于内部稳压器的陶瓷电容时，建议选择额定电压较高的电容，这样工作电压占电容最大电压的比例就比较小。例如，对于内核电压为 2.5V 或 1.8V 的应用，选择额定电压为 16V 的陶瓷电容。表 2-1 给出了建议的电容。

## 2.5 ICSP 引脚

PGECx 和 PGEDx 引脚用于进行在线串行编程 (ICSP) 和调试。建议尽可能减小 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，电阻值介于几十欧姆的范围，不要超出  $100\Omega$ 。

建议不要在 PGECx 和 PGEDx 引脚上连接上拉电阻、串联二极管和电容，因为它们会影响与器件的编程器 / 调试器通信。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取出。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 ( $V_{IH}$ ) 和输入低电压 ( $V_{IL}$ ) 要求的信息。

对于器件仿真，请确保烧写到器件中的“通信通道选择”（即，PGECx/PGEDx 引脚）符合 ICSP 到 Microchip 调试器 / 仿真器工具的物理连接。

关于可用 Microchip 开发工具连接要求的更多信息，请参见第 30.0 节“开发支持”。

## 2.6 外部振荡器引脚

许多单片机都有至少两个振荡器可供选择：高频主振荡器和低频辅助振荡器（详细信息请参见第 9.0 节“振荡器配置”）。

振荡器电路与器件应放置在电路板的同一层。请将振荡器电路放置在靠近相应振荡器引脚的位置，电路元件与引脚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自身，位于电路板的同一层。

请在振荡器电路周围使用接地灌铜区，以将其与周围电路隔离。接地灌铜区应与 MCU 地直接连接。不要在接地灌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。

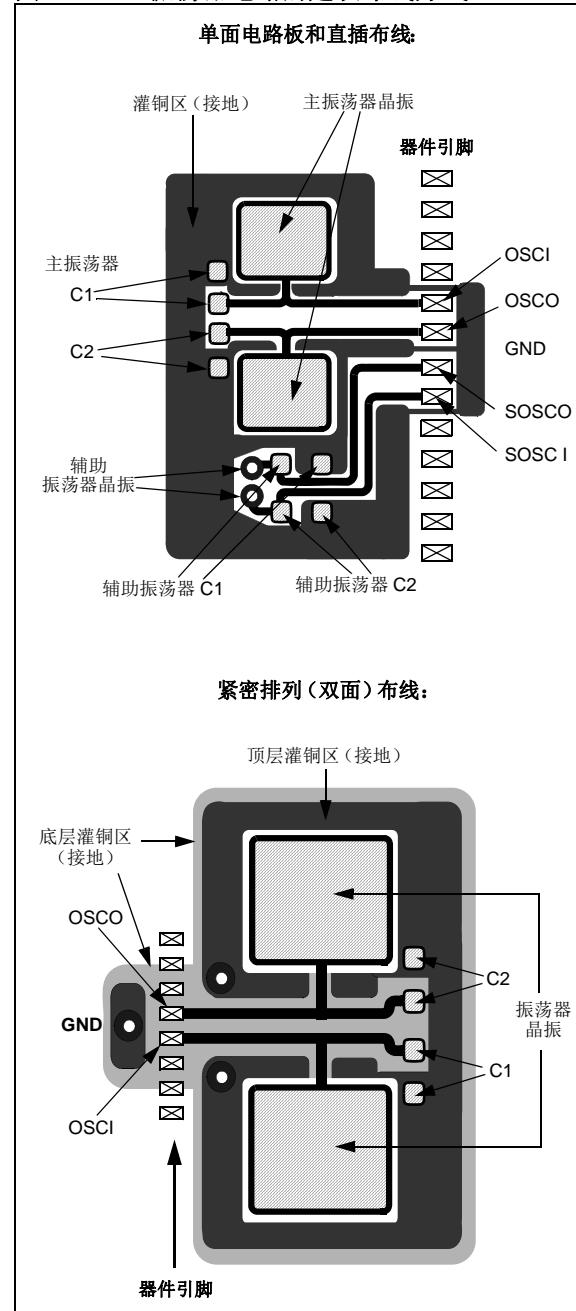
图 2-5 给出了一些布线建议。直插式封装可以采用可完全容纳振荡器引脚的单面布线来处理。对于引脚排列紧密的器件，单面布局则可能无法始终完全地容纳所有引脚和元件。一种适合的解决方案是将含有保护走线的部分连接到反面的接地层。在所有情形中，保护走线都必须回接到地。

在规划应用的走线和 I/O 分配时，需要确保相邻端口引脚和其他邻近振荡器的信号是无害的（如，无高频、无短暂上升和下降时间，以及无其他类似噪声）。

关于振荡器电路的其他信息和设计指南，请参见 Microchip 公司网站 ([www.microchip.com](http://www.microchip.com)) 上提供的以下应用笔记：

- AN826, “Crystal Oscillator Basics and Crystal Selection for rfPIC™ and PICmicro® Devices”
- AN849, “Basic PICmicro® Oscillator Design”
- AN943, “Practical PICmicro® Oscillator Analysis and Design”
- AN949, “Making Your Oscillator Work”

图 2-5：振荡器电路的建议布线方式



## 2.7 ICSP 工作期间模拟和数字引脚配置

如果将兼容 ICSP 的仿真器选择为调试器，它会自动初始化所有 A/D 输入引脚 (ANx) 为“数字”引脚。根据具体器件，可通过将 ADnPCFG 寄存器中的所有位置 1 或清零 ANSx 寄存器的所有位来实现。

所有 PIC24F 器件将具有一个或多个 ADnPCFG 寄存器或几个 ANSx 寄存器（每个端口一个），但没有器件同时具有这两者。更多详细信息，请参见[第 11.2 节“配置模拟端口引脚 \(ANSx\)”](#)。

用户应用固件不得更改这些寄存器中与 A/D 引脚相对应的由仿真器初始化的位；否则将导致调试器和器件之间发生通信错误。

如果在调试会话期间应用需要使用某些 A/D 引脚作为模拟输入引脚，那么用户应用必须在 A/D 模块初始化期间修改相应位，具体如下：

- 对于具有 ADnPCFG 寄存器的器件，清零引脚相对应的位以将其配置为模拟引脚。在任何时候都不要更改任何其他位，特别是与 PGECx/PGEDx 对对应的位。
- 对于具有 ANSx 寄存器的器件，置 1 引脚相对应的位将其配置为模拟引脚。在任何时候都不要更改任何其他位，特别是与 PGECx/PGEDx 对对应的位。

当 Microchip 调试器 / 仿真器用作编程器时，用户应用固件必须正确配置 ADnPCFG 或 ANSx 寄存器。仅在调试器操作期间自动初始化此寄存器。如果未能正确配置该寄存器，将导致所有 A/D 引脚被识别为模拟输入引脚，以致端口值被读为逻辑 0，从而可能影响用户应用的功能。

## 2.8 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，将未用引脚通过一个  $1\text{ k}\Omega$  至  $10\text{ k}\Omega$  的电阻与 V<sub>SS</sub> 连接，并将输出驱动为逻辑低电平。

## 3.0 CPU

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 44 章“具有扩展数据空间 (EDS) 的 CPU”(DS39732A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

PIC24F CPU 采用 16 位 (数据) 的改进型哈佛架构, 具有增强的指令集和带有长度可变的操作码字段的 24 位指令字。程序计数器 (Program Counter, PC) 为 23 位宽, 可以寻址最大 4M 指令字的用户程序存储空间。单周期指令预取机制可帮助维持吞吐量, 并使指令的执行具有预测性。除了改变程序流的指令、双字传送 (MOV.D) 指令和表指令以外, 所有指令都在单个周期内执行。使用 REPEAT 指令可以支持无开销的程序循环结构, 在任何时候都可被中断。

PIC24F 器件的编程模型中有 16 个 16 位的工作寄存器。每个工作寄存器都可以充当数据、地址或地址偏移量寄存器。第 16 个工作寄存器 (W15) 用作软件堆栈指针, 用于中断和调用。

可对数据空间的低 32 KB 进行线性访问。数据空间的高 32 KB 是指扩展数据 RAM、EPMP 存储空间或程序存储器可映射到的扩展数据空间。

指令集架构 (Instruction Set Architecture, ISA) 与 PIC18 的相比有了显著的提升, 并保持了一定程度的向下兼容性。该架构直接或通过简单的宏支持所有的 PIC18 指令和寻址模式。对编译器执行效率的需求促使了对 ISA 的许多改进。

内核支持固有 (无操作数) 寻址、相对寻址、立即数寻址和存储器直接寻址及其他三组寻址模式。所有模式都支持寄存器直接寻址和各种寄存器间接寻址模式。每组都提供了最多 7 种寻址模式。指令根据其功能要求, 与预定义的寻址模式相关。

对于大多数指令, 内核能在每个指令周期内执行一次数据 (或程序数据) 存储器读操作、一次工作寄存器 (数据) 读操作、一次数据存储器写操作和一次程序 (指令) 存储器读操作。因此, 可以支持 3 个操作数的指令, 使 3 操作数的运算 (即,  $A + B = C$ ) 能在单周期内执行。

包含了一个高速 17 位  $\times$  17 位乘法器, 显著提高了内核的运算能力和吞吐量。此乘法器支持有符号、无符号和混合模式的 16 位  $\times$  16 位或 8 位  $\times$  8 位整数乘法。所有的乘法指令都在单周期内执行。

已对 16 位 ALU 进行了改进使其具备一个支持整数除法的硬件, 该硬件支持迭代的不可撤销的除法算法。它可以和 REPEAT 指令循环机制和迭代除法指令 (可选) 一起工作, 支持 32 位 (或 16 位) 除以 16 位有符号和无符号整数的除法运算。所有除法运算都需要 19 个周期来完成, 但是可在任何周期边界被中断。

PIC24F 具有向量异常机制, 具有最多 8 个不可屏蔽的陷阱源和 118 个中断源。可以为每个中断源分配 7 个优先级之一。

CPU 的框图如图 3-1 所示。

### 3.1 编程模型

图 3-2 中所示为 PIC24F 的编程模型。编程模型中的所有寄存器都是存储器映射的, 并且可以由指令直接控制。表 3-1 中提供了对每个寄存器的描述。所有与编程模型相关联的寄存器都是存储器映射的。

# PIC24FJ128GA310 系列

图 3-1: PIC24F CPU 内核框图

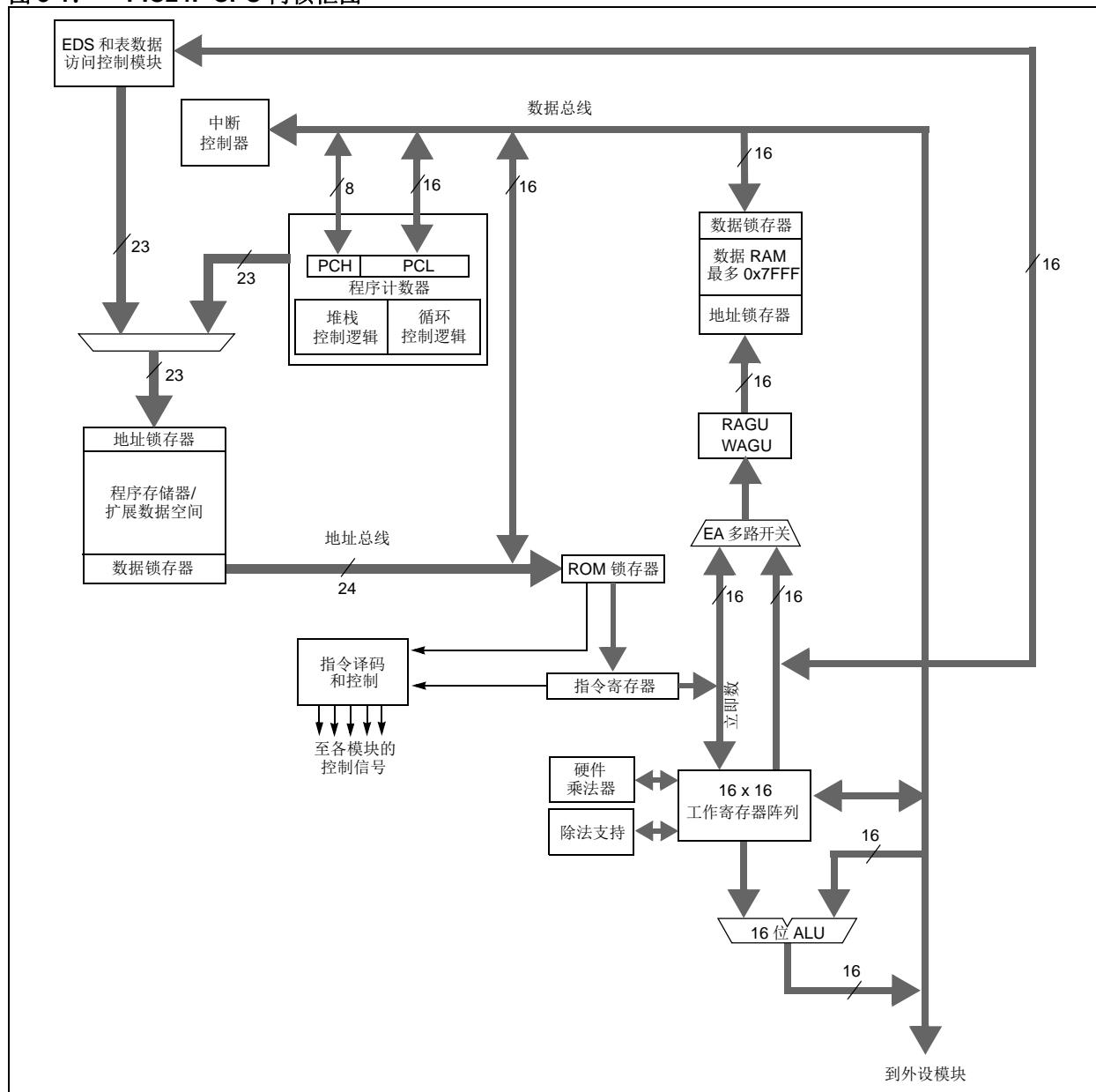
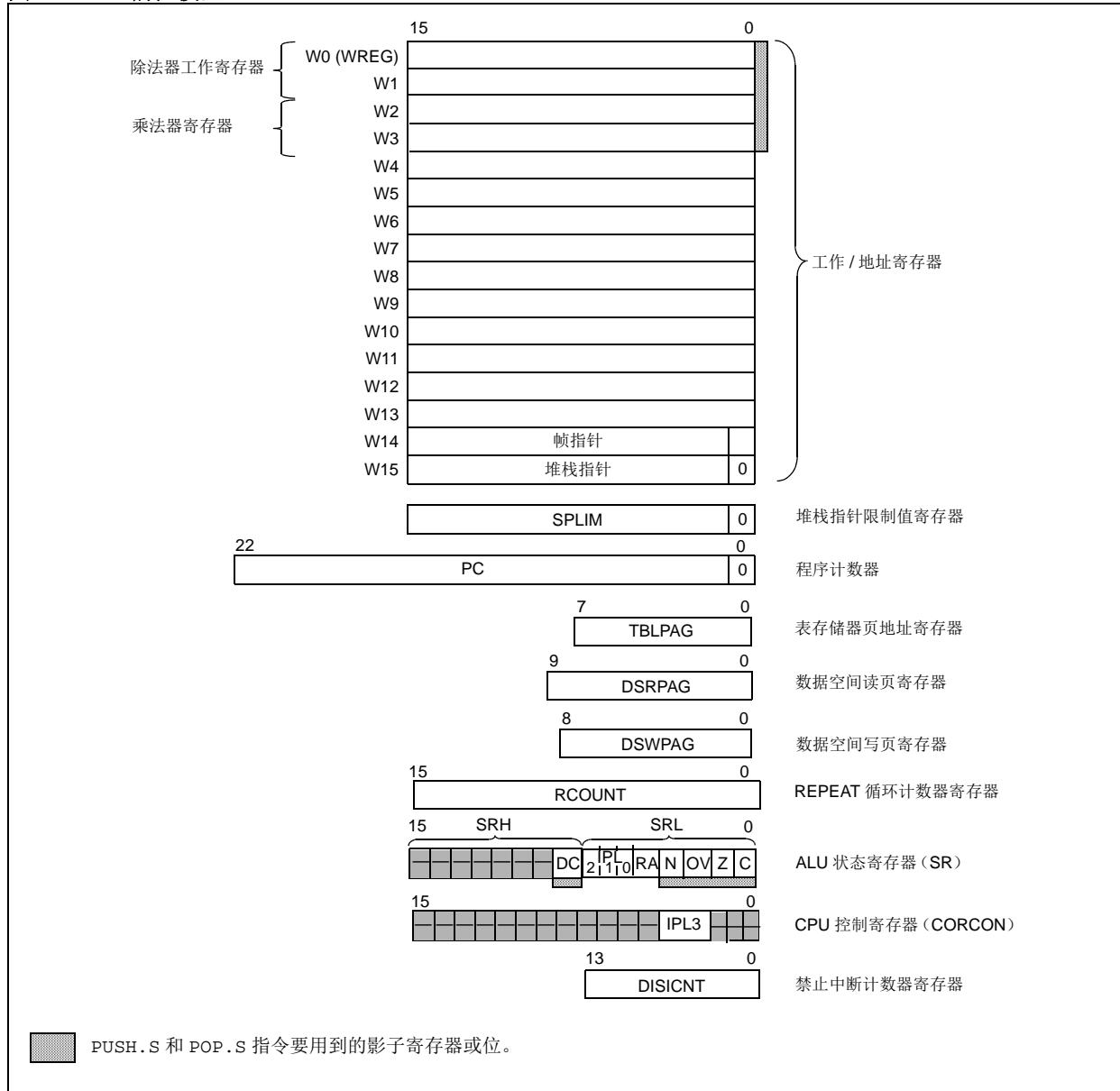


表 3-1: CPU 内核寄存器

寄存器名称	说明
W0 到 W15	工作寄存器阵列
PC	23 位程序计数器
SR	ALU 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
RCOUNT	Repeat 循环计数器寄存器
CORCON	CPU 控制寄存器
DISICNT	禁止中断计数器寄存器
DSRPAG	数据空间读页寄存器
DSWPAG	数据空间写页寄存器

图 3-2： 编程模型



# PIC24FJ128GA310 系列

## 3.2 CPU 控制寄存器

寄存器 3-1: SR: ALU 状态寄存器

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

R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R/W-0 <sup>(1)</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:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-9 未实现: 读为 0

bit 8 DC: 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）发生了向高位的进位  
0 = 结果的第 4 个低位或第 8 个低位未发生向高位的进位

bit 7-5 IPL<2:0>: 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 RA: REPEAT 循环活动位

1 = 正在进行 REPEAT 循环  
0 = 未进行 REPEAT 循环

bit 3 N: ALU 负标志位

1 = 结果为负  
0 = 结果为非负（零或正值）

bit 2 OV: ALU 溢出标志位

1 = 在本次算术运算中有符号（二进制补码）运算发生了溢出  
0 = 未发生溢出

bit 1 Z: 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经将该位清零（即运算结果非零）

bit 0 C: ALU 进位/借位标志位

1 = 结果的最高位（MSb）发生了进位  
0 = 结果的最高位未发生进位

注 1: 当 NSTDIS (INTCON1<15>) = 1 时, IPL 状态位只读。

2: IPL 状态位与 IPL3 位 (CORCON<3>) 共同决定 CPU 的中断优先级 (Interrupt Priority Level, IPL)。如果 IPL3 = 1, 则括号中的值表示 IPL。

## 寄存器 3-2: CORCON: CPU 内核控制寄存器

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/C-0	R-1	U-0	U-0
—	—	—	—	IPL3 <sup>(1)</sup>	r	—	—
bit 7							bit 0

图注:	C = 可清零位	r = 保留位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
		x = 未知

- bit 15-4 未实现: 读为 0  
 bit 3 IPL3: CPU 中断优先级状态位<sup>(1)</sup>  
     1 = CPU 中断优先级大于 7  
     0= CPU 中断优先级为 7 或更小  
 bit 2 保留: 读为 1  
 bit 1-0 未实现: 读为 0

注 1: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 共同决定 CPU 中断优先级; 关于位描述, 请参见[寄存器 3-1](#)。

### 3.3 算术逻辑单元 (ALU)

PIC24F ALU 为 16 位宽，能进行加法、减法、移位和逻辑运算。除非另外声明，算术运算一般是以二进制补码方式进行的。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和半借位位。

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

PIC24F CPU 融入了对乘法和除法的硬件支持。它带有专用的硬件乘法器以及支持 16 位除数除法的硬件。

#### 3.3.1 乘法器

ALU 包含一个高速的 17 位  $\times$  17 位乘法器。它支持几种乘法模式下的无符号、有符号或混合符号运算：

1. 16 位  $\times$  16 位有符号数
2. 16 位  $\times$  16 位无符号数
3. 16 位有符号数  $\times$  5 位（立即数）无符号数
4. 16 位无符号数  $\times$  16 位无符号数
5. 16 位无符号数  $\times$  5 位（立即数）无符号数
6. 16 位无符号数  $\times$  16 位有符号数
7. 8 位无符号数  $\times$  8 位无符号数

表 3-2： 使用单位和多位移位操作的指令

指令	说明
ASR	将源寄存器算术右移一位或多位。
SL	将源寄存器左移一位或多位。
LSR	将源寄存器逻辑右移一位或多位。

#### 3.3.2 除法器

除法模块支持具有下列数据长度的有符号和无符号整数除法运算：

1. 32 位有符号数 / 16 位有符号数
2. 32 位无符号数 / 16 位无符号数
3. 16 位有符号数 / 16 位有符号数
4. 16 位无符号数 / 16 位无符号数

所有除法指令的商都被放在 W0 中，余数放在 W1 中。16 位有符号和无符号 DIV 指令可为 16 位除数指定任一 W 寄存器 (Wn)，为 32 位被除数指定任意两个连续的 W 寄存器 (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。

#### 3.3.3 多位移位支持

PIC24F ALU 支持单位和单周期多位算术和逻辑移位操作。由一个移位寄存器电路实现多位移位，在单个周期内最多可将数据算术右移或左移 15 位。所有的多位移位指令仅支持源操作数和目标结果的寄存器直接寻址模式。

在下面的表 3-2 中汇总了所有使用移位操作的指令。

## 4.0 存储器构成

作为哈佛架构器件，PIC24F 单片机具有独立的程序和数据存储空间以及独立的程序和数据总线。此架构还允许在代码执行过程中直接通过数据空间访问程序空间。

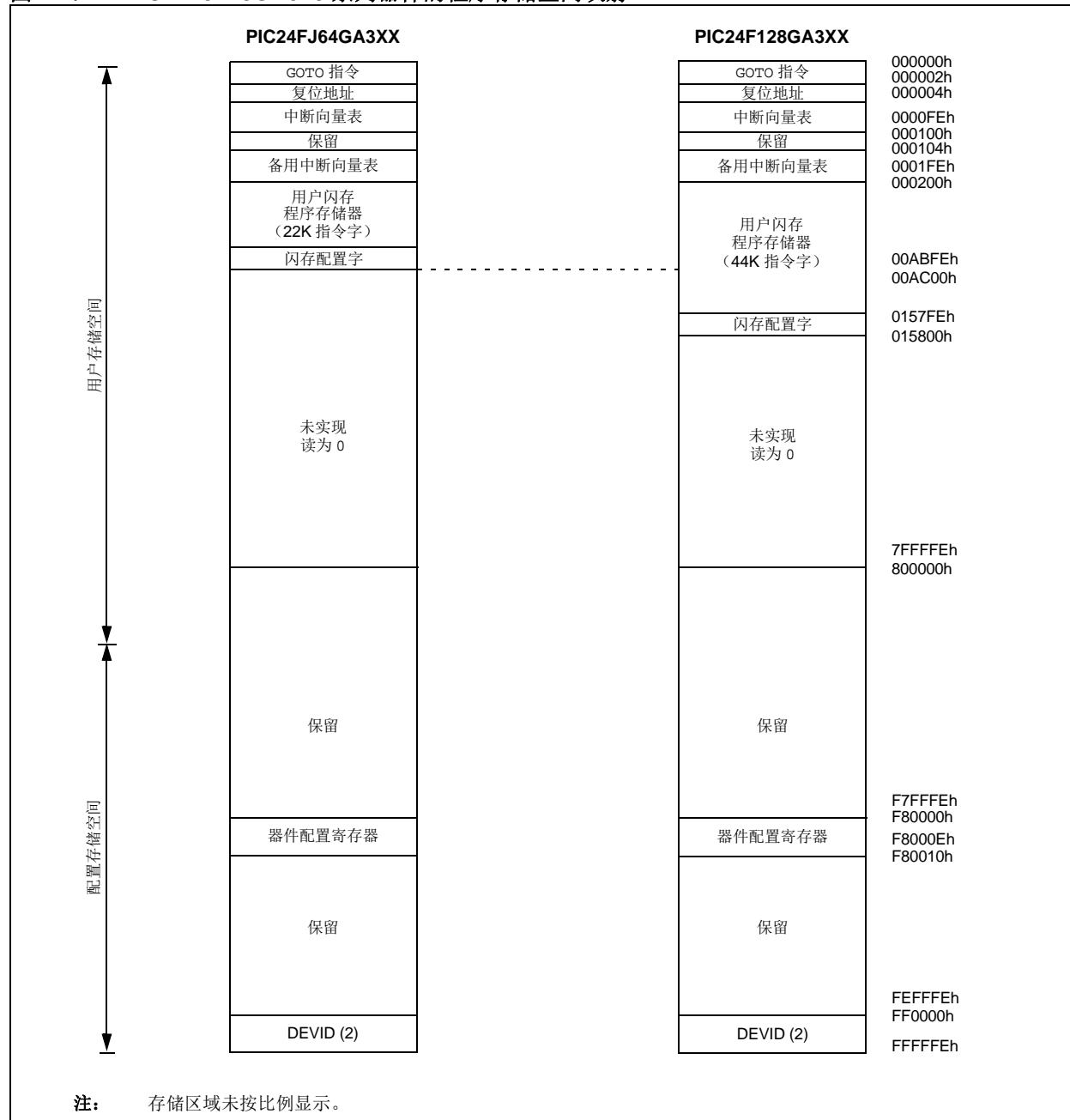
## 4.1 程序存储空间

PIC24FJ128GA310 系列器件的程序地址存储空间可存储 4M 指令字。可通过由程序执行过程中 23 位程序计数器 (PC) 或第 4.3 节“[程序存储空间与数据存储空间的接口](#)”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。

用户只能访问程序存储空间的低半地址部分（地址范围为 000000h 至 7FFFFFh）。使用 TBLRD/TBLWT 指令时，情况有所不同，这两条指令使用 TBLPAG<7> 位以允许访问配置存储空间中的配置位和器件 ID。

[图 4-1](#) 给出了 PIC24FJ128GA310 系列器件的存储器映射情况。

[图 4-1： PIC24FJ128GA310 系列器件的程序存储空间映射](#)



# PIC24FJ128GA310 系列

## 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部分没有实现。低位字的地址总是偶数，而高位字的地址为奇数（图 4-2）。

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。这种对齐方式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。

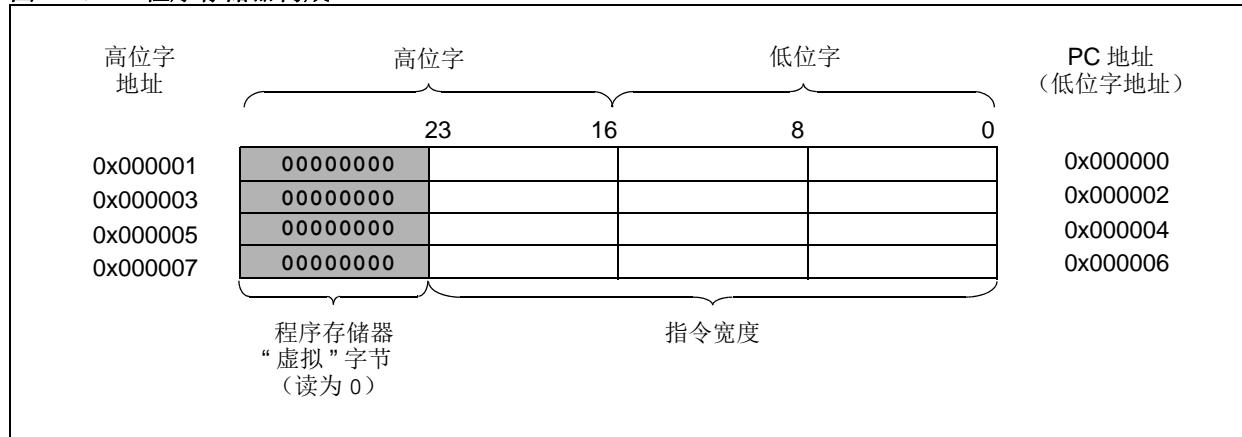
## 4.1.2 存储器硬编码向量

所有 PIC24F 器件中从 000000h 到 000200h 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 的默认值重新定位到代码实际开始处。用户可在 000000h 地址编写一条 GOTO 指令以将代码的实际起始处定义为 000002h。

PIC24F 器件也具有 2 个中断向量表，地址范围分别为 000004h 至 0000FFh 和 000100h 至 0001FFh。这两个中断向量表允许使用不同的 ISR 处理每个器件中断源。

**第 8.1 节“中断向量表”** 提供了有关中断向量表的更多详细信息。

图 4-2：程序存储器构成



## 4.1.3 闪存配置字

在 PIC24FJ128GA310 系列器件中，保留片上程序存储器的高 4 个字来存储配置信息。器件复位时，将配置信息复制到相应的配置寄存器中。PIC24FJ128GA310 系列器件的闪存配置字的地址如表 4-1 所示。图 4-1 给出了闪存配置字以及其他存储器向量在存储器映射中的位置。

程序存储器中的配置字为紧凑的格式。实际配置位被映射到配置存储空间的几个不同的寄存器中。它们在闪存配置字中的顺序并不反映它们在配置空间中的相应顺序。第 29.0 节“特殊功能”给出了器件配置字的更多详细信息。

表 4-1：PIC24FJ128GA310 系列器件的闪存配置字

器件	程序 存储器 (字)	配置字地址
PIC24FJ64GA3XX	22,016	00ABF8h:00ABFEh
PIC24FJ128GA3XX	44,032	0157F8h:0157FEh

## 4.2 数据存储空间

**注:** 本数据手册总结了该组PIC24F器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息,请参见《PIC24F系列参考手册》中的第45章“带有扩展数据空间(EDS)的数据存储器”(DS39733A\_CN)。本数据手册中的信息取代了FRM中的信息。

PIC24F内核具有一个16位宽的数据存储空间,可将其作为一个线性空间寻址。使用两个地址发生单元(Address Generation Unit, AGU)对数据空间执行读写操作。图4-3给出了数据存储空间映射。

数据存储空间中的16位宽数据地址指向数据空间(Data Space, DS)中的字节。这种构成方式使得数据空间的地址范围为64 KB或32K字。低半部分(0000h到7FFFh)供实现的(片上)存储器地址使用。

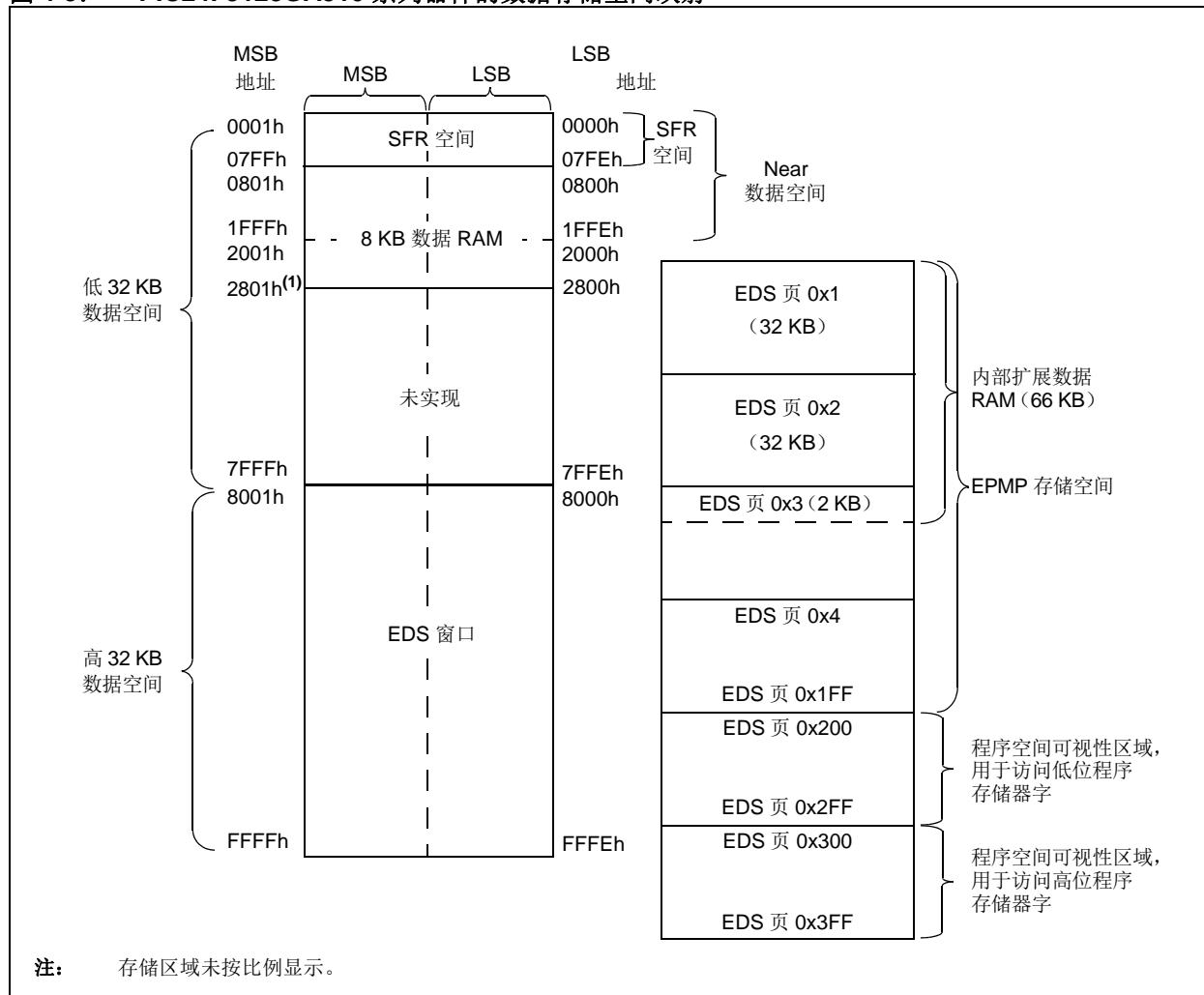
数据存储器地址空间的高半部分(8000h到FFFFh)用作通向扩展数据空间(Extended Data Space, EDS)的窗口。这使单片机能够直接访问比标准16位地址范围更大的数据范围。第4.2.5节“扩展数据空间(EDS)”详细介绍了EDS。

DS的低半部分与以前没有EDS的PIC24F单片机兼容。所有PIC24FJ128GA310系列器件都在从0800h到27FFh的低半部分地址空间中实现8 KB数据RAM。

### 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,但所有数据空间的EA都被解析为字节。每个字的低字节(Least Significant Byte, LSB)部分具有偶地址,而高字节(Most Significant Byte, MSB)部分则具有奇地址。

图4-3: PIC24FJ128GA310系列器件的数据存储空间映射



## 4.2.2 数据存储器构成和对齐方式

为了保持与PIC<sup>®</sup> MCU 向后兼容和提高数据存储空间的使用效率, PIC24F 指令集既支持字操作, 也支持字节操作。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[W<sub>s</sub>++]<sup>1</sup>的结果, 字节操作时, 内核将其识别为值W<sub>s</sub>+1, 而字操作时, 内核将其识别为W<sub>s</sub>+2。

数据字节读操作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。选中的字节将被放在数据总线的低字节处。也就是说, 数据存储器和寄存器被组织为两个字节宽的并行实体, 它们共享(字)地址译码, 而写入线相互独立。数据字节写操作仅写入存储阵列和寄存器中与字节地址匹配的相应部分。

所有字访问都必须按偶地址对齐。不支持不对齐的数据取操作。因此当混合字节和字操作或从8位MCU代码移植时, 必须要小心。若试图进行这种不对齐的读或写操作, 则会产生地址错误陷阱。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, 从而系统和/或用户能够检查地址错误发生之前的机器状态。

所有装入任意W寄存器的字节都将装入W寄存器的低字节。W寄存器的高字节(MSB)不变。

提供了一条符号扩展指令(SE), 允许用户将8位的有符号数据转换为16位的有符号值。或者, 对于16位无符号数据, 用户可以清零任何W寄存器的MSB, 方法是在相应的地址处执行一条零扩展(ZE)指令。

尽管大多数指令能够对字或字节大小的数据进行操作, 但需要注意的是, 某些指令只对字大小的数据进行操作。

## 4.2.3 NEAR 数据空间

0000h 和 1FFFh 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地址单元。剩余的数据空间通过间接寻址访问。此外, 还可以使用 MOV 指令寻址整个数据空间, 支持使用 16 位地址字段的存储器直接寻址模式。

## 4.2.4 特殊功能寄存器(SFR)空间

Near 数据空间的前 2 KB (0000h 至 07FFh) 主要被特殊功能寄存器(Special Function Register, SFR)占用。PIC24F 的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, 通常一个模块会使用一组 SFR。大部分 SFR 空间包含未用的地址单元, 它们读为 0。表 4-2 给出了 SFR 空间的布局表, 显示了实际实现 SFR 的位置。每个实现的区域代表一个 32 字节区域, 其中至少有一个地址实现为一个 SFR。表 4-3 至表 4-34 给出了所有实现的 SFR 及其地址的完整列表。

表 4-2: SFR 数据空间的已实现区域

SFR 空间地址								
	xx00	xx20	xx40	xx60	xx80	xxA0	xxC0	xxE0
000h	内核			ICN	中断			—
100h	定时器		捕捉		—	比较		—
200h	I <sup>2</sup> C <sup>TM</sup>	UART	SPI/UART	—	—	UART	I/O	
300h	A/D/CTMU			—	DMA		—	
400h	—	—	—	—	—	—	ANA	—
500h	—	—	—	—	LCD		—	LCD
600h	EPMP	RTC/CMP	CRC	—	PPS			—
700h	—	—	系统	NVM/PMD	—	—	—	—

图注: — = 在该存储块中未实现任何 SFR

表 4-3: CPU 内核寄存器映射

寄存器名称	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	所有复位时的状态
WREG0	0000																0000	
WREG1	0002																0000	
WREG2	0004																0000	
WREG3	0006																0000	
WREG4	0008																0000	
WREG5	000A																0000	
WREG6	000C																0000	
WREG7	000E																0000	
WREG8	0010																0000	
WREG9	0012																0000	
WREG10	0014																0000	
WREG11	0016																0000	
WREG12	0018																0000	
WREG13	001A																0000	
WREG14	001C																0000	
WREG15	001E																0800	
SPLIM	0020																xxxx	
PCL	002E																0000	
PCH	0030	—	—	—	—	—	—	—	—	—							0000	
DSRPAG	0032	—	—	—	—	—	—	—									0001	
DSWPAG	0034	—	—	—	—	—	—	—	—								0001	
RCOUNT	0036																xxxx	
SR	0042	—	—	—	—	—	—	—	DC	IPL2	IPL1	IPL0	RA	无	OV	Z	C	0000
CORCON	0044	—	—	—	—	—	—	—	—	—	—	—	—	IPL3	r	—	—	0004
DISICNT	0052	—	—														xxxx	
TBLPAG	0054	—	—	—	—	—	—	—	—	—							0000	

图注: — = 未实现, 读为 0, r = 保留, 不要修改。复位值以十六进制格式显示。

表 4-4: ICN 寄存器映射

寄存器名称	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	所有复位时的状态
CNPD1	0056	CN15PDE	CN14PDE	CN13PDE	CN12PDE	CN11PDE	CN10PDE	CN9PDE	CN8PDE	CN7PDE	CN6PDE	CN5PDE	CN4PDE	CN3PDE	CN2PDE	—	—	0000
CNPD2	0058	CN31PDE	CN30PDE	CN29PDE	CN28PDE	CN27PDE	CN26PDE	CN25PDE	CN24PDE	CN23PDE	CN22PDE	CN21PDE <sup>(1)</sup>	CN20PDE <sup>(1)</sup>	CN19PDE <sup>(1)</sup>	CN18PDE	CN17PDE	CN16PDE	0000
CNPD3	005A	CN47PDE <sup>(1)</sup>	CN46PDE <sup>(2)</sup>	CN45PDE <sup>(1)</sup>	CN44PDE <sup>(1)</sup>	CN43PDE <sup>(1)</sup>	CN42PDE <sup>(1)</sup>	CN41PDE <sup>(1)</sup>	CN40PDE <sup>(2)</sup>	CN39PDE <sup>(2)</sup>	CN38PDE <sup>(2)</sup>	CN37PDE <sup>(2)</sup>	CN36PDE <sup>(2)</sup>	CN35PDE <sup>(2)</sup>	CN34PDE <sup>(2)</sup>	CN33PDE <sup>(2)</sup>	CN32PDE	0000
CNPD4	005C	CN63PDE	CN62PDE	CN61PDE	CN60PDE	CN59PDE	CN58PDE	CN57PDE	CN56PDE	CN55PDE	CN54PDE	CN53PDE	CN52PDE	CN51PDE	CN50PDE	CN49PDE	CN48PDE <sup>(2)</sup>	0000
CNPD5	005E	CN79PDE <sup>(2)</sup>	CN78PDE <sup>(1)</sup>	CN77PDE <sup>(1)</sup>	CN76PDE <sup>(2)</sup>	CN75PDE <sup>(2)</sup>	CN74PDE <sup>(1)</sup>	CN73PDE	CN72PDE	CN71PDE	CN70PDE	CN69PDE	CN68PDE	CN67PDE <sup>(1)</sup>	CN66PDE <sup>(1)</sup>	CN65PDE	CN64PDE	0000
CNPD6	0060	—	—	—	—	—	—	—	—	—	—	—	CN84PDE	CN83PDE <sup>(1)</sup>	CN82PDE <sup>(2)</sup>	CN81PDE <sup>(2)</sup>	CN80PDE <sup>(2)</sup>	0000
CNEN1	0062	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	—	—	0000
CNEN2	0064	CN31IE	CN30IE	CN29IE	CN28IE	CN27IE	CN26IE	CN25IE	CN24IE	CN23IE	CN22IE	CN21IE <sup>(1)</sup>	CN20IE <sup>(1)</sup>	CN19IE <sup>(1)</sup>	CN18IE	CN17IE	CN16IE	0000
CNEN3	0066	CN47IE <sup>(1)</sup>	CN46IE <sup>(1)</sup>	CN45IE <sup>(1)</sup>	CN44IE <sup>(1)</sup>	CN43IE <sup>(1)</sup>	CN42IE <sup>(1)</sup>	CN41IE <sup>(1)</sup>	CN40IE <sup>(2)</sup>	CN39IE <sup>(2)</sup>	CN38IE <sup>(2)</sup>	CN37IE <sup>(2)</sup>	CN36IE <sup>(2)</sup>	CN35IE <sup>(2)</sup>	CN34IE <sup>(2)</sup>	CN33IE <sup>(2)</sup>	CN32IE	0000
CNEN4	0068	CN63IE	CN62IE	CN61IE	CN60IE	CN59IE	CN58IE	CN57IE	CN56IE	CN55IE	CN54IE	CN53IE	CN52IE	CN51IE	CN50IE	CN49IE	CN48IE <sup>(2)</sup>	0000
CNEN5	006A	CN79IE <sup>(2)</sup>	CN78IE <sup>(1)</sup>	CN77IE <sup>(1)</sup>	CN76IE <sup>(2)</sup>	CN75IE <sup>(2)</sup>	CN74IE <sup>(1)</sup>	CN73IE	CN72IE	CN71IE	CN70IE	CN69IE	CN68IE	CN67IE <sup>(1)</sup>	CN66IE <sup>(1)</sup>	CN65IE	CN64IE	0000
CNEN6	006C	—	—	—	—	—	—	—	—	—	—	—	CN84IE	CN83IE <sup>(1)</sup>	CN82IE <sup>(2)</sup>	CN81IE <sup>(2)</sup>	CN80IE <sup>(2)</sup>	0000
CNPU1	006E	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	—	—	0000
CNPU2	0070	CN31PUE	CN30PUE	CN29PUE	CN28PUE	CN27PUE	CN26PUE	CN25PUE	CN24PUE	CN23PUE	CN22PUE	CN21PUE <sup>(1)</sup>	CN20PUE <sup>(1)</sup>	CN19PUE <sup>(1)</sup>	CN18PUE	CN17PUE	CN16PUE	0000
CNPU3	0072	CN47PUE <sup>(1)</sup>	CN46PUE <sup>(1)</sup>	CN45PUE <sup>(1)</sup>	CN44PUE <sup>(1)</sup>	CN43PUE <sup>(1)</sup>	CN42PUE <sup>(1)</sup>	CN41PUE <sup>(1)</sup>	CN40PUE <sup>(2)</sup>	CN39PUE <sup>(2)</sup>	CN38PUE <sup>(2)</sup>	CN37PUE <sup>(2)</sup>	CN36PUE <sup>(2)</sup>	CN35PUE <sup>(2)</sup>	CN34PUE <sup>(2)</sup>	CN33PUE <sup>(2)</sup>	CN32PUE	0000
CNPU4	0074	CN63PUE	CN62PUE	CN61PUE	CN60PUE	CN59PUE	CN58PUE	CN57PUE	CN56PUE	CN55PUE	CN54PUE	CN53PUE	CN52PUE	CN51PUE	CN50PUE	CN49PUE	CN48PUE <sup>(2)</sup>	0000
CNPU5	0076	CN79PUE <sup>(2)</sup>	CN78PUE <sup>(1)</sup>	CN77PUE <sup>(1)</sup>	CN76PUE <sup>(2)</sup>	CN75PUE <sup>(2)</sup>	CN74PUE <sup>(1)</sup>	CN73PUE	CN72PUE	CN71PUE	CN70PUE	CN69PUE	CN68PUE	CN67PUE <sup>(1)</sup>	CN66PUE <sup>(1)</sup>	CN65PUE	CN64PUE	0000
CNPU6	0078	—	—	—	—	—	—	—	—	—	—	—	CN84PUE	CN83PUE <sup>(1)</sup>	CN82PUE <sup>(2)</sup>	CN81PUE <sup>(2)</sup>	CN80PUE <sup>(2)</sup>	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

表 4-5：中断控制器寄存器映射

寄存器名称	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	所有复位时的状态
INTCON1	0080	NSTDIS	—	—	—	—	—	—	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000	
INTCON2	0082	ALTIVT	DISI	—	—	—	—	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000	
IFS0	0084	—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF	T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF	0000
IFS1	0086	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF	—	IC7IF	—	INT1IF	CNIF	CMIF	M2C1IF	SI2C1IF	0000
IFS2	0088	—	DMA4IF	PMP1IF	—	OC7IF	OC6IF	OC5IF	IC6IF	IC5IF	IC4IF	IC3IF	DMA3IF	—	—	SPI2IF	SPF2IF	0000
IFS3	008A	—	RTCIF	DMA5IF	—	—	—	—	—	—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—	0000
IFS4	008C	—	—	CTMUIF	—	—	—	—	LVDIF	—	—	—	—	CRCIF	U2ERIF	U1ERIF	—	0000
IFS5	008E	—	—	—	—	—	—	U4TXIF	U4RXIF	U4ERIF	—	—	—	U3TXIF	U3RXIF	U3ERIF	—	0000
IFS6	0090	—	—	—	—	—	—	—	—	—	—	—	LCDIF	—	—	—	—	0000
IFS7	0092	—	—	—	—	—	—	—	—	—	—	JTAGIF	—	—	—	—	0000	
IEC0	0094	—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE	T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE	0000
IEC1	0096	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	DMA2IE	—	IC7IE	—	INT1IE	CNIE	CMIE	M2C1IE	SI2C1IE	0000
IEC2	0098	—	DMA4IE	PMPIE	—	OC7IE	OC6IE	OC5IE	IC6IE	IC5IE	IC4IE	IC3IE	DMA3IE	—	—	SPI2IE	SPF2IE	0000
IEC3	009A	—	RTCIE	DMA5IE	—	—	—	—	—	INT4IE	INT3IE	—	—	MI2C2IE	SI2C2IE	—	0000	
IEC4	009C	—	—	CTMUIE	—	—	—	—	LV DIE	—	—	—	—	CRCIE	U2ERIE	U1ERIE	—	0000
IEC5	009E	—	—	—	—	—	—	U4TXIE	U4RXIE	U4ERIE	—	—	—	U3TXIE	U3RXIE	U3ERIE	—	0000
IEC6	00A0	—	—	—	—	—	—	—	—	—	—	—	LCDIE	—	—	—	—	0000
IEC7	00A2	—	—	—	—	—	—	—	—	—	—	JTAGIE	—	—	—	—	0000	
IPC0	00A4	—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0	—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0	4444
IPC1	00A6	—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0	—	IC2IP2	IC2IP1	IC2IP0	—	DMA0IP2	DMA0IP1	DMA0IP0	4444
IPC2	00A8	—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0	—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0	4444
IPC3	00AA	—	—	—	—	—	DMA1IP2	DMA1IP1	DMA1IP0	—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0	0044
IPC4	00AC	—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0	—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0	4444
IPC5	00AE	—	—	—	—	—	IC7IP2	IC7IP1	IC7IP0	—	—	—	—	INT1IP2	INT1IP1	INT1IP0	4404	
IPC6	00B0	—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0	—	OC3IP2	OC3IP1	OC3IP0	—	DMA2IP2	DMA2IP1	DMA2IP0	4444
IPC7	00B2	—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0	—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0	4444
IPC8	00B4	—	—	—	—	—	—	—	—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0	0044	
IPC9	00B6	—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0	—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0	4444
IPC10	00B8	—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0	—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0	4444
IPC11	00BA	—	—	—	—	—	DMA4IP2	DMA4IP1	DMA4IP0	—	PMP1IP2	PMP1IP1	PMP1IP0	—	—	—	—	0440
IPC12	00BC	—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0	—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—	0440
IPC13	00BE	—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0	—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—	0440
IPC15	00C2	—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0	—	DMA5IP2	DMA5IP1	DMA5IP0	—	—	—	—	0440

图注：— = 未实现，读为 0。复位值以十六进制格式显示。

**表 4-5： 中断控制器寄存器映射（续）**

寄存器名称	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	所有复位时的状态
IPC16	00C4	—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0	—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—	4440
IPC18	00C8	—	—	—	—	—	—	—	—	—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0	0004
IPC19	00CA	—	—	—	—	—	—	—	—	—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—	0040
IPC20	00CC	—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0	—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—	4440
IPC21	00CE	—	U4ERIP2	U4ERIP1	U4ERIP0	—	—	—	—	—	—	—	—	—	—	—	—	4000
IPC22	00D0	—	—	—	—	—	—	—	—	—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0	0044
IPC25	00D6	—	—	—	—	—	—	—	—	—	—	—	—	—	LCDIP2	LCDIP1	LCDIP0	0004
IPC29	00DE	—	—	—	—	—	—	—	—	—	JTAGIP2	JTAGIP1	JTAGIP0	—	—	—	—	0040
INTTREG	00E0	CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0	—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0	0000

图注：— = 未实现，读为 0。复位值以十六进制格式显示。

**表 4-6： 定时器寄存器映射**

寄存器名称	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	所有复位时的状态
TMR1	0100	Timer1 寄存器															0000	
PR1	0102	Timer1 周期寄存器															FFFF	
T1CON	0104	TON	—	TSIDL	—	—	—	TIECS1	TIECS0	—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—	0000
TMR2	0106	Timer2 寄存器															0000	
TMR3HLD	0108	Timer3 保持寄存器（仅适用于 32 位定时器操作）															0000	
TMR3	010A	Timer3 寄存器															0000	
PR2	010C	Timer2 周期寄存器															FFFF	
PR3	010E	Timer3 周期寄存器															FFFF	
T2CON	0110	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T32	—	TCS	—	0000
T3CON	0112	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000
TMR4	0114	Timer4 寄存器															0000	
TMR5HLD	0116	Timer5 保持寄存器（仅适用于 32 位定时器操作）															0000	
TMR5	0118	Timer5 寄存器															0000	
PR4	011A	Timer4 周期寄存器															FFFF	
PR5	011C	Timer5 周期寄存器															FFFF	
T4CON	011E	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	T45	—	TCS	—	0000
T5CON	0120	TON	—	TSIDL	—	—	—	—	—	—	TGATE	TCKPS1	TCKPS0	—	—	TCS	—	0000

图注：— = 未实现，读为 0。复位值以十六进制格式显示。

表 4-7: 输入捕捉寄存器映射

寄存器名称	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	所有复位时的状态
IC1CON1	0140	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC1CON2	0142	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC1BUF	0144	输入捕捉 1 缓冲寄存器														—	0000	
IC1TMR	0146	定时器值 1 寄存器														—	xxxx	
IC2CON1	0148	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC2CON2	014A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC2BUF	014C	输入捕捉 2 缓冲寄存器														—	0000	
IC2TMR	014E	定时器值 2 寄存器														—	xxxx	
IC3CON1	0150	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC3CON2	0152	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC3BUF	0154	输入捕捉 3 缓冲寄存器														—	0000	
IC3TMR	0156	定时器值 3 寄存器														—	xxxx	
IC4CON1	0158	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC4CON2	015A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC4BUF	015C	输入捕捉 4 缓冲寄存器														—	0000	
IC4TMR	015E	定时器值 4 寄存器														—	xxxx	
IC5CON1	0160	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC5CON2	0162	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC5BUF	0164	输入捕捉 5 缓冲寄存器														—	0000	
IC5TMR	0166	定时器值 5 寄存器														—	xxxx	
IC6CON1	0168	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC6CON2	016A	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC6BUF	016C	输入捕捉 6 缓冲寄存器														—	0000	
IC6TMR	016E	定时器值 6 寄存器														—	xxxx	
IC7CON1	0170	—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—	IC11	IC10	ICOV	ICBNE	ICM2	ICM1	ICM0	0000	
IC7CON2	0172	—	—	—	—	—	—	—	—	IC32	ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
IC7BUF	0174	输入捕捉 7 缓冲寄存器														—	0000	
IC7TMR	0176	定时器值 7 寄存器														—	xxxx	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

**表 4-8： 输出比较寄存器映射**

寄存器名称	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	所有复位时的状态
OC1CON1	0190	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC1CON2	0192	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC1RS	0194	输出比较 1 辅助寄存器																0000
OC1R	0196	输出比较 1 寄存器																0000
OC1TMR	0198	定时器值 1 寄存器																xxxx
OC2CON1	019A	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC2CON2	019C	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC2RS	019E	输出比较 2 辅助寄存器																0000
OC2R	01A0	输出比较 2 寄存器																0000
OC2TMR	01A2	定时器值 2 寄存器																xxxx
OC3CON1	01A4	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC3CON2	01A6	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC3RS	01A8	输出比较 3 辅助寄存器																0000
OC3R	01AA	输出比较 3 寄存器																0000
OC3TMR	01AC	定时器值 3 寄存器																xxxx
OC4CON1	01AE	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC4CON2	01B0	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC4RS	01B2	输出比较 4 辅助寄存器																0000
OC4R	01B4	输出比较 4 寄存器																0000
OC4TMR	01B6	定时器值 4 寄存器																xxxx
OC5CON1	01B8	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT1	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC5CON2	01BA	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC5RS	01BC	输出比较 5 辅助寄存器																0000
OC5R	01BE	输出比较 5 寄存器																0000
OC5TMR	01C0	定时器值 5 寄存器																xxxx
OC6CON1	01C2	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCM0	0000
OC6CON2	01C4	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC6RS	01C6	输出比较 6 辅助寄存器																0000
OC6R	01C8	输出比较 6 寄存器																0000
OC6TMR	01CA	定时器值 6 寄存器																xxxx

图注：— = 未实现，读为 0。复位值以十六进制格式显示。

表 4-8: 输出比较寄存器映射 (续)

寄存器名称	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	所有复位时的状态
OC7CON1	01CC	—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2	ENFLT1	ENFLT0	OCFLT2	OCFLT1	OCFLT0	TRIGMODE	OCM2	OCM1	OCMO	0000
OC7CON2	01CE	FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1	DCB0	OC32	OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0	000C
OC7RS	01D0	输出比较 7 辅助寄存器															0000	
OC7R	01D2	输出比较 7 寄存器															0000	
OC7TMR	01D4	定时器值 7 寄存器															xxxx	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-9: I<sup>2</sup>C<sup>TM</sup> 寄存器映射

寄存器名称	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	所有复位时的状态		
I2C1RCV	0200	—	—	—	—	—	—	—	—	接收寄存器									0000	
I2C1TRN	0202	—	—	—	—	—	—	—	—	发送寄存器									00FF	
I2C1BRG	0204	—	—	—	—	—	—	—	—	波特率发生器寄存器									0000	
I2C1CON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000		
I2C1STAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000		
I2C1ADD	020A	—	—	—	—	—	—	—	地址寄存器									0000		
I2C1MSK	020C	—	—	—	—	—	—	—	地址掩码寄存器									0000		
I2C2RCV	0210	—	—	—	—	—	—	—	—	—	接收寄存器									0000
I2C2TRN	0212	—	—	—	—	—	—	—	—	—	发送寄存器									00FF
I2C2BRG	0214	—	—	—	—	—	—	—	—	波特率发生器寄存器									0000	
I2C2CON	0216	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	1000		
I2C2STAT	0218	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF	0000		
I2C2ADD	021A	—	—	—	—	—	—	—	地址寄存器									0000		
I2C2MSK	021C	—	—	—	—	—	—	—	地址掩码寄存器									0000		

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

**表 4-10: UART 寄存器映射**

寄存器名称	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	所有复位时的状态
U1MODE	0220	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UE0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U1STA	0222	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U1TXREG	0224	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx	
U1RXREG	0226	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
U1BRG	0228	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
波特率发生器预分频寄存器																		
U2MODE	0230	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UE0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U2STA	0232	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U2TXREG	0234	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx	
U2RXREG	0236	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
U2BRG	0238	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
波特率发生器预分频寄存器																		
U3MODE	0250	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UE0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U3STA	0252	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U3TXREG	0254	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx	
U3RXREG	0256	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
U3BRG	0258	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
波特率发生器预分频寄存器																		
U4MODE	02B0	UARTEN	—	USIDL	IREN	RTSMD	—	UEN1	UE0	WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL	0000
U4STA	02B2	UTXISEL1	UTXINV	UTXISEL0	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0110
U4TXREG	02B4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	xxxx	
U4RXREG	02B6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	
U4BRG	02B8	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000	

图注：— = 未实现，读为 0。复位值以十六进制格式显示。

表 4-11: SPI 寄存器映射

寄存器名称	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	所有复位时的状态
SPI1STAT	0240	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI1CON1	0242	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI1CON2	0244	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI1BUF	0248	发送和接收缓冲器																0000
SPI2STAT	0260	SPIEN	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0	SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF	0000
SPI2CON1	0262	—	—	—	DISSCK	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000
SPI2CON2	0264	FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—	—	—	—	—	—	—	SPIFE	SPIBEN	0000
SPI2BUF	0268	发送和接收缓冲器																0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-12: PORTA 寄存器映射<sup>(1)</sup>

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7 <sup>(2)</sup>	Bit 6 <sup>(2)</sup>	Bit 5 <sup>(2)</sup>	Bit 4 <sup>(2)</sup>	Bit 3 <sup>(2)</sup>	Bit 2 <sup>(2)</sup>	Bit 1 <sup>(2)</sup>	Bit 0 <sup>(2)</sup>	所有复位时的状态
TRISA	02C0	TRISA15	TRISA14	—	—	—	TRISA10	TRISA9	—	TRISA7	TRISA6	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	C6FF
PORTA	02C2	RA15	RA14	—	—	—	RA10	RA9	—	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0	xxxx
LATA	02C4	LATA15	LATA14	—	—	—	LATA10	LATA9	—	LATA7	LATA6	LATA5	LATA4	LATA3	LATA2	LATA1	LATA0	xxxx
ODCA	02C6	ODA15	ODA14	—	—	—	ODA10	ODA9	—	ODA7	ODA6	ODA5	ODA4	ODA3	ODA2	ODA1	ODA0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: PORTA 和所有相关位在 64 引脚器件上均未实现, 读为 0。

2: 这些位在 80 引脚器件上也未实现, 读为 0。

表 4-13: PORTB 寄存器映射

寄存器名称	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	所有复位时的状态
TRISB	02C8	TRISB15	TRISB14	TRISB13	TRISB12	TRISB11	TRISB10	TRISB9	TRISB8	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	FFFF
PORTB	02CA	RB15	RB14	RB13	RB12	RB11	RB10	RB9	RB8	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx
LATB	02CC	LATB15	LATB14	LATB13	LATB12	LATB11	LATB10	LATB9	LATB8	LATB7	LATB6	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	xxxx
ODCB	02CE	ODB15	ODB14	ODB13	ODB12	ODB11	ODB10	ODB9	ODB8	ODB7	ODB6	ODB5	ODB4	ODB3	ODB2	ODB1	ODB0	0000

图注: 复位值以十六进制格式显示。

**表 4-14: PORTC 寄存器映射**

寄存器名称	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 <sup>(1)</sup>	Bit 3 <sup>(2)</sup>	Bit 2 <sup>(1)</sup>	Bit 1 <sup>(2)</sup>	Bit 0	所有复位时的状态
TRISC	02D0	TRISC15	—	—	TRISC12	—	—	—	—	—	—	—	TRISC4	TRISC3	TRISC2	TRISC1	—	901E
PORTC	02D2	RC15 <sup>(3,4)</sup>	RC14 <sup>(5)</sup>	RC13 <sup>(5)</sup>	RC12 <sup>(3)</sup>	—	—	—	—	—	—	—	RC4	RC3	RC2	RC1	—	xxxx
LATC	02D4	LATC15	LATC14	LATC13	LATC12	—	—	—	—	—	—	—	LATC4	LATC3	LATC2	LATC1	—	xxxx
ODCC	02D6	ODC15	ODC14	ODC13	ODC12	—	—	—	—	—	—	—	ODC4	ODC3	ODC2	ODC1	—	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚器件上未实现这些位, 读为 0。

3: RC12 和 RC15 仅在禁止主振荡器和选择 EC 模式 (POSCMD<1:0> 配置位 = 11 或 00) 时可用, 否则均读为 0。

4: RC15 仅在 POSCMD<1:0> 配置位 = 11 或 00 且 OSCIOFN 配置位 = 1 时可用。

5: RC13 和 RC14 仅为输入端口, 不可用作输出端口。

**表 4-15: PORTD 寄存器映射**

寄存器名称	地址	Bit 15 <sup>(1)</sup>	Bit 14 <sup>(1)</sup>	Bit 13 <sup>(1)</sup>	Bit 12 <sup>(1)</sup>	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISD	02D8	TRISD15	TRISD14	TRISD13	TRISD12	TRISD11	TRISD10	TRISD9	TRISD8	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	FFFF
PORTD	02DA	RD15	RD14	RD13	RD12	RD11	RD10	RD9	RD8	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx
LATD	02DC	LATD15	LATD14	LATD13	LATD12	LATD11	LATD10	LATD9	LATD8	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	xxxx
ODCD	02DE	ODD15	ODD14	ODD13	ODD12	ODD11	ODD10	ODD9	ODD8	ODD7	ODD6	ODD5	ODD4	ODD3	ODD2	ODD1	ODD0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

**表 4-16: PORTE 寄存器映射**

寄存器名称	地址	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9 <sup>(1)</sup>	Bit 8 <sup>(1)</sup>	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISE	02E0	—	—	—	—	—	—	TRISE9	TRISE8	TRISE7	TRISE6	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	03FF
PORTE	02E2	—	—	—	—	—	—	RE9	RE8	RE7	RE6	RE5	RE4	RE3	RE2	RE1	RE0	xxxx
LATE	02E4	—	—	—	—	—	—	LATE9	LATE8	LATE7	LATE6	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	xxxx
ODEC	02E6	—	—	—	—	—	—	ODE9	ODE8	ODE7	ODE6	ODE5	ODE4	ODE3	ODE2	ODE1	ODE0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

表 4-17: PORTF 寄存器映射

寄存器名称	地址	Bit 15	Bit 14	Bit 13 <sup>(1)</sup>	Bit 12 <sup>(1)</sup>	Bit 11	Bit 10	Bit 9	Bit 8 <sup>(2)</sup>	Bit 7 <sup>(2)</sup>	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	所有复位时的状态
TRISF	02E8	—	—	TRISF13	TRISF12	—	—	—	TRISF8	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	TRISF0	31FF
PORTF	02EA	—	—	RF13	RF12	—	—	—	RF8	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0	xxxx
LATF	02EC	—	—	LATF13	LATF12	—	—	—	LATF8	LATF7	LATF6	LATF5	LATF4	LATF3	LATF2	LATF1	LATF0	xxxx
ODCF	02EE	—	—	ODF13	ODF12	—	—	—	ODF8	ODF7	ODF6	ODF5	ODF4	ODF3	ODF2	ODF1	ODF0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

表 4-18: PORTG 寄存器映射

寄存器名称	地址	Bit 15 <sup>(1)</sup>	Bit 14 <sup>(1)</sup>	Bit 13 <sup>(1)</sup>	Bit 12 <sup>(1)</sup>	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1 <sup>(2)</sup>	Bit 0 <sup>(2)</sup>	所有复位时的状态
TRISG	02F0	TRISG15	TRISG14	TRISG13	TRISG12	—	—	TRISG9	TRISG8	TRISG7	TRISG6	—	—	TRISG3	TRISG2	TRISG1	TRISG0	F3CF
PORTG	02F2	RG15	RG14	RG13	RG12	—	—	RG9	RG8	RG7	RG6	—	—	RG3	RG2	RG1	RG0	xxxx
LATG	02F4	LATG15	LATG14	LATG13	LATG12	—	—	LATG9	LATG8	LATG7	LATG6	—	—	LATG3	LATG2	LATG1	LATG0	xxxx
ODCG	02F6	ODG15	ODG14	ODG13	ODG12	—	—	ODG9	ODG8	ODG7	ODG6	—	—	ODG3	ODG2	ODG1	ODG0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。显示的复位值适用于 100 引脚器件。

注 1: 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚器件上未实现这些位, 读为 0。

表 4-19: 焊盘配置寄存器映射 (PADCFIG1)

寄存器名称	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	所有复位时的状态
PADCFIG1	02FC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PMPTTL	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-20: A/D 寄存器映射

寄存器名称	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	所有复位时的状态
ADC1BUF0	0300																	xxxxx
ADC1BUF1	0302																	xxxxx
ADC1BUF2	0304																	xxxxx
ADC1BUF3	0306																	xxxxx
ADC1BUF4	0308																	xxxxx
ADC1BUF5	030A																	xxxxx
ADC1BUF6	030C																	xxxxx
ADC1BUF7	030E																	xxxxx
ADC1BUF8	0310																	xxxxx
ADC1BUF9	0312																	xxxxx
ADC1BUF10	0314																	xxxxx
ADC1BUF11	0316																	xxxxx
ADC1BUF12	0318																	xxxxx
ADC1BUF13	031A																	xxxxx
ADC1BUF14	031C																	xxxxx
ADC1BUF15	031E																	xxxxx
ADC1BUF16	0320																	xxxxx
ADC1BUF17	0322																	xxxxx
ADC1BUF18	0324																	xxxxx
ADC1BUF19	0326																	xxxxx
ADC1BUF20	0328																	xxxxx
ADC1BUF21	032A																	xxxxx
ADC1BUF22	032C																	xxxxx
ADC1BUF23	032E																	xxxxx
ADC1BUF24	0330																	xxxxx
ADC1BUF25	0332																	xxxxx
AD1CON1	0340	ADON	—	ADSIDL	DMABM	DMAEN	MODE12	FORM1	FORM0	SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE	0000
AD1CON2	0342	PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—	BUFS	SMPI4	SMPI3	SMPI2	SMPI1	SMPIO	BUFM	ALTS	0000
AD1CON3	0344	ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMCO	ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0	0000
AD1CHS	0348	CH0NB2	CH0NB1	CH0NB0	CH0SB4	CH0SB3	CH0SB2	CH0SB1	CH0SB0	CH0NA2	CH0NA1	CH0NA0	CH0SA4	CH0SA3	CH0SA2	CH0SA1	CH0SA0	0000
AD1CSSH	034E	—	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24	CSS23	CSS22	CSS21	CSS20	CSS19	CSS18	CSS17	CSS16	0000
AD1CSSL	0350	CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8	CSS7	CSS6	CSS5	CSS4	CSS3	CSS2	CSS1	CSS0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

表 4-20: A/D 寄存器映射 (续)

寄存器名称	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	所有复位时的状态
AD1CON4	0352	—	—	—	—	—	—	—	—	—	—	—	—	—	DMABL2	DMABL1	DMABL0 0000	
AD1CON5	0354	ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0	—	—	—	—	WM1	WM0	CM1	CM0 0000	
AD1CHITH	0356	—	—	—	—	—	—	CHH25 <sup>(1)</sup>	CHH24 <sup>(1)</sup>	CHH23 <sup>(1)</sup>	CHH22 <sup>(1)</sup>	CHH21 <sup>(1)</sup>	CHH20 <sup>(1)</sup>	CHH19 <sup>(1)</sup>	CHH18 <sup>(1)</sup>	CHH17 <sup>(1)</sup>	CHH16 <sup>(1)</sup> 0000	
AD1CHITL	0358	CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8	CHH7	CHH6	CHH5	CHH4	CHH3	CHH2	CHH1	CHH0 0000	
AD1CTMENH	0360	—	CTMEN30	CTMEN29	CTMEN28	CTMEN27	CTMEN26	CTMEN25	CTMEN24	CTMEN23	CTMEN22	CTMEN21	CTMEN20	CTMEN19	CTMEN18	CTMEN17	CTMEN16 0000	
AD1CTMENL	0362	CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMEN11	CTMEN10	CTMEN9	CTMEN8	CTMEN7	CTMEN6	CTMEN5	CTMEN4	CTMEN3	CTMEN2	CTMEN1	CTMEN0 0000	
AD1DMBUF	0364	转换数据缓冲区 (扩展缓冲区模式)															xxxx	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

表 4-21: CTMU 寄存器映射

寄存器名称	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	所有复位时的状态
CTMUCON1	035A	CTMUEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG	—	—	—	—	—	—	—	0000	
CTMUCON2	035C	EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT	EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	0000	
CTMUCON	035E	ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0	—	—	—	—	—	—	—	0000	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-22: 模拟配置寄存器映射

寄存器名称	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	所有复位时的状态
ANCFG	04DE	—	—	—	—	—	—	—	—	—	—	—	—	—	VBG6EN	VBG2EN	VBGEN 0000	
ANSA	04E0	—	—	—	—	—	—	—	—	ANS A7 <sup>(1)</sup>	ANS A6 <sup>(1)</sup>	—	—	—	—	—	00C0	
ANSB	04E2	ANSB15	ANSB14	ANSB13	ANSB12	ANSB11	ANSB10	ANSB9	ANSB8	ANSB7	ANSB6	ANSB5	ANSB4	ANSB3	ANSB2	ANSB1	ANSB0 FFFF	
ANSC	04E4	—	—	—	—	—	—	—	—	—	—	—	ANSC4 <sup>(1)</sup>	—	—	—	0010	
ANSD	04E6	—	—	—	—	ANS D11	ANS D10	—	—	ANS D7	ANS D6	—	—	—	—	—	0CC0	
ANSE	04E8	—	—	—	—	—	—	ANSE9 <sup>(2)</sup>	—	ANSE7	ANSE6	ANSE5	ANSE4	—	—	—	02F0	
ANSG	04EC	—	—	—	—	—	—	ANS G9	ANS G8	ANS G7	ANS G6	—	—	—	—	—	03C0	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

注 2: 在 64 引脚器件未实现这些位。在 80 引脚器件上, 此位需要清零才能使 RE9 具有数字功能。

表 4-23: DMA 寄存器映射

寄存器名称	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	所有复位时的状态
DMACON	0380	DMAEN	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRSSEL	0000
DMABUF	0382																	0000
DMAL	0384																	0000
DMAH	0386																	0000
DMACH0	0388	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT0	038A	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC0	038C																	0000
DMADST0	038E																	0000
DMACNT0	0390																	0001
DMACH1	0392	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT1	0394	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC1	0396																	0000
DMADST1	0398																	0000
DMACNT1	039A																	0001
DMACH2	039C	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT2	039E	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC2	03A0																	0000
DMADST2	03A2																	0000
DMACNT2	03A4																	0001
DMACH3	03A6	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT3	03A8	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC3	03AA																	0000
DMADST3	03AC																	0000
DMACNT3	03AE																	0001
DMACH4	03B0	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT4	03B2	DBUFWF	—	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC4	03B4																	0000
DMADST4	03B6																	0000
DMACNT4	03B8																	0001
DMACH5	03BA	—	—	—	—	—	NULLW	RELOAD	CHREQ	SAMODE1	SAMODE0	DAMODE1	DAMODE0	TRMODE1	TRMODE0	BYTE	CHEN	0000
DMAINT5	03BC	DBUFWF	—	—	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0	HIGHIF	LOWIF	DONEIF	HALFIF	OVRUNIF	—	—	HALFEN	0000
DMASRC5	03BE																	0000
DMADST5	03C0																	0000
DMACNT5	03C2																	0001

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-24: LCD 寄存器映射

寄存器名称	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	所有复位时的状态
LCDREG	0580	CPEN	—	—	—	—	—	—	—	—	BIAS2	BIAS1	BIAS0	MODE13	CKSEL1	CKSEL0	0000	
LCDREF	0582	LCDIRE	—	LCDCST2	LCDCST1	LCDCST0	VLCD3PE	VLCD2PE	VLCD1PE	LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0	0000
LCDCON	0584	LCDEN	—	LCDSIDL	—	—	—	—	—	—	SLPEN	WERR	CS1	CS0	LMUX2	LMUX1	LMUX0	0000
LCDPS	0586	—	—	—	—	—	—	—	WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0	0000	
LCDDATA0	0590	S15C0	S14C0	S13C0	S12C0	S11C0	S10C0	S09C0	S08C0	S07C0	S06C0	S05C0	S04C0	S03C0	S02C0	S01C0	S00C0	0000
LCDDATA1	0592	S31C0	S30C0	S29C0	S28C0	S27C0	S26C0	S25C0	S24C0	S23C0	S22C0	S21C0	S20C0	S19C0	S18C0	S17C0	S16C0	0000
LCDDATA2	0594	S47C0	S46C0 <sup>(1)</sup>	S45C0 <sup>(1)</sup>	S44C0 <sup>(1)</sup>	S43C0 <sup>(1)</sup>	S42C0 <sup>(1)</sup>	S41C0 <sup>(1)</sup>	S40C0 <sup>(1)</sup>	S39C0 <sup>(1)</sup>	S38C0 <sup>(1)</sup>	S37C0 <sup>(1)</sup>	S36C0 <sup>(1)</sup>	S35C0 <sup>(1)</sup>	S34C0 <sup>(1)</sup>	S33C0 <sup>(1)</sup>	S32C0 <sup>(1)</sup>	0000
LCDDATA3	0596	S63C0 <sup>(2)</sup>	S62C0 <sup>(2)</sup>	S61C0 <sup>(2)</sup>	S60C0 <sup>(2)</sup>	S59C0 <sup>(2)</sup>	S58C0 <sup>(2)</sup>	S57C0 <sup>(2)</sup>	S56C0 <sup>(2)</sup>	S55C0 <sup>(2)</sup>	S54C0 <sup>(2)</sup>	S53C0 <sup>(2)</sup>	S52C0 <sup>(2)</sup>	S51C0 <sup>(2)</sup>	S50C0 <sup>(1)</sup>	S49C0 <sup>(2)</sup>	S48C0	0000
LCDDATA4	0598	S15C1	S14C1	S13C1	S12C1	S11C1	S10C1	S09C1	S08C1	S07C1	S06C1	S05C1	S04C1	S03C1	S02C1	S01C1	S00C1	0000
LCDDATA5	059A	S31C1	S30C1	S29C1	S28C1	S27C1	S26C1	S25C1	S24C1	S23C1	S22C1	S21C1	S20C1	S19C1	S18C1	S17C1	S16C1	0000
LCDDATA6	059C	S47C1	S46C1 <sup>(1)</sup>	S45C1 <sup>(1)</sup>	S44C1 <sup>(1)</sup>	S43C1 <sup>(1)</sup>	S42C1 <sup>(1)</sup>	S41C1 <sup>(1)</sup>	S40C1 <sup>(1)</sup>	S39C1 <sup>(1)</sup>	S38C1 <sup>(1)</sup>	S37C1 <sup>(1)</sup>	S36C1 <sup>(1)</sup>	S35C1 <sup>(1)</sup>	S34C1 <sup>(1)</sup>	S33C1 <sup>(1)</sup>	S32C1 <sup>(1)</sup>	0000
LCDDATA7	059E	S63C1 <sup>(2)</sup>	S62C1 <sup>(2)</sup>	S61C1 <sup>(2)</sup>	S60C1 <sup>(2)</sup>	S59C1 <sup>(2)</sup>	S58C1 <sup>(2)</sup>	S57C1 <sup>(2)</sup>	S56C1 <sup>(2)</sup>	S55C1 <sup>(2)</sup>	S54C1 <sup>(2)</sup>	S53C1 <sup>(2)</sup>	S52C1 <sup>(2)</sup>	S51C1 <sup>(2)</sup>	S50C1 <sup>(1)</sup>	S49C1 <sup>(2)</sup>	S48C1	0000
LCDDATA8	05A0	S15C2	S14C2	S13C2	S12C2	S11C2	S10C2	S09C2	S08C2	S07C2	S06C2	S05C2	S04C2	S03C2	S02C2	S01C2	S00C2	0000
LCDDATA9	05A2	S31C2	S30C2	S29C2	S28C2	S27C2	S26C2	S25C2	S24C2	S23C2	S22C2	S21C2	S20C2	S19C2	S18C2	S17C2	S16C2	0000
LCDDATA10	05A4	S47C2	S46C2 <sup>(1)</sup>	S45C2 <sup>(1)</sup>	S44C2 <sup>(1)</sup>	S43C2 <sup>(1)</sup>	S42C2 <sup>(1)</sup>	S41C2 <sup>(1)</sup>	S40C2 <sup>(1)</sup>	S39C2 <sup>(1)</sup>	S38C2 <sup>(1)</sup>	S37C2 <sup>(1)</sup>	S36C2 <sup>(1)</sup>	S35C2 <sup>(1)</sup>	S34C2 <sup>(1)</sup>	S33C2 <sup>(1)</sup>	S32C2 <sup>(1)</sup>	0000
LCDDATA11	05A6	S63C2 <sup>(2)</sup>	S62C2 <sup>(2)</sup>	S61C2 <sup>(2)</sup>	S60C2 <sup>(2)</sup>	S59C2 <sup>(2)</sup>	S58C2 <sup>(2)</sup>	S57C2 <sup>(2)</sup>	S56C2 <sup>(2)</sup>	S55C2 <sup>(2)</sup>	S54C2 <sup>(2)</sup>	S53C2 <sup>(2)</sup>	S52C2 <sup>(2)</sup>	S51C2 <sup>(2)</sup>	S50C2 <sup>(1)</sup>	S49C2 <sup>(2)</sup>	S48C2	0000
LCDDATA12	05A8	S15C3	S14C3	S13C3	S12C3	S11C3	S10C3	S09C3	S08C3	S07C3	S06C3	S05C3	S04C3	S03C3	S02C3	S01C3	S00C3	0000
LCDDATA13	05AA	S31C3	S30C3	S29C3	S28C3	S27C3	S26C3	S25C3	S24C3	S23C3	S22C3	S21C3	S20C3	S19C3	S18C3	S17C3	S16C3	0000
LCDDATA14	05AC	S47C3	S46C3 <sup>(1)</sup>	S45C3 <sup>(1)</sup>	S44C3 <sup>(1)</sup>	S43C3 <sup>(1)</sup>	S42C3 <sup>(1)</sup>	S41C3 <sup>(1)</sup>	S40C3 <sup>(1)</sup>	S39C3 <sup>(1)</sup>	S38C3 <sup>(1)</sup>	S37C3 <sup>(1)</sup>	S36C3 <sup>(1)</sup>	S35C3 <sup>(1)</sup>	S34C3 <sup>(1)</sup>	S33C3 <sup>(1)</sup>	S32C3 <sup>(1)</sup>	0000
LCDDATA15	05AE	S63C3 <sup>(2)</sup>	S62C3 <sup>(2)</sup>	S61C3 <sup>(2)</sup>	S60C3 <sup>(2)</sup>	S59C3 <sup>(2)</sup>	S58C3 <sup>(2)</sup>	S57C3 <sup>(2)</sup>	S56C3 <sup>(2)</sup>	S55C3 <sup>(2)</sup>	S54C3 <sup>(2)</sup>	S53C3 <sup>(2)</sup>	S52C3 <sup>(2)</sup>	S51C3 <sup>(2)</sup>	S50C3 <sup>(1)</sup>	S49C3 <sup>(2)</sup>	S48C3	0000
LCDSE3	058E	SE63 <sup>(2)</sup>	SE62 <sup>(2)</sup>	SE61 <sup>(2)</sup>	SE60 <sup>(2)</sup>	SE59 <sup>(2)</sup>	SE58 <sup>(2)</sup>	SE57 <sup>(2)</sup>	SE56 <sup>(2)</sup>	SE55 <sup>(2)</sup>	SE54 <sup>(2)</sup>	SE53 <sup>(2)</sup>	SE52 <sup>(2)</sup>	SE51 <sup>(2)</sup>	SE50 <sup>(1)</sup>	SE49 <sup>(2)</sup>	SE48	0000
LCDSE2	058C	SE47	SE46 <sup>(1)</sup>	SE45 <sup>(1)</sup>	SE44 <sup>(1)</sup>	SE43 <sup>(1)</sup>	SE42 <sup>(1)</sup>	SE41 <sup>(1)</sup>	SE40 <sup>(1)</sup>	SE39 <sup>(1)</sup>	SE38 <sup>(1)</sup>	SE37 <sup>(1)</sup>	SE36 <sup>(1)</sup>	SE35 <sup>(1)</sup>	SE34 <sup>(1)</sup>	SE33 <sup>(1)</sup>	SE32 <sup>(1)</sup>	0000
LCDSE1	058A	SE31	SE30	SE29	SE28	SE27	SE26	SE25	SE24	SE23	SE22	SE21	SE20	SE019	SE18	SE17	SE16	0000
LCDSE0	0588	SE15	SE14	SE13	SE12	SE11	SE10	SE09	SE08	SE07	SE06	SE05	SE04	SE03	SE02	SE01	SE00	0000
LCDDATA16	05B0	S15C4	S14C4	S13C4	S12C4	S11C4	S10C4	S09C4	S08C4	S07C4	S06C4	S05C4	S04C4	S03C4	S02C4	S01C4	S00C4	0000
LCDDATA17	05B2	S31C4	S30C4	S29C4	S28C4	S27C4	S26C4	S25C4	S24C4	S23C4	S22C4	S21C4	S20C4	S19C4	S18C4	S17C4	S16C4	0000
LCDDATA18	05B4	S47C4	S46C4 <sup>(1)</sup>	S45C4 <sup>(1)</sup>	S44C4 <sup>(1)</sup>	S43C4 <sup>(1)</sup>	S42C4 <sup>(1)</sup>	S41C4 <sup>(1)</sup>	S40C4 <sup>(1)</sup>	S39C4 <sup>(1)</sup>	S38C4 <sup>(1)</sup>	S37C4 <sup>(1)</sup>	S36C4 <sup>(1)</sup>	S35C4 <sup>(1)</sup>	S34C4 <sup>(1)</sup>	S33C4 <sup>(1)</sup>	S32C4 <sup>(1)</sup>	0000
LCDDATA19	05B6	S63C4 <sup>(2)</sup>	S62C4 <sup>(2)</sup>	S61C4 <sup>(2)</sup>	S60C4 <sup>(2)</sup>	S59C4 <sup>(2)</sup>	S58C4 <sup>(2)</sup>	S57C4 <sup>(2)</sup>	S56C4 <sup>(2)</sup>	S55C4 <sup>(2)</sup>	S54C4 <sup>(2)</sup>	S53C4 <sup>(2)</sup>	S52C4 <sup>(2)</sup>	S51C4 <sup>(2)</sup>	S50C4 <sup>(1)</sup>	S49C4 <sup>(2)</sup>	S48C4	0000
LCDDATA20	05B8	S15C5	S14C5	S13C5	S12C5	S11C5	S10C5	S09C5	S08C5	S07C5	S06C5	S05C5	S04C5	S03C5	S02C5	S01C5	S00C5	0000
LCDDATA21	05BA	S31C5	S30C5	S29C5	S28C5	S27C5	S26C5	S25C5	S24C5	S23C5	S22C5	S21C5	S20C5	S19C5	S18C5	S17C5	S16C5	0000
LCDDATA22	05BC	S47C5	S46C5 <sup>(1)</sup>	S45C5 <sup>(1)</sup>	S44C5 <sup>(1)</sup>	S43C5 <sup>(1)</sup>	S42C5 <sup>(1)</sup>	S41C5 <sup>(1)</sup>	S40C5 <sup>(1)</sup>	S39C5 <sup>(1)</sup>	S38C5 <sup>(1)</sup>	S37C5 <sup>(1)</sup>	S36C5 <sup>(1)</sup>	S35C5 <sup>(1)</sup>	S34C5 <sup>(1)</sup>	S33C5 <sup>(1)</sup>	S32C5 <sup>(1)</sup>	0000
LCDDATA23	05BE	S63C5 <sup>(2)</sup>	S62C5 <sup>(2)</sup>	S61C5 <sup>(2)</sup>	S60C5 <sup>(2)</sup>	S59C5 <sup>(2)</sup>	S58C5 <sup>(2)</sup>	S57C5 <sup>(2)</sup>	S56C5 <sup>(2)</sup>	S55C5 <sup>(2)</sup>	S54C5 <sup>(2)</sup>	S53C5 <sup>(2)</sup>	S52C5 <sup>(2)</sup>	S51C5 <sup>(2)</sup>	S50C5 <sup>(1)</sup>	S49C5 <sup>(2)</sup>	S48C5	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

**表 4-24: LCD 寄存器映射 (续)**

寄存器名称	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	所有复位时的状态
LCDDATA24	05C0	S15C6	S14C6	S13C6	S12C6	S11C6	S10C6	S09C6	S08C6	S07C6	S06C6	S05C6	S04C6	S03C6	S02C6	S01C6	S00C6	0000
LCDDATA25	05C2	S31C6	S30C6	S29C6	S28C6	S27C6	S26C6	S25C6	S24C6	S23C6	S22C6	S21C6	S20C6	S19C6	S18C6	S17C6	S16C6	0000
LCDDATA26	05C4	S47C6	S46C6 <sup>(1)</sup>	S45C6 <sup>(1)</sup>	S44C6 <sup>(1)</sup>	S43C6 <sup>(1)</sup>	S42C6 <sup>(1)</sup>	S41C6 <sup>(1)</sup>	S40C6 <sup>(1)</sup>	S39C6 <sup>(1)</sup>	S38C6 <sup>(1)</sup>	S37C6 <sup>(1)</sup>	S36C6 <sup>(1)</sup>	S35C6 <sup>(1)</sup>	S34C6 <sup>(1)</sup>	S33C6 <sup>(1)</sup>	S32C6 <sup>(1)</sup>	0000
LCDDATA27	05C6	S63C6 <sup>(2)</sup>	S62C6 <sup>(2)</sup>	S61C6 <sup>(2)</sup>	S60C6 <sup>(2)</sup>	S59C6 <sup>(2)</sup>	S58C6 <sup>(2)</sup>	S57C6 <sup>(2)</sup>	S56C6 <sup>(2)</sup>	S55C6 <sup>(2)</sup>	S54C6 <sup>(2)</sup>	S53C6 <sup>(2)</sup>	S52C6 <sup>(2)</sup>	S51C6 <sup>(2)</sup>	S50C6 <sup>(1)</sup>	S49C6 <sup>(2)</sup>	S48C6	0000
LCDDATA28	05C8	S15C7	S14C7	S13C7	S12C7	S11C7	S10C7	S09C7	S08C7	S07C7	S06C7	S05C7	S04C7	S03C7	S02C7	S01C7	S00C7	0000
LCDDATA29	05CA	S31C7	S30C7	S29C7	S28C7	S27C7	S26C7	S25C7	S24C7	S23C7	S22C7	S21C7	S20C7	S19C7	S18C7	S17C7	S16C7	0000
LCDDATA30	05CC	S47C7	S46C7 <sup>(1)</sup>	S45C7 <sup>(1)</sup>	S44C7 <sup>(1)</sup>	S43C7 <sup>(1)</sup>	S42C7 <sup>(1)</sup>	S41C7 <sup>(1)</sup>	S40C7 <sup>(1)</sup>	S39C7 <sup>(1)</sup>	S38C7 <sup>(1)</sup>	S37C7 <sup>(1)</sup>	S36C7 <sup>(1)</sup>	S35C7 <sup>(1)</sup>	S34C7 <sup>(1)</sup>	S33C7 <sup>(1)</sup>	S32C7 <sup>(1)</sup>	0000
LCDDATA31	05CE	S63C7 <sup>(2)</sup>	S62C7 <sup>(2)</sup>	S61C7 <sup>(2)</sup>	S60C7 <sup>(2)</sup>	S59C7 <sup>(2)</sup>	S58C7 <sup>(2)</sup>	S57C7 <sup>(2)</sup>	S56C7 <sup>(2)</sup>	S55C7 <sup>(2)</sup>	S54C7 <sup>(2)</sup>	S53C7 <sup>(2)</sup>	S52C7 <sup>(2)</sup>	S51C7 <sup>(2)</sup>	S50C7 <sup>(1)</sup>	S49C7 <sup>(2)</sup>	S48C7	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚和 80 引脚器件上未实现这些位, 读为 0。

**表 4-25: 并行主 / 从端口寄存器映射**

寄存器名称	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	所有复位时的状态		
PMCON1	0600	PMPEN	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0	CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0	0000		
PMCON2	0602	BUSY	—	ERROR	TIMEOUT	—	—	—	RADDR23	RADDR22	RADDR21	RADDR20	RADDR19	RADDR18	RADDR17	RADDR16	0000			
PMCON3	0604	PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE	—	PTEN22	PTEN21	PTEN20	PTEN19	PTEN18	PTEN17	PTEN16	0000		
PMCON4	0606	PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8	PTEN7	PTEN6	PTEN5	PTEN4	PTEN3	PTEN2	PTEN1	PTEN0	0000		
PMCS1CF	0608	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000		
PMCS1BS	060A	BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16	BASE15	—	—	—	BASE11	—	—	—	0200		
PMCS1MD	060C	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000			
PMCS2CF	060E	CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM	ACKP	PTSZ1	PTSZ0	—	—	—	—	—	0000		
PMCS2BS	0610	BASE23	BASE22	BASE21	BASE20	BASE19	BASE18	BASE17	BASE16	BASE15	—	—	—	BASE11	—	—	—	0600		
PMCS2MD	0612	ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	DWAITB1	DWAITB0	DWAITM3	DWAITM2	DWAITM1	DWAITM0	DWAITE1	DWAITE0	0000			
PMDOUT1	0614	数据输出寄存器 1<15:8>										数据输出寄存器 1<7:0>								xxxxx
PMDOUT2	0616	数据输出寄存器 2<15:8>										数据输出寄存器 2<7:0>								xxxxx
PMDIN1	0618	数据输入寄存器 1<15:8>										数据输入寄存器 1<7:0>								xxxxx
PMDIN2	061A	数据输入寄存器 2<15:8>										数据输入寄存器 2<7:0>								xxxxx
PMSTAT	061C	IBF	IBOV	—	—	IB3F	IB2F	IB1F	IB0F	OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E	008F		

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-26: 实时时钟和日历 (RTCC) 寄存器映射

寄存器名称	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	所有复位时的状态
ALRMVAL	0620																	xxxx
ALCFGGRPT	0622	ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0	ARPT7	ARPT6	ARPT5	ARPT4	ARPT3	ARPT2	ARPT1	ARPT0	0000
RTCVAL	0624																	xxxx
RCFGCAL	0626	RTCEN	—	RTCWREN	RTCSYNC	HALFSEC	RTCOE	RTC PTR1	RTC PTR0	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	注 1
RTCPWC	0628	PWCEN	PWCPOL	PWCPRE	PWSPRE	RTCLK1	RTCLK0	RTCOUT1	RTCOUT0	—	—	—	—	—	—	—	—	注 1

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: POR 时 RCFGCAL 和 RTCPWR 寄存器状态为 0000, 在其他复位时保持不变。

表 4-27: 数据信号调制器 (DSM) 寄存器映射

寄存器名称	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	所有复位时的状态
MDCON	062A	MDEN	—	MDSIDL	—	—	—	—	—	—	MDOE	MDSLR	MDOPOL	—	—	—	MDBIT	0020
MDSRC	062C	—	—	—	—	—	—	—	—	SODIS	—	—	—	MS3	MS2	MS1	MS0	000x
MDCAR	062E	CHODIS	CHPOL	CHSYNC	—	CH3	CH2	CH1	CH0	CLODIS	CLPOL	CLSYNC	—	CL3	CL2	CL1	CL0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

表 4-28: 比较器寄存器映射

寄存器名称	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	所有复位时的状态
CMSTAT	0630	CMIDL	—	—	—		C3EVT	C2EVT	C1EVT	—	—	—	—	C3OUT	C2OUT	C1OUT	0000	
CVRCON	0632	—	—	—	—	—	CVREFP	CVREFM1	CVREFM0	CVREN	CVROE	CVRR	CVRSS	CVR3	CVR2	CVR1	CVR0	0000
CM1CON	0634	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCHO	0000
CM2CON	0636	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCHO	0000
CM3CON	0638	CON	COE	CPOL	—	—	—	CEVT	COUT	EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCHO	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

**表 4-29: CRC 寄存器映射**

寄存器名称	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	所有复位时的状态
CRCCON1	0640	CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0	CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—	0040
CRCCON2	0642	—	—	—	DWIDTH4	DWIDTH3	DWIDTH2	DWIDTH1	DWIDTH0	—	—	—	PLEN4	PLEN3	PLEN2	PLEN1	PLEN0	0000
CRCXORL	0644	X15	X14	X13	X12	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	—	0000
CRCXORH	0646	X31	X30	X29	X28	X27	X26	X25	X24	X23	X22	X21	X20	X19	X18	X17	X16	0000
CRCDATL	0648	CRC 数据输入寄存器低字节												0000				0000
CRCDATH	064A	CRC 数据输入寄存器高字节												0000				0000
CRCWDTL	064C	CRC 结果寄存器低字节												0000				0000
CRCWDATH	064E	CRC 结果寄存器高字节												0000				0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

**表 4-30: 外设引脚选择寄存器映射**

寄存器名称	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	所有复位时的状态
RPINR0	0680	—	—	INT1R5	INT1R4	INT1R3	INT1R2	INT1R1	INT1R0	—	—	—	—	—	—	—	—	3F3F
RPINR1	0682	—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0	—	—	INT2R5	INT2R4	INT2R3	INT2R2	INT2R1	INT2R0	3F3F
RPINR2	0684	—	—	—	—	—	—	—	—	—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0	3F3F
RPINR3	0686	—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0	—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0	3F3F
RPINR4	0688	—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0	—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0	3F3F
RPINR7	068E	—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0	—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0	3F3F
RPINR8	0690	—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0	—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0	3F3F
RPINR9	0692	—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0	—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0	3F3F
RPINR10	0694	—	—	—	—	—	—	—	—	—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0	003F
RPINR11	0696	—	—	OCFBFR5	OCFBFR4	OCFBFR3	OCFBFR2	OCFBFR1	OCFBFR0	—	—	OCFAR5	OCFAR4	OCFAR3	OCFAR2	OCFAR1	OCFAR0	3F3F
RPINR17	06A2	—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0	—	—	—	—	—	—	—	—	3F00
RPINR18	06A4	—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0	—	—	U1RXR5	U1RXR4	U1RXR3	U1RXR2	U1RXR1	U1RXR0	3F3F
RPINR19	06A6	—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0	—	—	U2RXR5	U2RXR4	U2RXR3	U2RXR2	U2RXR1	U2RXR0	3F3F
RPINR20	06A8	—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0	—	—	SDI1R5	SDI1R4	SDI1R3	SDI1R2	SDI1R1	SDI1R0	3F3F
RPINR21	06AA	—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0	—	—	SS1R5	SS1R4	SS1R3	SS1R2	SS1R1	SS1R0	3F3F
RPINR22	06AC	—	—	SCK2R5	SCK2R4	SCK2R3	SCK2R2	SCK2R1	SCK2R0	—	—	SDI2R5	SDI2R4	SDI2R3	SDI2R2	SDI2R1	SDI2R0	3F3F
RPINR23	06AE	—	—	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0	—	—	SS2R5	SS2R4	SS2R3	SS2R2	SS2R1	SS2R0	003F
RPINR27	06B6	—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0	—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0	3F3F
RPINR30	06BC	—	—	—	—	—	—	—	—	—	—	MDMIR5	MDMIR4	MDMIR3	MDMIR2	MDMIR1	MDMIR0	003F
RPINR31	06BE	—	—	MDC2R5	MDC2R4	MDC2R3	MDC2R2	MDC2R1	MDC2R0	—	—	MDC1R5	MDC1R4	MDC1R3	MDC1R2	MDC1R1	MDC1R0	3F3F

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

注 2: 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

表 4-30: 外设引脚选择寄存器映射 (续)

寄存器名称	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	所有复位时的状态
RPOR0	06C0	—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0	—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0	0000
RPOR1	06C2	—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0	—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0	0000
RPOR2	06C4	—	—	RP5R5 <sup>(1)</sup>	RP5R4 <sup>(1)</sup>	RP5R3 <sup>(1)</sup>	RP5R2 <sup>(1)</sup>	RP5R1 <sup>(1)</sup>	RP5R0 <sup>(1)</sup>	—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0	0000
RPOR3	06C6	—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0	—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0	0000
RPOR4	06C8	—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0	—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0	0000
RPOR5	06CA	—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0	—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0	0000
RPOR6	06CC	—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0	—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0	0000
RPOR7	06CE	—	—	RP15R5 <sup>(1)</sup>	RP15R4 <sup>(1)</sup>	RP15R3 <sup>(1)</sup>	RP15R2 <sup>(1)</sup>	RP15R1 <sup>(1)</sup>	RP15R0 <sup>(1)</sup>	—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0	0000
RPOR8	06D0	—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0	—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0	0000
RPOR9	06D2	—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0	—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0	0000
RPOR10	06D4	—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0	—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0	0000
RPOR11	06D6	—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0	—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0	0000
RPOR12	06D8	—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0	—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0	0000
RPOR13	06DA	—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0	—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0	0000
RPOR14	06DC	—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0	—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0	0000
RPOR15	06DE	—	—	RP31R5 <sup>(2)</sup>	RP31R4 <sup>(2)</sup>	RP31R3 <sup>(2)</sup>	RP31R2 <sup>(2)</sup>	RP31R1 <sup>(2)</sup>	RP31R0 <sup>(2)</sup>	—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0	0000

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 在 64 引脚器件上未实现这些位, 读为 0。

2: 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

表 4-31: 系统控制 (时钟和复位) 寄存器映射

寄存器名称	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	所有复位时的状态
RCON	0740	TRAPR	IOPUWR	—	RETEN	—	DPSLP	CM	VREGS	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	BOR	POR	注 1
OSCCON	0742	—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0	CLKLOCK	IOLOCK	LOCK	—	CF	POSCEN	SOSCEN	OSWEN	注 2
CLKDIV	0744	ROI	DOZE2	DOZE1	DOZE0	DOZEN	RCDIV2	RCDIV1	RCDIV0	—	—	—	—	—	—	—	3100	
OSCTUN	0748	—	—	—	—	—	—	—	—	—	—	TUN5	TUN4	TUN3	TUN2	TUN1	TUN0	0000
REFOCON	074E	ROEN	—	ROSSLP	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0	—	—	—	—	—	—	—	0000	
LVDLCON	0756	LVDEN	—	LSIDL	—	—	—	—	—	DIR	BGVST	IRVST	—	LVDL3	LVDL2	LVDL1	LVDL0	0000
RCON2	0762	—	—	—	—	—	—	—	—	—	—	—	r	VDDBOR	VDDPOR	VBPOR	VBAT	注 1

图注: — = 未实现, 读为 0; r = 保留。复位值以十六进制格式显示。

注 1: RCON 寄存器的复位值与复位事件的类型有关。更多信息, 请参见第 7.0 节“复位”。

2: OSCCON 寄存器的复位值与复位事件的类型以及器件配置有关。更多信息, 请参见第 9.0 节“振荡器配置”。

**表 4-32:** 深度休眠寄存器映射

寄存器名称	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	所有复位时的状态
DSCON	0758	DSEN	—	—	—	—	—	—	—	—	—	—	—	r	DSBOR	RELEASE	0000 <sup>(1)</sup>	
DSWAKE	075A	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	—	0000 <sup>(1)</sup>	
DSGPR0	075C	深度休眠信号数据 0														0000 <sup>(1)</sup>		
DSGPR1	075E	深度休眠信号数据 1														0000 <sup>(1)</sup>		

图注: — = 未实现, 读为 0; r = 保留。复位值以十六进制格式显示。

注 1: 仅在发生 VDD 上电复位事件时复位这些寄存器。

**表 4-33:** NVM 寄存器映射

寄存器名称	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	所有复位时的状态
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	—	—	ERASE	—	—	NVMOP3	NVMOP2	NVMOP1	NVMOP0	0000 <sup>(1)</sup>
NVMKEY	0766	—	—	—	—	—	—	—	—	—	NVMKEY 寄存器 <7:0>						0000	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

注 1: 显示的复位值仅针对上电复位。其他复位状态的值取决于复位时存储器写操作或擦除操作的状态。

**表 4-34:** PMD 寄存器映射

寄存器名称	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	所有复位时的状态
PMD1	0770	T5MD	T4MD	T3MD	T2MD	T1MD	—	—	I2C1MD	U2MD	U1MD	SPI2MD	SPI1MD	—	—	ADC1MD	0000	
PMD2	0772	—	IC7MD	IC6MD	IC5MD	IC4MD	IC3MD	IC2MD	IC1MD	—	OC7MD	OC6MD	OC5MD	OC4MD	OC3MD	OC2MD	OC1MD	0000
PMD3	0774	—	—	—	—	DSMMD	CMPMD	RTCCMD	PMPMD	CRCMD	—	—	—	U3MD	—	I2C2MD	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	UPWMMMD	U4MD	—	REFOMD	CTMUMD	LVDMD	—	0000	
PMD6	077A	—	—	—	—	—	—	—	—	LCDMD	—	—	—	—	—	SPI3MD	0000	
PMD7	077C	—	—	—	—	—	—	—	—	—	DMA1MD	DMA0MD	—	—	—	—	0000	

图注: — = 未实现, 读为 0。复位值以十六进制格式显示。

## 4.2.5 扩展数据空间 (EDS)

扩展数据空间 (EDS) 使 PIC24F 器件能够对更大的数据范围进行寻址，否则只可对 16 位地址范围进行寻址。EDS 包括低 32 KB 数据地址空间不可直接访问的任何其他内部数据存储空间和任何通过 EPMP 访问的外部存储空间。

此外，EDS 还允许对程序存储空间进行读访问。此功能称为程序空间可视性 (Program Space Visibility, PSV)，第 4.3.3 节“使用 EDS 从程序存储器读数据”中有详细介绍。

图 4-4 显示了整个 EDS 空间。EDS 由页组成，称为 EDS 页，单页大小等于 EDS 窗口的大小 (32 KB)。通过数据空间读寄存器 (DSRPAG) 或数据空间写寄存器 (DSWPAG) 选择特定 EDS 页。对于 PSV，仅使用 DSRPAG 寄存器。DSRPAG 寄存器的值和 16 位宽数据地址的组合构成了 24 位有效地址 (Effective Address, EA)。

