

- 内核：32位的Core
 - 最高216MHz工作频率，可达2.54DMips/MHz(CoreMark1.0)
 - 单周期乘法和硬件除法
- 存储器
 - 128K/256K字节的闪存程序存储器
 - 32K/64K/96K字节的SRAM
- 时钟、复位和电源管理
 - 2.0 ~ 3.6伏供电和I/O引脚
 - 上电/断电复位(POR/PDR)、可编程电压监测器(PVD)
 - 4 ~ 32MHz晶体振荡器
 - 内嵌经出厂调校的8MHz的RC振荡器
 - 内嵌带校准的40kHz的RC振荡器
 - 带校准功能的32kHz RTC振荡器
- 低功耗
 - 睡眠、停机和待机模式
 - VBAT为RTC和后备寄存器供电
- 3个12位模数转换器，1us转换时间(多达16个输入通道)
 - 转换范围：0至3.6V
 - 三组采样和保持功能
 - 温度传感器
- 2个12位D/A转换器
- DMA：12通道 DMA控制器
 - 支持的外设：定时器、ADC、DAC、SDIO、SPI、I2S、I2C和USART
- 调试模式
 - 串行单线调试(SWD)和JTAG接口
 - 内嵌跟踪模块(ETM)
- 多达51个I/O端口
 - 51个多功能双向的I/O口，所有I/O口可以映像到16个外部中断
 - 所有GPIO均可强制配置上下拉电阻
- 增强型CRC计算单元
- 17个定时器
 - 10个16位定时器，每个定时器有多达4个用于输入捕获/输出比较/PWM或脉冲计数的通道和增量编码器输入
- 多达13个通信接口
 - 2个16位带死区控制和紧急刹车，用于电机控制的PWM高级控制定时器
 - 2个看门狗定时器(独立的和窗口型)
 - 系统时间定时器：24位自减型计数器
 - 2个16位基本定时器
- 多达2个I2C接口(支持MBus/PMBus)
- 多达5个USART接口(支持ISO7816, LIN, IrDA接口和调制解调控制)
- 多达3个SPI接口，2个带I2S接口多路复用
- CAN接口(2.0B 主动)
- USB 2.0全速接口(可选内部1.5K上拉电阻)
- SDIO接口
- 硬件加密算法单元
 - 内置硬件算法(DES、AES、SHA、SM1、SM3、SM4、M7)
 - 提供完整高性能算法库
- TRNG：TRNG单元用于产生真随机数序
 - 四个独立真随机源，可单独配置
 - 一次可产生128BIT随机数
 - 可选数字处理后功能
 - 攻击检测
- SENSOR：电压温度传感器报警
 - 可单独检测VBAT和VDD电压
 - 提供温度检测传感器
 - 可选报警后复位或者中断
- SRAM加扰
 - 支持地址、数据加扰
- 一次性可编程(OTP)
 - 支持32 Byte
- AES加密下载(部分型号支持):
 - 可下载使用AES加密后的HEX，硬件进行解密执行
 - 支持区域加密，解密功能

AIR32系列产品命名规则

示例: AIR32 F 103 C 8 T 6 A xxx xxxx

产品系列

AIR32 = 基于ARM®的32位微控制器

产品类型

F = 通用类型

S = 紧凑类型

产品子型

103 = 增强型

003 = 经济型

引脚数目

T = 36脚

C = 48脚

R = 64脚

V = 100脚

Z = 144脚

闪存存储器容量

4 = 16K字节的闪存存储器

6 = 32K字节的闪存存储器

8 = 64K字节的闪存存储器

B = 128K字节的闪存存储器

C = 256K字节的闪存存储器

封装

H = BGA

T = LQFP

U = VFQFPN

Y = WLCSP64

温度范围

6 = 工业级温度范围, -40°C~85°C

7 = 工业级温度范围, -40°C~105°C

内部代码

Axxx表示内部版本

生产周期

xxxx = 芯片生产时间, 前两位为年份后两位为周数

例如2231表示2022年第31周生产

目录

1 文中的缩写	5
1.1 寄存器描述表中使用的缩写列表	5
1.2 术语表	5
2 储存器和总线架构	6
2.1 系统架构	6
2.2 储存器	7
2.3 储存器映像	7
2.3.1 嵌入式SRAM	10
2.3.2 位段	10
2.3.3 嵌入式闪存 (flash)	10
2.4 启动配置 (BOOT)	13
3 CRC计算单元 (CRC)	14
3.1 CRC简介	14
3.2 CRC主要特性	14
3.3 CRC功能描述	14
3.4 CRC寄存器	15
3.4.1 数据寄存器(CRC_DR)	15
3.4.2 独立数据寄存器 (CRC_IDR)	16
3.4.3 控制寄存器 (CRC_CR)	16
3.4.4 控制状态寄存器 (CRC_CSR)	17
3.4.5 初始值寄存器 (CRC_INI)	17
4 电源控制 (PWR)	18
4.1 电源	18
4.1.1 独立的A/D转换器供电和参考电压	18
4.1.2 电池备份区域	19
4.1.3 电压调节器	19
4.2 电源管理器	20
4.2.1 上电复位(POR)和掉电复位(PDR)	20
4.2.2 可编程电压监测器(PVD)	20
4.3 低功耗模式	21
4.3.1 降低系统时钟	22
4.3.2 外部时钟的控制	22
4.3.3 睡眠模式	22
4.3.4 停止模式	23
4.3.5 待机模式	24
4.3.6 低功耗模式下的自动唤醒	25
4.4 电源控制寄存器 (PWR)	25
4.4.1 电源控制寄存器 (PWR_CR)	26
4.4.2 电源控制/状态寄存器(PWR_CSR)	27
5 备份寄存器 (BKP)	28

5.1 BKP简介.....	28
5.2 BKP特性.....	28
5.3 BKP功能描述.....	28
5.3.1 侵入检测.....	28
5.3.2 RTC校准.....	29
5.4 BKP寄存器.....	29
5.4.1 备份数据寄存器X (BKP_DRX) (X为1到42).....	30
5.4.2 RTC时钟校准寄存器(BKP_RTCCR).....	30
5.4.3 备份控制寄存器(BKP_CR).....	31
5.4.4 备份控制/状态寄存器(BKP_CSR).....	31
6 复位和时钟控制 (RCC)	32
6.1 复位.....	32
6.1.1 系统复位.....	32
6.1.2 电源复位.....	33
6.1.3 备份域复位.....	33
6.2 时钟.....	33
6.2.1 HSE时钟.....	35
6.2.2 HSI时钟.....	36
6.2.3 PLL.....	36
6.2.4 LSE时钟.....	36
6.2.5 LSI时钟.....	37
6.2.6 系统时钟(SYSCLK)选择.....	37
6.2.7 时钟安全系统(CSS).....	37
6.2.8 RTC时钟.....	38
6.2.9 看门狗时钟.....	38
6.2.10 时钟输出.....	38
6.3 RCC寄存器.....	38
6.3.1 时钟控制寄存器(RCC_CR).....	39
6.3.2 时钟配置寄存器(RCC_CFGR).....	41
6.3.3 时钟中断寄存器 (RCC_CIR).....	43
6.3.4 APB2 外设复位寄存器 (RCC_APB2RSTR).....	46
6.3.5 APB1 外设复位寄存器 (RCC_APB1RSTR).....	47
6.3.6 AHB外设时钟使能寄存器 (RCC_AHBENR).....	50
6.3.7 APB2外设时钟使能寄存器(RCC_APB2ENR).....	51
6.3.8 APB1 外设时钟使能寄存器(RCC_APB1ENR).....	53
6.3.9 备份域控制寄存器 (RCC_BDCR).....	56
6.3.10 控制/状态寄存器 (RCC_CSR).....	57
6.2.11 MCO分频控制寄存器器 (RCC_MCO_VAL).....	58

1 文中的缩写

1.1 寄存器描述表中使用的缩写列表

在对寄存器的描述中使用了下列缩写：

read / write (rw)	软件能读写此位。
read-only (r)	软件只能读此位。
write-only (w)	软件只能写此位，读此位将返回复位值。
read/clear (rc_w1)	软件可以读此位，也可以通过写'1'清除此位，写'0'对此位无影响。
read / clear (rc_w0)	软件可以读此位，也可以通过写'0'清除此位，写'1'对此位无影响。
read / clear by read (rc_r)	软件可以读此位；读此位将自动地清除它为'0'，写'0'对此位无影响。
read / set (rs)	软件可以读也可以设置此位，写'0'对此位无影响。
read-only write trigger (rt_w)	软件可以读此位；写'0'或'1'触发一个事件但对此位数值没有影响。
toggle (t)	软件只能通过写'1'来翻转此位，写'0'对此位无影响。
Reserved(Res.)	保留位，必须保持默认值不变

1.2 术语表

- **SWD**: 为Serial Wire Debug的首字母缩写。其是Cortex-M3内核集成的一个调试口，是基于 SWD协议的2线调试接口。
- **Word**: 字，32位长的数据或指令长度。
- **Half word**: 半字，16位长的数据或指令长度。
- **Byte**: 字节，8位数据长度
- **IAP**: in-application programming的首字母缩写。直译为在线应用编程，即用户程序可以更新自身的程序，从而达到应用升级。
- **ICP**: in-circuit programming的首字母缩写。直译为在电路编程，即用户可通过应用板上的JTAG口或SWD口对MCU的FLASH进行编程。
- **Option bytes**: 选项字节，保存在Flash中的MCU配置字节。
- **OBL**: option byte loader的首字母缩写，选项字节装载器。
- **AHB**: advanced high-performance bus的首字母缩写，直译为先进高性能总线。

2 储存器和总线架构

2.1 系统架构

系统由以下部分构成：

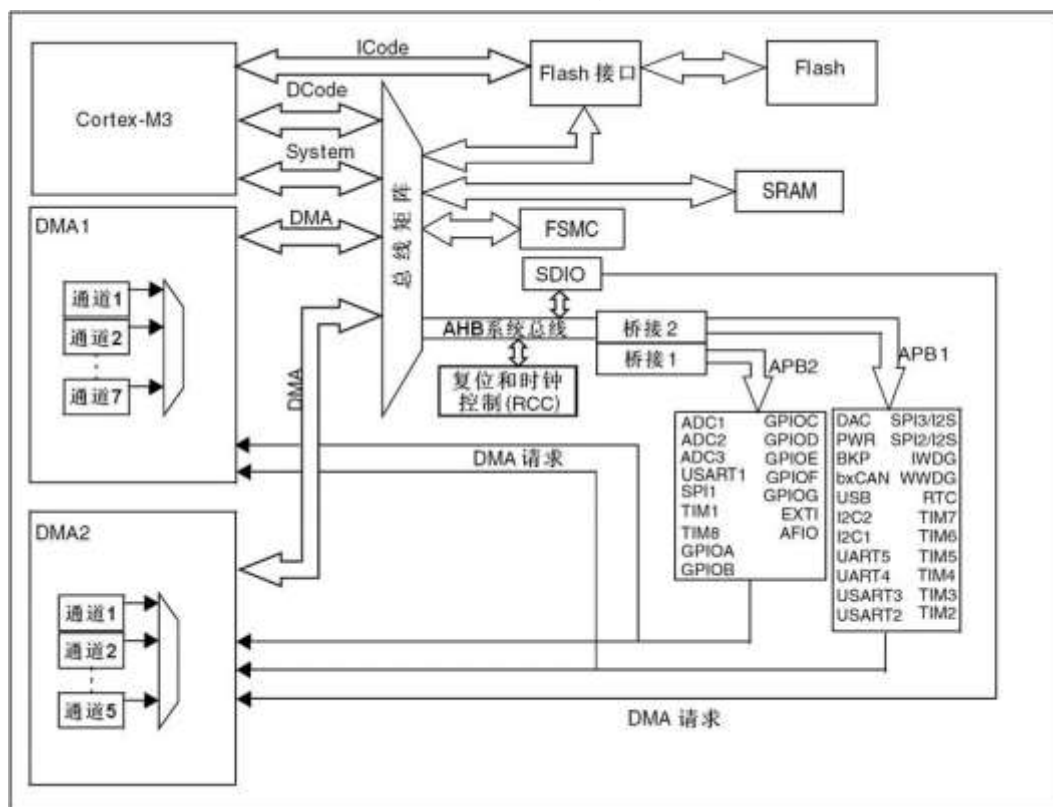
四个驱动单元：

- Cortex™-M3内核DCode总线(D-bus)，和系统总线(S-bus)
- 通用DMA1和通用DMA2

四个被动单元

- 内部SRAM
- 内部闪存存储器
- FSMC
- AHB到APB的桥(AHB2APBx)，它连接所有的APB设备

这些都是通过一个多级的AHB总线构架相互连接的，如下图图表1所示：



图表 1

ICode总线

该总线将Cortex™-M3内核的指令总线与闪存指令接口相连接。指令预取在此总线上完成。

DCode总线

该总线将Cortex™-M3内核的DCode总线与闪存存储器的数据接口相连接(常量加载和调试访问)。

系统总线

此总线连接Cortex™-M3内核的系统总线(外设总线)到总线矩阵，总线矩阵协调着内核和DMA间的访问。

DMA总线

此总线将DMA的AHB主控接口与总线矩阵相联，总线矩阵协调着CPU的DCode和DMA到SRAM、闪存和外设的访问。

总线矩阵

总线矩阵协调内核系统总线和DMA主控总线之间的访问仲裁，仲裁利用轮换算法。总线矩阵包含4个驱动部件(CPU的DCode、系统总线、DMA1总线和DMA2总线)和4个被动部件(闪存存储器接口(FLITF)、SRAM、FSMC和AHB2APB桥)。

AHB外设通过总线矩阵与系统总线相连，允许DMA访问。

AHB/APB桥(APB)

两个AHB/APB桥在AHB和2个APB总线间提供同步连接。APB1操作速度限于HCLK的二分之一，APB2操作于全速(最高216M)。

有关连接到每个桥的不同外设的地址映射请参考2.3存储器映像章节的表1。在每一次复位以后，所有除SRAM和 FLITF以外的外设都被关闭，在使用一个外设之前，必须设置寄存器RCC_AHBBENR来打开该外设的时钟。

注意： 当对APB寄存器进行8位或者16位访问时，该访问会被自动转换成32位的访问：桥会自动将8位或者32位的数据扩展以配合32位的向量。

2.2 存储器

程序存储器、数据存储器、寄存器和输入输出端口被组织在同一个4GB的线性地址空间内。数据字节以小端格式存放在存储器中。一个字里的低地址字节被认为是该字的低有效字节，而高地址字节是高有效字节。

外设寄存器的映像请参考相关章节。

可访问的存储器空间被分成8个主要块，每个块为512MB。

其他所有没有分配给片上存储器和外设的存储器空间都是保留的地址空间，请参考相应器件的数据手册中的存储器映像图。

2.3 存储器映像

表格1为AIR32F103系列芯片内置外设的起始地址。

表格 1 寄存器起始地址

起始地址	外设	总线	寄存器映像
0x5000 0000 - 0x5003 FFFF	USB OTG 全速	AHB	
0x4003 0000 - 0x4FFF FFFF	保留		
0xA000 0000 - 0xA000 0FFF	FSMC		
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	APB2	
0x4001 3C00 - 0x4001 3FFF	ADC3		
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	
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 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	

