用户手册

G32R501

基于 Arm® Cortex® -M52 内核的 32 位实时微控制器

版本：V1.6

© 珠海极海半导体有限公司

目录

1 简介及文档描述规则 12

1.1 简介 12

1.2 文档描述规则 13

2 CPU 及系统架构 15

2.1 术语全称、缩写描述 15

2.2 简介 15

2.3 主要特征 15

2.4 结构框图 19

2.5 功能描述 20

2.6 寄存器组地址 32

2.7 寄存器地址映射 32

2.8 寄存器功能描述 33

3 数学指令拓展（Zidian） 42

4 系统启动（BOOT） 43

4.1 简介 43

4.2 功能描述 43

4.3 寄存器 44

5 写寄存器保护（WRPRT） 45

5.1 术语全称、缩写描述 45

5.2 简介 45

5.3 主要特征 45

5.4 功能描述 45

5.5 寄存器组地址 46

5.6 寄存器地址映射 47

5.7 寄存器功能描述 47

6 UID 48

6.1 寄存器组地址 48

6.2 寄存器地址映射 48

6.3 寄存器功能描述 48

7 电源管理 50

8 设备标识和配置寄存器 51
8.1 简介 51
8.2 寄存器组地址 51
8.3 寄存器地址映射 51
8.4 寄存器功能描述 52
9 低功耗模式 59
9.1 IDLE 59
9.2 HALT 59
10 复位 62
10.1 功能描述 62
11 时钟 64
11.1 结构框图 64
11.2 功能描述 64
11.3 寄存器组地址 71
11.4 寄存器地址映射 72
11.5 寄存器功能描述 72
12 系统控制 79
12.1 SPEC 功能描述 79
12.2 SPEC 寄存器组地址 79
12.3 SPEC 寄存器地址映射 79
12.4 SPEC 寄存器功能描述 80
12.5 NMI_INTRUPT 寄存器组地址 82
12.6 NMI_INTRUPT 寄存器地址映射 83
12.7 NMI_INTRUPT 寄存器功能描述 83
12.8 PERIPH_AC 寄存器组地址 86
12.9 PERIPH_AC 寄存器地址映射 86
12.10 PERIPH_AC 寄存器功能描述 88
12.11 ROM_PREFETCH 寄存器组地址 99
12.12 ROM_PREFETCH 寄存器地址映射 99
12.13 ROM_PREFETCH 寄存器功能描述 99
12.14 ROM_WAIT_STATE 寄存器组地址 100
12.15 ROM_WAIT_STATE 寄存器地址映射 100

12.16 ROM_WAIT_STATE 寄存器功能描述 100
13 嵌套向量中断控制器（NVIC） 102
13.1 术语全称、缩写描述 102
13.2 简介 102
13.3 主要特征 102
13.4 中断和异常向量表 102
14 外部中断与事件控制器（EXTI） 110
14.1 简介 110
14.2 主要特征 110
14.3 功能描述 110
14.4 寄存器组地址 112
14.5 寄存器地址映射 112
14.6 寄存器功能描述 113
15 双码安全（DCS） 118
16 可编程静态存储子系统（CFGSMS） 119
17 非易失性存储器控制器（NVMC） 120
17.1 简介 120
17.2 结构框图 121
17.3 主要特征 122
17.4 功能描述 123
17.5 寄存器组地址 131
17.6 寄存器地址映射 131
17.7 寄存器功能描述 132
18 定时器 0/1/2（TMR0/1/2） 146
18.1 术语全称、缩写描述 146
18.2 简介 146
18.3 主要特征 146
18.4 结构框图 146
18.5 功能描述 146
18.6 寄存器组地址 147
18.7 寄存器地址映射 147

18.8 寄存器功能描述 148
19 看门狗定时器（WDT） 151
19.1 术语全称、缩写描述 151
19.2 简介 151
19.3 主要特征 151
19.4 结构框图 152
19.5 功能描述 152
19.6 寄存器组地址 154
19.7 寄存器地址映射 155
19.8 寄存器功能描述 155
20 双时钟比较器（DCCOMP） 159
20.1 简介 159
20.2 主要特征 159
20.3 结构框图 159
20.4 功能描述 160
20.5 寄存器组地址 162
20.6 寄存器地址映射 163
20.7 寄存器功能描述 163
21 通用输入/输出引脚（GPIO） 168
21.1 术语全称、缩写描述 168
21.2 简介 168
21.3 主要特征 168
21.4 结构框图 168
21.5 功能描述 170
21.6 GPIO 复用功能配置 178
21.7 寄存器组地址 184
21.8 寄存器地址映射 184
21.9 寄存器功能描述 186
22 处理器间通信单元（IPC） 238
22.1 简介 238
22.2 主要特征 238
22.3 功能描述 238

22.4 寄存器地址映射 241
22.5 寄存器功能描述 242
23 X-BAR 249
23.1 简介 249
23.2 主要特征 249
23.3 功能描述 250
23.4 寄存器组地址 258
23.5 寄存器地址映射 258
23.6 寄存器功能描述 261
24 直接存储器存取（DMA） 291
24.1 术语全称、缩写描述 291
24.2 简介 291
24.3 主要特征 291
24.4 结构框图 292
24.5 功能描述 292
24.6 寄存器组地址 300
24.7 寄存器地址映射 301
24.8 寄存器功能描述 302
25 双核调试系统（DCDS） 315
25.1 术语全称、缩写描述 315
25.2 简介 315
25.3 结构框图 316
25.4 主要特征 317
25.5 功能描述 317
26 模拟子系统（AS） 325
26.1 简介 325
26.2 主要特征 325
26.3 结构框图 325
26.4 功能描述 330
26.5 寄存器组地址 337
26.6 寄存器地址映射 338
26.7 寄存器功能描述 338

27 模数转换器（ADC） 344
27.1 术语全称、缩写描述 344
27.2 简介 344
27.3 主要特征 344
27.4 结构框图 345
27.5 功能描述 345
27.6 寄存器组地址 362
27.7 寄存器地址映射 362
27.8 寄存器功能描述 365
28 缓冲数模转换器（DAC） 387
28.1 术语全称、缩写描述 387
28.2 简介 387
28.3 主要特征 387
28.4 结构框图 387
28.5 功能描述 388
28.6 锁定寄存器 389
28.7 寄存器组地址 389
28.8 寄存器地址映射 389
28.9 寄存器功能描述 390
29 比较器（COMP） 393
29.1 术语全称、缩写描述 393
29.2 简介 393
29.3 主要特征 393
29.4 结构框图 394
29.5 功能描述 395
29.6 寄存器组地址 402
29.7 寄存器地址映射 403
29.8 寄存器功能描述 403
30 Sigma-Delta 滤波器（SDF） 412
30.1 术语全称、缩写描述 412
30.2 简介 412
30.3 主要特征 412

30.4 结构框图 413
30.5 功能描述 415
30.6 寄存器组地址 427
30.7 寄存器地址映射 428
30.8 寄存器功能描述 429
31 脉冲宽度调制器（PWM） 438
31.1 术语全称、缩写描述 438
31.2 简介 438
31.3 主要特征 438
31.4 结构框图 442
31.5 功能描述 444
31.6 寄存器组地址 509
31.7 寄存器地址映射 509
31.8 寄存器功能描述 512
32 高分辨率脉冲宽度调制器（HRPWM） 580
32.1 术语全称、缩写描述 580
32.2 简介 580
32.3 主要特征 580
32.4 结构框图 581
32.5 功能描述 581
32.6 寄存器 596
33 捕获器（CAP） 597
33.1 术语全称、缩写描述 597
33.2 简介 597
33.3 主要特征 597
33.4 结构框图 599
33.5 CAP 引脚功能描述 599
33.6 捕获描述功能描述 601
33.7 APWM 模式功能描述 606
33.8 CAP 模块应用 609
33.9 寄存器组地址 612
33.10 寄存器地址映射 612

33.11 寄存器功能描述 613
34 高分辨率捕获器（HRCAP） 620
34.1 术语全称、缩写描述 620
34.2 简介 620
34.3 主要特征 620
34.4 结构框图 621
34.5 功能描述 621
34.6 寄存器组地址 624
34.7 寄存器地址映射 624
34.8 寄存器功能描述 624
35 正交编码器脉冲（QEP） 628
35.1 术语全称、缩写描述 628
35.2 简介 628
35.3 主要特征 628
35.4 结构框图 629
35.5 功能描述 629
35.6 正交模式适配器（QMA）功能描述 635
35.7 正交解码器单元（QDU）功能描述 637
35.8 位置计数器和控制单元（PCCU）功能描述 640
35.9 边沿捕获单元功能描述 646
35.10 寄存器组地址 649
35.11 寄存器地址映射 649
35.12 寄存器功能描述 650
36 串行外设接口（SPI） 665
36.1 术语全称、缩写描述 665
36.2 简介 665
36.3 主要特征 665
36.4 结构框图 666
36.5 功能描述 666
36.6 寄存器组地址 684
36.7 寄存器地址映射 685
36.8 寄存器功能描述 685

37 通用异步收发器（UART） 695
37.1 术语全称、缩写描述 695
37.2 简介 695
37.3 主要特征 695
37.4 结构框图 696
37.5 功能描述 696
37.6 寄存器组地址 703
37.7 寄存器地址映射 703
37.8 寄存器功能描述 704
38 内部集成电路接口（I2C） 713
38.1 术语全称、缩写描述 713
38.2 简介 713
38.3 主要特征 713
38.4 结构框图 714
38.5 功能描述 714
38.6 寄存器组地址 723
38.7 寄存器地址映射 723
38.8 寄存器功能描述 724
39 电源管理总线（PMBus） 736
39.1 术语全称、缩写描述 736
39.2 简介 736
39.3 主要特征 736
39.4 结构框图 737
39.5 功能描述 737
39.6 寄存器组地址 754
39.7 寄存器地址映射 754
39.8 寄存器功能描述 754
40 控制器局域网（CAN） 765
40.1 术语全称、缩写描述 765
40.2 简介 765
40.3 主要特征 765
40.4 结构框图 766

40.5 功能描述 768
40.6 寄存器组地址 808
40.7 寄存器地址映射 808
40.8 寄存器功能描述 809
41 局域互联网络（LIN） 834
41.1 术语全称、缩写描述 834
41.2 简介 834
41.3 UART 主要特征 834
41.4 LIN 主要特征 835
41.5 结构框图 836
41.6 LIN 功能描述 838
41.7 低功耗模式 858
41.8 仿真模式 860
41.9 UART 功能描述 860
41.10 寄存器组地址 875
41.11 寄存器地址映射 875
41.12 寄存器功能描述 876
42 四线串行外设接口（QSPI） 909
43 灵活逻辑块（FLB） 910
44 版本历史 911

1 简介及文档描述规则

1.1 简介

G32R501x 是极海面向实时控制应用研发的高性能微控制器，内置基于 Arm v8.1-M 架构的 Arm® Cortex®-M52 微处理器，最多支持 2 个内核，内核可以并行工作，高效协同，适用于运动控制、光伏逆变，数字电源和车载充电机（OBC）等应用场景。

G32R501x 微控制器最高运行频率可达 250MHz，实现了 Arm Rv8-M 的自定义数据路径扩展（CDE），搭载基于矢量扩充方案（MVE）的 Helium™技术，其通过创新的 Zidian 数学计算扩展指令集，使得处理性能得到进一步提升。Zidian 数学计算扩展指令集可以快速执行包含变换和扭矩环路计算中常见三角运算的算法，以及可以降低编码应用中常见复杂数学运算的延迟。

G32R501x 内部支持最高 640KB 的 Flash，这些 Flash 分为 512KB 和 128KB 两个独立存储体，支持并行编程和执行。此外，通过内置 CFGSMS，可对片上 128KB 的 SRAM 存储进行高效的系统分区，每个逻辑块的大小为 8KB，可配置做不同类型的用法，如 ITCM、DTCM 和 SRAM。此外，G32R501x 还支持 Flash ECC、RAM 奇偶校验和安全属性配置。

G32R501x 片内集成了高性能模拟单元，可以进一步提升系统控制性能。三个独立的 12 位 ADC 可准确、高效地采集&处理多个模拟信号，从而提高系统吞吐量。7 个比较器子系统可以通过跳闸功能对输入电压电平进行持续监控。

G32R501x 包含性能领先的控制外设（具有独立于频率的 PWM 和 CAP/HRCAP），可对系统进行出色的控制。内置的 4 通道 SDF 适配外部隔离 Σ-Δ 调制器。

G32R501x 内置通用通信端口（如 SPI、UART、I2C、LIN 和 CAN），并且提供了多个多路复用选项，可满足各种应用的通信需求。G32R501x 器件还内置完全符合标准的 PMBus 接口以及高速 QSPI 接口。此外，G32R501x 还支持 JTAG、cJTAG 以及 SWD 调试接口，多种调试模式可适用不同系统环境及性能要求。

G32R501x 支持-40℃~105℃/125℃的环境工作温度，提供 QFN56、LQFP64、LQFP80、LQFP100 等多种封装供用选择。

1.1.1 寄存器访问方式

如果手册中描述的寄存器为 16 位寄存器，则只能按照 16 位操作去访问这些寄存器；如果手册中描述的寄存器为 32 位寄存器，则只能按照 32 位操作去访问这些寄存器。CPU 不能发起外设区域非对齐访问，否则会进入 hardfault（硬件错误中断）。

1.2 文档描述规则

1.2.1 寄存器读写方式

表格 1 R/W 缩写及描述

R/W	描述	缩写
read/write	软件能读写此位。	R/W
read-only	软件只能读此位。	R
write-only	软件只能写此位，读此位将返回复位值。	W
read/clear	软件可以读此位，也可以通过写 1 清除此位，写 0 对此位无影响。	RC_W1
read/clear	软件可以读此位，也可以通过写 0 清除此位，写 1 对此位无影响。	RC_W0
read/clear by read	软件可以读此位，读此位将自动地清除它为 0，写此位无效。	RC_R
read/set	软件可以读也可以设置此位，写 0 对此位无影响。	R/S
read-only write trigger	软件可以读此位，写 0 或 1 触发一个事件但对此位数值没有影响。	RT_W
toggle	软件只能通过写 1 来翻转此位，写 0 对此位无影响。	T
Read Returns 0s	读取返回 0	R-0
Write 1 to set	写 1 设置	W1S
Write 1 to clear	写 1 清除	W1C
Write once	写一次	WOnce
Write Set once	写设置一次	WSonce
Write once	能写入一次此位	WO

1.2.2 术语全称、缩写描述

表格 2 模块全称、简写

中文全称	英文全称	英文缩写
复位与时钟管理	Reset and Clock Management	RCM
时钟复原系统	Clock Recovery System	CRS
电源管理单元	Power Management Unit	PMU
嵌套向量中断控制器	Nested Vector Interrupt Controller	NVIC
外部中断/事件控制器	External Interrupt /Event Controller	EXTI
直接存储器存取	Direct Memory Access	DMA
调试 MCU	Debug MCU	DBG MCU
通用输入/输出引脚	General-Purpose Input Output Pin	GPIO
复用功能输入/输出引脚	Alternate Function Input Output Pin	AFIO

中文全称	英文全称	英文缩写
定时器	Timer	TMR
看门狗定时器	Watchdog Timer	WDT
独立看门狗	Independent Watchdog Timer	IWDT
窗口看门狗	Windows Watchdog Timer	WWDT
实时时钟	Real-Time Clock	RTC
通用同步异步收发器	Universal Synchronous Asynchronous Receiver Transmitter	UART
内部集成电路接口	Inter-integrated Circuit Interface	I2C
串行外设接口	Serial Peripheral Interface	SPI
片上音频接口	Inter-IC Sound Interface	I2S
控制器局域网	Controller Area Network	CAN
模拟数字转换器	Analog-to-Digital Converter	ADC
数字模拟转换器	Digital-to-Analog Converter	DAC
比较器	Comparator	COMP
循环冗余校验计算单元	Cyclic Redundancy Check Calculation Unit	CRC

2 CPU 及系统架构

2.1 术语全称、缩写描述

表格 3 术语全称、缩写描述

中文全称	英文全称	英文缩写
可编程静态存储子系统	Configurable Static Memory Subsystem	CFGSMS
软件	Software.	SW
硬件	Hardware.	HW
接口	Interface	IF
AHB 从接口	AHB slave interface.	ahbs_if
面向硬件，预合成用户定义	Hardware-oriented, pre-synthesis user-defined.	-
存储队列	Store queue.	STQ
内嵌自测试	Built-in self test	BIST
所有从机数量	The number of all slaves (include ITCM/DTCM/AHB). The default value is 7.	IF_NUM
密钥位宽	The width of tha key. The default value is 8.	KEY_DW
低功耗计数器位宽	The width of the low power counter. The default value is 8.	CNT_WID
ITCM 从机数量	The number of the ITCM. The default value is 2.	ITCM_NUM
DTCM 从机数量	The number of the DTCM. The default value is 2.	DTCM_NUM
逻辑块	Logic Bank	BANK

2.2 简介

R501 SoC 是一款基于 Cortex-M52 双核心的系统级芯片（SoC）。该 SoC 采用了广泛应用的 AMBA 2.0 总线来集成外设 IP 组件：需要大带宽的组件连接到 AMBA AHB 总线上，而低速组件连接到 AMBA APB 总线上。DMA 不能访问与 FLB 相关的寄存器。

2.3 主要特征

计算部分

（1） CPU0

- Cortex-M52
- 缓存：支持 4KB 缓存

- TCM 接口：支持 36 位宽 16KB（默认大小）的 ITCM，36 位宽 8KB（默认大小）的 DTCM0 和 36 位宽 8KB（默认大小）的 DTCM1，均可通过 CFGSMS 进行配置。支持奇偶校验功能。
- 支持 FPU/CDE/COP

（2） CPU1

- Cortex-M52
- 缓存：支持 4KB 缓存
- TCM 接口：支持 36 位宽 16KB（默认大小）的 ITCM，36 位宽 8KB（默认大小）的 DTCM0 和 36 位宽 8KB（默认大小）的 DTCM1，均可通过 CFGSMS 进行配置。支持奇偶校验功能。
- FPU/CDE/COP

存储部分

（1） 片上 SRAM1

- 36 位宽，32KB（默认大小），可通过 CFGSMS 进行配置
- 支持奇偶校验

（2） 片上 SRAM2

- 36 位宽，16KB（默认大小），可通过 CFGSMS 进行配置
- 支持奇偶校验

（3） 片上 SRAM3

- 36 位宽，32KB（默认大小），可通过 CFGSMS 进行配置
- 支持奇偶校验

（4） 片上 FLASH

- 144 位宽
- 总容量为 640KB（512KB + 128KB）
- 支持单/双 Bank 切换

（5） 片上 Boot ROM

- 32 位宽，128KB

（6） 片上 Secure ROM

- 32 位宽，64KB

（7） 片上 CAN 专用 SRAM

- 1KB (CANA) + 1KB (CANB)

总线架构

（1） AHB 总线

- 32 位宽
- 7 个主机：CPU0-C-BUS，CPU0-S-BUS，CPU0-P-BUS，CPU1-C-BUS，CPU1-S-BUS，CPU1-P-BUS，DMA

- 15 个从机：CPU0-AHBT，CPU1-AHBT，SRAM1，SRAM2，SRAM3，FLASH1，FLASH2，Boot ROM，Secure ROM，DEMUX0，DEMUX1，DEMUX2，APB0，APB1，APB2。

（2） DEMUX0
- 4 slaves：GPIOCTRL，CPIODATA，INPUT_XBAR，XBAR

（3） DEMUX1
- 9 slaves：NVMC，CFGSMS，SYSC，DCS，QSPI，ANALOG SUBSYSTEM，WDT，NMIWDT，APB3

（4） DEMUX2
- 6 slaves：CPUTIM0~2，DMA，EXTI，APB4

（5） APB0
- 16 slaves：PWM1~8，CAP1~7，SDFM

（6） APB1
- 9 slaves：COMP1~7，QEP1，QEP2

（7） APB2
- 3 slaves：ADCA，ADCB，ADCC

（8） APB3
- 7 slaves：LIN，UARTA，SPIA，CANA，I2C，DACA，DACB

（9） APB4
- 9 slaves：UARTB，SPIB，CANB，PMBUS， DCCOPM，FLB1~4

时钟和复位

（1） 时钟
- 内部时钟源：INTOSC1 (10MHz)，INTOSC2 (10MHz)，PLL (250MHz)
- 外部时钟源：XTAL (2~20MHz)

（2） 复位源
- 片上上电复位(POR)
- 外部复位(XRS)
- 调试复位 (SYSRS)
- 看门狗复位 (WDRS)
- NMI 看门狗复位 (NMIWDRS)
- DCS Safe Code Copy Reset (SCCRESET)

物理层特征

（1） 供电电压：3.3V/1.1V

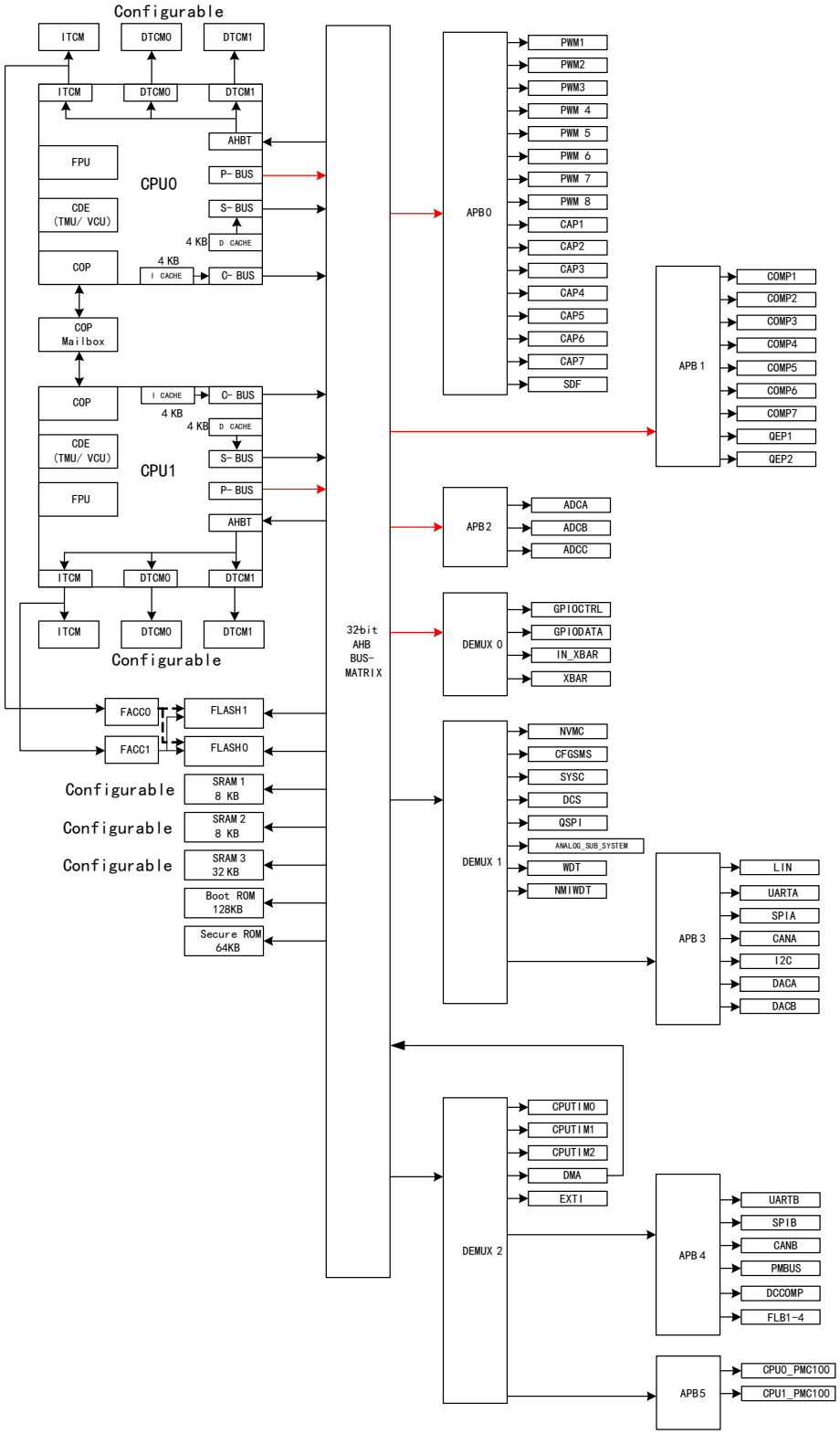
（2） 频率
- SYSCLK 频率最高为 250MHz

- 低速外设时钟频率最高为 125MHz

（3） 字节序：小端模式

2.4 结构框图

图 1 CPU 架构



注意：

（1） 图上红色线为总线桥。

（2） ITCM/DTCM/SRAM 大小通过 CFGSMS 进行配置。

2.5 功能描述

2.5.1 内存地址映射

表格 4 通用内存映射

地址范围	大小 (最大值)	描述
0x0000 0000-0x0001 FFFF	128KB	CPU0 ITCM（单核版本默认为 64KB，双核版本默认为 48KB）
0xA000 0000-0xA001 FFFF	128KB	CPU0 ITCM（CPU0-AHBT） （单核版本默认为 64KB，双核版本默认为 48KB）
0x0000 0000-0x0001 FFFF	128KB	CPU1 ITCM（单核版本默认为 0KB，双核版本默认为 8KB）
0xA100 0000-0xA101 FFFF	128KB	CPU1 ITCM（CPU1-AHBT） （单核版本默认为 0KB，双核版本默认为 8KB）
0x0008 0000-0x0008 BFFF	48KB	FLASH INFO on ITCM
0x0009 0000-0x0009 3FFF	16KB	FLASH INFO1 on ITCM
0x0010 0000-0x0019 FFFF	640KB	FLASH memory on ITCM
0x0800 0000-0x0809 FFFF	640KB	FLASH memory on BUSMATRIX
0x0810 0000-0x0810 BFFF	48KB	FLASH INFO on BUSMATRIX
0x0818 0000-0x0818 3FFF	16KB	FLASH INFO1 on BUSMATRIX
0x0900 0000-0x0901 FFFF	128KB	FLASH ECC
0x1000-0000-0x1001 FFFF	128KB	Boot ROM
0x1002 0000-0x1002 FFFF	64KB	Secure ROM
0x2000 0000-0x2001 FFFF	128KB	CPU0 DTCM（单核版本默认为 16KB，双核版本默认为 16KB）
0xA010 0000-0xA011 FFFF	128KB	CPU0 DTCM（CPU0-AHBT） （单核版本默认为 16KB，双核版本默认为 16KB）
0x2000 0000-0x2001 FFFF	128KB	CPU1 DTCM（单核版本默认为 0KB，双核版本默认为 8KB）
0xA110 0000-0xA111 FFFF	128KB	CPU1 DTCM（CPU1-AHBT） （单核版本默认为 0KB，双核版本默认为 8KB）
0x2010 0000-0x2011 FFFF	128KB	SRAM1（Default 8KB）
0x2020 0000-0x2021 FFFF	128KB	SRAM2（Default 8KB）
0x2030 0000-0x2031 FFFF	128KB	SRAM3（Default 32KB）

表格 5 通用总线地址映射

地址范围	总线	功能
0x4000 0000-0x4000 FFFF	APB0	APB 外设
0x4001 0000-0x4001 FFFF	APB1	APB 外设
0x4002 0000-0x4002 FFFF	APB2	APB 外设
0x4003 0000-0x4003 FFFF	DEMUX0	AHB 外设
0x5000 0000-0x5000 FFFF	APB3	APB 外设
0x5000 0000-0x5002 FFFF	DEMUX1	AHB 外设
0x6000 0000-0x6FFF FFFF		
0x5010 0000-0x5010 3FFF	APB4	APB 外设
0x5010 4000-0x5010 FFFF	APB5	APB 外设
0x5010 0000-0x5011 FFFF	DEMUX2	AHB 外设

2.5.2 外设地址映射

大多数片上外设的寄存器块寄存器地址的分配以 1KB 为单位。对于需要额外地址空间的模块被分配了额外的 1KB 块。在一个 1KB 的块内，外设寄存器可能未完全解码。每个外设的描述将定义对未解码寄存器的访问结果。在读取 32 位的寄存器中保留的位时返回零，并且对保留位的写入无影响。一般来说，为了确保与之后产品的兼容性，未实现的位应当返回零。

表格 6 外设地址映射

Bus	IP	Address
APB0	PWM1	0x4000 0000-0x4000 03FF
	PWM2	0x4000 0400-0x4000 07FF
	PWM3	0x4000 0800-0x4000 0BFF
	PWM4	0x4000 0C00-0x4000 0FFF
	PWM5	0x4000 1000-0x4000 13FF
	PWM6	0x4000 1400-0x4000 17FF
	PWM7	0x4000 1800-0x4000 1BFF
	PWM8	0x4000 1C00-0x4000 1FFF
	CAP1	0x4000 2000-0x4000 23FF
	CAP2	0x4000 2400-0x4000 27FF
	CAP3	0x4000 2800-0x4000 2BFF
	CAP4	0x4000 2C00-0x4000 2FFF
	CAP5	0x4000 3000-0x4000 33FF

Bus	IP	Address
	CAP6	0x4000 3400-0x4000 37FF
	CAP7	0x4000 3800-0x4000 3BFF
	SDF	0x4000 3C00-0x4000 3FFF
	Reserved	0x4000 4000-0x4000 FFFF
APB1	COMP1	0x4001 1C00-0x4001 1FFF
	COMP2	0x4001 2000-0x4001 23FF
	COMP3	0x4001 2400-0x4001 27FF
	COMP4	0x4001 2800-0x4001 2BFF
	COMP5	0x4001 2C00-0x4001 2FFF
	COMP6	0x4001 3000-0x4001 33FF
	COMP7	0x4001 3400-0x4001 37FF
	QEP1	0x4001 3800-0x4001 3BFF
	QEP2	0x4001 3C00-0x4001 3FFF
	Reserved	0x4001 4000-0x4001 FFFF
APB2	ADCA	0x4002 0000-0x4002 03FF
	ADCB	0x4002 0400-0x4002 07FF
	ADCC	0x4002 0800-0x4002 0BFF
	Reserved	0x4002 0C00-0x4002 FFFF
DEMUX0	GPIOCTRL	0x4003 0000-0x4003 07FF
	GPIODATA	0x4003 0800-0x4003 0BFF
	INPUTXBAR SyncSocREG	0x4003 0C00-0x4003 0C7F 0x4003 0C80-0x4003 0FFF
	XBAR_REG PWM_XBAR_REG FLB_XBAR_REG OUTPUT_XBAR_REG	0x4003 1000-0x4003 103F 0x4003 11C0-0x4003 123F 0x4003 1240-0x4003 12BF 0x4003 12C0-0x4003 133F
	Reserved	0x4003 1400-0x4003 FFFF
APB3	LIN	0x5000 0000-0x5000 03FF
	Reserved	0x5000 0400-0x5000 0BFF
	UARTA	0x5000 0C00-0x5000 0FFF
	SPIA	0x5000 1000-0x5000 13FF
	I2C	0x5000 1400-0x5000 17FF

Bus	IP	Address
	DACA	0x5000 1800-0x5000 1BFF
	DACB	0x5000 1C00-0x5000 1FFF
	CANA	0x5000 2000-0x5000 27FF
	Reserved	0x5000 2800-0x5000 FFFF
DEMUX1	APB3	0x5000 0000-0x5000 FFFF
	NVMC	0x5001 0000-0x5001 07FF
	CFGSMS	0x5001 0800-0x5001 0BFF
	Reserved	0x5001 0C00-0x5001 FFFF
	SYSC	0x5002 0000-0x5002 3FFF
	DCS	0x5002 4000-0x5002 5FFF
	QSPI	0x5002 6000-0x5002 63FF
	WWDT NMIWDT	0x5002 6400-0x5002 64BF 0x5002 64C0-0x5002 67FF
	Reserved	0x5002 6800-0x5002 7FFF
	ANALOG_SUB_SYSTEM	0x5002 8000-0x5002 83FF
	Reserved	0x5002 8400-0x5002 FFFF
	QSPI_MEMORY	0x6000 0000-0x6FFF FFFF
APB4	UARTB	0x5010 0000-0x5010 03FF
	SPIB	0x5010 0400-0x5010 07FF
	PMBUS	0x5010 0800-0x5010 0BFF
	Reserved	0x5010 0C00-0x5010 0FFF
	Reserved	0x5010 1000-0x5010 13FF
	DCCOMP	0x5010 1400-0x5010 17FF
	CANB	0x5010 1800-0x5010 1FFF
	FLB1 FLB2 FLB3 FLB4	0x5010 2000-0x5010 27FF 0x5010 2800-0x5010 2FFF 0x5010 3000-0x5010 37FF 0x5010 3800-0x5010 3FFF
APB5	CPU0_PMC100	0x5010 4000-0x5010 4FFF
	CPU1_PMC100	0x5010 5000-0x5010 5FFF
	Reserved	0x5010 6000-0x5010 FFFF
DEMUX2	APB4	0x5010 0000-0x5010 3FFF

Bus	IP	Address
	APB5	0x5010 4000-0x5010 FFFF
	CPUTIM0	0x5011 0000-0x5011 03FF
	CPUTIM1	0x5011 0400-0x5011 07FF
	CPUTIM2	0x5011 0800-0x5011 0BFF
	DMA	0x5011 0C00-0x5011 0FFF
	EXTI	0x5011 1000-0x5011 13FF
	Reserved	0x5011 1400-0x5011 FFFF

2.5.3 总线矩阵

表格 7 总线矩阵互联

总线从机	总线主机						
	CPU0 C-BUS	CPU0 S-BUS	CPU0 P-BUS	CPU1 C-BUS	CPU1 S-BUS	CPU1 P-BUS	DMA
CPU0-AHBT	×	×	×	×	√	×	√
CPU1-AHBT	×	√	×	×	×	×	√
FLASH BANK0	√	×	×	√	×	×	√
FLASH BANK1	√	×	×	√	×	×	√
Boot ROM	√	×	×	√	×	×	√
Secure ROM	√	×	×	√	×	×	√
SRAM1	×	√	×	×	√	×	√
SRAM2	×	√	×	×	√	×	√
SRAM3	×	√	×	×	√	×	√
DEMUX0	×	×	√	×	×	√	√
DEMUX1	×	√	×	×	√	×	√
DEMUX2	×	√	×	×	√	×	√
APB0	×	×	√	×	×	√	√
APB1	×	×	√	×	×	√	√
APB2	×	×	√	×	×	√	√

2.5.4 中断

表格 8 中断向量表

Vector ID	名称
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	DCCOMP interrupt
12	Reserved
13	TMR1 Interrupt
14	TMR2 Interrupt
15	Reserved
16	TAOSMS Interrupt
17	cop_te_irq_to_cpu0[0]
18	cop_te_irq_to_cpu0[1]
19	cop_te_irq_to_cpu0[2]
20	cop_te_irq_to_cpu0[3]
21	cop_rf_irq_to_cpu0[0]
22	cop_rf_irq_to_cpu0[1]
23	cop_rf_irq_to_cpu0[2]
24	cop_rf_irq_to_cpu0[3]
25	cop_gp_irq_to_cpu0[0]
26	cop_gp_irq_to_cpu0[1]
27	cop_gp_irq_to_cpu0[2]
28	cop_gp_irq_to_cpu0[3]

Vector ID	名称
29	Reserved
30	Reserved
31	Reserved
32	ADCA1 interrupt
33	ADCB1 interrupt
34	ADCC1 interrupt
35	Reserved
36	Reserved
37	Reserved
38	TMR0 interrupt
39	WAKEUP interrupt
40	PWM1 trip zone interrupt
41	PWM2 trip zone interrupt
42	PWM3 trip zone interrupt
43	PWM4 trip zone interrupt
44	PWM5 trip zone interrupt
45	PWM6 trip zone interrupt
46	PWM7 trip zone interrupt
47	PWM8 trip zone interrupt
48	PWM1 interrupt
49	PWM2 interrupt
50	PWM3 interrupt
51	PWM4 interrupt
52	PWM5 interrupt
53	PWM6 interrupt
54	PWM7 interrupt
55	PWM8 interrupt
56	CAP1 interrupt
57	CAP2 interrupt
58	CAP3 interrupt
59	CAP4 interrupt

Vector ID	名称
60	CAP5 interrupt
61	CAP6 interrupt
62	CAP7 interrupt
63	Reserved
64	QEP1 interrupt
65	QEP2 interrupt
66	Reserved
67	Reserved
68	FLB1 interrupt
69	FLB2 interrupt
70	FLB3 interrupt
71	FLB4 interrupt
72	SPIA_RX interrupt
73	SPIA_TX interrupt
74	SPIB_RX interrupt
75	SPIB_TX interrupt
76	Reserved
77	Reserved
78	Reserved
79	Reserved
80	DMA_CH1 interrupt
81	DMA_CH2 interrupt
82	DMA_CH3 interrupt
83	DMA_CH4 interrupt
84	DMA_CH5 interrupt
85	DMA_CH6 interrupt
86	Reserved
87	Reserved
88	I2CA interrupt
89	I2CA FIFO interrupt
90	QSPI interrupt

Vector ID	名称
91	Reserved
92	Reserved
93	Reserved
94	Reserved
95	Reserved
96	UARTA RX interrupt
97	UARTA TX interrupt
98	UARTB RX interrupt
99	UARTB TX interrupt
100	CANA interrupt 0
101	CANA interrupt 1
102	CANB interrupt 0
103	CANB interrupt 1
104	ADCA event interrupt
105	ADCA2 interrupt
106	ADCA3 interrupt
107	ADCA4 interrupt
108	ADCB event interrupt
109	ADCB2 interrupt
110	ADCB3 interrupt
111	ADCB4 interrupt
112	exti_line0 interrupt
113	exti_line1 interrupt
114	exti_line2 interrupt
115	exti_line3 interrupt
116	exti_line4 interrupt
117	exti_line5 interrupt
118	exti_line6 interrupt
119	exti_line7 interrupt
120	exti_line8 interrupt
121	exti_line9 interrupt

Vector ID	名称
122	exti_line10 interrupt
123	exti_line11 interrupt
124	exti_line12 interrupt
125	exti_line13 interrupt
126	exti_line14 interrupt
127	exti_line15 interrupt
128	Reserved
129	FPU DZC
130	FPU IDC
131	FPU IOC
132	FPU OFC
133	FPU UFC
134	FPU IXC
135	Reserved
136	Reserved
137	Reserved
138	Reserved
139	Reserved
140	Reserved
141	Reserved
142	Reserved
143	Reserved
144	Reserved
145	Reserved
146	Reserved
147	Reserved
148	Reserved
149	Reserved
150	Reserved
151	Reserved
152	Reserved

Vector ID	名称
153	Reserved
154	Reserved
155	Reserved
156	Reserved
157	CAP6 HRcalibration interrupt
158	CAP7 HRcalibration interrupt
159	Reserved
160	SDFM1 interrupt
161	Reserved
162	Reserved
163	Reserved
164	SDFM1 DR interrupt 1
165	SDFM1 DR interrupt 2
166	SDFM1 DR interrupt 3
167	SDFM1 Dr interrupt 4
168	Reserved
169	Reserved
170	Reserved
171	Reserved
172	Reserved
173	Reserved
174	Reserved
175	Reserved
176	Reserved
177	Reserved
178	Reserved
179	Reserved
180	Reserved
181	Reserved
182	Reserved
183	Reserved

Vector ID	名称
184	LINA interrupt 0
185	LINA interrupt 1
186	Reserved
187	Reserved
188	PMBUSA interrupt
189	Reserved
190	Reserved
191	Reserved
192	Reserved
193	Reserved
194	Reserved
195	Reserved
196	Reserved
197	Reserved
198	Reserved
199	Reserved
200	ADCC event interrupt
201	ADCC2 interrupt
202	ADCC3 interrupt
203	ADCC4 interrupt
204	Reserved
205	Reserved
206	Reserved
207	Reserved
208	Reserved
209	Reserved
210	Reserved
211	Reserved
212	Reserved
213	Reserved
214	Reserved

Vector ID	名称
215	Reserved
216	Reserved
217	Reserved
218	FLASH_ECC_ERR interrupt
219	Reserved
220	SYS_PLL_SLIP interrupt
221	CPU0_pmctf
222	CPU0_pmcte
223	CPU1_pmctf
224	CPU1_pmcte

2.6 寄存器组地址

表格 9 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
CpuSysRegs	CPU_SYS_REGS	0x5002_0A00	0x5002_0BFF

2.7 寄存器地址映射

表格 10 CPU_SYS_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
CPUSYSLOCK1	锁定 CPUSYS 寄存器	0x00	√
PCLKCR0	外设时钟门控寄存器 0	0x44	√
PCLKCR1	外设时钟门控寄存器 1	0x48	√
PCLKCR2	外设时钟门控寄存器 2	0x4C	√
PCLKCR3	外设时钟门控寄存器 3	0x50	√
PCLKCR4	外设时钟门控寄存器 4	0x54	√
PCLKCR6	外设时钟门控寄存器 6	0x5C	√
PCLKCR7	外设时钟门控寄存器 7	0x60	√
PCLKCR8	外设时钟门控寄存器 8	0x64	√
PCLKCR9	外设时钟门控寄存器 9	0x68	√
PCLKCR10	外设时钟门控寄存器 10	0x6C	√
PCLKCR13	外设时钟门控寄存器 13	0x78	√

寄存器名称	寄存器描述	偏移地址	WRPRT
PCLKCR14	外设时钟门控寄存器 14	0x7C	√
PCLKCR16	外设时钟门控寄存器 16	0x84	√
PCLKCR17	外设时钟门控寄存器 17	0x88	√
保留	保留	0x8C	√
PCLKCR19	外设时钟门控寄存器 19	0x90	√
PCLKCR20	外设时钟门控寄存器 20	0x94	√
PCLKCR21	外设时钟门控寄存器 21	0x98	√
LPMCR	LPM 控制寄存器	0xEC	√
GPIOLPMSEL0	GPIO LPM 唤醒选择寄存器 0	0xF0	√
GPIOLPMSEL1	GPIO LPM 唤醒选择寄存器 1	0xF4	√
TMR2CLKCTL	TMR2 时钟测量功能控制寄存器	0xF8	√
RESCCLR	复位原因清除寄存器	0xFC	-
RESC	复位原因寄存器	0x100	-

2.8 寄存器功能描述

2.8.1 锁定 CPUSYS 寄存器（CPUSYSLOCK1）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	保留			0h
x+3	PCLKCRx	R/WSonce	PCLKCRx 寄存器的锁定位（Lock bit for PCLKCRx Register）（x=0-4） 0：未锁定 1：锁定	0h
8	保留			0h
x+3	PCLKCRx	R/WSonce	PCLKCRx 寄存器的锁定位（Lock bit for PCLKCRx Register）（x=6-10） 0：未锁定 1：锁定	0h
15:14	保留			0h
x+3	PCLKCRx	R/WSonce	PCLKCRx 寄存器的锁定位（Lock bit for PCLKCRx Register）（x=13-16） 0：未锁定 1：锁定	0h
20	保留			0h

位/域	名称	R/W	描述	复位值
21	LPMCR	R/WSonce	LPMCR 寄存器的锁定位（Lock bit for LPMCR Register） 0：未锁定 1：锁定	0h
x+22	GPIOLPMSELx	R/WSonce	GPIOLPMSELx 寄存器的锁定位（Lock bit for GPIOLPMSELx Register）（x=0-1） 0：未锁定 1：锁定	0h
x+3	PCLKCRx	R/WSonce	PCLKCRx 寄存器的锁定位（Lock bit for PCLKCRx Register）（x=17-21） 0：未锁定 1：锁定	0h
31:29	保留			0h

2.8.2 外设时钟门控寄存器 0（PCLKCR0）

偏移地址：0x44

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	保留			0h
2	DMA	R/W	使能 DMA 时钟（DMA Clock Enable） 0：禁止 1：使能	0h
x+3	CPUTIMERx	R/W	使能 CPUTIMER 时钟（CPUTIMER Clock Enable） 0：禁止 1：使能	0h
15:6	保留			0h
16	HRPWM	R/W	使能 HRPWM 时钟（HRPWM Clock Enable） 设置时使时钟到 HRPWM 模块。 0：禁止 1：使能	0h
17	保留			0h
18	TBCLKSYNC	R/W	使能 PWM 时基时钟同步（PWM Time Base Clock sync Enable） 设置后将所有使能的 PWM 模块同步到基准时钟(TBCLK)。 0：禁止 1：使能	0h
31:19	保留			0h

2.8.3 外设时钟门控寄存器 1（PCLKCR1）

偏移地址：0x48

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	QSPI	R/W	使能 QSPI 时钟（QSPI Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.4 外设时钟门控寄存器 2（PCLKCR2）

偏移地址：0x4C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	PWMx	R/W	使能 PWMx 时钟（PWMx Clock Enable）（x=1-8） 0：禁止 1：使能	0h
31:8	保留			0h

2.8.5 外设时钟门控寄存器 3（PCLKCR3）

偏移地址：0x50

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	CAPx	R/W	使能 CAPx 时钟（CAPx Clock Enable）（x=1-7） 0：禁止 1：使能	0h
31:7	保留			0h

2.8.6 外设时钟门控寄存器 4（PCLKCR4）

偏移地址：0x54

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	QEPx	R/W	使能 QEPx 时钟（QEPx Clock Enable）（x=1-2） 0：禁止 1：使能	0h
31:2	保留			0h

2.8.7 外设时钟门控寄存器 6（PCLKCR6）

偏移地址：0x5C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SD1	R/W	使能 SD1 时钟（SD1 Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.8 外设时钟门控寄存器 7（PCLKCR7）

偏移地址：0x60

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	UART_A	R/W	使能 UARTA 时钟（UARTA Clock Enable） 0：禁止 1：使能	0h
1	UART_B	R/W	使能 UARTB 时钟（UARTB Clock Enable） 0：禁止 1：使能	0h
31:2	保留			0h

2.8.9 外设时钟门控寄存器 8（PCLKCR8）

偏移地址：0x64

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SPI_A	R/W	使能 SPIA 时钟（SPIA Clock Enable） 0：禁止 1：使能	0h
1	SPI_B	R/W	使能 SPIB 时钟（SPIB Clock Enable） 0：禁止 1：使能	0h
31:2	保留			0h

2.8.10 外设时钟门控寄存器 9（PCLKCR9）

偏移地址：0x68

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	I2C_A	R/W	使能 I2CA 时钟（I2CA Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.11 外设时钟门控寄存器 10（PCLKCR10）

偏移地址：0x6C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CAN_A	R/W	使能 CANA 时钟（CANA Clock Enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
1	CAN_B	R/W	使能 CANB 时钟（CANB Clock Enable） 0：禁止 1：使能	0h
31:2	保留			0h

2.8.12 外设时钟门控寄存器 13（PCLKCR13）

偏移地址：0x78

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ADC_A	R/W	使能 ADCA 时钟（ADCA Clock Enable） 0：禁止 1：使能	0h
1	ADC_B	R/W	使能 ADCB 时钟（ADCB Clock Enable） 0：禁止 1：使能	0h
2	ADC_C	R/W	使能 ADCC 时钟（ADCC Clock Enable） 0：禁止 1：使能	0h
31:3	保留			0h

2.8.13 外设时钟门控寄存器 14（PCLKCR14）

偏移地址：0x7C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	COMPx	R/W	使能 COMPx 时钟（COMPx Clock Enable）（x=1-7） 0：禁止 1：使能	0h
31:7	保留			0h

2.8.14 外设时钟门控寄存器 16（PCLKCR16）

偏移地址：0x84

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	保留			0h
16	DAC_A	R/W	使能 DACA 时钟（DACA Clock Enable） 0：禁止 1：使能	0h
17	DAC_B	R/W	使能 DACB 时钟（DACB Clock Enable） 0：禁止 1：使能	0h
31:18	保留			0h

2.8.15 外设时钟门控寄存器 17（PCLKCR17）

偏移地址：0x88

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	FLBx	R/W	使能 FLBx 时钟（FLBx Clock Enable）（x=1-4） 0：禁止 1：使能	0h
31:4	保留			0h

2.8.16 外设时钟门控寄存器 19（PCLKCR19）

偏移地址：0x90

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	LIN_A	R/W	使能 LINA 时钟（LINA Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.17 外设时钟门控寄存器 20（PCLKCR20）

偏移地址：0x94

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PMBUS_A	R/W	使能 PMBUSA 时钟（PMBUSA Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.18 外设时钟门控寄存器 21（PCLKCR21）

偏移地址：0x98

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DCCOMP_0	R/W	使能 DCCOMP0 时钟（DCCOMP0 Clock Enable） 0：禁止 1：使能	0h
31:1	保留			0h

2.8.19 LPM 控制寄存器（LPMCR）

偏移地址：0xEC

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	LPM	R/W	配置低功耗模式（low power mode configure） 当 CPU 执行 IDLE 指令时生效(IDLE 指令不在流水线的 EXE 阶段时) 00：空闲模式 01：保留 1x：停机模式	0h
7:2	保留			3Fh
31:8	保留			0h

2.8.20 GPIO LPM 唤醒选择寄存器 0（GPIOLPMSEL0）

偏移地址：0xF0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x	GPIOx	R/W	引脚连接 LPM 电路（pin is connected to LPM circuit）（x=0-31） 0：断开 1：连接	0h

2.8.21 GPIO LPM 唤醒选择寄存器 1（GPIOLPMSEL1）

偏移地址：0xF4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-32	GPIOx	R/W	引脚连接 LPM 电路（pin is connected to LPM circuit）（x=32-63） 0：断开 1：连接	0h

2.8.22 TMR2 时钟测量功能控制寄存器（TMR2CLKCTL）

偏移地址：0xF8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	TMR2CLKSRCSEL	R/W	选择 TMR2 时钟源（TMR2 Clock Source Select） 000：SYSCLK 选择(重置时默认值，绕过预分频) 001：INTOSC1 010：INTOSC2 011：XTAL 100：FLPUMPOSC 101：FOSCCLK 110：AUXPLLCLK（保留） 111：保留	0h

位/域	名称	R/W	描述	复位值
5:3	TMR2CLKPRESCALE	R/W	选择 TMR2 时钟预分频（TMR2 Clock Prescale Select） 000：/1：(复位时默认值) 001：/2 010：/4 011：/8 100：/16 其他：备用(默认值为/16) 注意：TMR 时钟同步逻辑在配置为除 SYSCLK 以外的任何时钟源时检测输入时钟边，并生成适当的时钟脉冲到 TMR2。如果 SYSCLK 与输入时钟源大致相同或更小，则需要配置预分频值，使 SYSCLK 的速度至少是预分频值的两倍。	0h
31:6	保留			0h

2.8.23 复位原因清除寄存器（RESCCLR）

偏移地址：0xFC

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	保留			0h
2	WDRSn	W1S	清除看门狗复位标志（WDRSn Flag Clear） 读取 RESCCLR 总是返回 0。在此位上写 1，将 RESC 中的状态位清除为 0。写 0 无效。	0h
3	NMIWDRSn	W1S	清除 NMI 看门狗复位标志（NMIWDRSn Flag Clear） 读取 RESCCLR 总是返回 0。在此位上写 1，将 RESC 中的状态位清除为 0。写 0 无效。	0h
7:4	保留			0h
8	SCCRESETn	W1S	清除 DCS 安全代码复制复位标志（SCCRESETn Flag Clear） 读取 RESCCLR 总是返回 0。在此位上写 1，将 RESC 中的状态位清除为 0。写 0 无效。	0h
31:9	保留			0h

2.8.24 复位原因寄存器（RESC）

偏移地址：0x100

复位类型：除[31:30]为 N/A，其余为 POR

位/域	名称	R/W	描述	复位值
1:0	保留			0h
2	WDRSn	R	看门狗复位标志（WDRSn Flag） 为了保持一致，WDT 模块内的位也提供相同的信息。因为该寄存器是软件了解 Cortex-M52 内核复位原因的一站式服务。 0：无作用 1：强制该位为 0	0h

位/域	名称	R/W	描述	复位值
3	NMIWDRSn	R	NMI 看门狗复位标志（NMIWDRSn Flag） NMI 复位的确切原因需要软件读取 CPU1.NMISHDFLG 寄存器。 0：无作用 1：强制该位为 0	0h
7:4	保留			0h
8	SCCRESETn	R	DCS 安全代码复制复位标志（SCCRESETn Flag） 设备被 SCCRESETn 复位(由 DCS 触发)。 0：无作用 1：强制该位为 0	0h
29:9	保留			0h
30	XRSn_pin_status	R	XRSn 引脚状态（XRSn pin status） 读取此位提供 XRSn 引脚的当前状态。复位值反映引脚状态。	0h
31	DCON	R	调试器连接到 Arm® Cortex®-M52 CPU 标志（debugger connection to the Arm® Cortex®-M52 CPU Flag） 读取此位提供调试器连接到 CPU 的状态。 0：未连接 1：连接 注意：该位连接到 Arm® Cortex®-M52 CPU 的 DCON o/p 信号。此位通过上电复位清零。	0h

3 数学指令拓展（Zidian）

Zidian 通过新增指令来加速某些特定的整数和浮点运算，从而扩展了 Arm 处理器的功能。这是通过使用 Arm 提供的 ACI 功能来实现的。本模块的描述请参见《Zidian 用户手册》 。

4 系统启动（BOOT）

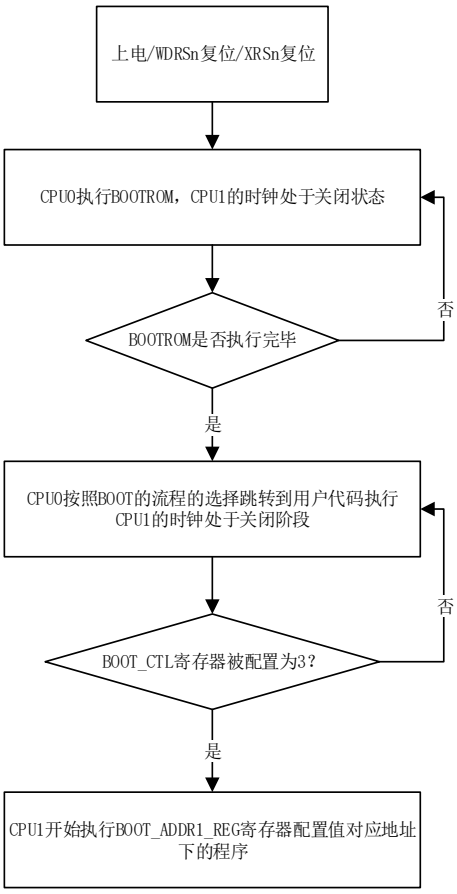
4.1 简介

主核 CPU0 需要从 BOOTROM 启动。CPU0 启动后配置 SPEC_BOOTCTRL 寄存器以开启 CPU1 的时钟才能使用次核 CPU1。在架构上 CPU0 与 CPU1 的结构完全对称，用户代码中可以不分主次。

4.2 功能描述

4.2.1 步骤

图 2 系统启动流程图



4.2.2 Boot Modes

Boot Modes 配置及选择引脚请参见《G32R501 数据手册》引导 ROM 和外设引导章节。

4.3 寄存器

BOOT 寄存器请参见系统控制寄存器章节。

5 写寄存器保护（WRPRT）

5.1 术语全称、缩写描述

表格 11 术语全称、缩写描述

中文全称	英文全称	英文缩写
协处理器 12	Co-Processor 12	CP12
中断服务程序	Interrupt Service Routine	ISR

5.2 简介

WRPRT 保护机制是为了防止 CPU 对系统中的一些寄存器进行错误的写操作，该机制使用特殊的协处理器指令（“WRPRT_DISABLE”和“WRPRT_ENABLE”）来失能和开启写保护。

5.3 主要特征

（1） 访问权限由 CP12 寄存器中的 WRPRT 位决定

（2） 执行“WRPRT_ENABLE”指令可以使能写保护（清除 WRPRT 位），不允许 CPU 写入受保护寄存器

（3） 执行“WRPRT_DISABLE”指令失能写保护（设置 WRPRT 位为 1），允许 CPU 写入受保护寄存器

（4） CPU0，CPU1 各自的 WRPRT 位是独立的

（5） ISR 开始时，WRPRT 位清零

- 当前 WRPRT 值会被保存在影子寄存器中

5.4 功能描述

5.4.1 WRPRT 保护机制

表格 12 对受 WRPRT 保护寄存器的访问

WRPRT 位	访问方式	DEBUG	DMA	CPU0/1
0	读取	允许	允许	允许
	写入	允许	忽略	忽略
1	读取	允许	允许	允许
	写入	允许	允许	允许

注意： JTAG/SWD 端口覆盖 WRPRT 位，允许在调试模式下完全访问受保护的寄存器。

G32R501 启动时，默认情况下启用寄存器保护功能：

- 受保护状态下，允许 CPU 读取、DEBUG 读取 /写入，忽略 CPU 对受保护寄存器的写操作
- 执行“WRPRT_ENABLE”指令可以禁用保护,CPU 允许自由写入受保护的寄存器
- 修改寄存器后，执行“WRPRT_DISABLE”指令可以清除 WRPRT 位（再次启用保护）

5.4.1.1 WRPRT 位描述

G32R501 中有两个内核：CPU0 和 CPU1。CPU0 与 CPU1 的 WRPRT 位是独立的（CPU0 的 WRPRT 位无法阻止 CPU1 访问受保护的寄存器，CPU1 的 WRPRT 位也无法阻止 CPU0 访问受保护的寄存器）。

5.4.2 ISR 中的 WRPRT 保护

表格 13 CPU0 /1 对受 WRPRT 保护寄存器的写操作

保护状态	CPU0 写操作	CPU1 写操作
CPU0 WRPRT_ENABLE	忽略	允许
CPU0 WRPRT_DISABLE	允许	允许
CPU1 WRPRT_ENABLE	允许	忽略
CPU1 WRPRT_DISABLE	允许	允许

当 CPU 服务中断时，WRPRT 当前值保存在影子寄存器中，WRPRT 清零。因此，在 ISR 开始时禁止访问对受保护寄存器。如果 ISR 必须访问受保护寄存器，它必须包含一个“WRPRT_DISABLE”指令。在 ISR 结束时，可以从影子寄存器中恢复 WRPRT。

5.4.3 使用说明

由于 WRPRT_ENABLE/WRPRT_DISABLE 指令通过 COP 接口实现，所以它们的生效时间可能与应用目的不同。此错误可以通过在寄存器访问和 WRPRT_ENABLE/WRPRT_DISABLE 指令之间添加 DSB/ISB 指令来避免，在写入受保护的寄存器后添加读检查解决。

5.5 寄存器组地址

表格 14 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
WrprtRegs	WRPRT_REGS	-	-

注意：寄存器只能通过协处理器指令访问，不能通过总线访问。

5.6 寄存器地址映射

表格 15 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
WRPRT_WRPRT	写保护寄存器	-	-

5.7 寄存器功能描述

5.7.1 写保护寄存器 0（WRPRT_WRPRT0）

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	WRPRT0	R/W	保护 CPU0 访问的寄存器（Protecting registers accessed by CPU0）	0h
31:1	保留			0h

5.7.2 写保护寄存器 1（WRPRT_WRPRT1）

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	WRPRT1	R/W	保护 CPU1 访问的寄存器（Protecting registers accessed by CPU1）	0h
31:1	保留			0h

6 UID

6.1 寄存器组地址

表格 16 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
UIDRegs	UID_REGS	0x0810_AF70	0x0810_AF8F

6.2 寄存器地址映射

表格 17 UID 寄存器地址映射

寄存器名称	描述	偏移地址	WRPRT
UID_PSRAND0	UID 的伪随机部分 0	0x00	-
UID_PSRAND1	UID 的伪随机部分 1	0x04	-
UID_PSRAND2	UID 的伪随机部分 2	0x08	-
UID_PSRAND3	UID 的伪随机部分 3	0x0C	-
UID_PSRAND4	UID 的伪随机部分 4	0x10	-
UID_PSRAND5	UID 的伪随机部分 5	0x14	-
UID_UNIQUE	UID 唯一 32 位编号寄存器	0x18	-
UID_CHECKSUM	UID_PSRANDx 和 UID_UNIQUE 的校验和寄存器	0x1C	-

6.3 寄存器功能描述

6.3.1 UID 的伪随机部分寄存器 x（UID_PSRANDx）

偏移地址：0x00+0x04*x（x=0-5）

复位类型：N/A

位/域	名称	R/W	描述	复位值
31:0	RandomID	R	UID 的伪随机部分（Psuedorandom portion of the UID） UID[191:128]：随机值 UID[127:40]：LOT ID，ASCLL 码（数据向高位对齐，低位若缺少数据则补零，十六进制存入） UID[39:32]：Wafer ID，8 位无符号数（十六进制存入） UID[31:16]：X 坐标，BCD 码（十进制存入） UID[15:0]：Y 坐标，BCD 码（十进制存入）	0h

6.3.2 UID 唯一 32 位编号寄存器（UID_UNIQUE）

偏移地址：0x18

复位类型：N/A

位/域	名称	R/W	描述	复位值
31:0	UniqueID	R	UID 的唯一部分（Unique portion of the UID） 该标识符在具有相同 PARTIDH 的所有设备中是唯一的。	0h

6.3.3 UID_PSRANDx 和 UID_UNIQUE 的校验和寄存器（UID_CHECKSUM）

偏移地址：0x1C

复位类型：N/A

位/域	名称	R/W	描述	复位值
31:0	Checksum	R	UID_PSRANDx 和 UID_UNIQUE 的校验和（Fletcher checksum of UID_PSRANDx and UID_UINIQUE） UID CRC 校验值，由 UID_PSRAND0-5CRC 校验得到。	0h

7 电源管理

芯片上电后默认由 LDO 供电。LDO 支持调压功能，供电电压为 1.1V~0.9V。

表格 18 电源名称及描述

电源名称	描述
VDD	1.1V 数字逻辑电源引脚。建议在每个 VDD 引脚附近放置一个最小总电容约为 20μF 的去耦电容。当不使用内部稳压器时，去耦电容的值由系统稳压解决方案确定。
VDDA	3.3V 模拟电源引脚。设定每个引脚上的 VSSA 至少 2.2μF 的去耦电容。
VDDIO	3.3V 数字 I/O 电源引脚。设定每个引脚上至少 0.1μF 的去耦电容。
VSS	数字地
VSSA	模拟地

8 设备标识和配置寄存器

8.1 简介

设备标识寄存器和配置寄存器提供有关部件编号，产品系列，修订，引脚数，资格状态和设备功能可用性的信息。所有设备信息都在 DEV_CFG_REGS 中。标识寄存器是 PARTIDL，PARTIDH 和 REVID。

8.2 寄存器组地址

表格 19 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
DevRegs	DEV_REGS	0x5002_0500	0x5002_07FF

8.3 寄存器地址映射

表格 20 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
PARTIDL	设备部件标识号低寄存器	0x10	-
PARTIDH	设备部件标识号高寄存器	0x14	-
REVID	设备版本号寄存器	0x18	-
DC21	设备功能寄存器	0x74	-
SOFTPRES0	处理块软件复位寄存器 0	0x104	√
SOFTPRES2	处理块软件复位寄存器 2	0x10C	√
SOFTPRES3	处理块软件复位寄存器 3	0x110	√
SOFTPRES4	处理块软件复位寄存器 4	0x114	√
SOFTPRES6	处理块软件复位寄存器 6	0x11C	√
SOFTPRES7	处理块软件复位寄存器 7	0x120	√
SOFTPRES8	处理块软件复位寄存器 8	0x124	√
SOFTPRES9	处理块软件复位寄存器 9	0x128	√
SOFTPRES10	处理块软件复位寄存器 10	0x12C	√
SOFTPRES11	处理块软件复位寄存器 11	0x130	√
SOFTPRES13	处理块软件复位寄存器 13	0x138	√
SOFTPRES14	处理块软件复位寄存器 14	0x13C	√

寄存器名称	寄存器描述	偏移地址	WRPRT
SOFTPRES16	处理块软件复位寄存器 16	0x144	√
SOFTPRES17	处理块软件复位寄存器 17	0x148	√
SOFTPRES19	处理块软件复位寄存器 19	0x150	√
SOFTPRES20	处理块软件复位寄存器 20	0x154	√
PACKSEL	选择芯片封装寄存器	0x25C	√

8.4 寄存器功能描述

8.4.1 设备部件标识号低寄存器(PARTIDL)

偏移地址：0x10

复位类型：XRSn

位/域	名称	R/W	描述	复位值
5:0	保留			0h
7:6	QUAL	R	选择标准（criteria Select） 00：工程样品(TMX) 01：中试产品(TMP) 10：完全合格(TMS)	0h
10:8	PIN_COUNT	R	选择引脚数量（Pin Number Select） 000：56 个引脚 001：64 个引脚 010：80 个引脚 011-100：保留 101：100 个引脚 110-111：保留	X
12:11	保留			0h
14:13	保留			0h
15	保留			0h
23:16	FLASH_SIZE	R	选择存储大小（FLASH Size Select） 00000000-00000011：保留 00000100：256KB 00000101：640KB 00000110-11111111：保留	8h
31:24	保留			0h

8.4.2 设备部件标识号高寄存器(PARTIDH)

偏移地址：0x14

复位类型：XRSn

位/域	名称	R/W	描述	复位值
7:0	保留			0h
15:8	FAMILY	R	设备系列（Device series） 00000000-00000011：保留 00000100：单核 00000101-11111111：保留	X
23:16	PARTNO	R	设备部件号（Device Part Number） 可参考数据表	X
31:24	DEVICE_CLASS_ID	R	设备类别 ID（Device class ID）	X

8.4.3 设备版本号寄存器(REVID)

偏移地址：0x18

复位类型：XRSn

位/域	名称	R/W	描述	复位值
15:0	REVID	R	设备修改（Device Revision） OPT_KEY 为 0x5A 时加载。	0h
31:16	保留			0h

8.4.4 设备功能(DC21)

偏移地址：0x74

复位类型：XRSn

位/域	名称	R/W	描述	复位值
x-1	FLBx	R	FLBx 功能（FLBx function）（x=1-4） 0：设备不存在该特性 1：设备存在该特性	0h
31:4	保留			0h

8.4.5 处理块软件复位寄存器 0(SOFTPRES0)

偏移地址：0x104

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CPU1	R	配置 CPU1 复位（CPU1 reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.6 处理块软件复位寄存器 2(SOFTPRES2)

偏移地址：0x10C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	PWMx	R/W	配置 PWMx 复位（PWMx reset Configuration）（x=1-8） 0：复位由正常的设备复位结构决定 1：复位	0h
31:8	保留			0h

8.4.7 处理块软件复位寄存器 3(SOFTPRES3)

偏移地址：0x110

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	CAPx	R/W	配置 CAPx 复位（CAPx reset Configuration）（x=1-7） 0：复位由正常的设备复位结构决定 1：复位	0h
31:7	保留			0h

8.4.8 处理块软件复位寄存器 4(SOFTPRES4)

偏移地址：0x114

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	QEPx	R/W	配置 QEPx 复位（QEPx reset Configuration）（x=1-2） 0：复位由正常的设备复位结构决定 1：复位	0h
31:2	保留			0h

8.4.9 处理块软件复位寄存器 6(SOFTPRES6)

偏移地址：0x11C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SD1	R/W	配置 SD1 复位（SD1 reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.10 处理块软件复位寄存器 7(SOFTPRES7)

偏移地址：0x120

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	UART_A	R/W	配置 UARTA 复位（UARTA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h

位/域	名称	R/W	描述	复位值
1	UART_B	R/W	配置 UARTB 复位（UARTB reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:2	保留			0h

8.4.11 处理块软件复位寄存器 8(SOFTPRES8)

偏移地址：0x124

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SPI_A	R/W	配置 SPIA 复位（SPIA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
1	SPI_B	R/W	配置 SPIB 复位（SPIB reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:2	保留			0h

8.4.12 处理块软件复位寄存器 9(SOFTPRES9)

偏移地址：0x128

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	I2C_A	R/W	配置 I2CA 复位（I2CA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.13 处理块软件复位寄存器 10(SOFTPRES10)

偏移地址：0x12C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CAN_A	R/W	配置 CANA 复位（CANA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
1	CAN_B	R/W	配置 CANB 复位（CANB reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:2	保留			0h

8.4.14 处理块软件复位寄存器 11(SOFTPRES11)

偏移地址：0x130

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	QSPI	R/W	配置 QSPI 复位（QSPI reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.15 处理块软件复位寄存器 13(SOFTPRES13)

偏移地址：0x138

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ADC_A	R/W	配置 ADCA 复位（ADCA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
1	ADC_B	R/W	配置 ADCB 复位（ADCB reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
2	ADC_C	R/W	配置 ADCC 复位（ADCC reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:3	保留			0h

8.4.16 处理块软件复位寄存器 14(SOFTPRES14)

偏移地址：0x13C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	COMPx	R/W	配置 COMPx 复位（COMPx reset Configuration）（x=1-7） 0：复位由正常的设备复位结构决定 1：复位	0h
31:7	保留			0h

8.4.17 处理块软件复位寄存器 16(SOFTPRES16)

偏移地址：0x144

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	保留			0h
16	DACA	R/W	配置 DACA 复位（DACA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
17	DACB	R/W	配置 DACB 复位（DACB reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h

位/域	名称	R/W	描述	复位值
31:18	保留			0h

8.4.18 处理块软件复位寄存器 17(SOFTPRES17)

偏移地址：0x148

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	FLBx	R/W	配置 FLBx 复位（FLBx reset Configuration）（x=1-4） 0：复位由正常的设备复位结构决定 1：复位	0h
31:4	保留			0h

8.4.19 处理块软件复位寄存器 19(SOFTPRES19)

偏移地址：0x150

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	LIN_A	R/W	配置 LINA 复位（LINA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.20 处理块软件复位寄存器 20(SOFTPRES20)

偏移地址：0x154

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PMBUS_A	R/W	配置 PMBUSA 复位（PMBUSA reset Configuration） 0：复位由正常的设备复位结构决定 1：复位	0h
31:1	保留			0h

8.4.21 选择芯片封装寄存器(PACKSEL)

偏移地址：0x25C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	PACKSEL	R/W	选择芯片封装（Chip Package Select） 0000：100pin 0001：80pin 0010：64pin 0011：56pin 0100：48pin 0101-1111：保留	0h

位/域	名称	R/W	描述	复位值
31:4	保留			0h

9 低功耗模式

G32R501 支持 IDLE 和 HALT 两种低功耗模式。两种低功耗模式都是通过配置 LPMCR 寄存器并执行 WFI 或 WFE 指令进入。此外，根据低功耗模式的不同，需要相应的配置 CPU SCB 寄存器的 SLEEPDEEP 位。低功耗模式不能在 FLASH 当前正在执行擦、写操作时进入。进入 HALT 模式会停掉所有的 CPU 和外设活动。包括当前正在进行的通信传输和控制算法。

在准备进入 HALT 模式之前，应用程序需要先确保系统已经做好进入低功耗模式的准备。进入 HALT 模式前，还需要检查 GPIODAT 寄存器的值，确保通过 GPIOLPMSEL0/1 寄存器所选中的 HALT 唤醒引脚当前处于非有效状态，以便系统能够顺利进入 HALT 模式。

9.1 IDLE

IDLE 模式下，CPU 会进入 SLEEPING 模式。此时 CPU 的时钟关闭，但是系统时钟以及外设时钟仍然保留。因此，IDLE 模式可用于在 CPU 等待外设事件或中断的过程中节省功耗。根据进入 IDLE 时执行的指令不同，唤醒方式不同。

进入 IDLE 模式的方式：

- 设置 LPMCR.LPM 为 0x0
- 执行 WFI/WFE 指令

注意：根据 SLEEPONEXIT 位的配置不同，有两种进入 IDLE 模式的选择。

（1） 如果 SLEEPONEXIT 位被清除，MCU 在执行 WFI 或 WFE 指令时立即进入 IDLE 模式。

（2） 如果设置 SLEEPONEXIT 位，MCU 在退出最低优先级 ISR 后立即进入 IDLE 模式。

9.2 HALT

HALT 模式是一个全局低功耗模式，能够关闭绝大部分系统时钟，并且允许关闭部分模拟时钟源和模拟 IP。进入 HALT 模式不会自动将 XTAL 掉电，并且从 HALT 模式唤醒不会自动为 XTAL 上电。可以通过配置 XTALCR.OSCOFF 位，以软件实现的方式将 XTAL 掉电。

对于需要在 HALT 模式下维持最小功耗的应用程序，软件需要在进入 HALT 模式前先关闭 XTAL。需要注意的是，如果当前系统时钟源为 XTAL，需要先将系统时钟源切换为 INTOSC1 或 INTOSC2。

GPIO0~63 可以被配置为 HALT 模式的唤醒引脚。然而，看门狗在 HALT 模式依然可以被配置为工作状态。并且可以被配置为在超时事件发生时产生 watchdog 复位。

进入 HALT 模式的方式：

（1） 在 EXTI 中使能 WAKEINT。

（2） 设置 LPMCR.LPM 为 0x2，GPIOLPMSEL0 和 GPIOLPMSEL1，将选中的 GPIO 连接到 LPM 模块。

（3） 设置 CPU SCB.SLEEPDEEP。

（4） 设置 CLKSRCCTL1.WDHALTI，使能看门狗定时器，使 INTOSC1 和 INTOSC2 在 HALT 中上电。

（5） 复位 CLKSRCCTL1.WDHALTI，关闭看门狗定时器，使 INTOSC1 和 INTOSC2 在 HALT 中下电。

（6） 执行 WFI/WFE 指令进入 HALT。

注意：如果当前 FLASH memory 正在进行写操作，HALT 模式的进入将会延迟到 FLASH memory 的写操作完成之后。此外，如果处理器正在访问 APB 上面的外设，HALT 模式的进入将会延迟到 APB 外设访问的完成之后。

表格 21 低功耗模式

模式名称	进入	唤醒	对时钟的影响
IDLE	LPMCR 寄存器的 LPM 位为 2'b00 + WFI（当系统处于双核工作状态时，只有执行 WFI 指令的 CPU 进入 IDLE 模式）	任意中断	CPU CLK 关闭，对其他时钟或模拟时钟源无影响
	LPMCR 寄存器的 LPM 位为 2'b00 + WFE（当系统处于双核工作状态时，只有执行 WFE 指令的 CPU 进入 IDLE 模式）	唤醒事件	
HALT	LPMCR 寄存器的 LPM 位为 2'b10 + CPU 的 SCB->SLEEPDEEP 位置 1 + WFE/WFI（需要注意，当系统处于双核工作状态时，进入 HALT 模式需要 CPU0 和 CPU1 均将 SCB->SLEEPDEEP 位置 1，并且均执行 WFE/WFI 指令。单核或双核工作状态通过 Boot 模式控制寄存器控制）	任意中断	关闭除模拟时钟源之外的绝大部分系统时钟，包括 CPU CLK 和外设时钟等。根据功耗需求，允许进一步关闭 XTAL、PLL、INTOSC1、INTOSC2 等模拟时钟源。

表格 22 低功耗模式时钟状态

时钟域/模块	IDLE	HALT
SYSCLK	活跃	门控
CPU0_CLK	门控	门控
CPU1_CLK	门控	门控
APBCLK	活跃	门控
WDTCLK	活跃	如果 CLKSRCTL1.WDHALTI = 0 则门控
PLL	上电	软件必须在进入 HALT 之前将锁相环断电。
INTOSC1	上电	如果 CLKSRCTL1.WDHALTI = 0 则下电
INTOSC2	上电	如果 CLKSRCTL1.WDHALTI = 0 则下电

时钟域/模块	IDLE	HALT
XTAL(1)	上电	上电
FLASH(2)	上电	上电

注意：

（1） 在任何 LPM 下，XTAL 不会由硬件断电。它可以通过软件将 XTALCR.OSCOFF 位设置为 1 来断电。如果不需要 XTAL，可以在应用程序中的任何时间完成此操作。

（2） 在任何 LPM 下，Flash 模块不会由硬件断电。如果应用需要，可使用软件将其断电。

10 复位

表格 23 复位源

名称	CPU 内核复位	JTAG/调试逻辑复位	外设复位	IOs	XRSn 输出
POR	√	×	√	Hi-Z	√
WDRSn	√	×	√	Hi-Z	√
NMIWDRSn	√	×	√	Hi-Z	√
SCCRESETn	√	×	√	Hi-Z	×
SYSRSn（调试器复位）	√	×	√	Hi-Z	×
XRSn 引脚	√	×	√	Hi-Z	-

注意：表格内的“n”表示低电平有效。

复位可分为芯片级复位和系统复位，芯片级复位用于复位全部或几乎全部设备，系统复位用于复位设备的一个大子集，但保留一些系统级配置。复位后 RESC 寄存器更新，该寄存器中的位只能通过上电复位或由 RESCCLR 寄存器中的相关位来清除，个别位由引导 ROM 作为启动例程的一部分清除。

10.1 功能描述

10.1.1 上电复位（POR）

为了抑制 GPIO 上的故障，上电复位电路需要在上电期间建立一个干净的复位。XRSn 引脚在 POR 期间保持低电平。XRSn 一般保持较低的时间就足以复位其他系统 IC。如果需要更长的脉冲，XRSn 引脚可以从外部驱动为低电平来提供正确的复位持续时间。POR 可以复位 XRSn 的所有行动，JTAG 端口使用的调试逻辑及 RESC 寄存器，NMISHDFLG 寄存器和 X1CNT 寄存器。执行 POR 操作后，POR 位和 XRSn 位被设置。随后被引导 ROM 清除。

10.1.2 外部复位（XRSn）

外部复位是设备的主要芯片级复位。用于复位 CPU，所有外设，I/O 引脚配置及大部分系统控制寄存器。XRSn 可由外部源驱动，但在看门狗，NMI 和上电复位时由内部驱动。有一个可用于驱动应用程序中其他 IC 的复位引脚的开漏引脚。在 XRSn 被驱动为低时将设置 XRSn 位，然后被 boot ROM 清除。

10.1.3 调试器复位（SYSRSn）

有时需要在不断开调试器连接或中断系统级配置的情况下复位 CPU 和外设。可以使用 Keil 等 IDE 的调试器触发 CPU 的单独子系统复位。SYSRSn 可以复位 CPU，外设，I/O 引脚配置和许多系统控制寄存器。

10.1.4 看门狗复位（WDRSn）

如果看门狗定时器在指定的时间内没有被 CPU 服务，则可以选择性地触发复位。WDRSn 产生一个持续 512 个 INTOSC1 周期的 XRSn。看门狗复位后，RESC 中的 WDRSn 位被设置。如果 RESC 中 XRSn 和 WDRSn 位同时设置，则由看门狗发起复位。如果在 RESC 中只设置了 XRSn 位，则看门狗是从 XRSn 复位的。

10.1.5 NMI 看门狗复位（NMIWDRSn）

NMI 用于检测系统中的硬件错误。它有一个看门狗定时器，如果 CPU 在指定的时间内没有响应错误，则触发复位。NMIWDRSn 产生一个持续 810 个 INTOSC1 周期的 XRSn。NMI 看门狗复位后，NMIWDRSn 位被设置。

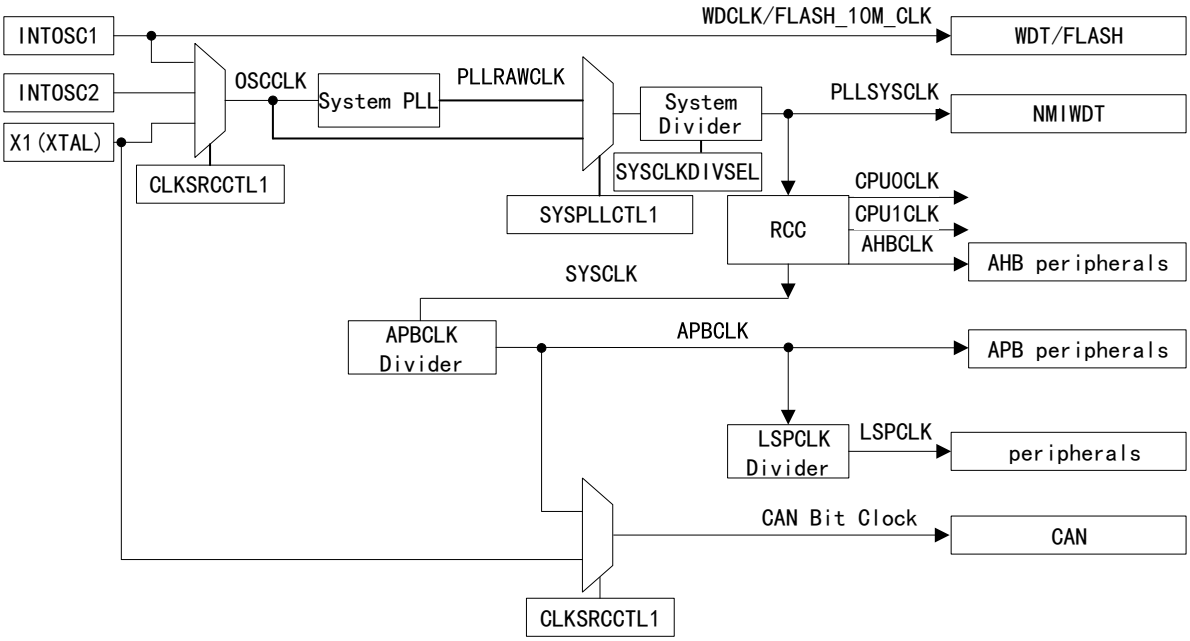
10.1.6 DCS 安全代码复制复位（SCCRESETn）

DCS 可以阻止对闪存某些区域的读访问。提供 ROM 功能来安全地访问这些内存区域可以更加便于 CRC 检查。如果在安全复制或 CRC 函数中发生矢量获取，则 DCS 触发复位。SCCRESETn 与 SYSRS 相似。为了拒绝潜在攻击者的访问，会同时复位调试逻辑。调用函数之前禁用中断可以防止安全漏洞。安全复位后，SCCRESETn 位被设置。

11 时钟

11.1 结构框图

图 3 时钟源



PLL 输出频率最高可支持 250MHz 时钟输出（PLLRAWCLK）。此外，由于是双核架构，且 CPU 以及 CPU 子系统工作频率高于外设（外设最高工作频率 125MHz），因此额外的时钟控制部分由新增的时钟控制单元模块进行控制。时钟控制单元模块在原有控制逻辑之外新增加了 APB 时钟分频以及双核系统时钟控制。计算公式如下：

$$f_{PLLRAWCLK}=（f_{oscclk}）*NF/ODIV$$

$$NF=IMULT+FMULT/4$$

“IMULT”表示 PLL 倍频的整数部分；“FMULT”表示 PLL 倍频的小数部分；“ODIV”表示 PLL 输出分频器。详细功能描述请查阅“SYSPLL 倍频器寄存器”。

11.2 功能描述

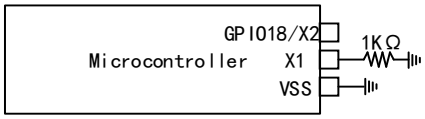
11.2.1 时钟源

主要内部振荡器（INTOSC2）

主要内部时钟源 INTOSC2 是复位时的默认系统时钟。上电时，器件由片上 10MHz 的 INTOSC2 进行时钟控制。用于运行引导 ROM。当 INTOSC2 作为系统时钟源时，可以使用 GPIO18 (X2)作为 GPIO。此时需要在 X1 上连接一个 1k 的下拉电阻。由于与振荡器电路的相互作用，GPIO18 与其他 GPIO 的电气特性不同。需要注意的是，INTOSC2

的频率公差比较松，使用 CAN 模块需要一个外部振荡器才能满足 CAN 的时序要求。

图 4 当 INTOSC2 是 SYSCLK 源时使用 GPIO18



备用内部振荡器（INTOSC1）

10MHz 的 INTOSC1 是一个备份时钟源，一般只对看门狗定时器和缺失的时钟检测电路进行时钟控制。如果使能 MCD 并且检测到系统时钟缺失，则将绕过系统锁相环，所有系统时钟将自动连接到 INTOSC1，也可以手动选择 INTOSC1 作为系统时钟源。

外部振荡器（XTAL）

XTAL 可以作为主系统时钟源和 CAN 位时钟源。外部时钟源使用 X1 和 X2/GPIO18 引脚。

根据 ALT 设置的模式不同，可以支持三种外部时钟源：

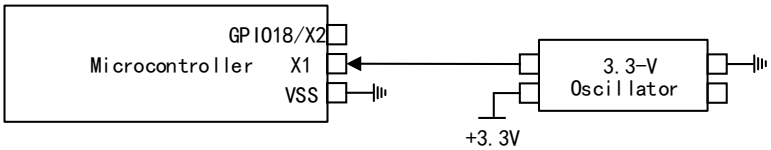
表格 24 ALT 模式

XTALCR[OSCOFF]	SE	模式	GPIO 是否能在 X2 上使用
0	0	晶体方式：石英晶体连接 X1/X2	×
0	1	单端模式：X1 为外部时钟	×
1	0	振荡器关闭	√（使用 GPIO18 时，X1 上需要 1KΩ 的下拉）
1	1	单端模式：X1 为外部时钟（X1 上有大约 1KΩ 的下拉，外部单端时钟要能够驱动此负载）	×

根据上表所示可以支持以下三种外部时钟源：

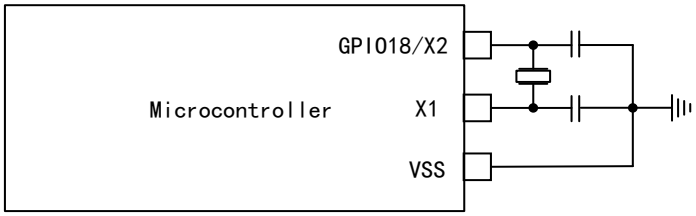
（1） 单端 3.3V 外部时钟。时钟信号接在 X1 上，X2/GPIO18 不能作为 GPIO。

图 5 单端 3.3V 外部时钟



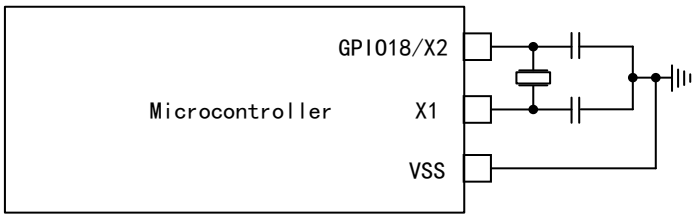
（2） 外部晶体。晶体穿过 X1 和 X2 连接，负载电容连接到 VSS 上。

图 6 外部晶体



（3） 外部谐振器。谐振器跨 X1 和 X2 连接，接地连接到 VSS 上。

图 7 外部谐振器



11.2.2 衍生时钟

时钟源可以使用锁相环相乘并分解，以产生所需的时钟频率。在这个过程中会产生一组衍生时钟。

振荡器时钟（OSCCLK）

必须在三个时钟源中选择一个作为 CPU 和大多数外设的主参考时钟(OSCCLK)。OSCCLK 可以直接使用或通过系统锁相环馈送以达到更高的频率。复位时 OSCCLK 是默认的系统时钟，并连接到 INTOSC2。

系统锁相环输出时钟（PLLRAWCLK）

PLLRAWCLK 是锁相环的压控振荡器(VCO)的输出。允许设备在其最大额定工作频率下运行，并在大多数应用中产生主系统时钟。该锁相环将 OSCCLK 作为参考，并具有分数乘法器。

11.2.3 设备时钟域

为设备中各个模块提供时钟输入。可以直接或通过附加分压器连接到衍生时钟。

系统时钟（PLLSYSCLK）

PLLSYSCLK 是 NMI 看门狗定时器单独的时钟域，最高可支持 250MHz 时钟输出（PLLRAWCLK）。在 HALT 模式下进行门控。虽然名称不同，但 PLLSYSCLK 可以连接到系统 PLL (PLLRAWCLK)或 OSCCLK。所选的时钟源通过分频器运行，由 SYSCLKDIVSEL 寄存器配置。

CPU0 时钟（CPU0CLK）

CPU0CLK 与 SYSCLK 都是由 PLLSYSCLK 产生，为同步时钟。CPU0CLK 的开关由时钟控制单元控制。当 CPU0 处于 SLEEPING（对应 IDLE 模式）或 SLEEPDEEP（对应 HALT 模式）时，时钟控制单元关闭 CPU0CLK。在退出 IDLE 或 HALT 模式时，时钟控制单元开启 CPU0CLK。此外，当系统处于单核工作模式时，时钟控制单元将基于系统当前选择的工作核心来开启或关闭 CPU0CLK。

CPU1 时钟（CPU1CLK）

CPU1CLK 与 SYSCLK 都是由 PLLSYSCLK 产生，为同步时钟。CPU1CLK 的开关由时钟控制单元控制。当 CPU1 处于 SLEEPING（对应 IDLE 模式）或 SLEEPDEEP（对应 HALT 模式）时，时钟控制单元关闭 CPU1CLK。在退出 IDLE 或 HALT 模式时，时钟控制单元开启 CPU1CLK。此外，当系统处于单核工作模式时，时钟控制单元将基于系统当前选择的工作核心来开启或关闭 CPU1CLK。

CPU 子系统时钟（SYSCLK）

SYSCLK 由 SYSCLK divider 生成。用于 WWDT 以外的所有外设，总线和 SRAM 等的工作。SYCLK 的开关由时钟控制单元控制，当系统进入 HALT 模式时，时钟控制单元会自动关闭 SYSCLK 以降低功耗。此时，所有外设时钟均被关闭（不受当前 PCLKCR 的配置影响）。

APBCLK

APBCLK 由 SYSCLK 分频得到，是系统时钟的同步时钟。由于外设无法跑到与 CPU 即 AHB 总线相同的工作频率（250MHz），需要将 SYSCLK 分频以确保外设可以正常工作。APBCLK 支持 1，2 分频。上电后默认 2 分频，当系统需要切换到 250MHz 时，需要先将 APBCLK 配置为至少 2 分频，以确保 APBCLK 不会超过外设的工作频率。

注意：

（1） 配置 APB 分频前需要先确保 DMA 处于 IDLE 状态。

（2） 配置分频后需要读一下分频配置寄存器，然后插入 10 个 nop 指令。以确保分频切换完毕后再对 APB 外设进行读写访问。

低速外设时钟（LSPCLK）

UART 和 SPI 模块连接到一个共享时钟分配器，该分配器生成一个从 APBCLK 衍生的低速外设时钟(LSPCLK)。LSPCLK 默认使用 1/4 分频器，也可以使用 LOSPCP 寄存器更改该比率。模块可以以比 CPU 频率慢得多的比特率进行通信。每个 UART 和 SPI 模块的时钟可以使用 PCLKCRx 寄存器独立门控。

TMR2 时钟（TMR2CLK）

定时器默认连接到 APBCLK。但 TMR2 可以使用 TMR2CLKCTL 寄存器配置连接到 INTOSC1，INTOSC2 或 XTAL。同时为 TMR2 提供了一个单独的预量程分频器。一般情

况下，TMR2 在 SYSCLK 之外运行。如果需要内部频率测量而使用非 SYSCLK 源，必须将源的频率降低到 SYSCLK 频率的一半以下。

CAN 位时钟

位定时设置和网络配置 CAN 位时钟所需的频率容差，可以严格到 0.1%。CAN 位时钟的频率不能比 APBCLK 频率大。由于以 APBCLK 形式的主系统时钟不够精确，位时钟可以使用 CLKSRCCTL2 寄存器连接到 XTAL。每个 CAN 模块都可以独立选择。

11.2.4 时钟连接

表格 25 时钟连接

时钟域	模块
PLLSYSCLK	NMIWDT
CAN Bit Clock	CAN
WDTCLK (INTOSC1)	WDT
LSPCLK	UART
	SPI
CPUCLK	CPU
	Flash
	FPU
	TMU
	VCU
	TCM（即 ITCM 和 DTCM）
	Boot ROM
SYSCLK	GPIO 输入同步
	CFGSMS
	NVMC
	DCS
	QSPI
	EXTI
	SYSC
	TMR
	DMA
	Analog sub system
APBCLK	DCC

时钟域	模块
	PWM
	I2C
	LIN
	ADC
	DAC
	COMP
	PMBUS
	CAP
	QEP
	SDF
	FLB

11.2.5 时钟源和锁相环

应用程序需求决定时钟配置。应用性能、功耗、系统总成本、EMC 等需要决定期望的 CPU 频率，是否需要 CAN 及要用什么类型的外部振荡器或时钟源。如果需要 CAN，则使用精确频率的外部时钟源作为参考时钟。否则只使用 INTOSC2，以避免需要更多的外部组件。

11.2.6 系统时钟设置

默认使用 INTOSC2 作为系统时钟（PLLSYSCLK），配置应用程序所需时钟的步骤如下：

（1） 写入 OSCCLKSRCSEL 位选择参考时钟源。时钟源修改需等待至少 300 个 NOP 指令才能生效。若要启用 XTAL，请按照 11.2.9 节的说明进行操作。

（2） 若需使用 PLL，使用 SysCtl_setClock()函数配置 PLL 时钟。

（3） 写入 LSPCLK 位选择 LSPCLK 分频。

（4） 若需切换 CAN BIT CLK，写入 CLKSRCCTL2。

（5） 写入 PCLKCRx 寄存器使能所需的外设时钟。

系统时钟配置可以在运行时更改。改变 OSCCLK 的时钟源会自动绕过 PLL 并将倍频系数设为零。将 PLL 的倍频系数从非零值修改为其他值会暂时绕过 PLL，直到 PLL 重新锁定。改变 OSSCLK 的时钟源后，至少需要等待 300 个 CPU 时钟周期。

11.2.7 选择锁相环设置

锁相环分为倍频和分频，具体参数与他们之间的关系如下：

表格 26 参数说明

参数	描述
f_{OSCCLK}	系统振荡器时钟频率
IMULT	倍频的整数部分
FMULT	倍频的小数部分
ODIV	PLL 输出分频器
SYSCLKDIVSEL	系统时钟分频器

公式如下：

$f_{PLLSYSCLK} = f_{OSCCLK} * (IMULT + FMULT) / (ODIV * PLLSYSCLKDIV)$

倍频和分频的多种组合可以产生相同的输出频率。但是参考时钟频率与 VCO 频率的乘积不能超过 f_{VCO} 的范围。压控振荡器频率除以 PLL 输出频率不能超过 $f_{PLLRAWCLK}$ 的范围。系统时钟频率不能超过 f_{SYSCLK} 的范围。不允许振荡公差。

11.2.8 外部时钟输出（XCLKOUT）

外部时钟输出特性通过将时钟连接到外部引脚使得可以直接观察时钟。时钟源有 PLLSYSCLK，PLLRAWCLK，SYSCLK，APBCLK，FLPUMPOSC，INTOSC1，INTOSC2 和 XTAL。如果要使用 XCLKOUT，需要先配置 CLKSRCCTL3 寄存器选择时钟源，然后用 XCLKOUTDIVSEL 寄存器选择输出分频器，再将 GPIO16 或 GPIO18 连接到多路通道 11。

11.2.9 使用外部晶体或谐振器

GPIO18 在 X2 引脚上。上电时 X2 引脚处于 GPIO 模式，片上晶体振荡器关闭电源。

将引脚切换到 X2 模式并使能振荡器的步骤如下：

（1） 清除 XTALCR[OSCOFF]位。

（2） 等待晶体通电。（一般需要等待 1 毫秒，具体取决于晶体）

（3） 不停往 X1CNT 的 CLR 位写 1，直至 X1CNT 寄存器中的计数值不等于 1023，然后等待计数值变为 1023，重复此过程 3 次。

（4） 写入 OSCCLKSRCSEL 位，选择 XTAL 作为 OSCCLK 源。

（5） 检查 MCDCR 寄存器的 MCLKSTS 位，若 MCLKSTS 为 1，则需要更多时间等待振荡器上电：
- 写入 MCLKCLR 位清除 MCLKSTS
- 重复步骤 2~5（此时不能复位设备，否则振荡器会断电，需从步骤 1 开始重新执行）
- 如果振荡器在 10 毫秒内没有完成上电，则存在真实的时钟故障

（6） 若 MCLKSTS 为 0，表明振荡器启动成功，系统时钟已来自 XTAL。

11.2.10 使用外部振荡器

使用连接到 X1 引脚的外部振荡器的过程与使用晶体或谐振器的过程相似：

（1） 清除 XTALCR[OSCOFF]位。

（2） 置位 SE 位使能单端模式。

（3） 不停往 X1CNT 的 CLR 位写 1，直至 X1CNT 寄存器中的计数值不等于 1023，然后等待计数值变为 1023，重复此过程 3 次。

（4） 写入 OSCCLKSRCSEL 位，选择 XTAL 作为 OSCCLK 源。

（5） 如果 MCLKSTS 位为 1，则外部振荡器或设备故障。

（6） MCLKSTS 为 0 表示振荡器启动成功，系统时钟从 XTAL 衍生而来。

（7） 时钟丢失检测（MCD）

MCD 子系统使用 INTOSC1 作为参考时钟检测 OSCCLK 故障，只检测 OSCCLK 的丢失，不检测频率漂移。OSCCLK 驱动一个 7 位计数器 MCDPCNT。INTOSC1 驱动一个 13 位计数器 MCDSCNT。当 MCDPCNT 溢出时，MCDSCNT 复位。因此，如果 OSCCLK 存在并且频率大于 INTOSC1 频率的 1/64，MCDSCNT 将永远不会溢出。如果 OSCCLK 停止，MCDSCNT 将溢出此时即检测到了时钟丢失。

当 MCD 子系统检测到 OSCCLK 丢失时：

- MCLKSTS 置位，此时 OSCCLKSRCSEL 位不起作用
- CLOCKFAIL 信号拉高，产生跳闸事件到 PWM 模块并触发 NMI
- 为了防止继续检测时钟丢失，MCDSCNT 计数器被冻结
- 系统时钟不再来自 PLL，OSCCLK 时钟源切换为 INTOSC1。PLLMULT 自动归零

往 MCLKCLR 写 1 可以清除已检测到的时钟丢失。同时恢复 OSCCLKSRCSEL 位的功能并复位 MCD 计数器。若要在时钟丢失的情况下锁定 PLL，配置 OSCCLKSRCSEL，选择 INTOSC1 为 OSCCLK 时钟源，清除已检测到的时钟丢失情况，写入 PLL 寄存器。

11.3 寄存器组地址

表格 27 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
ClkRegs	CLK_REGS	0x5002_0800	0x5002_09FF

11.4 寄存器地址映射

表格 28 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
CLKCFGLOCK1	CLK 锁定寄存器	0x04	√
CLKSRCCTL1	时钟源控制寄存器 1	0x10	√
CLKSRCCTL2	时钟源控制寄存器 2	0x14	√
CLKSRCCTL3	时钟源控制寄存器 3	0x18	√
SYSPLLCTL1	SYSPLL 控制寄存器 1	0x1C	√
SYSPLLMULT	SYSPLL 倍频器寄存器	0x28	√
SYSPLLSTS	SYSPLL 标志寄存器	0x2C	-
SYSCLKDIVSEL	系统时钟分频器寄存器	0x44	√
XCLKOUTDIVSEL	XCLKOUT 分频器寄存器	0x50	√
LOSPCP	低速时钟源预分频器寄存器	0x58	√
MCDCR	缺少时钟控制寄存器	0x5C	√
X1CNT	X1 时钟上的 10 位计数器	0x60	-
XTALCR	XTAL 控制寄存器	0x64	√

11.5 寄存器功能描述

11.5.1 CLK 锁定寄存器(CLKCFGLOCK1)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CLKSRCCTL1	R/WSonce	锁定 CLKSRCCTL1 寄存器（CLKSRCCTL1 register Lock） 0：未锁定 1：锁定	0h
1	CLKSRCCTL2	R/WSonce	锁定 CLKSRCCTL2 寄存器（CLKSRCCTL2 register Lock） 0：未锁定 1：锁定	0h
2	CLKSRCCTL3	R/WSonce	锁定 CLKSRCCTL3 寄存器（CLKSRCCTL3 register Lock） 0：未锁定 1：锁定	0h

位/域	名称	R/W	描述	复位值
3	SYSPLLCTL1	R/WSonce	锁定 SYSPLLCTL1 寄存器（SYSPLLCTL1 register Lock） 0：未锁定 1：锁定	0h
5:4	保留			0h
6	SYSPLLMULT	R/WSonce	锁定 SYSPLLMULT 寄存器（SYSPLLMULT register Lock） 0：未锁定 1：锁定	0h
10:7	保留			0h
11	SYSCLKDIVSEL	R/WSonce	锁定 SYSCLKDIVSEL 寄存器（SYSCLKDIVSEL register Lock） 0：未锁定 1：锁定	0h
14:12	保留			0h
15	LOSPCP	R/WSonce	锁定 LOSPCP 寄存器（LOSPCP register Lock） 0：未锁定 1：锁定	0h
16	XTALCR	R/WSonce	锁定 XTALCR 寄存器（XTALCR register Lock） 0：未锁定 1：锁定	0h
31:17	保留			0h

11.5.2 时钟源控制寄存器 1（CLKSRCCTL1）

偏移地址：0x10

复位类型：XRSn

注意：由于向 CLKSRCCTL1[INTOSC2OFF]或 CLKSRCCTL1[WDHALTI]位执行写配置（写 1、清 0），将会清除 SYSPLLMULT 寄存器，因此建议在软件上先配置 CLKSRCCTL1 寄存器后再配置 SYSPLLMULT 寄存器。

位/域	名称	R/W	描述	复位值
1:0	OSCCLKSRCSEL	R/W	选择振荡器时钟源（Oscillator Clock Source Select） 00：INTOSC2 01：外部振荡器(XTAL) 10：INTOSC1（建议在缺少时钟检测时使用） 11：保留(默认为 INTOSC1) 在上电或 XRSn 后，默认选择 INTOSC2。使用这个位域改变时钟源时，SYSPLLMULT 被强制为零，PLL 将被绕过并断电。这可以防止潜在的锁相环过冲。必须在 SYSPLLMULT 中配置适当的乘数。需要等待 10 个 OSCCLK 周期才能在 SYSPLLMULT 中配置适当的倍频或禁用前一个时钟源以允许更改完成。	0h
2	保留			0h

位/域	名称	R/W	描述	复位值
3	INTOSC2OFF	R/W	关闭内部振荡器 2（Internal Oscillator 2 Off） 0：开启 1：关闭 要确保没有使用时钟源才能禁用时钟源。	0h
4	保留			0h
5	WDHALTI	R/W	忽略看门狗 HALT 模式（Watchdog HALT Mode Ignore） 0：看门狗在 HALT 模式下不工作。当系统进入 HALT 模式时，时钟对看门狗进行门控，INTOSC1/2 断电 1：看门狗在 HALT 模式下工作。当系统进入 HALT 模式时，时钟不对看门狗进行门控，INTOSC1/2 也不断电	0h
31:6	保留			0h

11.5.3 时钟源控制寄存器 2（CLKSRCCTL2）

偏移地址：0x14

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	保留			0h
3:2	CANABCLKSEL	R/W	选择 CANA 位时钟源（CANA Bit-Clock Source Select） 00：APBCLK 01：外部振荡器(XTAL) 1x：保留 丢失的时钟检测电路对这个位域没有影响。	0h
5:4	CANBBCLKSEL	R/W	选择 CANB 位时钟源（CANB Bit-Clock Source Select） 00：APBCLK 01：外部振荡器(XTAL) 1x：保留 丢失的时钟检测电路对这个位域没有影响。	0h
31:6	保留			0h

11.5.4 时钟源控制寄存器 3（CLKSRCCTL3）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	XCLKOUTSEL	R/W	选择 XCLKOUT 源（XCLKOUT Source Select） 000：PLLSYSCLK 001：PLRAWCLK 010：SYSCLK 011：APBCLK 100：FLPUMPOSC 101：INTOSC1 110：INTOSC 2 111：XTAL OSC o/p 时钟	0h
31:3	保留			0h

11.5.5 SYSPLL 控制寄存器 1（SYSPLLCTL1）

偏移地址：0x1C

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	PLLEN	R/W	使能 SYSPLL（SYSPLL Enabled） 0：禁止，时钟系统从 OSCCLK 直接传输 1：使能	0h
1	PLLCLKEN	R/W	使能 SYSPLLCLK（SYSPLLCLK Enabled） 决定生成 PLLSYSCLK 时是否绕过 SYSPLL。 0：绕过 SYSPLL。时钟系统从 OSCCLK 直接传输 1：从 SYSPLL 时钟输出中提供 PLLSYSCLK。要确保锁相环在使能这个时钟给系统之前是锁定的。	0h
31:2	保留			0h

11.5.6 SYSPLL 倍频器寄存器（SYSPLLMULT）

偏移地址：0x28

复位类型：XRSn

注意：

（1） 由于向 CLKSRCCTL1[INTOSC2OFF]或 CLKSRCCTL1[WDHALTI]位执行写配置（写 1、清 0），将会清除 SYSPLLMULT 寄存器。因此建议在软件上先配置 CLKSRCCTL1 寄存器后再配置 SYSPLLMULT 寄存器。

（2） FMULT 和 IMULT 位域必须同时写入， 以确保 PLL 正常工作。

位/域	名称	R/W	描述	复位值
6:0	IMULT	R/W	选择 SYSPLL 整数倍频（SYSPLL Integer Multiplier Select） 0000000- 0000001：1 0000010：2 0000011：3 ... 1111111：127	0h
7	保留			0h

位/域	名称	R/W	描述	复位值
9:8	FMULT	R/W	选择 SYSPLL 小数倍频（SYSPLL Fractional Multiplier Select） 00：0 01：0.25 10：0.5 11：0.75	0h
15:10	保留			0h
18:16	ODIV	R/W	SYSPLL 输出时钟分频器（SYSPLL Output Clock Divider） 锁相环输出分压器= ODIV + 1	0h
31:19	保留			0h

11.5.7 SYSPLL 标志寄存器（SYSPLLSTS）

偏移地址：0x2C

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	LOCKS	R	SYSPLL 锁定标志（SYSPLL Lock Flag） 0：未锁定 1：锁定	0h
1	SLIPS	R	SYSPLL 超出锁定范围标志（SYSPLL is out of lock range Flag） 0：未超出锁定范围 1：超出锁定范围	0h
31:2	保留			0h

11.5.8 系统时钟分频器寄存器（SYSCLKDIVSEL）

偏移地址：0x44

复位类型：XRSn

位/域	名称	R/W	描述	复位值
5:0	PLLSYSCLKDIV	R/W	选择 SYSCLK 分频器（SYSCLK Divide Select） 000000：1 分频 000001：2 分频 000010：4 分频(复位时默认) 000011：6 分频 000100：8 分频 ... 111111：126 分频	2h
31:6	保留			0h

11.5.9 XCLKOUT 分频器寄存器（XCLKOUTDIVSEL）

偏移地址：0x50

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	XCLKOUTDIV	R/W	选择 XCLKOUT 分频（XCLKOUT Divide Select） 00：1 分频 01：2 分频 10：4 分频 11：8 分频(复位时默认)	3h
31:2	保留			0h

11.5.10 低速时钟源预分频器寄存器（LOSPCP）

偏移地址：0x58

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	LSPCLKDIV	R/W	选择低速外设时钟频率（low-speed peripheral clock Divide Select） 该时钟用作 UART 和 SPI 模块的频闪。 000：1 分频 001：2 分频 010：4 分频(复位时默认) 011：6 分频 ... 111：14 分频	2h
31:3	保留			0h

11.5.11 缺少时钟控制寄存器（MCDCR）

偏移地址：0x5C

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	MCLKSTS	R	缺少时钟标志（Missing Clock Flag） 0：不缺少时钟标志 1：缺少时钟标志	0h
1	MCLKCLR	R-0/W1S	清除缺少时钟标志（Missing Clock Flag Clear） 置位此位可以清除 MCLKSTS 位并复位丢失的时钟检测电路。	0h
2	MCLKOFF	R/W	禁止缺少时钟检测（Missing Clock Detect Disable） 0：使能 1：禁止	0h
3	OSCOFF	R/W	振荡器时钟与 MCD 断开连接（Disconnect the OSCCLK from the MCD） 0：连接 1：断开连接	0h
31:4	保留			0h

11.5.12 X1 时钟上的 10 位计数器（X1CNT）

偏移地址：0x60

复位类型：POR（仅[9:0]有复位类型）

位/域	名称	R/W	描述	复位值
9:0	X1CNT	R	X1 计数器（X1 Counter:） 此计数器在每个 X1 时钟的正边上递增，一旦达到 0x3FF 的值就会冻结。这个计数器必须要饱和才能从 INTOSC2 切换到 X1，以确保连接到 X1/X2 的晶体振荡。	0h
15:10	保留			0h
16	CLR	R-0/W1S	清除 X1 计数器（X1 Counter clear） 对此位域写 0 忽略，写 1 将清除 X1CNT 位，并再次从 0x0 开始计数。	0h
31:17	保留			0h

11.5.13 XTAL 控制寄存器（XTALCR）

偏移地址：0x64

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	OSCOFF	R/W	禁止 XTAL 振荡器宏（XTAL oscillator macro Disable） 0：使能 1：禁止 如果晶体连接到 X1/X2，需要先清除这个位，等振荡器上电后将时钟源切换到 X1/X2。	1h
1	SE	R/W	配置 XTAL 振荡器模式（XTAL oscillator mode Configures） 配置当 XTAL 振荡器上电时的模式。 0：晶体模式 1：单端模式	0h
2	保留			1h
31:3	保留			0h

12 系统控制

12.1 SPEC 功能描述

12.1.1 系统控制寄存器配置限制

系统控制中的内存映射寄存器在 INTOSC1 时钟域上操作。为了避免第二次写入可能会丢失，CPU 对这些寄存器的写入都需要在后续写入之间有一个延迟。应用程序需要考虑到这一点，并根据以下提到的每次写入这些寄存器后 NOP 指令的数量添加延迟。计算后续写入之间延迟的公式：

$$延迟(SYSCLK\ 周期) = 3 \times (F_{SYSCLK} \div F_{INTOSC1}) + 9$$

每次写入后需要延迟的寄存器为：

- FLBCLKCTL
- PERCLKDIVSEL
- SYSCLKDIVSEL
- SYSPLLCTL1
- SYSPLLMULT
- WDCR
- XCLKOUTDIVSEL
- XTALCR
- CLKSRCCTL1
- CLKSRCCTL2
- CLKSRCCTL3
- CPU1TMR2CTL (TMR2CLKCTL)

12.2 SPEC 寄存器组地址

表格 29 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
SpecRegs	SPEC_REG	0x5002 0000	0x5002 03FF

12.3 SPEC 寄存器地址映射

表格 30 SPEC_REG 寄存器组地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
SPEC_EMUBOOTPINCFG	仿真 BOOT 配置寄存器	0x00	-
SPEC_EMUBOOTDEFL	仿真 BOOT 定义寄存器低	0x04	-
SPEC_EMUBOOTDEFH	仿真 BOOT 定义寄存器高	0x08	-

寄存器名称	寄存器描述	偏移地址	WRPRT
SPEC_BOOTADDR1	CPU1 BOOT 地址寄存器	0x54	-
SPEC_BOOTCTRL	BOOT 控制寄存器	0x58	-
SPEC_CFGSMSSEL	选择 CFGSMS 静态配置寄存器	0x64	-
SPEC_APBCLKDIVCFG	APB 时钟分频配置寄存器	0x74	-
SPEC_VOS	调压寄存器	0x78	-

12.4 SPEC 寄存器功能描述

12.4.1 仿真 BOOT 配置寄存器(SPEC_EMUBOOTPINCFG)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	BOOTSEL0	R	BOOT 方式选择引脚 0（Boot Mode Select Pin 0） 设置为启动时使用的 GPIO 引脚(最多 255)。 0x00 = GPIO0 0x01 = GPIO1 ….. 如果 0xFF 无效，其他 BOOTSEL 也设置为 0xFF，则选择出厂默认选择的 BOOTSEL 0。如果其他 BOOTSEL 没有设置为 0xFF，那么将 BOOTSEL 设置为 0xFF 将禁用该特定的 BOOTSEL。	0h
15:8	BOOTSEL1	R	BOOT 方式选择引脚 1（Boot Mode Select Pin 1） 参考 BOOTSEL0 描述。	0h
23:16	BOOTSEL2	R	BOOT 方式选择引脚 2（Boot Mode Select Pin 2） 参考 BOOTSEL0 描述。	0h
31:24	KEY	R	关键字（Key Word） 将 0x5A 写入这 8 位，告诉引导 ROM 代码这个寄存器中的位有效。	0h

12.4.2 仿真 BOOT 定义寄存器低(SPEC_EMUBOOTDEFL)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	BOOTDEF0	R	BOOT_DEF0 模式/选项（BOOT_DEF0 Mode/Options） 包括更改特定引导外设的 GPIO 或指定不同的 Flash 接入点。任何不支持的引导模式都会导致设备复位。	0h
15:8	BOOTDEF1	R	BOOT_DEF1 模式/选项（BOOT_DEF1 Mode/Options） 参考 BOOTDEF0 描述。	0h
23:16	BOOTDEF2	R	BOOT_DEF2 模式/选项（BOOT_DEF2 Mode/Options） 参考 BOOTDEF0 描述。	0h
31:24	BOOTDEF3	R	BOOT_DEF3 模式/选项（BOOT_DEF3 Mode/Options） 参考 BOOTDEF0 描述。	0h

12.4.3 仿真 BOOT 定义寄存器高(SPEC_EMUBOOTDEFH)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	BOOTDEF4	R	BOOT_DEF4 模式/选项（BOOT_DEF4 Mode/Options） 参考 BOOTDEF0 描述。	0h
15:8	BOOTDEF5	R	BOOT_DEF5 模式/选项（BOOT_DEF5 Mode/Options） 参考 BOOTDEF0 描述。	0h
23:16	BOOTDEF6	R	BOOT_DEF6 模式/选项（BOOT_DEF6 Mode/Options） 参考 BOOTDEF0 描述。	0h
31:24	BOOTDEF7	R	BOOT_DEF7 模式/选项（BOOT_DEF7 Mode/Options） 参考 BOOTDEF0 描述。	0h

12.4.4 CPU1 BOOT 地址寄存器(SPEC_BOOTADDR1)

偏移地址：0x54

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	BOOTADDR1	R	CPU1 boot 地址（CPU1 boot address） 启动地址由用户选项字节定义或通过总线写入。如果启动地址设置为不可访问的存储区域，会导致启动失败。该寄存器会被选项字节加载并且系统复位时不会复位，当 Option byte Opt_byte_key 值不为 0x55AA 时，一旦加载值为 0x2010 0000 就会启动地址。	0h
31:10		W/R		X

12.4.5 BOOT 控制寄存器(SPEC_BOOTCTRL)

偏移地址：0x58

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			1h
1	BOOTCTRL1	R/W	允许 CPU1 启动（Allows CPU1 to boot） 启动 CTL 由用户通过总线写入定义。 可以由软件设置，在系统复位后由硬件清除。 0：如果 BOOT_CFG[1]设置为 0，CPU1 将不启动。(重置后默认) 1：CPU1 将独立于 BOOT_CFG[1]的值启动。	0h
31:2	保留			0h

12.4.6 选择 CFGSMS 静态配置寄存器(SPEC_CFGSMSSEL)

偏移地址：0x64

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	SCAHRPWMCFG	R/W	选择 CFGSMS 静态配置（CFGSMS Static Configuration selcet） 0000：静态配置 0 0001：静态配置 1 …… 1111：静态配置 16	0h
31:4	保留			0h

12.4.7 APB 时钟分频配置寄存器（SPEC_APBCLKDIVCFG）

偏移地址：0x74

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	APBCLKDIVCFG	R/W	配置 APB 时钟分频（APB clock divide configure） 00000：1:1 00001：2:1 其他：保留	2h
31:5	保留			0h

12.4.8 调压寄存器（SPEC_VOS）

偏移地址：0x78

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	VOSEN	R/W	使能 VOS（VOS Enable） 0：禁止 1：使能	0h
3:1	LDOVSCACFG	R/W	配置 LDO 电压扩展（LDO voltage scaling configure） 000：VDD 电压 1.1V 001：保留 010：保留 011：VDD 电压 1.0V 100：VDD 电压 1.05V 101：VDD 电压 1.1V 11x：VDD 电压 1.15V	0h
31:4	保留			0h

12.5 NMI_INTRUPT 寄存器组地址

表格 31NMI_INTRUPT 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
NmiIntruptRegs	NMI_INTRUPT_REGS	0x5002 64C0	0x5002 64CC

12.6 NMI_INTRUPT 寄存器地址映射

表格 32 NMI_INTRUPT 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
NMICFG	配置寄存器	0x00	√
NMIFLG	标志寄存器	0x02	-
NMIFLGCLR	标志清除寄存器	0x04	√
NMIFLGFRC	标志强制寄存器	0x06	√
NMIWDCNT	看门狗计数器寄存器	0x08	-
NMIWDPRD	看门狗周期寄存器	0x0A	√
NMISHDFLG	影子标志寄存器	0x0C	-

12.7 NMI_INTRUPT 寄存器功能描述

12.7.1 配置寄存器（NMICFG）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	NMIE	R/W1S	使能 NMI（Global NMI Enable） 该位表示 NMI 模块已使能，允许系统错误情况触发对 CPU 的 NMI。引导 ROM 在启动时设置这个位。只能通过系统重置来清除。	0h
15:1	保留			0h

12.7.2 标志寄存器（NMIFLG）

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	NMIINT	R	NMI 标志（NMI Flag） 0：无标志 1：产生标志 表示是否产生了 NMI。它可以通过系统复位或通过写入 NMIFLGCLR 寄存器中的相应位来清除。在清除此标志之前，不会再生成 NMI。	0h
1	CLOCKFAIL	R	时钟错误标志（Clock Error Flag） 0：无标志 1：产生标志 表示是否检测到时钟错误情况。它可以通过系统复位或通过写入 NMIFLGCLR 寄存器中的相应位来清除。	0h
2	保留			0h

位/域	名称	R/W	描述	复位值
3	FLUNCERR	R	Flash 无法修复错误标志（Flash Uncorrectable Error Flag） 0：无标志 1：产生标志 表示 flash 访问过程中是否发生不可纠正的 ECC 错误。它可以通过系统复位或通过写入 NMIFLGCLR 寄存器中的相应位来清除。	0h
7:4	保留			0h
8	FLBNMI	R	FLB 产生 NMI 标志（FLB Generates NMI flag） 0：无标志 1：产生标志 表示是否由 FLB 生成 NMI。它可以通过系统复位或通过写入 NMIFLGCLR 寄存器中的相应位来清除。	0h
15:14	保留			0h
13	SWERR	R	软件强制 NMI 标志（Software-Forced NMI Flag） 0：无标志 1：产生标志 该位只能通过写入 NMIFLGFRC 寄存器中的相应位来设置。它可以通过系统复位或通过写入 NMIFLGCLR 寄存器中的相应位来清除。在清除此标志之前，不会再生成 NMI。	0h
15:14	保留			0h

12.7.3 标志清除寄存器（NMIFLGCLR）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	NMIINT	R-0/W1S	清除 NMI 标志（NMI Flag Clear） 清除 NMIFLG 寄存器中相应的位。只有在清除了所有其他活动标志之后，才应该清除此标志。	0h
1	CLOCKFAIL	R-0/W1S	清除时钟错误标志（Clock Error Flag Clear） 清除 NMIFLG 寄存器中相应的位。	0h
2	保留			0h
3	FLUNCERR	R-0/W1S	清除 Flash 无法修复错误标志（ Flash Uncorrectable Error Flag Clear） 清除 NMIFLG 寄存器中相应的位。	0h
7:4	保留			0h
8	FLBNMI	R-0/W1S	清除 FLB 产生 NMI 标志（FLB Generates NMI flag Clear） 清除 NMIFLG 寄存器中相应的位。	0h
15:14	保留			0h
13	SWERR	R-0/W1S	清除软件强制 NMI 标志（Software-Forced NMI Flag Clear） 清除 NMIFLG 寄存器中相应的位。	0h

位/域	名称	R/W	描述	复位值
15:14	保留			0h

12.7.4 标志强制寄存器（NMIFLGFRC）

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	CLOCKFAIL	R-0/W1S	设置时钟错误标志（Clock Error Flag Set） 设置 NMIFLG 寄存器中相应的位。	0h
2	保留			0h
3	FLUNCERR	R-0/W1S	设置 Flash 无法修复错误标志（ Flash Uncorrectable Error Flag Set） 设置 NMIFLG 寄存器中相应的位。	0h
7:4	保留			0h
8	FLBNMI	R-0/W1S	设置 FLB 产生 NMI 标志（FLB Generates NMI flag Set） 设置 NMIFLG 寄存器中相应的位。	0h
15:14	保留			0h
13	SWERR	R-0/W1S	设置软件强制 NMI 标志（Software-Forced NMI Flag Set） 设置 NMIFLG 寄存器中相应的位。	0h
15:14	保留			0h

12.7.5 看门狗计数器寄存器（NMIWDCNT）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	NMIWDCNT	R	设置 NMI 看门狗计数器（NMI Watchdog Counter Set） 当设置 NMIFLG 寄存器的位时，看门狗计数器每个 SYSCLK 周期增加一次。如果计数器达到 NMIWDPRD 寄存器中的周期值，则 NMI 模块产生 NMIWDRS 复位。复位后，计数器复位为零并停止计数。如果所有的 NMI 标志被清除，计数器将复位为零并停止计数，直到另一个 NMI 标志被设置。	0h

12.7.6 看门狗周期寄存器（NMIWDPRD）

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	NMIWDPRD	R/W	设置 NMI 看门狗周期（NMI Watchdog Period Set） 指定 NMI 看门狗定时器在 SYSCLK 周期中的周期。写入小于当前计数器值的周期值将立即强制复位(NMIWDRS)。	FFFFh

12.7.7 影子标志寄存器（NMISHDFLG）

偏移地址：0x0C

复位类型：PORESETn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	CLOCKFAIL	R-0/W1S	影子时钟错误标志（Shadow Clock Error Flag）	0h
2	保留			0h
3	FLUNCERR	R-0/W1S	影子 Flash 无法修复错误标志（Shadow Flash Uncorrectable Error Flag）	0h
7:4	保留			0h
8	FLBNMI	R-0/W1S	影子 FLB 产生 NMI 标志（Shadow FLB Generates NMI flag）	0h
12:9	保留			0h
13	SWERR	R-0/W1S	影子软件强制 NMI 标志（Shadow Software-Forced NMI Flag）	0h
15:14	保留			0h

12.8 PERIPH_AC 寄存器组地址

表格 33 PERIPH_AC 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
PeriphAcRegs	PERIPH_AC_REGS	0x5002_0C00	0x5002_0FFF

12.9 PERIPH_AC 寄存器地址映射

表格 34 PERIPH_AC 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
ADCA_AC	ADCA 主访问控制寄存器	0x00	√
ADCB_AC	ADCB 主访问控制寄存器	0x04	√
ADCC_AC	ADCC 主访问控制寄存器	0x08	√
COMP1_AC	COMP1 主访问控制寄存器	0x20	√
COMP2_AC	COMP2 主访问控制寄存器	0x24	√
COMP3_AC	COMP3 主访问控制寄存器	0x28	√
COMP4_AC	COMP4 主访问控制寄存器	0x2C	√
COMP5_AC	COMP5 主访问控制寄存器	0x30	√
COMP6_AC	COMP6 主访问控制寄存器	0x34	√

寄存器名称	寄存器描述	偏移地址	WRPRT
COMP7_AC	COMP7 主访问控制寄存器	0x38	√
DACA_AC	DACA 主访问控制寄存器	0x50	√
DACB_AC	DACB 主访问控制寄存器	0x54	√
PWM1_AC	PWM1 主访问控制寄存器	0x90	√
PWM2_AC	PWM2 主访问控制寄存器	0x94	√
PWM3_AC	PWM3 主访问控制寄存器	0x98	√
PWM4_AC	PWM4 主访问控制寄存器	0x9C	√
PWM5_AC	PWM5 主访问控制寄存器	0xA0	√
PWM6_AC	PWM6 主访问控制寄存器	0xA4	√
PWM7_AC	PWM7 主访问控制寄存器	0xA8	√
PWM8_AC	PWM8 主访问控制寄存器	0xAC	√
QEP1_AC	QEP1 主访问控制寄存器	0xE0	√
QEP2_AC	QEP2 主访问控制寄存器	0xE4	√
CAP1_AC	CAP1 主访问控制寄存器	0x100	√
CAP2_AC	CAP2 主访问控制寄存器	0x104	√
CAP3_AC	CAP3 主访问控制寄存器	0x108	√
CAP4_AC	CAP4 主访问控制寄存器	0x10C	√
CAP5_AC	CAP5 主访问控制寄存器	0x110	√
CAP6_AC	CAP6 主访问控制寄存器	0x114	√
CAP7_AC	CAP7 主访问控制寄存器	0x118	√
SDF1_AC	SDF1 主访问控制寄存器	0x150	√
FLB1_AC	FLB1 主访问控制寄存器	0x160	√
FLB2_AC	FLB2 主访问控制寄存器	0x164	√
FLB3_AC	FLB3 主访问控制寄存器	0x168	√
FLB4_AC	FLB4 主访问控制寄存器	0x16C	√
PROMCRC_AC	PROMCRC 主访问控制寄存器	0x180	√
SPIA_AC	SPIA 主访问控制寄存器	0x220	√
SPIB_AC	SPIB 主访问控制寄存器	0x224	√
PMBUS_A_AC	PMBUSA 主访问控制寄存器	0x260	√
LIN_A_AC	LINA 主访问控制寄存器	0x270	√
CANA_AC	CANA 主访问控制寄存器	0x280	√

寄存器名称	寄存器描述	偏移地址	WRPRT
CANB_AC	CANB 主访问控制寄存器	0x284	√
HRPWM_A_AC	HRPWMA 主访问控制寄存器	0x354	√
PERIPH_AC_LOCK	锁定寄存器	0x3FC	√

12.10 PERIPH_AC 寄存器功能描述

12.10.1 ADCA 主访问控制寄存器（ADCA_AC）

偏移地址：0x00

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.2 ADCB 主访问控制寄存器（ADCB_AC）

偏移地址：0x04

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h

位/域	名称	R/W	描述	复位值
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.3 ADCC 主访问控制寄存器（ADCC_AC）

偏移地址：0x08

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.4 COMPx 主访问控制寄存器（COMPx_AC）（x=1-7）

偏移地址：0x20+（a-1）*0x04（a=1-7）

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.5 DACA 主访问控制寄存器（DACA_AC）

偏移地址：0x50

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.6 DACB 主访问控制寄存器（DACB_AC）

偏移地址：0x54

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.7 PWMx 主访问控制寄存器（PWMx_AC）（x=1-8）

偏移地址：0x90+（a-1）*0x04（a=1-8）

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h

位/域	名称	R/W	描述	复位值
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.8 QEPx 主访问控制寄存器（QEPx_AC）（x=1-2）

偏移地址：0xE0+（a-1）*0x04（a=1-2）

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.9 CAPx 主访问控制寄存器（CAPx_AC）（x=1-7）

偏移地址：0x100+（a-1）*0x04（a=1-7）

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h

位/域	名称	R/W	描述	复位值
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.10 SDF1 主访问控制寄存器（SDF1_AC）

偏移地址：0x150

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.11 FLBx 主访问控制寄存器（FLBx_AC）（x=1-4）

偏移地址：0x160+（a-1）*0x04（a=1-4）

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:4	保留			0h

12.10.12 PROMCRC 主访问控制寄存器（PROMCRC_AC）

偏移地址：0x180

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:4	保留			0h

12.10.13 SPIA 主访问控制寄存器（SPIA_AC）

偏移地址：0x220

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h

位/域	名称	R/W	描述	复位值
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.14 SPIB 主访问控制寄存器（SPIB_AC）

偏移地址：0x224

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.15 PMBUSA 主访问控制寄存器（PMBUS_A_AC）

偏移地址：0x260

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.16 LINA 主访问控制寄存器（LIN_A_AC）

偏移地址：0x270

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.17 CANA 主访问控制寄存器（CANA_AC）

偏移地址：0x280

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.18 CANB 主访问控制寄存器（CANB_AC）

偏移地址：0x284

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h

位/域	名称	R/W	描述	复位值
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.19 HRPWMA 主访问控制寄存器（HRPWM_A_AC）

偏移地址：0x354

复位类型：XRSn

位/域	名称	R/W	描述	复位值
1:0	CPU0_ACC	R/W	选择 CPU0 的访问权限（CPU0 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
3:2	CPU1_ACC	R/W	选择 CPU1 的访问权限（CPU1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
5:4	DMA1_ACC	R/W	选择 DMA1 的访问权限（DMA1 access permission Select） 00：不对外设进行读写访问 01：保留 10：保护 RD 不改变读寄存器的 FIFO，Clear 的访问，无写访问 11：读写全访问	3h
31:6	保留			0h

12.10.20 锁定寄存器（PERIPH_AC_LOCK）

偏移地址：0x3FC

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	LOCK_AC_WR0	R/WSonce	选择 CPU0 的访问权限（CPU0 access permission Select） 0：访问控制寄存器允许读写访问 1：访问控制寄存器为只读 写“1”设置位，写“0”不起作用。	0h

位/域	名称	R/W	描述	复位值
1	LOCK_AC_WR1	R/WSonce	选择 CPU1 的访问权限（CPU1 access permission Select） 0：访问控制寄存器允许读写访问 1：访问控制寄存器为只读 写“1”设置位，写“0”不起作用。	0h
31:2	保留			0h

12.11 ROM_PREFETCH 寄存器组地址

表格 35 ROM_PREFETCH 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
RomprefetchRegs	ROM_PREFETCH_REGS	0x5002_1000	0x5002_10FF

12.12 ROM_PREFETCH 寄存器地址映射

表格 36 ROM_PREFETCH 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
BOOT_ROM_PREFETCH	使能 BOOT ROM 预取寄存器	0x00	√
SECURE_ROM_PREFETCH	使能 SECURE ROM 预取寄存器	0x04	√

12.13 ROM_PREFETCH 寄存器功能描述

12.13.1 使能 BOOT ROM 预取寄存器（BOOT_ROM_PREFETCH）

偏移地址：0x00

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	BOOT_ROM_PREFETCH	R/W	使能 BOOT ROM 预取功能（BOOT ROM prefetch function enable） 0：禁止 1：使能	0h
31:1	保留			0h

12.13.2 使能 SECURE ROM 预取寄存器（SECURE_ROM_PREFETCH）

偏移地址：0x04

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	SECURE_ROM_PREFETCH	R/W	使能 SECURE ROM 预取功能（SECURE ROM prefetch function enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
31:1	保留			0h

12.14 ROM_WAIT_STATE 寄存器组地址

表格 37 ROM_WAIT_STATE 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
RomwaitstateRegs	ROM_WAIT_STATE_REGS	0x5002_1100	0x5002_11FF

12.15 ROM_WAIT_STATE 寄存器地址映射

表格 38 ROM_WAIT_STATE 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
BOOT_ROM_LATENCY	BOOT ROM 延迟配置寄存器	0x00	√
SECURE_ROM_LATENCY	SECURE ROM 延迟配置寄存器	0x04	√

12.16 ROM_WAIT_STATE 寄存器功能描述

12.16.1 BOOT ROM 延迟配置寄存器（BOOT_ROM_LATENCY）

偏移地址：0x00

复位类型：XRSn

位/域	名称	R/W	描述	复位值
2:0	BOOT_ROM_LATENCY	R/W	BOOT ROM 延迟配置（BOOT ROM delay configuration） 000：无延迟 001：1 HCLK 延迟 … 111：7 HCLK 延迟	1h
31:3	保留			0h

12.16.2 SECURE ROM 延迟配置寄存器（SECURE_ROM_LATENCY）

偏移地址：0x04

复位类型：XRSn

位/域	名称	R/W	描述	复位值
2:0	SECURE_ROM_LATENCY	R/W	SECURE ROM 延迟配置（SECURE ROM delay configuration） 000：无延迟 001：1 HCLK 延迟 … 111：7 HCLK 延迟	1h