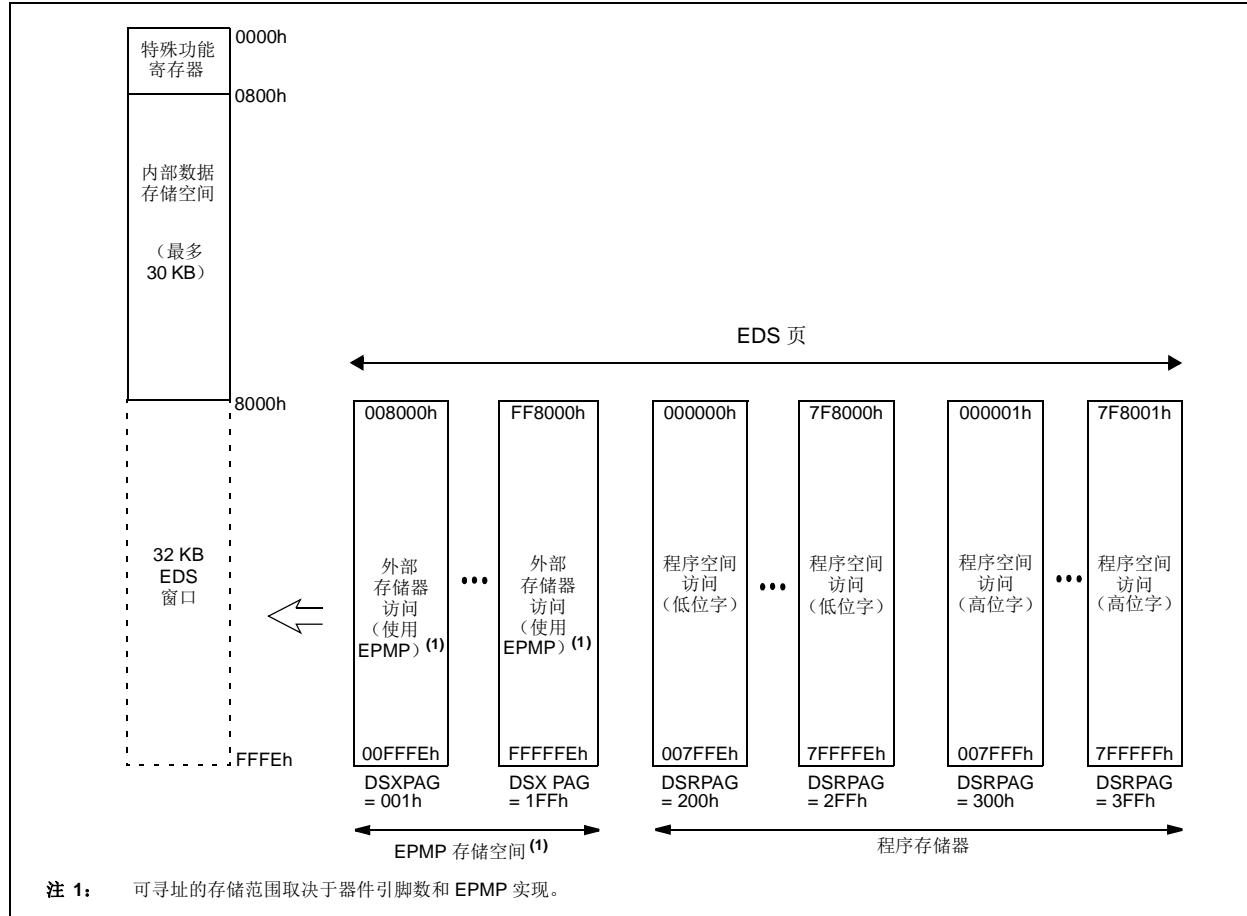
PIC24FJ128GA310 系列器件的数据寻址范围取决于特定器件上实现的增强型并行主端口版本；同时这也是器件引脚数的函数。表 4-35 列出了此系列中各器件可访问的总存储空间。有关使用 EPMP 访问外部存储空间的更多详细信息，请参见《PIC24F 系列参考手册》，第 42 章“增强型并行主端口 (EPMP)”(DS39730A\_CN)。

表 4-35：可访问的总数据存储空间

系列	内部 RAM	外部 RAM 访问 (使用 EPMP)
PIC24FJXXXGA310	8K	最多 16 MB
PIC24FJXXXGA308	8K	最多 64K
PIC24FJXXXGA306	8K	最多 64K

注：访问 EDS 窗口的 Page 0 将生成地址错误陷阱，因为 Page 0 是基础数据存储器 (低数据空间中的数据存储单元 0800h 至 7FFFh)。

图 4-4：扩展数据空间



## 4.2.5.1 从 EDS 读取数据

为了从 EDS 空间读取数据，首先，通过将所需的 EDS 页数装入到 DSRPAG 寄存器并将偏移地址分配给其中一个 W 寄存器来设置地址指针。一旦完成上述分配任务，通过将分配了偏移地址的工作寄存器的 bit 15 置 1 使能 EDS 窗口；然后，将读取所指 EDS 存储单元的内容。

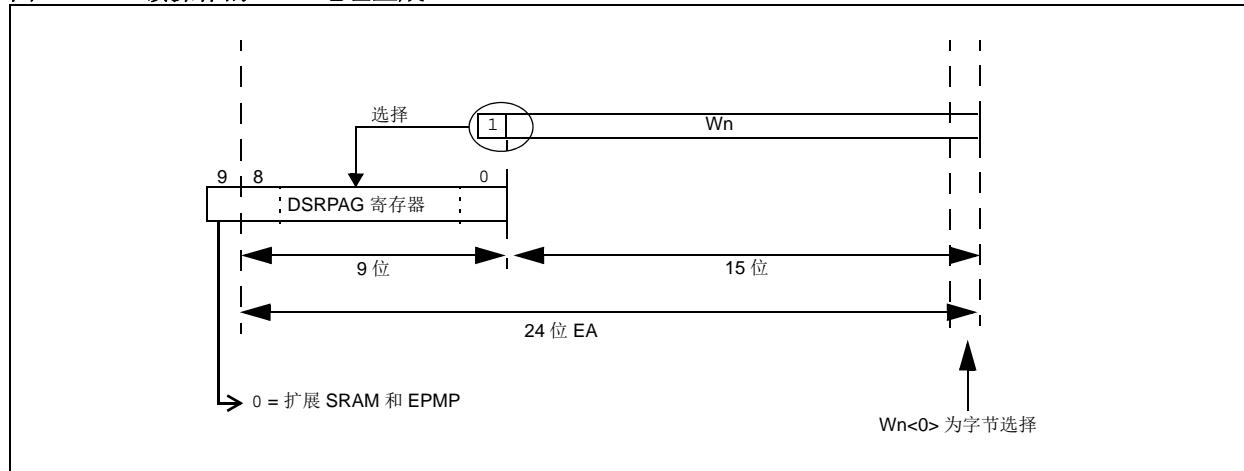
图 4-5 描述了如何为读操作生成 EDS 空间地址。

当 EA 的最高有效位 (MSb) 为 1 且 DSRPAG<9> = 0 时，DSRPAG 的低 9 位与 EA 的低 15 位共同组成一个 24 位 EDS 空间地址以进行读操作。

例 4-1 显示如何从 EDS 读取一个字节、字和双字。

**注：** 从 EDS 空间的所有读操作均有一个指令周期的开销。因此，完成一个 EDS 读操作最少需要两个指令周期。EDS 在 REPEAT 指令下进行读操作，前两个访问需要三个周期，而后续的访问需要 1 个周期。

图 4-5：读操作的 EDS 地址生成



## 例 4-1：EDS 读代码（汇编语言）

```
; Set the EDS page from where the data to be read
    mov      #0x0002 , w0
    mov      w0 , DSRPAG      ;page 2 is selected for read
    mov      #0x800 , w1      ;select the location (0x800) to be read
    bset    w1 , #15          ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
    mov.b   [w1++], w2      ;read Low byte
    mov.b   [w1++], w3      ;read High byte

;Read a word from the selected location
    mov      [w1] , w2        ;

;Read Double - word from the selected location
    mov.d   [w1] , w2        ;two word read, stored in w2 and w3
```

## 4.2.5.2 数据写入 EDS

要将数据写入 EDS 空间，与 EDS 读操作类似，通过将所需的 EDS 页数装入到 DSWPAG 寄存器并将偏移地址分配给其中一个 W 寄存器来设置地址指针。一旦完成上述的分配任务，通过将分配了偏移地址的工作寄存器的 bit 15 置 1 使能 EDS 窗口，从而写入所访问的地址单元。

图 4-6 描述了如何为写操作生成 EDS 空间地址。

当 EA 的 MSb 为 1 时，DSWPAG 的低 9 位与 EA 的低 15 位共同组成一个 24 位 EDS 地址以进行写操作。

例 4-2 显示了如何将一个字节、字或双字写入 EDS。

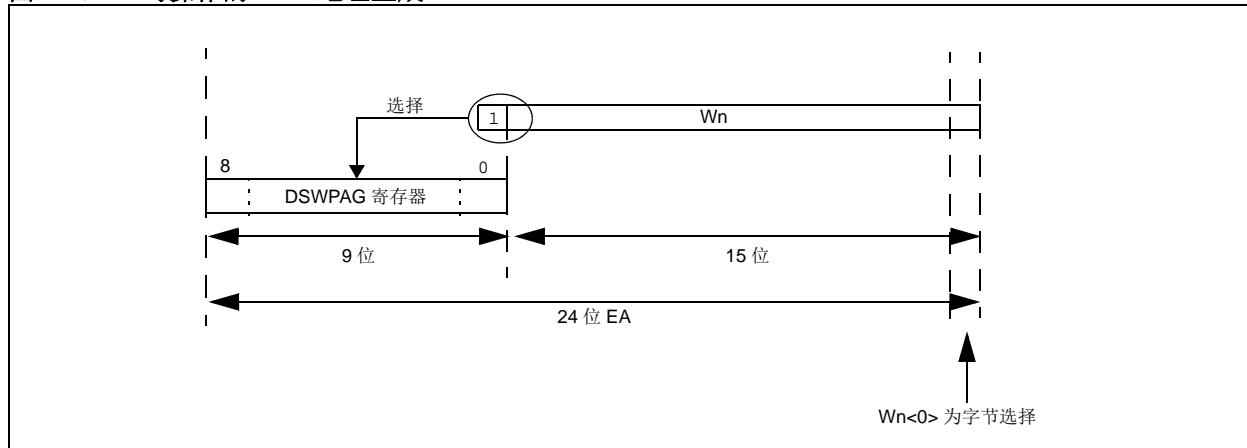
页寄存器 (DSRPAG/DSWPAG) 在跨页边界（发生从 0xFFFF 至 0x8000 的计满返回）时不会自动更新。用汇编语言开发代码时，在地址指针跨页边界时更新页寄存器必须注意。C 编译器跟踪寻址，并在访问连续数据存储单元时相应增加或减少页寄存器值。

**注 1:** 所有写 EDS 操作均在一个单周期内执行。

**2:** 不支持在 REPEAT 指令下对任何 EDS 存储单元进行读 / 修改 / 写操作。例如，BCLR、BSW、BTG、RLC f、RLNC f、RRC f、RRNC f、ADD f、SUB f、SUBR f、AND f、IOR f、XOR f、ASR f 和 ASL f。

**3:** 在执行读/修改/写操作时，使用 DSRPAG 寄存器。

图 4-6：写操作的 EDS 地址生成



例 4-2: EDS 写代码 (汇编语言)

```
; Set the EDS page where the data to be written
mov    #0x0002 , w0
mov    w0 , DSWPAG ;page 2 is selected for write
mov    #0x0800 , w1 ;select the location (0x800) to be written
bset   w1 , #15      ;set the MSB of the base address, enable EDS mode

;Write a byte to the selected location
mov    #0x00A5 , w2
mov    #0x003C , w3
mov.b  w2 , [w1++]    ;write Low byte
mov.b  w3 , [w1++]    ;write High byte

;Write a word to the selected location
mov    #0x1234 , w2 ;
mov    w2 , [w1] ;

;Write a Double - word to the selected location
mov    #0x1122 , w2
mov    #0x4455 , w3
mov.d  w2 , [w1]      ;2 EDS writes
```

表 4-36： 具有不同页和地址的 EDS 存储地址

DSRPAG (数据空间 读寄存器)	DSWPAG (数据空间 写寄存器)	间接寻址时的源/ 目标地址	指向 EDS 的 24 位 EA	备注
x <sup>(1)</sup>	x <sup>(1)</sup>	0000h 至 1FFFh	000000h 至 001FFFh	Near 数据空间 <sup>(2)</sup>
		2000h 至 7FFFh	002000h 至 007FFFh	
001h	001h	8000h 至 FFFFh	008000h 至 00FFFEh	EPMP 存储空间
002h	002h		010000h 至 017FFEh	
003h	003h		018000h 至 0187FEh	
•	•		•	
•	•		•	
•	•		•	
•	•		FF8000h 至 FFFFFEh	
1FFh	1FFh			
000h	000h		无效地址	地址错误陷阱 <sup>(3)</sup>

注 1：如果源 / 目标地址低于 8000h，将忽略 DSRPAG 和 DSWPAG 寄存器。

2：该数据空间也可通过直接寻址访问。

3：当源 / 目标地址高于 8000h 且 DSRPAG/DSWPAG 为 0 时，将发生地址错误陷阱。

## 4.2.6 软件堆栈

除了用作工作寄存器外，PIC24F 器件中的 W15 寄存器也可用作软件堆栈指针（Software Stack Pointer，SSP）。此指针总是指向第一个可用的空字，从低地址向高地址方向增长。它在弹出堆栈之前递减，而在压入堆栈后递增，如图4-7所示。注意，对于执行任何CALL指令时的PC压栈操作，在压入堆栈之前，PC的MSB要进行零扩展，从而确保 MSB 始终清零。

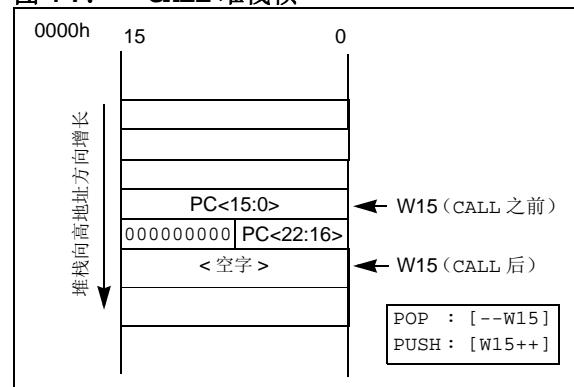
**注：** 在异常处理期间，在 PC 压入堆栈之前，要先将 PC 的 MSB 与 SRL 寄存器组合在一起。

堆栈指针限制值寄存器（SPLIM）与堆栈指针相关联，它设置堆栈上边界的地址。SPLIM在复位时不会被初始化。与堆栈指针一样，SPLIM<0> 被强制为 0，因为所有的堆栈操作必须是字对齐的。每当使用 W15 作为源指针或目标指针产生有效地址时，有效地址会与SPLIM 中的值进行比较。如果堆栈指针（W15）的内容与 SPLIM寄存器的内容相等，则会执行压栈操作而不产生堆栈错误陷阱。但在随后的压栈操作时将会产生堆栈错误陷阱。因此，例如如果想要在堆栈增长到超过 RAM 中的地址 2000h 时产生堆栈错误陷阱，则需用值 1FFEh 来初始化 SPLIM。

同理，当堆栈指针地址小于 0800h 时，将产生堆栈指针下溢（堆栈错误）陷阱。这可防止堆栈进入特殊功能寄存器空间。

在对 SPLIM 寄存器进行写操作后，不应紧跟使用 W15 进行间接读操作的指令。

图 4-7： CALL 堆栈帧



## 4.3 程序存储空间与数据存储空间的接口

PIC24F 架构采用 24 位宽的程序空间和 16 位宽的数据空间。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。要成功使用该数据，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，PIC24F 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新映射到数据空间（程序空间可视性）

表指令允许应用程序读写程序存储器中的一小块区域。这一功能对于访问需要随时更新的数据表来说非常理想。也可通过表操作访问一个程序字的所有字节。重映射方式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。这一方式只能访问程序字的低位字。

### 4.3.1 对程序空间进行寻址

由于数据和程序空间的地址范围分别为 16 位和 24 位，因此需要一个从 16 位数据寄存器创建一个 23 位或 24 位程序地址的方法。方法取决于所采用的接口方式。

对于表操作，使用 8 位的表存储器页地址寄存器（TBLPAG）定义程序空间中一个 32K 字的区域。TBLPAG 寄存器的 8 位与 16 位 EA 组合形成了一个完整的 24 位程序空间地址。在这种地址形式下，TBLPAG 的 MSb 用来决定操作是发生在用户存储区（TBLPAG<7> = 0）中还是配置存储区（TBLPAG<7> = 1）中。

对于重映射操作，使用 10 位的扩展数据空间读寄存器（DSRPAG）定义程序空间中的 16K 字页。当 EA 的最高位（MSb）为 1 时且 DSRPAG 的 MSb（bit 9）为 1 时，DSRPAG 的低 8 位与 EA 的低 15 位组合形成一个 23 位的程序空间地址。DSRPAG<8>位决定是映射程序存储空间的低字位（当该位为 0 时）还是高字位（当该位为 1 时）。与表操作不同，重映射操作被严格限制在用户存储区中。

表 4-37 和图 4-8 显示了如何从数据 EA 创建程序 EA 来进行表操作和重映射访问。本文中，P<23:0> 指的是程序空间字，而 D<15:0> 指的是数据空间字。

表 4-37：程序空间地址构成

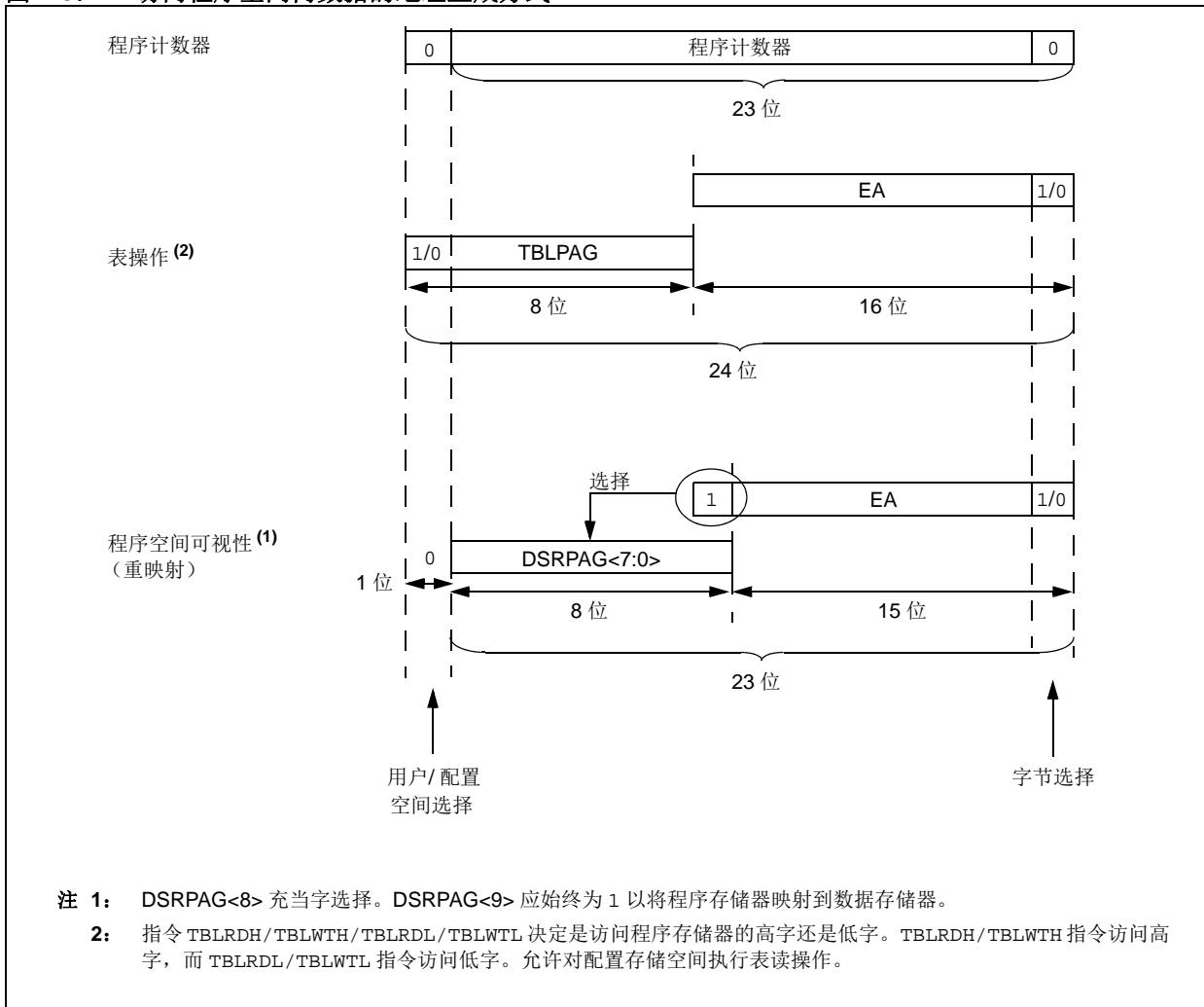
访问类型	访问空间	程序空间地址						
		<23>	<22:16>	<15>	<14:1>	<0>		
通过指令访问 (代码执行)	用户	0	PC<22:1>			0		
		0xx xxxx xxxx xxxx xxxx xxxx xxxx0						
TBLRD/TBLWT (读/写字节或字)	用户	TBLPAG<7:0>		数据 EA<15:0>				
		0xxx xxxx		xxxx xxxx xxxx xxxx				
	配置	TBLPAG<7:0>		数据 EA<15:0>				
		1xxx xxxx		xxxx xxxx xxxx xxxx				
程序空间可视性 (块重映射/读)	用户	0	DSRPAG<7:0> <sup>(2)</sup>		数据 EA<14:0> <sup>(1)</sup>			
		0	xxxx xxxx		xxx xxxx xxxx xxxx			

注 1：在这种情况下，数据 EA<15> 始终为 1，但并不用它来计算程序空间地址。地址的 bit 15 为 DSRPAG<0>。

2：在这种情况下，DSRPAG<9> 始终为 1。DSRPAG<8> 决定读取的是程序存储空间的低字位还是高字位。当 DSRPAG<8> 为 0，读低字，而为 1 时，读高字。

# PIC24FJ128GA310 系列

图 4-8：访问程序空间内数据的地址生成方式



### 4.3.2 使用表指令访问程序存储器中的数据

TBLRDL 和 TBLWTL 指令提供了读或写程序空间内任何地址的低位字的直接方法，无需通过数据空间。TBLRDH 和 TBLWTH 指令是可将一个程序空间字的高 8 位作为数据读写的惟一方法。

对于每个连续的 24 位程序字，PC 的递增量为 2。这使得程序存储器地址能够被直接映射到数据空间地址中。于是，程序存储器可以被看作是两个 16 位字宽的地址空间，它们并排放置，具有相同的地址范围。TBLRDL 和 TBLWTL 访问包含数据低位字的空间，TBLRDH 和 TBLWTH 访问包含高数据字节的空间。

提供了两条表指令来对程序空间执行字节或字（16 位）大小的数据读写。读和写都可以采用字节或字操作的形式。

1. TBLRDL（表读低位字）：在字模式下，该指令将程序空间地址的低位字（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。

2. TBLRDH（表读高位字）：在字模式下，该指令将程序地址的整个高位字（P<23:16>）映射到数据地址中。注意，D<15:8> 为“虚拟”字节，始终为 0。

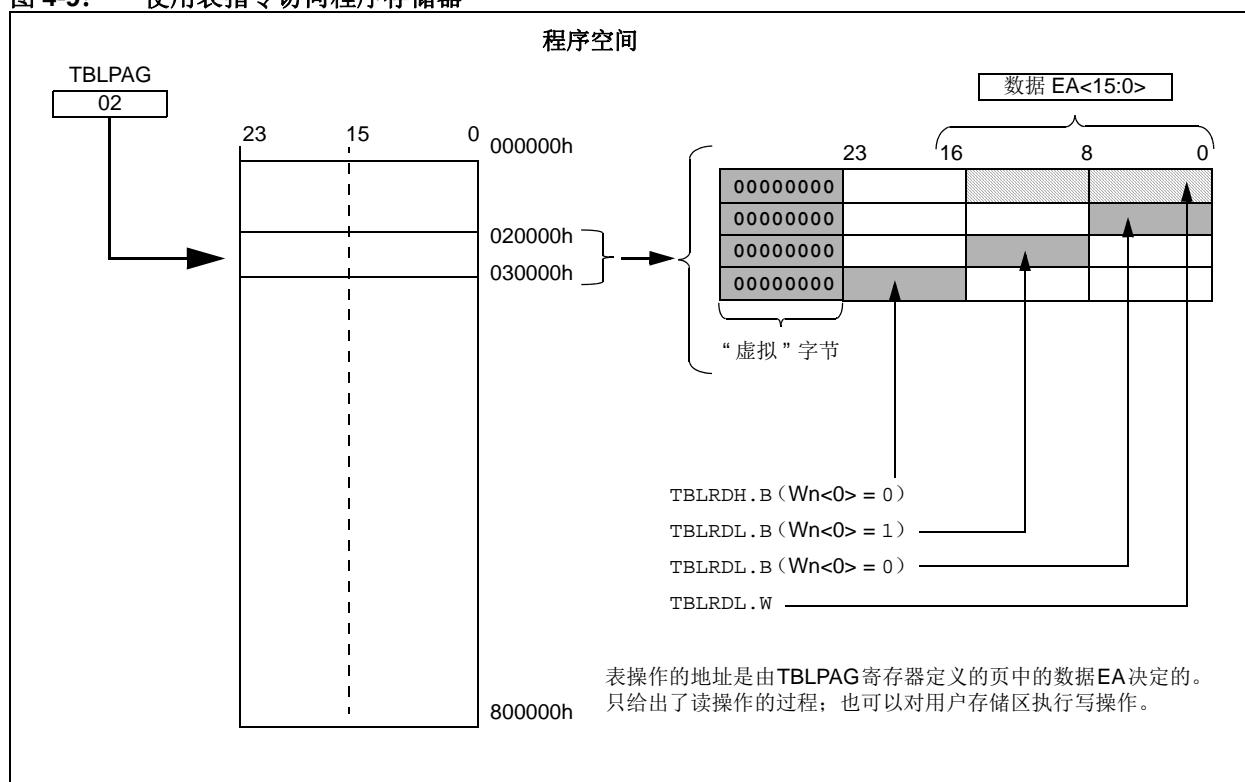
在字节模式下，程序字的高字节或低字节被映射到数据地址的 D<7:0> 中，这和 TBLRDL 指令相同。注意当选择高位“虚拟”字节（字节选择位 = 1）时，数据将始终为 0。

表指令 TBLWTH 和 TBLWTL 以类似的方式向程序地址空间写入各字节或字。[第 6.0 节“闪存程序存储器”](#) 对这两条指令的详细操作给出了说明。

对于所有的表操作，要访问程序存储空间的哪个区域是由表存储器页地址寄存器（TBLPAG）决定的。TBLPAG 可寻址器件的整个程序存储空间，包括用户空间和配置空间。当 TBLPAG<7> = 0 时，表页位于用户存储空间中。当 TBLPAG<7> = 1 时，表页位于配置空间中。

**注：**仅表读操作可在存放器件 ID 的配置存储空间内执行。不允许表写操作。

图 4-9：使用表指令访问程序存储器



### 4.3.3 使用 EDS 从程序存储器读数据

可选择将数据空间的高 32 KB 映射到程序空间中的任一 16K 字页中。这提供了通过数据空间对存储的常量数据的透明访问，而无需使用特殊指令（即 TBLRDL/H 指令）。

当 EA 的 MSb 为 1 且 DSRPAG<9> 也为 1 时，就可以通过数据空间访问程序空间。DSRPAG 的低 8 位与 Wn<14:0> 组合形成了一个 23 位 EA 来访问程序存储器。DSRPAG<8> 决定应访问哪个字；当该位为 0 时，访问低位字，而当该位为 1 时，访问程序存储器的高位字。

整个程序存储器分为 512 个 EDS 页，从 200h 至 3FFh，每个页由 16K 字数据组成。页 200h 至 2FFh 对应程序存储器的低位字，而 300h 至 3FFh 对应程序存储器的高位字。

使用 EDS 技术，可访问整个程序存储器。此前，不支持对程序存储器高位字的访问。

表 4-38 提供了与 EDS 页和原地址对应的程序存储器的 23 位 EDS 地址。

对于使用 PSV 而又在 REPEAT 循环之外执行的指令，MOV 和 MOV.D 指令除了规定的执行时间之外，还需要一个额外的指令周期。其他所有的指令，除了规定的指令执行时间之外，还需要两个额外的指令周期。

对于使用 PSV 而又在 REPEAT 循环内执行的指令，下列情况，除了规定的指令执行时间之外，需要两个额外的指令周期：

- 在第一次迭代中执行的指令
- 在最后一次迭代中执行的指令
- 由于中断而退出循环之前执行的指令
- 中断得到处理后再次进入循环时执行的指令

REPEAT 循环的所有其他各次迭代，都允许使用 PSV 访问数据的指令在一个周期内执行。

表 4-38：具有不同页和地址的 EDS 程序地址

DSRPAG (数据空间读寄存器)	间接寻址时的源地址	指向 EDS 的 23 位 EA	备注
200h	8000h 至 FFFFh	000000h 至 007FFEh	仅用于读操作的低 4M 字程序指令（8 MB）
·		·	
·		·	
2FFh		7F8000h 至 7FFFFFFh	
300h	8000h 至 FFFFh	000001h 至 007FFFh	仅限于读操作的高 4M 字程序指令（4 MB 保留；4 MB 为虚拟字节）。
·		·	
·		·	
3FFh		7F8001h 至 7FFFFFFh	
000h		无效地址	地址错误陷阱①

注 1：当源 / 目标地址高于 8000h 且 DSRPAG/DSWPAG 为 0 时，将发生地址错误陷阱。

### 例 4-3：从程序存储器进行 EDS 读操作的代码（汇编语言）

```
; Set the EDS page from where the data to be read
    mov    #0x0202 , w0
    mov    w0 , DSRPAG           ;page 0x202, consisting lower words, is selected for read
    mov    #0x000A , w1           ;select the location (0xA) to be read
    bset   w1 , #15              ;set the MSB of the base address, enable EDS mode

;Read a byte from the selected location
    mov.b  [w1++], w2            ;read Low byte
    mov.b  [w1++], w3            ;read High byte

;Read a word from the selected location
    mov    [w1] , w2             ;
;Read Double - word from the selected location
    mov.d  [w1] , w2             ;two word read, stored in w2 and w3
```

图 4-10：访问低位字的程序空间可视性

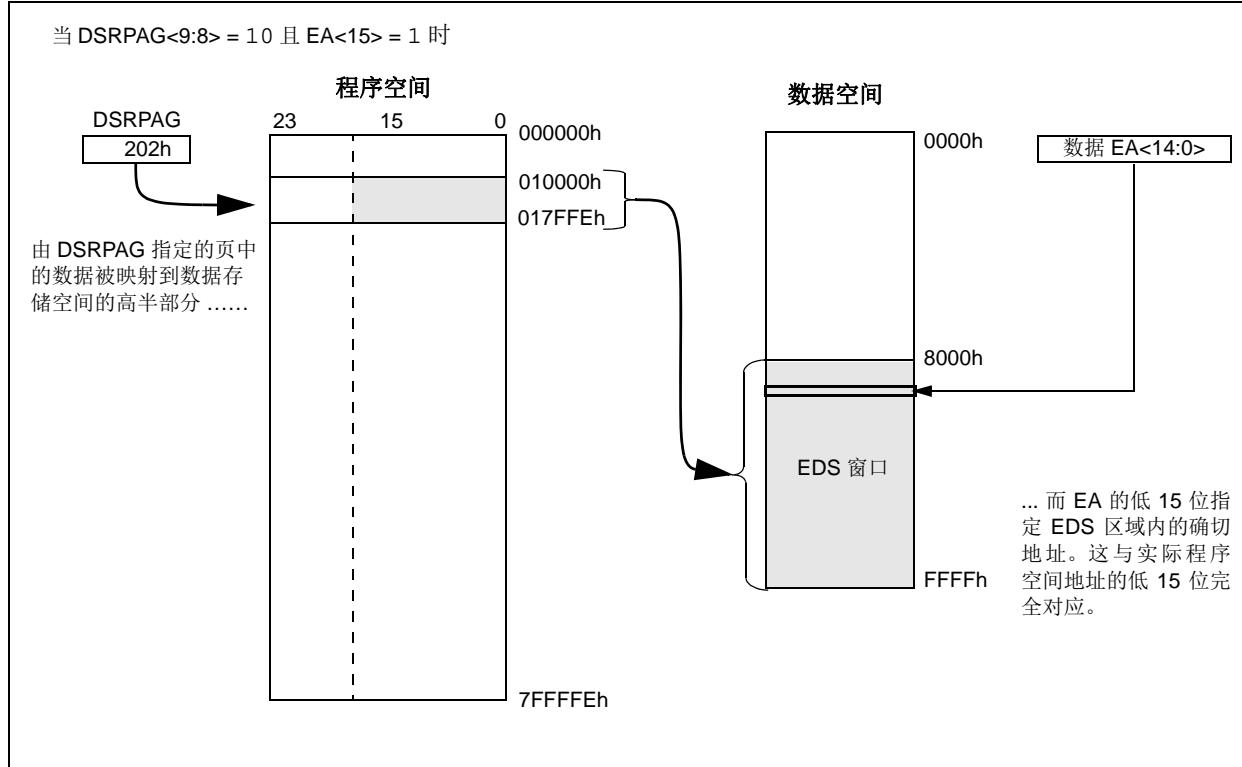
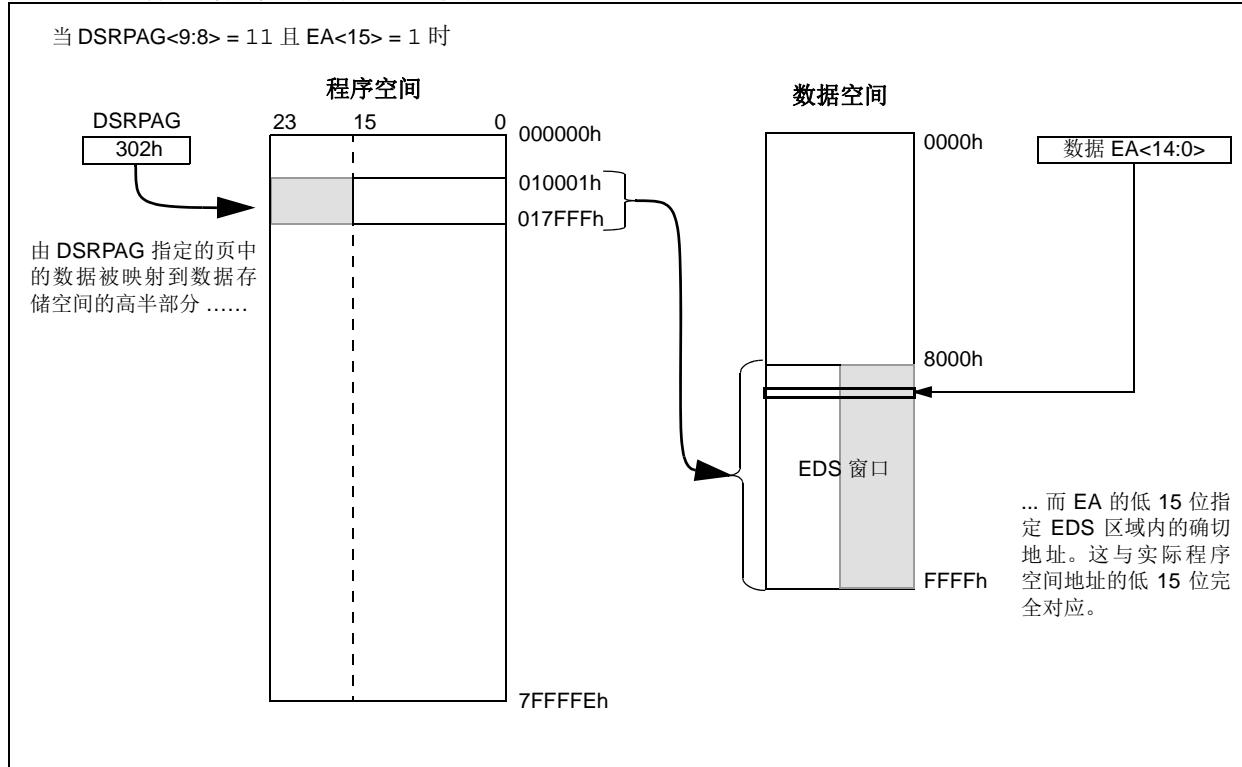


图 4-11：访问高位字的程序空间可视性



# **PIC24FJ128GA310 系列**

---

---

注:

## 5.0 直接存储器访问控制器 (DMA)

**注:** 本数据手册总结了 PIC24FJ128GA310 系列器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关本数据手册的补充信息, 请参见《PIC24F 系列参考手册》中的第 54 章 “直接存储器访问控制器 (DMA)” (DS39742A\_CN)。本数据手册中的信息将取代 FRM 中的信息。

直接存储器访问控制器 (Direct Memory Access Controller, DMA) 旨在为工作在 SFR 总线上的高数据吞吐量外设提供服务, 使这些外设可直接访问数据存储器, 而降低对 CPU 密集型管理的需要。通过使这些数据密集型外设共享各自的数据路径, 也可减轻主数据总线负荷, 从而进一步降低功耗。

DMA 控制器充当外设及 CPU 的直接扩展。DMA 控制器位于 CPU 和使能 DMA 的外设之间的单片机数据总线上, 可直接访问 SRAM。这将 SFR 总线分为两个总线, 使 DMA 控制器可访问位于新的 DMA SFR 总线上具有 DMA 功能的外设。该控制器充当 DMA SFR 总线上的主器件, 控制具有 DMA 功能的外设的数据流。

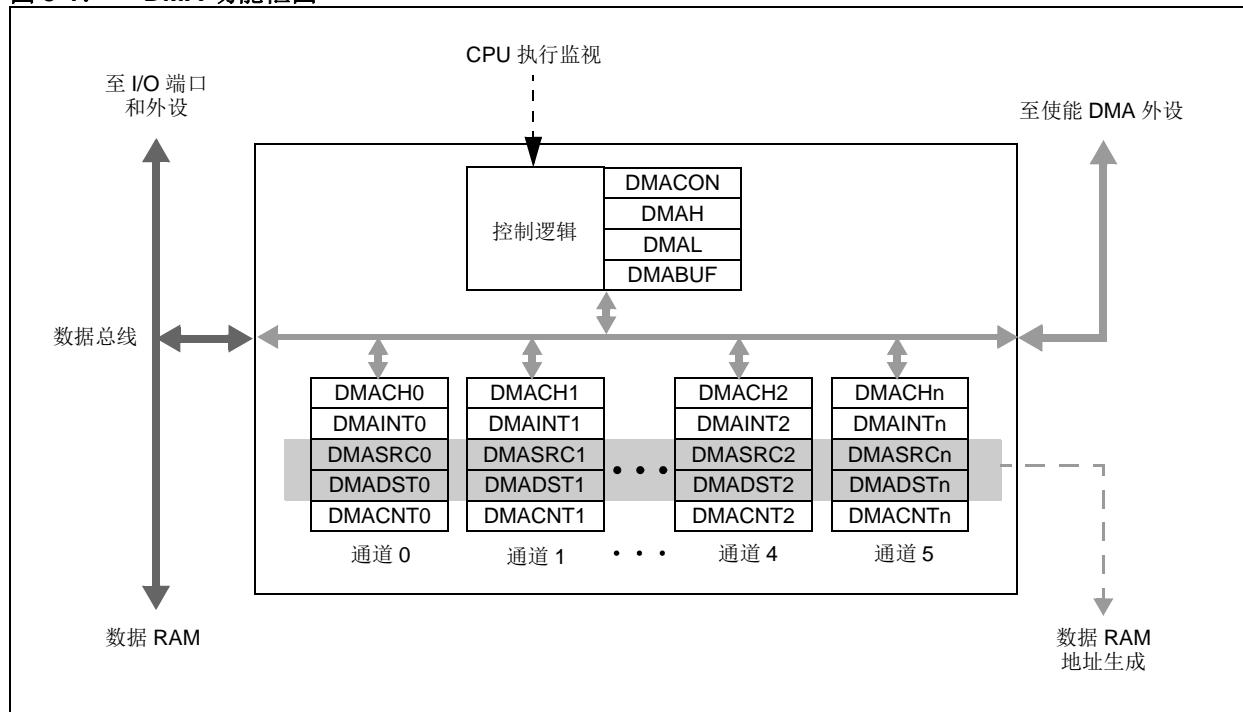
该控制器还直接监视 CPU 指令处理, 使其知道何时 CPU 需要访问 DMA 总线上的外设, 并根据需要自动放弃对 CPU 的控制。这会增加数据处理有效带宽, 而 DMA 操作又不会造成处理器停顿。这使得控制器对用户基本透明。

DMA 控制器具有以下特性:

- 6 条独立的可独立编程通道
- 与 CPU 并行操作 (无 DMA 导致的等待状态)
- DMA 总线仲裁
- 5 个可编程地址模式
- 4 个可编程传输模式
- 4 个灵活的内部数据传输模式
- 支持字节或字数据传输
- 每个通道具有 16 位源和目标地址寄存器, 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
- 高 / 低地址限制寄存器
- 计数器半满中断
- 软件触发传输
- 用于对称缓冲操作的空写模式

图 5-1 显示了 DMA 控制器的简化框图。

图 5-1: DMA 功能框图



## 5.1 DMA 操作汇总

DMA 控制器可根据多个不同的参数在各地址间传输数据。可为任一事务独立配置这些参数中的每一个，此外，任一或所有 DMA 通道均可同时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，带或不带地址递增 / 递减）

此外，DMA控制器还为所有通道提供通道优先级仲裁。

### 5.1.1 源和目标

使用 DMA 控制器，数据可在数据空间中任意两个地址之间传输。SFR 空间（0000h 至 07FFh）或数据 RAM 空间（0800h 或 FFFFh）既可作为源，又可作为目标。数据可在这些区域之间双向移动，或在任一区域内的各地址之间移动。[图 5-2](#) 给出了四种不同的组合。

如有必要保护数据 RAM 的区域，DMA 控制器允许用户为 SFR 空间之外的数据空间操作设置高/低地址边界。边界由 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，则终止事务并产生中断。

### 5.1.2 数据大小

DMA 控制器可处理 8 位和 16 位事务。用户可使用 SIZE 位（DMACHn<1>）选择大小。默认情况下，每条通道都配置为字大小事务。选择字节大小事务时，源和 / 或目标地址的 Lsb 确定数据是否代表数据 RAM 单元的高字节或低字节。

### 5.1.3 触发源

DMA 控制器可使用器件的 60 个中断源中的任何一个来发起事务。DMA 触发源以其自然中断优先级的反向顺序列出，如[表 5-1](#) 所示。

由于任何事务的源和目标地址均可独立于触发源进行编程，所以 DMA 控制器可使用任何触发器对任何外设执行操作。这还可使用级联的 DMA 通道执行更复杂的传输操作。

### 5.1.4 传输模式

根据每次触发时要传输的数据量，DMA 控制器支持四种类型的数据传输。

- 单次：每次触发时发生一个事务。
- 连续：每次触发时发生一系列连续事务，事务数量由 DMACNT 事务计数器确定。
- 重复单次：重复执行一个事务，每次触发执行一次，直到 DMA 通道不可用。
- 重复连续：重复执行一系列事务，每次触发执行一轮，直到 DMA 通道不可用。

所有传输模式都允许选择在事务完成后自动重载源和目标地址与计数器值。重复传输模式自动完成此操作。

### 5.1.5 寻址模式

DMA 控制器还支持在单独地址或地址范围之间进行传输。这四种基本选项是：

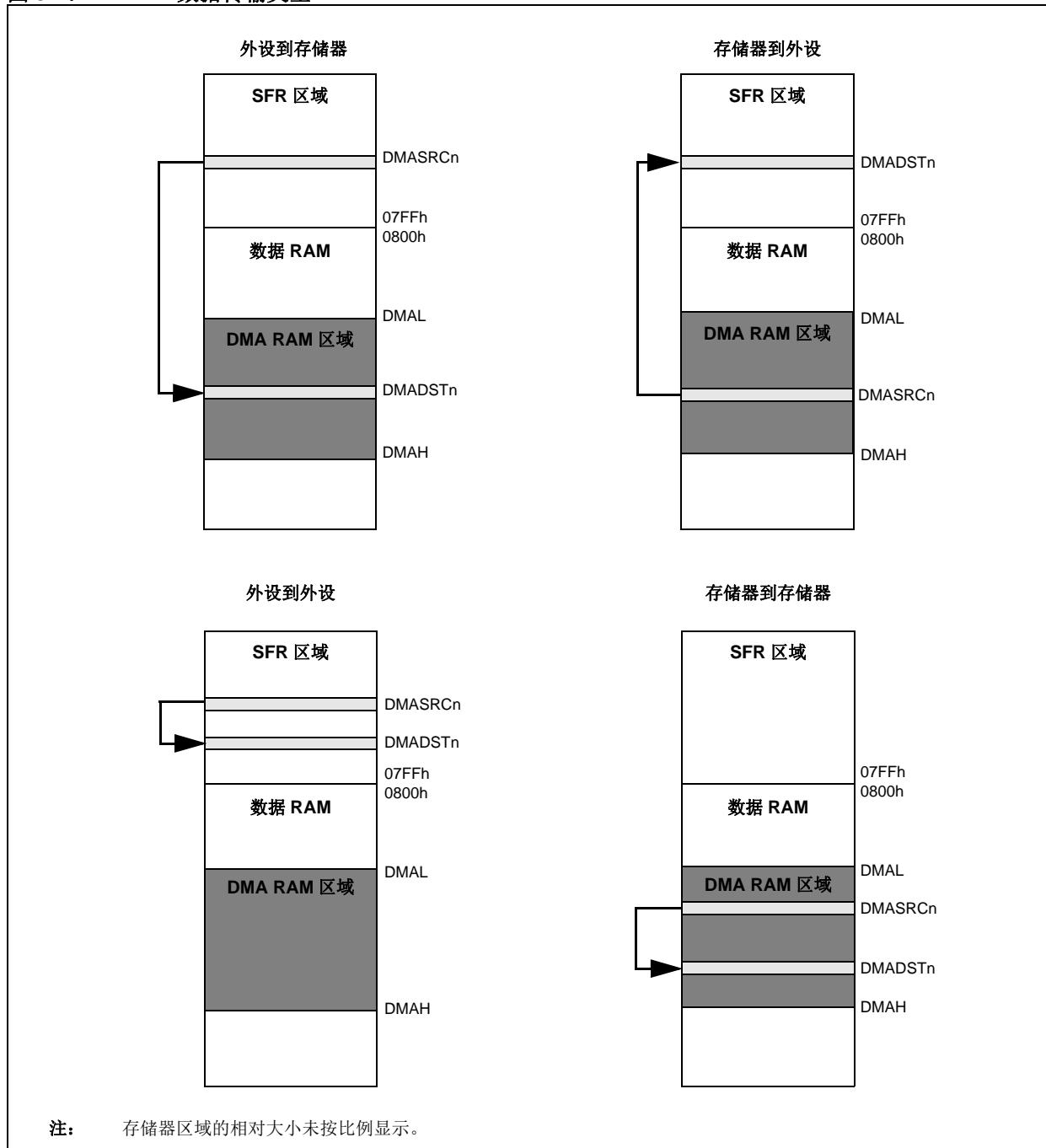
- 固定地址到固定地址：两个恒定地址之间
- 固定地址到地址块：从恒定源地址到目标地址范围
- 地址块到固定地址：从源地址范围到单个恒定目标地址
- 地址块到地址块：从源地址范围到目标地址范围

块寻址模式提供源和 / 或目标地址的自动递增和自动递减选项。

除了这四种基本模式之外，DMA 控制器还支持外设间接寻址（PIA）模式，其中源或目标地址由 DMA 控制器和支持 PIA 的外设共同生成。使能时，DMA 通道提供基本源和 / 或目标地址，而外设提供固定范围的偏移地址。

对于 PIC24FJ128GA310 系列器件，12 位 A/D 转换器模块是唯一支持 PIA 的外设。有关其在 PIA 模式下应用的详细信息，请参见[第 24.0 节 “带阈值扫描功能的 12 位 A/D 转换器”](#)。

图 5-2： DMA 数据传输类型



## 5.1.6 通道优先级

各条 DMA 通道相互独立工作，但为访问数据和 DMA 总线也相互竞争。发生访问冲突时，DMA 控制器使用用户选择的优先级机制对各个通道进行仲裁。有两种机制：

- 轮转：两条或多条通道发生冲突时，编号较低的通道在第一次冲突时获得优先权。随后再次冲突时，编号较高的通道根据通道编号依次获得优先权。
- 固定：两条或多条通道发生冲突时，编号最低的通道始终获得优先权，无论之前发生过什么情况。

## 5.2 典型设置

要设置 DMA 通道用于基本数据传输：

1. 使能 DMA 控制器 (**DMAEN = 1**) 并通过置 1 或清零 **PRSSEL** 选择相应的通道优先级机制。
2. 使用用于数据 RAM 操作的适当高/低地址边界设置 **DMAH** 和 **DMAL**。
3. 选择要使用的 DMA 通道并禁止其操作 (**CHEN = 0**)。
4. 将事务的相应源和目标地址编程到通道的 **DMASRCn** 和 **DMADSTn** 寄存器。对于 PIA 寻址模式，使用基址值。
5. 设置 **DMACNTn** 寄存器以获得每次传输的触发数量（单数据块或连续数据块模式）或要传输的字（字节）数（重复模式）。
6. 置 1 或清零 **SIZE** 位以选择数据大小。
7. 设置 **TRMODE** 位以选择数据传输模式。
8. 设置 **SAMODE** 和 **DAMODE** 位以选择寻址模式。
9. 通过置 1 **CHEN** 使能 DMA 通道。
10. 允许触发源中断。

## 5.3 外设模块禁止

与其他外设模块不同，DMA 控制器的通道无法使用外设模块禁止（PMD）寄存器单独断电。而是将通道分为两组进行控制。DMA0MD 位（**PMD7<4>**）有选择地控制 DMACH0 到 DMACH3。DMA1MD 位（**PMD7<5>**）控制 DMACH4 和 DMACH5。将这两位置 1 可实际上禁止 DMA 控制器。

## 5.4 寄存器

DMA 控制器使用一些寄存器来控制其操作。寄存器的数量取决于特定器件所实现的通道数。

始终有 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 控制寄存器（[寄存器 5-1](#)）
- **DMAH** 和 **DMAL**: 高 / 低地址限制寄存器
- **DMABUF**: DMA 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 通道控制寄存器（[寄存器 5-2](#)）
- **DMAINTn**: DMA 通道中断控制寄存器（[寄存器 5-3](#)）
- **DMASRCn**: 通道 n 的数据源地址指针
- **DMADSTn**: 通道 n 的数据目标地址指针
- **DMACNTn**: 通道 n 的事务计数器

PIC24FJ128GA310 系列器件共有 34 个寄存器。

## 寄存器 5-1: 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 = 上电复位时的值

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:** 通道优先级机制选择位

1 = 轮转机制

0 = 固定优先级机制

# PIC24FJ128GA310 系列

寄存器 5-2: 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
—	—	—	r	—	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 = 上电复位时的值	1 = 置 1
	0 = 清零
	x = 未知

- bit 15-12 未实现: 读为 0  
bit 12 保留: 保持为 0  
bit 11 未实现: 读为 0  
bit 10 **NULLW:** 空写模式位  
    1 = 每次写 DMADST 时, 启动对 DMASRC 的假写操作  
    0 = 不启动任何假写操作  
bit 9 **RELOAD:** 地址和计数重载位<sup>(1)</sup>  
    1 = 在启动下一个操作时, DMASRC、DMADST 和 DMACNT 寄存器重新装载它们先前的值  
    0 = 在启动下一个操作时, DMASRC、DMADST 和 DMACNT 不会重载<sup>(2)</sup>  
bit 8 **CHREQ:** DMA 通道软件请求位<sup>(3)</sup>  
    1 = DMA 请求由软件启动; 在 DMA 传输结束时自动清零  
    0 = 无任何 DMA 请求在等待  
bit 7-6 **SAMODE<1:0>:** 源地址模式选择位  
    11 = DMASRC 用于外设间接寻址且保持不变  
    10 = 传输完成后, DMASRC 根据 SIZE 位递减  
    01 = 传输完成后, DMASRC 根据 SIZE 位递增  
    00 = 传输完成后, DMASRC 保持不变  
bit 5-4 **DAMODE<1:0>:** 目标地址模式选择位  
    11 = DMADST 用于外设间接寻址且保持不变  
    10 = 传输完成后, DMADST 根据 SIZE 位递减  
    01 = 传输完成后, DMADST 根据 SIZE 位递增  
    00 = 传输完成后, DMADST 保持不变  
bit 3-2 **TRMODE<1:0>:** 传输模式选择位  
    11 = 重复连续数据块  
    10 = 连续数据块  
    01 = 重复单数据块  
    00 = 单数据块  
bit 1 **SIZE:** 数据大小选择位  
    1 = 字节 (8 位)  
    0 = 字 (16 位)  
bit 0 **CHEN:** DMA 通道使能位  
    1 = 使能相应的通道  
    0 = 禁止相应的通道

注 1: 只需存储 DMACNT 的原始值来恢复 DMASRC 和 DMADST 的原始值。

2: DMASRC、DMADST 和 DMACNT 始终在重复传输模式 (DMACHn<2> = 1) 时重载, 而与 RELOAD 位的状态无关。

3: CHREQ 置 1 时执行的传输数量取决于 TRMODE<1:0> 的配置。

## 寄存器 5-3: DMAINTn: DMA 通道 n 中断寄存器

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DBUFWF <sup>(1)</sup>	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **DBUFWF:** 缓冲数据写标志位 <sup>(1)</sup>  
 1 = 在空写模式下, DMA 缓冲区的内容未被写入 DMADST 或 DMASRC 中指定的地址单元  
 0 = 在空写模式下, DMA 缓冲区的内容被写入 DMADST 或 DMASRC 中指定的地址单元
- bit 14      未实现: 读为 0
- bit 13-8     **CHSEL<5:0>:** DMA 通道触发源选择位  
 完整列表请参见表 5-1。
- bit 7        **HIGHIF:** DMA 高地址限制中断标志位 <sup>(1,2)</sup>  
 1 = DMA 通道尝试访问高于 DMAH 的地址或数据 RAM 空间的上限。  
 0 = DMA 通道未调用高地址限制中断。
- bit 6        **LOWIF:** DMA 低地址限制中断标志位 <sup>(1,2)</sup>  
 1 = DMA 通道尝试访问低于 DMAL 但超出 SFR 范围 (07FFh) 的 DMA SFR 地址  
 0 = DMA 通道未调用低地址限制中断。
- bit 5        **DONEIF:** DMA 完成操作中断标志位 <sup>(1)</sup>  
如果 CHEN = 1:  
 1 = 前一 DMA 会话结束且完成  
 0 = 当前 DMA 会话尚未完成  
如果 CHEN = 0:  
 1 = 前一 DMA 会话结束且完成  
 0 = 前一 DMA 会话结束但未完成
- bit 4        **HALFIF:** DMA 50% 水线级中断标志位 <sup>(1)</sup>  
 1 = DMACNT 已达到与 0000h 相距的中点  
 0 = DMACNT 未达到中点
- bit 3        **OVRUNIF:** DMA 通道溢出标志位 <sup>(1)</sup>  
 1 = DMA 通道仍在完成基于前一触发信号的操作时被触发  
 0 = 未发生溢出条件
- bit 2-1     未实现: 读为 0
- bit 0        **HALFEN:** 完成一半水线位  
 1 = 当 DMACNT 达到其中点和完成时调用中断  
 0 = 仅在传输完成时调用中断

**注 1:** 用软件将这些标志位置 1 不会产生中断。

**2:** 实际访问之前未完成地址超限测试 (DMASRC 或 DMADST 大于 DMAH 或小于 DMAL)。

# PIC24FJ128GA310 系列

表 5-1: DMA 触发源

CHSEL<5:0>	触发 (中断)	CHSEL<5:0>	触发 (中断)
000000	(未实现)	100000	UART2 发送
000001	JTAG	100001	UART2 接收
000010	LCD	100010	外部中断 2
000011	UART4 发送	100011	Timer5
000100	UART4 接收	100100	Timer4
000101	UART4 错误	100101	输出比较 4
000110	UART3 发送	100110	输出比较 3
000111	UART3 接收	100111	DMA 通道 2
001000	UART3 错误	101000	输入捕捉 7
001001	CTMU 事件	101001	外部中断 1
001010	HLVD	101010	电平变化中断
001011	CRC 完成	101011	比较器事件
001100	UART2 错误	101100	I2C1 主事件
001101	UART1 错误	101101	I2C1 从事件
001110	RTCC	101110	DMA 通道 1
001111	DMA 通道 5	101111	A/D 转换器
010000	外部中断 4	110000	UART1 发送
010001	外部中断 3	110001	UART1 接收
010010	I2C2 主事件	110010	SPI1 事件
010011	I2C2 从事件	110011	SPI1 错误
010100	DMA 通道 4	110100	Timer3
010101	EPMP	110101	Timer2
010110	输出比较 7	110110	输出比较 2
010111	输出比较 6	110111	输入捕捉 2
011000	输出比较 5	111000	DMA 通道 0
011001	输入捕捉 6	111001	Timer1
011010	输入捕捉 5	111010	输出比较 1
011011	输入捕捉 4	111011	输入捕捉 1
011100	输入捕捉 3	111100	外部中断 0
011101	DMA 通道 3	111101	(未实现)
011110	SPI2 事件	111110	(未实现)
011111	SPI2 错误	111111	(未实现)

## 6.0 闪存程序存储器

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息, 请参见《PIC24F 系列参考手册》中的第 4 章“程序存储器”(D39715B\_CN)。本数据手册中的信息取代了 FRM 中的信息。

PIC24FJ128GA310 系列器件包含内部闪存程序存储器, 用于存储和执行应用程序代码。程序存储器是可读写并可擦除的。闪存有以下四种编程方式:

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

ICSP 允许在最终应用电路中对 PIC24FJ128GA310 系列器件进行串行编程。只需五根线就可以完成编程, 它们分别是编程时钟线 (PGE<sub>Cx</sub>)、编程数据线 (PGED<sub>x</sub>)、电源线 (VDD)、接地线 (VSS) 和主复位线 (MCLR)。这允许用户使用未编程器件制造电路

板, 仅在产品交付前才对单片机进行编程。从而可以将最新版本的固件或定制固件烧写到单片机中。

RTSP 是通过 TBLRD (表读) 和 TBLWT (表写) 指令完成的。使用 RTSP, 用户可以一次将 64 条指令 (192 字节) 的数据块写入程序存储器, 也可以一次擦除 512 条指令 (1536 字节) 的数据块。

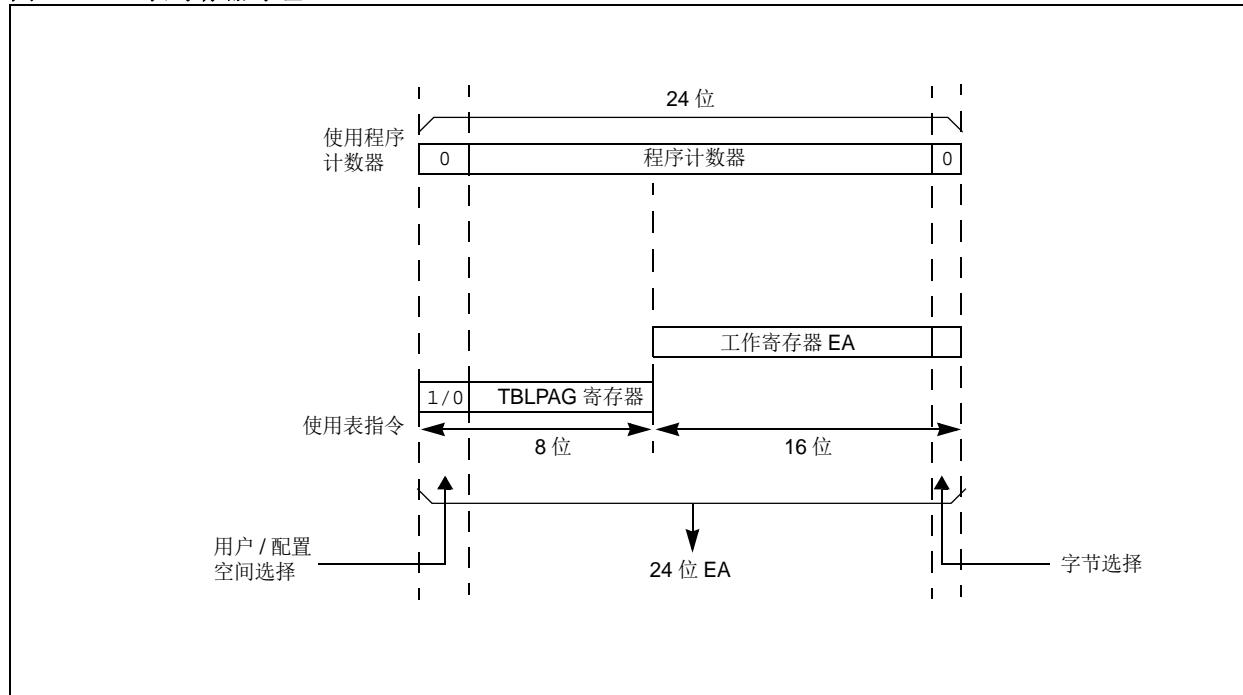
## 6.1 表指令和闪存编程

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。这些指令允许在器件正常工作模式下通过数据存储器直接读写程序存储空间。24 位程序存储器目标地址由 TBLPAG<7:0> 位和表指令中指定的 W 寄存器中的有效地址 (Effective Address, EA) 组成, 如图 6-1 所示。

TBLRDL 和 TBLWTL 指令用于读或写程序存储器的 bit<15:0>。TBLRDL 和 TBLWTL 指令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 指令也能以字或字节模式访问程序存储器。

图 6-1: 表寄存器寻址



## 6.2 RTSP 操作

PIC24F 闪存程序存储器阵列以 64 条指令或 192 字节的行为单位构成。RTSP 允许用户一次擦除 8 行（512 条指令）的块并一次编程一行，它还可以编程单字。

8 行擦除块和单行写入块都是边界对齐的，从程序存储器起始地址开始，分别到 1536 字节边界和 192 字节边界。

用 TBLWT 指令将数据写入程序存储器时，数据并未直接写入存储器，而是存储在保持锁存器中，直到执行编程序列。

可以执行任意数量的 TBLWT 指令，且写操作都将成功执行。但是，需要 64 条 TBLWT 指令来写存储器的整行。

要确保在写操作期间没有数据被改动，应将所有未使用的地址编程为 FFFFFFFh。这是因为保持锁存器复位为未知状态，因此当地址处于复位状态时，就可能改写未被重写的行上的存储单元。

RTSP 编程的基本步骤是先建立一个表指针，然后执行一系列 TBLWT 指令以将数据装入缓冲器。通过将 NVMCON 寄存器的控制位置 1 来执行编程。

可按任何顺序装入数据，且在执行写操作之前可以多次写保持寄存器。但后续写操作将覆盖先前的所有写操作。

**注：** 不推荐多次写入一个存储单元而不擦除其内容。

因为只写缓冲器，因此所有表写操作都是单字写操作（2 个指令周期）。编程每行都需要一个编程周期。

## 6.3 JTAG 操作

PIC24F 系列器件支持 JTAG 边界扫描。边界扫描可以通过验证引脚到 PCB 的连通性改进制造工艺。

## 6.4 增强型在线串行编程

增强型在线串行编程使用片上自举程序（称为编程执行程序）管理编程过程。通过使用 SPI 数据帧格式，编程执行程序可以擦除、编程和校验程序存储器。如需了解更多有关增强型 ICSP 的信息，请参见器件编程规范。

## 6.5 控制寄存器

有两个用于读写闪存程序存储器的 SFR：NVMCON 和 NVMKEY。

NVMCON 寄存器（[寄存器 6-1](#)）控制要擦除的块、要编程的存储器类型以及编程周期的开始时间。

NVMKEY 是只写寄存器，用于写保护。要启动编程或擦除过程，用户必须把 55h 和 AAh 连续写入 NVMKEY 寄存器。更多详细信息，请参见[第 6.6 节“编程操作”](#)。

## 6.6 编程操作

在 RTSP 模式下，对内部闪存进行编程或擦除需要完整的编程过程。在编程或擦除操作期间，处理器将停止（等待），直到操作完成。将 WR 位（NVMCON<15>）置 1 启动操作，当操作完成时 WR 位会自动清零。

## 寄存器 6-1: NVMCON: 闪存控制寄存器

R/S-0, HC <sup>(1)</sup>	R/W-0 <sup>(1)</sup>	R-0, HSC <sup>(1)</sup>	U-0	U-0	U-0	U-0	U-0
WR	WREN	WRERR	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0 <sup>(1)</sup>	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>
—	ERASE	—	—	NVMOP3 <sup>(2)</sup>	NVMOP2 <sup>(2)</sup>	NVMOP1 <sup>(2)</sup>	NVMOP0 <sup>(2)</sup>
bit 7	bit 0						

### 图注:

R = 可读位

-n = 上电复位时的值

HSC = 可由硬件置 1/清零的位

S = 可置 1 位

W = 可写位

1 = 置 1

HC = 可由硬件清零的位

U = 未实现位, 读为 0

0 = 清零

x = 未知

bit 15

### WR: 写控制位<sup>(1)</sup>

1 = 启动闪存编程或擦除操作。操作是自定时的, 一旦操作完成此位即由硬件清零。

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 = 尝试执行错误的编程或擦除序列或发生终止 (任何试图将 WR 位置 1 的操作都会自动置 1 此位)

0 = 编程或擦除操作正常完成

bit 12-7

未实现: 读为 0

bit 6

### ERASE: 擦除 / 编程使能位<sup>(1)</sup>

1 = 在下一个写命令执行由 NVMOP<3:0> 指定的擦除操作

0 = 在下一个写命令执行由 NVMOP<3:0> 指定的编程操作

bit 5-4

未实现: 读为 0

bit 3-0

### NVMOP<3:0>: NVM 操作选择位<sup>(1,2)</sup>

1111 = 存储器块擦除操作 (ERASE = 1) 或无操作 (ERASE = 0)<sup>(3)</sup>

0011 = 存储器字编程操作 (ERASE = 0) 或无操作 (ERASE = 1)

0010 = 存储器页擦除操作 (ERASE = 1) 或无操作 (ERASE = 0)

0001 = 存储器行编程操作 (ERASE = 0) 或无操作 (ERASE = 1)

**注 1:** 只能在上电复位时复位这些位。

**2:** NVMOP<3:0> 的所有其他组合均未实现。

**3:** 仅在 ICSP™ 模式下可用, 请参见器件编程规范。

## 6.6.1 闪存程序存储器编程算法

用户一次可编程闪存程序存储器的一行。要实现此操作，必须擦除包含该行在内的一个 8 行擦除块。一般步骤如下：

1. 读程序存储器中的 8 行（512 条指令），并将其存储在数据 RAM 中。
2. 用需要的新数据更新 RAM 中对应的程序数据。
3. 擦除块（见例 6-1）：
  - a) 将 NVMOP 位（NVMCON<3:0>）设置为 0010，以配置为块擦除操作。将 ERASE（NVMCON<6>）和 WREN（NVMCON<14>）位置 1。
  - b) 将要擦除块的起始地址写入 TBLPAG 和 W 寄存器。
  - c) 将 55h 写入 NVMKEY。
  - d) 将 AAh 写入 NVMKEY。
  - e) 将 WR 位（NVMCON<15>）置 1。启动擦除周期，CPU 停止工作等待擦除周期结束。擦除操作完成时，WR 位自动清零。

4. 将数据 RAM 的前 64 条指令写入程序存储器缓冲器（见例 6-3）。
5. 将程序块写入闪存：
  - a) 将 NVMOP 位设置为 0001，以配置为行编程操作。将 ERASE 位清零，将 WREN 位置 1。
  - b) 将 55h 写入 NVMKEY。
  - c) 将 AAh 写入 NVMKEY。
  - d) 将 WR 位置 1。启动编程周期，CPU 停止工作等待写周期完成。当写闪存的操作完成时，WR 位自动清零。
6. 将 TBLPAG 的值递增 1，使用数据 RAM 中下一个 64 条指令的块重复步骤 4 和 5，直到将所有 512 条指令写回闪存。

为了防止意外操作，必须向 NVMKEY 写入启动序列从而允许执行任何擦除或编程操作。执行了编程命令以后，用户必须等待一定的编程时间，直到编程操作完成。编程序列开始后紧跟的两条指令必须为 NOP，如例 6-4 所示。

## 例 6-1：擦除程序存储器块（汇编语言代码）

```
; Set up NVMCON for block erase operation
MOV    #0x4042, W0 ;                         ; Initialize NVMCON
MOV    W0, NVMCON
; Init pointer to row to be ERASED
MOV    #tblpage(PROG_ADDR), W0                 ; Initialize Program Memory (PM) Page Boundary SFR
MOV    W0, TBLPAG
MOV    #tbloffset(PROG_ADDR), W0                ; Initialize in-page EA<15:0> pointer
TBLWTL W0, [W0]                                ; Set base address of erase block
DISI   #5                                     ; Block all interrupts with priority <7
                                                ; for next 5 instructions

MOV.B #0x55, W0                                 ; Write the 0x55 key
MOV    W0, NVMKEY
MOV.B #0xAA, W1 ;                               ; Write the 0xAA key
MOV    W1, NVMKEY
BSET  NVMCON, #WR
NOP
NOP
                                                ; Start the erase sequence
                                                ; Insert two NOPs after the erase
                                                ; command is asserted
```

## 例 6-2：擦除程序存储器块（C 语言代码）

```

// C example using MPLAB C30
unsigned long progAddr = 0xFFFFFFFF;           // Address of row to write
unsigned int offset;
//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16;                         // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;                       // Initialize lower word of address
__builtin_tblwtl(offset, 0x0000);                 // Set base address of erase block
// with dummy latch write
NVMCON = 0x4042;                                // Initialize NVMCON
asm("DISI #5");                                 // Block all interrupts with priority <7
// for next 5 instructions
__builtin_write_NVM();                           // check function to perform unlock
// sequence and set WR

```

## 例 6-3：装载写缓冲器

```

; Set up NVMCON for row programming operations
MOV    #0x4001, W0                                ;
MOV    W0, NVMCON                                  ; Initialize NVMCON
; Set up a pointer to the first program memory location to be written
; program memory selected, and writes enabled
MOV    #0x0000, W0                                ;
MOV    W0, TBLPAG                                 ; Initialize PM Page Boundary SFR
MOV    #0x6000, W0                                ; An example program memory address
; Perform the TBLWT instructions to write the latches
; 0th_program_word
MOV    #LOW_WORD_0, W2                            ;
MOV    #HIGH_BYTE_0, W3                            ;
TBLWTL W2, [W0]                                  ; Write PM low word into program latch
TBLWTH W3, [W0++]                                ; Write PM high byte into program latch
; 1st_program_word
MOV    #LOW_WORD_1, W2                            ;
MOV    #HIGH_BYTE_1, W3                            ;
TBLWTL W2, [W0]                                  ; Write PM low word into program latch
TBLWTH W3, [W0++]                                ; Write PM high byte into program latch
; 2nd_program_word
MOV    #LOW_WORD_2, W2                            ;
MOV    #HIGH_BYTE_2, W3                            ;
TBLWTL W2, [W0]                                  ; Write PM low word into program latch
TBLWTH W3, [W0++]                                ; Write PM high byte into program latch
•
•
•
; 63rd_program_word
MOV    #LOW_WORD_63, W2                           ;
MOV    #HIGH_BYTE_63, W3                           ;
TBLWTL W2, [W0]                                  ; Write PM low word into program latch
TBLWTH W3, [W0]                                  ; Write PM high byte into program latch

```

## 例 6-4：启动编程序列

```

DISI    #5                                     ; Block all interrupts with priority <7
                                                ; for next 5 instructions
MOV.B  #0x55, W0                               ; Write the 0x55 key
MOV    W0, NVMKEY                             ;
MOV.B  #0xAA, W1                               ; Write the 0xAA key
MOV    W1, NVMKEY                             ;
BSET   NVMCON, #WR                            ; Start the programming sequence
NOP
NOP
BTSC   NVMCON, #15                            ; and wait for it to be
                                                ; completed
BRA    $-2

```

# PIC24FJ128GA310 系列

## 6.6.2 编程闪存程序存储器的一个单字

若已擦除了一个闪存单元，则可用表写指令对此单元进行编程以将一个指令字（24位）写入写锁存器。将闪存地址的8个最高位（MSb）装入 TBLPG寄存器。TBLWTL 和 TBLWTH 指令将所需数据写入写锁存器，并指定要写入的程序存储器地址的低16位。要将

NVMCON 寄存器配置为字写操作，将 NVMOP 位（NVMCON<3:0>）设置为 0011。通过执行解锁序列并将 WR 位置 1 来执行此写操作（见例 6-5）。例 6-6 为使用 MPLAB C30 编译器和内置的硬件功能的等效 C 编译器过程。

### 例 6-5： 编程闪存程序存储器的一个单字

```
; Setup a pointer to data Program Memory
MOV      #tblpage(PROG_ADDR), W0          ;
MOV      W0, TBLPG                         ;Initialize PM Page Boundary SFR
MOV      #tbloffset(PROG_ADDR), W0          ;Initialize a register with program memory address

MOV      #LOW_WORD_N, W2                  ;
MOV      #HIGH_BYTE_N, W3                  ;
TBLWTL  W2, [W0]                          ; Write PM low word into program latch
TBLWTH  W3, [W0++]                         ; Write PM high byte into program latch

; Setup NVMCON for programming one word to data Program Memory
MOV      #0x4003, W0                      ;
MOV      W0, NVMCON                        ; Set NVMOP bits to 0011

DISI    #5                                ; Disable interrupts while the KEY sequence is written
MOV.B   #0x55, W0                           ; Write the key sequence
MOV     W0, NVMKEY
MOV.B   #0xAA, W0
MOV     W0, NVMKEY
BSET   NVMCON, #WR                         ; Start the write cycle
NOP
NOP
NOP
```

### 例 6-6： 编程闪存程序存储器的一个单字（C 语言代码）

```
// C example using MPLAB C30
unsigned int offset;
unsigned long progAddr = 0xFFFFFFFF;           // Address of word to program
unsigned int progDataL = 0xFFFF;                // Data to program lower word
unsigned char progDataH = 0xFF;                 // Data to program upper byte

//Set up NVMCON for word programming
NVMCON = 0x4003;                             // Initialize NVMCON

//Set up pointer to the first memory location to be written
TBLPAG = progAddr>>16;                       // Initialize PM Page Boundary SFR
offset = progAddr & 0xFFFF;                     // Initialize lower word of address

//Perform TBLWT instructions to write latches
__builtin_tblwtl(offset, progDataL);           // Write to address low word
__builtin_tblwth(offset, progDataH);           // Write to upper byte
asm("DISI #5");                            // Block interrupts with priority <7
                                            // for next 5 instructions
__builtin_write_NVM();                         // C30 function to perform unlock
                                            // sequence and set WR
```

## 7.0 复位

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息, 请参见《PIC24F 系列参考手册》中的第 7 章“复位”(DS39712D\_CN)。本数据手册中的信息取代了 FRM 中的信息。

复位模块将所有的复位源组合在一起并控制器件主复位信号 **SYSRST**。以下所列是器件的复位源:

- **POR:** 上电复位
- **MCLR:** 引脚复位
- **SWR:** RESET 指令
- **WDT:** 看门狗定时器复位
- **BOR:** 欠压复位
- **CM:** 配置失配复位
- **TRAPR:** 陷阱冲突复位
- **IOPUWR:** 非法操作码复位
- **UWR:** 未初始化的 W 寄存器复位

复位模块的简化框图如图 7-1 所示。

任何激活的复位源都会激活 **SYSRST** 信号。许多与 CPU 和外设有关的寄存器被强制为一个已知的复位状态。大多数寄存器不受复位的影响; 在 POR 时它们的状态未知, 而在所有其他复位时它们的状态不变。

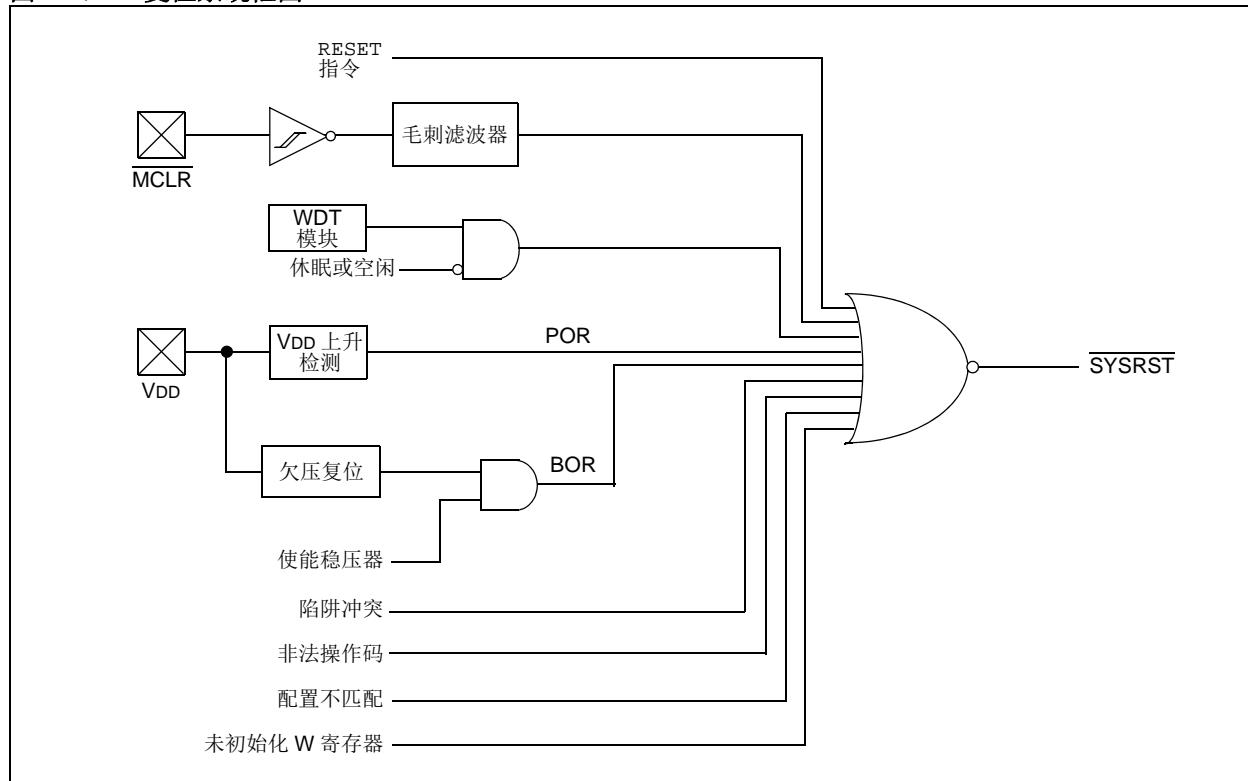
**注:** 有关寄存器复位状态的信息, 请参见本手册中的特定外设或 CPU 章节。

任何类型的器件复位都会将 RCON 寄存器中相应的位置 1 以表明复位类型(见 [寄存器 7-1](#))。此外, 在使用极其实能的功能(例如 VBAT)时发生的复位事件会将 RCON2 寄存器([寄存器 7-2](#))中的一个或多个状态位置 1。上电复位会清零 BOR 和 POR 位(RCON<1:0>)外的所有位, 而 BOR 和 POR 位会被置 1。用户可在代码执行过程中的任何时间置 1 或清零任何位。RCON 位仅作为状态位。用软件将特定的复位状态位置 1 不会导致器件发生复位。

RCON 寄存器也有一些与看门狗定时器和器件节能状态相关的位。本数据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, 这样下一次器件复位后的 RCON 寄存器值才会有意义。

图 7-1: 复位系统框图



# PIC24FJ128GA310 系列

寄存器 7-1: RCON: 复位控制寄存器

R/W-0	R/W-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
TRAPR <sup>(1)</sup>	IOPUWR <sup>(1)</sup>	—	RETEN <sup>(2)</sup>	—	DPSLP <sup>(1)</sup>	CM <sup>(1)</sup>	VREGS <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-1	R/W-1
EXTR <sup>(1)</sup>	SWR <sup>(1)</sup>	SWDTEN <sup>(4)</sup>	WDTO <sup>(1)</sup>	SLEEP <sup>(1)</sup>	IDLE <sup>(1)</sup>	BOR <sup>(1)</sup>	POR <sup>(1)</sup>
bit 7				bit 0			

图注:	U = 未实现位, 读为 0
R = 可读位	W = 可写位
-n = 上电复位时的值	HS = 可由硬件置 1 的位 1 = 置 1 0 = 清零 x = 未知

- bit 15      **TRAPR:** 陷阱复位标志位 <sup>(1)</sup>  
1 = 发生了陷阱冲突复位  
0 = 未发生陷阱冲突复位
- bit 14      **IOPUWR:** 非法操作码或因访问未初始化的 W 寄存器而复位的标志位 <sup>(1)</sup>  
1 = 检测到非法操作码、非法地址模式或未初始化的 W 寄存器用作地址指针而导致的复位  
0 = 未发生由非法操作码或未初始化的 W 寄存器而导致的复位
- bit 13-10    未实现: 读为 0
- bit 12      **RETEN:** 保持模式使能位 <sup>(2)</sup>  
1 = 在器件处于休眠模式时（向内核供应 1.2V 稳压电源）启用保持模式  
0 = 禁用保持模式；存在正常电平
- bit 11      未实现: 读为 0
- bit 10      **DPSLP:** 深度休眠标志位 <sup>(1)</sup>  
1 = 器件处于深度休眠模式  
0 = 器件未处于深度休眠模式
- bit 9        **CM:** 配置字失配复位标志位 <sup>(1)</sup>  
1 = 发生了配置字失配复位  
0 = 未发生配置字失配复位
- bit 8        **VREGS:** 休眠期间程序存储器上电位 <sup>(3)</sup>  
1 = 休眠期间，程序存储器偏置电压保持上电状态  
0 = 休眠期间，程序存储器偏置电压掉电
- bit 7        **EXTR:** 外部复位 (MCLR) 引脚位 <sup>(1)</sup>  
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:** 软件复位（指令）标志位 <sup>(1)</sup>  
1 = 执行了 RESET 指令  
0 = 未执行 RESET 指令

- 注 1:** 所有复位状态位都可以用软件置 1 或清零。用软件将任何一位置 1 不会导致器件复位。
- 2:** 如果 LPCFG 配置位为 1（未编程），则禁用保持稳压器且 RETEN 不起作用。
- 3:** 在稳压器进入待机模式后将其重新使能，将在从休眠中唤醒时增加一个延时 TVREG。不使用稳压器的应用应该将此位置 1，以防止发生此延时。
- 4:** 如果 FWDTEN 配置位为 1（未编程），则 WDT 总是使能，而不管 SWDTEN 位的设置如何。

## 寄存器 7-1: RCON: 复位控制寄存器 (续)

bit 5	<b>SWDTEN:</b> 软件使能 / 禁止 WDT 位 <a href="#">(4)</a>
	1 = 使能 WDT 0 = 禁止 WDT
bit 4	<b>WDTO:</b> 看门狗定时器超时标志位 <a href="#">(1)</a>
	1 = 发生了 WDT 超时 0 = 未发生 WDT 超时
bit 3	<b>SLEEP:</b> 从休眠状态唤醒标志位 <a href="#">(1)</a>
	1 = 器件处于休眠模式 0 = 器件未处于休眠模式
bit 2	<b>IDLE:</b> 从空闲状态唤醒标志位 <a href="#">(1)</a>
	1 = 器件处于空闲模式 0 = 器件未处于空闲模式
bit 1	<b>BOR:</b> 欠压复位标志位 <a href="#">(1)</a>
	1 = 发生了欠压复位 (上电复位后也置 1)。 0 = 未发生欠压复位
bit 0	<b>POR:</b> 上电复位标志位 <a href="#">(1)</a>
	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。用软件将任何一位置 1 不会导致器件复位。

**2:** 如果 **LPCFG** 配置位为 1 (未编程)，则禁用保持稳压器且 **RETEN** 不起作用。

**3:** 在稳压器进入待机模式后将其重新使能，将在从休眠中唤醒时增加一个延时 **TVREG**。不使用稳压器的应用应该将此位置 1，以防止发生此延时。

**4:** 如果 **FWDTEN** 配置位为 1 (未编程)，则 **WDT** 总是使能，而不管 **SWDTEN** 位的设置如何。

# PIC24FJ128GA310 系列

寄存器 7-2: RCON2: 复位和系统控制寄存器 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-0	R/CO-1	R/CO-1	R/CO-1	R/CO-0
—	—	—	r	VDDBOR <sup>(1)</sup>	VDDPOR <sup>(1,2)</sup>	VBPOR <sup>(1,3)</sup>	VBAT <sup>(1)</sup>
bit 7							bit 0

图注:	CO = 只可清零位	r = 保留位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零

- bit 15-5      未实现: 读为 0
- bit 4      保留: 保持为 0
- bit 3      **VDDBOR:** VDD 欠压复位标志位<sup>(1)</sup>  
1 = 发生了 VDD 欠压复位 (由硬件置 1)  
0 = 未发生 VDD 欠压复位
- bit 2      **VDDPOR:** VDD 上电复位标志位<sup>(1,2)</sup>  
1 = 发生了 VDD 上电复位 (由硬件置 1)  
0 = 未发生 VDD 上电复位
- bit 1      **VBPOR:** VBAT 标志位<sup>(1,3)</sup>  
1 = 发生了 VBAT 上电复位 (VBAT 引脚未连接电池, 或 VBAT 电平低于深度休眠信号保持电平, 由硬件置 1)  
0 = 未发生 VBAT 上电复位
- bit 0      **VBAT:** VBAT 标志位<sup>(1)</sup>  
1 = 向 VBAT 引脚供电时发生 POR 退出 (由硬件置 1)  
0 = 未发生 VBAT POR 退出

- 注 1: 此位仅可由硬件置 1; 仅可由软件清零。  
2: 表示 VDD POR。POR 位 (RCON<0>) 置 1 表示 VCORE POR。  
3: 器件最初上电时此位置 1, 即使 VBAT 已通电。

表 7-1：复位标志位操作

标志位	置 1 事件	清零事件
TRAPR (RCON<15>)	陷阱冲突事件	POR
IOPUWR (RCON<14>)	非法操作码或访问未初始化的 W 寄存器	POR
CM (RCON<9>)	配置失配复位	POR
EXTR (RCON<7>)	MCLR 复位	POR
SWR (RCON<6>)	RESET 指令	POR
WDTO (RCON<4>)	WDT 超时	CLRWDT、PWRSAV 指令和 POR
SLEEP (RCON<3>)	PWRSAV #0 指令	POR
DPSLP (RCON<10>)	PWRSAV #0 指令 (DSEN 位置 1 时)	POR
IDLE (RCON<2>)	PWRSAV #1 指令	POR
BOR (RCON<1>)	POR, BOR	—
POR (RCON<0>)	POR	—

注：所有复位标志位可由用户软件置 1 或清零。

## 7.1 特殊功能寄存器的复位状态

大部分与 PIC24F CPU 和外设有关的特殊功能寄存器 (SFR) 会在器件复位时复位为某个特定值。SFR 是按其外设或 CPU 功能分组的，其复位值在本手册的各章节中有所说明。

除了四个寄存器之外，其他 SFR 的复位值都不受复位类型的影响。复位控制寄存器 RCON 的复位值取决于器件复位的类型。振荡器控制寄存器 OSCCON 的复位值取决于复位的类型和闪存配置字 2 (CW2) 中 FNOSC 位的编程值 (见表 7-2)。RCFGCAL 和 NVMCON 寄存器只受 POR 影响。

## 7.2 器件复位时间

表 7-3 总结了各种类型的器件复位所需的时间。注意，在 POR 延时结束后会发出系统复位信号 SYSRST。

器件实际开始执行代码的时间还取决于系统振荡器延时，它包括振荡器起振定时器 (OST) 延时和 PLL 锁定时间。OST 和 PLL 锁定时间与相应的 SYSRST 延时同时发生。

故障保护时钟监视器 (FSCM) 延时决定在 SYSRST 信号发出后 FSCM 开始监视系统时钟源的时间。

## 7.3 欠压复位 (BOR)

PIC24FJ128GA310 系列器件带有一个 BOR 电路，它将为用户提供多个配置和节能选项。BOR 由 BOREN (CW3<12>) 配置位控制。

BOR 使能时，VDD 只要降至低于 BOR 阈值，都将导致器件欠压复位。有关阈值电平，请参见第 32.1 节“直流特性”(参数 DC17)。

## 7.4 复位时的时钟源选择

如果使能了时钟切换，器件复位时的系统时钟源可按照表 7-2 进行选择。如果禁止时钟切换，则总是根据振荡器配置位选择系统时钟源。有关详细信息，请参见《PIC24F 系列参考手册》第 6.0 章“振荡器”(DS39700C\_CN)。

表 7-2：振荡器选择与复位类型的关系 (使能时钟切换)

复位类型	时钟源的确定
POR	FNOSC 配置位 (CW2<10:8>)
BOR	FNOSC 配置位 (CW2<10:8>)
MCLR	FNOSC 配置位 (CW2<10:8>)
WDTO	COSC 控制位 (OSCCON<14:12>)
SWR	COSC 控制位 (OSCCON<14:12>)

# PIC24FJ128GA310 系列

表 7-3：各种器件复位的复位延时

复位类型	时钟源	SYSRST 延时	系统时钟延时	注
POR	EC	Tpor + Tstartup + TRST	—	<b>1, 2, 3</b>
	ECPLL	Tpor + Tstartup + TRST	TLOCK	<b>1, 2, 3, 5</b>
	XT、HS 和 SOSC	Tpor + Tstartup + TRST	TOST	<b>1, 2, 3, 4, 8</b>
	XTPLL 和 HSPLL	Tpor + Tstartup + TRST	TOST + TLOCK	<b>1, 2, 3, 4, 5, 8</b>
	FRC 和 FRCDIV	Tpor + Tstartup + TRST	TFRC	<b>1, 2, 3, 6, 7</b>
	FRCPLL	Tpor + Tstartup + TRST	TFRC + TLOCK	<b>1, 2, 3, 5, 6</b>
	LPRC	Tpor + Tstartup + TRST	TLRPC	<b>1, 2, 3, 6</b>
BOR	EC	Tstartup + TRST	—	<b>2, 3</b>
	ECPLL	Tstartup + TRST	TLOCK	<b>2, 3, 5</b>
	XT、HS 和 SOSC	Tstartup + TRST	TOST	<b>2, 3, 4, 8</b>
	XTPLL 和 HSPLL	Tstartup + TRST	TOST + TLOCK	<b>2, 3, 4, 5, 8</b>
	FRC 和 FRCDIV	Tstartup + TRST	TFRC	<b>2, 3, 6, 7</b>
	FRCPLL	Tstartup + TRST	TFRC + TLOCK	<b>2, 3, 5, 6</b>
	LPRC	Tstartup + TRST	TLRPC	<b>2, 3, 6</b>
MCLR	任何时钟	TRST	—	<b>3</b>
WDT	任何时钟	TRST	—	<b>3</b>
软件	任何时钟	TRST	—	<b>3</b>
非法操作码	任何时钟	TRST	—	<b>3</b>
未初始化的 W 寄存器	任何时钟	TRST	—	<b>3</b>
陷阱冲突	任何时钟	TRST	—	<b>3</b>

注 1: TPOR = 上电复位延时（标称值为 10 μs）。

2: TSTARTUP = TVREG (当 VREGS = 1 或 VREGS = 0 时, 标称值为 10 μs; 取决于 WDTWIN<1:0> 位设置)。

3: TRST = 内部状态复位时间 (标称值为 2 μs)。

4: TOST = 振荡器起振定时器 (OST) 延时。10 位计数器计满 1024 个振荡器周期后, 才将振荡器时钟释放给系统使用。

5: TLOCK = PLL 锁定时间。

6: TFRC 和 TLRPC = RC 振荡器起振时间。

7: 如果使能了双速启动, 无论选择了哪种主振荡器, 器件都将使用 FRC 启动, 因此, 系统时钟延时只是 TFRC, 在这些情况下, FRC 起振时间有效。将在各自的时钟延时后切换到主振荡器。

8: TOST = 振荡器起振定时器 (OST) 延时。10 位计数器等待 1024 个振荡器周期后, 才将振荡器时钟释放给系统使用。

## 7.4.1 POR 和长振荡器起振时间

振荡器起振电路及其相关的延时定时器与上电时发生的器件复位延时没有关系。某些晶振电路 (尤其是低频晶振) 的起振时间会相对较长。因此, 在发出 SYSRST 之后, 可能会发生以下一种或多种情况:

- 振荡器电路未起振。
- 振荡器起振定时器尚未超时 (如果使用了晶振)。
- PLL 还未锁定 (如果使用了 PLL)。

在有效时钟源供系统使用之前, 器件不会开始执行代码。因此, 如果必须确定复位延时, 还须考虑振荡器和 PLL 起振延时。

## 7.4.2 故障保护时钟监视器 (FSCM) 和器件复位

如果使能了 FSCM, 它将在发出 SYSRST 时开始监视系统时钟源。如果此时没有可用的有效时钟源, 器件将会自动切换到 FRC 振荡器, 且用户可在陷阱服务程序 (Trap Service Routine, TSR) 中将系统时钟源切换到所需的晶振。

## 8.0 中断控制器

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息, 请参见《PIC24F 系列参考手册》中的第 8 章“中断”(DS39707A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

PIC24F 中断控制器将诸多外设中断请求信号缩减到一个到 PIC24F CPU 的中断请求信号。该控制器具有以下特性:

- 多达 8 个处理器异常和软件陷阱
- 7 个用户可选择的优先级
- 具有多达 118 个向量的中断向量表 (Interrupt Vector Table, IVT)
- 每个中断或异常源对应一个惟一的向量
- 指定的用户优先级中具有固定优先级
- 用于支持调试的备用中断向量表 (Alternate Interrupt Vector Table, AIVT)
- 固定的中断进入和返回延时

### 8.1 中断向量表

中断向量表 (IVT) 如图 8-1 所示。IVT 位于程序存储器中, 起始单元地址为 000004h。IVT 包含 126 个向量, 由 8 个不可屏蔽的陷阱向量和多达 118 个中断源组成。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。每个中断向量存储单元中设置的值是其对应中断服务程序 (Interrupt Service Routine, ISR) 的起始地址。

中断向量有一个自然优先级; 也就是说每个中断向量的优先级与其在向量表中的位置有关。如果其他方面都相同, 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。

PIC24FJ128GA310 系列器件实现了不可屏蔽陷阱和唯一中断。表 8-1 和表 8-2 对此做了总结。

#### 8.1.1 备用中断向量表

如图 8-1 所示, 备用中断向量表 (AIVT) 位于 IVT 之后。ALTI<sub>VT</sub> (INTCON2<15>) 控制位提供对 AIVT 的访问。如果 ALTI<sub>VT</sub> 位置 1, 所有中断和异常处理将使用备用向量而不是默认的向量。备用向量与默认向量的构成方式相同。

AIVT 支持仿真和调试功能, 它提供了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。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软件算法。如果不需要 AIVT, 应该用 IVT 中使用的地址设置 AIVT。

### 8.2 复位过程

由于复位过程中不涉及到中断控制器, 所以器件复位并不是真的异常。作为对复位的响应, PIC24F 器件清零其寄存器, 同时强制 PC 为零。然后, 单片机从单元 000000h 开始执行程序。用户在复位地址中设置一条 GOTO 指令, 该指令会使程序执行重新定位到相应的启动程序。

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

# PIC24FJ128GA310 系列

图 8-1： PIC24F 中断向量表

自然优先级降序排列 ↓

复位 ——GOTO 指令	000000h
复位 ——GOTO 地址	000002h
保留	000004h
振荡器故障陷阱向量	
地址错误陷阱向量	
堆栈错误陷阱向量	
算术错误陷阱向量	
保留	
保留	
保留	
中断向量 0	000014h
中断向量 1	
—	
—	
—	
中断向量 52	00007Ch
中断向量 53	00007Eh
中断向量 54	000080h
—	
—	
—	
中断向量 116	0000FCCh
中断向量 117	0000FEh
保留	000100h
保留	000102h
保留	
振荡器故障陷阱向量	
地址错误陷阱向量	
堆栈错误陷阱向量	
算术错误陷阱向量	
保留	
保留	
保留	
中断向量 0	000114h
中断向量 1	
—	
—	
—	
中断向量 52	00017Ch
中断向量 53	00017Eh
中断向量 54	000180h
—	
—	
—	
中断向量 116	0001FEh
中断向量 117	
代码起始	000200h

中断向量表 (IVT)<sup>(1)</sup>

备用中断向量表 (AIVT)<sup>(1)</sup>

**注 1：** 有关中断向量列表的信息，请参见表 8-2。

表 8-1： 陷阱向量详细信息

向量编号	IVT 地址	AIVT 地址	陷阱源
0	000004h	000104h	保留
1	000006h	000106h	振荡器故障
2	000008h	000108h	地址错误
3	00000Ah	00010Ah	堆栈错误
4	00000Ch	00010Ch	算术错误
5	00000Eh	00010Eh	保留
6	000010h	000110h	保留
7	000012h	000112h	保留

表 8-2： 实现的中断向量

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	使能	优先级
ADC1 转换完成	13	00002Eh	00012Eh	IFS0<13>	IEC0<13>	IPC3<6:4>
比较器事件	18	000038h	000138h	IFS1<2>	IEC1<2>	IPC4<10:8>
CRC 发生器	67	00009Ah	00019Ah	IFS4<3>	IEC4<3>	IPC16<14:12>
CTMU 事件	77	0000AEh	0001AEh	IFS4<13>	IEC4<13>	IPC19<6:4>
DMA 通道 0	4	00001Ch	00011Ch	IFS0<4>	IEC0<4>	IPC1<2:0>
DMA 通道 1	14	000030h	000130h	IFS0<14>	IEC0<14>	IPC3<10:8>
DMA 通道 2	24	000044h	000144h	IFS1<8>	IEC1<8>	IPC6<2:0>
DMA 通道 3	36	00005Ch	00015Ch	IFS2<4>	IEC2<4>	IPC9<2:0>
DMA 通道 4	46	000070h	000170h	IFS2<14>	IEC2<14>	IPC11<10:8>
DMA 通道 5	61	00008Eh	00018Eh	IFS3<13>	IEC3<13>	IPC15<6:4>
外部中断 0	0	000014h	000114h	IFS0<0>	IEC0<0>	IPC0<2:0>
外部中断 1	20	00003Ch	00013Ch	IFS1<4>	IEC1<4>	IPC5<2:0>
外部中断 2	29	00004Eh	00014Eh	IFS1<13>	IEC1<13>	IPC7<6:4>
外部中断 3	53	00007Eh	00017Eh	IFS3<5>	IEC3<5>	IPC13<6:4>
外部中断 4	54	000080h	000180h	IFS3<6>	IEC3<6>	IPC13<10:8>
I2C1 主事件	17	000036h	000136h	IFS1<1>	IEC1<1>	IPC4<6:4>
I2C1 从事件	16	000034h	000134h	IFS1<0>	IEC1<0>	IPC4<2:0>
I2C2 主事件	50	000078h	000178h	IFS3<2>	IEC3<2>	IPC12<10:8>
I2C2 从事件	49	000076h	000176h	IFS3<1>	IEC3<1>	IPC12<6:4>
输入捕捉 1	1	000016h	000116h	IFS0<1>	IEC0<1>	IPC0<6:4>
输入捕捉 2	5	00001Eh	00011Eh	IFS0<5>	IEC0<5>	IPC1<6:4>
输入捕捉 3	37	00005Eh	00015Eh	IFS2<5>	IEC2<5>	IPC9<6:4>
输入捕捉 4	38	000060h	000160h	IFS2<6>	IEC2<6>	IPC9<10:8>
输入捕捉 5	39	000062h	000162h	IFS2<7>	IEC2<7>	IPC9<14:12>
输入捕捉 6	40	000064h	000164h	IFS2<8>	IEC2<8>	IPC10<2:0>
输入捕捉 7	22	000040h	000140h	IFS1<6>	IEC1<6>	IPC5<10:8>
JTAG	117	0000FEh	0001FEh	IFS7<5>	IEC7<5>	IPC29<6:4>
输入电平变化通知 (ICN)	19	00003Ah	00013Ah	IFS1<3>	IEC1<3>	IPC4<14:12>
LCD 控制器	100	0000DCh	0001DCh	IFS6<4>	IEC6<4>	IPC25<2:0>
低电压检测 (LVD)	72	0000A4h	0001A4h	IFS4<8>	IEC4<8>	IPC18<2:0>
输出比较 1	2	000018h	000118h	IFS0<2>	IEC0<2>	IPC0<10:8>
输出比较 2	6	000020h	000120h	IFS0<6>	IEC0<6>	IPC1<10:8>
输出比较 3	25	000046h	000146h	IFS1<9>	IEC1<9>	IPC6<6:4>
输出比较 4	26	000048h	000148h	IFS1<10>	IEC1<10>	IPC6<10:8>
输出比较 5	41	000066h	000166h	IFS2<9>	IEC2<9>	IPC10<6:4>
输出比较 6	42	000068h	000168h	IFS2<10>	IEC2<10>	IPC10<10:8>
输出比较 7	43	00006Ah	00016Ah	IFS2<11>	IEC2<11>	IPC10<14:12>
增强型并行主端口 (EPMP)	45	00006Eh	00016Eh	IFS2<13>	IEC2<13>	IPC11<6:4>
实时时钟和日历 (RTCC)	62	000090h	000190h	IFS3<14>	IEC3<14>	IPC15<10:8>
SPI1 错误	9	000026h	000126h	IFS0<9>	IEC0<9>	IPC2<6:4>
SPI1 事件	10	000028h	000128h	IFS0<10>	IEC0<10>	IPC2<10:8>
SPI2 错误	32	000054h	000154h	IFS2<0>	IEC2<0>	IPC8<2:0>
SPI2 事件	33	000056h	000156h	IFS2<1>	IEC2<1>	IPC8<6:4>

表 8-2： 实现的中断向量

中断源	向量编号	IVT 地址	AIVT 地址	中断位位置		
				标志	使能	优先级
Timer1	3	00001Ah	00011Ah	IFS0<3>	IEC0<3>	IPC0<14:12>
Timer2	7	000022h	000122h	IFS0<7>	IEC0<7>	IPC1<14:12>
Timer3	8	000024h	000124h	IFS0<8>	IEC0<8>	IPC2<2:0>
Timer4	27	00004Ah	00014Ah	IFS1<11>	IEC1<11>	IPC6<14:12>
Timer5	28	00004Ch	00014Ch	IFS1<12>	IEC1<12>	IPC7<2:0>
UART1 错误	65	000096h	000196h	IFS4<1>	IEC4<1>	IPC16<6:4>
UART1 接收器	11	00002Ah	00012Ah	IFS0<11>	IEC0<11>	IPC2<14:12>
UART1 发送器	12	00002Ch	00012Ch	IFS0<12>	IEC0<12>	IPC3<2:0>
UART2 错误	66	000098h	000198h	IFS4<2>	IEC4<2>	IPC16<10:8>
UART2 接收器	30	000050h	000150h	IFS1<14>	IEC1<14>	IPC7<10:8>
UART2 发送器	31	000052h	000152h	IFS1<15>	IEC1<15>	IPC7<14:12>
UART3 错误	81	0000B6h	0001B6h	IFS5<1>	IEC5<1>	IPC20<6:4>
UART3 接收器	82	0000B8h	0001B8h	IFS5<2>	IEC5<2>	IPC20<10:8>
UART3 发送器	83	0000BAh	0001BAh	IFS5<3>	IEC5<3>	IPC20<14:12>
UART4 错误	87	0000C2h	0001C2h	IFS5<7>	IEC5<7>	IPC21<14:12>
UART4 接收器	88	0000C4h	0001C4h	IFS5<8>	IEC5<8>	IPC22<2:0>
UART4 发送器	89	0000C6h	0001C6h	IFS5<9>	IEC5<9>	IPC22<6:4>

### 8.3 中断控制和状态寄存器

PIC24FJ128GA310 系列器件共实现了用于中断控制器的 42 个寄存器：

- INTCON1
- INTCON2
- IFS0 至 IFS7
- IEC0 至 IEC7
- IPC0 至 IPC13、IPC15 和 IPC16，IPC18 至 IPC22、IPC25 和 IPC29
- INTTREG

INTCON1 和 INTCON2 控制全局中断。INTCON1 包含中断嵌套禁止（NSTDIS）位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和备用中断向量表（AIVT）的使用。

IFSx 寄存器包含所有中断请求标志。每个中断源都有一个状态位，由各自的外设或外部信号置 1，且由软件清零。

IECx 寄存器包含所有中断允许位。这些控制位用于单独允许外设或外部信号中断。

IPCx 寄存器用于设置每个中断源的中断优先级。可以给每个用户中断源分配 8 个优先级之一。

INTTREG 寄存器包含相关的中断向量编号和新的 CPU 中断优先级，它们被锁存到 INTTREG 寄存器中的向量编号（VECNUM<6:0>）和中断优先级（ILR<3:0>）位域中。新的中断优先级是待处理中断的优先级。

中断源按其在表 8-2 中的向量编号顺序分配给 IFSx、IECx 和 IPCx 寄存器。例如，INT0（外部中断 0）表示向量编号为 0，且自然优先级为 0。所以 INT0IF 状态位在 IFS0<0>中，INT0IE 允许位在 IEC0<0>中，INT0IP<2:0>优先级位在 IPC0 的最前面的位置（IPC0<2:0>）中。

尽管两个 CPU 控制寄存器都不是中断控制硬件的特定组成部分，但它们仍包含控制中断功能的位。ALU 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。用户可以通过写 IPL 位更改 CPU 的当前优先级。

