第 31 章 DMA 控制器

目录

本章包括下列主题：

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 PIC32MX 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，PBUS）设备之一：如 SPI、UART 和 I^2C™ 等。

下面列出了 DMA 模块的部分主要特性：

- 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 的数据传输
- 自动字长检测，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
- 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 调试支持以下功能：
  - 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

DMA 控制器还提供了以下特性：

- 未对齐传输
- 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
- CRC 计算

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，并且通道保持使能，直到数据块传输完成为止。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： 典型的 DMA 源 / 目标传输图

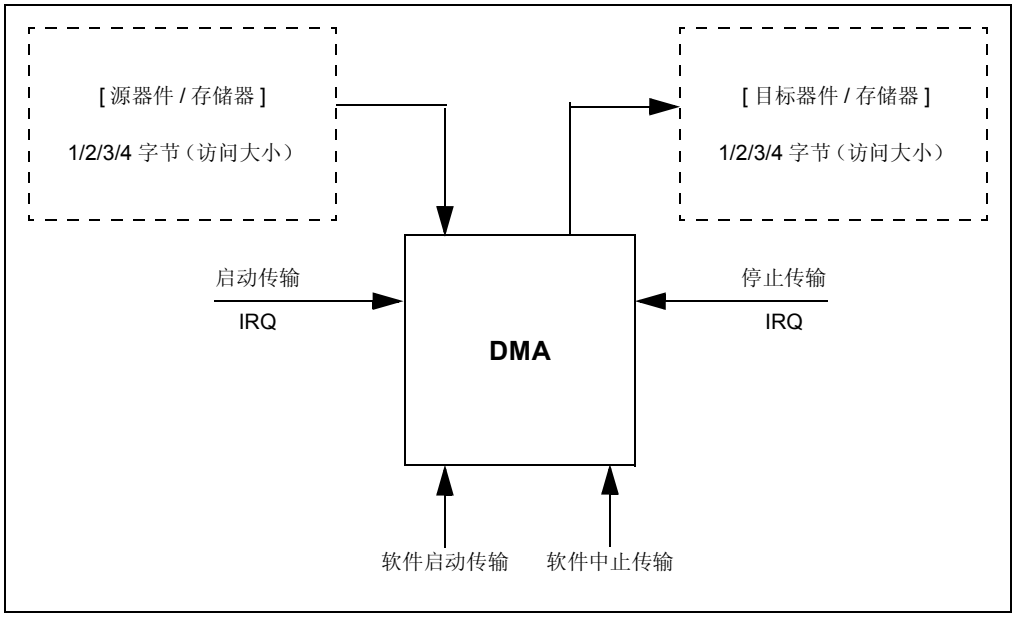


图 31-2： DMA 模块框图

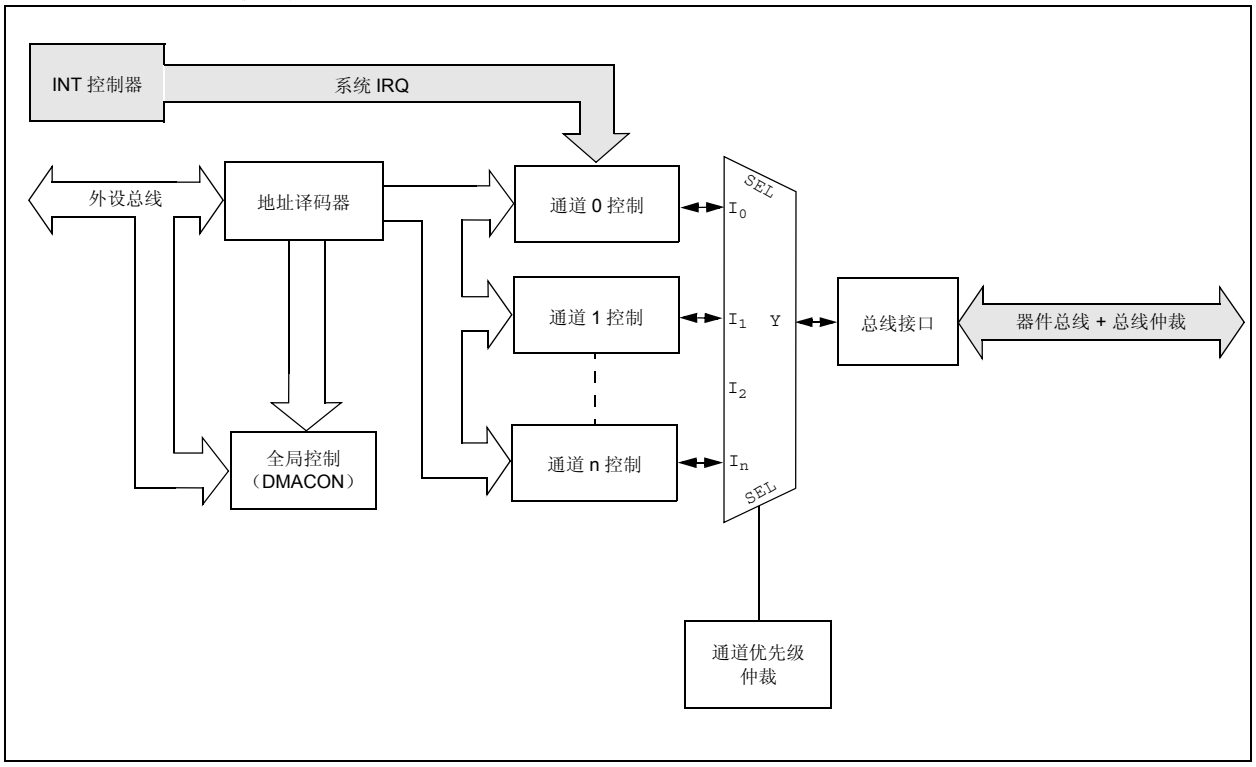
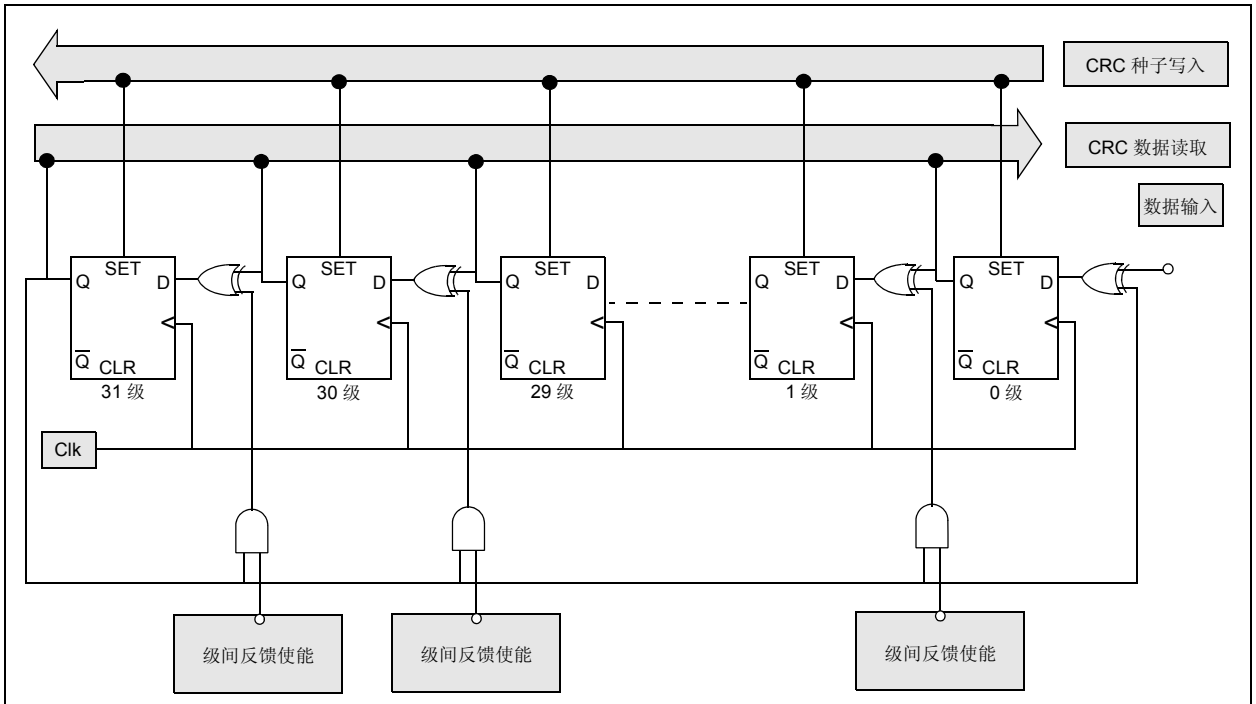


