UM32x130/131、UM32MP31/P32

用户手册

版本：V1.2

广芯微电子（广州）股份有限公司

http://www.unicmicro.com/

条款协议

本文档的所有部分，其著作产权归广芯微电子（广州）股份有限公司（以下简称广芯微电子）所有，未经广芯微电子授权许可，任何个人及组织不得复制、转载、仿制本文档的全部或部分组件。本文档没有任何形式的担保、立场表达或其他暗示，若有任何因本文档或其中提及的产品所有资讯所引起的直接或间接损失，广芯微电子及所属员工恕不为其担保任何责任。除此以外，本文档所提到的产品规格及资讯仅供参考，内容亦会随时更新，恕不另行通知。

1. 本文档中所记载的关于电路、软件和其他相关信息仅用于说明半导体产品的操作和应用实例。用户如在设备设计中应用本文档中的电路、软件和相关信息，请自行负责。对于用户或第三方因使用上述电路、软件或信息而遭受的任何损失，广芯微电子不承担任何责任。
2. 在准备本文档所记载的信息的过程中，广芯微电子已尽量做到合理注意，但是，广芯微电子并不保证这些信息都是准确无误的。用户因本文档中所记载的信息的错误或遗漏而遭受的任何损失，广芯微电子不承担任何责任。
3. 对于因使用本文档中的广芯微电子产品或技术信息而造成的侵权行为或因此而侵犯第三方的专利、版权或其他知识产权的行为，广芯微电子不承担任何责任。本文档所记载的内容不应视为对广芯微电子或其他人所有的专利、版权或其他知识产权作出任何明示、默示或其它方式的许可及授权。
4. 使用本文档中记载的广芯微电子产品时，应在广芯微电子指定的范围内，特别是在最大额定值、电源工作电压范围、热辐射特性、安装条件以及其他产品特性的范围内使用。对于在上述指定范围之外使用广芯微电子产品而产生的故障或损失，广芯微电子不承担任何责任。
5. 虽然广芯微电子一直致力于提高广芯微电子产品的质量和可靠性，但是，半导体产品有其自身的具体特性，如一定的故障发生率以及在某些使用条件下会发生故障等。此外，广芯微电子产品均未进行防辐射设计。所以请采取安全保护措施，以避免当广芯微电子产品在发生故障而造成火灾时导致人身事故、伤害或损害的事故。例如进行软硬件安全设计（包括但不限于冗余设计、防火控制以及故障预防等）、适当的老化处理或其他适当的措施等。

目录

图目录

表目录

1 文档约定

1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

R/W	可读可写，软件可以读写该位
R	只读，软件只能读取该位
W	只写，软件只能写入该位，软件读返回复位值
R/W1C	可读，写 1 清零软件可以读取该位。写 1 清除该位，写 0 无效
R/W0C	可读，写 0 清除，软件可以读取该位。写 0 清除该位，写 1 无效
RSV	保留位，必须保持复位值

1.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- CPU 内核集成了 SWD 调试端口 (SWD-DP)：提供基于串行线调试 (SWD) 协议的 2 引脚（时钟和数据）接口。
- 字：32 位数据/指令。
- 半字：16 位数据/指令。
- 字节：8 位数据。
- 双字：64 位数据。
- IAP（在应用中编程）：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- ICP（在线编程）：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。
- 选项字节：存储于 Flash 中的产品配置位。
- OBL：选项字节加载器。
- AHB：高级高性能总线。
- APB：低速外设总线

2 产品简介

2.1 系统概述

UM32x130/131、UM32MP31/P32 系列芯片是广芯微电子（广州）股份有限公司研制的基于 ARM Cortex-M0+内核的超低功耗、Low Pin Count、宽电压工作范围的 32 位 IoTP 处理器 SoC 芯片系列，重点面向物联网行业便携式传感测量系统、汽车电动化、智能家居、电机控制及工业控制等应用场景。依据行业应用场景的具体应用需求，芯片系统采用了独特的低功耗设计技术，内部集成了 CAN、LIN、12 位 SAR ADC、UART、SPI、I2C 等通用外围通讯接口，ADC、OPA、COMP 等传感获取接口，LPUART、LPTIMER、WDT 等超低功耗模块接口，以及 AES、DIV 除法器等硬件算法模块。具有高整合度、高抗干扰、高可靠性和超低功耗等技术特点。内置 RC 高频和低频振荡器，支持免晶振应用。支持 Keil MDK 集成开发环境，支持 C 语言和汇编语言进行软件开发。

典型应用场景

- 工业物联网应用
- 智能交通，智慧城市，智能家居
- 智能门锁，资产追踪、无线监控等智能传感器终端应用
- 电池供电应用

2.2 主要特性

- **超低功耗电源管理系统**
  - 1.2μA @3.0V DeepSleep+RTC 模式，RCL 运行，IO、SRAM 以及寄存器数据保持
  - 0.9μA @3.0V Stop 模式，所有时钟停止，IO、SRAM 以及寄存器数据保持
  - 50μA/MHz @3.0V @32MHz Active 模式
  - 3.7μs 快速睡眠唤醒系统
  - 低功耗模块 LPTIMER、LPUART、RTC、WDT
  - 内置 ROSC/LDO/POR，可免晶振/LDO/复位电路
- **处理器**
  - 32 位 ARM Cortex-M0+，系统最高主频 32MHz
  - 单周期硬件乘法器
- **存储器**
  - 64KB/32KB FLASH
  - 1KB EEPROM
  - 8KB SRAM
  - Sector 大小：512B，擦写次数：20,000 次
  - 数据保存时间：10 年@常温
- **时钟**
  - 外部高速晶振 4MHz 到 24MHz
  - 外部低速晶振 32.768KHz
  - 内部高速时钟 32MHz
  - 内部低速时钟 32KHz
- **模拟外设**
  - 1 个 12 位 1Msps ADC 转换器，多达 14 通道
  - 1 个运算放大器
  - 4 个电压比较器
- **通信接口**
  - 最大 3 路 UART 串口
  - 最大 2 路低功耗 LPUART 串口
  - 2 路通用的 SPI 接口
  - 1 路 LIN 接口（仅 UM32A130 支持）
  - 1 路 I2C 接口
  - 1 路 CAN 接口（仅 UM32A130 及 UM32G131 支持）

- **定时器**
  - 1 个 16 位高级定时器 ATIMER 支持输入捕获、死区互补 PWM 输出
  - 3 个 16 位通用定时器 GTIMER 支持输入捕获、PWM 输出
  - 4 个 16 位 BTIMER，支持 4 路 PWM 输出
  - 2 个 32 位低功耗定时器 LPTIMER 支持 4 路 PWM 输出
  - 1 个低功耗 RTC 定时/计数器
  - 1 个 32 位低功耗看门狗 WDT，可复位/中断
  - 1 个 18 位窗口看门狗 WWDT，可复位/中断
  - 最大共支持 21 路 PWM 输出
- **Buzzer 蜂鸣器**
  - 输出频率和极性可配置
- **GPIO 通用输入/输出端口**
  - 最大 30 个通用输入/输出管脚
  - 支持边沿/电平触发中断
  - 16/8mA 两档驱动能力可配置
- **安全功能**
  - 防抄板设计，防止 eFlash 中程序被盗取
  - 支持 AES（128/192/256）算法
  - 低电压检测 LVD
  - 掉电复位 LVR，防死机设计
  - CRC16-CCITT 数据校验算法硬件加速
  - RNG 硬件随机数发生器
  - 16 字节全球唯一芯片序列号 ID
- **硬件加速引擎**
  - 除法器 DIV
- **支持 SIP 预驱**
  - UM32MP31
    - ✓ 内置 6 路 NMOS Driver
    - ✓ 栅极驱动电压从 8V 到 20V
    - ✓ 输出级拉电流/灌电流能力 1.5A/1.8A
  - UM32MP32
    - ✓ 内置 6 路 PMOS+NMOS Driver
    - ✓ 内置 5V/40mA LDO
    - ✓ 栅极驱动电压从 5V 到 40V
    - ✓ 输出级拉电流/灌电流能力 50mA/300mA

- **主要电气参数**
  - 工作电压：2.5 ~ 5.5V
  - 工作温度：-40 ~ +85℃
  - ESD 防护：±8KV(HBM)
- **开发支持**
  - 内置 Boot 引导程序，支持 UART 下载，支持 ISP 和 IAP 应用程序更新
  - JTAG->SWD 模式在线调试/下载功能
  - 完整 SDK 开发包、EVB 硬件开发套件
  - 离线烧录器

3 存储器和总线架构

3.1 系统架构

主系统由以下部分构成：

- 2个AHB总线Master：
  - Cortex-M0+
  - DMA 控制器
- 3个AHB总线Slaves：
  - FLASH 存储器
  - SRAM 存储器
  - AHB，AHB to APB Bridge，包含所有APB接口外设，包含所有AHB接口外设

3.2 总线架构图

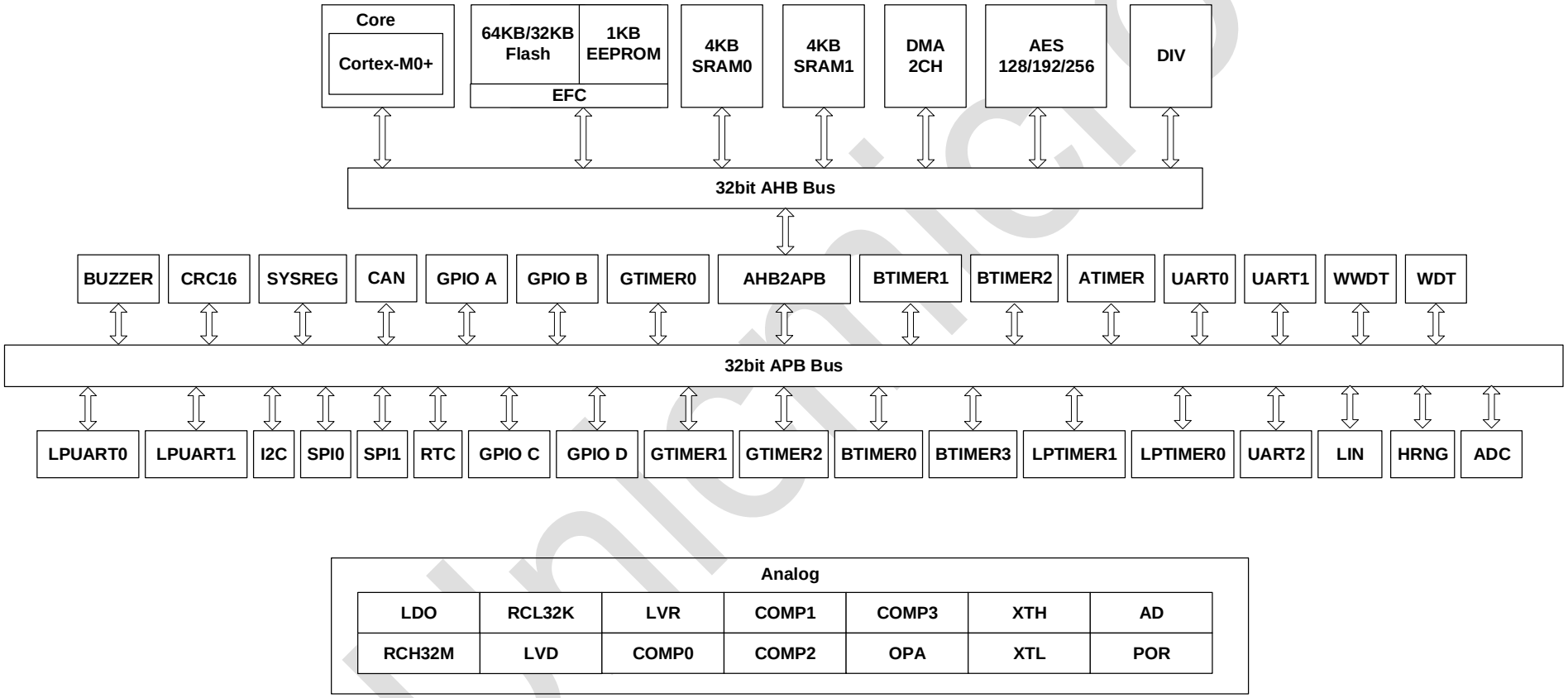


图 3-1：总线架构图

3.3 存储器映射（Memory Mapping）

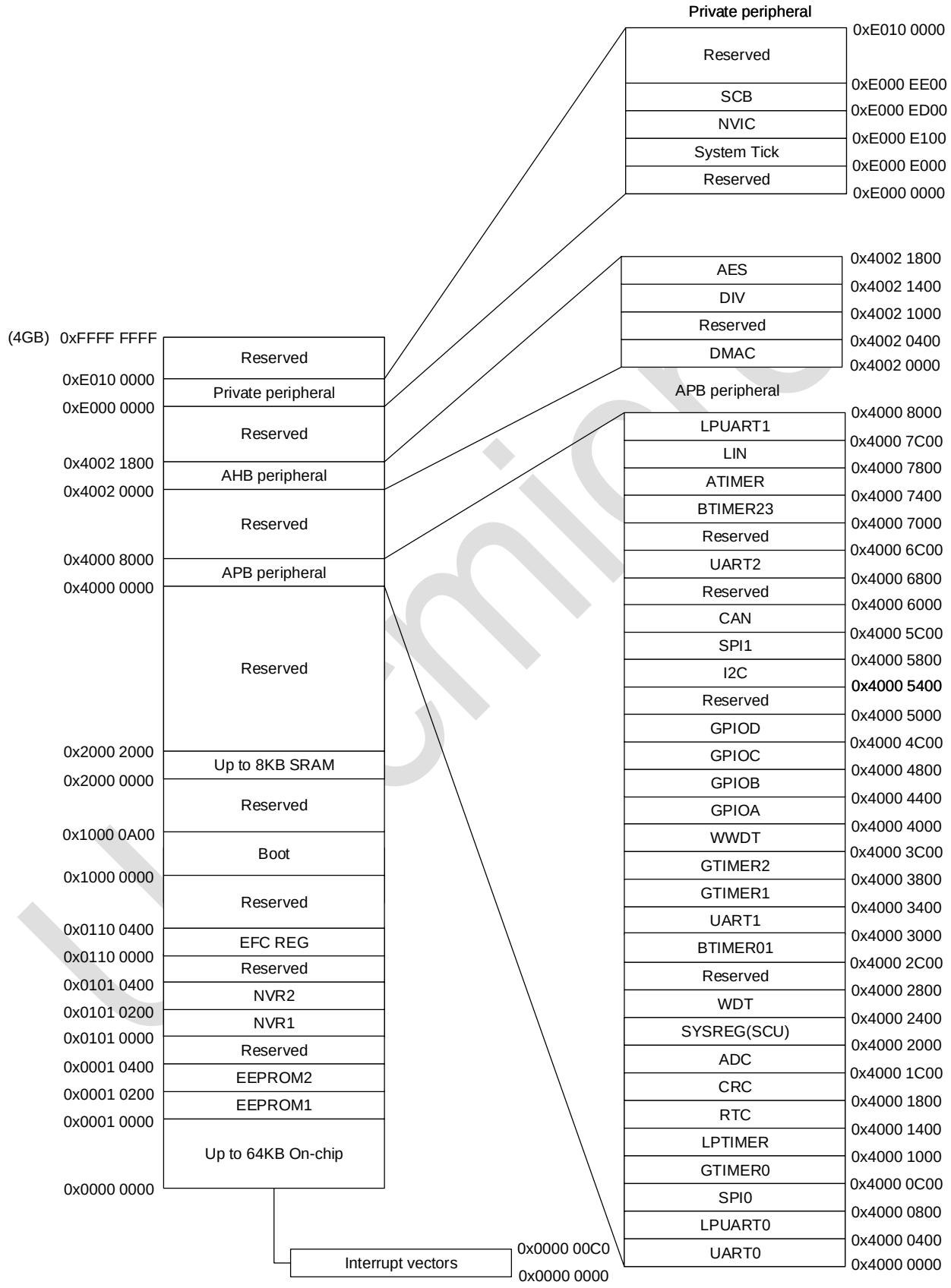


图 3-2：存储器地址映射图

表 3-1：存储器和模块地址分配

模块名	地址分配	大小
EFLASH	0x0000_0000——0x0001_0000	64kBytes
EEPROM	0x0001_0000——0x0001_0400	1KBytes
NVR1	0x0101_0000——0x0101_0200	512Bytes
NVR2	0x0101_0200——0x0101_0400	512Bytes
EFC Register	0x0110_0000——0x0110_006C	108Bytes
BootLoader	0x1000_0000——0x1000_0A00	2.5KBytes
SRAM1	0x2000_0000——0x2000_1000	4kBytes
SRAM2	0x2000_1000——0x2000_2000	4kBytes
UART0	0x4000_0000——0x4000_0400	1kBytes
LPUART0	0x4000_0400——0x4000_0800	1kBytes
SPI0	0x4000_0800——0x4000_0C00	1kBytes
GTIMER0	0x4000_0C00——0x4000_1000	1kBytes
LPTIMER	0x4000_1000——0x4000_1400	1kBytes
RTC	0x4000_1400——0x4000_1800	1kBytes
CRC	0x4000_1800——0x4000_1C00	1kBytes
ADC	0x4000_1C00——0x4000_2000	1kBytes
SYSREG(SCU)	0x4000_2000——0x4000_2400	1kBytes
WDT	0x4000_2400——0x4000_2800	1kBytes
BTIMER01	0x4000_2C00——0x4000_3000	1kBytes
UART1	0x4000_3000——0x4000_3400	1kBytes
GTIMER1	0x4000_3400——0x4000_3800	1kBytes
GTIMER2	0x4000_3800——0x4000_3C00	1kBytes
WWDT	0x4000_3C00——0x4000_4000	1kBytes
GPIOA	0x4000_4000——0x4000_4400	1kBytes
GPIOB	0x4000_4400——0x4000_4800	1kBytes
GPIOC	0x4000_4800——0x4000_4C00	1kBytes
GPIOD	0x4000_4C00——0x4000_5000	1kBytes
I2C	0x4000_5400——0x4000_5800	1kBytes
SPI1	0x4000_5800——0x4000_5C00	1kBytes
CAN	0x4000_5C00——0x4000_6000	1Kbytes
UART2	0x4000_6800——0x4000_6C00	1Kbytes
BTIMER23	0x4000_7000——0x4000_7400	1Kbytes
ATIMER	0x4000_7400——0x4000_7800	1Kbytes
LIN	0x4000_7800——0x4000_7C00	1Kbytes
LPUART1	0x4000_7C00——0x4000_8000	1Kbytes
DMAC	0x4002_0000——0x4002_0400	1kBytes
DIV	0x4002_1000——0x4002_1400	1kBytes
AES	0x4002_1400——0x4002_1800	1kBytes

4 处理器

4.1 概述

Cortex™ M0+处理器是 32 位的两级流水线 RISC 处理器，内嵌 AMBA-Lite 接口和嵌套向量中断控制器（NVIC）。具有硬件调试功能，可以执行 Thumb 指令，并与其它 Cortex-M 系列兼容。同时加入多项全新设计，改进调试和追踪能力、减少每条指令循环（IPC）数量和改进 Flash 访问的两级流水线等，更纳入了节能降耗技术。Cortex M0+处理器全面支持已整合 Keil & IAR 调试器。

4.2 主要特性

- ARMv6 M Thumb
- Thumb/Thumb 2 技术
- ARMv6 M 兼容 24 位 SysTick 定时器
- 32 位硬件乘法器
- 系统接口支持小端（little-endian）数据访问
- 准确而及时的中断处理能力
- 加载、存储多个数据和多周期乘法指令可被终止然后重新开始从而实现快速中断处理
- C 应用程序二进制接口的异常兼容模式（C-ABI）。ARMv6 M 的模式允许用户使用纯 C 函数实现中断处理
- 使用中断唤醒（WFI）与事件唤醒（WFE）指令进入低功耗的休眠模式，或者从中断退出休眠模式

4.3 功能框图

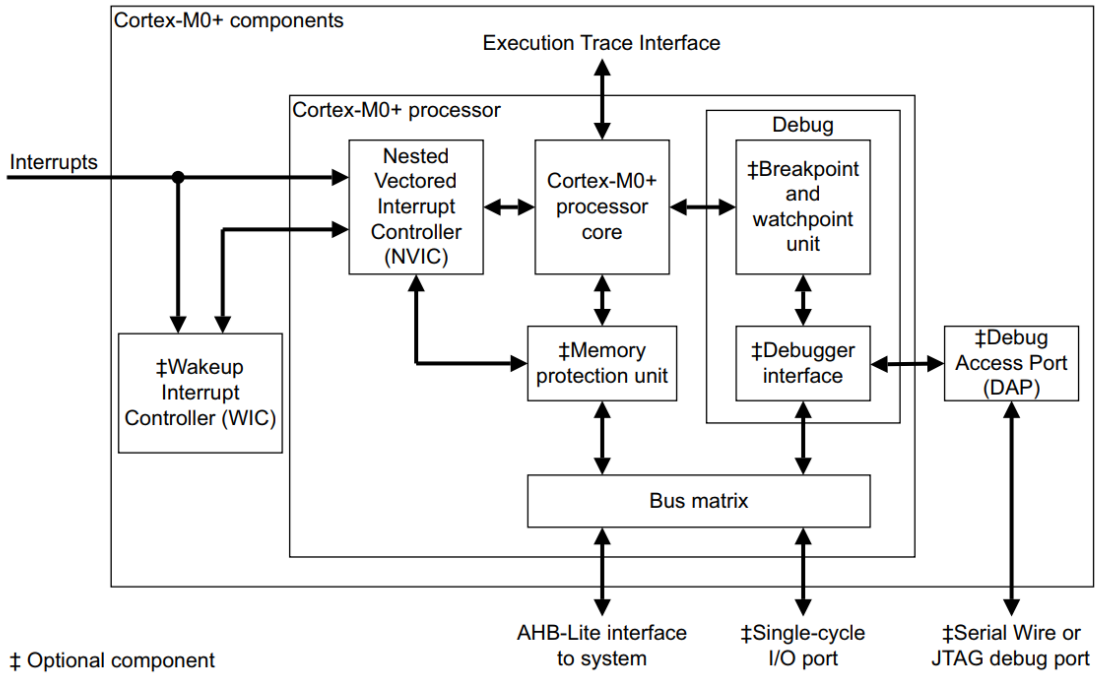


图 4-1：Cortex-M0+处理器功能框图

4.4 内核寄存器组

Cortex-M0+处理器寄存器组如下图：

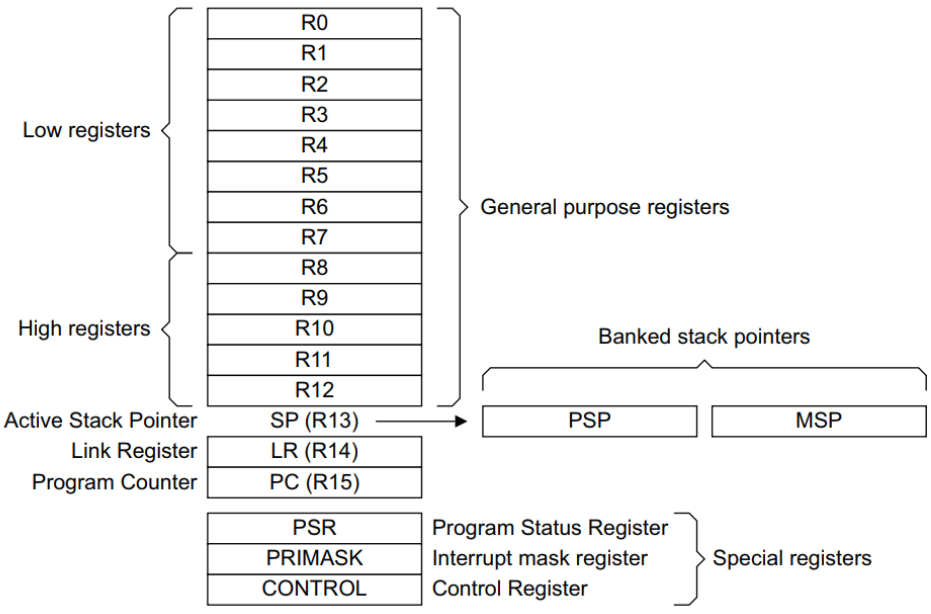


图 4-2：Cortex-M0+的寄存器组

5 系统配置(SCU)

5.1 时钟框图

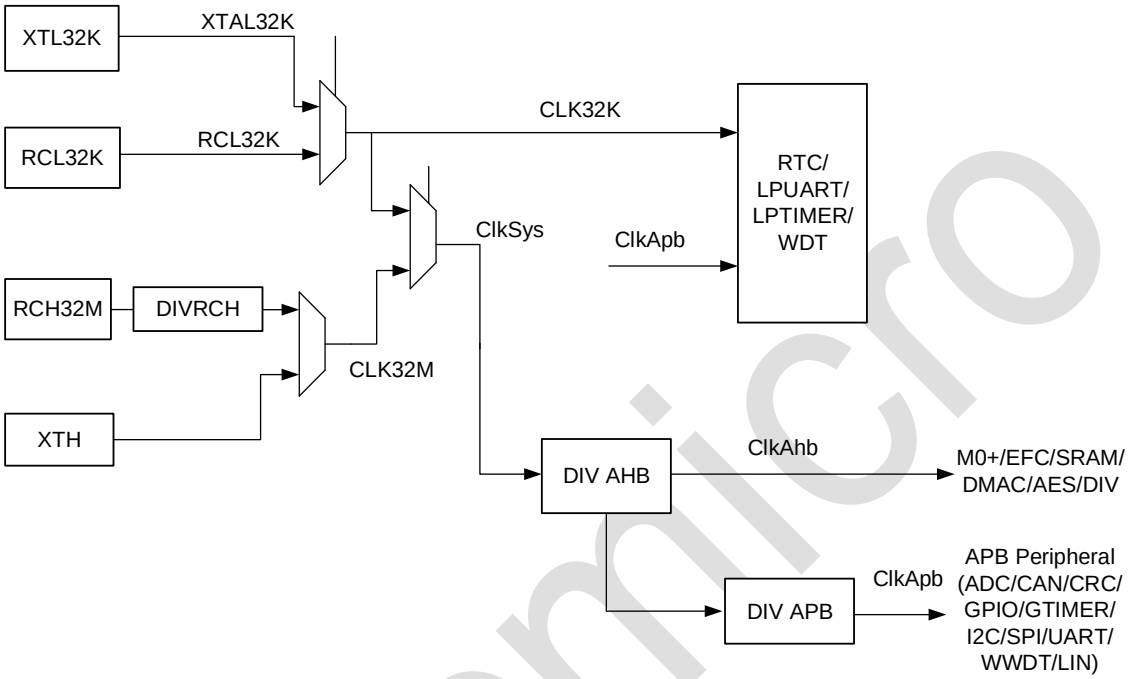


图 5-1：时钟模块框图

5.2 时钟选择

芯片共有四个系统时钟源：

1. 32MHz 高精度内部时钟 RCH，作为系统时钟源。
2. 4M~24MHz 的外部晶振 XTH，作为系统时钟源。
3. 32KHz 的内部时钟 RCL，作为低功耗时钟，可作为系统时钟源
4. 32.768KHz 的外部晶振 XTL，主要提供 RTC 实时时钟，也可作为系统时钟源。

根据工作模式不同，采用不同时钟方案，通过配置系统控制寄存器 0（SYSCTRL0）[14:12]位 CLK_SEL，CLK_SEL_HF 和 CLK_SEL_LF 来选择系统时钟的来源。关系如下表所示：

表 5-1：系统时钟选择

CLK_SEL	CLK_SEL_HF	系统时钟来源
0	0	RCH
0	1	XTH
CLK_SEL	**CLK_SEL_LF**	**系统时钟来源**
1	0	RCL
1	1	XTL

5.3 复位源

芯片有多个复位源，包括 POR 复位，RESETEN 复位，WDT 复位，WWDT 复位，SOFT_RESETN 复位，模块软件复位，LVD 复位，LOCKUP 复位。具体复位源如下表：

表 5-2：系统复位源

复位源	描述
内部模拟 POR 上电复位	复位所有
LVR 复位	
RESETEN 复位	复位除 CPU DEBUG 逻辑外的所有
LOCKUP 复位	复位除 EFC和IO相关 以外的其它逻辑
LVD 复位	
WDT	
WWDT	
SOFT_RESETN	
各模块复位	复位对应 IP 模块

注：Px_SEL/PAD_ADS/PAD_DR/PAD_PU/PAD_PD/PAD_OD/PAD_CS/PAD_IE/PAD_SR 寄存器只能由 POR、LVR 和 RESETEN 外部复位进行复位。

5.3.1 内部 POR 上电复位

内部上电复位 POR：无条件复位整个芯片。

5.3.2 LVR 复位

LVR 复位：无条件复位整个芯片。

5.3.3 RESETEN 复位

外部复位 RESETEN 复位除 CPU DEBUG 逻辑外的整个芯片。在 RESETEN 复位状态时，芯片可以连接 SWD 接口。RESETEN 管脚在上电完成后默认作为外部复位，可以通过软件关闭外部复位功能。

5.3.4 LOCKUP 复位

当系统连续发生两次 HardFault 时，CPU 会进入 LOCKUP 状态，系统会产生 LOCKUP 复位。LOCKUP 复位除 EFC 和 IO 相关以外的其它逻辑。

5.3.5 LVD 复位

LVD 复位除 EFC 和 IO 相关以外的其它逻辑。

5.3.6 WDT 复位

看门狗定时器复位除 EFC 和 IO 相关外的整个用户电路，缺省为释放状态。

当软件未能有效阻止超时事件时，看门狗定时器复位。该复位仅发生在软件无法正常执行，可能破坏数据时。在 CPU 处于 HALT 状态时，WDT 停止计数，不会产生复位信号。

5.3.7 WWDT 复位

复位除 EFC 和 IO 相关以外的其它逻辑。

5.3.8 SOFT_RESETN 复位

此复位由系统产生。系统可以通过写软复位重启，但不复位 EFC 控制器和 IO 相关设置。

5.3.9 模块复位

模块复位，通过软件复位各数字模块。

5.4 低功耗模式

芯片除正常工作模式外，为了降低芯片的电流消耗，提供三种低功耗模式：休眠（Sleep）模式、深度休眠（Deepsleep）模式、停止（Stop）模式和低时钟运行（Lprun）模式。

在休眠模式下，CM0+停止工作，保留中断处理功能。其它外设等模块时钟和复位可由软件设置。休眠模式由 M0+特定指令 WFI / WFE 进入，唤醒由中断触发。

深度休眠模式是休眠模式的升级，在此模式下，CM0+停止运行，高速时钟停止运行，低功耗功能模块（LPTIMER、LPUART、RTC、WDT）可以运行。深度休眠模式要先设置 CM0+内部的 DEEPSLEEP 寄存器，然后由 M0+特定指令 WFI / WFE 进入，唤醒由中断触发。

停止模式下，高速时钟和低速时钟均停止运行，系统无任何运行的时钟，一切外围模块均停止

运行。上电复位信号有效，IO 状态保持，IO 中断有效，所有寄存器，RAM 和 CPU 数据保存状态时的功耗；停止模式要先设置系统寄存器中 STOPMODE_SEL 寄存器和 CM0+内部的 DEEPSLEEP 寄存器，然后由 M0+特定指令 WFI / WFE 进入，唤醒可以通过 GPIO 边沿/电平唤醒或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒。

低时钟运行模式是将系统时钟切换到 RCL 或 XTL 低频时钟源下的一种工作模式。在系统时钟选择为低频时钟源后，可以通过设置系统寄存器中的 LDO_SOFT 寄存器，使得 LDO 进入低功耗模式工作，降低动态功耗。注意，此芯片工作在 Lprun 模式下，只能通过复位恢复到正常工作模式。

详细的描述如下表：

表 5-3：低功耗模式

模式	模式描述	进入条件	退出条件
Sleep	LDO Active 供电，CPU 大部分休眠（包括 NVIC），WIC 不休眠；软件可关闭各模块时钟。	1. 根据需要，关闭各外设模块时钟，仅留下需要监测中断事件的模块。 2. 执行 WFI/WFE 指令。	1. CM0+检测到中断或事件发生。 2. 进入中断服务程序清中断并返回。 3. 继续执行后续指令。
DeepSleep	LDO Standby 供电，CPU 大部分休眠（包括 NVIC），WIC 不休眠；高速时钟源关闭，RCL 低速时钟源运行。	1. 根据需要，关闭各外设模块时钟，仅留下需要监测中断事件的模块。 2. 设置 CM0+ 内部的 DEEPSLEEP 寄存器。 3. 执行 WFI/WFE 指令。	1. CM0+检测到中断或事件发生。 2. 进入中断服务程序清中断并返回。 3. 继续执行后续指令。
Stop	LDO Standby 供电，关闭系统所有时钟。	1. 根据需要，设置 IO 唤醒的条件。 2. 设置 CM0+ 内部的 DEEPSLEEP 寄存器。 3. 设置系统寄存器中的 STOPMODE_SEL 寄存器。 4. 执行 WFI/WFE 指令。	1. 外部 IO 唤醒事件到来。 2. CM0+检测到 IO 唤醒事件中断发生。 3. 进入中断服务程序清中断并返回。 4. 继续执行后续指令。
Lprun	系统时钟切换到 RCL 或者 XTL 低频时钟源，LDO 进入低功耗模式工作。	1. 设置系统寄存器中的 SYSCTRL0 寄存器，将系统时钟切换到 RCL 或者 XTL 低频时钟源上。 2. 设置系统寄存器中的 LDO_SOFT 寄存器，将 LDO 设置进入低功耗模式。	系统经过复位，退出 Lprun 模式。

低功耗模式的进入和唤醒条件阐述如下：

- Sleep，DeepSleep，Stop 模式进入条件都需要设置 SCB->SCR 寄存器，调用 WFI/WFE；三者模式的唤醒本质是都是需要产生中断或者事件发生。

- Sleep 模式下，高速时钟 RCH、XTH 以及内部低速时钟 RCL、XTL 维持进入低功耗模式前的设置，只要系统产生中断就可以唤醒退出。
- DeepSleep 模式下，RCH、XTH 时钟关闭了，RCL 或 XTL 时钟根据设置在工作，所以只有工作在低频时钟源的模块如 WDT、RTC、LPTIMER、LPUART 可以产生中断唤醒退出，以及 GPIO 边沿/电平模式在无时钟情况下产生中断唤醒退出。
- Stop 模式下，所有时钟源关闭，可以通过 GPIO 边沿/电平模式在无时钟情况下产生中断唤醒退出，或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒退出。

5.4.1 Sleep 模式

进入 sleep 模式设置：

- SCB->SCR 的 bit2 配置为 0
- 调用 WFI/WFE 进入 sleep 模式

唤醒条件：中断唤醒。

5.4.2 DeepSleep 模式

DeepSleep 唤醒源包括：WDT、RTC、LPTIMER、LPUART（XTL 时钟源情况下）中断唤醒，GPIO 边沿/电平中断唤醒。

下面以 PD4 管脚下降沿唤醒为例阐述配置流程：

1. 配置外围模块时钟控制寄存器PERI_CLKEN[19]，使能GPIOD时钟。
2. 配置外围模块复位控制寄存器PERI_RESET[19]，GPIOD设置正常工作。
3. 配置端口PD功能寄存器PD_SEL[19:16]，配置PD4为GPIO功能。
4. 配置GPIOD数据方向寄存器GPIO_DIR[4]，PD4为输入。
5. 配置GPIOD中断触发模式寄存器GPIO_IS[4]，PD4边沿触发中断。
6. 配置GPIOD中断边沿触发设置寄存器GPIO_IBE[4]，PD4单边触发。
7. 配置GPIOD中断高低电平触发设置寄存器GPIO_IEV，下降沿触发。
8. 配置GPIOD中断使能寄存器GPIO_IEN[4]，使能PD4中断。
9. 配置端口输入配置寄存器 PAD_IE[28]，PD4输入使能。
10. 配置端口上拉配置寄存器 PAD_PU[28]，PD4上拉使能。
11. SCB -> SCR[2]配置为 1。
12. 调用 WFI 进入 Deepsleep 模式。
13. 中断产生后，能将系统从深度休眠模式下唤醒。

5.4.3 Stop 模式

Stop 模式唤醒源包括：GPIO 边沿/电平中断唤醒，或者通过 LPTIMER 外部异步脉冲计数产生中断唤醒。

下面以 PD4 管脚下降沿唤醒为例阐述配置流程：

1. 配置外围模块时钟控制寄存器PERI_CLKEN[19]，使能GPIOD时钟。
2. 配置外围模块复位控制寄存器PERI_RESET[19]，GPIOD设置正常工作。
3. 配置端口PD功能寄存器PD_SEL[19:16]，配置PD4为GPIO功能。
4. 配置GPIOD数据方向寄存器GPIO_DIR[4]，PD4为输入。
5. 配置GPIOD中断触发模式寄存器GPIO_IS[4]，PD4边沿触发中断。
6. 配置GPIOD中断边沿触发设置寄存器GPIO_IBE[4]，PD4单边触发。
7. 配置GPIOD中断高低电平触发设置寄存器GPIO_IEV，下降沿触发。
8. 配置GPIOD中断使能寄存器GPIO_IEN[4]，使能PD4中断。
9. 配置端口输入配置寄存器 PAD_IE[28]，PD4输入使能。
10. 配置端口上拉配置寄存器 PAD_PU[28]，PD4上拉使能。
11. 配置STOPMODE_SEL = 0XA5A50001; //STOP模式有效
12. SCB -> SCR[2]配置为 1。
13. 调用__WFI(); //进入STOP模式
14. 中断产生后，能将系统从Stop模式下唤醒。

5.5 系统寄存器

SYSREG（SCU）寄存器基地址：0x4000_2000

表 5-4：系统寄存器列表

偏置	名称	描述
0x000	SYSCTRL0	系统控制寄存器 0
0x004	SYSCTRL1	系统控制寄存器 1
0x008	SYSCTRL_PROTECT	系统控制保护寄存器
0x00C	OSC_CTRL	时钟控制寄存器
0x010	PERI_CLKEN	外围模块时钟寄存器
0x020	RESET_FLAG	复位标识寄存器
0x024	PERI_RESET	外围模块复位控制寄存器
0x028	EXT_RESETCTRL	外部复位滤波控制寄存器
0x030	PA_SEL	端口 PA 功能配置寄存器，只能外部复位和 POR 复位此寄存器
0x034	PB_SEL	端口 PB 功能配置寄存器，只能外部复位和 POR 复位此寄存器
0x038	PC_SEL	端口 PC 功能配置寄存器，只能外部复位和 POR 复位此寄存器

偏置	名称	描述
0x03C	PD_SEL	端口 PD 功能配置寄存器，只能外部复位和 POR 复位此寄存器
0x054	PAD_ADS	端口数模配置寄存器，只能外部复位和 POR 复位此寄存器
0x060	PAD_DR	端口驱动能力配置寄存器，只能外部复位和 POR 复位此寄存器
0x06C	PAD_PU	端口上拉配置寄存器，只能外部复位和 POR 复位此寄存器
0x078	PAD_PD	端口下拉配置寄存器，只能外部复位和 POR 复位此寄存器
0x084	PAD_OD	端口开漏输出配置寄存器，只能外部复位和 POR 复位此寄存器
0x090	PAD_CS	端口输入类型配置寄存器，只能外部复位和 POR 复位此寄存器
0x09C	PAD_IE	端口输入配置寄存器，只能外部复位和 POR 复位此寄存器
0x0A4	PAD_STATUS	端口输入电平寄存器，只能外部复位和 POR 复位此寄存器
0x0A8	PAD_SR	端口速率配置寄存器，只能外部复位和 POR 复位此寄存器
0x0B4	IOCTRL_PROTECT	IO 控制保护寄存器
0x0B8	LVD_CFG	LVD 控制寄存器
0x0D0	EXTRST_SEL	外部复位端口选择寄存器
0x0D4	STOPMODE_SEL	停止模式选择寄存器
0x0D8	REMAP_ADDR	REMAP 寄存器
0x0DC	VECTOR_OFFSET	中断向量地址重映射寄存器
0x0EC	BUZZER_CR	蜂鸣器控制寄存器
0x0FC	ANALOG_STATUS	模拟状态寄存器
0x110	LDO_SOFT	LDO 低功耗软件控制寄存器
0x114	PDSTBB_SOFT	FLASH 低功耗软件控制寄存器

5.5.1 系统控制寄存器 0 SYSCTRL0（偏移：000h）

比特	名称	属性	复位值	描述
31:30	RSV	-	-	保留
29:27	CLKOUT_SEL	R/W	0	PC2 作为时钟输出时，输出时钟选择： 000：HCLK_OUT（系统时钟） 001：XTH 晶振时钟 010：RCL 时钟 011：XTL 时钟 100：PCLK_OUT 101：RTC_1Hz 111：RCH_DIV 其它：保留

比特	名称	属性	复位值	描述
26:24	CLK16M_DIV	R/W	1	RTC 16.384M 时钟分频设置寄存器： 00：RCH 和 XTH 不分频 01：RCH 和 XTH 经过选择后的时钟二分频 10：RCH 和 XTH 经过选择后的时钟三分频 11：RCH 和 XTH 经过选择后的时钟四分频
23	RSV	-	-	保留
22	LPTIMER_EXTIG_SEL	R/W	0	LPTIMER0 的 EXTIG（外部触发）信号来源选择信号： 1：EXTIG 来自于 RTC 的 1HZ 时钟输出 0：EXTIG 来自于外部引脚
21:19	RSV	-	-	保留
18:17	RCH_DIV	R/W	1	RCH 时钟源分频设置位： 00：不分频 01：二分频 10：三分频 11：四分频
16	SWD_WACK_EN	R/W	1	连接 ULINK 或者 JLINK 后，在 DEEPSLEEP 或者 STOP 模式下，唤醒系统的使能位： 1：在上述条件下，使用 NMI 中断唤醒系统 0：在上述条件下，不唤醒系统
15	RSV	-	-	保留
14	CLK_SEL	R/W	0	系统时钟来源选择： 1：低速时钟 CLK_SEL_LF 0：高速时钟 CLK_SEL_HF
13	CLK_SEL_HF	R/W	0	1：高频时钟选择外部高速时钟 XTH 0：高频时钟选择内部高速时钟 RCH
12	CLK_SEL_LF	R/W	0	1：低频时钟选择外部低速时钟 XTL 0：低频时钟选择内部低速时钟 RCL
11	RSV	-	-	保留
10:9	PCLK_DIV	R/W	0	PCLK 分频选择： 00：HCLK 01：HCLK/2 10：HCLK/4 11：HCLK/8
8:6	HCLK_DIV	R/W	0	HCLK 分频选择： 000：SystemClk 001：SystemClk/2 010：SystemClk/4 011：SystemClk/8 100：SystemClk/16 101：SystemClk/32 110：SystemClk/64 111：SystemClk/128

比特	名称	属性	复位值	描述
5:4	RSV	-	-	保留
3	XTL_EN	R/W	0	外部低速晶振 XTL 使能控制： 1：使能 0：关闭 此位寄存器只能由 POR 和外部管脚复位。
2	RCL_EN	R/W	1	内部低速时钟 RCL 使能控制： 1：使能 0：关闭
1	XTH_EN	R/W	0	外部高速晶振 XTH 使能控制： 1：使能 0：关闭 注：当系统进入 DeepSleep，此高速时钟会自动关闭。此位寄存器只能由 POR 和外部管脚复位。
0	RCH_EN	R/W	1	内部高速时钟 RCH 使能信号： 1：使能 0：关闭 注：当系统进入 DeepSleep，此高速时钟会自动关闭。

5.5.2 系统控制寄存器 1 SYSCTRL1（偏移：004h）

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	EXTL_EN	R/W	0	外部 XTL 时钟输入控制： 1：XTL 输出时钟从 PA0 输入 0：XTL 输出时钟由晶振产生 此位寄存器只能由 POR 和外部管脚复位。
0	EXTH_EN	R/W	0	外部 XTH 输入控制： 1：XTH 输出时钟从 PC3 输入 0：XTH 输出时钟由晶振产生 此位寄存器只能由 POR 和外部管脚复位。

5.5.3 系统控制保护寄存器 SYSCTRL_PROTECT（偏移：008h）

比特	名称	属性	复位值	描述
31:0	SYSCTRL_PROTECT	R/W	0	寄存器 SYSCTRL0 和 SYSCTRL1 写保护的控制寄存器。给此寄存器写 0xA5A5_5A5A，启动寄存器 SYSCTL0 和 SYSCTRL1 的写使能。给此寄存器写其他值，关闭它们的写使能。SYSCTRL0 或 SYSCTRL1 寄存器配置完后，它们的写使能会自动关闭。 读取此寄存器返回 SYSCTRL0 和 SYSCTRL1 寄存器的写使能状态。 1：写已经使能 0：写未使能

5.5.4 时钟控制寄存器 OSC_CTRL（偏移：0x00Ch）

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:16	WAKP_delay	R/W	12’h138	系统从 DEEPSLEEP/STOP 模式唤醒，CLK 给出的时间延时。此寄存器的时间为系统时钟计数的时钟周期。
15	RSV	-	-	保留
14	XTH_stable	R	0	外部高速时钟 XTH 稳定标志位： 1：代表 XTH 已经稳定，可以被内部电路使用 0：代表 XTH 未稳定，不可以被内部电路使用
13:12	XTH_startup	R/W	2’b10	外部高速时钟 XTH 稳定时间选择： 11：65535 个周期 10：32768 个周期 01：16384 个周期 00：4096 个周期
11	RSV	-	-	保留
10	RCH_stable	R	1	内部高速时钟 RCH 稳定标志位： 1：代表 RCH 已经稳定，可以被内部电路使用 0：代表 RCH 未稳定，不可以被内部电路使用
9:8	RCH_startup	R/W	00	内部高速时钟 RCH 稳定时间选择： 11：256 个周期 10：64 个周期 01：16 个周期 00：4 个周期
7	RSV	-	-	保留
6	XTL_stable	R	0	外部低速时钟 XTL 稳定标志位： 1：代表 XTL 已经稳定，可以被内部电路使用 0：代表 XTL 未稳定，不可以被内部电路使用

比特	名称	属性	复位值	描述
5:4	XTL_startup	R/W	2'b10	外部低速时钟 XTL 稳定时间选择： 11：32767 个周期 10：4096 个周期 01：1024 个周期 00：256 个周期
3	RSV	-	-	保留
2	RCL_stable	R	1	内部低速时钟 RCL 稳定标志位： 1：代表 RCL 已经稳定，可以被内部电路使用 0：代表 RCL 未稳定，不可以被内部电路使用
1:0	RCL_startup	R/W	00	内部低速时钟 RCL 稳定时间选择： 11：256 个周期 10：64 个周期 01：16 个周期 00：4 个周期

5.5.5 外围模块时钟寄存器 PERI_CLKEN（偏移：010h）

比特	名称	属性	复位值	描述
31	DIV_CLKEN	R/W	0	DIV 模块时钟使能： 1：使能 0：关闭
30	UART2_CLKEN	R/W	0	UART2 模块时钟使能： 1：使能 0：关闭
29	LIN_CLKEN	R/W	0	LIN 模块时钟使能： 1：使能 0：关闭
28	LPUART1_CLKEN	R/W	0	LPUART1 模块时钟使能： 1：使能 0：关闭
27	ATIMER_CLKEN	R/W	0	ATIMER 模块时钟使能： 1：使能 0：关闭
26	AES_CLKEN	R/W	0	AES 模块时钟使能： 1：使能 0：关闭
25	WWDT_CLKEN	R/W	0	WWDT 模块时钟使能： 1：使能 0：关闭
24	CAN_CLKEN	R/W	0	CAN 模块时钟使能： 1：使能 0：关闭
23	RSV	-	-	保留
22	DMA_CLKEN	R/W	0	DMA 模块时钟使能： 1：使能 0：关闭

比特	名称	属性	复位值	描述
21	SPI1_CLKEN	R/W	0	SPI1 模块时钟使能： 1：使能 0：关闭
20	BTIMER23_CLKEN	R/W	0	BTIMER2、BTIMER3 模块时钟使能： 1：使能 0：关闭
19	GPIOD_CLKEN	R/W	0	GPIOD 模块时钟使能： 1：使能 0：关闭
18	GPIOC_CLKEN	R/W	0	GPIOC 模块时钟使能： 1：使能 0：关闭
17	GPIOB_CLKEN	R/W	0	GPIOB 模块时钟使能： 1：使能 0：关闭
16	GPIOA_CLKEN	R/W	0	GPIOA 模块时钟使能： 1：使能 0：关闭
15	I2C_CLKEN	R/W	0	I2C 模块时钟使能： 1：使能 0：关闭
14	ADC_CLKEN	R/W	0	ADC 控制器模块时钟使能： 1：使能 0：关闭
13	RTC_CLKEN	R/W	1	RTC 模块时钟使能： 1：使能 0：关闭
12	WDT_CLKEN	R/W	0	WDT 模块时钟使能： 1：使能 0：关闭
11	CRC_CLKEN	R/W	0	CRC 模块时钟使能： 1：使能 0：关闭
10	UART1_CLKEN	R/W	0	UART1 模块时钟使能： 1：使能 0：关闭
9	GTIMER0_CLKEN	R/W	0	GTIMER0 模块时钟使能： 1：使能 0：关闭
8	LPTIMER01_CLKEN	R/W	0	LPTIMER0、LPTIMER1 模块时钟使能： 1：使能 0：关闭
7	RSV	-	-	保留
6	BTIMER01_CLKEN	R/W	0	BTIMER0、BTIMER1 模块时钟使能 1：使能 0：关闭
5	RSV	-	-	保留

比特	名称	属性	复位值	描述
4	SPI0_CLKEN	R/W	0	SPI0 模块时钟使能： 1：使能 0：关闭
3	GTIMER2_CLKEN	R/W	0	GTIMER2 模块时钟使能： 1：使能 0：关闭
2	GTIMER1_CLKEN	R/W	0	GTIMER1 模块时钟使能： 1：使能 0：关闭
1	LPUART0_CLKEN	R/W	0	LPUART0 模块时钟使能： 1：使能 0：关闭
0	UART0_CLKEN	R/W	0	UART0 模块时钟使能： 1：使能 0：关闭

5.5.6 复位标识寄存器 RESET_FLAG（偏移：020h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	SYS_RESET_REQ_FLAG	R/W1C	0	CPU 复位状态。需要软件初始化和清除： 1：Cortex M0+ 系统复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0
6	LOCKUP_RSTN_FLAG	R/W1C	0	CPU 死锁复位状态。需要软件初始化和清除： 1：Lockup 复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0
5	LVD_RSTN_FLAG	R/W1C	0	低电压复位状态。需要软件初始化和清除： 1：LVD 复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0
4	SOFT_RSTN_FLAG	R/W1C	0	软件复位状态。需要软件初始化和清除： 1：写 REMAP_ADDR 寄存器的 SOFT_RESETN 位复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0
3	WDT_FLAG	R/W1C	0	看门狗复位状态。需要软件初始化和清除： 1：WDT 复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0

比特	名称	属性	复位值	描述
2	RESETN_FLAG	R/W1C	0	外部复位状态。需要软件初始化和清除。 1：外部复位发生 0：无复位发生 此位只能被 PORN 复位，写 1 清 0
1:0	RSV	-	-	保留

Note：复位标识寄存器只能被 PORN 复位

5.5.7 外围模块复位控制寄存器 PERI_RESET（偏移：024h）

比特	名称	属性	复位值	描述
31	DIV_RESET	R/W	0	DIV 模块复位使能： 1：正常工作 0：模块处于复位状态
30	UART2_RESET	R/W	0	UART2 模块复位使能： 1：正常工作 0：模块处于复位状态
29	LIN_RESET	R/W	0	LIN 模块复位使能： 1：使能 0：关闭
28	LPUART1_RESET	R/W	0	LPUART1 模块复位使能： 1：使能 0：关闭
27	ATIMER_RESET	R/W	0	ATIMER 模块复位使能： 1：正常工作 0：模块处于复位状态
26	AES_RESET	R/W	0	AES 模块复位使能： 1：正常工作 0：模块处于复位状态
25	WWDT_RESET	R/W	0	WWDT 模块复位使能： 1：正常工作 0：模块处于复位状态
24	CAN_RESET	R/W	0	CAN 控制器模块复位使能： 1：正常工作 0：模块处于复位状态
23	RSV	-	-	保留
22	DMA_RESET	R/W	0	DMA 控制器模块复位使能： 1：正常工作 0：模块处于复位状态
21	SPI1_RESET	R/W	0	SPI1 控制器模块复位使能： 1：正常工作 0：模块处于复位状态
20	BTIMER23_RESET	R/W	0	BTIMER2、BTIMER3 模块复位使能： 1：正常工作 0：模块处于复位状态
19	GPIOD_RESET	R/W	0	GPIOD 模块复位使能： 1：正常工作 0：模块处于复位状态

比特	名称	属性	复位值	描述
18	GPIOC_RESET	R/W	0	GPIOC 模块复位使能： 1：正常工作 0：模块处于复位状态
17	GPIOB_RESET	R/W	0	GPIOB 模块复位使能： 1：正常工作 0：模块处于复位状态
16	GPIOA_RESET	R/W	0	GPIOA 模块复位使能： 1：正常工作 0：模块处于复位状态
15	I2C_RESET	R/W	0	I2C 模块复位使能： 1：正常工作 0：模块处于复位状态
14	ADC_RESET	R/W	0	ADC 控制器模块复位使能： 1：正常工作 0：模块处于复位状态
13	RTC_RESET	RW	1	RTC 模块复位使能： 1：正常工作 0：模块处于复位状态
12	WDT_RESET	R/W	0	WDT 模块复位使能： 1：正常工作 0：模块处于复位状态
11	CRC_RESET	R/W	0	CRC 模块复位使能： 1：正常工作 0：模块处于复位状态
10	UART1_RESET	R/W	0	UART1 模块复位使能： 1：正常工作 0：模块处于复位状态
9	GTIMER0_RESET	R/W	0	GTIMER0 模块复位使能： 1：正常工作 0：模块处于复位状态
8	LPTIMER01_RESET	R/W	0	LPTIMER0、LPTIMER1 模块复位使能： 1：正常工作 0：模块处于复位状态
7	RSV	-	-	保留
6	BTIMER01_RESET	R/W	0	BTIMER0、BTIMER1 模块复位使能： 1：正常工作 0：模块处于复位状态
5	RSV	-	-	保留
4	SPI0_RESET	R/W	0	SPI0 控制器模块复位使能： 1：正常工作 0：模块处于复位状态
3	GTIMER2_RESET	R/W	0	GTIMER2 模块复位使能： 1：正常工作 0：模块处于复位状态
2	GTIMER1_RESET	R/W	0	GTIMER1 模块复位使能： 1：正常工作 0：模块处于复位状态

比特	名称	属性	复位值	描述
1	LPUART_RESET	R/W	0	LPUART 模块复位使能： 1：正常工作 0：模块处于复位状态
0	UART0_RESET	R/W	0	UART0 模块复位使能： 1：正常工作 0：模块处于复位状态

5.5.8 外部复位滤波控制寄存器 EXT_RESETCTRL（偏移：028h）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	EXT_FILTER_EN	R/W	0	外部复位滤波使能位： 1：外部复位滤波使能 0：外部复位滤波禁止

5.5.9 端口 PA 功能配置寄存器 PA_SEL（偏移：030h）

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:24	PA6_SEL	R/W	4’b0000	端口 PA6 功能选择： 4’b0000：GPIO PA6 4’b0001：GTIM2_CH 4’b0010：UART1_TX 4’b0011：SPI0_CSN0 4’b0100：LPUART0_TX 4’b0101：RTC_FOUT 4’b0110：COMP1_OUT 4’b0111：RTC_TAMP1 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：LPUART1_TX 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：ATIM_CH3N 4’b1110：ATIM_BK2 4’b1111：CAN_TX

比特	名称	属性	复位值	描述
23:20	PA5_SEL	R/W	4'b0000	端口 PA5 功能选择： 4'b0000：GPIO PA5 4'b0001：GTIM1_CH 4'b0010：LPUART0_TX 4'b0011：UART1_RTS 4'b0100：SPI0_SCK 4'b0101：LPTIM1_IN 4'b0110：SPI1_CSN1 4'b0111：SPI1_MI1 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LIN_RX 4'b1011：GTIM0_CH 4'b1100：GTIM2_CH 4'b1101：ATIM_CH3 4'b1110：LPTIM0_CAP0 4'b1111：LVD_OUT
19:16	PA4_SEL	R/W	4'b0000	端口 PA4 功能选择： 4'b0000：GPIO PA4 4'b0001：GTIM0_CH 4'b0010：UART1_RX 4'b0011：UART1_CTS 4'b0100：COMP0_OUT 4'b0101：RTC_TAMP0 4'b0110：LPUART0_RX 4'b0111：LPTIM0_IN 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：BTIM1_OUT1 4'b1011：GTIM1_CH 4'b1100：GTIM2_CH 4'b1101：ATIM_CH2N 4'b1110：ATIM_CH4 4'b1111：CAN_RX

比特	名称	属性	复位值	描述
15:12	PA3_SEL	R/W	4'b0000	端口 PA3 功能选择： 4'b0000：GPIO PA3 4'b0001：UART0_TX 4'b0010：I2C_SDA 4'b0011：SPI0_MI1 4'b0100：LPTIM1_OUT0 4'b0101：BTIM1_OUT0 4'b0110：UART1_RX 4'b0111：SPI1_CSN1 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LIN_TX 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH2 4'b1111：ATIM_BK1
11:8	PA2_SEL	R/W	4'b0000	端口 PA2 功能选择： 4'b0000：GPIO PA2 4'b0001：NC 4'b0010：UART1_RX 4'b0011：UART0_RX 4'b0100：LPUART0_RX 4'b0101：I2C_SCL 4'b0110：I2C_SDA 4'b0111：NC
7:4	PA1_SEL	R/W	4'b0000	端口 PA1 功能选择： 4'b0000：GPIO PA1 4'b0001：SPI1_MI1 4'b0010：SPI0_MOSI 4'b0011：LPTIM1_EXT 4'b0100：UART0_RX 4'b0101：GTIM1_CHN 4'b0110：LPUART1_RX 4'b0111：LIN_TX 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：ATIM_CH4 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH1N 4'b1111：ATIM_ETR

比特	名称	属性	复位值	描述
3:0	PA0_SEL	R/W	4'b0000	端口 PA0 功能选择： 4'b0000：GPIO PA0 4'b0001：GTIM2_CHN 4'b0010：RTC_FOUT 4'b0011：SPI0_CSN1 4'b0100：COMP2_OUT 4'b0101：BTIM0_OUT0 4'b0110：UART0_RX 4'b0111：LPUART1_TX 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LIN_RX 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH1 4'b1111：ATIM_BK2

5.5.10 端口 PB 功能配置寄存器 PB_SEL（偏移：034h）

比特	名称	属性	复位值	描述
31:28	PB7_SEL	R/W	4'b0000	端口 PB7 功能选择： 4'b0000：GPIO PB7 4'b0001：SPI0_SCK 4'b0010：LPTIM0_OUT 4'b0011：ATIM_CH1N 4'b0100：RTC_TAMP0 4'b0101：GTIM2_CHN 4'b0110：ATIM_CH4 4'b0111：GTIM2_BK 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LPUART1_RX 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH2N 4'b1111：LIN_TX

比特	名称	属性	复位值	描述
27:24	PB6_SEL	R/W	4'b0000	端口 PB6 功能选择： 4'b0000：GPIO PB6 4'b0001：LPTIM0_IN 4'b0010：SPI1_MOSI 4'b0011：SPI0_CSN1 4'b0100：GTIM0_CHN 4'b0101：RTC_TAMP1 4'b0110：COMP2_OUT 4'b0111：SPI1_SCK 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LPUART1_TX 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH3N 4'b1111：LPTIM1_OUT1
23:20	PB5_SEL	R/W	4'b0000	端口 PB5 功能选择： 4'b0000：GPIO PB5 4'b0001：GTIM2_CH 4'b0010：SPI1_MISO/SPI1_TRI_MO 4'b0011：SPI0_MI1 4'b0100：UART1_RTS 4'b0101：GTIM1_CH 4'b0110：LPTIM1_OUT0 4'b0111：GTIM1_BK 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LIN_TX 4'b1011：GTIM0_CH 4'b1100：ATIM_CH2N 4'b1101：ATIM_BK1 4'b1110：LPTIM0_OUT1 4'b1111：LPTIM1_CAP1

比特	名称	属性	复位值	描述
19:16	PB4_SEL	R/W	4'b0000	端口 PB4 功能选择： 4'b0000：GPIO PB4 4'b0001：SPI0_MOSI 4'b0010：COMP1_OUT 4'b0011：UART1_CTS 4'b0100：SPI1_MOSI 4'b0101：LPTIM0_OUT 4'b0110：CAN_TX 4'b0111：COMP3_OUT 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：ATIM_ETR 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_CH1N 4'b1111：LIN_RX
15:12	PB3_SEL	R/W	4'b0000	端口 PB3 功能选择： 4'b0000：GPIO PB3 4'b0001：SPI1_MISO/SPI1_TRI_MO 4'b0010：COMP0_OUT 4'b0011：LPTIM0_EXT 4'b0100：CAN_RX 4'b0101：RTC_TAMP1 4'b0110：ATIM_CH3 4'b0111：GTIM0_BK 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：COMP3_OUT 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_BK1 4'b1111：LIN_RX

比特	名称	属性	复位值	描述
11:8	PB2_SEL	R/W	4'b0000	端口 PB2 功能选择： 4'b0000：GPIO PB2 4'b0001：SPI1_SCK 4'b0010：SPI0_CSN0 4'b0011：GTIM0_CH 4'b0100：SPI0_MOSI 4'b0101：LPTIM1_IN 4'b0110：GTIM2_CHN 4'b0111：ATIM_CH1 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LPTIM0_CAP1 4'b1011：GTIM1_CH 4'b1100：GTIM2_CH 4'b1101：ATIM_CH4 4'b1110：LVD_OUT 4'b1111：BTIM0_OUT1
7:4	PB1_SEL	R/W	4'b0000	端口 PB1 功能选择： 4'b0000：GPIO PB1 4'b0001：SPI1_CSN0 4'b0010：GTIM1_CHN 4'b0011：LPTIM0_EXT 4'b0100：LPTIM0_IN 4'b0101：LPUART0_TX 4'b0110：I2C_SCL 4'b0111：COMP1_OUT 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：LPTIM0_OUT1 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：LPTIM1_CAP0 4'b1111：ATIM_CH2

比特	名称	属性	复位值	描述
3:0	PB0_SEL	R/W	4’b0000	端口 PB0 功能选择： 4’b0000：GPIO PB0 4’b0001：GTIM0_CHN 4’b0010：GTIM1_CH 4’b0011：UART1_RX 4’b0100：BUZZER_OUT 4’b0101：SPI1_MOSI 4’b0110：SPI0_MISO/SPI0_TRI_MO 4’b0111：LPUART_RX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM0_OUT0 4’b1011：GTIM0_CH 4’b1100：GTIM2_CH 4’b1101：LPTIM1_OUT1 4’b1110：LPUART1_RX 4’b1111：ATIM_CH1N

5.5.11 端口 PC 功能配置寄存器 PC_SEL（偏移：038h）

比特	名称	属性	复位值	描述
31:28	RSV	-	-	保留
27:24	PC6_SEL	R/W	4’b001	端口 PC6 功能选择： 4’b0000：GPIO PC6 4’b0001：SWCLK 4’b0010：UART1_TX 4’b0011：SPI1_MISO/SPI1_TRI_MO 4’b0100：COMP1_OUT 4’b0101：LPUART0_TX 4’b0110：LPTIM0_OUT0 4’b0111：UART2_TX 4’b1000：UART2_RX 4’b1001：NA 4’b1010：GTIM0_CH 4’b1011：GTIM1_CH 4’b1100：GTIM2_CH

比特	名称	属性	复位值	描述
23:20	PC5_SEL	R/W	4'b001	端口 PC5 功能选择： 4'b0000：GPIO PC5 4'b0001：SWIO 4'b0010：SPI1_SCK 4'b0011：LPTIM0_EXT 4'b0100：I2C_SDA 4'b0101：COMP0_OUT 4'b0110：LPUART0_RX 4'b0111：UART2_TX 4'b1000：UART2_RX 4'b1001：NA 4'b1010：GTIM0_CH 4'b1011：GTIM1_CH 4'b1100：GTIM2_CH
19:16	PC4_SEL	R/W	4'b000	端口 PC4 功能选择： 4'b0000：GPIO PC4 4'b0001：UART1_RTS 4'b0010：SPI1_MOSI 4'b0011：UART0_RX 4'b0100：SPI0_MI1 4'b0101：COMP1_OUT 4'b0110：ATIM_CH3N 4'b0111：LPTIM0_EXT 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：BTIM1_OUT0 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：GTIM2_CH 4'b1110：ATIM_ETR 4'b1111：LPTIM1_OUT1
15:12	PC3_SEL	R/W	4'b000	端口 PC3 功能选择： 4'b0000：GPIO PC3 4'b0001：COMP0_OUT 4'b0010：UART1_CTS 4'b0011：BUZZER_OUT 4'b0100：SPI1_MISO/SPI1_TRI_MO 4'b0101：GTIM2_CH 4'b0110：UART0_TX 4'b0111：LPTIM0_OUT0 4'b1000：UART2_TX 4'b1001：UART2_RX 4'b1010：BTIM0_OUT1 4'b1011：GTIM0_CH 4'b1100：GTIM1_CH 4'b1101：ATIM_CH4 4'b1110：ATIM_CH1 4'b1111：LPTIM1_CAP1

比特	名称	属性	复位值	描述
11:8	PC2_SEL	R/W	4’b000	端口 PC2 功能选择： 4’b0000：GPIO PC2 4’b0001：I2C_SDA 4’b0010：UART1_RX 4’b0011：COMP0_OUT 4’b0100：SPI0_CSN1 4’b0101：GTIM2_CH 4’b0110：LPTIM1_IN 4’b0111：CLKOUT 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：LPTIM0_OUT1 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：ATIM_CH3 4’b1110：LPTIM0_CAP1 4’b1111：LVD_OUT
7:4	PC1_SEL	R/W	4’b000	端口 PC1 功能选择： 4’b0000：GPIO PC1 4’b0001：I2C_SCL 4’b0010：UART1_TX 4’b0011：COMP0_OUT 4’b0100：SPI0_MISO/SPI0_TRI_MO 4’b0101：GTIM1_CH 4’b0110：LPTIM0_OUT0 4’b0111：CAN_RX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：ATIM_CH3N 4’b1011：GTIM0_CH 4’b1100：GTIM2_CH 4’b1101：ATIM_CH2 4’b1110：LPTIM0_OUT1 4’b1111：LPTIM1_CAP0

