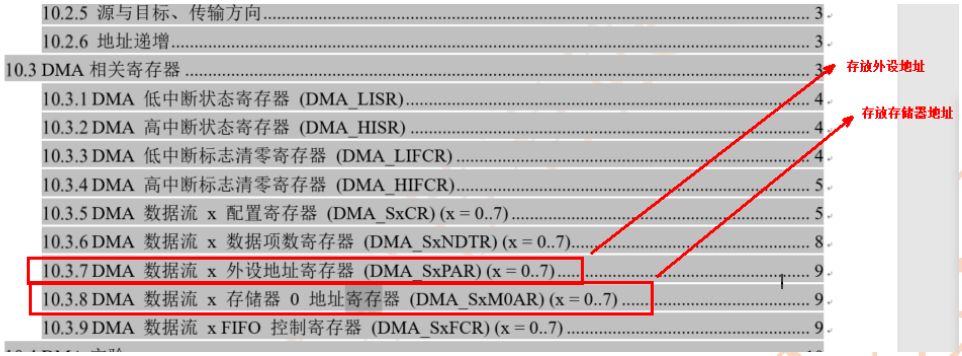
目录

第10章 Cortex-M4-DMA

10.1 DMA 概述



10.1.1 何谓 DMA

直接存储器访问。→直接存储访问控制器

串口发送字符串

```
Void Uart_Send_Data(u8* data)
{
    While(*data)
    {
        while( !(USART1->SR & (0x1 << 7)) );等待发送缓冲区为空//死等
        UART1->DR=*data++;
    }
}
```

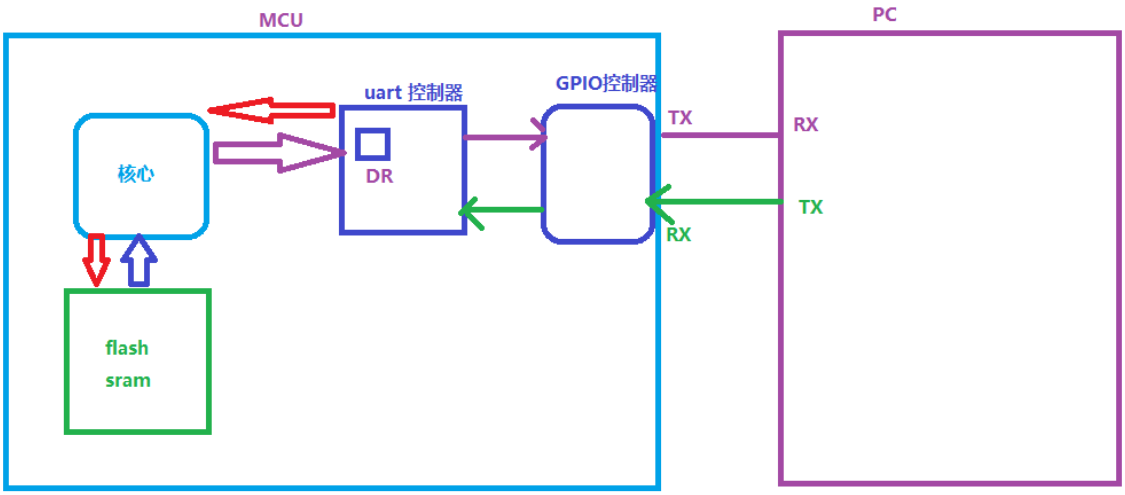
上面的代码存在让 CPU 死等的情况，导致 CPU 效率低下。另外可以采用中断的方法，可以提高 CPU 效率，但是尽管使用中断，仍然是需要 CPU 参与，造成 CPU 频繁切换，这样 CPU 的效率仍然不是最高。有什么方法可以在这样大量数据的传输中解放 CPU，让 CPU 的效率达到最高？-------采用 DMA 传输

DMA 就是芯片上一个功能模块，它的唯一能力就是搬数据。使用它之前需要设置，只要告诉它一共有多少个数据，从哪里搬到哪里，一次搬多少个，让后开启它即可，这个数据搬运的过程就完全不需要 CPU 参与。

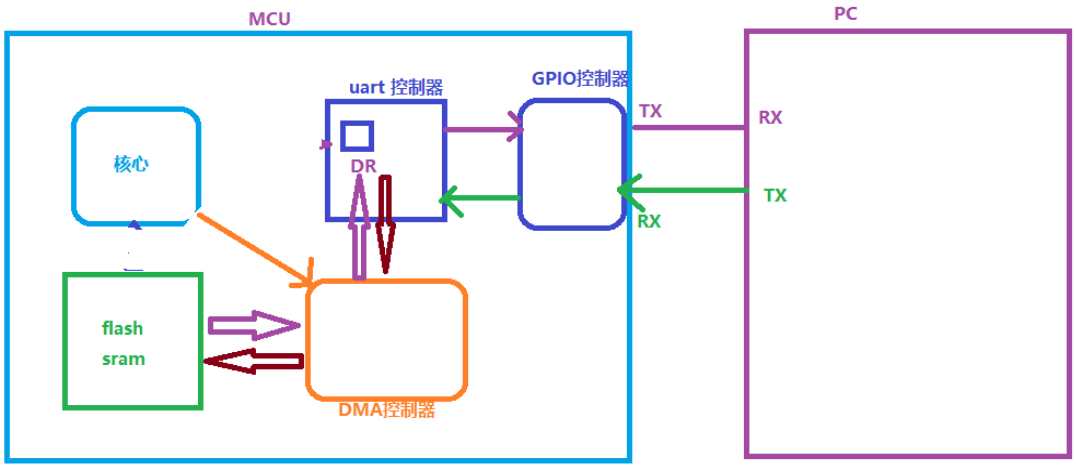
DMA 主要就是做大量数据搬运，简单而又频繁的工作。----DMA 主要功能提高 CPU 的运行效率。

10.1.2 DMA 工作流程

1. 没有 DMA 参与的 uart 的收发过程



2. 有 DMA 参与的 uart 的收发过程



10.1.3 STM32DMA 控制器概述

外设：片上外设(核心外芯片内 UART TIM….)
存储器：FLASH、Sram、FSMC(外扩存储器)
定义数组变量—>将会在内存中开辟一个地址。

DMA 搬运的方向只有三种(使用之前要选择好)