图 31-3： CRC 实现详细信息



注： 根据不同的器件类型，最多提供 32 级（0 至 31）。详情请参见具体器件数据手册。

31.2 状态和控制寄存器

注： 不同的 PIC32MX 器件型号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。

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

- DMACON：DMA 控制器的控制寄存器
- DMASTAT：DMA 模块的状态寄存器
- DMAADDR：DMA 地址寄存器
- DCRCCON：DMA CRC 控制寄存器
- DCRCDATA：DMA CRC 数据寄存器——CRC 发生器的初始值
- DCRCXOR：DMA CRC 异或 （XOR）使能寄存器——提供用于 CRC 发生器计算的多项式的说明
- DCHxCON：DMA 通道 x 控制寄存器
- DCHxECON：DMA 通道 x 事件控制寄存器
- DCHxINT：DMA 通道 x 中断控制寄存器
- DCHxSSA：DMA 通道 x 源起始地址寄存器
- DCHxDSA：DMA 通道 x 目标起始地址寄存器
- DCHxSSIZ：DMA 通道 x 源大小寄存器
- DCHxDSIZ：DMA 通道 x 目标大小寄存器
- DCHxSPTR：DMA 通道 x 源指针寄存器
- DCHxDPTR：DMA 通道 x 目标指针寄存器
- DCHxCSIZ：DMA 通道 x 单元大小寄存器
- DCHxCPTR：DMA 通道 x 单元指针寄存器
- DCHxDAT：DMA 通道 x 模式数据寄存器

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

地址偏移	名称	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
0x00	DMACON(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	ON	FRZ	SIDL(4)	SUSPEND	BUSY(4)	—	—	—
		7:0	—	—	—	—	—	—	—	—
0x10	DMASTAT	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	—	—	—	—	—	—	—	—
		7:0	—	—	—	—	RDWR	DMACH<2:0>(5)		
0x20	DMAADDR	31:24	DMAADDR<31:24>							
		23:16	DMAADDR<23:16>							
		15:8	DMAADDR<15:8>							
		7:0	DMAADDR<7:0>							
0x30	DCRCCON(1,2,3)	31:24	—	—	BYTO1(4)	BYTO0(4)	WBO(4)	—	—	BITO(4)
		23:16	—	—	—	—	—	—	—	—
		15:8	—	—	—	PLEN<4:0>(5)				
		7:0	CRCEN	CRCAPP	CRCTYP(4)	—	—	CRCCH<2:0>(5)		
0x40	DCRCDATA(1,2,3)	31:24	DCRCDATA<31:24>(5)							
		23:16	DCRCDATA<23:16>(5)							
		15:8	DCRCDATA<15:8>							
		7:0	DCRCDATA<7:0>							
0x50	DCRCXOR(1,2,3)	31:24	DCRCXOR<31:24>(5)							
		23:16	DCRCXOR<23:16>(5)							
		15:8	DCRCXOR<15:8>							
		7:0	DCRCXOR<7:0>							
0x60	DCHxCON(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHBUSY(4)	—	—	—	—	—	—	CHCHNS
		7:0	CHEN	CHAED	CHCHN	CHAEN	—	CHEDET	CHPRI<1:0>	
0x70	DCHxECON(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	CHAIRQ<7:0>							
		15:8	CHSIRQ<7:0>							
		7:0	CFORCE	CABORT	PATEN	SIRQEN	AIRQEN	—	—	—
0x80	DCHxINT(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	CHSDIE	CHSHIE	CHDDIE	CHDHIE	CHBCIE	CHCCIE	CHTAIE	CHERIE
		15:8	—	—	—	—	—	—	—	—
		7:0	CHSDIF	CHSHIF	CHDDIF	CHDHIF	CHBCIF	CHCCIF	CHTAIF	CHERIF
0x90	DCHxSSA(1,2,3)	31:24	CHSSA<31:24>							
		23:16	CHSSA<23:16>							
		15:8	CHSSA<15:8>							
		7:0	CHSSA<7:0>							

图注： — = 未实现，读为 0。地址偏移值以十六进制显示。

注 1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR（例如， DMACONCLR）。向清零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。对清零寄存器的读操作将被忽略。

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

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

4： 该位并非在所有器件上都可用。详情请参见具体器件数据手册。

5： 根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

表 31-1： DMA 寄存器汇总 （续）

地址偏移	名称	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
0xA0	DCHxDSA	31:24	CHDSA<31:24>							
		23:16	CHDSA<23:16>							
		15:8	CHDSA<15:8>							
		7:0	CHDSA<7:0>							
0xB0	DCHxSSIZ(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHSSIZ<15:8>(5)							
		7:0	CHSSIZ<7:0>							
0xC0	DCHxDSIZ(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHDSIZ<15:8>(5)							
		7:0	CHDSIZ<7:0>							
0xD0	DCHxSPTR	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHSPTR<15:8>(5)							
		7:0	CHSPTR<7:0>							
0xE0	DCHxDPTR	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHDPTR<15:8>(5)							
		7:0	CHDPTR<7:0>							
0xF0	DCHxCSIZ(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHCSIZ<15:8>(5)							
		7:0	CHCSIZ<7:0>							
0x100	DCHxCPTR	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	CHCPTR<15:8>(5)							
		7:0	CHCPTR<7:0>							
0x110	DCHxDAT(1,2,3)	31:24	—	—	—	—	—	—	—	—
		23:16	—	—	—	—	—	—	—	—
		15:8	—	—	—	—	—	—	—	—
		7:0	CHPDAT<7:0>							

