

---

## 第 31 章 DMA 控制器

---

### 目录

本章包括下列主题：

31.1	简介 .....	31-2
31.2	状态和控制寄存器 .....	31-6
31.3	工作模式 .....	31-29
31.4	通道控制 .....	31-40
31.5	中断 .....	31-44
31.6	节能和调试模式下的操作 .....	31-46
31.7	各种复位的影响 .....	31-46
31.8	相关应用笔记 .....	31-47
31.9	版本历史 .....	31-48

**注：** 本系列参考手册章节旨在用作对器件数据手册的补充。根据不同的器件型号，本手册章节可能并不适用于所有 PIC32 器件。

请参见当前器件数据手册中“**直接存储器访问（DMA）控制器**”章节开头部分的注释，以检查本文档是否支持您所使用的器件。

器件数据手册和系列参考手册章节可从 Microchip 网站下载：<http://www.microchip.com>。

## 31.1 简介

直接存储器访问（Direct Memory Access, DMA）控制器是总线主模块，用于无需 CPU 干预的情况下在不同外设之间传输数据。DMA 传输的源和目标可以是 PIC32 系列器件中包含的任何存储器映射的模块。例如，存储器本身，或外设总线（Peripheral Bus, PB）设备之一：如 SPI 或 UART 等。

DMA 模块的主要特性包括：

- 根据不同的器件型号，最多提供 8 个相同的通道，每个通道都具有：
  - 自动递增源和目标的地址寄存器
  - 源指针和目标指针
- 根据不同的器件型号，最多支持 64 KB 的不可中断数据传输
- 灵活的数据传输，具有以下特性：
  - 传输粒度，细到字节级别
  - 无需在源和目标处对字节进行字对齐
- 固定优先级通道仲裁
- 灵活的 DMA 通道工作模式，包括：
  - 手动（软件）或自动（中断）DMA 请求
  - 单数据块或自动重复数据块传输模式
  - 通道至通道链接
- 灵活的 DMA 请求，具有以下特性：
  - 可从任何外设中断源选择 DMA 请求
  - 每个通道可以选择任何中断作为其 DMA 请求源
  - 可从任何外设中断源选择 DMA 传输中止
  - 数据模式匹配时，自动传输终止
- 多个 DMA 通道状态中断，提供：
  - DMA 通道数据块传输完成
  - 源空或半空
  - 目标满或半满
  - 由于外部事件导致 DMA 传输中止
  - 产生无效 DMA 地址
- DMA 调试支持以下特性：
  - DMA 通道最近访问的地址
  - 最近传输数据的 DMA 通道
- CRC 发生模块，具有以下特性：
  - CRC 模块可分配给任何可用通道
  - 在某些器件型号上，可以对从源读取的数据重新排序
  - CRC 模块具有很强的可配置能力
  - CRC 计算

DMA 控制器还提供了以下特性：

- 源和目标大小不同
- 存储器至存储器传输
- 存储器至外设传输
- 通道自动使能
- 事件启动 / 停止
- 模式匹配检测
- 通道链接

### 31.1.1 DMA 工作原理

DMA 通道可以在无需 CPU 干预的情况下将数据从源传输到目标。源和目标起始地址分别定义源和目标的起始地址。

源和目标的大小均可独立配置，并且传输的字节数与源和目标大小无关。

传输通过软件或通过中断请求启动。用户可以选择器件上的任意中断来启动 DMA 传输。

在传输启动时，DMA 控制器将执行单元传输（由单元大小寄存器定义），并且通道保持使能，直到数据块中的所有字节（源或目标中的较大者）传输完成为止。当通道被禁止时，将禁止进一步的传输，直到通道重新使能为止。

DMA 通道使用独立的指针来跟踪源和目标的当前字单元。

在源 / 目标指针处于源 / 目标大小一半位置时，或者在源 / 目标计数器达到源 / 目标结束位置时，可以产生中断。

DMA 传输可以通过软件、通过模式匹配或通过中断事件中止。在检测到地址错误时，传输也会停止。

图 31-1 显示了典型的 DMA 传输。通过将源大小（DCHxSSIZ）和目标大小（DCHxDSIZ）设置为 4 和 2 字节（块大小为 4 字节）来设定块传输大小。从而为源（DCHxSSA）和目标（DCHxDSA）寄存器提供了起始地址位置。源地址是称为缓冲区阵列的物理 SRAM 位置。目标地址为物理 PMDIN（PMP 输出缓冲区）存储器位置。单元大小（DCHxCSIZ）也设置为 2 字节。这意味着 4 字节的块传输将完成 2 次 2 字节的单元传输。对于 DMA 而言传输事件设为 PMP 写操作，也就是当 PMP 写操作发生时开始单元传输。请注意，将 DCHxCON 寄存器中的 CHAEN 位置 1 将自动使能 DMA 通道。向 DCHxECON 寄存器中的 CFORCE 位写入 1 也将强制开始 DMA 传输。如果通道自动使能，则在块传输结束时，所有通道寄存器会在开始单元传输之前复位到其初始设置状态。否则，DMA 通道将禁止。

图 31-1: 典型的 DMA 源 - 目标传输图

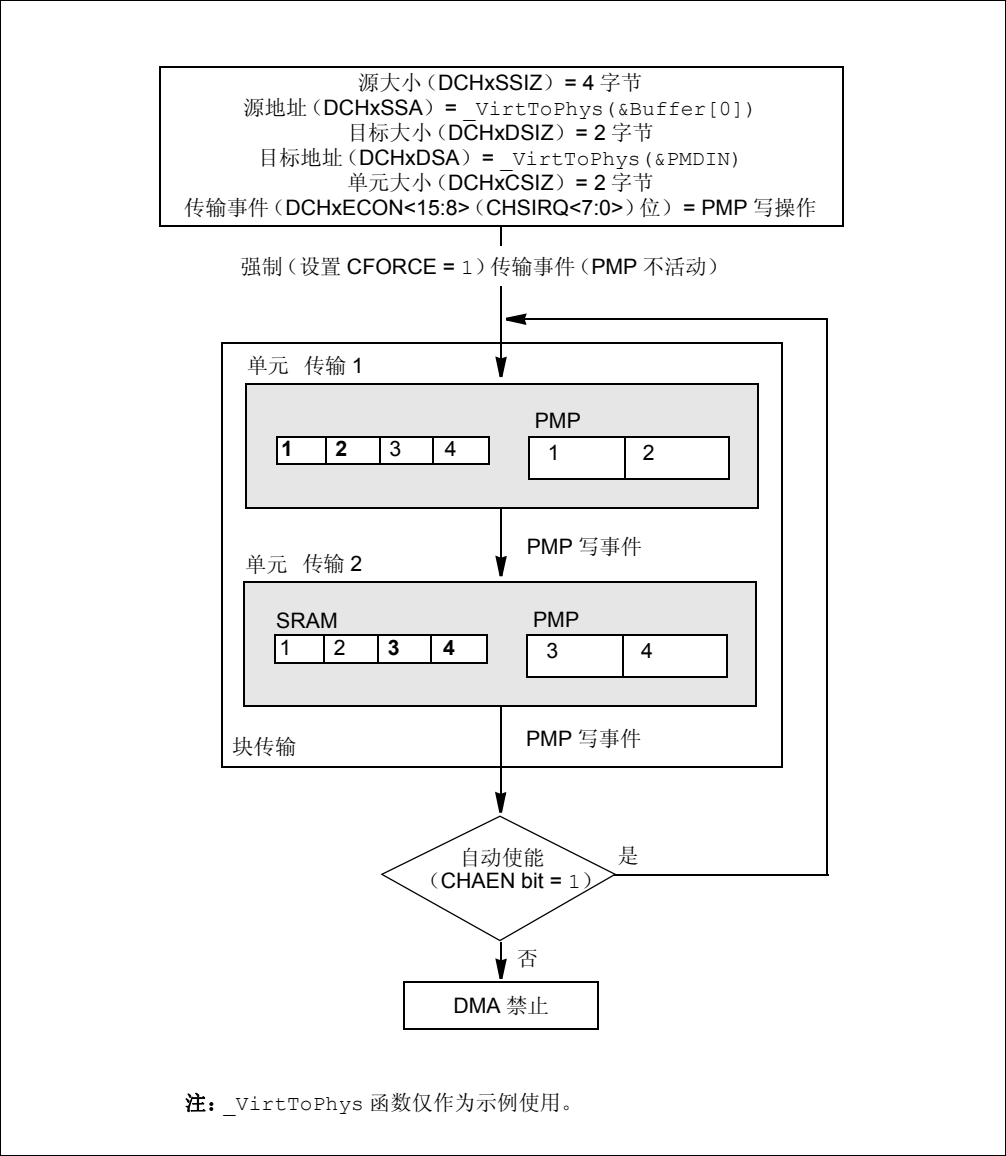
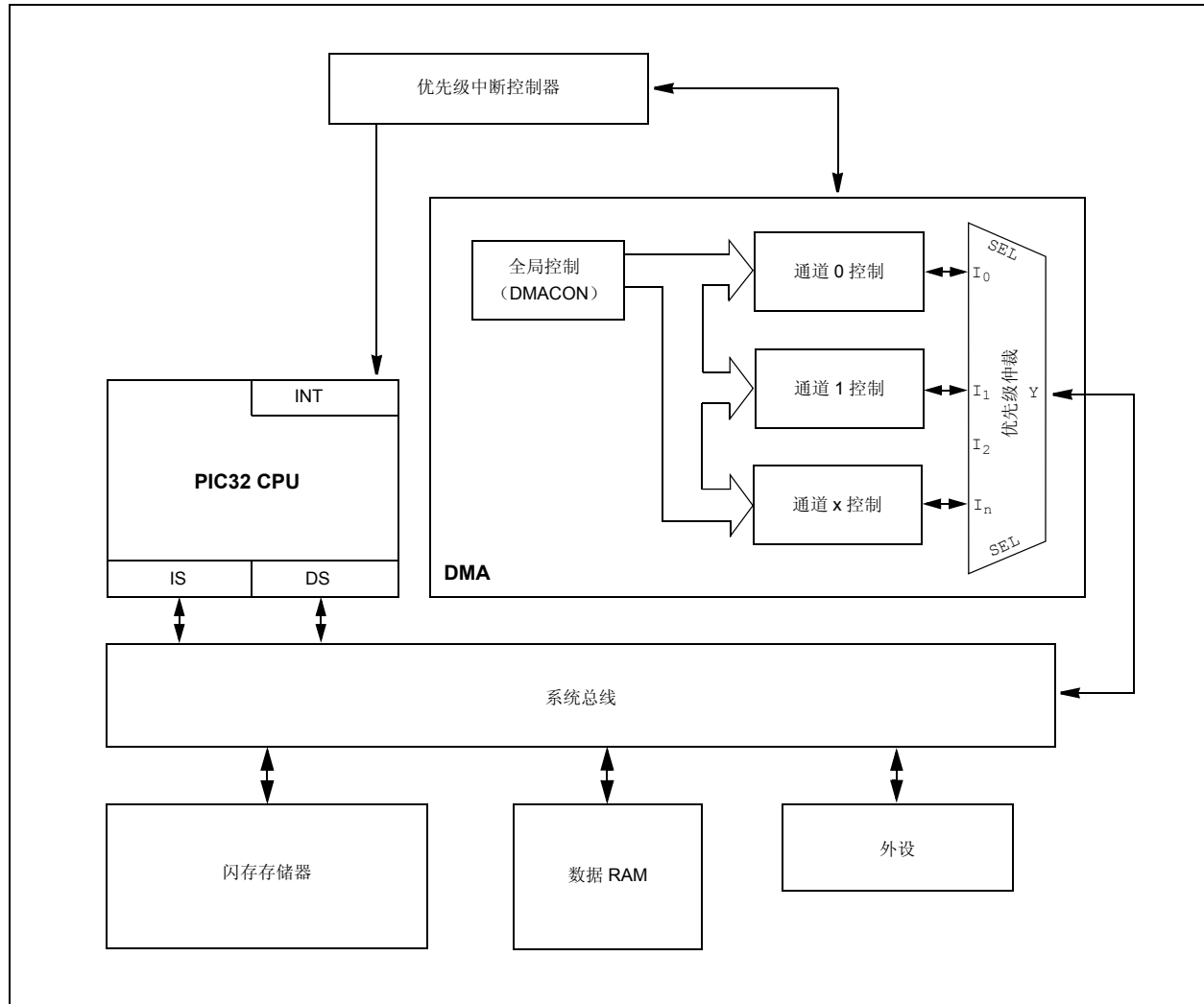