比特	名称	属性	复位值	描述
3:0	PC0_SEL	R/W	4’b000	端口 PC0 功能选择： 4’b0000：GPIO PC0 4’b0001：SPI0_MOSI 4’b0010：GTIM0_CH 4’b0011：LPTIM0_IN 4’b0100：ATIM_CH2 4’b0101：CAN_TX 4’b0110：SPI1_MI1 4’b0111：GTIM0_BK 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：LPTIM1_CAP1 4’b1011：GTIM1_CH 4’b1100：GTIM2_CH 4’b1101：ATIM_CH1 4’b1110：LPTIM1_EXT 4’b1111：ATIM_CH3

5.5.12 端口 PD 功能配置寄存器 PD_SEL（偏移：03Ch）

比特	名称	属性	复位值	描述
31:28	PD7_SEL	R/W	4’b000	端口 PD7 功能选择： 4’b0000：GPIO PD7 4’b0001：UART1_TX 4’b0010：SPI1_CSN0 4’b0011：I2C_SCL 4’b0100：SPI0_SCK 4’b0101：GTIM1_CHN 4’b0110：LPTIM1_OUT0 4’b0111：UART0_RX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM1_OUT1 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：GTIM2_CH 4’b1110：ATIM_ETR 4’b1111：LPTIM0_CAP0

比特	名称	属性	复位值	描述
27:24	PD6_SEL	R/W	4’b000	端口 PD6 功能选择： 4’b0000：GPIO PD6 4’b0001：UART0_TX 4’b0010：SPI0_MISO/SPI0_TRI_MO 4’b0011：LPTIM1_EXT 4’b0100：CAN_TX 4’b0101：ATIM_CH1N 4’b0110：SPI0_CSN0 4’b0111：LIN_TX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM1_OUT0 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：GTIM2_CH 4’b1110：ATIM_CH3 4’b1111：ATIM_BK1
23:20	PD5_SEL	R/W	4’b000	端口 PD5 功能选择： 4’b0000：GPIO PD5 4’b0001：I2C_SDA 4’b0010：LPTIM1_IN 4’b0011：UART1_RX 4’b0100：SPI1_MI1 4’b0101：GTIM0_CHN 4’b0110：CAN_RX 4’b0111：LPUART0_RX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：ATIM_CH1N 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：GTIM2_CH 4’b1110：ATIM_CH2 4’b1111：ATIM_BK2

比特	名称	属性	复位值	描述
19:16	PD4_SEL	R/W	4’b000	端口 PD4 功能选择： 4’b0000：GPIO PD4 4’b0001：UART1_TX 4’b0010：I2C_SCL 4’b0011：LPUART0_TX 4’b0100：SPI1_CSN1 4’b0101：SPI0_SCK 4’b0110：GTIM2_CH 4’b0111：LPTIM0_EXT 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM0_OUT1 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：GTIM2_CHN 4’b1110：ATIM_CH1 4’b1111：ATIM_CH2N
15:12	PD3_SEL	R/W	4’b000	端口 PD3 功能选择： 4’b0000：GPIO PD3 4’b0001：SPI1_MOSI 4’b0010：LPTIM0_IN 4’b0011：GTIM0_CH 4’b0100：SPI0_CSN1 4’b0101；RTC_TAMP1 4’b0110：LPUART1_RX 4’b0111：CAN_TX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：GTIM0_CHN 4’b1011：GTIM1_CH 4’b1100：GTIM1_CHN 4’b1101：GTIM2_CH 4’b1110：ATIM_BK1 4’b1111：LPTIM1_OUT1

比特	名称	属性	复位值	描述
11:8	PD2_SEL	R/W	4’b000	端口 PD2 功能选择： 4’b0000：GPIO PD2 4’b0001：SPI1_MISO/SPI1_TRI_MO 4’b0010：SPI0_MI1 4’b0011：LPUART1_TX 4’b0100：SPI0_CSN0 4’b0101：LPTIM1_OUT 4’b0110：COMP2_OUT 4’b0111：GTIM1_BK 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM0_OUT0 4’b1011：GTIM0_CH 4’b1100：GTIM1_CH 4’b1101：GTIM2_CH 4’b1110：ATIM_ETR 4’b1111：ATIM_CH3N
7:4	PD1_SEL	R/W	4’b000	端口 PD1 功能选择： 4’b0000：GPIO PD1 4’b0001：SPI1_SCK 4’b0010：GTIM1_CH 4’b0011：LPTIM1_EXT 4’b0100：SPI1_MI1 4’b0101：LIN_TX 4’b0110：I2C_SCL 4’b0111：GTIM2_BK 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：ATIM_CH1N 4’b1011：GTIM0_CH 4’b1100：GTIM2_CH 4’b1101：ATIM_BK2 4’b1110：ATIM_CH2N 4’b1111：LVD_OUT

比特	名称	属性	复位值	描述
3:0	PD0_SEL	R/W	0	端口 PD0 功能选择： 4’b0000：GPIO PD0 4’b0001：SPI1_CSN0 4’b0010：GTIM0_CH 4’b0011：UART1_RX 4’b0100：LPTIM1_IN 4’b0101：RTC_TAMP0 4’b0110：GTIM2_CHN 4’b0111：LIN_RX 4’b1000：UART2_TX 4’b1001：UART2_RX 4’b1010：BTIM1_OUT1 4’b1011：LPTIM0_OUT1 4’b1100：GTIM1_CH 4’b1101：GTIM2_CH 4’b1110：ATIM_CH3 4’b1111：LPTIM1_CAP0

5.5.13 端口数模配置寄存器 PAD_ADS（偏移：054h）

比特	名称	属性	复位值	描述
31	PD7_ADS	R/W	0	端口 PD7 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
30	PD6_ADS	R/W	0	端口 PD6 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
29	PD5_ADS	R/W	0	端口 PD5 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
28	PD4_ADS	R/W	0	端口 PD4 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
27	PD3_ADS	R/W	0	端口 PD3 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
26	PD2_ADS	R/W	0	端口 PD2 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
25	PD1_ADS	R/W	0	端口 PD1 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
24	PD0_ADS	R/W	0	端口 PD0 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
23:21	RSV	-	-	保留

比特	名称	属性	复位值	描述
20	PC4_ADS	R/W	0	端口 PC4 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
19	PC3_ADS	R/W	0	端口 PC3 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
18	PC2_ADS	R/W	0	端口 PC2 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
17	PC1_ADS	R/W	0	端口 PC1 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
16	PC0_ADS	R/W	0	端口 PC0 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
15	PB7_ADS	R/W	0	端口 PB7 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
14	PB6_ADS	R/W	0	端口 PB6 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
13	PB5_ADS	R/W	0	端口 PB5 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
12	PB4_ADS	R/W	0	端口 PB4 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
11	PB3_ADS	R/W	0	端口 PB3 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
10	PB2_ADS	R/W	0	端口 PB2 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
9	PB1_ADS	R/W	0	端口 PB1 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
8	PB0_ADS	R/W	0	端口 PB0 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
7	RSV	-	-	保留
6	PA6_ADS	R/W	0	端口 PA6 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
5	PA5_ADS	R/W	0	端口 PA5 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
4	PA4_ADS	R/W	0	端口 PA4 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口

比特	名称	属性	复位值	描述
3	PA3_ADS	RW	0	端口 PA3 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
2	RSV	-	-	保留
1	PA1_ADS	RW	0	端口 PA1 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口
0	PA0_ADS	RW	0	端口 PA0 数模配置寄存器： 1：配置为模拟接口 0：配置为数字接口

5.5.14 端口驱动能力配置寄存器 PAD_DR（偏移：060h）

比特	名称	属性	复位值	描述
31	PD7_DR	R/W	0	端口 PD7 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
30	PD6_DR	R/W	0	端口 PD6 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
29	PD5_DR	R/W	0	端口 PD5 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
28	PD4_DR	R/W	0	端口 PD4 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
27	PD3_DR	R/W	0	端口 PD3 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
26	PD2_DR	R/W	0	端口 PD2 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
25	PD1_DR	R/W	0	端口 PD1 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
24	PD0_DR	R/W	0	端口 PD0 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
23	RSV	-	-	保留
22	PC6_DR	R/W	0	端口 PC6 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
21	PC5_DR	R/W	0	端口 PC5 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
20	PC4_DR	R/W	0	端口 PC4 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力

比特	名称	属性	复位值	描述
19	PC3_DR	R/W	0	端口 PC3 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
18	PC2_DR	R/W	0	端口 PC2 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
17	PC1_DR	R/W	0	端口 PC1 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
16	PC0_DR	R/W	0	端口 PC0 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
15	PB7_DR	R/W	0	端口 PB7 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
14	PB6_DR	R/W	0	端口 PB6 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
13	PB5_DR	R/W	0	端口 PB5 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
12	PB4_DR	R/W	0	端口 PB4 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
11	PB3_DR	R/W	0	端口 PB3 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
10	PB2_DR	R/W	0	端口 PB2 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
9	PB1_DR	R/W	0	端口 PB1 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
8	PB0_DR	R/W	0	端口 PB0 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
7	RSV	-	-	保留
6	PA6_DR	R/W	0	端口 PA6 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
5	PA5_DR	R/W	0	端口 PA5 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
4	PA4_DR	R/W	0	端口 PA4 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
3	PA3_DR	R/W	0	端口 PA3 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力

比特	名称	属性	复位值	描述
2	PA2_DR	R/W	0	端口 PA2 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
1	PA1_DR	R/W	0	端口 PA1 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力
0	PA0_DR	R/W	0	端口 PA0 驱动能力配置寄存器： 1：低驱动能力 0：高驱动能力

5.5.15 端口上拉配置寄存器 PAD_PU（偏移：06Ch）

比特	名称	属性	复位值	描述
31	PD7_PU	R/W	0	端口 PD7 上拉配置寄存器： 1：使能 0：禁止
30	PD6_PU	R/W	0	端口 PD6 上拉配置寄存器： 1：使能 0：禁止
29	PD5_PU	RW/	0	端口 PD5 上拉配置寄存器： 1：使能 0：禁止
28	PD4_PU	R/W	0	端口 PD4 上拉配置寄存器： 1：使能 0：禁止
27	PD3_PU	R/W	0	端口 PD3 上拉配置寄存器： 1：使能 0：禁止
26	PD2_PU	R/W	0	端口 PD2 上拉配置寄存器： 1：使能 0：禁止
25	PD1_PU	R/W	0	端口 PD1 上拉配置寄存器： 1：使能 0：禁止
24	PD0_PU	R/W	0	端口 PD0 上拉配置寄存器： 1：使能 0：禁止
23	RSV	-	-	保留
22	PC6_PU	R/W	1	端口 PC6 上拉配置寄存器： 1：使能 0：禁止
21	PC5_PU	R/W	1	端口 PC5 上拉配置寄存器： 1：使能 0：禁止
20	PC4_PU	R/W	0	端口 PC4 上拉配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
19	PC3_PU	R/W	0	端口 PC3 上拉配置寄存器： 1：使能 0：禁止
18	PC2_PU	R/W	0	端口 PC2 上拉配置寄存器： 1：使能 0：禁止
17	PC1_PU	R/W	0	端口 PC1 上拉配置寄存器： 1：使能 0：禁止
16	PC0_PU	R/W	0	端口 PC0 上拉配置寄存器： 1：使能 0：禁止
15	PB7_PU	R/W	0	端口 PB7 上拉配置寄存器： 1：使能 0：禁止
14	PB6_PU	R/W	0	端口 PB6 上拉配置寄存器： 1：使能 0：禁止
13	PB5_PU	R/W	0	端口 PB5 上拉配置寄存器： 1：使能 0：禁止
12	PB4_PU	R/W	0	端口 PB4 上拉配置寄存器： 1：使能 0：禁止
11	PB3_PU	R/W	0	端口 PB3 上拉配置寄存器： 1：使能 0：禁止
10	PB2_PU	R/W	0	端口 PB2 上拉配置寄存器： 1：使能 0：禁止
9	PB1_PU	R/W	0	端口 PB1 上拉配置寄存器： 1：使能 0：禁止
8	PB0_PU	RW	0	端口 PB0 上拉配置寄存器： 1：使能 0：禁止
7	RSV	-	-	保留
6	PA6_PU	R/W	0	端口 PA6 上拉配置寄存器： 1：使能 0：禁止
5	PA5_PU	R/W	0	端口 PA5 上拉配置寄存器： 1：使能 0：禁止
4	PA4_PU	R/W	0	端口 PA4 上拉配置寄存器： 1：使能 0：禁止
3	PA3_PU	R/W	0	端口 PA3 上拉配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
2	PA2_PU	R/W	1	端口 PA2 上拉配置寄存器： 1：使能 0：禁止
1	PA1_PU	R/W	0	端口 PA1 上拉配置寄存器： 1：使能 0：禁止
0	PA0_PU	R/W	0	端口 PA0 上拉配置寄存器： 1：使能 0：禁止

5.5.16 端口下拉配置寄存器 PAD_PD（偏移：078h）

比特	名称	属性	复位值	描述
31	PD7_PD	R/W	0	端口 PD7 下拉配置寄存器： 1：使能 0：禁止
30	PD6_PD	R/W	0	端口 PD6 下拉配置寄存器： 1：使能 0：禁止
29	PD5_PD	R/W	0	端口 PD5 下拉配置寄存器： 1：使能 0：禁止
28	PD4_PD	R/W	0	端口 PD4 下拉配置寄存器： 1：使能 0：禁止
27	PD3_PD	R/W	0	端口 PD3 下拉配置寄存器： 1：使能 0：禁止
26	PD2_PD	R/W	0	端口 PD2 下拉配置寄存器： 1：使能 0：禁止
25	PD1_PD	R/W	0	端口 PD1 下拉配置寄存器： 1：使能 0：禁止
24	PD0_PD	R/W	0	端口 PD0 下拉配置寄存器： 1：使能 0：禁止
23	RSV	-	-	保留
22	PC6_PD	R/W	0	端口 PC6 下拉配置寄存器： 1：使能 0：禁止
21	PC5_PD	R/W	0	端口 PC5 下拉配置寄存器： 1：使能 0：禁止
20	PC4_PD	R/W	0	端口 PC4 下拉配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
19	PC3_PD	R/W	0	端口 PC3 下拉配置寄存器： 1：使能 0：禁止
18	PC2_PD	R/W	0	端口 PC2 下拉配置寄存器： 1：使能 0：禁止
17	PC1_PD	R/W	0	端口 PC1 下拉配置寄存器： 1：使能 0：禁止
16	PC0_PD	R/W	0	端口 PC0 下拉配置寄存器： 1：使能 0：禁止
15	PB7_PD	R/W	0	端口 PB7 下拉配置寄存器： 1：使能 0：禁止
14	PB6_PD	R/W	0	端口 PB6 下拉配置寄存器： 1：使能 0：禁止
13	PB5_PD	R/W	0	端口 PB5 下拉配置寄存器： 1：使能 0：禁止
12	PB4_PD	R/W	0	端口 PB4 下拉配置寄存器： 1：使能 0：禁止
11	PB3_PD	R/W	0	端口 PB3 下拉配置寄存器： 1：使能 0：禁止
10	PB2_PD	R/W	0	端口 PB2 下拉配置寄存器： 1：使能 0：禁止
9	PB1_PD	R/W	0	端口 PB1 下拉配置寄存器： 1：使能 0：禁止
8	PB0_PD	R/W	0	端口 PB0 下拉配置寄存器： 1：使能 0：禁止
7	RSV	-	-	保留
6	PA6_PD	R/W	0	端口 PA6 下拉配置寄存器： 1：使能 0：禁止
5	PA5_PD	R/W	0	端口 PA5 下拉配置寄存器： 1：使能 0：禁止
4	PA4_PD	R/W	0	端口 PA4 下拉配置寄存器： 1：使能 0：禁止
3	PA3_PD	R/W	0	端口 PA3 下拉配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
2	PA2_PD	R/W	0	端口 PA2 下拉配置寄存器： 1：使能 0：禁止
1	PA1_PD	R/W	0	端口 PA1 下拉配置寄存器： 1：使能 0：禁止
0	PA0_PD	R/W	0	端口 PA0 下拉配置寄存器： 1：使能 0：禁止

5.5.17 端口开漏输出配置寄存器 PAD_OD（偏移：084h）

比特	名称	属性	复位值	描述
31	PD7_OD	R/W	0	端口 PD7 开漏输出配置寄存器： 1：使能 0：禁止
30	PD6_OD	R/W	0	端口 PD6 开漏输出配置寄存器： 1：使能 0：禁止
29	PD5_OD	R/W	0	端口 PD5 开漏输出配置寄存器： 1：使能 0：禁止
28	PD4_OD	R/W	0	端口 PD4 开漏输出配置寄存器： 1：使能 0：禁止
27	PD3_OD	R/W	0	端口 PD3 开漏输出配置寄存器： 1：使能 0：禁止
26	PD2_OD	R/W	0	端口 PD2 开漏输出配置寄存器： 1：使能 0：禁止
25	PD1_OD	R/W	0	端口 PD1 开漏输出配置寄存器： 1：使能 0：禁止
24	PD0_OD	R/W	0	端口 PD0 开漏输出配置寄存器： 1：使能 0：禁止
23	RSV	-	-	保留
22	PC6_OD	R/W	0	端口 PC6 开漏输出配置寄存器： 1：使能 0：禁止
21	PC5_OD	R/W	0	端口 PC5 开漏输出配置寄存器： 1：使能 0：禁止
20	PC4_OD	R/W	0	端口 PC4 开漏输出配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
19	PC3_OD	R/W	0	端口 PC3 开漏输出配置寄存器： 1：使能 0：禁止
18	PC2_OD	R/W	0	端口 PC2 开漏输出配置寄存器： 1：使能 0：禁止
17	PC1_OD	R/W	0	端口 PC1 开漏输出配置寄存器： 1：使能 0：禁止
16	PC0_OD	RW	0	端口 PC0 开漏输出配置寄存器： 1：使能 0：禁止
15	PB7_OD	R/W	0	端口 PB7 开漏输出配置寄存器： 1：使能 0：禁止
14	PB6_OD	R/W	0	端口 PB6 开漏输出配置寄存器： 1：使能 0：禁止
13	PB5_OD	R/W	0	端口 PB5 开漏输出配置寄存器： 1：使能 0：禁止
12	PB4_OD	R/W	0	端口 PB4 开漏输出配置寄存器： 1：使能 0：禁止
11	PB3_OD	R/W	0	端口 PB3 开漏输出配置寄存器： 1：使能 0：禁止
10	PB2_OD	R/W	0	端口 PB2 开漏输出配置寄存器： 1：使能 0：禁止
9	PB1_OD	R/W	0	端口 PB1 开漏输出配置寄存器： 1：使能 0：禁止
8	PB0_OD	R/W	0	端口 PB0 开漏输出配置寄存器： 1：使能 0：禁止
7	RSV	-	-	保留
6	PA6_OD	R/W	0	端口 PA6 开漏输出配置寄存器： 1：使能 0：禁止
5	PA5_OD	R/W	0	端口 PA5 开漏输出配置寄存器： 1：使能 0：禁止
4	PA4_OD	R/W	0	端口 PA4 开漏输出配置寄存器： 1：使能 0：禁止
3	PA3_OD	R/W	0	端口 PA3 开漏输出配置寄存器： 1：使能 0：禁止

比特	名称	属性	复位值	描述
2	PA2_OD	R/W	0	端口 PA2 开漏输出配置寄存器： 1：使能 0：禁止
1	PA1_OD	R/W	0	端口 PA1 开漏输出配置寄存器： 1：使能 0：禁止
0	PA0_OD	R/W	0	端口 PA0 开漏输出配置寄存器： 1：使能 0：禁止

5.5.18 端口输入类型配置寄存器 PAD_CS（偏移：090h）

比特	名称	属性	复位值	描述
31	PD7_CS	R/W	1	端口 PD7 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
30	PD6_CS	R/W	1	端口 PD6 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
29	PD5_CS	R/W	1	端口 PD5 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
28	PD4_CS	R/W	1	端口 PD4 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
27	PD3_CS	R/W	1	端口 PD3 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
26	PD2_CS	R/W	1	端口 PD2 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
25	PD1_CS	R/W	1	端口 PD1 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
24	PD0_CS	R/W	1	端口 PD0 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
23	RSV	-	-	保留
22	PC6_CS	R/W	1	端口 PC6 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
21	PC5_CS	R/W	1	端口 PC5 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
20	PC4_CS	R/W	1	端口 PC4 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入

比特	名称	属性	复位值	描述
19	PC3_CS	R/W	1	端口 PC3 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
18	PC2_CS	R/W	1	端口 PC2 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
17	PC1_CS	R/W	1	端口 PC1 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
16	PC0_CS	R/W	1	端口 PC0 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
15	PB7_CS	R/W	1	端口 PB7 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
14	PB6_CS	R/W	1	端口 PB6 输入类型配置寄存器 1：CMOS 输入 0：Schmitt 输入
13	PB5_CS	R/W	1	端口 PB5 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
12	PB4_CS	R/W	1	端口 PB4 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
11	PB3_CS	R/W	1	端口 PB3 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
10	PB2_CS	R/W	1	端口 PB2 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
9	PB1_CS	R/W	1	端口 PB1 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
8	PB0_CS	R/W	1	端口 PB0 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
7	RSV	-	-	保留
6	PA6_CS	R/W	1	端口 PA6 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
5	PA5_CS	R/W	1	端口 PA5 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
4	PA4_CS	R/W	1	端口 PA4 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
3	PA3_CS	R/W	1	端口 PA3 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入

比特	名称	属性	复位值	描述
2	PA2_CS	R/W	1	端口 PA2 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
1	PA1_CS	R/W	1	端口 PA1 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入
0	PA0_CS	R/W	1	端口 PA0 输入类型配置寄存器： 1：CMOS 输入 0：Schmitt 输入

5.5.19 端口输入配置寄存器 PAD_IE（偏移：09Ch）

比特	名称	属性	复位值	描述
31	PD7_IE	R/W	0	端口 PD7 输入配置寄存器： 1：输入使能 0：输入禁止
30	PD6_IE	R/W	0	端口 PD6 输入配置寄存器： 1：输入使能 0：输入禁止
29	PD5_IE	R/W	0	端口 PD5 输入配置寄存器： 1：输入使能 0：输入禁止
28	PD4_IE	R/W	0	端口 PD4 输入配置寄存器： 1：输入使能 0：输入禁止
27	PD3_IE	R/W	0	端口 PD3 输入配置寄存器： 1：输入使能 0：输入禁止
26	PD2_IE	R/W	0	端口 PD2 输入配置寄存器： 1：输入使能 0：输入禁止
25	PD1_IE	R/W	0	端口 PD1 输入配置寄存器： 1：输入使能 0：输入禁止
24	PD0_IE	R/W	0	端口 PD0 输入配置寄存器： 1：输入使能 0：输入禁止
23	RSV	-	-	保留
22	PC6_IE	R/W	1	端口 PC6 输入配置寄存器： 1：输入使能 0：输入禁止
21	PC5_IE	R/W	1	端口 PC5 输入配置寄存器： 1：输入使能 0：输入禁止
20	PC4_IE	R/W	0	端口 PC4 输入配置寄存器： 1：输入使能 0：输入禁止

比特	名称	属性	复位值	描述
19	PC3_IE	R/W	0	端口 PC3 输入配置寄存器： 1：输入使能 0：输入禁止
18	PC2_IE	R/W	0	端口 PC2 输入配置寄存器： 1：输入使能 0：输入禁止
17	PC1_IE	R/W	0	端口 PC1 输入配置寄存器： 1：输入使能 0：输入禁止
16	PC0_IE	R/W	0	端口 PC0 输入配置寄存器： 1：输入使能 0：输入禁止
15	PB7_IE	R/W	0	端口 PB7 输入配置寄存器： 1：输入使能 0：输入禁止
14	PB6_IE	R/W	0	端口 PB6 输入配置寄存器： 1：输入使能 0：输入禁止
13	PB5_IE	R/W	0	端口 PB5 输入配置寄存器： 1：输入使能 0：输入禁止
12	PB4_IE	R/W	0	端口 PB4 输入配置寄存器： 1：输入使能 0：输入禁止
11	PB3_IE	R/W	0	端口 PB3 输入配置寄存器： 1：输入使能 0：输入禁止
10	PB2_IE	R/W	0	端口 PB2 输入配置寄存器： 1：输入使能 0：输入禁止
9	PB1_IE	R/W	0	端口 PB1 输入配置寄存器： 1：输入使能 0：输入禁止
8	PB0_IE	R/W	0	端口 PB0 输入配置寄存器： 1：输入使能 0：输入禁止
7	RSV	-	-	保留
6	PA6_IE	R/W	0	端口 PA6 输入配置寄存器： 1：输入使能 0：输入禁止
5	PA5_IE	R/W	0	端口 PA5 输入配置寄存器： 1：输入使能 0：输入禁止
4	PA4_IE	R/W	0	端口 PA4 输入配置寄存器： 1：输入使能 0：输入禁止
3	PA3_IE	R/W	0	端口 PA3 输入配置寄存器： 1：输入使能 0：输入禁止

比特	名称	属性	复位值	描述
2	PA2_IE	R/W	1	端口 PA2 输入配置寄存器： 1：输入使能 0：输入禁止
1	PA1_IE	R/W	0	端口 PA1 输入配置寄存器： 1：输入使能 0：输入禁止
0	PA0_IE	R/W	0	端口 PA0 输入配置寄存器： 1：输入使能 0：输入禁止

5.5.20 端口输入电平寄存器 PAD_STATUS（偏移：0A4h）

比特	名称	属性	复位值	描述
31	PD7_I	R	0	端口 PD7 输入电平寄存器： 1：输入高电平 0：输入低电平
30	PD6_I	R	0	端口 PD6 输入电平寄存器： 1：输入高电平 0：输入低电平
29	PD5_I	R	0	端口 PD5 输入电平寄存器： 1：输入高电平 0：输入低电平
28	PD4_I	R	0	端口 PD4 输入电平寄存器： 1：输入高电平 0：输入低电平
27	PD3_I	R	0	端口 PD3 输入电平寄存器： 1：输入高电平 0：输入低电平
26	PD2_I	R	0	端口 PD2 输入电平寄存器： 1：输入高电平 0：输入低电平
25	PD1_I	R	0	端口 PD1 输入电平寄存器： 1：输入高电平 0：输入低电平
24	PD0_I	R	0	端口 PD0 输入电平寄存器： 1：输入高电平 0：输入低电平
23	RSV	R	0	-
22	PC6_I	R	1	端口 PC6 输入电平寄存器： 1：输入高电平 0：输入低电平
21	PC5_I	R	1	端口 PC5 输入电平寄存器： 1：输入高电平 0：输入低电平
20	PC4_I	R	0	端口 PC4 输入电平寄存器： 1：输入高电平 0：输入低电平

比特	名称	属性	复位值	描述
19	PC3_I	R	0	端口 PC3 输入电平寄存器： 1：输入高电平 0：输入低电平
18	PC2_I	R	0	端口 PC2 输入电平寄存器： 1：输入高电平 0：输入低电平
17	PC1_I	R	0	端口 PC1 输入电平寄存器： 1：输入高电平 0：输入低电平
16	PC0_I	R	0	端口 PC0 输入电平寄存器： 1：输入高电平 0：输入低电平
15	PB7_I	R	0	端口 PB7 输入电平寄存器： 0：输入低电平 1：输入高电平
14	PB6_I	R	0	端口 PB6 输入电平寄存器： 1：输入高电平 0：输入低电平
13	PB5_I	R	0	端口 PB5 输入电平寄存器： 1：输入高电平 0：输入低电平
12	PB4_I	R	0	端口 PB4 输入电平寄存器： 1：输入高电平 0：输入低电平
11	PB3_I	R	0	端口 PB3 输入电平寄存器： 1：输入高电平 0：输入低电平
10	PB2_I	R	0	端口 PB2 输入电平寄存器： 1：输入高电平 0：输入低电平
9	PB1_I	R	0	端口 PB1 输入电平寄存器： 1：输入高电平 0：输入低电平
8	PB0_I	R	0	端口 PB0 输入电平寄存器： 1：输入高电平 0：输入低电平
7	RSV	R	0	-
6	PA6_I	R	0	端口 PA6 输入电平寄存器： 1：输入高电平 0：输入低电平
5	PA5_I	R	0	端口 PA5 输入电平寄存器： 1：输入高电平 0：输入低电平
4	PA4_I	R	0	端口 PA4 输入电平寄存器： 1：输入高电平 0：输入低电平
3	PA3_I	R	0	端口 PA3 输入电平寄存器： 1：输入高电平 0：输入低电平

比特	名称	属性	复位值	描述
2	PA2_I	R	1	端口 PA2 输入电平寄存器： 1：输入高电平 0：输入低电平
1	PA1_I	R	0	端口 PA1 输入电平寄存器： 1：输入高电平 0：输入低电平
0	PA0_I	R	0	端口 PA0 输入电平寄存器： 1：输入高电平 0：输入低电平

5.5.21 端口速度配置寄存器 PAD_SR（偏移：0A8h）

比特	名称	属性	复位值	描述
31	PD7_SR	R/W	1	端口 PD7 速度配置寄存器： 1：低速 0：高速
30	PD6_SR	R/W	1	端口 PD6 速度配置寄存器： 1：低速 0：高速
29	PD5_SR	R/W	1	端口 PD5 速度配置寄存器： 1：低速 0：高速
28	PD4_SR	R/W	1	端口 PD4 速度配置寄存器： 1：低速 0：高速
27	PD3_SR	R/W	1	端口 PD3 速度配置寄存器： 1：低速 0：高速
26	PD2_SR	R/W	1	端口 PD2 速度配置寄存器： 1：低速 0：高速
25	PD1_SR	R/W	1	端口 PD1 速度配置寄存器： 1：低速 0：高速
24	PD0_SR	R/W	1	端口 PD0 速度配置寄存器： 1：低速 0：高速
23	RSV	-	-	保留
22	PC6_SR	R/W	1	端口 PC6 速度配置寄存器： 1：低速 0：高速
21	PC5_SR	R/W	1	端口 PC5 速度配置寄存器： 1：低速 0：高速
20	PC4_SR	R/W	1	端口 PC4 速度配置寄存器： 1：低速 0：高速

比特	名称	属性	复位值	描述
19	PC3_SR	R/W	1	端口 PC3 速度配置寄存器： 1：低速 0：高速
18	PC2_SR	R/W	1	端口 PC2 速度配置寄存器： 1：低速 0：高速
17	PC1_SR	R/W	1	端口 PC1 速度配置寄存器： 1：低速 0：高速
16	PC0_SR	R/W	1	端口 PC0 速度配置寄存器： 1：低速 0：高速
15	PB7_SR	R/W	1	端口 PB7 速度配置寄存器： 1：低速 0：高速
14	PB6_SR	R/W	1	端口 PB6 速度配置寄存器： 1：低速 0：高速
13	PB5_SR	R/W	1	端口 PB5 速度配置寄存器： 1：低速 0：高速
12	PB4_SR	R/W	1	端口 PB4 速度配置寄存器： 1：低速 0：高速
11	PB3_SR	R/W	1	端口 PB3 速度配置寄存器： 1：低速 0：高速
10	PB2_SR	R/W	1	端口 PB2 速度配置寄存器： 1：低速 0：高速
9	PB1_SR	R/W	1	端口 PB1 速度配置寄存器： 1：低速 0：高速
8	PB0_SR	R/W	1	端口 PB0 速度配置寄存器： 1：低速 0：高速
7	RSV	-	-	保留
6	PA6_SR	R/W	1	端口 PA6 速度配置寄存器： 1：低速 0：高速
5	PA5_SR	R/W	1	端口 PA5 速度配置寄存器： 1：低速 0：高速
4	PA4_SR	R/W	1	端口 PA4 速度配置寄存器： 1：低速 0：高速
3	PA3_SR	R/W	1	端口 PA3 速度配置寄存器： 1：低速 0：高速

比特	名称	属性	复位值	描述
2	PA2_SR	R/W	1	端口 PA2 速度配置寄存器： 1：低速 0：高速
1	PA1_SR	R/W	1	端口 PA1 速度配置寄存器： 1：低速 0：高速
0	PA0_SR	R/W	1	端口 PA0 速度配置寄存器： 1：低速 0：高速

5.5.22 IO 控制保护寄存器 IOCTRL_PROTECT（偏移：0B4h）

比特	名称	属性	复位值	描述
31:0	IOCTRL_PROTECT	RW	0	IO 寄存器 PA_SEL / PB_SEL / PC_SEL / PD_SEL / PAD_ADS/ PAD_DR / PAD_PU / PAD_PD / PAD_OD /PAD_CS/ PAD_IE / PAD_SR 保护的控制寄存器。给此寄存器写 0xA5A5_5A5A，启动这些 IO 寄存器的写使能。配置完 IO 寄存器后，它们的写使能不会自动关闭。可以给此寄存器写其它值，来关闭 IO 寄存器的写使能。 读取此寄存器返回 IO 寄存器的写使能状态。 1：写已经使能 0：写未使能

5.5.23 LVD 配置寄存器 LVD_CFG（偏移：0B8h）

比特	名称	属性	复位值	描述
31:25	RSV	-	-	保留
24	LVD_LVEN	RW	1	LVD 滤波使能位
23:16	LVD_FILTER	RW	8’h20	LVD 滤波配置位： 0：对 LVD 滤除 1 个 32K 系统低速时钟的毛刺； 1：对 LVD 滤除 2 个 32K 系统低速时钟的毛刺； …… 255：对 LVD 滤除 256 个 32K 系统低速时钟毛刺。
15:10	RSV	-	-	保留
9	LVD_INTR_EN	RW	0	LVD 中断使能控制位： 1：使能 LVD 中断 0：不使能 LVD 中断
8	LVD_RESET_EN	RW	0	LVD 复位使能控制位： 1：使能 LVD 复位 0：不使能 LVD 复位
7:1	RSV	-	-	保留

比特	名称	属性	复位值	描述
0	LVD_EN	RW	0	LVD 模块使能寄存器： 1：使能 0：禁止

5.5.24 外部复位端口选择寄存器 EXTRST_SEL（偏移：0D0h）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	RESETN_SEL	R/W	0	外部复位端口选择寄存器。只有该寄存器的[31:16]的高 16 位写 0xA5A5 时才能写这个 bit。 1：外部复位信号无效。该管脚可以作为 PA2 其他功能使用 0：外部复位信号有效

5.5.25 停止模式选择寄存器 STOPMODE_SEL（偏移：0D4h）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	STOPMODE_SEL	R/W	0	停止模式选择寄存器。只有该寄存器的[31:16]的高 16 位写 0xA5A5 时才能写这个 bit 1：STOP 模式有效 0：STOP 模式无效

5.5.26 REMAP 寄存器 REMAP_ADDR（偏移：0D8h）

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	REMAP	R	0	Eflash 地址 remap 标志位： 1：Eflash 地址进行重映射，Main 区启动 0：Eflash 地址没有重映射，Bootloader 启动
1	REMAP_IM	W	1	立即进行 REMAP 操作，但是系统不发生复位： 1：eflash 地址不进行重映射 0：立即进行 eflash 地址重映射
0	SOFT_RESETN	W	1	软复位，当此位写 0 时，会产生一次软件复位，复位 CPU 及 AHB/APB 总线上的所有 IP。并且，eflash 地址重新映射（remap 为 1） 1：系统不进行软复位 0：系统进行软复位

注：REMAP 寄存器只能由 RESETEN 外部复位和 POR 上电复位复位，其他复位不能改变这个寄存器的值。

5.5.27 中断向量地址重映射寄存器 VECTOR_OFFSET（偏移：0DCh）

比特	名称	属性	复位值	描述
31:10	VECTOROFFSET	R/W	0	中断向量重映射功能使能后，中断向量的基地址是本寄存器中的值。例如，想要将中断向量表地址映射到 0x20001000 地址，那么此字段应该写入 0x20001000 的[31:10]位，即 0x80004。
9:1	RSV	-	-	保留
0	VECTOROFFSET_EN	R/W	0	中断向量重映射功能使能： 1：使能中断向量重映射功能 0：不使能中断向量重映射功能

5.5.28 蜂鸣器控制寄存器 BUZZER_CR（偏移：0ECh）

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	BUZZER_EN	R/W	1	蜂鸣器时钟输出使能： 1：使能 0：不使能，信号为 BUZZER_POL
16	BUZZER_POL	R/W	0	蜂鸣器时钟极性选择： 1：反极性（停止的时候为 1） 0：原极性（停止的时候为 0）
15:0	BUZZER_DIV	R/W	0x3B	蜂鸣器时钟分频值：分频数为寄存器值＋1

5.5.29 模拟状态寄存器 ANALOG_STATUS（偏移：0FCh）

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21	COMP3_CST	R	0	COMP3 实时状态寄存器： 1：当前 COMP3 有真事件发生 0：当前 COMP3 没有真事件
20	COMP3_INTR	RW	0	COMP3 中断状态： 1：COMP3 发生中断 0：COMP3 未发生中断 写 1 清 0
19:18	RSV	-	-	保留
17	OPA_CST	R		OPA 实时状态寄存器： 1：当前 OPA 有真事件发生 0：当前 OPA 没有真事件
16	OPA_INTR	RW	0	OPA 中断状态： 1：OPA 发生中断 0：OPA 未发生中断 写 1 清 0
15:14	RSV	-	-	保留

比特	名称	属性	复位值	描述
13	COMP2_CST	R	0	COMP2 实时状态寄存器： 1：当前 COMP2 有真事件发生 0：当前 COMP2 没有真事件
12	COMP2_INTR	R/W	0	COMP2 中断状态： 1：COMP2 发生中断 0：COMP2 未发生中断 写 1 清 0
11:10	RSV	-	-	保留
9	COMP1_CST	R	0	COMP1 实时状态寄存器： 1：当前 COMP1 有真事件发生 0：当前 COMP1 没有真事件
8	COMP1_INTR	R/W	0	COMP1 中断状态： 1：COMP1 发生中断 0：COMP1 未发生中断 写 1 清 0
7:6	RSV	-	-	保留
5	COMP0_CST	R	0	COMP0 实时状态寄存器： 1：当前 COMP0 有真事件发生 0：当前 COMP0 没有真事件
4	COMP0_INTR	R/W	0	COMP0 中断状态： 1：COMP0 发生中断 0：COMP0 未发生中断 写 1 清 0
3:2	RSV	-	-	保留
1	LVD_FLAG	R	0	LVD 当前状态： 1：当前 LVD 检测到电压过低的事件 0：当前 LVD 未检测到电压过低的事件
0	LVD_INTR	R/W	0	LVD 中断状态标志： 1：发生过 LVD 中断 0：未发生过 LVD 中断 写 1 清 0

5.5.30 LDO 低功耗软件控制寄存器 LDO_SOFT（偏移：110h）

比特	名称	属性	复位值	描述
31:0	SOFT_LMP	RW	0	给此寄存器写 0x55AAAA55，此寄存器为 1，启动 LDO 的低功耗使能。此寄存器写其他值，此寄存器为 0，关闭 LDO 的低功耗使能。读取此寄存器返回 LDO 的低功耗使能状态。 1：LDO 低功耗使能 0：LDO 低功耗未使能

5.5.31 FLASH 低功耗软件控制寄存器 PDSTBB_SOFT（偏移：114h）

比特	名称	属性	复位值	描述
31:0	SOFT_DPSTBB	RW	1	此寄存器写 0x55AAAA55，此寄存器为 0，启动 FLASH 的低功耗使能。此寄存器写其他值，此寄存器为 1，关闭 FLASH 的低功耗使能。读取此寄存器返回 FLASH 的低功耗使能状态。 1：FLASH 低功耗未使能 0：FLASH 低功耗已经使能

6 EFC

6.1 概述

芯片上集成了64KB/32KB的eFlash存储器，用于保存芯片所有的关键脱机信息和数据。EFC为eFlash控制器，在CPU的配合下，完成Flash读、写、擦除等操作。

6.2 主要特性

- 支持 eFlash 的读写（8/16/32bit）、sector 擦除和 chip 擦除等操作流程
- 读等待时间可以配置
- 主区有 128 个 sector，每个 512 字节
- 支持擦写保护功能
- 支持自动锁总线功能

6.3 寄存器描述

寄存器基地址：0x0110_0000

表 6-1：EFC 寄存器列表

偏置	名称	描述
0x00	EFC_CTRL	控制寄存器
0x04	EFC_SEC	写擦除操作安全寄存器
0x08	EFC_STATUS	状态寄存器
0x0C	EFC_INTSTATUS	中断状态寄存器
0x10	EFC_INEN	中断使能寄存器
0x14	EFC_HALFUS	时间标尺寄存器
0x54	LVD_VDDSL	LVD档位设置寄存器

6.3.1 控制寄存器 EFC_CTRL (偏移：00h)

比特	名称	属性	默认值	功能描述
31:7	RSV	-	-	保留

比特	名称	属性	默认值	功能描述
6:3	Rd_Wait	RW	1	读等待时间设置位。其为读操作eFlash端口等待的时钟周期数。eFlash要求读等待时间至少为25ns，以满足Eflash读的最大延迟。 0：eFlash端口等待1个系统时钟周期； 1：eFlash端口等待2个系统时钟周期； ….
2	Chip_Erase_Mode	RW	0	Chip Erase Mode模式设置位： 1：Chip Erase Mode模式使能 0：Chip Erase Mode模式禁止
1	Sector_Erase_Mode	RW	0	Sector Erase Mode模式设置位： 1：Sector Erase Mode模式使能 0：Sector Erase Mode模式禁止
0	Write_Mode	RW	0	Write模式设置位： 1：写操作模式使能 0：写操作模式禁止

6.3.2 写擦安全寄存器 EFC_SEC (偏移：04h)

比特	名称	属性	默认值	功能描述
31:0	Write_Lock_Ser	W	0	向eFlash写数据/擦除前，须向此寄存器内写0x55AAAA55值，否则控制器其忽略此次擦写操作。

6.3.3 状态寄存器 EFC_STATUS (偏移：08h)

比特	名称	属性	默认值	功能描述
31:4	RSV	-	-	保留
3	Eeprom0_Lock	RO	0	Eeprom0区是否锁住： 1：Eeprom0区已经锁住 0：Eeprom0区没有锁住
2	Nvr2_Lock	RO	0	Nvr2区是否锁住： 1：Nvr2区已经锁住 0：Nvr2区没有锁住
1	Nvr1_Lock	RO	0/1	Nvr1区是否锁住： 1：Nvr1区已经锁住 0：Nvr1区没有锁住
0	eFlash_Ready	RO	1	eFlash状态指示位。该位反映EFlash工作的状态： 1：eFlash状态空闲 0：eFlash状态忙

6.3.4 中断状态寄存器 EFC_INTSTATUS (偏移：0Ch)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5	Eeprom_Err	R/W	0	1：Eeprom0区发生操作错误。当对应的Eeprom0区Lock住时，对此Eeprom0区域进行写擦操作，或者对Eeprom0区域进行Chip Erase操作时，此位均会置1 0：正常状态 写1清0。
4	Wrong_Prog	R/W	0	1：写或擦除操作有误。当EFC_CTRL寄存器的Chip_Erase_Mode、Sector_Erase_Mode、Write_Mode位有2位或以上为1时并进行擦写操作，此位均会置1； 0：正常状态； 写1清0。
3	Boot_Err	R/W	0	1：Boot区发生操作错误。当对应的Boot区Lock住时，对此Boot区域进行写擦操作，或者对Boot区域进行Chip Erase操作时，此位均会置1； 0：正常状态； 写1清0。
2	Nvr2_Err	R/W	0	1：Nvr2区发生操作错误。当对应的Nvr2区Lock住时，对此Nvr2区域进行写擦操作，或者对Nvr2区域进行Chip Erase操作时，此位均会置1； 0：正常状态； 写1清0。
1	Nvr1_Err	R/W	0	1：Nvr1区发生操作错误。当对应的Nvr1区Lock住时，对此Nvr1区域进行写擦操作，或者对Nvr1区域进行Chip Erase操作时，此位均会置1； 0：正常状态； 写1清0。
0	ErWr_done	R/W	0	写/擦除完成中断状态位： 1：写/擦除完成，写1清除该位。如果中断允许，则产生中断； 0：写/擦除未完成。

6.3.5 中断使能寄存器 EFC_INEN (偏移：10h)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5	Eeprom_Err_En	R/W	0	Eeprom_Err中断使能： 1：Eeprom_Err中断使能 0：Eeprom_Err中断不使能

比特	名称	属性	默认值	功能描述
4	Wrong_Prog_En	R/W	0	Wrong_Prog中断使能： 1：Wrong_Prog中断使能 0：Wrong_Prog中断不使能
3	Boot_Err_En	R/W	0	Boot_Err中断使能： 1：Boot_Err中断使能 0：Boot_Err中断不使能
2	Nvr2_Err_En	R/W	0	Nvr2_Err中断使能： 1：Nvr2_Err中断使能 0：Nvr2_Err中断不使能
1	Nvr1_Err_En	R/W	0	Nvr1_Err中断使能： 1：Nvr1_Err中断使能 0：Nvr1_Err中断不使能
0	Er_done_En	R/W	0	擦除完成中断使能： 1：写/擦除中断使能 0：写/擦除中断不使能

6.3.6 时间标尺寄存器 EFC_HALFUS (偏移：14h)

比特	名称	属性	默认值	功能描述
31:8	RSV	-	-	保留
7:0	Half_Us	R/W	0x1E	对eFlash进行擦写前，需根据系统hclk的时钟频率，设置此寄存器。此寄存器的设置值，应为hclk的时钟频率值减1。 例：当hclk时钟频率为32MHz时，此寄存器设置值为31

6.3.7 LVD 档位设置寄存器 LVD_VDDSL (偏移：54h)

比特	名称	属性	默认值	功能描述
31:3	RSV	-	-	保留
2:0	LVDS	R/W	0	LVD 检测点电压设置： LVDS: LVD point 000: 4.29V 001: 3.87V 010: 3.53V 011: 3.22V 100: 2.97V 101: 2.77V 110: 2.58V 111: 2.42V

6.4 软件流程

6.4.1 Write 操作

eFlash上电稳定后可以执行写操作。写操作之前需要向EFC_SEC寄存器内写0x55AAAA55，否

则EFC忽略此次写操作。

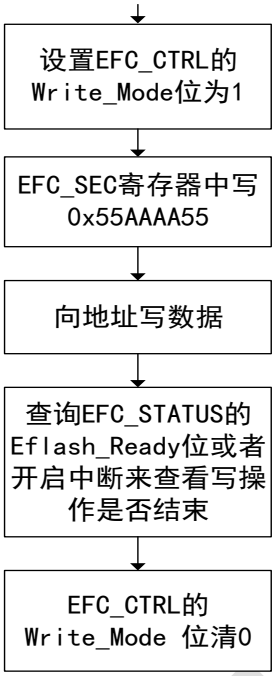


图 6-1：写操作流程

6.4.2 Erase 操作

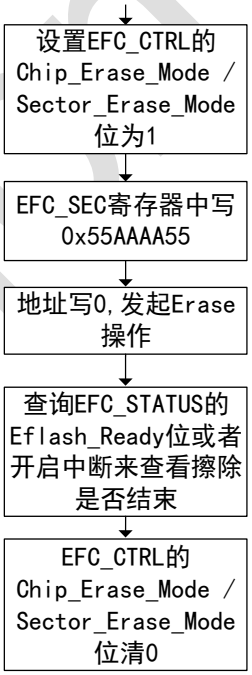


图 6-2：擦除操作流程

7 NVIC

7.1 概述

内嵌套向量中断控制器(NVIC) 是 Cortex-M0+的一个重要组成部分。它与 CPU 处理器内核紧密耦合，实现低中断延迟以及对新到中断的有效处理，外部中断信号连接到 NVIC，NVIC 将对这些中断进行优先级排序。

Cortex-M0+处理器内置了嵌套向量中断控制器（NVIC），可支持最多 32 个中断请求（IRQ）输入：有 4 个中断优先级，可处理复杂逻辑，能进行实时控制和中断处理。

所有的 NVIC 寄存器只能采用字传输。任何试图读/写半字或字节的结果都是不可预知的。

NVIC 寄存器都是小端格式。访问处理器要正确处理处理器的大小端配置。

(关于 NVIC 更详细的内容可查看 Cortex-M0+系列内核的相关官方文档)

7.2 主要特性

- 32 个外部中断，每个中断具有 4 级优先级
- 专用的不可屏蔽中断（NMI）
- 同时支持电平和脉冲中断触发
- 中断唤醒控制器，支持极低功耗休眠模式

7.3 中断源

表 7-1：中断源

中断号	中断源	备注
[0]	GPIO_PA	-
[1]	GPIO_PB	-
[2]	GPIO_PC	-
[3]	GPIO_PD	-
[4]	DMA	-
[5]	LIN	-
[6]	UART0	-
[7]	LPUART0	-
[8]	UART1	-
[9]	I2C	-
[10]	SPI0	-
[11]	SPI1	-
[12]	CAN	-
[13]	DIV	-
[14]	GTIMER0	-
[15]	GTIMER1	-
[16]	GTIMER2	-

中断号	中断源	备注
[17]	LPUART1	-
[18]	LPTIMER	-
[19]	ANALOG	-
[20]	AES	-
[21]	ATIMER	-
[22]	WDT	-
[23]	RTC	-
[24]	ADC	-
[25]	BTIMER01	-
[26]	NA	-
[27]	WWDT	-
[28]	UART2	-
[29]	BTIMER23	-
[30]	EFC	-
[31]	NA	-

8 GPIO

8.1 概述

GPIO 包含通用数据输入输出接口，这些管脚可以与其他功能管脚共享，这取决于芯片的配置。通过这些数据接口，可以配置任意数目的管脚作为中断信号。芯片有 4 组 GPIO，分别是 GPIOA、GPIOB、GPIOC、GPIOD 分别简称为 PA、PB、PC、PD。GPIO 的相关寄存器的功能需要设置对应的比特位，例如设置 PA1 方向为输出，GPIO_DIR 的 bit[1]控制位需要设置为 1，其他位的设置遵循此原则，也即是 PAx 对应寄存器 GPIO_DIR 的 bit[x]控制位。

8.2 主要特性

- 所有输入/输出引脚方向都可以通过软件进行配置
- 每个 GPIO_IN 引脚可配置成边沿或电平方式触发中断

8.3 寄存器描述

GPIOA 寄存器基地址：0x4000_4000

GPIOB 寄存器基地址：0x4000_4400

GPIOC 寄存器基地址：0x4000_4800

GPIOD 寄存器基地址：0x4000_4C00

表 8-1：GPIO 寄存器列表

偏置	名称	描述
0x00	GPIO_DIR	GPIO 数据方向寄存器
0x08	GPIO_SET	GPIO 输出置位寄存器
0x0C	GPIO_CLR	GPIO 输出清零寄存器
0x10	GPIO_ODATA	GPIO 输出引脚映射寄存器
0x14	GPIO_IDATA	GPIO 输入引脚映射寄存器
0x18	GPIO_IEN	GPIO 中断使能寄存器
0x1C	GPIO_IS	GPIO 中断触发模式寄存器
0x20	GPIO_IBE	GPIO 中断边沿触发设置寄存器
0x24	GPIO_IEV	GPIO 中断高低电平触发设置寄存器
0x28	GPIO_IC	GPIO 中断状态清除寄存器
0x2C	GPIO_RIS	GPIO 原始中断状态寄存器
0x30	GPIO_MIS	GPIO 屏蔽后中断状态寄存器

8.3.1 数据方向寄存器 GPIO_DIR(偏移：00h)

比特	名称	属性	复位值	描述
7:0	GPIO_DIR	R/W	0x00	8 位寄存器，GPIO 输入输出控制寄存器： 1：输出 0：输入

注：GPIOx_ DIR[y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ DIR[1]与 PA1 对应。

8.3.2 输出置位寄存器 GPIO_SET(偏移：08h)

比特	名称	属性	复位值	描述
7:0	GPIO_SET	W	0x00	8 位寄存器，GPIO 输出置位寄存器： 1：当 IO 为输出时，IO 置位 0：无效操作

注：GPIOx_ SET [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ SET [1] 与 PA1 对应。

8.3.3 输出清零寄存器 GPIO_CLR(偏移：0Ch)

比特	名称	属性	复位值	描述
7:0	GPIO_CLR	W	0x00	8 位寄存器，GPIO 输出清零寄存器： 1：当 IO 为输出时，IO 清零 0：无效操作

注：GPIOx_ CLR [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ CLR [1] 与 PA1 对应。

8.3.4 GPIO 输出引脚映射寄存器 GPIO_ODATA(偏移：10h)

比特	名称	属性	复位值	描述
7:0	GPIO_ODATA	R/W	0x00	8 位寄存器，GPIO 输出引脚映射寄存器： 当 GPIO 方向为输出有效，写操作直接写至外部引脚，读获得外部引脚值。

注：GPIOx_ ODATA [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ODATA[1] 与 PA1 对应。

8.3.5 GPIO 输入引脚映射寄存器 GPIO_IDATA(偏移：14h)

比特	名称	属性	复位值	描述
7:0	GPIO_IDATA	R	0x00	8 位寄存器，GPIO 输入引脚映射寄存器： 当 GPIO 方向为输入有效，读获得外部引脚值；此寄存器为只读寄存器。

注：GPIOx_ IDATA [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ IDATA [1]与 PA1 对应。

8.3.6 GPIO 中断使能寄存器 GPIO_IEN(偏移：18h)

比特	名称	属性	复位值	描述
7:0	GPIO_IEN	R/W	0x00	8 位寄存器，GPIO 中断使能寄存器： 1：使能相应引脚中断 0：禁止相应引脚中断

注：GPIOx_ IEN [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ IEN [1]与 PA1 对应。

8.3.7 GPIO 中断触发模式寄存器 GPIO_IS(偏移：1Ch)

比特	名称	属性	复位值	描述
7:0	GPIO_IS	R/W	0x00	7 位寄存器，GPIO 中断触发模式： 1：电平检测 0：边沿检测

注：GPIOx_ IS [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ IS [1]与 PA1 对应。

8.3.8 GPIO 中断边沿触发设置寄存器 GPIO_IBE(偏移：20h)

比特	名称	属性	复位值	描述
7:0	GPIO_IBE	R/W	0x00	8 位寄存器，GPIO 中断边沿触发设置： 1：双边沿触发 0：单边沿触发

注：GPIOx_ SET [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ SET [1]与 PA1 对应。

8.3.9 GPIO 中断高低电平触发设置寄存器 GPIO_IEV(偏移：24h)

比特	名称	属性	复位值	描述
7:0	GPIO_IEV	R/W	0x00	8 位寄存器，GPIO 中断高低电平触发设置： 1：上升沿/高电平触发 0：下降沿/低电平触发

注：GPIOx_ IEV [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ IEV [1]与 PA1 对应。

8.3.10 GPIO 中断状态清除寄存器 GPIO_IC(偏移：28h)

比特	名称	属性	复位值	描述
7:0	GPIO_IC	W	0x00	8 位寄存器，GPIO 中断清除寄存器： 1：清除对应引脚中断 0：无效操作

注：GPIOx_IC[y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_IC[1]与 PA1 对应。

8.3.11 GPIO 原始中断状态寄存器 GPIO_RIS(偏移：2Ch)

比特	名称	属性	复位值	描述
7:0	GPIO_RIS	R	0x00	8 位寄存器，GPIO 原始中断寄存器： 1：对应引脚有中断挂起 0：对应引脚无中断挂起

注：GPIOx_ RIS [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_RIS[1]与 PA1 对应。

8.3.12 GPIO 屏蔽后中断状态寄存器 GPIO_MIS(偏移：30h)

比特	名称	属性	复位值	描述
31:0	GPIO_MIS	R	0x00	32 位寄存器，GPIO 屏蔽后中断状态寄存器：反映对应引脚屏蔽后的中断状态。

注：GPIOx_ MIS [y]（x=A…D）寄存器中的 8 位与该组 8 个引脚一一对应，例如 GPIOA_ MIS [1] 与 PA1 对应。

8.4 使用流程

8.4.1 输入输出 IO

1. 配置 GPIO_DIR 寄存器，选择 GPIO 方向为输出。
2. 可使用 GPIO_SET/GPIO_CLR 或 GPIO_ODATA 来设置输出电平。
3. 使用 GPIO_IDATA 来获取输入引脚电平。

8.4.2 中断触发模式

中断初始化过程：

1. 设置 GPIO_DIR 为输入。
2. 清除 GPIO_IEN 以避免异常。
3. 配置寄存器 GPIO_IS，选择是边沿/电平触发类型。
4. 在单边沿触发方式下，配置寄存器 GPIO_IBE，确定是单边触发还是双边触发。

5. 在单边沿触发方式下，配置寄存器 GPIO_IEV，确定是哪种边沿触发类型。
6. 在电平触发方式下，配置寄存器 GPIO_IEV，确定是哪种电平触发类型。
7. 配置寄存器 GPIO_IC 来清除中断。
8. 配置寄存器 GPIO_IEN 使能相应位中断。

8.4.3 清除中断

ISR 写 GPIO_IC 来清除中断状态。如果在清除寄存器的同时有新的边沿触发中断产生，这个新的中断将会保持有效直到下一次清除。读取中断状态操作应该在清 GPIO_IEN 之前进行，清 GPIO_IEN 操作将清除相应中断状态。

9 UART0/2

9.1 概述

UART0和UART2串口模块，带有8比特4级的接收FIFO，支持全双工数据交换，支持与外部接口设备的串行通信。

9.2 主要特性

- 提供标准的异步通讯位（起始位、奇偶位和停止位）
  - 生成 1 位起始位
  - 支持 8bit 的数据位宽
  - 生成 1 位校验位(可设置奇校验或偶校验)，或无校验位
  - 生成 1 位停止位
  - 字节从低位到高位依次传输
- 8 比特 4 级的接收 FIFO，无发送 FIFO
- 可编程波特率(波特率可以根据参数 F/D 调整)，2*8bits 波特率参数寄存器
- 支持数据通讯及错误处理中断
  - 状态位的访问可采用查询或者中断两种方式
  - FIFO 非空、半满、全满、溢出标志
  - 奇偶校验错误标志
- 具有起始位有效性检测功能
- 可支持 9600bps、19200bps、115200bps 等常见波特率的传输

9.3 寄存器描述

UART0 寄存器基地址：0x4000_0000

UART2 寄存器基地址：0x4000_6800

表 9-1：UART0/2 寄存器列表

偏置	名称	描述
0x00	UART_ISR	中断状态寄存器
0x04	UART_IER	中断使能寄存器
0x08	UART_CR	控制寄存器
0x0C	UART_TDR	发送数据寄存器
0x0C	UART_RDR	接收数据寄存器
0x10	UART_BRPL	波特率参数低位寄存器
0x14	UART_BRPH	波特率参数高位寄存器

9.3.1 中断状态寄存器 UART_ISR (偏移：00h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	FIFO_NE	R	0	FIFO 非空标志： 1：FIFO 非空 0：FIFO 空 当 FIFO 读空时，此位自动清 0。软件也可以清除此位，写 0 清除。
4	FIFO_HF	R	0	FIFO 半满标志： 1：FIFO 半满 0：FIFO 非半满 当 FIFO 中数据读空时，此位自动清 0。软件也可以清除此位，写 0 清除。
3	FIFO_FU	R	0	FIFO 全满标志： 1：FIFO 全满 0：FIFO 非全满 当读取 FIFO 中数据，此位自动清 0。软件也可以清除此位，写 0 清除。
2	FIFO_OV	R	0	Rx-FIFO 接收溢出错误： 1：发生了接收溢出错误 0：没有接收溢出错误发生 软件清除此位，写 0 清除。
1	TXEND	R	0	UART 发送完成标志： 1：发送完成 0：发送没有完成 此位硬件置 1，软件清除，写 0 清除。
0	TRE	R	0	UART 发送/接收奇偶校验错误标示： 1：UART 发送/接收完成时有奇偶校验错误 0：UART 发送/接收完成时无奇偶校验错误 此位硬件置 1，软件清除，写 0 清除。

9.3.2 中断使能寄存器 UART_IER (偏移：04h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	FIFO_EN	R/W	0	FIFO 非空中断使能： 1：使能中断 0：禁止中断
4	FIFO_HFEn	R/W	0	FIFO 半满中断使能： 1：使能中断 0：禁止中断
3	FIFO_FUEn	R/W	0	FIFO 全满中断使能： 1：使能中断 0：禁止中断
2	FIFO_OVEn	R/W	0	Rx-FIFO 接收溢出中断使能： 1：使能中断 0：禁止中断

比特	名称	属性	复位值	描述
1	TXENDEn	R/W	0	UART 发送完成中断使能： 1：使能中断 0：禁止中断
0	TREEn	R/W	0	UART 发送/接收奇偶校验错误中断使能： 1：使能中断 0：禁止中断

9.3.3 控制寄存器 UART_CR (偏移：08h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	TX_EN	R/W	0	UART 单线模式使能控制： 1：使能 0：不使能
5	TX_OEN	R/W	0	UART 单线模式 TX 管脚数据传输方向控制： 1：TX 管脚作为数据输入脚 0：TX 管脚作为数据输出脚
4	UART_LB	R/W	0	UART 自测模式使能控制： 1：使能 0：不使能
3	UART_P0	R/W	0	奇偶校验使能控制： 1：没有奇偶校验 0：有奇偶校验
2	FLUSH	R/W	0	清除 UART 接收 FIFO 中的数据和指针： 1：清除 0：不清除
1	TRS	R/W	0	UART 发送数据标志： 1：发送数据使能 0：发送数据不使能
0	ODD_EN	R/W	0	奇偶校验方式选择： 1：奇校验 0：偶校验

9.3.4 发送数据寄存器 UART_TDR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	UARTDATA	W	0	存放待发送的数据

9.3.5 接收数据寄存器 UART_RDR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	UARTDATA	R	0	存放待接收到的数据

9.3.6 波特率参数低位寄存器 UART_BRPL (偏移：10h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	UARTBRPL	R/W	0x74	波特率参数寄存器 UARTBPRH、UARTBPRL 构成 16 位分频器。 例如：系统时钟为 32MHz，为获得 9600 波特率，则 UARTBPR＝32×1000000÷9600＝1388H， 即 UARTBPRH＝DH，UARTBPRL＝05H。

9.3.7 波特率参数高位寄存器 UART_BRPH (偏移：14h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	UARTBRPH	R/W	0x01	波特率参数寄存器 UARTBPRH、UARTBPRL 构成 16 位分频器。

9.4 使用流程

9.4.1 串口的发送和接收

1. 配置系统配置寄存器的串口模块时钟。
2. 配置系统配置寄存器的串口模块复位使能。
3. 配置系统配置寄存器的串口引脚复用功能。
4. 配置串口中断。
5. 配置串口中断使能寄存器(是否使用中断)。
6. 配置串口控制寄存器（清除接收 FIFO 中的数据和指针）。
7. 配置串口控制寄存器（奇偶校验位等）。
8. 配置波特率。
9. 使能串口。

9.4.2 串口初始化

1. 清除 UART_ISR 寄存器，写 0 清除。
2. 配置 UART_IER，中断使能寄存器，是否产生相应的中断脉冲。
3. 设置 UART_CR[2]，清除 FIFO 中数据及 FIFO 指针。
4. 清除 UART_CR 寄存器，写 0 清除。

5. 配置 UART_BPRL[7:0]和 UART_BPRH[7:0]，设置波特率。

9.4.3 串口发送字节

1. 发送、接收数据前软件可以配置波特率参数，奇偶校验类型、中断使能。
2. 设置 UART_CR[1]=1。
3. 写入第一个字节数据到 UART_TDR 寄存器。
4. 查询发送完成标志 UART_ISR[1]，如果 UART_ISR[1]=1 表示当前数据发送完成；软件清除此位（写 0 清除）。
5. 如果发送出错：UART 产生中断或者查询 SCCISR 寄存器标志，判断错误类型，执行相应的错误处理，处理完之后软件清除标志位。
6. 可以继续写入下一个字节到 UART_TDR。

9.4.4 串口接收字节

1. 发送、接收数据前软件可以配置波特率参数、奇偶校验类型、中断使能。
2. 接收数据，查询 UART_ISR 标志位或者等待中断，UART_ISR[5]（即接收数据 FIFO 非空），或者 UART_ISR[4]（即接收数据 FIFO 半满），或者 UART_ISR[3]（即接收数据 FIFO 全满）；查询到相应标志位则读取 UART_RDR 中的数据，FIFO 相应的标志位自动清除。
3. 接收错误处理：等待中断或者查询 UART_ISR 寄存器标志位，判断错误类型，执行相应的错误处理，处理完之后软件清除标志位。
4. 继续接收数据。

10 UART1

10.1 概述

UART1 串口模块，带有 16 字节的 FIFO，可小数分频。

10.2 主要特性

- 16 字节的硬件 FIFO
- 波特率支持整数和小数分频
- 支持 9Bit 模式
- 支持 CTS，RTS 硬件流控制
- 错误起始位侦测
- 帧中断检测
- 可编程位宽，奇偶校验，停止位个数
- 支持 DMA 传输方式

10.3 寄存器描述

UART1 寄存器基地址：0x4000_3000

表 10-1：UART1 寄存器列表

偏置	名称	描述
0x00	UART1_RBR	接收缓冲寄存器
0x00	UART1_THR	发送缓冲寄存器
0x00	UART1_DLL	波特率分频低位寄存器
0x04	UART1_DLH	波特率分频高位寄存器
0x04	UART1_IER	中断使能寄存器
0x08	UART1_IIR	中断状态寄存器
0x08	UART1_FCR	FIFO 控制寄存器
0x0C	UART1_LCR	LINE 控制寄存器
0x10	UART1_MCR	流控制寄存器
0x14	UART1_LSR	LINE 中断状态寄存器
0x18	UART1_MSR	流状态寄存器
0x7C	UART1_USR	状态寄存器
0x80	UART1_TFL	发送 FIFO 数据个数寄存器
0x84	UART1_RFL	接收 FIFO 数据个数寄存器
0xC0	UART1_DLF	小数分频寄存器
0xC4	UART1_RAR	接收地址匹配寄存器
0xC8	UART1_TAR	发送地址匹配寄存器
0xCC	UART1_LCRE	LINE 控制扩展寄存器

10.3.1 接收缓冲寄存器 UART1_RBR (偏移：00h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	RBR	R	0	接收数据寄存器。此字段为接收 FIFO 入口，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。

10.3.2 发送缓冲寄存器 UART1_THR (偏移：00h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	THR	W	0	发送数据寄存器。此字段为发送 FIFO 入口，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。

10.3.3 波特率分频低位寄存器 UART1_DLL (偏移：00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	DLL	R/W	0	波特率配置寄存器低位。仅当 UART1_LCR 的 DLAB 位为 1 时，此字段才可以访问。 波特率整数部分计算公式： baud rate = Fclk / (16 * {DLH, DLL})

10.3.4 波特率分频高位寄存器 UART1_DLH (偏移：04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	DLH	R/W	0	波特率配置寄存器高位。仅当 UART1_LCR 的 DLAB 位为 1 时，此字段才可以访问。 波特率整数部分计算公式： baud rate = Fclk / (16 * {DLH, DLL})