图注： — = 未实现，读为 0。地址偏移值以十六进制显示。

注 1： 该寄存器具有关联的清零寄存器，位于 0x4 字节偏移处。这些清零寄存器的命名方式是在关联寄存器的名称末尾附加 CLR （例如， DMACONCLR）。向清零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。对清零寄存器的读操作将被忽略。

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

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

4： 该位并非在所有器件上都可用。详情请参见具体器件数据手册。

5： 根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-1： DMACON：DMA 控制器控制寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	r-x	r-x	r-x
ON	FRZ	SIDL(1)	SUSPEND	BUSY(1)	—	—	—
bit 15							bit 8

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 7							bit 0

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

bit 31-16 **保留**：写入 0；忽略读操作

bit 15 **ON**：DMA 使能位
1 = 使能 DMA 模块
0 = 禁止 DMA 模块

注： 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 **FRZ**：DMA 冻结位
1 = DMA 在 Debug （调试）模式下停止运行
0 = DMA 在 Debug （调试）模式下继续运行

注： FRZ 仅在调试异常模式下可写，在正常模式下强制为 0。

bit 13 **SIDL**：空闲模式停止位 (1)
1 = 在 Idle （空闲）期间停止 DMA 传输
0 = 在 Idle （空闲）期间继续进行 DMA 传输

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 **BUSY**：DMA 模块忙状态位 (1)
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 **保留**：写入 0；忽略读操作

注 1： 该位并非在所有器件上都可用。详情请参见具体器件数据手册。

寄存器 31-2： DMASTAT：DMA 状态寄存器 (1)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 15							bit 8

r-x	r-x	r-x	r-x	R-0	R-0	R-0	R-0
—	—	—	—	RDWR	DMACH<2:0>(2)		
bit 7							bit 0

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

bit 31-4 **保留：**写入 0；忽略读操作

bit 3 **RDWR：**读 / 写状态位
1 = 上一次 DMA 总线访问是读操作
0 = 上一次 DMA 总线访问是写操作

bit 2-0 **DMACH<2:0>：**DMA 通道位 (2)

注 1：该寄存器包含最近工作的 DMA 通道的值。
2：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-3： DMAADDR：DMA 地址寄存器 (1)

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DMAADDR<31:24>							
bit 31							bit 24

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DMAADDR<23:16>							
bit 23							bit 16

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DMAADDR<15:8>							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
DMAADDR<7:0>							
bit 7							bit 0

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

bit 31-0 **DMAADDR<31:0>**：DMA 模块地址位

注 1：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

r-x	r-x	R/W-0	R/W-0	R/W-0	r-x	r-x	R/W-0
—	—	BYTO<1:0>(1)		WBO(1,2)	—	—	BITO(1)
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	r-x	r-x	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	PLEN<4:0>(2)				
bit 15							bit 8

R/W-0	R/W-0	R/W-0	r-x	r-x	R/W-0	R/W-0	R/W-0
CRCEN	CRCAPP(2)	CRCTYP(1)	—	—	CRCCH<2:0>(1)		
bit 7							bit 0

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

bit 31-30 **保留：**写入 0；忽略读操作

bit 29-28 **BYTO<1:0>：**CRC 字节顺序选择位 (1)
11 = 在半字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 (1,2)
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 **保留：**写入 0；忽略读操作

bit 24 **BITO：**CRC 位顺序选择位 (1)
当 DCRCCON<CRCTYP> = 1 时（CRC 模块处于 IP 头模式）：
1 = IP 头校验和使用从最低有效位（LSb）开始的方式计算（即，进行反射）
0 = IP 头校验和使用从最高有效位（MSb）开始的方式计算（即，不进行反射）

当 DCRCCON<CRCTYP> = 0 时（CRC 模块处于 LFSR 模式）：
1 = LFSR CRC 使用从最低有效位开始的方式计算（即，进行反射）
0 = LFSR CRC 使用从最高有效位开始的方式计算（即，不进行反射）

bit 23-13 **保留：**写入 0；忽略读操作

注 1：根据不同的器件类型，并非所有位在所有器件上都可用。详情请参见具体器件数据手册。
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 12-8 **PLEN<4:0>**：多项式长度位 (2)

当 DCRCCON<CRCTYP> = 1 时 （CRC 模块处于 IP 头模式）：
不使用这些位。

当 DCRCCON<CRCTYP> = 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 (2)
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

bit 5 **CRCTYP**：CRC 类型选择位 (1)
1 = CRC 模块将计算 IP 头校验和
0 = CRC 模块将计算 LFSR CRC

bit 4-3 **保留**：写入 0；忽略读操作

bit 2-0 **CRCCH<2:0>**：CRC 通道选择位 (1)
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：根据不同的器件类型，并非所有位在所有器件上都可用。详情请参见具体器件数据手册。
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

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>(1)							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DCRCDATA<23:16>(1)							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DCRCDATA<15:8>(1)							
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
DCRCDATA<7:0>(1)							
bit 7							bit 0

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

bit 31-0 **DCRCDATA<31:0>**：CRC 数据寄存器位 (1)

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 的位将返回 0。

当 DCRCCON<CRCTYP> = 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，并以二进制补码的形式回读 （即，当前 IP 头校验和值）。

当 DCRCCON<CRCTYP> = 0 时 （CRC 模块处于 LFSR 模式）：
在每次读取时，高于 PLEN 的位将返回 0。

注 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-6： DCRCXOR：DMA CRC 异或使能寄存器

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DCRCXOR<31:24>(1)							
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DCRCXOR<23:16>(1)							
bit 23							bit 16

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
DCRCXOR<15:8>(1)							
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
DCRCXOR<7:0>(1)							
bit 7							bit 0

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

bit 31-0 **DCRCXOR<31:0>**：CRC 异或寄存器位 (1)