图 31-2: DMA 模块框图



## 31.2 状态和控制寄存器

**注：** PIC32 器件可能具有一个或多个 DMA 通道。在控制 / 状态位和寄存器的名称中使用的“x”表示特定的通道。更多详细信息，请参见具体器件数据手册的“直接存储器访问 (DMA) 控制器”章节。

DMA 模块包含以下特殊功能寄存器 (Special Function Register, SFR)：

- **DMACON: DMA 控制器控制寄存器**  
该寄存器配置相应的 DMA 通道。
- **DMASTAT: DMA 状态寄存器**  
该寄存器包含发生的最近读或写传输的状态。
- **DMAADDR: DMA 地址寄存器**  
该寄存器包含了最近 DMA 传输的地址。
- **DCRCCON: DMA CRC 控制寄存器**  
该寄存器控制 DMA CRC 及其如何工作。
- **DCRCDATA: DMA CRC 数据寄存器**  
该寄存器设置 CRC 发生器的初始值。写入该寄存器将设置 CRC 发生器种子。读取该寄存器将返回 CRC 的当前值。
- **DCRCXOR: DMA CRCXOR 使能寄存器**  
该寄存器提供用于 CRC 计算的发生器多项式的说明。
- **DCHxCON: DMA 通道 x 控制寄存器**  
该寄存器控制特定 DMA 通道的配置。
- **DCHxECON: DMA 通道 x 事件控制寄存器**  
该寄存器控制特定 DMA 通道的事件。
- **DCHxINT: DMA 通道 x 中断控制寄存器**  
该寄存器控制特定 DMA 通道的 DMA 中断。
- **DCHxSSA: DMA 通道 x 源起始地址寄存器**  
该寄存器配置特定 DMA 通道的源起始地址。
- **DCHxDSA: DMA 通道 x 目标起始地址寄存器**  
该寄存器配置特定 DMA 通道的目标起始地址。
- **DCHxSSIZ: DMA 通道 x 源大小寄存器**  
该寄存器配置特定 DMA 通道的源大小。
- **DCHxDSIZ: DMA 通道 x 目标大小寄存器**  
该寄存器配置特定 DMA 通道的目标大小。
- **DCHxSPTR: DMA 通道 x 源指针寄存器**  
该寄存器包含特定 DMA 通道的源当前位置地址。
- **DCHxDPTR: DMA 通道 x 目标指针寄存器**  
该寄存器包含特定 DMA 通道的目标当前位置地址。
- **DCHxCSIZ: DMA 通道 x 单元大小寄存器**  
该寄存器配置特定 DMA 通道每次事件发生时的传输次数。
- **DCHxCPTR: DMA 通道 x 单元指针寄存器**  
该寄存器计数特定 DMA 通道从上次事件发生到现在的传输次数。
- **DCHxDAT: DMA 通道 x 模式数据寄存器**  
该寄存器包含用于匹配时终止特定 DMA 通道的匹配数据。

表 31-1 简要汇总了与 DMA 模块相关的寄存器。该汇总表之后列出了相应的寄存器，并且每个寄存器均附有详细的说明。

表 31-1: DMA 寄存器汇总

名称	位 范围	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
DMACON <sup>(1)</sup>	31:15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	ON	—	—	SUSPEND	DMABUSY <sup>(2)</sup>	—	—	—
DMASTAT	31:16	RDWR <sup>(2)</sup>	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	—	—	—	—	—	—	—	—	—	—	—	—	RDWR <sup>(2)</sup>	DMACH<2:0> <sup>(2)</sup>		
DMAADDR	31:16	DMAADDR<31:16>															
	15:0	DMAADDR<15:0>															
DCRCCON <sup>(1)</sup>	31:16	—	—	BYTO<1:0> <sup>(2)</sup>		WBO <sup>(2)</sup>		—	—	BITO <sup>(2)</sup>	—	—	—	—	—	—	—
	15:0	—	—	—	PLEN<4:0> <sup>(2)</sup>						CRCEN	CRCAPP	CRCTYP <sup>(2)</sup>	—	—	CRCCH<2:0> <sup>(2)</sup>	
DCRCDATA <sup>(1)</sup>	31:16	DCRCDATA<31:16> <sup>(2)</sup>															
	15:0	DCRCDATA<15:0> <sup>(2)</sup>															
DCRCXOR <sup>(1)</sup>	31:16	DCRCXOR<31:16> <sup>(2)</sup>															
	15:0	DCRCXOR<15:0> <sup>(2)</sup>															
DCHxCON <sup>(1)</sup>	31:16	CHPIGN<7:0>									—	—	—	—	—	—	—
	15:0	CHBUSY <sup>(2)</sup>	—	CHPIGNE <sup>(2)</sup>	—	CHPATLEN <sup>(2)</sup>	—	—	—	CHCHNS	CHEN	CHAED	CHCHN	CHAEN	—	CHEDET	CHPRI<1:0>
DCHxECON <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	CHAIRQ<7:0>							
	15:0	CHSIRQ<7:0>									CFORCE	CABORT	PATEN	SIRQEN	AIRQEN	—	—
DCHxINT <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	CHSDIE	CHSHIE	CHDDIE	CHDHIE	CHBCIE	CHCCIE	CHTAIE	CHERIE
	15:0	—	—	—	—	—	—	—	—	CHSDIF	CHSHIF	CHDDIF	CHDHIF	CHBCIF	CHCCIF	CHTAIF	CHERIF
DCHxSSA <sup>(1)</sup>	31:16	CHSSA<31:16>															
	15:0	CHSSA<15:0>															
DCHxDSA	31:16	CHDSA<31:16>															
	15:0	CHDSA<15:0>															
DCHxSSIZ <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHSSIZ<15:0> <sup>(2)</sup>															
DCHxDSIZ <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHDSIZ<15:0> <sup>(2)</sup>															
DCHxSPTR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHSPTR<15:0>															

图注: — = 未实现，读为 0。地址偏移量以十六进制表示。

注

1: 这些寄存器具有关联的清零、置 1 和取反寄存器，分别位于 0x4、0x8 和 0xC 字节偏移处。清零、置 1 和取反寄存器的名称分别为相应寄存器的名称加上 CLR、SET 或 INV 后缀（如 DMACONCLR）。向这些寄存器的任意位写入 1 时，会将关联寄存器中的有效位清零、置 1 或取反。对这些寄存器的读操作将被忽略。

2: 不是所有器件都有该位。关于可用性，请参见具体数据手册的“直接存储器访问（DMA）控制器”章节。

表 31-1: DMA 寄存器汇总 (续)

名称	位范围	Bit 31/15	Bit 30/14	Bit 29/13	Bit 28/12	Bit 27/11	Bit 26/10	Bit 25/9	Bit 24/8	Bit 23/7	Bit 22/6	Bit 21/5	Bit 20/4	Bit 19/3	Bit 18/2	Bit 17/1	Bit 16/0
DCHxDPTR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHDPTR<15:0> <sup>(2)</sup>															
DCHxCSIZ <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHCSIZ<15:0> <sup>(2)</sup>															
DCHxCPTR	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHCPTR<15:0> <sup>(2)</sup>															
DCHxDAT <sup>(1)</sup>	31:16	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	15:0	CHPDAT<15:8> <sup>(2)</sup>								CHPDAT<7:0>							

图注: — = 未实现, 读为 0。地址偏移量以十六进制表示。

注 1: 这些寄存器具有关联的清零、置 1 和取反寄存器, 分别位于 0x4、0x8 和 0xC 字节偏移处。清零、置 1 和取反寄存器的名称分别为相应寄存器的名称加上 CLR、SET 或 INV 后缀 (如 DMAONCLR)。向这些寄存器的任意位写入 1 时, 会将关联寄存器中的有效位清零、置 1 或取反。对这些寄存器的读操作将被忽略。

2: 不是所有器件都有该位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。



寄存器 31-1: DMACON: DMA 控制器控制寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0
	ON	—	—	SUSPEND	DMABUSY <sup>(1)</sup>	—	—	—
7:0	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15 **ON**: DMA 使能位

1 = 使能 DMA 模块

0 = 禁止 DMA 模块

使用 1:1 PBCLK 分频比时, 在清零模块 ON 位的指令之后, 用户的软件不应立即在 SYSCLK 周期中读 / 写外设的 SFR。

bit 14-13 未实现: 读为 0

bit 12 **SUSPEND**: DMA 暂停位

1 = DMA 传输暂停, 以允许 CPU 无中断地访问数据总线

0 = DMA 正常工作

bit 11 **DMABUSY**: DMA 模块忙状态位 <sup>(1)</sup>

1 = DMA 模块处于活动状态

0 = DMA 模块已被禁止, 当前不在传输数据

bit 10-0 未实现: 读为 0

注 1: 不是所有器件都有该位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

---

--	--	--	--	--	--	--	--	--

---

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

bit 2-0 **DMACH<2:0>**: DMA 通道位  
当检测到错误时, 这些位包含最近激活的 DMA 通道值

\_\_\_\_\_

寄存器 31-3: DMAADDR: DMA 地址寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<31:24>							
23:16	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<23:16>							
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	DMAADDR<7:0>							

图注:

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

bit 31-0 DMAADDR<31:0>: DMA 模块地址位

注: 该寄存器包含最近 DMA 访问的地址。

**寄存器 31-4: DCRCCON: DMA CRC 控制寄存器**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0
	—	—	BYTO<1:0> <sup>(1)</sup>		WBO <sup>(1,2)</sup>	—	—	BITO <sup>(1)</sup>
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	—	—	—	PLEN<4:0> <sup>(2)</sup>				
7:0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
	CRCEN	CRCAPP <sup>(2)</sup>	CRCTYP <sup>(1)</sup>	—	—	CRCCH<2:0> <sup>(1)</sup>		

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-30 **未实现:** 读为 0

bit 29-28 **BYTO<1:0>:** CRC 字节顺序选择位 <sup>(1)</sup>

11 = 在半字节边界处进行字节顺序交换 (即, 使用源数据的半字顺序, 并对于每半个字, 使用源字节的相反顺序)

10 = 在字边界处交换半字 (即, 使用源数据半字的相反顺序, 并对于每半个字, 使用源字节顺序)

01 = 在字边界处进行字节顺序交换 (即, 使用源字节的相反顺序)

00 = 不交换 (即, 使用源数据的字节顺序)

bit 27 **WBO:** CRC 写字节顺序选择位 <sup>(1,2)</sup>

1 = 源数据按照 BYTO<1:0> 的定义重新排序后写入目标

0 = 源数据在保持不变的情况下写入目标

bit 26-25 **未实现:** 读为 0

bit 24 **BITO:** CRC 位顺序选择位 <sup>(2)</sup>

1 = 校验和 /CRC 使用从最低有效位 (LSb) 开始的方式计算 (即, 进行反射)

0 = 校验和 /CRC 使用从最高有效位 (MSb) 开始的方式计算 (即, 不进行反射)

bit 23-13 **未实现:** 读为 0

bit 12-8 **PLEN<4:0>:** 多项式长度位 <sup>(2)</sup>

当 CRCTYP (DCRCCON<15>) = 1 时 (CRC 模块处于 IP 头模式):

不使用这些位。

当 CRCTYP (DCRCCON<15>) = 0 时 (CRC 模块处于 LFSR 模式):

表示多项式长度 - 1。

bit 7 **CRCEN:** CRC 使能位

1 = 使能 CRC 模块, 通道传输经过 CRC 模块

0 = 禁止 CRC 模块, 通道传输正常进行

bit 6 **CRCAPP:** CRC 追加模式位 <sup>(2)</sup>

1 = DMA 将数据从源传输到 CRC 中, 但不传输到目标中。当数据块传输完成时, DMA 会将计算得到的 CRC 值写入由 CHxDSA 指定的单元中

0 = 在 DMA 将数据从源写入目标时, 它会按照 WBO 的设置将数据传输经过 CRC

**注 1:** 不是所有器件都有所有的位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

**2:** 当 WBO = 1 时, 不支持未对齐传输, 并且 CRCAPP 位不能置 1。

**寄存器 31-4: DCRCCON: DMA CRC 控制寄存器 (续)**

bit 5	<b>CRCTYP:</b> CRC 类型选择位 <sup>(1)</sup> 1 = CRC 模块将计算 IP 头校验和 0 = CRC 模块将计算 LFSR CRC
bit 4-3	<b>未实现:</b> 读为 0
bit 2-0	<b>CRCCH&lt;2:0&gt;:</b> CRC 通道选择位 <sup>(1)</sup> 111 = CRC 分配给通道 7 110 = CRC 分配给通道 6 101 = CRC 分配给通道 5 100 = CRC 分配给通道 4 011 = CRC 分配给通道 3 010 = CRC 分配给通道 2 001 = CRC 分配给通道 1 000 = CRC 分配给通道 0

- 注 1:** 不是所有器件都有所有的位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。
- 2:** 当 WBO = 1 时, 不支持未对齐传输, 并且 CRCAPP 位不能置 1。

寄存器 31-5: DCRCDATA: DMA CRC 数据寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCDATA<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCDATA<23:16>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCDATA<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCDATA<7:0>							

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

bit 31-0 **DCRCDATA<31:0>**: CRC 数据寄存器位

写入该寄存器会为 CRC 发生器设置种子值。读取该寄存器将返回 CRC 的当前值。在每次读取时, 高于 PLEN<4:0> (DCRCCON<12:8>) 的位将返回 0。

当 CRCTYP (DCRCCON<15>) = 1 时 (CRC 模块处于 IP 头模式):

只有低 16 位包含 IP 头校验和信息。高 16 位总是为 0。写入该寄存器的数据会被进行转换, 并以二进制反码的形式回读 (即, 当前 IP 头校验和值)。

当 CRCTYP (DCRCCON<15>) = 0 时 (CRC 模块处于 LFSR 模式):

在每次读取时, 高于 PLEN<4:0> (DCRCCON<12:8>) 的位将返回 0。

注:	不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的 “直接存储器访问 (DMA) 控制器” 章节。
----	---

寄存器 31-6: DCRCXOR: DMA CRCXOR 使能寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCXOR<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCXOR<23:16>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCXOR<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	DCRCXOR<7:0>							

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-0 DCRCXOR&lt;31:0&gt;: CRC 异或寄存器位