2.3.1 嵌入式SRAM

AIR32F103系列内置32/64/96K字节的SRAM，CPU能以0等待周期访问（读/写）SRAM，可以以字节、半字(16位)或全字(32位)访问。SRAM的起始地址是0x2000 0000。

2.3.2 位段

Cortex™-M3存储器映像包括两个位段(bit-band)区。这两个位段区将别名存储器区中的每个字映射到位段存储器区的一个位，在别名存储区写入一个字具有对位段区的目标位执行读-改-写操作的相同效果。

在AIR32F103里，外设寄存器和SRAM都被映射到一个位段区里，这允许执行单一的位段的写和读操作。

下面的映射公式给出了别名区中的每个字是如何对应位带区的相应位的：

$$\text{bit_word_addr} = \text{bit_band_base} + (\text{byte_offset} \times 32) + (\text{bit_number} \times 4)$$

其中：

bit_word_addr是别名存储器区中字的地址，它映射到某个目标位。

bit_band_base是别名区的起始地址。

byte_offset是包含目标位的字节在位段里的序号

bit_number是目标位所在位置(0-31)

例子：

下面的例子说明如何映射别名区中SRAM地址为0x20000300的字节中的位2：

$$0x22006008 = 0x22000000 + (0x300 \times 32) + (2 \times 4).$$

对0x22006008地址的写操作与对SRAM中地址0x20000300字节的位2执行读-改-写操作有着相同的效果。

读0x22006008地址返回SRAM中地址0x20000300字节的位2的值(0x01 或 0x00)。

请参考《Cortex™-M3技术参考手册》以了解更多有关位段的信息。

2.3.3 嵌入式闪存 (flash)

最大512K的内置闪存用于存放程序和数据，内置闪存由两部分组成，分别为主储存和信息块。

小于等于128K的产品主储存为每页1K字节（表格2）；

大于128K的产品主储存每页为2K字节（表格3）。

闪存存储器接口的特性为：

- 带预取缓冲器的读接口(每字为2×64位)
- 选择字节加载器
- 闪存编程/擦除操作
- 访问/写保护

表格 2 128K以内产品闪存分布

模块	名称	地址	大小(字节)
主存储块	页 0	0x0800 0000 - 0x0800 03FF	1K
	页 1	0x0800 0400 - 0x0800 07FF	1K
	页 2	0x0800 0800 - 0x0800 0BFF	1K
	页 3	0x0800 0C00 - 0x0800 0FFF	1K
	页 4	0x0800 1000 - 0x0800 13FF	1K

	页 127	0x0801 FC00 - 0x0801 FFFF	1K
信息块	系统存储器	0x1FFF F000 - 0x1FFF F7FF	2K
	选择字节	0x1FFF F800 - 0x1FFF F80F	16
闪存存储器 接口寄存器	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FALSH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	保留	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRP	0x4002 2020 - 0x4002 2023	4

表格 3 128K以上产品闪存分布

模块	名称	地址	大小(字节)
主存储块	页 0	0x0800 0000 - 0x0800 07FF	2K
	页 1	0x0800 0800 - 0x0800 0FFF	2K
	页 2	0x0800 1000 - 0x0800 17FF	2K
	页 3	0x0800 1800 - 0x0800 1FFF	2K

	页 255	0x0807 F800 - 0x0807 FFFF	2K
信息块	系统存储器	0x1FFF F000 - 0x1FFF F7FF	2K
	选择字节	0x1FFF F800 - 0x1FFF F80F	16

闪存存储器 接口寄存器	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	保留	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRP	0x4002 2020 - 0x4002 2023	4

闪存的指令和数据访问是通过AHB总线完成的。预取模块是用于通过ICode总线读取指令的。仲裁是作用在闪存接口，并且DCode总线上的数据访问优先。

读访问可以有以下配置选项：

- 等待时间：可以随时更改的用于读取操作的等待状态的数量。
- 预取缓冲区(2个64位)：在每一次复位以后被自动打开，由于每个缓冲区的大小(64位)与闪存的带宽相同，因此只通过需一次读闪存的操作即可更新整个缓冲区的内容。由于预取缓冲区的存在，CPU可以工作在更高的主频。CPU每次取指 多为32位的字，取一条指令时，下一条指令已经在缓冲区中等待。
- 半周期：用于功耗优化。

注：

1. 这些选项应与闪存存储器的访问时间一起使用。等待周期体现了系统时钟(SYSCLK)频率与闪存访问时间的关系：

Flash Delay等级	HCLK(MHz)	
	Voltage Range 2.3V - 3.6V	Voltage Range 2.0V - 2.3V
0	0 < HCLK ≤ 108	0 < HCLK ≤ 32
1	108 < HCLK ≤ 216	32 < HCLK ≤ 64
2	-	64 < HCLK ≤ 128
3	-	128 < HCLK ≤ 192
4	-	192 < HCLK ≤ 216

表格 4

2. 半周期配置不能与使用了预分频器的AHB一起使用，时钟系统应该等于HCLK时钟。该特性只能用在时钟频率为8MHz或低于8MHz时，可以直接使用的内部RC振荡器(HSI)，或者是主振荡器(HSE)，但不能用PLL。

3. 当AHB预分频系数不为1时，必须置预取缓冲区处于开启状态。

4. 只有在系统时钟(SYSCLK)小于24MHz并且没有打开AHB的预分频器(即HCLK必须等于SYSCLK)时，才能执行预取缓冲器的打开和关闭操作。一般而言，在初始化过程中执行预取缓冲器的打开和关闭操作，这时微控制器的时钟由8MHz的内部RC振荡器(HSI)提供。

5. 使用DMA：DMA在DCode总线上访问闪存存储器，它的优先级比ICode上的取指高。DMA在每次传送完成后具有一个空余的周期。有些指令可以和DMA传输一起执行。

编程和擦除闪存

闪存编程一次可以写入16位(半字)。

闪存擦除操作可以按页面擦除或完全擦除(全擦除)。全擦除不影响信息块。

为了确保不发生过度编程，闪存编程和擦除控制器块是由一个固定的时钟控制的。

写操作(编程或擦除)结束时可以触发中断。仅当闪存控制器接口时钟开启时，此中断可以用来从WFI模式退出。

2.4 启动配置 (BOOT)

在AIR32F103里，可以通过BOOT[1:0]引脚选择三种不同启动模式。

启动模式选择引脚		启动模式	说明
BOOT1	BOOT0		
X	0	主闪存存储器	主闪存存储器被选为启动区域
0	1	系统存储器	系统存储器被选为启动区域
1	1	内置SRAM	内置SRAM被选为启动区域

表格 5

在系统复位后，SYSCLK的第4个上升沿，BOOT引脚的值将被锁存。用户可以通过设置BOOT1和BOOT0引脚的状态，来选择在复位后的启动模式。

在从待机模式退出时，BOOT引脚的值将被重新锁存；因此，在待机模式下BOOT引脚应保持为需要的启动配置。在启动延迟之后，CPU从地址0x0000 0000获取堆栈顶的地址，并从启动存储器的0x0000 0004指示的地址开始执行代码。

因为固定的存储器映像，代码区始终从地址0x0000 0000开始(通过ICode和DCode总线访问)，而数据区(SRAM)始终从地址0x2000 0000开始(通过系统总线访问)。Cortex-M3的CPU始终从 ICode总线获取复位向量，即启动仅适合于从代码区开始(典型地从Flash启动)。

AIR32F103微控制器实现了一个特殊的机制，系统可以不仅仅从Flash存储器或系统存储器启动，还可以从内置SRAM启动。

根据选定的启动模式，主闪存存储器、系统存储器或SRAM可以按照以下方式访问：

- 从主闪存存储器启动：主闪存存储器被映射到启动空间(0x0000 0000)，但仍然能够在它原有的地址(0x0800 0000)访问它，即闪存存储器的内容可以在两个地址区域访问，0x0000 0000或0x0800 0000。
- 从系统存储器启动：系统存储器被映射到启动空间(0x0000 0000)，但仍然能够在它原有的地址(互联型产品原有地址为0x1FFF B000，其它产品原有地址为0x1FFF F000)访问它。
- 从内置SRAM启动：只能在0x2000 0000开始的地址区访问SRAM。

注意： 当从内置SRAM启动，在应用程序的初始化代码中，必须使用NVIC的异常表和偏移寄存器，从新映射向量表之SRAM中。

内嵌的自举程序 (ISP)

内嵌的自举程序存放在系统存储区，固化在ROM中，用于通过可用的串行接口对闪存存储器进行重新编程：目前发布的AIR32F103系列可以通过USART1接口启用自举程序。

3 CRC计算单元（CRC）

3.1 CRC简介

循环冗余校验(CRC)计算单元是根据固定的生成多项式得到任一32位全字的CRC计算结果。在其他的应用中，CRC技术主要应用于核实数据传输或者数据存储的正确性和完整性。标准EN/IEC 60335-1即提供了一种核实闪存存储器完整性的方法。CRC计算单元可以在程序运行时计算出软件的标识，之后与在连接时生成的参考标识比较，然后存放在指定的存储器空间。

3.2 CRC主要特性

- 支持CRC-16/32两种CRC计算方式
- 支持CRC-16多项式：0x8005和0x1021
- 支持CRC-32多项式：0x04C11DB7
- 数据按字节输入
- 输入数据可配置由硬件进行大小端翻转
- 输出数据可配置由硬件进行大小端翻转
- 支持指定CRC计算初始值

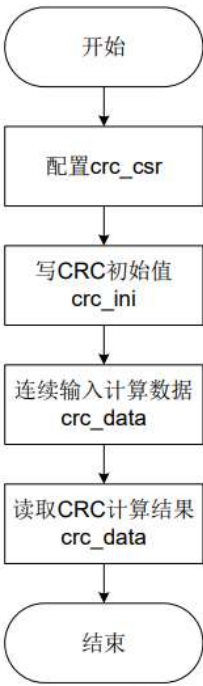
3.3 CRC功能描述

CRC单元包含一个32位的数据输入寄存器

- 对其进行写操作时，作为 8 位的输入寄存器（仅低 8 位有效）
- 对其进行读操作时，作为输出寄存器，其有效位宽由 CRC 控制状态寄存器（CRC_CSR）决定。

根据需要配置完成控制状态寄存器（CRC_CSR）和 CRC 计算初始值寄存器（CRC_INI）后对数据寄存器进行连续写操作，需要校验的数据写操作完成后，对 CRC 数据寄存器进行读操作获取CRC校验值。

CRC操作流程图如下图表2:



图表 2

3.4 CRC寄存器

CRC计算单元包括3个数据寄存器和2个控制寄存器

地址范围	基地址	外设	总线
0x4002 3000 - 0x4002 33FF	0x4001_2000	CRC	AHB

寄存器列表

偏移地址	寄存器名称	宽度(bit)	复位值
0x00	CRC_DR	32	0xFFFF FFFF
0x04	CRC_IDR	8	0x0000 0000
0x08	CRC_CR	1	0x0000 0000
0x12	CRC_CSR	32	0x0000 0000
0x16	CRC_INI	32	0x0000 0000

3.4.1 数据寄存器(CRC_DR)

地址偏移: 0x00
复位值: 0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

DR[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															
rw															

31:0	数据寄存器位 写入CRC计算器的新数据时，作为输入寄存器 读取时返回CRC计算的结果
------	--

3.4.2 独立数据寄存器（CRC_IDR）

地址偏移：0x04
复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留								IDR[7:0]							
								rw							

31:8	保留
7:0	通用8位数据寄存器位 可用于临时存放1字节的数据。 寄存器CRC_CR的RESET位产生的CRC复位对本寄存器没有影响

此寄存器不参与CRC计算，可以存放任何数据。

3.4.3 控制寄存器（CRC_CR）

地址偏移：0x08
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														RESET	
														w	

31:1	保留
1:0	RESET位 复位CRC计算单元，设置数据寄存器为0xFFFF FFFF。 只能对该位写'1'，它由硬件自动清'0'。

3.4.4 控制状态寄存器（CRC_CSR）

地址偏移：0x12
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				xor_out_sel			rev_out_sel			rev_in_sel			type_sel	poly_sel	
				rw			rw			rw			rw	rw	

31:5	保留
4	xor_out_sel位 CRC计算结果和0xffff进行异或；（此步骤在rev_out_sel之后进行） 1：和 0xffff 进行异或； 0：计算结果直接输出。
3	CRC计算结果高低位反转； 1：反转； 0：不反转。
2	CRC8-bit输入大小端反转进行计算，如bit7作为bit0参与运算，bit6作为bit1，以此类推。 1：反转； 0：不反转
1	CRC类型选择； 1：CRC32； 0：CRC16。
0	CRC16 的多项式选择，选择 CRC32时，该位无效。 1：0x1021； 0：0x8005。

3.4.5 初始值寄存器（CRC_INI）

地址偏移：0x16
复位值：0x0000 0000

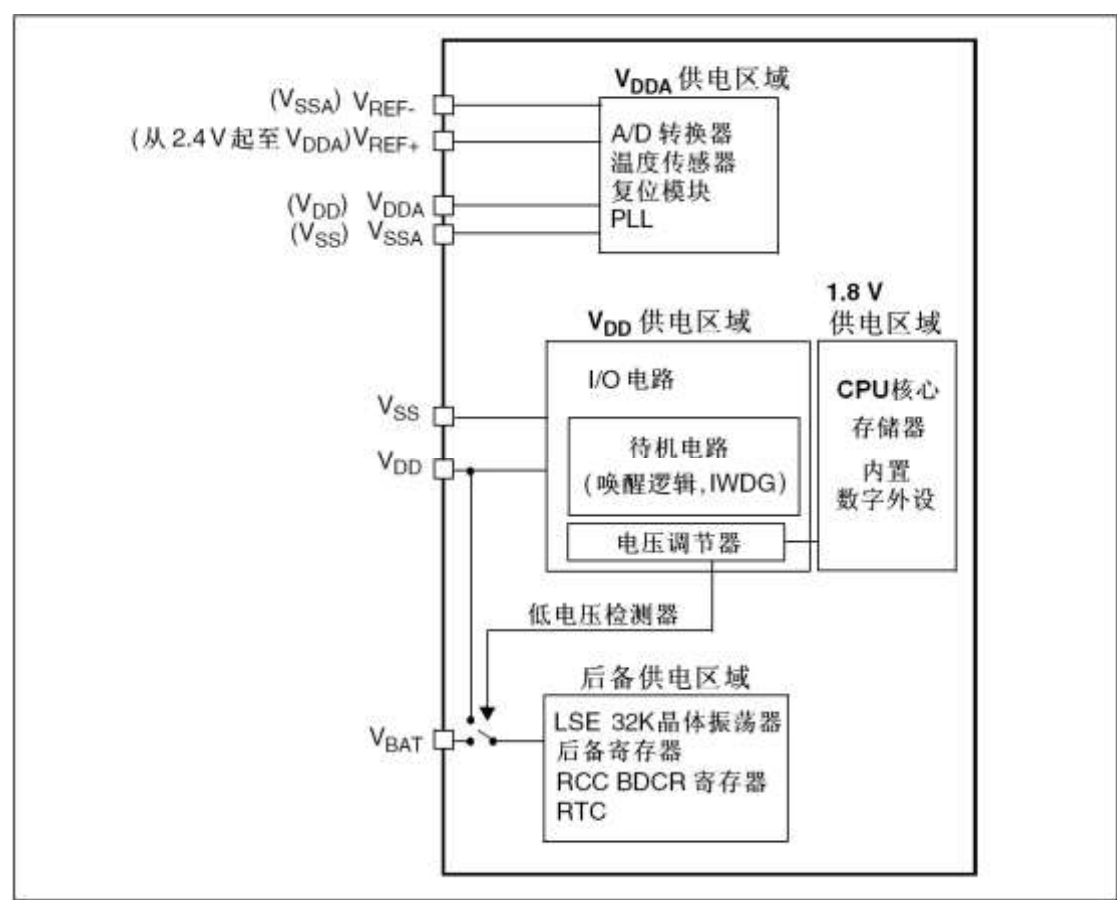
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRC_INI [31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_INI [15:0]															
rw															