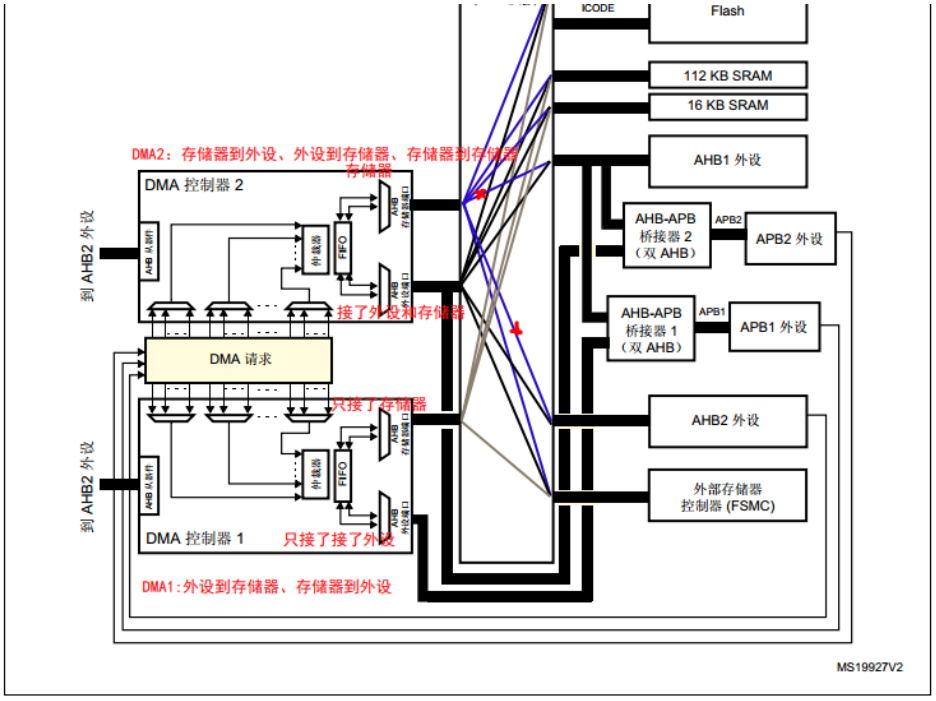
1. 外设到存储器----（串口发送数据到内存中---内存接收数据）
2. 存储器到外设-----（串口接收数据也就是数据有内存--发送的串口数据寄存器里面）
3. 存储器到存储器----（定义内存里面的两个变量数组—实现数据的相互传输）

DMA1 只能实现外设到存储器和存储器到外设

问：uart 发送数据，设置 DMA 的搬运方向？ 存储器到外设
uart 接收数据，设置 DMA 的搬运方向？ 外设到存储器

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。（USB3.0 和 USB2.0）



1. DMA1 控制器 AHB 外设端口与 DMA2 控制器的情况不同，不连接到总线矩阵，因此，仅 DMA2 数据流能够执行存储器到存储器的传输。

两个 DMA 控制器总共有 16 个数据流（每个控制器 8 个数据流），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。（通过优先级来判断，谁先通过）。

1 个 DMA 控制器有 8 个数据流，每个数据流有 8 个通道 -→64 个通道

STM32 单片机一共有 128 个通道(每一个通道都硬件连接起来)

所以你使用外设到存储器或者存储器到外设 都要查看通道映射表-→像之前学习复用的映射表（不一样）

表 35. DMA1 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
通道 1	I2C1_RX		TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
通道 2	TIM4_CH1		I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
通道 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
通道 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
通道 5	UART8_TX(1)	UART7_TX(1)	TIM3_CH4 TIM3_UP	UART7_RX(1)	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX(1)	TIM3_CH3
通道 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2		TIM5_UP	
通道 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

表 36. DMA2 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
通道 1		DCMI	ADC2	ADC2		SPI6_TX(1)	SPI6_RX(1)	DCMI
通道 2	ADC3	ADC3		SPI5_RX(1)	SPI5_TX(1)	CRYP_OUT	CRYP_IN	HASH_IN
通道 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
通道 4	SPI4_RX(1)	SPI4_TX(1)	USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
通道 5		USART6_RX	USART6_RX	SPI4_RX(1)	SPI4_TX(1)		USART6_TX	USART6_TX
通道 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
通道 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX(1)	SPI5_TX(1)	TIM8_CH4 TIM8_TRIG TIM8_COM

表 36. DMA2 请求映射

数据流不一样

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
通道 1		DCMI	ADC2	ADC2		SPI6_TX(1)	SPI6_RX(1)	DCMI
通道 2	ADC3	ADC3		SPI5_RX(1)	SPI5_TX(1)	CRYP_OUT	CRYP_IN	HASH_IN
通道 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
通道 4	[illegible]	[illegible]	[illegible]	SDIO		USART1_RX	SDIO	USART1_TX
通道 5		USART6_RX	USART6_RX	SPI4_RX(1)	SPI4_TX(1)		USART6_TX	USART6_TX
通道 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
通道 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX(1)	SPI5_TX(1)	TIM8_CH4 TIM8_TRIG TIM8_COM

中断请求(相应的标志位置一，就会向 NVIC 发送中断请求---进入中断服务函数进行相应的应用操作，NVIC 就会帮你抢 CPU)

DMA 请求(跟中断请求类似，相应的标志位置一，就会向 DMA 控制器发送 DMA 请求，DMA 就会帮你搬运数据)