当 CRCTYP (DCRCCON&lt;15&gt;) = 1 时 (CRC 模块处于 IP 头模式):

不使用该寄存器。

当 CRCTYP (DCRCCON&lt;15&gt;) = 0 时 (CRC 模块处于 LFSR 模式):

1 = 使能移位寄存器的异或输入

0 = 禁止移位寄存器的异或输入; 数据从寄存器中的前一级直接移入

注: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

**寄存器 31-7: DCHxCON: DMA 通道 x 控制寄存器**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
	CHPIGN<7:0> <sup>(1)</sup>							
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	U-0	R/W-0	U-0	R/W-0	U-0	U-0	R/W-0
	CHBUSY <sup>(1)</sup>	—	CHPIGNEN <sup>(1)</sup>	—	CHPATLEN <sup>(1)</sup>	—	—	CHCHNS <sup>(2)</sup>
7:0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R-0	R/W-0	R/W-0
	CHEN <sup>(3)</sup>	CHAED	CHCHN	CHAEN	—	CHEDET	CHPRI<1:0>	

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-24 **CHPIGN<7:0>**: 通道寄存器数据位 <sup>(1)</sup>

**模式终止模式:**

在模式匹配确定期间当 CHPIGNEN 位置 1 时, 模式匹配时与这些位匹配的任何字节可被忽略。当模式匹配逻辑使能且 CHPIGNEN 位置 1 时, 如果读取的字节与该数据字节一致, 模式匹配逻辑会将其作为“无关”对待。

bit 23-16 **未实现**: 读为 0

bit 15 **CHBUSY**: 通道忙状态位 <sup>(1)</sup>

1 = 通道处于活动状态或已使能

0 = 通道处于非活动状态, 并且已禁止

bit 14 **未实现**: 读为 0

bit 13 **CHPIGNEN**: 使能模式忽略字节位 <sup>(1)</sup>

1 = 当使能模式匹配时, 对任何与 CHPIGN<7:0> 位匹配的字节作为“无关”对待

0 = 禁止该功能

bit 12 **未实现**: 读为 0

bit 11 **CHPATLEN**: 模式长度位 <sup>(1)</sup>

1 = 2 字节长度

0 = 1 字节长度

bit 8 **CHCHNS**: 链接通道选择位 <sup>(2)</sup>

1 = 与自然优先级较低的通道链接 (CH1 将在 CH2 传输完成时使能)

0 = 与自然优先级较高的通道链接 (CH1 将在 CH0 传输完成时使能)

bit 7 **CHEN**: 通道使能位 <sup>(3)</sup>

1 = 使能通道

0 = 禁止通道

bit 6 **CHAED**: 通道禁止时允许事件位

1 = 即使通道被禁止时, 也登记通道启动 / 中止事件

0 = 通道被禁止时, 将忽略通道启动 / 中止事件

bit 5 **CHCHN**: 通道链接使能位

1 = 允许对通道进行链接

0 = 不允许对通道进行链接

**注 1:** 不是所有器件都有该位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

**2:** 链接通道选择位在使能通道链接 (即, CHCHN = 1) 时有效。

**3:** 当通过清零该位暂停通道时, 用户应用程序应通过查询 CHBUSY 位 (如果器件上提供该位) 来确定通道何时被暂停, 因为在通道暂停之前, 可能需要一些时钟周期来完成当前事务。



**寄存器 31-7: DCHxCON: DMA 通道 x 控制寄存器 (续)**

- bit 4      **CHAEN**: 通道自动使能位  
1 = 连续使能通道, 在数据块传输完成之后不自动禁止  
0 = 在数据块传输完成时禁止通道
- bit 3      **未实现**: 读为 0
- bit 2      **CHEDET**: 通道事件检测位  
1 = 检测到事件  
0 = 未检测到事件
- bit 1-0    **CHPRI<1:0>**: 通道优先级位  
11 = 通道优先级为 3 (最高)  
10 = 通道优先级为 2  
01 = 通道优先级为 1  
00 = 通道优先级为 0

- 注 1:** 不是所有器件都有该位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。
- 2:** 链接通道选择位在使能通道链接 (即, CHCHN = 1) 时有效。
- 3:** 当通过清零该位暂停通道时, 用户应用程序应通过查询 CHBUSY 位 (如果器件上提供该位) 来确定通道何时被暂停, 因为在通道暂停之前, 可能需要一些时钟周期来完成当前事务。

**寄存器 31-8: DCHxECON: DMA 通道 x 事件控制寄存器**

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	CHAIRQ<7:0>							
15:8	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	CHSIRQ<7:0>							
7:0	S-0	S-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
	CFORCE	CABORT	PATEN	SIRQEN	AIRQEN	—	—	—

**图注:**

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-24 **未实现:** 读为 0

bit 23-16 **CHAIRQ<7:0>:** 通道传输中止 IRQ 位

11111111 = 中断 255 将中止任何正在进行的传输并将 CHAIF 标志置 1

•  
•  
•

00000001 = 中断 1 将中止任何正在进行的传输并将 CHAIF 标志置 1

00000000 = 中断 0 将中止任何正在进行的传输并将 CHAIF 标志置 1

bit 15-8 **CHSIRQ<7:0>:** 通道传输启动 IRQ 位

11111111 = 中断 255 将启动 DMA 传输

•  
•  
•

00000001 = 中断 1 将启动 DMA 传输

00000000 = 中断 0 将启动 DMA 传输

bit 7 **CFORCE:** DMA 强制传输位

1 = 向该位写入 1 时, 将强制开始 DMA 传输

0 = 该位总是读为 0

bit 6 **CABORT:** DMA 中止传输位

1 = 向该位写入 1 时, 将中止 DMA 传输

0 = 该位总是读为 0

bit 5 **PATEN:** 通道模式匹配中止使能位

1 = 在发生模式匹配时中止传输并清零 CHEN

0 = 禁止模式匹配

bit 4 **SIRQEN:** 通道启动 IRQ 使能位

1 = 如果发生与 CHSIRQ 匹配的中断, 则启动通道单元传输

0 = 忽略中断号 CHSIRQ, 并且不启动传输

bit 3 **AIRQEN:** 通道中止 IRQ 使能位

1 = 如果发生与 CHAIRQ 匹配的中断, 则中止通道传输

0 = 忽略中断号 CHAIRQ, 并且不中止传输

bit 2-0 **未实现:** 读为 0

寄存器 31-9: DCHxINT: DMA 通道 x 中断控制寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSDIE	CHSHIE	CHDDIE	CHDHIE	CHBCIE	CHCCIE	CHTAIE	CHERIE
15:8	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSDIF	CHSHIF	CHDDIF	CHDHIF	CHBCIF	CHCCIF	CHTAIF	CHERIF

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-24 未实现: 读为 0

bit 23 CHSDIE: 通道源完成中断允许位

1 = 允许中断

0 = 禁止中断

bit 22 CHSHIE: 通道源半空中断允许位

1 = 允许中断

0 = 禁止中断

bit 21 CHDDIE: 通道目标完成中断允许位

1 = 允许中断

0 = 禁止中断

bit 20 CHDHIE: 通道目标半满中断允许位

1 = 允许中断

0 = 禁止中断

bit 19 CHBCIE: 通道数据块传输完成中断允许位

1 = 允许中断

0 = 禁止中断

bit 18 CHCCIE: 通道单元传输完成中断允许位

1 = 允许中断

0 = 禁止中断

bit 17 CHTAIE: 通道传输中止中断允许位

1 = 允许中断

0 = 禁止中断

bit 16 CHERIE: 通道地址错误中断允许位

1 = 允许中断

0 = 禁止中断

bit 15-8 未实现: 读为 0

bit 7 CHSDIF: 通道源完成中断标志位

1 = 通道源指针已到达源结束位置 (CHSPTR = CHSSIZ)

0 = 没有待处理的中断

bit 6 CHSHIF: 通道源半空中断标志位

1 = 通道源指针已到达源中点位置 (CHSPTR = CHSSIZ/2)

0 = 没有待处理的中断

bit 5 CHDDIF: 通道目标完成中断标志位

1 = 通道目标指针已到达目标结束位置 (CHDPTR = CHDSIZ)

0 = 没有待处理的中断

## 寄存器 31-9: DCHxINT: DMA 通道 x 中断控制寄存器 (续)

- bit 4     **CHDHIF**: 通道目标半满中断标志位  
1 = 通道目标指针已到达目标中点位置 (CHDPTR = CHDSIZ/2)  
0 = 没有待处理的中断
- bit 3     **CHBCIF**: 通道数据块传输完成中断标志位  
1 = 数据块传输已完成 (已传输了 CHSSIZ/CHDSIZ 中较大者对应的字节数), 或者发生了模式匹配事件  
0 = 没有待处理的中断
- bit 2     **CHCCIF**: 通道单元传输完成中断标志位  
1 = 单元传输已完成 (已传输了 CHCSIZ 字节)  
0 = 没有待处理的中断
- bit 1     **CHTAIF**: 通道传输中止中断标志位  
1 = 已检测到与 CHAIRQ 匹配的中断, DMA 传输已中止  
0 = 没有待处理的中断
- bit 0     **CHERIF**: 通道地址错误中断标志位  
1 = 检测到通道地址错误  
      源地址或目标地址无效。  
0 = 没有待处理的中断

寄存器 31-10: DCHxSSA: DMA 通道 x 源起始地址寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSA<31:24>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSA<23:16>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSA<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSA<7:0>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-0 CHSSA&lt;31:0&gt;: 通道源起始地址位

这些位定义通道源的起始地址。

注: 该寄存器的值必须是源的物理地址。

寄存器 31-11: DCHxDSA: DMA 通道 x 目标起始地址寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSA<31:24> <sup>(1)</sup>							
23:16	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSA<23:16> <sup>(1)</sup>							
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSA<15:8> <sup>(1)</sup>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSA<7:0> <sup>(1)</sup>							

图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-0 CHDSA&lt;31:0&gt;: 通道目标起始地址位

这些位定义通道目标的起始地址。

注: 该寄存器的值必须是目标的物理地址。

# PIC32 系列参考手册

寄存器 31-12: DCHxSSIZ: DMA 通道 x 源大小寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSIZ<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHSSIZ<7:0>							

图注:

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

bit 31-16 未实现: 读为 0

bit 15-0 CHSSIZ<15:0>: 通道源大小位

1111111111111111 = 源大小为 65,535 字节

•

•

•

0000000000000010 = 源大小为 2 字节

0000000000000001 = 源大小为 1 字节

0000000000000000 = 源大小为 65,536 字节

注: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的 “直接存储器访问 (DMA) 控制器” 章节。

寄存器 31-13: DCHxDSIZ: DMA 通道 x 目标大小寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSIZ<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHDSIZ<7:0>							

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 CHDSIZ<15:0>: 通道目标大小位<sup>(1)</sup>

1111111111111111 = 目标大小为 65,535 字节

•

•

•

0000000000000010 = 目标大小为 2 字节

0000000000000001 = 目标大小为 1 字节

0000000000000000 = 目标大小为 65,536 字节

注: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

寄存器 31-14: DCHxSPTR: DMA 通道 x 源指针寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHSPTR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHSPTR<7:0>							

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

bit 31-16 未实现: 读为 0

bit 15-0 CHSPTR<15:0>: 通道源指针位 <sup>(1)</sup>

1111111111111111 = 指向源的字节 65,535

•

•

•

0000000000000001 = 指向源的字节 1

0000000000000000 = 指向源的字节 0

注 1: 处于模式检测模式时, 该寄存器会在模式检测时复位。



寄存器 31-15: DCHxDPTR: DMA 通道 x 目标指针寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHDPTR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHDPTR<7:0>							

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 CHDPTR&lt;15:0&gt;: 通道目标指针位

1111111111111111 = 指向目标的字节 65,535

•

•

•

0000000000000001 = 指向目标的字节 1

0000000000000000 = 指向目标的字节 0

注: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

寄存器 31-16: DCHxCSIZ: DMA 通道 x 单元大小寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHCSIZ<15:8>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHCSIZ<7:0>							

图注:

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

bit 31-16 未实现: 读为 0

bit 15-0 CHCSIZ<15:0>: 通道单元大小位

1111111111111111 = 在发生事件时传输 65,535 字节

•

•

•

0000000000000010 = 在发生事件时传输 2 字节

0000000000000001 = 在发生事件时传输 1 字节

0000000000000000 = 在发生事件时传输 65,536 字节

注: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

寄存器 31-17: DCHxCPTR: DMA 通道 x 单元指针寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHCPTR<15:8>							
7:0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
	CHCPTR<7:0>							

## 图注:

R = 可读位

W = 可写位

U = 未实现位, 读为 0

-n = POR 时的值

1 = 置 1

0 = 清零

x = 未知

bit 31-16 未实现: 读为 0

bit 15-0 CHCPTR<15:0>: 通道单元进度指针位<sup>(2)</sup>

1111111111111111 = 自上一个事件以来已传输了 65,535 字节

•

•

•

0000000000000001 = 自上一个事件以来已传输了 1 字节

0000000000000000 = 自上一个事件以来已传输了 0 字节

注 1: 不是所有器件都有该寄存器中的所有位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

2: 处于模式检测模式时, 该寄存器会在模式检测时复位。

# PIC32 系列参考手册

寄存器 31-18: DCHxDAT: DMA 通道 x 模式数据寄存器

位范围	Bit 31/23/15/7	Bit 30/22/14/6	Bit 29/21/13/5	Bit 28/20/12/4	Bit 27/19/11/3	Bit 26/18/10/2	Bit 25/17/9/1	Bit 24/16/8/0
31:24	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
23:16	U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
	—	—	—	—	—	—	—	—
15:8	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHPDAT<15:8> <sup>(1)</sup>							
7:0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	CHPDAT<7:0> <sup>(1)</sup>							