位/域	名称	R/W	描述	复位值
31:3	保留			0h

13 嵌套向量中断控制器（NVIC）

13.1 术语全称、缩写描述

表格 39 术语全称、缩写描述

中文全称	英文全称	英文缩写
不可屏蔽中断	Non Maskable Interrupt	NMI
嵌套向量中断控制器	Nested Vectored Interrupt Controller	NVIC

13.2 简介

内核集成了嵌套向量中断控制器（Nested Vectored Interrupt Controller（NVIC）），它和内核紧密耦合，能高效、低延迟处理异常和中断、电源管理控制。

13.3 主要特征

（1） 227 个可屏蔽中断通道（不包括 16 个内核中断线）

（2） 低延时的异常和中断处理

（3） 电源管理控制

（4） 系统控制寄存器的实现

13.4 中断和异常向量表

表格 40 中断和异常向量表

异常类型	向量编号	优先级	向量地址	描述
-	-	-	0x0000_0000	保留
Reset	-	-15	0x0000_0004	复位
NMI	-	-14	0x0000_0008	不可屏蔽中断
HardFault	-	-13	0x0000_000C	各种硬件故障
MemManage	-	-12	0x0000_0010	内存故障
BusFault	-	-11	0x0000_0014	总线故障
UsageFault	-	-10	0x0000_0018	指令执行故障
SecureFault	~	-9	0x0000_001C	数据安全故障
SVCall	-	-5	0x0000_002C	通用 SWIU 指令调用的系统服务
Debug Monitor	-	-4	0x0000_0030	调试功能异常
PendSV	-	-2	0x0000_0038	可挂起的系统服务

异常类型	向量编号	优先级	向量地址	描述
SysTick	-	-1	0x0000_003C	系统滴答定时器
-	0	可设置	0x0000_0040	保留
-	1	-	0x0000_0044	保留
-	2	可设置	0x0000_0048	保留
-	3	可设置	0x0000_004C	保留
-	4	-	0x0000_0050	保留
-	5	可设置	0x0000_0054	保留
-	6	可设置	0x0000_0058	保留
-	7	可设置	0x0000_005C	保留
-	8	可设置	0x0000_0060	保留
-	9	可设置	0x0000_0064	保留
-	10	可设置	0x0000_0068	保留
DCCOMP interrupt	11	可设置	0x0000_006C	DCCOMP 中断
-	12	可设置	0x0000_0070	保留
TMR1	13	可设置	0x0000_0074	TMR1 中断
TMR2	14	可设置	0x0000_0078	TMR2 中断
-	15	可设置	0x0000_007C	保留
RAM	16	可设置	0x0000_0080	系统存储器中断
COP TE0_INT	17	可设置	0x0000_0084	COP TE0 中断
COP TE1_INT	18	可设置	0x0000_0088	COP TE1 中断
COP TE2_INT	19	可设置	0x0000_008C	COP TE2 中断
COP TE3_INT	20	可设置	0x0000_0090	COP TE3 中断
COP RF0_IN T	21	可设置	0x0000_0094	COP RF0 中断
COP RF1_INT	22	可设置	0x0000_0098	COP RF1 中断
COP RF2_INT	23	可设置	0x0000_009C	COP RF2 中断
COP RF3_INT	24	可设置	0x0000_00A0	COP RF3 中断
COP GP0_INT	25	可设置	0x0000_00A4	COP GP0 中断
COP GP1_INT	26	可设置	0x0000_00A8	COP GP1 中断
COP GP2_INT	27	可设置	0x0000_00AC	COP GP2 中断
COP GP3_INT	28	可设置	0x0000_00B0	COP GP3 中断
-	29	可设置	0x0000_00B4	保留
-	30	可设置	0x0000_00B8	保留
-	31	可设置	0x0000_00BC	保留

异常类型	向量编号	优先级	向量地址	描述
ADCA1	32	可设置	0x0000_00C0	ADCA1 中断
ADCB1	33	可设置	0x0000_00C4	ADCB1 中断
ADCC1	34	可设置	0x0000_00C8	ADCC1 中断
-	35	可设置	0x0000_00CC	保留
-	36	可设置	0x0000_00D0	保留
-	37	可设置	0x0000_00D4	保留
TMR0	38	可设置	0x0000_00D8	TMR0 中断
WWDT	39	可设置	0x0000_00DC	WWDT 中断
PWM1 TRIP ZONE	40	可设置	0x0000_00E0	PWM1 TRIP 中断
PWM2 TRIP ZONE	41	可设置	0x0000_00E4	PWM2 TRIP 中断
PWM3 TRIP ZONE	42	可设置	0x0000_00E8	PWM3 TRIP 中断
PWM4 TRIP ZONE	43	可设置	0x0000_00EC	PWM4 TRIP 中断
PWM5 TRIP ZONE	44	可设置	0x0000_00F0	PWM5 TRIP 中断
PWM6 TRIP ZONE	45	可设置	0x0000_00F4	PWM6 TRIP 中断
PWM7 TRIP ZONE	46	可设置	0x0000_00F8	PWM7 TRIP 中断
PWM8 TRIP ZONE	47	可设置	0x0000_00FC	PWM8 TRIP 中断
PWM1	48	可设置	0x0000_0100	PWM1 中断
PWM2	49	可设置	0x0000_0104	PWM2 中断
PWM3	50	可设置	0x0000_0108	PWM3 中断
PWM4	51	可设置	0x0000_010C	PWM4 中断-
PWM5	52	可设置	0x0000_0110	PWM5 中断
PWM6	53	可设置	0x0000_0114	PWM6 中断
PWM7	54	可设置	0x0000_0118	PWM7 中断
PWM8	55	可设置	0x0000_011C	PWM8 中断
CAP1	56	可设置	0x0000_0120	CAP1 中断
CAP2	57	可设置	0x0000_0124	CAP2 中断
CAP3	58	可设置	0x0000_0128	CAP3 中断
CAP4	59	可设置	0x0000_012C	CAP4 中断
CAP5	60	可设置	0x0000_0130	CAP5 中断
CAP6	61	可设置	0x0000_0134	CAP6 中断
CAP7	62	可设置	0x0000_0138	CAP7 中断
-	63	可设置	0x0000_013C	保留
QEP1	64	可设置	0x0000_0140	QEP1 中断

异常类型	向量编号	优先级	向量地址	描述
QEP2	65	可设置	0x0000_0144	QEP2 中断
-	66	可设置-	0x0000_0148	保留
-	67	可设置	0x0000_014C	保留
FLB1	68	可设置	0x0000_0150	FLB1 中断
FLB2	69	可设置	0x0000_0154	FLB2 中断
FLB3	70	可设置	0x0000_0158	FLB3 中断
FLB4	71	可设置	0x0000_015C	FLB4 中断
SPIA_RX	72	可设置	0x0000_0160	SPIA 接收中断
SPIA_TX	73	可设置	0x0000_0164	SPIA 发送中断
SPIB_RX	74	可设置	0x0000_0168	SPIB 接收中断
SPIB_TX	75	可设置	0x0000_016C	SPIB 发送中断
-	76	-	0x0000_0170	保留
-	77	-	0x0000_0174	保留
-	78	-	0x0000_0178	保留
-	79	-	0x0000_017C	保留
DMA_CH1	80	可设置	0x0000_0180	DMA 通道 1 中断
DMA_CH2	81	可设置	0x0000_0184	DMA 通道 2 中断
DMA_CH3	82	可设置	0x0000_0188	DMA 通道 3 中断
DMA_CH4	83	可设置	0x0000_018C	DMA 通道 4 中断
DMA_CH5	84	可设置	0x0000_0190	DMA 通道 5 中断
DMA_CH6	85	可设置	0x0000_0194	DMA 通道 6 中断
-	86	-	0x0000_0198	保留
-	87	-	0x0000_019C	保留
I2CA	88	可设置	0x0000_01A0	I2CA 中断
I2CA FIFO	89	可设置	0x0000_01A4	I2CA FIFO 中断
QSPI	90	可设置	0x0000_01A8	QSPI 中断
-	91	-	0x0000_01AC	保留
-	92	-	0x0000_01B0	保留
-	93	-	0x0000_01B4	保留
-	94	-	0x0000_01B8	保留
-	95	-	0x0000_01BC	保留
UARTA_RX	96	可设置	0x0000_01C0	UARTA 接收中断
UARTA_TX	97	可设置	0x0000_01C4	UARTA 发送中断

异常类型	向量编号	优先级	向量地址	描述
UARTB_RX	98	可设置	0x0000_01C8	UARTB 接收中断
UARTB_TX	99	可设置	0x0000_01CC	UARTB 发送中断
CANA INT0	100	可设置	0x0000_01D0	CANA 中断 0
CANA INT1	101	可设置	0x0000_01D4	CANA 中断 1
CANB INT0	102	可设置	0x0000_01D8	CANB 中断 0
CANB INT1	103	可设置	0x0000_01DC	CANB 中断 1
ADCA EVENT	104	可设置	0x0000_01E0	ADCA EVENT 中断
ADCA2	105	可设置	0x0000_01E4	ADCA2 中断
ADCA3	106	可设置	0x0000_01E8	ADCA3 中断
ADCA4	107	可设置	0x0000_01EC	ADCA4 中断
ADCB EVENT	108	可设置	0x0000_01F0	ADCB EVENT 中断
ADCB2	109	可设置-	0x0000_01F4	ADCB2 中断
ADCB3	110	可设置	0x0000_01F8	ADCB3 中断
ADCB4	111	可设置	0x0000_01FC	ADCB4 中断
EXTI_LINE0	112	可设置	0x0000_0200	EXTI LINE0 中断
EXTI_LINE1	113	可设置	0x0000_0204	EXTI LINE1 中断
EXTI_LINE2	114	可设置	0x0000_0208	EXTI LINE2 中断
EXTI_LINE3	115	可设置	0x0000_020C	EXTI LINE3 中断
EXTI_LINE4	116	可设置	0x0000_0210	EXTI LINE4 中断
EXTI_LINE5	117	可设置	0x0000_0214	EXTI LINE5 中断
EXTI_LINE6	118	可设置	0x0000_0218	EXTI LINE6 中断
EXTI_LINE7	119	可设置	0x0000_021C	EXTI LINE7 中断
EXTI_LINE8	120	可设置	0x0000_0220	EXTI LINE8 中断
EXTI_LINE9	121	可设置	0x0000_0224	EXTI LINE9 中断
EXTI_LINE10	122	可设置	0x0000_0228	EXTI LINE10 中断
EXTI_LINE11	123	可设置	0x0000_022C	EXTI LINE11 中断
EXTI_LINE12	124	可设置	0x0000_0230	EXTI LINE12 中断
EXTI_LINE13	125	可设置	0x0000_0234	EXTI LINE13 中断
EXTI_LINE14	126	可设置	0x0000_0238	EXTI LINE14 中断
EXTI_LINE15	127	可设置	0x0000_023C	EXTI LINE15 中断
-	128	-	0x0000_0240	保留
FPU DZC	129	可设置	0x0000_0244	FPU DZC 中断
FPU IDC	130	可设置	0x0000_0248	FPU IDC 中断

异常类型	向量编号	优先级	向量地址	描述
FPU IOC	131	可设置	0x0000_024C	FPU IOC 中断
FPU OFC	132	可设置	0x0000_0250	FPU OFC 中断
FPU UFC	133	可设置	0x0000_0254	FPU UFC 中断
FPU IXC	134	可设置	0x0000_0258	FPU IXC 中断
-	135	-	0x0000_025C	保留
-	136	-	0x0000_0260	保留
-	137	-	0x0000_0264	保留
-	138	-	0x0000_0268	保留
-	139	-	0x0000_026C	保留
-	140	-	0x0000_0270	保留
-	141	-	0x0000_0274	保留
-	142	-	0x0000_0278	保留
-	143	-	0x0000_027C	保留
-	144	-	0x0000_0280	保留
-	145	-	0x0000_0284	保留
-	146	-	0x0000_0288	保留
-	147	-	0x0000_028C	保留
-	148	-	0x0000_0290	保留
-	149	-	0x0000_0294	保留
-	150	-	0x0000_0298	保留
-	151	-	0x0000_029C	保留
-	152	-	0x0000_02A0	保留
-	153	-	0x0000_02A4	保留
-	154	-	0x0000_02A8	保留
-	155	-	0x0000_02AC	保留
-	156	-	0x0000_02B0	保留
CAP6 HRcalibration	157	可设置	0x0000_02B4	CAP6 HRcalibration 中断
CAP7 HRcalibration	158	可设置	0x0000_02B8	CAP7 HRcalibration 中断
-	159	可设置	0x0000_02BC	保留
SDF	160	可设置	0x0000_02C0	SDF 中断
-	161	-	0x0000_02C4	保留
-	162	-	0x0000_02C8	保留
-	163	-	0x0000_02CC	保留

异常类型	向量编号	优先级	向量地址	描述
SDF DR INT1	164	可设置	0x0000_02D0	SDF DR 中断 1
SDF DR INT2	165	可设置	0x0000_02D4	SDF DR 中断 2
SDF DR INT3	166	可设置	0x0000_02D8	SDF DR 中断 3
SDF DR INT4	167	可设置	0x0000_02DC	SDF DR 中断 4
-	168	-	0x0000_02E0	保留
-	169	-	0x0000_02E4	保留
-	170	-	0x0000_02E8	保留
-	171	-	0x0000_02EC	保留
-	172	-	0x0000_02F0	保留
-	173	-	0x0000_02F4	保留
-	174	-	0x0000_02F8	保留
-	175	-	0x0000_02FC	保留
-	176	-	0x0000_0300	保留
-	177	-	0x0000_0304	保留
-	178	-	0x0000_0308	保留
-	179	-	0x0000_030C	保留
-	180	-	0x0000_0310	保留
-	181	-	0x0000_0314	保留
-	182	-	0x0000_0318	保留
-	183	-	0x0000_031C	保留
LINA INT1	184	可设置	0x0000_0320	LINA 中断 1
LINA INT2	185	可设置	0x0000_0324	LINA 中断 2
-	186	-	0x0000_0328	保留
-	187	-	0x0000_032C	保留
PMBUSA interrupt	188	可设置	0x0000_0330	PMBUSA 中断
-	189	-	0x0000_0334	保留
-	190	-	0x0000_0338	保留
-	191	-	0x0000_033C	保留
-	192	-	0x0000_0340	保留
-	193	-	0x0000_0344	保留
-	194	-	0x0000_0348	保留
-	195	-	0x0000_034C	保留
-	196	-	0x0000_0350	保留

异常类型	向量编号	优先级	向量地址	描述
-	197	-	0x0000_0354	保留
-	198	-	0x0000_0358	保留
-	199	-	0x0000_035C	保留
ADCC EVENT	200	可设置	0x0000_0360	ADCC EVENT 中断
ADCC2	201	可设置	0x0000_0364	ADCC2 中断
ADCC3	202	可设置	0x0000_0368	ADCC3 中断
ADCC4	203	可设置	0x0000_036C	ADCC4 中断
-	204	-	0x0000_0370	保留
-	205	-	0x0000_0374	保留
-	206	-	0x0000_0378	保留
-	207	-	0x0000_037C	保留
-	208	-	0x0000_0380	保留
-	209	-	0x0000_0384	保留
-	210	-	0x0000_0388	保留
-	211	-	0x0000_038C	保留
-	212	-	0x0000_0390	保留
-	213	-	0x0000_0394	保留
-	214	-	0x0000_0398	保留
-	215	-	0x0000_039C	保留
-	216	-	0x0000_03A0	保留
-	217	-	0x0000_03A4	保留
FLASH_ECC_ERR	218	可设置	0x0000_03A8	FLASH ECC 中断
-	219	-	0x0000_03AC	保留
SYS_PLL_SLIP	220	可设置-	0x0000_03B0	PLL SLIP 中断
-	221	-	0x0000_03B4	保留
-	222	-	0x0000_03B8	保留
-	223	-	0x0000_03BC	保留
-	224	-	0x0000_03C0	保留
CTI INT0	225	可设置	0x0000_03C4	CTI 中断 0
CTI INT1	226	可设置	0x0000_03C8	CTI 中断 1

14 外部中断与事件控制器（EXTI）

14.1 简介

中断/事件分为内部中断/事件、外部中断/事件。在该手册中，外部中断指从 I/O 引脚输入信号引起的中断/事件，在中断向量表中指 EXTIx；其它中断指内部中断/事件，这些事件线可将器件从 IDEL 模式唤醒。

14.2 主要特征

（1） 支持 16 个事件/中断请求

（2） 可独立配置作为外部/内部事件请求的线

（3） 每个事件/中断线都可独立屏蔽

（4） 可独立配置有效边沿

（5） 每个外部事件/中断线都可独立触发

（6） 每个外部中断线都有专用的状态位

（7） 可配置 IO 触发选择

（8） 仿真所有外部事件中断

14.3 功能描述

14.3.1 “外部中断与事件”类别及差异点

“外部中断与事件”按照触发源、配置及执行过程，可分为：外部硬件中断、外部硬件事件、外部软件事件、外部软件中断，差异点见下表：

表格 41“外部中断与事件”分类及差异点

名称	触发源	配置及执行过程
外部硬件中断	外部信号	（1）设置触发方式，允许中断请求，使能对应外设中断线（在 NVIC 中使能）； （2）当外部中断线上产生了和配置一致的边沿时，产生中断请求，对应的挂起位被置 1，在挂起寄存器对应位写 1，将清除该中断请求。
外部硬件事件	外部信号	（1）设置触发方式，使能事件线； （2）当外部事件线上产生了和配置一致的边沿时，产生 1 个事件请求脉冲，对应的挂起位不被置 1。

名称	触发源	配置及执行过程
外部软件事件	软件中断寄存器/发送事件(SEV)指令	（1）使能事件线； （2）对应事件线的软件中断事件寄存器写 1，产生 1 个事件请求脉冲，对应的挂起位不被置 1。
外部软件中断	软件中断寄存器	（1）允许中断请求，使能对应外设中断线（在 NVIC 中使能）； （2）对应中断线的软件中断事件寄存器写 1，产生中断请求，对应的挂起位被置 1，在挂起寄存器对应位写 1，将清除该中断请求。

14.3.2 内核唤醒

使用 WFI、WFE 指令都可以使内核停止工作。使用 WFI 指令，任一中断可唤醒内核；使用 WFE 指令，需通过事件唤醒。

使用中断唤醒，会触发中断处理函数，正常的中断配置即可唤醒内核。使用事件唤醒内核不触发中断处理函数，会减少唤醒时间，配置方法是：

（1） 触发内部中断（内部硬件事件）但不触发中断处理函数唤醒

- 使能外设中的一个内部中断，但不使能 NVIC 中对应的中断，避免触发中断处理函数
- 在内核的系统控制器中使能 SEVONPEND 位，执行 WFE 指令使内核进入睡眠
- 产生中断唤醒内核，当内核从 WFE 恢复后，需要清除相应外设的中断挂起位和外设 NVIC 中断通道挂起位（在 NVIC 中断清除挂起寄存器中）

（2） 通过 EXTI 线事件（外部硬件事件）唤醒

- 配置 EXTI 线为事件模式
- 执行 WFE 指令使内核进入睡眠
- 产生中断唤醒内核，CPU 从 WFE 恢复后，因为对应事件线的挂起位没有被置位，不必清除相应外设的中断挂起位或 NVIC 中断通道挂起位

14.3.3 外部中断与事件线映射

表格 42 外部中断与事件线映射

外部中断与事件通道名称	外部中断与事件线编号	是否可以触发 DMA	说明
GPIO0-GPIO14	EXTI 0	否	GPIO0~GPIO14 的 IO 触发
GPIO15-GPIO29	EXTI 1	否	GPIO15~GPIO29 的 IO 触发
GPIO30-GPIO44	EXTI 2	否	GPIO30~GPIO44 的 IO 触发
GPIO45-GPIO59	EXTI 3	否	GPIO45~GPIO59 的 IO 触发
INPUTXBAR4	EXTI 4	是	INPUTXBAR4 的信号触发
INPUTXBAR5	EXTI 5	是	INPUTXBAR5 的信号触发
INPUTXBAR6	EXTI 6	是	INPUTXBAR6 的信号触发

外部中断与事件通道名称	外部中断与事件线编号	是否可以触发 DMA	说明
INPUTXBAR13	EXTI 7	是	INPUTXBAR13 的信号触发
INPUTXBAR14	EXTI 8	是	INPUTXBAR14 的信号触发
COMP1	EXTI 9	否	COMP1.CTRIPH_OR_CTRIPL 的信号触发
COMP2	EXTI 10	否	COMP2.CTRIPH_OR_CTRIPL 的信号触发
…	…	…	…
COMP7	EXTI 15	否	COMP7.CTRIPH_OR_CTRIPL 的信号触发

14.4 寄存器组地址

表格 43 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
EXTIRegs	EXTI_REGS	0x5011 1000	0x5011 13FF

14.5 寄存器地址映射

表格 44 外部中断/事件控制器寄存器地址映射

寄存器名	描述	偏移地址	WRPRT
EXTI_RTEN	上升沿触发选择寄存器	0x00	-
EXTI_FTEN	下降沿触发选择寄存器	0x04	-
EXTI_SWINTE	软件中断事件寄存器	0x08	-
EXTI_GPIOSEL	选择 GPIO 触发信号	0x0C	-
EXTI_IMASK0	CPU0 中断屏蔽寄存器	0x10	-
EXTI_EMASK0	CPU0 事件屏蔽寄存器	0x14	-
EXTI_IPEND0	CPU0 中断挂起寄存器	0x18	-
EXTI_IMASK1	CPU1 中断屏蔽寄存器	0x20	-
EXTI_EMASK1	CPU1 事件屏蔽寄存器	0x24	-
EXTI_IPEND1	CPU1 中断挂起寄存器	0x28	-
EXTI_INT4CNT	中断线 4 计数器	0x30	-
EXTI_INT5CNT	中断线 5 计数器	0x34	-
EXTI_INT6CNT	中断线 6 计数器	0x38	-

14.6 寄存器功能描述

14.6.1 使能上升沿触发选择寄存器（EXTI_RTEN）

偏移地址：0x00

复位类型：

位/域	名称	R/W	描述	复位值
15:0	RTENx	R/W	使能线 x 上的上升沿触发事件和中断（Rising Trigger Event Enable and Interrupt of Line x）（x=0-15） 0：禁止 1：使能	0h
31:16	保留			0h

注意：由于外部唤醒线是边沿触发，所以这些线上不能有毛刺信号；在写 EXTI_RTEN 寄存器时，若上升沿信号在外部中断线上则不能被识别，置位挂起位也不会置位；在同一中断线上，上升沿触发和下降沿触发可以同时被设置。

14.6.2 使能下降沿触发选择寄存器（EXTI_FTEN）

偏移地址：0x04

复位类型：

位/域	名称	R/W	描述	复位值
15:0	FTENx	R/W	使能线 x 上的下降沿触发事件和中断（Falling Trigger Event Enable and Interrupt of Line x）（x=0-15） 0：禁止 1：使能	0h
31:16	保留			0h

注意：由于外部唤醒线是边沿触发，所以这些线上不能有毛刺信号；在写 EXTI_FTEN 寄存器时，若上升沿信号在外部中断线上则不能被识别，置位挂起位也不会置位；在同一中断线上，上升沿触发和下降沿触发可以同时被设置。

14.6.3 软件中断事件寄存器（EXTI_SWINTE）

偏移地址：0x08

复位类型：

位/域	名称	R/W	描述	复位值
15:0	SWINTEx	R/W	线 x 上的软件中断（Software Interrupt Event on Line x）（x=0-15） 当该位为 1 时，写入 1 则无作用。 当该位为 0 时，写 1 将置位 EXTI_IPEND 的挂起位。若置位 EXTI_IMASK（EXTI_EMASK）开放中断（事件）请求，则此时将产生一个中断（事件）。 0：清除软件中断（事件） 1：产生软件中断（事件）	0h
31:16	保留			0h

14.6.4 选择 GPIO 触发信号寄存器（EXTI_GPIOSEL）

偏移地址：0x0C

复位类型：

位/域	名称	R/W	描述	复位值
3:0	GPIOSEL0	R/W	选择 GPIO 触发信号 0（GPIO Trigger signal select 0） 0000：GPIO0 0001：GPIO1 … 1110：GPIO14 1111：保留	0h
7:4	GPIOSEL1	R/W	选择 GPIO 触发信号 1（GPIO Trigger signal select 1） 0000：GPIO15 0001：GPIO16 … 1110：GPIO29 1111：保留	0h
11:8	GPIOSEL2	R/W	选择 GPIO 触发信号 2（GPIO Trigger signal select 2） 0000：GPIO30 0001：GPIO31 … 1110：GPIO44 1111：保留	0h
15:12	GPIOSEL3	R/W	选择 GPIO 触发信号 3（GPIO Trigger signal select 3） 0000：GPIO45 0001：GPIO46 … 1110：GPIO59 1111：保留	0h
31:16	保留			0h

14.6.5 CPU0 中断屏蔽寄存器（EXTI_IMASK0）

偏移地址：0x10

复位类型：

位/域	名称	R/W	描述	复位值
15:0	IMASKx	R/W	屏蔽线 x 上的中断请求（Interrupt Request Mask on Line x）（x=0-15） 0：屏蔽 1：开放	0h
31:16	保留			0h

14.6.6 CPU0 事件屏蔽寄存器（EXTI_EMASK0）

偏移地址：0x14

复位类型：

位/域	名称	R/W	描述	复位值
15:0	EMASKx	R/W	屏蔽线 x 上的事件请求（Event Request Mask on Line x）（x=0-15） 0：屏蔽 1：开放	0h
31:16	保留			0h

14.6.7 CPU0 中断挂起寄存器（EXTI_IPEND0）

偏移地址：0x18

复位类型：

位/域	名称	R/W	描述	复位值
15:0	IPENDx	RC_W1	发生线 x 上的中断挂起标志（Interrupt Pending Occur of Line x Flag）（x=0-15） 是否发生可选择的触发请求 0：无 1：发生 当发生软件触发中断或者在外部中断线上发生了 EXTI_RTEN/EXTI_FTEN 对应的边沿触发请求时，由硬件置 1；可通过改变边沿检测的极性清 0，或通过向该位写入 1 清 0。	0h
31:16	保留			0h

14.6.8 CPU1 中断屏蔽寄存器（EXTI_IMASK1）

偏移地址：0x20

复位类型：

位/域	名称	R/W	描述	复位值
15:0	IMASKx	R/W	屏蔽线 x 上的中断请求（Interrupt Request Mask on Line x）（x=0-15） 0：屏蔽 1：开放	0h
31:16	保留			0h

14.6.9 CPU1 事件屏蔽寄存器（EXTI_EMASK1）

偏移地址：0x24

复位类型：

位/域	名称	R/W	描述	复位值
15:0	EMASKx	R/W	屏蔽线 x 上的事件请求（Event Request Mask on Line x）（x=0-15） 0：屏蔽 1：开放	0h
31:16	保留			0h

14.6.10 CPU1 中断挂起寄存器（EXTI_IPEND1）

偏移地址：0x28

复位类型：

位/域	名称	R/W	描述	复位值
15:0	IPENDx	RC_W1	发生线 x 上的中断挂起标志（Interrupt Pending Occur of Line x Flag）（x=0-15） 是否发生可选择的触发请求 0：无 1：发生 当发生软件触发中断或者在外部中断线发生了 EXTI_RTEN/EXTI_FTEN 对应的边沿触发请求时，由硬件置 1；可通过改变边沿检测的极性清 0，或通过向该位写入 1 清 0。	0h
31:16	保留			0h

14.6.11 中断线 4 计数器（EXTI_INT4CNT）

偏移地址：0x30

复位类型：

位/域	名称	R/W	描述	复位值
15:0	INT4CNT	R	中断线 4 计数器（Interrupt Line 4 Counter） 产生有效的中断或系统复位可以使计数器复位。 当 CPU0 或 CPU1 使能 EXTI 4 中断时开始以 SYSCLK 的时钟频率进行向上计数，计数最大值为 0xFFFF。当使能中断并且达到 0xFFFF 后，计数器将重新开始计数，直到 EXTI 4 产生中断时该计数器被清 0。 当 CPU0 和 CPU1 都失能 EXTI 4 中断时计数器将停止计数。	0h
31:16	保留			0h

14.6.12 中断线 5 计数器（EXTI_INT5CNT）

偏移地址：0x34

复位类型：

位/域	名称	R/W	描述	复位值
15:0	INT5CNT	R	中断线 5 计数器（Interrupt Line 5 Counter） 产生有效的中断或系统复位可以使计数器复位。 当 CPU0 或 CPU1 使能 EXTI 5 中断时开始以 SYSCLK 的时钟频率进行向上计数，计数最大值为 0xFFFF。当使能中断并且达到 0xFFFF 后，计数器将重新开始计数，直到 EXTI 5 产生中断时该计数器被清 0。 当 CPU0 和 CPU1 都失能 EXTI 5 中断时计数器将停止计数。	0h
31:16	保留			0h

14.6.13 中断线 6 计数器（EXTI_INT6CNT）

偏移地址：0x38

复位类型：

位/域	名称	R/W	描述	复位值
15:0	INT6CNT	R	中断线 6 计数器（Interrupt Line 6 Counter） 产生有效的中断或系统复位可以使计数器复位。 当 CPU0 或 CPU1 使能 EXTI 6 中断时开始以 SYSCLK 的时钟频率进行向上计数，计数最大值为 0xFFFF。当使能中断并且达到 0xFFFF 后，计数器将重新开始计数，直到 EXTI 6 产生中断时该计数器被清 0。 当 CPU0 和 CPU1 都失能 EXTI 6 中断时计数器将停止计数。	0h
31:16	保留			0h

15 双码安全（DCS）

本模块的描述请参见《DCS 用户手册》 。

16 可编程静态存储子系统（CFGSMS）

本模块的描述请参见《CFGSMS 用户手册》 。

17 非易失性存储器控制器（NVMC）

17.1 简介

CPU0 和 CPU1 都可以通过 ITCM -> FACC -> Flash 和 CPU CACHE-> C-Bus -> busmatrix -> Flash 两条路径读取 Flash 中的内容；FACC 用于加速 ITCM 的访问，CPU CACHE 用于加速 C-BUS 的访问；两条路径访问的物理存储空间相同，使用的地址不相同。

所有 Flash 的擦/写操作都需要通过调用 Flash API 完成，Flash MEM 区地址对于 CPU0/1，DMA 只可读，所有总线写操作将被忽视。擦/写和读请求同时发起时，擦写优先级高于读优先级。

当各接口同时发起读访问时，Flash 响应读访问优先级为：

（1） CPU0 ITCM ICODE 访问

（2） CPU1 ITCM ICODE 访问

（3） CPU0 ITCM DCODE 访问

（4） CPU1 ITCM DCODE 访问

（5） BUS Interface0 访问

（6） BUS Interface1 访问

CPU0_ITCM，CPU1_ITCM，BUS Interface0/1 可以同时访问，用户可根据实际情况自己灵活分配哪个 CPU 通过哪个接口访问哪块 Flash，以免发生资源竞争。

通过 CBUS 访问时：

0x08XX_XXXX 的 Flash 地址空间通过 BUS Interface0 访问。

0x09XX_XXXX 的 Flash 地址空间通过 BUS Interface1 访问。

CPU0/1 CBUS 访问通过 busmtrix 仲裁后给到 BUS Interface0 和 BUS Interface1；CPU0 CBUS 优先级高于 CPU1 CBUS。

17.2 结构框图

图 8 结构框图

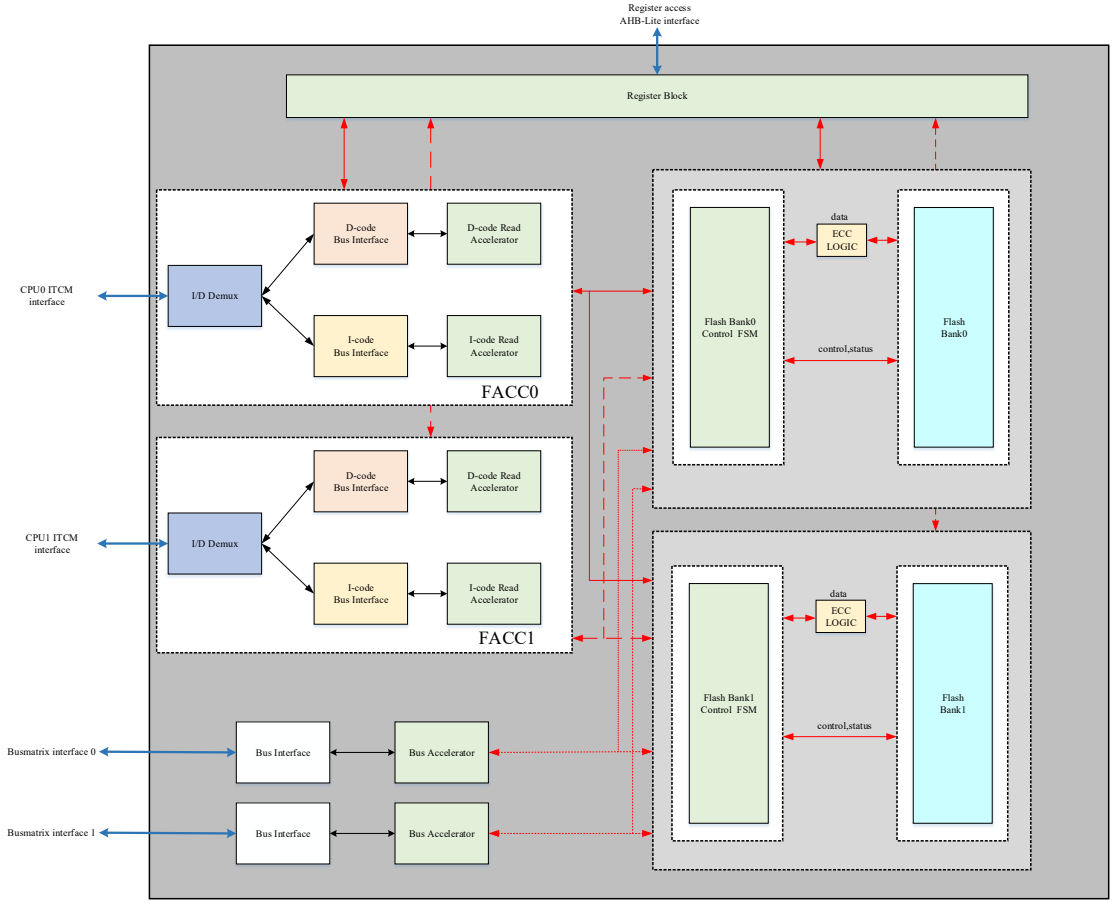
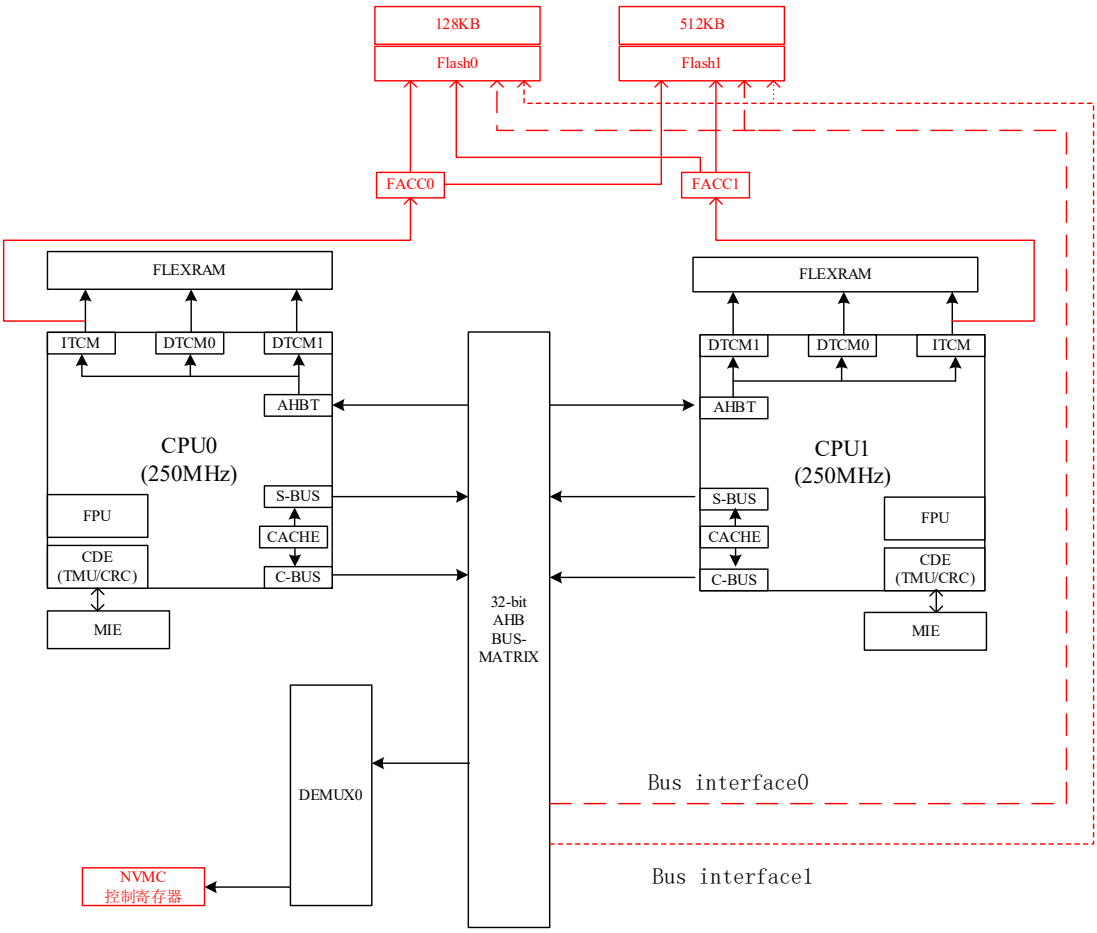


图 9 系统架构内部的块链接



17.3 主要特征

（1） Flash 支持单双 bank 切换，位宽可变

- 单 bank 模式下：nDBANK = 1；Flash 位宽 256bits
- 双 bank 模式下：nDBANK = 0；Flash 位宽 128bits

（2） 读写并行（仅双 bank 模式）

（3） 读并行（仅双 bank 模式）.

（4） FACC 加速：用于加速 CPU0/1 通过 ITCM 去访问 Flash，包括 FACC 预取和 FACC cache，其中 FACC 预取 buffer 一个，1x128/256bit；FACC cache 包括 64x128/256 bits 的指令分支缓存和 8x128/256bits 的数据缓存（取决于读位宽）。

（5） busmatrix 预取：用于加速 CPU0/1 通过 busmatrix 去访问 Flash，两个预取 buffer 进行乒乓预取，buffer 位宽为 128/256bit（取决于位宽）。

（6） 支持带 ECC 的 64/128 位程序

（7） 支持带 ECC 的 256 位程序（单 bank 模式 bank0&1 mix 区域）

（8） 支持 Flash 扇区擦除，扇区大小 16KB（单 bank 模式 bank0&1 mix 区域）

（9） 支持 Flash 扇区擦除，扇区大小 8KB（双 bank 模式；或者单 bank 模式 bank1 专属区域）

（10） 支持 OTP 区域，DCS 相关 OTP 配置可以被编程一次；OTP 扇区不能被擦除。

（11） 支持用户选项字节，可以配置 Flash 单/双 bank 模式；CPU boot 方式与启动地址；CFGSMS 空间分配。

（12） 支持 ECC
- SECDED- FMC 支持单纠错和双纠错
- 地址位包含在 ECC 中
- 测试模式用于检查 ECC 逻辑的健康状况

（13） Flash 支持低功耗模式

（14） Flash 擦写均通过调用相关 API 函数完成。

17.4 功能描述

17.4.1 Flash memory 组织

CPU0 和 CPU1 都可以通过 ITCM/C-BUS 两种方式访问 Flash MEM，两条路径访问的物理存储空间相同，但使用的地址不相同。

（1） 主要块
- Total (8K+32K)x128bit(128K+512K Byte)

（2） IFREN 块
- Total (2K+1K)x128bit(32K+16K Byte)
- 512x128bit(8K Byte) user option byte
- 512x128bit(8K Byte) OTP

（3） Redundancy 块
- Total 1Kx128bit(16K Byte)

（4） ECC
- Total 43Kx16bit(86K Byte)
- Half-word access only

表格 45 存储器单组组织(256 位读宽)

IP	名称	基于 C-BUS 接口的扇区	基于 ITCM 接口的扇区	扇区大小	Bank number
主存储器	Sector0	0x0800_0000~0x0800_3FFF	0x0010_0000~0x0010_3FFF	16KByte	Bank0 & Bank1 mix
	Sector1	0x0800_4000~0x0800_7FFF	0x0010_4000~0x0010_7FFF	16KByte	Bank0 & Bank1 mix
	Sector2	0x0800_8000~0x0800_BFFF	0x0010_8000~0x0010_BFFF	16KByte	Bank0 & Bank1 mix
					Bank0 & Bank1 mix
	Sector14	0x0803_8000~0x0803_BFFF	0x0013_8000~0x0013_BFFF	16KByte	Bank0 & Bank1 mix
	Sector15	0x0803_C000~0x0803_FFFF	0x0013_C000~0x0013_FFFF	16KByte	Bank0 & Bank1 mix
	Sector16	0x0804_0000~0x0804_1FFF	0x0014_0000~0x0014_1FFF	8KByte	Bank1
					Bank1
	Sector63	0x0809_E000~0x0809_FFFF	0x0019_E000~0x0019_FFFF	8KByte	Bank1
IFREN	Bank0 back-up sector0	0x0810_0000~0x0810_1FFF	0x0008_0000~0x0008_1FFF	8Kbyte	Bank0
IFREN	Bank0 back-up sector1	0x0810_2000~0x0810_3FFF	0x0008_2000~0x0008_3FFF	8Kbyte	Bank0
IFREN	Bank0 back-up sector2	0x0810_4000~0x0810_5FFF	0x0008_4000~0x0008_5FFF	8KByte	Bank0
IFREN	Bank0 back-up sector3	0x0810_6000~0x0810_7FFF	0x0008_6000~0x0008_7FFF	8KByte	Bank0
IFREN	Bank1 User Option byte	0x0810_8000~0x0810_9FFF	0x0008_8000~0x0008_9FFF	8Kbyte	Bank1

IP	名称	基于 C-BUS 接口的扇区	基于 ITCM 接口的扇区	扇区大小	Bank number
IFREN	Bank1 OTP	0x0810_A000~0x0810_BFFF	0x0008_A000~0x0008_BFFF	8Kbyte	Bank1
Main memory	ECC	0x0900_0000~0x0901_3FFF	NA	80Kbyte (半字访问)	Bank0 & Bank1 mix
IFREN	ECC	0x0902_0000~0x0902_17FF	NA	6Kbyte (半字访问)	Bank0/1

表格 46 存储器双组组织(128 位读宽)

IP	名称	基于 C-BUS 接口的扇区	基于 ITCM 接口的扇区	扇区大小	Bank number
Main memory bank0	Sector0	0x0800_0000~0x0800_1FFF	0x0010_0000~0x0010_1FFF	8KByte	Bank0
	Sector1	0x0800_2000~0x0800_3FFF	0x0010_2000~0x0010_3FFF	8KByte	Bank0
	Sector2	0x0800_4000~0x0800_5FFF	0x0010_4000~0x0010_5FFF	8KByte	Bank0
					Bank0
	Sector14	0x0801_C000~0x0801_DFFF	0x0011_C000~0x0011_DFFF	8KByte	Bank0
	Sector15	0x0801_E000~0x0801_FFFF	0x0011_E000~0x0011_FFFF	8KByte	Bank0
Main memory bank1	Sector16	0x0802_0000~0x0802_1FFF	0x0012_0000~0x0012_1FFF	8KByte	Bank1
	Sector17	0x0802_2000~0x0802_3FFF	0x0012_2000~0x0012_3FFF	8KByte	Bank1
	Sector18	0x0802_4000~0x0802_5FFF	0x0012_4000~0x0012_5FFF	8Kbyte1	Bank1
					Bank1
	Sector30	0x0803_C000~0x0803_DFFF	0x0013_C000~0x0013_DFFF	8KByte	Bank1
	Sector31	0x0803_E000~0x0803_FFFF	0x0013_E000~0x0013_FFFF	8KByte	Bank1
	Sector32	0x0804_0000~0x0804_1FFF	0x0014_0000~0x0014_1FFF	8KByte	Bank1
					Bank1
	Sector79	0x0809_E000~0x0809_FFFF	0x0019_E000~0x0019_FFFF	8KByte	Bank1
IFREN	Bank0 back-up sector0	0x0810_0000~0x0810_1FFF	0x0008_0000~0x0008_1FFF	8Kbyte	Bank0
IFREN	Bank0 back-up sector1	0x0810_2000~0x0810_3FFF	0x0008_2000~0x0008_3FFF	8Kbyte	Bank0
IFREN	Bank0 back-up sector2	0x0810_4000~0x0810_5FFF	0x0008_4000~0x0008_5FFF	8KByte	Bank0
IFREN	Bank0 back-up sector3	0x0810_6000~0x0810_7FFF	0x0008_6000~0x0008_7FFF	8KByte	Bank0

IP	名称	基于 C-BUS 接口的扇区	基于 ITCM 接口的扇区	扇区大小	Bank number
IFREN	Bank1 User Option byte	0x0810_8000~0x0810_9FFF	0x0008_8000~0x0008_9FFF	8Kbyte	Bank1
IFREN	Bank1 OTP	0x0810_A000~0x0810_BFFF	0x0008_A000~0x0008_BFFF	8Kbyte	Bank1
IFREN1	Flash trimming	0x0818_0000~0x0818_3FFF	0x0009_0000~0x0009_3FFF	16KByte	Bank0 & Bank1 mix
Main memory	ECC	0x0900_0000~0x0901_3FFF	NA	80Kbyte (半字访问)	Bank0 & Bank1 mix
IFREN	ECC	0x0902_0000~0x0902_17FF	NA	6Kbyte (半字访问)	Bank0/1

17.4.2 DCS OTP

0x0810_A000~0x0810_A7FF 与 0x0810_B000~0x0810_B7FF 是 DCS OTP 区域，该区域可以编程。

DCS 的 LINKPOINTER 区域：

0x0810_A000~0x0810_A017，0x0810_A400~0x0810_A417；

0x0810_B000~0x0810_B017，0x0810_B400~0x0810_B417；

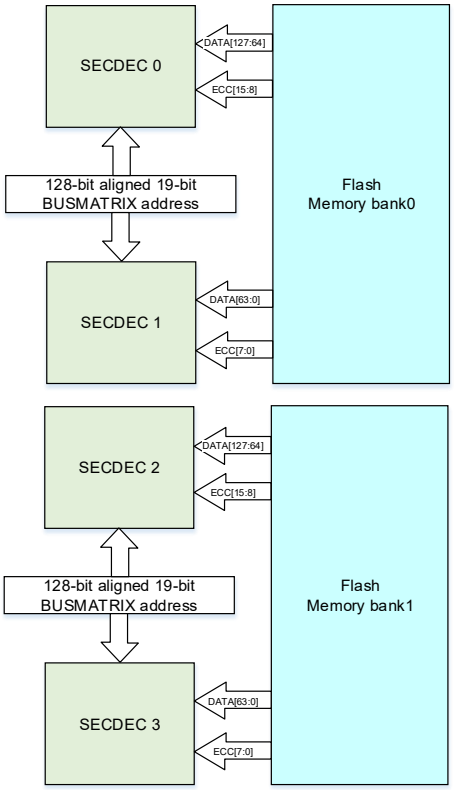
该区域可以多次编程（将 1 编程为 0），且不受 ECC 保护。可以调用 API 进行编程。

DCS 的其余 OTP 受 ECC 保护，只可以编程一次，可以调用 API 进行编程。

17.4.3 Flash ECC

本芯片中因需 2 块 Flash 并行读取数据，使用了 4 份 SECDED ECC 硬件校验模块。

图 10Flash ECC 解码框图

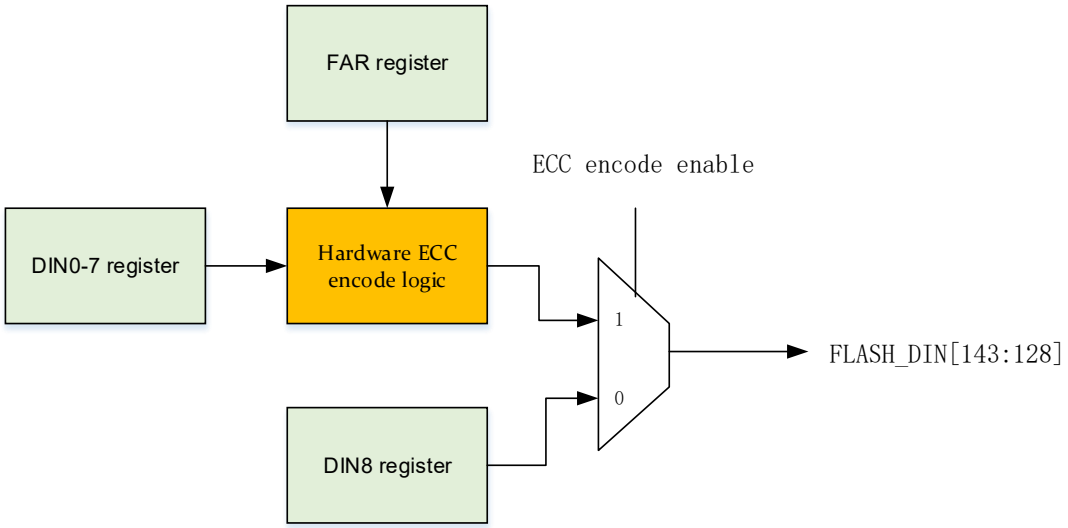


通过 ECC_ENABLE.ENCODE_ENABLE 来选择使用硬件产生 ECC 或用户自定义 ECC，硬件结构如下图。

当 ENCODE_ENABLE 值为 0xA 时，硬件 ECC 编码模块根据 DIN0-7 与 FAR 寄存器中的值计算相应的 ECC 值，并在编程时存入 Flash 中。

当 ENCODE_ENABLE 值不为 0xA 时，可通过隐藏寄存器 DIN8 设置需要写入 Flash 中的 ECC 值。编程时，DIN8[15:0]将写入 Flash bank0 的 ECC 区，DIN8[31:16]将写入 Flash bank1 的 ECC 区。

图 11Flash ECC 编码框图



通过地址直接访问 Flash 主存储器区存放的 ECC 校验值。可用于直接访问 ECC 的地址区间。

通过 Flash ECC 区域，即 0x0900_0000~0x0901_FFFF 地址区间，可以以 half-word 的方式直接读取主存储器中存放的 ECC 校验值。

每次读出的数据为 16bits，对应的是主存储器中所读地址相对 0x0900_0000 的偏移*8 处的 ECC 值。即(0x0900_000 + offset)处读出的是(0x0800_0000 + offset*8)处数据的 ECC 校验值。

对 main 区，back-up 区，user option byte 区、user OTP 区，具体映射关系如下：

表格 47 Flash ECC 地址映射关系表

IP	名称	总线上的地址范围	ITCM 上的地址范围	对应的 ECC 地址 (半字访问)
Main memory	Main Sectors	0x0800_0000~0x080A_0000	0x0010_0000~0x001A_0000	0x0900_0000~0x0901_4000
IFREN	Bank0 back-up sector0	0x0810_0000~0x0810_1FFF	0x0008_0000~0x0008_1FFF	0x0902_0000~0x0902_0400
IFREN	Bank0 back-up sector1	0x0810_2000~0x0810_3FFF	0x0008_2000~0x0008_3FFF	0x0902_0400~0x0902_0800
IFREN	Bank0 back-up sector2	0x0810_4000~0x0810_5FFF	0x0008_4000~0x0008_5FFF	0x0902_0800~0x0902_0c00
IFREN	Bank0 back-up sector3	0x0810_6000~0x0810_7FFF	0x0008_6000~0x0008_7FFF	0x0902_0c00~0x0902_1000
IFREN	Bank1 User Option byte	0x0810_8000~0x0810_9FFF	0x0008_8000~0x0008_9FFF	0x0902_1000~0x0902_1400
IFREN	Bank1 User OTP	0x0810_A000~0x0810_BFFF	0x0008_A000~0x0008_BFFF	0x0902_1400~0x0902_1800

Flash ECC 出错时将产生两种中断：Flash_ecc_correctable_int 与 Flash_ecc_uncorrectable_int。其中 Flash_ecc_uncorrectable_int 连接 CPU NMI，Flash_ecc_correctable_int 连接 CPU NVIC。

17.4.4 Flash 低功耗

（1） Bank 电源模式支持待机和活跃模式

（2） Charge Pump 电源模式仅支持活跃模式（Flash 不会掉电）

（3） Bank 待机到活跃的时间 = 65 INTOSC1 clock(10MHz) cycles

表格 48 Bank 电源模式

Flash 状态	描述
standby	Flash standby 状态，最低功耗
active	正常模式

进入低功耗流程：

- 配置 FBAC 寄存器，配置活跃状态能保持的周期时间数。
- 配置 FBFALLBACK 寄存器，配置 Flash bank 要进入的状态。
- 停止访问 Flash(停止任何读写擦)。
- 当没访问 Flash 的时间长度达到 FBAC 寄存器中配置的周期数时，Flash 自动进入 FBFALLBACK 中配置的低功耗状态。
- 进入低功耗状态后，对 Flash 的任意一次访问将会唤醒 Flash。

17.4.5 Flash 读操作

系统时钟频率与 Flash 读取时间之间的关系

为了准确读取 Flash 指令和数据，必须根据系统时钟 SYSCLK 频率去配置 FRDCNTL 寄存器中 RWAIT 来满足 Flash 读等待延时的需求。Flash 零等待读支持最大频率为 36MHz。

表格 49 系统频率的读取时间

系统频率	RWAIT
0<f<=36MHz	0
36MHz<f<=72MHz	1
72MHz<f<=108MHz	2
108MHz<f<=144MHz	3
144MHz<f<=180MHz	4
180MHz<f<=216MHz	5
216MHz<f<=250MHz	6

注意：

（1） 升高系统频率时，先将 RWAIT 配置到目标频率范围内，然后再配置系统频率升高。

（2） 降低系统频率时，先配置系统频率将时钟降低后，然后再将 RWAIT 配置到目标频率范围内。

ITCM 访问接口

ITCM 访问时，通过 FACC 进行读取加速，Flash 控制器会将访问请求区分为指令（ICODE）访问和数据（DCODE）访问。为了发挥处理器的全部性能，在 ICODE 处理上该加速器将实施指令预取队列和分支缓存；对于 DCODE 提供数据 Cache 缓存。

FACC 加速包括 FACC 预取和 FACC cache，其中 FACC 预取 buffer 一个，1x128/256 位；FACC cache 包括 64x128/256 位的指令分支缓存和 8x128/256 位的数据缓存。

FACC 预取机制：每个 Flash 读操作可读取 128 位（双 bank 模式），可以是 4 行 32 位指令，也可以是 8 行 16 位指令，具体取决于烧录在 Flash 中的程序。因此对于顺序执行的代码，至少需要 4 个 CPU 周期来执行前一次读取的 128 位指令行。在 CPU 请求当前指令行时，可使用 ICode 总线的预取操作读取 Flash 中的下一个连续存放的 128 位指令行。（如果是单 bank 模式则是 256 位）

指令分支缓存：为了减少因指令跳转而产生的时间损耗，可将 64 行 128 位（如果是单 bank 模式则是 256 位）的指令保存到指令缓存存储器中。每当出现指令缺失（即请求的指令未存在于当前使用的指令行，预取指令行或指令缓存存储器中）时，系统会将新读取的行复制到指令缓存存储器中。如果 CPU 请求的指令已存在于指令缓存区中，则无需任何延时即可立即获取。指令缓存存储器存满后，可采用 LRU 策略确定指令缓存存储器中待替换的指令行。此特性非常适用于包含循环的代码。

数据缓存：如果频繁使用某些数据，可将数据缓存使能，此特性的工作原理与指令缓存存储器类似，但保留的数据大小限制在 8 行 128 位（如果是单 bank 模式则是 256 位）。

CBUS 访问接口

CBUS 访问由于 CPU 内部已经有 Cache 了，因此在 Flash 控制器接口上只有预取加速。总共两个 buffer 进行轮流乒乓预取，每个 buffer 的位宽为 128 位（如果是单 bank 模式则是 256 位）。

17.4.6 Flash 上电初始化流程

上电初始化流程由硬件模块来完成，包括以下步骤：

（1） 上电复位释放，时钟稳定

（2） Flash 开始加载自身的 trim，使 Flash 处于可靠的状态

（3） Flash 开始加载 DCS linkpointer，寻找 DCS 配置起始地址

（4） Flash 开始加载 DCS 配置

（5） Flash 开始加载用户选项字节

（6） Flash 开始加载 efuse 和芯片系统配置

（7） 所有配置加载完成，释放系统复位

17.5 寄存器组地址

表格 50 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
NVMCRegs	NVMC_REGS	0x5001 0000	0x5001 065C

17.6 寄存器地址映射

表格 51 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
FRDCNTL	Flash 读控制寄存器	0x00	√
FBAC	Flash Bank 访问控制寄存器	0x3C	√
FBFALLBACK	Flash Bank 回退功率寄存器	0x40	√
FBPRDY	Flash Bank 泵就绪寄存器	0x44	√
FMSTAT	Flash 模块状态寄存器	0x54	√
FRD_INTF_CTRL	Flash 读接口控制寄存器	0x300	√
ECC_ENABLE	ECC_ENABLE 寄存器	0x600	√
SINGLE_ERR_ADDR_LOW_0	Bank0 单错误地址低寄存器	0x604	√
SINGLE_ERR_ADDR_HIGH_0	Bank0 单错误地址高寄存器	0x608	√
UNC_ERR_ADDR_LOW_0	Bank0 不可纠正的错误地址低寄存器	0x60C	√
UNC_ERR_ADDR_HIGH_0	Bank0 不可纠正的错误地址高寄存器	0x610	√
ERR_STATUS	错误状态寄存器	0x614	√
ERR_POS	错误位置寄存器	0x618	√
ERR_STATUS_CLR	错误状态清除寄存器	0x61C	√
ERR_CNT	错误计数器寄存器	0x620	√
ERR_THRESHOLD	错误阈值寄存器	0x624	√
ERR_INTFLG	错误中断标志寄存器	0x628	√
ERR_INTCLR	清除错误中断寄存器	0x62C	√
FDATAH_TEST	数据高测试寄存器	0x630	√

寄存器名称	寄存器描述	偏移地址	WRPRT
FDATAL_TEST	数据低测试寄存器	0x634	√
FADDR_TEST	ECC 测试地址寄存器	0x638	√
FECC_TEST	ECC 测试 ECC 寄存器	0x63C	√
FECC_CTRL	ECC 控制寄存器	0x640	√
FOUH_TEST	测试数据输出高寄存器	0x644	√
FOUL_TEST	测试数据输出低寄存器	0x648	√
FECC_STATUS	ECC 状态寄存器	0x64C	√
SINGLE_ERR_ADDR_LOW_1	Bank1 单错误地址低寄存器	0x650	√
SINGLE_ERR_ADDR_HIGH_1	Bank1 单错误地址高寄存器	0x654	√
UNC_ERR_ADDR_LOW_1	Bank1 不可纠正的错误地址低寄存器	0x658	√
UNC_ERR_ADDR_HIGH_1	Bank1 不可纠正的错误地址高寄存器	0x65C	√

17.7 寄存器功能描述

17.7.1 Flash 读控制寄存器（FRDCNTL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	保留			0h
15:8	RWAIT	R/W	Flash 读等待周期数（Flash read wait period） 可以设置为 0 到 0xF 之间的任意值。 对于 flash 访问，数据以 RWAIT+1 个 SYSCLK 周期返回。 0：0~36MHz 1：36MHz~72MHz 2：72MHz~108MHz: 3：108MHz~144MHz 4：144MHz~180MHz 5：180MHz~216MHz 6：216MHz~250MHz	Fh
31:16	保留			0h

17.7.2 Flash Bank 访问控制寄存器（FBAC）

偏移地址：0x3C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	保留			Fh

位/域	名称	R/W	描述	复位值
15:8	BAGP	R/W	Flash Bank 活动宽限期（Bank Active Grace Period） 这些位包含了 BAGP 递减计数器的起始计数值。对某个 bank 的任何访问都会导致其 BAGP 计数器重新加载到该 bank 的 BAGP 值。在最后一次访问该 flash 存储区后，递减计数器延迟 0~255 个预分频 SYSCLK 时钟周期，然后将该存储区置于 FBFALLBACK 寄存器所确定的回退功耗模式之一。在回退模式不是处于 Active 状态时，该值必须大于 1。 注意：用于 BAGP 递减计数器的预分频时钟是输入 SYSCLK 的 16 分频时钟。	0h
31:16	保留			0h

17.7.3 Flash Bank 回退功率寄存器（FBFALLBACK）

偏移地址：0x40

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	BNKPWR0	R/W	回退功耗模式 0（Fallback power mode 0） 00：Standby 01：Standby 10：Reserved 11：Active	0h
3:2	BNKPWR1	R/W	回退功耗模式 1（Fallback power mode 1） 00：Standby 01：Standby 10：Reserved 11：Active	0h
31:4	保留			0h

17.7.4 Flash Bank 泵就绪寄存器（FBPRDY）

偏移地址：0x44

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	BANK0RDY	R	Bank 0 准备就绪标志（Bank 0 Ready Flag） 允许软件在尝试访问之前确定 Bank 0 做好进行 Flash 访问的准备。需要在泵和 Bank 准备好之前尝试访问。 0：Bank0 未就绪 1：Bank0 处于活跃电源模式，准备接入。	0h
1	BANK1RDY	R	Bank 1 准备就绪标志（Bank 1 Ready Flag） 允许软件在尝试访问之前确定 Bank 1 是否做好进行 Flash 访问的准备。需要在泵和 Bank 准备好之前尝试访问。 0：Bank1 未就绪 1：Bank1 处于有功状态，准备接入	0h
14:2	保留			0h

位/域	名称	R/W	描述	复位值
15	PUMPRDY	R	泵准备就绪标志（Pump Ready Flag） 允许软件在尝试访问之前确定泵是否做好进行 Flash 访问的准备。如果泵未准备好时访问，则断言等待状态，直到泵准备好为止。 0：泵未就绪 1：泵准备就绪，处于有功状态	0h
31:16	保留			0h

17.7.5 Flash 状态寄存器 （FMSTAT）

偏移地址：0x54

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	PSUSP	R	当此位置位时，表示 Flash 模块已经接收并处理了一个程序挂起操作。在发出程序恢复指令或运行 Clear_More 指令之前，此位保持置位。	0h
2	ESUSP	R	当此位置位时，表示 Flash 模块已经接收并处理了一个程序擦除挂起操作。在发出擦除恢复指令或运行 Clear_More 指令之前，此位保持置位。	0h
3	保留			0h
4	CSTST	R	指令状态（Command Status）。 一旦 FSM stFACCs 错误时将置位 CSTST。置位 CSTST 时，此位通知主机编程、擦除或验证扇区指令失败，指令被停止。 由清除状态命令清除此位。对于某些错误，由于不属于其他错误位类型，此位将作为 FSM 错误的唯一指示。	0h
5	INVDAT	R	无效数据（Invalid Data）。 当此位置位时，表示用户试图对已经存在“0”写入一个“1”。	0h
6	PGM	R	编程活动（Program Active）。 当此位置位时，表示 Flash 模块当前正在执行编程操作。此位在对 stFACCs 编程时置 1，在编程完成时清 0。当编程暂停时，此位也会被清除；当编程恢复时，此位会被置位。	0h
7	ERS	R	擦除活动（Erase Active）。 此位置位时，表示 Flash 模块正在执行擦除操作。擦除 stFACCs 时此位被置 1，擦除完成后此位被清 0。当擦除暂停时，此位也被清除；当擦除恢复时，此位被置位。	0h
8	BUSY	R	此位置位时，表示正在处理编程、擦除或挂起操作。	0h
9	保留			0h

位/域	名称	R/W	描述	复位值
10	EV	R	擦除验证（Erase Verify）。 此位置位时，表示在为擦除操作给出了允许的最大擦除脉冲数后，没有成功擦除扇区。	0h
11	保留			0h
12	PGV	R	编程验证（Program verify）。 此位置位时，表示在为编程操作给出了允许的最大编程脉冲数后，没有成功编程字段。	0h
13	REG_NOT_WRITABLE	R	限制 NVMC 寄存器写入（Limit NVMC register write） 0：NVMC 寄存器可写入 1：NVMC 寄存器无法写入	0h
31:14	保留			0h

17.7.6 Flash 读接口控制寄存器（FRD_INTF_CTRL）

偏移地址：0x300

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	FACC0_prften	R/W	使能 CPU0 FACC 加速器预取（CPU0 FACC accelerator prefetch enable） 0：禁止 1：使能	0h
1	FACC0_icen	R/W	使能 CPU0 FACC 加速器 ICACHE（CPU0 FACC accelerator ICACHE enable） 0：禁止 1：使能	0h
2	FACC0_dcen	R/W	使能 CPU0 FACC 加速器 DCACHE（CPU0 FACC accelerator DCACHE enable） 0：禁止 1：使能	0h
3	FACC0_icrst	R/W	清除 CPU0 FACC 加速器 ICACHE（CPU0 FACC accelerator ICACHE clear） 0：无作用 1：清除	0h
4	FACC0_dcrst	R/W	清除 CPU0 FACC 加速器 DCACHE（CPU0 FACC accelerator DCACHE clear） 0：无作用 1：清除	0h
5	FACC1_prften	R/W	使能 CPU1 FACC 加速器预取（CPU1 FACC accelerator prefetch enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
6	FACC1_icen	R/W	使能 CPU1 FACC 加速器 ICACHE（CPU1 FACC accelerator ICACHE enable） 0：禁止 1：使能	0h
7	FACC1_dcen	R/W	使能 CPU1 FACC 加速器 DCACHE（CPU1 FACC accelerator DCACHE enable） 0：禁止 1：使能	0h
8	FACC1_icrst	R/W	清除 CPU1 FACC 加速器 ICACHE（CPU1 FACC accelerator ICACHE clear） 0：无作用 1：清除	0h
9	FACC1_dcrst	R/W	清除 CPU1 FACC 加速器 DCACHE（CPU1 FACC accelerator DCACHE clear） 0：无作用 1：清除	0h
10	Bus_prften	R/W	使能 busmatrix 接口预取（busmatrix interface prefetch enable） 0：禁止 1：使能	0h
11	Bus_buf_clr	R/W	清除 busmatrix 接口缓冲区（busmatrix interface buffer clear） 0：无作用 1：清除	0h
31:12	保留			0h

17.7.7 ECC_ENABLE 寄存器（ECC_ENABLE）

偏移地址：0x600

复位类型：SYSRSn

位/域	名称	R/W	说明	复位值
3:0	DECODE ENABLE	R/W	使能 ECC 解码（ECC decode enable）。 0xA 值将启用硬件 ECC 解码，其他任何值都会禁用 ECC	Ah
7:4	ENCODE ENABLE	R/W	使能 ECC 编码（ECC encode enable）。 0xA 值将启用硬件 ECC 编码，其他任何值都会禁用 ECC	Ah

注意：控制除 DCS LINKPOINTER OTP 外其余所有 Flash 区域的 ECC 使能关闭。

17.7.8 Bank0 单错误地址寄存器（SINGLE_ERR_ADDR_LOW_0）

偏移地址：0x604

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ERR_ADDR_L_0	R/W	在 128 位对齐 Flash bank0 存储器的低 64 位出现单个位错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,x50，结尾的地址属于 bank1。

17.7.9 Bank0 单错误地址寄存器（SINGLE_ERR_ADDR_HIGH_0）

偏移地址：0x608

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ERR_ADDR_H_0	R/W	在 128 位对齐 Flash bank0 存储器的高 64 位出现单个位错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.10 Bank0 不可纠正的错误地址低寄存器 （UNC_ERR_ADDR_LOW_0）

偏移地址：0x60C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	UNC_ADDR_L_0	R/W	在 128 位对齐 Flash bank0 存储器的低 64 位出现不可纠正错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.11 Bank0 不可纠正的错误地址高寄存器（UNC_ERR_ADDR_HIGH_0)

偏移地址：0x610

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	UNC_ADDR_H_0	R/W	在 128 位对齐 Flash bank0 存储器的高 64 位出现不可纠正错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.12 错误状态 Register （ERR_STATUS）

偏移地址：0x614

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	Flash0_FAIL_0_L	R	Fail on 0. 0：在 128 位对齐地址的 Flash bank0 低 64 位中未出现 Fail on 0 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的低 64 位中出现单个位错误，并且校正后值为 0。向 ERR_STATUS_CLR 寄存器的 Flash0_FAIL_0_L_CLR 位写 1 可清零。	0h
1	Flash0_FAIL_1_L	R	Fail on 1. 0：在 128 位对齐地址的 Flash bank0 低 64 位中未出现 Fail on 1 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的低 64 位中出现单个位错误，并且校正后值为 1。向 ERR_STATUS_CLR 寄	0h

位/域	名称	R/W	描述	复位值
			存器的 Flash0_FAIL_1_L_CLR 位写 1 可清零。	
2	Flash0_UNC_ERR_L	R	不可纠正的错误。 1：值为 1 表示 128 位对齐地址的 Flash bank0 低 64 位出现不可纠正的错误。向 ERR_STATUS_CLR 寄存器的 Flash0_UNC_ERR_L_CLR 位写 1 可清零。	0h
7:3	保留			0h
8	Flash0_FAIL_0_H	R	Fail on 0. 0：在 128 位对齐地址的 Flash bank0 高 64 位中未出现 Fail on 0 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的高 64 位中出现单个位错误，并且校正后值为 0。向 ERR_STATUS_CLR 寄存器的 Flash0_FAIL_0_H_CLR 位写 1 可清零。	0h
9	Flash0_FAIL_1_H	R	Fail on 1. 0：在 128 位对齐地址的 Flash bank0 高 64 位中未出现 Fail on 1 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的高 64 位中出现单个位错误，并且校正后值为 1。向 ERR_STATUS_CLR 寄存器的 Flash0_FAIL_1_H_CLR 位写 1 可清零。	0h
10	Flash0_UNC_ERR_H	R	不可纠正的错误。 1：值为 1 表示 128 位对齐地址的 Flash bank0 高 64 位出现不可纠正的错误。向 ERR_STATUS_CLR 寄存器的 Flash0_UNC_ERR_H_CLR 位写 1 可清零。	0h
15:11	保留			0h
16	Flash1_FAIL_0_L	R	Fail on 0. 0：在 128 位对齐地址的 Flash bank1 低 64 位中未出现 Fail on 0 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的低 64 位中出现单个位错误，并且校正后值为 0。向 ERR_STATUS_CLR 寄存器的 Flash1_FAIL_0_L_CLR 位写 1 可清零。	0h
17	Flash1_FAIL_1_L	R	Fail on 1. 0：在 128 位对齐地址的 Flash bank1 低 64 位中未出现 Fail on 1 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的低 64 位中出现单个位错误，并且校正后值为 1。向 ERR_STATUS_CLR 寄存器的 Flash1_FAIL_1_L_CLR 位写 1 可清零。	0h
18	Flash1_UNC_ERR_L	R	不可纠正的错误。 1：值为 1 表示 128 位对齐地址的 Flash bank1 低 64 位出现不可纠正的错误。向 ERR_STATUS_CLR 寄存器的 Flash1_UNC_ERR_L_CLR 位写 1 可清零。	0h

位/域	名称	R/W	描述	复位值
23:19	保留			0h
24	Flash1_FAIL_0_H	R	Fail on 0. 0：在 128 位对齐地址的 Flash bank1 高 64 位中未出现 Fail on 0 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的高 64 位中出现单个位错误，并且校正后值为 0。向 ERR_STATUS_CLR 寄存器的 Flash1_FAIL_0_H_CLR 位写 1 可清零。	0h
25	Flash1_FAIL_1_H	R	Fail on 1. 0：在 128 位对齐地址的 Flash bank1 高 64 位中未出现 Fail on 1 的单个位错误。 1：值为 1 则表示在 128 位对齐地址的高 64 位中出现单个位错误，并且校正后值为 1。向 ERR_STATUS_CLR 寄存器的 Flash1_FAIL_1_H_CLR 位写 1 可清零。	0h
26	Flash1_UNC_ERR_H	R	不可纠正的错误。 1：值为 1 表示 128 位对齐地址的 Flash bank1 高 64 位出现不可纠正的错误。向 ERR_STATUS_CLR 寄存器的 Flash1_UNC_ERR_H_CLR 位写 1 可清零。	0h
31:27	保留			0h

17.7.13 错误地址寄存器（ERR_POS）

偏移地址：0x618

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
5:0	Flash0_ERR_POS_L	R/W	错误位置（Error position）. Flash bank0 单个位错误的位处于 128 位对齐地址的低 64 位。根据 ERR_位指示校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，则错误位置的范围可以从 0 到 7，否则可以从 0 到 63。	0h
6	保留			0h
7	Flash0_ERR_TYPE_L	R/W	错误类型（Error type）. 0：在 128 位对齐地址的 Flash bank0 低 64 数据位中出现单个位错误。 1：在 128 位对齐地址的 Flash bank0 低 64 位的 ECC 校验位中出现了单个位错误。	0h
13:8	Flash0_ERR_POS_H	R/W	错误位置（Error position）. Flash bank0 单个位错误的位处于 128 位对齐地址的高 64 位。根据 ERR_TYPE 位指示校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，则错误位置的范围可以从 0 到 7，否则可以从 0 到 63。	0h
14	保留			0h

位/域	名称	R/W	描述	复位值
15	Flash0_ERR_TYPE_H	R/W	错误类型（Error type） 0：在 128 位对齐地址的 Flash bank0 高 64 数据位中出现单个位错误。 1：在 128 位对齐地址的 Flash bank0 高 64 位的 ECC 校验位中出现了单个位错误。	0h
21:16	Flash1_ERR_POS_L	R/W	错误位置（Error position） Flash bank1 单个位错误的位处于 128 位对齐地址的低 64 位。根据 ERR_TYPE 位指示校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，则错误位置的范围可以从 0 到 7，否则可以从 0 到 63。	0h
22	保留			0h
23	Flash1_ERR_TYPE_L	R/W	错误类型（Error type） 0：在 128 位对齐地址的 Flash bank1 低 64 数据位中出现单个位错误。 1：在 128 位对齐地址的 Flash bank1 低 64 位的 ECC 校验位中出现了单个位错误。	0h
29:24	Flash1_ERR_POS_H	R/W	错误位置（Error position） Flash bank1 单个位错误的位处于 128 位对齐地址的低 64 位。根据 ERR_TYPE 位指示校验位还是数据位来解释该位置。如果 ERR_TYPE 指示校验位错误，则错误位置的范围可以从 0 到 7，否则可以从 0 到 63。	0h
30	保留			0h
31	Flash1_ERR_TYPE_H	R/W	错误类型（Error type） 0：在 128 位对齐地址的 Flash bank1 高 64 数据位中出现单个位错误。 1：在 128 位对齐地址的 Flash bank1 高 64 位的 ECC 校验位中出现了单个位错误。	0h

17.7.14 错误状态清除寄存器（ERR_STATUS_CLR）

偏移地址：0x61C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	Flash0_FAIL_0_L_CLR	R-0/W1S	清除 Fail on 0 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_FAIL_0_L 位。	0h
1	Flash0_FAIL_1_L_CLR	R-0/W1S	清除 Fail on 1 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_FAIL_1_L 位。	0h
2	Flash0_UNC_ERR_L_CLR	R-0/W1S	清除不可纠正的错误。 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_UCN_ERR_L 位。	0h
7:3	保留			

位/域	名称	R/W	描述	复位值
8	Flash0_FAIL_0_H_CLR	R-0/W1S	清除 Fail on 0 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_FAIL_0_H 位。	0h
9	Flash0_FAIL_1_H_CLR	R-0/W1S	清除 Fail on 1 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_FAIL_1_H 位。	0h
10	Flash0_UNC_ERR_H_CLR	R-0/W1S	清楚不可纠正的错误。 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash0_UCN_ERR_H 位。	0h
15:11	保留			
16	Flash1_FAIL_0_L_CLR	R-0/W1S	清除 Fail on 0 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_FAIL_0_L 位。	0h
17	Flash1_FAIL_1_L_CLR	R-0/W1S	清除 Fail on 1 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_FAIL_1_L 位。	0h
18	Flash1_UNC_ERR_L_CLR	R-0/W1S	清楚不可纠正的错误。 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_UCN_ERR_L 位。	0h
23:19	保留			
24	Flash1_FAIL_0_H_CLR	R-0/W1S	清除 Fail on 0 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_FAIL_0_H 位。	0h
25	Flash1_FAIL_1_H_CLR	R-0/W1S	清除 Fail on 1 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_FAIL_1_H 位。	0h
26	Flash1_UNC_ERR_H_CLR	R-0/W1S	清楚不可纠正的错误。 向该位写 1 将清除 ERR_STATUS 寄存器的 Flash1_UCN_ERR_H 位。	0h
31:27	保留			0h

17.7.15 错误计数器寄存器（ERR_CNT）

偏移地址：0x620

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ERR_CNT	R/W	单位错误计数器（Single bit error count） 每增加一位都会发生 ECC 错误。当达到阈值时，计数器停止计数单位错误。不管是否满足阈值，SINGLE_ERR_INTCLR 位都可以清除 ERR_CNT 位。适用于 ECC 逻辑测试模式和正常工作模式。	0h

17.7.16 错误阈值寄存器（ERR_THRESHOLD）

偏移地址：0x624

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ERR_THRESHOLD	R/W	单位错误阈值（Single bit error threshold） 当 ERR_CNT 值等于 THRESHOLD 值并且发生单比特错误时，将设置 SINGLE_ERR_INT 标志，并触发中断。	0h

17.7.17 错误中断标志寄存器（ERR_INTFLG）

偏移地址：0x628

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SINGLE_ERR_INTFLG	R	单位错误中断标志（Single bit error interrupt flag） 当 ERR_CNT 值等于 ERR_THRESHOLD 值并且发生单位错误时，将设置 SINGLE_ERR_INT 标志并触发 SINGLE_ERR_INT 中断。SINGLE_ERR_INTCLR 位可以清除此位。	0h
1	UNC_ERR_INTFLG	R	不可纠正错误中断标志（Uncorrectable error interrupt flag） 当发生不可纠正错误时，设置此位并触发 UNC_ERR_INT 中断。UNC_ERR_INTCLR 位可以清除此位。	0h
31:2	保留			0h

17.7.18 清除错误中断寄存器（ERR_INTCLR）

偏移地址：0x62C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SINGLE_ERR_INTCLR	R/W	清除单位错误中断标志（Single bit error interrupt flag clear） 此位置 1 将清除 SINGLE_ERR_INTFLG。写 0 没有影响。	0h
1	UNC_ERR_INTCLR	R/W	清除不可纠正错误中断标志（Uncorrectable error interrupt flag clear） 此位置 1 将清除 UNC_ERR_INTFLG。写 0 没有影响。	0h
31:2	保留			0h

17.7.19 数据高测试寄存器（FDATAH_TEST）

偏移地址：0x630

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	FDATAH	R/W	所选数据高（selected Data High） ECC 测试模式下所选数据块的 63:32 位用户可配置。	0h

17.7.20 数据低测试寄存器（FDATAL_TEST）

偏移地址：0x634

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	FDATAL	R/W	所选数据低（selected Data Low） ECC 测试模式下所选数据块的 31:0 位用户可配置。	0h

17.7.21 ECC 测试地址寄存器（FADDR_TEST）

偏移地址：0x638

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	保留			0h
15:3	ADDRL	R/W	所选 64 位数据的地址（Address for selected 64-bit data） ECC 测试模式下所选数据的用户可配置地址位。忽略地址的三个最低有效位，并将位 15:3 写入该域的剩余地址位。	0h
21:16	ADDRH	R/W	所选 64 位数据的地址（Address for selected 64-bit data） ECC 测试模式下所选数据的用户可配置地址位。忽略地址的三个最低有效位，并将位 21:16 写入该域的剩余地址位。	0h
31:22	保留			0h

17.7.22 ECC 测试 ECC 寄存器（FECC_TEST）

偏移地址：0x63C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	ECC	R/W	所选数据的 ECC（selected data ECC） 在 ECC 测试模式下，用户可配置所选 64 位数据块的 ECC 位。	0h
31:8	保留			0h

17.7.23 ECC 控制寄存器（FECC_CTRL）

偏移地址：0x640

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ECC_TEST_EN	R/W	使能 ECC 测试模式（ECC test mode enable） 0：禁止 1：使能	0h
2:1	ECC_SELECT	R/W	ECC 块选择（ECC block select） 2'b00 选择在 Flash bank0 位[63:0]上的 ECC 块的 bank 数据。 2'b01 选择在 Flash bank0 位[127:64]上的 ECC 块的 bank 数据。 2'b10 选择在 Flash bank1 位[63:0]上的 ECC 块的 bank 数据。 2'b11 选择在 Flash bank1 位[127:64]上的 ECC 块的 bank 数据。	0h
3	DO_ECC_CALC	R/W	使能 ECC 计算（Enable ECC calculation） 当通过设置 ECC_TEST_EN 启用 ECC 测试逻辑时，ECC 逻辑将在一个周期内为写入 ECC 测试寄存器的数据和地址进行 ECC 计算。	0h
31:4	保留			0h

17.7.24 测试数据输出高寄存器(FOUTH_TEST)

偏移地址：0x644

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	DATAOUTH	R	测试数据高（test data High.） 保存 63:32 位所选 ECC 块的数据。	0h

17.7.25 测试数据输出高寄存器(FOUTL_TEST)

偏移地址：0x648

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	DATAOUTL	R	测试数据低（test data Low） 保存 31:0 位所选 ECC 块的数据。	0h

17.7.26 ECC 状态寄存器(FECC_STATUS)

偏移地址：0x64C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SINGLE_ERR	R	测试模式 ECC 单位错误标志（Test mode ECC single bit error Flag） 0：无标志 1：测试模式 ECC 单位错误标志	0h
1	UNC_ERR	R	测试模式不可纠正错误高标志（Test mode Uncorrectable Error Flag） 0：无标志 1：测试模式不可纠正错误高标志	0h
7:2	DATA_ERR_POS	R	测试模式单位错误位置（Test mode single bit error position） 保存发生单位错误的位置。取决于 CHK_ERR 位是校验位还是数据位。如果 CHK_ERR 为校验位错误，则错误位置的取值范围为 0 ~ 7 或 0 ~ 63。	0h
8	ERR_TYPE	R	测试模式单位错误类型（Test mode single bit error type） 0：数据位（如果设置了 SINGLE_ERR 位） 1：校验位	0h
31:9	保留			0h

17.7.27 Bank1 单错误地址低寄存器（SINGLE_ERR_ADDR_LOW_1）

偏移地址：0x650

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ERR_ADDR_L_1	R/W	在 128 位对齐 Flash bank1 存储器的低 64 位出现单个位错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.28 Bank1 单错误地址高寄存器 （SINGLE_ERR_ADDR_HIGH_1）

偏移地址：0x654

复位类型：0x0000 0000

位/域	名称	R/W	描述	复位值
31:0	ERR_ADDR_H_1	R/W	在 128 位对齐 Flash bank1 存储器的高 64 位出现单个位错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.29 Bank1 不可纠正的错误地址低寄存器（UNC_ERR_ADDR_LOW_1）

偏移地址：0x658

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	UNC_ADDR_L_1	R/W	在 128 位对齐 Flash bank1 存储器的低 64 位出现不可纠正错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

17.7.30 Bank1 不可纠正的错误地址高寄存器（UNC_ERR_ADDR_HIGH_1）

偏移地址：0x65C

复位类型：SYSRSn

Bits	Name	R/W	Description	复位值
31:0	UNC_ADDR_H_1	R/W	在 128 位对齐 Flash bank1 存储器的高 64 位出现不可纠正错误的 64 位对齐地址。	0h

注意：Bank0，Bank1 mix 区域，128bit 对齐 0x0,0x20,0x40，结尾的地址属于 bank0，128bit 对齐 0x10,0x30,0x50，结尾的地址属于 bank1。

18 定时器 0/1/2（TMR0/1/2）

18.1 术语全称、缩写描述

表格 52 术语全称、缩写描述

中文全称	英文全称	英文缩写
定时器	Timer	TMR

18.2 简介

Cortex-M52 集成 3 个定时器，分为 TMR0/1/2，如果 TMR2 不用于实时操作系统（RTOS）中，那么三个定时器都可以用于用户应用程序中，否则，只有 TMR0 和 TMR1 可以用于应用程序中。

18.3 主要特征

（1） 定时器中断信号

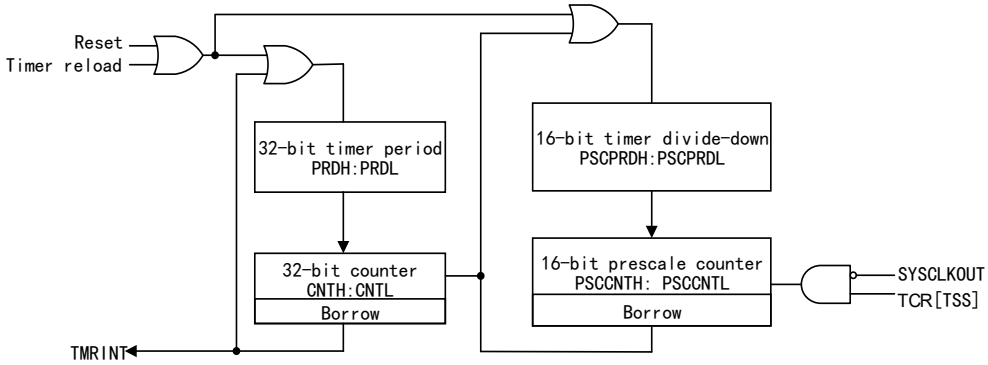
- TMRINT0、TMRINT1、TMRINT2

（2） 可设置 32 位定时器周期并存储当前计数值

（3） 16 位定时器可对时钟信号预分频并存储预分频计数值

18.4 结构框图

图 12 定时器结构框图



18.5 功能描述

18.5.1 定时器操作步骤

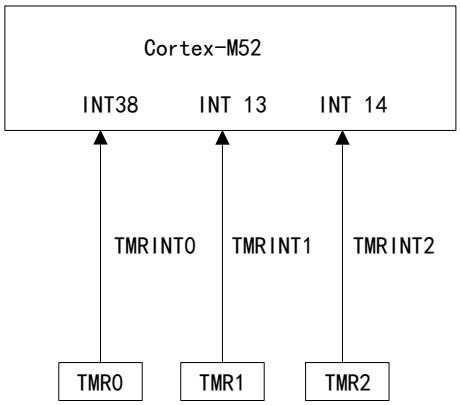
（1） 使用 32 位计数器寄存器 TIM[MSW]：TIM[LSW]，将其加载为周期寄存器 PRD[MSW]：PRD[LSW]中的值

（2） 每经过（TPR [TDDRH：TDDR]+1）个 SYSCLK 周期，计数器递减一次，其中 TDDRH：TDDR 是定时器的分频器

（3） 当计数器达到 0 时，会触发定时器中断输出信号并生成中断脉冲

18.5.2 定时器中断信号和输出信号

图 13 定时器中断信号和输出信号



注意：

（1） 定时器寄存器与 AHB 总线相连。

（2） 定时器与 SYSCLKOUT 同步。

18.6 寄存器组地址

表格 53 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Tmr0Regs	TMR0_REGS	0x5011 0000	0x5011 03FF
Tmr1Regs	TMR1_REGS	0x5011 0400	0x5011 07FF
Tmr2Regs	TMR2_REGS	0x5011 0800	0x5011 0BFF

18.7 寄存器地址映射

注意：TMRx 中 x=0…2

表格 54 TMR 偏移地址

名称	寄存器描述	偏移地址	WRPRT
TIM	计数器寄存器	0x00	-
PRD	周期寄存器	0x04	-
TCR	控制寄存器	0x08	-
TPR	预分频器低位寄存器	0x0C	-

名称	寄存器描述	偏移地址	WRPRT
TPRH	预分频器高位寄存器	0x0E	-

18.8 寄存器功能描述

18.8.1 计数器寄存器（TIM）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	LSW	R/W	计数器值低位（Counter Value Low） 该位域保存计数器当前 32 位计数的低 16 位。当 TIM[MSW]：TIM[LSW]减至零时，会生成 TMRINT 信号，重新加载 PRD[MSW]:PRD[LSW]的周期值。TIM[MSW]：TIM[LSW]的值可以通过设置预分频值 TDDRH：TDDR 决定，每经过（TDDRH：TDDR + 1）个时钟周期，TIM[MSW]：TIM[LSW]的值减 1。	FFFFh
31:16	MSW	R/W	计数器值高位（Counter Value High） 该位域保存计数器当前 32 位计数的高 16 位。当 TIM[MSW]：TIM[LSW]减至零时，会生成 TMRINT 信号，重新加载 PRD[MSW]:PRD[LSW]的周期值。TIM[MSW]：TIM[LSW]的值可以通过设置预分频值 TDDRH：TDDR 决定，每经过（TDDRH：TDDR + 1）个时钟周期，TIM[MSW]：TIM[LSW]的值减 1。	0h

18.8.2 周期寄存器（PRD）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	LSW	R/W	周期低位（Period Value Low） 该位域保存 32 位周期的低 16 位。在下一个定时器输入时钟周期开始时（即预分频器输出），如果 TIM[MSW]：TIM[LSW]减至零，TIM[MSW]：TIM[LSW]会被重新加载为 PRD[MSW]:PRD[LSW]包含的周期值。此外，在 TCR 寄存器中设置 TRB 位，也可以将 PRD[MSW]:PRD[LSW]的内容加载到 TIM[MSW]：TIM[LSW]中。	FFFFh
31:16	MSW	R/W	周期高位（Period Value High） 该位域保存 32 位周期的高 16 位。在下一个定时器输入时钟周期开始时（即预分频器输出），如果 TIM[MSW]：TIM[LSW]减至零，TIM[MSW]：TIM[LSW]会被重新加载为 PRD[MSW]:PRD[LSW]包含的周期值。此外，在 TCR 寄存器中设置 TRB 位，也可以将 PRD[MSW]:PRD[LSW]的内容加载到 TIM[MSW]：TIM[LSW]中。	1h

18.8.3 控制寄存器（TCR）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	保留			1h

位/域	名称	R/W	描述	复位值
4	TSS	R/W	使能定时器停止（Timer Stop Enable） 该标志位指示定时器的启动或停止状态。 0：指示定时器正在运行 1：指示定时器已停止	0h
5	TRB	R/W	重装载定时器（Timer Reload） 0：该位始终读为零，写入 0 时被忽略 1：当将 1 写入该位时，可以将 PRD[MSW]:PRD[LSW]的值加载到 TIM[MSW]：TIM[LSW]中，并将 PSCH：PSC 加载为 TDDRH：TDDR 的值	0h
9:6	保留			0h
10	SOFT	R/W	选择计数器停止时刻（Counter Stop Time Select） 0：在递减少数周期后停止，具体多少个周期并不固定，从汇编程序看，周期与程序断点所在程序有关；从 C 程序看，周期与打在断点上的程序有关。（仅当 FREE=0 时，如果 FREE=1，则此位无效） 1：软停止，在 TIM[MSW]：TIM[LSW]递减到 0 后停止，此时定时器会在关闭之前生成中断（仅当 FREE=0 时，如果 FREE=1，则此位无效）	0h
11	FREE	R/W	选择计数器运行模式（Counter Run Mode Select） 0：硬停止，在 TIM[MSW]：TIM[LSW]下一次递减后停止（SOFT 位控制仿真行为） 1：自由运行（SOFT 位无效）	0h
13:12	保留			0h
14	TIE	R/W	使能定时器中断（Timer Interrupt Enable） 0：禁止定时器中断 1：使能定时器中断，此时如果计时器递减到零，则计时器断言其中断请求	0h
15	TIF	R/W1C	定时器溢出标志（Timer Overflow Flag） 该位表示产生了定时器溢出中断，如果从上次清除 TIF 以来发生了定时器溢出，则设置此位。TIF 不会自动清除，并且不需要清除 TIF 来启用下一个定时器中断 0：定时器未减至零，写入 0 时被忽略 1：定时器减至零时，该标志位将设置此为 1。将 1 写入此位可以清除标志位。	0h

18.8.4 预分频器低位寄存器（TPR）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TDDR	R/W	定时器预分频器重载值低位（Timer Prescaler Period Low） 每经过（TDDRH：TDDR+1）个定时器时钟源循环， TIM[MSW]：TIM[LSW]递减一。在复位时，TDDRH：TDDR 位被清零。通过设置 TDDRH：TDDR 位（将所需的倍数减 1 后写入 TDDRH：TDDR 位），可以调整定时器的计数速度。当 PSCH：PSC 值为 0 时，下一个定时器时钟源循环会触发 TDDRH：TDDR 的内容被重新加载到 PSCH：PSC 中，并且 TIM[MSW]：TIM[LSW]递减一。此外，当软件设置 TRB 位时，TDDRH：TDDR 也会重新加载 PSCH：PSC。	0h
15:8	PSC	R	定时器预分频器计数器低位（Timer Prescaler Counter Low） 该位域保持定时器的当前预分频计数。在每个定时器时钟源周期中，只要 PSCH：PSC 值大于 0，PSCH：PSC 递减一。PSCH：PSC 减至 0 时，在下一个定时器时钟周期内（定时器预分频器的输出），PSCH：PSC 加载 TDDRH：TDDR 的内容，同时定时器计数器寄存器（TIM[MSW]：TIM[LSW]）递减一。当软件设置 TRB 位时，PSCH：PSC 也会重新加载。无法直接设置 PSCH：PSC 的值，但是可以通过读取寄存器检查。该值必须从 TDDRH：TDDR 中获取。复位时，PSCH：PSC 被设置为 0。	0h

18.8.5 预分频器高位寄存器（TPRH）

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TDDRH	R/W	定时器预分频器重载值高位（Timer Prescaler Period High） 请参见 TPR 寄存器的描述	0h
15:8	PSCH	R	定时器预分频器计数器高位（Timer Prescaler Counter High） 请参见 TPR 寄存器的描述	0h

19 看门狗定时器（WDT）

19.1 术语全称、缩写描述

表格 55 术语全称、缩写描述

中文全称	英文全称	英文缩写
看门狗定时器	Watchdog Timer	WDT

19.2 简介

看门狗是专供系统使用的定时器，由 8 位计数器组成，用来监测由软件错误引起的系统故障，保障系统正常运行。

19.3 主要特征

（1） 两个预分频器：
- 预分频器 1：分频比范围从/1 到/64（默认为 1）
- 预分频器 2：分频比范围从/2 到/4096（默认为/512）

（2） 4 种调试模式
- 挂起模式
- 实时单步模式
- 运行自由模式
- 实时运行自由模式

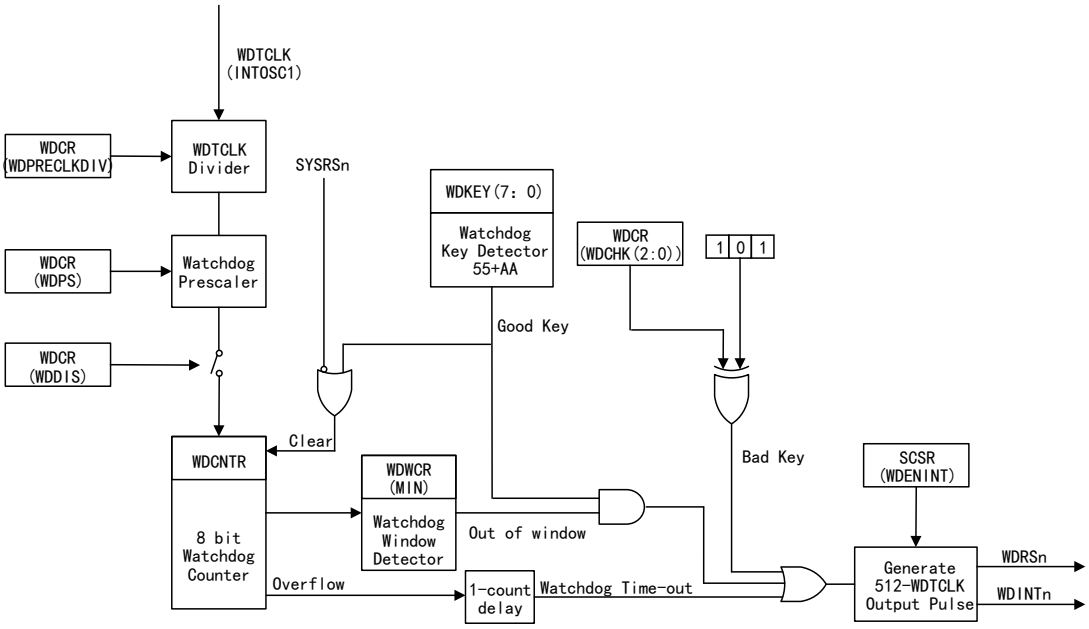
（3） 可配置为中断或复位模式

（4） 低功耗模式分为 IDLE 模式和 HALT 模式

（5） 可设置最小窗口值

19.4 结构框图

图 14 看门狗定时器结构框图



19.5 功能描述

19.5.1 计数器原理

看门狗计数器的时钟源是连接到 INTOSC1 的预分频时钟 WDTCLK，并通过预分频器 1 和预分频器 2 进行分频。预分频器 1 的分频比率范围从/1 到/64，分频值是 2 的幂次；预分频器 2 的分频比率范围从/2 到/4096，分频值是 2 的幂次，其默认值设置为/512（向后兼容）。

当看门狗计数器达到最大值时，它会生成一个脉冲（宽度为 512 个 WDTCLK）触发中断或复位。为了防止看门狗计时器误判系统已停止工作并触发复位，CPU 需要定期向看门狗钥匙寄存器写入 0x55 和 0xAA 序列来重置看门狗计数器。如果不使用看门狗功能，可以将其禁用

19.5.2 调试模式

- 挂起模式：该模式下看门狗时钟被挂起
- 实时单步模式：该模式下看门狗时钟被挂起（即使处于实时中断）
- 运行自由模式：该模式下看门狗会继续正常运行
- 实时运行自由模式：该模式下看门狗模块会正常运行

19.5.3 配置看门狗中断/复位模式

在 SCSR 寄存器中，可以对看门狗进行配置，在看门狗计数器达到最大值时，可以断言看门狗中断（WDINTn）或设定看门狗复位（WDRSn）。

19.5.3.1 中断模式

当看门狗计数器计数结束时，它会执行以下操作：

- 拉低 WDINTn 信号：该操作断言中断请求，这个低电平状态会持续 512 个 OSCCLK 周期；
- 触发 WAKE 中断：WDINTn 的下降沿会触发该中断，但这种触发只会发生在该中断已被使能。

在 WDINTn 处于活动状态时重新使能 WAKEINT 并不会导致中断被多次触发。

读取 SCSR[WDINTS]位可以确定 WDINTn 的当前状态。当 WDINTn 处于活动状态时不应更改看门狗的软件配置，以下是可能导致的后果：

- 如果将中断模式更改为复位模式，将导致设备立即复位；
- 如果从禁用看门狗更改为使能看门狗，将会导致重复中断；
- 如果进行调试复位，则 RESC 寄存器显示复位的原因是由于看门狗复位。

19.5.3.2 复位模式

如果看门狗设定为复位设备，那么当看门狗计数器达到最大值时，WDRSn 信号会在 512 个 OSCCLK 周期内将设备复位（通过拉低设备复位引脚 XRSn）。

19.5.4 低功耗模式下看门狗操作

19.5.4.1 IDLE

当触发看门狗中断（WDINTn）时，该信号将保持低电平（持续 512 个 OSCCLK 周期）。在进入 IDLE 模式之前，如果 WDINTn 信号恢复高电平，它能够向 CPU 发出中断信号使 CPU 退出 IDLE 模式。在 IDLE 模式下，看门狗中断具备与其他外设一样的能力，它可以在 NVIC 寄存器中触发 WAKE 中断，可通过软件确定触发中断的外设。

读取 SCSR[WDINTS]位可以确定 WDINTn 的当前状态，WDINTS 位会在 WDINTn 信号改变后的两个 SYSCLKOUT 周期内更新。

19.5.4.2 HALT

如果将 CLKSRCCTL1[WDHALTI]位设置为 1，内部振荡器和看门狗定时器将保持活动状态。看门狗复位可以从 HALT 模式唤醒系统，但是看门狗中断不能唤醒系统。

19.5.5 最小窗口检查

看门狗有一个可选的“窗口”功能来增强超时机制，它要求在计数器复位之间存在最小延迟。这个功能可以帮助防止绕过正常程序流程的错误情况（看门狗处理除外）。

将所需的最小看门狗计数写入 WDWCR 寄存器设置窗口最小值。此值将在下一个 WDKEY 序列之后生效，此后，当 WDCNTR≥WDWCR 时，可以正常地服务看门狗；当 WDCNTR＜WDWCR 时，任何尝试服务看门狗的操作都将触发看门狗中断或复位。复位时（窗口最小值为零），禁用窗口功能。

19.5.6 维护看门狗定时器

在 WDCNTR 溢出之前，如果连续写入 0x55 和 0xAA，可以复位 WDCNTR：

- 0x55 写入 WDKEY 寄存器时，使能复位
- 写入的下一个值为 0xAA 时，WDCNTR 将复位

只有将 0x55 和 0xAA 写入 WDKEY 才会触发操作。只有向 WDKEY 寄存器写入 0x55，然后写入 0xAA 时，WDCNTR 才会复位。下表为看门狗密钥序列示例。

表格 56 看门狗密钥序列示例

步骤	写入 WDKEY 的值	结果
1	0xAA	无反应
2	0xaa	向 WDKEY 写入除 0x55 和 0xAA 以外的值 无反应
3	0x55	使能复位（如果下一个值为 0xAA 将复位 WDCNTR）
4	0x55	使能复位（如果下一个值为 0xAA 将复位 WDCNTR）
5	0xAA	复位 WDCNTR
6	0x20	向 WDKEY 写入除 0x55 和 0xAA 以外的值 无反应
7	0xAA	无反应（上一个值不是 0x55）
8	0x55	使能复位（如果下一个值为 0xAA 将复位 WDCNTR）
9	0xAA	复位 WDCNTR
10	0x55	使能复位（如果下一个值为 0xAA 将复位 WDCNTR）
11	0xaa	复位 WDCNTR
12	0x55	使能复位（如果下一个值为 0xAA 将复位 WDCNTR）
13	0xAA	复位 WDCNTR

看门狗配置为复位设备时，WDCR 寄存器溢出或向 WDCR[WDCHK]位写入除 1，0，1 以外的值，将会触发设备复位，并设置 RESC[WDRSn]位。复位后，程序可以读取该位状态确定是否发生了由看门狗引起的复位，此后程序清除该位以便检测到后续的看门狗复位）即使设置 WDRSn 标志位，仍然不会阻止看门狗复位。

19.6 寄存器组地址

表格 57 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
WdtRegs	WDT_REGS	0x5002_6400	0x5002_6454

19.7 寄存器地址映射

表格 58 WDT 地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
SCSR	控制状态寄存器	0x44	√
WDCNTR	计数器寄存器	0x46	√
WDKEY	计数器复位寄存器	0x4A	√
WDCR	控制寄存器	0x52	√
WDWCR	窗口寄存器	0x54	√

19.8 寄存器功能描述

19.8.1 控制状态寄存器（SCSR）

偏移地址：0x44

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	WDOVERRIDE	R/W1S	锁定禁用看门狗（Lock Disable Watchdog） 当向该位写入 1 时，该位将被清除并锁定 WDCR 寄存器中的 WDDIS 位。该位将保持此状态，直到下一次系统重置。 读取该位将返回其当前值，写 0 无效。	1h
1	WDENINT	R/W	配置看门狗中断复位（Watchdog Interrupt reset configure） 0：计数器到期，触发复位（上电和任何系统重置后的默认状态） 1：计数器到期，触发中断	0h
2	WDINTS	R-0	看门狗中断状态（Watchdog Interrupt Status） 该位与 SYSCLK 同步。如果要禁用和重启看门狗，该位需为高电平，此时看门狗中断可以进入低功耗模式唤醒系统。 0：看门狗中断处于活动状态 1：看门狗中断处于非活动状态	1h
15:3	保留			0h

19.8.2 计数器寄存器（WDCNTR）

偏移地址：0x46

复位类型：IORSn

位/域	名称	R/W	描述	复位值
7:0	WDCNTR	R	看门狗计数器（Watchdog Counter） 该位包含了看门狗计数器的当前值。该计数器在每个 WDTCLK（INTOSC1）周期中递增，如果向 WDKEY 寄存器写入正确的值，该计数器将复位为 0；如果计数器溢出将生成中断或复位（根据 SCSR 寄存器中 WDENINT 位的值）。	0h
15:8	保留			0h

19.8.3 计数器复位寄存器（WDKEY）

偏移地址：0x4A

复位类型：IORSn

位/域	名称	R/W	描述	复位值
7:0	WDKEY	R/W	复位看门狗计数器（Watchdog Counter Reset） 为了防止看门狗计数器溢出，需要将其复位（写入 0x55 后再写入 0xAA，写其他值无效）。读取该寄存器将返回 WDCR 寄存器的值。	0h
15:8	保留			0h

19.8.4 控制寄存器（WDCR）

偏移地址：0x52

复位类型：IORSn

位/域	名称	R/W	描述	复位值
2:0	WDPS	R/W	看门狗时钟预分频器 1（Watchdog Clock prescaler1） 该位域确定了 WDTCLK。WDTCLK 的频率由以下公式给出： PSCCLK = INTOSC1 / WDPRECLKDIV WDTCLK = PSCCLK / WDPS 看门狗复位或中断脉冲的长度为 512 个 INTOSC1 周期，因此计数器周期须比它长（预分频器 1 和预分频器 2 的乘积必须大于或等于四）。 0：WDTCLK = PSCCLK / 1 1：WDTCLK = PSCCLK / 1（默认） 2：WDTCLK = PSCCLK / 2 3：WDTCLK = PSCCLK / 4 4：WDTCLK = PSCCLK / 8 5：WDTCLK = PSCCLK/ 16 6：WDTCLK = PSCCLK / 32 7：WDTCLK = PSCCLK/ 64	0h
5:3	WDCHK	R-0/W	校验看门狗（Watchdog Check） 对该寄存器进行写操作时，用户必须写入 1，0，1，此时如果启用了看门狗，写入任何其他值将导致核心立即复位。	0h
6	WDDIS	R/W	禁止看门狗（Watchdog Disable） 该位可以被 SCSR 寄存器中的 WDOVERRIDE 位锁定，复位将使能看门狗。	0h

位/域	名称	R/W	描述	复位值
7	WDFLG	R/W1S	看门狗复位标志（Watchdog reset flag） 0：外部设备或上电复位 1：WDRSn 产生复位条件 直到用户写 1 来清除条件之前这个位都保持锁存状态。写 0 将被忽略。	0h
11:8	WDPRECLKDIV	R/W	看门狗时钟预分频器 2（Watchdog Clock prescaler2） 该位域确定了 INTOSC1。WDTCLK 的频率由以下公式确定： PSCCLK = INTOSC1 / WDPRECLKDIV WDTCLK = PSCCLK / WDPS 看门狗复位或中断脉冲的长度为 512 个 INTOSC1 周期，因此计数器周期须比它长（预分频器 1 和预分频器 2 的乘积必须大于或等于四）。预分频器 2 值默认为 512。 0：PSCCLK = INTOSC1 / 512 1：PSCCLK = INTOSC1 / 1024 2：PSCCLK = INTOSC1 / 2048 3：PSCCLK = INTOSC1 / 4096 4-7：保留 8：PSCCLK = INTOSC1 / 2 9：PSCCLK = INTOSC1 / 4 A：PSCCLK = INTOSC1 / 8 B：PSCCLK = INTOSC1 / 16 C：PSCCLK = INTOSC1 / 32 D：PSCCLK = INTOSC1 / 64 E：PSCCLK = INTOSC1 / 128 F：PSCCLK = INTOSC1 / 256	0h
15:12	保留			0h

19.8.5 窗口寄存器（WDWCR）

偏移地址：0x54

复位类型：IORSn

位/域	名称	R/W	描述	复位值
7:0	MIN	R/W	看门狗窗口阈值（Watchdog Window Threshold） 通过 WDKEY 寄存器对计数器进行复位（在计数器的值达到该寄存器中的数值之前），看门狗会立即触发复位或中断。	0h
8	FIRSTKEY	R	检测到第一个有效 WDKEY（First valid WDKEY detected） MIN 配置为非零值后检测到第一个有效的 WDKEY。 0：未检测到 1：检测到 注意： （1）如果 MIN = 0，不会设置此位 （2）如果 MIN 从非零值更改回 0x0，此位将自动清除 （3）此位仅用于调试目的	0h
15:9	保留			0h

20 双时钟比较器（DCCOMP）

20.1 简介

双时钟比较器模块基于另一个更准确和可靠的时钟，用来评估和监控时钟输入。该仪器用于检查时钟源或时钟结构的故障，从而增强系统的安全性。

20.2 主要特征

（1） 支持单序列模式的点测量

（2） 允许选择每个计数器的时钟源

（3） 支持在无需应用程序干预的情况下进行持续监控

（4） 支持可编程的容差窗口（以参考时钟周期的数量来定义）

（5） 允许应用程序确保两个时钟信号的频率保持固定的比例

20.3 结构框图

图 15 DCCOMP 与系统连接

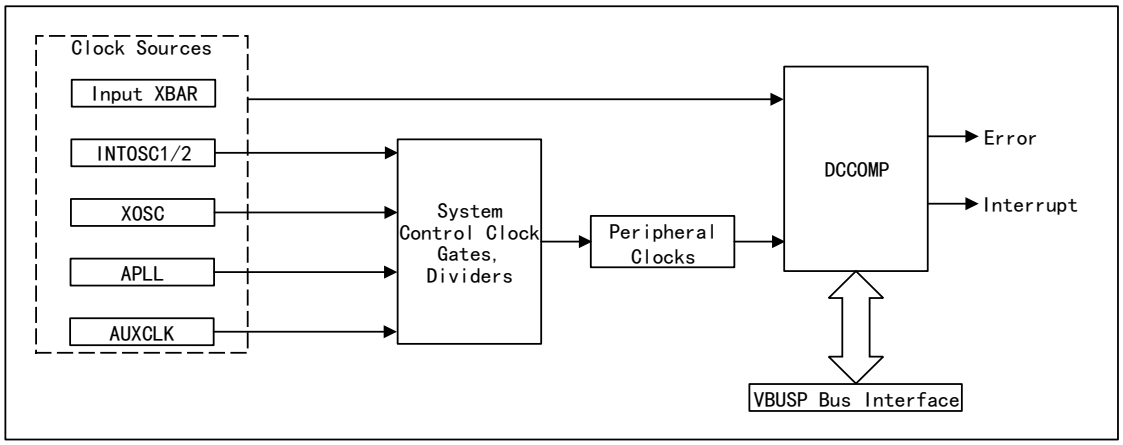
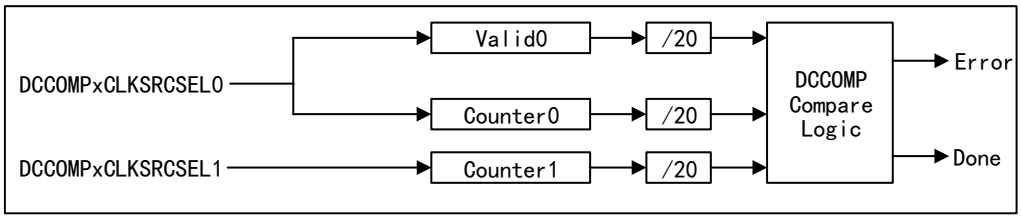


图 16 DCCOMP 结构框图



20.4 功能描述

DCCOMP 模块包含 Counter0、Counter1 和 Valid0 三个计数器。需要先定义所有计数器装载的预装载值。当使能 DCCOMP 后，Counter0 和 Counter1 开始递减，其速率分别由 Clock0 和 Clock1 的频率决定。如果 Counter0 等于 0，即超时的时候，则 Valid0 计数器以 Clock0 的速率递减。当在有效窗口内，Counter1 递减至 0 时，不会产生错误且认为 Clock1 在用户配置的容差内工作良好。

20.4.1 DCCOMP 计数器

Counter0 和 Counter1 根据 Clock0 和 Clock1 之间的比例来配置，即 F_{CLK0} * Counter1 = F_{CLK1} * Counter0。Valid0 计数器用于提供容差并基于 DCCOMP 的误差来配置。

配置计数器需考虑的误差源

由于 Clock0 和 Clock1 异步，因此，所有计数器的启动和停止不会同步发生。所以在配置计数器时，必须考虑以下不同的误差源：

（1） 数字化误差等于 8 个 Clock0 周期

（2） 由于 Clock0 和 Clock1 异步导致 DCCOMP 误差（取决于 Clock0 和 Clock1 的频率）

- 当 F_{CLK1} 未知时，异步误差 = 2*[max（F_{SYSCLK}/F_{CLK0}）]+2，其单位为 Clock0 周期
- 当 F_{CLK1} 大于 F_{CLK0} 时，异步误差 = 2*（F_{SYSCLK}/F_{CLK0}）+2，其单位为 Clock0 周期
- 当 F_{CLK1} 小于 F_{CLK0} 时，异步误差 = 2*（F_{CLK0}/F_{CLK1}）+2*[max（F_{SYSCLK}/F_{CLK0}）]，其单位为 Clock0 周期

计算 DCCOMP 总误差

（1） DCCOMP 误差计算公式

- DCCOMP 误差=异步误差+数字化误差，其单位为 Clock0 周期

时钟测量下的频率误差称为 DCCOMP 误差，是由 DCCOMP 导致的，不代表时钟测量的频率误差。在配置计数器时需要注意这个误差，并选取合适的容差，即 DCCOMP 误差决定了测量的窗口。

（2） 窗口计算公式

- 窗口 = DCCOMP 误差/（容差*0.01），其单位为 Clock0 周期。

假如 DCCOMP 误差为 10 且期望容差为±0.1%，则窗口等于 10000 Clock0 周期。

基于窗口计算公式可知，期望的容差越低计数器值越大，那么测量窗口也越大，也就是说容差为 0.1%时的计数器值大于容差为 0.2%时的计数器值。所以，测量窗口由具体应

用所需的容差来决定，由 Clock0 周期来定义。

（3） 允许频率误差计算公式

测量时钟可能存在频率误差范围。当预测存在误差时，在配置计数器时也要考虑到该误差。当测量 INTOSC1/2 的频率时用外部晶振被作为参考时钟，在计数器配置中应考虑到 INTOSC1/2 所允许的频率容差。公式如下所示：

- 允许频率误差 =（频率容差/100）*窗口，其中允许频率误差单位为 Clock0 周期，频率容差单位为%。

（4） 总误差计算公式

- 总误差 = DCCOMP 误差 + 允许频率误差

配置计数器值

- CNTSEED0 = 窗口 – 总误差
- CNTSEED1 = （F_{CLK1} / F_{CLK0}）* 窗口
- VALSEED0 = 总误差 * 2

20 位计数器 Counter1 的最大计数值不能超过 1048575，当 Counter1 超过该值时，DCCOMP 误差的目标容差就会增大，从而减小测量窗口。因此，最小容差的计算公式如下所示：

- 容差=[（F_{CLK1} / F_{CLK0}）* DCCOMP 误差 * 100]/ 1048575

20.4.2 单发测试模式

DCCOMP 模块可以通过使能单发模式来进行一次倒数。在该模式下，DCCOMP 在 Counter0 和 Valid0 到达 0 之后停止工作

为了阻止进一步的计数，在该模式下的倒数序列结束时，将会自动禁用 DCCOMP。该模式通常用于对信号频率进行抽样检查。

使用案例：验证 PLLRAWCLK 的频率

该例子是将 XTAL 用作参考时钟，验证 PLL 输出时钟频率的实际用法。

假设条件：

- XTAL 为 10MHz
- PLL 输出频率为 100MHz
- SYSCLK 为 100MHz
- 允许的频率容差为 0.1%
- DCCOMP 容差为 0.1%

测量顺序如下：

（1） 将 Counter0 和 Valid0 的 Clock0 源设置为 XTAL，将 Counter1 的 Clock1 源设置为 PLL 输出时钟。

（2） 根据计算 DCCOMP 计数器章节中的公式，计算出计数器的种子值为 CNTSEED0 = 29940，CNTSEED1 = 300000，VALSEED0 = 120

（3） 一旦使能 DCCOMP 之后，Counter0/1 都从种子值开始向下计数。

（4） Counter0 到达 0 之后将会自动触发 Valid0 计数器。

（5） 当 Valid0 到达 0 而 Counter1 为非 0 时，ERRFLG 置位且 DCCOMP 错误被送至 NVIC。此时，Counter1 停止计数（冻结）。用户可以使能一个中断，使得任何发生该 DCCOMP 错误的时候，NVIC 产生中断。

（6） 为了使 DCCOMP 模块准备下一次抽样测量，应用程序需要清除 ERRFLG 位并重启该模块

当在序列结束没有发生错误时，SIGMDONEFLG 置位并产生中断。应用程序必须在重启 DCCOMP 前清除 SIGMDONEFLG 标志位。

错误状态

发生以下任一情况时，都会产生错误状态：

- Counter1 在 Counter0 到达 0 之前向下计数至 0。表示 Clock1 比预期要快，或 Clock0 比预期要慢。
- Counter1 在 Counter0 和 Valid0 都到达 0 后还未到达 0。表示 Clock1 比预期慢。

两种错误都会暂停计数。应用程序可以读取此时计数器的值，以辅助确定错误原因。

20.4.3 中断

DCCOMP 在以下任意一个事件中产生中断：

（1） 当 DONEEN=1 时，在 DCCOMP 完成计数且所有计数器在给定的窗口内超（表示完成信号操作）时产生一个中断。

（2） 当 ERREN=1 时，在计数器在给定的窗口内未超时，DCCOMP 计数出现错误（表示错误事件）时产生一个中断。

完成信号事件和错误事件引发的中断被相或运算并被标志为 DCCOMP 中断。为了确定该中断是否由完成信号事件或错误事件引发，应用程序的中断服务程序需要检查 DCCOMP_FLG 寄存器内的状态标志。

20.5 寄存器组地址

表格 59 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Dcomp0Regs	DCCOMP_REGS	0x5010 1400	0x5010 17FF

20.6 寄存器地址映射

表格 60 DCCOMP_REGS 地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
DCCOMP_CTRL	控制寄存器	0x00	-
DCCOMP_REV	版本寄存器	0x08	-
DCCOMP_CNTSEED0	计数器 0 种子值寄存器	0x10	-
DCCOMP_VALSEED0	有效种子值 0 寄存器	0x18	-
DCCOMP_CNTSEED1	计数器 1 种子值寄存器	0x20	-
DCCOMP_FLG	标志寄存器	0x28	-
DCCOMP_CLKSRCCNT0	计数器 0 值寄存器	0x30	-
DCCOMP_CLKSRCVAL0	有效值 0 寄存器	0x38	-
DCCOMP_CLKSRCCNT1	计数器 1 值寄存器	0x40	-
DCCOMP_CLKSRCSEL1	时钟源 1 寄存器	0x48	-
DCCOMP_CLKSRCSEL0	时钟源 0 寄存器	0x50	-

20.7 寄存器功能描述

20.7.1 控制寄存器（DCCOMP_CTRL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	DCCOMPEN	R/W	使能 DCCOMP（DCCOMP Enable） 0101：计数器停止 其他：正在运行计数器	5h
7:4	ERREN	R/W	使能错误（Error Enable） 0101：禁止错误信号 其他：使能错误信号	5h
11:8	SIGEN	R/W	使能一次性模式（Single Shot Mode Enable） 该位用于使能或禁止 DCCOMP 的重复操作。 1010：当 CLKSRCCNT0 和 CLKSRCVAL0 都达到 0 时，停止计数 其他：保留	5h
15:12	DONEEN	R/W	使能完成信号（DONE Enable） 该位用于使能或禁止完成中断信号，DCCOMP_FLG 寄存器中的完成状态不受该位影响 0101：禁止完成信号 其他：使能完成信号	5h

位/域	名称	R/W	描述	复位值
31:16	保留			0h

20.7.2 版本寄存器（DCCOMP_REV）

偏移地址：0x08

复位类型：SYSRSn

该寄存器用于指定模块的版本。

位/域	名称	R/W	描述	复位值
5:0	MINOR	R	次要修订编号（Minor Revision Number） 这些位域用于决定模块的次要更改，例如增强现有功能。	3h
7:6	保留			0h
10:8	MAJOR	R	主要修订编号（Major Revision Number） 这些位域用于决定模块的主要更改，例如更改或添加新功能。	0h
15:11	保留			2h
27:16	保留			4h
29:28	保留			0h
31:30	保留			1h

20.7.3 计数器 0 种子值寄存器（DCCOMP_CNTSEED0）

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
19:0	CNTSEED0	R/W	计数器 0 的种子值（Counter 0 Seed Value） 该位域用于表示时钟源 0 的计数器种子附加值。如果将该位域的值设置为 0，则会导致 DCCOMP 未定义的操作。	0h
31:20	保留			0h

20.7.4 有效种子值 0 寄存器（DCCOMP_VALSEED0）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	VALSEED0	R/W	有效时间计数器 0 的种子值（Valid Duration Counter 0 Seed Value） 该位域用于表示时钟源 0 的超时计数器种子附加值。如果将该位域的值设置为 0，则会导致 DCCOMP 未定义的操作。DCCOMP_CLKSRCVAL0 寄存器定义了计数器 1 到期的时间窗口，该窗口的宽度至少应为 4 个周期。禁止将小于 4 的值编程到 DCCOMP_VALSEED0 寄存器中。	0h

位/域	名称	R/W	描述	复位值
31:16	保留			0h

20.7.5 计数器 1 种子值寄存器（DCCOMP_CNTSEED1）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
19:0	CNTSEED1	R/W	计数器 1 的种子值（Counter 1 Seed Value） 该位域用于表示时钟源 0 的计数器种子附加值。如果将该位域的值设置为 0，则会导致 DCCOMP 未定义的操作。	0h
31:20	保留			0h

20.7.6 标志寄存器（DCCOMP_FLG）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ERRFLG	R/W	错误标志（Error Flag） 通过写 1 清除该位。 0：未产生错误 1：产生错误	0h
1	SIGMDONEFLG	R/W	单次模式完成标志（Single-Shot Mode Done Flag） 该位表示单次模式何时完成且未产生错误。通过写 1 清除该位。 0：单次模式未完成 1：单次模式完成	0h
31:2	保留			0h

20.7.7 计数器 0 值寄存器（DCCOMP_CLKSRCCNT0）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
19:0	CLKSRCCNT0	R	时钟源 0 的计数器附加值（Clock Source 0 counter attached Value） 该位域用于表示时钟源 0 的计数器附加值。	0h
31:20	保留			0h

20.7.8 有效值 0 寄存器（DCCOMP_CLKSRCVAL0）

偏移地址：0x38

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CLKSRCVAL0	R	时钟源 0 上的有效计数器附加值（Clock Source 0 valid counter attached Value） 该位域用于表示时钟源 0 的有效计数器附加值。	0h
31:16	保留			0h

20.7.9 计数器 1 值寄存器（DCCOMP_CLKSRCCNT1）

偏移地址：0x40

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
19:0	CLKSRCCNT1	R	时钟源 1 的计数器附加值（Clock Source 1 counter attached Value） 该位域用于表示时钟源 1 的计数器附加值。	0h
31:20	保留			0h

20.7.10 时钟源 1 寄存器（DCCOMP_CLKSRCSEL1）

偏移地址：0x48

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	CLKSRCSEL1	R/W	选择计数器 1 的时钟源（Counter 1 Clock Source Select） 当 CLKSRCWEN 位域使能计数器 1 的时钟源写功能时，指定计数器 1 的时钟源。 0000：SYSPLLCLKOUT 的直接输出 0001：保留 0010：INTOSC1 输出时钟 0011：INTOSC2 输出时钟 0100：保留 0101：保留 0110：系统时钟 0111：保留 1000：保留 1001：保留 1010：保留 1011：系统时钟 1100：SPI 和 UART 模块的位时钟 1101：ADC 转换时钟 1110：保留 1111：CAN0 位时钟	2h
11:4	保留			0h

位/域	名称	R/W	描述	复位值
15:12	CLKSRCWEN	R/W	使能计数器 1 的时钟源写功能（Counter 1 Clock Source Write Enable） 1010：当特定密钥写入时，CLKSRCSEL1 位域将更新为时钟计数器 1 的新选择 其他：保留先前数值，对寄存器位域的新写入无影响	5h
31:16	保留			0h

20.7.11 时钟源 0 寄存器（DCCOMP_CLKSRCSEL0）

偏移地址：0x50

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	CLKSRCSEL0	R/W	选择计数器 0 的时钟源（Counter 0 Clock Source Select） 当 CLKSRCWEN 位域使能计数器 1 的时钟源写功能时，指定计数器 0 的时钟源。 0000：晶体振荡器输出 0001：INTOSC1 输出时钟 0010：INTOSC2 输出时钟 其他：保留	1h
31:4	保留			0h

21 通用输入/输出引脚（GPIO）

21.1 术语全称、缩写描述

表格 61 术语全称、缩写描述

中文全称	英文全称	英文缩写
通用输入/输出引脚	General-purpose Input/Output	GPIO

21.2 简介

GPIO 模块中，可以单独将每个引脚配置为数字 I/O（即 GPIO 模式），或者与外设信号连接。GPIO 端口控制引脚的数字和模拟 I/O 多路复用功能，通过引脚共享的方式，最大程度地提高了引脚应用的灵活性。此外，输入信号可以去除不必要的噪声。

21.3 主要特征

（1） 输入引脚限定类型可配置
- 同步输入限定
- 异步输入限定
- 3 采样输入限定
- 6 采样输入限定

（2） 单引脚设置/清除/切换

（3） 输入路径与输出路径独立

（4） 支持外设多路复用功能

（5） CPU0 和 CPU1 可独立读取引脚状态

（6） 支持低功耗模式唤醒

21.4 结构框图

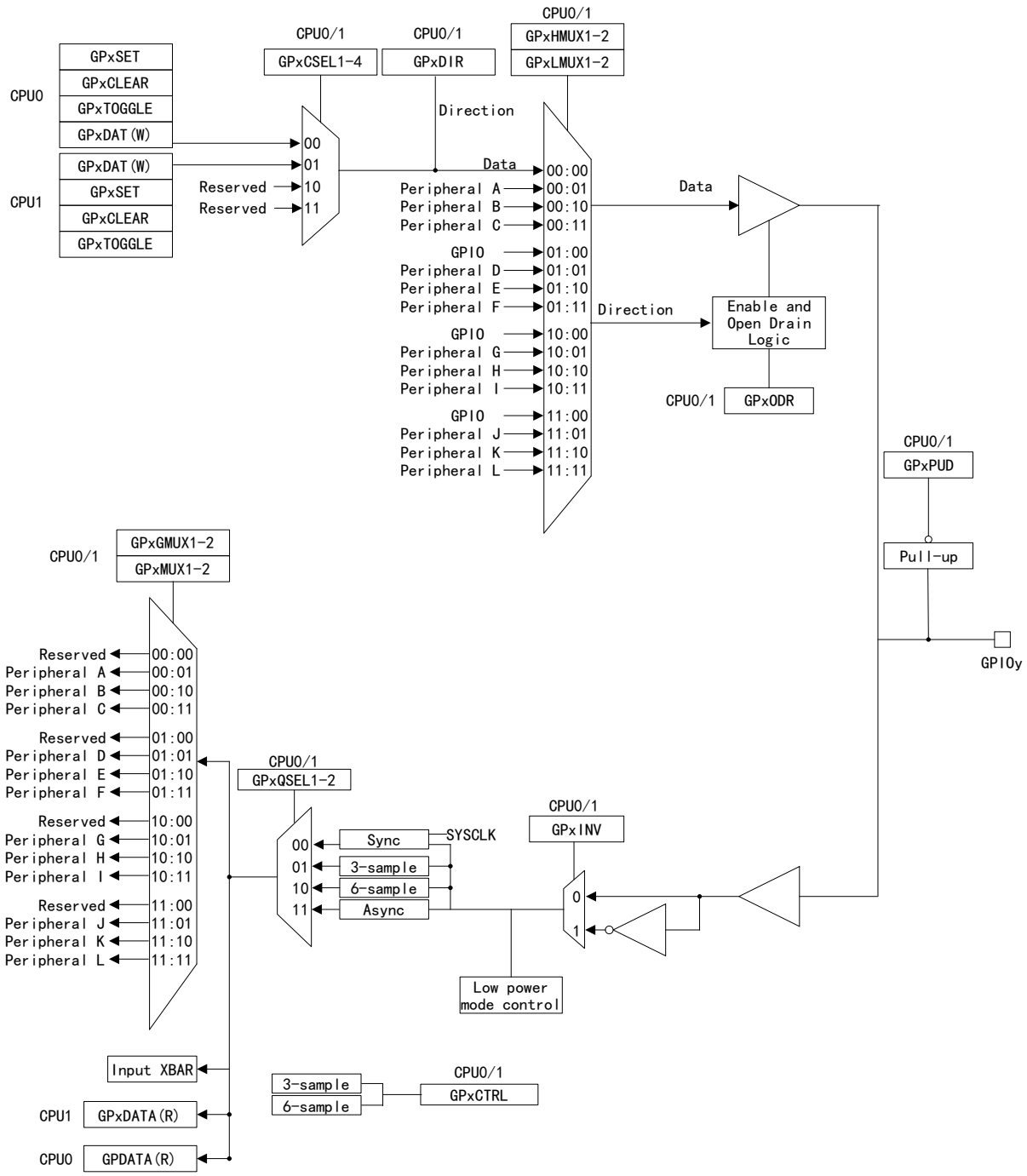
如结构框图所示，GPIO 端口有两个特性：

（1） 在远离引脚处进行外设复用；

（2） 输入路径和输出路径各自独立，而仅在引脚处进行连接。

基于以上两个特性，CPU0 和 CPU1 读取引脚物理状态的过程是始终独立于 CPU 主控和外设复用的。例如，对于所有的主机和外设，输入限定和开漏输出的所有引脚选项都有效。

图 17 GPIO 结构框图



注意：

（1） 复位时，GPIO22 和 GPIO23 处于特殊的模拟模式，当需要重新配置为 GPIO 使用时，必须清除 GPAAMSEL 中的对应位。此外，GPIO23 的最大触发频率不得超过 12 MHz。

（2） 数字信号连接到模拟 IO（AIO）时，若此时的数字信号的边沿变化速率（即 dv/dt）很高，可能会干扰到相邻的模拟信号。当相邻的通道用于模拟功能时，需要限制该数字信号连接到模拟 IO（AIO）的边沿变化速率，以减少数字信号与相邻的模拟信号之间的干扰。

（3） GPIO18/X2 的时序特性因振荡器电路施加的负载不同而与一般 GPIO 有差异。有关 GPIO18/X2 作为 GPIO 引脚使用的特性，详情请参见用户手册的时钟章节和数据手册。

（4） JTAG 信号的传输路径与上图不同，无法进行反相或输入限定操作。

21.5 功能描述

21.5.1 端口介绍

端口组成

（1） GPIOA：GPIO0-GPIO31

（2） GPIOB：GPIO32-GPIO59

（3） GPIOH：GPIO224-GPIO254

端口功能

在 GPIOA 和 GPIOB 端口中，CPU0 和 CPU1 之一或外设可以控制每个引脚的输出。引脚的输入/输出功能除了可以被 CPU 控制外，单个使能的引脚还可以被外设信号通过复用功能使用。外设信号多达 12 个，并互相独立。