DMA 通道(固定每个外设连到哪个 DMA 控制器的哪个数据流上)(肯定有表格给你查看)—根据映射表对应起来。

例如：串口 1 发送数据(存储器到外设) 找 uart1 发送数据的通道

就要根据映射表，我们发现 DMA2 的数据流 7 的通道 4 作为串口 1 发送通道

10.1.4 STM32DMA 控制器特征

DMA 主要特性是：-→可以对应到中文数据手册

- 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

  DMA1：外设到存储器 存储器到外设

  DMA2：外设到存储器 存储器到外设 --→存储器到存储器（实现内存中数据的相互传输）
- 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）（相当于 8 条高速公路—每条高速公路对应 8 个车道，下高速一次只允许一条车道上的车通过→仲裁器决定的通过判断车道的优先级）
- 每个数据流有单独的四级 32 位(16 字节)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(4)、 1/2(8) 或 3/4(12)，满(16)

— 直接模式（不使用 FIFO）--只要不使用 FIFO 模式，默认都是直接模式。

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(DMA2)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：1、非常高、2、高、3、中、4、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，数据流 0 的优先级高于数据流 1）--（就像学习中断中的自然优先级）--相当设计厂家固定的优先级。---解决谁先通过的问题
- 每个数据流也支持通过（使能数据流）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：数据项的数目：搬运的次数

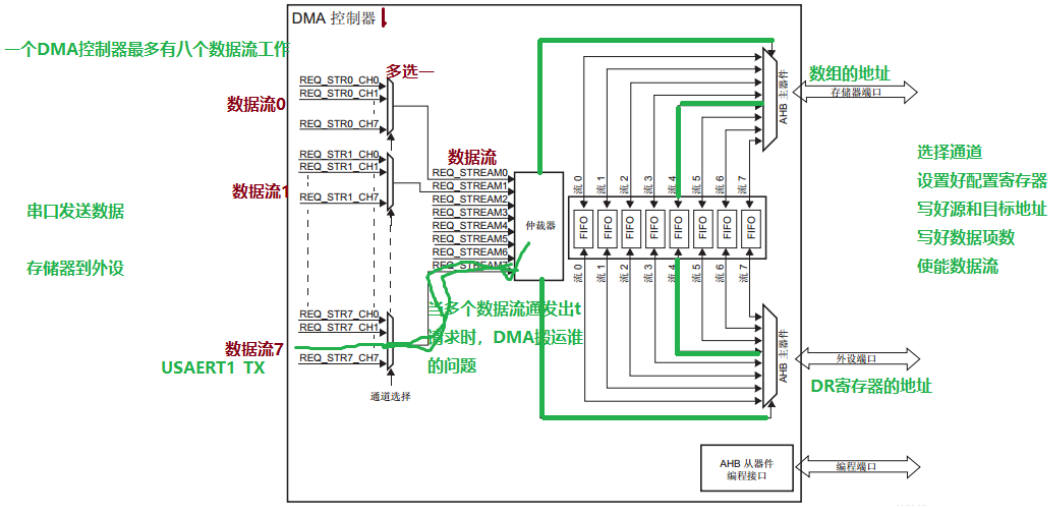
— DMA 流控制器：要传输的数据项的数目是 1 到 65535-→占用的位有限，可用软件编程(数据项数目已知)

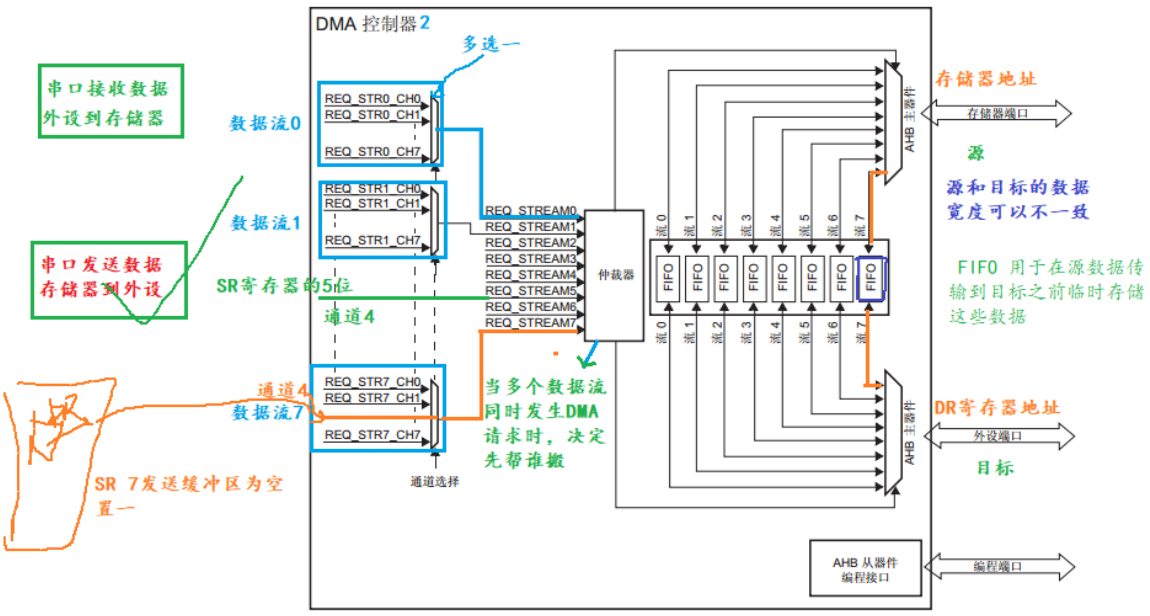
— 外设流控制器：要传输的数据项的数目未知并由源或目标外设控制，这些外设通过硬件发出传输结束的信号

- 独立的源和目标传输宽度（字节、半字(2 个字节)、字(4 个字节)）：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