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

bit 31-16 未实现: 读为 0

bit 15-0 **CHPDAT<15:0>**: 通道数据寄存器位 <sup>(1)</sup>

模式终止模式:

要用于进行匹配的数据必须存储在该寄存器中, 以允许在发生匹配时终止。

所有其他模式:

未使用。

**注 1:** 不是所有器件都有这些位。关于可用性, 请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

### 31.3 工作模式

DMA 模块提供以下工作模式：

- 基本传输模式
- 模式匹配终止模式
- 通道链接模式
- 通道自动使能模式
- 特殊功能模块（Special Function Module, SFM）模式：LFSR CRC 和 IP 头校验和

这些工作模式不是互斥的，可以同时工作。例如，DMA 控制器可以使用链接的通道执行 CRC 计算，并在发生模式匹配时终止传输。

在说明 DMA 控制器的各种操作模式时，会用到以下术语：

- **事件**：可以启动或中止 DMA 传输的任何系统事件
- **事务**：单字传输（最多 4 字节），由读操作和写操作组成
- **单元传输**：在 DMA 通道启动传输时传输的字节数（由 DCHxCSIZ 寄存器指定），之后通道会等待另一个事件。单元传输由一个或多个事务组成。
- **数据块传输**：定义为在通道使能时传输的字节数。字节数对应于 DCHxSSIZ 或 DCHxDSIZ 的较大者。数据块传输由一个或多个单元传输组成。

**注：** 为避免在具有 L1 高速缓存器件上发生高速缓存一致性问题，DMA 模块访问的所有缓冲区必须位于 KSEG1 和 / 或 KSEG3（不可高速缓存）区段。

#### 31.3.1 基本传输模式工作原理

DMA 通道可以在无需 CPU 干预的情况下将数据从源传输到目标。通道源起始地址寄存器（DCHxSSA）用于定义源的物理起始地址。通道目标起始地址寄存器（DCHxDSA）用于定义目标的物理起始地址。源和目标可使用 DCHxSSIZ 和 DCHxDSIZ 寄存器独立配置。

单元传输通过以下两种方式之一启动：

- 软件可以通过将通道 CFORCE（DCHxECON<7>）位置 1 来启动传输
- 器件上发生与 CHSIRQ 中断匹配的中断事件，并且 SIRQEN（DCHxECON<4>）= 1。用户可以选择器件上的任意中断来启动 DMA 传输

在启动 DMA 传输（发生事件）时，它将传输 DCHxCSIZ（单元传输）个字节。通道将保持使能，直到 DMA 通道传输的字节数达到 DCHxSSIZ 和 DCHxDSIZ 中较大者对应的字节数（即，数据块传输完成）。如果 DCHxCSIZ 大于 DCHxSSIZ 和 DCHxDSIZ 中的较大者，则将传输 DCHxSSIZ 和 DCHxDSIZ 较大者对应的字节数。当通道被禁止时，将禁止进一步的传输，直到通道重新使能为止（CHEN 设置为 1）。

每个通道会使用指针 DCHxSPTR 和 DCHxDPTR 来跟踪从源和目标传输的字数。当源或目标指针达到一半位置（DCHxSSIZ/2 或 DCHxDSIZ/2）时，或者源或目标计数器达到结束位置时，将会产生中断。这些中断分别为 CHSHIF（DCHxINT<6>）、CHDHIF（DCHxINT<4>）、CHSDIF（DCHxINT<7>）或 CHDDIF（DCHxINT<5>）。

DMA 传输请求可以通过以下事件复位：

- 写入 CABORT（DCHxECON<6>）位，如 31.4.6 “复位通道”中所述
- 在通道自动使能模式位 CHAEN（DCHxCON<4>）未置 1 的情况下，发生了模式匹配（如果已按照第 31.3.2 节“模式匹配终止模式”中所述使能了模式匹配）
- 器件上发生与 CHAIRQ <7:0>（DCHxECON<23:16>）位中断匹配的中断事件（如果已通过 AIRQEN（DCHxECON<3>）位允许）
- 检测到地址错误

- 单元传输完成
- 数据块传输完成且通道自动使能模式（CHAEN）位未置 1

在发生通道中止中断时，通道传输中止中断标志位 CHTAIF（DCHxINT<1>）会置 1。通过它，用户可以检测中止的 DMA 传输，并进行恢复。在传输被中止时，将会完成当前正在进行的任意事务。

源指针和目标指针会随传输进度而更新。这些指针是只读的。这些指针会在以下条件下复位：

- 如果通道源地址（DCHxSSA）被更新，则源指针（DCHxSPTR）将复位
- 目标地址（DCHxDSA）发生类似更新时，将导致目标指针（DCHxDPTR）复位
- 通过写入 CABORT（DCHxECON<6>）位中止通道传输

**注：** 关于通道事件行为的详细信息，请参见表 31-2。

## 例 31-1: 基本传输模式下 DMA 通道初始化的代码示例

```
/* This code example illustrates the DMA channel 0 configuration for a data transfer. */
IEC1CLR=0x00010000;           // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;           // clear existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;          // enable the DMA controller
DCH0CON=0x3;                   // channel off, priority 3, no chaining
DCH0ECON=0;                    // no start or stop IRQs, no pattern match

// program the transfer
DCH0SSA=VirtToPhys(flashBuff); // transfer source physical address
DCH0DSA=VirtToPhys(ramBuff);    // transfer destination physical address
DCH0SSIZ=200;                   // source size 200 bytes
DCH0DSIZ=200;                   // destination size 200 bytes
DCH0CSIZ=200;                   // 200 bytes transferred per event
DCH0INTCLR=0x00ff00ff;          // clear existing events, disable all interrupts
DCH0CONSET=0x80;                // turn channel on

// initiate a transfer
DCH0ECONSET=0x00000080;         // set CFORCE to 1

// do something else
// poll to see that the transfer was done

while(TRUE)
{
    register int pollCnt;        // use a poll counter.
                                // continuously polling the DMA controller in a tight
                                // loop would affect the performance of the DMA transfer

    int dmaFlags=DCH0INT;
    if( (dmaFlags&0xb)
    {
        // one of CHERIF (DCHxINT<0>), CHTAIF (DCHxINT<1>)
        // or CHBCIF (DCHxINT<3>) flags set
        break;                  // transfer completed
    }
    pollCnt=100;                 // use an adjusted value here
    while(pollCnt--);            // wait before reading again the DMA controller
}
// check the transfer completion result
```

### 31.3.1.1 中断和指针更新

在每个事务完成之后，源和目标指针会更新。也会在此时设置或清除中断。如果在事务期间指针超过中点，中断会相应地进行更新。

指针会在发生以下事件时复位：

- 任何器件复位时
- DMA 关闭（ON（DMACON<15>）位为 0）时
- 数据块传输完成，无论 CHAEN（DCHxCON<4>）位的状态如何
- 模式匹配终止传输，无论 CHAEN（DCHxCON<4>）位的状态如何
- 写入 CABORT（DCHxECON<6>）位标志
- 源或目标起始地址被更新

### 31.3.2 模式匹配终止模式

通过模式匹配终止模式，用户可以在事务期间写入的数据与特定模式（由 DCHxDAT 寄存器定义）匹配时结束传输。对待模式匹配的方式与数据块传输完成相同，这种情况下 CHBCIF 位（DCHxINT<3>）会置 1，CHEN 位（DCHxCON<7>）会清零。

该功能在需要使用可变数据大小的应用中很有用，并且可以方便 DMA 通道的设置。UART 模块是可以有效使用该功能的一个很好示例。

假设系统具有一系列的消息，这些消息定期发送到外部主机，消息最大容量为 86 个字符，用户可以在通道上设置以下参数：

- DCHxSSIZ 设置为 87 字节 —— 如果发生意外情况，CPU 程序将在缓冲区溢出时产生中断，并执行相应的操作
- DCHxDSIZ 设置为 1 字节
- 目标地址设置为 UART TXREG
- DCHxDAT 设置为 0x00，这将在任何字节轨道中检测到 NULL 字符时停止传输
- CHSIRQ<7:0>（DCHxECON<15:8>）位设置为 UART “发送缓冲区为空” IRQ
- SIRQEN（DCHxECON<4>）位置 1，允许通道响应启动中断事件
- 起始地址设置为要传输消息的起始地址
- 使能通道，CHEN（DCHxCON<7>）= 1
- 然后，用户将通过 CFORCE（DCHxECON<7>）位强制开始单元传输，UART 将传输第一个字节
- 每次 UART 发送一个字节时，传输缓冲区空中断将启动从源向 UART 传输下一个字节
- 当 DMA 通道在通道的任何字节轨道中检测到 NULL 字符时，事务将完成，通道将被禁止

模式匹配与源数据的字节轨道无关。如果源缓冲区中的任意字节与 DCHxDAT 匹配，则会检测到模式匹配事件。事务将完成，从源读取的数据将写入目标。

#### 31.3.2.1 模式匹配忽略模式

在具有 CHPATLEN 位的器件中，模式可为 8 位或 16 位宽。该模式长度由 DCHxCON 寄存器中的 CHPATLEN 位决定。如果 CHPATLEN 位置 1，可使用模式匹配忽略模式。如果使能模式忽略字节位（CHPIGNEN）置 1，且通道寄存器数据位（CHPIGN<7:0>）的值满足的话，则在单元传输期间当尝试寻找终止模式时，传输的数据将作为“无关”对待。这种情况的一个示例就是，在行尾和换行之间出现空格符。如果行尾表示为“X”、换行表示为“Y”且 CHPIGN<7:0> 位设置为“\_”，则在 DMA 单元传输期间传输“x\_y”时，将检测到模式匹配终止。因为当检测到模式匹配时，SFM 将忽略之间的零。

**例 31-2: 模式匹配传输模式下 DMA 通道初始化的代码示例**

```
/* This code example illustrates the DMA channel 0 configuration for data transfer with
pattern match enabled. Transfer from the UART1 a <CR> ended string, at most 200 characters
long */

IEC1CLR=0x00010000;           // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;           // clear any existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;         // enable the DMA controller
DCH0CON=0x03;                  // channel off, priority 3, no chaining

DCH0ECON=(27 <<8) | 0x30;      // start irq is UART1 RX, pattern match enabled
DCH0DAT='\r';                  // pattern value, carriage return

                                // program the transfer
DCH0SSA=VirtToPhys(&U1RXREG);  // transfer source physical address
DCH0DSA=VirtToPhys(ramBuff);    // transfer destination physical address
DCH0SSIZ=1;                    // source size is 1 byte
DCH0DSIZ=200;                  // destination size at most 200 bytes
DCH0CSIZ=1;                    // one byte per UART transfer request

DCH0INTCLR=0x00ff00ff;         // clear existing events, disable all interrupts
DCH0INTSET=0x00090000;         // enable Block Complete and error interrupts

IPC9CLR=0x0000001f;           // clear the DMA channel 0 priority and sub-priority
IPC9SET=0x00000016;           // set IPL 5, sub-priority 2
IEC1SET=0x00010000;           // enable DMA channel 0 interrupt

DCH0CONSET=0x80;               // turn channel on
```

## 31.3.3 通道链接模式

通道链接是对 DMA 通道操作的增强。通道（从通道）可以与邻接通道（主通道）进行链接。在主通道的数据块传输完成（即，CHBCIF（DCHxINT<3>）位置 1）时，从通道会被使能。

此时，从通道上的任何事件都可以启动单元传输。如果通道具有待处理事件，则单元传输会立即开始。

主通道将以正常方式设置其中断标志位 CHBCIF（DCHxINT<3>），并且不会知道从通道的“链”状态。如果 CHSDIE/CHDDIE/CHBCIE（DCHxINT<23/21/19>）位中有一个位置 1，主通道仍然能够在 DMA 传输结束时产生中断。

在通道自然优先级中，通道 0 优先级最高。在使能通道链接（CHCHN（DCHxCON<5>）= 1）的情况下，可以使能特定通道的较高或较低优先级通道（通过 CHCHNS（DCHxCON<8>）位进行选择）。

DMA 模块具有在通道被禁止时允许事件的功能，该功能通过 CHAED（DCHxCON<6>）位使能。该位在通道链接模式下特别有用，在该模式下，从通道需要在通道被主通道使能时立即准备好启动传输。

以下示例说明了通道链接可以产生作用的情形：

1. 从一个外设（例如，通过 DMA 通道 0 以 9600 波特率从 UART1 到 SRAM）向另一个外设（例如，通过 DMA 通道 1 以 19200 波特率从 SRAM 到 UART2）传输数据。

在该示例中，CHAED 在两个通道中均置 1；在发送最后一个字节之后，UART2 会将通道 1 上的事件检测位 CHEDET（DCHxCON<2>）置 1。在通道 0 完成传输时，通道 1 会立即使能，并立即传输数据。

2. ADC 模块向一个缓冲区传输数据（与通道 0 连接）。

当目标缓冲区 0 变满（数据块传输完成）时，通道 1 会被使能，后续的转换数据将传输到缓冲区 1。在此例中，CHAED 不会被使能。如果它被使能，则通道 1 会再次传输由通道 0 传输的最后一个字（因为 ADC 中断事件会将两个通道中的事件检测标志 CHEDET 都置 1）。