当 DCRCCON<CRCTYP> = 1 时 （CRC 模块处于 IP 头模式）：
不使用该寄存器。

当 DCRCCON<CRCTYP> = 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 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-7： DCHxCON：DMA 通道 x 控制寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-0	r-x	r-x	r-x	r-x	r-x	r-x	R/W-0
CHBUSY(1)	—	—	—	—	—	—	CHCHNS(2)
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	r-x	R-0	R/W-0	R/W-0
CHEN(3)	CHAED	CHCHN	CHAEN	—	CHEDET	CHPRI<1:0>	
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15 **CHBUSY：**通道忙状态位 (1)
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 14-9 **保留：**写入 0；忽略读操作

bit 8 **CHCHNS：**链通道选择位 (2)
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 (3)
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：**通道链使能位
1 = 允许对通道进行链接
0 = 不允许对通道进行链接

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 **保留：**写入 0；忽略读操作

注 1：该位并非在所有器件上都可用。详情请参见具体器件数据手册。

2：链通道选择位在使能通道链 （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 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：该位并非在所有器件上都可用。详情请参见具体器件数据手册。

2：链通道选择位在使能通道链 （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

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

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>							
bit 23							bit 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
CHSIRQ<7:0>							
bit 15							bit 8

S-0	S-0	R/W-0	R/W-0	R/W-0	r-x	r-x	r-x
CFORCE	CABORT	PATEN	SIRQEN	AIRQEN	—	—	—
bit 7							bit 0

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

bit 31-24 **保留：**写入 0；忽略读操作

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，并且不终止传输

bit 2-0 **保留：**写入 0；忽略读操作

寄存器 31-9： DCHxINT：DMA 通道 x 中断控制寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CHSDIE	CHSHIE	CHDDIE	CHDHIE	CHBCIE	CHCCIE	CHTAIE	CHERIE
bit 23							bit 16

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
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
CHSDIF	CHSHIF	CHDDIF	CHDHIF	CHBCIF	CHCCIF	CHTAIF	CHERIF
bit 7							bit 0

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

bit 31-24 **保留**：写入 0；忽略读操作

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 **保留**：写入 0；忽略读操作

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

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

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

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

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHSSA<31:24>							
bit 31							bit 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
CHSSA<23:16>							
bit 23							bit 16

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHSSA<15:8>							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHSSA<7:0>							
bit 7							bit 0

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

bit 31-0 **CHSSA<31:0>：**通道源起始地址位
通道源起始地址。

注： 这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

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHDSA<31:24>							
bit 31							bit 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
CHDSA<23:16>							
bit 23							bit 16

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHDSA<15:8>							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHDSA<7:0>							
bit 7							bit 0

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

bit 31-0 **CHDSA<31:0>：**通道目标起始地址位

通道目标起始地址。

注： 这必须是目标的物理地址。

寄存器 31-12： DCHxSSIZ：DMA 通道 x 源大小寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHSSIZ<15:8>(1)							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHSSIZ<7:0>(1)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHSSIZ<15:0>：**通道源大小位 (1)

65335 = 源大小为 65,535 字节

•

•

•

2 = 源大小为 2 字节

1 = 源大小为 1 字节

0 = 源大小为 65,536 字节

注 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-13： DCHxDSIZ：DMA 通道 x 目标大小寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHDSIZ<15:8>(1)							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHDSIZ<7:0>(1)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHDSIZ<15:0>：**通道目标大小位 (1)

65535 = 目标大小为 65,535 字节

•

•

•

2 = 目标大小为 2 字节

1 = 目标大小为 1 字节

0 = 目标大小为 65,536 字节

注 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-14： DCHxSPTR：DMA 通道 x 源指针寄存器 (1)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHSPTR<15:8>(2)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHSPTR<7:0>(2)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHSPTR<15:0>：**通道源指针位 (2)

65535 = 指向源的字节 65,535

•

•

•

1 = 指向源的字节 1

0 = 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。

2：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-15： DCHxDPTR：DMA 通道 x 目标指针寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHDPTR<15:8>(1)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHDPTR<7:0>(1)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHDPTR<15:0>：**通道目标指针位 (1)

65535 = 指向目标的字节 65,535

•

•

•

1 = 指向目标的字节 1

0 = 指向目标的字节 0

注 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-16： DCHxCSIZ：DMA 通道 x 单元大小寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHCSIZ<15:8>(1)							
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHCSIZ<7:0>(1)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHCSIZ<15:0>：**通道单元大小位 (1)

65535 = 在发生事件时传输 65,535 字节

•

•

•

2 = 在发生事件时传输 2 字节

1 = 在发生事件时传输 1 字节

0 = 在发生事件时传输 65,536 字节

注 1：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-17： DCHxCPTR：DMA 通道 x 单元指针寄存器 (1)

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHCPTR<15:8>(2)							
bit 15							bit 8

R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
CHCPTR<7:0>(2)							
bit 7							bit 0

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

bit 31-16 **保留：**写入 0；忽略读操作

bit 15-0 **CHCPTR<7:0>：**通道单元进度指针位 (2)

65535 = 自上一个事件以来已传输了 65,535 字节

•

•

•

1 = 自上一个事件以来已传输了 1 字节

0 = 自上一个事件以来已传输了 0 字节

注 1：处于模式检测模式时，该寄存器会在模式检测时复位。

2：根据不同的器件类型，并非所有位都可用。详情请参见具体器件数据手册。

寄存器 31-18： DCHxDAT：DMA 通道 x 模式数据寄存器

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 31							bit 24

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 23							bit 16

r-x	r-x	r-x	r-x	r-x	r-x	r-x	r-x
—	—	—	—	—	—	—	—
bit 15							bit 8

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
CHPDAT<7:0>							
bit 7							bit 0

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

bit 31-8 **保留：**写入 0；忽略读操作

bit 7-0 **CHPDAT<7:0>：**通道数据寄存器位

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。

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

请注意，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。

31.3.1 DMA 控制器术语

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。

31.3.2 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寄存器传输到目标寄存器。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 位 （DCHxECON<6>）
- 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 31.3.3 节 “模式匹配终止模式工作原理”**中所述使能了模式匹配）
- 器件上发生与 CHAIRQ <7:0> 位 （DCHxECON<23:16>）中断匹配的中断事件 （如果已通过 AIRQEN （DCHxECON<3>）允许）
- 检测到地址错误
- 单元传输完成
- 在通道自动使能模式 （CHAEN）未置 1 的情况下，数据块传输完成

在发生通道中止中断时，通道传输中止中断标志 CHTAIF （DCHxINT<1>）位会置 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 31-4。

例 31-1：　　基本传输模式下 DMA 通道初始化的代码示例

```
/*
The following 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, pri 3, no chaining

       CH0ECON=0;                    // no start or stop irq's, no pattern match

                                     // program the transfer
       DCH0SSA=0x1d010000;           // transfer source physical address
       DCH0DSA=0x1d020000;           // 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
```

31.3.2.1　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 位 （DMACON<15>）为 0）时
- 数据块传输完成，无论 CHAEN （DCHxCON<4>）的状态如何
- 模式匹配终止传输，无论 CHAEN （DCHxCON<4>）的状态如何
- 写入 CABORT （DCHxECON<6>）标志
- 源或目标起始地址被更新