在 GPIOH 端口中，引脚的数字输入功能和模拟信号功能复用，不具有数字输出功能。

21.5.2 引脚配置

21.5.2.1 配置概述

通过以下步骤对 I/O 引脚进行配置：

（1） 选择引脚

根据目标应用程序，选择用于外设和每个 CPU 输入/输出的引脚，引脚的复用功能详情请参见数据手册。所有引脚在默认情况下都是通用 I/O，而不作为外设信号。通过写入 GPxMUX1/2 和 GPxGMUX1/2 寄存器中的对应位，以配置所选择的引脚功能。

注意：为了避免多路复用时出现毛刺，需先将 GPxMUX1/2 寄存器中的对应位置 0，再更改 GPxGMUX1/2 寄存器中对应位的值。

（2） 使能内部上拉电阻（可选）

默认情况下，所有引脚的上拉电阻都为禁用状态。通过向端口 x 上拉禁用寄存器（GPxPUD）中引脚的对应位写入 0 或 1，以启用或禁用该引脚的上拉电阻。利用上拉电阻可以确保在没有外部信号输入引脚时，引脚的状态是已知的（高电平或低电平）。

（3） 选择输入引脚限定

默认情况下，所有限定同步，采样周期为 PLLSYSCLK。当 GPIO 作为输入引脚时，需要配置引脚所需的输入限定（如果有限定类型）。输入限定由限定采样周期和限定类型组成，在 GPxCTRL 寄存器中配置引脚的采样周期，在 GPxQSEL1/2 寄存器中配置引脚

的输入限定类型。更多输入限定的说明，请参考输入限定章节。

（4） 配置引脚方向

所有 GPIO 引脚在默认情况下为输入引脚。通过向 GPxDIR 寄存器写入 0 或 1，为配置成 GPIO 的每个引脚指定为输入或输出方向。写入 GPxDIR 寄存器的步骤如下：

- 将值写入 GPxSET/GPxCLR/GPxDAT 寄存器中
- 将以上寄存器中的值装载到输出锁存器中（所有输出锁存器默认情况下为 0）；
- 锁存器中的值写入 GPxDIR 寄存器；

（5） 选择低功耗模式唤醒源

低功耗模式唤醒系统可以通过 GPIO0-59 进行。向 CPU 系统寄存器空间的 GPIOLPMSEL0/1 寄存器的对应位写入 0 或 1，可选择用于唤醒的一个或多个 GPIO。

（6） 配置外部中断

配置外部中断分为以下两步：

- 使能中断，并通过 TIM 寄存器配置中断的极性。
- 分别选择 X-BAR 信号 4，5，6，13 和 14 的源以设置 EXTI1-5 的 GPIO 引脚。

更多 Input X-BAR 架构的相关信息，详情请参见 X-BAR 章节。

21.5.2.2 复位后 IO 配置

复位后，默认 GPIO 为输入引脚且内部上拉电阻处于禁用状态。因此，必须将每个 GPIO 引脚配置成输出模式，由另一个组件在电路板上将电平驱动到 V_{ih} 以上或 V_{il} 以下的输入模式或使能内部上拉电阻的输入模式三种模式之一，以避免未被驱动的输入引脚浮动到中间电平并造成输入缓冲区上的穿透电流浪费。

为了防止浮空输入，在引脚数较少的产品中，未绑定 GPIO 的上拉电阻在默认情况下是使能的。在实际使用时，注意不要在应用程序代码中禁用这部分上拉电阻。

21.5.2.3 输入配置

输入配置指配置输入模式中引脚的输入限定模式。

引脚的输入限定模式可以灵活配置：在寄存器 GPxQSEL1/2[GPIOn]中配置每个引脚的输入类型。对于不同功能的输入引脚，可选择的输入类型不同：

（1） GPIO 输入引脚：

- 仅与 SYSCLKOUT 同步（[GPIOn]=00）
- 通过采样窗口进行限定（[GPIOn]=01/10）

（2） 外设输入引脚：

- 仅与 SYSCLKOUT 同步（[GPIOn]=00）
- 通过采样窗口进行限定（[GPIOn]=01/10）
- 异步输入（[GPIOn]=11）

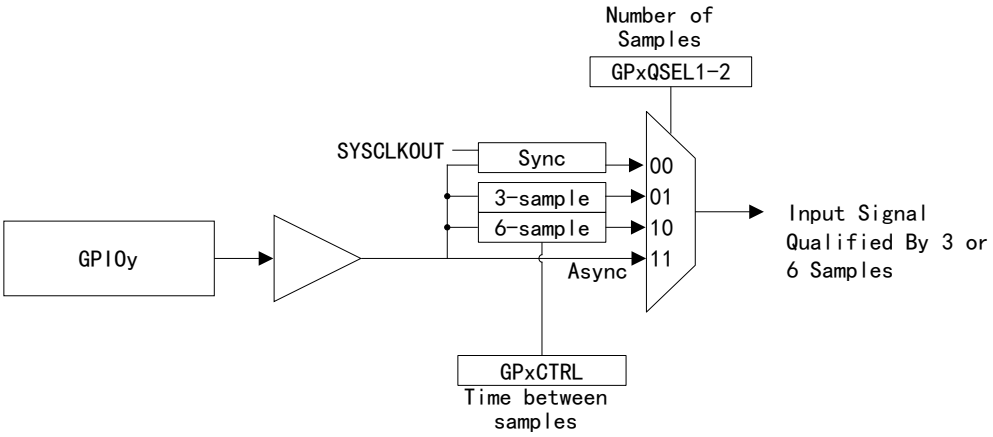
同步输入限定（仅与 SYSCLKOUT 同步）

即 GPxQSEL1/2[GPIOn]=00，所有引脚复位后默认为此限定模式。输入信号仅与系统时钟（SYSCLKOUT）同步，不对信号执行进一步的限定。由于输入信号是异步信号，因此改变 MCU 的输入可能需要长达一个 SYSCLKOUT 的延迟时间。

使用采样窗口的输入限定

即 GPxQSEL1/2[GPIOn]=01/10，如下图所示，信号首先与系统时钟（SYSCLKOUT）同步，接着在输入更改许可前对输入执行指定周期数的限定。在此模式中，需要为限定配置采样周期（即信号采样频率）和采样数两个参数。

图 18 使用采样窗口的输入限定



如图采样窗口的输入限定和图输入限定时钟周期所示，通过执行输入限定可以消除多余噪声。

（1） 采样周期（采样间隔时间）

输入信号以固定周期进行采样以保证输入信号符合要求。用户可以指定相对于 CPU 时钟（SYSCLKOUT）的采样周期，并确定采样点之间的持续时间（即采样频率）。

通过 GPxCTRL[QUALPRDn]配置采样周期，如位域所示，在八个输入信号的组中配置采样周期。更多采样周期配置信息，详情请参见 GPxCTRL 寄存器。

GPxCTRL[QUALPRDn]的配置与采样周期或采样频率的关系如下表所示：

表格 62 QUALPRDn 与采样周期或采样频率的关系

GPxCTRL[QUALPRDn]	采样周期	采样频率
0	$T_{SYSCLKOUT}$ (1) *1	$f_{SYSCLKOUT}$ (2)
非 0	$T_{SYSCLKOUT}$*2*GPxCTRL[QUALPRDn]	$f_{SYSCLKOUT}$*1/(2*GPxCTRL[QUALPRDn])

注意

（1） $T_{SYSCLKOUT}$ 为 SYSCLKOUT 的时间周期；

（2） $f_{SYSCLKOUT}$ 为 SYSCLKOUT 的频率。

根据以上关系，当 SYSCLKOUT 的频率确定时，可以计算采样频率的最小值和最大值，示例如下：

表格 63 $f_{SYSCLKOUT}$=60MHz 时，最小/最大采样频率

GPxCTRL[QUALPRDn]	采样频率计算公式	采样频率/周期值
00	$f_{SYSCLKOUT}$=60MHz	60MHz/16.67ns
FF	$f_{SYSCLKOUT}$*1/(2*GPxCTRL[QUALPRDn]) =60MHz*1/(2*255)	117.647kHz/8.5μs

（2） 采样数

根据 GPxQSEL1/2[GPIOn]（x=A/B，n 表示引脚编号）位域所示，可以将采样数配置为 3 个或 6 个。

即如果连续 3 个或 6 个周期的输入信号都相同时，输入信号的变化将传递给 MCU，否则变化将被忽略。

（3） 总采样窗口宽度

采样窗口宽度指对输入信号进行采样的时间，由采样周期和采样数共同决定。为了使输入限定器可以检测到输入信号的变化，在采样窗口期间或更长时间内，输入信号的电平必须保持稳定。采样窗口总宽度计算如下所示：

表格 64 总采样窗口宽度计算

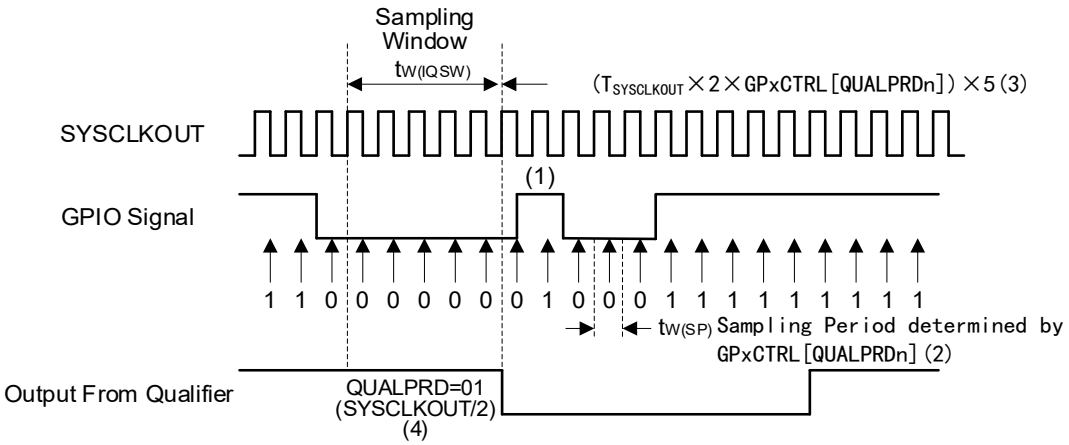
GPxCTRL[QUALPRDn]	3 采样窗口宽度	6 采样窗口宽度
0	$T_{SYSCLKOUT}$*2	$T_{SYSCLKOUT}$*5
非 0	$T_{SYSCLKOUT}$*2*GPxCTRL[QUALPRDn]*2	$T_{SYSCLKOUT}$*2*GPxCTRL[QUALPRDn]*5

如上表所示，采样周期数=采样数-1。即对于 3 采样窗口，采样窗口宽度为 2 个采样周期 (t_w (IQSW)=2*t_w(SP))，而对于 6 采样窗口，采样窗口宽度为 5 个采样周期 (t_w(IQSW)=5*t_w(SP))。

注意：外部信号变化与 SYSCLKOUT 和采样周期之间存在异步关系，因此，为了外部信号变化可以正确被逻辑电路识别到，在超出采样窗口宽度的一定时间内，输入信号应保持稳定。超出采样窗口的时间（即额外时间）最多可以为 $T_{SYSCLKOUT}$+t_w(SP)。更多输入信号保持稳定的时间的相关信息，详情请参见数据手册。

输入限定时间周期计算示例：

图 19 输入限定时钟周期



注意：

（1） 由于此故障比限定窗口短，输入限定器将忽略此错误。采样周期由 QUALPRDn 位域指定，取值范围 0x00-0xFF。当 QUALPRDn=00 时，则采样周期为 1 个 $T_{SYSCLKOUT}$；当 QUALPRDn≠00 时，采样周期为 $T_{SYSCLKOUT}$*2n(即对 GPIO 引脚进行采样的时间间隔为 $T_{SYSCLKOUT}$*2n)。

（2） 选择哪 8 个 GPIO 构成的引脚组用于此限定周期。

（3） 选择采样数，可选 3 采样或 6 采样。

（4） 示例中，为了使输入信号变化被限定器检测到，在 10（即 QUALPRDn*5*2）个以上 $T_{SYSCLKOUT}$ 周期内，输入信号应保持稳定，以保证 5 采样周期的检测顺利进行。从外部信号被异步驱动开始，输入脉冲长达 13 个 $T_{SYSCLKOUT}$ 能保证信号的变化可以被识别到。

如上图所示，其中：

t_w(SP)：采样周期

t_w (IQSW)：输入限定采样窗口

采样配置如下：

GPxQSEL1/2[GPIOn]=10，GPxCTRL[QUALPRDn] =01

即此示例中，各参数计算公式如下。

- 采样数：6，
- 采样周期：t_w(SP)=$T_{SYSCLKOUT}$*2*GPxCTRL[QUALPRDn]=$T_{SYSCLKOUT}$*2。
- 采样窗口宽度：t_w (IQSW) = t_w (SP)*5= $T_{SYSCLKOUT}$*2*GPxCTRL[QUALPRDn]*5= $T_{SYSCLKOUT}$*2*5
- 假设 $T_{SYSCLKOUT}$=16.67ns，则各参数值如下：
- 采样周期：t_w(SP)=$T_{SYSCLKOUT}$*2=16.67ns*2=33.3ns。
- 采样窗口宽度：t_w(IQSW)=t_w(SP)*5=33.3ns*5=166.5ns。

由于输入信号的异步特性，所需的额外时间：

- $T_{SYSCLKOUT}$+一个额外的采样周期=$T_{SYSCLKOUT}$+t_w(SP)=16.67ns+33.3ns=50.0ns

故总采样时间至少为：

- t_w(IQSW)+$T_{SYSCLKOUT}$+t_w(SP)=166.5ns+50.0ns=216.5ns

异步输入限定

即 GPxQSEL1/2[GPIOn]=11，此模式用于不需要执行同步输入的外设或本身执行同步的外设，这些外设包含通信端口 McBSP，I2C，SPI 和 UART。此外，PWM Trip Zone(TZn)信号的功能独立于 SYSCLKOUT 是可取的。

当引脚配置为 GPIO 功能时，此异步模式无效。如果引脚配置为 GPIO 功能，并配置为异步输入限定，则此引脚的限定默认同步到 SYSCLKOUT（详情请参见同步输入限定（仅与 SYSCLKOUT 同步））。

注意：当外设本身执行同步时，将引脚配置为同步输入限定可能会导致不可以预测的结果。因此，当外设本身执行同步时，应确保 GPIO 引脚配置为异步输入限定。

21.5.2.4 IO 电平

每个 I/O 端口都有对应的寄存器，且每个 GPIO 引脚都有对应的位/域，每个 GPIO 引脚的值可以通过配置对应的位域进行更改。

GPxDAT 端口 x 数据寄存器

引脚经过限定后的当前状态由此寄存器中的对应位表示，并且不考虑引脚是配置成 GPIO 还是外设功能。详情请参见寄存器功能描述中的端口 x 数据寄存器。

GPxSET 端口 x 设置寄存器

设置寄存器用于将指定的 GPIO 引脚驱动为高电平，并且不会干扰其他引脚。更多关于设置寄存器的相关信息，详情请参见寄存器功能描述中的端口 x 设置寄存器。

GPxCLR 端口 x 清除寄存器

清除寄存器用于将指定的 GPIO 引脚驱动为低电平，并且不会干扰其他引脚。更多关于清除寄存器的相关信息，详情请参见寄存器功能描述中的端口 x 清除寄存器。

GPIO_xTOG 端口 x 切换寄存器

切换寄存器用于将指定的 GPIO 引脚驱动成相反的电平，并且不会干扰其他引脚。更多关于切换寄存器的相关信息，详情请参见寄存器功能描述中的端口 x 切换寄存器。

21.5.3 外设在 IO 中的应用

21.5.3.1 可复用为数字量输入的模拟量输入口（AIO）

在 GPIOH 端口中，GPIO 引脚与模拟引脚多路复用，因此这些引脚也称为 AIO。GPIOH 只能在输入模式下工作，不具有数字输出功能。默认情况下，GPIO 高阻抗，且 GPIOH（GPIO224-254）为模拟引脚，这些引脚的数字或模拟操作在 GPHAMSEL 寄存器配置。

21.5.3.2 SPI 的非高速模式

SPI 支持高速模式，可实现高达 40 Mbps 的通信。通过在每个 SPI 的单个 GPIO MUX 选项上使用特殊的 GPIO 配置，以达到尽可能高的速度。这些 SPI 在非高速模式(HSEN=0)时也可以使用 GPIO。

每个引脚可复用多达 12 种不同的外设功能以及通用输入/输出(GPIO)功能。更多有关复用的信息，详情请参见数据手册。

21.5.4 外设复用

引脚复用

通过配置 GPxMUX1/2 和 GPxGMUX1/2 寄存器，以配置 GPIO 引脚的多路复用功能。以 GPIO6 为例，根据 GPAMUX1[GPIO6]和 GPAGMUX1[GPIO6]的不同配置，可以将 GPIO6 设置为不同的四种外设功能或通用数字 I/O。

表格 65 GPIO6 引脚功能配置

GPAGMUX[13:12]	GPAMUX[13:12]	引脚功能
00	00	GPIO6
	01	PWM4_A
	10	OUTPUTXBAR4
	11	SYNCOUT
01	00	GPIO6
	01	QEP1_A
	10	CANB_TX
	11	SPIB_SOMI
10	00	GPIO6
	01	-
	10	-
	11	QSPI_IO0
11	00	GPIO6
	01	-
	10	-
	11	-

注意：

（1） 不同的设备采用的 GPIO 多路复用方案不同。当某个外设在该设备上不可用时，则不可以使用相应的多路复用方案，复用方案在此设备上被保留。

（2） 当选择了一个保留的 GPIO 多路复用方案（即该引脚既不作为 GPIO 引脚，也不配置为外设功能）时，该引脚的被驱动，且引脚的电平状态不确定。在多路复用表中，用“-”和“保留”来表示这些未使用的配置。由于未使用的配置是为了之后的扩展而保留，因此应避免选择未使用的配置。

外设复用配置

通过配置 GPxMUX1/2[GPIOy]和 GPxGMUX1/2[GPIOy]寄存器，还可以将外设分配给不同的引脚。以 OUTPUTXBAR1 为例，根据系统的不同需求，可以选择 GPIO2，GPIO24，GPIO34 或 GPIO58 引脚。

表格 66 OUTPUTXBAR1 多路复用方案

方案	GPxGMUX	GPxMUX	选择的 GPIO
1	GPAGMUX1[5:4]=01	GPAMUX1[5:4]=01	2
2	GPAGMUX2[17:16]=00	GPAMUX2[17:16]=01	24
3	GPBGMUX1[5:4]=00	GPBMUX1[5:4]=01	34
4	GPBGMUX2[21:20]=01	GPBMUX2[21:20]=01	58

注意：当一个以上 GPIO 引脚或所有 GPIO 引脚未被配置为外设输入引脚时，则将未配置的 GPIO 引脚设置为硬连线的默认值。

21.6 GPIO 复用功能配置

表格 67 GPIO 复用功能配置

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO0	-	PWM1_A	-	SPIA_STE	-	-	I2CA_SDA	-	-	-	-	QSPI_IO1	-	-	-	-
GPIO1	-	PWM1_B	-	SPIA_SOMI	-	-	I2CA_SCL	-	-	-	-		-	-	-	-
GPIO2	-	PWM2_A	CANA_TX	SPIA_SIMO	-	OUTPUTXBAR1	PMBUSA_SDA	-	-	UARTA_TX	-	QSPI_IO3	-	-	-	-
GPIO3	-	PWM2_B	OUTPUTXBAR2	CANA_RX	-	OUTPUTXBAR2	PMBUSA_SCL	SPIA_CLK	-	UARTA_RX	-	QSPI_IO2	-	-	-	-
GPIO4	-	PWM3_A	QEP2_STROBE	SPIB_CLK	-	OUTPUTXBAR3	CANA_TX	-	-	-	-	QSPI_SCLK	-	-	-	-
GPIO5	-	PWM3_B	-	OUTPUTXBAR3	-	-	CANA_RX	SPIA_STE	-	-	-	QSPI_IO1	-	-	-	-
GPIO6	-	PWM4_A	OUTPUTXBAR4	SYNCOUT	-	QEP1_A	CANB_TX	SPIB_SOMI	-	-	-	QSPI_IO0	-	-	-	-
GPIO7	-	PWM4_B	-	OUTPUTXBAR5	-	QEP1_B	CANB_RX	SPIB_SIMO	-	-	-	QSPI_SS_N	-	-	-	-

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO8	-	PWM5_A	CANB_TX	ADCSOCAO	-	QEP1_STROBE	UARTA_TX	SPIA_SIMO	-	I2CA_SCL	-	QSPI_IO3	-	-	-	-
GPIO9	-	PWM5_B	UARTB_TX	OUTPUTXBAR6	-	QEP1_INDEX	UARTA_RX	SPIA_CLK	-	-	-	QSPI_IO2	-	-	-	-
GPIO10	-	PWM6_A	CANB_RX	ADCSOCBO	-	QEP1_A	UARTB_TX	SPIA_SOMI	-	I2CA_SDA	-	QSPI_SCLK	-	-	-	-
GPIO11	-	PWM6_B	UARTB_RX	OUTPUTXBAR7	-	QEP1_B	UARTB_RX	SPIA_STE	-	-	QEP2_A	QSPI_IO1	-	SPIA_SIMO	-	-
GPIO12	-	PWM7_A	CANB_TX	SPIA_CLK	-	QEP1_STROBE	UARTB_TX	PMBUSA_CTL	-	-	CANA_RX	QSPI_IO0	-	-	-	TRACED2
GPIO13	-	PWM7_B	CANB_RX	SPIA_SOMI	-	QEP1_INDEX	UARTB_RX	PMBUSA_ALERT	-	-	CANA_TX	QSPI_SS_N	-	-	-	TRACED3
GPIO14	-	PWM8_A	UARTB_TX	-	-	PWM3_A	OUTPUTXBAR3	PMBUSA_SDA	-	SPIB_CLK	QEP2_A	-	-	-	-	-
GPIO15	-	PWM8_B	UARTB_RX	-	-	PWM3_B	OUTPUTXBAR4	PMBUSA_SCL	-	SPIB_STE	QEP2_B	-	-	-	-	-
GPIO16	-	SPIA_SIMO	CANB_TX	OUTPUTXBAR7	-	PWM5_A	UARTA_TX	SD1_D1	-	QEP1_STROBE	PMBUSA_SCL	XCLKOUT	-	SPIB_SOMI	QEP2_B	TRACED0

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO17	-	SPIA_SOMI	CANB_RX	OUTPUTXBAR8	-	PWM5_B	UARTA_RX	SD1_C1	-	QEP1_INDEX	PMBUSA_SDA	-	-	CANA_TX	-	TRGIO
GPIO18_X2	-	SPIA_CLK	UARTB_TX	CANA_RX	-	PWM6_A	I2CA_SCL	SD1_D2	-	QEP2_A	PMBUSA_CTL	XCLKOUT	-	-	-	-
GPIO20	-	QEP1_A	-	CANB_TX	-		SPIB_SIMO	SD1_D3	-	SPIB_CLK	-	-	-	-	-	-
GPIO21	-	QEP1_B	-	CANB_RX	-		SPIB_SOMI	SD1_C3	-		-	I2CA_SCL	-	-	-	-
GPIO22	-	QEP1_STROBE	-	UARTB_TX	-	PWM4_A	SPIB_CLK	SD1_D4	-	LINA_TX	-	-	-	-	-	-
GPIO23	-	QEP1_INDEX	-	UARTB_RX	-	PWM4_B	SPIB_STE	SD1_C4	-	LINA_RX	-	I2CA_SDA	-	SPIB_SIMO	-	-
GPIO24	-	OUTPUTXBAR1	QEP2_A	-	-	PWM8_A	SPIB_SIMO	SD1_D1	-	-	PMBUSA_SCL	UARTA_TX	-	ERROR STS	-	TRACE CLK
GPIO25	-	OUTPUTXBAR2	QEP2_B	-	-	QEP1_A	SPIB_SOMI	SD1_C1	-	-	PMBUSA_SDA	UARTA_RX	-	QSPI_IO3	-	-
GPIO26	-	OUTPUTXBAR3	QEP2_INDEX	-	-	OUTPUTXBAR3	SPIB_CLK	SD1_D2	-	-	PMBUSA_CTL	I2CA_SDA	-	QSPI_IO2	-	-
GPIO27	-	OUTPUTXBAR4	QEP2_STROBE	-	-	OUTPUTXBAR4	SPIB_STE	SD1_C2	-	-	PMBUSA_ALERT	I2CA_SCL	-	QSPI_SCLK	-	-
GPIO28	-	UARTA_RX	-	PWM7_A	-	OUTPUTXBAR5	QEP1_A	SD1_D3	-	QEP2_STROBE	LINA_TX	SPIB_CLK	-	ERROR STS	-	TRACE CLK

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO29	-	UARTA_TX	-	PWM7_B	-	OUTPUTXBAR6	QEP1_B	SD1_C3	-	QEP2_INDEX	LINA_RX	SPIB_STE	-	ERRORSTS	-	TRACED0
GPIO30	-	CANA_RX	PWM1_A	SPIB_SIMO	-	OUTPUTXBAR7	QEP1_STROBE	SD1_D4	-	-	-	QSPI_IO3	-	-	-	TRACED1
GPIO31	-	CANA_TX	PWM1_B	SPIB_SOMI	-	OUTPUTXBAR8	QEP1_INDEX	SD1_C4	-	-	-	QSPI_IO1	-	-	-	TRACED2
GPIO32	-	I2CA_SDA	-	SPIB_CLK	-	PWM8_B	LINA_TX	SD1_D3	-	-	CANA_TX	QSPI_IO0	-	ADCSOCBO	-	TRGIO
GPIO33	-	I2CA_SCL	QEP2_B	SPIB_STE	-	OUTPUTXBAR4	LINA_RX	SD1_C3	-	-	CANA_RX	QSPI_SS_N	-	ADCSOCAO	-	TRACED1
GPIO34	-	OUTPUTXBAR1	-	-	-	-	PMBUSA_SDA	-	-	-	-	-	-	-	-	TRACED3
GPIO35	-	UARTA_RX	-	I2CA_SDA	-	CANA_RX	PMBUSA_SCL	LINA_RX	-	QEP1_A	PMBUSA_CTL	-	-	-	-	TDI
GPIO37	-	OUTPUTXBAR2	-	I2CA_SCL	-	UARTA_TX	CANA_TX	LINA_TX	-	QEP1_B	PMBUSA_ALERT	-	-	-	-	TDO
GPIO39	-	-	-	-	-	-	CANB_RX	-	-	-	-	QSPI_IO3	-	-	-	-
GPIO40	-	-	-	-	-	-	PMBUSA_SDA	-	-	UARTB_TX	QEP1_A	QSPI_IO2	-	-	-	-

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO41	-	PWM2_A	-	SPIB_STE	-		PMBUSA_SCL	-	-	UARTB_RX	QEP1_B	QSPI_SCLK	-	SPIB_SOMI	-	-
GPIO42	-	-	LINA_RX	OUTPUTXBAR5	-	PMBUSA_CTL	I2CA_SDA	-	-	UARTB_TX	QEP1_STROBE	-	-	-	-	-
GPIO43	-	-	-	OUTPUTXBAR6	-	PMBUSA_ALERT	I2CA_SCL	-	-	UARTB_RX	QEP1_INDEX	-	-	-	-	-
GPIO44	-	-	QEP1_A	OUTPUTXBAR7	-	-	-	-	-	-	-	QSPI_IO2	-	-	-	-
GPIO45	-	-	-	OUTPUTXBAR8	-	-	-	-	-	-	-	QSPI_IO0	-	-	-	-
GPIO46	-	-	-	LINA_TX	-	-	-	-	-	-	-	QSPI_SS_N	-	-	-	-
GPIO47	-	PWM2_B	QEP1_A	LINA_RX	-	-	SPIB_SIMO	PMBUSA_SDA	-	-	-	-	-	-	-	-
GPIO48	-	OUTPUTXBAR3	-	CANB_TX	-	-	UARTA_TX	SD1_D1	-	-	-	-	-	-	-	-
GPIO49	-	OUTPUTXBAR4	QEP1_INDEX	CANB_RX	-	-	UARTA_RX	SD1_C1	-	QEP2_INDEX	-	SYNCOUT	-	-	-	-
GPIO50	-	QEP1_A	-	-	-	-	SPIB_SIMO	SD1_D2	-	-	-	-	-	-	-	-

Port	AF0	AF1	AF2	AF3	AF4	AF5	AF6	AF7	AF8	AF9	AF10	AF11	AF12	AF13	AF14	AF15
GPIO51	-	QEP1_B	-	-	-	-	SPIB_SOMI	SD1_C2	-	-	-	-	-	-	-	-
GPIO52	-	QEP1_STROBE	-	-	-	-	SPIB_CLK	SD1_D3	-	-	-	-	-	-	-	-
GPIO53	-	QEP1_INDEX	-	-	-	-	SPIB_STE	SD1_C3	-	UARTB_RX	-	-	-	-	-	-
GPIO54	-	SPIA_SIMO	-	-	-	QEP2_A	UARTB_TX	SD1_D4	-		-	-	-	-	-	-
GPIO55	-	SPIA_SOMI	-	-	-	QEP2_B	UARTB_RX	SD1_C4	-		-	-	-	-	-	-
GPIO56	-	SPIA_CLK	-	-	-	QEP2_STROBE	UARTB_TX	SD1_D3	-	SPIB_SIMO	-	QEP1_A	-	-	-	-
GPIO57	-	SPIA_STE	-	-	-	QEP2_INDEX	UARTB_RX	SD1_C3	-	SPIB_SOMI	-	QEP1_B	-	-	-	-
GPIO58	-	-	-	-	-	OUTPUTXBAR1	SPIB_CLK	SD1_D4	-	LINA_TX	CANB_TX	QEP1_STROBE	-	-	-	-
GPIO59	-	-	-	-	-	OUTPUTXBAR2	SPIB_STE	SD1_C4	-	LINA_RX	CANB_RX	QEP1_INDEX	-	-	-	-

21.7 寄存器组地址

表格 68 GPIO 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
GpioCtrlRegs	GPIO_CTRL_REGS	0x4003 0000	0x4003 07FF
GpioDataRegs	GPIO_CTRL_REGS	0x4003 0800	0x4003 08FF

21.8 寄存器地址映射

表格 69 GPIO_CTRL 地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
GPACTRL	端口 A 采样周期寄存器	0x00	√
GPAQSEL1	端口 A 输入类型寄存器 1	0x04	√
GPAQSEL2	端口 A 输入类型寄存器 2	0x08	√
GPAMUX1	端口 A 复用低位寄存器 1	0x0C	√
GPAMUX2	端口 A 复用低位寄存器 2	0x10	√
GPADIR	端口 A 数据方向寄存器	0x14	√
GPAPUD	端口 A 上拉禁用寄存器	0x18	√
GPAINV	端口 A 输入反相寄存器	0x20	√
GPAODR	端口 A 开漏输出寄存器	0x24	√
GPAAMSEL	端口 A 模拟模式选择寄存器	0x28	√
GPADSEL1	端口 A 驱动能力选择寄存器 1	0x2C	√
GPADSEL2	端口 A 驱动能力选择寄存器 2	0x30	√
GPAFEN	端口 A 最大输出频率使能寄存器	0x34	√
GPAGMUX1	端口 A 复用高位寄存器 1	0x40	√
GPAGMUX2	端口 A 复用高位寄存器 2	0x44	√
GPACSEL1	端口 A 主核选择寄存器 1	0x50	√
GPACSEL2	端口 A 主核选择寄存器 2	0x54	√
GPACSEL3	端口 A 主核选择寄存器 3	0x58	√
GPACSEL4	端口 A 主核选择寄存器 4	0x5C	√
GPALOCK	端口 A 锁定寄存器	0x78	√
GPACR	端口 A 提交锁定寄存器	0x7C	√
GPBCTRL	端口 B 采样周期寄存器	0x80	√
GPBQSEL1	端口 B 输入类型寄存器 1	0x84	√
GPBQSEL2	端口 B 输入类型寄存器 2	0x88	√

寄存器名称	寄存器描述	偏移地址	WRPRT
GPBMUX1	端口 B 复用低位寄存器 1	0x8C	√
GPBMUX2	端口 B 复用低位寄存器 2	0x90	√
GPBDIR	端口 B 数据方向寄存器	0x94	√
GPBPUD	端口 B 上拉禁用寄存器	0x98	√
GPBINV	端口 B 输入反相寄存器	0xA0	√
GPBODR	端口 B 开漏输出寄存器	0xA4	√
GPBDSEL1	端口 B 驱动能力选择寄存器 1	0xAC	√
GPBDSEL2	端口 B 驱动能力选择寄存器 2	0xB0	√
GPBFEN	端口 B 最大输出频率使能寄存器	0xB4	√
GPBGMUX1	端口 B 复用高位寄存器 1	0xC0	√
GPBGMUX2	端口 B 复用高位寄存器 2	0xC4	√
GPBCSEL1	端口 B 主核选择寄存器 1	0xD0	√
GPBCSEL2	端口 B 主核选择寄存器 2	0xD4	√
GPBCSEL3	端口 B 主核选择寄存器 3	0xD8	√
GPBCSEL4	端口 B 主核选择寄存器 4	0xDC	√
GPBLOCK	端口 B 锁定寄存器	0xF8	√
GPBCR	端口 B 提交锁定寄存器	0xFC	√
GPHCTRL	端口 H 采样周期寄存器	0x380	√
GPHQSEL1	端口 H 输入类型寄存器 1	0x384	√
GPHQSEL2	端口 H 输入类型寄存器 2	0x388	√
GPHPUD	端口 H 上拉禁用寄存器	0x398	√
GPHINV	端口 H 输入反相寄存器	0x3A0	√
GPHAMSEL	端口 H 模拟模式选择寄存器	0x3A8	√
GPHLOCK	端口 H 锁定寄存器	0x3F8	√
GPHCR	端口 H 提交锁定寄存器	0x3FC	√

表格 70 GPIO_DATA 地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
GPADAT	端口 A 数据寄存器	0x00	-
GPASET	端口 A 设置寄存器	0x04	-
GPACLR	端口 A 清除寄存器	0x08	-
GPATOGGLE	端口 A 切换寄存器	0x0C	-
GPBDAT	端口 B 数据寄存器	0x10	-
GPBSET	端口 B 设置寄存器	0x14	-

寄存器名称	寄存器描述	偏移地址	WRPRT
GPBCLR	端口 B 清除寄存器	0x18	-
GPBTOGGLE	端口 B 切换寄存器	0x1C	-
GPHDAT	端口 H 数据寄存器	0x70	-

21.9 寄存器功能描述

21.9.1 端口 A 采样周期寄存器（GPACTRL）

偏移地址：0x00

复位类型：SYSRSn

GPIOA 引脚的限定采样周期（GPIO0 到 GPIO31）。每个位域 8 个 GPIO 引脚构成的组，为每个组选择 SYSCLK 周期中的限定采样周期。

位/域	名称	R/W	描述	复位值
7:0	QUALPRD0	R/W	GPIO0 到 GPIO7 的采样周期（GPIO0 to GPIO7 sampling cycle） 选择 GPIO0 到 GPIO7 的采样周期。 0000 0000：周期=$T_{SYSCLKOUT}$*1 0000 0001：周期=$T_{SYSCLKOUT}$*2 0000 0010：周期=$T_{SYSCLKOUT}$*4 …… 1111 1111：周期=$T_{SYSCLKOUT}$*510	0h
15:8	QUALPRD1	R/W	GPIO8 到 GPIO15 的采样周期（GPIO8 to GPIO15 sampling cycle） 具体描述参考 QUALPRD0。	0h
23:16	QUALPRD2	R/W	GPIO16 到 GPIO23 的采样周期（GPIO16 to GPIO23 sampling cycle） 具体描述参考 QUALPRD0。	0h
31:24	QUALPRD3	R/W	GPIO24 到 GPIO31 的采样周期（GPIO24 to GPIO31 sampling cycle） 具体描述参考 QUALPRD0。	0h

21.9.2 端口 A 输入类型寄存器 1（GPAQSEL1）

偏移地址：0x04

复位类型：SYSRSn

GPIOA 引脚的输入限定类型（GPIO0 到 GPIO15）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
2y+1:2y	GPIOy	R/W	GPIOy 的输入类型（GPIOy Input type）(y=0…15) 00：同步输入限定 01：3 采样窗口输入限定 10：6 采样窗口输入限定 11：异步输入限定	0h

21.9.3 端口 A 输入类型寄存器 2（GPAQSEL2）

偏移地址：0x08

复位类型：SYSRSn

GPIOA 引脚的输入限定类型（GPIO16 到 GPIO31）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
1:0	GPIO16	R/W	GPIO16 的输入类型（GPIO16 Input type） 00：同步输入限定 01：3 采样窗口输入限定 10：6 采样窗口输入限定 11：异步输入限定	0h
3:2	GPIO17	R/W	GPIO17 的输入类型（GPIO17 Input type） 具体描述参考 GPIO16。	0h
5:4	GPIO18	R/W	GPIO18 的输入类型（GPIO18 Input type） 具体描述参考 GPIO16。	0h
7:6	GPIO19	R/W	GPIO19 的输入类型（GPIO19 Input type） 具体描述参考 GPIO16。	0h
9:8	GPIO20	R/W	GPIO20 的输入类型（GPIO20 Input type） 具体描述参考 GPIO16。	0h
11:10	GPIO21	R/W	GPIO21 的输入类型（GPIO21 Input type） 具体描述参考 GPIO16。	0h
13:12	GPIO22	R/W	GPIO22 的输入类型（GPIO22 Input type） 具体描述参考 GPIO16。	0h
15:14	GPIO23	R/W	GPIO23 的输入类型（GPIO23 Input type） 具体描述参考 GPIO16。	0h
17:16	GPIO24	R/W	GPIO24 的输入类型（GPIO24 Input type） 具体描述参考 GPIO16。	0h
19:18	GPIO25	R/W	GPIO25 的输入类型（GPIO25 Input type） 具体描述参考 GPIO16。	0h
21:20	GPIO26	R/W	GPIO26 的输入类型（GPIO26 Input type） 具体描述参考 GPIO16。	0h
23:22	GPIO27	R/W	GPIO27 的输入类型（GPIO27 Input type） 具体描述参考 GPIO16。	0h
25:24	GPIO28	R/W	GPIO28 的输入类型（GPIO28 Input type） 具体描述参考 GPIO16。	0h
27:26	GPIO29	R/W	GPIO29 的输入类型（GPIO29 Input type） 具体描述参考 GPIO16。	0h
29:28	GPIO30	R/W	GPIO30 的输入类型（GPIO30 Input type） 具体描述参考 GPIO16。	0h
31:30	GPIO31	R/W	GPIO31 的输入类型（GPIO31 Input type） 具体描述参考 GPIO16。	0h

21.9.4 端口 A 复用低位寄存器 1（GPAMUX1）

偏移地址：0x0C

复位值：0x0000 0000

复位类型：SYSRSn

此寄存器配置 GPIOA 引脚多路复用功能的低位部分（GPIO0 到 GPIO15），当位域值为 0x00，0x04，0x08，0x0C 时，对应的引脚被配置为 GPIO 模式，位域值为其他值时，选择外设控制该引脚。有关每个引脚的配置信息，详情请参见数据手册。

在更改 GPAGMUX1 寄存器的位域前，必须使用此寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
2y+1:2y	GPIOy	R/W	GPIOy 的外设多路复用配置的低两位（GPIOy peripheral mux configuration in the low 2 bits）(y=0…15)	0h

21.9.5 端口 A 复用低位寄存器 2（GPAMUX2）

偏移地址：0x10

复位类型：SYSRSn

此寄存器配置 GPIOA 引脚多路复用功能的低位部分（GPIO16 到 GPIO31），当位域值为 0x00，0x04，0x08，0x0C 时，对应的引脚被配置为 GPIO 模式，位域值为其他值时，选择外设控制该引脚。有关每个引脚的配置信息，详情请参见数据手册。

在更改 GPAGMUX2 寄存器的对应位域前，必须使用此寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO16	R/W	GPIO16 的外设多路复用配置的低两位（GPIO16 peripheral mux configuration in the low 2 bits）	0h
3:2	GPIO17	R/W	GPIO17 的外设多路复用配置的低两位（GPIO17 peripheral mux configuration in the low 2 bits）	0h
5:4	GPIO18	R/W	GPIO18 的外设多路复用配置的低两位（GPIO18 peripheral mux configuration in the low 2 bits）	0h
7:6	GPIO19	R/W	GPIO19 的外设多路复用配置的低两位（GPIO19 peripheral mux configuration in the low 2 bits）	0h
9:8	GPIO20	R/W	GPIO20 的外设多路复用配置的低两位（GPIO20 peripheral mux configuration in the low 2 bits）	0h
11:10	GPIO21	R/W	GPIO21 的外设多路复用配置的低两位（GPIO21 peripheral mux configuration in the low 2 bits）	0h
13:12	GPIO22	R/W	GPIO22 的外设多路复用配置的低两位（GPIO22 peripheral mux configuration in the low 2 bits）	0h
15:14	GPIO23	R/W	GPIO23 的外设多路复用配置的低两位（GPIO23 peripheral mux configuration in the low 2 bits）	0h
17:16	GPIO24	R/W	GPIO24 的外设多路复用配置的低两位（GPIO24 peripheral mux configuration in the low 2 bits）	0h
19:18	GPIO25	R/W	GPIO25 的外设多路复用配置的低两位（GPIO25 peripheral mux configuration in the low 2 bits）	0h
21:20	GPIO26	R/W	GPIO26 的外设多路复用配置的低两位（GPIO26 peripheral mux configuration in the low 2 bits）	0h
23:22	GPIO27	R/W	GPIO27 的外设多路复用配置的低两位（GPIO27 peripheral mux configuration in the low 2 bits）	0h
25:24	GPIO28	R/W	GPIO28 的外设多路复用配置的低两位（GPIO28 peripheral mux configuration in the low 2 bits）	0h

位/域	名称	R/W	描述	复位值
27:26	GPIO29	R/W	GPIO29 的外设多路复用配置的低两位（GPIO29 peripheral mux configuration in the low 2 bits）	0h
29:28	GPIO30	R/W	GPIO30 的外设多路复用配置的低两位（GPIO30 peripheral mux configuration in the low 2 bits）	0h
31:30	GPIO31	R/W	GPIO31 的外设多路复用配置的低两位（GPIO31 peripheral mux configuration in the low 2 bits）	0h

21.9.6 端口 A 数据方向寄存器（GPADIR）

偏移地址：0x14

复位类型：SYSRSn

此寄存器配置 GPIOA 引脚的数据方向。若引脚未配置成 GPIO 模式，则此寄存器无效。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	GPIOy 的数据方向（GPIOy Data direction）(y=0…31) 配置 GPIO 模式下引脚的数据方向。 0：输入 1：输出	0h

21.9.7 端口 A 上拉禁用寄存器（GPAPUD）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	GPIOy 的上拉禁用（GPIOy Pull-up disable）(y=0…31) 配置 GPIOy 引脚的内部上拉电阻状态。 0：使能内部上拉电阻 1：禁用内部上拉电阻	1h

21.9.8 端口 A 输入反相寄存器（GPAINV）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	GPIOy 的输入反相（GPIOy Input inversion）(y=0…31) 配置 GPIOy 引脚的输入值是否通过反相器。 0：输入值不反相 1：输入值反相	0h

21.9.9 端口 A 开漏输出寄存器（GPAODR）

偏移地址：0x24

复位类型：SYSRSn

选择通用输出引脚的输出模式：推挽输出或开漏输出。在推挽输出模式中，向输出数据锁存器写入 1 将驱动引脚为高电平。在开漏模式下，向输出数据锁存器写入 1 使输出缓冲器处于三态。在两种输出模式中，向输出数据锁存器写入 0 将驱动引脚为低电平。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	GPIOy 的开漏输出（GPIOy Open-drain output mode）(y=0…31) 0：推挽输出 1：开漏输出	0h

21.9.10 端口 A 模拟模式选择寄存器（GPAAMSEL）

偏移地址：0x28

复位类型：SYSRSn

选择 GPIOA 引脚的模拟模式（GPIO22 和 GPIO23）。复位时，GPIO22 和 GPIO23 引脚处于特殊的模拟模式，必须清除其在 GPAAMSEL 寄存器中的位来重新配置为 GPIO 使用。

位/域	名称	R/W	描述	复位值
21:0	保留			0h
22	GPIO22	R/W	选择 GPIO22 的模拟模式（Analog mode select for GPIO22） 选择 GPIO22 为模拟模式或数字模式。 0：数字模式 1：模拟模式	1h
23	GPIO23	R/W	选择 GPIO23 的模拟模式（Analog mode select for GPIO23） 选择 GPIO23 为模拟模式或数字模式。 0：数字模式 1：模拟模式	1h
31:24	保留			0h

21.9.11 端口 A 驱动能力选择寄存器 1（GPADSEL1）

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	GPIO0	R/W	选择 GPIO0 的驱动能力（Driver capability select for GPIO0） 当 GPAFEN[GPIO0]=1 时，最大输出频率为 25MHz； 当 GPAFEN[GPIO0]=0 时，驱动能力如下： 00：最大输出频率为 30MHz，驱动能力 1mA 01：最大输出频率为 50MHz，驱动能力 4mA 10：最大输出频率为 40MHz，驱动能力 2mA 11：最大输出频率为 60MHz，驱动能力 6mA 注意：驱动能力不受 GPAFEN 寄存器影响。	0h
3:2	GPIO1	R/W	选择 GPIO1 的驱动能力（Driver capability select for GPIO1） 具体描述参考 GPIO0。	0h
5:4	GPIO2	R/W	选择 GPIO2 的驱动能力（Driver capability select for GPIO2） 具体描述参考 GPIO0。	0h
7:6	GPIO3	R/W	选择 GPIO3 的驱动能力（Driver capability select for GPIO3） 具体描述参考 GPIO0。	0h

位/域	名称	R/W	描述	复位值
9:8	GPIO4	R/W	选择 GPIO4 的驱动能力（Driver capability select for GPIO4） 具体描述参考 GPIO0。	0h
11:10	GPIO5	R/W	选择 GPIO5 的驱动能力（Driver capability select for GPIO5） 具体描述参考 GPIO0。	0h
13:12	GPIO6	R/W	选择 GPIO6 的驱动能力（Driver capability select for GPIO6） 具体描述参考 GPIO0。	0h
15:14	GPIO7	R/W	选择 GPIO7 的驱动能力（Driver capability select for GPIO7） 具体描述参考 GPIO0。	0h
17:16	GPIO8	R/W	选择 GPIO8 的驱动能力（Driver capability select for GPIO8） 具体描述参考 GPIO0。	0h
19:18	GPIO9	R/W	选择 GPIO9 的驱动能力（Driver capability select for GPIO9） 具体描述参考 GPIO0。	0h
21:20	GPIO10	R/W	选择 GPIO10 的驱动能力（Driver capability select for GPIO10） 具体描述参考 GPIO0。	0h
23:22	GPIO11	R/W	选择 GPIO11 的驱动能力（Driver capability select for GPIO11） 具体描述参考 GPIO0。	0h
25:24	GPIO12	R/W	选择 GPIO12 的驱动能力（Driver capability select for GPIO12） 具体描述参考 GPIO0。	0h
27:26	GPIO13	R/W	选择 GPIO13 的驱动能力（Driver capability select for GPIO13） 具体描述参考 GPIO0。	0h
29:28	GPIO14	R/W	选择 GPIO14 的驱动能力（Driver capability select for GPIO14） 具体描述参考 GPIO0。	0h
31:30	GPIO15	R/W	选择 GPIO15 的驱动能力（Driver capability select for GPIO15） 具体描述参考 GPIO0。	0h

21.9.12 端口 A 驱动能力选择寄存器 2（GPADSEL2）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	GPIO16	R/W	选择 GPIO16 的驱动能力（Driver capability select for GPIO16） 当 GPAFEN[GPIO16]=1 时，最大输出频率为 25MHz； 当 GPAFEN[GPIO16]=0 时，驱动能力如下： 00：最大输出频率为 30MHz，驱动能力 1mA 01：最大输出频率为 50MHz，驱动能力 4mA 10：最大输出频率为 40MHz，驱动能力 2mA 11：最大输出频率为 60MHz，驱动能力 6mA 注意：驱动能力不受 GPAFEN 寄存器影响。	0h

位/域	名称	R/W	描述	复位值
3:2	GPIO17	R/W	选择 GPIO17 的驱动能力（Driver capability select for GPIO17） 具体描述参考 GPIO16。	0h
5:4	GPIO18	R/W	选择 GPIO18 的驱动能力（Driver capability select for GPIO18） 具体描述参考 GPIO16。	0h
7:6	GPIO19	R/W	选择 GPIO19 的驱动能力（Driver capability select for GPIO19） 具体描述参考 GPIO16。	0h
9:8	GPIO20	R/W	选择 GPIO20 的驱动能力（Driver capability select for GPIO20） 具体描述参考 GPIO16。	0h
11:10	GPIO21	R/W	选择 GPIO21 的驱动能力（Driver capability select for GPIO21） 具体描述参考 GPIO16。	0h
13:12	GPIO22	R/W	选择 GPIO22 的驱动能力（Driver capability select for GPIO22） 具体描述参考 GPIO16。	0h
15:14	GPIO23	R/W	选择 GPIO23 的驱动能力（Driver capability select for GPIO23） 具体描述参考 GPIO16。	0h
17:16	GPIO24	R/W	选择 GPIO24 的驱动能力（Driver capability select for GPIO24） 具体描述参考 GPIO16。	0h
19:18	GPIO25	R/W	选择 GPIO25 的驱动能力（Driver capability select for GPIO25） 具体描述参考 GPIO16。	0h
21:20	GPIO26	R/W	选择 GPIO26 的驱动能力（Driver capability select for GPIO26） 具体描述参考 GPIO16。	0h
23:22	GPIO27	R/W	选择 GPIO27 的驱动能力（Driver capability select for GPIO27） 具体描述参考 GPIO16。	0h
25:24	GPIO28	R/W	选择 GPIO28 的驱动能力（Driver capability select for GPIO28） 具体描述参考 GPIO16。	0h
27:26	GPIO29	R/W	选择 GPIO29 的驱动能力（Driver capability select for GPIO29） 具体描述参考 GPIO16。	0h
29:28	GPIO30	R/W	选择 GPIO30 的驱动能力（Driver capability select for GPIO30） 具体描述参考 GPIO16。	0h
31:30	GPIO31	R/W	选择 GPIO31 的驱动能力（Driver capability select for GPIO31） 具体描述参考 GPIO16。	0h

21.9.13 端口 A 最大输出频率使能寄存器（GPAFEN）

偏移地址：0x34

复位类型：SYSRSn

y=0~31

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	使能 GPIOy 的最大输出频率（GPIOy Maximum Output Frequency Enable）（y=0~31） 0：GPIOy 的最大输出频率详情请参见 GPADSEL1/2[GPIOy] 1：GPIOy 的最大输出频率为 25MHz	0h

21.9.14 端口 A 复用高位寄存器 1（GPAGMUX1）

偏移地址：0x40

复位类型：SYSRSn

此寄存器配置 GPIOA 引脚多路复用功能的高位部分（GPIO0 到 GPIO15）。有关每个引脚的配置信息，详情请参见数据手册。

在更改此寄存器的位域前，必须使用 GPAMUX1 寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
2y+1:2y	GPIOy	R/W	GPIOy 的外设多路复用配置的高两位（GPIOy peripheral mux configuration in the High 2 bits）(y=0…15)	0h

21.9.15 端口 A 复用高位寄存器 2（GPAGMUX2）

偏移地址：0x44

复位类型：SYSRSn

此寄存器配置 GPIOA 引脚多路复用功能的高位部分（GPIO16 到 GPIO31）。有关每个引脚的配置信息，详情请参见数据手册。

在更改此寄存器的位域前，必须使用 GPAMUX2 寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO16	R/W	GPIO16 的外设多路复用配置的高两位（GPIO16 peripheral mux configuration in the High 2 bits）	0h
3:2	GPIO17	R/W	GPIO17 的外设多路复用配置的高两位（GPIO17 peripheral mux configuration in the High 2 bits）	0h
5:4	GPIO18	R/W	GPIO18 的外设多路复用配置的高两位（GPIO18 peripheral mux configuration in the High 2 bits）	0h
7:6	GPIO19	R/W	GPIO19 的外设多路复用配置的高两位（GPIO19 peripheral mux configuration in the High 2 bits）	0h
9:8	GPIO20	R/W	GPIO20 的外设多路复用配置的高两位（GPIO20 peripheral mux configuration in the High 2 bits）	0h
11:10	GPIO21	R/W	GPIO21 的外设多路复用配置的高两位（GPIO21 peripheral mux configuration in the High 2 bits）	0h
13:12	GPIO22	R/W	GPIO22 的外设多路复用配置的高两位（GPIO22 peripheral mux configuration in the High 2 bits）	0h
15:14	GPIO23	R/W	GPIO23 的外设多路复用配置的高两位（GPIO23 peripheral mux configuration in the High 2 bits）	0h

位/域	名称	R/W	描述	复位值
17:16	GPIO24	R/W	GPIO24 的外设多路复用配置的高两位（GPIO24 peripheral mux configuration in the High 2 bits）	0h
19:18	GPIO25	R/W	GPIO25 的外设多路复用配置的高两位（GPIO25 peripheral mux configuration in the High 2 bits）	0h
21:20	GPIO26	R/W	GPIO26 的外设多路复用配置的高两位（GPIO26 peripheral mux configuration in the High 2 bits）	0h
23:22	GPIO27	R/W	GPIO27 的外设多路复用配置的高两位（GPIO27 peripheral mux configuration in the High 2 bits）	0h
25:24	GPIO28	R/W	GPIO28 的外设多路复用配置的高两位（GPIO28 peripheral mux configuration in the High 2 bits）	0h
27:26	GPIO29	R/W	GPIO29 的外设多路复用配置的高两位（GPIO29 peripheral mux configuration in the High 2 bits）	0h
29:28	GPIO30	R/W	GPIO30 的外设多路复用配置的高两位（GPIO30 peripheral mux configuration in the High 2 bits）	0h
31:30	GPIO31	R/W	GPIO31 的外设多路复用配置的高两位（GPIO31 peripheral mux configuration in the High 2 bits）	0h

21.9.16 端口 A 主核选择寄存器 1（GPACSEL1）

偏移地址：0x50

复位类型：SYSRSn

此寄存器选择 GPIOA 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPADAT，GPASET，GPACLR，GPATOGGLE 四个寄存器以控制引脚。（读）GPADAT 时始终转入 CPU0 和 CPU1 中。

位/域	名称	R/W	描述	复位值
4y+3:4y	GPIOy	R/W	选择 GPIOy 的主核（GPIOy Master core select）(y=0…7) 00：选择 CPU0 为主核 01：选择 CPU1 为主核 其他：保留	0h

21.9.17 端口 A 主核选择寄存器 2（GPACSEL2）

偏移地址：0x54

复位类型：SYSRSn

此寄存器选择 GPIOA 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPADAT，GPASET，GPACLR，GPATOGGLE 四个寄存器以控制引脚。（读）GPADAT 时始终转入 CPU0 和 CPU1 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO8	R/W	选择 GPIO8 的主核（GPIO8 Master core select） 00：选择 CPU0 为主核 01：选择 CPU1 为主核 其他：保留	0h
7:4	GPIO9	R/W	选择 GPIO9 的主核（GPIO9 Master core select） 具体描述参考 GPIO8。	0h
11:8	GPIO10	R/W	选择 GPIO10 的主核（GPIO10 Master core select） 具体描述参考 GPIO8。	0h

位/域	名称	R/W	描述	复位值
15:12	GPIO11	R/W	选择 GPIO11 的主核（GPIO11 Master core select） 具体描述参考 GPIO8。	0h
19:16	GPIO12	R/W	选择 GPIO12 的主核（GPIO12 Master core select） 具体描述参考 GPIO8。	0h
23:20	GPIO13	R/W	选择 GPIO13 的主核（GPIO13 Master core select） 具体描述参考 GPIO8。	0h
27:24	GPIO14	R/W	选择 GPIO14 的主核（GPIO14 Master core select） 具体描述参考 GPIO8。	0h
31:28	GPIO15	R/W	选择 GPIO15 的主核（GPIO15 Master core select） 具体描述参考 GPIO8。	0h

21.9.18 端口 A 主核选择寄存器 3（GPACSEL3）

偏移地址：0x58

复位类型：SYSRSn

此寄存器选择 GPIOA 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPADAT，GPASET，GPACLR，GPATOGGLE 四个寄存器以控制引脚。（读）GPADAT 时始终转入 CPU0 和 CPU1 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO16	R/W	选择 GPIO16 的主核（GPIO16 Master core select） 具体描述参考 GPIO8。	0h
7:4	GPIO17	R/W	选择 GPIO17 的主核（GPIO17 Master core select） 具体描述参考 GPIO8。	0h
11:8	GPIO18	R/W	选择 GPIO18 的主核（GPIO18 Master core select） 具体描述参考 GPIO8。	0h
15:12	GPIO19	R/W	选择 GPIO19 的主核（GPIO19 Master core select） 具体描述参考 GPIO8。	0h
19:16	GPIO20	R/W	选择 GPIO20 的主核（GPIO20 Master core select） 具体描述参考 GPIO8。	0h
23:20	GPIO21	R/W	选择 GPIO21 的主核（GPIO21 Master core select） 具体描述参考 GPIO8。	0h
27:24	GPIO22	R/W	选择 GPIO22 的主核（GPIO22 Master core select） 具体描述参考 GPIO8。	0h
31:28	GPIO23	R/W	选择 GPIO23 的主核（GPIO23 Master core select） 具体描述参考 GPIO8。	0h

21.9.19 端口 A 主核选择寄存器 4（GPACSEL4）

偏移地址：0x5C

复位类型：SYSRSn

此寄存器选择 GPIOA 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPADAT，GPASET，GPACLR，GPATOGGLE 四个寄存器以控制引脚。（读）GPADAT 时始终转入 CPU0 和 CPU1 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO24	R/W	选择 GPIO24 的主核（GPIO24 Master core select） 具体描述参考 GPIO8。	0h

位/域	名称	R/W	描述	复位值
7:4	GPIO25	R/W	选择 GPIO25 的主核（GPIO25 Master core select） 具体描述参考 GPIO8。	0h
11:8	GPIO26	R/W	选择 GPIO26 的主核（GPIO26 Master core select） 具体描述参考 GPIO8。	0h
15:12	GPIO27	R/W	选择 GPIO27 的主核（GPIO27 Master core select） 具体描述参考 GPIO8。	0h
19:16	GPIO28	R/W	选择 GPIO28 的主核（GPIO28 Master core select） 具体描述参考 GPIO8。	0h
23:20	GPIO29	R/W	选择 GPIO29 的主核（GPIO29 Master core select） 具体描述参考 GPIO8。	0h
27:24	GPIO30	R/W	选择 GPIO30 的主核（GPIO30 Master core select） 具体描述参考 GPIO8。	0h
31:28	GPIO31	R/W	选择 GPIO31 的主核（GPIO31 Master core select） 具体描述参考 GPIO8。	0h

21.9.20 端口 A 锁定寄存器（GPALOCK）

偏移地址：0x78

复位类型：SYSRSn

锁定 GPIOA 引脚的配置（GPIO0 到 GPIO31）。此寄存器将写入 GPAMUX1/2，GPAGMUX1/2，GPADIR，GPAINV，GPAODR 和 GPACSEL1/2/3/4 寄存器中。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	配置 GPIOy 锁定（GPIOy Configuration lock）(y=0…31) 0：不锁定 GPIOy 的配置 1：锁定 GPIOy 的配置	0h

21.9.21 端口 A 提交锁定寄存器（GPACR）

偏移地址：0x7C

复位类型：SYSRSn

该位确认 GPIOA 引脚的配置（GPIO0 到 GPIO31）是否锁定。一旦设置此位，只能通过复位清除。此寄存器将写入 GPALOCK[LOCKy]寄存器中。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/WSO	提交 GPIOy 锁定（GPIOy lock is committed） 该位提交 GPALOCK[GPIOy]锁定。 0：不提交锁定 GPSLOCK[GPIOy]的配置 1：提交锁定 GPALOCK[GPIOy]的配置	0h

21.9.22 端口 B 采样周期寄存器（GPBCTRL）

偏移地址：0x80

复位类型：SYSRSn

GPIOB 引脚的限定采样周期（GPIO32 到 GPIO59）。每个位域 8 个 GPIO 引脚构成的组，为每个组选择 SYSCLK 周期中的限定采样周期，周期数为位域值的两倍。

位/域	名称	R/W	描述	复位值
7:0	QUALPRD0	R/W	GPIO32 到 GPIO39 的采样周期（GPIO32 to GPIO39 sampling cycle） 选择 GPIO32 到 GPIO39 的采样周期。 0000 0000：周期=$T_{SYSCLKOUT}$*1 0000 0001：周期=$T_{SYSCLKOUT}$*2 0000 0010：周期=$T_{SYSCLKOUT}$*4 …… 1111 1111：周期=$T_{SYSCLKOUT}$*510	0h
15:8	QUALPRD1	R/W	GPIO40 到 GPIO47 的采样周期（GPIO40 to GPIO47 sampling cycle） 具体描述参考 QUALPRD0。	0h
23:16	QUALPRD2	R/W	GPIO48 到 GPIO55 的采样周期（GPIO48 to GPIO55 sampling cycle） 具体描述参考 QUALPRD0。	0h
27:24	QUALPRD3	R/W	GPIO56 到 GPIO59 的采样周期（GPIO56 to GPIO59 sampling cycle） 具体描述参考 QUALPRD0。	0h
31:28	保留			0h

21.9.23 端口 B 输入类型寄存器 1（GPBQSEL1）

偏移地址：0x84

复位类型：SYSRSn

GPIOB 引脚的输入限定类型（GPIO32 到 GPIO47）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
1:0	GPIO32	R/W	GPIO32 的输入类型（GPIO32 Input type） 00：同步输入限定 01：3 采样窗口输入限定 10：6 采样窗口输入限定 11：异步输入限定	0h
3:2	GPIO33	R/W	GPIO33 的输入类型（GPIO33 Input type） 具体描述参考 GPIO32。	0h
5:4	GPIO34	R/W	GPIO34 的输入类型（GPIO34 Input type） 具体描述参考 GPIO32。	0h
7:6	GPIO35	R/W	GPIO35 的输入类型（GPIO35 Input type） 具体描述参考 GPIO32。	0h
9:8	保留			0h
11:10	GPIO37	R/W	GPIO37 的输入类型（GPIO37 Input type） 具体描述参考 GPIO32。	0h
13:12	保留			0h
15:14	GPIO39	R/W	GPIO39 的输入类型（GPIO39 Input type） 具体描述参考 GPIO32。	0h
17:16	GPIO40	R/W	GPIO40 的输入类型（GPIO40 Input type） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
19:18	GPIO41	R/W	GPIO41 的输入类型（GPIO41 Input type） 具体描述参考 GPIO32。	0h
21:20	GPIO42	R/W	GPIO42 的输入类型（GPIO42 Input type） 具体描述参考 GPIO32。	0h
23:22	GPIO43	R/W	GPIO43 的输入类型（GPIO43 Input type） 具体描述参考 GPIO32。	0h
25:24	GPIO44	R/W	GPIO44 的输入类型（GPIO44 Input type） 具体描述参考 GPIO32。	0h
27:26	GPIO45	R/W	GPIO45 的输入类型（GPIO45 Input type） 具体描述参考 GPIO32。	0h
29:28	GPIO46	R/W	GPIO46 的输入类型（GPIO46 Input type） 具体描述参考 GPIO32。	0h
31:30	GPIO47	R/W	GPIO47 的输入类型（GPIO47 Input type） 具体描述参考 GPIO32。	0h

21.9.24 端口 B 输入类型寄存器 2（GPBQSEL2）

偏移地址：0x88

复位类型：SYSRSn

GPIOB 引脚的输入限定类型（GPIO48 到 GPIO59）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
1:0	GPIO48	R/W	GPIO48 的输入类型（GPIO48 Input type） 具体描述参考 GPIO32。	0h
3:2	GPIO49	R/W	GPIO49 的输入类型（GPIO49 Input type） 具体描述参考 GPIO32。	0h
5:4	GPIO50	R/W	GPIO50 的输入类型（GPIO50 Input type） 具体描述参考 GPIO32。	0h
7:6	GPIO51	R/W	GPIO51 的输入类型（GPIO51 Input type） 具体描述参考 GPIO32。	0h
9:8	GPIO52	R/W	GPIO52 的输入类型（GPIO52 Input type） 具体描述参考 GPIO32。	0h
11:10	GPIO53	R/W	GPIO53 的输入类型（GPIO53 Input type） 具体描述参考 GPIO32。	0h
13:12	GPIO54	R/W	GPIO54 的输入类型（GPIO54 Input type） 具体描述参考 GPIO32。	0h
15:14	GPIO55	R/W	GPIO55 的输入类型（GPIO55 Input type） 具体描述参考 GPIO32。	0h
17:16	GPIO56	R/W	GPIO56 的输入类型（GPIO56 Input type） 具体描述参考 GPIO32。	0h
19:18	GPIO57	R/W	GPIO57 的输入类型（GPIO57 Input type） 具体描述参考 GPIO32。	0h
21:20	GPIO58	R/W	GPIO58 的输入类型（GPIO58 Input type） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
23:22	GPIO59	R/W	GPIO59 的输入类型（GPIO59 Input type） 具体描述参考 GPIO32。	0h
31:24	保留			0h

21.9.25 端口 B 复用低位寄存器 1（GPBMUX1）

偏移地址：0x8C

复位类型：SYSRSn

此寄存器配置 GPIOB 引脚多路复用功能的低位部分（GPIO32 到 GPIO47），当位域值为 0x00，0x04，0x08，0x0C 时，对应的引脚被配置为 GPIO 模式，位域值为其他值时，选择外设控制该引脚。有关每个引脚的配置信息，详情请参见数据手册。

在更改 GPBGMUX1 寄存器的位域前，必须使用此寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO32	R/W	GPIO32 的外设多路复用配置的低两位（GPIO32 peripheral mux configuration in the low 2 bits）	0h
3:2	GPIO33	R/W	GPIO33 的外设多路复用配置的低两位（GPIO33 peripheral mux configuration in the low 2 bits）	0h
5:4	GPIO34	R/W	GPIO34 的外设多路复用配置的低两位（GPIO34 peripheral mux configuration in the low 2 bits）	0h
7:6	GPIO35	R/W	GPIO35 的外设多路复用配置的低两位（GPIO35 peripheral mux configuration in the low 2 bits）	3h
9:8	保留			0h
11:10	GPIO37	R/W	GPIO37 的外设多路复用配置的低两位（GPIO37 peripheral mux configuration in the low 2 bits）	3h
13:12	保留			0h
15:14	GPIO39	R/W	GPIO39 的外设多路复用配置的低两位（GPIO39 peripheral mux configuration in the low 2 bits）	0h
17:16	GPIO40	R/W	GPIO40 的外设多路复用配置的低两位（GPIO40 peripheral mux configuration in the low 2 bits）	0h
19:18	GPIO41	R/W	GPIO41 的外设多路复用配置的低两位（GPIO41 peripheral mux configuration in the low 2 bits）	0h
21:20	GPIO42	R/W	GPIO42 的外设多路复用配置的低两位（GPIO42 peripheral mux configuration in the low 2 bits）	0h
23:22	GPIO43	R/W	GPIO43 的外设多路复用配置的低两位（GPIO43 peripheral mux configuration in the low 2 bits）	0h
25:24	GPIO44	R/W	GPIO44 的外设多路复用配置的低两位（GPIO44 peripheral mux configuration in the low 2 bits）	0h
27:26	GPIO45	R/W	GPIO45 的外设多路复用配置的低两位（GPIO45 peripheral mux configuration in the low 2 bits）	0h
29:28	GPIO46	R/W	GPIO46 的外设多路复用配置的低两位（GPIO46 peripheral mux configuration in the low 2 bits）	0h
31:30	GPIO47	R/W	GPIO47 的外设多路复用配置的低两位（GPIO47 peripheral mux configuration in the low 2 bits）	0h

21.9.26 端口 B 复用低位寄存器 2（GPBMUX2）

偏移地址：0x90

复位类型：SYSRSn

此寄存器配置 GPIOB 引脚多路复用功能的低位部分（GPIO48 到 GPIO59），当位域值为 0x00，0x04，0x08，0x0C 时，对应的引脚被配置为 GPIO 模式，位域值为其他值时，选择外设控制该引脚。有关每个引脚的配置信息，详情请参见数据手册。

在更改 GPBGMUX2 寄存器的对应位域前，必须使用此寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO48	R/W	GPIO48 的外设多路复用配置的低两位（GPIO48 peripheral mux configuration in the low 2 bits）	0h
3:2	GPIO49	R/W	GPIO49 的外设多路复用配置的低两位（GPIO49 peripheral mux configuration in the low 2 bits）	0h
5:4	GPIO40	R/W	GPIO50 的外设多路复用配置的低两位（GPIO50 peripheral mux configuration in the low 2 bits）	0h
7:6	GPIO51	R/W	GPIO51 的外设多路复用配置的低两位（GPIO51 peripheral mux configuration in the low 2 bits）	0h
9:8	GPIO52	R/W	GPIO52 的外设多路复用配置的低两位（GPIO52 peripheral mux configuration in the low 2 bits）	0h
11:10	GPIO53	R/W	GPIO53 的外设多路复用配置的低两位（GPIO53 peripheral mux configuration in the low 2 bits）	0h
13:12	GPIO54	R/W	GPIO54 的外设多路复用配置的低两位（GPIO54 peripheral mux configuration in the low 2 bits）	0h
15:14	GPIO55	R/W	GPIO55 的外设多路复用配置的低两位（GPIO55 peripheral mux configuration in the low 2 bits）	0h
17:16	GPIO56	R/W	GPIO56 的外设多路复用配置的低两位（GPIO56 peripheral mux configuration in the low 2 bits）	0h
19:18	GPIO57	R/W	GPIO57 的外设多路复用配置的低两位（GPIO57 peripheral mux configuration in the low 2 bits）	0h
21:20	GPIO58	R/W	GPIO58 的外设多路复用配置的低两位（GPIO58 peripheral mux configuration in the low 2 bits）	0h
23:22	GPIO59	R/W	GPIO59 的外设多路复用配置的低两位（GPIO59 peripheral mux configuration in the low 2 bits）	0h
31:24	保留			0h

21.9.27 端口 B 数据方向寄存器（GPBDIR）

偏移地址：0x94

复位类型：SYSRSn

此寄存器配置 GPIOB 引脚的数据方向。若引脚未配置成 GPIO 模式，则此寄存器无效。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	GPIO32 的数据方向（GPIO32 Data direction） 配置 GPIO 模式下引脚的数据方向。 0：输入 1：输出	0h

位/域	名称	R/W	描述	复位值
1	GPIO33	R/W	GPIO33 的数据方向（GPIO33 Data direction） 具体描述参考 GPIO32。	0h
2	GPIO34	R/W	GPIO34 的数据方向（GPIO34 Data direction） 具体描述参考 GPIO32。	0h
3	GPIO35	R/W	GPIO35 的数据方向（GPIO35 Data direction） 具体描述参考 GPIO32。	0h
4	保留			0h
5	GPIO37	R/W	GPIO37 的数据方向（GPIO37 Data direction） 具体描述参考 GPIO32。	0h
6	保留			0h
7	GPIO39	R/W	GPIO39 的数据方向（GPIO39 Data direction） 具体描述参考 GPIO32。	0h
8	GPIO40	R/W	GPIO40 的数据方向（GPIO40 Data direction） 具体描述参考 GPIO32。	0h
9	GPIO41	R/W	GPIO41 的数据方向（GPIO41 Data direction） 具体描述参考 GPIO32。	0h
10	GPIO42	R/W	GPIO42 的数据方向（GPIO42 Data direction） 具体描述参考 GPIO32。	0h
11	GPIO43	R/W	GPIO43 的数据方向（GPIO43 Data direction） 具体描述参考 GPIO32。	0h
12	GPIO44	R/W	GPIO44 的数据方向（GPIO44 Data direction） 具体描述参考 GPIO32。	0h
13	GPIO45	R/W	GPIO45 的数据方向（GPIO45 Data direction） 具体描述参考 GPIO32。	0h
14	GPIO46	R/W	GPIO46 的数据方向（GPIO46 Data direction） 具体描述参考 GPIO32。	0h
15	GPIO47	R/W	GPIO47 的数据方向（GPIO47 Data direction） 具体描述参考 GPIO32。	0h
16	GPIO48	R/W	GPIO48 的数据方向（GPIO48 Data direction） 具体描述参考 GPIO32。	0h
17	GPIO49	R/W	GPIO49 的数据方向（GPIO49 Data direction） 具体描述参考 GPIO32。	0h
18	GPIO50	R/W	GPIO50 的数据方向（GPIO50 Data direction） 具体描述参考 GPIO32。	0h
19	GPIO51	R/W	GPIO51 的数据方向（GPIO51 Data direction） 具体描述参考 GPIO32。	0h
20	GPIO52	R/W	GPIO52 的数据方向（GPIO52 Data direction） 具体描述参考 GPIO32。	0h
21	GPIO53	R/W	GPIO53 的数据方向（GPIO53 Data direction） 具体描述参考 GPIO32。	0h
22	GPIO54	R/W	GPIO54 的数据方向（GPIO54 Data direction） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
23	GPIO55	R/W	GPIO55 的数据方向（GPIO55 Data direction） 具体描述参考 GPIO32。	0h
24	GPIO56	R/W	GPIO56 的数据方向（GPIO56 Data direction） 具体描述参考 GPIO32。	0h
25	GPIO57	R/W	GPIO57 的数据方向（GPIO57 Data direction） 具体描述参考 GPIO32。	0h
26	GPIO58	R/W	GPIO58 的数据方向（GPIO58 Data direction） 具体描述参考 GPIO32。	0h
27	GPIO59	R/W	GPIO59 的数据方向（GPIO59 Data direction） 具体描述参考 GPIO32。	0h
31:28	保留			0h

21.9.28 端口 B 上拉禁用寄存器（GPBPUD）

偏移地址：0x98

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	GPIO32 的上拉禁用（GPIO32 Pull-up disable） 配置 GPIO32 引脚的内部上拉电阻状态。 0：使能内部上拉电阻 1：禁用内部上拉电阻	1h
1	GPIO33	R/W	GPIO33 的上拉禁用（GPIO33 Pull-up disable） 具体描述参考 GPIO32。	1h
2	GPIO34	R/W	GPIO34 的上拉禁用（GPIO34 Pull-up disable） 具体描述参考 GPIO32。	1h
3	GPIO35	R/W	GPIO35 的上拉禁用（GPIO35 Pull-up disable） 具体描述参考 GPIO32。	1h
4	保留			1h
5	GPIO37	R/W	GPIO37 的上拉禁用（GPIO37 Pull-up disable） 具体描述参考 GPIO32。	1h
6	保留			1h
7	GPIO39	R/W	GPIO39 的上拉禁用（GPIO39 Pull-up disable） 具体描述参考 GPIO32。	1h
8	GPIO40	R/W	GPIO40 的上拉禁用（GPIO40 Pull-up disable） 具体描述参考 GPIO32。	1h
9	GPIO41	R/W	GPIO41 的上拉禁用（GPIO41 Pull-up disable） 具体描述参考 GPIO32。	1h
10	GPIO42	R/W	GPIO42 的上拉禁用（GPIO42 Pull-up disable） 具体描述参考 GPIO32。	1h
11	GPIO43	R/W	GPIO43 的上拉禁用（GPIO43 Pull-up disable） 具体描述参考 GPIO32。	1h
12	GPIO44	R/W	GPIO44 的上拉禁用（GPIO44 Pull-up disable） 具体描述参考 GPIO32。	1h

位/域	名称	R/W	描述	复位值
13	GPIO45	R/W	GPIO45 的上拉禁用（GPIO45 Pull-up disable） 具体描述参考 GPIO32。	1h
14	GPIO46	R/W	GPIO46 的上拉禁用（GPIO46 Pull-up disable） 具体描述参考 GPIO32。	1h
15	GPIO47	R/W	GPIO47 的上拉禁用（GPIO47 Pull-up disable） 具体描述参考 GPIO32。	1h
16	GPIO48	R/W	GPIO48 的上拉禁用（GPIO48 Pull-up disable） 具体描述参考 GPIO32。	1h
17	GPIO49	R/W	GPIO49 的上拉禁用（GPIO49 Pull-up disable） 具体描述参考 GPIO32。	1h
18	GPIO50	R/W	GPIO50 的上拉禁用（GPIO50 Pull-up disable） 具体描述参考 GPIO32。	1h
19	GPIO51	R/W	GPIO51 的上拉禁用（GPIO51 Pull-up disable） 具体描述参考 GPIO32。	1h
20	GPIO52	R/W	GPIO52 的上拉禁用（GPIO52 Pull-up disable） 具体描述参考 GPIO32。	1h
21	GPIO53	R/W	GPIO53 的上拉禁用（GPIO53 Pull-up disable） 具体描述参考 GPIO32。	1h
22	GPIO54	R/W	GPIO54 的上拉禁用（GPIO54 Pull-up disable） 具体描述参考 GPIO32。	1h
23	GPIO55	R/W	GPIO55 的上拉禁用（GPIO55 Pull-up disable） 具体描述参考 GPIO32。	1h
24	GPIO56	R/W	GPIO56 的上拉禁用（GPIO56 Pull-up disable） 具体描述参考 GPIO32。	1h
25	GPIO57	R/W	GPIO57 的上拉禁用（GPIO57 Pull-up disable） 具体描述参考 GPIO32。	1h
26	GPIO58	R/W	GPIO58 的上拉禁用（GPIO58 Pull-up disable） 具体描述参考 GPIO32。	1h
27	GPIO59	R/W	GPIO59 的上拉禁用（GPIO59 Pull-up disable） 具体描述参考 GPIO32。	1h
31:28	保留			1h

21.9.29 端口 B 输入反相寄存器（GPBINV）

偏移地址：0xA0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INVER32	R/W	GPIO32 的输入反相（GPIO32 Input inversion） 配置 GPIO32 引脚的输入值是否通过反相器。 0：输入值不反相 1：输入值反相	0h
1	INVER33	R/W	GPIO33 的输入反相（GPIO33 Input inversion） 具体描述参考 INVER32。	0h
2	INVER34	R/W	GPIO34 的输入反相（GPIO34 Input inversion） 具体描述参考 INVER32。	0h

位/域	名称	R/W	描述	复位值
3	INVER35	R/W	GPIO35 的输入反相（GPIO35 Input inversion） 具体描述参考 INVER32。	0h
4	保留			0h
5	INVER37	R/W	GPIO37 的输入反相（GPIO37 Input inversion） 具体描述参考 INVER32。	0h
6	保留			0h
7	INVER39	R/W	GPIO39 的输入反相（GPIO39 Input inversion） 具体描述参考 INVER32。	0h
8	INVER40	R/W	GPIO40 的输入反相（GPIO40 Input inversion） 具体描述参考 INVER32。	0h
9	INVER41	R/W	GPIO41 的输入反相（GPIO41 Input inversion） 具体描述参考 INVER32。	0h
10	INVER42	R/W	GPIO42 的输入反相（GPIO42 Input inversion） 具体描述参考 INVER32。	0h
11	INVER43	R/W	GPIO43 的输入反相（GPIO43 Input inversion） 具体描述参考 INVER32。	0h
12	INVER44	R/W	GPIO44 的输入反相（GPIO44 Input inversion） 具体描述参考 INVER32。	0h
13	INVER45	R/W	GPIO45 的输入反相（GPIO45 Input inversion） 具体描述参考 INVER32。	0h
14	INVER46	R/W	GPIO46 的输入反相（GPIO46 Input inversion） 具体描述参考 INVER32。	0h
15	INVER47	R/W	GPIO47 的输入反相（GPIO47 Input inversion） 具体描述参考 INVER32。	0h
16	INVER48	R/W	GPIO48 的输入反相（GPIO48 Input inversion） 具体描述参考 INVER32。	0h
17	INVER49	R/W	GPIO49 的输入反相（GPIO49 Input inversion） 具体描述参考 INVER32。	0h
18	INVER50	R/W	GPIO50 的输入反相（GPIO50 Input inversion） 具体描述参考 INVER32。	0h
19	INVER51	R/W	GPIO51 的输入反相（GPIO51 Input inversion） 具体描述参考 INVER32。	0h
20	INVER52	R/W	GPIO52 的输入反相（GPIO52 Input inversion） 具体描述参考 INVER32。	0h
21	INVER53	R/W	GPIO53 的输入反相（GPIO53 Input inversion） 具体描述参考 INVER32。	0h
22	INVER54	R/W	GPIO54 的输入反相（GPIO54 Input inversion） 具体描述参考 INVER32。	0h
23	INVER55	R/W	GPIO55 的输入反相（GPIO55 Input inversion） 具体描述参考 INVER32。	0h
24	INVER56	R/W	GPIO56 的输入反相（GPIO56 Input inversion） 具体描述参考 INVER32。	0h

位/域	名称	R/W	描述	复位值
25	INVER57	R/W	GPIO57 的输入反相（GPIO57 Input inversion） 具体描述参考 INVER32。	0h
26	INVER58	R/W	GPIO58 的输入反相（GPIO58 Input inversion） 具体描述参考 INVER32。	0h
27	INVER59	R/W	GPIO59 的输入反相（GPIO59 Input inversion） 具体描述参考 INVER32。	0h
31:28	保留			0h

21.9.30 端口 B 开漏输出寄存器（GPBODR）

偏移地址：0xA4

复位类型：SYSRSn

选择通用输出引脚的输出模式：推挽输出或开漏输出。在推挽输出模式中，向输出数据锁存器写入 1 将驱动引脚为高电平。在开漏模式下，向输出数据锁存器写入 1 使输出缓冲器处于三态。在两种输出模式中，向输出数据锁存器写入 0 将驱动引脚为低电平。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	GPIO32 的开漏输出（GPIO32 Open-drain output mode） 0：推挽输出 1：开漏输出	0h
1	GPIO33	R/W	GPIO33 的开漏输出（GPIO33 Open-drain output mode） 具体描述参考 GPIO32。	0h
2	GPIO34	R/W	GPIO34 的开漏输出（GPIO34 Open-drain output mode） 具体描述参考 GPIO32。	0h
3	GPIO35	R/W	GPIO35 的开漏输出（GPIO35 Open-drain output mode） 具体描述参考 GPIO32。	0h
4	保留			0h
5	GPIO37	R/W	GPIO37 的开漏输出（GPIO37 Open-drain output mode） 具体描述参考 GPIO32。	0h
6	保留			0h
7	GPIO39	R/W	GPIO39 的开漏输出（GPIO39 Open-drain output mode） 具体描述参考 GPIO32。	0h
8	GPIO40	R/W	GPIO40 的开漏输出（GPIO40 Open-drain output mode） 具体描述参考 GPIO32。	0h
9	GPIO41	R/W	GPIO41 的开漏输出（GPIO41 Open-drain output mode） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
10	GPIO42	R/W	GPIO42 的开漏输出（GPIO42 Open-drain output mode） 具体描述参考 GPIO32。	0h
11	GPIO43	R/W	GPIO43 的开漏输出（GPIO43 Open-drain output mode） 具体描述参考 GPIO32。	0h
12	GPIO44	R/W	GPIO44 的开漏输出（GPIO44 Open-drain output mode） 具体描述参考 GPIO32。	0h
13	GPIO45	R/W	GPIO45 的开漏输出（GPIO45 Open-drain output mode） 具体描述参考 GPIO32。	0h
14	GPIO46	R/W	GPIO46 的开漏输出（GPIO46 Open-drain output mode） 具体描述参考 GPIO32。	0h
15	GPIO47	R/W	GPIO47 的开漏输出（GPIO47 Open-drain output mode） 具体描述参考 GPIO32。	0h
16	GPIO48	R/W	GPIO48 的开漏输出（GPIO48 Open-drain output mode） 具体描述参考 GPIO32。	0h
17	GPIO49	R/W	GPIO49 的开漏输出（GPIO49 Open-drain output mode） 具体描述参考 GPIO32。	0h
18	GPIO50	R/W	GPIO50 的开漏输出（GPIO50 Open-drain output mode） 具体描述参考 GPIO32。	0h
19	GPIO51	R/W	GPIO51 的开漏输出（GPIO51 Open-drain output mode） 具体描述参考 GPIO32。	0h
20	GPIO52	R/W	GPIO52 的开漏输出（GPIO52 Open-drain output mode） 具体描述参考 GPIO32。	0h
21	GPIO53	R/W	GPIO53 的开漏输出（GPIO53 Open-drain output mode） 具体描述参考 GPIO32。	0h
22	GPIO54	R/W	GPIO54 的开漏输出（GPIO54 Open-drain output mode） 具体描述参考 GPIO32。	0h
23	GPIO55	R/W	GPIO55 的开漏输出（GPIO55 Open-drain output mode） 具体描述参考 GPIO32。	0h
24	GPIO56	R/W	GPIO56 的开漏输出（GPIO56 Open-drain output mode） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
25	GPIO57	R/W	GPIO57 的开漏输出（GPIO57 Open-drain output mode） 具体描述参考 GPIO32。	0h
26	GPIO58	R/W	GPIO58 的开漏输出（GPIO58 Open-drain output mode） 具体描述参考 GPIO32。	0h
27	GPIO59	R/W	GPIO59 的开漏输出（GPIO59 Open-drain output mode） 具体描述参考 GPIO32。	0h
31:28	保留			0h

21.9.31 端口 B 驱动能力选择寄存器 1（GPBDSEL1）

偏移地址：0xAC

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	GPIO32	R/W	选择 GPIO32 的驱动能力（Driver capability select for GPIO32） 当 GPBFEN[GPIO32]=1 时，最大输出频率为 25MHz； 当 GPBFEN[GPIO32]=0 时，驱动能力如下： 00：最大输出频率为 30MHz，驱动能力 1mA 01：最大输出频率为 50MHz，驱动能力 4mA 10：最大输出频率为 40MHz，驱动能力 2mA 11：最大输出频率为 60MHz，驱动能力 6mA 注意：驱动能力不受 GPBFEN 寄存器影响。	0h
3:2	GPIO33	R/W	选择 GPIO33 的驱动能力（Driver capability select for GPIO33） 具体描述参考 GPIO32。	0h
5:4	GPIO34	R/W	选择 GPIO34 的驱动能力（Driver capability select for GPIO34） 具体描述参考 GPIO32。	0h
7:6	GPIO35	R/W	选择 GPIO35 的驱动能力（Driver capability select for GPIO35） 具体描述参考 GPIO32。	0h
9:8	保留			0h
11:10	GPIO37	R/W	选择 GPIO37 的驱动能力（Driver capability select for GPIO37） 具体描述参考 GPIO32。	0h
13:12	保留			0h
15:14	GPIO39	R/W	选择 GPIO39 的驱动能力（Driver capability select for GPIO39） 具体描述参考 GPIO32。	0h
17:16	GPIO40	R/W	选择 GPIO40 的驱动能力（Driver capability select for GPIO40） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
19:18	GPIO41	R/W	选择 GPIO41 的驱动能力（Driver capability select for GPIO41） 具体描述参考 GPIO32。	0h
21:20	GPIO42	R/W	选择 GPIO42 的驱动能力（Driver capability select for GPIO42） 具体描述参考 GPIO32。	0h
23:22	GPIO43	R/W	选择 GPIO43 的驱动能力（Driver capability select for GPIO43） 具体描述参考 GPIO32。	0h
25:24	GPIO44	R/W	选择 GPIO44 的驱动能力（Driver capability select for GPIO44） 具体描述参考 GPIO32。	0h
27:26	GPIO45	R/W	选择 GPIO45 的驱动能力（Driver capability select for GPIO45） 具体描述参考 GPIO32。	0h
29:28	GPIO46	R/W	选择 GPIO46 的驱动能力（Driver capability select for GPIO46） 具体描述参考 GPIO32。	0h
31:30	GPIO47	R/W	选择 GPIO47 的驱动能力（Driver capability select for GPIO47） 具体描述参考 GPIO32。	0h

21.9.32 端口 B 驱动能力选择寄存器 2（GPBDSEL2）

偏移地址：0xB0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	GPIO48	R/W	选择 GPIO48 的驱动能力（Driver capability select for GPIO48） 具体描述参考 GPIO32。	0h
3:2	GPIO49	R/W	选择 GPIO49 的驱动能力（Driver capability select for GPIO49） 具体描述参考 GPIO32。	0h
5:4	GPIO50	R/W	选择 GPIO50 的驱动能力（Driver capability select for GPIO50） 具体描述参考 GPIO32。	0h
7:6	GPIO51	R/W	选择 GPIO51 的驱动能力（Driver capability select for GPIO51） 具体描述参考 GPIO32。	0h
9:8	GPIO52	R/W	选择 GPIO52 的驱动能力（Driver capability select for GPIO52） 具体描述参考 GPIO32。	0h
11:10	GPIO53	R/W	选择 GPIO53 的驱动能力（Driver capability select for GPIO53） 具体描述参考 GPIO32。	0h
13:12	GPIO54	R/W	选择 GPIO54 的驱动能力（Driver capability select for GPIO54） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
15:14	GPIO55	R/W	选择 GPIO55 的驱动能力（Driver capability select for GPIO55） 具体描述参考 GPIO32。	0h
17:16	GPIO56	R/W	选择 GPIO56 的驱动能力（Driver capability select for GPIO56） 具体描述参考 GPIO32。	0h
19:18	GPIO57	R/W	选择 GPIO57 的驱动能力（Driver capability select for GPIO57） 具体描述参考 GPIO32。	0h
21:20	GPIO58	R/W	选择 GPIO58 的驱动能力（Driver capability select for GPIO58） 具体描述参考 GPIO32。	0h
23:22	GPIO59	R/W	选择 GPIO59 的驱动能力（Driver capability select for GPIO59） 具体描述参考 GPIO32。	0h
31:24	保留			0h

21.9.33 端口 B 最大输出频率使能寄存器（GPBFEN）

偏移地址：0xB4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 0：GPIO32 的最大输出频率详情请参见 GPBDSEL1[GPIO32] 1：GPIO32 的最大输出频率为 25MHz	0h
1	GPIO33	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
2	GPIO34	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
3	GPIO35	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
4	保留			0h
5	GPIO37	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
6	保留			0h
7	GPIO39	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
8	GPIO40	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
9	GPIO41	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
10	GPIO42	R/W	使能 GPIO32 的最大输出频率 （GPIO32 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
11	GPIO43	R/W	使能 GPIO43 的最大输出频率（GPIO43 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
12	GPIO44	R/W	使能 GPIO44 的最大输出频率（GPIO44 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
13	GPIO45	R/W	使能 GPIO45 的最大输出频率（GPIO45 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
14	GPIO46	R/W	使能 GPIO46 的最大输出频率（GPIO46 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
15	GPIO47	R/W	使能 GPIO47 的最大输出频率（GPIO47 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
16	GPIO48	R/W	使能 GPIO48 的最大输出频率（GPIO48 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
17	GPIO49	R/W	使能 GPIO49 的最大输出频率（GPIO49 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
18	GPIO50	R/W	使能 GPIO50 的最大输出频率（GPIO50 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
19	GPIO51	R/W	使能 GPIO51 的最大输出频率（GPIO51 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
20	GPIO52	R/W	使能 GPIO52 的最大输出频率（GPIO52 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
21	GPIO53	R/W	使能 GPIO53 的最大输出频率（GPIO53 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
22	GPIO54	R/W	使能 GPIO54 的最大输出频率（GPIO54 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
23	GPIO55	R/W	使能 GPIO55 的最大输出频率（GPIO55 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
24	GPIO56	R/W	使能 GPIO56 的最大输出频率（GPIO56 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
25	GPIO57	R/W	使能 GPIO57 的最大输出频率（GPIO57 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
26	GPIO58	R/W	使能 GPIO58 的最大输出频率（GPIO58 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
27	GPIO59	R/W	使能 GPIO59 的最大输出频率（GPIO59 Maximum Output Frequency Enable） 具体描述参考 GPIO32。	0h
31:28	保留			0h

21.9.34 端口 B 复用高位寄存器 1（GPBGMUX1）

偏移地址：0xC0

复位类型：SYSRSn

此寄存器配置 GPIOB 引脚多路复用功能的高位部分（GPIO32 到 GPIO47）。有关每个引脚的配置信息，详情请参见数据手册。

在更改此寄存器的位域前，必须使用 GPBMUX1 寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO32	R/W	GPIO32 的外设多路复用配置的高两位（GPIO32 peripheral mux configuration in the High 2 bits）	0h
3:2	GPIO33	R/W	GPIO33 的外设多路复用配置的高两位（GPIO33 peripheral mux configuration in the High 2 bits）	0h
5:4	GPIO34	R/W	GPIO34 的外设多路复用配置的高两位（GPIO34 peripheral mux configuration in the High 2 bits）	0h
7:6	GPIO35	R/W	GPIO35 的外设多路复用配置的高两位（GPIO35 peripheral mux configuration in the High 2 bits）	3h
9:8	保留			0h
11:10	GPIO37	R/W	GPIO37 的外设多路复用配置的高两位（GPIO37 peripheral mux configuration in the High 2 bits）	3h
13:12	保留			0h
15:14	GPIO39	R/W	GPIO39 的外设多路复用配置的高两位（GPIO39 peripheral mux configuration in the High 2 bits）	0h
17:16	GPIO40	R/W	GPIO40 的外设多路复用配置的高两位（GPIO40 peripheral mux configuration in the High 2 bits）	0h
19:18	GPIO41	R/W	GPIO41 的外设多路复用配置的高两位（GPIO41 peripheral mux configuration in the High 2 bits）	0h
21:20	GPIO42	R/W	GPIO42 的外设多路复用配置的高两位（GPIO42 peripheral mux configuration in the High 2 bits）	0h

位/域	名称	R/W	描述	复位值
23:22	GPIO43	R/W	GPIO43 的外设多路复用配置的高两位（GPIO43 peripheral mux configuration in the High 2 bits）	0h
25:24	GPIO44	R/W	GPIO44 的外设多路复用配置的高两位（GPIO44 peripheral mux configuration in the High 2 bits）	0h
27:26	GPIO45	R/W	GPIO45 的外设多路复用配置的高两位（GPIO45 peripheral mux configuration in the High 2 bits）	0h
29:28	GPIO46	R/W	GPIO46 的外设多路复用配置的高两位（GPIO46 peripheral mux configuration in the High 2 bits）	0h
31:30	GPIO47	R/W	GPIO47 的外设多路复用配置的高两位（GPIO47 peripheral mux configuration in the High 2 bits）	0h

21.9.35 端口 B 复用高位寄存器 2（GPBGMUX2）

偏移地址：0xC4

复位类型：SYSRSn

此寄存器配置 GPIOB 引脚多路复用功能的高位部分（GPIO48 到 GPIO59）。有关每个引脚的配置信息，详情请参见数据手册。

在更改此寄存器的位域前，必须使用 GPBMUX2 寄存器将引脚配置为 GPIO 模式。

位/域	名称	R/W	描述	复位值
1:0	GPIO48	R/W	GPIO48 的外设多路复用配置的高两位（GPIO48 peripheral mux configuration in the High 2 bits）	0h
3:2	GPIO49	R/W	GPIO49 的外设多路复用配置的高两位（GPIO49 peripheral mux configuration in the High 2 bits）	0h
5:4	GPIO40	R/W	GPIO50 的外设多路复用配置的高两位（GPIO50 peripheral mux configuration in the High 2 bits）	0h
7:6	GPIO51	R/W	GPIO51 的外设多路复用配置的高两位（GPIO51 peripheral mux configuration in the High 2 bits）	0h
9:8	GPIO52	R/W	GPIO52 的外设多路复用配置的高两位（GPIO52 peripheral mux configuration in the High 2 bits）	0h
11:10	GPIO53	R/W	GPIO53 的外设多路复用配置的高两位（GPIO53 peripheral mux configuration in the High 2 bits）	0h
13:12	GPIO54	R/W	GPIO54 的外设多路复用配置的高两位（GPIO54 peripheral mux configuration in the High 2 bits）	0h
15:14	GPIO55	R/W	GPIO55 的外设多路复用配置的高两位（GPIO55 peripheral mux configuration in the High 2 bits）	0h
17:16	GPIO56	R/W	GPIO56 的外设多路复用配置的高两位（GPIO56 peripheral mux configuration in the High 2 bits）	0h
19:18	GPIO57	R/W	GPIO57 的外设多路复用配置的高两位（GPIO57 peripheral mux configuration in the High 2 bits）	0h
21:20	GPIO58	R/W	GPIO58 的外设多路复用配置的高两位（GPIO58 peripheral mux configuration in the High 2 bits）	0h
23:22	GPIO59	R/W	GPIO59 的外设多路复用配置的高两位（GPIO59 peripheral mux configuration in the High 2 bits）	0h
31:24	保留			0h

21.9.36 端口 B 主核选择寄存器 1（GPBCSEL1）

偏移地址：0xD0

复位类型：SYSRSn

此寄存器选择 GPIOB 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPBDAT，GPBSET，GPBCLR，GPBTOGGLE 四个寄存器以控制引脚。

（读）GPBDAT 时始终转入 CPU0 和 CPU1 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO32	R/W	选择 GPIO32 的主核（GPIO32 Master core select） 00：选择 CPU0 为主核 01：选择 CPU1 为主核 其他：保留	0h
7:4	GPIO33	R/W	选择 GPIO33 的主核（GPIO33 Master core select） 具体描述参考 GPIO32。	0h
11:8	GPIO34	R/W	选择 GPIO34 的主核（GPIO34 Master core select） 具体描述参考 GPIO32。	0h
15:12	GPIO35	R/W	选择 GPIO35 的主核（GPIO35 Master core select） 具体描述参考 GPIO32。	0h
19:16	保留			0h
23:20	GPIO37	R/W	选择 GPIO37 的主核（GPIO37 Master core select） 具体描述参考 GPIO32。	0h
27:24	保留			0h
31:28	GPIO39	R/W	选择 GPIO39 的主核（GPIO39 Master core select） 具体描述参考 GPIO32。	0h

21.9.37 端口 B 主核选择寄存器 2（GPBCSEL2）

偏移地址：0xD4

复位类型：SYSRSn

此寄存器选择 GPIOB 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPBDAT，GPBSET，GPBCLR，GPBTOGGLE 四个寄存器以控制引脚。

（读）GPBDAT 时始终转入两个 CPU 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO40	R/W	选择 GPIO40 的主核（GPIO40 Master core select） 具体描述参考 GPIO32。	0h
7:4	GPIO41	R/W	选择 GPIO41 的主核（GPIO41 Master core select） 具体描述参考 GPIO32。	0h
11:8	GPIO42	R/W	选择 GPIO42 的主核（GPIO42 Master core select） 具体描述参考 GPIO32。	0h
15:12	GPIO43	R/W	选择 GPIO43 的主核（GPIO43 Master core select） 具体描述参考 GPIO32。	0h
19:16	GPIO44	R/W	选择 GPIO44 的主核（GPIO44 Master core select） 具体描述参考 GPIO32。	0h
23:20	GPIO45	R/W	选择 GPIO45 的主核（GPIO45 Master core select） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
27:24	GPIO46	R/W	选择 GPIO46 的主核（GPIO46 Master core select） 具体描述参考 GPIO32。	0h
31:28	GPIO47	R/W	选择 GPIO47 的主核（GPIO47 Master core select） 具体描述参考 GPIO32。	0h

21.9.38 端口 B 主核选择寄存器 3（GPBCSEL3）

偏移地址：0xD8

复位类型：SYSRSn

此寄存器选择 GPIOB 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPBDAT，GPBSET，GPBCLR，GPBTOGGLE 四个寄存器以控制引脚。

（读）GPBDAT 时始终转入两个 CPU 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO48	R/W	选择 GPIO48 的主核（GPIO48 Master core select） 具体描述参考 GPIO32。	0h
7:4	GPIO49	R/W	选择 GPIO49 的主核（GPIO49 Master core select） 具体描述参考 GPIO32。	0h
11:8	GPIO50	R/W	选择 GPIO50 的主核（GPIO50 Master core select） 具体描述参考 GPIO32。	0h
15:12	GPIO51	R/W	选择 GPIO51 的主核（GPIO51 Master core select） 具体描述参考 GPIO32。	0h
19:16	GPIO52	R/W	选择 GPIO52 的主核（GPIO52 Master core select） 具体描述参考 GPIO32。	0h
23:20	GPIO53	R/W	选择 GPIO53 的主核（GPIO53 Master core select） 具体描述参考 GPIO32。	0h
27:24	GPIO54	R/W	选择 GPIO54 的主核（GPIO54 Master core select） 具体描述参考 GPIO32。	0h
31:28	GPIO55	R/W	选择 GPIO55 的主核（GPIO55 Master core select） 具体描述参考 GPIO32。	0h

21.9.39 端口 B 主核选择寄存器 4（GPBCSEL4）

偏移地址：0xDC

复位类型：SYSRSn

此寄存器选择 GPIOB 引脚的主控制器。主控制器在 GPIO 模式下通过配置 GPBDAT，GPBSET，GPBCLR，GPBTOGGLE 四个寄存器以控制引脚。

（读）GPBDAT 时始终转入两个 CPU 中。

位/域	名称	R/W	描述	复位值
3:0	GPIO56	R/W	选择 GPIO56 的主核（GPIO56 Master core select） 具体描述参考 GPIO32。	0h
7:4	GPIO57	R/W	选择 GPIO57 的主核（GPIO57 Master core select） 具体描述参考 GPIO32。	0h
11:8	GPIO58	R/W	选择 GPIO58 的主核（GPIO58 Master core select） 具体描述参考 GPIO32。	0h
15:12	GPIO59	R/W	选择 GPIO59 的主核（GPIO59 Master core select） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
31:16	保留			0h

21.9.40 端口 B 锁定寄存器（GPBLOCK）

偏移地址：0xF8

复位类型：SYSRSn

锁定 GPIOB 引脚的配置（GPIO32 到 GPIO59）。此寄存器将写入 GPBMUX1/2，GPBGMUX1/2，GPBDIR，GPBINV，GPBODR 和 GPBCSEL1/2/3/4 寄存器中。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	配置 GPIO32 锁定（GPIO32 Configuration lock） 0：不锁定 GPIO32 的配置 1：锁定 GPIO32 的配置	0h
1	GPIO33	R/W	配置 GPIO33 锁定（GPIO33 Configuration lock） 具体描述参考 GPIO32。	0h
2	GPIO34	R/W	配置 GPIO34 锁定（GPIO34 Configuration lock） 具体描述参考 GPIO32。	0h
3	GPIO35	R/W	配置 GPIO35 锁定（GPIO35 Configuration lock） 具体描述参考 GPIO32。	0h
4	保留			0h
5	GPIO37	R/W	配置 GPIO37 锁定（GPIO37 Configuration lock） 具体描述参考 GPIO32。	0h
6	保留			0h
7	GPIO39	R/W	配置 GPIO39 锁定（GPIO39 Configuration lock） 具体描述参考 GPIO32。	0h
8	GPIO40	R/W	配置 GPIO40 锁定（GPIO40 Configuration lock） 具体描述参考 GPIO32。	0h
9	GPIO41	R/W	配置 GPIO41 锁定（GPIO41 Configuration lock） 具体描述参考 GPIO32。	0h
10	GPIO42	R/W	配置 GPIO42 锁定（GPIO42 Configuration lock） 具体描述参考 GPIO32。	0h
11	GPIO43	R/W	配置 GPIO43 锁定（GPIO43 Configuration lock） 具体描述参考 GPIO32。	0h
12	GPIO44	R/W	配置 GPIO44 锁定（GPIO44 Configuration lock） 具体描述参考 GPIO32。	0h
13	GPIO45	R/W	配置 GPIO45 锁定（GPIO45 Configuration lock） 具体描述参考 GPIO32。	0h
14	GPIO46	R/W	配置 GPIO46 锁定（GPIO46 Configuration lock） 具体描述参考 GPIO32。	0h
15	GPIO47	R/W	配置 GPIO47 锁定（GPIO47 Configuration lock） 具体描述参考 GPIO32。	0h
16	GPIO48	R/W	配置 GPIO48 锁定（GPIO48 Configuration lock） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
17	GPIO49	R/W	配置 GPIO49 锁定（GPIO49 Configuration lock） 具体描述参考 GPIO32。	0h
18	GPIO50	R/W	配置 GPIO50 锁定（GPIO50 Configuration lock） 具体描述参考 GPIO32。	0h
19	GPIO51	R/W	配置 GPIO51 锁定（GPIO51 Configuration lock） 具体描述参考 GPIO32。	0h
20	GPIO52	R/W	配置 GPIO52 锁定（GPIO52 Configuration lock） 具体描述参考 GPIO32。	0h
21	GPIO53	R/W	配置 GPIO53 锁定（GPIO53 Configuration lock） 具体描述参考 GPIO32。	0h
22	GPIO54	R/W	配置 GPIO54 锁定（GPIO54 Configuration lock） 具体描述参考 GPIO32。	0h
23	GPIO55	R/W	配置 GPIO55 锁定（GPIO55 Configuration lock） 具体描述参考 GPIO32。	0h
24	GPIO56	R/W	配置 GPIO56 锁定（GPIO56 Configuration lock） 具体描述参考 GPIO32。	0h
25	GPIO57	R/W	配置 GPIO57 锁定（GPIO57 Configuration lock） 具体描述参考 GPIO32。	0h
26	GPIO58	R/W	配置 GPIO58 锁定（GPIO58 Configuration lock） 具体描述参考 GPIO32。	0h
27	GPIO59	R/W	配置 GPIO59 锁定（GPIO59 Configuration lock） 具体描述参考 GPIO32。	0h
31:28	保留			0h

21.9.41 端口 B 提交锁定寄存器（GPBCR）

偏移地址：0xFC

复位类型：SYSRSn

该位确认 GPIOB 引脚的配置（GPIO32 到 GPIO59）是否锁定。一旦设置此位，只能通过复位清除。此寄存器将写入 GPBLOCK[GPIOy]寄存器中。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/WSO	提交 GPIO32 锁定（GPIO32 lock is committed） 该位提交 GPBLOCK[GPIO32]锁定。 0：不提交锁定 GPBLOCK[GPIO32]的配置 1：提交锁定 GPBLOCK[GPIO32]的配置	0h
1	GPIO33	R/WSO	锁定 GPIO33 锁定（GPIO33 lock is committed） 具体描述参考 GPIO32。	0h
2	GPIO34	R/WSO	锁定 GPIO34 锁定（GPIO34 lock is committed） 具体描述参考 GPIO32。	0h
3	GPIO35	R/WSO	锁定 GPIO35 锁定（GPIO35 lock is committed） 具体描述参考 GPIO32。	0h
4	保留			0h
5	GPIO37	R/WSO	锁定 GPIO37 锁定（GPIO37 lock is committed） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
6	保留			0h
7	GPIO39	R/WSO	锁定 GPIO39 锁定（GPIO39 lock is committed） 具体描述参考 GPIO32。	0h
8	GPIO40	R/WSO	锁定 GPIO40 锁定（GPIO40 lock is committed） 具体描述参考 GPIO32。	0h
9	GPIO41	R/WSO	锁定 GPIO41 锁定（GPIO41 lock is committed） 具体描述参考 GPIO32。	0h
10	GPIO42	R/WSO	锁定 GPIO42 锁定（GPIO42 lock is committed） 具体描述参考 GPIO32。	0h
11	GPIO43	R/WSO	锁定 GPIO43 锁定（GPIO43 lock is committed） 具体描述参考 GPIO32。	0h
12	GPIO44	R/WSO	锁定 GPIO44 锁定（GPIO44 lock is committed） 具体描述参考 GPIO32。	0h
13	GPIO45	R/WSO	锁定 GPIO45 锁定（GPIO45 lock is committed） 具体描述参考 GPIO32。	0h
14	GPIO46	R/WSO	锁定 GPIO46 锁定（GPIO46 lock is committed） 具体描述参考 GPIO32。	0h
15	GPIO47	R/WSO	锁定 GPIO47 锁定（GPIO47 lock is committed） 具体描述参考 GPIO32。	0h
16	GPIO48	R/WSO	锁定 GPIO48 锁定（GPIO48 lock is committed） 具体描述参考 GPIO32。	0h
17	GPIO49	R/WSO	锁定 GPIO49 锁定（GPIO49 lock is committed） 具体描述参考 GPIO32。	0h
18	GPIO50	R/WSO	锁定 GPIO50 锁定（GPIO50 lock is committed） 具体描述参考 GPIO32。	0h
19	GPIO51	R/WSO	锁定 GPIO51 锁定（GPIO51 lock is committed） 具体描述参考 GPIO32。	0h
20	GPIO52	R/WSO	锁定 GPIO52 锁定（GPIO52 lock is committed） 具体描述参考 GPIO32。	0h
21	GPIO53	R/WSO	锁定 GPIO53 锁定（GPIO53 lock is committed） 具体描述参考 GPIO32。	0h
22	GPIO54	R/WSO	锁定 GPIO54 锁定（GPIO54 lock is committed） 具体描述参考 GPIO32。	0h
23	GPIO55	R/WSO	锁定 GPIO55 锁定（GPIO55 lock is committed） 具体描述参考 GPIO32。	0h
24	GPIO56	R/WSO	锁定 GPIO56 锁定（GPIO56 lock is committed） 具体描述参考 GPIO32。	0h
25	GPIO57	R/WSO	锁定 GPIO57 锁定（GPIO57 lock is committed） 具体描述参考 GPIO32。	0h
26	GPIO58	R/WSO	锁定 GPIO58 锁定（GPIO58 lock is committed） 具体描述参考 GPIO32。	0h
27	GPIO59	R/WSO	锁定 GPIO59 锁定（GPIO59 lock is committed） 具体描述参考 GPIO32。	0h

位/域	名称	R/W	描述	复位值
31:28	保留			0h

21.9.42 端口 H 采样周期寄存器（GPHCTRL）

偏移地址：0x380

复位类型：SYSRSn

GPIOH 引脚的限定采样周期（GPIO224 到 GPIO254）。每个位域 8 个 GPIO 引脚构成的组，为每个组选择 SYSCLK 周期中的限定采样周期，周期数为位域值的两倍。

位/域	名称	R/W	描述	复位值
7:0	QUALPRD0	R/W	GPIO224 到 GPIO231 的采样周期（GPIO224 to GPIO231 sampling cycle） 选择 GPIO224 到 GPIO231 的采样周期。 0000 0000：周期=$T_{SYSCLKOUT}$*1 0000 0001：周期=$T_{SYSCLKOUT}$*2 0000 0010：周期=$T_{SYSCLKOUT}$*4 …… 1111 1111：周期=$T_{SYSCLKOUT}$*510	0h
15:8	QUALPRD1	R/W	GPIO232 到 GPIO239 的采样周期（GPIO232 to GPIO239 sampling cycle） 具体描述参考 QUALPRD0。	0h
23:16	QUALPRD2	R/W	GPIO240 到 GPIO247 的采样周期（GPIO240 to GPIO247 sampling cycle） 具体描述参考 QUALPRD0。	0h
30:24	QUALPRD3	R/W	GPIO248 到 GPIO254 的采样周期（GPIO248 to GPIO254 sampling cycle） 具体描述参考 QUALPRD0。	0h
31	保留			0h

21.9.43 端口 H 输入类型寄存器 1（GPHQSEL1）

偏移地址：0x384

复位类型：SYSRSn

GPIOH 引脚的输入限定类型（GPIO224 到 GPIO239）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
1:0	GPIO224	R/W	GPIO224 的输入类型（GPIO224 Input type） 00：同步输入限定 01：3 采样窗口输入限定 10：6 采样窗口输入限定 11：异步输入限定	0h
3:2	GPIO225	R/W	GPIO225 的输入类型（GPIO225 Input type） 具体描述参考 GPIO224。	0h
5:4	GPIO226	R/W	GPIO226 的输入类型（GPIO226 Input type） 具体描述参考 GPIO224。	0h
7:6	GPIO227	R/W	GPIO227 的输入类型（GPIO227 Input type） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
9:8	GPIO228	R/W	GPIO228 的输入类型（GPIO228 Input type） 具体描述参考 GPIO224。	0h
11:10	GPIO229	R/W	GPIO229 的输入类型（GPIO229 Input type） 具体描述参考 GPIO224。	0h
13:12	GPIO230	R/W	GPIO230 的输入类型（GPIO230 Input type） 具体描述参考 GPIO224。	0h
15:14	GPIO231	R/W	GPIO231 的输入类型（GPIO231 Input type） 具体描述参考 GPIO224。	0h
17:16	GPIO232	R/W	GPIO232 的输入类型（GPIO232 Input type） 具体描述参考 GPIO224。	0h
19:18	GPIO233	R/W	GPIO233 的输入类型（GPIO233 Input type） 具体描述参考 GPIO224。	0h
21:20	GPIO234	R/W	GPIO234 的输入类型（GPIO234 Input type） 具体描述参考 GPIO224。	0h
23:22	GPIO235	R/W	GPIO235 的输入类型（GPIO235 Input type） 具体描述参考 GPIO224。	0h
25:24	GPIO236	R/W	GPIO236 的输入类型（GPIO236 Input type） 具体描述参考 GPIO224。	0h
27:26	GPIO237	R/W	GPIO237 的输入类型（GPIO237 Input type） 具体描述参考 GPIO224。	0h
29:28	GPIO238	R/W	GPIO238 的输入类型（GPIO238 Input type） 具体描述参考 GPIO224。	0h
31:30	GPIO239	R/W	GPIO239 的输入类型（GPIO239 Input type） 具体描述参考 GPIO224。	0h

21.9.44 端口 H 输入类型寄存器 2（GPHQSEL2）

偏移地址：0x388

复位类型：SYSRSn

GPIOH 引脚的输入限定类型（GPIO240 到 GPIO254）。为每个引脚选择对应的限定输入类型。

位/域	名称	R/W	描述	复位值
1:0	GPIO240	R/W	GPIO240 的输入类型（GPIO240 Input type） 具体描述参考 GPIO224。	0h
3:2	GPIO241	R/W	GPIO241 的输入类型（GPIO241 Input type） 具体描述参考 GPIO224。	0h
5:4	GPIO242	R/W	GPIO242 的输入类型（GPIO242 Input type） 具体描述参考 GPIO224。	0h
7:6	GPIO243	R/W	GPIO243 的输入类型（GPIO243 Input type） 具体描述参考 GPIO224。	0h
9:8	GPIO244	R/W	GPIO244 的输入类型（GPIO244 Input type） 具体描述参考 GPIO224。	0h
11:10	GPIO245	R/W	GPIO245 的输入类型（GPIO245 Input type） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
13:12	GPIO246	R/W	GPIO246 的输入类型（GPIO246 Input type） 具体描述参考 GPIO224。	0h
15:14	GPIO247	R/W	GPIO247 的输入类型（GPIO247 Input type） 具体描述参考 GPIO224。	0h
17:16	GPIO248	R/W	GPIO248 的输入类型（GPIO248 Input type） 具体描述参考 GPIO224。	0h
19:18	GPIO249	R/W	GPIO249 的输入类型（GPIO249 Input type） 具体描述参考 GPIO224。	0h
21:20	GPIO250	R/W	GPIO250 的输入类型（GPIO250 Input type） 具体描述参考 GPIO224。	0h
23:22	GPIO251	R/W	GPIO251 的输入类型（GPIO251 Input type） 具体描述参考 GPIO224。	0h
25:24	GPIO252	R/W	GPIO252 的输入类型（GPIO252 Input type） 具体描述参考 GPIO224。	0h
27:26	GPIO253	R/W	GPIO253 的输入类型（GPIO254 Input type） 具体描述参考 GPIO224。	0h
29:28	GPIO254	R/W	GPIO254 的输入类型（GPIO254 Input type） 具体描述参考 GPIO224。	0h
31:30	保留			0h

21.9.45 端口 H 上拉禁用寄存器（GPHPUD）

偏移地址：0x398

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GPIO224	R/W	GPIO224 的上拉禁用（GPIO224 Pull-up disable） 配置 GPIO224 引脚的内部上拉电阻状态。 0：使能内部上拉电阻 1：禁用内部上拉电阻	1h
1	GPIO225	R/W	GPIO225 的上拉禁用（GPIO225 Pull-up disable） 具体描述参考 GPIO224。	1h
2	GPIO226	R/W	GPIO226 的上拉禁用（GPIO226 Pull-up disable） 具体描述参考 GPIO224。	1h
3	GPIO227	R/W	GPIO227 的上拉禁用（GPIO227 Pull-up disable） 具体描述参考 GPIO224。	1h
4	GPIO228	R/W	GPIO228 的上拉禁用（GPIO228 Pull-up disable） 具体描述参考 GPIO224。	1h
5	GPIO229	R/W	GPIO229 的上拉禁用（GPIO229 Pull-up disable） 具体描述参考 GPIO224。	1h
6	GPIO230	R/W	GPIO230 的上拉禁用（GPIO230 Pull-up disable） 具体描述参考 GPIO224。	1h
7	GPIO231	R/W	GPIO231 的上拉禁用（GPIO231 Pull-up disable） 具体描述参考 GPIO224。	1h
8	GPIO232	R/W	GPIO232 的上拉禁用（GPIO232 Pull-up disable） 具体描述参考 GPIO224。	1h

位/域	名称	R/W	描述	复位值
9	GPIO233	R/W	GPIO233 的上拉禁用（GPIO233 Pull-up disable） 具体描述参考 GPIO224。	1h
10	GPIO234	R/W	GPIO234 的上拉禁用（GPIO234 Pull-up disable） 具体描述参考 PUD224。	1h
11	GPIO235	R/W	GPIO235 的上拉禁用（GPIO235 Pull-up disable） 具体描述参考 GPIO224。	1h
12	GPIO236	R/W	GPIO236 的上拉禁用（GPIO236 Pull-up disable） 具体描述参考 GPIO224。	1h
13	GPIO237	R/W	GPIO237 的上拉禁用（GPIO237 Pull-up disable） 具体描述参考 GPIO224。	1h
14	GPIO238	R/W	GPIO238 的上拉禁用（GPIO238 Pull-up disable） 具体描述参考 GPIO224。	1h
15	GPIO239	R/W	GPIO239 的上拉禁用（GPIO239 Pull-up disable） 具体描述参考 GPIO224。	1h
16	GPIO240	R/W	GPIO240 的上拉禁用（GPIO240 Pull-up disable） 具体描述参考 GPIO224。	1h
17	GPIO241	R/W	GPIO241 的上拉禁用（GPIO241 Pull-up disable） 具体描述参考 GPIO224。	1h
18	GPIO242	R/W	GPIO242 的上拉禁用（GPIO242 Pull-up disable） 具体描述参考 GPIO224。	1h
19	GPIO243	R/W	GPIO243 的上拉禁用（GPIO243 Pull-up disable） 具体描述参考 GPIO224。	1h
20	GPIO244	R/W	GPIO244 的上拉禁用（GPIO244 Pull-up disable） 具体描述参考 GPIO224。	1h
21	GPIO245	R/W	GPIO245 的上拉禁用（GPIO245 Pull-up disable） 具体描述参考 GPIO224。	1h
22	GPIO246	R/W	GPIO246 的上拉禁用（GPIO246 Pull-up disable） 具体描述参考 GPIO224。	1h
23	GPIO247	R/W	GPIO247 的上拉禁用（GPIO247 Pull-up disable） 具体描述参考 GPIO224。	1h
24	GPIO248	R/W	GPIO248 的上拉禁用（GPIO248 Pull-up disable） 具体描述参考 GPIO224。	1h
25	GPIO249	R/W	GPIO249 的上拉禁用（GPIO249 Pull-up disable） 具体描述参考 GPIO224。	1h
26	GPIO250	R/W	GPIO250 的上拉禁用（GPIO250 Pull-up disable） 具体描述参考 GPIO224。	1h
27	GPIO251	R/W	GPIO251 的上拉禁用（GPIO251 Pull-up disable） 具体描述参考 GPIO224。	1h
28	GPIO252	R/W	GPIO252 的上拉禁用（GPIO252 Pull-up disable） 具体描述参考 GPIO224。	1h
29	GPIO253	R/W	GPIO253 的上拉禁用（GPIO253 Pull-up disable） 具体描述参考 GPIO224。	1h
30	GPIO254	R/W	GPIO254 的上拉禁用（GPIO254 Pull-up disable） 具体描述参考 GPIO224。	1h

位/域	名称	R/W	描述	复位值
31	保留			1h

21.9.46 端口 H 输入反相寄存器（GPHINV）

偏移地址：0x3A0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GPIO224	R/W	GPIO224 的输入反相（GPIO224 Input inversion） 配置 GPIO224 引脚的输入值是否通过反相器。 0：输入值不反相 1：输入值反相	0h
1	GPIO225	R/W	GPIO225 的输入反相（GPIO225 Input inversion） 具体描述参考 GPIO224。	0h
2	GPIO226	R/W	GPIO226 的输入反相（GPIO226 Input inversion） 具体描述参考 GPIO224。	0h
3	GPIO227	R/W	GPIO227 的输入反相（GPIO227 Input inversion） 具体描述参考 GPIO224。	0h
4	GPIO228	R/W	GPIO228 的输入反相（GPIO228 Input inversion） 具体描述参考 GPIO224。	0h
5	GPIO229	R/W	GPIO229 的输入反相（GPIO229 Input inversion） 具体描述参考 GPIO224。	0h
6	GPIO230	R/W	GPIO230 的输入反相（GPIO230 Input inversion） 具体描述参考 GPIO224。	0h
7	GPIO231	R/W	GPIO231 的输入反相（GPIO231 Input inversion） 具体描述参考 GPIO224。	0h
8	GPIO232	R/W	GPIO232 的输入反相（GPIO232 Input inversion） 具体描述参考 GPIO224。	0h
9	GPIO233	R/W	GPIO233 的输入反相（GPIO233 Input inversion） 具体描述参考 GPIO224。	0h
10	GPIO234	R/W	GPIO234 的输入反相（GPIO234 Input inversion） 具体描述参考 GPIO224。	0h
11	GPIO235	R/W	GPIO235 的输入反相（GPIO235 Input inversion） 具体描述参考 GPIO224。	0h
12	GPIO236	R/W	GPIO236 的输入反相（GPIO236 Input inversion） 具体描述参考 GPIO224。	0h
13	GPIO237	R/W	GPIO237 的输入反相（GPIO237 Input inversion） 具体描述参考 GPIO224。	0h
14	GPIO238	R/W	GPIO238 的输入反相（GPIO238 Input inversion） 具体描述参考 GPIO224。	0h
15	GPIO239	R/W	GPIO239 的输入反相（GPIO239 Input inversion） 具体描述参考 GPIO224。	0h
16	GPIO240	R/W	GPIO240 的输入反相（GPIO240 Input inversion） 具体描述参考 GPIO224。	0h
17	GPIO241	R/W	GPIO241 的输入反相（GPIO241 Input inversion） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
18	GPIO242	R/W	GPIO242 的输入反相（GPIO242 Input inversion） 具体描述参考 GPIO224。	0h
19	GPIO243	R/W	GPIO243 的输入反相（GPIO243 Input inversion） 具体描述参考 GPIO224。	0h
20	GPIO244	R/W	GPIO244 的输入反相（GPIO244 Input inversion） 具体描述参考 GPIO224。	0h
21	GPIO245	R/W	GPIO245 的输入反相（GPIO245 Input inversion） 具体描述参考 GPIO224。	0h
22	GPIO246	R/W	GPIO246 的输入反相（GPIO246 Input inversion） 具体描述参考 GPIO224。	0h
23	GPIO247	R/W	GPIO247 的输入反相（GPIO247 Input inversion） 具体描述参考 GPIO224。	0h
24	GPIO248	R/W	GPIO248 的输入反相（GPIO248 Input inversion） 具体描述参考 GPIO224。	0h
25	GPIO249	R/W	GPIO249 的输入反相（GPIO249 Input inversion） 具体描述参考 GPIO224。	0h
26	GPIO250	R/W	GPIO250 的输入反相（GPIO250 Input inversion） 具体描述参考 GPIO224。	0h
27	GPIO251	R/W	GPIO251 的输入反相（GPIO251 Input inversion） 具体描述参考 GPIO224。	0h
28	GPIO252	R/W	GPIO252 的输入反相（GPIO252 Input inversion） 具体描述参考 GPIO224。	0h
29	GPIO253	R/W	GPIO253 的输入反相（GPIO253 Input inversion） 具体描述参考 GPIO224。	0h
30	GPIO254	R/W	GPIO254 的输入反相（GPIO254 Input inversion） 具体描述参考 GPIO224。	0h
31	保留			0h

21.9.47 端口 H 模拟模式选择寄存器（GPHAMSEL）

偏移地址：0x3A8

复位类型：SYSRSn

选择 GPIOH 引脚的模拟模式（GPIO224 到 GPIO254）。

位/域	名称	R/W	描述	复位值
0	GPIO224	R/W	选择 GPIO224 的模拟模式（Analog mode select for GPIO224） 选择 GPIO224 为模拟模式或数字模式。 0：数字模式 1：模拟模式	1h
1	GPIO225	R/W	选择 GPIO225 的模拟模式（Analog mode select for GPIO225） 具体描述参考 GPIO224。	1h
2	GPIO226	R/W	选择 GPIO226 的模拟模式（Analog mode select for GPIO226） 具体描述参考 GPIO224。	1h

位/域	名称	R/W	描述	复位值
3	GPIO227	R/W	选择 GPIO227 的模拟模式（Analog mode select for GPIO227） 具体描述参考 GPIO224。	1h
4	GPIO228	R/W	选择 GPIO228 的模拟模式（Analog mode select for GPIO228） 具体描述参考 GPIO224。	1h
5	GPIO229	R/W	选择 GPIO229 的模拟模式（Analog mode select for GPIO229） 具体描述参考 GPIO224。	1h
6	GPIO230	R/W	选择 GPIO230 的模拟模式（Analog mode select for GPIO230） 具体描述参考 GPIO224。	1h
7	GPIO231	R/W	选择 GPIO231 的模拟模式（Analog mode select for GPIO231） 具体描述参考 GPIO224。	1h
8	GPIO232	R/W	选择 GPIO232 的模拟模式（Analog mode select for GPIO232） 具体描述参考 GPIO224。	1h
9	GPIO233	R/W	选择 GPIO233 的模拟模式（Analog mode select for GPIO233） 具体描述参考 GPIO224。	1h
10	GPIO234	R/W	选择 GPIO234 的模拟模式（Analog mode select for GPIO234） 具体描述参考 GPIO224。	1h
11	GPIO235	R/W	选择 GPIO235 的模拟模式（Analog mode select for GPIO235） 具体描述参考 GPIO224。	1h
12	GPIO236	R/W	选择 GPIO236 的模拟模式（Analog mode select for GPIO236） 具体描述参考 GPIO224。	1h
13	GPIO237	R/W	选择 GPIO237 的模拟模式（Analog mode select for GPIO237） 具体描述参考 GPIO224。	1h
14	GPIO238	R/W	选择 GPIO238 的模拟模式（Analog mode select for GPIO238） 具体描述参考 GPIO224。	1h
15	GPIO239	R/W	选择 GPIO239 的模拟模式（Analog mode select for GPIO239） 具体描述参考 GPIO224。	1h
16	GPIO240	R/W	选择 GPIO240 的模拟模式（Analog mode select for GPIO240） 具体描述参考 GPIO224。	1h
17	GPIO241	R/W	选择 GPIO241 的模拟模式（Analog mode select for GPIO241） 具体描述参考 GPIO224。	1h

位/域	名称	R/W	描述	复位值
18	GPIO242	R/W	选择 GPIO242 的模拟模式（Analog mode select for GPIO242） 具体描述参考 GPIO224。	1h
19	GPIO243	R/W	选择 GPIO243 的模拟模式（Analog mode select for GPIO243） 具体描述参考 GPIO224。	1h
20	GPIO244	R/W	选择 GPIO244 的模拟模式（Analog mode select for GPIO244） 具体描述参考 GPIO224。	1h
21	GPIO245	R/W	选择 GPIO245 的模拟模式（Analog mode select for GPIO245） 具体描述参考 GPIO224。	1h
22	GPIO246	R/W	选择 GPIO246 的模拟模式（Analog mode select for GPIO246） 具体描述参考 GPIO224。	1h
23	GPIO247	R/W	选择 GPIO247 的模拟模式（Analog mode select for GPIO247） 具体描述参考 GPIO224。	1h
24	GPIO248	R/W	选择 GPIO248 的模拟模式（Analog mode select for GPIO248） 具体描述参考 GPIO224。	1h
25	GPIO249	R/W	选择 GPIO249 的模拟模式（Analog mode select for GPIO249） 具体描述参考 GPIO224。	1h
26	GPIO250	R/W	选择 GPIO250 的模拟模式（Analog mode select for GPIO250） 具体描述参考 GPIO224。	1h
27	GPIO251	R/W	选择 GPIO251 的模拟模式（Analog mode select for GPIO251） 具体描述参考 GPIO224。	1h
28	GPIO252	R/W	选择 GPIO252 的模拟模式（Analog mode select for GPIO252） 具体描述参考 GPIO224。	1h
29	GPIO253	R/W	选择 GPIO253 的模拟模式（Analog mode select for GPIO253） 具体描述参考 GPIO224。	1h
30	GPIO254	R/W	选择 GPIO254 的模拟模式（Analog mode select for GPIO254） 具体描述参考 GPIO224。	1h
31	保留			1h

21.9.48 端口 H 锁定寄存器（GPHLOCK）

偏移地址：0x3F8

复位类型：SYSRSn

锁定 GPIOH 引脚的配置（GPIO224 到 GPIO254）。此寄存器将写入 GPHINV 寄存器中。

位/域	名称	R/W	描述	复位值
0	GPIO224	R/W	配置 GPIO224 锁定（GPIO224 Configuration lock） 0：不锁定 GPIO224 的配置 1：锁定 GPIO224 的配置	0h
1	GPIO225	R/W	配置 GPIO225 锁定（GPIO225 Configuration lock） 具体描述参考 GPIO224。	0h
2	GPIO226	R/W	配置 GPIO226 锁定（GPIO226 Configuration lock） 具体描述参考 GPIO224。	0h
3	GPIO227	R/W	配置 GPIO227 锁定（GPIO227 Configuration lock） 具体描述参考 GPIO224。	0h
4	GPIO228	R/W	配置 GPIO228 锁定（GPIO228 Configuration lock） 具体描述参考 GPIO224。	0h
5	GPIO229	R/W	配置 GPIO229 锁定（GPIO229 Configuration lock） 具体描述参考 GPIO224。	0h
6	GPIO230	R/W	配置 GPIO230 锁定（GPIO230 Configuration lock） 具体描述参考 GPIO224。	0h
7	GPIO231	R/W	配置 GPIO231 锁定（GPIO231 Configuration lock） 具体描述参考 GPIO224。	0h
8	GPIO232	R/W	配置 GPIO232 锁定（GPIO232 Configuration lock） 具体描述参考 GPIO224。	0h
9	GPIO233	R/W	配置 GPIO233 锁定（GPIO233 Configuration lock） 具体描述参考 GPIO224。	0h
10	GPIO234	R/W	配置 GPIO234 锁定（GPIO234 Configuration lock） 具体描述参考 GPIO224。	0h
11	GPIO235	R/W	配置 GPIO235 锁定（GPIO235 Configuration lock） 具体描述参考 GPIO224。	0h
12	GPIO236	R/W	配置 GPIO236 锁定（GPIO236 Configuration lock） 具体描述参考 GPIO224。	0h
13	GPIO237	R/W	配置 GPIO237 锁定（GPIO237 Configuration lock） 具体描述参考 GPIO224。	0h
14	GPIO238	R/W	配置 GPIO238 锁定（GPIO238 Configuration lock） 具体描述参考 GPIO224。	0h
15	GPIO239	R/W	配置 GPIO239 锁定（GPIO239 Configuration lock） 具体描述参考 GPIO224。	0h
16	GPIO240	R/W	配置 GPIO240 锁定（GPIO240 Configuration lock） 具体描述参考 GPIO224。	0h
17	GPIO241	R/W	配置 GPIO241 锁定（GPIO241 Configuration lock） 具体描述参考 GPIO224。	0h
18	GPIO242	R/W	配置 GPIO242 锁定（GPIO242 Configuration lock） 具体描述参考 GPIO224。	0h
19	GPIO243	R/W	配置 GPIO243 锁定（GPIO243 Configuration lock） 具体描述参考 GPIO224。	0h
20	GPIO244	R/W	配置 GPIO244 锁定（GPIO244 Configuration lock） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
21	GPIO245	R/W	配置 GPIO245 锁定（GPIO245 Configuration lock） 具体描述参考 GPIO224。	0h
22	GPIO246	R/W	配置 GPIO246 锁定（GPIO246 Configuration lock） 具体描述参考 GPIO224。	0h
23	GPIO247	R/W	配置 GPIO247 锁定（GPIO247 Configuration lock） 具体描述参考 GPIO224。	0h
24	GPIO248	R/W	配置 GPIO248 锁定（GPIO248 Configuration lock） 具体描述参考 GPIO224。	0h
25	GPIO249	R/W	配置 GPIO249 锁定（GPIO249 Configuration lock） 具体描述参考 GPIO224。	0h
26	GPIO250	R/W	配置 GPIO250 锁定（GPIO250 Configuration lock） 具体描述参考 GPIO224。	0h
27	GPIO251	R/W	配置 GPIO251 锁定（GPIO251 Configuration lock） 具体描述参考 GPIO224。	0h
28	GPIO252	R/W	配置 GPIO252 锁定（GPIO252 Configuration lock） 具体描述参考 GPIO224。	0h
29	GPIO253	R/W	配置 GPIO253 锁定（GPIO253 Configuration lock） 具体描述参考 GPIO224。	0h
30	GPIO254	R/W	配置 GPIO254 锁定（GPIO254 Configuration lock） 具体描述参考 GPIO224。	0h
31	保留			0h

