# 目录

[目录 1](#_Toc524700804)

[第16章 Cortex-M4-DMA 2](#_Toc524700805)

[16.1 DMA控制器介绍 2](#_Toc524700806)

[16.1.1 DMA控制器简介 2](#_Toc524700807)

[16.1.2 DMA控制器特征 2](#_Toc524700808)

[16.1.3 DMA控制器原理 3](#_Toc524700809)

[16.1.4 DMA控制器框架 4](#_Toc524700810)

[16.2 DMA控制器功能说明 5](#_Toc524700811)

[16.2.1 通道选择 5](#_Toc524700812)

[16.2.2 仲裁器 6](#_Toc524700813)

[16.2.3 源、目标、传输方向 6](#_Toc524700814)

[16.2.4 直接模式与FIFO模式 6](#_Toc524700815)

[16.2.5 突发增量 7](#_Toc524700816)

[16.2.6 地址递增 8](#_Toc524700817)

[16.2.7 循环模式 8](#_Toc524700818)

[16.3 DMA控制器相关寄存器 8](#_Toc524700819)

[16.3.1 DMA低中断状态寄存器（DMA\_LISR） 8](#_Toc524700820)

[16.3.2 DMA高中断状态寄存器（DMA\_HISR） 8](#_Toc524700821)

[16.3.3 DMA低中断标志清零寄存器（DMA\_LIFCR） 8](#_Toc524700822)

[16.3.4 DMA高中断标志清零寄存器（DMA\_HIFCR） 8](#_Toc524700823)

[16.3.5 DMA 数据流 x 配置寄存器 （DMA\_SxCR）（x = 0..7） 8](#_Toc524700824)

[16.3.6 DMA 数据流 x 数据项数寄存器 （DMA\_SxNDTR）（x = 0..7） 8](#_Toc524700825)

[16.3.7 DMA数据流 x 外设地址寄存器（DMA\_SxPAR）（x = 0..7） 8](#_Toc524700826)

[16.3.8 DMA数据流 x存储器0地址寄存器（DMA\_SxM0AR）（x = 0..7） 8](#_Toc524700827)

[16.3.9 DMA数据流 x FIFO 控制寄存器 （DMA\_SxFCR）（x = 0..7） 8](#_Toc524700828)

[16.4 DMA控制器实验 8](#_Toc524700829)

[16.4.1 硬件分析 8](#_Toc524700830)

[16.4.2 软件设计 8](#_Toc524700831)

# Cortex-M4-DMA

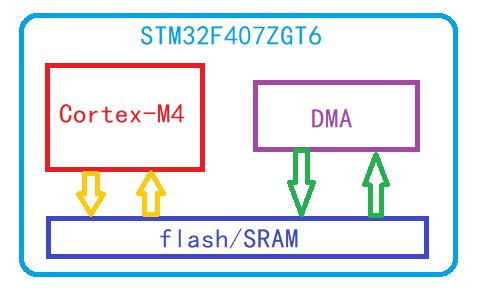
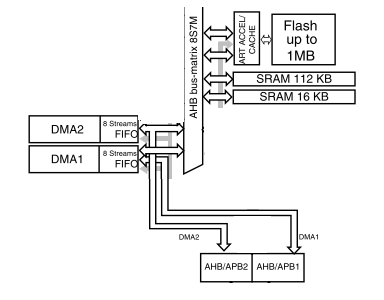
## DMA控制器介绍

### DMA控制器简介

直接存储器访问 (DMA) 用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可 供其它操作使用。

DMA 控制器基于复杂的总线矩阵架构，将功能强大的双 AHB 主总线架构与独立的 FIFO 结合在一起，优化了系统带宽。

两个 DMA 控制器总共有 16 个数据流(数据流：高速公路上的车道)（每个控制器 8 个），每一个 DMA 控制器都用于管理一个或多个外设的存储器访问请求。每个数据流总共可以有多达 8 个通道（或称请求）。每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。



### DMA控制器特征

DMA 主要特性是：

● 双 AHB 主总线架构，一个用于存储器访问，另一个用于外设访问

● 仅支持 32 位访问的 AHB 从编程接口

● 每个 DMA 控制器有 8 个数据流，每个数据流有多达 8 个通道（或称请求）

● 每个数据流有单独的四级32 位先进先出存储器缓冲区 (FIFO)，可用于 FIFO 模式或直接模式：

—FIFO 模式：可通过软件将阈值级别选取为 FIFO 大小的 1/4、1/2 或 3/4

—直接模式 每个 DMA 请求会立即启动对存储器的传输。

当在直接模式（禁止 FIFO）下将 DMA 请求配置为以存储器到外设模式传输数据时，DMA 仅会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。

● 通过硬件可以将每个数据流配置为：

— 支持外设到存储器、存储器到外设和存储器到存储器传输的常规通道

— 也支持在存储器方双缓冲的双缓冲区通道

● 8 个数据流中的每一个都连接到专用硬件 DMA 通道（请求）

● DMA 数据流请求之间的优先级可用软件编程（4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，请求 0 的优先级高于请求 1）