10.3.5 中断使能寄存器 UART1_IER (偏移：04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	PTIME	R/W	0	THRE 中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问： 1：使能 THRE 中断 0：禁止 THRE 中断
6:3	RSV	-	-	保留

比特	名称	属性	复位值	描述
2	ELSI	R/W	0	LINE 中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。 1：使能 LINE 中断 0：禁止 LINE 中断
1	ETBEI	R/W	0	发送 FIFO 空中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。 1：使能发送 FIFO 空中断 0：禁止发送 FIFO 空中断
0	ERBFI	R/W	0	接收数据中断使能，仅当 UART1_LCR 的 DLAB 位为 0 时，此字段才可以访问。 1：使能接收 FIFO 非空中断 0：禁止接收 FIFO 非空中断

10.3.6 中断状态寄存器 UART1_IIR (偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:6	FIFOSE	R	0	FIFO 使能标志： 11：FIFO 使能 00：FIFO 禁止
5:4	RSV	-	-	保留
3:0	IID	R	0x1	状态 ID： 0000：CTS/RTS 中断状态 0001：无中断 0010：发送 FIFO 空 0100：接收 FIFO 非空 0110：LINE 中断状态 0111：Busy 状态 1100：TimeOut 状态，当使能 FIFO 和接收 FIFO 非空中断后，如果在接收 FIFO 中存在至少 1 个数据，在 4 个 UART 帧内，CPU 如果未读 FIFO，则此字段会进入 TimeOut 中断状态 其它：保留

10.3.7 FIFO 控制寄存器 UART1_FCR (偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:6	RT	W	0	接收 FIFO 非空中断设置，当 FIFO 中数据大于或等于此设置对应的 FIFO 状态时，接收 FIFO 非空中断置位： 00：1 帧数据 01：4 帧数据 10：8 帧数据 11：14 帧数据

比特	名称	属性	复位值	描述
5:4	TET	W	0	发送 FIFO 空中断设置，当 FIFO 中数据少于或等于此设置对应的 FIFO 状态时，发送 FIFO 空中断置位： 00：FIFO 空 01：2 帧数据 10：4 帧数据 11：8 帧数据
3	RSV	-	-	保留
2	XFIFOR	W	0	发送 FIFO 复位位： 1：复位发送 FIFO 0：不复位发送 FIFO
1	RFIFOR	W	0	接收 FIFO 复位位： 1：复位接收 FIFO 0：不复位接收 FIFO
0	FIFOE	W	0	FIFO 使能位： 1：使能 FIFO 0：禁止 FIFO 改变此位的值将会同时复位接收和发送 FIFO。

10.3.8 LINE 控制寄存器 UART1_LCR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	DLAB	R/W	0	UART1_DLL 和 UART1_DLH 寄存器访问设置位。 1：UART1_DLL 可以通过偏移地址 0x0 访问，UART1_DLH 可以通过偏移地址 0x4 访问； 0：UART1_RBR/UART1_THR 可以通过偏移地址 0x0 访问，UART1_IER 可以通过偏移地址 0x4 访问。
6	RSV	-	-	保留
5	SEPS	R/W	0	奇偶校验位强制设置位，仅当 UART 处于空闲状态时可写： 1：当 PEN 为 1，EPS 为 1，奇偶校验位被传输并检查为逻辑 0；PEN 为 1，当 EPS 为 0 时，奇偶校验位被传输并检查为逻辑 1；当 PEN 为 0 时，发送和接收均无奇偶校验。 0：奇偶校验位强制设置功能禁止。
4	EPS	R/W	0	奇偶校验选择位，仅当 UART 处于空闲状态时可写： 1：偶校验 0：奇校验
3	PEN	R/W	0	奇偶校验位使能设置，仅当 UART 处于空闲状态时可写： 1：奇偶校验位使能 0：奇偶校验位禁止
2	STOP	R/W	0	STOP 比特长度设置，仅当 UART 处于空闲状态时可写： 1：1.5 比特 STOP 位 0：1 比特 STOP 位

比特	名称	属性	复位值	描述
1:0	DLS	R/W	0	UART 帧数据长度设置位，仅当 UART 处于空闲状态时可写： 00：5 比特 01：6 比特 10：7 比特 11：8 比特

10.3.9 流控制寄存器 UART1_MCR (偏移：10h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5	AFCE	R/W	0	1：CTS/RTS 自动流控制使能 0：CTS/RTS 自动流控制禁止
4:2	RSV	-	-	保留
1	RTS	R/W	0	RTS 接口软件控制位： 1：RTS 请求输出有效； 0：RTS 请求输出无效
0	RSV	-	-	保留

10.3.10 LINE 中断状态寄存器 UART1_LSR (偏移：14h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	ADDR_RCVD	R	0	9 比特数据模式下，接收到的数据是地址还是数据的标志： 1：接收的数据为地址信息 0：接收的数据为数据信息 读取此寄存器，清 0。
7	RFE	R	0	接收 FIFO 错误标志（可触发 LINE 中断）： 1：接收 FIFO 中数据至少有一个有奇偶校验错误或者 UART 帧格式错误 0：接收 FIFO 中数据没有错误 当接收 FIFO 中出错的数据是下一个将要读取的数据，且接收 FIFO 中其它的数据没有错误时，读取此寄存器清 0。
6	TEMT	R	1	发送完成标志： 1：发送完成，发送 FIFO 为空，且移位寄存器为空 0：发送未完成
5	THRE	R	1	发送 FIFO 空标志： 1：发送 FIFO 空 0：发送 FIFO 满
4	RSV	-	-	保留
3	FE	R	0	帧格式出错标志（可触发 LINE 中断）： 1：帧格式错误 0：帧格式未出错 读此寄存器清 0。

比特	名称	属性	复位值	描述
2	PE	R	0	奇偶校验出错标志（可触发 LINE 中断）： 1：奇偶校验错误 0：奇偶校验未出错 读此寄存器清 0。 注：在启用奇偶校验生成和检测 (LCR[3]=1) 并且奇偶校验设置为奇数 (LCR[4]=0)的情况下，若 Break 中断产生，奇偶校验错误中断会置位。
1	OE	R	0	接收 FIFO 溢出标志（可触发 LINE 中断）： 1：接收 FIFO 溢出 0：接收 FIFO 非溢出 读此寄存器清 0。
0	DR	R	0	接收 FIFO 非空标志： 1：接收 FIFO 非空 0：接收 FIFO 空

10.3.11 流状态寄存器 UART1_MSR (偏移：18h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4	CTS	R	0	CTS 标志位： 1：有 CTS 请求 0：无 CTS 请求
3:0	RSV	-	-	保留

10.3.12 状态寄存器 UART1_USR (偏移：7Ch)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4	RFF	R	0	接收 FIFO 满标志： 1：接收 FIFO 满 0：接收 FIFO 非满
3	RFNE	R	0	接收 FIFO 非空标志： 1：接收 FIFO 非空 0：接收 FIFO 空
2	TFE	R	1	发送 FIFO 空标志： 1：发送 FIFO 空 0：发送 FIFO 非空
1	TFNF	R	1	发送 FIFO 非满标志： 1：发送 FIFO 非满 0：发送 FIFO 满
0	BUSY	R	0	1：UART1 正在传输 0：UART1 处于空闲状态

10.3.13 发送 FIFO 数据个数寄存器 UART1_TFL (偏移：80h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	TFL	R	0	发送 FIFO 中数据个数位

10.3.14 接收 FIFO 数据个数寄存器 UART1_RFL (偏移：84h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	RFL	R	0	接收 FIFO 中数据个数位

10.3.15 小数分频寄存器 UART1_DLF(偏移：C0h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3:0	DLF	R/W	0	小数分频寄存器。 小数部分波特率为 DLF/16。 计算公式为：(PCLK%(BAUDRATE*16)) / BAUDRATE。

10.3.16 接收地址匹配寄存器 UART1_RAR(偏移：C4h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	RAR	R/W	0	接收地址匹配寄存器。此寄存器只有在 UART 处于空闲状态时可写。

10.3.17 发送地址匹配寄存器 UART1_TAR (偏移：C8h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	TAR	R/W	0	发送地址匹配寄存器。此寄存器只有在 UART 处于空闲状态时可写。

10.3.18 LINE 控制扩展寄存器 UART1_LCRE (偏移：CCh)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留

比特	名称	属性	复位值	描述
3	TRANSMIT_MODE	R/W	0	9 比特发送模式选择： 1：发送 FIFO 为 9 比特，地址和数据的指示标志来自发送 FIFO 0：发送 FIFO 为 8 比特，发送的地址由 SEND_ADDR 位和 UART1_TAR 来决定，发送的数据来自发送 FIFO
2	SEND_ADDR	R/W	0	发送地址匹配使能位： 1：当 9 比特模式下，UART 帧中的第 9 比特为 1（地址指示），UART 将发送 UART1_TAR 寄存器中的数据 0：当 9 比特模式下，UART 帧中的第 9 比特为 0（数据指示），UART 将发送 FIFO 中的数据
1	ADDR_MATCH	R/W	0	接收数据地址匹配模式使能位： 1：地址匹配模式使能 0：地址匹配模式禁止 此位仅在 DLS_E 为 1 时有效。
0	DLS_E	R/W	0	1：9 比特模式使能 0：帧格式由 DLS 位来决定

10.4 使用流程

10.4.1 UART1 发送流程

1. 设置 UART1_MCR 寄存器。
2. 设置 UART1_LCR[7]为 1。
3. 设置 UART1_DLL/UART1_DLH/UART1_DLF 寄存器。
4. 设置 UART1_LCR[7]为 0，设置 UART1_LCR 寄存器的其它位。
5. 设置 UART1_FCR 寄存器。
6. 设置 UART1_IER 寄存器。
7. 写 UART1_THR 寄存器，向发送 FIFO 中填写数据。
8. 查询 UART1_IIR 中断状态。
9. 完成传输。

10.4.2 UART1 接收流程

1. 设置 UART1_MCR 寄存器。
2. 设置 UART1_LCR[7]为 1。
3. 设置 UART1_DLL/UART1_DLH/UART1_DLF 寄存器。
4. 设置 UART1_LCR[7]为 0，设置 UART1_LCR 寄存器的其它位。
5. 设置 UART1_FCR 寄存器。

6. 设置 UART1_IER 寄存器。
7. 查询 UART1_IIR 中断状态。
8. 读取 UART1_RBR，取走收到的数据。
9. 完成传输。

10.4.3 CTS 和 RTS 控制流功能设置流程

- **CTS**

CTS为UART 输入端口，低电平有效，表示UART可以发送数据。如果CTS输入状态为 1，写UART1_THR 寄存器时，数据只会保存在发送FIFO中不会被发出，为0时开始发送。

CTS配置流程如下：

1. 配置 UART1_CTS 管脚。
2. 配置 UART1_MCR 寄存器，使能 CTS/RTS 自动流控制。
3. 配置 UART1_FCR 寄存器，使能 FIFO。
4. UART1_CTS 管脚输入为低电平时，UART 正常发送数据；输入为高电平时，数据保存在发送 FIFO 中。

- **RTS**

RTS 为UART 输出端口，低电平有效，输出为低电平时，表示UART已经准备好可以接收数据了；当接收FIFO中数据个数大于FIFO控制寄存器UART1_FCR中接收FIFO非空中断设置的帧数据个数，触发中断条件时，RTS输出状态为高电平，表示UART不能接收更多数据。

RTS 配置流程如下：

1. 配置 UART1_RTS 管脚。
2. 配置 UART1_MCR 寄存器，使能 CTS/RTS 自动流控制，RTS 接口软件控制位为请求输出有效。
3. 配置 UART1_FCR 寄存器，使能 FIFO，接收 FIFO 非空中断设置。

10.4.4 UART1 DMA 传输配置流程

UART1 模块可支持 DMA 传输功能，支持三种传输模式：Memory to Peripheral 模式、Peripheral to Memory 模式、Peripheral to Peripheral 模式。配置流程如下：

1. 配置开启 DMA 模块时钟与复位 PERI_RESET / PERI_CLKEN。
2. 配置通道控制信息寄存器 DMA_CHCTRLCx。

3. 根据实际应用配置数据位宽，传输模式（8 位位宽、内存到外设模式）。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_TR_WIDTH_8。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_MEM_TO_ PERIP。
4. 配置【目的外设】和【源外设】（目的外设为 UART1_tx，源外设为 mem）。
   此位在用到外设的模式下起效。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_DST_PER_UART1_TX。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_SRC_PER_MEMORY。
5. 配置【目标地址】和【源地址】是否随数据传输递增（原地址递增、目标地址不变）。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_SINC_INC。
   - 例如：DMA_CHCTRLC(channel_index) |= DMA_DINC_NOC。
6. 如需使用中断，则配置 DMA 中断指示寄存器 DMA_INTSTATUS，使能对应的通道中断。
7. 配置【源地址】和【目标地址】及【数据块尺寸】。
   DMA_SRCADDRCx、DMA_DSTADDRCx、DMA_CHCTRLCx
   - 例如：DMA_SRCADDRC(channel_index) = (uint32_t)src_addr。
   - 例如：DMA_DSTADDRC(channel_index) = (uint32_t)dest_addr。
   - 例如：DMA_CHCTRLC(channel_index) |= (length<<15)。
8. 等待上述配置、以及相应的原地址和目标准备就绪，使能 DMA（DMAC_EN）。
9. 根据实际使用情况，检测 DMA 中断状态寄存器 DMA_INTSTATUS，跟踪传输状态。

10.4.5 UART1 9BIT 模式收发配置流程

1. 设置 PERI_CLKEN 和 PERI_RESET 寄存器，开启和复位 UART1 时钟模块。
2. 配置相应的引脚复用为 UART1_TX 和 UART1_RX。
3. 设置 UART1_MCR 寄存器。
4. 设置 UART1_LCR[7]为 1。
5. 设置 UART1_DLL/UART1_DLH/UART1_DLF 寄存器。
6. 设置 UART1_LCR[7]为 0，设置 UART1_LCR 寄存器的其它位。
7. 设置 UART1_FCR 寄存器。
8. 设置 UART1_LCRE 寄存器设置 9bit 发送或者接收模式配置【开启地址匹配是需要往 UART1_TAR 中填写地址】。
9. 设置 UART1_LCRE[0]为 1，使能 9bit 模式。
10. 设置 UART1_IER 寄存器，设置中断方式。
11. 写 UART1_THR 寄存器，向发送 FIFO 中填写数据。
12. 查询 UART1_IIR 中断状态。
13. 读取 UART1_RBR，取走收到的数据。

14. 完成传输。

11 LPUART0/1

11.1 概述

芯片有两个低功耗串口模块 LPUART0 及 LPUART1，其工作仅需 32kHz 时钟，可以支持到最高 9600 波特率的数据接收。LPUART 功耗极低，可以在 Sleep/DeepSleep 模式下工作。

11.2 主要特性

- 异步数据收发
- 标准 UART 帧格式
  - 1bit 起始位
  - 7 或 8bit 数据
  - 奇校验、偶校验或无校验位
  - 1 或 2bit 停止位
- 使用 32768Hz XTL 时钟或者 32KHz RCL 时钟工作，支持波特率 300 bps~9600bps
- 可编程数据极性
- 支持 Sleep/DeepSleep 模式下的数据收发
- 休眠模式下唤醒芯片
  - RXD 下降沿唤醒
  - 起始位检测唤醒
  - 1 字节接收完成唤醒
  - 1 字节数据匹配唤醒

11.3 寄存器描述

LPUART0 寄存器基地址：0x4000_0400

LPUART1 寄存器基地址：0x4000_7C00

表 11-1：LPUART 寄存器列表

偏置	名称	描述
0x00	LPUART_RXD	接收数据寄存器
0x04	LPUART_TXD	发送数据寄存器
0x08	LPUART_STA	状态寄存器
0x0C	LPUART_CON	控制寄存器
0x10	LPUART_IF	中断标志寄存器
0x14	LPUART_BAUD	波特率寄存器
0x18	LPUART_EN	接收使能寄存器

偏置	名称	描述
0x1C	LPUART_CMPARE	数据匹配寄存器
0x20	LPUART_MCTL	波特率调制控制寄存器
0x24	LPUART_WKCKE	匹配中断唤醒配置寄存器

11.3.1 接收数据寄存器 LPUART_RXD (偏移：00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	LPURXD	R	0	接收数据缓冲

11.3.2 发送数据寄存器 LPUART_TXD (偏移：04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	LPUTXD	W	0	发送数据缓冲

11.3.3 状态寄存器 LPUART_STA (偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TC	R	0	发送完成标志，当一帧数据发送完成且发送 buffer 为空时置位。数据发送时清零。
6	TXE	R	0	发送 buffer 空标志，硬件置位，软件向发送 buffer 写数据时自动清零。
5	START	R/W	0	起始位检测标志，写 1 清零。
4	PERR	R/W	0	校验位错误，写 1 清零。
3	FERR	R/W	0	帧格式错误，写 1 清零。
2	RXOV	R/W	0	接收缓冲溢出，写 1 清零。
1	RXF	R	0	接收缓冲满，读 LPUART_DATA 寄存器清零。
0	MATCH	R/W	0	数据匹配标志，表示接收缓冲区内的数据与比较寄存器相同，写 1 清零。

11.3.4 控制寄存器 LPUART_CON (偏移：0Ch)

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	TXPOL	R/W	0	数据发送极性： 0：非取反 1：取反
11	TCIE	R/W	0	发送完成中断使能： 0：禁止发送完成中断 1：允许发送完成中断
10	TXIE	R/W	0	发送 buffer 空中断使能： 0：禁止发送 buffer 空中断 1：允许发送 buffer 空中断

比特	名称	属性	复位值	描述
9	NEDET	R/W	0	下降沿采样使能位： 0：使用 32k 时钟上升沿检测 start bit 1：使用 32k 时钟下降沿检测 start bit
8	PAREN	R/W	0	校验位使能： 0：数据帧无奇偶校验位 1：数据帧有奇偶校验位
7	PTYP	R/W	0	校验位类型： 0：偶校验 1：奇校验
6	SL	R/W	0	停止位长度： 0：1bit 1：2bits
5	DL	R/W	0	数据长度： 0：8bits 1：7bits
4	RXPOL	R/W	0	接收极性： 0：非取反 1：取反
3	ERRIE	R/W	0	错误中断使能： 0：禁止接收错误中断 1：允许接收错误中断
2	RXIE	R/W	0	接收中断使能： 0：禁止接收中断 1：允许接收中断
1:0	RXEV	R/W	00	接收中断事件配置，用于控制何种事件下向 CPU 提供接收中断： 00：START 位检测唤醒 01：1byte 数据接收完成 10：接收数据匹配成功 11：下降沿检测唤醒

11.3.5 中断标志寄存器 LPUART_IF (偏移：10h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	TC_IF	R/W	0	发送完成中断标志： 1：发送完一帧数据后中断产生 0：无中断产生 写 1 清 0
2	TXIF	R/W	1	发送 buffer 空中断标志： 1：发送 buffer 空后中断产生 0：无中断产生 写 1 清 0
1	RXNEGIF	R/W	0	RXD 下降沿中断标志： 1：中断产生 0：无中断产生 写 1 清 0

比特	名称	属性	复位值	描述
0	RXIF	R/W	0	接收完成中断标志： 1：接收完一帧数据后中断产生 0：无中断产生 写 1 清 0

11.3.6 波特率寄存器 LPUART_BAUD (偏移：14h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	BAUD	R/W	000	波特率控制（bps）： 000：9600 001：4800 010：2400 011：1200 100：600 其它：300

11.3.7 接收使能寄存器 LPUART_EN (偏移：18h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	TXEN	R/W	0	发送使能： 1：打开 LPUART 发送 0：关闭 LPUART 发送 CPU 写 1 使能后，要反复读取此寄存器，直到读到 1 为止才能进行后面的操作。
0	RXEN	R/W	0	接收使能： 1：打开 LPUART 接收 0：关闭 LPUART 接收 CPU 写 1 使能后，要反复读取此寄存器，直到读到 1 为止才能进行后面的操作。

11.3.8 数据匹配寄存器 LPUART_CMPARE (偏移：1Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	CMPARE	R/W	000000000	比较数据，如果 RXEV=10/11，当接收缓冲区内的数据与 CMPARE 相同时，触发接收完成中断

11.3.9 波特率调制控制寄存器 LPUART_MCTL(偏移：20h)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	MCTL	R/W	0	LPUART 每个 bit 的调制控制信号

11.3.10 匹配中断唤醒配置寄存器 LPUART_WKCKE（偏移：24h）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	WKCKE	R/W	0	LPUART 匹配中断唤醒设置位： 0：接收数据匹配可唤醒 1：接收数据和奇偶检验位都匹配可唤醒

11.4 软件流程

11.4.1 数据接收

1. 配置 LPUART_BAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置调制控制寄存器 LPUART_MCTL 的 MCTL 值。
3. 配置 LPUART_CON 寄存器，选择帧格式、极性、中断参数等。
4. 配置 LPUART_EN 寄存器打开接收使能。
5. 等待中断事件。

11.4.2 数据发送

1. 配置 LPUART_BAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置调制控制寄存器 LPUART_MCTL 的 MCTL 值。
3. 配置 LPUART_CON 寄存器，选择帧格式、极性、中断参数等。
4. 配置 LPUART_EN 寄存器打开发送使能。
5. 等待中断事件。

11.4.3 调制控制寄存器配置建议

软件需要根据通信波特率的不同合理配置调制控制寄存器 LPUART_MCTL 的 MCTL，建议的配置参数表如下：

表 11-2：调制控制寄存器配置建议

Baud	MCTL											
	Bit0（start）	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11
9600	0	1	0	0	1	0	1	0	1	0	0	1
4800	1	1	0	1	1	1	1	1	0	1	1	1

Baud	MCTL											
	Bit0（start）	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	Bit9	Bit10	Bit11
2400	1	1	0	1	1	0	1	1	0	1	1	0
1200	0	1	0	0	1	0	0	1	0	0	1	0
600	0	1	1	0	1	0	1	1	0	1	1	0
300	0	1	0	0	0	0	1	0	0	0	0	1

以上参数表假设 LPUART 工作时钟为准确的 32768Hz，如果使用 RCL 工作，则会引入额外的误差，可能需要微调波特率调制方案来获得更好的通信效果。

11.4.4 休眠模式下的数据接收唤醒

LPUART 支持在 Sleep、DeepSleep 模式下进行数据接收并唤醒芯片。此时芯片功耗极低，并保持对 RXD 引脚的监听，直到特定事件到来后唤醒芯片退出休眠模式。

1. 配置 LPUART_BAUD 寄存器决定波特率。
2. 根据波特率选择合适的调制参数，配置 LPUART_MCTL 寄存器。
3. 配置 LPUART_CON 寄存器，选择帧格式、极性，通过 LPUART_CON[1:0]选择唤醒事件为 START 位、一帧接收完成、一帧数据匹配或 RXD 下降沿检测。
4. 配置 LPUART_EN 寄存器打开接收使能。
5. 软件进入 Sleep/DeepSleep。
6. RXD 引脚唤醒。

12 I2C

12.1 概述

I2C 总线接口连接微控制器和串行 I2C 总线。I2C 模块接收和发送数据，并将数据从串行转换成并行，或并行转换成串行。I2C 模块通过数据引脚 SDA 和时钟引脚 SCL 连接到 I2C 总线，控制所有 I2C 总线规定的时序。本模块支持主模式和从模式。

12.2 主要特征

- 支持主机接收、发送，从机接收、发送四种工作模式
- 支持标准（100Kbps）/快速(400Kbps)/高速(1Mbps)三种工作速率
- 支持 7 位寻址功能和 10 位寻址功能
- 支持中断查询功能

12.3 寄存器描述

I2C 寄存器基地址：0x4000_5400

表 12-1：I2C 寄存器列表

偏置	名称	描述
0x00	I2C_CR	I2C 配置寄存器
0x04	I2C_CLR	I2C 配置清除寄存器
0x08	I2C_STAT	I2C 状态寄存器
0x0C	I2C_DATA	I2C 数据寄存器
0x10	I2C_CCR	I2C 波特率配置寄存器
0x14	I2C_SAD0	I2C SLAVE 地址寄存器 0
0x18	I2C_SADM0	I2C SLAVE 地址屏蔽寄存器 0
0x1C	I2C_XSAD	I2C SLAVE 扩展地址寄存器
0x20	I2C_XSADM	I2C SLAVE 扩展地址屏蔽寄存器
0x24	I2C_SRST	I2C 复位寄存器
0x28	I2C_SAD1	I2C SLAVE 地址寄存器 1
0x2C	I2C_SADM1	I2C SLAVE 地址屏蔽寄存器 1
0x30	I2C_SAD2	I2C SLAVE 地址寄存器 2
0x34	I2C_SADM2	I2C SLAVE 地址屏蔽寄存器 2
0x38	I2C_SAD3	I2C SLAVE 地址寄存器 3
0x3C	I2C_SADM3	I2C SLAVE 地址屏蔽寄存器 3

12.3.1 I2C 配置寄存器 I2C_CR（偏移：00h）

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8	GCAVAL	R	0	General Call 地址标志位： 1：收到 General Call 地址 0：未收到 Ceneral Call 地址
7	IEN	R/W	0	I2C 模块中断使能： 1：中断使能 此位写 0 无效。
6	ENAB	R/W	0	I2C 模块接口使能位： 1：I2C 模块接口使能 此位写 0 无效。
5	STA	R/W	0	开始标志使能： 1：I2C 模块进入主机模式，在总线空闲状态下发送 START 标志： ● 当 I2C 处于主机模式下时置位此位，将会发送一个 RESTART 信号。 ● 当 I2C 处于从机模式下时置位此位，I2C 模块将会在完成当前数据传输以后，然后在总线释放空闲以后进入主机模式并发送 START 信号。发送 START 信号之后，自动清 0。 此位写 0 无效。
4	STP	R/W	0	停止标志使能： 1：发送 STOP 标志： ● 当 I2C 处于主机模式下时置位此位，将会发送一个 STOP 信号。 ● 当 I2C 处于从机模式下时置位此位，I2C 模块将会认为接收到了一个 STOP 信号，而不会在总线上发送 STOP 信号。 ● 若 STA 和 STP 位同时置位，I2C 模块将会先发送 STOP 信号（主机模式下）再发送 START 信号。发送 STOP 信号之后，自动清 0。 此位写 0 无效。
3	IFLG	R/W	0	中断标志位： I2C_STAT 寄存器处于 0xf8 以外的任何状态，此位都会置位。 写 I2C_CLR 寄存器的 CLR_IFLG 位，清 0。STP 位写 1 时，即发送 STOP 标志后，此位也将清 0。

比特	名称	属性	复位值	描述
2	AAK	R/W	0	应答标志使能： 1：应答 ACK 0：应答 NACK（在从机模式下不响应） 当 AAK=1 时，I2C 模块将会在如下情况下相应 ACK： ● 接收到了从机地址寄存器里的地址。 ● 在 GC 位使能后，接收到了通用调用地址。 ● 主/从模式下，接收到一个字节数据。 置 1 后，写 I2C_CLR 寄存器的 CLR_AAK 位清 0。 此位写 0 无效。
1	SLAV10M	R	0	作为 SLAVE 时，收到的数据与扩展地址寄存器中数据相匹配的标志位： 1：接收到的数据与 SLAVE 扩展地址寄存器中的数值相匹配 0：接收到的数据与 SLAVE 扩展地址寄存器中的数据不相同
0	SLAV7M	R	0	作为 SLAVE 时，收到的数据与地址寄存器中数据相匹配的标志位： 1：接收到的数据与 SLAVE 地址寄存器中的数值相匹配 0：接收到的数据与 SLAVE 地址寄存器中的数据不相同

12.3.2 I2C 配置清除寄存器 I2C_CLR（偏移：04h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	CLR_IEN	W	0	I2C 模块中断使能清除寄存器： 1：清除中断使能 此位写 0 无效
6	CLR_ENAB	W	0	I2C 模块使能清除寄存器： 1：关闭 I2C 模块接口，I2C 不进行地址匹配，忽略掉 SCL/SDA 线上的信息 此位写 0 无效
5	CLR_STA	W	0	开始标志清除寄存器： 1：清除发送 START 标志 此位写 0 无效
4	RSV	-	-	保留
3	CLR_IFLG	W	0	中断标志清除寄存器： 1：清除中断标志 此位写 0 无效
2	CLR_AAK	W	0	应答标志清除寄存器： 1：清除应答标志 此位写 0 无效
1:0	RSV	-	-	保留

12.3.3 I2C 状态寄存器 I2C_STAT(偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	STA	R	0xF8	I2C 状态寄存器

I2C_STAT 寄存器 STA 字段不同代码代表的意义：

状态代码	I2C 总线和硬件状态
0x00	由于非法的起始或停止条件的出现，在主机或被选中的从机将出现总线错误;当外部干扰使 I2C 进入未定义的状态时也会出 0x00 状态
0x08	已发送 START 标志
0x10	已发送 RESTART 标志
0x18	已发送 SLAVE 地址加 W 标志，并接收 ACK 位
0x20	已发送 SLAVE 地址加 W 标志，并接收 NAK 位
0x28	主机模式下，已发送 I2C_DATA 中的数据，已接收 ACK
0x30	主机模式下，已发送 I2C_DATA 中的数据，接收到 NAK
0x38	丢失仲裁（地址或数据字节）
0x40	已发送 SLAVE 地址加 R 标志，并加收到 ACK
0x48	已发送 SLAVE 地址加 R 标志，并加收到 NAK
0x50	主机模式下，已接收数据字节，ACK 已发出
0x58	主机模式下，已接收数据字节，NAK 已发出
0x60	已接收自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出。
0x68	作为主机时，丢失掉仲裁，并且以接收到自身的 SLAVE 寄存器地址加 W 标志，ACK 已发出。
0x70	已接收通用调用地址（0x00）；已发出 ACK
0x78	作为主机时，丢失掉仲裁，并且已接收到通用调用地址加 W 标志，ACK 已发出。
0x80	在接收到自身从地址后，接收到数据字节，已返回 ACK
0x88	在接收到自身从地址后，接收到数据字节，已返回 NAK
0x90	在接收到通用调用地址后，接收到数据字节，已返回 ACK
0x98	在接收到通用调用地址后，接收到数据字节，已返回 NAK
0xA0	从机模式下，接收到停止条件或重复起始条件
0xA8	已接收自身的从地址加 R 标志；已返回 ACK
0xB0	作为主机时，丢失掉仲裁，并且已接收到通用调用地址加 R 标志，ACK 已发出。
0xB8	从机模式下（AAK=1），已发送数据；已收到 ACK
0xC0	从机模式下（AAK=1），已发送数据；已收到 NAK
0xC8	从机模式下（AAK=0），已发送最后一个字节数据，已收到 ACK
0xD0	从机模式下（AAK=0），已发送最后一个字节数据，已收到 NAK
0xE0	已发送 10 位 SLAVE 第二段地址+W，接收到 ACK
0xE8	已发送 10 位 SLAVE 第二段地址+W，接收到 NACK
0xF8	无可用的相关状态信息，IFLG=0

注：若 I2C 总线上出现非法的起始或停止信号时，会进入总线错误状态（状态代码 0x00）。用户可以通过置位 STP 来清除 IFLG 位状态，I2C 模块会返回到空闲状态，此操作不会在 I2C 总线上发送 STOP 信号。

12.3.4 I2C 数据寄存器 I2C_DATA(偏移：0Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	DATA	R/W	0x0	I2C 数据寄存器： ● 在 I2C 发送模式下，写发送数据到这个寄存器 ● 在 I2C 接收模式下，读接收数据从这个寄存器

12.3.5 I2C 波特率配置寄存器 I2C_CCR(偏移：10h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6:4	CCRM	R/W	0x0	波特率配置位 M
3:0	CCRN	R/W	0x0	波特率配置位 N

FOSCL = F_{SCL} = P_{CLK} / (2^M×(N+1)×10)；其中，FOSCL 是 I2C 接口输出的 SCL 的频率。

12.3.6 I2C SLAVE 地址寄存器 0 I2C_SAD0 (偏移：14h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	ADR0	R/W	0x0	I2C 从机模式地址 0
0	GC0	R/W	0x0	广播地址应答使能： 1：使能 0：不使能

12.3.7 I2C SLAVE 地址屏蔽寄存器 0 I2C_SADM0 (偏移：18h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	AMR0	R/W	0x7f	I2C 从机模式地址屏蔽寄存器 0。每一位与 ADR0 中的位相对应。 ● 对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR0 中对应位的值。 ● 对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR0 中对应位的值。
0	RSV	-	-	保留

12.3.8 10 比特 I2C SLAVE 地址寄存器 I2C_XSAD (偏移：1Ch)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10:1	XADR	R/W	0x0	I2C 从机模式 10 比特地址位
0	XGC	R/W	0x0	10 比特地址模式下，广播地址应答使能: 1：使能 0：不使能

12.3.9 10 比特 I2C SLAVE 地址屏蔽寄存器 I2C_XSADM (偏移：20h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:1	XAMR	R/W	0xff	10 比特 I2C 从机模式地址屏蔽寄存器 0。每一位与 XADR 中的位相对应。 ● 对应位为 1 表示，在此 I2C 模块作为从机时，比较 XADR 中对应位的值。 ● 对应位为 0 表示，在此 I2C 模块作为从机时，不比较 XADR 中对应位的值。
0	RSV	-	-	保留

12.3.10 I2C 复位寄存器 I2C_SRST (偏移：24h)

比特	名称	属性	复位值	描述
31:0	SRST	W	0x0	写此寄存器，复位 I2C 模块。

12.3.11 I2C SLAVE 地址寄存器 1 I2C_SAD1 (偏移：28h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	ADR1	R/W	0x0	I2C 从机模式地址 1
0	GC1	R/W	0x0	广播地址应答使能： 1：使能 0：不使能

12.3.12 I2C SLAVE 地址屏蔽寄存器 1 I2C_SADM1 (偏移：2Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	AMR1	R/W	0x7f	I2C 从机模式地址屏蔽寄存器 1。每一位与 ADR1 中的位相对应。 ● 对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR1 中对应位的值。 ● 对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR1 中对应位的值。
0	RSV	-	-	保留

12.3.13 I2C SLAVE 地址寄存器 2 I2C_SAD2 (偏移：30h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	ADR2	R/W	0x0	I2C 从机模式地址 2
0	GC2	R/W	0x0	广播地址应答使能： 1：使能 0：不使能

12.3.14 I2C SLAVE 地址屏蔽寄存器 2 I2C_SADM2 (偏移：34h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	AMR2	R/W	0x7f	I2C 从机模式地址屏蔽寄存器 2。每一位与 ADR2 中的位相对应。 ● 对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR2 中对应位的值。 ● 对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR2 中对应位的值。
0	RSV	-	-	保留

12.3.15 I2C SLAVE 地址寄存器 2 I2C_SAD3 (偏移：38h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	ADR3	R/W	0x0	I2C 从机模式地址 3
0	GC3	R/W	0x0	广播地址应答使能： 1：使能 0：不使能

12.3.16 I2C SLAVE 地址屏蔽寄存器 3 I2C_SADM3 (偏移：3Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:1	AMR3	R/W	0x7f	I2C 从机模式地址屏蔽寄存器 2。每一位与 ADR3 中的位相对应。 ● 对应位为 1 表示，在此 I2C 模块作为从机时，比较 ADR3 中对应位的值。 ● 对应位为 0 表示，在此 I2C 模块作为从机时，不比较 ADR3 中对应位的值。
0	RSV	-	-	保留

12.4 使用流程

12.4.1 初始化程序

主机初始化：

1. 将相应 GPIO 复用成 I2C_SCL 和 I2C_SDA。
2. 在系统寄存器中开启 I2C 时钟和使 I2C 处于工作状态。
3. 将 I2C_SRST 写 1，复位 I2C 模块。
4. 配置 I2C_CCR[3:0]和[6:4]，设置 I2C 通信速率（标准/快速/高速）。
5. 对于从机模式，设置 I2C_CR[2]为 1，。设置 I2C_CR[6]为 1。

从机初始化：

1. 将相应 GPIO 复用成 I2C_SCL 和 I2C_SDA。
2. 在系统寄存器中开启 I2C 时钟和使 I2C 处于工作状态。
3. 将 I2C 复位寄存器写 1，复位 I2C 模块。
4. 将自身的从机地址装入 I2C_SAD0[7:1] / I2C_SAD1[7:1] / I2C_SAD2[7:1] / I2C_SAD3[7:1] / I2C_XSAD[10:1]，设置好地址匹配寄存器 I2C_SADM0[7:1] / I2C_SADM1[7:1] / I2C_SADM2[7:1] / I2C_SADM3[7:1] / I2C_XSADM[10:1] / I2C_SADx[0]，使能广播地址应答（如果需要）。
5. 配置 I2C_CR[7]为 1，使能 I2C 中断（如果需要）。
6. 设置 I2C_CR[2]为 1，设置 I2C_CR[6]为 1。

使用建议：master TX/RX 均用查询方式；slave RX 中断/查询方式均可，TX 查询方式。

12.4.2 主机发送功能

1. 向 I2C_CR[5]写 1，发出 START 标志。
2. 等待 I2C_STAT[7:0]数值变为 0x08（已发送 START 标志）；若 START 标志发送成功，则 STA 发送标志自动清 0。
3. 向 I2C_DATA[7:0]写入 SAL（7 位/10 位）+W（0）。
4. 向 I2C_CLR[3]写 1，清除中断标志后，主机 SCL 线发送时钟信号，SDA 线发送 SLA+W。
5. 等待 I2C_STAT[7:0]数值变为 0x18（已发送 SLAVE 地址加 W 标志，并接收 ACK 位）。
6. 向 I2C_DATA[7:0]寄存器写入待发送的数据/要写入的从设备内存地址（10bit 寻址为发送第二段设备地址+w）。
7. 接收到从设备 I2C_STAT[7:0]=0xe0 状态为第二段地址已经发送并收到 ACK，接着循环发送数据（10bit 寻址才有此处通讯动作）。
8. 向 I2C_CLR[3]写 1，发送数据。
9. 等待 I2C_STAT[7:0]数值变为 0x28（已发送 I2C_DATA 中的数据，已接收 ACK）。
10. 重复步骤 7-10，直到待发的数据全部发送完毕。
11. 向 I2C_CR[4]写 1，发送 STOP 标志，传输完成。

12.4.3 主机接收功能

1. 向 I2C_CR[5]写 1，发出 START 标志。
2. 等待 I2C_STAT[7:0]为 0x08（已发送 START 标志）；若 START 标志发送成功，则 STA 发送标志自动清 0。
3. 向 I2C_DATA[7:0]写入 SLA（7 位）+R(1)（7bit 寻址为此操作）。

4. 向 I2C_CLR[3]写 1，清除中断标志后，主机 SCL 线发送时钟信号，SDA 线发送 SLA+R（7bit 寻址为此操作）。
5. 向 I2C_DATA[7:0]写入第一段 SLA+W(0)（10bit 寻址为此操作）。
6. 等待 I2C_STAT[7:0]数值变为 0x18（已发送 SLAVE 地址加 W 标志，并接收 ACK 位）（10bit 寻址为此操作）。
7. 向 I2C_DATA[7:0]写入第二段 SLA（10bit 寻址为此操作）。
8. 接收到从设备 I2C_STAT[7:0]=0xE0 状态为第二段地址已经发送并收到 ACK（10bit 寻址为此操作）。
9. 向 I2C_CR[5]写 1，发出 RESTART 标志。
10. 向 I2C_CLR[3]写 1，清除中断标志后，等待 I2C_STAT[7:0]数值变为 0x10（已发送 RESTART 标志）（10bit 寻址为此操作）。
11. 向 I2C_DATA[7:0]写入第一段 SLA+R(1)（10bit 寻址为此操作）。
12. 等待 I2C_STAT[7:0]数值变为 0x40（已发送 SLAVE 地址加读标志，并接收 ACK）（10bit 寻址为此操作）。
13. 向 I2C_CR[2]写 1，设置 I2C_CR 寄存器的 AAK 位。
14. 向 I2C_CLR[3]写 1，开始接收数据。
15. 等待 I2C_STAT[7:0]数值变为 0x50（已接收数据字节，ACK 已发出），读取 I2C_DATA 中收到的数据。
16. 重复 14-15 步骤，直到接收完所有数据。
17. 向 I2C_CR[4]写 1，发送 STOP 标志，传输完成。

12.4.4 从机接收功能

查询方式：

1. 等待 I2C_STAT[7:0]数值变为 0x60（已接收自身 SLAVE 地址+W，ACK 已发出）。
2. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号，SDA 线接收数据。
3. 等待 I2C_STAT[7:0]数值变为 0x80（在接收到自身从地址后，接收到数据字节，已返回 ACK）。
4. 读取 I2C_DATA 里面的数据保存到变量。
5. 重复 2-4 步骤，直到接收完所有数据。
6. 等待 I2C_STAT[7:0]数值变为 0xA0（从机模式下，接收到停止条件或重复起始条件）。
7. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号，SDA 线接收数据。

中断方式：

1. 等待 I2C_STAT[7:0]数值变为 0x80（在接收到自身从地址后，接收到数据字节，已返回 ACK），产生中断。
2. 在中断服务函数中读取 I2C_DATA 里面的数据保存到变量。

3. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号，SDA 线接收数据。

12.4.5 从机发送功能

1. 向 I2C_CR 寄存器的 AAK 位写 1，使能应答。
2. 等待 I2C_STAT[7:0]数值变为 0xA8（已接收自身 SLAVE 地址+R，ACK 已发出）。
3. 向 I2C_DATA[7:0]写入发送的字节数据。
4. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号。
5. 等待 I2C_STAT[7:0]数值变为 0xB8（从机模式下（AAK=1），已发送数据；已收到 ACK）。
6. 重复 5-6 步骤，直到发送完所有数据。
7. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号。
8. 等待 I2C_STAT[7:0]数值变为 0xA0（从机模式下，接收到停止条件或重复起始条件）。
9. 向 I2C_CLR[3]写 1，清除中断标志后，从机 SCL 线释放时钟信号。

13 SPI0/1

13.1 概述

串行外设接口（Serial Peripheral Interface，SPI）是外部设备通过单线交换数据的串行同步通讯手段。芯片提供了 2 个 SPI 接口模块，可配置为主设备或从设备，实现与外部的 SPI 通信。

13.2 主要特性

- 全双工或半双工单数据线串行同步收发
- 主从模式
- 可编程时钟极性和相位（支持模式 0、1、2、3）
- 可编程比特速率
- 从模式最大频率为 Fsys/2
- 传输结束中断标志
- 写冲突错标志
- 主模式错误检测、保护和中断标志
- 支持 DMA
- 8 个 byte fifo 深度

13.3 寄存器描述

SPI0 寄存器基地址：0x4000_0800

SPI1 寄存器基地址：0x4000_5800

表 13-1：SPI 寄存器列表

偏置	名称	描述
0x0	SPI_CR	SPI 配置寄存器
0x4	SPI_CS0	SPI 主模式控制寄存器 0
0x8	SPI_CS1	SPI 主模式控制寄存器 1
0x14	SPI_OPCR	SPI 过程控制寄存器
0x18	SPI_IE	SPI 中断控制寄存器
0x1C	SPI_IF	SPI 中断标志寄存器
0x20	SPI_TXBUF	SPI 发送缓存寄存器
0x24	SPI_RXBUF	SPI 接收缓存寄存器
0x28	SPI_DMARXLEV	SPI DMA 接收设置寄存器
0x2c	SPI_DMATXLEV	SPI DMA 发送设置寄存器

13.3.1 SPI 配置寄存器 SPI_CR (偏移：00h)

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19	MOSI_POL	R/W	0	SPI 主模式下，MOSI 对应默认电平选择： 1：MOSI 空闲时为高电平 0：MOSI 空闲时为低电平 注：当通信在进行时不能改变该位的值 注：当 SSN 为低时不能改变该位的值
18	DC_REG	R/W	0	DC 输出寄存器控制位： 1：DC 输出为 1 0：DC 输出取决于 FIFO 中的第 8bit
17	DC_DIR	R/W	0	DC 方向控制使能： 1：DC 为输出模式 0：DC 为输入模式
16	DC_EN	R/W	0	DC 控制使能： 1：使能 DC 模式 0：关闭 DC 模式
15:14	SPI_FRS	R/W	10	SPI 帧格式设定： 00：6bit 格式 01：7bit 格式 10：8bit 格式 11：9bit 格式
13	DMA_TX_EN	R/W	0	DMA TX 使能： 1：使能 DMA TX 请求 0：关闭 DMA TX 请求
12	DMA_RX_EN	R/W	0	DMA RX 使能： 1：使能 DMA RX 请求 0：关闭 DMA RX 请求
11	FLTEN	R/W	1	Slave 输入管脚滤波使能（SSN/SCK/MOSI）： 1：使能 4ns 滤波 0：不滤波
10	SSNM	R/W	0	Master 模式下 SSN 控制模式选择： 1：每发送完 8bit 后 Master 拉高 SSN，维持高电平时间由 WAIT 寄存器控制 0：每发送完 8bit 后 Master 保持 SSN 为低，维持低电平时间由 WAIT 寄存器控制
9	TXO_AC	R/W	1	TXONLY 硬件自动清空的使能： 1：TXONLY 硬件自动清零有效，软件使能 TXO 后，等待发送完毕后，硬件清零 0：关闭 TXONLY 硬件自动清零
8	TXO	R/W	0	TXONLY 控制位： 1：启动 Master 的单发送模式 0：关闭单发送模式
7	MSPA	R/W	0	Master Sampling Position Adjustment，Master 对 MISO 信号的采样位置调整，用于高速通信时补偿 PCB 走线延迟： 1：采样点延迟半个 SCK 周期 0：不调整

比特	名称	属性	复位值	描述
6	SSPA	R/W	0	Slave Sending Position Adjustment，Slave MISO 发送位置调整： 1：提前半个 SCK 周期发送 0：不调整
5	MM	R/W	1	Master/Slave 模式选择： 1：Master 模式 0：Slave 模式
4:3	WAIT	R/W	0	Master 模式下，每发完 8Bit 后加入至少(1+WAIT)个 SCK cycle 等待时间再传输下一个 8Bit 的数据
2	RSV	-	-	保留
1	SSNSEN	R/W	0	Master 模式下，软件控制 SSN 使能： 1：Master 模式下 SSN 输出由软件控制 0：Master 模式下 SSN 输出由硬件自动控制
0	SPIEN	R/W	0	SPI 使能。采用关闭时钟的方式来关闭使能： 1：使能 SPI 0：关闭 SPI，清空发送接收缓存

13.3.2 SPI 主模式控制寄存器 0 SPI_CS0 (偏移：04h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	TRI0_EN	R/W	0	SPI 三线模式使能： 1：使能三线模式 0：禁止三线模式
6	SSN0	R/W	0	SPI 主模式下，CS0 对应 Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平： 1：SSN 输出低电平 0：SSN 输出高电平
5:3	BAUD0	R/W	001	SPI 主模式下，CS0 对应 Master 模式波特率配置位（通信速率最高设置为 12M）： 000： f_{PCLK}/2 001： f_{PCLK} /4 010： f_{PCLK} /8 011： f_{PCLK} /16 100： f_{PCLK} /32 101： f_{PCLK} /64 110： f_{PCLK} /128 111： f_{PCLK} /256 当通信正在进行的时候，不能修改这些位。
2	LSBF0	R/W	0	SPI 主模式下，CS0 对应帧格式 (Frame format)： 1：先发送 LSB 0：先发送 MSB 注：当通信在进行时不能改变该位的值。
1	CPOL0	R/W	0	SPI 主模式下，CS0 对应时钟极性选择： 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注：当通信在进行时不能改变该位的值。当 SSN 为低时不能改变该位的值

比特	名称	属性	复位值	描述
0	CPHA0	R/W	0	SPI 主模式下，CS0 对应时钟相位选择： 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

13.3.3 SPI 主模式控制寄存器 1 SPI_CS1 (偏移：08h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	SSN1	R/W	0	SPI 主模式下，CS1 对应 Master 模式下，如果 SSNSEN 为 1，软件可以通过此位控制 SSN 输出电平： 1：SSN 输出低电平 0：SSN 输出高电平
5:3	BAUD1	R/W	001	SPI 主模式下，CS1 对应 Master 模式波特率配置位： 000： f_{PCLK}/2 001： f_{PCLK} /4 010： f_{PCLK} /8 011： f_{PCLK} /16 100： f_{PCLK} /32 101： f_{PCLK} /64 110： f_{PCLK} /128 111： f_{PCLK} /256 当通信正在进行的时候，不能修改这些位。
2	LSBF1	R/W	0	SPI 主模式下，CS1 对应帧格式 (Frame format) ： 1：先发送 LSB 0：先发送 MSB 注：当通信在进行时不能改变该位的值。
1	CPOL1	R/W	0	SPI 主模式下，CS1 对应时钟极性选择： 1：串行时钟停止在高电平 0：串行时钟停止在低电平 注： ● 当通信在进行时不能改变该位的值。 ● 当 SSN 为低时不能改变该位的值。
0	CPHA1	R/W	0	SPI 主模式下，CS1 对应时钟相位选择： 1：第二个时钟边沿是第一个捕捉边沿 0：第一个时钟边沿是第一个捕捉边沿 注：当通信在进行时不能改变该位的值。

13.3.4 SPI 过程控制寄存器 SPI_OPCR (偏移：14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留 读为 0
5	SSNNEGC	W1C	0	SSN Negedge Clear，软件写 1 清除 SSN 下降沿中断，写 0 无效
4	SSNPOSC	W1C	0	SSN Posedge Clear，软件写 1 清除 SSN 上升沿中断，写 0 无效
3	TXBFC	W1C	0	Transmit Buffer Clear，软件写 1 清除发送缓存，写 0 无效

比特	名称	属性	复位值	描述
2	RXBFC	W1C	0	Receive Buffer Clear，软件写 1 清除接收缓存，写 0 无效
1	MERRC	W1C	0	Master Error Clear，软件写 1 清除 SPI_IF. MERR 位
0	SERRC	W1C	0	Slave Error Clear，软件写 1 清除 SPI_IF.SERR 位

13.3.5 SPI 中断控制寄存器 SPI_IE (偏移：18h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留 读为 0
23:20	RXFIFOLEVEL	R	0	RX FIFO 当前所存数据个数
19:16	TXFIFOLEVEL	R	0	TX FIFO 当前所存数据个数
15:11	RSV	-	-	保留
10	SSNNEGIE	R/W	0	SSN Negedge 中断使能： 1：使能 0：不使能
9	SSNPOSIE	R/W	0	SSN Posedge 中断使能： 1：使能 0：不使能
8	RNFIE	R/W	0	Rx Fifo Full 中断使能： 1：使能 0：不使能
7	TNFIE	R/W	0	Tx Fifo Not Full 中断使能： 1：使能 0：不使能
6	MERRIE	R/W	0	Master Error 中断使能： 1：使能 0：不使能
5	SERRIE	R/W	0	Slave Error 中断使能： 1：使能 0：不使能
4	RXCOLIE	R/W	0	接收缓存溢出中断使能： 1：使能 0：不使能
3	TXCOLIE	R/W	0	发送缓存溢出中断使能： 1：使能 0：不使能
2	IDLEIE	R/W	0	SPI 空闲标志中断使能： 1：使能 0：不使能
1	TXBEIE	R/W	0	TX Buffer Empty 中断使能： 1：使能 0：不使能
0	RXBFIE	R/W	0	RX Buffer 中断使能： 1：使能 0：不使能

13.3.6 SPI 中断标志寄存器 SPI_IF (偏移：1Ch)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10	SSNNEG	R	0	SSN Negedge 标志： 检测到 SSN 下降沿，SSNNEG 置位
9	SSNPOS	R	0	SSN Posedge 标志： 检测到 SSN 上升沿，SSNPOS 置位
8	RNF	R	0	Spi Rx Fifo Full： 1：SPI0 Rx Fifo 满 0：SPI0 Rx Fifo 未满
7	TNF	R	1	Spi Tx Fifo Not Full： 1：SPI0 Tx Fifo 未满 0：SPI0 Tx Fifo 满
6	MERR	R	0	Master Error 标志： 当 Master 下传输未满 8 位 SSN 就被拉高时，MERR 置位
5	SERR	R	0	Slave Error 标志： 当 Slave 下传输未满 8 位 SSN 就被拉高时，SERR 置位
4	RXCOL	R/W	0	接收缓存溢出： 1：接收缓存溢出 0：接收缓存未溢出 注：软件写 1 清零
3	TXCOL	R/W	0	发送缓存溢出： 1：发送缓存溢出 0：发送缓存未溢出 注：软件写 1 清零
2	IDLE	R	1	SPI0 空闲标志，只读： 1：SPI0 传输空闲 0：SPI0 传输进行中
1	TXBE	R	1	TX Buffer Empty 标志位： 1：发送缓存空，软件写 TXBUF 清零 0：发送缓存非空
0	RXBF	R	0	RX Buffer 非空标志位： 1：接收缓存非空 0：接收缓存空

13.3.7 SPI 发送缓存寄存器 SPI_TXBUF (偏移：20h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	TXBUF	W	0	SPI 发送缓存，发送 FIFO 入口地址。此 IP 一共含有 8 个 9-bit 数据的发送 FIFO，写此地址，将要发送的数据写入 FIFO 中。

13.3.8 SPI 接收缓存寄存器 SPI_RXBUF (偏移：24h)

比特	名称	属性	复位值	描述
31:9	RSV	-	-	保留
8:0	RXBUF	R	0	SPI 接收缓存，接收 FIFO 入口地址。此 IP 一共含有 8 个 9-bit 数据的接收 FIFO，读此地址，将接收的数据从 FIFO 中读取出来。

13.3.9 SPI DMA 接收设置寄存器 SPI_DMARXLEV (偏移：28h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	DMA_RX_LEV	R/W	0	SPI0 接收 FIFO DMA 请求设置： 当 RX FIFO 中的数据个数大于此寄存器设置值时，产生 DMA RX 请求。

13.3.10 SPI DMA 发送设置寄存器 SPI_DMATXLEV (偏移：2Ch)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3:0	DMA_TX_LEV	R/W	0	SPI0 发送 FIFO DMA 请求设置： 当 TX FIFO 中的数据个数小于此寄存器设置值时，产生 DMA TX 请求。

13.4 使用流程

SPI0 两种引脚搭配方式如下表所示：

表 13-2：SPI0 两种引脚搭配方式表

信号描述功能位	SPI0 配置 1	SPI0 配置 2
CS	SPI0_CS0	SPI0_CS1
MISO	SPI0_MISO	SPI0_MI1
MOSI	SPI0_MOSI	SPI0_MOSI
CLK	SPI0_SCK	SPI0_SCK

注:引脚 SPI0_CS1 和 SPI0_MI1 搭配使用，SPI0_CS0 与 SPI0_MISO 搭配使用

SPI1 两种引脚搭配方式如下表所示：

表 13-3：SPI1 两种引脚搭配方式表

信号描述功能位	SPI1 配置 1	SPI1 配置 2
CS	SPI1_CS0	SPI1_CS1
MISO	SPI1_MISO	SPI1_MI1
MOSI	SPI1_MOSI	SPI1_MOSI
CLK	SPI1_SCK	SPI1_SCK

注:引脚 SPI1_CSN1 和 SPI1_MI1 搭配使用，SPI1_CSN0 与 SPI1_MISO 搭配使用

13.4.1 初始化程序

1. 配置开启 SPI 模块时钟与复位 PERI_CLKEN /PERI_RESET。
2. 配置相应的 GPIO 引脚复用为 SPI_SCK、 SPI_MISO、 SPI_MOSI、 SPI_CLK，用作输入功能的引脚需配置 PADIEx 寄存器使能引脚输入。
3. 配置 SPI_CR[5]，设置主从模式。
4. 配置 SPI_CR[10]，设置 SSN 控制模式。
5. 配置 SPI_CR[1]，设置 SSN 输出由软件还是硬件控制。
6. 配置 SPI_CSx[6]，设置 SSN 输出高电平还是低电平。
7. 配置 SPI_CSx[2]，设置先发送的是 MSB 还是 LSB。
8. 配置 SPI_CSx[0]，设置第一个时钟边沿采样还是第二个时钟边沿采样。
9. 配置 SPI_CSx[1]，设置串行时钟停止在高电平还是低电平。
10. 配置 SPI_CSx.BAUDx[2:0]，以设置串行时钟波特率（若为从器件模式则不用设置，串行时钟速率由主器件决定）。需要中断时，配置 SPI_IE 使能相应的中断。
11. 配置 SPI_CR[0]，使能 SPI。

13.4.2 发送流程

- 主器件发送流程：

  配置 SPI_CSx[6]拉低 SSN 引脚启动传输，配置 SPI_CR[8]位为高，将数据写入 SPI_TXBUF 寄存器，等待 SPI_IF[2]置位发送完成，配置 SPI_CR[8]位为低，传输完成后将 SSN 拉高。
- 从器件发送流程：

  配置 SPI_CR[8]为高，将数据写入 SPI_TXBUF 寄存器，等待 SPI_IF[2]置位发送完成，配置 SPI_CR[8]为低。

13.4.3 接收流程

- 主器件接收流程：

  配置 SPI_CSx[6]拉低 SSN 引脚启动传输，将数据写入 SPI_TXBUF 寄存器，等待 SPI_IF[0]置位，读取 SPI_RXBUF 寄存器数据完成数据接收，传输完成后将 SSN 拉高。
- 从器件接收流程：

  等待 SPI_IF[0]置位，读取 SPI_RXBUF 寄存器数据完成数据接收。

13.4.4 SPI DMA 发送流程

1. 配置 SPI_DMATXLEV[2:0]，设置产生 DMA TX 请求的 FIFO 数据个数。
2. 使能 SPI_CR[13]，使能 DMA TX 请求。
3. 配置开启 DMA 模块时钟与复位 PERI_CLKEN / PERI_RESET。
4. 配置通道控制信息寄存器 DMA_CHCTRLCx，根据实际应用配置数据位宽，传输模式（8 位位宽、内存到外设模式）。
5. 配置 DMA_CHCTRLCx[4:3]和 DMA_CHCTRLCx[2:1]，选择【目的外设】和【源外设】（目的外设为 SPI 发送，源外设为 MEM），此位在用到外设的模式下起效。
6. 配置 DMA_CHCTRLCx[2:1]和 DMA_CHCTRLCx[4:3]，选择【目的地址】和【源地址】是否随数据传输递增（源地址递增、目的地址不变）。
7. 如需使用中断，则配置 DMA 中断屏蔽寄存器 DMA_INTMASK，使能对应的通道中断。
8. 配置 DMA_SRCADDRCx，配置通道源地址。
9. 配置 DMA_DSTADDRCx，配置通道目的地址。
10. 配置 DMA_CHCTRLCx[29:15]，配置传输块数量。
11. 等待上述配置、以及相应的源地址和目的地址准备就绪，使能 DMA（DMAC_EN）。
12. 配置 DMA_CHCTRLCx[0]，使能 DMA 通道传输。
13. 等待 DMA_CHCTRLCx[0]为 0，传输完成。若使能了传输结束中断，则等待传输结束中断后再处理。

13.4.5 SPI DMA 接收流程

1. 配置 SPI_DMARXLEV[2:0]，设置产生 DMA RX 请求的 FIFO 数据个数。
2. 使能 SPI_CR[12]，使能 DMA RX 请求。
3. 配置开启 DMA 模块时钟与复位 PERI_CLKEN / PERI_RESET。
4. 配置通道控制信息寄存器 DMA_CHCTRLCx，根据实际应用配置数据位宽，传输模式（8 位位宽、外设到内存模式）。
5. 配置 DMA_CHCTRLCx[4:3]和 DMA_CHCTRLCx[2:1]，选择【目的外设】和【源外设】（目的外设为 MEM，源外设为 SPI 接收），此位在用到外设的模式下起效。
6. 配置 DMA_CHCTRLCx[2:1]和 DMA_CHCTRLCx[4:3]，选择【目的地址】和【源地址】是否随数据传输递增（源地址不变、目的地址递增）。
7. 如需使用中断，则配置 DMA 中断屏蔽寄存器 DMA_INTMASK，使能对应的通道中断。
8. 配置 DMA_SRCADDRCx，配置通道源地址。
9. 配置 DMA_DSTADDRCx，配置通道目的地址。
10. 配置 DMA_CHCTRLCx[29:15]，配置传输块数量。

11. 等待上述配置、以及相应的源地址和目的地址准备就绪，使能 DMA（DMAC_EN）；
12. 配置 DMA_CHCTRLCx[0]，使能 DMA 通道传输。
13. 等待 DMA_CHCTRLCx[0]为 0，传输完成。若使能了传输结束中断，则等待传输结束中断后再处理。

14 CAN

14.1 概述

CAN(Controller Area Network) 控制器可以用于汽车电子和工业控制领域，支持 CAN2.0A/B 协议。

14.2 主要特性

- 具有优先权和仲裁功能
- 可根据报文的 ID 决定接收或屏蔽该报文
- 可靠的错误处理和检错机制
- 发送的信息遭到破坏后，可自动重发
- 节点在错误严重的情况下具有自动退出总线的功能

14.3 寄存器描述

CAN 寄存器基地址：0x4000_5C00

表 14-1：CAN 寄存器列表

偏置	名称	描述
0x00	CAN_MR	模式寄存器
0x04	CAN_CMR	指令寄存器
0x08	CAN_SR	状态寄存器
0x0C	CAN_ISR	中断状态寄存器
0x10	CAN_IMR	中断使能寄存器
0x14	CAN_RMC	接收数据计数寄存器
0x18	CAN_BTR0	总线时序寄存器 0
0x1C	CAN_BTR1	总线时序寄存器 1
0x20	CAN_TXBUF	发送缓存寄存器
0x24	CAN_RXBUF	接收缓存寄存器
0x28	CAN_ACR	接收过滤匹配寄存器
0x2C	CAN_AMR	接收过滤屏蔽寄存器
0x30	CAN_ECC	错误码捕捉寄存器
0x34	CAN_RXERR	接收错误计数寄存器
0x38	CAN_TXERR	发送错误计数寄存器
0x3C	CAN_ALC	仲裁丢失捕获寄存器
0x40	CAN_RXADDR	接收缓存基地址设置寄存器

14.3.1 模式寄存器 CAN_MR (偏移：00h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7	RXF_CLR	W	0	RX_FIFO 指针清除位： 1：复位 RX_FIFO 的读写指针，复位后此位自动恢复为 0 0：无变化
6:3	RSV	R	0	保留
2	RM	R/W	1	复位模式设置位： 1：CAN 工作在复位模式 0：CAN 工作在其他模式 在复位模式中不进行数据的发送和接收，此模式用于进行一些硬件的配置（某些寄存器只能在复位模式下进行写操作），在复位模式以后，可进入监听模式或者正常模式。
1	LOM	R/W	0	监听模式设置位： 1：若 RM=0，CAN 进入监听模式* 0：若 RM=0，CAN 进入正常模式 此位只能在复位模式设置
0	AFM	R/W	0	硬件匹配数据选择位： 1：使用单过滤器 0：使用双过滤器 此位只能在复位模式设置

注：在监听模式下，即使成功接收到消息，CAN 控制器也不会对 CAN 总线进行应答（不会发送 ACK 响应）。错误计数器将停止在当前值。监听模式主要用于比特率检测，不会干扰网络流量，监听模式还可用于 CAN 总线分析仪。

14.3.2 指令寄存器 CAN_CMR (偏移：04h)

比特	名称	属性	复位值	描述
31:3	RSV	R	0	保留
2	TR	W	0	发送请求设置位： 1：启动发送，进行帧传输 0：禁止发送
1	AT	W	0	中止传输允许位： 1：允许中止传输 0：禁止中止传输 同时设置 TR 和 AT 可启动单发传输，在总线错误或者仲裁丢失的情况下，不会执行帧的重新传输。中止只对即将要传输的帧作用，已经发出的帧无法中止。如果在上一个命令中将 TR 设为 1 来启动传输，则无法通过将 TR 位设为 0 来取消，此时可通过设置 AT 为 1 来取消传输。
0	RSV	R	0	保留

14.3.3 状态寄存器 CAN_SR (偏移：08h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7	RBS	R	0	接收 FIFO 状态： 1：FIFO 中至少有一条消息 0：FIFO 中没有消息
6	DSO	R	0	数据溢出状态： 1：RX FIFO 溢出，RX 溢出中断触发（如使能） 0：自上次清除数据溢出以来未发生溢出
5	TBS	R	1	发送 BUFFER 状态： 1：发送 BUFFER 可被 CPU 写入 0：发送 BUFFER 已锁定。正在发送消息或正在等待发送。如果 CPU 在锁定状态下（TBS = 0）尝试写入发送缓冲区，则不接受已写入的数据
4	RSV	R	0	保留
3	RS	R	0	接收状态位： 1：CAN 正在接收 0：CAN 未处于接收状态
2	TS	R	0	发送状态位： 1：CAN 正在传输 0：CAN 未处于传输状态
1	ES	R	0	错误状态位： 1：至少一个 CAN 错误计数器达到错误警告限制 0：正常状态
0	BS	R	0	总线状态位 1：离线状态。CAN 控制器处于复位模式，错误警告中断触发（如使能）。发送错误计数器设为 127，接收错误计数器设为 0。CAN 将一直处于复位模式，直到 CPU 将 RM 位清掉。完成此操作后，CAN 将等待 128 次总线空闲信号的出现（11 个连续的隐性位），发送错误计数器向下计数。然后 BS 位清 0，错误计数器复位，错误警告中断触发（如使能） 0：正常状态。可进行帧传输和接收

14.3.4 中断状态/应答寄存器 CAN_ISR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:7	RSV	R	0	保留
6	ALI	R/W	0	仲裁丢失中断状态位： 当 CAN 在消息传输过程中丢失仲裁并成为接收端时，此位置位，可以读取 ALC 寄存器以检查丢失了仲裁段中的哪一位，写 1 清除此中断
5	EWI	R/W	0	错误警告中断状态位： 当 SR 寄存器的 ES 或 BS 位改变时，错误警告中断置位。因此，它可用于检测 CAN 是否进入或退出总线关闭状态。写 1 清除中断
4	EPI	R/W	0	错误被动中断状态位： 当 CAN 总线控制器达到或退出错误被动级别（即在状态更改为主动到被动或被动到主动）时，此位置位。写 1 清除中断
3	RI	R/W	0	接收中断状态位： 当接收 FIFO 中至少有一条 CAN 帧数据时，CAN 将此位置 1。读取消息后，CPU 必须将 RI 位写 1（消息读取确认），以减少 RX 消息计数器（RMC）计数，RMC 不会自动递减
2	TI	R/W	0	发送中断状态位： 成功发送后，发送中断位被置位。在写入新的数据帧之前可通过清除 TI 位（写 1 清除）将写指针复位到 TX RAM
1	BEI	R/W	0	总线错误中断状态位： 当 CAN 在发送或接收消息时遇到总线错误时，将 BEI 置位。写 1 清除中断
0	DOI	R/W	0	接收数据溢出中断状态位： 发生接收 FIFO 溢出时，DOI 置位。写 1 清除中断