31:0	CRC计算的初始值； 计算CRC16时，低16位有效。
------	--------------------------------

4 电源控制（PWR）

4.1 电源

AIR32的工作电压(VDD)为2.0 ~ 3.6V。通过内置的电压调节器提供所需的1.8V电源。
当主电源VDD掉电后，通过VBAT脚为实时时钟(RTC)和备份寄存器提供电源。



图表 3

4.1.1 独立的A/D转换器供电和参考电压

为了提高转换的精确度，ADC使用一个独立的电源供电，过滤和屏蔽来自PCB上的毛刺干扰。

- ADC的电源引脚为VDDA
- 独立的电源地VSSA

如果有VREF-引脚(根据封装而定)，它必须连接到VSSA。

100脚和144脚封装：

为了确保输入为低压时获得更好精度，用户可以连接一个独立的外部参考电压ADC到VREF+和VREF-脚上。在VREF+的电压范围为2.4V ~ VDDA。

64脚或更少封装：

没有VREF+和VREF-引脚，他们在芯片内部与ADC的电源(VDDA)和地(VSSA)相联。

4.1.2 电池备份区域

使用电池或其他电源连接到VBAT脚上，当VDD断电时，可以保存备份寄存器的内容和维持RTC的功能。当正常工作状态时，备份寄存器和RTC是由VDD供电的，当VDD断电时才通过VBAT引脚供电。

VBAT脚也为RTC、LSE振荡器和PC13至PC15供电，这保证当主要电源被切断时RTC能继续工作。切换到VBAT供电由复位模块中的掉电复位功能控制。

如果应用中没有使用外部电池，VBAT必须连接到VDD引脚上。

注意：

在VDD上升阶段($t_{RSTTEMPO}$)或者探测到PVD之后，VBAT和VDD之间的电源开关仍会保持连接在VBAT。在VDD上升阶段，如果VDD在小于 $t_{RSTTEMPO}$ 的时间内达到稳定状态(关于 $t_{RSTTEMPO}$ 可参考数据手册中的相关部分)，且 $VDD > VBAT + 0.6V$ 时，电流可能通过VDD和VBAT之间的内部二极管注入到VBAT。

如果与VBAT连接的电源或者电池不能承受这样的注入电流，强烈建议在外部VBAT和电源之间连接一个低压降二极管。

如果在应用中没有外部电池，建议VBAT在外部通过一个100nF的陶瓷电容与VDD相连。

当备份区域由VDD(内部模拟开关连到VDD)供电时，下述功能可用：

- PC14和PC15可以用于GPIO或LSE引脚
- PC13可以作为通用I/O口、TAMPER引脚、RTC校准时钟、RTC闹钟或秒输出(参见第5章：备份寄存器(BKP))

注：因为模拟开关只能通过少量的电流(3mA)，在输出模式下使用PC13至PC15的I/O口功能是有限制的：速度必须限制在2MHz以下，大负载为30pF，而且这些I/O口绝对不能当作电流源(如驱动LED)。

当后备区域由VBAT供电时(VDD消失后模拟开关连到VBAT)，可以使用下述功能：

- PC14和PC15只能用于LSE引脚
- PC13可以作为TAMPER引脚、RTC闹钟或秒输出

4.1.3 电压调节器

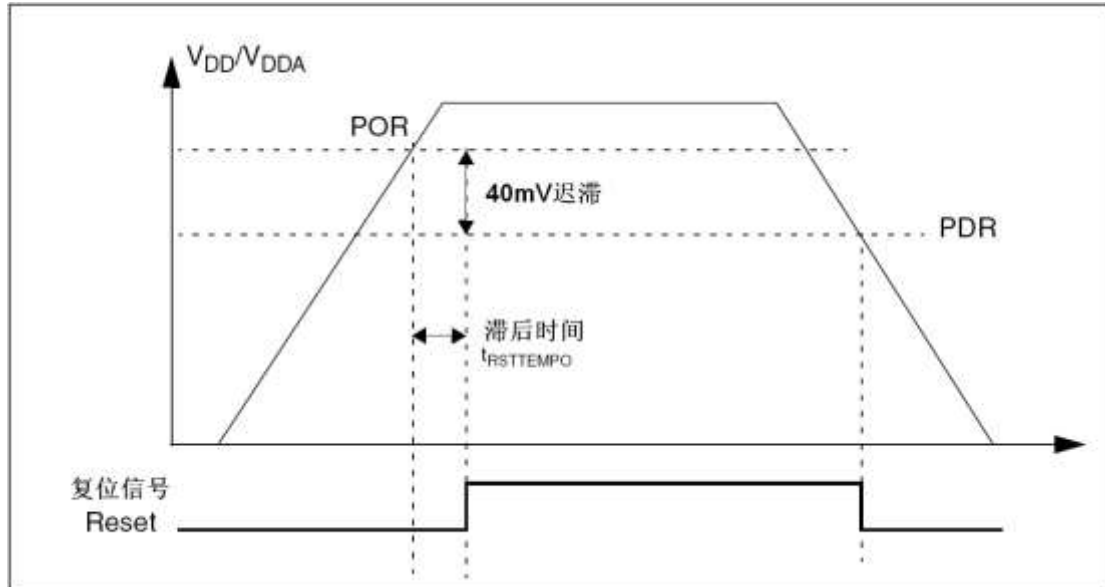
调压器有三个操作模式：主模式(MR)、低功耗模式(LPR)和关断模式

- 主模式(MR)用于正常的运行操作
 - 低功耗模式(LPR)用于CPU的停机模式（这个模式下寄存器和SRAM内容不丢失）
 - 关断模式用于CPU的待机模式：调压器的输出为高阻状态，内核电路的供电切断，调压器处于零消耗状态(但寄存器和SRAM的内容将丢失)，备份区域内容可保持
- 该调压器在复位后始终处于工作状态，在待机模式下关闭处于高阻输出。

4.2 电源管理器

4.2.1 上电复位(POR)和掉电复位(PDR)

AIR32F103全系产品内部都有一个完整的上电复位(POR)和掉电复位(PDR)电路，当供电电压达到2V时系统既能正常工作。当 V_{DD}/V_{DDA} 低于指定的限位电压 V_{POR}/V_{PDR} 时，系统保持为复位状态，而无需外部复位电路。



图表 4

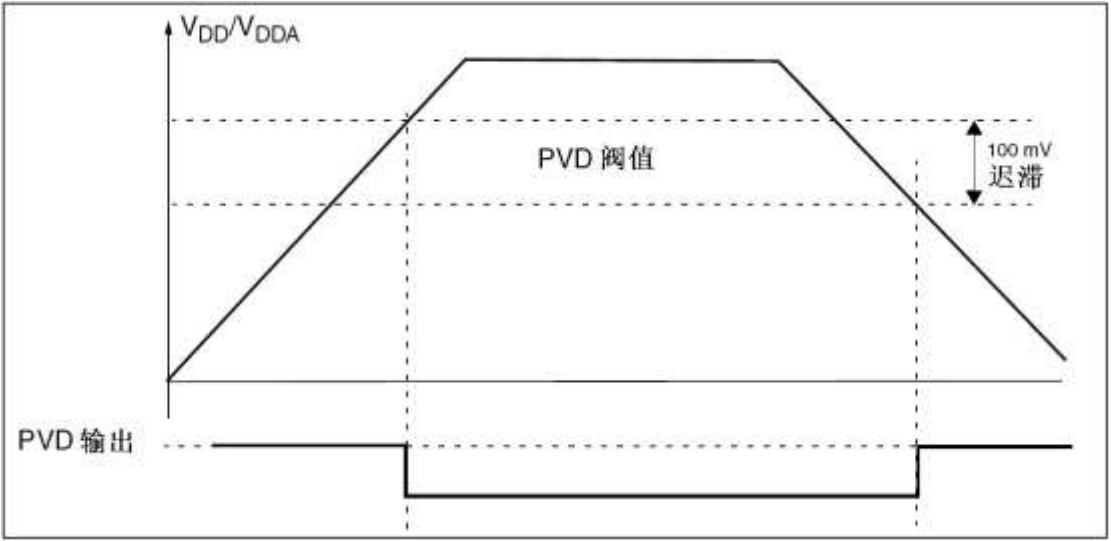
4.2.2 可编程电压监测器(PVD)

AIR32F103还有一个可编程电压监测器(PVD)，它监视 V_{DD}/V_{DDA} 供电并与阈值 V_{PVD} 比较，当 V_{DD} 低于或高于阈值 V_{PVD} 时产生中断，中断处理程序可以发出警告信息或将微控制器转入安全模式。PVD功能需要通过程序开启。用户可以利用PVD对 V_{DD} 电压与电源控制寄存器(PWR_CR)中的PLS[2:0]位进行比较来监控电源，这几位选择监控电压的阈值。

通过设置PVDE位来使能PVD。

电源控制/状态寄存器(PWR_CSR)中的PVD0标志用来表明 V_{DD} 是高于还是低于PVD的电压阈值。该事件在内部连接到外部中断的第16线，如果该中断在外部中断寄存器中是使能的，该事件就会产生中断。当 V_{DD} 下降到PVD阈值以下和(或)当 V_{DD} 上升到PVD阈值之上时，根据外部中断第16线的上升/下降边沿触发设置，就会产生PVD中断。例如，这一特性可用于用于执行紧急关闭任务。

图表 5



4.3 低功耗模式

在系统或电源复位以后，微控制器处于运行状态。当CPU不需继续运行时，可以利用多种低功耗模式来节省功耗，例如等待某个外部事件时。用户需要根据低电源消耗、快速启动时间和可用的唤醒源等条件，选定一个最佳的低功耗模式。

AIR32F103采用先进的28nm工艺生产，在相同工作条件下，相比其他竞品功耗仅有一半左右。

AIR32F103有三种低功耗模式：

- 睡眠模式sleep(Cortex™-M3内核停止，所有外设包括Cortex-M3核心的外设，如NVIC、系统时钟(SysTick)等仍在运行)
 - 停止模式stop(所有的时钟都已停止)
 - 待机模式standby (1.8V电源关闭)
- 此外，在运行模式下，可以通过以下方式中的一种降低功耗：
- 降低系统时钟
 - 关闭APB和AHB总线上未被使用的外设时钟。

模式	进入	唤醒	对 1.8V 区域 时钟的影响	对 V _{DD} 区域 时钟的影响	电压调节器
睡眠睡眠 (SLEEP-NOW 或 SLEEP-ON- EXIT)	WFI	任一中断	CPU 时钟 关，对其他 时钟和 ADC 时钟无影响	无	开
	WFE	唤醒事件			
停机	PDDS 和 LPDS 位 +SLEEPDEEP 位 +WFI 或 WFE	任一外部中断(在外 部中断寄存器中设 置)	关闭所有 1.8V 区域的 时钟	HSI 和 HSE 的振荡器关 闭	开启或处于低 功耗模式(依据 电源控制寄存 器(PWR_CR) 的设置)

待机	PDDS 位 +SLEEPDEEP 位 +WFI 或 WFE	WKUP 引脚的上升沿、RTC 闹钟事件、 NRST 引脚上的外部复位、IWDG 复位			关
----	--------------------------------------	--	--	--	---

表格 6

4.3.1 降低系统时钟

在运行模式下，通过对预分频寄存器进行编程，可以降低任意一个系统时钟(SYSCLK、HCLK、PCLK1、PCLK2)的速度。进入睡眠模式前，也可以利用预分频器来降低外设的时钟。不同频率下功耗数据详见数据手册。

4.3.2 外部时钟的控制

在运行模式下，任何时候都可以通过停止为外设和内存提供时钟(HCLK和PCLKx)来减少功耗。为了在睡眠模式下更多地减少功耗，可在执行WFI或WFE指令前关闭所有外设的时钟。通过设置 AHB 外设时钟使能寄存器 (RCC_AHBENR)、APB2 外设时钟使能寄存器 (RCC_APB2ENR)和APB1外设时钟使能寄存器(RCC_APB1ENR)来开关各个外设模块的时钟。每个外设的耗电数据在数据手册中也有详细介绍。

4.3.3 睡眠模式

进入睡眠模式

通过执行WFI或WFE指令进入睡眠状态。根据Cortex™-M3系统控制寄存器中的SLEEPONEXIT位的值，有两种选项可用于选择睡眠模式进入机制：

- SLEEP-NOW：如果SLEEPONEXIT位被清除，当WFI或WFE被执行时，微控制器立即进入睡眠模式。
- SLEEP-ON-EXIT：如果SLEEPONEXIT位被置位，系统从 低优先级的中断处理程序中退出时，微控制器就立即进入睡眠模式。

在睡眠模式下，所有的I/O引脚都保持它们在运行模式时的状态。关于如何进入睡眠模式，更多的细节参考表9和表10。

退出睡眠模式

如果执行WFI指令进入睡眠模式，任意一个被嵌套向量中断控制器响应的外设中断都能将系统从睡眠模式唤醒。

如果执行WFE指令进入睡眠模式，则一旦发生唤醒事件时，微处理器都将从睡眠模式退出。唤醒事件可以通过下述方式产生：

- 在外设控制寄存器中使能一个中断，而不是在NVIC(嵌套向量中断控制器)中使能，并且在Cortex-M3系统控制寄存器中使能SEVONPEND位。当MCU从WFE中唤醒后，外设的中断挂起位和外设的NVIC中断通道挂起位(在NVIC中断清除挂起寄存器中)必须被清除。

- 配置一个外部或内部的EXIT线为事件模式。当MCU从WFE中唤醒后，因为与事件线对应的挂起位未被设置，不必清除外设的中断挂起位或外设的NVIC中断通道挂起位。