● 每个数据流也支持通过软件触发存储器到存储器的传输 （仅限 DMA2 控制器）

● 可供每个数据流选择的通道请求多达 8 个。此选择可由软件配置，允许几个外设启动 DMA 请求

● 要传输的数据项的数目可以由 DMA 控制器或外设管理：

—DMA 流控制器：要传输的数据项的数目是 1 到 65535，可用软件编程 (存储器到存储器/外设)

—外设流控制器：要传输的数据项的数目未知并由源或目标外设控制，这些外设通过硬件发出传输结束的信号

● 独立的源和目标传输宽度（字节、半字、字）：源和目标的数据宽度不相等时，DMA 自动封装/解封必要的传输数据来优化带宽。这个特性仅在 FIFO 模式下可用。

● 对源和目标的增量或非增量寻址

● 支持 4 个、8 个和 16 个节拍的增量突发传输。突发增量的大小可由软件配置，通常等于外设 FIFO 大小的一半。

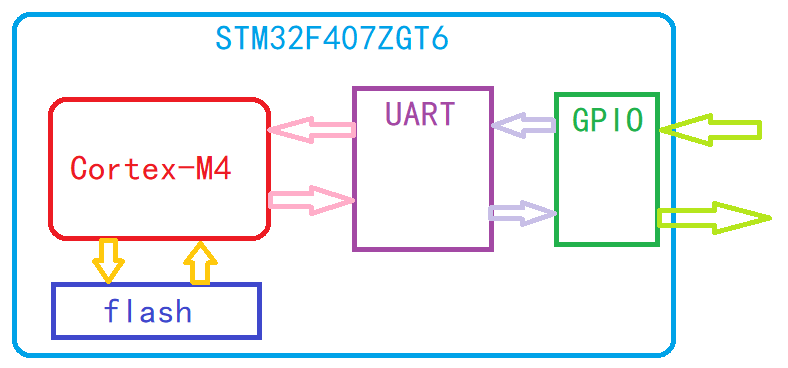
● 每个数据流都支持循环缓冲区管理

● 5 个事件标志（DMA 半传输、DMA 传输完成、DMA 传输错误、DMA FIFO 错误、直接模式错误），进行逻辑或运算，从而产生每个数据流的单个中断请求

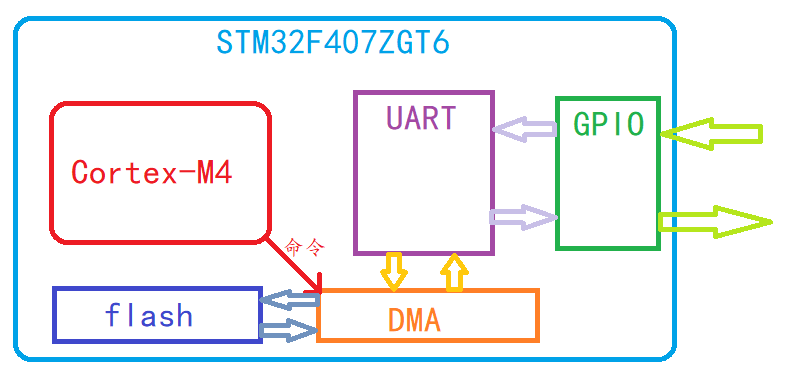
### DMA控制器原理

DMA控制器工作流程。

1. 芯片内部不使用DMA控制器时



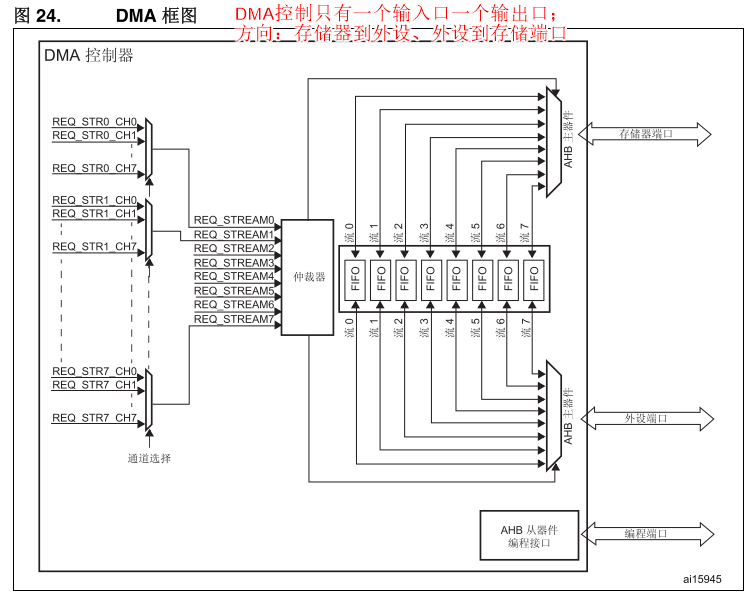
1. 芯片内部使用DMA控制器时

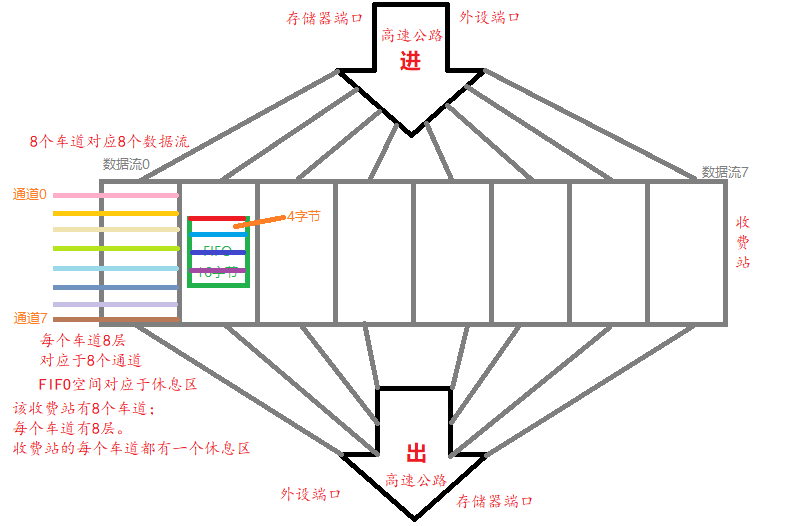


1. CPU要发送相应的命令

命令包含：从哪个位置搬数据、搬移到哪个位置、一次搬移多少字节、要搬移多少次、是否要地址增加。

### DMA控制器框架



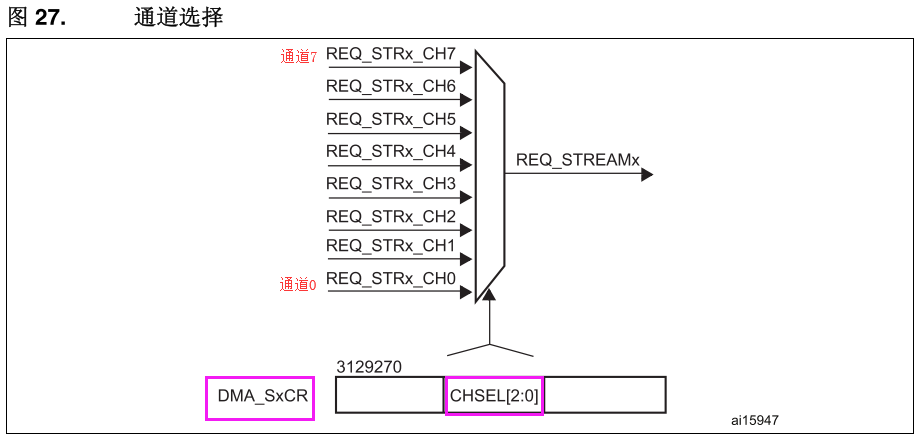


1. 选择方向(存储器到外设、外设到存储器、存储器到存储器)
2. 确定储存器地址或外设地址
3. 确定数据流
4. 确定通道
5. 确定相对应的通道优先级
6. 确定相应的缓冲区是否开启
7. 确定一次搬移的数据量(1个字节、2个字节、4个字节)
8. 确定搬移的次数(DMA流控制：1~65535；外设流：外设控制)
9. 存储器地址是否递增以及外设地址是否需要递增