例 31-3: 通道链接模式下 DMA 通道初始化的代码示例

```

/* This code example illustrates the DMA channel 0 configuration for data transfer with pattern
match enabled. DMA channel 0 transfer from the UART1 to a RAM buffer while DMA channel 1
transfers data from the RAM buffer to UART2. Transferred strings are at most 200 characters
long. Transfer on UART2 starts as soon as the UART1 transfer is completed. */

unsigned char myBuff<200>;    // transfer buffer

IEC1CLR=0x00010000;          // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;          // clear any existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;         // enable the DMA controller

DCH0CON=0x3;                  // channel 0 off, priority 3, no chaining
DCH1CON=0x62;                 // channel 1 off, priority 2
                                // chain to higher priority
                                // (channel 0), enable events detection while disabled

DCH0ECON=(27 <<8) | 0x30;     // start IRQ is UART1 RX, pattern enabled
DCH1ECON=(42 <<8) | 0x30;     // start IRQ is UART1 TX, pattern enabled

DCH0DAT=DCH1DAT='\r';         // pattern value, carriage return

                                // program channel 0 transfer
DCH0SSA=VirtToPhys(&U1RXREG); // transfer source physical address
DCH0DSA=VirtToPhys(myBuff);   // transfer destination physical address
DCH0SSIZ=1;                   // source size is 1 byte
DCH0DSIZ=200;                 // dst size at most 200 bytes
DCH0CSIZ=1;                   // one byte per UART transfer request

                                // program channel 1 transfer
DCH1SSA=VirtToPhys(myBuff);   // transfer source physical address
DCH1DSA=VirtToPhys(&U2TXREG); // transfer destination physical address
DCH1SSIZ=200;                 // source size at most 200 bytes
DCH1DSIZ=0;                   // dst size is 1 byte
DCH1CSIZ=1;                   // one byte per UART transfer request

DCH0INTCLR=0x00ff00ff;        // DMA0: clear events, disable interrupts
DCH1INTCLR=0x00ff00ff;        // DMA1: clear events, disable interrupts
DCH1INTSET=0x00090000;        // DMA1: enable Block Complete and error interrupts

IPC9CLR=0x00001f1f;           // clear the DMA channels 0 and 1 priority and
                                // sub-priority
IPC9SET=0x00000b16;           // set IPL 5, sub-priority 2 for DMA channel 0
                                // set IPL 2, sub-priority 3 for DMA channel 1
IEC1SET=0x00020000;           // enable DMA channel 1 interrupt

DCH0CONSET=0x80;              //turn channel on

```

## 31.3.4 通道自动使能模式

通道自动使能可以用于使通道保持活动状态，即使数据块传输已完成或发生模式匹配。这使用户可以不必在每次数据块传输完成时重新使能通道。要使用该模式，用户需要配置通道：先将 CHAEN (DCHxCON<4>) 位置 1，然后再使能通道（即将 CHEN (DCHxCON<7>) 位置 1）。通道将按正常模式工作，只是正常传输终止不会导致通道被禁止。

正常数据块传输完成定义为：

- 数据块传输完成
- 检测到模式匹配

如前面所述，通道指针会发生复位。该模式对于需要重复进行模式匹配的应用很有用。

**注：** CHAEN 位可以防止通道在使能之后自动被禁止。通道仍然必须通过软件使能。

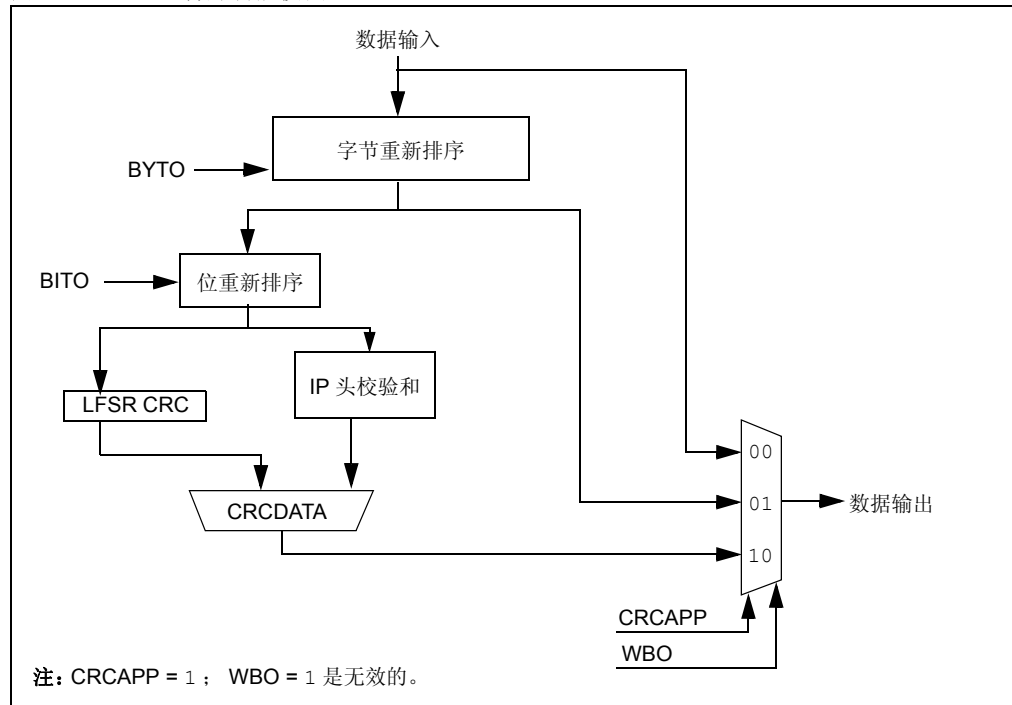
## 31.3.5 特殊功能模块 (SFM) 模式

DMA 模块具有一个集成的特殊功能模块 (SFM)，由所有通道共用。

如图 31-3 所示，SFM 具有以下功能块：

- 线性反馈移位寄存器 (Linear Feedback Shift Register, LFSR) CRC
- IP 头校验和
- 字节重新排序
- 位重新排序

图 31-3： 特殊功能模块 (SFM)



根据不同的器件型号，SFM 是高度可配置的 16 位或 32 位 CRC 发生器。SFM 可以分配给任意可用 DMA 通道，方法是相应地设置 CRCCH<2:0> (DCRCCON<2:0>) 位。SFM 通过将 CRCEN (DCRCCON<7>) 位置 1 来使能。

通过使用 WBO 位，可以选择对源数据进行字节重新排序。然后，可以选择根据 DCRCCON 寄存器中 CRCTYP (DCRCCON<15>) 位的设置，将数据传递给 LFSR CRC 或 IP 头校验和功能块，如图 31-3 中所示。

此外, SFM 可以修改与 SFM 关联的 DMA 通道的行为。通道的行为通过 CRCAPP (DCRCCON<6>) 位进行选择, 产生以下两种模式:

- 后台模式: CRC 在后台进行计算, 并保持正常的 DMA 行为 (见第 31.3.5.1 节 “CRC 后台模式 (CRCAPP = 0)”)。
- 追加模式: 从源读取的数据写入目标中, 但 CRC 数据在 CRC 数据寄存器中累加。在数据块传输完成时, 累加的 CRC 写入由 DCHxDSA 寄存器指定的单元中 (见第 31.3.5.2 节 “CRC 追加模式 (CRCAPP = 1)”)。

数据写入目标的顺序可以使用 WBO 位 (DCRCCON<27>) 进行选择。如果 WBO 位清零, 则数据写入目标时保持不变。如果 WBO 位置 1, 则数据写入目标时, 将按照 CRC 字节顺序选择位 BYTO<1:0> (DCRCCON<29:28>) 重新排序。

**注:** 不是所有器件都有该功能。关于可用性, 请参见具体数据手册的 “直接存储器访问 (DMA) 控制器” 章节。

SFM 发生器可以通过在使能通道之前写入 DCRCDATA 寄存器来设置种子值。

请注意, 处于 IP 头校验和模式 (CRCTYP (DCRCCON<15>) = 1) 时, 数据以二进制反码的形式写入和读回, 因为这是校验和的当前值。

DCRCDATA 中的 CRC 值可以在 CRC 发生期间的任意时刻读取, 但只有在传输完成之后它才有效。

- 注**
- 1: 如果在进行 CRC 计算时终止 DMA 传输, 则 DMA 通道应在下次 CRC 计算开始前复位。此外, 可以配置同一个通道或其他未使用的通道来传输两个或更多字节。应开始传输并使其完成。然后, CRC 模块就绪, 可用于下次 CRC 计算。
  - 2: 如果在进行 CRC 计算时禁止 DMA 通道 (CHEN (DCHxCON<7>) = 0), 则不会更新 DCRCDATA 寄存器的值。可以配置同一个通道或其他未使用的通道来传输两个或更多字节。应开始传输并使其完成。当传输完成时, 在停止传输前针对已处理字节数的 DCRCDATA 值将是正确的。DMA 地址寄存器可用于确定当前 CRC 值的地址范围。

### 31.3.5.1 CRC 后台模式 (CRCAPP = 0)

在该模式下, 将保持 DMA 通道的行为。DMA 会从源读取数据, 将数据传递经过 CRC 模块, 并将它写入目标。数据写入目标时将遵从 WBO 选择。在该模式下, 计算得到的 CRC 在数据块传输结束时留在 DCRCDATA 寄存器中。

该模式可以用于在数据从源地址传送到目标地址时计算 CRC。数据源可以为存储器缓冲区或外设中的 FIFO。类似地, 目标可以为存储器缓冲区或 FIFO。当数据传输完成时, 用户可以读取计算得到的 CRC 值, 并将它追加到传送数据末尾或用于校验接收到的 CRC 数据。

后台模式可能会长时间占用 CRC 模块。例如, 当分配给 UART 数据流时, 在 UART 数据流完成之前, 其他通道无法使用 SFM。

## 例 31-4: 后台模式下 DMA LFSR CRC 计算的代码示例

```
/* This code example illustrates a DMA calculation using the CRC background mode. Data is
transferred from a 200 bytes Flash buffer to a RAM buffer and the CRC is calculated while the
transfer takes place. */

unsigned int blockCrc;           // CRC of the Flash block

IEC1CLR=0x00010000;           // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;           // clear any existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;          // enable the DMA controller

DCRCDATA=0xffff;               // seed the CRC generator
DCRCXOR=0x1021;                // Use the standard CCITT CRC 16 polynomial: X^16+X^12+X^5+1
DCRCCON=0x0f80;                // CRC enabled, polynomial length 16, background mode
                                // CRC attached to the DMA channel 0.

DCH0CON=0x03;                  // channel off, priority 3, no chaining
DCH0ECON=0;                    // no start irqs, no match enabled

                                // program channel transfer
DCH0SSA=VirtToPhys(flashBuff); // transfer source physical address
DCH0DSA=VirtToPhys(ramBuff);    // transfer destination physical address
DCH0SSIZ=200;                   // source size
DCH0DSIZ=200;                   // destination size
DCH0CSIZ=200;                   // 200 bytes per event

DCH0INTCLR=0x00ff00ff;         // DMA0: clear events, disable interrupts

DCH0CONSET=0x80;                // channel 0 on

                                // initiate a transfer
DCH0ECONSET=0x00000080;         // set CFORCE to 1

                                // do something else while the transfer takes place
                                // poll to see that the transfer was done

BOOL error=FALSE;
while(TRUE)
{
    register int pollCnt;        // don't poll in a tight loop
    int dmaFlags=DCH0INT;
    if( (dmaFlags& 0x3)
    {
        error=TRUE;              // CHERIF (DCHxINT<0>) or CHTAIF (DCHxINT<1>) set
        break;                   // error or aborted...
    }
    else if (dmaFlags&0x8)
    {
        break;                   // CHBCIF (DCHxINT<3>) set
        // transfer completed normally
    }
    pollCnt=100;                 // use an adjusted value here
    while(pollCnt--);            // wait before polling again
}

if(!error)
{
    blockCrc=DCRCDATA;           // read the CRC of the transferred Flash block
}
else
{
    // process error
}
```

## 31.3.5.2 CRC 追加模式 (CRCAPP = 1)

在该模式下，DMA 只会将源数据送到 CRC 模块；它不会将源数据写入目标地址中。但是，在数据块传输完成或发生模式匹配时，DMA 会将 CRC 值写入目标地址。

以下使用说明适用于 CRC 追加模式：

- 在确定数据块传输是否完成时只需要查看源缓冲区，目标地址 (DCHxDSA) 仅用作写入所生成 CRC 值的单元。
- 目标大小 (DCHxDSIZ) 最大可以为 4。
  - 如果 DCHxDSIZ 大于 4，则仅写入 4 个字节
  - 如果 DCHxDSIZ 小于 4，则仅写入 CRC 的 DCHxDSIZ 个字节
  - PLEN<4:0> 位对于写入的 CRC 字节数或位数没有影响
- 写入之后，通道会被禁止。
- 任何中止（即，中止 IRQ 置为有效）都会阻止写入 CRC 值。
- 如果 WBO 设置为 0，则在追加模式下不支持重新排序。