CORCON 寄存器包含 IPL3 位，此位与 **IPL<2:0>** 位一起指示当前 CPU 优先级。IPL3 是一个只读位，所以用户软件不能屏蔽陷阱事件。

中断控制器具有显示其状态的中断控制器测试寄存器（INTTREG）。产生中断请求时，其相关的向量编号和新的中断优先级被锁存到 INTTREG 中。当一个通用

ISR 用于多个向量（如在自举程序中使用 ISR 重映射）时，该信息可用来确定特定的中断源，或用于检查在 ISR 中是否有其他中断在等待。

在下面各页中的[寄存器 8-1 到寄存器 8-44](#)说明了所有的中断寄存器。

## 寄存器 8-1: SR: ALU 状态寄存器 (CPU 中)

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

R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 <sup>(2,3)</sup>	IPL1 <sup>(2,3)</sup>	IPL0 <sup>(2,3)</sup>	RA <sup>(1)</sup>	N <sup>(1)</sup>	OV <sup>(1)</sup>	Z <sup>(1)</sup>	C <sup>(1)</sup>
bit 7							bit 0

### 图注:

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-9

未实现：读为 0

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：**有关并非专用于中断控制功能的其他位 (bit 0、1、2、3、4 和 8) 的描述，请参见[寄存器 3-1](#)。

**2:** IPL 位与 IPL3 (CORCON<3>) 位一起构成 CPU 的中断优先级。如果 IPL3 = 1，则括号中的值表示中断优先级。

**3:** 当 NSTDIS (INTCON1<15>) = 1 时，IPL 状态位只读。

# PIC24FJ128GA310 系列

寄存器 8-2: CORCON: CPU 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/C-0	r-1	U-0	U-0
—	—	—	—	IPL3 <sup>(1)</sup>	r	—	—
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-4 未实现: 读为 0

bit 3 IPL3: CPU 中断优先级状态位<sup>(1)</sup>

1 = CPU 中断优先级大于 7

0= CPU 中断优先级为 7 或更小

bit 2 保留: 读为 1

bit 1-0 未实现: 读为 0

注 1: IPL3 位与 IPL<2:0> 位 (SR<7:5>) 共同决定 CPU 中断优先级; 关于位描述, 请参见[寄存器 3-2](#)。

## 寄存器 8-3: INTCON1: 中断控制寄存器 1

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

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |          |  |
|----------|--|
| bit 15   | <b>NSTDIS:</b> 中断嵌套禁止位<br>1 = 禁止中断嵌套<br>0 = 允许中断嵌套             |
| bit 14-5 | 未实现: 读为 0  |
| bit 4    | <b>MATHERR:</b> 算法错误陷阱状态位<br>1 = 发生了溢出陷阱<br>0 = 未发生溢出陷阱        |
| bit 3    | <b>ADDRERR:</b> 地址错误陷阱状态位<br>1 = 发生了地址错误陷阱<br>0 = 未发生地址错误陷阱    |
| bit 2    | <b>STKERR:</b> 堆栈错误陷阱状态位<br>1 = 发生了堆栈错误陷阱<br>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<br>1 = 发生了振荡器故障陷阱<br>0 = 未发生振荡器故障陷阱 |
| bit 0    | 未实现: 读为 0  |

# PIC24FJ128GA310 系列

寄存器 8-4: INTCON2: 中断控制寄存器 2

R/W-0	R-0, HSC	U-0	U-0	U-0	U-0	U-0	U-0
ALTIvt	DISI	—	—	—	—	—	—
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
—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP
bit 7	bit 0						

图注:	HSC = 可由硬件置 1/ 清零的位
R = 可读位	W = 可写位
-n = 上电复位时的值	1 = 置 1 0 = 清零 x = 未知

- bit 15      **ALTIvt:** 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（默认）中断向量表
- bit 14      **DISI:** DISI 指令状态位  
1 = 执行了 DISI 指令  
0 = 没有执行 DISI 指令
- bit 13-5     未实现: 读为 0
- bit 4        **INT4EP:** 外部中断 4 边沿检测极性选择位  
1 = 下降沿中断  
0 = 上升沿中断
- bit 3        **INT3EP:** 外部中断 3 边沿检测极性选择位  
1 = 下降沿中断  
0 = 上升沿中断
- bit 2        **INT2EP:** 外部中断 2 边沿检测极性选择位  
1 = 下降沿中断  
0 = 上升沿中断
- bit 1        **INT1EP:** 外部中断 1 边沿检测极性选择位  
1 = 下降沿中断  
0 = 上升沿中断
- bit 0        **INT0EP:** 外部中断 0 边沿检测极性选择位  
1 = 下降沿中断  
0 = 上升沿中断

## 寄存器 8-5: IFS0: 中断标志状态寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IF	AD1IF	U1TXIF	U1RXIF	SPI1IF	SPF1IF	T3IF
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
T2IF	OC2IF	IC2IF	DMA0IF	T1IF	OC1IF	IC1IF	INT0IF
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |        |   |
|--------|---|
| bit 15 | 未实现: 读为 0   |
| bit 14 | <b>AD1IF:</b> A/D 转换完成中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求   |
| bit 13 | <b>DMA1IF:</b> DMA 通道 1 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 12 | <b>U1TXIF:</b> UART1 发送器中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 11 | <b>U1RXIF:</b> UART1 接收器中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 10 | <b>SPI1IF:</b> SPI1 事件中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求   |
| bit 9  | <b>SPF1IF:</b> SPI1 故障中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求   |
| bit 8  | <b>T3IF:</b> Timer3 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 7  | <b>T2IF:</b> Timer2 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 6  | <b>OC2IF:</b> 输出比较通道 2 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 5  | <b>IC2IF:</b> 输入捕捉通道 2 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 4  | <b>DMA0IF:</b> DMA 通道 0 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 3  | <b>T1IF:</b> Timer1 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |

# PIC24FJ128GA310 系列

---

---

## 寄存器 8-5: IFS0: 中断标志状态寄存器 0 (续)

bit 2      **OC1IF:** 输出比较通道 1 中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 1      **IC1IF:** 输入捕捉通道 1 中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 0      **INT0IF:** 外部中断 0 标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

## 寄存器 8-6: IFS1: 中断标志状态寄存器 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
U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	DMA2IF
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
—	IC7IF	—	INT1IF	CNIF	CMIF	MI2C1IF	SI2C1IF
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |        |   |
|--------|---|
| bit 15 | <b>U2TXIF:</b> UART2 发送器中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 14 | <b>U2RXIF:</b> UART2 接收器中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 13 | <b>INT2IF:</b> 外部中断 2 标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 12 | <b>T5IF:</b> Timer5 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 11 | <b>T4IF:</b> Timer4 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 10 | <b>OC4IF:</b> 输出比较通道 4 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 9  | <b>OC3IF:</b> 输出比较通道 3 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 8  | <b>DMA2IF:</b> DMA 通道 2 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 7  | 未实现: 读为 0   |
| bit 6  | <b>IC7IF:</b> 输入捕捉通道 7 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 5  | 未实现: 读为 0   |
| bit 4  | <b>INT1IF:</b> 外部中断 1 标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求     |
| bit 3  | <b>CNIF:</b> 输入电平变化通知中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求    |

# PIC24FJ128GA310 系列

---

---

## 寄存器 8-6: IFS1: 中断标志状态寄存器 1 (续)

bit 2	<b>CMIF:</b> 比较器中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求
bit 1	<b>MI2C1IF:</b> I2C1 主事件中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求
bit 0	<b>SI2C1IF:</b> I2C1 从事件中断标志状态位 1 = 产生了中断请求 0 = 未产生中断请求

## 寄存器 8-7：IFS2：中断标志状态寄存器 2

U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA4IF	MPPIF	—	OC7IF	OC6IF	OC5IF	IC6IF
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
IC5IF	IC4IF	IC3IF	DMA3IF	—	—	SPI2IF	SPF2IF
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |         |   |
|---------|---|
| bit 15  | 未实现：读为 0  |
| bit 14  | <b>DMA4IF:</b> DMA 通道 4 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 13  | <b>MPPIF:</b> 并行主端口中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求      |
| bit 12  | 未实现：读为 0  |
| bit 11  | <b>OC7IF:</b> 输出比较通道 7 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 10  | <b>OC6IF:</b> 输出比较通道 6 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 9   | <b>OC5IF:</b> 输出比较通道 5 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 8   | <b>IC6IF:</b> 输入捕捉通道 6 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 7   | <b>IC5IF:</b> 输入捕捉通道 5 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 6   | <b>IC4IF:</b> 输入捕捉通道 4 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 5   | <b>IC3IF:</b> 输入捕捉通道 3 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求  |
| bit 4   | <b>DMA3IF:</b> DMA 通道 3 中断标志状态位<br>1 = 产生了中断请求<br>0 = 未产生中断请求 |
| bit 3-2 | 未实现：读为 0  |

# PIC24FJ128GA310 系列

## 寄存器 8-7: IFS2: 中断标志状态寄存器 2 (续)

bit 1      **SPI2IF:** SPI2 事件中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 0      **SPF2IF:** SPI2 故障中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

## 寄存器 8-8: IFS3: 中断标志状态寄存器 3

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	RTCIF	DMA5IF	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IF	INT3IF	—	—	MI2C2IF	SI2C2IF	—
bit 7	bit 0						

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      未实现: 读为 0

bit 14      **RTCIF:** 实时时钟 / 日历中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 13      **DMA5IF:** DMA 通道 5 中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 12-7      未实现: 读为 0

bit 6      **INT4IF:** 外部中断 4 标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 5      **INT3IF:** 外部中断 3 标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 4-3      未实现: 读为 0

bit 2      **MI2C2IF:** I2C2 主事件中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 1      **SI2C2IF:** I2C2 从事件中断标志状态位

1 = 产生了中断请求  
0 = 未产生中断请求

bit 0      未实现: 读为 0

## 寄存器 8-9: IFS4: 中断标志状态寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIF	—	—	—	—	LVDIF
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIF	U2ERIF	U1ERIF	—
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-14 未实现: 读为 0
- bit 13 **CTMUIF:** CTMU 中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 12-9 未实现: 读为 0
- bit 8 **LVDIF:** 低电压检测中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 7-4 未实现: 读为 0
- bit 3 **CRCIF:** CRC 发生器中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 2 **U2ERIF:** UART2 错误中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 1 **U1ERIF:** UART1 错误中断标志状态位  
1 = 产生了中断请求  
0 = 未产生中断请求
- bit 0 未实现: 读为 0

# PIC24FJ128GA310 系列

## 寄存器 8-10： IFS5： 中断标志状态寄存器 5

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U4TXIF	U4RXIF
bit 15							bit 8

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIF	—	—	—	U3TXIF	U3RXIF	U3ERIF	—
bit 7							bit 0

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现：读为 0

bit 9 **U4TXIF:** UART4 发送器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 8 **U4RXIF:** UART4 接收器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 7 **U4ERIF:** UART4 错误中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 6-4 未实现：读为 0

bit 3 **U3TXIF:** UART3 发送器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 2 **U3RXIF:** UART3 接收器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 1 **U3ERIF:** UART3 错误中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 0 未实现：读为 0

## 寄存器 8-11: IFS6: 中断标志状态寄存器 6

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

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-5 未实现: 读为 0

bit 4 **LCDIF:** LCD 控制器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 3-0 未实现: 读为 0

## 寄存器 8-12: IFS7: 中断标志状态寄存器 7

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

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5 **JTAGIF:** JTAG 控制器中断标志状态位

1 = 产生了中断请求

0 = 未产生中断请求

bit 4-0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 8-13： IEC0： 中断允许控制寄存器 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA1IE	AD1IE	U1TXIE	U1RXIE	SPI1IE	SPF1IE	T3IE
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
T2IE	OC2IE	IC2IE	DMA0IE	T1IE	OC1IE	IC1IE	INT0IE
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **未实现：** 读为 0
- bit 14      **DMA1IE:** DMA 通道 1 中断标志使能位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 13      **AD1IE:** A/D 转换完成中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 12      **U1TXIE:** UART1 发送器中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 11      **U1RXIE:** UART1 接收器中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 10      **SPI1IE:** SPI1 传输完成中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 9      **SPF1IE:** SPI1 故障中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 8      **T3IE:** Timer3 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 7      **T2IE:** Timer2 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 6      **OC2IE:** 输出比较通道 2 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 5      **IC2IE:** 输入捕捉通道 2 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 4      **DMA0IE:** DMA 通道 0 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 3      **T1IE:** Timer1 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求

## 寄存器 8-13: IEC0: 中断允许控制寄存器 0 (续)

bit 2	<b>OC1IE:</b> 输出比较通道 1 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 1	<b>IC1IE:</b> 输入捕捉通道 1 中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 0	<b>INT0IE:</b> 外部中断 0 允许位 1 = 允许中断请求 0 = 禁止中断请求

# PIC24FJ128GA310 系列

## 寄存器 8-14: IEC1: 中断允许控制寄存器 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
U2TXIE	U2RXIE	INT2IE <sup>(1)</sup>	T5IE	T4IE	OC4IE	OC3IE	DMA2IE
bit 15	bit 8						

U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	IC7IE	—	INT1IE <sup>(1)</sup>	CNIE	CMIE	MI2C1IE	SI2C1IE
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **U2TXIE:** UART2 发送器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 14      **U2RXIE:** UART2 接收器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 13      **INT2IE:** 外部中断 2 使能位<sup>(1)</sup>

1 = 允许中断请求

0 = 禁止中断请求

bit 12      **T5IE:** Timer5 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 11      **T4IE:** Timer4 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 10      **OC4IE:** 输出比较通道 4 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 9      **OC3IE:** 输出比较通道 3 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 8      **DMA2IE:** DMA 通道 2 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 7      未实现: 读为 0

bit 6      **IC7IE:** 输入捕捉通道 7 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 5      未实现: 读为 0

bit 4      **INT1IE:** 外部中断 1 允许位<sup>(1)</sup>

1 = 允许中断请求

0 = 禁止中断请求

**注 1:** 如果允许了外部中断, 则此中断输入必须配置给可用的 RPx 或 RPIx 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 寄存器 8-14: IEC1: 中断允许控制寄存器 1 (续)

bit 3	<b>CNIE:</b> 输入电平变化通知中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 2	<b>CMIE:</b> 比较器中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 1	<b>MI2C1IE:</b> I2C1 主事件中断允许位 1 = 允许中断请求 0 = 禁止中断请求
bit 0	<b>SI2C1IE:</b> I2C1 从事件中断允许位 1 = 允许中断请求 0 = 禁止中断请求

**注 1:** 如果允许了外部中断，则此中断输入必须配置给可用的 RPx 或 RPIx 引脚。更多信息，请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。

# PIC24FJ128GA310 系列

## 寄存器 8-15： IEC2： 中断允许控制寄存器 2

U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	DMA4IE	PMPIE	—	OC7IE	OC6IE	OC5IE	IC6IE
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
IC5IE	IC4IE	IC3IE	DMA3IE	—	—	SPI2IE	SPF2IE
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 未实现：读为 0
- bit 14 **DMA4IE:** DMA 通道 4 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 13 **PMPIE:** 并行主端口中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 12 未实现：读为 0
- bit 11 **OC7IE:** 输出比较通道 7 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 10 **OC6IE:** 输出比较通道 6 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 9 **OC5IE:** 输出比较通道 5 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 8 **IC6IE:** 输入捕捉通道 6 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 7 **IC5IE:** 输入捕捉通道 5 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 6 **IC4IE:** 输入捕捉通道 4 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 5 **IC3IE:** 输入捕捉通道 3 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 4 **DMA3IF:** DMA 通道 3 中断允许位  
1 = 允许中断请求  
0 = 禁止中断请求
- bit 3-2 未实现：读为 0

## 寄存器 8-15: IEC2: 中断允许控制寄存器 2

bit 1      **SPI2IE:** SPI2 事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0      **SPF2IE:** SPI2 故障中断允许位

1 = 允许中断请求

0 = 禁止中断请求

## 寄存器 8-16: IEC3: 中断允许控制寄存器 3

U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
—	RTCIE	DMA5IE	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0
—	INT4IE <sup>(1)</sup>	INT3IE <sup>(1)</sup>	—	—	MI2C2IE	SI2C2IE	—
bit 7	bit 0						

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      未实现: 读为 0

bit 14      **RTCIE:** 实时时钟 / 日历中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 13      **DMA5IE:** DMA 通道 5 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 12-7      未实现: 读为 0

bit 6      **INT4IE:** 外部中断 4 允许位<sup>(1)</sup>

1 = 允许中断请求

0 = 禁止中断请求

bit 5      **INT3IE:** 外部中断 3 允许位<sup>(1)</sup>

1 = 允许中断请求

0 = 禁止中断请求

bit 4-3      未实现: 读为 0

bit 2      **MI2C2IE:** I2C3 主事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 1      **SI2C2IE:** I2C2 从事件中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0      未实现: 读为 0

**注 1:** 如果允许了外部中断, 则此中断输入必须配置给可用的 RPx 或 RPlx 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GA310 系列

寄存器 8-17： IEC4： 中断允许控制寄存器 4

U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
—	—	CTMUIE	—	—	—	—	LVDIE
bit 15							bit 8

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
—	—	—	—	CRCIE	U2ERIE	U1ERIE	—
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13 **CTMUIE:** CTMU 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 12-9 未实现：读为 0

bit 8 **LVDIE:** 低压检测中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 7-4 未实现：读为 0

bit 3 **CRCIE:** CRC 发生器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 2 **U2ERIE:** UART2 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **U1ERIE:** UART1 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 未实现：读为 0

## 寄存器 8-18: IEC5: 中断允许控制寄存器 5

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	U4TXIE	U4RXIE
bit 15						bit 8	

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
U4ERIE	—	—	—	U3TXIE	U3RXIE	U3ERIE	—
bit 7						bit 0	

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9 **U4TXIE:** UART4 发送器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 8 **U4RXIE:** UART4 接收器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 7 **U4ERIE:** UART4 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 6-4 未实现: 读为 0

bit 3 **U3TXIE:** UART3 发送器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 2 **U3RXIE:** UART3 接收器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 1 **U3ERIE:** UART3 错误中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 0 未实现: 读为 0

# PIC24FJ128GA310 系列

## 寄存器 8-19: IEC6: 中断允许控制寄存器 6

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-5 未实现: 读为 0

bit 4 **LCDIE:** LCD 控制器中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 3-0 未实现: 读为 0

## 寄存器 8-20: IEC7: 中断允许控制寄存器 7

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

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5 **JTAGIE:** JTAG 中断允许位

1 = 允许中断请求

0 = 禁止中断请求

bit 4-0 未实现: 读为 0

## 寄存器 8-21：IPC0：中断优先级控制寄存器 0

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T1IP2	T1IP1	T1IP0	—	OC1IP2	OC1IP1	OC1IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC1IP2	IC1IP1	IC1IP0	—	INT0IP2	INT0IP1	INT0IP0
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |           |   |
|-----------|---|
| bit 15    | 未实现：读为 0  |
| bit 14-12 | <b>T1IP&lt;2:0&gt;</b> : Timer1 中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源    |
| bit 11    | 未实现：读为 0  |
| bit 10-8  | <b>OC1IP&lt;2:0&gt;</b> : 输出比较通道 1 中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 7     | 未实现：读为 0  |
| bit 6-4   | <b>IC1IP&lt;2:0&gt;</b> : 输入捕捉通道 1 中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 3     | 未实现：读为 0  |
| bit 2-0   | <b>INT0IP&lt;2:0&gt;</b> : 外部中断 0 优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源    |

# PIC24FJ128GA310 系列

寄存器 8-22: IPC1: 中断优先级控制寄存器 1

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T2IP2	T2IP1	T2IP0	—	OC2IP2	OC2IP1	OC2IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC2IP2	IC2IP1	IC2IP0	—	DMA0IP2	DMA0IP1	DMA0IP0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **T2IP<2:0>**: Timer2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **OC2IP<2:0>**: 输出比较通道 2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **IC2IP<2:0>**: 输入捕捉通道 2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **DMA0IP<2:0>**: DMA 通道 0 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-23: IPC2: 中断优先级控制寄存器 2

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U1RXIP2	U1RXIP1	U1RXIP0	—	SPI1IP2	SPI1IP1	SPI1IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPF1IP2	SPF1IP1	SPF1IP0	—	T3IP2	T3IP1	T3IP0
bit 7	bit 0						

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **U1RXIP<2:0>**: UART1 接收器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **SPI1IP<2:0>**: SPI1 事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **SPF1IP<2:0>**: SPI1 故障中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **T3IP<2:0>**: Timer3 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

# PIC24FJ128GA310 系列

寄存器 8-24: IPC3: 中断优先级控制寄存器 3

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA1IP2	DMA1IP1	DMA1IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	AD1IP2	AD1IP1	AD1IP0	—	U1TXIP2	U1TXIP1	U1TXIP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 DMA1IP<2:0>: DMA 通道 1 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 AD1IP<2:0>: A/D 转换完成中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 U1TXIP<2:0>: UART1 发送器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-25: IPC4: 中断优先级控制寄存器 4

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CNIP2	CNIP1	CNIP0	—	CMIP2	CMIP1	CMIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	MI2C1IP2	MI2C1IP1	MI2C1IP0	—	SI2C1IP2	SI2C1IP1	SI2C1IP0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |           |  |
|-----------|--|
| bit 15    | 未实现: 读为 0  |
| bit 14-12 | <b>CNIP&lt;2:0&gt;</b> : 输入电平变化通知中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源    |
| bit 11    | 未实现: 读为 0  |
| bit 10-8  | <b>CMIP&lt;2:0&gt;</b> : 比较器中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源         |
| bit 7     | 未实现: 读为 0  |
| bit 6-4   | <b>MI2C1IP&lt;2:0&gt;</b> : I2C1 主事件中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 3     | 未实现: 读为 0  |
| bit 2-0   | <b>SI2C1IP&lt;2:0&gt;</b> : I2C1 从事件中断优先级位<br>111 = 中断优先级为 7 (最高优先级中断)<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |

# PIC24FJ128GA310 系列

寄存器 8-26： IPC5： 中断优先级控制寄存器 5

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	IC7IP2	IC7IP1	IC7IP0
bit 15							bit 8

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT1IP2	INT1IP1	INT1IP0
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现：读为 0

bit 10-8 **IC7IP<2:0>**：输入捕捉通道 7 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7-3 未实现：读为 0

bit 2-0 **INT1IP<2:0>**：外部中断 1 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-27: IPC6: 中断优先级控制寄存器 6

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	T4IP2	T4IP1	T4IP0	—	OC4IP2	OC4IP1	OC4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC3IP2	OC3IP1	OC3IP0	—	DMA2IP2	DMA2IP1	DMA2IP0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **T4IP<2:0>**: Timer4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现: 读为 0

bit 10-8 **OC4IP<2:0>**: 输出比较通道 4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **OC3IP<2:0>**: 输出比较通道 3 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **DMA2IP<2:0>**: DMA 通道 2 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

# PIC24FJ128GA310 系列

寄存器 8-28： IPC7： 中断优先级控制寄存器 7

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U2TXIP2	U2TXIP1	U2TXIP0	—	U2RXIP2	U2RXIP1	U2RXIP0
bit 15							

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	INT2IP2	INT2IP1	INT2IP0	—	T5IP2	T5IP1	T5IP0
bit 7							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现：读为 0

bit 14-12 **U2TXIP<2:0>**: UART2 发送器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现：读为 0

bit 10-8 **U2RXIP<2:0>**: UART2 接收器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现：读为 0

bit 6-4 **INT2IP<2:0>**: 外部中断 2 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现：读为 0

bit 2-0 **T5IP<2:0>**: Timer5 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-29: IPC8: 中断优先级控制寄存器 8

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	SPI2IP2	SPI2IP1	SPI2IP0	—	SPF2IP2	SPF2IP1	SPF2IP0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **SPI2IP<2:0>: SPI2 事件中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现: 读为 0

bit 2-0 **SPF2IP<2:0>: SPI2 故障中断优先级位**

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

# PIC24FJ128GA310 系列

## 寄存器 8-30：IPC9：中断优先级控制寄存器 9

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC5IP2	IC5IP1	IC5IP0	—	IC4IP2	IC4IP1	IC4IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	IC3IP2	IC3IP1	IC3IP0	—	DMA3IP2	DMA3IP1	DMA3IP0
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现：读为 0

bit 14-12 **IC5IP<2:0>**：输入捕捉通道 5 中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现：读为 0

bit 10-8 **IC4IP<2:0>**：输入捕捉通道 4 中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现：读为 0

bit 6-4 **IC3IP<2:0>**：输入捕捉通道 3 中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现：读为 0

bit 2-0 **DMA3IP<2:0>**：DMA 通道 3 中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-31：IPC10：中断优先级控制寄存器 10

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC7IP2	OC7IP1	OC7IP0	—	OC6IP2	OC6IP1	OC6IP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	OC5IP2	OC5IP1	OC5IP0	—	IC6IP2	IC6IP1	IC6IP0
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |           |   |
|-----------|---|
| bit 15    | 未实现：读为 0  |
| bit 14-12 | <b>OC7IP&lt;2:0&gt;</b> ：输出比较通道 7 中断优先级位<br>111 = 中断优先级为 7（最高优先级中断）<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 11    | 未实现：读为 0  |
| bit 10-8  | <b>OC6IP&lt;2:0&gt;</b> ：输出比较通道 6 中断优先级位<br>111 = 中断优先级为 7（最高优先级中断）<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 7     | 未实现：读为 0  |
| bit 6-4   | <b>OC5IP&lt;2:0&gt;</b> ：输出比较通道 5 中断优先级位<br>111 = 中断优先级为 7（最高优先级中断）<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |
| bit 3     | 未实现：读为 0  |
| bit 2-0   | <b>IC6IP&lt;2:0&gt;</b> ：输入捕捉通道 6 中断优先级位<br>111 = 中断优先级为 7（最高优先级中断）<br>•<br>•<br>•<br>001 = 中断优先级为 1<br>000 = 禁止中断源 |

# PIC24FJ128GA310 系列

寄存器 8-32: IPC11: 中断优先级控制寄存器 11

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	DMA4IP2	DMA4IP1	DMA4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	PMPIP2	PMPIP1	PMPIP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **DMA4IP<2:0>**: DMA 通道 4 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **PMPIP<2:0>**: 并行主端口中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

## 寄存器 8-33: IPC12: 中断优先级控制寄存器 12

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	MI2C2IP2	MI2C2IP1	MI2C2IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	SI2C2IP2	SI2C2IP1	SI2C2IP0	—	—	—	—
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **MI2C2IP<2:0>**: I2C2 主事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **SI2C2IP<2:0>**: I2C2 从事件中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 8-34: IPC13: 中断优先级控制寄存器 13

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	INT4IP2	INT4IP1	INT4IP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	INT3IP2	INT3IP1	INT3IP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 INT4IP<2:0>: 外部中断 4 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 INT3IP<2:0>: 外部中断 3 优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

## 寄存器 8-35: IPC15: 中断优先级控制寄存器 15

U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-0	R/W-0
—	—	—	—	—	RTCIP2	RTCIP1	RTCIP0
bit 15							bit 8

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	DMA5IP2	DMA5IP1	DMA5IP0	—	—	—	—
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **RTCIP<2:0>**: 实时时钟 / 日历中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现: 读为 0

bit 6-4 **DMA5IP<2:0>**: DMA 通道 5 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

# PIC24FJ128GA310 系列

## 寄存器 8-36： IPC16： 中断优先级控制寄存器 16

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	CRCIP2	CRCIP1	CRCIP0	—	U2ERIP2	U2ERIP1	U2ERIP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U1ERIP2	U1ERIP1	U1ERIP0	—	—	—	—
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现：读为 0

bit 14-12 **CRCIP<2:0>**: CRC 发生器错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11 未实现：读为 0

bit 10-8 **U2ERIP<2:0>**: UART2 错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 7 未实现：读为 0

bit 6-4 **U1ERIP<2:0>**: UART1 错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现：读为 0

## 寄存器 8-37: IPC18: 中断优先级控制寄存器 18

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-1	R/W-0	R/W-0
—	—	—	—	—	LVDIP2	LVDIP1	LVDIP0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 未实现: 读为 0

bit 2-0 **LVDIP<2:0>**: 低电压检测中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-38: IPC19: 中断优先级控制寄存器 19

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	CTMUIP2	CTMUIP1	CTMUIP0	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **CTMUIP<2:0>**: CTMU 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 8-39： IPC20： 中断优先级控制寄存器 20

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U3TXIP2	U3TXIP1	U3TXIP0	—	U3RXIP2	U3RXIP1	U3RXIP0
bit 15	bit 8						

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U3ERIP2	U3ERIP1	U3ERIP0	—	—	—	—
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15 未实现：读为 0
- bit 14-12 **U3TXIP<2:0>：UART3 发送器中断优先级位**  
111 = 中断优先级为 7 (最高优先级中断)  
•  
•  
•  
001 = 中断优先级为 1  
000 = 禁止中断源
- bit 11 未实现：读为 0
- bit 10-8 **U3RXIP<2:0>：UART3 接收器中断优先级位**  
111 = 中断优先级为 7 (最高优先级中断)  
•  
•  
•  
001 = 中断优先级为 1  
000 = 禁止中断源
- bit 7 未实现：读为 0
- bit 6-4 **U3ERIP<2:0>：UART3 错误中断优先级位**  
111 = 中断优先级为 7 (最高优先级中断)  
•  
•  
•  
001 = 中断优先级为 1  
000 = 禁止中断源
- bit 3-0 未实现：读为 0

## 寄存器 8-40: IPC21: 中断优先级控制寄存器 21

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	U4ERIP2	U4ERIP1	U4ERIP0	—	—	—	—
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **U4ERIP<2:0>**: UART4 错误中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 11-0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 8-41： IPC22： 中断优先级控制寄存器 22

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

U-0	R/W-1	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0
—	U4TXIP2	U4TXIP1	U4TXIP0	—	U4RXIP2	U4RXIP1	U4RXIP0
bit 7							bit 0

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现：读为 0

bit 6-4 **U4TXIP<2:0>**：UART4 发送器中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3 未实现：读为 0

bit 2-0 **U4RXIP<2:0>**：UART4 接收器中断优先级位

111 = 中断优先级为 7（最高优先级中断）

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-42: IPC25: 中断优先级控制寄存器 25

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-1	R/W-0	R/W-0
—	—	—	—	—	LCDIP2	LCDIP1	LCDIP0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 未实现: 读为 0

bit 2-0 **LCDIP<2:0>**: LCD 控制器中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

## 寄存器 8-43: IPC29: 中断优先级控制寄存器 29

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

U-0	R/W-1	R/W-0	R/W-0	U-0	U-0	U-0	U-0
—	JTAGIP2	JTAGIP1	JTAGIP0	—	—	—	—
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-7 未实现: 读为 0

bit 6-4 **JTAGIP<2:0>**: JTAG 中断优先级位

111 = 中断优先级为 7 (最高优先级中断)

•

•

•

001 = 中断优先级为 1

000 = 禁止中断源

bit 3-0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 8-44: INTTREG: 中断控制器测试寄存器

R-0, HSC	U-0	R/W-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
CPUIRQ	—	VHOLD	—	ILR3	ILR2	ILR1	ILR0
bit 15	bit 8						

U-0	R-0, HSC						
—	VECNUM6	VECNUM5	VECNUM4	VECNUM3	VECNUM2	VECNUM1	VECNUM0
bit 7	bit 0						

图注:

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **CPUIRQ:** 来自中断控制器 CPU 的中断请求位  
1 = 产生了中断请求, 但 CPU 尚未应答。这发生在 CPU 的优先级高于所请求的中断优先级时  
0 = 未应答任何中断请求
- bit 14      未实现: 读为 0
- bit 13      **VHOLD:** 向量编号捕捉配置位  
1 = VECNUM 位保存优先级最高的待处理中断的值  
0 = VECNUM 位保存上一次应答的中断值 (即已发生且优先级高于 CPU 的上一个中断, 即使有其他待处理中断时也是如此)
- 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      未实现: 读为 0
- bit 6-0     **VECNUM<5:0>:** 待处理中断或上一次应答的中断的向量编号位  
VHOLD = 1: VECNUM 位指示上一次发生的中断的向量编号 (0-118)  
VHOLD = 0: VECNUM 位指示目前正在处理的中断请求的向量编号 (0-118)

## 8.4 中断设置过程

### 8.4.1 初始化

按以下步骤配置中断源：

1. 如果不需要嵌套中断，则将 **NSTDIS** 控制位 (**INTCON1<15>**) 置 1。
2. 通过写相应的 **IPCx** 寄存器中的控制位来为中断源选择用户分配的优先级。优先级将取决于具体的应用和中断源的类型。如果不需要多个优先级，可以将所有允许的中断源的 **IPCx** 寄存器控制位编程为相同的非零值。
3. 将相应 **IFSx** 寄存器中与外设相关的中断标志状态位清零。
4. 通过将相应 **IECx** 寄存器中与中断源相关的中断允许控制位置 1 来允许中断源。

**注：** 在器件复位时，**IPCx** 寄存器被初始化，为所有用户中断源分配优先级 4。

### 8.4.2 中断服务程序 (ISR)

用于声明中断服务程序 (ISR) 以及使用正确的向量地址初始化 IVT 的方法取决于编程语言 (即 C 语言或汇编语言) 和用于开发应用程序的语言开发工具套件。一般情况下，用户必须将相应 **IFSx** 寄存器中与 ISR 处理的中断源相对应的中断标志清零；否则，在退出 ISR 后会立即再次进入 ISR。如果 ISR 用汇编语言编码，则必须使用 **RETFIE** 指令结束 ISR，以便将保存的 PC 值、SRL 值和原先的 CPU 优先级弹出堆栈。

### 8.4.3 陷阱服务程序 (TSR)

陷阱服务程序 (Trap Service Routine, TSR) 的编码方式类似于 ISR，只是必须将 **INTCON1** 寄存器中相应的陷阱状态标志清零，以避免重新进入 TSR。

### 8.4.4 禁止中断

可以通过以下步骤禁止所有用户中断：

1. 使用 **PUSH** 指令将当前 SR 值压入软件堆栈。
2. 通过将值 **0Eh** 与 **SRL** 进行逻辑或运算来强制把 CPU 的优先级设置为 7。

要允许用户中断，可以使用 **POP** 指令恢复先前的 **SR** 值。

注意只能禁止优先级小于或等于 7 的用户中断。不能禁止陷阱源（优先级为 8-15）。

使用 **DISI** 指令可以方便地将优先级为 1-6 的中断禁止一段固定的时间。**DISI** 指令不能禁止优先级为 7 的中断源。

# **PIC24FJ128GA310 系列**

---

---

注:

## 9.0 振荡器配置

**注：**本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息，请参见《PIC24F 系列参考手册》中的第 6 章“振荡器”(DS39700C\_CN)。

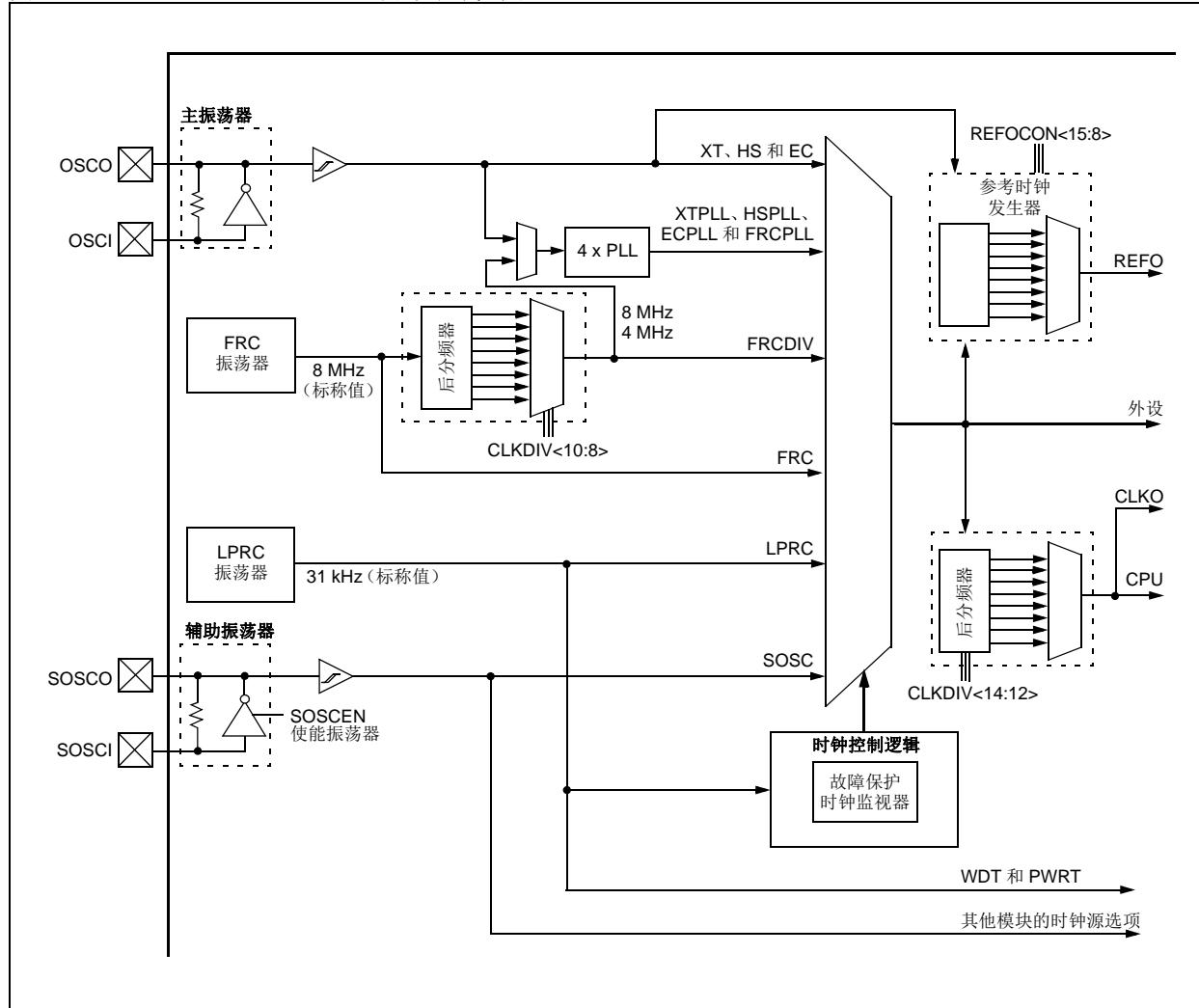
PIC24FJ128GA310 系列器件的振荡器系统具有以下特性：

- 共有四个外部和内部振荡器可选作为时钟源，提供 11 种不同的时钟模式
- 片内 4x PLL 模块可在选择内部和外部振荡器源时提高内部工作频率

- 可采用软件控制在多个时钟源之间切换
- 由软件控制的后分频器对 CPU 的时钟进行选择以节省系统的功耗
- 具有故障保护时钟监视器(FSCM)，可检测时钟故障，以使应用可安全地恢复或关闭
- 可单独使用且独立配置的系统时钟输出，以同步外部硬件

图 9-1 所示为振荡器系统的简化框图。

图 9-1：PIC24FJ128GA310 系列时钟框图



## 9.1 CPU 时钟机制

系统时钟源可由以下四种时钟源之一提供：

- OSC1 和 OSC0 引脚上的主振荡器（POSC）
- SOSCI 和 SOSCO 引脚上的辅助振荡器（SOSC）
- 内部快速 RC（FRC）振荡器
- 内部低功耗 RC（LPRC）振荡器

主振荡器和 FRC 源可使用内部  $4\times$  PLL。还可选择使用可编程时钟分频器将 FRC 时钟源的频率降低。选择的时钟源提供处理器和外设的时钟。

将处理器时钟源二分频可以产生内部指令周期时钟  $F_{CY}$ 。在本文档中，指令周期时钟由  $F_{osc}/2$  表示。在 OSCO I/O 引脚上可以提供内部指令周期时钟  $F_{osc}/2$ ，用于主振荡器的某些工作模式。

## 9.2 上电复位时的初始配置

通过使用配置位设置选择器件发生上电复位事件时使用的振荡器源（和工作模式）。振荡器配置位设置位于程序存储器中的配置寄存器中（更多详细信息请参见第 29.0 节“[特殊功能](#)”）。主振荡器配置位 POSCMD<1:0>（配置字 2<1:0>）和初始振荡器选择配置位 FNOSC<2:0>（配置字 2<10:8>）选择上电复位时使用的振荡器源。带有后分频器的 FRC 主振荡器（FRCDIV）是默认（未编程）的选择。对这些位进行编程可以选择辅助振荡器或其中一个内部振荡器。

配置位允许用户在多个时钟模式中进行选择，如表 9-1 所示。

### 9.2.1 时钟切换模式配置位

FCKSM 配置位（配置字 2<7:6>）用于联合配置器件时钟切换和故障保护时钟监视器（FSCM）。只有将 FCKSM1 编程为 0 时才可以使能时钟切换功能。只有将 FCKSM<1:0> 位编程为 00 时才可以使能 FSCM。

表 9-1：用于时钟选择的配置位值

振荡器模式	振荡器源	POSCMD<1:0>	FNOSC<2:0>	注
带有后分频器（FRCDIV）的快速 RC 振荡器	内部	11	111	<a href="#">1, 2</a>
（保留）	内部	xx	110	<a href="#">1</a>
低功耗 RC 振荡器（LPRC）	内部	11	101	<a href="#">1</a>
辅助（Timer1）振荡器（SOSC）	辅助	11	100	<a href="#">1</a>
带有 PLL 模块的主振荡器（XT）（XTPLL）	主	01	011	
带有 PLL 模块的主振荡器（EC）（ECPLL）	主	00	011	
主振荡器（HS）	主	10	010	
主振荡器（XT）	主	01	010	
主振荡器（EC）	主	00	010	
带有 PLL 模块的快速 RC 振荡器（FRCPLL）	内部	11	001	<a href="#">1</a>
快速 RC 振荡器（FRC）	内部	11	000	<a href="#">1</a>

注 1：OSCO 引脚功能由 OSCIOFCN 配置位决定。

2：对于未编程（已擦除）器件，这是默认的振荡器模式。

### 9.3 控制寄存器

振荡器的操作由三个特殊功能寄存器控制：

- OSCCON
- CLKDIV
- OSCTUN

OSCCON 寄存器（[寄存器 9-1](#)）是振荡器的主控制寄存器。它控制时钟源的切换并允许监视时钟源。

CLKDIV 寄存器（[寄存器 9-2](#)）控制打盹模式的相关特性，以及 FRC 振荡器的后分频器。

OSCTUN 寄存器（[寄存器 9-3](#)）允许用户在大约  $\pm 12\%$  的范围内精确地调整 FRC 振荡器。每次位递增或递减时，FRC 振荡器的出厂校准频率都会发生一定量的变化。

**寄存器 9-1： OSCCON：振荡器控制寄存器**

U-0	R-0	R-0	R-0	U-0	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>	R/W-x <sup>(1)</sup>
—	COSC2	COSC1	COSC0	—	NOSC2	NOSC1	NOSC0
bit 15							bit 8

R/SO-0	R/W-0	R-0 <sup>(3)</sup>	U-0	R/CO-0	R/W-0	R/W-0	R/W-0
CLKLOCK	IOLOCK <sup>(2)</sup>	LOCK	—	CF	POSCEN	SOSCEN	OSWEN
bit 7							bit 0

图注：

CO = 只可清零位

SO = 只能置 1 的位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

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

111 = 带后分频器的快速 RC 振荡器 (FRCDIV)

110 = 保留

101 = 低功耗 RC 振荡器 (LPRC)

100 = 辅助振荡器 (SOSC)

011 = 带 PLL 模块的主振荡器 (XTPLL、HSPLL 和 ECPLL)

010 = 主振荡器 (XT、HS 和 EC)

001 = 带后分频器和 PLL 模块的快速 RC 振荡器 (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>(1)</sup>

111 = 带后分频器的快速 RC 振荡器 (FRCDIV)

110 = 保留

101 = 低功耗 RC 振荡器 (LPRC)

100 = 辅助振荡器 (SOSC)

011 = 带 PLL 模块的主振荡器 (XTPLL、HSPLL 和 ECPLL)

010 = 主振荡器 (XT、HS 和 EC)

001 = 带后分频器和 PLL 模块的快速 RC 振荡器 (FRCPLL)

000 = 快速 RC 振荡器 (FRC)

**注 1：** 这些位的复位值由 FNOSC 配置位决定。

**2：** 只能在执行解锁序列后更改 IOLOCK 位的状态。另外，如果 IOL1WAY 配置位为 1，一旦 IOLOCK 位置 1，就不能清零。

**3：** 在进行有效的时钟切换或选择非 PLL 时钟模式时，此位也复位为 0。

# PIC24FJ128GA310 系列

---

## 寄存器 9-1: OSCCON: 振荡器控制寄存器 (续)

bit 7	<b>CLKLOCK:</b> 时钟选择锁定使能位 <u>如果使能 FSCM (FCKSM1 = 1) :</u> 1 = 时钟和 PLL 选择被锁定 0 = 时钟和 PLL 选择未锁定, 可通过将 OSWEN 位置 1 来进行修改 <u>如果禁止 FSCM (FCKSM1 = 0) :</u> 时钟和 PLL 选择始终未锁定, 可通过将 OSWEN 位置 1 来进行修改。
bit 6	<b>IOLOCK:</b> I/O 锁定使能位 <sup>(2)</sup> 1 = I/O 锁定有效 0 = I/O 锁定无效
bit 5	<b>LOCK:</b> PLL 锁定状态位 <sup>(3)</sup> 1 = PLL 模块处于锁定状态或 PLL 模块的起振定时器延时结束 0 = PLL 模块不处于锁定状态, PLL 起振定时器正在运行或 PLL 被禁止
bit 4	未实现: 读为 0
bit 3	<b>CF:</b> 时钟故障检测位 1 = FSCM 检测到一个时钟故障 0 = 未检测到时钟故障
bit 2	<b>POSCEN:</b> 主振荡器休眠使能位 1 = 主振荡器在休眠模式期间继续工作 0 = 主振荡器在休眠模式期间被禁止
bit 1	<b>SOSCEN:</b> 32 kHz 辅助振荡器 (SOSC) 使能位 1 = 使能辅助振荡器 0 = 禁止辅助振荡器
bit 0	<b>OSWEN:</b> 振荡器切换使能位 1 = 启动振荡器切换, 切换到由 NOSC<2:0> 位指定的时钟源 0 = 完成振荡器切换

**注 1:** 这些位的复位值由 FNOSC 配置位决定。

**2:** 只能在执行解锁序列后更改 IOLOCK 位的状态。另外, 如果 IOL1WAY 配置位为 1, 一旦 IOLOCK 位置 1, 就不能清零。

**3:** 在进行有效的时钟切换或选择非 PLL 时钟模式时, 此位也复位为 0。

## 寄存器 9-2: CLKDIV: 时钟分频器寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-1
ROI	DOZE2	DOZE1	DOZE0	DOZEN <sup>(1)</sup>	RCDIV2	RCDIV1	RCDIV0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **ROI:** 中断时恢复位

1 = 中断时清零 DOZEN 位并将 CPU/ 外设的时钟比复位为 1:1

0 = 中断不影响 DOZEN 位

bit 14-12    **DOZE<2:0>:** CPU/ 外设的时钟比选择位

111 = 1:128

110 = 1:64

101 = 1:32

100 = 1:16

011 = 1:8

010 = 1:4

001 = 1:2

000 = 1:1

bit 11      **DOZEN:** DOZE 使能位<sup>(1)</sup>

1 = DOZE<2:0> 位指定 CPU/ 外设时钟比

0 = 将 CPU 和外设的时钟比设置为 1:1

bit 10-8     **RCDIV<2:0>:** FRC 后分频比选择位

111 = 31.25 kHz (256 分频)

110 = 125 kHz (64 分频)

101 = 250 kHz (32 分频)

100 = 500 kHz (16 分频)

011 = 1 MHz (8 分频)

010 = 2 MHz (4 分频)

001 = 4 MHz (2 分频)

000 = 8 MHz (1 分频)

bit 7-0      未实现: 读为 0

**注 1:** 当 ROI 位置 1 并发生中断时, 该位会自动清零。

# PIC24FJ128GA310 系列

寄存器 9-3： 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
—	—	TUN5 <sup>(1)</sup>	TUN4 <sup>(1)</sup>	TUN3 <sup>(1)</sup>	TUN2 <sup>(1)</sup>	TUN1 <sup>(1)</sup>	TUN0 <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 = 上电复位时的值

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<sup>(1)</sup>

011111 = 最大频率偏差

011110 =

•

•

•

000001 =

000000 = 中心频率，振荡器运行在经过出厂前校准的频率上

111111 =

•

•

•

100001 =

100000 = 最小频率偏差

注 1： TUN<5:0> 的增加或减少不能在 FRC 调节范围内同步更改 FRC 频率且可能不具有单调性。

## 9.4 时钟切换工作原理

应用可在软件控制下随时在四个时钟源（POSC、SOSC、FRC 和 LPRC）间自由切换，几乎没有什限制。为限制该灵活性可能带来的负面影响，PIC24F 器件在时钟切换过程中采用了安全锁定。

注： 主振荡器模式有三种不同的子模式（XT、HS 和 EC），这三个子模式由 POSCMDx 配置位决定。在应用中可以用软件实现从主振荡器模式切换到其他模式，或从其他模式切换到主振荡器模式，但不能在不对器件进行再编程的情况下在主振荡器模式的不同子模式之间进行切换。

### 9.4.1 使能时钟切换

要使能时钟切换，必须将 CW2 的 FCKSM 配置位编程为 00。（更多详细信息请参见第 29.1 节“配置位”。）如果没有对 FCKSM 配置位编程（1x），则时钟切换功能和故障保护时钟监视器功能都将被禁止。这是默认设置。

在时钟切换被禁止的情况下，NOSCx 控制位（OSCCON<10:8>）不控制时钟选择。但是，COSCx 位（OSCCON<14:12>）将会反映 FNOSCx 配置位选择的时钟源。

禁止了时钟切换功能时，OSWEN 控制位（OSCCON<0>）不起作用。它始终保持为 0。

## 9.4.2 振荡器切换序列

执行时钟切换至少需要下列基本序列：

1. 如果需要，读 **COSC<sub>x</sub>** 位 (**OSCCON<14:12>**) 确定当前振荡器源。
2. 执行解锁序列以允许写入 **OSCCON** 寄存器的高字节。
3. 将新振荡器源的相关值写入 **NOSC<sub>x</sub>** 位 (**OSCCON<10:8>**)。
4. 执行解锁序列以允许写入 **OSCCON** 寄存器的低字节。
5. 将 **OSWEN** 位置 1 来启动振荡器切换。

一旦基本序列完成，系统时钟硬件将自动进行如下响应：

1. 时钟切换硬件将 **NOSC<sub>x</sub>** 控制位的新值和 **COSC<sub>x</sub>** 位做比较。如果相等，时钟切换为冗余操作。在这种情况下，**OSWEN** 位被自动清零且时钟切换被中止。
2. 如果启动了有效时钟切换，则 **LOCK** (**OSCCON<5>**) 和 **CF** (**OSCCON<3>**) 位被清零。
3. 如果新振荡器现在不运行，硬件会将其开启。如果必须打开晶振，硬件将等待直至 **OST** 超时。如果新振荡源正在使用 **PLL**，硬件将等待直到检测到 **PLL** 锁定 (**LOCK = 1**)。
4. 硬件会等待新时钟源的 10 个时钟周期，然后执行时钟切换。
5. 硬件清零 **OSWEN** 位以表示时钟切换成功。此外，**NOSC<sub>x</sub>** 位的值被传送到 **COSC<sub>x</sub>** 位中。
6. 此时旧时钟源被关闭，但 **LPRC** (如果使能了 **WDT** 或 **FSCM**) 或 **SOSC** (如果 **SOSCEN** 位保持置 1 状态) 除外。

- 注 1:** 在整个时钟切换过程中，处理器将继续执行代码。对时序敏感的代码不应在此时执行。
- 2:** 不允许直接在使能 **PLL** 的任何主振荡器模式和使能 **FRCPLL** 的任何主振荡器模式之间进行时钟切换。这适用于任一方向的时钟切换。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。

时钟切换的建议代码序列通常包括：

1. 在 **OSCCON** 寄存器解锁和写序列过程中禁止中断。
2. 通过两条连续的指令将 **78h** 和 **9Ah** 写入 **OSCCON<15:8>** 执行 **OSCCON** 高字节的解锁序列。
3. 执行解锁序列之后，立即使用指令将新振荡器源写入 **NOSC<sub>x</sub>** 位。
4. 通过两条连续的指令将 **46h** 和 **57h** 写入 **OSCCON<7:0>** 执行 **OSCCON** 低字节的解锁序列。
5. 执行解锁序列之后，立即使用指令将 **OSWEN** 位置 1。
6. 继续执行对时钟不敏感的代码（可选）。
7. 调用适当时间的软件延时（周期计数），允许选定的振荡器和 / 或 **PLL** 启动并稳定下来。
8. 检查 **OSWEN** 位是否为 0。如果是，则切换成功。如果 **OSWEN** 位仍为置 1 状态，则检查 **LOCK** 位以确定故障原因。

解锁 **OSCCON** 寄存器并启动时钟切换的核心序列如例 9-1 所示。

### 例 9-1：时钟切换的基本代码序列

```
;Place the new oscillator selection in W0
;OSCCONH (high byte) Unlock Sequence
MOV      #OSCCONH, w1
MOV      #0x78, w2
MOV      #0x9A, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Set new oscillator selection
MOV.b   WREG, OSCCONH
;OSCCONL (low byte) unlock sequence
MOV      #OSCCONL, w1
MOV      #0x46, w2
MOV      #0x57, w3
MOV.b   w2, [w1]
MOV.b   w3, [w1]
;Start oscillator switch operation
BSET    OSCCON, #0
```

## 9.5 辅助振荡器 (SOSC)

### 9.5.1 基本 SOSC 工作

PIC24FJ128GA310 系列器件不必置 1 SOSCEN 位来使用辅助振荡器。任何需要 SOSC 的模块（例如 RTCC、Timer1 或 DSWDT）在需要时钟信号时将自动开启 SOSC。但是，SOSC 的起振时间较长。要避免外设启动时产生延迟，可使用 SOSCEN 位手动启动 SOSC。

要使用辅助振荡器，必须将 SOSCSEL<1:0> 位 (CW3<9:8>) 配置为某种振荡器模式——11 或 01。SOSCSEL 位设置为 00 时可配置 SOSC 引脚为数字模式，从而使能该引脚上的数字 I/O 功能。如果 SOSC 配置为任一振荡器模式，则数字功能将不可用。

### 9.5.2 低功耗 SOSC 工作

辅助振荡器可根据器件配置的要求工作在两种不同的功耗级别下。在低功耗模式下，振荡器工作在低驱动强度、低功耗状态下。默认情况下，振荡器使用高驱动强度，因而需要更多的功耗。辅助振荡器模式配置位 SOSCSEL<1:0> (CW3<9:8>) 可决定辅助振荡器的功耗模式。将 SOSCSEL 编程为 01 选择低功耗工作模式。

由于此模式具有低驱动强度，使得 SOSC 对噪声更加敏感，从而需要较长的起振时间。使用低功耗模式时，必须小心设计和布置 SOSC 电路，以确保振荡器正确启动并振荡。

### 9.5.3 外部（数字）时钟模式 (SCLKI)

SOSC 也可配置为使用外部 32 kHz 时钟源而不是内部振荡器提供的时钟。在此模式（也称作数字模式）下，可使用在 SCLKI 引脚上提供的时钟源为配置为使用辅助振荡器的任何模式提供时钟。在此模式下，晶振驱动电路被禁止，SOSCEN 位 (OSCCON<1>) 不起作用。

### 9.5.4 SOSC 布线注意事项

低引脚数器件（例如 PIC24FJ128GA310 系列中的低引脚数器件）引脚排列的局限性使得 SOSC 比其他 PIC24FJ 器件对噪声更加敏感。除非在 SOSC 的设计和布线过程中非常小心，否则这种外部噪声会使振荡器的周期不精确。

通常情况下，晶振电路的连接应尽可能短。在晶振电路周围布置一个接地回路或接地板也是一种好的做法。关于晶振电路设计的更多信息，请参见《PIC24F 系列参考手册》的第 6 章“振荡器”(DS39700C\_CN)。其他信息也可在以下这些 Microchip 应用笔记中找到：

- AN826, “*Crystal Oscillator Basics and Crystal Selection for rfPIC® and PICmicro® Devices*” (DS00826)
- AN849, “*Basic PICmicro® Oscillator Design*” (DS00849)。

## 9.6 参考时钟输出

除了某些振荡器模式中使用的 CLKO 输出 (Fosc/2) 外，还可对 PIC24FJ128GA310 系列器件中的器件时钟进行配置，以为端口引脚提供参考时钟输出信号。该功能在所有振荡器配置中都可用，允许用户选择更大范围的时钟分频比来驱动应用中的外部器件。

该参考时钟输出由 REFOCON 寄存器（[寄存器 9-4](#)）控制。置 1 ROEN 位 (REFOCON<15>) 可使 REFO 引脚上的时钟信号可用。RODIV 位 (REFOCON<11:8>) 允许选择 16 种不同的时钟分频比选项。

ROSSLP 和 ROSEL 位 (REFOCON<13:12>) 控制休眠模式期间参考时钟输出是否可用。ROSEL 位决定用 OSC1 和 OSC2 上的振荡器还是当前系统时钟源作为参考时钟输出。ROSSLP 位决定当器件处于休眠模式时 REFO 上的参考时钟源是否可用。

要在休眠模式下使用参考时钟输出，ROSSLP 和 ROSEL 位都必须置 1。器件时钟也必须配置为其中一种主模式 (EC、HS 或 XT)。否则，如果 POSCEN 位也没有置 1，那么当器件进入休眠模式时，OSC1 和 OSC2 上的振荡器会掉电。任何时钟切换期间清零 ROSEL 位都会允许参考时钟输出频率随着系统时钟的变化而变化。

## 寄存器 9-4: REFOCON: 参考振荡器控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ROEN	—	ROSSL	ROSEL	RODIV3	RODIV2	RODIV1	RODIV0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |          |   |
|----------|---|
| bit 15   | <b>ROEN:</b> 参考振荡器输出使能位<br>1 = 使能 REFO 引脚上的参考振荡器<br>0 = 禁止参考振荡器   |
| bit 14   | 未实现: 读为 0   |
| bit 13   | <b>ROSSL:</b> 参考振荡器输出在休眠模式下停止的位<br>1 = 休眠模式下参考振荡器继续工作<br>0 = 休眠模式下禁止参考振荡器   |
| bit 12   | <b>ROSEL:</b> 参考振荡源选择位<br>1 = 主振荡器用作基本时钟。注意, 必须使用 FOSC<2:0> 位使能晶振; 晶振在休眠模式下继续工作。<br>0 = 系统时钟用作基本时钟; 基本时钟反映了器件的任何时钟切换  |
| bit 11-8 | <b>RODIV&lt;3:0&gt;:</b> 参考振荡器分频比选择位<br>1111 = 时基值的 32,768 分频<br>1110 = 时基值的 16,384 分频<br>1101 = 时基值的 8,192 分频<br>1100 = 时基值的 4,096 分频<br>1011 = 时基值的 2,048 分频<br>1010 = 时基值的 1,024 分频<br>1001 = 时基值的 512 分频<br>1000 = 时基值的 256 分频<br>0111 = 时基值的 128 分频<br>0110 = 时基值的 64 分频<br>0101 = 时基值的 32 分频<br>0100 = 时基值的 16 分频<br>0011 = 时基值的 8 分频<br>0010 = 时基值的 4 分频<br>0001 = 时基值的 2 分频<br>0000 = 时基值 |
| bit 7-0  | 未实现: 读为 0   |

# **PIC24FJ128GA310 系列**

---

---

注:

## 10.0 节能特性

**注:** 本数据手册总结了该组 PIC24FJ 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息, 请参见《PIC24F 系列参考手册》中的第 57 章“带 VBAT 的节能特性”(DS30622A\_CN)。

PIC24FJ128GA310 系列器件提供了管理功耗的功能, 该功能是通过选择性地管理 CPU 和外设的时钟来实现的。一般而言, 较低的时钟频率和减少时钟源驱动电路的数目会降低功耗。

PIC24FJ128GA310 系列器件通过以下五种策略管理功耗:

- 基于指令的降耗模式
- 基于硬件的降耗特性
- 时钟频率控制
- 软件控制的打盹模式
- 通过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应用关键特性(如对于时序要求高的通信)的情况下有选择地调节应用的功耗。

### 10.1 节能模式概述

除了全功耗工作模式(或称运行模式)之外, PIC24FJ128GA310 系列器件还提供三种基于指令的节能模式和一种基于硬件的模式:

- 空闲
- 休眠(休眠和低压休眠)
- 深度休眠
- VBAT(使用和不使用 RTCC)

可通过关闭单片机的不同功能区来激活所有四种模式, 使运行和空闲时的功耗逐步降低。另外, 可对其中三种模式进行定制以降低更多功耗, 但要损失某些运行功能。[表 10-1](#) 以节能程度高低为顺序列出所有工作模式。

[表 10-2](#) 总结了单片机如何退出不同的模式。具体信息请见下文。

表 10-1: PIC24FJ128GA310 系列器件的工作模式

模式	进入	活动系统				
		内核	外设	数据 RAM 保持	RTCC <sup>(1)</sup>	DSGPR0/ DSGPR1 保持
运行(默认)	N/A	有	有	有	有	有
空闲	指令	无	有	有	有	有
休眠:						
休眠	指令	无	可选 <sup>(2)</sup>	有	有	有
低压休眠	指令 + RETEN 位	无	可选 <sup>(2)</sup>	有	有	有
深度休眠:						
深度休眠	指令 +DSEN 位	无	无	无	有	有
VBAT:						
使用 RTCC	硬件	无	无	无	有	有

**注 1:** 如果在固件中另外使能 RTCC。

**2:** 在此模式下, 选定外设可通过 LPRC 或外部时钟运行。

# PIC24FJ128GA310 系列

表 10-2：退出节能模式

模式	退出条件							代码执行恢复 <sup>(2)</sup>	
	中断		复位		RTCC 闹钟	WDT	VDD 恢复		
	全部	INT0	全部	POR					
空闲	有	有	有	有	有	有	N/A	下一条指令	
休眠（所有模式）	有	有	有	有	有	有	N/A		
深度休眠	无	有	无	有	有	有 <sup>(1)</sup>	N/A	复位向量	
VBAT	无	无	无	无	无	无	有	复位向量	

注 1：深度休眠 WDT。

2：代码执行恢复还对所有退出条件有效；例如 MCLR 和 POR 退出将导致从复位向量执行代码。

## 10.1.1 基于指令的节能模式

通过执行 PWRSAV 指令进入三种节能模式。休眠模式下时钟工作停止，所有代码暂停执行。空闲模式下 CPU 和代码暂停执行，但允许外设模块继续工作。深度休眠模式下时钟停止工作，代码执行停止，以及 RTCC 和 DSWDT 之外的所有外设都停止。它还冻结 I/O 状态并切断对闪存的供电，并有可能切断对 SRAM 的供电。

PWRSAV 指令的汇编语法如例 10-1 所示。通过一条汇编器命令直接进入休眠模式和空闲模式。深度休眠需要一个附加序列才能解锁并进入深度休眠模式，第 10.4.1 节“进入深度休眠模式”对此进行了介绍。

注：SLEEP\_MODE 和 IDLE\_MODE 是在所选器件的汇编器包含文件中定义的常数。  
要进入深度休眠模式，应该在将 DSEN 位置 1 之前将 DS<sub>CON<0></sub> 位清零。

允许的中断、WDT 超时或器件复位会导致器件从休眠和空闲模式退出。器件退出这两种模式的过程称为“唤醒”。

通过低压 / 保持稳压器使能的特性会导致休眠模式行为方式发生更改。请参见第 10.3 节“休眠模式”。

### 10.1.1.1 在节能指令执行期间的中断

在 PWRSAV 指令执行期间发生的中断都将延迟到进入休眠/深度休眠或空闲模式后才产生。器件然后将从休眠/深度休眠或空闲模式唤醒。

## 例 10-1：PWRSAV 指令语法

```
// Syntax to enter Sleep mode:  
PWRSAV      #SLEEP_MODE           ; Put the device into SLEEP mode  
  
//  
// Syntax to enter Idle mode:  
PWRSAV      #IDLE_MODE            ; Put the device into IDLE mode  
  
//  
// Syntax to enter Deep Sleep mode:  
// First use the unlock sequence to set the DSEN bit (见例 10-2)  
CLR    DSCON  
CLR    DSCON                  ; (repeat the command)  
BSET  DSCON, #DSEN             ; Enable Deep Sleep  
BSET  DSCON, #DSEN             ; Enable Deep Sleep (repeat the command)  
PWRSAV      #SLEEP_MODE           ; Put the device into Deep SLEEP mode
```

## 10.1.2 基于硬件的节能模式

基于硬件的VBAT模式在代码开发期间不需要用户进行任何操作。它是一项硬件设计功能，允许单片机（使用DSGPRn寄存器）保留重要数据以及在从应用中移除VDD时保持RTCC。通过为特定电源引脚提供备用电源来实现此功能。在第10.5节“[Vbat模式](#)”中详细介绍了VBAT模式。

## 10.1.3 低压/保持稳压器

PIC24FJ128GA310系列器件集成了另一个片内稳压器，旨在提供电源以选择标称值为1.2V的单片机功能。此稳压器允许原本处于非活动状态的功能（如数据RAM和WDT）保持在节能模式，或者保持在与其他情况相比功耗更低的状态。

低压/保持稳压器仅在调用休眠模式或深度休眠模式时可用。它由LPCFG配置位(CW1<10>)控制，在固件中由RETEN位(RCON<12>)控制。必须将LPCFG编程为0并将RETEN位置1，才能使能该稳压器。

## 10.2 空闲模式

空闲模式有以下这些特性：

- CPU将停止执行指令。
- WDT自动清零。
- 系统时钟源保持活动状态。默认情况下，所有外设模块将使用系统时钟源继续正常工作，但也可被有选择地禁止（见第10.8节“[选择性外设模块控制](#)”）。
- 若使能了WDT或FSCM，LPRC也将保持活动状态。

当发生以下任何事件时，器件将从空闲模式唤醒：

- 产生被单独允许的中断
- 任何器件复位
- WDT超时

从空闲模式唤醒时，为CPU重新提供时钟，且立即从PWRSAV指令之后的下一条指令或ISR中的第一条指令开始执行。

## 10.3 休眠模式

休眠模式包括以下特性：

- 系统时钟源关闭。如果使用了片上振荡器，它也将关闭。
- 如果没有I/O引脚拉电流，则器件电流消耗将降至最低。
- 冻结I/O引脚方向和状态。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果采用LPRC作为时钟源的WDT或RTCC被使能，LPRC时钟将继续在休眠模式下运行。
- 如果使能了WDT，它将在进入休眠模式之前自动清零。
- 某些器件功能或外设可在休眠模式下继续工作。这包括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。任何需要系统时钟源工作的外设都会在休眠模式下被禁止。

当发生以下任何事件时，器件将从休眠模式唤醒：

- 产生任何被单独允许的中断源
- 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的有效时钟源重新启动。

### 10.3.1 低压 / 保持休眠模式

低压 / 保持休眠模式的作用相当于带有相同功能和唤醒触发的休眠模式。区别在于低压 / 保持稳压器允许内核数字逻辑电压(VCORE)降至1.2V标称值。与VCORE保持在1.8V(最低)电压时相比，这可进一步降低功耗。

低压休眠模式需要的唤醒时间长于休眠模式，因为需要额外的时间将VCORE恢复到1.8V(称为TREG)。此外，使用低压 / 保持稳压器会限制可为任意有效外设(例如RTCC/LCD等)提供的电流值。

## 10.4 深度休眠模式

深度休眠模式提供基于指令的模式可以实现的最低程度的功耗。

深度休眠模式有以下特性：

- 系统时钟源关闭。如果使用了片上振荡器，它也将关闭。
- 器件电流消耗将降至最低水平。
- 冻结 I/O 引脚方向和状态。
- 由于禁止了系统时钟源，所以故障保护时钟监视器在休眠模式下不工作。
- 如果使能了采用 LPRC 作为时钟源的 WDT 或 RTCC，LPRC 时钟将继续在深度休眠模式下运行。
- 使用专用的深度休眠WDT和BOR系统（如果使能）。
- RTCC 及其时钟源继续运行（如果使能）。所有其他外设均被禁止。

深度休眠模式的进入是完全受软件控制的。发生以下任何事件时可使器件退出深度休眠模式：

- POR 事件
- MCLR 事件
- RTCC 闹钟（如果使用了 RTCC 的话）
- 外部中断 0
- 深度休眠看门狗定时器（DSWDT）超时

### 10.4.1 进入深度休眠模式

进入深度休眠模式的方法是：将 DSCON 寄存器中的 DSEN 位置 1，然后在一个指令周期内执行休眠命令（PWRSAV #SLEEP\_MODE）以减少误进入深度休眠模式的可能性。

如果 PWRSAV 命令不是在一个指令周期内给出的，DSEN 位将由硬件清零，且必须由软件再次置 1，然后才能进入深度休眠模式。退出深度休眠模式时，DSEN 位也将自动清零。

**注：**要在深度休眠唤醒之后再次进入深度休眠模式，允许在清零 RELEASE 位之后至少有一个 3 TCY 的延迟。

进入深度休眠模式的步骤是：

1. 如果应用需要深度休眠 WDT，可使能它并配置其时钟源。有关深度休眠 WDT 的详细信息，请参见第 10.4.5 节“深度休眠 WDT”。
2. 如果应用需要深度休眠 BOR，可通过编程 DSBOREN 配置位（CW4<6>）使能它。
3. 如果应用需要在发生 RTCC 闹钟事件时从深度休眠模式唤醒，可使能并配置 RTCC 模块。有关 RTCC 的详细信息，请参见第 22.0 节“实时时钟和日历（RTCC）”。
4. 如有需要，可通过把应用的任何重要的上下文数据写入 DSGPR0 和 DSGPR1 寄存器（可选）来保存这些数据。
5. 通过置 1 DSEN 位（DSCON<15>）使能深度休眠模式。

**注：**需要重复序列以将 DSEN 位置 1。需要重复序列（将指令重复两遍）以写入任一深度休眠寄存器（DSCON、DSWAKE、DSGPR0 和 DSGPR1）。此要求是为了避免用户错误地进入深度休眠模式。这些寄存器的任何写入操作都必须执行两次才能真正完成写入（见例 10-2）。

6. 通过发出 3 个 NOP 命令，然后发出 PWRSAV #0 指令进入深度休眠模式。

任何时候置 1 DSEN 位，DSWAKE 寄存器中的所有位都将自动清零。

### 例 10-2：重复序列

```
例 1:  
mov #8000, w2      ; enable DS  
mov w2, DSCON  
mov w2, DSCON      ; second write required to  
actually write to DSCON  
  
例 2:  
bset    DSCON, #15  
nop  
nop  
nop  
bset    DSCON, #15 ; enable DS (two writes required)
```

## 10.4.2 退出深度休眠模式

发生以下任一事件时退出深度休眠模式：

- VDD 电源上的 POR 事件。如果没有提供 DSBOR 电路重新激活 VDD 电源 POR 电路，那么必须将外部 VDD 电源降低到 POR 电路的自然激活电压。
- DSWDT 超时。DSWDT 定时器超时时，器件退出深度休眠模式。
- RTCC 闹钟（如果 RTCEN = 1）。
- MCLR 引脚有效 (0)。
- INTO 引脚有效（如果在进入深度休眠模式之前允许此中断的话）。极性配置用于确定导致器件退出深度休眠模式的引脚的有效电平 (0 或 1)。深度休眠模式期间，INTO 引脚上的电平变化会导致器件退出深度休眠模式。

**注：** 进入深度休眠模式时挂起的任何中断都将清除。

退出深度休眠模式通常不保持器件的状态，它等同于器件的上电复位 (POR)。RTCC (如果存在，在唤醒过程中始终保持工作状态)、DSGPRx 寄存器和 DSWDT 例外。

从退出深度休眠模式到 POR 序列完成过程中发生的唤醒事件不会被忽略。DSWAKE 寄存器将捕捉从 DSEN 置 1 到 RELEASE 清零的所有唤醒事件。

退出深度休眠模式的步骤是：

1. 发生唤醒事件后，器件退出深度休眠模式并执行 POR。DSEN 位自动清零。从复位向量处继续执行代码。
2. 要确定器件是否已退出深度休眠模式，请读深度休眠位 DPSLP (RCON<10>)。如果确定已退出深度休眠模式，该位将置 1。如果该位置 1，请将其清零。
3. 通过读 DSWAKE 寄存器确定唤醒源。
4. 通过读 DSBOR 位 (DSCON<1>) 确定深度休眠模式期间是否发生 DSBOR 事件。
5. 如果已保存应用的上下文数据，请将其从 DSGPR0 和 DSGPR1 寄存器读回。
6. 清零 RELEASE 位 (DSCON<0>)。

## 10.4.3 使用 DSGPRn 寄存器保存上下文数据

由于退出深度休眠模式会导致 POR，大部分特殊功能寄存器都将复位为其默认的 POR 值。而且，由于深度休眠模式下不提供 VCore 电源，所以退出此模式时有可能丢失数据 RAM 中的信息。

需要在进入深度休眠模式之前保存重要数据的应用可能使用深度休眠通用寄存器 (DSGPR0 和 DSGPR1) 或数据 EEPROM (如果可用的话)。这些寄存器与其他 SFR 不同，当器件处于深度休眠模式时其内容被保留下来。退出深度休眠模式后，可用软件通过读这些寄存器并清零 RELEASE 位 (DSCON<0>) 来恢复数据。

## 10.4.4 深度休眠模式下的 I/O 引脚

深度休眠模式期间，通用 I/O 引脚保留其之前的状态且辅助振荡器 (SOSC) 将继续运行 (如果使能的话)。在进入深度休眠模式之前配置为输入 (TRISx 位置 1) 的引脚在深度休眠模式期间保持高阻态。在进入深度休眠模式之前配置为输出 (TRISx 位清零) 的引脚在深度休眠模式期间保持为输出引脚。在此模式期间，它们继续驱动在进入深度休眠模式时其对应的 LATx 位所确定的输出电平。

一旦器件被唤醒，所有 I/O 引脚将继续保持其之前的状态，即使在器件完成了 POR 序列并再次执行应用程序代码后也是如此。配置为输入的引脚在深度休眠模式期间保持高阻态，配置为输出的引脚继续驱动其之前的电平值。唤醒后，TRIS 和 LAT 寄存器以及 SOSCEN 位 (OSCCON<1>) 都将复位。如果固件修改了这些位或寄存器中的任何一个，I/O 将不会立即进入新配置的状态。在固件清零 RELEASE 位 (DSCON<0>) 时，I/O 引脚将“被释放”，从而导致 I/O 引脚处于其对应的 TRIS 和 LAT 位的值所配置的状态。

这意味着唤醒后保持 SOSC 运行将需要在清零 RELEASE 之前将 SOSCEN 位置 1。

如果使能了深度休眠 BOR (DSBOR)，且深度休眠期间发生了 DSBOR 或真正的 POR 事件，那么 I/O 引脚将立即被释放，这与清零 RELEASE 位有些类似。所有之前的状态信息都将丢失，包括通用 DSGPR0 和 DSGPR1 的内容。

如果深度休眠期间发生了 MCLR 复位事件，那么 DSGPRx、DSCON 和 DSWAKE 寄存器将保持有效且 RELEASE 位将保持置 1 状态。SOSC 的状态也将保持。但是，I/O 引脚将复位为其 MCLR 复位状态。由于 RELEASE 仍然置 1，所以对 SOSCEN 位 (OSCCON<1>) 的更改直到 RELEASE 位清零之后才会生效。

在所有其他深度休眠唤醒情况下，应用固件必须清零 RELEASE 位才能重新配置 I/O 引脚。

## 10.4.5 深度休眠 WDT

要使能深度休眠模式下的 DSWDT，请编程配置位 DSWDTEN (CW4<7>)。无需为了 DSWDT 正常工作而使能器件的 WDT。进入深度休眠模式将自动复位 DSWDT。

通过 DSWDTONC 配置位 (CW4<4>) 选择 DSWDT 的时钟源。通过 DSWDPS<4:0> 配置位 (CW4<4:0>) 编程后分频器选项。可实现的最小超时时间为 1 ms，最大为 25.7 天。有关 CW4 配置寄存器和 DSWDT 配置选项的详细信息，请参见[第 29.0 节“特殊功能”](#)。

### 10.4.5.1 深度休眠模式下切换时钟

RTCC 和 DSWDT 可以从 SOSC 或 LPRC 时钟源提供的时钟运行。这使得 RTCC 和 DSWDT 可以在不需要同时使能 LPRC 和 SOSC 的情况下运行，从而降低功耗。

从 LPRC 提供的时钟运行 RTCC 将导致 RTCC 的精度损失大约 5% 至 10%。如果需要精度更高的 RTCC，必须从 SOSC 时钟源提供的时钟运行 RTCC。通过 RTCOSC 配置位 (RTCLK<1:0>) 选择 RTCC 的时钟源。

在某些环境下，可以在进入深度休眠模式时关闭 DSWDT 的时钟源。在这种情况下，无需使用软件，就可自动开启时钟源（如果使能了 DSWDT）。但是，这会导致 DSWDT 计数器启动时有一段延时。为了避免 SOSC 用作时钟源时的这段延时，应用可在进入深度休眠模式之前激活 SOSC。

## 10.4.6 检查并清零深度休眠的状态位

进入深度休眠模式时，状态位 DPSLP (RCON<10>) 置 1，且必须由软件清零。

上电时，软件应读取该状态位以确定复位是否由退出深度休眠模式引起，如果该位置 1 的话，则将其清零。在 DPSLP 和 POR 位状态的四种可能组合中，须注意以下三种情况：

- DPSLP 和 POR 位都清零。在这种情况下，复位是由其他事件引起，而不是退出深度休眠模式。
- DPSLP 位清零，但是 POR 位置 1；这是正常的 POR。
- DPSLP 和 POR 位都置 1。这表示进入深度休眠模式，器件掉电，然后退出了深度休眠模式。

## 10.4.7 上电复位 (POR)

监视 VDD 电压以产生 POR。由于退出深度休眠模式在功能上看起来与 POR 类似，但是应使用[第 10.4.6 节“检查并清零深度休眠的状态位”](#)中描述的技巧来区别退出深度休眠和真正的 POR 事件。发生真正的 POR 时，整个器件包括所有深度休眠逻辑（深度休眠寄存器、RTCC 和 DSWDT 等）在内都将复位。

## 10.5 VBAT 模式

此模式代表单片机可实现并仍可继续工作的最低功耗状态。单片机的主电源 VDD 发生故障时自动触发 VBAT 模式。此时，单片机的片内电源开关连接到为 VBAT 引脚供电的备用电源（例如电池）。在 VDD 恢复之前，这使一些关键系统保持在极低功耗水平。

VBAT 提供的电源只运行两个系统：RTCC 和深度休眠信号寄存器 (DSGPR0 和 DSGPR1)。为了在突然丢失 VDD 时保持这些系统，必须将 VDD 或 AVDD 之外的电源连接到 VBAT 引脚。

使能 RTCC 后，将继续使用在进入 VBAT 模式之前选择的同一时钟源（SOSC 或 LPRC）进行工作。在模式切换后，并不切换到更低功耗的时钟源。

由于丢失 VDD 通常是不可预见的事件，建议在代码执行的早期将要保留的数据装入深度休眠信号寄存器。

## 10.5.1 不使用 RTCC 的 VBAT 模式

在 VBAT 模式下禁止 RTCC，功耗降至所有节能模式中的最低水平。在此模式下，只保持深度休眠信号寄存器。

## 10.5.2 从 VBAT 模式唤醒

当恢复处于 VBAT 模式下的器件的 VDD 时，器件将自动唤醒。唤醒通过 POR 实现，之后器件开始从复位向量执行代码。除深度休眠信号寄存器和 RTCC 寄存器之外，所有 SFR 都复位到各自 POR 值。如果 RTCC 未配置为在 VBAT 模式下运行，它将保持禁止状态且 RTCC 不会运行。唤醒时序类似于正常 POR 的时序。

要区分从 VBAT 模式的唤醒和从其他 POR 状态的唤醒，请检查 VBAT 状态位（RCON2<0>）。如果在器件开始从复位向量执行代码时此位置 1，则表示已退出 VBAT 模式。应用必须清零 VBAT 位以确保捕获未来的 VBAT 唤醒事件。

如果在电源并未连接到 VBAT 引脚时发生 POR，则 VBPORE 位（RCON2<1>）置 1。如果在发生 POR 时此位置 1，则表示 VBAT 引脚需要连接电池。

此外，如果在 VBAT 模式下 VBAT 电源低于深度休眠信号所需的工作电平（例如电池耗尽），VBPORE 位将置 1。在单片机第一次上电时 VBPORE 也置 1，即使已为 VBAT 供电。

VBPORE 置 1 时用户应将其清零，之后此位只在 VDD = 0 且 VBAT 引脚电平过低时（小于 0.4V—0.6V）置 1。

## 10.5.3 VBAT 模式期间的 I/O 引脚

所有 I/O 引脚都应保持 VSS 电平； VBAT 模式期间不应为任何 I/O 引脚提供 VDD（请参见“[绝对最大额定值](#)”）。只有 SOSCI 和 SOSCO 引脚例外，如果使用辅助振荡器作为 RTCC 时钟源，这两个引脚保持各自状态。恢复 VDD 后，用户应使用 TRIS 位和 LAT 位将 I/O 引脚恢复至正确状态。

## 10.5.4 使用 DSGPRn 寄存器保存上下文数据

和深度休眠模式一样，恢复 VDD 后所有 SFR 都复位到各自 POR 值。只保留深度休眠信号寄存器。需要保存关键数据的应用应将这些数据保存在 DSGPRO 和 DSGPR1。

**注：** 如果未使用 VBAT 模式，建议将 VBAT 引脚连接到 VDD。

无论是否使用 VBAT 模式（连接电池），始终建议在 VBAT 引脚与接地端之间连接 0.1  $\mu$ F 电容。此电容的位置应十分接近于 VBAT 引脚。

应使能 BOR 以便 VBAT 可靠工作。

# PIC24FJ128GA310 系列

寄存器 10-1: DSCON: 深度休眠控制寄存器 <sup>(1)</sup>

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

U-0	U-0	U-0	U-0	U-0	r-0	R/W-0	R/C-0, HS
—	—	—	—	—	r	DSBOR <sup>(2)</sup>	RELEASE
bit 7	bit 0						

图注:

C = 可清零位

U = 未实现位, 读为 0

R = 可读位

W = 可写位

HS = 可由硬件置 1 的位

r = 保留位

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **DSEN:** 深度休眠使能位

1 = 执行 PWRSAV #0 时进入深度休眠模式  
0 = 执行 PWRSAV #0 时进入正常休眠模式

bit 14-3      未实现: 读为 0

bit 2      保留: 保持为 0

bit 1      **DSBOR:** 深度休眠 BOR 事件位 <sup>(2)</sup>

1 = 深度休眠模式期间, DSBOR 处于活动状态且检测到 BOR 事件  
0 = 深度休眠期间 DSBOR 无效, 或有效但未检测到 BOR 事件

bit 0      **RELEASE:** I/O 引脚状态释放位

1 = 从深度休眠模式唤醒时, I/O 引脚保持在进入深度休眠模式之前的状态  
0 = 将 I/O 引脚从进入深度休眠模式之前的状态释放, 并允许对应的 TRIS 和 LAT 位控制其状态

注 1: 仅在 POR 事件不在深度休眠模式期间发生时复位所有寄存器位。

2: 与其他所有事件不同, 深度休眠 BOR 事件不会导致从深度休眠模式唤醒; 这会重新激活 POR。

## 寄存器 10-2: DSWAKE: 深度休眠唤醒源寄存器<sup>(1)</sup>

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0, HS
—	—	—	—	—	—	—	DSINT0
bit 15							bit 8

R/W-0, HS	U-0	U-0	R/W-0, HS	R/W-0, HS	R/W-0, HS	U-0	U-0
DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	—
bit 7							bit 0

图注:

HS = 可由硬件置 1 的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-9 未实现: 读为 0
- bit 8 **DSINT0:** 电平变化中断位  
1 = 深度休眠期间, 电平变化中断有效  
0 = 深度休眠期间, 电平变化中断无效
- bit 7 **DSFLT:** 深度休眠故障检测位  
1 = 深度休眠期间发生故障且一些深度休眠配置设置可能受损  
0 = 深度休眠期间未检测到任何故障
- bit 6-5 未实现: 读为 0
- bit 4 **DSWDT:** 深度休眠看门狗定时器超时位  
1 = 深度休眠期间, 深度休眠看门狗定时器已超时  
0 = 深度休眠期间, 深度休眠看门狗定时器未超时
- bit 3 **DSRTCC:** 实时时钟和日历闹钟位  
1 = 深度休眠期间实时时钟和日历已触发闹钟  
0 = 深度休眠期间实时时钟和日历未触发闹钟
- bit 2 **DSMCLR:** MCLR 事件位  
1 = 深度休眠期间, MCLR 引脚处于活动状态且有效  
0 = 深度休眠期间, MCLR 引脚处于非活动状态, 或处于活动状态但无效
- bit 1-0 未实现: 读为 0

注 1: DSEN (DSCON<15>) 位置 1 时, 所有寄存器位都清零。

# PIC24FJ128GA310 系列

## 寄存器 10-3: RCON2: 复位和系统控制寄存器 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-0	R/CO-1	R/CO-1	R/CO-1	R/CO-0
—	—	—	r	VDDBOR <sup>(1)</sup>	VDDPOR <sup>(1,2)</sup>	VBPOR <sup>(1,3)</sup>	VBAT <sup>(1)</sup>
bit 7							bit 0

图注:

CO = 只可清零位

r = 保留位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-5      未实现: 读为 0
- bit 4      保留: 保持为 0
- bit 3      **VDDBOR:** V<sub>DD</sub> 欠压复位标志位<sup>(1)</sup>  
1 = 发生了 V<sub>DD</sub> 欠压复位 (由硬件置 1)  
0 = 未发生 V<sub>DD</sub> 欠压复位
- bit 2      **VDDPOR:** V<sub>DD</sub> 上电复位标志位<sup>(1,2)</sup>  
1 = 发生了 V<sub>DD</sub> 上电复位 (由硬件置 1)  
0 = 未发生 V<sub>DD</sub> 上电复位
- bit 1      **VBPOR:** V<sub>BAT</sub> 标志位<sup>(1,3)</sup>  
1 = 发生了 V<sub>BAT</sub> 上电复位 (V<sub>BAT</sub> 引脚未连接电池, 或 V<sub>BAT</sub> 电平低于深度休眠信号保持电平, 由硬件置 1)  
0 = 未发生 V<sub>BAT</sub> 上电复位
- bit 0      **VBAT:** V<sub>BAT</sub> 标志位<sup>(1)</sup>  
1 = 向 V<sub>BAT</sub> 引脚供电时发生 POR 退出 (由硬件置 1)  
0 = V<sub>BAT</sub> 未发生 POR 退出

注 1: 此位仅可由硬件置 1; 仅可由软件清零。

2: 表示 V<sub>DD</sub> POR。POR 位 (RCON<0>) 置 1 表示 VCORE POR。

3: 器件最初上电时此位置 1, 即使 V<sub>BAT</sub> 已通电。建议用户将此标志清零, 之后此位仅在 V<sub>BAT</sub> 电压低于 0.4-0.6V 时且 V<sub>DD</sub> = 0 时置 1。

## 10.6 时钟频率和时钟切换

在运行和空闲模式下，所有 PIC24FJ 器件都允许在应用程序控制下，从较大的时钟频率范围内进行选择。如果未锁定系统时钟配置，用户只需更改 NOSC 位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，将在第 9.0 节“振荡器配置”中进行更详细的讨论。

## 10.7 打盹模式

通常，更改时钟速度和进入某种节能模式是降低功耗的首选策略。然而，有些情况下不可行。例如，某些应用可能必须保持不间断的同步通信，即使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导致通信错误，而使用节能模式就可能完全停止通信。

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，继续以相同的时钟源和相同的速度驱动系统时钟。外设模块时钟速度保持不变，但 CPU 时钟的速度降低了。保持两个时钟域同步，以允许外设在 CPU 以较慢的速率执行代码时访问 SFR。

通过置 1 DOZEN 位 (CLKDIV<11>) 使能打盹模式。外设和内核时钟速度之比由 DOZE<2:0> 位 (CLKDIV<14:12>) 决定。有 8 种可能的配置，从 1:1 到 1:128，其中 1:1 是默认设置。

还可使用打盹模式在事件驱动应用中有选择地降低功耗。这样就可以实现不间断地运行对时钟要求高的功能（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操作没有影响。

## 10.8 选择性外设模块控制

空闲和打盹模式允许用户通过降低 CPU 时钟速度或停止 CPU 时钟大幅降低功耗。然而，外设模块的时钟仍然保持运行，因此会有功耗产生。可能在有些情况下应用需要这些模式无法提供的功能，比如将绝大部分能源分配给 CPU 处理工作，而只为外设提供最低的功耗。

PIC24F 器件允许有选择地禁止外设模块，从而降低或消除它们的功耗，以此满足上述需求。这可以通过两个控制位来实现：

- 外设使能位，通常称为 “XXXEN”，位于模块主控制 SFR 中。
- 外设模块禁止 (PMD) 位，通常称为 “XXXMD”，位于某个 PMD 控制寄存器中 (XXXMD 位于 PMD1、PMD2、PMD3、PMD4、PMD6 和 PMD7 寄存器)。

这两位在使能或禁止相关模块时具有相似的功能。将模块的 PMD 位置 1 会禁止该模块的所有时钟源，从而将其功耗降至绝对最低。在此状态下，与此外设相关的控制和状态寄存器也会被禁止，所以无法写这些寄存器且读取值无效。许多外设模块都有一个对应的 PMD 位。

而通过清零某个模块的 XXXEN 位将会禁止其功能，但是仍然允许对其寄存器进行读写操作。功耗会降低，但是没有使用 PMD 位所降低的程度高。大多数外设模块都有一个使能位，但捕捉、比较和 RTCC 模块除外。

要节省更多的功耗，也可在器件进入空闲模式时有选择地禁止外设模块。使用通用名格式为“XXXIDL”的控制位可以执行此操作。默认情况下，可以在空闲模式下工作的所有模块都可以执行此操作。使用“在空闲模式下禁止”功能可在空闲模式下禁止模块，以便进一步降低空闲模式下的功耗，在功耗尤其重要的应用中，更大程度地节省功耗。

# **PIC24FJ128GA310 系列**

---

---

注:

## 11.0 I/O 端口

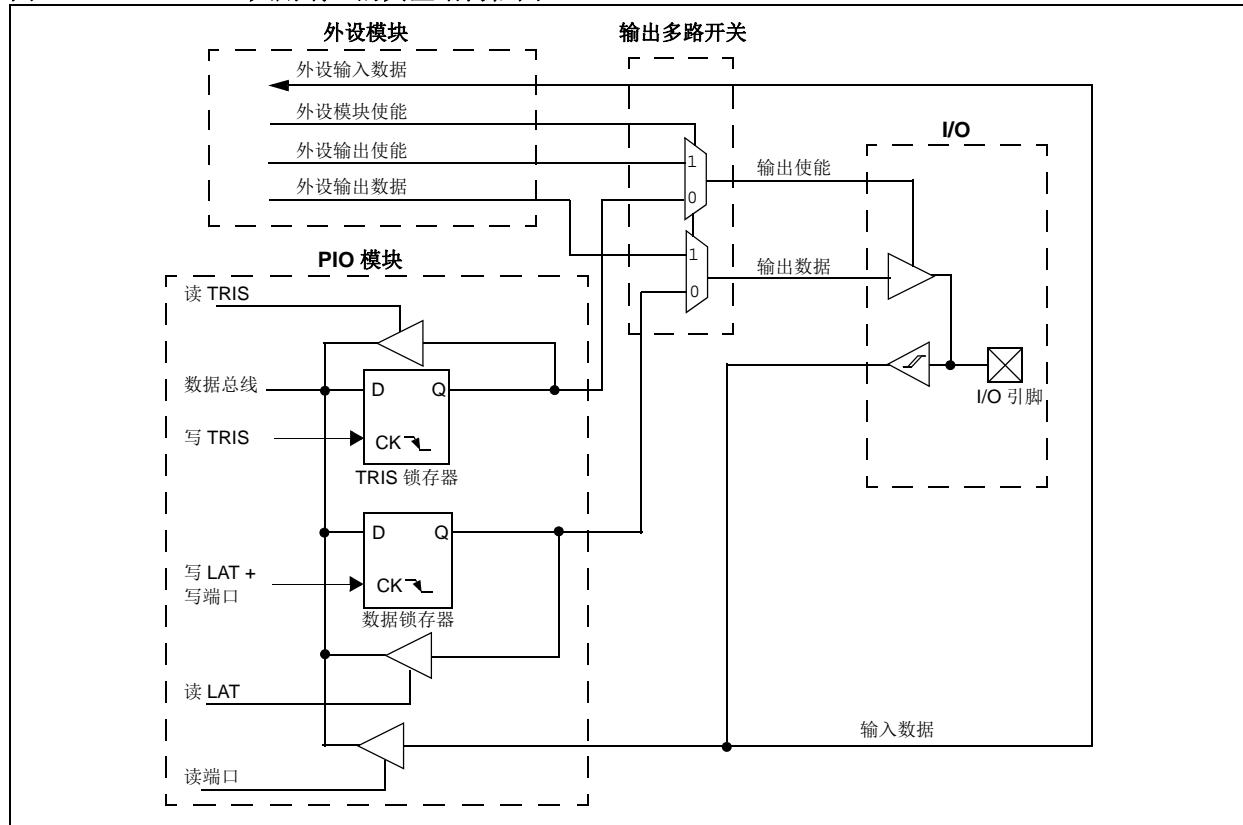
**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 12 章“带外设引脚选择 (PPS) 的 I/O 端口”(DS39711B\_CN)。本数据手册中的信息取代了 FRM 中的信息。

所有器件引脚（除了 VDD、VSS、MCLR 和 OSC1/CLK1 以外）均由外设和并行 I/O 端口共用。所有 I/O 输入端口都为施密特触发器 (ST) 输入，以便增强抗干扰性。

### 11.1 并行 I/O (PIO) 端口

通常, 与某个外设共用一个引脚的并行 I/O 端口总是服务于该外设。外设的输出缓冲器数据和控制信号提供给一对多路开关。这对多路开关用于选择 I/O 引脚的输出数据和控制信号是用于外设还是相应的端口。该逻辑同时还会阻止“环回进入 (loop through)”, 即一个端口的数字输出可以驱动共用同一个引脚的外设输入。图 11-1 中显示端口是如何与其他外设复用的, 及其连接的关联 I/O 引脚。

图 11-1: 共用端口的典型结构框图



当使能某外设并驱动与其相对应的引脚时, 将禁止此引脚的通用输出功能。可以读该 I/O 引脚, 但并行端口位的输出驱动器将被禁止。若使能某外设但没有驱动引脚时, 则该引脚可由一个端口驱动。

所有端口引脚都有三个与端口引脚作为数字 I/O 时的工作直接相关的寄存器, 和一个与端口引脚作为模拟输入时的工作相关的寄存器。数据方向寄存器 (TRISx) 决定引脚是输入引脚还是输出引脚。如果数据方向位为 1, 则为输入引脚。复位以后, 所有端口引脚被定义为输入引脚。读输出锁存寄存器 (LATx) 时, 读到的是锁存器中的值; 写锁存器时, 写入的是锁存器。但读取端口 (PORTx) 时, 读到的是端口引脚的值; 而写入端口引脚时, 写入的是相应的锁存器。

对于某个特定器件, 无效的位及其相关联的数据和控制寄存器都将被禁止。这意味着相应的 LATx 和 TRISx 寄存器以及该端口引脚将读为 0。

当一个定义为只用作输入的引脚与另一个外设或功能复用时, 由于没有其他竞争的输入源, 它将被视为专用端口。RC13 和 RC14 只能是输入端口; 不能配置为输出。

## 11.1.1 I/O 端口写 / 读时序

在改变端口方向或对端口执行写操作，与对同一端口执行读操作之间需要间隔一个指令周期。通常在两者之间插一条 NOP 指令。

## 11.1.2 漏极开路配置

除端口、LAT 和 TRIS 寄存器用于数据控制外，每个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对应的漏极开路控制寄存器 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。

这种开漏特性允许通过使用外部上拉电阻在任何只能用作数字功能的引脚上产生高于 VDD（如 5V）的输出电平。允许的最大开漏电压与最大 VIH 规范相同。

## 11.2 配置模拟端口引脚 (ANSx)

ANSx 和 TRISx 寄存器用于控制具有模拟功能的引脚的操作。每个具有模拟功能的端口引脚都与某个 ANS 位相关（见 [寄存器 11-1 至寄存器 11-6](#)），该位决定了引脚功能是模拟的还是数字的。有关不同 ANSx 和 TRISx 位设置时引脚的详细操作，请参见 [表 11-1](#)。

当读取端口寄存器时，所有配置为模拟输入通道的引脚均读为零（低电平）。

### 11.2.1 模拟输入引脚和电压注意事项

用作器件输入的引脚可承受的电压与该引脚的输入功能有关。大多数输入引脚能够承受最高 5.5V 的直流电压，这是数字逻辑电路的典型电平值。但是，几个引脚只能承受最高为 VDD 的电压。应始终避免使这些引脚上的电压超过 VDD。

[表 11-2](#) 总结了不同的电压容差。更多详细信息，请参见 [第 32.0 节“电气特性”](#)。

**表 11-1：配置 I/O 引脚的模拟/数字功能**

引脚功能	ANSx 设置	TRISx 设置	备注
模拟输入	1	1	建议保持 ANSx = 1。
模拟输出	1	0	建议保持 ANSx = 1。
数字输入	0	1	固件必须在将引脚配置为数字输入后等待至少一个指令周期，然后才能读取有效输入值。
数字输出	0	0	请确保禁止该引脚上的模拟输出功能（如果具备的话）。

**表 11-2：端口或引脚可承受的输入电压说明**

端口或引脚	可承受的输入电压	说明
PORATA<15:14, 7:0> <sup>(1)</sup>	5.5V	可承受高于 VDD 的输入电平；这对于大部分标准逻辑有用。
PORTB<15:7, 5:2>		
PORTC<3:1> <sup>(1)</sup>		
PORTD<15:8, 5:0> <sup>(1)</sup>		
PORTE<9:8, 4:0> <sup>(1)</sup>		
PORTF<13:12, 8:0> <sup>(1)</sup>		
PORTG<15:12, 9, 6:0> <sup>(1)</sup>		
PORATA<10:9> <sup>(1)</sup>	VDD	只能承受 VDD 输入电平。
PORTB<6, 1:0>		
PORTC<15:12, 4> <sup>(1)</sup>		
PORTD<7:6>		
PORTE<7:5> <sup>(1)</sup>		
PORTG<8:7>		

**注 1：**并非所有这些引脚都在 64 引脚或 80 引脚器件上实现。有关端口引脚实现的完整说明，请参见 [第 1.0 节“器件概述”](#)。

## 寄存器 11-1: ANSA: PORTA 模拟功能选择寄存器

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

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSA7 <sup>(1)</sup>	ANSA6 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-6 **ANSA<7:6>**: 模拟功能选择位 <sup>(1)</sup>

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 5-0 未实现: 读为 0

**注 1:** 64 引脚和 80 引脚器件中没有这些位。

## 寄存器 11-2: ANSB: PORTB 模拟功能选择寄存器

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
ANSB15	ANSB14	ANSB13	ANSB12	ANSB11	ANSB10	ANSB9	ANSB8
bit 15							bit 8

| R/W-1 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ANSB7 | ANSB6 | ANSB5 | ANSB4 | ANSB3 | ANSB2 | ANSB1 | ANSB0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 **ANSB<15:0>**: 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口

0 = 引脚配置为数字模式; 使能读 I/O 端口

# PIC24FJ128GA310 系列

## 寄存器 11-3: ANSC: PORTC 模拟功能选择寄存器

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-1	U-0	U-0	U-0	U-0
—	—	—	ANSC4 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-5 未实现: 读为 0

bit 4 ANSC4: 模拟功能选择位<sup>(1)</sup>

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 3-0 未实现: 读为 0

注 1: 64 引脚和 80 引脚器件中没有该位。

## 寄存器 11-4: ANSD: PORTD 模拟功能选择寄存器

U-0	U-0	U-0	U-0	R/W-1	R/W-1	U-0	U-0
—	—	—	—	ANSD11	ANSD10	—	—
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSD7	ANSD6	—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-12 未实现: 读为 0

bit 11 ANSD<11:10>: 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 9-8 未实现: 读为 0

bit 7-6 ANSD<7:6>: 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 5-0 未实现: 读为 0

## 寄存器 11-5: ANSE: PORTE 模拟功能选择寄存器<sup>(1)</sup>

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	U-0
—	—	—	—	—	—	ANSE9 <sup>(2)</sup>	—
bit 15							bit 8

R/W-1	R/W-1	R/W-1	R/W-1	U-0	U-0	U-0	U-0
ANSE7	ANSE6	ANSE5	ANSE4	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9 **ANSE9:** 模拟功能选择位<sup>(2)</sup>

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 8 未实现: 读为 0

bit 7-4 **ANSE<7:4>:** 模拟功能选择位<sup>(1)</sup>

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 3-0 未实现: 读为 0

注 1: 64 引脚和 80 引脚器件中没有该寄存器。

2: 64 引脚器件未实现该位。在 80 引脚器件上, 此位需要清零才能使 RE9 具有数字功能。

## 寄存器 11-6: ANSG: PORTG 模拟功能选择寄存器

U-0	U-0	U-0	U-0	U-0	U-0	R/W-1	R/W-1
—	—	—	—	—	—	ANSG9	ANSG8
bit 15							bit 8

R/W-1	R/W-1	U-0	U-0	U-0	U-0	U-0	U-0
ANSG7	ANSG6	—	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9-6 **ANSG<9:6>:** 模拟功能选择位

1 = 引脚配置为模拟模式; 禁止读 I/O 端口  
0 = 引脚配置为数字模式; 使能读 I/O 端口

bit 5-0 未实现: 读为 0

## 11.3 输入电平变化通知

I/O 端口的输入电平变化通知功能允许 PIC24FJ128GA310 系列器件在选定输入引脚的电平状态变化 (Change-Of-State, COS) 时向处理器发出中断请求。即便是在时钟被禁止的休眠模式下，该特性也可检测到输入电平状态变化。根据器件的引脚数，最多可以选择（允许）82 个外部输入在电平状态发生变化时产生中断请求。

CNEN1 至 CNEN6 寄存器包含每个 CN 输入引脚的中断允许控制位。将其中任一位置 1 将允许相应引脚的 CN 中断。

每个 CN 引脚都有一个与之相连的弱上拉和弱下拉电路。弱上拉电路提供连接到该引脚的拉电流，而弱下拉电路提供连接到该引脚的灌电流。这样的话，当连接按钮或键盘设备时，不再需要使用外部电阻。使用 CNPU1 至 CNPU6 寄存器可单独使能上拉电路，使用 CNPD1 至 CNPD6 寄存器可单独使能下拉电路。每个 CN 引脚都为其上拉电路和下拉电路提供了独立的控制位。将某个控制位置 1 可使能其对应引脚的弱上拉或弱下拉电路。

选择内部上拉时，引脚将拉至 VDD - 1.1V（典型值）。当选择内部下拉电路时，引脚电平下拉至 VSS。

**注：**只要端口引脚被配置为数字输出引脚，电平变化通知引脚上的弱上拉电路将始终被禁止。

### 例 11-1：端口写/读（汇编语言）

```
MOV 0xFF00, W0      ; Configure PORTB<15:8> as inputs
MOV W0, TRISB       ; and PORTB<7:0> as outputs
NOP                 ; Delay 1 cycle
BTSS PORTB, #13     ; Next Instruction
```

### 例 11-2：端口写/读（C 语言）

```
TRISB = 0xFF00;          // Configure PORTB<15:8> as inputs and PORTB<7:0> as outputs
Nop();                  // Delay 1 cycle
If (PORTBbits.RB13){ }; // Next Instruction
```

## 11.4 外设引脚选择 (PPS)

通用器件的最大挑战是在最小化 I/O 引脚上功能冲突的同时，提供最大可能的外设功能集。在需要使用复用一个引脚的多个外设的应用中，对应用代码进行繁琐的更改或彻底重新设计可能是惟一的选择。

外设引脚选择 (PPS) 功能通过使能用户外设集选择并将外设功能放置到大量的 I/O 引脚中提供了一个替代这两种选择的方案。通过增加特定器件上的引脚配置选项，用户可以更好地调节单片机以满足整个应用的需要，而不是调整应用来满足器件。

外设引脚选择功能通过固定数量的数字 I/O 引脚进行操作。用户可将任一数字外设的输入和 / 或输出独立映射到这些 I/O 引脚之一。外设引脚选择通过软件执行，通常不需要器件重新编程。器件带有保护硬件，可防止在建立了外设映射后意外或误更改此映射。

### 11.4.1 可用引脚

外设引脚选择功能可在最多 44 个引脚的范围内使用；可用的引脚数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其完整引脚名称中有“R<sup>n</sup>P<sup>n</sup>”或“R<sup>n</sup>P<sup>n</sup>I<sup>n</sup>”标识，其中 “n” 指的是可重映射的引脚的编号。“RP” 用于指定支持可重映射输入和输出功能的引脚，而 “RPI” 指定仅支持可重映射输入功能的引脚。