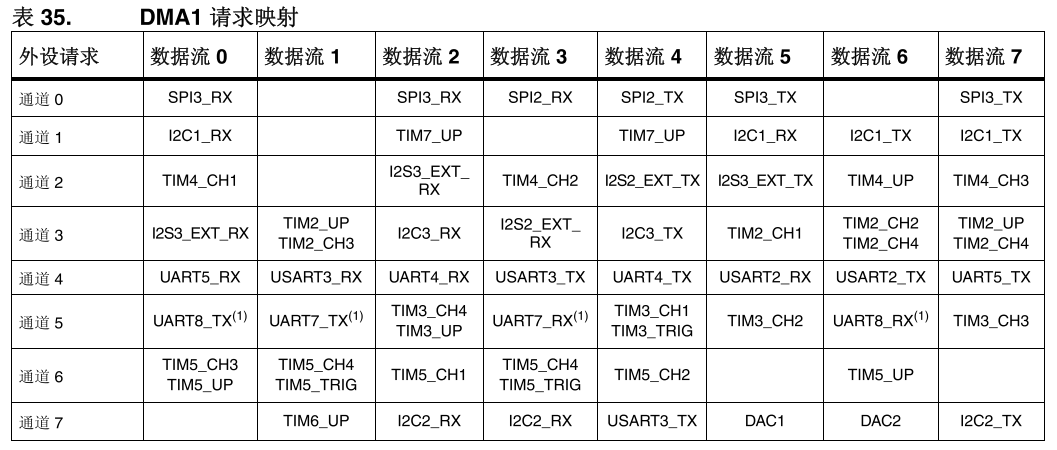
## DMA控制器功能说明

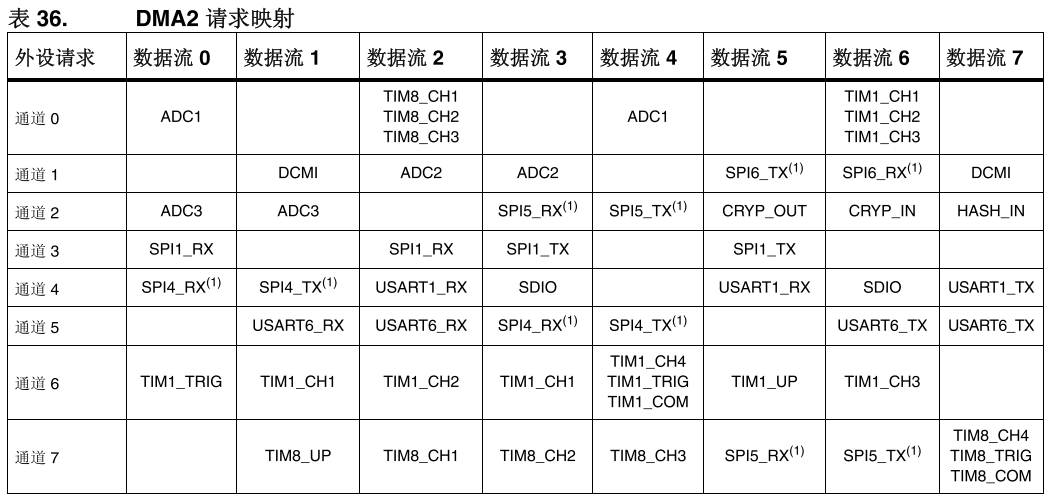
### 通道选择

每个数据流都与一个 DMA 请求相关联，此 DMA 请求可以从 8 个可能的通道请求中选出。此选择由 DMA\_SxCR 寄存器中的CHSEL[2:0] 位控制。