10.1.5 STM32DMA 控制器框架





一个 DMA 最多有多少个数据流同时工作？ 8 个数据流可以同时工作

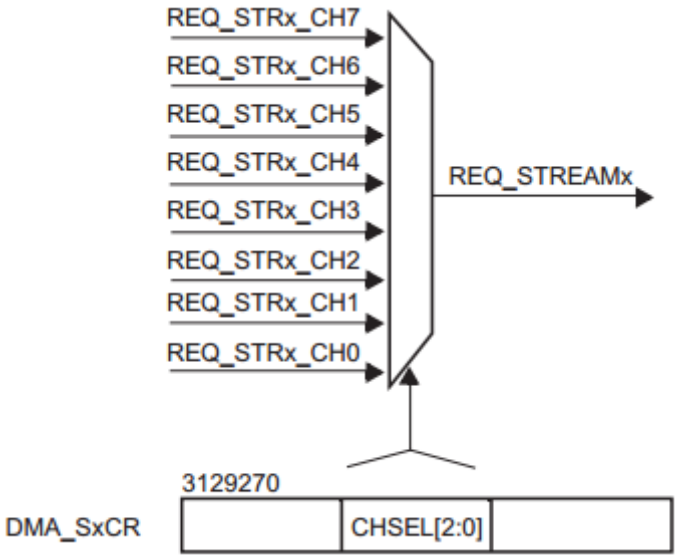
一个数据流最多有多少个通道同时工作？ 1 个通道---单个数据流同一时刻只允许一条通道流过。

两个 DMA 控制器可以同时工作吗？ 可以

10.2 DMA 功能

10.2.1 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。



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

每个数据流的每个通道硬件都已经帮你连经好了，所以要想实现外设到存储器或者存储器到外设需要查看映射表

存储器到到存储器不需要查看映射表，所以只要使用 DMA2 的任意一个通道即可

表 35. DMA1 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	SPI3_RX		SPI3_RX	SPI2_RX	SPI2_TX	SPI3_TX		SPI3_TX
通道 1	I2C1_RX		TIM7_UP		TIM7_UP	I2C1_RX	I2C1_TX	I2C1_TX
通道 2	TIM4_CH1		I2S3_EXT_RX	TIM4_CH2	I2S2_EXT_TX	I2S3_EXT_TX	TIM4_UP	TIM4_CH3
通道 3	I2S3_EXT_RX	TIM2_UP TIM2_CH3	I2C3_RX	I2S2_EXT_RX	I2C3_TX	TIM2_CH1	TIM2_CH2 TIM2_CH4	TIM2_UP TIM2_CH4
通道 4	UART5_RX	USART3_RX	UART4_RX	USART3_TX	UART4_TX	USART2_RX	USART2_TX	UART5_TX
通道 5	UART8_TX(1)	UART7_TX(1)	TIM3_CH4 TIM3_UP	UART7_RX(1)	TIM3_CH1 TIM3_TRIG	TIM3_CH2	UART8_RX(1)	TIM3_CH3
通道 6	TIM5_CH3 TIM5_UP	TIM5_CH4 TIM5_TRIG	TIM5_CH1	TIM5_CH4 TIM5_TRIG	TIM5_CH2		TIM5_UP	
通道 7		TIM6_UP	I2C2_RX	I2C2_RX	USART3_TX	DAC1	DAC2	I2C2_TX

表 36. DMA2 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
通道 1		DCMI	ADC2	ADC2		SPI6_TX(1)	SPI6_RX(1)	DCMI
通道 2	ADC3	ADC3		SPI5_RX(1)	SPI5_TX(1)	CRYP_OUT	CRYP_IN	HASH_IN
通道 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
通道 4	SPI4_RX(1)	SPI4_TX(1)	USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
通道 5		USART6_RX	USART6_RX	SPI4_RX(1)	SPI4_TX(1)		USART6_TX	USART6_TX
通道 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
通道 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX(1)	SPI5_TX(1)	TIM8_CH4 TIM8_TRIG TIM8_COM

1. 这些请求在 STM32F42xxx 和 STM32F43xxx 上可用。

只有 DMA2 才能实现存储器到存储器传输

10.2.2 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。

位 17:16 **PL[1:0]**：优先级 (Priority level)
这些位将由软件置 1 和清零。
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。

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：如果两个数据流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。----建立在软件优先级相同的情况下

软件优先级相当于中断的响应优先级，不存在抢占现象。硬件优先级就相当于自然优先级

10.2.3 FIFO 模式与直接模式

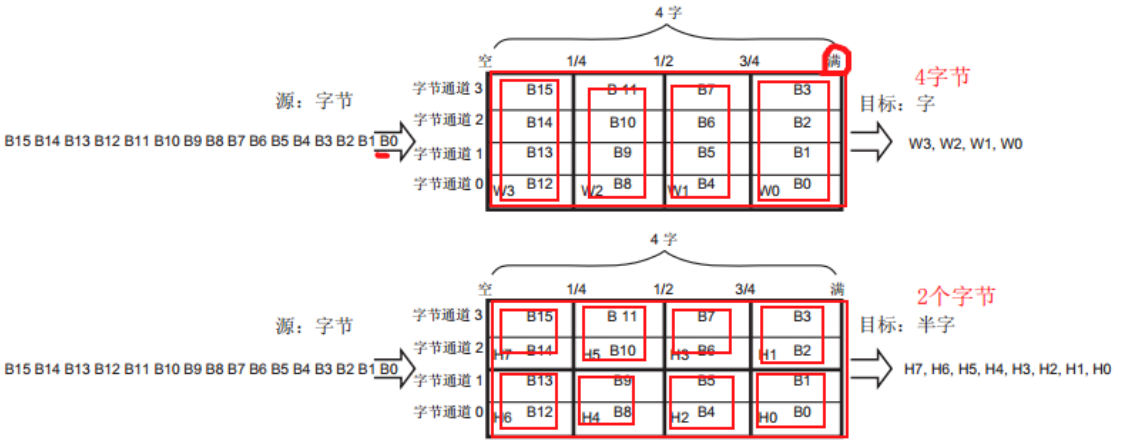
1. FIFO 模式

FIFO 用于在源数据传输到目标之前临时存储这些数据。源和目标的传输的数据宽度可以不一致

每个数据流都有一个独立的 4 字 FIFO(16 字节)，阈值级别可由软件配置为 1/4、 1/2、 3/4 或满。

为了使能 FIFO 阈值级别，必须通过将 DMA_SxFCR 寄存器中的 DMDIS 位置 1 来禁止直接模式。

（禁止直接模式相当于使能 FIFO 模式）



2. 直接模式

默认情况下， FIFO 以直接模式操作（将 DMA_SxFCR 中的 DMDIS 位置 0），不使用 FIFO 阈值级别。如果在每次 DMA 请求之后，系统需要至/自存储器的立即和单独传输，这种模式非常有用。当在直接模式（禁止 FIFO）下将 DMA 配置为以存储器到外设模式传输数据时， DMA 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。