PIC24FJ128GA310 系列器件支持的可重映射仅输入引脚数远大于可重映射输入 / 输出引脚数。在此器件系列中，最多有 32 个可重映射输入 / 输出引脚，具体取决于所选特定器件的引脚数。这些引脚编号为 RP0 至 RP31。可重映射仅输入引脚的编号在该范围之上，为 RPI32 至 RPI43（或特定器件的最高限制）。

请参见表 1-4，了解各封装提供的引脚排列选项汇总。

### 11.4.2 可用外设

外设引脚选择管理的外设都是仅数字外设，包括通用串行通信 (UART 和 SPI)、通用定时器时钟输入、定时器相关外设 (输入捕捉和输出比较) 以及外部中断输入。由于比较器模块的输出为离散数字信号，因此也包括在内。

PPS 不适用于以下外设：

- I<sup>2</sup>C<sup>TM</sup> (输入和输出)
- 变化通知输入
- RTCC 闹钟输出
- EPMP 信号 (输入和输出)
- LCD 信号
- 模拟输入
- INT0

有引脚选择和无引脚选择的外设的最大区别是有引脚选择的外设与默认 I/O 引脚之间无关联。必须在使用外设前始终将其分配给指定的 I/O 引脚。相反，假设无引脚选择的外设处于激活状态且未与其他外设发生冲突，则此外设可始终通过默认引脚使用。

#### 11.4.2.1 外设引脚选择功能优先级

可选引脚的外设输出（例如 OC 和 UART 发送）的优先级高于该引脚上的所有通用数字功能（例如 EPMP 和端口 I/O）。特殊的数字输出（例如使能 USB 的器件上的 USB）的优先级高于同一引脚上的 PPS 输出。本数据手册的引脚框图按照优先级顺序列出了外设输出。请参见这些引脚框图以了解关于某个特定引脚的优先级信息。

具有固定外设的 PIC24F 器件不同，可选引脚的外设输入从不具有该引脚的所有权。引脚的输出缓冲器由该引脚的 TRIS<sup>x</sup> 位设置或固定外设控制。如果引脚配置为数字模式，则 PPS 输入正常工作。如果使能了同一引脚上的模拟功能，则禁止 PPS 输入。

#### 11.4.3 控制外设引脚选择

PPS 功能是通过以下两组特殊功能寄存器 (SFR) 控制的：一组用于映射外设输入，另一组用于映射输出。由于输入和输出是单独控制的，因此特定外设的输入和输出（若该外设都有）均可施加到任何可选的功能引脚上，而没有限制。

根据映射的是输入还是输出，有两种方法可处理外设与外设可选引脚之间的关联。

# PIC24FJ128GA310 系列

## 11.4.3.1 输入映射

外设引脚选择选项的输入根据外设进行映射；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（见[寄存器 11-7 至寄存器 11-26](#)）。

各寄存器均包含两组 6 位位域，每组都与一个可选引脚的外设相关。给指定外设的位域赋上正确的 6 位值，会将具有此值的 RPn/RPin 引脚映射到该外设。对于任何给定器件，任何位域值的有效范围对应于此器件所支持的外设引脚选择的最大值。

表 11-3： 可选输入源（将输入映射到功能）<sup>(1)</sup>

输入名称	功能名称	寄存器	功能映射位
DSM 调制输入	MDMIN	RPINR30	MDMIR<5:0>
DSM 载波输入 1	MDCIN1	RPINR31	MDC1R<5:0>
DSM 载波输入 2	MDCIN2	RPINR31	MDC2R<5:0>
外部中断 1	INT1	RPINR0	INT1R<5:0>
外部中断 2	INT2	RPINR1	INT2R<5:0>
外部中断 3	INT3	RPINR1	INT3R<5:0>
外部中断 4	INT4	RPINR2	INT4R<5:0>
输入捕捉 1	IC1	RPINR7	IC1R<5:0>
输入捕捉 2	IC2	RPINR7	IC2R<5:0>
输入捕捉 3	IC3	RPINR8	IC3R<5:0>
输入捕捉 4	IC4	RPINR8	IC4R<5:0>
输入捕捉 5	IC5	RPINR9	IC5R<5:0>
输入捕捉 6	IC6	RPINR9	IC6R<5:0>
输入捕捉 7	IC7	RPINR10	IC7R<5:0>
输出比较故障 A	OCFA	RPINR11	OCFAR<5:0>
输出比较故障 B	OCFB	RPINR11	OCFBR<5:0>
SPI1 时钟输入	SCK1IN	RPINR20	SCK1R<5:0>
SPI1 数据输入	SDI1	RPINR20	SDI1R<5:0>
SPI1 从选择输入	SS1IN	RPINR21	SS1R<5:0>
SPI2 时钟输入	SCK2IN	RPINR22	SCK2R<5:0>
SPI2 数据输入	SDI2	RPINR22	SDI2R<5:0>
SPI2 从选择输入	SS2IN	RPINR23	SS2R<5:0>
Timer1 外部时钟	T1CK	RPINR23	T1CKR<5:0>
Timer2 外部时钟	T2CK	RPINR3	T2CKR<5:0>
Timer3 外部时钟	T3CK	RPINR3	T3CKR<5:0>
Timer4 外部时钟	T4CK	RPINR4	T4CKR<5:0>
Timer5 外部时钟	T5CK	RPINR4	T5CKR<5:0>
UART1 允许发送	U1CTS	RPINR18	U1CTSR<5:0>
UART1 接收	U1RX	RPINR18	U1RXR<5:0>
UART2 允许发送	U2CTS	RPINR19	U2CTSR<5:0>
UART2 接收	U2RX	RPINR19	U2RXR<5:0>
UART3 允许发送	U3CTS	RPINR21	U3CTSR<5:0>
UART3 接收	U3RX	RPINR17	U3RXR<5:0>
UART4 允许发送	U4CTS	RPINR27	U4CTSR<5:0>
UART4 接收	U4RX	RPINR27	U4RXR<5:0>

注 1：除非另外声明，否则所有输入均使用施密特触发器（ST）输入缓冲器。

### 11.4.3.2 输出映射

与输入相反，外设引脚选择选项的输出根据引脚进行映射。这种情况下，与特定引脚相关的控制寄存器指示要映射的外设输出。**RPORx** 寄存器用于控制输出映射。每个寄存器均包含两组 6 位位域；各位域都与一个 **RPn** 引脚相关（见[寄存器 11-27 至寄存器 11-42](#)）。位域值对应一个外设，该外设的输出映射到引脚（见[表 11-4](#)）。

由于采用的映射技术，输出映射的外设列表中还包含一个空值 000000。此值允许任何给定引脚始终与所有可选引脚的外设的输出断开。

**表 11-4：** 可选输出源（将功能映射到输出）

输出功能编号 <sup>(1)</sup>	功能	输出名称
0	NULL <sup>(2)</sup>	空
1	C1OUT	比较器 1 的输出
2	C2OUT	比较器 2 的输出
3	U1TX	UART1 发送
4	U1RTS <sup>(3)</sup>	UART1 请求发送
5	U2TX	UART2 发送
6	U2RTS <sup>(3)</sup>	UART2 请求发送
7	SDO1	SPI1 数据输出
8	SCK1OUT	SPI1 时钟输出
9	SS1OUT	SPI1 从选择输出
10	SDO2	SPI2 数据输出
11	SCK2OUT	SPI2 时钟输出
12	SS2OUT	SPI2 从选择输出
18	OC1	输出比较 1
19	OC2	输出比较 2
20	OC3	输出比较 3
21	OC4	输出比较 4
22	OC5	输出比较 5
23	OC6	输出比较 6
24	OC7	输出比较 7
28	U3TX	UART3 发送
29	U3RTS <sup>(3)</sup>	UART3 请求发送
30	U4TX	UART4 发送
31	U4RTS <sup>(3)</sup>	UART4 请求发送
36	C3OUT	比较器 3 的输出
37	MDOUT	DSM 调制器输出
38-63	(未使用)	未连接

**注 1：** 使用列出的值设置 **RPORx** 寄存器会将输出功能指定到相应的 **RPn** 引脚。

**2:** 在器件复位时将 NULL 功能分配给所有的 **RPn** 输出，并禁止 **RPn** 输出功能。

**3:** IrDA® BCLK 功能使用此输出。

### 11.4.3.3 映射限制

外设引脚选择的控制模式相当灵活。除阻止由两个配置为相同功能输入的物理引脚或配置给同一引脚的两个功能输出引起的信号冲突的系统电路外，无其他依靠硬件的输出锁定电路。此灵活性可扩展到允许一个输入驱动多个外设或一个功能输出驱动多个输出引脚。

### 11.4.3.4 PIC24FJ128GA310 系列器件的映射例外

尽管 PPS 寄存器理论上可具有最多 64 个可重映射 I/O 引脚，但并非在所有器件中都全部实现。对于 PIC24FJ128GA310 系列器件，可用的可重映射引脚的最大数量为 44，其中包含 12 个仅输入引脚。另外，RP 和 RPI 序列中的一些引脚在低引脚数器件上未实现。表 11-5 总结了可用的可重映射引脚的差异。

当开发使用可重映射引脚的应用时，用户还应记住以下事项：

- 对于 RPINRx 寄存器，对应于特定器件的未实现引脚的位组合被视为无效；对应的模块将不具有映射到它的输入。对于所有 PIC24FJ128GA310 系列器件，位组合包含所有大于 43 (101011) 的值。
- 对于 RPORx 寄存器，对应未实现引脚的位域也未实现。写入这些位域不会有影响。

### 11.4.4 控制配置更改

由于可在运行时更改外设重映射，因此需要对外设重映射设置一些限制条件以防止意外更改配置。PIC24F 器件有以下三种用于阻止更改外设映射的功能：

- 控制寄存器锁定序列
- 连续状态监视
- 配置位重映射锁定

### 11.4.4.1 控制寄存器锁定

正常工作状态下，不允许写 RPINRx 和 RPORx 寄存器。尝试的写操作看似正常执行，但寄存器的内容并没有发生变化。要更改这些寄存器的内容，寄存器必须用硬件解锁。寄存器锁定由 IOLOCK 位 (OSCCON<6>) 控制。将 IOLOCK 置 1 将阻止写入控制寄存器；而将 IOLOCK 清零将允许写入。

要置 1 或清零 IOLOCK，必须执行以下指定命令序列：

- 将 46h 写入 OSCCON<7:0>。
- 将 57h 写入 OSCCON<7:0>。
- 通过一次操作清零（或置 1）IOLOCK。

与振荡器 LOCK 位的类似序列不同，IOLOCK 在更改前一直保持一种状态。这允许使用一个解锁序列对所有外设引脚选择进行配置，然后对所有控制寄存器进行更新，最后用第二个锁定序列锁定。

### 11.4.4.2 连续状态监视

除了阻止直接写入外，RPINRx 和 RPORx 寄存器的内容还由影子寄存器通过硬件不停地进行监视。如果任何寄存器发生了不希望的更改（例如，由 ESD 或其他外部事件导致单元值发生误更改），则将触发配置失配复位。

### 11.4.4.3 配置位引脚选择锁定

作为又一层保护，可配置器件以阻止对 RPINRx 和 RPORx 寄存器执行多次写会话。IOL1WAY (CW2<4>) 配置位会阻止 IOLOCK 位在置 1 后再被清零。若 IOLOCK 保持置 1 状态，寄存器解锁过程将不会执行，且不能写入外设引脚选择控制寄存器。清零该位并重新使能外设重映射的唯一方法是执行器件复位。

默认（未编程）状态下，IOL1WAY 置 1，限制用户只能进行一次写会话。编程 IOL1WAY 允许用户对外设引脚选择寄存器进行不受限制的访问（通过正确地使用解锁序列）。

表 11-5：PIC24FJ128GA310 系列器件的可重映射引脚的例外情况

器件	RP 引脚 (I/O)		RPI 引脚	
	总数	未实现	总数	未实现
PIC24FJXXXGA306	29	RP5、RP15 和 RP31	1	RPI32-36 和 RPI38-43
PIC24FJXXXGA308	31	RP15	9	RPI32、RPI39 和 RPI41
PIC24FJXXXGA310	32	—	12	—

## 11.4.5 外设引脚选择注意事项

控制外设引脚选择的功能需要注意应用设计中几个容易被忽视的事项。对于仅可用作可重映射外设的几个常用外设尤其如此。

主要问题是在器件默认（复位）状态下外设引脚选择功能在默认引脚上不可用。由于所有 RPINRx 寄存器复位为 111111 且所有 RPORx 寄存器复位为 000000，因此所有外设引脚选择输入与 VSS 相连，而所有外设引脚选择输出断开连接。

**注：** 尝试将外设引脚选择输入到 RP63 时，器件上不必具有 RP63，这是因为寄存器复位会实现这一配置。

此情景需要用户在执行任何其他应用代码前使用正确的外设配置初始化器件。由于 IOLOCK 位复位为解锁状态，所以不必在器件退出复位后执行解锁序列。但是，考虑到应用的安全性，最好在写入控制寄存器之后将 IOLOCK 位置 1 并锁定配置。

由于解锁序列对时序有严格要求，因此必须用汇编语言编程，执行方式与更改振荡器配置的方式相同。若应用程序是用 C 或其他高级语言编写的，则仍应通过编写内嵌的汇编语言执行解锁序列。

选择配置需要查看所有外设引脚选择及其引脚分配，尤其是那些未在应用中使用的引脚。在所有情况下，未用的可选引脚的外设应被完全禁止。未用外设的输入应分配给未用的 RPn/RPin 引脚功能。具有未用的 RPn 功能的 I/O 引脚应配置为空外设输出。

将外设分配给特定引脚的操作不能自动执行引脚 I/O 电路的任何其他配置。理论上，也就是说将可选引脚的输出添加到某个引脚意味着在驱动输出时可能会无意中驱动现有的外设输入。用户必须熟悉其他共用一个可重映射引脚的固定外设的行为，了解何时使能或禁止这些外设。为了安全起见，共用一个引脚的固定数字外设在不用时应禁止。

遵照这些方针，对特定外设的可重映射的引脚进行配置并不会自动使能该功能。必须特别配置外设实现特定的操作并使能外设，如同将外设连接到固定引脚时一样。此操作在应用程序代码中的位置（紧跟在器件复位和外设配置之后或在主应用程序内）取决于外设及其在应用中的使用。

最后一个注意事项是外设引脚选择功能既不改写模拟输入也不会将具有模拟功能的引脚重新配置为数字 I/O。若某个引脚在器件复位时配置为模拟输入，则在使用该引脚的外设引脚选择功能时，必须明确将此引脚重新配置为数字 I/O。

例 11-3 所示为使用 UART1 实现带有流控制的双向通信的配置。使用了以下输入和输出功能：

- 输入功能：U1RX 和 U1CTS
- 输出功能：U1TX 和 U1RTS

## 例 11-3：配置 UART1 输入和输出功能

```
// Unlock Registers
asm volatile( "MOV    #OSCCON,    w1  \n"
              "MOV    #0x46,      w2  \n"
              "MOV    #0x57,      w3  \n"
              "MOV.b w2,        [w1] \n"
              "MOV.b w3,        [w1] \n"
              "BCLR OSCCON,#6" ) ;

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON & 0xbf);

// Configure Input Functions (Table 11-2)
// Assign U1RX To Pin RP0
RPINR18bits.U1RXR = 0;

// Assign U1CTS To Pin RP1
RPINR18bits.U1CTSR = 1;

// Configure Output Functions (Table 11-4)
// Assign U1TX To Pin RP2
RPOR1bits.RP2R = 3;

// Assign U1RTS To Pin RP3
RPOR1bits.RP3R = 4;

// Lock Registers
asm volatile ("MOV    #OSCCON,    w1  \n"
              "MOV    #0x46,      w2  \n"
              "MOV    #0x57,      w3  \n"
              "MOV.b w2,        [w1]\n"
              "MOV.b w3,        [w1]\n"
              "BSET   OSCCON,    #6" );

// or use C30 built-in macro:
// __builtin_write_OSCCONL(OSCCON | 0x40);
```

# PIC24FJ128GA310 系列

## 11.4.6 外设引脚选择寄存器

PIC24FJ128GA310 系列器件共实现了 35 个寄存器用于配置可重映射的外设：

- 输入可重映射外设寄存器 (19)
- 输出可重映射外设寄存器 (16)

注：仅在 IOLOCK (OSCCON<6>) = 0 时才能更改输入和输出寄存器的值。请参见第 11.4.4.1 节“控制寄存器锁定”了解特定的命令序列。

### 寄存器 11-7：RPINR0：外设引脚选择输入寄存器 0

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 INT1R<5:0>：将外部中断 1 (INT1) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-0 未实现：读为 0

### 寄存器 11-8：RPINR1：外设引脚选择输入寄存器 1

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT3R5	INT3R4	INT3R3	INT3R2	INT3R1	INT3R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 INT3R<5:0>：将外部中断 3 (INT3) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现：读为 0

bit 5-0 INT2R<5:0>：将外部中断 2 (INT2) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-9: RPINR2: 外设引脚选择输入寄存器 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	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	INT4R5	INT4R4	INT4R3	INT4R2	INT4R1	INT4R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 **INT4R<5:0>**: 将外部中断 4 (INT4) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-10: RPINR3: 外设引脚选择输入寄存器 3

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T3CKR5	T3CKR4	T3CKR3	T3CKR2	T3CKR1	T3CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T2CKR5	T2CKR4	T2CKR3	T2CKR2	T2CKR1	T2CKR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **T3CKR<5:0>**: 将 Timer3 外部时钟 (T3CK) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **T2CKR<5:0>**: 将 Timer2 外部时钟 (T2CK) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GA310 系列

## 寄存器 11-11： RPINR4：外设引脚选择输入寄存器 4

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T5CKR5	T5CKR4	T5CKR3	T5CKR2	T5CKR1	T5CKR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T4CKR5	T4CKR4	T4CKR3	T4CKR2	T4CKR1	T4CKR0
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 **T5CKR<5:0>**：将 Timer5 外部时钟 (T5CK) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现：读为 0

bit 5-0 **T4CKR<5:0>**：将 Timer4 外部时钟 (T4CK) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-12： RPINR7：外设引脚选择输入寄存器 7

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC2R5	IC2R4	IC2R3	IC2R2	IC2R1	IC2R0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC1R5	IC1R4	IC1R3	IC1R2	IC1R1	IC1R0
bit 7	bit 0						

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 **IC2R<5:0>**：将输入捕捉 2 (IC2) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现：读为 0

bit 5-0 **IC1R<5:0>**：将输入捕捉 1 (IC1) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-13: RPINR8: 外设引脚选择输入寄存器 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC4R5	IC4R4	IC4R3	IC4R2	IC4R1	IC4R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC3R5	IC3R4	IC3R3	IC3R2	IC3R1	IC3R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC4R<5:0>**: 将输入捕捉 4 (IC4) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC3R<5:0>**: 将输入捕捉 3 (IC3) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-14: RPINR9: 外设引脚选择输入寄存器 9

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC6R5	IC6R4	IC6R3	IC6R2	IC6R1	IC6R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC5R5	IC5R4	IC5R3	IC5R2	IC5R1	IC5R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **IC6R<5:0>**: 将输入捕捉 6 (IC6) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **IC5R<5:0>**: 将输入捕捉 5 (IC5) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GA310 系列

## 寄存器 11-15: RPINR10: 外设引脚选择输入寄存器 10

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-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	IC7R5	IC7R4	IC7R3	IC7R2	IC7R1	IC7R0
bit 7	bit 0						

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 **IC7R<5:0>**: 将输入捕捉 7 (IC7) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-16: RPINR11: 外设引脚选择输入寄存器 11

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	OCFBR5	OCFBR4	OCFBR3	OCFBR2	OCFBR1	OCFBR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **OCFBR<5:0>**: 将输出比较故障 B (OCFB) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **OCFAR<5:0>**: 将输出比较故障 A (OCFA) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-17: RPINR17: 外设引脚选择输入寄存器 17

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3RXR5	U3RXR4	U3RXR3	U3RXR2	U3RXR1	U3RXR0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U3RXR<5:0>**: 将 UART3 接收 (U3RX) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-0 未实现: 读为 0

## 寄存器 11-18: RPINR18: 外设引脚选择输入寄存器 18

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U1CTSR5	U1CTSR4	U1CTSR3	U1CTSR2	U1CTSR1	U1CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U1CTSR<5:0>**: 将 UART1 允许发送 ( $\overline{U1CTS}$ ) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U1RXR<5:0>**: 将 UART1 接收 (U1RX) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GA310 系列

## 寄存器 11-19: RPINR19: 外设引脚选择输入寄存器 19

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U2CTSR5	U2CTSR4	U2CTSR3	U2CTSR2	U2CTSR1	U2CTSR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U2CTSR<5:0>**: 将 UART2 允许发送 (U2CTS) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U2RXR<5:0>**: 将 UART2 接收 (U2RX) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-20: RPINR20: 外设引脚选择输入寄存器 20

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	SCK1R5	SCK1R4	SCK1R3	SCK1R2	SCK1R1	SCK1R0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

bit 15-14 未实现: 读为 0

bit 13-8 **SCK1R<5:0>**: 将 SPI2 时钟输入 (SCK2IN) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SDI1R<5:0>**: 将 SPI2 数据输入 (SDI2) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-21: RPINR21: 外设引脚选择输入寄存器 21

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U3CTSR5	U3CTSR4	U3CTSR3	U3CTSR2	U3CTSR1	U3CTSR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **U3CTSR<5:0>**: 将 UART3 允许发送 ( $\overline{\text{U3CTS}}$ ) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SS1R<5:0>**: 将 SPI1 从选择输入 (SS1IN) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-22: RPINR22: 外设引脚选择输入寄存器 22

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **T1CKR<5:0>**: 将 Timer1 外部时钟 (T1CK) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **SDI2R<5:0>**: 将 SPI2 数据输入 (SDI2) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GA310 系列

寄存器 11-23: RPINR23: 外设引脚选择输入寄存器 23

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	T1CKR5	T1CKR4	T1CKR3	T1CKR2	T1CKR1	T1CKR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **T1CKR<5:0>:**

bit 7-6 未实现: 读为 0

bit 5-0 **SS2R<5:0>:** 将 SPI2 从选择输入 (SS2IN) 分配给相应的 RPn 或 RPIn 引脚的位

寄存器 11-24: RPINR27: 外设引脚选择输入寄存器 27

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4CTSR5	U4CTSR4	U4CTSR3	U4CTSR2	U4CTSR1	U4CTSR0
bit 15	bit 8						

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	U4RXR5	U4RXR4	U4RXR3	U4RXR2	U4RXR1	U4RXR0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

bit 15-14 未实现: 读为 0

bit 13-8 **U4CTSR<5:0>:** 将 UART4 允许发送输入 (U4CTS) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **U4RXR<5:0>:** 将 UART4 接收输入 (U4RX) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-25: 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

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	MDMIR5	MDMIR4	MDMIR3	MDMIR2	MDMIR1	MDMIR0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-6 未实现: 读为 0

bit 5-0 **MDMIR<5:0>**: 将 TX 调制输入 (MDMI) 分配给相应的 RPn 或 RPIn 引脚的位

## 寄存器 11-26: RPINR31: 外设引脚选择输入寄存器 31

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	MDC2R5	MDC2R4	MDC2R3	MDC2R2	MDC2R1	MDC2R0
bit 15							bit 8

U-0	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
—	—	MDC1R5	MDC1R4	MDC1R3	MDC1R2	MDC21R1	MDC1R0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **MDC2R<5:0>**: 将 TX 载波输入 2 (MDCIN2) 分配给相应的 RPn 或 RPIn 引脚的位

bit 7-6 未实现: 读为 0

bit 5-0 **MDC1R<5:0>**: 将 SPI3 数据输入 (MDCIN1) 分配给相应的 RPn 或 RPIn 引脚的位

# PIC24FJ128GA310 系列

寄存器 11-27： 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
—	—	RP1R5	RP1R4	RP1R3	RP1R2	RP1R1	RP1R0
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
—	—	RP0R5	RP0R4	RP0R3	RP0R2	RP0R1	RP0R0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 RP1R<5:0>：RP1 输出引脚映射位

将外设输出编号 n 分配给引脚 RP1 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现：读为 0

bit 5-0 RP0R<5:0>：RP0 输出引脚映射位

将外设输出编号 n 分配给引脚 RP0 (请参见表 11-4 了解外设功能编号)

寄存器 11-28： 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
—	—	RP3R5	RP3R4	RP3R3	RP3R2	RP3R1	RP3R0
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
—	—	RP2R5	RP2R4	RP2R3	RP2R2	RP2R1	RP2R0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

x = 未知

bit 15-14 未实现：读为 0

bit 13-8 RP3R<5:0>：RP3 输出引脚映射位

将外设输出编号 n 分配给引脚 RP3 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现：读为 0

bit 5-0 RP2R<5:0>：RP2 输出引脚映射位

将外设输出编号 n 分配给引脚 RP2 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-29: 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
—	—	RP5R5 <sup>(1)</sup>	RP5R4 <sup>(1)</sup>	RP5R3 <sup>(1)</sup>	RP5R2 <sup>(1)</sup>	RP5R1 <sup>(1)</sup>	RP5R0 <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
—	—	RP4R5	RP4R4	RP4R3	RP4R2	RP4R1	RP4R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP5R<5:0>**: RP5 输出引脚映射位<sup>(1)</sup>

将外设输出编号 n 分配给引脚 RP5 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP4R<5:0>**: RP4 输出引脚映射位

将外设输出编号 n 分配给引脚 RP4 (请参见表 11-4 了解外设功能编号)

**注 1:** 在 64 引脚器件上未实现这些位, 读为 0。

## 寄存器 11-30: 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
—	—	RP7R5	RP7R4	RP7R3	RP7R2	RP7R1	RP7R0
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
—	—	RP6R5	RP6R4	RP6R3	RP6R2	RP6R1	RP6R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP7R<5:0>**: RP7 输出引脚映射位

将外设输出编号 n 分配给引脚 RP7 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP6R<5:0>**: RP6 输出引脚映射位

将外设输出编号 n 分配给引脚 RP6 (请参见表 11-4 了解外设功能编号)

# PIC24FJ128GA310 系列

寄存器 11-31: 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
—	—	RP9R5	RP9R4	RP9R3	RP9R2	RP9R1	RP9R0
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
—	—	RP8R5	RP8R4	RP8R3	RP8R2	RP8R1	RP8R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP9R<5:0>: RP9 输出引脚映射位

将外设输出编号 n 分配给引脚 RP9 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP8R<5:0>: RP8 输出引脚映射位

将外设输出编号 n 分配给引脚 RP8 (请参见表 11-4 了解外设功能编号)

寄存器 11-32: 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
—	—	RP11R5	RP11R4	RP11R3	RP11R2	RP11R1	RP11R0
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
—	—	RP10R5	RP10R4	RP10R3	RP10R2	RP10R1	RP10R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP11R<5:0>: RP11 输出引脚映射位

将外设输出编号 n 分配给引脚 RP11 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP10R<5:0>: RP10 输出引脚映射位

将外设输出编号 n 分配给引脚 RP10 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-33: 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
—	—	RP13R5	RP13R4	RP13R3	RP13R2	RP13R1	RP13R0
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
—	—	RP12R5	RP12R4	RP12R3	RP12R2	RP12R1	RP12R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP13R<5:0>: RP13** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP13 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP12R<5:0>: RP12** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP12 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-34: 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
—	—	RP15R5 <sup>(1)</sup>	RP15R4 <sup>(1)</sup>	RP15R3 <sup>(1)</sup>	RP15R2 <sup>(1)</sup>	RP15R1 <sup>(1)</sup>	RP15R0 <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
—	—	RP14R5	RP14R4	RP14R3	RP14R2	RP14R1	RP14R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP15R<5:0>: RP15** 输出引脚映射位 <sup>(1)</sup>

将外设输出编号 n 分配给引脚 RP15 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP14R<5:0>: RP14** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP14 (请参见表 11-4 了解外设功能编号)

**注 1:** 在 64 引脚器件上未实现这些位, 读为 0。

# PIC24FJ128GA310 系列

寄存器 11-35: 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
—	—	RP17R5	RP17R4	RP17R3	RP17R2	RP17R1	RP17R0
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
—	—	RP16R5	RP16R4	RP16R3	RP16R2	RP16R1	RP16R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP17R<5:0>: RP17 输出引脚映射位

将外设输出编号 n 分配给引脚 RP17 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP16R<5:0>: RP16 输出引脚映射位

将外设输出编号 n 分配给引脚 RP16 (请参见表 11-4 了解外设功能编号)

寄存器 11-36: 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
—	—	RP19R5	RP19R4	RP19R3	RP19R2	RP19R1	RP19R0
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
—	—	RP18R5	RP18R4	RP18R3	RP18R2	RP18R1	RP18R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP19R<5:0>: RP19 输出引脚映射位

将外设输出编号 n 分配给引脚 RP19 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP18R<5:0>: RP18 输出引脚映射位

将外设输出编号 n 分配给引脚 RP18 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-37: 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
—	—	RP21R5	RP21R4	RP21R3	RP21R2	RP21R1	RP21R0
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
—	—	RP20R5	RP20R4	RP20R3	RP20R2	RP20R1	RP20R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP21R<5:0>: RP21** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP21 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP20R<5:0>: RP20** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP20 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-38: 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
—	—	RP23R5	RP23R4	RP23R3	RP23R2	RP23R1	RP23R0
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
—	—	RP22R5	RP22R4	RP22R3	RP22R2	RP22R1	RP22R0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP23R<5:0>: RP23** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP23 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP22R<5:0>: RP22** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP22 (请参见表 11-4 了解外设功能编号)

# PIC24FJ128GA310 系列

寄存器 11-39: 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
—	—	RP25R5	RP25R4	RP25R3	RP25R2	RP25R1	RP25R0
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
—	—	RP24R5	RP24R4	RP24R3	RP24R2	RP24R1	RP24R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP25R<5:0>: RP25 输出引脚映射位

将外设输出编号 n 分配给引脚 RP25 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP24R<5:0>: RP24 输出引脚映射位

将外设输出编号 n 分配给引脚 RP24 (请参见表 11-4 了解外设功能编号)

寄存器 11-40: 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
—	—	RP27R5	RP27R4	RP27R3	RP27R2	RP27R1	RP27R0
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
—	—	RP26R5	RP26R4	RP26R3	RP26R2	RP26R1	RP26R0
bit 7				bit 0			

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 RP27R<5:0>: RP27 输出引脚映射位

将外设输出编号 n 分配给引脚 RP27 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 RP26R<5:0>: RP26 输出引脚映射位

将外设输出编号 n 分配给引脚 RP26 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-41: 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
—	—	RP29R5	RP29R4	RP29R3	RP29R2	RP29R1	RP29R0
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
—	—	RP28R5	RP28R4	RP28R3	RP28R2	RP28R1	RP28R0
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP29R<5:0>: RP29** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP29 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP28R<5:0>: RP28** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP28 (请参见表 11-4 了解外设功能编号)

## 寄存器 11-42: 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
—	—	RP31R5 <sup>(1)</sup>	RP31R4 <sup>(1)</sup>	RP31R3 <sup>(1)</sup>	RP31R2 <sup>(1)</sup>	RP31R1 <sup>(1)</sup>	RP31R0 <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
—	—	RP30R5	RP30R4	RP30R3	RP30R2	RP30R1	RP30R0
bit 7							bit 0

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现: 读为 0

bit 13-8 **RP31R<5:0>: RP31** 输出引脚映射位<sup>(1)</sup>

将外设输出编号 n 分配给引脚 RP31 (请参见表 11-4 了解外设功能编号)

bit 7-6 未实现: 读为 0

bit 5-0 **RP30R<5:0>: RP30** 输出引脚映射位

将外设输出编号 n 分配给引脚 RP30 (请参见表 11-4 了解外设功能编号)

**注 1:** 在 64 引脚器件和 80 引脚器件上未实现这些位, 读为 0。

# **PIC24FJ128GA310 系列**

---

---

注:

## 12.0 TIMER1

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 14 章“定时器”(DS39704A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

Timer1 模块是一个 16 位定时器, 可作为实时时钟 (RTC) 的时间计数器, 或作为独立运行的间隔定时器/计数器。Timer1 可在以下三种模式下工作:

- 16 位定时器
- 16 位同步计数器
- 16 位异步计数器

而且, Timer1 还支持下列特性:

- 定时器门控操作
- 可选的预分频比设置
- CPU 空闲和休眠模式下的定时器操作
- 在 16 位周期寄存器匹配或  
外部门控信号的下降沿产生中断

图 12-1: 16 位 TIMER1 模块框图

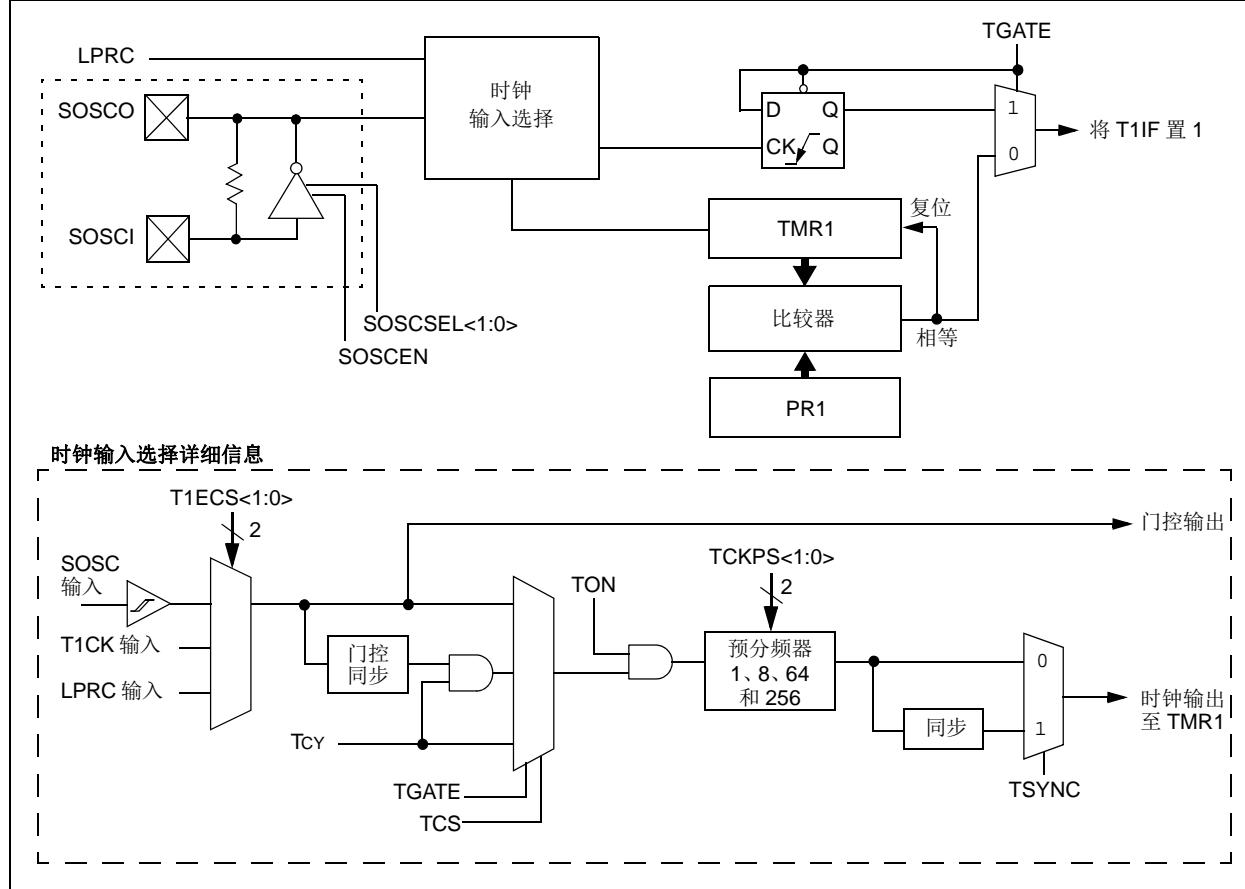


图 12-1 给出了 16 位定时器模块的框图。

配置 Timer1 的步骤:

1. 将 TON 位置 1 (=1)。
2. 使用 TCKPS<1:0> 位选择定时器预分频比。
3. 使用 TCS、TECS 和 TGATE 位设置时钟和门控模式。
4. 将 TSYNC 位置 1 或清零以配置同步或异步操作。
5. 将定时器周期值装入 PR1 寄存器。
6. 如果需要中断, 则将中断允许位 T1IE 置 1。使用优先级位 T1IP<2:0> 来设置中断的优先级。

# PIC24FJ128GA310 系列

## 寄存器 12-1: T1CON: TIMER1 控制寄存器<sup>(1)</sup>

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
TON	—	TSIDL	—	—	—	TIECS1	TIECS0
bit 15	bit 8						

U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	—	TSYNC	TCS	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **TON:** Timer1 使能位  
1 = 启动 16 位 Timer1  
0 = 停止 16 位 Timer1

bit 14      未实现: 读为 0

bit 13      **TSIDL:** 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    未实现: 读为 0

bit 9-8      **TIECS<1:0>:** Timer1 扩展时钟源选择位 (当 TCS = 1 时选择)  
11 = 未实现, 不要使用  
10 = LPRC 振荡器  
01 = T1CK 外部时钟输入  
00 = SOSC

bit 7      未实现: 读为 0

bit 6      **TGATE:** Timer1 门控时间累加使能位  
当 TCS = 1 时:  
忽略此位。

当 TCS = 0 时:  
1 = 使能门控时间累加  
0 = 禁止门控时间累加

bit 5-4     **TCKPS<1:0>:** 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      **TSYNC:** Timer1 外部时钟输入同步选择位  
当 TCS = 1 时:  
1 = 同步外部时钟输入  
0 = 不同步外部时钟输入  
当 TCS = 0 时:  
忽略此位。

bit 1      **TCS:** Timer1 时钟源选择位  
1 = 由定时器选择扩展时钟  
0 = 内部时钟 (Fosc/2)

bit 0      未实现: 读为 0

**注 1:** 定时器运行 (TON = 1) 时改变 T1CON 的值会导致定时器预分频计数器复位, 不推荐这么做。

## 13.0 TIMER2/3 和 TIMER4/5

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 14 章“定时器”(DS39704A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

Timer2/3 和 Timer4/5 模块各为一个 32 位定时器, 它们也可以被配置为 4 个具有可选工作模式的独立 16 位定时器。

作为 32 位定时器, Timer2/3 和 Timer4/5 可以在以下三种模式下工作:

- 两个独立的 16 位定时器, 可实现所有 16 位工作模式 (异步计数器模式除外)
- 一个 32 位定时器
- 一个 32 位同步计数器

它们还支持以下功能:

- 定时器门控操作
- 可选的预分频比设置
- 空闲和休眠模式下作为定时器工作
- 在 32 位周期寄存器匹配时产生中断
- A/D 事件触发信号 (仅 32 位模式的 Timer2/3 和 16 位模式的 Timer3)

在作为 16 位定时器时, 所有 4 个定时器都可以单独用作同步定时器或计数器。它们也可以提供上面列出的功能, 但触发 A/D 事件除外。该触发器只能在 32 位模式的 Timer2/3 和 16 位模式的 Timer3 上实现。工作模式和使能的功能由 T2CON、T3CON、T4CON 和 T5CON 寄存器中的相应位的设置决定。寄存器 13-1 给出了 T2CON 和 T4CON 的一般形式; 寄存器 13-2 给出了 T3CON 和 T5CON 的一般形式。

当工作在 32 位定时器/计数器模式时, Timer2 和 Timer4 作为低字, 而 Timer3 和 Timer5 则作为高字。

**注:** 当定时器工作在 32 位模式时, 将不使用 T3CON 和 T5CON 的控制位。只有 T2CON 和 T4CON 的控制位用于设置和控制。Timer2 和 Timer4 提供 32 位定时器模块的时钟和门控输入, 但产生中断时, 会将 Timer3 或 Timer5 的中断标志位置 1。

要将 Timer2/3 或 Timer4/5 配置为 32 位工作模式:

1. 将 T32 或 T45 位置 1 (T2CON<3> 或 T4CON<3> = 1)。
2. 使用 TCKPS<1:0> 位选择 Timer2 或 Timer4 的预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。如果 TCS 被设置为外部时钟, 则必须将 RPINRx (TxCK) 配置给可用的 RPn/RPln 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。
4. 装载定时器周期值。PR3 (或 PR5) 将存放该值的高字 (msw), 而 PR2 (或 PR4) 则存放低字 (lsw)。
5. 如果需要中断, 则将中断允许位 T3IE 或 T5IE 置 1。使用优先级位 T3IP<2:0> 或 T5IP<2:0> 来设置中断的优先级。请注意, 虽然 Timer2 或 Timer4 控制定时器, 但其中断却表现为 Timer3 或 Timer5 中断。
6. 将 TON 位置 1 (= 1)。

任意时刻的定时器值都将存储在寄存器对 TMR<3:2> (或 TMR<5:4>) 中。TMR3 (TMR5) 始终存放计数值的高字, 而 TMR2 (TMR4) 存放低字。

要将任何定时器配置为独立的 16 位工作模式:

1. 将与该定时器对应的位 T32 (对于 Timer2 和 Timer3 来说是 T2CON<3>; 对于 Timer4 和 Timer5 来说是 T4CON<3>) 清零。
2. 使用 TCKPS<1:0> 位选择定时器预分频比。
3. 使用 TCS 和 TGATE 位设置时钟和门控模式。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。
4. 将定时器周期值装入 PRx 寄存器。
5. 如果需要中断, 则将中断允许位 TxIE 置 1。使用优先级位 TxIP<2:0> 来设置中断的优先级。
6. 将 TON 位置 1 (TxCON<15> = 1)。

# PIC24FJ128GA310 系列

图 13-1：TIMER2/3 和 TIMER4/5（32 位）框图

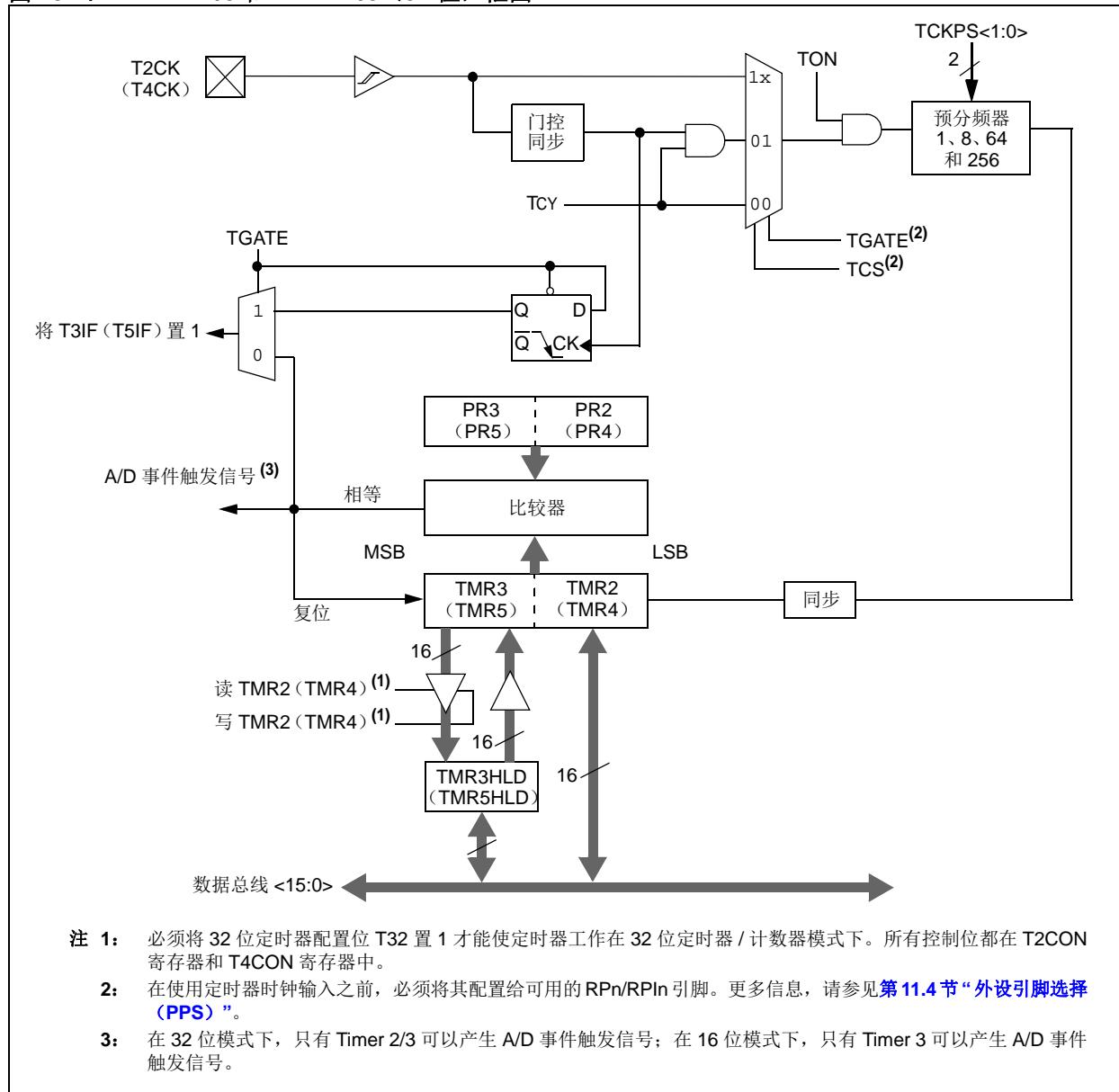


图 13-2: TIMER2 和 TIMER4 (16 位同步) 框图

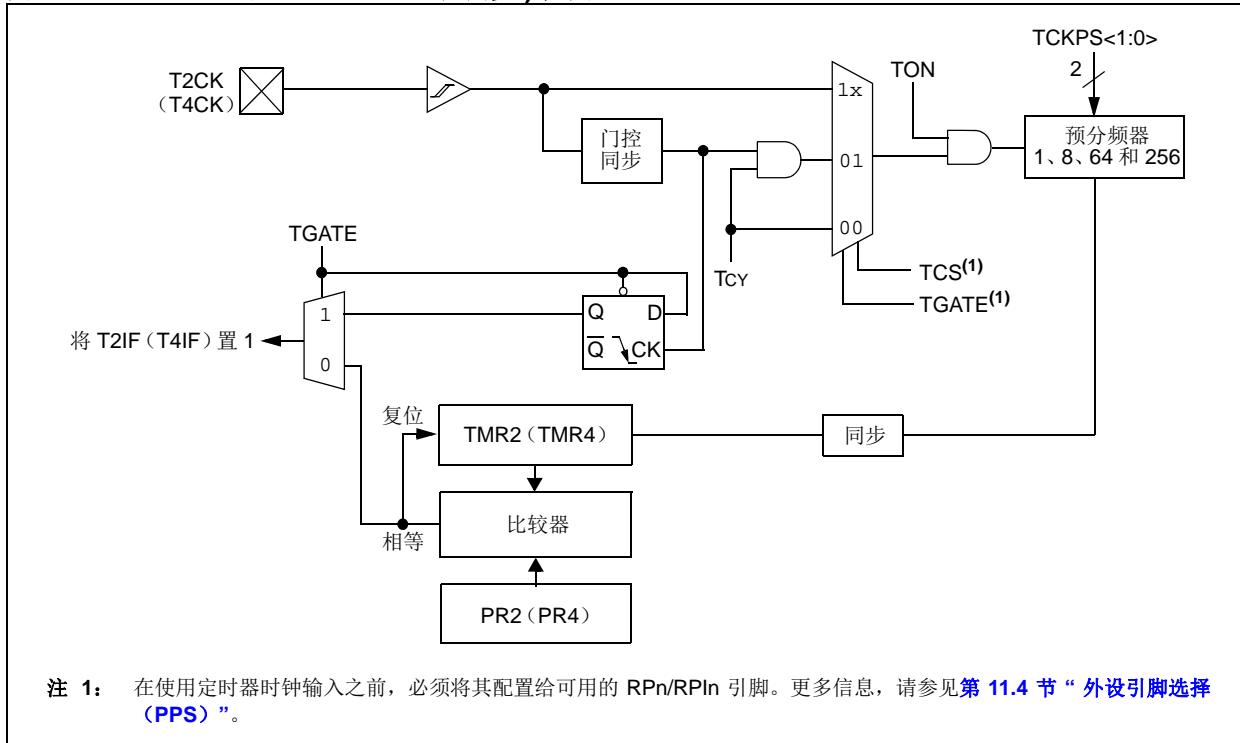
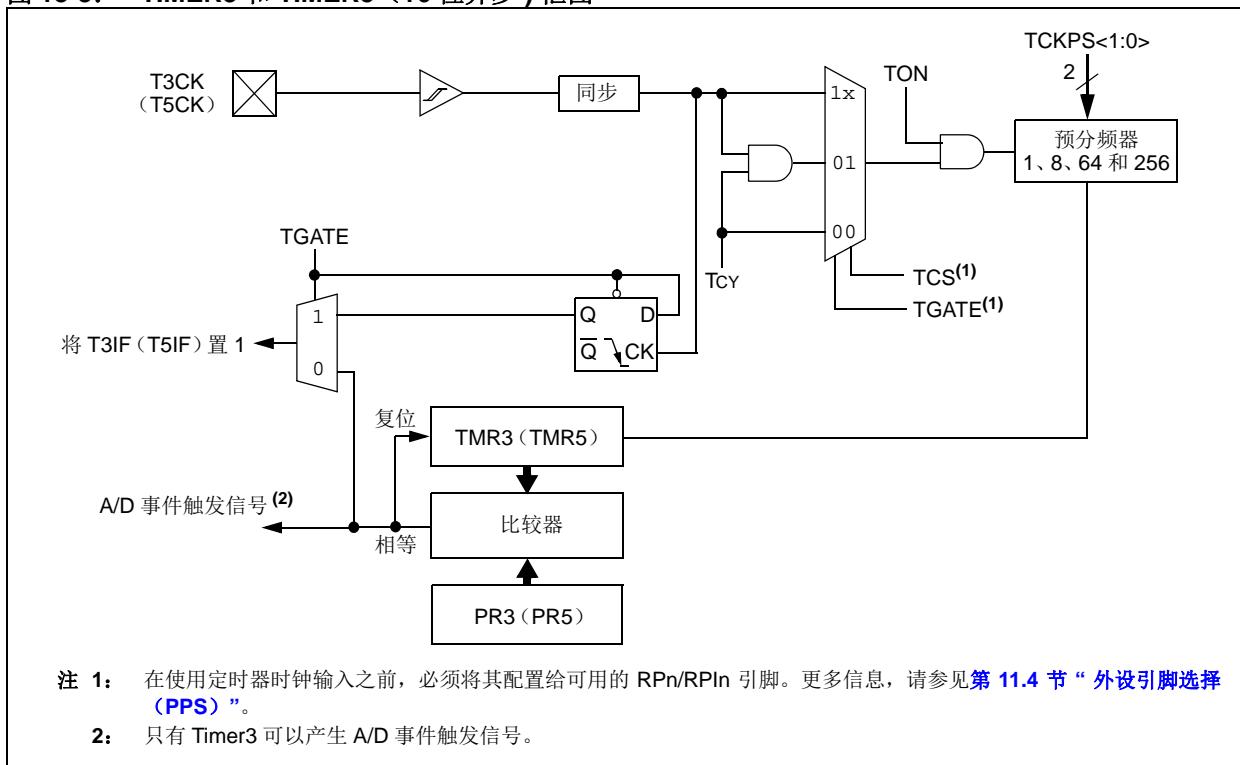


图 13-3: TIMER3 和 TIMER5 (16 位异步) 框图



# PIC24FJ128GA310 系列

寄存器 13-1：TxCON：TIMER2 和 TIMER4 控制寄存器<sup>(3)</sup>

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS1	TCKPS0	T32 <sup>(1)</sup>	—	TCS <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **TON:** Timerx 使能位

当 TxCON<3> = 1 时：

1 = 启动 32 位 Timerx/y

0 = 停止 32 位 Timerx/y

当 TxCON<3> = 0 时：

1 = 启动 16 位 Timerx

0 = 停止 16 位 Timerx

bit 14      未实现：读为 0

bit 13      **TSIDL:** 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 12-7    未实现：读为 0

bit 6        **TGATE:** Timerx 门控时间累加使能位

当 TCS = 1 时：

忽略此位。

当 TCS = 0 时：

1 = 使能门控时间累加

0 = 禁止门控时间累加

bit 5-4     **TCKPS<1:0>:** Timerx 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        **T32:** 32 位定时器模式选择位<sup>(1)</sup>

1 = Timerx 和 Timery 构成一个 32 位定时器

0 = Timerx 和 Timery 作为两个 16 位定时器

在 32 位模式下，T3CON 控制位不影响 32 位定时器的工作。

bit 2        未实现：读为 0

bit 1        **TCS:** Timerx 时钟源选择位<sup>(2)</sup>

1 = TxCK 引脚的外部时钟的上升沿

0 = 内部时钟 (Fosc/2)

bit 0        未实现：读为 0

注 1：在 T4CON 中，将实现 T45 位（而非 T32 位）来选择 32 位模式。在 32 位模式下，T3CON 或 T5CON 控制位不影响 32 位定时器的工作。

2：如果 TCS = 1，必须将 RPINRx (TxCK) 配置给可用的 RPn/RPIn 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS)”。

3：定时器运行 (TON = 1) 时改变 TxCON 的值会导致定时器预分频计数器复位，不推荐这么做。

## 寄存器 13-2: TyCON: TIMER3 和 TIMER5 控制寄存器<sup>(3)</sup>

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

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE <sup>(1)</sup>	TCKPS1 <sup>(1)</sup>	TCKPS0 <sup>(1)</sup>	—	—	TCS <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>TON:</b> Timery 使能位 <sup>(1)</sup>
	1 = 启动 16 位 Timery
	0 = 停止 16 位 Timery
bit 14	未实现: 读为 0
bit 13	<b>TSIDL:</b> 空闲模式停止位 <sup>(1)</sup>
	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 12-7	未实现: 读为 0
bit 6	<b>TGATE:</b> Timery 门控时间累加使能位 <sup>(1)</sup>
	<u>当 TCS = 1 时:</u> 忽略此位。
	<u>当 TCS = 0 时:</u> 1 = 使能门控时间累加 0 = 禁止门控时间累加
bit 5-4	<b>TCKPS&lt;1:0&gt;:</b> Timery 输入时钟预分频比选择位 <sup>(1)</sup>
	11 = 1:256
	10 = 1:64
	01 = 1:8
	00 = 1:1
bit 3-2	未实现: 读为 0
bit 1	<b>TCS:</b> Timery 时钟源选择位 <sup>(1,2)</sup>
	1 = TyCK 引脚的外部时钟的上升沿
	0 = 内部时钟 (FOSC/2)
bit 0	未实现: 读为 0

**注 1:** 当使能 32 位工作模式时 (T2CON<3> 或 T4CON<3> = 1) , 这些位不会对 Timery 的工作产生影响。定时器的所有功能都通过 T2CON 和 T4CON 寄存器设置。

**2:** 如果 TCS = 1, 必须将 RPINRx (TxCK) 配置给可用的 RPn/RPIn 引脚。更多信息, 请参见第 11.4 节 “外设引脚选择 (PPS) ”。

**3:** 定时器运行 (TON = 1) 时改变 TyCON 的值会导致定时器预分频计数器复位, 不推荐这么做。

# **PIC24FJ128GA310 系列**

---

---

注:

## 14.0 带专用定时器的输入捕捉

**注：**本数据手册总结了该组PIC24F器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F系列参考手册》中的第34章“带专用定时器的输入捕捉”（DS39722A\_CN）。本数据手册中的信息取代了FRM中的信息。

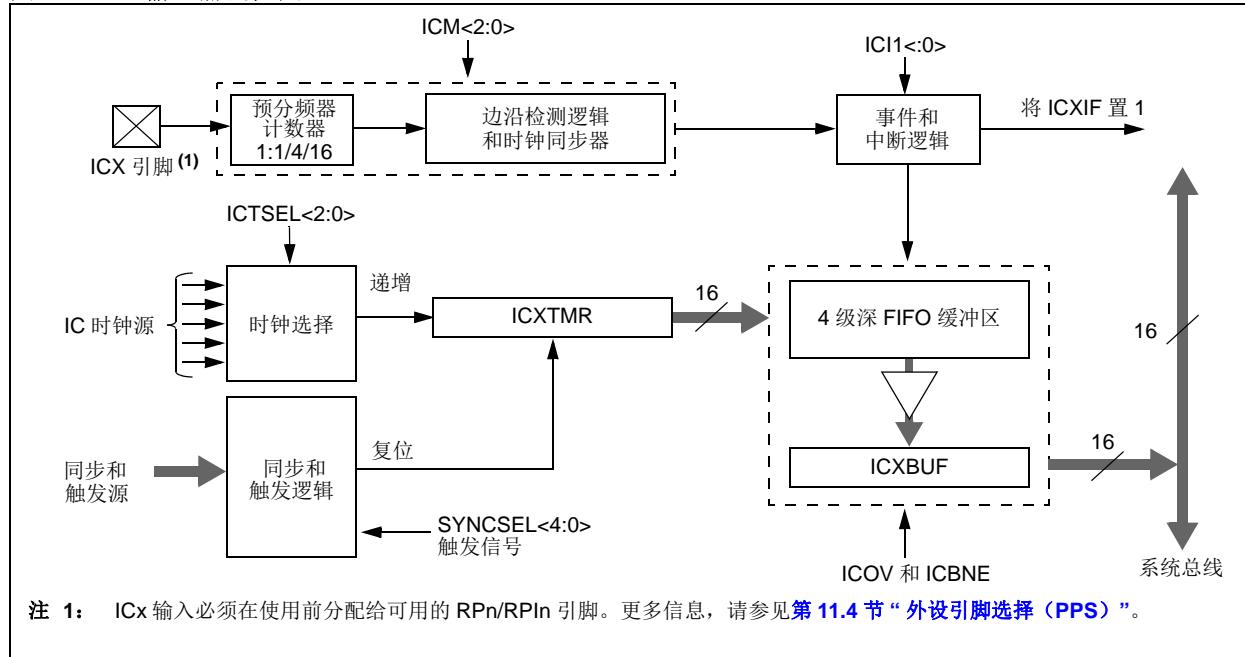
PIC24FJ128GA310 系列中的器件包含 7 个独立的输入捕捉模块。每个模块都为捕捉外部脉冲事件和产生中断提供了各种配置和工作选项。

输入捕捉模块的主要特性包括：

- 通过级联两个相邻的模块，可采用硬件方式配置为所有的 32 位工作模式
- 输出比较操作有同步和触发两种模式，最多具有 30 个用户可选的同步 / 触发源
- 一个 4 级深的 FIFO 缓冲区，用于捕捉并保存多个事件的定时器值
- 可配置中断产生
- 为每个模块提供了最多 6 个时钟源，驱动独立的内部 16 位计数器

此模块可通过两个寄存器进行控制：ICxCON1（[寄存器 14-1](#)）和 ICxCON2（[寄存器 14-2](#)）。模块的一般框图如图 14-1 所示。

图 14-1： 输入捕捉框图



## 14.1 一般工作模式

### 14.1.1 同步和触发模式

输入捕捉模块工作在自由运行模式下时，内部 16 位计数器 ICxTMR 可连续向上计数，并在每次溢出时从 FFFFh 折回 0000h。其周期与所选的外部时钟源同步。发生捕捉事件时，将内部计数器的当前 16 位值写入 FIFO 缓冲区。

在同步模式下，一旦使能了所选的时钟源，模块就开始捕捉 ICx 引脚上的事件。只要所选的同步源发生事件，内部计数器就会复位。在触发模式下，模块等待来自其他内部模块的同步事件发生后，才允许内部计数器运行。

通过将 SYNCSEL 位（ICxCON2<4:0>）设置为 00000 并清零 ICTRIG 位（ICxCON2<7>）选择标准的自由运行工作模式。当 SYNCSEL 位设置为除 00000 以外的其他值时，将选择同步和触发模式。ICTRIG 位选择同步或触发模式；置 1 该位将选择触发工作模式。在两种模式下，SYNCSEL 位用于决定同步 / 触发源。

当 SYNCSEL 位设置为 00000 且 ICTRIG 置 1 时，模块工作在软件触发模式下。在此情况下，可通过手动将 TRIGSTAT 位（ICxCON2<6>）置 1 来启动捕捉操作。

## 14.1.2 级联 (32 位) 模式

默认情况下，每个模块都可以使用自己的 16 位定时器工作。要提高分辨率，可将相邻的偶数编号的模块和奇数编号的模块配置为一个 32 位模块。（例如，模块 1 和 2 配对，模块 3 和 4 配对等）。奇数编号的模块（ICx）提供 32 位寄存器对中的低 16 位，而偶数编号的模块（ICy）提供高 16 位。ICx 寄存器的折回操作会导致其相应的 ICy 寄存器递增。

通过将两个模块的 IC32 位（ICxCON2<8>）置 1 使用硬件来配置级联操作。

## 14.2 捕捉操作

输入捕捉模块可配置为在 ICx 信号的上升沿或所有跳变沿处捕捉定时器值并产生中断。捕捉可配置在所有上升沿上发生，或仅在某些上升沿（例如每 4 个或 16 个上升沿）发生。可单独配置中断以在每个事件或一组事件发生时产生中断。

将模块设置为捕捉操作：

1. 将 ICx 输入配置为可用的外设引脚选择引脚之一。
2. 如果要使用同步模式，应在继续前禁止同步源。
3. 确保 ICBNE 位（ICxCON1<3>）清零之前已通过读 ICxBUF 从 FIFO 中除去了所有以前的数据。
4. 设置 SYNCSEL 位（ICxCON2<4:0>）以得到所需的同步 / 触发源。
5. 设置 ICTSEL 位（ICxCON1<12:10>）以得到所需的时钟源。
6. 设置 ICI 位（ICxCON1<6:5>）以得到所需的中断频率。
7. 选择同步或触发工作模式：
  - a) 检查 SYNCSEL 位是否设置为 00000。
  - b) 对于同步模式，清零 ICTRIG 位（ICxCON2<7>）。
  - c) 对于触发模式，将 ICTRIG 位置 1 并清零 TRIGSTAT 位（ICxCON2<6>）。
8. 设置 ICM 位（ICxCON1<2:0>）以得到所需的工作模式。
9. 使能所选的触发 / 同步源。

对于 32 位级联操作，设置过程稍有不同：

1. 将两个模块的 IC32 位（ICyCON2<8>和 ICxCON2<8>）置 1，首先使能偶数编号的模块。这可以确保模块一起启动。
2. 分别设置两个模块的 ICTSEL 和 SYNCSEL 位以选择相同的同步 / 触发模式和时基源。首先设置偶数编号的模块，然后设置奇数编号的模块。两个模块必须使用相同的 ICTSEL 和 SYNCSEL 位设置。
3. 清零偶数编号的模块的 ICTRIG 位（ICyCON2<7>）。这可以强制该模块与奇数编号的模块运行在同步模式下（不管其触发设置如何）。
4. 使用奇数编号的模块的 ICI 位（ICxCON1<6:5>）设置所需的中断频率。
5. 使用奇数编号的模块的 ICTRIG 位（ICxCON2<7>）配置触发或同步工作模式。

**注：** 对于同步工作模式，最后一个步骤是使能同步源。使能同步源之前，两个输入捕捉模块都保持复位状态。

6. 使用奇数编号的模块的 ICM 位（ICxCON1<2:0>）设置所需的捕捉模式。

使能了时基和同步 / 触发源之后，模块准备就绪，可以捕捉事件了。ICBNE 位（ICxCON1<3>）置 1 时，FIFO 中至少有一个捕捉值。从 FIFO 读取输入捕捉值，直到 ICBNE 清零。

对于 32 位操作，读取 ICxBUF 和 ICyBUF 以得到整个 32 位定时器值（ICxBUF 用于低 16 位，ICyBUF 用于高 16 位）。当奇数编号的模块的 ICBNE 位（ICxCON1<3>）置 1 时，FIFO 缓冲区中至少有一个捕捉值。继续读缓冲寄存器直到 ICBNE 清零（由硬件自动执行）。

## 寄存器 14-1： ICxCON1： 输入捕捉 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
—	—	ICSIDL	ICTSEL2	ICTSEL1	ICTSEL0	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
—	ICI1	ICI0	ICOV	ICBNE	ICM2 <sup>(1)</sup>	ICM1 <sup>(1)</sup>	ICM0 <sup>(1)</sup>
bit 7	bit 0						

图注： HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14	未实现：读为 0
bit 13	<b>ICSIDL：</b> 空闲时输入捕捉 x 模块停止控制位 1 = 输入捕捉模块将在 CPU 空闲模式下停止工作 0 = 输入捕捉模块将在 CPU 空闲模式下继续工作
bit 12-10	<b>ICTSEL&lt;2:0&gt;：</b> 输入捕捉定时器选择位 111 = 系统时钟 (Fosc/2) 110 = 保留 101 = 保留 100 = Timer1 011 = Timer5 010 = Timer4 001 = Timer2 000 = Timer3
bit 9-7	未实现：读为 0
bit 6-5	<b>ICI&lt;1:0&gt;：</b> 选择每次中断发生的捕捉次数的位 11 = 每四次捕捉事件中断一次 10 = 每三次捕捉事件中断一次 01 = 每两次捕捉事件中断一次 00 = 每次捕捉事件中断一次
bit 4	<b>ICOV：</b> 输入捕捉 x 溢出状态标志位（只读） 1 = 已发生输入捕捉溢出 0 = 未发生输入捕捉溢出
bit 3	<b>ICBNE：</b> 输入捕捉 x 缓冲器空状态位（只读） 1 = 输入捕捉缓冲器不为空，至少可以读取一个以上的捕捉值 0 = 输入捕捉缓冲器为空
bit 2-0	<b>ICM&lt;2:0&gt;：</b> 输入捕捉模式选择位 <sup>(1)</sup> 111 = 中断模式：只有当器件处于休眠或空闲模式时输入捕捉才可作为中断引脚（只检测上升沿，所有其他控制位都不适用） 110 = 未使用（禁止模块） 101 = 预分频器捕捉模式：每 16 个上升沿捕捉一次 100 = 预分频器捕捉模式：每 4 个上升沿捕捉一次 011 = 简单捕捉模式：每个上升沿捕捉一次 010 = 简单捕捉模式：每个下降沿捕捉一次 001 = 边沿检测捕捉模式：每个沿（上升和下降）捕捉一次， ICI<1:0> 位不控制此模式下中断的产生 000 = 输入捕捉模块关闭

注 1： ICx 输入也必须配置给可用的 RPn/RPIn 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS)”。

# PIC24FJ128GA310 系列

## 寄存器 14-2: ICxCON2: 输入捕捉 x 控制寄存器 2

U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	IC32
bit 15							bit 8

R/W-0	R/W-0 HS	U-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-1
ICTRIG	TRIGSTAT	—	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7							bit 0

图注:	HS = 可由硬件置 1 的位
R = 可读位	W = 可写位
-n = 上电复位时的值	1 = 置 1
	0 = 清零
	x = 未知

bit 15-9 未实现: 读为 0

bit 8 **IC32:** 级联两个输入捕捉模块使能位 (32 位操作)

1 = ICx 和 ICy 以级联的形式作为 32 位模块使用 (两个模块中的此位都必须置 1)  
0 = ICx 作为 16 位模块独立工作

bit 7 **ICTRIG:** ICx 同步 / 触发选择位

1 = 用 SYNCSELx 位指定的源触发 ICx  
0 = 使 ICx 与 SYNCSELx 位指定的源同步

bit 6 **TRIGSTAT:** 定时器触发状态位

1 = 定时器源被触发且正在运行 (该位可用硬件置 1, 也可用软件置 1)  
0 = 定时器源未被触发, 保持清零状态

bit 5 未实现: 读为 0

**注 1:** 这些输入仅用作触发源, 不能用作同步源。

**2:** 请勿通过选择该模式将输入捕捉模块作为自己的触发源使用。

## 寄存器 14-2: IC<sub>x</sub>CON2: 输入捕捉 x 控制寄存器 2 (续)

bit 4-0      **SYNCSEL<4:0>**: 同步 / 触发源选择位

11111 = 保留  
11110 = 保留 <sup>(2)</sup>  
11101 = 保留 <sup>(2)</sup>  
11100 = CTMU<sup>(1)</sup>  
11011 = A/D<sup>(1)</sup>  
11010 = 比较器 3<sup>(1)</sup>  
11001 = 比较器 2<sup>(1)</sup>  
11000 = 比较器 1<sup>(1)</sup>  
10111 = 保留 <sup>(2)</sup>  
10110 = 输入捕捉 7<sup>(2)</sup>  
10101 = 输入捕捉 6<sup>(2)</sup>  
10100 = 输入捕捉 5<sup>(2)</sup>  
10011 = 输入捕捉 4<sup>(2)</sup>  
10010 = 输入捕捉 3<sup>(2)</sup>  
10001 = 输入捕捉 2<sup>(2)</sup>  
10000 = 输入捕捉 1<sup>(2)</sup>  
01111 = Timer5  
01110 = Timer4  
01101 = Timer3  
01100 = Timer2  
01011 = Timer1  
01010 = 保留  
01001 = 保留  
01000 = 保留  
00111 = 输出比较 7  
•  
•  
•  
00010 = 输出比较 2  
00001 = 输出比较 1  
00000 = 不与任何其他模块同步

**注 1:** 这些输入仅用作触发源，不能用作同步源。

**2:** 请勿通过选择该模式将输入捕捉模块作为自己的触发源使用。

# **PIC24FJ128GA310 系列**

---

---

注:

## 15.0 带专用定时器的输出比较

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 35 章“带专用定时器的输出比较”(DS39723A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

PIC24FJ128GA310 系列中的器件采用了 7 个独立的输出比较模块。每个模块都为发生器件内部事件时产生连续脉冲信号提供了各种配置和工作选项, 还可产生脉宽调制波形以驱动功率应用。

输出比较模块的主要特性包括:

- 通过级联两个相邻的模块, 可采用硬件方式配置为所有的 32 位工作模式
- 输出比较操作有同步和触发两种模式, 可使用最多 31 个用户可选的触发 / 同步源
- 两个单独的周期寄存器 (一个主寄存器 OCxR 和一个辅助寄存器 OCxRS), 可更加灵活地产生宽度可变的脉冲
- 可配置为在发生输出事件时产生单脉冲信号或连续脉冲信号, 或产生连续的 PWM 波形
- 每个模块最多可提供 6 种时钟源来驱动独立的内部 16 位计数器

## 15.1 一般工作模式

### 15.1.1 同步和触发模式

输出比较模块工作在自由运行模式下时, 内部 16 位计数器 OCxTMR 可连续向上计数, 并在每次溢出时从 0xFFFF 折回 0x0000。其周期与所选的外部时钟源同步。当内部计数器与其中一个周期寄存器发生匹配时, 产生一个比较或 PWM 事件。

在同步模式下, 一旦使能了所选的时钟源, 模块就开始执行比较或 PWM 操作。只要所选的同步源发生事件, 模块的内部计数器都会复位。在触发模式下, 模块等待来自其他内部模块的同步事件发生后, 才允许该计数器运行。

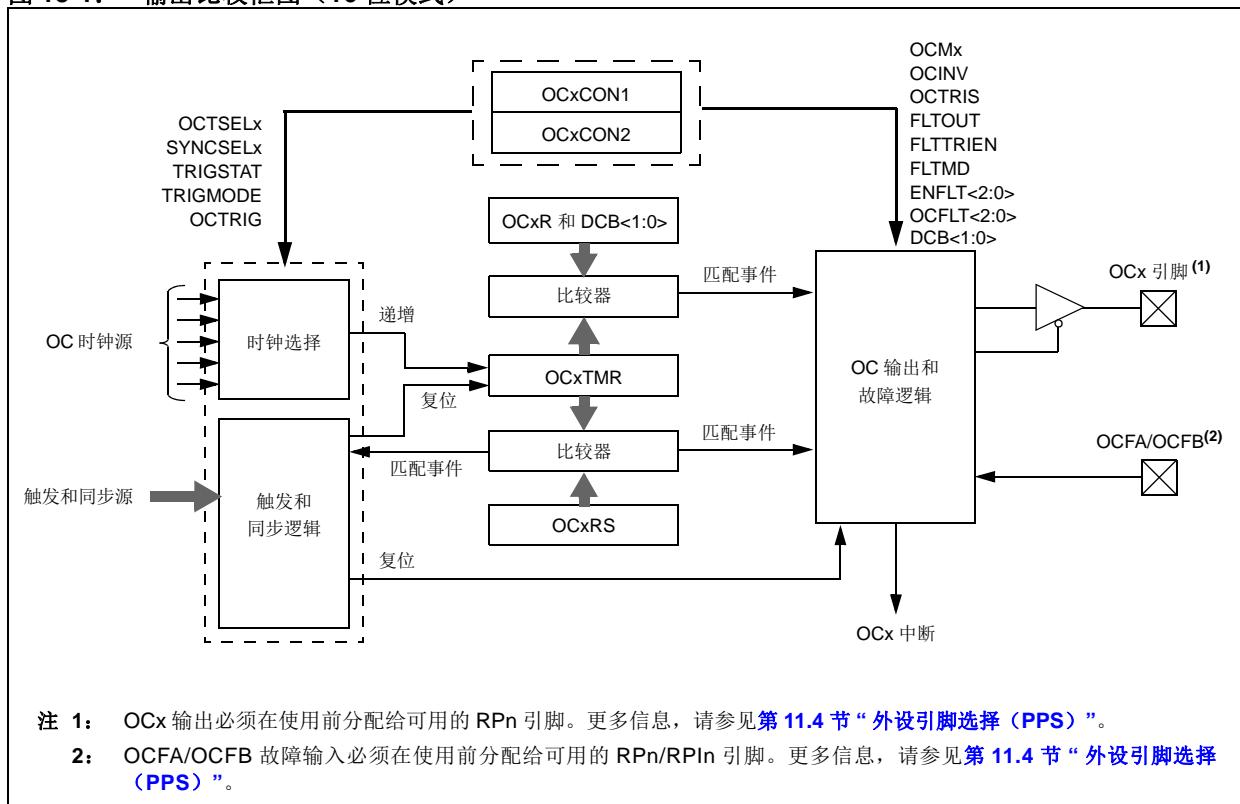
自由运行模式是默认选择的, 或者在 SYNCSEL 位 (OCxCON2<4:0>) 设置为 00000 时也可以选择自由运行模式。当 SYNCSEL 位设置为 00000 之外的其他值时, 可选择同步或触发模式。OCTRIG 位 (OCxCON2<7>) 用于选择同步或触发模式; 该位置 1 表示选择触发工作模式。在两种模式下, SYNCSEL 位决定同步 / 触发源。

### 15.1.2 级联 (32 位) 模式

默认情况下, 每个模块使用自己的 16 位定时器和占空比寄存器独立运行。要提高分辨率, 可将相邻的偶数编号的模块和奇数编号的模块配置为一个 32 位模块。(例如, 模块 1 和 2 配对, 模块 3 和 4 配对等)。奇数编号的模块 (OCx) 提供 32 位寄存器对中的低 16 位, 而偶数编号的模块 (OCy) 提供高 16 位。OCx 寄存器的折回操作会导致其相应的 OCy 寄存器递增。

用硬件配置级联操作的方法是: 对两个模块的 OC32 位 (OCxCON2<8>) 都置 1。有关级联的更多详细信息, 请参见《PIC24F 系列参考手册》中的第 35 章“带专用定时器的输出比较”(DS39723A\_CN)。

图 15-1：输出比较框图（16 位模式）



## 15.2 比较操作

在比较模式下（图15-1），输出比较模块可配置为产生单脉冲信号或产生连续脉冲信号。它还可在发生每个定时器事件时重复翻转输出引脚。

要将模块设置为比较操作：

1. 将OCx输出配置为其中一个可用的外设引脚选择引脚。
2. 计算 OCxR 和（用于双比较模式）OCxRS 占空比寄存器所需的值：
  - a) 确定指令时钟周期。考虑定时器源的外部时钟频率（如果使用）和定时器预分频比设置。
  - b) 计算从定时器起始值（0000h）到输出脉冲的上升沿所需的时间。
  - c) 根据所需的脉冲宽度和到脉冲上升沿的时间计算出现脉冲下降沿的时间。

3. 将上升沿的值写入 OCxR，下降沿的值写入 OCxRS。
4. 将定时器周期寄存器 PRy 的值设置为大于或等于 OCxRS 中的值。
5. 设置 OCM<2:0> 位以选择合适的比较操作（= 0xx）。
6. 如需触发工作模式，将 OCTRIG 置 1 以使能触发模式。置 1 或清零 TRIGMODE 可配置触发工作模式，置 1 或清零 TRIGSTAT 可选择硬件触发或软件触发。如需同步模式，清零 OCTRIG。
7. 设置 SYNCSEL<4:0> 位可配置触发或同步源。如需定时器工作在自由运行模式下，请将 SYNCSEL 位设置为 00000（无同步/触发源）。
8. 使用 OCTSEL<2:0> 位选择时基源。如有必要，将使能比较时基进行计数的选定定时器的 TON 位置 1。使能了时基后，将立即启动同步工作模式；发生触发源事件后，启动触发工作模式。

对于 32 位级联操作，还包含以下必需步骤：

1. 将两个寄存器的 OC32 位（即 OC<sub>y</sub>CON2<8> 和 OC<sub>x</sub>CON2<8>）置 1。首先使能偶数编号的模块，以确保模块一起启动。
2. 清零偶数编号的模块的 OCTRIG 位 (OC<sub>y</sub>CON2<7>)，使其在同步模式下运行。
3. 为 OC<sub>y</sub> 配置所需的输出和故障设置。
4. 通过清零 OCTRIS 位强制 OC<sub>x</sub> 的输出引脚为输出状态。
5. 如需触发工作模式，请使用 OCTRIG (OC<sub>x</sub>CON2<7>)、TRIGMODE (OC<sub>x</sub>CON2<3>) 和 SYNCSEL (OC<sub>x</sub>CON1<4:0>) 位配置触发选项。
6. 首先为 OC<sub>y</sub> 配置所需的比较或 PWM 工作模式 (OCM<2:0>)，然后再为 OC<sub>x</sub> 配置。

根据选择的输出模式，模块保持 OC<sub>x</sub> 引脚处于默认状态并在 OC<sub>x</sub>R 中的值与定时器中的值匹配时强制翻转该引脚的状态。在双比较模式下，当定时器中的值与 OC<sub>x</sub>RS 中的值匹配时，OC<sub>x</sub> 被强制返回默认状态。在单比较模式下发生定时器与 OC<sub>x</sub>R 的匹配、以及在双比较模式下每次定时器与 OC<sub>x</sub>RS 发生匹配后，OC<sub>x</sub>IF 中断标志都会置 1。

单脉冲事件仅发生一次，但是可通过简单地重写 OC<sub>x</sub>CON1 寄存器的值使其重复发生。连续脉冲事件在终止之前可无限次地发生。

## 15.3 脉宽调制 (PWM) 模式

在 PWM 模式下，输出比较模块可配置为产生边沿对齐或中心对齐的脉冲波形。所有 PWM 操作都是双缓冲的（缓冲寄存器是模块内部的寄存器，没有映射到 SFR 空间）。

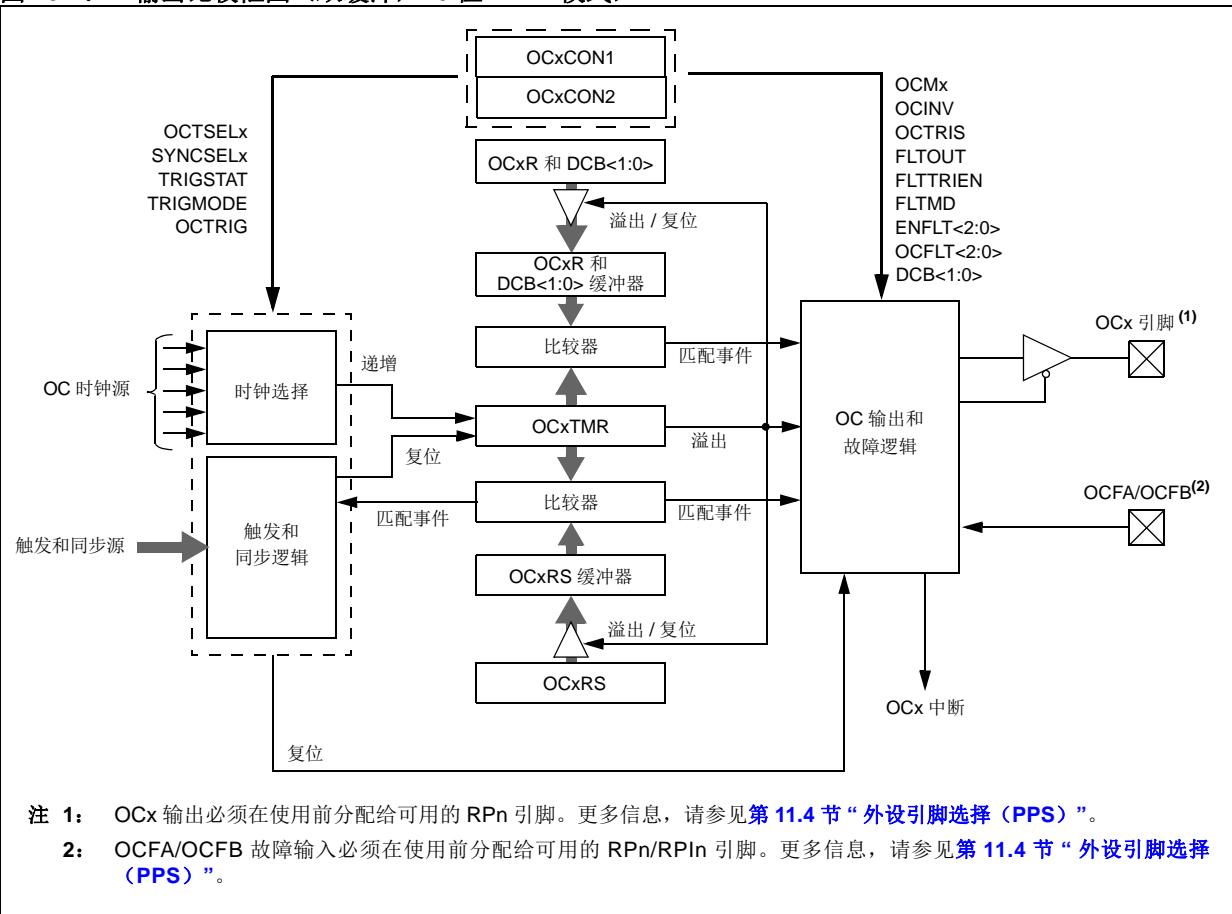
将输出比较模块配置为 PWM 工作模式：

1. 将 OC<sub>x</sub> 输出配置为其中一个可用的外设引脚选择引脚。
2. 计算所需的占空比并将计算结果装入 OC<sub>x</sub>R 寄存器。
3. 计算所需的周期并将计算结果装入 OC<sub>x</sub>RS 寄存器。
4. 选择当前的 OC<sub>x</sub> 作为同步源，方法是将 0x1F 写入 SYNCSEL<4:0> 位 (OC<sub>x</sub>CON2<4:0>)，将 0 写入 OCTRIG 位 (OC<sub>x</sub>CON2<7>)。
5. 通过写 OCTSEL<2:0> 位 (OC<sub>x</sub>CON<12:10>) 选择时钟源。
6. 如果需要的话，允许定时器和输出比较模块中断。使用 PWM 故障引脚时需要输出比较中断。
7. 通过 OCM<2:0> (OC<sub>x</sub>CON1<2:0>) 位选择所需的 PWM 模式。
8. 通过使用 ENFLT<2:0> 可以使能适当的故障输入，如 [寄存器 15-1](#) 中所述。
9. 如果定时器被选为时钟源，请设置选定定时器预分频值。选定定时器的预分频器输出将用作 OC<sub>x</sub> 定时器的时钟输入，而不是选定定时器的输出。

**注：** 本外设包含的输入和输出功能可能需要通过外设引脚选择功能进行配置。更多信息，请参见 [第 11.4 节“外设引脚选择 \(PPS\)”](#)。

# PIC24FJ128GA310 系列

图 15-2：输出比较框图（双缓冲，16 位 PWM 模式）



## 15.3.1 PWM 周期

PWM 周期可通过写入 PRy (定时器周期寄存器) 来指定。可使用公式 15-1 来计算 PWM 周期。

### 公式 15-1：计算 PWM 周期<sup>(1)</sup>

$$\text{PWM 周期} = [(PRy) + 1] \cdot T_{CY} \cdot (\text{定时器预分频值})$$

其中：

$$\text{PWM 频率} = 1/\text{[PWM 周期]}$$

注 1：基于  $T_{CY} = T_{OSC} * 2$ ；禁止打盹模式和 PLL。

注：若 PRy 的值为 N，则会使 PWM 周期为 N + 1 个时基计数周期。例如：写入 PRy 寄存器的值为 7 将产生由 8 个时基周期组成的 PWM 周期。

### 15.3.2 PWM 占空比

PWM 占空比是通过写 OCxRS 和 OCxR 寄存器指定的。在任何时间都可以写入 OCxRS 和 OCxR 寄存器，但是在 PRy 和 TMry 发生匹配（即周期完成）前不会锁存占空比值。这可以为 PWM 占空比提供双重缓冲，对于 PWM 的无毛刺操作是极其重要的。

以下是 PWM 占空比的部分重要边界参数：

- 如果 OCxR、OCxRS 和 PRy 装入 0000h，那么 OCx 引脚将保持低电平（占空比为 0%）。
- 如果 OCxRS 大于 PRy，则引脚将保持高电平（占空比为 100%）。

请参见例 15-1 了解 PWM 模式时序的详细信息。表 15-1 和表 15-2 给出了当器件分别工作在 4 MIPS 和 10 MIPS 时，所对应的 PWM 频率和分辨率的示例。

**公式 15-2：计算最大 PWM 分辨率<sup>(1)</sup>**

$$\text{最大 PWM 分辨率(位)} = \frac{\log_{10} \left( \frac{FCY}{FPWM \cdot (\text{定时器预分频值})} \right)}{\log_{10}(2)} \text{ 位}$$

**注 1：** 基于  $FCY = FOSC/2$ ；禁止打盹模式和 PLL。

**例 15-1：计算 PWM 周期和占空比<sup>(1)</sup>**

- 在所需的 PWM 频率为 52.08 kHz、Fosc = 8 MHz 并带 PLL（32 MHz 器件时钟速率）且 Timer2 预分频比为 1:1 的条件下计算定时器周期寄存器的值。

$$TCY = 2 * TOSC = 62.5 \text{ ns}$$

$$\text{PWM 周期} = 1/\text{PWM 频率} = 1/52.08 \text{ kHz} = 19.2 \text{ ms}$$

$$\text{PWM 周期} = (PR2 + 1) * TCY * (\text{Timer2 预分频值})$$

$$19.2 \text{ ms} = (PR2 + 1) * 62.5 \text{ ns} * 1$$

$$PR2 = 306$$

- 在 PWM 频率为 52.08 kHz 且器件的时钟速率为 32 MHz 时，计算占空比的最大分辨率：

$$\text{PWM 分辨率} = \log_{10}(FCY/FPWM)/\log_{10}(2) \text{ 位}$$

$$= (\log_{10}(16 \text{ MHz}/52.08 \text{ kHz})/\log_{10}(2)) \text{ 位}$$

$$= 8.3 \text{ 位}$$

**注 1：** 基于  $TCY = 2 * TOSC$ ，禁止打盹模式和 PLL。

**表 15-1：器件工作在 4 MIPS ( $FCY = 4 \text{ MHz}$ ) 时的 PWM 频率和分辨率示例<sup>(1)</sup>**

PWM 频率	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率(位)	16	16	15	12	10	7	5

**注 1：** 基于  $FCY = FOSC/2$ ；禁止打盹模式和 PLL。

**表 15-2：器件工作在 16 MIPS ( $FCY = 16 \text{ MHz}$ ) 时的 PWM 频率和分辨率示例<sup>(1)</sup>**

PWM 频率	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
定时器预分频比	8	1	1	1	1	1	1
周期寄存器值	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
分辨率(位)	16	16	15	12	10	7	5

**注 1：** 基于  $FCY = FOSC/2$ ；禁止打盹模式和 PLL。

# PIC24FJ128GA310 系列

寄存器 15-1： OCxCON1： 输出比较 x 控制寄存器 1

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	OCSIDL	OCTSEL2	OCTSEL1	OCTSEL0	ENFLT2 <sup>(2)</sup>	ENFLT1 <sup>(2)</sup>
bit 15							bit 8

R/W-0	R/W-0, HSC	R/W-0, HSC	R/W-0, HSC	R/W-0	R/W-0	R/W-0	R/W-0
ENFLT0 <sup>(2)</sup>	OCFLT2 <sup>(2,3)</sup>	OCFLT1 <sup>(2,4)</sup>	OCFLT0 <sup>(2,4)</sup>	TRIGMODE	OCM2 <sup>(1)</sup>	OCM1 <sup>(1)</sup>	OCM0 <sup>(1)</sup>
bit 7							bit 0

图注：

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 未实现：读为 0

bit 13 **OCSIDL**：在空闲模式下停止输出比较 x 控制位

1 = 输出比较 x 在 CPU 空闲模式下停止

0 = 输出比较 x 将在 CPU 空闲模式下继续工作

bit 12-10 **OCTSEL<2:0>**：输出比较 x 定时器选择位

111 = 外设时钟 (FcY)

110 = 保留

101 = 保留

100 = Timer1 时钟 (仅支持同步时钟)

011 = Timer5 时钟

010 = Timer4 时钟

001 = Timer3 时钟

000 = Timer2 时钟

bit 9 **ENFLT2**：故障输入 2 使能位 <sup>(2)</sup>

1 = 使能故障 2 (比较器 1/2/3 输出) <sup>(3)</sup>

0 = 禁止故障 2

bit 8 **ENFLT1**：故障输入 1 使能位 <sup>(2)</sup>

1 = 使能故障 1 (OCFB 引脚) <sup>(4)</sup>

0 = 禁止故障 1

bit 7 **ENFLT0**：故障输入 0 使能位 <sup>(2)</sup>

1 = 使能故障 0 (OCFA 引脚) <sup>(4)</sup>

0 = 禁止故障 0

bit 6 **OCFLT2**：PWM 故障 2 (比较器 1/2/3) 条件状态位 <sup>(2,3)</sup>

1 = 已发生 PWM 故障 2

0 = 未发生 PWM 故障 2

bit 5 **OCFLT1**：PWM 故障 1 (OCFB 引脚) 条件状态位 <sup>(2,4)</sup>

1 = 已发生 PWM 故障 1

0 = 未发生 PWM 故障 1

注 1： OCx 输出也必须配置给可用的 RPn 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS) ”。

2： 当 OCM<2:0> = 111 或 110 时，故障输入使能位和故障状态位有效。

3： 比较器 1 输出控制 OC1-OC3 通道；比较器 2 输出控制 OC4-OC6 通道；比较器 3 输出控制 OC7-OC9 通道。

4： OCFA/OCFB 故障输入也必须分配给可用的 RPn/RPIn 引脚。更多信息，请参见第 11.4 节“外设引脚选择 (PPS) ”。

## 寄存器 15-1： OCxCON1： 输出比较 x 控制寄存器 1（续）

bit 4	<b>OCFLTO:</b> PWM 故障 0 (OCFA 引脚) 条件状态位 <a href="#">(2,4)</a> 1 = 已发生 PWM 故障 0 0 = 未发生 PWM 故障 0
bit 3	<b>TRIGMODE:</b> 触发状态模式选择位 1 = OCxRS = OCxTMR 时清零 TRIGSTAT (OCxCON2<6>)，或由软件清零 TRIGSTAT 位 0 = TRIGSTAT 只能由软件清零
bit 2-0	<b>OCM&lt;2:0&gt;:</b> 输出比较 x 模式选择位 <a href="#">(1)</a> 111 = OCx 处于中心对齐 PWM 模式 <a href="#">(2)</a> 110 = OCx 处于边沿对齐 PWM 模式 <a href="#">(2)</a> 101 = 双比较连续脉冲模式：初始化 OCx 引脚为低电平，与 OCxR 和 OCxRS 交替匹配时连续翻转 OCx 的状态 100 = 双比较单脉冲模式：初始化 OCx 引脚为低电平；在一个周期内与 OCxR 和 OCxRS 匹配时翻转 OCx 的状态 011 = 单比较连续脉冲模式：比较事件连续翻转 OCx 引脚 010 = 单比较单脉冲模式：初始化 OCx 引脚为高电平，比较事件强制 OCx 引脚为低电平 010 = 单比较单脉冲模式：初始化 OCx 引脚为低电平；比较事件强制 OCx 引脚为高电平 000 = 禁止输出比较通道

**注 1：** OCx 输出也必须配置给可用的 RPn 引脚。更多信息，请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。

**2：** 当 OCM<2:0> = 111 或 110 时，故障输入使能位和故障状态位有效。

**3：** 比较器 1 输出控制 OC1-OC3 通道；比较器 2 输出控制 OC4-OC6 通道；比较器 3 输出控制 OC7-OC9 通道。

**4：** OCFA/OCFB 故障输入也必须分配给可用的 RPn/RPIn 引脚。更多信息，请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。

# PIC24FJ128GA310 系列

## 寄存器 15-2: OCxCON2: 输出比较 x 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
FLTMD	FLTOUT	FLTTRIEN	OCINV	—	DCB1 <sup>(3)</sup>	DCB0 <sup>(3)</sup>	OC32
bit 15	bit 8						

R/W-0	R/W-0 HS	R/W-0	R/W-0	R/W-1	R/W-1	R/W-0	R/W-0
OCTRIG	TRIGSTAT	OCTRIS	SYNCSEL4	SYNCSEL3	SYNCSEL2	SYNCSEL1	SYNCSEL0
bit 7	bit 0						

图注: HS = 可由硬件置 1 的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **FLTMD:** 故障模式选择位  
1 = 故障模式保持, 直到消除故障源并且相应的 OCFLTO 位由软件清零  
0 = 故障模式保持, 直到消除故障源并且启动了新的 PWM 周期

bit 14 **FLTOUT:** 故障输出位  
1 = 发生故障时 PWM 输出被驱动为高电平  
0 = 发生故障时 PWM 输出被驱动为低电平

bit 13 **FLTTRIEN:** 故障输出状态选择位  
1 = 发生故障条件时引脚被强制为输出状态  
0 = 引脚 I/O 状态不受故障影响

bit 12 **OCINV:** OCMP 反相位  
1 = OCx 输出反相  
0 = OCx 输出不反相

bit 11 未实现: 读为 0

bit 10-9 **DCB<11:0>:** PWM 占空比最低有效位<sup>(3)</sup>  
11 = OCx 下降沿延迟 3/4 指令周期  
10 = OCx 下降沿延迟 1/2 指令周期  
01 = OCx 下降沿延迟 1/4 指令周期  
00 = OCx 下降沿在指令周期一开始就出现

bit 8 **OC32:** 级联两个输出比较模块使能位 (32 位操作)  
1 = 使能级联模块操作  
0 = 禁止级联模块操作

bit 7 **OCTRIG:** OCx 触发 / 同步选择位  
1 = 用 SYNCSELx 位指定的源触发 OCx  
0 = OCx 与 SYNCSELx 位指定的源同步

bit 6 **TRIGSTAT:** 定时器触发状态位  
1 = 定时器源已被触发并运行  
0 = 定时器源未被触发, 保持清零状态

bit 5 **OCTRIS:** OCx 输出引脚方向选择位  
1 = OCx 引脚是三态引脚  
0 = OCx 引脚上连接了输出比较外设 x

注 1: 不要通过以下方式将输出比较模块作为自己的触发源: 选择这种模式或其他同等 SYNCSELx 设置。

2: 这些输入仅用作触发源, 不能用作同步源。

3: DCB<1:0> 位只有在 PWM 模式 (OCM<2:0> (OCxCON1<2:0>) = 111、110) 下才是双缓冲位。

## 寄存器 15-2: OCxCON2: 输出比较 x 控制寄存器 2 (续)

bit 4-0      **SYNCSEL<4:0>**: 触发 / 同步源选择位

11111 = 此输出比较模块 <sup>(1)</sup>

11110 = 输入捕捉 9<sup>(2)</sup>

11101 = 输入捕捉 6<sup>(2)</sup>

11100 = CTMU<sup>(2)</sup>

11011 = A/D<sup>(2)</sup>

11010 = 比较器 3<sup>(2)</sup>

11001 = 比较器 2<sup>(2)</sup>

11000 = 比较器 1<sup>(2)</sup>

10111 = 输入捕捉 4<sup>(2)</sup>

10110 = 输入捕捉 3<sup>(2)</sup>

10101 = 输入捕捉 2<sup>(2)</sup>

10100 = 输入捕捉 1<sup>(2)</sup>

10011 = 输入捕捉 8<sup>(2)</sup>

10010 = 输入捕捉 7<sup>(2)</sup>

1000x = 保留

01111 = Timer5

01110 = Timer4

01101 = Timer3

01100 = Timer2

01011 = Timer1

01010 = 输入捕捉 5<sup>(2)</sup>

01001 = 输出比较 9<sup>(1)</sup>

01000 = 输出比较 8<sup>(1)</sup>

00111 = 输出比较 7<sup>(1)</sup>

00110 = 输出比较 6<sup>(1)</sup>

00101 = 输出比较 5<sup>(1)</sup>

00100 = 输出比较 4<sup>(1)</sup>

00011 = 输出比较 3<sup>(1)</sup>

00010 = 输出比较 2<sup>(1)</sup>

00001 = 输出比较 1<sup>(1)</sup>

00000 = 不与任何其他模块同步

**注 1:** 不要通过以下方式将输出比较模块作为自己的触发源：选择这种模式或其他同等 SYNCSELx 设置。

**2:** 这些输入仅用作触发源，不能用作同步源。

**3:** DCB<1:0> 位只有在 PWM 模式 (OCM<2:0> (OCxCON1<2:0>) = 111、110) 下才是双缓冲位。

# **PIC24FJ128GA310 系列**

---

---

注:

## 16.0 串行外设接口（SPI）

**注:** 本数据手册总结了该组PIC24F器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息,请参见《PIC24F系列参考手册》中的第 23 章“串行外设接口（SPI）”(DS39699A\_CN)。本数据手册中的信息取代了FRM中的信息。

串行外设接口（Serial Peripheral Interface, SPI）模块是一个同步串行接口，可用于与其他外设或者单片机进行通信。这些外设器件可以是串行 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。PIC24FJ128GA310 系列的所有器件都包含两个 SPI 模块。

该模块可在两种缓冲模式下工作。在标准模式下，通过单个串行缓冲器移动数据。在增强型缓冲模式下，通过 8 级深 FIFO 缓冲器移动数据。

**注:** 无论是在标准还是增强型缓冲模式下，都不要对 SPIxBUF 寄存器执行读 - 修改 - 写操作（如位操作指令）。

工作在主模式或从模式下时，该模块还支持基本的帧 SPI 协议。共支持四种帧 SPI 配置。

SPI 串行接口由以下四个引脚组成：

- **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

SPI 模块可以被配置为使用 2 个、3 个或 4 个引脚工作。在 3 引脚模式下，不使用 SSx。在 2 引脚模式下，不使用 SDOx 和 SSx。

图 16-1 和图 16-2 所示为标准模式和增强型模式下该模块的框图。

**注:** 在本章中，SPI 模块统称为 SPIx，或分别称为 SPI1 和 SPI2。特殊功能寄存器也使用类似的符号表示。例如，SPIxCON1 和 SPIxCON2 表示这 2 个 SPI 模块中任何一个的控制寄存器。

# PIC24FJ128GA310 系列

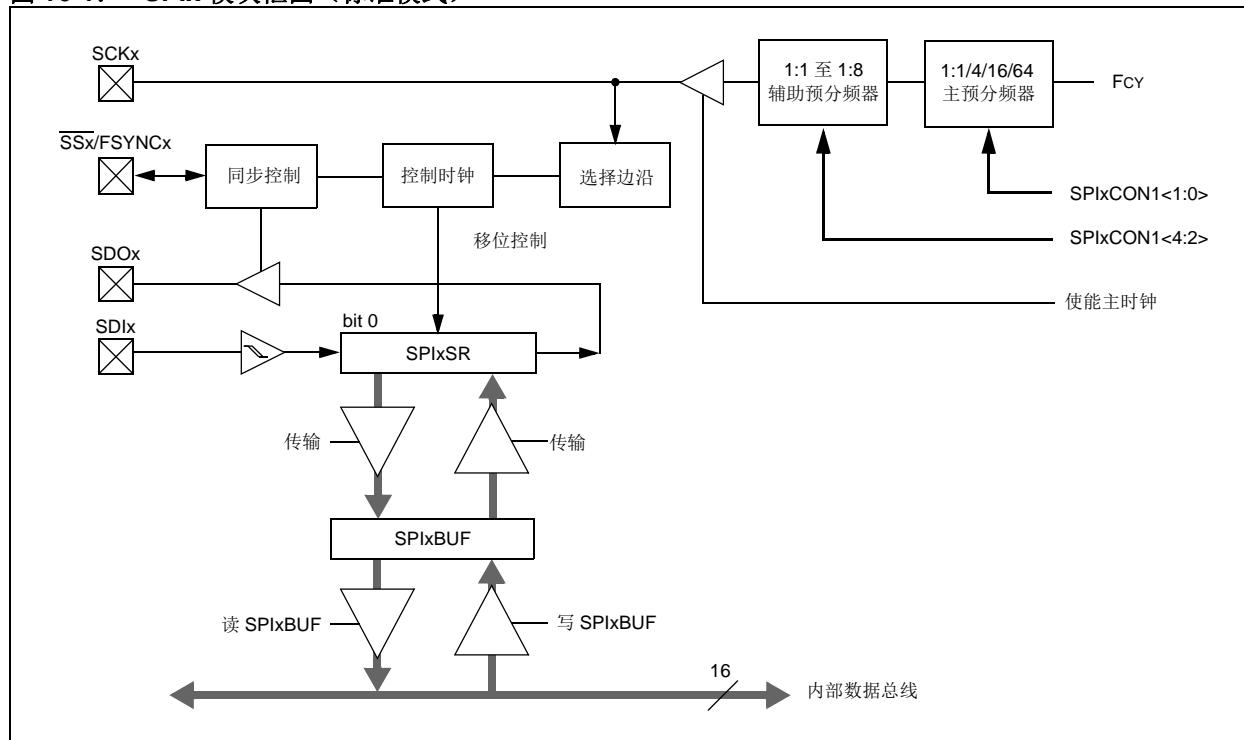
要将 SPI 模块设置为工作在标准主模式下，请遵循以下步骤：

1. 如果使用中断：
  - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
  - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
2. 将所需设置写入 SPIxCON1 和 SPIxCON2 寄存器，同时 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 操作。
5. 将待发送数据写入 SPIxBUF 寄存器。数据一写入 SPIxBUF 寄存器发送 (和接收) 就会立即开始。

要将 SPI 模块设置为工作在标准从模式下，请遵循以下步骤：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
  - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
  - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将所需设置写入 SPIxCON1 和 SPIxCON2 寄存器，同时 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位清零。
5. 如果将 CKE 位 (SPIxCON1<8>) 置 1，则 SSEN 位 (SPIxCON1<7>) 也必须置 1 以使能 SS<sub>x</sub> 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 操作。

图 16-1： SPIx 模块框图（标准模式）



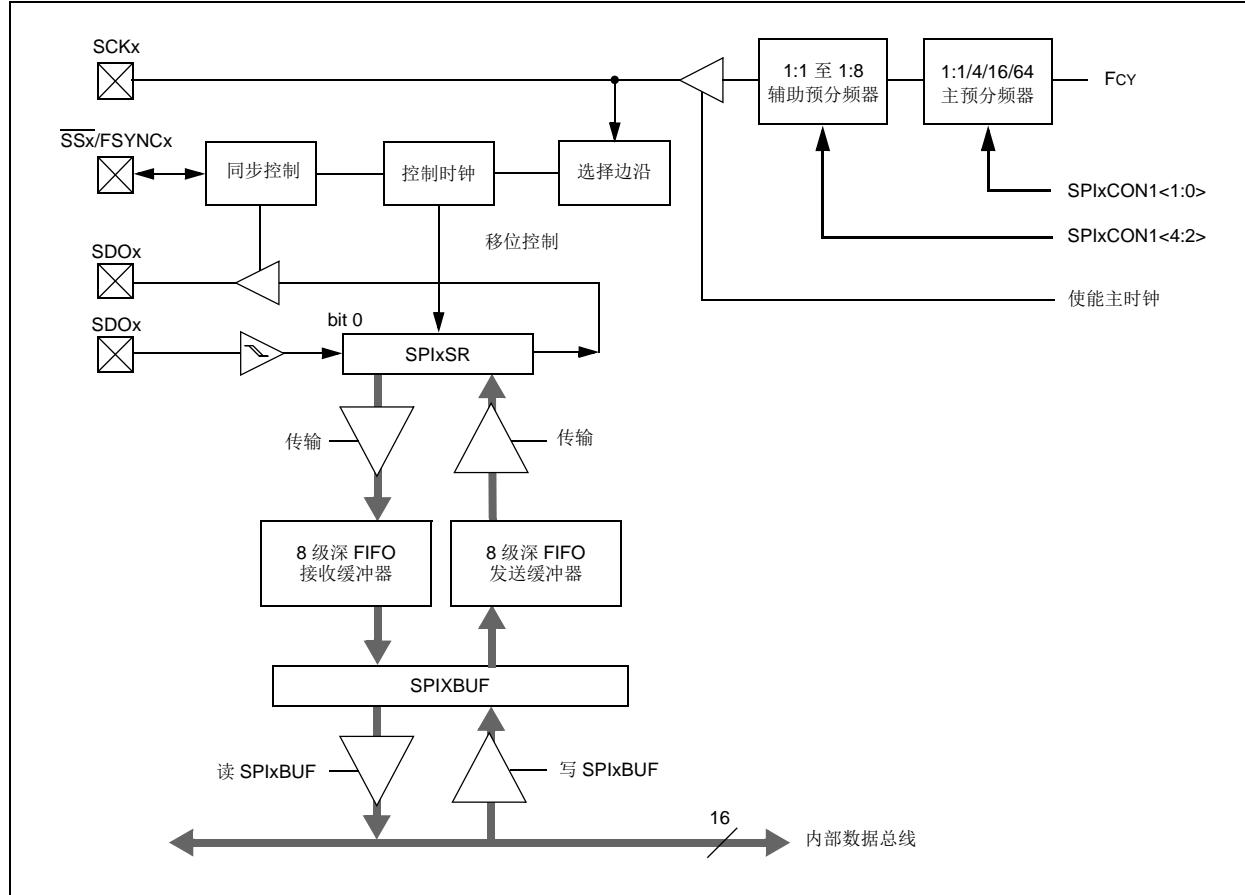
要将 SPI 模块设置为工作在增强型缓冲主模式下，请遵循以下步骤：

1. 如果使用中断：
  - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
  - c) 写相应 IPC 寄存器中的 SPIxIP 位。
2. 将所需设置写入 SPIxCON1 和 SPIxCON2 寄存器，同时 MSTEN (SPIxCON1<5>) = 1。
3. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
4. 通过将 SPIBEN 位 (SPIxCON2<0>) 置 1 选择增强型缓冲模式。
5. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 操作。
6. 将待发送数据写入 SPIxBUF 寄存器。数据一写入 SPIxBUF 寄存器发送（和接收）就会立即开始。

要将 SPI 模块设置为工作在增强型缓冲从模式下，请遵循以下步骤：

1. 将 SPIxBUF 寄存器清零。
2. 如果使用中断：
  - a) 将相应 IFS 寄存器中的 SPIxIF 位清零。
  - b) 将相应 IEC 寄存器中的 SPIxIE 位置 1。
  - c) 通过写相应 IPC 寄存器中的 SPIxIP 位来设置中断优先级。
3. 将所需设置写入 SPIxCON1 和 SPIxCON2 寄存器，同时 MSTEN (SPIxCON1<5>) = 0。
4. 将 SMP 位清零。
5. 如果 CKE 位置 1，则 SSEN 位也必须置 1 以使能 SS<sub>x</sub> 引脚。
6. 将 SPIROV 位 (SPIxSTAT<6>) 清零。
7. 通过将 SPIBEN 位 (SPIxCON2<0>) 置 1 选择增强型缓冲模式。
8. 通过将 SPIEN 位 (SPIxSTAT<15>) 置 1 使能 SPI 操作。

图 16-2： SPIx 模块框图（增强型模式）



# PIC24FJ128GA310 系列

## 寄存器 16-1: SPIxSTAT: SPIx 状态和控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
SPIEN <sup>(1)</sup>	—	SPISIDL	—	—	SPIBEC2	SPIBEC1	SPIBEC0
bit 15	bit 8						

R-0, HSC	R/C-0, HS	R-0, HSC	R/W-0	R/W-0	R/W-0	R-0, HSC	R-0, HSC
SRMPT	SPIROV	SRXMPT	SISEL2	SISEL1	SISEL0	SPITBF	SPIRBF
bit 7	bit 0						

图注:	C = 可清零位	HS = 可由硬件置 1 的位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
HSC = 可由硬件置 1/清零的位		x = 未知

bit 15	<b>SPIEN:</b> SPIx 使能位 <sup>(1)</sup>
	1 = 使能模块并将 SCKx、SDOx、SDIx 和 SSx 配置为串行端口引脚 0 = 禁止模块
bit 14	<b>未实现:</b> 读为 0
bit 13	<b>SPISIDL:</b> 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 12-11	<b>未实现:</b> 读为 0
bit 10-8	<b>SPIBEC&lt;2:0&gt;:</b> SPIx 缓冲器元素计数位 (在增强型缓冲模式下有效)
	<u>主模式:</u> 等待中的 SPI 传输数。 <u>从模式:</u> 未读取的 SPI 传输数。
bit 7	<b>SRMPT:</b> 移位寄存器 (SPIxSR) 空位 (在增强型缓冲模式下有效)
	1 = SPIx 移位寄存器为空并已做好发送或接收准备 0 = SPIx 移位寄存器不为空
bit 6	<b>SPIROV:</b> 接收溢出标志位
	1 = 一个新字节 / 字已被完全接收并丢弃。 用户软件尚未读取 SPIxBUF 寄存器中先前接收到的数据。 0 = 未发生溢出
bit 5	<b>SRXMPT:</b> 接收 FIFO 空位 (在增强型缓冲模式下有效)
	1 = 接收 FIFO 为空 0 = 接收 FIFO 不为空
bit 4-2	<b>SISEL&lt;2:0&gt;:</b> SPIx 缓冲器中断模式位 (在增强型缓冲模式下有效)
	111 = 当 SPIx 发送缓冲器满时产生中断 (SPITBF 位置 1) 110 = 最后一位移入 SPIxSR 导致发送 FIFO 为空时产生中断 101 = 最后一位被移出 SPIxSR 时产生中断, 此时发送完成 100 = 有数据移入 SPIxSR 导致发送 FIFO 有待发送数据时产生中断 011 = 当 SPIx 接收缓冲器满时产生中断 (SPIRBF 位置 1) 010 = 当 SPIx 接收缓冲器被填满 3/4 或更多时产生中断 001 = 接收缓冲器中有数据时产生中断 (SRMPT 位置 1) 000 = 读取接收缓冲器中的最后一个数据, 导致缓冲器为空时产生中断 (SRXMPT 位置 1)

**注 1:** 如果 SPIEN = 1, 这些功能在使用之前必须分配给可用的 RPn/RPln 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 寄存器 16-1: SPIxSTAT: SPIx 状态和控制寄存器 (续)

bit 1	<b>SPITB<sub>F</sub>:</b> SPIx 发送缓冲器满状态位 1 = 尚未开始发送, SPIxTXB 已满 0 = 已开始发送, SPIxTXB 为空 <u>在标准缓冲模式下:</u> 当 CPU 通过写 SPIxBUF 地址单元装载 SPIxTXB 时, 该位由硬件自动置 1。当 SPIx 模块将数据从 SPIxTXB 传输到 SPIxSR 时, 该位由硬件自动清零。 <u>在增强型缓冲模式下:</u> 当 CPU 通过写 SPIxBUF 地址单元装载最后一个可用缓冲单元时, 该位由硬件自动置 1。当有空的缓冲单元可由 CPU 写入时, 该位由硬件自动清零。
bit 0	<b>SPIRB<sub>F</sub>:</b> SPIx 接收缓冲器满状态位 1 = 接收完成, SPIxRXB 已满 0 = 接收未完成, SPIxRXB 为空 <u>在标准缓冲模式下:</u> 当 SPIx 将数据从 SPIxSR 传输到 SPIxRXB 时, 该位由硬件自动置 1。当内核通过读 SPIxBUF 地址单元读 SPIxRXB 时, 该位由硬件自动清零。 <u>在增强型缓冲模式下:</u> 当 SPIx 将数据从 SPIxSR 传输到缓冲器填充了最后一个未读的缓冲单元时, 该位由硬件自动置 1。当有空的缓冲单元可接收来自 SPIxSR 的传输数据时, 该位由硬件自动清零。

**注 1:** 如果 SPIEN = 1, 这些功能在使用之前必须分配给可用的 RPn/RPIn 引脚。更多信息, 请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。

# PIC24FJ128GA310 系列

寄存器 16-2: SPIxCON1: SPIx 控制寄存器 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DISSCK <sup>(1)</sup>	DISSDO <sup>(2)</sup>	MODE16	SMP	CKE <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
SSEN <sup>(4)</sup>	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-13 未实现: 读为 0
- bit 12 **DISSCK:** 禁止 SCKx 引脚位 (仅限 SPI 主模式) <sup>(1)</sup>  
1 = 禁止内部 SPI 时钟; 引脚用作 I/O  
0 = 使能内部 SPI 时钟
- bit 11 **DISSDO:** 禁止 SDOx 引脚位 <sup>(2)</sup>  
1 = 模块不使用 SDOx 引脚, 引脚用作 I/O  
0 = SDOx 引脚受模块控制
- bit 10 **MODE16:** 字 / 字节通信选择位  
1 = 通信为字宽 (16 位)  
0 = 通信为字节宽 (8 位)
- bit 9 **SMP:** SPIx 数据输入采样阶段位  
主模式:  
1 = 在数据输出时间的结尾采样输入数据  
0 = 在数据输出时间的中间采样输入数据  
从模式:  
当在从模式下使用 SPIx 时, 必须将 SMP 清零。
- bit 8 **CKE:** SPIx 时钟边沿选择位 <sup>(3)</sup>  
1 = 串行输出数据在时钟由有效状态变为空闲状态时改变 (见 bit 6)  
0 = 串行输出数据在时钟由空闲状态变为有效状态时改变 (见 bit 6)
- bit 7 **SSEN:** 从选择使能 (从模式) 位 <sup>(4)</sup>  
1 = SSx 引脚用于从模式  
0 = 模块不使用 SSx 引脚, 引脚由端口功能控制
- bit 6 **CKP:** 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 **MSTEN:** 主模式使能位  
1 = 主模式  
0 = 从模式

- 注 1:** 如果 DISSCK = 0, 必须将 SCKx 配置给可用的 RPn 引脚。更多信息, 请参见第 11.4 节 “外设引脚选择 (PPS) ”。
- 2:** 如果 DISSDO = 0, 必须将 SDOx 配置给可用的 RPn 引脚。更多信息, 请参见第 11.4 节 “外设引脚选择 (PPS) ”。
- 3:** 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 用户应将该位编程为 0。
- 4:** 如果 SSEN = 1, 必须将 SSx 配置给可用的 RPn/RPIn 引脚。更多信息, 请参见第 11.4 节 “外设引脚选择 (PPS) ”。

## 寄存器 16-2: SPIxCON1: SPIx 控制寄存器 1 (续)

bit 4-2      **SPRE<2:0>**: 辅助预分频比 (主模式) 位

111 = 辅助预分频比为 1:1

110 = 辅助预分频比为 2:1

.