DMA控制器通道的对应关系表：





### 仲裁器

仲裁器为两个 AHB 主端口（存储器和外设端口）提供基于请求优先级的 8 个 DMA 数据流请 求管理，并启动外设/存储器访问序列。

优先级管理分为两个阶段：

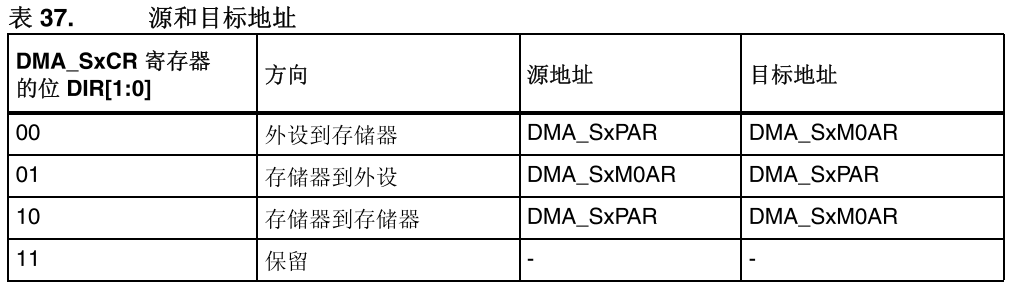
● 软件：每个数据流优先级都可以在 DMA\_SxCR 寄存器中配置。

分为四个级别： — 非常高优先级 — 高优先级 — 中优先级 — 低优先级

● 硬件：如果两个请求具有相同的软件优先级，则编号低的数据流优先于编号高的数据流。例如，数据流 2 的优先级高于数据流 4。

### 源、目标、传输方向

传输方向使用 DMA\_SxCR 寄存器中的 DIR[1:0] 位进行配置，有三种可能的传输方向：存储器到外设、外设到存储器、存储器到存储器。



注意点：存储器到存储器传输过程中，源地址：外设地址寄存器，但是这个外设地址寄存器用来保存存储器地址的结果。

### 直接模式与FIFO模式

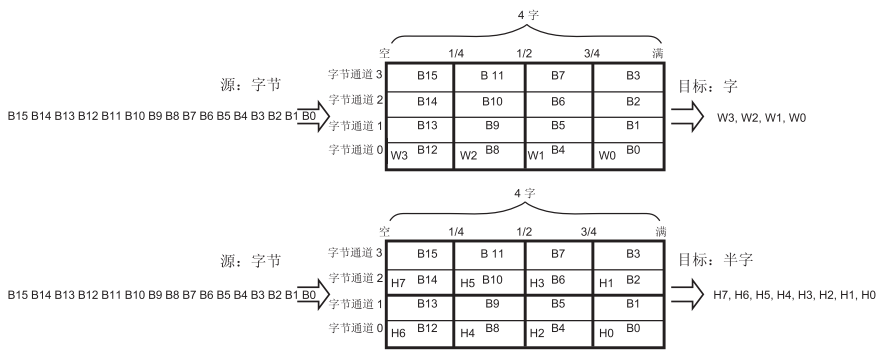
直接模式：从源地址中读取内容，直接发送到目标地址中。

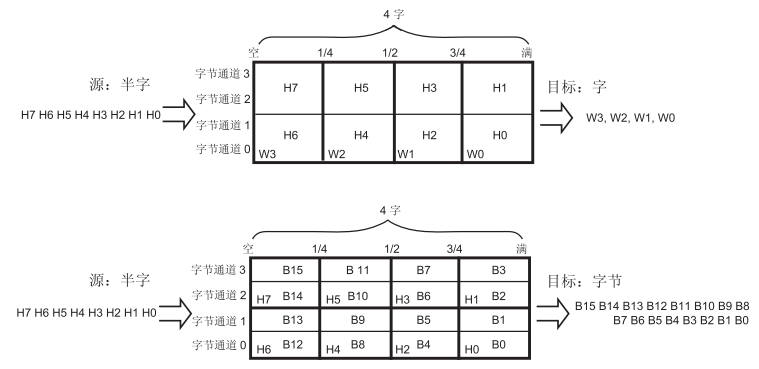
注意点：从源地址取数据的大小(一次多少个字节)要与发送目标地址的数据大小一致。

FIFO模式：从源地址中读取内容，间接发送到目标地址中。(需要进行缓冲)。

注意点：FIFO有专用的缓冲区，并且缓冲区的大小还可以进行设置(分成了四个级别；一个级别有4个字节的存储空间)。

注意点：FIFO模式的时候，源与目标传输大小可以不一致。





### 突发增量



所有这些情况下，突发大小与数据大小的乘积不得超过 FIFO 大小（数据大小可以为：1（字 节）、2（半字）或 4（字））。

### 地址递增

根据 DMA\_SxCR 寄存器中 PINC 和 MINC 位的状态，外设和存储器指针在每次传输后可以自动向后递增或保持常量。

通过单个寄存器访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增模式，则根据在 DMA\_SxCR 寄存器 PSIZE 或 MSIZE 位中编程的数据宽度，下一次传输的地址将是前一次传输的地址递增 1（对于字节）、2（对于半字）或 4（对 于字）。

为了优化封装操作，可以不管 AHB 外设端口上传输的数据的大小，将外设地址的增量偏移大小固定下来。DMA\_SxCR 寄存器中的 PINCOS 位用于将增量偏移大小与外设 AHB 端口或 32 位地址（此时地址递增 4）上的数据大小对齐。PINCOS 位仅对 AHB 外设端口有影响。

如果将 PINCOS 位置 1，则不论 PSIZE 值是多少，下一次传输的地址总是前一次传输的地址递增 4（自动与 32 位地址对齐）。但是，AHB 存储器端口不受此操作影响。