**例 31-5: 追加模式下 CRC 计算的代码示例**

```
/* This code example illustrates a DMA calculation using the CRC append mode. The CRC of a 256
bytes Flash buffer is calculated without performing any data transfer. As soon as the CRC
calculation is completed the CRC value of the Flash buffer is available in a local variable for
further use. */

unsigned int blockCrc;                // CRC of the Flash block

IEC1CLR=0x00010000;                 // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;                 // clear any existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;                // enable the DMA controller

DCRCDATA=0xffff;                     // seed the CRC generator
DCRCXOR=0x1021;                      // Use the standard CCITT CRC 16 polynomial: X^16+X^12+X^5+1
DCRCCON=0x0fc0;                     // CRC enabled, polynomial length 16, append mode
                                   // CRC attached to the DMA channel 0.

DCH0CON=0x03;                        // channel off, priority 3, no chaining
DCH0ECON=0;                          // no start irqs, no match enabled

                                   // program channel transfer
DCH0SSA=VirtToPhys(flashBuff);       // transfer source physical address
DCH0DSA=VirtToPhys(&blockCrc);       // transfer destination physical address
DCH0SSIZ=200;                         // source size
DCH0DSIZ=200;                         // dst size
DCH0CSIZ=200;                         // 200 bytes transferred per event

DCH0INTCLR=0x00ff00ff;               // DMA0: clear events, disable interrupts
DCH1INTCLR=0x00ff00ff;               // DMA1: clear events, disable interrupts

DCH0CONSET=0x80;                     // channel 0 on

                                   // initiate a transfer
DCH0ECONSET=0x00000080;               // set CFORCE to 1

                                   // do something else while the CRC calculation takes place
                                   // poll to see that the transfer was done

BOOL error=FALSE;
while(TRUE)
{
    register int pollCnt;              // don't poll in a tight loop
    int dmaFlags=DCH0INT;
    if( (dmaFlags& 0x3)
    {
        error=TRUE;                   // CHERIF (DCHxINT<0>) or CHTAIF (DCHxINT<1>) set
        break;                         // error or aborted...
    }
    else if (dmaFlags&0x8)
    {
        break;                         // CHBCIF (DCHxINT<3>) set
        // transfer completed normally
    }
    pollCnt=100;                       // use an adjusted value here
    while(pollCnt--);                 // wait before polling again
}

if(error)
{
    // process error
}
```

## 31.3.5.3 数据顺序

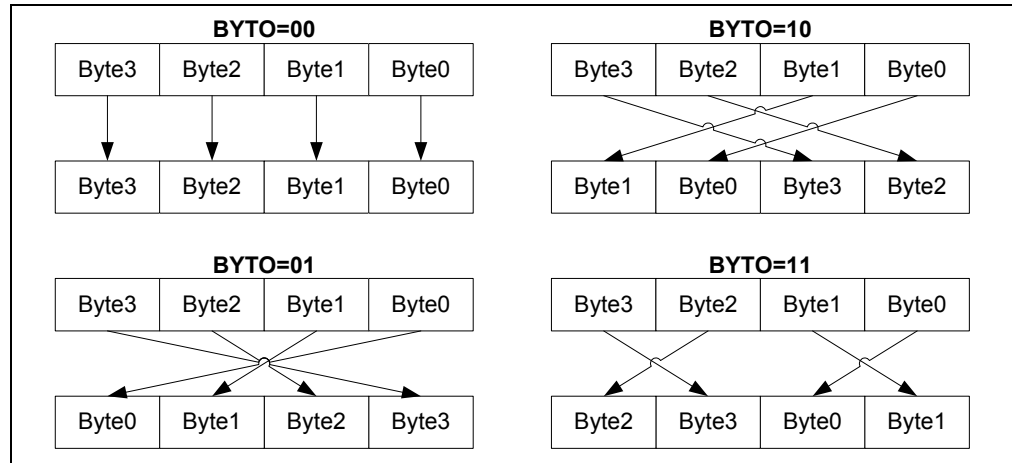
从源读取的数据可以进行重新排序，从而支持不同的源数据字节顺序。在  $WBO = 1$  时，将向通道目标中写入重新排序后的源数据。在  $WBO = 0$  时，将向目标中写入保持不变的源数据。

**BYTO<1:0>** 位控制由模块处理的数据的字节顺序。图 31-4 显示了不同的字节顺序设置，以及对于数据读操作的影响。**BYTO<1:0>** 值 01 用于对字中的字节进行重新排序。而 **BYTO<1:0>** 值 10 和 11 用于对半字中的字节进行重新排序。

数据在读取时进行重新排序，注意到这一点非常重要。这意味着未字对齐的数据可能无法正确进行重新排序。

使用 SFM 的 LFSR CRC 模式或 IP 头校验和模式时，可以通过使用 **BITO** (**DCRCCON<24>**) 位更改位顺序 (**MSB** 或 **LSB**)。

图 31-4: **BYTO** 值的字节顺序



## 31.3.5.4 LFSR CRC

CRC 发生器需要一个系统时钟的时间来处理从源读取的每个数据字节。这意味着，如果从源读取 32 位的数据，则 CRC 生成将需要 4 个系统时钟来处理数据。

当 **CRYTYP** 位清零时，SFM 设置为 LFSR CRC 模式，并会计算 LFSR CRC。

**注：** 不是所有器件都有该功能。关于可用性，请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

CRC 模块的实现可通过软件进行配置。多项式的各项及其长度可以分别使用 **DCRCXOR<31:0>** 位和 **PLEN<4:0>** (**DCRCCON<12:8>**) 位设定。

例 31-6 和例 31-7 给出了 16 位和 32 位 CRC 的多项式。位值中的“x”作为“无关”对待，因为它们永远进行异或运算。

### 例 31-6: 16 位 CRC 多项式

$$x^{16} E x^{12} E x^5 E 1$$

**PLEN<4:0>** = 'b01111

**DCRCXOR<15:0>** = 'bx001 0000 0010 000x

### 例 31-7: 32 位 CRC 多项式

$$x^{32} E x^{26} E x^{23} E x^{22} E x^{16} E x^{12} E x^{11} E x^{10} E x^8 E x^7 E x^5 E x^4 E x^2 E x E 1$$

**PLEN<4:0>** = 'b11111

**DCRCXOR<31:0>** = 'bx000 0100 1100 0001 0001 1101 1011 011x

CRC 发生器中的  $PLEN<4:0>$  ( $DCRCCON<12:8>$ ) 位用于选择哪个位用作 CRC 的反馈点。对于 16 位 CRC 示例，如果  $PLEN<4:0> = 00110$ ，则移位寄存器的 bit 6 送到 CRCXOR 寄存器中置 1 的所有位的异或门。

CRCXOR 反馈点使用 DCRCXOR 寄存器指定。将 DCRCXOR 寄存器中的第 N 位置 1 时，会使 CRC 移位寄存器第 N 位的输入与 CRC 移位寄存器第 ( $PLEN + 1$ ) 位进行异或运算。CRC 发生器的 bit 0 和 bit 15 总是进行异或运算。

### 31.3.5.5 计算 IP 头校验和

当 CRCTYP ( $DCRCCON<15>$ ) 位置 1 时，SFM 会计算 IP 头校验和。使用以下过程来计算 IP 头校验和：

1. 配置通道，使之指向 IP 头。
2. 配置 CRCCON 来使能 SFM，并选择所使用的通道。
3. 将 CRCTYP 位置 1，这会选择 IP 头校验和。
4. 将 DCRCDATA 设置为 0000。
5. 开始传输。
6. 在传输完成时，从 DCRCDATA 寄存器中读取数据。

**注：** 不是所有器件都有该功能。关于可用性，请参见具体数据手册的“直接存储器访问 (DMA) 控制器”章节。

## 31.4 通道控制

### 31.4.1 通道使能

每个通道都具有使能位 CHEN (DCHxCON<7>)，它可以用于使能通道或禁止出问题的通道。当该位置 1 时，DMA 控制器会处理通道传输请求。

当 CHEN 位清零时，通道状态会被保留（这使得可以在传输开始之后暂停通道）。

在以下条件下，硬件会将 CHEN 位清零：

- 数据块传输完成，指向源或目标中较大者的指针与大小匹配（仅当 CHAEN (DCHxCON<4>) 位清零时）
- 在模式匹配模式下发生模式匹配（仅当 CHAEN 位清零时）
- 发生中止中断
- 用户写入 CABORT (DCHxECON<6>) 位

### 31.4.2 通道传输行为

使能通道之后 (CHEN = 1)，任何启动单元传输的事件都会传输 CHCSIZ<15:0> (DCHxCSIZ<15:0>) 个数据字节。这需要一个或多个事务。在单元传输完成时，通道将恢复为非活动状态，并等待另一个通道启动事件，之后才会启动另一个单元传输。

当传输的字节数大于 CHSSIZ<15:0> (DCHxSSIZ<15:0>) 或 CHDSIZ<15:0> (DCHxDSIZ<15:0>) 对应的字节数时，数据块传输完成，通道传输将暂停，并且通道会被禁止（即，硬件将 CHEN 设置为 0，指针会复位）。

#### 31.4.2.1 通道事件传输启动

指定通道传输可以通过以下事件启动：

- 写入 CFORCE (DCHxECON<7>) 位
- 在通过 SIRQEN (DCHxECON<4>) 位允许的情况下，发生与 CHSIRQ<7:0> (DCHxECON<15:8>) 值匹配的中断

如果使能了通道 (CHEN = 1)，或者如果“禁止时允许事件”位置 1（即 CHAED (DCHxCON<6>) = 1），则会登记通道事件。

#### 31.4.2.2 通道事件传输终止

在以下任意情况下，通道传输会被终止：

- 按照第 31.4.4 节“通道中止中断”中所述中止传输
- 单元传输完成（传输了 CHCSIZ<15:0> (DCHxCSIZ<15:0>) 个字节）
- DMA 传输了 CHSSIZ<15:0> 或 CHDSIZ<15:0> 较大者对应的字节数（数据块传输完成），通道在硬件中被禁止，只有用户软件重新使能通道时，通道才会响应通道事件
- 发生模式匹配（如使能）
- 在通过 AIRQEN (DCHxECON<3>) 位允许中止中断的情况下，发生中止中断 CHAIRQ<7:0> (DCHxECON<23:16>)
- 发生地址错误

一个可以说明如何使用中止中断的示例就是从 UART 通道向存储器传输数据。UART 接收数据可用中断可以用于启动传输，而 UART 错误中断可以中止传输。通过这种方式，每次通信通道上发生错误时（帧/奇偶错误，甚至是发生溢出），传输会被停止，用户代码在 ISR 中获得控制权（如果对于 DMA 控制器允许中止中断）。

表 31-2 中汇总了会受通道传输启动或终止影响的状态标志。如果使能通道 (CHEN = 1)，或者如果用户选择在通道被禁止时允许事件 (CHAED = 1)，则允许通道中止事件。



表 31-2: 通道事件行为

事件	说明和功能	受影响的寄存器
<b>启动传输的事件</b>		
系统中断与 CHSIRQ<7:0> 匹配 <sup>(1,2)</sup>	通道事件检测位将置 1。	CHEDET = 1
通道链接事件	如果尚未置 1，该事件将使能通道。如果事件检测位置 1，则会立即开始通道传输。	CHEN = 1
用户写入 CFORCE 位 <sup>(1)</sup>	通道事件检测位将置 1。	CHEDET = 1
<b>终止传输的事件</b>		
系统中断与 CHAIRQ<7:0> 匹配 <sup>(1,2)</sup>	通道事件检测位将复位，通道将被关闭。中止中断标志置 1。	CHEDET = 0 CHEN = 0 CHAIF = 1
模式匹配终止 <sup>(1)</sup>	当在事务中写入的任意数据字节与 CHPDAT 中的数据匹配时，会发生该事件。 通道事件检测位复位。 如果 CHAEN = 0，通道会被关闭。该事件视为数据块传输完成事件。 指针复位。	CHEDET = 0 CHEN = 0 CHBCIF = 1 CHSPTR = 0 CHDPTR = 0 CHCPTR = 0
单元传输完成	当传输的字节数达到 CHCSIZ 字节时，会发生该事件。传输事件检测位复位，通道保持使能，等待下一个事件。	CHEDET = 0 CHCCIF = 1
数据块传输完成	通道事件检测位复位。 如果 CHAEN = 0，通道会被关闭。该事件视为传输完成事件。 指针复位。	CHEDET = 0 CHEN = 0 CHBCIF = 1 CHSPTR = 0 CHDPTR = 0 CHCPTR = 0
用户写入 CABORT 位	通道被关闭，通道事件检测位复位。指针复位。	CHEDET = 0 CHEN = 0 CHSPTR = 0 CHDPTR = 0 CHCPTR = 0
检测到地址错误	通道被关闭，事件检测位复位。地址错误中断标志置 1。	CHEDET = 0 CHEN = 0 CHERIF = 1

- 注 1: 只有使能通道时，或者用户在通道被禁止时允许事件时（CHEN = 1 或 CHAED = 1），才允许事件。
- 2: DMA 控制器会维护它自己的一些标志，用于检测系统中的启动和中止中断请求（IRQ），并且这些标志完全独立于 INT 控制器 IESx/IFSx 标志。在触发启动或中止 IRQ 系统事件时，DMA 控制器内部逻辑可以自动检测它们，无需用户干预。

### 31.4.3 通道 IRQ 检测

DMA 控制器会维护它自己的一些标志，用于检测系统中的启动和中止 IRQ，并且这些标志完全独立于 INT 控制器和 IESx/IFSx 标志。在执行传输之前，不一定要使能相应的 IRQ，在 DMA 传输结束时，也不一定要清除它。

