



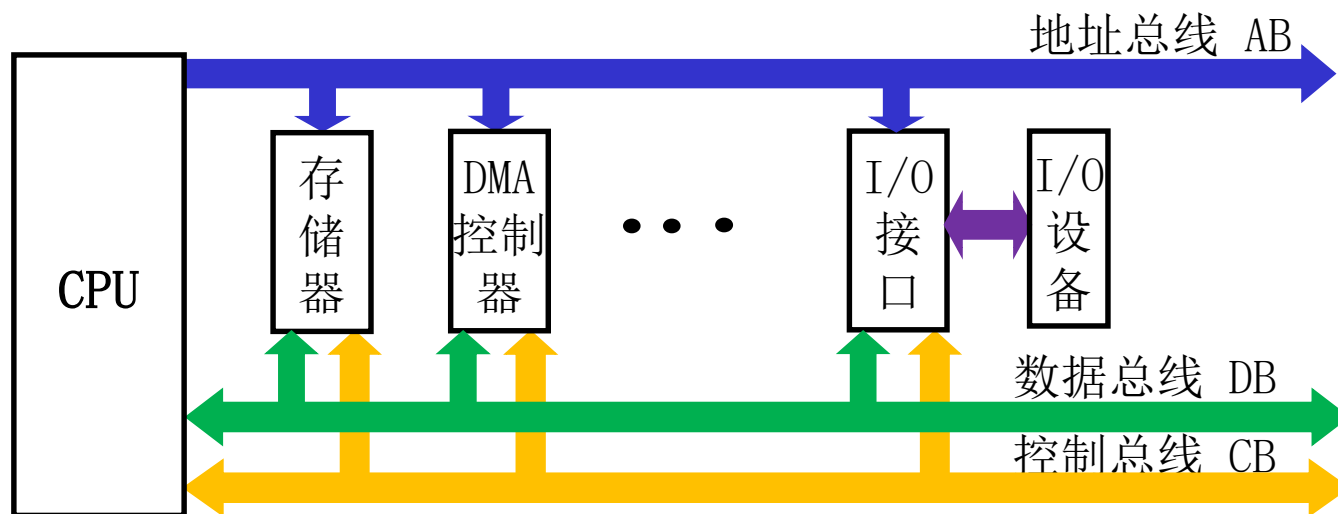
第七章 DMA



isszym 2019.12.11

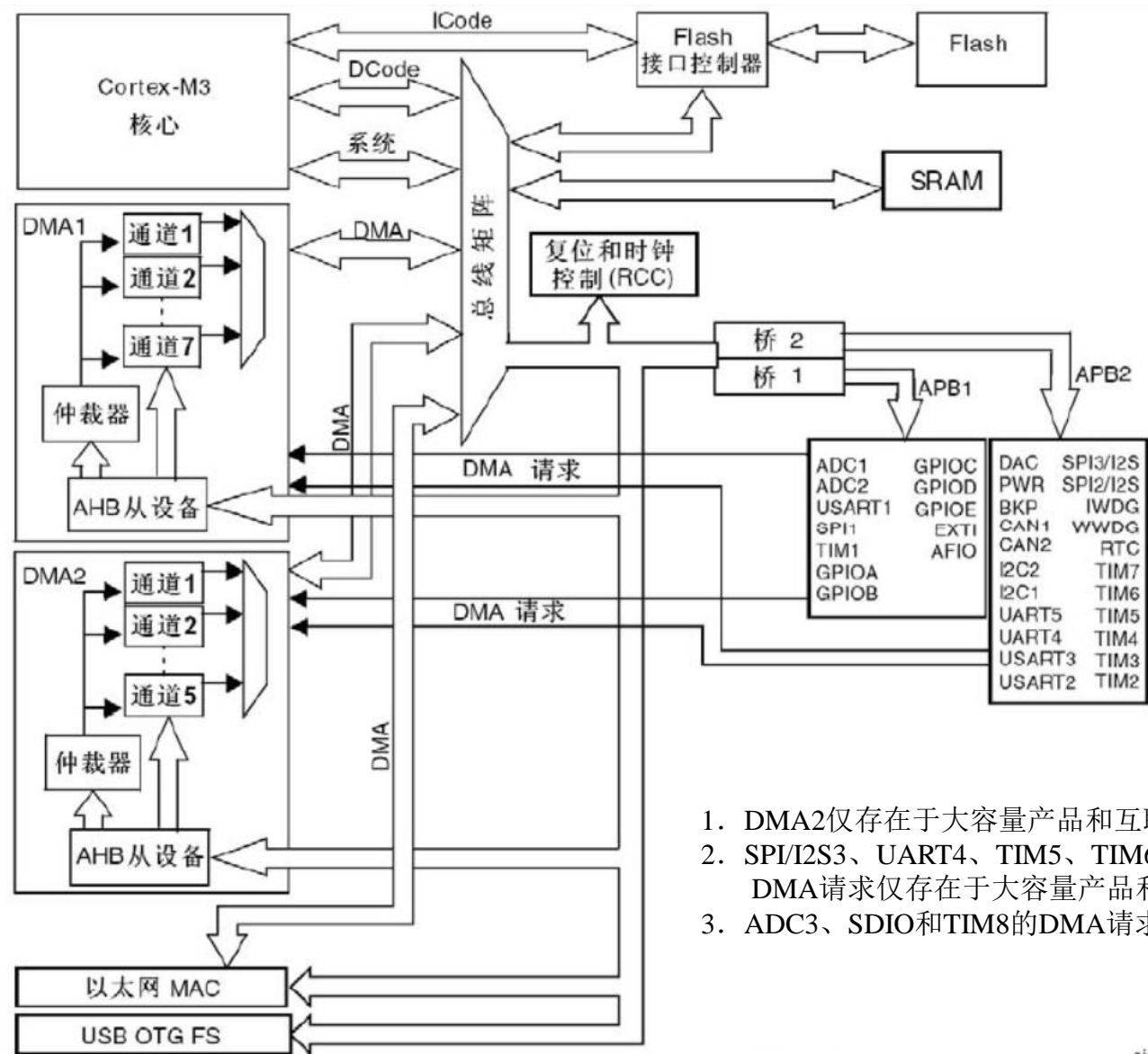
概述

- 直接存储器存取(DMA, Direct Memory Access)用来提供在外设和存储器之间或者存储器和存储器之间的高速数据传输。
- DMA控制器通过从CPU接管总线（总线窃取）实现数据传送，即，DMA传输前，CPU要把总线控制权交给DMA控制器，而在结束DMA传输后，DMA控制器应立即把总线控制权再交回给CPU。由于总线结构不同，STM32F103C8T6的DMA略有不同。