21.9.49 端口 H 提交锁定寄存器（GPHCR）

偏移地址：0x3FC

复位类型：SYSRSn

该位提交 GPIOH 引脚的配置（GPIO224 到 GPIO254）锁定。一旦设置此位，只能通过复位清除。此寄存器将写入 GPHLOCK[GPIOy]寄存器中。

位/域	名称	R/W	描述	复位值
0	GPIO224	R/WSO	提交 GPIO224 锁定（GPIO224 lock is commit） 该位提交 GPHLOCK[GPIO224]锁定。 0：不提交锁定 GPHLOCK[GPIO224]的配置 1：提交锁定 GPHLOCK[GPIO224]的配置	0h
1	GPIO225	R/WSO	提交 GPIO224 锁定（GPIO225 lock is committed） 具体描述参考 GPIO224。	0h
2	GPIO226	R/WSO	锁定 GPIO226 锁定（GPIO226 lock is committed） 具体描述参考 GPIO224。	0h
3	GPIO227	R/WSO	锁定 GPIO227 锁定（GPIO227 lock is committed） 具体描述参考 GPIO224。	0h
4	GPIO228	R/WSO	锁定 GPIO228 锁定（GPIO228 lock is committed） 具体描述参考 GPIO224。	0h
5	GPIO229	R/WSO	锁定 GPIO229 锁定（GPIO229 lock is committed） 具体描述参考 GPIO224。	0h
6	GPIO230	R/WSO	锁定 GPIO230 锁定（GPIO230 lock is committed） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
7	GPIO231	R/WSO	锁定 GPIO231 锁定（GPIO231 lock is committed） 具体描述参考 GPIO224。	0h
8	GPIO232	R/WSO	锁定 GPIO232 锁定（GPIO232 lock is committed） 具体描述参考 GPIO224。	0h
9	GPIO233	R/WSO	锁定 GPIO233 锁定（GPIO233 lock is committed） 具体描述参考 GPIO224。	0h
10	GPIO234	R/WSO	锁定 GPIO234 锁定（GPIO234 lock is committed） 具体描述参考 GPIO224。	0h
11	GPIO235	R/WSO	锁定 GPIO235 锁定（GPIO235 lock is committed） 具体描述参考 GPIO224。	0h
12	GPIO236	R/WSO	锁定 GPIO236 锁定（GPIO236 lock is committed） 具体描述参考 GPIO224。	0h
13	GPIO237	R/WSO	锁定 GPIO237 锁定（GPIO237 lock is committed） 具体描述参考 GPIO224。	0h
14	GPIO238	R/WSO	锁定 GPIO238 锁定（GPIO238 lock is committed） 具体描述参考 GPIO224。	0h
15	GPIO239	R/WSO	锁定 GPIO239 锁定（GPIO239 lock is committed） 具体描述参考 GPIO224。	0h
16	GPIO240	R/WSO	锁定 GPIO240 锁定（GPIO240 lock is committed） 具体描述参考 GPIO224。	0h
17	GPIO241	R/WSO	锁定 GPIO241 锁定（GPIO241 lock is committed） 具体描述参考 GPIO224。	0h
18	GPIO242	R/WSO	锁定 GPIO242 锁定（GPIO242 lock is committed） 具体描述参考 GPIO224。	0h
19	GPIO243	R/WSO	锁定 GPIO243 锁定（GPIO243 lock is committed） 具体描述参考 GPIO224。	0h
20	GPIO244	R/WSO	锁定 GPIO244 锁定（GPIO244 lock is committed） 具体描述参考 GPIO224。	0h
21	GPIO245	R/WSO	锁定 GPIO245 锁定（GPIO245 lock is committed） 具体描述参考 GPIO224。	0h
22	GPIO246	R/WSO	锁定 GPIO246 锁定（GPIO246 lock is committed） 具体描述参考 GPIO224。	0h
23	GPIO247	R/WSO	锁定 GPIO247 锁定（GPIO247 lock is committed） 具体描述参考 GPIO224。	0h
24	GPIO248	R/WSO	锁定 GPIO248 锁定（GPIO248 lock is committed） 具体描述参考 GPIO224。	0h
25	GPIO249	R/WSO	锁定 GPIO249 锁定（GPIO249 lock is committed） 具体描述参考 GPIO224。	0h
26	GPIO250	R/WSO	锁定 GPIO250 锁定（GPIO250 lock is committed） 具体描述参考 GPIO224。	0h
27	GPIO251	R/WSO	锁定 GPIO251 锁定（GPIO251 lock is committed） 具体描述参考 GPIO224。	0h
28	GPIO252	R/WSO	锁定 GPIO252 锁定（GPIO252 lock is committed） 具体描述参考 GPIO224。	0h

位/域	名称	R/W	描述	复位值
29	GPIO253	R/WSO	锁定 GPIO253 锁定（GPIO253 lock is committed） 具体描述参考 GPIO224。	0h
30	GPIO254	R/WSO	锁定 GPIO254 锁定（GPIO254 lock is committed） 具体描述参考 GPIO224。	0h
31	保留			0h

21.9.50 端口 A 数据寄存器（GPADAT）

偏移地址：0x00

复位类型：SYSRSn

读取此寄存器中的位时对应的引脚将返回限定和反相（反相可选）后的输入电平。输入值始终可读，而不考虑引脚被配置成 GPIO 输出还是外设信号功能。系统复位时，所有的输出数据锁存器都将清零。

写入此寄存器会清除或设置相应的输出数据锁存器，驱动引脚操作具体如下：

（1） 当引脚配置成 GPIO 输出功能并使能时，将此引脚驱动为高电平或低电平；

（2） 当引脚未配置成 GPIO 输出功能时，写入值将被锁存并忽略。只有当引脚随后被重新配置成 GPIO 输出功能时，锁存的值才会驱动到此引脚上。

注意：由于读取值和写入值之间存在的延迟，按照读取-修改-写入的操作顺序可能会破坏输出数据锁存器的状态。GPADAT 寄存器中表示的是引脚的状态而非锁存的状态，即数据寄存器中反映的是实际的引脚值，但是从写入 GPADAT 到更新的引脚值反馈回寄存器的过程中存在延迟。当在随后的程序语句中使用此数据寄存器来改变 GPIO 引脚的状态时，可能会产生问题。因此，当使用 GPADAT 寄存器更改 GPIO 输出引脚的电平时，需注意是否干扰了其他引脚的电平的修改。

解决方案：

（1） 在指令间放置一些 NOP；

（2） 通过使用替代 GPADAT 寄存器的 GPASET/GPACLR/GPATOGGLE 寄存器。由于这三类寄存器写 0 无效，读回 0 的特性，可以指定仅需要更改的位，而不会干扰当前正在进行更改的其他位，使用这三类寄存器可以安全地控制输出数据锁存器。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	GPIOy 的读写数据位（GPIOy Read/Write Data Bit） （n=0…31）	0h

21.9.51 端口 A 设置寄存器（GPASET）

偏移地址：0x04

复位类型：SYSRSn

此寄存器的所有位始终读为 0，写 0 无效，对此寄存器写 1 会将对应的输出数据锁存器设置为高电平，驱动引脚操作具体如下：

（1） 当引脚配置成 GPIO 输出功能并使能时，将此引脚驱动为高电平；

（2） 当引脚未配置成 GPIO 输出功能时，写入 1 将被锁存，但是不会驱动此引脚。只有当引脚随后被配置成 GPIO 输出功能时，锁存的值才会驱动到此引脚上。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	设置 GPIOy 的输出数据锁存器（Set the GPIOy output data latch）（n=0…31）	0h

21.9.52 端口 A 清除寄存器（GPACLR）

偏移地址：0x08

复位类型：SYSRSn

此寄存器的所有位始终读为 0，写 0 无效，对此寄存器写 1 将清除对应的输出数据锁存器，驱动引脚操作具体如下：

（1） 当引脚配置成 GPIO 输出功能并使能时，将此引脚驱动为低电平；

（2） 当引脚未配置成 GPIO 输出功能时，写入 1 将被锁存，但是不会驱动此引脚。只有当引脚随后被配置成 GPIO 输出功能时，锁存的值才会驱动到此引脚上。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	清除 GPIOy 的输出数据锁存器（Clear the GPIOy output data latch）（n=0…31）	0h

21.9.53 端口 A 切换寄存器（GPATOGGLE）

偏移地址：0x0C

复位类型：SYSRSn

此寄存器的所有位始终读为 0，写 0 无效，对此寄存器写 1 将切换对应的输出数据锁存器的值，并以相反的方向驱动引脚，操作具体如下：

（1） 当引脚配置成 GPIO 输出功能并使能时，将此引脚驱动为相反的电平；

（2） 当引脚未配置成 GPIO 输出功能时，写入 1 将被锁存，但是不会驱动此引脚。只有当引脚随后被配置成 GPIO 输出功能时，锁存的值才会驱动到此引脚上。

位/域	名称	R/W	描述	复位值
y	GPIOy	R/W	切换 GPIOy 的输出数据锁存器（Toggle the GPIOy output data latch）（n=0…31）	0h

21.9.54 端口 B 数据寄存器（GPBDAT）

偏移地址：0x10

复位类型：SYSRSn

具体描述请参考端口 A 数据寄存器（GPADAT）。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	GPIO32 的读写数据位（GPIO32 Read/Write Data Bit）	0h
1	GPIO33	R/W	GPIO33 的读写数据位（GPIO33 Read/Write Data Bit）	0h
2	GPIO34	R/W	GPIO34 的读写数据位（GPIO34 Read/Write Data Bit）	0h
3	GPIO35	R/W	GPIO35 的读写数据位（GPIO35 Read/Write Data Bit）	0h
4	保留			0h
5	GPIO37	R/W	GPIO37 的读写数据位（GPIO37 Read/Write Data Bit）	0h

位/域	名称	R/W	描述	复位值
6	保留			0h
7	GPIO39	R/W	GPIO39 的读写数据位（GPIO39 Read/Write Data Bit）	0h
8	GPIO40	R/W	GPIO40 的读写数据位（GPIO40 Read/Write Data Bit）	0h
9	GPIO41	R/W	GPIO41 的读写数据位（GPIO41 Read/Write Data Bit）	0h
10	GPIO42	R/W	GPIO42 的读写数据位（GPIO42 Read/Write Data Bit）	0h
11	GPIO43	R/W	GPIO43 的读写数据位（GPIO43 Read/Write Data Bit）	0h
12	GPIO44	R/W	GPIO44 的读写数据位（GPIO44 Read/Write Data Bit）	0h
13	GPIO45	R/W	GPIO45 的读写数据位（GPIO45 Read/Write Data Bit）	0h
14	GPIO46	R/W	GPIO46 的读写数据位（GPIO46 Read/Write Data Bit）	0h
15	GPIO47	R/W	GPIO47 的读写数据位（GPIO47 Read/Write Data Bit）	0h
16	GPIO48	R/W	GPIO48 的读写数据位（GPIO48 Read/Write Data Bit）	0h
17	GPIO49	R/W	GPIO49 的读写数据位（GPIO49 Read/Write Data Bit）	0h
18	GPIO50	R/W	GPIO50 的读写数据位（GPIO50 Read/Write Data Bit）	0h
19	GPIO51	R/W	GPIO51 的读写数据位（GPIO51 Read/Write Data Bit）	0h
20	GPIO52	R/W	GPIO52 的读写数据位（GPIO52 Read/Write Data Bit	0h
21	GPIO53	R/W	GPIO53 的读写数据位（GPIO53 Read/Write Data Bit）	0h
22	GPIO54	R/W	GPIO54 的读写数据位（GPIO54 Read/Write Data Bit）	0h
23	GPIO55	R/W	GPIO55 的读写数据位（GPIO55 Read/Write Data Bit）	0h
24	GPIO56	R/W	GPIO56 的读写数据位（GPIO56 Read/Write Data Bit）	0h
25	GPIO57	R/W	GPIO57 的读写数据位（GPIO57 Read/Write Data Bit）	0h
26	GPIO58	R/W	GPIO58 的读写数据位（GPIO58 Read/Write Data Bit）	0h
27	GPIO59	R/W	GPIO59 的读写数据位（GPIO59 Read/Write Data Bit）	0h
31:28	保留			0h

21.9.55 端口 B 设置寄存器（GPBSET）

偏移地址：0x14

复位类型：SYSRSn

具体描述请参考端口 A 设置寄存器（GPASET）。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	设置 GPIO32 的输出数据锁存器（Set the GPIO32 output data latch）	0h
1	GPIO33	R/W	设置 GPIO33 的输出数据锁存器（Set the GPIO33 output data latch）	0h
2	GPIO34	R/W	设置 GPIO34 的输出数据锁存器（Set the GPIO34 output data latch）	0h
3	GPIO35	R/W	设置 GPIO35 的输出数据锁存器（Set the GPIO35 output data latch）	0h

位/域	名称	R/W	描述	复位值
4	保留			0h
5	GPIO37	R/W	设置 GPIO37 的输出数据锁存器（Set the GPIO37 output data latch）	0h
6	保留			0h
7	GPIO39	R/W	设置 GPIO39 的输出数据锁存器（Set the GPIO39 output data latch）	0h
8	GPIO40	R/W	设置 GPIO40 的输出数据锁存器（Set the GPIO40 output data latch）	0h
9	GPIO41	R/W	设置 GPIO41 的输出数据锁存器（Set the GPIO41 output data latch）	0h
10	GPIO42	R/W	设置 GPIO42 的输出数据锁存器（Set the GPIO42 output data latch）	0h
11	GPIO43	R/W	设置 GPIO43 的输出数据锁存器（Set the GPIO43 output data latch）	0h
12	GPIO44	R/W	设置 GPIO44 的输出数据锁存器（Set the GPIO44 output data latch）	0h
13	GPIO45	R/W	设置 GPIO45 的输出数据锁存器（Set the GPIO45 output data latch）	0h
14	GPIO46	R/W	设置 GPIO46 的输出数据锁存器（Set the GPIO46 output data latch）	0h
15	GPIO47	R/W	设置 GPIO47 的输出数据锁存器（Set the GPIO47 output data latch）	0h
16	GPIO48	R/W	设置 GPIO48 的输出数据锁存器（Set the GPIO48 output data latch）	0h
17	GPIO49	R/W	设置 GPIO49 的输出数据锁存器（Set the GPIO49 output data latch）	0h
18	GPIO50	R/W	设置 GPIO50 的输出数据锁存器（Set the GPIO50 output data latch）	0h
19	GPIO51	R/W	设置 GPIO51 的输出数据锁存器（Set the GPIO51 output data latch）	0h
20	GPIO52	R/W	设置 GPIO52 的输出数据锁存器（Set the GPIO52 output data latch	0h
21	GPIO53	R/W	设置 GPIO53 的输出数据锁存器（Set the GPIO53 output data latch）	0h
22	GPIO54	R/W	设置 GPIO54 的输出数据锁存器（Set the GPIO54 output data latch）	0h
23	GPIO55	R/W	设置 GPIO55 的输出数据锁存器（Set the GPIO55 output data latch）	0h
24	GPIO56	R/W	设置 GPIO56 的输出数据锁存器（Set the GPIO56 output data latch）	0h
25	GPIO57	R/W	设置 GPIO57 的输出数据锁存器（Set the GPIO57 output data latch）	0h
26	GPIO58	R/W	设置 GPIO58 的输出数据锁存器（Set the GPIO58 output data latch）	0h
27	GPIO59	R/W	设置 GPIO59 的输出数据锁存器（Set the GPIO59 output data latch）	0h

位/域	名称	R/W	描述	复位值
31:28	保留			0h

21.9.56 端口 B 清除寄存器（GPBCLR）

偏移地址：0x18

复位类型：SYSRSn

具体描述请参考端口 A 清除寄存器（GPACLR）。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	清除 GPIO32 的输出数据锁存器（Clear the GPIO32 output data latch）	0h
1	GPIO33	R/W	清除 GPIO33 的输出数据锁存器（Clear the GPIO33 output data latch）	0h
2	GPIO34	R/W	清除 GPIO34 的输出数据锁存器（Clear the GPIO34 output data latch）	0h
3	GPIO35	R/W	清除 GPIO35 的输出数据锁存器（Clear the GPIO35 output data latch）	0h
4	保留			0h
5	GPIO37	R/W	清除 GPIO37 的输出数据锁存器（Clear the GPIO37 output data latch）	0h
6	保留			0h
7	GPIO39	R/W	清除 GPIO39 的输出数据锁存器（Clear the GPIO39 output data latch）	0h
8	GPIO40	R/W	清除 GPIO40 的输出数据锁存器（Clear the GPIO40 output data latch）	0h
9	GPIO41	R/W	清除 GPIO41 的输出数据锁存器（Clear the GPIO41 output data latch）	0h
10	GPIO42	R/W	清除 GPIO42 的输出数据锁存器（Clear the GPIO42 output data latch）	0h
11	GPIO43	R/W	清除 GPIO43 的输出数据锁存器（Clear the GPIO43 output data latch）	0h
12	GPIO44	R/W	清除 GPIO44 的输出数据锁存器（Clear the GPIO44 output data latch）	0h
13	GPIO45	R/W	清除 GPIO45 的输出数据锁存器（Clear the GPIO45 output data latch）	0h
14	GPIO46	R/W	清除 GPIO46 的输出数据锁存器（Clear the GPIO46 output data latch）	0h
15	GPIO47	R/W	清除 GPIO47 的输出数据锁存器（Clear the GPIO47 output data latch）	0h
16	GPIO48	R/W	清除 GPIO48 的输出数据锁存器（Clear the GPIO48 output data latch）	0h
17	GPIO49	R/W	清除 GPIO49 的输出数据锁存器（Clear the GPIO49 output data latch）	0h
18	GPIO50	R/W	清除 GPIO50 的输出数据锁存器（Clear the GPIO50 output data latch）	0h
19	GPIO51	R/W	清除 GPIO51 的输出数据锁存器（Clear the GPIO51 output data latch）	0h

位/域	名称	R/W	描述	复位值
20	GPIO52	R/W	清除 GPIO52 的输出数据锁存器（Clear the GPIO52 output data latch）	0h
21	GPIO53	R/W	清除 GPIO53 的输出数据锁存器（Clear the GPIO53 output data latch）	0h
22	GPIO54	R/W	清除 GPIO54 的输出数据锁存器（Clear the GPIO54 output data latch）	0h
23	GPIO55	R/W	清除 GPIO55 的输出数据锁存器（Clear the GPIO55 output data latch）	0h
24	GPIO56	R/W	清除 GPIO56 的输出数据锁存器（Clear the GPIO56 output data latch）	0h
25	GPIO57	R/W	清除 GPIO57 的输出数据锁存器（Clear the GPIO57 output data latch）	0h
26	GPIO58	R/W	清除 GPIO58 的输出数据锁存器（Clear the GPIO58 output data latch）	0h
27	GPIO59	R/W	清除 GPIO59 的输出数据锁存器（Clear the GPIO59 output data latch）	0h
31:28	保留			0h

21.9.57 端口 B 切换寄存器（GPBTOGGLE）

偏移地址：0x1C

复位类型：SYSRSn

具体描述请参考端口 A 切换寄存器（GPATOGGLE）。

位/域	名称	R/W	描述	复位值
0	GPIO32	R/W	切换 GPIO32 的输出数据锁存器（Toggle the GPIO32 output data latch）	0h
1	GPIO33	R/W	切换 GPIO33 的输出数据锁存器（Toggle the GPIO33 output data latch）	0h
2	GPIO34	R/W	切换 GPIO34 的输出数据锁存器（Toggle the GPIO34 output data latch）	0h
3	GPIO35	R/W	切换 GPIO35 的输出数据锁存器（Toggle the GPIO35 output data latch）	0h
4	保留			0h
5	GPIO37	R/W	切换 GPIO37 的输出数据锁存器（Toggle the GPIO37 output data latch）	0h
6	保留			0h
7	GPIO39	R/W	切换 GPIO39 的输出数据锁存器（Toggle the GPIO39 output data latch）	0h
8	GPIO40	R/W	切换 GPIO40 的输出数据锁存器（Toggle the GPIO40 output data latch）	0h
9	GPIO41	R/W	切换 GPIO41 的输出数据锁存器（Toggle the GPIO41 output data latch）	0h
10	GPIO42	R/W	切换 GPIO42 的输出数据锁存器（Toggle the GPIO42 output data latch）	0h
11	GPIO43	R/W	切换 GPIO43 的输出数据锁存器（Toggle the GPIO43 output data latch）	0h

位/域	名称	R/W	描述	复位值
12	GPIO44	R/W	切换 GPIO44 的输出数据锁存器（Toggle the GPIO44 output data latch）	0h
13	GPIO45	R/W	切换 GPIO45 的输出数据锁存器（Toggle the GPIO45 output data latch）	0h
14	GPIO46	R/W	切换 GPIO46 的输出数据锁存器（Toggle the GPIO46 output data latch）	0h
15	GPIO47	R/W	切换 GPIO47 的输出数据锁存器（Toggle the GPIO47 output data latch）	0h
16	GPIO48	R/W	切换 GPIO48 的输出数据锁存器（Toggle the GPIO48 output data latch）	0h
17	GPIO49	R/W	切换 GPIO49 的输出数据锁存器（Toggle the GPIO49 output data latch）	0h
18	GPIO50	R/W	切换 GPIO50 的输出数据锁存器（Toggle the GPIO50 output data latch）	0h
19	GPIO51	R/W	切换 GPIO51 的输出数据锁存器（Toggle the GPIO51 output data latch）	0h
20	GPIO52	R/W	切换 GPIO52 的输出数据锁存器（Toggle the GPIO52 output data latch）	0h
21	GPIO53	R/W	切换 GPIO53 的输出数据锁存器（Toggle the GPIO53 output data latch）	0h
22	GPIO54	R/W	切换 GPIO54 的输出数据锁存器（Toggle the GPIO54 output data latch）	0h
23	GPIO55	R/W	切换 GPIO55 的输出数据锁存器（Toggle the GPIO55 output data latch）	0h
24	GPIO56	R/W	切换 GPIO56 的输出数据锁存器（Toggle the GPIO56 output data latch）	0h
25	GPIO57	R/W	切换 GPIO57 的输出数据锁存器（Toggle the GPIO57 output data latch）	0h
26	GPIO58	R/W	切换 GPIO58 的输出数据锁存器（Toggle the GPIO58 output data latch）	0h
27	GPIO59	R/W	切换 GPIO59 的输出数据锁存器（Toggle the GPIO59 output data latch）	0h
31:28	保留			0h

21.9.58 端口 H 数据寄存器（GPHDAT）

偏移地址：0x70

复位类型：SYSRSn

在这个寄存器中读取一个字段，在限定和(可选的)反转之后返回相应 IO 引脚的输入电平。在数字模式下，这些引脚仅支持输入功能。

位/域	名称	R/W	描述	复位值
0	GPIO224	R/W	GPIO224 的读写数据位（GPIO224 Read/Write Data Bit）	0h
1	GPIO225	R/W	GPIO225 的读写数据位（GPIO225 Read/Write Data Bit）	0h
2	GPIO226	R/W	GPIO226 的读写数据位（GPIO226 Read/Write Data Bit）	0h

位/域	名称	R/W	描述	复位值
3	GPIO227	R/W	GPIO227 的读写数据位（GPIO227 Read/Write Data Bit）	0h
4	GPIO228	R/W	GPIO228 的读写数据位（GPIO228 Read/Write Data Bit）	0h
5	GPIO229	R/W	GPIO229 的读写数据位（GPIO229 Read/Write Data Bit）	0h
6	GPIO230	R/W	GPIO230 的读写数据位（GPIO230 Read/Write Data Bit）	0h
7	GPIO231	R/W	GPIO231 的读写数据位（GPIO231 Read/Write Data Bit）	0h
8	GPIO232	R/W	GPIO232 的读写数据位（GPIO232 Read/Write Data Bit）	0h
9	GPIO233	R/W	GPIO233 的读写数据位（GPIO233 Read/Write Data Bit）	0h
10	GPIO234	R/W	GPIO234 的读写数据位（GPIO234 Read/Write Data Bit）	0h
11	GPIO235	R/W	GPIO235 的读写数据位（GPIO235 Read/Write Data Bit）	0h
12	GPIO236	R/W	GPIO236 的读写数据位（GPIO236 Read/Write Data Bit）	0h
13	GPIO237	R/W	GPIO237 的读写数据位（GPIO237 Read/Write Data Bit）	0h
14	GPIO238	R/W	GPIO238 的读写数据位（GPIO238 Read/Write Data Bit）	0h
15	GPIO239	R/W	GPIO239 的读写数据位（GPIO239 Read/Write Data Bit）	0h
16	GPIO240	R/W	GPIO240 的读写数据位（GPIO240 Read/Write Data Bit）	0h
17	GPIO241	R/W	GPIO241 的读写数据位（GPIO241 Read/Write Data Bit）	0h
18	GPIO242	R/W	GPIO242 的读写数据位（GPIO242 Read/Write Data Bit）	0h
19	GPIO243	R/W	GPIO243 的读写数据位（GPIO243 Read/Write Data Bit）	0h
20	GPIO244	R/W	GPIO244 的读写数据位（GPIO244 Read/Write Data Bit）	0h
21	GPIO245	R/W	GPIO245 的读写数据位（GPIO245 Read/Write Data Bit）	0h
22	GPIO246	R/W	GPIO246 的读写数据位（GPIO246 Read/Write Data Bit）	0h
23	GPIO247	R/W	GPIO247 的读写数据位（GPIO247 Read/Write Data Bit）	0h
24	GPIO248	R/W	GPIO248 的读写数据位（GPIO248 Read/Write Data Bit）	0h
25	GPIO249	R/W	GPIO249 的读写数据位（GPIO249 Read/Write Data Bit）	0h

位/域	名称	R/W	描述	复位值
26	GPIO250	R/W	GPIO250 的读写数据位（GPIO250 Read/Write Data Bit）	0h
27	GPIO251	R/W	GPIO251 的读写数据位（GPIO251 Read/Write Data Bit）	0h
28	GPIO252	R/W	GPIO252 的读写数据位（GPIO252 Read/Write Data Bit）	0h
29	GPIO253	R/W	GPIO253 的读写数据位（GPIO253 Read/Write Data Bit）	0h
30	GPIO254	R/W	GPIO254 的读写数据位（GPIO254 Read/Write Data Bit）	0h
31	保留			0h

22 处理器间通信单元（IPC）

22.1 简介

在多核处理器中，不同的处理器核之间可以通过数据共享、消息传递、中断控制等方式进行 IPC 核间通讯。IPC 只能通过 Arm 提供的协处理器接口（协处理器编号 1）访问。

中断控制和数据共享（邮箱）：

- 每个 CPU 可以生成 4 个寄存器共享的数据和 12 个中断。
- 事件控制
- 使用轮询方法和 RXEV 实现两个 CPU 之间的通信。
- IPC 只能通过 Arm 提供的协处理器接口（协处理器编号 1）访问。请参阅 STAR 的文档以获取更多的说明信息和软件指南。

22.2 主要特征

（1） 中断控制和数据共享（邮箱）：每个 CPU 可以生成 4 个寄存器共享的数据和 12 个中断

（2） 事件控制：使用轮询方法和 RXEV 实现两个 CPU 之间的通信

（3） WRPRT：实现了 WRPRT 函数结构框图

22.3 功能描述

22.3.1 事件控制相关寄存器操作

22.3.1.1 TASK_REG

（1） CPU0 侧

- 当 CPU0 读取 TASK_REG 时，它读取的值不是来自它之前写入的数据，而是 CPU1 侧的 TASK_REG 的值。
- 寄存器的写入将更新 CPU1 侧的 TASK_REG

（2） CPU1 侧

- CPU1 读取 TASK_REG 时，其读取的值不是来自其先前写入的数据，而是 CPU0 侧的 TASK_REG 的值。
- 寄存器的写入将更新 CPU0 侧的 TASK_REG

22.3.1.2 STATE_REG

（1） CPU0 侧

- 读取时，CPU1 表示正在执行的任务的寄存器。CPU0 可以读取该寄存器以获得 CPU1 正在执行的任务。
- 寄存器的写入将更新 CPU1 侧的 STATE_REG

（2） CPU1 侧

- 读取时，CPU0 表示正在执行的任务的寄存器。CPU1 可以读取该寄存器以获得 CPU0 正在执行的任务。
- 寄存器的写入将更新 CPU0 侧的 STATE_REG

22.3.2 生成唤醒

该模块通过比较 TAST0_REG 寄存器值是否为 0，在睡眠模式下向 MCU1 生成唤醒信号。当 MCU1 处于运行模式时，将不会产生唤醒信号，因为它将检查 TAST0I_REG 寄存器是否为零。

22.3.3 生成中断

该模块通过 IPC.SR_REG 中的状态位向每个 CPU 产生 12 个中断信号和 IPC.CTRL_REG 中相应的中断启用位，共 24 个。以下列出了所有中断：

（1） CPU0 的中断：

- TX_IRQ0[3:0]：CPU0 发送寄存器空中断，当 CPU0 侧的 SR_REG.TEn 和 CTRL_REG.TIEn 都有效时，TX_IRQ0[n]。
- RX_IRQ0[3:0]：CPU0 接收寄存器满中断，当 CPU0 侧的 SR_REG.RFn 和 CTRL_REG.RIEn 都有效时，RX_IRQO[n]。
- GP_IRQ0[3:0]：CPU0 通用中断，当 CPU0 侧的 SR_REG.GIPn 和 CTRL_REG.GIEn 都有效时，GP_IRQ0[n]。

（2） CPU1 中断：

- TX_IRQ1[3:0]：CPU1 发送寄存器空中断，当 CPU1 侧的 SR_REG.TEn 和 CTRL_REG.TIEn 都被断言时，TX_IRQ1[n]。
- RX_IRQ1[3:0]：CPU1 接收寄存器满中断，当 CPU1 侧的 SR_REG.RFn 和 CTRL_REG.RIEn 都被断言时，RX_IRQ1[n] 。
- GP_IRQ1[3:0]：CPU1 通用中断，当 CPU1 侧的 SR_REG.GIPn 和 CTRL_REG.GIEn 都有效时，GP_IRQ1[n]。

22.3.4 编程指南

22.3.4.1 寄存器访问

详情请参见 SDK 说明文档。

22.3.4.2 数据共享与中断控制

消息传递逻辑与外部存储器结合使用。有各种消息传递方法，可用于落实消息传递协议。其中一些消息可能具有以下含义：“从内存中的偏移量 X 开始，刚刚写入了一条含 N 个字的消息”，或者“刚刚读取了之前发送的数据块”。使消息传递逻辑独立于存储器阵列并不将用户限制为预定义的硬件协议。另一方面，管理消息传递所需的软件简洁明了。大多数消息传递机制是对称的；它们是重复的并且在 CPU1 侧和 CPU0 侧都可用。消息传递机制包括：

（1） 四个 32 位数据发送寄存器，每个寄存器反映在另一个处理器侧的 4 个只读接收寄存器中。用户可以使用这些寄存器来传输 32 位字消息或写入共享存储器的消息的帧信息（字数、初始地址和消息类型代码）。

（2） 对发送器侧的发送寄存器的写入将清除发送器侧状态寄存器中的“发送器空”位，并在接收器侧的状态寄存器中设置“接收满”位。在接收器侧的位设置可以选择性地触发接收器侧的中断（可屏蔽的接收中断）。

（3） 读取接收器侧的一个接收寄存器会清除接收器侧状态寄存器中的“接收器满”位，并设置发送器侧状态寄存器的“发送器空”位。“发送器空”位的设置可以选择性地触发发送器侧的中断（可屏蔽的发送中断）。

四个通用标志反映在接收器侧的状态寄存器中——对任何保留位置的读/写访问和对 IPC 的 CPU0 侧只读寄存器的写入将生成对 CPU0 的总线错误确认（将发出硬故障中断）。对任何保留位置的读/写访问以及对 IPC 的 CPU1 侧的只读寄存器的写入将生成对 CPU1 的模块传输错误确认。

22.3.4.3 处理器中断

从 IPC 到 CPU0 有 12 个中断源，到 CPU1 有 12 个：

（1） 每个接收寄存器有四个接收中断：当处理器出现以下情况时，有效
- 设置接收完整位（SR_REG.RFn）并且启用 CTRL_REG.RIEn 位时

（2） 每个发送寄存器有四个传输中断：当处理器传输空位（SR_REG.TEn）被设置并且 CTRL_REG.TIEn 位被启用时，有效

（3） 四个通用中断：当 SR_REG.GIPn 位被设置并且 CTRL_REG.GIEn 位被启用时，有效

所有中断在处理器控制寄存器（CTRL_REG）中都是可屏蔽的。IPC 不承担这些中断的任何内部优先级。一次性多个中断（例如，接收 0 和接收 1 中断或任何发送中断和通用中断）可能有效。这些中断的优先级应由芯片级的中断控制器来解决。通用中断待决位（GIP0、GIP1、GIP2 和 GIP3）应由软件清除（作为中断服务例程的一部分），以使中断控制器的请求无效。

22.3.4.4 通用中断清除序列

当一个处理器写入通用中断位（GIR）时，写入事件与另一个处理器时钟同步，以设置通用中断请求待决位（GIP）。当设置 GIP 位时，如果通用中断在接收处理器侧被启用（GIE 位被设置），则发送处理器通用中断被发送至接收处理器。接收处理器通过在 GIP 位上写入“1”来清除此中断。一旦写入 GIP 位，中断即无效。GIP 位的写入事件与其他处理器时钟同步。同步信号清除 GIR 位。在清除 GIR 位之前，软件不应再次写入 GIR 位。

22.3.5 使用中断的消息传递协议

消息传递硬件可以由软件用来实施用于多种消息类型的消息传递协议。完全支持中断和轮询管理。

22.3.5.1 基本中断和数据共享示例

下面的示例通过数据相关中断说明了消息传递序列。这是邮箱的基本用法。

在本例中，处理器想要传输一个字，并根据共享数据的内容中断另一个处理器以

执行某些操作。

（1） 写入序列

- 处理器按顺序将消息信息写入其发送寄存器 n（n 可以是 0~3）
- 当发生对发送寄存器 n 的写入时，SR_REG 的 RFn 位在同步后被设置，并且它立即触发对另一个处理器的接收 n 中断

（2） 读取序列

- 另一个处理器接收接收 n 中断，并开始读取从接收寄存器传输的消息。
- 在读取接收寄存器 n 之后，中断位被清除。

22.3.6 事件控制

以下内容说明了事件控制流的使用。

（1） 第 1 步：CPU0 通过协处理器指令 MCR 写入 COP TASK1_REG。此操作将向 CPU1 产生疲弱信号。

（2） 第 2 步：CPU1 从 SLEEP 模式访问 RUN 模式，并通过协处理器指令 MRC 读取 COP TASK1_REG。此操作将清除 TASK1_REG。

（3） 第 3 步：CPU0 可以通过协处理器指令 MRC 读取 COP 的 STATE1_REG，以了解 CPU1 的状态。

（4） 第 4 步：CPU1 完成任务时，将“0”写入寄存器 STATE1_REG 以清除它。

（5） 第 5 步：CPU1 读取 TASK1_REG，如果 TASK1_REG 值不为零，CPU1 执行相应的任务，否则 CPU 将进入睡眠模式。

22.3.7 将邮箱与事件控制进行比较

（1） 它们均设计成在 CPU 之间共享信息。

（2） 它们的机制是相当独立的。

（3） 它们都可以用来唤醒另一个 CPU，但邮箱使用中断，事件控制使用事件。

（4） TX_REG/RX_REG/SR_REG/CTRL_REG 属于邮箱，TASK_REG 和 STATE_REG 属于事件控制。

22.4 寄存器地址映射

表格 71 CPU0 侧寄存器地址映射

寄存器	描述	偏移地址
TX_REG0	CPU0 数据发送寄存器	0x0
TX_REG1	CPU0 数据发送寄存器	0x1
TX_REG2	CPU0 数据发送寄存器	0x2
TX_REG3	CPU0 数据发送寄存器	0x3

寄存器	描述	偏移地址
RX_REG0	CPU0 数据接收寄存器	0x4
RX_REG1	CPU0 数据接收寄存器	0x5
RX_REG2	CPU0 数据接收寄存器	0x6
RX_REG3	CPU0 数据接收寄存器	0x7
SR_REG	CPU0 邮箱状态寄存器	0x8
CTRL_REG	CPU0 邮箱控制寄存器	0x9
TASK_REG	CPU0 任务寄存器	0xA
STATE_REG	CPU0 状态寄存器	0xB

表格 72 CPU1 侧寄存器地址映射

寄存器	描述	偏移地址
TX_REG0	CPU1 数据发送寄存器	0x0
TX_REG1	CPU1 数据发送寄存器	0x1
TX_REG2	CPU1 数据发送寄存器	0x2
TX_REG3	CPU1 数据发送寄存器	0x3
RX_REG0	CPU1 数据接收寄存器	0x4
RX_REG1	CPU1 数据接收寄存器	0x5
RX_REG2	CPU1 数据接收寄存器	0x6
RX_REG3	CPU1 数据接收寄存器	0x7
SR_REG	CPU1 邮箱状态寄存器	0x8
CTRL_REG	CPU1 邮箱控制寄存器	0x9
TASK_REG	CPU1 任务寄存器	0xA
STATE_REG	CPU1 状态寄存器	0xB

22.5 寄存器功能描述

22.5.1 CPU0 数据发送寄存器（TX_REGn）

偏移地址：0x00+n*0x01（n=0…3）

位/域	名称	R/W	描述	复位值
31:0	DATA	R/W	发送数据（Transmit data） 用于 CPU0 向 CPU1 发送数据的寄存器。	0h

22.5.2 CPU0 数据接收寄存器（RX_REGn）

偏移地址：0x04+n*0x01（n=0…3）

位/域	名称	R/W	描述	复位值
31:0	DATA	R/W	接收数据（receive data） 用于 CPU0 向 CPU1 接收数据的寄存器。	0h

22.5.3 CPU0 邮箱状态寄存器（SR_REG）

偏移地址：0x08

位/域	名称	R/W	描述	复位值
3:0	TEn	R	CPU0 侧的 IPC.TX_REGn 为空（IPC.TX REGn on the CPU0 side is empty）（n=0~3）	Fh

位/域	名称	R/W	描述	复位值
			当 IPC 复位时，TEn 将设置为“1”。 当 CPU1 读取 CPU1 侧的 RX_REGn 后，TEn 将设置为“1”。 当 CPU0 写入 TX_REGn 时，TEn 被清除。 0：IPC TX_REGn 不为空 1：IPC TX_REGn 为空	
7:4	保留			0h
11:8	RFn	R	CPU0 侧的 RX_REGn 为满（RX REGn of CPU0 is full）（n=0~3） 当 IPC 复位时，REn 将设置为“0”。 当 CPU1 写入 CPU1 侧的 TX_REGn 之后，REn 将设置为“1”。 当读取 RX_REGn（在 CPU0 侧）时，REn 被清除。 0：RX_REGn 未满 1：RX_REGn 已满	0h
15:12	保留			0h
19:16	GIPn	RC_W1	CPU0 通用中断请求 n 是待决的（CPU0 Universal Interrupt request n is pending）（n=0~3） 当 IPC 复位时，GIPn 将设置为“0”。 当 CPU1 侧的 CTRL_REG.GIRn 设置后，GIPn 将设置为“1”。如果 CPU0 侧的 CTRL_REG.GIEn 设置为“1”，则 CPU0 将产生中断。 写入“1”以清除此位。将“0”写入此位无效。 0：CPU0 通用中断 n 不是待决的 1：CPU0 通用中断 n 是待决的	0h
30:20	保留			0h
31	LPS	RO	CPU0 的低功率状态（CPU0 Indicates the low power status） 当 IPC 复位时，LPS 将设置为“0”。 当 CPU0 处于低功率模式并需要唤醒时，LPS 将设置为“1”。CPU0 可以读取该位以获得 CPU0 的低功率状态，无论 CPU0 执行 WFE 还是 WFI。 0：CPU0 不处于低功率模式 1：CPU0 处于低功率模式	0h

22.5.4 CPU0 邮箱控制寄存器（TIMING_CTRL_REG）

偏移地址：0x09

位/域	名称	R/W	描述	复位值
3:0	TIEn	R/W	CPU0 发送寄存器空中断启用 n（CPU0 Transmit register air break enabled n）（n=0~3） TIEn 位启用 CPU0 发送寄存器空中断 n。 如果 TIEn 位设置为“1”（启用），则当 CPU0 中的 SR_REG 寄存器中的 TEn 位设置为“1”时，会发出 CPU0 传输中断 n 请求。 如果 TIEn（在 CPU0 侧）位被清除（禁用），则 TEn 位的值被忽略，并且不会发出 CPU0 传输中断 n 请求。 当 IPC 复位时，TIEn 位被清除。	0h

位/域	名称	R/W	描述	复位值
			0：禁用 CPU0 发送寄存器空中断 n 1：启用 CPU0 发送寄存器空中断 n	
7:4	保留			0h
11:8	RIEn	R/W	CPU0 接收寄存器满中断启用 n（CPU0 Receive register full interrupt enabled n）（n=0~3） RIEn 位启用 CPU0 接收中断 n。 如果 RIEn 位设置为“1”（启用），则当 CPU0 侧的 IPC.SR_REG 寄存器中的 RFn 位设置为“1”时，发出 CPU0 接收中断 n 请求。 如果 RIEn 位被清除（禁用），则 RFn 位的值被忽略，并且不会发出 CPU0 接收中断 n 请求。 当 IPC 复位时，RIEn 位被清除。 0：禁用 CPU0 寄存器满接收中断 n 1：启用 CPU0 寄存器满接收中断 n	0h
15:12	保留			0h
19:16	GIEn	R/W	CPU0 通用中断启用 n（CPU0 Universal interrupt enabled n）（n=0~3） GIEn 位启用 CPU0 通用中断 n。 如果两者都是 CPU0.IPC.SR_REG.GIEn 和 CPU0.IPC，则通用中断 n 将发送至 CPU0。SR_REG.GIPn 为高。 如果 GIEn 被清除（禁用），则 GIPn 位的值被忽略，并且不会向 CPU0 发送通用中断 n。 将发送请求。 当 IPC 复位时，GIEn 位被清除。 0：禁用 CPU0 通用中断 n 1：启用 CPU0 通用中断 n	0h
23:20	GIRn	R/W	CPU0 通用中断请求 n（CPU0 General Interrupt request n）（n=0~3） 将“1”写入 GIRn 位将设置 CPU1 侧的 IPC.SR_REG 寄存器中的 GIPn 位。如果 IPC.SR_REG 寄存器中的 GIEn 位设置为 CPU1 侧的“1”，将向 CPU1 发出通用中断 n 请求。 如果 GIPn 位（在 CPU1 侧的 IPC.SR_REG 寄存器中）被清除，则 GIRn 位被清除，从而向 CPU0 发出中断被接受（由软件清除）的信号。 为确保正确操作，在设置 GIPn 位（GIRn 位）之前，验证该位是否已被清除（意思是没有待决的中断）。 当 IPC 复位时，GIRn 位被清除。 0：CPU0 通用中断 n 未被请求发送至 CPU1。 1：CPU0 通用中断 n 请求发送至 CPU1。	0h
30:24	保留			0h
31	IPCR	R/W	CPU0 IPC 复位请求（CPU0 IPC reset request） 将 IPCR 位设置为“1”将重置 IPC 的 CPU1 和 CPU0 侧，强制所有 CTRL_REG 和 SR_REG 为其默认值，并清除所有内部状态。数据寄存器将不会重置。 在将 IPCR 位设置为“1”之前，建议中断 CPU1，因为设置 IPCR 位可能会影响正在进行的 CPU1 程序。 IPCR 位只能写为“1”。 IPCR 位总是读取为“0”。	0h

位/域	名称	R/W	描述	复位值
			IPCR 位在 IPC 复位序列期间被清除。 0：自清除位（默认）。 不适用。	

22.5.5 CPU0 任务寄存器（TASK_REG）

偏移地址：0x0A

位/域	名称	R/W	描述	复位值
31:0	TASK	R/W	执行任务（Perform Task） 写入时，CPU0 表示要执行的任务的寄存器。 读取时，CPU0 从 CPU1 获取要执行的任务的寄存器	0h

22.5.6 CPU0 状态寄存器（STATE_REG）

偏移地址：0x14

位/域	名称	R/W	描述	复位值
31:0	STATE	R/W	执行任务状态（Perform task status） 读取时，CPU1 表示正在执行的任务的寄存器。CPU0 可以读取该寄存器以获得 CPU1 正在执行的任务。 写入时，CPU0 将当前任务编号更新为 CPU1。	0h

22.5.7 CPU1 数据发送寄存器（TX_REGn）

偏移地址：0x00+n*0x01（n=0…3）

位/域	名称	R/W	描述	复位值
31:0	DATA	R/W	发送数据（Transmit data） 用于 CPU1 向 CPU0 发送数据的寄存器。	0h

22.5.8 CPU1 数据接收寄存器（RX_REGn）

偏移地址：0x04+n*0x01（n=0…3）

位/域	名称	R/W	描述	复位值
31:0	DATA	R/W	接收数据（receive data） 用于 CPU1 向 CPU0 接收数据的寄存器。	0h

22.5.9 CPU1 邮箱状态寄存器（SR_REG）

偏移地址：0x08

位/域	名称	R/W	描述	复位值
3:0	TEn	R	CPU1 侧的 IPC.TX_REGn 为空（IPC.TX REGn on the CPU1 side is empty）（n=0~3） 当 IPC 复位时，TEn 将设置为“1”。 当 CPU0 读取 CPU0 侧的 RX_REGn 后，TEn 将设置为“1”。 当 CPU1 写入 TX_REGn 时，TEn 被清除。 0：IPC TX_REGn 不为空 1：IPC TX_REGn 为空	Fh
7:4	保留			0h
11:8	RFn	R	CPU1 侧的 RX_REGn 为满（RX REGn of CPU1 is full）（n=0~3）	0h

位/域	名称	R/W	描述	复位值
			当 IPC 复位时，REn 将设置为“0”。 当 CPU1 写入 CPU1 侧的 TX_REGn 之后，REn 将设置为“1”。 当读取 RX_REGn（在 CPU1 侧）时，REn 被清除。 0：RX_REGn 未满 1：RX_REGn 已满	
15:12	保留			0h
19:16	GIPn	RC_W1	CPU1 通用中断请求 n 是待决的（CPU1 Universal Interrupt request n is pending）（n=0~3） 当 IPC 复位时，GIPn 将设置为“0”。 当 CPU1 侧的 CTRL_REG.GIRn 设置后，GIPn 将设置为“1”。如果 CPU1 侧的 CTRL_REG.GIEn 设置为“1”，则 CPU1 将产生中断。 写入“1”以清除此位。将“0”写入此位无效。 0：CPU1 通用中断 n 不是待决的 1：CPU1 通用中断 n 是待决的	0h
30:20	保留			0h
31	LPS	RO	CPU1 的低功率状态（CPU1 Indicates the low power status） 当 IPC 复位时，LPS 将设置为“0”。 当 CPU1 处于低功率模式并需要唤醒时，LPS 将设置为“1”。CPU1 可以读取该位以获得 CPU1 的低功率状态，无论 CPU1 执行 WFE 还是 WFI。 0：CPU1 不处于低功率模式 1：CPU1 处于低功率模式	0h

22.5.10 CPU1 邮箱控制寄存器（TIMING_CTRL_REG）

偏移地址：0x09

位/域	名称	R/W	描述	复位值
3:0	TIEn	R/W	CPU1 发送寄存器空中断启用 n（CPU1 Transmit register air break enabled n）（n=0~3） TIEn 位启用 CPU1 发送寄存器空中断 n。 如果 TIEn 位设置为“1”（启用），则当 CPU1 中的 SR_REG 寄存器中的 TEn 位设置为“1”时，会发出 CPU1 传输中断 n 请求。 如果 TIEn（在 CPU1 侧）位被清除（禁用），则 TEn 位的值被忽略，并且不会发出 CPU1 传输中断 n 请求。 当 IPC 复位时，TIEn 位被清除。 0：禁用 CPU1 发送寄存器空中断 n 1：启用 CPU1 发送寄存器空中断 n	0h
7:4	保留			0h
11:8	RIEn	R/W	CPU1 接收寄存器满中断启用 n（CPU1 Receive register full interrupt enabled n）（n=0~3） RIEn 位启用 CPU1 接收中断 n。 如果 RIEn 位设置为“1”（启用），则当 CPU1 侧的 IPC.SR_REG 寄存器中的 RFn 位设置为“1”时，发出 CPU1 接收中断 n 请求。	0h

位/域	名称	R/W	描述	复位值
			如果 RIEn 位被清除（禁用），则 RFn 位的值被忽略，并且不会发出 CPU1 接收中断 n 请求。 当 IPC 复位时，RIEn 位被清除。 0：禁用 CPU1 寄存器满接收中断 n 1：启用 CPU1 寄存器满接收中断 n	
15:12	保留			0h
19:16	GIEn	R/W	CPU1 通用中断启用 n（CPU1 Universal interrupt enabled n）（n=0~3） 对于 n={0~3}，CPU1 通用中断启用 n。 GIEn 位启用 CPU1 通用中断 n。 如果两者都是 CPU1.IPC.SR_REG.GIEn 和 CPU1.IPC，则通用中断 n 将发送至 CPU1。SR_REG.GIPn 为高。 如果 GIEn 被清除（禁用），则 GIPn 位的值被忽略，并且不会向 CPU1 发送通用中断 n。 将发送请求。 当 IPC 复位时，GIEn 位被清除。 0：禁用 CPU1 通用中断 n 1：启用 CPU1 通用中断 n	0h
23:20	GIRn	R/W	CPU1 通用中断请求 n（CPU1 General Interrupt request n）（n=0~3） 将“1”写入 GIRn 位将设置 CPU1 侧的 IPC.SR_REG 寄存器中的 GIPn 位。如果 IPC.SR_REG 寄存器中的 GIEn 位设置为 CPU1 侧的“1”，将向 CPU1 发出通用中断 n 请求。 如果 GIPn 位（在 CPU1 侧的 IPC.SR_REG 寄存器中）被清除，则 GIRn 位被清除，从而向 CPU1 发出中断被接受（由软件清除）的信号。 为确保正确操作，在设置 GIPn 位（GIRn 位）之前，验证该位是否已被清除（意思是没有待决的中断）。 当 IPC 复位时，GIRn 位被清除。 0：CPU1 通用中断 n 未被请求发送至 CPU1 1：CPU1 通用中断 n 请求发送至 CPU1	0h
30:24	保留			0h
31	IPCR	R/W	CPU1 IPC 复位请求（CPU1 IPC reset request） 将 IPCR 位设置为“1”将重置 IPC 的 CPU1 和 CPU1 侧，强制所有 CTRL_REG 和 SR_REG 为其默认值，并清除所有内部状态。数据寄存器将不会重置。 在将 IPCR 位设置为“1”之前，建议中断 CPU1，因为设置 IPCR 位可能会影响正在进行的 CPU1 程序。 IPCR 位只能写为“1”。 IPCR 位总是读取为“0”。 IPCR 位在 IPC 复位序列期间被清除。 0：自清除位（默认）。 不适用。	0h

22.5.11 CPU1 任务寄存器（TASK_REG）

偏移地址：0x0A

位/域	名称	R/W	描述	复位值
31:0	TASK	R/W	执行任务（Perform Task） 写入时，CPU1 表示要执行的任务的寄存器。 读取时，CPU1 从 CPU1 获取要执行的任务的寄存器	0h

22.5.12 CPU1 状态寄存器（STATE_REG）

偏移地址：0x14

位/域	名称	R/W	描述	复位值
31:0	STATE	R/W	执行任务状态（Perform task status） 读取时，CPU0 表示正在执行的任务的寄存器。CPU1 可以读取该寄存器以获得 CPU0 正在执行的任务。 写入时，CPU1 将当前任务编号更新为 CPU0。	0h

23 X-BAR

23.1 简介

X-BAR 提供了灵活的交叉连接结构，可以实现输入，输出和内部资源之间的数据的快速传输。根据 X-BAR 接收信号的位置，分为 Input X-BAR，Output X-BAR，FLB X BAR 和 PWM X-BAR。

23.2 主要特征

（1） 灵活的交叉连接结构

（2） Input X-BAR：将外部信号经过 GPIO 引脚输入到设备中

（3） Output X-BAR：将内部信号从设备输出到 GPIO 引脚

（4） FLB X BAR：将信号传输到 FLB 模块

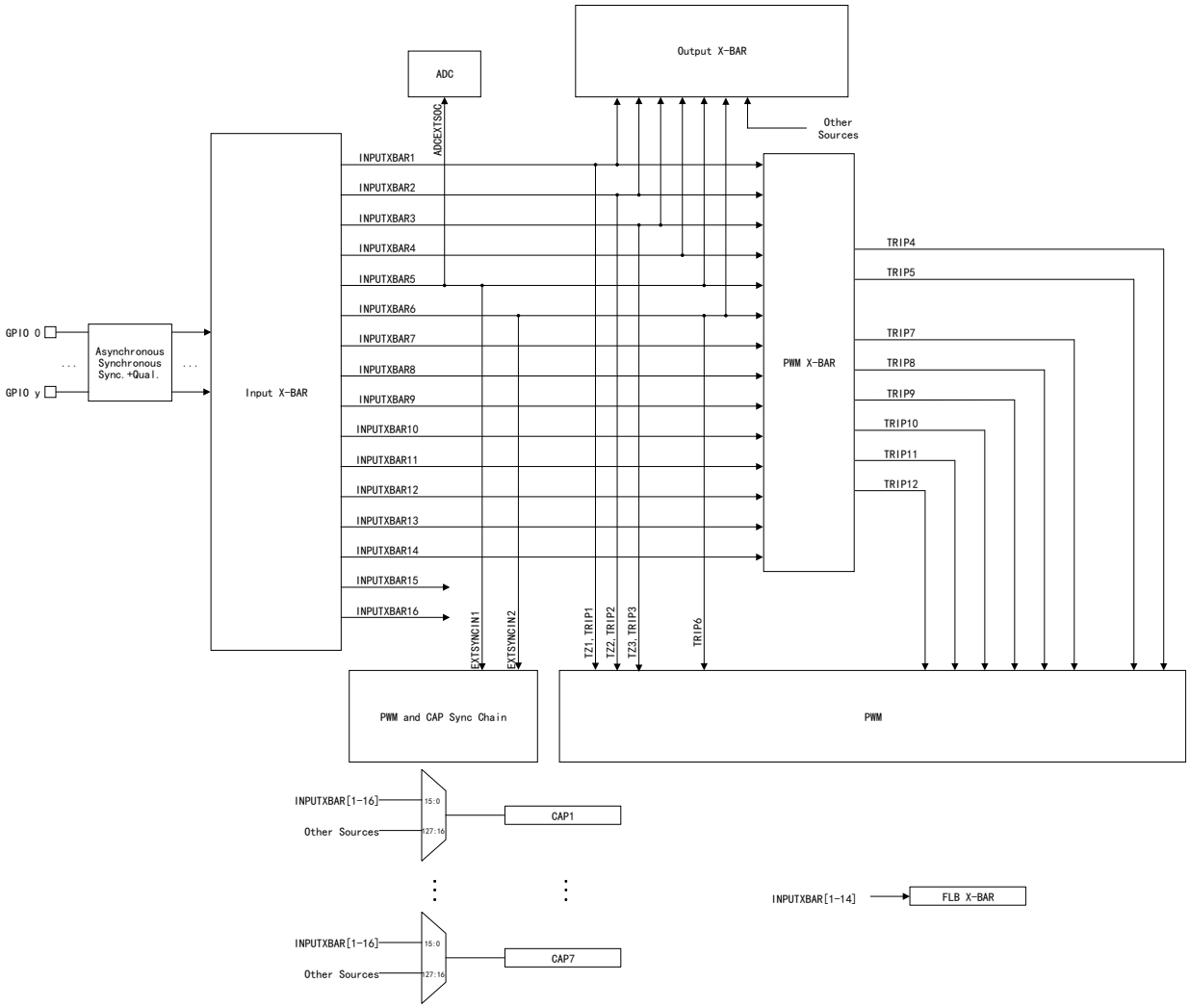
（5） PWM X-BAR：将信号传输到 PWM 模块

23.3 功能描述

23.3.1 Input X-BAR

23.3.1.1 结构框图

图 20 Input X-BAR 结构框图



23.3.1.2 信号连接

Input X-BAR 由 16 个 INPUT 信号构成，可以访问每个 GPIO 引脚（包括 AIO 的数字输入引脚），并将外部信号经过 GPIO 引脚输入到一个或多个不同的 IP 模块中。这些 IP 模块包括外部中断，PWM，ADC 和 CAP。由于只需要任何一个可用的 GPIO 即可实现灵活的信号路由，因此使用 Input X-BAR 可以减轻外设复用的一些限制。Input X-BAR 仅将输入缓冲区中的信号连接到所选择的目标 IP 模块上，Input X-BAR 的操作不会受到 GPIO 复用功能选择的影响。通过使用 Input X-BAR，也可以将一个外设的输出信号路由到另一个外设上，例如进行频率测试时，使用 CAP 测量 PWM 的输出。

通过 INPUTxSELECT 寄存器可以配置每个 INPUT 信号的 GPIO，更多 INPUT

信号配置的信息，详情请参见 XBAR_INPUT 寄存器组。每个 INPUT 信号可选择的目标 IP 模块如下表所示：

表格 73 Input X-BAR 目标 IP 模块

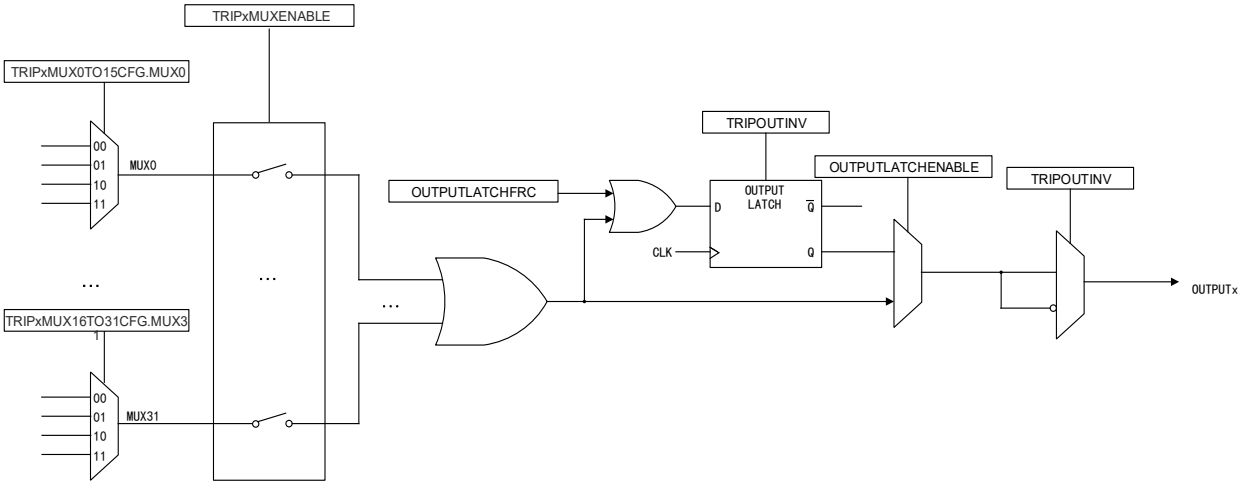
INPUTx	目标 IP 模块
INPUT1	Output X-BAR，PWM X-BAR，CAPx，PWM[TZ1,TRIP1]，FLB
INPUT2	Output X-BAR，PWM X-BAR，CAPx，PWM[TZ2,TRIP2] ，FLB
INPUT3	Output X-BAR，PWM X-BAR，CAPx，PWM[TZ3,TRIP3] ，FLB
INPUT4	Output X-BAR，PWM X-BAR，CAPx，FLB
INPUT5	Output X-BAR，PWM X-BAR，CAPx，ADCEXTSOC，EXTSYNCIN1，FLB
INPUT6	Output X-BAR，PWM X-BAR，CAPx，PWM[TRIP2]，PWM[TRIP6]，EXTSYNCIN2，FLB
INPUT7	PWM X-BAR，CAPx，FLB
INPUT8	PWM X-BAR，CAPx，FLB
INPUT9	PWM X-BAR，CAPx，FLB
INPUT10	PWM X-BAR，CAPx，FLB
INPUT11	PWM X-BAR，CAPx，FLB
INPUT12	PWM X-BAR，CAPx，FLB
INPUT13	PWM X-BAR，CAPx，FLB
INPUT14	PWM X-BAR，CAPx，FLB
INPUT15	CAPx
INPUT16	CAPx

23.3.2 Output X-BAR

23.3.2.1 结构框图

除输出锁存器外，Output X-BAR 的结构与 PWM X-BAR 的结构相似。

图 21 Output X-BAR 结构框图



23.3.2.2 信号传输

Output X-BAR 由 8 个 OUTPUT 构成，可以从设备内部接收信号，并路由到 GPIO 引脚。每个 OUTPUT 至少占用 GPIO 多路复用上的一个位置，由 OUTPUTXBARx 表示，可以选择单个信号，或者使用逻辑或连接的最多 32 个信号。

信号传输步骤：

（1） 选择需要传输到 GPIO 的信号：对于每个 OUTPUTXBARx，在 OUTPUTxMUX0TO15CFG 或 OUTPUTxMUX16TO31CFG 寄存器中选择一个信号作为每个多路复用器的输入（总共有 32 个多路复用器）。

（2） 使能多路复用器：在 OUTPUTxMUXENABLE 寄存器中使能的多路复用器。在传输到 GPIO 模块的 OUTPUTx 信号前，所有的 MUX 进行逻辑或运算。

（3） （可选）反相信号：在 OUTPUTINV 寄存器中对信号进行反相。

只有在 GPxMUX1/2 和 GPxGMUX1/2 寄存器中配置了正确的 OUTPUTx 复用选项时，GPIO 才可以识别到信号。

表格 74 Output X-BAR 复用配置

MUX	0	1	2	3
MUX0	COMP1.CTRIPOUTH	COMP1.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT1	CAP1OUT
MUX1	COMP1.CTRIPOUTL	INPUTXBAR1	FLB1_OUT12	ADCCEVT1
MUX2	COMP2.CTRIPOUTH	COMP2.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT2	CAP2OUT
MUX3	COMP2.CTRIPOUTL	INPUTXBAR2	FLB1_OUT13	ADCCEVT2
MUX4	COMP3.CTRIPOUTH	COMP3.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT3	CAP3OUT
MUX5	COMP3.CTRIPOUTL	INPUTXBAR3	FLB2_OUT12	ADCCEVT3
MUX6	COMP4.CTRIPOUTH	COMP4.CTRIPOUTH_OR_CTRIPOUTL	ADCAEVT4	CAP4OUT
MUX7	COMP4.CTRIPOUTL	INPUTXBAR4	FLB2_OUT13	ADCCEVT4
MUX8	COMP5.CTRIPOUTH	COMP5.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT1	CAP5OUT
MUX9	COMP5.CTRIPOUTL	INPUTXBAR5	FLB3_OUT12	保留
MUX10	COMP6.CTRIPOUTH	COMP6.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT2	CAP6OUT
MUX11	COMP6.CTRIPOUTL	INPUTXBAR6	FLB3_OUT13	保留
MUX12	COMP7.CTRIPOUTH	COMP7.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT3	CAP7OUT
MUX13	COMP7.CTRIPOUTL	ADCSOCAO	FLB4_OUT12	保留
MUX14	保留	保留	ADCBEVT4	EXTSYNCOUT
MUX15	保留	ADCSOCBO	FLB4_OUT13	保留
MUX16	SD1FLT1.COMPH	SD1FLT1.COMPH_OR_COMPL	保留	保留
MUX17	SD1FLT1.COMPL	保留	保留	保留

MUX	0	1	2	3
MUX18	SD1FLT2.COMPH	SD1FLT2.COMPH_OR_COMPL	保留	保留
MUX19	SD1FLT2.COMPL	保留	保留	保留
MUX20	SD1FLT3.COMPH	SD1FLT3.COMPH_OR_COMPL	保留	保留
MUX21	SD1FLT3.COMPL	保留	保留	保留
MUX22	SD1FLT4.COMPH	SD1FLT4.COMPH_OR_COMPL	保留	保留
MUX23	SD1FLT4.COMPL	保留	保留	保留

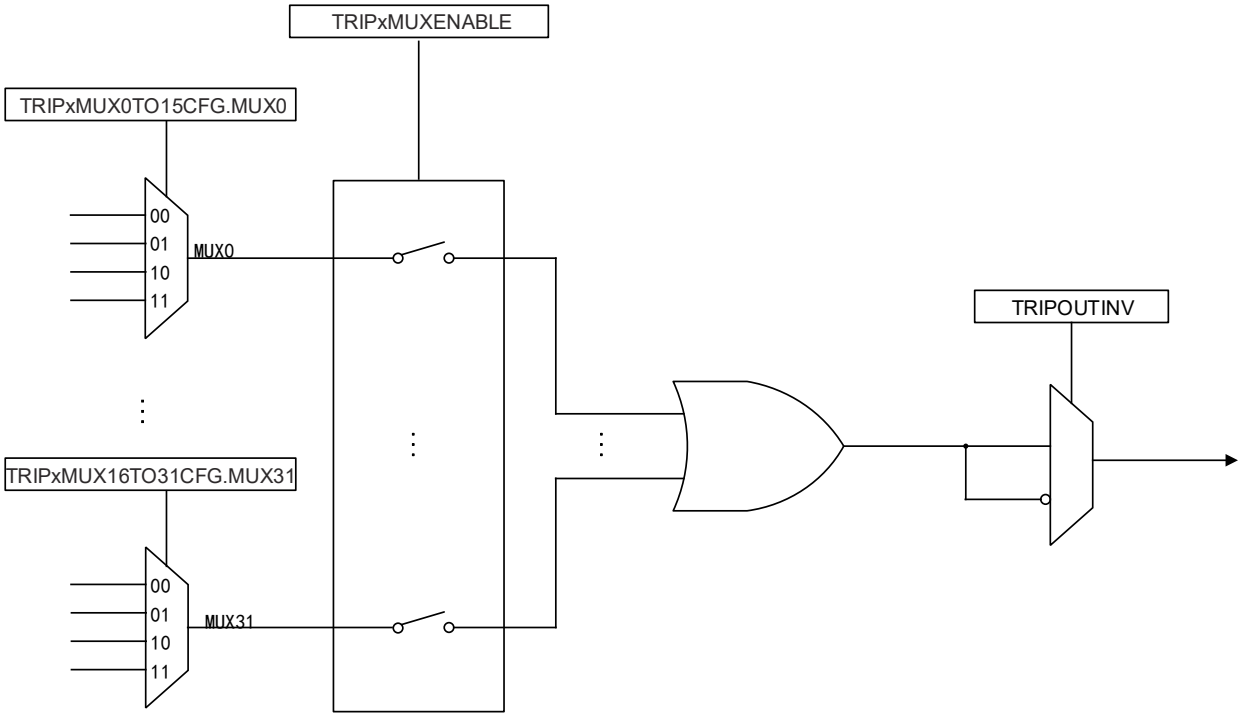
注意："保留"的信号不可使用，所有的未使用和保留的信号都设定为 0。

23.3.3 PWM X-BAR

23.3.3.1 结构框图

除输出锁存器外，Output X-BAR 的结构与 PWM X-BAR 的结构相似。

图 22 PWM X-BAR 结构框图



23.3.3.2 信号传输

PWM X-BAR 由 8 个输出构成，可以将信号路由到 PWM 模块，与 PWM 模块的数字比较（DC）子模块连接，执行跳闸和同步等操作。更多 DC 子模块的信息，详情请参见 PWM 章节。每个输出由 TRIPx 表示，可以在 MUX 上选择单个信号，或者使用逻辑或连接的最多 32 个信号。

信号传输步骤：

（1） 选择需要传输到 PWM 的信号：对于每个 TRIPx，在 TRIPxMUX0TO15CFG 或 TRIPxMUX16TO31CFG 寄存器中选择一个信号作为每个 MUX 的输入（总共有 32 个 MUX）。

（2） 使能多路复用器：在 TRIPxMUXENABLE 寄存器中使能多路复用器。在传输到 PWM 模块的 TRIPx 信号前，所有的多路复用器进行逻辑或运算。

（3） （可选）反相信号：在 TRIPOUTINV 寄存器中对信号进行反相。

表格 75 PWM X-BAR 复用配置

MUX	0	1	2	3
MUX0	COMP1.CTRIPH	COMP1.CTRIPH_OR_CTRIPL	ADCAEVT1	CAP1OUT
MUX1	COMP1.CTRIPL	INPUTXBAR1	FLB1_OUT12	ADCCEVT1
MUX2	COMP2.CTRIPH	COMP2.CTRIPH_OR_CTRIPL	ADCAEVT2	CAP2OUT
MUX3	COMP2.CTRIPL	INPUTXBAR2	FLB1_OUT13	ADCCEVT2
MUX4	COMP3.CTRIPH	COMP3.CTRIPH_OR_CTRIPL	ADCAEVT3	CAP3OUT
MUX5	COMP3.CTRIPL	INPUTXBAR3	FLB2_OUT12	ADCCEVT3
MUX6	COMP4.CTRIPH	COMP4.CTRIPH_OR_CTRIPL	ADCAEVT4	CAP4OUT
MUX7	COMP4.CTRIPL	INPUTXBAR4	FLB2_OUT13	ADCCEVT4
MUX8	COMP5.CTRIPH	COMP5.CTRIPH_OR_CTRIPL	ADCBEVT1	CAP5OUT
MUX9	COMP5.CTRIPL	INPUTXBAR5	FLB3_OUT12	保留
MUX10	COMP6.CTRIPH	COMP6.CTRIPH_OR_CTRIPL	ADCBEVT2	CAP6OUT
MUX11	COMP6.CTRIPL	INPUTXBAR6	FLB3_OUT13	保留
MUX12	COMP7.CTRIPH	COMP7.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT3	CAP7OUT
MUX13	COMP7.CTRIPL	ADCSOCAO	FLB4_OUT12	保留
MUX14	保留	保留	ADCBEVT4	EXTSYNCOUT
MUX15	保留	ADCSOCBO	FLB4_OUT13	保留
MUX16	SD1FLT1.COMPH	SD1FLT1.COMPH_OR_COMPL	保留	保留
MUX17	SD1FLT1.COMPL	INPUTXBAR7	保留	保留
MUX18	SD1FLT2.COMPH	SD1FLT2.COMPH_OR_COMPL	保留	保留
MUX19	SD1FLT2.COMPL	INPUTXBAR8	保留	保留
MUX20	SD1FLT3.COMPH	SD1FLT3.COMPH_OR_COMPL	保留	保留
MUX21	SD1FLT3.COMPL	INPUTXBAR9	保留	保留
MUX22	SD1FLT4.COMPH	SD1FLT4.COMPH_OR_COMPL	保留	保留
MUX23	SD1FLT4.COMPL	INPUTXBAR10	保留	保留
MUX24	保留	保留	保留	保留

MUX	0	1	2	3
MUX25	保留	INPUTXBAR11	保留	保留
MUX26	保留	保留	保留	保留
MUX27	保留	INPUTXBAR12	保留	保留
MUX28	保留	保留	保留	保留
MUX29	保留	INPUTXBAR13	保留	保留
MUX30	保留	保留	保留	保留
MUX31	保留	INPUTXBAR14	保留	保留

注意："保留"的信号不可使用，所有的未使用和保留的信号都设定为 0。

23.3.4 FLB X-BAR

23.3.4.1 结构框图

除输出锁存器外，Output X-BAR 的结构与 FLB X-BAR 的结构相似。

图 23 FLB X-BAR 结构框图

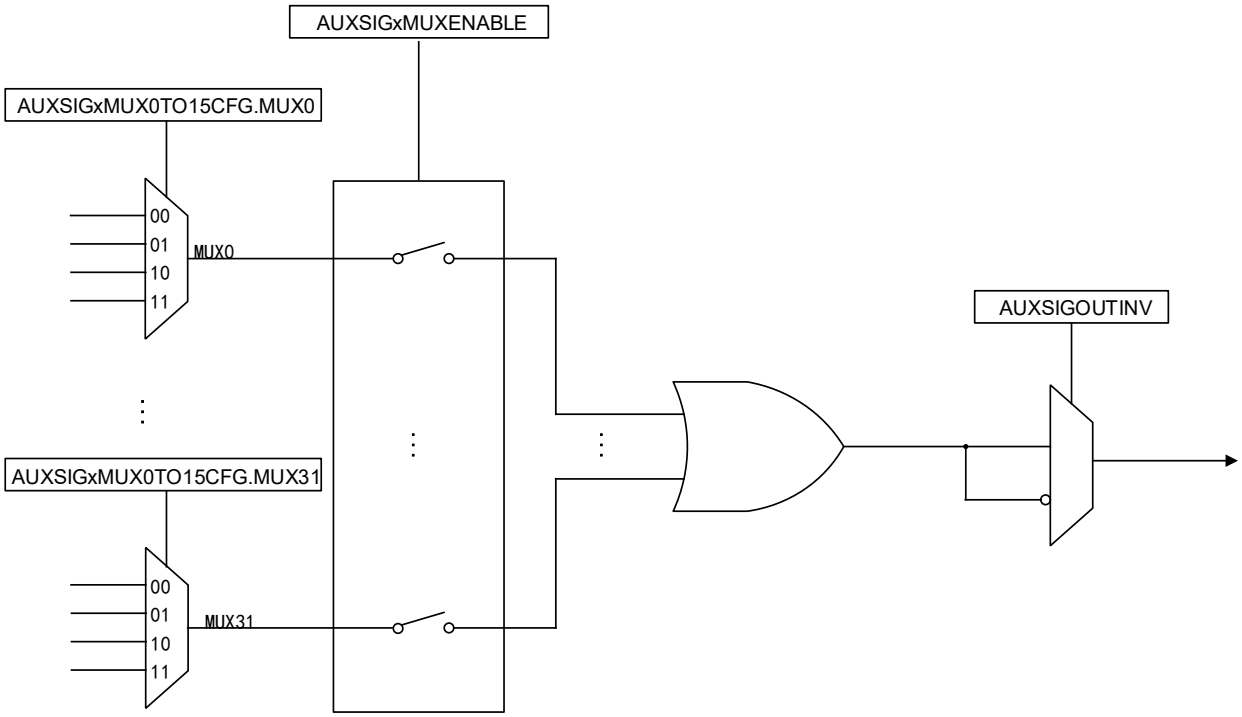
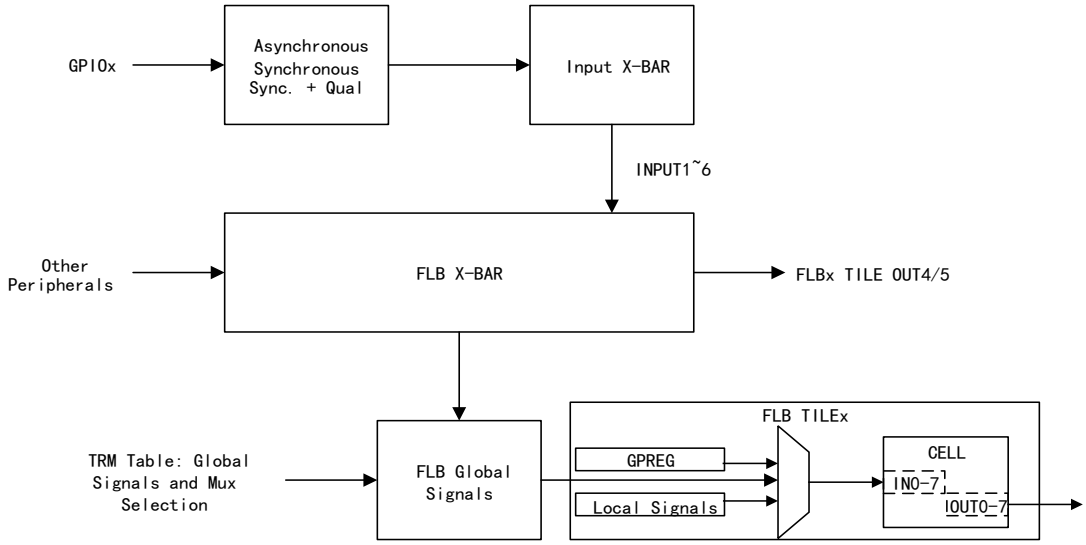


图 24 GPIO 到 FLB 的 TILE 连接



23.3.4.2 信号传输

FLB X-BAR 由 8 个输出构成，可以将信号路由到 FLB 模块。每个输出由 AUXSIGx 表示，可以在 MUX 上选择单个信号，或者使用逻辑或连接的最多 32 个信号。

信号传输步骤：

（1） 选择需要传输到 PWM 的信号：对于每个 AUXSIGx，在 AUXSIGxMUX0TO15CFG 或 AUXSIGxMUX16TO31CFG 选择一个信号作为每个 MUX 的输入（总共有 32 个 MUX）。

（2） 使能多路复用器：在 AUXSIGxMUXENABLE 寄存器中使能多路复用器。在传输到 PWM 模块的 AUXSIGx 信号前，所有的多路复用器进行逻辑或运算。

（3） （可选）反相信号：在 AUXSIGOUTINV 寄存器中对信号进行反相。

表格 76 FLB X-BAR 复用配置

MUX	0	1	2	3
MUX0	COMP1.CTRIPH	COMP1.CTRIPH_OR_CTRIPL	ADCAEVT1	CAP1OUT
MUX1	COMP1.CTRIPL	INPUTXBAR1	FLB1_OUT12	ADCCEVT1
MUX2	COMP2.CTRIPH	COMP2.CTRIPH_OR_CTRIPL	ADCAEVT2	CAP2OUT
MUX3	COMP2.CTRIPL	INPUTXBAR2	FLB1_OUT13	ADCCEVT2
MUX4	COMP3.CTRIPH	COMP3.CTRIPH_OR_CTRIPL	ADCAEVT3	CAP3OUT
MUX5	COMP3.CTRIPL	INPUTXBAR3	FLB2_OUT12	ADCCEVT3
MUX6	COMP4.CTRIPH	COMP4.CTRIPH_OR_CTRIPL	ADCAEVT4	CAP4OUT
MUX7	COMP4.CTRIPL	INPUTXBAR4	FLB2_OUT13	ADCCEVT4
MUX8	COMP5.CTRIPH	COMP5.CTRIPH_OR_CTRIPL	ADCBEVT1	CAP5OUT

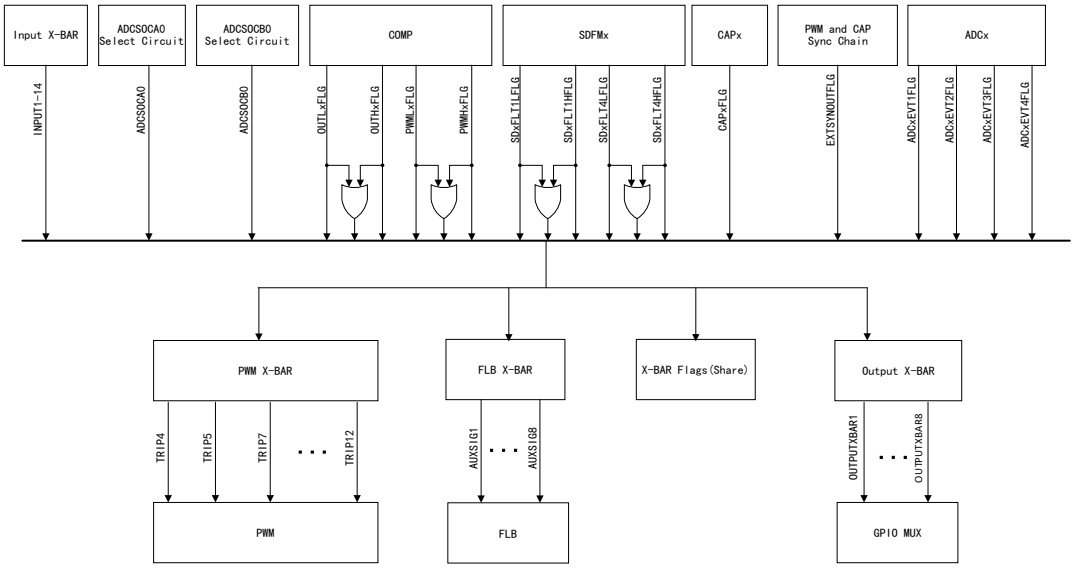
MUX	0	1	2	3
MUX9	COMP5.CTRIPL	INPUTXBAR5	FLB3_OUT12	保留
MUX10	COMP6.CTRIPH	COMP6.CTRIPH_OR_CTRIPL	ADCBEVT2	CAP6OUT
MUX11	COMP6.CTRIPL	INPUTXBAR6	FLB3_OUT13	保留
MUX12	COMP7.CTRIPH	COMP7.CTRIPOUTH_OR_CTRIPOUTL	ADCBEVT3	CAP7OUT
MUX13	COMP7.CTRIPL	ADCSOCAO	FLB4_OUT12	保留
MUX14	保留	保留	ADCBEVT4	EXTSYNCOUT
MUX15	保留	ADCSOCBO	FLB4_OUT13	保留
MUX16	SD1FLT1.COMPH	SD1FLT1.COMPH_OR_COMPL	SD1FLT1_COMPZ	SD1FLT1_DRINT
MUX17	SD1FLT1.COMPL	INPUTXBAR7	保留	保留
MUX18	SD1FLT2.COMPH	SD1FLT2.COMPH_OR_COMPL	SD1FLT2_COMPZ	SD1FLT2_DRINT
MUX19	SD1FLT2.COMPL	INPUTXBAR8	保留	保留
MUX20	SD1FLT3.COMPH	SD1FLT3.COMPH_OR_COMPL	SD1FLT3_COMPZ	SD1FLT3_DRINT
MUX21	SD1FLT3.COMPL	INPUTXBAR9	保留	保留
MUX22	SD1FLT4.COMPH	SD1FLT4.COMPH_OR_COMPL	SD1FLT4_COMPZ	SD1FLT4_DRINT
MUX23	SD1FLT4.COMPL	INPUTXBAR10	保留	保留
MUX24	保留	保留	保留	保留
MUX25	保留	INPUTXBAR11	保留	保留
MUX26	保留	保留	保留	保留
MUX27	保留	INPUTXBAR12	保留	保留
MUX28	保留	保留	保留	保留
MUX29	保留	INPUTXBAR13	保留	保留
MUX30	保留	保留	保留	保留
MUX31	保留	INPUTXBAR14	保留	保留

注意："保留"的信号不可使用，所有的未使用和保留的信号都设定为 0。

23.3.5 X-BAR 标志

Output X-BAR 和 PWM X-BAR 的输入信号相同（COMP 信号除外），因此 Output X-BAR，PWM X-BAR 和 FLB X-BAR 共用一组输入标志，用于指示那些输入信号已经被触发。在 XBAR_FLG 寄存器中，每个输入信号都有对应的标志位。当相应的信号触发后，标志位将保持置位状态，直到 XBAR_FLG 寄存器中的对应位进行清除此标志。软件可以在事件发生时检查对应的输入标志。

图 25 X-BAR 输入源



23.4 寄存器组地址

表格 77 XBAR 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
InputXbarRegs	XBAR_INPUT	0x4003_0C00	0x4003_0C7F
XbarRegs	XBAR_FLG	0x4003_1000	0x4003_103F
PWMXbarRegs	XBAR_PWM	0x4003_11C0	0x4003_123F
FlbXbarRegs	XBAR_FLB	0x4003_1240	0x4003_12BF
OutputXbarRegs	XBAR_OUTPUT	0x4003_12C0	0x4003_133F

23.5 寄存器地址映射

表格 78 XBAR_INPUT 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
INPUTxSELECT(n=1…16)	选择寄存器	0x00+(n-1)*0x02	√
INPUTSELECTLOCK	INPUTxSELECT 锁定寄存器	0x3C	√

表格 79 XBAR_FLG 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
XBARFLG1	标志寄存器 1	0x00	-
XBARFLG2	标志寄存器 2	0x04	-
XBARFLG3	标志寄存器 3	0x08	-
XBARFLG4	标志寄存器 4	0x0C	-

寄存器名称	寄存器描述	偏移地址	WRPRT
XBARCLR1	清除寄存器 1	0x10	-
XBARCLR2	清除寄存器 2	0x14	-
XBARCLR3	清除寄存器 3	0x18	-
XBARCLR4	清除寄存器 4	0x1C	-

表格 80 XBAR_PWM 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
TRIP4MUX0TO15CFG	TRIP4 MUX0_15 配置寄存器	0x00	√
TRIP4MUX16TO31CFG	TRIP4 MUX16_31 配置寄存器	0x04	√
TRIP5MUX0TO15CFG	TRIP5 MUX0_15 配置寄存器	0x08	√
TRIP5MUX16TO31CFG	TRIP5 MUX16_31 配置寄存器	0x0C	√
TRIP7MUX0TO15CFG	TRIP7 MUX0_15 配置寄存器	0x10	√
TRIP7MUX16TO31CFG	TRIP7 MUX16_31 配置寄存器	0x14	√
TRIP8MUX0TO15CFG	TRIP8 MUX0_15 配置寄存器	0x18	√
TRIP8MUX16TO31CFG	TRIP8 MUX16_31 配置寄存器	0x1C	√
TRIP9MUX0TO15CFG	TRIP9 MUX0_15 配置寄存器	0x20	√
TRIP9MUX16TO31CFG	TRIP9 MUX16_31 配置寄存器	0x24	√
TRIP10MUX0TO15CFG	TRIP10 MUX0_15 配置寄存器	0x28	√
TRIP10MUX16TO31CFG	TRIP10 MUX16_31 配置寄存器	0x2C	√
TRIP11MUX0TO15CFG	TRIP11 MUX0_15 配置寄存器	0x30	√
TRIP11MUX16TO31CFG	TRIP11 MUX16_31 配置寄存器	0x34	√
TRIP12MUX0TO15CFG	TRIP12 MUX0_15 配置寄存器	0x38	√
TRIP12MUX16TO31CFG	TRIP12 MUX16_31 配置寄存器	0x3C	√
TRIP4MUXENABLE	TRIP4 MUX 使能寄存器	0x40	√
TRIP5MUXENABLE	TRIP5 MUX 使能寄存器	0x44	√
TRIP7MUXENABLE	TRIP7 MUX 使能寄存器	0x48	√
TRIP8MUXENABLE	TRIP8 MUX 使能寄存器	0x4C	√
TRIP9MUXENABLE	TRIP9 MUX 使能寄存器	0x50	√
TRIP10MUXENABLE	TRIP10 MUX 使能寄存器	0x54	√
TRIP11MUXENABLE	TRIP11 MUX 使能寄存器	0x58	√
TRIP12MUXENABLE	TRIP12 MUX 使能寄存器	0x5C	√
TRIPOUTINV	TRIP 极性选择寄存器	0x70	√
TRIPLOCK	TRIP 锁定寄存器	0x7C	√

表格 81 XBAR_FLB 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
AUXSIG0MUX0TO15CFG	AUXSIG0 MUX0_15 配置寄存器	0x00	√
AUXSIG0MUX16TO31CFG	AUXSIG0 MUX16_31 配置寄存器	0x04	√
AUXSIG1MUX0TO15CFG	AUXSIG1 MUX0_15 配置寄存器	0x08	√
AUXSIG1MUX16TO31CFG	AUXSIG1 MUX16_31 配置寄存器	0x0C	√
AUXSIG2MUX0TO15CFG	AUXSIG2 MUX0_15 配置寄存器	0x10	√
AUXSIG2MUX16TO31CFG	AUXSIG2 MUX16_31 配置寄存器	0x14	√
AUXSIG3MUX0TO15CFG	AUXSIG3 MUX0_15 配置寄存器	0x18	√
AUXSIG3MUX16TO31CFG	AUXSIG3 MUX16_31 配置寄存器	0x1C	√
AUXSIG4MUX0TO15CFG	AUXSIG4 MUX0_15 配置寄存器	0x20	√
AUXSIG4MUX16TO31CFG	AUXSIG4 MUX16_31 配置寄存器	0x24	√
AUXSIG5MUX0TO15CFG	AUXSIG5 MUX0_15 配置寄存器	0x28	√
AUXSIG5MUX16TO31CFG	AUXSIG5 MUX16_31 配置寄存器	0x2C	√
AUXSIG6MUX0TO15CFG	AUXSIG6 MUX0_15 配置寄存器	0x30	√
AUXSIG6MUX16TO31CFG	AUXSIG6 MUX16_31 配置寄存器	0x34	√
AUXSIG7MUX0TO15CFG	AUXSIG7 MUX0_15 配置寄存器	0x38	√
AUXSIG7MUX16TO31CFG	AUXSIG7 MUX16_31 配置寄存器	0x3C	√
AUXSIG0MUXENABLE	AUXSIG0 MUX 使能寄存器	0x40	√
AUXSIG1MUXENABLE	AUXSIG1 MUX 使能寄存器	0x44	√
AUXSIG2MUXENABLE	AUXSIG2 MUX 使能寄存器	0x48	√
AUXSIG3MUXENABLE	AUXSIG3 MUX 使能寄存器	0x4C	√
AUXSIG4MUXENABLE	AUXSIG4 MUX 使能寄存器	0x50	√
AUXSIG5MUXENABLE	AUXSIG5 MUX 使能寄存器	0x54	√
AUXSIG6MUXENABLE	AUXSIG6 MUX 使能寄存器	0x58	√
AUXSIG7MUXENABLE	AUXSIG7 MUX 使能寄存器	0x5C	√
AUXSIGOUTINV	AUXSIG 极性选择寄存器	0x70	√
AUXSIGLOCK	AUXSIG 锁定寄存器	0x7C	√

表格 82 XBAR_OUTPUT 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
OUTPUTxMUX0TO15CFG（x=1…8）	OUTPUTx MUX0_15 配置寄存器	0x00+(x-1)*0x08	√
OUTPUTxMUX16TO31CFG（x=1…8）	OUTPUTx MUX16_31 配置寄存器	0x04+(x-1)*0x08	√

寄存器名称	寄存器描述	偏移地址	WRPRT
OUTPUTxMUXENABLE（x=1…8）	OUTPUTx MUX 使能寄存器	0x40+(x-1)*0x04	√
OUTPUTLATCH	OUTPUT 锁存寄存器	0x60	-
OUTPUTLATCHCLR	OUTPUT 锁存清除寄存器	0x64	-
OUTPUTLATCHFRC	OUTPUT 锁存设置器	0x68	-
OUTPUTLATCHENABLE	OUTPUT 锁存使能器	0x6C	√
OUTPUTINV	OUTPUT 锁存反相寄存器	0x70	√
OUTPUTLOCK	OUTPUT 锁定寄存器	0x7C	√

23.6 寄存器功能描述

23.6.1 选择寄存器（INPUTxSELECT）(n=1…16)

偏移地址：0x00+(n-1)*0x02

复位类型：CPU1.SYSRSn

位/域	名称	R/W	描述	复位值
15:0	SELECT	R/W	为 INPUTn 信号选择 GPIO（INPUTn signal Select GPIO）(n=1…16) 当 SELECT 的值大于可用的 GPIO 数量时，目标模块将被驱动为高电平(0xFFFF 除外)。 0x0000：GPIO0 0x0001：GPIO1 0x0002：GPIO2 … 0xFFFD：将目标模块驱动为高电平 0xFFFE：将目标模块驱动为高电平 0xFFFF：将目标模块驱动为低电平	FFFEh

23.6.2 XBAR_SEL 锁定寄存器（INPUTSELECTLOCK）

偏移地址：0x3C

复位类型：CPU1.SYSRSn

位/域	名称	R/W	描述	复位值
n-1	INPUTnSELECT	R/WSO	INPUTxSELECT 寄存器的锁定位（INPUTxSELECT Register Lock bit）(n=1…16) 置位后，只能通过 SYSRSn 寄存器清除。对于具有锁定保护的寄存器，始终允许读取操作。 0：INPUTxSELECT 寄存器未锁定 1：INPUTxSELECT 寄存器已锁定	0h
31:16	保留			0h

23.6.3 标志寄存器 1（XBARFLG1）

偏移地址：0x00

复位类型：CPU1.SYSRSn

此寄存器表示输入源的触发状态。

位/域	名称	R/W	描述	复位值
0	COMP1_CTRIPL	R	COMP1_PWML 输入标志 （COMP1_PWML input flag） 此位表示是否触发 COMP1_PWML 输入。 注意：设置此位的操作优先级高于软件清除操作。 0：未触发 1：已触发	0h
1	COMP1_CTRIPH	R	COMP1_PWMH 输入标志 （COMP1_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
2	COMP2_CTRIPL	R	COMP2_PWML 输入标志 （COMP2_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
3	COMP2_CTRIPH	R	COMP2_PWMH 输入标志 （COMP2_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
4	COMP3_CTRIPL	R	COMP3_PWML 输入标志 （COMP3_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
5	COMP3_CTRIPH	R	COMP3_PWMH 输入标志 （COMP3_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
6	COMP4_CTRIPL	R	COMP4_PWML 输入标志 （COMP4_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
7	COMP4_CTRIPH	R	COMP4_PWMH 输入标志 （COMP4_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
8	COMP5_CTRIPL	R	COMP5_PWML 输入标志 （COMP5_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
9	COMP5_CTRIPH	R	COMP5_PWMH 输入标志 （COMP5_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
10	COMP6_CTRIPL	R	COMP6_PWML 输入标志 （COMP6_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
11	COMP6_CTRIPH	R	COMP6_PWMH 输入标志 （COMP6_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h
12	COMP7_CTRIPL	R	COMP7_PWML 输入标志 （COMP7_PWML input flag） 具体描述参考 COMP1_CTRIPL。	0h
13	COMP7_CTRIPH	R	COMP7_PWMH 输入标志 （COMP7_PWMH input flag） 具体描述参考 COMP1_CTRIPL。	0h

位/域	名称	R/W	描述	复位值
15:14	保留			0h
16	COMP1_CTRIPOUTL	R	COMP1_OUTL 输入标志 （COMP1_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
17	COMP1_CTRIPOUTH	R	COMP1_OUTH 输入标志 （COMP1_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
18	COMP2_CTRIPOUYL	R	COMP2_OUTL 输入标志 （COMP2_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
19	COMP2_CTRIPOUTH	R	COMP2_OUTH 输入标志 （COMP2_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
20	COMP3_CTRIPOUTL	R	COMP3_OUTL 输入标志 （COMP3_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
21	COMP3_CTRIPOUTH	R	COMP3_OUTH 输入标志 （COMP3_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
22	“COMP4_CTRIPOUTL	R	COMP4_OUTL 输入标志 （COMP4_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
23	COMP4_CTRIPOUTH	R	COMP4_OUTH 输入标志 （COMP4_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
24	COMP5_CTRIPOUTL	R	COMP2_OUTL 输入标志 （COMP5_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
25	COMP5_CTRIPOUTH	R	COMP5_OUTH 输入标志 （COMP5_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
26	COMP6_CTRIPOUTL	R	COMP6_OUTL 输入标志 （COMP6_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
27	COMP6_CTRIPOUTH	R	COMP6_OUTH 输入标志 （COMP6_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
28	COMP7_CTRIPOUTL	R	COMP7_OUTL 输入标志 （COMP7_OUTL input flag） 具体描述参考 COMP1_CTRIPL。	0h
29	COMP7_CTRIPOUTH	R	COMP7_OUTH 输入标志 （COMP7_OUTH input flag） 具体描述参考 COMP1_CTRIPL。	0h
31:30	保留			0h

23.6.4 标志寄存器 2（XBARFLG2）

偏移地址：0x04

复位类型：CPU1.SYSRSn

此寄存器表示输入源的触发状态。

位/域	名称	R/W	描述	复位值
0	INPUT1	R	INPUT1 输入标志（INPUT1 input flag） 此位表示是否触发 INPUT1 输入。 注意：设置此位的操作优先级高于软件清除操作。 0：未触发 1：已触发	0h
1	INPUT2	R	INPUT2 输入标志（INPUT2 input flag） 具体描述参考 INPUT1。	0h
2	INPUT3	R	INPUT3 输入标志（INPUT3 input flag） 具体描述参考 INPUT1。	0h
3	INPUT4	R	INPUT4 输入标志（INPUT4 input flag） 具体描述参考 INPUT1。	0h
4	INPUT5	R	INPUT5 输入标志（INPUT5 input flag） 具体描述参考 INPUT1。	0h
5	INPUT6	R	INPUT6 输入标志（INPUT6 input flag） 具体描述参考 INPUT1。	0h
6	ADCSOCA	R	ADCSOCA 输入标志（ADCSOCA input flag） 具体描述参考 INPUT1。	0h
7	ADCSOCB	R	ADCSOCB 输入标志（ADCSOCB input flag） 具体描述参考 INPUT1。	0h
8	INPUT7	R	INPUT7 输入标志（INPUT7 input flag） 具体描述参考 INPUT1。	0h
9	INPUT8	R	INPUT8 输入标志（INPUT8 input flag） 具体描述参考 INPUT1。	0h
10	INPUT9	R	INPUT9 输入标志（INPUT9 input flag） 具体描述参考 INPUT1。	0h
11	INPUT10	R	INPUT10 输入标志（INPUT10 input flag） 具体描述参考 INPUT1。	0h
12	INPUT11	R	INPUT11 输入标志（INPUT11 input flag） 具体描述参考 INPUT1。	0h
13	INPUT12	R	INPUT12 输入标志（INPUT12 input flag） 具体描述参考 INPUT1。	0h
14	INPUT13	R	INPUT13 输入标志（INPUT13 input flag） 具体描述参考 INPUT1。	0h
15	INPUT14	R	INPUT14 输入标志（INPUT14 input flag） 具体描述参考 INPUT1。	0h
16	CAP1_OUT	R	CAP1_OUT 输入标志（CAP1_OUT input flag） 具体描述参考 INPUT1。	0h
17	CAP2_OUT	R	CAP2_OUT 输入标志（CAP2_OUT input flag） 具体描述参考 INPUT1。	0h

位/域	名称	R/W	描述	复位值
18	CAP3_OUT	R	CAP3_OUT 输入标志（CAP3_OUT input flag） 具体描述参考 INPUT1。	0h
19	CAP4_OUT	R	CAP4_OUT 输入标志（CAP4_OUT input flag） 具体描述参考 INPUT1。	0h
20	CAP5_OUT	R	CAP5_OUT 输入标志（CAP5_OUT input flag） 具体描述参考 INPUT1。	0h
21	CAP6_OUT	R	CAP6_OUT 输入标志（CAP6_OUT input flag） 具体描述参考 INPUT1。	0h
22	EXTSYNCOUT	R	EXTSYNOUT 输入标志（EXTSYNOUT input flag） 具体描述参考 INPUT1。	0h
23	ADCAEVT1	R	ADCAEVT1 输入标志（ADCAEVT1 input flag） 具体描述参考 INPUT1。	0h
24	ADCAEVT2	R	ADCAEVT2 输入标志（ADCAEVT2 input flag） 具体描述参考 INPUT1。	0h
25	ADCAEVT3	R	ADCAEVT3 输入标志（ADCAEVT3 input flag） 具体描述参考 INPUT1。	0h
26	ADCAEVT4	R	ADCAEVT4 输入标志（ADCAEVT4 input flag） 具体描述参考 INPUT1。	0h
27	ADCBEVT1	R	ADCBEVT1 输入标志（ADCBEVT1 input flag） 具体描述参考 INPUT1。	0h
28	ADCBEVT2	R	ADCBEVT2 输入标志（ADCBEVT2 input flag） 具体描述参考 INPUT1。	0h
29	ADCBEVT3	R	ADCBEVT3 输入标志（ADCBEVT3 input flag） 具体描述参考 INPUT1。	0h
30	ADCBEVT4	R	ADCBEVT4 输入标志（ADCBEVT4 input flag） 具体描述参考 INPUT1。	0h
31	ADCCEVT1	R	ADCCEVT1 输入标志（ADCCEVT1 input flag） 具体描述参考 INPUT1。	0h

23.6.5 标志寄存器 3（XBARFLG3）

偏移地址：0x08

复位类型：CPU1.SYSRSn

此寄存器表示输入源的触发状态。

位/域	名称	R/W	描述	复位值
0	ADCCEVT2	R	ADCCEVT2 输入标志（ADCCEVT2 input flag） 此位表示是否触发 ADCCEVT2 输入。 注意：设置此位的操作优先级高于软件清除操作。 0：未触发 1：已触发	0h
1	ADCCEVT3	R	ADCCEVT3 输入标志（ADCCEVT3 input flag） 具体描述参考 ADCCEVT2。	0h

位/域	名称	R/W	描述	复位值
2	ADCCEVT4	R	ADCCEVT4 输入标志（ADCCEVT4 input flag） 具体描述参考 ADCCEVT2。	0h
6:3	保留			0h
7	SD1FLT1_COMPL	R	SD1FLT1_COMPL 输入标志 （SD1FLT1_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
8	SD1FLT1_COMPH	R	SD1FLT1_COMPH 输入标志 （SD1FLT1_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
9	SD1FLT2_COMPL	R	SD1FLT2_COMPL 输入标志 （SD1FLT2_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
10	SD1FLT2_COMPL	R	SD1FLT2_COMPH 输入标志 （SD1FLT2_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
11	SD1FLT3_COMPL	R	SD1FLT3_COMPL 输入标志 （SD1FLT3_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
12	SD1FLT3_COMPH	R	SD1FLT3_COMPH 输入标志 （SD1FLT3_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
13	SD1FLT4_COMPL	R	SD1FLT4_COMPL 输入标志 （SD1FLT4_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
14	SD1FLT4_COMPH	R	SD1FLT4_COMPH 输入标志 （SD1FLT4_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
15	SD2FLT1_COMPL	R	SD2FLT1_COMPL 输入标志 （SD2FLT1_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
16	SD2FLT1_COMPH	R	SD2FLT1_COMPH 输入标志 （SD2FLT1_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
17	SD2FLT2_COMPL	R	SD2FLT2_COMPL 输入标志 （SD2FLT2_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
18	SD2FLT2_COMPH	R	SD2FLT2_COMPH 输入标志 （SD2FLT2_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
19	SD2FLT3_COMPL“	R	SD2FLT3_COMPL 输入标志 （SD2FLT3_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
20	SD2FLT3_COMPH	R	SD2FLT3_COMPH 输入标志 （SD2FLT3_COMPH input flag） 具体描述参考 ADCCEVT2。	0h

位/域	名称	R/W	描述	复位值
21	SD2FLT4_COMPL	R	SD2FLT4_COMPL 输入标志 （SD2FLT4_COMPL input flag） 具体描述参考 ADCCEVT2。	0h
22	SD2FLT4_COMPH	R	SD2FLT4_COMPH 输入标志 （SD2FLT4_COMPH input flag） 具体描述参考 ADCCEVT2。	0h
23	CAP7_OUT	R	CAP7_OUT 输入标志（CAP7_OUT input flag） 具体描述参考 ADCCEVT2。	0h
24	SD1FLT1_COMPZ	R	SD1FLT1_COMPZ 输入标志 （SD1FLT1_COMPZ input flag） 具体描述参考 ADCCEVT2。	0h
25	SD1FLT1_DRINT	R	SD1FLT1_DRINT 输入标志（SD1FLT1_DRINT input flag） 具体描述参考 ADCCEVT2。	0h
26	SD1FLT2_COMPZ	R	SD1FLT2_COMPZ 输入标志 （SD1FLT2_COMPZ input flag） 具体描述参考 ADCCEVT2。	0h
27	SD1FLT2_DRINT	R	SD1FLT2_DRINT 输入标志（SD1FLT2_DRINT input flag） 具体描述参考 ADCCEVT2。	0h
28	SD1FLT3_COMPZ	R	SD1FLT3_COMPZ 输入标志 （SD1FLT3_COMPZ input flag） 具体描述参考 ADCCEVT2。	0h
29	SD1FLT3_DRINT	R	SD1FLT3_DRINT 输入标志（SD1FLT3_DRINT input flag） 具体描述参考 ADCCEVT2。	0h
30	SD1FLT4_COMPZ	R	SD1FLT4_COMPZ 输入标志 （SD1FLT4_COMPZ input flag） 具体描述参考 ADCCEVT2。	0h
31	SD1FLT4_DRINT	R	SD1FLT4_DRINT 输入标志（SD1FLT4_DRINT input flag） 具体描述参考 ADCCEVT2。	0h

23.6.6 标志寄存器 4（XBARFLG4）

偏移地址：0x0C

复位类型：CPU1.SYSRSn

此寄存器表示输入源的触发状态。

位/域	名称	R/W	描述	复位值
0	SD2FLT1_COMPZ	R	SD2FLT1_COMPZ 输入标志 （SD2FLT1_COMPZ input flag） 此位表示是否触发 SD2FLT1_COMPZ 输入。 注意：设置此位的操作优先级高于软件清除操作。 0：未触发 1：已触发	0h

位/域	名称	R/W	描述	复位值
1	SD2FLT1_DRINT	R	SD2FLT1_DRINT 输入标志 （SD2FLT1_DRINT input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
2	SD2FLT2_COMPZ	R	SD2FLT2_COMPZ 输入标志 （SD2FLT2_COMPZ input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
3	SD2FLT2_DRINT	R	SD2FLT2_DRINT 输入标志 （SD2FLT2_DRINT input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
4	SD2FLT3_COMPZ	R	SD2FLT3_COMPZ 输入标志 （SD2FLT3_COMPZ input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
5	SD2FLT3_DRINT	R	SD2FLT3_DRINT 输入标志 （SD2FLT3_DRINT input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
6	SD2FLT4_COMPZ	R	SD2FLT4_COMPZ 输入标志 （SD2FLT4_COMPZ input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
7	SD2FLT4_DRINT	R	SD2FLT4_DRINT 输入标志 （SD2FLT4_DRINT input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
8	EMAC_PPS0	R	EMAC_PPS0 输入标志（EMAC_PPS0 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
9	MCANA_FEVT0	R	MCANA_FEVT0 输入标志（MCANA_FEVT 0 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
10	MCANA_FEVT1	R	MCANA_FEVT1 输入标志（MCANA_FEVT 1 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
11	MCANA_FEVT2	R	MCANA_FEVT2 输入标志（MCANA_FEVT 2 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
12	FLB7_4_1	R	FLB7_4_1 输入标志（FLB7_4_1 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
13	FLB7_5_1	R	FLB7_5_1 输入标志（FLB7_5_1 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
14	FLB8_4_1	R	FLB8_4_1 输入标志（FLB8_4_1 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
15	FLB8_5_1	R	FLB8_5_1 输入标志（FLB8_5_1 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
16	FLB1_OUT4	R	FLB1_OUT4 输入标志（FLB1_OUT4 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
17	FLB1_OUT5	R	FLB1_OUT5 输入标志（FLB1_OUT5 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h

位/域	名称	R/W	描述	复位值
18	FLB2_OUT4	R	FLB2_OUT4PH 输入标志（FLB2_OUT4 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
19	FLB2_OUT5	R	FLB2_OUT5 输入标志（FLB2_OUT5 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
20	FLB3_OUT4	R	FLB3_OUT4 输入标志（FLB3_OUT4 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
21	FLB3_OUT5	R	FLB3_OUT5 输入标志（FLB3_OUT5 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
22	FLB4_OUT4	R	FLB4_OUT4 输入标志（FLB4_OUT4 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
23	FLB4_OUT5	R	FLB4_OUT5 输入标志（FLB4_OUT5 input flag） 具体描述参考 SD2FLT1_COMPZ。	0h
31:24	保留			0h

23.6.7 清除寄存器 1（XBARCLR1）

偏移地址：0x10

复位类型：CPU1.SYSRSn

此寄存器用于清除 XBARFLG1 寄存器中相应的标志位。

位/域	名称	R/W	描述	复位值
0	COMP1_CTRIPL	R-0/W1S	清除 COMP1_PWML 标志 （COMP1_PWML input flag clear） 清除 COMP1_PWML 输入标志位 COMP1_CTRIPL。 0：无影响 1：清除 COMP1_CTRIPL 标志位	0h
1	COMP1_CTRIPH	R-0/W1S	清除 COMP1_PWMH 标志 （COMP1_PWMH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
2	COMP2_CTRIPL	R-0/W1S	清除 COMP2_PWML 标志 （COMP2_PWML input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
3	COMP2_CTRIPH	R-0/W1S	清除 COMP2_PWMH 标志 （COMP2_PWMH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
4	COMP3_CTRIPL	R-0/W1S	清除 COMP3_PWML 标志 （COMP3_PWML input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
5	COMP3_CTRIPH	R-0/W1S	清除 COMP3_PWMH 标志 （COMP3_PWMH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
6	COMP4_CTRIPL	R-0/W1S	清除 COMP4_PWML 标志 （COMP4_PWML input flag clear） 具体描述参考 COMP1_CTRIPL。	0h

位/域	名称	R/W	描述	复位值
7	COMP4_CTRIPH	R-0/W1S	清除 COMP4_PWMH 标志（COMP4_PWMH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
8	COMP5_CTRIPL	R-0/W1S	清除 COMP5_PWML 标志（COMP5_PWML input flag clear）具体描述参考 COMP1_CTRIPL。	0h
9	COMP5_CTRIPH	R-0/W1S	清除 COMP5_PWMH 标志（COMP5_PWMH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
10	COMP6_CTRIPL	R-0/W1S	清除 COMP6_PWML 标志（COMP6_PWML input flag clear）具体描述参考 COMP1_CTRIPL。	0h
11	COMP6_CTRIPH	R-0/W1S	清除 COMP6_PWMH 标志（COMP6_PWMH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
12	COMP7_CTRIPL	R-0/W1S	清除 COMP7_PWML 标志（COMP7_PWML input flag clear）具体描述参考 COMP1_CTRIPL。	0h
13	COMP7_CTRIPH	R-0/W1S	清除 COMP7_PWMH 标志（COMP7_PWMH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
15:14	保留			0h
16	COMP1_CTRIPOUTL	R-0/W1S	清除 COMP1_OUTL 标志（COMP1_OUTL input flag clear）具体描述参考 COMP1_CTRIPL。	0h
17	COMP1_CTRIPOUTH	R-0/W1S	清除 COMP1_OUTH 标志（COMP1_OUTH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
18	COMP2_CTRIPOUTL	R-0/W1S	清除 COMP2_OUTL 标志（COMP2_OUTL input flag clear）具体描述参考 COMP1_CTRIPL。	0h
19	COMP2_CTRIPOUTH	R-0/W1S	清除 COMP2_OUTH 标志（COMP2_OUTH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
20	COMP3_CTRIPOUTL	R-0/W1S	清除 COMP3_OUTL 标志（COMP3_OUTL input flag clear）具体描述参考 COMP1_CTRIPL。	0h
21	COMP3_CTRIPOUTH	R-0/W1S	清除 COMP3_OUTH 标志（COMP3_OUTH input flag clear）具体描述参考 COMP1_CTRIPL。	0h
22	COMP4_CTRIPOUTL	R-0/W1S	清除 COMP4_OUTL 标志（COMP4_OUTL input flag clear）具体描述参考 COMP1_CTRIPL。	0h
23	COMP4_CTRIPOUTH	R-0/W1S	清除 COMP4_OUTH 标志（COMP4_OUTH input flag clear）具体描述参考 COMP1_CTRIPL。	0h

位/域	名称	R/W	描述	复位值
24	COMP5_CTRIPOUTL	R-0/W1S	清除 COMP2_OUTL 标志（COMP5_OUTL input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
25	COMP5_CTRIPOUTH	R-0/W1S	清除 COMP5_OUTH 标志（COMP5_OUTH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
26	COMP6_CTRIPOYTL	R-0/W1S	清除 COMP6_OUTL 标志（COMP6_OUTL input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
27	COMP6_CTRIPOUTH	R-0/W1S	清除 COMP6_OUTH 标志（COMP6_OUTH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
28	COMP7_CTRIPOUTL	R-0/W1S	清除 COMP7_OUTL 标志（COMP7_OUTL input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
29	COMP7_CTRIPOUTH	R-0/W1S	清除 COMP7_OUTH 标志（COMP7_OUTH input flag clear） 具体描述参考 COMP1_CTRIPL。	0h
31:30	保留			0h

23.6.8 清除寄存器 2（XBARCLR2）

偏移地址：0x14

复位类型：CPU1.SYSRSn

此寄存器用于清除 XBARFLG2 寄存器中相应的标志位。

位/域	名称	R/W	描述	复位值
0	INPUT1	R-0/W1S	清除 INPUT1 标志（INPUT1 input flag clear） 清除 INPUT1 输入标志位 INPUT1。 0：无影响 1：清除 INPUT1 标志位	0h
1	INPUT2	R-0/W1S	清除 INPUT2 标志（INPUT2 input flag clear） 具体描述参考 INPUT1。	0h
2	INPUT3	R-0/W1S	清除 INPUT3 标志（INPUT3 input flag clear） 具体描述参考 INPUT1。	0h
3	INPUT4	R-0/W1S	清除 INPUT4 标志（INPUT4 input flag clear） 具体描述参考 INPUT1。	0h
4	INPUT5	R-0/W1S	清除 INPUT5 标志（INPUT5 input flag clear） 具体描述参考 INPUT1。	0h
5	INPUT6	R-0/W1S	清除 INPUT6 标志（INPUT6 input flag clear） 具体描述参考 INPUT1。	0h

位/域	名称	R/W	描述	复位值
6	ADCSOCA	R-0/W1S	清除 ADCSOCA 标志（ADCSOCA input flag clear） 具体描述参考 INPUT1。	0h
7	ADCSOCB	R-0/W1S	清除 ADCSOCB 标志（ADCSOCB input flag clear） 具体描述参考 INPUT1。	0h
8	INPUT7	R-0/W1S	清除 INPUT7 标志（INPUT7 input flag clear） 具体描述参考 INPUT1。	0h
9	INPUT8	R-0/W1S	清除 INPUT8 标志（INPUT8 input flag clear） 具体描述参考 INPUT1。	0h
10	INPUT9	R-0/W1S	清除 INPUT9 标志（INPUT9 input flag clear） 具体描述参考 INPUT1。	0h
11	INPUT10	R-0/W1S	清除 INPUT10 标志（INPUT10 input flag clear） 具体描述参考 INPUT1。	0h
12	INPUT11	R-0/W1S	清除 INPUT11 标志（INPUT11 input flag clear） 具体描述参考 INPUT1。	0h
13	INPUT12	R-0/W1S	清除 INPUT12 标志（INPUT12 input flag clear） 具体描述参考 INPUT1。	0h
14	INPUT13	R-0/W1S	清除 INPUT13 标志（INPUT13 input flag clear） 具体描述参考 INPUT1。	0h
15	INPUT14	R-0/W1S	清除 INPUT14 标志（INPUT14 input flag clear） 具体描述参考 INPUT1。	0h
16	CAP1_OUT	R-0/W1S	清除 CAP1_OUT 标志（CAP1_OUT input flag clear） 具体描述参考 INPUT1。	0h
17	CAP2_OUT	R-0/W1S	清除 CAP2_OUT 标志（CAP2_OUT input flag clear） 具体描述参考 INPUT1。	0h
18	CAP3_OUT	R-0/W1S	清除 CAP3_OUT 标志（CAP3_OUT input flag clear） 具体描述参考 INPUT1。	0h
19	CAP4_OUT	R-0/W1S	清除 CAP4_OUT 标志（CAP4_OUT input flag clear） 具体描述参考 INPUT1。	0h
20	CAP5_OUT	R-0/W1S	清除 CAP5_OUT 标志（CAP5_OUT input flag clear） 具体描述参考 INPUT1。	0h

位/域	名称	R/W	描述	复位值
21	CAP6_OUT	R-0/W1S	清除 CAP6_OUT 标志（CAP6_OUT input flag clear） 具体描述参考 INPUT1。	0h
22	EXTSYNCOUT	R-0/W1S	清除 EXTSYNOUT 标志（EXTSYNOUT input flag clear） 具体描述参考 INPUT1。	0h
23	ADCAEVT1	R-0/W1S	清除 ADCAEVT1 标志（ADCAEVT1 input flag clear） 具体描述参考 INPUT1。	0h
24	ADCAEVT2	R-0/W1S	清除 ADCAEVT2 标志（ADCAEVT2 input flag clear） 具体描述参考 INPUT1。	0h
25	ADCAEVT3	R-0/W1S	清除 ADCAEVT3 标志（ADCAEVT3 input flag clear） 具体描述参考 INPUT1。	0h
26	ADCAEVT4	R-0/W1S	清除 ADCAEVT4 标志（ADCAEVT4 input flag clear） 具体描述参考 INPUT1。	0h
27	ADCBEVT1	R-0/W1S	清除 ADCBEVT1 标志（ADCBEVT1 input flag clear） 具体描述参考 INPUT1。	0h
28	ADCBEVT2	R-0/W1S	清除 ADCBEVT2 标志（ADCBEVT2 input flag clear） 具体描述参考 INPUT1。	0h
29	ADCBEVT3	R-0/W1S	清除 ADCBEVT3 标志（ADCBEVT3 input flag clear） 具体描述参考 INPUT1。	0h
30	ADCBEVT4	R-0/W1S	清除 ADCBEVT4 标志（ADCBEVT4 input flag clear） 具体描述参考 INPUT1。	0h
31	ADCCEVT1	R-0/W1S	清除 ADCCEVT1 标志（ADCCEVT1 input flag clear） 具体描述参考 INPUT1。	0h

23.6.9 清除寄存器 3（XBARCLR3）

偏移地址：0x18

复位类型：CPU1.SYSRSn

此寄存器用于清除 XBARFLG3 寄存器中相应的标志位。

位/域	名称	R/W	描述	复位值
0	ADCCEVT2	R-0/W1S	清除 ADCCEVT2 标志（ADCCEVT2 input flag clear） 清除 ADCCEVT2 输入标志位 ADCCEVT2。 0：无影响 1：清除 ADCCEVT2 标志位	0h
1	ADCCEVT3	R-0/W1S	清除 ADCCEVT3 标志（ADCCEVT3 input flag clear） 具体描述参考 ADCCEVT2。	0h

位/域	名称	R/W	描述	复位值
2	ADCCEVT4	R-0/W1S	清除 ADCCEVT4 标志（ADCCEVT4 input flag clear） 具体描述参考 ADCCEVT2。	0h
6:3	保留			0h
7	SD1FLT1_COMPL	R-0/W1S	清除 SD1FLT1_COMPL 标志（SD1FLT1_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
8	SD1FLT1_COMPH	R-0/W1S	清除 SD1FLT1_COMPH 标志（SD1FLT1_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
9	SD1FLT2_COMPL	R-0/W1S	清除 SD1FLT2_COMPL 标志（SD1FLT2_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
10	SD1FLT2_COMPH	R-0/W1S	清除 SD1FLT2_COMPH 标志（SD1FLT2_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
11	SD1FLT3_COMPL	R-0/W1S	清除 SD1FLT3_COMPL 标志（SD1FLT3_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
12	SD1FLT3_COMPH	R-0/W1S	清除 SD1FLT3_COMPH 标志（SD1FLT3_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
13	SD1FLT4_COMPL	R-0/W1S	清除 SD1FLT4_COMPL 标志（SD1FLT4_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
14	SD1FLT4_COMPH	R-0/W1S	清除 SD1FLT4_COMPH 标志（SD1FLT4_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
15	SD2FLT1_COMPL	R-0/W1S	清除 SD2FLT1_COMPL 标志（SD2FLT1_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
16	SD2FLT1_COMPH	R-0/W1S	清除 SD2FLT1_COMPH 标志（SD2FLT1_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
17	SD2FLT2_COMPL	R-0/W1S	清除 SD2FLT2_COMPL 标志（SD2FLT2_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
18	SD2FLT2_COMPH	R-0/W1S	清除 SD2FLT2_COMPH 标志（SD2FLT2_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
19	SD2FLT3_COMPL	R-0/W1S	清除 SD2FLT3_COMPL 标志（SD2FLT3_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
20	SD2FLT3_COMPH	R-0/W1S	清除 SD2FLT3_COMPH 标志（SD2FLT3_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h

位/域	名称	R/W	描述	复位值
21	SD2FLT4_COMPL	R-0/W1S	清除 SD2FLT4_COMPL 标志（SD2FLT4_COMPL input flag clear） 具体描述参考 ADCCEVT2。	0h
22	SD2FLT4_COMPH	R-0/W1S	清除 SD2FLT4_COMPH 标志（SD2FLT4_COMPH input flag clear） 具体描述参考 ADCCEVT2。	0h
23	CAP7_OUT	R-0/W1S	清除 CAP7_OUT 标志（CAP7_OUT input flag clear） 具体描述参考 ADCCEVT2。	0h
24	“SD1FLT1_COMPZ	R-0/W1S	清除 SD1FLT1_COMPZ 标志（SD1FLT1_COMPZ input flag clear） 具体描述参考 ADCCEVT2。	0h
25	SD1FLT1_DRINT	R-0/W1S	清除 SD1FLT1_DRINT 标志（SD1FLT1_DRINT input flag clear） 具体描述参考 ADCCEVT2。	0h
26	SD1FLT2_COMPZ	R-0/W1S	清除 SD1FLT2_COMPZ 标志（SD1FLT2_COMPZ input flag clear） 具体描述参考 ADCCEVT2。	0h
27	SD1FLT2_DRINT	R-0/W1S	清除 SD1FLT2_DRINT 标志（SD1FLT2_DRINT input flag clear） 具体描述参考 ADCCEVT2。	0h
28	SD1FLT3_COMPZ	R-0/W1S	清除 SD1FLT3_COMPZ 标志（SD1FLT3_COMPZ input flag clear） 具体描述参考 ADCCEVT2。	0h
29	SD1FLT3_DRINT	R-0/W1S	清除 SD1FLT3_DRINT 标志（SD1FLT3_DRINT input flag clear） 具体描述参考 ADCCEVT2。	0h
30	SD1FLT4_COMPZ	R-0/W1S	清除 SD1FLT4_COMPZ 标志（SD1FLT4_COMPZ input flag clear） 具体描述参考 ADCCEVT2。	0h
31	SD1FLT4_DRINT	R-0/W1S	清除 SD1FLT4_DRINT 标志（SD1FLT4_DRINT input flag clear） 具体描述参考 ADCCEVT2。	0h

23.6.10 清除寄存器 4（XBARCLR4）

偏移地址：0x1C

复位类型：CPU1.SYSRSn

此寄存器用于清除 XBARFLG4 寄存器中相应的标志位。

位/域	名称	R/W	描述	复位值
0	SD2FLT1_COMPZ	R-0/W1S	清除 SD2FLT1_COMPZ 标志（SD2FLT1_COMPZ input flag clear） 清除 SD2FLT1_COMPZ 输入标志位 SD2FLT1_COMPZ。 0：无影响 1：清除 SD2FLT1_COMPZ 标志位	0h

位/域	名称	R/W	描述	复位值
1	SD2FLT1_DRINT	R-0/W1S	清除 SD2FLT1_DRINT 标志（SD2FLT1_DRINT input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
2	SD2FLT2_COMPZ	R-0/W1S	清除 SD2FLT2_COMPZ 标志（SD2FLT2_COMPZ input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
3	SD2FLT2_DRINT	R-0/W1S	清除 SD2FLT2_DRINT 标志（SD2FLT2_DRINT input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
4	SD2FLT3_COMPZ	R-0/W1S	清除 SD2FLT3_COMPZ 标志（SD2FLT3_COMPZ input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
5	SD2FLT3_DRINT	R-0/W1S	清除 SD2FLT3_DRINT 标志（SD2FLT3_DRINT input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
6	SD2FLT4_COMPZ	R-0/W1S	清除 SD2FLT4_COMPZ 标志（SD2FLT4_COMPZ input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
7	SD2FLT4_DRINT	R-0/W1S	清除 SD2FLT4_DRINT 标志（SD2FLT4_DRINT input flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
11:8	保留			0h
12	FLB7_4_1	R-0/W1S	清除 FLB7_4_1 标志（FLB7_4_1 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
13	FLB7_5_1	R-0/W1S	清除 FLB7_5_1 标志（FLB7_5_1 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
14	FLB8_4_1	R-0/W1S	清除 FLB8_4_1 标志（FLB8_4_1 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
15	FLB8_5_1	R-0/W1S	清除 FLB8_5_1 标志（FLB8_5_1 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
16	FLB1_OUT4	R-0/W1S	清除 FLB1_OUT4 标志（FLB1_OUT4 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
17	FLB1_OUT5	R-0/W1S	清除 FLB1_OUT5 标志（FLB1_OUT5 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
18	FLB2_OUT4	R-0/W1S	清除 FLB2_OUT4 标志（FLB2_OUT4 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
19	FLB2_OUT5	R-0/W1S	清除 FLB2_OUT5 标志（FLB2_OUT5 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h

位/域	名称	R/W	描述	复位值
20	FLB3_OUT4	R-0/W1S	清除 FLB3_OUT 标志（FLB3_OUT4 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
21	FLB3_OUT5	R-0/W1S	清除 FLB3_OUT5 标志（FLB3_OUT5 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
22	FLB4_OUT4	R-0/W1S	清除 FLB4_OUT4 标志（FLB4_OUT4 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
23	FLB4_OUT5	R-0/W1S	清除 FLB4_OUT5 标志（FLB4_OUT5 flag clear） 具体描述参考 SD2FLT1_COMPZ。	0h
31:24	保留			0h

23.6.11 TRIPx MUX0_15 配置寄存器（TRIPxMUX0TO15CFG）（x=4...5,7...12）

偏移地址：0x00（TRIP4MUX0TO15CFG）
0x08（TRIP5MUX0TO15CFG）
0x10（TRIP7MUX0TO15CFG）
0x18（TRIP8MUX0TO15CFG）
0x20（TRIP9MUX0TO15CFG）
0x28（TRIP10MUX0TO15CFG）
0x30（TRIP11MUX0TO15CFG）
0x38（TRIP12MUX0TO15CFG）
复位类型：CPU1.SYSRSn
配置 XBAR_PWM TRIPx Muxy 的输入信号(y=0…15)。

位/域	名称	R/W	描述	复位值
2y+1:2y	MUXy	R/W	XBAR_PWM TRIPx Muxy 选择输入(XBAR_PWM TRIPx Muxy Select input) (y=0…15) 选择 XBAR_PWM TRIPx Muxy 的输入信号 00：选择 Muxy 的第 1 个输入信号 01：选择 Muxy 的第 2 个输入信号 10：选择 Muxy 的第 3 个输入信号 11：选择 Muxy 的第 4 个输入信号	0h

23.6.12 TRIPx MUX16_31 配置寄存器（TRIPxMUX16TO31CFG）（x=4...5,7...12）

偏移地址：0x04（TRIP4MUX16TO31CFG）
0x0C（TRIP5MUX16TO31CFG）
0x14（TRIP7MUX16TO31CFG）
0x1C（TRIP8MUX16TO31CFG）
0x24（TRIP9MUX16TO31CFG）
0x2C（TRIP10MUX16TO31CFG）
0x34（TRIP11MUX16TO31CFG）
0x3C（TRIP12MUX16TO31CFG）

复位类型：CPU1.SYSRSn

配置 XBAR_PWM TRIPx Muxy 的输入信号(y=16…31)。

位/域	名称	R/W	描述	复位值
1:0	MUX16	R/W	XBAR_PWM TRIPx Mux16 选择输入(XBAR_PWM TRIPx Mux16 Select input) 选择 XBAR_PWM TRIPx Mux16 的输入信号 00：选择 Mux16 的第 1 个输入信号 01：选择 Mux16 的第 2 个输入信号 10：选择 Mux16 的第 3 个输入信号 11：选择 Mux16 的第 4 个输入信号	0h
3:2	MUX17	R/W	XBAR_PWM TRIPx Mux17 选择输入(XBAR_PWM TRIPx Mux17 Select input) 具体描述参考 MUX16。	0h
5:4	MUX18	R/W	XBAR_PWM TRIPx Mux18 选择输入(XBAR_PWM TRIPx Mux18 Select input) 具体描述参考 MUX16。	0h
7:6	MUX19	R/W	XBAR_PWM TRIPx Mux19 选择输入(XBAR_PWM TRIPx Mux19 Select input) 具体描述参考 MUX16。	0h
9:8	MUX20	R/W	XBAR_PWM TRIPx Mux20 选择输入(XBAR_PWM TRIPx Mux20 Select input) 具体描述参考 MUX16。	0h
11:10	MUX21	R/W	XBAR_PWM TRIPx Mux21 选择输入(XBAR_PWM TRIPx Mux21 Select input) 具体描述参考 MUX16。	0h
13:12	MUX22	R/W	XBAR_PWM TRIPx Mux22 选择输入(XBAR_PWM TRIPx Mux22 Select input) 具体描述参考 MUX16。	0h
15:14	MUX23	R/W	XBAR_PWM TRIPx Mux23 选择输入(XBAR_PWM TRIPx Mux23 Select input) 具体描述参考 MUX16。	0h
17:16	MUX24	R/W	XBAR_PWM TRIPx Mux24 选择输入(XBAR_PWM TRIPx Mux24 Select input) 具体描述参考 MUX16。	0h
19:18	MUX25	R/W	XBAR_PWM TRIPx Mux25 选择输入(XBAR_PWM TRIPx Mux25 Select input) 具体描述参考 MUX16。	0h
21:20	MUX26	R/W	XBAR_PWM TRIPx Mux26 选择输入(XBAR_PWM TRIPx Mux26 Select input) 具体描述参考 MUX16。	0h
23:22	MUX27	R/W	XBAR_PWM TRIPx Mux27 选择输入(XBAR_PWM TRIPx Mux27 Select input) 具体描述参考 MUX16。	0h
25:24	MUX28	R/W	XBAR_PWM TRIPx Mux28 选择输入(XBAR_PWM TRIPx Mux28 Select input) 具体描述参考 MUX16。	0h

位/域	名称	R/W	描述	复位值
27:26	MUX29	R/W	XBAR_PWM TRIPx Mux29 选择输入(XBAR_PWM TRIPx Mux29 Select input) 具体描述参考 MUX16。	0h
29:28	MUX30	R/W	XBAR_PWM TRIPx Mux30 选择输入(XBAR_PWM TRIPx Mux30 Select input) 具体描述参考 MUX16。	0h
31:30	MUX31	R/W	XBAR_PWM TRIPx Mux31 选择输入(XBAR_PWM TRIPx Mux31 Select input) 具体描述参考 MUX16。	0h