31.3.3 模式匹配终止模式工作原理

通过模式匹配终止模式，用户可以在事务期间写入的数据字节与特定模式（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 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 （DCHxECON<15:8>）设置为 UART “发送缓冲区为空” IRQ。
- SIRQEN （DCHxECON<4>）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>）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。

例 31-2：　　模式匹配传输模式下 DMA 通道初始化的代码示例

```
/*
The following 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=0x1d020000;            // 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


                                      // wait for the UART1 RX interrupt to initiate a
                                      // transfer

                                      // do something else

                                      // will get an interrupt when the transfer is done
                                      // or when an address error occurred
```

31.3.4 通道链模式工作原理

通道链是对 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>）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 CHBCIF（CHxINT<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，通道 7 优先级最低。在使能通道链（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>）进行选择）。

注：	在某些器件中，通道 0 优先级最高，通道 4 优先级最低。关于可用性，请参见具体器件数据手册。

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>）使能。该位在通道链模式下特别有用，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. A/D 转换器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 A/D 转换器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：　　通道链模式下 DMA 通道初始化的代码示例

```
/*
The following 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 will start 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

                             // do something else

                             // the UART1 RX interrupts will initiate the DMA channel 0 transfer
                             // once this transfer is complete, the DMA channel 1 will start
                             // upon DMA channel 1 transfer completion will get an interrupt

 while(!intCh1Ocurred);      // poll DMA channel 1 interrupt
```

31.3.5 通道自动使能模式工作原理

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 位 （DCHxCON<7>）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 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。

31.3.6 暂停传输

用户应用程序可以通过写入 SUSPEND 位 （DMACON<12>）来立即暂停 DMA 模块。这会立即暂停 DMA 控制器，使之不再执行任何后续的总线事务。

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 BUSY （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：	BUSY 位并非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。

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。

根据不同的器件型号，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>）置 1 而继续接收事件 （中止中断等）。

31.3.7 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。这会清零通道标志位 （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。

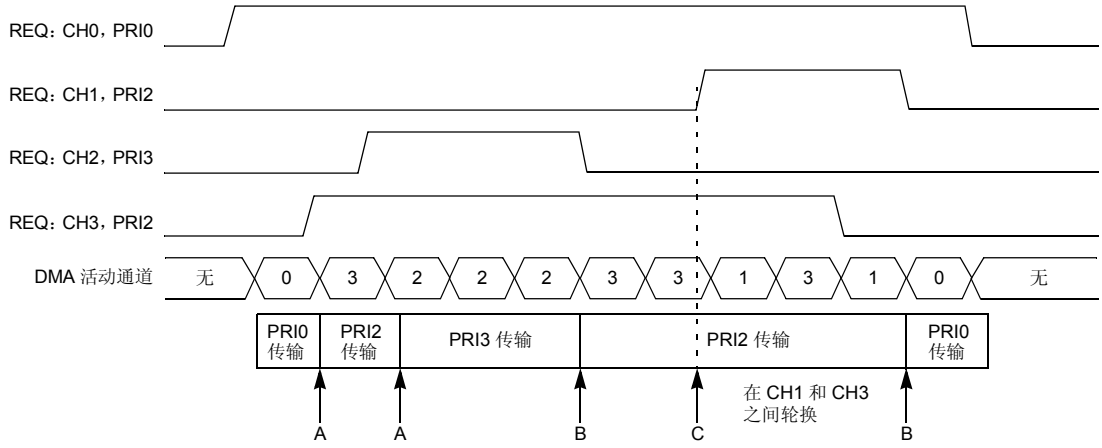
注：	只有在通道被禁止时，才能更改通道大小。

31.3.8 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。每个通道具有两个优先级位 CHPRI<1:0>（DCHxCON<1:0>）。这两个位标识通道的优先级。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 31-4 中的 “PRI3 传输”行为）。
- 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 31-4 中标记 “C”和 “B”之间的 “PRI2 传输”行为）。
- 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 31-4 中标记 “A”处的事件）。

图 31-4： 通道优先级行为



转换图注：

A——优先级较高的传输请求；暂停当前传输，并紧接着进行传输。

B——所有优先级最高的传输完成；切换到优先级较低的通道。

C——在具有当前优先级的所有通道之间轮换。

31.3.9 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。

表 31-2 给出了该行为的示例。示例 1 说明了在两个大缓冲区之间传输 9 个字节的简单传输，在该示例中，CHxSSA = 0x1000，CHxSSIZ = 100，CHxDSA = 0x43F9，CHxDSIZ = 100，以及 CHxCSIZ = 9。

表 31-2： 源和目标指针更新——示例 1

事务	工作状态	源指针	目标指针	传输计数 / 大小	读地址	写地址	读数据 (1)	写数据 (2)
1	读	9	11	0/9	1009	xxxx	33_22_11_XX	XX_XX_XX_XX
1	写 1	9	11	0/9	1009	440A	33_22_11_XX	22_11_XX_XX
1	指针更新 (3)	B	13	2/9	1009	440A	33_22_11_XX	XX_XX_XX_XX
1	写 2	B	13	2/9	1009	440C	33_22_11_XX	XX_XX_XX_33
1	指针更新 (3)	C	14	3/9	1009	440C	33_22_11_XX	XX_XX_XX_XX
2	读	C	14	3/9	100C	440C	77_66_55_44	XX_XX_XX_XX
2	写 1	C	14	3/9	100C	440D	77_66_55_44	66_55_44_XX
2	指针更新 (3)	F	17	6/9	100C	440D	77_66_55_44	XX_XX_XX_XX
2	写 2	F	17	6/9	100C	4410	77_66_55_44	XX_XX_XX_77
2	指针更新 (3)	10	18	7/9	100C	4410	77_66_55_44	XX_XX_XX_XX
3	读	10	18	7/9	1010	4410	XX_XX_99_88	XX_XX_XX_XX
3	写 1	10	18	7/9	1010	4411	XX_XX_XX_88	XX_99_88_XX
3	指针更新 (3)	12	1A	9/9	1010	4411	XX_XX_XX_88	XX_XX_XX_XX

注 1： XX 表示读取的数据被丢弃。
2： XX 表示数据未被写入。
3： 在指针发生更新时，中断按需要进行更新。

表 31-3 给出了该行为的另一个示例。示例 2 说明了最坏情况下的总线利用情况（即，缓冲区未对齐，目标缓冲区出现折回现象）；在该示例中，CHxSSA = 0x1000，CHxSSIZ = 100，CHxDSA = 0x4402，CHxDSIZ = 4，以及 CHxCSIZ = 8。