- STM32F103C8T6: DMA1有7个通道，DMA2有5个通道。有优先权控制，闪存、SRAM、APB1、APB2和AHB外设均可作为访问的源和目标。

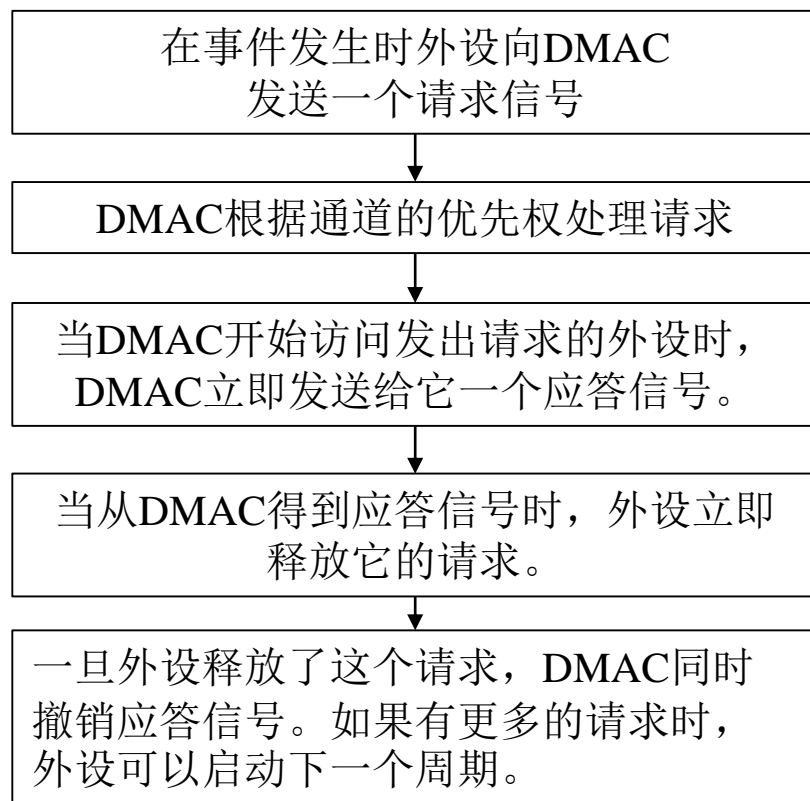
DMA框图



1. DMA2仅存在于大容量产品和互联型产品。
2. SPI/I2S3、UART4、TIM5、TIM6、TIM7和DAC的DMA请求仅存在于大容量产品和互联型产品。
3. ADC3、SDIO和TIM8的DMA请求仅存在于大容量产。

功能描述

- **DMA**控制器(DMAC)和Cortex™-M3核心共享系统数据总线，执行直接存储器数据传输。当CPU和DMA同时访问相同的目标(RAM或外设)时，DMA请求会暂停CPU访问系统总线达若干个周期，总线仲裁器执行循环调度，以保证CPU至少可以得到一半的系统总线(存储器或外设)带宽。



□ 每次DMA传送由3个操作组成：

- 取数据

从外设数据寄存器或者从当前外设/存储器地址寄存器指示的存储器地址取数据，第一次传输时的开始地址是DMA_CPARx或DMA_CMARx寄存器指定的外设基地址或存储器单元。

- 存数据

存数据到外设数据寄存器或者当前外设/存储器地址寄存器指示的存储器地址，第一次传输时的开始地址是DMA_CPARx或DMA_CMARx寄存器指定的外设基地址或存储器单元。

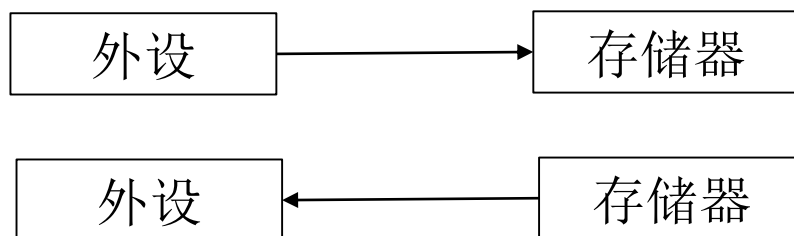
- 计数

执行一次DMA_CNDTRx寄存器的递减操作，该寄存器包含未完成的操作数目。

DMA_CPARx	DMA channel x peripheral address register
DMA_CMARx	DMA channel x memory address register
DMA_CNDTRx	DMA channel x number of data to transfer register

DMA通道

□ 传送源和目的



- 外设地址 DMA_CPAR_x
- 存储器地址 DMA_CMAR_x
- 传送的数据量 PSIZE 外设的数据宽度(比特)
 ~~MSIZE 外设的数据宽度(比特) (≤65535)~~
- 地址增量值 等于所选的数据宽度 (1、2或4)
- 传送计数 DMA_CNDTR_x
- 循环模式: 当启动了该模式, 计数值变为0(传送结束)时, 将会自动地被恢复成配置通道时设置的初值, DMA操作将会继续进行。
- 非循环模式: 传送结束后要重启DMA, 需要关闭DMA通道和写寄存器的值后再启动。
- 存储器和存储器: 它们之间可以传送数据。

❑ 通道x的配置过程

1. 在DMA_CPARx寄存器中设置外设寄存器的地址。
2. 在DMA_CMARx寄存器中设置数据存储器的地址。
3. 在DMA_CNDTRx寄存器中设置要传输的数据量。在每个数据传输后，这个数值递减。
4. 在DMA_CCRx寄存器的PL[1:0]位中设置通道的优先级。
5. 在DMA_CCRx寄存器中设置数据传输的方向、循环模式、外设和存储器的增量模式、外设和存储器的数据宽度、传输一半产生中断或传输完成产生中断。
6. 设置DMA_CCRx寄存器的ENABLE位，启动该通道。