.

.

000 = 辅助预分频比为 8:1

bit 1-0      **PPRE<1:0>**: 主预分频比 (主模式) 位

11 = 主预分频比为 1:1

10 = 主预分频比为 4:1

01 = 主预分频比为 16:1

00 = 主预分频比为 64:1

**注 1:** 如果 DISSCK = 0, 必须将 SCKx 配置给可用的 RPn 引脚。更多信息, 请参见[第 11.4 节 “外设引脚选择 \(PPS\)”](#)。

**2:** 如果 DISSDO = 0, 必须将 SDOx 配置给可用的 RPn 引脚。更多信息, 请参见[第 11.4 节 “外设引脚选择 \(PPS\)”](#)。

**3:** 在帧 SPI 模式下不使用 CKE 位。在帧 SPI 模式 (FRMEN = 1) 下, 用户应将该位编程为 0。

**4:** 如果 SSEN = 1, 必须将 SS<sub>x</sub> 配置给可用的 RPn/RPI<sub>n</sub> 引脚。更多信息, 请参见[第 11.4 节 “外设引脚选择 \(PPS\)”](#)。

# PIC24FJ128GA310 系列

寄存器 16-3: SPIxCON2: SPIx 控制寄存器 2

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
FRMEN	SPIFSD	SPIFPOL	—	—	—	—	—
bit 15	bit 8						

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	SPIFE	SPIBEN
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **FRMEN:** 帧 SPIx 支持位

1 = 使能帧 SPIx 支持

0 = 禁止帧 SPIx 支持

bit 14      **SPIFSD:**  $\overline{SS_x}$  引脚上的帧同步脉冲方向控制位

1 = 帧同步脉冲输入 (从模式)

0 = 帧同步脉冲输出 (主模式)

bit 13      **SPIFPOL:** 帧同步脉冲极性位 (仅用于帧模式)

1 = 帧同步脉冲高电平有效

0 = 帧同步脉冲低电平有效

bit 12-2    未实现: 读为 0

bit 1        **SPIFE:** 帧同步脉冲边沿选择位

1 = 帧同步脉冲与第一个位时钟同步

0 = 帧同步脉冲比第一个位时钟超前

bit 0        **SPIBEN:** 增强型缓冲器使能位

1 = 使能增强型缓冲器

0 = 禁止增强型缓冲器 (传统模式)

图 16-3： SPI 主 / 从连接（标准模式）

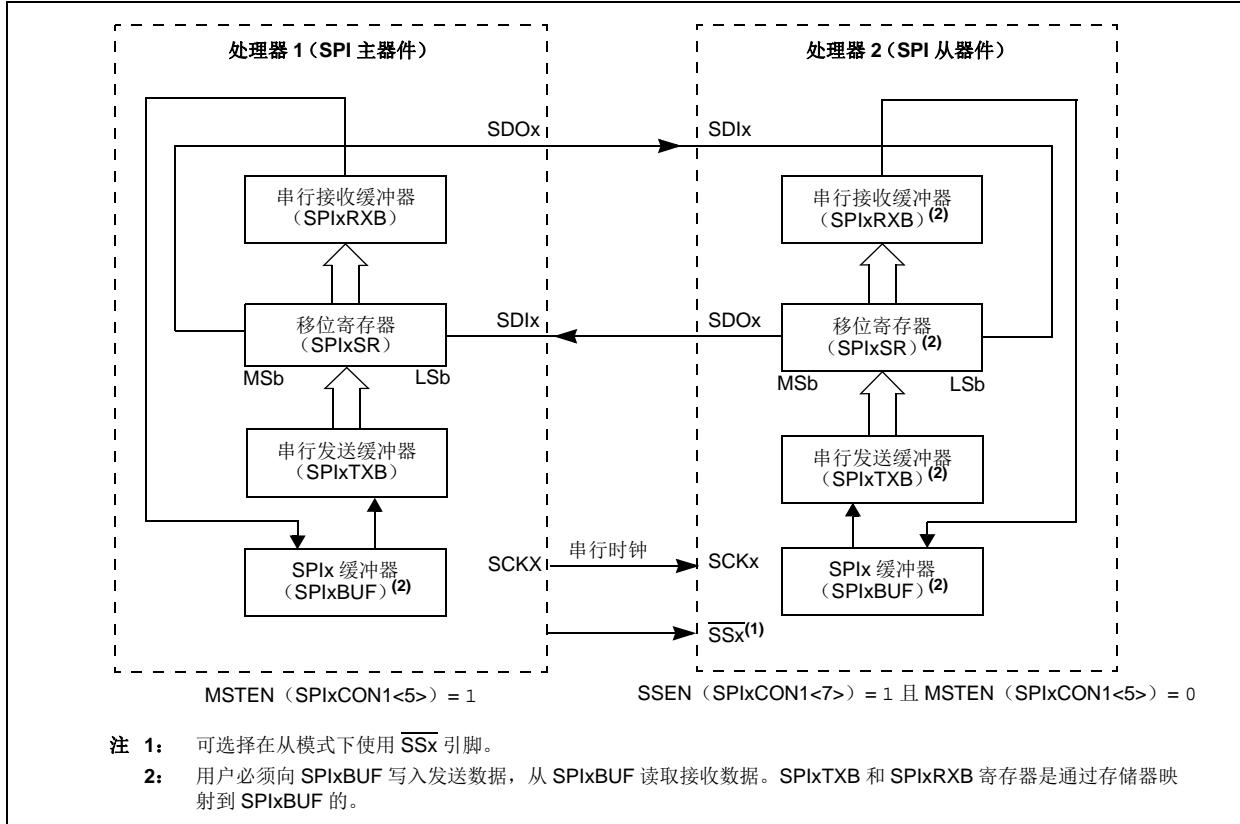
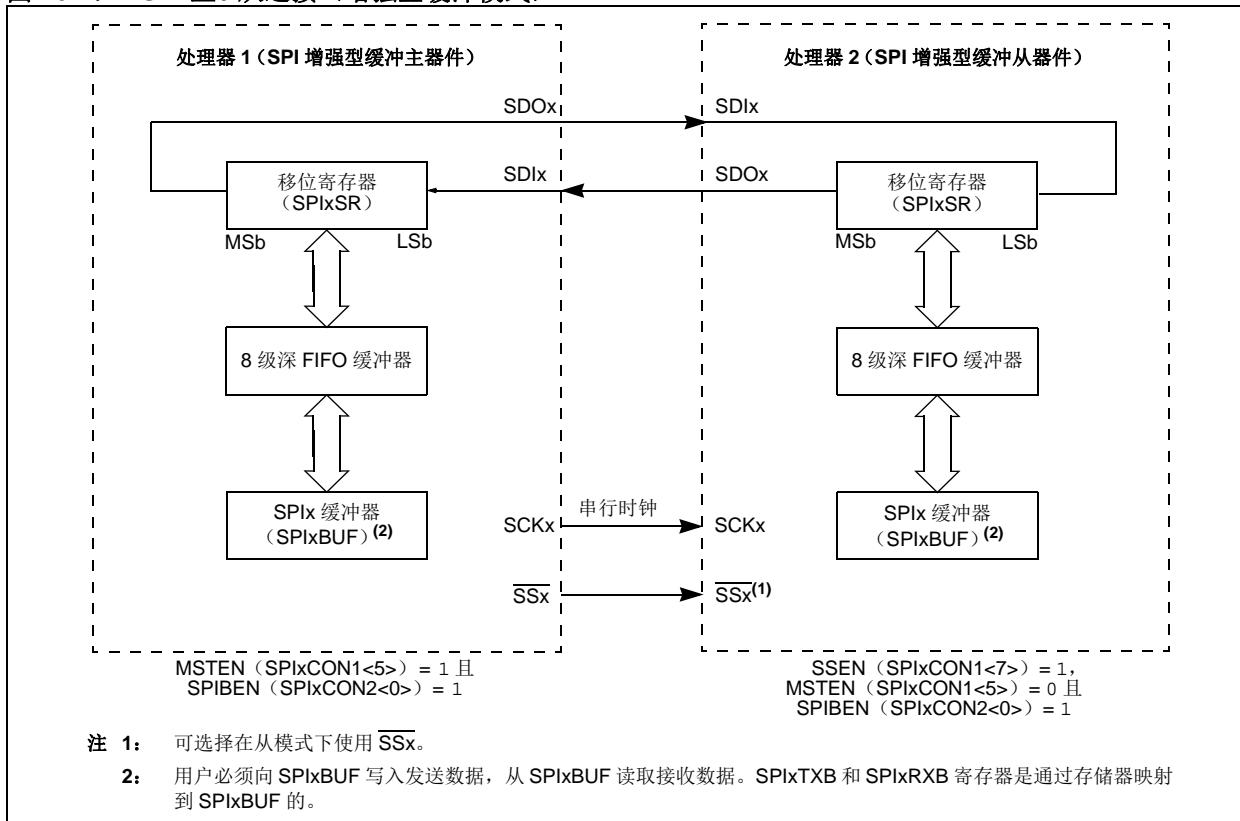


图 16-4： SPI 主 / 从连接（增强型缓冲模式）



# PIC24FJ128GA310 系列

图 16-5: SPI 主器件、帧主器件连接图

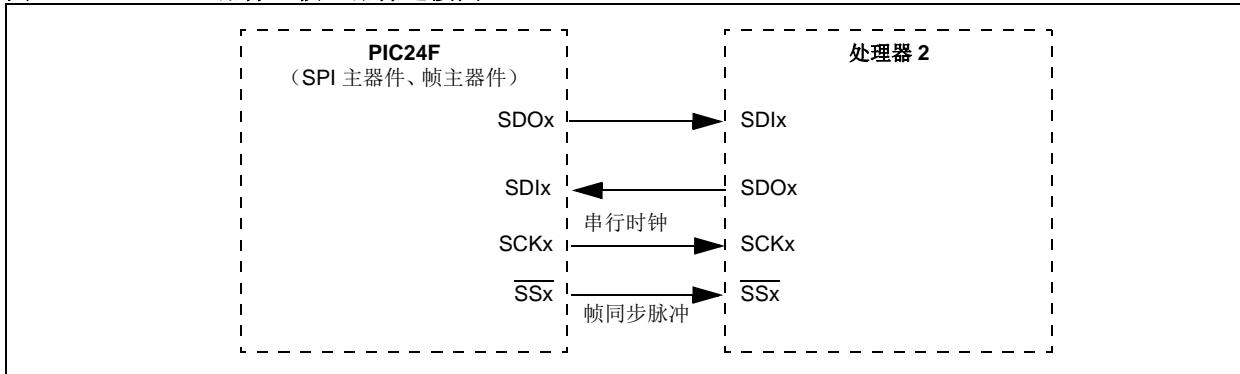


图 16-6: SPI 主器件、帧从器件连接图

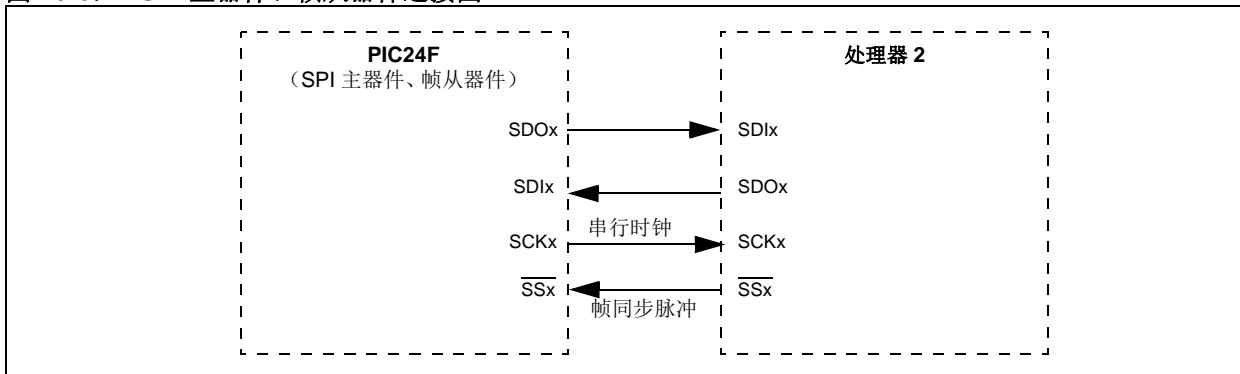


图 16-7: SPI 从器件、帧主器件连接图

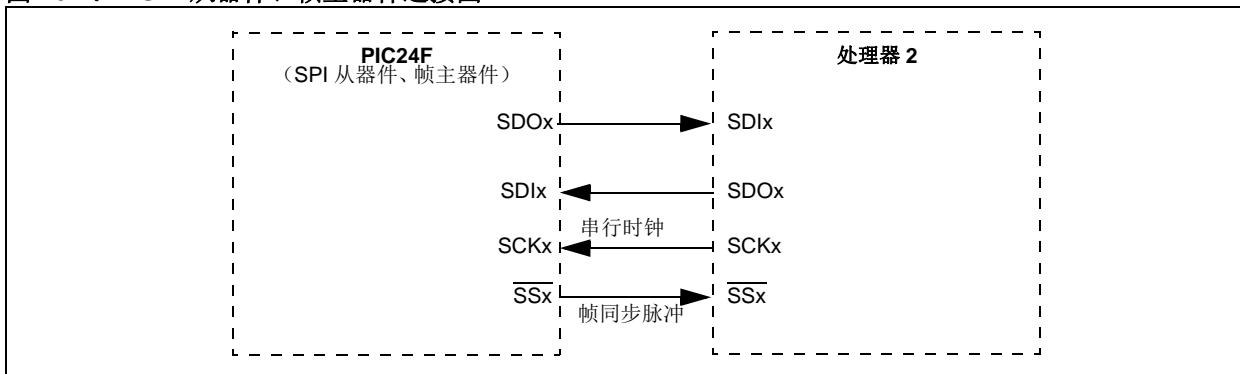
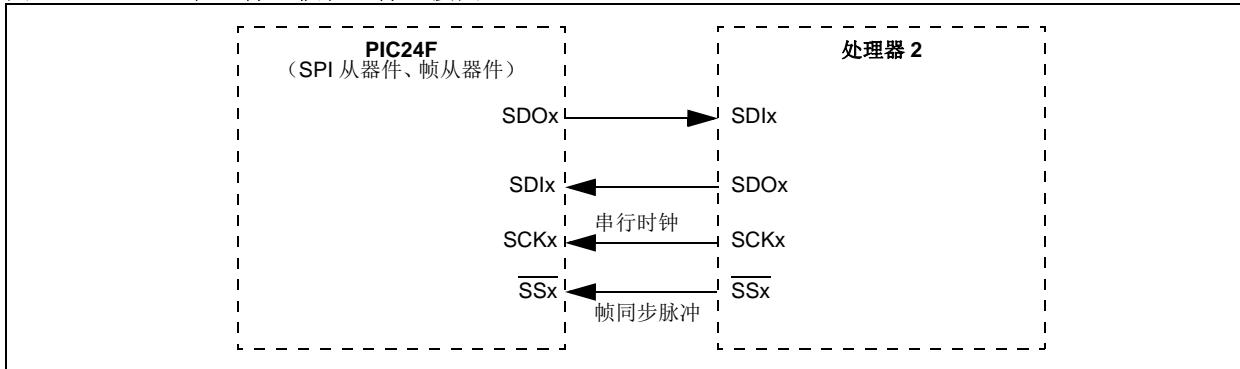


图 16-8: SPI 从器件、帧从器件连接图



公式 16-1： 器件速率和 SPI 时钟速率之间的关系<sup>(1)</sup>

$$FSCK = \frac{FCY}{\text{主预分频值} * \text{辅助预分频值}}$$

注 1： 基于  $FCY = Fosc/2$ ；禁止打盹模式和 PLL。

表 16-1： 采样 SCKx 频率<sup>(1,2)</sup>

FCY = 16 MHz		辅助预分频比设置				
		1:1	2:1	4:1	6:1	8:1
主预分频比设置	1:1	无效	8000	4000	2667	2000
	4:1	4000	2000	1000	667	500
	16:1	1000	500	250	167	125
	64:1	250	125	63	42	31
FCY = 5 MHz						
主预分频比设置	1:1	5000	2500	1250	833	625
	4:1	1250	625	313	208	156
	16:1	313	156	78	52	39
	64:1	78	39	20	13	10

注 1： 基于  $FCY = Fosc/2$ ；禁止打盹模式和 PLL。

2： SCKx 频率的单位为 kHz。

# **PIC24FJ128GA310 系列**

---

---

注:

## 17.0 I<sup>2</sup>C<sup>TM</sup>

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 24 章“**I<sup>2</sup>C<sup>TM</sup>**”(DS39702A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

I<sup>2</sup>C<sup>TM</sup> 模块是用来与其他外设或单片机器件通信的串行接口。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:

- 独立的主 / 从逻辑
- 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, 可为处理器提供响应从器件数据请求的延时
- 支持 100 kHz 和 400 kHz 两种总线规范
- 可配置的地址掩码
- 多主器件模式以防止仲裁时报文丢失
- 总线重发器模式, 允许作为从器件接收来自所有地址的所有报文
- 自动 SCL

图 17-1 给出了该模块的框图。

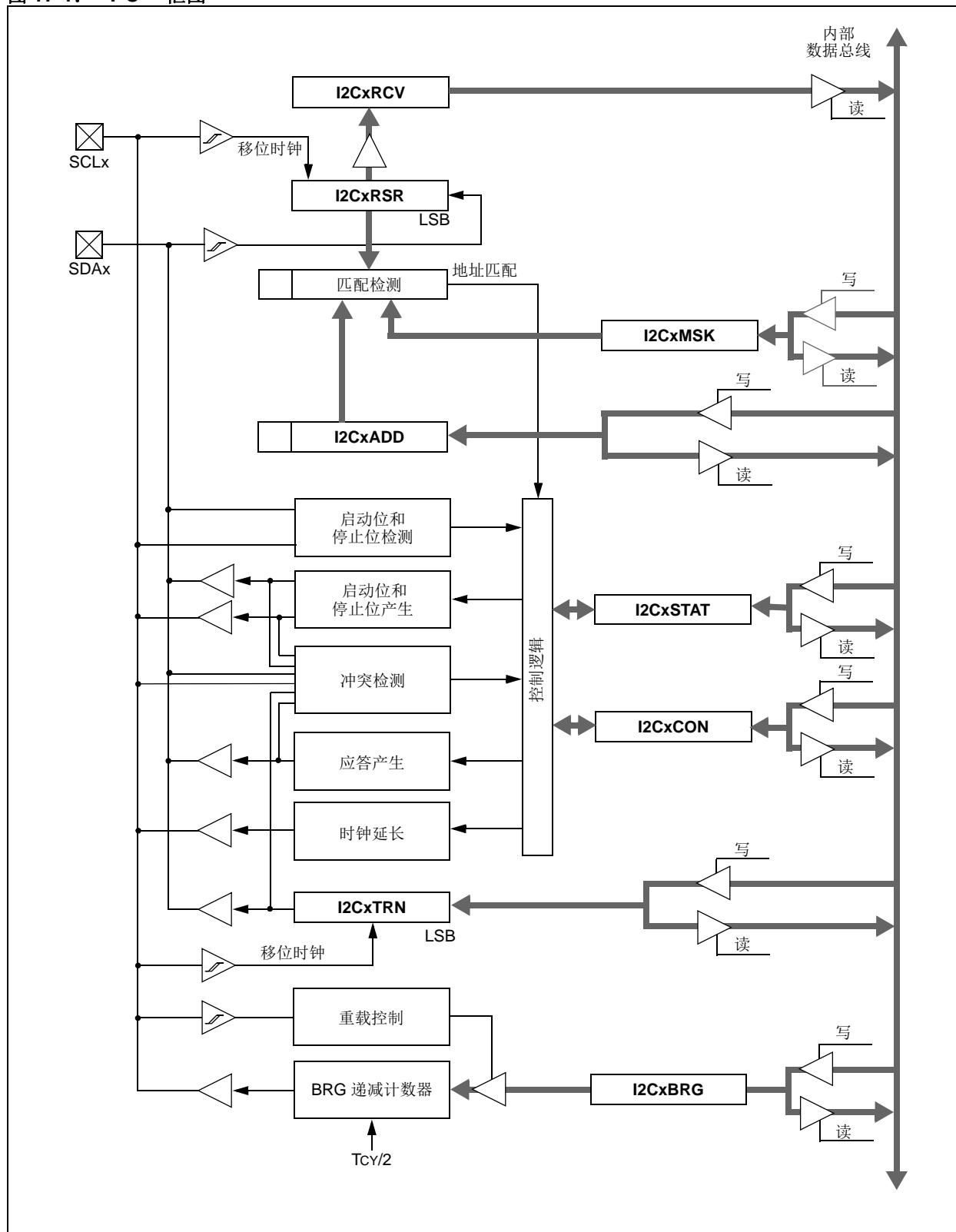
### 17.1 作为主器件在单主器件环境中通信

在主模式下发送报文的细节取决于要进行通信的器件的通信协议。通常, 事件序列如下:

1. 在 SDAx 和 SCLx 上发送一个启动条件。
2. 发送一个 I<sup>2</sup>C 器件地址字节到从器件, 表明要执行写操作。
3. 等待并验证从器件的应答。
4. 发送第一个数据字节 (有时是命令) 到从器件。
5. 等待并验证从器件的应答。
6. 发送串行存储器地址低字节到从器件。
7. 重复步骤 4 和 5, 直至发送完所有数据字节。
8. 在 SDAx 和 SCLx 上发送一个重复启动条件。
9. 发送一个器件地址字节到从器件, 表明要执行读操作。
10. 等待并验证从器件的应答。
11. 使能主器件接收以接收串行存储器数据。
12. 在数据字节接收完毕时, 产生 ACK 或 NACK 条件。
13. 在 SDAx 和 SCLx 上产生一个停止条件。

# PIC24FJ128GA310 系列

图 17-1: I<sup>2</sup>C™ 框图



## 17.2 设置用作总线主器件时的波特率

使用 [公式 17-1](#) 计算波特率发生器的重载值。

**公式 17-1：计算波特率重载值 (1,2)**

$$FSCL = \frac{FCY}{I2CxBRG + 1 + \frac{FCY}{10,000,000}}$$

或:

$$I2CxBRG = \left( \frac{FCY}{FSCL} - \frac{FCY}{10,000,000} - 1 \right)$$

- 注 1:** 基于  $FCY = FOSC/2$ ; 禁止打盹模式和 PLL。  
**2:** 这些时钟速率值仅供参考。实际的时钟速率与各个系统级参数有关。应在其专用的应用中测量实际的时钟速率。

## 17.3 从地址掩码

I2CxMSK 寄存器（见 [寄存器 17-3](#)）指定了 7 位和 10 位寻址模式下某些“无关”地址位的位置。将 I2CxMSK 寄存器中某个特定位置 1 (=1)，不论相应的地址位是 0 还是 1，从模块都会做出响应。例如，当将 I2CxMSK 设置为 00100000 时，从模块将检测两个地址 0000000 和 0100000。

为了使能地址掩码，必须通过将 IPMIEN 位 (I2CxCON <11>) 清零来禁止智能外设管理接口 (Intelligent Peripheral Management Interface, IPMI)。

**注:** 为了符合 I<sup>2</sup>C<sup>TM</sup> 协议所作的更改，表 17-2 中保留了一些地址，工作在从模式下的 I<sup>2</sup>C 模块不会应答此类地址。这同样适用于包含任意这些地址在内的所有地址掩码设置。

**表 17-1：I<sup>2</sup>C<sup>TM</sup> 时钟速率 (1,2)**

所需的系统 FSCL	FCY	I2CxBRG 值		实际 FSCL
		(十进制)	(十六进制)	
100 kHz	16 MHz	157	9D	100 kHz
100 kHz	8 MHz	78	4E	100 kHz
100 kHz	4 MHz	39	27	99 kHz
400 kHz	16 MHz	37	25	404 kHz
400 kHz	8 MHz	18	12	404 kHz
400 kHz	4 MHz	9	9	385 kHz
400 kHz	2 MHz	4	4	385 kHz
1 MHz	16 MHz	13	D	1.026 MHz
1 MHz	8 MHz	6	6	1.026 MHz
1 MHz	4 MHz	3	3	0.909 MHz

**注 1:** 基于  $FCY = FOSC/2$ ; 禁止打盹模式和 PLL。

**2:** 这些时钟速率值仅供参考。实际的时钟速率与各个系统级参数有关。应在其目标应用中测量实际的时钟速率。

**表 17-2：I<sup>2</sup>C<sup>TM</sup> 保留地址 (1)**

从地址	R/W 位	说明
0000 000	0	广播呼叫地址 (2)
0000 000	1	启动字节
0000 001	x	CBus 地址
0000 01x	x	保留
0000 1xx	x	HS 模式主代码
1111 0xx	x	10 位从地址高字节 (3)
1111 1xx	x	保留

**注 1:** 上述地址位永远不会导致地址匹配，无论地址掩码的设置如何。

**2:** 仅当 GCEN = 1 时，才会应答地址。

**3:** 仅在 10 位寻址模式下才会与地址的高字节匹配。

# PIC24FJ128GA310 系列

## 寄存器 17-1：I2CxCON：I2Cx 控制寄存器

R/W-0	U-0	R/W-0	R/W-1, HC	R/W-0	R/W-0	R/W-0	R/W-0
I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0, HC				
GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN
bit 7	bit 0						

图注：	HC = 可由硬件清零的位
R = 可读位	W = 可写位
-n = 上电复位时的值	1 = 置 1 0 = 清零 x = 未知

- bit 15      **I2CEN:** I2Cx 使能位  
1 = 使能 I2Cx 模块并将 SDA<sub>x</sub> 和 SCL<sub>x</sub> 引脚配置为串行端口引脚  
0 = 禁止 I2Cx 模块；所有 I<sup>2</sup>C<sup>TM</sup> 引脚由端口功能控制
- bit 14      未实现：读为 0
- bit 13      **I2CSIDL:** 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 = 模块在空闲模式下继续工作
- bit 12      **SCLREL:** SCL<sub>x</sub> 释放控制位（作为 I<sup>2</sup>C 从器件工作时）  
1 = 释放 SCL<sub>x</sub> 时钟  
0 = 保持 SCL<sub>x</sub> 时钟为低电平（时钟延长）  
如果 STREN = 1:  
该位可读可写（即软件可写入 0 来启动时钟延长或写入 1 来释放时钟）。在从器件发送开始或接收结束时由硬件清零。  
如果 STREN = 0:  
该位可读且可被置 1（即软件只能写入 1 来释放时钟）。在从器件发送开始时由硬件清零。
- bit 11      **IPMIEN:** 智能外设管理接口（IPMI）使能位  
1 = 使能 IPMI 支持模式；应答所有地址  
0 = 禁止 IPMI 模式
- bit 10      **A10M:** 10 位从器件寻址位  
1 = I2CxADD 是一个 10 位从地址  
0 = I2CxADD 是一个 7 位从地址
- bit 9      **DISSLW:** 禁止压摆率控制位  
1 = 禁止压摆率控制  
0 = 使能压摆率控制
- bit 8      **SMEN:** SMBus 输入电平位  
1 = 使能符合 SMBus 规范的 I/O 引脚门限值  
0 = 禁止 SMBus 输入门限值
- bit 7      **GCEN:** 广播呼叫使能位（作为 I<sup>2</sup>C 从器件工作时）  
1 = 允许在 I2CxRSR 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      **STREN:** SCL<sub>x</sub> 时钟延长使能位（作为 I<sup>2</sup>C 从器件工作时）  
与 SCLREL 位配合使用。  
1 = 使能软件或接收时钟延长  
0 = 禁止软件或接收时钟延长

## 寄存器 17-1: I2CxCON: I<sup>2</sup>Cx 控制寄存器 (续)

bit 5	<b>ACKDT:</b> 应答数据位 (作为 I <sup>2</sup> C 主器件工作时; 适用于主器件接收操作) 当软件启动应答序列时将发送的值。 1 = 在应答时发送 NACK 0 = 在应答时发送 ACK
bit 4	<b>ACKEN:</b> 应答序列使能位 (作为 I <sup>2</sup> C 主器件工作时; 适用于主器件接收操作) 1 = 在 SDA <sub>x</sub> 和 SCL <sub>x</sub> 引脚上启动应答序列，并发送 ACKDT 数据位。在主器件应答序列结束时由硬件清零。 0 = 没有启动应答序列
bit 3	<b>RCEN:</b> 接收使能位 (作为 I <sup>2</sup> C 主器件工作时) 1 = 使能 I <sup>2</sup> C 接收模式。在主器件接收完数据字节的 8 位后由硬件清零。 0 = 没有启动接收序列
bit 2	<b>PEN:</b> 停止条件使能位 (作为 I <sup>2</sup> C 主器件工作时) 1 = 在 SDA <sub>x</sub> 和 SCL <sub>x</sub> 引脚上发出停止条件。在主器件停止序列结束时由硬件清零。 0 = 没有发起停止条件
bit 1	<b>RSEN:</b> 重复启动条件使能位 (作为 I <sup>2</sup> C 主器件工作时) 1 = 在 SDA <sub>x</sub> 和 SCL <sub>x</sub> 引脚上发出重复启动条件。在主器件重复启动序列结束时由硬件清零。 0 = 没有发起重复启动条件
bit 0	<b>SEN:</b> 启动条件使能位 (作为 I <sup>2</sup> C 主器件工作时) 1 = 在 SDA <sub>x</sub> 和 SCL <sub>x</sub> 引脚上发出启动条件。在主器件启动序列结束时由硬件清零。 0 = 没有发起启动条件

# PIC24FJ128GA310 系列

## 寄存器 17-2: I2CxSTAT: I2Cx 状态寄存器

R-0, HSC	R-0, HSC	U-0	U-0	U-0	R/C-0, HS	R-0, HSC	R-0, HSC
ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10
bit 15							bit 8

R/C-0, HS	R/C-0, HS	R-0, HSC	R/C-0, HSC	R/C-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IWCOL	I2COV	D/A	P	S	R/W	RBF	TBF
bit 7							bit 0

图注:	C = 可清零位	HS = 可由硬件置 1 的位
R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零
HSC = 可由硬件置 1/ 清零的位		x = 未知

- bit 15      **ACKSTAT:** 应答状态位  
1 = 上次检测到 NACK  
0 = 上次检测到 ACK  
在应答结束时由硬件置 1 或清零。
- bit 14      **TRSTAT:** 发送状态位  
(作为 I<sup>2</sup>C<sup>TM</sup> 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  
在主器件发送开始时由硬件置 1。在从器件应答结束时由硬件清零。
- bit 13-11    未实现: 读为 0
- bit 10       **BCL:** 主器件总线冲突检测位  
1 = 主器件工作期间检测到了总线冲突  
0 = 无冲突  
检测到总线冲突时由硬件置 1。
- bit 9         **GCSTAT:** 广播呼叫状态位  
1 = 接收到广播呼叫地址  
0 = 未接收到广播呼叫地址  
当地址与广播呼叫地址匹配时由硬件置 1。当检测到停止条件时由硬件清零。
- bit 8         **ADD10:** 10 位地址状态位  
1 = 10 位地址匹配  
0 = 10 位地址不匹配  
当与匹配的 10 位地址的第 2 个字节匹配时由硬件置 1; 当检测到停止条件时由硬件清零。
- bit 7         **IWCOL:** 写冲突检测位  
1 = 因为 I<sup>2</sup>C 模块忙, 尝试写 I2CxTRN 寄存器的操作失败。  
0 = 无冲突  
当 I<sup>2</sup>C 模块忙时写 I2CxTRN 会使硬件将该位置 1 (由软件清零)。
- bit 6         **I2COV:** 接收溢出标志位  
1 = 当 I2CxRCV 寄存器仍然保存原先的字节时接收到了新字节  
0 = 无溢出  
尝试将数据从 I2CxRSR 传输到 I2CxRCV 时由硬件置 1 (由软件清零)。
- bit 5         **D/A:** 数据 / 地址位 (作为 I<sup>2</sup>C 从器件工作时)  
1 = 表示上次接收的字节为数据  
0 = 表示上次接收的字节为器件地址  
器件地址匹配时由硬件清零。在发送完成后或接收从器件字节时由硬件置 1。

## 寄存器 17-2: I2CxSTAT: I2Cx 状态寄存器 (续)

bit 4	<b>P:</b> 停止位 1 = 表示上次检测到停止位 0 = 表示上次未检测到停止位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
bit 3	<b>S:</b> 启动位 1 = 表示上次检测到启动位 (或重复启动位) 0 = 表示上次未检测到启动位 当检测到启动、重复启动或停止条件时由硬件置 1 或清零。
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 接收到 I <sup>2</sup> C 器件地址字节后由硬件置 1 或清零。
bit 1	<b>RBF:</b> 接收缓冲器满状态位 1 = 接收完成, I2CxRCV 为满 0 = 接收未完成, I2CxRCV 为空 用接收到的字节写 I2CxRCV 时由硬件置 1。当用软件读 I2CxRCV 时由硬件清零。
bit 0	<b>TBF:</b> 发送缓冲器满状态位 1 = 发送正在进行, I2CxTRN 为满 0 = 发送完成, I2CxTRN 为空 用软件写 I2CxTRN 时由硬件置 1。数据发送完成时由硬件清零。

## 寄存器 17-3: 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
—	—	—	—	—	—	AMSK9	AMSK8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| AMSK7 | AMSK6 | AMSK5 | AMSK4 | AMSK3 | AMSK2 | AMSK1 | AMSK0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

### 图注:

R = 可读位	W = 可写位	U = 未实现位, 读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零

bit 15-10	未实现: 读为 0
bit 9-0	<b>AMSK&lt;9:0&gt;:</b> 地址位 x 的掩码选择位 1 = 使能输入报文地址 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

# **PIC24FJ128GA310 系列**

---

---

注:

## 18.0 通用异步收发器 (UART)

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 21 章“UART”(DS39708A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

通用异步收发器 (Universal Synchronous Receiver Transmitter, UART) 模块是 PIC24F 系列器件提供的串行 I/O 模块之一。UART 是可以和外设 (例如个人电脑、LIN/J2602、RS-232 和 RS-485 接口) 通信的全双工异步系统。UART 模块还通过 UxCTS 和 UxRTS 引脚支持硬件流控制, 该模块中还包含 IrDA® 编码器和解码器。

UART 模块的主要特性有:

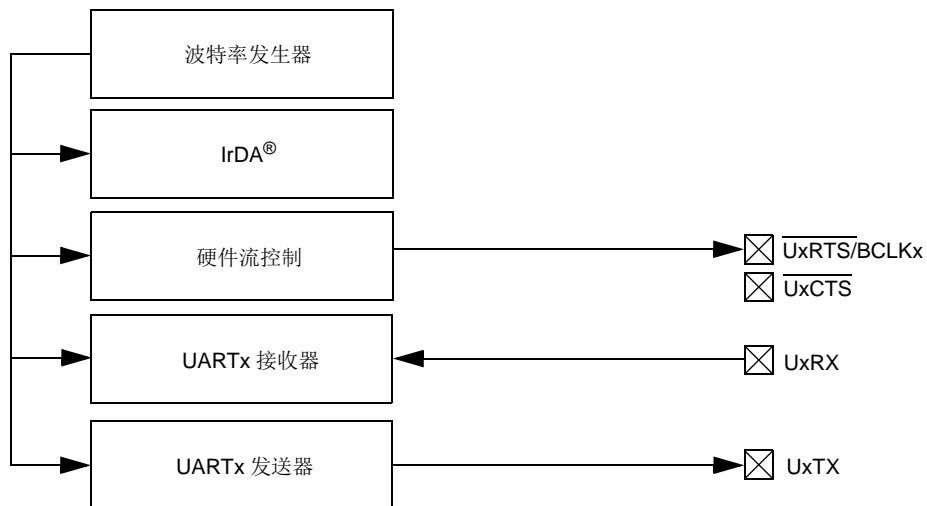
- 通过 UxTX 和 UxRX 引脚进行全双工 8 位或 9 位数据传输
- 偶校验、奇校验或无奇偶校验选项 (对于 8 位数据)
- 一个或两个停止位
- 通过 UxCTS 和 UxRTS 引脚实现硬件流控制

- 完全集成的具有 16 位预分频器的波特率发生器
- 当器件工作在 16 MIPS 时, 波特率范围从 15 bps 到 1 Mbps
- 4 级深先进先出 (First-In-First-Out, FIFO) 发送数据缓冲器
- 4 级深 FIFO 接收数据缓冲器
- 奇偶校验、帧和缓冲器溢出错误检测
- 支持带地址检测的 9 位模式 (第 9 位 = 1)
- 发送和接收中断
- 支持诊断的环回模式
- 支持同步和间隔字符
- 支持自动波特率检测
- IrDA® 编码器和解码器逻辑
- 支持 IrDA 的 16 倍频波特率时钟输出

图 18-1 显示了 UART 的简化框图。UART 模块由以下重要硬件组成:

- 波特率发生器
- 异步发送器
- 异步接收器

图 18-1: UART 简化框图



**注:** UART 输入和输出必须在使用之前分配给可用的 RPn/RPin 引脚。更多信息, 请参见第 11.4 节“外设引脚选择 (PPS)”。

## 18.1 UART 波特率发生器 (BRG)

UART 模块包含一个专用的 16 位波特率发生器。UxBRG 寄存器控制一个独立运行的 16 位定时器的周期。[公式 18-1](#) 给出了 BRGH = 0 时计算波特率的公式。

### 公式 18-1: BRGH = 0 时的 UART 波特率 [\(1,2\)](#)

$$\text{波特率} = \frac{\text{FCY}}{16 \cdot (\text{UxBRG} + 1)}$$
$$\text{UxBRG} = \frac{\text{FCY}}{16 \cdot \text{波特率}} - 1$$

**注 1:** FCY 表示指令周期的时钟频率 (Fosc/2)。

**2:** 基于  $\text{FCY} = \text{Fosc}/2$ ; 禁止打盹模式和 PLL。

[例 18-1](#) 给出了下列条件下的波特率误差计算:

- $\text{FCY} = 4 \text{ MHz}$
- 目标波特率 = 9600

可能的最大波特率 ( $\text{BRGH} = 0$ ) 是  $\text{FCY}/16$  (当  $\text{UxBRG} = 0$  时), 可能的最小波特率是  $\text{FCY}/(16 * 65536)$ 。

[公式 18-2](#) 给出了  $\text{BRGH} = 1$  时计算波特率的公式。

### 公式 18-2: BRGH = 1 时的 UART 波特率 [\(1,2\)](#)

$$\text{波特率} = \frac{\text{FCY}}{4 \cdot (\text{UxBRG} + 1)}$$
$$\text{UxBRG} = \frac{\text{FCY}}{4 \cdot \text{波特率}} - 1$$

**注 1:** FCY 表示指令周期时钟频率。

**2:** 基于  $\text{FCY} = \text{Fosc}/2$ ; 禁止打盹模式和 PLL。

可能的最大波特率 ( $\text{BRGH} = 1$ ) 是  $\text{FCY}/4$  (当  $\text{UxBRG} = 0$  时), 可能的最小波特率是  $\text{FCY}/(4 * 65536)$ 。

向 UxBRG 寄存器中写入新值会导致 BRG 定时器复位 (清零)。这保证了 BRG 在产生新的波特率之前不需要等待定时器溢出。

### 例 18-1: 波特率误差计算 (BRGH = 0) [\(1\)](#)

$$\text{目标波特率} = \text{FCY}/(16 (\text{BRGx} + 1))$$

BRGx 值的计算方法:

$$\begin{aligned}\text{BRGx} &= ((\text{FCY} / \text{目标波特率})/16) - 1 \\ \text{BRGx} &= ((4000000/9600)/16) - 1 \\ \text{BRGx} &= 25\end{aligned}$$

$$\begin{aligned}\text{计算的波特率} &= 4000000/(16 (25 + 1)) \\ &= 9615\end{aligned}$$

$$\begin{aligned}\text{误差} &= (\text{计算的波特率} - \text{目标波特率})/\text{目标波特率} \\ &= (9615 - 9600)/9600 \\ &= 0.16\%\end{aligned}$$

**注 1:** 基于  $\text{FCY} = \text{Fosc}/2$ ; 禁止打盹模式和 PLL。

## 18.2 8位数据模式下的发送

1. 设置 UART:
  - a) 将适当的值写入数据位、奇偶校验位和停止位。
  - b) 将适当的波特率值写入 UxBRG 寄存器。
  - c) 设置发送和接收中断允许位和优先级位。
2. 使能 UART。
3. 将 UTXEN 位置 1（置 1 后两个周期内产生发送中断）。
4. 将数据字节写入 UxTXREG 字的低字节。该数据字节将被立即传送给发送移位寄存器 (TSR) 且在波特率时钟的下一个上升沿开始移出串行比特流。
5. 另外，数据字节也可在 UTXEN = 0 时被传送，且随后用户可将 UTXEN 置 1。由于波特率时钟是从清零状态启动的，所以该行为能立即开始发送串行比特流。
6. 将根据中断控制位 UTXISELx 的设置产生发送中断。

## 18.3 9位数据模式下的发送

1. 设置 UART（如第 18.2 节“8位数据模式下的发送”所述）。
2. 使能 UART。
3. 将 UTXEN 位置 1（产生发送中断）。
4. 只将 16 位值写入 UxTXREG。
5. 向 UxTXREG 写入一个字可触发 9 位数据向 TSR 传送。串行比特流将会在波特率时钟的第一个上升沿开始移出。
6. 将根据中断控制位 UTXISELx 的设置产生发送中断。

## 18.4 间隔和同步发送序列

下面的序列将发送一个由间隔字符和其后的自动波特率同步字节组成的报文帧头。

1. 将 UART 配置为所需的模式。
2. 将 UTXEN 和 UTXBRK 置 1 以设置间隔字符。
3. 将一个“虚拟”字符装入 UxTXREG 寄存器中以启动发送（该值会被忽略）。
4. 向 UxTXREG 写入 “55h”，这会将同步字符装入发送 FIFO 中。
5. 当间隔字符发送完成后，由硬件将 UTXBRK 位复位。然后开始发送同步字符。

## 18.5 8位或9位数据模式下的接收

1. 设置 UART（如第 18.2 节“8位数据模式下的发送”所述）。
2. 使能 UART。
3. 当接收到一个或多个数据字符时，将会根据中断控制位 URXISELx 的设置产生接收中断。
4. 读 OERR 位以确定是否发生了溢出错误。必须用软件将 OERR 位复位。
5. 读 UxRXREG。

读取 UxRXREG 字符的行为会将下一个字符传送到接收 FIFO 的顶部，其中包含一组新的 PERR 和 FERR 值。

## 18.6 UxCTS 和 UxRTS 控制引脚的操作

UARTx 允许发送 (UxCTS) 和 UARTx 请求发送 (UxRTS) 是两个与 UART 模块有关由硬件控制的引脚。这两个引脚允许 UART 运行在单工模式和流控制模式下。这两个引脚控制数据终端设备 (Data Terminal Equipment, DTE) 之间的数据发送和接收。通过 UxMODE 寄存器的 UEN<1:0> 位来配置这两个引脚。

## 18.7 红外支持

UART 模块提供两种类型的红外 UART 支持：一种是 IrDA 时钟输出，支持外部 IrDA 编码器和解码器（支持传统模块）；另一种是完全实现的 IrDA 编码器和解码器。注意，因为 IrDA 模式需要 16 倍频的波特率时钟，所以它们只能工作在 BRGH 位 (UxMODE<3>) 为 0 时。

### 18.7.1 支持外部 IrDA 的 IrDA 时钟输出

为了支持外部 IrDA 编码器和解码器，可将 BCLKx 引脚（与 UxRTS 引脚相同）配置为产生 16 倍频的波特率时钟。当使能了 UART 模块且 UEN<1:0> = 11 时，BCLKx 引脚将输出 16 倍频的波特率时钟，用于支持 IrDA 编解码器芯片。

### 18.7.2 内置 IrDA 编码器和解码器

UART 模块在其内部完全实现了 IrDA 编码器和解码器。内置 IrDA 编码器和解码器的功能可通过 IREN 位 (UxMODE<12>) 来使能。当使能（即 IREN = 1）时，接收引脚 (UxRX) 可作为红外接收器的输入引脚。发送引脚 (UxTX) 可作为红外发送器的输出引脚。

# PIC24FJ128GA310 系列

寄存器 18-1：UxMODE：UARTx 模式寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
UARTEN <sup>(1)</sup>	—	USIDL	IREN <sup>(2)</sup>	RTSMD	—	UEN1	UEN0
bit 15							bit 8

R/W-0, HC	R/W-0	R/W-0, HC	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	RXINV	BRGH	PDSEL1	PDSEL0	STSEL
bit 7							bit 0

图注：	HC = 可由硬件清零的位
R = 可读位	W = 可写位
-n = 上电复位时的值	1 = 置 1
	0 = 清零
	x = 未知

bit 15	<b>UARTEN:</b> UARTx 使能位 <sup>(1)</sup>
	1 = 使能 UARTx； UARTx 根据 UEN<1:0> 的定义控制所有 UARTx 引脚 0 = 禁止 UARTx； 由端口锁存器控制所有 UARTx 引脚； UARTx 的功耗最小
bit 14	<b>未实现:</b> 读为 0
bit 13	<b>USIDL:</b> 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>IREN:</b> IrDA® 编码器和解码器使能位 <sup>(2)</sup> 1 = 使能 IrDA 编码器和解码器 0 = 禁止 IrDA 编码器和解码器
bit 11	<b>RTSMD:</b> <u>UxRTS</u> 引脚模式选择位 1 = <u>UxRTS</u> 引脚处于单工模式 0 = <u>UxRTS</u> 引脚处于流控制模式
bit 10	<b>未实现:</b> 读为 0
bit 9-8	<b>UEN&lt;1:0&gt;:</b> UARTx 使能位 11 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 和 <u>BCLKx</u> 引脚； <u>UxCTS</u> 引脚由端口锁存器控制 10 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 、 <u>UxCTS</u> 和 <u>UxRTS</u> 引脚 01 = 使能并使用 <u>UxTX</u> 、 <u>UxRX</u> 和 <u>UxRTS</u> 引脚； <u>UxCTS</u> 引脚由端口锁存器控制 00 = 使能并使用 <u>UxTX</u> 和 <u>UxRX</u> 引脚； <u>UxCTS</u> 和 <u>UxRTS/BCLKx</u> 引脚由端口锁存器控制
bit 7	<b>WAKE:</b> 在休眠模式下检测到启动位时唤醒的使能位 1 = UARTx 将继续采样 <u>UxRX</u> 引脚； 在出现下降沿时产生中断； 在随后的上升沿该位由硬件清零 0 = 未使能唤醒
bit 6	<b>LPBACK:</b> UARTx 环回模式选择位 1 = 使能环回模式 0 = 禁止环回模式
bit 5	<b>ABAUD:</b> 自动波特率使能位 1 = 使能在下一个字符传输过程中对波特率进行测量——需要接收到同步字段 (55h)； 完成时由硬件清零 0 = 禁止或已完成波特率测量

**注 1:** 如果 UARTEN = 1，外设的输入和输出必须配置给可用的 RPin/RPin 引脚。更多信息，请参见第 11.4 节“[外设引脚选择（PPS）](#)”。

**2:** 此功能仅适用于 16 倍 BRG 模式 (BRGH = 0)。

## 寄存器 18-1: UxMODE: UARTx 模式寄存器 (续)

bit 4	<b>RXINV:</b> 接收极性翻转位 1 = UxRX 的空闲状态为 0 0 = UxRX 的空闲状态为 1
bit 3	<b>BRGH:</b> 高波特率使能位 1 = 高速模式 (每个位周期内产生 4 个 BRG 时钟信号) 0 = 标准模式 (每个位周期内产生 16 个 BRG 时钟信号)
bit 2-1	<b>PDSEL&lt;1:0&gt;:</b> 奇偶校验和数据选择位 11 = 9 位数据, 无奇偶校验 10 = 8 位数据, 奇校验 01 = 8 位数据, 偶校验 00 = 8 位数据, 无奇偶校验
bit 0	<b>STSEL:</b> 停止位选择位 1 = 2 个停止位 0 = 1 个停止位

**注 1:** 如果 UARTEN = 1, 外设的输入和输出必须配置给可用的 RPn/RPI<sub>n</sub> 引脚。更多信息, 请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。

**2:** 此功能仅适用于 16 倍 BRG 模式 (BRGH = 0)。

# PIC24FJ128GA310 系列

## 寄存器 18-2: UxSTA: UARTx 状态和控制寄存器

R/W-0	R/W-0	R/W-0	U-0	R/W-0 HC	R/W-0	R-0, HSC	R-1, HSC
UTXISEL1	UTXINV <sup>(1)</sup>	UTXISEL0	—	UTXBRK	UTXEN <sup>(2)</sup>	UTXBF	TRMT <sup>(3)</sup>
bit 15							

R/W-0	R/W-0	R/W-0	R-1, HSC	R-0, HSC	R-0, HSC	R/C-0, HS	R-0, HSC
URXISEL1	URXISEL0	ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							

图注:

C = 可清零位

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

HS = 可由硬件置 1 的位

HC = 可由硬件清零的位

bit 15,13 **UTXSEL<1:0>**: 发送中断模式选择位

11 = 保留; 不要使用

10 = 当一个字符被传送到发送移位寄存器 (TSR) 导致发送缓冲器为空时, 产生中断

01 = 当最后一个字符被移出发送移位寄存器时产生中断; 发送操作全部完成

00 = 当一个字符被传送到发送移位寄存器 (前提是发送缓冲器中至少还有一个字符可供发送) 时产生中断

bit 14 **UTXINV**: IrDA® 编码器发送极性翻转位<sup>(1)</sup>

IREN = 0:

1 = UxTX 空闲状态为 0

0 = UxTX 空闲状态为 1

IREN = 1:

1 = UxTX 空闲状态为 1

0 = UxTX 空闲状态为 0

bit 12 未实现: 读为 0

bit 11 **UTXBRK**: 发送间隔位

1 = 在下次发送时发出同步间隔字符 —— 起始位, 后跟 12 个 0 位, 然后是停止位; 完成时由硬件清零

0 = 禁止或已完成同步间隔字符的发送

bit 10 **UTXEN**: 发送使能位<sup>(2)</sup>

1 = 使能发送, UARTx 控制 UxTX 引脚

0 = 禁止发送, 中止所有等待的发送, 缓冲器复位; 端口控制 UxTX 引脚

bit 9 **UTXBF**: 发送缓冲器满状态位 (只读)

1 = 发送缓冲器满

0 = 发送缓冲器未满, 至少还可写入一个字符

bit 8 **TRMT**: 发送移位寄存器空位 (只读)<sup>(3)</sup>

1 = 发送移位寄存器为空, 同时发送缓冲器为空 (上一次发送已完成)

0 = 发送移位寄存器非空, 发送正在进行或等待发送

**注 1:** 仅当使能了 IrDA® 编码器 (IREN = 1) 时, 该位的值才影响模块的发送属性。

**2:** 如果 UARTEM = 1, 外设的输入和输出必须配置给可用的 RPn/RPIn 引脚。更多信息, 请参见第 11.4 节“[外设引脚选择 \(PPS\)](#)”。

**3:** 一旦装入 UTXREG, 只有在两个指令周期后, TRMT 才会有效。

## 寄存器 18-2: UxSTA: UARTx 状态和控制寄存器 (续)

bit 7-6	<b>URXISEL&lt;1:0&gt;</b> : 接收中断模式选择位 11 = 当 RSR 传输使接收缓冲器为满时 (即, 有 4 个数据字符), 中断标志位置 1 10 = 当 RSR 传输使接收缓冲器为 3/4 满时 (即, 有 3 个数据字符), 中断标志位置 1 0x = 当接收缓冲器接收到来自 RSR 的任何字符时, 中断标志位置 1。接收缓冲器有一个或多个字符
bit 5	<b>ADDEN</b> : 地址字符检测位 (接收数据的第 8 位 = 1) 1 = 使能地址检测模式 (如果没有选择 9 位模式, 这个控制位将无效) 0 = 禁止地址检测模式
bit 4	<b>RIDLE</b> : 接收器空闲位 (只读) 1 = 接收器空闲 0 = 接收器工作
bit 3	<b>PERR</b> : 奇偶校验错误状态位 (只读) 1 = 检测到当前字符 (在接收 FIFO 顶部的字符) 的奇偶校验错误 0 = 没有检测到奇偶校验错误
bit 2	<b>FERR</b> : 帧错误状态位 (只读) 1 = 检测到当前字符 (在接收 FIFO 顶部的字符) 的帧错误 0 = 没有检测到帧错误
bit 1	<b>OERR</b> : 接收缓冲器溢出错误状态位 (只读 / 清零) 1 = 接收缓冲器已经溢出 0 = 接收缓冲器没有溢出。清零先前置 1 的 OERR 位 ( $1 \rightarrow 0$ 的跳变) 会使接收缓冲器和 RSR 复位为空状态
bit 0	<b>URXDA</b> : 接收缓冲器中数据可用位 (只读) 1 = 接收缓冲器中有数据, 有至少一个字符可被读取 0 = 接收缓冲器为空

- 注 1:** 仅当使能了 IrDA® 编码器 ( $\text{IREN} = 1$ ) 时, 该位的值才影响模块的发送属性。
- 2:** 如果  $\text{UARTEN} = 1$ , 外设的输入和输出必须配置给可用的 RPn/RPIn 引脚。更多信息, 请参见[第 11.4 节“外设引脚选择 \(PPS\) ”](#)。
- 3:** 一旦装入 UTXREG, 只有在两个指令周期后, TRMT 才会有效。

# **PIC24FJ128GA310 系列**

---

---

注:

## 19.0 数据信号调制器

**注：**本数据手册总结了该组PIC24F器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F系列参考手册》中的第55章“数据信号调制器（DSM）”（DS39744A\_CN）。本数据手册中的信息将取代FRM中的信息。

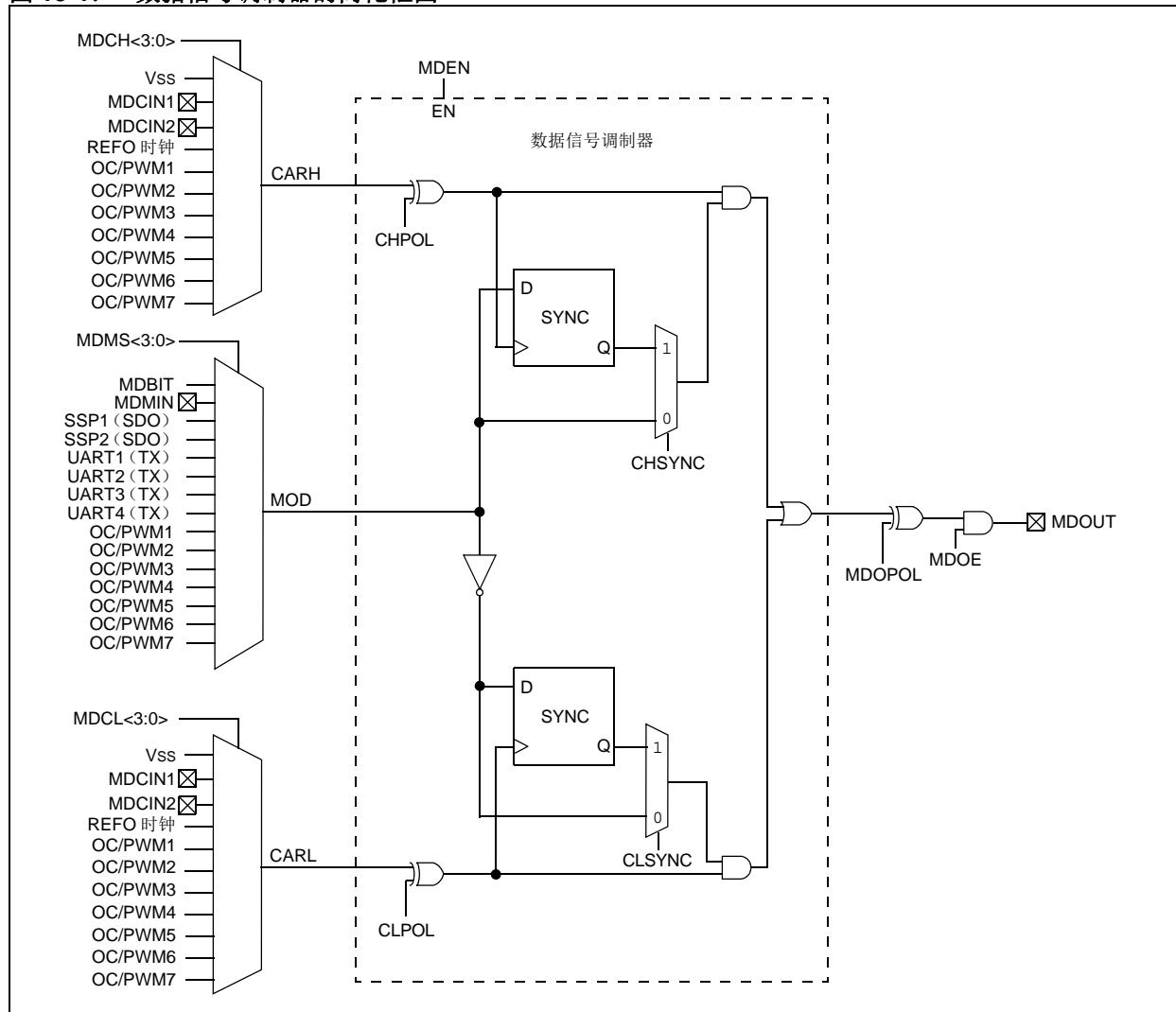
数据信号调制器（DSM）允许用户将数字数据流（“调制器信号”）与载波信号混合来产生调制输出。载波和调制器信号均送到 DSM 模块，信号可以来自某个片内外设的输出，也可以通过某个输入引脚从外部提供。

调制输出信号的产生方式是：对载波和调制器信号执行逻辑与操作，然后送到 MDOUT 引脚上。通过这种方法，DSM 可以产生以下几种键控调制方案：

- 频移键控（Frequency Shift Keying, FSK）
- 相移键控（Phase Shift Keying, PSK）
- 开关键控（On-Off Keying, OOK）

图 19-1 给出了数据信号调制器外设的简化框图。

图 19-1：数据信号调制器的简化框图



# PIC24FJ128GA310 系列

寄存器 19-1： MDCON： 调制器控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
MDEN	—	MSIDL	—	—	—	—	—
bit 15	bit 8						

U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0
—	MDOE	MDSLR	MDOPOL	—	—	—	MDBIT <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **MDEN:** 调制器模块使能位  
1 = 使能调制器模块，并对输入信号进行混合  
0 = 禁止调制器模块，不产生任何输出
- bit 14      未实现：读为 0
- bit 13      **MSIDL:** 调制器空闲模式停止位  
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 12-7     未实现：读为 0
- bit 6        **MDOE:** 调制器模块引脚输出使能位  
1 = 使能调制器引脚输出  
0 = 禁止调制器引脚输出
- bit 5        **MDSLR:** MDOUT 引脚压摆率限制位  
1 = 使能 MDOUT 引脚压摆率限制  
0 = 禁止 MDOUT 引脚压摆率限制
- bit 4        **MDOPOL:** 调制器输出极性选择位  
1 = 调制器输出信号反相  
0 = 调制器输出信号不反相
- bit 3-1      未实现：读为 0
- bit 0        **MDBIT:** 手动调制输入位<sup>(1)</sup>  
1 = 载波已调制  
0 = 载波未调制

注 1： 必须选择 MDBIT 作为调制源 (MDSRC<3:0> = 0000)。

## 寄存器 19-2: MDSRC: 调制器源控制寄存器

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

R/W-x	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
SODIS <sup>(1)</sup>	—	—	—	MS3 <sup>(2)</sup>	MS2 <sup>(2)</sup>	MS1 <sup>(2)</sup>	MS0 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 SODIS: 调制源输出禁止位<sup>(1)</sup>

1 = 禁止驱动外设输出引脚 (通过 MS<3:0> 选择) 的输出信号

0 = 使能驱动外设输出引脚 (通过 MS<3:0> 选择) 的输出信号

bit 6-4 未实现: 读为 0

bit 3-0 MS<3:0> 调制源选择位<sup>(2)</sup>

1111 = 未实现

1110 = 输出比较 /PWM 模块 7 输出

1101 = 输出比较 /PWM 模块 6 输出

1100 = 输出比较 /PWM 模块 5 输出

1011 = 输出比较 /PWM 模块 4 输出

1010 = 输出比较 /PWM 模块 3 输出

1001 = 输出比较 /PWM 模块 2 输出

1000 = 输出比较 /PWM 模块 1 输出

0111 = UART4 TX 输出

0110 = UART3 TX 输出

0101 = UART2 TX 输出

0100 = UART1 TX 输出

0011 = SPI2 模块输出 (SDO2)

0010 = SPI1 模块输出 (SDO1)

0001 = MDMIN 引脚上的输入

0000 = 使用 MDBIT (MDCON<0>) 的手动调制

注 1: 此位仅受到 POR 影响。

2: 这些位不受 POR 影响。

# PIC24FJ128GA310 系列

## 寄存器 19-3: MDCAR: 调制器载波控制寄存器

R/W-x	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
CHODIS	CHPOL	CHSYNC	—	CH3 <sup>(1)</sup>	CH2 <sup>(1)</sup>	CH1 <sup>(1)</sup>	CH0 <sup>(1)</sup>
bit 15							bit 8

R/W-0	R/W-x	R/W-x	U-0	R/W-x	R/W-x	R/W-x	R/W-x
CLODIS	CLPOL	CLSYNC	—	CL3 <sup>(1)</sup>	CL2 <sup>(1)</sup>	CL1 <sup>(1)</sup>	CL0 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **CHODIS:** 调制器载波高信号输出禁止位  
1 = 禁止驱动外设输出引脚（通过 CH<3:0> 选择）的输出信号  
0 = 使能驱动外设输出引脚的输出信号
- bit 14      **CHPOL:** 调制器载波高信号极性选择位  
1 = 选定的载波高信号反相  
0 = 选定的载波高信号不反相
- bit 13      **CHSYNC:** 调制器载波高信号同步使能位  
1 = 调制器先等待载波高信号上出现下降沿, 然后再切换为载波低信号  
0 = 调制器输出不与载波高信号进行同步<sup>(1)</sup>
- bit 12      未实现: 读为 0
- bit 11-8     **CH<3:0>** 调制器数据载波高信号选择位<sup>(1)</sup>  
1111  
... = 保留  
1011  
1010 = 输出比较/PWM 模块 7 输出  
1001 = 输出比较/PWM 模块 6 输出  
1000 = 输出比较/PWM 模块 5 输出  
0111 = 输出比较/PWM 模块 4 输出  
0110 = 输出比较/PWM 模块 3 输出  
0101 = 输出比较/PWM 模块 2 输出  
0100 = 输出比较/PWM 模块 1 输出  
0011 = 参考时钟 (REFO) 输出  
0010 = MDCIN2 引脚上的输入  
0001 = MDCIN1 引脚上的输入  
0000 = Vss
- bit 7        **CLODIS:** 调制器载波低信号输出禁止位  
1 = 禁止驱动外设输出引脚（通过 CL<3:0> 选择）的输出信号  
0 = 使能驱动外设输出引脚的输出信号
- bit 6        **CLPOL:** 调制器载波低信号极性选择位  
1 = 选定的载波低信号反相  
0 = 选定的载波低信号不反相
- bit 5        **CLSYNC:** 调制器载波低信号同步使能位  
1 = 调制器先等待载波低信号上出现下降沿, 然后再切换为载波高信号  
0 = 调制器输出不与载波低信号进行同步<sup>(1)</sup>
- bit 4        未实现: 读为 0
- bit 3-0      **CL<3:0>** 调制器数据载波低信号选择位<sup>(1)</sup>  
位设置与 CH<3:0> 的位设置相同。

**注 1:** 如果载波未进行同步, 则信号流中的载波脉宽可能会变窄, 或者可能出现尖峰。

## 20.0 增强型并行主端口 (EPMP)

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息, 请参见《PIC24F 系列参考手册》中的第 42 章“增强型并行主端口 (EPMP) (DS39730A\_CN)”。本数据手册中的信息取代了 FRM 中的信息。

增强型并行主端口 (EPMP) 模块提供一个 4 位 (仅限主模式)、8 位 (主模式和从模式) 或 16 位 (仅限主模式) 并行数据总线接口, 可与存储器、FIFO、LCD 控制器和其他单片机等片外模块进行通信。此模块可充当通信总线上的主器件或从器件。

对于 EPMP 主模式, 所有外部地址均映射到内部扩展数据空间 (EDS)。这通过为每个片选分配 EDS 区域, 然后将片选分配到特定外部资源 (如存储器或外部控制器) 来实现。该区域不再被分配给其他器件资源, 如 RAM 或 SFR。要对外部资源执行读或写操作, CPU 只需在为 EPMP 分配的地址范围内简单地执行读或写操作。

EPMP 模块的主要特性包括:

- 扩展数据空间 (EDS) 接口支持直接从 CPU 访问
- 最多 23 条可编程地址线
- 最多 2 条片选信号线
- 最多 2 条应答线 (每个片选信号一条)
- 4 位、8 位或 16 位宽数据总线
- 可编程选通选项 (每条片选信号)
  - 独立的读和写选通, 或;
  - 带使能选通的读 / 写选通
- 可编程地址 / 数据复用

- 可编程地址等待状态
- 可编程数据等待状态 (每条片选信号)
- 控制信号的可编程极性 (每条片选信号)
- 支持传统的并行从端口
- 支持增强型并行从端口
  - 地址支持
  - 4 字节深自动递增缓冲器

### 20.1 具体封装类型

虽然所有 PIC24FJ128GA310 系列器件实现了 EPMP, 但 I/O 引脚对某些封装类型中的 16 位主模式工作有所限制。这可从实现的专用片选引脚数和可用的专用地址线数反映出来。[表 20-1](#) 汇总了各项差异。[表 20-2](#) 汇总了所有可用的 EPMP 引脚功能。

64 引脚器件未实现专用片选引脚 (PMCS1 和 PMCS2)。此外, 只可使用 16 条地址线 ( $\text{PMA}_{<15:0>}$ )。如果需要, 可重新映射 PMA14 和 PMA15 以将其分别用作 PMCS1 和 PMCS2。

80 引脚器件未实现专用 PMCS2 引脚。它也只实现了 16 条地址线 ( $\text{PMA}_{<15:0>}$ )。如果需要, 可重新映射 PMA15 以将其用作 PMCS2。

可由器件寻址的存储空间取决于可用的地址线数以及应用所需的片选信号数。引脚数较低的器件更容易受到片选需求的影响, 因为它们占用地址线。[表 20-1](#) 显示了每个引脚数的最大寻址范围。

表 20-1: 按器件引脚数列出的 EPMP 特性差异

器件	专用片选		地址线	地址范围 (字节)		
	CS1	CS2		无 CS	1 CS	2 CS
PIC24FJXXXGA306 (64 引脚)	—	—	16	64K	32K	16K
PIC24FJXXXGA308 (80 引脚)	X	—	16	64K	32K	
PIC24FJXXXGA310 (100 引脚)	X	X	23	16M		

# PIC24FJ128GA310 系列

表 20-2：增强型并行主端口引脚说明

引脚名称（备用功能）	类型	说明
PMA<22:16>	O	地址总线位 <22:16>
PMA<15> (PMCS2)	O	地址总线位 15
	I/O	数据总线位 15 (带复用寻址的 16 位端口)
	O	片选 2 (备用位置)
PMA<14> (PMCS1)	O	地址总线位 14
	I/O	数据总线位 14 (带复用寻址的 16 位端口)
	O	片选 1 (备用位置)
PMA<13:8>	O	地址总线位 <13:8>
	I/O	数据总线位 <13:8> (带复用寻址的 16 位端口)
PMA<7:3>	O	地址总线位 <7:3>
PMA<2> (PMALU)	O	地址总线位 2
	O	复用地址的地址锁存器较高选通端
PMA<1> (PMALH)	I/O	地址总线位 1
	O	复用地址的地址锁存器高选通端
PMA<0> (PMALL)	I/O	地址总线位 0
	O	复用地址的地址锁存器低选通端
PMD<15:8>	I/O	数据总线位 <15:8> (复用寻址)
PMD<7:4>	I/O	数据总线位 <7:4>
	O	地址总线位 <7:4> (带单相复用寻址的 4 位端口)
PMD<3:0>	I/O	数据总线位 <3:0>
PMCS1 <sup>(1)</sup>	I/O	片选 1
PMCS2 <sup>(2)</sup>	O	片选 2
PMWR (PMENB)	I/O	写选通 <sup>(3)</sup>
PMRD	I/O	读选通 <sup>(3)</sup>
(PMRD/PMWR)	I/O	读 / 写信号 <sup>(3)</sup>
PMBE1	O	字节指示
PMBE0	O	半字节或字节指示
PMACK1	I	应答信号 1
PMACK2	I	应答信号 2

注 1：这些引脚仅在 80 引脚和 100 引脚器件中实现。

2：这些引脚仅在 100 引脚器件中实现。

3：信号功能取决于 MODE<1:0> 和 SM 位 (PMCON1<9:8> 和 PMCSxCF<8>) 的设置。

## 寄存器 20-1： PMCON1： EPMP 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
PMPE	—	PSIDL	ADRMUX1	ADRMUX0	—	MODE1	MODE0
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
CSF1	CSF0	ALP	ALMODE	—	BUSKEEP	IRQM1	IRQM0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>PMPE</b> : 并行主端口使能位 1 = 使能 EPMP 0 = 禁止 EPMP
bit 14	未实现：读为 0
bit 13	<b>PSIDL</b> : 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 12-11	<b>ADRMUX&lt;1:0&gt;</b> : 地址 / 数据复用选择位 11 = 低地址位使用 3 个地址段与数据位复用 10 = 低地址位使用 2 个地址段与数据位复用 01 = 低地址位使用 1 个地址段与数据位复用 00 = 地址和数据使用独立的引脚
bit 10	未实现：读为 0
bit 9-8	<b>MODE&lt;1:0&gt;</b> : 并行端口模式选择位 11 = 主模式 10 = 增强型 PSP；所使用的引脚为 PMRD、PMWR、PMCS、PMD<7:0> 和 PMA<1:0> 01 = 缓冲的 PSP；所使用的引脚为 PMRD、PMWR、PMCS 和 PMD<7:0> 00 = 传统并行从端口；使用 PMRD、PMWR、PMCS 和 PMD<7:0> 引脚
bit 7-6	<b>CSF&lt;1:0&gt;</b> : 片选功能位 11 = 保留 10 = PMA<15> 用于片选 2，PMA<14> 用于片选 1 01 = PMA<15> 用于片选 2，PMCS1 用于片选 1 00 = PMCS2 用于片选 2，PMCS1 用于片选 1
bit 5	<b>ALP</b> : 地址锁存器极性位 1 = 高电平有效 ( <u>PMALL</u> 、 <u>PMALH</u> 和 <u>PMALU</u> ) 0 = 低电平有效 ( <u>PMALL</u> 、 <u>PMALH</u> 和 <u>PMALU</u> )
bit 4	<b>ALMODE</b> : 地址锁存器选通模式位 1 = 使能“智能”地址选通（仅在当前访问将导致锁存地址不同于先前地址时，才存在各地址段） 0 = 禁止“智能”地址选通
bit 3	未实现：读为 0
bit 2	<b>BUSKEEP</b> : 总线保持位 1 = 当未驱动数据总线时，保持其最后一个值 0 = 当未驱动数据总线时，其处于高阻态
bit 1-0	<b>IRQM&lt;1:0&gt;</b> : 中断请求模式位 11 = 当读取读缓冲区 3 或写入写缓冲区 3 时产生中断（缓冲的 PSP 模式），或当 PMA<1:0> = 11 时执行读或写操作时产生中断（仅限可寻址 PSP 模式） 10 = 保留 01 = 在读 / 写周期结束时产生中断 00 = 不产生中断

# PIC24FJ128GA310 系列

寄存器 20-2: PMCON2: EPMP 控制寄存器 2

R-0, HSC	U-0	R/C-0, HS	R/C-0, HS	U-0	U-0	U-0	U-0
BUSY	—	ERROR	TIMEOUT	—	—	—	—
bit 15							bit 8

| R/W-0                  |
|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| RADDR23 <sup>(1)</sup> | RADDR22 <sup>(1)</sup> | RADDR21 <sup>(1)</sup> | RADDR20 <sup>(1)</sup> | RADDR19 <sup>(1)</sup> | RADDR18 <sup>(1)</sup> | RADDR17 <sup>(1)</sup> | RADDR16 <sup>(1)</sup> |
| bit 7                  |                        |                        |                        |                        |                        |                        | bit 0                  |

图注:

HS = 可由硬件置 1 的位

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **BUSY:** 忙位 (仅用于主模式)

1 = 端口忙

0 = 端口不忙

bit 14      未实现: 读为 0

bit 13      **ERROR:** 错误位

1 = 传输错误 (请求非法传输)

0 = 传输成功完成

bit 12      **TIMEOUT:** 超时标志位

1 = 传输超时

0 = 传输成功完成

bit 11-8      未实现: 读为 0

bit 7-0      **RADDR<23:16>:** 并行主端口保留地址空间位<sup>(1)</sup>

注 1: 如果 RADDR<23:16> = 00000000, 则片选 2 的最后一个 EDS 地址将为 FFFFFFFh。

## 寄存器 20-3: PMCON3: EPMP 控制寄存器 3

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PTWREN	PTRDEN	PTBE1EN	PTBE0EN	—	AWAITM1	AWAITM0	AWAITE
bit 15							bit 8

U-0	R/W-0						
—	PTEN22 <sup>(1)</sup>	PTEN21 <sup>(1)</sup>	PTEN20 <sup>(1)</sup>	PTEN19 <sup>(1)</sup>	PTEN18 <sup>(1)</sup>	PTEN17 <sup>(1)</sup>	PTEN16 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **PTWREN:** 写 / 使能选通端口使能位

1 = 使能 PMWR/PMENB 端口  
0 = 禁止 PMWR/PMENB 端口

bit 14 **PTRDEN:** 读 / 写选通端口使能位

1 = 使能 PMRD/PMWR  
0 = 禁止 PMRD/PMWR

bit 13 **PTBE1EN:** 高半字节 / 字节使能端口使能位

1 = 使能 PMBE1 端口  
0 = 禁止 PMBE1 端口

bit 12 **PTBE0EN:** 低半字节 / 字节使能端口使能位

1 = 使能 PMBE0 端口  
0 = 禁止 PMBE0 端口

bit 11 未实现: 读为 0

bit 10-9 **AWAITM<1:0>:** 地址锁存器选通等待状态位

11 = 等待 3½ 个 TCY  
10 = 等待 2½ 个 TCY  
01 = 等待 1½ 个 TCY  
00 = 等待 ½ 个 TCY

bit 8 **AWAITE:** 地址锁存器选通后地址保持等待状态位

1 = 等待 1¼ 个 TCY  
0 = 等待 ¼ 个 TCY

bit 7 未实现: 读为 0

bit 6-0 **PTEN<22:16>:** EPMP 地址端口使能位<sup>(1)</sup>

1 = PMA<22:16> 用作 EPMP 地址线  
0 = PMA<22:16> 用作端口 I/O

**注 1:** 80 和 64 引脚器件 (PIC24FJXXXGA306 和 PIC24FJXXXGA308) 中没有这些位。

# PIC24FJ128GA310 系列

寄存器 20-4: PMCON4: EPMP 控制寄存器 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
PTEN15	PTEN14	PTEN13	PTEN12	PTEN11	PTEN10	PTEN9	PTEN8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| PTEN7 | PTEN6 | PTEN5 | PTEN4 | PTEN3 | PTEN2 | PTEN1 | PTEN0 |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **PTEN15:** PMA15 端口使能位

1 = PMA15 用作地址线 15 或片选 2  
0 = PMA15 用作端口 I/O

bit 14      **PTEN14:** PMA14 端口使能位

1 = PMA14 用作地址线 14 或片选 1  
0 = PMA14 用作端口 I/O

bit 13-3    **PTEN<13:3>:** EPMP 地址端口使能位

1 = PMA<13:3> 用作 EPMP 地址线  
0 = PMA<13:3> 用作端口 I/O

bit 2-0     **PTEN<2:0>:** PMALU/PMALH/PMALL 选通使能位

1 = PMA<2:0> 用作地址线或地址锁存器选通  
0 = PMA<2:0> 用作端口 I/O

## 寄存器 20-5: PMCSxCF: 片选 x 配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
CSDIS	CSP	CSPTEN	BEP	—	WRSP	RDSP	SM
bit 15							bit 8

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
ACKP	PTSZ1	PTSZ0	—	—	—	—	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- |         |  |
|---------|--|
| bit 15  | <b>CSDIS:</b> 片选 x 禁止位<br>1 = 禁止片选 x 功能<br>0 = 使能片选 x 功能   |
| bit 14  | <b>CSP:</b> 片选 x 极性位<br>1 = 高电平有效 (PMCSx)<br>0 = 低电平有效 (PMCSx)   |
| bit 13  | <b>CSPTEN:</b> PMCSx 端口使能位<br>1 = 使能 PMCSx 端口<br>0 = 禁止 PMCSx 端口   |
| bit 12  | <b>BEP:</b> 片选 x 半字节 / 字节使能极性位<br>1 = 半字节 / 字节使能高电平有效 (PMBE0 和 PMBE1)<br>0 = 半字节 / 字节使能低电平有效 (PMBE0 和 PMBE1)   |
| bit 11  | 未实现: 读为 0  |
| bit 10  | <b>WRSP:</b> 片选 x 写选通极性位<br>对于从模式和主模式 (SM = 0) :<br>1 = 写选通高电平有效 (PMWR)<br>0 = 写选通低电平有效 (PMWR)<br>对于主模式 (SM = 1) :<br>1 = 使能选通高电平有效 (PMENB)<br>0 = 使能选通低电平有效 (PMENB)               |
| bit 9   | <b>RDSP:</b> 片选 x 读选通极性位<br>对于从模式和主模式 (SM = 0) :<br>1 = 读选通高电平有效 (PMRD)<br>0 = 读选通低电平有效 (PMRD)<br>对于主模式 (SM = 1) :<br>1 = 读 / 写选通高电平有效 (PMRD/PMWR)<br>0 = 读 / 写选通低电平有效 (PMRD/PMWR) |
| bit 8   | <b>SM:</b> 片选 x 选通模式位<br>1 = 读 / 写和使能选通 (PMRD/PMWR 和 PMENB)<br>0 = 读和写选通 (PMRD 和 PMWR)   |
| bit 7   | <b>ACKP:</b> 片选 x 应答极性位<br>1 = ACK 高电平有效 (PMACK1)<br>0 = ACK 低电平有效 (PMACK1)  |
| bit 6-5 | <b>PTSZ&lt;1:0&gt;:</b> 片选 x 端口大小位<br>11 = 保留<br>10 = 16 位端口大小 (PMD<15:0>)<br>01 = 4 位端口大小 (PMD<3:0>)<br>00 = 8 位端口大小 (PMD<7:0>)   |
| bit 4-0 | 未实现: 读为 0  |

# PIC24FJ128GA310 系列

寄存器 20-6: PMCSxBS: 片选 x 基地址寄存器 <sup>(2)</sup>

| R/W <sup>(1)</sup> |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| BASE23             | BASE22             | BASE21             | BASE20             | BASE19             | BASE18             | BASE17             | BASE16             |
| bit 15             |                    |                    |                    |                    |                    |                    |                    |

R/W <sup>(1)</sup>	U-0						
BASE15	—	—	—	—	—	—	—
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8      **BASE<23:15>**: 片选 x 基地址位 <sup>(1)</sup>

bit 6-0      未实现: 读为 0

注 1: POR 时, PMCS1BS 的值为 0080h, PMCS2BS 的值为 0880h。

2: 如果对整个 PMCS2BS 寄存器写入 0x0000, 则片选 1 的最后一个 EDS 地址将为 FFFFFFFh。在这种情况下, 不应使用片选 2。PMCS1BS 不具有此类功能。

## 寄存器 20-7: PMCSxMD: 片选 x 模式寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
ACKM1	ACKM0	AMWAIT2	AMWAIT1	AMWAIT0	—	—	—
bit 15							bit 8

| R/W-0   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| DWAITB1 | DWAITB0 | DWAITM3 | DWAITM2 | DWAITM1 | DWAITM0 | DWAITE1 | DWAITE0 |
| bit 7   |         |         |         |         |         |         | bit 0   |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 **ACKM<1:0>**: 片选 x 应答模式位

11 = 保留

10 = PMACKx 用于确定何时完成读 / 写操作

01 = PMACKx 用于确定何时完成读 / 写操作且超时

(如果 DWAITM<3:0> = 0000, 则最大超时为 255 个 TCY, 否则为 DWAITM<3:0> 个周期)。

00 = 未使用 PMACKx

bit 13-11 **AMWAIT<2:0>**: 片选 x 备用主等待状态位

111 = 等待 10 个备用主周期

...

001 = 等待 4 个备用主周期

000 = 等待 3 个备用主周期

bit 10-8 未实现: 读为 0

bit 7-6 **DWAITB<1:0>**: 读 / 写选通前片选 x 数据建立等待状态位

11 = 等待 3½ 个 TCY

10 = 等待 2½ 个 TCY

01 = 等待 1½ 个 TCY

00 = 等待 ½ 个 TCY

bit 5-2 **DWAITM<3:0>**: 片选 x 数据读 / 写选通等待状态位

对于写操作:

1111 = 等待 15½ 个 TCY

...

0001 = 等待 1½ 个 TCY

0000 = 等待 ½ 个 TCY

对于读操作:

1111 = 等待 15½ 个 TCY

...