该模式仅限以下方式的传输：

● 源和目标传输宽度相等，并均由 DMA_SxCR 中的 PSIZE[1:0] 位定义（MSIZE[1:0] 位的状态是“无关”）

● 不可能进行突发传输(FIFO 缓冲区)（DMA_SxCR 中的 PBURST[1:0] 和 MBURST[1:0] 位的状态是“无
关”）

当实现存储器到存储器传输时不得使用直接模式。(FIFO 模式)

1. 只能用 DMA2(任意一个通道)
2. 并且还在 FIFO 模式

10.2.4 突发增量(仅在 FIFO 模式才能使用)

提高数据的速度和保证数据的完整性

表 41. FIFO 阈值配置 4节拍 8节拍 16节拍

MSIZE	FIFO 级别	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
字节 1	1/4 4	4 个节拍的 1 次突发	禁止	禁止
	1/2 8	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
	3/4 12	4 个节拍的 3 次突发	禁止	
	满 16	4 个节拍的 4 次突发	8 个节拍的 2 次突发	16 个节拍的 1 次突发
半字 2	1/4 4	禁止	禁止	禁止
	1/2 8	4 个节拍的 1 次突发		
	3/4 12	禁止		
	满 16	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
字 4	1/4 4	禁止	禁止	
	1/2 8			
	3/4 12			
	满 16	4 个节拍的 1 次突发		

一次突发传输的字节数
等于 节拍数*MSIZE

10.2.5 源与目标、传输方向

表 37. 源和目标地址

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR
01	存储器到外设	DMA_SxM0AR	DMA_SxPAR
10	存储器到存储器	DMA_SxPAR	DMA_SxM0AR
11	保留	-	-

10.2.5 源与目标、传输方向

表 37. 源和目标地址

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR
01	存储器到外设	DMA_SxM0AR	DMA_SxPAR
10	存储器到存储器	DMA_SxPAR	DMA_SxM0AR
11	保留	-	-

10.2.5.1 外设到存储器模式

在 FIFO 模式（将 DMA_SxCR 寄存器中的位 EN 置 1(使能数据流)） 时，每次产生外设请求，数据流都会启动数据源到 FIFO 的传输。

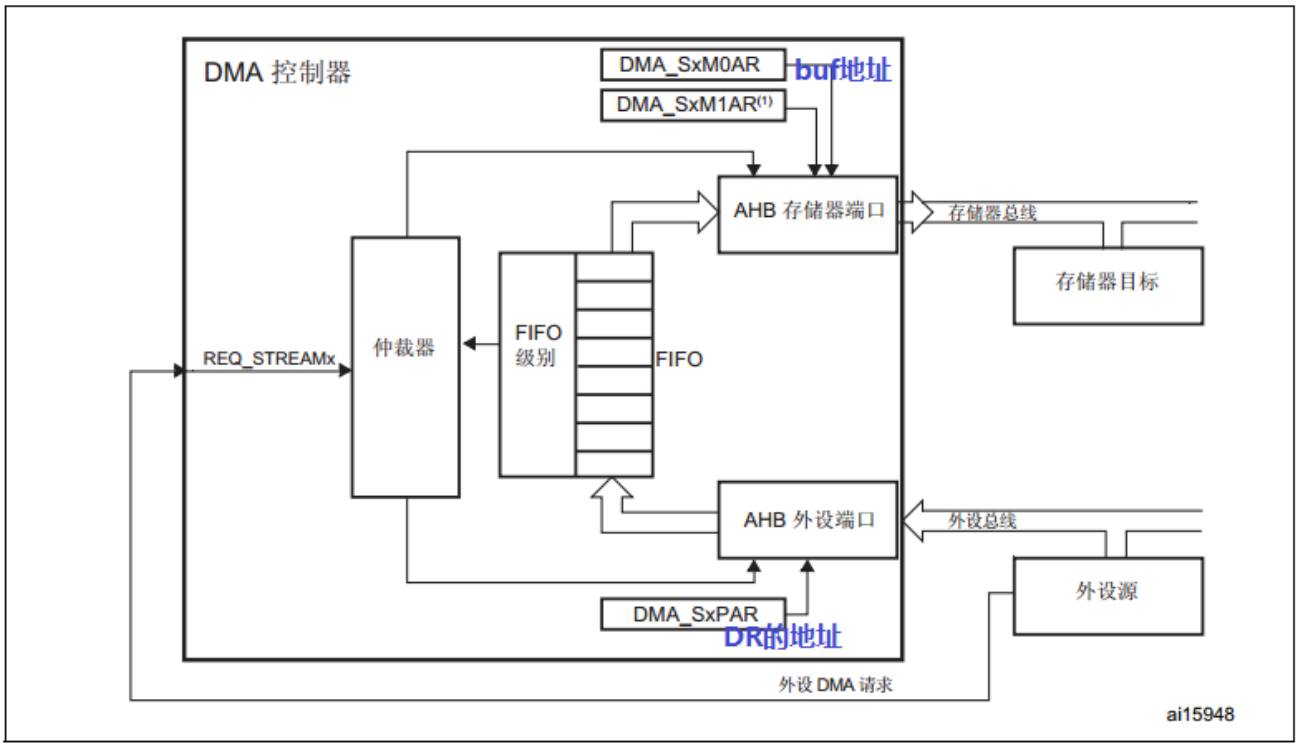
达到 FIFO 的阈值级别时， FIFO 的内容移出并存储到目标中。

如果 DMA_SxNDTR(数据项) 寄存器达到零、外设请求传输终止(SDIO)（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止(停止传输的三种条件)。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别控制：每完成一次从外设到 FIFO 的数据传输后，相应的数据立即就会移出并存储到目标中。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 28. 外设到存储器模式 串口接收数据