中断

- 每个**DMA**通道都可以在**DMA**传输过半、传输完成和传输错误时产生中断。为应用的灵活性考虑，通过设置寄存器的不同位来打开这些中断。

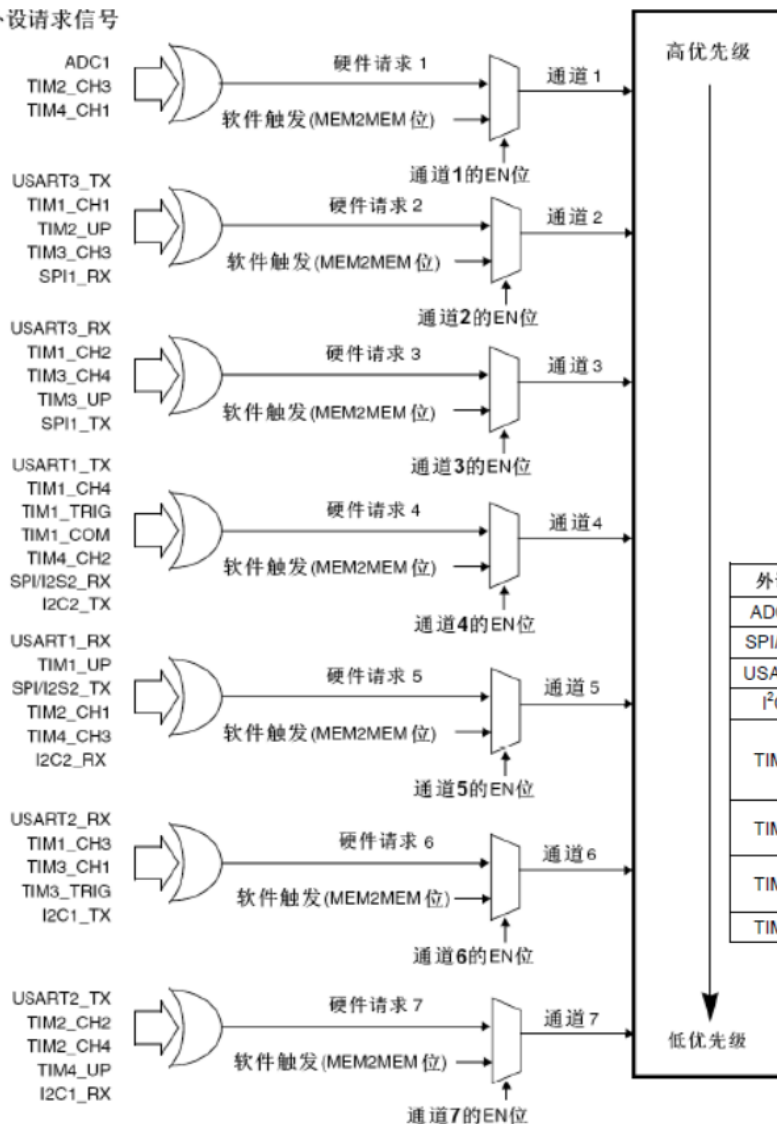
中断事件	事件标志位	使能控制位
传输过半	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

在大容量产品中，DMA2通道4和DMA2通道5的中断被映射在同一个中断向量上。
在互联型产品中，DMA2通道4和DMA2通道5的中断分别有独立的中断向量。所有其他的DMA通道都有自己的中断向量。

- 读写一个保留的地址区域，将会产生**DMA**传输错误。当在**DMA**读写操作时发生**DMA**传输错误时，硬件会自动地清除发生错误的通道所对应的通道配置寄存器(**DMA_CCRx**)的**EN**位，该通道操作被停止。此时，在**DMA_IFR**寄存器中对应该通道的传输错误中断标志位(**TEIF**)将被置位，如果在**DMA_CCRx**寄存器中设置了传输错误中断允许位，则将产生中断。
- 当**PSIZE**和**MSIZE**不相同，要求按照手册传输宽度表进行设置，否则将出错。

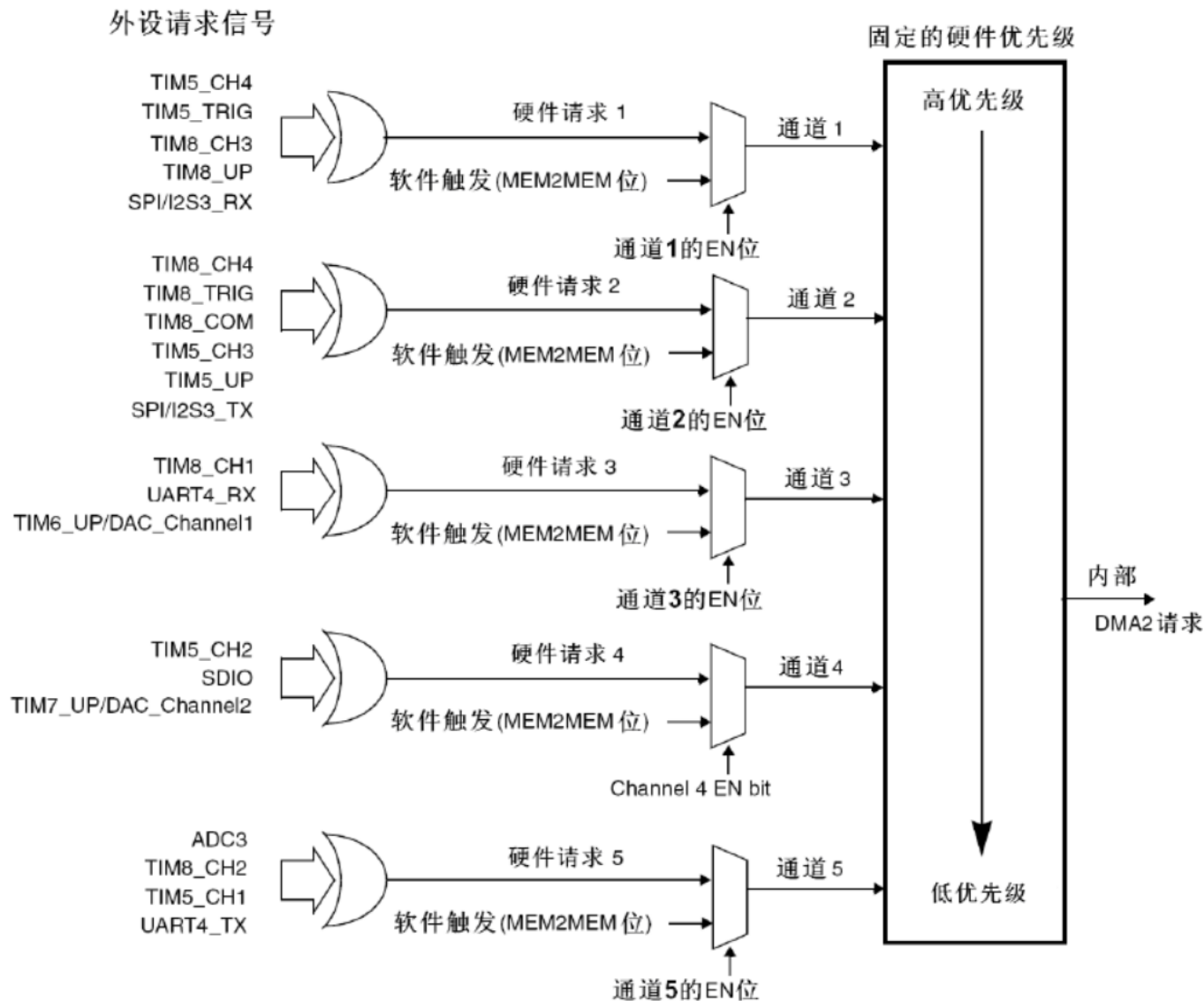
DMA1的请求映像

外设请求信号



外设	通道1	通道2	通道3	通道4	通道5	通道6	通道7
ADC1	ADC1						
SPI/I ² S		SPI1_RX	SPI1_TX	SPI/I2S2_RX	SPI/I2S2_TX		
USART		USART3_TX	USART3_RX	USART1_TX	USART1_RX	USART2_RX	USART2_TX
I ² C				I2C2_TX	I2C2_RX	I2C1_TX	I2C1_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_TX4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIM3		TIM3_CH3	TIM3_CH4 TIM3_UP			TIM3_CH1 TIM3_TRIG	
TIM4	TIM4_CH1			TIM4_CH2	TIM4_CH3		TIM4_UP