14.3.5 中断使能寄存器 CAN_IMR (偏移：10h)

比特	名称	属性	复位值	描述
7	RSV	R	0	保留
6	ALIM	R/W	0	仲裁丢失中断使能位。使能 CAN 发送器在发送期间丢失仲裁并成为 CAN 接收器时触发中断： 1：使能 ALI 中断 0：禁止 ALI 中断
5	EWIM	R/W	0	错误警告中断使能位。使能当 CAN_SR 寄存器的 BS 或 ES 位状态改变时触发中断： 1：使能 EWI 中断 0：禁止 EWI 中断
4	EPIM	R/W	0	错误被动中断使能位。使能当 CAN 控制器进入或离开被动错误模式时触发中断： 1：使能 EPI 中断 0：禁止 EPI 中断

比特	名称	属性	复位值	描述
3	RIM	R/W	0	接收中断使能位： 1：使能 RI 中断 0：禁止 RI 中断
2	TIM	R/W	0	发送中断使能位： 1：使能 TI 中断 0：禁止 TI 中断
1	BEIM	R/W	0	总线错误中断使能位。使能当 CAN 在发送或接收过程中发生总线错误时触发中断： 1：使能 BEI 中断 0：禁止 BEI 中断
0	DOIM	R/W	0	接收数据溢出中断使能位： 1：使能 DOI 中断 0：禁止 DOI 中断

14.3.6 接收数据计数寄存器 CAN_RMC (偏移：14h)

比特	名称	属性	复位值	描述
7:5	RSV	R	0	保留
4:0	RMC	R	0	接收 FIFO 中 CAN 帧个数。 接收 FIFO 最多可以存储 16 条消息。 以下等式允许计算要存储的最大消息数-RX FIFO: $n = \frac{64}{3 + data\_length\_code}$ 注：此处 data_length_code 至少为 1，若 CAN 数据段长度为 0，data_length_code=1。

14.3.7 总线时序寄存器 CAN_BTR0 (偏移：18h)

此寄存器只能在复位模式写入，可在任何模式读取。

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7:6	SJW	R/W	0	同步跳跃宽度： $t_{SJW}=t_{SCLK}$ x (2 x SJW.1 + SJW.0 + 1) 为了补偿不同 CAN 总线控制器的时钟振荡器之间的相移，必须相应地缩短或延长位周期。 SJW 定义了一个重新同步可以改变一个位周期的最大时钟周期数。再同步过程中，硬件会通过在 PBS1 段内增加 1+SJW 个 t_{SCLK}，或者在 PBS2 段内减少 1~（1+SJW）个 t_{SCLK} 来与接收信号达到同步
5:0	BRP	R/W	0	波特率预分频值: tSCLK=2 x tCLK x (32 x BRP.5 + 16 x BRP.4 + 8 x BRP.3 + 4 x BRP.2 +2 x BRP.1 + BRP.0 +1) 其中，tCLK = 1/fPCLK

14.3.8 总线时序寄存器 CAN_BTR1 (偏移：1Ch)

此寄存器只能在复位模式写入，可在任何模式读取。

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7	SAM	R/W	0	总线电平采样数选择位： 1：采样三次总线电平（适用于中/低速总线） 0：采样一次总线电平（适用于高速总线）
6:4	TSEG2	R/W	0	Time Segment 2 的时钟周期数 t_{TSEG2}= t_{SCLK} x (4 x TSEG2.2 +2 x TSEG2.1 + TSEG2.0 +1)
3:0	TSEG1	R/W	0	Time Segment 1 的时钟周期数 t_{TSEG1}= t_{SCLK} x (8 x TSEG1.3 + 4 x TSEG1.2 +2 x TSEG1.1 + TSEG1.0 +1)

CAN 的位周期结构如下图。其中同步段（SYNC SEG）为 1 x t_{SCLK}，相位缓冲段 1 和 2 长度由 TSEG1 和 TSEG2 决定。

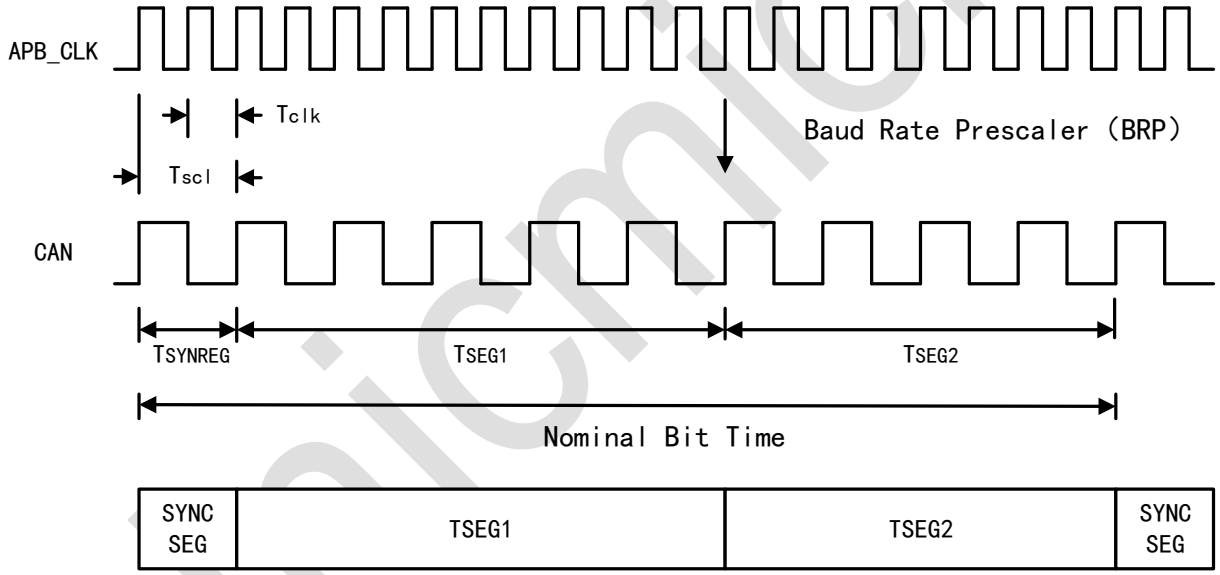


图 14-1：CAN 的位周期结构图

14.3.9 发送缓存寄存器 CAN_TXBUF (偏移：20h)

比特	名称	属性	复位值	描述
31:0	TXBUF	W	0	发送缓存寄存器用于写入要通过 CAN 网络发送的 CAN 帧。 写入该寄存器执行内部写指针的自动递增，通过在 ISR 寄存器中写入 TI 位，可以将写指针复位到发送内存的地址 0h 处

14.3.10 接收缓存寄存器 CAN_RXBUF (偏移：24h)

比特	名称	属性	复位值	描述
31:0	RXBUF	R	0	接收缓存寄存器用于读取从 CAN 网络接收的 CAN 帧。 读取该寄存器将自动递增内部 FIFO 的读取地址指针（读取后递增）

在收到一帧 CAN 数据后，RXBUF 寄存器读到的数据格式如下（databuf 段长度由 DLC 段决定，数量为 0-8Bytes）：

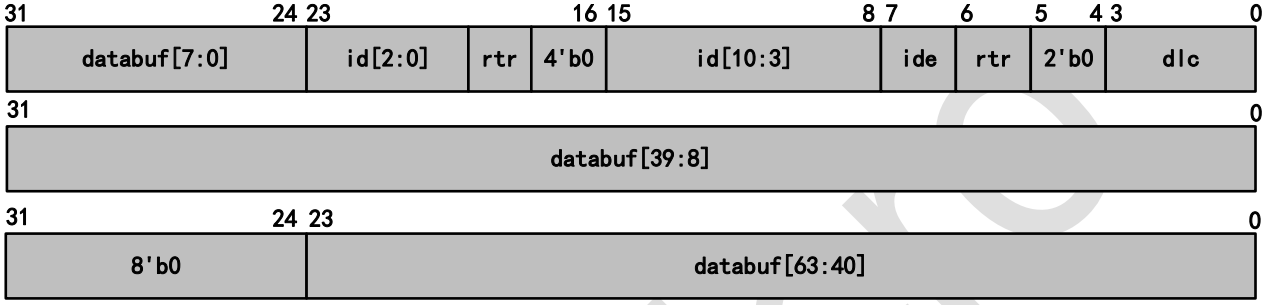


图 14-2：CAN Rx Fifo 11bits ID

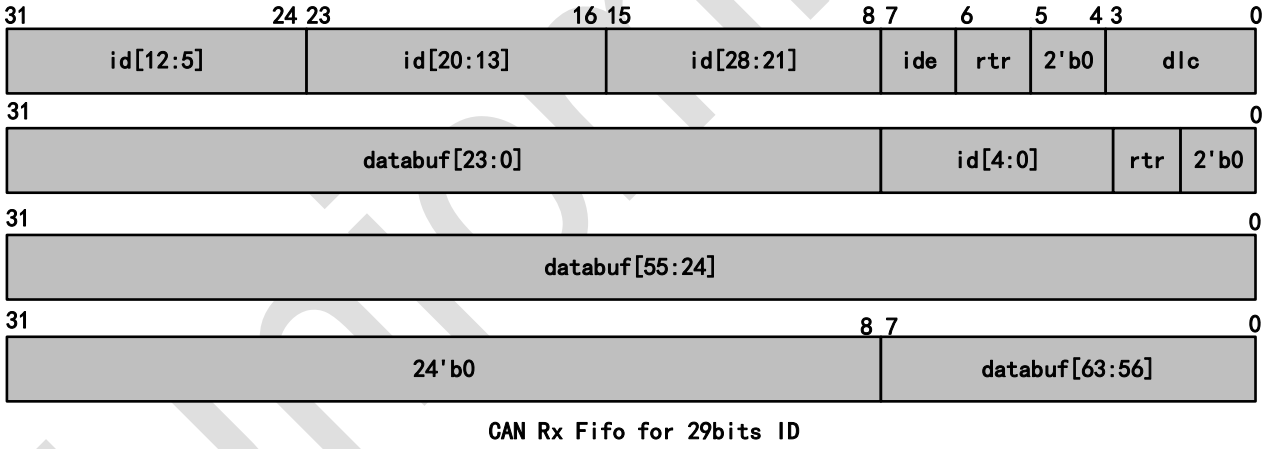


图 14-3：CAN Rx Fifo 29bits ID

在接收完一帧 CAN 数据后，RMC 寄存器计数加 1，此时 CAN 控制器会往 RX FIFO 中逐个写入数据，当写入一个 32 位数据后，RBS 置位。在写入完一帧数据后，RI 标志位置位。

14.3.11 接收过滤匹配寄存器 CAN_ACR(偏移：28h)

只有当接收到的消息的标识符位等于接收过滤器中的预定义位时，CAN 控制器中的接收过滤器才有可能将接收到的消息传递给 RX FIFO。接收过滤器由接收过滤匹配寄存器（ACR3:ACR0）和接收过滤屏蔽寄存器（AMR3:AMR0）组成。模式寄存器的 AFM 位可设置单/双过滤器。在单过滤器配置中，过滤器为 4 字节长。若接收的数据为标准帧模式，可接收到包括仲裁位，RTR 位和数据位的前 2 个字节（数据字节不是必须接收的部分）。所有单个位的比较都必须发出信号，表示成功接收到

数据；若接收的数据为拓展帧格式，可接收到仲裁位和 RTR 位数据。对于格式中没定义的位，过滤器将不进行对比。

双过滤器配置会定义两个长度更短的过滤器。接收到的数据将会跟两个过滤器对比，决定是否应该将数据存入 RX FIFO 中。如果至少一个接收过滤器对比成功，接收到的数据将被存储在 FIFO 中。如果接收到标准格式的帧，第一个过滤器将会对比标准格式的仲裁，RTR 位和第一个数据字节。第二个过滤器只对比标准格式的仲裁和 RTR 位。如果过滤器 1 中不对数据字节过滤，需要把 AMR1 和 AMR3 的低四位设成逻辑 1（不对比此位）。

比特	名称	属性	复位值	描述
31:24	ACR3	R/W	0	接收过滤匹配寄存器包含要接收的消息的仲裁位，而相应的接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位
23:16	ACR2	R/W	0	接收过滤匹配寄存器包含要接收的消息的仲裁位，而相应的接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位
15:8	ACR1	R/W	0	接收过滤匹配寄存器包含要接收的消息的仲裁位，而相应的接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位
7:0	ACR0	R/W	0	接收过滤匹配寄存器包含要接收的消息的仲裁位，而相应的接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位

14.3.12 接收过滤屏蔽寄存器 CAN_AMR(偏移：2Ch)

只有当接收到的消息的标识符位等于接收过滤器中的预定义位时，CAN 控制器中的接收过滤器才有可能将接收到的消息传递给 RX FIFO。接收过滤器由接收过滤匹配寄存器（ACR3:ACR0）和接收过滤屏蔽寄存器（AMR3:AMR0）定义。

比特	名称	属性	复位值	描述
31:24	AMR3	R/W	0	接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位。将相应的位设为 1 表示不对比 ACR 寄存器中相应的位
23:16	AMR2	R/W	0	接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位。将相应的位设为 1 表示不对比 ACR 寄存器中相应的位
15:8	AMR1	R/W	0	接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位。将相应的位设为 1 表示不对比 ACR 寄存器中相应的位
7:0	AMR0	R/W	0	接收过滤屏蔽寄存器定义了将比较哪些位位置和无关位。将相应的位设为 1 表示不对比 ACR 寄存器中相应的位

不同过滤器设置以及不同仲裁长度（标准帧 11 位/拓展帧 29 位）对应的位格式如下图：

- **当设置为单过滤器时：**

ID28 ~ ID21	ID20 ~ ID18	RTR	XXXX	DATA BYTE1	DATA BYTE2
ACR0	ACR1[7:4]		ACR1[3:0] 无效	ACR2[7:0]	ACR3[7:0]
AMR0	AMR1[7:4]		AMR1[3:0] 无效	AMR2[7:0]	AMR3[7:0]

图 14-4：标准帧单滤波器模式

ID28 ~ ID21	ID20 ~ ID13	ID12 ~ ID5	ID4 ~ ID0	RTR	XX
ACR0	ACR1	ACR2	ACR3[7:2]		ACR3[1:0] 无效
AMR0	AMR1	AMR2	AMR3[7:2]		AMR3[1:0] 无效

图 14-5：扩展帧单滤波器模式

- **当设置为双过滤器时：**

标准模式下，当接收到数据后，将会与第一个过滤器的 ID 包括 RTR 位，以及第一个收到的数据字节进行对比，或者与第二个过滤器的 ID 位包括 RTR 位进行对比。

	ID28 ~ ID21	ID20 ~ ID18	RTR	XXXX	DATA BYTE1[7:4]	DATA BYTE1[3:0]	DATA BYTE2
滤波器1	ACR0	ACR1[7:4]			ACR1[3:0]	ACR3[3:0]	
	AMR0	AMR1[7:4]			AMR1[3:0]	AMR3[3:0]	
滤波器2	ACR2	ACR3[7:4]					
	AMR2	AMR3[7:4]					

图 14-6：标准帧双滤波器模式

	ID28 ~ ID21	ID20 ~ ID13	ID12 ~ ID5	ID4 ~ ID0	RTR	XX
滤波器1	ACR0	ACR1				
	AMR0	AMR1				
滤波器2	ACR2	ACR3				
	AMR2	AMR3				

图 14-7：扩展帧双滤波器模式

14.3.13 错误码捕捉寄存器 CAN_ECC (偏移：30h)

ECC 只读寄存器保存有关 CAN 网络上发生的最后总线错误的错误代码。该寄存器是只读的。在确认先前的总线错误之前（通过确认总线错误中断），CAN 内核不会更新该寄存器。

比特	名称	属性	复位值	描述
31:8	RSV	R	-	保留
7	RXWRN	R	0	当 RXERR 计数器大于或等于 96 时置 1
6	TXWRN	R	0	当 TXERR 计数器大于或等于 96 时置 1
5	EDIR	R	0	表示错误发生时数据传输方向： 1：接收 0：发送
4	ACKER	R	0	发生 ACK 错误时置位
3	FRMER	R	0	发生帧格式错误时置位
2	CRCER	R	0	发生 CRC 错误时置位
1	STFER	R	0	发生填充错误时置位
0	BER	R	0	发生位错误时置位

14.3.14 接收错误计数寄存器 CAN_RXERR (偏移：34h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7:0	RXERR	R	0	接收错误计数器的当前值。如果发生总线关闭事件，则 RX 错误计数器将初始化为 0

14.3.15 发送错误计数寄存器 CAN_TXERR (偏移：38h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7:0	TXERR	R	0	发送错误计数器的当前值的低 8 位。如果发生总线关闭事件,则将传输错误计数器初始化为 127,以计算最小协议定义的时间（出现 128 次总线空闲信号）。在这段时间内读取 TXERR 可获得有关总线关闭恢复状态的信息

14.3.16 仲裁丢失捕获寄存器 CAN_ALC (偏移：3Ch)

CAN 控制器能够确定仲裁丢失的确切帧内位置。紧随其后，将产生“仲裁丢失中断”。 此外，在仲裁丢失捕获寄存器中捕获位数。一旦主机控制器读取了该寄存器的内容，就会为下一个仲裁丢失情况激活捕获功能。此功能允许 CAN 监视每个 CAN 总线访问。对于诊断或在系统配置期间，可以确定仲裁不成功的每种情况。

比特	名称	属性	复位值	描述
31:5	RSV	R	0	保留
4:0	ALC	R	0	仲裁失败位置值为 ALC+1

Bits					Decimal Value	Description
ALC4	ALC3	ALC2	ALC1	ALC0		
0	0	0	0	0	00	ID28/10 位仲裁丢失
0	0	0	0	1	01	ID27/9 位仲裁丢失
0	0	0	1	0	02	ID26/8 位仲裁丢失
0	0	0	1	1	03	ID25/7 位仲裁丢失
0	0	1	0	0	04	ID24/6 位仲裁丢失
0	0	1	0	1	05	ID23/5 位仲裁丢失
0	0	1	1	0	06	ID22/4 位仲裁丢失
0	0	1	1	1	07	ID21/3 位仲裁丢失
0	1	0	0	0	08	ID20/2 位仲裁丢失
0	1	0	0	1	09	ID19/1 位仲裁丢失
0	1	0	1	0	10	ID18/0 位仲裁丢失
0	1	0	1	1	11	SRTR/RTR 位仲裁丢失
0	1	1	0	0	12	IDLE 位仲裁丢失
0	1	1	0	1	13	ID17 位仲裁丢失
0	1	1	1	0	14	ID16 位仲裁丢失
0	1	1	1	1	15	ID15 位仲裁丢失
1	0	0	0	0	16	ID14 位仲裁丢失
1	0	0	0	1	17	ID13 位仲裁丢失
1	0	0	1	0	18	ID12 位仲裁丢失
1	0	0	1	1	19	ID11 位仲裁丢失
1	0	1	0	0	20	ID10 位仲裁丢失
1	0	1	0	1	21	ID9 位仲裁丢失
1	0	1	1	0	22	ID8 位仲裁丢失
1	0	1	1	1	23	ID7 位仲裁丢失
1	1	0	0	0	24	ID6 位仲裁丢失
1	1	0	0	1	25	ID5 位仲裁丢失
1	1	0	1	0	26	ID4 位仲裁丢失
1	1	0	1	1	27	ID3 位仲裁丢失
1	1	1	0	0	28	ID2 位仲裁丢失
1	1	1	0	1	29	ID1 位仲裁丢失
1	1	1	1	0	30	ID0 位仲裁丢失
1	1	1	1	1	31	RTR 位仲裁丢失

14.3.17 接收缓存基地址设置寄存器 CAN_RXADDR (偏移：40h)

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11:2	RX_ADDR_BASE	R/W	0	设置 RX fifo 使用 SRAM 中的地址的偏移地址（RX fifo 是基地址为 0x20001000（+偏移地址）后的 64bytes 空间，读取 fifo 数据时可从 0x20001000（+偏移地址）开始读取，软件在编译时需注意 SRAM 空间在此地址处的内存划分，避免与其他数据存放有冲突。偏移地址为 {RX_ADDR_BASE[9:0], 2'b0}）
1:0	RSV	R	0	保留

14.4 使用流程

14.4.1 发送 CAN 数据帧

1. 配置开启 CAN 模块时钟与释放复位 PERI_CLKEN / PERI_RESET。
2. 配置 CAN_TX、CAN_RX 功能管脚复用，CAN_RX 对应管脚需配置 PAD_IEx 寄存器输入使能。
3. 配置总线时序寄存器 CAN_BTR 和 CAN_BTR1，设置 can 的通信速率。
4. 配置中断状态寄存器 CAN_ISR[6:0]，清除错误标志位/中断标志位。
5. 配置 CAN_RXADDR[11:2]寄存器设置接收缓存基地址。
6. 配置中断使能寄存器 CAN_IMR[2]为 1，使能 TI 中断（可选）。
7. 配置模式寄存器 CAN_MR[1]为 0，进入正常模式。
8. 配置发送缓存寄存器 CAN_TXBUF[31:0]，根据定义的格式写入 CAN 数据帧内容，按发送的先后顺序写入，每次写入 32 位数据。
9. 配置指令寄存器 CAN_CMR[2]为 1，启动发送。
10. 等待状态寄存器 CAN_SR[5]置 1 后（若使能 TI 中断，此处可选择等待 TI 中断触发），数据发送完毕。

14.4.2 接收 CAN 数据帧

1. 配置开启 CAN 模块时钟与释放复位 PERI_CLKEN / PERI_RESET。
2. 配置 CAN_TX、CAN_RX 功能管脚复用，CAN_RX 对应管脚需配置 PAD_IEx 寄存器输入使能。
3. 配置总线时序寄存器 CAN_BTR0/CAN_BTR1。
4. 配置中断状态寄存器 CAN_ISR[6:0]，清除错误标志位/中断标志位。
5. 配置 CAN_RXADDR[11:2]寄存器设置接收缓存基地址。
6. 配置中断使能寄存器 CAN_IMR[3]，使能 RI 中断（可选）。

7. 设置接收过滤器配置，若使用单过滤器，CAN_MR[0]置 1。CAN_ACR 寄存器配置用户需要过滤筛选的内容，CAN_AMR[31:0]选择需要与 CAN_ACR[31:0]进行对比的位。若不需要进行对，CAN_AMR[31:0]设为 0xFFFFFFFF。
8. 配置模式寄存器 CAN_MR[1]为 0，进入正常模式。
9. 等待状态寄存器 CAN_SR[7]置 1 后（若使能 RI 中断，此处可选择等待 RI 中断触发），读取接收缓存寄存器 CAN_RXBUF 数据，多次读取直到取出所有数据。

14.4.3 CAN 速率计算

CAN 的波特率可以用以下四个变量可以算出：

A．最小时间段：Tscl；

B．时间段 1：tesg1；

C．时间段 2：tesg2；

D．同步跳转宽度：SJW。

其中最小时间段由 CAN 控制器的时钟频率以及分频决定。

tesg1=TS1+1, tesg2=TS2+1; prescaler=2(BRP+1)

$$\text{BitRate} = \frac{\text{ClockFrequency}}{\text{prescaler x } (\text{tesg1} + \text{tesg2} + 1)}$$

例如：

速率可以选择为 1M/500k/250k/125k bps

APB 时钟=PCLK=48Mhz,

CAN 波特率 BitRate = Fpclk/(2*((BRP+1)*(TS1+TS2+3))), 默认约定:TS1>=TS2

- 设波特率为 1M 的参数：
  设 BRP=2(6 分频), BitRate = 1M = 48M/(2((2+1)*(TS1+TS2+3))),所以可以设置 TS1=3,TS2=2
- 设波特率为 500K 的参数：
  设 BRP=5(12 分频), BitRate = 0.5M = 48M/(2((5+1)*(TS1+TS2+3))),所以可以设置 TS1=3,TS2=2
- 设波特率为 250K 的参数：
  设 BRP=11(24 分频), BitRate = 0.25M = 48M/(2((11+1)*(TS1+TS2+3))),所以可以设置 TS1=3,TS2=2
- 设波特率为 125K 的参数：
  设 BRP=23(48 分频), BitRate = 0.125M = 48M/(2((23+1)*(TS1+TS2+3))),所以可以设置 TS1=3,TS2=2

15 LIN

15.1 概述

LIN(Local Interconnect Network) 控制器可用于分布式汽车应用中机电一体化节点的控制，本模块支持 LIN 总线上的主节点和从节点的连接。

15.2 主要特性

- 支持 LIN 1.3 和 LIN 2.0 标准
- 支持 LIN 总线上的主节点或从节点配置
- 可配置的波特率产生器
- 数据传输速率高达 20kbit/s
- 可发送多达 256 bytes 数据
- 从节点自同步波特率
- 硬件自动计算/校对标识符和数据的奇偶校验，Checksum 自动计算/校对
- 多种传输错误检测
- 帧间隔模式：主节点自动为调度的帧分配时隙
- 可产生唤醒信号

15.3 模块描述

15.3.1 波特率发生器

波特率发生器基于一个 16 位的分频器，该分频器使用 LIN_BRGR 寄存器的 CD 字段进行编程。如果将 0 写入 CD，波特率发生器不会产生任何时钟。如果将 1 写入 CD，则分频器将被绕过并变为 inactive 状态。

- LIN 主节点：波特率在 LIN_BRGR 寄存器中配置
- LIN 从节点：初始波特率在 LIN_BRGR 寄存器中配置。此配置在写入 BRGR 时自动复制到 LIN_BRR 寄存器。在同步过程之后，波特率在 LIN_BRR 寄存器中更新。

所选源时钟首先被 CD 分频（LIN_BRGR 寄存器）。 产生的时钟作为采样时钟提供给接收器，然后除以 16 或 8，具体取决于 MR 中 OVER 位的值。如果 OVER=1，则接收器采样比波特率时钟高 8 倍。 如果 OVER=0，则以 16 倍波特率时钟执行采样。如果 FP 不为 0，则激活小数部分。分辨率是时钟分频器的八分之一。

波特率按以下公式计算：

$$Baudrate = \frac{Fpclk}{\{8 * (2 - \mathrm{Over}) * (\mathrm{CD} + \mathrm{FP}/8)\}}$$

15.3.2 Break 发送

用户可以在 TXD 线上产生一个 Break 信号。通过将 LIN_CR 寄存器的 STTBRK 写 1 来发送 Break。一旦执行了 STTBRK 指令，直到 Break 条件发送结束，所有再次写 STTBRK 的指令都被视为无效。

通过将 LIN_CR 寄存器的 STPBRK 位写 1 来清除 Break 信号。如果在很短的 Break 持续时间（一个字符，包括开始位、数据位、奇偶校验位和停止位）结束之前请求 STPBRK，发送器会确保 Break 条件发送完成。

发送器将 Break 条件视为字符，仅当 LIN_CSR 寄存器中的 TXRDY 位为 1 时才会执行 STTBRK 和 STPBRK 命令。在 Break 条件开始执行后，跟发送字符的情况一样，TXRDY 和 TXEMPTY 位清除。

同时将 LIN_CR 寄存器的 STTBRK 和 STPBRK 位写为 1 会导致不可预知的结果。如果没有先发送 STTBRK 命令，请求的 STPBRK 命令都将被忽略。当中断挂起但未开始时，写入 LIN_THR 寄存器的字节将被忽略。

在 Break 信号发送以后，发送器将 TXD 线返回高电平至少 12 个位时间（BIT TIME）， 因此，发送器确保远程接收器正确检测到中断的结束和下一个字符的开始。如果发送器时间保护值设置为高于 12，则 TXD 线在时间保护期间保持高电平。在此期间保持 TXD 线后，发送器恢复正常操作。

15.3.3 LIN 报头发送过程（主节点配置）

所有 LIN 帧都以主节点发送的报头开头，由同步间隔场（Break）、同步场（Sync）和标识符场（Identifier）组成。因此，在主节点配置中，帧处理从发送报头开始。

一旦 ID 标识符写入 LIN 标识符寄存器 LIN_LINIR，就会发送报头。此时标志 TXRDY 变为 0。

Break 字段、Sync 字段和 Identifier 字段将会一个接一个地自动发送。

Break 字段由 13 个显性位和 1 个隐性位组成，Sync 字段是字符 0x55，Identifier 对应于写入 LIN_LINIR 寄存器中的字符。ID 标识符奇偶校验位可以自动计算和发送。

当 Identifier 字符被传送到发送器的移位寄存器时，标志 TXRDY 置位。

Break 字段一经发送，LIN_CSR 寄存器中的标志位 LINBK 置 1。同样，一旦发送 Identifier 字段，LIN_CSR 寄存器中的标志位 LINID 置 1。这些标志位可通过向 LIN_CR 寄存器中的 RSTSTA 位写入 1 来重置。

15.3.4 LIN 报头接收过程（从节点配置）

LIN 控制器在实际波特率下使用 11 个标称位时间的 Break 检测阈值。如果在总线上检测到 11 个连续的显性位，LIN 控制器就会认为是一个 Break 字段。只要没有检测到 Break 字段，LIN 控制器就会保持空闲，并且不考虑接收到的数据。

当检测到 Break 字段时，LIN_CSR 寄存器中的标志 LINBK 设置为 1，此时 LIN 控制器期望 SYNC 字段字符为 0x55，该字段用于更新实际波特率以保持同步。如果接收到的 SYNC 字符不是 0x55，则会生成不一致的 SYNC 字段错误。

收到 SYNC 字段后，LIN 控制器期望收到 ID 字段。当接收到 ID 字段时，LIN_CSR 寄存器中的标志位 LINID 设置为 1。此时 LIN_LINIR 寄存器中的字段 IDCHR 用接收到的字符进行更新。ID 标识符奇偶校验位可以自动计算和检查。

如果在报头最大长度 THeader_Maximum 给定的时间内没有完全接收到报头，则 LIN_CSR 寄存器中的错误标志 LINHTE 置 1。

标志位 LINID、LINBK 和 LINHTE 可通过向 LIN_CR 寄存器中的 RSTSTA 位写入 1 来复位。

15.3.5 LIN 错误

- **位错误**

当 LIN 控制器正在发送并且 Tx 线上的发送值与 Rx 线上的采样值不同时，会在主从节点生成此错误。如果检测到位错误，会在下一个字节中止传输。此错误由 LIN_CSR 寄存器中的 LINBE 位说明。

- **不一致同步场错误**

如果接收到的同步字段字符不是 0x55，则在从节点生成此错误。此错误由 LIN_CSR 寄存器中的 LINISFE 位说明。

- **标识符奇偶校验位错误**

在从节点接收中，如果标识符的奇偶校验错误，则会产生此错误。仅当启用奇偶校验功能 (PARDIS = 0) 时才会生成此错误。此错误由 LIN_CSR 寄存器中的 LINIPE 位说明。

- **Checksum 错误**

如果接收到的校验和错误，则会在主从节点中生成此错误。仅当启用校验和功能 (CHKDIS = 0) 时，该标志才能设置为“1”。此错误由 LIN_CSR 寄存器中的 LINCE 位说明。

- **从节点无响应错误**

当 LIN 控制器期望来自另一个节点的响应 (NACT = SUBSCRIBE) 但在消息帧的最大长度 TFrame_Maximum 给定的时间内总线上没有出现有效消息时，会在主从节点中生成此错误。如果 LIN 控制器不期望任何消息（NACT = PUBLISH 或 NACT = IGNORE），则禁用此错误。此错误由 LIN_CSR 寄存器中的 LINSNRE 位说明。

- **同步容差错误**

如果在时钟同步过程之后，计算出的波特率与初始波特率相比的偏差大于最大容差 FTol_Unsynch (±15%)，则在从节点中生成此错误。此错误由 LIN_CSR 寄存器中的 LINSTE 位说明。

- **报头超时错误**

在从节点中，如果在 Header 的最大长度 THeader_Maximum 给定的时间内没有完全接收到 Header，则会产生此错误。此错误由 LIN_CSR 寄存器中的 LINHTE 位说明。

- **接收溢出错误**

当一个字符接收完成后，它被传送到 LIN_RHR 寄存器并且 LIN_CSR 寄存器中的 RXRDY 位上升。如果在 RXRDY 置位时又完成了一个字符的接收，则会产生此错误。最后一个接收到的字符将被传送到 LIN_RHR 寄存器并覆盖前一个字符。此错误由 LIN_CSR 寄存器中的 OVRE 位说明。

15.4 寄存器描述

寄存器基地址：0x4000_7800

表 15-1：LIN 寄存器列表

偏置	名称	描述
0x00	LIN_CR	控制寄存器
0x04	LIN_MR1	模式寄存器
0x08	LIN_IER	中断使能寄存器
0x0c	LIN_CSR	状态寄存器
0x10	LIN_THR	发送设置寄存器
0x14	LIN_RHR	接收数据寄存器
0x18	LIN_BRGR	波特率产生寄存器
0x1c	LIN_RTOR	接收超时设置寄存器
0x20	LIN_MR2	LIN 模式寄存器
0x24	LIN_IR	LIN ID 标识寄存器
0x28	LIN_BRR	LIN 波特率寄存器
0x3c	LIN_VERSION	版本寄存器

15.4.1 控制寄存器 LIN_CR (偏移：00h)

比特	名称	属性	复位值	描述
31:26	RSV	W	-	保留
25	LINWKUP	W	-	发送 LIN 总线唤醒信号位： 1：在 LIN 总线上发送一个唤醒信号 0：无效

比特	名称	属性	复位值	描述
24	LINABT	W	-	LIN 传输中止位： 1：中止 LIN 总线传输 0：无效
23:20	RSV	W	-	保留
19	RETTO	W	-	启动 Time-out 位： 1：重启 Time-out 超时 0：无效
18	STTTO	W	-	停止 Time_out 位： 1：在 Time-out 计数器开始计数前开始等待字符 0：无效
17:11	RSV	W	-	保留
10	RSTSTA	W	-	状态重置位： 1：复位 CSR 寄存器里的状态位（PARE, FRAME, OVRE, LINBE, LINISFE, LINIPE, LINCE, LINSNRE, LINSTE, LINHTE, LINID, LINTC, LINBK） 0：无效
9	RSTRX	W	-	接收器重置位： 1：复位接收器 0：无效
8	RSTTX	W	-	发射器重置位： 1：复位发射器 0：无效
7:4	RSV	W	-	保留
3	RXDIS	W	-	接收器禁止位： 1：关闭接收器 0：无效
2	RXEN	W	-	接收器使能位： 1：RXDIS 为 0 时，使能接收器 0：无效
1	TXDIS	W	-	发射器禁止位： 1：关闭发射器 0：无效
0	TXEN	W	-	发射器使能位： 1：TXDIS 为 0 时，使能发射器 0：无效

15.4.2 模式寄存器 LIN_MR1 (偏移：04h)

比特	名称	属性	复位值	描述
31:25	RSV	R	-	保留
24	WACKUP_EN	R/W	-	唤醒功能使能位： 使能此位和 WAKEUPIEN 以后，在 Deepsleep 模式下，检测到 rx 线上的低电平超过 120us，就会唤醒整个系统
23:16	RSV	R	-	保留

比特	名称	属性	复位值	描述
15	FILTER	R/W	-	接收滤波选择位： 1：使用三采样滤波器过滤接收线上的毛刺，可滤除长度小于 3 个 PCLK 时钟周期的毛刺 0：不进行滤波
14	OVER	R/W	-	过采样模式选择位： 1：8x 过采样模式 0：16x 过采样模式
13	MSBF	R/W	0	Bit Order 位： 1：先发送/接收 MSB 0：先发送/接收 LSB
12:2	RSV	R	-	保留
1:0	LIN_MODE	R/W	0	LIN 主从模式选择位： 2'h2：LIN 主节点模式 2'h3：LIN 从节点模式 其它值，保留；

15.4.3 中断使能寄存器 LIN_IER (偏移：08h)

比特	名称	属性	复位值	描述
31:19	RSV	-	-	保留
18	WAKEUPIEN	R/W	0	DeepSleep 低功耗模式 LIN 唤醒中断使能位： 1：使能 DeepSleep 下 LIN RX 低电平唤醒 0：禁止 DeepSleep 下 LIN RX 低电平唤醒
17	LINHTE	R/W	0	LIN 报头超时错误中断使能位（LIN Header Timeout Error）： 1：使能 LIN 报头超时错误中断 0：禁止 LIN 报头超时错误中断
16	LINSTE	R/W	0	同步容差错误中断使能位（LIN Sync Tolerance Error）： 1：使能同步容差错误中断 0：禁止同步容差错误中断
15	LINSNRE	R/W	0	从节点无响应错误中断使能位（LIN Slave Not Responding Error）： 1：使能从节点无响应错误中断 0：禁止从节点无响应错误中断
14	LINCE	R/W	0	Checksum 错误中断使能位（LIN Checksum Error）： 1：使能 Checksum 错误中断 0：禁止 Checksum 错误中断
13	LINIPE	R/W	0	LIN ID 奇偶校验位错误中断使能位（LIN Identifier Parity Error）： 1：使能 LIN ID 奇偶校验位错误中断 0：禁止 LIN ID 奇偶校验位错误中断
12	LINISFE	R/W	0	不一致同步错误中断使能位（LIN Inconsistent Sync Field Error）： 1：使能不一致同步错误中断 0：禁止不一致同步错误中断

比特	名称	属性	复位值	描述
11	LINBE	R/W	0	LIN 位错误中断使能位（LIN Bit Error）： 1：使能 LIN 位错误中断 0：禁止 LIN 位错误中断
10	LINTC	R/W	0	LIN 传输完成中断使能位（LIN Transfer Completed）： 1：使能 LIN 传输完成中断 0：禁止 LIN 传输完成中断
9	LINID	R/W	0	ID 已发送/已接收中断使能位（LIN Identifier Send or LIN Identifier Received）： 1：使能 ID 已发送/已接收中断 0：禁止 ID 已发送/已接收中断
8	LINBK	R/W	0	Break 已发送/已接收中断使能位（LIN Break Sent or LIN Break Received）： 1：使能 Break 已发送/已接收中断 0：禁止 Break 已发送/已接收中断
7	RSV	-	0	保留
6	TXEMPTY	R/W	0	发送器空中断使能位（Transmitter empty）： 1：使能发送器空中断 0：禁止发送器空中断
5	TIMEOUT	R/W	0	接收器超时中断使能位（Receiver time-out）： 1：使能接收器超时中断 0：禁止接收器超时中断
4	RSV	-	0	保留
3	FRAME	R/W	0	帧错误中断使能位（Framing error）： 1：使能帧错误中断 0：禁止帧错误中断
2	OVRE	R/W	0	溢出错误中断使能位（Overrun error）： 1：使能溢出错误中断 0：禁止溢出错误中断
1	RXRDY	R/W	0	接收器就绪中断使能位（Receiver ready）： 1：使能接收器就绪中断 0：禁止接收器就绪中断
0	TXRDY	R/W	0	发送器就绪中断使能位（Transmitter ready）： 1：使能发送器就绪中断 0：禁止发送器就绪中断

15.4.4 状态寄存器 LIN_CSR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:25	RSV	R	-	保留
24	LIN_RX	R	0	LIN 接收线实时值（LIN_RX_DATA）
23:19	RSV	R	-	保留
18	WACKUP	R	0	唤醒状态位： 1：DeepSleep 模式下 LIN 被唤醒 0：DeepSleep 模式下 LIN 未唤醒 此位写 1 清 0

比特	名称	属性	复位值	描述
17	LINHTE	R/W	0	LIN 报头超时错误位（LIN Header Timeout Error）： 1：自上次 RSTSTA 以来检测到 LIN 报头超时错误 0：自上次 RSTSTA 以来未检测到 LIN 报头超时错误 此位写 1 清 0
16	LINSTE	R/W	0	同步容差错误位（LIN Sync Tolerance Error）： 1：自上次 RSTSTA 以来检测到 LIN 同步容差错误 0：自上次 RSTSTA 以来未检测到 LIN 同步容差错误 此位写 1 清 0
15	LINSNRE	R/W	0	从节点无响应错误位（LIN Slave Not Responding Error）： 1：自上次 RSTSTA 以来检测到 LIN 从节点无响应错误 0：自上次 RSTSTA 以来未检测到 LIN 从节点无响应错误 此位写 1 清 0
14	LINCE	R/W	0	Checksum 错误位（LIN Checksum Error）： 1：自上次 RSTSTA 以来检测到 LIN Checksum 错误 0：自上次 RSTSTA 以来未检测到 LIN Checksum 错误 此位写 1 清 0
13	LINIPE	R/W	0	LIN ID 奇偶校验位错误（LIN Identifier Parity Error）： 1：自上次 RSTSTA 以来检测到 LIN 标识符奇偶校验错误 0：自上次 RSTSTA 以来未检测到 LIN 标识符奇偶校验错误 此位写 1 清 0
12	LINISFE	R/W	0	同步错误（LIN Inconsistent Sync Field Error）： 1：LIN 被配置为从节点，并且自上次 RSTSTA 以来检测到 LIN 不一致同步字段错误 0：自上次 RSTSTA 以来未检测到 LIN 不一致同步字段错误 此位写 1 清 0
11	LINBE	R/W	0	LIN 位错误（LIN Bit Error）： 1：自上次 RSTSTA 以来检测到位错误 0：自上次 RSTSTA 以来未检测到位错误 此位写 1 清 0
10	LINTC	R/W	0	LIN 传输完成位（LIN Transfer Completed）： 1：自上次 RSTSTA 以来已完成 LIN 传输 0：空闲或正在进行 LIN 传输 此位写 1 清 0

比特	名称	属性	复位值	描述
9	LINID	R/W	0	ID 已发送/已接收位（LIN Identifier Send or LIN Identifier Received）： 1：自上次 RSTSTA 以来至少发送/接收了一个 LIN 标识符 0：自上次 RSTSTA 以来未发送/接收任何 LIN 标识符 此位写 1 清 0
8	LINBK	R/W	0	Break 已发送/已接收位（LIN Break Sent or LIN Break Received）： 1：自上次 RSTSTA 以来已发送/收到至少一个 LIN Break 0：自上次 RSTSTA 以来未发送/收到任何 LIN Break 此位写 1 清 0
7	RSV	R	0	保留
6	TXEMPTY	R/W	0	发送器空标志位（Transmitter empty）： 1：THR 寄存器中没有字符，发送移位寄存器中也没有字符 0：THR 寄存器或发送移位寄存器中有字符，或者发送器被禁用 此位写 1 清 0
5	TIMEOUT	R/W	0	接收器超时标志位（Receiver time-out）： 1：自上次启动超时命令以来已超时 0：自上次启动超时命令以来没有超时或超时寄存器为 0 此位写 1 清 0
4	RSV	R	0	保留
3	FRAME	R/W	0	帧错误位（Framing error）： 1：自上次 RSTSTA 以来，至少检测到一个停止位为低 0：自上次 RSTSTA 以来未检测到低停止位 此位写 1 清 0
2	OVRE	R/W	0	溢出错误位（Overrun error）： 1：自上次 RSTSTA 以来至少发生过一次溢出错误 0：自上次 RSTSTA 以来未发生溢出错误 此位写 1 清 0
1	RXRDY	R/W	0	接收器就绪位（Receiver ready）： 1：已收到至少一个完整字符，并且尚未读取 RHR 寄存器 0：自上次读取 RHR 寄存器或接收器被禁用后，未收到完整字符。如果在接收器禁用时接收到字符，则在接收器使能后 RXRDY 变为 1 此位写 1 清 0
0	TXRDY	R/W	0	发送器就绪位（Transmitter ready）： 1：THR 寄存器中没有字符 0：THR 寄存器中有一个字符等待传输到发送移位寄存器。一旦发射器使能，TXRDY 变为 1 此位写 1 清 0

15.4.5 发送设置寄存器 LIN_THR (偏移：10h)

比特	名称	属性	复位值	描述
31:8	RSV	W	0	保留
7:0	TXCHR	W	0	发送数据缓存： 如果未设置 TXRDY，发送缓存则在当前字符之后的下一个字符传输

15.4.6 接收数据寄存器 LIN_RHR (偏移：14h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7:0	RXCHR	R	0	接收数据缓存： 若 RXRDY 置位，此处缓存上一个收到的字符

15.4.7 波特率产生寄存器 LIN_BRGR (偏移：18h)

比特	名称	属性	复位值	描述
31:19	RSV	R	0	保留
18:16	LINFP	R/W	0	1-7：由 FP*1/8 定义 0：禁止小数分频
15:0	LINCD	R/W	0	波特率分频值

CD	SYNC=0		SYNC=1
	OVER=0	OVER=1	
0	波特率时钟禁止		
1-65535	BR=clk/(16*CD)	BR=clk/(8*CD)	BR=clk/CD

15.4.8 接收超时设置寄存器 LIN_RTOR (偏移：1Ch)

比特	名称	属性	复位值	描述
31:17	RSV	R	0	保留
16:0	TO	R/W	0	接收器 Time-out 值： 0：禁止接收器 Time-out 1-131071：使能接收器 Time-out，Time-out 延迟为 TO*Bit Period

15.4.9 LIN 模式寄存器 LIN_MR2 (偏移：20h)

比特	名称	属性	复位值	描述
31:23	RSV	R	0	保留

比特	名称	属性	复位值	描述
22:20	BREAK_LEN	R/W	0	同步间隔长度： 0:13 比特时间 1:14 比特时间 2:15 比特时间 3:16 比特时间 … 7:20 比特时间
19:17	RSV	R	0	保留
16	SYNCDIS	R/W	0	LIN 同步禁止位： 0：在 LIN 从节点配置中执行同步过程 1：不在 LIN 从节点配置中执行同步过程
15:8	DLC	R/W	8’h0	数据长度控制： 0-255：若 DLM = 0，则此位定义响应数据长度，在这种情况下，响应数据长度等于 DLC+1 字节
7	WKUPTYP	R/W	0	唤醒信号类型： 1：设置 LINWKUP 位会发送一个 LIN 1.3 唤醒信号 0：设置 LINWKUP 位会发送一个 LIN 2.0 唤醒信号
6	FSDIS	R/W	0	帧间隔模式禁止位： 1：禁止帧间隔模式 0：使能帧间隔模式
5	DLM	R/W	0	数据长度模式： 1：响应数据长度由 Identifier[5:4]定义 0：响应数据长度由 DLC 字段定义
4	CHKTYP	R/W	0	Checksum 类型选择位： 1：LIN 1.3 “Classic” Checksum 0：LIN 2.0 “Enhanced” Checksum
3	CHKDIS	R/W	0	Checksum 禁止位： 1：不计算/发送 Checksum，不检查 Checksum 0：主节点发送模式下，硬件自动计算/发送 Checksum；主从节点接收模式下，硬件自动对 Checksum 进行检查
2	PARDIS	R/W	0	Parity 禁止位： 1：不计算/发送奇偶校验位，不检查检验位 0：主节点发送模式下，硬件自动计算/发送检验位；主从节点接收模式下，硬件自动对校验位进行检查
1:0	NACT	R/W	2’b00	LIN 节点行为选择位： 2’b00：节点发送响应（PUBLISH） 2’b01：节点接送响应（SUBSCRIBE） 2’b10：节点不发送也不接收响应（IGNORE） 2’b11：无效

15.4.10 LIN ID 标识寄存器 LIN_IR (偏移：24h)

比特	名称	属性	复位值	描述
31:8	RSV	R	0	保留
7:0	IDCHR	R/W	2'b00	Identifier 标识： 在 Master 模式下，此位可读可写，可写入要发送的 ID 字符；在 Slave 模式下，此位只读，缓存从节点接收的 ID 字符。

15.4.11 LIN 波特率寄存器 LIN_BRR (偏移：28h)

比特	名称	属性	复位值	描述
31:19	RSV	R	0	保留
18:16	LINFP	R	0	波特率小数分频
15:0	LINCD	R	0	波特率分频值

15.4.12 版本寄存器 LIN_VERSION (偏移：3Ch)

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11:0	VERSION	R	12'h201	硬件版本号

15.5 LIN 使用流程

15.5.1 主节点配置

1. 开启 LIN 时钟，释放复位，LIN 功能管脚复用。
2. 写 LIN_CR[0]和 LIN_CR[2]使能接收器和发射器。
3. 写 LIN_MR1[1:0]选择 LIN 主节点模式。
4. 配置 LIN_BRGR[15:0]和 LIN_BRGR[18:16]，配置通信波特率。
5. 写 LIN_MR2[1:0]，LIN_MR2[2]，LIN_MR2[3]，LIN_MR2[4]，LIN_MR2[6]和 LIN_MR2[15:8]来配置帧传输参数。
6. 查询 LIN_CSR[0]是否为 1。
7. LIN_IR[7:0]写入 ID 值启动帧头段发送。

接下来的操作取决于 NACT 的配置：

CASE 1：NACT=PUBLISH，主节点发送响应

1. 等待 LIN_CSR[0]从 0 变成 1

2. LIN_THR[7:0]写入要发送的字符
3. 若数据还没发送完，重复前 2 个步骤
4. 等待 LIN_CSR[10]从 0 变成 1
5. 检查 LIN 错误

CASE 2：NACT=SUBSCRIBE，主节点接收响应

1. 等待 LIN_CSR[1]从 0 变成 1
2. 从 LIN_RHR[7:0]读出接收的数据
3. 若数据还没接收完，重复前 2 个步骤
4. 等待 LIN_CSR[10]从 0 变成 1
5. 检查 LIN 错误

CASE 3：NACT=IGNORE，主节点不关心响应

1. 等待 LIN_CSR[10]从 0 变成 1
2. 检查 LIN 错误

15.5.2 从节点配置

1. 开启 LIN 时钟，释放复位，LIN 功能管脚复用。
2. 写 LIN_CR[0]和 LIN_CR[2]使能接收器和发射器。
3. 写 LIN_MR1[1:0]选择 LIN 从节点模式。
4. 配置 LIN_BRGR[15:0]和 LIN_BRGR[18:16]，配置通信波特率。
5. 等待 LIN_CSR[9]置位。
6. 检查 LIN_CSR[12]和 LIN_CSR[13]错误。
7. 从 LIN_RHR 寄存器读出 IDCHR 值。
8. 写 LIN_MR2[1:0]，LIN_MR2[2]，LIN_MR2[3]，LIN_MR2[4]，LIN_MR2[6]和 LIN_MR2[15:8]来配置帧传输参数。

   注：如果此帧的 NACT 配置为 PUBLISH，即使该字段已正确配置，LIN_MR2 寄存器也必须写入 NACT=PUBLISH，以便设置 TXREADY 标志和相应的写传输请求。

接下来的操作取决于 NACT 的配置：

CASE 1：NACT=PUBLISH，从节点发送响应

1. 等待 LIN_CSR[0]从 0 变成 1。
2. LIN_THR[7:0]写入要发送的字符。

3. 若数据还没发送完，重复前 2 个步骤。
4. 等待 LIN_CSR[10]从 0 变成 1。
5. 检查 LIN 错误。

CASE 2：NACT=SUBSCRIBE，从节点接收响应

1. 等待 LIN_CSR[1]从 0 变成 1。
2. 从 LIN_RHR[7:0]读出接收的数据。
3. 若数据还没接收完，重复前 2 个步骤。
4. 等待 LIN_CSR[10]从 0 变成 1。
5. 检查 LIN 错误。

CASE 3：NACT=IGNORE，从节点不关心响应

1. 等待 LIN_CSR[10]从 0 变成 1。
2. 检查 LIN 错误。

16 ATIMER

16.1 概述

高级定时器 ATIMER 包含一个 16bit 自动重载计数器及一个可编程预分频器，可以支持多种应用，包括输入捕捉、输出比较、PWM、带死区插入的互补 PWM 等。

16.2 主要特性

- 16 位向上、向下、向上/下计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 4 个独立通道可用于输入捕捉、输出比较、PWM、单脉冲输出
- 可编程死区插入的互补输出
- 支持定时器间的级联
- 重复计数器，支持定时器多个循环后更新状态
- 两路刹车引脚输入、比较器刹车、LVD 刹车，刹车信号滤波和极性选择，刹车信号组合配置
- 支持在以下事件发生时产生中断或 DMA 事件：
  - 计数器上/下溢出，计数器初始化（软件或硬件 trigger）
  - Trigger 事件（计数器启动、停止、初始化、内外部触发）
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持增量正交编码器和霍尔传感器

16.3 功能描述

16.3.1 定时单元

定时单元由一个 16 位计数器和自动重载寄存器组成。计数器可以向上、向下或双向计数。计数时钟可以通过 16 位预分频器对时钟进行分频后得到。

计数器、自动重载寄存器预分频寄存器都可以由软件改写或读取，即使在计数器正在运行时也是如此。

定时单元包含如下寄存器：

- 计数器（ATIM_CNT）
- 预分频寄存器（ATIM_PSC）
- 自动重载寄存器（ATIM_ARR）
- 重复计数寄存器（ATIM_RCR）

ARR 包含预装载功能，该功能通过 ARPE（Auto Reload Preload Enable）寄存器控制。当 ARPE=0 时，对 ARR 寄存器执行写入，写入数据将直接传入到影子寄存器；当 ARPE=1 时，对 ARR 寄存器执行写入的数据在 update event（ATIM_CNT 上溢出或者下溢出）发生时，传送到影子寄存器。软件也可以通过寄存器操作主动触发 ARR 更新（UEV）。

ATIM_CNT 工作时钟由 ATIM_PSC 产生的分频时钟驱动，只有在计数器使能寄存器（CEN）置位时，CNT 才开始计数。当 CNT=ARR 时，本轮计数结束，发送 update event。

ATIM_PSC 是一个同步预分频器，能够对时钟进行 1~65536 分频。PSC 寄存器同样被缓存，改写 PSC 实际不改写影子寄存器，只有当新的 update event 到来时，才会从 PSC 更新至影子寄存器。因此在 CNT 计数过程中，软件可以实时改写 PSC，而新的预分频比将在下一更新事件发生时被采用。

16.3.2 定时器工作模式

定时器支持向上计数、向下计数和中心计数模式。

16.3.2.1 向上计数

此模式中，计数器使能后从 0 开始计数，直到 CNT=ARR，产生溢出事件，然后重新从 0 开始计数。

如果使能了重复计数功能，则计数器按照 RCR 的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF（Update Interrupt Flag）中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- RCR 影子寄存器被更新为 ATIM_RCR 内容
- ARR 影子寄存器被更新为 ATIM_ARR 内容
- PSC 影子寄存器被更新为 ATIM_PSC 内容

16.3.2.2 向下计数

向下计数模式中，计数器从 ARR 值开始递减，到 0 后产生下溢出事件，并且重新从 ARR 开始计数。

如果使能了重复计数功能，则计数器按照 RCR 的定义重复上述过程若干次（RCR+1），才会产生溢出事件。

软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器自动清零。设置 UG 寄存器是否触发 UIF（Update Interrupt Flag）中断标志置位由 URS 寄存器的设置决定。

通过设置 UDIS 寄存器可以禁止 update event，这样可以避免将 preload 寄存器中的值更新到工作寄存器中。

当 update event 发生时，以下寄存器被更新，并且 UIF 置位：

- RCR 影子寄存器被更新为 ATIM_RCR 内容。
- ARR 影子寄存器被更新为 ATIM_ARR 内容。
- PSC 影子寄存器被更新为 ATIM_PSC 内容。

16.3.2.3 中心对齐计数

在中心对齐模式下，计数器从 0 开始向上计数，到 ARR-1 产生上溢出事件，然后从 ARR 开始向下计数到 1，产生下溢出事件，再从 0 重新开始向上计数。

CMS[1:0]寄存器用于使能中心对齐模式，并选择中心对齐模式下的输出比较工作方式。当 CMS!=00 时为中心对齐计数，当 CMS=01 时，输出比较功能仅在向下计数时有效，当 CMS=10 时，输出比较功能仅在向上计数时有效，当 CMS=11 时，输出比较功能在上下计数时都有效。

中心对齐模式下，DIR 寄存器无法由软件改写，而是随着计数方向变化硬件自动更新，表示当前计数方向。

计数器在 overflow 和 underflow 的事件上都会更新 ARR、PSC 和 RCR 的影子寄存器。

16.3.3 重复计数器

Update event 在计数器 overflow 或 underflow，并且重复计数器为 0 的情况下产生。这意味着 ARR、PSC、CCR（比较/捕捉寄存器，输出比较模式下）的 preload 寄存器会在 N+1 次 overflow 或 underflow 之后，才将数据传输给影子寄存器，其中 N 是 RCR 寄存器值。

重复计数器在以下情况下递减：

- 向上计数模式下发生上溢出
- 向下计数模式下发生下溢出
- 中心计数模式下每次上溢出或者下溢出

注意，当 update event 由软件或 slave mode controller 触发时，更新事件会立即发生，而不管

当前 RCR 是什么值，同时重复计数器也会被立即更新为 RCR 的值。

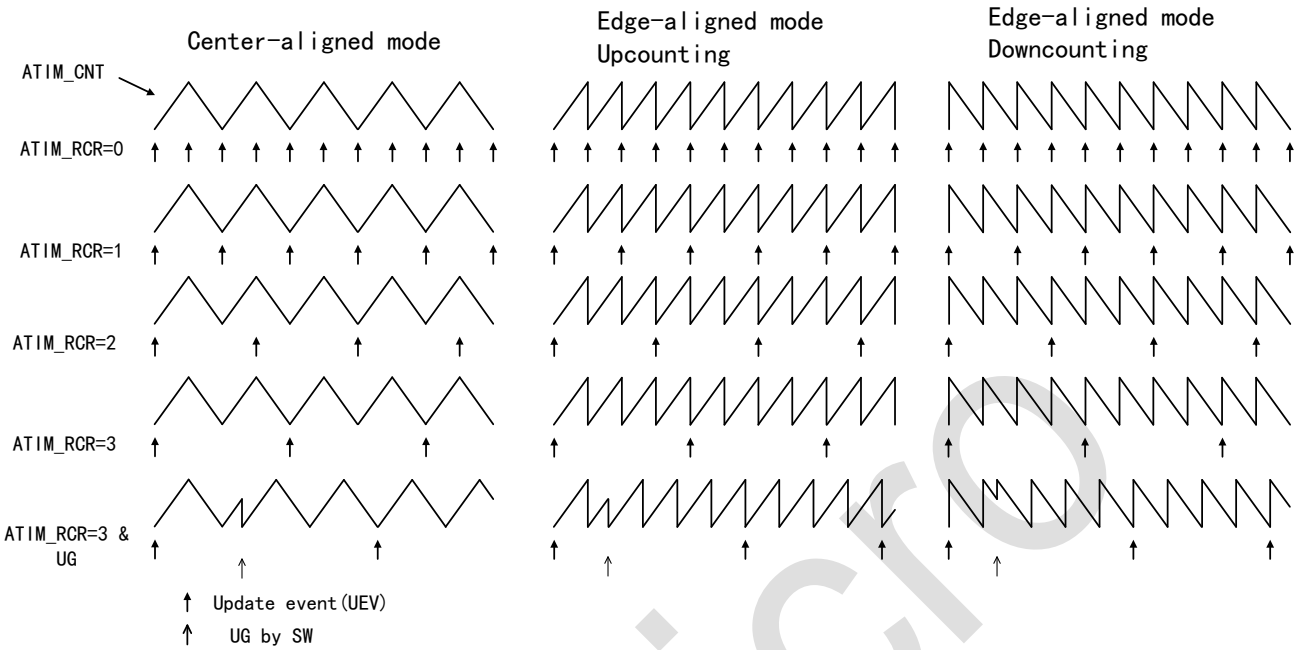


图 16-1：不同模式下更新速率的例子，及 ATIM_RCR 的寄存器设置

16.3.4 Preload 寄存器

- 以下功能寄存器支持 Preload 功能：
  - 自动重载寄存器 ATIM_ARR
  - 预分频寄存器 ATIM_PSC（不可关闭 preload 功能）
  - 通道控制寄存器 ATIM_CCRx
  - CCxE 和 CCxNE 控制寄存器
  - OCxM 控制寄存器

  以上寄存器，除了 ATIM_PSC 之外，都可以由软件选择使能或者禁止 preload 功能。
- 具备 Preload 功能的寄存器，包含两组物理实体：
  - Shadow register（影子寄存器）：实际定时器正在使用的寄存器
  - Preload register（预装载寄存器）：软件可以访问的寄存器
- 当禁止 Preload 时，具备 Preload 功能的寄存器特性如下：
  - Preload 寄存器可以实时由软件访问、改写
  - Shadow 寄存器与 Preload 寄存器同步更新
- 如果使能了 Preload，则：
  - 所有软件操作访问的是 preload 寄存器
  - 当 update event 发生时，所有 Preload 寄存器内容将同步被转移到对应的 shadow 寄存器

16.3.5 计数器工作时钟

计数器可以使用如下时钟工作：

- APBCLK——内部时钟模式
- 外部引脚输入时钟（TIx）——外部时钟模式 1
- 外部引脚触发输入（ETR）——外部时钟模式 2
- 内部触发（ITRx）——使用一个 timer 的触发输出（TRGO）作为计数时钟

16.3.6 内部触发信号(ITRx)

ATIMER 支持 4 个 ITR 输入，可用于计数触发或者内部信号捕捉。当用于内部信号捕捉时，需要将 TS 配置为 000~011 用于选择 ITR0~ITR3，并将 CCxS 配置为 11，即将 TRC 选为捕捉信号。

每个 ITR 输入支持 4 个内部信号扩展，由 ITRxSEL 寄存器配置。输入信号源参考下表：

Slave	ITR0(TS=000)	ITR1(TS=001)	ITR2(TS=010)	ITR3(TS=011)
ATIMER	GTIMER0_TRGO	GTIMER1_TRGO	GTIMER2_TRGO	LPTIM0_OUT1

16.3.7 捕捉/比较通道

ATIMER 包含 4 个捕捉/比较通道，每个通道由一个捕捉比较寄存器（CCR）（包含影子寄存器）、一个捕捉输入级、一个比较输出级组成。

输入级电路会采样 TIx 输入并产生滤波后的信号 TIxF，然后边沿检测和极性选择产生对应的 TIxFPx 信号，此信号可作为计数触发或者待捕捉信号，并且在被捕捉前经过预分频。

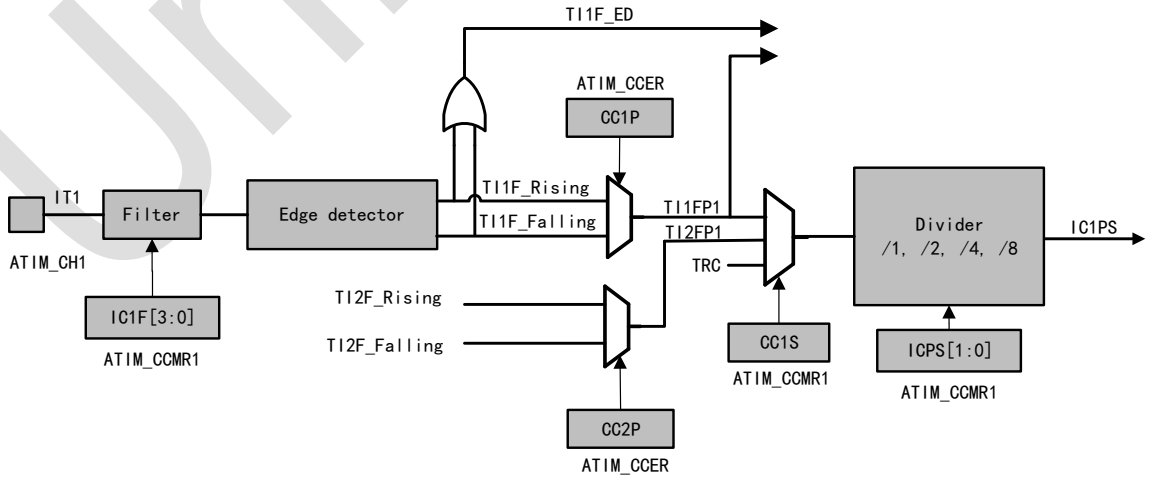


图 16-2：捕获/比较通道(通道 1 输入部分)

输出级电路会产生一个输出基准信号 OCxREF，此信号固定为高电平有效，作为最终输出电路的参考输入。其中通道 1~3 支持互补输出和死区插入，通道 4 则比较简单，不支持互补输出。

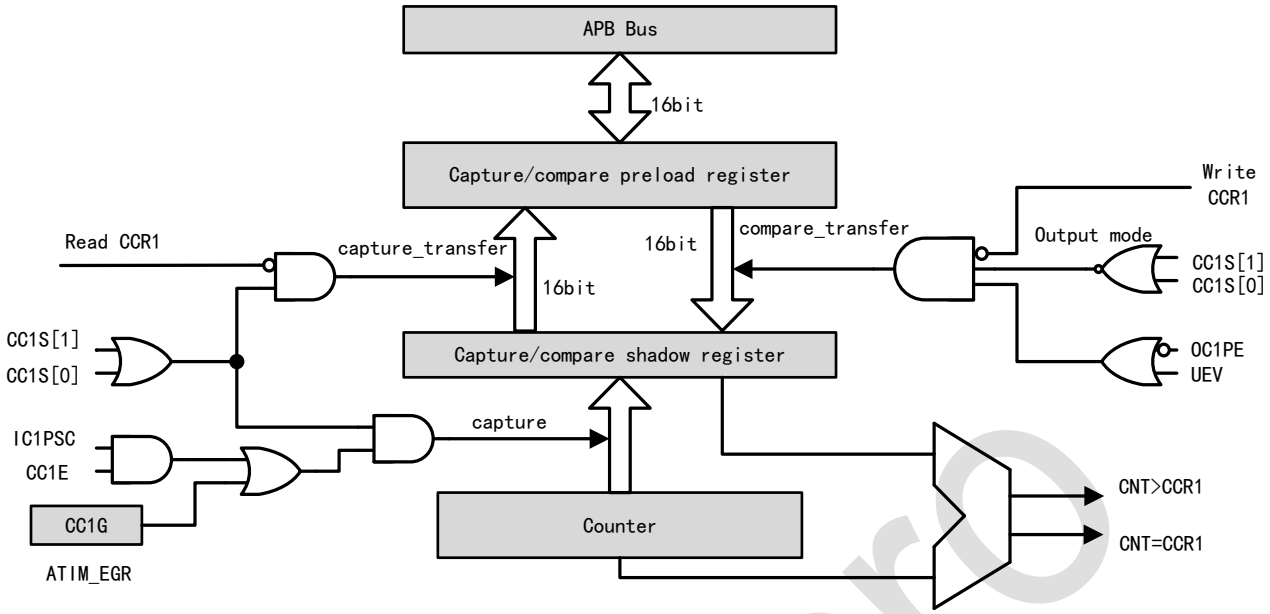


图 16-3：捕获/比较通道 1 的主电路

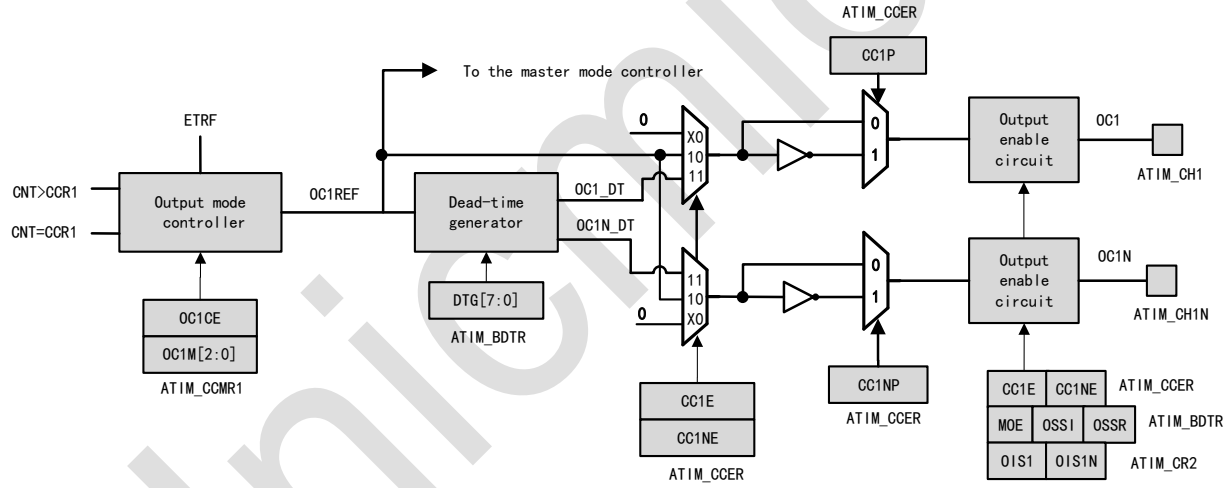


图 16-4：捕获/比较通道的输出部分(通道 1 至 3)