该模式唤醒所需的时间短，因为没有时间损失在中断的进入或退出上。

关于如何退出睡眠模式，更多的细节参考表7和表8。

SLEEP-NOW模式	说明
进入	在以下条件下执行WFI(等待中断)或WFE(等待事件)指令： <ul style="list-style-type: none"> – SLEEPDEEP = 0 和 – SLEEPONEXIT = 0 参考Cortex-M3系统控制寄存器。
退出	如果执行WFI进入睡眠模式： 中断：参考中断向量表(表54) 如果执行WFE进入睡眠模式：唤醒事件：参考唤醒事件管理(第9.2.3节)
唤醒延时	无

表格 7

SLEEP-ON_EXIT模式	说明
进入	在以下条件下执行WFI指令： <ul style="list-style-type: none"> – SLEEPDEEP = 0和 – SLEEPONEXIT = 1 参考Cortex™-M3系统控制寄存器
退出	中断：参考中断向量表(表54)
唤醒延时	无

表格 8

4.3.4 停止模式

停止模式是在Cortex™-M3的深睡眠模式基础上结合了外设的时钟控制机制，在停止模式下电压调节器可运行在正常或低功耗模式。此时在1.8V供电区域的所有时钟都被停止，PLL、HSI和HSE RC振荡器的功能被禁止，SRAM和寄存器内容被保留下来。在停止模式下，所有的I/O引脚都保持它们在运行模式时的状态。

进入停止模式

关于如何进入停止模式，详见表9。

在停止模式下，通过设置电源控制寄存器(PWR_CR)的LPDS位使内部调节器进入低功耗模式，能够降低更多的功耗。

如果正在进行闪存编程，直到对内存访问完成，系统才进入停止模式。

如果正在进行对APB的访问，直到对APB访问完成，系统才进入停止模式。

可以通过对独立的控制位进行编程，可选择以下功能：

- 独立看门狗(IWDG)：可通过写入看门狗的键寄存器或硬件选择来启动IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。
- 实时时钟(RTC)：通过备份域控制寄存器(RCC_BDCR)的RTCEN位来设置。
- 内部RC振荡器(LSI RC)：通过控制/状态寄存器(RCC_CSR)的LSION位来设置。
- 外部32.768kHz振荡器(LSE)：通过备份域控制寄存器(RCC_BDCR)的LSEON位设置。在停

止模式下，如果在进入该模式前ADC和DAC没有被关闭，那么这些外设仍然消耗电流。通过设置寄存器ADC_CR2的ADON位和寄存器DAC_CR的ENx位为0可关闭这2个外设。

退出停止模式

关于如何退出停止模式，详见下表。

当一个中断或唤醒事件导致退出停止模式时，HSI RC振荡器被选为系统时钟。

当电压调节器处于低功耗模式下，当系统从停止模式退出时，将会有一段额外的启动延时。如果在停止模式期间保持内部调节器开启，则退出启动时间会缩短，但相应的功耗会增加。

停止模式	说明
进入	在以下条件下执行WFI(等待中断)或WFE(等待事件)指令： – 设置Cortex-M3系统控制寄存器中的SLEEPDEEP位 – 清除电源控制寄存器(PWR_CR)中的PDDS位 – 通过设置PWR_CR中LPDS位选择电压调节器的模式 注：为了进入停止模式，所有的外部中断的请求位(挂起寄存器(EXTI_PR))和RTC的闹钟标志都必须被清除，否则停止模式的进入流程将会被跳过，程序继续运行。
退出	如果执行WFI进入停止模式：设置任一外部中断线为中断模式(在NVIC中必须使能相应的外部中断向量)。参见中断向量表(表54)。 如果执行WFE进入停止模式：设置任一外部中断线为事件模式。参见唤醒事件管理(第9.2.3节)。
唤醒延时	HSI RC唤醒时间 + 电压调节器从低功耗唤醒的时间。

表格 9

4.3.5 待机模式

待机模式可实现系统的 低功耗。该模式是在Cortex-M3深睡眠模式时关闭电压调节器。整个1.8V供电区域被断电。PLL、HSI和HSE振荡器也被断电。SRAM和寄存器内容丢失。只有备份的寄存器和待机电路维持供电。

进入待机模式

关于如何进入待机模式，详见表12。

可以通过设置独立的控制位，选择以下待机模式的功能：

- 独立看门狗(IWDG)：可通过写入看门狗的键寄存器或硬件选择来启动IWDG。一旦启动了独立看门狗，除了系统复位，它不能再被停止。
- 实时时钟(RTC)：通过备用区域控制寄存器(RCC_BDCR)的RTCEN位来设置。
- 内部RC振荡器(LSI RC)：通过控制/状态寄存器(RCC_CSR)的LSION位来设置。
- 外部32.768kHz振荡器(LSE)：通过备用区域控制寄存器(RCC_BDCR)的LSEON位设置。

退出待机模式

当一个外部复位(NRST引脚)、IWDG复位、WKUP引脚上的上升沿或RTC闹钟事件的上升沿发生时(见图154：简化的RTC框图)，微控制器从待机模式退出。从待机唤醒后，除了电源控制/状态

寄存器(PWR_CSR)(见第4.4.2节)，所有寄存器被复位。

从待机模式唤醒后的代码执行等同于复位后的执行(采样启动模式引脚、读取复位向量等)。电源控制/状态寄存器(PWR_CSR)(见第4.4.2节)将会指示内核由待机状态退出。

关于如何退出待机模式，详见下表。

待机模式	说明
进入	在以下条件下执行WFI(等待中断)或WFE(等待事件)指令： -设置Cortex™-M3系统控制寄存器中的SLEEPDEEP位 -设置电源控制寄存器(PWR_CR)中的PDDS位 -清除电源控制/状态寄存器(PWR_CSR)中的WUF位
退出	WKUP引脚的上升沿、RTC闹钟事件的上升沿、NRST引脚上外部复位、IWDG复位。
唤醒延时	HSI RC唤醒时间 + 电压调节器从低功耗唤醒的时间。

表格 10

待机模式下的输入/输出端口状态

在待机模式下，所有的I/O引脚处于高阻态，除了以下的引脚：

- 复位引脚(始终有效)
- 当被设置为防侵入或校准输出时的TAMPER引脚
- 被使能的唤醒引脚调试模式

默认情况下，如果在进行调试微处理器时，使微处理器进入停止或待机模式，将失去调试连接。这是因为Cortex™-M3的内核失去了时钟。

然而，通过设置DBGMCU_CR寄存器中的某些配置位，可以在使用低功耗模式下调试软件。

4.3.6 低功耗模式下的自动唤醒

RTC可以在不需要依赖外部中断的情况下唤醒低功耗模式下的微控制器(自动唤醒模式)。RTC提供一个可编程的时间基数，用于周期性从停止或待机模式下唤醒。通过对备份区域控制寄存器(RCC_BDCR)的RTCSEL[1:0]位的编程，三个RTC时钟源中的二个时钟源可以选作实现此功能。

- 低功耗32.768kHz外部晶振(LSE)

该时钟源提供了一个低功耗且精确的时间基准。(在典型情形下消耗小于1μA)

- 低功耗内部RC振荡器(LSI RC)

使用该时钟源，节省了一个32.768kHz晶振的成本。但是RC振荡器将少许增加电源消耗。

为了用RTC闹钟事件将系统从停止模式下唤醒，必须进行如下操作：

- 配置外部中断线17为上升沿触发。
- 配置RTC使其可产生RTC闹钟事件。

如果要从待机模式中唤醒，不必配置外部中断线17。

4.4 电源控制寄存器（PWR）

电源控制共包含2个控制寄存器

地址范围	基地址	外设	总线
------	-----	----	----

0x4000 7000 - 0x4000 73FF	0x4000 7000	电源控制(PWR)	APB1
---------------------------	-------------	-----------	------

寄存器列表

偏移地址	寄存器名称	宽度(bit)	复位值
0x00	PWR_CR	32	0x0000 0000
0x04	PWR_CSR	32	0x0000 0000

4.4.1 电源控制寄存器 (PWR_CR)

地址偏移: 0x00

复位值: 0x0000 0000 (从待机模式唤醒时清除)

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		DBP		PLS[2:0]		PVDE		CSBF		CWUF		PDDS		LPDS	
		rw		rw		rw		rw		rw		rw		rw	
31:9		保留													
8		DBP ：取消后备区域的写保护 在复位后，RTC和后备寄存器处于被保护状态以防意外写入。设置这位允许写入这些寄存器。 0：禁止写入RTC和后备寄存器 1：允许写入RTC和后备寄存器 注：如果RTC的时钟是HSE/128，该位必须保持为'1'。													
7:5		PLS[2:0] ：PVD电平选择 这些位用于选择电源电压监测器的电压阈值 000：2.2V 100：2.6V 001：2.3V 101：2.7V 010：2.4V 110：2.8V 011：2.5V 注：详细说明参见数据手册中的电气特性部 111：2.9V 分。													
4		PVDE ：电源电压监测器(PVD)使能 0：禁止PVD 1：开启PVD													
3		CSBF ：清除待机位始终读出为0 0：无功效 1：清除SBF待机位(写)													
2		CWUF ：清除唤醒位始终读出为0 0：无功效 1：2个系统时钟周期后清除WUF唤醒位(写)													

1	PDDS : 掉电深睡眠与LPDS位协同操作 0: 当CPU进入深睡眠时进入停机模式, 调压器的状态由LPDS位控制。 1: CPU进入深睡眠时进入待机模式。
0	LPDS : 深睡眠下的低功耗 PDDS=0时, 与PDDS位协同操作 0: 在停机模式下电压调压器开启 1: 在停机模式下电压调压器处于低功耗模式

4.4.2 电源控制/状态寄存器(PWR_CSR)

地址偏移: 0x04
复位值: 0x0000 0000 (从待机模式唤醒时不被清除)
与标准的APB读相比, 读此寄存器需要额外的APB周期

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					EWUP		保留			PVDD		SBF		WUF	
					rw					rw		rw		rw	

31:9	保留
8	EWUP : 使能WKUP引脚 0: WKUP引脚为通用I/O。WKUP引脚上的事件不能将CPU从待机模式唤醒 1: WKUP引脚用于将CPU从待机模式唤醒, WKUP引脚被强置为输入下拉的配置 (WKUP引脚上的上升沿将系统从待机模式唤醒) 注: 在系统复位时清除这一位。
7:3	保留。始终读为0。
2	PVDO : PVD输出当PVD被PVDE位使能后该位才有效 0: VDD/VDDA高于由PLS[2:0]选定的PVD阈值 1: VDD/VDDA低于由PLS[2:0]选定的PVD阈值 注: 在待机模式下PVD被停止。因此, 待机模式后或复位后, 直到设置PVDE位之前, 该位为0。
1	SBF : 待机标志 该位由硬件设置, 并只能由POR/PDR(上电/掉电复位)或设置电源控制寄存器(PWR_CR)的CSBF位清除。 0: 系统不在待机模式 1: 系统进入待机模式

0	<p>WUF：唤醒标志</p> <p>该位由硬件设置，并只能由POR/PDR(上电/掉电复位)或设置电源控制寄存器(PWR_CR)的CWUF位清除。</p> <p>0：没有发生唤醒事件</p> <p>1：在WKUP引脚上发生唤醒事件或出现RTC闹钟事件。</p> <p>注：当WKUP引脚已经是高电平时，在(通过设置EWUP位)使能WKUP引脚时，会检测到一个额外的事件。</p>
---	---

5 备份寄存器 (BKP)

5.1 BKP简介

备份寄存器是42个16位的寄存器，可用来存储84个字节的用户应用程序数据。他们处在备份域里，当VDD电源被切断，他们仍然由VBAT维持供电。当系统在待机模式下被唤醒，或系统复位或电源复位时，他们也不会被复位。

此外，BKP控制寄存器用来管理侵入检测和RTC校准功能。

复位后，对备份寄存器和RTC的访问被禁止，并且备份域被保护以防止可能存在的意外的写操作。执行以下操作可以使能对备份寄存器和RTC的访问。

- 通过设置寄存器RCC_APB1ENR的PWREN和BKPEN位来打开电源和后备接口的时钟
- 电源控制寄存器(PWR_CR)的DBP位来使能对后备寄存器和RTC的访问。

5.2 BKP特性

- AIR32F103拥有84字节数据后备寄存器
- 用来管理防侵入检测并具有中断功能的状态/控制寄存器
- 用来存储RTC校验值的校验寄存器。
- 在PC13引脚(当该引脚不用于侵入检测时)上输出RTC校准时钟，RTC闹钟脉冲或者秒脉冲

5.3 BKP功能描述

5.3.1 侵入检测

当TAMPER引脚上的信号从0变成1或者从1变成0(取决于备份控制寄存器BKP_CR的TPAL位)，会产生一个侵入检测事件。侵入检测事件将所有数据备份寄存器内容清除。

然而为了避免丢失侵入事件，侵入检测信号是边沿检测的信号与侵入检测允许位的逻辑与，从而在侵入检测引脚被允许前发生的侵入事件也可以被检测到。

- 当TPAL=0时：如果在启动侵入检测TAMPER引脚前(通过设置TPE位)该引脚已经为高电

平，一旦启动侵入检测功能，则会产生一个额外的侵入事件(尽管在TPE位置'1'后并没有出现上升沿)。

- 当TPAL=1时：如果在启动侵入检测引脚TAMPER前(通过设置TPE位)该引脚已经为低电平，一旦启动侵入检测功能，则会产生一个额外的侵入事件(尽管在TPE位置'1'后并没有出现下降沿)。

设置BKP_CSR寄存器的TPIE位为'1'，当检测到侵入事件时就会产生一个中断。

在一个侵入事件被检测到并被清除后，侵入检测引脚TAMPER应该被禁止。然后，在再次写入备份数据寄存器前重新用TPE位启动侵入检测功能。这样，可以阻止软件在侵入检测引脚上仍然有侵入事件时对备份数据寄存器进行写操作。这相当于对侵入引脚TAMPER进行电平检测。

注：当VDD电源断开时，侵入检测功能仍然有效。为了避免不必要的复位数据备份寄存器，TAMPER 引脚应该在片外连接到正确的电平。

5.3.2 RTC校准

为方便测量，RTC时钟可以经64分频输出到侵入检测引脚TAMPER上。通过设置RTC校验寄存器(BKP_RTCCR)的CCO位来开启这一功能。

通过配置CAL[6:0]位，此时钟可以多减慢121ppm。

关于RTC校准和如何提高精度，可以查看SDK的示例

5.4 BKP寄存器

BKP包含42个数据寄存器和3个控制寄存器

地址范围	基地址	外设	总线
0x4000 6C00 - 0x4000 6FFF	0x4000 6C00	后备寄存器(BKP)	APB1

偏移地址	寄存器名称	宽度(bit)	复位值
0x04	BKP_DR1	16	0x0000 0000
0x08	BKP_DR2	16	0x0000 0000
.....	BKP_DRX	16	0x0000 0000
0x28	BKP_DR10	16	0x0000 0000
0x2C	BKP_RTCCR	16	0x0000 0000
0x30	BKP_CR	16	0x0000 0000
0x34	BKP_CSR	16	0x0000 0000
0x40	BKP_DR11	16	0x0000 0000
0x44	BKP_DR12	16	0x0000 0000
.....	BKP_DRX	16	0x0000 0000
0xBC	BKP_DR42	16	0x0000 0000