表 31-3： 源和目标指针更新——示例 2

事务	工作状态	源指针	目标指针	传输计数 / 大小	读地址	写地址	读数据 (1)	写数据 (2)
1	读	9	0	0/8	1009	xxxx	33_22_11_XX	XX_XX_XX_XX
1	写 1	9	0	0/8	1009	4402	33_22_11_XX	22_11_XX_XX
1	指针更新 (3)	B	2	2/8	1009	4402	33_22_11_XX	XX_XX_XX_XX
1	写 2	B	2	2/8	1009	4404	33_22_11_XX	XX_XX_XX_33
1	指针更新 (3)	C	3	3/8	1009	4404	33_22_11_XX	XX_XX_XX_XX
2	读	C	3	3/8	100C	4404	77_66_55_44	XX_XX_XX_XX
2	写 1	C	3	3/8	100C	4405	77_66_55_44	XX_XX_44_XX
2	指针更新 (3)	D	0	4/8	100C	4405	77_66_55_44	XX_XX_XX_XX
2	写 2	D	0	4/8	100C	4402	77_66_55_44	66_55_XX_XX
2	指针更新 (3)	F	2	6/8	100C	4402	77_66_55_44	XX_XX_XX_XX
3	写 3	F	2	6/8	100F	4404	77_66_55_44	XX_XX_XX_77
3	指针更新 (3)	10	3	7/8	100F	4404	77_66_55_44	XX_XX_XX_XX
3	读	10	18	7/8	1010	4404	BB_AA_99_88	XX_XX_XX_XX
3	写 1	10	18	7/8	1010	4405	BB_AA_99_88	XX_XX_88_XX
3	指针更新 (3)	11	1A	8/8	1010	4405	77_66_55_44	XX_XX_XX_XX

注 1： XX 表示读取的数据被丢弃。

2： XX 表示数据未被写入。

3： 在指针发生更新时，中断按需要进行更新。

31.3.10 通道传输行为

使能通道之后（CHEN（DCHxCON<7>）= 1），任何启动单元传输的事件都会传输 CHCSIZ（DCHxCSIZ）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（DCHxSSIZ）或 CHDSIZ（DCHxDSIZ）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）。

31.3.11 通道使能

每个通道都具有使能位 CHEN，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 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 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>）清零时）。
- 在模式匹配模式下发生模式匹配 （仅当 CHAEN 清零时）。
- 发生中止中断。
- 用户写入 CABORT （DCHxECON<6>）标志。

31.3.12 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 IES/IFS 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。

31.3.13 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 位 （DCHxECON<7>）。
- 在通过SIRQEN（DCHxECON<4>）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。

31.3.14 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 31.3.16 节 “通道中止”**中所述中止传输。
- 单元传输完成 （传输了 CHCSIZ （DCHxCSIZ）个字节）。
- DMA 传输了 CHSSIZ 或 CHDSIZ 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>）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 31-4 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 31-4： 通道事件行为

事件	说明和功能	受影响的寄存器
启动传输的事件		
系统中断与 CHSIRQ<7:0> 匹配 (1,2)	通道事件检测位将置 1。	CHEDET = 1
通道链事件	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 (1)	通道事件检测位将置 1。	CHEDET = 1
终止传输的事件		
系统中断与 CHAIRQ<7:0> 匹配 (1,2)	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 (1)	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 IES/IFS 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。

31.3.15 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（CHAIRQ<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>）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 位（DCHxINT<1>）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。

31.3.16 通道中止

用户可以通过写入 CABORT 位（DCHxECON<6>）中止通道传输。在传输被中止时，控制器会完成当前总线事务，剩下的所有事务会被中止。 CHEN（DCHxCON<7>）位将清零。在用户写入 CABORT 位时，源和目标指针会复位。

31.3.17 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 CHERIF（DCHxINT<0>）将置 1。通道将被禁止，即硬件会将 CHEN 复位。

通道状态不会受影响，以便协助对问题进行调试。

31.3.18 DMA 暂停

如果 SUSPEND 位（DMACON<12>）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 BUSY（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：	BUSY 位并非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。

例 31-4： DMA 控制器暂停

```
/*
The following code example will suspend the DMA Controller.
*/
 DMACONSET=0x00001000;       // suspend the DMA controller

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

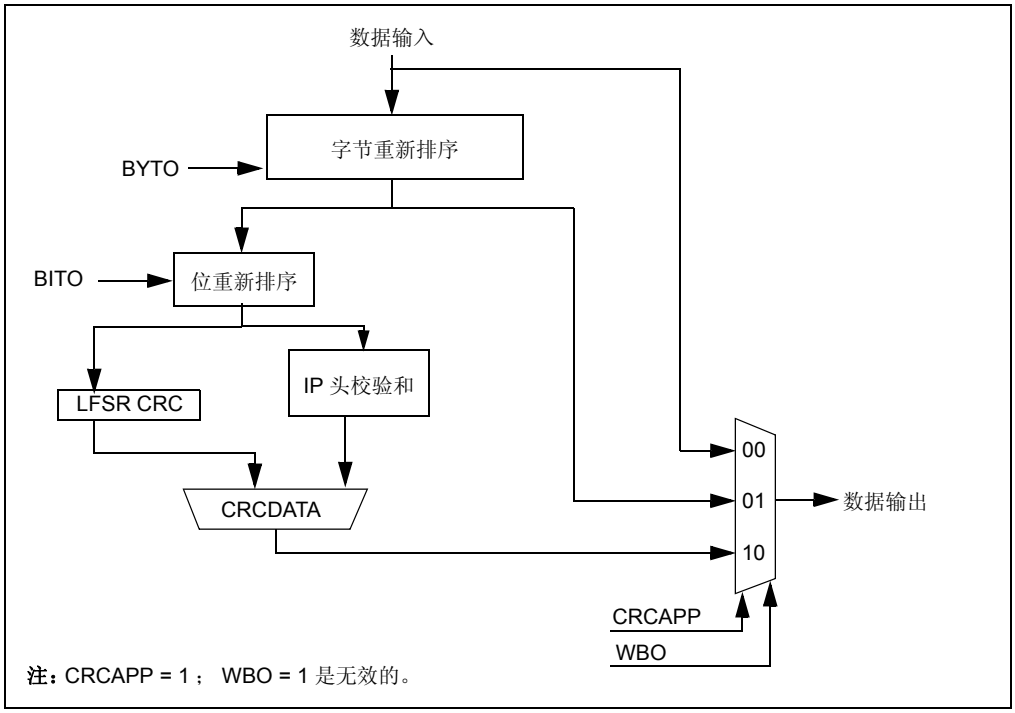
31.3.19 特殊功能模块 （SFM）模式

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 31-5 所示， SFM 具有以下功能块：

- LFSR CRC
- IP 头校验和
- 字节重新排序
- 位重新排序

图 31-5： 特殊功能模块 （SFM）



注： CRCAPP = 1 ； WBO = 1 是无效的。

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 位 （DCRCCON） 。 SFM 通过将 CRCEN 位（DCRCCON<7>）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 位的设置，将数据传递给 LFSR CRC 或 IP 头校验和功能块，如图 31-5 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位（DCRCCON<6>）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 31.3.19.1 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 指定的单元中（见**第 31.3.19.2 节 “CRC 追加模式 （CRCAPP = 1）”**）。

数据写入目标的顺序可以使用 WBO 位 （DRCCON<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> （DRCCON<29:28>）重新排序。

注：	该功能并非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。

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 = 1）时，数据以二进制补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 生成期间的任意时刻读取，但只有在传输完成之后它才有效。

31.3.19.1 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 31-5：　后台模式下 DMA LFSR CRC 计算的代码示例