23.6.13 TRIPx MUX 使能寄存器（TRIPxMUXENABLE）（x=4...5,7...12）

偏移地址：0x40（TRIP4MUXENABLE）

0x44（TRIP5MUXENABLE）

0x48（TRIP7MUXENABLE）

0x4C（TRIP8MUXENABLE）

0x50（TRIP9MUXENABLE）

0x54（TRIP10MUXENABLE）

0x58（TRIP11MUXENABLE）

0x5C（TRIP12MUXENABLE）

复位类型：CPU1.SYSRSn

使能 Muxy(y=0…31)输出，Muxy 输出可以用于驱动 PWM-XBAR TRIPx。

位/域	名称	R/W	描述	复位值
y	MUXy	R/W	使能 Muxy 输出(Muxy output Enable) (y=0…31) 使能 XBAR_PWM TRIPx Muxy 的输入信号 0：禁用，此时不能驱动 PWM-XBAR TRIPx 1：使能，此时可以驱动 PWM-XBAR TRIPx	0h

23.6.14 TRIP 极性选择寄存器（TRIPOUTINV）

偏移地址：0x70

复位类型：CPU1.SYSRSn

使能 Muxy(y=0…31)输出，Muxy 输出可以用于驱动 PWM-XBAR TRIPx。

位/域	名称	R/W	描述	复位值
0	TRIP4	R/W	选择 XBAR_PWM 的 TRIP4 极性(XBAR_PWM TRIP4 polarity Select) 0：驱动 TRIP4 输出为高电平有效 1：驱动 TRIP4 输出为低电平有效	0h
1	TRIP5	R/W	选择 XBAR_PWM 的 TRIP5 极性(XBAR_PWM TRIP5 polarity Select) 具体描述参考 TRIP4。	0h
2	TRIP7	R/W	选择 XBAR_PWM 的 TRIP7 极性(XBAR_PWM TRIP7 polarity Select) 具体描述参考 TRIP4。	0h
3	TRIP8	R/W	选择 XBAR_PWM 的 TRIP8 极性(XBAR_PWM TRIP8 polarity Select) 具体描述参考 TRIP4。	0h

位/域	名称	R/W	描述	复位值
4	TRIP9	R/W	选择 XBAR_PWM 的 TRIP9 极性(XBAR_PWM TRIP9 polarity Select) 具体描述参考 TRIP4。	0h
5	TRIP10	R/W	选择 XBAR_PWM 的 TRIP10 极性(XBAR_PWM TRIP10 polarity Select) 具体描述参考 TRIP4。	0h
6	TRIP11	R/W	选择 XBAR_PWM 的 TRIP11 极性(XBAR_PWM TRIP11 polarity Select) 具体描述参考 TRIP4。	0h
7	TRIP12	R/W	选择 XBAR_PWM 的 TRIP12 极性(XBAR_PWM TRIP12 polarity Select) 具体描述参考 TRIP4。	0h
31:8	保留			0h

23.6.15 TRIP 锁定寄存器（TRIPLOCK）

偏移地址：0x7C

复位类型：CPU1.SYSRSn

此寄存器用于阻止对 XBAR_PWM 寄存器组的写入操作，读取操作不受影响。

被锁定的寄存器包括：

- TRIPxMUX0TO15CFG（x=4...5,7...12）
- TRIPxMUX16TO31CFG（x=4...5,7...12）
- TRIPxMUXENABLE（x=4...5,7...12）
- TRIPOUTINV

位/域	名称	R/W	描述	复位值
0	LOCK	R/WSO	锁定 XBAR_PWM 的配置（XBAR_PWM configuration Lock） 当此位置 1 时，锁定 XBAR_PWM 寄存器组寄存器的配置，将禁止对这些寄存器进行写入操作。 0：允许对 XBAR_PWM 的寄存器进行写入操作 1：禁止对 XBAR_PWM 的寄存器进行写入操作	0h
15:1	保留			0h
31:16	KEY	R-0/W	关键字（key） 此位域决定 LOCK 位是否被激活。只有当 KEY=0x5A5A 时，才可以设置 LOCK 位。	0h

23.6.16 AUXSIGx MUX0_15 配置寄存器（AUXSIGxMUX0TO15CFG）（x=4...5,7...12）

偏移地址：0x00（AUXSIG0MUX0TO15CFG）

0x08（AUXSIG1MUX0TO15CFG）

0x10（AUXSIG2MUX0TO15CFG）

0x18（AUXSIG3MUX0TO15CFG）

0x20（AUXSIG4MUX0TO15CFG）

0x28（AUXSIG5MUX0TO15CFG）
0x30（AUXSIG6MUX0TO15CFG）
0x38（AUXSIG7MUX0TO15CFG）
复位类型：CPU1.SYSRSn
配置 XBAR_FLB AUXSIGx Muxy 的输入信号(y=0…15)。

位/域	名称	R/W	描述	复位值
2y+1:2y	MUXy	R/W	XBAR_FLB AUXSIGx Muxy 选择输入(XBAR_FLB AUXSIGx Muxy Select input) (y=0…15) 选择 XBAR_FLB AUXSIGx Muxy 的输入信号 00：选择 Muxy 的第 1 个输入信号 01：选择 Muxy 的第 2 个输入信号 10：选择 Muxy 的第 3 个输入信号 11：选择 Muxy 的第 4 个输入信号	0h

23.6.17 AUXSIGx MUX16_31 配置寄存器（AUXSIGxMUX16TO31CFG）（x=4...5,7...12）

偏移地址：0x04（AUXSIG0MUX16TO31CFG）
0x0C（AUXSIG1MUX16TO31CFG）
0x14（AUXSIG2MUX16TO31CFG）
0x1C（AUXSIG3MUX16TO31CFG）
0x24（AUXSIG4MUX16TO31CFG）
0x2C（AUXSIG5MUX16TO31CFG）
0x34（AUXSIG6MUX16TO31CFG）
0x3C（AUXSIG7MUX16TO31CFG）
复位类型：CPU1.SYSRSn
配置 XBAR_FLB AUXSIGx Muxy 的输入信号(y=16…31)。

位/域	名称	R/W	描述	复位值
1:0	MUX16	R/W	XBAR_FLB AUXSIGx Mux16 选择输入(XBAR_FLB AUXSIGx Mux16 Select input) 选择 XBAR_FLB AUXSIGx Mux16 的输入信号 00：选择 Mux16 的第 1 个输入信号 01：选择 Mux16 的第 2 个输入信号 10：选择 Mux16 的第 3 个输入信号 11：选择 Mux16 的第 4 个输入信号	0h
3:2	MUX17	R/W	XBAR_FLB AUXSIGx Mux17 选择输入(XBAR_FLB AUXSIGx Mux17 Select input) 具体描述参考 MUX16。	0h
5:4	MUX18	R/W	XBAR_FLB AUXSIGx Mux18 选择输入(XBAR_FLB AUXSIGx Mux18 Select input) 具体描述参考 MUX16。	0h
7:6	MUX19	R/W	XBAR_FLB AUXSIGx Mux19 选择输入(XBAR_FLB AUXSIGx Mux19 Select input) 具体描述参考 MUX16。	0h
9:8	MUX20	R/W	XBAR_FLB AUXSIGx Mux20 选择输入(XBAR_FLB AUXSIGx Mux20 Select input) 具体描述参考 MUX16。	0h

位/域	名称	R/W	描述	复位值
11:10	MUX21	R/W	XBAR_FLB AUXSIGx Mux21 选择输入(XBAR_FLB AUXSIGx Mux21 Select input) 具体描述参考 MUX16。	0h
13:12	MUX22	R/W	XBAR_FLB AUXSIGx Mux22 选择输入(XBAR_FLB AUXSIGx Mux22 Select input) 具体描述参考 MUX16。	0h
15:14	MUX23	R/W	XBAR_FLB AUXSIGx Mux23 选择输入(XBAR_FLB AUXSIGx Mux23 Select input) 具体描述参考 MUX16。	0h
17:16	MUX24	R/W	XBAR_FLB AUXSIGx Mux24 选择输入(XBAR_FLB AUXSIGx Mux24 Select input) 具体描述参考 MUX16。	0h
19:18	MUX25	R/W	XBAR_FLB AUXSIGx Mux25 选择输入(XBAR_FLB AUXSIGx Mux25 Select input) 具体描述参考 MUX16。	0h
21:20	MUX26	R/W	XBAR_FLB AUXSIGx Mux26 选择输入(XBAR_FLB AUXSIGx Mux26 Select input) 具体描述参考 MUX16。	0h
23:22	MUX27	R/W	XBAR_FLB AUXSIGx Mux27 选择输入(XBAR_FLB AUXSIGx Mux27 Select input) 具体描述参考 MUX16。	0h
25:24	MUX28	R/W	XBAR_FLB AUXSIGx Mux28 选择输入(XBAR_FLB AUXSIGx Mux28 Select input) 具体描述参考 MUX16。	0h
27:26	MUX29	R/W	XBAR_FLB AUXSIGx Mux29 选择输入(XBAR_FLB AUXSIGx Mux29 Select input) 具体描述参考 MUX16。	0h
29:28	MUX30	R/W	XBAR_FLB AUXSIGx Mux30 选择输入(XBAR_FLB AUXSIGx Mux30 Select input) 具体描述参考 MUX16。	0h
31:30	MUX31	R/W	XBAR_FLB AUXSIGx Mux31 选择输入(XBAR_FLB AUXSIGx Mux31 Select input) 具体描述参考 MUX16。	0h

23.6.18 AUXSIGx MUX 使能寄存器（AUXSIGxMUXENABLE）（x=4...5,7...12）

偏移地址：0x40（AUXSIG0MUXENABLE）

0x44（AUXSIG1MUXENABLE）

0x48（AUXSIG2MUXENABLE）

0x4C（AUXSIG3MUXENABLE）

0x50（AUXSIG4MUXENABLE）

0x54（AUXSIG5MUXENABLE）

0x58（AUXSIG6MUXENABLE）

0x5C（AUXSIG7MUXENABLE）

复位类型：CPU1.SYSRSn

使能 Muxy(y=0…31)输出，Muxy 输出可以用于驱动 FLB-XBAR AUXSIGx。

位/域	名称	R/W	描述	复位值
y	MUXy	R/W	使能 Muxy 输出(Muxy output Enable) (y=0…31) 使能 XBAR_FLB AUXSIGx Muxy 的输入信号 0：禁用，此时不能驱动 FLB-XBAR AUXSIGx 1：使能，此时可以驱动 FLB-XBAR AUXSIGx	0h

23.6.19 AUXSIG 极性选择寄存器（AUXSIGOUTINV）

偏移地址：0x70

复位类型：CPU1.SYSRSn

使能 Muxy(y=0…31)输出，Muxy 输出可以用于驱动 FLB-XBAR AUXSIGx。

位/域	名称	R/W	描述	复位值
0	AUXSIG0	R/W	选择 XBAR_FLB 的 AUXSIG0 极性(XBAR_FLB AUXSIG0 polarity Select) 0：驱动 AUXSIG0 输出为高电平有效 1：驱动 AUXSIG0 输出为低电平有效	0h
1	AUXSIG1	R/W	选择 XBAR_FLB 的 AUXSIG1 极性(XBAR_FLB AUXSIG1 polarity Select) 具体描述参考 AUXSIG0。	0h
2	AUXSIG2	R/W	选择 XBAR_FLB 的 AUXSIG2 极性(XBAR_FLB AUXSIG2 polarity Select) 具体描述参考 AUXSIG0。	0h
3	AUXSIG3	R/W	选择 XBAR_FLB 的 AUXSIG3 极性(XBAR_FLB AUXSIG3 polarity Select) 具体描述参考 AUXSIG0。	0h
4	AUXSIG4	R/W	选择 XBAR_FLB 的 AUXSIG4 极性(XBAR_FLB AUXSIG4 polarity Select) 具体描述参考 AUXSIG0。	0h
5	AUXSIG5	R/W	选择 XBAR_FLB 的 AUXSIG5 极性(XBAR_FLB AUXSIG5 polarity Select) 具体描述参考 AUXSIG0。	0h
6	AUXSIG6	R/W	选择 XBAR_FLB 的 AUXSIG6 极性(XBAR_FLB AUXSIG6 polarity Select) 具体描述参考 AUXSIG0。	0h
7	AUXSIG7	R/W	选择 XBAR_FLB 的 AUXSIG7 极性(XBAR_FLB AUXSIG7 polarity Select) 具体描述参考 AUXSIG0。	0h
31:8	保留			0h

23.6.20 AUXSIG 锁定寄存器（AUXSIGLOCK）

偏移地址：0x7C

复位类型：CPU1.SYSRSn

此寄存器用于阻止对 XBAR_FLB 寄存器组的写入操作，读取操作不受影响。

被锁定的寄存器包括：

- AUXSIGxMUX0TO15CFG（x=0…7）
- AUXSIGxMUX16TO31CFG（x=0…7）
- AUXSIGxMUXENABLE（x=0…7）
- AUXSIGOUTINV

位/域	名称	R/W	描述	复位值
0	LOCK	R/WSO	锁定 XBAR_FLB 的配置（XBAR_FLB configuration Lock） 当此位置 1 时，锁定 XBAR_FLB 寄存器组寄存器的配置，将禁止对这些寄存器进行写入操作。 0：允许对 XBAR_FLB 的寄存器进行写入操作 1：禁止对 XBAR_FLB 的寄存器进行写入操作	0h
15:1	保留			0h
31:16	KEY	R-0/W	关键字（key） 此位域决定 LOCK 位是否被激活。只有当 KEY=0x5A5A 时，才可以设置 LOCK 位。	0h

23.6.21 OUTPUTx MUX0_15 配置寄存器（OUTPUTxMUX0TO15CFG）（x=1...8）

偏移地址：0x00+(x-1)*0x08

复位类型：CPU1.SYSRSn

配置 XBAR_OUTPUTx Muxy 的输入信号(y=0…15)。

位/域	名称	R/W	描述	复位值
2y+1:2y	MUXy	R/W	OUTPUTx MUX0 选择输入(OUTPUTx Muxy Select input) (y=0…15) 选择 OUTPUTx Muxy 的输入信号 00：选择 Muxy 的第 1 个输入信号 01：选择 Muxy 的第 2 个输入信号 10：选择 Muxy 的第 3 个输入信号 11：选择 Muxy 的第 4 个输入信号	0h

23.6.22 OUTPUTx MUX16_31 配置寄存器（OUTPUTxMUX16TO31CFG）（x=1...8）

偏移地址：0x04+(x-1)*0x08

复位类型：CPU1.SYSRSn

配置 XBAR_ OUTPUTx Muxy 的输入信号(y=16…31)。

位/域	名称	R/W	描述	复位值
1:0	MUX16	R/W	OUTPUTx Mux16 选择输入(OUTPUTx Mux16 Select input) 选择 OUTPUTx Mux16 的输入信号 00：选择 Mux16 的第 1 个输入信号 01：选择 Mux16 的第 2 个输入信号 10：选择 Mux16 的第 3 个输入信号 11：选择 Mux16 的第 4 个输入信号	0h
3:2	MUX17	R/W	OUTPUTx Mux17 选择输入(OUTPUTx Mux17 Select input) 具体描述参考 MUX16。	0h
5:4	MUX18	R/W	OUTPUTx Mux18 选择输入(OUTPUTx Mux18 Select input) 具体描述参考 MUX16。	0h

位/域	名称	R/W	描述	复位值
7:6	MUX19	R/W	OUTPUTx Mux19 选择输入(OUTPUTx Mux19 Select input) 具体描述参考 MUX16。	0h
9:8	MUX20	R/W	OUTPUTx Mux20 选择输入(OUTPUTx Mux20 Select input) 具体描述参考 MUX16。	0h
11:10	MUX21	R/W	OUTPUTx Mux21 选择输入(OUTPUTx Mux21 Select input) 具体描述参考 MUX16。	0h
13:12	MUX22	R/W	OUTPUTx Mux22 选择输入(OUTPUTx Mux22 Select input) 具体描述参考 MUX16。	0h
15:14	MUX23	R/W	OUTPUTx Mux23 选择输入(OUTPUTx Mux23 Select input) 具体描述参考 MUX16。	0h
17:16	MUX24	R/W	OUTPUTx Mux24 选择输入(OUTPUTx Mux24 Select input) 具体描述参考 MUX16。	0h
19:18	MUX25	R/W	OUTPUTx Mux25 选择输入(OUTPUTx Mux25 Select input) 具体描述参考 MUX16。	0h
21:20	MUX26	R/W	OUTPUTx Mux26 选择输入(OUTPUTx Mux26 Select input) 具体描述参考 MUX16。	0h
23:22	MUX27	R/W	OUTPUTx Mux27 选择输入(OUTPUTx Mux27 Select input) 具体描述参考 MUX16。	0h
25:24	MUX28	R/W	OUTPUTx Mux28 选择输入(OUTPUTx Mux28 Select input) 具体描述参考 MUX16。	0h
27:26	MUX29	R/W	OUTPUTx Mux29 选择输入(OUTPUTx Mux29 Select input) 具体描述参考 MUX16。	0h
29:28	MUX30	R/W	OUTPUTx Mux30 选择输入(OUTPUTx Mux30 Select input) 具体描述参考 MUX16。	0h
31:30	MUX31	R/W	OUTPUTx Mux31 选择输入(OUTPUTx Mux31 Select input) 具体描述参考 MUX16。	0h

23.6.23 OUTPUTx MUX 使能寄存器（OUTPUTxMUXENABLE）（x=1…8）

偏移地址：0x40+(x-1)*0x08

复位类型：CPU1.SYSRSn

使能 Muxy(y=0…31)输出，Muxy 输出可以用于驱动 PWM-XBAR OUTPUTx。

位/域	名称	R/W	描述	复位值
y	MUXy	R/W	使能 Muxy 输出(Muxy output Enable) (y=0…31) 使能 OUTPUTx Muxy 的输入信号 0：禁用，此时不能驱动输出 XBAR 的 OUTPUTx 1：使能，此时可以驱动输出 XBAR 的 OUTPUTx	0h

23.6.24 OUTPUT 锁存寄存器（OUTPUTLATCH）

偏移地址：0x60

复位类型：CPU1.SYSRSn

此寄存器表示 OUTPUTx 的触发状态。

位/域	名称	R/W	描述	复位值
0	OUTPUT1	R	OUTPUT1 锁存（OUTPUT1 Latch） 此位表示是否触发 Output X-BAR 的 OUTPUT1 锁存。 注意：设置此位的操作优先级高于软件清除操作。 0：未触发 1：已触发	0h
1	OUTPUT2	R	OUTPUT2 锁存（OUTPUT2 Latch） 具体描述参考 OUTPUT1。	0h
2	OUTPUT3	R	OUTPUT3 锁存（OUTPUT3 Latch） 具体描述参考 OUTPUT1。	0h
3	OUTPUT4	R	OUTPUT4 锁存（OUTPUT4 Latch） 具体描述参考 OUTPUT1。	0h
4	OUTPUT5	R	OUTPUT5 锁存（OUTPUT5 Latch） 具体描述参考 OUTPUT1。	0h
5	OUTPUT6	R	OUTPUT6 锁存（OUTPUT6 Latch） 具体描述参考 OUTPUT1。	0h
6	OUTPUT7	R	OUTPUT7 锁存（OUTPUT7 Latch） 具体描述参考 OUTPUT1。	0h
7	OUTPUT8	R	OUTPUT8 锁存（OUTPUT8 Latch） 具体描述参考 OUTPUT1。	0h
31:8	保留			0h

23.6.25 OUTPUT 锁存清除寄存器（OUTPUTLATCHCLR）

偏移地址：0x60

复位类型：CPU1.SYSRSn

此寄存器用于清除 OUTPUT 锁存寄存器（OUTPUTLATCH）中的对应位。

位/域	名称	R/W	描述	复位值
0	OUTPUT1	W	清除 XBAR_OUTPUT OUTPUT1 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT1） 清除 OUTPUT1 位。 0：无影响 1：清除 OUTPUT1 位	0h

位/域	名称	R/W	描述	复位值
1	OUTPUT2	W	清除 XBAR_OUTPUT OUTPUT2 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT2） 具体描述参考 OUTPUT1。	0h
2	OUTPUT3	W	清除 XBAR_OUTPUT OUTPUT3 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT3） 具体描述参考 OUTPUT1。	0h
3	OUTPUT4	W	清除 XBAR_OUTPUT OUTPUT4 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT4） 具体描述参考 OUTPUT1。	0h
4	OUTPUT5	W	清除 XBAR_OUTPUT OUTPUT5 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT5） 具体描述参考 OUTPUT1。	0h
5	OUTPUT6	W	清除 XBAR_OUTPUT OUTPUT6 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT6） 具体描述参考 OUTPUT1。	0h
6	OUTPUT7	W	清除 XBAR_OUTPUT OUTPUT7 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT7） 具体描述参考 OUTPUT1。	0h
7	OUTPUT8	W	清除 XBAR_OUTPUT OUTPUT8 的 OUTPUTLATCH 的位（clear the OUTPUTLATCH for XBAR_OUTPUT OUTPUT8） 具体描述参考 OUTPUT1。	0h
31:8	保留			0h

23.6.26 OUTPUT 锁存设置寄存器（OUTPUTLATCHFRC）

偏移地址：0x60

复位类型：CPU1.SYSRSn

此寄存器用于设置 OUTPUT 锁存寄存器（OUTPUTLATCH）中的对应位。

位/域	名称	R/W	描述	复位值
0	OUTPUT1	W	设置 XBAR_OUTPUT OUTPUT1 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT1） 设置 OUPUT1 位。 0：无影响 1：设置 OUPUT1 位	0h
1	OUTPUT2	W	设置 XBAR_OUTPUT OUTPUT2 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT2） 具体描述参考 OUTPUT1。	0h

位/域	名称	R/W	描述	复位值
2	OUTPUT3	W	设置 XBAR_OUTPUT OUTPUT3 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT3） 具体描述参考 OUTPUT1。	0h
3	OUTPUT4	W	设置 XBAR_OUTPUT OUTPUT4 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT4） 具体描述参考 OUTPUT1。	0h
4	OUTPUT5	W	设置 XBAR_OUTPUT OUTPUT5 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT5） 具体描述参考 OUTPUT1。	0h
5	OUTPUT6	W	设置 XBAR_OUTPUT OUTPUT6 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT6） 具体描述参考 OUTPUT1。	0h
6	OUTPUT7	W	设置 XBAR_OUTPUT OUTPUT7 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT7） 具体描述参考 OUTPUT1。	0h
7	OUTPUT8	W	设置 XBAR_OUTPUT OUTPUT8 的 OUTPUTLATCH 的位（set the OUTPUTLATCH for XBAR_OUTPUT OUTPUT8） 具体描述参考 OUTPUT1。	0h
31:8	保留			0h

23.6.27 OUTPUT 锁存使能寄存器（OUTPUTLATCHENABLE）

偏移地址：0x6C

复位类型：CPU1.SYSRSn

此寄存器用于使能 OUTPUT 锁存寄存器（OUTPUTLATCH）中的对应位。

位/域	名称	R/W	描述	复位值
0	OUTPUT1	R/W	使能 XBAR_OUTPUT OUTPUT1 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT1） 使能 OUPUT1 位，以驱动 OUTPUT1. 0：禁用，此时不能驱动 OUTPUT1 1：使能，此时可以驱动 OUTPUT1	0h
1	OUTPUT2	R/W	使能 XBAR_OUTPUT OUTPUT2 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT2） 具体描述参考 OUTPUT1。	0h
2	OUTPUT3	R/W	使能 XBAR_OUTPUT OUTPUT3 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT3） 具体描述参考 OUTPUT1。	0h

位/域	名称	R/W	描述	复位值
3	OUTPUT4	R/W	使能 XBAR_OUTPUT OUTPUT4 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT4）具体描述参考 OUTPUT1。	0h
4	OUTPUT5	R/W	使能 XBAR_OUTPUT OUTPUT5 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT5）具体描述参考 OUTPUT1。	0h
5	OUTPUT6	R/W	使能 XBAR_OUTPUT OUTPUT6 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT6）具体描述参考 OUTPUT1。	0h
6	OUTPUT7	R/W	使能 XBAR_OUTPUT OUTPUT7 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT7）具体描述参考 OUTPUT1。	0h
7	OUTPUT8	R/W	使能 XBAR_OUTPUT OUTPUT8 的 OUTPUTLATCH 的位（enable the OUTPUTLATCH for XBAR_OUTPUT OUTPUT8）具体描述参考 OUTPUT1。	0h
31:8	保留			0h

23.6.28 OUTPUT 锁存反相寄存器（OUTPUTINV）

偏移地址：0x70

复位类型：CPU1.SYSRSn

位/域	名称	R/W	描述	复位值
0	OUTPUT1	R/W	选择 XBAR_OUTPUT 的 OUTPUT1 极性 (XBAR_OUTPUT OUTPUT1 polarity Select) 0：驱动 OUTPUT1 为高电平有效 1：驱动 OUTPUT1 为低电平有效	0h
1	OUTPUT2	R/W	选择 XBAR_OUTPUT 的 OUTPUT2 极性 (XBAR_OUTPUT OUTPUT2 polarity Select) 具体描述参考 OUTPUT1。	0h
2	OUTPUT3	R/W	选择 XBAR_OUTPUT 的 OUTPUT3 极性 (XBAR_OUTPUT OUTPUT3polarity Select) 具体描述参考 OUTPUT1。	0h
3	OUTPUT4	R/W	选择 XBAR_OUTPUT 的 OUTPUT4 极性 (XBAR_OUTPUT OUTPUT4 polarity Select) 具体描述参考 OUTPUT1。	0h
4	OUTPUT5	R/W	选择 XBAR_OUTPUT 的 OUTPUT5 极性 (XBAR_OUTPUT OUTPUT5 polarity Select) 具体描述参考 OUTPUT1。	0h
5	OUTPUT6	R/W	选择 XBAR_OUTPUT 的 OUTPUT6 极性 (XBAR_OUTPUT OUTPUT6 polarity Select) 具体描述参考 OUTPUT1。	0h

位/域	名称	R/W	描述	复位值
6	OUTPUT7	R/W	选择 XBAR_OUTPUT 的 OUTPUT7 极性 (XBAR_OUTPUT OUTPUT7 polarity Select) 具体描述参考 OUTPUT1。	0h
7	OUTPUT8	R/W	选择 XBAR_OUTPUT 的 OUTPUT8 极性 (XBAR_OUTPUT OUTPUT8 polarity Select) 具体描述参考 OUTPUT1。	0h
31:8	保留			0h

23.6.29 OUTPUT 锁定寄存器（OUTPUTLOCK）

偏移地址：0x7C

复位类型：CPU1.SYSRSn

此寄存器用于阻止对 XBAR_OUTPUT 寄存器组的中下列寄存器的写入操作，读取操作不受影响。

被锁定的寄存器包括：

- OUTPUTxMUX0TO15CFG（x=1...8）
- OUTPUTxMUX16TO31CFG（x=1…8）
- OUTPUTxMUXENABLE（x=1…8）
- OUTPUTLATCHENABLE
- OUTPUTINV

位/域	名称	R/W	描述	复位值
0	LOCK	R/WSO	锁定 XBAR_PWM 的配置（XBAR_PWM configuration Lock） 当此位置 1 时，锁定 XBAR_PWM 寄存器组寄存器的配置，将禁止对这些寄存器进行写入操作。 0：允许对寄存器进行写入操作 1：禁止对寄存器进行写入操作	0h
15:1	保留			0h
31:16	KEY	W	关键字（key） 此位域决定 LOCK 位是否被激活。只有当 KEY=0x5A5A 时，才可以设置 LOCK 位。	0h

24 直接存储器存取（DMA）

24.1 术语全称、缩写描述

表格 83 术语全称、缩写描述

中文全称	英文全称	英文缩写
指针	pointer	PTR
乒乓操作	ping-pong	-

24.2 简介

DMA 模块提供在没有 CPU 干预的情况下，外设和存储器之间传输数据的硬件方法，以此为其他系统功能释放带宽。DMA 还具有在传输数据时对数据进行正交的重构以及在缓冲区之间“乒乓”数据的能力。这对于将数据结构化成块以供 CPU 处理非常有用。

DMA 的通道 1 可以配置为比其他通道更高的优先级，除通道 1 外的其他通道完全相同。DMA 是一个基于事件的机器，所以需要一个外设或软件触发器来启动 DMA 传输。DMA 的内核是一个状态机和紧耦合的地址控制逻辑。它允许在传输过程中对数据块重新排序，和在缓冲区之间来回传输数据。DMA 本身无法周期性地启动内存传输，但是可以通过配置定时器作为触发源来实现。

控制器的强度需要用整个系统的能力来衡量。如果 CPU 带宽减少，系统能力越强。一般情况下，数据的格式不利于 CPU 最佳处理，但 DMA 可以释放 CPU 带宽，并将数据重新排列成更精简的处理模式。

24.3 主要特征

（1） 6 个 DMA 通道
- 具有独立 NVIC 中断的通道，可以单独配置
- 可以从多个外设触发源独立触发

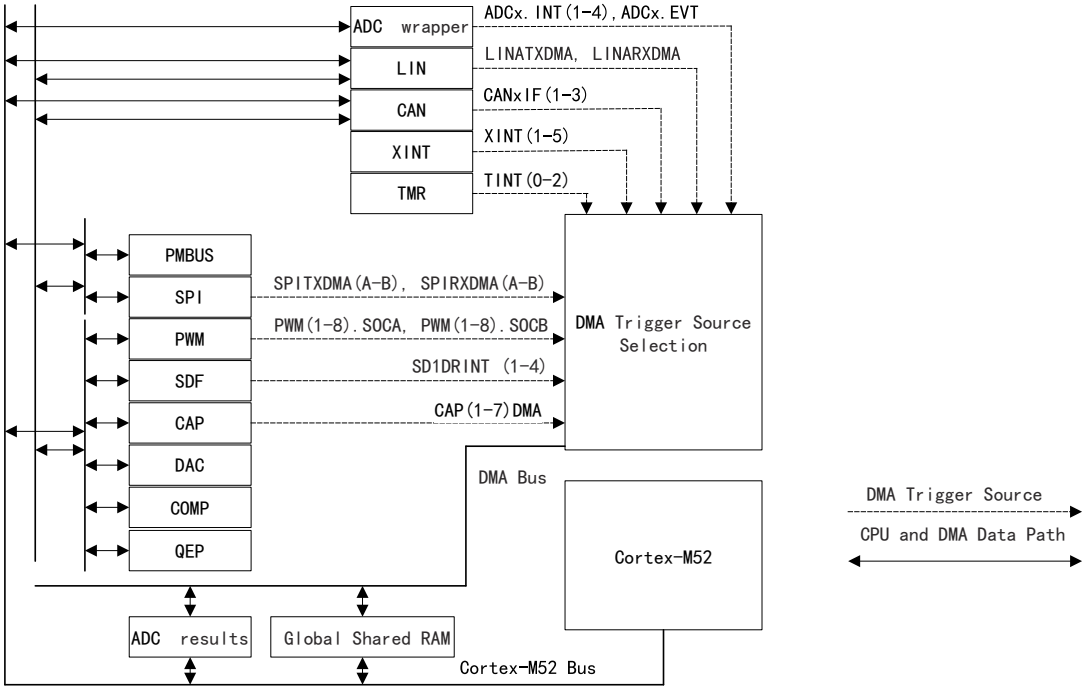
（2） 数据传输方式有三种：片外存储器到片内存储器，外设到 RAM，外设到外设

（3） 字长：16 位或 32 位(SPI 限制为 16 位)

（4） 吞吐量：3 个周期/字无仲裁

24.4 结构框图

图 26 结构框图



24.5 功能描述

24.5.1 DMA 总线

32 位地址总线，32 位数据写入总线和 32 位数据读取总线通过接口连接到 DMA 总线的存储器和寄存器位置，有时会与 CPU 存储器或外设总线共享资源。

24.5.2 外设中断事件触发源

所有 DMA 通道都可以通过软件和其他外设触发事件进行触发。CHx 寄存器和 PERINTSEL 位域可以为每个通道配置 DMA 触发源和选择外设事件触发源。软件可以始终用 PERINTFRC 位强制触发或用 PERINTCLR 位清除 DMA 触发。DMA 触发源中包括五个外部中断信号，可以连接到大部分 GPIO 引脚，以此提高事件触发功能的灵活性。在接收到外设中断事件信号时，DMA 会自动向中断源发送清除信号，以便后续发生中断事件。

如果特定的外设触发事件设置 PERINTFLG 位，该位在状态机的优先级逻辑启动该通道的突发传输前都会保持置位。如果在突发传输中生成了新的外设触发事件，突发会在响应新的外设触发事件之前完成。如果在清除置位之前又发生了新的外设触发事件，则会发生溢出，设置 OVRFLG 位。如果在清除该锁存标志的同时发生外设触发事件，则触发事件的优先级更高，PERINTFLG 保持置位。

表格 84 DMA 触发器源

选择值（8bit）	DMA 触发器源
0	无
1	ADCA.1
2	ADCA.2
3	ADCA.3
4	ADCA.4
5	ADCAEVT
6	ADCB.1
7	ADCB.2
8	ADCB.3
9	ADCB.4
10	ADCBEVT
11	ADCC.1
12	ADCC.2
13	ADCC.3
14	ADCC.4
15	ADCCEVT
16-28	无
29	XINT1
30	XINT2
31	XINT3
32	XINT4
33	XINT5
34-35	无
36	PWM1.SOCA
37	PWM1.SOCB
38	PWM2.SOCA
39	PWM2.SOCB
40	PWM3.SOCA
41	PWM3.SOCB
42	PWM4.SOCA
43	PWM4.SOCB
44	PWM5.SOCA

选择值（8bit）	DMA 触发器源
45	PWM5.SOCB
46	PWM6.SOCA
47	PWM6.SOCB
48	PWM7.SOCA
49	PWM7.SOCB
50	PWM8.SOCA
51	PWM8.SOCB
52-67	无
68	TINT0
69	TINT1
70	TINT2
71-74	无
75	CAP1DMA
76	CAP2DMA
77	CAP3DMA
78	CAP4DMA
79	CAP5DMA
80	CAP6DMA
81	CAP7DMA
82-95	无
96	SD1FLT1
97	SD1FLT2
98	SD1FLT3
99	SD1FLT4
100-108	无
109	SPITXDMAA
110	SPIRXDMAA
111	SPITXDMAB
112	SPIRXDMAB
113-116	无
117	LINATXDMA
118	LINARXDMA
119-126	无

选择值（8bit）	DMA 触发器源
127	FLB1_INT
128	FLB2_INT
129	FLB3_INT
130	FLB4_INT
131-166	无
167	CANAIF1
168	CANAIF2
169	CANAIF3
170	CANBIF1
171	CANBIF2
172	CANBIF3
173-255	无

图 27 DMA 触发器结构

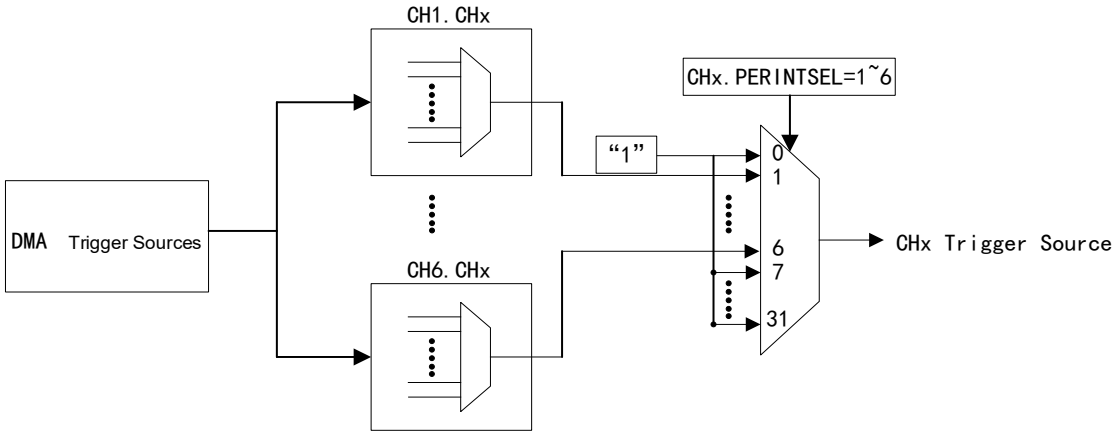
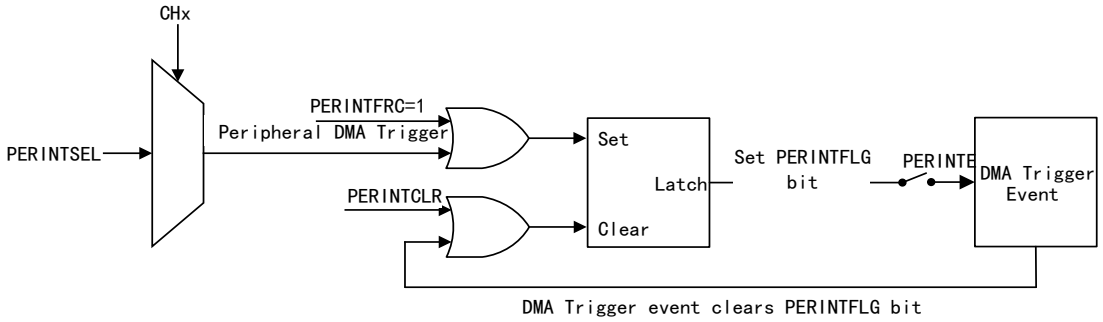


图 28 DMA 外设中断触发输入



24.5.3 通道优先级

可以根据以下两种优先级方案来确定通道优先级：

- 循环轮询模式
- 通道 1 高优先级模式

循环轮询模式

所有通道具有相同的优先级，每个使能的通道按照循环轮询的方式服务。一个通道传输完一个突发的数据后，下一个通道开始工作。可以配置每个通道传输的突发大小。如果在最后一个使能的通道传输完成后，没有其他待处理的通道，循环轮询状态机进入空闲状态。

PRIORITYRESET 位可以将循环轮询状态机重置为空闲状态。从空闲状态开始，优先响应通道 1。但是如果 DMA 正在处理另一个通道 x，则需要处理 x 到轮询结束之间的所有其他待处理通道后才会处理已使能的通道 1。所有通道的优先级相同。

通道 1 高优先级模式

通道 1 优先级高于所有其他通道。其他通道优先级相同，并且已启用的通道都按循环轮询方式进行服务。如果在最后一个使能的通道传输完成后，没有其他待处理的通道，循环轮询状态机进入空闲状态。

此模式下通道 1 一般用于数据速率比较高的 ADC，但是也可以与其他外设一起使用。在通道 1 上，不能同时使用高优先级模式和单次模式。其他通道可以在通道 1 处于高优先级模式时使用单次模式。

24.5.4 地址指针和传输控制

突发（内部）循环

突发循环在接收到 DMA 通道触发时发送 BURSTSIZE 位域配置的字，字的数量为 BURSTSIZE+1。BURSTSIZE 允许在一个突发中传输最多 32 个 16 位字。所有通道都可以通过 DATASIZE 位选择支持 16 位或 32 位字的突发。

通道都包含源地址和目标地址的影子地址指针。每次传输开始时，每个指针的影子版本都会被复制到相应的活跃寄存器中。在突发循环中，每传输一个字都会将 SRCBURSTSTEP 或 DSTBURSTSTEP 位域中有符号的值加到对应的活跃寄存器中。

发送（外部）循环

每个通道的发送循环发送 TRANSFERSIZE 位域配置的突发。由于 TRANSFERSIZE 是 16 位，允许的传输量超出所有实际要求。在发送循环中，每个突发传输完成后，可以通过禁用地址包装或当突发数量等于 SRC_WRAP_SIZE 寄存器的 WRAPSIZE+1 或 DST_WRAP_SIZE 寄存器的 WRAPSIZE+1 来修改活跃地址指针。在 DMA 传输结束后，DMA 可以传输 BURSTSIZE+1 x TRANSFERSIZE+ 1 个字。

单次模式

单次模式默认禁用。当禁用单次模式时，每接收到 DMA 通道触发，就会传输一次突发。突发传输完成后，状态机无论刚完成的通道中是否有另一个触发在等

待，都继续处理优先级方案中的下一个待处理通道。以此来防止单个通道垄断 DMA 总线。当使能单次模式时，DMA 会在单个 DMA 通道触发时传输所有突发。此模式可能会导致一个触发使用了大部分 DMA 带宽。

连续模式

连续模式默认禁用。当禁用连续模式时，状态机在发送循环传输完所有突发后禁用通道。置位 RUN 位可以启动通道然后开始另一次传输。使能连续模式时，状态机会在发送循环传输完所有突发后让通道保持活动状态。每个 DMA 通道都可以使用 CHINTMODE 位在 DMA 开始传输或结束时触发自己的 NVIC 中断。

（1） 源/目标地址指针

数据写入或读取的第一个位置的起始地址会写入影子寄存器。在传输开始时，活跃寄存器加载影子寄存器的数据。活跃寄存器充当当前地址的指针。

（2） 源/目标起始地址指针

包装指针。写入影子起始寄存器的值，在传输开始时加载到活跃起始寄存器中。在包装条件下，活跃起始寄存器加上对应的包装步进寄存器中的有符号值后会加载到活跃寄存器中。

（3） 起始/目标突发大小

指定一次突发中传输的字的数量，在每次突发传输开始时加载到 BURST_COUNT 寄存器中。每传输一个字，BURST_COUNT 寄存器都会减少，当寄存器为零时，突发传输完成，此时可以为下一个通道提供服务。ONESHOT 位和外设决定通道使用的模式和突发的最大大小。

表格 85 **BURSTSIZE 与 DATASIZE 的关系**

BURST_SIZE	DATASIZE=0	DATASIZE=1
0	1	2
1	2	2
2	3	4
3	4	4
4	5	6
5	6	6
…	…	…
30	31	32
31	32	32

（4） 起始/目标发送大小

指定使能 CPU 中断时要发送的突发数量。CONTINUOUS 位决定传输模式，CHINTMODE 位决定是否在传输开始或结束时生成中断。TRANSFERSIZE 位在每次发送开始时加载到 TRANSFERCOUNT 位中。通道已传输的突发数量为零

时，DMA 发送完成。

（5） 起始/目标包装大小

可以实现循环寻址类型功能。指定在当前地址指针绕回到起始位置前要传输的突发数量。每次传输开始时加载到适当的包装计数器寄存器中。通道已传输突发数量为零时，将在适当的起始/目的地址指针上执行包装过程。起始和目的指针都有单独的大小和计数寄存器。这些寄存器大于 TRANSFERSIZE 位时才能禁用包装功能。

（6） 起始/目标突发步进

指定突发传输中起始和目标地址的步进。这是一个带符号的二进制补码数，可以递增或递减地址指针。不需要递增时这些寄存器的值可以设置为零。

（7） 起始/目标发送步进

指定在完成当前突发传输后开始下一个突发传输的地址偏移量。这是一个带符号的二进制补码数，可以递增或递减地址指针。在寄存器或数据存储器位置以恒定间隔排列的情况下使用。

（8） 起始/目标包装步进

这是一个带符号的二进制补码数，可以递增或递减地址指针。当包装计数器为零时，代表要从起始地址寄存器的指针中增加或减少的字数，以此设置新的起始地址。这实现了一种循环寻址模式。

注意：步进寄存器只针对 16 位地址。如果要递增一个 32 位地址，可以在这些寄存器中写入值 2。

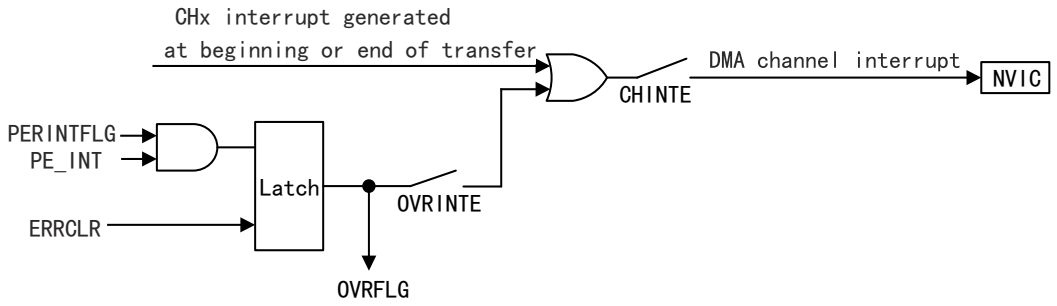
（9） 通道中断模式

此模式位选择相应通道的 DMA 中断是发生在新传输开始时或者结束时。如果不使用连续模式，则一般在传输完成时生成中断。如果 DMA 使用连续模式的乒乓缓冲区方案，则可以在活跃寄存器被复制到影子寄存器组后生成中断。

24.5.5 溢出检测

当 DMA 收到外设事件触发时设置 PERINTFLG 位，将通道挂起到 DMA 状态机，当该通道的突发传输开始时清零。如果在 PERINTFLG 位被置位和清零间出现额外的事件触发，第二个触发会丢失，此时 OVRFLG 位被设置。如果使能了溢出中断，通道中断将生成到 NVIC 模块。

图 29 DMA 溢出检测逻辑



24.5.6 流水线和吞吐量

DMA 模块由一个三级流水线组成。

图 30 三级流水线 DMA 传输

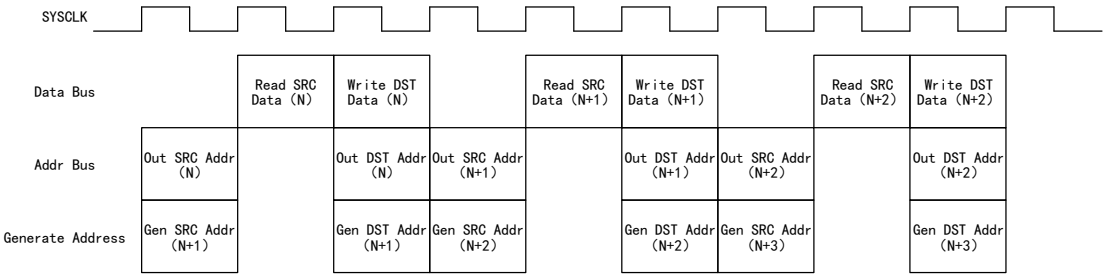
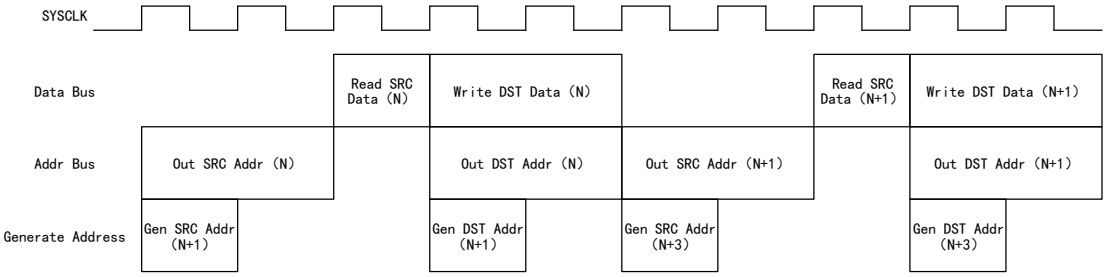


图 31 有一个读档的三级流水线传输



DMA 的总吞吐量受到流水线和以下条件的影响：

- 与 CPU 的冲突可能会增加延迟槽
- 每个突发传输开始时会加入 1 个周期的延迟
- 从 CH1 高优先级中断返回时会加入 1 个周期的延迟
- 32 位传输的速度是 16 位传输的两倍

如果要从 SRAM1 传输 128 个 16 位字到 SRAM3，可以配置一个通道以传输每突发为 16 个字的 8 个突发。传输需要以下时间：

8 bursts * [(3 cycles/word * 16 words/burst) + 1] = 392 cycles

如果通道配置为一次传输相同数量的 32 位数据，传输需要以下时间：

8 bursts * [(3 cycles/word * 8 words/burst) + 1] = 200 cycles

24.5.7 CPU 仲裁

DMA 搬运数据是独立于 CPU0、CPU1 运行。当 DMA 和 CPU0、CPU1 试图同时访问一个地址时，仲裁逻辑将会解决这种冲突。

DMA、CPU0 和 CPU1 访问任意一个从机的优先级顺序为：
DMA>CPU0>CPU1。

相同外设类型的实例彼此冲突，并且由于不同外设类型的实例可以共用一个总线接口，将造成更多访问冲突的情况。这些共用总线的外设包括：

- 外设帧 1：COMP，DAC ，QEP，SDF，CAP，PWM

- 外设帧 2：PMBus，SPI

但是 ADC 结果寄存器是一个例外，它为每个总线主机复制。CPU 和 DMA 不需要任何主机的停顿或仲裁，就可以同时读取这些结果寄存器。DMA 传输包括四个阶段：发送源地址、读取源数据、发送目的地址和写目的数据。当 CPU 在 DMA 传输过程中访问外设/内存导致冲突时，CPU 会一直停滞直到当前的 DMA 访问完成，而不是直到整个 DMA 传输完成。

对于设备上不同的接口，按以下优先级顺序：

（1） 仲裁采用固定的优先级最高的仲裁方案：DMA 写>DMA 读>CPU 写>CPU 读

（2） SRAM 访问的优先级方案为轮询。

（3） ADC 结果为 CPU 和 DMA 重复，因此在读取结果寄存器时不需要仲裁。这允许所有主机同时访问 ADC 结果寄存器，而不会延迟。

注意：

（1） 如果 CPU0 在进行读-写-读操作，同时 DMA 执行一个写的操作，则 DMA 写的数据将会被丢失。因此需要避免 CPU 写操作和 DMA 写操作在同一拍执行（例外：DMA 和 CPU0、CPU1 零等待访问 ADC 结果寄存器可在同一拍执行）。

（2） DMA 在传输数据的时候不可以被配置。

24.6 寄存器组地址

表格 86 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
DMAregs	DMA_REGS	0x5011 0C00	0x5011 0C3F
DMACH1regs	DMA_CH	0x5011 0C40	0x5011 0C7F
DMACH2regs	DMA_CH	0x5011 0C80	0x5011 0CBF
DMACH3regs	DMA_CH	0x5011 0CC0	0x5011 0CFF
DMACH4regs	DMA_CH	0x5011 0D00	0x5011 0D3F
DMACH5regs	DMA_CH	0x5011 0D40	0x5011 0D7F
DMACH6regs	DMA_CH	0x5011 0D80	0x5011 0DBF
DMASRCSELregs	DMA_SRC_SEL	0x5002 0400	0x5002 0430

24.7 寄存器地址映射

表格 87 DMA_REGS 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
DMACTRL	控制寄存器	0x00	√
DEBUGCTRL	调试控制寄存器	0x02	√
PRIORITYCTRL1	优先级控制 1 寄存器	0x08	√
PRIORITYSTAT	优先级状态寄存器	0x0C	√

表格 88 DMA_CH 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
MODE	模式寄存器	0x00	√
CONTROL	控制寄存器	0x02	√
BURST_SIZE	突发大小寄存器	0x04	√
BURST_COUNT	突发计数器寄存器	0x06	√
SRC_BURST_STEP	源突发步进寄存器	0x08	√
DST_BURST_STEP	目的突发步进寄存器	0x0A	√
TRANSFER_SIZE	发送器大小寄存器	0x0C	√
TRANSFER_COUNT	发送器计数器寄存器	0x0E	√
SRC_TRANSFER_STEP	源发送器步进寄存器	0x10	√
DST_TRANSFER_STEP	目的发送器步进寄存器	0x12	√
SRC_WRAP_SIZE	源包装大小寄存器	0x14	√
SRC_WRAP_COUNT	源包装计数器寄存器	0x16	√
SRC_WRAP_STEP	源包装步进寄存器	0x18	√
DST_WRAP_SIZE	目的包装大小寄存器	0x1A	√
DST_WRAP_COUNT	目的包装计数器寄存器	0x1C	√
DST_WRAP_STEP	目的包装步进寄存器	0x1E	√
SRC_BEG_ADDR_SHADOW	影子源起始地址寄存器	0x20	√
SRC_ADDR_SHADOW	影子源地址寄存器	0x24	√
SRC_BEG_ADDR_ACTIVE	活跃源起始地址寄存器	0x28	√
SRC_ADDR_ACTIVE	活跃源地址寄存器	0x2C	√
DST_BEG_ADDR_SHADOW	影子目的起始地址寄存器	0x30	√
DST_ADDR_SHADOW	影子目的地址寄存器	0x34	√
DST_BEG_ADDR_ACTIVE	活跃目的起始地址寄存器	0x38	√
DST_ADDR_ACTIVE	活跃目的地址寄存器	0x3C	√

表格 89 DMA_SRC_SEL 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
DMACHSRCSELLOCK	通道触发源选择锁定寄存器	0x08	√
DMACHSRCSEL1	通道触发源选择寄存器 1	0x2C	√
DMACHSRCSEL2	通道触发源选择寄存器 2	0x30	√

24.8 寄存器功能描述

24.8.1 控制寄存器(DMACTRL)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	HARDRESET	R-0/W1S	硬件复位（Hard Reset） 写入 1 时，复位整个 DMA 并中止当前访问。软件复位为每个通道提供一个位来执行较温和的复位。如果需要访问其他 DMA 寄存器，软件中需要一个周期延迟。写 0 被忽略，读回 0。	0h
1	PRIORITYRESET	R-0/W1S	重置优先级（ priority reset） 写入 1 时，此位重置轮询状态机。服务从第一个使能的通道启动。需要完成挂起的突发传输，才能重置通道优先级机器。如果 CH1 是高优先级，则从下一个最高使能的通道重新启动状态机，如果同时在 CH1 服务突发时写入该位，则 CH1 突发和下一个等待的低优先级突发都将在状态机重置之前完成。写 0 被忽略，读回 0。	0h
15:2	保留			0h

24.8.2 调试控制寄存器(DEBUGCTRL)

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
14:0	保留			0h
15	FREE	R/W	模拟停止时，配置 DMA 的工作状态（Configure DMA Work Status When Core is in Halted） 0：完成当前的读写操作后停止 1：在模拟暂停期间继续运行	0h

24.8.3 优先级控制 1 寄存器(PRIORITYCTRL1)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h

位/域	名称	R/W	描述	复位值
15:1	CH1PRIORITY	R/W	配置 DMA 通道 1 优先级（DMA Channel 1 Priority configuration） 要改变优先级，需要禁用所有通道。修改优先级后要进行优先级复位再重新启动通道 0：CH1 优先级与其他通道相同 1：CH1 优先级高于其他通道	0h

24.8.4 优先级状态寄存器(PRIORITYSTAT)

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	ACTIVESTS	R	活动通道状态（Active Channel Status） 表示当前处于活跃状态或正在执行传输的通道。 0：无活跃通道 1：CH 1 2：CH 2 … 6：CH 6 7：保留	0h
3	保留			0h
6:4	ACTIVESTS_SHADOW	R	活跃通道状态影子（Active Channel Status Shadow） 作用于通道 1 高优先级模式。当 CH1 被服务时，ACTIVESTS 位被复制到此位，并指示被 CH1 中断的通道。CH1 服务完成后，此位被复制回 ACTIVESTS 位。如果此位为零或与 ACTIVESTS 位相同，则不会有通道挂起。 0：无活跃通道 1：CH 1 2：CH 2 … 6：CH 6 7：保留	0h
15:7	保留			0h

24.8.5 模式寄存器(MODE)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	PERINTSEL	R/W	选择外设事件触发源（Peripheral Event Trigger Source Select） 遗留的位，应该设置为通道号。实际的源选择通过 DMACHSRCSELn 寄存器完成，它是 DMA_SRC_SEL 组的一部分。	0h
6:5	保留			0h

位/域	名称	R/W	描述	复位值
7	OVRINTE	R/W	使能溢出中断（Overflow Interrupt Enable） 0：禁止 1：使能	0h
8	PERINTE	R/W	使能外设事件触发（Peripheral Event Trigger Enable） 0：禁止，所选的外设和软件都不能启动 DMA 突发 1：使能	0h
9	CHINTMODE	R/W	配置产生通道中断模式（Generation Channel Interrupt Mode configuration） 0：中断在新传输开始时产生 1：中断在传输结束时产生	0h
10	ONESHOT	R/W	禁止一次模式（Single Shot Mode disable） 0：使能，通道每个触发器只执行一次突发 1：禁止，每个外设事件触发导致通道执行整个传输	0h
11	CONTINUOUS	R/W	使能连续模式（Continuous Mode Enable） 当 TRANSFERCOUNT 为零时是否使能连续模式。 0：禁止，DMA 停止并清除 RUNSTS 位 1：使能，重新初始化通道并等待下一个事件触发器	0h
13:12	保留			0h
14	DATASIZE	R/W	配置数据大小（Data Size configuration） 这个位决定了 DMA 通道每次读/写操作的传输数据大小。其他 DMA 寄存器中的所有数据长度和偏移量都指向 16 位字。指针步长增量配置为容纳 32 位字。 0：16 位 1：32 位	0h
15	CHINTE	R/W	使能通道中断（Channel Interrupt Enable） 0：禁止 1：使能	0h

24.8.6 控制寄存器(CONTROL)

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	RUN	R-0/W1S	使能通道（Channel Enable） 该位用于在通道中断后恢复。通常用于在配置后使能 DMA 通道。然后等待第一个外设事件触发器来使能突发事件。 0：禁止 1：使能，并使 CHENFLG=1	0h
1	HALT	R-0/W1S	停止通道（Halt Channel） 在当前读/写访问完成后，设置此位将使 DMA 通道停止在当前状态。	0h

位/域	名称	R/W	描述	复位值
2	SOFTRESET	R-0/W1S	通道软件复位（Channel Soft Reset） 在访问其他 DMA 寄存器之前，软件中需要一个周期延迟。设置此位将使通道在当前读写访问完成后进入默认状态： TRANSFERSTS= 0 BURSTSTS= 0 RUNSTS= 0 BURSTCOUNT = 0 TRANSFERCOUNT = 0 SRC_WRAP_COUNT 的 WRAPSIZE = 0 DST_WRAP_COUNT 的 WRAPSIZE = 0	0h
3	PERINTFRC	R-0/W1S	强制外设事件触发（Force Peripheral Event Trigger） 此位可以用来在软件中使能 DMA 传输。如果设置了 PERINTE 位，则设置该位将设置 PERINTFLG 位，从而触发 DMA 突发。	0h
4	PERINTCLR	R-0/W1S	清除外设事件触发（Peripheral Event Trigger Clear） 在初始化 DMA 模块时设置此位可以清除 PERINTFLG，可以取消挂起的事件触发器。如果在设置此位时有一个事件触发器到达，则该触发器具有优先级并设置 PERINTFLG。	0h
6:5	保留			0h
7	ERRCLR	R-0/W1S	清除错误（Error Clear） 在初始化 DMA 模块或检测到溢出条件时设置此位，可以清除 OVRFLG 位。如果在设置该位时发生溢出事件，则设置溢出优先级和 OVRFLG 位。	0h
8	PERINTFLG	R	外设事件触发标志（Peripheral Event Trigger Flag） 设置此位时，此位表示外设事件触发器已经挂起。该位在第一次突发传输开始时，该位自动清除。 0：等待事件触发器 1：事件触发器挂起	0h
10:9	保留			0h
11	TRANSFERSTS	R	正在发送标志（Transferring Flag） 此位在 DMA 传输开始时设置。将地址寄存器的内容被复制到阴影集，设置 TRANSFERCOUNT 与 TRANSFERSIZE 相等。当 TRANSFERCOUNT 达到零或者设置 HARDRESET 位或 SOFTRESET 位时，清除该位。 0：无发送 1：无论 DMA 是否在传输突发数据，当前都在传输过程中	0h

位/域	名称	R/W	描述	复位值
12	BURSTSTS	R	突发标志（Burst Flag） 此位在 DMA 突发开始时设置。设置 BURSTCOUNT 与 BURSTSIZE 相等。当 BURSTCOUNT 达到零或者设置 HARDRESET 位或 SOFTRESET 位时，清除该位。 0：无突发 1：DMA 当前正在服务或暂停来自该通道的突发传输	0h
13	RUNSTS	R	使能通道标志（Channel Enable Flg） 表示 DMA 通道已经准备好响应外设事件触发器。设置 RUN 位时设置此位。 当发生以下情况时清除此位：当传输完成并且禁用连续模式，设置 HARDRESET 位或 SOFTRESET 或 HALT 位。 0：禁止 1：已使能	0h
14	OVRFLG	R	溢出标志（Overflow Flag） 表示重复收到了外设事件触发器。置位 ERRCLR 位可以清除此位。 0：无溢出 1：溢出	0h
15	保留			0h

24.8.7 突发大小寄存器(BURST_SIZE)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	BURSTSIZE	R/W	突发大小（Burst Size） 以 16 位字表示突发大小。实际尺寸等于 BURSIZE + 1。	0h
15:5	保留			0h

24.8.8 突发计数器寄存器(BURST_COUNT)

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	BURSTCNT	R	突发计数器（Burst Count） 表示当前突发中剩余的字数。 00000：0 00001：1 00010：2 … 11111：31	0h

位/域	名称	R/W	描述	复位值
15:5	保留			0h

24.8.9 源突发步进寄存器(SRC_BURST_STEP)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	SRCBURSTSTEP	R/W	源突发步进（Source Burst Step） 表示一个突发完成后源地址的变化，大小在-4096~4095 之间的 16 位二进制补码。在每次读/写操作后将该值添加到源地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.10 目的突发步进寄存器(DST_BURST_STEP)

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DESBURSTSTEP	R/W	目的突发步进（Destination Burst Step） 表示在一个突发完成后目的地址的变化。大小在-4096~4095 之间的 16 位二进制补码。在每次读/写操作后将该值添加到目的地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.11 发送器大小寄存器(TRANSFER_SIZE)

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	TRANSFERSIZE	R/W	发送器大小（Transfer Size） 此位域以突发的形式指定发送大小。实际大小等于TXSIZE+1。	0h

24.8.12 发送器计数器寄存器(TRANSFER_COUNT)

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	TRANSFERCNT	R	发送器计数器（Transfer Count） 表示当前传输中剩余的突发数。	0h

24.8.13 源发送器步进寄存器(SRC_TRANSFER_STEP)

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	SRCTRANSFERSTEP	R/W	源发送器步进（Source Transfer Step） 表示一个突发完成后源地址的变化，大小在-4096~4095 之间的 16 位二进制补码。在每次读/写操作后将该值添加到源地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.14 目的发送器步进寄存器(DST_TRANSFER_STEP)

偏移地址：0x12

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DSTTRANSFERSTEP	R/W	目的突发步进（Destination Burst Step） 表示在一个突发完成后目的地址的变化。大小在-4096~4095 之间的 16 位二进制补码。在每次读/写操作后将该值添加到目的地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.15 源包装大小寄存器(SRC_WRAP_SIZE)

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPSIZE	R/W	源包装大小（Source Wrap Size ） 表示在源地址绕到开始地址之前要传输的突发数。实际数量等于 SRCWSIZE + 1。如果此位大于 TXSIZE 位的值，可以关闭换行功能。	FFFFh

24.8.16 源包装计数器寄存器(SRC_WRAP_COUNT)

偏移地址：0x16

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPCNT	R	源包装计数器（Source Wrap Count） 表示在包装源地址之前剩余的突发数。	0h

24.8.17 源包装步进寄存器(SRC_WRAP_STEP)

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPSTEP	R/W	源包装步进（Source Transfer Step） 表示当包装计数器达到零时源起始地址的变化。大小在-4096~4095 之间的 16 位二进制补码。包装发生时将该值添加到源地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.18 目的包装大小寄存器(DST_WRAP_SIZE)

偏移地址：0x1A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPSIZE	R/W	目的包装大小（Destination Wrap Size） 表示在目的地址绕到开始地址之前要传输的突发数。实际数量等于 DESWSIZE + 1。如果此位大于 TXSIZE 位的值，可以关闭换行功能。	FFFFh

24.8.19 目的包装计数器寄存器(DST_WRAP_COUNT)

偏移地址：0x1C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPCNT	R	目的包装计数器（Destination Wrap Count） 表示在包装目的地址之前剩余的突发数。	0h

24.8.20 目的包装步进寄存器(DST_WRAP_STEP)

偏移地址：0x1E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	WRAPSTEP	R/W	源包装步进（Source Transfer Step） 表示当包装计数器达到零时目的起始地址的变化。大小在-4096~4095 之间的 16 位二进制补码。包装发生时将该值添加到目的地址中。 0000000000000000：无地址变化 0000000000000001：地址+1 0000000000000010：地址+2 … 0000111111111110：地址+4094 0000111111111111：地址+4095 0001000000000000：地址-4096 0001000000000001：地址-4095 … 1111111111111110：地址-2 1111111111111111：地址-1	0h

24.8.21 影子源起始地址寄存器(SRC_BEG_ADDR_SHADOW)

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	BEGADDR	R/W	影子源起始地址（Shadow Source Begin Address） 此位中的值在发送开始时被加载到 SRC_BEG_ADDR_ACTIVE 寄存器的 BEGADDR 位域中，并作为源地址的起始值。在发送过程中可以安全地更新此寄存器。	0h

24.8.22 影子源地址寄存器(SRC_ADDR_SHADOW)

偏移地址：0x24

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ADDR	R/W	影子源地址（Shadow Source Address） 此位的值在发送开始时被加载到 SRC_ADDR_ACTIVE 寄存器的 ADDR 位域中，并作为源地址的值。在发送过程中可以安全地更新此寄存器。	0h

24.8.23 活跃源起始地址寄存器(SRC_BEG_ADDR_ACTIVE)

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	BEGADDR	R	活跃源起始地址（Active Source Begin Address） 此位域有两个功能，当正在传输时，保存源地址的当前起始值，可能在包装后更新。当传输开始时，装载来自 SRC_BEG_ADDR_SHADOW 寄存器的 BEGADDR 位域的影子地址。	0h

24.8.24 活跃源地址寄存器(SRC_ADDR_ACTIVE)

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ADDR	R	活跃源地址（Active Source Address） 当正在传输时，这个位域保存源地址的当前起始值，可能在包装，写入，突发后更新。当传输开始时，这个位域装载来自 SRC_ADDR_SHADOW 寄存器的 ADDR 位域的影子地址	0h

24.8.25 影子目的起始地址寄存器(DST_BEG_ADDR_SHADOW)

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	BEGADDR	R/W	影子目的起始地址（Shadow Destination Begin Address） 此位域的值发送开始时加载到 DST_BEG_ADDR_ACTIVE 寄存器的 BEGADDR 位域中，并作为目的地址的起始值。在发送过程中可以安全地更新此寄存器。	0h

24.8.26 影子目的地址寄存器(DST_ADDR_SHADOW)

偏移地址：0x34

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ADDR	R/W	影子目的地址（Shadow Destination Address） 此位域的值发送开始时被加载到 DST_ADDR_ACTIVE 寄存器的 ADDR 位域中，并作为目的地址的值。在发送过程中可以安全地更新此寄存器。	0h

24.8.27 活跃目的起始地址寄存器(DST_BEG_ADDR_ACTIVE)

偏移地址：0x38

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	BEGADDR	R	活跃目的起始地址（Active Destination Begin Address） 此位域有两个功能，当正在传输时，保存源地址的当前目的值。此地址可能在包装后更新。当传输开始时，装载来自 DST_BEG_ADDR_SHADOW 寄存器的 BEGADDR 位域的影子地址。	0h

24.8.28 活跃目的地址寄存器(DST_ADDR_ACTIVE)

偏移地址：0x3C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	ADDR	R	活跃目的地址（Active Destination Address） 在传输的过程中，保存目的地址的当前值，可能在包装，写入，突发后更新。	0h

24.8.29 通道触发源选择锁定寄存器(DMACHSRCSELLOCK)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DMACHSRCSEL1	R/WSonce	锁定 DMACHSRCSEL1 寄存器（DMACHSRCSEL1 Register Lock） 锁定机制只适用于写入。允许对具有 LOCK 保护的寄存器进行读操作 0：未锁定 1：锁定	0h
1	DMACHSRCSEL2	R/WSonce	锁定 DMACHSRCSEL2 寄存器（DMACHSRCSEL2 Register Lock） 锁定机制只适用于写入。允许对具有 LOCK 保护的寄存器进行读操作 0：未锁定 1：锁定	0h
31:2	保留			0h

24.8.30 通道触发源选择寄存器 1(DMACHSRCSEL1)

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	CH1	R/W	选择 DMA CH1 的触发和同步源（DMA CH1 Trigger and Sync Source Selects） 选择通道 1 的触发和同步源	0h
15:8	CH2	R/W	选择 DMA CH2 的触发和同步源（DMA CH2 Trigger and Sync Source Selects） 选择通道 2 的触发和同步源	0h
23:16	CH3	R/W	选择 DMA CH3 的触发和同步源（DMA CH3 Trigger and Sync Source Selects） 选择通道 3 的触发和同步源	0h
31:24	CH4	R/W	选择 DMA CH4 的触发和同步源（DMA CH4 Trigger and Sync Source Selects） 选择通道 4 的触发和同步源	0h

24.8.31 通道触发源选择寄存器 2(DMACHSRCSEL2)

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	CH5	R/W	选择 DMA CH5 的触发和同步源（DMA CH5 Trigger and Sync Source Selects） 选择通道 5 的触发和同步源	0h
15:8	CH6	R/W	选择 DMA CH6 的触发和同步源（DMA CH6 Trigger and Sync Source Selects） 选择通道 6 的触发和同步源	0h
31:16	保留			0h

25 双核调试系统（DCDS）

25.1 术语全称、缩写描述

表格 90 术语全称、缩写描述

中文全称	英文全称	英文缩写
调试系统	Debug system	Debugsys
高级跟踪总线	Advanced trace bus	ATB
系统 ROM 表	System ROM table	SYS ROM
跟踪端口接口单元	Trace port Interface unit	TPIU
CoreSight 跟踪漏斗	CoreSight tracking funnel	CSTF
系统交叉触发器接口	System cross trigger interface	SYS CTI
串行线跟踪端口	Serial line tracking port	SWO
SWO 跟踪漏斗	SWO tracking funnel	SWTF
全局时间戳生成器	Global timestamp generator	TSG
微控制器调试单元	Microcontroller debugging unit	DBGMCU

25.2 简介

Debugsys（调试系统）采用双核 SoC。

25.3 结构框图

图 32 结构框图

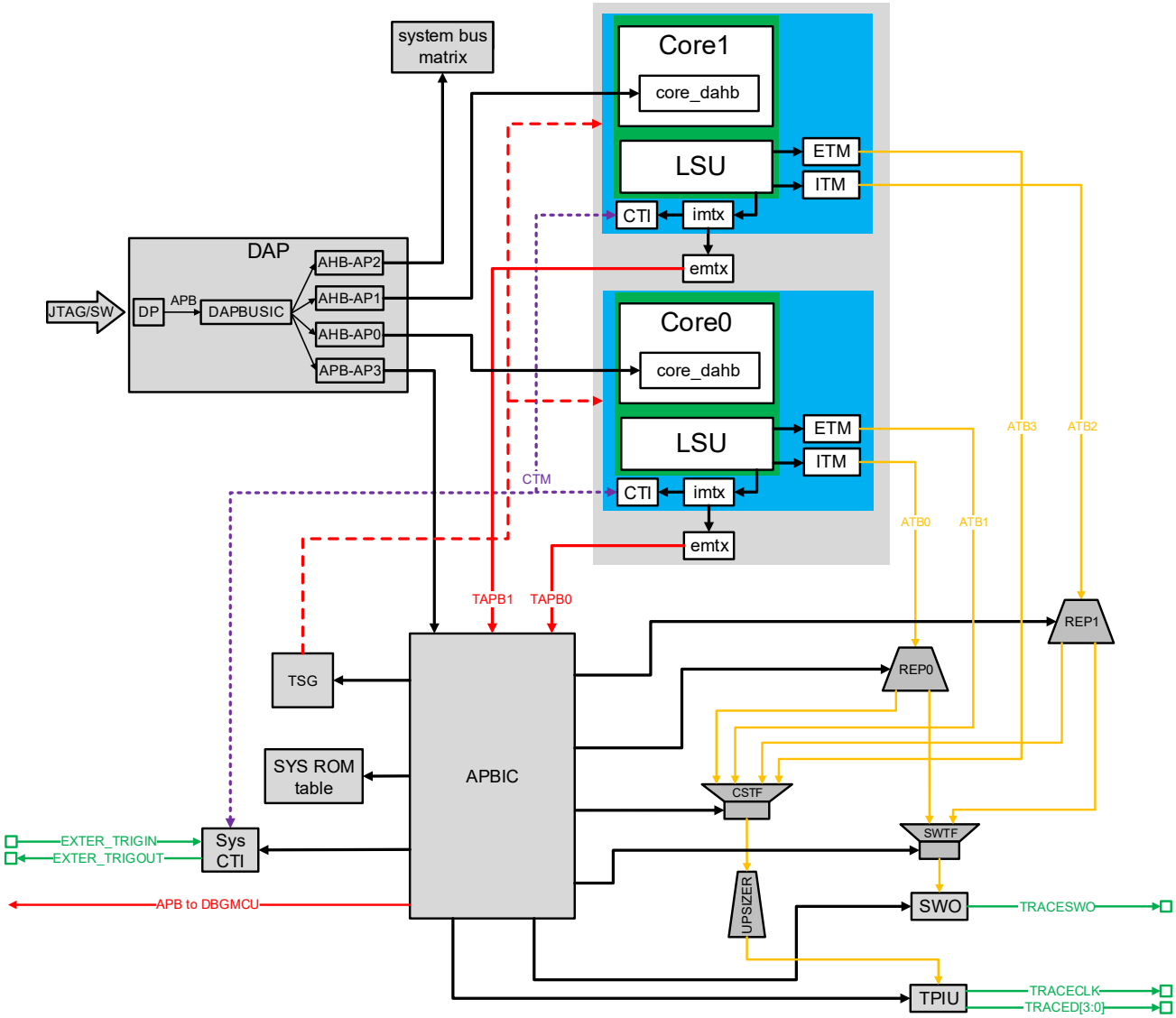
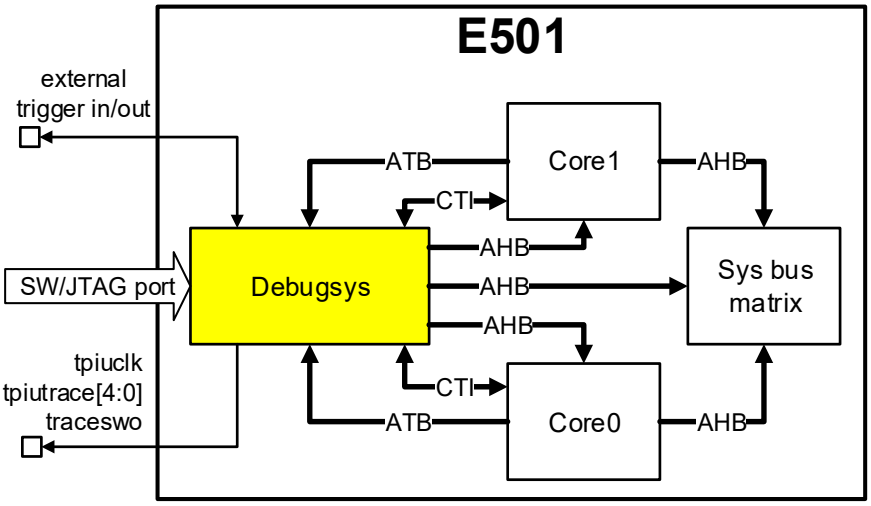


图 33 调试系统的典型集成



25.4 主要特征

（1） 支持系统中每个 CPU 内核的独立断点调试

（2） 支持代码执行跟踪

（3） 支持 JTAG 调试端口

（4） 支持串行线调试端口

（5） 支持软件仪器化

（6） 支持交叉触发

（7） 支持触发器输入和触发器输出

（8） 支持访问系统总线矩阵

（9） 支持串行线（SWO）跟踪端口

（10） 支持 TPIU 跟踪端口

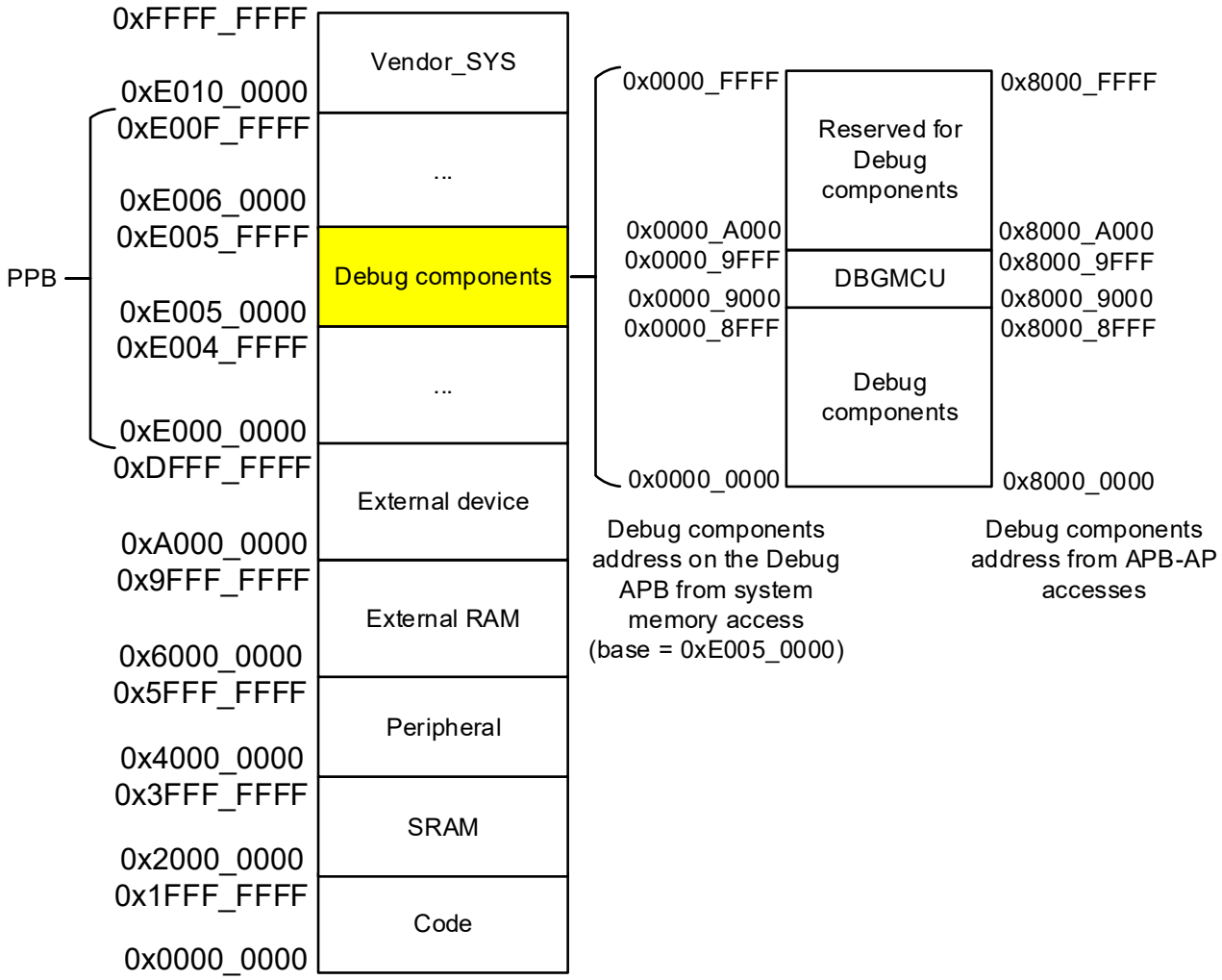
（11） 支持 Arm CoreSight SoC-400 调试和跟踪组件

25.5 功能描述

25.5.1 地址映射

系统地址映射

图 34 系统地址映射



表格 91 PPB 地址映射

系统内存地址范围	应用
0xE000_0000~0xE000_0FFF	仪器跟踪宏单元（ITM）
0xE000_1000~0xE000_1FFF	数据观察点和跟踪（DWT）单元观察点
0xE000_2000~0xE000_2FFF	断点单元（BPU）断点
0xE000_3000~0xE000_DFFF	保留
0xE000_E000~0xE000_EFFF	系统控制空间（SCS）
0xE000_F000~0xE001_EFFF	保留
0xE002_E000~0xE002_EFFF	保留
0xE003_E000~0xE004_0FFF	保留
0xE004_1000~0xE004_1FFF	嵌入式跟踪宏单元（ETM）
0xE004_2000~0xE004_2FFF	交叉触发接口（CTI）
0xE004_3000~0xE004_5FFF	保留
0xE004_6000~0xE004_6FFF	可编程 MBIST 控制器（PMC-100）
0xE004_7000~0xE004_7FFF	保留
0xE004_8000~0xE004_8FFF	软件内置自检（SBIST）控制器
0xE004_9000~0xE004_9FFF	保留

系统内存地址范围	应用
0xE005_0000~0xE005_9FFF	外部调试组件和 DBGMCU
0xE005_A000~0xE005_FFFF	为外部调试组件保留
0xE006_0000~0xE00F_EFFF	保留
0xE00F_F000~0xE00F_FFFF	处理器 ROM 表

调试组件地址映射

在下表中，地址范围 0 显示了自系统内存访问的调试 APB 上的调试组件地址，其基本地址为 0xE005_0000。地址范围 1 显示了自 APB-AP 访问的调试组件地址。

表格 92 调试组件地址映射

地址范围 0	地址范围 1	应用
0x0000_0000~0x0000_0FFF	0x8000_0000~0x8000_0FFF	系统 ROM 表（SYS ROM）
0x0000_1000~0x0000_1FFF	0x8000_1000~0x8000_1FFF	跟踪端口接口单元（TPIU）
0x0000_2000~0x0000_2FFF	0x8000_2000~0x8000_2FFF	CoreSight 跟踪漏斗（CSTF）
0x0000_3000~0x0000_3FFF	0x8000_3000~0x8000_3FFF	系统交叉触发接口（SYS CTI）
0x0000_4000~0x0000_4FFF	0x8000_4000~0x8000_4FFF	串行线跟踪端口（SWO）
0x0000_5000~0x0000_5FFF	0x8000_5000~0x8000_5FFF	ITM0 复制器
0x0000_6000~0x0000_6FFF	0x8000_6000~0x8000_6FFF	ITM1 复制器
0x0000_7000~0x0000_7FFF	0x8000_7000~0x8000_7FFF	SWO 跟踪漏斗（SWTF）
0x0000_8000~0x0000_8FFF	0x8000_8000~0x8000_8FFF	全局时间戳生成器（TSG）
0x0000_9000~0x0000_9FFF	0x8000_9000~0x8000_9FFF	微控制器调试单元（DBGMCU）

25.5.2 软件 I/F

DAP

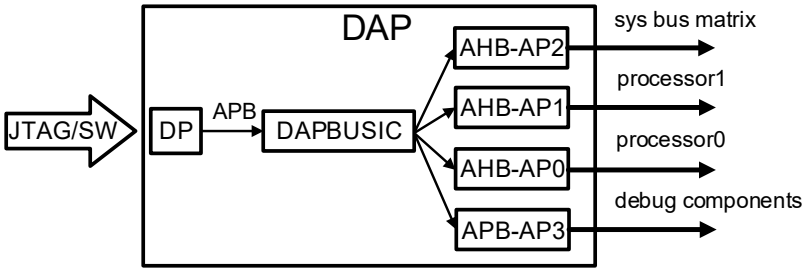
DAP（调试访问端口）如下图所示。由 SWJ-DP（串行线或 JTAG 调试端口）、DAPBUSIC（DAP 总线）、AHB-AP（AHB 接入端口）和 APB-AP（APB 接入端口）组成。

- AHB-AP0：core0 访问端口。允许访问调试系统中的外部调试组件和 core0 中的内部调试组件。
- AHB-AP1：core1 访问端口。允许访问调试系统中的外部调试组件和 core1 中的内部调试组件。
- AHB-AP2：总线矩阵访问端口。允许访问系统总线矩阵。
- APB-AP3：调试访问端口。允许访问外部调试组件。

所有访问端口采用 MEM-AP 类型，也就是说，调试组件被映射到相关联的调试总线的地址空间中。

有关 DAP 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.9 章“DAP 寄存器”。

图 35DAP

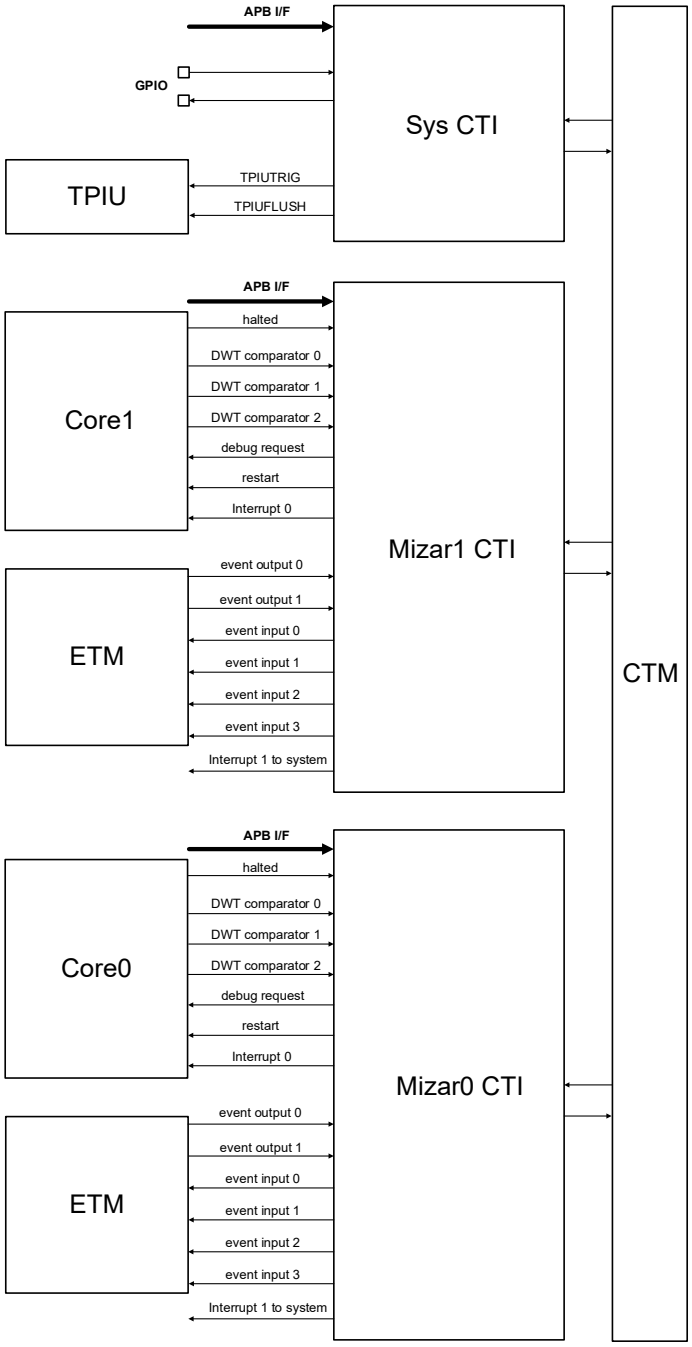


CTM 和 CTI

在下图中，交叉触发矩阵（CTM）和交叉触发接口（CTI）共同构成了 CoreSight 嵌入式交叉触发功能。这三个 CTI 通过 CTM 相互连接。例如，当任一 CPU 内核遇到断点时，也可以停止另一个内核。或者同步重新启动两个核心。

有关 SYS CTI 和 CTI 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.8 章“CTI 寄存器”和《Arm China 处理器技术参考手册》中的附录 B“交叉触发接口”。

图 36 SoC 中的嵌入式交叉触发器



CSTF 和 TPIU

CoreSight 跟踪漏斗（CSTF）将来自四个跟踪源的 ATB 总线合并为一个 ATB。CSTF 有四个 ATB 从端口和一个 ATB 主端口。判优器根据可编程优先级来选择从端口。

从端口的连接方式如下：

- EnS0：CPU0 ITM
- EnS1：CPU0 ETM

- EnS2：CPU1 ITM
- EnS3：CPU1 ETM

有关 CSTF 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.4 章“ATB 漏斗寄存器”。

TPIU 是一个 CoreSight 组件，用于格式化跟踪流并将其输出到外部跟踪端口信号上。TPIU 有一个 ATB 从端口，用于输入跟踪数据。跟踪端口为时钟输出 tpiu_traceclk。

有关 TPIU 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.7 章“TPIU 寄存器”。

SWTF 和 SWO

SWO 是 CPU 的 TPIU，它格式化来自处理器 ITM 的跟踪流，并将其输出到单线跟踪 SWO 上。

有关 SWO 的更多信息，请参阅《Arm China 处理器技术参考手册》中的附录 B“跟踪端口接口单元”。

必须对 SWO 跟踪漏斗（SWTF）进行编程，以便在启用跟踪之前选择哪个处理器 ITM 将使用它。SWTF 有两个 ATB 从端口和一个 ATB 主端口。

从端口的连接方式如下：

- EnS0：CPU0 ITM
- EnS1：CPU1 ITM

有关 SWTF 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.4 章“ATB 漏斗寄存器”。

复制器

ATB 复制器（REP0/REP1）将数据从单一 ATB 主设备同时传播到两台 ATB 从设备。

有关 ATB 复制器的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.5 章“ATB 复制器寄存器”。

TSG

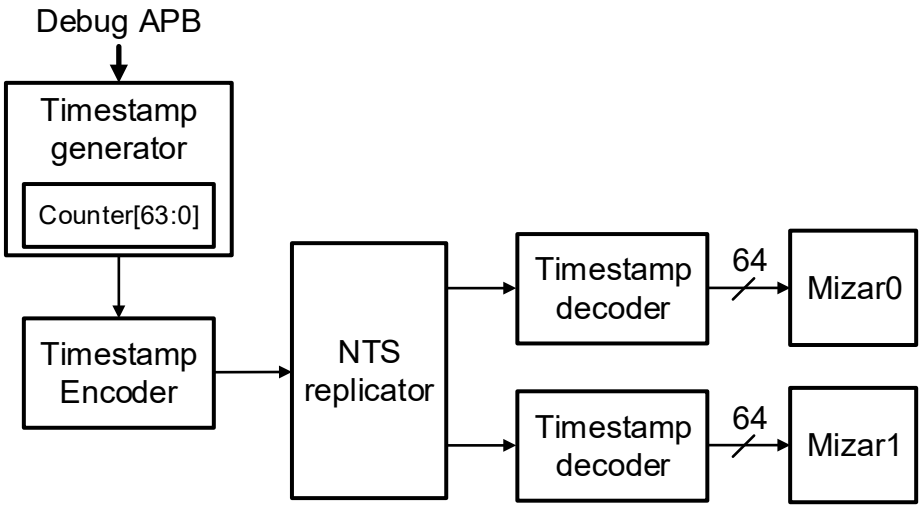
在下图中，全局时间戳生成器（TSG）包含一个 64 位计数器，该计数器为系统中的所有跟踪源（即每个处理器核心中的 ITM 和 ETM）提供公共时序参考。这些组件在跟踪流中插入时间戳，使跟踪分析器能够恢复跟踪数据包的时间顺序，当多个跟踪源在漏斗处复用为一条流时，跟踪数据包可能会丢失。

TSG 寄存器可通过 APB-AP 访问。这允许调试器或调试软件：

- 开始和停止时间戳递增
- 读取当前时间戳值

有关 TSG 的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册 r3p2》。

图 37 R501 SoC 中的全局时间戳分布



DBGMCU

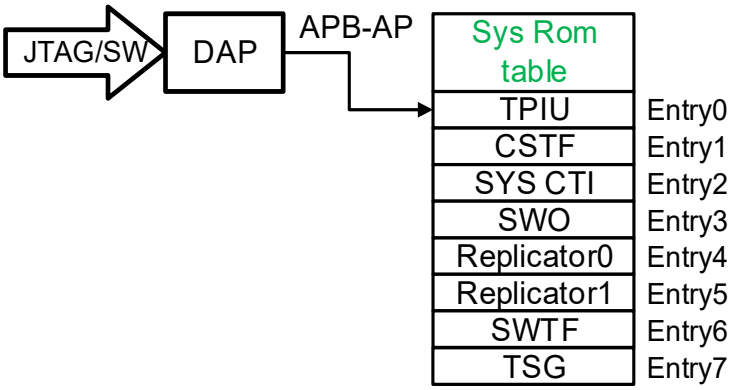
DBGMCU 组件包含许多寄存器，用于控制调试模式下的电源和时钟行为。调试器可以通过位于基地址 0x8000_9000 的 APB-AP 总线访问它们。它们也可由基址为 0xE005_9000 的钻孔处理器核心访问。

注意：DBGMCU 不是标准的 CoreSight 组件。因此，它不会出现在系统 ROM 表中。

系统 Rom 表

在下图中，APB-AP 总线上有一台 ROM 表。ROM 表是一个 CoreSight 组件，它包含 APB-AP 总线上所有 CoreSight 部件的基址。此表允许调试器自动发现 CoreSight 组件的拓扑结构。

图 38 系统 Rom 表



有关系统 Rom 表的更多信息，请参阅《Arm CoreSight SoC-400 技术参考手册》中的第 3.3 章“APB 互连寄存器”。

编程指南

有关编程的更多详细信息，请参阅：

- Arm CoreSight 架构规范 3.0 版
- Arm 调试接口架构规范 ADI 5.0 版至 ADI 5.2 版
- 《Arm CoreSight SoC-400 技术参考手册 r3p2》
- 《Arm China 处理器技术参考手册》

26 模拟子系统（AS）

26.1 简介

该器件的模拟模块包含模数转换器（ADC）、温度传感器、缓冲数模转换器（DAC）和比较器（COMP）。

26.2 主要特征

灵活的电压基准

（1） ADC 以 VREFHIx 和 VREFLOx 引脚为参考
- 内部电压参考范围为 0V~3.3V 或 0V~2.5V
- VREFHIx 引脚电压由外部驱动或内部基准参考电压产生

（2） DAC 的引脚参考
- 缓冲 DAC 以 VREFHIx 和 VSSA 为参考或者以 VDAC 引脚和 VSSA 为参考
- 比较器 DAC 以 VDDA 和 VSSA 为参考或者以 VDAC 引脚和 VSSA 为参考

灵活地使用引脚

（1） 内部连接到所有 ADC 上的 VREFLO，可用于偏置自校准

（2） 缓冲 DAC 输出，比较器输入和数字输入与 ADC 输入多路复用

26.3 结构框图

结构框图显示了不同集成模拟模块与模拟模块的输入/输出及参考引脚之间的连接。

模拟引脚与 COMP 一起构成了若干个模拟组。框图显示了每个模拟组连接的引脚，不显示从引脚到 ADC，DAC，COMP 模块。

VDAC 参考引脚用于为 DAC A，DAC B 和 COMP 内的 DAC 设置其它的范围（COMP 内 DAC 默认引用 VDDA 和 VSSA）。如果使用 VDAC 引脚提供参考电压，则该通道不能用作 ADC 输入。每个 COMP 或缓冲 DAC 的参考选择可通过模块的配置寄存器进行配置。

注意事项：

（1） 需要确定 VREFHI 和 VREFLO 的电压范围。

（2） VREFHI 引脚需要外接电容。

（3） 不是所有模拟引脚都适用于所有设备。

（4） 无论缓冲 DAC 的高参考电压是 VREFHIx 还是 VDAC，VSSA 都是低参考电压。

（5） 无论 COMP 的高参考电压是 VDAC 还是 VDDA，VSSA 都是低参考电压。

图 39 模拟子系统框图（100-pin）

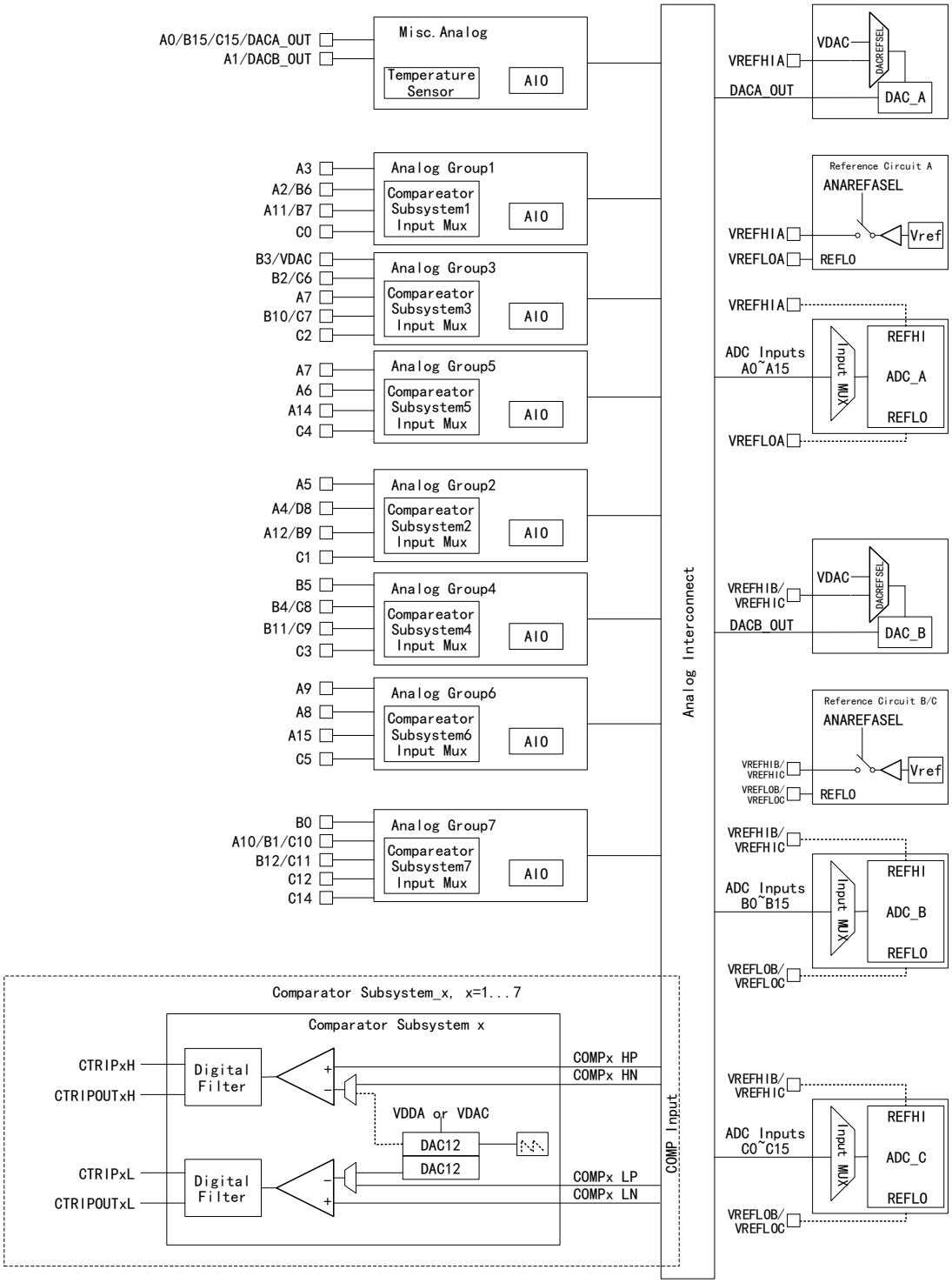


图 40 模拟子系统框图（80-pin）

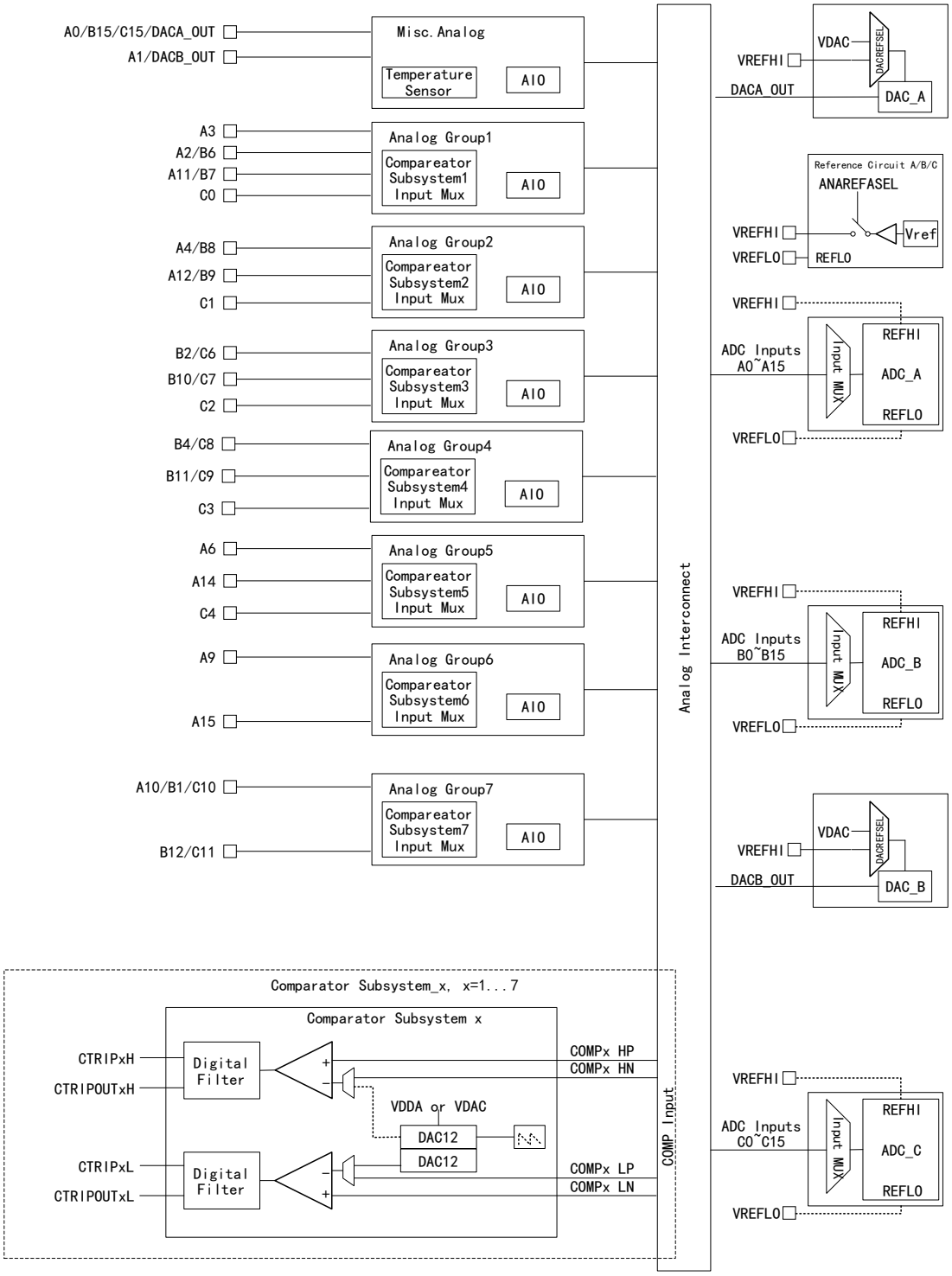


图 41 模拟子系统框图（64-pin）

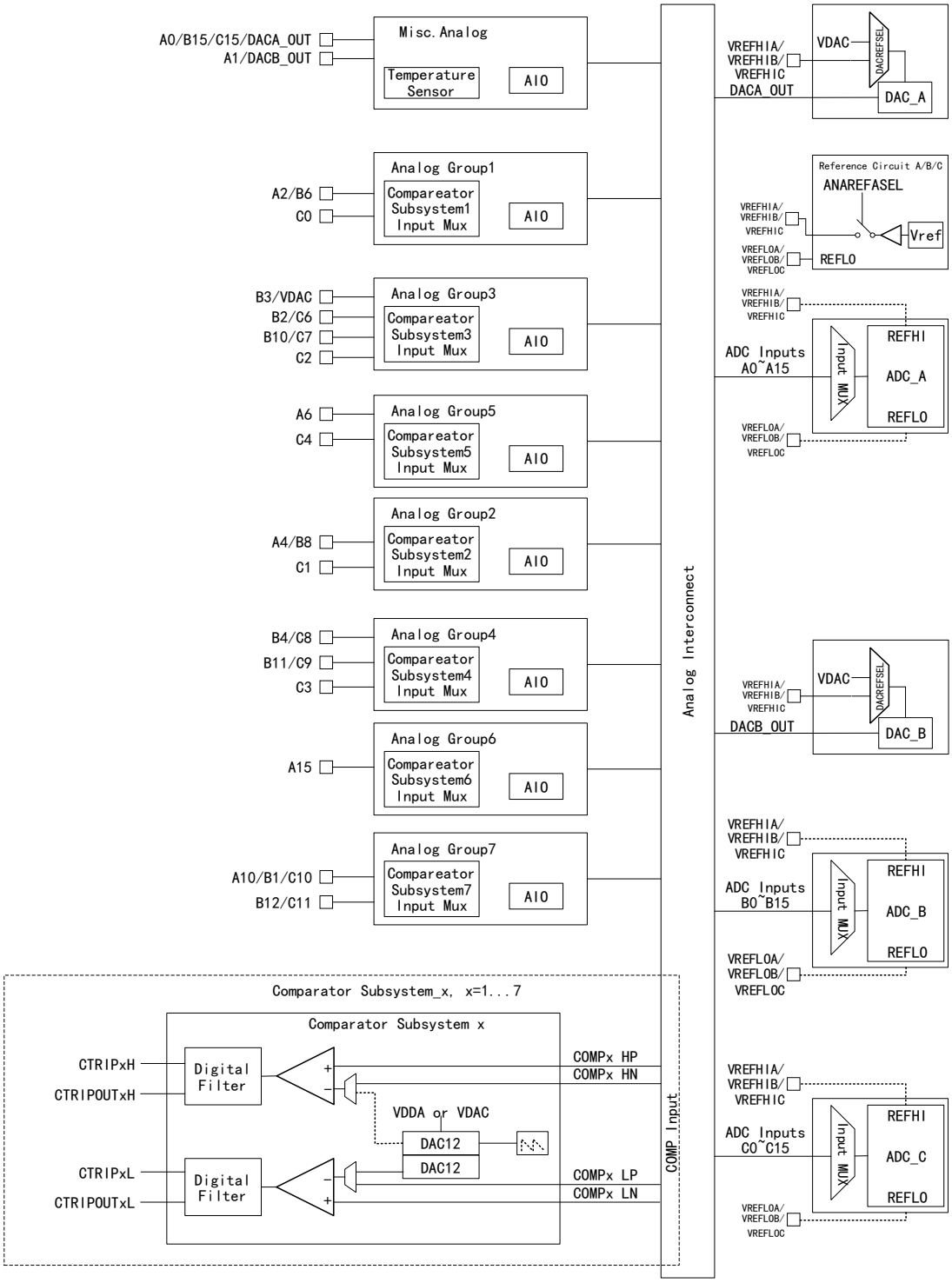


图 42 模拟子系统框图（56-pin）

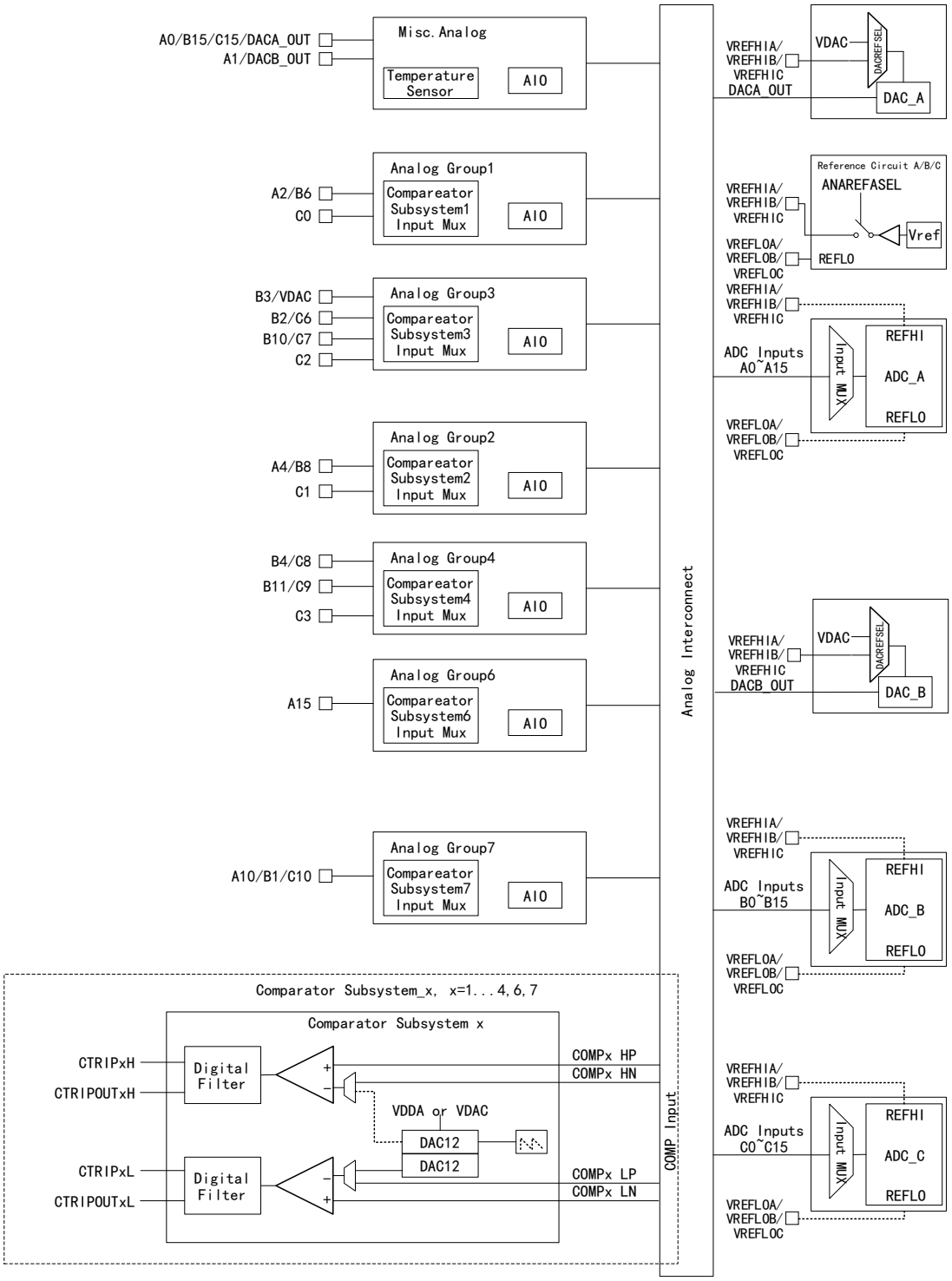
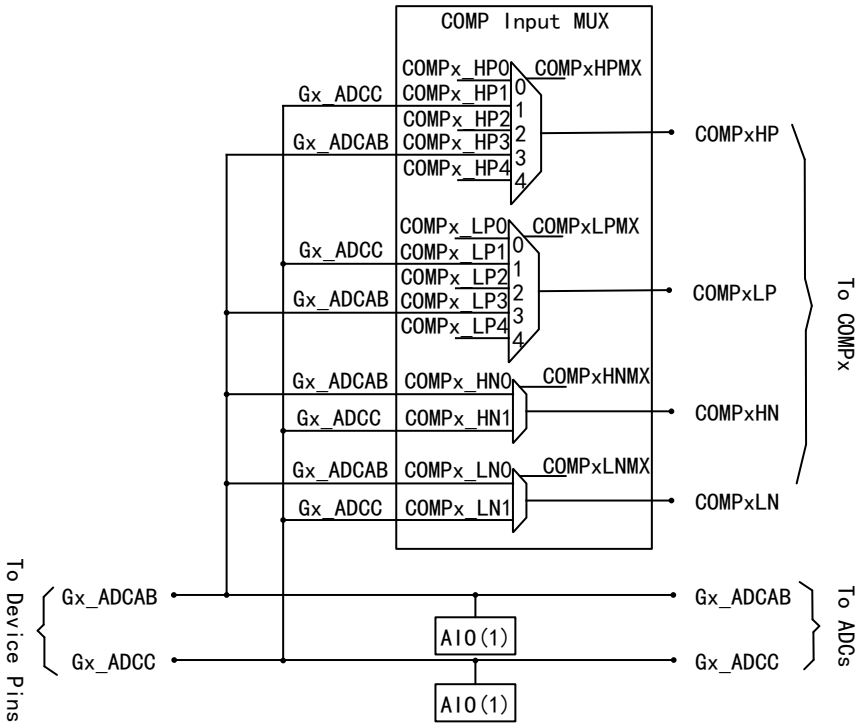


图 43 模拟组连接



注意：

（1） AIO 仅支持数字输入模式。

26.4 功能描述

26.4.1 模拟信号

表格 93 模拟信号

信号名称	描述
AIOx	可复用为数字量输入的模拟量输入口
Ax	ADCA 输入
Bx	ADCB 输入
Cx	ADCC 输入
VDAC	片上 DAC 的可选外部参考电压。在这个引脚上有一个 100pF 的 VSSA 电容，ADC 输入和 DAC 参考都不能禁用。如果用作片上 DAC 的参考，则在此引脚上放置至少 1μF 的电容器。
DACx_OUT	缓冲 DAC 输出
COMPx_DACH	比较器子系统高电平 DAC 输出
COMPx_DACL	比较器子系统低电平 DAC 输出
COMPx_HPy	比较器子系统高边比较器正输入
COMPx_HNy	比较器子系统高边比较器负输入

信号名称	描述
COMPx_LPy	比较器子系统低边比较器正输入
COMPx_LNy	比较器子系统低边比较器负输入
TempSensor	内部温度传感器

26.4.2 模拟引脚与内部连接

表格 94 模拟引脚与内部连接

引脚名称	组名称	封装				始终连接				比较器				AIO
		LQFP 100	LQFP 80	LQFP 64	QFN 56	ADC A	ADC B	ADC C	DAC	高正（HPMXSEL）	高负（HNMXSEL）	低正（LPMXSEL）	低负（LNMXSEL）	
VREFHIA	-	25	20(2) (VREFHI)	16(1)	14(1)									
VREFHIB	-	24(1)												
VREFHIC	-													
VREFLOA	-	27	21(2) (VREFLO)	17(1)	15(1)									
VREFLOB	-	26(1)												
VREFLOC	-													
模拟组 1										COMP1				
A3	G1_ADCAB	10	13			A3				3	0	3	0	AIO233
A2/B6		9	12	9	8	A2	B6			0		0		AIO224
A11/B7		18	16			A11	B7			2		2		AIO248
										4		4		
C0	G1_ADCC	19	16	12	10			C0		1	1	1	1	AIO237
模拟组 2										COMP2				

引脚名称	组名称	封装				始终连接				比较器				AIO
		LQFP 100	LQFP 80	LQFP 64	QFN 56	ADC A	ADC B	ADC C	DAC	高正（HPMXSEL）	高负（HNMXSEL）	低正（LPMXSEL）	低负（LNMXSEL）	
A5	G2_ADCAB	35				A5				3	0	3	0	AIO234
A4/B8		36	27	23	21	A4	B8			0		0		AIO225
A12/B9		30	22			A12	B9			2		2		AIO249
										4		4		
C1	G2_ADCC	29	22	18	16			C1		1	1	1	1	AIO238
模拟组 3										COMP3				
B3/VDAC	G3_ADCAB	8		8	7		B3		VDAC	3	0	3	0	AIO242
B2/C6		7	11	7	6		B2	C6		0		0		AIO226
A7		20				A7				2		2		AIO235
B10/C7		15	14	10	9		B10	C7		4		4		AIO250
C2	G3_ADCC	21	17	13	11			C2		1	1	1	1	AIO244
模拟组 4										COMP4				

引脚名称	组名称	封装				始终连接				比较器				AIO
		LQFP 100	LQFP 80	LQFP 64	QFN 56	ADC A	ADC B	ADC C	DAC	高正（HPMXSEL）	高负（HNMXSEL）	低正（LPMXSEL）	低负（LNMXSEL）	
B5	G4_ADCAB	42					B5			3	0	3	0	AIO243
B4/C8		39	28	24	22		B4	C8		0		0		AIO227
B11/C9		13	13	10	9		B11	C9		4		4		AIO251
C3	G4_ADCC	31	23	19	17			C3		1	1	1	1	AIO245
										2		2		
模拟组 5										COMP5				
A7	G5_ADCAB	20				A7				3	0	3	0	AIO235
A6		6	10	6		A6				0		0		AIO228
A14		16	15			A14				4		4		AIO252
										2		2		
C4	G5_ADCC	17	15	11				C4		1	1	1	1	AIO239
模拟组 6										COMP6				
A9	G6_ADCAB	38	28			A9				3	0	3	0	AIO236

引脚名称	组名称	封装				始终连接				比较器				AIO
		LQFP 100	LQFP 80	LQFP 64	QFN 56	ADC A	ADC B	ADC C	DAC	高正（HPMXSEL）	高负（HNMXSEL）	低正（LPMXSEL）	低负（LNMXSEL）	
A8		37				A8				0		0		AIO229
A15		14	14	10	15	A15				4		4		AIO253
C5	G6_ADCC	28						C5		1	1	1	1	AIO240
										2		2		
模拟组 7										COMP7				
B0	G7_ADCAB	41					B0			3	0	3	0	AIO241
A10/B1/C10		40	29	25	23	A10	B1	C10		0		0		AIO230
B12/C11		32	24	20	18		B12	C11		4		4		AIO254
C12	G7_ADCC	43						C12		2		2		AIO247
C14		44						C14		1	1	1	1	AIO246
其他模拟														
A0/B15/C15/DACA_OUT		23	19	15	13	A0	B15	C15	DACA_OUT					AIO231

引脚名称	组名称	封装				始终连接				比较器				AIO
		LQFP 100	LQFP 80	LQFP 64	QFN 56	ADC A	ADC B	ADC C	DAC	高正（HPMXSEL）	高负（HNMXSEL）	低正（LPMXSEL）	低负（LNMXSEL）	
A1/DACB_OUT		22	18	14	12	A1			DACB_OUT					AIO232
	TempSensor(3)						B14							

注意：（1）经过封装后，对于 LQFP100，VREFHIB 与 VREFHIC 是同一个端口，而 VREFLOB 与 VREFLOC 是同一个端口；对于 LQFP64 和 QFN56，VREFHIA、VREFHIB 与 VREFHIC 是同一个端口，而 VREFLOA、VREFLOB 与 VREFLOC 是同一个端口。

（2）对于 LQFP80，第 20 引脚是 VREFHI，第 21 引脚是 VREFLO。

（3）TempSensor 仅限内部连接;没有到达设备引脚。

26.4.3 优化上电时间

ADC 和 DAC 共享同一个参考电路。当使用一个或多个模块的应用程序时，共享参考电路可以优化上电时间。如果共享参考电路的其中一个模块已经在内部参考模式下初始化，后续模块可以从要求最小上电时间中减去参考上电时间来优化后续模块整个上电时间。

在切换 0V~2.5 V 和 0V~3.3 V 时，内部参考模式的上电也有等待时间，相关的等待时间值参见数据手册。

26.4.4 可复用为数字量输入的模拟量输入口（AIO）

AIO 端口 H 上的 GPIO，仅具有数字输入功能且只能在输入模式下工作，与模拟引脚复用。这些引脚一般作为模拟引脚，GPIO 处于高阻态。GPHAMSEL 寄存器可以用于配置这些引脚的数字或模拟操作。

如果相邻通道用于模拟功能，应该限制连接到 AIO 的信号的压摆率。否则将具有尖锐边沿的数字信号连接到 AIO，可能会与相邻的模拟信号发生串扰。

26.4.5 可复用为数字量输入的模拟量输入/输出（AGPIO）

AGPIO 是一些与模拟引脚复用的 GPIO，并具有完整的数字输入和输出功能。

AGPIOs 默认未连接，必须进行配置。设置模拟子系统 GPXAMSEL 寄存器可以使能模拟功能。设置通用输入/输出（GPIO）GPXAMSEL 寄存器可以使能数字功能。

表格 95 AGPIOs 配置

AGPIOCTRLx.GPIOy (Default = 0)	GPxAMSEL.GPIOy (Default = 1)	引脚连接	
		GPIOy	ADC
0	0	√	-
0	1	一般没有信号与 AGPIO 引脚连接。必须为引脚功能选择表中的其他一行	
1	0	√	-
1	1	-	√

如果相邻通道用于模拟功能，应该限制连接到 AGPIOs 的信号的边缘速率。否则将具有尖锐边沿的数字信号连接到 AGPIOs，可能会与相邻的模拟信号发生串扰。

26.5 寄存器组地址

表格 96 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
ASRegs	AS_REGS	0x5002_8000	0x5002_83FF

26.6 寄存器地址映射

表格 97 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
TSNSCTL	温度传感器控制寄存器	0xC0	-
ANAREFCTL	模拟参考控制寄存器	0xD0	-
VMONCTL	电压监视器控制寄存器	0xE0	-
CMPHPMXSEL	COMP_HP 选择源寄存器	0x104	-
CMPLPMXSEL	COMP_LP 选择源寄存器	0x108	-
CMPHNMXSEL	COMP_HN 选择源寄存器	0x10C	-
CMPLNMXSEL	COMP_LN 选择源寄存器	0x10E	-
ADCDACLOOPBACK	环回寄存器	0x110	-
LOCK	锁定寄存器	0x11C	-

26.7 寄存器功能描述

26.7.1 温度传感器控制寄存器(TSNSCTL)

偏移地址：0xC0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ENABLE	R/W	使能温度传感器（Temperature Sensor Enable） 0：禁止 1：使能	0h
15:1	保留			0h

26.7.2 模拟参考控制寄存器(ANAREFCTL)

偏移地址：0xD0

复位类型：XRSn

位/域	名称	R/W	描述	复位值
0	ANAREFASEL	R/W	选择模拟参考 A 模式（Analog reference A mode select） 如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终相同。 0：内部参考模式 1：外部参考模式	1h
1	ANAREFBSEL	R/W	选择模拟参考 B 模式（Analog reference B mode select） 如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终相同。 0：内部参考模式	1h

位/域	名称	R/W	描述	复位值
			1：外部参考模式	
2	ANAREFCSEL	R/W	选择模拟参考 C 模式（Analog reference C mode select） 如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终相同。 0：内部参考模式 1：外部参考模式	1h
7:3	保留			1h
8	ANAREFA2P5SEL	R/W	选择模拟参考 A 2.5V 源（Analog referenc A 2.5V source select） 在内部参考模式下，该位选择内部参考缓冲区驱动到 VREFHI 引脚上的电压。 0：内部 1.65V 参考模式 1：内部 2.5V 参考模式 可以选择参考范围,配置 2.5V 参考模式时产生 0~2.5V 的参考范围，配置 1.65V 参考模式时产生 0~3.3V 的参考范围。当进行切换模式时，必须留出足够的时间让外部电容器充电到新的电压在才能在使用 ADC 或缓冲 DAC。如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终保持相同设置。	0h
9	ANAREFB2P5SEL	R/W	选择模拟参考 B 2.5V 源（Analog referenc B 2.5V source select） 在内部参考模式下，该位选择内部参考缓冲区驱动到 VREFHI 引脚上的电压。 0：内部 1.65V 参考模式 1：内部 2.5V 参考模式 可以选择参考范围,配置 2.5V 参考模式时产生 0~2.5V 的参考范围，配置 1.65V 参考模式时产生 0~3.3V 的参考范围。当切换模式时，必须留出足够的时间让外部电容器充电到新的电压才能使用 ADC 或缓冲 DAC。如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终保持相同设置。	0h
10	ANAREFC2P5SEL	R/W	选择模拟参考 C 2.5V 源（Analog referenc C 2.5V source select） 在内部参考模式下，该位选择内部参考缓冲区驱动到 VREFHI 引脚上的电压。 0：内部 1.65V 参考模式 1：内部 2.5V 参考模式 可以选择参考范围,配置 2.5V 参考模式时产生 0~2.5V 的参考范围，配置 1.65V 参考模式时产生 0~3.3V 的参考范围。当切换模式时，必须留出足够的时间让外部电容器充电到新的电压才能使用 ADC 或缓冲 DAC。	0h

位/域	名称	R/W	描述	复位值
			如果多个 VREFHI 引脚连接在一起，则连接引脚的参考电压始终保持相同设置。	
15:11	保留			0h

26.7.3 电压监视器控制寄存器(VMONCTL)

偏移地址：0xE0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	保留			0h
8	BORLVMONDIS	R/W	在 VDDIO 上禁用 BORL（BORL disable on VDDIO） 当 VDDIO 低于 VDDIO 的下 BOR 阈值时是否会触发 BOR 电路。 0：使能，会触发 1：禁止，不会触发	0h
15:9	保留			0h

26.7.4 COMP_HP 选择源寄存器(CMPHPMXSEL)

偏移地址：0x104

复位类型：XRSn

位/域	名称	R/W	描述	复位值
2:0	CMP1HPMXSEL	R/W	在 COMP1_HP 输入上选择源（Select source on the COMP1_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
5:3	CMP2HPMXSEL	R/W	在 COMP2_HP 输入上选择源（Select source on the COMP2_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
8:6	CMP3HPMXSEL	R/W	在 COMP3_HP 输入上选择源（Select source on the COMP3_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
11:9	CMP4HPMXSEL	R/W	在 COMP4_HP 输入上选择源（Select source on the COMP4_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
14:12	CMP5HPMXSEL	R/W	在 COMP5_HP 输入上选择源（Select source on the COMP5_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h

位/域	名称	R/W	描述	复位值
15	保留			0h
18:16	CMP6HPMXSEL	R/W	在 COMP6_HP 输入上选择源（Select source on the COMP6_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
21:19	CMP7HPMXSEL	R/W	在 COMP7_HP 输入上选择源（Select source on the COMP7_HP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
31:22	保留			0h

26.7.5 COMP_LP 选择源寄存器(CMPLPMXSEL)

偏移地址：0x108

复位类型：XRSn

位/域	名称	R/W	描述	复位值
2:0	CMP1LPMXSEL	R/W	在 COMP1_LP 输入上选择源（Select source on the COMP1_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
5:3	CMP2LPMXSEL	R/W	在 COMP2_LP 输入上选择源（Select source on the COMP2_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
8:6	CMP3LPMXSEL	R/W	在 COMP3_LP 输入上选择源（Select source on the COMP3_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
11:9	CMP4LPMXSEL	R/W	在 COMP4_LP 输入上选择源（Select source on the COMP4_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
14:12	CMP5LPMXSEL	R/W	在 COMP5_LP 输入上选择源（Select source on the COMP5_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
15	保留			0h
18:16	CMP6LPMXSEL	R/W	在 COMP6_LP 输入上选择源（Select source on the COMP6_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h

位/域	名称	R/W	描述	复位值
21:19	CMP7LPMXSEL	R/W	在 COMP7_LP 输入上选择源（Select source on the COMP7_LP input） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。取值范围为 0 ~ 4，其余保留。	0h
31:22	保留			0h

26.7.6 COMP_HN 选择源寄存器(CMPHNMXSEL)

偏移地址：0x10C

复位类型：XRSn

位/域	名称	R/W	描述	复位值
x-1	CMPxHNMXSEL	R/W	在 COMPx_HN 输入上选择源（Select source on the COMPx_HN input）（x=1-7） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。	0h
15:7	保留			0h

26.7.7 COMP_LN 选择源寄存器(CMPLNMXSEL)

偏移地址：0x10E

复位类型：XRSn

位/域	名称	R/W	描述	复位值
x-1	CMPxLNMXSEL	R/W	在 COMPx_LN 输入上选择源（Select source on the COMPx_LN input）（x=1-7） COMP 输入复用器的具体连接请参考模拟引脚和内部连接表，一般结构请参考模拟组连接图。	0h
15:7	保留			0h

26.7.8 环回寄存器(ADCDACLOOPBACK)

偏移地址：0x110

复位类型：KEY 位为 SYSRSn，其余为 XRSn

位/域	名称	R/W	描述	复位值
0	ENLB2ADCA	R/W	环回 COMPDACA 输出到 ADCA（Loopback COMPDACA output to ADCA） 设置此位时，将覆盖 ADC 的 CHANSEL。 0：环回断开 1：环回 COMPDACA 输出给 ADCA。	0h
1	ENLB2ADCB	R/W	环回 COMPDACA 输出到 ADCB（Loopback COMPDACA output to ADCB） 设置此位时，将覆盖 ADC 的 CHANSEL。 0：环回断开 1：环回 COMPDACA 输出给 ADCB。	0h
2	ENLB2ADCC	R/W	环回 COMPDACA 输出到 ADCC（Loopback COMPDACA output to ADCC） 设置此位时，将覆盖 ADC 的 CHANSEL。 0：环回断开 1：环回 COMPDACA 输出给 ADCC。	0h

位/域	名称	R/W	描述	复位值
15:3	保留			0h
31:16	KEY	R-0/W	关键字（Key） 必须在此位包含值 0xA5A5 才能写入此寄存器。否则寄存器将保持在写入尝试之前的状态。读操作返回 0。	0h

26.7.9 锁定寄存器(LOCK)

偏移地址：0x11C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TSNSCTL	R/ WSonce	锁定 TSNSCTL 寄存器（TSNSCTL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
1	ANAREFCTL	R/ WSonce	锁定 ANAREFCTL 寄存器（ANAREFCTL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
2	VMONCTL	R/ WSonce	锁定 VMONCTL 寄存器（VMONCTL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
4:3	保留			0h
5	CMPHPMXSEL	R/ WSonce	锁定 CMPHPMXSEL 寄存器（CMPHPMXSEL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
6	CMPLPMXSEL	R/ WSonce	锁定 CMPLPMXSEL 寄存器（CMPLPMXSEL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
7	CMPHNMXSEL	R/ WSonce	锁定 CMPHNMXSEL 寄存器（CMPHNMXSEL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
8	CMPLNMXSEL	R/ WSonce	锁定 CMPLNMXSEL 寄存器（CMPLNMXSEL Register Lock） 此位控制相应寄存器的写入，置 1 将禁止其写入。通过复位可以清除此位。	0h
31:9	保留			0h

27 模数转换器（ADC）

27.1 术语全称、缩写描述

表格 98 术语全称、缩写描述

中文全称	英文全称	英文缩写
多路复用器	Multiplexer	MUX
采样保持	Sample and hold	S/H
ADC 转换开始	ADC Start of Conversion	ADCSOC

27.2 简介

12 位精度的 ADC 包含一个封装器和一个内核。封装器基于转换开始(SOC)，由配置和控制 ADC 的数字电路组成，这些电路包括结果寄存器、可编程转换逻辑、模拟电路接口、外设总线接口、后处理电路和其他片上模块接口。内核由模拟电路组成，模拟电路包括采样保持(S/H)电路、逐次逼近电路、通道选择 MUX、参考电压电路和其他模拟支持电路。每个 ADC 模块由一个采样保持(s/h)电路组成。允许多个 ADC 同时采样或独立操作。