5.4.1 备份数据寄存器X (BKP_DRX) (X为1到42)

地址偏移：1到10为0x04 到 0x28，11到42为0x40到0xBC

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKP_DRX [15:0]															
rw															

15:0	<p>BKP_DRX [15:0]：备份数据 这些位可以被用来写入用户数据。 注意：BKP_DRx寄存器不会被系统复位、电源复位、从待机模式唤醒所复位。 它们可以由备份域复位来复位或(如果侵入检测引脚TAMPER功能被开启时)由侵入引脚事件复位。</p>
------	---

5.4.2 RTC时钟校准寄存器(BKP_RTCCR)

地址偏移：0x2C

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留				ASOS			ASOE			CCO			CAL[6:0]		
				rw			rw			rw			rw		

15:8	保留，始终读为0。
9	<p>ASOS：闹钟或秒输出选择(Alarm or second output selection) 当设置了ASOE位，ASOS位可用于选择在TAMPER引脚上输出的是RTC秒脉冲还是闹钟脉冲信号。 0：输出RTC闹钟脉冲 1：输出秒脉冲 注：该位只能被后备区的复位所清除</p>
8	<p>ASOE：允许输出闹钟或秒脉冲(Alarm or second output enable) 根据ASOS位的设置，该位允许RTC闹钟或秒脉冲输出到TAMPER引脚上。 输出脉冲的宽度为一个RTC时钟的周期。设置了ASOE位时不能开启TAMPER的功能。 注：该位只能被后备区的复位所清除</p>
7	<p>CCO：校准时钟输出(Calibration clock output) 0：无影响 1：此位置1可以在侵入检测引脚输出经64分频后的RTC时钟。当CCO位置1时，必须关闭侵入检测功能以避免检测到无用的侵入信号。 注：当VDD供电断开时，该位被清除。</p>

6:0	CAL[6:0] : 校准值(Calibration value) 校准值表示在每 2^{20} 个时钟脉冲内将有多少个时钟脉冲被跳过。这可以用来对RTC进行校准, 以 $1000000/2^{20}$ ppm的比例减慢时钟。 RTC时钟可以被减慢0~121ppm。
-----	---

5.4.3 备份控制寄存器(BKP_CR)

偏移地址: 0x30

复位值: 0x0000 0000

地址位：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留					TPAL					TPE					
					rw					rw					

15:2	保留, 始终读为0。
1	TPAL : 侵入检测TAMPER引脚有效电平(TAMPER pin active level) 0: 侵入检测TAMPER引脚上的高电平会清除所有数据备份寄存器(如果TPE位为1) 1: 侵入检测TAMPER引脚上的低电平会清除所有数据备份寄存器(如果TPE位为1)
0	TPE : 启动侵入检测TAMPER引脚(TAMPER pin enable) 0: 侵入检测TAMPER引脚作为通用IO口使用 1: 开启侵入检测引脚作为侵入检测使用

注: 同时设置TPAL和TPE位总是安全的。然而, 同时清除两者会产生一个假的侵入事件。因此, 推荐只在TPE为0时才改变TPAL位的状态。

5.4.4 备份控制/状态寄存器(BKP_CSR)

偏移地址: 0x34

复位值: 0x0000 0000

地址位: 0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留		TIF		TEF		保留		TPIE		CTI		CTE			
		rw		rw		rw		rw		rw		rw			

15:10	保留, 始终读为0。
9	TIF : 侵入中断标志(Tamper interrupt flag) 当检测到有侵入事件且TPIE位为1时, 此位由硬件置1。通过向CTI位写1来清除此标志位(同时也清除了中断)。如果TPIE位被清除, 则此位也会被清除。 0: 无侵入中断 1: 产生侵入中断 注意: 仅当系统复位或由待机模式唤醒后才复位该位

8	TEF : 侵入事件标志(Tamper event flag) 当检测到侵入事件时此位由硬件置1。通过向CTE位写1可清除此标志位 0: 无侵入事件 1: 检测到侵入事件 注: 侵入事件会复位所有的BKP_DRx寄存器。只要TEF为1, 所有的BKP_DRx寄存器就一直保持复位状态。当此位被置1时, 若对BKP_DRx进行写操作, 写入的值不会被保存。
7:3	保留, 始终读为0。
2	TPIE : 允许侵入TAMPER引脚中断(TAMPER pin interrupt enable) 0: 禁止侵入检测中断 1: 允许侵入检测中断(BKP_CR寄存器的TPE位也必须被置1) 注1: 侵入中断无法将系统内核从低功耗模式唤醒。 注2: 仅当系统复位或由待机模式唤醒后才复位该位。
1	CTI : 清除侵入检测中断(Clear tamper interrupt) 此位只能写入, 读出值为0。 0: 无效 1: 清除侵入检测中断和TIF侵入检测中断标志
0	CTE : 清除侵入检测事件(Clear tamper event) 此位只能写入, 读出值为0。 0: 无效 1: 清除TEF侵入检测事件标志(并复位侵入检测器)。

6 复位和时钟控制 (RCC)

6.1 复位

AIR32F103有三种复位形式, 分别为系统复位、上电复位和备份区域复位。

6.1.1 系统复位

除了时钟控制器的RCC_CSR寄存器中的复位标志位和备份区域中的寄存器(见图4)以外, 系统复位将复位所有寄存器至它们的复位状态。

当发生以下任一事件时, 产生一个系统复位:

1. NRST引脚上的低电平(外部复位)
2. 窗口看门狗计数终止(WWDG复位)
3. 独立看门狗计数终止(IWDG复位)
4. 软件复位(SW复位)
5. 低功耗管理复位可通过查看RCC_CSR控制状态寄存器中的复位状态标志位识别复位事件来源。

软件复位

通过将Cortex™-M3中断应用和复位控制寄存器中的SYSRESETREQ位置'1', 可实现软件复位。请参考Cortex™-M3技术参考手册获得进一步信息。

低功耗管理复位

在以下两种情况下可产生低功耗管理复位：

- 1.在进入待机模式时产生低功耗管理复位：通过将用户选择字节中的nRST_STDBY位置'1'将使能该复位。这时，即使执行了进入待机模式的过程，系统将被复位而不是进入待机模式。
- 2.在进入停止模式时产生低功耗管理复位：通过将用户选择字节中的nRST_STOP位置'1'将使能该复位。这时，即使执行了进入停机模式的过程，系统将被复位而不是进入停机模式。

如何配置用户选项字节，请参考相关手册

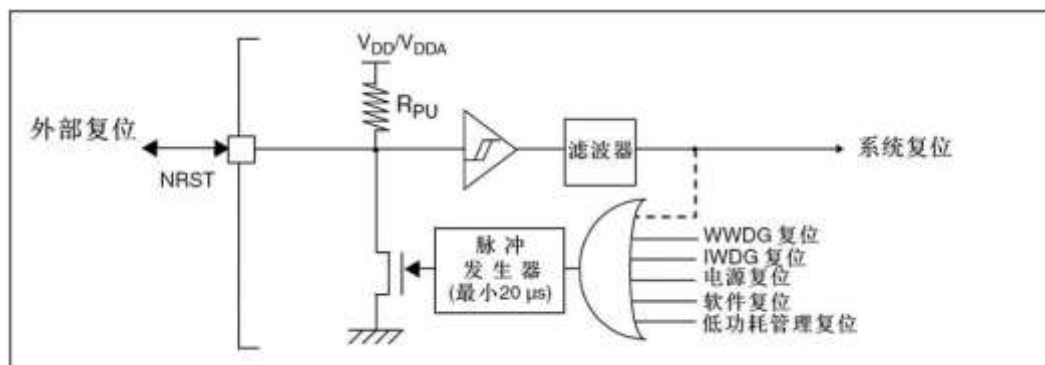
6.1.2 电源复位

当以下事件之一发生时，产生电源复位：

- 1.上电/掉电复位(POR/PDR复位)
- 2.从待机模式中返回

电源复位将复位除了备份区域外的所有寄存器。(见图4)

图中复位源将 终作用于RESET引脚，并在复位过程中保持低电平。复位入口矢量被固定在地址0x0000_0004。更多细节，参阅表55：其它STM32F10xxx产品(小容量、中容量和大容量)的向量表。芯片内部的复位信号会在NRST引脚上输出，脉冲发生器保证每一个(外部或内部)复位源都能有至少20μs的脉冲延时；当NRST引脚被拉低产生外部复位时，它将产生复位脉冲。



图表 6

6.1.3 备份域复位

备份区域拥有两个专门的复位，它们只影响备份区域(见图4)。

当以下事件之一发生时，产生备份区域复位。

- 1.软件复位，备份区域复位可由设置备份域控制寄存器（RCC_BDCR)(见6.3.9节)中的BDRST位产生。
- 2.在VDD和VBAT两者掉电的前提下，VDD或VBAT上电将引发备份区域复位。

6.2 时钟

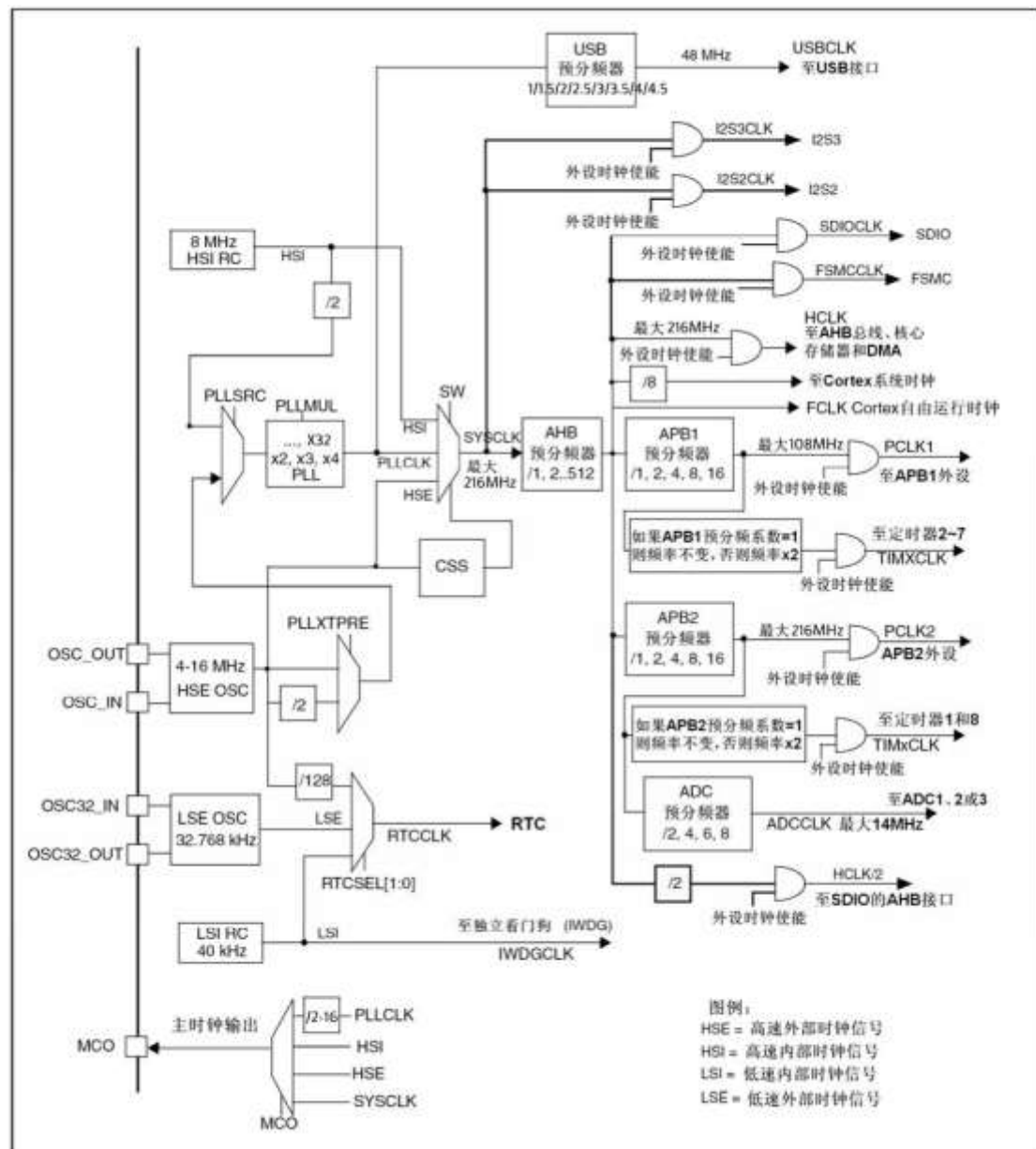
三种不同的时钟源可被用来驱动系统时钟(SYSCLK)：

- HSI振荡器时钟

- HSE振荡器时钟
- PLL时钟

这些设备有以下2种二级时钟源：

- 40kHz低速内部RC，可以用于驱动独立看门狗和通过程序选择驱动RTC。RTC用于从停机/待机模式下自动唤醒系统。
 - 32.768kHz低速外部晶体也可用来通过程序选择驱动RTC(RTCCLK)。
- 当不被使用时，任一个时钟源都可被独立地启动或关闭，由此优化系统功耗。



图表 7

1. 当HSI被用于作为PLL时钟的输入时，系统时钟能得到的大频率是128MHz，使用HSE时最大可达到256MHz，但是不建议长时间保持256MHz运行。
2. 对于内部和外部时钟源的特性，请参考相应产品数据手册中“电气特性”章节。

用户可通过多个预分频器配置AHB、高速APB(APB2)和低速APB(APB1)域的频率。AHB和 APB2域的大频率是216MHz。APB1域的大允许频率是108MHz。SDIO接口的时钟频率固定为HCLK/2。

RCC通过AHB时钟(HCLK)8分频后作为Cortex系统定时器(SysTick)的外部时钟。通过对SysTick控制与状态寄存器的设置，可选择上述时钟或Cortex(HCLK)时钟作为SysTick时钟。ADC时钟由高速APB2时钟经2、4、6或8分频后获得。

定时器时钟频率分配由硬件按以下2种情况自动设置：

1. 如果相应的APB预分频系数是1，定时器的时钟频率与所在APB总线频率一致。
2. 否则，定时器的时钟频率被设为与其相连的APB总线频率的2倍。

FCLK是Cortex™-M3的自由运行时钟。详情见ARM的Cortex™-M3技术参考手册。

6.2.1 HSE时钟

高速外部时钟信号(HSE)由以下两种时钟源产生：

- HSE外部晶体/陶瓷谐振器
- HSE用户外部时钟

为了减少时钟输出的失真和缩短启动稳定时间，晶体/陶瓷谐振器和负载电容器必须尽可能地靠近振荡器引脚。负载电容值必须根据所选择的振荡器来调整。

	硬件配置
外部时钟	
晶体/陶瓷谐振器	

表格 11

外部时钟源(有源时钟，也叫旁路时钟)