如果 AHB 外设端口或 AHB 存储器端口分别请求突发事务，为了满足 AMBA 协议（在固定地址模式下不允许突发事务），则需要将 PINC 或 MINC 位置 1。

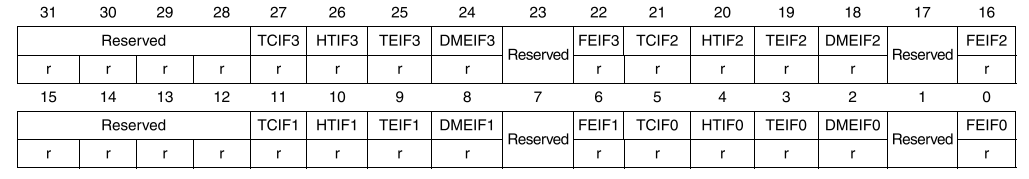
### 循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA\_SxCR寄存器中的 CIRC 位使能此特性。

当激活循环模式时，要传输的数据项的数目在数据流配置阶段自动用设置的初始值进行加载，并继续响应 DMA 请求。

## DMA控制器相关寄存器

### DMA低中断状态寄存器（DMA\_LISR）



**位 31:28、15:12** 保留，必须保持复位值。

**位 27、21、11、5 TCIFx：**数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag) (x = 3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA\_LIFCR 寄存器的相应位。

0：数据流 x 上无传输完成事件

1：数据流 x 上发生传输完成事件

**位 26、20、10、4 HTIFx：**数据流 x 半传输中断标志 (Stream x half transfer interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA\_LIFCR 寄存器的相应位。

0：数据流 x 上无半传输事件

1：数据流 x 上发生半传输事件

**位 25、19、9、3 TEIFx：**数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA\_LIFCR 寄存器的相应位。

0：数据流 x 上无传输错误

1：数据流 x 上发生传输错误

**位 24、18、8、2 DMEIFx：**数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA\_LIFCR 寄存器的相应位。

0：数据流 x 上无直接模式错误

1：数据流 x 上发生直接模式错误

**位 23、17、7、1** 保留，必须保持复位值。

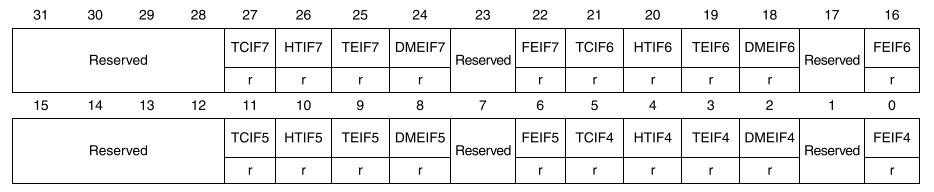
**位 22、16、6、0 FEIFx：**数据流 x FIFO错误中断标志 (Stream x FIFO error interrupt flag) (x=3..0)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA\_LIFCR 寄存器的相应位。