27.3 主要特征

（1） ADC 内部参考电压选项：2.5V 或 3.3V

（2） 单端转换模式

（3） 12 位精度的分辨率

（4） 由 VREFHI 和 VREFLO 引脚设置外部参考

（5） 突发模式

（6） 16 个结果寄存器，可单独寻址

（7） 16 个通道的输入多路复用器

（8） 16 个可配置的 SOC

（9） 中断
- 4 个 NVIC 中断
- 可配置的中断布局

（10） 触发源
- 软件立即启动
- TMR0/1/2
- ADCINT1/2
- GPIO ADCEXTSOC

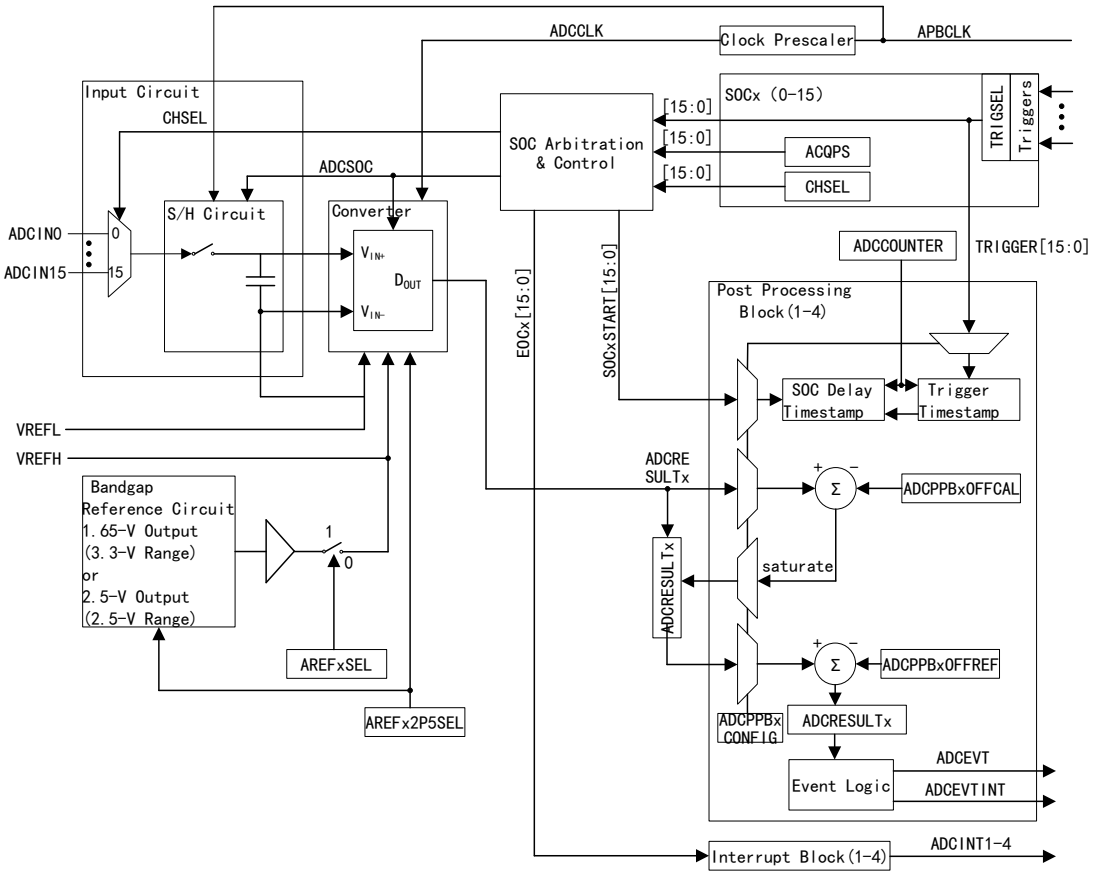
- 所有 PWM - ADCSOC A 或 B

（11） 四个后处理块都具有：

- 饱和偏移校准
- 触发器到采样延迟捕获
- 设定值计算错误
- 高、低、过零比较可触发中断或 PWM

27.4 结构框图

图 44 结构框图



27.5 功能描述

27.5.1 可配置性

ADC 配置分为由 SOC 单独控制和由每个 ADC 模块全局控制，具体如下：

表格 99ADC 选项和配置

选项	可配置性
分辨率	不可配置（仅 12 位）
信号模式	不可配置（仅单端）

选项	可配置性
转换通道	SOC 级
触发源	SOC 级 (1)
采样窗口持续时间	
EOC 位置	模块级
突发模式	模块级 (1)
时钟	
参考电压源	模块级（外部或内部）(2)

注意：

（1） 在这些配置中，向不同的 ADC 模块写入不同的值，可能导致 ADC 非同步工作。

（2） 低引脚数封装可能在多个 ADC 之间共享一个 VREFHI 引脚。此时共享参考引脚的 ADC 的必须配置为相同的参考模式。

27.5.2 时钟配置

APBCLK 时钟提供基本 ADC 时钟，可用于生成 ADC 采样窗口。ADCCLK 用于对转换器进行时钟处理，由 ADCCTL2 [PRESCALE]决定。内核需要大约 10.5 个 ADCCLK 周期来将电压处理成转换结果。用户需要确定所需的采样窗口的持续时间。

27.5.3 信号模式

ADC 支持单端信号。在单端模式下，转换器的输入电压通过一个引脚 ADCINx 采样，参考 VREFLO。

27.5.4 参考电压

外部参考模式

每个 ADC 都有一个 VREFHI 输入和一个 VREFLO 输入。在外部参考模式下，为了确定 ADC 转换输入范围，这两个引脚需要用作比率参考。外部参考模式需要在 VREFHI 引脚上连接一个外部电容。如果没有外部 VREFLO 信号，VREFLO 在内部连接到器件模拟地 VSSA。

内部参考模式

在内部参考模式下，设备驱动电压到 VREFHI 引脚上。VREFHI 和 VREFLO 引脚设置 ADC 转换范围，可将内部参考电压配置为 2.5V 或 1.65V。配置为 1.65V 时，有效 ADC 转换输入范围为 VREFLO 至 3.3V。内部参考模式还需要 VREFHI 引脚上的外部电容。

成组参考

内部设备硬件可以确保多个引用不会将冲突电压驱动到同一引脚上，所以可以随

意配置顺序和时间。在部分封装中，多个 ADC 的电压参考引脚可能会组合在一起。此时选择外部和内部参考模式及选择 2.5V 或 3.3V 的内部参考电压范围时，需要配置相同的组合参考。如果 ADC B 和 ADC C 参考引脚组合，所需参考模式为 2.5V 内部参考模式。

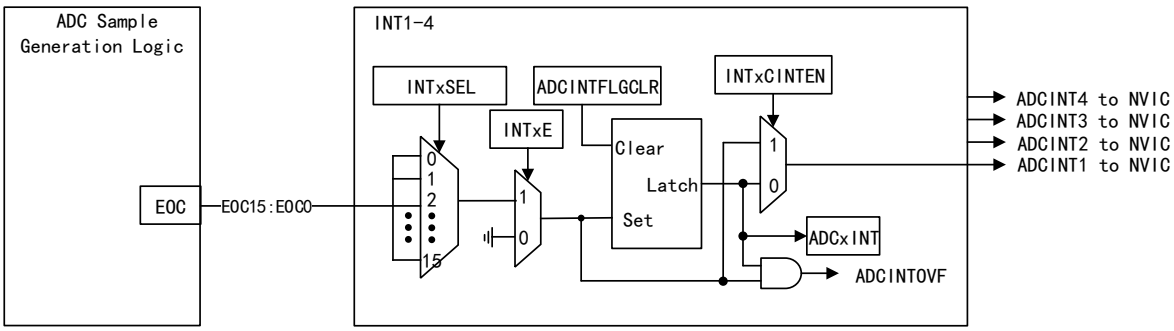
选择参考模式

参考电压模式需要使用 SetVREF 或 ADC_setVREF 函数配置。使用其中任意一个函数都可以被加载到 ADCOFFTRIM 寄存器中。不能直接写入 ANAREFCTL 寄存器来配置。此函数必须在设备复位后至少调用一次。

27.5.5 EOC 和中断

每个 SOC 都有相应的转换结束（EOC）信号。EOC 信号可用于触发 ADC 中断。ADCCTL1 [INTPULSEPOS]决定在采样窗口的末端或电压转换的末端产生 EOC 脉冲。每个 ADC 模块有 4 个可配置的 ADC 中断。这些中断可以由 16 个 EOC 信号任意触发。可以直接读取 ADCINTx 的标志位，以确定相关的 SOC 是否完成或中断是否可以传递给 NVIC。ADCCTL1 [ADCBSY]清零表示 ADC 已准备好处理下一次转换。如果要确定 SOC 序列是否完整，需要将 ADCINTx 标志链接到序列中的最后一个 SOC 并监控该标志。

图 45ADC 中断结构框图



中断溢出

中断溢出发生在 EOC 信号在 ADCINTFLG 寄存器中设置已经被设置的标志时。一般情况下，溢出中断不会传递给 NVIC 模块。当 ADCINTFLG 寄存器中的标志溢出时，将设置 ADCINTOVF 寄存器中的相应标志，此标志仅用于检测是否发生溢出。当可能发生 ADC 中断溢出时，应用程序需要检查 ISR 内或后台循环中适当的溢出标志，并在检测到溢出时做出相应行动。

继续中断模式

默认此模式禁用，且不会向 NVIC 发出额外的重叠中断。INTxCONT 位决定当 ADCINTFLG 未从先前中断中清除时如何处理中断。通过激活此模式，ADC 中断将始终到达 NVIC。如果在 ADCINTFLG 置位时发生中断，无论如何配置，ADCINTOVF 仍然被设置。

早期中断配置模式

DELAY 决定 ADC 中断进入早期中断模式的时间。早期中断模式允许应用程序在 ADC 结果就绪前进入 ADC 中断服务程序。这可以使应用程序进行前期工作，使其可以在 ADC 结果可用时立即对其进行操作。但是如果早期中断的时机太早，应用程序则需要等到更新的 ADC 结果可用。

- 将 INTPULSEPOS 清除可以使 ADC 处于早期中断模式，这时才能使用可配置的中断时间
- 当 INTPULSEPOS 设置为 1 时，写入 DELAY 将不会对中断产生影响
- DELAY 值决定在 ADCINTx 标志被设置之前 SOC 脉冲下降沿之后额外的 APBCLK 周期数
- 如果 DELAY 值超过 EOC，将同时产生 ADC 中断与 EOC

27.5.6 分辨率

支持 12 位分辨率。分辨率决定了模拟范围被量化为数字值的精细程度。

27.5.7 预期转换结果

基于给定的模拟输入电压，模数转换公式如下：

- 当 ADCINx≤VREFL 时，ADC_RULTx = 0
- 当 VREFLO＜ADCINx＜VREFHI 时，$\mathrm{ADC_RULTx} = \frac{4096(\mathrm{ADCINx}-\mathrm{VREFLO})}{\mathrm{VREFHI}-\mathrm{VREFLO}}$
- 当 ADCINx≥VREFHI 时，ADC_RULTx =4095

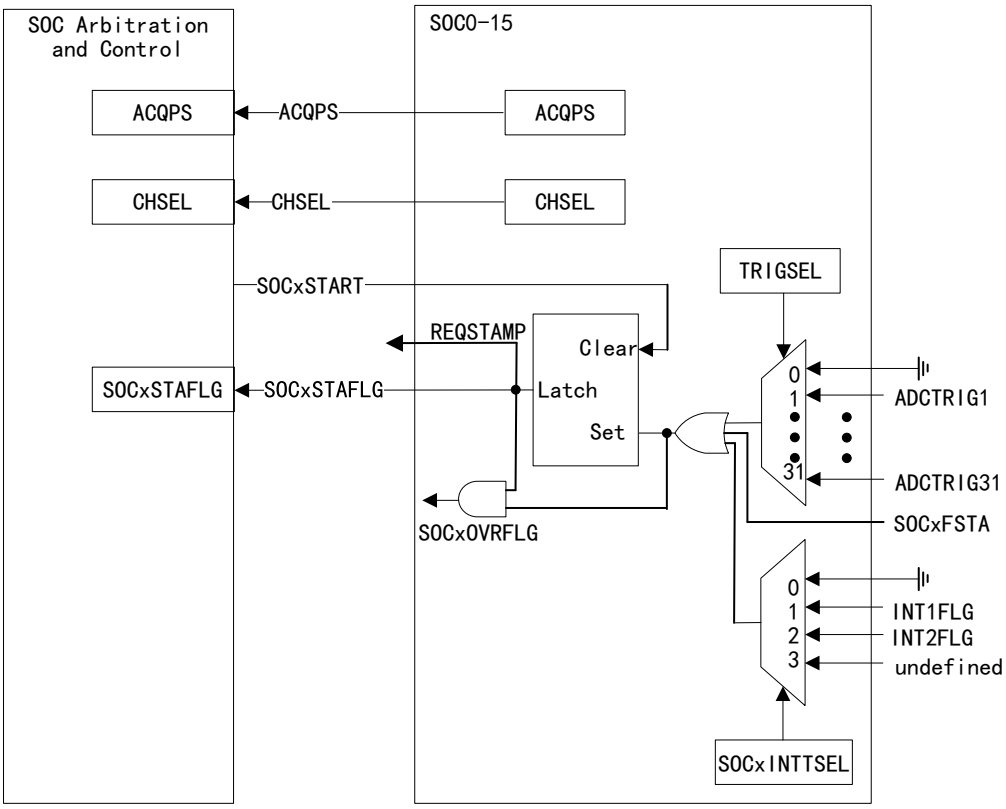
27.5.8 解释转换结果

根据给定的 ADC 转换结果，对应模拟输入如下。这对应于可能产生此转换结果的模拟电压范围的中心。

- 当 ADCRESULTx=0 时，ADCINx≤VREFLO
- 当 0＜ADCRESULTx＜4095 时，$\mathrm{ADCINx} = (\mathrm{VREFHI} - \mathrm{VREFLO})(\frac{ADC_RULTx}{4096}) + \mathrm{VREFLO}$
- 当 ADCRESULTx=4095 时，ADCINx≥VREFHI

27.5.9 SOC

图 46SOC 结构框图



工作原理

ADC 通过可配置的 SOC 完成触发和转换顺序。每个 SOC 是一个配置集，决定单个通道的单次转换。该集合中有三种配置：分别为采样窗口持续时间、要转换的通道和启动转换的触发源。

在接收到为 SOC 配置的触发器后，包装器将确保使用指定的采样窗口持续时间捕获指定的通道。可以根据需要为相同的采样窗口、通道和/或触发器配置多个 SOC：

- 配置多个 SOC 使用相同的通道和触发器将允许过采样
- 配置多个 SOC 使用相同的触发器将允许触发器生成一系列转换

SOC 配置

每个 SOC 都有单独的控制寄存器，可以为 SOCx 配置触发源、要转换的通道和采样窗口持续时间。

触发

每个 SOC 可以配置为在任意输入触发器上启动。由 ADCSOCxCTL [TRIGSEL] 选择主要触发器。具体选择如下：

- 在软件中禁用

- TMR0/1/2
- ADCINT1/2
- GPIO：输入 X-Bar INPUT5
- 所有 PWM - ADCSOC A 或 B

ADC 采样窗口

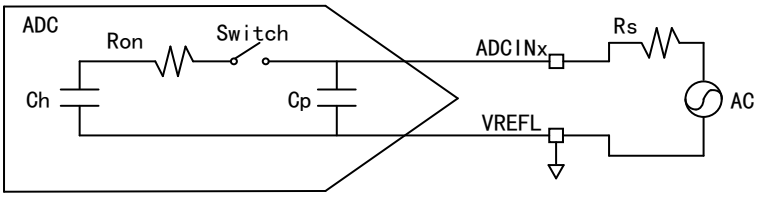
每个外部信号源驱动模拟信号的能力不同。信号源需要对 ADC 内核中的采样电容充电至信号电压的 0.5 LSBs 以内，才能达到额定分辨率。由 ADCSOCxCTL [ACQPS]配置采样窗口，采样窗口是采样电容器允许充电的时间量，可以设置为 0 到 511 之间的值，所以采样窗口持续时间为:采样窗口= (ACQPS + 1) x (APB 时钟周期时间)。

由此可见采样窗口持续时间基于 APBCLK。选择的采样窗口持续时间必须至少与一个 ADCCLK 周期一样长。

ADC 输入模型

在单端操作中，输入模型需要与实际信号源阻抗一起使用，才能确定采样窗口持续时间。

图 47 单端输入模型



通道选择

每个 SOC 都可以配置为转换任何 ADC 通道。由 ADCSOCxCTL [CHSEL]选择。

表格 100 选择输入引脚通道

输入模式	CHSEL	输入
单端型	0	ADCIN0
	1	ADCIN1
	2	ADCIN2
	3	ADCIN3
	4	ADCIN4
	5	ADCIN5
	6	ADCIN6
	7	ADCIN7
	8	ADCIN8

输入模式	CHSEL	输入
	9	ADCIN9
	10	ADCIN10
	11	ADCIN11
	12	ADCIN12
	13	ADCIN13
	14	ADCIN14
	15	ADCIN15

27.5.10 SOC 配置

PWM 触发器的单次转换

如果要配置 ADCA 在 PWM 定时器达到其周期匹配时在通道 ADCIN1 上执行单个转换，需要先配置 PWM3 产生 SOCA 或 SOCB 信号。可以使用 16 个 SOC 中的任意一个。如果 APBCLK 频率为 100MHz，需要 100 ns 采样窗口，则采样窗口持续时间应为 100 ns/10 ns = 10 个周期。所以 ACQPS 应该设置为 10 - 1 = 9。

当 PWM3 匹配其周期并产生 SOCB 信号时，如果 ADC 空闲，则立即开始采样通道 ADCINA1。如果 ADC 繁忙，则当通道获得优先级时 ADCINA1 才开始采样。步骤如下：

（1） ADC 控制逻辑将以指定的 100ns 采样窗口宽度对 ADCINA1 进行采样

（2） 采样完成后，ADC 立即开始将采样电压转换为数字值

（3） 转换完成后结果呈现在 ADCRESULT5 寄存器中

PWM 触发器的过采样转换

当 PWM3 匹配其周期并产生 SOCB 信号时，如果 ADC 空闲，则立即开始采样通道 ADCINA1。如果 ADC 繁忙，则当 SOC5 获得优先级时 ADCINA1 才开始采样。一旦 SOC5 的转换完成，SOC6 将开始转换 ADCINA1。一般情况下会按顺序转换，由 SOC5 最先转换，这取决于接收到 PWM 触发器时轮询指针的位置。转换结果与放置的寄存器的对应关系如下：

表格 101 转换结果与寄存器的对应关系

转换结果	寄存器
SOC5	ADCRESULT5
SOC6	ADCRESULT6
SOC7	ADCRESULT7
SOC8	ADCRESULT8

定时器触发器的多次转换

TMR2 用于产生触发器。要设计包含多个信号的采样方案，需要根据以下步骤：

（1） 列出信号及其所需的采样窗口

（2） 计算每个信号所需的 APBCLK 周期数

（3） 设置 ACQPS

（4） 确定将 ADC 引脚连接到每个信号

（5） 确定 CHSEL 值

表格 102 多信号采样

信号名称	采样窗口要求（ns）	采样窗口 APBCLK 周期=采样窗口要求/10ns（ns）	ACQPS 值=采样窗口 APBCLK 周期-1	ADC 引脚	CHSEL 值
信号 1	Min=240	24	23	ADCINA5	5
信号 2	Min=883.333	89	88	ADCINA0	0
信号 3	Min=220	22	21	ADCINA3	3
信号 4	Min=585	59	58	ADCINA2	2

当 TMR2 产生一个事件时，将依次对 SOC0、SOC1、SOC2、SOC3 进行采样和转换。一般情况下会按顺序转换，由 SOC0 最先转换，这取决于接收到定时器触发器时轮询指针的位置。转换结果与放置的寄存器的对应关系如下：

表格 103 转换结果与寄存器的对应关系

转换结果	寄存器
ACINA5	ADCRESULT0
ACINA0	ADCRESULT1
ACINA3	ADCRESULT2
ACINA2	ADCRESULT3

SOC 的软件触发

无论 SOC 是否被配置为接受特定的触发器，只要在 ADCSOCFRC1 寄存器中写入位，软件触发器都可以设置要转换的 SOC。

27.5.11 ADC 转换优先级

同时设置多个 SOC 标志时，需要决定它们的转换顺序。默认的优先级方法是轮询，由轮询指针决定。ADCSOCPRICTL [RRPOINTER]指向最后转换的 SOC。最高优先级的 SOC 被分配给下一个比 RRPOINTER 值大的值，在 SOC15 之后绕回 SOC0。当 RRPOINTER 等于 16 时，优先级最高的是 SOC0。重置时该值为 16，因为 0 表示已经发生了转换。当 ADC 被复位或将复位值写入 ADCSOCPRICTL 寄存器时，RRPOINTER 会被重置。通过写入和清除与 ADC 实例对应的 SOFTPRES 寄存器来复位 ADC。

ADCSOCPRICTL [SOCPRIORITY]可以从单个 SOC 分配高优先级到所有 SOC。被分配的 SOC 将在当前转换完成后将自己插入下一个转换。转换完成后，轮询继续在它被中断的地方运行。如果两个高优先级的 SOC 同时被触发，则低优先级的 SOC 将优先。高优先级模式首先分配给 SOC0，然后按数字顺序递增。写入 SOCPRIORITY 位的值定义了第一个优先级不高的 SOC。

27.5.12 突发模式

设置 ADCBURSTCTL [BURSTEN]位，可以将 ADC 包装器配置为突发模式。突发模式允许单个触发器一次通过一个或多个循环 SOC。仅适用于轮询的 SOC。ADCBURSTCTL [BURSTTRIGSEL]控制所有轮询 SOC 的触发。在接收突发触发器后，ADC 包装器设置（BURST_SIZE+ 1）个 SOC。基于轮询指针的最高优先级的 SOC 设置第一个 SOC，然后设置其他 SOC，直到设置了 BURST_SIZE 个 SOC。将 ADC 配置为突发模式时，需要在每个突发转换完成后才能接收下一个突发触发。

示例

所有信号都用 20 个 APBCLK 周期宽的采样窗口进行转换，每个 SOC 都可以配置不同的持续时间。突发模式可以在其他触发器上采样一组不同的信号。例如，ADCIN7 和 ADCIN5 在 TMR2 的第一个触发器和之后的每个触发器上转换。

当收到第一个 TMR2 触发器时，如果 ADC 空闲，则立即转换 SOC12 和 SOC13。如果 ADC 繁忙，当 SOC12 和 SOC13 的 SOC 获得优先级时才被转换。SOC13 完成后，轮询指针将给 SOC14 最高优先级。所以当收到下一个 TMR2 触发器时，SOC14 和 SOC15 被挂起和转换。后续触发器将继续在转换 SOC12 和 SOC13 以及转换 SOC14 和 SOC15 之间切换。转换结果与放置的寄存器的对应关系如下：

表格 104 转换结果与寄存器的对应关系

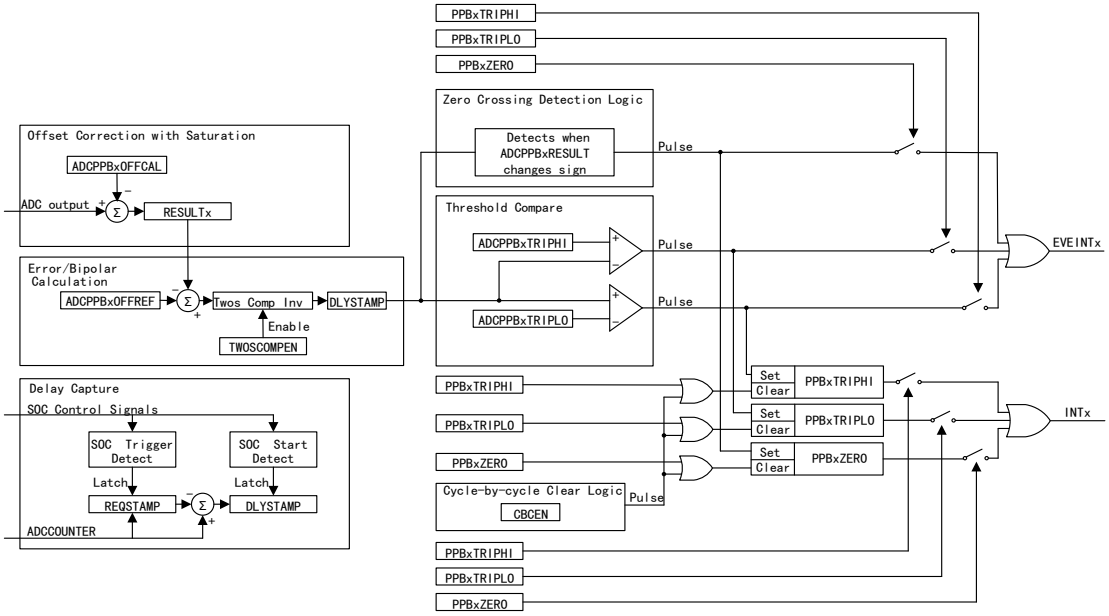
转换结果	寄存器
SOC12	ADCRESULT12
SOC13	ADCRESULT13
SOC14	ADCRESULT14
SOC15	ADCRESULT15

注意：虽然本示例只在两组转换之间切换，但可以使用类似的方法实现三个或更多不同的转换。

27.5.13 后处理模块

每个 ADC 包含四个后处理模块。这些块可以使用 ADCPPBxCONFIG 与 ADCRESULTx 寄存器中相关联。每个 PPB 可以同时去除与 ADCIN 通道相关的偏移量，标记零交叉点，减去参考值以及标记比较上限或下限。过零和比较标志还可以跳闸 PWM 和/或产生中断。PPB 还可以记录触发与 PPB 相关的 SOC 和实际开始采样之间的延迟时间。

图 48PPB 结构框图



PPB 偏移校正

外部传感器和信号源可能会产生一些偏移，这些偏移因通道而异，ADC 偏移的全局微调不足以补偿。为了在严格的控制循环中节省周期，后处理模块可以在零开销的情况下消除这些偏移。偏移校正需要先将 ADCPPBxCONFIG 指向所需的 SOC，然后将偏移校正值写入 ADCPPBxOFFCAL 寄存器才能完成。向 OFFCAL 写入 0 可以禁用偏移校正功能，将原始结果传递给 ADCRESULT 寄存器。后处理块自动从原始转换结果中添加或减去 OFFCAL 的值，并存储在 ADCRESULT 寄存器中。这个加法/减法将在高端饱和为 4095，在低端饱和为 0。

注意：可以将多个 PPB 指向相同的 SOC，此时实际应用的 ADCPPBxOFFCAL 值是具有最高数值的 PPB 的值。因为所有 PPB 默认都指向 SOC0，所以在 SOC0 上使用 PPB 时需要谨慎。否则编号较低的 PPB 可能被编号较高的 PPB 无意地覆盖偏移校正。

PPB 误差计算

一般情况下，可以使用双极信号控制计算设定值或期望值的误差，特殊情况下，必须从 ADC 转换的数字输出中计算。PPB 可以自动执行这些功能，减少采样到输出的延迟并减少软件开销。误差计算需要先将 ADCPPBxCONFIG 指向所需的 SOC，然后将一个值写入 ADCPPBxOFFCAL 寄存器才能完成。向 OFFREF 写入 0 可以禁用误差计算功能，将 ADCRESULT 传递给 ADCPPBxRESULT 寄存器。后处理模块将自动从 ADCRESULT 值中减去 OFFREF 中的值并存储在 ADCPPBxRESULT 寄存器中。此时将产生一个符号扩展的 32 位结果。设置 ADCPPBxCONFIG [TWOSCOMPEN]，可以在将计算值存储到 ADCPPBxRESULT 寄存器之前选择性地反转计算值。

注意：

（1） 写入 OFFREF 的值必须小于等于 12 位。

（2） 由于 ADCPPBxRESULT 寄存器对于每个 PPB 都是唯一的，所以可以将多个 PPB 指向相同的 SOC 并为每个 PPB 获得不同的结果。

PPB 极限检测和过零检测

特殊情况下，会对 ADC 转换结果执行限制检测。PPB 可以根据上限或下限在 ADCPPBxRESULT 更改符号时自动执行检测。过零检测电路认为零的结果为正。基于这些比较，可以自动生成到 PWM 的跳闸和/或中断，降低采样到 PWM 的延迟并减少软件开销。要执行检测需要先将 ADCPPBxCONFIG 指向所需的 SOC，然后将值写入 ADCPPBxTRIPHI [LIMITHI]和 ADCPPBxTRIPLO [LIMITLO]才能启用。一旦超过限制，LIMITHI 被删除，LIMITLO 将在 ADCEVTSTAT 寄存器中设置。

该功能还具有安全意识的应用程序，可以在没有任何 CPU 干预的情况下基于超出范围的 ADC 转换触发 PWM。ADCEVTSTAT [PPBxZERO]位由 EOC 门控，ADCEVTCLR 寄存器中有相应的位来清除这些标志。ADCEVTSEL 寄存器有相应的位允许其通过 PWM 传播。ADCEVTINTSEL 寄存器有相应的位允许其通过 NVIC 传播。一个 NVIC 在给定 ADC 模块的所有 PPB 之间共享。将多个 PPB 指向相同的 SOC，可以为高/低比较，和/或过零分别生成不同的 PWM 跳闸。

PPB 采样延迟捕获

当同一 ADC 上有多个控制循环在异步运行时，来自两个或更多循环的 ADC 请求可能会发生冲突，导致其中一个采样延迟，这称为测量误差。通过了解延迟发生的时间和已经发生的延迟量，软件采用外推技术来减少错误，所以每个 PPB 都有一个包含关联 SOC 被触发和开始转换之间的 APBCLK 周期数的 DLYSTAMP 位域，通过一个位于 FREECOUNT 位域的 12 位计数器实现，其基于 APBCLK。在使用很慢的转换时容易发生以下情况：如果在 SOC 触发器和 SOC 获取的实际开始之间超过 4096 个 APBCLK 周期，FREECOUNT 可能会溢出多次，导致 DLYSTAMP 值不正确。

当相关 SOC 的触发器到达时，该计数器的值被加载到 REQSTAMP 中。当 SOC 的实际示例窗口开始时，REQSTAMP 中的值将从当前 DLYSTAMP 值中减去，并存储在 DLYSTAMP 中。

如果相关的 SOC 通过软件触发，则采样延迟捕获将不起作用。但是，如果不同 SOC 的软件触发导致与 PPB 相关的 SOC 延迟，则它将正确记录延迟。

27.5.14 开路/短路检测电路

可用于检测系统中的引脚故障。电路连接到 ADC 输入通道选择复用器之后，S+H 电路之前。向 DETECTCFG 位域进行写操作可以控制该电路，电路在任何转换的 S+H 相位期间向输入端提供电压。

表格 105 DETECTCFG 配置

DETECTCFG	电压源	S1	S2	S3	S4	驱动器阻抗
000	关闭	开	开	开	开	开

DETECTCFG	电压源	S1	S2	S3	S4	驱动器阻抗
001	零刻度	关	开	开	关	7.77K||9.07K
010	满刻度	开	关	关	开	7.77K||9.07K
011	7.77/16.84 VDDA	关	开	关	开	7.77K||9.07K
100	9.07/16.84 VDDA	开	关	开	关	7.77K||9.07K
101	零刻度	开	开	开	关	7.77K
110	满刻度	开	开	关	开	7.77K
111	零刻度	关	开	开	开	9.07K

实施

并联电容 C_P，串联电阻 R_S 的信号源，等效 DETECTCFG 电阻 $R_{DETECTCFG}$ 和电压 $V_{DETECTCFG}$ 组成 DETECTCFG 实现的代表性电路可以作为计算进入采样电容的信号电平的基础。DETECTCFG 电路提供等效输入电阻 $R_{DETECTCFG}$ 和电压源 $V_{DETECTCFG}$，具体配置参见上表。在开启 DETECTCFG 特性的情况下，如果信号源 V_S 处于驱动状态，将输入信号导出为 S/H。

输入阻抗 R_S 和 C_P 可以是信号源的组成部分，或者是在设计中使信号前置进行或控制信号沉降时间以满足 S/H 要求。因为输入路径会影响转换结果，所以在使用 DETECTCFG 特性时必须要考虑。大于数百 pF 的 C_P 值需要使用更高的 ACQPS 来确保输入端的信号在转换之前已经稳定。

使能电路的配置步骤如下：

（1） 配置用于转换的 ADC。

（2） 为分压器连接配置 ADCOSDETECT 寄存器。

（3） 启动转换并检查转换结果。

需要根据输入端驱动的因素以及 R_S 和 C_P 的值来解释结果。如果 V_S 信号可以与输入引脚断开，则电路可以用于检测断开和短路的输入引脚。

检测开路输入引脚

循环各种 DETECTCFG 配置，输入信号被拉向源电压。在引脚未打开时，具有良好驱动强度的输入将受到最小影响。如果引脚打开，则采样电压将接近上表所示的电压源。

检测短路输入引脚

循环各种 DETECTCFG 配置，输入信号将被拉向源电压。在引脚不短路时，驱动强度有限的输入将被拉向每个源电压。如果引脚短路，信号将保持在相同的电压。

27.5.15 升高顺序

在设备上电或系统复位时，ADC 将断电并禁用。给 ADC 上电的步骤如下：

（1） 在 PCLKCR13 寄存器中设置 ADC 时钟。

（2） 在 ADCCTL2 [PRESCALE]中设置 ADC 时钟分频器。

（3） 设置 ADCCTL1 [ADCPWDNZ]位上电 ADC。

（4） 允许采样前的延迟。

如果多个同时上电，可以在一条写指令中对所有 ADC 分别完成步骤 1 和步骤 3。只要在所有 ADC 开始上电之后发生，就只需要一个延迟。

27.5.16 校准

这个功能可以在制造和测试过程中，校准 ADC 的增益，偏移和线性以及缓冲 DAC 的偏移。这些微调设置被嵌入到保留的 OTP 内存中。必须要加载适当的工厂配置才能保证 ADC 在规范运行，微调值也同样需要为出厂设置才能保证。引导 ROM 将调用校准函数，所以要在没有干预的情况下初始填充微调值。如果由于模块重置或其他原因修改而清除了微调，则可以调用校准函数。

（1） Device_cal()函数将 ADC 和 DAC 偏移量的微调值从 OTP 内存复制到各自的寄存器中。

（2） Device_cal()中的 trim 函数可以调用 ADC_setOFFSETTRIM()，ADC_setINLTRIM()，DAC_setDACTRIM()。这些函数从模拟模块寄存器目的地及 OTP 内存源位置获取微调值。

（3） 每个分辨率和信号模式的组合需要不同的偏移微调。GetAdcOffsetTrimOTP(Uint16)函数接受与 ADC，分辨率和信号模式相对应的输入值，并从 OTP 内存返回相应的偏移微调值，然后将其移动到 ADCOFFTRIM 中。

ADC 零偏移校准

在 VREFLO 转换电压时发生的与 0 的差值叫零偏移误差，它分为正负。为了纠正这个错误，需要在 ADC 内核中将相等幅度和相反极性的微调写入 ADCOFFTRIM 寄存器。这些值将在 ADCRESULTx 可用之前应用。结果的时序不会受到影响，并且对于所有微调值都将保持 ADC 的完整动态范围。使用 GetAdcOffsetTrimOTP(Uint16)函数，ADCOFFTRIM 寄存器可以装载出厂校准的偏移量纠错。必要时还可以修改 ADCOFFTRIM 寄存器来补偿由应用程序环境引起的额外偏移错误。

在 12 位单端模式下重新校准 ADC 偏移量的步骤如下：

（1） 设置 ADCOFFTRIM 为+112 步(0x70)，增加一个人为偏移量来解释可能驻留在 ADC 内核中的负偏移量。

（2） 在 VREFLO 上执行 16 次转换的倍数，累积结果。

（3） 将累积结果除以 16 的倍数。

（4） 将 ADCOFFTRIM 设置为 112 -步骤 3 的结果。

注意：与转换器分辨率无关，每个 ADCOFFTRIM 的大小都是(VREFHI-VREFLO)/ 65536。

27.5.17 时序

将模拟电压转换为数字值的过程分为 S+H 相位和转换相位。ADCCTL2 [PRESCALE]对 APBCLK 分频可以产生 ADCCLK。APBCLK 时钟控制 ADC 采样和保持电路(S+H)，ADCCLK 时钟控制 ADC 转换过程。转换时间是 APBCLK 周期的整数，约为 10.5 个 ADCCLK 周期。必须确保此持续时间超过最小 S+H 持续时间和 1 ADCCLK 周期。

时序参数

（1） t_{INT}：从 S+H 窗口结束到设置 ADCINT 标志的时间（如果配置了）。

- 如果设置了 ADCCTL1 [INTPULSEPOS]，t_{INT} 与锁存到 ADCRESULTx 中的转换结果一致。
- 如果 INTPULSEPOS 位为 0，且 ADCINTCYCLE [DELAY]不为 0，那么在 ADCINTx 标志设置之前将有 DELAY 个 APBCLK 周期的延迟。这个延迟可以用来在采样就绪时进入 ISR 或触发 DMA。
- 如果 INTPULSEPOS 位为 0，且 ADCINTCYCLE [DELAY]为 0，t_{INT} 将与 S+H 窗口的末端重合。
- 如果 t_{INT} 触发对 ADCRESULTx 的读取，需要确保读取发生在结果锁存之后，否则将读取先前的结果。

（2） t_{LAT}：从 S+H 窗口结束到将 ADC 结果锁存到 ADCRESULTx 寄存器的时间。如果在此之前读取 ADCRESULTx 寄存器，则将返回先前的转换结果。

（3） t_{EOC}：从 S+H 窗口结束到下一个 ADC 转换的 S+H 窗口可以开始的时间。随后的采样可以在转换结果锁定之前开始。

（4） t_{SH}：S+H 窗口的持续时间。在该窗口结束时，S+H 电容上的值变为要转换为数字值的电压。持续时间由(ACQPS+ 1) APBCLK 周期决定。ACQPS 可以为每个 SOC 单独配置，因此不同 SOC 的 t_{SH} 也不同。

注意：无论如何设置器件时钟，S+H 电容上的值都会在 S+H 窗口结束前大约 5ns 时被捕获。

时序图

在以下情况下两个 SOC 的 ADC 转换时序：

- SOC0 和 SOC1 使用相同的触发器
- 轮询指针指向 SOC0 先转换
- 触发器发生时，没有其他 SOC 正在转换或挂起
- ADCINTSEL 配置为在 SOC0 转换结束时设置 ADCINTx 标志

图 49 早期中断模式下 12 位模式的 ADC 时序

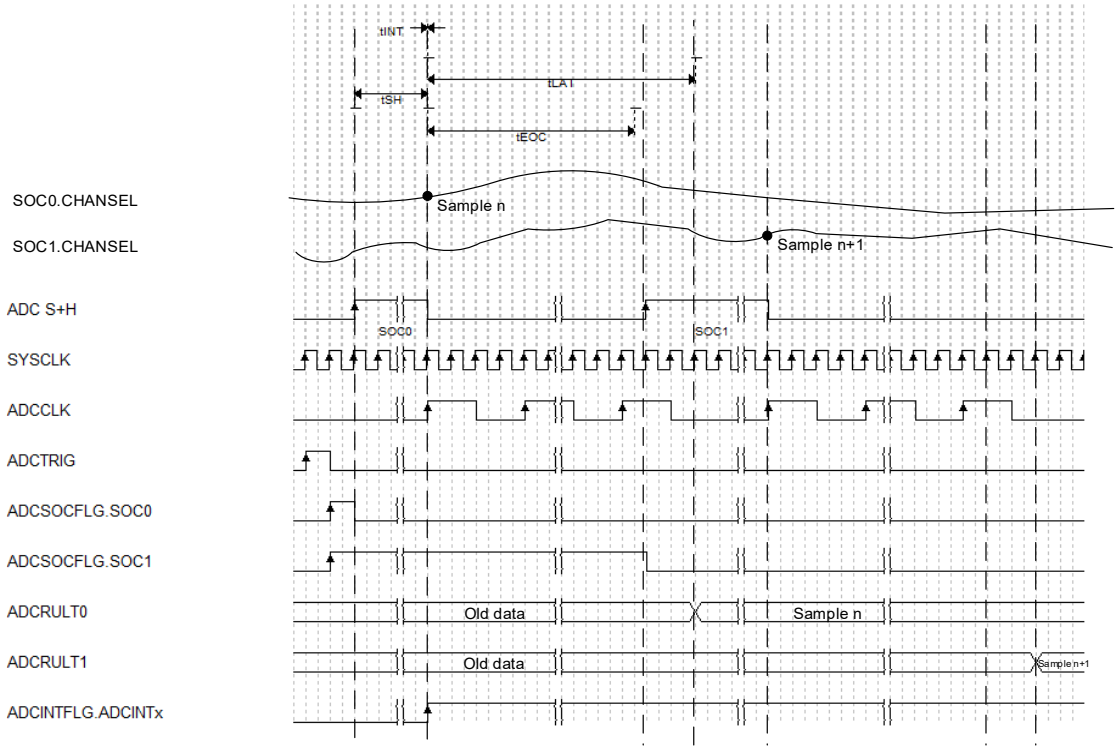
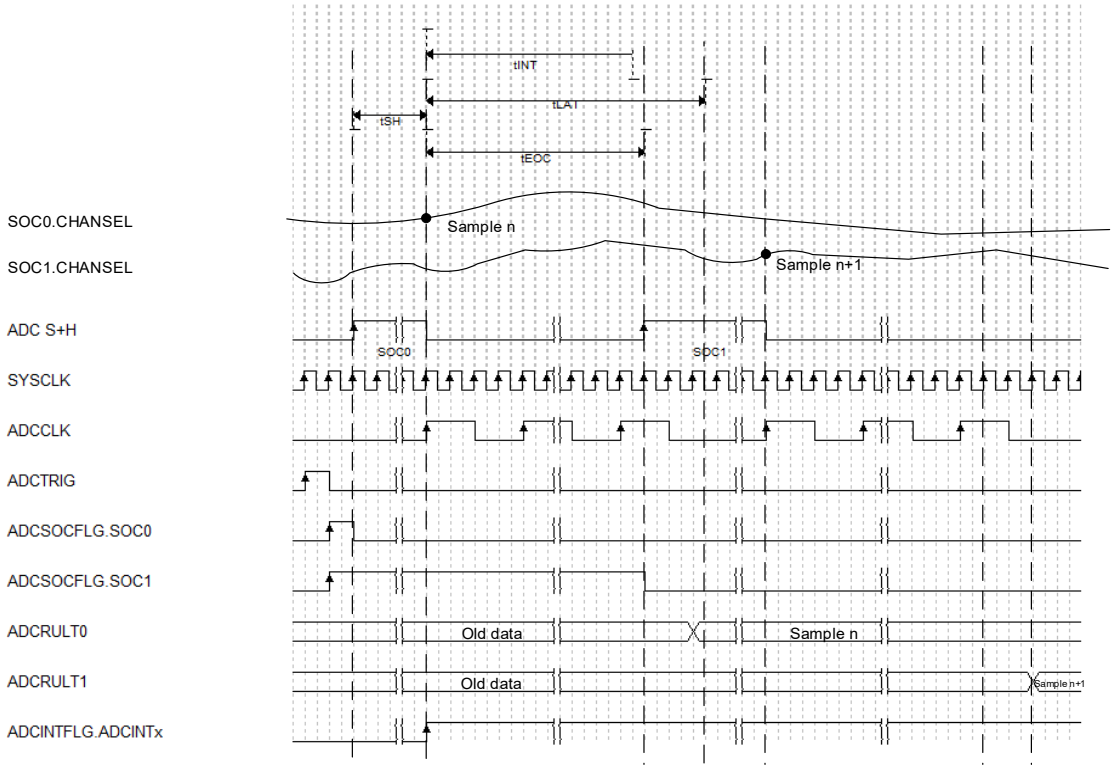


图 50 延迟中断模式下 12 位模式的 ADC 时序



表格 106 12 位模式的 ADC 时序

ADCCLK 预分频		APBCLK 周期			
预分频比	PRESCALE	t_{LAT}	t_{EOC}	t_{INT}（早期）	t_{INT}（后期）
1	0	13	11	1	11
2	2	23	21	1	21
3	4	34	31	1	31
4	6	44	41	1	41
5	8	55	51	1	51
6	10	65	61	1	61
7	12	76	71	1	71
8	14	86	81	1	81

27.5.18 同步运行

为了达到最佳效果，器件上的所有 ADC 需要同步操作。向所有 ADC 写入配置可以实现同步，这些配置使所有 ADC 的采样和转换阶段精确对齐。为每个 ADC 的触发选择和 ACQPS (S+H 持续时间)的 SOC 配置写入相同的值可以实现。

基本同步

在 ADCA 和 ADCB 上配置两个 SOC，每个 SOC 具有相同的触发选择和 ACQPS 值将导致 ADCA 和 ADCB 同步。配置两个以上 ADC 的器件原理相同。相同编号的 SOC 的 ACQPS 值必须相同，反之可以不同。同步操作不需要单个全局 S+H 时间，只有同时采样的通道需要相同的 S+H 持续时间。任何通道选择值都可以用于 SOC。如果使用高优先级的 SOC，则所有 ADC 的优先级必须设置为一致。

与多个触发源同步

要使用多个触发源和同步，需要每组 SOC 具有相同的触发选择和 ACQPS 设置。使用三个 SOC 和两个触发器源时 ADCA 和 ADCB 之间也可以同步。

除软件触发外，TRIGSEL 可以选择的任何触发源都可以使用。无法同时为所有 ADC 发出软件触发器可能会导致异步操作。如果 ADCINTSOCSEL1 和 ADCINTSOCSEL2 寄存器为所有 ADC 配置相同且不使用软件触发来启动转换链，那么 ADCINT1 或 ADCINT2 也可以用作触发器。

SOC 数不均匀的同步

如果只使用一个触发源，一个 ADC 可以比其他 ADC 使用更多的 SOC，同时仍然进行同步操作。如果触发器在所有 SOC 完成转换前再次出现，则 ADCB 立即在 SOC0 上开始转换，ADCA 在 SOC2 完成前不再开始转换 SOC0。这将导致异步操作，所以需要注意不能溢出触发器。

不重叠的转换

如果可以保证转换时间不重叠，就可以不实现同步操作。例如，如果系统中的两个 ADC 触发器来自两个始终异相 180 度的 PWM 源，那么 SOC0 可以用于具有不同触发源和不同 ACQPS 值的 ADCA 和 ADCB。

27.5.19 同步采样

虽然 ADC 没有双 S+H 电路，但可以通过在两个或多个 ADC 上设置 SOC 触发器以使用相同的触发源实现同步采样。可以基于 PWM3 事件的 x3 次同时采样。对 ADCINA3、ADCINB5、ADCINC5 进行采样。使用 20 个 APBCLK 周期的采样窗口，但可以使用不同的持续时间。当接收到 PWM3 触发器时，三个 ADC 将立即开始并行转换。结果将在每个 ADC 的 ADCRESULT0 寄存器中。

注意：此时所有 ADC 在接收到触发器时都需处于空闲状态，否则采样不会在完全相同的时间发生。

27.5.20 内部温度传感器

内部温度传感器可以测量设备的结温度，通过设置温度传感器的 ENABLE 位开启。传感器的输出可以通过内部连接与 ADC 一起采样。要将温度传感器读数转换为温度，将温度传感器读数传递给 ADC_getTemperatureC ()函数。

TempSensor 具体引脚请查阅 AS 模块的“模拟引脚与内部连接”章节。

27.5.21 选择获取窗口持续时间

ADC 的输入信号必须有足够的时间对样品充电并保持电容。一般情况下，S+H 持续时间使采样电容将充电到最终值的$\frac{1}{2}$LSB 或$\frac{1}{4}$LSB 范围内，具体取决于可容忍的沉降误差。所需沉降时间的近似值可以使用 RC 沉降模型来确定。

S+H 时间如下：

$$t = \tau * k$$

模型的时间常数如下：

$$\tau = (Rs + Ron) * Ch + Rs * (Cs + Cp)$$

时间常数的个数如下：

$$k = \frac{2^n}{\text{settling error}} - ln\frac{Cs + Cp}{Ch}$$

表格 107 参数描述

参数	描述
Rs	ADC 驱动电路源阻抗
Ron	ADC 采样开关电阻
Ch	ADC 采样电容器
Cs	ADC 输入引脚电容
Cp	ADC 通道寄生输入电容

参数	描述
settling error	容忍沉降误差
n	ADC 分辨率

27.5.22 结果寄存器映射

ADC 结果和 ADC PPB 结果对系统中的每个存储器总线控制器都是重复的。总线控制器可以直接对结果寄存器进行读访问，且不会在多个总线控制器同时读取 ADC 结果时发生冲突。

27.5.23 外部参考电路的设计

所有 ADC 模块共享一个参考电压源可以最大限度地减少 ADC 模块之间的参考电压不匹配。高/低参考引脚之间的电容器应放在 PCB 上尽可能靠近引脚，这样可以帮助吸收高频电流。需要先由具有良好带宽和低输出阻抗的精密运算放大器缓冲参考电压后才能驱动到参考引脚。需要确保运放稳定性才能让串联电阻器与电容器串联。可以在一个运算放大器驱动器之间共享两个参考引脚。但性能比每个参考引脚都有专用运算放大器缓冲时略低，其仍然可以实现所有 ADC 规格。

27.6 寄存器组地址

表格 108 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
ADCARegs	ADC_REGS	0x4002_0000	0x4002_03FF
ADCBRegs	ADC_REGS	0x4002_0400	0x4002_07FF
ADCCRegs	ADC_REGS	0x4002_0800	0x4002_0BFF
ADCARESULTRegs	ADC_RESULT	0x4002_0100	0x4002_012F
ADCBRESULTRegs	ADC_RESULT	0x4002_0500	0x4002_052F
ADCCRESULTRegs	ADC_RESULT	0x4002_0900	0x4002_092F

27.7 寄存器地址映射

表格 109 ADC_REGS 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
ADCCTL1	控制寄存器 1	0x00	√
ADCCTL2	控制寄存器 2	0x02	√
ADCBURSTCTL	突发控制寄存器	0x04	√
ADCINTFLG	中断标志寄存器	0x06	-

寄存器名称	寄存器描述	偏移地址	WRPRT
ADCINTFLGCLR	清除中断标志寄存器	0x08	-
ADCINTOVF	中断溢出寄存器	0x0A	-
ADCINTOVFCLR	清除中断溢出寄存器	0x0C	-
ADCINTSEL1N2	选择中断 1/2 寄存器	0x0E	√
ADCINTSEL3N4	选择中断 3/4 寄存器	0x10	√
ADCSOCPRICTL	SOC 优先级控制寄存器	0x12	√
ADCINTSOCSEL1	选择 SOC 中断触发寄存器 1	0x14	√
ADCINTSOCSEL2	选择 SOC 中断触发寄存器 2	0x16	√
ADCSOCFLG1	SOC 等待标志寄存器 1	0x18	-
ADCSOCFRC1	SOC 强制启动寄存器 1	0x1A	-
ADCSOCOVF1	SOC 溢出标志寄存器 1	0x1C	-
ADCSOCOVFCLR1	清除 SOC 溢出寄存器 1	0x1E	-
ADCSOC0CTL	SOC0 控制寄存器	0x20	√
ADCSOC1CTL	SOC1 控制寄存器	0x24	√
ADCSOC2CTL	SOC2 控制寄存器	0x28	√
ADCSOC3CTL	SOC3 控制寄存器	0x2C	√
ADCSOC4CTL	SOC4 控制寄存器	0x30	√
ADCSOC5CTL	SOC5 控制寄存器	0x34	√
ADCSOC6CTL	SOC6 控制寄存器	0x38	√
ADCSOC7CTL	SOC7 控制寄存器	0x3C	√
ADCSOC8CTL	SOC8 控制寄存器	0x40	√
ADCSOC9CTL	SOC9 控制寄存器	0x44	√
ADCSOC10CTL	SOC10 控制寄存器	0x48	√
ADCSOC11CTL	SOC11 控制寄存器	0x4C	√
ADCSOC12CTL	SOC12 控制寄存器	0x50	√
ADCSOC13CTL	SOC13 控制寄存器	0x54	√
ADCSOC14CTL	SOC14 控制寄存器	0x58	√
ADCSOC15CTL	SOC15 控制寄存器	0x5C	√
ADCEVTSTAT	事件状态寄存器	0x60	-
ADCEVTCLR	清除事件寄存器	0x64	-
ADCEVTSEL	使能事件寄存器	0x68	√
ADCEVTINTSEL	使能事件中断寄存器	0x6C	√

寄存器名称	寄存器描述	偏移地址	WRPRT
ADCOSDETECT	配置 ADC 开路和短路检测寄存器	0x70	√
ADCCOUNTER	自由运行计数器寄存器	0x72	-
ADCREV	版本寄存器	0x74	-
ADCOFFTRIM	偏移微调寄存器	0x76	√
ADCPPB1CONFIG	配置 PPB1 寄存器	0x80	√
ADCPPB1STAMP	PPB1 采样延迟时间戳寄存器	0x82	-
ADCPPB1OFFCAL	PPB1 偏移校准寄存器	0x84	√
ADCPPB1OFFREF	PPB1 偏移参考寄存器	0x86	-
ADCPPB1TRIPHI	PPB1 上限寄存器	0x88	√
ADCPPB1TRIPLO	PPB1 下限寄存器	0x8C	√
ADCPPB2CONFIG	配置 PPB2 寄存器	0x90	√
ADCPPB2STAMP	PPB2 采样延迟时间戳寄存器	0x92	-
ADCPPB2OFFCAL	PPB2 偏移校准寄存器	0x94	√
ADCPPB2OFFREF	PPB2 偏移参考寄存器	0x96	-
ADCPPB2TRIPHI	PPB2 上限寄存器	0x98	√
ADCPPB2TRIPLO	PPB2 下限寄存器	0x9C	√
ADCPPB3CONFIG	配置 PPB3 寄存器	0xA0	√
ADCPPB3STAMP	PPB3 采样延迟时间戳寄存器	0xA2	-
ADCPPB3OFFCAL	PPB3 偏移校准寄存器	0xA4	√
ADCPPB3OFFREF	PPB3 偏移参考寄存器	0xA6	-
ADCPPB3TRIPHI	PPB3 上限寄存器	0xA8	√
ADCPPB3TRIPLO	PPB3 下限寄存器	0xAC	√
ADCPPB4CONFIG	配置 PPB4 寄存器	0xB0	√
ADCPPB4STAMP	PPB4 采样延迟时间戳寄存器	0xB2	-
ADCPPB4OFFCAL	PPB4 偏移校准寄存器	0xB4	√
ADCPPB4OFFREF	PPB4 偏移参考寄存器	0xB6	-
ADCPPB4TRIPHI	PPB4 上限寄存器	0xB8	√
ADCPPB4TRIPLO	PPB4 下限寄存器	0xBC	√
ADCINTCYCLE	早期中断周期延迟寄存器	0xDE	√
ADCINLTRIM2	线性微调寄存器 2	0xE4	√
ADCINLTRIM3	线性微调寄存器 3	0xE8	√

表格 110 ADC_RESULT 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
ADCRESULT0	结果寄存器 0	0x00	-
ADCRESULT1	结果寄存器 1	0x02	-
ADCRESULT2	结果寄存器 2	0x04	-
ADCRESULT3	结果寄存器 3	0x06	-
ADCRESULT4	结果寄存器 4	0x08	-
ADCRESULT5	结果寄存器 5	0x0A	-
ADCRESULT6	结果寄存器 6	0x0C	-
ADCRESULT7	结果寄存器 7	0x0E	-
ADCRESULT8	结果寄存器 8	0x10	-
ADCRESULT9	结果寄存器 9	0x12	-
ADCRESULT10	结果寄存器 10	0x14	-
ADCRESULT11	结果寄存器 11	0x16	-
ADCRESULT12	结果寄存器 12	0x18	-
ADCRESULT13	结果寄存器 13	0x1A	-
ADCRESULT14	结果寄存器 14	0x1C	-
ADCRESULT15	结果寄存器 15	0x1E	-
ADCPPB1RESULT	后处理块 1 结果寄存器	0x20	-
ADCPPB2RESULT	后处理块 2 结果寄存器	0x24	-
ADCPPB3RESULT	后处理块 3 结果寄存器	0x28	-
ADCPPB4RESULT	后处理块 4 结果寄存器	0x2C	-

27.8 寄存器功能描述

27.8.1 控制寄存器 1(ADCCTL1)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	保留			0h
2	INTPULSEPOS	R/W	配置 ADC 中断脉冲位置（ADC Interrupt Pulse Position Configure） 0：中断脉冲产生发生在 ADC 开始转换时加上 ADCINTCYCLE 指定的 APBCLK 周期。 1：中断脉冲产生发生在转换结束时，在 ADC 结果存储到其结果寄存器之前 1 个周期。	0h
6:3	保留			0h

位/域	名称	R/W	描述	复位值
7	ADCPWDNZ	R/W	配置 ADC 掉电（ADC Power Down Configure） 控制模拟内核内所有模拟电路的上电和掉电。 0：掉电 1：上电	0h
11:8	ADCBSYCHN	R	ADC 忙碌通道标志（ADC Busy Channel Flag） 当产生 ADC 的 SOC 时设置。 ADCBSY =0 时：保存最后一次转换的 SOC 值。 ADCBSY =1 时：正在处理的 SOC。 0000：SOC0 目前正在处理或上一次 SOC 转换 0001：SOC1 目前正在处理或上一次 SOC 转换 … 1111：SOC15 目前正在处理或上一次 SOC 转换	0h
12	保留			0h
13	ADCBSY	R	ADC 忙碌标志（ADC Busy Flag） 确定 ADC 是否可用于采样。当产生 ADC 的 SOC 时设置，S/H 脉冲负沿后由四个 ADC 时钟硬件清除。 0：ADC 可用于下一个通道采样。 1：ADC 忙碌，无法对另一个通道采样。	0h
15:14	保留			0h

27.8.2 控制寄存器 2(ADCCTL2)

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	PRESCALE	R/W	ADC 时钟预分频器（ADC Clock Prescaler） 0000 ADCCLK =输入时钟/ 1.0 0001 保留 0010 ADCCLK =输入时钟/ 2.0 0011 保留 … 1110 ADCCLK =输入时钟/ 8.0 1111 保留	0h
15:4	保留			0h

27.8.3 突发控制寄存器(ADCBURSTCTL)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
5:0	BURSTTRIGSEL	R/W	选择 SOC 突发触发源（SOC Burst Trigger Source Select） 配置将启动突发转换序列的触发器。 000000：BURSTTRIG0 -仅软件 000001：BURSTTRIG1 - CPU1 定时器 0，TINT0 000010：BURSTTRIG2 - CPU1 定时器 1，TINT1 000011：BURSTTRIG3 - CPU1 定时器 2，TINT2 000100：BURSTTRIG4 - GPIO，Input X-Bar INPUT5 000101：BURSTTRIG5 - PWM1，ADCSOCA 000110：BURSTTRIG6 - PWM1，ADCSOCB 000111：BURSTTRIG7 - PWM2，ADCSOCA 001000：BURSTTRIG8 - PWM2，ADCSOCB … 010011：BURSTTRIG19 - PWM8，ADCSOCA 010100：BURSTTRIG20 - PWM8，ADCSOCB 010101-111111：保留	0h
7:6	保留			0h
11:8	BURSTSIZE	R/W	选择 SOC 突发大小（SOC Burst Size Select） 选择当一个突发转换序列开始时有多少 SOC 被转换。转换的第一个 SOC 由轮询指针决定，该指针随着每个 SOC 的转换而推进。 0000：1 个 SOC 转换 0001：2 个 SOC 转换 … 1111：16 个 SOC 转换	0h
14:12	保留			0h
15	BURSTEN	R/W	使能 SOC 突发模式（SOC Burst Mode Enable） 0：禁止 1：使能	0h

27.8.4 中断标志寄存器(ADCINTFLG)

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1	ADCINTx	R	ADC 中断 x 标志（ADC Interrupt x Flag）（x=1-4） 表示自上次清除后是否产生了相关的 ADCINT 脉冲。如果 ADC 中断被放置在继续中断模式，当发生一个选定的 EOC 事件时，无论标志位是否设置，都会产生进一步的中断脉冲。如果不启用连续模式，则不会产生进一步的中断脉冲，直到清除该位。	0h
15:4	保留			0h

27.8.5 清除中断标志寄存器(ADCINTFLGCLR)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	ADCINTx	R-0/W1C	清除 ADC 中断 x 标志（ADC Interrupt x Flag Clear）（x=1-4） 0：无处理 1：清除 ADCINTFLG 寄存器中相应的中断标志。如果软件将清除位和硬件试图设置标志位设置在同一周期上，则硬件具有优先级，且不会设置溢出位。读取返回 0。	0h
15:4	保留			0h

27.8.6 中断溢出寄存器(ADCINTOVF)

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	ADCINTx	R	ADC 中断 x 溢出标志（ADC Interrupt x Overflow Flag）（x=1-4） 表示产生 ADCINT 脉冲时是否发生溢出。如果设置了相应的标志位，并生成了一个选定的附加 EOC 触发器，则会发生溢出。 0：未检测到 ADC 中断溢出 1：检测到 ADC 中断溢出	0h
15:4	保留			0h

27.8.7 清除中断溢出寄存器(ADCINTOVFCLR)

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	ADCINTx	R-0/W1C	清除 ADC 中断 x 溢出标志（ADC Interrupt x Overflow Flag Clear）（x=1-4） 0：无处理 1：清除 ADCINTOVF 寄存器中相应的溢出标志。如果软件将清除位和硬件试图设置溢出位设置在同一周期上，则硬件具有优先级，且不会设置溢出位。	0h
15:4	保留			0h

27.8.8 选择中断 1/2 寄存器(ADCINTSEL1N2)

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	INT1SEL	R/W	选择 ADC 的 INT1 EOC 源（ADC INT1 EOC Source Select） 0000：EOC0 触发 ADCINT1 0001：EOC1 触发 ADCINT1 … 1111：EOC15 触发 ADCINT1	0h
4	保留			0h

位/域	名称	R/W	描述	复位值
5	INT1E	R/W	使能 ADC 的 INT1 中断（ADC INT1 Enable） 0：禁止 1：中断	0h
6	INT1CONT	R/W	使能 ADC 的 INT1 连续中断模式（ADC INT1 Continue to Interrupt Mode Enable） 0：在 ADCINT1 标志被清除前，不再产生 ADCINT1 脉冲。 1：无论标志位是否清除，只要产生 EOC 脉冲，就会产生 ADCINT1 脉冲。	0h
7	保留			0h
11:8	INT2SEL	R/W	选择 ADC 的 INT2 EOC 源（ADC INT2 EOC Source Select） 0000：EOC0 触发 ADCINT2 0001：EOC1 触发 ADCINT2 … 1111：EOC15 触发 ADCINT2	0h
12	保留			0h
13	INT2E	R/W	使能 ADC 的 INT2 中断（ADC INT2 Enable） 0：禁止 1：中断	0h
14	INT2CONT	R/W	使能 ADC 的 INT2 连续中断模式（ADC INT2 Continue to Interrupt Mode Enable） 0：在 ADCINT2 标志被清除前，不再产生 ADCINT2 脉冲。 1：无论标志位是否清除，只要产生 EOC 脉冲，就会产生 ADCINT2 脉冲。	0h
15	保留			0h

27.8.9 选择中断 3/4 寄存器(ADCINTSEL3N4)

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	INT3SEL	R/W	选择 ADC 的 INT3 EOC 源（ADC INT3 EOC Source Select） 0000：EOC0 触发 ADCINT3 0001：EOC1 触发 ADCINT3 … 1111：EOC15 触发 ADCINT3	0h
4	保留			0h
5	INT3E	R/W	使能 ADC 的 INT3 中断（ADC INT3 Enable） 0：禁止 1：中断	0h
6	INT3CONT	R/W	使能 ADC 的 INT3 连续中断模式（ADC INT3 Continue to Interrupt Mode Enable） 0：在 ADCINT3 标志被清除前，不再产生 ADCINT3 脉冲。 1：无论标志位是否清除，只要产生 EOC 脉冲，就会产生 ADCINT3 脉冲。	0h
7	保留			0h

位/域	名称	R/W	描述	复位值
11:8	INT4SEL	R/W	选择 ADC 的 INT4 EOC 源（ADC INT4 EOC Source Select） 0000：EOC0 触发 ADCINT4 0001：EOC1 触发 ADCINT4 … 1111：EOC15 触发 ADCINT4	0h
12	保留			0h
13	INT4E	R/W	使能 ADC 的 INT4 中断（ADC INT4 Enable） 0：禁止 1：中断	0h
14	INT4CONT	R/W	使能 ADC 的 INT4 连续中断模式（ADC INT4 Continue to Interrupt Mode Enable） 0：在 ADCINT4 标志被清除前，不再产生 ADCINT4 脉冲。 1：无论标志位是否清除，只要产生 EOC 脉冲，就会产生 ADCINT4 脉冲。	0h
15	保留			0h

27.8.10 SOC 优先级控制寄存器(ADCSOCPRICTL)

偏移地址：0x12

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	SOCPRIORITY	R/W	配置 SOC 优先级（SOC Priority Configure） 确定 SOCx 的优先级模式和轮询仲裁的截止点。 00000：SOC 优先级在所有通道的轮询模式中处理。 00001：SOC0 为高优先级，其余通道为轮询模式。 00010：SOC0-SOC1 为高优先级，SOC2-SOC15 为轮询模式。 00011：SOC0-SOC2 为高优先级，SOC3-SOC15 为轮询模式。 … 01111：SOC0-SOC14 为高优先级，SOC15 为轮询模式。 10000：所有的 SOC 都为高优先级模式，根据 SOC 号进行仲裁。 10001-11111：保留	0h
9:5	RRPOINTER	R	轮询指针（Round Robin Pointer） 保存最后一次转换的轮询 SOCx 的值，轮询模式将使用该值来确定转换的顺序。 00000：SOC0 是最后一个轮询 SOC 转换，SOC1 是最高的轮询优先级。 00001：SOC1 是最后一个轮询 SOC 转换，SOC2 是最高的轮询优先级。 … 01111：SOC15 是最后一个轮询 SOC 转换，SOC0 是最高的轮询优先级。 10000：复位值，表示未转换 SOC。SOC0 是最高的轮询优先级。当设备复位时设置此值，当写入	10h

位/域	名称	R/W	描述	复位值
			ADCSOCPRICTL 寄存器时正在进行转换，则转换完成，然后新的优先级生效。 10001-11111：保留	
15:10	保留			0h

27.8.11 选择 SOC 中断触发寄存器 1 (ADCINTSOCSEL1)

这个寄存器选择触发 SOCx 的 ADCINT，和 ADCSOCxCTL [TRIGSEL]共同决定 SOC 的触发器。

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2x+1:2x	SOCx	R/W	选择 SOCx 的 ADC 中断触发（SOCx ADC Interrupt Trigger Select）（x=0-7） 00：没有 ADCINT 触发 SOCx 01：ADCINT1 触发 SOCx 10：ADCINT2 触发 SOCx 11：保留	0h

27.8.12 选择 SOC 中断触发寄存器 2 (ADCINTSOCSEL2)

这个寄存器选择触发 SOCx 的 ADCINT，和 ADCSOCxCTL [TRIGSEL]共同决定 SOC 的触发器。

偏移地址：0x16

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	SOC8	R/W	选择 SOC8 的 ADC 中断触发（SOC8 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC8 01：ADCINT1 触发 SOC8 10：ADCINT2 触发 SOC8 11：保留	0h
3:2	SOC9	R/W	选择 SOC9 的 ADC 中断触发（SOC9 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC9 01：ADCINT1 触发 SOC9 10：ADCINT2 触发 SOC9 11：保留	0h
5:4	SOC10	R/W	选择 SOC10 的 ADC 中断触发（SOC10 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC10 01：ADCINT1 触发 SOC10 10：ADCINT2 触发 SOC10 11：保留	0h

位/域	名称	R/W	描述	复位值
7:6	SOC11	R/W	选择 SOC11 的 ADC 中断触发（SOC11 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC11 01：ADCINT1 触发 SOC11 10：ADCINT2 触发 SOC11 11：保留	0h
9:8	SOC12	R/W	选择 SOC12 的 ADC 中断触发（SOC12 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC12 01：ADCINT1 触发 SOC12 10：ADCINT2 触发 SOC12 11：保留	0h
11:10	SOC13	R/W	选择 SOC13 的 ADC 中断触发（SOC13 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC13 01：ADCINT1 触发 SOC13 10：ADCINT2 触发 SOC13 11：保留	0h
13:12	SOC14	R/W	选择 SOC14 的 ADC 中断触发（SOC14 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC14 01：ADCINT1 触发 SOC14 10：ADCINT2 触发 SOC14 11：保留	0h
15:14	SOC15	R/W	选择 SOC15 的 ADC 中断触发（SOC15 ADC Interrupt Trigger Select） 00：没有 ADCINT 触发 SOC15 01：ADCINT1 触发 SOC15 10：ADCINT2 触发 SOC15 11：保留	0h

27.8.13 SOC 等待标志寄存器 1(ADCSOCFLG1)

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x	SOCx	R	SOCx 等待标志（SOCx Wait Flag）（x=0-15） 0：没有待处理的采样 1：采样正在等待中 SOCx 转换开始时，该位自动清除。如果该位存在争用，并且在同一周期内同时收到设置请求和清除请求，设置请求的优先级较高，此时会忽略清除请求，无论 ADCSOCOVF1 寄存器的位是否已经设置，都不会受到影响。	0h

27.8.14 SOC 强制启动寄存器 1(ADCSOCFRC1)

偏移地址：0x1A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x	SOCx	R-0/W1S	SOCx 强制启动转换（SOCx Force Start Conversion） 0：无动作。 1：强制 ADCSOCFLG1 的 SOCx 位为 1。导致转换在给予 SOCx 优先级后开始。 如果软件将该位和硬件试图清除 ADCSOCFLG1 的 SOCx 位设置在同一周期上，则软件具有优先级，同时设置 ADCSOCFLG1 的 SOCx 位。此时，无论 ADCSOCFLG1 的 SOCx 位之前是否设置，相应的溢出位都不会受到影响。可以用来启动软件启动的转换。这个位读起来总是 0。	0h

27.8.15 SOC 溢出标志寄存器 1（ADCSOCOVF1）

偏移地址：0x1C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x	SOCx	R	SOCx 事件溢出标志（SOCx Event Overflow Flag） 表示在已经有 SOCx 事件已经挂起时硬件中又生成了 SOCx 事件。溢出条件不会阻止 SOCx 事件的处理。写入 ADCSOCFRC1 不影响该位。 0：无溢出 1：溢出	0h

27.8.16 清除 SOC 溢出寄存器 1（ADCSOCOVFCLR1）

偏移地址：0x1E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x	SOCx	R-0/W1S	清除 SOCx 溢出（SOCx Overflow Clear） 0：无动作 1：清除 SOCx 溢出 如果软件将该位和硬件试图设置 ADCSOCOVF1 的 SOCx 位设置在同一周期上，则硬件具有优先级，同时设置 ADCSOCOVF1 的 SOCx 位。这个位读起来总是 0。	0h

27.8.17 SOCx 控制寄存器（ADCSOCxCTL）（x=0-15）

偏移地址：0x20+a*0x04（a=0-15）

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
8:0	ACQPS	R/W	配置 SOCx 采样窗口持续时间（SOCx Sample Window Duration Configure） 配置的采样时间至少要为一个 ADCCLK 周期，才能使 ADC 正常工作。 000000000：采样窗口为 1 个 APBCLK 周期宽 000000001：采样窗口为 2 个 APBCLK 周期宽 000000010：采样窗口为 3 个 APBCLK 周期宽 … 111111111：采样窗口为 512 个 APBCLK 周期宽	0h

位/域	名称	R/W	描述	复位值
14:9	保留			0h
18:15	CHSEL	R/W	选择 SOCx 转换的通道（SOCx Convert Channel Select） 0000：ADCIN0 0001：ADCIN1 … 1111：ADCIN15	0h
19	保留			0h
24:20	TRIGSEL	R/W	选择 SOCx 触发源（SOCx Trigger Source Select） 此位域和 ADCINTSOCSELx 共同决定 SOC 的触发器。 00000：ADCTRIG0 -只有软件 00001：ADCTRIG1 -TMR0，TINT0n 00010：ADCTRIG2 - TMR1，TINT1n 00011：ADCTRIG3 - TMR2，TINT2n 00100：ADCTRIG4 - GPIO ADCEXTSOC 00101：ADCTRIG5 - PWM1 ADCSOCA 00110：ADCTRIG6 - PWM1 ADCSOCB 00111：ADCTRIG7 - PWM2 ADCSOCA 01000：ADCTRIG8 - PWM2 ADCSOCB … 10101：ADCTRIG19 - PWM8, ADCSOCA 10100：ADCTRIG20 - PWM8, ADCSOCB 10101-11111：保留	0h
31:25	保留			0h

27.8.18 事件状态寄存器（ADCEVTSTAT）

偏移地址：0x60

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PPB1TRIPHI	R/W	后处理块 1 TRIP 高标志（Post Processing Block 1 Trip High Flag） 0：未发生 1：发生数字比较跳闸高事件	0h
1	PPB1TRIPLO	R/W	后处理块 1 TRIP 低标志（Post Processing Block 1 Trip Low Flag） 0：未发生 1：发生数字比较跳闸低事件	0h
2	PPB1ZERO	R/W	后处理块 1 过零点标志（Post Processing Block 1 Zero Crossing Flag） 这个位由 EOC 信号进行门控。 0：未改变 1：ADCPPB1RESULT 寄存器已经改变符号	0h
3	保留			0h

位/域	名称	R/W	描述	复位值
4	PPB2TRIPHI	R/W	后处理块 2 TRIP 高标志（Post Processing Block 2 Trip High Flag） 0：未发生 1：发生数字比较跳闸高事件	0h
5	PPB2TRIPLO	R/W	后处理块 2 TRIP 低标志（Post Processing Block 2 Trip Low Flag） 0：未发生 1：发生数字比较跳闸低事件	0h
6	PPB2ZERO	R/W	后处理块 2 过零点标志（Post Processing Block 2 Zero Crossing Flag） 这个位由 EOC 信号进行门控。 0：未改变 1：ADCPPB2RESULT 寄存器已经改变符号	0h
7	保留			0h
8	PPB3TRIPHI	R/W	后处理块 3 TRIP 高标志（Post Processing Block 3 Trip High Flag） 0：未发生 1：发生数字比较跳闸高事件	0h
9	PPB3TRIPLO	R/W	后处理块 3 TRIP 低标志（Post Processing Block 3 Trip Low Flag） 0：未发生 1：发生数字比较跳闸低事件	0h
10	PPB3ZERO	R/W	后处理块 3 过零点标志（Post Processing Block 3 Zero Crossing Flag） 这个位由 EOC 信号进行门控。 0：未改变 1：ADCPPB3RESULT 寄存器已经改变符号	0h
11	保留			0h
12	PPB4TRIPHI	R	后处理块 4 TRIP 高标志（Post Processing Block 4 Trip High Flag） 0：未发生 1：发生数字比较跳闸高事件	0h
13	PPB4TRIPLO	R	后处理块 4 TRIP 低标志（Post Processing Block 4 Trip Low Flag） 0：未发生 1：发生数字比较跳闸低事件	0h
14	PPB4ZERO	R	后处理块 4 过零点标志（Post Processing Block 4 Zero Crossing Flag） 这个位由 EOC 信号进行门控。 0：未改变 1：ADCPPB4RESULT 寄存器已经改变符号	0h
15	保留			0h

27.8.19 清除事件寄存器（ADCEVTCLR）

偏移地址：0x64

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PPB1TRIPHI	R-0/W1S	清除后处理块 1 TRIP 高（Post Processing Block 1 Trip High Clear） 0：不清除 1：清除	0h
1	PPB1TRIPLO	R-0/W1S	清除后处理块 1 TRIP 低（Post Processing Block 1 Trip Low Clear） 0：不清除 1：清除	0h
2	PPB1ZERO	R-0/W1S	清除后处理块 1 过零点（Post Processing Block 1 Zero Crossing Clear） 0：不清除 1：清除	0h
3	保留			0h
4	PPB2TRIPHI	R-0/W1S	清除后处理块 2 TRIP 高（Post Processing Block 2 Trip High Clear） 0：不清除 1：清除	0h
5	PPB2TRIPLO	R-0/W1S	清除后处理块 2 TRIP 低（Post Processing Block 2 Trip Low Clear） 0：不清除 1：清除	0h
6	PPB2ZERO	R-0/W1S	清除后处理块 2 过零点（Post Processing Block 2 Zero Crossing Clear） 0：不清除 1：清除	0h
7	保留			0h
8	PPB3TRIPHI	R-0/W1S	清除后处理块 3 TRIP 高（Post Processing Block 3 Trip High Clear） 0：不清除 1：清除	0h
9	PPB3TRIPLO	R-0/W1S	清除后处理块 3 TRIP 低（Post Processing Block 3 Trip Low Clear） 0：不清除 1：清除	0h
10	PPB3ZERO	R-0/W1S	清除后处理块 3 过零点（Post Processing Block 3 Zero Crossing Clear） 0：不清除 1：清除	0h
11	保留			0h
12	PPB4TRIPHI	R-0/W1S	清除后处理块 4 TRIP 高（Post Processing Block 4 Trip High Clear） 0：不清除 1：清除	0h

位/域	名称	R/W	描述	复位值
13	PPB4TRIPLO	R-0/W1S	清除后处理块 4 TRIP 低（Post Processing Block 4 Trip Low Clear） 0：不清除 1：清除	0h
14	PPB4ZERO	R-0/W1S	清除后处理块 4 过零点（Post Processing Block 4 Zero Crossing Clear） 0：不清除 1：清除	0h
15	保留			0h

27.8.20 使能事件寄存器（ADCEVTSEL）

偏移地址：0x68

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PPB1TRIPHI	R/W	使能后处理块 1TRIP 高事件（Post Processing Block 1 Trip High Event Enable） 允许相应的上升跳闸高标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
1	PPB1TRIPLO	R/W	使能后处理块 1TRIP 低事件（Post Processing Block 1 Trip Low Event Enable） 允许相应的上升跳闸低标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
2	PPB1ZERO	R/W	使能后处理块 1 过零点事件（Post Processing Block 1 Zero Crossing Event Enable） 允许相应的上升过零标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
3	保留			0h
4	PPB2TRIPHI	R/W	使能后处理块 2TRIP 高事件（Post Processing Block 2 Trip High Event Enable） 允许相应的上升跳闸高标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
5	PPB2TRIPLO	R/W	使能后处理块 2TRIP 低事件（Post Processing Block 2 Trip Low Event Enable） 允许相应的上升跳闸低标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
6	PPB2ZERO	R/W	使能后处理块 2 过零点事件（Post Processing Block 2 Zero Crossing Event Enable） 允许相应的上升过零标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
7	保留			0h
8	PPB3TRIPHI	R/W	使能后处理块 3TRIP 高事件（Post Processing Block 3 Trip High Event Enable） 允许相应的上升跳闸高标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
9	PPB3TRIPLO	R/W	使能后处理块 3TRIP 低事件（Post Processing Block 3 Trip Low Event Enable） 允许相应的上升跳闸低标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
10	PPB3ZERO	R/W	使能后处理块 3 过零点事件（Post Processing Block 3 Zero Crossing Event Enable） 允许相应的上升过零标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
11	保留			0h
12	PPB4TRIPHI	R/W	使能后处理块 4TRIP 高事件（Post Processing Block 4 Trip High Event Enable） 允许相应的上升跳闸高标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
13	PPB4TRIPLO	R/W	使能后处理块 4TRIP 低事件（Post Processing Block 4 Trip Low Event Enable） 允许相应的上升跳闸低标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
14	PPB4ZERO	R/W	使能后处理块 4 过零点事件（Post Processing Block 4 Zero Crossing Event Enable） 允许相应的上升过零标志激活到 PWM 块的事件信号。必须在产生额外的事件到 PWM 块之前被清除。 0：禁止 1：使能	0h
15	保留			0h

27.8.21 使能事件中断寄存器（ADCEVTINTSEL）

偏移地址：0x6C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PPB1TRIPHI	R/W	使能后处理块 1TRIP 高中断（Post Processing Block 1 Trip High Interrupt Enable） 允许相应的上升跳闸高标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
1	PPB1TRIPLO	R/W	使能后处理块 1TRIP 低中断（Post Processing Block 1 Trip Low Interrupt Enable） 允许相应的上升跳闸低标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
2	PPB1ZERO	R/W	使能后处理块 1 过零点中断（Post Processing Block 1 Zero Crossing Interrupt Enable） 允许相应的上升过零标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
3	保留			0h
4	PPB2TRIPHI	R/W	使能后处理块 2TRIP 高中断（Post Processing Block 2 Trip High Interrupt Enable） 允许相应的上升跳闸高标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
5	PPB2TRIPLO	R/W	使能后处理块 2TRIP 低中断（Post Processing Block 2 Trip Low Interrupt Enable） 允许相应的上升跳闸低标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
6	PPB2ZERO	R/W	使能后处理块 2 过零点中断（Post Processing Block 2 Zero Crossing Interrupt Enable） 允许相应的上升过零标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
7	保留			0h
8	PPB3TRIPHI	R/W	使能后处理块 3TRIP 高中断（Post Processing Block 3 Trip High Interrupt Enable） 允许相应的上升跳闸高标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
9	PPB3TRIPLO	R/W	使能后处理块 3TRIP 低中断（Post Processing Block 3 Trip Low Interrupt Enable） 允许相应的上升跳闸低标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
10	PPB3ZERO	R/W	使能后处理块 3 过零点中断（Post Processing Block 3 Zero Crossing Interrupt Enable） 允许相应的上升过零标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
11	保留			0h
12	PPB4TRIPHI	R/W	使能后处理块 4TRIP 高中断（Post Processing Block 4 Trip High Interrupt Enable） 允许相应的上升跳闸高标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
13	PPB4TRIPLO	R/W	使能后处理块 4TRIP 低中断（Post Processing Block 4 Trip Low Interrupt Enable） 允许相应的上升跳闸低标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
14	PPB4ZERO	R/W	使能后处理块 4 过零点中断（Post Processing Block 4 Zero Crossing Interrupt Enable） 允许相应的上升过零标志激活事件中断信号到 NVIC。必须在对 NVIC 产生额外的中断之前被清除。 0：禁止 1：使能	0h
15	保留			0h

27.8.22 配置 ADC 开路和短路检测寄存器（ADCOSDETECT）

偏移地址：0x70

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	DETECTCFG	R/W	配置 ADC 开路和短路检测（ADC Opens and Shorts Detect Configure） 000：关闭开路/短路检测电路 001：在零量程使能开路/短路检测电路 010：在满量程使能开路/短路检测电路 011：开路/短路检测电路以（标称）7.77/16.84 刻度使能 100：开路/短路检测电路以（标称）9.07/16.84 刻度使能 101：使能开路/短路检测电路，（标称）7.77K 下拉到 VSSA 110：使能开路/短路检测电路，（标称）7.77K 上拉到 VDDA 111：使能开路/短路检测电路，（标称）9.07K 下拉到 VSSA	0h
15:3	保留			0h

27.8.23 自由运行计数器寄存器（ADCCOUNTER）

偏移地址：0x72

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	FREECOUNT	R	ADC 自由运行计数器数值（ADC Free Running Counter Value） 表示 ADC 自由运行的计数器数值。	0h
15:12	保留			0h

27.8.24 版本寄存器（ADCREV）

偏移地址：0x74

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TYPE	R	ADC 类型（ADC Type） 时钟设置为 5。	5h
15:8	REV	R	ADC 版本（ADC Revision） 记录不同版本之间的差异，第一个版本被标记为 00。	0h

27.8.25 偏移微调寄存器（ADCOFFTRIM）

偏移地址：0x76

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	OFFTRIM	R/W	ADC 偏移微调（ADC Offset Trim） 向上或向下调整转换器的转换结果，以考虑 ADC 中的偏移误差。在设备启动期间加载出厂配置。偏移量可以在+7 到-8 LSB 的范围内进行校正，值是 16* 8 位补码的偏移量，具体如下： 7 LSB (16* 7) = 112 6 LSB (16* 6) = 96 5 LSB (16* 5) = 80 4 LSB (16* 4) = 64 3 LSB (16* 3) = 48 2 LSB (16* 2) = 32 1 LSB (16* 1) = 16 0 LSB (16* 0) = 0 -1 LSB (16*(-1)) = 240 … -7LSB(16*(-7)) = 144	0h
15:8	保留			0h

27.8.26 配置 PPBx 寄存器（ADCPPBxCONFIG）（x=1-4）

偏移地址：0x80+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	CONFIG	R/W	配置 ADC 后处理块 x（ADC Post Processing Block x Configure） 配置与这个后处理块相关联的 SOC/EOC/RESULT。 0000：SOC0 / EOC0 / RESULT0 0001：SOC1 / EOC1 / RESULT1 … 1111：SOC15/EOC15/RESULT15	0h
4	TWOSCOMPEN	R/W	使能 ADC 后处理块 x 的二进制补码（ADC Post Processing Block x Two's Complement Enable） 在处理块将结果存储到 ADCPPBxRESULT 寄存器之前，对偏移/参考减法单元的输出执行 2 的补码。 0：ADCPPBxRESULT = ADCRESULTx – ADC_OFFREF 1：ADCPPBxRESULT = ADC_OFFREF – ADCRESULTx	0h
5	CBCEN	R/W	周期使能 ADC 后处理块（ADC Post Processing Block Cycle By Cycle Enable） 如果事件条件不存在，则转换后硬件处理电路可以自动清除转换时的 ADCEVTSTAT。	0h
15:6	保留			0h

27.8.27 PPBx 采样延迟时间戳寄存器（ADCPPBxSTAMP）（x=1-4）

偏移地址：0x82+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DLYSTAMP	R	ADC 后处理块 x 采样延迟时间戳（ADC Post Processing Block x Sample Delay Timestamp） 当 SOC 开始采样时，REQSTAMP 中包含的值从 FREECOUNT 的值中减去并加载到这个位域中，从而给出 SOC 触发和实际开始采样之间的时钟周期延迟的数量。	0h
15:12	保留			0h

27.8.28 PPBx 偏移校准寄存器（ADCPPBxOFFCAL）（x=1-4）

偏移地址：0x84+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	OFFCAL	R/W	ADC 后处理块 x 偏移校准（ADC Post Processing Block x Offset Calibrate） 可用于数字消除 ADCIN 电路中固有的系统电平偏移。需要在这个值被存储到 ADCRESULT 寄存器之前从 ADC 输出中减去。 0000000000：无变化。ADC 输出直接存储到 ADCRESULT 中。 0000000001：ADC 输出- 1 存储在 ADCRESULT 中。 0000000010：ADC 输出- 2 存储在 ADCRESULT 中。 ... 1000000000：ADC 输出+ 512 存储在 ADCRESULT 中。 ... 1111111111：ADC 输出+ 1 存储在 ADCRESULT 中。 在 12 位模式下，减法将在 000000000000 和 111111111111 饱和并且存储到 ADCRESULT 寄存器中。 在 16 位模式下，减法将在 0000000000000000 和 1111111111111111 时饱和并且存储到 ADCRESULT 寄存器中。	0h
15:10	保留			0h

27.8.29 PPBx 偏移参考寄存器（ADCPPBxOFFREF）（x=1-4）

偏移地址：0x86+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	OFFREF	R/W	ADC 后处理块 x 偏移参考值（ADC Post Processing Block x Offset Reference） 可用于计算反馈误差或通过减去参考值将单极信号转换为双极信号。这个值从 ADCRESULT 寄存器中减去，然后通过可选的二进制补码函数传递并存储在 ADCPPBxRESULT 寄存器中。这个减法是不饱和的。在 12 位模式下，这个寄存器的大小不会从 16 位改变。需要确保在 12 位模式下只有 12 位值被写入该寄存器。 0000000000000000：无变化。传递 ADCRESULT 值 0000000000000001：传递 ADCRESULT - 1 0000000000000010：传递 ADCRESULT – 2 ... 1111111111111111：传递 ADCRESULT - 65,535	0h

27.8.30 PPBx 上限寄存器（ADCPPBxTRIPHI）（x=1-4）

偏移地址：0x88+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	LIMITHI	R/W	ADC 后处理块 x 上限（ADC Post Processing Block x Trip upper Threshold） 数字比较器跳闸上限。 在 16 位模式下，所有 17 位将与 PPBRESULTx 位域的 17 位进行比较。 在 12 位模式下，此位的[12:0]将与 PPBRESULTx[12:0]进行比较。	0h
16	HSIGN	R/W	上限符号（upper Threshold Sign） 16 位 ADC 模式下 LIMITHI 的第 17 位 (符号位)。	0h
31:17	保留			0h

27.8.31 PPBx 下限寄存器（ADCPPBxTRIPLO）（x=1-4）

偏移地址：0x8C+（a-1）*0x10(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	LIMITLO	R/W	ADC 后处理块 x 下限（ADC Post Processing Block x Trip Lower Threshold） 数字比较器跳闸下限。 在 16 位模式下，所有 17 位将与 PPBRESULTx 位域的 17 位进行比较。 在 12 位模式下，此位的[12:0]将与 PPBRESULTx [12:0]进行比较。	0h
16	LSIGN	R/W	下限符号（Lower Threshold Sign） 16 位 ADC 模式下 LIMITLO 的第 17 位 (符号位)。	0h
19:17	保留			0h
31:20	REQSTAMP	R/W	ADC 后处理块 x 请求时间戳（ADC Post Processing Block x Request Timestamp） 当触发器在 ADCSOCFLGx 中设置相关的 SOC 标志时，FREECOUNT 被加载到这个位域。	0h

27.8.32 早期中断周期延迟寄存器（ADCINTCYCLE）

偏移地址：0xDE

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DELAY	R/W	ADC 早期中断周期延迟（ADC Early Interrupt Cycle Delay） 根据时钟周期定义从 ADCSOC 下降沿的延迟，以产生中断。	0h

27.8.33 线性微调寄存器 2（ADCINLTRIM2）

偏移地址：0xE4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	INLTRIM63TO32	R/W	ADC 线性微调 63-32 位（ADC Linearity Trim 63-32 Bits） 修改此寄存器的内容可能会导致 ADC 在规范外运行，所以非特殊情况不修改此寄存器。	0h

27.8.34 线性微调寄存器 3（ADCINLTRIM3）

偏移地址：0xE8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	INLTRIM95TO64	R/W	ADC 线性微调 95-64 位（ADC Linearity Trim 95_64 Bits） 修改此寄存器的内容可能会导致 ADC 在规范外运行，所以非特殊情况不修改此寄存器。	0h

27.8.35 结果寄存器 x(ADCRESULTx) (x=0-15)

偏移地址：0x00 + a*0x02(a=0-15)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RESULT	R	ADC 转换结果 x（ADC Conversion Result x）（x=0-15） 16 位 ADC 结果。在 ADC 完成 SOCx 的转换后，数字结果存储在这个位域中。	0h

27.8.36 后处理块 x 结果寄存器(ADCPPBxRESULT) (x=1-4)

偏移地址：0x1C+a*0x04(a=1-4)

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	PPBRESULT	R	ADC 后处理块转换结果 x（ADC Post Processing Block Conversion Result x）（x=1-4） 偏移/参考减法后转换的结果存储在这个位域中。如果 ADCINTFLG 在读取此位域时被轮询，需要添加一个 NOP 指令，才能确保在此寄存器中填充转换后处理。 注意：如果与此后处理块相关的转换是 12 位转换，则此位域为[12:0]。	0h
31:16	SIGN	R	符号扩展（Sign Extended） 这些位与 16 位反映相同的值。 注意：如果与此后处理块相关的转换是 12 位转换，则此位域向下延伸到 12 位，且都反映与 12 位相同的值。	0h

28 缓冲数模转换器（DAC）

28.1 术语全称、缩写描述

表格 111 术语全称、缩写描述

中文全称	英文全称	英文缩写
偏移微调	Offset Trim	OFFSET_TRIM

28.2 简介

缓冲 DAC 是一种通用 DAC，由一个内部 12 位 DAC 和一个能够驱动外部负载的模拟输出缓冲区组成，可以产生正弦波、方波、三角波等交流波形和直流电压。如果要驱动比典型更高的负载，可以在负载大小和输出电压摆幅之间进行权衡。软件对 DACVALS 的写入立即生效，也可以与 PWMSYNCPER 事件同步。

28.3 主要特征

（1） 12 位可编程内部 DAC

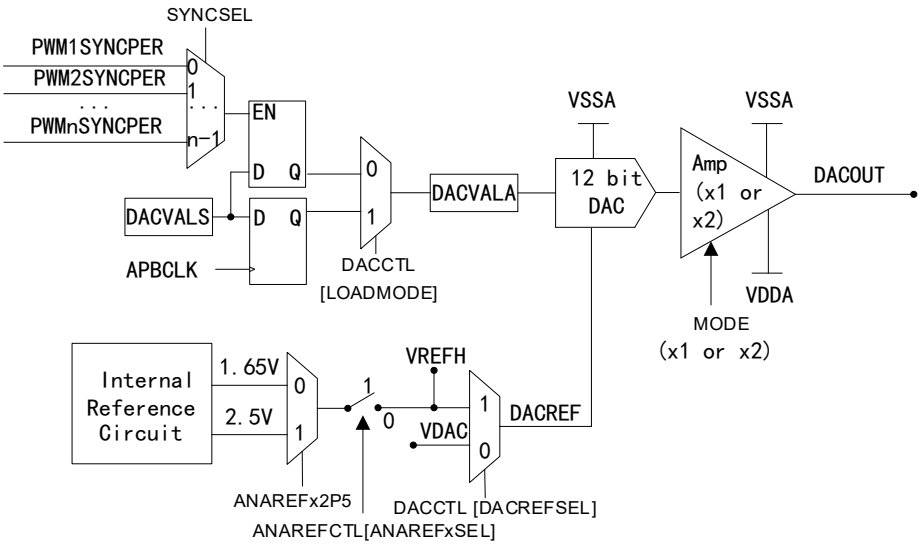
（2） 可选参考电压源

（3） 能够与 PWMSYNCPER 同步

（4） x1 和 x2 增益模式时，使用内部 VREFHI

28.4 结构框图

图 51 结构框图



28.5 功能描述

在 VDAC 和 VREFHI 之间选择内部 DAC 的参考电压源 DACREF。当 VREFHI 设置为 DACREF 并使用内部参考模式时可用 x2 增益模式。即使缓冲 DAC 具有 x2 增益模式，最大输出电压也比 VDDA 小。缓冲 DAC 支持的增益模式组合如下：

表格 112 DAC 支持的增益模式组合

DACREFSEL	ANAREFxSEL (A/B)	ANAREFx2P5 (A/B)	参考源	电压(V)	模式	最大 DAC 输出(V)	支持状态
0	0/1	0/1	外部	VDAC	0	2.5	支持
0	0/1	0/1	外部	VDAC	1	2.5	不支持
1	1	0/1	外部	VREFHI	0	2.5	支持
1	1	0/1	外部	VREFHI	1	2.5	不支持
1	0	0	内部	1.65	0	1.65	不支持
1	0	0	内部	1.65	1	3.3	支持
1	0	1	内部	2.5	0	2.5	支持
1	0	1	内部	2.5	1	2.5	不支持

注意：VDAC/VREFHI = 2.5V，VDDA = 3.3V，VAL = 4095。

缓冲 DAC 包含 DACVALA 和 DACVALS 寄存器。DACVALS 是一个可写的影子寄存器，可以立即加载到 DACVALA 中或与下一个 PWMSYNCPER 事件同步。DACVALA 是一个只读寄存器，主动控制缓冲的 DAC 值。如果缓冲 DAC 的时钟在其输出电压时被禁用，则输出电压不受影响，但 DACVALA 和 DACVALS 不再通过寄存器写入更新。重新使能缓冲 DAC 的时钟可以使其恢复到禁用时钟之前的状态。缓冲 DAC 的输出缓冲器可能在 VDDA/VSSA 附近表现出非线性行为。

内部 DAC 的理想输出如下面公式：

$$\mathrm{DACOUT} = \frac{DACVALA \times DACREF}{4096}$$

28.5.1 初始化

初始化步骤如下：

（1） 使能 DAC 缓冲时钟

（2） 选择 DACREF

（3） 使能缓冲 DAC

（4） 等待上电时间后再输出电压，根据 DAC 电气特性表确认上电时间

（5） 根据缓冲 DAC 的稳定时间间隔对 DACVALS 连续写入可以支持缓冲 DAC 的可预测行为，具体参数参见 DAC 电气特性表。

28.5.2 DAC 偏移调整

中码（2048）和 1.25V（2.5V 参考电压）之间的电压差叫做零偏移误差。DAC 偏移误差在 2.5V 参考电压下进行校准，并包含在 Device_cal()函数里加载到 DACTRIM 中。如果 DAC 不在 2.5V 参考电压下使用，为了确保偏移误差性能保持在限制范围内，必须调整偏移。DACTRIM [OFFSET_TRIM]是有符号的偏移调整。可以使用 DAC_tuneOffsetTrim()函数来调整偏移量。

28.5.3 PWMSYNCPER 信号

PWMSYNCPER 来自 PWM 的时基子模块。当设置 LOADMODE 位时，DACVALA 的 PWMSYNCPER 信号由电平触发加载。如果 PWM 的 TBCNT 和 TBPRD 都为 0，PWMSYNCPER 将保持在高电平，并且立即从 DACVALS 加载到 DACVALA。所以需要先配置 PWM 再设置 LOADMODE。

注意：GPDAC 从 PWM 接收的同步信号名称已经从 PWMSYNC 更新为 PWMSYNCPER，以避免与其他 PWM 同步信号混淆。

28.6 锁定寄存器

提供一个 DACLOCK 寄存器来防止假的写操作修改 DACCTL，DACVALS 和 DACOUTEN 寄存器。如果通过 DACLOCK 保护寄存器，直到设备被复位前写访问都将被锁定。

28.7 寄存器组地址

表格 113 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
DACARegs	DAC_REGS	0x5000_1800	0x5000_1BFF
DACBRegs	DAC_REGS	0x5000_1C00	0x5000_1FFF

28.8 寄存器地址映射

表格 114 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
DACREV	修订寄存器	0x00	-
DACCTL	控制寄存器	0x02	√
DACVALA	DAC 值活动寄存器	0x04	-
DACVALS	DAC 值影子寄存器	0x06	-
DACOUTEN	使能输出寄存器	0x08	√
DACLOCK	锁定寄存器	0x0A	√
DACTRIM	偏移微调寄存器	0x0C	√

28.9 寄存器功能描述

28.9.1 修订寄存器（DACREV）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	REV	R/W	修订 DAC（DAC Revision）	0h
15:8	保留			0h

28.9.2 控制寄存器（DACCTL）

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DACREFSEL	R/W	选择 DAC 参考电压（DAC reference voltage select） 0：VDAC/VSSA 为参考电压 1：ADC VREFHI/VSSA 为参考电压	0h
1	MODE	R/W	选择 DAC 增益模式（DAC gain mode select） 为缓冲输出选择增益模式。仅在设置 DACREFSEL 位且选择 ADC 内部参考模式时使用。 0：增益为 1 1：增益为 2	0h
2	LOADMODE	R/W	选择加载模式（DACVALA load mode select） 选择用 DACVALS 中的值更新 DACVALA 寄存器的时间。 0：加载下一个 APBCLK 1：加载 SYNCSEL 指定的下一个 PWMSYNCPER	0h
3	保留			0h
7:4	SYNCSEL	R/W	选择更新 DACVALA 的信号（update DACVALA signal Select） n 为设备可获得的最大 PWMSYNCPER 信号数： 0：PWM1SYNCPER 1：PWM2SYNCPER 2：PWM3SYNCPER … n-1：PWMnSYNCPER	0h
15:8	保留			0h

28.9.3 DAC 值活动寄存器（DACVALA）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVALA	R	当前由 DAC 驱动的活动代码（Active output code currently driven by the DAC）	0h
15:12	保留			0h

28.9.4 DAC 值影子寄存器（DACVALS）

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVALS	R/W	要加载到 DACVALA 中的影子输出代码（Shadow output code to be loaded into DACVALA）	0h
15:12	保留			0h

28.9.5 使能输出寄存器（DACOUTEN）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DACOUTEN	R/W	使能 DAC 输出（DAC output enable） 0：禁止 1：使能	0h
15:1	保留			0h

28.9.6 锁定寄存器（DACLOCK）

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DACCTL	R/WSonce	锁定对 DACCTL 的写访问（DACCTL register Lock write-access） 0：未锁定。将 0 写入此位无效 1：被锁定。只有系统复位才能清除这个位	0h
1	DACVAL	R/WSonce	锁定对 DACVALS 的写访问（DACVALS register Lock write-access） 0：未锁定。将 0 写入此位无效 1：被锁定。只有系统复位才能清除这个位	0h
2	DACOUTEN	R/WSonce	锁定对 DACOUTEN 的写访问（DACOUTEN register Lock write-access） 0：未锁定。将 0 写入此位无效 1：被锁定。只有系统复位才能清除这个位	0h
11:3	保留			0h
15:12	KEY	R-0/W	关键字（Key） 此位域为 0xA 时，才能写入这个寄存器。KEY 匹配后只有 16 位的写入会成功。read-modify-writes 对该寄存器中单个位的写入被忽略。	0h

28.9.7 偏移微调寄存器（DACTRIM）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	OFFSET_TRIM	R/W	DAC 偏移微调（DAC Offset Trim） 非特殊情况不修改此寄存器。否则可能导致该模块在规范外运行。	0h

位/域	名称	R/W	描述	复位值
15:8	保留			0h

29 比较器（COMP）

29.1 术语全称、缩写描述

表格 115 术语全称、缩写描述

中文全称	英文全称	英文缩写
比较器 DAC	Comparator DAC	Compdac

29.2 简介

比较器（COMP）模块集成了模拟比较器和相关电路，适用于电压跳变监控，开关模式电源，峰值电流模式控制和功率因数校正等应用场景。COMP 模块的每个子系统的由 2 个模拟比较器，1 个递减型斜坡发生器，2 个数字滤波器和 2 个 12 位 DAC 构成。

注意：在此章节中，COMP 表示比较器模块，而 comp 表示比较器模块子系统中的模拟比较器。

29.3 主要特征

（1） COMP 的每个子系统构成：
- 2 个模拟比较器
- 1 个递减型斜坡发生器
- 2 个数字滤波器
- 2 个 12 位 DAC

（2） DAC 基准电压可编程，可配置为 V_{DDA} 或 V_{DAC}

（3） 数字滤波器最大时钟预分频值为 2^{10}

（4） 与 APBCLK 信号同步输出

（5） 输入/输出功能
- （可选）迟滞输入
- 反相输出
- 锁存输出

（6） 比较器正/负输入
- 正输入端：由外部信号驱动
- 负输入端：由外部信号或基准 DAC 驱动

（7） PWMBLANK 信号，用于扩展清除信号

（8） PWMSYNC 信号，用于同步子模块

29.4 结构框图

图 52 COMP 结构框图

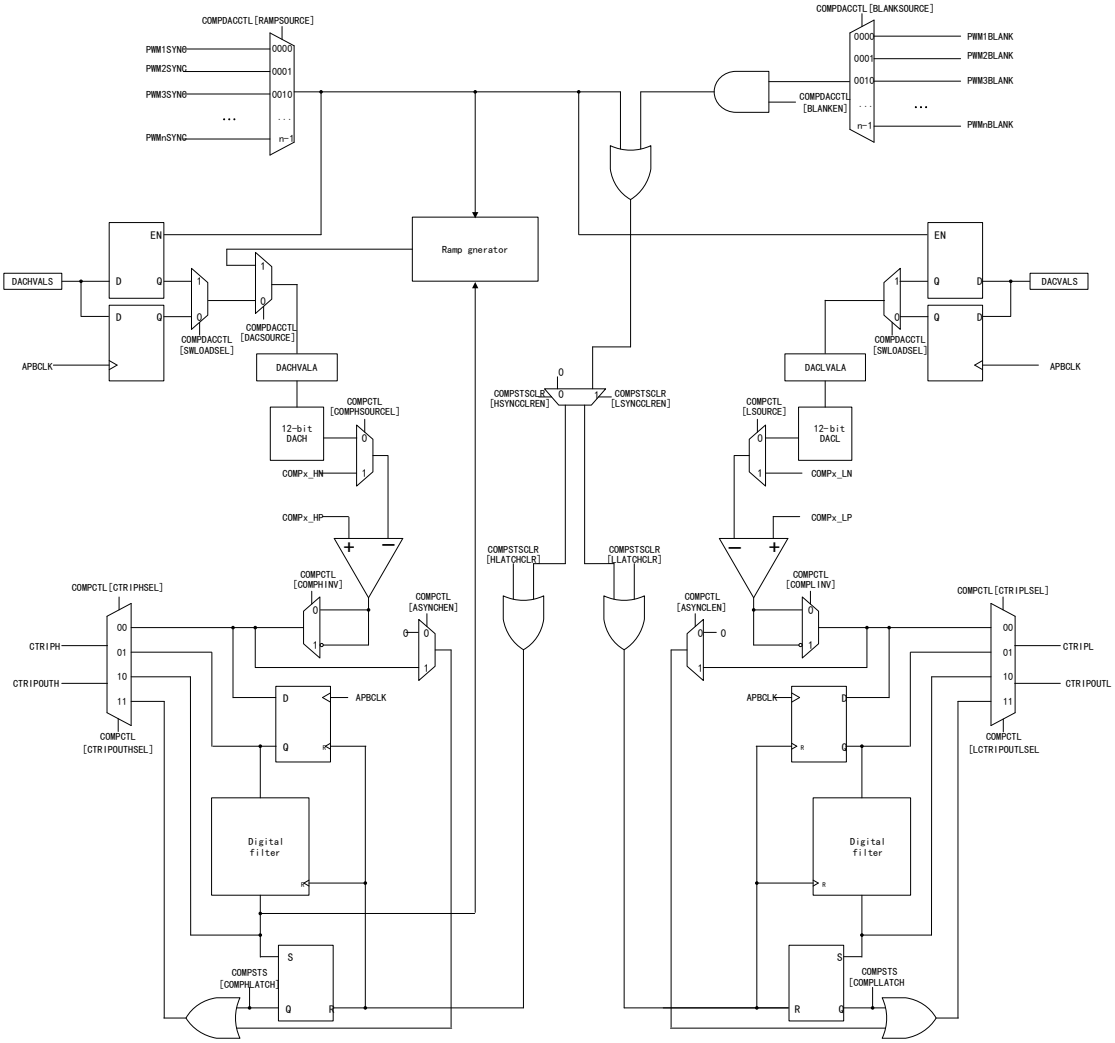
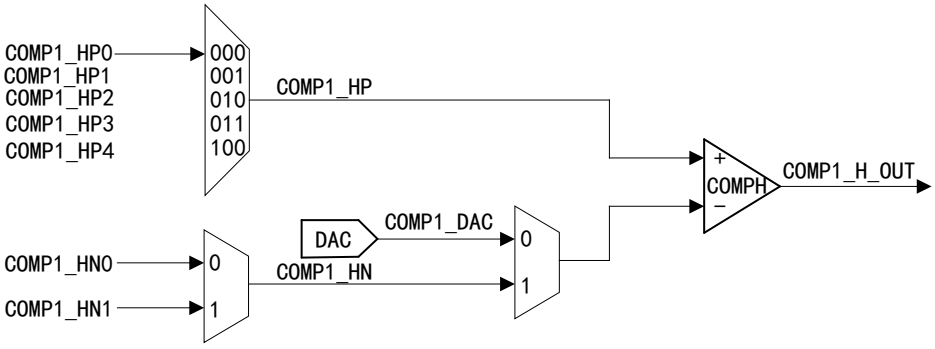


图 53 外部输入引脚选择（以 COMP1 为例）



如框图所示，每个 COMP 模块由多个子模块构成：

（1） 高位比较器和低位比较器

- “H”和“L”分别表示高位比较器和低位比较器；

- 比较器正输入端由外部引脚驱动，负输入端由外部引脚或 12 位可编程基准 DAC 驱动
- 每个比较器产生一个数字输出，用于指示正输入端的电压是否大于负输入端的电压。每个比较器输出通过一个可编程的数字滤波器，该滤波器可以去除偶然触发的信号。如果不需要滤波，也可以使用未滤波的输出。

（2） 2 个 12 位基准 DAC

（3） 2 个数字滤波器

（4） 1 个递减型斜坡发生器

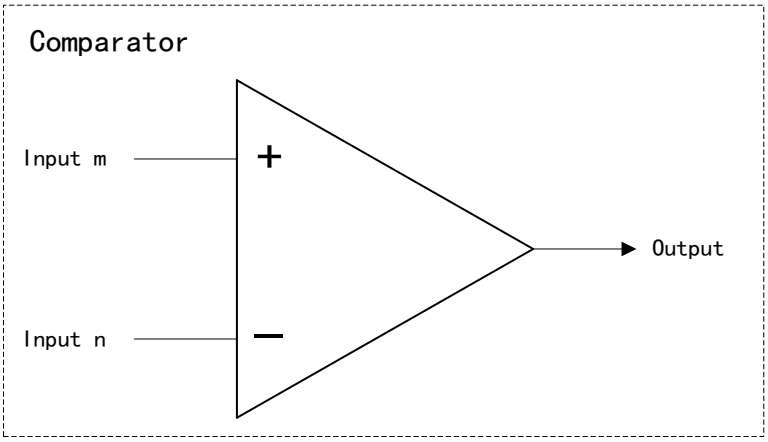
- （可选）用于控制高位比较器的 12 位基准 DAC 值

如上图所示，CTRIPH/L 信号与 PWM X-BAR 连接，用于 PWM 跳闸响应，CTRIPOUTH/L 信号与 Output X-BAR 连接，用于发出外部信号，有关 PWM X-BAR 和 Output X-BAR 多路复用配置信息，详情请参见 PWM 和 GPIO 章节。

29.5 功能描述

29.5.1 比较器基本结构

图 54 比较器基本结构



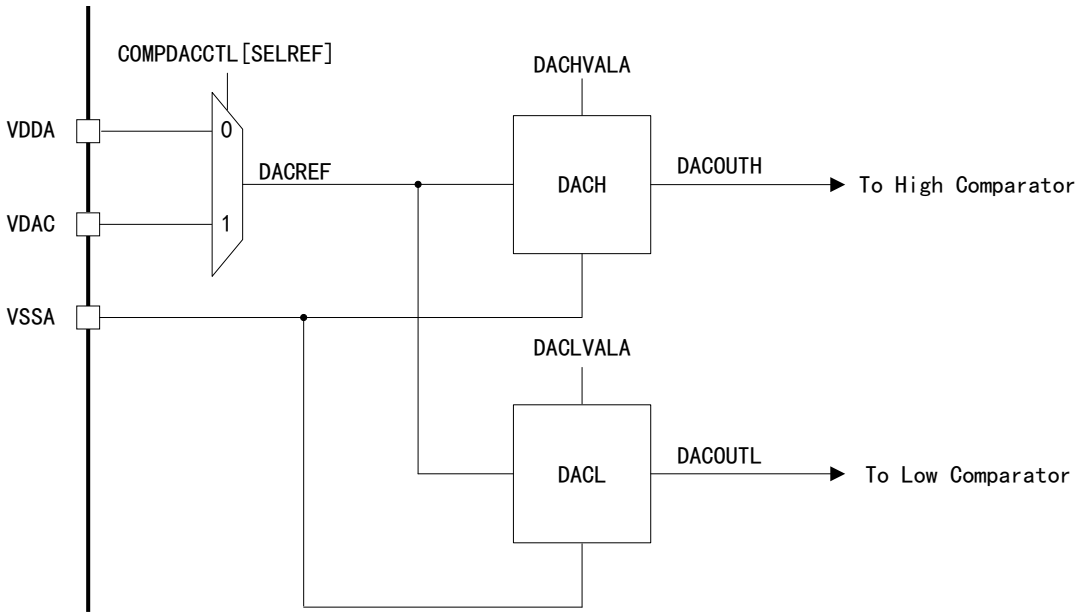
比较器基本结构如上图所示：

（1） 当比较器正输入端的电压大于负输入端的电压时，输出高电平，即 m＞n 时，Output=1；

（2） 当比较器正输入端的电压小于负输入端的电压时，输出低电平，即 m＜于 n 时，Output=0。

29.5.2 基准 DAC

图 55 内部 DAC 框图



（1） 内部基准电压输出：

- 每个 12 位基准 DAC 被配置用于生成一个基准电压，这个基准电压被用作相应比较器的负输入端；
- 12 位基准 DAC 的输出是内部的，因此内部基准电压输出不能从外部直接观察。

（2） 双寄存器组，DACxVALA 寄存器组和 DACxVALS 寄存器组（x 表示 H/L）：

- DACxVALS：DAC 影子寄存器是一个可写的寄存器，此寄存器的值可以立即装载到对应的 DACxVALA 寄存器中，也可以选择与下一个 PWMSYNC 事件同步载入 DACxVALA 寄存器

（3） DACx 输入源： DACx 的值可以选择从 DACxVALA 寄存器中获取，也可以选择从斜坡发生器中获取

（4） 12 位基准 DAC 工作范围：V_{DACREF}~ V_{SSA}

（5） 高基准电压默认值为 V_{DDA}，通过 COMPDACCTL[SELREF]位域配置，也可以选择 V_{DAC} 作为高基准电压。

（6） 理想条件下，12 位基准 DAC 输出的计算如以下公式所示：

$$Out_{DAC} = \frac{(1 + DACxVALA) * DACREF}{4096}$$

注意：可以在配置基准 DAC 值之前使能 COMP 模块实例。

29.5.3 斜坡发生器操作

通过设置 COMPDACCTL[DACSOURCE] =1，选择斜坡发生器作为高 12 位基准 DAC 的源。此时，斜坡发生器产生一个向下的斜坡输入供给高 12 位基准 DAC。

COMPDACCTL[DACSOURCE] =1 时，斜坡值寄存器（RAMPSTS）的值从斜坡最大基准值影子寄存器（RAMPMAXREFS）中装载并保持静态。一旦接收到选定的 PWMSYNC 信号后，在之后的每个 APBCLK 周期，都会从 RAMPSTS 中减去 RAMPMAXREFS 的值。

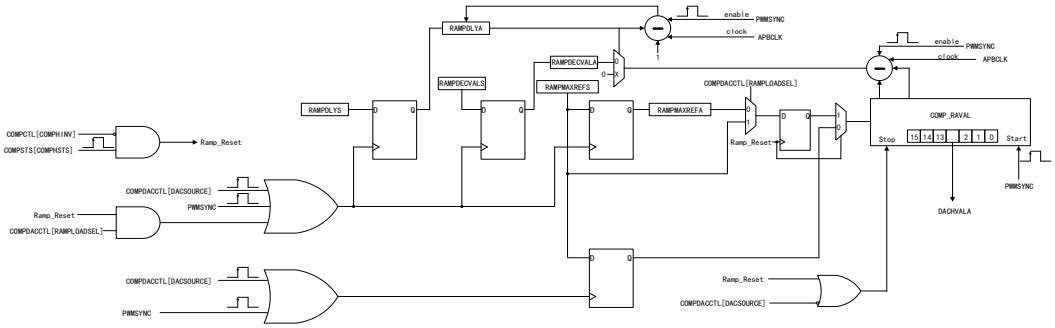
当选择斜坡发生器时，斜坡值寄存器（RAMPSTS）中的[15:4]位作为高 12 位基准 DAC 的输入，[3:0]位作为递减斜坡速率的预分频值，通过 RAMPDECVALA 寄存器进行配置。

延迟计数器

斜坡延迟值活动寄存器（RAMPDLYA）作为延迟计数器，用于控制开始减去 RAMPMAXREFS 的值的时间。当接收到选定的 PWMSYNC 信号后，在每个 APBCLK 周期 RAMPDLYA 递减 1，直到 RAMPDLYA 的值为 0 时，才开始从 RAMPSTS 中减去 RAMPMAXREFS 的值的操作。

29.5.3.1 正常情况下的操作

图 56 斜坡发生器



如上图所示，在每个 COMPDACCTL[DACSOURCE]，PWMSYNC 和 COMPSTS[COMPHSTS]的上升沿，斜坡发生器的状态都会改变：

表格 116 斜坡发生器操作

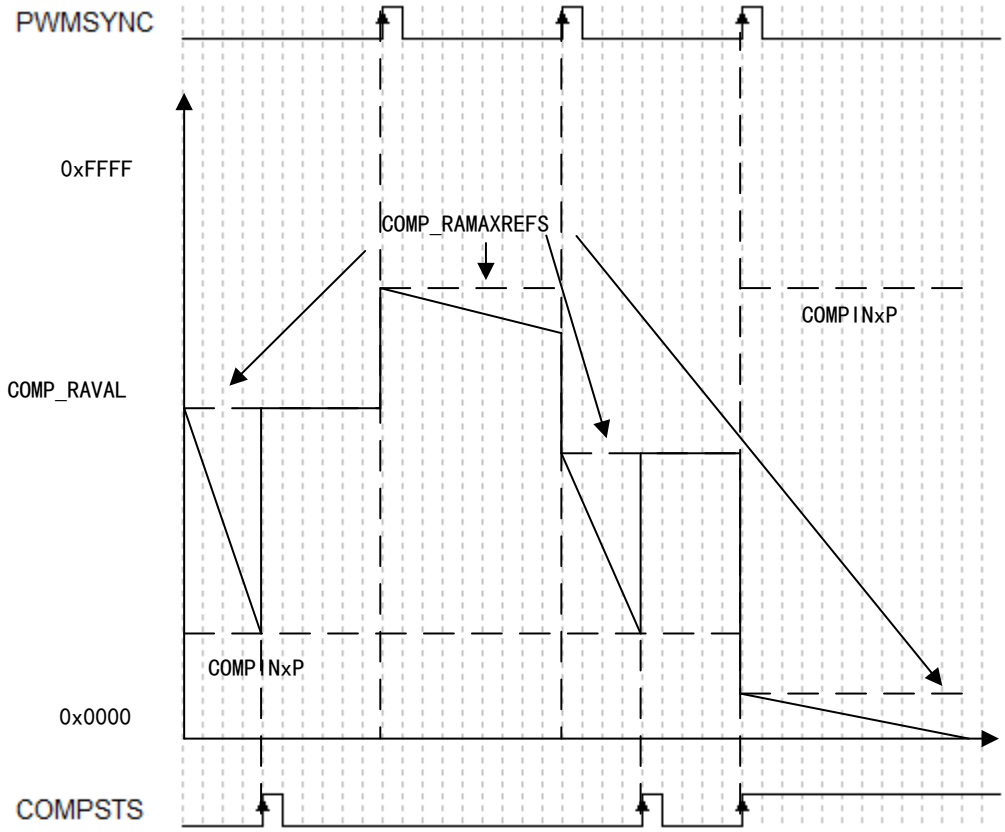
条件	RAMPMAXREFA, RAMPDECVALA 和 RAMPDLYA 寄存器	RAMPSTS 寄存器
COMPDACCTL[DACSOURCE]上升沿	从对应的影子寄存器中装载值	从 RAMPMAXREFS 寄存器中装载值
COMPSTS[COMPHSTS]上升沿，且 COMPDACCTL[RAMPLOADSEL] = 0	-	从 RAMPMAXREFA 寄存器中装载值，并停止递减
COMPSTS[COMPHSTS]上升沿，且 COMPDACCTL[RAMPLOADSEL] = 1	从对应的影子寄存器中装载值	从 RAMPMAXREFS 寄存器中装载值，并停止递减。

条件	RAMPMAXREFA, RAMPDECVALA 和 RAMPDLYA 寄存器	RAMPSTS 寄存器
PWMSYNC	从对应的影子寄存器中装载值	从 RAMPMAXREFS 寄存器中装载值，并在 RAMPDLYA 计数器为零时开始递减

此外，当 RAMPSTS 的值递减到 0 时，将静止为 0，直到接收到下一个选定的 PWMSYNC 信号。

29.5.3.2 极端情况下的操作

图 57 极端情况斜坡发生器操作



如上图所示，由于在每个 PWMSYNC 和 COMPSTS[COMPHSTS]的上升沿，斜坡发生器的状态都会改变，当这两个事件同时处于上升沿或上升沿时间非常接近时，会产生以下操作：

表格 117 极端情况操作

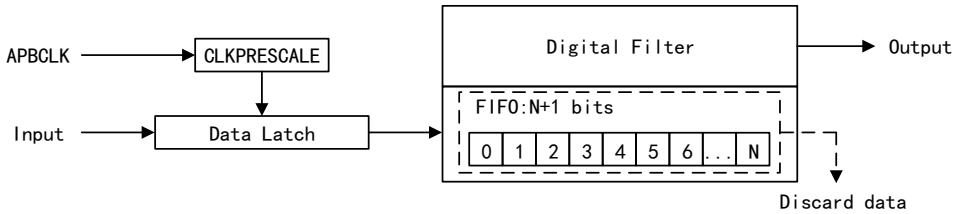
条件	操作
在 PWMSYNC 上升沿之前的一个或多个周期内发生 COMPSTS[COMPHSTS]上升沿	发生 COMPSTS[COMPHSTS]上升沿时，RAMPSTS 停止递减；当 RAMPDLYA 计数器递减至 0 时，RAMPSTS 寄存器在接收到 PWMSYNC 上升沿后开始递减。

条件	操作
同时发生	PWMSYNC 上升沿事件优先并忽略 COMPSTS[COMPHSTS]上升沿事件，RAMPSTS 寄存器不会停止递减。当 RAMPDLYA 计数器递减至 0 时，RAMPSTS 寄存器开始递减。
在 PWMSYNC 上升沿之后的一个或多个周期，且在 RAMPDLYA 计数器递减至 0 前发生 COMPSTS[COMPHSTS]上升沿	当 RAMPDLYA 计数器递减至 0 时，RAMPSTS 寄存器不会递减。
在 PWMSYNC 上升沿之后且在 RAMPDLYA 计数器递减至 0 时发生 COMPSTS[COMPHSTS]上升沿	RAMPSTS 寄存器不会递减。

29.5.4 数字滤波器

29.5.4.1 概念模型

图 58 数字滤波器概念模型



数字滤波器概念模型如上图所示。

采样窗口

数字滤波器工作基于一个 FIFO 采样窗口（SAMPWIN），该采样窗口从滤波器输入中进行取样，内部采样窗口的实际大小为 SAMPWIN+1。

滤波器输出

滤波器输出为采样窗口中的多数阈值（即出现次数最多的值），在 THRESH 中定义了多数阈值，只有当采样窗口内至少出现 THRESH 个与当前输出状态相反的样本时，滤波器输出状态才会改变，否则滤波器输出保持不变。内部多数阈值的实际值为 THRESH+1。

多数阈值条件

为了保证滤波器可以正常运行，多数阈值必须满足以下条件：

$$\frac{\text{SAMPWIN}}{2} < \text{THRESH} \leq \text{SAMPWIN}$$

滤波器采样速率

滤波器采样速率由 CLKPRESCALE 定义，滤波器 FIFO 每隔预分频时钟进行一次采样操作，预分频实际值为 CLKPRESCALE+ 1。滤波器将丢弃 FIFO 中的旧数据以便于存储新数据。

示例：

如概念模型所示，当设置 SAMPWIN 为 8，则内部采样窗口值为 9，设置多数阈值为 4，则内部阈值为 5。

（1） 当采样窗口内输入“1”的个数为 7 时，假设为 110111110，则滤波器输出状态改变，输出“1”；

（2） 当采样窗口内输入“1”的个数为 5 时，假设为 110101100，则滤波器输出状态改变，输出“1”；

（3） 当采样窗口内输入“1”的个数为 3 时，假设为 011000010，则滤波器输出状态不变，输出“0”。

29.5.4.2 滤波器初始化顺序

（1） 配置并使能比较器

（2） 配置滤波器参数：配置采样窗口（SAMPWIN），多数阈值（THRESH）和时钟预分频值（CLKPRESCALE）

（3） 初始化数字 FIFO 窗口中的样本值：设置 FILINIT，将所有样本值初始化为滤波器输入值。

（4） 清除 COMPSTS 锁存：当需要锁存路径时，使用 COMPSTSCLR 寄存器清除 COMPSTS 锁存。

（5） 配置 CTRIP 和 CTRIPOUT 信号路径

（6） 配置目标模块（如果需要）：根据需求配置目标模块，如 GPIO 模块等，以接收过滤后的信号。

29.5.5 延迟

此模块增加的延迟如下：

（1） 同步块：增加 1~2 个时钟的延迟；

（2） 数字滤波器：当数字滤波器处于被旁路的状态时（即所有的数字滤波器都被置 0），会增加 2 个时钟的延迟；

（3） 锁存器：增加 1 个时钟的延迟。

29.5.6 LATCLR，PWMSYNC 和 PWMBLANK 信号

LATCLR 信号

在所需的延迟后，LATCHCLR 信号用于将数字滤波器、同步块以及锁存器的输出保持在复位状态。

LATCHCLR 信号可以通过软件激活，也可以通过 PWMSYNC 信号激活。

- 软件激活：通过设置 xLATCLR 激活
- PWMSYNC 信号激活：当设置了 xSYNCCLR 时，可以通过 PWMSYNC 信号激活 LATCLR 信号。

延长 LATCLR 信号：当需要更长时间的 LATCLR 信号时，可以通过设置 BLANKEN，使用 PWMBLANK 信号来延长 LATCLR 信号。

PWMSYNC 和 PWMBLANK 信号

PWMSYNC 信号由 PWM 模块的时基子模块产生，PWMBLANK 信号由 PWM 模块的数字比较子模块产生，详情请参见 PWM 模块的时基子模块功能特征和数字比较子模块功能特征章节。

当 COMPDACCTL [SWLOADSEL]置 1 时，加载 DACxVALA 的 PWMSYNC 信号由电平触发，如果 PWMSYNC 信号一直处于高电平状态（时基计数器和时基周期都是 0），那么不管此位是否置 1，DACxVALA 都会立即从 DACxVALS 加载。因此，建议先配置 PWM，再将 COMPDACCTL [SWLOADSEL]置 1。

注意：有关 PWMSNC 信号描述的相关信息，详情请参见 PWM 章节。

29.5.7 COMP 校准

29.5.7.1 偏移误差源

COMP 模块中，有两种偏移误差源，有关误差的实际值，详情请参见数据手册。

（1） 比较器偏移误差：在数据手册中称为“输入偏移误差”，当比较器的两个输入都由一个引脚驱动时，只考虑比较器偏移误差；

（2） 比较器 DAC 偏移误差（以下称为 compdac 偏移误差）：在数据手册中称为“静态偏移误差”，当比较器的反相输入由 compdac 驱动时，由于 compdac 偏移误差包括比较器偏移误差，因此只考虑 compdac 偏移误差。

29.5.7.2 校准

由于上述两种误差的存在，因此需要对 COMP 模块进行校准，以确保在预期的水平上正确跳闸。

当比较器的反相输入由 compdac 驱动时，通过向下扫描 compdac，校准流程如下：

（1） 设置 compdac 初始值：将 compdac 的值设置为最大，即 0xFFF；

- （可选）如果已知非反相输入端的近似直流电压（Vtarget），则可以将 compdac 初始值设置为 Vtarget +静态偏移误差+ Margin 以快速进行校准。其中，Margin 表示一定程度的保护余量。当 Vtarget 未知时，可以通过 ADC 转换 Vtarget。

（2） compdac 值递减：compdac 值递减 1；

（3） 等待 compdac 稳定：在 compdac 值递减后，等待其输出稳定；

（4） 清除锁存

（5） 等待锁存器是否置位

（6） 检查锁存状态：当锁存置位时，表明找到了跳闸代码，流程结束。

- （可选）通过以下步骤对跳闸代码进行验证：compdac 值递加 1，清除锁存，等待锁存是否置位，此时锁存应处于未置位状态，以验证跳闸代码正确。

（7） 当锁存未置位时，返回步骤（2），直到找到正确的跳闸代码。

注意：

（1） 校准前，需确保比较器的非反相输入端有一个静态直流信号。

（2） 校准期间，应禁用比较器的迟滞功能，并在校准完成后重新启用。

（3） 输入噪声处理：如果非反相输入端的信号有噪声，可以使用具有非零滤波器的锁存器并根据噪声程度设置适当的滤波参数，以比较校准过程中受到噪声干扰。

当比较器的两个输入都由一个引脚驱动时，也可以对 COMP 模块进行校准，在这种情况下，流程保持不变，但需要在比较器的反相输入引脚上外部施加扫描电压。

29.5.8 COMP 时钟

比较器激活时，如果禁用 COMP 模块时钟，COMP 模块中各结构将进行以下操作：

（1） 比较器：即使禁用 COMP 模块时钟，比较器仍然会根据输入电压的变化触发，不受禁用时钟的影响。

（2） 斜坡发生器，同步块和数字滤波器：在当前状态下冻结

（3） 12 位基准 DAC：当 12 位基准 DAC 驱动比较器的负输入端时，负输入端上的电压保持静态不变，不受禁用时钟的影响，但 DACLVALA 不能从斜坡发生器或 DACLVALS 中更新数据。

29.6 寄存器组地址

表格 118 COMP 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Comp1Regs	COMP_REGS	0x4001 1C00	0x4001 1FFF
Comp2Regs	COMP_REGS	0x4001 2000	0x4001 23FF
Comp3Regs	COMP_REGS	0x4001 2400	0x4001 27FF
Comp4Regs	COMP_REGS	0x4001 2800	0x4001 2BFF
Comp5Regs	COMP_REGS	0x4001 2C00	0x4001 2FFF
Comp6Regs	COMP_REGS	0x4001 3000	0x4001 33FF
Comp7Regs	COMP_REGS	0x4001 3400	0x4001 37FF

29.7 寄存器地址映射

表格 119 COMP_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
COMPCTL	控制寄存器	0x00	√
COMPHYSCTL	迟滞控制寄存器	0x02	√
COMPSTS	状态寄存器	0x04	-
COMPSTSCLR	状态清除寄存器	0x06	√
COMPDACCTL	DAC 控制寄存器	0x08	√
DACHVALS	高位 DAC 影子寄存器	0x0C	-
DACHVALA	高位 DAC 活动寄存器	0x0E	-
RAMPMAXREFA	斜坡最大基准值活动寄存器	0x10	-
RAMPMAXREFS	斜坡最大基准值影子寄存器	0x14	-
RAMPDECVALA	斜坡递减值活动寄存器	0x18	-
RAMPDECVALS	斜坡递减值影子寄存器	0x1C	-
RAMPSTS	斜坡值寄存器	0x20	-
DACLVALS	低位 DAC 影子寄存器	0x24	-
DACLVALA	低位 DAC 活动寄存器	0x26	-
RAMPDLYA	斜坡延迟值活动寄存器	0x28	-
RAMPDLYS	斜坡延迟值影子寄存器	0x2A	-
CTRIPLFILCTL	低位滤波控制寄存器	0x2C	√
CTRIPLFILCLKCTL	低位滤波时钟控制寄存器	0x2E	√
CTRIPHFILCTL	高位滤波控制寄存器	0x30	√
CTRIPHFILCLKCTL	高位滤波时钟控制寄存器	0x32	√
COMPLOCK	锁定寄存器	0x34	√

29.8 寄存器功能描述

29.8.1 控制寄存器（COMPCTL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	COMPHSOURCE	R/W	选择高比较器输入源（High comparator input source select） 此位选择比较器的反向输入源。 0：选择内部 DAC 驱动的比较器反相输入 1：选择外部引脚驱动的比较器反相输入	0h

位/域	名称	R/W	描述	复位值
1	COMPHINV	R/W	使能高比较器反向输出（High comparator invert output enable） 0：禁用（不反相） 1：使能（反相）	0h
3:2	CTRIPHSEL	R/W	选择高比较器 PWM 源（High comparator PWMH source select） 选择驱动 PWMH 的输出。 00：异步比较器 01：同步比较器 10：数字滤波器 11：锁存的数字滤波器	0h
5:4	CTRIPOUTHSEL	R/W	选择高比较器 OUTH 源（High comparator OUTH source select） 选择驱动 OUTH 的输出。 00：异步比较器 01：同步比较器 10：数字滤波器 11：锁存的数字滤波器	0h
6	ASYNCHEN	R/W	使能高比较器异步输出发送（High comparator asynchronous output transmit enable） 异步比较器输出和一个锁存的数字滤波器信号输出连接至一个 OR 逻辑门，此位域用于控制当 CTRIPHSEL =11 或 CTRIPOUTHSEL =11 时，是否允许异步比较器的输出结果反馈至 OR 逻辑门。 0：异步比较器的输出不会馈入到 OR 逻辑门 1：异步比较器的输出会反馈到 OR 逻辑门	0h
7	保留			0h
8	COMPLSOURCE	R/W	选择低比较器输入源（Low comparator input source） 0：选择内部 DAC 驱动的比较器反相输入 1：选择外部引脚驱动的比较器反相输入	0h
9	COMPLINV	R/W	使能低比较器反向输出（Low comparator invert output enable） 0：禁用（不反相） 1：使能（反相）	0h
11:10	CTRIPLSEL	R/W	选择低比较器 PWML 源（Low comparator PWML source select） 00：异步比较器 01：同步比较器 10：数字滤波器 11：锁存的数字滤波器	0h
13:12	CTRIPOUTLSEL	R/W	选择低比较器 OUTL 源（Low comparator OUTL source select） 00：异步比较器 01：同步比较器 10：数字滤波器 11：锁存的数字滤波器	0h