DMA2的请求映像



外设	通道1	通道2	通道3	通道4	通道5
ADC3 ⁽¹⁾					ADC3
SPI/I2S3	SPI/I2S3_RX	SPI/I2S3_TX			
UART4			UART4_RX		UART4_TX
SDIO ⁽¹⁾				SDIO	
TIM5	TIM5_CH4 TIM5_TRIG	TIM5_CH3 TIM5_UP		TIM5_CH2	TIM5_CH1
TIM6/ DAC通道1			TIM6_UP/ DAC通道1		
TIM7/ DAC通道2				TIM7_UP/ DAC通道2	
TIM8 ⁽¹⁾	TIM8_CH3 TIM8_UP	TIM8_CH4 TIM8_TRIG TIM8_COM	TIM8_CH1		TIM8_CH2

1. ADC3、SDIO和TIM8的DMA请求只在大容量的产品中存在。

寄存器

DMA1基地址: 0x40020000

□ DMA中断状态寄存器(DMA_ISR)

偏移地址: 0x00

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				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
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位31:28	保留, 始终读为0。
位27, 23, 19, 15, 11, 7, 3	TEIFx : 通道x的传输错误标志(x = 1 ... 7) (Channel x transfer error flag) 硬件设置这些位。在DMA_IFCR寄存器的相应位写入'1'可以清除这里对应的标志位。 0: 在通道x没有传输错误(TE); 1: 在通道x发生了传输错误(TE)。
位26, 22, 18, 14, 10, 6, 2	HTIFx : 通道x的半传输标志(x = 1 ... 7) (Channel x half transfer flag) 硬件设置这些位。在DMA_IFCR寄存器的相应位写入'1'可以清除这里对应的标志位。 0: 在通道x没有半传输事件(HT); 1: 在通道x产生了半传输事件(HT)。
位25, 21, 17, 13, 9, 5, 1	TCIFx : 通道x的传输完成标志(x = 1 ... 7) (Channel x transfer complete flag) 硬件设置这些位。在DMA_IFCR寄存器的相应位写入'1'可以清除这里对应的标志位。 0: 在通道x没有传输完成事件(TC); 1: 在通道x产生了传输完成事件(TC)。

位24, 20, 16, 12, 8, 4, 0	GIFx: 通道x的全局中断标志(x = 1 ... 7) (Channel x global interrupt flag) 硬件设置这些位。在DMA_IFCR寄存器的相应位写入'1'可以清除这里对应的标志位。 0: 在通道x没有TE、HT或TC事件; 1: 在通道x产生了TE、HT或TC事件。
--------------------------------	---

□ DMA中断标志清除寄存器(DMA_IFCR)

偏移地址: 0x04

复位值: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留				CTEIF 7	CHTIF 7	CTCIF 7	CGIF 7	CTEIF 6	CHTIF 6	CTCIF 6	CGIF 6	CTEIF 5	CHTIF 5	CTCIF 5	CGIF 5
				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
CTEIF 4	CHTIF 4	CTCIF 4	CGIF 4	CTEIF 3	CHTIF 3	CTCIF 3	CGIF 3	CTEIF 2	CHTIF 2	CTCIF 2	CGIF 2	CTEIF 1	CHTIF 1	CTCIF 1	CGIF 1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位31:28	保留, 始终读为0。
位27, 23, 19, 15, 11, 7, 3	CTEIFx: 清除通道x的传输错误标志(x = 1 ... 7) (Channel x transfer error clear) 这些位由软件设置和清除。 0: 不起作用 1: 清除DMA_ISR寄存器中的对应TEIF标志。
位26, 22, 18, 14, 10, 6, 2	CHTIFx: 清除通道x的半传输标志(x = 1 ... 7) (Channel x half transfer clear) 这些位由软件设置和清除。 0: 不起作用 0: 清除DMA_ISR寄存器中的对应HTIF标志。
位25, 21, 17, 13, 9, 5, 1	CTCIFx: 清除通道x的传输完成标志(x = 1 ... 7) (Channel x transfer complete clear) 这些位由软件设置和清除。 0: 不起作用 0: 清除DMA_ISR寄存器中的对应TCIF标志。
位24, 20, 16, 12, 8, 4, 0	CGIFx: 清除通道x的全局中断标志(x = 1 ... 7) (Channel x global interrupt clear) 这些位由软件设置和清除。 0: 不起作用 0: 清除DMA_ISR寄存器中的对应的GIF、TEIF、HTIF和TCIF标志。

❑ DMA通道x配置寄存器(DMA_CCRx)(x = 1...7)

偏移地址: 0x08 + 20 x (通道编号 - 1) 复位值: 0x0000 0000

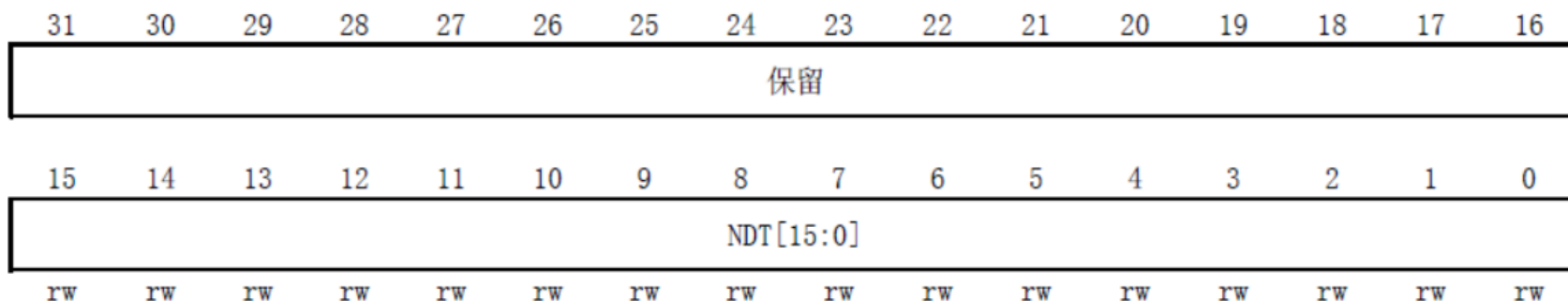
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	MEM2 MEM	PL[1:0]	MSIZE[1:0]	PSIZE[1:0]	MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN			
位14		MEM2MEM: 存储器到存储器模式 (Memory to memory mode) 该位由软件设置和清除。 0: 非存储器到存储器模式; 1: 启动存储器到存储器模式。													
位13:12		PL[1:0]: 通道优先级 (Channel priority level) 这些位由软件设置和清除。 00: 低 01: 中 10: 高 11: 最高													