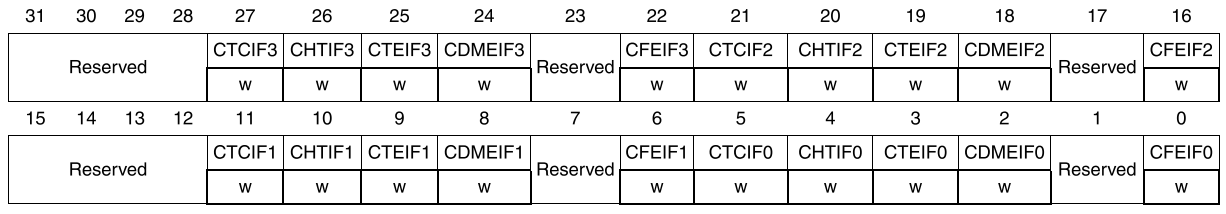
0：数据流 x 上无 FIFO 错误事件

1：数据流 x 上发生 FIFO 错误事件

### DMA高中断状态寄存器（DMA\_HISR）



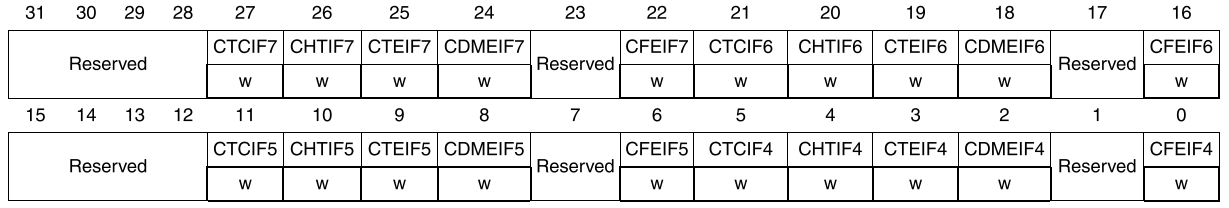
### DMA低中断标志清零寄存器（DMA\_LIFCR）



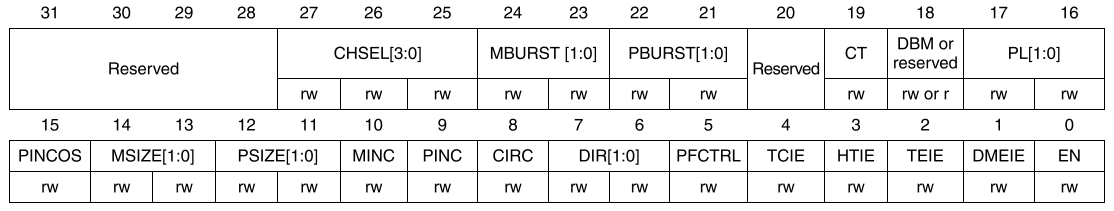
**位 27、21、11、5 CTCIFx：**数据流 x 传输完成中断标志清零 (x = 3..0)

将 1 写入此位时，DMA\_LISR 寄存器中相应的 TCIFx 标志将清零

### DMA高中断标志清零寄存器（DMA\_HIFCR）



### DMA 数据流 x 配置寄存器 （DMA\_SxCR）（x = 0..7）



**位 27:25 CHSEL[2:0]：**通道选择 (Channel selection)

这些位将由软件置 1 和清零。

000：选择通道 0

001：选择通道 1

010：选择通道 2

011：选择通道 3

100：选择通道 4

101：选择通道 5

110：选择通道 6

111：选择通道 7

这些位受到保护，只有 EN 为“0”时才可以写入

**位 24:23 MBURST：**存储器突发传输配置 (Memory burst transfer configuration)

这些位将由软件置 1 和清零。

00：单次传输 (不使用突发增量)

01：INCR4（4 个节拍的增量突发传输）

10：INCR8（8 个节拍的增量突发传输）

11：INCR16（16 个节拍的增量突发传输）

这些位受到保护，只有 EN 为“0”时才可以写入

在直接模式中，当位 EN =“1”时，这些位由硬件强制置为 0x0。

**位 22:21 PBURST[1:0]：**外设突发传输配置 (Peripheral burst transfer configuration)

这些位将由软件置 1 和清零。

00：单次传输

01：INCR4（4 个节拍的增量突发传输）

10：INCR8（8 个节拍的增量突发传输）

11：INCR16（16 个节拍的增量突发传输）

这些位受到保护，只有 EN 为“0”时才可以写入

在直接模式下，这些位由硬件强制置为 0x0。

**位 20 保留**，必须保持复位值。

**位 19 CT：**当前目标（仅在双缓冲区模式下）(Current target (only in double buffer mode))

此位由硬件置 1 和清零，也可由软件写入。(双缓冲的存储器标志位)