在触发启动或中止 IRQ 系统事件时，DMA 控制器内部逻辑可以自动检测它们，无需用户干预。

## 31.4.4 通道中止中断

通道可以选择在发生中断事件时中止单元传输。中断通过通道的中止 IRQ（CHAIQ<7:0>（DCHxECON<23:16>））进行选择。任一器件中断事件都可以导致通道中止。只有通过 AIRQEN（DCHxECON<3>）位使能时，才会发生中止。

如果发生这种情况（通常由于定时器超时或模块错误标志），通道的状态标志会通过将其 CHTAIF（DCHxINT<1>）位置 1 来指示出问题通道上的外部中止事件。源和目标指针不会复位，让用户可以从错误中进行恢复。

## 31.4.5 DMA 暂停

如果 SUSPEND（DMACON<12>）位置 1，则会立即暂停 DMA 事务。控制器将会完成当前的读操作或写操作。如果暂停是在事务的读操作部分中发生的，则事务将被暂停，写操作将被搁置。如果暂停是在事务的写操作部分中发生的，则会完成写操作，指针按正常情况进行更新。在 SUSPEND 位清零时，任何先前正在进行的事务将从退出点继续。

根据不同的器件型号，当通过将 SUSPEND 位置 1 而暂停 DMA 模块时，用户应用程序应通过查询 DMABUSY（DMACON<11>）位来确定在当前事务完成之后模块完全暂停的时间。

**注：** 不是所有器件都有 DMABUSY 位。关于可用性，请参见具体数据手册的“直接存储器访问（DMA）控制器”章节。

### 例 31-8: DMA 控制器暂停

```
/* This code example will suspend the DMA Controller. */

DMACONSET=0x00001000;          // suspend the DMA controller

while(!(DMACONbits.busy)); // wait for the transfer to be suspended

                                // let the CPU have complete control of the bus

DMACONCLR=0x00001000;          // clear the suspend mode and let the DMA
                                // operate normally. From now on, the CPU and
                                // DMA controller share the bus access
```

每个通道可以使用 CHEN（DCHxCON<7>）位暂停。如果 DMA 传输正在进行，并且 CHEN 位清零，则会完成通道上的当前事务，而后续事务则会被暂停。

根据不同的器件，当通过将 CHEN 位清零而暂停通道时，用户应用程序应通过查询 CHBUSY（DCHxCON<15>）位来确定在当前事务完成之后通道完全暂停的时间。

清零使能位（CHEN）不会影响通道指针或事务计数器。在通道暂停时，用户可以选择通过将 CHAED（DCHxCON<6>）位置 1 而继续接收事件（中止中断等）。

## 31.4.6 复位通道

每次器件复位时，通道逻辑都会复位。在写入通道标志位 CABORT（DCHxECON<6>）时，通道也会复位。这会清零通道使能位（CHEN = 0），清除源和目标指针，并复位事件检测器。当 CABORT 位置 1 时，将会在通道复位之前完成当前正在进行的事务（如果有），但所有剩余的事务将被中止。

只有在通道被禁止（CHEN = 0）时，用户才应修改通道寄存器。修改源和目标寄存器会使相应的指针寄存器（DCHxSPTR 或 DCHxDPTR）复位。

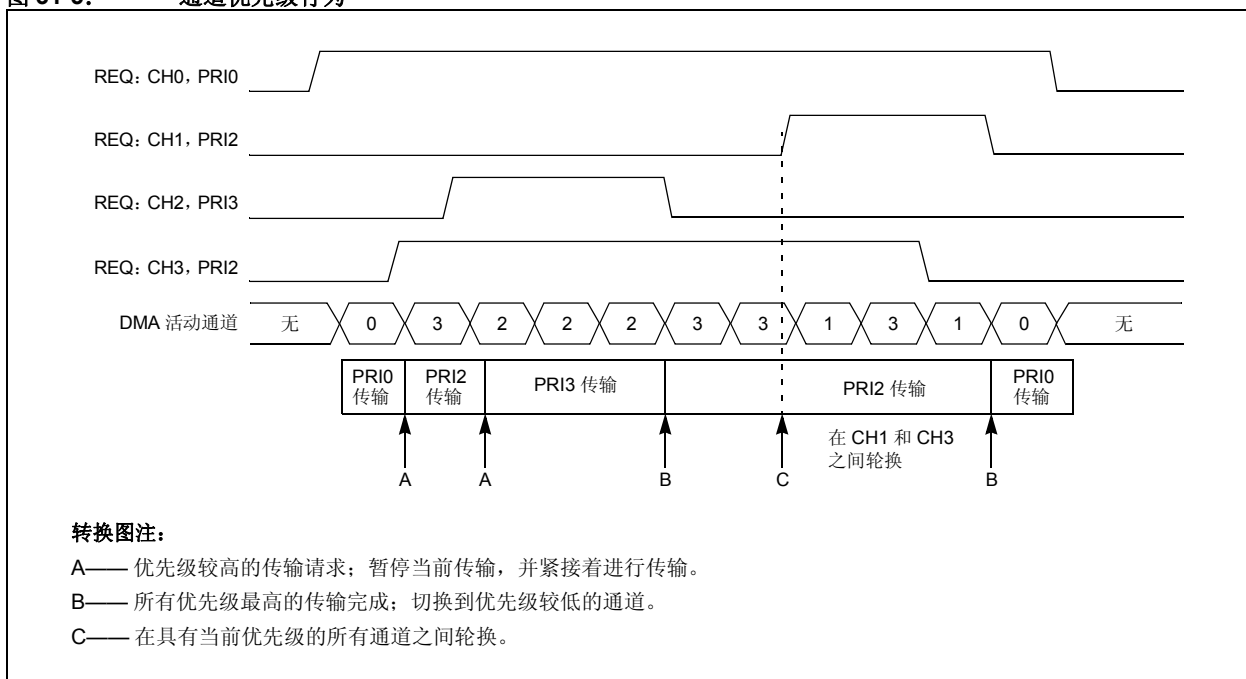
**注：** 只有在通道被禁止时，才能更改通道大小。

### 31.4.7 通道优先级和选择

DMA 控制器的每个通道都具有关联的自然优先级。通道 0 的自然优先级最高。通过  $\text{CHPRI}<1:0>$  位 ( $\text{DCHxCON}<1:0>$ ) 改变通道的优先级。这两位标识通道的优先级，值为零时优先级最低。如果不设置优先级，DMA 控制器将使用与每个通道关联的自然优先级。当多个通道具有待处理传输时，将按照以下方式选择下一个发送数据的通道：

- 优先级最高的通道将完成所有单元传输，然后再切换到优先级较低的通道（请参见图 31-5 中的 PRI3 传输）。
- 如果多个通道的优先级相同（CHPRI 相同），则控制器将在具有该优先级的所有通道之间轮换。对于具有最高优先级的每个通道，控制器将允许正在进行单元传输的通道完成单个事务（当前单元传输），然后允许同一优先级的下一个通道完成单个事务（请参见图 31-5 中标记 C 和 B 之间的 PRI2 传输）。
- 如果在某个优先级较低的通道正在执行事务时，另一个优先级较高的通道请求进行传输，则控制器会完成当前传输，然后再切换到优先级较高的通道（请参见图 31-5 中标记 A 处的事件）。

图 31-5: 通道优先级行为



### 31.4.8 字节对齐

DMA 控制器的字节对齐功能让用户无需对源与目标地址进行对齐。事务的读操作部分将读取给定字中可读取的最大字节数。例如，如果源指针与源大小之间的差距为  $N > 4$  字节，则在源指针指向字节 0 时，将读取 4 字节，在源指针指向字节 1 时将读取 3 字节，如此类推。如果源中剩余的字节数为  $N < 4$ ，则只会读取前  $N$  个字节。当所读取的内容包含了读取时更新的寄存器时，这非常重要。

在每次写操作之后，源指针和目标指针使用已写入的字节数进行更新。用户应当注意，当传输被中止时，在事务完成之前，源指针不一定会反映已发生的读操作。

### 31.4.9 地址错误

如果在传输期间出现的地址（源或目标）为非法地址，通道的地址错误中断标志  $\text{CHERIF}$  ( $\text{DCHxINT}<0>$ ) 位将置 1。通道将被禁止（即硬件会将  $\text{CHEN}$  位复位）。

通道状态不会受影响，以便协助对问题进行调试。

## 31.5 中断

DMA 器件能够产生一些中断，以反映在通道的数据传输期间发生的事件：

- 错误中断，通过每个通道的 **CHERIF** 位 (**DCHxINT<0>**) 指示，并使用 **CHERIE** (**DCHxINT<16>**) 位允许。在通道传输操作期间发生地址错误时，将发生该事件。
- 中止中断，通过每个通道的 **CHTAIF** 位 (**DCHxINT<1>**) 指示，并使用 **CHTAIE** (**DCHxINT<17>**) 位允许。在允许中止中断请求 (**AIRQEN** (**DCHxECON<3>**) = 1) 的情况下，当 DMA 通道传输由于系统事件 (中断) 与 **CHAIQ<7:0>** (**DCHxECON<23:16>**) 位匹配而中止时，将发生该事件。
- 数据块传输完成中断，通过每个通道的 **CHBCIF** (**DCHxINT<3>**) 位指示，并使用 **CHBCIE** (**DCHxINT<19>**) 位允许。当 DMA 通道数据块传输完成时，会发生该事件。
- 单元传输完成中断，通过每个通道的 **CHCCIF** (**DCHxINT<2>**) 位指示，并使用 **CHCCIE** (**DCHxINT<18>**) 位允许。当 DMA 通道单元传输完成时，会发生该事件。
- 源地址指针活动中断：在通道源指针达到源结束位置时产生，通过 **CHSDIF** (**DCHxINT<7>**) 位指示，并通过 **CHSDIE** (**DCHxINT<23>**) 位允许；或者在通道源指针达到源中点位置时产生，通过 **CHSHIF** (**DCHxINT<6>**) 位指示，并通过 **CHSHIE** (**DCHxINT<22>**) 位允许。
- 目标地址指针活动中断：在通道目标指针达到目标结束位置时产生，通过 **CHDDIF** (**DCHxINT<5>**) 位指示，并通过 **CHDDIE** (**DCHxINT<21>**) 位允许；或者在通道目标指针达到目标中点位置时产生，通过 **CHDHIF** (**DCHxINT<4>**) 位指示，并通过 **CHDHIE** (**DCHxINT<20>**) 位允许。

所有属于 DMA 通道的中断都映射到相应的通道中断向量。

**注：** 并非所有 DMA 通道在所有器件上都可用。关于可用性，请参见具体器件数据手册的“中断”章节。

### 31.5.1 中断配置

每个 DMA 通道在内部具有多个中断标志 (**CHSDIF**、**CHSHIF**、**CHDDIF**、**CHDHIF**、**CHBCIF**、**CHCCIF**、**CHTAIF** 和 **CHERIF**) 和相应的中断允许控制位 (**CHSDIE**、**CHSHIE**、**CHDDIE**、**CHDHIE**、**CHBCIE**、**CHCCIE**、**CHTAIE** 和 **CHERIE**)。

但是，对于中断控制器，每个通道只有一个专用的中断标志位 **DMAxIF** 和相应的中断允许 / 屏蔽位 **DMAxIE**。

**注：** 根据不同的器件型号，最多有 8 个 (即 0-7) 中断标志和中断允许 / 屏蔽位可用。关于可用性，请参见具体器件数据手册的“中断”章节。

因此请注意，特定 DMA 通道的所有中断条件仅共用一个中断向量。每个 DMA 通道可以具有独立于其他 DMA 通道的优先级。

**例 31-9: 允许中断的 DMA 通道初始化代码示例**

```

/* This code example illustrates a DMA channel 0 interrupt configuration. When the DMA channel
0 interrupt is generated, the CPU will jump to the vector assigned to DMA0 interrupt. */

IEC1CLR=0x00010000;           // disable DMA channel 0 interrupts
IFS1CLR=0x00010000;           // clear any existing DMA channel 0 interrupt flag

DMACONSET=0x00008000;          // enable the DMA controller
DCH0CON=0x03;                  // channel off, priority 3, no chaining

DCH0ECON=0;                    // no start or stop irq's, no pattern match

                                // program the transfer
DCH0SSA=VirtToPhys(flashBuff); // transfer source physical address
DCH0DSA=VirtToPhys(ramBuff);    // transfer destination physical address
DCH0SSIZ=200;                   // source size 200 bytes
DCH0DSIZ=200;                   // destination size 200 bytes
DCH0CSIZ=200;                   // 200 bytes transferred per event

DCH0INTCLR=0x00ff00ff;          // clear existing events, disable all interrupts
DCH0INTSET=0x00090000;          // enable Block Complete and error interrupts

IPC9CLR=0x0000001f;             // clear the DMA channel 0 priority and sub-priority
IPC9SET=0x00000016;             // set IPL 5, sub-priority 2
IEC1SET=0x00010000;             // enable DMA channel 0 interrupt

DCH0CONSET=0x80;                // turn channel on
                                // initiate a transfer
DCH0ECONSET=0x00000080;         // set CFORCE to 1