位11:10	MSIZE[1:0]: 存储器数据宽度 (Memory size) 这些位由软件设置和清除。 00: 8位 01: 16位 10: 32位 11: 保留
位9:8	PSIZE[1:0]: 外设数据宽度 (Peripheral size) 这些位由软件设置和清除。 00: 8位 01: 16位 10: 32位 11: 保留
位7	MINC: 存储器地址增量模式 (Memory increment mode) 该位由软件设置和清除。 0: 不执行存储器地址增量操作 1: 执行存储器地址增量操作
位6	PINC: 外设地址增量模式 (Peripheral increment mode) 该位由软件设置和清除。 0: 不执行外设地址增量操作 1: 执行外设地址增量操作
位5	CIRC: 循环模式 (Circular mode) 该位由软件设置和清除。 0: 不执行循环操作 1: 执行循环操作

位4	DIR: 数据传输方向 (Data transfer direction) 该位由软件设置和清除。 0: 从外设读 1: 从存储器读
位3	TEIE: 允许传输错误中断 (Transfer error interrupt enable) 该位由软件设置和清除。 0: 禁止TE中断 0: 允许TE中断
位2	HTIE: 允许半传输中断 (Half transfer interrupt enable) 该位由软件设置和清除。 0: 禁止HT中断 0: 允许HT中断
位1	TCIE: 允许传输完成中断 (Transfer complete interrupt enable) 该位由软件设置和清除。 0: 禁止TC中断 0: 允许TC中断
位0	EN: 通道开启 (Channel enable) 该位由软件设置和清除。 0: 通道不工作 1: 通道开启

❑ DMA通道x传输数量寄存器(DMA_CNDTRx)(x = 1...7)

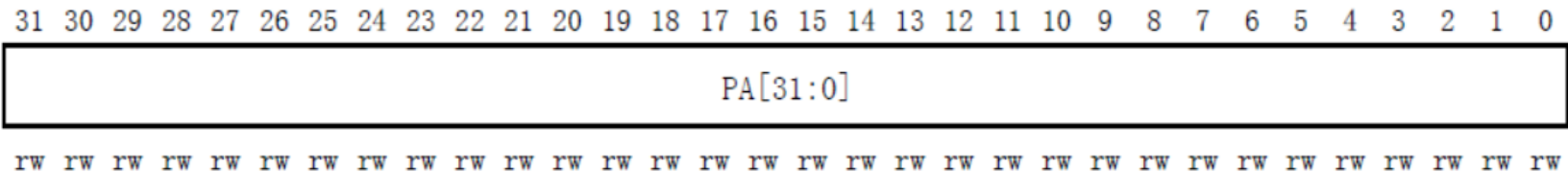
偏移地址: $0x0C + 20 \times (\text{通道编号} - 1)$ 复位值: 0x0000 0000



位31:16	保留，始终读为0。
位15:0	<p>NDT[15:0]: 数据传输数量 (Number of data to transfer)</p> <p>数据传输数量为0至65535。这个寄存器只能在通道不工作(DMA_CCRx的EN=0)时写入。通道开启后该寄存器变为只读，指示剩余的待传输字节数目。寄存器内容在每次DMA传输后递减。</p> <p>数据传输结束后，寄存器的内容或者变为0；或者当该通道配置为自动重加载模式时，寄存器的内容将被自动重新加载为之前配置时的数值。</p> <p>当寄存器的内容为0时，无论通道是否开启，都不会发生任何数据传输。</p>

❑ **DMA通道x外设地址寄存器(DMA_CPARx)(x = 1...7)**

偏移地址: $0x10 + 20 \times (\text{通道编号} - 1)$ 复位值: 0x0000 0000

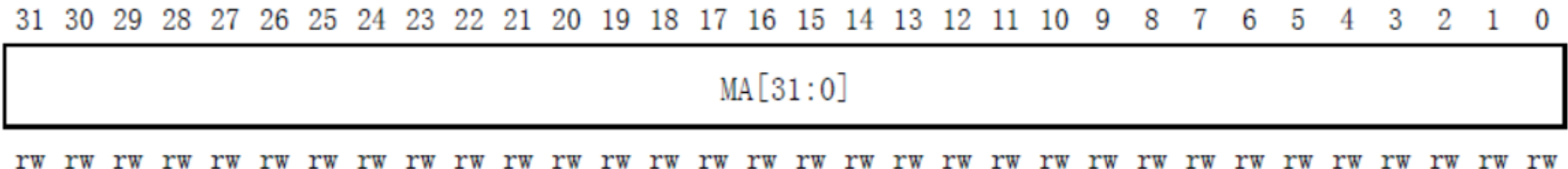


位31:0	<p>PA[31:0]: 外设地址 (Peripheral address)</p> <p>外设数据寄存器的基地址，作为数据传输的源或目标。</p> <p>当PSIZE='01'(16位)，不使用PA[0]位。操作自动地与半字地址对齐。</p> <p>当PSIZE='10'(32位)，不使用PA[1:0]位。操作自动地与字地址对齐。</p>
-------	--

当开启通道(DMA_CCRx的EN=1)时不能写该寄存器。

❑ **DMA通道x存储器地址寄存器(DMA_CMARx)(x = 1...7)**

偏移地址: $0x14 + 20 \times (\text{通道编号} - 1)$ 复位值: 0x0000 0000



位31:0	MA[31:0]: 存储器地址 存储器地址作为数据传输的源或目标。
-------	---

当MSIZE='01'(16位), 不使用MA[0]位。操作自动地与半字地址对齐。 当MSIZE='10'(32位), 不使用MA[1:0]位。操作自动地与字地址对齐。当开启通道 (DMA_CCRx的EN=1) 时不能写该寄存器。

附录1、AHB使能寄存器 (RCC_AHBENR)

偏移地址：0x14 复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					SDIOEN	保留	FSMCEN	保留	CRCEN	保留	FLITF EN	保留	SRAM EN	DMA2 EN	DMA1 EN
					rW		rW		rW		rW		rW	rW	rW