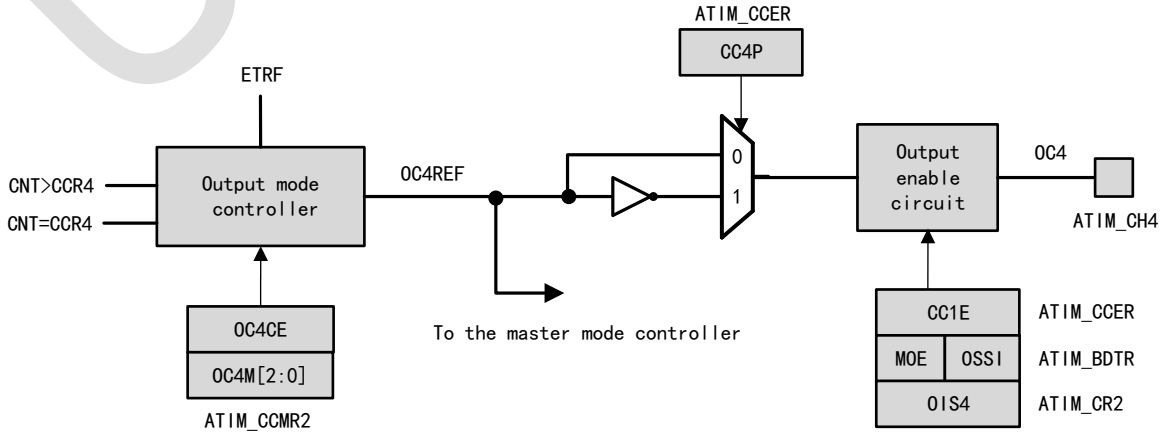


图 16-5：捕获/比较通道的输出部分(通道 4)

捕捉/比较寄存器（CCR）包含 Preload 寄存器和 shadow 寄存器，软件读写总是访问 Preload 寄存器。在捕捉模式下，捕捉值保存在 shadow 寄存器中并复制到 Preload 寄存器。在比较模式下，Preload 寄存器的值被拷贝到 shadow 寄存器用来与计数器比较。

16.3.8 输入捕捉模式

当 ICx 信号上出现预期的电平变换，将触发一次 capture，当前计数器值被锁存进 CCR，与此同时，CCxIF 中断标志置位，并且可以触发对应的中断或者 DMA 请求。如果一个捕捉事件在 CCxIF 为高的情况下出现，则捕捉数据冲突标志（CCxOF, Over-Capture）置位（CCR 中上次捕捉值被覆盖）。CCxIF 可以由软件清零，或者通过读取 CCR 寄存器自动清零。CCxOF 标志通过软件写 1 清零。

通过两个或更多通道配合，可以实现 PWM 信号的输入捕捉。比如要计算一个输入信号的周期和占空比，可以将此信号从 TI1 引脚输入，芯片内部将滤波后的信号取上升沿得到 TI1FP1，将滤波后的信号取下降沿得到 TI1FP2，将 TI1FP1 输入给捕捉通道 1，将 TI1FP2 输入给捕捉通道 2，即可实现通道 1 对输入信号上升沿捕捉，同时通道 2 对输入信号下降沿捕捉；捕捉中断定期发生后，软件通过 CCR1 和 CCR2 寄存器的值，即可计算输入信号的周期和占空比。

16.3.9 软件 Force 输出

在比较输出模式下，软件可以直接将 OCxREF force 成特定电平，而独立于 CCR 和计数器的比较结果。软件通过写 OCxM=101 寄存器，可以直接将 OCxREF 强制为有效（OCxREF 固定为高有效），通过写 OCxM=100 可以直接将 OCxREF 强制为无效（低电平）。但是软件 force 操作不会取消比较过程，CCR 和计数器的比较还会一直进行。

16.3.10 输出比较模式

输出比较模式下，当 CCR 与计数器值相等，OCxREF 可以被置位成有效、无效、或电平翻转。同时，中断标志也会置位，DMA 请求可以发送（改写配置寄存器）。

输出比较也可以被用于输出一个特定宽度的脉冲信号（单次输出）。使用步骤如下：

1. 选择计数时钟（内部、外部、预分频等）。
2. 向 ARR 和 CCR 寄存器写入期望数据。
3. 根据需要设置中断使能和 DMA 使能。
4. 选择输出模式。
5. 使能计数器。

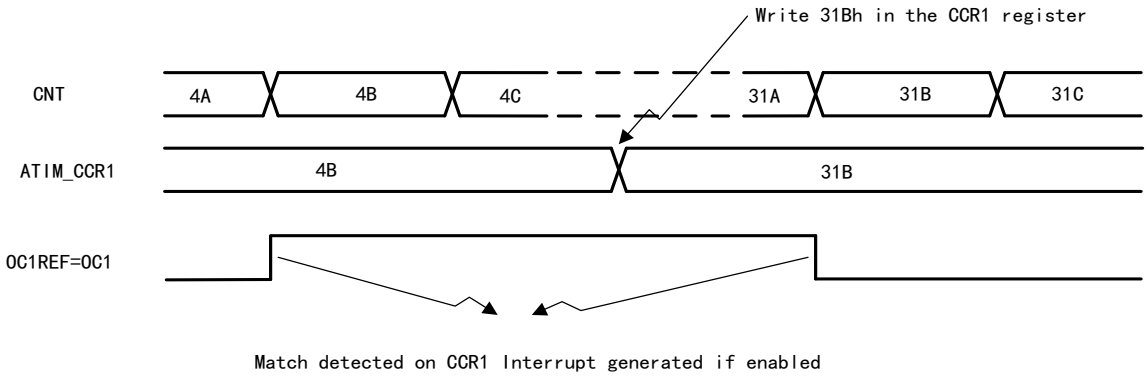


图 16-6：输出比较模式，翻转 OC1

在不使能 Preload 的情况下，软件可以随时改写 CCR 寄存器实现对输出波形的实时控制。如果使能了 Preload，则 CCR shadow 寄存器仅在下一次 update event 发生时更新为 Preload 寄存器的内容。

16.3.11 PWM 输出

PWM 模式可以输出脉宽调制信号，其周期由 ARR 寄存器决定，占空比由 CCR 寄存器决定。输出信号的极性可以由 CCxP 寄存器配置。PWM 模式工作中，CNT 和 CCR 实时比较。由于计数器支持边缘对齐和中央对齐计数模式，PWM 输出也支持边缘对齐和中央对齐模式。

- **PWM 边缘对齐模式**

在向上计数的情况下，配置为 PWM 模式 1 时，OCxREF 信号在 CNT<CCR 时为高电平，否则为低电平。如果 CCR 值大于 ARR 值，则 OCxREF 被固定为 1；如果 CCR 为 0 则 OCxREF 被固定为 0。在向下计数时，OCxREF 电平高低定义与向上计数时相同。

- **PWM 中央对齐模式**

OCxREF 电平定义与边缘对齐模式相同。当启动中央对齐计数时，一开始的计数方向是由 DIR 寄存器决定的；随后在计数过程中，DIR 寄存器的状态由硬件直接控制。安全起见，建议用户程序在启动计数器之前，通过 UG 寄存器做一次 update，并且在计数过程中不要改写计数器。

16.3.12 互补输出和死区插入

ATIMER 的通道 1~3 支持互补输出和死区插入。DTG[7:0]寄存器用于设置死区时间（对所有通道同时有效）。输出信号 OCx 与参考信号 OCxREF 同相，OCxN 与参考信号反相；OCx 的上升沿是 OCxREF 上升沿的 delay，OCxN 的上升沿是 OCxREF 下降沿的 delay。

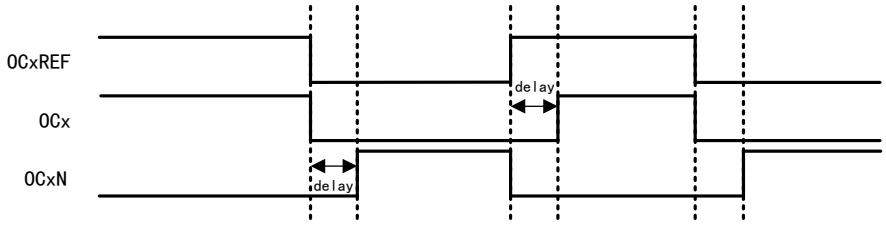


图 16-7：带死区插入的互补输出

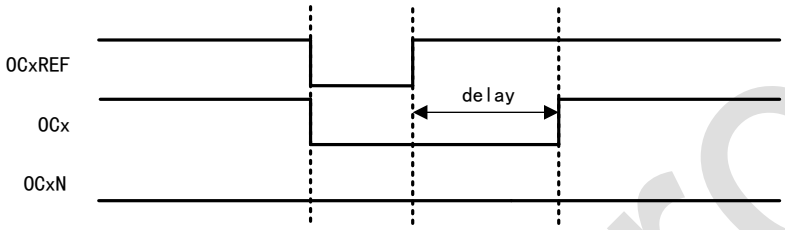


图 16-8：死区波形延迟大于负脉冲

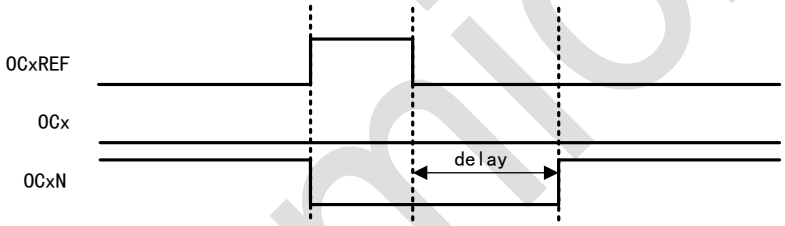


图 16-9：死区波形延迟大于正脉冲

16.3.13 刹车功能

刹车功能可以使用外部 BRK 引脚输入的 2 路刹车信号，或者比较器、LVD；上电复位后刹车电路被禁止，用户通过置位 BKE 寄存器使能刹车功能；2 路刹车输入可以配置为相与或者相或操作。组合后的刹车信号可以配置有效极性，以及数字滤波。

BRKx 复用 GPIO 功能，当 GPIO 设置为数字外设功能时，其输入信号直接连接到 ATIMER 的刹车输入上；当 GPIO 设置为其他功能时，ATIMER 的刹车输入端口被固定成 1。通过 BRKxGATE 寄存器，可以控制门控后的 BRKx 信号的实际电平，软件能够灵活的将不使用的 BRKx 设置为 0 或者 1 电平，以适应后级逻辑电路的需要。

当一个刹车事件发生时：

- 输出使能寄存器被异步清零，可以通过 OSSI 寄存器选择输出被强制为 inactive/idle/reset 状态
- 每个输出通道被驱动为 OISx 寄存器定义的电平
- 当互补输出使能时，输出被异步置位成 inactive 和 reset 状态，死区插入电路开始工作，在死区时间后驱动输出为 OISx 和 OISxN 定义的电平
- 刹车标志寄存器置位，根据配置可以触发中断或 DMA

- 如果使能了自动输出（AOE=1），输出使能位（MOE）将在下一个 update event 发生时被自动置位；否则 MOE 将保持为 0 直到被软件重新置位

注意 BRK 信号是电平有效的，因此在 BRK 保持有效的情况下，无法使能 MOE，同时刹车标志 BIF 也无法清除。

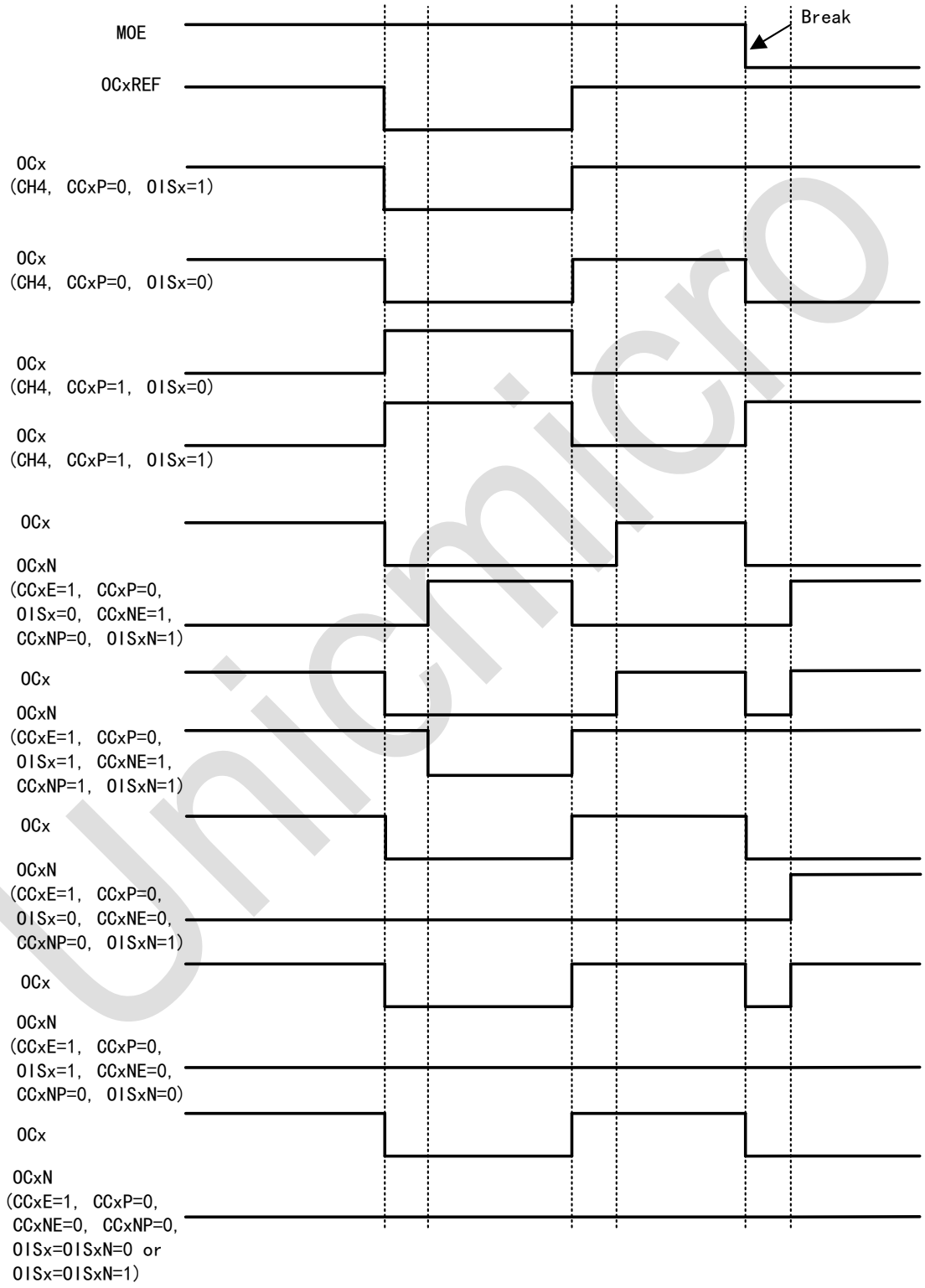


图 16-10：响应刹车的输出

16.3.14 6-step PWM 输出

当某个通道使用互补输出时，OCxM, CCxE, CCxNE 寄存器支持 Preload 功能，Preload 寄存器的值在换相（COM）事件发生时被装载到 shadow 寄存器中。用户因此可以预先设置下一步配置，并在 COM 事件发生时同步更新所有通道。COM 事件可以由软件写 ATIM_EGR 中的 COM 位触发，或者由 TRGI 上升沿硬件触发。

当 COM 事件发生时，换相标志寄存器置位，并且可以产生中断或 DMA 请求。

下图是一个 6 步换相控制的例子，当 COM 事件发生时，三个例子显示不同配置下的输出变化。

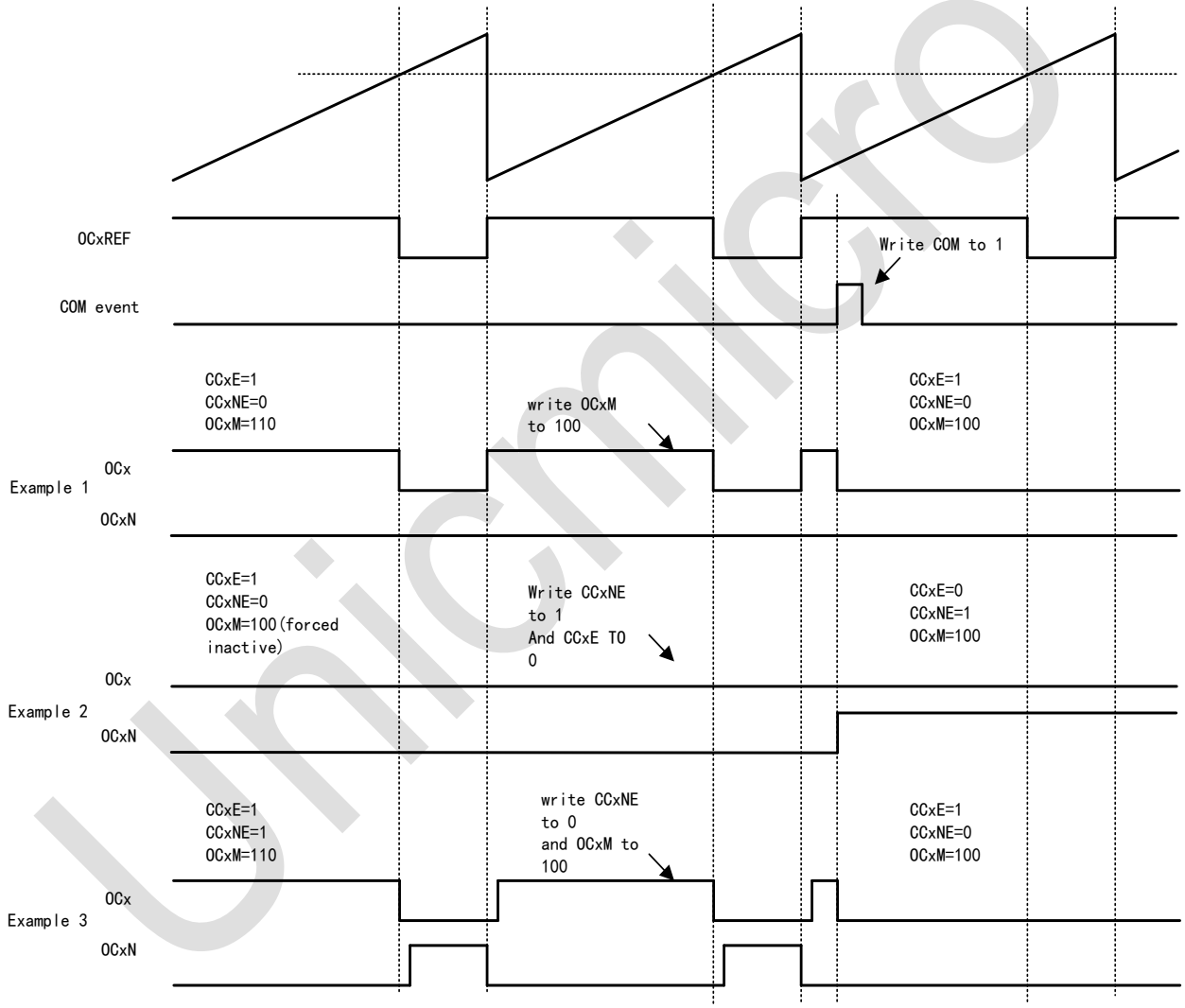


图 16-11：产生六步 PWM，使用 COM 的例子(OSSR=1)

16.3.15 外部事件清除 OCxREF

ETRF 是 ETR 最终生效的信号。OCxREF 的有效状态未高电平，通过对外部 ETR 引脚施加高电平，可以直接拉低 OCxREF，直到下一次 update event。此功能仅在输出比较和 PWM 模式下有效，无法在软件 force 模式下起作用。使能此功能需要将 OCxCE 置 1。

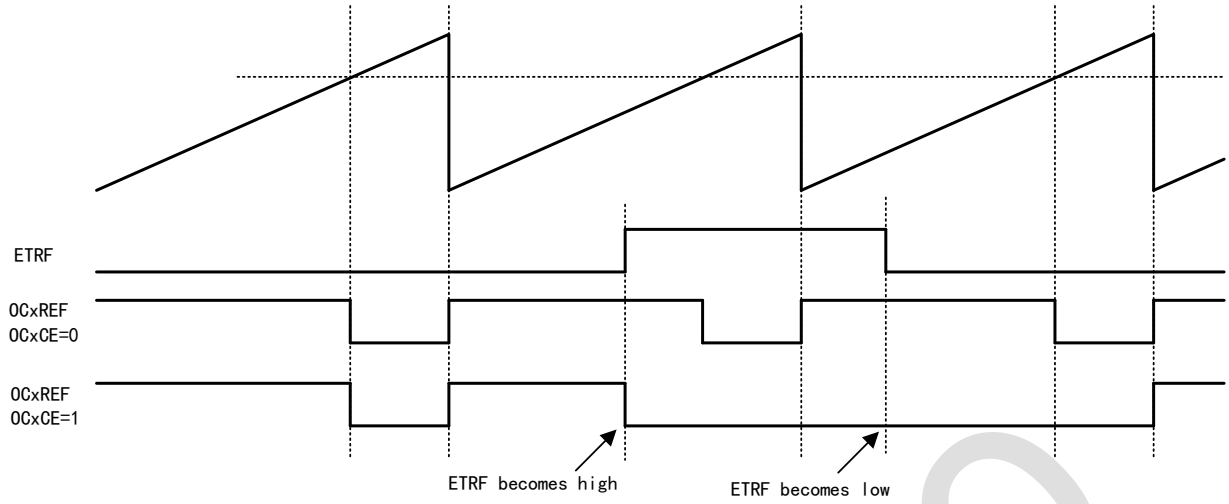


图 16-12：ETR 信号清除 ATIMER 的 OCxREF

16.3.16 编码器接口模式

编码器接口模式涉及到两个外部输入信号，ATIMER 根据其中一个信号的边沿相对于另一个信号的电平来决定递增还是递减计数值。下表是计数方式与两路输入信号之间的关系：

表 16-1：encoder interface 计数方式

有效沿	对应信号的电平（TI1 对应 TI2，TI2 对应 TI1）	TI1 信号		TI2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

比如在计数器以 TI1 信号为时钟计数时，如果 TI1 上升沿采样到 TI2 为高电平，则计数器递减；如果 TI1 下降沿采样到 TI2 为高电平，则计数器递增。

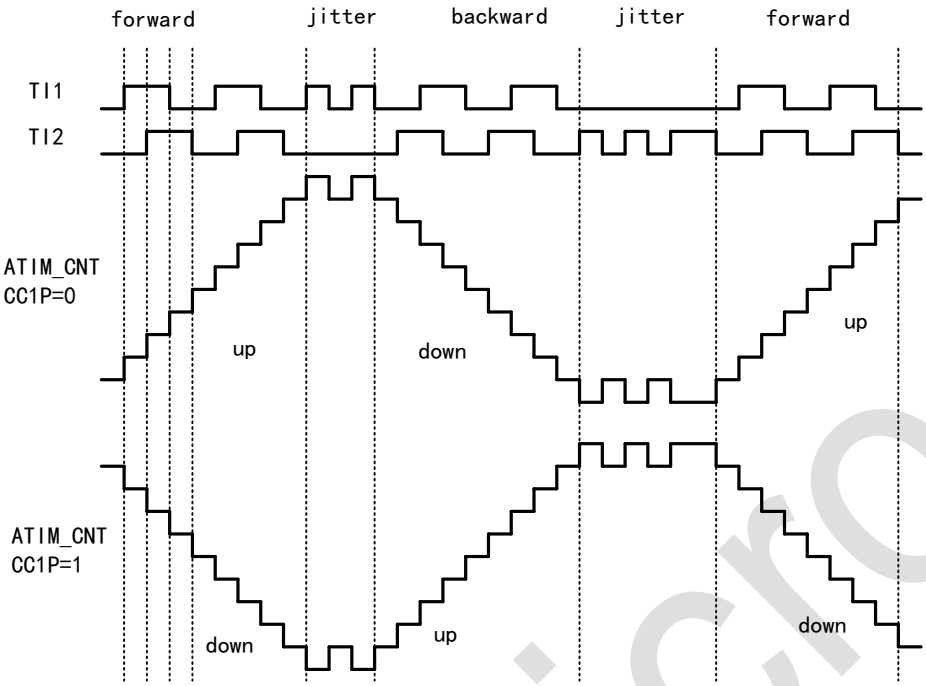


图 16-13：编码器模式下的计数器操作实例

编码模式输入通道需进行如下设置：

1. 在 GPIO 模块中，配置相应管脚为 ATIM_CH1，ATIM_CH2 功能。
2. 关闭通道使能，配置 ATIM_CCER[0]=0，ATIM_CCER[4]=0，确保之后通道配置成功。
3. 选择输入通道，配置 ATIM_CCMR1[1:0]=01，ATIM_CCMR1[9:8]=01。
4. 选择计数有效沿，配置 ATIM_CCER[1]=0，ATIM_CCER[5]=0。
5. 设定从模式控制器为编码模式 3，配置 ATIM_SMCR.SMS[2:0]=011 。
6. 打开通道使能，配置 ATIM_CCER[0]=1，ATIM_CCER[4]=1。

16.3.17 DMA 访问

ATIMER 支持 7 种 DMA 请求，分别为 4 个 CC 通道请求、外部触发请求、用户软件触发请求和 COM 触发请求。

其中每个 CC 通道各自产生一个 DMA 请求，在捕捉模式下用于将 CCRx 中的内容传输给 RAM，在比较模式下则用于将 RAM 中的数据写入 CCRx；CC 通道的 DMA 请求可以配置为单次传输或 Burst 传输（CCxBURSTEN），单次传输仅访问 CCRx 寄存器，Burst 传输则根据 DCR 寄存器配置对特定的一组寄存器进行访问。

此外，外部触发事件、软件触发事件和 COM 事件也可以产生 DMA 请求，当这些请求发生时，会启动 DMA Burst 传输，向 ATIMER 内部 1 个或多个寄存器写入数据，或者从 ATIMER 读取 1 个或多个寄存器值。

表 16-2：DMA 访问计数方式

DMA 请求	CCxBURSTEN	DMA.CHxCTRL.DIR	DMA 访问对象	一次传输长度
ATIM_CH1	0	0	Read CCR1	1
		1	Write CCR1	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH2	0	0	Read CCR2	1
		1	Write CCR2	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH3	0	0	Read CCR3	1
		1	Write CCR3	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_CH4	0	0	Read CCR4	1
		1	Write CCR4	
	1	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_TRIG	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_UEV	N/A	0	Read DMAR	DBL
		1	Write DMAR	
ATIM_COM	N/A	0	Read DMAR	DBL
		1	Write DMAR	

此外，外部触发事件、软件触发事件和 COM 事件也可以产生 DMA 请求，当这些请求发生时，会启动 DMA Burst 传输，向 ATIMER 内部 1 个或多个寄存器写入数据，或者从 ATIMER 读取 1 个或多个寄存器值。

16.3.18 DMA Burst

ATIMER 支持 DMA 和 DMA-Burst 访问，可以配置 ATIMER 在特定事件发生时触发 DMA 请求，可以将 CCR 中的捕捉结果写入 RAM，或者从 RAM 中将一个或多个寄存器内容写入 ATIMER 的 Preload 寄存器中。

DMA-Burst 支持一个事件触发连续多次 DMA 请求，主要作用是在事件发生后连续更新多个寄存器的内容，因此可以实现动态实时调整输出波形等功能。

DMA 控制器需将外设目标地址指向一个虚拟寄存器 ATIM_DMAR。在特定的定时器事件发生时，ATIMER 会连续发射多个 DMA 请求。每个 DMA 对 ATIM_DMAR 的写操作都会被 ATIMER 重新定向到实际的功能寄存器上。

DBL 寄存器用于设置 DMA burst 长度，DBA 寄存器用于设置 DMA 访问 ATIMER 内部的基地址（相对于 ATIM_CR 的 offset）。

DMA-Burst 模式下，DMA 所有访问都要指向 DMAR 虚拟寄存器，由 ATIMER 自动根据访问来

累加内部 offset 地址。DBA 寄存器用于指定 ATIMER 内部首次 DMA 传输的目标地址，而 DBL 用于指定 Burst 长度。

16.3.19 输入异或功能

通道 1~3 的输入信号可以被异或起来之后，接入到通道 1 的滤波和边沿电路输入，用于通道 1 的输入捕捉或者触发。ATIM_CR2 寄存器的 TI1S 位用于选择通道 1 的输入是否来自于三个通道输入的异或。

16.4 寄存器描述

ATIMER 寄存器基地址：0x4000_7400

表 16-3：ATIMER 寄存器列表

偏置	名称	描述
0x00	ATIM_CR1	ATIMER 控制寄存器 1
0x04	ATIM_CR2	ATIMER 控制寄存器 2
0x08	ATIM_SMCR	ATIMER 从机模式控制寄存器
0x0C	ATIM_DIER	ATIMER DMA 和中断使能寄存器
0x10	ATIM_SR	ATIMER 状态寄存器
0x14	ATIM_EGR	ATIMER 事件产生寄存器
0x18	ATIM_CCMR1	ATIMER 捕捉/比较模式寄存器 1
0x1C	ATIM_CCMR2	ATIMER 捕捉/比较模式寄存器 2
0x20	ATIM_CCER	ATIMER 捕捉/比较使能寄存器
0x24	ATIM_CNT	ATIMER 计数器寄存器
0x28	ATIM_PSC	ATIMER 预分频寄存器
0x2C	ATIM_ARR	ATIMER 自动重载寄存器
0x30	ATIM_RCR	ATIMER 重复计数寄存器
0x34	ATIM_CCR1	ATIMER 捕捉/比较寄存器 1
0x38	ATIM_CCR2	ATIMER 捕捉/比较寄存器 2
0x3C	ATIM_CCR3	ATIMER 捕捉/比较寄存器 3
0x40	ATIM_CCR4	ATIMER 捕捉/比较寄存器 4
0x44	ATIM_BDTR	ATIMER 刹车和死区控制寄存器
0x48	ATIM_DCR	ATIMER DMA 控制寄存器
0x4C	ATIM_DMAR	ATIMER DMA 访问寄存器
0x60	ATIM_BKCTL	ATIMER 刹车输入控制寄存器

16.4.1 ATIMER 控制寄存器 1 ATIM_CR1（偏移：00h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留

比特	名称	属性	复位值	描述
15	CEN_ALL	R/W	0	1：同时使能 BTIMER0/1/2/3、GTIMER0/1/2、LPTIMER0/1 和 ATIMER，写此位后，BTIMER0/1/2/3、GTIMER0/1/2、LPTIMER0/1、ATIMER 的 CEN 位同时为 1，开始计数； 0：无操作； 读此位始终为 0；
14	COMP3BKEN	R/W	0	比较器 3 刹车功能使能： 1：使能 0：禁止
13	COMP2BKEN	R/W	0	比较器 2 刹车功能使能： 1：使能 0：禁止
12	COMP1BKEN	R/W	0	比较器 1 刹车功能使能： 1：使能 0：禁止
11	COMP0BKEN	R/W	0	比较器 0 刹车功能使能： 1：使能 0：禁止
10	CAEN	R/W	0	CEN_ALL 控制使能： 1：当前 ATIMER 计数开始受 CEN_ALL 控制 0：当前 ATIMER 计数不受 CEN_ALL 控制
9:8	CKD	R/W	0	Dead time 和数字滤波时钟频率分频寄存器（相对 CK_INT 的分频比）： 00：tDTS=tCK_INT 01：tDTS=2*tCK_INT 10：tDTS=4*tCK_INT 11：保留
7	ARPE	R/W	0	Auto-reload 预装载使能： 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload
6:5	CMS	R/W	0	计数器对齐模式选择： 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位
4	DIR	R/W	0	计数方向寄存器： 1：向下计数 0：向上计数 注意：当定时器配置为中央计数模式或编码器模式时，此寄存器只读

比特	名称	属性	复位值	描述
3	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
2	URS	R/W	0	更新请求选择： 1：仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求 0：以下事件能够产生 update 中断或 DMA 请求：计数器上溢出或下溢出、软件置位 UG 寄存器、从机控制器产生 update
1	UDIS	R/W	0	禁止 update： 1：禁止 update 事件，不更新 shadow 寄存器。当 UG 置位或从机控制器收到硬件 reset 时重新初始化计数器和预分频器。 0：使能 update 事件；以下事件发生时产生 update 事件：计数器上溢出或下溢出、软件置位 UG 寄存器、从机控制器产生 update；
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭 注意：外部触发模式可以自动置位 CEN

16.4.2 ATIMER 控制寄存器 2 ATIM_CR2（偏移：04h）

比特	名称	属性	复位值	描述
31:15	RSV	-	-	保留
14	OIS4	R/W	0	定义 OC4 的输出 IDLE 状态： 1：当 MOE=1 时（如果使能了互补输出，需经过 dead time 后），OC4=1 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC4=0
13	OIS3N	R/W	0	定义 OC3N 的输出 IDLE 状态： 1：当 MOE=1 时，经过 dead time 后，OC3N=1 0：当 MOE=0 时，经过 dead time 后，OC3N=0
12	OIS3	R/W	0	定义 OC3 的输出 IDLE 状态： 1：当 MOE=1 时（如果使能了互补输出，需经过 dead time 后），OC3=1 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC3=0
11	OIS2N	R/W	0	定义 OC2N 的输出 IDLE 状态： 1：当 MOE=1 时，经过 dead time 后，OC2N=1 0：当 MOE=0 时，经过 dead time 后，OC2N=0

比特	名称	属性	复位值	描述
10	OIS2	R/W	0	定义 OC2 的输出 IDLE 状态： 1：当 MOE=1 时（如果使能了互补输出，需经过 dead time 后），OC2=1 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC2=0
9	OIS1N	R/W	0	定义 OC1N 的输出 IDLE 状态： 1：当 MOE=1 时，经过 dead time 后，OC1N=1 0：当 MOE=0 时，经过 dead time 后，OC1N=0
8	OIS1	R/W	0	定义 OC1 的输出 IDLE 状态： 1：当 MOE=1 时（如果使能了互补输出，需经过 dead time 后），OC1=1 0：当 MOE=0 时（如果使能了互补输出，需经过 dead time 后），OC1=0
7	TI1S	R/W	0	ATIMER 输入 TI1 选择： 1：ATIMER_CH1、CH2、CH3 引脚 XOR 后连接到 TI1 输入 0：ATIMER_CH1 引脚连接到 TI1 输入
6:4	MMS	R/W	0	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源 000：ATIMER_EGR 的 UG 寄存器被用作 TRGO： 001：计数器使能信号 CNT_EN 被用作 TRGO，可用于同时启动多个定时器 010：UE（update event）信号被用作 TRGO 011：比较脉冲，如果 CC1IF 标志将要置位，TRGO 输出一个正脉冲 100：OC1REF 用作 TRGO 101：OC2REF 用作 TRGO 110：OC3REF 用作 TRGO 111：OC4REF 用作 TRGO 注意：从机定时器必须事先使能工作时钟，才能接收主机定时器发送的 TRGO
3	CCDS	R/W	0	捕捉/比较 DMA 选择： 1：Update Event 发生时发送 DMA 请求 0：捕捉/比较事件发生时发送 DMA 请求
2	CCUS	R/W	0	捕捉/比较控制寄存器更新选择： 1：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们在置位 COMG 寄存器或者 TRGI 上升沿时更新 0：当捕捉/比较控制寄存器使能了 preload（CCPC=1），他们仅在置位 COMG 寄存器时更新
1	RSV	-	-	保留

比特	名称	属性	复位值	描述
0	CCPC	R/W	0	捕捉/比较预装载控制 1：CCxE, CCxNE, OCxM 寄存器使能 preload 0：CCxE, CCxNE, OCxM 寄存器不使能 preload 注意：此寄存器仅在拥有互补输出功能的通道上有效

16.4.3 ATIMER 从机模式控制寄存器 ATIM_SMCR（偏移：08h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	ETP	R/W	0	外部触发信号极性配置： 1：低电平或下降沿有效 0：高电平或上升沿有效
14	ECE	R/W	0	外部时钟使能： 1：使能外部时钟模式 2，计数器时钟为 ETRF 有效沿 0：关闭外部时钟模式 2
13:12	ETPS	R/W	0	外部触发信号预分频寄存器： 外部触发信号 ETRP 的频率最多只能是 ATIMER 工作时钟的 1/4，当输入信号频率较高时，可以使用预分频。 00：不分频 01：2 分频 10：4 分频 11：8分频
11:8	ETF	R/W	0	外部触发信号滤波时钟和长度选择： 0000：无滤波 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1111：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8

比特	名称	属性	复位值	描述
7	MSM	R/W	0	主机从机模式选择： 1：TRGI 触发的动作被延迟，以便于通过 TRGO 将当前定时器和从机定时器完美同步起来 0：无动作
6:4	TS	R/W	0	触发选择，用于选择同步计数器的触发源： 000：内部触发信号（ITR0） 001：内部触发信号（ITR1） 010：内部触发信号（ITR2） 011：内部触发信号（ITR3） 100：TI1 边沿检测（TI1F_ED） 101：滤波后 TI1（TI1FP1） 110：滤波后 TI2（TI2FP2） 111：外部触发输入（ETRF） 注意：仅当 SMS=000 即禁止从机模式的情况下，可以改写 TS 寄存器
3	RSV	-	-	保留
2:0	SMS	R/W	0	从机模式选择： 000：从机模式禁止；CEN 使能后预分频电路时钟源来自内部时钟 001：Encoder 模式 1；计数器使用 TI2FP1 边沿，根据 TI1FP2 电平高低来计数 010：Encoder 模式 2；计数器使用 TI1FP2 边沿，根据 TI2FP1 电平高低来计数 011：Encoder 模式 3；计数器同时使用 TI1FP1 和 TI2FP2 边沿，根据其他输入信号电平来计数 100：复位模式；TRGI 上升沿初始化计数器，并触发寄存器 update 101：闸门模式；TRGI 为高电平时，计数时钟使能，TRGI 为低电平时，计数时钟停止 110：触发模式；TRGI 上升沿触发计数器开始计数（不会复位计数器） 111：外部时钟模式 1；TRGI 上升沿直接驱动计数器

16.4.4 ATIMER DMA 和中断使能寄存器 ATIM_DIER（偏移：0Ch）

比特	名称	属性	复位值	描述
31:20	RSV	-	-	保留
19	CC4BURSTEN	R/W	0	捕捉比较通道 4 的 DMA 模式配置： 1：Burst 模式，通过 DCR 配置访问的地址和长度 0：Single 模式，仅访问 CCR

比特	名称	属性	复位值	描述
18	CC3BURSTEN	R/W	0	捕捉比较通道 3 的 DMA 模式配置： 1：Burst 模式，通过 DCR 配置访问的地址和长度 0：Single 模式，仅访问 CCR
17	CC2BURSTEN	R/W	0	捕捉比较通道 2 的 DMA 模式配置： 1：Burst 模式，通过 DCR 配置访问的地址和长度 0：Single 模式，仅访问 CCR
16	CC1BURSTEN	R/W	0	捕捉比较通道 1 的 DMA 模式配置： 1：Burst 模式，通过 DCR 配置访问的地址和长度 0：Single 模式，仅访问 CCR
15	RSV	-	-	保留
14	TDE	R/W	0	外部触发 DMA 请求使能： 1：从机模式下，允许外部触发事件产生 DMA 请求（可用于自动更新 preload 寄存器） 0：从机模式下，禁止外部触发事件产生 DMA 请求
13	COMDE	R/W	0	COM 事件 DMA 请求使能： 1：COM 事件发生时，允许产生 DMA 请求 0：COM 事件发生时，禁止产生 DMA 请求
12	CC4DE	R/W	0	捕捉比较通道 4 的 DMA 请求使能： 1：允许 CC4 DMA 请求 0：禁止 CC4 DMA 请求
11	CC3DE	R/W	0	捕捉比较通道 3 的 DMA 请求使能： 1：允许 CC3 DMA 请求 0：禁止 CC3 DMA 请求
10	CC2DE	R/W	0	捕捉比较通道 2 的 DMA 请求使能： 1：允许 CC2 DMA 请求 0：禁止 CC2 DMA 请求
9	CC1DE	R/W	0	捕捉比较通道 1 的 DMA 请求使能： 1：允许 CC1 DMA 请求 0：禁止 CC1 DMA 请求
8	UDE	R/W	0	Update Event DMA 请求使能： 1：Update Event 发生时，允许产生 DMA 请求 0：Update Event 发生时，禁止产生 DMA 请求
7	BIE	R/W	0	刹车事件中断使能： 1：允许刹车事件中断 0：禁止刹车事件中断
6	TIE	R/W	0	触发事件中断使能： 1：允许触发事件中断 0：禁止触发事件中断

比特	名称	属性	复位值	描述
5	COMIE	R/W	0	COM 事件中断使能： 1：允许 COM 事件中断 0：禁止 COM 事件中断
4	CC4IE	R/W	0	捕捉/比较通道 4 中断使能： 1：允许捕捉/比较 4 中断 0：禁止捕捉/比较 4 中断
3	CC3IE	R/W	0	捕捉/比较通道 3 中断使能： 1：允许捕捉/比较 3 中断 0：禁止捕捉/比较 3 中断
2	CC2IE	R/W	0	捕捉/比较通道 2 中断使能： 1：允许捕捉/比较 2 中断 0：禁止捕捉/比较 2 中断
1	CC1IE	R/W	0	捕捉/比较通道 1 中断使能： 1：允许捕捉/比较 1 中断 0：禁止捕捉/比较 1 中断
0	UIE	R/W	0	Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

16.4.5 ATIMER 状态寄存器 ATIM_SR（偏移：10h）

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12	CC4OF	R/W	0	捕捉/比较通道 4 的 Overcapture 中断： 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 1：在 CC4IF 标志为 1 的情况下发生新的捕捉 0：无overcapture事件
11	CC3OF	R/W	0	捕捉/比较通道 3 的 Overcapture 中断： 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 1：在 CC3IF 标志为 1 的情况下发生新的捕捉 0：无overcapture事件
10	CC2OF	R/W	0	捕捉/比较通道 2 的 Overcapture 中断： 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零。 1：在 CC2IF 标志为 1 的情况下发生新的捕捉 0：无overcapture事件

比特	名称	属性	复位值	描述
9	CC1OF	R/W	0	捕捉/比较通道 1 的 Overcapture 中断 此寄存器仅在对应通道设置为输入捕捉模式的情况下有效。硬件置位，软件写 1 清零： 1：在 CC1IF 标志为 1 的情况下发生新的捕捉 0：无 overcapture 事件
8	RSV	-	-	保留
7	BIF	R/W	0	刹车事件中断标志，硬件置位，软件写 1 清零
6	TIF	R/W	0	触发事件中断标志，硬件置位，软件写 1 清零
5	COMIF	R/W	0	COM 事件中断标志，硬件置位，软件写 1 清零
4	CC4IF	R/W	0	捕捉/比较通道 4 中断标志： 如果 CC4 通道配置为输出：CC4IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC4通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读ATIMER_CCR4自动清零。
3	CC3IF	R/W	0	捕捉/比较通道 3 中断标志： 如果 CC3 通道配置为输出：CC3IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC3通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读ATIMER_CCR3自动清零。
2	CC2IF	R/W	0	捕捉/比较通道 2 中断标志： 如果 CC2 通道配置为输出：CC2IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC2通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读ATIMER_CCR2自动清零。
1	CC1IF	R/W	0	捕捉/比较通道 1 中断标志： 如果 CC1 通道配置为输出：CC1IF 在计数值等于比较值时置位，软件写 1 清零。 如果CC1通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读ATIMER_CCR1自动清零。
0	UIF	R/W	0	Update 事件中断标志，硬件置位，软件写 1 清零： 当以下事件发生时，UIF 置位，并更新 shadow 寄存器： 重复计数器=0，并且 UDIS=0 的情况下，计数器发生溢出； URS=0 且 UDIS=0 的情况下，软件置位 UG 寄存器初始化计数器； URS=0且UDIS=0的情况下，触发事件初始化计数器；

16.4.6 ATIMER 事件产生寄存器 ATIM_EGR（偏移：14h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	BG	W	0	软件刹车。软件写 1 产生刹车事件，硬件自动清零
6	TIF	W	0	软件触发。软件写 1 产生触发事件，硬件自动清零
5	COMG	W	0	软件 COM 事件。软件写 1 产生 COM 事件，硬件自动清 0
4	CC4G	W	0	捕捉/比较通道 4 软件触发，软件写 1，硬件自动清 0。此位写 1 后： 如果 CC4 通道配置为输出，CC4IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC4 通道配置为输入，当前计数值被捕捉到 ATIMER_CCR4 寄存器，CC4IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求。
3	CC3G	W	0	捕捉/比较通道 3 软件触发，软件写 1，硬件自动清 0。此位写 1 后： 如果 CC3 通道配置为输出，CC3IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC3 通道配置为输入，当前计数值被捕捉到 ATIMER_CCR3 寄存器，CC3IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求。
2	CC2G	W	0	捕捉/比较通道 2 软件触发，软件写 1，硬件自动清 0。此位写 1 后： 如果 CC2 通道配置为输出，CC2IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果 CC2 通道配置为输入，当前计数值被捕捉到 ATIMER_CCR2 寄存器，CC2IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求。
1	CC1G	W	0	捕捉/比较通道 1 软件触发，软件写 1，硬件自动清 0。此位写 1 后： 如果 CC1 通道配置为输出，CC1IF 置位，在使能的情况下可以产生相应的中断和 DMA 请求 如果CC1通道配置为输入，当前计数值被捕捉到ATIMER_CCR1寄存器，CC1IF置位，在使能的情况下可以产生相应的中断和DMA请求。
0	UG	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零。 软件置位UG时会重新初始化计数器并更新Shadow寄存器，预分频计数器被清零。

16.4.7 ATIMER 捕捉/比较模式寄存器 1 ATIM_CCMR1（偏移：18h）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能。

1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC2CE	R/W	0	输出比较 1 清零使能： 1：检测到 ETRF 高电平时，自动清零 OC2REF 0：OC2REF 不受 ETRF 影响
14:12	OC2M	R/W	0	输出比较 2 模式配置，此寄存器定义 OC2REF 信号的行为： 000：输出比较寄存器 CCR2 和计数器 CNT 的比较结果不会影响输出； 001：CCR2=CNT 时，将 OC2REF 置高； 010：CCR2=CNT 时，将 OC2REF 置低； 011：CCR2=CNT 时，翻转 OC2REF； 100：OC2REF 固定为低（inactive）； 101：OC2REF 固定为高（active）； 110：PWM 模式 1，在向上计数时，OC2REF 在 CNT<CCR2 时置高，否则置低；在向下计数时，OC2REF 在 CNT>CCR2 时置低，否则置高； 111：PWM 模式 2，在向上计数时，OC2REF 在 CNT<CCR2 时置低，否则置高；在向下计数时，OC2REF 在 CNT>CCR2 时置高，否则置低。
11	OC2PE	R/W	0	输出比较 2 预装载使能： 1：CCR2 preload 寄存器有效，针对 CCR2 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 Shadow 寄存器中 0：CCR2 preload 寄存器无效，CCR2 可以直接写入
10	OC2FE	R/W	0	输出比较 2 快速使能： 1：打开快速使能，trigger 输入会立即将 OC2REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效 0：关闭快速使能，trigger 输入不会影响比较输出

比特	名称	属性	复位值	描述
9:8	CC2S	R/W	0	捕捉/比较 2 通道选择： 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S仅在通道关闭时（CC2E=0）可以写
7	OC1CE	R/W	0	输出比较 1 清零使能： 1：检测到 ETRF 高电平时，自动清零 OC1REF 0：OC1REF 不受 ETRF 影响
6:4	OC1M	R/W	0	输出比较 1 模式配置，此寄存器定义 OC1REF 信号的行为： 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR1=CNT 时，将 OC1REF 置高 010：CCR1=CNT 时，将 OC1REF 置低 011：CCR1=CNT 时，翻转 OC1REF 100：OC1REF 固定为低（inactive） 101：OC1REF 固定为高（active） 110：PWM 模式 1 ● 在向上计数时，OC1REF 在 CNT<CCR1 时置高，否则置低； ● 在向下计数时，OC1REF 在 CNT>CCR1 时置低，否则置高 111：PWM模式2 ● 在向上计数时，OC1REF 在 CNT<CCR1时置低，否则置高 ● 在向下计数时，OC1REF 在 CNT>CCR1 时置高，否则置低
3	OC1PE	R/W	0	输出比较 1 预装载使能： 1：CCR1 preload 寄存器有效，针对 CCR1 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中 0：CCR1 preload 寄存器无效，CCR1 可以直接写入
2	OC1FE	R/W	0	输出比较 1 快速使能： 1：打开快速使能，trigger 输入会立即将 OC1REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效 0：关闭快速使能，trigger 输入不会影响比较输出

比特	名称	属性	复位值	描述
1:0	CC1S	R/W	0	捕捉/比较 1 通道选择： 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S仅在通道关闭时（CC1E=0）可以写

2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:12	IC2F	R/W	0	输入捕捉 2 滤波 此寄存器定义 TI2 的采样频率和滤波长度 0000：无滤波，使用 F_{DTS} 采样 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1110：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8
11:10	IC2PSC	R/W	0	输入捕捉 2 预分频： 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC2PSC 寄存器在 CC2E=0 时复位
9:8	CC2S	R/W	0	捕捉/比较 2 通道选择： 00：CC2 通道配置为输出 01：CC2 通道配置为输入，IC2 映射到 TI2 10：CC2 通道配置为输入，IC2 映射到 TI1 11：CC2 通道配置为输入，IC2 映射到 TRC 注意：CC2S仅在通道关闭时（CC2E=0）可以写

比特	名称	属性	复位值	描述
7:4	IC1F	R/W	0	输入捕捉 1 滤波： 此寄存器定义 TI1 的采样频率和滤波长度 0000：无滤波，使用 F_{DTS} 采样 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1110：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8
3:2	IC1PSC	R/W	0	输入捕捉 1 预分频： 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC1PSC寄存器在CC1E=0时复位
1:0	CC1S	R/W	0	捕捉/比较 1 通道选择： 00：CC1 通道配置为输出 01：CC1 通道配置为输入，IC1 映射到 TI1 10：CC1 通道配置为输入，IC1 映射到 TI2 11：CC1 通道配置为输入，IC1 映射到 TRC 注意：CC1S仅在通道关闭时（CC1E=0）可以写

16.4.8 ATIMER 捕捉/比较模式寄存器 2 ATIM_CCMR2（偏移：1Ch）

此寄存器在输出比较和输入捕捉配置下复用为两组不同功能。

1. 输出比较模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	OC4CE	R/W	0	输出比较 4 清零使能： 1：检测到 ETRF 高电平时，自动清零 OC4REF 0：OC4REF 不受 ETRF 影响

比特	名称	属性	复位值	描述
14:12	OC4M	R/W	0	输出比较 4 模式配置，此寄存器定义 OC4REF 信号的行为： 000：输出比较寄存器 CCR4 和计数器 CNT 的比较结果不会影响输出 001：CCR4=CNT 时，将 OC4REF 置高 010：CCR4=CNT 时，将 OC4REF 置低 011：CCR4=CNT 时，翻转 OC4REF 100：OC4REF 固定为低（inactive） 101：OC4REF 固定为高（active） 110：PWM 模式 1 ● 在向上计数时，OC4REF 在 CNT<CCR4 时置高，否则置低 ● 在向下计数时，OC4REF 在 CNT>CCR4 时置低，否则置高 111：PWM 模式 2 ● 在向上计数时，OC4REF 在 CNT<CCR4 时置低，否则置高 ● 在向下计数时，OC4REF 在 CNT>CCR4 时置高，否则置低
11	OC4PE	R/W	0	输出比较 3 预装载使能： 1：CCR4preload 寄存器有效，针对 CCR4 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中 0：CCR4 preload 寄存器无效，CCR4 可以直接写入
10	OC4FE	R/W	0	输出比较 4 快速使能： 1：打开快速使能，trigger 输入会立即将 OC4REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效 0：关闭快速使能，trigger 输入不会影响比较输出
9:8	CC4S	R/W	0	捕捉/比较 4 通道选择： 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S仅在通道关闭时（CC4E=0）可以写
7	OC3CE	R/W	0	输出比较 3 清零使能： 1：检测到 ETRF 高电平时，自动清零 OC3REF 0：OC3REF 不受 ETRF 影响

比特	名称	属性	复位值	描述
6:4	OC3M	R/W	0	输出比较 3 模式配置，此寄存器定义 OC3REF 信号的行为： 000：输出比较寄存器 CCR1 和计数器 CNT 的比较结果不会影响输出 001：CCR3=CNT 时，将 OC3REF 置高 010：CCR3=CNT 时，将 OC3REF 置低 011：CCR3=CNT 时，翻转 OC3REF 100：OC3REF 固定为低（inactive） 101：OC3REF 固定为高（active） 110：PWM 模式 1 ● 在向上计数时，OC3REF 在 CNT<CCR3 时置高，否则置低 ● 在向下计数时，OC3REF 在 CNT>CCR3 时置低，否则置高 111：PWM模式2 ● 在向上计数时，OC3REF在CNT<CCR3时置低，否则置高 ● 在向下计数时，OC3REF在CNT>CCR3 时置高，否则置低
3	OC3PE	R/W	0	输出比较 3 预装载使能： 1：CCR3 preload 寄存器有效，针对 CCR3 的读写操作都是访问 preload 寄存器，当 update event 发生时才将 preload 寄存器的内容转移到 shadow 寄存器中 0：CCR3 preload 寄存器无效，CCR3 可以直接写入
2	OC3FE	R/W	0	输出比较 3 快速使能： 1：打开快速使能，trigger 输入会立即将 OC3REF 改变为比较值匹配时的输出，而不管当前实际比较情况 此功能仅在当前通道配置为 PWM1 或 PWM2 模式时有效 0：关闭快速使能，trigger 输入不会影响比较输出
1:0	CC3S	R/W	0	捕捉/比较 3 通道选择： 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 映射到 TRC 注意：CC3S仅在通道关闭时（CC3E=0）可以写

2. 输入捕捉模式

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留

比特	名称	属性	复位值	描述
15:12	IC4F	R/W	0	输入捕捉 4 滤波： 此寄存器定义 TI4 的采样频率和滤波长度 0000：无滤波，使用 F_{DTS} 采样 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1110：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8
11:10	IC4PSC	R/W	0	输入捕捉 4 预分频： 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC4PSC 寄存器在 CC4E=0 时复位
9:8	CC4S	R/W	0	捕捉/比较 4 通道选择： 00：CC4 通道配置为输出 01：CC4 通道配置为输入，IC4 映射到 TI4 10：CC4 通道配置为输入，IC4 映射到 TI3 11：CC4 通道配置为输入，IC4 映射到 TRC 注意：CC4S仅在通道关闭时（CC4E=0）可以写

比特	名称	属性	复位值	描述
7:4	IC3F	R/W	0	输入捕捉 3 滤波： 此寄存器定义 TI3 的采样频率和滤波长度 0000：无滤波，使用 F_{DTS} 采样 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1110：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8
3:2	IC3PSC	R/W	0	输入捕捉 3 预分频： 00：无分频 01：每 2 个事件输入产生一次捕捉 10：每 4 个事件输入产生一次捕捉 11：每 8 个事件输入产生一次捕捉 IC3PSC寄存器在CC3E=0时复位
1:0	CC3S	R/W	0	捕捉/比较 3 通道选择： 00：CC3 通道配置为输出 01：CC3 通道配置为输入，IC3 映射到 TI3 10：CC3 通道配置为输入，IC3 映射到 TI4 11：CC3 通道配置为输入，IC3 映射到 TRC 注意：CC3S仅在通道关闭时（CC1E=0）可以写

16.4.9 ATIMER 捕捉/比较使能寄存器 ATIM_CCER（偏移：20h）

比特	名称	属性	复位值	描述
31:14	RSV	-	-	保留
13	CC4P	R/W	0	捕捉/比较 4 输出极性： CC4 通道配置为输出时 1：OC4 低电平 active 0：OC4 高电平 active CC4 通道配置为输入时， 1：取反模式：捕捉在 IC4 的下降沿进行 0：非取反模式：捕捉在 IC4 的上升沿进行

比特	名称	属性	复位值	描述
12	CC4E	R/W	0	捕捉/比较 4 输出使能： **CC4 通道配置为输出时** 1：OC4 active 0：OC4 不 active **CC4 通道配置为输入时** 1：使能捕捉功能 0：关闭捕捉功能
11	CC3NP	R/W	0	捕捉/比较 3 互补输出极性： 1：OC3N 低电平为 active 0：OC3N 高电平为 active
10	CC3NE	R/W	0	捕捉/比较 3 互补输出使能： 1：互补输出使能； 0：OC3N 无效，OC3N 电平由 MOE, OSSI, OSSR, OIS3, OIS3N, CC3E 寄存器决定
9	CC3P	R/W	0	捕捉/比较 3 输出极性： **CC3 通道配置为输出时** 1：OC3 低电平 active 0：OC3 高电平 active **CC3 通道配置为输入时** 1：取反模式：捕捉在 IC3 的下降沿进行 0：非取反模式：捕捉在 IC3 的上升沿进行
8	CC3E	R/W	0	捕捉/比较 3 输出使能： **CC3 通道配置为输出时** 1：OC3 active 0：OC3 不 active **CC3 通道配置为输入时** 1：使能捕捉功能 0：关闭捕捉功能
7	CC2NP	R/W	0	捕捉/比较 2 互补输出极性 1：OC2N 低电平为 active 0：OC2N 高电平为 active
6	CC2NE	R/W	0	捕捉/比较 2 互补输出使能； 1：互补输出使能 0：OC2N 无效，OC2N 电平由 MOE, OSSI, OSSR, OIS2, OIS2N, CC2E 寄存器决定
5	CC2P	R/W	0	捕捉/比较 2 输出极性： **CC2 通道配置为输出时** 1：OC2 低电平 active 0：OC2 高电平 active **CC2 通道配置为输入时** 1：取反模式：捕捉在 IC2 的下降沿进行 0：非取反模式：捕捉在 IC2 的上升沿进行

比特	名称	属性	复位值	描述
4	CC2E	R/W	0	捕捉/比较 2 输出使能： **CC2 通道配置为输出时** 1：OC2 active 0：OC2 不 active **CC2 通道配置为输入时** 1：使能捕捉功能 0：关闭捕捉功能
3	CC1NP	R/W	0	捕捉/比较 1 互补输出极性： 1：OC1N 低电平为 active 0：OC1N 高电平为 active
2	CC1NE	R/W	0	捕捉/比较 1 互补输出使能： 1：互补输出使能 0：OC1N 无效，OC1N 电平由 MOE, OSSI, OSSR, OIS1, OIS1N, CC1E 寄存器决定
1	CC1P	R/W	0	捕捉/比较 1 输出极性： **CC1 通道配置为输出时** 1：OC1 低电平 active 0：OC1 高电平 active **CC1 通道配置为输入时** 1：取反模式–捕捉在 IC1 的下降沿进行 0：非取反模式–捕捉在 IC1 的上升沿进行
0	CC1E	R/W	0	捕捉/比较 1 输出使能： **CC1 通道配置为输出时** 1：OC1 active 0：OC1 不 active **CC1 通道配置为输入时** 1：使能捕捉功能 0：关闭捕捉功能

以下控制寄存器和互补输出通道的状态对应表，其中 MOE 为定时器总输出使能位，OSSI 为 IDLE 状态（MOE=0）下的 off_state 选择位，OSSR 为 RUN 状态（MOE=1）下的 off_state 选择位。

Off-state：

控制寄存器					输出状态	
MOE	**OSSI**	**OSSR**	**CCxE**	**CCxNE**	**OCx输出状态**	**OCxN输出状态**
1	X	0	0	0	输出关闭（不由ATIMER驱动）,OCx=0,OCx_EN=0	输出关闭（不由ATIMER驱动）,OCxN=0,OCxN_EN=0
		0	0	1	输出关闭（不由ATIMER驱动）,OCx=0,OCx_EN=0	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		0	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Output Disabled (not driven by the timer) OCxN=0, OCxN_EN=0

控制寄存器					输出状态	
MOE	**OSSI**	**OSSR**	**CCxE**	**CCxNE**	**OCx输出状态**	**OCxN输出状态**
		0	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
		1	0	0	Output Disabled (not driven by the timer) OCx=CCxP, OCx_EN=0	Output Disabled (not driven by the timer) OCxN=CCxNP, OCxN_EN=0
		1	0	1	Off-State (output enabled with inactive state) OCx=CCxP, OCx_EN=1	OCxREF + Polarity OCxN=OCxREF xor CCxNP, OCxN_EN=1
		1	1	0	OCxREF + Polarity OCx=OCxREF xor CCxP, OCx_EN=1	Off-State (output enabled with inactive state) OCxN=CCxNP, OCxN_EN=1
		1	1	1	OCREF + Polarity + dead-time OCx_EN=1	Complementary to OCREF (not OCREF) + Polarity + dead-time OCxN_EN=1
0	0	X	0	0	输出关闭（不由ATIMER驱动） OCx=CCxP, OCx_EN=0	输出关闭（不由ATIMER驱动） OCxN=CCxNP, OCxN_EN=0
	0		0	1	输出关闭（不由ATIMER驱动） 如果无时钟：OCx=CCxP, OCx_EN=0, OCxN=CCxNP, OCxN_EN=0 如果有时钟：经过死区时间后OCx=OISx, OCxN=OISxN	
	0		1	0		
	0		1	1		
	1		0	0	输出关闭（不由ATIMER驱动） OCx=CCxP, OCx_EN=0	输出关闭（不由ATIMER驱动） OCxN=CCxNP, OCxN_EN=0
	1		0	1	Off-state（输出使能，inactive输出） 如果无时钟：OCx=CCxP, OCx_EN=1, OCxN=CCxNP, OCxN_EN=1 如果有时钟：经过死区时间后OCx=OISx, OCxN=OISxN	
	1		1	0		
	1		1	1		

16.4.10 ATIMER 计数器寄存器 ATIM_CNT（偏移：24h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT	R/W	0	计数值

16.4.11 ATIMER 预分频寄存器 ATIM_PSC（偏移：28h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC	R/W	0	计数器时钟（CK_CNT）预分频值 $F_{CK\_CNT}=F_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器； 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

16.4.12 ATIMER 自动重载寄存器 ATIM_ARR（偏移：2Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR	R/W	0	计数溢出时的自动重载值； 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器； 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

16.4.13 ATIMER 重复计数寄存器 ATIM_RCR（偏移：30h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	REP	R/W	0	重复计数值。REP不为0时，每次update条件发生时REP递减，当REP=0时触发update事件

16.4.14 ATIMER 捕捉/比较寄存器 1 ATIM_CCR1（偏移：34h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR1	R/W	0	捕捉/比较通道1寄存器 **如果通道1配置为输出：** 这是一个preload寄存器，其内容被载入shadow寄存器后用于与计数器比较产生OC1输出 **如果通道1配置为输入：** CCR1保存最近一次输入捕捉事件发生时的计数器值，此时CCR1为只读

16.4.15 ATIMER 捕捉/比较寄存器 2 ATIM_CCR2（偏移：38h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR2	R/W	0	捕捉/比较通道2寄存器 **如果通道2配置为输出：** 这是一个preload寄存器，其内容被载入shadow寄存器后用于与计数器比较产生OC2输出 **如果通道2配置为输入：** CCR2保存最近一次输入捕捉事件发生时的计数器值，此时CCR2为只读

16.4.16 ATIMER 捕捉/比较寄存器 3 ATIM_CCR3（偏移：3Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR3	R/W	0	捕捉/比较通道3寄存器 **如果通道3配置为输出：** 这是一个preload寄存器，其内容被载入shadow寄存器后用于与计数器比较产生OC3输出 **如果通道3配置为输入：** CCR3保存最近一次输入捕捉事件发生时的计数器值，此时CCR3为只读

16.4.17 ATIMER 捕捉/比较寄存器 4 ATIM_CCR4（偏移：40h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR4	R/W	0	捕捉/比较通道4寄存器 **如果通道4配置为输出：** 这是一个preload寄存器，其内容被载入shadow寄存器后用于与计数器比较产生OC4输出 **如果通道4配置为输入：** CCR4保存最近一次输入捕捉事件发生时的计数器值，此时CCR4为只读