在这个模式里，必须提供外部时钟。它的频率最高可达25MHz。用户可通过设置在时钟控制寄存器中的HSEBYP和HSEON位来选择这一模式。外部时钟信号(50%占空比的方波、正弦波或三角波)必须连到SOC_IN引脚，同时保证OSC_OUT引脚悬空。见图9。

外部晶体/陶瓷谐振器(HSE晶体，无源时钟)

4~16Mz外部振荡器可为系统提供更为精确的主时钟。相关的硬件配置可参考图9，进一步信息可参考数据手册的电气特性部分。

在时钟控制寄存器RCC_CR中的HSERDY位用来指示高速外部振荡器是否稳定。在启动时，直

到这一位被硬件置'1'，时钟才被释放出来。如果在时钟中断寄存器RCC_CIR中允许产生中断，将会产生相应中断。

HSE晶体可以通过设置时钟控制寄存器里RCC_CR中的HSEON位被启动和关闭。

6.2.2 HSI时钟

HSI时钟信号由内部8MHz的RC振荡器产生，可直接作为系统时钟或在2分频后作为PLL输入。HSI RC振荡器能够在不需要任何外部器件的条件下提供系统时钟。它的启动时间比HSE晶体振荡器短。然而，即使在校准之后它的时钟频率精度仍较差。

校准

制造工艺决定了不同芯片的RC振荡器频率会不同，这就是为什么每个芯片的HSI时钟频率在出厂前已经被校准到1%(25°C)的原因。系统复位时，工厂校准值被装载到时钟控制寄存器的HSICAL[7:0]位。

如果用户的应用基于不同的电压或环境温度，这将会影响RC振荡器的精度。可以通过时钟控制寄存器里的HSITRIM[4:0]位来调整HSI频率。

时钟控制寄存器中的HSIRDY位用来指示HSI RC振荡器是否稳定。在时钟启动过程中，直到这一位被硬件置'1'，HSI RC输出时钟才被释放。HSI RC可由时钟控制寄存器中的HSION位来启动和关闭。

如果HSE晶体振荡器失效，HSI时钟会被作为备用时钟源。参考6.2.7节时钟安全系统。

6.2.3 PLL

内部PLL可以用来倍频HSI RC的输出时钟或HSE晶体输出时钟。参考图8和时钟控制寄存器。

PLL的设置(选择HSI振荡器除2或HSE振荡器为PLL的输入时钟，和选择倍频因子)必须在其被激活前完成。一旦PLL被激活，这些参数就不能被改动。

如果PLL中断在时钟中断寄存器里被允许，当PLL准备就绪时，可产生中断申请。

如果需要在应用中使用USB接口，PLL必须被设置为输出48或72MHz时钟，用于提供48MHz的USBCLK时钟。

6.2.4 LSE时钟

LSE晶体是一个32.768kHz的低速外部晶体或陶瓷谐振器。它为实时时钟或者其他定时功能提供一个低功耗且精确的时钟源。

LSE晶体通过在备份域控制寄存器(RCC_BDCR)里的LSEON位启动和关闭。

在备份域控制寄存器(RCC_BDCR)里的LSERDY指示LSE晶体振荡是否稳定。在启动阶段，直到这个位被硬件置'1'后，LSE时钟信号才被释放出来。如果在时钟中断寄存器里被允许，可产生中断申请。

外部时钟源(LSE旁路)

在这个模式里必须提供一个32.768kHz频率的外部时钟源。你可以通过设置在备份域控制寄存器(RCC_BDCR)里的LSEBYP和LSEON位来选择这个模式。具有50%占空比的外部时钟信号(方波、正弦波或三角波)必须连到OSC32_IN引脚，同时保证OSC32_OUT引脚悬空，见图9。

6.2.5 LSI时钟

LSI RC担当一个低功耗时钟源的角色，它可以在停机和待机模式下保持运行，为独立看门狗和自动唤醒单元提供时钟。LSI时钟频率大约40kHz(在30kHz和60kHz之间)。进一步信息请参考数据手册中有关电气特性部分。

LSI RC可以通过控制/状态寄存器(RCC_CSR)里的LSION位来启动或关闭。

在控制/状态寄存器(RCC_CSR)里的LSIRDY位指示低速内部振荡器是否稳定。在启动阶段，直到这个位被硬件设置为'1'后，此时钟才被释放。如果在时钟中断寄存器(RCC_CIR)里被允许，将产生LSI中断申请。

AIR32F103全系产品都可以进行LSI校准

LSI校准

可以通过校准内部低速振荡器LSI来补偿其频率偏移，从而获得精度可接受的RTC时间基数，以及独立看门狗(IWDG)的超时时间(当这些外设以LSI为时钟源)。

校准可以通过使用TIM5的输入时钟(TIM5_CLK)测量LSI时钟频率实现。测量以HSE的精度为保证，软件可以通过调整RTC的20位预分频器来获得精确的RTC时钟基数，以及通过计算得到精确的独立看门狗(IWDG)的超时时间。

LSI校准步骤如下：

1. 打开TIM5，设置通道4为输入捕获模式；
2. 设置AFIO_MAPR的TIM5_CH4_IEMAP位为'1'，在内部把LSI连接到TIM5的通道4；
3. 通过TIM5的捕获/比较4事件或者中断来测量LSI时钟频率；
4. 根据测量结果和期望的RTC时间基数和独立看门狗的超时时间，设置20位预分频器。

注:相关校准示例代码在SDK的RTC文件夹内可以查看

6.2.6 系统时钟(SYSCLK)选择

系统复位后，HSI振荡器被选为系统时钟。当时钟源被直接或通过PLL间接作为系统时钟时，它将不能被停止。

只有当目标时钟源准备就绪了(经过启动稳定阶段的延迟或PLL稳定)，从一个时钟源到另一个时钟源的切换才会发生。在被选择时钟源没有就绪时，系统时钟的切换不会发生。直至目标时钟源就绪，才发生切换。

在时钟控制寄存器(RCC_CR)里的状态位指示哪个时钟已经准备好了，哪个时钟目前被用作系统时钟。

6.2.7 时钟安全系统(CSS)

时钟安全系统可以通过软件被激活。一旦其被激活，时钟监测器将在HSE振荡器启动延迟后被使能，并在HSE时钟关闭后关闭。

如果HSE时钟发生故障，HSE振荡器被自动关闭，时钟失效事件将被送到高级定时器(TIM1和TIM8)的刹车输入端，并产生时钟安全中断CSSI，允许软件完成营救操作。此CSSI中断连接到Cortex™-M3的NMI中断(不可屏蔽中断)。

注意：一旦CSS被激活，并且HSE时钟出现故障，CSS中断就产生，并且NMI也自动产生。NMI将被不断执行，直到CSS中断挂起位被清除。因此，在NMI的处理程序中必须通过设置时钟中断寄存器(RCC_CIR)里的CSSC位来清除CSS中断。

如果HSE振荡器被直接或间接地作为系统时钟，(间接的意思是：它被作为PLL输入时钟，并且PLL时钟被作为系统时钟)，时钟故障将导致系统时钟自动切换到HSI振荡器，同时外部HSE振荡器被关闭。在时钟失效时，如果HSE振荡器时钟(被分频或未被分频)是用作系统时钟的PLL的输入时钟，PLL也将被关闭。

6.2.8 RTC时钟

通过设置备份域控制寄存器(RCC_BDCR)里的RTCSEL[1:0]位，RTCCLK时钟源可以由HSE/128、LSE或LSI时钟提供。除非备份域复位，此选择不能被改变。

LSE时钟在备份域里，但HSE和LSI时钟不是。因此：

- 如果LSE被选为RTC时钟：
 - 只要VBAT维持供电，尽管VDD供电被切断，RTC仍继续工作。
- 如果LSI被选为自动唤醒单元(AWU)时钟：
 - 如果VDD供电被切断，AWU状态不能被保证。有关LSI校准，详见6.2.5节LSI时钟。
- 如果HSE时钟128分频后作为RTC时钟：
 - 如果VDD供电被切断或内部电压调压器被关闭(1.8V域的供电被切断)，则RTC状态不确定。
 - 必须设置电源控制寄存器(见4.4.1节)的DPB位(取消后备区域的写保护)为'1'。

6.2.9 看门狗时钟

如果独立看门狗已经由硬件选项或软件启动，LSI振荡器将被强制在打开状态，并且不能被关闭。在LSI振荡器稳定后，时钟供应给IWDG。

6.2.10 时钟输出

微控制器允许输出时钟信号到外部MCO引脚。

相应的GPIO端口寄存器必须被配置为相应功能。以下四个时钟信号可被选作MCO时钟：

- SYSCLK
- HSI
- HSE
- 除2的PLL时钟时钟的选择由时钟配置寄存器(RCC_CFGR)中的MCO[2:0]位控制。

6.3 RCC寄存器

地址范围	基地址	外设	总线
0x4002 1000 - 0x4002 13FF	0x4002 1000	复位和时钟控制(RCC)	AHB

偏移地址	寄存器名称	宽度(bit)	复位值
0x00	RCC_CR	32	0x0000 XX83
0x04	RCC_CFGR	32	0x0000 0000
0x08	RCC_CIR	32	0x0000 0000
0x0C	RCC_APB2RSTR	32	0x0000 0000
0x10	RCC_APB1RSTR	32	0x0000 0000
0x14	RCC_AHBENR	32	0x0000 0014
0x18	RCC_APB2ENR	32	0x0000 0000
0x1C	RCC_APB1ENR	32	0x0000 0000
0x20	RCC_BDCR	32	0x0000 0000
0x24	RCC_CSR	32	0x0000 0000
0x30	RCC_MCO_VAL	32	0x0000 0001

6.3.1 时钟控制寄存器(RCC_CR)

偏移地址: 0x00

复位值: 0x0000 XX83, X代表未定义

访问: 无等待状态, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PLLRDY		PLLON		保留		CSSON		HSEBYP		HSERDY		HSEON	
		r		rw				rw		rw		r		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]			HSITRIM[4:0]			保留			HSIRDY			HSION			
r			rw						r			rw			

31:26	保留, 始终读为0。
25	PLLRDY : PLL时钟就绪标志 (PLL clock ready flag) PLL锁定后由硬件置'1'。 0: PLL未锁定; 1: PLL锁定。
24	PLLON : PLL使能 (PLL enable) 由软件置'1'或清零。 当进入待机和停止模式时, 该位由硬件清零。当PLL时钟被用作或被选择将要作为系统时钟时, 该位不能被清零。 0: PLL关闭; 1: PLL使能。
23:20	保留, 始终读为0。

19	<p>CSSON: 时钟安全系统使能 (Clock security system enable) 由软件置'1'或清零以使能时钟监测器。</p> <p>0: 时钟监测器关闭;</p> <p>1: 如果外部4-16MHz振荡器就绪, 时钟监测器开启。</p>
18	<p>HSEBYP: 外部高速时钟旁路 (External high-speed clock bypass)</p> <p>在调试模式下由软件置'1'或清零来旁路外部晶体振荡器。只有在外部4-16MHz振荡器关闭的情况下, 才能写入该位。</p> <p>0: 外部4-16MHz振荡器没有旁路;</p> <p>1: 外部4-16MHz外部晶体振荡器被旁路。</p>
17	<p>HSERDY: 外部高速时钟就绪标志 (External high-speed clock ready flag)</p> <p>由硬件置'1'来指示外部4-16MHz振荡器已经稳定。在HSEON位清零后, 该位需要6个外部425MHz振荡器周期清零。 0: 外部4-16MHz振荡器没有就绪; 1: 外部4-16MHz振荡器就绪。</p>
16	<p>HSEON: 外部高速时钟使能 (External high-speed clock enable) 由软件置'1'或清零。</p> <p>当进入待机和停止模式时, 该位由硬件清零, 关闭4-16MHz外部振荡器。当外部4-16MHz振荡器被用作或被选择将要作为系统时钟时, 该位不能被清零。</p> <p>0: HSE振荡器关闭;</p> <p>1: HSE振荡器开启。</p>
15:8	<p>HSICAL[7:0]: 内部高速时钟校准 (Internal high-speed clock calibration)</p> <p>在系统启动时, 这些位被自动初始化</p>
7:3	<p>HSITRIM[4:0]: 内部高速时钟调整 (Internal high-speed clock trimming) 由软件写入来调整内部高速时钟, 它们被叠加在HSICAL[5:0]数值上。</p> <p>这些位在HSICAL[7:0]的基础上, 让用户可以输入一个调整数值, 根据电压和温度的变化调整内部HSI RC振荡器的频率。默认数值为16, 可以把HSI调整到$8\text{MHz} \pm 1\%$; 每步HSICAL的变化调整约40kHz。</p>
2	保留, 始终读为0。
1	<p>HSIRDY: 内部高速时钟就绪标志 (Internal high-speed clock ready flag)</p> <p>由硬件置'1'来指示内部8MHz振荡器已经稳定。在HSION位清零后, 该位需要6个内部8MHz振荡器周期清零。</p> <p>0: 内部8MHz振荡器没有就绪;</p> <p>1: 内部8MHz振荡器就绪。</p>
0	<p>HSION: 内部高速时钟使能 (Internal high-speed clock enable) 由软件置'1'或清零。</p> <p>当从待机和停止模式返回或用作系统时钟的外部4-16MHz振荡器发生故障时, 该位由硬件置'1'来启动内部8MHz的RC振荡器。当内部8MHz振荡器被直接或间接地用作或被选择将要作为系统时钟时, 该位不能被清零。</p> <p>0: 内部8MHz振荡器关闭;</p> <p>1: 内部8MHz振荡器开启。</p>

6.3.2 时钟配置寄存器(RCC_CFGR)

偏移地址: 0x04
复位值: 0x0000 0000
访问: 0到2个等待周期，字，半字和字节访问只有当访问发生在时钟切换时，才会插入1或2个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		PLLX		MCO[2:0]		保留		USBPRE		PLLMUL[3:0]		PLLXTPRE		PLLSRC	
		rw		rw		rw		rw		rw		rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[1:0]			PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]			SWS[1:0]		SW[1:0]	
rw			rw			rw			rw			r		rw	

31	USBPREX: USB预分频拓展字，具体使用见USBPRE
30: 29	保留，始终读为0。
28	PLLMULX:PLL倍频拓展字，具体使用见PLLMUL
27:24	MCO ：微控制器时钟输出（Microcontroller clock output）由软件置'1'或清零。 000xx：没有时钟输出； 0100：系统时钟(SYSCLK)输出； 0101：内部RC振荡器时钟(HSI)输出； 0110：外部振荡器时钟(HSE)输出； 0111：PLL时钟2分频后输出。 注意：- 该时钟输出在启动和切换MCO时钟源时可能会被截断。 - 在系统时钟作为输出至MCO引脚时，请保证输出时钟频率不超过50MHz (I/O口最高频率)。
23:22	USBPRE ：USB预分频（USB prescaler） 由软件置'1'或清'0'来产生48MHz的USB时钟。在RCC_APB1ENR寄存器中使能USB时钟之前，必须保证该位已经有效。如果USB时钟被使能，该位不能被清零。其中最高位为拓展字 USBPREX 000：PLL时钟1.5倍分频作为USB时钟 001：PLL时钟直接作为USB时钟 010：PLL时钟2.5倍分频作为USB时钟 011：PLL时钟2倍分频作为USB时钟 100：PLL时钟3.5倍分频作为USB时钟 101：PLL时钟3倍分频作为USB时钟 110：PLL时钟4.5倍分频作为USB时钟 111：PLL时钟4倍分频作为USB时钟