0：当前目标存储器为存储器 0（使用 DMA\_SxM0AR 指针寻址）

1：当前目标存储器为存储器 1（使用 DMA\_SxM1AR 指针寻址）

只有 EN 为“0”时，此位才可以写入，以指示第一次传输的目标存储区。

在使能数据流后，此位相当于一个状态标志，用于指示作为当前目标的存储区。

**位 18 DBM：**双缓冲区模式 (Double buffer mode)

此位由软件置 1 和清零。

0：传输结束时不切换缓冲区 (不使用双缓冲)

1：DMA 传输结束时切换目标存储区 (使用双缓冲)

此位受到保护，只有 EN 为“0”时才可以写入。

**位 17:16 PL[1:0]：**优先级 (Priority level)

这些位将由软件置 1 和清零。 (根据实际情况定)

00：低

01：中

10：高

11：非常高

这些位受到保护，只有 EN 为“0”时才可以写入。

**位 15 PINCOS：**外设增量偏移量 (Peripheral increment offset size)

此位由软件置 1 和清零

0：用于计算外设地址的偏移量与 PSIZE 相关

1：用于计算外设地址的偏移量固定为 4（32 位对齐）。

如果位 PINC =“0”，则此位没有意义。

此位受到保护，只有 EN 为“0”时才可以写入。

如果选择直接模式或者 PBURST 不等于“00”，则当使能数据流（位 EN =“1”）时，

此位由硬件强制置为低电平。

**位 14:13 MSIZE[1:0]：**存储器数据大小 (Memory data size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：保留

这些位受到保护，只有 EN 为“0”时才可以写入。

在直接模式下，当位 EN =“1”时，MSIZE 位由硬件强制置为与 PSIZE 相同的值。

**位 12:11 PSIZE[1:0]：**外设数据大小 (Peripheral data size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：保留

这些位受到保护，只有 EN 为“0”时才可以写入

**位 10 MINC：**存储器递增模式 (Memory increment mode)

此位由软件置 1 和清零。

0：存储器地址指针固定

1：每次数据传输后，存储器地址指针递增（增量为 MSIZE 值）

此位受到保护，只有 EN 为“0”时才可以写入。

**位 9 PINC：**外设递增模式 (Peripheral increment mode)

此位由软件置 1 和清零。

0：外设地址指针固定 (在存储器到外设/外设到存储器)

1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值） (在存储器到存储器)

此位受到保护，只有 EN 为“0”时才可以写入。

**位 8 CIRC：**循环模式 (Circular mode)

此位由软件置 1 和清零，并可由硬件清零。

0：禁止循环模式 (通用模式)

1：使能循环模式 (使用模数转换器的扫描功能采用)

如果外设为流控制器（位 PFCTRL=1）且使能数据流（位 EN=1），此位由硬件自动强制清零。

如果 DBM 位置 1，当使能数据流（位 EN =“1”）时，此位由硬件自动强制置 1。

**位 7:6 DIR[1:0]：**数据传输方向 (Data transfer direction)

这些位将由软件置 1 和清零。 (视具体情况定)

00：外设到存储器

01：存储器到外设

10：存储器到存储器

11：保留

这些位受到保护，只有 EN 为“0”时才可以写入。

**位 5 PFCTRL：**外设流控制器 (Peripheral flow controller)

此位由软件置 1 和清零。

0：DMA 是流控制器 (存储器到存储器/存储器到外设)

1：外设是流控制器 (外设到存储器)