16.4.18 ATIMER 刹车和死区控制寄存器 ATIM_BDTR（偏移：44h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	MOE	R/W	0	输出使能主控： 此寄存器控制所有通道的输出使能，每个通道独立的输出使能还需要CCxE和CCxNE来控制。MOE由软件置位，或者在AOE=1的情况下硬件触发自动置位。当刹车输入有效时，MOE被硬件异步清零。 1：使能OC和OCN输出（仍需各个通道的CCxE和CCxNE状态来决定是否输出） 0：关闭OC和OCN输出，具体IO输出状态由OSSI决定
14	AOE	R/W	0	自动输出使能： 1：MOE可以软件置位，或者由update事件自动置位 0：MOE仅能由软件置位
13	BKP	R/W	0	刹车极性： 1：刹车输入为高电平有效 0：刹车输入为低电平有效
12	BKE	R/W	0	刹车使能： 1：允许刹车输入 0：禁止刹车输入
11	OSSR	R/W	0	运行状态下的输出关闭状态选择： 仅在MOE=1的情况下，针对使能了互补输出的通道有效。 1：输出通道不使能时，OC和OCN驱动GPIO为无效状态 0：输出通道不使能时，OC和OCN不驱动GPIO
10	OSSI	R/W	0	IDLE状态下的输出关闭状态选择： 仅在MOE=0的情况下，针对输出通道有效。 1：输出通道不使能时，OC和OCN先驱动空闲状态，待死区时间结束后，启动无效状态 0：输出通道不使能时，OC和OCN不驱动GPIO

比特	名称	属性	复位值	描述
9:8	LOCK	R/W	0	寄存器写保护配置： 00：无写保护 01：保护等级1 – DTG, OISx, OISxN, BKE, BKP, AOE不能改写 10：保护等级2 –在等级1基础上，CCxP, CCxNP, OSSR, OSSI不能改写 11：保护等级3 –在等级2基础上，OCxM, OcxPE在相应通道配置为输出时不能改写 注意：LOCK寄存器在被写入非00值之后无法再改写，写保护后的寄存器只有在ATIMER模块被复位后才能重新写入。
7:0	DTG	R/W	0	死区时间插入，用于配置互补输出插入的死区时间长度： DTG[7:5]=0xx：DT=DTG[7:0] * T_{DTS} DTG[7:5]=10x：DT=(64+DTG[5:0]) * 2 * T_{DTS} DTG[7:5]=110：DT=(32+DTG[4:0]) * 8 * T_{DTS} DTG[7:5]=111：DT=(32+DTG[4:0]) * 16 * T_{DTS}

16.4.19 ATIMER DMA 控制寄存器 ATIM_DCR（偏移：48h）

比特	名称	属性	复位值	描述
31:13	RSV	-	-	保留
12:8	DBL	R/W	0	DMA Burst长度： 对ATIM_DMAR寄存器的读写将触发burst DMA操作，burst长度为1~18 00000：长度=1 00001：长度=2 …… 10001：长度=18 其他：无效值，禁止写入
7:5	RSV	-	-	保留
4:0	DBA	R/W	0	DMA基地址，定义指向寄存器的偏移地址： 00000：ATIM_CR1 00001：ATIM_CR2 00010：ATIM_SMCR …… 注意：当DBA+DBL超出了ATIMER寄存器地址范围，则实际burst传输到ATIMER最高寄存器地址后自动停止，即burst长度会缩短。

16.4.20 ATIMER DMA 访问寄存器 ATIM_DMAR（偏移：4Ch）

比特	名称	属性	复位值	描述
31:0	DMAR	R/W	0	DMA burst访问寄存器 在使用DMA burst传输时，将DMA通道外设地址设置为ATIM_DMAR，ATIMER会根据DBL的值产生多次DMA请求

16.4.21 ATIMER 刹车输入控制寄存器 ATIM_BKCTL（偏移：60h）

比特	名称	属性	复位值	描述
31:10	RSV	-	-	保留
9	BRK2GATE	R/W	1	ATIMER BRK2引脚输入门控信号 1：不门控 0：将ATIMER BRK2的输入门控成0
8	BRK1GATE	R/W	1	ATIMER BRK1引脚输入门控信号 1：不门控 0：将ATIMER BRK1的输入门控成0
7:4	BRKF	R/W	0	刹车信号的滤波时钟和长度选择 0000：无滤波 0001：$F_{SAMPLING}=F_{CK\_INT}$, N=2 0010：$F_{SAMPLING}=F_{CK\_INT}$, N=4 0011：$F_{SAMPLING}=F_{CK\_INT}$, N=8 0100：$F_{SAMPLING}=F_{DTS/2}$, N=6 0101：$F_{SAMPLING}=F_{DTS/2}$, N=8 0110：$F_{SAMPLING}=F_{DTS/4}$, N=6 0111：$F_{SAMPLING}=F_{DTS/4}$, N=8 1000：$F_{SAMPLING}=F_{DTS/8}$, N=6 1001：$F_{SAMPLING}=F_{DTS/8}$, N=8 1010：$F_{SAMPLING}=F_{DTS/16}$, N=5 1011：$F_{SAMPLING}=F_{DTS/16}$, N=6 1100：$F_{SAMPLING}=F_{DTS/16}$, N=8 1101：$F_{SAMPLING}=F_{DTS/32}$, N=5 1110：$F_{SAMPLING}=F_{DTS/32}$, N=6 1111：$F_{SAMPLING}=F_{DTS/32}$, N=8
3	BRKCOMB	R/W	0	刹车组合控制： 1：两路刹车信号相与 0：两路刹车信号相或
2	COMP_BRKEN	R/W	0	比较器输出刹车信号使能： 1：使能比较器刹车信号 0：禁止比较器刹车信号
1	LVD_BRKEN	R/W	0	LVD刹车信号使能： 1：使能LVD刹车信号 0：禁止LVD刹车信号
0	GTIMER2_TRGO_BRKEN	R/W	0	GTIMER2 TRGO输出刹车信号使能： 1：使能比较器刹车信号 0：禁止比较器刹车信号

16.5 使用流程

16.5.1 定时计数模式

1. 配置 ATIM_CR1[4]，设置计数方向。
2. 配置 ATIM_CR1[7]为 1，使能 Auto-reload 预装载。
3. 配置 ATIM_PSC，设置预分频值。
4. 配置 ATIM_ARR，设置自动重载值。
5. 配置 ATIM_RCR 为 0，不进行重复计数。
6. 配置 ATIM_CR1[1]为 0，使能 update 事件。
7. 配置 ATIM_EGR[0]为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
8. 配置 ATIM_CR1[2]为 1，设置仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求。
9. 配置 ATIM_CR1[0]为 1，使能计数器。
10. 配置 ATIM_DIER[0]为 1，允许 Update 事件中断。

16.5.2 PWM 模式

1. 配置 ATIM_CR1[4]，设置计数方向。
2. 配置 ATIM_CR1[7]为 1，使能 Auto-reload 预装载。
3. 配置 ATIM_PSC，设置预分频值。
4. 配置 ATIM_ARR，设置自动重载值。
5. 配置 ATIM_RCR 为 0，不进行重复计数。
6. 根据输出通道配置 ATIM_CCMRx[1:0]和 ATIM_CCMRx[9:8]为 0，设置通道 x 为输出。
7. 配置 ATIM_CCMRx 的 OCxM，设置为 PWM 模式 1//2。
8. 配置 ATIM_CCER 的 CCxP，设置输出极性。
9. 配置 ATIM_CCER 的 CCxE 为 1，通道 x 输出使能。
10. 配置 ATIM_BDTR[15]为 1，该位为输出使能主控，使能 OC 和 OCN 输出。
11. 配置 ATIM_CR1[2]为 1，设置仅计数器上溢出或下溢出会产生 update 中断或 DMA 请求。
12. 配置 ATIM_CR1[1]为 0，使能 update 事件。
13. 配置 ATIM_EGR[0]为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
14. 配置 ATIM_CR1[0]为 1，使能计数器。
15. 配置 ATIM_DIER[0]为 1，允许 Update 事件中断。
16. 配置 ATIM_CCRx，设置通道 x 的比较值。

16.5.3 输入捕捉模式

1. 配置 ATIM_CR1[4]，设置计数方向。
2. 配置 ATIM_CR1[7]为 1，使能 Auto-reload 预装载。
3. 配置 ATIM_PSC，设置预分频值。
4. 配置 ATIM_ARR，设置自动重载值。
5. 配置 ATIM_RCR 为 0，不进行重复计数。
6. 配置 ATIM_CCMRx[1:0]和 ATIMER_CCMRx[9:8]，设置 CCx 通道为输入，并更根据需求映射。
7. 配置 ATIM_CCER 的 CCxP 和 CCxNP，设置捕捉极性。
8. 配置 ATIM_CCMRx 的 ICxF，设置采样频率和滤波长度，一般设置为 0 即可。
9. 配置 ATIM_CCMRx 的 ICxPSC，设置输入捕捉预分频。
10. 配置 ATIM_CCER 的 CCxE 为 1，使能捕捉功能。
11. 配置 ATIM_EGR[0]为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
12. 配置 ATIM_CR1[0]为 1，使能计数器。
13. 配置 ATIM_DIER 的 CCxIE 为 1，允许通道 x 捕捉中断。

16.5.4 互补输出和死区插入

96MHz 的周期时间 T_{DTS} = 10.42 ns

1. DT = (0~127) * 10.42 = 0~1323.34 ns，DTG[7:5]=0xx：DT=DTG[7:0] * T_{DTS}
2. DT = (64 + (0~63)) * 2 * 10.42 = 1333.76~2646.68 ns，DTG[7:5]=10x：DT=(64+ DTG[5:0]) * 2 * T_{DTS}
3. DT = (32 + (0~31)) * 8 * 10.42 = 2667.52~5251.68 ns，DTG[7:5]=110：DT=(32+ DTG[4:0]) * 8 * T_{DTS}
4. DT = (32 + (0~31)) * 16 * 10.42 = 5335.04~10503.36 ns，
   DTG[7:5]=111：DT= (32 + DTG[4:0]) * 16 * T_{DTS}

在初始化 PWM 模式前，补充以下配置：

- 配置 ATIM_BDTR[7:0]，设置互补输出的死区时间长度。
- 配置 ATIM_CCER 的 CCxNE 为 1，设置预分频值。

16.5.5 刹车功能

1. 在初始化 PWM 模式前，补充以下配置。
2. 配置 ATIM_BDTR[11]，设置运行状态下的输出关闭状态。

3. 配置 ATIM_BDTR[10]，设置空闲状态下的输出关闭状态。
4. 配置 ATIM_BDTR[13]，设置刹车极性。
5. 配置 ATIM_CCER 的 CCxP，设置 OCx 的输出极性。
6. 配置 ATIM_CCER 的 CCxNP，设置 OCxN 的输出极性。
7. 配置 ATIM_CR2 的 OISx，设置 OCx 的空闲输出状态。
8. 配置 ATIM_CR2 的 OISxN，设置 OCxN 的空闲输出状态。
9. 配置 ATIM_BDTR[14]，设置 ATIMER_BDTR 的 MOE 置位方式。
10. 配置 ATIM_BDTR[12]为 1，允许刹车输入。

16.5.6 编码器接口模式

1. 配置 ATIM_CR1[4]，设置计数方向。
2. 配置 ATIM_CR1[7]为 1，使能 Auto-reload 预装载。
3. 配置 ATIM_PSC，设置预分频值。
4. 配置 ATIM_ARR，设置自动重载值。
5. 配置 ATIM_RCR 为 0，不进行重复计数。
6. 配置 ATIM_CCMR1[1:0]为 1，设置 CC1 通道为输入，IC1 映射到 TI1。
7. 配置 ATIM_CCMR1[9:8]为 1，设置 CC2 通道为输入，IC2 映射到 TI2。
8. 配置 ATIM_CCER[1]和 ATIM_CCER[3]，设置捕捉极性。
9. 配置 ATIMER_CCER[5]和 ATIM_CCER[7]，设置捕捉极性。
10. 配置 ATIM_CCMR1 的 IC1F，设置采样频率和滤波长度，一般设置为 0 即可。
11. 配置 ATIM_CCMR1 的 IC2F，设置采样频率和滤波长度，一般设置为 0 即可。
12. 配置 ATIM_SMCR[2:0]，设置 Encoder 模式 1/2/3。
13. 配置 ATIM_CCER[0]为 1，使能通道 1 捕捉功能。
14. 配置 ATIM_CCER[4]为 1，使能通道 2 捕捉功能。
15. 配置 ATIM_EGR[0]为 1，软件置位 UG 时会重新初始化计数器并更新 shadow 寄存器，预分频计数器被清零。
16. 配置 ATIM_CR1[0]为 1，使能计数器。
17. 配置 ATIM_DIER[1]为 1，允许通道 1 捕捉中断。

16.5.7 DMA 模式

输出比较模式下，SRAM 中的值通过 DMA 传输到 ATIMER 的比较寄存器：

1. 在 PWM 模式中软件置位 UG 和使能计数器前，补充以下配置。
2. 配置 ATIM_DCR[12:8]，设置 DMA Burst 长度。

3. 配置 ATIM_DCR[4:0]，设置 DMA 基地址，一般该处的基地址选择相应比较通道对应的捕捉/比较寄存器。
4. 配置 ATIM_DIER 的 CCxDE 为 1，允许 CCx DMA 请求。
5. 配置 ATIM_CR2[3]为 0，发生 CCx 事件时产生 CCxDMA 请求。
6. DMA 控制器配置详细请看 22 DMA 章节。
7. 开启 DMA 传输后，当计数器计数值等于比较值时，DMA 将 SRAM 中的值传到基地址。

17 GTIMER0/1/2

17.1 概述

有 3 个 16 位的通用定时/计数器 GTIMER0/1/2，每个定时器都有自己独立的中断。这些 Timer 可以有多种用途，包括测量输入信号的脉冲宽度（输入捕获），产生输出波形（PWM、带死区时间的互补 PWM），计数器可以向上，向下，向上/下三种计数方向，且计数值可以随时由软件读取。每个 Timer 有 2 路 PWM 输出(可选是否互补)，有 1 路输入捕获。

17.2 主要特性

- 16 位向上、向下、向上/下计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 灵活的计数时钟源选择
- 通道可用于输入捕获、输出比较、PWM(边沿或中央对齐模式)、单脉冲输出
- 支持定时器间的级联
- 可编程的带死区时间互补输出
- 带刹车输入信号控制功能，可使 PWM 输出置于一个可设置的状态
- 中断在以下几种情况产生：
  - Update 中断：计数器向上/向下溢出
  - 输入捕获
  - 输出比较
  - 刹车信号输入
- 支持定时器同步使能

17.3 寄存器描述

GTIMER0 寄存器基地址：0x4000_0C00

GTIMER1 寄存器基地址：0x4000_3400

GTIMER2 寄存器基地址：0x4000_3800

表 17-1：GTIMER 寄存器列表

偏置	名称	描述
0x00	GTIM_CR	GTIMER 控制寄存器
0x04	GTIM_IER	GTIMER 中断使能寄存器
0x08	GTIM_SR	GTIMER 状态寄存器
0x0C	GTIM_EGR	GTIMER 事件产生寄存器
0x10	GTIM_CCMR	GTIMER 捕捉/比较模式寄存器
0x14	GTIM_CCER	GTIMER 捕获/比较使能寄存器
0x18	GTIM_CNT	GTIMER 计数器寄存器
0x1C	GTIM_PSC	GTIMER 预分频寄存器
0x20	GTIM_ARR	GTIMER 自动重载寄存器
0x24	GTIM_CCR	GTIMER 捕捉/比较寄存器
0x28	GTIM_CARS1	GTIMER 硬件触发寄存器

17.3.1 GTIMER 控制寄存器 GTIM_CR (偏移：00h)

比特	名称	属性	复位值	描述
31	CRAR_STRE	R/W	0	CCR和ARR影子寄存器停止更新使能： 1：CCR和ARR影子停止更新使能 0：CCR和ARR影子寄存器维持其它设置
30	CRPE	R/W	0	Auto-reload 预装载使能： 1：CCR 寄存器使能 preload 0：CCR 寄存器不使能 preload
29	COMP3_BKEN	R/W	0	COMP3 作为刹车源使能： 1：使能 COMP3 作为刹车源 0：关闭 COMP3 作为刹车源
28	ADC_HDT	R/W	0	ADC 硬件触发功能使能： 1：ADC 硬件触发功能使能 0：ADC硬件触发功能禁止
27	OPA_BKEN	R/W	0	OPA 作为刹车源使能： 1：使能 OPA 作为刹车源 0：关闭 OPA 作为刹车源
26	COMP2_BKEN	R/W	0	COMP2 作为刹车源使能： 1：使能 COMP2 作为刹车源 0：关闭 COMP2 作为刹车源
25	COMP1_BKEN	R/W	0	COMP1 作为刹车源使能： 1：使能 COMP1 作为刹车源 0：关闭 COMP1 作为刹车源
24	COMP0_BKEN	R/W	0	COMP0 作为刹车源使能： 1：使能 COMP0 作为刹车源 0：关闭 COMP0 作为刹车源

比特	名称	属性	复位值	描述
23	LVD_BKEN	R/W	0	LVD 作为刹车源使能： 1：使能 LVD 作为刹车源 0：关闭 LVD 作为刹车源
22	BREAK2_SEL	R/W	0	**GTIMER0 IO 引脚刹车源选择：** 1：GTIMER0 选择 GTIMER2 配置的 IO 作为刹车源 0:GTIMER0 不选择 GTIMER2 配置的 IO 作为刹车源 **GTIMER1 IO 引脚刹车源选择：** 1：GTIMER1 选择 GTIMER2 配置的 IO 作为刹车源 0:GTIMER1 不选择 GTIMER2 配置的 IO 作为刹车源 **GTIMER2 IO 引脚刹车源选择：** 1：GTIMER2 选择 GTIMER1 配置的 IO 作为刹车源 0:GTIMER2 不选择 GTIMER1 配置的 IO 作为刹车源
21	BREAK1_SEL	R/W	0	**GTIMER0 IO 引脚刹车源选择：** 1：GTIMER0 选择 GTIMER1 配置的 IO 作为刹车源 0:GTIMER0 不选择 GTIMER1 配置的 IO 作为刹车源 **GTIMER1 IO 引脚刹车源选择：** 1：GTIMER1 选择 GTIMER0 配置的 IO 作为刹车源 0:GTIMER1 不选择 GTIMER0 配置的 IO 作为刹车源 **GTIMER2 IO 引脚刹车源选择：** 1：GTIMER2 选择 GTIMER0 配置的 IO 作为刹车源 0:GTIMER2 不选择 GTIMER0 配置的 IO 作为刹车源
20	PWMN_IDLE	R/W	0	PWM 输出负向电平 IDLE 状态： 1：电平为高电平 0：电平为低电平
19	PWMP_IDLE	R/W	0	PWM 输出正向电平 IDLE 状态： 1：电平为高电平 0：电平为低电平
18	MOE	R/W	0	输出使能： 1：输出总使能 0：输出禁止
17:16	PWMN_B_S	R/W	0	PWM 刹车触发后，PWM 互补电平状态设置位： 00：低电平 01：高电平 10/11：高阻状态

比特	名称	属性	复位值	描述
15:14	PWMS_B_S	R/W	0	PWM 刹车触发后，PWM 正向电平状态设置位： 00：低电平 01：高电平 10/11：高阻状态
13	SOFT_BK	R/W	0	软件触发刹车功能设置位： 写 1：软件触发刹车功能 0：软件不触发刹车功能
12	CEN_ALL	W	0	GTIMER0 1：同时使能 GTIMER0/1/2 和 LPTIMER0/1/2/3 ， ATIMER ，BTIMER0/1/2/3 后，以上定时器 CEN 位同时为 1 0：无操作； 读此位始终为 0
11	BKE_POL	R/W	0	刹车信号极性配置： 1：刹车信号低电平有效 0：刹车信号高电平有效
10	BKE	R/W	0	刹车功能使能： 1：刹车功能使能 0：刹车功能禁止
9	PWM_DEAD	R/W	0	PWM 死区插入功能使能： 1：避免死区功能使能 0：避免死区功能关闭
8	PWM_INV	R/W	0	互补 PWM 与原 PWM 差分使能： 1：互补 PWM 和原 PWM 反相位 0：互补 PWM 和原 PWM 同相位
7	MMS	R/W	0	主机模式选择，用于配置主机模式下向从机发送的同步触发信号（TRGO）源： 1：UE（update event）信号被用作 TRGO 0：OC1REF 用作 TRGO
6	ARPE	R/W	0	Auto-reload 预装载使能： 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload
5:4	CMS	R/W	0	计数器对齐模式选择： 00：边沿对齐模式 01：中央对齐模式 1，输出比较中断标志仅在计数器向下计数的过程中置位 10：中央对齐模式 2，输出比较中断标志仅在计数器向上计数的过程中置位 11：中央对齐模式 3，输出比较中断标志在计数器向上向下计数的过程中都会置位

比特	名称	属性	复位值	描述
3	CEN_ALL_EN	R/W	0	CEN_ALL 使能： 1：当前 GTIMER 可以被 CEN_ALL 控制 0：当前 GTIMER 对 CEN_ALL 信号无效
2	DIR	R/W	0	计数方向寄存器： 1：向下计数 0：向上计数 注意：当定时器配置为中央计数模式时，此寄存器只读
1	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭

注：

- PWMN_B_S 和 PWMS_B_S 同设置为高电平或低电平时，触发刹车后，PWM 输出会概率出现 2us 的电平跳变行为。
- BREAK2_SEL 和 BREAK1_SEL 这两个 bit 配置只能为 00b，01b，10b，当为 11b 时，GTIMER0/1 是 BREAK1_SEL 有效，GTIMER2 是 BREAK2_SEL 有效。

17.3.2 GTIMER 中断使能寄存器 GTIM_IER (偏移：04h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	BKE_IE	R/W	0	刹车中断使能： 1：刹车中断使能 0：刹车中断禁止
1	CCIE	R/W	0	捕捉/比较通道中断使能： 0：禁止捕捉/比较中断 1：允许捕捉/比较中断
0	UIE	R/W	0	Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

17.3.3 GTIMER 状态寄存器 GTIM_SR (偏移：08h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	BKE_F	R/W1C	0	刹车中断标志： 1：处于刹车状态 0：未处于刹车状态 写1清0。
1	CCIF	R/W1C	0	捕捉/比较通道中断标志： 如果 CC 通道配置为输出：CCIF 在计数值等于比较值时置位，软件写 1 清零。 如果CC通道配置为输入：发生捕捉事件时置位，软件写1清零，或者软件读GTIM_CCR自动清零。
0	UIF	R/W1C	0	Update 事件中断标志： 硬件置位，软件写 1 清零。 当发生更新事件时，UIF置位，并更新shadow寄存器。

17.3.4 GTIMER 事件产生寄存器 GTIM_EGR (偏移：0Ch)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	UG	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零。 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

17.3.5 GTIMER 捕捉/比较模式寄存器 GTIM_CCMR (偏移：10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	CAPCLR	R/W	0	首次捕捉清零控制位： 1：计数器使能后保持为0，直到捕捉到第一个沿开始计数 0：计数器使能后立即计数
14:13	ICPSC	R/W	0	捕捉源预分频位： 00：除1 01：除2 10：除4 11：除8

比特	名称	属性	复位值	描述
12	CAPFLT	R/W	0	输入捕捉信号滤波使能： 1：有输入滤波功能 0：无输入滤波功能
11:10	CAPEDGE	R/W	0	捕获沿触发控制位： 00：上升沿触发 01：下降沿触发 10/11：上升或下降沿触发
9:7	CAPSSEL	R/W	0	捕捉源选择位： **GTIMER0** 000：GTIM0_CH 001：UART0_RX 010：LPTIM0_LPOUT 011：CLK32K_GTIMER0 100：COMP0 101：COMP1 110：COMP2 111：COMP3 **GTIMER1** 000：GTIM1_CH 001：UART1_RX 010：LPTIM1_LPOUT 011：CLK32K_GTIMER1 100：COMP0 101：COMP1 110：COMP2 111：COMP3 **GTIMER2** 000：GTIM2_CH 001：I2C_SCL 010：LPTIM2_LPOUT 011：CLK32K_GTIMER2 100：COMP0 101：COMP1 110：COMP2 111：COMP3
6	TEDGE	R/W	0	计数源边沿选择： 1：下降沿计数 0：上升沿计数
5	RSV	-	-	保留
4	TFLT	R/W	0	外部计数源滤波使能： 1：有滤波功能 0：无滤波功能

比特	名称	属性	复位值	描述
3:1	TSSEL	R/W	0	计数源选择位： **GTIMER0** 000：APBCLK（PCLK） 001：GTIMER2_TRGO（GTIMER2同步触发信号） 010：CLK32K_GTIMER0（32k时钟） 011：GTIMER0_CH（GTIMER0捕获输入） 100：COMP0 101：COMP1 110：COMP2 111：COMP3 **GTIMER1** 000：APBCLK（PCLK） 001：GTIMER0_TRGO（GTIMER0同步触发信号） 010：CLK32K_GTIMER1（32k时钟） 011：GTIM1_CH（GTIMER1捕获输入） 100：COMP0 101：COMP1 110：COMP2 111：COMP3 **GTIMER2** 000：APBCLK（PCLK） 001：GTIMER1_TRGO（GTIMER1同步触发信号） 010：CLK32K_GTIMER1（32k时钟） 011：GTIM2_CH（GTIMER2捕获输入） 100：COMP0 101：COMP1 110：COMP2 111：COMP3
0	CCS	R/W	0	捕捉/比较 1 通道选择： 1：CC 通道配置为输入 0：CC 通道配置为输出 注意：CCS仅在通道关闭时（CCE=0）可以写

注： TSSEL 计数源选择位，若设置 11b：GTIM0_CH 捕获输入，即通过 GTIM0_CH 的 GPIO 获取时钟输入，若此时同时设置 GTIMx_CH 和 GTIMx_CHN 作为 PWM 输出，则 GTIMx_CH 会有冲突，按照 PA-PG 的顺序，靠前的 GPIO 复用功能优先。一般应用不这么使用，需要注意此点。

17.3.6 GTIMER 捕捉/比较使能寄存器 GTIM_CCER (偏移：14h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	CCP	R/W	0	CC 通道配置为输出时极性： 1：CNT>CCR 时输出高电平 0：CNT<CCR 时输出高电平
0	CCE	R/W	0	捕捉/比较输出使能： **CC 通道配置为输出时：** 1：OC 有输出 0：OC 无输出 **CC 通道配置为输入时：** 1：使能捕捉功能 0：关闭捕捉功能

17.3.7 GTIMER 计数寄存器 GTIM_CNT (偏移：18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT	R	0	计数器值

17.3.8 GTIMER 预分频寄存器 GTIM_PSC(偏移：1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC	R/W	0	计数器时钟（CK_CNT）预分频值： $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器。 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

注：不使能 preload，依旧要等到有 update 事件才能使 psc 值载入 shadow 寄存器

17.3.9 GTIMER 自动重载寄存器 GTIM_ARR (偏移：20h)

比特	名称	属性	复位值	描述
31:16	ARRN	R/W	0	计数溢出时的自动重载值，互补计数器 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器

比特	名称	属性	复位值	描述
15:0	ARR	R/W	0	计数溢出时的自动重载值： 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器。 注：支持的最高的PWM输出为 12MHz，配置的PSC和ARR需注意该条件。

17.3.10 GTIMER 捕捉/比较寄存器 GTIM_CCR (偏移：24h)

比特	名称	属性	复位值	描述
31:16	CCRN	R/W	0	捕捉/比较通道寄存器，互补计数器 **如果通道配置为输出：** 用于与计数器比较产生 OC 输出
15:0	CCR	R/W	0	捕捉/比较通道寄存器 **如果通道配置为输出：** 用于与计数器比较产生 OC 输出。注意：中央对齐模式中，当此值设置和 ARR 相等时，向上计数过程中无法产生 CCIF 中断，只有向下计数过程会产生中断。 **如果通道配置为输入：** CCR 保存最近一次输入捕捉事件发生时的计数器值，此时 CCR 为只读

注：CCR 寄存器不是 preload 寄存器，写入后将立即生效，需注意配置时机。

17.3.11 GTIMER 硬件触发寄存器 GTIM_CARS1(偏移：28h)

比特	名称	属性	复位值	描述
31:16	ARRS1	R/W	0	ADC 硬件触发时间设置： 此位用来设置 ADC 硬件触发点。 通过此组寄存器，设置一组PWM时间点，通过PWM的上升沿和下降沿（有效沿在ADC中可以设置），来硬件触发ADC

比特	名称	属性	复位值	描述
15:0	CCRS1	R/W	0	ADC 硬件触发时间设置： 此位用来设置 ADC 硬件触发点。 通过此组寄存器，设置一组PWM时间点，通过PWM的上升沿和下降沿（有效沿在ADC中可以设置），来硬件触发ADC

17.4 使用说明

17.4.1 计数器模式

- 往上计数
  - 在往上计数模式中，counter 从 0 计数到自动重载值，然后重新到 0 开始计数，并产生中断。而且在此时，UEV 事件发生。
  - 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。
- 往下计数
  - 在往下计数模式中，counter 从自动重载值计数到 0，然后重新到自动重载值开始计数，并产生中断。而且在此时，UEV 事件发生。
  - 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。
- 中央对齐模式（上下计数）
  - 在中央对齐模式中，counter 从 0 计数到自动重载值-1，产生中断；然后又从自动重载值计数到 1，产生中断；然后又从 0 开始计数。当 counter 处于中央对齐模式时，DIR 寄存器无效。
  - 每次向上溢出和向下溢出时，UEV 事件发生。
  - 当 UEV 事件发生时，芯片内部加载寄存器才会被更新。

17.4.2 输入捕获模式

在输入捕获模式中，当在相应的 ICx 信号出现触发沿的时候，捕捉寄存器（CCR）会把当时的 counter 值保存下来。当一次捕获发生后，相应的中断标志被置位，同时产生一次捕获中断。CCIF 由软件清 0。触发变化沿可以由寄存器控制是上升沿或下降沿。捕捉源可以选择滤波或不滤波

17.4.3 PWM 模式

PWM 模式可以产生波形，其频率取决于 ARR 寄存器和 PSC，而占空比取决于 CCR 寄存器。

- PWM 边沿对齐模式
  - 向上计数

    向上计数的情况下，配置 GTIM_CCER[1]为 0 时，OCxREF 信号在 CNT<CCR 时为高电平，否则为低电平。如果 CCR 值大于 ARR 值，则 OCxREF 被锁定为 1；如果 CRR 为 0，则 OCxREF 被固定为 0。

  - 向下计数

    向下计数的情况下，配置 GTIM_CCER[1]为 0 时，OCxREF 信号在 CNT>CCR 时为低电平，否则为高电平。如果 CCR 值大于 ARR 值，则 OCxREF 被锁定为 1；如果 CRR 为 0，则 OCxREF 被固定为 0。

- PWM 中央对齐模式

根据 GTIM_CR[5:4]位的配置，比较标志可以在计数器向上计数时置 1、比较标志可以在计数器向下计数时置 1、比较标志可以在计数器向上和向下计数时置 1。

17.4.4 互补输出和死区插入

互补输出：GTIMER0/1/2 均可输出两路互补 PWM，配置 GTIM_CCR[15:0]和 GTIM_CCR[31:16]，使能 GTIM_CR[9]后，可输出互补信号 OCx 和 OCxN。(注：使能了 GTIM_CR[9]后，会以 CCRN 和 CCR 的最小值去做 CCR，以 ARRN 和 ARR 的最大值做 ARR)。

死区插入：死区时间由 GTIM_ARR[15:0]和 GTIM_ARR[31:16]的差值、GTIM_CCR[15:0]和 GTIM_CCR[31:16]的差值决定。(注：当 ARR=2 CCR=1 这种情况下不能调节出死区)。

17.4.5 刹车功能

可使用软件刹车或硬件刹车功能，且可选择多种片内外设触发刹车功能。刹车触发后 PWM 的极性状态由 GTIM_CR[17:16]和 GTIM_CR[15:14]决定。刹车生效后，GTIM_CR[18]会被置 0，若需要重新输出 PWM，则需要重新手动将 GTIM_CR[18]置 1。

17.5 使用流程

若想将 GTIM_ARR、GTIM_CCR、GTIM_PSC 的值立即载入到 shadow 寄存器中，则写入后手动将 GTIM_EGR[0]写 1 触发 Update 事件，并清除 GTIM_SR[0]状态。使能了 GTIM_CR[6]后，新的 GTIM_ARR、GTIM_CCR、GTIM_PSC 值会在触发 Update 事件后才会载入到 shadow 寄存器中。

17.5.1 普通定时器

1. 配置 GTIM_CCMR[3:1]，选择计数时钟源。
2. 配置 GTIM_ARR[15:0]，设置重载值。
3. 配置 GTIM_PSC[15:0]，设置预分频值。
4. 配置 GTIM_EGR[0]为 1，手动产生 Update 事件将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 GTIM_SR[0]。
5. 配置 GTIM_CR[2]和 GTIM_CR[5:4]，设置计数方向和计数器对齐模式。
6. 选择配置 GTIM_IER[0]中断使能。
7. 配置 GTIM_CR[0]，启动 GTIMER 计数。

17.5.2 PWM 输出

1. 根据 IO 复用关系，将 IO 复用为 GTIM_CH 和 GTIM_CHN。
2. 配置 GTIM_CCMR[3:1]，选择计数时钟源。
3. 配置 GTIM_ARR[15:0]和 GTIM_ARR[31:16]，设置重载值和互补重载值(注：这两个值本身并不互补)。
4. 配置 GTIM_CCR[15:0]和 GTIM_CCR[31:16]，设置比较值和互补比较值(注：同上)。
5. 配置 GTIM_PSC[15:0]，设置预分频值。
6. 配置 GTIM_EGR[0]为 1，手动产生 Update 事件将 CCR、ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 GTIM_SR[0]。
7. 配置 GTIM_CR[8]，设置互补 PWM 和原 PWM 的相位。
8. 若步骤 7 配置为反相位，则必须配置 GTIM_CR[9]，使能死区功能。
9. 配置 GTIM_CR[2]和 GTIM_CR[5:4]，设置计数方向和计数器对齐模式。
10. 配置 GTIM_CCMR[0]为 0，设置通道为输出。
11. 配置 GTIM_CCER[1]，设置通道输出极性。
12. 配置 GTIM_CCER[0]为 1，使能 OC 通道输出。
13. 选择配置 GTIM_IER[1]中断使能。
14. 配置 GTIM_CR[0]，启动 Gtimer 计数。
15. 配置 GTIM_CR[18]为 1，使能总输出。

17.5.3 输入捕获

1. 根据 IO 复用关系，将 IO 复用为 GTIM_CH。
2. 配置 GTIM_CCMR[9:7]，设置捕捉源。

3. 配置 GTIM_CCMR[14:13]，设置捕捉源分频。
4. 配置 GTIM_CCMR[11:10]，设置捕捉源触发方式。
5. 配置 GTIM_CCMR[3:1]，选择计数时钟源。
6. 配置 GTIM_ARR[15:0]，设置重载值。
7. 配置 GTIM_PSC[15:0]，设置预分频值。
8. 配置 GTIM_EGR[0]为 1，手动产生 Update 事件将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 GTIM_SR[0]。
9. 配置 GTIM_CR[2]和 GTIM_CR[5:4]，设置计数方向和计数器对齐模式。
10. 配置 GTIM_CCMR[0]为 1，设置通道为输入。
11. 配置 GTIM_CCER[0]为 1，使能捕捉功能。
12. 若使用捕捉中断，配置 GTIM_IER[1]为 1，使能捕捉中断。
13. 配置 GTIM_CR[0]，启动 GTIMER 计数。

17.5.4 刹车功能

硬件刹车使用流程：

1. 配置 GTIM_CR[27]、GTIM_CR[26]、GTIM_CR[25]、GTIM_CR[24]、GTIM_CR[23]、GTIM_CR[21]和 GTIM_CR[22]，选择刹车源，可同时选择多个刹车源。
2. 根据刹车源，将 IO 复用为 GTIM_BK 或模拟接口。
3. 配置 GTIM_CR[11]，设置刹车信号极性。
4. 配置 GTIM_CR[17:16]和 GTIM_CR[15:14]，刹车触发后 PWM 的极性状态。
5. 配置 GTIM_CR[19]和 GTIM_CR[20]，刹车触发后 PWM 的 IDLE 极性状态。
6. 配置 GTIM_CR[10]为 1，使能刹车功能。
7. 配置 PWM 输出，见 19.5.2 PWM 输出章节。
8. 选择配置刹车中断，配置 GTIM_IER[2]为 1，使能刹车中断。
9. 配置 GTIM_CR[0]，启动 GTIMER 计数。
10. 配置 GTIM_CR[18]为 1，使能总输出。

软件刹车使用流程：

1. 配置 GTIM_CR[21]和 GTIM_CR[22]，选择刹车源。
2. 配置 GTIM_CR[11]，设置刹车信号极性。
3. 配置 GTIM_CR[17:16]和 GTIM_CR[15:14]，刹车触发后 PWM 的极性状态。
4. 配置 GTIM_CR[19]和 GTIM_CR[20]，刹车触发后 PWM 的 IDLE 极性状态。
5. 配置 GTIM_CR[10]为 1，使能刹车功能。
6. 配置 PWM 输出，见 19.5.2 PWM 输出章节。
7. 选择配置刹车中断，配置 GTIM_IER[2]为 1，使能刹车中断。

8. 配置 GTIM_CR[0]，启动 GTIMER 计数。
9. 配置 GTIM_CR[18]为 1，使能总输出。
10. 配置 GTIM_CR[13]为 1，软件触发刹车信号。

18 BTIMER0/1

18.1 概述

基本定时/计数器 BTIMER0 和 BTIMER1 共用同一个中断，包含多种用途，16bit 向上定时/计数器，产生输出 PWM 波形，脉冲输出，且计数值可以随时由软件设置和读取。

18.2 主要特性

- 16 位向上计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 单通道 PWM 输出比较、单脉冲输出
- 中断在以下几种情况产生：
  - 计数器向上溢出产生 UE 中断
  - 输出比较中断
- 支持定时器同步使能

18.3 寄存器描述

BTIMER0、BTIMER1 寄存器基地址：0x4000_2C00

表 18-1：BTIMER0/1 寄存器列表

偏置	名称	描述
0x00	BTIM0_CR0	BTIMER0 控制寄存器
0x04	BTIM01_DIER	BTIMER01 中断使能寄存器
0x08	BTIM01_SR	BTIMER01 原始中断状态寄存器
0x0C	BTIM0_EGR0	BTIMER0 事件产生寄存器
0x10	BTIM0_CNT0	BTIMER0 计数器寄存器
0x14	BTIM0_PSC0	BTIMER0 预分频寄存器
0x18	BTIM0_ARR0	BTIMER0 自动重载寄存器
0x1C	BTIM0_CCR0	BTIMER0 比较寄存器
0x20	BTIM1_CR1	BTIMER1 控制寄存器
0x24	BTIM1_EGR1	BTIMER1 事件产生寄存器
0x28	BTIM1_CNT1	BTIMER1 计数器寄存器
0x2C	BTIM1_PSC1	BTIMER1 预分频寄存器
0x30	BTIM1_ARR1	BTIMER1 自动重载寄存器
0x34	BTIM1_CCR1	BTIMER1 比较寄存器

18.3.1 BTIMER0 控制寄存器 BTIM0_CR0 (偏移：00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	MMS	R/W	0	1：输出 PWM 频率 0：输出 beep 频率
6	ARPE	R/W	0	Auto-reload 预装载使能 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload
5	RSV	-	-	保留
4	CEN_ALL	R/W	0	除 BTIMER0 外所有计数器使能 1：写此位后，BTIMER0 以外的其它 BTIMER，GTIMER0/1/2，LPTIMER0/1/2/3，ATIMER 的 CEN 位同时为 1，开始计数； 0：无操作 读此位始终为 0
3	CEN_ALL_EN	R/W	0	CEN_ALL 使能： 1：当前 BTIMER0 可以受除 BTIMER0 以外的其它 BTIMER，ATIMER，GTIMER0/1/2，LPTIMER0/1/2/3 的 CEN_ALL 控制 0：当前 BTIMER0 对 CEN_ALL 信号无效
2	CCP	R/W	0	TRGO 输出极性： 1：反向输出 0：正向输出
1	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭

18.3.2 BTIMER01 中断使能寄存器 BTIM01_DIER (偏移：04h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	CCIE1	R/W	0	BTIMER1 比较通道中断使能： 1：允许比较中断 0：禁止比较中断
2	UIE1	R/W	0	BTIMER1 Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

比特	名称	属性	复位值	描述
1	CCIE0	R/W	0	BTIMER0 比较通道中断使能： 1：允许比较中断 0：禁止比较中断
0	UIE0	R/W	0	BTIMER0 Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

18.3.3 BTIMER01 原始中断状态寄存器 BTIM01_SR (偏移：08h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	CCIF1	R/W1C	0	BTIMER1 比较通道中断标志 CCIF在计数值等于比较值时置位，软件写1清零。
2	UIF1	R/W1C	0	BTIMER1 Update 事件中断标志，硬件置位，软件写 1 清零。 当发生更新事件时，UIF置位，并更新shadow寄存器
1	CCIF0	R/W1C	0	BTIMER0 比较通道中断标志 CCIF 在计数值等于比较值时置位，软件写 1 清零。
0	UIF0	R/W1C	0	BTIMER0 Update 事件中断标志，硬件置位，软件写 1 清零。 当发生更新事件时，UIF置位，并更新shadow寄存器

18.3.4 BTIMER0 事件产生寄存器 BTIM0_EGR0 (偏移：0Ch)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	UG0	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

注：预分频计数器为内部对应 PSC 寄存器的一个计数器，该计数器不可见。

18.3.5 BTIMER0 计数器寄存器 BTIM0_CNT0 (偏移：10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT0	R	0	计数器值

18.3.6 BTIMER0 预分频寄存器 BTIM0_PSC0 (偏移：14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC0	R/W	0	计数器时钟（CK_CNT）预分频值 $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

注：使能或不使能 preload，该寄存器都要等到 update 事件才能将 psc 值载入 shadow 寄存器

18.3.7 BTIMER0 自动重载寄存器 BTIM0_ARR0 (偏移：18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR0	R/W	0	计数溢出时的自动重载值： ● 这是一个preload寄存器，当preload功能使能时，在update事件发生时其值被载入shadow寄存器。 ● 当preload功能不使能时，写入该寄存器的值会立刻被载入shadow寄存器。 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

18.3.8 BTIMER0 比较寄存器 BTIM0_CCR0 (偏移：1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR0	R/W	0	比较通道寄存器： ● 这是一个 preload 寄存器，当 preload 功能使能时，在 update 事件发生时其值被载入 shadow 寄存器。 ● 当 preload 功能不使能时，写入该寄存器的值会立刻被载入 shadow 寄存器。

注：CCR 寄存器不是 preload 寄存器，写入后将立即生效，需注意配置时机。

18.3.9 BTIMER1 控制寄存器 BTIM1_CR1 (偏移：20h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	MMS	R/W	0	1：输出 PWM 频率 0：输出 beep 频率
6	ARPE	R/W	0	Auto-reload 预装载使能： 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload
5	RSV	-	-	保留
4	CEN_ALL	R/W	0	除 BTIMER1 外所有计数器使能 1：写此位后，BTIMER1 以外的其它 BTIMER，GTIMER0/1/2，LPTIMER0/1/2/3，ATIMER 的 CEN 位同时为 1，开始计数。 0：无操作 读此位始终为 0。
3	CEN_ALL_EN	R/W	0	CEN_ALL 使能： 1：当前 BTIMER1 可以受除 BTIMER1 以外的其它 BTIMER，ATIMER，GTIMER0/1/2，LPTIMER0/1/2/3 的 CEN_ALL 控制。 0：当前 BTIMER1 对 CEN_ALL 信号无效。
2	CCP	R/W	0	TRGO 输出极性： 1：反向输出 0：正向输出
1	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭

18.3.10 BTIMER1 事件产生寄存器 BTIM1_EGR1 (偏移：0Ch)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留

比特	名称	属性	复位值	描述
0	UG1	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

注：预分频计数器为内部对应 PSC 寄存器的一个计数器，该计数器不可见。

18.3.11 BTIMER1 计数器寄存器 BTIM1_CNT1 (偏移：10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT1	R	0	计数器值

18.3.12 BTIMER1 预分频寄存器 BTIM1_PSC1 (偏移：14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC1	R/W	0	计数器时钟（CK_CNT）预分频值 $f_{CK\_CNT}=f_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器

注：使能或不使能 preload，该寄存器都要等到 update 事件才能将 psc 值载入 shadow 寄存器

18.3.13 BTIMER1 自动重载寄存器 BTIM1_ARR1 (偏移：18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR1	R/W	0	计数溢出时的自动重载值： ● 这是一个preload寄存器，当preload功能使能时，在update事件发生时其值被载入shadow寄存器。 ● 当preload功能不使能时，写入该寄存器的值会立刻被载入shadow寄存器。

18.3.14 BTIMER1 比较寄存器 BTIM1_CCR1 (偏移：1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留

比特	名称	属性	复位值	描述
15:0	CCR1	R/W	0	比较通道寄存器： 寄存器用于与计数器比较产生 OC 输出

注：CCR 寄存器不是 preload 寄存器，写入后将立即生效，需注意配置时机。

18.4 使用说明

18.4.1 计数器模式

基本定时器只支持向上计数模式。计数器使能后，CNT 计数器从 0 计数到自动重载值，产生溢出事件并且原始中断标志 UIF(Update Interrupt Flag)置位，然后重新从 0 开始计数。软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器(该计数器不可见)自动清零。设置 UG 寄存器会触发 UIF（Update Interrupt Flag）中断标志置位。当 update event 发生时，BTIM_ARR 和 BTIM_PSC 的值会更新到相应的 shadow 寄存器中。

18.4.2 PWM 模式

PWM 模式可以产生波形，其频率取决于 ARR 和 PSC 寄存器，而占空比取决于 CCR 寄存器。配置 BTIM_CR 寄存器中的 CCP 为 0 时，OCxREF 信号在 CNT<CCR 时为高电平，否则为低电平。当 BTIM_CR 寄存器中的 CCP 为 1 时，OCxREF 信号在 CNT<CCR 时为低电平，否则为高电平。如果 CCR 值大于 ARR 值，则 OCxREF 被锁定为 1；如果 CRR 为 0，则 OCxREF 被固定为 0。

18.4.3 蜂鸣器频率输出

当配置 BTIM_CR 寄存器中的 MMS 为 0 时，输出蜂鸣器频率：

$$Fbeep = \frac{Fck\_cnt}{2(\mathrm{ARR} + 1)}$$

18.5 使用流程

下文将以 BTIMER0 为例详细介绍软件使用流程。

若想将 BTIM0_ARR0、BTIM0_CCR0、BTIM0_PSC0 的值立即载入到 shadow 寄存器中，则写入后手动将 BTIM0_EGR0 写 1 触发 Update 事件，并清除 BTIM01_SR[0]状态。使能了 BTIM0_CR0[6]后，新的 BTIM0_ARR0、BTIM0_CCR0、BTIM0_PSC0 值会在触发 update event 后才会载入到 shadow 寄存器中。

18.5.1 普通定时器（以 BTIMER0 为例）

1. 配置 BTIM0_ARR0，设置重载值。
2. 配置 BTIM0_PSC0，设置预分频值。
3. 配置 BTIM0_EGR0 为 1，手动产生 update event 将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 BTIM01_SR[0]。
4. 配置 BTIM0_CR0[1]为 0，即 UE 事件发生时计数器不停止。
5. 使能 BTIM0_CR0[6]为 1，即使能 preload 功能。
6. 选择配置 BTIM01_DIER[0]中断使能。
7. 配置 BTIM0_CR0[0]，启动 Btimer 计数。

18.5.2 PWM 输出（以 BTIMER0 为例）

1. 根据 IO 复用关系，将 IO 复用为 BTIMER_OUT。
2. 配置 BTIM0_ARR0，设置重载值。
3. 配置 BTIM0_CCR0，设置比较值。
4. 配置 BTIM0_PSC0，设置预分频值。
5. 配置 BTIM0_EGR0[0]为 1，手动产生 update event 将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 BTIM01_SR[0]。
6. 配置 BTIM0_CR0[1]为 0，即 UE 事件发生时计数器不停止。
7. 配置 BTIM0_CR0[2]为 0，即 TRGO 为正向输出极性。
8. 配置 BTIM0_CR0[7]为 1，使能输出 PWM 波形。
9. 使能 BTIM0_CR0[6]为 1，即使能 preload 功能。
10. 选择配置 BTIM01_DIER[1]中断使能。
11. 配置 BTIM0_CR0[0]，启动 BTIMER 计数。

19 BTIMER2/3

19.1 概述

基本定时/计数器 BTIMER2 和 BTIMER3 共用同一个中断，包含多种用途，16bit 向上定时/计数器，产生输出 PWM 波形，脉冲输出，且计数值可以随时由软件设置和读取。

19.2 主要特性

- 16 位向上计数自动重载计数器
- 16 位可编程预分频器，支持实时调整计数时钟分频
- 单通道 PWM 输出比较、单脉冲输出
- 中断在以下几种情况产生：
  - 计数器向上溢出产生 UE 中断
  - 输出比较中断
- 支持定时器同步使能

19.3 寄存器描述

BTIMER2、BTIMER3 寄存器基地址：0x4000_7000

表 19-1：BTIMER2/3 寄存器列表

偏置	名称	描述
0x00	BTIM2_CR2	BTIMER2 控制寄存器
0x04	BTIM23_DIER	BTIMER23 中断使能寄存器
0x08	BTIM23_SR	BTIMER23 原始中断状态寄存器
0x0C	BTIM2_EGR2	BTIMER2 事件产生寄存器
0x10	BTIM2_CNT2	BTIMER2 计数器寄存器
0x14	BTIM2_PSC2	BTIMER2 预分频寄存器
0x18	BTIM2_ARR2	BTIMER2 自动重载寄存器
0x1C	BTIM2_CCR2	BTIMER2 比较寄存器
0x20	BTIM3_CR3	BTIMER3 控制寄存器
0x24	BTIM3_EGR3	BTIMER3 事件产生寄存器
0x28	BTIM3_CNT3	BTIMER3 计数器寄存器
0x2C	BTIM3_PSC3	BTIMER3 预分频寄存器
0x30	BTIM3_ARR3	BTIMER3 自动重载寄存器
0x34	BTIM3_CCR3	BTIMER3 比较寄存器

19.3.1 BTIMER2 控制寄存器 BTIM2_CR2 (偏移：00h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	MMS	R/W	0	1：输出 PWM 频率 0：输出 beep 频率
6	ARPE	R/W	0	Auto-reload 预装载使能： 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload
5	RSV	-	-	保留
4	CEN_ALL	R/W	0	除 BTIMER2 外所有计数器使能： 1：写此位后，BTIMER2 以外的其它 BTIMER，GTIMER0/1/2，LPTIMER0/1/2/3，ATIMER 的 CEN 位同时为 1，开始计数； 0：无操作 读此位始终为 0
3	CEN_ALL_EN	R/W	0	CEN_ALL 使能： 1：当前 BTIMER2 可以受除 BTIMER2 以外的其它 BTIMER，ATIMER，GTIMER0/1/2，LPTIMER0/1/2/3 的 CEN_ALL 控制； 0：当前 BTIMER2 对 CEN_ALL 信号无效；
2	CCP	R/W	0	TRGO 输出极性： 1：反向输出 0：正向输出
1	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭

19.3.2 BTIMER23 中断使能寄存器 BTIM23_DIER (偏移：04h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	CCIE3	R/W	0	BTIMER3 比较通道中断使能： 1：允许比较中断 0：禁止比较中断
2	UIE3	R/W	0	BTIMER3 Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

比特	名称	属性	复位值	描述
1	CCIE2	R/W	0	BTIMER2 比较通道中断使能： 1：允许比较中断 0：禁止比较中断
0	UIE2	R/W	0	BTIMER2 Update 事件中断使能： 1：允许 Update 事件中断 0：禁止 Update 事件中断

19.3.3 BTIMER23 原始中断状态寄存器 BTIM23_SR (偏移：08h)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	CCIF3	R/W1C	0	BTIMER3 比较通道中断标志： CCIF在计数值等于比较值时置位，软件写1清零。
2	UIF3	R/W1C	0	BTIMER3 Update 事件中断标志： 硬件置位，软件写 1 清零。 当发生更新事件时，UIF置位，并更新shadow寄存器
1	CCIF2	R/W1C	0	BTIMER2 比较通道中断标志： CCIF 在计数值等于比较值时置位，软件写 1 清零。
0	UIF2	R/W1C	0	BTIMER2 Update 事件中断标志： 硬件置位，软件写 1 清零。 当发生更新事件时，UIF置位，并更新shadow寄存器

19.3.4 BTIMER2 事件产生寄存器 BTIM2_EGR2 (偏移：0Ch)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	UG2	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零。 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

注：预分频计数器为内部对应 PSC 寄存器的一个计数器，该计数器不可见。

19.3.5 BTIMER2 计数器寄存器 BTIM2_CNT2 (偏移：10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT2	R	0	计数器值

19.3.6 BTIMER2 预分频寄存器 BTIM2_PSC2 (偏移：14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC2	R/W	0	计数器时钟（CK_CNT）预分频值： $F_{CK\_CNT}=F_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器

注：使能或不使能 preload，该寄存器都要等到 update 事件才能将 psc 值载入 shadow 寄存器

19.3.7 BTIMER2 自动重载寄存器 BTIM2_ARR2 (偏移：18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	ARR2	R/W	0	计数溢出时的自动重载值： ● 这是一个preload寄存器，当preload功能使能时，在update事件发生时其值被载入shadow寄存器。 ● 当preload功能不使能时，写入该寄存器的值会立刻被载入shadow寄存器。

19.3.8 BTIMER2 比较寄存器 BTIM2_CCR2 (偏移：1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR2	R/W	0	比较通道寄存器： ● 这是一个 preload 寄存器，当 preload 功能使能时，在 update 事件发生时其值被载入 shadow 寄存器。 ● 当 preload 功能不使能时，写入该寄存器的值会立刻被载入 shadow 寄存器。

注：CCR 寄存器不是 preload 寄存器，写入后将立即生效，需注意配置时机。

19.3.9 BTIMER3 控制寄存器 BTIM3_CR3 (偏移：20h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7	MMS	R/W	0	1：输出 PWM 频率 0：输出 beep 频率
6	ARPE	R/W	0	Auto-reload 预装载使能： 1：ARR 寄存器使能 preload 0：ARR 寄存器不使能 preload

比特	名称	属性	复位值	描述
5	RSV	-	-	保留
4	CEN_ALL	R/W	0	除 BTIMER3 外所有计数器使能： 1：写此位后，BTIMER3 以外的其它 BTIMER，GTIMER0/1/2，LPTIM0/1/2/3，ATIMER 的 CEN 位同时为 1，开始计数 0：无操作 读此位始终为 0
3	CEN_ALL_EN	R/W	0	CEN_ALL 使能： 1：当前 BTIMER3 可以受除 BTIMER3 以外的其它 BTIMER，ATIMER，GTIMER0/1/2，LPTIMER0/1/2/3 的 CEN_ALL 控制。 0：当前 BTIMER3 对 CEN_ALL 信号无效。
2	CCP	R/W	0	TRGO 输出极性： 1：反向输出 0：正向输出
1	OPM	R/W	0	单脉冲输出模式： 1：Update Event 发生时计数器停止（自动清零 CEN） 0：Update Event 发生时计数器不停止
0	CEN	R/W	0	计数器使能： 1：计数器使能 0：计数器关闭

19.3.10 BTIMER3 事件产生寄存器 BTIM3_EGR3 (偏移：0Ch)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	UG3	W	0	软件 Update 事件，软件置位此寄存器产生 Update 事件，硬件自动清零 软件置位UG时会重新初始化计数器并更新shadow寄存器，预分频计数器被清零。

注：预分频计数器为内部对应 PSC 寄存器的一个计数器，该计数器不可见。

19.3.11 BTIMER3 计数器寄存器 BTIM3_CNT3 (偏移：10h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CNT3	R	0	计数器值

19.3.12 BTIMER3 预分频寄存器 BTIM3_PSC3 (偏移：14h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	PSC3	R/W	0	计数器时钟（CK_CNT）预分频值： $F_{CK\_CNT}=F_{CK\_PSC}/(PSC[15:0]+1)$ 这是一个preload寄存器，在update事件发生时其内容被载入shadow寄存器。 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

注：使能或不使能 preload，该寄存器都要等到 update 事件才能将 psc 值载入 shadow 寄存器

19.3.13 BTIMER3 自动重载寄存器 BTIM3_ARR3 (偏移：18h)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	
15:0	ARR3	R/W	0	计数溢出时的自动重载值： ● 这是一个preload寄存器，当preload功能使能时，在update事件发生时其值被载入shadow寄存器。 ● 当preload功能不使能时，写入该寄存器的值会立刻被载入shadow寄存器。 注：支持的最高的PWM输出为12MHz，配置的PSC和ARR需注意该条件。

19.3.14 BTIMER3 比较寄存器 BTIM3_CCR3 (偏移：1Ch)

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CCR3	R/W	0	比较通道寄存器 寄存器用于与计数器比较产生 OC 输出

注：CCR 寄存器不是 preload 寄存器，写入后将立即生效，需注意配置时机。

19.4 使用说明

19.4.1 计数器模式

基本定时器只支持向上计数模式。计数器使能后，CNT 计数器从 0 计数到自动重载值，产生溢出事件并且原始中断标志 UIF(Update Interrupt Flag)置位，然后重新从 0 开始计数。软件可以通过设置 UG 寄存器直接触发 update event，此时 CNT 和预分频计数器(该计数器不可见)自动清零。设

置 UG 寄存器会触发 UIF（Update Interrupt Flag）中断标志置位。当 update event 发生时，BTIM_ARR 和 BTIM_PSC 的值会更新到相应的 shadow 寄存器中。

19.4.2 PWM 模式

PWM 模式可以产生波形，其频率取决于 ARR 和 PSC 寄存器，而占空比取决于 CCR 寄存器。配置 BTIM_CR 寄存器中的 CCP 为 0 时，OCxREF 信号在 CNT<CCR 时为高电平，否则为低电平。当 BTIM_CR 寄存器中的 CCP 为 1 时，OCxREF 信号在 CNT<CCR 时为低电平，否则为高电平。如果 CCR 值大于 ARR 值，则 OCxREF 被锁定为 1；如果 CRR 为 0，则 OCxREF 被固定为 0。

19.4.3 蜂鸣器频率输出

当配置 BTIM_CR 寄存器中的 MMS 为 0 时，输出蜂鸣器频率：

$$Fbeep = \frac{Fck\_cnt}{2(\mathrm{ARR} + 1)}$$

19.5 使用流程

下文将以 BTIMER2 为例详细介绍软件使用流程。

若想将 BTIM2_ARR2、BTIM2_CCR2、BTIM2_PSC2 的值立即载入到 shadow 寄存器中，则写入后手动将 BTIM2_EGR2 写 1 触发 Update 事件，并清除 BTIM23_SR[0]状态。使能了 BTIM2_CR2[6]后，新的 BTIM2_ARR2、BTIM2_CCR2、BTIM2_PSC2 值会在触发 update event 后才会载入到 shadow 寄存器中。

19.5.1 普通定时器（以 BTIMER2 为例）

1. 配置 BTIM2_ARR2，设置重载值。
2. 配置 BTIM2_PSC2，设置预分频值。
3. 配置 BTIM2_EGR2[0]为 1，手动产生 update event 将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 BTIM23_SR[0]。
4. 配置 BTIM2_CR2[1]为 0，即 UE 事件发生时计数器不停止。
5. 使能 BTIM2_CR[6]为 1，即使能 preload 功能。
6. 选择配置 BTIM23_DIER[0]中断使能。
7. 配置 BTIM2_CR2[0]，启动 Btimer 计数。

19.5.2 PWM 输出（以 BTIMER2 为例）

1. 根据 IO 复用关系，将 IO 复用为 BTIMER_OUT。
2. 配置 BTIM2_ARR2，设置重载值。
3. 配置 BTIM2_CCR2，设置比较值。
4. 配置 BTIM2_PSC2，设置预分频值。
5. 配置 BTIM2_EGR2 为 1，手动产生 update event 将 ARR 和 PSC 的值立即载入到 shadow 寄存器，并清除 BTIM23_SR[0]。
6. 配置 BTIM2_CR2[1]为 0，即 UE 事件发生时计数器不停止。
7. 配置 BTIM2_CR2[2]为 0，即 TRGO 为正向输出极性。
8. 配置 BTIM2_CR2[7]为 1，使能输出 PWM 波形。
9. 使能 BTIM2_CR[6]为 1，即使能 preload 功能。
10. 选择配置 BTIM23_DIER[1]中断使能。
11. 配置 BTIM2_CR2[0]，启动 BTIMER 计数。

20 LPTIMER0/1

20.1 概述

LPTIMER0 和 LPTIMER1 是两个 32 位的低功耗定时/计数器模块。由于其时钟源具有多样性，因此能够在所有电源模式下保持运行状态，并且只消耗很低的功耗。LPTIMER0 和 LPTIMER1 可以在没有内部时钟的条件下工作，实现休眠模式下的外部脉冲计数功能，还可以与外部输入的触发信号结合，可以实现低功耗超时唤醒功能。

20.2 主要特性

- 独立的 32bit 向上计数器
- 3bit 异步时钟预分频器，8 种分频系数（1、2、4、8、16、32、64、128）
- 可选工作时钟：
  - 内部时钟源：LSCLK（CLK32K）、RCLP(CLK1HZ)、PCLK
  - 外部时钟源：LPTIN（带有模拟滤波）
- 32bit 比较/捕捉寄存器
- 32bit 目标值寄存器
- 连续/单触发模式
- 输入极性选择
- 无时钟外部脉冲计数
- 外部触发的休眠超时唤醒
- 支持 PWM 输出

20.3 寄存器描述

LPTIMER0、LPTIMER1 寄存器基地址：0x4000_1000

表 20-1：LPTIMER0/1 寄存器列表

偏置	名称	描述
0x00	LPTIM0_CR1	LPTIMER0 控制寄存器 1
0x04	LPTIM0_CR2	LPTIMER0 控制寄存器 2
0x08	LPTIM01_IER	LPTIMER01 中断使能寄存器
0x0C	LPTIM01_SR	LPTIMER01 中断标志寄存器
0x10	LPTIM0_CNT1	LPTIMER0 计数值寄存器 1
0x14	LPTIM0_CCMR1	LPTIMER0 捕捉比较配置寄存器 1
0x18	LPTIM0_CCMR2	LPTIMER0 捕捉比较配置寄存器 2

偏置	名称	描述
0x1C	LPTIM0_ARR1	LPTIMER0 自动重装载寄存器 1
0x20	LPTIM0_CCR1	LPTIMER0 捕捉比较寄存器 1
0x24	LPTIM0_CCR2	LPTIMER0 捕捉比较寄存器 2
0x28	LPTIM0_LOAD1	LPTIMER0 计数值 load 寄存器 1
0x2C	LPTIM0_BUFFER1	LPTIMER0 计数缓存寄存器 1
0x30	LPTIM1_CR3	LPTIMER1 控制寄存器 3
0x34	LPTIM1_CR4	LPTIMER1 控制寄存器 4
0x38	LPTIM1_CNT2	LPTIMER1 计数值寄存器 2
0x3C	LPTIM1_CCMR3	LPTIMER1 捕捉比较配置寄存器 3
0x40	LPTIM1_CCMR4	LPTIMER1 捕捉比较配置寄存器 4
0x44	LPTIM1_ARR2	LPTIMER1 自动重转载寄存器 2
0x48	LPTIM1_CCR3	LPTIMER1 捕捉比较寄存器 3
0x4C	LPTIM1_CCR4	LPTIMER1 捕捉比较寄存器 4
0x50	LPTIM1_LOAD2	LPTIMER1 计数值 load 寄存器 2
0x54	LPTIM1_BUFFER2	LPTIMER1 计数缓存寄存器 2

20.3.1 LPTIMER0 控制寄存器 1 LPTIM0_CR1（偏移：00h）

比特	名称	属性	默认值	功能描述
31:2	RSV	-	-	保留
1	CEN_ALL	R/W	0	CEN_ALL 控制使能： 1：当前 LPTIMER 计数开始受 CEN_ALL 控制； 0：当前 LPTIMER 计数不受 CEN_ALL 控制
0	EN	R/W	0	LPTIMER0 使能位： 1：使能计数器计数 0：禁止计数器计数

20.3.2 LPTIMER0 控制寄存器 2 LPTIM0_CR2（偏移：04h）

比特	名称	属性	默认值	功能描述
31:15	RSV	-	-	保留
14:12	CAP1SRSEL	R/W	0	通道 1 捕捉信号源选择： 000：LPT_CH1 输入 001：RCLP 010：COMP0_IN 011：COMP1_IN 100：COMP2_IN 101：COMP3_IN 其它：OPA_IN

比特	名称	属性	默认值	功能描述
11:9	DIVCFG	R/W	0	计数时钟分频选择： 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
8:6	CLKSEL	R/W	0	时钟源选择： 000：LSCLK（由 SYSREG->SYSCTRL0[12]选择的系统低速时钟 CLK32K，即 RCL 或 XTL）作为计数时钟 001：RCLP 作为计数时钟（RTC_1Hz） 010：PCLK 的门控时钟（PCLK）作为计数时钟 011：LPTIN 输入作为计数时钟 100：COMP0_IN 101：COMP1_IN 110：COMP2_IN 111：COMP3_IN
5	EDGESEL	R/W	0	ETR 计数时钟输入边沿选择： 1：ETR 的下降沿计数 0：ETR 的上升沿计数
4	SINGLE	R/W	0	单次计数模式使能： 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断； 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。
3:2	TRIGEDGE	R/W	00	ETR 触发边沿选择： 00: ETR 输入信号上升沿触发 01: ETR 输入信号下降沿触发 10/11: ETR 输入信号上升下降沿触发
1:0	LPMOD	R/W	00	工作模式选择： 00：计数模式 01：ETR 脉冲触发计数模式 10：ETR 脉冲计数模式 11：Timeout 模式

20.3.3 LPTIMER01 中断使能寄存器 LPTIM01_IER（偏移：08h）

比特	名称	属性	默认值	功能描述
31:8	RSV	-	-	保留
7	TIE1	R/W	0	LPTIMER1 触发事件中断使能： 1：使能触发事件中断 0：禁止触发事件中断
6	CC2IE1	R/W	0	LPTIMER1 捕捉/比较通道 2 中断使能： 1：使能捕捉/比较通道 2 中断 0：禁止捕捉/比较通道 2 中断
5	CC1IE1	R/W	0	LPTIMER1 捕捉/比较通道 1 中断使能： 1：使能捕捉/比较通道 1 中断 0：禁止捕捉/比较通道 1 中断
4	UIE1	R/W	0	LPTIMER1 更新中断使能： 1：使能更新事件中断 0：禁止更新事件中断
3	TIE	R/W	0	LPTIMER0 触发事件中断使能： 1：使能触发事件中断 0：禁止触发事件中断
2	CC2IE	R/W	0	LPTIMER0 捕捉/比较通道 2 中断使能： 1：使能捕捉/比较通道 2 中断 0：禁止捕捉/比较通道 2 中断
1	CC1IE	R/W	0	LPTIMER0 捕捉/比较通道 1 中断使能： 1：使能捕捉/比较通道 1 中断 0：禁止捕捉/比较通道 1 中断
0	UIE	R/W	0	LPTIMER0 更新中断使能： 1：使能更新事件中断 0：禁止更新事件中断