```
/*
The following 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
 DCHOCSIZ=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)
       {                           // CHERIF (DCHxINT<0>) or CHTAIF (DCHxINT<1> set
              error=TRUE;          // error or aborted...
              break;
       }
              else if (dmaFlags&0x8)
       {                           // CHBCIF (DCHxINT<3>) set
              break;               // transfer completed normally
       }
       pollCnt=100;                // use an adjusted value here
       while(pollCnt--);           // wait before polling again
 }

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

31.3.19.2 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。

该模式最适合用于多个外设需要使用 CRC 发生器的情况。这种情况下，输入数据在器件的缓冲区中进行累加。当缓冲区的输入数据完整时，将会在缓冲区中产生 CRC，并相应地进行使用。由于 DMA 不需要等待多个事件（通常为中断），所以数据块可以在相当短的时间内传输通过 CRC，从而允许 CRC 模块分配给不同通道，或者重定向到不同的数据块。

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 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 31-6：　　追加模式下 CRC 计算的代码示例

```
/*
The following 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
 DCHOCSIZ=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)
       {                          // CHERIF (DCHxINT<0>) or CHTAIF (DCHxINT<1> set
              error=TRUE;         // error or aborted...
              break;
       }
              else if (dmaFlags&0x8)
       {                          // CHBCIF (DCHxINT<3>) set
              break;              // transfer completed normally
       }
       pollCnt=100;               // use an adjusted value here
       while(pollCnt--);          // wait before polling again
 }

 if(error)
 {
                                  // process error
 }

                                  // the block CRC is available in the blockCrc variable
```

31.3.19.3 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。

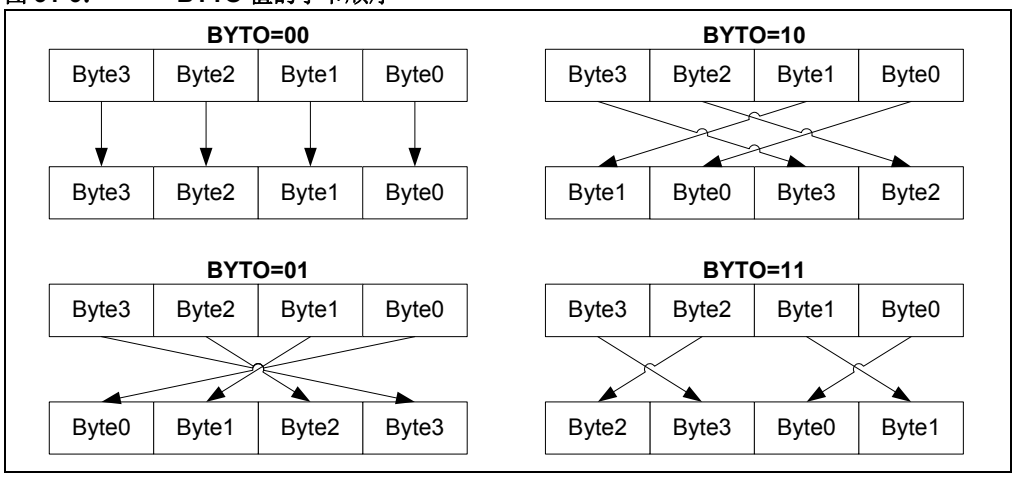
即使用户不使用在 DCRCDATA 中存储的结果，模块也会执行 CRC 计算。

BYTO 控制由模块处理的数据的字节顺序。图 31-6 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 值 01 用于对字中的字节进行重新排序。而 BYTO 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 位更改位顺序。

图 31-6： BYTO 值的字节顺序



31.3.19.4 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：	该功能并非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。

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 位和 PLEN （DCRCCON）位设定。

例 31-7 和例 31-8 给出了 16 位和 32 位 CRC 的多项式。

例 31-7： 16 位 CRC 多项式

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

例 31-8： 32 位 CRC 多项式

$$x^{31} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

要在 CRC 发生器中设定任一多项式，应按下表所示设置 CRC 寄存器位：

表 31-5： CRC 设置示例

CRC 类型	位名称	位值
具有 16 位 CRC 的器件	PLEN<3:0>	b1111
	DCRCXOR<15:0>	b0001 0000 0010 000
具有 32 位 CRC 的器件	PLEN<4:0>	b11111
	DCRCXOR<31:0>	b0000 0100 1100 0001 0001 1101 1011 0110

CRC 发生器中的 PLEN 位（DCRCCON）用于选择哪个位用作 CRC 的反馈点。对于 16 位 CRC 示例，如果 PLEN<3:0> = 0x0110，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 总是进行异或运算。

31.3.19.5 计算 IP 头校验和

当 CRCTYP 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 DCRCCON 寄存器中的 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： 该功能并非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。

31.4 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 位（DCHxINT<16>）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 位（DCHxINT<17>）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 CHAIRQ<7:0>（DCHxECON<23:16>）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 位（DCHxINT<3>）指示，并使用 CHBCIE 位（DCHxINT<19>）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 位（DCHxINT<2>）指示，并使用 CHCCIE 位（DCHxINT<18>）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 位（DCHxINT<7>）指示，并通过 CHSDIE 位（DCHxINT<23>）允许；或者在通道源指针达到源中点位置时产生，通过 CHSHIF 位（DCHxINT<6>）指示，并通过 CHSHIE 位（DCHxINT<22>）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位（DCHxINT<5>）指示，并通过 CHDDIE 位（DCHxINT<21>）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 位（DCHxINT<4>）指示，并通过 CHDHIE 位（DCHxINT<20>）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。

以下是相应的 DMA 通道中断标志：

- DMA0IF（IFS1<16>）
- DMA1IF（IFS1<17>）
- DMA2IF（IFS1<18>）
- DMA3IF（IFS1<19>）
- DMA4IF（IFS1<20>）
- DMA5IF（IFS1<21>）
- DMA6IF（IFS1<22>）
- DMA7IF（IFS1<23>）