RCC基地址：0X40021000

附录2、基地址

起始地址	外设	总线
0x5000 0000 – 0x5003 FFFF	USB OTG 全速	AHB
0x4003 0000 – 0x4FFF FFFF	保留	
0x4002 8000 – 0x4002 9FFF	以太网	
0x4002 3400 - 0x4002 3FFF	保留	AHB
0x4002 3000 - 0x4002 33FF	CRC	
0x4002 2000 - 0x4002 23FF	闪存存储器接口	
0x4002 1400 - 0x4002 1FFF	保留	
0x4002 1000 - 0x4002 13FF	复位和时钟控制(RCC)	
0x4002 0800 - 0x4002 0FFF	保留	
0x4002 0400 - 0x4002 07FF	DMA2	
0x4002 0000 - 0x4002 03FF	DMA1	
0x4001 8400 - 0x4001 7FFF	保留	
0x4001 8000 - 0x4001 83FF	SDIO	
0x4001 4000 - 0x4001 7FFF	保留	
0x4001 3C00 - 0x4001 3FFF	ADC3	APB2
0x4001 3800 - 0x4001 3BFF	USART1	
0x4001 3400 - 0x4001 37FF	TIM8定时器	
0x4001 3000 - 0x4001 33FF	SPI1	
0x4001 2C00 - 0x4001 2FFF	TIM1定时器	
0x4001 2800 - 0x4001 2BFF	ADC2	
0x4001 2400 - 0x4001 27FF	ADC1	
0x4001 2000 - 0x4001 23FF	GPIO端口G	
0x4001 2000 - 0x4001 23FF	GPIO端口F	
0x4001 1800 - 0x4001 1BFF	GPIO端口E	
0x4001 1400 - 0x4001 17FF	GPIO端口D	
0x4001 1000 - 0x4001 13FF	GPIO端口C	
0x4001 0C00 - 0x4001 0FFF	GPIO端口B	
0x4001 0800 - 0x4001 0BFF	GPIO端口A	
0x4001 0400 - 0x4001 07FF	EXTI	
0x4001 0000 - 0x4001 03FF	AFIO	
0x4000 7800 - 0x4000FFFF	保留	APB1
0x4000 7400 - 0x4000 77FF	DAC	
0x4000 7000 - 0x4000 73FF	电源控制(PWR)	
0x4000 6C00 - 0x4000 6FFF	后备寄存器(BKP)	
0x4000 6800 - 0x4000 6BFF	bxCAN2	
0x4000 6400 - 0x4000 67FF	bxCAN1	
0x4000 6000 ⁽¹⁾ - 0x4000 63FF	USB/CAN共享的512字节SRAM	

0x4000 5C00 - 0x4000 5FFF	USB全速设备寄存器	APB1
0x4000 5800 - 0x4000 5BFF	I2C2	
0x4000 5400 - 0x4000 57FF	I2C1	
0x4000 5000 - 0x4000 53FF	UART5	
0x4000 4C00 - 0x4000 4FFF	UART4	
0x4000 4800 - 0x4000 4BFF	USART3	
0x4000 4400 - 0x4000 47FF	USART2	
0x4000 4000 - 0x4000 3FFF	保留	
0x4000 3C00 - 0x4000 3FFF	SPI3/I2S3	
0x4000 3800 - 0x4000 3BFF	SPI2/I2S3	
0x4000 3400 - 0x4000 37FF	保留	
0x4000 3000 - 0x4000 33FF	独立看门狗(IWDG)	
0x4000 2C00 - 0x4000 2FFF	窗口看门狗(WWDG)	
0x4000 2800 - 0x4000 2BFF	RTC	
0x4000 1800 - 0x4000 27FF	保留	
0x4000 1400 - 0x4000 17FF	TIM7定时器	
0x4000 1000 - 0x4000 13FF	TIM6定时器	
0x4000 0C00 - 0x4000 0FFF	TIM5定时器	
0x4000 0800 - 0x4000 0BFF	TIM4定时器	
0x4000 0400 - 0x4000 07FF	TIM3定时器	
0x4000 0000 - 0x4000 03FF	TIM2定时器	

中断向量表(Interrupt Vector Table)

NVIC
register
map

位置	优先级	优先级类型	名称	说明	地址
	-	-	-	保留	0x0000_0000
	-3	固定	Reset	复位	0x0000_0004
	-2	固定	NMI	不可屏蔽中断 RCC时钟安全系统(CSS)联接到NMI向量	0x0000_0008
	-1	固定	硬件失效(HardFault)	所有类型的失效	0x0000_000C
	0	可设置	存储管理(MemManage)	存储器管理	0x0000_0010
	1	可设置	总线错误(BusFault)	预取指失败, 存储器访问失败	0x0000_0014
	2	可设置	错误应用(UsageFault)	未定义的指令或非法状态	0x0000_0018
	-	-	-	保留	0x0000_001C ~0x0000_002B
	3	可设置	SVCall	通过SWI指令的系统服务调用	0x0000_002C
	4	可设置	调试监控(DebugMonitor)	调试监控器	0x0000_0030
	-	-	-	保留	0x0000_0034
	5	可设置	PendSV	可挂起的系统服务	0x0000_0038
	6	可设置	SysTick	系统嘀嗒定时器	0x0000_003C
0	7	可设置	WWDG	窗口定时器中断	0x0000_0040
1	8	可设置	PVD	连到EXTI的电源电压检测(PVD)中断	0x0000_0044
2	9	可设置	TAMPER	侵入检测中断	0x0000_0048
3	10	可设置	RTC	实时时钟(RTC)全局中断	0x0000_004C
4	11	可设置	FLASH	闪存全局中断	0x0000_0050

5	12	可设置	RCC	复位和时钟控制(RCC)中断	0x0000_0054
6	13	可设置	EXTI0	EXTI线0中断	0x0000_0058
7	14	可设置	EXTI1	EXTI线1中断	0x0000_005C
8	15	可设置	EXTI2	EXTI线2中断	0x0000_0060
9	16	可设置	EXTI3	EXTI线3中断	0x0000_0064
10	17	可设置	EXTI4	EXTI线4中断	0x0000_0068
11	18	可设置	DMA1通道1	DMA1通道1全局中断	0x0000_006C
12	19	可设置	DMA1通道2	DMA1通道2全局中断	0x0000_0070
13	20	可设置	DMA1通道3	DMA1通道3全局中断	0x0000_0074
14	21	可设置	DMA1通道4	DMA1通道4全局中断	0x0000_0078
15	22	可设置	DMA1通道5	DMA1通道5全局中断	0x0000_007C
16	23	可设置	DMA1通道6	DMA1通道6全局中断	0x0000_0080
17	24	可设置	DMA1通道7	DMA1通道7全局中断	0x0000_0084
18	25	可设置	ADC1_2	ADC1和ADC2的全局中断	0x0000_0088
19	26	可设置	USB_HP_CAN_TX	USB高优先级或CAN发送中断	0x0000_008C
20	27	可设置	USB_LP_CAN_RX0	USB低优先级或CAN接收0中断	0x0000_0090
21	28	可设置	CAN_RX1	CAN接收1中断	0x0000_0094
22	29	可设置	CAN_SCE	CAN SCE中断	0x0000_0098
23	30	可设置	EXTI9_5	EXTI线[9:5]中断	0x0000_009C
24	31	可设置	TIM1_BRK	TIM1刹车中断	0x0000_00A0
25	32	可设置	TIM1_UP	TIM1更新中断	0x0000_00A4
26	33	可设置	TIM1_TRG_COM	TIM1触发和通信中断	0x0000_00A8
27	34	可设置	TIM1_CC	TIM1捕获比较中断	0x0000_00AC
28	35	可设置	TIM2	TIM2全局中断	0x0000_00B0