20.3.4 LPTIMER01 中断标志寄存器 LPTIM01_SR（偏移：0Ch）

比特	名称	属性	默认值	功能描述
31:12	RSV	-	-	保留
11	LPT1_CH2	R	0	LPTIMER1 捕捉通道 2 电平状态
10	LPT1_CH1	R	0	LPTIMER1 捕捉通道 1 电平状态
9	LPT0_CH2	R	0	LPTIMER0 捕捉通道 2 电平状态
8	LPT0_CH1	R	0	LPTIMER0 捕捉通道 1 电平状态
7	TIF1	R/W	0	LPTIMER1 触发中断标志，硬件置位，软件写 1 清零： 1：触发中断挂起 0：无触发事件
6	CC2IF1	R/W	0	LPTIMER1 捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零： 1：CNT 值和 CCR2 相等，或者发生输入捕捉事件 0：无比较或捕捉中断产生

比特	名称	属性	默认值	功能描述
5	CC1IF1	R/W	0	LPTIMER1 捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零： 1：CNT 值和 CCR1 相等，或者发生输入捕捉事件 0：无比较或捕捉中断产生
4	UIF1	R/W	0	LPTIMER1 更新中断标志，硬件置位，软件写 1 清零： 1：CNT 值等于 ARR 值产生中断 0：无更新中断产生
3	TIF	R/W	0	LPTIMER0 触发中断标志，硬件置位，软件写 1 清零： 1：触发中断挂起 0：无触发事件
2	CC2IF	R/W	0	LPTIMER0 捕捉/比较通道 2 中断标志，硬件置位，软件写 1 清零： 1：CNT 值和 CCR2 相等，或者发生输入捕捉事件 0：无比较或捕捉中断产生
1	CC1IF	R/W	0	LPTIMER0 捕捉/比较通道 1 中断标志，硬件置位，软件写 1 清零： 1：CNT 值和 CCR1 相等，或者发生输入捕捉事件 0：无比较或捕捉中断产生
0	UIF	R/W	0	LPTIMER0 更新中断标志，硬件置位，软件写 1 清零： 1：CNT 值等于 ARR 值产生中断 0：无更新中断产生

20.3.5 LPTIMER0 计数值寄存器 LPTM0_CNT1（偏移：10h）

比特	名称	属性	默认值	功能描述
31:0	CNT	R	0	计数器计数值

20.3.6 LPTIMER0 捕捉比较配置寄存器 1 LPTIM0_CCMR1（偏移：14h）

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP1EDGE	R/W	0	通道 1 捕捉边沿选择： 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：未定义
3	RSV	-	-	保留
2	CC1P	R/W	0	通道 1 比较输出波形极性选择： 1：CNT<=CCR1 时置高，CNT>CCR1 时为低 0：CNT<=CCR1 时置低，CNT>CCR1 时为高

比特	名称	属性	默认值	功能描述
1	CC1S	R/W	0	通道 1 捕捉/比较输出选择： 1：通道 1 配置为输入 0：通道 1 配置为输出
0	CC1E	R/W	0	通道 1 捕捉使能： 1：通道 1 捕获功能使能 0：通道 1 捕获功能禁止

20.3.7 LPTIMER0 捕捉比较配置寄存器 2 LPTIM0_CCMR2（偏移：18h）

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP2EDGE	R/W	0	通道 2 捕捉边沿选择： 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：未定义
3	RSV	-	-	保留
2	CC2P	R/W	0	通道 2 输出极性选择： 1：CNT<=CCR2 时置高，CNT>CCR2 时为低 0：CNT<=CCR2 时置低，CNT>CCR2 时为高
1	CC2S	R/W	0	通道 2 捕捉/比较选择： 1：通道 2 配置为输入 0：通道 2 配置为输出
0	CC2E	R/W	0	通道 2 捕捉使能： 1：通道 2 捕获功能使能 0：通道 2 捕获功能禁止

20.3.8 LPTIMER0 自动重载置寄存器 LPTIM0_ARR1(偏移：1Ch)

比特	名称	属性	默认值	功能描述
31:0	ARR1	R/W	0	自动重载目标寄存器： 当计数器计数值等于 ARR 时，计数器回到 0。 注：支持的最高的 PWM 输出为 12MHz，配置的 DIVCFG 和 ARR 需注意该条件。

20.3.9 LPTIMER0 捕捉比较寄存器 1 LPTIM0_CCR1(偏移：20h)

比特	名称	属性	默认值	功能描述
31:0	CCR1	R/W	0	捕捉/比较值寄存器 1 当 ARR=CCR1 时，以 CCR1 为准

20.3.10 LPTIMER0 捕捉比较寄存器 2 LPTIM0_CCR2(偏移：24h)

比特	名称	属性	默认值	功能描述
31:0	CCR2	R/W	0	捕捉/比较值寄存器 2 当 ARR=CCR2 时，以 CCR2 为准

20.3.11 LPTIMER0 计数值 load 寄存器 LPTIM0_LOAD1(偏移：28h)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	LOAD1	R/W	0	1：表示正在进行 LOAD CNT 的操作 0：表示 LOAD CNT 操作结束

20.3.12 LPTIMER0 计数缓存寄存器 LPTIM0_BUFFER1(偏移：2Ch)

比特	名称	属性	默认值	功能描述
31:0	BUFFER1	R/W	0	CNT 缓存寄存器 寄存器用于当软件发出 LOAD 指令后，存储计数器 CNT 的当前值

20.3.13 LPTIMER1 控制寄存器 3 LPTIM1_CR3(偏移：30h)

比特	名称	属性	默认值	功能描述
31:2	RSV	-	-	保留
1	CEN_ALL	R/W	0	CEN_ALL 控制使能： 1：当前 LPTIMER 计数开始受 CEN_ALL 控制 0：当前 LPTIMER 计数不受 CEN_ALL 控制
0	EN	R/W	0	LPTIMER1 使能位： 1：使能计数器计数 0：禁止计数器计数

20.3.14 LPTIMER1 控制寄存器 4 LPTIM1_CR4(偏移：34h)

比特	名称	属性	默认值	功能描述
31:15	RSV	-	-	保留

比特	名称	属性	默认值	功能描述
14:12	CAP1SRSEL	R/W	0	通道 1 捕捉信号源选择： 000：LPT_CH1 输入 001：RCLP 010：COMP0_IN 011：COMP1_IN 100：COMP2_IN 101：COMP3_IN 其它：OPA_IN
11:9	DIVCFG	R/W	0	计数时钟分频选择： 000：1 分频 001：2 分频 010：4 分频 011：8 分频 100：16 分频 101：32 分频 110：64 分频 111：128 分频
8:6	CLKSEL	R/W	0	时钟源选择： 000：LSCLK（由 SYSREG->SYSCTRL0[12] 选择的系统低速时钟 CLK32K，即 RCL 或 XTL）作为计数时钟 001：RCLP 作为计数时钟（RTC_1Hz） 010：PCLK 的门控时钟（PCLK）作为计数时钟 011：LPTIN 输入作为计数时钟 100：COMP0_IN 101：COMP1_IN 110：COMP2_IN 111：COMP3_IN
5	EDGESEL	R/W	0	ETR 计数时钟输入边沿选择： 1：ETR 的下降沿计数 0：ETR 的上升沿计数
4	SINGLE	R/W	0	单次计数模式使能： 1：单次计数模式：计数器被触发后计数到目标值后回到 0，并自动停止，产生溢出中断。 0：连续计数模式：计数器被触发后保持运行，直到被关闭为止。计数器达到目标值后回到 0 重新开始计数，并产生溢出中断。
3:2	TRIGEDGE	R/W	00	ETR 触发边沿选择： 00: ETR 输入信号上升沿触发 01: ETR 输入信号下降沿触发 10/11: ETR 输入信号上升下降沿触发
1:0	LPMOD	R/W	00	工作模式选择： 00：计数模式 01：ETR 脉冲触发计数模式 10：ETR 脉冲计数模式 11：Timeout 模式

20.3.15 LPTIMER1 计数值寄存器 2 LPTIM1_CNT2(偏移：38h)

比特	名称	属性	默认值	功能描述
31:0	CNT	R/W	0	计数器计数值

20.3.16 LPTIMER1 捕捉比较配置寄存器 3 LPTIM1_CCMR3(偏移：3Ch)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP1EDGE	R/W	0	通道 1 捕捉边沿选择： 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：未定义
3	RSV	-	-	保留
2	CC1P	R/W	0	通道 1 比较输出波形极性选择： 1：CNT<=CCR3 时置高，CNT>CCR3 时为低 0：CNT<=CCR3 时置低，CNT>CCR3 时为高
1	CC1S	R/W	0	通道 1 捕捉/比较选择： 1：通道 1 配置为输入 0：通道 1 配置为输出
0	CC1E	R/W	0	通道 1 捕捉使能： 1：通道 1 捕获功能使能 0：通道 1 捕获功能禁止

20.3.17 LPTIMER1 捕捉比较配置寄存器 4 LPTIM1_CCMR4(偏移：40h)

比特	名称	属性	默认值	功能描述
31:6	RSV	-	-	保留
5:4	CAP2EDGE	R/W	0	通道 2 捕捉边沿选择： 00：上升沿捕捉 01：下降沿捕捉 10：上升下降沿捕捉 11：未定义
3	RSV	-	-	保留
2	CC2P	R/W	0	通道 2 输出极性选择： 1：CNT<=CCR4 时置高，CNT>CCR4 时为低 0：CNT<=CCR4 时置低，CNT>CCR4 时为高
1	CC2S	R/W	0	通道 2 捕捉/比较选择： 1：通道 2 配置为输入 0：通道 2 配置为输出
0	CC2E	R/W	0	通道 2 捕捉使能： 1：通道 2 捕获功能使能 0：通道 2 捕获功能禁止

20.3.18 LPTIMER1 自动重装置寄存器 2 LPTIM1_ARR2(偏移：44h)

比特	名称	属性	默认值	功能描述
31:0	ARR2	R/W	0	自动重载目标寄存器： 当计数器计数值等于 ARR 时，计数器回到 0。 注：支持的最高的 PWM 输出为 12MHz，配置的 DIVCFG 和 ARR 需注意该条件。

20.3.19 LPTIMER1 捕捉比较寄存器 3 LPTIM1_CCR3(偏移：48h)

比特	名称	属性	默认值	功能描述
31:0	CCR3	R/W	0	捕捉/比较值寄存器 3 当 ARR=CCR3 时，以 CCR3 为准

20.3.20 LPTIMER1 捕捉比较寄存器 4 LPTIM1_CCR4(偏移：4Ch)

比特	名称	属性	默认值	功能描述
31:0	CCR4	R/W	0	捕捉/比较值寄存器 4 当 ARR=CCR4 时，以 CCR4 为准

20.3.21 LPTIM1 计数值 load 寄存器 2 LPTIM1_LOAD2(偏移：50h)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	LOAD2	R/W	0	1：表示正在进行 LOAD CNT 的操作 0：表示 LOAD CNT 操作结束

20.3.22 LPTIM1 计数缓存寄存器 2 LPTIM1_BUFFER2(偏移：54h)

比特	名称	属性	默认值	功能描述
31:0	BUFFER2	R/W	0	CNT 缓存寄存器 寄存器用于当软件发出 LOAD 指令后，存储计数器 CNT 的当前值

20.4 使用流程

下文将以 LPTIMER0 为例详细介绍软件使用流程。

20.4.1 普通定时器（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置 LPTIM0_CR2[1:0]，选择工作模式。
3. 配置 LPTIM0_CR2[4]，设置计数模式。
4. 配置 LPTIM0_CR2[11:9]，设置分频值。
5. 配置 LPTIM0_CR2[8:6]，设置时钟源。
6. 配置 LPTIM0_ARR1 目标寄存器值。
7. 使能 LPTIM01_IER 中断寄存器，选择溢出中断。
8. 使能 LPTIM0_CR1[0]，启动计数器。

20.4.2 结合 DMA 输入捕获功能（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置 LPTIM0_CR2[4]，设置计数模式。
3. 配置 LPTIM0_CR2[11:9]，设置分频值。
4. 配置 LPTIM0_CR2[8:6]，设置时钟源。
5. 配置 LPTIM0_CR2[3:2]，设置外部输入信号捕捉边沿。
6. DMA 模块时钟初始化。
7. 配置 DMA_CHSPERC、DMA_CHCTRLC、DMA_CHDPERC，设置源传输和目的传输数据格式，选择传输通道。
8. 配置 DMA_SRCADDRC 、DMA_CHCTRLC，设置源地址和目的地址，设置传输块大小。
9. 配置 DMA_EN=1，使能 DMA 传输。
10. 使能 LPTIM0_CR1[0]启动计数器。

20.4.3 PWM 输出（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置引脚为复用为 LPTIM0 pwm_out。
3. 配置 LPTIM0_CR2[4]，设置计数模式。
4. 配置 LPTIM0_CR2[9:11]，设置分频值。
5. 配置 LPTIM0_CR2[6:8]，设置时钟源。
6. 配置 LPTIM0_CCR1 捕捉比较寄存器值。
7. 配置 LPTIM0_ARR1 目标寄存器值。
8. 配置 LPTIM0_CCMR1[2]，选择 PWM 输出波形极性。

9. 配置 LPTIM0_CCMR1[1]，选择 PWM 输出模式。
10. 配置 LPTIM0_CCMR1[0]，使能 LPTIM0 捕获比较功能。
11. 使能 LPTIM01_IER 中断寄存器，打开中断。
12. 使能 LPTIM0_CR1[0]，启动计数器。

20.4.4 Trigger 脉冲触发计数模式（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置引脚为复用为 LPTIM0_EXT。
3. 配置 LPTIM0_CR2[4]，设置计数模式。
4. 配置 LPTIM0_CR2[9:11]，设置分频值。
5. 配置 LPTIM0_CR2[6:8]，设置时钟源。
6. 配置 LPTIM0_ARR1 目标寄存器值。
7. 配置 LPTIM0_CR2[2:3]，设置外部触发边沿。
8. 配置 LPTIM0_CR2[0:1]，选择 Trigger 脉冲触发计数模式。
9. 使能 LPTM01_IER[3]中断寄存器，打开外部触发中断。
10. 使能 LPTIM0_CR1[0]，启动计数器。

20.4.5 外部异步脉冲计数模式（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置引脚为复用为 LPTIM0_IN。
3. 配置 LPTIM0_CR2[4]，设置计数模式。
4. 配置 LPTIM0_CR2[9:11]，设置分频值。
5. 配置 LPTIM0_CR2[6:8]，设置时钟源。
6. 配置 LPTIM0_ARR1 目标寄存器值。
7. 配置 LPTM0_CR2[5]，设置 LPTIN 输入边沿。
8. 配置 LPTIM0_CR2[0:1]，选择 Trigger 脉冲触发计数模式。
9. 使能 LPTM01_IER 中断寄存器，打开中断。
10. 使能 LPTIM0_CR1[0]，启动计数器。

20.4.6 Timeout 模式（基于 LPTIMER0）

1. 初始化 LPTIM01 时钟模块。
2. 配置引脚为复用为 LPTIM0_EXT。
3. 配置 LPTIM0_CR2[4]，设置计数模式。

4. 配置 LPTIM0_CR2[9:11]，设置分频值。
5. 配置 LPTIM0_CR2[6:8]，设置时钟源。
6. 配置 LPTIM0_ARR1 目标寄存器值。
7. 配置 LPTM0_CR2[2:3]，设置外部触发边沿。
8. 配置 LPTM0_CR2[0:1]，选择 Timeout 模式。
9. 使能 LPTM01_IER 中断寄存器，打开溢出中断。
10. 使能 LPTIM0_CR1[0]，启动计数器。

    注：计数器溢出前没有出现新的 trigger，则产生溢出中断并停止计数，并清除使能，如果要重新使用，需要再次使能该中断。

21 RTC

21.1 概述

实时时钟（RTC）是一个独立的定时器/计数器，可提供基本的闹钟中断或者长时间的计数服务。闹钟中断通过可配置的实时时钟计数周期实现。

21.2 主要特性

- 内部或者外部 32.768KHz 时钟源
- 使用 BCD 时间实现可编程的完整万年历
- 周期唤醒中断功能
- 可编程的闹钟功能
- 可从 PAD 输出 XTLF 时钟信号供用户校准
- 数字调校，精度+/-0.119ppm
- RTC 计时器部分不复位
- 2 路输入上下沿时间戳功能

21.3 低功耗时基分频器（LTBC）

21.3.1 LTBC 功能

低功耗时基计数器(LTBC)模块用于产生系统所需的低速工作时钟，功能包括：

- 通过对 RCL 的预分频得到 64Hz 的 RTC 工作时钟
- 可通过调整计数周期实现 RTC 时钟的数字调校，每 128s 调校一次可实现最小步长为 0.23842ppm 调校后理论精度+/-0.1192ppm
- 16.384MHz 时钟虚拟调校可得到精确秒时标
- 可产生 1KHz、256Hz、64Hz、16Hz、4Hz、1Hz 周期中断，其中 1K 和 256Hz 是未经调校的，其他是经过数字调校的（如果使能了数字调校）
- 64Hz 预分频电路不受芯片复位影响

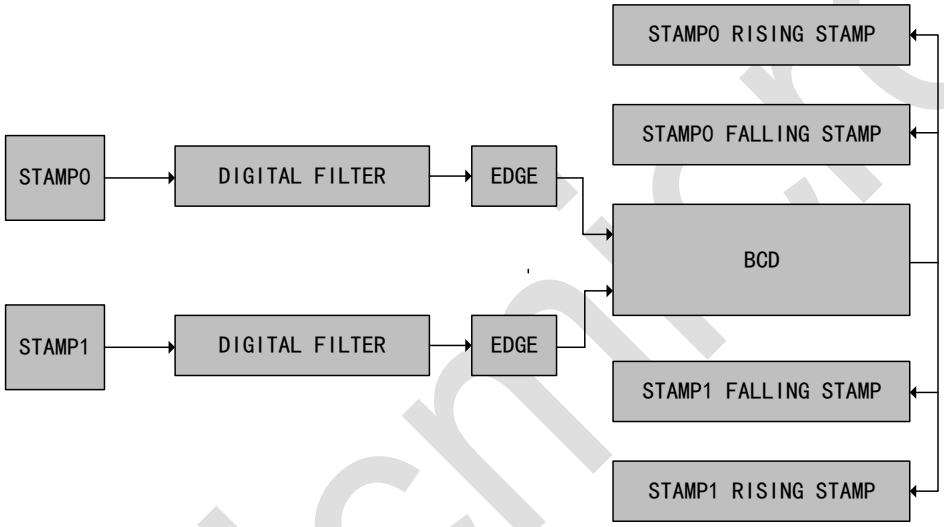
21.3.2 LTBC 数字调校

数字调校用来弥补外部时钟源带来的误差。使用 ADJUST，ADSIGN，PR1SEN 寄存器对 RTC 进行正向或者负向的数字调校，具体使用方法见寄存器的定义。

21.4 时间戳功能

为了支持开盖合盖检测，RTC 支持外部 IO 事件触发的时间戳功能。外部 IO 触发源为 STAMP0 和 STAMP1 的输入电平变化。使用此功能时，将相应的 IO 复用为 STAMP0 和 STAMP1 功能，打开 RTC_STORREN 寄存器使能相应通道，当 STAMP0 或 STAMP1 出现任何上升沿或下降沿时，RTC 会自动记录当前时间到 STAMP 寄存器组中，同时产生相应的标志，可用于产生中断或者供软件查询。

注意时间戳功能仅在 SLEEP 和 DEEPSLEEP 休眠模式下有效，ACTIVE 模式下时间戳功能不起作用，开盖合盖检测由软件中断来处理。



时间戳仅在相应标志寄存器为 0 的情况下记录事件发生时间，如果对应标志已经为 1，则忽略相应事件。因此如果有多次事件发生，时间戳仅记录第一次事件发生的时间，除非软件在事件发生后清除了标志寄存器。

21.5 寄存器描述

RTC 寄存器基地址：0x4000_1400

表 21-1：RTC 寄存器列表

偏置	名称	描述
0x00	RTC_WE	RTC 写使能寄存器
0x04	RTC_IE	RTC 中断使能寄存器
0x08	RTC_IF	RTC 中断标志寄存器
0x0C	RTC_BCDSEC	BCD 秒寄存器
0x10	RTC_BCDMIN	BCD 分寄存器
0x14	RTC_BCDHOUR	BCD 时寄存器
0x18	RTC_BCDDATE	BCD 天寄存器
0x1C	RTC_BCDWEEK	BCD 星期几寄存器
0x20	RTC_BCDMONTH	BCD 月寄存器

偏置	名称	描述
0x24	RTC_BCDYEAR	BCD 年寄存器
0x28	RTC_ALARM	闹铃设置寄存器
0x2C	RTC_FSEL	时钟信号输出控制寄存器
0x30	RTC_ADJUST	LTBC 数值调整寄存器
0x34	RTC_ADSIGN	LTBC 数值调整方向寄存器
0x38	RTC_PR1SEN	LTBC 虚拟调校使能寄存器
0x3C	RTC_SECCNT	毫秒计数寄存器
0x40	RTC_STAMPEN	RTC 时间戳使能寄存器
0x44	RTC_CLKSTAMP0R	RTC 上升沿时间戳 0 寄存器
0x48	RTC_CALSTAMP0R	RTC 上升沿日历戳 0 寄存器
0x4C	RTC_CLKSTAMP0F	RTC 下降沿时间戳 0 寄存器
0x50	RTC_CALSTAMP0F	RTC 下降沿日历戳 0 寄存器
0x54	RTC_CLKSTAMP1R	RTC 上升沿时间戳 1 寄存器
0x58	RTC_CALSTAMP1R	RTC 上升沿日历戳 1 寄存器
0x5C	RTC_CLKSTAMP1F	RTC 下降沿时间戳 1 寄存器
0x60	RTC_CALSTAMP1F	RTC 下降沿日历戳 1 寄存器

本小结对 RTC 寄存器进行了详细介绍。

21.5.1 写使能寄存器（RTC_WE）（偏移：00h）

比特	名称	属性	复位值	描述
31:0	RTCWE	R/W	0	RTC 写使能寄存器，当 CPU 向 RTC_WE 写入 0xACACACAC 时，允许 CPU 向 RTC 的 BCD 时间寄存器写入初值，这时 RTCWE 置 1；当 CPU 向 RTC_WE 写入不为 0xACACACAC 的任意值时恢复写保护,这时 RTCWE 清 0。

21.5.2 中断使能寄存器（RTC_IE）（偏移：04h）

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	STPR1IE	R/W	0	RTC STAMP1 上升沿事件中断使能： 1：使能中断 0：禁止中断
15	STPF1IE	R/W	0	RTC STAMP1 下降沿事件中断使能： 1：使能中断 0：禁止中断
14	STPR0IE	R/W	0	RTC STAMP0 上升沿事件中断使能： 1：使能中断 0：禁止中断
13	STPF0IE	R/W	0	RTC STAMP0 下降沿事件中断使能： 1：使能中断 0：禁止中断

比特	名称	属性	复位值	描述
12	ADJ128_IE	R/W	0	128 秒中断使能： 1：使能中断 0：禁止中断
11	ALARM_IE	R/W	0	闹钟中断使能： 1：使能中断 0：禁止中断
10	1KHZ_IE	R/W	0	1kHz 中断使能： 1：使能中断 0：禁止中断
9	256HZ_IE	R/W	0	256Hz 中断使能： 1：使能中断 0：禁止中断
8	64HZ_IE	R/W	0	64Hz 中断使能： 1：使能中断 0：禁止中断
7	16HZ_IE	R/W	0	16Hz 中断使能： 1：使能中断 0：禁止中断
6	8HZ_IE	R/W	0	8Hz 中断使能： 1：使能中断 0：禁止中断
5	4HZ_IE	R/W	0	4Hz 中断使能： 1：使能中断 0：禁止中断
4	2HZ_IE	R/W	0	2Hz 中断使能： 1：使能中断 0：禁止中断
3	SEC_IE	R/W	0	秒中断使能： 1：使能中断 0：禁止中断
2	MIN_IE	R/W	0	分中断使能： 1：使能中断 0：禁止中断
1	HOUR_IE	R/W	0	小时中断使能： 1：使能中断 0：禁止中断
0	DATE_IE	R/W	0	天中断使能： 1：使能中断 0：禁止中断

21.5.3 中断标志寄存器（RTC_IF）（偏移：08h）

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留

比特	名称	属性	复位值	描述
16	STPR1IF	R/W	0	RTC STAMP1 上升沿事件中断标志 1：中断置位 0：无中断产生 此寄存器为 1 的情况下时间戳 1 不再记录新的上升沿事件
15	STPF1IF	R/W	0	RTC STAMP1 下降沿事件中断标志： 1：中断置位 0：无中断产生 此寄存器为 1 的情况下时间戳 1 不再记录新的下降沿事件
14	STPR0IF	R/W	0	RTC STAMP0 上升沿事件中断标志： 1：中断置位 0：无中断产生 此寄存器为 1 的情况下时间戳 0 不再记录新的上升沿事件
13	STPF0IF	R/W	0	RTC STAMP0 下降沿事件中断标志： 1：中断置位 0：无中断产生 此寄存器为 1 的情况下时间戳 0 不再记录新的下降沿事件
12	ADJ128_IF	R/W	0	128 秒中断标志。写 1 清零： 1：中断置位 0：无中断产生
11	ALARM_IF	R/W	0	闹钟中断标志。写 1 清零： 1：中断置位 0：无中断产生
10	1KHZ_IF	R/W	0	1kHz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
9	256HZ_IF	R/W	0	256Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
8	64HZ_IF	R/W	0	64Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
7	16HZ_IF	R/W	0	16Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
6	8HZ_IF	R/W	0	8Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
5	4HZ_IF	R/W	0	4Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生
4	2HZ_IF	R/W	0	2Hz 中断标志。写 1 清零： 1：中断置位 0：无中断产生

比特	名称	属性	复位值	描述
3	SEC_IF	R/W	0	秒中断标志。写 1 清零： 1：中断置位 0：无中断产生
2	MIN_IF	R/W	0	分中断标志。写 1 清零： 1：中断置位 0：无中断产生
1	HOUR_IF	R/W	0	小时中断标志。写 1 清零： 1：中断置位 0：无中断产生
0	DATE_IF	R/W	0	天中断标志。写 1 清零： 1：中断置位 0：无中断产生

21.5.4 BCD 时间秒寄存器（RTC_BCDSEC）（偏移：0Ch）

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6:0	BCDSEC	R/W	不会被复位	秒时间数值，BCD 格式。

21.5.5 BCD 时间分钟寄存器（RTC_BCDMIN）(偏移：10h)

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6:0	BCDMIN	R/W	不会被复位	分钟时间数值，BCD 格式。

21.5.6 BCD 时间小时寄存器（RTC_BCDHOUR）(偏移：14h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:0	BCDHOUR	R/W	不会被复位	小时数值，BCD 格式。

21.5.7 BCD 时间天寄存器（RTC_BCDDATE）(偏移：18h)

比特	名称	属性	复位值	描述
31:6	RSV	-	-	保留
5:0	BCDDATE	R/W	不会被复位	天数值，BCD 格式。

21.5.8 BCD 时间星期寄存器（RTC_BCDWEEK）(偏移：1Ch)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	BCDWEEK	R/W	不会被复位	周数值，BCD 格式。

21.5.9 BCD 时间月寄存器（RTC_BCDMONTH）(偏移：20h)

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	BCDMONTH	R/W	不会被复位	月数值，BCD 格式。

21.5.10 BCD 时间年寄存器（RTC_BCDYEAR）(偏移：24h)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	BCDYEAR	R/W	不会被复位	年数值，BCD 格式。

21.5.11 闹钟寄存器（RTC_ALARM）(偏移：28h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	ALARMHOUR	R/W	0	闹钟的小时数值。
15	RSV	-	-	保留
14:8	ALARMMIN	R/W		闹钟的分数值。
7	RSV	-	-	保留
6:0	ALARMSEC	R/W		闹钟的秒数值。

21.5.12 时钟信号输出控制寄存器（RTC_FSEL）(偏移：2Ch)

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3:0	FSEL	R/W	0	频率输出选择信号： 4’b0000：输出 16.384M 时钟分频得到的精确 1 秒方波 4’b0001：输出 16.384M 时钟分频的高电平宽度 80ms 的秒时标 4’b0010：输出秒计数器进位信号，高电平宽度 1s 4’b0011：输出分计数器进位信号，高电平宽度 1s 4’b0100：输出小时计数器进位信号，高电平宽度 1s 4’b0101：输出天计数器进位信号，高电平宽度 1s 4’b0110：输出闹钟匹配信号 4’b0111：输出 128 秒方波信号 4’b1000：反向输出 16.384M 时钟分频的高电平宽度 80ms 的秒时标 4’b1001：反向输出秒计数器进位信号 4’b1010：反向输出分计数器进位信号 4’b1011：反向输出小时计数器进位信号 4’b1100：反向输出天计数器进位信号 4’b1101：反向输出闹钟匹配信号 4’b1110：反向输出 16.384M 时钟分频的精确 1s 方波信号 4’b1111：输出 RTC 内部秒时标方波

21.5.13 LTBC 数值调整寄存器（RTC_ADJUST）(偏移：30h)

比特	名称	属性	复位值	描述
31:11	RSV	-	-	保留
10:0	ADJUST	R/W	不会被复位	LTBC 补偿调整数值。 在进行数字调校时，ADJUST[10:7]为数字调校的公共值，在每秒的计数里调整 ADJUST[10:7] 个 32768Hz 时钟周期；ADJUST[6:0]为数字调校的私有值，在 128s 计数里的第 0 秒到第 ADJUST[6:0]-1 秒各调整 1 个 32768Hz 时钟周期。在进行虚拟调校时，RTC 在 128s 计数里调整 ADJUST[10:0]个 30.5us。

21.5.14 LTBC 数值调整方向寄存器（RTC_ADSIGN）(偏移：34h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	ADSIGN	R/W	不会被复位	LTBC 补偿方向： 1：表示减少计数初值 0：表示增加计数初值

21.5.15 LTBC 虚拟调校使能寄存器（RTC_PR1SEN）(偏移：38h)

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	PR1SEN	R/W	0	虚拟调校使能信号： 1：表示禁止虚拟调校功能，使用 32768Hz 时钟分频对 RTC 进行调校 0：表示使能虚拟调校功能，使用 16.384MHz 时钟分频对 RTC 进行调校

21.5.16 毫秒计数寄存器（RTC_SECCNT）(偏移：3Ch)

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	MSCNT	R	不会被复位	毫秒计数器值。以 256Hz 为周期计数，精度 3.9ms。

21.5.17 RTC 时间戳使能寄存器（RTC_STAMPEN）(偏移：40h)

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留

比特	名称	属性	复位值	描述
1	STAMP1EN	R/W	不会被复位	STAMP1 触发的时间戳功能使能位。无复位值，建议软件上电后进行初始化： 1：打开时间戳 0：关闭时间戳
0	STAMP0EN	R/W	不会被复位	STAMP0 触发的时间戳功能使能位。无复位值，建议软件上电后进行初始化： 1：打开时间戳 0：关闭时间戳

21.5.18 RTC 上升沿时间戳 0 寄存器（RTC_CLKSTAMP0R）(偏移：44h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	HRSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 秒寄存器的值。

21.5.19 RTC 上升沿日历戳 0 寄存器（RTC_CALSTAMP0R）(偏移：48h)

比特	名称	属性	复位值	描述
31:24	YRSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP0R	R/W	不会被复位	检测到 STAMP0 输入上升沿后存储 BCD 天寄存器的值。

21.5.20 RTC 下降沿时间戳 0 寄存器（RTC_CLKSTAMP0F）(偏移：4Ch)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	HRSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 分寄存器的值。
7	RSV	-	-	保留

比特	名称	属性	复位值	描述
6:0	SECSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 秒寄存器的值。

21.5.21 RTC 下降沿日历戳 0 寄存器（RTC_CALSTAMP0F）(偏移：50h)

比特	名称	属性	复位值	描述
31:24	YRSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP0F	R/W	不会被复位	检测到 STAMP0 输入下降沿后存储 BCD 天寄存器的值。

21.5.22 RTC 上升沿时间戳 1 寄存器（RTC_CLKSTAMP1R）(偏移：54h)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	HRSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 秒寄存器的值。

21.5.23 RTC 上升沿日历戳 1 寄存器（RTC_CALSTAMP1R）(偏移：58h)

比特	名称	属性	复位值	描述
31:24	YRSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP1R	R/W	不会被复位	检测到 STAMP1 输入上升沿后存储 BCD 天寄存器的值。

21.5.24 RTC 下降沿时间戳 1 寄存器（RTC_ CLKSTAMP1F）(偏移：5Ch)

比特	名称	属性	复位值	描述
31:22	RSV	-	-	保留
21:16	HRSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 小时寄存器的值。
15	RSV	-	-	保留
14:8	MINSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 分寄存器的值。
7	RSV	-	-	保留
6:0	SECSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 秒寄存器的值。

21.5.25 RTC 下降沿日历戳 1 寄存器（RTC_ CALSTAMP1F）(偏移：60h)

比特	名称	属性	复位值	描述
31:24	YRSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 年寄存器的值。
23:21	RSV	-	-	保留
20:16	MONSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 月寄存器的值。
15:11	RSV	-	-	保留
10:8	WKSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 周寄存器的值。
7:6	RSV	-	-	保留
5:0	DAYSTP1F	R/W	不会被复位	检测到 STAMP1 输入下降沿后存储 BCD 天寄存器的值。

21.6 使用流程

21.6.1 RTC 时间设置

由于 RTC 走时时钟较慢，为了提高抗 EMC 干扰能力，提供时间写保护功能，必须先对写保护寄存器写入 0xACACACAC，才能改写时间寄存器，软件可以通过写入除 0xACACACAC 外的任意值来禁止时间寄存器的写入，恢复写保护。

软件应在秒中断发生后再去置时，降低异步风险。同时支持 ms 级授时，即可以设置时间到 3.9ms 级别精度（1/256s）。此外，当软件写入秒时间时，硬件自动清零 64Hz->1Hz 的秒内计数器，以便实现秒对齐。

推荐的 RTC 时间设置流程如下：

1. 等待秒中断时间发生。
2. 连续写入年月日时分秒寄存器。
3. 如需 ms 级授时，再写入毫秒计数器。

4. 读出时间寄存器进行校验。

经过以上操作后最大授时误差在 4ms 以内。

21.6.2 RTC 时间读取

- 时间读取方式 1：
  1. 读 BCDTIME 值
  2. 再次读 BCDTIME 值

  如果 2 次读取内容一致，则为正确的当前时间；如果两次读取内容不一致，则重复前两个步骤。
- 时间读取方式 2：

  软件在 1s 中断发生后去读取时间，在 1s 内当前时间不会变化，因此能保证读到正确的当前时间值。

21.6.3 时间戳使用

1. 配置 Px_SEL 寄存器，触发使用的 GPIO 为 RTC_STAMP 模式。
2. 配置 PAD_IE 寄存器，触发使用的 GPIO 为输入使能模式。
3. 配置 GPIO_DIR 寄存器，触发使用的 GPIO 为输入模式。
4. 配置 RTC_IE[16:13]寄存器，选择 GPIO 边沿触发中断的模式。
5. 配置 RTC_STAMPEN[1:0]寄存器，使能时间戳功能。
6. 使能 RTC 相关的中断。
7. 当 GPIO 检测到匹配信号时，触发中断，并记录时间戳。

21.6.4 RTC 设置闹钟

1. 配置 SYSCTRL0 寄存器，32kHz RCL 时钟，RTC 模块内部自动分频成 1Hz 时钟。
2. 配置 RTC_WE 寄存器，使能写功能。
3. 设定 BCD 码的时分秒初始值。
4. 配置 RTC_WE 寄存器，关闭写功能。
5. 配置 RTC_ALARM[21:16]、RTC_ALARM[14:8]、RTC_ALARM[6:0]寄存器，设定 RTC 闹钟匹配值。
6. 写 RTC_IF[11]寄存器清除 RTC 中断。
7. 配置 RTC_I[11]E 寄存器，使能 RTC 中断。
8. RTC 中断触发后，配置 RTC_IF[11]寄存器清除中断。

22 DMA

22.1 概述

直接存储器访问(DMA)，支持 2 通道数据传输。

22.2 主要特性

- 支持单 MASTER 口。
- 可以控制 FLASH、SRAM、SPI0、SPI1、UART1、ADC、GPIOA、LPTIMER、ATIMER 模块之间的数据传输，其中 FLASH 仅可以作为源地址。
- 支持 Memory to Memory 模式、Memory to Peripheral 模式、Peripheral to Memory 模式、Peripheral to Peripheral 模式。
- 内部含有 2 个 DMA 通道。
- 数据传输的位宽可设、传输的 Block 长度可设。
- Block 最大长度可设为 32767 words。
- 支持源地址不变传输、递增传输。支持目的地址的不变传输、递增传输。
- 支持无限传输。
- 支持 burst 功能，其中传输数目可配 2、4、8、16。

22.3 寄存器描述

DMA 寄存器基地址：0x4002_0000

表 22-1：DMA 寄存器列表

偏置	名称	描述
0x00	DMA_SRCADDRC0	通道 0 源传送地址寄存器
0x04	DMA_DSTADDRC0	通道 0 目的传送地址寄存器
0x08	DMA_CHCTRLC0	通道 0 控制信息寄存器
0x0C	DMA_CHSTSC0	通道 0 传送状态寄存器
0x10	DMA_CHSPERC0	通道 0 源外设选择寄存器
0x14	DMA_CHDPERC0	通道 0 目标外设选择寄存器
0x20	DMA_SRCADDRC1	通道 1 源传送地址寄存器
0x24	DMA_DSTADDRC1	通道 1 目的传送地址寄存器
0x28	DMA_CHCTRLC1	通道 1 控制信息寄存器
0x2C	DMA_CHSTSC1	通道 1 传送状态寄存器
0x30	DMA_CHSPERC1	通道 1 源外设选择寄存器
0x34	DMA_CHDPERC1	通道 1 目标外设选择寄存器
0x40	DMAC_EN	DMA 控制器使能寄存器
0x44	DMA_SOFTRESET	DMA 软复位寄存器

偏置	名称	描述
0x48	DMA_INTSTATUS	DMA 中断指示寄存器
0x4C	DMA_INTMASK	DMA 中断屏蔽寄存器
0x54	DMA_PERREQ	DMA 外设请求寄存器

22.3.1 通道源传送地址寄存器 DMA_SRCADDRCx（偏移：20*x+00h）（x=0,1）

比特	名称	属性	复位值	描述
31:21	HI_SRC_ADDR	R/W	0	源高位地址，主要用于 decoder 选通
20:0	LOW_SRC_ADDR	R/W	0	源低位地址，主要用于具体外设的存储访问

注：在使能 burst 功能时，LOW_SRC_ADDR 地址必须对齐使用。对齐的位长根据 WIDTH 和 WAP_SRC_NUM 值有所不同。

	8 位数据位宽	16 位数据位宽	32 位数据位宽
Burst 2	LOW_SRC_ADDR[0]=0	LOW_SRC_ADDR[1:0]=0	LOW_SRC_ADDR[2:0]=0
Burst 4	LOW_SRC_ADDR[1:0]=0	LOW_SRC_ADDR[2:0]=0	LOW_SRC_ADDR[3:0]=0
Burst 8	LOW_SRC_ADDR[2:0]=0	LOW_SRC_ADDR[3:0]=0	LOW_SRC_ADDR[4:0]=0
Burst 16	LOW_SRC_ADDR[3:0]=0	LOW_SRC_ADDR[4:0]=0	LOW_SRC_ADDR[5:0]=0

22.3.2 通道目的传送地址寄存器 DMA_DSTADDRCx（偏移：20*x+04h）（x=0,1）

比特	名称	属性	复位值	描述
31:21	HI_DST_ADDR	R/W	0	目的高位地址，主要用于 decoder 选通
20:0	LOW_DST_ADDR	R/W	0	目的低位地址，主要用于具体外设的存储访问

注：在使能 burst 功能时，LOW_DST_ADDR 地址必须对齐使用。对齐的位长根据 WIDTH 和 WAP_DST_NUM 值有所不同。

	8 位数据位宽	16 位数据位宽	32 位数据位宽
Burst 2	LOW_DST_ADDR[0]=0	LOW_DST_ADDR[1:0]=0	LOW_DST_ADDR[2:0]=0
Burst 4	LOW_DST_ADDR[1:0]=0	LOW_DST_ADDR[2:0]=0	LOW_DST_ADDR[3:0]=0
Burst 8	LOW_DST_ADDR[2:0]=0	LOW_DST_ADDR[3:0]=0	LOW_DST_ADDR[4:0]=0
Burst 16	LOW_DST_ADDR[3:0]=0	LOW_DST_ADDR[4:0]=0	LOW_DST_ADDR[5:0]=0

22.3.3 通道控制信息寄存器 DMA_CHCTRLCx（偏移：20*x+08h）（x=0,1）

比特	名称	属性	复位值	描述
31:30	WIDTH	R/W	0	数据位宽， 0：8 位数据位宽 1：16 位数据位宽 2：32 位数据位宽 3：非法，但模块表现为 32 位的读写 源和目的数据位宽一样
29:15	XFER_SIZE	R/W	0	传输块大小，对 8 位模式支持 32767byte 的块，对 32 位模式支持 32767words 的块
14:12	FLOW_CTRL	R/W	0	流控模式： 数据值 源 目的 流控 0 Memory Memory DMAC 1 Memory Peripheral DMAC 2 Peripheral Memory DMAC 3 Peripheral Peripheral DMAC
11	TRANS_FREE	RW	0	无限传输使能位： 1：无限传输开启 0：无限传输结束 无限传输功能是传输数量不受 XFER_SIZE 寄存器控制，可以一直传输下去。
10	WAP_SRC_EN	RW	0	Src Burst 功能使能： 1：开启 Src Burst 功能 0：禁止 Src Burst 功能 Src Burst 功能是开启 Burst 传输。可以循环读取 2、4、8、16 个连续的地址空间的数据。读取方式可以递增、递减。
9:8	WAP_SRC_NUM	RW	0	Src Burst 传输数目配置： 0：Src Burst 2 1：Src Burst 4 2：Src Burst 8 3：Src Burst 16
7	WAP_DST_EN	RW	0	Dst Burst 功能使能： 1：开启 Dst Burst 功能 0：禁止 Dst Burst 功能 Dst Burst 功能是开启 Burst 传输。可以循环读取 2、4、8、16 个连续的地址空间的数据。读取方式可以递增、递减
6:5	WAP_DST_NUM	RW	0	Dst Burst 传输数目配置： 0：Dst Burst 2 1：Dst Burst 4 2：Dst Burst 8 3：Dst Burst 16
4:3	DST_INC	R/W	0	目的地址递增指示位，如果有效，则目的地址将随读取的数据递增，否则保持不变： 01：地址递增 10：地址递减

比特	名称	属性	复位值	描述
2:1	SRC_INC	R/W	0	源地址递增指示位，如果有效，则源地址将随读取的数据递增，否则保持不变： 01：地址递增 10：地址递减
0	CH_EN	R/W	0	通道使能标志，对于 DMAC 流控时，块传送结束后自动清 0

22.3.4 通道传送状态寄存器 DMA_CHSTSCx（偏移：20*x +0Ch）（x=0,1）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:1	LENGTH	R	0	在 DMA 传输时，表示此通道已经传输的数据长度
0	CH_BUSY	R	0	通道工作状态信息： 0： Idle 1： Busy

22.3.5 通道源外设选择寄存器 DMA_CHSPERCx（偏移：20*x +10h）（x=0,1）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
4:0	SPER	R/W	0	源外设，主要用于源外设的请求选取，具体外设分配为： 1：SPI0 接收 3：SPI1 接收 5：UART1 接收 6：ADC 接收 9：LPUART 接收 11：LPUART1 接收 12：LPTIMER1 RX2 13：LPTIMER1 RX1 14：LPTIMER0 RX2 15：LPTIMER0 RX1 16：ATIMER CTU 17：ATIMER CC4 18：ATIMER CC3 19：ATIMER CC2 20：ATIMER CC1 22：LIN 接收

22.3.6 通道目标外设选择寄存器 DMA_CHDPERCx（偏移：20*x +14h）（x=0,1）

比特	名称	属性	复位值	描述
31:5	RSV	-	-	保留
4:0	DPER	R/W	0	目的外设，主要用于目的外设的请求选取，具体外设分配为： 0：SPI0 发送 2：SPI1 发送 4：UART1 发送 8：LPUART0 发送 10：LPUART1 发送 16：ATIMER CTU（外部触发事件、软件触发事件和 COM 事件） 17：ATIMER CC4 18：ATIMER CC3 19：ATIMER CC2 20：ATIMER CC1 21：LIN 发送

22.3.7 DMA 控制器使能寄存器 DMAC_EN（偏移：40h）

比特	名称	属性	复位值	描述
31:2	RSV	-	-	保留
1	GRAND_SET	R/W	0	1：使能通道优先级的轮询功能，下一次传输通道优先级将进行轮询（若当前传输下通道 0 优先级高于通道 1，下次传输时通道 1 优先级高于通道 0） 0：通道之间为固定优先级（默认为通道 0 优先级高于通道 1）
0	DMAC_EN	R/W	0	1：使能 DMA 控制器 0：关闭 DMA 控制寄存器

22.3.8 DMA 软复位寄存器 DMA_SOFTRESET（偏移：44h）

比特	名称	属性	复位值	描述
31:0	DMA_SOFT_RESET	W	-	本寄存器为写操作虚拟寄存器，当 DMAC 模块采样到对此寄存器有写操作时，DMAC 将复位状态机以及需要复位的寄存器。没有实际的比特存在

22.3.9 DMA 中断指示寄存器 DMA_INTSTATUS（偏移：48h）

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	INT_TC_C1	R/W	0	通道 1 块传输结束中断指示，写 1 清 0

比特	名称	属性	复位值	描述
2	INT_TC_C0	R/W	0	通道 0 块传输结束中断指示，写 1 清 0
1	INT_ERR_C1	R/W	0	通道 1 总线出错中断指示，写 1 清 0
0	INT_ERR_C0	R/W	0	通道 0 总线出错中断指示，写 1 清 0

22.3.10 DMA 中断屏蔽寄存器 DMA_INTMASK（偏移：4Ch）

比特	名称	属性	复位值	描述
31:4	RSV	-	-	保留
3	MASK_TC_C1	R/W	0	通道 1 块传输结束中断屏蔽，如果为低，将不输出 IntTc 中断，即 IntTc=0
2	MASK_TC_C0	R/W	0	通道 0 块传输结束中断屏蔽，如果为低，将不输出 IntTc 中断，即 IntTc=0
1	MASK_ERR_C1	R/W	0	通道 1 总线出错中断屏蔽；如果为低，将不输出 IntErr 中断，即 IntErr=0
0	MASK_ERR_C0	R/W	0	通道 0 总线出错中断屏蔽；如果为低，将不输出 IntErr 中断，即 IntErr=0

22.3.11 DMA 外设请求寄存器 DMA_PERREQ（偏移：54h）

直接把 8 个外设请求连过来，没有实际的寄存器。

比特	名称	属性	复位值	描述
31:21	RSV	-	-	保留
20	ATIMER_CC1	R	0	ATIMER_CC1 通道请求
19	ATIMER_CC2	R	0	ATIMER_CC2 通道请求
18	ATIMER_CC3	R	0	ATIMER_CC3 通道请求
17	ATIMER_CC4	R	0	ATIMER_CC4 通道请求
16	ATIMER_CTU	R	0	ATIMER 外部触发请求/用户软件触发请求/COM 触发请求
15	LPTIM0_RX1	R	0	LPTIMER0 输入捕获通道 1 请求
14	LPTIM0_RX2	R	0	LPTIMER0 输入捕获通道 2 请求
13	LPTIM1_RX1	R	0	LPTIMER1 输入捕获通道 1 请求
12	LPTIM1_RX2	R	0	LPTIMER1 输入捕获通道 2 请求
11	LPUART1_RX_REQ	R	0	LPUART1 接收请求
10	LPUART1_TX_REQ	R	0	LPUART1 发送请求
9	LPUART_RX_REQ	R	0	LPUART 接收请求
8	LPUART_TX_REQ	R	0	LPUART 发送请求
7	RSV	-	-	保留
6	ADC_REQ	R	0	ADC 接收请求
5	UART1_RX_REQ	R	0	UART1 接收请求
4	UART1_TX_REQ	R	0	UART1 发送请求
3	SPI1_RX_REQ	R	0	SPI1 接收请求
2	SPI1_TX_REQ	R	0	SPI1 发送请求
1	SPI0_RX_REQ	R	0	SPI0 接收请求
0	SPI0_TX_REQ	R	0	SPI0 发送请求

22.4 使用流程

软件配置步骤：

1. 配置 DMA_CHCTRLCx[14:12]，选择 DMA 传输模式。
2. 配置 DMA_CHCTRLCx[4:0]和 DMA_CHDPERCx[4:0]，选择外设握手信号(传输地址为外设时才需要设置)。
3. 配置 DMA_CHCTRLCx[4:3]和 DMA_CHCTRLCx [2:1]，选择源地址和目的地址是否递增或不变。
4. 配置 DMA_CHCTRLCx[31:30]，选择传输数据的位宽。
5. 配置 DMAC_EN[0]为 1，使能 DMA 控制器。
6. 配置 DMA_SRCADDRCx，配置通道源地址。
7. 配置 DMA_DSTADDRCx，配置通道目的地址。
8. 配置 DMA_CHCTRLCx[29:15]，配置传输块数量。
9. 配置 DMA_CHCTRLCx[0]为 1，使能 DMA 通道传输。
10. 等待 DMA_CHCTRLCx[0]为 0，传输完成。若使能了传输结束中断，则等待传输结束中断后再处理。

23 CRC16

23.1 概述

CRC16 是一个以多项式 G(x) = $x^{16}+x^{12}+x^5+1$ 为计算式的硬件 16 位 CRC 循环冗余校验计算电路。可以根据用户预设的 CRC 初值，通讯数据计算出合适的 CRC 结果，并且支持设置输入数据与结果的正反向。

23.2 寄存器描述

CRC 寄存器基地址：0x4000_1800

表 23-1：CRC 寄存器列表

偏置	名称	描述
0x00	CRC16_DATA	写入需校验的数据与读出 CRC 结果
0x04	CRC16_INIT	写入 16 位 CRC 初值
0x08	CRC16_CTRL	CRC 控制寄存器

23.2.1 数据寄存器 CRC16_DATA（偏移：00H）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:8	RSLT2	R	0x0	读出 16 位 CRC 计算结果的高 8 位
7:0	DATA_RSLT1	R/W	0x0	写：写入需要进行 CRC 校验计算的数据，如需要校验的数据不止 8 位需按顺序逐次写入 读：读出 16 位 CRC 计算结果的低 8 位

低 8 位对于需校验数据来说为只写，写入后无法再次读出。

读操作返回 16 位 CRC 计算结果，其中低 8 位为与数据寄存器复用。

读操作将会对 CRC 计算清零，即读操作后将会重新载入初始值，次输入数据时会进行与读之前结果无关的新一轮计算。

23.2.2 初始值寄存器 CRC16_INIT（偏移：04H）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	INIT	R/W	0x0	写入 16 位 CRC 初始值

23.2.3 控制寄存器 CRC16_CTRL（偏移：08H）

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	RSLT_REV	R/W	0	CRC 计算结果是否进行高低位倒序： 1：倒序 0：不倒序
1	DATA_REV	R/W	0	CRC 计算数据是否进行高低位倒序： 1：倒序 0：不倒序
0	INITIAL_REV	R/W	0	CRC 初始值是否进行高低位倒序： 1：倒序 0：不倒序

23.3 使用流程

1. 设置 16 位初始值 CRC16_INIT[15:0]。
2. 设置 CRC16_CTRL[2:0]，选择数据是否倒序。
3. 向 CRC16_DATA 中写入 8 位 CRC 计算数据，如没有完成 CRC 数据输入顺次输入之后 8 位数据直至输入完成。
4. 读 CRC16_DATA，将一次返回 CRC 计算结果。

注意：读取结果后 CRC 计算模块将结束当前计算并重新载入初始值以备下次计算使用。

24 RNG

24.1 概述

RNG 是一款随机数生成器。可通过写入不同的随机数种子来生成不同的随机数序列。

24.2 主要特性

- 32 位随机数
- 可连续读取随机数序列

24.3 寄存器描述

寄存器基地址：0x4000_2000

表 24-1：RNG 寄存器列表

偏置	名称	描述
0x0E0	RNG_CR	随机数控制寄存器
0x0E4	RNG_SEED	随机数种子寄存器
0x0E8	RNG_DATA	随机数数据寄存器

24.3.1 随机数控制寄存器 RNG_CR（偏移：0E0h）

比特	名称	属性	复位值	描述
31:1	RSV	R	0	保留
0	RNG_EN	RW	0	1：随机数使能 0：随机数禁止

24.3.2 随机数种子寄存器 RNG_SEED（偏移：0E4h）

比特	名称	属性	复位值	描述
31:0	RNG_SEED	RW	0	随机数种子寄存器

24.3.3 随机数数据寄存器 RNG_DATA（偏移：0E8h）

比特	名称	属性	复位值	描述
31:0	RNG_DATA	R	32’hFF00FF	随机数寄存器。读取此寄存器，读出随机数值。

24.4 使用流程

1. 配置 RNG_CR[0]为 1，使能随机数。
2. 向 RNG_SEED[31:0]写入随机数种子值。
3. 读取 RNG_DATA[31:0]，读出随机数值，可连续读取。

25 WDT

25.1 概述

看门狗定时器在到达超时的值的时候可以产生不可屏蔽中断或者是复位。当系统由于软件错误或是由于外部设备故障而无法按照预期的方式响应的时候，使用看门狗定时器可以重新获得控制权。

25.2 主要特性

- 32 位递减并且可编程装载的寄存器
- 独立的看门狗时钟使能
- 带中断屏蔽的中断生成逻辑
- 软件跑飞保护锁定寄存器
- 复位使能/禁止产生逻辑
- 调试期间,微处理器的 CPU 暂停时，用户可使能的停滞

25.3 寄存器描述

WDT 寄存器基地址：0x4000_2400

表 25-1：WDT 寄存器列表

偏置	名称	描述
0x00	WDT_LOAD	装载寄存器
0x04	WDT_CNT	计数寄存器
0x08	WDT_CTRL	控制寄存器
0x0C	WDT_CLR	清除寄存器
0x10	WDT_INTRAW	RAW 中断状态寄存器
0x14	WDT_MINTS	MASK 中断状态寄存器
0x18	WDT_STALL	STALL 寄存器
0x1C	WDT_LOCK	LOCK 寄存器

25.3.1 装载寄存器 WDT_LOAD(偏移：00h)

比特	名称	属性	默认值	功能描述
31:0	LOAD	R/W	0xFFFFFFFF	WDOG初始装载值

25.3.2 计数寄存器 WDT_CNT(偏移：04h)

比特	名称	属性	默认值	功能描述
31:0	CNT	R	0xFFFFFFFF	WDOG内部CNT计数值

25.3.3 控制寄存器 WDT_CTRL(偏移：08h)

比特	名称	属性	默认值	功能描述
31	WRC	R	1	WDT加载值设置或写WDT_CTRL寄存器生效。向WDT_LOAD或者WDT_CTRL寄存器进行写操作时，设置位生效会有一定时间的延时。 0：设置为仍未生效 1：设置位生效
30:2	RSV	-	-	保留
1	RSTEN	R/W	0	WDT溢出复位使能： 0：不使能溢出复位功能 1：使能溢出复位功能
0	INTEN	R/W	0	WDT中断使能： 0：不使能中断 1：使能中断

25.3.4 清除寄存器 WDT_CLR(偏移：0Ch)

比特	名称	属性	默认值	功能描述
31:0	CLR_CARRY	W	0	向此寄存器写入任何值，将清除WDT溢出状态，从而清除掉中断和复位。

25.3.5 RAW 中断状态寄存器 WDT_INTRAW(偏移：10h)

比特	名称	属性	默认值	功能描述
31:0	INTRAW	R	0	原始中断寄存器，未经中断使能屏蔽： 0：WDT内部未发生溢出 1：WDT内部发生溢出

25.3.6 MASK 中断状态寄存器 WDT_MINTS(偏移：14h)

比特	名称	属性	默认值	功能描述
31:0	INTMS	R	0	0：WDT未产生中断 1：WDT产生中断

25.3.7 STALL 控制寄存器 WDT_STALL(偏移：18h)

比特	名称	属性	默认值	功能描述
31:16	CLKDIV	R/W	0	WDT计数时钟分频值： 0x0：不分频 0x1：2分频 0x2：3分频 …. 0xFFFE：0xFFFF分频 0xFFFF：保留
15:9	RSV	-	-	保留
8	STALL	R/W	0	WDT在芯片处于HALT状态时不计数功能的使能位： 0：不使能HALT状态计数器停止工作功能 1：使能HALT状态计数器停止工作功能
7:0	RSV	-	-	保留

25.3.8 LOCK 寄存器 WDT_LOCK(偏移：1Ch)

比特	名称	属性	默认值	功能描述
31:0	LOCK	W	0	WDT LOCK功能使能，当使能LOCK功能时，除此寄存器外的所有WDT寄存器均不可写。向此寄存器写任意值，使能WDT LOCK功能，向此寄存器写0x1ACCE551清除LOCK功能。

25.4 使用流程

- WDT 定时器配置:
  1. 向 WDT_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。
  2. 在 WDT_STALL 寄存器设置分频值。
  3. 将 WDT_CTRL[0]置 1，使能 INTEN 中断功能。
  4. 等待 WDT_CTRL[31]被置位，即设置生效。
  5. 在 WDT_LOAD 寄存器里装载所需要的加载值。
  6. 等待 WDT_CTRL[31]被置位，即设置生效。
  7. 向 WDT_LOCK 寄存器写入任意值锁定寄存器。
  8. 系统根据装载值及计数时钟分频值定时进入中断。
- WDT 喂狗流程配置
  1. 向 WDT_LOCK 寄存器写入 0x1ACCE551 解锁寄存器。
  2. 在 WDT_STALL 寄存器设置分频值。
  3. 将 WDT_CTRL[1]置 1，使能 RSTEN 复位功能；可选择将 WDT_CTRL[0]置 1，使能 INTEN

中断功能；等待 WDT_CTRL[31]被置位，即设置生效。

4. 在 WDT_LOAD 寄存器里装载所需要的加载值。
5. 等待 WDT_CTRL[31]被置位，即设置生效。
6. 向 WDT_LOCK 寄存器写入任意值锁定寄存器。
7. 系统可通过软件定时更新 WDT_LOAD 或者通过清中断标志位定时喂狗。若未能在定时时间内喂狗，则系统会在 2 倍定时时间后复位。

26 WWDT

26.1 概述

窗口看门狗是一个与 CPU 同步运行的看门狗，目的是实时监控 CPU 运行状态，在 CPU 运行异常的情况下复位 CPU，避免不可预计的后果。

26.2 主要特性

- 18 位递增并且可编程装载的寄存器
- 系统内部的故障探测器
- 时钟与系统时钟相同
- 用于监视软件错误
- 窗口前喂狗或超时不喂狗都会触发复位(喂狗有效窗口为 50%-100%时间内)
- 计数器达到溢出时间的 75%时触发预警中断

26.3 寄存器描述

WWDT 寄存器基地址：0x4000_3C00

表 26-1：WWDT 寄存器列表

偏置	名称	描述
0x00	WWDT_CON	控制寄存器
0x04	WWDT_CFG	配置寄存器
0x08	WWDT_CNT	计数寄存器
0x0C	WWDT_IE	中断使能寄存器
0x10	WWDT_IF	中断标志寄存器

26.3.1 控制寄存器 WWDT_CON(偏移：00h)

比特	名称	属性	默认值	功能描述
31:8	RSV	-	-	保留
7:0	CON	W	0	当CPU向此地址写入0x5A时启动WWDT定时器 在启动WWDT后，当CPU向此地址写入0xAC时清零计数器

26.3.2 配置寄存器 WWDT_CFG(偏移：04h)

比特	名称	属性	默认值	功能描述
31:4	RSV	-	-	保留
3:0	CFG	R/W	0	配置看门狗溢出时间： 0000：TPCLK * 4096 * 1 0001：TPCLK * 4096 * 4 0010：TPCLK * 4096 * 16 0011：TPCLK * 4096 * 64 0100：TPCLK * 4096 * 128 0101：TPCLK * 4096 * 256 0110：TPCLK * 4096 * 512 0111：TPCLK * 4096 * 1024 1000：TPCLK * 4096 * 2048 1001：TPCLK * 4096 * 4096 1010：TPCLK * 4096 * 8192 1011：TPCLK * 4096 * 16384 1100：TPCLK * 4096 * 32768 1101：TPCLK * 4096 * 65536 1110：TPCLK * 4096 * 131072 1111：TPCLK * 4096 * 262144

26.3.3 计数寄存器 WWDT_CNT(偏移：08h)

比特	名称	属性	默认值	功能描述
31:18	RSV	-	-	保留
17:0	CNT	R	0	WWDT计数寄存器值，软件可通过查询此寄存器了解WWDT计时进度

26.3.4 中断使能寄存器 WWDT_IE(偏移：0Ch)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	IE	R/W	0	WWDT中断使能： 0：中断使能禁止 1：中断使能打开

26.3.5 中断标志寄存器 WWDT_IF(偏移：10h)

比特	名称	属性	默认值	功能描述
31:1	RSV	-	-	保留
0	IF	R/W	0	WWDT 75%计时中断标志，写1清零： 0：无中断产生 1：中断标志置位

26.4 使用流程

- WWDT 定时器配置：
  1. 在 WWDT_CFG[3:0]设置溢出时间长度。
  2. 将 WWDT_IE[0]置 1，打开中断使能。
  3. 在 WWDT_CON 寄存器中写入 0x5A 启动 WWDT 定时器。
  4. 等待发生中断(计数到 75%时间产生中断)。
  5. 等待发生复位(溢出后产生复位)。
- WWDT 喂狗流程配置：

  在计数时间 50%~100%之内，在 WWDT_CON 寄存器中写入 0xAC 清零计数器。

27 ADC

27.1 概述

这是一个 12 位的 ADC 逐次接近型模数转换器。它具有多达 16 个输入通道，可测量来自 14 个外部源的信号、1 个内部 LDO 输出和 1 个内部 1/4 VDDH 输出。这些通道的 A/D 转换可在单次或连续扫描模式下进行。 ADC 控制器实现 CPU 和 SAR ADC 之间的通信。ADC 转换的结果存储在数据寄存器的低 12 位。

27.2 主要特性

- 支持 DMA 传输模式
- 16 位的可编程分频器，用于产生 A/D 时钟
- 支持 12 位分辨率 A/D 输入数据，最大采样率为 1MSps，采样率可通过软件配置
- 支持 16 通道 ADC 输入:14 个引脚通道、1 个内部 LDO 输入和 1 个内部 1/4 VDDH 输入
- 支持关闭模拟 ADC
- 支持轮询（poll）和中断（interrupt）传输模式
- 支持单次扫描或连续扫描模式
- 中断源：通道数据有效（16 个通道各有一个中断源）、FIFO 满（32 个 word）、FIFO 数据量达到设定值（1 或 16 个数据）
- 支持片内外设触发 ADC 转换
- ADC 电压输入范围:0~Vref
- ADC 参考电压源可选择：芯片供电电压 VDDH、IO 管脚外接电压 VREFIO

27.3 ADC 管脚分布

表 27-1：ADC 管脚分布

通道	引脚名称
ADC_CH0	PB7
ADC_CH1	PB6
ADC_CH2	PB5
ADC_CH3	PB4
ADC_CH4	PB3
ADC_CH5	PB2
ADC_CH6	PB1
ADC_CH7	PB0

通道	引脚名称
ADC_CH8	PA6
ADC_CH9	PA4
ADC_CH10	PA3
ADC_CH11	PC3
ADC_CH12	PC4
ADC_CH13	PC0
ADC_CH14	内部通道 VDDH 1/4
ADC_CH15	内部通道 LDO

27.4 寄存器描述

ADC 寄存器基地址：0x4000_1C00

表 27-2：ADC 寄存器列表

偏置	名称	描述
0x00	ADC_GCR	ADC 通用控制寄存器
0x04	ADC_DR0	A/D 通道 0 数据寄存器
0x08	ADC_DR1	A/D 通道 1 数据寄存器
0x0C	ADC_DR2	A/D 通道 2 数据寄存器
0x10	ADC_DR3	A/D 通道 3 数据寄存器
0x14	ADC_DR4	A/D 通道 4 数据寄存器
0x18	ADC_DR5	A/D 通道 5 数据寄存器
0x1C	ADC_DR6	A/D 通道 6 数据寄存器
0x20	ADC_DR7	A/D 通道 7 数据寄存器
0x24	ADC_DR8	A/D 通道 8 数据寄存器
0x28	ADC_DR9	A/D 通道 9 数据寄存器
0x2C	ADC_DR10	A/D 通道 10 数据寄存器
0x30	ADC_DR11	A/D 通道 11 数据寄存器
0x34	ADC_DR12	A/D 通道 12 数据寄存器
0x38	ADC_DR13	A/D 通道 13 数据寄存器
0x3C	ADC_DR14	A/D 通道 14 数据寄存器
0x40	ADC_DR15	A/D 通道 15 数据寄存器
0x44	ADC_CDR	A/D 时钟分频寄存器
0x48	ADC_ISR	A/D中断状态寄存器
0x4C	ADC_IER	A/D 中断使能寄存器
0x50	ADC_ICR	A/D 中断清除寄存器
0x54	ADC_COUNT	A/D 切换间隔计数寄存器
0x58	ADC_RXREG	A/D 接收数据寄存器
0x5C	ADC_CSTAT	ADC 当前状态寄存器
0x60	ADC_SPW	ADC 采样脉宽寄存器
0x64	ADC_TCRL	模拟 ADC 配置寄存器
0x68	ADC_HDT	ADC 硬件触发使能配置寄存器
0x6C	ADC_HDSET0	ADC 硬件设置寄存器 0
0x70	ADC_HDSET1	ADC 硬件设置寄存器 1