0001 = 等待 1½ 个 TCY

0000 = 等待 ½ 个 TCY

bit 1-0 **DWAITE<1:0>**: 读 / 写选通后片选 x 数据保持等待状态位

对于写操作:

11 = 等待 3½ 个 TCY

10 = 等待 2½ 个 TCY

01 = 等待 1½ 个 TCY

00 = 等待 ½ 个 TCY

对于读操作:

11 = 等待 3 个 TCY

10 = 等待 2 个 TCY

01 = 等待 1 个 TCY

00 = 等待 0 个 TCY

# PIC24FJ128GA310 系列

寄存器 20-8： PMSTAT： EPMP 状态寄存器（仅从模式）

R-0, HSC	R/W-0 HS	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC	R-0, HSC
IBF	IBOV	—	—	IB3F <sup>(1)</sup>	IB2F <sup>(1)</sup>	IB1F <sup>(1)</sup>	IB0F <sup>(1)</sup>
bit 15	bit 8						

R-1, HSC	R/W-0 HS	U-0	U-0	R-1, HSC	R-1, HSC	R-1, HSC	R-1, HSC
OBE	OBUF	—	—	OB3E	OB2E	OB1E	OB0E
bit 7	bit 0						

图注：

HS = 可由硬件置 1 的位

HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **IBF:** 输入缓冲区满状态位

1 = 所有可写的输入缓冲寄存器已满

0 = 部分或所有可写的输入缓冲寄存器为空

bit 14 **IBOV:** 输入缓冲器溢出状态位

1 = 尝试对已满的输入寄存器执行写操作（必须由软件清零）

0 = 未发生溢出

bit 13-12 未实现：读为 0

bit 11-8 **IB3F:IB0F:** 输入缓冲器 x 满状态位<sup>(1)</sup>

1 = 输入缓冲器包含尚未读取的数据（读缓冲器将清零此位）

0 = 输入缓冲器不包含任何未读取的数据

bit 7 **OBE:** 输出缓冲器空状态位

1 = 所有可读的输出缓冲寄存器均为空

0 = 部分或所有可读的输出缓冲寄存器已满

bit 6 **OBUF:** 输出缓冲器下溢状态位

1 = 对空的输出寄存器执行了读操作（必须由软件清零）

0 = 未发生下溢

bit 5-4 未实现：读为 0

bit 3-0 **OB3E:OB0E:** 输出缓冲器 x 空状态位

1 = 输出缓冲器为空（将数据写入该缓冲器会将该位清零）

0 = 输出缓冲器包含未发送的数据

注 1： 即使某个位表示缓冲器中的字节，在进行字节读取时与字（字节 0 和 1 或字节 2 和 3）对应的位也被清零。

## 寄存器 20-9: PADCFG1: 烙盘配置控制寄存器

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	U-0	U-0	U-0	R/W-0
—	—	—	—	—	—	—	PMPTTL
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-1 未实现: 读为 0

bit 0 **PMPTTL:** EPMP 模块 TTL 输入缓冲器选择位

1 = EPMP 模块输入 (PMDx 和 PMCS1) 使用 TTL 输入缓冲器

0 = EPMP 模块输入使用施密特触发器输入缓冲器

# **PIC24FJ128GA310 系列**

---

---

注:

## 21.0 液晶显示（LCD）控制器

**注：**本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关详细信息，请参见《PIC24F 系列参考手册》中的第 52 章“液晶显示模块（LCD）”（39740A\_CN）。本数据手册中的信息取代了 FRM 中的信息。

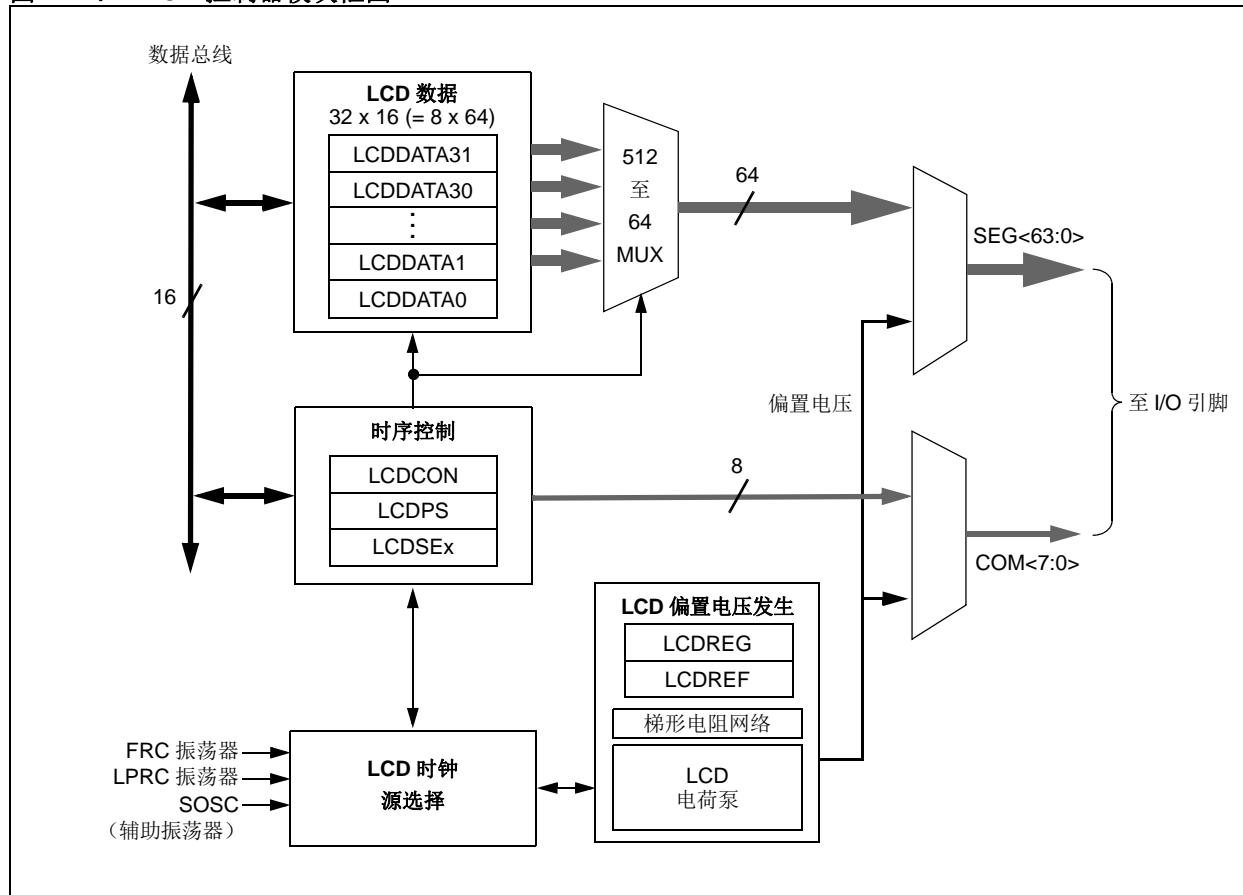
液晶显示（Liquid Crystal Display, LCD）控制器生成直接驱动静态或复用的 LCD 面板所需的数据和时序控制。在 100 引脚器件（PIC24FJXXXGA310）中，该模块最多可驱动 8 个公共端，在使用 5 至 8 个公共端时最多驱动 60 段，在使用 1 至 4 个公共端时最多驱动 64 段。

该模块具有以下特性：

- 直接驱动 LCD 面板
- 3 个预分频比可选的 LCD 时钟源
- 最多 8 个公共端：
  - 静态（1 个公共端）
  - 1/2 复用（2 个公共端）
  - 1/3 复用（3 个公共端）
  - 1/8 复用（8 个公共端）
- 根据所选的复用模式，能够驱动 30（64 引脚器件）至 64（100 引脚器件）段
- 静态、1/2 或 1/3 LCD 偏置
- 带有专用电荷泵的片上偏置电压发生器，支持一系列固定和可变偏压选项
- 用于生成偏压的内部电阻
- 使用内部偏置的 LCD 软件对比度控制

图 21-1 给出了该模块的简化框图。

图 21-1：LCD 控制器模块框图



# PIC24FJ128GA310 系列

## 21.1 寄存器

LCD 控制器最多具有 40 个寄存器：

- LCD 控制寄存器 (LCDCON)
- LCD 电荷泵控制寄存器 (LCDREG)
- LCD 相位寄存器 (LCDPS)
- LCD 梯形电阻分压网络控制寄存器 (LCDREF)
- 4 个 LCD 段使能寄存器 (LCDSE3:LCDSE0)
- 最多 32 个 LCD 数据寄存器  
(LCDDATA31:LCDDATA0)

寄存器 21-1：LCDCON：LCD 控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
LCDEN	—	LCDSIDL	—	—	—	—	—
bit 15							bit 8

U-0	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	SLPEN	WERR	CS1	CS0	LMUX2	LMUX1	LMUX0
bit 7							bit 0

### 图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>LCDEN:</b> LCD 驱动使能位 1 = 使能 LCD 驱动模块 0 = 禁止 LCD 驱动模块
bit 14	未实现：读为 0
bit 13	<b>LCDSIDL:</b> CPU 空闲模式下停止 LCD 驱动控制位 1 = CPU 空闲模式下 LCD 驱动器停止工作 0 = CPU 空闲模式下 LCD 驱动器继续工作
bit 12-7	未实现：读为 0
bit 6	<b>SLPEN:</b> 休眠模式下 LCD 驱动使能位 1 = 休眠模式下禁止 LCD 驱动模块 0 = 休眠模式下使能 LCD 驱动模块
bit 5	<b>WERR:</b> LCD 写失败错误位 1 = LCDDATAx 寄存器在 WA (LCDPS<4>) = 0 时写入 (必须在软件中清零) 0 = 无 LCD 写错误
bit 4-3	<b>CS&lt;1:0&gt;:</b> 时钟源选择位 00 = FRC 01 = LPRC 1x = SOSC

## 寄存器 21-1: LCDCON: LCD 控制寄存器 (续)

bit 2-0      **LMUX<2:0>**: 公共端选择位

LMUX<2:0>	复用	偏置
111	1/8 MUX (COM<7:0>)	1/3
110	1/7 MUX (COM<6:0>)	1/3
101	1/6 MUX (COM<5:0>)	1/3
100	1/5 MUX (COM<4:0>)	1/3
011	1/4 MUX (COM<3:0>)	1/3
010	1/3 MUX (COM<2:0>)	1/2 或 1/3
001	1/2 MUX (COM<1:0>)	1/2 或 1/3
000	静态 (COM0)	静态

注: 对于 4 个公共端以上的复用, COM4、COM5、COM6 和 COM7 也有段功能。因此, 如果在复用时使能 COM, 段将不可用于引脚。

# PIC24FJ128GA310 系列

寄存器 21-2: LCDREG: LCD 电荷泵控制寄存器

RW-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
CPEN <sup>(1)</sup>	—	—	—	—	—	—	—
bit 15	bit 8						

U-0	U-0	RW-1	RW-1	RW-1	RW-1	RW-0	RW-0
—	—	BIAS2	BIAS1	BIAS0	MODE13	CKSEL1	CKSEL0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **CPEN:** 3.6V 电荷泵使能位<sup>(1)</sup>

1 = 稳压器产生最高电压 (3.6V)

0 = 系统中的最高电压由外部提供 (AVDD)

bit 14-6      未实现: 读为 0

bit 5-3      **BIAS<2:0>:** 稳压器电压输出控制位

111 = 3.60V 峰值 (在 LCDBIAS0 上偏移 0V)

110 = 3.47V 峰值 (在 LCDBIAS0 上偏移 0.13V)

101 = 3.34V 峰值 (在 LCDBIAS0 上偏移 0.26V)

100 = 3.21V 峰值 (在 LCDBIAS0 上偏移 0.39V)

011 = 3.08V 峰值 (在 LCDBIAS0 上偏移 0.52V)

010 = 2.95V 峰值 (在 LCDBIAS0 上偏移 0.65V)

001 = 2.82V 峰值 (在 LCDBIAS0 上偏移 0.78V)

000 = 2.69V 峰值 (在 LCDBIAS0 上偏移 0.91V)

bit 2      **MODE13:** 1/3 LCD 偏置模式使能位

1 = 稳压器输出支持 1/3 LCD 偏置模式

0 = 稳压器输出支持静态 LCD 偏置模式

bit 1-0      **CLKSEL<1:0>:** 稳压器时钟选择控制位

11 = LPRC 31 kHz

10 = 8 MHz FRC

01 = SOSC

00 = 禁止稳压器和悬空稳压器电压输出

注 1: 使用电荷泵时, LCDBIASx 引脚和 VLCAP1/VLACAP2 引脚应设置为模拟, 并且相应的 TRIS 位应设置为输入。

## 寄存器 21-3: LCDPS: LCD 相位寄存器

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-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
WFT	BIASMD	LCDA	WA	LP3	LP2	LP1	LP0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7 **WFT:** 波形类型选择位

1 = B 型波形 (在每一帧边界改变相位)

0 = A 型波形 (在每种公共端复用类型的一帧内改变相位)

bit 6 **BIASMD:** 偏置模式选择位

当 LMUX<2:0> = 000 或 011 至 111 时:

0 = 静态偏置模式 (不要将此位置 1)

当 LMUX<2:0> = 001 或 010 时:

1 = 1/2 偏置模式

0 = 1/3 偏置模式

bit 5 **LCDA:** LCD 工作状态位

1 = LCD 驱动模块正在工作

0 = LCD 驱动模块不在工作

bit 4 **WA:** LCD 写允许状态位

1 = 允许写入 LCDDATAx 寄存器

0 = 不允许写入 LCDDATAx 寄存器

bit 3-0 **LP<3:0>:** LCD 预分频比选择位

1111 = 1:16

1110 = 1:15

1101 = 1:14

1100 = 1:13

1011 = 1:12

1010 = 1:11

1001 = 1:10

1000 = 1:9

0111 = 1:8

0110 = 1:7

0101 = 1:6

0100 = 1:5

0011 = 1:4

0010 = 1:3

0001 = 1:2

0000 = 1:1

# PIC24FJ128GA310 系列

## 寄存器 21-4: LCDSE<sub>x</sub>: LCD 段 x 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
SE(n+15)	SE(n+14)	SE(n+13)	SE(n+12)	SE(n+11)	SE(n+10)	SE(n+9)	SE(n+8)
bit 15							bit 8

R/W-0	R/W-0						
SE(n+7)	SE(n+6)	SE(n+5)	SE(n+4)	SE(n+3)	SE(n+2)	SE(n+1)	SE(n)
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 7-0      **SE(n + 15):SE(n):** 段使能位

对于 LCDSE0: n = 0

对于 LCDSE1: n = 16

对于 LCDSE2: n = 32

对于 LCDSE3: n = 48<sup>(1)</sup>

1 = 使能引脚的段功能, 禁止数字 I/O

0 = 禁止引脚的段功能, 使能数字 I/O

注 1: 为使 SEG49 正常工作, 需要禁止 JTAG。

## 寄存器 21-5: LCDDATA<sub>x</sub>: LCD 数据 x 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
S(n+15)Cy	S(n+14)Cy	S(n+13)Cy	S(n+12)Cy	S(n+11)Cy	S(n+10)Cy	S(n+9)Cy	S(n+8)Cy
bit 15							bit 8

R/W-0	R/W-0						
S(n+7)Cy	S(n+6)Cy	S(n+5)Cy	S(n+4)Cy	S(n+3)Cy	S(n+2)Cy	S(n+1)Cy	S(n)Cy
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0      **S(n + 15)Cy:S(n)Cy:** 像素点亮位

对于寄存器 LCDDATA0 至 LCDDATA3: n = (16x), y = 0

对于寄存器 LCDDATA4 至 LCDDATA7: n = (16(x - 4)), y = 1

对于寄存器 LCDDATA8 至 LCDDATA11: n = (16(x - 8)), y = 2

对于寄存器 LCDDATA12 至 LCDDATA15: n = (16(x - 12)), y = 3

对于寄存器 LCDDATA16 至 LCDDATA19: n = (16(x - 16)), y = 4

对于寄存器 LCDDATA20 至 LCDDATA23: n = (16(x - 20)), y = 5

对于寄存器 LCDDATA24 至 LCDDATA27: n = (16(x - 24)), y = 6

对于寄存器 LCDDATA28 至 LCDDATA31: n = (16(x - 28)), y = 7

1 = 像素点亮

0 = 像素不点亮

表 21-1：各种段和 COM 组合的 LCDDATA 寄存器和位

COM 线	段			
	0 至 15	16 至 31	32 至 47	48 至 64
0	LCDDATA0 S00C0:S15C0	LCDDATA1 S16C0:S31C0	LCDDATA2 S32C0:S47C0	LCDDATA3 S48C0:S63C0
1	LCDDATA4 S00C1:S15C1	LCDDATA5 S16C1:S31C1	LCDDATA6 S32C1:S47C1	LCDDATA7 S48C1:S63C1
2	LCDDATA8 S00C2:S15C2	LCDDATA9 S16C2:S31C2	LCDDATA10 S32C2:S47C2	LCDDATA11 S48C2:S63C2
3	LCDDATA12 S00C3:S15C3	LCDDATA13 S16C3:S31C3	LCDDATA14 S32C3:S47C3	LCDDATA15 S48C3:S63C3
4	LCDDATA16 S00C4:S15C4	LCDDATA17 S16C4:S31C4	LCDDATA18 S32C4:S47C4	LCDDATA19 S48C4:S59C4
5	LCDDATA20 S00C5:S15C5	LCDDATA21 S16C5:S31C5	LCDDATA22 S32C5:S47C5	LCDDATA23 S48C5:S69C5
6	LCDDATA24 S00C6:S15C6	LCDDATA25 S16C6:S31C6	LCDDATA26 S32C6:S47C6	LCDDATA27 S48C6:S59C6
7	LCDDATA28 S00C7:S15C7	LCDDATA29 S16C7:S31C7	LCDDATA30 S32C7:S47C7	LCDDATA31 S48C7:S59C7

# PIC24FJ128GA310 系列

寄存器 21-6: LCDREF: LCD 参考电压梯形电阻网络控制寄存器

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDIRE	—	LCDCST2	LCDCST1	LCDCST0	VLCD3PE <sup>(1)</sup>	VLCD2E <sup>(1)</sup>	VLCD1E <sup>(1)</sup>
bit 15							

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
LRLAP1	LRLAP0	LRLBP1	LRLBP0	—	LRLAT2	LRLAT1	LRLAT0
bit 7							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **LCDIRE:** LCD 内部参考电压使能位  
1 = 使能内部 LCD 参考电压, 并将其连接到内部对比度控制电路  
0 = 禁止内部 LCD 参考电压
- bit 14      未实现: 读为 0
- bit 13-11    **LCDCST<2:0>:** LCD 对比度控制位  
选择 LCD 对比度控制梯形电阻网络的阻值:  
111 = 梯形电阻网络为最大阻值 (最小对比度)  
110 = 梯形电阻网络为最大阻值的 6/7  
101 = 梯形电阻网络为最大阻值的 5/7  
100 = 梯形电阻网络为最大阻值的 4/7  
011 = 梯形电阻网络为最大阻值的 3/7  
010 = 梯形电阻网络为最大阻值的 2/7  
001 = 梯形电阻网络为最大阻值的 1/7  
000 = 最小阻值 (最大对比度); 梯形电阻网络短路
- bit 10      **VLCD3PE:** 偏置电压 3 引脚使能位<sup>(1)</sup>  
1 = 偏置电压 3 电平连接外部引脚 LCDBIAS3  
0 = 偏置电压 3 电平在内部 (内部梯形电阻网络)
- bit 9        **VLCD2PE:** 偏置电压 2 引脚使能位<sup>(1)</sup>  
1 = 偏置电压 2 电平连接外部引脚 LCDBIAS2  
0 = 偏置电压 2 电平在内部 (内部梯形电阻网络)
- bit 8        **VLCD1PE:** 偏置电压 1 引脚使能位<sup>(1)</sup>  
1 = 偏置电压 1 电平连接外部引脚 LCDBIAS1  
0 = 偏置电压 1 电平在内部 (内部梯形电阻网络)
- bit 7-6     **LRLAP<1:0>:** LCD 参考电压梯形电阻网络 A 时间功耗控制位  
时间间隔 A 期间:  
11 = 内部 LCD 参考电压梯形电阻网络工作在高功耗模式下  
10 = 内部 LCD 参考电压梯形电阻网络工作在中等功耗模式下  
11 = 内部 LCD 参考电压梯形电阻网络工作在低功耗模式下  
00 = 内部 LCD 参考电压梯形电阻网络掉电并断开连接
- bit 5-4     **LRLBP<1:0>:** LCD 参考电压梯形电阻网络 B 时间功耗控制位  
时间间隔 B 期间:  
11 = 内部 LCD 参考电压梯形电阻网络工作在高功耗模式下  
10 = 内部 LCD 参考电压梯形电阻网络工作在中等功耗模式下  
11 = 内部 LCD 参考电压梯形电阻网络工作在低功耗模式下  
00 = 内部 LCD 参考电压梯形电阻网络掉电并断开连接
- bit 3        未实现: 读为 0

注 1: 使用外部梯形电阻网络偏置时, LCDBIASx 引脚应设置为模拟, 并且相应的 TRIS 位应设置为输入。

## 寄存器 21-6: LCDREF: LCD 参考电压梯形电阻网络控制寄存器 (续)

bit 2-0

**LRLAT<2:0>**: LCD 参考电压梯形电阻网络 A 时间间隔控制位

设置 32 个时钟计数，表示 A 时间间隔功耗模式有效的时间。

对于 A 型波形 (WFT = 0) :

111 = 内部LCD参考电压梯形电阻网络在7个时钟周期内处于A功耗模式而在9个时钟周期内处于B功耗模式  
110 = 内部LCD参考电压梯形电阻网络在6个时钟周期内处于A功耗模式而在10个时钟周期内处于B功耗模式  
101 = 内部LCD参考电压梯形电阻网络在5个时钟周期内处于A功耗模式而在11个时钟周期内处于B功耗模式  
100 = 内部LCD参考电压梯形电阻网络在4个时钟周期内处于A功耗模式而在12个时钟周期内处于B功耗模式  
011 = 内部LCD参考电压梯形电阻网络在3个时钟周期内处于A功耗模式而在13个时钟周期内处于B功耗模式  
010 = 内部LCD参考电压梯形电阻网络在2个时钟周期内处于A功耗模式而在14个时钟周期内处于B功耗模式  
001 = 内部LCD参考电压梯形电阻网络在1个时钟周期内处于A功耗模式而在15个时钟周期内处于B功耗模式  
000 = 内部 LCD 参考电压梯形电阻网络始终处于 B 功耗模式下

对于 B 型波形 (WFT = 1) :

111 = 内部LCD参考电压梯形电阻网络在7个时钟周期内处于A功耗模式而在25个时钟周期内处于B功耗模式  
110 = 内部LCD参考电压梯形电阻网络在6个时钟周期内处于A功耗模式而在26个时钟周期内处于B功耗模式  
101 = 内部LCD参考电压梯形电阻网络在5个时钟周期内处于A功耗模式而在27个时钟周期内处于B功耗模式  
100 = 内部LCD参考电压梯形电阻网络在4个时钟周期内处于A功耗模式而在28个时钟周期内处于B功耗模式  
011 = 内部LCD参考电压梯形电阻网络在3个时钟周期内处于A功耗模式而在29个时钟周期内处于B功耗模式  
010 = 内部LCD参考电压梯形电阻网络在2个时钟周期内处于A功耗模式而在30个时钟周期内处于B功耗模式  
001 = 内部LCD参考电压梯形电阻网络在1个时钟周期内处于A功耗模式而在31个时钟周期内处于B功耗模式  
000 = 内部 LCD 参考电压梯形电阻网络始终处于 B 功耗模式下

**注 1:** 使用外部梯形电阻网络偏置时，LCDBIASx 引脚应设置为模拟，并且相应的 TRIS 位应设置为输入。

# **PIC24FJ128GA310 系列**

---

---

注:

## 22.0 实时时钟和日历 (RTCC)

**注:** 本数据手册总结了该组PIC24F器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。有关实时时钟和日历的更多信息,请参见《PIC24F系列参考手册》中的第29章“实时时钟和日历(RTCC)”(39696A\_CN)。

RTCC为用户提供了可校准的实时时钟和日历(Real-Time Clock and Calendar, RTCC)功能。

RTCC模块的主要特性包括:

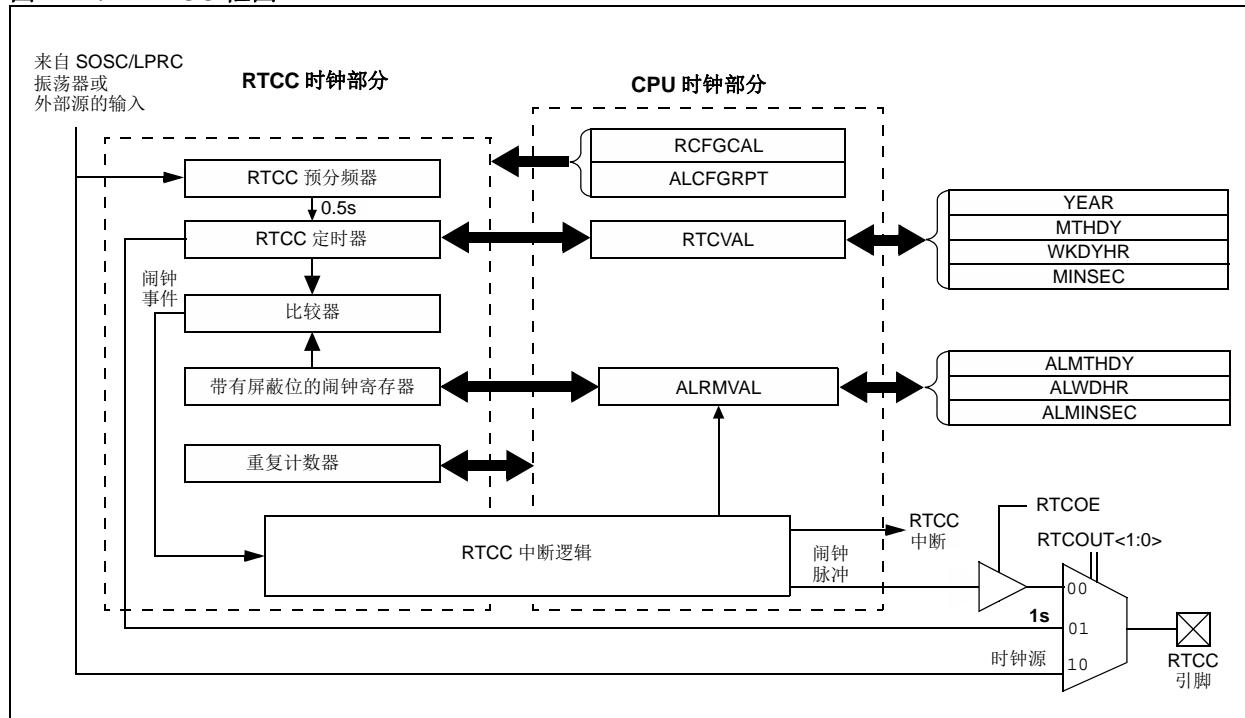
- 可在深度休眠模式下工作
- 可选的时钟源
- 采用24小时格式提供小时、分和秒
- 可看到一个半秒周期
- 提供日历——星期、日期、月和年
- 可配置闹钟每半秒、1秒、10秒、1分、10分、1小时、1天、1周、1月或1年响一次
- 闹钟使用递减计数器进行重复
- 闹钟具有无限重复响铃
- 年份2000至2099闰年纠正

- 采用BCD格式,以减小软件开销
- 为长期电池工作进行了优化
- 32.768 kHz时钟晶振的用户校准/带周期性自动调整的32 kHz INTRC频率
- 小数秒同步
- 校准每月误差在±2.64秒内
- 校准最高260 ppm的晶振误差
- 能够定期唤醒外部器件,而无需CPU干预(外部电源控制)
- 针对外部电路控制的电源控制输出
- 校准每15秒生效一次
- 从以下任意一项运行:
  - 外部32.768 kHz实时时钟(RTC)
  - 内部31.25 kHz LPRC时钟
  - 50 Hz或60 Hz外部输入

### 22.1 RTCC 源时钟

用户可选择SOSC/LPRC晶体振荡器、LPRC内部振荡器或外部50 Hz/60 Hz电力线输入作为RTCC模块的参考时钟。用户可以基于系统总体的需要,通过此方法使系统成本、精度和功耗之间达到平衡。

图 22-1: RTCC 框图



## 22.2 RTCC 模块寄存器

RTCC 模块寄存器分为三类：

- RTCC 控制寄存器
- RTCC 值寄存器
- 闹钟值寄存器

### 22.2.1 寄存器映射

要限制寄存器接口，则需通过相应的寄存器指针访问 RTCC 定时器和闹钟时间寄存器。RTCC 值寄存器窗口 (RTCVALH 和 RTCVALL) 使用 RTCPTR 位 (RCFGCAL<9:8>) 来选择需要的定时器寄存器对 (见表 22-1)。

通过写 RTCVALH 字节，使 RTCC 指针值 (RTCPTR<1:0> 位) 逐次递减 1 直至 00。一旦其达到 00，在手动更改指针值前，可通过 RTCVALH 和 RTCVALL 来访问 MINUTES 和 SECONDS 的值。

表 22-1： RTCVAL 寄存器映射

RTCPTR<1:0>	RTCC 值寄存器窗口	
	RTCVAL<15:8>	RTCVAL<7:0>
00	MINUTES	SECONDS
01	WEEKDAY	HOURS
10	MONTH	DAY
11	—	YEAR

闹钟值寄存器窗口 (ALRMVALH 和 ALRMVALL) 使用 ALRMPTR 位 (ALCFGGRPT<9:8>) 来选择所需的闹钟寄存器对 (见表 22-2)。

通过写 ALRMVALH 字节，使闹钟指针值 (ALRMPTR<1:0> 位) 逐次递减直至 00，一旦其达到 00，在手动更改指针值前，可通过 ALRMVALH 和 ALRMVALL 来访问 ALRMMIN 和 ALRMSEC 的值。

### 例 22-1： 将 RTCWREN 位置 1

```
asm volatile("push w7");
asm volatile("push w8");
asm volatile("disi #5");
asm volatile("mov #0x55, w7");
asm volatile("mov w7, _NVMKEY");
asm volatile("mov #0xAA, w8");
asm volatile("mov w8, _NVMKEY");
asm volatile("bset _RCFGCAL1, #13"); //set the RTCWREN bit
asm volatile("pop w8");
asm volatile("pop w7");
```

表 22-2： ALRMVAL 寄存器映射

ALRMPTR <1:0>	闹钟值寄存器窗口	
	ALRMVAL<15:8>	ALRMVAL<7:0>
00	ALRMMIN	ALRMSEC
01	ALRMWD	ALRMHR
10	ALRMMNTH	ALRMDAY
11	—	—

考虑到 16 位内核不能区分 8 位和 16 位读操作，用户必须注意读 ALRMVALH 或 ALRMVALL 字节会递减 ALRMPTR<1:0> 的值。同样，读 RTCVALH 或 RTCVALL 字节会使 RTCPTR<1:0> 的值递减。

**注：** 这仅适用于读操作，而不适用于写操作。

### 22.2.2 写锁定

为了对任何 RTCC 定时器寄存器执行写操作，必须将 RTCWREN 位 (RCFGCAL1<13>) 置 1 (见例 22-1)。

**注：** 为避免意外写入定时器，建议在除要特意执行写操作之外的任何时候都保持 RTCWREN 位 (RCFGCAL1<13>) 清零。对于要使 RTCWREN 位置 1 的操作，由于在 55h/AA 序列和 RTCWREN 置 1 之间仅允许有一个指令周期的时间；因此，建议按照例 22-1 中的过程执行代码。

### 22.2.3 选择 RTCC 时钟源

可使用 RTCPWC 寄存器中的 RTCLK<1:0> 位选择 RTCC 模块的时钟源。这些位设置为 00 时，辅助振荡器 (SOSC) 用作参考时钟；这些位为 01 时，LPRC 用作参考时钟。当 RTCLK<1:0> = 10 和 11 时，外部电力线 (50 Hz 和 60 Hz) 用作时钟源。

## 22.3 寄存器

### 22.3.1 RTCC 控制寄存器

**寄存器 22-1: RCFGCAL: RTCC 校准 / 配置寄存器<sup>(1)</sup>**

R/W-0	U-0	R/W-0	R-0, HSC	R-0, HSC	R/W-0	R/W-0	R/W-0
RTCEN <sup>(2)</sup>	—	RTCWREN	RTCSYNC	HALFSEC <sup>(3)</sup>	RTCOE	RTC PTR1	RTC PTR0
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CAL7  | CAL6  | CAL5  | CAL4  | CAL3  | CAL2  | CAL1  | CAL0  |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注: HSC = 可由硬件置 1/ 清零的位

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>RTCEN:</b> RTCC 使能位 <sup>(2)</sup>
	1 = 使能 RTCC 模块
	0 = 禁止 RTCC 模块
bit 14	未实现: 读为 0
bit 13	<b>RTCWREN:</b> RTCC 值寄存器写使能位
	1 = RTCVALH 和 RTCVALL 寄存器可被用户写入
	0 = RTCVALH 和 RTCVALL 寄存器被锁定不允许用户写入
bit 12	<b>RTCSYNC:</b> RTCC 值寄存器读同步位
	1 = 由于下溢返回 RTCVALH、RTCVALL 和 ALCFGRPT 寄存器可能会在读取期间变化, 从而导致读到的数据无效。如果两次读取寄存器得到的数据相同, 则认为数据有效。
	0 = RTCVALH、RTCVALL 或 ALCFGRPT 寄存器可被读取, 而不用担心计满返回
bit 11	<b>HALFSEC:</b> 半秒状态位 <sup>(3)</sup>
	1 = 一秒的后半周期
	0 = 一秒的前半周期
bit 10	<b>RTCOE:</b> RTCC 输出使能位
	1 = 使能 RTCC 输出
	0 = 禁止 RTCC 输出
bit 9-8	<b>RTC PTR&lt;1:0&gt;:</b> RTCC 值寄存器窗口指针位
	当读 RTCVALH 和 RTCVALL 寄存器时, 指向相应的 RTCC 值寄存器; 每次读或写 RTCVALH 时, RTC PTR<1:0> 的值都会减 1, 直至 00。
	<b>RTCVAL&lt;15:8&gt;:</b>
	11 = 保留
	10 = MONTH
	01 = WEEKDAY
	00 = MINUTES
	<b>RTCVAL&lt;7:0&gt;:</b>
	11 = YEAR
	10 = DAY
	01 = HOURS
	00 = SECONDS

**注 1:** RCFGCAL 寄存器仅受到 POR 影响。

**2:** 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。

**3:** 该位是只读位。当写入 MINSEC 寄存器的低半部分时, 该位清 0。

# PIC24FJ128GA310 系列

---

---

## 寄存器 22-1： RCFGCAL： RTCC 校准 / 配置寄存器<sup>(1)</sup> (续)

bit 7-0      **CAL<7:0>**: RTC 漂移校准位

01111111 = 最大正调整；每 15 秒增加 127 次 RTC 时钟脉冲

■

■

01111111 = 最小正调整；每 15 秒增加 1 次 RTC 时钟脉冲

00000000 = 无调整

11111111 = 最小负调整，每 15 秒减少 1 次 RTC 时钟脉冲

■

■

10000000 = 最大负调整，每 15 秒减少 128 次 RTC 时钟脉冲

注 1: RCFGCAL 寄存器仅受到 POR 影响。

2: 仅当 RTCWREN = 1 时才允许写入 RTCEN 位。

3: 该位是只读位。当写入 MINSEC 寄存器的低半部分时，该位清 0。

## 寄存器 22-2: RTCPWC: RTCC 电源控制寄存器<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
PWCEN	PWCPOL	PWCPRE	PWSPRE	RTCLK1 <sup>(2)</sup>	RTCLK0 <sup>(2)</sup>	RTCOUT1	RTCOUT0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **PWCEN:** 电源控制使能位

1 = 使能电源控制

0 = 禁止电源控制

bit 14      **PWCPOL:** 电源控制的极性控制位

1 = 高电压有效

0 = 低电压有效

bit 13      **PWCPRE:** 电源控制 / 稳定性预分频比位

1 = PWC 稳定性窗口时钟为源 RTCC 时钟的 2 分频

0 = PWC 稳定性窗口时钟为源 RTCC 时钟的 1 分频

bit 12      **PWSPRE:** 电源控制采样预分频比位

1 = PWC 采样窗口时钟为源 RTCC 时钟的 2 分频

0 = PWC 采样窗口时钟为源 RTCC 时钟的 1 分频

bit 11-10    **RTCLK<1:0>:** RTCC 时钟源选择位<sup>(2)</sup>

11 = 外部电力线 (60 Hz)

10 = 外部电力线源 (50 Hz)

01 = 内部 LPRC 振荡器

00 = 外部辅助振荡器 (SOSC)

bit 9-8      **RTCOUT<1:0>:** RTCC 输出源选择位

11 = 电源控制

10 = RTCC 时钟

01 = RTCC 秒时钟

00 = RTCC 阔钟脉冲

bit 7-0      未实现: 读为 0

**注 1:** RTCPWC 寄存器仅受到 POR 影响。

**2:** 当有新值写入这些寄存器位时, 还应写入 MINSEC 寄存器的低半部分, 以正确复位 RTCC 中的时钟预分频器。

# PIC24FJ128GA310 系列

寄存器 22-3： ALCFGRPT： 闹钟配置寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ALRMEN	CHIME	AMASK3	AMASK2	AMASK1	AMASK0	ALRMPTR1	ALRMPTR0
bit 15	bit 8						

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| ARPT7 | ARPT6 | ARPT5 | ARPT4 | ARPT3 | ARPT2 | ARPT1 | ARPT0 |
| bit 7 | bit 0 |       |       |       |       |       |       |

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **ALRMEN:** 闹钟使能位  
1 = 使能闹钟（只要 ARPT<7:0> = 00h 且 CHIME = 0，就会在闹钟事件后自动清零）  
0 = 禁止闹钟
- bit 14      **CHIME:** 响铃使能位  
1 = 使能响铃；允许 ARPT<7:0> 位从 00h 返回到 FFh  
0 = 禁止响铃；ARPT<7:0> 位到达 00h 时停止
- bit 13-10    **AMASK<3:0>:** 闹钟掩码配置位  
0000 = 每半秒  
0001 = 每秒  
0010 = 每 10 秒  
0011 = 每分钟  
0100 = 每 10 分钟  
0101 = 每小时  
0110 = 一天一次  
0111 = 一周一次  
1000 = 一月一次  
1001 = 每年一次（除配置为 2 月 29 日每 4 年一次外）  
101x = 保留 —— 不要使用  
11xx = 保留 —— 不要使用
- bit 9-8      **ALRMPTR<1:0>:** 闹钟值寄存器窗口指针位  
当读 ALRMVALH 和 ALRMVALL 寄存器时指向相应的闹钟值寄存器；每次读或写 ALRMVALH 时，ALRMPTR<1:0> 的值都会减 1，直至 00。  
ALRMVAL<15:8>:  
00 = ALRMMIN  
01 = ALRMWD  
10 = ALRMMNTH  
11 = PWCSTAB  
ALRMVAL<7:0>:  
00 = ALRMSEC  
01 = ALRMHR  
10 = ALRMDAY  
11 = PWCSAMP
- bit 7-0      **ARPT<7:0>:** 闹钟重复计数器值位  
11111111 = 闹钟将重复 255 次  
·  
·  
·  
00000000 = 闹钟不重复  
每发生一次闹钟事件，计数器就减 1；只有在 CHIME = 1 的情况下，计数器才会在递减到 00h 后返回 FFh。

## 22.3.2 RTCVAL 寄存器映射

**寄存器 22-4: YEAR: 年值寄存器<sup>(1)</sup>**

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

| R/W-x  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| YRTEN3 | YRTEN2 | YRTEN2 | YRTEN1 | YRONE3 | YRONE2 | YRONE1 | YRONE0 |
| bit 7  |        |        |        |        |        |        | bit 0  |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8 未实现: 读为 0

bit 7-4 **YRTEN<3:0>**: 年的十位数字的 BCD 值位  
包含从 0 到 9 的值。

bit 3-0 **YRONE<3:0>**: 年的个位数字的 BCD 值位  
包含从 0 到 9 的值。

**注 1:** 仅当 RTCWREN = 1 时才允许写入 YEAR 寄存器。

**寄存器 22-5: MTHDY: 月和日值寄存器<sup>(1)</sup>**

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13 未实现: 读为 0

bit 12 **MTHTEN0**: 月的十位数字的 BCD 值位  
包含为 0 或 1 的值。

bit 11-8 **MTHONE<3:0>**: 月的个位数字的 BCD 值位  
包含从 0 到 9 的值。

bit 7-6 未实现: 读为 0

bit 5-4 **DAYTEN<1:0>**: 日的十位数字的 BCD 值位  
包含从 0 到 3 的值。

bit 3-0 **DAYONE<3:0>**: 日的个位数字的 BCD 值位  
包含从 0 到 9 的值。

**注 1:** 仅当 RTCWREN = 1 时才允许写入该寄存器。

# PIC24FJ128GA310 系列

## 寄存器 22-6: WKDYHR: 星期和小时值寄存器<sup>(1)</sup>

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10-8 **WDAY<2:0>**: 星期数字的 BCD 值位  
包含从 0 到 6 的值。

bit 7-6 未实现: 读为 0

bit 5-4 **HRTEN<1:0>**: 小时的十位数字的 BCD 值位  
包含从 0 到 2 的值。

bit 3-0 **HRONE<3:0>**: 小时的个位数字的 BCD 值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

## 寄存器 22-7: MINSEC: 分钟和秒值寄存器

U-0	R/W-x						
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x						
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **MINTEN<2:0>**: 分钟的十位数字的 BCD 值位  
包含从 0 到 5 的值。

bit 11-8 **MINONE<3:0>**: 分钟的个位数字的 BCD 值位  
包含从 0 到 9 的值。

bit 7 未实现: 读为 0

bit 6-4 **SECTEN<2:0>**: 秒的十位数字的 BCD 值位  
包含从 0 到 5 的值。

bit 3-0 **SECONE<3:0>**: 秒的个位数字的 BCD 值位  
包含从 0 到 9 的值。

### 22.3.3 ALRMVAL 寄存器映射

**寄存器 22-8: ALMTHDY: 阵钟月和日值寄存器<sup>(1)</sup>**

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-13 未实现: 读为 0
- bit 12 **MTHTEN0:** 月的十位数字的 BCD 值位  
包含为 0 或 1 的值。
- bit 11-8 **MTHONE<3:0>:** 月的个位数字的 BCD 值位  
包含从 0 到 9 的值。
- bit 7-6 未实现: 读为 0
- bit 5-4 **DAYTEN<1:0>:** 日的十位数字的 BCD 值位  
包含从 0 到 3 的值。
- bit 3-0 **DAYONE<3:0>:** 日的个位数字的 BCD 值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

**寄存器 22-9: ALWDHHR: 阵钟星期和小时值寄存器<sup>(1)</sup>**

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-11 未实现: 读为 0
- bit 10-8 **WDAY<2:0>:** 星期数字的 BCD 值位  
包含从 0 到 6 的值。
- bit 7-6 未实现: 读为 0
- bit 5-4 **HRTEN<1:0>:** 小时的十位数字的 BCD 值位  
包含从 0 到 2 的值。
- bit 3-0 **HRONE<3:0>:** 小时的个位数字的 BCD 值位  
包含从 0 到 9 的值。

注 1: 仅当 RTCWREN = 1 时才允许写入该寄存器。

# PIC24FJ128GA310 系列

寄存器 22-10: ALMINSEC: 闹钟分钟和秒值寄存器

U-0	R/W-x						
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15	bit 8						

U-0	R/W-x						
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-12 **MINTEN<2:0>**: 分钟的十位数字的 BCD 值位  
包含从 0 到 5 的值。

bit 11-8 **MINONE<3:0>**: 分钟的个位数字的 BCD 值位  
包含从 0 到 9 的值。

bit 7 未实现: 读为 0

bit 6-4 **SECTEN<2:0>**: 秒的十位数字的 BCD 值位  
包含从 0 到 5 的值。

bit 3-0 **SECONE<3:0>**: 秒的个位数字的 BCD 值位  
包含从 0 到 9 的值。

## 寄存器 22-11： RTCCSWT： 电源控制和采样窗口定时器寄存器<sup>(1)</sup>

| R/W-x    |
|----------|----------|----------|----------|----------|----------|----------|----------|
| PWCSTAB7 | PWCSTAB6 | PWCSTAB5 | PWCSTAB4 | PWCSTAB3 | PWCSTAB2 | PWCSTAB1 | PWCSTAB0 |
| bit 15   | bit 8    |          |          |          |          |          |          |

| R/W-x                   |
|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| PWCSAMP7 <sup>(2)</sup> | PWCSAMP6 <sup>(2)</sup> | PWCSAMP5 <sup>(2)</sup> | PWCSAMP4 <sup>(2)</sup> | PWCSAMP3 <sup>(2)</sup> | PWCSAMP2 <sup>(2)</sup> | PWCSAMP1 <sup>(2)</sup> | PWCSAMP0 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-8      **PWCSTAB<7:0>**: 电源控制稳定性窗口定时器位

11111111 = 稳定性窗口为 255 个 TPWCCLK 时钟周期

11111110 = 稳定性窗口为 254 个 TPWCCLK 时钟周期

...

00000001 = 稳定性窗口为 1 个 TPWCCLK 时钟周期

00000000 = 无稳定性窗口；采样窗口在闹钟事件触发时启动

bit 7-0      **PWCSAMP<7:0>**: 电源控制采样窗口定时器位<sup>(2)</sup>

11111111 = 即使当 PWCEN = 0 时，仍始终使能采样窗口

11111110 = 采样窗口为 254 个 TPWCCLK 时钟周期

...

00000001 = 采样窗口为 1 个 TPWCCLK 时钟周期

00000000 = 无采样窗口

注 1： 仅当 RTCWREN = 1 时才允许写入该寄存器。

2： 当稳定性窗口定时器超时后，始终会启动采样窗口，除非其初始值为 00h。

## 22.4 校准

实时晶振输入可使用周期性自动调整功能来校准。正确校准后，RTCC 可实现每月小于 3 秒的误差。这可通过确定误差时钟脉冲数并将其存储到 RCFGCAL 寄存器的低半字节完成。装入到 RCFGCAL 低半字节的 8 位有符号值与 4 相乘，然后每分钟与 RTCC 定时器的值相加或相减一次。请参见以下步骤校准 RTCC：

1. 使用器件上的其他定时器资源，用户必须确定 32.768 kHz 晶振的误差。
2. 一旦误差已知，必须将其转换为每分钟的误差时钟脉冲数。
3. a) 如果振荡器频率高于理想值（步骤 2 中的计算结果为负），那么 RCFGCAL 寄存器的值就必须为负。这样每分钟会在定时器计数值上加上指定的时钟脉冲数。  
b) 如果振荡器频率低于理想值（步骤 2 中的计算结果为正），那么 RCFGCAL 寄存器的值就必须为正。这样每分钟会在定时器计数值上加上指定的时钟脉冲数。

### 公式 22-1:

$$(\text{理想频率 } \dagger - \text{测得频率}) * 60 = \text{每分钟的时钟数}$$

$$\dagger \text{ 理想频率} = 32,768 \text{ Hz}$$

仅当定时器关闭或者紧跟在秒脉冲的上升沿后才可写入 RCFGCAL 寄存器的低半字节，除非 SECONDS = 00、15、30 或 45。这是因为 RTCC 自动调整的间隔为 15 秒。

**注：** 用户决定误差值是否包含晶振初始误差  
(由温度和晶体老化所引起的漂移)。

## 22.5 闹钟

- 闹钟的时间间隔可配置为从半秒到一年
- 使用 ALRMEN 位 (ALCFGRPT<15>) 使能
- 可选择一次性闹钟和重复闹钟

### 22.5.1 配置闹钟

使用 ALRMEN 位使能闹钟功能。闹钟事件发生后该位清零。只有在 ALRMEN = 0 时才允许写入 ALRMVAL。

如图 22-2 所示，可通过 AMASK 位 (ALCFGRPT<13:10>) 配置闹钟时间间隔。这些位确定要触发闹钟，闹钟的哪些位、多少位必须和时钟值匹配。

也可配置闹钟根据预配置的时间间隔重复。一旦使能了闹钟，其重复次数就会被存储到 ARPT<7:0> 位 (ALCFGRPT<7:0>)。当 ARPT 位的值等于 00h 且 CHIME 位 (ALCFGRPT<14>) 清零时，禁止重复功能，仅发生单次闹钟。通过将 FFh 装入 ARPT<7:0>，可使闹钟重复最多 255 次。

每发生一次闹钟事件，ARPT 位的值都会减 1。一旦该值达到 00h，就会发生最后一次闹钟事件，然后 ALRMEN 位自动清零，关闭闹钟。

如果 CHIME 位 = 1，闹钟可无限次重复发生。在这种情况下，当 ARPT 位的值达到 00h 时，不会禁止闹钟，而是返回 FFh 并继续无限次地计数。

### 22.5.2 闹钟中断

每一次闹钟事件都会产生一个中断。此外，还将输出一个频率为闹钟频率一半的闹钟脉冲序列。该输出与 RTCC 时钟完全同步，且可用作其他外设的触发时钟。

**注：** 当使能闹钟 (ALRMEN = 1) 时，更改除 RCFGCAL 和 ALCFGRPT 寄存器外的任何寄存器或 CHIME 位都将导致错误的闹钟事件，从而引起错误的闹钟中断。为避免错误闹钟事件，只能在禁止闹钟 (ALRMEN = 0) 时更改定时器和闹钟值。建议在 RTCSYNC = 0 时更改 ALCFGRPT 寄存器和 CHIME 位。

图 22-2：闹钟屏蔽设置

闹钟掩码设置 (AMASK<3:0>)		星期几	月	日	小时	分钟	秒	
0000 - 每半秒		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	
0001 - 每秒		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	
0010 - 每 10 秒		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/> s	
0011 - 每分钟		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> <input type="checkbox"/> s s	
0100 - 每 10 秒		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> m	: <input type="checkbox"/> <input type="checkbox"/> s s	
0101 - 每小时		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	: <input type="checkbox"/> m m	: <input type="checkbox"/> <input type="checkbox"/> s s	
0110 - 每日		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h	: <input type="checkbox"/> m m	: <input type="checkbox"/> <input type="checkbox"/> s s	
0111 - 每周		<input type="checkbox"/> d	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> h	: <input type="checkbox"/> m m	: <input type="checkbox"/> <input type="checkbox"/> s s	
1000 - 每月		<input type="checkbox"/>	<input type="checkbox"/> <input type="checkbox"/>	/ <input type="checkbox"/> d	<input type="checkbox"/> d	<input type="checkbox"/> h	: <input type="checkbox"/> m m	: <input type="checkbox"/> <input type="checkbox"/> s s
1001 - 每年 <sup>(1)</sup>		<input type="checkbox"/>	<input type="checkbox"/> m	/ <input type="checkbox"/> d	<input type="checkbox"/> d	<input type="checkbox"/> h	: <input type="checkbox"/> m m	: <input type="checkbox"/> <input type="checkbox"/> s s

注 1：每年，除非配置为 2 月 29 日。

## 22.6 电源控制

RTCC 包含电源控制功能，允许器件定期唤醒外部器件，等待器件稳定后从该器件采样唤醒事件，然后关闭外部器件。这可完全由 RTCC 自行完成，无需从当前的低功耗模式（休眠、深度休眠等）唤醒。

要使用该功能：

- 使能 RTCC (RTCEN = 1)。
- 将 PWCEN 位 (RTCPWC<15>) 置 1。
- 将 RTCC 引脚配置为驱动 PWC 控制信号 (RTCOE = 1 且 RTCOUT<1:0> = 11)。

可使用 PWCPOL 位 (RTCPWC<14>) 选择 PWC 控制信号的极性。低电平有效信号或高电平有效信号可以与相应的外部开关配合使用，以接通或关断一个或多个外部器件的电源。低电平有效设置还可以与 RTCC 引脚的漏极开路结合使用，以直接驱动外部器件的接地引脚（通过相应的外部 VDD 上拉器件），无需使用外部开关。最后，CHIME 位应置 1，以使能 PWC 周期性。

## 22.7 RTCC 的 VBAT 工作模式

当 VDD 引脚断电时，RTCC 可工作在 VBAT 模式下。如果 VBAT 引脚接通电源（通常连接到电池），RTCC 将继续工作。

**注：**如果不使用 VBAT 模式（未与电池连接），建议将 VBAT 引脚连接到 VDD。

VBAT BOR 可以通过 CW3 配置寄存器中的 VBTBOR 位 (CW3<7>) 来使能/禁止。如果 VBTBOR 使能位清零，则 VBAT BOR 将始终禁止，并且不会有任何关于 VBAT BOR 的指示。如果 VBTBOR 位置 1，RTCC 可收到一个复位，并且 RTCEN 位将清零；这可在 1.95-1.4V（典型值）之间的任意点发生。

# **PIC24FJ128GA310 系列**

---

---

注:

## 23.0 32 位可编程循环冗余校验 (CRC) 发生器

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第41章“32 位可编程循环冗余校验 (CRC)” (39729A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

32 位可编程 CRC 发生器为各种网络和安全应用提供快速产生校验和的硬件实现方法。它提供以下特性:

- 用户可编程的 CRC 多项式方程 (最多 32 位)
- 可编程移位方向 (小尾数或大尾数格式)
- 独立的数据和多项式长度
- 可配置中断输出
- 数据 FIFO

CRC 发生器的简化框图如图 23-1 所示。图 23-2 中给出了 CRC 移位引擎的简化形式。

图 23-1: CRC 框图

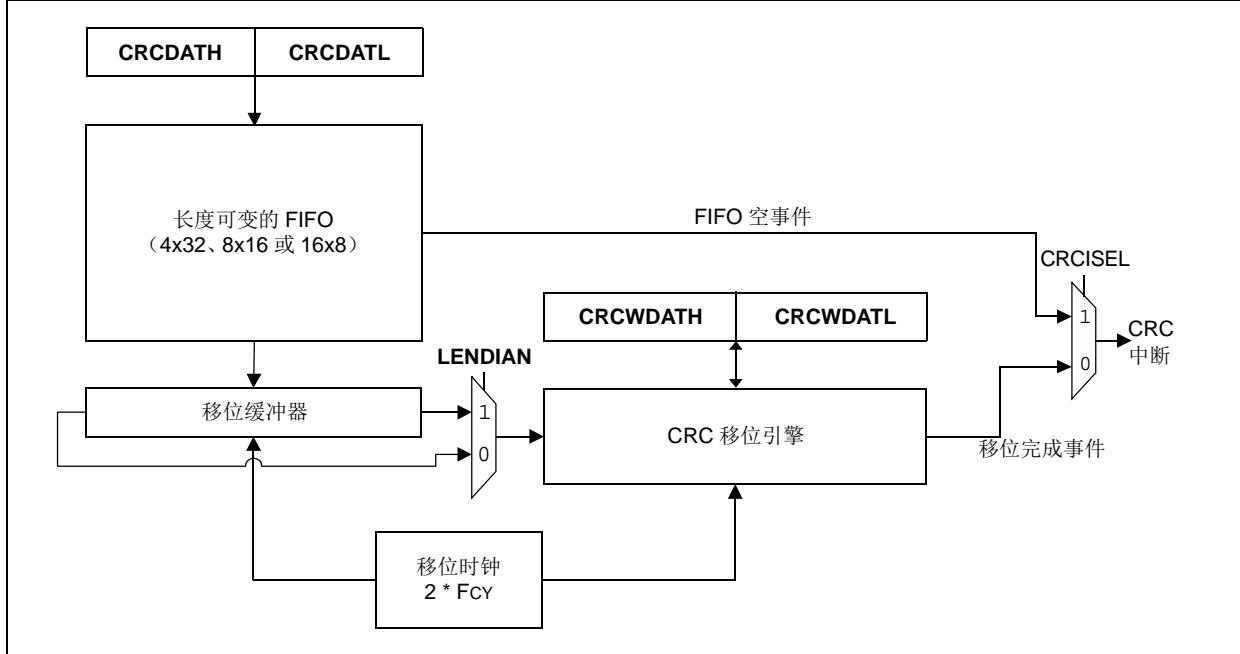
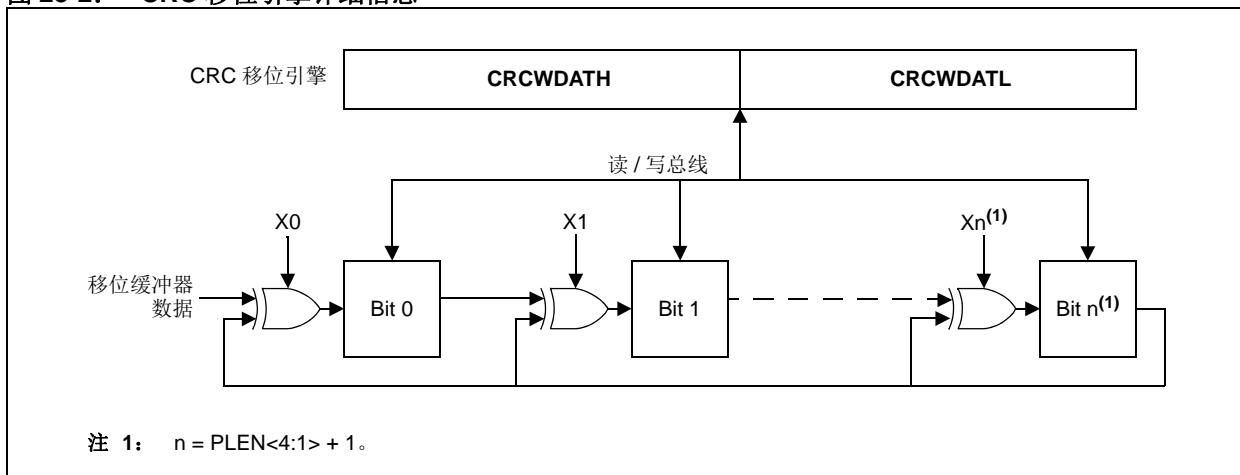


图 23-2: CRC 移位引擎详细信息



## 23.1 用户接口

### 23.1.1 多项式接口

可使用最多 32 个位，将 CRC 模块编程为产生至多 32 阶的 CRC 多项式。

通过 PLEN<4:0> 位 (CRCCON2<4:0>) 选择反映方程中最高次幂的多项式长度。

CRCXORL 和 CRCXORH 寄存器控制方程中包含的幂项。将某个特定位置 1 将会在方程中包含相应的幂项。在功能上相当于与 CRC 引擎的相应位进行异或操作。清零该位会禁止此异或操作。

例如，考虑两个 CRC 多项式，一个是 16 位方程，另一个是 32 位方程：

公式 23-1： 16 位和 32 位多项式

$$X^{16} + X^{12} + X^5 + 1$$

和

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

要将这些多项式编程到 CRC 发生器中，请将表 23-1 中的寄存器位置 1。

注意，请将适当的位置 1 以表示方程中将使用相应的项（例如 X<sub>26</sub> 和 X<sub>23</sub>）。因为方程的 0 位始终参与逻辑异或运算，因此 X<sub>0</sub> 位是无关位。对于长度为 32 的多项式，假设将使用第 32 位。因此，X<31:1> 位没有第 32 位。

### 23.1.2 数据接口

CRC 模块具有可缓存各种数据宽度的 FIFO。可使用 DWIDTH<4:0> 位 (CRCCON2<12:8>) 将输入数据宽度配置为 1 到 32 位之间的任何数值。当数据宽度大于 15 时，FIFO 为 4 字深。当 DWIDTH 值在 8 至 15 之间时，FIFO 为 8 字深。当 DWIDTH 值小于 8 时，FIFO 为 16 字深。

首先将要进行 CRC 计算所需的数据写入 FIFO。即使数据宽度小于 8，可被写入 FIFO 的最小数据元素也是 1 个字节。例如，如果 DWIDTH 位为 5，则数据大小为 DWIDTH<4:0>+1 或 6。数据作为整个字节写入，CRC 模块忽略了两个未使用的高位。

数据一旦被写入 CRCDAT 寄存器的最高位（由数据宽度定义），VWORD<4:0> 位 (CRCCON<12:8>) 的值将递增 1。例如，如果 DWIDTH 位为 24，则写入 CRCDATH 的 bit 7 时，VWORD 位递增 1。因此，CRCDATL 必须始终先于 CRCDATH 写入。

当 CRCGO 位置 1 且 VWORD 位的值大于零时，CRC 引擎开始移入数据。

每个字都被从 FIFO 复制到缓冲寄存器（这会导致 VWORD 位减 1）。然后从缓冲器中移出数据。CRC 引擎会以每指令周期两个位的速率继续移入数据直到 VWORD 位达到 0。因此对于给定的数据宽度，每个字完成计算所需的指令周期数为数据宽度的一半。例如，完成 32 位数据单字的 CRC 计算需要 16 个周期。

当 VWORD 位达到所配置 DWIDTH 位的最大值（4、8 或 16）时，CRCFUL 位置 1。当 VWORD 位达到零时，CRCMPT 位置 1。当 CRCEN 为 0 时，FIFO 为空且 VWORD<4:0> 设置为 00000。

在写入 CRCWDAT 之后，必须至少经过一个指令周期才可以读 VWORD 位。

表 23-1： CRC 设置示例（16 和 32 位多项式）

CRC 控制位	位值	
	16 位多项式	32 位多项式
PLEN<4:0>	01111	11111
X<31:16>	0000 0000 0000 0001	0000 0100 1100 0001
X<15:0>	0001 0000 0010 000X	0001 1101 1011 011x

### 23.1.3 数据移位方向

LENDIAN位（CRCCON1<3>）用于控制移位方向。默认情况下，CRC 通过引擎从最高有效位开始移入数据。将 LENDIAN 置 1 (= 1) 可导致 CRC 从最低有效位开始移入数据。此设置可使 CRC 与各种通信机制更好地集成，并省去了用软件使位反序的开销。注意，这只是改变了将数据移入引擎的方向。CRC 计算的结果仍然是正常的 CRC 结果，不是反序的 CRC 结果。

### 23.1.4 中断操作

用户可以配置该模块在以下两种条件下产生中断。

如果 CRCISEL 为 0，当 VWORD<4:0> 位的值从 1 跳变到 0 时产生中断。如果 CRCISEL 为 1，在 CRC 操作完成且模块将 CRCGO 位设置为 0 时产生中断。手动将 CRCGO 设置为 0 不会产生中断。请注意，产生中断时，CRC 计算尚未完成。产生中断后，模块将仍需要 (PLEN + 1)/2 个时钟周期，才会完成 CRC 计算。

### 23.1.5 典型操作

要使用模块进行典型 CRC 计算，请执行以下步骤：

1. 将 CRCEN 位置 1 以使能模块。
2. 配置模块以实现所需的操作：
  - a) 使用 CRCXORL 和 CRCXORH 寄存器以及 PLEN<4:0> 位对所需的多项式编程。
  - b) 使用 DWIDTH 和 LENDIAN 位配置数据宽度和移位方向。
  - c) 使用 CRCISEL 位选择所需的中断模式。

3. 在 CRCFUL 位置 1 之前或 FIFO 中无数据时，通过写入 CRCDATL 和 CRCDATH 寄存器来预装载 FIFO。
4. 通过写入 00h 到 CRCWDATL 和 CRCWDATH 来清除旧数据。CRCWDAT 寄存器也可以保持不变以继续上一个暂停的计算。
5. 将 CRCGO 位置 1 以启动计算。
6. 在 FIFO 有可用空间时将剩余数据写入 FIFO 中。
7. 计算完成时，CRCGO 自动清零。如果 CRCISEL = 1，则将产生中断。
8. 读取 CRCWDATL 和 CRCWDATH 以获得计算结果。

有 8 个寄存器用于控制可编程 CRC 操作：

- CRCCON1
- CRCCON2
- CRCXORL
- CRCXORH
- CRCDATL
- CRCDATH
- CRCWDATL
- CRCWDATH

CRCCON1 和 CRCCON2 寄存器（[寄存器 23-1](#) 和 [寄存器 23-2](#)）用于控制模块的操作和配置各种设置。

CRCXOR 寄存器（[寄存器 23-3](#) 和 [寄存器 23-4](#)）用于选择 CRC 方程中要使用的多项式项。CRCDAT 和 CRCWDAT 寄存器是分别用作双字输入数据和 CRC 处理输出的缓冲器的寄存器对。

# PIC24FJ128GA310 系列

寄存器 23-1：CRCCON1：CRC 控制寄存器 1

R/W-0	U-0	R/W-0	R-0, HSC				
CRCEN	—	CSIDL	VWORD4	VWORD3	VWORD2	VWORD1	VWORD0
bit 15	bit 8						

R-0, HSC	R-1, HSC	R/W-0	R/W-0, HC	R/W-0	U-0	U-0	U-0
CRCFUL	CRCMPT	CRCISEL	CRCGO	LENDIAN	—	—	—
bit 7	bit 0						

图注：	HC = 可由硬件清零的位	HSC = 可由硬件置 1/ 清零的位
R = 可读位	W = 可写位	U = 未实现位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零

- bit 15      **CRCEN:** CRC 使能位  
1 = 使能模块  
0 = 禁止模块；所有状态机、指针和 CRCWDAT/CRCDAATH 寄存器都复位；其他 SFR 不复位
- bit 14      未实现：读为 0
- bit 13      **CSIDL:** 空闲模式 CRC 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  
指出 FIFO 中有效字的个数。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:** FIFO 满位  
1 = FIFO 为满  
0 = FIFO 未满
- bit 6        **CRCMPT:** FIFO 空位  
1 = FIFO 为空  
0 = FIFO 非空
- bit 5        **CRCISEL:** CRC 中断选择位  
1 = FIFO 空时中断；仍在通过 CRC 移位最后的数据字  
0 = 移位完成时中断且结果已计算好
- bit 4        **CRCGO:** 启动 CRC 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 = 数据字从最低有效位开始移入 CRC（小尾数格式）  
0 = 数据字从最高有效位开始移入 CRC（大尾数格式）
- bit 2-0      未实现：读为 0

## 寄存器 23-2: CRCCON2: CRC 控制寄存器 2

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 = 上电复位时的值

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)。

## 寄存器 23-3: CRCXORL: CRC 多项式异或操作寄存器, 低字节

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X15	X14	X13	X12	X11	X10	X9	X8
bit 15							bit 8

R/W-0	U-0						
X7	X6	X5	X4	X3	X2	X1	—
bit 7							bit 0

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-1 X<15:1>: 多项式的项  $X^n$  的异或操作使能位

bit 0 未实现: 读为 0

# PIC24FJ128GA310 系列

寄存器 23-4: CRCXORH: CRC 异或操作寄存器 (高字节)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
X31	X30	X29	X28	X27	X26	X25	X24
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| X23   | X22   | X21   | X20   | X19   | X18   | X17   | X16   |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0

**X<31:16>**: 多项式的项  $X^n$  的异或操作使能位

## 24.0 带阈值扫描功能的12位A/D转换器

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。有关 12 位 A/D 转换器的详细信息, 请参见《PIC24F 系列参考手册》第 51 章“带阈值检测功能的 12 位 A/D 转换器”(DS39739B\_CN)。

12 位 A/D 转换器具有以下主要特性:

- 逐次逼近寄存器 (Successive Approximation Register, SAR) 转换
- 转换速度最高可达 200 kspS
- 最多 32 个模拟输入通道 (内部和外部)
- 可选 10 位或 12 位 (默认) 转换分辨率
- 多个内部参考电压输入通道
- 外部参考电压输入引脚
- 单极型差分采样 / 保持 (S/H) 放大器
- 用以预先评估转换结果的自动阈值扫描和比较操作
- 可选择转换触发源
- 固定长度 (每通道一个字) 的可配置转换结果缓冲区
- 4 个结果对齐选项
- 中断产生可配置
- 采用间接地址生成的增强型 DMA 操作
- 可在 CPU 休眠和空闲模式下工作

12 位 A/D 转换器模块是早期 PIC24 器件中提供的 10 位模块的改进版本。它是逐次逼近寄存器 (SAR) 转换器, 相比之前所作的改进有: 12 位分辨率, 提供广泛的自动采样选项, 与其他模拟模块更紧密地集成, 以及具有可配置的结果缓冲区。

该模块还包含了一个特有的阈值检测功能, 让模块自身可以基于转换结果做出简单决定, 通过外设间接寻址 (Peripheral Indirect Addressing, PIA) 增强 DMA 控制器的操作。

图 24-1 给出了该模块的简化框图。

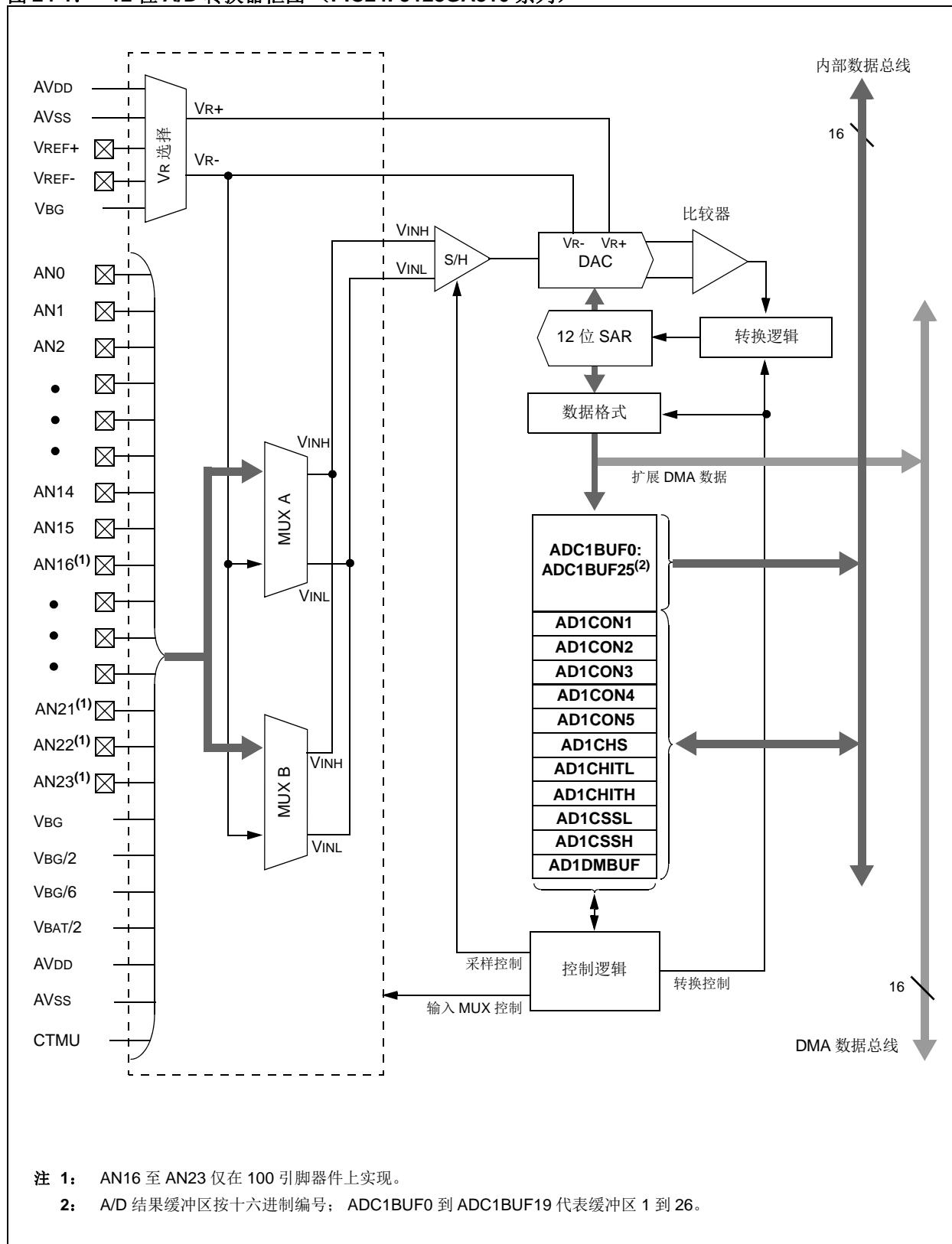
## 24.1 基本操作

要执行标准 A/D 转换:

1. 配置模块:
  - a) 通过将 ANSELn 寄存器的相应位置 1 将端口引脚配置为模拟输入 (详细信息请参见第 11.2 节“[配置模拟端口引脚 \(ANSx\)](#)”)
  - b) 选择能满足模拟输入预期范围的参考电压源 (AD1CON2<15:13>)。
  - c) 为每条通道选择正负多路开关输入 (AD1CHS<15:0>)。
  - d) 选择模拟转换时钟以便使期望的数据速率与处理器时钟匹配 (AD1CON3<7:0>)。
  - e) 选择适当的采样/转换序列 (AD1CON1<7:5> 和 AD1CON3<12:8>)。
  - f) 对于通道 A 扫描操作, 选择要包括的正通道 (AD1CSSH 和 AD1CSSL 寄存器)。
  - g) 选择转换结果在缓冲器中的存储方式 (AD1CON1<9:8> 和 AD1CON5 寄存器)。
  - h) 选择中断发生的频率 (AD1CON2<5:2>)。
  - i) 启动 A/D 模块 (AD1CON1<15>)。
2. 配置 A/D 中断 (如果需要):
  - a) 清零 AD1IF 位 (IFS0<13>)。
  - b) 使能 AD1IE 中断 (IEC0<13>)。
  - c) 选择 A/D 中断优先级 (IPC3<6:4>)。
3. 如果模块配置为手动采样, 将 SAMP 位 (AD1CON1<1>) 置 1 开始采样。

# PIC24FJ128GA310 系列

图 24-1: 12 位 A/D 转换器框图 (PIC24FJ128GA310 系列)



注 1: AN16 至 AN23 仅在 100 引脚器件上实现。

2: A/D 结果缓冲区按十六进制编号: ADC1BUF0 到 ADC1BUF19 代表缓冲区 1 到 26。

## 24.2 扩展 DMA 操作

除所有 12 位 A/D 转换器都具备的标准功能外，PIC24FJ128GA310 系列器件还有一定的 DMA 扩展功能。此项扩展添加了器件的 DMA 控制器配套功能，在模块内置缓冲区的基础上扩展了 A/D 模块的数据存储能力。

DMA 扩展功能由 DMAEN 位 (AD1CON1<10>) 控制；将此位置 1 将使能该功能。DMABM 位 (AD1CON1<11>) 配置 DMA 功能的工作方式。

### 24.2.1 扩展缓冲区模式

扩展缓冲区模式 (DMABM = 1) 对于转换上半部分通道（即 26 及以上）的存储转换结果非常有用，该部分通道在 A/D 模块内没有对应的存储器映射缓冲区。它还可用于在数据 RAM 的任意已实现的地址存储任意 A/D 通道的转换结果。

在扩展缓冲区模式下，来自 A/D 缓冲寄存器（以及 26 号通道及以上）的所有数据都将映射到数据 RAM。转换数据写入由 DMA 控制器（确切地说，由 DMADST 寄存器）指定的目标位置。这使用户能够从数据存储器读取 26 号及以上通道（自身没有存储器映射 A/D 缓冲单元）的转换结果。

使用扩展缓冲区模式时，始终将 BUFREGEN 位置 1 以禁止 FIFO 操作。此外，还应将 BUFM 位清零以禁止拆分缓冲区模式。

### 24.2.2 PIA 模式

DMABM = 0 时，A/D 模块配置为与 DMA 控制器协同进行外设间接寻址 (PIA) 模式操作。在此模式下，A/D 模块生成一个 11 位间接地址 (Indirect Address, IA)。这与 DMA 控制器中的目标地址进行或运算，以定义 A/D 转换数据的存储位置。

在 PIA 模式下，缓冲区空间创建为一系列连续的小缓冲区（每个模拟通道一个）。通道缓冲区的大小确定可容纳多少条模拟通道。通过 DMABL 位 (AD1CON4<2:0>) 选择缓冲区的大小。缓冲区大小的选择范围是 1 个字到 128 个字。每条通道都分配一个这种大小的缓冲区，无论通道是否真正具有转换数据。

通过将通道缓冲区内的基地址与三到五个位（取决于缓冲区大小）相结合来创建 IA 以标识通道。基地址的宽度范围是零到七位，具体取决于缓冲区大小。该地址在右侧填充 0，以便在数据空间中保持地址对齐。然后根据需要在连接而成的通道和基地址左侧填充零补全 11 位 IA。写操作期间使用 SMPI 位 (AD1CON2<6:2>) 将 IA 配置为自动递增。

和任何具有 DMA 功能的模块的 PIA 操作一样，DMADST 寄存器中的目标基址必须适当使用掩码以适应 IA。表 24-1 给出了完整地址的形成方式。请注意，地址掩码根据缓冲区大小的不同而不同。由于存在掩码要求，特定缓冲区大小可能无法使用某些地址范围。用户应验证 DMA 基址是否与所选缓冲区大小兼容。

图 24-2 显示了地址中各个部分对数据存储器中缓冲单元的定义方式。此例中，模块为 32 个 4 字缓冲区“分配”256 字节的数据 RAM (1000h 至 1100h)。但这并不是硬性分配，而且不能阻止这些单元用于其他用途。例如在此例中，如果模拟通道 1、3 和 8 正在进行采样和转换，转换数据只会写入通道缓冲区（分别从 1008h、1018h 和 1040h 开始）。PIA 缓冲区空间中的未使用部分可用于其他用途。用户应负责跟踪记录缓冲单元并防止数据覆盖。

## 24.3 使用 VBAT 的 A/D 操作

一条 A/D 通道连接到 VBAT 引脚监视 VBAT 电压。这能够在没有外部连接情况下监视 VBAT 引脚电压（电池电压）。使用 A/D VBAT 监视器测得的电压为 VBAT/2。可通过读取  $A/D = ((VBAT/2)/VDD) * 1024$ （对于 10 位 A/D）和  $((VBAT/2)/VDD) * 4096$ （对于 12 位 A/D）计算电压。

使用 VBAT A/D 监视器时：

- 将 A/D 通道接地以将采样电容放电。
- 由于 VBAT 的高阻态，应选择更长的采样时间以获取精确读数。

由于在采样期间 VBAT 引脚连接到 A/D，所以要延长 VBAT 电池寿命，建议在需要时选择 VBAT 通道。

# PIC24FJ128GA310 系列

## 24.4 寄存器

12 位 A/D 转换器通过总计 13 个寄存器进行控制：

- AD1CON1至AD1CON5（[寄存器 24-1](#)至[寄存器 24-5](#)）
- AD1CS（[寄存器 24-6](#)）
- AD1CHITH 和 AD1CHITL（[寄存器 24-8](#) 和 [寄存器 24-9](#)）

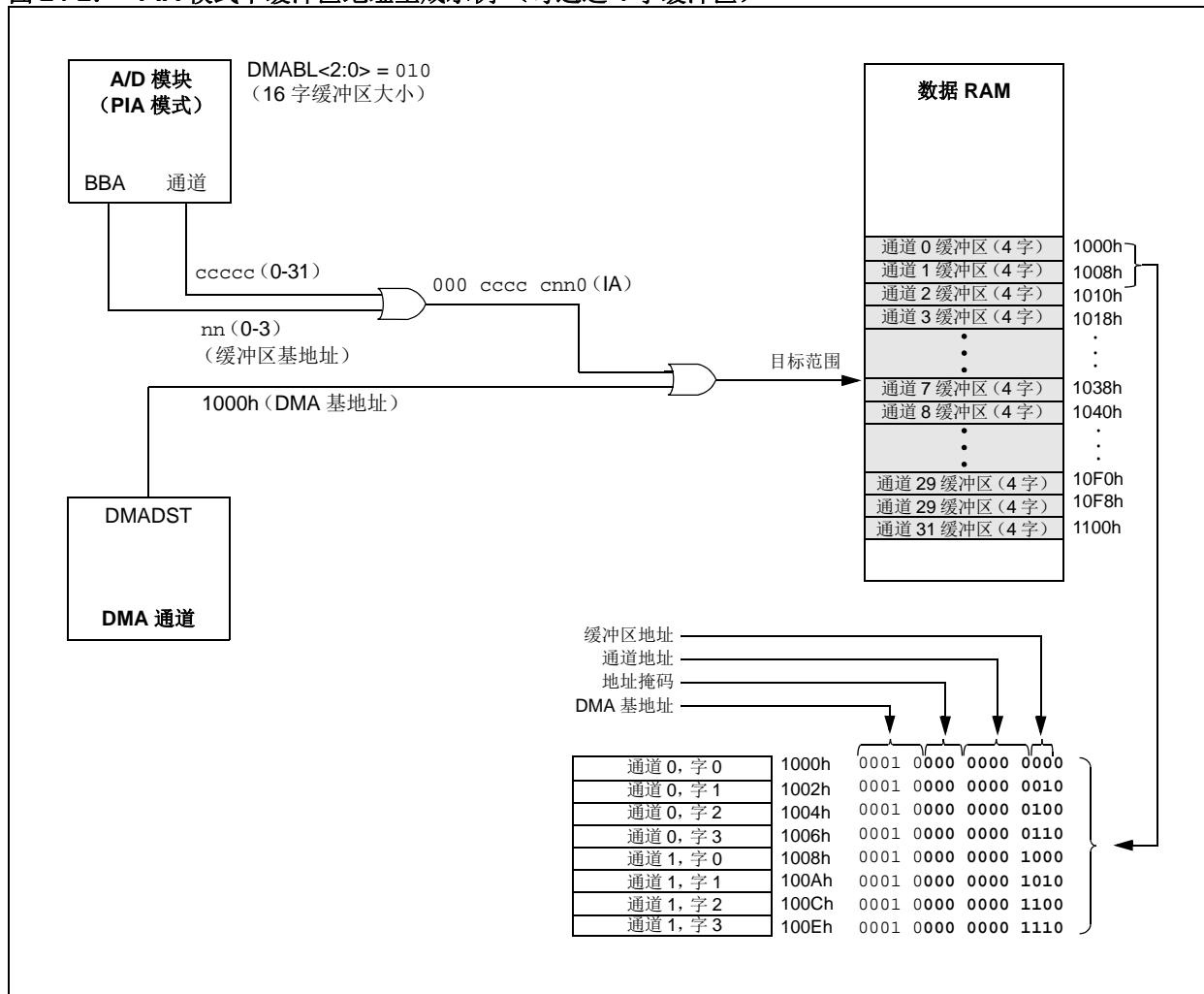
- AD1CSSH 和 AD1CSSL（[寄存器 24-10](#) 和 [寄存器 24-11](#)）
- AD1CTMENH 和 AD1CTMENL（[寄存器 24-12](#) 和 [寄存器 24-13](#)）
- AD1DMBUF（未显示）——扩展缓冲区模式的 16 位转换缓冲区

表 24-1：PIA 模式下的间接地址生成

DMABL<2:0>	每条通道的缓冲区大小（字）	生成的偏移地址（低 11 位）	可用输入通道	允许的 DMADST 地址
000	1	000 00cc ccc0	32	xxxxx xxxx xx00 0000
001	2	000 0ccc ccn0	32	xxxxx xxxx x000 0000
010	4	000 cccc cnn0	32	xxxxx xxxx 0000 0000
011	8	00c cccc nnn0	32	xxxxx xxx0 0000 0000
100	16	0cc cccn nnn0	32	xxxxx xx00 0000 0000
101	32	ccc ccnn nnn0	32	xxxxx x000 0000 0000
110	64	ccc cnnn nnn0	16	xxxxx x000 0000 0000
111	128	ccc nnnn nnn0	8	xxxxx x000 0000 0000

图注：  
ccc = 通道编号（三到五位），n = 缓冲区基地址（零到七位），  
x = 用户可为基地址定义的 DMADST 范围，0 = IA 的 DMADST 掩码位。

图 24-2: PIA 模式下缓冲区地址生成示例（每通道 4 字缓冲区）



# PIC24FJ128GA310 系列

寄存器 24-1：AD1CON1：A/D 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADON	—	ADSIDL	DMABM <sup>(1)</sup>	DMAEN	MODE12	FORM1	FORM0
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0, HCS	R/C-0, HCS
SSRC3	SSRC2	SSRC1	SSRC0	—	ASAM	SAMP	DONE
bit 7	bit 0						

图注：	U = 未实现位，读为 0
R = 可读位	W = 可写位
-n = 上电复位时的值	HSC = 可由硬件置 1/ 清零的位
	1 = 置 1
	0 = 清零
	x = 未知

- bit 15      **ADON:** A/D 工作模式位  
1 = A/D 转换器模块正在工作  
0 = A/D 转换器关闭
- bit 14      未实现：读为 0
- bit 13      **ADSIDL:** 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      **DMABM:** 扩展 DMA 缓冲区模式选择位<sup>(1)</sup>  
1 = 扩展缓冲区模式：缓冲区地址由 DMAAndDST 寄存器定义  
0 = PIA 模式：缓冲区地址由 DMA 控制器和 AD1CON4<2:0> 定义
- bit 11      **DMAEN:** 扩展 DMA/ 缓冲区使能位  
1 = 使能扩展 DMA 和缓冲区功能  
0 = 禁止扩展功能
- bit 10      **MODE12:** 12 位工作模式位  
1 = 12 位 A/D 工作  
0 = 10 位 A/D 工作
- bit 9-8      **FORM<1:0>:** 数据输出格式位（参见以下格式）  
11 = 小数结果，有符号，左对齐  
10 = 绝对小数结果，无符号，左对齐  
01 = 十进制结果，有符号，右对齐  
00 = 绝对十进制结果，无符号，右对齐
- bit 7-4      **SSRC<3:0>:** 采样时钟源选择位  
1xxxx = 未实现；不要使用  
0111 = 内部计数器结束采样并启动转换（自动转换）。不要在自动扫描模式下使用  
0110 = 未实现  
0101 = TMR1  
0100 = CTMU  
0011 = TMR5  
0010 = TMR3  
0001 = INT0  
0000 = SAMP 位必须用软件清零才能开始转换
- bit 3      未实现：读为 0
- bit 2      **ASAM:** A/D 采样自动启动位  
1 = 最后一次转换结束后立即开始采样；SAMP 位自动置 1  
0 = 手动将 SAMP 位置 1 时开始采样

注 1：只有在扩展 DMA/ 缓冲区功能可用（DMAEN = 1）时才可使用此位。

## 寄存器 24-1: AD1CON1: A/D 控制寄存器 1 (续)

bit 1	<b>SAMP:</b> A/D 采样使能位 1 = A/D 采样 / 保持放大器正在采样 0 = A/D 采样 / 保持放大器正在保持
bit 0	<b>DONE:</b> A/D 转换状态位 1 = A/D 转换已完成 0 = A/D 转换尚未开始或在进行

**注 1:** 只有在扩展 DMA/ 缓冲区功能可用 (DMAEN = 1) 时才可使用此位。

# PIC24FJ128GA310 系列

寄存器 24-2: AD1CON2: A/D 控制寄存器 2

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
PVCFG1	PVCFG0	NVCFG0	OFFCAL	BUFREGEN	CSCNA	—	—
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
BUFS <sup>(1)</sup>	SMPI4	SMPI3	SMPI2	SMPI1	SMPI0	BUFM <sup>(1)</sup>	ALTS
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-14 **PVCFG<1:0>**: 转换器正参考电压配置位

1x = 未实现; 不要使用

01 = 外部 VREF+

00 = AVDD

bit 13 **NVCFG0**: 转换器负参考电压配置位

1 = 外部 VREF-

0 = AVss

bit 12 **OFFCAL**: 失调校准模式选择位

1 = 采样 / 保持通道的反相和同相输入与 AVss 连接

0 = 采样 / 保持通道的反相和同相输入与一般输入连接

bit 11 **BUFREGEN**: A/D 缓冲寄存器使能位

1 = 转换结果装入根据转换通道确定的缓冲单元

0 = A/D 结果缓冲区视为 FIFO 缓冲区

bit 10 **CSCNA**: 采样 A 期间 CH0+ 的扫描输入选择位

1 = 扫描输入

0 = 不扫描输入

bit 9-8 未实现: 读为 0

bit 7 **BUFS**: 缓冲区填充状态位<sup>(1)</sup>

1 = A/D 正在填充缓冲区的上半部分; 用户应访问下半部分中的数据

0 = A/D 正在填充缓冲区的下半部分; 用户应访问上半部分中的数据

bit 6-2 **SMPI<4:0>**: 中断采样速率 /DMA 递增速率选择位

当 DMAEN = 1 时:

0001 = 2 通道 DMA A/D 操作

0000 = 1 通道 DMA A/D 操作

当 DMAEN = 0 时:

选择每次中断的采样 / 转换次数

11111 = 每转换完 32 个采样时产生中断 / 地址递增

11110 = 每转换完 31 个采样时产生中断 / 地址递增

• • •

00001 = 每转换完 2 个采样时产生中断 / 地址递增

00000 = 每转换完 1 个采样时产生中断 / 地址递增

注 1: 这些位仅在缓冲区采用 FIFO 模式 (BUFREGEN = 0) 时适用。此外, BUFS 仅在 BUFM = 1 时使用。

## 寄存器 24-2: AD1CON2: A/D 控制寄存器 2 (续)

bit 1	<b>BUFM:</b> 缓冲区填充模式选择位 <sup>(1)</sup>
	1 = A/D 缓冲区为 2 个 13 字缓冲区 (从 ADC1BUF0 和 ADC1BUF12 开始), 顺次转换交替填充缓冲区 (拆分模式)
	0 = A/D 缓冲区为 1 个 26 字缓冲区, 从 ADC1BUF0 开始按顺序填充 (FIFO 模式)
bit 0	<b>ALTS:</b> 交替输入采样模式选择位
	1 = 在第一次采样时使用采样多路开关 A 选择的输入通道, 而在下一次采样时使用采样多路开关 B 选择的输入通道
	0 = 总是使用采样多路开关 A 选择的输入通道

**注 1:** 这些位仅在缓冲区采用 FIFO 模式 (BUFREGEN = 0) 时适用。此外, BUFS 仅在 BUFM = 1 时使用。

## 寄存器 24-3: AD1CON3: A/D 控制寄存器 3

R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	EXTSAM	PUMPEN	SAMC4	SAMC3	SAMC2	SAMC1	SAMC0
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
ADCS7	ADCS6	ADCS5	ADCS4	ADCS3	ADCS2	ADCS1	ADCS0
bit 7							bit 0

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>ADRC:</b> A/D 转换时钟源位
	1 = RC 时钟
	0 = 时钟由系统时钟产生
bit 14	<b>EXTSAM:</b> 扩展采样时间位
	1 = A/D 在 SAMP = 0 之后仍然在采样
	0 = A/D 完成采样
bit 13	<b>PUMPEN:</b> 电荷泵使能位
	1 = 使能开关的电荷泵
	0 = 禁止开关的电荷泵
bit 12-8	<b>SAMC&lt;4:0&gt;:</b> 自动采样时间选择位
	11111 = 31 TAD
	•••
	00001 = 1 TAD
	00000 = 0 TAD
bit 7-0	<b>ADCS&lt;7:0&gt;:</b> A/D 转换时钟选择位
	11111111
	••• = 保留
	01000000
	00111111 = 64 · TCY = TAD
	•••
	00000001 = 2 · TCY = TAD
	00000000 = TCY = TAD

# PIC24FJ128GA310 系列

寄存器 24-4: AD1CON4: A/D 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	R/W-0	R/W-0	R/W-0
—	—	—	—	—	DMABL2 <sup>(1)</sup>	DMABL1 <sup>(1)</sup>	DMABL0 <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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-3 未实现: 读为 0

bit 2-0 **DMABL<2:0>**: DMA 缓冲区大小选择位 <sup>(1)</sup>

111 = 为每个模拟输入分配 128 字缓冲区

110 = 为每个模拟输入分配 64 字缓冲区

101 = 为每个模拟输入分配 32 字缓冲区

100 = 为每个模拟输入分配 16 字缓冲区

011 = 为每个模拟输入分配 8 字缓冲区

010 = 为每个模拟输入分配 4 字缓冲区

001 = 为每个模拟输入分配 2 字缓冲区

000 = 为每个模拟输入分配 1 字缓冲区

注 1: DMABL<2:0> 位只在 AD1CON1<11> = 1 且 AD1CON<12> = 0 时使用; 否则将值忽略。

## 寄存器 24-5：AD1CON5：A/D 控制寄存器 5

R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0
ASEN	LPEN	CTMREQ	BGREQ	—	—	ASINT1	ASINT0
bit 15 bit 8							

U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	WM1	WM0	CM1	CM0
bit 7 bit 0							

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15      **ASEN:** 自动扫描使能位  
               1 = 使能自动扫描  
               0 = 禁止自动扫描
- bit 14      **LPEN:** 低功耗使能位  
               1 = 扫描后使能低功耗  
               0 = 扫描后使能全功耗
- bit 13      **CTMREQ:** CTMU 请求位  
               1 = 当 A/D 使能并处于活动状态时使能 CTMU  
               0 = 不通过 A/D 使能 CTMU
- bit 12      **BGREQ:** 带隙请求位  
               1 = 当 A/D 使能并处于活动状态时使能带隙  
               0 = 不通过 A/D 使能带隙
- bit 11-10    未实现：读为 0
- bit 9-8      **ASINT<1:0>:** 自动扫描（阈值检测）中断模式位  
               11 = 在阈值检测序列完成且发生有效比较之后产生中断  
               10 = 在发生有效比较之后产生中断  
               01 = 在阈值检测序列完成之后产生中断  
               00 = 不产生中断
- bit 7-4      未实现：读为 0
- bit 3-2      **WM<1:0>:** 写模式位  
               11 = 保留  
               10 = 仅自动比较（不保存转换结果，但在发生由 CM 和 ASINT 位定义的有效匹配时产生中断）  
               01 = 转换并保存（当发生由 CM 位定义的匹配时，转换结果保存到由寄存器位决定的存储单元中）  
               00 = 传统操作（转换数据保存到由缓冲寄存器位决定的存储单元中）
- bit 1-0      **CM<1:0>:** 比较模式位  
               11 = 窗外模式（如果转换结果超出相应缓冲区对定义的窗口，则发生有效匹配）  
               10 = 窗内模式（如果转换结果处于相应缓冲区对定义的窗口中，则发生有效匹配）  
               01 = 大于模式（如果结果大于相应缓冲寄存器中的值，则发生有效匹配）  
               00 = 小于模式（如果结果小于相应缓冲寄存器中的值，则发生有效匹配）

# PIC24FJ128GA310 系列

## 寄存器 24-6: AD1CHS: A/D 采样选择寄存器

| R/W-0  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CH0NB2 | CH0NB1 | CH0NB0 | CH0SB4 | CH0SB3 | CH0SB2 | CH0SB1 | CH0SB0 |
| bit 15 |        |        |        |        |        |        | bit 8  |

| R/W-0  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CH0NA2 | CH0NA1 | CH0NA0 | CH0SA4 | CH0SA3 | CH0SA2 | CH0SA1 | CH0SA0 |
| bit 7  |        |        |        |        |        |        | bit 0  |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-13      **CH0NB<2:0>**: 采样多路开关 B 的通道 0 的反相输入选择位

1xx = 未实现

011 = 未实现

010 = AN1

001 = 未实现

000 = VREF-/AVSS

bit 12-8      **CH0SB<4:0>**: 采样多路开关 B 的通道 0 的同相输入选择位

11111 = VBAT/2<sup>(1)</sup>

11110 = AVDD<sup>(1)</sup>

11101 = AVss<sup>(1)</sup>

11100 = 带隙参考 (VBG) <sup>(1)</sup>

11011 = VBG/2<sup>(1)</sup>

11010 = VBG/6<sup>(1)</sup>

11001 = CTMU

11000 = CTMU 温度传感器输入 (不需要将 AD1CTMENH<8> 置 1)

10111 = AN23<sup>(2)</sup>

10110 = AN22<sup>(2)</sup>

10101 = AN21<sup>(2)</sup>

10100 = AN20<sup>(2)</sup>

10011 = AN19<sup>(2)</sup>

10010 = AN18<sup>(2)</sup>

10001 = AN17<sup>(2)</sup>

10000 = AN16<sup>(2)</sup>

01111 = AN15

01110 = AN14

01101 = AN13

01100 = AN12

01011 = AN11

01010 = AN10

01001 = AN9

01000 = AN8

00111 = AN7

00110 = AN6

00101 = AN5

00100 = AN4

00011 = AN3

00010 = AN2

00001 = AN1

00000 = AN0

注 1: 这些输入通道没有相应的存储器映射结果缓冲区。

2: 这些通道仅在 100 引脚器件中实现。

## 寄存器 24-6: AD1CHS: A/D 采样选择寄存器 (续)

- bit 7-5      **CH0NA<2:0>**: 采样多路开关 A 的通道 0 的反相输入选择位  
与 CH0NB<2:0> 定义相同。
- bit 4-0      **CH0SA<4:0>**: 采样多路开关 A 的通道 0 的同相输入选择位  
与 CH0SB<4:0> 定义相同。

**注 1:** 这些输入通道没有相应的存储器映射结果缓冲区。

**2:** 这些通道仅在 100 引脚器件中实现。

## 寄存器 24-7: ANCFG: A/D 带隙参考配置

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

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	VBG6EN	VBG2EN	VBGEN
bit 7							

### 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 15-3      未实现: 读为 0
- bit 2      **VBG6EN:** A/D 输入 VBG/6 使能位  
1 = 使能六分之一带隙参考电压 (VBG/6)  
0 = 禁止六分之一带隙参考电压 (VBG/6)
- bit 1      **VBG2EN:** A/D 输入 VBG/2 使能位  
1 = 使能二分之一带隙参考电压 (VBG/2)  
0 = 禁止二分之一带隙参考电压 (VBG/2)
- bit 0      **VBGEN:** A/D 输入 VBG 使能位  
1 = 使能带隙参考电压 (VBG)  
0 = 禁止带隙参考电压 (VBG)

# PIC24FJ128GA310 系列

寄存器 24-8: AD1CHITH: A/D 扫描比较命中寄存器 (高位字)

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CHH25	CHH24
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CHH23 | CHH22 | CHH21 | CHH20 | CHH19 | CHH18 | CHH17 | CHH16 |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10 未实现: 读为 0

bit 9-0 CHH<25:16>: A/D 比较命中位

如果 CM<1:0> = 11:

1 = A/D 结果缓冲区 n 已写入数据或已发生匹配

0 = A/D 结果缓冲区 n 尚未写入数据

对于 CM<1:0> 的所有其他值:

1 = A/D 结果通道 n 上发生了匹配

0 = A/D 结果通道 n 上未发生匹配

寄存器 24-9: AD1CHITL: A/D 扫描比较命中寄存器 (低位字)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHH15	CHH14	CHH13	CHH12	CHH11	CHH10	CHH9	CHH8
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CHH7  | CHH6  | CHH5  | CHH4  | CHH3  | CHH2  | CHH1  | CHH0  |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 CHH<15:0>: A/D 比较命中位

如果 CM<1:0> = 11:

1 = A/D 结果缓冲区 n 已写入数据或已发生匹配

0 = A/D 结果缓冲区 n 尚未写入数据

对于 CM<1:0> 的所有其他值:

1 = A/D 结果通道 n 上发生了匹配

0 = A/D 结果通道 n 上未发生匹配

## 寄存器 24-10: AD1CSSH: A/D 输入扫描选择寄存器 (高位字)

U-0	R/W-0						
—	CSS30	CSS29	CSS28	CSS27	CSS26	CSS25	CSS24
bit 15	bit 8						

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CSS23 | CSS22 | CSS21 | CSS20 | CSS19 | CSS18 | CSS17 | CSS16 |
| bit 7 | bit 0 |       |       |       |       |       |       |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 未实现: 读为 0

bit 14-0 **CSS<30:16>: A/D 输入扫描选择位**

1 = 输入扫描时包含相应的通道

0 = 输入扫描时跳过通道

## 寄存器 24-11: AD1CSSL: A/D 输入扫描选择寄存器 (低位字)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSS15	CSS14	CSS13	CSS12	CSS11	CSS10	CSS9	CSS8
bit 15	bit 8						

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CSS7  | CSS6  | CSS5  | CSS4  | CSS3  | CSS2  | CSS1  | CSS0  |
| bit 7 | bit 0 |       |       |       |       |       |       |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0 **CSS<15:0>: A/D 输入扫描选择位**

1 = 输入扫描时包含相应的通道

0 = 输入扫描时跳过通道

# PIC24FJ128GA310 系列

寄存器 24-12: AD1CTMENH: CTMU 使能寄存器 (高位字) (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
—	CTMEN30	CTMEN29	CTMEN28	CTMEN27	CTMEN26	CTMEN25	CTMEN24
bit 15	bit 8						

| R/W-0   |
|---------|---------|---------|---------|---------|---------|---------|---------|
| CTMEN23 | CTMEN22 | CTMEN21 | CTMEN20 | CTMEN19 | CTMEN18 | CTMEN17 | CTMEN16 |
| bit 7   | bit 0   |         |         |         |         |         |         |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-0      **CTMEN<31:16>**: 转换期间 CTMU 使能位

1 = 在转换期间使能 CTMU 并连接到选定的通道

0 = CTMU 不连接到该通道

注 1: 实际的可用通道数取决于特定器件上实现了哪些通道; 详情请参见器件数据手册。未实现的通道, 读为 0。

寄存器 24-13: AD1CTMENL: CTMU 使能寄存器 (低位字) (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
CTMEN15	CTMEN14	CTMEN13	CTMEN12	CTMUEN11	CTMEN10	CTMEN9	CTMEN8
bit 15	bit 8						

| R/W-0  |
|--------|--------|--------|--------|--------|--------|--------|--------|
| CTMEN7 | CTMEN6 | CTMEN5 | CTMEN4 | CTMEN3 | CTMEN2 | CTMEN1 | CTMEN0 |
| bit 7  | bit 0  |        |        |        |        |        |        |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

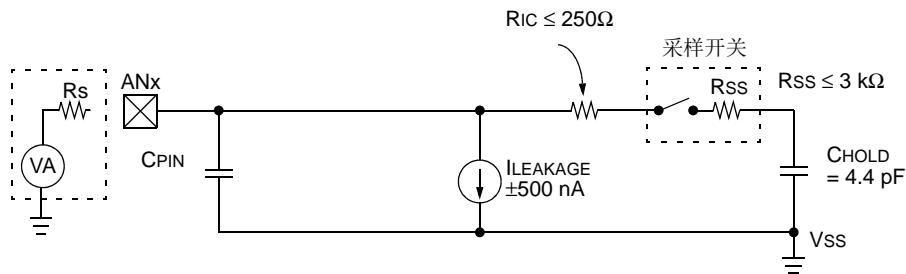
bit 15-0      **CTMEN<15:0>**: 转换期间 CTMU 使能位

1 = 在转换期间使能 CTMU 并连接到选定的通道

0 = CTMU 不连接到该通道

注 1: 实际的可用通道数取决于特定器件上实现了哪些通道; 详情请参见器件数据手册。未实现的通道, 读为 0。

图 24-3：10 位 A/D 转换器模拟输入模型



图注：	CPIN = 输入电容 VT = 阈值电压 ILEAKAGE = 引脚上由各结点产生的泄漏电流 RIC = 内部走线等效电阻 RSS = 采样开关电阻 CHOLD = 采样 / 保持电容 (来自 DAC)
-----	---

注：CPIN 值取决于器件封装，未经测试。如果  $Rs \leq 5 \text{ k}\Omega$ ，可忽略 CPIN 的影响。

公式 24-1：A/D 转换时钟周期

$$T_{AD} = T_{CY} (ADCS + 1)$$

$$ADCS = \frac{T_{AD}}{T_{CY}} - 1$$

注：基于  $T_{CY} = 2/F_{OSC}$ ；打盹模式和 PLL 被禁止。

# PIC24FJ128GA310 系列

图 24-4：12 位 A/D 传递函数

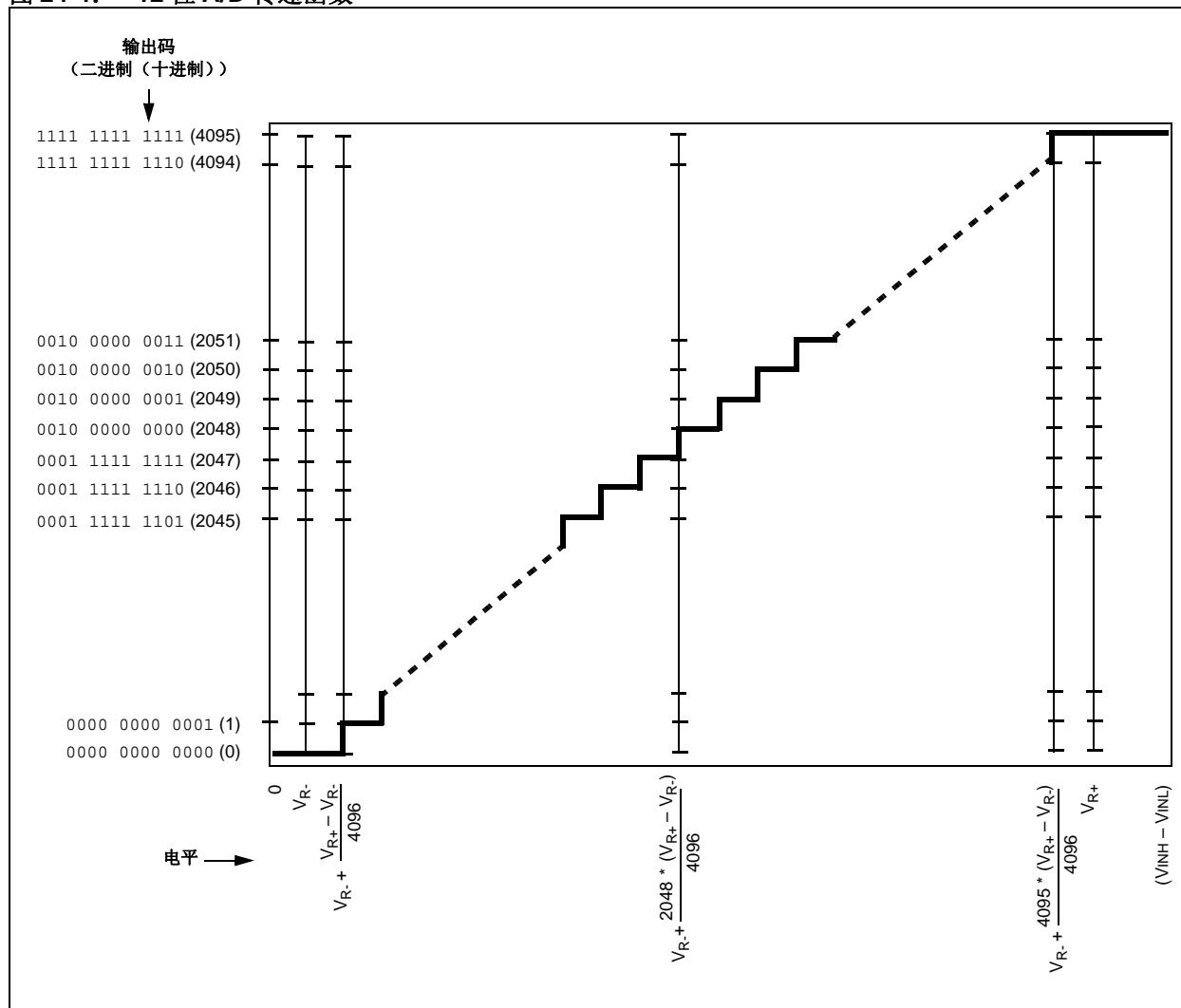
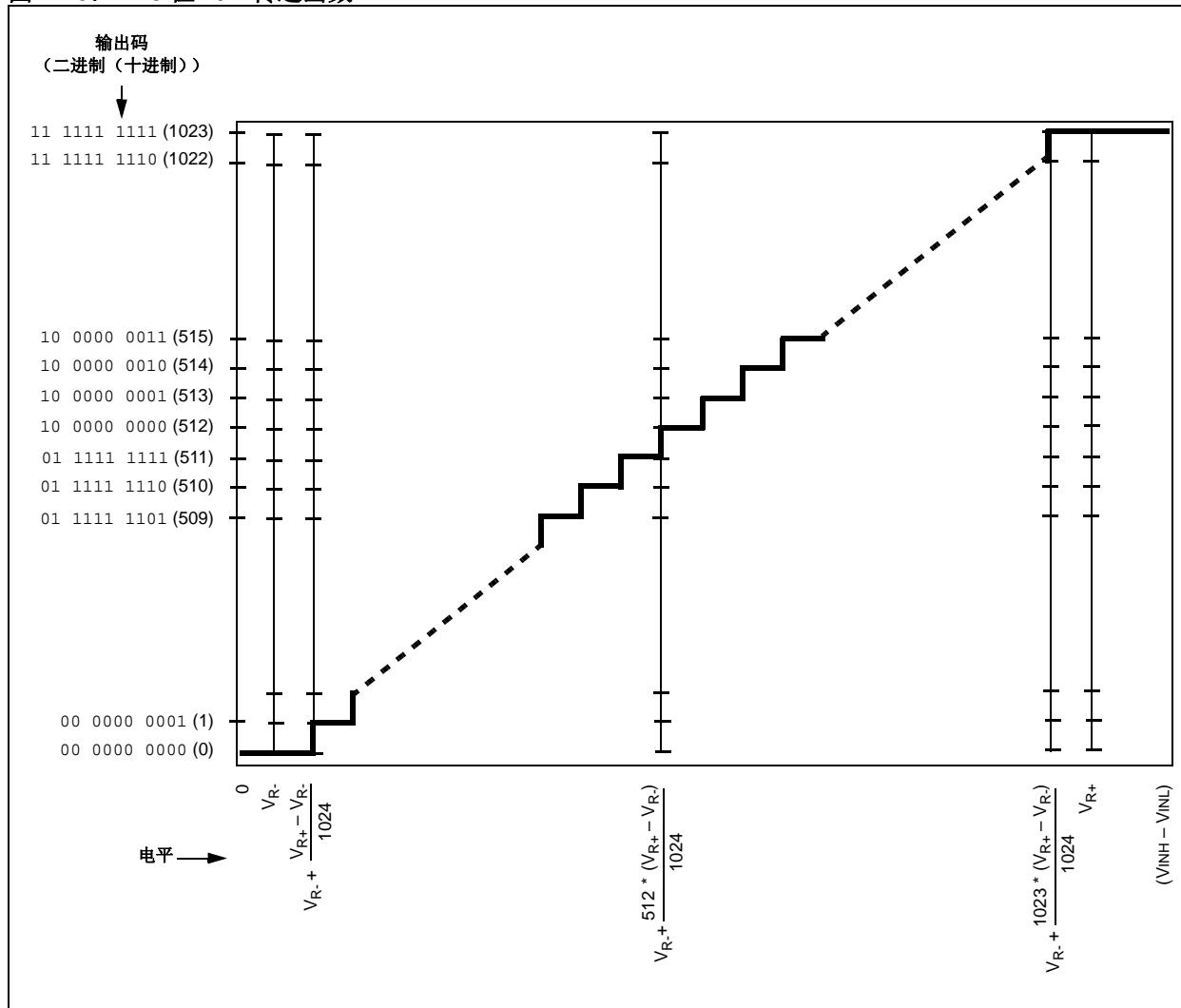


图 24-5：10 位 A/D 传递函数



# **PIC24FJ128GA310 系列**

---

---

注:

## 25.0 三比较器模块

**注：**本数据手册总结了该组 PIC24F 器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解更多信息，请参见《PIC24F 系列参考手册》中的第 46 章“可扩展的比较器模块”(DS39734A\_CN)。本数据手册中的信息取代了 FRM 中的信息。

三比较器模块提供三个双输入比较器。至比较器的输入可配置为使用五个外部模拟输入 (CxINA、CxINB、CxINC、CxIND 和 VREF+) 之一，且参考电压输入可以

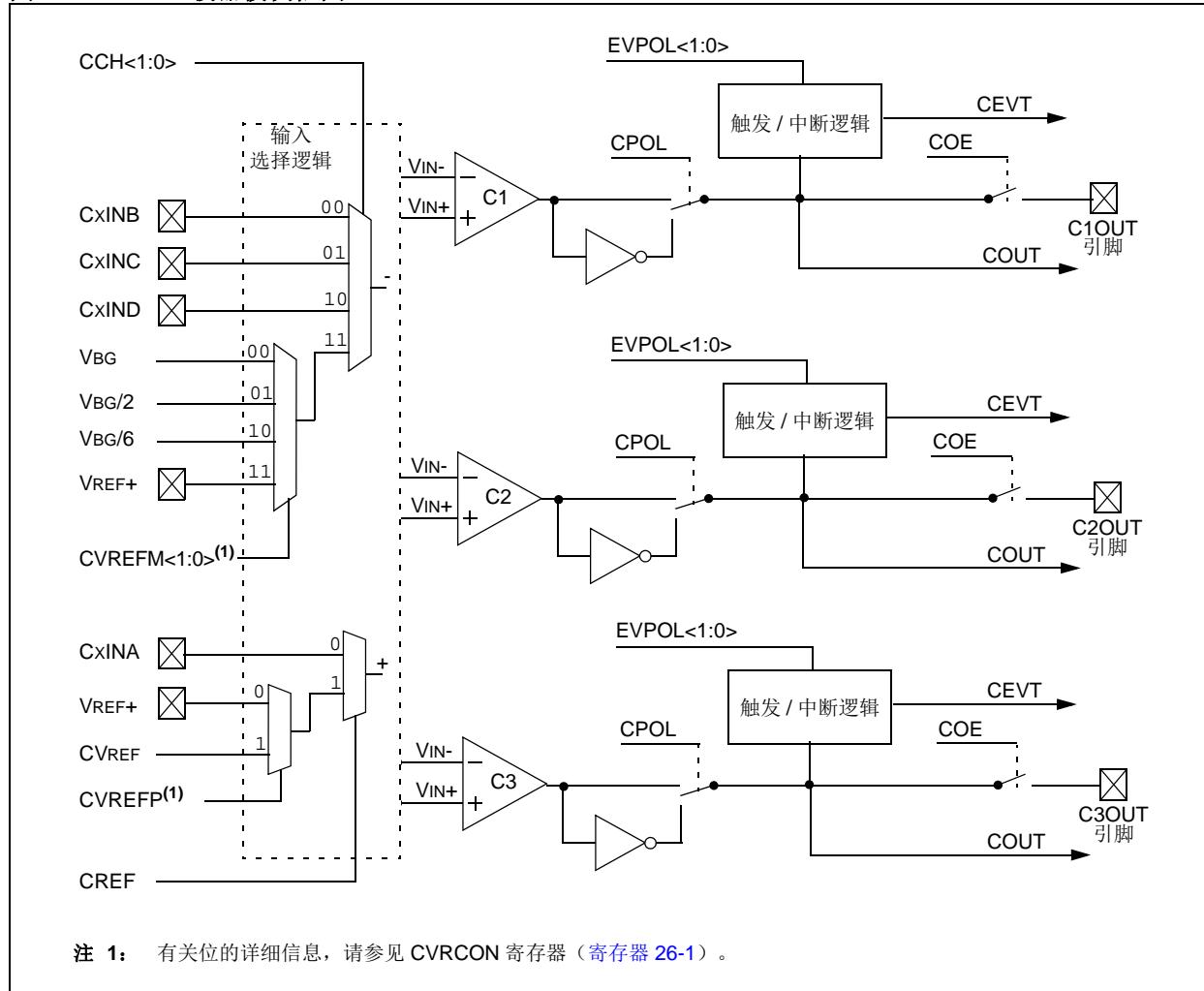
来自其中一个内部带隙参考或比较器参考电压发生器 (VBG、VBG/2、VBG/6 和 CVREF)。

比较器输出可直接连接到 CxOUT 引脚。当对应的 COE 等于 1 时，I/O 填充逻辑使得引脚输出不同步的比较器输出信号。

模块的简化框图如图 25-1 所示。图 25-2 至图 25-4 中给出了各种可能的比较器配置图。

每个比较器都有自己的控制寄存器 CMxCON (寄存器 25-1)，用于使能和配置其操作。所有三个比较器的输出和事件状态都在 CMSTAT 寄存器 (寄存器 25-2) 中给出。

图 25-1：三比较器模块框图



# PIC24FJ128GA310 系列

图 25-2: CREF = 0 时的各个比较器配置:

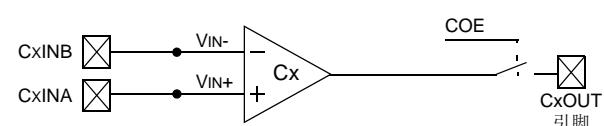
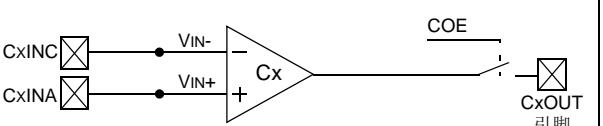
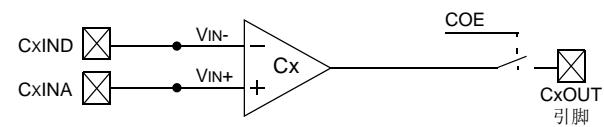
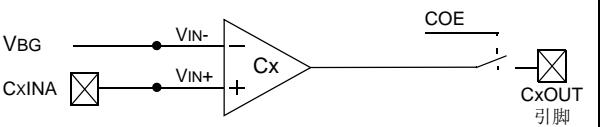
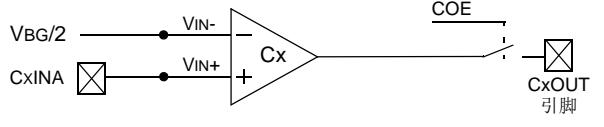
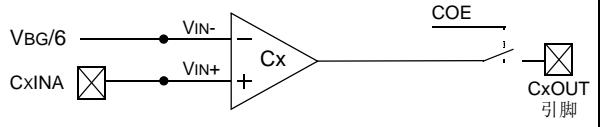
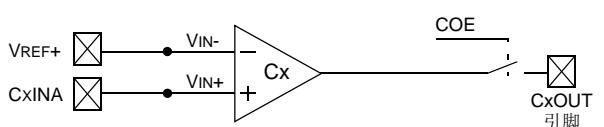
比较器关闭 CEN = 0, CREF = x, CCH<1:0> = xx	
	
比较器 CxINB > CxINA 比较 CEN = 1, CCH<1:0> = 00, CVREFM<1:0> = xx	比较器 CxINC > CxINA 比较 CEN = 1, CCH<1:0> = 01, CVREFM<1:0> = xx
	