位/域	名称	R/W	描述	复位值
14	ASYNCLEN	R/W	使能低比较器异步输出发送（Low comparator asynchronous output transmit enable） 异步比较器输出和一个锁存的数字滤波器信号输出连接至一个 OR 逻辑门，此位域用于控制当 CTRIPLSEL =11 或 CTRIPOUTLSEL =11 时，是否允许异步比较器的输出结果反馈至 OR 逻辑门。 0：异步比较器的输出不会馈入到 OR 逻辑门 1：异步比较器的输出会反馈到 OR 逻辑门	0h
15	COMPDACE	R/W	使能比较器/DAC（Comparator/DAC enable） 0：禁用 1：使能	0h

29.8.2 迟滞控制寄存器（COMPHYSCTL）

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	COMPHYS	R/W	选择比较器迟滞（Comparator hysteresis select） 选择比较器的迟滞值。 0000：无迟滞 0001：迟滞值为典型值 0010：迟滞值为典型值*2 0011：迟滞值为典型值*3 0100：迟滞值为典型值*4 其他：未定义	0h
15:4	保留			0h

29.8.3 状态寄存器（COMPSTS）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	COMPHSTS	R	高比较器数字滤波器输出标志（High comparator digital filter output Flag）	0h
1	COMPHLATCH	R	高比较器数字滤波器输出的锁存值（High comparator digital filter output Latched value）	0h
7:2	保留			0h
8	COMPLSTS	R	低比较器数字滤波器输出标志（Low comparator digital filter output Flag）	0h
9	COMPLLATCH	R	低比较器数字滤波器输出的锁存值（Low comparator digital filter output Latched value）	0h
15:10	保留			0h

29.8.4 状态清除寄存器（COMPSTSCLR）

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	HLATCHCLR	R-0/W1S	清除高比较器锁存（High comparator latch clear） 此位用于软件清除 COMPSTS[COMPHLATCH]。 0：无影响 1：产生一个脉冲信号，以清除 COMPSTS[COMPHLATCH]	0h
2	HSYNCCLREN	R/W	清除高比较器锁存 PWMSYNC（High comparator latch PWMSYNC clear） 此位用于控制是否允许 PWMSYNC 清除 COMPSTS[COMPHLATCH] 0：PWMSYNC 不会清除 COMPSTS[COMPHLATCH] 1：允许 PWMSYNC 清除 COMPSTS[COMPHLATCH]	0h
8:3	保留			0h
9	LLATCHCLR	R-0/W1S	清除低比较器锁存（Low comparator latch clear） 此位用于软件清除 COMPSTS[COMPLLATCH]。 0：无影响 1：产生一个脉冲信号，以清除 COMPSTS[COMPLLATCH]	0h
10	LSYNCCLREN	R/W	清除低比较器锁存 PWMSYNC（Low comparator latch PWMSYNC clear） 此位用于控制是否允许 PWMSYNC 清除 COMPSTS[COMPLLATCH] 0：PWMSYNC 不会清除 COMPSTS[COMPLLATCH] 1：允许 PWMSYNC 清除 COMPSTS[COMPLLATCH]	0h
15:11	保留			0h

29.8.5 DAC 控制寄存器（ COMPDACCTL）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DACSOURCE	R/W	选择 DAC 源（DAC source select） 选择更新到 DACHVALA 的源。 0：由 DACHVALS 更新 DACHVALA 1：由斜坡发生器更新 DACHVALA	0h

位/域	名称	R/W	描述	复位值
4:1	RAMPSOURCE	R/W	选择 PWMSYNC 源（PWMSYNC source select） 选择用于 COMP 模块的 PWMSYNC 信号 0000：PWM1SYNC 0001：PWM2SYNC 0010：PWM3SYNC …… n：PWM(n-1)SYNC 其中，n（十进制）表示可用 PWMSYNC 信号的最大值	0h
5	SELREF	R/W	选择 DAC 的基准电压（DAC reference voltage Select） 选择内部比较器 DAC 的基准电压源。 0：VDDA 1：VDAC	0h
6	RAMPLOADSEL	R/W	选择斜坡加载（Ramp load select） 当触发 COMPSTS[COMPHSTS]后，选择更新到 RAMPSTS 的源。 0：RAMPMAXREFA 1：RAMPMAXREFS	0h
7	SWLOADSEL	R/W	选择软件加载（Software load select） 选择更新到 DACxVALA(x=H/L)的源。 0：APBCLK 更新 DACxVALS 1：PWMSYNC 信号更新 DACxVALS	0h
11:8	BLANKSOURCE	R/W	选择 PWMBLANK 源（PWMBLANK source select） 选择作为 PWMBLANK 信号的 PWMnBLANK 0000：PWM1BLANK 0001：PWM2BLANK 0010：PWM3BLANK …… n：PWM(n-1) BLANK 其中，n（十进制）表示可用 PWMSYNC 信号的最大值	0h
12	BLANKEN	R/W	使能 PWMBLANK（PWMBLANK enable） 使能 PWMBLANK 信号。 0：禁用 1：使能	0h
13	保留			0h
15:14	FREESOFT	R/W	选择斜坡发生器运行方式（ramp generator run mode select） 在仿真暂停时，选择斜坡发生器的运行方式。 00：斜坡发生器立即停止运行 01：斜坡发生器完成当前斜坡并在下一个 PWMSYNC 停止 其他：斜坡发生器自由运行，不受仿真暂停的影响	0h

29.8.6 高位 DAC 影子寄存器（DACHVALS）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVAL	R/W	高位 DAC 影子值（High DAC shadow value） 当 COMPDACCTL[DACSOURCE]=0 时，DACHVALS 寄存器的值在 COMPDACCTL[SWLOADSEL]选择的触发信号上加载到 DACHVALA 寄存器中。	0h
15:12	保留			0h

29.8.7 高位 DAC 活动寄存器（DACHVALA）

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVAL	R	高位 DAC 活动值（High DAC Active value） 由高位 DAC 主动驱动的值。	0h
15:12	保留			0h

29.8.8 斜坡最大基准值活动寄存器（RAMPMAXREFA）

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RAMPMAXREF	R	斜坡最大基准活动值（Ramp maximum reference active value） 待加载到斜坡发生器（即 RAMPSTS 寄存器）中的锁存值。	0h

29.8.9 斜坡最大基准值影子寄存器（RAMPMAXREFS）

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RAMPMAXREF	R/W	斜坡最大基准影子值（Ramp maximum reference shadow value） 待加载到斜坡发生器（即 RAMPSTS 寄存器）中的未锁存值。	0h

29.8.10 斜坡递减值活动寄存器（RAMPDECVALA）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RAMPDECVAL	R	斜坡递减活动值（Ramp decrement value active） 将从 RAMPSTS 中减去的锁存值。	0h

29.8.11 斜坡递减值影子寄存器（RAMPDECVALS）

偏移地址：0x1C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RAMPDECVAL	R/W	斜坡递减影子值（Ramp decrement value shadow） 待加载到 RAMPDECVALA 中的未锁存值。	0h

29.8.12 斜坡值寄存器（RAMPSTS）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	RAMPVALUE	R	斜坡值（Ramp value） 斜坡发生器的当前值。	0h

29.8.13 低位 DAC 影子寄存器（DACLVALS）

偏移地址：0x24

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVAL	R/W	低位 DAC 影子值（Low DAC shadow value） DACVAL 寄存器的值在 COMPDACCTL[SWLOADSEL] 选择的触发信号上加载到 DACLVALA 寄存器中。	0h
15:12	保留			0h

29.8.14 低位 DAC 活动寄存器（DACLVALA）

偏移地址：0x26

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	DACVAL	R	低位 DAC 活跃值（Low DAC Active value） 由低位 DAC 主动驱动的值。	0h
15:12	保留			0h

29.8.15 斜坡延迟值活动寄存器（RAMPDLYA）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
12:0	DELAY	R	斜坡延迟活动值（Ramp delay active value） 在接收到 PWMSYNC 信号后，斜坡发生器开始递减操作前必须等待的时钟周期数的锁存值。	0h
15:13	保留			0h

29.8.16 斜坡延迟值影子寄存器（RAMPDLYS）

偏移地址：0x2A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
12:0	DELAY	R/W	斜坡延迟影子值（Ramp delay shadow value） 待加载到 RAMPDLYA 寄存器中的未锁存值。	0h
15:13	保留			0h

29.8.17 低位滤波控制寄存器（CTRIPLFILCTL）

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	保留			0h
8:4	SAMPWIN	R/W	低位滤波器采样窗口尺寸（Low filter sample window size） 待监测的样本数量实际值为 SAMPWIN+1。	0h
13:9	THRESH	R/W	低位滤波器多数阈值（Low filter majority threshold） 为了改变输出状态，在采样窗口内必须至少出现 THRESH 个与当前输出状态相反的样本。 阈值数量的实际值为 THRESH+1。	0h
14	保留			0h
15	FILINIT	R-0/W1S	低位滤波器初始化（Low filter initialization） 0：无影响 1：将所有样本值初始化为滤波器输入值	0h

29.8.18 低位滤波时钟控制寄存器（CTRIPLFILCLKCTL）

偏移地址：0x2E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	CLKPRESCALE	R/W	低滤波器采样时钟预分频（Low filter sample clock prescale） 采样间隔的时钟数的实际值为 LFLTCLKPSC+1。	0h
15:10	保留			0h

29.8.19 高位滤波控制寄存器（CTRIPHFILCTL）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	保留			0h
8:4	SAMPWIN	R/W	高位滤波器采样窗口尺寸（High filter sample window size） 待监测的样本数量实际值为 SAMPWIN+1。	0h
13:9	THRESH	R/W	高位滤波器多数阈值（High filter majority threshold） 为了改变输出状态，在采样窗口内必须至少出现 THRESH 个与当前输出状态相反的样本。 阈值数量的实际值为 THRESH+1。	0h
14	保留			0h
15	FILINIT	R-0/W1S	高位滤波器初始化（High filter initialization） 0：无影响 1：将所有样本值初始化为滤波器输入值	0h

29.8.20 高位滤波时钟控制寄存器（CTRIPHFILCLKCTL）

偏移地址：0x32

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	CLKPRESCALE	R/W	高滤波器采样时钟预分频（High filter sample clock prescale） 采样间隔的时钟数的实际值为 CLKPRESCALE+1。	0h
15:10	保留			0h

29.8.21 锁定寄存器（COMPLOCK）

偏移地址：0x34

复位类型：SYSRSn

对此寄存器的[3:0]位写 0 没有作用。

位/域	名称	R/W	描述	复位值
0	COMPCTL	R/WSO	锁定 COMPCTL 的写访问（write-access to the COMPCTL register Lock） 0：不锁定 COMPCTL 的配置 1：锁定 COMPCTL 的配置 此位只能通过系统复位清除。	0h
1	COMPHYSCTL	R/WSO	锁定 COMPHYSCTL 的写访问（write-access to the COMPHYSCTL register Lock） 0：不锁定 COMPHYSCTL 的配置 1：锁定 COMPHYSCTL 的配置 此位只能通过系统复位清除。	0h
2	DACCTL	R/WSO	锁定 COMPDACCTL 的写访问（write-access to the COMPDACCTL register Lock） 0：不锁定 COMPDACCTL 的配置 1：锁定 COMPDACCTL 的配置 此位只能通过系统复位清除。	0h
3	CTRIP	R/WSO	锁定 CTRIPxFILTCTL 和 CTRIPxFILCLKCTL 的写访问（write-access to the COMPCTL register Lock） 0：不锁定 CTRIPxFILTCTL 和 CTRIPxFILCLKCTL 的配置 1：锁定 CTRIPxFILTCTL 和 CTRIPxFILCLKCTL 的配置 此位只能通过系统复位清除。	0h
15:4	保留			0h

30 Sigma-Delta 滤波器（SDF）

30.1 术语全称、缩写描述

表格 120 术语全称、缩写描述

中文全称	英文全称	英文缩写
Sigma Delta 滤波器	Sigma Delta Filter	SDF
过采样率	OverSampling Ratio	OSR
数据滤波器过采样率	Data Filter Oversampling Ratio	DOSR
比较器滤波器过采样率	Comparator Filter Over-Sampling Ratio	COSR
有效位数	Effective Number of Bits	ENOP

30.2 简介

SDF 是一个设计用于电机控制系统的高级数字滤波装置，拥有四个独立的输入通道。在电机控制应用中，这些输入通道通常用于电流测量和解算器位置解码，每个输入通道都能独立地接收来自 Sigma Delta 调制器的比特流。具备四个各自可编程的数字解调滤波器，这些滤波器能够精确调整以适应各种信号处理要求。此外，SDF 还配备了一个快速比较器作为次级滤波器，它可以迅速执行数字阈值比对，用于即时监测过电流和欠电流，以及信号的过零检测。

30.3 主要特征

（1） SDF 拥有四个可独立配置的主滤波器单元：
- 可以使能或禁用滤波器模块；
- 提供 4 种的滤波器类型选择，包括 Sinc1/Sinc2/SincFast/Sinc3；
- 数据滤波单元的过采样率（OSR，DOSR）设置在 1 到 256；
- 可以通过主滤波器使能位或 PWM 信号同步四个独立滤波器；

（2） SDF 模块包含四个可独立配置的二级滤波器单元：
- 可以进行超值、低值和阈值交叉检测；
- 提供 4 种的滤波器类型选择，包括 Sinc1/Sinc2/SincFast/Sinc3；
- 比较器滤波单元的过采样率（OSR，COSR）设置在 1 到 32 之间；

（3） SDF 模块具有 8 个外部引脚，分别为四个数据输入引脚和四个时钟输入引脚：
- 数据输入引脚接收从 Sigma-Delta 数据信号；
- 时钟输入引脚接收 sigma-delta 时钟信号；

（4） PWM 信号还可以用来为 Sigma-Delta 调制器生成所需的调制器时钟；

（5） 数据滤波器单元配备了一个可编程的 FIFO 模式；

（6） SDF 允许使用 PWM 信号作为同步数据滤波器通道的来源；

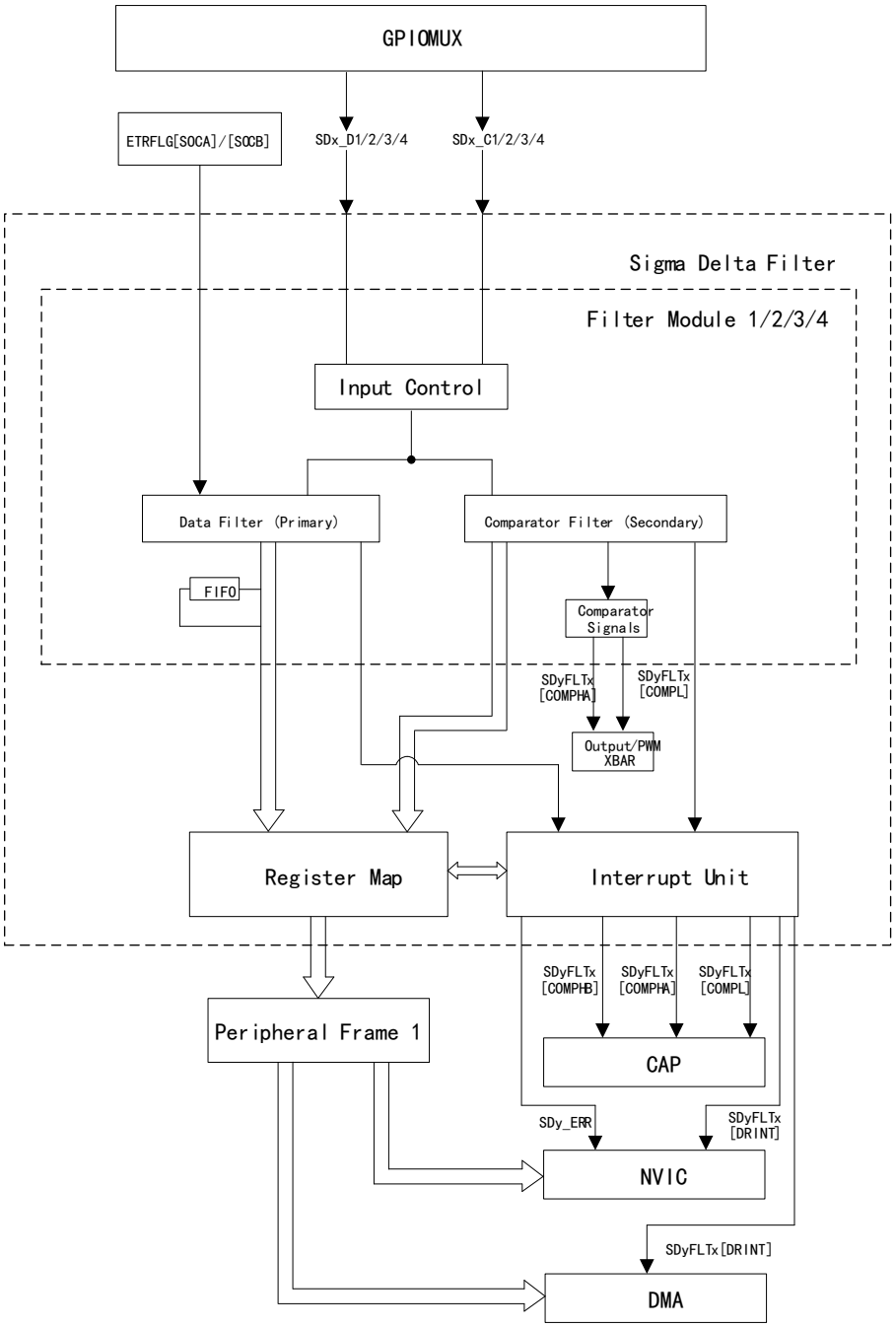
（7） SDF 支持不同的调制器时钟工作模式，以适应不同的应用需求：

- 模式 0：调制器的时钟频率与数据频率相同
- 模式 1：调制器的时钟频率是数据频率的一半
- 模式 2：采用曼彻斯特编码的调制器数据，不需额外的时钟信号
- 模式 3：调制器的时钟频率是数据频率的两倍

30.4 结构框图

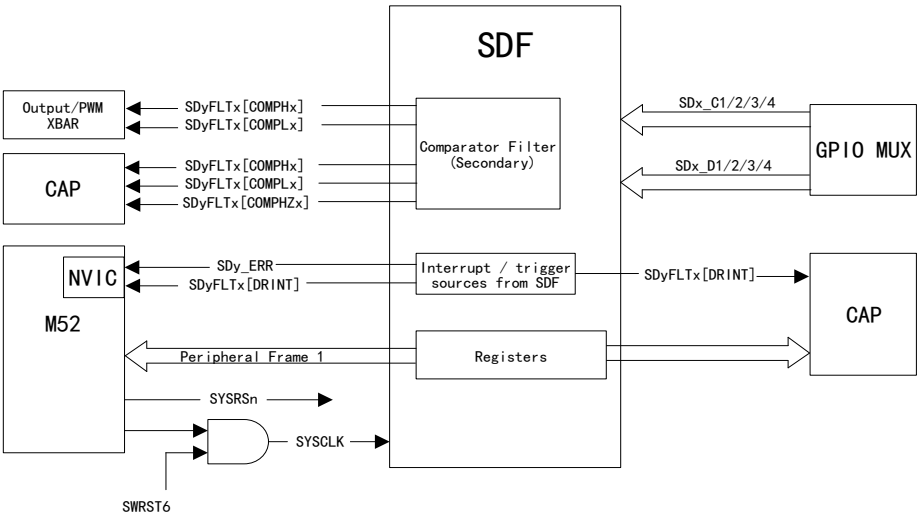
SDF 端口的操作是由寄存器进行配置和管理的，每个 SDF 模块由四个独立的滤波单元组成，每个滤波单元包括输入控制单元、数据滤波器（主）、比较器滤波器（次），其中比较器滤波器配备 4 个独立的比较器。

图 59 SDF 结构框图



在 SDF 模块中共有四个数据滤波器和四个比较器滤波器，其中每个滤波单元都包含了一个数据滤波器和一个接收相同比特流的比较器滤波器，除此以外数据滤波器和比较器滤波器是完全独立的。

图 60 SDF CPU 接口



30.5 功能描述

30.5.1 GPIO 配置

复用 GPIO 引脚可将将此外设连接到设备引脚上。为了要避免引脚上产生毛刺，需要按照以下步骤进行配置：

（1）设置 GPxGMUX 寄存器，此时对应的 GPxMUX 位域保持默认值零

（2）将 GPxMUX 寄存器设置为目标值

在 SDF 正常工作模式下，建议将 GPIO 输入限定设置为 SYNC，以提供对偶发随机噪声干扰的附加硬件保护；如果 GPIO 输入限定设置为 ASYNC，容易受到随机噪声的干扰。使用 SYNC / ASYNC 都应该检查 SDF 的电气数据和时序要求。不支持其他 GPIO 输入限定。

注意：

（1） SDF 时钟输入（SDx_Cy 引脚）提供了对 SDF 模块的直接时钟信号，因此对于这些输入的任何干扰或振铃噪声可能会影响 SDF 模块的操作。为确保信号干净无噪声并满足 SDF 时序要求，需采取特别的预防措施，如串联终端以减少振铃，或将信号线与其他噪声信号进行隔离；

（2） SDF 模块通常期望数据信号（SD-Dx）在时钟信号（SD-Cx）的下降沿发生变化，在上升沿探测 SD-Dx。如果使用的 SD 调制器在上升沿改变 SD-Dx，并在下降沿探测 SD-Dx，那么可以使用 GPIO 反相特性（GPxINV）改变 SD-Cx 引脚上的极性，以实现与 SDF 的兼容；

（3） SYNC 可防止 SDx_Cy 引脚上偶然的随机噪声干扰，避免错误触发比较器和滤波器输出。然而，SYNC 选项不能保护持续违反时序要求的情况，这种违规行为会导致数据损坏，程度取决于违规的严重性。

（4） 有关 GPIO 复用和设置的更多细节，详情请参阅 GPIO 章节。

30.5.2 Sigma-Delta 调制器数据接收与处理

输入控制单元的作用是接收 sigma-delta 调制数据和 sigma-delta 调制时钟，这些数据在被接收后将被发送给数据过滤单元和比较器单元进行处理。为了能够适应不同的输入条件和系统要求，输入控制单元可以根据 SDCTLPARMx[MOD]位的配置，按照以下四种不同的模式来接收调制数据：

（1） 模式 0：调制器时钟频率与数据率相同，调制器数据在每个调制器时钟的上升沿被锁存；

（2） 模式 1：调制器时钟频率是数据率的一半，调制器数据在调制器时钟的上升沿和下降沿被锁存；

（3） 模式 2：调制器时钟关闭，调制器数据使用曼彻斯特编码；

（4） 模式 3：调制器时钟是调制器数据率的两倍，调制器数据在每两个调制器时钟的正向沿被锁存。

图 61 调制器模式 0 操作

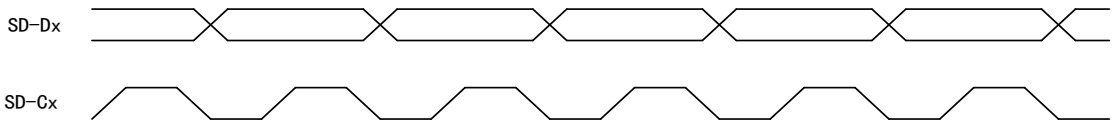


图 62 调制器模式 1 操作

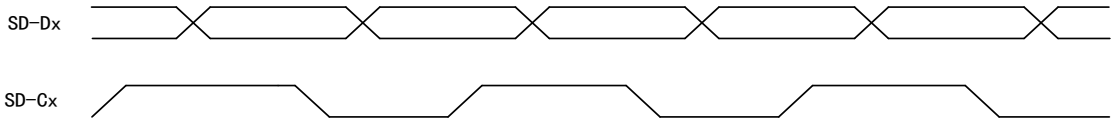


图 63 调制器模式 2 操作

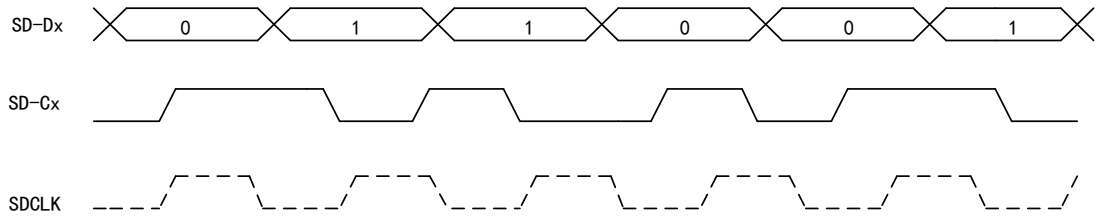
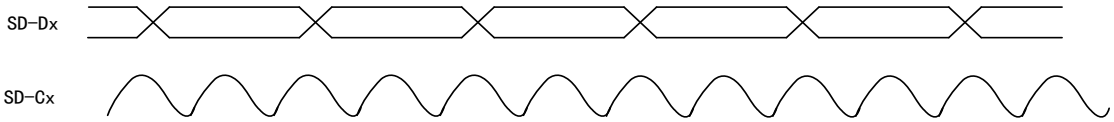


图 64 调制器模式 3 操作



注意：

（1） 为了保证调制器能够达到最大的动态范围，可能需要将调制器运行在其绝对最大的满量程，这通常超出了推荐的 80%的最大满量程范围。

（2） 如上图的模式 2 操作所示，数据信号和调制时钟信号被编码进调制数据，此时时钟输入引脚（SD-Cx）引脚不需要连接任何信号。输入控制单元执行连续自动校准，保证能够以最优的方式解码接收到的信号。

30.5.3 输入控制单元 Sinc

在 SDF 中，无论是比较器滤波器还是数据滤波器，它们的核心均采用了 $Sinc^N$ 类型的滤波器。$Sinc^N$ 滤波器在本质上作为低通滤波器，其功能是将输入的比特流通过数字滤波和抽取，进而转换成更易于处理的数字数据。这种经过滤波的数字数据能够代表 Sigma-Delta 调制器原本的模拟输入信号。下图分别展示了 $Sinc^N$ 滤波器和 N 阶 Sinc 滤波器的 Z-传递函数，其中 $Sinc^N$ 滤波器的结构相对简单，它由一系列的积分器和微分器顺序排列而成，两者之间由下采样器隔开，以降低信号的采样频率。

图 65 简化 SincN 滤波器结构

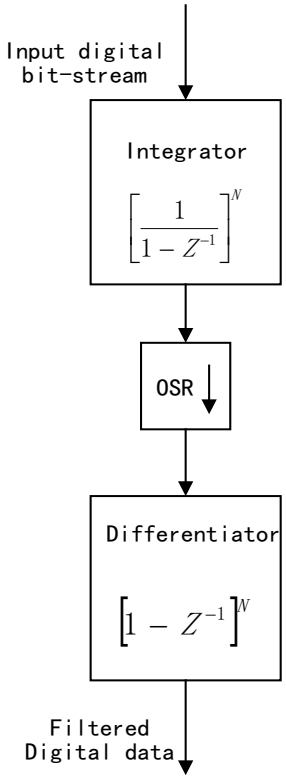


图 66 N 阶 Sinc 滤波器的 Z-传递函数

$$H(Z)=\left[\frac{1-Z^{-OSR}}{1-Z^{-1}}\right]^N$$

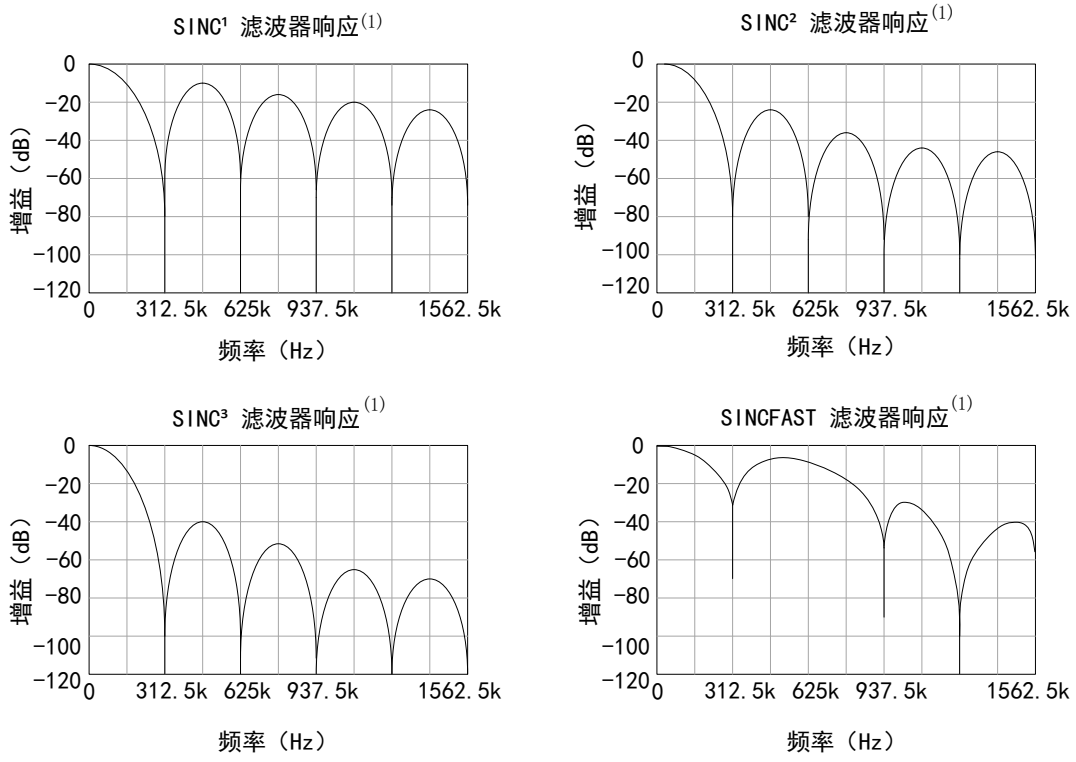
其中 N 代表滤波器的阶数，OSR 代表过采样率

Sinc 滤波器的有效分辨率（即 ENOB）会受到 OSR、滤波器种类以及 sigma-delta 调制器的工作频率的影响。一般来说，提升过采样率可以在特定的滤波器设计下实现更高分辨率或获得更多 ENOB。但是，增加 OSR 会延长滤波器响应时间，这是设计中需要平衡的因素。

在挑选 sigma-delta 调制器的过程中，分析并找到处理速率和分辨率之间的最佳平衡点是至关重要的。为了准确知道在某一特定的 Sinc 滤波器设置下能够达到的有效分辨率，我们需要参考该 sigma-delta 调制器的技术规格说明书。

下图展示了当过采样率设定为 32，以及 sigma-delta 调制器的频率为 10 兆赫时，不同滤波器结构的频率响应特性，其中 $SINC^1$代表 1 阶 SINC 滤波器，$SINC^2$ 代表 2 阶 SINC 滤波器，$SINC^3$ 和 SincFast 代表 3 阶 SINC 滤波器。

图 67 不同 Sinc 滤波器的频率响应



注意：（1）f_{DATA} = 312.5KHz = $\frac{f_{CLK}}{OSR}$ = $\frac{10MHz}{32}$

Sinc 滤波器的数据速率

Sinc 滤波器的数据速率（即滤波器的处理速度），单位为采样样本数/秒，可以根据下面的公式得出：

Sinc 滤波器的数据传输率 = 调制器数据速率 / OSR

Sinc 滤波器的延迟

Sinc 滤波器的延时是指滤波器从开始处理到输出准确过滤结果所需的时间长度，以秒为单位。对于给定的滤波器，其延时公式为：

Sinc 滤波器的延时 = Sinc 滤波器的阶数 / Sinc 滤波器的数据速率

示例配置

表格 121 示例配置

Sinc 滤波器类型	调制器数据速率	OSR	Sinc 滤波器的数据速率	Sinc 滤波器延迟
$sinc^3$	10 MHz	256	$\frac{10MHz}{256}$ = 39.1 K 采样样本数/秒	76.8μs

Sinc 滤波器类型	调制器数据速率	OSR	Sinc 滤波器的数据速率	Sinc 滤波器延迟
$sinc^2$	10 MHz	256	$\frac{10MHz}{256}$ = 39.1 K 采样样本数/秒	51.2μs

30.5.4 数据滤波器（主级）

数据滤波器是一种可配置的 Sinc 滤波器，它能够进行不同程度的信号处理，具体取决于它的类型（$Sinc^1$、$Sinc^2$、$Sinc^3$ 或 SincFast）。OSR 是数据滤波器的关键设置，它可以在 1 到 256 之间调整，与比较滤波器独立；数据滤波器的 ENOB 是衡量滤波器输出信号质量的指标，取决于滤波器类型、OSR 以及 sigma-delta 调制器的工作频率。

该数据滤波器默认是禁止的，但可以通过设置参数 SDDFPARMx[FEN]的值为 1 来启用它。一旦启用，滤波器输出的结果是一个 26 位的有符号整数，遵循二进制补码格式表示法，这意味着它可以表示正数和负数值。根据输入信号的高低，该滤波器将其转换为数值“1”或“-1”。下表列出了不同 OSR 设置下数据滤波器的最大存储值。

关于如何计算数据滤波器的数据速率和延迟请参阅该章节的 Sinc 滤波器部分。

表格 122 不同 DOSR/滤波器组合的峰值数据

DOSR	$Sinc^1$	$Sinc^2$	$Sinc^3$	SincFast
x	x	x^2	x^3	$2x^2$
4	-4 到 4	-16 到 16	-64 到 64	-32 到 32
8	-8 到 8	-64 到 64	-512 到 512	-128 到 128
16	-16 到 16	-256 到 256	-4096 到 4096	-512 到 512
32	-32 到 32	-1024 到 1024	-32768 到 32768	-2048 到 2048
64	-64 到 64	-4096 到 4096	-262144 到 262144	-8192 到 8192
128	-128 到 128	-16384 到 16384	-2097152 到 2097152	-32768 到 32768
256	-256 到 256	-65536 到 65536	-16777216 到 16777216	-131072 到 131072

30.5.4.1 数据 FIFO

每个主数据滤波通道包含了一个可以存储 32 位数据的 FIFO，它的深度为 16 级。这个队列的设计允许它先暂存一系列数据滤波器的输出样本，然后一次性通知处理器数据准备就绪，而不是对每个样本都发出一个中断。这种方法可以有效减少中断的频率，降低处理器处理中断的负担，从而更高效地控制数据传输。

在默认状态下禁用 FIFO 队列。如果要启用这个队列，需要修改配置 SDFIFOCTLx[FFEN]位为 1。启用之后，每当数据滤波器有新的数据可用，这个数据就会被送入 FIFO 队列，并且 FIFO 队列的当前状态会实时反映在 SDFIFOCTLx [SDFFST]位中。

设置 FIFO（接收到预设数量的数据就绪事件后触发中断）

（1） 设置 SDFIFOCTLx[FFEN]位为 1，启动 FIFO

（2） 设置 SDFIFOCTLx[FFIEN]位为 1，启动 FIFO 中断

（3） 配置 SDFIFOCTLx[SDFFIL]位字段值为 0-16

（4） 设置 SDIFLG[SDFFINTx]为 1，配置 SDF 数据就绪事件在 FIFO 中断时触发

（5） 设置 SDFIFOCTLx [DRINTSEL]位为 1，配置 SDIFLG[SDFFINT]位为数据就绪中断源：

- 当 SDFIFOCTLx[SDFFST] ≥ SDFIFOCTLx[SDFFIL]时，SDIFLG[SDFFINT]位被置位并在数据就绪中断源生成中断
- 设置 SDIFLGCLR[SDFFINTx]位可清除 SDIFLG[SDFFINTx]标志

等待同步特性

等待同步特性允许系统在产生特定的同步事件之前，暂时忽略或不处理来自数据滤波器的数据就绪事件。这个特定的同步事件通常是由 PWM 发出的信号，称为 SDSYNC 事件。

（1） 当等待同步特性禁用时：

- 默认情况下，禁用等待同步特性。等待同步特性禁用时，每当有新的数据就绪事件发生时，FIFO 都会收到数据，直到它被填满或 SDFIFOCTLx[SDFFST]大于等于 SDFIFOCTLx[SDFFIL]。

（2） 当等待同步特性启用时：

- FIFO 不会随着每个数据就绪事件而被填充，除非接收到 SDSYNC 事件；
- 当检测到 SDSYNC 事件时，FIFO 设置 SDSYNCx[WTSYNFLG]位为 1，此时 FIFO 接收来自主过滤器的数据，直到 FIFO 被填满或 SDFIFOCTLx[SDFFST]大于等于 SDFIFOCTLx[SDFFIL]；
- SDSYNCx[WTSYNFLG]位可以设定为自动清零或手动清零，清零后 FIFO 存在的数据会被冻结，后续的数据就绪事件不会填充 FIFO，直到下一个 SDSYNC 事件。
  - WTSYNCFLG 自动清除模式：默认情况下为该模式。当 SDSYNCx[WTSCLREN]为 1 时，WTSYNFLG 会在 SDFFINT 事件发生时自动清除。
  - WTSYNCFLG 手动清除模式：设定 SDSYNCx [WTSYNCLR] 位为 1 来手动清除 WTSYNFLG。

清空 FIFO 内容

可以通过以下任一方式来清空 FIFO 内容：

（1） 禁用 FIFO 可以清空 FIFO 内容（将 SDFIFOCTLx[FFEN] 位置 0）

（2） 禁用主过滤器可以清空 FIFO 内容（将 SDDFPARMx[FEN]位置 0 或 SDMFILEN[MIE] 位置 0）

（3） 接收到 SDSYNC 事件会自动清空 FIFO 内容（将 SDSYNCx[FFSYNCCLREN]位置 1 来启用该功能，默认情况下禁用该功能）。

注意：上述功能只有在等待同步特性（SDSYNCx[WTSCLREN] 为 1）开启时才可用。

调试访问 SDDATFIFOx 寄存器不会影响 FIFO 指针。在 CPU/RTDMA 访问 SDDATFIFOx 寄存器时，FIFO 读取指针会自动前进到下一个条目。

30.5.4.2 数据过滤器输出格式

数据过滤器的输出默认为 16 位格式。

16 位数据滤波器表示

若 SDDPARMx[DR] 位为 0，则数据滤波器的输出为 16 位格式。由于数据过滤器的 26 位带符号输出范围从–16777216 到 16777216，而 16 位带符号输出的范围仅为–32768 到 32767。因此用户需调整 SDDPARM[SH]位，以决定 32 位字中的哪 16 位部分输出到寄存器映射。

下表展示了不同 OSR 和滤波器类型下 SDDPARM[SH]位的配置设置。需要注意的是若 SDDPARM[SH]位配置不当，会得到错误的 16 位数据过滤器输出。

表格 123 SDDPARM[SH]位的配置设置

OSR 范围	Sinc1	Sinc2	Sinc3	SincFast
1 - 31	0	0	0	0
32 - 40	0	0	1	0
41 - 50	0	0	2	0
51 - 63	0	0	3	0
64 - 80	0	0	4	0
81 - 101	0	0	5	0
102 - 127	0	0	6	0
128 - 161	0	0	7	1
162 - 181	0	0	8	1
182 - 203	0	1	8	2
204 - 255	0	1	9	2
256	0	2	10	3

32 位数据滤波器表示

若 SDDPARMx[DR] 位为 1，数据过滤器的输出采用 32 位格式，此时改变 SDDPARM[SH]并不会影响数据过滤器的最终输出。

30.5.4.3 与 PWM 事件同步配置

主数据过滤器可以与 PWM 事件同步，也称为 SDSYNC 事件。PWM 模块的

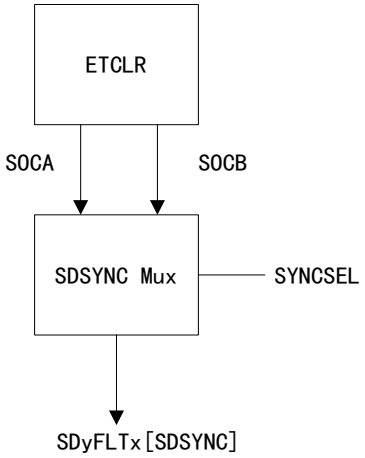
SDSYNC 信号用于重置 DOSR 计数器。默认情况下禁用此同步功能，但用户可以通过将 SDDFPARMx[SDSYNCEN]设置为 1 来启用。每个主数据过滤器都能够与任意一个可用的 PWMx SOCA/SOCB 信号同步，如下表所示。

表格 124 SDSYNCx[SYNCSEL]对应输入信号

SDSYNCx[SYNCSEL]	输入信号
0	PWM1_SOCA
1	PWM1_SOCB
4	PWM2_SOCA
5	PWM2_SOCB
8	PWM3_SOCA
9	PWM3_SOCB
12	PWM4_SOCA
13	PWM4_SOCB
16	PWM5_SOCA
17	PWM5_SOCB
20	PWM6_SOCA
21	PWM6_SOCB
24	PWM7_SOCA
25	PWM7_SOCB
28	PWM8_SOCA
29	PWM8_SOCB
2-3，6-7，10-11，14-15，18-19，22-23，26-27，30-63	保留

通过 SDSYNCx[SYNCSEL]位的配置，用户可以选择哪一个 PWM 信号来为主过滤器提供 SDSYNC 脉冲。下图展示了 PWM 信号如何与 SDF 模块连接。

图 68 SDSYNC 事件



注意：需要确保在 PWM 的每个定时周期内只产生一个 SDSYNC 事件。这是为了防止在 PWM 的双向计数模式下产生两个同步事件，这样的情况会导致 SDF 的定时出现问题。在 PWM 的单向计数模式（只向上数或只向下数）下，自然只会产生一个 SDSYNC 事件，从而避免了问题的发生。

Sinc 滤波器在初次启用、在运行中被重新启用、重新配置时或者接收到来自 PWM 的 SDSYNC 信号时，会产生错误的数据样本。这是因为 Sinc 滤波器的本身设计的特性决定了其最初几个输出样本可能不准确。不同类型的 Sinc 滤波器（Sinc1、Sinc2、Sinc3、SincFast）在这方面的表现不同，例如，Sinc1 滤波器没有错误样本，而 Sinc2 和 Sinc3 滤波器的第一个样本、SincFast 滤波器的前两个样本会存在错误。

注意：在比较器滤波器足够稳定之后，才能启用 SDF 的比较器中断。这是为了防止由于上述提到的错误样本而意外触发比较器滤波器。启用比较器中断之前，需要等待一个足够的延迟时间，以确保比较器滤波器已达到稳定状态。这个延迟时间是通过将比较器滤波器的固有延迟与额外的 5 个 SD-Cx 时钟周期相加来计算的。

30.5.5 比较器滤波器（次级）

大多数控制系统需要通过中断 PWM 信号来防止电流或电压超出安全极限时的潜在系统损害。这是通过使用比较器滤波器（次级）实现的，其主要功能是允许快速监测输入条件，从而在必要时能够迅速切断 PWM 信号以保护系统，但该滤波器不能与 PWM 事件同步。

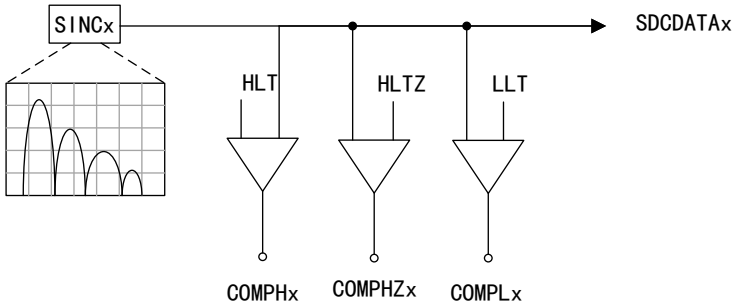
比较器滤波器是一种可配置的 Sinc 滤波器，它支持 Sinc1、Sinc2、Sinc3 和 SincFast 这几种滤波器类型。默认情况下禁用比较器滤波器，可以通过设置 SDCPARMx[CEN]位为 1 来启用。它的过采样率（COSR）可以设置在 1 到 32 之间，这一设置与数据滤波器是独立的。比较器滤波器的有效解析度（ENOB）取决于滤波器类型、COSR 和 sigma-delta 调制频率。比较器滤波器的输出是 16 位无符号格式，其中低输入信号被解释为 0，高输入信号被解释为 1，这意味着比较器滤波器的输出总是正值。下表展示了不同 OSR 设置下比较器滤波器能存储的最大值，可参考本章节中 Sinc 滤波器小节了解计算比较器滤波器的数据速率和延迟的方法。

表格 125 不同 OSR/滤波器组合的峰值数据

OSR	Sinc1	Sinc2	Sinc3	SincFast
x	0 - x	0 - x^2	0 - x^3	0 - $2x^2$
4	0 - 4	0 - 16	0 - 64	0 - 32
8	0 - 8	0 - 64	0 - 512	0 - 128
16	0 - 16	0 - 256	0 - 4096	0 - 512
32	0 - 32	0 - 1024	0 - 32768	0 - 2048

比较器滤波器的输出通过内存映射到 SDCDATAx 寄存器，该寄存器每经过 COSR 指定的 SD-Cx 周期数更新一次。比较器滤波器的数字输出与数字比较器相连，后续部分将会进一步解释这些数字比较器的功能和操作。

图 69 比较器单元结构



30.5.5.1 低阈值比较器

当监测到的信号值低于或等于预设的低阈值时，系统会生成一个低阈值事件。这些事件可以被配置用来触发一系列的响应动作，包括 CPU 中断，或者是为了保护系统而切断 PWM。除此之外，低阈值事件还可以与 CAP 模块一起使用，这样可以测量信号经过这个低阈值点时的频率或者占空比。

设备内部有一个低阈值比较器：

（1） LLT 比较器：在 SDCTL[MIE] 设置为 1 且 SDCPARMx[IEL] 也设置为 1 时，会触发 SDF 中断，并且设置 SDIFLG[IFLx]标志，这个标志表明系统检测到了低于设定阈值的情况。

如果中断的原因（即信号值低于低阈值）不再存在，可以通过设置 SDIFLGCLR 寄存器的相应位来清除这个标志。

30.5.5.2 高阈值比较器

高阈值比较器是一种监控设备，用于检测特定信号是否超过了预设的上限阈值。当监测到的数据超过或等于设定的高阈值时，系统将自动触发高阈值事件。这些事件不仅可以用来生成 CPU 中断，提醒系统采取进一步的措施，还可以触发 PWM 切断，这在需要快速断电以保护设备时特别有用。此外，这些事件也可以用来与 CAP 模块配合，以测量信号的频率或占空比。

设备内部有两个高阈值比较器，分别用于不同的监测目的：

（1） HLT 比较器：通过 SDCMPHx[HLT]位进行配置，当 SDCTL[MIE]设为 1 且 SDCPARMx[IEH]设为 1 时，高阈值事件会触发 SDF 中断，并将 SDIFLG[IFHx]标志置位，表明发生了超限条件。当相应的 SDIFLGCLR 寄存器位被置位并且中断源不再存在时，这个标志可以被清除。

（2） HLTZ 比较器： 当比较器的读数超过 SDCMPHZx [HLTZ]时，它能够触发一个 COMPHZx 事件并置位相应的 SDSTATUS[HZx]标志，但此事件不会生成 SDF 中断。这种比较器特别适用于与 CAP 模块一起使用，测量信号阈值变化的频率或占空比。

如要测定信号过零点的频率或者占空比，需要进行以下配置：

- 将 SDCMPHZx [HLTZ]位的值设为 0x4000，设定高电平阈值交叉的触发条件；
- 将 SDCPARMx[HZEN]设置为 1，激活高电平（B）阈值交叉功能；

- 为了确保可以测量频率或占空比，需要对 ECCTL0[INPUTSEL]位进行配置，使其能够把 COMPHZx 信号送往 CAP 模块。

30.5.6 SDF 滤波器理论输出

下面的公式可以用来推导出一个比较器滤波器和数据滤波器的 SDF 滤波器输出的理想滤波器输出。

$$\text{比特流中 1 的密度} = \frac{\text{输入电压} + V_{clipping}}{2 \times V_{clipping}} \qquad (1)$$

其中，Vclipping =调制器最大差分电压输入范围

$$\text{比较器滤波器输出理论值} = \text{比特流中 1 的密度} \times \text{最大滤波器输出} \qquad (2)$$

其中，最大滤波器输出取决于滤波器的类型和过采样率（COSR）

$$\text{滤波器输出} = \left\{\frac{\text{输入电压绝对值}}{Vclipping}\right\} \times \text{最大滤波器输出} \qquad (3)$$

其中，最大滤波器输出取决于滤波器的类型和过采样率（DOSR）

$$\text{数据滤波器输出 32 位理论值} = \begin{cases} \text{滤波器输出（输入电压为正电压）} \\ \text{滤波器输出的 2 的补码（输入电压为负电压）} \end{cases} \qquad (4)$$

$$\text{数据滤波器输出 16 位理论值} = \text{数据滤波器输出 32 位理论值右移} \qquad (5)$$

其中，右移值取决于滤波器类型和 OSR

示例

当使用 AMC1306x25 调制器，其中滤波器类型为 3，输入电压=100mV，Vclipping =320mV，COSR=32，DOSR=100 时：

根据方程（1）可求出：比特流中 1 的密度=0.65625；

根据方程（2）可求出：比较器滤波器输出=21504；

根据方程（3）、（4）可求出：数据滤波器输出 32 位理论值=312500；

根据方程（5）可求出：数据滤波器输出 32 位理论值=9765

30.5.7 SDF 中断

每个 SDF 能够针对每一个滤波模块产生五种 CPU 中断，包括 SDF 错误（SDy_ERR）中断和 SDF 数据就绪（SDy_DRINT1，SDy_DRINT2，SDy_DRINT3，SDy_DRINT4）中断。

30.5.7.1 SDF 错误中断源

下图展示了 SDy_ERR 中断的构成，SDy_ERR 中断可以由下图列出的 16 种事件中的任何一种激发。

高阈值比较器事件（COMPHx）

任一比较器滤波器模块的 COMPH 事件能够触发 CPU 的中断。要使该事件只触发 SDy_ERR 中断，必须进行以下设置：

（1） 设置 SDCTL [MIE]位为 1，激活主中断使能；

（2） 设置 SDCPARMx [IEH]位为 1，激活比较器滤波器的高阈值中断使能。如果发生 COMPH 事件，将会设置 SDIFLG [IFHx]的相应标志位。为了清除这个标志位，必须在相应的 SDIFLGCLR 寄存器中设置对应的位，并确保触发该中断的条件已经消除。

低阈值比较器事件（COMPLx）

任一比较器滤波器模块的 COMPL 事件能够触发 CPU 的中断。要使该事件只触发 SDy_ERR 中断，必须进行以下设置：

（1） 设置 SDCTL [MIE]位为 1，激活主中断使能；

（2） 设置 SDCPARMx [IEL]位为 1，激活比较器滤波器的高阈值中断使能。如果发生 COMPH 事件，将会设置 SDIFLG [IFLx]的标志位会被设定。的相应标志位。为了清除这个标志位，必须在相应的 SDIFLGCLR 寄存器中设置对应的位，并确保触发该中断的条件已经消除。

调制器故障事件

调制器时钟丢失会触发调制器故障事件（即 SD-Cx 信号在 127 个 SDFCLK 时钟周期内没有发生变化）。任一比较器滤波器模块的 COMPL 事件能够触发 CPU 的中断。要使该事件只触发 SDy_ERR 中断，必须进行以下设置：

（1） 设置 SDCTL[MIE]位为 1，使能主中断

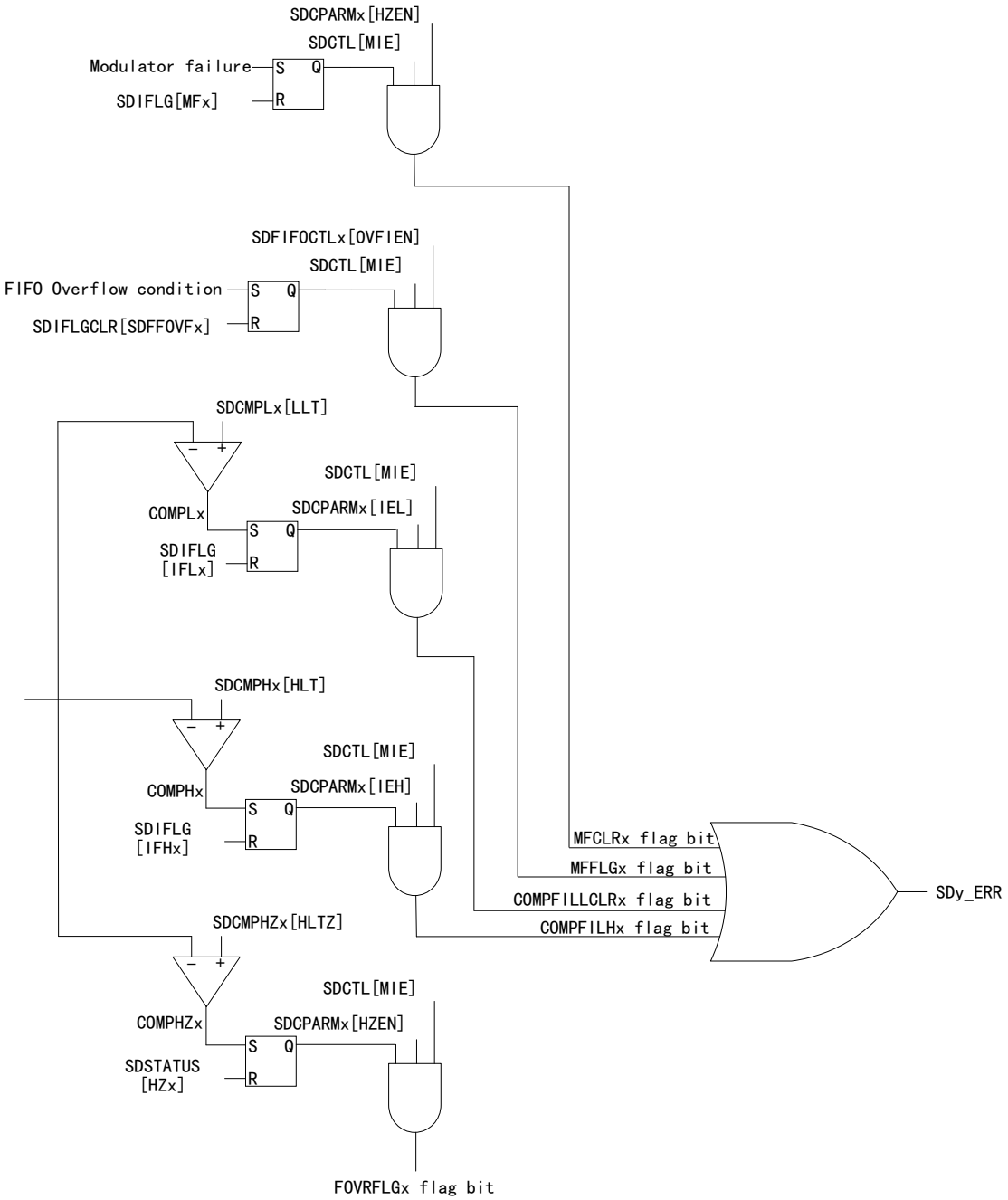
（2） 设置 SDCPARMx[MFIE]位为 1，使能调制器时钟失效为中断源

调制器故障事件发生后会设置标志位 SDIFLG[MFx]。为了清除这个标志位，必须在相应的 SDIFLGCLR 寄存器中设置对应的位，并确保触发该中断的条件已经消除。

30.5.7.2 SDF 数据就绪中断源

每个 SDy_DRINTx 中断由相应的数据滤波器通道触发，该中断结构如下图所示。

图 70 SDy_ERR 中断源



注意：高阈值（Z）比较器事件不能配置为触发 SDy_ERR 中断信号。相反，这个事件被内部连接到 CAP1 的 MUX1，用于频率/占空比测量的阈值交叉事件。

30.6 寄存器组地址

表格 126 SDF 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Sdf1Regs	SDF_REGS	0x4000_3C00	0x4000_3FFF

30.7 寄存器地址映射

表格 127 SDF_REGS 偏移地址

寄存器名称	寄存器描述	英文全称	偏移地址	WRPRT
SDIFLG	中断标志寄存器	Interrupt Flag Register	0x00	-
SDIFLGCLR	中断标志清除寄存器	Interrupt Flag Clear Register	0x04	-
SDCTL	控制寄存器	Control Register	0x08	√
SDMFILEN	主滤波器使能寄存器	Master Filter Enable Register	0x0C	√
SDSTATUS	标志寄存器	Flag Register	0x0E	-
SDCTLPARMx（x=1-4）	控制参数寄存器	Control Parameter Register	0x20+（x-1）*0x20	√
SDDFPARMx（x=1-4）	数据滤波器参数寄存器	Data Filter Parameter Register	0x22+（x-1）*0x20	√
SDDPARMx（x=1-4）	数据参数寄存器	Data Parameter Register	0x24+（x-1）*0x20	√
SDCMPHx（x=1-4）	高电平阈值寄存器	High-level Threshold Register	0x26+（x-1）*0x20	√
SDCMPLx（x=1-4）	低电平阈值寄存器	Low-level Threshold Register	0x28 +（x-1）*0x20	√
SDCPARMx（x=1-4）	比较器滤波器参数寄存器	Comparator Filter Parameter Register	0x2A+（x-1）*0x20	√
SDDATAx（x=1-4）	数据滤波器数据寄存器	Data Filter Data Register	0x2C+（x-1）*0x20	-
SDDATFIFOx（x=1-4）	数据 FIFO 输出寄存器	Data FIFO Output Register	0x30+（x-1）*0x20	-
SDCDATAx（x=1-4）	比较器滤波器数据寄存器	Comparator Filter Data Register	0x34+（x-1）*0x20	-
SDCMPHZx（x=1-4）	高电平（Z）阈值寄存器	High-level (Z) Threshold Register	0x38+（x-1）*0x20	√

寄存器名称	寄存器描述	英文全称	偏移地址	WRPRT
SDFIFOCTLx（x=1-4）	FIFO 控制寄存器	FIFO Control Register	0x3A+（x-1）*0x20	√
SDF_SYNCTRLx（x=1-4）	SYNC 控制寄存器	SYNC Control Register	0x3C+（x-1）*0x20	√

30.8 寄存器功能描述

30.8.1 中断标志寄存器（SDIFLG）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	IFH1	R	通道 1 的高级比较器滤波（Ch1 High Comparator filter） 0：比较器滤波器输出 ＜ SDCMPHx[HLT]（x=1） 1：比较器滤波器输出 ≥ SDCMPHx[HLT]（x=1）	0h
1	IFL1	R	通道 1 的低级比较器滤波（Ch1 Low Comparator filter） 0：比较器滤波器输出 ＞ SDCMPLx [LLT]（x=1） 1：比较器滤波器输出 ≤ SDCMPLx [LLT]（x=1）	0h
2	IFH2	R	通道 2 的高级比较器滤波（Ch2 High Comparator filter） 0：比较器滤波器输出 ＜ SDCMPHx[HLT]（x=2） 1：比较器滤波器输出 ≥ SDCMPHx[HLT]（x=2）	0h
3	IFL2	R	通道 2 的低级比较器滤波（Ch2 Low Comparator filter） 0：比较器滤波器输出 ＞ SDCMPLx [LLT]（x=2） 1：比较器滤波器输出 ≤ SDCMPLx [LLT]（x=2）	0h
4	IFH3	R	通道 3 的高级比较器滤波（Ch3 High Comparator filter） 0：比较器滤波器输出 ＜ SDCMPHx[HLT]（x=3） 1：比较器滤波器输出 ≥ SDCMPHx[HLT]（x=3）	0h
5	IFL3	R	通道 3 的低级比较器滤波（Ch3 Low Comparator filter） 0：比较器滤波器输出 ＞ SDCMPLx [LLT]（x=3） 1：比较器滤波器输出 ≤ SDCMPLx [LLT]（x=3）	0h
6	IFH4	R	通道 4 的高级比较器滤波（Ch4 High Comparator filter） 0：比较器滤波器输出 ＜ SDCMPHx[HLT]（x=4） 1：比较器滤波器输出 ≥ SDCMPHx[HLT]（x=4）	0h
7	IFL4	R	通道 4 的低级比较器滤波（Ch4 Low Comparator filter） 0：比较器滤波器输出 ＞ SDCMPLx [LLT]（x=4） 1：比较器滤波器输出 ≤ SDCMPLx [LLT]（x=4）	0h
8	MF1	R	滤波器 1 调制器故障标志（Filter 1 Modulator Failure Flag） 0：运转正常 1：故障	0h
9	MF2	R	滤波器 2 调制器故障标志（Filter 2 Modulator Failure Flag） 0：运转正常 1：故障	0h

位/域	名称	R/W	描述	复位值
10	MF3	R	滤波器 3 调制器故障标志（Filter 3 Modulator Failure Flag） 0：运转正常 1：故障	0h
11	MF4	R	滤波器 4 调制器故障标志（Filter 4 Modulator Failure Flag） 0：运转正常 1：故障	0h
12	AF1	R	过滤器 1 的数据标志（Filter 1 Data flag ） 0：在非 FIFO 模式下，没有新数据可用于该过滤器 1：在非 FIFO 模式下，新数据可用于该过滤器	0h
13	AF2	R	过滤器 2 的数据标志（Filter 2 Data flag ） 0：在非 FIFO 模式下，没有新数据可用于该过滤器 1：在非 FIFO 模式下，新数据可用于该过滤器	0h
14	AF3	R	过滤器 3 的数据标志（Filter 3 Data flag ） 0：在非 FIFO 模式下，没有新数据可用于该过滤器 1：在非 FIFO 模式下，新数据可用于该过滤器	0h
15	AF4	R	过滤器 4 的数据标志（Filter 4 Data flag ） 0：在非 FIFO 模式下，没有新数据可用于该过滤器 1：在非 FIFO 模式下，新数据可用于该过滤器	0h
16	SDFFOVF1	R	通道 1 的溢出标志（Ch1 FIFO Overflow Flag） 0：FIFO 未溢出 1：FIFO 溢出。 FIFO 缓冲区中接收到的数据字数超过了 FIFO 的深度 16，新接收的数据字丢失。	0h
17	SDFFOVF2	R	通道 2 的溢出标志（Ch2 FIFO Overflow Flag）	0h
18	SDFFOVF3	R	通道 3 的溢出标志（Ch3 FIFO Overflow Flag）	0h
19	SDFFOVF4	R	通道 4 的溢出标志（Ch4 FIFO Overflow Flag）	0h
20	SDFFINT1	R	通道 1 数据就绪中断标志（Ch1 SDFIFO data ready interrupt Flag） 0：没有发生数据就绪中断标志 1：发生数据就绪中断标志	0h
21	SDFFINT2	R	通道 2 数据就绪中断标志（Ch2 SDFIFO data ready interrupt Flag）	0h
22	SDFFINT3	R	通道 3 数据就绪中断标志（Ch3 SDFIFO data ready interrupt Flag）	0h
23	SDFFINT4	R	通道 4 数据就绪中断标志（Ch4 SDFIFO data ready interrupt Flag）	0h
30:24	保留			0h
31	MIF	R	错误中断活跃标志（error interrupt active Flag） 位域 MF1- MF4，IFH1- IFH4，IFL1- IFL4，SDFFOVF1-SDFFOVF4 是活跃的	0h

30.8.2 中断标志清除寄存器（SDIFLGCLR）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	IFH1	R-0/W1S	清除通道 1 的高级比较器滤波（Ch1 High Comparator filter Clear ）	0h
1	IFL1	R-0/W1S	清除通道 1 的低级比较器滤波（Ch1 Low Comparator filter Clear ）	0h
2	IFH2	R-0/W1S	清除通道 2 的高级比较器滤波（Ch2 High Comparator filter Clear ）	0h
3	IFL2	R-0/W1S	清除通道 2 的低级比较器滤波（Ch2 Low Comparator filter Clear ）	0h
4	IFH3	R-0/W1S	清除通道 3 的高级比较器滤波（Ch3 High Comparator filter Clear ）	0h
5	IFL3	R-0/W1S	清除通道 3 的低级比较器滤波（Ch3 Low Comparator filter Clear ）	0h
6	IFH4	R-0/W1S	清除通道 4 的高级比较器滤波（Ch4 High Comparator filter Clear ）	0h
7	IFL4	R-0/W1S	清除通道 4 的低级比较器滤波（Ch4 Low Comparator filter Clear ）	0h
8	MF1	R-0/W1S	清除滤波器 1 调制器故障标志（Filter 1 Modulator Failure Flag Clear）	0h
9	MF2	R-0/W1S	清除滤波器 2 调制器故障标志（Filter 2 Modulator Failure Flag Clear）	0h
10	MF3	R-0/W1S	清除滤波器 3 调制器故障标志（Filter 3 Modulator Failure Flag Clear）	0h
11	MF4	R-0/W1S	清除滤波器 4 调制器故障标志（Filter 4 Modulator Failure Flag Clear）	0h
12	AF1	R-0/W1S	清除过滤器 1 的数据标志（Filter 1 Data flag Clear）	0h
13	AF2	R-0/W1S	清除过滤器 2 的数据标志（Filter 2 Data flag Clear）	0h
14	AF3	R-0/W1S	清除过滤器 3 的数据标志（Filter 3 Data flag Clear）	0h
15	AF4	R-0/W1S	清除过滤器 4 的数据标志（Filter 4 Data flag Clear）	0h
16	SDFFOVF1	R-0/W1S	清除通道 1 的溢出标志（Ch1 FIFO Overflow Flag Clear）	0h
17	SDFFOVF2	R-0/W1S	清除通道 2 的溢出标志（Ch2 FIFO Overflow Flag Clear）	0h
18	SDFFOVF3	R-0/W1S	清除通道 3 的溢出标志（Ch3 FIFO Overflow Flag Clear）	0h
19	SDFFOVF4	R-0/W1S	清除通道 4 的溢出标志（Ch4 FIFO Overflow Flag Clear）	0h
20	SDFFINT1	R-0/W1S	清除通道 1 数据就绪中断标志（Ch1 SDFIFO data ready interrupt Flag Clear）	0h
21	SDFFINT2	R-0/W1S	清除通道 2 数据就绪中断标志（Ch2 SDFIFO data ready interrupt Flag Clear）	0h
22	SDFFINT3	R-0/W1S	清除通道 3 数据就绪中断标志（Ch3 SDFIFO data ready interrupt Flag Clear）	0h
23	SDFFINT4	R-0/W1S	清除通道 4 数据就绪中断标志（Ch4 SDFIFO data ready interrupt Flag Clear）	0h
30:24	保留			0h

位/域	名称	R/W	描述	复位值
31	MIF	R-0/W1S	清除错误中断活跃标志（error interrupt active Flag Clear） 向 SDIFLG 寄存器写入 1 以清除 MIF 位，写 0 忽略。 注意：如果清除了 MIF 位后，其他中断仍在等待处理，则在下一个系统时钟周期，该位将再次被设置为 1 且中断输出将被重新断言，产生一个低电平脉冲。	0h

30.8.3 控制寄存器（SDCTL）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	HZ1	R-0/W1S	清除错误中断活跃标志（error interrupt active Flag Clear）	0h
1	HZ2	R-0/W1S	清除 CH1 的高级别阈值跨越标志（CH1 High-level Threshold crossing (Z) flag Clear）	0h
2	HZ3	R-0/W1S	清除 CH2 的高级别阈值跨越标志（CH2 High-level Threshold crossing (Z) flag Clear）	0h
3	HZ4	R-0/W1S	清除 CH3 的高级别阈值跨越标志（CH3 High-level Threshold crossing (Z) flag Clear）	0h
12:4	保留			0h
13	MIE	R/W	使能主机 SDFy_ERR 中断（Master SDFy_ERR interrupt enable） 0：禁止 1：使能	0h
15:14	保留			0h

30.8.4 主滤波器使能寄存器（SDMFILEN）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
10:0	保留			0h
11	MFE	R/W	使能主机滤波器（Master Filter Enable） 0：SDF 的四个数据滤波器被禁止（清除所有 FIFO） 1：如果 SDDFPARMx [FEN]位为 1，使能数据滤波器	0h
15:12	保留			0h

30.8.5 标志寄存器（SDSTATUS）

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
x-1（x=1-4）	HZx	R	CHx 的高级别阈值跨越标志（CHx High-level Threshold crossing (Z) flag ） 这些标志位主要用于检测当输入信号从正值穿越到负值或从负值穿越到正值时的瞬间，但该位不能生成中断（与 SDIFLG[IFH1]位不一样）。 0：比较器滤波器输出 ＜ SDCMPHZx（x=1）[HLTZ] 1：比较器滤波器输出 ≥ SDCMPHZx（x=1）[HLTZ]	0h
15:4	保留			0h

30.8.6 控制参数寄存器（SDCTLPARMx（x=1-4））

偏移地址：0x20+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	MOD	R/W	选择调制器时钟模式（Modulator clock modes select） 模式 0：调制器时钟以 1 倍数据速率运行 模式 1：调制器时钟以 1/2 数据速率运行（双边缘时钟） 模式 2：调制器时钟缺席（曼彻斯特编码数据） 模式 3：调制器时钟以 2 倍数据速率运行	0h
15:2	保留			0h

30.8.7 数据滤波器参数寄存器（SDDFPARMx（x=1-4））

偏移地址：0x22+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	DOSR	R/W	设置数据过滤器过采样比（Data filter Oversampling ratio Set） 数据滤波器的实际过采样比率为 DOSR + 1。例如当设置值为 0x0FF 时，实际的过采样比率是 256。	0h
8	FEN	R/W	使能过滤器（Filter Enable） 0：禁止数据滤波器（不会生成任何数据），此时 DOSR 计数器将停留在重置状态，滤波器内的数据会被删除，并且重置 FIFO 指针并清除 FIFO 内容。 1：使能数据滤波器（数据将在滤波器内生成）	0h
9	AE	R/W	使能数据过滤器确认（Data filter Acknowledge Enable） 0：禁止 1：使能	0h
11:10	SST	R/W	选择数据过滤器结构（Data filter structure select） 00：数据滤波器使用 Sincfast 结构运行 01：数据滤波器使用 Sinc1 结构运行 10：数据滤波器使用 Sinc2 结构运行 11：数据滤波器使用 Sinc3 结构运行	0h

位/域	名称	R/W	描述	复位值
12	SDSYNCEN	R/W	使能数据过滤器的 PWM 同步（data filter PWM synchronization (SDSYNC) enable） 0：禁止 1：使能 注意：SDSYNCx [SYNCSEL] 定义了用于同步 PWM 的是哪一个 PWM 信号	0h
15:13	保留			0h

30.8.8 数据参数寄存器（SDDPARMx（x=1-4））

偏移地址：0x24+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	保留			0h
10	DR	R/W	配置数据过滤器中的数据（Data in the data filter configurae） 0：数据以 16 位二进制补码形式存储 1：数据以 32 位二进制补码形式存储	0h
15:11	SH	R/W	移动控制（Shift Control） 该位域当选择 16 位数据表示时，16 位窗口向上移动了多少位。	0h

30.8.9 高电平阈值寄存器（SDCMPHx（x=1-4））

偏移地址：0x26+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
14:0	HLT	R/W	比较器滤波器输出的无符号高电平阈值（unsigned high level threshold of the comparator filter output）	7FFFh
15	保留			0h

30.8.10 低电平阈值寄存器（SDCMPLx（x=1-4））

偏移地址：0x28 +（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
14:0	LLT	R/W	比较器滤波器输出的无符号低电平阈值（unsigned low level threshold of the comparator filter output）	0h
15	保留			0h

30.8.11 比较器滤波器参数寄存器（SDCPARMx（x=1-4））

偏移地址：0x2A+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	COSR	R/W	设置比较器过采样比（Comparator Oversampling ratio Set） 数据滤波器的实际过采样比率为 COSR + 1。例如当设置值为 0x1F 时，实际的过采样比率是 32。	0h

位/域	名称	R/W	描述	复位值
5	IEH	R/W	使能高级中断（High-level interrupt enable） 0：禁止 1：使能	0h
6	IEL	R/W	使能低级中断（Low-level interrupt enable） 0：禁止 1：使能	0h
8:7	CS1_CS0	R/W	选择比较器过滤器结构（Comparator filter structure select） 00：比较器过滤器使用 Sincfast 结构运行 01：比较器过滤器使用 Sinc1 结构运行 10：比较器过滤器使用 Sinc2 结构运行 11：比较器过滤器使用 Sinc3 结构运行	0h
9	MFIE	R/W	使能调制器故障中断（Modulator Failure Interrupt Enable） 0：禁止调制器故障中断及其标志 1：使能调制器故障中断及其标志	0h
10	HZEN	R/W	使能高电平阈值交叉输出（High level (Z) Threshold crossing output enable） 0：禁止 1：使能	0h
12:11	保留			0h
13	CEN	R/W	使能比较器过滤器（Comparator Filter enable） 0：禁止 1：使能	0h
15:14	保留			0h

30.8.12 数据滤波器数据寄存器（SDDATAx（x=1-4））

偏移地址：0x2C+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DATA16	R	低阶 16b 在 32b 模式（Lo-order 16b in 32b mode）	0h
31:16	DATA32HI	R	高阶 16b 在 32b 模式（Hi-order 16b in 32b mode）	0h

30.8.13 数据 FIFO 输出寄存器（SDDATFIFOx（x=1-4））

偏移地址：0x30+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DATA16	R	低阶 16b 在 32b 模式（Lo-order 16b in 32b mode）	0h
31:16	DATA32HI	R	高阶 16b 在 32b 模式（Hi-order 16b in 32b mode）	0h

30.8.14 比较器滤波器数据寄存器（SDCDATAx（x=1-4））

偏移地址：0x34+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DATA16	R	比较器数据输出（Comparator Data output）	0h

30.8.15 高电平（Z）阈值寄存器（SDCMPHZx（x=1-4））

偏移地址：0x38+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
14:0	HLTZ	R/W	比较器滤波器输出的无符号高电平阈值（unsigned high level threshold (Z) of the comparator filter output） 这些标志位主要用于检测当输入信号从正值穿越到负值或从负值穿越到正值时的瞬间，但该位不能生成中断（与 SDCMPHx 寄存器不一样）。	0h
15	保留			0h

30.8.16 FIFO 控制寄存器（SDFIFOCTLx（x=1-4））

偏移地址：0x3A+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	SDFFIL	R/W	SDFIFO 中断级（SDFIFO interrupt level） 当 SDFFST 位≥SDFFIL 位时，FIFO 将产生一个中断。	0h
5	保留			0h
10:6	SDFFST	R	SDFIFO 状态（SDFIFO Status） 00000：FIFO 为空 00001：FIFO 含有 1 字 … 10000：FIFO 含有 16 字	0h
11	保留			0h
12	FFIEN	R/W	使能 SDFIFO 数据就绪中断（SDFIFO data ready Interrupt Enable）	0h
13	FFEN	R/W	使能 SDFIFO（SDFIFO Enable） 0：禁止，此时 FIFO 内容被清除 1：使能	0h
14	DRINTSEL	R/W	选择数据就绪中断源（Data Ready Interrupt source select） 0：选择非 FIFO 数据就绪中断，通过 SDIFLG[AF1]位通知 1：选择 FIFO 数据就绪中断，通过 SDIFLGCLR[SDFFINT1]位通知	0h
15	OVFIEN	R/W	使能 SDFIFO 溢出中断（SDFIFO Overflow interrupt enable） 0：SDFIFO 溢出条件不会生成中断 1：SDFIFO 溢出条件会在 SDy_ERR 上生成中断	0h

30.8.17 SYNC 控制寄存器（SDF_SYNCTRLx（x=1-4））

偏移地址：0x3C+（x-1）*0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
5:0	SYNCSEL	R/W	SDSYNC 定义源（SDSYNC Defines source）	0h

位/域	名称	R/W	描述	复位值
6	WTSYNCEN	R/W	使能等待同步（Wait-for-Sync Enable） 0：每当有数据就绪事件发生，传入数据被写入 SDFIFO 1：只有在 SDSYNC 事件发生后，当再次发生数据就绪事件时，传入数据才被写入 SDFIFO	0h
7	WTSYNFLG	R	等待同步标志（Wait-for-Sync Flag） 0：未发生 SDSYNC 事件 1：发生 SDSYNC 事件	0h
8	WTSYNCLR	R-0/W	清除等待同步标志（Wait-for-Sync Flag Clear ） 0：无影响 1：清除 WTSYNFLG 位	0h
9	FFSYNCCLREN	R/W	使能 SDSYNC 自动清除 SDFIFO （SDSYNC automatically clear SDFIFO Enable） 0： SDFIFO 接收到 SDSYNC 后不自动清除 1:：SDFIFO 接收到 SDSYNC 后自动清除	0h
10	WTSCLREN	R/W	使能 FIFOINT 自动清除 WTSYNFLG（FIFOINT automatically clear WTSYNFLG Enable） 0：只能通过 WTSYNCLR 位手动清除 WTSYNFLG 1：在 RDYINTFLG 位上自动清除 WTSYNFLG	0h
15:11	保留			0h

31 脉冲宽度调制器（PWM）

31.1 术语全称、缩写描述

表格 128 术语全称、缩写描述

中文全称	英文全称	英文缩写
上升沿延迟	Rising Edge Delay	RED
下降沿延迟	Falling Edge Delay	FED
时基	Time-Base	TB
时基计数器	Time Base Counter	TBCTR
时基时钟	Time-Base Clock	TBCLK
计数器比较	Counter-Compare	CMP
动作限定器	Action-Qualifier	AQ
死区发生器	Dead-Band Generator	DB
PWM 斩波	PWM Chopper	PC
跳闸区	Trip-Zone	TZ
单次触发	One-Shot	OST
逐周期触发	Cycle-by-Cycle	CBC
事件触发	Event-Trigger	ETRG
数字比较	Digital Compare	DC
高分辨率定位器	High Resolution Positioner	HRP

31.2 简介

PWM 是一种用于控制许多电力电子系统的关键要素，经常在消费级和工业级设备中使用。这些电力电子系统包括 UPS、开关电源控制、数字电机控制以及其他形式的能量转换。同时 PWM 也具备 DAC 的功能，此时占空比等价于 DAC 的模拟值。

每个 PWM 模块包含八个子模块：时基子模块，计数器比较子模块，动作限定器子模块，死区发生器子模块，PWM 斩波子模块，跳闸区子模块，事件触发子模块和数字比较子模块。每个子模块执行特定的功能，且可由软件配置。

31.3 主要特征

（1） 具有两个 PWM 输出，分别为 PWMxA 和 PWMxB，可以将其配置为单边模式、双边对称模式和双边非对称模式

（2） 通过软件对 PWM 信号进行异步覆写控制

（3） 一个专用 16 位时基计数器，可以控制 PWM 的周期和频率

（4） 通过编程调整 PWM 信号的相位，使其与其他 PWM 模块的信号产生滞后或超前的效果

（5） 上升沿和下降沿延迟的死区生成分别独立控制

（6） 采用高频载波信号进行 PWM 调制，适合用于脉冲变压器的栅极驱动

（7） 可以逐周期实现硬件锁定相位关系

（8） 在故障情况下，跳闸区既可分配给逐周期跳闸，也可分配给单次跳闸

（9） 跳闸状态可以强制 PWM 输出高，低逻辑电平或高阻抗状态

（10） 编程事件分频可以减小 CPU 响应中断的开销

（11） 所有事件都是 ADCSOC 和 CPU 中断的触发源

31.3.1 时基子模块功能特征

（1） 时基计数器模式：
- 向上计数模式
- 向下计数模式
- 中心对齐计数模式

（2） 时基时钟：通过对 PWMCLK 预分频得到

（3） 通过配置 TBCTR 频率或周期，可以控制事件发生的频率

（4） 产生 TBCTR=0 或 TBCTR=TBPRD 事件

（5） 可以配置相对于其它 PWM 模块的时基相位

（6） 通过软件或硬件实现 PWM 模块之间的 TBCTR 同步

（7） 在同步事件后，配置 TBCTR 方向为向上或向下

（8） 可以将相同的值同时写入多个 PWM 模块 TBPRD 寄存器

（9） 在仿真器暂停器件时，可以配置 TBCTR 的行为

（10） PWM 模块同步输出信号源：
- 同步输入信号
- TBCTR=0
- TBCTR=CMPB
- 不产生同步输出信号

（11） 配置 PWM 模块中寄存器的单次和全局装载

31.3.2 计数器比较子模块功能特征

（1） 可以指定 PWMxA 和 PWMxB 输出切换电平的时刻

（2） 可以指定 PWMxA 和 PWMxB 输出的占空比，通过配置 CMPA 和 CMPB 寄存器来控制占空比

（3） 可以将相同的值同时写入多个 PWM 模块 CMPx 寄存器

（4） 配置 PWM 模块中寄存器的单次和全局装载

（5） 通过额外的比较器(CMPC/D)，指定可编程延迟的中断和 SOC 生成

（6） 影子缓冲新的比较值，可以避免在 PWM 有效周期内产生毛刺或波形失真

（7） 使用 CMPx 寄存器，基于可编程时间戳生成事件：
- TBCTR=CMPA
- TBCTR=CMPB
- TBCTR=CMPC
- TBCTR=CMPD

31.3.3 动作限定器子模块功能特征

（1） 可以通过软件强制控制 PWM 输出状态

（2） 可以通过软件配置和控制 PWM 死区

（3） 当发生时基计数器比较，跳闸区子模块或比较器事件时，可以执行以下动作：
- PWMxA 和 PWMxB 切换为高电平
- PWMxA 和 PWMxB 切换为低电平
- PWMxA 和 PWMxB 电平翻转
- 无任何动作

（4） 在以下事件中，进行定性和生成动作（置位、清除、翻转）：
- TBCTR=0
- TBCTR=TBPRD
- TBCTR=CMPA
- TBCTR=CMPB

（5） T1、T2 事件：基于比较器、跳闸或同步事件的触发事件，当以上事件同时发生时管理优先级

（6） 提供在 TBCTR 递增和递减时独立控制事件的能力

（7） 配置 PWM 模块中寄存器的单次和全局装载

31.3.4 死区发生器子模块功能特征

（1） 从单个 PWMxA 输入生成相应的具有死区关系的信号对，即 PWMxA 和 PWMxB

（2） 可以使能半周期时钟模式，从而加倍分辨率

（3） 允许 PWMxB 输出相对于 PWMxA 相移

（4） 将信号对编程为：
- 高电平有效（AH）
- 低电平有效（AL）
- 高电平有效互补（AHC）

- 低电平有效互补（ALC）

（5） 插入可编程的 RED 和 FED 的值

（6） 可以控制上下开关之间，传统的互补死区关系

（7） 可以完全绕过死区发生器子模块：
- PWM 波形可以没有任何修改地通过死区发生器子模块

（8） 配置 PWM 模块中寄存器的单次和全局装载

31.3.5 PWM 斩波子模块功能特征

（1） 可编程的斩波频率

（2） 首个脉冲可编程脉宽，第二个和后续的脉冲可编程占空比

（3） 可以完全绕过 PWM 斩波子模块：
- PWM 波形可以没有任何修改地通过 PWM 斩波子模块

31.3.6 跳闸区子模块功能特征

（1） 跳闸输入 TZ1n 至 TZ6n 可被灵活地映射到任意 PWM 模块

（2） 支持软件强制跳闸

（3） 配置 PWM 模块对跳闸信号或数字比较事件做出反应：
- 一个跳闸信号或数字比较事件
- 所有跳闸信号或数字比较事件
- 不做出反应

（4） 在故障状态下，可以配置跳闸动作为：
- PWMxA 和/或 PWMxB 强制输出为高阻态
- PWMxA 和/或 PWMxB 强制输出为高电平
- PWMxA 和/或 PWMxB 强制输出为低电平
- 忽略任何跳闸动作

（5） DC 子模块 DCAEVT1/2 或 DCBEVT1/2 强制事件和每个跳闸区输入都可以被分配给单次或逐周期操作

（6） 单次跳闸（OST Trip）：支持用于短路和过流保护。

（7） 逐周期跳闸（CBC Trip）：支持用于限流操作。

（8） 使能跳闸区发起中断，任何一个跳闸区输入都可以产生中断

（9） 支持基于片上模拟比较器和/或 TZ1n 至 TZ3n 的数字比较跳闸（DC）

（10） 可以完全绕过跳闸区子模块

31.3.7 事件触发子模块功能特征

（1） 使能触发中断或 ADCSOC 的 PWM 事件

（2） 允许软件强制触发中断和 ADCSOC

（3） 使用预分频电路指定事件触发的频率（每 2 次至最多每 15 次）

（4） 事件计数器和标志位提供了事件生成的完整可见性

（5） 接收由 TB、CC 和 DC 子模块的事件输入。

（6） 使用时基方向信息来区分（上/下）事件

（7） 轮询、设置、或清除事件标志位

31.3.8 数字比较子模块功能特征

（1） 利用 COMP 模块输出和跳闸区信号来产生事件，这些信号通过输入 X-BAR 传入

（2） 通过以下信号产生 DCxH 和 DCxL 信号：
- 内部 NVIC
- TZ1n、TZ2n、TZ3n 输入
- 使用 Input X-BAR，PWM X-BAR 或者经过外部使用 GPIO 外设输入的 COMP 模块输出

（3） 可以指定事件滤波选项来捕获 TBCTR 并产生空白窗口或基于捕获值在 PWM 输出或 TBCTR 插入延迟

（4） 事件滤波（空白窗口逻辑电路）可用来过滤输入信号，以移除噪声

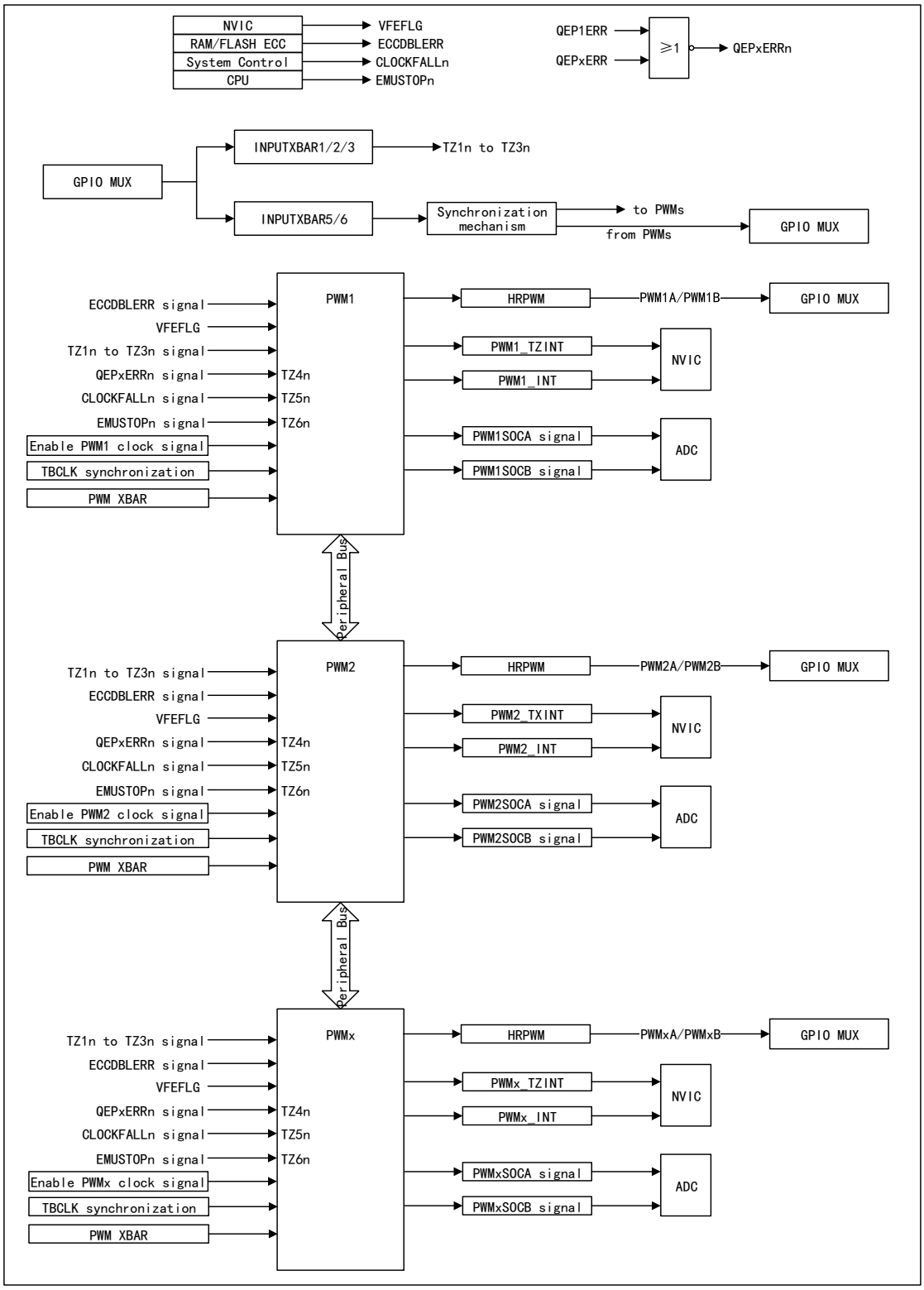
（5） DCxH 和 DCxL 信号可以触发事件，这些事件可以选择进行过滤或直接传递给 TZ、ETRG 和 TB 子模块，从而：
- 生成用于同步 PWM 模块 TBCTR 的同步事件
- 强制产生一个事件
- 生成 ADCSOC
- 生成跳闸区中断

31.4 结构框图

31.4.1 多个 PWM 模块结构框图

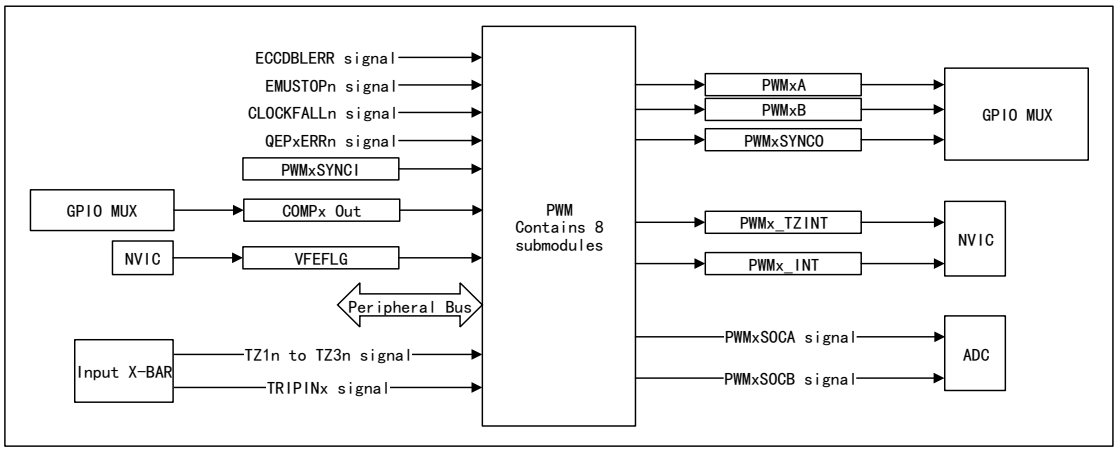
如下图所示，PWM 模块连接顺序不同，详情请参见时基计数器同步小节。每个 PWM 模块连接的信号，详情请参见 PWM 主要信号章节。

图 71 多个 PWM 模块结构框图



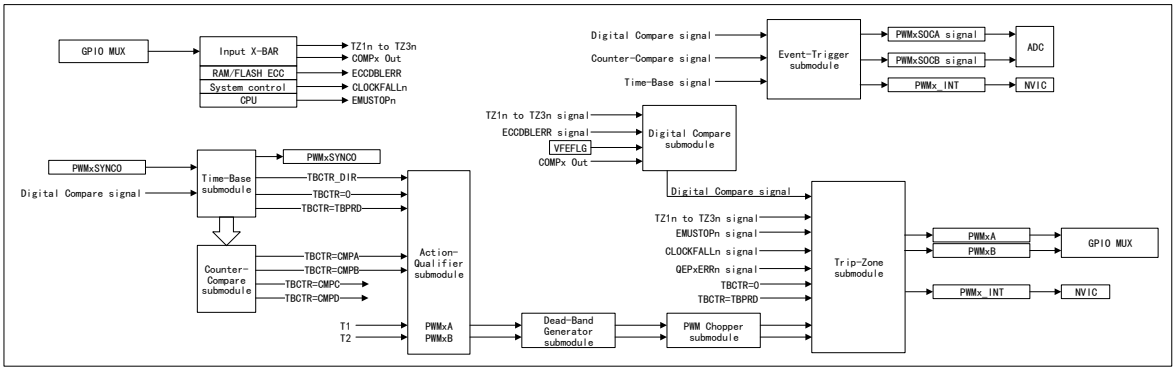
31.4.2 单个 PWM 模块结构框图

图 72 单个 PWM 模块结构框图



31.4.3 PWM 子模块结构框图

图 73 PWM 子模块结构框图



31.5 功能描述

PWM 器件内有多个 PWM 模块的实例，但除了一个例外之外，每个 PWM 实例都是相同的。PWM 模块通过时钟同步进行连接，其子模块可以独立运行。

31.5.1 PWM 增强功能

（1） 数字比较子模块：该模块提供过滤、屏蔽和改进的跳闸功能，增强 ETRG 和 TZ 子模块功能。

（2） 死区分辨率：增强死区时钟功能，可以提高一倍半周期时钟的分辨率。

（3） 高分辨率周期：使能高分辨率周期功能。

（4） 中断和 ADCSOC 生成：可以在 TBCTR=0 或 TBCTR=TBPRD 事件上生成中断和 ADCSOC。实现双边沿 PWM 控制，且可以通过数字比较子模块定义的事件生成 ADCSOC。

（5） PWMxB 输出：可以在该输出上使用高分辨率扩展，提供使能高分辨率周期和占空比控制的功能。

（6） 高分辨率死区能力：死区上升沿延迟和下降沿延迟在半周期时钟模式下具备高分辨率能力。

（7） 计数器比较子模块：通过其他计数器比较 CMPC 和 CMPD 的生成中断和 SOC 事件。

（8） 死区发生子模块：具有 14 位死区计数器，死区寄存器和死区高分辨率寄存器有影子寄存器的功能。在 PWM 输出上可以使能 RED 和 FED 功能。

（9） 事件触发子模块：预分频逻辑可扩展到每 15 个事件发出中断请求和 ADCSOC，且可以在同步事件上对事件计数器进行软件初始化。

（10） 数字比较子模块：数字比较跳闸选择逻辑 [DCTRIPSEL] 具有由输入 X-BAR 逻辑选择的最多 12 个外部跳闸源，此外还能够对所有这些源（最多 14 个 [外部和内部源]）进行或运算以创建相应的 DCxEVT。

（11） 同时写入 CMPx 和 TBPRD 寄存器：
- 任一 PWM 模块的 CMPx，CMPxHR 和 TBPRD 的写入都可以与其他任一 PWM 模块进行绑定，还可以将所有 PWM 模块绑定到特定的 PWM 模块。
- 在同步事件时，CMPx 和 TBPRD 活动寄存器装载到影子寄存器

（12） 寄存器地址映射：需要额外的寄存器。为了更好地对齐和方便使用，PWM 寄存器地址空间被重新映射。

（13） 延迟跳闸功能：增加了实现死区插入功能的更改以支持此功能，例如峰值电流模式控制类型应用场景所需的延迟跳闸功能。这是通过允许比较器事件作为触发事件进入 Action Qualifier(事件 T1 和 T2)来实现的。如果比较器 T1 / T2 事件用于改变 PWM，对 PWM 波形的改变不会立即发生。相反，波形与下一个 TBCLK 同步。

（14） 死区生成器子模块增强：增加 DBCTL 影子寄存器以允许动态配置更改。

（15） 寄存器的单次和全局装载：从影子寄存器的装载支持单次装载和全局装载，以避免部分装载，例如多相应用。装载时间支持预分频。全局装载可以通过删除中断来简化软件，并且确保所有寄存器被同时装载。

（16） 跳闸区子模块增强：添加了独立的标志来反映每个 TZ 源的跳闸状态。对跳闸区子模块进行了更改，以支持某些功率转换器开关技术，如谷底开关。

（17） 数字比较子模块增强：消隐窗口滤波器寄存器宽度从 8 位增加到 16 位。DCCAP 功能已得到增强，以提供更多的可编程性。

（18） PWM 同步相关增强：基于 CMPC 和 CMPD 事件的 PWM 同步产生。这些事件也可用于 PWMSYNC 脉冲选择。

31.5.2 PWM 引脚功能描述

使用输入 X-BAR 和 PWM XBAR 才能将器件引脚上的信号接入本模块。本器件上的任意 GPIO 均可配置为输入模式。需要通过向相应的 GPACSELy 寄存器写入 11，将 GPIO 输入模式设置为异步模式。可在 GPxPUD 寄存器中配置内部上拉。在使用 GPIO 模式时，GPxINV 寄存器可以对信号进行翻转。此外，TRIP4 至 TRIP12（除了 TRIP6）信号在经过输入 X-BAR 之后，还必须经过 PWM X-

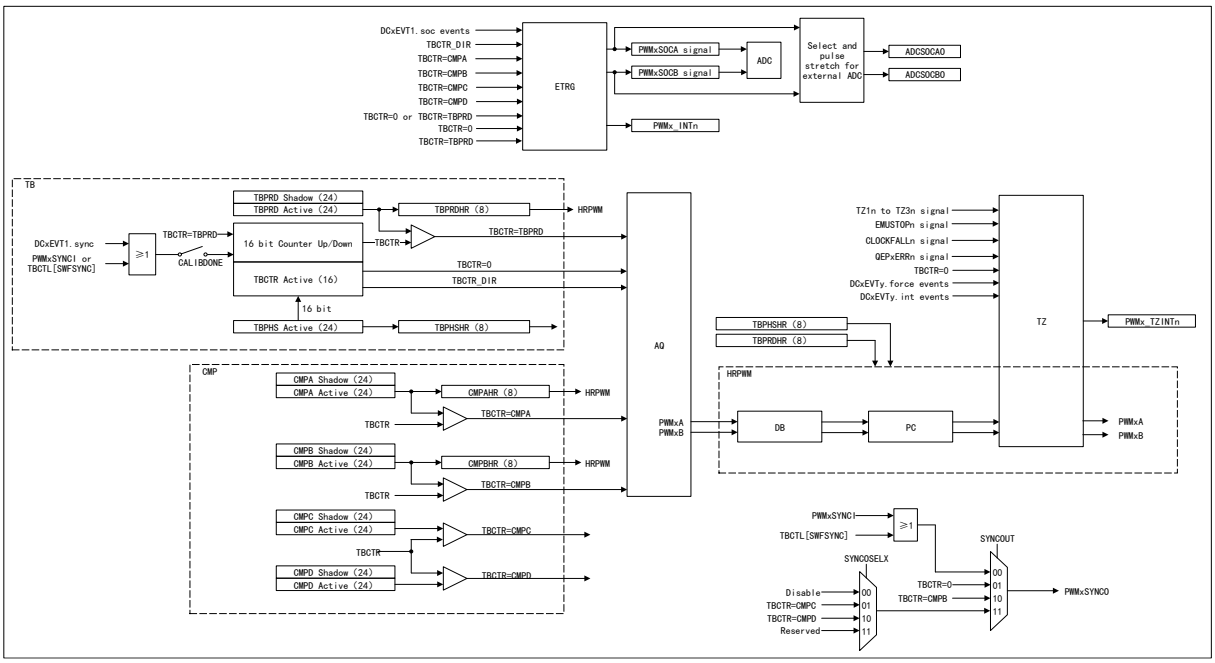
BAR。

需要为 PWM 配置 GPIO 复用寄存器。需要先配置 GPACSELy 位域并保持相应的 GPACSELy 位域为默认 0 值，然后向 GPACSELy 寄存器写入期望值，以避免引脚上信号的抖动。

GPIO 多路复用器详细请参见数据手册，GPIO 设置和 XBAR 配置详细请参见 GPIO 章节。

31.5.2.1 PWM 主要信号

图 74 PWM 模块和关键内部信号互联结构框图



注意：由 PWM DC 子模块根据 TRIPINPUT 输入的电平生成 DCxEVTy 事件。

表格 129 PWM 主要信号

信号类型	信号名称	描述
PWM 输出信号	PWMxA	输出到器件外的 PWM 信号
	PWMxB	
跳闸区信号	TZ1n 至 TZ6n	每一个器件上的子模块可以配置为使用或忽略任何跳闸区信号，且这些输入信号通知 PWM 模块外部的故障情况。这允许配置在时钟故障或 CPU 停止时的跳闸动作。 TZ1n 至 TZ3n：可以通过 GPIO 外设和 X-BAR 配置为异步输入。 TZ4n：连接到反相的 QEPx 错误信号。 TZ5n：连接系统时钟失效逻辑。 TZ6n：连接到 CPU 的 EMUSTOPn 输出。
时基同步信号	PWMxSYNCI	同步信号以菊花链的形式连接各 PWM 模块。每个模块都可以配置为使用或忽略同步输入。出于同步的目的，PWM 模块被分成三组：

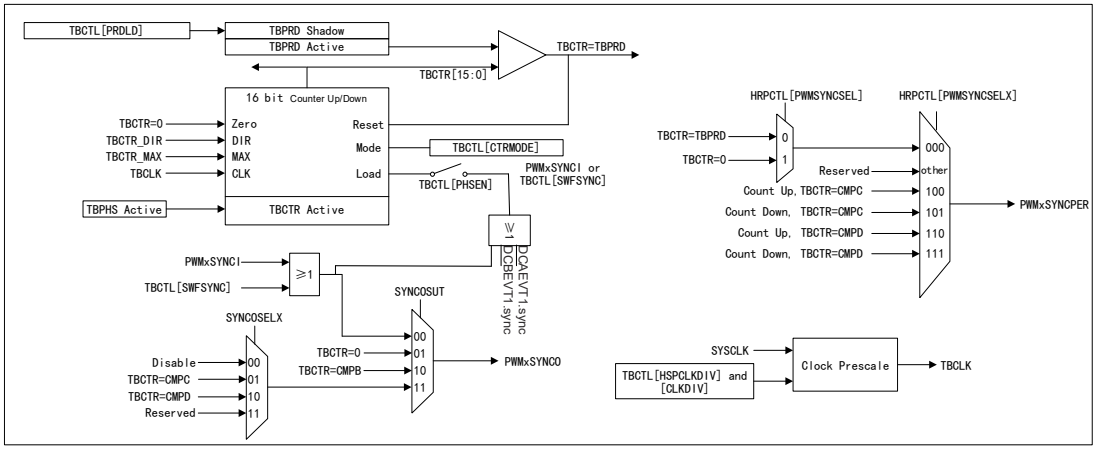
信号类型	信号名称	描述
	PWMxSYNCO	外部的 SYNC 信号：EXTSYNCIN1 或 EXTSYNCIN2 可被用来发出 SYNC 信号到每条链上的第一个 PWM 模块。PWMxSYNCO 信号：这些模块也可以将 PWMxSYNCO 信号发出到 GPIO，详情请参见同步章节。PWMxSYNCSEL 信号：每个 PWM 模块都可以产生该信号，且该信号用于同步 GPDAC 和 COMP。其功能由 HRPCTL 寄存器配置，但是与 HRPWM 没有关系。有关 GPDAC 和 COMP 使用 PWMxSYNCSEL 的内容，详情请参见对应的章节。
	PWMxSYNCSEL	
ADC 转换开始信号	PWMxSOCA	每个 PWM 模块都具有 PWMxSOCA 和 PWMxSOCB 信号。任何 PWM 模块都可以触发转换开始，且由 ETRG 子模块配置哪一个事件触发转换开始。
	PWMxSOCB	
比较器输出信号	COMPxOUT	COMP 模块的输出信号可以通过输入 X-BAR 传递到一个或所有 12 个跳闸输入（TRIPIN1-TRIPIN12），并且与跳闸区信号搭配，可以产生数字比较事件。
外设总线	-	外设总线宽 32 位，同时支持 PWM 寄存器的 16 位和 32 位写入。

31.5.3 时基子模块功能描述

时基子模块用于确定该 PWM 模块的所有事件时序，所以每个 PWM 模块都有自己的时基子模块。内部的同步逻辑允许多个 PWM 模块的时基协同工作，构成一个系统。

31.5.3.1 时基子模块结构框图

图 75 时基子模块结构框图



时基子模块关键信号

表格 130 时基子模块关键信号

信号类型	信号名称	描述
时基时钟	TBCLK	通过对 PWMCLK 预分频得到，可以用于所有 PWM 子模块，且该时钟决定了 TBCTR 递增或递减的速率。
时基计数器方向	TBCTR_DIR	表示 PWM 时基计数器当前的方向。该信号在计数器递增时为高电平，递减时为低电平。
时基计数器等于最大值	TBCTR_MAX	当 TBCTR 达到最大值，即 TBCTR=0xFFFF 时，产生该事件。该信号仅用作状态位。
时基计数器等于给定周期	TBCTR=TBPRD	计数器值等于活动周期寄存器值，即 TBCTR=TBPRD 时，产生该信号。
时基计数器等于 0	TBCTR=0	计数器值等于零，即 TBCTR=0 时，产生该信号。
时基计数器等于活动计数器比较 B 寄存器	TBCTR=CMPB	该事件由 COMP 子模块产生，并用于同步输出逻辑。
时基同步信号	PWMxSYNCI 时基同步输入	该信号为脉冲输入，用于将 TBCTR 与同步链中的前一个 TBCTR 同步。PWM 模块可配置为采用或忽略此信号。对于每个同步链中的第一个 PWM 模块，该信号来自前一个 PWM 模块或来自器件引脚，通过输入 X-BAR 的 INPUT5 或 INPUT6 传输进来。对于每个同步链中后续的 PWM 模块，该信号来自前一个 PWM 模块。例如，PWM1 产生 PWM2SYNCI 信号，PWM2 产生 PWM3SYNCI 信号，以此类推。关于特定器件的同步顺序，详情请参见同步章节。
	PWMxSYNCO 时基同步输出	该输出脉冲用于同步同步链中后一个 PWM 模块的 TBCTR，PWM 模块通过以下事件源产生该信号： （1）TBCTR=0 （2）TBCTR=CMPB （3）PWMxSYNCI
	PWMxSYNCPER 时基外设同步输出	该信号用于将 GPDAC 和 COMP 与 PWM 同步，其功能由 HRPCTL 寄存器配置，但是与 HRPWM 没有关系。

31.5.3.2 PWM 周期和频率计算

通过 TBCTR 寄存器和计数器模式控制 PWM 事件的频率，其计数器模式通过 TBCTL 寄存器来选择。由下图所示，当 TBPRD 设置为 4 时，在三种工作模式下的周期 T_{PWM} 和频率 F_{PWM} 关系。由 TBCLK 决定每一步的时间增量。

（1） 向上计数模式

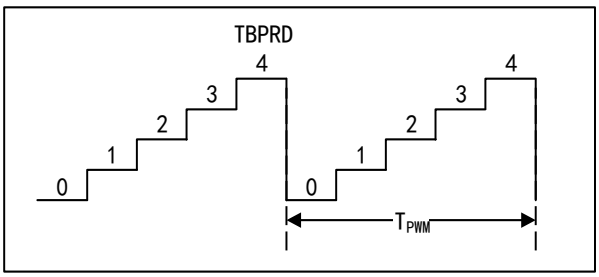
在该模式下，TBCTR 从 0 开始递增，直到达到 TBPRD 寄存器中的值。当达到 TBPRD 值时，时基计数器重置为 0 并再次开始增加。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式为：

$$T_{PWM} = (TBPRD + 1) * T_{TBCLK}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

图 76 向上计数模式的时基频率和周期

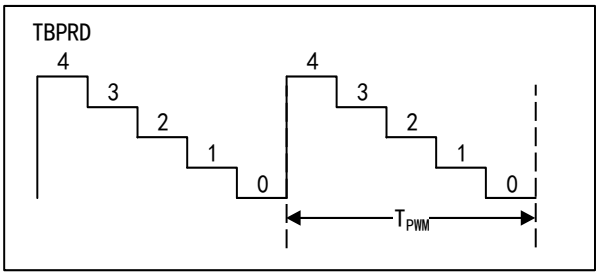


（2） 向下计数模式

在该模式下，TBCTR 从 TBPRD 值开始递减，直到达到 0。当它达到 0 时，TBCTR 重置为 TBPRD 值，并再次开始递减。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式详情请参见向上计数模式。

图 77 向下计数模式的时基频率和周期



（3） 中心对齐计数模式

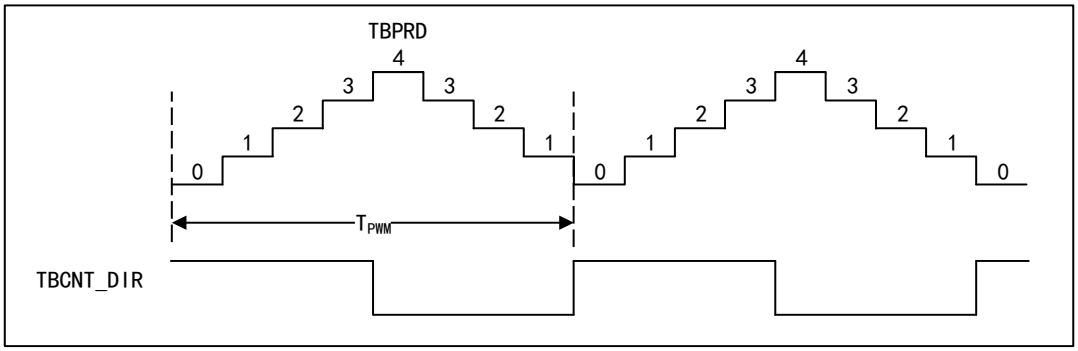
在该模式下，TBCTR 从 0 开始递增，一直到 TBPRD 值。当到达 TBPRD 值后，TBCTR 开始递减直到为 0。此时，计数器重复该模式并开始递增。

在该模式下，周期 T_{PWM} 和频率 F_{PWM} 计算公式为：

$$T_{PWM} = 2 * TBPRD * T_{TBCLK}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

图 78 中心对齐计数模式的时基频率和周期



TBPRD 影子寄存器

所有关于 PWM 模块的影子寄存器描述采用以下定义：

表格 131 时基子模块关键信号

寄存器分类	描述
活动寄存器	活动寄存器控制硬件，并对硬件引起或调用的操作负责。
影子寄存器	影子寄存器为活动寄存器提供缓冲，即提供一个临时保存位置，所以它对任何硬件控制没有直接影响。为了防止由于软件异步修改寄存器而导致的数据损坏或误操作，影子寄存器会在一个特定时间点上，将内容转移到活动寄存器。

TBPRD 寄存器有一个影子寄存器，并且活动寄存器和影子寄存器的内存映射地址相同。影子化允许寄存器的更新与硬件同步。由 PRDLD 位决定向哪个寄存器写入或从哪个寄存器读取，使能和禁止 TBPRD 影子寄存器。

表格 132 时基子模块关键信号

TBPRD 寄存器工作模式	描述
影子模式	PRDLD=0，使能影子模式（默认），此时，对 TBPRD 内存地址的读取和写入操作都进入影子寄存器。当 TBCTR=0 或且/或由 PRDLDSYNC 位确定同步事件时，将影子寄存器的内容转移到活动寄存器中。PRDLDSYNC 位仅当 PRDLD=0 时才有效。同步输入的来源详细请参见同步章节。 通过配置 GLDCFG 寄存器，时基周期寄存器也可以采用全局加载机制。若选择全局加载模式，则所有使能此模式的寄存器，在同一时刻发生影子寄存器到活动寄存器的内容转移。由 GLDCTL 寄存器决定发生内容转移的时刻。全局加载机制详细请参见全局装载章节。
立即模式	PRDLD=1，使能立即装载模式，此时，对 TBPRD 内存地址的读取或写入操作将直接访问活动寄存器。

31.5.3.3 时基计数器工作模式

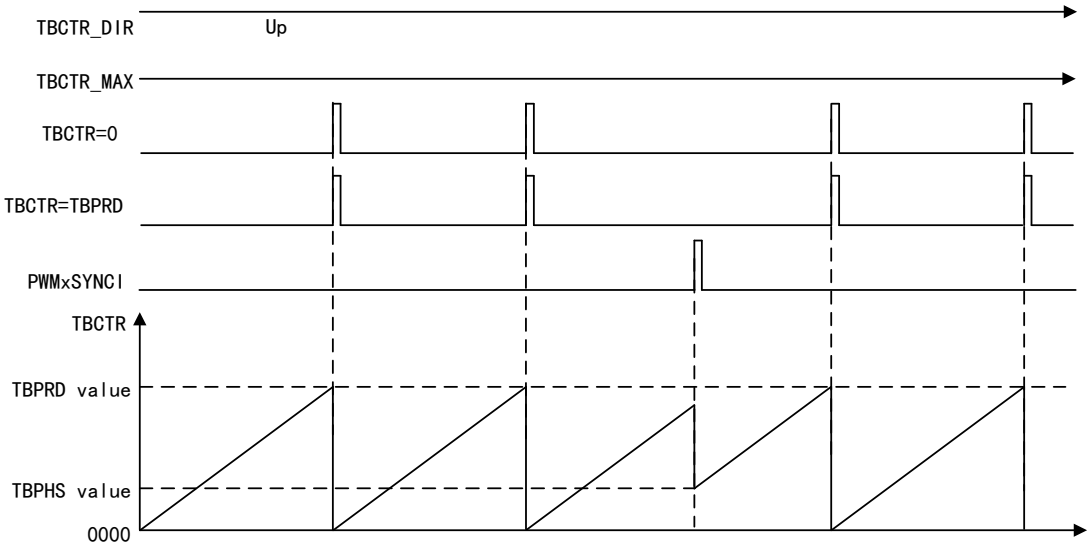
TBCTR 四种工作模式如下：

- 向上计数模式
- 向下计数模式
- 中心对齐计数模式
- 冻结模式

在向上计数模式、向下计数模式和中心对齐计数模式下，产生事件的时间以及时基响应 PWMxSYNCI 信号的方式如下图所示。

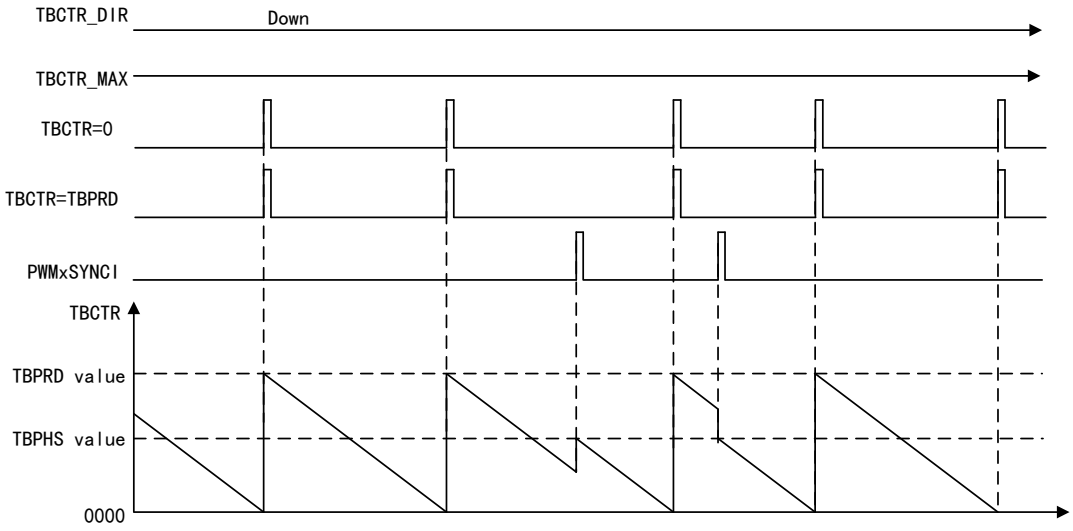
向上计数模式

图 79 向上计数模式时序图



向下计数模式

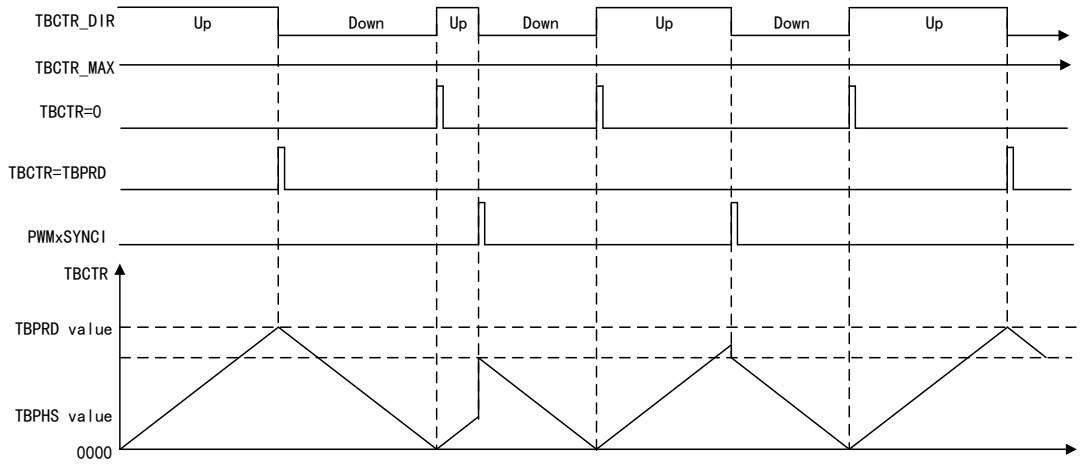
图 80 向下计数模式时序图



中心对齐计数模式

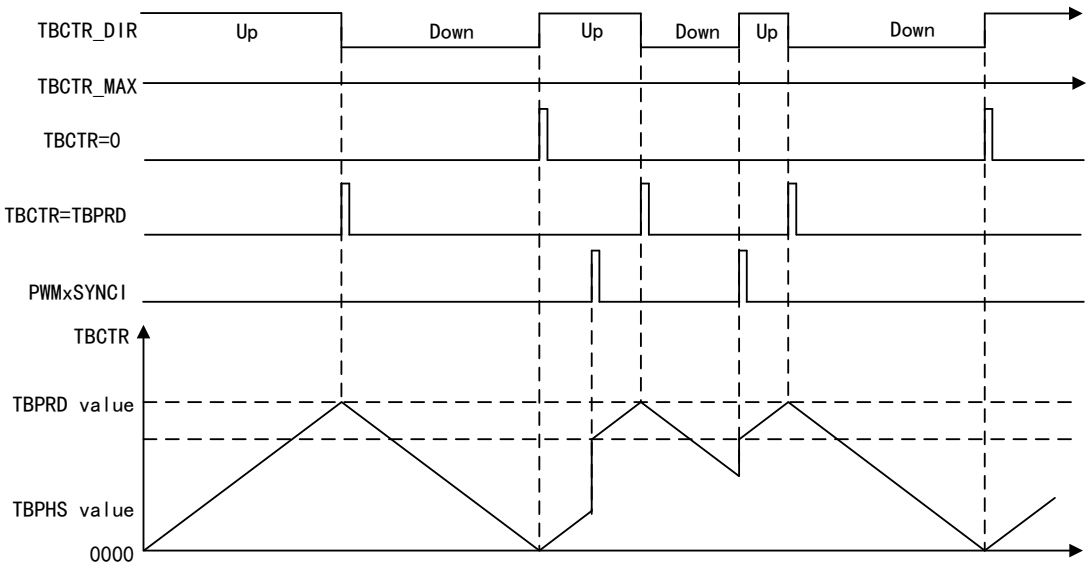
在该模式下，TBCTR 计数方向在同步事件后向下计数。

图 81 中心对齐计数模式时序图（PHSDIR=0）



在该模式下，TBCTR 计数方向在同步事件后向上计数。

图 82 中心对齐计数模式时序图（PHSDIR=1）



冻结模式

在该模式下，TBCTR 维持当前值不变。

31.5.3.4 同步

时基时钟同步

外设时钟使能寄存器中的 TBCLKSYNC 位可以全局同步所有已使能的 PWM 模块，使其与 TBCLK 同步。当置位时，所有已使能的 PWM 模块时钟同时启动，并与 TBCLK 的第一个上升沿对齐。为了最佳同步的 TBCLK，每个 PWM 模块的预分频器必须设置成一样的。

使能 PWMCLK 的操作步骤如下：

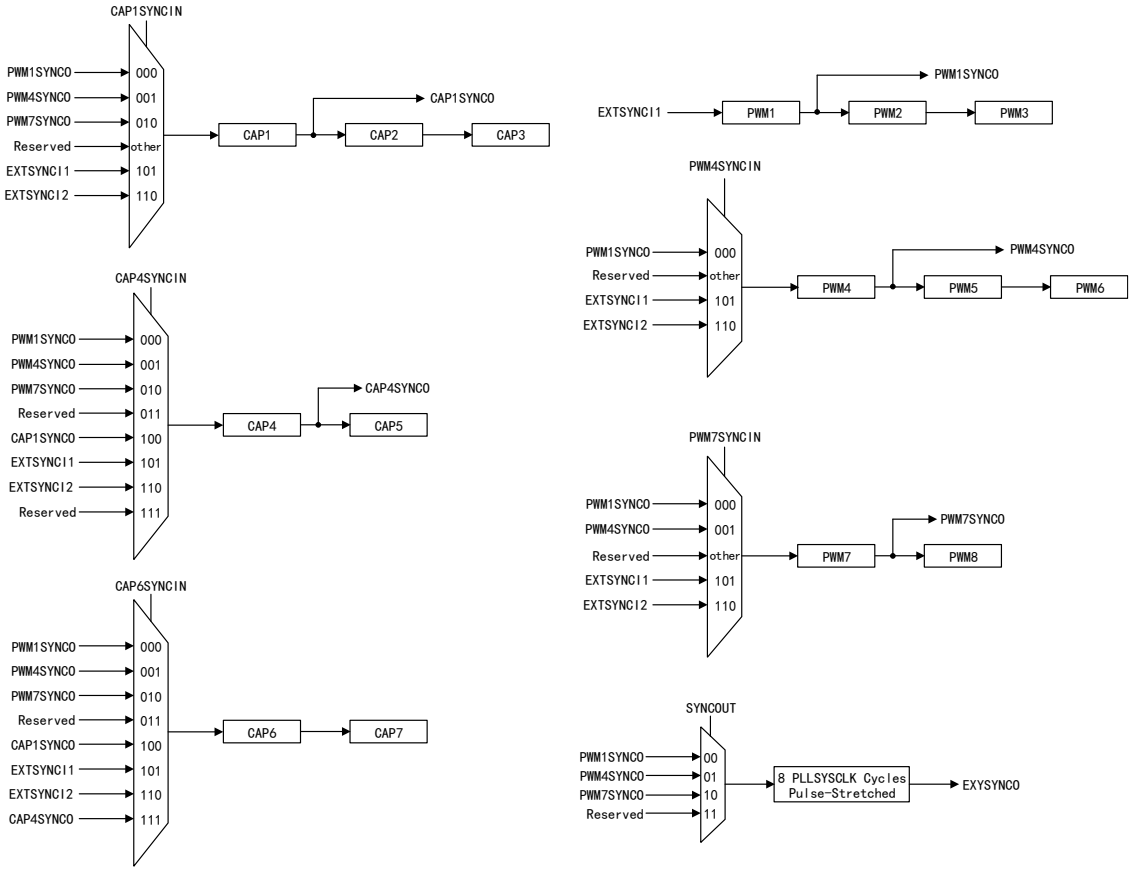
- 使能 PCLKCRx 寄存器中的 PWMx 模块时钟

- 清除 TBCLKSYNC
- 配置 PWM 模块
- 设置 TBCLKSYNC

时基计数器同步

同步方案增加了 PWM 模块同步的灵活性。每个 PWM 模块都有一个同步输入/输出和外设同步输出。如时基计数器同步结构框图所示，EXTSYNCIN1/2 分别源自 INPUTXBAR5/6，可以选择所有 GPIO 作为同步输入。要确保最长路径不超过 4 个 PWM/CAP 模块才能正确使用 SYNCSELECT 寄存器配置同步链传播路径。

图 83 时基计数器同步结构框图



注意：在特定设备上可用的 PWM 和 CAP 模块的数量详情请参见数据手册。

每个 PWM 模块可以配置为使用或忽略同步输入。当置位 TBCTL[PHSEN]时，满足其中一个条件，PWM 模块的 TBCTR 将自动装载 TBPHS 寄存器的内容。

表格 133 TBCTR 自动装载 TBPHS 寄存器内容的条件

条件	描述
输入同步脉冲	当检测到输入同步脉冲（PWMxSYNCI）时，将 TBPHS 寄存器的值装载到 TBCTR 寄存器中。该操作发生在下一个有效的 TBCLK 边沿。
软件强制同步脉冲	向 SWFSYNC 位写 1 会调用软件强制同步。该脉冲与同步输入信号相或，所以与输入同步脉冲具有相同的效果。

条件	描述
数字比较事件同步脉冲	数字比较事件 DCAEVT1 和 DCBEVT1 可配置为产生与输入同步脉冲具有相同效果的同步脉冲。

注意：当 PWMxSYNCI 信号保持高电平时，不会持续发生同步。PWMxSYNCI 信号上升沿有效。

这个功能使两个 PWM 模块之间能够自动实现时基同步。通过同步不同的 PWM 模块，可以对它们生成的波形增加超前或滞后的相位控制。在中心对齐计数模式下，同步事件发生后通过 PHSDIR 位配置 TBCTR 的方向。新的方向独立于同步事件之前的方向。在向上或向下计数模式下会忽略 PHSDIR 位，如三种计数模式的时序图所示。

清零 TBCTL[PHSEN]位可以让 PWM 模块忽略同步输入脉冲，但是同步脉冲仍然可以传至 PWMxSYNCO 并用于同步其他 PWM 模块。这使得我们可以设立一个 PWM1 这样的控制器时基，使下游模块其他 PWM 与该控制器同步运行，详情请参见电源拓扑应用章节。

同步 PWM 模块时钟

通过 TBCLKSYNC 位来全局同步该器件上所有已使能的 PWM 模块的时钟。当 TBCLKSYNC=0 时，停止所有 PWM 模块的时基时钟。当 TBCLKSYNC=1 时，在对齐的 TBCLK 上升沿使能所有 PWM 时基时钟。为了最佳同步的 TBCLK，每个 PWM 模块的预分频器必须设置成一样的。

同步 PWM 时钟操作步骤如下：

- 使能各 PWM 模块的时钟，详情请参见系统控制和中断章节
- 清除 TBCLKSYNC，任何已使能的 PWM 模块中的时基时钟都将被停止
- 配置期望的 PWM 模式和预分频值
- 设置 TBCLKSYNC

31.5.3.5 同时写入多个 TBPRD 和 CMPx 寄存器

在变频应用中，TBPRD 和 CMPx 寄存器需要在 PWM 模块之间同时写入。这避免了在修改完所有 PWM 的寄存器之前，TBCTR=0 或 TBCTR=TBPRD 脉冲强制影子寄存器装载到活动寄存器，从而导致部分寄存器从新的影子值加载，而其他寄存器从旧的影子值加载。为了支持这一点，在 CMPx 寄存器增加了 PWM 模块间寄存器的链接方案。

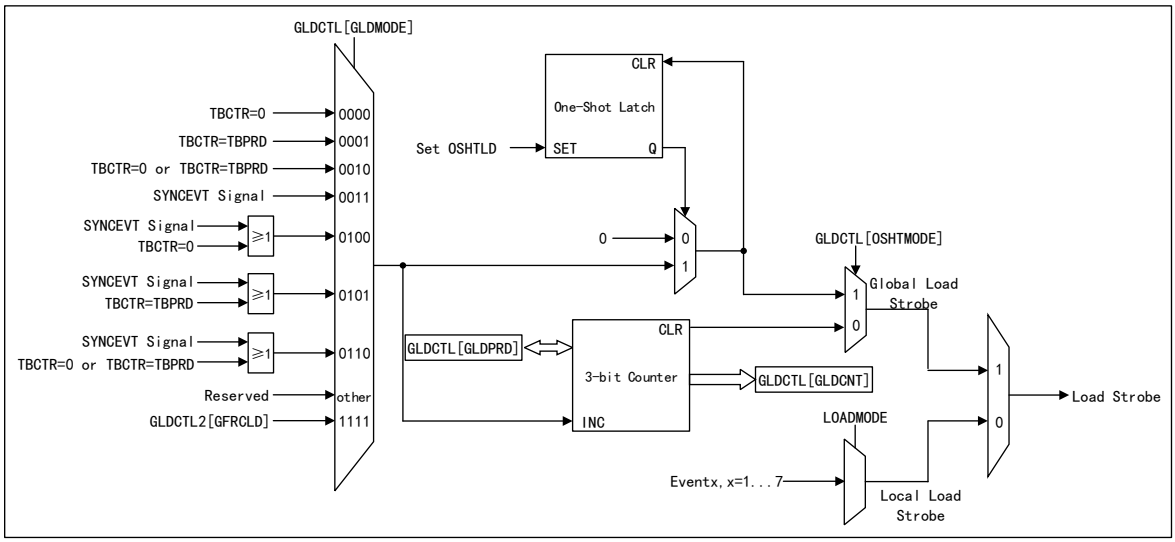
对于特定的 PWM 模块#A，用户代码将“B+1”写入 PWMXLINK 寄存器中，被链接寄存器对应的位域。“B”是被链接的 PWM 模块编号，也就是说，写入 PWM 模块“B”的 TBPRD、CMPA：CMPAHR、CMPB：CMPBHR 或 CMPC 会同时写入 PWM 模块“A”中相应的寄存器。例如，如果 PWM3 PWMXLINK 寄存器配置为 CMPA：CMPAHR 链接到 PWM1，那么对 PWM1 的 CMPA：CMPAHR 的写入将同时向 PWM3 中的 CMPA：CMPAHR 写入相同的值。如果 PWM4 也将其 CMPA：CMPAHR 寄存器链接到 PWM1，那么对 PWM1 的写入将向 PWM3 和 PWM4 中的 CMPA：CMPAHR 寄存器写入相同的值。

31.5.3.6 全局装载

如果使能全局装载功能，则对于所有使能此模式的寄存器，影子寄存器内容到活动寄存器的转移发生在同一时刻，由 GLDCTL[GLDMODE]位域决定该时刻。当 GLDCTL[GLD]=1 时，忽略各寄存器自己的影子寄存器到活动寄存器的加载事件选择，而由 GLDCFG 寄存器使能的对应寄存器的全局加载配置。

当 GLDCTL[GLD]=1 且 GLDCFG 寄存器中相应位为 0 时，相应寄存器不受全局加载模式影响。影子寄存器装载到活动寄存器的时刻由各寄存器自己的影子寄存器到活动寄存器加载事件选择位决定。

图 84 全局装载结构框图



注意：仅在 PHSEN 置位时才可以通过 SYNCEVT 信号。

脉冲预分频功能

该特性提供了如下配置：GLDCTL[GLDMODE]位域所选择的全局加载事件每发生“N”次，就发生一次影子寄存器到活动寄存器的转移。对于不能或没有配置为使用全局加载机制的寄存器，即 GLDCTL[GLD]=0 或 GLDCFG 寄存器中相应位为 0 时，此预分频功能不可用。