10.2.5.2 存储器到外设模式

在 FIFO 模式，使能这种模式（将 DMA_SxCR 寄存器中的 EN 位置 1）时，数据流会立即启动传输，从源完全填充 FIFO。

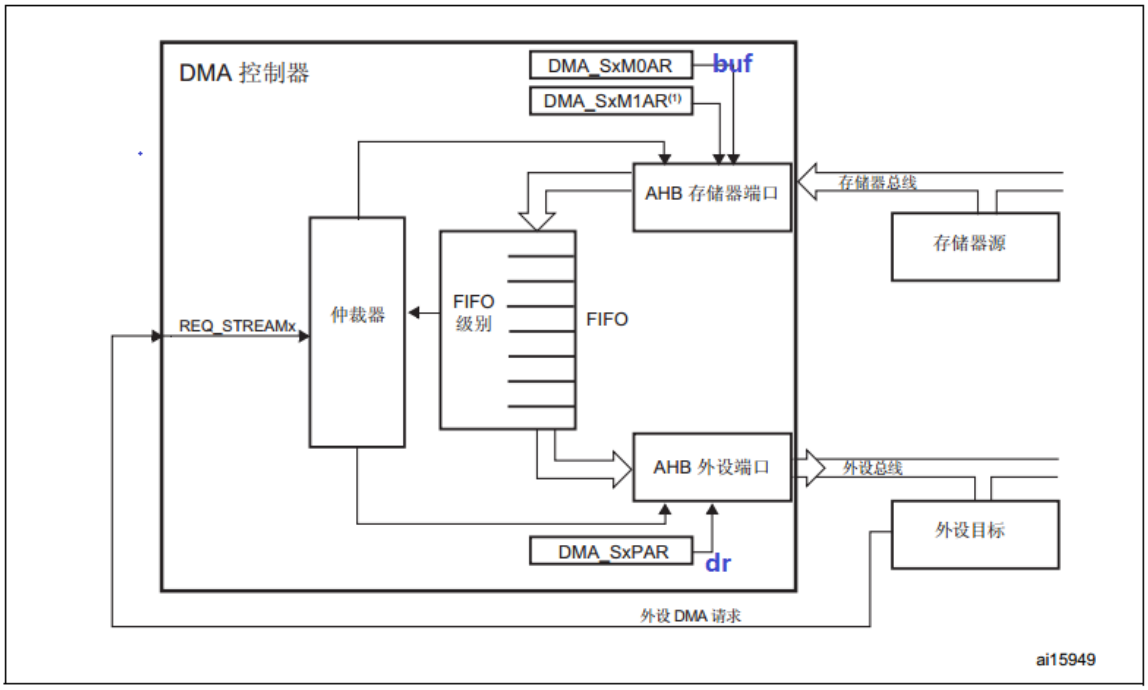
每次发生外设请求， FIFO 的内容都会移出并存储到目标中 。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止(三种停止传输的方式)。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流， DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输， DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位字段的值。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 29. 存储器到外设模式 串口发送数据(buf --> DR)



10.2.5.3 存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此为图 30 中介绍的存储器到存储器模式。

通过将 DMA_SxCR 寄存器中的使能位 (EN) 置 1 来使能数据流时，数据流会立即开始填充 FIFO，直至达到阈值级别。达到阈值级别后， FIFO 的内容便会移出，并存储到目标中。

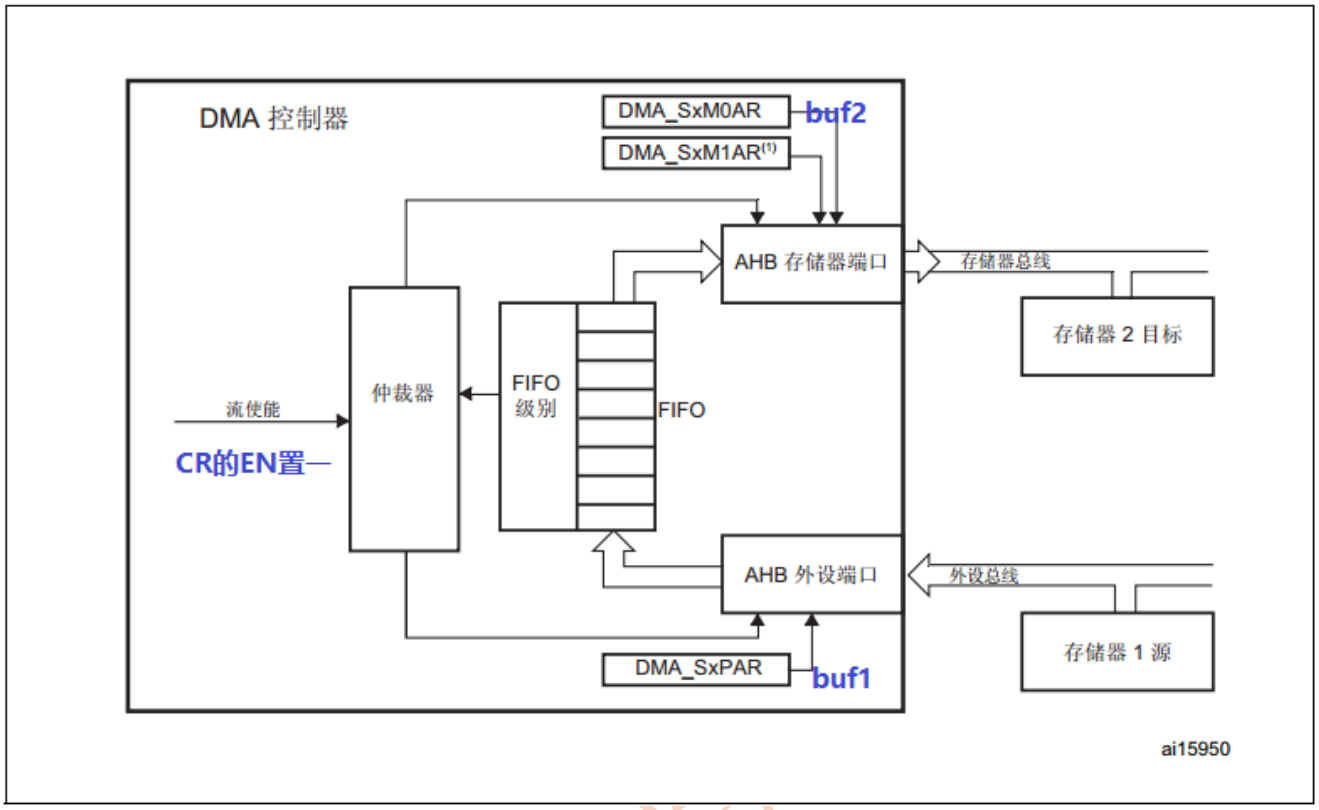
如果 DMA_SxNDTR 寄存器达到零或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