此位受到保护，只有 EN 为“0”时才可以写入。

选择存储器到存储器模式（位 DIR[1:0]=10）后，此位由硬件自动强制清零。

**位 4 TCIE：**传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止 TC 中断

1：使能 TC 中断

**位 3 HTIE：**半传输中断使能 (Half transfer interrupt enable)

此位由软件置 1 和清零。

0：禁止 HT 中断

1：使能 HT 中断

**位 2 TEIE：**传输错误中断使能 (Transfer error interrupt enable)

此位由软件置 1 和清零。

0：禁止 TE 中断

1：使能 TE 中断

**位 1 DMEIE：**直接模式错误中断使能 (Direct mode error interrupt enable)

此位由软件置 1 和清零。

0：禁止 DME 中断

1：使能 DME 中断

**位 0 EN：**数据流使能/读作低电平时数据流就绪标志 (Stream enable / flag stream ready when read low)

此位由软件置 1 和清零。

0：禁止数据流

1：使能数据流

以下情况下，此位可由硬件清零：

—DMA 传输结束时（准备好配置数据流）

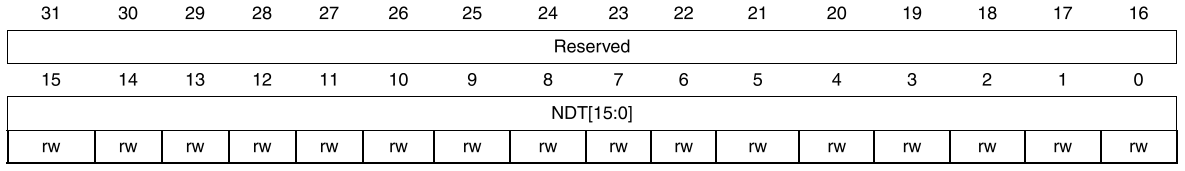
—AHB 主总线出现传输错误时

— 存储器 AHB 端口上的 FIFO 阈值与突发大小不兼容时

此位读作 0 时，软件可以对配置和 FIFO 位寄存器编程。

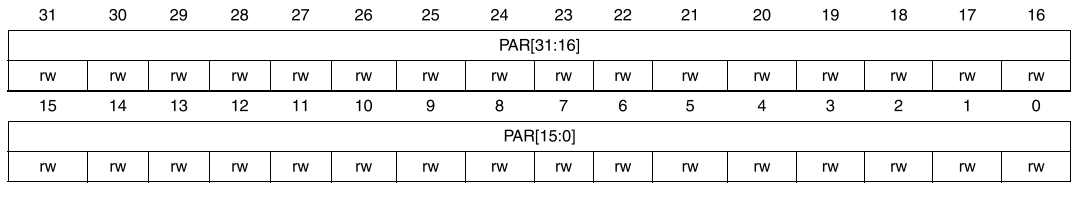
EN 位读作 1 时，禁止向这些寄存 器执行写操作。

### DMA 数据流 x 数据项数寄存器 （DMA\_SxNDTR）（x = 0..7）



搬移多少次。这个寄存器的使用必须是在DMA是流控制器时。

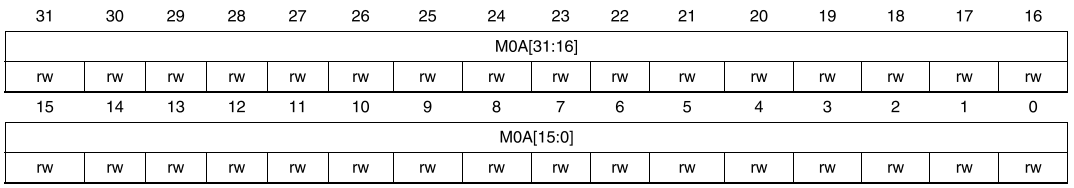
### DMA数据流 x 外设地址寄存器（DMA\_SxPAR）（x = 0..7）



例如：存储UART的DR寄存器。

DMA\_SxPAR = （U32）&(UART->DR)；

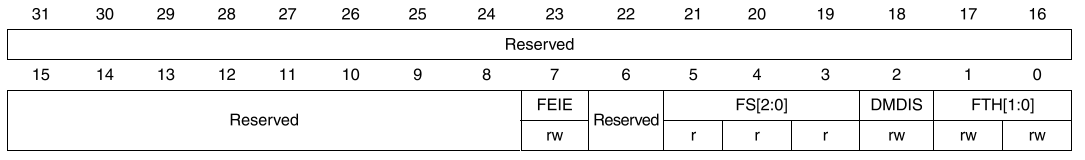
### DMA数据流 x存储器0地址寄存器（DMA\_SxM0AR）（x = 0..7）



例如：存储器u8 data\_buff[100]。

DMA\_SxM0AR = (u32)data\_buff;

### DMA数据流 x FIFO 控制寄存器 （DMA\_SxFCR）（x = 0..7）



FIFO模式下进行FIFO的配置。

## DMA控制器实验

实验对象：利用USART1\_TX将MCU中的数据发送到PC中。

### 硬件分析



PA9中

PA9配置：模式复用；复用类型：USART1。

### 软件设计

1. 初始化USART1的IO口
   1. 开时钟
   2. 配置模式
   3. 配置上下拉
   4. 配置复用类型
2. 初始化USART1(必须保证在查询模式能够正常工作)
   1. 开时钟
   2. 配置波特率
   3. 配置控制寄存器1(使能发送器、使能接收器)
   4. 配置控制寄存器2(配置停止位长度为1)
   5. 使能USART1
3. 初始化DMA控制器
   1. 开时钟
   2. 关闭数据流
   3. 等待数据流确定关闭
   4. 配置配置寄存器(选择通道、突发传输配置、双缓冲区、优先级配置、地址增量设置、传输大小、循环模式、传输方向)
   5. 配置直接模式、FIFO模式
   6. 开启对应外设的DMA使能(存储器到外设&外设到存储器)
   7. 配置外设地址
   8. 配置存储器地址
   9. 配置数据项数目(DMA流控制(存储器到外设&存储器到存储器))
   10. 开启一次数据流
4. 数据流控制函数
   1. 配置外设地址
   2. 配置存储器地址
   3. 配置数据项数目(DMA流控制(存储器到外设&存储器到存储器))
   4. 开启一次数据流