29	36	可设置	TIM3	TIM3全局中断	0x0000_00B4
30	37	可设置	TIM4	TIM4全局中断	0x0000_00B8
31	38	可设置	I2C1_EV	I ² C1事件中断	0x0000_00BC
32	39	可设置	I2C1_ER	I ² C1错误中断	0x0000_00C0
33	40	可设置	I2C2_EV	I ² C2事件中断	0x0000_00C4
34	41	可设置	I2C2_ER	I ² C2错误中断	0x0000_00C8
35	42	可设置	SPI1	SPI1全局中断	0x0000_00CC
36	43	可设置	SPI2	SPI2全局中断	0x0000_00D0
37	44	可设置	USART1	USART1全局中断	0x0000_00D4
38	45	可设置	USART2	USART2全局中断	0x0000_00D8
39	46	可设置	USART3	USART3全局中断	0x0000_00DC
40	47	可设置	EXTI15_10	EXTI线[15:10]中断	0x0000_00E0
41	48	可设置	RTCAlarm	连到EXTI的RTC闹钟中断	0x0000_00E4
42	49	可设置	USB唤醒	连到EXTI的从USB待机唤醒中断	0x0000_00E8
43	50	可设置	TIM8_BRK	TIM8刹车中断	0x0000_00EC
44	51	可设置	TIM8_UP	TIM8更新中断	0x0000_00F0
45	52	可设置	TIM8_TRG_COM	TIM8触发和通信中断	0x0000_00F4
46	53	可设置	TIM8_CC	TIM8捕获比较中断	0x0000_00F8
47	54	可设置	ADC3	ADC3全局中断	0x0000_00FC
48	55	可设置	FSMC	FSMC全局中断	0x0000_0100

49	56	可设置	SDIO	SDIO全局中断	0x0000_0104
50	57	可设置	TIM5	TIM5全局中断	0x0000_0108
51	58	可设置	SPI3	SPI3全局中断	0x0000_010C
52	59	可设置	UART4	UART4全局中断	0x0000_0110
53	60	可设置	UART5	UART5全局中断	0x0000_0114
54	61	可设置	TIM6	TIM6全局中断	0x0000_0118
55	62	可设置	TIM7	TIM7全局中断	0x0000_011C
56	63	可设置	DMA2通道1	DMA2通道1全局中断	0x0000_0120
57	64	可设置	DMA2通道2	DMA2通道2全局中断	0x0000_0124
58	65	可设置	DMA2通道3	DMA2通道3全局中断	0x0000_0128
59	66	可设置	DMA2通道4_5	DMA2通道4和DMA2通道5全局中断	0x0000_012C

位置即NVIC(Nested vectored interrupt controller) number for register map

□ NVIC register map

The table provides shows the NVIC register map and reset values. The base address of the main NVIC register block is 0xE000E100. The NVIC_STIR register is located in a separate block at 0xE000EF00.

NVIC register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	NVIC_ISER0	SETENA[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	NVIC_ISER1	SETENA[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	NVIC_ISER2	Reserved																SETENA [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x080	NVIC_ICER0	CLRENA[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	NVIC_ICER1	CLRENA[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	NVIC_ICER2	Reserved																CLRENA [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x100	NVIC_ISPR0	SETPEND[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	NVIC_ISPR1	SETPEND[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	NVIC_ISPR2	Reserved																SETPEND [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	NVIC_ICPR0	CLRPEND[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x184	NVIC_ICPR1	CLRPEND[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x188	NVIC_ICPR2	Reserved																CLRPEND [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x200	NVIC_IABR0	ACTIVE[31:0]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x204	NVIC_IABR1	ACTIVE[63:32]																															
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x208	NVIC_IABR2	Reserved																ACTIVE [80:64]															
	Reset Value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x300	NVIC_IPR0	IP[3]								IP[2]								IP[1]								IP[0]							
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
:	:	:																															
0x320	NVIC_IPR20	Reserved																							IP[80]								
	Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
SCB registers																																	
Reserved																																	
0xE00	NVIC_STIR	Reserved																							INTID[8:0]								
	Reset Value	Reserved																							0	0	0	0	0	0	0	0	0

□ Interrupt set-enable registers (NVIC_ISERx)

Address offset: 0x00 - 0x0B Reset value: 0x00000000

Required privilege: Privileged

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

Bits 31:0 SETENA[31:0]: Interrupt set-enable bits.

Write: 0: No effect 1: Enable interrupt

Read: 0: Interrupt disabled 1: Interrupt enabled.

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

❑ Interrupt clear-enable registers (NVIC_ICERx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ICER0-ICER2 registers disable interrupts, and show which interrupts are enabled.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRENA[31:16]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRENA[15:0]															
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bits 31:0 CLRENA[31:0]: Interrupt clear-enable bits.

Write: 0: No effect 1: Disable interrupt

Read: 0: Interrupt disabled 1: Interrupt enabled.

- ❑ Interrupt set-pending registers (NVIC_ISPRx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ISPR0-ISPR2 registers force interrupts into the pending state, and show which interrupts are pending.

[illegible]

Bits 31:0 SETPEND[31:0]: Interrupt set-pending bits

Write: 0: No effect 1: Changes interrupt state to pending

Read: 0: Interrupt is not pending 1: Interrupt is pending

Writing 1 to the ISPR bit corresponding to an interrupt that is pending:

- has no effect.

Writing 1 to the ISPR bit corresponding to a disabled interrupt:

- sets the state of that interrupt to pending.

- ❑ Interrupt clear-pending registers (NVIC_ICPRx)

Address offset: 0x00 - 0x0B Reset value: 0x0000 0000 Required privilege: Privileged

The ICPR0-ICPR2 registers remove the pending state from interrupts, and show which interrupts are pending.

[illegible]

Bits 31:0 CLRPEND[31:0]: Interrupt clear-pending bits

[illegible]

Read: 0: Interrupt is not pending 1: Interrupt is pending

Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.

通过 DMA把数据存入CCR

- ; TIM2的CH1~CH4均采用比较翻转功能且CCR1~CCR4的值均不相同,
- ; 在TIM2的20次更新中断之后启动更新DMA, 该DMA将SRAM的值(均相同)写入CCR2~CCR4。
- ; PA1~PA4分别通过发光二极管和电阻连至地线。

```
CFGAPERIPHCLK    EQU 0x0000BBBB ;PA0~PA3--复用推挽输出, 50MHz
GPIOAEN          EQU 0x00000004 ;GPIOA使能位
TIM2EN           EQU 0x00000001 ;TIM2使能位
DMA1EN           EQU 0x00000001 ;DMA1使能位
```

```
GPIOA            EQU 0x40010800 ;GPIOA基地址
GPIOA_CRL        EQU 0x40010800 ;低配置寄存器
GPIOA_CRH        EQU 0x40010804 ;高配置寄存器
GPIOA_ODR        EQU 0x4001080C ;输出, 偏移地址0Ch
GPIOA_BSRR       EQU 0x40010810 ;低置位, 高清除偏移地址10h
GPIOA_BRR        EQU 0x40010814 ;清除, 偏移地址14h
```

```
TIM2             EQU 0x40000000 ;TIM2基地址
TIM2_ARR         EQU TIM2+0x2C ;自动装载寄存器
TIM2_PSC         EQU TIM2+0x28 ;预分频器
TIM2_DIER        EQU TIM2+0x0C ;DMA/中断使能寄存器
TIM2_CR1         EQU TIM2+0x00 ;控制寄存器1
TIM2_SR          EQU TIM2+0x10 ;状态寄存器
TIM2_CCMR1       EQU TIM2+0x18 ;捕获/比较模式寄存器
TIM2_CCMR2       EQU TIM2+0x1C ;捕获/比较模式寄存器
TIM2_CCER        EQU TIM2+0x20 ;捕获/比较使能寄存器
TIM2_SMCR        EQU TIM2+0x08 ;从模式控制寄存器
```

```
AREA MYDATA2,DATA,READWRITE
IntCnt  DCD 0x50
```



```

STACK_TOP EQU 0X20002000
AREA RESET, CODE, READONLY ; AREA不能顶格写
DCD STACK_TOP ; MSP主堆栈指针
DCD START ; 复位, PC初始值
SPACE 0xB0-8;
DCD TIM2_IRQHandler ; TIM2
ENTRY ; 指示开始执行, 有了C文件, ENTRY注释掉
START ; 所有的标号必须顶格写, 且无冒号