注意： 使用存储器到存储器模式时，不允许循环模式(DMA 搬运 ADC 的结果)和直接模式。

只有 DMA2 控制器能够执行存储器到存储器的传输。

图 30. 存储器到存储器模式 实例:从BUF1 --> buf2



10.2.6 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→决定数据宽度，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。

10.3 DMA 相关寄存器

10.3.1 DMA 低中断状态寄存器 (DMA_LISR)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF3	HTIF3	TEIF3	DMEIF3	Reserved	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Reserved	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TCIF1	HTIF1	TEIF1	DMEIF1	Reserved	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Reserved	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

10.3.2 DMA 高中断状态寄存器 (DMA_HISR)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TCIF7	HTIF7	TEIF7	DMEIF7	Reserved	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Reserved	FEIF6
				r	r	r	r		r	r	r	r	r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				TCIF5	HTIF5	TEIF5	DMEIF5	Reserved	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Reserved	FEIF4
				r	r	r	r		r	r	r	r	r		r

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=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 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=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 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=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 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=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 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=7..4)

此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 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

10.3.3 DMA 低中断标志清零寄存器 (DMA_LIFCR)

用于清除低中断状态寄存器的对应位，写 1 清除

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Reserved	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Reserved	CFEIF2
				w	w	w	w		w	w	w	w	w		w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Reserved	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Reserved	CFEIF0
				w	w	w	w		w	w	w	w	w		w

10.3.4 DMA 高中断标志清零寄存器 (DMA_HIFCR)

用于清除高中断状态寄存器的对应位，写 1 清除

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Reserved	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Reserved	CFEIF6
				w	w	w	w		w	w	w	w	w		w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Reserved	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Reserved	CFEIF4
				w	w	w	w		w	w	w	w	w		w

10.3.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) (x = 0..7)

偏移地址：0x10 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CHSEL[3:0]			MBURST [1:0]		PBURST[1:0]		Reserved	CT	DBM or reserved	PL[1:0]	
				rw	rw	rw	rw	rw	rw	rw		rw	rw or r	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

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 （不使用突发增量模式）
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。

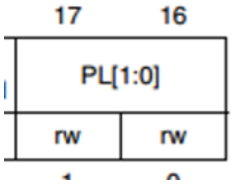
位 19 CT：当前目标（仅在双缓冲区模式下） (Current target (only in double buffer mode))

此位由硬件置 1 和清零，也可由软件写入。
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（禁止双缓冲区模式）
1： DMA 传输结束时切换目标存储区
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。
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
1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值）
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
1：使能循环模式
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。

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（数据项数是已知）
1：外设是流控制器（数据项数未知）
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（在配置时要禁止数据流）
1：使能数据流（配置完成后使能数据流）
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。
注意：将 EN 位置“1”以启动新传输之前， DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。

10.3.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)

偏移地址：0x14 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 NDT[15:0]：要传输的数据项数目 (Number of data items to transfer)

要传输的数据项数目（0 到 65535）。只有在禁止数据流时，才能向此寄存器执行写操作。

使能数据流后，此寄存器为只读，用于指示要传输的剩余数据项数。每次 DMA 传输后，此寄存器将递减。

传输完成后，此寄存器保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：

— 以循环模式配置数据流时。

— 通过将 EN 位置“1”来重新使能数据流时

如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。

10.3.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7)

10.3.8 DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7)

10.3.9 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7)

偏移地址：0x24 + 0x24 × *数据流编号*

复位值：0x0000 0021

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								FEIE	Reserved	FS[2:0]			DMDIS	FTH[1:0]	
								rw		r	r	r	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7 FEIE： FIFO 错误中断使能 (FIFO error interrupt enable)

此位由软件置 1 和清零。

0：禁止 FE 中断

1：使能 FE 中断

位 6 保留，必须保持复位值

位 5:3 FS[2:0]： FIFO 状态 (FIFO status)

这些位为只读。

000： 0 < fifo_level < 1/4

001： 1/4 fifo_level < 1/2

010： 1/2 fifo_level < 3/4

011： 3/4 fifo_level < 满

100： FIFO 为空

101： FIFO 已满

其它：无意义

在直接模式（DMDIS 位为零）下，这些位无意义。

位 2 DMDIS：直接模式禁止 (Direct mode disable)

此位由软件置 1 和清零。它可由硬件置 1。

0：使能直接模式

1：禁止直接模式

此位受到保护，只有 EN 为“0”时才可以写入。

如果选择存储器到存储器模式（DMA_SxCR 中的 DIR 位为“10”），并且 DMA_SxCR 寄存器中的 EN 位为“1”，则此位由硬件置 1，因为在存储器到存储器配置不能使用直接模式。

位 1:0 FTH[1:0]： FIFO 阈值选择 (FIFO threshold selection)