21:18	<p>PLLMUL: PLL倍频系数 (PLL multiplication factor)</p> <p>由软件设置来确定PLL倍频系数。只有在PLL关闭的情况下才可被写入。其中最高位为拓展字PLLMULX</p> <p>00000: PLL 2倍输出</p> <p>00001: PLL 3倍输出</p> <p>..... 递增一位, PLL 输出增加1</p> <p>01111: PLL 16倍输出</p> <p>10000: PLL 17倍输出</p> <p>.....递增一位, PLL 输出增加1</p> <p>11111: PLL 32倍输出</p> <p>注意: PLL的输出频率不能超过256MHz</p>
17	<p>PLLXTPRE: HSE分频器作为PLL输入 (HSE divider for PLL entry)</p> <p>由软件置'1'或清'0'来分频HSE后作为PLL输入时钟。只能在关闭PLL时才能写入此位。</p> <p>0: HSE不分频</p> <p>1: HSE 2分频</p>
16	<p>PLLSRC: PLL输入时钟源 (PLL entry clock source)</p> <p>由软件置'1'或清'0'来选择PLL输入时钟源。只能在关闭PLL时才能写入此位。</p> <p>0: HSI振荡器时钟经2分频后作为PLL输入时钟 1: HSE时钟作为PLL输入时钟。</p>
15:14	<p>ADCPRE[1:0]: ADC预分频 (ADC prescaler) 由软件置'1'或清'0'来确定ADC时钟频率</p> <p>00: PCLK2 2分频后作为ADC时钟</p> <p>01: PCLK2 4分频后作为ADC时钟</p> <p>10: PCLK2 6分频后作为ADC时钟</p> <p>11: PCLK2 8分频后作为ADC时钟</p>
13:11	<p>PPRE2[2:0]: 高速APB预分频(APB2) (APB high-speed prescaler (APB2)) 由软件置'1'或清'0'来控制高速APB2时钟(PCLK2)的预分频系数。</p> <p>0xx: HCLK不分频</p> <p>100: HCLK 2分频</p> <p>101: HCLK 4分频</p> <p>110: HCLK 8分频</p> <p>111: HCLK 16分频</p>
10:8	<p>PPRE1[2:0]: 低速APB预分频(APB1) (APB low-speed prescaler (APB1)) 由软件置'1'或清'0'来控制低速APB1时钟(PCLK1)的预分频系数。警告: 软件必须保证APB1时钟频率不超过36MHz。</p> <p>0xx: HCLK不分频</p> <p>100: HCLK 2分频</p> <p>101: HCLK 4分频</p> <p>110: HCLK 8分频</p> <p>111: HCLK 16分频</p>

7:4	<p>HPRE[3:0]: AHB预分频 (AHB Prescaler) 由软件置'1'或清'0'来控制 AHB时钟的预分频系数 0xxx: SYSCLK不分频 1000: SYSCLK 2分频 1001: SYSCLK 4分频 1010: SYSCLK 8分频 1011: SYSCLK 16分频 1100: SYSCLK 64分频 1101: SYSCLK 128分频 1110: SYSCLK 256分频 1111: SYSCLK 512分频 注意: 当AHB时钟的预分频系数大于1时, 必须开启预取缓冲器。详见闪存读取(第2.3.3节)。</p>
3:2	<p>SWS[1:0]: 系统时钟切换状态 (System clock switch status) 由硬件置'1'或清'0'来指示哪一个时钟源被作为系统时钟。 00: HSI作为系统时钟; 01: HSE作为系统时钟; 10: PLL输出作为系统时钟; 11: 不可用。</p>
1:0	<p>SW[1:0]: 系统时钟切换 (System clock switch) 由软件置'1'或清'0'来选择系统时钟源。 在从停止或待机模式中返回时或直接或间接作为系统时钟的HSE出现故障时, 由硬件强制选择 HSI作为系统时钟(如果时钟安全系统已经启动) 00: HSI作为系统时钟; 01: HSE作为系统时钟; 10: PLL输出作为系统时钟; 11: 不可用。</p>

6.3.3 时钟中断寄存器 (RCC_CIR)

偏移地址: 0x08

复位值: 0x0000 0000

访问:无等待周期, 字, 半字 和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留		CSSC		保留		PLLRDYC		HSERDYC		HISRDYC		LSERDYC		LSIRDYC	
		w				w		w		w		w		w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF		保留	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF		
	rw	rw	rw	rw	rw	r			r	r	r	r	r		

32:24	保留，始终读为0。
23	CSSC：清除时钟安全系统中断（Clock security system interrupt clear）由软件置'1'来清除CSSF安全系统中断标志位CSSF。 0：无作用； 1：清除CSSF安全系统中断标志位。
22:21	保留，始终读为0。
20	PLLRDYC：清除PLL就绪中断（PLL ready interrupt clear）由软件置'1'来清除PLL就绪中断标志位PLLRDYF。 0：无作用； 1：清除PLL就绪中断标志位PLLRDYF。
19	HSEIDYC：清除HSE就绪中断（HSE ready interrupt clear）由软件置'1'来清除HSE就绪中断标志位HSEIDYF。 0：无作用； 1：清除HSE就绪中断标志位HSEIDYF。
18	HSIRDYC：清除HSI就绪中断（HSI ready interrupt clear）由软件置'1'来清除HSI就绪中断标志位HSIRDYF。 0：无作用； 1：清除HSI就绪中断标志位HSIRDYF。
17	LSERDYC：清除LSE就绪中断（LSE ready interrupt clear）由软件置'1'来清除LSE就绪中断标志位LSERDYF。 0：无作用； 1：清除LSE就绪中断标志位LSERDYF。
16	LSIRDYC：清除LSI就绪中断（LSI ready interrupt clear）由软件置'1'来清除LSI就绪中断标志位LSIRDYF。 0：无作用； 1：清除LSI就绪中断标志位LSIRDYF。
15:13	保留，始终读为0。
12	PLLRDYIE：PLL就绪中断使能（PLL ready interrupt enable）由软件置'1'或清'0'来使能或关闭PLL就绪中断。 0：PLL就绪中断关闭； 1：PLL就绪中断使能。
11	HSEIDYIE：HSE就绪中断使能（HSE ready interrupt enable）由软件置'1'或清'0'来使能或关闭外部4-16MHz振荡器就绪中断。 0：HSE就绪中断关闭； 1：HSE就绪中断使能。
10	HSIRDYIE：HSI就绪中断使能（HSI ready interrupt enable）由软件置'1'或清'0'来使能或关闭内部8MHz RC振荡器就绪中断。 0：HSI就绪中断关闭；

	1: HSI就绪中断使能。
9	LSERDYIE : LSE就绪中断使能 (LSE ready interrupt enable) 由软件置'1'或清'0'来使能或关闭外部32kHz RC振荡器就绪中断。 0: LSE就绪中断关闭; 1: LSE就绪中断使能。
8	LSIRDYIE : LSI就绪中断使能 (LSI ready interrupt enable) 由软件置'1'或清'0'来使能或关闭内部40kHz RC振荡器就绪中断。 0: LSI就绪中断关闭; 1: LSI就绪中断使能。
7	CSSF : 时钟安全系统中断标志 (Clock security system interrupt flag) 在外部4-16MHz振荡器时钟出现故障时, 由硬件置'1'。 由软件通过置'1' CSSC位来清除。 0: 无HSE时钟失效产生的安全系统中断; 1: HSE时钟失效导致了时钟安全系统中断。
6:5	保留, 始终读为0。
4	PLLRDYF : PLL就绪中断标志 (PLL ready interrupt flag) 在PLL就绪且PLLRDYIE位被置'1'时, 由硬件置'1'。 由软件通过置'1' PLLRDYC位来清除。 0: 无PLL上锁产生的时钟就绪中断; 1: PLL上锁导致时钟就绪中断。
3	HSERDYF : HSE就绪中断标志 (HSE ready interrupt flag) 在外部低速时钟就绪且HSERDYIE位被置'1'时, 由硬件置'1'。 由软件通过置'1' HSERDYC位来清除。 0: 无外部4-16MHz振荡器产生的时钟就绪中断; 1: 外部4-16MHz振荡器导致时钟就绪中断。
2	HSIRDYF : HSI就绪中断标志 (HSI ready interrupt flag) 在内部高速时钟就绪且HSIRDYIE位被置'1'时, 由硬件置'1'。 由软件通过置'1' HSIRDYC位来清除。 0: 无内部8MHz RC振荡器产生的时钟就绪中断; 1: 内部8MHz RC振荡器导致时钟就绪中断。
1	LSERDYF : LSE就绪中断标志 (LSE ready interrupt flag) 在外部低速时钟就绪且LSERDYIE位被置'1'时, 由硬件置'1'。 由软件通过置'1' LSERDYC位来清除。 0: 无外部32kHz振荡器产生的时钟就绪中断; 1: 外部32kHz振荡器导致时钟就绪中断。

0	LSIRDYF : LSI就绪中断标志 (LSI ready interrupt flag) 在内部低速时钟就绪且LSIRDYIE位被置'1'时, 由硬件置'1'。 由软件通过置'1' LSIRDYC位来清除。 0: 无内部40kHz RC振荡器产生的时钟就绪中断; 1: 内部40kHz RC振荡器导致时钟就绪中断。
---	--

6.3.4 APB2 外设复位寄存器 (RCC_APB2RSTR)

偏移地址: 0x0C
复位值: 0x0000 0000
访问: 无等待周期, 字, 半字 和字节访问

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
ADC 3 RST	USA RT1 RST	TIM8 RST	SPI1 RST	TIM1 RST	ADC 2 RST	ADC 1 RST	IOPG RST	IOPF RST	IOPE RST	IOPD RST	IOPC RST	IOPB RST	IOPA RST	保留	AFIO RST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

31:16	保留
15	ADC3RST : ADC3接口复位 (ADC3 interface reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位ADC3接口。
14	USART1RST : USART1复位 (USART1 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位USART1。
13	TIM8RST : TIM8定时器复位 (TIM8 timer reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM8定时器。
12	SPI1RST : SPI1复位 (SPI 1 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位SPI1。
12	TIM1RST : TIM1定时器复位 (TIM1 timer reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM1定时器。
10	ADC2RST : ADC2接口复位 (ADC 2 interface reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位ADC2接口。

9	ADC1RST : ADC1接口复位 (ADC 1 interface reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位ADC1接口。
8	IOPGRST : IO端口G复位 (IO port G reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口G。
7	IOPFRST : IO端口F复位 (IO port F reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口F。
6	IOPERST : IO端口E复位 (IO port E reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口E。
5	IOPDRST : IO端口D复位 (IO port D reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口D。
4	IOPCRST : IO端口C复位 (IO port C reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口C。
3	IOPBRST : IO端口B复位 (IO port B reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口B。
2	IOPARST : IO端口A复位 (IO port A reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位IO端口A。
1	保留, 始终读为0。
0	AFIORST : 辅助功能IO复位 (Alternate function I/O reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位辅助功能。

6.3.5 APB1 外设复位寄存器 (RCC_APB1RSTR)

偏移地址: 0x10

复位值: 0x0000 0000

访问: 无等待周期, 字, 半字和字节访问

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DAC RST	PW R RST	BKP RST	保留	CAN RST	保留	USB RST	I2C2 RST	I2C1 RST	UAR T5R ST	UAR T4R ST	USA RT3 RST	USA RT2 RST	保留	
	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						
SPI3 RST		SPI2 RST		保留		WWDG RST		保留		TIM7 RST		TIM6 RST		TIM5 RST		TIM4 RST		TIM3 RST		TIM2 RST	
rw		rw				rw				rw		rw		rw		rw		rw		rw	

31:30	保留，始终读为0。
29	DACRST ：DAC接口复位 (DAC interface reset) 由软件置'1'或清'0' 0：无作用； 1：复位DAC接口。
28	PWRRST ：电源接口复位 (Power interface reset) 由软件置'1'或清'0' 0：无作用； 1：复位电源接口。
27	BKPRST ：备份接口复位 (Backup interface reset) 由软件置'1'或清'0' 0：无作用； 1：复位备份接口。
26	保留，始终读为0。
25	CANRST ：CAN复位 (CAN reset) 由软件置'1'或清'0' 0：无作用； 1：复位CAN。
24	保留，始终读为0。
23	USBRST ：USB复位 (USB reset) 由软件置'1'或清'0' 0：无作用； 1：复位USB。
22	I2C2RST ：I2C 2复位 (I2C 2 reset) 由软件置'1'或清'0' 0：无作用； 1：复位I2C 2。
21	I2C1RST ：I2C 1复位 (I2C 1 reset) 由软件置'1'或清'0' 0：无作用； 1：复位I2C 1。
20	UART5RST ：UART5复位 (UART 5 reset) 由软件置'1'或清'0' 0：无作用； 1：复位UART5。
19	UART4RST ：UART4复位 (UART 4 reset) 由软件置'1'或清'0' 0：无作用； 1：复位UART4。
18	USART3RST ：USART3复位 (USART 3 reset) 由软件置'1'或清'0' 0：无作用； 1：复位USART3。
17	USART2RST ：USART2复位 (USART 2 reset) 由软件置'1'或清'0' 0：无作用； 1：复位USART2。
16	保留，始终读为0。

15	SPI3RST SPI3 复位 (SPI 3 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位SPI3。
14	SPI2RST : SPI2复位 (SPI 2 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位SPI2。
13:12	保留, 始终读为0。
11	WWDGRST : 窗口看门狗复位 (Window watchdog reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位窗口看门狗。
10:6	保留, 始终读为0。
5	TIM7RST : 定时器7复位 (Timer 7 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM7定时器。
4	TIM6RST : 定时器6复位 (Timer 6 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM6定时器。
3	TIM5RST : 定时器5复位 (Timer 5 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM5定时器。
2	TIM4RST : 定时器4复位 (Timer 4 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM4定时器。
1	TIM3RST : 定时器3复位 (Timer 3 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM3定时器。
0	TIM2RST : 定时器2复位 (Timer 2 reset) 由软件置'1'或清'0' 0: 无作用; 1: 复位TIM2定时器。

6.3.6 AHB外设时钟使能寄存器 (RCC_AHBENR)

偏移地址: 0x14

复位值: 0x0000 0014

访问: 无等待周期, 字, 半字 和字节访问

注: 当外设时钟没有启用时, 软件不能读出外设寄存器的数值, 返回的数值始终是0x0。

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	CRCEEN	保留	FLITFEN	保留	SRAMEN	DMA2EN	DMA1EN				
	rw		rw		rw		rw		rw		rw	rw	rw		