```

```

BL InitRamArea ; ram必须赋初值

```

```

; 设置RCC的APB2使能寄存器, 启动GPIOA时钟

```

```

LDR R1, =RCC_APB2ENR
LDR R0, [R1]
LDR R2, =GPIOAEN
ORR R0, R2
STR R0, [R1]

```

```

; 设置RCC的APB1使能寄存器, 启动TIM2时钟

```

```

LDR R1, =RCC_APB1ENR
LDR R0, [R1]
LDR R2, =TIM2EN
ORR R0, R2
STR R0, [R1]

```

```

; 设置RCC的AHB使能寄存器, 启动DMA1时钟

```

```

LDR R1, =RCC_AHBENR
LDR R0, [R1]
LDR R2, =DMA1EN
ORR R0, R2
STR R0, [R1]

```

```

; 设置NVIC的中断设置允许寄存器 (ISER)
MOV     R0, #TIM2_ITEN
LDR     R1, =ISER_TIM2
STR     R0, [R1]

;===== GPIO =====
; 设置GPIO配置寄存器: PA0~PA3: 50MHz的复用推挽输出
LDR     R0, =CFG_A
LDR     R1, =GPIOA_CRL
STR     R0, [R1]

;===== DMA =====
LDR     R0, =CCRDatA      ; 源地址--SRAM
LDR     R1, =DMA1_CMAR2
STR     R0, [R1]

LDR     R0, =TIM2_DMAR    ; 目的地-TIM2的DMAR
LDR     R1, =DMA1_CPAR2
STR     R0, [R1]

MOV     R0, #1            ; 3次传送
LDR     R1, =DMA1_CNDTR2
STR     R0, [R1]

;===== TIM2 =====
; 设置定时器TIMx的分频器
MOV     R0, #(4000-1)
LDR     R1, =TIM2_PSC
STR     R0, [R1]

; 设置定时器TIMx的重装载寄存器
MOV     R0, #(1000-1)
LDR     R1, =TIM2_ARR
STR     R0, [R1]

```

```

; 设置定时器TIMx的OC1M='011'（匹配时翻转OC1输出）,OC1PE='0'
MOV      R0,#0x3030
LDR  R1, =TIM2_CCMR1
STR  R0, [R1]

; 设置定时器TIMx的OC1M='011'（匹配时翻转OC1输出）,OC1PE='0'
MOV      R0,#0x3030
LDR  R1, =TIM2_CCMR2
STR  R0, [R1]

; 配置TIMx_CCER寄存器的,CC1P='0'(OC2高电平有效),CC1E='1'(开启OC1输出到引脚)
MOV      R0,#0x1111
LDR  R1, =TIM2_CCER
STR  R0, [R1]

; 设置定时器Timx的捕获/比较寄存器
MOV  R0, #10
LDR  R1, =TIM2_CCR1
STR  R0, [R1]

; 设置定时器Timx的捕获/比较寄存器
MOV  R0, #10
LDR  R1, =TIM2_CCR2
STR  R0, [R1]

; 设置定时器Timx的捕获/比较寄存器
MOV  R0, #10
LDR  R1, =TIM2_CCR3
STR  R0, [R1]

; 设置定时器Timx的捕获/比较寄存器
MOV  R0, #10
LDR  R1, =TIM2_CCR4
STR  R0, [R1]

```

```

MOV      R0,#0x00    ; SMS='000' 关闭从模式，采用默认的时钟方式：HSI/2（4Mhz）
LDR  R1, =TIM2_SMCR
STR  R0, [R1]

MOV  R0,#0x010D  ;
LDR  R1, =TIM2_DCR ;每次传送2个字节，TIM2_DMAR地址映射到实际寄存器地址的方法：CR1地址
+(0xE+i)*4 i=0~2 （0x38 0x3C 0x40）
STR  R0, [R1]

; Enable the TIMx update DMA request (set the UDE bit in the DIER register) and interrupt
MOV  R0,#0x0101
LDR  R1, =TIM2_DIER
STR  R0, [R1]

; 设置定时器Tim3的CR1寄存器的CEN='1'，启动计数器
MOV  R0, #1
LDR  R1, =TIM2_CR1
STR  R0, [R1]

LOOP2
B LOOP2

TIM2_IRQHandler PROC
    PUSH {R0,R1,R2,LR}

    LDR  R0, =IntCnt
    LDR  R1, [R0]
    ADD  R1, #1    ; 中断次数加1
    STR  R1, [R0]
    CMP  R1,#0x10
    BEQ  TURNON    ; 如果中断次数为10，则启动DMA
    B EX

```

```

TURNON
    ; Enable the DMA channel
    MOV     R0, #0x05D1          ; MEM2MEM =0 非存储器到存储器模式; PL[1:0]=00 低优先权;
                                ; MSIZE[1:0]=01 存储器数据宽度=16位;
                                ; PSIZE[1:0]=01 外设数据宽度=16位; MINC=1 执行存储器地址增量操作;
                                ; PINC=1 执行外设地址增量操作;
                                ; CIRC=0 不执行循环操作; DIR=1 从存储器读; TEIE=0 禁止TE(Transfer Error)中断
                                ; TCIE=0 禁止TC(Transfer Complete)中断; EN=1 通道开启

    LDR     R1, =DMA1_CCR2
    STR     R0, [R1]
EX
    LDR     R1, =TIM2_SR
    MOV     R2, #0               ; 清除更新事件状态位
    STR     R2, [R1]

    POP     {R0, R1, R2, PC}
ENDP

; 所有RAM(MYDATA2)中的数据必须进行初始化
InitRamArea PROC
PUSH {R5, R6, LR}

; 初始化CURSELECT
LDR R5, =IntCnt
    MOV R6, #0
    STR R6, [R5]

POP {R5, R6, PC}
ENDP

END

```