27.4.1 ADC 通用控制寄存器 ADC_GCR（偏移：000h）

比特	名称	属性	复位值	描述
31:16	CH_EN[15:0]	R/W	0x0	启用相关 ADC 通道进行模数转换。默认值：0。 1：通道启用 Bit[16]: ch_en[0] 通道 0 使能信号 Bit[17]: ch_en[1] 通道 1 使能信号 Bit[18]: ch_en[2] 通道 2 使能信号 Bit[19]: ch_en[3] 通道 3 使能信号. Bit[20]: ch_en[4] 通道 4 使能信号 Bit[21]: ch_en[5] 通道 5 使能信号 Bit[22]: ch_en[6] 通道 6 使能信号. Bit[23]: ch_en[7] 通道 7 使能信号 Bit[24]: ch_en[8] 通道 8 使能信号 Bit[25]: ch_en[9] 通道 9 使能信号 Bit[26]: ch_en[10] 通道 10 使能信号 Bit[27]: ch_en[11] 通道 11 使能信号 Bit[28]: ch_en[12] 通道 12 使能信号 Bit[29]: ch_en[13] 通道 13 使能信号 Bit[30]: ch_en[14] 通道 14 使能信号 Bit[31]: ch_en[15] 通道 15 使能信号 0：通道禁用 注意： 1）在单次扫描模式下，每个通道只进行 1 次模数转换。例如，需要选择通道 0、3、6 进行模数转换，则设置 ch_en[15:0] = 16'b0000 0000 0100 1001，通道 6 转换结束后操作完成。 2）在连续扫描模式下，按通道编号从低到高的顺序，重复循环地进行模数转换。例如，ch_en[15:0] = 16'b0000 0000 0100 1001，则选择了通道 0、3、6 一共 3 个通道。通道 0 将首先执行模数转换，接着通道 3 执行，然后通道 6 执行，再循环回通道 0。
15:12	RSV	-	-	保留
11	DATA_SAMP_NEG	R/W	0x0	ADC 数据在 EOC 信号的边沿采样选择： 1：ADC 数据在 EOC 的下降沿被采样 0：ADC 数据在 EOC 的上升沿被采样 注意：在本芯片设计中此位只能设置为 0。
10	ADC_START_EN	R/W	0x0	ADC 转换开始使能信号。当信号从低到高转换时，ADC 转换开始。当信号从高到低转换时，ADC 转换完成。此位在 ADC_EN=0 时清零。默认值为 0.
9	RSV	-	-	保留

比特	名称	属性	复位值	描述
8	ADC_PD_EN	R/W	1	SAR ADC 掉电使能信号： 1：SAR ADC 掉电 0：SAR ADC 上电
7	RSV	-	-	保留
6	ADC_CLK_SEL	R/W	0	A/D 时钟源选择信号： 1：ADC 时钟由系统时钟产生 0：ADC 时钟由内部时钟分频器产生
5	ADC_RCLR_EN	R/W	0	ADC 数据寄存器读取后清除使能： 1：使能 ADC 数据寄存器读取后清除 0：禁止 ADC 数据寄存器读取后清除
4	RXTLF	R/W	0	RX FIFO 中断触发条件位： 1：RX FIFO 里有 8 个或以上有效数据 0：RX FIFO 里有 1 个或以上有效数据
3	RXFIFO_EN	R/W	0	RX FIFO 使能位： 1：RX FIFO 使能 0：RX FIFO 禁用，刷新 RX FIFO 中的数据
2	DMAMODE	R/W	0	DMA 访问模式位： 1:DMA 访问模式(只有 DMA 能访问 RX FIFO) 0:CPU 访问模式(只有 CPU 能访问 RX FIFO)
1	CONTINUOUS	R/W	0	ADC 工作模式选择位。默认值为 0： 1：连续扫描模式 0：单次扫描模式
0	ADC_EN	R/W	0	ADC 控制器使能信号。默认值为 0： 1：使能 ADC 模块 0：禁用 ADC 模块

27.4.2 A/D 通道 0 数据寄存器 ADC_DR0（偏移：004h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID0	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH0_DATA	R	0x000	A/D 通道 0 接收数据寄存器 ： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.3 A/D 通道 1 数据寄存器 ADC_DR1（偏移：008h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留

比特	名称	属性	复位值	描述
15	DATA_VALID1	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH1_DATA	R	0x000	A/D 通道 1 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.4 A/D 通道 2 数据寄存器 ADC_DR2（偏移：00Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID2	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH2_DATA	R	0x000	A/D 通道 2 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.5 A/D 通道 3 数据寄存器 ADC_DR3（偏移：010h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID3	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH3_DATA	R	0x000	A/D 通道 3 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.6 A/D 通道 4 数据寄存器 ADC_DR4（偏移：014h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID4	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH4_DATA	R	0x000	A/D 通道 4 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.7 A/D 通道 5 数据寄存器 ADC_DR5（偏移：018h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID5	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH5_DATA	R	0x000	A/D 通道 5 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.8 A/D 通道 6 数据寄存器 ADC_DR6（偏移：01Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID6	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除。 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH6_DATA	R	0x000	A/D 通道 6 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.9 A/D 通道 7 数据寄存器 ADC_DR7（偏移：020h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID7	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除。 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH7_DATA	R	0x000	A/D 通道 7 接收数据寄存器 ： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.10 A/D 通道 8 数据寄存器 ADC_DR8（偏移：024h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID8	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除。 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH8_DATA	R	0x000	A/D 通道 8 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.11 A/D 通道 9 数据寄存器 ADC_DR9（偏移：028h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID9	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH9_DATA	R	0x000	A/D 通道 9 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.12 A/D 通道 10 数据寄存器 ADC_DR10（偏移：02Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID10	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH10_DATA	R	0x000	A/D 通道 10 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.13 A/D 通道 11 数据寄存器 ADC_DR11（偏移：030h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID11	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH11_DATA	R	0x000	A/D 通道 11 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.14 A/D 通道 12 数据寄存器 ADC_DR12（偏移：034h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID12	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH12_DATA	R	0x000	A/D 通道 12 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.15 A/D 通道 13 数据寄存器 ADC_DR13（偏移：038h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID13	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH13_DATA	R	0x000	A/D 通道 13 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.16 A/D 通道 14 数据寄存器 ADC_DR14（偏移：03Ch）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID14	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH14_DATA	R	0x000	A/D 通道 14 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除

27.4.17 A/D 通道 15 数据寄存器 ADC_DR15（偏移：040h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15	DATA_VALID15	R	0	数据有效位。此位在 ADC_EN=0 时清除，或者当 ADC_RCLR_EN=1 时的读取操作后清除： 1：数据有效 0：数据无效
14:12	RSV	-	-	保留
11:0	CH15_DATA	R	0x000	A/D 通道 15 接收数据寄存器： 当 ADC_RCLR_EN=1 时进行读取操作后，此位将被清除；

27.4.18 ADC 时钟分频寄存器 ADC_CDR（偏移：044h）

比特	名称	属性	复位值	描述
31:16	RSV	-	-	保留
15:0	CLKDIV	R/W	0x00FF	ADC 内部时钟分频寄存器： 分频公式： adc_clk = F_{pclk} / CLKDIV（F_{pclk} ：APB 总线时钟） 注：建议 CLKDIV >= 1。请勿把 clkdiv 设为 0 或 1，若把 clkdiv 设为 0 或 1，也当作 2 分频。如需使用 1 分频，建议使用外部时钟。

27.4.19 ADC 中断状态寄存器 ADC_ISR（偏移：048h）

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	RXFIFO_FULL_INTF	R	0x0	RX FIFO 满中断标志位： 1：接收 FIFO 满 0：接收 FIFO 未满
16	RX_INTF	R	0x0	接收器数据可用中断标志位： 当接收器 FIFO 接收到足够数据时，此标志位置 1（根据 RXTLF 位设置）： 1：接收器 FIFO 有可用数据 0：接收器 FIFO 无可用数据
15	CH15_INTF	R	0x0	通道 15 数据中断，高电平有效，软件给 ch15_int_clr 写 1 后清除： 1：中断激活 0：没有中断
14	CH14_INTF	R	0x0	通道 14 数据中断，高电平有效，软件给 ch14_int_clr 写 1 后清除：。 1：中断激活 0：没有中断
13	CH13_INTF	R	0x0	通道 13 数据中断，高电平有效，软件给 ch13_int_clr 写 1 后清除： 1：中断激活 0：没有中断
12	CH12_INTF	R	0x0	通道 12 数据中断，高电平有效，软件给 ch12_int_clr 写 1 后清除： 1: 中断激活 0: 没有中断
11	CH11_INTF	R	0x0	通道 11 数据中断，高电平有效，软件给 ch11_int_clr 写 1 后清除： 1：中断激活 0：没有中断

比特	名称	属性	复位值	描述
10	CH10_INTF	R	0x0	通道 10 数据中断，高电平有效，软件给 ch10_int_clr 写 1 后清除： 1：中断激活 0：没有中断
9	CH9_INTF	R	0x0	通道 9 数据中断，高电平有效，软件给 ch9_int_clr 写 1 后清除： 1：中断激活 0：没有中断
8	CH8_INTF	R	0x0	通道 8 数据中断，高电平有效，软件给 ch8_int_clr 写 1 后清除： 1：中断激活 0：没有中断
7	CH7_INTF	R	0x0	通道 7 数据中断，高电平有效，软件给 ch7_int_clr 写 1 后清除： 1：中断激活 0：没有中断
6	CH6_INTF	R	0x0	通道 6 数据中断，高电平有效，软件给 ch6_int_clr 写 1 后清除： 1：中断激活 0：没有中断
5	CH5_INTF	R	0x0	通道 5 数据中断，高电平有效，软件给 ch5_int_clr 写 1 后清除： 1：中断激活 0：没有中断
4	CH4_INTF	R	0x0	通道 4 数据中断，高电平有效，软件给 ch4_int_clr 写 1 后清除： 1：中断激活 0：没有中断
3	CH3_INTF	R	0x0	通道 3 数据中断，高电平有效，软件给 ch3_int_clr 写 1 后清除： 1：中断激活 0：没有中断
2	CH2_INTF	R	0x0	通道 2 数据中断，高电平有效，软件给 ch2_int_clr 写 1 后清除： 1：中断激活 0：没有中断
1	CH1_INTF	R	0x0	通道 1 数据中断，高电平有效，软件给 ch1_int_clr 写 1 后清除： 1：中断激活 0：没有中断
0	CH0_INTF	R	0x0	通道 0 数据中断，高电平有效，软件给 ch0_int_clr 写 1 后清除： 1：中断激活 0：没有中断

27.4.20 ADC 中断使能寄存器 ADC_IER（偏移：04Ch）

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	RXFIFO_FULL_IEN	R/W	0x0	RX FIFO 满中断使能（有 16 个数据时触发中断）： 1：中断使能 0：禁止中断
16	RXIEN	R/W	0x0	接收器 FIFO 中断使能（有 1 个或 8 个数据时触发中断）。 1：中断使能 0：禁止中断
15	CH15INT_EN	R/W	0x0	通道 15 数据中断使能，高电平有效： 1: 中断使能 0: 禁止中断
14	CH14_INT_EN	R/W	0x0	通道 14 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
13	CH13INT_EN	R/W	0x0	通道 13 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
12	CH12_INT_EN	R/W	0x0	通道 12 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
11	CH11_INT_EN	R/W	0x0	通道 11 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
10	CH10_INT_EN	R/W	0x0	通道 10 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
9	CH9_INT_EN	R/W	0x0	通道 9 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
8	CH8_INT_EN	R/W	0x0	通道 8 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
7	CH7_INT_EN	R/W	0x0	通道 7 数据中断使能，高电平有效： 1: 中断使能 0: 禁止中断
6	CH6_INT_EN	R/W	0x0	通道 6 数据中断使能，高电平有效： 1: 中断使能 0: 禁止中断
5	CH5_INT_EN	R/W	0x0	通道 5 数据中断使能，高电平有效： 1：中断使能 0：禁止中断

比特	名称	属性	复位值	描述
4	CH4_INT_EN	R/W	0x0	通道 4 数据中断使能，高电平有效： 1：中断使能 0：禁止中断
3	CH3_INT_EN	R/W	0x0	通道 3 数据中断使能，高电平有效 1：中断使能 0：禁止中断
2	CH2_INT_EN	R/W	0x0	通道 2 数据中断使能，高电平有效 1：中断使能 0：禁止中断
1	CH1_INT_EN	R/W	0x0	通道 1 数据中断使能，高电平有效. 1：中断使能 0：禁止中断
0	CH0_INT_EN	R/W	0x0	通道 0 数据中断使能，高电平有效 1：中断使能 0：禁止中断

27.4.21 ADC 中断清除寄存器 ADC_ICR（偏移：050h）

比特	名称	属性	复位值	描述
31:18	RSV	-	-	保留
17	RXFIFO_FULL_ICLR	W	0x0	RX FIFO 满中断清除： 1：清除中断 0：不清除中断
16	RXICLR	W	0x0	RX FIFO 中断清除： 1：清除中断 0：不清除中断
15	CH15_INT_CLR	W	0x0	通道 15 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
14	CH14_INT_CLR	W	0x0	通道 14 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
13	CH13_INT_CLR	W	0x0	通道 13 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
12	CH12_INT_CLR	W	0x0	通道 12 数据中断清除，高电平有效：. 1：清除中断 0：不清除中断
11	CH11_INT_CLR	W	0x0	通道 11 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
10	CH10_INT_CLR	W	0x0	通道 10 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
9	CH9_INT_CLR	W	0x0	通道 9 数据中断清除，高电平有效： 1：清除中断 0：不清除中断

比特	名称	属性	复位值	描述
8	CH8_INT_CLR	W	0x0	通道 8 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
7	CH7_INT_CLR	W	0x0	通道 7 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
6	CH6_INT_CLR	W	0x0	通道 6 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
5	CH5_INT_CLR	W	0x0	通道 5 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
4	CH4_INT_CLR	W	0x0	通道 4 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
3	CH3_INT_CLR	W	0x0	通道 3 数据中断清除，高电平有效： 1：清除中断 0：不清除中断；
2	CH2_INT_CLR	W	0x0	通道 2 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
1	CH1_INT_CLR	W	0x0	通道 1 数据中断清除，高电平有效： 1：清除中断 0：不清除中断
0	CH0_INT_CLR	W	0x0	通道 0 数据中断清除，高电平有效： 1：清除中断 0：不清除中断

27.4.22 ADC 切换间隔计数寄存器 ADC_COUNT（偏移：054h）

比特	名称	属性	复位值	描述
31:8	RSV	-	-	保留
7:0	ADC_COUNT	R/W	0x01	通道切换间隔时间，这个值的单位为 ADC 时钟周期。 实际通道切换时间=(adc_count+16) * ADC 时钟周期 注：此寄存器只能在 ADC 控制器使能前配置，否则无效

27.4.23 ADC 接收数据寄存器 ADC_RXREG（偏移：058h）

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:0	RXFIFO_OUT	R	0x0	接收器 FIFO 的输出，SAR ADC 的值。此寄存器只读。

27.4.24 ADC 当前状态寄存器 ADC_CSTAT（偏移：05Ch）

比特	名称	属性	复位值	描述
31:1	RSV	-	-	保留
0	RXAVL	R	0x0	CPU 轮询模式下的接收数据可用标志位。 当接收器接收到有效数据时此位置 1。 1：接收器 FIFO 有有效数据 0：接收器 FIFO 为空

27.4.25 ADC 采样脉宽寄存器 ADC_SPW（偏移：060h）

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2:0	SAMPCLK_WIDTH	R/W	0x3	采样脉宽配置。注意：在本芯片设计中，此寄存器应该设置大于或等于 3 的值。 3：SAMPCLK 为 4 个 ADC_CLK 脉冲信号 4：SAMPCLK 为 5 个 ADC_CLK 脉冲信号 5：SAMPCLK 为 6 个 ADC_CLK 脉冲信号 此寄存器的合法值范围为 3~5，超过此范围可能会引起 ADC 工作不正常，若测试精度不准确可适当增加此值。此寄存器值需与 ADC_COUNT 寄存器值同时更改。 注：请谨慎设置此值，增加此数值是以牺牲转换效率为代价的。

27.4.26 模拟 ADC 配置寄存器 ADC_TCRL（偏移：064h）

比特	名称	属性	复位值	描述
31:7	RSV	-	-	保留
6	PS	R/W	0x0	模拟ADC配置值
5	ENMPS	R/W	0x0	1：打开 ADC 内部 1/4 分压电阻 0：关闭 ADC 内部 1/4 分压电阻
4:3	SPEED	R/W	0x0	ADC 转换速度。建议设为默认值 0
2:1	VREF_SEL	R/W	0x0	选择 ADC 参考电压源： 00/01：选择 VDDH 作为 ADC 的参考电压源 10：选择 VREFIO 作为 ADC 的参考电压源 11：保留
0	USE_OPA	R/W	0x0	选择 ADC 输入通道是否经 OPA 缓冲： 1：开启 OPA 缓冲使能 0：关闭 OPA 缓冲使能

27.4.27 ADC 硬件触发使能配置寄存器 ADC_HDT（偏移：068h）

比特	名称	属性	复位值	描述
31	RTC_FOUT_HT	R/W	0x0	RTC FOUT 硬件触发使能： 1：FOUT 硬件触发使能 0：FOUT硬件触发关闭
30	PC2_HT	R/W	0x0	PC2 硬件触发使能： 1：PC2 硬件触发使能 0：PC2硬件触发关闭
29	PC1_HT	R/W	0x0	PC1 硬件触发使能： 1：PC1 硬件触发使能 0：PC1硬件触发关闭
28	PB3_HT	R/W	0x0	PB3 硬件触发使能： 1：PB3 硬件触发使能 0：PB3 硬件触发关闭
27	BTIM3_OUT_HT	R/W	0x0	BTIMER3 OUT 硬件触发使能： 1：BTIMER3 OUT 硬件触发使能 0：BTIMER3 OUT 硬件触发关闭
26	BTIM2_OUT_HT	R/W	0x0	BTIMER2 OUT 硬件触发使能： 1：BTIMER2 OUT 硬件触发使能 0：BTIMER2 OUT 硬件触发关闭
25	BTIM1_OUT_HT	R/W	0x0	BTIMER1 OUT 硬件触发使能： 1：BTIMER1 OUT 硬件触发使能 0：BTIMER1 OUT 硬件触发关闭
24	BTIM0_OUT_HT	R/W	0x0	BTIMER0 OUT 硬件触发使能： 1：BTIMER0 OUT 硬件触发使能 0：BTIMER0 OUT 硬件触发关闭
23	ATIMER_TRGO_HT	R/W	0x0	ATIMER TRGO 硬件触发使能： 1：ATIMER TRGO 硬件触发使能 0：ATIMER TRGO 硬件触发关闭
22	ATIMER_CH4_HT	R/W	0x0	ATIMER CH4 硬件触发使能： 1：ATIMER CH4 硬件触发使能 0：ATIMER CH4 硬件触发关闭
21	ATIMER_CH3_HT	R/W	0x0	ATIMER CH3 硬件触发使能： 1：ATIMER CH3 硬件触发使能 0：ATIMER CH3 硬件触发关闭
20	ATIMER_CH2_HT	R/W	0x0	ATIMER CH2 硬件触发使能： 1：ATIMER CH2 硬件触发使能 0：ATIMER CH2 硬件触发关闭
19	ATIMER_CH1_HT	R/W	0x0	ATIMER CH1 硬件触发使能： 1：ATIMER CH1 硬件触发使能 0：ATIMER CH1 硬件触发关闭
18	LVD_HT	R/W	0x0	LVD 硬件触发使能： 1：LVD 硬件触发使能 0：LVD 硬件触发关闭
17	OPA_HT	R/W	0x0	OPA 硬件触发使能： 1：OPA 硬件触发使能 0：OPA 硬件触发关闭

比特	名称	属性	复位值	描述
16	COMP2_HT	R/W	0x0	COMP2 硬件触发使能： 1：COMP2 硬件触发使能 0：COMP2 硬件触发关闭
15	COMP1_HT	R/W	0x0	COMP1 硬件触发使能： 1：COMP1 硬件触发使能 0：COMP1 硬件触发关闭
14	COMP0_HT	R/W	0x0	COMP0 硬件触发使能： 1：COMP0 硬件触发使能 0：COMP0 硬件触发关闭
13	COMP3_HT	R/W	0x0	COMP3 硬件触发使能： 1：COMP3 硬件触发使能 0：COMP3 硬件触发关闭
12	PB1_HT	R/W	0x0	PB1 硬件触发使能： 1：PB1 硬件触发使能 0：PB1硬件触发关闭
11	PA4_HT	R/W	0x0	PA4 硬件触发使能： 1：PA4 硬件触发使能 0：PA4硬件触发关闭
10	PA3_HT	R/W	0x0	PA3 硬件触发使能： 1：PA3 硬件触发使能 0：PA3硬件触发关闭
9	LPTIM1_HT1	RW	0x0	LPTIMER1 OUT1 硬件触发使能： 1：LPTIMER1 OUT1 硬件触发使能 0：LPTIMER1 OUT1 硬件触发关闭
8	LPTIM1_HT0	RW	0x0	LPTIMER1 OUT0 硬件触发使能： 1：LPTIMER1 OUT0 硬件触发使能 0：LPTIMER1 OUT0 硬件触发关闭
7	LPTIM0_HT1	RW	0x0	LPTIMER0 OUT1 硬件触发使能： 1：LPTIMER0 OUT1 硬件触发使能 0：LPTIMER0 OUT1 硬件触发关闭
6	LPTIM0_HT0	RW	0x0	LPTIMER0 OUT0 硬件触发使能： 1：LPTIMER0 OUT0 硬件触发使能 0：LPTIMER0 OUT0 硬件触发关闭
5	GTIMER2_TRGO_HT	RW	0x0	GTIMER2 TRGO 硬件触发使能： 1：GTIMER2 TRGO 硬件触发使能 0：GTIMER2 TRGO 硬件触发关闭
4	GTIMER2_HT	RW	0x0	GTIMER2 硬件触发使能： 1：GTIMER2 硬件触发使能 0：GTIMER2 硬件触发关闭
3	GTIMER1_TRGO_HT	RW	0x0	GTIMER1 TRGO 硬件触发使能： 1：GTIMER1 TRGO 硬件触发使能 0：GTIMER1 TRGO 硬件触发关闭
2	GTIMER1_HT	RW	0x0	GTIMER1 硬件触发使能： 1：GTIMER1 硬件触发使能 0：GTIMER1 硬件触发关闭
1	GTIMER0_TRGO_HT	RW	0x0	GTIMER0 TRGO 硬件触发使能： 1：GTIMER0 TRGO 硬件触发使能 0：GTIMER0 TRGO 硬件触发关闭

比特	名称	属性	复位值	描述
0	GTIMER0_HT	RW	0x0	GTIMER0 硬件触发使能： 1：GTIMER0 硬件触发使能 0：GTIMER0 硬件触发关闭

27.4.28 ADC 硬件设置寄存器 0 ADC_HDSET0（偏移：06Ch）

比特	名称	属性	复位值	描述
30:31	COMP1_SET	R/W	0x0	COMP1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
29:28	COMP0_SET	R/W	0x0	COMP0 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
27:26	COMP3_SET	R/W	0x0	COMP3 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发；
25:24	PB1_SET	RW	0x0	PB1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
23:22	PA4_SET	RW	0x0	PA4 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
21:20	PA3_SET0	R/W	0x0	PA3 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
19:18	LPTIM1_SET1	RW	0x0	LPTIMER1 OUT1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
17:16	LPTIM1_SET0	RW	0x0	LPTIMER1 OUT0 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
15:14	LPTIM0_SET1	RW	0x0	LPTIMER0 OUT1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
13:12	LPTIM0_SET0	RW	0x0	LPTIMER0 OUT0 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发

比特	名称	属性	复位值	描述
11:10	GTIMER2_TRGO_SET	RW	0x0	GTIMER2 TRGO 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
9:8	GTIMER2_SET	RW	0x0	GTIMER2 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
7:6	GTIMER1_TRGO_SET	RW	0x0	GTIMER1 TRGO 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
5:4	GTIMER1_SET	RW	0x0	GTIMER1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
3:2	GTIMER0_TRGO_SET	RW	0x0	GTIMER0 TRGO 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
1:0	GTIMER0_SET	RW	0x0	GTIMER0 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发

27.4.29 ADC 硬件设置寄存器 1 ADC_HDSET1（偏移：070h）

比特	名称	属性	复位值	描述
30:31	RTC_FOUT_SET	R/W	0x0	RTC FOUT 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
29:28	PC2_SET	R/W	0x0	PC2 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
27:26	PC1_SET	RW	0x0	PC1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
25:24	PB3_SET	RW	0x0	PB3 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
23:22	BTIM3_SET	RW	0x0	BTIMER3 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发

比特	名称	属性	复位值	描述
21:20	BTIM2_SET	R/W	0x0	BTIMER2 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
19:18	BTIM1_SET	RW	0x0	BTIMER1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
17:16	BTIM0_SET	RW	0x0	BTIMER0 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
15:14	ATIMER_TRGO_HT	RW	0x0	ATIMER TRGO 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
13:12	ATIMER_CH4_SET	RW	0x0	ATIMER CH4 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
11:10	ATIMER_CH3_SET	RW	0x0	ATIMER CH3 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
9:8	ATIMER_CH2_SET	RW	0x0	ATIMER CH2 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
7:6	ATIMER_CH1_SET	RW	0x0	ATIMER CH1 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
5:4	LVD_SET	RW	0x0	LVD 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
3:2	OPA_SET	RW	0x0	OPA 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发
1:0	COMP2_SET	RW	0x0	COMP2 硬件触发极性选择： 00：上升沿触发 01：下降沿触发 10/11：双边沿触发

27.5 ADC 使用流程

27.5.1 单次扫描模式单通道 A/D 转换

单次扫描模式下，ADC 启动转换后只执行一次转换。

1. 配置 ADC_GCR[8]为 0，模拟 ADC 上电。
2. 配置 ADC_GCR[6]为 0，选择 ADC 时钟源为内部时钟分频器产生时钟。
3. 配置 ADC_CDR[15:0]，设置 ADC 时钟分频。
4. 配置 ADC_SPW[2:0]和 ADC_COUNT[7:0]，设置 ADC 转换速度。
5. 配置 ADC_TCRL[2:1]，设置 ADC 参考电压源。
6. 配置 ADC_GCR[1]为 0，选择单次扫描模式。
7. 配置 ADC_GCR[0]为 1，启用 ADC 模块。
8. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口(PAD_ADS)。
9. 配置 ADC_GCR[31:16]，启用待转换的通道。
10. 配置 ADC_GCR[10]为 1,启动 ADC 转换。
11. 等待 ADC_ISR 的相应通道标志位为 1，读取 ADC_DR 中的数据。
12. 如需进行多次单次转换，则重复执行步骤 10 和 11(注：启动 ADC 转换前需确保 ADC_GCR[10]为 0)。
13. 如使能了 ADC_IER.CH_INT_EN 和 ADC_IER.RXIEN 中断，则等待中断触发后读取 ADC_DR。

27.5.2 单次扫描模式多通道 A/D 转换

单次扫描模式下，ADC 启动转换后只执行一次转换。当同时启用了多个通道时，开启 ADC 转换后，会依次按通道优先级 0-13 进行转换，当优先级最低的通道转换完成，就代表此次单次扫描转换完成。

1. 配置 ADC_GCR[8]为 0，模拟 ADC 上电。
2. 配置 ADC_GCR[6]为 0，选择 ADC 时钟源为内部时钟分频器产生时钟。
3. 配置 ADC_CDR[15:0]，设置 ADC 时钟分频。
4. 配置 ADC_SPW[2:0]和 ADC_COUNT[7:0]，设置 ADC 转换速度。
5. 配置 ADC_TCRL[2:1]，设置 ADC 参考电压源。
6. 配置 ADC_GCR[3]为 1，使能 RX FIFO。
7. 配置 ADC_GCR[1]为 0，选择单次扫描模式。
8. 配置 ADC_GCR[0]为 1，启用 ADC 模块。
9. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口(PAD_ADS)。
10. 配置 ADC_GCR[31:16]，启用待转换的通道。

11. 配置 ADC_GCR[10]为 1,启动 ADC 转换。
12. 等待转换优先级最低的通道 ADC_ISR 的相应通道标志位为 1 后，再读取每个通道的 ADC_DR 中的数据。
13. 如需进行多次单次转换，则重复执行以上 2 个步骤(注：启动 ADC 转换前需确保 ADC_GCR.ADC_START_EN 为 0)。
14. 如使能了 ADC_IER.CH_INT_EN，则等待中断触发后读取 ADC_DR 中的数据。

注：

- 在使用多通道转换时，可以通过判断最低优先级通道数据有效后，再去读取每个通道的 ADC_DR 中的数据。
- 在使用多通道转换时，若采样精度不足，建议适当修改 ADC_SPW 和 ADC_COUNT 的值。

27.5.3 连续扫描模式单通道 A/D 转换

连续扫描模式下，启动一次 ADC 转换后会对所选通道进行不断地连续转换，将 ADC_GCR.ADC_START_EN 写 0 可停止转换。

1. 配置 ADC_GCR[8]为 0，模拟 ADC 上电。
2. 配置 ADC_GCR[6]为 0，选择 ADC 时钟源为内部时钟分频器产生时钟。
3. 配置 ADC_CDR[15:0]，设置 ADC 时钟分频。
4. 配置 ADC_SPW[2:0]和 ADC_COUNT[7:0]，设置 ADC 转换速度。
5. 配置 ADC_TCRL[2:1]，设置 ADC 参考电压源。
6. 配置 ADC_GCR[1]为 1，选择连续扫描模式。
7. 配置 ADC_GCR[0]为 1，启用 ADC 模块。
8. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口(PAD_ADS)。
9. 配置 ADC_GCR[31:16]，启用待转换的通道。
10. 配置 ADC_GCR[10]为 1,启动 ADC 转换。
11. 等待 ADC_ISR 的相应通道标志位为 1，读取 ADC_DR 中的数据。
12. 如需读取多个 ADC 数据，则重复以上 1 个步骤。
13. 如使能了 ADC_IER.CH_INT_EN，则等待中断触发后读取 ADC_DR 中的数据。

27.5.4 连续扫描模式多通道 A/D 转换

连续扫描模式下，启动一次 ADC 转换后会对所选通道进行不断地连续转换，将 ADC_GCR.ADC_START_EN 写 0 可停止转换。当同时启用了多个通道时，开启 ADC 转换后，会依次按通道优先级 0-13 进行转换，当优先级最低的通道转换完成，就代表此次连续扫描转换完成。

1. 配置 ADC_GCR[8]为 0，模拟 ADC 上电。

2. 配置 ADC_GCR[6]为 0，选择 ADC 时钟源为内部时钟分频器产生时钟。
3. 配置 ADC_CDR[15:0]，设置 ADC 时钟分频。
4. 配置 ADC_SPW[2:0]和 ADC_COUNT[7:0]，设置 ADC 转换速度。
5. 配置 ADC_TCRL[2:1]，设置 ADC 参考电压源。
6. 配置 ADC_GCR[1]为 1，选择连续扫描模式。
7. 配置 ADC_GCR[0]为 1，启用 ADC 模块。
8. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口(PAD_ADS)。
9. 配置 ADC_GCR[31:16]，启用待转换的通道。
10. 配置 ADC_GCR[10]为 1,启动 ADC 转换。
11. 等待 ADC_ISR 的相应通道标志位为 1，读取 ADC_DR 中的数据。
12. 如需读取多个 ADC 数据，则重复以上 1 个步骤。
13. 如使能了 ADC_IER.CH_INT_EN 和 ADC_IER.RXIEN 中断，则等待中断触发后读取 ADC_DR。

注：

- 在使用多通道转换时，可以通过判断最低优先级通道数据有效后，再去读取每个通道的 ADC_DR 中的数据。
- 在使用多通道转换时，若采样精度不足，建议适当修改 ADC_SPW 和 ADC_COUNT 的值。

27.5.5 硬件触发事件 A/D 转换

硬件触发事件只能触发单次扫描模式，ADC 启动转换后只执行一次转换。

1. 配置 ADC_GCR[8]为 0，模拟 ADC 上电。
2. 配置 ADC_GCR[6]为 0，选择 ADC 时钟源为内部时钟分频器产生时钟。
3. 配置 ADC_CDR[15:0]，设置 ADC 时钟分频。
4. 配置 ADC_SPW[2:0]和 ADC_COUNT[7:0]，设置 ADC 转换速度。
5. 配置 ADC_TCRL[2:1]，设置 ADC 参考电压源。
14. 配置 ADC_GCR[1]为 0，选择单次扫描模式。
15. 配置 ADC_GCR[0]为 1，启用 ADC 模块。
6. 根据 ADC 输入通道对应的 GPIO 管脚，将待转换通道配置为模拟接口(PAD_ADS)。
7. 配置 ADC_GCR[31:16]，启用待转换的通道。
8. 配置 ADC_GCR[10]为 1,启动 ADC 转换。
9. 对需要作为硬件触发 ADC 转换的硬件初始化。
10. 配置 ADC_HDT 相应的硬件触发事件为 1,使能硬件触发。
11. 配置 ADC_HDSET0 或 ADC_HDSET1，选择硬件触发事件的极性。
12. 等待 ADC_ISR 的相应通道标志位为 1，读取 ADC_DR 中的数据。
13. 如需进行多次单次转换，则硬件多次触发事件即可(注：启动 ADC 转换前需确保 ADC_GCR[10]

为 0)。

14. 如使能了 ADC_IER.CHx_INT_EN 和 ADC_IER[16]中断，则等待中断触发后读取 ADC_DR。

27.5.6 注意事项

- 若在单次扫描模式中使用 RX_INTF 中断，须将 ADC_GCR[4]设为 0，否则 RXFIFO 中的数据会因为达不到 8 个而无法触发中断。
- 在单次扫描模式中，若某通道的数据中断 CHx_INTF 被使能而且触发，则 ADC 控制器将结束模数转换，停止读取后面未采样通道的数据。已采样通道的 CHx_DATA 均保留在相应的 ADC_DRx 寄存器。
- 在本芯片设计中，为了与 ADC 模块配合，ADC_GCR 寄存器的 DATA_SAMP_NEG 位只能设为 0，ADC_SPW 寄存器的值 SAMPCLK_WIDTH 应该大于或等于 3。
- 每产生一次硬件触发事件，ADC 只转换一次，即为单次采样模式，使能硬件触发时建议将 ADC 模式配置为单次扫描模式。

27.6 ADC 经 OPA 缓冲采样使用流程

27.6.1 ADC 经 OPA 缓冲采样图

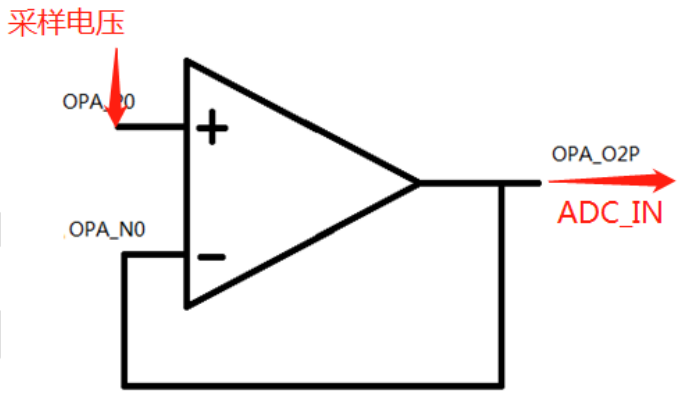


图 27-1：ADC 经 OPA 缓冲采样图

27.6.2 ADC 经 OPA 缓冲采样流程图

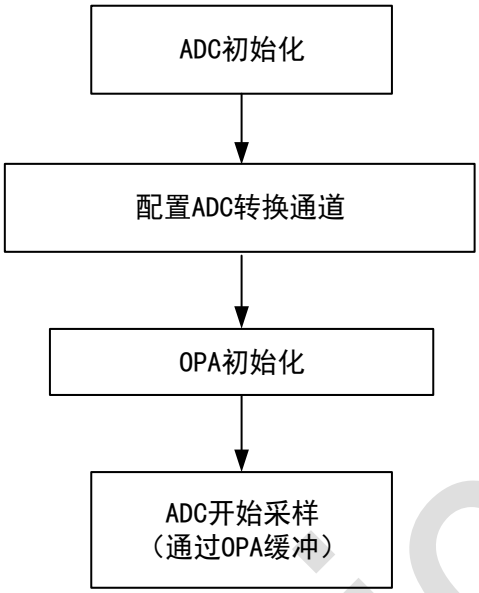


图 27-2：ADC 经 OPA 缓冲采样流程图

27.6.3 ADC 经 OPA 缓冲后采样使用流程

1. ADC 初始化。选择 ADC 参考电压源、ADCCLK 分频和采样模式。
2. 配置 ADC 转换通道。
3. OPA 初始化。配置成 OPA 正端内部接 ADC 输入，OPA 负端通过内部连接输出，使能 ADC 输入通道经 OPA 缓冲。
4. ADC 开始进行采样。

28 OPA

28.1 概述

OPA 是一款具有轨到轨输入和 AB 类输出级的运算放大器。输入输出端可以根据需要配置成不同连接。偏移电压可以被修调。

28.2 主要特性

- 一个运算放大器
- 电压范围：2.5~5.5V

28.3 功能框图

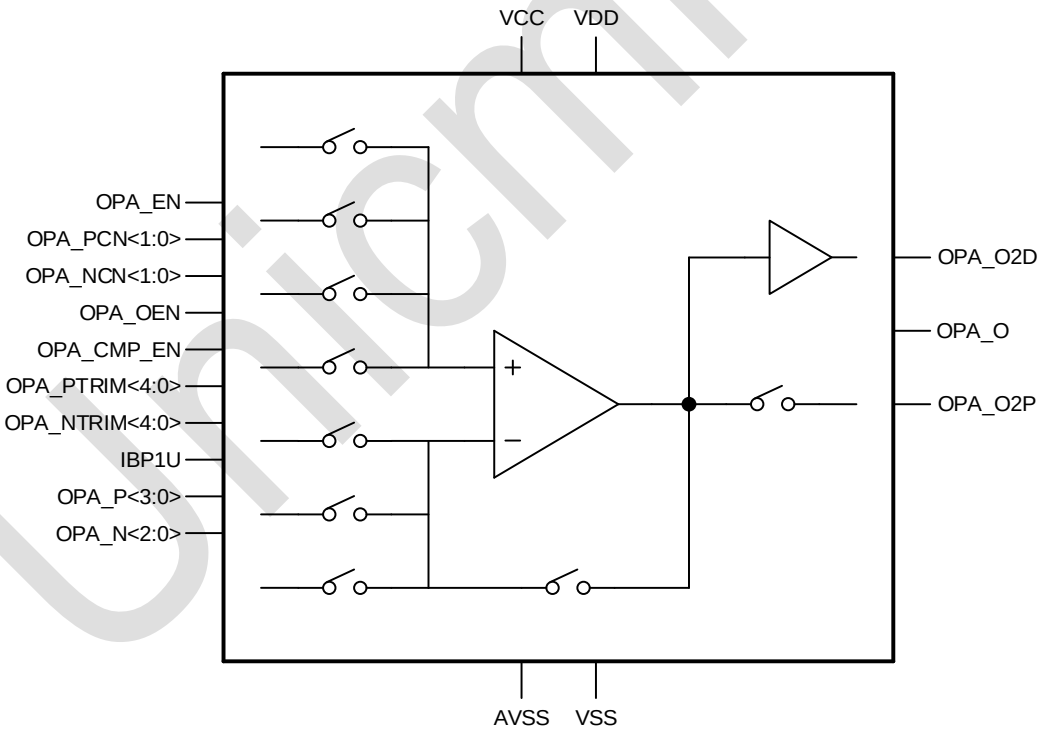


图 28-1：OPA 功能框图

28.4 寄存器描述

寄存器基地址：0x4000_2000

表 28-1：OPA 寄存器列表

偏置	名称	描述
0x0BC	OPA_CFG	OPA 控制寄存器

28.4.1 OPA 控制寄存器 OPA_CFG（偏移：0BCh）

比特	名称	属性	复位值	描述
31:14	RSV	R	0	保留
13	IBIAS_EN	RW	0	IBIAS 使能位： 1：使能 IBIAS，比较器和 OPA 正常工作 0：禁止 IBIAS，比较器和 OPA 禁止工作
12	OPA_PSET	RW	0	OPA 中断设置： 1：OPA 上升沿触发 0：OPA 电平触发
11	OPA_LVEN	RW	0	OPA 作为比较器滤波使能信号： 1：滤波使能 0：滤波关闭
10:9	OPA_LVSET	RW	0	OPA 作为比较器滤波时间设置： 00：输出滤波 2 个 32K 系统低速时钟 01：输出滤波 4 个 32K 系统低速时钟 10：输出滤波 8 个 32K 系统低速时钟 11：输出滤波 16 个 32K 系统低速时钟
8	OPA_POL	RW	1	OPA 作为比较器使用时极性选择： 1：P 端电压高时输出 1 0：N 端电压高时输出 1
7	OPA_INTEN	RW	0	OPA 作为比较器使用时，中断使能： 1：OPA 中断使能 0：OPA 中断不使能
6:5	OPA_PCN	RW	0	OPA 正端信号选择： 00：选择 ADC 输入 01：OPA_P0 10：OPA_P1 11：OPA_P2
4	OPA_OEN	RW	0	OPA 输出使能： 1：OPA 输出使能 0：OPA 输出关闭
3:2	OPA_NCN	RW	0	OPA 负端信号选择： 00：OPA_N0 01：OPA_N1 10：OPA_N2 11：内部连接 OPA 输出

比特	名称	属性	复位值	描述
1	OPA_EN	RW	0	OPA 使能： 1：开启 OPA 功能 0：关闭 OPA 功能
0	OPA_COMP_EN	RW	0	OPA 比较器功能使能 1：开启 OPA 比较器功能 0：关闭 OPA 比较器功能

28.5 OPA 使用流程

28.5.1 OPA 作信号跟随器

1. 配置 PAD_ADS 寄存器，配置 OPA 的输入引脚为模拟接口。
2. 配置 OPA_CFG[1]，使能 OPA 功能。
3. 配置 OPA_CFG[6:5]，配置 OPA 正端的信号选择（如 ADC 使用 OPA 缓冲则选择内部接 ADC 输入）。
4. 配置 OPA_CFG[3:2]，配置 OPA 负端的信号，选择负端通过内部连接输出。
5. 配置 OPA_CFG[4]，使能 OPA 输出。
6. OPA 正端输入电压信号，单倍放大电压在 OPA_O2P（即 PB2）输出。

28.5.2 OPA 作放大器

1. 配置 PAD_ADS 寄存器，配置 OPA 的输入引脚为模拟接口。
2. 配置 OPA_CFG[1]，使能 OPA 功能。
3. 配置 OPA_CFG[6:5]，配置 OPA 正端的信号选择（若 ADC 使用 OPA 缓冲则选择内部接 ADC 输入）。
4. 配置 OPA_CFG[3:2]，配置 OPA 负端的信号选择（若选择负端通过内部连接输出则 OPA 是作为信号跟随器，不放大电压）。
5. 配置 OPA_CFG[4]，使能 OPA 输出。
6. OPA 负端接 GND，OPA 正端接需要放大的电压信号，OPA 的放大倍数由硬件设计决定，放大的电压在 OPA_O2P（即 PB2）输出。

28.5.3 OPA 作比较器

1. 配置 PAD_ADS 寄存器，配置 OPA 的输入引脚为模拟接口。
2. 配置 OPA_CFG[1]，使能 OPA 功能。
3. 配置 OPA_CFG[6:5]，配置 OPA 正端的信号选择。

4. 配置 OPA_CFG[3:2]，配置 OPA 负端的信号选择。
5. 配置 OPA_CFG[0]，使能 OPA 作为 COMP 功能。
6. 配置 OPA_CFG[8]，选择 OPA 作为 COMP 时的极性。
7. 配置 OPA_CFG[10:9]，设置滤波时间。
8. 配置 OPA_CFG[11]，使能滤波。
9. 配置 OPA_CFG[7]，使能 OPA 中断。
10. 根据极性选择，当 P 端或者 N 端的电压比另外一个高时，会触发中断。

28.5.4 注意事项

OPA 的状态为在系统配置（SCU）中的模拟状态寄存器 ANALOG_STATUS 中查看。

29 COMP

29.1 概述

COMP 是一款具有轨到轨输入的迟滞比较器，输入端可根据需要配置。COMP 可用作电压比较，有两个输入端 IN+和 IN-，可选择其中一个输入端作为参考点来比较，当另一输入端电压小于参考电压时比较器输出低电平，反之输出高电平。

29.2 主要特性

- 4 个电压比较器
- 可产生比较中断

29.3 功能框图

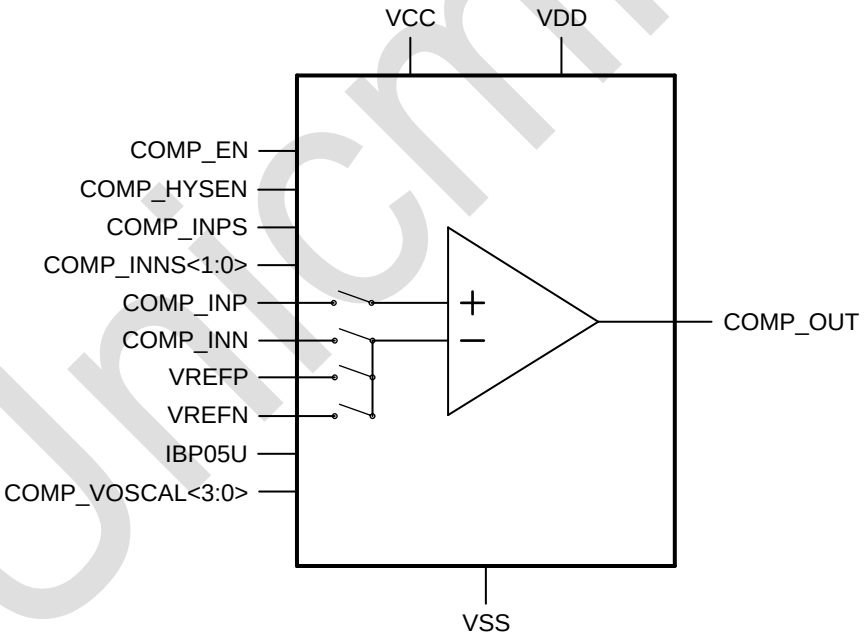


图 29-1：COMP 功能框图

29.4 寄存器描述

寄存器基地址：0x4000_2000

表 29-1：COMP 寄存器列表

偏置	名称	描述
0x0C0	COMP0_CFG	COMP0 控制寄存器
0x0C4	COMP1_CFG	COMP1 控制寄存器
0x0C8	COMP2_CFG	COMP2 控制寄存器
0x0CC	COMP3_CFG	COMP3 控制寄存器
0x0F8	COMP_INEN	COMP 中断控制寄存器

29.4.1 COMP0 控制寄存器 COMP0_CFG（偏移：0C0h）

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11	COMP0_PSET	RW	0	比较器 0 中断设置： 1：COMP0 上升沿触发 0：COMP0 电平触发
10	COMP0_LVEN	RW	0	比较器 0 滤波使能信号： 1：滤波使能 0：滤波关闭
9:8	COMP0_LVSET	RW	0	比较器 0 滤波时间设置： 00：比较器 0 输出滤波 2 个 PCLK 时钟 01：比较器 0 输出滤波 4 个 PCLK 时钟 10：比较器 0 输出滤波 8 个 PCLK 时钟 11：比较器 0 输出滤波 16 个 PCLK 时钟
7	COMP0_POL	RW	1	COMP0 极性选择： 1：COMP0 P 端电压高时输出 1 0：COMP0 N 端电压高时输出 1
6:5	COMP0_INPS	RW	0	COMP0 正端信号选择： 00：PD0 01：PB2 10：GND 11：LDO1.2v 输出
4:3	COMP0_INNS	RW	0	COMP0 负端信号选择： 00：PD1 01：PD4 10：GND 11：LDO1.2v 输出
2	COMP0_HYSEN	RW	0	使能 COMP0 迟滞滤波； 1：使能比较器迟滞滤波功能 0：关闭比较器迟滞滤波功能
1	COMP0_EN	RW	0	COMP0 使能： 1：使能 COMP0 0：关闭 COMP0

比特	名称	属性	复位值	描述
0	COMP0_CALEN	RW	0	COMP0 校准使能： 1：使能 COMP0 校准功能 0：关闭 COMP0 校准功能

29.4.2 COMP1 控制寄存器 COMP1_CFG（偏移：0C4h）

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11	COMP1_PSET	RW	0	比较器 1 中断设置： 1：COMP1 上升沿触发 0：COMP1 电平触发
10	COMP1_LVEN	RW	0	比较器 1 滤波使能信号： 1：滤波使能 0：滤波关闭
9:8	COMP1_LVSET	RW	0	比较器 1 滤波时间设置： 00：比较器 1 输出滤波 2 个 PCLK 时钟 01：比较器 1 输出滤波 4 个 PCLK 时钟 10：比较器 1 输出滤波 8 个 PCLK 时钟 11：比较器 1 输出滤波 16 个 PCLK 时钟
7	COMP1_POL	RW	1	COMP1 极性选择： 1：COMP1 P 端电压高时输出 1 0：COMP1 N 端电压高时输出 1
6:5	COMP1_INPS	RW	0	COMP1 正端信号选择： 00：PD3 01：PB2 10：GND 11：LDO1.2v 输出
4:3	COMP1_INNS	RW	0	COMP1 负端信号选择： 00：PD2 01：PD4 10：GND 11：LDO1.2v 输出；
2	COMP1_HYSEN	RW	0	使能 COMP1 迟滞滤波： 1：使能比较器迟滞滤波功能 0：关闭比较器迟滞滤波功能
1	COMP1_EN	RW	0	COMP1 使能： 1：使能 COMP1 0：关闭 COMP1
0	COMP1_CALEN	RW	0	COMP1 校准使能： 1：使能 COMP1 校准功能 0：关闭 COMP1 校准功能

29.4.3 COMP2 控制寄存器 COMP2_CFG（偏移：0C8h）

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11	COMP2_PSET	RW	0	比较器 2 中断设置： 1：COMP2 上升沿触发 0：COMP2 电平触发
10	COMP2_LVEN	RW	0	比较器 2 滤波使能信号： 1：滤波使能 0：滤波关闭
9:8	COMP2_LVSET	RW	0	比较器 2 滤波时间设置： 00：比较器 2 输出滤波 2 个 PCLK 时钟 01：比较器 2 输出滤波 4 个 PCLK 时钟 10：比较器 2 输出滤波 8 个 PCLK 时钟 11：比较器 2 输出滤波 16 个 PCLK 时钟
7	COMP2_POL	RW	1	COMP2 极性选择： 1：COMP2 P 端电压高时输出 1 0：COMP2 N 端电压高时输出 1
6:5	COMP2_INPS	RW	0	COM2 正端信号选择： 00：PD4 01：PD5 10：PB2 11：LDO1.2v 输出
4:3	COMP2_INNS	RW	0	COMP2 负端信号选择： 00：PD5 01：PD4 10：GND 11：LDO1.2v 输出
2	COMP2_HYSEN	RW	0	使能 COMP2 迟滞滤波： 1：使能比较器迟滞滤波功能 0：关闭比较器迟滞滤波功能
1	COMP2_EN	RW	0	COMP2 使能： 1：使能 COMP2 0：关闭 COMP2
0	COMP2_CALEN	RW	0	COMP2 校准使能： 1：使能 COMP2 校准功能 0：关闭 COMP2 校准功能

29.4.4 COMP3 控制寄存器 COMP3_CFG（偏移：0CCh）

比特	名称	属性	复位值	描述
31:12	RSV	R	0	保留
11	COMP3_PSET	RW	0	比较器 3 中断设置： 1：COMP3 上升沿触发 0：COMP3 电平触发

比特	名称	属性	复位值	描述
10	COMP3_LVEN	RW	0	比较器 3 滤波使能信号： 1：滤波使能 0：滤波关闭
9:8	COMP3_LVSET	RW	0	比较器 3 滤波时间设置： 00：比较器 3 输出滤波 2 个 PCLK 时钟 01：比较器 3 输出滤波 4 个 PCLK 时钟 10：比较器 3 输出滤波 8 个 PCLK 时钟 11：比较器 3 输出滤波 16 个 PCLK 时钟
7	COMP3_POL	RW	1	COMP3 极性选择： 1：COMP3 P 端电压高时输出 1 0：COMP3 N 端电压高时输出 1
6:5	COMP3_INPS	RW	0	COM3 正端信号选择： 00：PC2 01：PD2 10：PD5 11：PB2
4:3	COMP3_INNS	RW	0	COMP3 负端信号选择： 00：PC1 01：PD4 10：PD1 11：PD5
2	COMP3_HYSEN	RW	0	使能 COMP3 迟滞滤波： 1：使能比较器迟滞滤波功能 0：关闭比较器迟滞滤波功能
1	COMP3_EN	RW	0	COMP3 使能： 1：使能 COMP3 0：关闭 COMP3
0	COMP3_CALEN	RW	0	COMP3 校准使能： 1：使能 COMP3 校准功能 0：关闭 COMP3 校准功能

29.4.5 COMP 中断控制寄存器 COMP_INEN（偏移：0F8h）

比特	名称	属性	复位值	描述
31:4	RSV	R	0	保留
3	COMP3_INTEN	RW	0	COMP3 中断使能： 1：COMP3 中断使能 0：COMP3 中断关闭
2	COMP2_INTEN	RW	0	COMP2 中断使能： 1：COMP2 中断使能 0：COMP2 中断关闭
1	COMP1_INTEN	RW	0	COMP1 中断使能： 1：COMP1 中断使能 0：COMP1 中断关闭
0	COMP0_INTEN	RW	0	COMP0 中断使能： 1：COMP0 中断使能 0：COMP0 中断关闭

29.5 COMP 使用流程

1. 配置 IO 复用为 COMP_OUT，作为输出端。
2. 配置 PAD_ADS 寄存器，将两个输入端 IO 配置为模拟接口。
3. 配置 COMP_CFG 寄存器，配置 COMP 极性、正端信号选择、负端信号选择。
4. 配置 COMP_INEN 寄存器，使能中断并配置 ANALOG_STATUS 清中断状态标志。
5. 配置 COMP_CFG[1]寄存器，使能 COMP。

30 AES

30.1 概述

AES 算法是一个分组算法。加密算法与密钥扩展算法都采用非线性迭代结构。解密算法与加密算法的结构相同，只是轮密钥的使用顺序相反，解密轮密钥是加密轮密钥的逆序。

30.2 主要特性

- 支持密钥长度 128、192 和 256
- 支持 AES 加密和解密运算
- 支持 Electronic Code Book (ECB)模式和 Cipher Block Chaining (CBC)模式
- 数据输入和输出支持 SWAP 模式，即大小端可配置

30.3 寄存器描述

寄存器基地址：0x4002_1400

表 30-1：AES 寄存器列表

偏置	名称	描述
0x00	AES_DATAIN	数据输入寄存器
0x04	AES_KEYIN	密钥输入寄存器
0x08	AES_IVIN	初始向量输入寄存器
0x0C	AES_CONTROL	控制寄存器
0x10	AES_STATE	状态寄存器
0x14	AES_DATAOUT	数据输出寄存器

30.3.1 数据输入寄存器 AES_DATAIN (偏移：00h)

比特	名称	属性	复位值	描述
31:0	DATAIN	WO	0x00000000	32 位寄存器，用来存放明文或密文的数据。需要连续写入 4 次，组成 128 比特数据流。

30.3.2 密钥输入寄存器 AES_KEYIN (偏移：04h)

比特	名称	属性	复位值	描述
31:0	KEYIN	WO	0x00000000	32 位寄存器，用来存放密钥数据。需要连续写入 4 次，组成 128 比特密钥。

30.3.3 初始向量输入寄存器 AES_IVIN (偏移：08h)

比特	名称	属性	复位值	描述
31:0	IVIN	WO	0x00000000	32 位寄存器，用来存放 CBC 工作模式时的初始向量。需要连续写入 4 次，组成 128 比特初始向量。该寄存器只在 CBC 模式下有效，在 ECB 模式下，即使写入数据也不起作用。

30.3.4 控制寄存器 AES_CONTROL (偏移：0Ch)

比特	名称	属性	复位值	描述
31:12	RSV	-	-	保留
11:9	ALL_ROUND	R/W	0	打开伪 AES 运算时的总轮数： 0：2 轮 1：4 轮 2：8 轮 3：16 轮 4：32 轮 5：64 轮 其它：保留
8	VAES_EN	R/W	0	伪 AES 运算使能位： 1：使能 0：禁止
7:6	KEY_MODE	R/W	0	AES 密钥长度模式选择： 00：AES_128 01：AES_192 10：AES_256
5	ECB	R/W	0	模式指示位： 1 = CBC 模式。 0 = ECB 模式。
4	SWAP	R/W	0	数据输入输出 SWAP 模式： 1 = SWAP 模式使能 0 = SWAP 模式禁止
3	INT_EN	R/W	0	中断使能位： 1 = 使能加密/解密完成中断 0 = 禁止加密/解密完成中断
2	CRYPT	R/W	0	加密/解密指示位： 1 = 进行解密运算。 0 = 进行加密运算。
1	KEY_START	W	0	密钥扩展运算启动位： 1：写 1 启动密钥扩展运算。读该比特总是返回 0 0：写 0 不起作用。读该比特总是返回 0
0	CRYPT_START	W	0	加解/解密运算启动位： 1：写 1 启动加解/解密运算。读该比特总是返回 0 0：写 0 不起作用。读该比特总是返回 0

30.3.5 状态寄存器 AES_STATE (偏移：10h)

比特	名称	属性	复位值	描述
31:3	RSV	-	-	保留
2	CBCDONE	R	0	第一组 CBC 完成。 1：第一组 CBC 完成。对该位写 1，则清除该标志位。当一批数据采用 CBC 方式加密或解密完成后，必须要写 1 清除该标志位，这样才能保证下一批数据加解密正确。此标志位仅在 CBC 模式下有效，ECB 模式下无效。 0：第一组 CBC 没有完成。此标志位仅在 CBC 模式下有效，ECB 模式下无效。
1	KEY_DONE	R	0	AES 密钥扩展完成位： 1：AES 密钥扩展完成，对该位写 1，则清除标志位。 0：AES 密钥扩展没有完成。
0	CRYPT_DONE	R	0	AES 运算完成位： 1：AES 运算完成，对该位写 1，则清除标志位。如果中断使能，该位置位时，同时产生中断；向该位写 1，则清除中断。 0：AES 运算没有完成。

30.3.6 数据输出寄存器 AES_DATAOUT (偏移：14h)

比特	名称	属性	复位值	描述
31:0	DATAOUT	RO	0x00000000	结果寄存器用于存放加解密的结果数据，只读。

30.4 AES 使用流程

AES 模块加解密流程如下图所示：

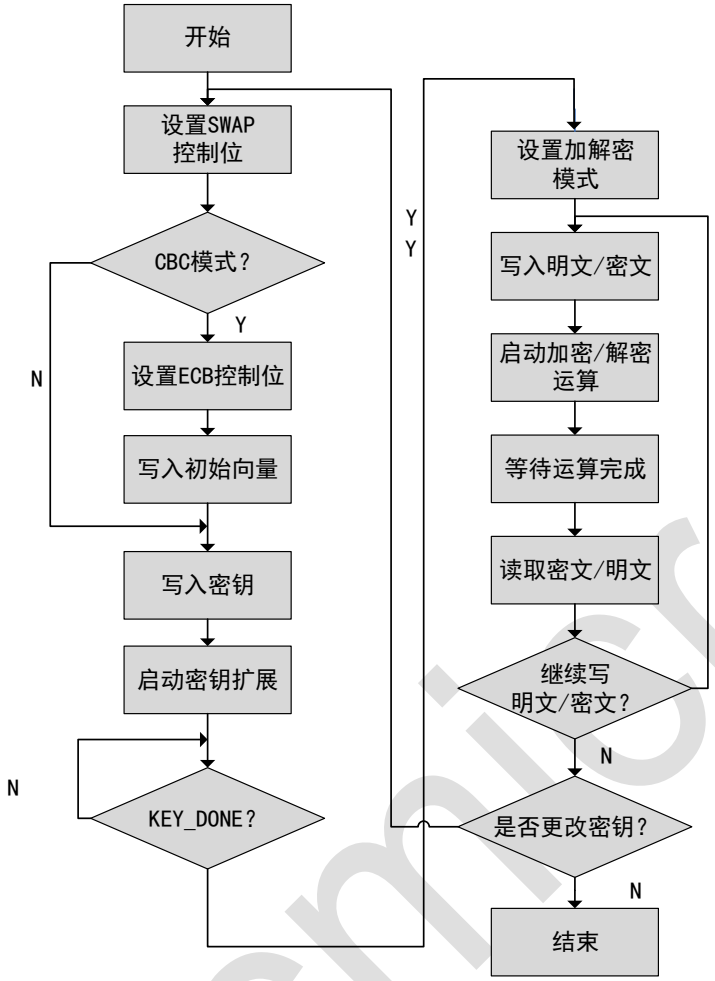


图 30-1：AES 模块加解密流程图

31 DIV

31.1 概述

DIV 的实现目标是能够支持不超过 32bit 的除法。除数不大于 32bit，被除数可以为任意 bit。主要应用于除数小于 32bit 的应用。

31.2 主要特性

支持不超过 32bit 的除法（除数不大于 32bit，被除数可以为任意 bit）；

31.3 寄存器描述

DIV 模块基地址为: 0x4002_1000

表 31-1：DIV 寄存器列表

偏置	名称	描述
0x00	DIV_DIVIDEND	被除数寄存器
0x04	DIV_DIVISOR	除数寄存器
0x08	DIV_REMAIN	余数寄存器
0x0C	DIV_QUOTIENT	商寄存器
0x10	DIV_STATUS	状态寄存器
0x14	DIV_INTEN	状态寄存器

31.3.1 被除数寄存器 DIV_DIVIDEND (偏移：00h)

比特	名称	属性	复位值	描述
31:0	DIVIDEND	R/W	0	被除数

31.3.2 除数寄存器 DIV_DIVISOR (偏移：04h)

比特	名称	属性	复位值	描述
31:0	DIVISOR	R/W	0	除数

31.3.3 余数寄存器 DIV_REMAIN (偏移：08h)

比特	名称	属性	复位值	描述
31:0	REMAIN	R/W	0	余数

31.3.4 商寄存器 DIV_QUOTIENT (偏移：0Ch)

比特	名称	属性	复位值	描述
31:0	QUOTIENT	R/W	0	商

31.3.5 状态寄存器 DIV_STATUS (偏移：10h)

比特	名称	属性	复位值	描述
31:1	RSV	-	0	保留位
0	DONE	R/W	0	运算完成标志位，写 1 清除标志： 1：运算完成 0：运算未完成

31.3.6 状态寄存器 DIV_INTEN (偏移：14h)

比特	名称	属性	复位值	描述
31:1	RSV	-	0	保留位
0	INTEN	R/W	0	运算完成中断使能位： 1：中断使能 0：中断禁止

31.4 使用流程

1. 往除数寄存器中写入除数。
2. 往被除数寄存器中写入被除数。
3. 隔 32 个算法时钟周期或者查询完成状态位从商寄存器中读出商，如果被除数还未输入完，则转至步骤 2，如果输入完则转至步骤 4。
4. 从余数寄存器中读出余数，运算结束。

32 SysTick

32.1 概述

OS 要想支持多任务，就需要周期执行上下文切换，这样就需要有定时器之类的硬件资源打断程序执行。当定时器中断产生时，处理器就会在异常处理中进行 OS 任务调度，同时还会进行 OS 维护的工作。Cortex-M0+处理器中有一个称为 SysTick 的简单定时器，用于产生周期性的中断请求。

SysTick 为 24 位的定时器，并且向下计数。定时器的计数减到 0 后，就会重新装载一个可编程的数值，并且同时产生 SysTick 异常（异常编号为 15），该异常事件会引起 SysTick 异常处理的执行，这个过程是 OS 的一部分。

对于不需要 OS 的系统，SysTick 定时器也可以用作其他用途，比如定时、计时或者为需要周期执行的任务提供中断源。SysTick 异常的产生是可控的，如果异常被禁止，仍然可以用轮询的方法使用 SysTick 定时器，比如检查当前的计数值或者轮询溢出标志。

32.2 寄存器描述

SysTick 寄存器基地址：0xE000_E010

表 32-1：SysTick 寄存器列表

偏置	名称	描述
0x00	SysTick_CTRL	SysTick 控制和状态寄存器
0x04	SysTick_LOAD	SysTick 重载值寄存器
0x08	SysTick_VAL	SysTick 当前值寄存器

32.2.1 控制和状态寄存器 SysTick_CTRL (偏移：00h)

比特	名称	属性	复位值	描述
31:17	RSV	-	-	保留
16	COUNTFLAG	R	0	Systick定时器溢出标志： 1：Systick定时器发生下溢出 0：Systick定时器未发生溢出 读该寄存器，可清除 COUNTFLAG 标志
15:2	RSV	-	-	保留
1	TICKINT	R/W	0	SysTick中断使能： 1：使能中断 0：禁止中断
0	ENABLE	R/W	0	SysTick定时器使能 ： 1：使能SysTick 0：禁止SysTick

32.2.2 重载值寄存器 SysTick_LOAD (偏移：04h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:0	RELOAD	R/W	0xFFFFFF	SysTick 定时器重载值

32.2.3 当前值寄存器 SysTick_VAL (偏移：08h)

比特	名称	属性	复位值	描述
31:24	RSV	-	-	保留
23:0	CURRENT	R/W	0xFFFFFF	读该寄存器,获取 SysTick 定时器的当前计数值；写任意值到该寄存器，清零该寄存器及 COUNTFLAG。

32.3 使用流程

由于 SysTick 定时器的重载值和当前值在复位时都是未定义的，为了防止产生异常结果，对 SysTick 的配置需要遵循一定的流程：

1. 配置 SysTick_CTRL[0]为 0，禁止 SysTick。
2. 配置 SysTick_LOAD，选择 SysTick 的溢出周期。
3. 向 SysTick_VAL 写入任意值，清零 SysTick_VAL 及 SysTick_CTRL。
4. COUNTFLAG。
5. 配置 SysTick_CTRL[1]为 1，使能 SysTick 中断。
6. 配置 SysTick_CTRL[0]为 1，使能 SysTick。
7. 查询等待定时器溢出标志到来之后关闭和清空计数器，或者在中断服务程序中读取 SysTick_CTRL 以清除溢出标志。

33 调试支持（DBG）

集成硬件调试模块的 Cortex®-M0 内核。支持指令断点（指令取值时停止）和数据断点（数据访问时停止）。当内核停止时，用户可以查看内核的内部状态和系统的外部状态。用户查询操作完成后，

可以恢复内核和外设，继续执行相应的程序。 芯片内核的硬件调试模块在连接到调试器时即可使用（在未被禁用的情况下）。

支持串行接口（两线 SWD）调试接口。

34 版本维护

日期	版本	描述
2023/03/17	V1.0	初始版本
2023/05/08	V1.1	更新控制寄存器 EFC_CTRL 描述； 更新 ADC 管脚 ADC_14，ADC_15 描述； 修订 WWDT 章节中递减描述为递增； 删除内置 VREF 的相关描述。
2023/07/04	V1.2	新增 UM32MP32 型号相关描述