31:11	保留，始终读为0。
10	SDIOEN ：SDIO时钟使能（SDIO clock enable）由软件置‘1’或清‘0’。 0：SDIO时钟关闭； 1：SDIO时钟开启。
9	保留，始终读为0。
8	FSMCEN ：FSMC时钟使能（FSMC clock enable）由软件置‘1’或清‘0’。 0：FSMC时钟关闭； 1：FSMC时钟开启。
7	保留，始终读为0。
6	CRCEN ：CRC时钟使能（CRC clock enable）由软件置‘1’或清‘0’。 0：CRC时钟关闭； 1：CRC时钟开启。
5	保留，始终读为0。
4	FLITFEN ：闪存接口电路时钟使能（FLITF clock enable） 由软件置‘1’或清‘0’来开启或关闭睡眠模式时闪存接口电路时钟。 0：睡眠模式时闪存接口电路时钟关闭； 1：睡眠模式时闪存接口电路时钟开启。
3	保留，始终读为0。
2	SRAMEN ：SRAM时钟使能（SRAM interface clock enable）由软件置‘1’或清‘0’ 来开启或关闭睡眠模式时SRAM时钟。 0：睡眠模式时SRAM时钟关闭； 1：睡眠模式时SRAM时钟开启。
1	DMA2EN ：DMA2时钟使能（DMA2 clock enable）由软件置‘1’或清‘0’。 0：DMA2时钟关闭； 1：DMA2时钟开启。
0	DMA1EN ：DMA1时钟使能（DMA1 clock enable）由软件置‘1’或清‘0’。 0：DMA1时钟关闭； 1：DMA1时钟开启。

6.3.7 APB2外设时钟使能寄存器(RCC_APB2ENR)

偏移地址：0x18

复位值：0x0000 0000

访问：字，半字和字节访问

通常无访问等待周期。但在APB2总线上的外设被访问时，将插入等待状态直到APB2的外设访问结束。

注：当外设时钟没有启用时，软件不能读出外设寄存器的数值，返回的数值始终是0x0。

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
ADC 3 EN	USA RT1 EN	TIM8 EN	SPI1 EN	TIM1 EN	ADC 2 EN	ADC 1 EN	IOPG EN	IOPF EN	IOPE EN	IOPD EN	IOPC EN	IOPB EN	IOPA EN	保留	AFIO EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

31:16	保留，始终读为0。
15	ADC3EN ：ADC3接口时钟使能（ADC 3 interface clock enable）由软件置'1'或清'0' 0：ADC3接口时钟关闭； 1：ADC3接口时钟开启。
14	USART1EN ：USART1时钟使能（USART1 clock enable）由软件置'1'或清'0' 0：USART1时钟关闭； 1：USART1时钟开启。
13	TIM8EN ：TIM8定时器时钟使能（TIM8 Timer clock enable）由软件置'1'或清'0' 0：TIM8定时器时钟关闭； 1：TIM8定时器时钟开启。
12	SPI1EN ：SPI1时钟使能（SPI 1 clock enable）由软件置'1'或清'0' 0：SPI1时钟关闭； 1：SPI1时钟开启。
11	TIM1EN ：TIM1定时器时钟使能（TIM1 Timer clock enable）由软件置'1'或清'0' 0：TIM1定时器时钟关闭； 1：TIM1定时器时钟开启。
10	ADC2EN ：ADC2接口时钟使能（ADC 2 interface clock enable）由软件置'1'或清'0' 0：ADC2接口时钟关闭； 1：ADC2接口时钟开启。
9	ADC1EN ：ADC1接口时钟使能（ADC 1 interface clock enable）由软件置'1'或清'0' 0：ADC1接口时钟关闭； 1：ADC1接口时钟开启。
8	IOPGEN ：IO端口G时钟使能（I/O port G clock enable）由软件置'1'或清'0' 0：IO端口G时钟关闭； 1：IO端口G时钟开启。

7	IOPFEN : IO端口F时钟使能 (I/O port F clock enable) 由软件置'1'或清'0' 0: IO端口F时钟关闭; 1: IO端口F时钟开启。
6	IOPEEN : IO端口E时钟使能 (I/O port E clock enable) 由软件置'1'或清'0' 0: IO端口E时钟关闭; 1: IO端口E时钟开启。
5	IOPDEN : IO端口D时钟使能 (I/O port D clock enable) 由软件置'1'或清'0' 0: IO端口D时钟关闭; 1: IO端口D时钟开启。
4	IOPCEN : IO端口C时钟使能 (I/O port C clock enable) 由软件置'1'或清'0' 0: IO端口C时钟关闭; 1: IO端口C时钟开启。
3	IOPBEN : IO端口B时钟使能 (I/O port B clock enable) 由软件置'1'或清'0' 0: IO端口B时钟关闭; 1: IO端口B时钟开启。
2	IOPAEN : IO端口A时钟使能 (I/O port A clock enable) 由软件置'1'或清'0' 0: IO端口A时钟关闭; 1: IO端口A时钟开启。
1	保留, 始终读为0。
0	AFIOEN : 辅助功能IO时钟使能 (Alternate function I/O clock enable) 由软件置'1'或清'0' 0: 辅助功能IO时钟关闭; 1: 辅助功能IO时钟开启。

6.3.8 APB1 外设时钟使能寄存器(RCC_APB1ENR)

偏移地址: 0x1C

复位值: 0x0000 0000

访问: 字、半字和字节访问

通常无访问等待周期。但在APB1总线上的外设被访问时, 将插入等待状态直到APB1外设访问结束。

注: 当外设时钟没有启用时, 软件不能读出外设寄存器的数值, 返回的数值始终是0x0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留	DAC EN	PWR EN	BKP EN	保留	CAN EN	保留	USB EN	I2C2 EN	I2C1 EN	UAR T5 EN	UAR T4 EN	USA RT3 EN	USA RT2 EN		
	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
SPI3	SPI2	保留			WWDG	保留		TIM7	TIM6	TIM5		TIM4		TIM3	

EN	EN		EN		EN	EN	EN	EN	EN
rw	rw		rw		rw	rw	rw	rw	rw

31:30	保留，始终读为0。
29	DACEN : DAC接口时钟使能 (DAC interface clock enable) 由软件置'1'或清'0' 0: DAC接口时钟关闭; 1: DAC接口时钟开启。
28	PWREN : 电源接口时钟使能 (Power interface clock enable) 由软件置'1'或清'0' 0: 电源接口时钟关闭; 1: 电源接口时钟开启。
27	BKPEN : 备份接口时钟使能 (Backup interface clock enable) 由软件置'1'或清'0' 0: 备份接口时钟关闭; 1: 备份接口时钟开启。
26	保留，始终读为0。
25	CANEN : CAN时钟使能 (CAN clock enable) 由软件置'1'或清'0' 0: CAN时钟关闭; 1: CAN时钟开启。
24	保留，始终读为0。
23	USBEN : USB时钟使能 (USB clock enable) 由软件置'1'或清'0' 0: USB时钟关闭; 1: USB时钟开启。
22	I2C2EN : I2C 2时钟使能 (I2C 2 clock enable) 由软件置'1'或清'0' 0: I2C 2时钟关闭; 1: I2C 2时钟开启。
21	I2C1EN : I2C 1时钟使能 (I2C 1 clock enable) 由软件置'1'或清'0' 0: I2C 1时钟关闭; 1: I2C 1时钟开启。
20	UART5EN : UART5时钟使能 (UART 5 clock enable) 由软件置'1'或清'0' 0: UART5时钟关闭; 1: UART5时钟开启。
19	UART4EN : UART4时钟使能 (UART 4 clock enable) 由软件置'1'或清'0' 0: UART4时钟关闭; 1: UART4时钟开启。
18	USART3EN : USART3时钟使能 (USART 3 clock enable) 由软件置'1'或清'0' 0: USART3时钟关闭; 1: USART3时钟开启。

17	USART2EN : USART2时钟使能 (USART 2 clock enable) 由软件置'1'或清'0' 0: USART2时钟关闭; 1: USART2时钟开启。
16	保留, 始终读为0。
15	SPI3EN : SPI 3时钟使能 (SPI 3 clock enable) 由软件置'1'或清'0' 0: SPI 3时钟关闭; 1: SPI 3时钟开启。
14	SPI2EN : SPI 2时钟使能 (SPI 2 clock enable) 由软件置'1'或清'0' 0: SPI 2时钟关闭; 1: SPI 2时钟开启。
13:12	保留, 始终读为0。
11	WWDGEN : 窗口看门狗时钟使能 (Window watchdog clock enable) 由软件置'1'或清'0' 0: 窗口看门狗时钟关闭; 1: 窗口看门狗时钟开启。
10:6	保留, 始终读为0。
5	TIM7EN : 定时器7时钟使能 (Timer 7 clock enable) 由软件置'1'或清'0' 0: 定时器7时钟关闭; 1: 定时器7时钟开启。
4	TIM6EN : 定时器6时钟使能 (Timer 6 clock enable) 由软件置'1'或清'0' 0: 定时器6时钟关闭; 1: 定时器6时钟开启。
3	TIM5EN : 定时器5时钟使能 (Timer 5 clock enable) 由软件置'1'或清'0' 0: 定时器5时钟关闭; 1: 定时器5时钟开启。
2	TIM4EN : 定时器4时钟使能 (Timer 4 clock enable) 由软件置'1'或清'0' 0: 定时器4时钟关闭; 1: 定时器4时钟开启。
1	TIM3EN : 定时器3时钟使能 (Timer 3 clock enable) 由软件置'1'或清'0' 0: 定时器3时钟关闭; 1: 定时器3时钟开启。
0	TIM2EN : 定时器2时钟使能 (Timer 2 clock enable) 由软件置'1'或清'0' 0: 定时器2时钟关闭; 1: 定时器2时钟开启。

6.3.9 备份域控制寄存器 (RCC_BDCR)

偏移地址：0x20

复位值：0x0000 0000，只能由备份域复位有效复位访问：0到3等待周期，字、半字和字节访问

当连续对该寄存器进行访问时，将插入等待状态。

注意： 备份域控制寄存器中(RCC_BDCR)的LSEON、LSEBYP、RTCSEL和RTCEN位处于备份域。因此，这些位在复位后处于写保护状态，只有在电源控制寄存器(PWR_CR)中的DBP位置'1'后才能对这些位进行改动。进一步信息请参考5.1节。这些位只能由备份域复位清除(见6.1.3节)。任何内部或外部复位都不会影响这些位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留								BDRST							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTC EN		保留			RTCSEL[1:0]			保留		LSEBYP		LSERDY		LSEON	
rw					rw					rw		r		rw	

31:17	保留，始终读为0。
16	BDRST ：备份域软件复位 (Backup domain software reset) 由软件置'1'或清'0' 0：复位未激活； 1：复位整个备份域。
15	RTCEN ：RTC时钟使能 (RTC clock enable) 由软件置'1'或清'0' 0：RTC时钟关闭； 1：RTC时钟开启。
14:10	保留，始终读为0。
9:8	RTCSEL[1:0] ：RTC时钟源选择 (RTC clock source selection) 由软件设置来选择RTC时钟源。一旦RTC时钟源被选定，直到下次后备域被复位，它不能在改变。可通过设置BDRST位来清除。 00：无时钟； 01：LSE振荡器作为RTC时钟； 10：LSI振荡器作为RTC时钟； 11：HSE振荡器在128分频后作为RTC时钟。
7:3	保留，始终读为0。
2	LSEBYP ：外部低速时钟振荡器旁路 (External low-speed oscillator bypass) 在调试模式下由软件置'1'或清'0'来旁路LSE。只有在外部32kHz振荡器关闭时，才能写入该位 0：LSE时钟未被旁路； 1：LSE时钟被旁路。

1	LSERDY : 外部低速LSE就绪 (External low-speed oscillator ready) 由硬件置'1'或清'0'来指示是否外部32kHz振荡器就绪。在LSEON被清零后, 该位需要6个外部低速振荡器的周期才被清零。 0: 外部32kHz振荡器未就绪; 1: 外部32kHz振荡器就绪。
0	LSEON : 外部低速振荡器使能 (External low-speed oscillator enable) 由软件置'1'或清'0' 0: 外部32kHz振荡器关闭; 1: 外部32kHz振荡器开启。

6.3.10 控制/状态寄存器 (RCC_CSR)

偏移地址: 0x24

复位值: 0x0C00 0000, 除复位标志外由系统复位清除, 复位标志只能由电源复位清除。

访问: 0到3等待周期, 字、半字和字节访问当连续对该寄存器进行访问时, 将插入等待状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
LPWRR STF		WWDGR STF		IWDG RSTF		SFT RSTF		POR RSTF		PIN RSTF		保留		RMVF		保留
rw		rw		rw		rw		rw		rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
保留						LSIRDY					LSION					

31	LPWRRSTF : 低功耗复位标志 (Low-power reset flag) 在低功耗管理复位发生时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无低功耗管理复位发生; 1: 发生低功耗管理复位。 关于低功耗管理复位的详细信息, 请参考6.1.1节的"低功耗管理复位"。
30	WWDGRSTF : 窗口看门狗复位标志 (Window watchdog reset flag) 在窗口看门狗复位发生时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无窗口看门狗复位发生; 1: 发生窗口看门狗复位。
29	IWDGRSTF : 独立看门狗复位标志 (Independent watchdog reset flag) 在独立看门狗复位发生在V _{DD} 区域时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无独立看门狗复位发生; 1: 发生独立看门狗复位。
28	SFTRSTF : 软件复位标志 (Software reset flag) 在软件复位发生时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无软件复位发生; 1: 发生软件复位。

27	PORRSTF : 上电/掉电复位标志 (POR/PDR reset flag) 在上电/掉电复位发生时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无上电/掉电复位发生; 1: 发生上电/掉电复位。
26	PINRSTF : NRST引脚复位标志 (PIN reset flag) 在NRST引脚复位发生时由硬件置'1'; 由软件通过写RMVF位清除。 0: 无NRST引脚复位发生; 1: 发生NRST引脚复位。
25	保留, 读操作返回0
24	RMVF : 清除复位标志 (Remove reset flag) 由软件置'1'来清除复位标志。 0: 无作用; 1: 清除复位标志。
23:2	保留, 读操作返回0
1	LSIRDY : 内部低速振荡器就绪 (Internal low-speed oscillator ready) 由硬件置'1'或清'0'来指示内部40kHz RC振荡器是否就绪。在LSION清零后, 3个内部40kHz RC 振荡器的周期后LSIRDY被清零。 0: 内部40kHz RC振荡器时钟未就绪; 1: 内部40kHz RC振荡器时钟就绪。
0	LSION : 内部低速振荡器使能 (Internal low-speed oscillator enable) 由软件置'1'或清'0'。 0: 内部40kHz RC振荡器关闭; 1: 内部40kHz RC振荡器开启。

6.2.11 MCO分频控制寄存器 (RCC_MCO_VAL)

偏移地址: 0x30

复位值: 0x0000 0001

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								MCOVAL							
								rw							

31:4	保留
3:0	MCOVAL 拓展分频控制 默认值为0001, 为不开启拓展 开启拓展条件为RCC_CFGR的MCO位配置为0111, 此时配置MCOVAL将生效。 0010 PLL时钟3分频后输出 0011 PLL时钟4分频后输出

 1111 PLL时钟16分频后输出
--	----------------------------