比较器 CxIND > CxINA 比较 CEN = 1, CCH<1:0> = 10, CVREFM<1:0> = xx	比较器 VBG > CxINA 比较 CEN = 1, CCH<1:0> = 11, CVREFM<1:0> = 00
	
比较器 VBG > CxINA 比较 CEN = 1, CCH<1:0> = 11, CVREFM<1:0> = 01	比较器 VBG > CxINA 比较 CEN = 1, CCH<1:0> = 11, CVREFM<1:0> = 10
	
比较器 CxIND > CxINA 比较 CEN = 1, CCH<1:0> = 11, CVREFM<1:0> = 11	
	

图 25-3: CREF = 1 且 CVREFP = 0 时的各个比较器配置

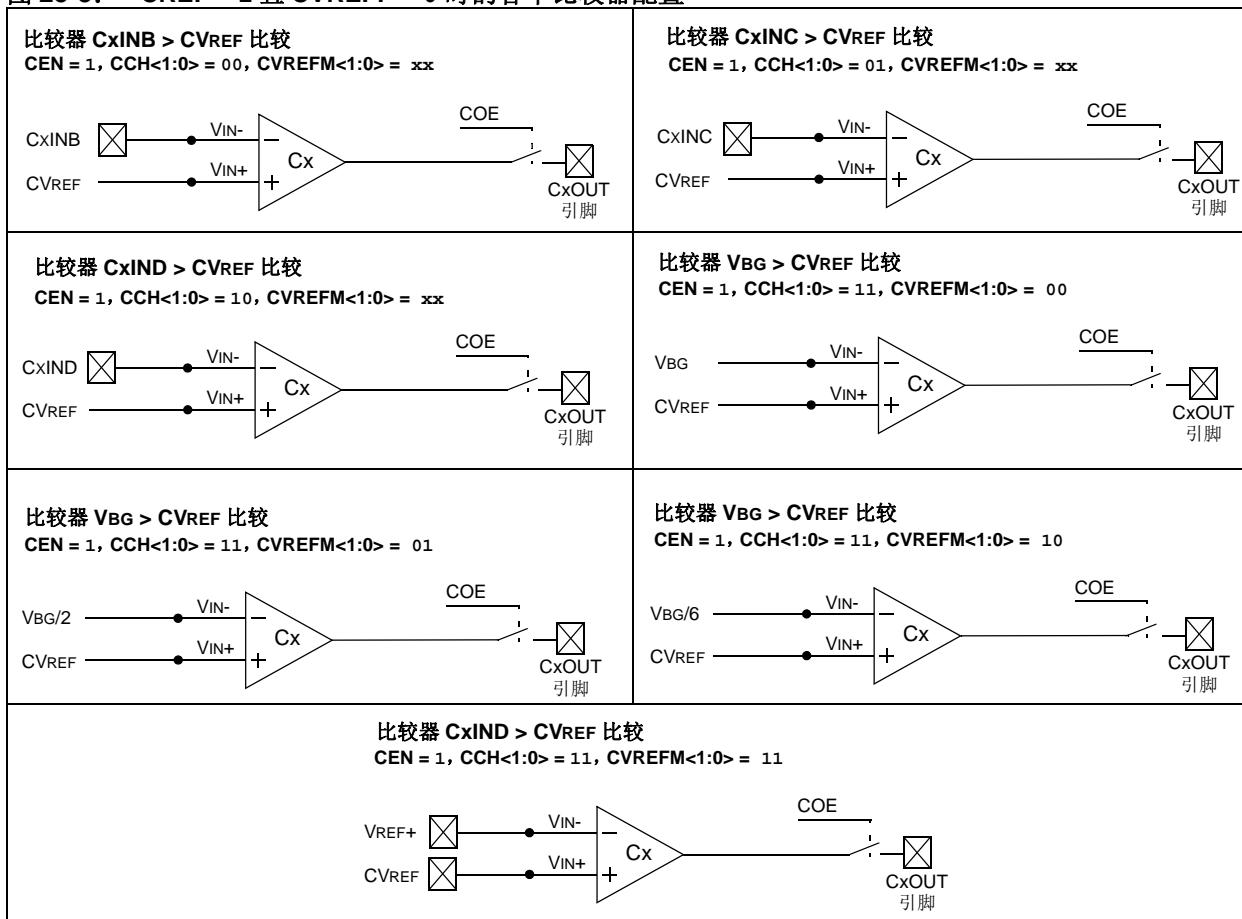
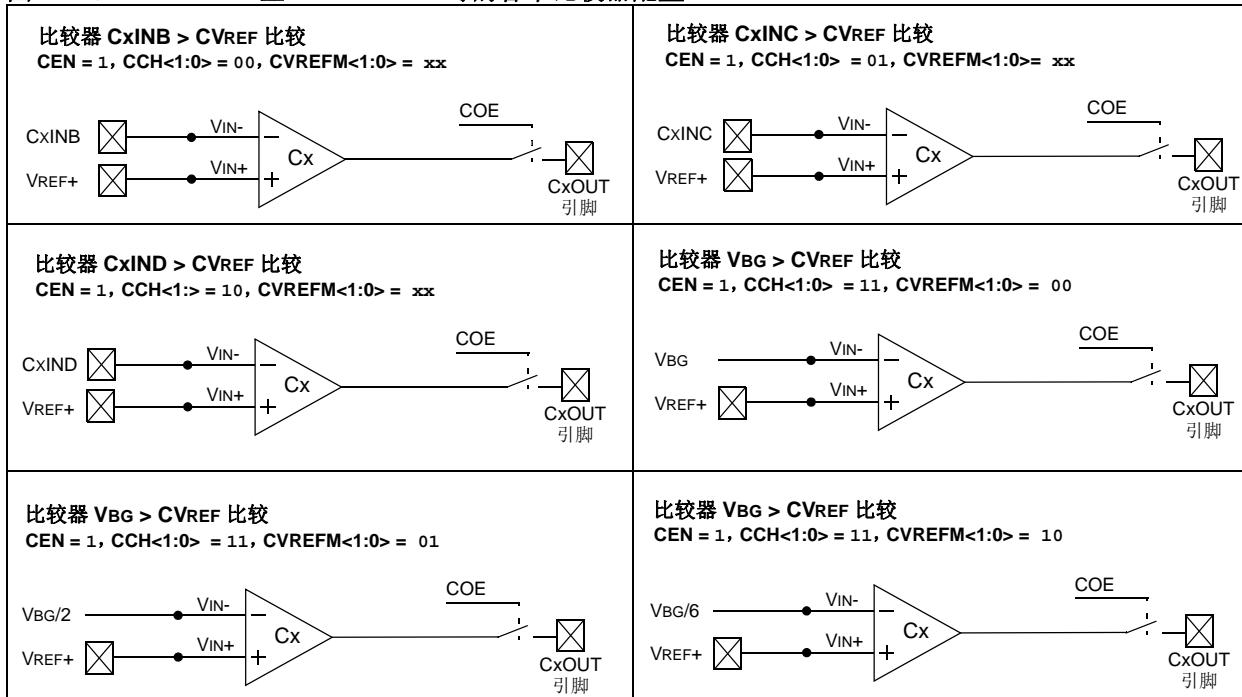


图 25-4: CREF = 1 且 CVREFP = 1 时的各个比较器配置



# PIC24FJ128GA310 系列

寄存器 25-1： CMxCON： 比较器 x 控制寄存器（比较器 1 到 3）

R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0, HS	R-0, HSC
CEN	COE	CPOL	—	—	—	CEVT	COUT
bit 15	bit 8						

R/W-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	R/W-0
EVPOL1	EVPOL0	—	CREF	—	—	CCH1	CCH0
bit 7	bit 0						

图注：	HS = 可由硬件置 1 的位	HSC = 可由硬件置 1/ 清零的位
R = 可读位	W = 可写位	U = 未实现位，读为 0
-n = 上电复位时的值	1 = 置 1	0 = 清零

- bit 15      **CEN:** 比较器使能位  
1 = 使能比较器  
0 = 禁止比较器
- bit 14      **COE:** 比较器输出使能位  
1 = 在 CxOUT 引脚上提供比较器输出  
0 = 比较器输出仅供内部使用
- bit 13      **CPOL:** 比较器输出极性选择位  
1 = 比较器输出反相  
0 = 比较器输出不反相
- bit 12-10    未实现：读为 0
- bit 9        **CEVT:** 比较器事件位  
1 = 发生由 EVPOL<1:0> 定义的比较器事件；该位清零之前禁止后续触发和中断  
0 = 未产生比较器事件
- bit 8        **COUT:** 比较器输出位  
当 CPOL = 0 时：  
1 = VIN+ > VIN-  
0 = VIN+ < VIN-  
当 CPOL = 1 时：  
1 = VIN+ < VIN-  
0 = VIN+ > VIN-
- bit 7-6      **EVPOL<1:0>:** 触发 / 事件 / 中断极性选择位  
11 = 比较器输出的任何变化导致发生触发 / 事件 / 中断（当 CEVT = 0 时）  
10 = 比较器输出的以下跳变导致发生触发 / 事件 / 中断：  
    如果 CPOL = 0 (极性不反相)：  
        仅从高电平到低电平的跳变。  
    如果 CPOL = 1 (极性反相)：  
        仅从低电平到高电平的跳变。  
01 = 比较器输出的以下跳变导致发生触发 / 事件 / 中断：  
    如果 CPOL = 0 (极性不反相)：  
        仅从低电平到高电平的跳变。  
    如果 CPOL = 1 (极性反相)：  
        仅从高电平到低电平的跳变。  
00 = 禁止发生触发 / 事件 / 中断
- bit 5        未实现：读为 0

## 寄存器 25-1: CMxCON: 比较器 x 控制寄存器 (比较器 1 到 3) (续)

bit 4	<b>CREF:</b> 比较器参考电压选择位 (同相输入)
	1 = 同相输入连接到内部 CVREF 电压
	0 = 同相输入连接到 CxINA 引脚
bit 3-2	未实现: 读为 0
bit 1-0	<b>CCH&lt;1:0&gt;:</b> 比较器通道选择位
	11 = 比较器反相输入连接到 CVRCON 寄存器中的 CVREFM<1:0> 位指定的内部可选参考电压
	10 = 比较器反相输入连接到 CxIND 引脚
	10 = 比较器反相输入连接到 CxINC 引脚
	10 = 比较器反相输入连接到 CxINB 引脚

## 寄存器 25-2: CMSTAT: 比较器模块状态寄存器

R/W-0	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
CMIDL	—	—	—	—	C3EVT	C2EVT	C1EVT
bit 15							
							bit 8
	U-0	U-0	U-0	U-0	R-0, HSC	R-0, HSC	R-0, HSC
	—	—	—	—	C3OUT	C2OUT	C1OUT
bit 7							
							bit 0

<b>图注:</b>	HSC = 可由硬件置 1/ 清零的位
R = 可读位	W = 可写位
-n = 上电复位时的值	1 = 置 1      0 = 清零      x = 未知

bit 15	<b>CMIDL:</b> 比较器空闲模式停止位
	1 = 当器件进入空闲模式时, 所有比较器停止工作
	0 = 所有使能的比较器在空闲模式下继续工作
bit 14-11	未实现: 读为 0
bit 10	<b>C3EVT:</b> 比较器 3 事件状态位 (只读)
	显示比较器 3 的当前事件状态 (CM3CON<9>)。
bit 9	<b>C2EVT:</b> 比较器 2 事件状态位 (只读)
	显示比较器 2 的当前事件状态 (CM2CON<9>)。
bit 8	<b>C1EVT:</b> 比较器 1 事件状态位 (只读)
	显示比较器 1 的当前事件状态 (CM1CON<9>)。
bit 7-3	未实现: 读为 0
bit 2	<b>C3OUT:</b> 比较器 3 输出状态位 (只读)
	显示比较器 3 的当前输出 (CM3CON<8>)。
bit 1	<b>C2OUT:</b> 比较器 2 输出状态位 (只读)
	显示比较器 2 的当前输出 (CM2CON<8>)。
bit 0	<b>C1OUT:</b> 比较器 1 输出状态位 (只读)
	显示比较器 1 的当前输出 (CM1CON<8>)。

# **PIC24FJ128GA310 系列**

---

---

注:

## 26.0 比较器参考电压

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息, 请参见《PIC24F 系列参考手册》中的第 19 章“比较器模块”(DS39710B\_CN)。本数据手册中的信息取代了 FRM 中的信息。

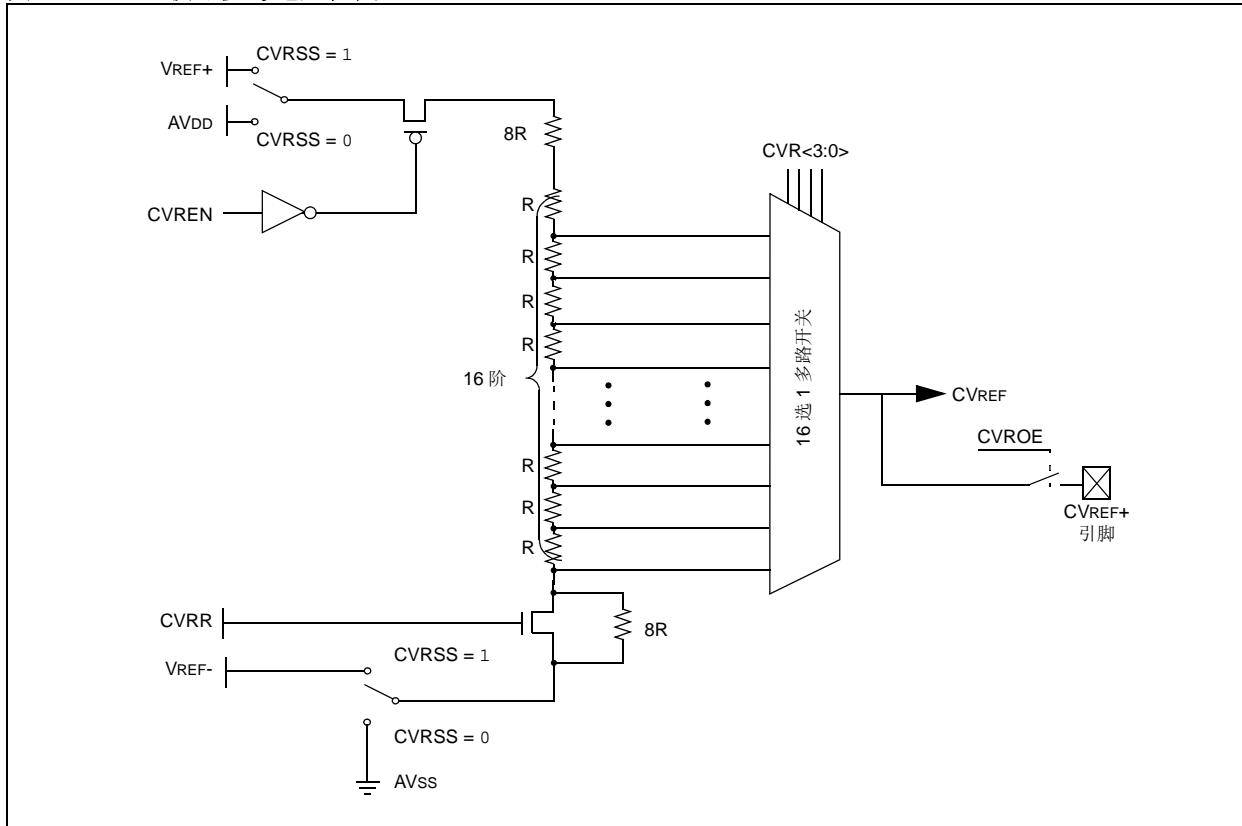
## 26.1 配置比较器参考电压

参考电压模块由 CVRCON 寄存器 (寄存器 26-1) 控制, 它能提供两种范围的输出电压, 每种范围都具有 16 种不同的电平。CVRR 位 (CVRCON<5>) 选择要使用的电压范围。这两种电压范围的主要区别在于 CVREF 选择位 (CVR<3:0>) 所选的步长不同, 其中一个范围提供更高的分辨率。

比较器参考电压模块的电压可以来自 VDD 和 VSS 或外部 VREF+ 和 VREF-。电压源通过 CVRSS 位 (CVRCON<4>) 选择。

在改变 CVREF 输出值时, 必须考虑到比较器参考电压的稳定时间。

图 26-1: 比较器参考电压框图



# PIC24FJ128GA310 系列

寄存器 26-1: CVRCON: 比较器参考电压控制寄存器

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	CVREFP	CVREFM1	CVREFM0
bit 15							bit 8

| R/W-0 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| CVREN | CVROE | CVRR  | CVRSS | CVR3  | CVR2  | CVR1  | CVR0  |
| bit 7 |       |       |       |       |       |       | bit 0 |

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-11 未实现: 读为 0

bit 10 **CVREFP:** 参考电压选择位 (仅当 CREF 为 1 时有效)

1 = VREF+ 用作比较器的参考电压

0 = 此模块内的 CVR (4 位 DAC) 为比较器提供参考电压

bit 9-8 **CVREFM<1:0>:** 带隙参考源选择位 (仅当 CCH<1:0> = 11 时有效)

00 = 提供带隙电压作为比较器的输入

01 = 提供二分之一带隙电压作为比较器的输入

10 = 提供六分之一带隙电压作为比较器的输入

11 = 提供 VREF+ 引脚作为比较器的输入

bit 7 **CVREN:** 比较器参考电压使能位

1 = CVREF 电路上电

0 = CVREF 电路掉电

bit 6 **CVROE:** 比较器 VREF 输出使能位

1 = CVREF 电压是 CVREF 引脚上的输出

0 = CVREF 电压与 CVREF 引脚断开

bit 5 **CVRR:** 比较器 VREF 范围选择位

1 = CVRSRC 范围应从 0 到 0.625 CVRSRC, 步长为 CVRSRC/24

0 = CVRSRC 范围应从 0.25 到 0.719 CVRSRC, 步长为 CVRSRC/32

bit 4 **CVRSS:** 比较器 VREF 源选择位

1 = 比较器参考源 CVRSRC = VREF+ - VREF-

0 = 比较器参考源 CVRSRC = AVDD - AVSS

bit 3-0 **CVR<3:0>:** 比较器 VREF 值选择位 ( $0 \leq \text{CVR}<3:0> \leq 15$ )

当 CVRR = 1 时:

$$\text{CVREF} = (\text{CVR}<3:0>/24) \bullet (\text{CVRSRC})$$

当 CVRR = 0 时:

$$\text{CVREF} = 1/4 \bullet (\text{CVRSRC}) + (\text{CVR}<3:0>/32) \bullet (\text{CVRSRC})$$

## 27.0 充电时间测量单元 (CTMU)

**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。如需了解有关充电时间测量单元的更多信息, 请参见《PIC24F 系列参考手册》中的第 11 章“充电时间测量单元 (CTMU)”(DS39724B\_CN)。

充电时间测量单元 (Charge Time Measurement Unit, CTMU) 是一个灵活的模拟模块, 它提供充电测量、脉冲源之间的精确时间差测量以及异步脉冲生成功能。它的主要特性包括:

- 十三个外部边沿输入触发源
- 每个边沿源的极性控制
- 边沿顺序控制
- 边沿电平或边沿跳变响应控制
- 时间测量分辨率为 1 纳秒
- 适合电容测量的精确电流源

CTMU 可与其他片上模拟模块一起, 用于精确测量时间、电容、电容的相对变化, 或生成独立于系统时钟的输出脉冲。CTMU 模块是连接电容式触摸传感器的理想选择。

CTMU 可通过三个寄存器进行控制: CTMUCON1、CTMUCON2 和 CTMUICON。CTMUCON1 用于使能模块和控制 CTMU 的工作模式, 以及控制边沿顺序。CTMUCON2 用于控制边沿源选择和边沿源极性选择。CTMUICON 寄存器用于选择电流源的电流范围并微调电流。

## 27.1 测量电容

CTMU 模块通过在两个单独的输入通道上产生输出脉冲 (脉冲宽度等于边沿事件之间的时间间隔) 来测量电容。这两个输入通道上的脉冲边沿事件源有四种选择: 两个内部外设模块 (OC1 和 Timer1) 和最多 13 个外部引脚 (CTEDG1 到 CTEDG13)。该脉冲和该模块的精确电流源一起使用, 可根据以下关系计算电容:

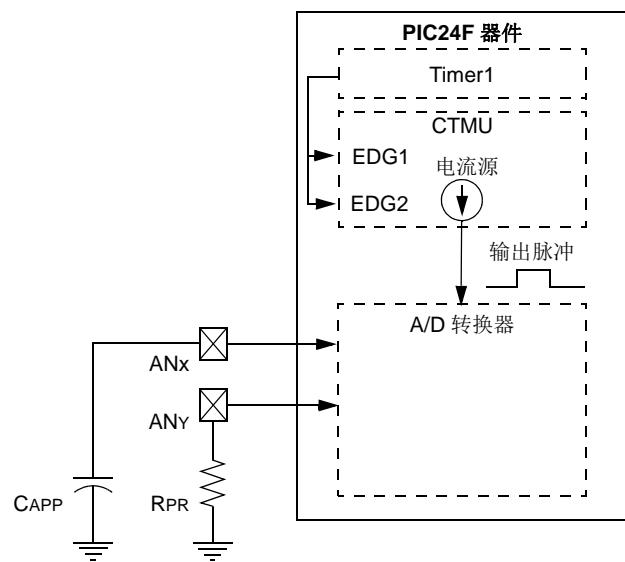
**公式 27-1:**

$$I = C \cdot \frac{dV}{dT}$$

如需测量电容, A/D 转换器在 CTMU 输出的脉冲信号之后对它的其中一个输入通道采样外部电容 (CAPP)。由第二个 A/D 通道上的高精度电阻 (RPR) 提供电流源校准。脉冲信号结束后, 转换器确定电容上的电压。电容的实际计算在软件中由应用程序执行。

图 27-1 给出了电容测量使用的外部连接以及此应用中 CTMU 和 A/D 模块的关系。此示例还展示了来自 Timer1 的边沿事件, 但使用外部边沿源的其他配置也是可能的。有关使用 CTMU 模块测量电容和时间的详细说明, 请参见《PIC24F 系列参考手册》中的第 11 章“充电时间测量单元 (CTMU)”。

图 27-1：电容测量的典型连接和内部配置



## 27.2 测量时间

对脉冲宽度的时间测量也可类似地执行，使用A/D模块的内部电容（CAD）和高精度电阻校准电流。图 27-2 显示了时间测量使用的外部连接以及此应用中 CTMU 和 A/D模块的关系。此示例还显示了来自外部 CTEDG 引脚的边沿事件，但使用内部边沿源的其他配置也是可能的。

## 27.3 脉冲生成和延时

CTMU 模块也可生成边沿和器件的系统时钟异步的输出脉冲。更明确地说，它可以产生具有可编程延时的脉冲，自边沿事件输入至模块后经过这段延时，脉冲才会输出。

当模块通过将 TGEN 位（CTMUCON1<12>）置 1 配置为脉冲生成延时时，内部电流源连接到比较器 2 的 B 输入。将电容（CDELAY）连接到比较器 2 的引脚 C2INB，且比较器参考电压 CVREF 连接到 C2INA。CVREF 随后被配置为特定跳变点。当检测到边沿事件时，模块开始对 CDELAY 充电。当 CDELAY 充电到超过 CVREF 跳变点时，在 CTPLS 上输出脉冲信号。脉冲延时的时间长度由 CDELAY 和 CVREF 跳变点的值决定。

图 27-3 给出了脉冲生成的外部连接，以及所需的不同模拟模块之间的关系。CTED1 显示为输入脉冲源时，其他选项可用。关于使用 CTMU 模块生成脉冲的详细说明，请参见《PIC24F 系列参考手册》。

图 27-2：时间测量的典型连接和内部配置测量

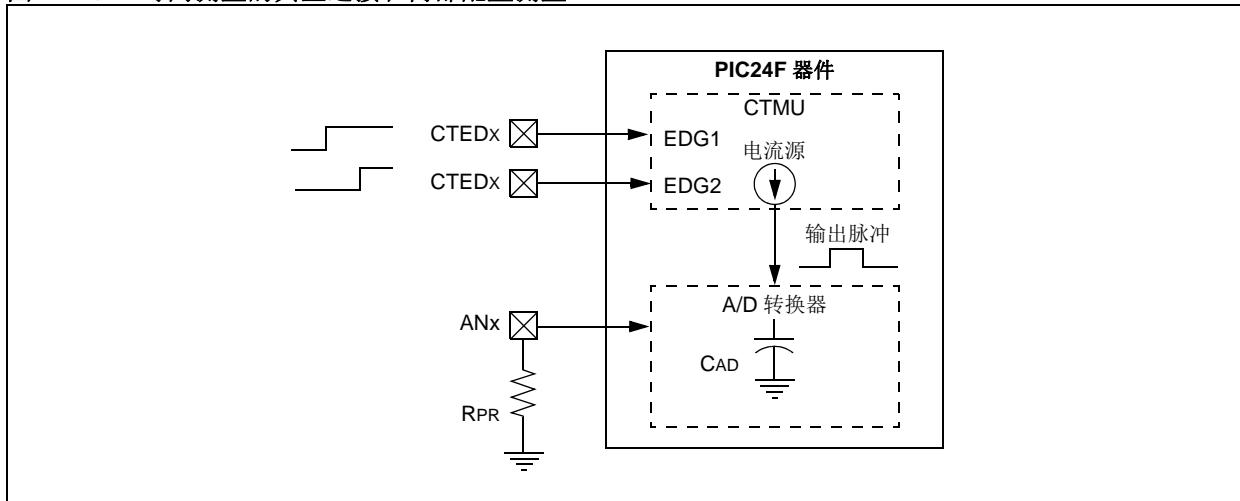
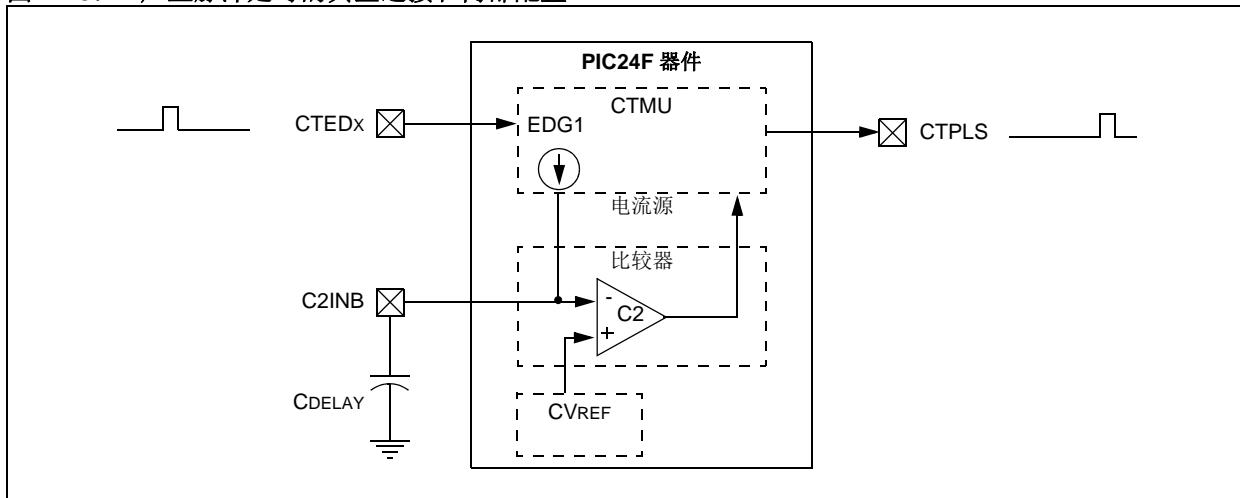


图 27-3：产生脉冲延时的典型连接和内部配置



# PIC24FJ128GA310 系列

寄存器 27-1: CTMUCON1: CTMU 控制寄存器 1

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CTMUEEN	—	CTMUSIDL	TGEN	EDGEN	EDGSEQEN	IDISSEN	CTTRIG
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15      **CTMUEEN:** CTMU 使能位

1 = 使能模块

0 = 禁止模块

bit 14      未实现: 读为 0

bit 13      **CTMUSIDL:** 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      **TGEN:** 延时产生使能位

1 = 使能边沿延时生成

0 = 禁止边沿延时生成

bit 11      **EDGEN:** 边沿使能位

1 = 未阻止边沿

0 = 阻止边沿

bit 10      **EDGSEQEN:** 边沿顺序使能位

1 = 边沿 1 事件必须在边沿 2 事件发生前发生

0 = 无需边沿序列

bit 9      **IDISSEN:** 模拟电流源控制位

1 = 模拟电流源输出接地

0 = 模拟电流源输出未接地

bit 8      **CTTRIG:** 触发信号控制位

1 = 使能触发器输出

0 = 禁止触发器输出

bit 7-0      未实现: 读为 0

## 寄存器 27-2: CTMUCON2: CTMU 控制寄存器 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
EDG1MOD	EDG1POL	EDG1SEL3	EDG1SEL2	EDG1SEL1	EDG1SEL0	EDG2STAT	EDG1STAT
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
EDG2MOD	EDG2POL	EDG2SEL3	EDG2SEL2	EDG2SEL1	EDG2SEL0	—	—
bit 7	bit 0						

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15 **EDG1MOD:** 边沿 1 边沿敏感选择位

1 = 输入边沿敏感

0 = 输入电平敏感

bit 14 **EDG1POL:** 边沿 1 极性选择位

1 = 边沿 1 编程为正边沿响应

0 = 边沿 1 编程为负边沿响应

bit 13-10 **EDG1SEL<3:0>:** 边沿 1 源选择位

1111 = 边沿 1 源是比较器 3 的输出

1110 = 边沿 1 源是比较器 2 的输出

1101 = 边沿 1 源是比较器 1 的输出

1100 = 边沿 1 源是 IC3

1011 = 边沿 1 源是 IC2

1010 = 边沿 1 源是 IC1

1001 = 边沿 1 源是 CTED8

1000 = 边沿 1 源是 CTED7<sup>(1)</sup>

0111 = 边沿 1 源是 CTED6

0110 = 边沿 1 源是 CTED5

0101 = 边沿 1 源是 CTED4

0100 = 边沿 1 源是 CTED3<sup>(1)</sup>

0011 = 边沿 1 源是 CTED1

0010 = 边沿 1 源是 CTED2

0001 = 边沿 1 源是 OC1

0000 = 边沿 1 源是 Timer1

bit 9 **EDG2STAT:** 边沿 2 状态位

指示边沿 2 的状态并可写入以控制电流源。

1 = 已发生边沿 2

0 = 未发生边沿 2

bit 8 **EDG1STAT:** 边沿 1 状态位

指示边沿 1 的状态并可写入以控制电流源。

1 = 已发生边沿 1

0 = 未发生边沿 1

bit 7 **EDG2MOD:** 边沿 2 边沿敏感选择位

1 = 输入边沿敏感

0 = 输入电平敏感

bit 6 **EDG2POL:** 边沿 2 极性选择位

1 = 边沿 2 编程为正边沿

0 = 边沿 2 编程为负边沿

**注 1:** 边沿源、CTED3、CTED7、CTED10 和 CTED11 仅在 100 引脚器件中提供。

# PIC24FJ128GA310 系列

---

---

## 寄存器 27-2: CTMUCON2: CTMU 控制寄存器 2 (续)

bit 5-2      **EDG2SEL<3:0>**: 边沿 2 源选择位

1111 = 边沿 2 源是比较器 3 的输出  
1110 = 边沿 2 源是比较器 2 的输出  
1101 = 边沿 2 源是比较器 1 的输出  
1100 = 未实现, 不要使用  
1011 = 边沿 2 源是 IC3  
1010 = 边沿 2 源是 IC2  
1001 = 边沿 2 源是 IC1  
1000 = 边沿 2 源是 CTED13  
0111 = 边沿 2 源是 CTED12  
0110 = 边沿 2 源是 CTED11<sup>(1)</sup>  
0101 = 边沿 2 源是 CTED10<sup>(1)</sup>  
0100 = 边沿 2 源是 CTED9  
0011 = 边沿 2 源是 CTED1  
0010 = 边沿 2 源是 CTED2  
0001 = 边沿 2 源是 OC1  
0000 = 边沿 2 源是 Timer1

bit 1-0      未实现: 读为 0

**注 1:** 边沿源、CTED3、CTED7、CTED10 和 CTED11 仅在 100 引脚器件中提供。

## 寄存器 27-3: CTMUICON: CTMU 电流控制寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ITRIM5	ITRIM4	ITRIM3	ITRIM2	ITRIM1	ITRIM0	IRNG1	IRNG0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15-10      **ITRIM<5:0>**: 电流源微调位

011111 = 对标称电流的最大正向调整  
011110

.

.

000001 = 对标称电流的最小正向调整

000000 = IRNG<1:0> 指定的标称电流输出

111111 = 对标称电流的最小负向调整

.

.

100010

100001 = 对标称电流的最大负向调整

bit 9-8      **IRNG<1:0>**: 电流源范围选择位

11 = 100 × 基本电流

10 = 10 × 基本电流

01 = 基本电流 (标称值为 0.55 μA)

00 = 1000 × 基本电流

bit 7-0      未实现: 读为 0

# **PIC24FJ128GA310 系列**

---

---

注:

## 28.0 高 / 低压检测 (HLVD)

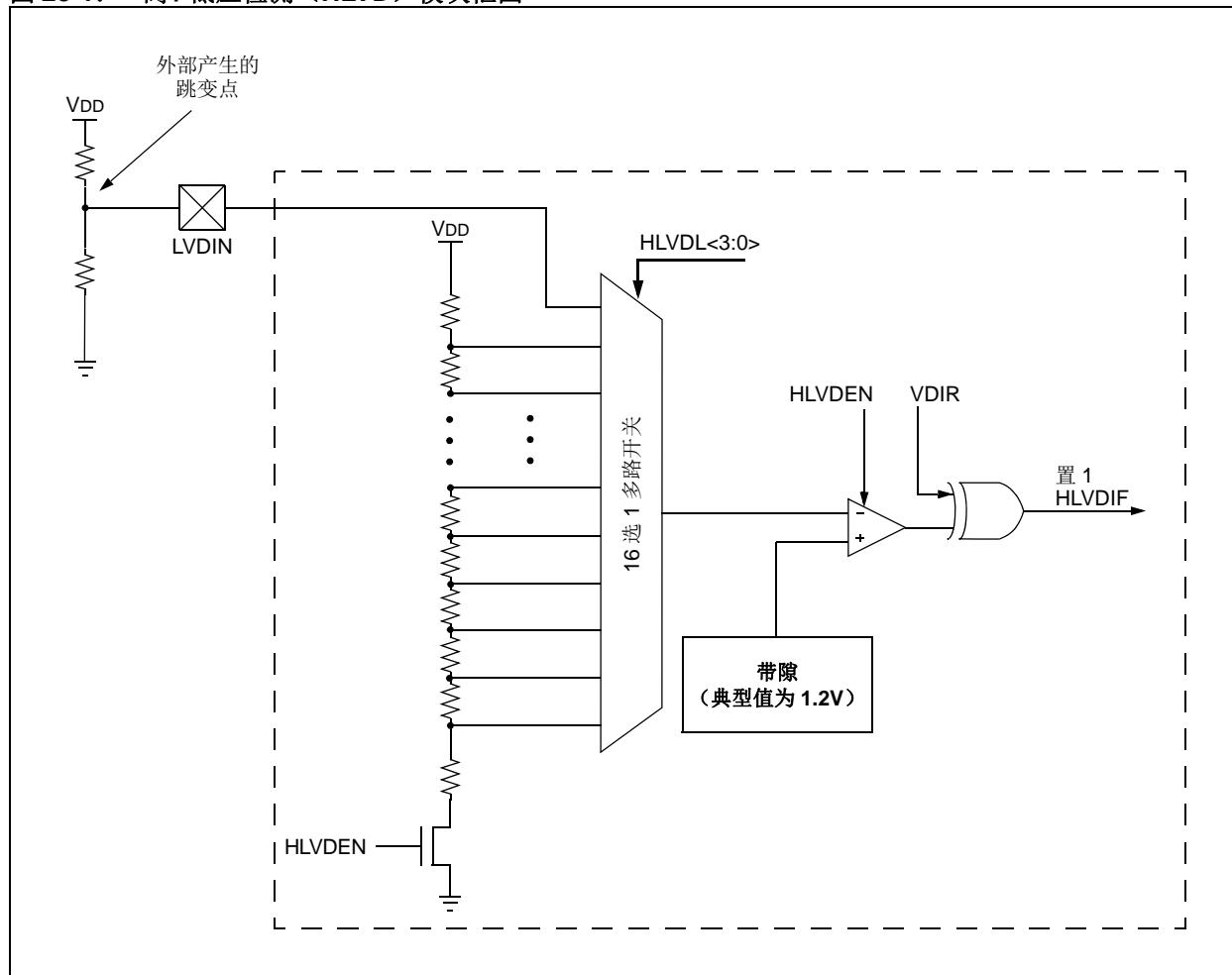
**注:** 本数据手册总结了该组 PIC24F 器件的功能。但是不应把本参考手册当作无所不包的参考手册来使用。有关高 / 低压检测的更多信息, 请参见《PIC24F 系列参考手册》中的第 36 章“高度集成的可编程高 / 低电压检测 (HLVD) 模块”(DS39725A\_CN)。

如果器件电压在变化方向相对于该跳变点发生了偏离, 就会将中断标志位置 1。如果允许了中断, 程序将跳转到中断向量地址处执行, 然后由软件响应该中断。

HLVD 控制寄存器 (见寄存器 28-1) 完全控制 HLVD 模块的操作。用户可通过软件控制该寄存器将电路“关闭”, 从而使器件的电流消耗降至最低。

高 / 低压检测 (HLVD) 模块是允许用户指定器件电压跳变点和变化方向的可编程电路。

图 28-1: 高 / 低压检测 (HLVD) 模块框图



# PIC24FJ128GA310 系列

寄存器 28-1： HLVDCON： 高 / 低压检测控制寄存器

R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
HLVDEN	—	LSIDL	—	—	—	—	—
bit 15	bit 8						

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
VDIR	BGVST	IRVST	—	HLVDL3	HLVDL2	HLVDL1	HLVDL0
bit 7	bit 0						

图注：

R = 可读位

W = 可写位

U = 未实现位，读为 0

-n = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

bit 15	<b>HLVDEN:</b> 高/低压检测功能使能位 1 = 使能 HLVD 0 = 禁止 HLVD
bit 14	<b>未实现:</b> 读为 0
bit 13	<b>LSIDL:</b> HLVD 在空闲模式停止位 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	<b>未实现:</b> 读为 0
bit 7	<b>VDIR:</b> 电压变化方向选择位 1 = 当电压等于或超过跳变点 (HLVDL<3:0>) 时，事件发生 0 = 当电压等于或低于跳变点 (HLVDL<3:0>) 时，事件发生
bit 6	<b>BGVST:</b> 带隙电压稳定标志位 1 = 表示带隙电压稳定 0 = 表示带隙电压不稳定
bit 5	<b>IRVST:</b> 内部参考电压稳定标志位 1 = 内部参考电压稳定；高压检测逻辑在检测到指定的电压范围时产生中断标志 0 = 内部参考电压不稳定；高压检测逻辑在检测到指定的电压范围时不会产生中断标志，并且 HLVD 中断不被允许
bit 4	<b>未实现:</b> 读为 0
bit 3-0	<b>HLVDL&lt;3:0&gt;:</b> 高 / 低压检测限制位 1111 = 使用外部模拟输入（输入来自于 LVDIN 引脚） 1110 = 跳变点 1 <sup>(1)</sup> 1101 = 跳变点 2 <sup>(1)</sup> 1100 = 跳变点 3 <sup>(1)</sup> · · · 0100 = 跳变点 11 <sup>(1)</sup> 00xx = 未使用

注 1： 有关实际跳变点，请参见第 32.0 节“电气特性”。

## 29.0 特殊功能

<b>注:</b>	本数据手册总结了该组 PIC24F 器件的功能。但是不应把本数据手册当作无所不包的参考手册来使用。如需了解更多信息,请参见《PIC24F 系列参考手册》中的以下章节。本数据手册中的信息取代了 FRM 中的信息。
• 第 9 章“看门狗定时器 (WDT) ”	(DS39697B_CN)
• 第 32 章“高级器件集成”	(DS39719D_CN)
• 第 33 章“编程和诊断”	(DS39716A_CN)

PIC24FJ128GA310 系列器件具有几项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的使用将成本降至最低。它们是：

- 灵活的配置
- 看门狗定时器 (WDT)
- 代码保护
- JTAG 边界扫描接口
- 在线串行编程
- 在线仿真

## 29.1 配置位

通过对配置位编程（读为 0）或不编程（读为 1）来选择不同的器件配置。这些配置位被映射到程序存储器中从 F80000h 开始的单元中。[寄存器 29-1 到寄存器 29-6](#) 详细说明了各配置位的不同功能。

注意，地址 F80000h 超出了用户程序存储空间的范围。事实上，它属于配置存储空间（800000h-FFFFFh），这一空间仅能通过表读和表写进行访问。

表 29-1： PIC24FJ128GA310 系列器件的闪存配置字位置

器件	配置字地址			
	1	2	3	4
PIC24FJ64GA3XX	ABFEh	ABFCh	ABFAh	ABF8h
PIC24FJ128GA3XX	157FEh	157FCh	157FAh	157F8h

### 29.1.1 配置 PIC24FJ128GA310 系列器件的注意事项

在 PIC24FJ128GA310 系列器件中，配置字节以易失性存储方式实现。这就意味着在器件每次上电时都必须对配置数据进行编程。配置数据存储在片上程序存储空间顶部的 3 个字中，这些字被称为闪存配置字。表 29-1 给出了它们的具体位置。这些字是实际器件配置位的紧凑表现形式，这些配置位实际上散布在配置空间的几个单元中。器件复位时，配置数据会被自动从闪存配置字装入相应的配置寄存器中。

**注:** 所有类型的器件复位都会重新装载配置数据。

当为这些器件创建应用程序时，用户应始终为配置数据分配特定的闪存配置字单元。此操作是为了确保在编译代码时不会把程序代码存储在该地址单元中。

程序存储器中的所有闪存配置字的高字节应始终为 0000 0000。这样当这些单元被远程事件意外执行时，会被当作一条 NOP 指令。由于配置位并未真正保存在对应的单元内，因此向这些单元写 0 不会影响器件工作。

**注:** 在程序存储器末页中执行页擦除操作会清除闪存配置字，从而使能代码保护。因此，用户应避免在程序存储器末页中执行页擦除操作。

# PIC24FJ128GA310 系列

寄存器 29-1： CW1：闪存配置字 1

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	R/PO-1						
r	JTAGEN	GCP	GWRP	DEBUG	LPCFG	ICS1	ICS0
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	FWDTEN1	FWDTEN0	FWPSA	WDTPS3	WDTPS2	WDTPS1	WDTPS0
bit 7							bit 0

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

- bit 23-16 未实现：读为 1  
bit 15 保留：该值未知；编程为 0  
bit 14 **JTAGEN:** JTAG 端口使能位  
    1 = 使能 JTAG 端口  
    0 = 禁止 JTAG 端口  
bit 13 **GCP:** 通用段程序存储器代码保护位  
    1 = 禁止代码保护  
    0 = 使能对整个程序存储空间的代码保护  
bit 12 **GWRP:** 通用段代码闪存写保护位  
    1 = 允许写程序存储器  
    0 = 禁止写程序存储器  
bit 11 **DEBUG:** 后台调试器使能位  
    1 = 器件复位至正常工作模式  
    0 = 器件复位至调试模式  
bit 10 **LPCFG:** 低压 / 保持稳压器配置位  
    1 = 始终禁止低压 / 保持稳压器  
    0 = 在固件中由 RETEN 位使能和控制低功耗、低压 / 保持稳压器  
bit 9-8 **ICS<1:0>:** 仿真器引脚位置选择位  
    11 = 仿真器功能与 PGEC1/PGED1 复用  
    10 = 仿真器功能与 PGEC2/PGED2 复用  
    01 = 仿真器功能与 PGEC3/PGED3 复用  
    00 = 保留；不要使用  
bit 7 **WINDIS:** 窗口看门狗定时器禁止位  
    1 = 使能标准看门狗定时器  
    0 = 使能窗口看门狗定时器；（FWDTEN<1:0> 不能为 00）  
bit 6-5 **FWDTEN<1:0>:** 看门狗定时器配置位  
    11 = 始终使能 WDT； SWDTEN 位没有任何作用  
    10 = 在固件中由 SWDTEN 位使能和控制 WDT  
    01 = 仅在运行模式下使能 WDT，在休眠模式下禁止 WDT； 禁止 SWDTEN 位  
    00 = 禁止 WDT； 禁止 SWDTEN 位  
bit 4 **FWPSA:** WDT 预分频比选择位  
    1 = 预分频比为 1:128  
    0 = 预分频比为 1:32

## 寄存器 29-1: CW1: 闪存配置字 1 (续)

bit 3-0      **WDTPS<3:0>**: 看门狗定时器后分频比选择位

1111 = 1:32,768

1110 = 1:16,384

1101 = 1:8,192

1100 = 1:4,096

1011 = 1:2,048

1010 = 1:1,024

1001 = 1:512

1000 = 1:256

0111 = 1:128

0110 = 1:64

0101 = 1:32

0100 = 1:16

0011 = 1:8

0010 = 1:4

0001 = 1:2

0000 = 1:1

# PIC24FJ128GA310 系列

## 寄存器 29-2: CW2: 闪存配置字 2

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R/PO-1	r-1	r-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1	R/PO-1
IESO	r	r	ALTVRF1	ALTVRF0	FNOSC2	FNOSC1	FNOSC0
bit 15							bit 8

R/PO-1	R/PO-1	R/PO-1	R/PO-1	r-1	r-1	R/PO-1	R/PO-1
FCKSM1	FCKSM0	OSCIOFCN	IOL1WAY	r	r	POSCMD1	POSCMD0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 23-16 未实现: 读为 1
- bit 15 **IESO:** 内部 / 外部时钟切换位  
1 = 使能 IESO 模式 (双速启动)  
0 = 禁止 IESO 模式 (双速启动)
- bit 14-13 保留: 始终保持为 1
- bit 12-11 **ALTVRF<1:0>:** 备用 VREF/CVREF 引脚选择位  
00 = 参考电压输入, A/D = RB0/RB1, 比较器 = RB0/RB1  
01 = 参考电压输入, A/D = RB0/RB1, 比较器 = RA9、RA10  
10 = 参考电压输入, A/D = RA9/RA10, 比较器 = RB0、RB1  
11 = 参考电压输入, A/D = RA9/RA10, 比较器 = RA9、RA10
- bit 10-8 **FNOSC<2:0>:** 初始振荡器选择位  
111 = 带后分频器的快速 RC 振荡器 (FRCDIV)  
110 = 保留  
101 = 低功耗 RC 振荡器 (LPRC)  
100 = 辅助振荡器 (SOSC)  
011 = 带有 PLL 模块的主振荡器 (XTPLL、HSPLL 和 ECPLL)  
010 = 主振荡器 (XT、HS 和 EC)  
001 = 带后分频器和 PLL 模块的快速 RC 振荡器 (FRCPLL)  
000 = 快速 RC 振荡器 (FRC)
- bit 7-6 **FCKSM<1:0>:** 时钟切换和故障保护时钟监视器配置位  
1x = 禁止时钟切换和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 = 使能时钟切换和故障保护时钟监视器
- bit 5 **OSCIOFCN:** OSCO 引脚配置位  
如果 POSCMD<1:0> = 11 或 00:  
1 = OSCO/CLKO/RC15 用作 CLK0 (Fosc/2)  
0 = OSCO/CLKO/RA3 用作端口 I/O (RC15)  
如果 POSCMD<1:0> = 10 或 01:  
OSCIOFCN 对 OSCO/CLKO/RC15 没有影响。
- bit 4 **IOL1WAY:** IOLOCK 单次置 1 使能位  
1 = 解锁序列完成后, 可将 IOLOCK 位 (OSCCON<6>) 置 1 一次。一旦置 1, 就不能再次写入外设引脚选择寄存器。  
0 = 解锁序列完成后, 可根据需要将 IOLOCK 位置 1 或清零。

## 寄存器 29-2: CW2: 闪存配置字 2 (续)

bit 3-2 保留: 始终保持为 1

bit 1-0 **POSCMD<1:0>**: 主振荡器配置位

11 = 禁止主振荡器模式

10 = 选择 HS 振荡器模式

01 = 选择 XT 振荡器模式

00 = 选择 EC 振荡器模式

# PIC24FJ128GA310 系列

寄存器 29-3: CW3: 闪存配置字 3

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-1	R/PO-1
WPEND	WPCFG	WPDIS	BOREN	WDTWIN1	WDTWIN0	r	SOSCSEL
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
VBTBOR	WPFP6 <sup>(3)</sup>	WPFP5	WPFP4	WPFP3	WPFP2	WPFP1	WPFP0
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 = 上电复位时的值

1 = 置 1

0 = 清零

x = 未知

- bit 23-16 未实现: 读为 1
- bit 15 **WPEND:** 段写保护结束页选择位  
1 = 受保护的代码段上边界是程序存储器的最后一页; 下边界是 WPFP<6:0> 指定的代码页  
0 = 受保护的代码段下边界是程序存储器的末尾 (000000h); 上边界是 WPFP<6:0> 指定的代码页
- bit 14 **WPCFG:** 配置字代码页写保护选择位  
1 = 最后一页 (位于程序存储器开头) 和闪存配置字不受写保护 <sup>(1)</sup>  
0 = 对最后一页和闪存配置字进行写保护, 前提是 WPDIS = 0
- bit 13 **WPDIS:** 段写保护禁止位  
1 = 禁止段代码保护  
0 = 使能段代码保护; 受保护的段由 WPEND、WPCFG 和 WPFPx 配置位定义
- bit 12 **BOREN:** 欠压复位使能位  
1 = 使能 BOR (除深度休眠外的所有模式)  
0 = 禁止 BOR
- bit 11-10 **WDTWIN<1:0>:** 看门狗定时器窗口宽度选择位  
11 = 25%  
10 = 37.5%  
01 = 50%  
00 = 75%
- bit 9 保留: 始终保持为 1
- bit 8 **SOSCSEL:** SOSC 选择位  
1 = 选择 SOSC 电路  
0 = 数字 (SCLKI) 模式 <sup>(2)</sup>
- bit 7 **VBTBOR:** VBAT BOR 使能位  
1 = 使能 VBAT BOR  
0 = 禁止 VBAT BOR

注 1: 无论 WPCFG 状态如何, 如果 WPEND = 1 或 WPFP 对应配置字页, 将保护配置字页。

2: 请确保在使用该配置时, SCLKI 引脚作为数字输入引脚 (见表 11-1)。

3: 对于 64K 器件: PIC24FJ64GA310、PIC24FJ64GA308 和 PIC24FJ64GA306, bit 6 应保持为 0。

## 寄存器 29-3: CW3: 闪存配置字 3 (续)

bit 6-0

**WPFP<6:0>**: 写保护代码段边界页位 [\(3\)](#)

指定受保护代码段的 256 条指令字页边界。

如果 WPEND = 1:

指定代码保护段的下页边界；最后一页为器件中最后实现的页。

如果 WPEND = 0:

指定代码保护段的上页边界；Page0 为下边界。

**注 1:** 无论 WPCFG 状态如何，如果 WPEND = 1 或 WPFP 对应配置字页，将保护配置字页。

**2:** 请确保在使用该配置时，SCLKI 引脚作为数字输入引脚（见 [表 11-1](#)）。

**3:** 对于 64K 器件：PIC24FJ64GA310、PIC24FJ64GA308 和 PIC24FJ64GA306，bit 6 应保持为 0。

# PIC24FJ128GA310 系列

寄存器 29-4: CW4: 闪存配置字 4

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

r-1	r-1	r-1	r-1	r-1	r-1	r-1	R/PO-1
r	r	r	r	r	r	r	DSSWEN
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
DSWDTEN	DSBOREN	DSWDTOSC	DSWDPS4	DSWDPS3	DSWDPS2	DSWDPS1	DSWDPS0
bit 7							bit 0

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

bit 23-16 未实现: 读为 1

bit 15-9 保留: 读为 1

bit 8 **DSSWEN:** 深度休眠软件控制选择位

1 = 由 DSEN 位使能和控制深度休眠操作

0 = 禁止深度休眠操作

bit 7 **DSWDTEN:** 深度休眠看门狗定时器使能位

1 = 使能深度休眠 WDT

0 = 禁止深度休眠 WDT

bit 6 **DSBOREN:** 深度休眠欠压复位使能位

1 = 深度休眠模式下使能 BOR

0 = 深度休眠模式下禁止 BOR (在其他休眠模式下保持活动状态)

bit 5 **DSWDTOSC:** 深度休眠看门狗定时器时钟选择位

1 = 时钟源为 LPRC

0 = 时钟源为 SOSC

## 寄存器 29-4: CW4: 闪存配置字 4 (续)

bit 4-0

**DSWDPS<4:0>**: 深度休眠看门狗定时器后分频比选择位

11111 = 1:68,719,476,736 (25.7 天)
11110 = 1:34,359,738,368 (12.8 天)
11101 = 1:17,179,869,184 (6.4 天)
11100 = 1:8,589,934592 (77.0 小时)
11011 = 1:4,294,967,296 (38.5 小时)
11010 = 1:2,147,483,648 (19.2 小时)
11001 = 1:1,073,741,824 (9.6 小时)
11000 = 1:536,870,912 (4.8 小时)
10111 = 1:268,435,456 (2.4 小时)
10110 = 1:134,217,728 (72.2 分钟)
10101 = 1:67,108,864 (36.1 分钟)
10100 = 1:33,554,432 (18.0 分钟)
10011 = 1:16,777,216 (9.0 分钟)
10010 = 1:8,388,608 (4.5 分钟)
10001 = 1:4,194,304 (135.3s)
10000 = 1:2,097,152 (67.7s)
01111 = 1:1,048,576 (33.825s)
01110 = 1:524,288 (16.912s)
01101 = 1:262,114 (8.456s)
01100 = 1:131,072 (4.228s)
01011 = 1:65,536 (2.114s)
01010 = 1:32,768 (1.057s)
01001 = 1:16,384 (528.5 ms)
01000 = 1:8,192 (264.3 ms)
00111 = 1:4,096 (132.1 ms)
00110 = 1:2,048 (66.1 ms)
00101 = 1:1,024 (33 ms)
00100 = 1:512 (16.5 ms)
00011 = 1:256 (8.3 ms)
00010 = 1:128 (4.1 ms)
00001 = 1:64 (2.1 ms)
00000 = 1:32 (1 ms)

# PIC24FJ128GA310 系列

寄存器 29-5: DEVID: 器件 ID 寄存器

U-1	U-1	U-1	U-1	U-1	U-1	U-1	U-1
—	—	—	—	—	—	—	—
bit 23							bit 16

R	R	R	R	R	R	R	R
FAMID7	FAMID6	FAMID5	FAMID4	FAMID3	FAMID2	FAMID1	FAMID0
bit 15							bit 8

R	R	R	R	R	R	R	R
DEV7	DEV6	DEV5	DEV4	DEV3	DEV2	DEV1	DEV0
bit 7							bit 0

图注: R = 可读位                            U = 未实现位

bit 23-16 未实现: 读为 1

bit 15-8 **FAMID<7:0>**: 器件系列标识符位

0100 0110 = PIC24FJ128GA310 系列

bit 7-0 **DEV<7:0>**: 单个器件标识符位

1110 0000 = PIC24FJ64GA306

1110 0010 = PIC24FJ128GA306

1110 0100 = PIC24FJ64GA308

1110 0110 = PIC24FJ128GA308

1110 1000 = PIC24FJ64GA310

1110 1010 = PIC24FJ128GA310

寄存器 29-6: 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
—	—	—	—	REV3	REV2	REV1	REVO
bit 7							bit 0

图注: R = 可读位                            U = 未实现位

bit 23-4 未实现: 读为 0

bit 3-0 **REV<3:0>**: 器件版本标识符位

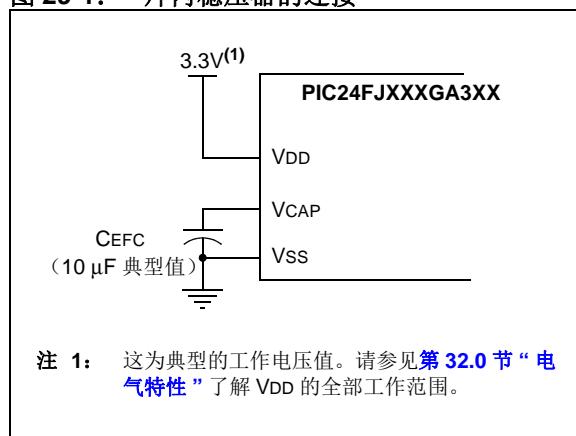
## 29.2 片内稳压器

所有 PIC24FJ128GA310 系列器件均使用标称值为 1.8V 的电压为其内核数字逻辑供电。对于需要工作在一个更高的典型电压值（如 3.3V）的设计来讲，这可能会带来问题。为简化系统设计，PIC24FJ128GA310 系列中的所有器件均包含一个片内稳压器，可使器件内核逻辑工作在 VDD 下。

此稳压器始终使能。它为数字内核逻辑提供恒定电压（标称值 1.8V），范围从约 2.1V 的 VDD 一直到器件的 VDDMAX。该稳压器无法将 VDD 电压提高。为防止用于稳压器的电压降得过低时产生“欠压”条件，此时会发生欠压复位。稳压器输出跟随 VDD，通常比 VDD 小 300 mV。

必须在 VCAP 引脚连接低 ESR 电容（如陶瓷电容）（图 29-1）。这有利于保持稳压器的稳定性。第 32.1 节“直流特性”中提供了该滤波电容的推荐值（CEFC）。

图 29-1：片内稳压器的连接



### 29.2.1 片内稳压器和 POR

稳压器需要约 10  $\mu$ s 的时间来生成输出。在这段称为 TVREG 的时间内，禁止代码执行。每次器件在掉电（包括休眠模式）后恢复工作时都需要 TVREG。TVREG 由 VREGS 位（RCON<8>）和 WDTWIN 配置位（CW3<11:10>）的状态决定。更多有关 TVREG 的信息，请参见第 32.0 节“电气特性”。

**注：**更多信息，请参见第 32.0 节“电气特性”。本数据手册中的信息取代了 FRM 中的信息。

### 29.2.2 稳压器待机模式

片内稳压器除消耗 IDD/IPD 外，还总是会额外消耗一个小的电流，器件工作在休眠模式下也是如此，尽管此时内核数字逻辑并不需要耗能。为了在供电紧张的应用中节省更多的功耗，设置稳压器在器件进入休眠模式时自动进入待机模式。VREGS（RCON<8>）位控制此特性。清零 VREGS 位，使能待机模式。稳压器需要等待 TVREG 时间才能从待机模式唤醒。

### 29.2.3 低压 / 保持稳压器

当使用节能模式（如休眠和深度休眠）时，PIC24FJ128GA310 系列器件可使用单独的低功耗、低压 / 保持稳压器来为关键电路供电。当所有其他数字内核逻辑掉电时，此稳压器（在标称电压 1.2V 下工作）保持对数据 RAM 和 RTCC 供电。它仅在休眠、深度休眠和 VBAT 模式下工作。

有关低压 / 保持稳压器的详细信息，请参见第 10.1.3 节“低压 / 保持稳压器”。

## 29.3 看门狗定时器 (WDT)

PIC24FJ128GA310 系列器件的 WDT 由 LPRC 振荡器驱动。当使能 WDT 时，也将同时使能该时钟源。

由 LPRC 提供的 WDT 时钟源的频率标称值为 31 kHz。将此时钟源提供给可配置为 5 位 (32 分频) 或 7 位 (128 分频) 工作模式的预分频器。分频比由 FWPSA 配置位设置。31 kHz 的输入使预分频器生成标称的 WDT 超时周期 ( $T_{WDT}$ ) ——5 位模式下为 1 ms，7 位模式下为 4 ms。

分频比可变的后分频器对 WDT 预分频器的输出进行分频，从而获得更大范围的超时周期。后分频比由 WDTPS<3:0> 配置位 (CW1<3:0>) 控制，该配置位共允许选择 16 种设置，从 1:1 到 1:32,768。使用预分频器和后分频器后，可获得 1 ms 到 131 秒的超时周期。

WDT、预分频器和后分频器在以下条件下复位：

- 任何器件复位
- 在时钟切换完成时，无论时钟切换是由软件（即改变 NOSC 位后将 OSWEN 位置 1）或是硬件（即故障保护时钟监视器）引起
- 当执行 PWRSAV 指令时（即进入休眠或空闲模式）
- 当器件退出休眠模式或空闲模式恢复正常工作时
- 在正常执行过程中，执行 CLRWDT 指令

如果使能 WDT，它将在休眠或空闲模式下继续运行。当发生 WDT 超时时，将唤醒器件并且代码将从 PWRSAV 指令处继续执行。器件被唤醒后，需要用软件将相应的 SLEEP 或 IDLE 位 (RCON<3:2>) 清零。

WDT 标志位 WDTO (RCON<4>) 不会在 WDT 超时后自动清零。要检测后续的 WDT 事件，必须用软件将该标志清零。

**注：**当执行 CLRWDT 和 PWRSAV 指令时，预分频器和后分频器的计数值将被清零。

### 29.3.1 窗口操作

看门狗定时器具有可选的固定窗口工作模式。在此窗口模式下，CLRWDT 指令只能在编程的 WDT 周期的后 1/4 周期复位 WDT。在该窗口前执行 CLRWDT 指令会导致 WDT 复位，这与 WDT 超时类似。

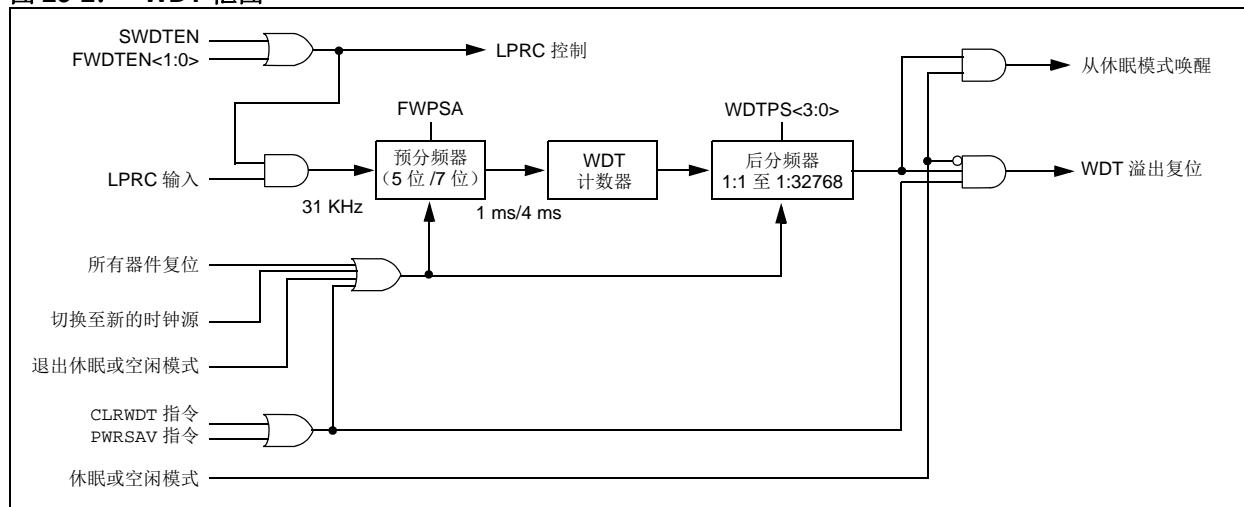
将 WINDIS 配置位 (CW1<7>) 编程为 0 使能 WDT 窗口模式。

### 29.3.2 控制寄存器

通过 FWDTEN<1:0> 配置位使能或禁止 WDT。当配置位 FWDTEN<1:0> = 11 时，始终使能 WDT。

当配置位 FWDTEN<1:0> = 10 时，可以使用软件来控制 WDT。当 FWDTEN<1:0> = 00 时，始终禁止看门狗定时器。通过在软件中将 SWDTEN 控制位 (RCON<5>) 置 1 使能 WDT。任何器件复位都会使 SWDTEN 控制位清零。软件 WDT 选项允许用户在关键代码段使能 WDT 并在非关键代码段禁止 WDT，以最大限度地降低功耗。

图 29-2：WDT 框图



## 29.4 程序校验和代码保护

PIC24FJ128GA310 系列器件提供了两种补充方法防止应用代码被改写和擦除。这两种方法也有助于在运行时防止器件配置被意外更改。

### 29.4.1 通用段保护

PIC24FJ128GA310 系列中所有器件的片内程序存储空间被视作一个存储区，即通用段（General Segment, GS）。配置位 GCP 控制该存储区的代码保护。该位阻止外部对程序存储空间的读写操作。但对正常的执行模式没有直接影响。

写保护由配置字中的 GWRP 位控制。当 GWRP 被编程为 0 时，还禁止在内部对程序存储器执行写和擦除操作。

### 29.4.2 代码段保护

除了全局通用段保护外，还可以单独保护程序存储空间的一部分区域以防止写和擦除操作。在需要对一部分代码进行写和擦除保护（例如自举程序）的时候，该区域有很多用途。与常见的引导区实现方案不同，PIC24FJ128GA310 系列器件中特定的受保护段可被用户放在程序空间的任何位置并配置为各种大小。

代码段保护还增加了对程序存储器的指定区域的新保护级别，这是通过在写或擦除地址发生在指定范围内时禁止 NVM 安全互锁实现的。它不会改写 GCP 或 GWRP 位控制的通用段保护。例如，如果使能了 GCP 和 GWRP，则使能程序存储器下半部分的段代码保护不会撤消对上半部分的通用段保护。

段代码保护范围的大小和类型由配置字 3 的 WPFPx、WPEND、WPCFG 和 WPDIS 位配置。将 WPDIS 位编程为 0 以使能代码段保护。WPFP 位通过指定 512 字节代码页（即受保护段的开始或结束位置）指定受保护的段大小。因为包含指定的存储区，所以，此页也将受到保护。

WPEND 位决定受保护段是否使用程序空间的开头或末尾作为边界。将 WPEND 编程为 0 可设置程序存储器的末尾（000000h）作为受保护段的下边界。保留 WPEND 未编程（即 = 1）将保护指定页到已实现程序存储器的最后一页（包含配置字单元）。

独立位 WPCFG 用于单独保护程序空间的最后一页（包含闪存配置字）。将 WPCFG 编程为 0 可保护最后一页，以及由 WPEND 和 WPFP<6:0> 位设置所选择的页。这可用在需要对存储器末尾的代码段以及闪存配置字进行写保护的情况下。

表 29-2 中给出了段代码保护的各种选项。

表 29-2：代码段保护配置选项

段配置位			代码段的写 / 擦除保护
WPDIS	WPEND	WPCFG	
1	x	x	未使能其他保护；所有程序存储器保护都由 GCP 和 GWRP 配置。
0	1	x	写 / 擦除保护范围从 WPFP<7:0> 定义的代码页的第一个地址到已实现程序存储器的末尾（包含末尾），包含闪存配置字。
0	0	1	写 / 擦除保护范围从地址 000000h 到 WPFP<7:0> 定义的代码页的最后一个地址（包含该地址）。
0	0	0	写 / 擦除保护范围从地址 000000h 到 WPFP<7:0> 定义的代码页的最后一个地址（包含该地址）以及最后一页（包含闪存配置字）。

### 29.4.3 配置寄存器保护

有两种方法保护配置寄存器使其免遭无意的或不期望的更改或读取。主要的保护方式与保护 RP 寄存器的方法相同——影子寄存器中包含了一个基准值，持续将该值与实际的值进行比较。

出于从防范不可预见事件方面的考虑，由于电池故障（如 ESD 事件）引起的配置位更改将导致奇偶校验错误并触发器件复位。

配置寄存器的数据来自于程序存储器中的闪存配置字。当 GCP 位置 1 时，也将保护器件配置的源数据。即使未使能通用段保护，使用适当的代码段保护设置也可以保护器件配置。

### 29.5 JTAG 接口

PIC24FJ128GA310 系列器件实现了 JTAG 接口，该接口支持边界扫描器件测试。

### 29.6 在线串行编程

PIC24FJ128GA310 系列单片机可以在最终的应用电路中进行串行编程。只需要 5 根线即可完成这一操作，其中时钟线（PGEC<sub>x</sub>）和数据线（PGED<sub>x</sub>）各一根，其余 3 根分别是电源线（V<sub>DD</sub>）、接地线（V<sub>SS</sub>）和 MCLR。这允许用户使用未编程器件制造电路板，仅在产品交付前才对单片机进行编程。从而可以将最新版本的固件或定制固件烧写到单片机中。

### 29.7 在线调试器

当选择 MPLAB® ICD 3 作为调试器时，使能在线调试功能。这一功能允许结合 MPLAB IDE 进行一些简单的调试。通过 PGEC<sub>x</sub>（仿真 / 调试时钟）和 PGED<sub>x</sub>（仿真 / 调试数据）引脚控制调试功能。

要使用器件的在线调试功能，在设计中必须实现至 MCLR、V<sub>DD</sub>、V<sub>SS</sub> 和 ICS 配置位指定的 PGEC<sub>x</sub>/PGED<sub>x</sub> 引脚对的 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 个字节和两个 I/O 引脚。

## 30.0 开发支持

一系列软件及硬件开发工具对 PIC® 单片机和 dsPIC® 数字信号控制器提供支持：

- 集成开发环境
  - MPLAB® IDE 软件
- 编译器 / 汇编器 / 链接器
  - 适用于各种器件系列的 MPLAB C 编译器
  - 适用于各种器件系列的 HI-TECH C® 编译器
  - MPASM™ 汇编器
  - MPLINK™ 目标链接器 /  
MPLIB™ 目标库管理器
  - 适用于各种器件系列的 MPLAB 汇编器 / 链接器 / 库管理器
- 模拟器
  - MPLAB SIM 软件模拟器
- 仿真器
  - MPLAB REAL ICE™ 在线仿真器
- 在线调试器
  - MPLAB ICD 3
  - PICkit™ 3 Debug Express
- 器件编程器
  - PICkit™ 2 编程器
  - MPLAB PM3 器件编程器
- 低成本演示 / 开发板、评估工具包及入门工具包

## 30.1 MPLAB 集成开发环境软件

MPLAB IDE 软件为 8/16/32 位单片机市场提供了前所未有的易于使用的软件开发平台。MPLAB IDE 是基于 Windows® 操作系统的应用软件，包括：

- 一个包含所有调试工具的图形界面
    - 模拟器
    - 编程器（单独销售）
    - 在线仿真器（单独销售）
    - 在线调试器（单独销售）
  - 具有彩色上下文代码显示的全功能编辑器
  - 多项目管理器
  - 内容可直接编辑的可定制式数据窗口
  - 高级源代码调试
  - 鼠标停留在变量上进行查看的功能
  - 将变量从源代码窗口拖放到 Watch（观察）窗口
  - 丰富的在线帮助
  - 集成了可选的第三方工具，如 IAR C 编译器
- MPLAB IDE 可以让您：
- 编辑源文件（C 语言或汇编语言）
  - 点击一次即可完成编译或汇编，并将代码下载到仿真器和模拟器工具中（自动更新所有项目信息）
  - 可使用如下各项进行调试：
    - 源文件（C 语言或汇编语言）
    - 混合 C 语言和汇编语言
    - 机器码

MPLAB IDE 在单个开发范例中支持使用多种调试工具，包括从成本效益高的模拟器到低成本的在线调试器，再到全功能的仿真器。这样缩短了用户升级到更加灵活而功能强大的工具时的学习时间。

## 30.2 适用于各种器件系列的 MPLAB C 编译器

MPLAB C 编译器代码开发系统是完全的 ANSI C 编译器，适用于 Microchip 的 PIC18、PIC24 和 PIC32 系列单片机及 dsPIC30 和 dsPIC33 系列数字信号控制器。这些编译器提供强大的集成功能和出众的代码优化能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

## 30.3 适用于各种器件系列的 HI-TECH C 编译器

HI-TECH C 编译器代码开发系统是完全的 ANSI C 编译器，适用于 Microchip 的 PIC 系列单片机及 dsPIC 系列数字信号控制器。这些编译器提供强大的集成功能和全知代码生成能力，且使用方便。

为便于源代码调试，编译器提供针对 MPLAB IDE 调试器优化的符号信息。

编译器包括一个宏汇编器、链接器、预处理程序和单步驱动程序，可以在多种平台上运行。

## 30.4 MPASM 汇编器

MPASM 汇编器是全功能通用宏汇编器，适用于 PIC10/12/16/18 MCU。

MPASM 汇编器可生成用于 MPLINK 目标链接器的可重定位目标文件、Intel® 标准 HEX 文件、详细描述存储器使用状况和符号参考的 MAP 文件、包含源代码行及生成机器码的绝对 LST 文件以及用于调试的 COFF 文件。

MPASM 汇编器具有如下特性：

- 集成在 MPLAB IDE 项目中
- 用户定义的宏可简化汇编代码
- 对多用途源文件进行条件汇编
- 允许完全控制汇编过程的指令

## 30.5 MPLINK 目标链接器 / MPLIB 目标库管理器

MPLINK 目标链接器包含了由 MPASM 汇编器、MPLAB C18 C 编译器产生的可重定位目标。通过使用链接器脚本中的指令，它还可链接预编译库中的可重定位目标。

MPLIB 目标库管理器管理预编译代码库文件的创建和修改。当从源文件调用库中的一段子程序时，只有包含此子程序的模块被链接到应用程序。这样可使大型库在许多不同应用中被高效地利用。

目标链接器 / 库管理器具有如下特性：

- 高效地连接单个的库而不是许多小文件
- 通过将相关的模块组合在一起增强代码的可维护性
- 只要列出、替换、删除和抽取模块，便可灵活地创建库

## 30.6 适用于各种器件系列的 MPLAB 汇编器、链接器和库管理器

MPLAB 汇编器为 PIC24、PIC32 和 dsPIC 器件从符号汇编语言生成可重定位机器码。MPLAB C 编译器使用该汇编器生成目标文件。汇编器产生可重定位目标文件之后，可将这些目标文件存档，或与其他可重定位目标文件和存档链接以生成可执行文件。该汇编器有如下显著特性：

- 支持整个器件指令集
- 支持定点数据和浮点数据
- 命令行界面
- 丰富的指令集
- 灵活的宏语言
- MPLAB IDE 兼容性

## 30.7 MPLAB SIM 软件模拟器

MPLAB SIM 软件模拟器通过在指令级对 PIC MCU 和 dsPIC® DSC 进行模拟，可在 PC 主机环境下进行代码开发。对于任何给定的指令，都可以对数据区进行检查或修改，并通过一个全面的激励控制器来施加激励。可以将各寄存器记录在文件中，以便进行进一步的运行时分析。跟踪缓冲区和逻辑分析器的显示使软件模拟器还能记录和跟踪程序的执行、I/O 的动作、大部分的外设及内部寄存器。

MPLAB SIM 软件模拟器完全支持使用 MPLAB C 编译器以及 MPASM 和 MPLAB 汇编器的符号调试。该软件模拟器可用于在硬件实验室环境外灵活地开发和调试代码，是一款完美且经济的软件开发工具。

## 30.8 MPLAB REAL ICE 在线仿真器系统

MPLAB REAL ICE 在线仿真器系统是 Microchip 针对其闪存 DSC 和 MCU 器件而推出的新一代高速仿真器。结合 MPLAB 集成开发环境（IDE）所具有的易于使用且功能强大的图形用户界面，该仿真器可对 PIC® 闪存 MCU 和 dsPIC® 闪存 DSC 进行调试和编程。IDE 是随每个工具包一起提供的。

该仿真器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与在线调试器系统兼容的连接器（RJ11）或新型抗噪声、高速低压差分信号（LVDS）互连电缆（CAT5）与目标板相连。

可通过 MPLAB IDE 下载将来版本的固件，对该仿真器进行现场升级。在即将推出的 MPLAB IDE 版本中，会支持许多新器件，还将增加一些新特性。在同类仿真器中，MPLAB REAL ICE 的优势十分明显：低成本、全速仿真、运行时变量查看、跟踪分析、复杂断点、耐用的探针接口及较长（长达 3 米）的互连电缆。

## 30.9 MPLAB ICD 3 在线调试器系统

MPLAB ICD 3 在线调试器系统是 Microchip 成本效益最高的高速硬件调试器 / 编程器，适用于 Microchip 闪存数字信号控制器（DSC）和单片机（MCU）器件。结合 MPLAB 集成开发环境（IDE）所具有的功能强大但易于使用的图形用户界面，该调试器可对 PIC® 闪存单片机和 dsPIC® DSC 进行调试和编程。

MPLAB ICD 3 在线调试器通过高速 USB 2.0 接口与设计工程师的 PC 相连，并利用与 MPLAB ICD 2 或 MPLAB REAL ICE 系统兼容的连接器（RJ-11）与目标板相连。MPLAB ICD 3 支持所有 MPLAB ICD 2 转接器。

## 30.10 PICkit 3 在线调试器 / 编程器及 PICkit 3 Debug Express

结合 MPLAB 集成开发环境（IDE）所具有的功能强大的图形用户界面，MPLAB PICkit 3 可对 PIC® 闪存单片机和 dsPIC® 数字信号控制器进行调试和编程，且价位较低。MPLAB PICkit 3 通过全速 USB 接口与设计工程师的 PC 相连，并利用 Microchip 调试（RJ-11）连接器（与 MPLAB ICD 3 和 MPLAB REAL ICE 兼容）与目标板相连。连接器使用两个器件 I/O 引脚和复位线来实现在线调试和在线串行编程。

PICkit 3 Debug Express 包括 PICkit 3、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器和 MPLAB IDE 软件）。

# PIC24FJ128GA310 系列

---

## 30.11 PICkit 2 开发编程器 / 调试器及 PICkit 2 Debug Express

PICkit™ 2 开发编程器 / 调试器是一款低成本开发工具，具有易于使用的界面，适用于对 Microchip 的闪存系列单片机进行编程和调试。这一全功能的 Windows® 编程界面支持低档 (PIC10F、PIC12F5xx 和 PIC16F5xx)、中档 (PIC12F6xx 和 PIC16F)、PIC18F、PIC24、dsPIC30、dsPIC33 和 PIC32 系列的 8 位、16 位及 32 位单片机，以及许多 Microchip 串行 EEPROM 产品。结合 Microchip 功能强大的 MPLAB 集成开发环境 (IDE)，PICkit 2 可对大多数 PIC® 单片机进行在线调试。即使 PIC 单片机已嵌入应用，在线调试功能仍可以运行、暂停和单步执行程序。在断点处暂停时，可以检查和修改文件寄存器。

PICkit 2 Debug Express 包括 PICkit 2、演示板和单片机、连接电缆和光盘（内含用户指南、课程、教程、编译器和 MPLAB IDE 软件）。

## 30.12 MPLAB PM3 器件编程器

MPLAB PM3 器件编程器是一款符合 CE 规范的通用器件编程器，在 VDDMIN 和 VDDMAX 点对其可编程电压进行校验以确保可靠性最高。它有一个用来显示菜单和错误消息的大 LCD 显示器 (128 x 64)，以及一个支持各种封装类型的可拆卸模块化插槽装置。编程器标准配置中带有一根 ICSPTM 电缆。在单机模式下，MPLAB PM3 器件编程器不必与 PC 相连即可对 PIC 器件进行读取、校验和编程。在该模式下它还可设置代码保护。MPLAB PM3 通过 RS-232 或 USB 电缆连接到 PC 主机上。MPLAB PM3 具备高速通信能力以及优化算法，可对具有大存储器的器件进行快速编程。它还包含了 MMC 卡，用于文件存储及数据应用。

## 30.13 演示 / 开发板、评估工具包及入门工具包

有许多演示、开发和评估板可用于各种 PIC MCU 和 dsPIC DSC，实现对全功能系统的快速应用开发。大多数的演示、开发和评估板都有实验布线区，供用户添加定制电路；还有应用固件和源代码，用于检查和修改。

这些板支持多种功能部件，包括 LED、温度传感器、开关、扬声器、RS-232 接口、LCD 显示器、电位计和附加 EEPROM 存储器。

演示和开发板可用于教学环境，在实验布线区设计定制电路，从而掌握各种单片机应用。

除了 PICDEM™ 和 dsPICDEM™ 演示 / 开发板系列电路外，Microchip 还有一系列评估工具包和演示软件，适用于模拟滤波器设计、KEELOQ® 数据安全产品 IC、CAN、IrDA®、PowerSmart 电池管理、SEEVVAL® 评估系统、Σ-Δ ADC、流速传感器，等等。

同时还提供入门工具包，其中包含体验指定器件功能所需的所有软硬件。通常提供单个应用以及调试功能，都包含在一块电路板上。

有关演示、开发和评估工具包的完整列表，请访问 Microchip 网站 ([www.microchip.com](http://www.microchip.com))。

## 31.0 指令集汇总

**注:** 本章是 PIC24F 指令集架构的简要汇总，并不能作为详尽的参考资料使用。

PIC24F 指令集与以前的 PIC® MCU 指令集相比，添加了许多增强功能，并保持了易于从以前的 PIC MCU 指令集移植的特点。大部分指令只占用一个程序存储字。只有 3 条指令需要两个程序存储单元。

每条单字指令都是一个 24 位字，由一个 8 位的操作码（指明指令类型）和一个或多个操作数（指定指令具体操作）组成。整个指令集具有高度的正交性，分为以下 4 种基本类型：

- 面向字或字节的操作类指令
- 面向位的操作类指令
- 立即数操作类指令
- 控制操作类指令

表31-1给出了在说明指令时要用到的通用符号。表31-2 中的 PIC24F 指令集汇总列出了所有指令及每条指令影响的状态标志。

大部分面向字或字节的 W 寄存器指令（包含桶形移位寄存器指令）含有三个操作数：

- 第一个源操作数通常是不带地址修改符的 “Wb” 寄存器
- 第二个源操作数通常是带或不带地址修改符的 “Ws” 寄存器
- 结果的目标地址通常是带或不带地址修改符的 “Wd” 寄存器

而面向字或字节的文件寄存器指令具有两个操作数：

- 文件寄存器（由 “f” 值指定）
- 目标寄存器（可以是文件寄存器 “f” 也可以是记作 “WREG”的 W0 寄存器）

大部分面向位的操作类指令（包括简单翻转 / 移位指令）含有两个操作数：

- W 寄存器（带或不带地址修改符）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（由立即数直接指定或由 “Wb” 寄存器中的内容间接指定）

涉及数据传送的立即数指令可能使用以下操作数：

- 将被装载到 W 寄存器或文件寄存器的立即数（由 “K” 的值指定）
- 将要装载立即数的 W 寄存器或文件寄存器（由 “Wb” 或 “f” 指定）

而涉及算术或逻辑运算的立即数指令使用以下操作数：

- 第一个源操作数是不带地址修改符的 “Wb” 寄存器
- 第二个源操作数是立即数
- 结果的目标地址（只有在与第一个源操作数不同的情况下）通常为带或不带地址修改符的 “Wd” 寄存器

控制操作类指令可使用以下操作数：

- 程序存储器地址
- 表读和表写指令的模式

除某些双字指令外所有指令都是单字指令。双字指令中所有必需的信息都在这 48 位中，第二个字的高 8 位全为 0。如果指令自身将第二个字当作一条指令来执行的话，它将作为一条 NOP 指令来执行。

执行大部分单字指令都只需要一个指令周期，除非条件测试结果为真或者指令执行改变了程序计数器的值。对于上述两种特殊情况，指令执行需要两个指令周期，在第二个指令周期中执行一条 NOP 指令。值得注意的特殊指令有 BRA（无条件/计算转移指令）、间接 CALL/GOTO 指令、所有表读和表写指令以及 RETURN/RETFIE 指令，这些指令都是单字指令，但执行起来需要 2 或 3 个指令周期。

某些涉及跳过下一条指令的指令，在执行跳过时需要两或三个指令周期，具体周期数取决于被跳过的指令是单字指令还是双字指令。此外需要传送两个字的指令需要两个周期。执行双字指令需要两个指令周期。

# PIC24FJ128GA310 系列

表 31-1：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“地址为 text 的单元”
{ }	可选字段或操作
<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（默认情况）
bit4	4 位位选择字段（用于字寻址指令） $\in \{0\ldots15\}$
C、DC、N、OV 和 Z	MCU 状态位：进位、半进位、负标志、溢出标志和全零标志
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 $\in \{0000h\ldots1FFFh\}$
lit1	1 位无符号立即数 $\in \{0,1\}$
lit4	4 位无符号立即数 $\in \{0\ldots15\}$
lit5	5 位无符号立即数 $\in \{0\ldots31\}$
lit8	8 位无符号立即数 $\in \{0\ldots255\}$
lit10	10 位无符号立即数，字节模式下， $\in \{0\ldots255\}$ ，字模式下 $\in \{0:1023\}$
lit14	14 位无符号立即数 $\in \{0\ldots16383\}$
lit16	16 位无符号立即数 $\in \{0\ldots65535\}$
lit23	23 位无符号立即数 $\in \{0\ldots8388607\}$ ； LSB 必须为 0
无	该字段不必有输入项，可以为空白
PC	程序计数器
Slit10	10 位有符号立即数 $\in \{-512\ldots511\}$
Slit16	16 位有符号立即数 $\in \{-32768\ldots32767\}$
Slit6	6 位有符号立即数 $\in \{-16\ldots16\}$
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）
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 $\in \{Wns, [Wns], [Wns++], [Wns-], [+Wns], [-Wns], [Wns+Wb]\}$

表 31-2：指令集汇总

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
ADD	ADD f	f = f + WREG	1	1	C、DC、N、OV 和 Z
	ADD f,WREG	WREG = f + WREG	1	1	C、DC、N、OV 和 Z
	ADD #lit10,Wn	Wd = lit10 + Wd	1	1	C、DC、N、OV 和 Z
	ADD Wb,Ws,Wd	Wd = Wb + Ws	1	1	C、DC、N、OV 和 Z
	ADD Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C、DC、N、OV 和 Z
ADDC	ADDC f	f = f + WREG + (C)	1	1	C、DC、N、OV 和 Z
	ADDC f,WREG	WREG = f + WREG + (C)	1	1	C、DC、N、OV 和 Z
	ADDC #lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C、DC、N、OV 和 Z
	ADDC Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C、DC、N、OV 和 Z
	ADDC Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C、DC、N、OV 和 Z
AND	AND f	f = f .AND.WREG	1	1	N 和 Z
	AND f,WREG	WREG = f .AND.WREG	1	1	N 和 Z
	AND #lit10,Wn	Wd = lit10 .AND.Wd	1	1	N 和 Z
	AND Wb,Ws,Wd	Wd = Wb .AND.Ws	1	1	N 和 Z
	AND Wb,#lit5,Wd	Wd = Wb .AND.lit5	1	1	N 和 Z
ASR	ASR f	f = 算术右移 f	1	1	C、N、OV 和 Z
	ASR f,WREG	WREG = 算术右移 f	1	1	C、N、OV 和 Z
	ASR Ws,Wd	Wd = 算术右移 Ws	1	1	C、N、OV 和 Z
	ASR Wb,Wns,Wnd	Wnd = 将 Wb 算术右移 Wns 位	1	1	N 和 Z
	ASR Wb,#lit5,Wnd	Wnd = 将 Wb 算术右移 lit5 位	1	1	N 和 Z
BCLR	BCLR f,#bit4	将 f 寄存器中的某位清零	1	1	无
	BCLR Ws,#bit4	将 Ws 中的指定位清零	1	1	无
BRA	BRA C,Expr	进位则跳转	1	1 (2)	无
	BRA GE,Expr	如果大于或等于则跳转	1	1 (2)	无
	BRA GEU,Expr	如果无符号大于或等于则跳转	1	1 (2)	无
	BRA GT,Expr	如果大于则跳转	1	1 (2)	无
	BRA GTU,Expr	如果无符号大于则跳转	1	1 (2)	无
	BRA LE,Expr	如果小于或等于则跳转	1	1 (2)	无
	BRA LEU,Expr	如果无符号小于或等于则跳转	1	1 (2)	无
	BRA LT,Expr	如果小于则跳转	1	1 (2)	无
	BRA LTU,Expr	如果无符号小于则跳转	1	1 (2)	无
	BRA N,Expr	为负则跳转	1	1 (2)	无
	BRA NC,Expr	无进位则跳转	1	1 (2)	无
	BRA NN,Expr	不为负则跳转	1	1 (2)	无
	BRA NOV,Expr	不溢出则跳转	1	1 (2)	无
	BRA NZ,Expr	不为零则跳转	1	1 (2)	无
	BRA OV,Expr	溢出则跳转	1	1 (2)	无
BSET	BSET f,#bit4	将 f 中的某位置 1	1	1	无
	BSET Ws,#bit4	将 Ws 中的指定位置 1	1	1	无
BSW	BSW.C Ws,Wb	将 C 位内容写入 Ws<Wb>	1	1	无
	BSW.Z Ws,Wb	将 Z 位内容写入 Ws<Wb>	1	1	无
BTG	BTG f,#bit4	将 f 中的某位取反	1	1	无
	BTG Ws,#bit4	将 Ws 中的某位取反	1	1	无
BTSC	BTSC f,#bit4	检测 f 中的某位，为 0 则跳过	1	1 (2 或 3)	无
	BTSC Ws,#bit4	检测 Ws 中的某位，为 0 则跳过	1	1 (2 或 3)	无

# PIC24FJ128GA310 系列

表 31-2：指令集汇总（续）

汇编指令 助记符	汇编语法	说明	字数	周期数	受影响的状态标志
BTSS	BTSS f,#bit4	检测 f 中的某位, 为 1 则跳过	1	1 (2 或 3)	无
	BTSS Ws,#bit4	检测 Ws 中的某位, 为 1 则跳过	1	1 (2 或 3)	无
BTST	BTST f,#bit4	对 f 中的指定位进行检测	1	1	Z
	BTST.C Ws,#bit4	对 Ws 中的指定位进行检测, 并将结果存储到进位标志位 C 中	1	1	C
	BTST.Z Ws,#bit4	对 Ws 中的指定位进行检测, 并将结果存储到全零标志位 Z 中	1	1	Z
	BTST.C Ws,Wb	对 Ws<Wb> 位进行检测, 并将结果存储到进位标志位 C 中	1	1	C
	BTST.Z Ws,Wb	对 Ws<Wb> 位进行检测, 并将结果存储到全零标志位 Z 中	1	1	Z
BTSTS	BTSTS f,#bit4	对 f 中的指定位进行检测, 并将 f 置为全 1	1	1	Z
	BTSTS.C Ws,#bit4	对 Ws 中的指定位进行检测, 并将结果存储到进位标志位 C 中, 然后将 Ws 中的该位置 1	1	1	C
	BTSTS.Z Ws,#bit4	对 Ws 中的指定位进行检测, 并将结果存储到全零标志位 Z 中, 然后将 Ws 中的该位置 1	1	1	Z
CALL	CALL lit23	调用子程序	2	2	无
	CALL Wn	间接调用子程序	1	2	无
CLR	CLR f	f = 0x0000	1	1	无
	CLR WREG	WREG = 0x0000	1	1	无
	CLR Ws	Ws = 0x0000	1	1	无
CLRWDT	CLRWDT	清零看门狗定时器	1	1	WDTO 和 Sleep
COM	COM f	f = $\bar{f}$	1	1	N 和 Z
	COM f,WREG	WREG = $\bar{f}$	1	1	N 和 Z
	COM Ws,Wd	Wd = $\bar{Ws}$	1	1	N 和 Z
CP	CP f	比较 f 和 WREG	1	1	C、DC、N、OV 和 Z
	CP Wb,#lit5	比较 Wb 和 lit5	1	1	C、DC、N、OV 和 Z
	CP Wb,Ws	比较 Wb 和 Ws (Wb - Ws)	1	1	C、DC、N、OV 和 Z
CP0	CP0 f	比较 f 和 0x0000	1	1	C、DC、N、OV 和 Z
	CP0 Ws	比较 Ws 和 0x0000	1	1	C、DC、N、OV 和 Z
CPB	CPB f	比较 f 和 WREG (通过减法实现)	1	1	C、DC、N、OV 和 Z
	CPB Wb,#lit5	比较 Wb 和 lit5 (通过减法实现)	1	1	C、DC、N、OV 和 Z
	CPB Wb,Ws	比较 Wb 和 Ws (Wb - Ws - C) (通过减法实现)	1	1	C、DC、N、OV 和 Z
CPSEQ	CPSEQ Wb,Wn	比较 Wb 和 Wn, 如果相等则跳过	1	1 (2 或 3)	无
CPSGT	CPSGT Wb,Wn	比较 Wb 和 Wn, 如果大于则跳过	1	1 (2 或 3)	无
CPSLT	CPSLT Wb,Wn	比较 Wb 和 Wn, 如果小于则跳过	1	1 (2 或 3)	无
CPSNE	CPSNE Wb,Wn	比较 Wb 和 Wn, 如果不相等则跳过	1	1 (2 或 3)	无
DAW	DAW.B Wn	Wn = 对 Wn 进行十进制调整	1	1	C
DEC	DEC f	f = f - 1	1	1	C、DC、N、OV 和 Z
	DEC f,WREG	WREG = f - 1	1	1	C、DC、N、OV 和 Z
	DEC Ws,Wd	Wd = Ws - 1	1	1	C、DC、N、OV 和 Z
DEC2	DEC2 f	f = f - 2	1	1	C、DC、N、OV 和 Z
	DEC2 f,WREG	WREG = f - 2	1	1	C、DC、N、OV 和 Z
	DEC2 Ws,Wd	Wd = Ws - 2	1	1	C、DC、N、OV 和 Z
DISI	DISI #lit14	在 k 个指令周期内禁止中断	1	1	无
DIV	DIV.SW Wm,Wn	有符号 16/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.SD Wm,Wn	有符号 32/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UW Wm,Wn	无符号 16/16 位整数除法	1	18	N、Z、C 和 OV
	DIV.UD Wm,Wn	无符号 32/16 位整数除法	1	18	N、Z、C 和 OV
EXCH	EXCH Wns,Wnd	将 Wns 和 Wnd 交换	1	1	无
FF1L	FF1L Ws,Wnd	从左边 (MSb) 查找第一个 1	1	1	C
FF1R	FF1R Ws,Wnd	从右边 (LSb) 查找第一个 1	1	1	C

表 31-2：指令集汇总（续）

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
GOTO	GOTO Expr	转移到地址	2	2	无
	GOTO Wn	间接转移到地址	1	2	无
INC	INC f	f = f + 1	1	1	C、DC、N、OV 和 Z
	INC f, WREG	WREG = f + 1	1	1	C、DC、N、OV 和 Z
	INC Ws, Wd	Wd = Ws + 1	1	1	C、DC、N、OV 和 Z
INC2	INC2 f	f = f + 2	1	1	C、DC、N、OV 和 Z
	INC2 f, WREG	WREG = f + 2	1	1	C、DC、N、OV 和 Z
	INC2 Ws, Wd	Wd = Ws + 2	1	1	C、DC、N、OV 和 Z
IOR	IOR f	f = f.IOR.WREG	1	1	N 和 Z
	IOR f, WREG	WREG = f.IOR.WREG	1	1	N 和 Z
	IOR #lit10, Wn	Wd = lit10.IOR.Wd	1	1	N 和 Z
	IOR Wb, Ws, Wd	Wd = Wb.IOR.Ws	1	1	N 和 Z
	IOR Wb, #lit5, Wd	Wd = Wb.IOR.lit5	1	1	N 和 Z
LNK	LINK #lit14	链接帧指针	1	1	无
LSR	LSR f	f = 逻辑右移 f	1	1	C、N、OV 和 Z
	LSR f, WREG	WREG = 逻辑右移 f	1	1	C、N、OV 和 Z
	LSR Ws, Wd	Wd = 逻辑右移 Ws	1	1	C、N、OV 和 Z
	LSR Wb, Wns, Wnd	Wnd = 将 Wb 逻辑右移 Wns 位	1	1	N 和 Z
	LSR Wb, #lit5, Wnd	Wnd = 将 Wb 逻辑右移 lit5 位	1	1	N 和 Z
MOV	MOV f, Wn	将 f 中的内容送入 Wn	1	1	无
	MOV [Wns+Slit10], Wnd	将 [Wns+Slit10] 中的内容送入 Wnd	1	1	无
	MOV f	将源寄存器的内容送入目标寄存器	1	1	N 和 Z
	MOV f, WREG	将 f 中的内容送入 WREG	1	1	N 和 Z
	MOV #lit16, Wn	将 16 位立即数送入 Wn	1	1	无
	MOV.b #lit8, Wn	将 8 位立即数送入 Wn	1	1	无
	MOV Wn, f	将 Wn 中的内容送入 f	1	1	无
	MOV Wns, [Wns+Slit10]	将 Wns 中的内容送入 [Wns+Slit10]	1	1	无
	MOV Wso, Wdo	将 Ws 中的内容送入 Wd	1	1	无
	MOV WREG, f	将 WREG 内容传送到 f	1	1	N 和 Z
	MOV.D Wns, Wd	将 W(ns):W(ns + 1) 中的双字内容送入 Wd	1	2	无
MUL	MOV.D Ws, Wnd	将 Ws 中的双字内容送入 W(nd + 1):W(nd)	1	2	无
	MUL.SS Wb, Ws, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Signed(Ws)	1	1	无
	MUL.SU Wb, Ws, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(Ws)	1	1	无
	MUL.US Wb, Ws, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Signed(Ws)	1	1	无
	MUL.UU Wb, Ws, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(Ws)	1	1	无
	MUL.SU Wb, #lit5, Wnd	{Wnd+1, Wnd} = Signed(Wb) * Unsigned(lit5)	1	1	无
	MUL.UU Wb, #lit5, Wnd	{Wnd+1, Wnd} = Unsigned(Wb) * Unsigned(lit5)	1	1	无
NEG	MUL f	W3:W2 = f * WREG	1	1	无
	NEG f	f = f + 1	1	1	C、DC、N、OV 和 Z
	NEG f, WREG	WREG = f + 1	1	1	C、DC、N、OV 和 Z
NOP	NEG Ws, Wd	Wd = Ws + 1	1	1	C、DC、N、OV 和 Z
	NOP	空操作	1	1	无
POP	NOPR	空操作	1	1	无
	POP f	从栈顶 (TOS) 弹出 f 寄存器的内容	1	1	无
POP	POP Wdo	将栈顶 (TOS) 的内容弹出到 Wdo 中	1	1	无
	POP.D Wnd	将栈顶 (TOS) 的内容弹出到 W(nd):W(nd+1) 中	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
	POP.W	将 W(ns):W(ns + 1) 中的内容弹出到影子寄存器	1	2	无
PUSH	PUSH f	将 f 的内容压入栈顶 (TOS)	1	1	无
	PUSH Wso	将 Wso 的内容压入栈顶 (TOS)	1	1	无
	PUSH.D Wns	将 W(ns):W(ns + 1) 中的内容压入栈顶 (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

# PIC24FJ128GA310 系列

表 31-2：指令集汇总（续）

汇编指令 助记符	汇编语法	说明	字数	周期数	受影响的状态标志
PWRSAV	PWRSAV #lit1	进入休眠或空闲模式	1	1	WDTO 和 Sleep
RCALL	RCALL Expr	相对调用	1	2	无
	RCALL Wn	计算调用	1	2	无
REPEAT	REPEAT #lit14	将下一条指令重复执行 lit14 + 1 次	1	1	无
	REPEAT Wn	将下一条指令重复执行 (Wn) + 1 次	1	1	无
RESET	RESET	软件器件复位	1	1	无
RETFIE	RETFIE	从中断返回	1	3 (2)	无
RETLW	RETLW #lit10,Wn	返回并将立即数存入 Wn	1	3 (2)	无
RETURN	RETURN	从子程序返回	1	3 (2)	无
RLC	RLC f	f = 对 f 执行带进位的循环左移	1	1	C、N 和 Z
	RLC f,WREG	WREG = 对 f 执行带进位的循环左移	1	1	C、N 和 Z
	RLC Ws,Wd	Wd = 对 Ws 执行带进位的循环左移	1	1	C、N 和 Z
RLNC	RLNC f	f = 循环左移 f (不带进位)	1	1	N 和 Z
	RLNC f,WREG	WREG = 循环左移 f (不带进位)	1	1	N 和 Z
	RLNC Ws,Wd	Wd = 循环左移 Ws (不带进位)	1	1	N 和 Z
RRC	RRC f	f = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC f,WREG	WREG = 对 f 执行带进位的循环右移	1	1	C、N 和 Z
	RRC Ws,Wd	Wd = 对 Ws 执行带进位的循环右移	1	1	C、N 和 Z
RRNC	RRNC f	f = 循环右移 f (不带进位)	1	1	N 和 Z
	RRNC f,WREG	WREG = 循环右移 f (不带进位)	1	1	N 和 Z
	RRNC Ws,Wd	Wd = 循环右移 Ws (不带进位)	1	1	N 和 Z
SE	SE Ws,Wnd	Wnd = 对 Ws 进行符号扩展	1	1	C、N 和 Z
SETM	SETM f	f = FFFFh	1	1	无
	SETM WREG	WREG = FFFFh	1	1	无
	SETM Ws	Ws = FFFFh	1	1	无
SL	SL f	f = 左移 f	1	1	C、N、OV 和 Z
	SL f,WREG	WREG = 左移 f	1	1	C、N、OV 和 Z
	SL Ws,Wd	Wd = 左移 Ws	1	1	C、N、OV 和 Z
	SL Wb,Wns,Wnd	Wnd = 将 Wb 左移 Wns 位	1	1	N 和 Z
	SL Wb,#lit5,Wnd	Wnd = 将 Wb 左移 lit5 位	1	1	N 和 Z
SUB	SUB f	f = f - WREG	1	1	C、DC、N、OV 和 Z
	SUB f,WREG	WREG = f - WREG	1	1	C、DC、N、OV 和 Z
	SUB #lit10,Wn	Wn = Wn - lit10	1	1	C、DC、N、OV 和 Z
	SUB Wb,Ws,Wd	Wd = Wb - Ws	1	1	C、DC、N、OV 和 Z
	SUB Wb,#lit5,Wd	Wd = Wb - lit5	1	1	C、DC、N、OV 和 Z
SUBB	SUBB f	f = f - WREG - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBB f,WREG	WREG = f - WREG - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBB #lit10,Wn	Wn = Wn - lit10 - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBB Wb,Ws,Wd	Wd = Wb - Ws - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBB Wb,#lit5,Wd	Wd = Wb - lit5 - (C̄)	1	1	C、DC、N、OV 和 Z
SUBR	SUBR f	f = WREG - f	1	1	C、DC、N、OV 和 Z
	SUBR f,WREG	WREG = WREG - f	1	1	C、DC、N、OV 和 Z
	SUBR Wb,Ws,Wd	Wd = Ws - Wb	1	1	C、DC、N、OV 和 Z
	SUBR Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C、DC、N、OV 和 Z
SUBBR	SUBBR f	f = WREG - f - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBBR f,WREG	WREG = WREG - f - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBBR Wb,Ws,Wd	Wd = Ws - Wb - (C̄)	1	1	C、DC、N、OV 和 Z
	SUBBR Wb,#lit5,Wd	Wd = lit5 - Wb - (C̄)	1	1	C、DC、N、OV 和 Z
SWAP	SWAP.b Wn	Wn = 将 Wn 的两个半字节相交换	1	1	无
	SWAP Wn	Wn = 将 Wn 的两个字节相交换	1	1	无
TBLRDH	TBLRDH Ws,Wd	将程序计数器的 <23:16> 读入 Wd<7:0>	1	2	无

表 31-2：指令集汇总（续）

汇编指令助记符	汇编语法	说明	字数	周期数	受影响的状态标志
TBLRDL	TBLRDL Ws,Wd	将程序计数器的 <15:0> 读入 Wd	1	2	无
TBLWTH	TBLWTH Ws,Wd	将 Ws<7:0> 写入程序计数器的 <23:16>	1	2	无
TBLWTL	TBLWTL Ws,Wd	将 Ws 写入程序计数器的 <15:0>	1	2	无
ULNK	ULNK	释放堆栈帧	1	1	无
XOR	XOR f	f = f .XOR.WREG	1	1	N 和 Z
	XOR f,WREG	WREG = f .XOR.WREG	1	1	N 和 Z
	XOR #lit10,Wn	Wd = lit10 .XOR.Wd	1	1	N 和 Z
	XOR Wb,Ws,Wd	Wd = Wb .XOR.Ws	1	1	N 和 Z
	XOR Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N 和 Z
ZE	ZE Ws,Wnd	Wnd = 对 Ws 进行零扩展	1	1	C、Z 和 N

# **PIC24FJ128GA310 系列**

---

---

注:

## 32.0 电气特性

本章提供 PIC24FJ128GA310 系列器件电气特性的概述。在文档后续版本中会添加其他信息。

下面列出了 PIC24FJ128GA310 系列器件的绝对最大值。器件长时间工作在最大值条件下，其稳定性会受到影响。我们建议不要使器件在该规范规定的参数范围外工作。