所有这些中断标志必须用软件清零。

DMA 通道通过相应的 DMA 中断允许位使能为中断源：

- DMA0IE（IEC1<16>）
- DMA1IE（IEC1<17>）
- DMA2IE（IEC1<18>）
- DMA3IE（IEC1<19>）
- DMA4IE（IEC1<20>）
- DMA5IE（IEC1<21>）
- DMA6IE（IEC1<22>）
- DMA7IE（IEC1<23>）

此外，还必须配置中断优先级位和中断子优先级位：

- DMA0IP<2:0> （IPC9<4:2>）和 DMA0IS<1:0> （IPC9<1:0>）
- DMA1IP<2:0> （IPC9<12:10>）和 DMA1IS<1:0> （IPC9<9:8>）
- DMA2IP<2:0> （IPC9<20:18>）和 DMA2IS<1:0> （IPC9<17:16>）
- DMA3IP<2:0> （IPC9<28:26>）和 DMA3IS<1:0> （IPC9<25:24>）
- DMA4IP<2:0> （IPC10<4:2>）和 DMA4IS<1:0> （IPC10<1:0>）
- DMA5IP<2:0> （IPC10<12:10>）和 DMA5IS<1:0> （IPC10<9:8>）
- DMA7IP<2:0> （IPC10<20:18>）和 DMA6IS<1:0> （IPC10<17:16>）
- DMA7IP<2:0> （IPC10<28:26>）和 DMA7IS<1:0> （IPC10<25:24>）

31.4.1 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。

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。

请注意， DMAxIF 位是否置 1 与相应允许位 DMAxIE 的状态无关。如果需要，可以用软件查询 DMAxIF 位。

DMAxIE 位用于定义在相应 DMAxIF 位置 1 时，向量中断控制器或 INT 的行为。当相应的 DMAxIE 位清零时， INT 模块不会为事件产生 CPU 中断。如果 DMAxIE 位置 1，则 INT 模块会在相应的 DMAxIF 位置 1 时向 CPU 产生中断 （受以下段落中优先级和子优先级制约）。

处理特定中断的用户软件程序负责在服务程序完成之前清零相应的中断标志位。

每个 DMA 通道的优先级可以使用 IPCx 寄存器中的 DMAxIP 位独立进行设置。这些优先级定义了中断源将分配到的优先级组。优先级组值的范围从 7 （最高优先级）到 0 （不产生中断）。较高优先级组中的中断会抢占正在处理、但优先级较低的中断。

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

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

产生允许的中断之后，CPU 将跳转到为该中断分配的向量处。该中断的向量编号与自然顺序编号相同。然后，CPU 将在向量地址处开始执行代码。该向量地址处的用户代码应执行特定于应用程序的操作、清零 DMAxIF 中断标志，然后退出。关于中断的更多信息，请参见 《PIC32MX 系列参考手册》的**第 8 章 “中断”**（DS61108）中的向量地址表详细信息。

表 31-6： 各种偏移量的 DMA 中断向量 （EBASE = 0x8000:0000）

中断	向量 / 自然顺序	IRQ 编号	向量地址 IntCtl.VS = 0x01	向量地址 IntCtl.VS = 0x02	向量地址 IntCtl.VS = 0x04	向量地址 IntCtl.VS = 0x08	向量地址 IntCtl.VS = 0x10
DMA0	36	48	8000 0680	8000 0B00	8000 1400	8000 2600	8000 4A00
DMA1	37	49	8000 06A0	8000 0B40	8000 1480	8000 2700	8000 4C00
DMA2	38	50	8000 06C0	8000 0B80	8000 1500	8000 2800	8000 4E00
DMA3	39	51	8000 06E0	8000 0BC0	8000 1580	8000 2900	8000 5000
DMA4(1)	40	52	8000 0700	8000 0C00	8000 1600	8000 2A00	8000 5200
DMA5(1)	41	53	8000 0720	8000 0C40	8000 1680	8000 2B00	8000 5400
DMA6(1)	42	54	8000 0740	8000 0C80	8000 1700	8000 2C00	8000 5600
DMA7(1)	43	55	8000 0760	8000 0CC0	8000 1780	8000 2D00	8000 5800

注 1： 这些中断并非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。

例 31-9： 允许中断的 DMA 通道初始化代码示例

```
/*
The following 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=0x1d010000;         // transfer source physical address
       DCH0DSA=0x1d020000;         // 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

                                   // do something else

                                   // will get an interrupt when the block transfer is done
                                   // or when error occurred
```

例 31-10： DMA 通道 0 ISR 代码示例

```
/*
The following 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。

31.5 节能和调试模式下的操作

31.5.1 空闲模式下的 DMA 操作

当器件进入 Idle （空闲）模式时，系统时钟源保持工作， DMA 模块继续工作。

在某些器件型号上， SIDL 位 （DMACON<13>）用于选择在 Idle （空闲）模式下模块是停止还是继续工作。

如果 SIDL = 0，则在 Idle （空闲）模式下模块会继续工作，并且会关闭时钟。如果 SIDL = 1，则在 Idle （空闲）模式下模块将停止工作。 DMA 模块将关闭时钟，从而降低功耗。

注：	DMA 不能由 SIDL 位设置为 1 的外设使用。

31.5.2 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。

31.5.3 调试模式下的 DMA 操作

FRZ 位 （DMACON<14>）决定 CPU 在 Debug （调试）模式下执行调试异常代码（即，应用程序暂停）时， DMA 模块是继续运行还是停止。当 FRZ = 0 时，即使应用程序在 Debug （调试）模式下暂停，DMA 模块也会继续运行。当 FRZ = 1 且应用程序在 Debug（调试）模式下暂停时，模块将冻结其操作，并且不会更改 DMA 模块的状态。在 CPU 继续开始执行代码之后，模块将继续工作。

注：	只有 CPU 在调试异常模式下执行时，FRZ 位才可读写。在所有其他模式下，FRZ 位读为 0。如果 FRZ 位在 Debug （调试）模式期间发生改变，则只有退出当前调试异常模式并重新进入该模式之后，新值才会生效。在调试异常模式期间，在进入 Debug （调试）模式时 FRZ 位会读取外设状态。

31.6 各种复位的影响

31.6.1 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

31.6.2 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。

31.6.3 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。

31.7 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 PIC32MX 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 PIC32MX 系列器件的应用笔记和代码示例，请访问 Microchip 网站（www.microchip.com）。

31.8 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 31.3.6 节 “暂停传输”**和**第 31.3.18 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 31.3.19 节 “特殊功能模块 （SFM）模式”**，以区分 16 位和 32 位 CRC
- 增加了**第 31.3.19.5 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 “空闲模式下的 DMA 操作”**

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。

ISBN：978-1-60932-524-4

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

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

全球销售及服务网点

美洲

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

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

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

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

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

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

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

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

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

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

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

亚太地区

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

亚太地区

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

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

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

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

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

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

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

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

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

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

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

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

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

欧洲

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

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

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

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

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

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

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

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

07/15/10