这些位将由软件置 1 和清零。

00： FIFO 容量的 1/4

01： FIFO 容量的 1/2

10： FIFO 容量的 3/4

11： FIFO 完整容量

在直接模式（DMDIS 值为零）下，不使用这些位。

这些位受到保护，只有 EN 为“1”时才可以写入。

10.4 DMA 实验

10.4.1 硬件分析

串口发送数据(DMA)：存储器到外设

只要是外设到存储器或者存储器到外设，都需要查看映射表

表 36. DMA2 请求映射

外设请求	数据流 0	数据流 1	数据流 2	数据流 3	数据流 4	数据流 5	数据流 6	数据流 7
通道 0	ADC1		TIM8_CH1 TIM8_CH2 TIM8_CH3		ADC1		TIM1_CH1 TIM1_CH2 TIM1_CH3	
通道 1		DCMI	ADC2	ADC2		SPI6_TX(1)	SPI6_RX(1)	DCMI
通道 2	ADC3	ADC3		SPI5_RX(1)	SPI5_TX(1)	CRYP_OUT	CRYP_IN	HASH_IN
通道 3	SPI1_RX		SPI1_RX	SPI1_TX		SPI1_TX		
通道 4	SPI4_RX(1)	SPI4_TX(1)	USART1_RX	SDIO		USART1_RX	SDIO	USART1_TX
通道 5		USART6_RX	USART6_RX	SPI4_RX(1)	SPI4_TX(1)		USART6_TX	USART6_TX
通道 6	TIM1_TRIG	TIM1_CH1	TIM1_CH2	TIM1_CH1	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
通道 7		TIM8_UP	TIM8_CH1	TIM8_CH2	TIM8_CH3	SPI5_RX(1)	SPI5_TX(1)	TIM8_CH4 TIM8_TRIG TIM8_COM

1. 这些请求在 STM32F42xxx 和 STM32F43xxx 上可用。

由上图可知，我们要选择是 DMA2 数据流 7 的 CH4 来作为外设请求触发的通道。

10.4.2 软件设计

配置流程

1. 确保串口能正常工作----串口初始化
2. 使能串口的 DMA 模式(发送)CR3 位 7 //相当于对应数据流上
3. 打开 DMA2 的时钟
4. 配置 CR 寄存器
5. 设置好源和目标的地址

6. 设置数据项数---变量存储的大小。---发送的次数。
7. 使能数据流

DMA2_Stream7->CR

在 KEIL 中新建.c 和.h 文件

程序实现流程：

1、//使能 DMA 发送器

USART1->CR3 |= (0x1 << 7); //DMA 映射到 USART1_TX 发送的端口上

位 7 **TXEIE**：TXE 中断使能 (TXE interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 USART_SR 寄存器中 TXE=1 时，生成 USART 中断。

2、//开启 DMA 时钟

RCC->AHB1ENR |= (0x1 << 22);

6.3.12 RCC AHB1 外设时钟使能寄存器 (RCC_AHB1ENR)

RCC AHB1 peripheral clock enable register

偏移地址：0x30

复位值：0x0010 0000

访问：无等待周期，按字、半字和字节访问。

开启DMA2时钟

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser-ved	OTGHS ULPIEN	OTGHS EN	ETHMA CPTPE N	ETHMA CRXEN	ETHMA CTXEN	ETHMA CEN	Reserved		DMA2EN	DMA1EN	CCMDATA RAMEN	Res.	BKPSR AMEN	Reserved	
	rw	rw	rw	rw	rw	rw			rw	rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEN	Reserved			GPIOIE N	GPIOH EN	GPIOGE N	GPIOF EN	GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

3、//配置 CR 寄存器

DMA2_Stream7->CR = 0; //CR 寄存器清零

DMA2_Stream7->CR |= (0x4 << 25);//通道 4

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

DMA2_Stream7->CR |= (0x1 << 10);//存储器地址增(MSIZE 1 字节)

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。

DMA2_Stream7->CR |= (0x1 << 6);//存储器到外设

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。
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。

4、源地址和目标地址设置

//设置源和目标地址

DMA2_Stream7->M0AR = m0ar; //源地址

位 31:0 **M0A[31:0]**：存储器 0 地址 (Memory 0 address)
读/写数据的存储区 0 的基址。
这些位受到写保护，只有在以下情况下才可以写入：
– 禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）或
– 使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“1”（在双缓冲区模式下）。

DMA2_Stream7->PAR = (u32)&USART1->DR; //目标地址

9.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7)

DMA stream x peripheral address register

偏移地址：0x18 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PAR[31:0]**：外设地址 (Peripheral address)
读/写数据的外设数据寄存器的基址。
这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。

5、搬运次数

//搬运的次数

DMA2_Stream7->NDTR = ndtr;

9.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)

DMA stream x number of data register

偏移地址：0x14 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：要传输的数据项数目 (Number of data items to transfer)
要传输的数据项数目（0 到 65535）。只有在禁止数据流时，才能向此寄存器执行写操作。
使能数据流后，此寄存器为只读，用于指示要传输的剩余数据项数。每次 DMA 传输后，此寄存器将递减。
传输完成后，此寄存器保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：
– 以循环模式配置数据流时。
– 通过将 EN 位置“1”来重新使能数据流时
如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。

6、 //使能数据流 7

DMA2_Stream7->CR |= (0x1 << 0);

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。

注意：将 EN 位置“1”以启动新传输之前，DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。

存储器到存储器实验：

区别部分代码：

第一部分：//使能 FIFO 模式

DMA2_Stream2->FCR |= (0x1 << 2);

位 2 **DMDIS**：直接模式禁止 (Direct mode disable)

此位由软件置 1 和清零。它可由硬件置 1。

0：使能直接模式

1：禁止直接模式

默认选中FIFO模式

此位受到保护，只有 EN 为“0”时才可以写入。

如果选择存储器到存储器模式（DMA_SxCR 中的 DIR 位为“10”），并且 DMA_SxCR 寄存器中的 EN 位为“1”，则此位由硬件置 1，因为在存储器到存储器配置不能使用直接模式。

第二部分：//配置 CR 寄存器

DMA2_Stream2->CR = 0; //对 DMA2 控制寄存器清零

DMA2_Stream2->CR |= (0x3 << 25);//通道 3

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

DMA2_Stream2->CR |= (0x1 << 21);//4 个节拍

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。

DMA2_Stream2->CR |= (0x1 << 10);//存储器地址增(MSIZE 1 字节)

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。

DMA2_Stream2->CR |= (0x1 << 9);//外设地址增(PSIZE 1 字节)

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
1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值）
此位受到保护，只有 EN 为“0”时才可以写入。

DMA2_Stream2->CR |= (0x2 << 6);//存储器到存储器

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。
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。