### 绝对极限参数值 (t)

偏置电压下的环境温度	.....	-40°C 至 +100°C
存储温度	.....	-65°C 至 +150°C
V <sub>DD</sub> 引脚相对于 V <sub>SS</sub> 的电压	.....	-0.3V 至 +4.0V
模拟数字组合引脚和 <u>MCLR</u> 引脚相对于 V <sub>SS</sub> 的电压	.....	-0.3V 至 (V <sub>DD</sub> + 0.3V)
当 V <sub>DD</sub> < 3.0V 时，只能用作数字功能的引脚相对于 V <sub>SS</sub> 的电压	.....	-0.3V 至 (V <sub>DD</sub> + 0.3V)
当 V <sub>DD</sub> > 3.0V 时，只能用作数字功能的引脚相对于 V <sub>SS</sub> 的电压	.....	-0.3V 至 (+5.5V)
V <sub>SS</sub> 引脚的最大输出电流	.....	300 mA
V <sub>DD</sub> 引脚的最大输入电流 (注 1)	.....	250 mA
任一 I/O 引脚的最大输出灌电流	.....	25 mA
任一 I/O 引脚的最大输出拉电流	.....	25 mA
所有端口的最大灌电流	.....	200 mA
所有端口的最大拉电流 (注 1)	.....	200 mA

**注 1:** 允许的最大电流由器件的最大功耗决定 (见表 32-1)。

† 注：如果器件工作参数超过上述各项最大额定值，可能对器件造成永久性损坏。上述值仅为运行条件极大值，我们建议不要使器件在该规范规定的范围以外运行。器件长时间工作在最大值条件下，其稳定性会受到影响。

# PIC24FJ128GA310 系列

## 32.1 直流特性

图 32-1： PIC24FJ128GA310 系列器件电压 — 频率关系图（工业级）

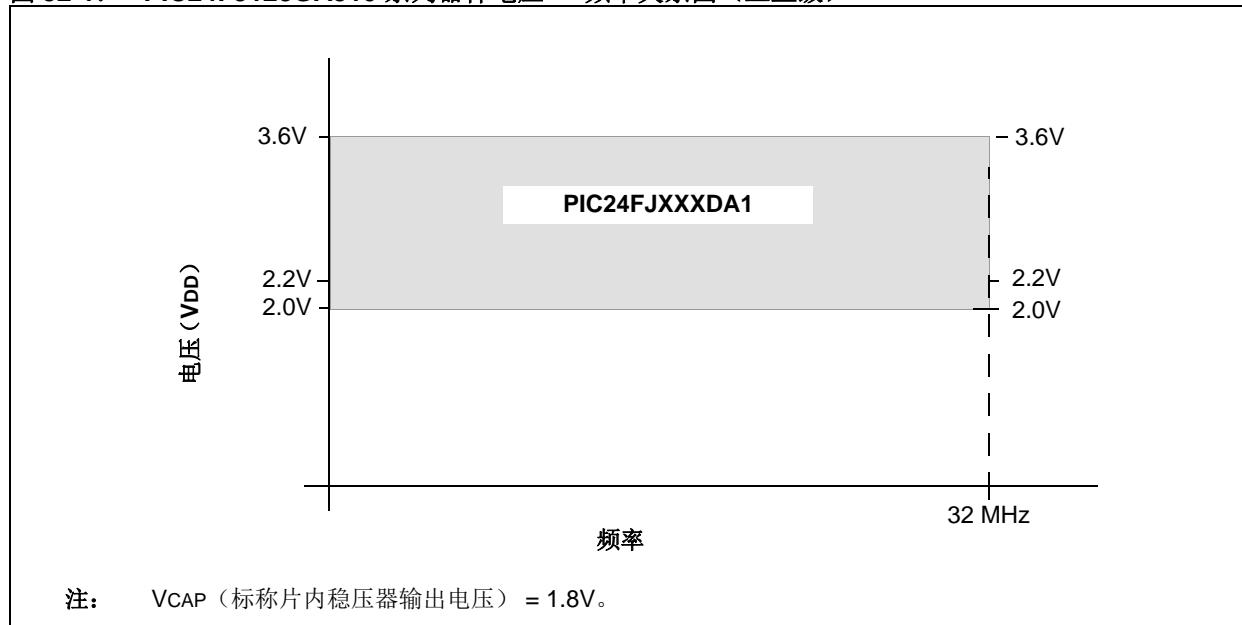


表 32-1：热工作条件

额定值	符号	最小值	典型值	最大值	单位
PIC24FJ128GA310 系列：					
工作结温范围	T <sub>J</sub>	-40	—	+125	°C
工作环境温度范围	T <sub>A</sub>	-40	—	+85	°C
功耗： 内部芯片功耗：P <sub>INT</sub> = V <sub>D</sub> x (I <sub>DD</sub> - Σ I <sub>OH</sub> ) I/O 引脚功耗： P <sub>I/O</sub> = Σ ({V <sub>D</sub> - V <sub>OH</sub> } x I <sub>OH</sub> ) + Σ (V <sub>OL</sub> x I <sub>OL</sub> )	P <sub>D</sub>	P <sub>INT</sub> + P <sub>I/O</sub>			W
允许的最大功耗	P <sub>DMAX</sub>	(T <sub>JMAX</sub> - T <sub>A</sub> )/θ <sub>JA</sub>			W

表 32-2：封装热阻特性

特性	符号	典型值	最大值	单位	注
封装热阻，14x14x1 mm 100 引脚 TQFP	θ <sub>JA</sub>	43.0	—	°C/W	(注 1)
封装热阻，12x12x1 mm 100 引脚 TQFP	θ <sub>JA</sub>	45.0	—	°C/W	(注 1)
封装热阻，12x12x1 mm 80 引脚 TQFP	θ <sub>JA</sub>	48.0	—	°C/W	(注 1)
封装热阻，10x10x1 mm 64 引脚 TQFP	θ <sub>JA</sub>	48.3	—	°C/W	(注 1)
封装热阻，9x9x0.9 mm 64 引脚 QFN	θ <sub>JA</sub>	28.0	—	°C/W	(注 1)
封装热阻，10x10x1.1 mm 121 引脚 BGA	θ <sub>JA</sub>	40.2	—	°C/W	(注 1)

注 1：通过封装模拟获得结点与环境的热阻值 Theta-JA (θ<sub>JA</sub>)。

表 32-3： 直流特性：温度和电压规范

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
<b>工作电压</b>							
DC10	VDD	电源电压	2	—	3.6	V	禁止 BOR
DC12	VDR	RAM 数据保持电压 <sup>(1)</sup>	1.9	—	—	V	
DC16	VPOR	确保内部上电复位信号的 <b>VDD 启动电压</b>	VSS	—	—	V	
DC17	SVDD	确保内部上电复位信号的 <b>VDD 上升速率</b>	0.05	—	—	V/ms	66 ms 内电压变化范围为 0-3.3V 50 ms 内电压变化范围为 0-2.5V
	VBOR	VDD 由高到低跳变时的欠压复位电压	2	—	2.2	V	

注 1：这是在不丢失 RAM 数据而片内稳压器输出电压开始跟随 VDD 的前提下，VDD 的下限值。

表 32-4： 直流特性：工作电流 (IDD)

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 <sup>(1)</sup>	最大值	单位	工作温度	VDD	条件
<b>工作电流 (IDD)</b>						
DC19	0.15	—	mA	-40°C 至 +85°C	2.0V	0.5 MIPS, FOSC = 1 MHz
DC20A	0.15	—	mA	-40°C 至 +85°C	3.3V	
DC20	0.31	—	mA	-40°C 至 +85°C	2.0V	1 MIPS, FOSC = 2 MHz
	0.32	—	mA	-40°C 至 +85°C	3.3V	
DC23	1.2	—	mA	-40°C 至 +85°C	2.0V	4 MIPS, FOSC = 8 MHz
	1.25	—	mA	-40°C 至 +85°C	3.3V	
DC24	4.8	6.8	mA	-40°C 至 +85°C	2.0V	16 MIPS, FOSC = 32 MHz
	4.9	6.9	mA	-40°C 至 +85°C	3.3V	
DC31	26	78	μA	-40°C 至 +85°C	2.0V	LPRC (15.5 KIPS) FOSC = 31 kHz
	26	80	μA	-40°C 至 +85°C	3.3V	

注 1：除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。典型参数仅作为设计参考，未经测试。

# PIC24FJ128GA310 系列

表 32-5： 直流特性：空闲电流（**I<sub>IDLE</sub>**）

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 <sup>(1)</sup>	最大值	单位	工作温度	V <sub>DD</sub>	条件
空闲电流（I <sub>IDLE</sub> ）						
DC40	81	—	μA	-40°C 至 +85°C	2.0V	1 MIPS, FOSC = 2 MHz
	86	—	μA	-40°C 至 +85°C	3.3V	
DC43	0.27	—	mA	-40°C 至 +85°C	2.0V	4 MIPS, FOSC = 8 MHz
	0.28	—	mA	-40°C 至 +85°C	3.3V	
DC47	1	1.35	mA	-40°C 至 +85°C	2.0V	16 MIPS, FOSC = 32 MHz
	1.07	1.4	mA	-40°C 至 +85°C	3.3V	
DC50	0.47	—	mA	-40°C 至 +85°C	2.0V	4 MIPS (FRC) , FOSC = 8 MHz
	0.48	—	mA	-40°C 至 +85°C	3.3V	
DC51	21	76	μA	-40°C 至 +85°C	2.0V	LPRC (15.5 KIPS) , FOSC = 31 kHz
	21	78	μA	-40°C 至 +85°C	3.3V	

注 1：除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

表 32-6： 直流特性：掉电电流 (IPD)

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	典型值 <sup>(1)</sup>	最大值	单位	工作温度	VDD	条件	
<b>掉电电流 (IPD)</b>							
DC60	—	—	μA	-40°C	2.0V	休眠 <sup>(2)</sup>	
	3.7	—	μA	+25°C			
	6.2	—	μA	+60°C			
	13.6	27.5	μA	+85°C			
	—	—	μA	-40°	3.3V		
	3.8	—	μA	+25°C			
	6.3	—	μA	+60°C			
	13.7	28	μA	+85°C			
DC61	—	—	μA	-40°	2.0V	低压休眠 <sup>(3)</sup>	
	0.33	—	μA	+25°C			
	2	—	μA	+60°C			
	7.7	14.5	μA	+85°C			
	—	—	μA	-40°	3.3V		
	0.34	—	μA	+25°C			
	2	—	μA	+60°C			
	7.9	15	μA	+85°C			
DC70	—	—	μA	-40°	2.0V	深度休眠	
	0.01	—	μA	+25°C			
	—	—	μA	+60°C			
	—	1.1	μA	+85°C			
	—	—	μA	-40°	3.3V		
	0.04	—	μA	+25°C			
	—	—	μA	+60°C			
	—	1.4	μA	+85°C			
	0.4	2.0	μA	-40°C 至 +85°C	0V	使用 VBAT 模式的 RTCC (LPRC/SOSC) <sup>(4)</sup>	

**注 1：** 除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

**2:** 禁止保持低压稳压器； RETEN (RCON<12>) = 0, LPCFG (CW1<10>) = 1。

**3:** 使能保持低压稳压器； RETEN (RCON<12>) = 1, LPCFG (CW1<10>) = 0。

**4:** VBAT 引脚连接电池且 RTCC 在 VDD = 0 的条件下运行。

# PIC24FJ128GA310 系列

表 32-7： 直流特性：Δ 电流（BOR、WDT、DSBOR、DSWDT 和 LCD）

直流特性			标准工作条件：2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)			
参数编号	典型值 <sup>(1)</sup>	最大值	单位	工作温度	V <sub>DD</sub>	条件
<b>增量电流欠压复位 (ΔBOR) <sup>(2)</sup></b>						
DC20	3.1	5	μA	-40°C 至 +85°C	2.0V	ΔBOR <sup>(2)</sup>
	4.3	6	μA	-40°C 至 +85°C	3.3V	
<b>增量电流欠压复位 (ΔWDT) <sup>(2)</sup></b>						
DC71	0.8	1.5	μA	-40°C 至 +85°C	2.0V	ΔWDT <sup>(2)</sup>
	0.8	1.5	μA	-40°C 至 +85°C	3.3V	
<b>增量电流 HLVD (ΔHLVD) <sup>(2)</sup></b>						
DC75	5.7	15	μA	-40°C 至 +85°C	2.0V	ΔHLVD <sup>(2)</sup>
	5.7	15	μA	-40°C 至 +85°C	3.3V	
<b>增量电流实时时钟和日历 (RTCC) <sup>(2)</sup></b>						
DC77	0.4	1	μA	-40°C 至 +85°C	2.0V	ΔRTCC <sup>(2)</sup> 使用 SOSC 的 RTCC
	0.4	1	μA	-40°C 至 +85°C	3.3V	
<b>增量电流实时时钟和日历 (RTCC) <sup>(2)</sup></b>						
DC77a	0.4	1	μA	-40°C 至 +85°C	2.0V	ΔRTCC <sup>(2)</sup> 使用 LPRC 的 RTCC
	0.4	1	μA	-40°C 至 +85°C	3.3V	
<b>增量电流深度休眠 BOR (Δ DSBOR) <sup>(2)</sup></b>						
DC81	0.07	0.3	μA	-40°C 至 +85°C	2.0V	Δ 深度休眠 BOR <sup>(2)</sup>
	0.07	0.3	μA	-40°C 至 +85°C	3.3V	
<b>增量电流深度休眠看门狗定时器复位 (Δ DSWDT) <sup>(2)</sup></b>						
DC80	0.27	0.4	μA	-40°C 至 +85°C	2.0V	Δ 深度休眠 WDT <sup>(2)</sup>
	0.27	0.4	μA	-40°C 至 +85°C	3.3V	
<b>增量电流 LCD (Δ LCD) <sup>(2)</sup></b>						
DC90	0.8	3	μA	-40°C 至 +85°C	3.3V	ΔLCD 外部 / 内部 <sup>(2,3)</sup> 1/8 复用 1/3 偏置
	20	30	μA	-40°C 至 +85°C	2.0V	
	24	40	μA	-40°C 至 +85°C	3.3V	ΔLCD 电荷泵 <sup>(2,4)</sup> 1/8 复用 1/3 偏置
<b>VBAT A/D 监视器<sup>(5)</sup></b>						
DC91	1.5	—	μA	-40°C 至 +85°C	3.3V	VBAT = 2V
	4	—	μA	-40°C 至 +85°C	3.3V	VBAT = 3.3V

**注 1：**除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

**2:** 模块使能和运行期间的增量电流。

**3:** 使能并运行 LCD；未连接显示屏；不包括梯形电阻网络电流。

**4:** 使能并运行 LCD；未连接显示屏。

**5:** A/D 通道内部连接到 VBAT 引脚，这是 A/D VBAT 运行期间的电流。

表 32-8： 直流特性：I/O 引脚输入规范

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
DI10 DI11 DI15 DI16 DI17 DI18 DI19	VIL	输入低电压 <sup>(3)</sup> 具有 ST 缓冲器的 I/O 引脚 具有 TTL 缓冲器的 I/O 引脚 <u>MCLR</u> OSC1 (XT 模式) OSC1 (HS 模式) 具有 I <sup>2</sup> C <sup>TM</sup> 缓冲器的 I/O 引脚 具有 SMBus 缓冲器的 I/O 引脚	Vss Vss Vss Vss Vss Vss Vss	— — — — — — —	0.2 VDD 0.15 VDD 0.2 VDD 0.2 VDD 0.2 VDD 0.3 VDD 0.8	V V V V V V V	使能 SMBus
		输入高电压 <sup>(3)</sup> 具有 ST 缓冲器的 I/O 引脚： 具有模拟功能， 仅有数字功能	0.8 VDD 0.8 VDD	— —	VDD 5.5	V V	
		具有 TTL 缓冲器的 I/O 引脚： 具有模拟功能， 仅有数字功能	0.25 VDD + 0.8 0.25 VDD + 0.8	— —	VDD 5.5	V V	
		<u>MCLR</u>	0.8 VDD	—	VDD	V	
		OSC1 (XT 模式)	0.7 VDD	—	VDD	V	
		OSC1 (HS 模式)	0.7 VDD	—	VDD	V	
		具有 I <sup>2</sup> C <sup>TM</sup> 缓冲器的 I/O 引脚： 具有模拟功能， 仅有数字功能	0.7 VDD 0.7 VDD	— —	VDD 5.5	V V	
DI29	VIH	具有 SMBus 缓冲器的 I/O 引脚： 具有模拟功能， 仅有数字功能	2.1 2.1	—	VDD 5.5	V V	2.5V ≤ VPIN ≤ VDD
DI30	ICNPU	<b>CNxx</b> 上拉电流	150	250	550	μA	VDD = 3.3V, VPIN = VSS
DI30A	ICNPD	<b>CNxx</b> 下拉电流	150	250	550	μA	VDD = 3.3V, VPIN = VDD
DI50 DI51 DI55 DI56	IIL	输入泄漏电流 <sup>(2)</sup> I/O 端口	— —	— —	±1 ±1	μA μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态 VSS ≤ VPIN ≤ 5.5, 引脚处于高阻态
		模拟输入引脚	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, 引脚处于高阻态
		<u>MCLR</u>	—	—	±1	μA	VSS ≤ VPIN ≤ VDD
		OSCI/CLKI	—	—	±1	μA	VSS ≤ VPIN ≤ VDD, EC、XT 和 HS 模式

**注 1：**除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

**2:** 负电流定义为引脚的拉电流。

**3:** I/O 引脚缓冲器类型请参见表 1-4。

# PIC24FJ128GA310 系列

表 32-9： 直流特性：I/O 引脚输出规范

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
DO10	VOL	输出低电压 I/O 端口	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2V
DO16		OSCO/CLKO	—	—	0.4	V	IOL = 6.6 mA, VDD = 3.6V
			—	—	0.4	V	IOL = 5.0 mA, VDD = 2V
DO20	VOH	输出高电压 I/O 端口	3.0	—	—	V	IOH = -3.0 mA, VDD = 3.6V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
DO26		OSCO/CLKO	1.65	—	—	V	IOH = -1.0 mA, VDD = 2V
			1.4	—	—	V	IOH = -3.0 mA, VDD = 2V
			2.4	—	—	V	IOH = -6.0 mA, VDD = 3.6V
			1.4	—	—	V	IOH = -1.0 mA, VDD = 2V

注 1：除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

表 32-10： 直流特性：程序存储器

直流特性			标准工作条件： 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
D130	EP	闪存程序存储器 电池耐用性	10000	—	—	E/W	-40°C 至 +85°C
D131	VPR	用于读操作的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D132B		自定时写操作使用的 VDD	VMIN	—	3.6	V	VMIN = 最小工作电压
D133A	TiW	自定时字写周期时间	—	20	—	μs	
		自定时行写周期时间	—	1.5	—	ms	
D133B	TiE	自定时页擦除时间	20	—	40	ms	
D134	TRETD	特性保持时间	20	—	—	年	假定未违反其他规范
D135	IDDP	编程期间的供电电流	—	16	—	mA	

注 1：除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。

表 32-11：内部稳压器规范

工作条件: $-40^{\circ}\text{C} < \text{TA} < +85^{\circ}\text{C}$ (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	VRGOUT	稳压器输出电压	—	1.8	—	V	
	VBG	内部带隙参考电压	1.14	1.2	1.26	V	
	C <sub>EFC</sub>	外部滤波电容值	4.7	10	—	μF	建议串联 $< 3$ 欧姆的电阻; 要求串联 $< 5$ 欧姆的电阻。
	T <sub>VREG</sub>		—	10	—	μs	VREGS = 1 (任何 POR 或 BOR)
	T <sub>BG</sub>	带隙参考启动时间	—	1	—	ms	
	V <sub>LVR</sub>	低压稳压器输出电压	—	1.2	—	V	RETEN = 1, LPCFG = 0

表 32-12：VBAT 工作电压规范

参数编号	符号	特性	最小值	典型值	最大值	单位	备注
	V <sub>BT</sub>	工作电压	1.8	—	3.6	V	电池连接到 V <sub>BAT</sub> 引脚
	V <sub>BTADC</sub>	V <sub>BAT</sub> A/D 监视电压规范 <sup>(1)</sup>	1.6	—	3.6	V	A/D 使用内部 A/D 通道监视 V <sub>BAT</sub> 引脚

注 1: A/D 值的测量 (使用 A/D) 由以下公式给出:

对于 10 位 A/D, 测量电压 =  $((V_{\text{BAT}}/2)/V_{\text{DD}}) * 1024$ ,

对于 12 位 A/D, 测量电压 =  $((V_{\text{BAT}}/2)V_{\text{DD}}) * 4096$ 。

表 32-13：CTMU 电流源规范

直流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 $-40^{\circ}\text{C} \leq \text{TA} \leq +85^{\circ}\text{C}$ (工业级)					2.5V < V <sub>DD</sub> < V <sub>DDMAX</sub>
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	备注	
	I <sub>OUT1</sub>	CTMU 电流源, 基本范围	—	550	—	nA	CTMUICON<1:0> = 00	
	I <sub>OUT2</sub>	CTMU 电流源, 10x 范围	—	5.5	—	μA	CTMUICON<1:0> = 01	
	I <sub>OUT3</sub>	CTMU 电流源, 100x 范围	—	55	—	μA	CTMUICON<1:0> = 10	
	I <sub>OUT4</sub>	CTMU 电流源, 1000x 范围	—	550	—	μA	CTMUICON<1:0> = 11 <sup>(2)</sup>	
	V <sub>Δ</sub>	每摄氏度电压 变化	—	3	—	mV/°C		

注 1: 电流微调范围的中点为标称值 (CTMUICON<7:2> = 000000)。

2: 不要对温度检测二极管使用此电流范围。

# PIC24FJ128GA310 系列

表 32-14: 高 / 低压检测特性

工作条件: $-40^{\circ}\text{C} < \text{TA} < +85^{\circ}\text{C}$ (除非另外声明)							
参数编号	符号	特性		最小值	典型值	最大值	单位
DC18	VHLVD	VDD 跳变时的 HLVD 电压	HLVLD<3:0> = 0100 <sup>(1)</sup>	3.45	—	3.75	V
			HLVLD<3:0> = 0101	3.30	—	3.6	V
			HLVLD<3:0> = 0110	3.00	—	3.3	V
			HLVLD<3:0> = 0111	2.80	—	3.1	V
			HLVLD<3:0> = 1000	2.70	—	2.95	V
			HLVLD<3:0> = 1001	2.50	—	2.75	V
			HLVLD<3:0> = 1010	2.40	—	2.60	V
			HLVLD<3:0> = 1011	2.30	—	2.5	V
			HLVLD<3:0> = 1100	2.20	—	2.4	V
			HLVLD<3:0> = 1101	2.10	—	2.3	V
			HLVLD<3:0> = 1110	2.00	—	2.2	V

注 1: 未实现 HLVLD<3:0> 值的跳变点 (从 0000 到 0011)。

表 32-15: 比较器直流规范

工作条件: $2.0\text{V} < \text{VDD} < 3.6\text{V}$ , $-40^{\circ}\text{C} < \text{TA} < +85^{\circ}\text{C}$ (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
D300	VIOFF	输入失调电压 *	—	20	40	mV	
D301	VICM	输入共模电压 *	0	—	VDD	V	
D302	CMRR	共模抑制比 *	55	—	—	dB	

\* 参数仅为特征值, 未经测试。

表 32-16: 比较器参考电压直流规范

工作条件: $2.0\text{V} < \text{VDD} < 3.6\text{V}$ , $-40^{\circ}\text{C} < \text{TA} < +85^{\circ}\text{C}$ (除非另外声明)							
参数编号	符号	特性	最小值	典型值	最大值	单位	备注
VRD310	CVRES	分辨率	VDD/24	—	VDD/32	LSb	
VRD311	CVRRA	绝对精度	—	—	AVDD - 1.5	LSb	
VRD312	CVRUR	单位电阻值 (R)	—	2K	—	$\Omega$	

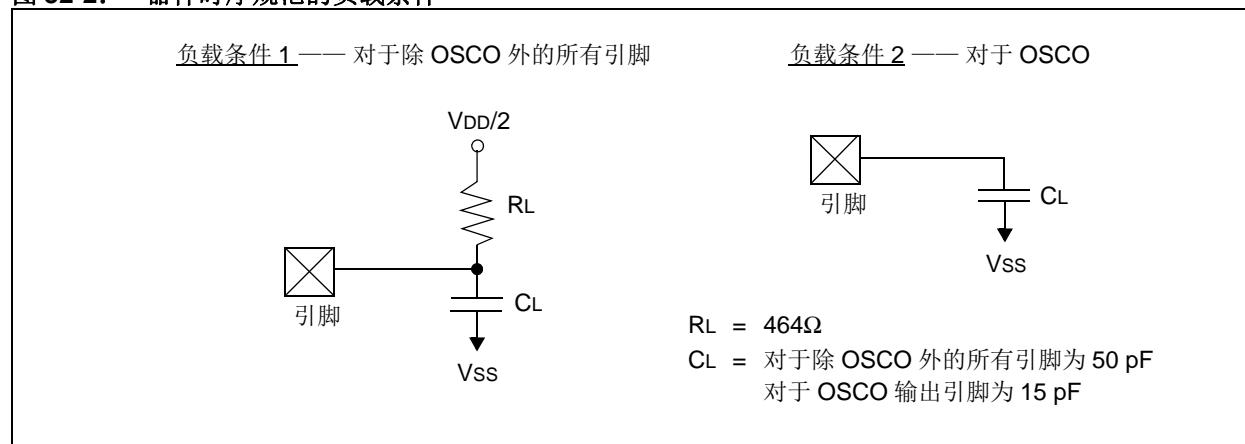
## 32.2 交流特性和时序参数

本节包含的信息说明了 PIC24FJ128GA310 系列器件的交流特性和时序参数。

**表 32-17：温度和电压规范——交流**

交流特性	标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级) 工作电压 VDD 范围如第 32.1 节“直流特性”所示。
------	--

**图 32-2：器件时序规范的负载条件**



**表 32-18：输出引脚上的容性负载要求**

参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
DO50	C <sub>OSCO</sub>	OSCO/CLKO 引脚	—	—	15	pF	当外部时钟用于驱动 OSCI 时，处于 XT 和 HS 模式。
DO56	C <sub>I/O</sub>	所有 I/O 引脚和 OSCO	—	—	50	pF	EC 模式
DO58	C <sub>B</sub>	SCL <sub>x</sub> 和 SDA <sub>x</sub>	—	—	400	pF	在 I <sup>2</sup> C <sup>TM</sup> 模式下

**注 1:** 除非另外声明，否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考，未经测试。

# PIC24FJ128GA310 系列

图 32-3：外部时钟时序

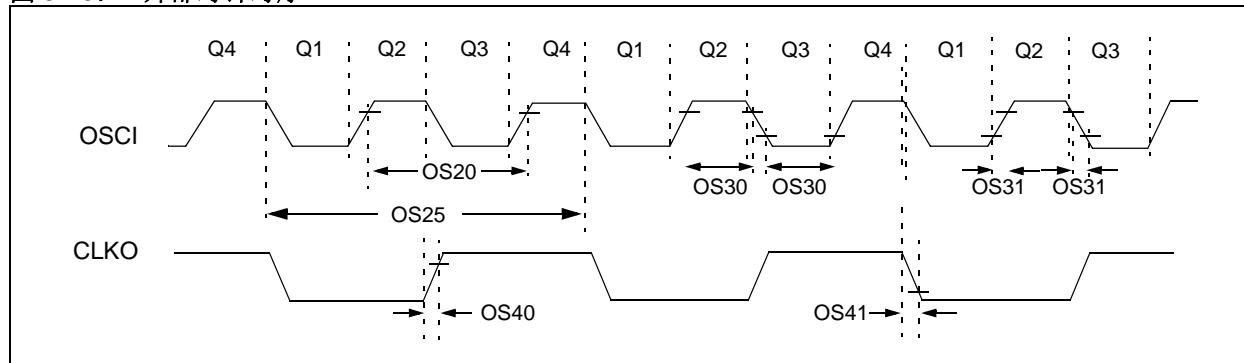


表 32-19：外部时钟时序要求

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
OS10	Fosc	外部 CLKI 频率 (仅在 EC 模式下允许使用外部时钟)	DC 4	—	32 8	MHz MHz	EC ECPLL
		振荡器频率	3.5 4 10 10 31	— — — — —	10 8 32 8 33	MHz MHz MHz MHz kHz	XT XTPLL HS HSPLL SOSC
OS20	Tosc	$Tosc = 1/Fosc$	—	—	—	—	关于 Fosc 的值, 见参数 OS10
OS25	Tcy	指令周期 <sup>(2)</sup>	62.5	—	DC	ns	
OS30	TosL, TosH	外部时钟输入 (OSCI) 的高电平或低电平时间	0.45 x Tosc	—	—	ns	EC
OS31	TosR, TosF	外部时钟输入 (OSCI) 的上升或下降时间	—	—	20	ns	EC
OS40	TckR	CLKO 上升时间 <sup>(3)</sup>	—	6	10	ns	
OS41	TckF	CLKO 下降时间 <sup>(3)</sup>	—	6	10	ns	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。这些参数仅供设计参考, 未经测试。

- 2: 指令周期 (Tcy) 等于输入振荡器基周期的 2 倍。所有规范值均基于器件在标准工作条件下执行代码时对应特定振荡器类型的特征数据。超过规范值可导致振荡器运行不稳定和 / 或使电流消耗超过预期值。所有器件在测试“最小值”时, 均在 OSCI/CLKI 引脚接入了外部时钟。当使用了外部时钟输入时, 所有器件的“最大”周期时间限制为“DC”(无时钟)。
- 3: 测量在 EC 模式下进行。CLKO 信号是在 OSCO 引脚上测得的。CLKO 在 Q1-Q2 周期 (1/2 Tcy) 中为低电平, 在 Q3-Q4 周期 (1/2 Tcy) 中为高电平。

表 32-20: PLL 时钟时序规范 ( $V_{DD} = 2.2V$  至  $3.6V$ )

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
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	条件
OS50	FPLL1	PLL 输入频率范围 <sup>(2)</sup>	4	—	8	MHz	ECPLL 模式
			4	—	8	MHz	HSPLL 模式
			4	—	8	MHz	XTPLL 模式
OS52	TLOCK	PLL 起振时间 (锁定时间)	—	—	128	μs	
OS53	DCLK	CLKO 稳定性 (去抖动性能)	-0.25	—	0.25	%	

注 1: 这些参数为特征值, 但未经生产测试。

2: 除非另外声明, 否则“典型值”栏中的数据都是在  $3.3V$ 、 $25^\circ C$  的条件下给出的。这些参数仅供设计参考, 未经测试。

表 32-21: 内部 RC 振荡器精度

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	特性	最小值	典型值	最大值	单位	条件	
F20	FRC 精度 @ 8 MHz <sup>(1,2)</sup>	-1	—	1	%	$-10^\circ C \leq TA \leq +85^\circ C$	$2V \leq V_{DD} \leq 3.6V$
		-1.5	—	1.5	%	$-40^\circ C \leq TA \leq -10^\circ C$	$2V \leq V_{DD} \leq 3.6V$
F21	LPRC @ 31 kHz	-20	—	20	%	$-40^\circ C \leq TA \leq +85^\circ C$	V <sub>CAP</sub> (片内稳压器输出电压) = 1.8V。

注 1: 已在  $25^\circ C$ 、 $3.3V$  条件下对频率进行了校准。OSCTUN 位可用来补偿温度漂移。

2: 要达到这一精度, 单片机封装所受到的物理应力 (例如: 弯曲 PCB) 必须保持最小。

表 32-22: RC 振荡器起振时间

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	特性	最小值	典型值	最大值	单位	条件	
	TFRC	—	15	—	μs		
	TLCRC	—	50	—	μs		

# PIC24FJ128GA310 系列

图 32-4: CLK0 和 I/O 时序特性

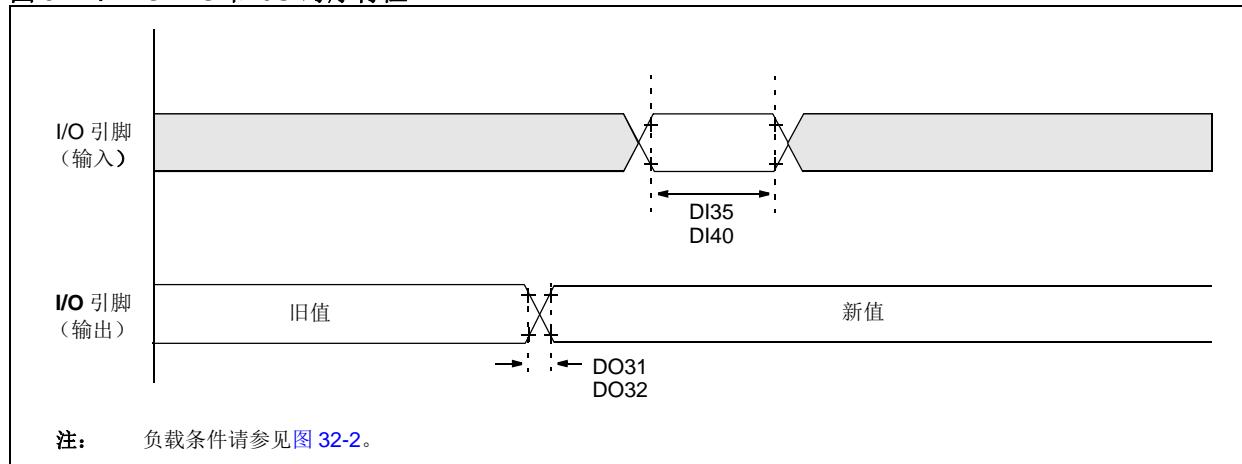


表 32-23: CLK0 和 I/O 时序要求

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值 <sup>(1)</sup>	最大值	单位	条件
DO31	T <sub>IoR</sub>	端口输出上升时间	—	10	25	ns	
DO32	T <sub>IoF</sub>	端口输出下降时间	—	10	25	ns	
DI35	T <sub>InP</sub>	INTx 引脚高电平或低电平时间 (输入)	20	—	—	ns	
DI40	T <sub>RBP</sub>	CNx 高电平或低电平时间 (输入)	2	—	—	T <sub>CY</sub>	

注 1: 除非另外声明, 否则“典型值”栏中的数据都是在 3.3V、25°C 的条件下给出的。

表 32-24：复位和欠压复位要求

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C (工业级)				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
SY10	T <sub>MCL</sub>	MCLR 脉宽 (低电平)	2	—	—	μs	
SY12	T <sub>POR</sub>	上电复位延时	—	2	—	μs	
SY13	T <sub>IOZ</sub>	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	—	—	100	ns	
SY25	T <sub>BOR</sub>	欠压复位脉冲宽度	1	—	—	μs	V <sub>D</sub> D ≤ V <sub>B</sub> OR
	T <sub>RST</sub>	内部状态复位时间	—	50	—	μs	
SY71	T <sub>PM</sub>	程序存储器唤醒时间	—	20	—	μs	V <sub>REGS</sub> = 0 时的休眠唤醒
			—	1	—	μs	V <sub>REGS</sub> = 1 时的休眠唤醒
SY72	T <sub>LVR</sub>	低压稳压器唤醒时间	—	90	—	μs	V <sub>REGS</sub> = 0 时的休眠唤醒
			—	70	—	μs	V <sub>REGS</sub> = 1 时的休眠唤醒
	T <sub>D</sub> SWU	深度休眠唤醒时间	—	200	—	μs	V <sub>C</sub> AP 在唤醒前完全放电

# PIC24FJ128GA310 系列

表 32-25: A/D 模块规范

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
器件电源							
AD01	AVDD	模块电源 VDD	取 VDD – 0.3 或 2.2 中的较大值	—	取 VDD + 0.3 或 3.6 中的较小值	V	
AD02	AVss	模块电源 Vss	Vss – 0.3	—	Vss + 0.3	V	
参考输入							
AD05	VREFH	参考电压高电压	AVss + 1.7	—	AVDD	V	
AD06	VREFL	参考电压低电压	AVss	—	AVDD – 1.7	V	
AD07	VREF	绝对参考电压	AVss – 0.3	—	AVDD + 0.3	V	
模拟输入							
AD10	VINH-VINL	满量程输入范围	VREFL	—	VREFH	V	(注 2)
AD11	VIN	绝对输入电压	AVss – 0.3	—	AVDD + 0.3	V	
AD12	VINL	绝对 VINL 输入电压	AVss – 0.3	—	AVDD/3	V	
AD13		泄漏电流	—	±1.0	±610	nA	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V, 信号源阻抗 = 2.5 kΩ
AD17	RIN	模拟信号源推荐阻抗	—	—	2.5K	Ω	10 位
A/D 精度							
AD20B	Nr	分辨率	—	12	—	位	
AD21B	INL	积分非线性误差	—	±1	<±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD22B	DNL	微分非线性误差	—	—	<±1	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD23B	GERR	增益误差	—	±1	±3	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD24B	EOFF	失调误差	—	±1	±2	LSb	VINL = AVSS = VREFL = 0V, AVDD = VREFH = 3V
AD25B		单调性 (1)	—	—	—	—	保证

注 1: A/D 转换结果不会因输入电压的增加而减小，并且不会丢失编码。

2: 测量是在使用外部 VREF+ 和 VREF- 作为 A/D 参考电压的情况下进行的。

表 32-26: A/D 转换时序要求 (1)

交流特性			标准工作条件: 2V 至 3.6V (除非另外声明) 工作温度 -40°C ≤ TA ≤ +85°C				
参数编号	符号	特性	最小值	典型值	最大值	单位	条件
时钟参数							
AD50	TAD	A/D 时钟周期	75	—	—	ns	T <sub>CY</sub> = 75 ns, AD1CON3 处于默认状态
AD51	t <sub>RC</sub>	A/D 内部 RC 振荡器周期	—	250	—	ns	
转换速率							
AD55	t <sub>CONV</sub>	转换时间	—	14	—	T <sub>AD</sub>	
AD56	F <sub>CNV</sub>	吞吐率	—	—	200	ksp/s	A <sub>VDD</sub> > 2.7V
AD57	t <sub>SAMP</sub>	采样时间	—	1	—	T <sub>AD</sub>	
时钟参数							
AD61	t <sub>PSS</sub>	从置 1 采样位 (SAMP) 到采样启动的延时	2	—	3	T <sub>AD</sub>	

注 1: 因为采样电容最终将释放电荷, 因此低于 10 kHz 的时钟频率可能会影响线性性能, 尤其是在温度较高时。

# **PIC24FJ128GA310 系列**

---

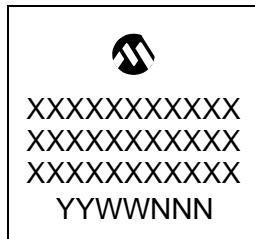
---

注:

## 33.0 封装信息

### 33.1 封装标识信息

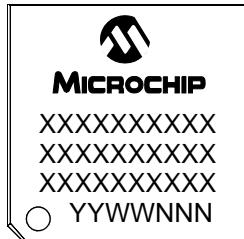
64 引脚 QFN (9x9x0.9 mm)



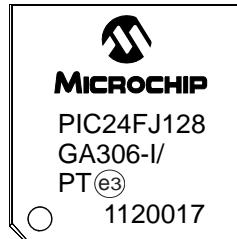
示例



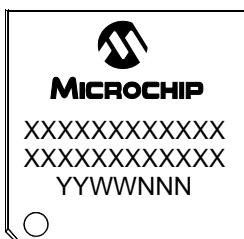
64 引脚 TQFP (10x10x1 mm)



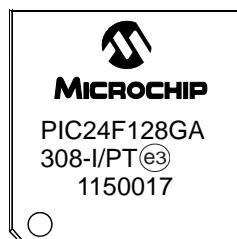
示例



80 引脚 TQFP (12x12x1 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 以字母数字排序的追踪代码

(e3) 雾锡（Matte Tin, Sn）的 JEDEC 无铅标志

\* 表示无铅封装。JEDEC 无铅标志（e3）标示于此种封装的外包装上。

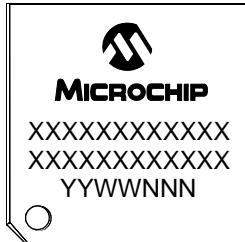
#### 注:

Microchip 元器件编号如果无法在同一行内完整标注，将换行标出，因此会限制表示客户信息的字符数。

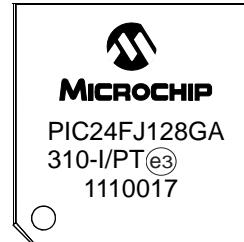
# PIC24FJ128GA310 系列

## 33.2 封装标识信息

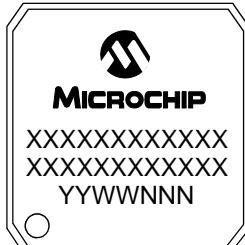
100 引脚 TQFP (12x12x1 mm)



示例



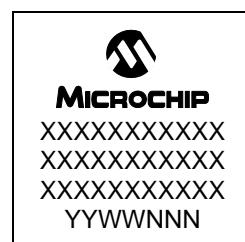
100 引脚 TQFP (14x14x1 mm)



示例



121-BGA (10x10x1.1 mm)



示例

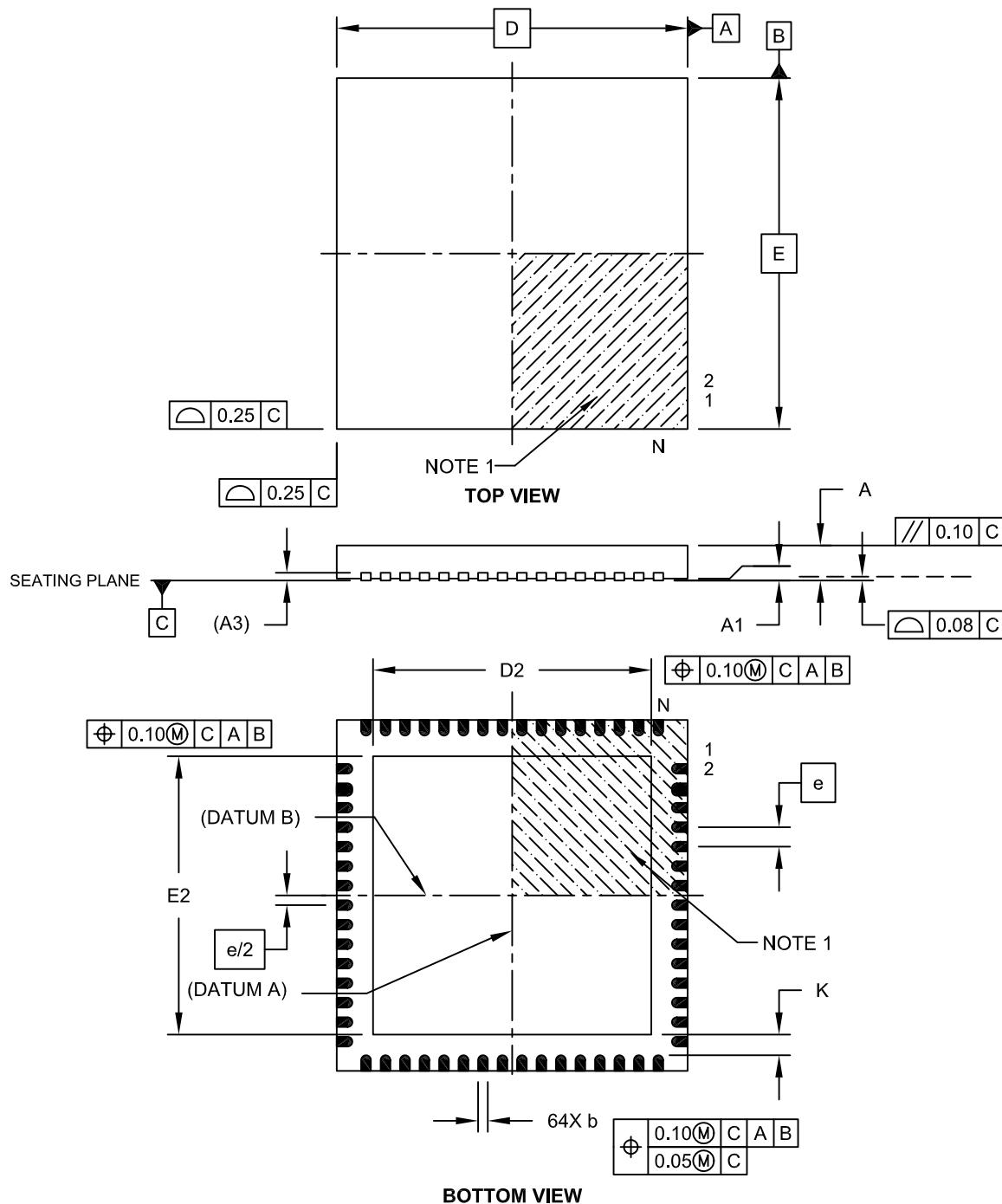


### 33.3 封装详细信息

以下部分将介绍各种封装的技术细节。

#### 64 引脚塑封四方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN]

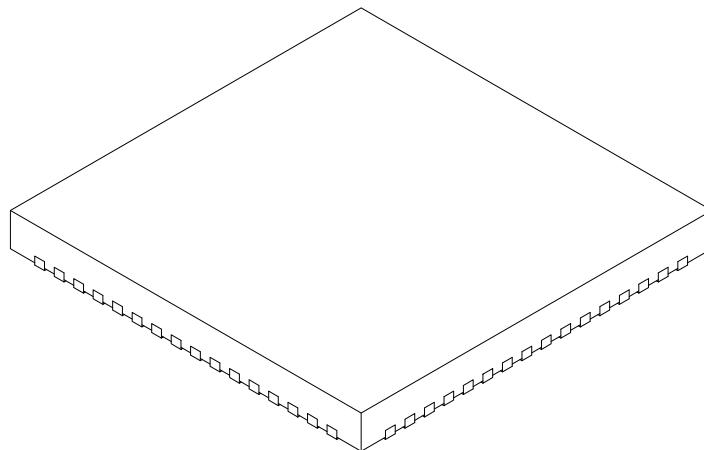
**注：** 最新封装图请至 <http://www.microchip.com/packaging> 查看封装规范。



# PIC24FJ128GA310 系列

## 64 引脚塑封四方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N	64		
Pitch	e	0.50	0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3	0.20	REF	
Overall Width	E	9.00	BSC	
Exposed Pad Width	E2	7.05	7.15	7.50
Overall Length	D	9.00	BSC	
Exposed Pad Length	D2	7.05	7.15	7.50
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.20	-	-

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Package is saw singulated.

3. Dimensioning and tolerancing per ASME Y14.5M.

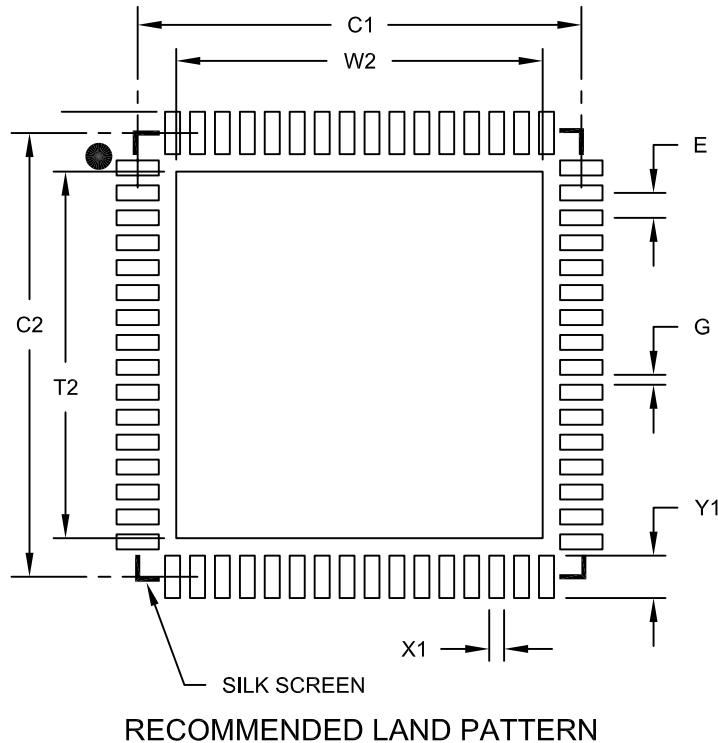
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-149B Sheet 2 of 2

## 64 引脚塑封四方扁平无脚封装 (MR) —— 主体 9x9x0.9 mm [QFN], 触点长度 0.40 mm

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



Dimension Limits	Units	MILLIMETERS		
		MIN	NOM	MAX
Contact Pitch	E		0.50 BSC	
Optional Center Pad Width	W2			7.35
Optional Center Pad Length	T2			7.35
Contact Pad Spacing	C1		8.90	
Contact Pad Spacing	C2		8.90	
Contact Pad Width (X64)	X1			0.30
Contact Pad Length (X64)	Y1			0.85
Distance Between Pads	G	0.20		

### Notes:

- Dimensioning and tolerancing per ASME Y14.5M

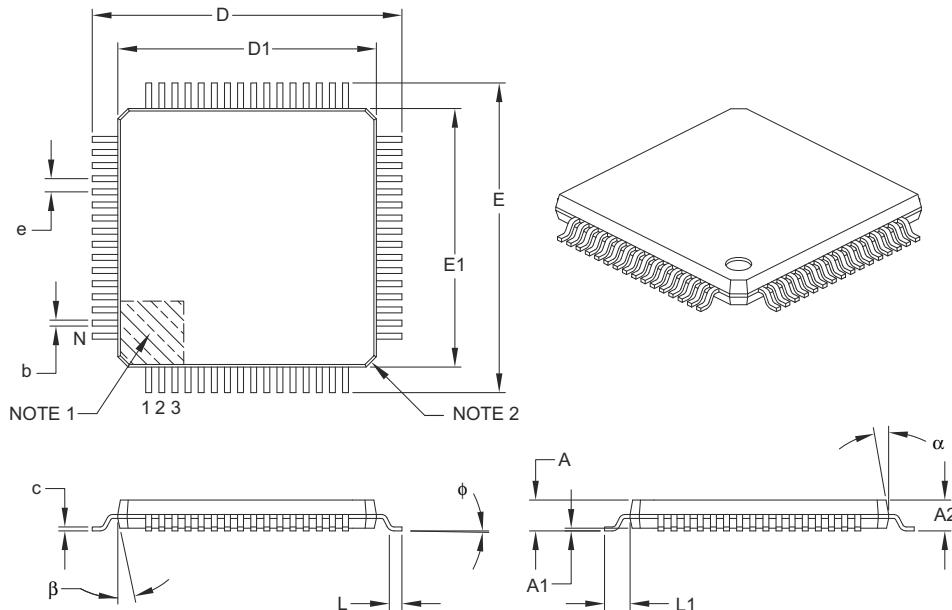
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2149A

# PIC24FJ128GA310 系列

## 64 引脚塑封薄型四方扁平封装 (PT) —— 主体 10x10x1 mm, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads		64		
Lead Pitch		0.50 BSC		
Overall Height		A	—	1.20
Molded Package Thickness		A2	0.95	1.00
Standoff		A1	0.05	—
Foot Length		L	0.45	0.60
Footprint		L1	1.00 REF	
Foot Angle		phi	0°	3.5°
Overall Width		E	12.00 BSC	
Overall Length		D	12.00 BSC	
Molded Package Width		E1	10.00 BSC	
Molded Package Length		D1	10.00 BSC	
Lead Thickness		c	0.09	—
Lead Width		b	0.17	0.22
Mold Draft Angle Top		alpha	11°	12°
Mold Draft Angle Bottom		beta	11°	12°
		gamma	12°	13°

### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

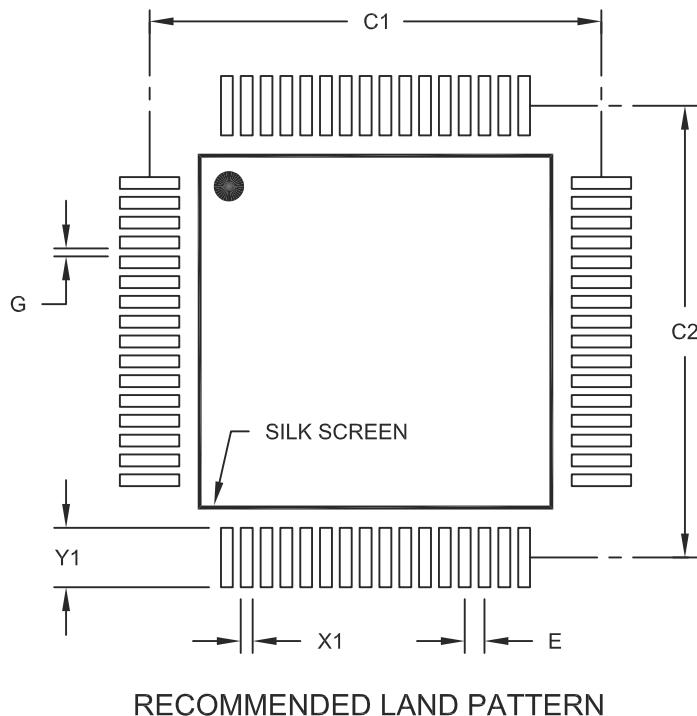
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-085B

## 64 引脚塑封薄型四方扁平封装 (PT) —— 主体 10x10x1 mm, 2.00 mm [TQFP]

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



		Units	MILLIMETERS		
Dimension Limits			MIN	NOM	MAX
Contact Pitch	E		0.50	BSC	
Contact Pad Spacing	C1		11.40		
Contact Pad Spacing	C2		11.40		
Contact Pad Width (X64)	X1			0.30	
Contact Pad Length (X64)	Y1				1.50
Distance Between Pads	G	0.20			

Notes:

- Dimensioning and tolerancing per ASME Y14.5M

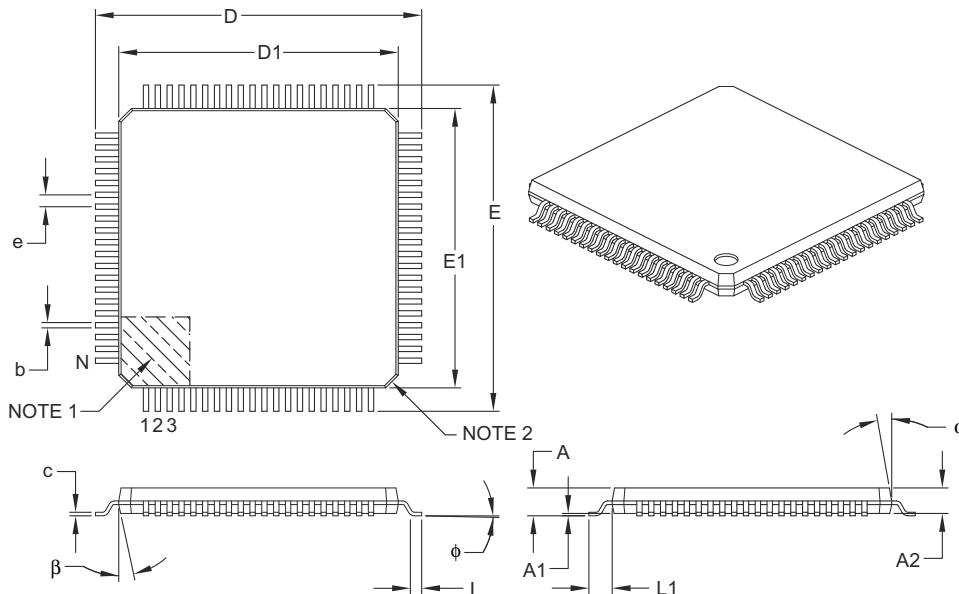
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2085A

# PIC24FJ128GA310 系列

## 80 引脚塑封薄型四方扁平封装 (PT) —— 主体 12x12x1 mm, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads		80		
Lead Pitch		0.50 BSC		
Overall Height		A	—	1.20
Molded Package Thickness		A2	0.95	1.00
Standoff		A1	0.05	—
Foot Length		L	0.45	0.60
Footprint		L1	1.00 REF	
Foot Angle		φ	0°	3.5°
Overall Width		E	14.00 BSC	
Overall Length		D	14.00 BSC	
Molded Package Width		E1	12.00 BSC	
Molded Package Length		D1	12.00 BSC	
Lead Thickness		c	0.09	—
Lead Width		b	0.17	0.22
Mold Draft Angle Top		α	11°	12°
Mold Draft Angle Bottom		β	11°	12°

### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

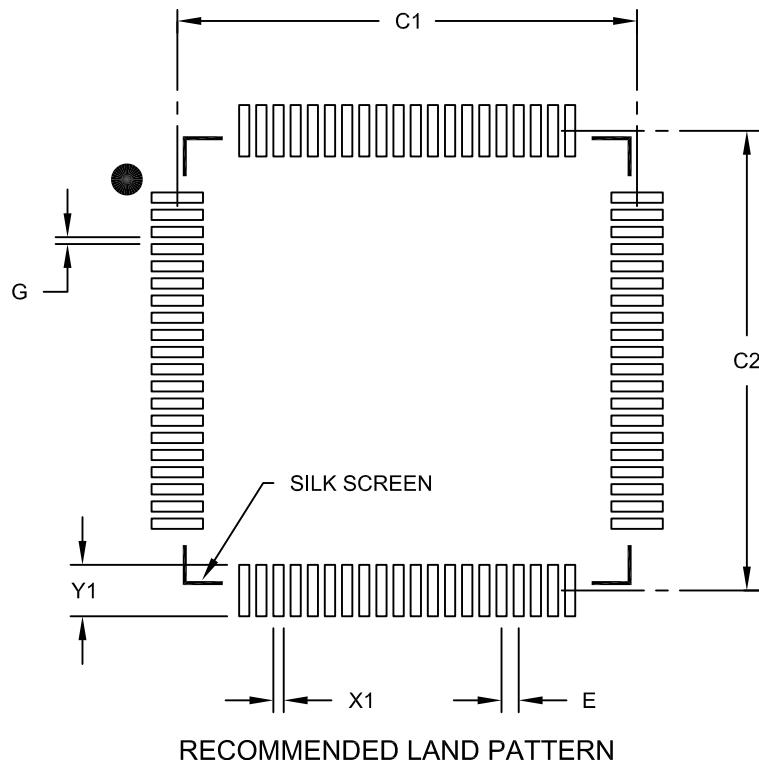
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-092B

## 80 引脚塑封薄型四方扁平封装 (PT) —— 主体 12x12x1 mm, 2.00 mm 引脚投影 长度 [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch		0.50 BSC		
Contact Pad Spacing	C1		13.40	
Contact Pad Spacing	C2		13.40	
Contact Pad Width (X80)	X1			0.30
Contact Pad Length (X80)	Y1			1.50
Distance Between Pads	G	0.20		

Notes:

- Dimensioning and tolerancing per ASME Y14.5M

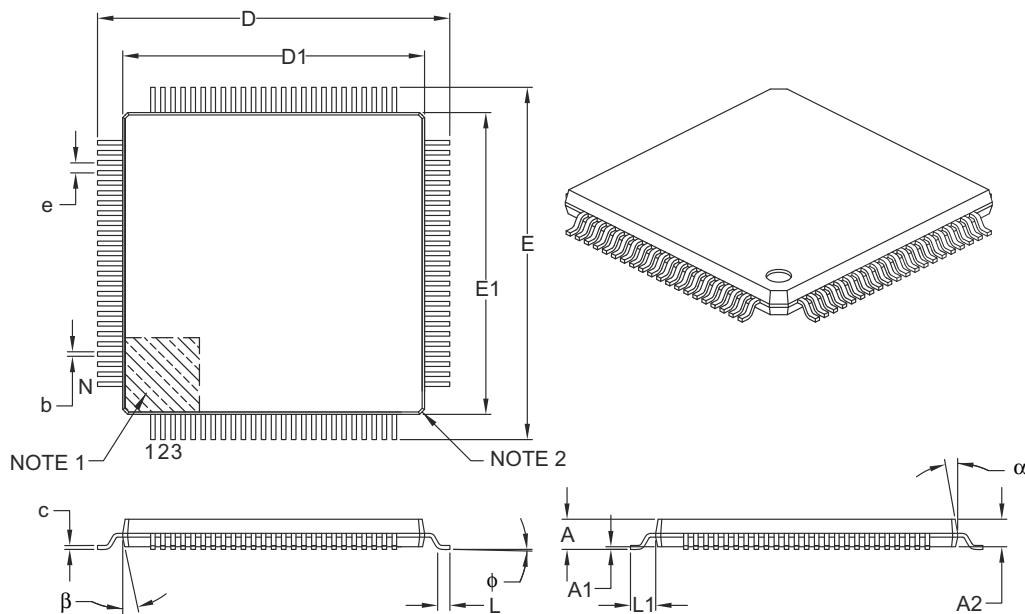
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2092B

# PIC24FJ128GA310 系列

100 引脚塑封薄型四方扁平封装 (PT) —— 主体 12x12x1 mm, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads		N		
Lead Pitch		e		
Overall Height		A		
Molded Package Thickness		A2		
Standoff		A1		
Foot Length		L		
Footprint		L1		
Foot Angle		ϕ		
Overall Width		E		
Overall Length		D		
Molded Package Width		E1		
Molded Package Length		D1		
Lead Thickness		c		
Lead Width		b		
Mold Draft Angle Top		α		
Mold Draft Angle Bottom		β		

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

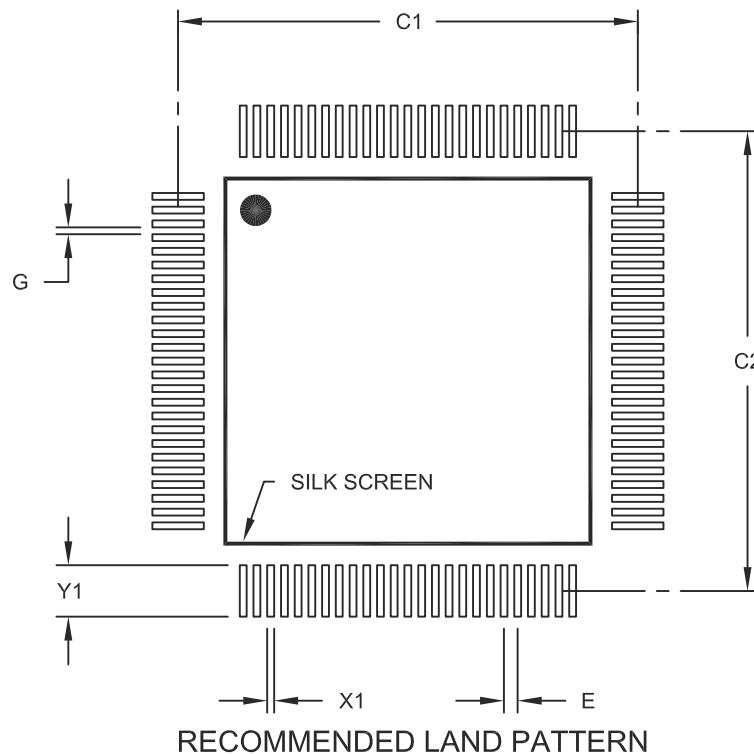
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-100B

## 100 引脚塑封薄型四方扁平封装 (PT) —— 主体 12x12x1 mm, 2.00 mm [TQFP]

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



		Units	MILLIMETERS		
Dimension		Limits	MIN	NOM	MAX
Contact Pitch	E		0.40	BSC	
Contact Pad Spacing	C1			13.40	
Contact Pad Spacing	C2			13.40	
Contact Pad Width (X100)	X1				0.20
Contact Pad Length (X100)	Y1				1.50
Distance Between Pads	G	0.20			

Notes:

- Dimensioning and tolerancing per ASME Y14.5M

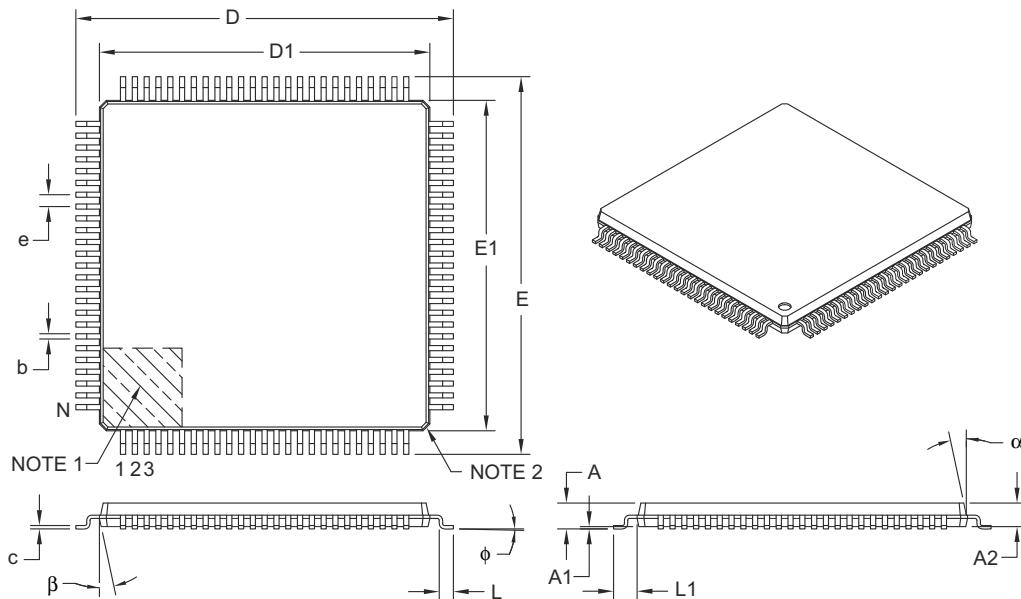
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2100A

# PIC24FJ128GA310 系列

100 引脚塑封薄型四方扁平封装 (PT) —— 主体 14x14x1 mm, 2.00 mm [TQFP]

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



Dimension Limits		MILLIMETERS		
		MIN	NOM	MAX
Number of Leads	N		100	
Lead Pitch	e		0.50 BSC	
Overall Height	A	—	—	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	—	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	ϕ	0°	3.5°	7°
Overall Width	E	16.00 BSC		
Overall Length	D	16.00 BSC		
Molded Package Width	E1	14.00 BSC		
Molded Package Length	D1	14.00 BSC		
Lead Thickness	c	0.09	—	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

**Notes:**

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M.

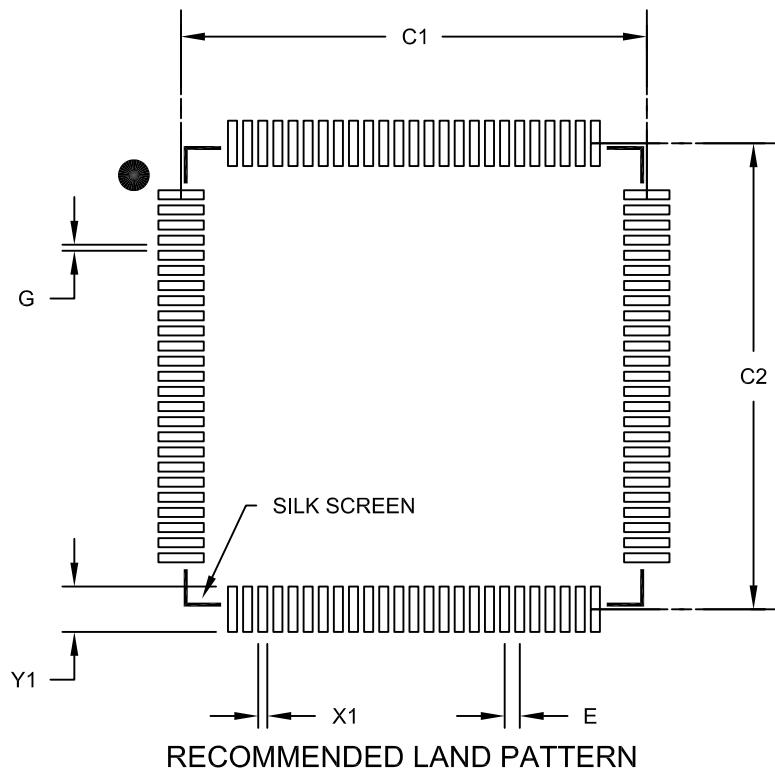
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-110B

## 100 引脚塑封薄型四方扁平封装 (PT) —— 主体 14x14x1 mm, 2.00 mm [TQFP]

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



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Contact Pitch	E		0.50	BSC
Contact Pad Spacing	C1		15.40	
Contact Pad Spacing	C2		15.40	
Contact Pad Width (X100)	X1			0.30
Contact Pad Length (X100)	Y1			1.50
Distance Between Pads	G	0.20		

### Notes:

- Dimensioning and tolerancing per ASME Y14.5M

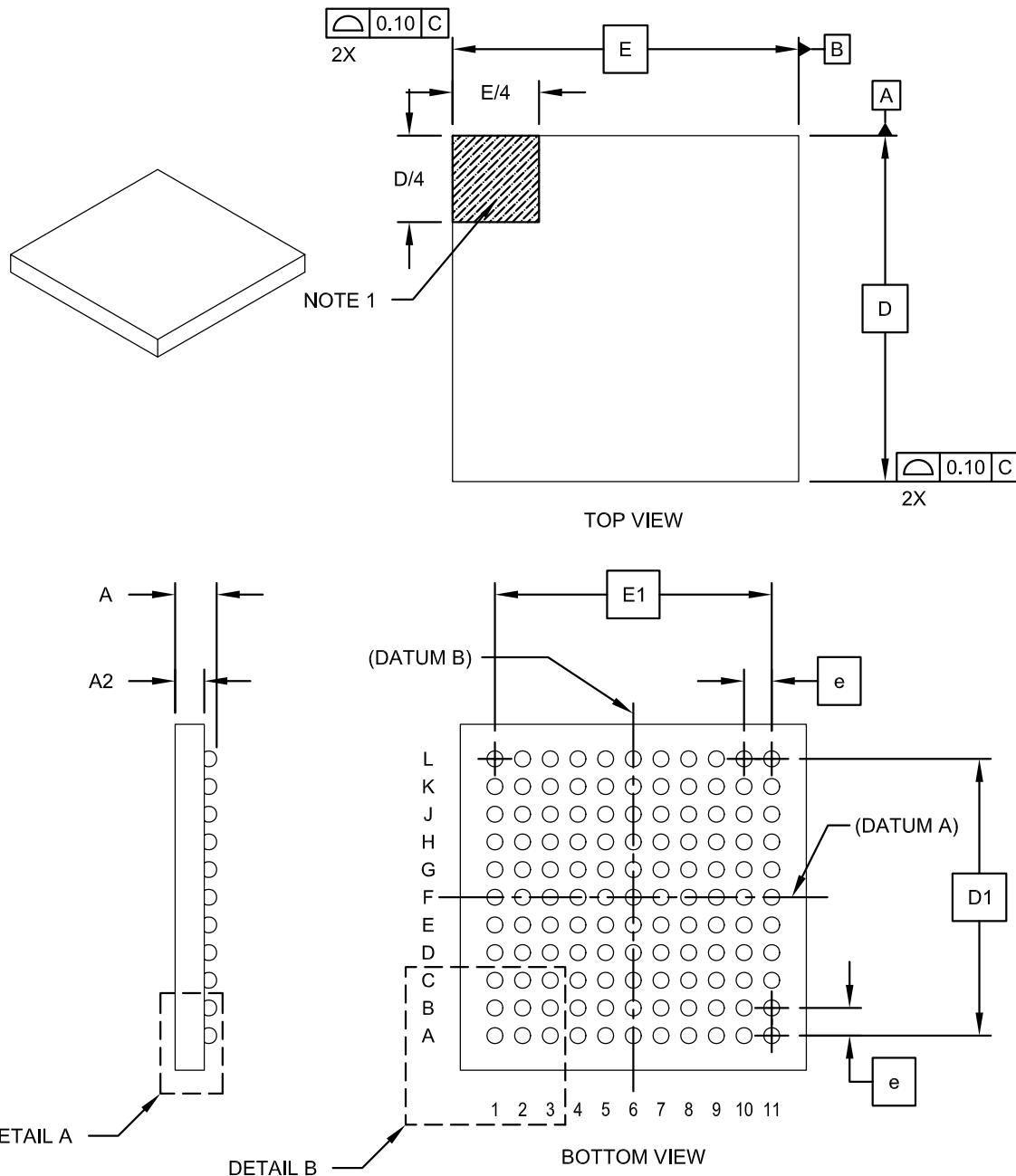
BSC: Basic Dimension. Theoretically exact value shown without tolerances.

Microchip Technology Drawing No. C04-2110B

# PIC24FJ128GA310 系列

## 121 引脚塑封薄型焊球阵列 (BG) —— 主体 10x10x1.10 mm [XBGA]

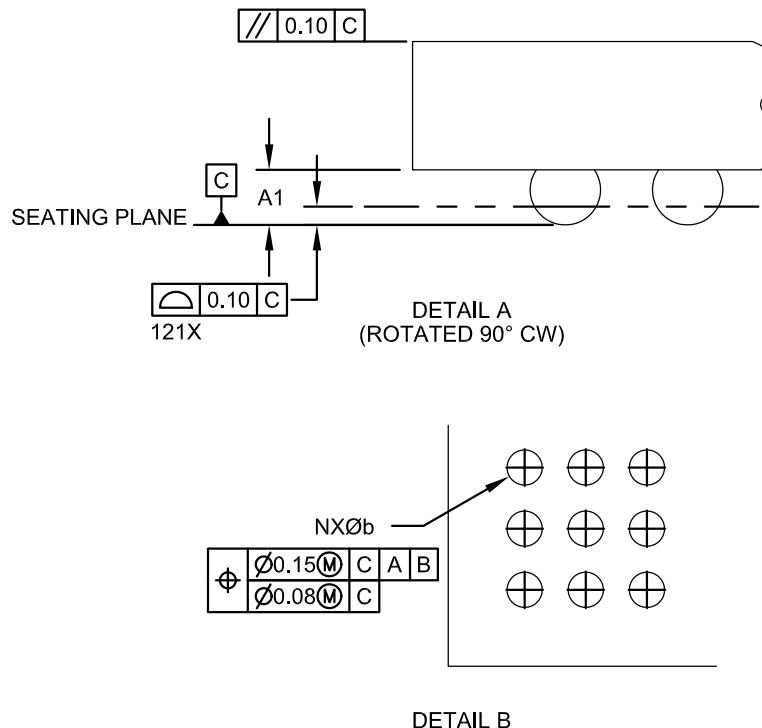
注： 最新封装图请至 <http://www.microchip.com/packaging> 查看封装规范。



Microchip Technology Drawing C04-148A Sheet 1 of 2

## 121 引脚塑封薄型焊球阵列 (BG) —— 主体 10x10x1.10 mm [XBGA]

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



	Units	MILLIMETERS		
Dimension	Limits	MIN	NOM	MAX
Number of Contacts	N		121	
Contact Pitch	e		0.80 BSC	
Overall Height	A	1.00	1.10	1.20
Standoff	A1	0.25	0.30	0.35
Molded Package Thickness	A2	0.55	0.60	0.65
Overall Width	E	10.00 BSC		
Array Width	E1	8.00 BSC		
Overall Length	D	10.00 BSC		
Array Length	D1	8.00 BSC		
Contact Diameter	b	0.40 TYP		

### Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. 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.

3. The outer rows and columns of balls are located with respect to datums A and B.

# **PIC24FJ128GA310 系列**

---

---

注:

## 附录 A: 版本历史

### 版本 A (2010 年 3 月)

PIC24FJ128GA310 系列器件的初始数据手册。

### 版本 B (2011 年 5 月)

更改了表 4-12 中 TRISA 的复位值。编辑了“[单片机的特性：](#)”

### 版本 C (2011 年 7 月)

更新了[第 32.0 节 “电气特性”](#)中的值。更改了特殊功能寄存器地址。删除了 OCTRIG1 和 OCTRIG2 引脚。对整篇文档的文字进行了少量编辑。

### 版本 D (2011 年 8 月)

更新了 VBAT 规范；更新了[第 32.0 节 “电气特性”](#)的最大值。

### 版本 E (2011 年 10 月)

- 删除了 CW4<9> 寄存器中的 RTCBAT 位。
- 在[第 32.0 节 “电气特性”](#)中增加了 IDD/IPD 数字。
- 增加了 VBAT 引脚电容的详细信息。
- 增加了[第 24.3 节 “使用 Vbat 的 A/D 操作”](#)。

### 版本 F (2011 年 11 月)

更新了[第 32.0 节 “电气特性”](#)中的值。对整篇文档的文字进行了少量编辑。

# **PIC24FJ128GA310 系列**

---

---

注:

## 索引

### 数字

32 位可编程循环冗余校验 (CRC) 发生器 ..... 289

### A

#### A/D

工作原理 ..... 295  
寄存器 ..... 298  
扩展 DMA 操作 ..... 297

### B

版本历史 ..... 393  
备用中断向量表 (AVIT) ..... 95  
比较器参考电压 ..... 321  
    配置 ..... 321  
变更通知客户服务 ..... 401

### C

#### C 编译器

MPLAB C18 ..... 348

#### CPU

编程模型 ..... 35  
控制寄存器 ..... 38  
内核寄存器 ..... 36  
算术逻辑单元 (ALU) ..... 40

#### CRC

16 和 32 位多项式设置示例 ..... 290  
多项式 ..... 290  
用户接口 ..... 290

#### CTMU

测量电容 ..... 323  
测量时间 ..... 324  
脉冲生成和延时 ..... 324

#### 产品标识体系

程序存储空间与数据存储空间的接口 ..... 69

#### 程序存储器

存储器硬编码向量 ..... 42  
存储器映射 ..... 41  
地址构成 ..... 69  
地址空间 ..... 41  
构成 ..... 42  
闪存配置字 ..... 42  
使用表指令访问 ..... 71  
    使用 EDS 从程序存储器读数据 ..... 72

#### 程序校验

345

充电时间测量单元。见 CTMUM。

串行外设接口。见 SPI。

串行外设接口 (SPI) ..... 221

存储器构成 ..... 41

### D

#### DMA

操作汇总 ..... 76  
典型设置 ..... 78  
通道触发源 ..... 82  
外设模块禁止 (PMD) ..... 78  
传输类型 ..... 77

#### DMA 控制器

12

#### 代码保护

345

代码段保护 ..... 345

    配置选项 ..... 345

配置保护 ..... 346

通用段保护 ..... 345

### 代码示例

编程闪存存储器单字 ..... 88  
编程闪存存储器的一个单字 (C 语言) ..... 88  
擦除程序存储器块 (C 语言) ..... 87  
擦除程序存储器块 (汇编) ..... 86  
从程序存储器进行 EDS 读操作的代码 (汇编语言) ..... 72  
端口读 / 写 (C 语言) ..... 172  
端口读 / 写 (汇编语言) ..... 172  
EDS 读 (汇编语言) ..... 66  
EDS 写 (汇编语言) ..... 67  
将 RTCWREN 位置 1 ..... 276  
PWRSAV 指令语法 ..... 156  
配置 UART1 输入 / 输出功能 (PPS) ..... 177  
启动编程序列 ..... 87  
时钟切换的基本序列 ..... 151  
装载写缓冲器 ..... 87  
带阈值扫描功能的 12 位 A/D 转换器 ..... 295

### 电气特性

CLKO 和 I/O 时序 ..... 372  
复位和欠压复位要求 ..... 373  
高 / 低压检测 ..... 368  
绝对最大额定值 ..... 359  
内部稳压器规范 ..... 367  
PLL 时钟时序规范 ..... 371  
RC 振荡器起振时间 ..... 371  
热条件 ..... 360  
输出引脚上的容性负载要求 ..... 369  
V/F 图 ..... 360  
外部时钟时序 ..... 370  
调制器。请参见数据信号调制器 ..... 249  
读者反馈表 ..... 402  
端口或引脚可承受的输入电压说明 ..... 168

### E

#### EPMP

封装类型 ..... 253  
主要特性 ..... 253

### F

封装 ..... 377  
标识 ..... 377  
详细信息 ..... 379

#### 复位

BOR (欠压复位) ..... 89  
CM (配置不匹配复位) ..... 89  
IOPUWR (非法操作码复位) ..... 89  
MCLR (引脚复位) ..... 89  
POR (上电复位) ..... 89  
器件时间 ..... 93  
欠压复位 (BOR) ..... 93  
RCON 标志, 操作 ..... 93  
SFR 状态 ..... 93  
SWR (RESET 指令) ..... 89  
时钟源选择 ..... 93  
TRAPR (陷阱冲突复位) ..... 89  
WDT (看门狗定时器复位) ..... 89  
UWR (未初始化的 W 寄存器复位) ..... 89  
延时时间 ..... 94

# PIC24FJ128GA310 系列

## G

高 / 低压检测 (HLVD) .....	331
公式	
16 位和 32 位 CRC 多项式 .....	290
A/D 转换时钟周期 .....	311
BRGH = 0 时的 UART 波特率 .....	242
BRGH = 1 时的 UART 波特率 .....	242
波特率重载值 .....	235
计算 PWM 周期 .....	214
计算最大 PWM 分辨率 .....	215
器件速率和 SPI 时钟速率之间的关系 .....	231

## H

汇编器	
MPASM 汇编器 .....	348

## I

I/O 端口	
并行 (PIO) .....	167
可选输入源 .....	174
漏极开路配置 .....	168
模拟端口引脚配置 (ANSx) .....	168
配置 I/O 引脚的模拟 / 数字功能 .....	168
上拉和下拉 .....	172
输入电平变化通知 .....	172
外设引脚选择 .....	173

## I<sup>2</sup>C

保留地址 .....	235
从地址掩码 .....	235
设置用作总线主器件时的波特率 .....	235
时钟速率 .....	235
作为主器件在单主器件环境中通信 .....	233

## J

寄存器	
AD1CHITH (A/D 扫描比较命中, 高位字) .....	308
AD1CHITL (A/D 扫描比较命中, 低位字) .....	308
AD1CHS (A/D 采样选择) .....	306
AD1CON1 (A/D 控制 1) .....	300
AD1CON2 (A/D 控制 2) .....	302
AD1CON3 (A/D 控制 3) .....	303
AD1CON4 (A/D 控制 4) .....	304
AD1CON5 (A/D 控制 5) .....	305
AD1CSSH (A/D 输入扫描选择, 高位字) .....	309
AD1CSSL (A/D 输入扫描选择, 低位字) .....	309
AD1CTMENH (CTMU 使能, 高位字) .....	310
AD1CTMENL (CTMU 使能, 低位字) .....	310
ALCFCRPT (闹钟配置) .....	280
ALMINSEC (闹钟分钟和秒值) .....	284
ALMTHDY (闹钟月和日值) .....	283
ALWDHRS (闹钟星期和小时值) .....	283
ANCFG (A/D 带隙参考) .....	307
ANSA (PORTA 模拟功能选择) .....	169
ANSB (PORTB 模拟功能选择) .....	169
ANSC (PORTC 模拟功能选择) .....	170
ANSD (PORTD 模拟功能选择) .....	170
ANSE (PORTE 模拟功能选择) .....	171
ANSG (PORTG 模拟功能选择) .....	171
CLKDIV (时钟分频器) .....	149
CMSTAT (比较器状态) .....	319
CMxCON (比较器 x 控制, 比较器 1-3) .....	318
CORCON (CPU 内核控制) .....	39, 100
CRCCON1 (CRC 控制 1) .....	292
CRCCON2 (CRC 控制 2) .....	293
CRCXORH (CRC 异或操作, 高字节) .....	294

CRCXORL (CRC 多项式异或操作, 低字节) .....	293
CTMUCON1 (CTMU 控制 1) .....	326
CTMUCON2 (CTMU 控制 2) .....	327
CTMUICON (CTMU 电流控制) .....	329
CW1 (闪存配置字 1) .....	334
CW2 (闪存配置字 2) .....	336
CW3 (闪存配置字 3) .....	338
CW4 (闪存配置字 4) .....	340
CVRCON (比较器参考电压控制) .....	322
DEVID (器件 ID) .....	342
DEVREV (器件版本) .....	342
DMACHn (DMA 通道 n 控制) .....	80
DMACON (DMA 引擎控制) .....	79
DMAINTn (DMA 通道 n 中断) .....	81
DSCON (深度休眠控制) .....	162
DSWAKE (深度休眠唤醒源) .....	163
HLVDCON (高 / 低压检测控制) .....	332
I2CxCON (I2Cx 控制) .....	236
I2CxMSK (I2Cx 从模式地址掩码) .....	239
I2CxSTAT (I2Cx 状态) .....	238
ICxCON1 (输入捕捉 x 控制 1) .....	207
ICxCON2 (输入捕捉 x 控制 2) .....	208
IEC0 (中断允许控制 0) .....	112
IEC1 (中断允许控制 1) .....	114
IEC2 (中断允许控制 2) .....	116
IEC3 (中断允许控制 3) .....	117
IEC4 (中断允许控制 4) .....	118
IEC5 (中断允许控制 5) .....	119
IEC6 (中断允许控制 6) .....	120
IEC7 (中断使能控制 7) .....	120
IFS0 (中断标志状态 0) .....	103
IFS1 (中断标志状态 1) .....	105
IFS2 (中断标志状态 2) .....	107
IFS3 (中断标志状态 3) .....	108
IFS4 (中断标志状态 4) .....	109
IFS5 (中断标志状态 5) .....	110
IFS6 (中断标志状态 6) .....	111
IFS7 (中断标志状态 7) .....	111
INTCON1 (中断控制 1) .....	101
INTCON2 (中断控制 2) .....	102
INTTREG (中断控制器测试) .....	142
IPC0 (中断优先级控制 0) .....	121
IPC1 (中断优先级控制 1) .....	122
IPC10 (中断优先级控制 10) .....	131
IPC11 (中断优先级控制 11) .....	132
IPC12 (中断优先级控制 12) .....	133
IPC13 (中断优先级控制 13) .....	134
IPC15 (中断优先级控制 15) .....	135
IPC16 (中断优先级控制 16) .....	136
IPC18 (中断优先级控制 18) .....	137
IPC19 (中断优先级控制 19) .....	137
IPC2 (中断优先级控制 2) .....	123
IPC20 (中断优先级控制 20) .....	138
IPC21 (中断优先级控制 21) .....	139
IPC22 (中断优先级控制 22) .....	140
IPC25 (中断优先级控制 25) .....	141
IPC29 (中断优先级控制 29) .....	141
IPC3 (中断优先级控制 3) .....	124
IPC4 (中断优先级控制 4) .....	125
IPC5 (中断优先级控制 5) .....	126
IPC6 (中断优先级控制 6) .....	127
IPC7 (中断优先级控制 7) .....	128
IPC8 (中断优先级控制寄存器 8) .....	129
IPC9 (中断优先级控制 9) .....	130
LCDCREG (LCD 电荷泵控制) .....	268

LCDDATAx (LCD 像素数据 x) .....	270
LCDREF (LCD 参考电压梯形电阻网络控制) .....	272
LCDSEX (LCD 段 x 使能) .....	270
MDCAR (调制器载波控制) .....	252
MDCON (调制器控制) .....	250
MDSRC (调制器源控制) .....	251
MINSEC (RTCC 分钟和秒值) .....	282
MTHDY (RTCC 月和日值) .....	281
OCxCON1 (输出比较 x 控制 1) .....	216
OCxCON2 (输出比较 x 控制 2) .....	218
OSCCON (振荡器控制) .....	147
OSCTUN (FRC 振荡器调节) .....	150
PADCFG1 (填充配置控制) .....	263
PMCON1 (EPMP 控制 1) .....	255
PMCON2 (EPMP 控制 2) .....	256
PMCON3 (EPMP 控制 3) .....	257
PMCON4 (EPMP 控制 4) .....	258
PMCSxBs (片选 x 基址址) .....	260
PMCSxCF (片选 x 配置) .....	259
PMCSxMD (片选 x 模式) .....	261
PMSTAT (EPMP 状态, 从模式) .....	262
RCFGCAL (RTCC 校准和配置) .....	277
RCON (复位控制) .....	90
RCON2 (复位和系统控制 2) .....	164
RCON2 (复位控制 2) .....	92
REFOCON (参考振荡器控制) .....	153
RPINR0 (PPS 输入 0) .....	178
RPINR1 (PPS 输入 1) .....	178
RPINR10 (PPS 输入 10) .....	182
RPINR11 (PPS 输入 11) .....	182
RPINR17 (PPS 输入 17) .....	183
RPINR18 (PPS 输入 18) .....	183
RPINR19 (PPS 输入 19) .....	184
RPINR2 (PPS 输入 2) .....	179
RPINR20 (PPS 输入 20) .....	184
RPINR21 (PPS 输入 21) .....	185
RPINR22 (PPS 输入 22) .....	185
RPINR23 (PPS 输入 23) .....	186
RPINR27 (PPS 输入 27) .....	186
RPINR3 (PPS 输入 3) .....	179
RPINR30 (PPS 输入 30) .....	187
RPINR31 (PPS 输入 31) .....	187
RPINR4 (PPS 输入 4) .....	180
RPINR7 (PPS 输入 7) .....	180
RPINR8 (PPS 输入 8) .....	181
RPINR9 (PPS 输入 9) .....	181
RPOR0 (PPS 输出 0) .....	188
RPOR1 (PPS 输出 1) .....	188
RPOR10 (PPS 输出 10) .....	193
RPOR11 (PPS 输出 11) .....	193
RPOR12 (PPS 输出 12) .....	194
RPOR13 (PPS 输出 13) .....	194
RPOR14 (PPS 输出 14) .....	195
RPOR15 (PPS 输出 15) .....	195
RPOR2 (PPS 输出 2) .....	189
RPOR3 (PPS 输出 3) .....	189
RPOR4 (PPS 输出 4) .....	190
RPOR5 (PPS 输出 5) .....	190
RPOR6 (PPS 输出 6) .....	191
RPOR7 (PPS 输出 7) .....	191
RPOR8 (PPS 输出 8) .....	192
RPOR9 (PPS 输出 9) .....	192
RTCCSWT (电源控制和采样窗口定时器) .....	285
RTCPWC (RTCC 电源控制) .....	279
SPIxCON1 (SPIx 控制 1) .....	226
SPIxCON2 (SPIx 控制 2) .....	228
SPIxSTAT (SPIx 状态和控制) .....	224
SR (ALU 状态) .....	38, 99
T1CON (Timer1 控制) .....	198
TxCON (Timer2 和 Timer4 控制) .....	202
TyCON (Timer3 和 Timer5 控制) .....	203
WKDYHR (RTCC 星期和小时值) .....	282
UxMODE (UARTx 模式) .....	244
UxSTA (UARTx 状态和控制) .....	246
YEAR (RTCC 年值) .....	281
寄存器映射	
A/D 转换器 .....	56
比较器 .....	61
并行主 / 从端口 .....	60
CRC .....	62
CTMU .....	57
DMA .....	58
到 CPU 内核 .....	45
定时器 .....	48
I <sup>2</sup> C .....	51
ICN .....	46
LCD 控制器 .....	59
模拟配置 .....	57
NVM .....	64
PMD .....	64
PORTA .....	53
PORTB .....	53
PORTC .....	54
PORTD .....	54
PORTE .....	54
PORTF .....	55
PORTG .....	55
RTCC .....	61
SPI .....	53
深度休眠 .....	64
输出比较 .....	50
数据信号调制器 (DSM) .....	61
输入捕捉 .....	49
UART .....	52
外设引脚选择 .....	62
系统控制 .....	63
引脚配置 .....	55
中断控制器 .....	47
JTAG 接口 .....	346
交流特性	
A/D 转换时序 .....	375
内部 RC 精度 .....	371
时序规范的负载条件和要求 .....	369
节能特性 .....	155
打盹模式 .....	165
低压	
保持稳压器 .....	157
基于指令的模式 .....	156
空闲 .....	157
深度休眠 .....	158
休眠 .....	157
时钟频率和时钟切换 .....	165
Vbat 模式 .....	160

# PIC24FJ128GA310 系列

## K

开发支持 .....	347
看门狗定时器 (WDT) .....	344
窗口操作 .....	344
控制寄存器 .....	344
勘误表 .....	10
客户通知服务 .....	401
客户支持 .....	401
框图 .....	
10 位 A/D 转换器模拟输入模型 .....	311
12 位 A/D 转换器 .....	296
16 位 Timer1 模块 .....	197
16 位同步 Timer2 和 Timer4 .....	201
16 位异步 Timer3 和 Timer5 .....	201
32 位 Timer2/3 和 Timer4/5 .....	200
比较器参考电压 .....	321
表寄存器寻址 .....	83
CALL 堆栈帧 .....	68
CPU 编程模型 .....	37
CRC .....	289
CRC 移位引擎详细信息 .....	289
CREF = 0 时各个比较器配置 .....	316
产生脉冲延时的 CTMU 典型连接和内部配置 .....	325
DMA .....	75
电容测量的 CTMU 连接和内部配置 .....	324
读操作的 EDS 地址生成 .....	66
访问程序空间内的数据的地址生成方式 .....	70
复位系统 .....	89
高 / 低压检测 (HLVD) .....	331
各个比较器配置, CREF = 1 且 CVREFP = 0 .....	317
各个比较器配置, CREF = 1 且 CVREFP = 1 .....	317
共用 I/O 端口的结构 .....	167
I <sup>2</sup> C 模块 .....	234
看门狗定时器 (WDT) .....	344
LCD 控制器 .....	265
PCI24FJ256GA310 系列 (一般) .....	16
PIA 模式下的缓冲区地址生成 .....	299
PIC24F CPU 内核 .....	36
PSV 操作 (低字位) .....	73
PSV 操作 (高位字) .....	73
片内稳压器的连接 .....	343
RTCC .....	275
SPI 从器件、帧从器件连接 .....	230
SPI 从器件、帧主器件连接 .....	230
SPIx 模块 (标准, 模式) .....	222
SPIx 模块 (增强型模式) .....	223
SPI 主 / 从连接 (标准模式) .....	229
SPI 主 / 从连接 (增强型缓冲模式) .....	229
SPI 主器件、帧从器件连接 .....	230
SPI 主器件、帧主器件连接 .....	230
三比较器模块 .....	315
时间测量的 CTMU 典型连接和内部配置 .....	325
使用表指令访问程序空间操作 .....	71
输出比较 (16 位模式) .....	212
输出比较 (双缓冲, 16 位 PWM 模式) .....	214
数据信号调制器 .....	249
输入捕捉 .....	205
UART (简化) .....	241
写操作的 EDS 地址生成 .....	67
扩展数据空间 (EDS) .....	253

## L

LCD 控制器 .....	12
---------------	----

## M

Microchip 因特网网站 .....	401
MPLAB ASM30 汇编器、链接器和库管理器 .....	348
MPLAB PM3 器件编程器 .....	350
MPLAB REAL ICE 在线仿真器系统 .....	349
MPLAB 集成开发环境软件 .....	347
MPLINK 目标链接器 / MPLIB 目标库管理器 .....	348
脉宽调制。见 PWM。	
脉宽调制 (PWM) 模式 .....	213

## N

Near 数据空间 .....	44
内部集成电路。请参见 I <sup>2</sup> C。 .....	233
内核特性 .....	11

## P

PWM	
占空比和周期 .....	214
配置位 .....	333
片内稳压器 .....	343
待机模式 .....	343
POR .....	343

## Q

器件特性	
100 引脚 .....	15
64 引脚 .....	13
80 引脚 .....	14

## R

RTCC	
电源控制 .....	287
寄存器映射 .....	276
闹钟配置 .....	286
闹钟屏蔽设置 (图) .....	287
时钟源选择 .....	276
校准 .....	286
写锁定 .....	276
源时钟 .....	275
软件堆栈 .....	68
软件模拟器 (MPLAB SIM) .....	349

## S

SFR 空间 .....	44
SPI .....	221
三比较器 .....	315
三比较器模块 .....	315
闪存程序存储器 .....	83
编程操作 .....	84
编程单字 .....	88
编程算法 .....	86
和表指令 .....	83
JTAG 操作 .....	84
RTSP 操作 .....	84
增强型 ICSP 操作 .....	84
闪存配置字 .....	42
闪存配置字位置 .....	333
实时时钟和日历 (RTCC) .....	275
时序图 .....	
CLKO 和 I/O 时序 .....	372
外部时钟 .....	370
使用专用定时器的输出比较 .....	211
使用专用定时器的输入捕捉 .....	205
输出比较 .....	
32 位级联模式 .....	211
同步和触发模式 .....	211

数据存储器	
存储器映射	43
地址空间	43
空间构成, 对齐方式	44
扩展数据空间 (EDS)	65
Near 数据空间	44
软件堆栈	68
SFR 空间	44
数据信号调制器	249
数据信号调制器 (DSM)	249
输入捕捉	
32 位级联模式	206
操作	206
同步和触发模式	205
<b>T</b>	
Timer2/3 和 Timer4/5	199
Timer1	197
特性	12
通用异步收发器。见 <b>UART</b> 。	
<b>U</b>	
UART	241
波特率发生器 (BRG)	242
发送	
8 位数据模式	243
9 位数据模式	243
间隔或同步序列	243
红外支持	243
接收	
8 位或 9 位数据模式	243
UxCTS 和 UxRTS 控制引脚的操作	243
<b>W</b>	
WWW 地址	401
外设引脚选择 (PPS)	173
寄存器	178
可用外设和引脚	173
配置控制	176
使用注意事项	177
输出映射	175
输入映射	174
外设优先级	173
映射例外	176
<b>X</b>	
选择性外设模块控制	165
<b>Y</b>	
液晶显示 (LCD) 控制器	265
引脚描述	
121 引脚器件 (BGA)	7
引脚排列说明	17
因特网地址	401
<b>Z</b>	
增强型并行主端口。见 <b>EPMP</b> 。	253
增强型并行主端口 (EPMP)	253
振荡器配置	
CPU 时钟机制	146
参考时钟输出	152
辅助振荡器 (SOSC)	152
控制寄存器	147
上电复位时的初始配置	146
时钟切换	150
序列	151
用于时钟选择的位值	146
直接存储器访问控制器。见 <b>DMA</b> 。	
指令集	
操作码说明中使用的符号	352
概述	353
汇总	351
直流特性	
比较器	368
比较器参考电压	368
CTMU 电流源	367
程序存储器	366
掉电电流	363
复位	364
工作电流	361
I/O 引脚输出规范	366
I/O 引脚输入规范	365
空闲电流	362
Vbat 工作电压规范	367
温度和电压规范	361
中断	
复位过程	95
控制和状态寄存器	98
设置和服务过程	143
实现的向量	97
陷阱向量	96
向量表	96
中断向量表 (IVT)	95
主要特性	333

# **PIC24FJ128GA310 系列**

---

---

注:

## MICROCHIP 网站

Microchip 网站 ([www.microchip.com](http://www.microchip.com)) 为客户提供在线支持。客户可通过该网站方便地获取文件和信息。只要使用常用的互联网浏览器即可访问。网站提供以下信息：

- **产品支持**——数据手册和勘误表、应用笔记和示例程序、设计资源、用户指南以及硬件支持文档、最新的软件版本以及归档软件
- **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 顾问计划成员名单
- **Microchip 业务**——产品选型和订购指南、最新 Microchip 新闻稿、研讨会和活动安排表、Microchip 销售办事处、代理商以及工厂代表列表

## 变更通知客户服务

Microchip 的变更通知客户服务有助于客户了解 Microchip 产品的最新信息。注册客户可在他们感兴趣的某个产品系列或开发工具发生变更、更新、发布新版本或勘误表时，收到电子邮件通知。

欲注册，请登录 Microchip 网站 [www.microchip.com](http://www.microchip.com)。在“支持”(Support) 下，点击“变更通知客户 (Customer Change Notification)”服务后按照注册说明完成注册。

## 客户支持

Microchip 产品的用户可通过以下渠道获得帮助：

- 代理商或代表
- 当地销售办事处
- 应用工程师 (FAE)
- 技术支持

客户应联系其代理商、代表或应用工程师 (FAE) 寻求支持。当地销售办事处也可为客户提供帮助。本文档后附有销售办事处的联系方式。

也可通过 <http://microchip.com/support> 获得网上技术支持。

# PIC24FJ128GA310 系列

---

---

## 读者反馈表

我们努力为您提供最佳文档，以确保您能够成功使用 Microchip 产品。如果您对文档的组织、条理性、主题及其他有助于提高文档质量的方面有任何意见或建议，请填写本反馈表并传真给我公司 TRC 经理，传真号码为 86-21-5407-5066。请填写以下信息，并从下面各方面提出您对本文档的意见。

致 TRC 经理

总页数 \_\_\_\_\_

关于： 读者反馈

发自： 姓名 \_\_\_\_\_

公司 \_\_\_\_\_

地址 \_\_\_\_\_

国家 / 省份 / 城市 / 邮编 \_\_\_\_\_

电话：(\_\_\_\_\_) \_\_\_\_\_ 传真：(\_\_\_\_\_) \_\_\_\_\_

应用 (选填)：

您希望收到回复吗？是\_\_\_\_\_ 否\_\_\_\_\_

器件： PIC24FJ128GA310 系列

文献编号：

DS39996F\_CN

问题

1. 本文档中哪些部分最有特色？

\_\_\_\_\_

2. 本文档是否满足了您的软硬件开发要求？如何满足的？

\_\_\_\_\_

3. 您认为本文档的组织结构便于理解吗？如果不便于理解，那么问题何在？

\_\_\_\_\_

4. 您认为本文档应该添加哪些内容以改善其结构和主题？

\_\_\_\_\_

5. 您认为本文档中可以删减哪些内容，而又不会影响整体使用效果？

\_\_\_\_\_

6. 本文档中是否存在错误或误导信息？如果存在，请指出是什么信息及其具体页数。

\_\_\_\_\_

7. 您认为本文档还有哪些方面有待改进？

\_\_\_\_\_

## 产品标识体系

欲订货或获取价格、交货等信息，请与我公司生产厂或各销售办事处联系。

PIC 24 FJ 128 GA3 10 T - I / PT - XXX	
Microchip 商标	
架构	
闪存系列	
程序存储器大小 (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	

**示例：**

a) PIC24FJ64GA306-I/MR:  
具有 LCD 控制器、nanoWatt XLP 技术、64 KB 程序存储器、64 引脚、工业级温度以及 QFN 封装的 PIC24F 器件。

b) PIC24FJ128GA308-I/PT:  
具有 LCD 控制器、nanoWatt XLP 技术、128 KB 程序存储器、80 引脚、工业级温度以及 TQFP 封装的 PIC24F 器件。

c) PIC24FJ128GA210-I/BG:  
具有 LCD 控制器、nanoWatt XLP 技术、128 KB 程序存储器、121 引脚、工业级温度以及 BGA 封装的 PIC24F 器件。

架构	24 = 不带 DSP 的 16 位改进型哈佛架构
闪存系列	FJ = 闪存程序存储器
产品组	GA3 = 具有 LCD 控制器和 nanoWatt XLP 技术的通用单片机
引脚数	06 = 64 引脚 08 = 80 引脚 10 = 100 引脚 (TQFP) 和 121 引脚 (BGA)
温度范围	I = -40°C 至 +85°C (工业级)
封装	BG = 121 引脚 (10x10x1.4 mm) BGA 封装 PT = 100 引脚 (12x12x1 mm) TQFP (薄型四方扁平) PF = 100 引脚 (14x14x1 mm) TQFP (薄型四方扁平) PT = 80 引脚 (12x12x1 mm) TQFP (薄型四方扁平) PT = 64 引脚 (10x10x1 mm) TQFP (薄型四方扁平) MR = 64 引脚 (9x9x0.9 mm) QFN (四方扁平, 无引脚)
定制编号	三位 QTP、SQTP、编码或特殊要求 (其它情况均为空白) ES = 工程样片

# **PIC24FJ128GA310 系列**

---

---

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。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。

#### 商标

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

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

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

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

在此提及的所有其他商标均为各持有公司所有。

© 2012, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62076-573-9

**QUALITY MANAGEMENT SYSTEM  
CERTIFIED BY DNV  
— ISO/TS 16949 —**

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



**MICROCHIP**

## 全球销售及服务网点

### 美洲

#### 公司总部 Corporate Office

2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 1-480-792-7200  
Fax: 1-480-792-7277

技术支持:

[http://www.microchip.com/  
support](http://www.microchip.com/support)

网址: [www.microchip.com](http://www.microchip.com)

#### 亚特兰大 Atlanta

Duluth, GA  
Tel: 1-678-957-9614  
Fax: 1-678-957-1455

#### 波士顿 Boston

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

#### 芝加哥 Chicago

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

#### 克里夫兰 Cleveland

Independence, OH  
Tel: 1-216-447-0464  
Fax: 1-216-447-0643

#### 达拉斯 Dallas

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

#### 底特律 Detroit

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

#### 印第安纳波利斯 Indianapolis

Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453

#### 洛杉矶 Los Angeles

Mission Viejo, CA  
Tel: 1-949-462-9523  
Fax: 1-949-462-9608

#### 圣克拉拉 Santa Clara

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

#### 加拿大多伦多 Toronto

Mississauga, Ontario,  
Canada  
Tel: 1-905-673-0699  
Fax: 1-905-673-6509

### 亚太地区

#### 亚太总部 Asia Pacific Office

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

中国 - 北京  
Tel: 86-10-8569-7000  
Fax: 86-10-8528-2104

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

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

中国 - 杭州  
Tel: 86-571-2819-3187  
Fax: 86-571-2819-3189

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

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

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

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

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

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

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

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

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

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

### 亚太地区

台湾地区 - 高雄  
Tel: 886-7-536-4818  
Fax: 886-7-330-9305

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

台湾地区 - 新竹  
Tel: 886-3-5778-366  
Fax: 886-3-5770-955

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

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

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

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

日本 Japan - Osaka  
Tel: 81-66-152-7160  
Fax: 81-66-152-9310

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

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

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

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

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

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

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

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

### 欧洲

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

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

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

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

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

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

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

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