```

**例 31-10: DMA 通道 0 ISR 代码示例**

```

/*This code example demonstrates a simple Interrupt Service Routine for DMA channel 0
interrupts. The user's code at this vector should perform any application specific operations
and must clear the DMA0 interrupt flags before exiting. */

void __ISR(_DMA_0_VECTOR, IPL5) __DMA0Interrupt(void)
{
    int dmaFlags=DCH0INT&0xff;    // read the interrupt flags

    /*
    perform application specific operations in response to any interrupt flag set
    */

    DCH0INTCLR=0x000000ff;         // clear the DMA channel interrupt flags
    IFS1CLR = 0x00010000;          // Be sure to clear the DMA0 interrupt flags
                                // before exiting the service routine.
}

```

**注:** DMA ISR 代码示例显示的是 MPLAB® C32 C 编译器的特定语法。关于对 ISR 的支持, 请参见编译器手册。

## 31.6 节能和调试模式下的操作

### 31.6.1 休眠模式下的 DMA 操作

当器件进入 Sleep（休眠）模式时，系统时钟被禁止。在该模式下不能发生任何 DMA 活动。

## 31.7 各种复位的影响

### 31.7.1 器件复位

在发生器件复位时，所有 DMA 寄存器会被强制设为它们的复位状态。当异步复位输入变为有效时，DMA 逻辑将：

- 复位 DMACON、DMASTAT、DMAADDR、DCRCCON、DCRCDATA 和 DCRCXOR 中的所有字段
- 在每个通道的寄存器字段中设置相应值：DCHxCON、DCHxECON、DCHxINT、DCHxSSIZ、DCHxDSIZ、DCHxSPTR、DCHxDPTR、DCHxCSIZ、DCHxCPTR 和 DCHxDAT
- 复位之后寄存器 DCHxSSA 和 DCHxDSA 具有随机值
- 中止所有正在进行的数据传输

### 31.7.2 上电复位

在发生上电复位时，所有 DMA 寄存器会被强制设为它们的复位状态。

### 31.7.3 看门狗定时器复位

在发生看门狗定时器复位时，所有 DMA 寄存器会被强制设为它们的复位状态。

31.8 相关应用笔记

本节列出了与手册本章内容相关的应用笔记。这些应用笔记可能并不是专为 PIC32 器件系列而编写的，但其概念是相近的，通过适当修改并受到一定限制即可使用。当前与直接存储器访问 (DMA) 模块相关的应用笔记有：

标题	应用笔记编号
目前没有相关的应用笔记。	N/A

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

## 31.9 版本历史

### 版本 A（2007 年 10 月）

这是本文档的初始版本。

### 版本 B（2007 年 10 月）

更新了文档（删除了“机密”状态）。

### 版本 C（2008 年 4 月）

将状态修改为“初稿”；将 U-0 修改为 r-x；修改了表 31-1；修改了表 31-2（DCHxCON，bit 3），删除了“注 1”；修改了寄存器 31-19、31-39、31-43、31-47、31-48、31-49 和 31-53；修改了第 31.3 节和第 31.3.2 节；修改了例 31-1、31-3、31-4、31-6、31-7 和 31-8；删除了例 31-2 并对示例进行了重新编号；删除了第 31.3.3 节并对小节进行了重新编号；修改了第 31.3.20.7 节。

### 版本 D（2008 年 6 月）

修改了寄存器 31-58 至 31-60 的脚注；修改了例 31-8；将保留位从“保持为”更改为“写入”；为 ON 位（DMACON 寄存器）增加了注释。

### 版本 E（2009 年 8 月）

该版本引入了一些仅在特定器件上提供的新的位和功能。以下详细介绍了产生的变化：

- DMA 寄存器汇总（表 31-1）
  - 增加了 BUSY、BYTO1、BYTO0、WBO、BITO、CRCTYP 和 CHBUSY 位
  - 删除了对 IEC1、IPC9 和 IFS1 寄存器的引用
  - 在“DMA 寄存器汇总”中增加了“地址偏移”栏
  - 增加了介绍清零、置 1 和取反寄存器的“注 1”、“注 2”和“注 3”
  - 增加了关于一些特定位和位范围是否可用取决于器件型号的“注 4”和“注 5”
- 在以下寄存器中增加了介绍清零、置 1 和取反寄存器的注释：
  - DMACON（寄存器 31-1）
  - DMASTAT（寄存器 31-2）
  - DMAADDR（寄存器 31-3）
  - DCRCCON（寄存器 31-4）
  - DCRCDATA（寄存器 31-5）
  - DCRCXOR（寄存器 31-6）
  - DCHxCON（寄存器 31-7）
  - DCHxECON（寄存器 31-8）
  - DCHxINT（寄存器 31-9）
  - DCHxSSA（寄存器 31-10）
  - DCHxDSA（寄存器 31-11）
  - DCHxSSIZ（寄存器 31-12）
  - DCHxDSIZ（寄存器 31-13）
  - DCHSPTR（寄存器 31-14）
  - DCHxDPTR（寄存器 31-15）
  - DCHxCSIZ（寄存器 31-16）
  - DCHxCPTR（寄存器 31-17）
  - DCHxDAT（寄存器 31-18）
- 删除了以下寄存器：IFS1、IEC1 和 IPC9
- 在寄存器 31-1 中增加了 BUSY 位（DMACON<11>）以及关于 SIDL 和 BUSY 位可用性的“注 1”



## 版本 E（2009 年 8 月）（续）

- 更新了寄存器 31-2 中的 DMACH 位（DMASTAT<2:0>），并增加了关于所有位的可用性的“注 2”
- 在寄存器 31-4 中增加了 BYTO1、BYTO0、WBO、BITO 和 CRCTYP 位，更新了 PLEN<4:0> 和 CRCCH<2:0> 位，并增加了“注 1”和“注 2”
- 更新了寄存器 31-5 中的 DCRCDATA 位，并增加了“注 1”
- 更新了寄存器 31-6 中的 DCRCXOR 位，并增加了“注 1”
- 在寄存器 31-7 中增加了 CHBUSY 位（DCHxCON<15>）和“注 1”
- 更新了寄存器 31-12 中的 DCHxSSIZ 位，并增加了“注 1”
- 更新了寄存器 31-13 中的 DCHxDSIZ 位，并增加了“注 1”
- 更新了寄存器 31-14 中的 DCHxSPTR 位，并增加了“注 2”
- 更新了寄存器 31-15 中的 DCHxDPTR 位，并增加了“注 1”
- 更新了寄存器 31-16 中的 DCHxCSIZ 位，并增加了“注 1”
- 更新了寄存器 31-17 中的 DCHxCPTR 位，并增加了“注 2”
- 更新了第 31.3.4 节“通道链接模式工作原理”中最低优先级通道编号，并为第四段增加了相关注释
- 在第 31.3.7 节“暂停传输”和第 31.4.6 节“DMA 暂停”中增加了关于暂停 DMA 模块的信息和相关注释
- 更新了第 31.3.6 节“特殊功能模块（SFM）模式”，以区分 16 位和 32 位 CRC
- 增加了第 31.3.6.5 节“计算 IP 头校验和”
- 在第 31.4 节“中断”中增加了 DMA 通道中断标志、允许位和优先级位
- 在表 31-6 中增加了 DMA 中断向量（DMA4-DMA7）
- 更新了第 31.5.1 节“休眠模式下的 DMA 操作”

## 版本 F（2010 年 10 月）

该版本包含以下更新：

- 在章节开始处添加注释框，提供补充文档信息
- 在所有寄存器中，将所有的“保留”写入 0；忽略读操作”修改为“未实现：读为 0”；并更新了默认的 POR 定义
- 在下列表和寄存器中添加注 1、2 和 3，说明清零、置 1 和取反寄存器：
  - 表 31-1: DMA 寄存器汇总
  - 寄存器 31-1: DMACON: DMA 控制器控制寄存器
  - 寄存器 31-4: DCRCCON: DMA CRC 控制寄存器
  - 寄存器 31-5: DCRCDATA: DMA CRC 数据寄存器
  - 寄存器 31-6: DCRCXOR: DMA CRCXOR 使能寄存器
  - 寄存器 31-7: DCHxCON: DMA 通道 x 控制寄存器
  - 寄存器 31-8: DCHxECON: DMA 通道 x 事件控制寄存器
  - 寄存器 31-9: DCHxINT: DMA 通道 x 中断控制寄存器
  - 寄存器 31-10: DCHxSSA: DMA 通道 x 源起始地址寄存器
  - 寄存器 31-12: DCHxSSIZ: DMA 通道 x 源大小寄存器
  - 寄存器 31-13: DCHxDSIZ: DMA 通道 x 目标大小寄存器
  - 寄存器 31-16: DCHxCSIZ: DMA 通道 x 单元大小寄存器
  - 寄存器 31-18: DCHxECON: DMA 通道 x 事件控制寄存器
- 删除所有清零、置 1 和取反寄存器
- 删除所有中断寄存器
- 将寄存器 31-1 和表 31-1 中的 BUSY 位改名为 DMABUSY
- 在第 31.3.6 节“特殊功能模块（SFM）模式”最后添加了一个注释框
- 对全文的格式和内容进行了少量更新

## 版本 G（2012 年 4 月）

该版本包含以下更新：

- 更新了典型的 DMA 源 - 目标传输图（见图 31-1）
- 更新了 DMA 模块框图（见图 31-2）
- 删除了 CRC 实现详细信息（图 31-3）
- 更新了第 31.2 节“状态和控制寄存器”的寄存器定义
- 从 DMACON 寄存器中删除 FRZ 位（见寄存器 31-1）
- 向 DMASTAT 寄存器中添加了 CMAWCH<2:0> 位（见寄存器 31-2）
- 向 DCHxCON 寄存器中添加了 CHPIGN<7:0>、CHPIGNEN 和 CHPATLEN 位（见寄存器 31-7）
- 向 DCHxDAT 寄存器中添加了 CHPDAT<15:8> 位（见寄存器 31-18）
- 添加了第 31.3.2.1 节“模式匹配忽略模式”
- 从现有内容中创建了新章节：第 31.4 节“通道控制”
- 删除了源和目标指针更新示例 1 和 2（表 31-3 和表 31-4）
- 删除了第 31.3.17 节“通道中止”
- 删除了第 31.5.3 节“调试模式下的 DMA 操作”
- 删除了第 31.6.1 节“空闲模式下的 DMA 操作”
- 更新了所有寄存器的格式，并对全文的格式和内容进行了少量更新

## 版本 H（2013 年 11 月）

该版本包含以下更新：

- 将所有 BMX 和总线矩阵更新为系统总线
- 在“DMA 寄存器汇总”中，删除了注 3 并更新了注 2。（见表 31-1）
- 在 DMASTAT 寄存器中，添加了额外的 RDWR 位并删除了 DMAWCH<2:0> 和 DMARCH<2:0> 位（见表 31-1 和寄存器 31-2）
- 对全文的格式和内容进行了少量更新

---

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。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、FlashFlex、KEELOQ、KEELOQ 徽标、MPLAB、PIC、PICmicro、PICSTART、PIC<sup>32</sup> 徽标、rPIC、SST、SST 徽标、SuperFlash 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、MTP、SEEVAL 和 The Embedded Control Solutions Company 均为 Microchip Technology Inc. 在美国的注册商标。

Silicon Storage Technology 为 Microchip Technology Inc. 在除美国外的国家或地区的注册商标。

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、MPF、MPLAB Certified 徽标、MPLIB、MPLINK、mTouch、Omniscient Code Generation、PICC、PICC-18、PICDEM、PICDEM.net、PICKIT、PICKIT 徽标、REAL ICE、rLAB、Select Mode、SQI、Serial Quad I/O、Total Endurance、TSHARC、UniWinDriver、WiperLock、ZENA 和 Z-Scale 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。

GestIC 和 ULPP 为 Microchip Technology Inc. 的子公司 Microchip Technology Germany II GmbH & Co. & KG 在除美国外的国家或地区的注册商标。

在此提及的所有其他商标均为各持有公司所有。

© 2010-2014, Microchip Technology Inc. 版权所有。

ISBN: 978-1-62077-805-0

**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® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品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。

## 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>

网址: [www.microchip.com](http://www.microchip.com)

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

**奥斯汀 Austin, TX**  
Tel: 1-512-257-3370

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

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

**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**  
Novi, MI  
Tel: 1-248-848-4000

**休斯敦 Houston, TX**  
Tel: 1-281-894-5983

**印第安纳波利斯 Indianapolis**  
Noblesville, IN  
Tel: 1-317-773-8323  
Fax: 1-317-773-5453

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

**纽约 New York, NY**  
Tel: 1-631-435-6000

**圣何塞 San Jose, CA**  
Tel: 1-408-735-9110

**加拿大多伦多 Toronto**  
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: 852-2943-5100  
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: 86-755-8864-2200  
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: 886-7-213-7830

**台湾地区 - 台北**  
Tel: 886-2-2508-8600  
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: 91-20-3019-1500

**日本 Japan - Osaka**  
Tel: 81-6-6152-7160  
Fax: 81-6-6152-9310

**日本 Japan - Tokyo**  
Tel: 81-3-6880-3770  
Fax: 81-3-6880-3771

**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 - Dusseldorf**  
Tel: 49-2129-3766400

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

**德国 Germany - Pforzheim**  
Tel: 49-7231-424750

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

**意大利 Italy - Venice**  
Tel: 39-049-7625286

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

**波兰 Poland - Warsaw**  
Tel: 48-22-3325737

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

**瑞典 Sweden - Stockholm**  
Tel: 46-8-5090-4654

**英国 UK - Wokingham**  
Tel: 44-118-921-5800  
Fax: 44-118-921-5820