单次装载模式

限制影子寄存器到活动寄存器的转移只发生一次。当 OSHTLD=1 时，所有配置为使用全局加载机制的寄存器，在 GLDCTL[GLDMODE].位域所选事件出现时，发生影子寄存器到活动寄存器的转移，同时 GFRCLD 位被自动清零，若要再次发生全局加载，需手动将该位置 1。

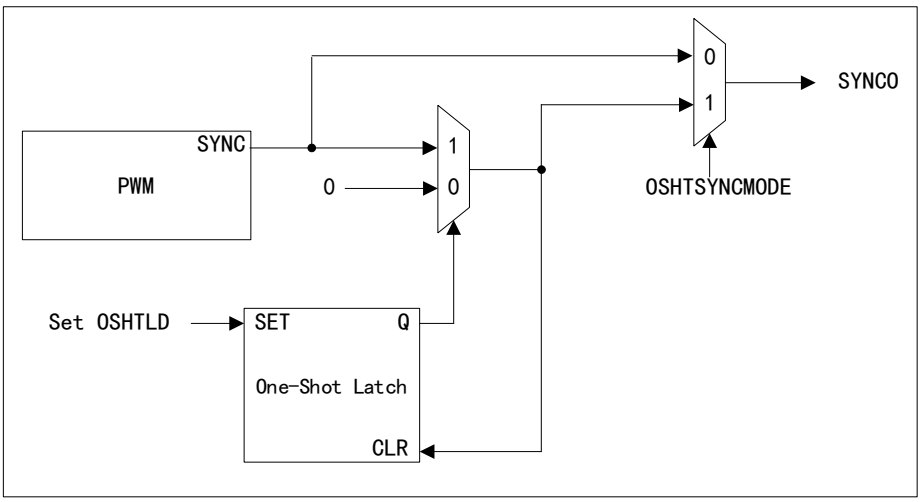
使用 GFRCLD 位可以通过软件强制影子寄存器到活动寄存器的转移。GLDCTL2 寄存器也可以通过 GLDCTL2LINK 位链接到多个 PWM 模块。结合上述的单次装载特性，这提供了一种在特定的 PWM 事件，甚至单个时钟周期内，精确地更新一个或多个 PWM 模块内的多个 PWM 寄存器的方法，可以用于变频应用和/或多相交错应用中。

注意：在高分辨率模式下禁止单次装载模式。

单次同步模式

需要配置 OSHTSYNCMODE 位和 OSP 位才能使能单次同步模式以生成同步输出信号（SYNCOUT）脉冲，如下图所示。

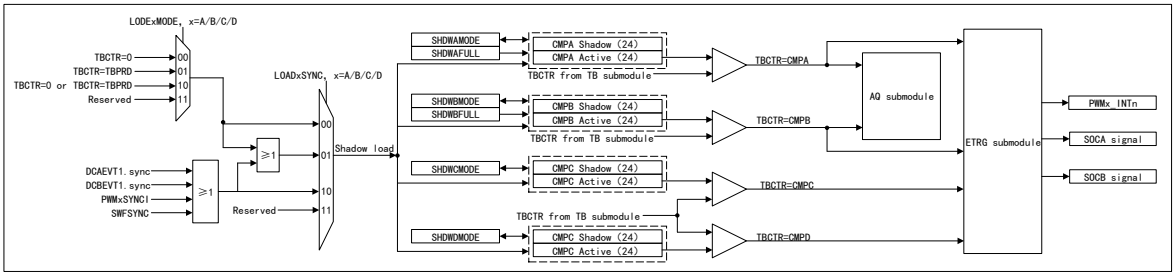
图 85 单次同步模式结构框图



31.5.4 计数器比较子模块功能描述

31.5.4.1 结构框图

图 86 计数器比较子模块结构框图

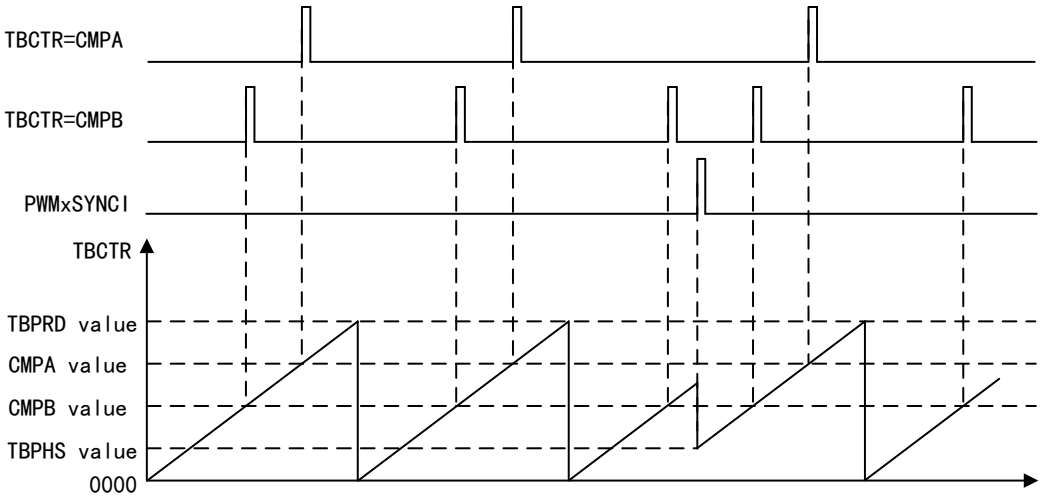


31.5.4.2 工作模式

CC 子模块可以在向上计数模式、向下计数模式和中心对齐模式下生成比较事件。三种计数模式下的计数器比较事件波形如下图所示，产生事件以及 PWMxSYNCI 信号起作用的时刻。

向上计数模式

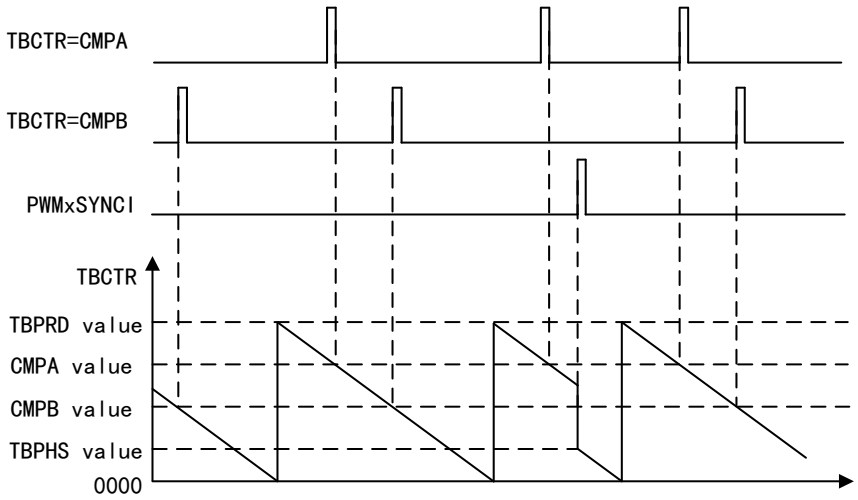
图 87 向上计数模式时序图



向下计数模式

PWMxSYNCI 外部同步事件会导致 TBCTR 计数值跳变。这会导致比较事件被跳过，如下图所示，同步事件导致 TBCTR=CMPA 事件被跳过。这种跳过现象被视为正常功能，必须纳入考虑。

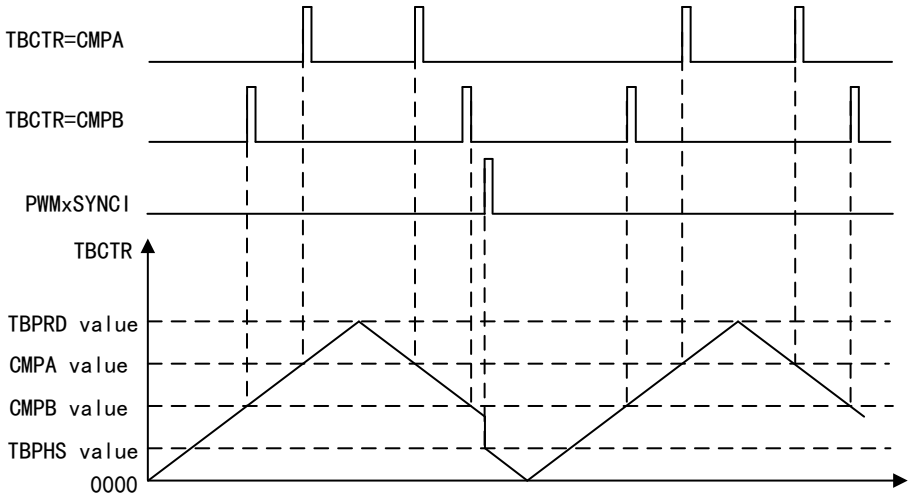
图 88 向下计数模式时序图



中心对齐计数模式

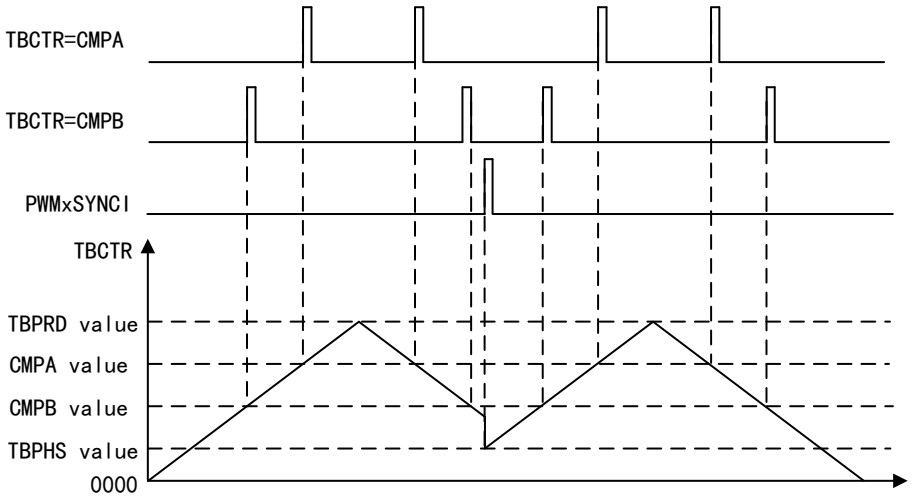
如下图所示，TBCTR 计数方向在同步事件后向下计数。

图 89 中心对齐计数模式时序图（PHSDIR=0）



如下图所示，TBCTR 计数方向在同步事件后向上计数。

图 90 中心对齐计数模式时序图（PHSDIR=1）



31.5.4.3 操作要点

CC 子模块负责产生事件，以供动作限定器且/或事件触发器使用。存在四种独立的比较事件：

- TBCTR=CMPA
- TBCTR=CMPB
- TBCTR=CMPC（仅供事件触发器使用）
- TBCTR=CMPD（仅供事件触发器使用）

对于向上计数或向下计数模式：每周期内每个事件仅发生一次

对于中心对齐计数模式：

- 若比较值等于 0x00 或 TBPRD，则每周期内每个事件发生一次
- 若比较值在 0x00 到 TBPRD 之间，则每周期内每个事件发生两次

这些事件被送入 AQ 子模块，在该子模块中，它们根据计数器的方向定性，并在使能时转换为相应的动作，详情请参见动作限定器子模块功能描述章节。

CMPA 和 CMPB 寄存器分别有一个相关联的，可以使寄存器更新与硬件同步的影子寄存器。为了防止由于软件异步修改寄存器而导致的波形失真或虚假操作，当使用影子寄存器时，活动寄存器的更新只发生在特定时间点上。活动寄存器和影子寄存器有相同的内存地址。由 SHDWAMODE 位和 SHDWBMODE 位决定写入或读取的寄存器。影子模式和立即装载模式的行为描述如下：

（1） 影子模式

清零 SHDWAMODE 位和 SHDWBMODE 位使能 CMPA 和 CMPB 寄存器的影子模式。默认使能 CMPA 和 CMPB 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADAMODE 位域、LOADBMODE 位域、LOADASYNC 位域和 LOADBSYNC 位域决定：

- TBCTR=0
- TBCTR=TBPRD
- TBCTR=0 且 TBCTR=TBPRD
- 同步事件：由 DCAEVT1 或 DCBEVT1 或 PWMx 同步输入或软件强制同步脉冲引起的
- LOADAMODE/LOADBMODE 选择的事件或同步事件

该子模块只使用活动寄存器的内容来产生送给动作限定器的事件。

（2） 立即模式

当 SHDWAMODE=1 或 SHDWBMODE=1，即选择立即加载模式时，那么对寄存器的读或写将直接访问活动寄存器。

附加比较器

CMP 子模块能够基于两个比较寄存器额外产生两路独立的比较事件，然后提供给 ETRG 子模块：

- TBCTR=CMPC
- TBCTR=CMPD

CMPC 和 CMPD 寄存器分别有一个默认使能的相关联影子寄存器。活动寄存器和影子寄存器有相同的内存地址。如果 CMPC 和 CMPD 活动寄存器的值与 TBCTR 相等，则 CMP 子模块分别产生一个 TBCTR 等于计数比较 C 事件或计数比较 D 事件。通过 SHDWCMODE 位域和 SHDWDMODE 位域来使能或禁止 CMPC 和 CMPD 影子寄存器，影子模式和立即装载模式的行为描述如下：

（1） 影子模式

清零 SHDWCMODE 位和 SHDWDMODE 位使能 CMPC 和 CMPD 寄存器的影子模式。默认使能 CMPC 和 CMPD 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADCMODE 位域、LOADDMODE 位域、LOADCSYNC 位域和 LOADDSYNC 位域决定：

- TBCTR=0

- TBCTR=TBPRD
- TBCTR=0 且 TBCTR=TBPRD
- 同步事件：由 DCAEVT1 或 DCBEVT1 或 PWMx 同步输入或软件强制同步脉冲引起
- LOADCMODE/LOADDMODE 选择的事件或同步事件

该子模块只使用活动寄存器的内容来产生送给动作限定器的事件。

（2） 立即模式

当 SHDWCMODE=1 或 SHDWDMODE=1，即选择立即加载模式时，那么对寄存器的读或写将直接访问活动寄存器。

全局装载

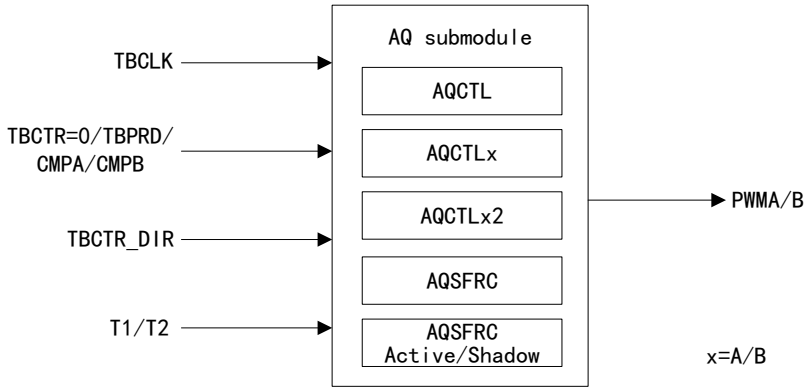
可以通过在 GLDCFG 寄存器中配置相应位，使得所有的计数器比较寄存器可以使用全局装载控制机制。如果使能全局装载功能，则使能此模式的寄存器中的影子寄存器内容装载到活动寄存器时，将同时发生在由 GLDCTL 寄存器定义的事件上。全局加载控制机制详情请参见全局装载章节。

31.5.5 动作限定器子模块功能描述

动作限定器子模块用于波形构造和 PWM 生成，并决定将哪些事件转换为各种动作类型，从而在 PWMxA 和 PWMxB 输出上产生所需的开关波形。

31.5.5.1 结构框图

图 91 动作限定器子模块结构框图



31.5.5.2 输入与输出

动作限定器子模块输入

AQ 子模块可以用作输入的事件如下所示：

表格 134 输入事件

输入事件	描述
TBCTR=0	时基计数器等于 0，即 TBCTR 寄存器等于 0
TBCTR=TBPRD	时基计数器等于周期值，即 TBCTR 寄存器等于 TBPRD 寄存器

输入事件	描述
TBCTR=CMPA	时基计数器等于计数器比较 A，即 TBCTR 寄存器等于 CMPA 寄存器
TBCTR=CMPB	时基计数器等于计数器比较 B，即 TBCTR 寄存器等于 CMPB 寄存器
TZ1 事件	基于比较器、跳闸、或同步输入事件
TZ2 事件	基于比较器、跳闸、或同步输入事件
软件强制事件	软件发起的异步事件，由 AQSFRC 和 AQCSFRC 寄存器控制

注意：如果不期望 CSFA 位域在影子模式下工作，则必须配置 RLDCSF 位域为禁用影子模式。

动作限定器子模块输出

AQ 子模块控制 PWMxA 和 PWMxB 两个输出在特定事件发生时的行为。该子模块的事件输入由计数器方向（向上或向下）进一步定性，这可以在计数器向上计数阶段和向下计数阶段产生相互独立的动作。

表格 135 可能施加在 PWMxA 和 PWMxB 上的动作

动作	描述
置高	PWMxA 或 PWMxB 输出被置为高电平
置低	PWMxA 或 PWMxB 输出被置位低电平
翻转	将 PWMxA 或 PWMxB 输出信号的状态进行翻转。若 PWMxA 或 PWMxB 当前为高，则将其拉低。反之将其拉高
无动作	保持 PWMxA 和 PWMxB 当前的电平不变。虽然该选项阻止事件在 PWMxA 和 PWMxB 输出上产生动作，但这些事件仍然可以触发中断或 ADCSOC。详情请参见事件触发子模块功能描述章节

PWMxA 或 PWMxB 输出指定的动作都是相互独立的，在给定的输出上所有事件均可生成动作。例如，TBCTR=CMPA 和 TBCTR=CMPB 都可作用于 PWMxA 输出。通过相关 PWM 控制寄存器可以配置所有限定器动作。

AQTSRCSEL 寄存器用于选择 T1 和 T2 事件源。AQ 子模块的跳闸或数字比较事件的 T1/T2 选择和配置，独立于 TZ 子模块中该事件的配置。一个特定的跳闸事件可以被配置为在 TZ 子模块中引起或不引起跳闸动作，但动作限定器可以使用相同的事件来生成 T1/T2 以控制 PWM 的产生。

31.5.5.3 工作模式

如下图所示使用静态比较寄存器值的 PWM 行为。在运行系统中，在每个周期结束时 CMPA 和 CMPB 活动寄存器从各自的影子寄存器中更新一次，可以指定在 TBCTR=0 或 TBCTR=TBPRD 时发生更新。在特定情况下，可以将基于旧值的动作额外生效一个周期，或者将基于新值的动作延迟一个周期。以下 PWM 配置可以避免这种情况：

向上计数模式

（1） 生成非对称 PWM

要实现占空比为 0%-100%的非对称 PWM，使用以下配置：

- 在 TBCTR=TBPRD 时装载 CMPA 或 CMPB
- 使用 TBCTR=0 动作拉高 PWM
- 使用向上计数时的比较动作拉低 PWM
- 调节比较值在 0 到 TBPRD+1 之间变化，可实现 0%-100%的 PWM 占空比

（2） 生成带死区的非对称 PWM

要实现占空比为 0%-100%的非对称 PWM，使用以下配置：

- 当 CMPA 值过于接近 0 或 TBPRD 时，如果 CMPx 小于死区或 CMPx 大于（TBPRD-死区），则 AQCTL 寄存器为 CMPx 指定的动作不起作用
- 为了避免这种情况，只有在向上计数时计数器等于 CMPA 或在向下计数时计数器等于 CMPA 事件（均为置高或置低时）产生高脉冲或低脉冲时才能改变 AQCTL 寄存器设置
- 确保使能影子模式且软件更新与 PWM 载波周期同步

中心对齐计数模式

（1） 生成对称 PWM

- 在等于零时装载 CMPA 或 CMPB，使用大于 1 的 CMPA 或 CMPB 值
- 在等于周期时装载 CMPA 或 CMPB，使用小于 TBPRD-1 的 CMPA 或 CMPB 值

在一个 PWM 周期中始终至少有一个 TBCLK 周期的脉冲的存在，当它很短时，系统往往会忽略它。

（2） 生成非对称 PWM

要实现占空比为 50%-0%的非对称 PWM，使用以下配置：

- 在 TBCTR=TBPRD 时装载 CMPA 或 CMPB
- 使用 TBCTR=TBPRD 动作拉低 PWM
- 使用向上计数时的比较动作拉高 PWM
- 调节比较值在 0 到 TBPRD 之间变化，可实现 50%-0%的 PWM 占空比

（3） 生成带死区的非对称 PWM

要实现占空比为 0%-100%的非对称 PWM，使用以下配置：

- 当 CMPA 值过于接近 0 或 TBPRD 时，如果 CMPx 小于死区/2 或 CMPx 大于（TBPRD-死区/2），则 AQCTL 寄存器为 CMPx 指定的动作不起作用
- 为了避免这种情况，只有在向上计数时计数器等于 CMPA 或在向下计数时计数器等于 CMPA 事件（均为置高或置低时）产生高脉冲或低脉冲时才能改变 AQCTL 寄存器设置
- 确保使能影子模式且软件更新与 PWM 载波周期同步

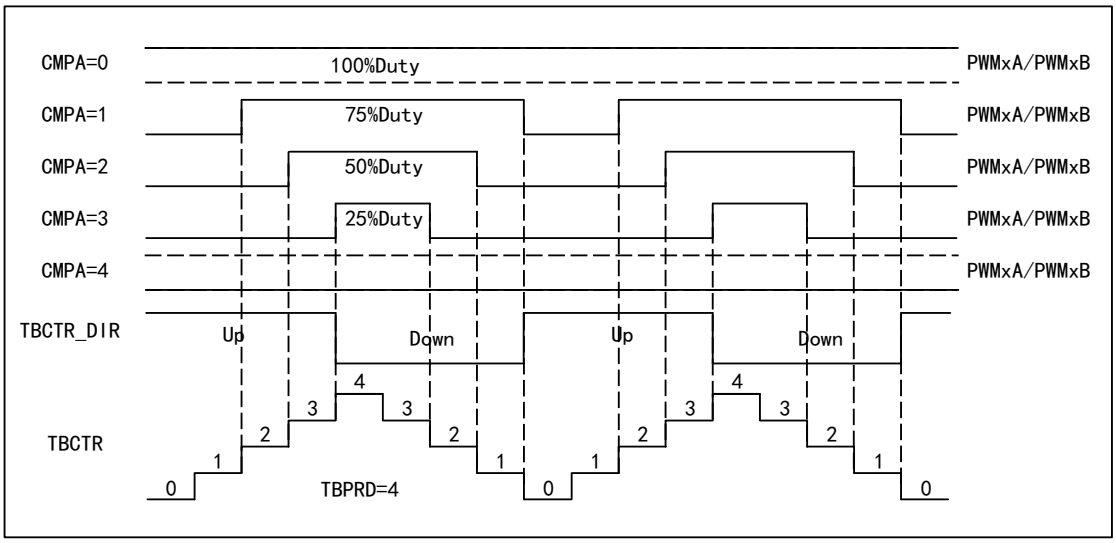
使用 TBCTR 的中心对齐计数模式生成对称 PWM 波形如下图所示。在该模式下，0%-100%的直流调制是通过在波形向上计数部分和向下计数部分匹配的等于

比较来实现的。使用 CMPA 进行比较示例：

- 当计数器向上计数时，CMPA 匹配将 PWM 输出拉高
- 当计数器向下计数时，比较匹配将 PWM 信号拉低
- 当 CMPA=0 时，PWM 信号在整个周期内为高电平，形成 100%占空比波形
- 当 CMPA=TBPRD 时，PWM 信号为低，达到 0%占空比

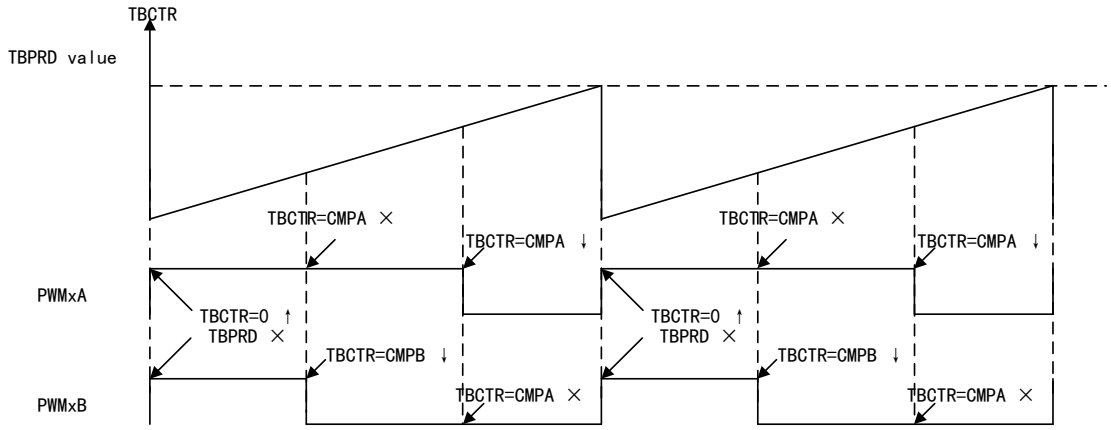
在实际使用此配置时，需要使用大于或等于 1 的 CMPA/B 值才允许在 TBCTR=0 时装载 CMPA/B。需要使用小于或等于 TBPRD-1 的 CMPA/B 值才允许在 TBCTR=TBPRD 时装载 CMPA/B。因此，在 PWM 周期中至少有一个 TBCLK 周期的脉冲，当它很短时，系统往往会忽略它。

图 92 中心对齐计数模式对称时序图



常见动作限定配置的 PWM 波形如下图所示，在图和示例中使用 TBPRD、CMPA 和 CMPB 指写入各自寄存器的值且硬件使用活动寄存器；CMPx 指 CMPA 或 CMPB；PWMxA 和 PWMxB 指 PWMx 输出信号的约定。

图 93 向上计数模式，PWMxA 和 PWMxB 上独立调制的单边非对称波形图（高电平有效）



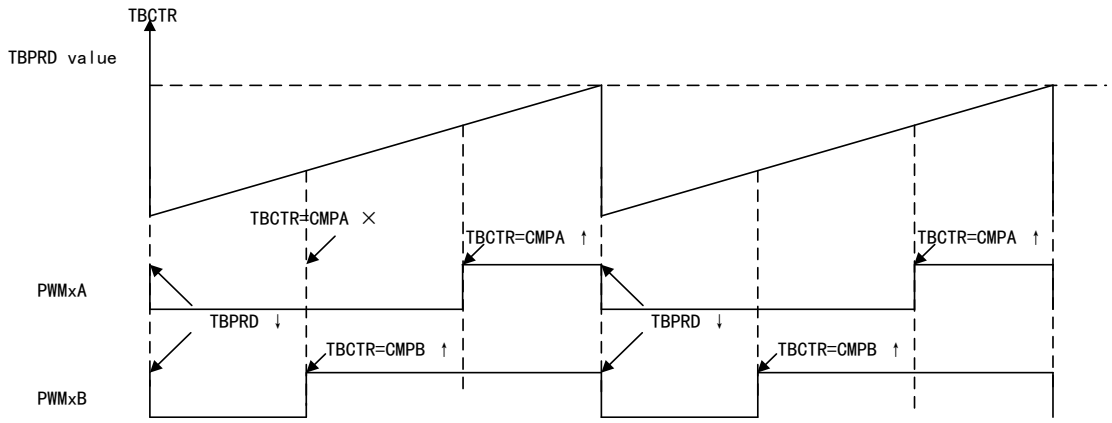
注意：

（1） PWM 周期 = T_{TBCLK} * （TBPRD+1）。

（2） 通过 CMPA/B 设置 PWMxA/B 的占空比调制且高电平有效。

（3） TBCTR 等于 0 和周期的动作不是同时发生的，它们被一个 TBCLK 周期分隔开。TBCTR 从周期结束为 0。

图 94 向上计数模式，PWMxA 和 PWMxB 上独立调制的单边非对称波形图（低电平有效）



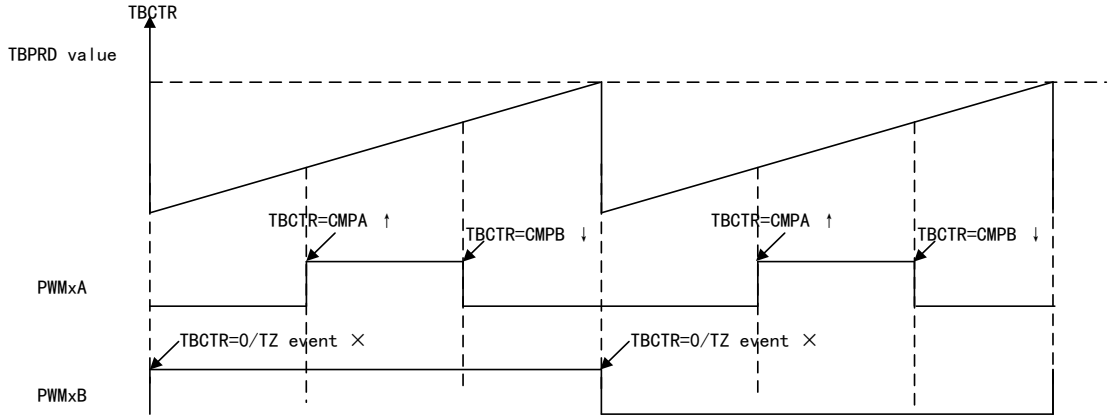
注意：

（1） PWM 周期 = T_{TBCLK} * （TBPRD+1）。

（2） 通过 CMPA/B 设置 PWMxA/B 的占空比调制且低电平有效。

（3） TBCTR 等于 0 和周期的动作不是同时发生的，它们被一个 TBCLK 周期分隔开。TBCTR 从周期结束为 0。

图 95 向上计数模式，PWMxA 上独立调制的脉冲放置型非对称波形图



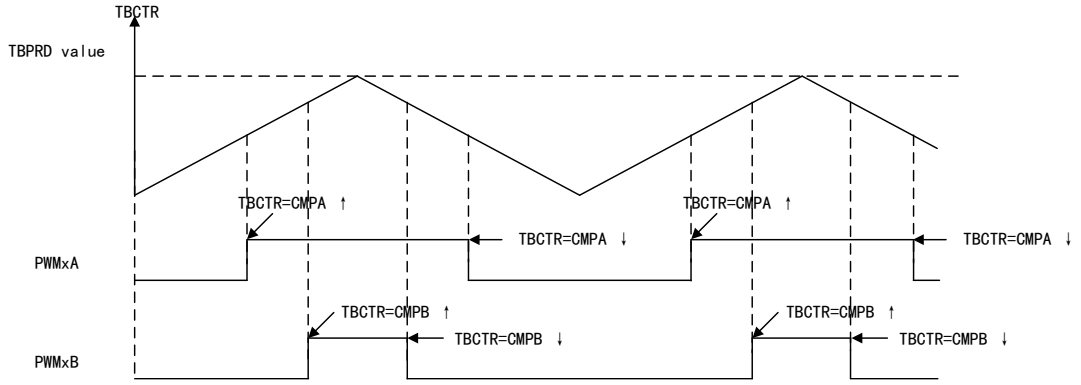
注意：

（1） PWM 频率 = 1/ [T_{TBCLK} * （TBPRD+1）]。

（2） 可以在 PWM 周期内（0~TBPRD）任意位置设置脉冲。

（3） 高电平占空比与（CMPB-CMPA）成比例关系。

图 96 中心对齐计数模式，PWMxA 和 PWMxB 上独立调制的双边沿对称波形图（低电平有效）



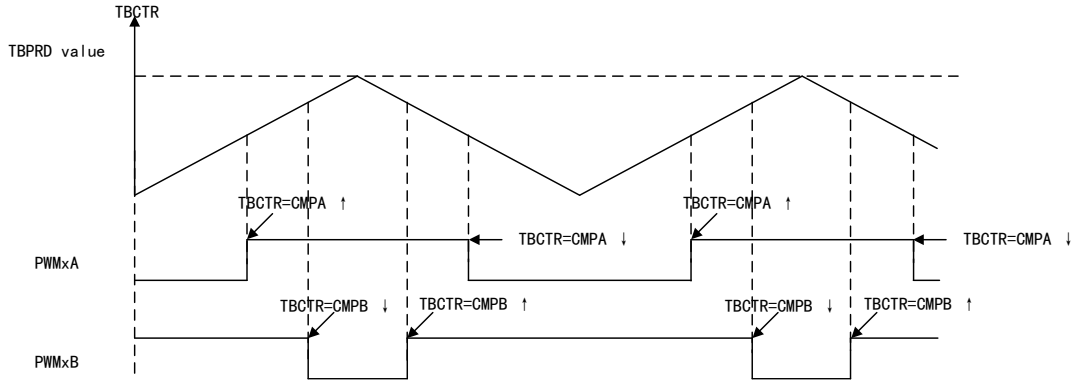
注意：

（1） PWM 周期 = 2* T_{TBCLK} * TBPRD。

（2） 通过 CMPA/B 设置 PWMxA/B 的占空比调制且低电平有效。

（3） 独立电源开关可以由输出 PWMxA 和 PWMxB 驱动。

图 97 中心对齐计数模式，PWMxA 和 PWMxB 上独立调制的双边沿对称波形图（互补）



注意：

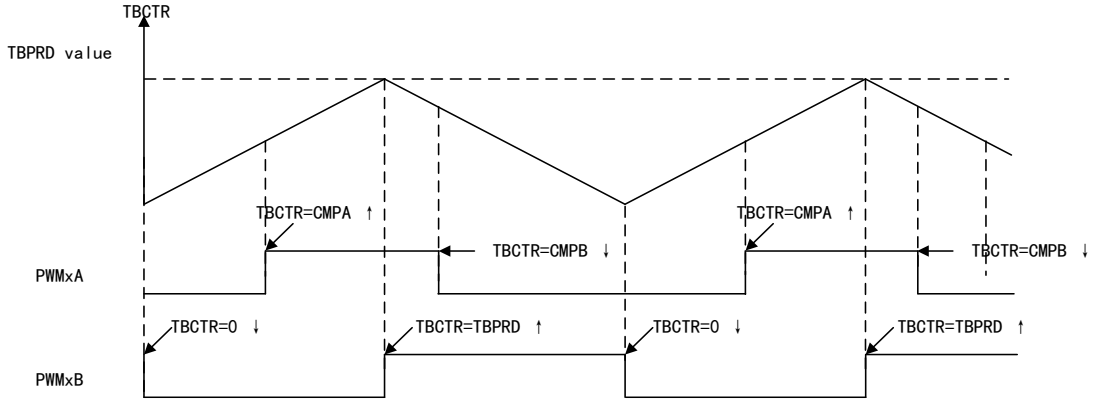
（1） PWM 周期 = 2* T_{TBCLK} * TBPRD。

（2） 通过 CMPA/B 设置 PWMxA/B 的占空比调制且低电平有效。

（3） 互补电源开关可以由输出 PWMxA 和 PWMxB 驱动。

（4） 通过软件完全可以编程边沿放置死区（死区=CMPB-CMPA），经典边沿延迟方法也适用于死区模块。

图 98 中心对齐计数模式，PWMxA 和 PWMxB 上独立调制的双边沿非对称波形图（低电平有效）



注意：

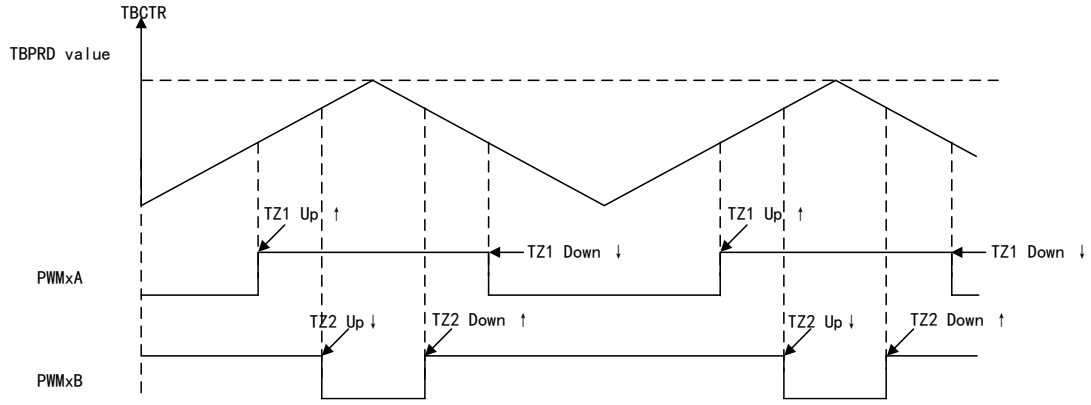
（1） PWM 周期 = 2* T_{TBCLK} * TBPRD。

（2） 可以 PWM 周期内不对称定位上升沿和下降沿。

（3） 通过 CMPA 和 CMPB 设置 PWMxA 的占空比调制，PWMxB 的占空比调制固定为 50%。（CMPA+CMPB）与 PWMxA 低电平占空比成正比。

（4） 如果将该示例更改为高电平有效，则需要在 CMPA 上置低和在 CMPB 上置高。

图 99 中心对齐计数模式，使用 T1 和 T2 事件生成 PWM 波形图



注意：

（1） PWM 周期 = 2* T_{TBCLK} * TBPRD。

（2） 当 TBCTR 向上计数和向下计数时，将独立 TZ1 和 TZ2 事件动作用来产生 PWMxA 和 PWMxB 输出。

31.5.5.4 事件优先级

PWM 动作限定器可以同时接收多个事件。硬件会为这些事件分配一个优先级。后发生的事件优先级更高，软件强制事件优先级最高。各种模式的动作限定事件优先级如下表格所示，优先级为 1 的优先级最高，级别越大则优先级最低。

在向上计数模式中，计数器方向总是向上，因此不会接收到向下计数事件。而在

向下计数模式中，计数器方向总是向下，因此不会接收到向上计数事件。在中心对齐计数模式中，TBCTR 的优先级随着方向的变化略有变化。

表格 136 各种模式的动作限定事件优先级

计数模式	优先级	事件
向上计数模式	1	软件强制事件
	2	TBCTR=TBPRD
	3	向上计数时的 TZ1
	4	向上计数时的 TZ2
	5	向上计数时计数器等于 CMPB
	6	向上计数时计数器等于 CMPA
	7	TBCTR=0
向下计数模式	1	软件强制事件
	2	TBCTR=0
	3	向下计数时的 TZ1
	4	向下计数时的 TZ2
	5	向上计数时计数器等于 CMPB
	6	向上计数时计数器等于 CMPA
	7	TBCTR=TBPRD
中心对齐计数模式	1	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：软件强制事件
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：软件强制事件
	2	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向上计数时的 TZ1
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向下计数时的 TZ1
	3	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向上计数时的 TZ2
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向下计数时的 TZ2
	4	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向上计数时计数器等于 CMPB
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向下计数时计数器等于 CMPB
	5	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向上计数时计数器等于 CMPA
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向下计数时计数器等于 CMPA

计数模式	优先级	事件
	6	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：TBCTR=0
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：TBCTR=TBPRD
	7	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向下计数时的 TZ1
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向上计数时的 TZ1
	8	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向下计数时的 TZ2
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向上计数时的 TZ2
	9	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向下计数时计数器等于 CMPB
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向上计数时计数器等于 CMPB
	10	TBCTR 向上计数时的事件，TBCTR=0 增至 TBCTR=TBPRD：向下计数时计数器等于 CMPA
		TBCTR 向下计数时的事件，TBCTR=TBPRD 减至 TBCTR=1：向上计数时计数器等于 CMPA

设置 CMPA/CMPB 大于周期时，事件将按如下表格所示发生：

表格 137 CMPA/CMPB 大于周期时的动作

计数模式	事件
向上计数模式	向上计数时比较事件： 当 CMPA/CMPB≤TBPRD 时，在比较值匹配时产生 TBCTR=CMPA 或 TBCTR=CMPA 事件。 当 CMPA/CMPB>TBPRD 时，不会产生事件。
	向下计数时比较事件： 不会产生事件
向下计数模式	向上计数时比较事件： 不会产生事件
	向下计数时比较事件： 当 CMPA/CMPB<TBPRD 时，在比较值匹配时产生 TBCTR=CMPA 或 TBCTR=CMPA 事件。 当 CMPA/CMPB≥TBPRD 时，在周期匹配时产生 TBCTR=TBPRD 事件。
中心对齐计数模式	向上计数时比较事件： 当 CMPA/CMPB<TBPRD 且计数器向上计数时，在比较值匹配时产生 TBCTR=CMPA 或 TBCTR=CMPA 事件。 当 CMPA/CMPB≥TBPRD 时，在周期匹配时产生 TBCTR=TBPRD 事件。

计数模式	事件
	向下计数时比较事件： 当 CMPA/CMPB<TBPRD 且计数器向下计数时，在比较值匹配时产生 TBCTR=CMPA 或 TBCTR=CMPA 事件。 当 CMPA/CMPB≥TBPRD 时，在周期匹配时产生 TBCTR=TBPRD 事件。

31.5.5.5 影子模式

动作限定器支持影子模式。此外，还支持在同步事件时发生影子寄存器到活动寄存器的加载。通过 SHDWAQAMODE 位和 SHDWAQBMODE 位来使能和禁止该寄存器的影子模式。影子模式和立即装载模式的行为描述如下：

（1） 影子模式

通过 SHDWAQAMODE 位和 SHDWAQBMODE 位来使能 AQCTLA 和 AQCTLB 寄存器的影子模式。默认禁止 AQCTLA 和 AQCTLB 寄存器的影子模式。

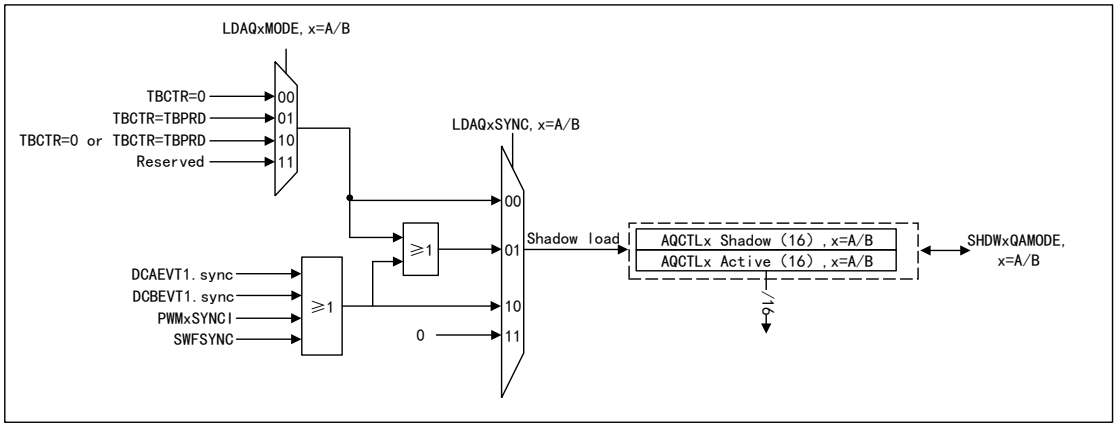
当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LDAQAMODE 位域、LDAQBMODE 位域、LDAQASYNC 位域和 LDAQBSYNC 位域决定：

- TBCTR=0
- TBCTR=TBPRD
- TBCTR=0 且 TBCTR=TBPRD
- 同步事件：由 DCAEVT1 或 DCBEVT1 或 PWMx 同步输入或软件强制同步脉冲引起的
- LDAQAMODE/LDAQBMODE 选择的事件或同步事件

（2） 立即模式

如下图所示，当 SHDWAQAMODE=0 或 SHDWAQBMODE=0，即选择立即加载模式时，那么对寄存器的读或写将直接访问活动寄存器。

图 100 SHDWAQAMODE/SHDWAQBMODE 结构框图



注意：

（1） 由 PWM DC 子模块基于 TRIPIN 输入的电平生成 DCxEVT1.sync 事件。

（2） 在 CMPA=0 或 CMPB=0 边界时，AQCTLA 和 AQCTLB 影子寄存器装载到活动寄存器。

（3） 当 CMPA 或 CMPB=0，且动作限定器对 AQCTLA 和 AQCTLB 寄存器的动作与影子寄存器装载到活动寄存器同一时刻发生时， 也就是说，当 CMPA=0 且 AQCTLA 在 TBCTR=0 时发生影子寄存器装载到活动寄存器时，两个事件就会产生冲突。因此，建议 AQCTLA 和 AQCTLB 寄存器在 TBCTR=0 边界使能影子模式时，使用非零的计数器比较值。

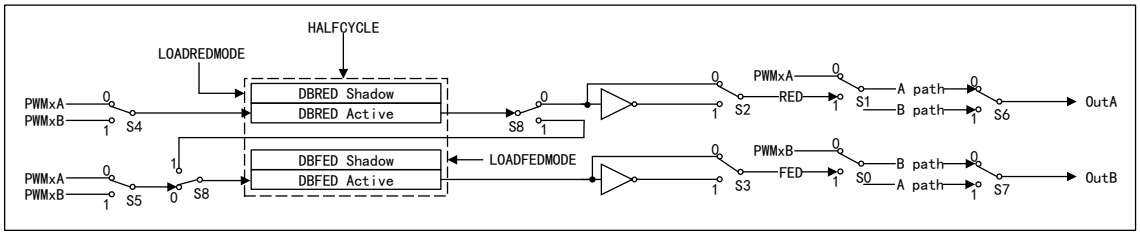
全局装载

可以通过在 GLDCFG 寄存器中配置相应位，也可以对 AQCTLA、AQCTLA2、AQCTLB、AQCTLB2 和 AQCSFRC 寄存器使用全局加载机制。如果使能全局装载，则相关寄存器的影子寄存器内容装载到活动寄存器同时发生在由 GLDCTL 寄存器定义的事件上。全局加载控制机制详情请参见全局装载章节。

31.5.6 死区发生器子模块功能描述

31.5.6.1 结构框图

图 101 死区发生器子模块结构框图



注意：S0 和 S1 由 OUT_MODE 位控制，S2 和 S3 由 POLSEL 位控制，S4 和 S5 由 IN_MODE 位控制，S6 和 S7 由 OUTSWAP 位控制，S8 由 DEDB_MODE 位控制。

31.5.6.2 增加的工作模式

死区发生器子模块增加的工作模式如下：

- 死区计数器为 14 位
- 通过 DBREDHR 和 DBFEDHR 寄存器使能高分辨率死区上升沿延迟和下降沿延迟
- 死区高分辨率寄存器和死区寄存器有影子寄存器

注意：

（1） 使能高分辨率死区时，将不能使用 PWM 斩波。

（2） 高分辨率死区 RED 和 FED 功能需要使能半周期时钟，即 HALFCYCLE=1。

（3） RED 和 FED 无法同时作用于 PWMxA 和 PWMxB。RED 和 FED 同时作用时，只可作用于 PWM 输出 A（OutA）或 PWM 输出 B（OutB）。

（4） 使 B 通道产生相对于 A 通道的相移：通过使用 DEDB_MODE 位，根据 PWMxA 派生得到 PWMxB，并将 RED 和 FED 相应的相移量。如果 PWMxA 的占空比小于相移量，则 PWMxA 的下降沿超前于 PWMxB 的上升沿。建议确保所需的相移量小于输入死区模块的当前波形占空比。

影子模式

通过 SHDWDBRED 位和 SHDWDBFED 位来使能 DBREDHR 和 DBFED 寄存器的影子模式。默认禁止 DBRED 和 DBFED 寄存器的影子模式。

此外，可以将 DBCTL 寄存器影子化。通过 SHDWDBCTLMODE 位使能 DBCTL 寄存器的影子模式。

当使能影子寄存器时，影子寄存器的内容将在以下事件之一转移到活动寄存器，这些事件由 LOADREDMODE 位域、LOADFEDMODE 位域和 LOADDBCTLMODE 位域决定：

- TBCTR=0
- TBCTR=TBPRD
- TBCTR=0 或 TBCTR=TBPRD

注意：必须在对 DBRED 和 DBFED 寄存器的值进行编程之前使能影子模式。如果在这些寄存器进行编程后使能影子寄存器，则这些寄存器被加载为 0。

全局装载

可以通过在 GLDCFG 寄存器中配置相应位，也可以对 DBCTL、DBRED：DBREDHR、DBFED：DBFEDHR 寄存器使用全局加载机制。如果使能全局装载功能，则对于所有使能此模式的寄存器，其影子寄存器内容装载到活动寄存器同时发生在由 GLDCTL 寄存器定义的事件上。

注意：

（1） 当 DBRED 或 DBFED 活动寄存器在死区计数器正在计数时，加载了新的影子值，新的 DBRED 或 DBFED 不会影响当前边沿，只会影响下一个 PWMx 边沿。

（2） 当全局装载时刻设定为 TBCTR=0x00 时，死区值不应为 0。同理，当全局装载时刻为 TBCTR=TBPRD 时，死区值不应为周期值。

（3） TBPRDHR 寄存器不能与全局装载一起使用。当在应用中必须改变高分辨率周期时，用户必须在 PWM 中断服务程序（ISR）写入单独的周期寄存器且该 ISR 必须与 PWM 切换周期同步，其中全局装载单次位也被写入。

31.5.6.3 操作要点

虽然死区发生器支持所有组合模式，但是经典死区工作模式只有七种。这些模式假设 IN_MODE 位域已配置 PWMxA In 为 RED 和 FED 的输入源。通过改变输入源可以实现非传统或增强模式。经典死区工作模式可以分为以下几类：

（1） 同时绕过 FED 和 RED：模式 1

允许用户在 PWM 信号路径上完全禁止 TB 子模块。

（2） 经典死区极性设置：模式 2-5

这些典型极性配置可以解决现有工业电源开关门驱所需的所有高电平有效和低电平有效模式，其时序图如 0%<占空比<100%典型模式的死区时序图所示。需要注意，如果要生成 0%<占空比<100%典型模式的死区时序图所示的等效波形，则需

要配置 AQ 子模块以生成 PWMxA 所示的信号。

（3） 绕过 RED：模式 6

绕过 RED 组合详情请参见死区工作模式分类章节。

（4） 绕过 FED：模式 7

绕过 FED 组合详情请参见死区工作模式分类章节。

死区工作模式分类

表格 138 死区配置

模式		模式描述	配置位						
			OUT_MODE	POLSEL	OUTSWAP	DEDB_MODE	OUT_MODE	POLSEL	OUTSWAP
			S0	S1	S2	S3	S6	S7	S8
典型模式	1	PWMxA 和 PWMxB 直通，同时绕过 RED 和 FED，即无延迟.	0	0	X	X	-	-	-
	2	高电平有效互补（AHC）	1	1	0	1	-	-	-
	3	低电平有效互补（ALC）	1	1	1	0	-	-	-
	4	高电平有效（AH）	1	1	0	0	-	-	-
	5	低电平有效（AL）	1	1	1	1	-	-	-
	6	PWMxA 输出 = PWMxA 输入 PWMxB 输出 = PWMxA 输入带 FED	1	0	0/1	0/1	-	-	-
	7	PWMxA 输出 = PWMxA 输入带 RED PWMxB 输出 = PWMxB 输入	0	1	0/1	0/1	-	-	-
附加模式		PWMxA 和 PWMxB 信号由 OUT_MODE 位域定义。	-	-	-	-	0	0	0
		PWMxA = 由 OUT_MODE 位域所定义的 A 路径。 PWMxB = 由 OUT_MODE 位域所定义的上升沿延迟或绕过延迟的 A 路径。	-	-	-	-	0	1	0
		PWMxA = 由 OUT_MODE 位域所定义的下降沿延迟或绕过延迟的 B 路径。 PWMxB = 由 OUT_MODE 位域所定义的 B 路径。	-	-	-	-	1	0	0

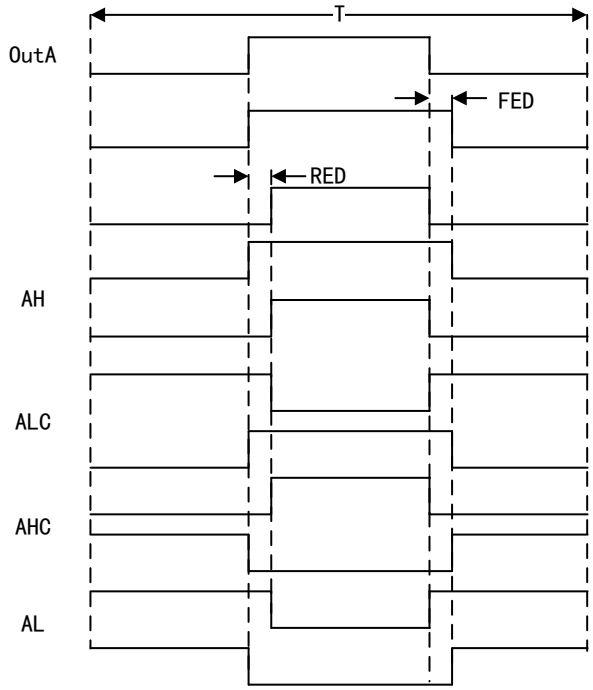
模式	模式描述	配置位						
		OUT_MODE	POLSEL	OUTSWAP	DEDB_MODE	OUT_MODE	POLSEL	OUTSWAP
		S0	S1	S2	S3	S6	S7	S8
	PWMxA = 由 OUT_MODE 位域所定义的下降沿延迟或绕过延迟的 B 路径。 PWMxB = 由 OUT_MODE 位域所定义的上升沿延迟或绕过延迟的 A 路径。	-	-	-	-	1	1	0
	RED 仅作用于 A 路径上 S4 开关（IN_MODE[4]位）所选的 PWMxA 或 PWMxB。 FRD 仅作用于 B 路径上 S5 开关（IN_MODE[5]位）所选的 PWMxA 或 PWMxB。	-	-	-	-	X	X	0
	RED 和 FRD 作用于 S4 开关（IN_MODE[4]位）所选的信号源，且仅输出到 B 信号路径。（1）	-	-	-	-	X	X	1

注意：（1）如果该位置位，用户则可以设置 OUT_MODE=01（使 A 路径等于 InA）或者设置 OUTSWAP=1x（使 PWMxA 输出信号等于 B 路径），否则 PWMxA 无信号。

上升沿延迟与下降沿延迟

DB 子模块支持独立的 RED 和 FED 值。通过 DBRED 和 DBFED 寄存器配置延迟的量，它们是 10-位寄存器，其值表示信号边沿延迟的 TBCLK 周期数。

图 102 0%<占空比<100%典型模式的死区时序图



（1） 开启全周期时钟

上升沿延迟计算公式为：RED=DBRED*T_{TBCLK}

下降沿延迟计算公式为：FED=DBFED*T_{TBCLK}

（2） 开启半周期时钟

（3） 上升沿延迟计算公式为：RED=DBRED*（T_{TBCLK}/2）

（4） 下降沿延迟计算公式为：FED=DBFED*（T_{TBCLK}/2）

其中 T_{TBCLK} 是 PWMCLK 预分频后的 TBCLK 的周期。

各种 TBCLK 选项的延迟值如下表格所示，其 PWM 输入时钟频率为 100MHz。

表格 139 各种 TBCLK 选项的延迟值

延迟值（DBRED 和 DBFED）	TBCLK			单位
	PWMCLK 4 分频	PWMCLK 2 分频	PWMCLK 不分频	
1	0.04	0.02	0.01	μs
5	0.2	0.1	0.05	
10	0.4	0.2	0.1	
100	4	2	1	

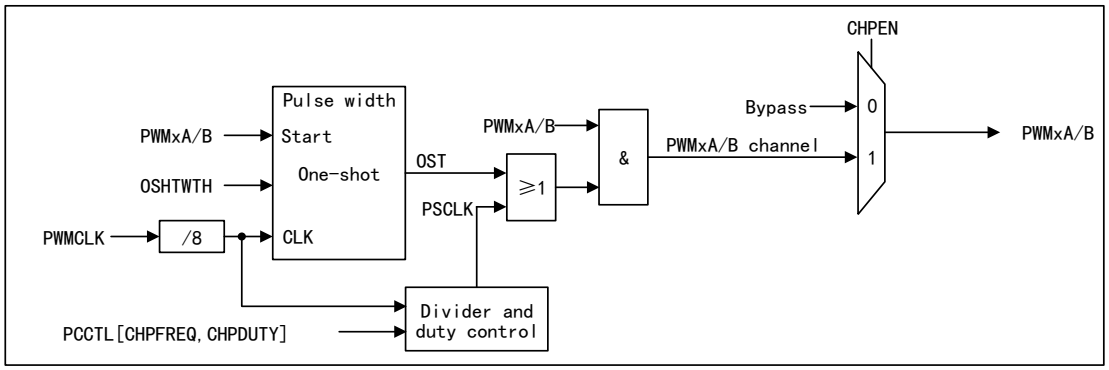
延迟值（DBRED 和 DBFED）	TBCLK			单位
	PWMCLK 4 分频	PWMCLK 2 分频	PWMCLK 不分频	
200	8	4	2	
400	16	8	4	
500	20	10	5	
600	24	12	6	
700	28	14	7	
800	32	16	8	
900	36	18	9	
1000	40	20	10	

31.5.7 PWM 斩波子模块功能描述

PWM 斩波器子模块允许高频载波信号对 AQ 和 DB 生成的 PWM 波形进行调制，该功能可用于需要基于脉冲变压器的门极驱动来控制功率开关元件。

31.5.7.1 结构框图

图 103 死区发生器子模块结构框图



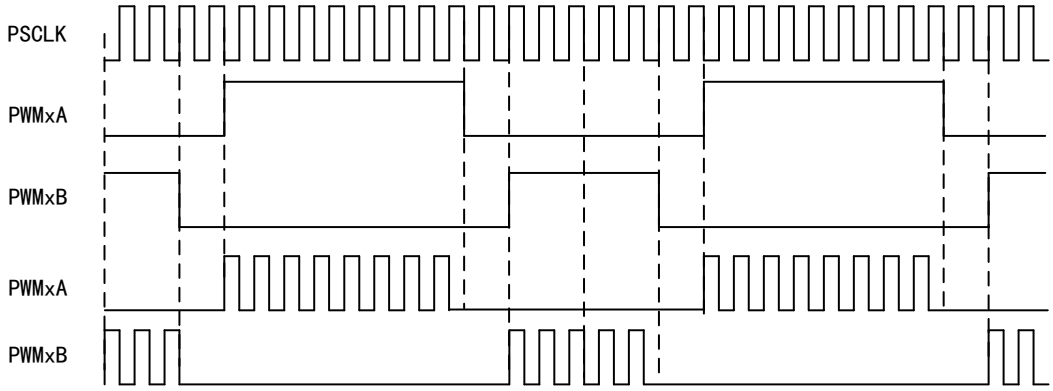
31.5.7.2 操作要点

- 载波时钟源自 PWMCLK
- 使用 CHPEN 位使能斩波功能
- 通过 OSHTWTH 位域可以编程单次脉冲宽度
- 由 CHPFREQ 位域和 CHPDUTY 位域分别控制斩波时钟的频率和占空比
- 单次块功能如下：
  - 提供一个高能量的首个脉冲，以确保功率开关稳定且快速地打开
  - 后续持续脉冲，以确保电源开关维持开启

31.5.7.3 波形

如下图所示仅有斩波动作的简化波形，其中未显示占空比和单次脉冲控制。

图 104 简单 PC 子模块时序图（仅有斩波动作）

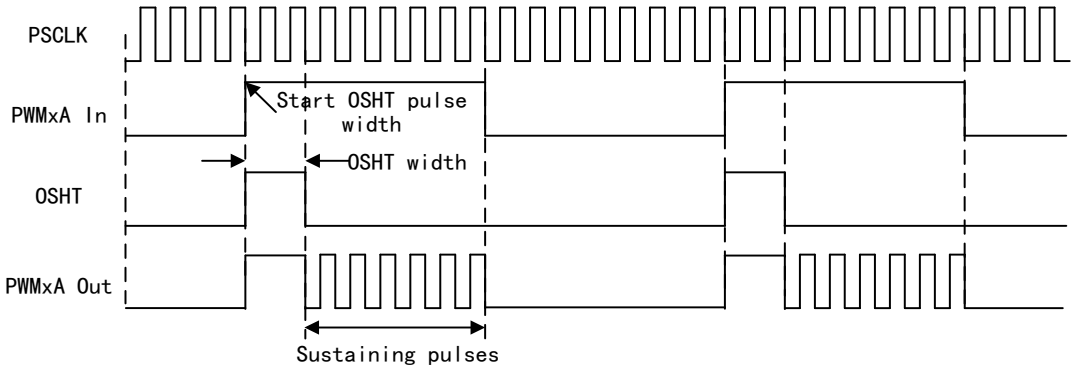


单次脉冲

可以将首个脉冲的宽度编程为任何一种脉冲宽度值（共 16 种）。T_{PWMCLK} 是系统时钟的周期，OSHTWTH 长度为 4 位，其值为 1 到 16，所以首个脉冲宽度或周期计算公式为：$T_{1st_pulse} = 8*T_{PWMCLK} * OSHTWTH$

如下所示首个脉冲和后续持续脉冲的 PWM 时序图以及 PWMCLK 为 80MHz 时的脉宽值。

图 105 首个脉冲和后续持续脉冲的 PWM 时序图



表格 140 PWMCLK 为 80MHz 时的脉宽值

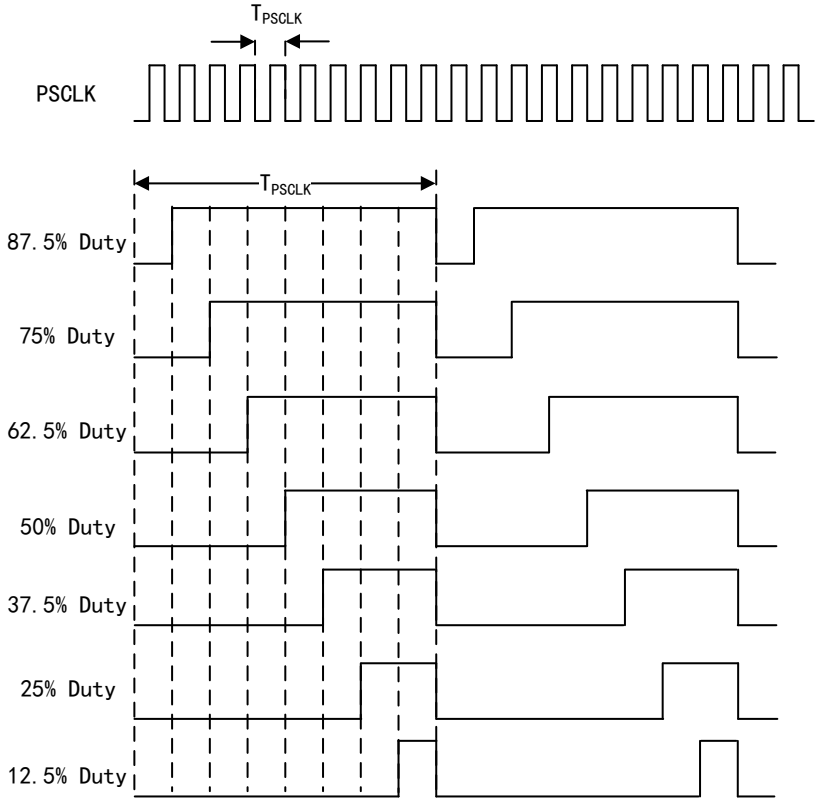
PWMCLK	OSHTWTH	脉宽值
80MHz	0x00	100ns
	0x01	200ns
	0x02	300ns
	0x03	400ns
	0x04	500ns
	...	...
	0x0D	1400ns
	0x0E	1500ns
	0x0F	1600ns

占空比控制

当设计基于脉冲变压器的门极驱动时，需要了解包括饱和度在内的变压器和相关电路的磁性能或特性。可编程第二个及以后续脉冲的占空比以辅助门极驱设计人员。这些持续脉冲确保在开启期间维持功率开关门极驱动强度和极性，可以使用软件控制来调整或优化设计可编程的占空比。

通过 CHPDUTY 位域配置任何一种占空比控制（共 7 种），其占空比范围为 12.5%~87.5%。

图 106 显示占空比控制的 PWM 时序图



31.5.8 跳闸区子模块功能描述

每个 PWM 模块连接到六个跳闸区信号，即 TZ1n 至 TZ6n：

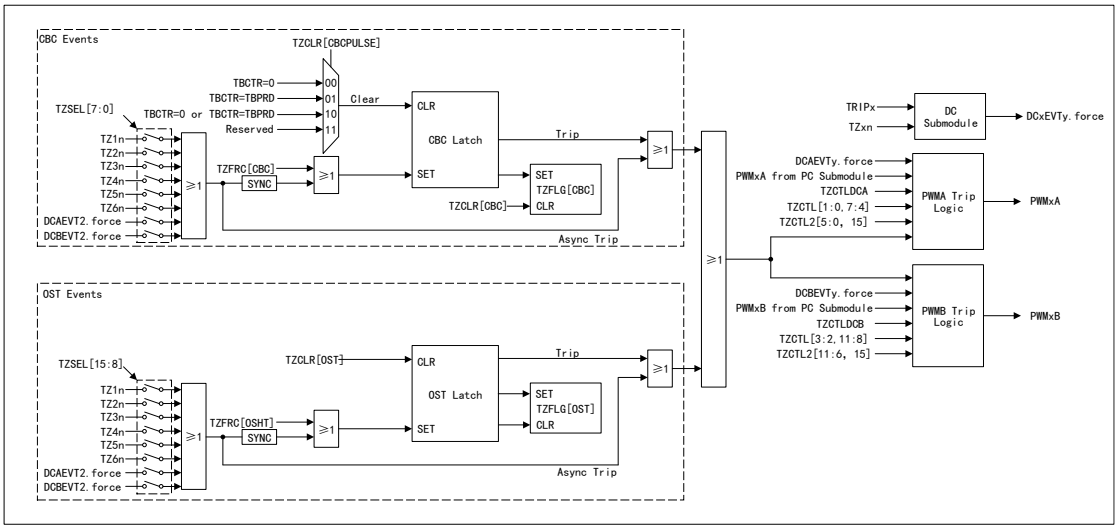
- TZ1n 至 TZ3n：来自在有 QEP 模块器件上的 GPIO 复用器
- TZ4n：来自反相的 QEPxERR 信号
- TZ5n：连接到系统时钟故障逻辑（CLOCLKn 信号）
- TZ6n：来自 CPU 的 EMUSTOP 输出

这些信号表示跳闸或外部错误状态，PWM 输出可以在故障发生时做出相应的响应。

31.5.8.1 控制逻辑

DCxEVTy 信号详情请参见 DC 子模块章节。

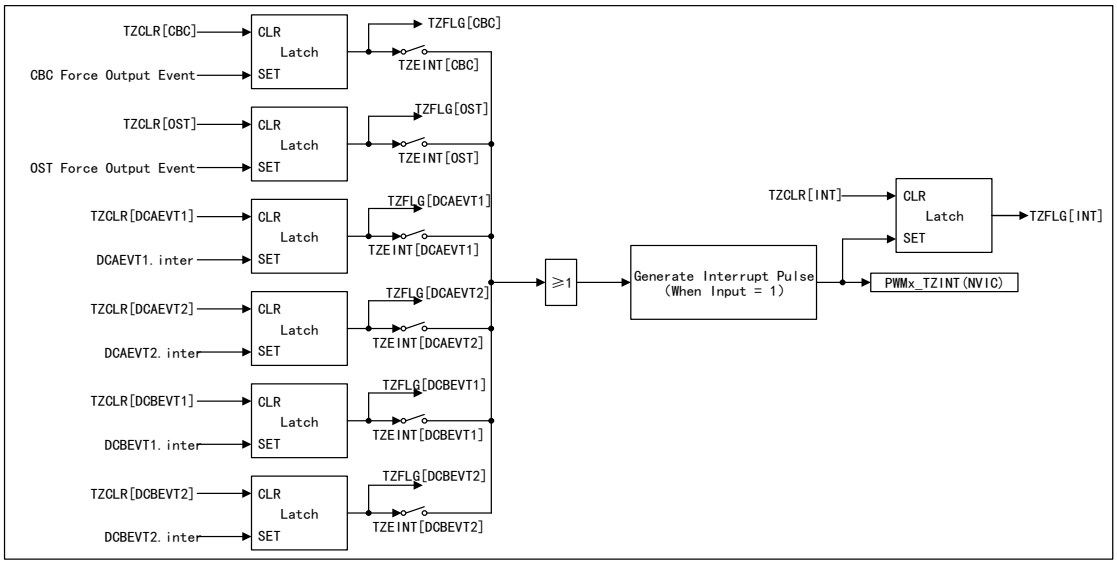
图 107 控制逻辑结构框图



31.5.8.2 中断

DCxEVTy 信号详情请参见 DC 子模块章节。通过 CBC、OST 和 DCxEVTy 标志位来检测 PWMx 跳闸区中断的来源。在使用多个中断源生成 PWMx 跳闸区中断时，可以读取和清除这些标志来识别和处理引发中断的具体事件。

图 108 中断逻辑结构框图



31.5.8.3 操作要点

（1） TZ1n 至 TZ6n 信号（TZxn 信号）为低电平有效输入信号

（2） 发生以下情况表明发生了跳闸事件：

- TZxn 信号中任何一个信号变低时
- 基于 TZDCSEL 寄存器选择的 DCxEy 强制事件发生时

（3） 每个 PWM 模块可以单独配置为使用或忽略每个跳闸区信号或数字比较事件。由 TZSEL 寄存器决定使用的跳闸区信号或数字比较事件

（4） TZxn 信号可以选择是否与 PWMCLK 同步，并在 GPIO 复用器内进行数字滤波

（5） 为了触发 PWM 模块的故障条件，需要在 TZxn 输入最小 3*TBCLK 的低电平脉冲宽度

（6） 当脉冲宽度小于此值时，OST 或 CBC 锁存器无法锁存跳闸条件

（7） 异步跳闸：即使由于任何原因时钟缺失，仍然可以确保 TZxn 输入上存在的有效事件跳闸

（8） GPIOs 或外设必须正确配置，详情请参见系统控制和中断章节。

（9） 配置选项如下，这些配置选项由 TZSEL 寄存器的 CBCx、OSHTx 和 DCxEVTy 位决定：

- 可以单独配置每个 TZxn 输入，可以为 PWM 模块提供 OST 或 CBC 事件
- DCxE1 事件可以配置为：
  - 直接使 PWM 模块跳闸
  - 为该模块提供 OST 事件
- DCxE2 事件可以配置为：
  - 直接使 PWM 模块跳闸
  - 为该模块提供 CBC 事件

单次跳闸（OST）事件

当发生一个 OST 事件时，立即对 PWMxA 和 PWMxB 输出执行 TZA 和 TZB 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 TZCTL2 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生 OST 事件，则 OST 位置位，并在 TZEINT 寄存器和 NVIC 外设中使能了中断时，还会生成 PWMx 跳闸区中断。在 TZCBCFLG 寄存器中，引起 OST 事件的那个事件对应标志也被置位。可以通过手动写入 TZOSTCLR 寄存器中的对应位，从而清零 TZOSTFLG 寄存器位（如果需要）。

当使能 OST 中断（OST=1）且通过 TZSEL 寄存器选择单次跳闸源为 DCxEVT1 时，无需使能 DCxEVT1 中断（DCAEVT1 /DCBEVT1=1），因为数字比较事件通过 OST 机制触发中断。

注意：在确认 OST 跳闸源变为非活动状态后，应清零 TZFLG 和 TZOSTFLG 寄存器标志。否则，当使能中断时，可能发生 OST 中断而 OST 标志为 0，这取决于何时清除标志。

当跳闸事件发生时采取的动作，可以通过 TZCTL、TZCTL2、TZCTLDCA 和 TZCTLDCB 寄存器分别独立地配置每个 PWM 输出引脚。可以采取的动作如下所示：

表格 141 跳闸事件发生时采取的动作

TZCTL 寄存器位域设置	PWMxA 和/或 PWMxB 动作	状态
00	高阻态	已跳闸
01	强制输出为高电平	已跳闸
10	强制输出为低电平	已跳闸
11	输出保持原状态	无操作

逐周期跳闸（CBC）事件

当发生一个 CBC 事件时，立即对 PWMxA 和 PWMxB 输出执行 TZA 和 TZB 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 TZCTL2 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生 CBC 事件，则 TZFLG[CBC]位置位，并在 TZEINT 寄存器和 NVIC 外设中使能了中断时，还会生成 PWMx 跳闸区中断。在 TZCBCFLG 寄存器中，引起 CBC 事件的那个事件对应标志也被置位。

当使能 CBC 中断（TZEINT[CBC]=1）且通过 TZSEL 寄存器选择逐周期跳闸源为 DCxEVT2 时，无需使能 DCxEVT2 中断（DCAEVT2 / DCBEVT2=1），因为数字比较事件通过 CBC 机制触发中断。

当不存在跳闸事件时，将会根据 TZCLR 寄存器的选择自动清除 CBC 跳闸锁存的状态。因此，在这种模式下，每个 PWM 周期都复位或清除跳闸事件。在通过手动写入 TZCBCCLR 寄存器标志位和 TZCLR[CBC]位清零之前，TZCBCFLG 寄存器标志位和 TZCLR[CBC]位会保持置位。如果清除 TZCBCFLG 寄存器标志位和 TZCLR[CBC]位后 CBC 事件仍然存在跳闸事件，则立即再次置位这些标志位。

数字比较事件（DCxEVTy）

基于 TZDCSEL 寄存器选择的 DCxH 和 DCxL 信号组合，可以产生数字比较事件 DCxEVTy。通过 DCTRIPSEL 寄存器选择 DCxH 和 DCxL 信号的源信号为跳闸区输入引脚或 COMPx 信号。DC 子模块信号详情请参见数字比较子模块功能描述章节。

当发生 DCxEVTy 时，立即对 PWMxA 和 PWMxB 输出执行 DCxEVTy 位域中指定的操作，详情请参见跳闸事件发生时采取的动作表格。当计数器向上计数或向下计数时，通过配置 TZCTLDCA 和 TZCTLDCB 寄存器相应位域指定发生跳闸事件时的独立动作，这些动作仅在 ETZE=1 的情况下才会生效。

如果发生数字比较跳闸事件，则 DCxEVTy 位置位，并在 TZEINT 寄存器和 NVIC 外设中使能了中断时，还会生成 PWMx 跳闸区中断。

当不存在数字比较跳闸事件时，引脚上的指定状态将被自动清除。在通过手动写入 DCAEVTy 位或 DCBEVTy 位清零之前，TZFLG[DCAEVTy]位和 TZFLG[DCBEVTy]位会保持置位。如果清除 TZFLG[DCAEVTy]位和

TZFLG[DCBEVTy]位后 CBC 事件仍然存在跳闸事件，则立即再次置位这些标志位。

跳闸区配置

表格 142 跳闸区配置选项

条件	PWM1 寄存器配置	PWM2 寄存器配置
TZ1n 上的 OST 事件将 PWM1A 和 PWM1B 输出都拉低，并强制 PWM2A 和 PWM2B 输出为高电平	（1）OSHT1=1：选择 TZ1n 作为 PWM1 单次跳闸源 （2）TZA=10：跳闸事件发生时，强制 PWM1A 输出为低电平 （3）TZB=10：跳闸事件发生时，强制 PWM1B 输出为低电平	（1）OSHT1=1：选择 TZ1n 作为 PWM2 单次跳闸源 （2）TZA=01：跳闸事件发生时，强制 PWM2A 输出为高电平 （3）TZB=01：跳闸事件发生时，强制 PWM2B 输出为高电平
（1）TZ5n 上的 CBC 事件将 PWM1A 和 PWM1B 输出都拉低 （2）TZ1n 或 TZ6n 上的 OST 事件将 PWM2A 置为高阻态	（1）CBC5 =1：选择 TZ5n 作为 PWM1 逐周期跳闸源 （2）TZA=10：跳闸事件发生时，强制 PWM1A 输出为低电平 （3）TZB=10：跳闸事件发生时，强制 PWM1B 输出为低电平	（1）OSHT1=1：选择 TZ1n 作为 PWM2 单次跳闸源 （2）OSHT6=1：选择 TZ6n 作为 PWM2 单次跳闸源 （3）TZA=00：跳闸事件发生时，PWM2A 被置为高阻态 （4）TZB=11：跳闸事件发生时，PWM2B 保持原状态，即忽略跳闸事件

当配置 PWM X-BAR/输入 X-BAR 和 GPIO 选项时，需要注意更改 X-BAR 输入选择可能会引起意外事件发送。建议将输入 X-BAR 和 GPIO 配置在使能 PWM 跳闸区之前设置完成。当必须要在使能 PWM 跳闸区后更改输入 X-BAR/GPIO 配置时，用户可以关闭跳闸（通过清零 TZSEL 寄存器），并在输入 X-BAR 选择更改后重新配置 TZSEL 寄存器。

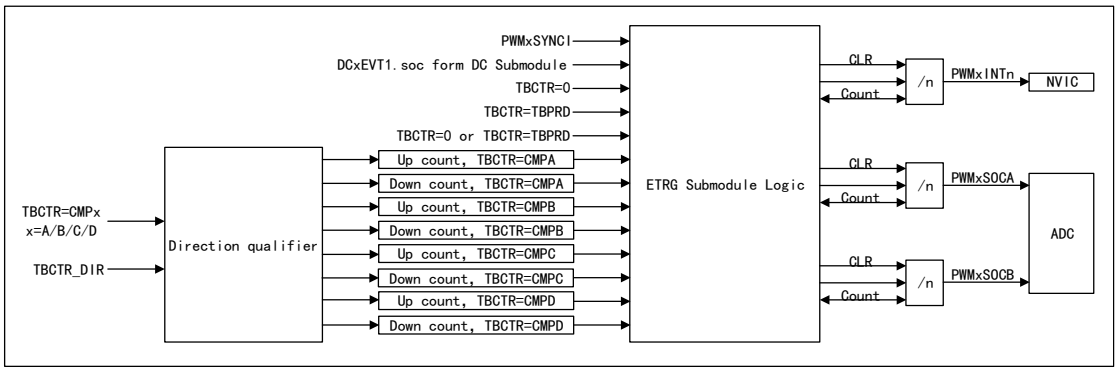
31.5.9 事件触发子模块功能描述

事件触发子模块负责管理由时基子模块、计数器比较子模块和数字比较子模块产生的事件，当选定事件发生时，它将会产生一个中断信号给 CPU，并向 ADC 发送一个转换开始脉冲。

31.5.9.1 结构框图

如下图所示，ETRG 子模块可以监视各种事件条件，根据预分频设置决定在每次事件、每两次事件直到每十五次事件发出中断请求和 ADCSOC。

图 109 事件触发子模块结构框图



表格 143 ETRG 子模块补充说明

寄存器	描述
ETSEL	选择触发中断或 ADC 转换开始的事件
ETPS	配置上面提到的事件分频选项
ETFLG	该寄存器表示所选事件和预缩放事件的状态
ETCLR	该寄存器用于以软件方式清零 ETFLG 寄存器中的标志位
ETFRC	这些位用于软件强制产生事件，用于调试或软件干预
ETINTPS	配置中断事件预分频选项，支持计数多达 15 个事件的周期
ETSOCPS	配置 SOC 事件预分频选项，支持计数多达 15 个事件的周期
ETCNTINITCTL	该寄存器使能使用同步事件或使用软件强制使能 ETCNTINIT 寄存器初始化
ETCNTINIT	在同步事件或软件强制时，该寄存器允许用户使用编程值对中断、SOCA 或 SOCB 计数器进行初始化

31.5.9.2 事件触发的脉冲发生器

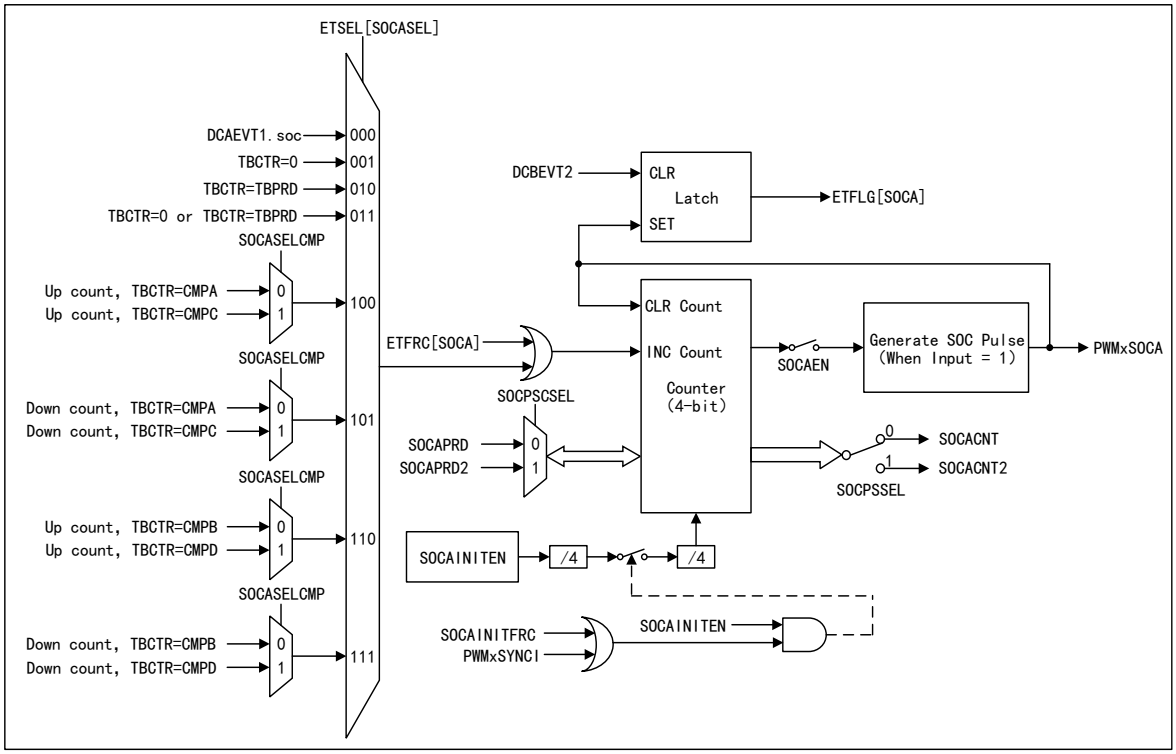
SOCA 脉冲触发器

事件触发 SOCA 脉冲触发器的操作如图所示，其工作方式如下所示：

- 增强功能：包括 SOCASELCMP 和 SOCBSELCMP 位分别使能 CMPC 和 CMPD 事件，从而引起转换的开始
- SOCPSSEL 位可以决定 SOCACNT2 位和 SOCAPRD2 位是否接管控制权
- SOCA 事件计数器和 SOCA 周期的行为类似于中断发生器，但是会连续产生脉冲。也就是说，在产生脉冲时，锁存 ETFLG[SOCA]位，但中断发生器不会中止后续脉冲产生
- SOCAEN 位使能/禁止产生脉冲，除非达到周期值，否则仍然计数输入事件
- 通过 SOCASEL 和 SOCBSEL 位域可以单独配置触发 SOCA 和 SOCB 脉冲的事件，其可以使用事件如下：
  - 与中断产生逻辑指定的事件相同
  - 来自 DC 子模块的 DCAEVT1.soc 和 DCBEVT1.soc 事件信号

- SOCA 事件计数器 2 初始化方案类似于中断生成器，都包括相应的使能、值初始化以及同步事件或软件强制选项

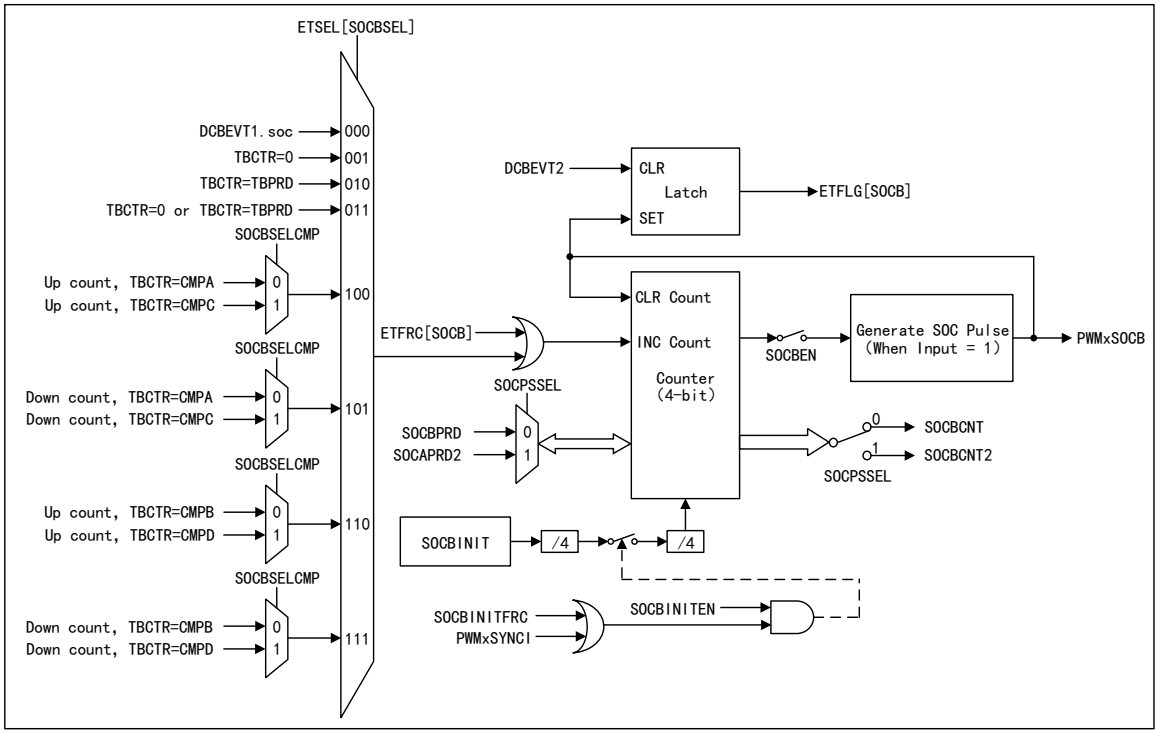
图 110 SOCA 脉冲触发器结构框图



SOCB 脉冲触发器

事件触发 SOCB 脉冲触发器的操作如图所示，其工作方式详情请参见 SOCA 脉冲触发器小节。

图 111 SOCB 脉冲触发器结构框图

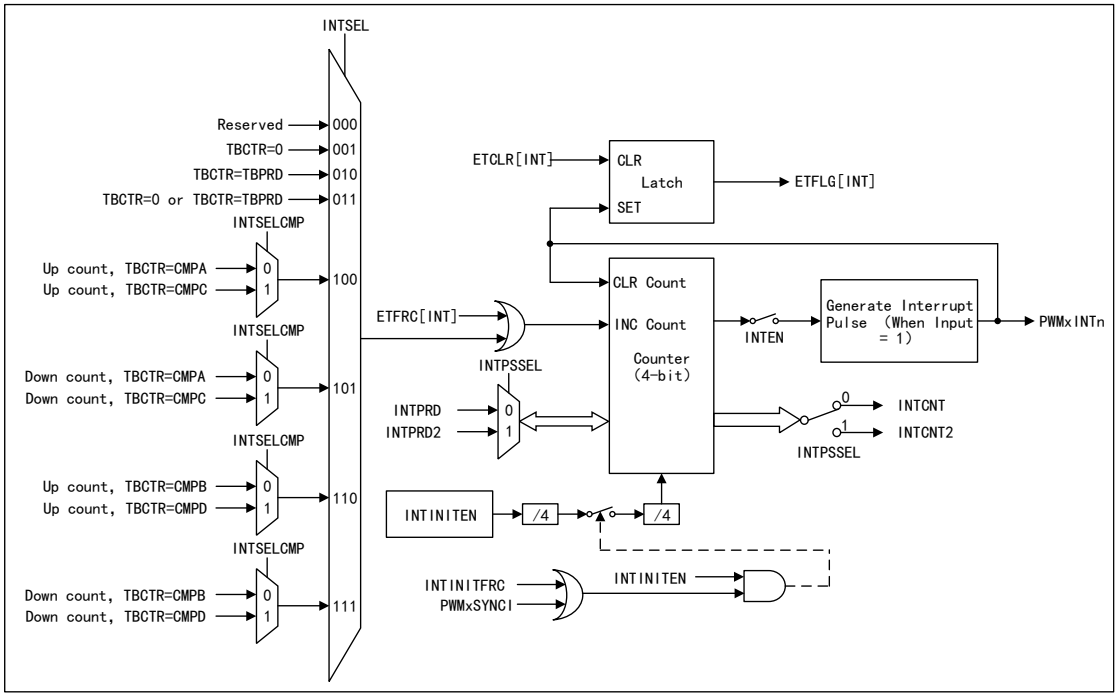


31.5.9.3 中断

事件触发的中断生成逻辑如下图所示，通过 INTPRD 位域选择产生中断脉冲所需的事件数为：不产生中断和每次/每两次/每三次事件产生中断。

INTPSSEL 位决定是否使用 ETINTPS 寄存器的 INTCNT2 位和 INTPRD2 位来确定事件频率为每 0...15 个事件产生一次中断。通过 INTSEL 位域和 INTSELCMP 位配置可以导致中断的事件如下表格所示。

图 112 中断发生器结构框图



表格 144 中断发生器补充说明

事件	描述
TBCTR=0	时基计数器等于 0
TBCTR=TBPRD	时基计数器等于时基周期
TBCTR=0 或 TBCTR=TBPRD	0 或等于时基周期
TBCTRU=CMPA	计数器向上计数时，时基计数器等于 CMPA
TBCTRD=CMPA	计数器向下计数时，时基计数器等于 CMPA
TBCTRU=CMPB	计数器向上计数时，时基计数器等于 CMPB
TBCTRD=CMPB	计数器向下计数时，时基计数器等于 CMPB
TBCTRU=CMPC	计数器向上计数时，时基计数器等于 CMPC
TBCTRD=CMPC	计数器向下计数时，时基计数器等于 CMPC
TBCTRU=CMPD	计数器向上计数时，时基计数器等于 CMPD
TBCTRD=CMPD	计数器向下计数时，时基计数器等于 CMPD

基于 INTPSSEL 位所做的选择，已经发生的事件数可以通过 INTCNT 位域或 INTCNT2 位域来读取。即当指定的事件发生时，基于 INTPSSEL 位所做的选择，INTCNT 位域或 INTCNT2 位域会不断增加直到达到 INTPRD 或 INTPRD2 指定的值。如果 INTCNT=INTPRD，则计数器停止计数并产生计数器对应的输出（中断）。计数器只有在中断发送到 NVIC 时才会清零。

当 INTCNT 达到 INTPRD 或 INTCNT2 达到 INTPRD2 时，计数器会发生的行为如表格所示。

表格 145 计数器的行为

条件	行为
INTEN=0 或 INT=1	当计数器到达周期值（INTCNT=INTPRD）时，计数器停止对事件的计数。
INTEN=1 且 INT=0	产生一个中断脉冲且 INT=1，INTCNT 位域清零。事件计数器再次开始计数事件。
INTEN=1 且 INT=1	计数器将保持输出高电平，直到清除 INT 位。这允许在一个中断被处理时，下一个中断处于等待状态。

以下方式适用于 INTCNT/INTCNT2 和 INTPRD/INTPRD2：

- 向 INTPRD 位域写入 0 会自动清零计数器且重置计数器输出，因此不会产生中断
- 所有写入 INTPRD 位域的操作会使 INTCNT 保留原来的值
- INTCNT 在溢出时复位
- ETFRC[INT]=1 会使事件计数器增加
- 当计数器到达周期值时，计数器的行为与表格描述的相同
- 如果 INTPRD=00，则禁用计数器且忽略 ETFRC[INT]位，因此不会检测到事件

当使用 INTCNT 和 INTPRD 时，每次/每两次/每三次事件可以产生一次中断。当使用 INTCNT2 和 INTPRD2 时，每次事件至每 15 次事件（最多）可以产生一次中断。

可以基于 INTINITEN 位的选择，通过 INTINIT 位域中的值对 INTCNT2 进行初始化。当 INTINITEN 置位时，在发生同步事件或 INTINITFRC 位域决定的软件强制事件时，使用 INTINIT 位域的内容对 INTCNT2 进行初始化。

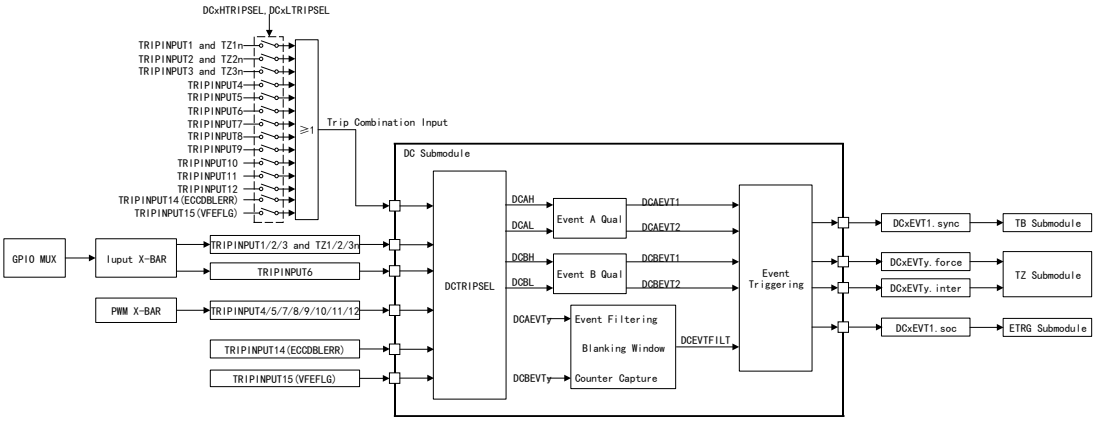
31.5.10 数字比较子模块功能描述

在 PWM 系统内，数字比较子模块与其他子模块交互的信号如结构框图所示。

该模块用于将外部的信号（如 COMPx 信号）与 PWM 模块进行比较，直接产生 PWM 事件/动作并反馈给 TB、TZ 和 ETRG 子模块。该模块支持用于滤除 DC 事件信号中的噪声或不需要的脉冲的空白窗口功能。

31.5.10.1 结构框图

图 113 数字比较子模块结构框图

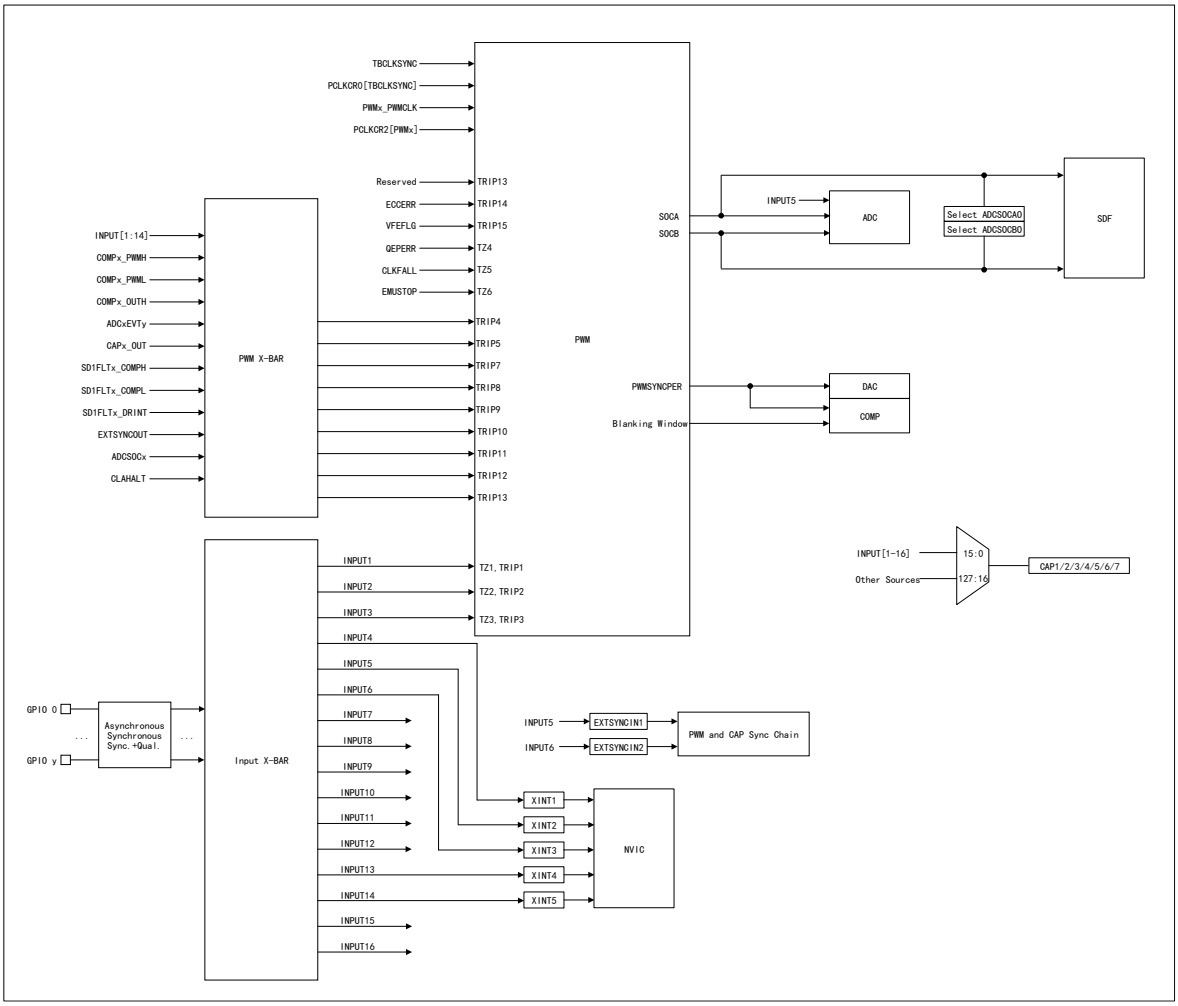


31.5.10.2 输入与输出

如下图所示，CAP 输入信号源自输入 X-BAR 信号。在该器件上，任何 GPIO 引脚都可以映射为 TZ 子模块和 DC 子模块的跳闸输入和/或跳闸区输入。通过 INPUTxSELECT 寄存器可以决定选择 GPIO 引脚为跳闸区输入/跳闸输入。

注意：为了避免误锁存跳闸信号，用户有责任在为相应的 PWM 外设使能时钟和配置跳闸输入之前，将所选引脚驱动到正确状态。

图 114 GPIO 复用器到跳闸输入结构框图



31.5.10.3 使用 COMP 跳闸 PWM

增强型跳闸动作

为了允许多个 COMP 能够同一时间影响 DCxEVTy 事件和跳闸动作，通过或逻辑电路将来自 PWM 模块外部的全部跳闸输入组合在一起，并通过 DCTRIPSEL 寄存器选择作为跳闸输入组合输入到 DCxH 和 DCxL。

可以通过 DCxHTRIPSEL 和 DCxLTRIPSEL 寄存器进行选择将哪些跳闸输入放入该组合逻辑电路，以生成 DCxH 和 DCxL 信号。选择作为跳闸输入组合的输入将会传递到 DCTRIPSEL 寄存器中。

逐周期方式关闭 PWM

当使用 COMP 以逐周期方式跳闸 PWM 时，为了防止在一个 PWM 周期中断言的比较器跳闸状态延续到下一个周期，应该采取一些措施来阻止。在特定条件下，为了防止在 PWM 周期结束时发生的延迟或持续跳闸事件延续到下一个 PWM 周期，需要采取一些预防措施。比较器可以给处于下游的 PWM 模块发送跳闸状态信号。对于峰值电流模式控制等应用，期望每个 PWM 周期只有一次跳闸事件。

当比较器数字数字滤波器或 PWM DC 子模块中的任何一个来定性比较器跳闸信

号，在 PWM 跳闸逻辑电路能够响应跳闸信号的逻辑电平变化之前，将引入 N 个时钟周期用于定性。如果定性 PWM 跳闸状态，则该跳闸状态将在比较器跳闸信号失效后的 N 个时钟周期内保持有效。当一个定性的比较器跳闸信号在一个 PWM 周期结束前的 N 个时钟周期内保持断言时，跳闸条件在下一个 PWM 周期开始后才会清除。因此，在新的 PWM 周期开始时立即检测到跳闸条件。

应用程序可以采取以下三种方式来确保在每个 PWM 周期结束之前，PWM 跳闸逻辑所看到的定性跳闸信号失效。并以此来避免上述期望之外的跳闸状态。

表格 146 解决方式

序号	解决方式
1	将系统设计为：在 PWM 周期结束前的 N 个时钟周期内，比较器跳闸不会断言。
2	在 PWMSYNC 信号之前至少两个时钟周期时，通过事件滤波器来激活比较器跳闸信号的空白，并继续空白至少跨越到下个 PWM 周期的 N 个时钟周期。
3	当 COMP 通过 COMPxLATCH 路径时，需要在 PWM 周期结束前至少 N 个时钟周期清除 COMPxLATCH 位。该锁存器可以通过产生提前的 PWMSYNC 信号或通过 COMPSTSCLR 寄存器来清除。该器件上的 PWM 模块可以通过 HRPCTL 寄存器进行 CMPC 或 CMPD 匹配生成 PWMSYNC 信号，用于在 PWM 周期内任意位置放置该信号。

31.5.10.4 操作要点

数字比较事件

如 DC 子模块功能特征所示，可以通过 DCTRIPSEL 寄存器选择模拟比较器模块的 COMPx 信号和跳闸输入 TZ1n 至 TZ3n 来产生 DCxH 和 DCxL 信号。使用 TZDCSEL 寄存器限定对所选 DCxH 和 DCxL 信号的动作，从而产生 DCxEVTy 事件。

可以不过滤 DCxEVTy 事件或可以通过对 DCxEVTy 事件进行过滤来提供经过过滤的事件信号 DCEVTFILT。过滤详情请参见事件过滤小节。DCxEVTy 事件信号或经过滤波的 DCEVTFILT 信号都可以产生对 PWM 同步信号、ADCSOC、TZ 子模块或 TZ 中断的强制事件。

当 TZxn 信号用于数字比较事件跳闸功能时，将该信号视为正常的输入信号且可以定义为低电平有效或高电平有效输入。在 TZxn 信号或者 DCxEVTy 强制信号中任一信号有效时，PWM 输出被异步跳闸。为了保持锁存状态，需要至少 3*TBCLK 同步脉冲宽度。如果脉冲宽度小于 3*TBCLK 同步脉冲宽度，则跳闸状态可能会被 CBC 或 OST 锁存器锁存或不锁存。

表格 147 事件信号对其他模块的影响

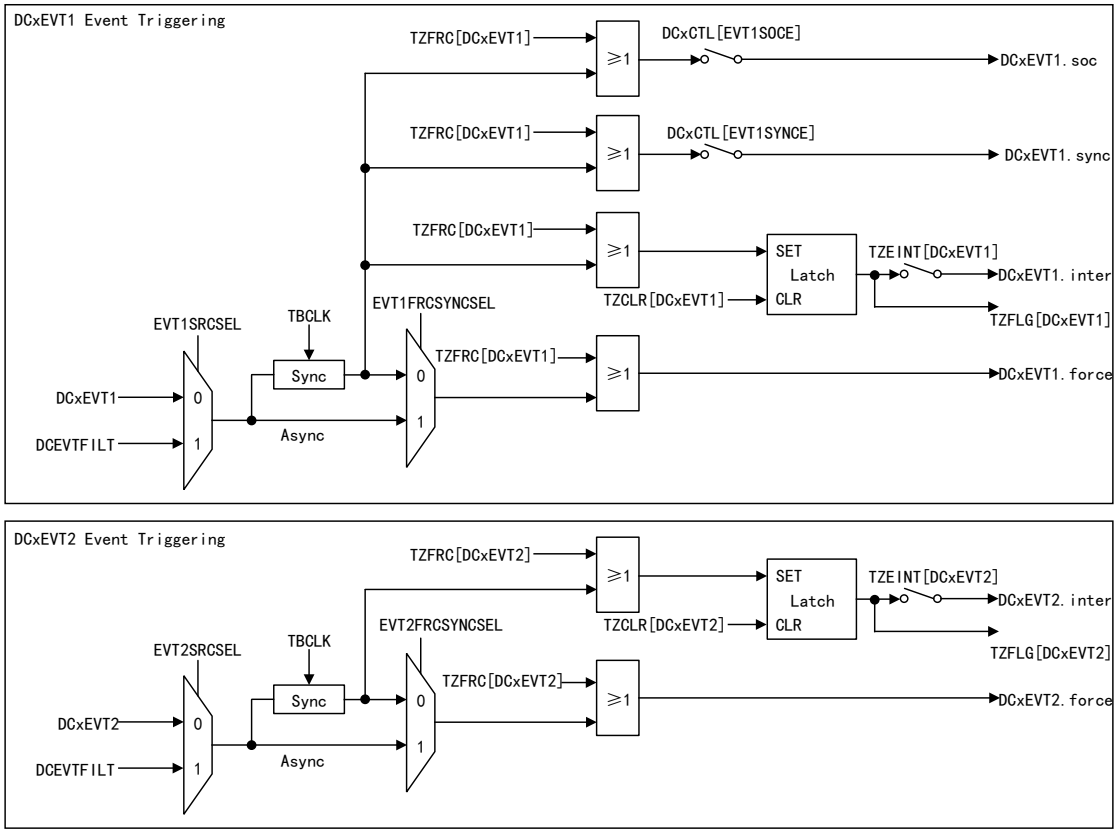
信号	描述
强制信号	通过 TZCTL，TZCTLDCA 和 TZCTLDCB 寄存器将 DCAEVTy 强制信号配置为强制跳闸区状态可以直接影响 PWMxA 引脚的输出或通过 TZSEL 寄存器将 DCBEy.force 信号配置为单次跳闸/逐周期跳闸源时可以使用 TZCTL 或 TZCTL2 寄存器配置来影响跳闸动作。 在 TZCTL，TZCTL2，TZCTLDCA 和 TZCTLDCB 寄存器中具有冲突操作时的优先级如 PWMxA/PWMxB 优先级冲突操作表格所示。

信号	描述
中断信号	DCAEVTy 中断信号生成给 NVIC 跳闸区中断。需要在 TZEINT 寄存器中设置 DCxEVTy 位才能使能中断。如果发生其中一个事件，则会触发 PWMx 跳闸区中断中断，必须置位 TZCLR 寄存器中的对应位来清除中断。
转换开始信号	DCxE1.soc 信号与 ETRG 子模块交互，可以使用 SOCASEL 位和 SOCBSEL 位选择生成 SOCA 和 SOCB 脉冲的事件。
同步信号	DCxE1 同步信号与 PWMx 同步输入信号和 SPFSW 位相或，从而生成同步脉冲给 TBCTR。

表格 148 PWMxA/PWMxB 优先级冲突操作

输出	优先级冲突操作
PWMxA	（1）优先级最高为 TZA，其次为 DCAEVT1，优先级最低为 DCAEVT2 （2）优先级最高为 TZAU，其次为 DCAEVT1U，优先级最低为 DCAEVT2U （3）优先级最高为 TZAD，其次为 DCAEVT1D，优先级最低为 DCAEVT2D
PWMxB	（1）优先级最高为 TZB，其次为 DCBEVT1，优先级最低为 DCBEVT2 （2）优先级最高为 TZBU，其次为 DCBEVT1U，优先级最低为 DCBEVT2U （3）优先级最高为 TZBD，其次为 DCBEVT1D，优先级最低为 DCBEVT2D

图 115 DCxEVTy 事件触发结构框图



事件过滤

（1） 空白控制逻辑

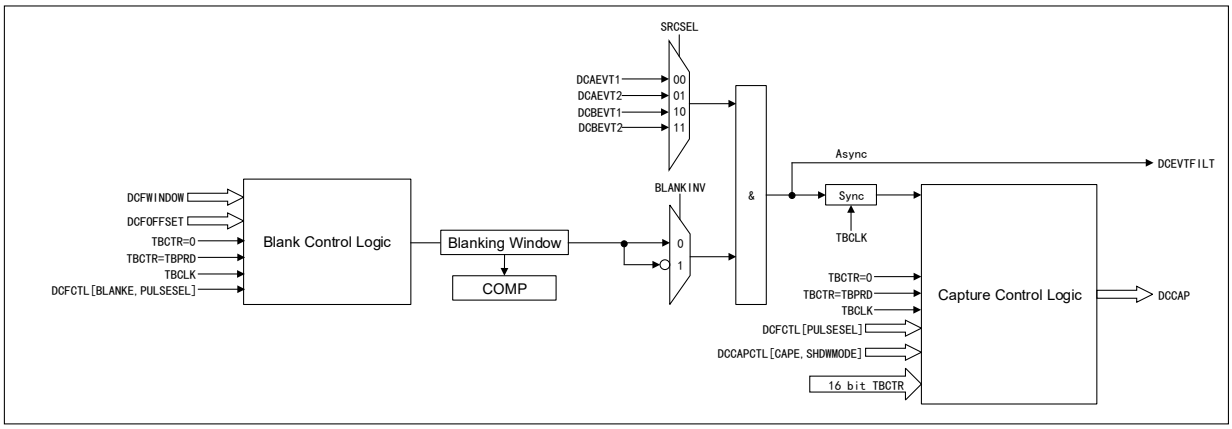
该逻辑用于定义空白窗口，在窗口处于活动状态时忽略信号上发生的所有事件。

通过 DCFCTL、DCFOFFSET 和 DCFWINDOW 寄存器来配置空白窗口，其中通过 BLANKE 位使能空白窗口，并按照 PULSESEL 位域的选择将空白窗口与所选择的脉冲同时对齐。

为了消除噪声，DCxEVTy 事件可以使用事件过滤逻辑电路来过滤，选择性地在一段时间内空白事件。该方式可以用于选择模拟比较器输出来触发 DCxEVTy 事件，且空白控制逻辑用于在跳闸 PWM 输出、产生中断或 ADCSOC 之前滤除信号上潜在的噪声。

通过 SRCSEL 位域选择滤波块信号源为 DCxEVTy 信号。将 TBCTR 计数偏移值编程到 DCFOFFSET 寄存器中，该寄存器确定空白窗口在 TBCTR=0 或 TBCTR=TBPRD 之后的开始点。空白窗口的持续时间，单位为偏移计数器到期后的 TBCLK 计数的数目，通过应用程序写入 DCFWINDOW 寄存器。在空白窗口结束之前和结束之后，事件仍然可以像以前一样生成强制、中断、同步和转换开始信号。

图 116 事件过滤结构框图



（2） 捕获控制逻辑

事件过滤还可以捕获基于 DCCAPCTL 寄存器所选 DCxEVTy 事件的计数值。

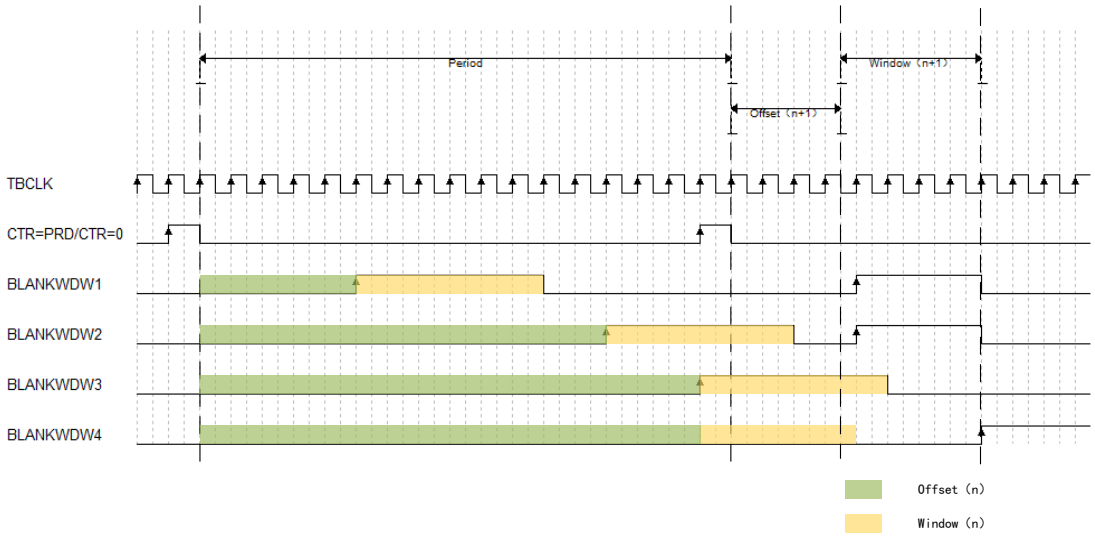
如果使能空白控制逻辑，则所选 DCxEVTy 事件触发将计数器捕获到活动寄存器中，CPU 从活动寄存器中读取。如果使能影子模式，则在 PULSESEL 位域所选的事件中将活动寄存器的值复制到影子寄存器，CPU 从影子寄存器中读取。

在所选的 DCxEVTy 事件之后直到 CAPMODE 位指定的事件发生之前，不会发生其他捕获事件。CAPMODE 可以通过以下方式进行配置：

- 通过 CAPCLR 位清除 CAPSTS 位之前，不会发生其他捕获事件
- 在 PULSESEL 位域所选的事件发生之前，不会发生其他捕获事件

在 PWM 周期内空白窗口和偏移的几种时序条件如下图所示。当空白窗口跨越 TBCTR=0 或 TBCTR=TBPRD 边界时，下一个窗口仍然从相同的偏移值处开始，即在 TBCTR=0 或 TBCTR=TBPRD 脉冲之后。

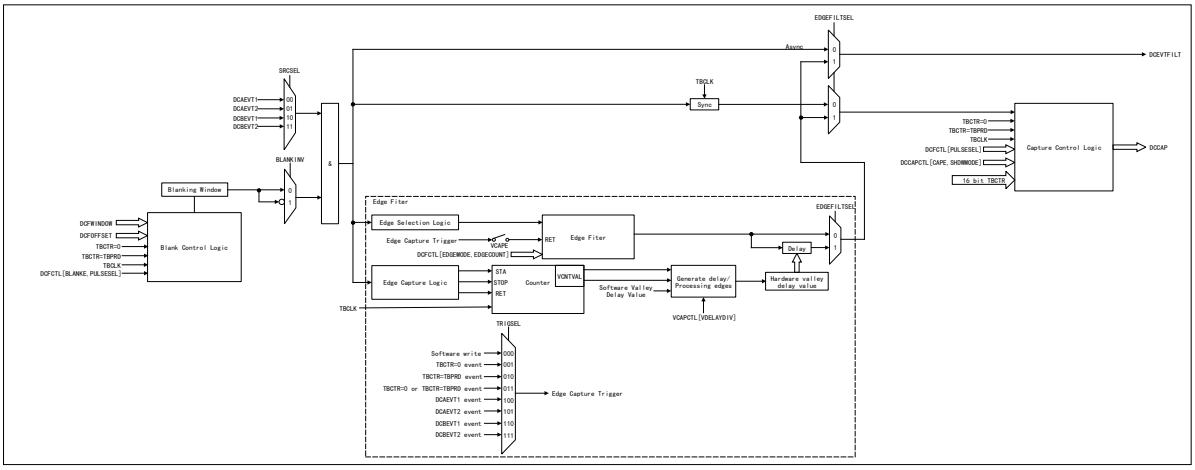
图 117 空白窗口时序图



谷底开关功能

事件过滤描述了事件过滤逻辑以及谷底开关功能，谷底开关功能可以在不增加任何外部电路的情况下实现可编程谷底开关。该功能提供一个片上硬件机制，支持捕获振荡周期，可以精确延迟 PWM 开关时刻，且在延迟生效前允许可编程的边沿数，此外，还提供多种触发器和事件选择以及易于适应变化的系统/操作条件下的最佳性能。

图 118 谷底开关结构框图



DCxEVTy 信号需要进一步处理以支持谷底开关，使能谷底开关功能操作步骤如下：

表格 149 使能谷底开关功能操作步骤

步骤	操作	描述
1	选择滤波块信号源	通过 SRCSEL 位域选择谷底开关的输入源，可以为此输入添加空白窗口（空白控制逻辑）。在这里，将谷底开关的输入选择为比较器输出或外部输入。

步骤	操作	描述
2	选择边沿模式和数量	通过 EDGEMODE 和 EDGECOUNT 位域配置边沿滤波器捕获上升沿、下降沿或双边沿的边沿数量。
3	选择事件重启边沿滤波器	根据 TRIGSEL 位域选择正确的事件来复位并重启沿滤波器，边沿捕获事件由选定的边沿触发或发送。
4	VCAPE=1	使能谷捕获逻辑
5	选择开始计数的边沿	通过 STARTEDGE 位域选择振荡周期测量捕获的开始边沿，即 16 位计数器开始计数
6	选择停止计数的边沿	通过 STOPEDGE 位域选择 16 位计数器停止计数的边沿。振荡周期信息由捕获的计数器值提供，且 STARTEDGE 值必须始终小于 STOPEDGE 值。
7	配置并应用捕获延迟	对边沿滤波后的 DCxEVTy 信号进行配置并应用捕获延迟。谷时基计数器的值可以直接应用、与用于偏移调整的软件编程值一起应用或只有一小部分延迟可以应用/不应用软件谷延迟值。这可以正确应用于谷值点的延迟。
8	配置边沿滤波器输出延迟	置位 EDGEFILTDLYSEL 位，根据上面捕获的值应用硬件延迟。

31.5.11 PWM X-BAR

31.5.11.1 PWM X-BAR 结构框图

该模块可以选择各种触发源，使其成为 8 个专用的 PWM 跳闸输入中的任意一个，即 TRIP4/5/7/8/9/10/11/12。

PWM X-BAR 结构框图请参见 X-BAR 标志章节。

注意：X-BAR 相关信息以及 X-BAR 标志位详情请参见 X-BAR 章节。

31.5.11.2 PWM X-BAR 单个输出结构框图

PWM X-BAR 有 8 个输出，路由到每个 PWM 模块。PWM X-BAR 单个输出结构框图请参见 PWM XBAR 章节，其输出与所有其他输出的架构相同。

首先确定传递到 PWM 的信号，详情请参见 X-BAR 章节的 Output X-BAR 复用配置表格。每个 TRIPx 输出的复用器（共 32 个复用器）至多选择一个信号。每个复用器的输入信号由 TRIPxMUX0TO15CFG 和 TRIPxMUX16TO31CFG 寄存器决定。需要使能 TRIPxMUXENABLE 寄存器中的复用器才能通过 PWM 传递任何信号。在传递到 PWM 上相应的 TRIPx 信号之前，所有使能的复用器都进行了逻辑相或。通过 TRIPOUTINV 寄存器选择 TRIPx 信号的极性。

注意："保留"的信号不可使用，所有的未使用和保留的信号都设定为 0。

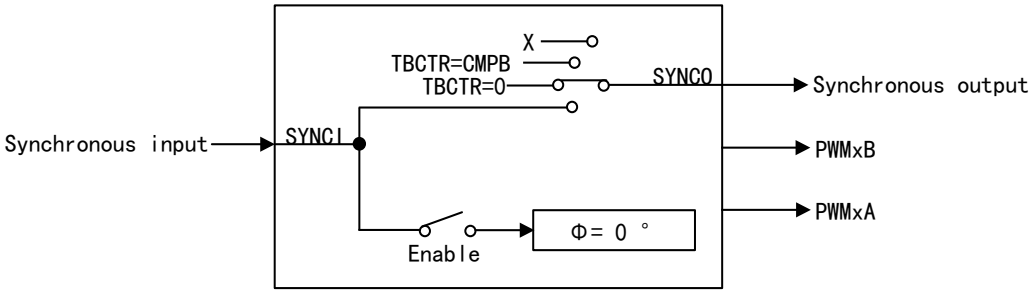
31.5.12 电源拓扑应用

PWM 模块具有所有必要的局部资源，从而可以作为一个完全独立的模块运行或与其他相同的 PWM 模块同步工作。

31.5.12.1 简化 PWM 模块结构框图

为便于理解在一个系统中协同工作的多个模块，简化 PWM 模块与多开关电源拓扑 PWM X-BAR 结构框图如下所示，该图只展示了多开关电源拓扑与多个 PWM 模块一起工作来控制所需的关键资源。

图 119 简化 PWM 模块结构框图



31.5.12.2 同步输入与同步输出配置

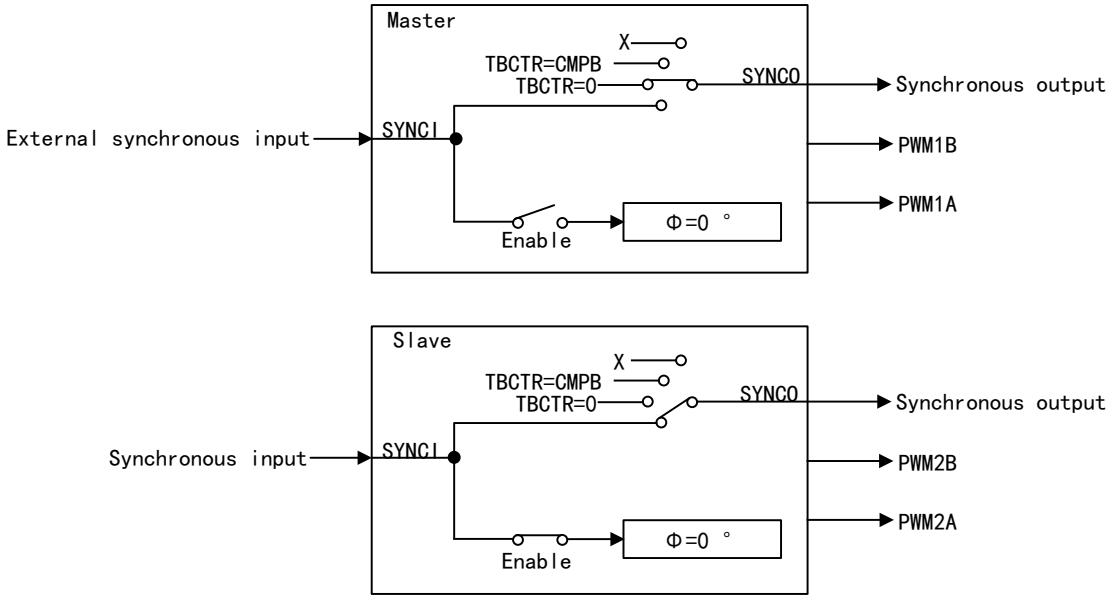
表格 150 同步输入与同步输出配置

选项	配置	描述
同步输入	使能开关闭合	在同步锁存信号到来时，将相位寄存器加载到自己的计数器
	使能开关断开	无任何操作，即忽略传入的同步锁存信号
	同步信号透传	将同步输出连接到同步输入
	同步输出连接到 TBCTR=TBPRD	在主模式下，在 PWM 边界提供同步信号
	同步输出连接到 TBCTR=CMPB	在主模式下，在可编程的任意时间点提供同步信号
	禁止同步输出	模块处于独立模式，不向其他模块提供同步信号
同步输出	同步信号透传	将同步输出连接到同步输入
	同步输出连接到 TBCTR=TBPRD	在主模式下，在 PWM 边界提供同步信号
	同步输出连接到 TBCTR=CMPB	在主模式下，在可编程的任意时间点提供同步信号
	禁止同步输出	模块处于独立模式，不向其他模块提供同步信号

模式组合

对于同步输出选项，模块可以通过使能开关来忽略或选择在同步输入选通输入时装载一个新的相位值。虽然有多种可能的组合，但主模式和从模式为最常见的两种模式，如下图所示。

图 120 PWM1 主模式和 PWM2 从模式配置图



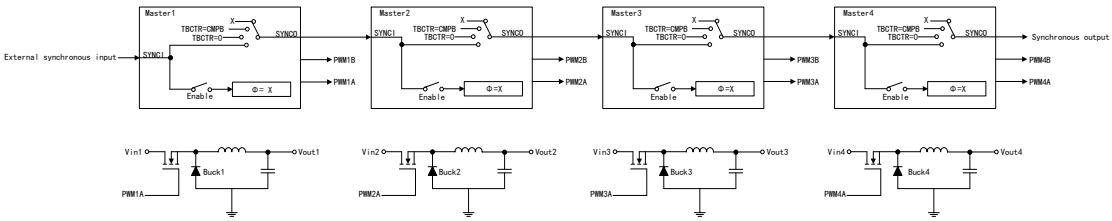
31.5.12.3 Buck 转换器

Buck 是最简单的功率变换器拓扑之一。当一个单独的 PWM 模块配置为一个主模块时，可以控制两个同频率的 Buck 级。

控制多个独立频率

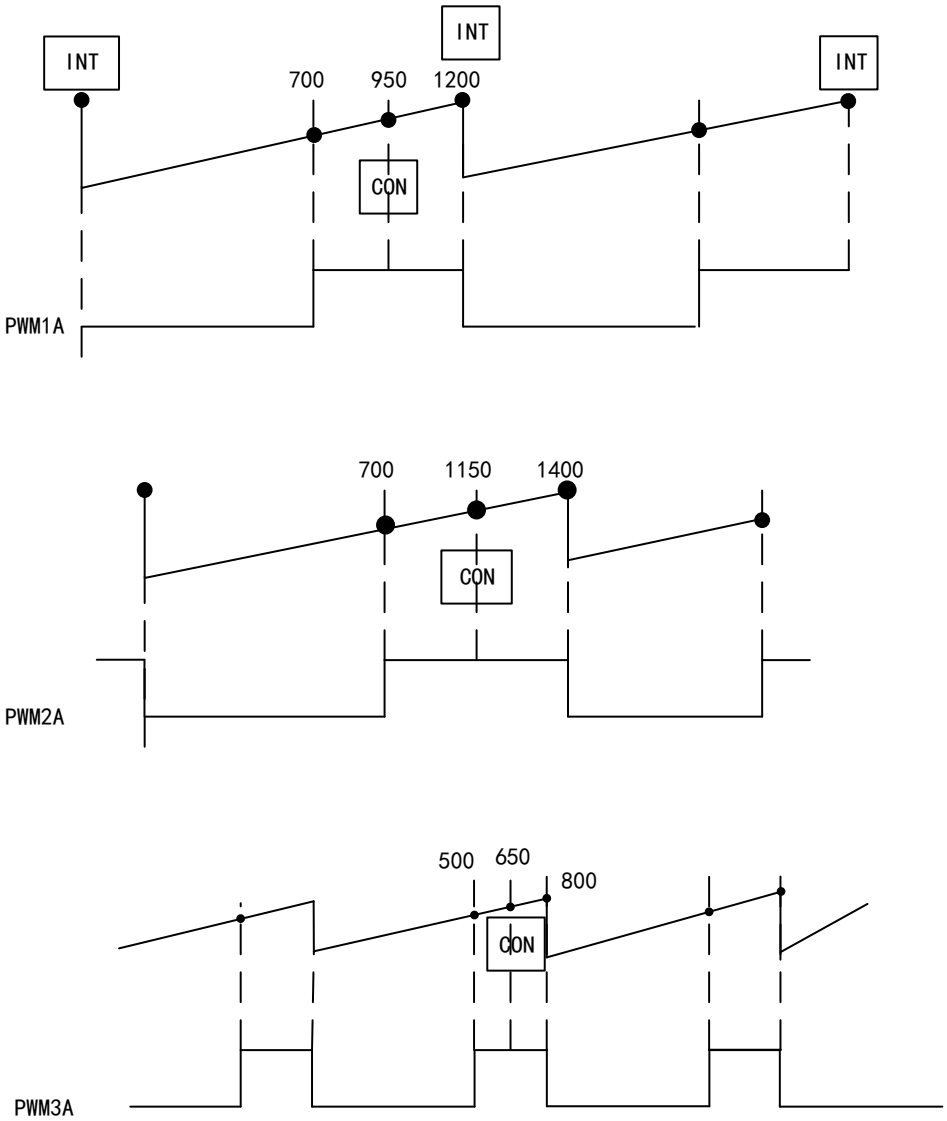
当每个 buck 变换器需要独立的频率控制时，每个变换器级必须分配一个 PWM 模块。，其中每级以独立的频率运行。在该情况下，4 个 PWM 模块都配置为主模块且不使用同步。

图 121 控制 4 个 Buck 级的结构框图（$F_{PWM1}≠F_{PWM2}≠F_{PWM3}≠F_{PWM4}$）



注意：如果 Φ=X，则可以不在乎时基相位偏移寄存器的值。

图 122 4 个 Buck 级波形图（只展示了三个波形）



注意：

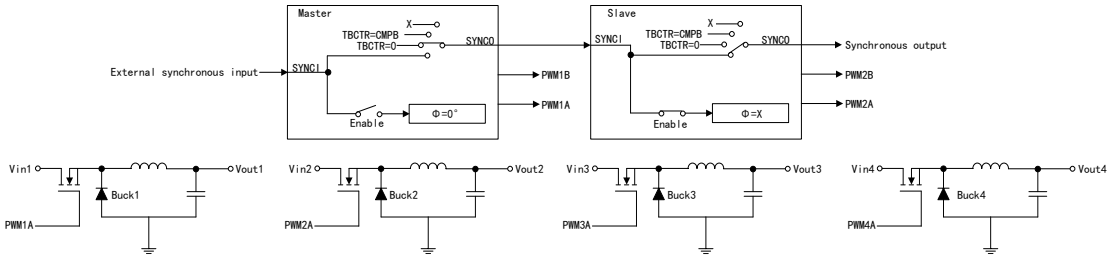
（1） INT 表示该事件触发中断，CON 表示该事件触发 ADC 开始转换。

（2） 其他未说明点为开始 ADC 转换。

控制多个相同频率

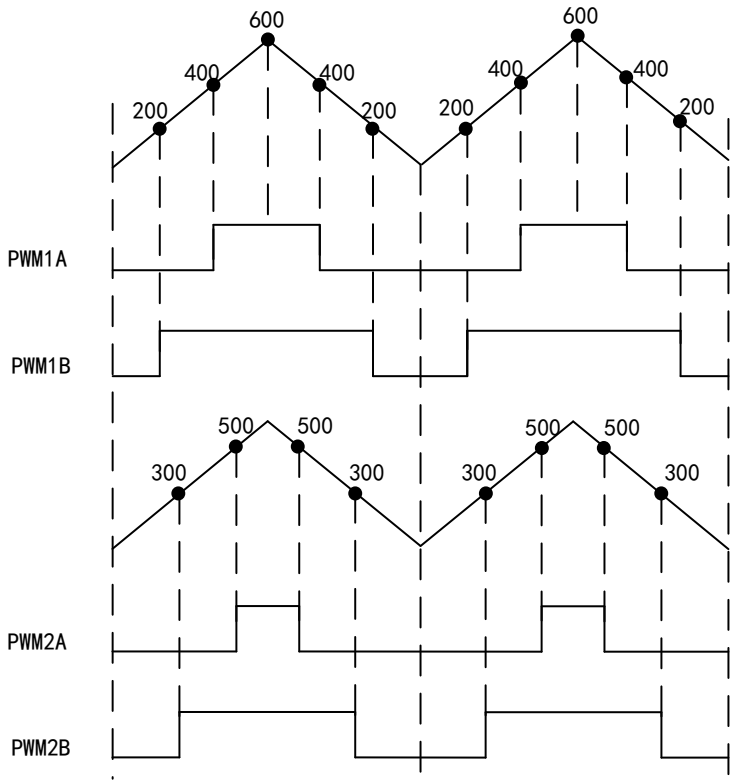
当需要同步时，将 PWM2 配置为从模块，并以 PWM1 的 N（整数）倍频率运行，即 $F_{PWM2}=N * F_{PWM1}$，其 4 个 buck 级结构框图以及波形图如下所示。从主模式到从模式的同步信号确保这些模块保持锁定。

图 123 控制 4 个 Buck 级的结构框图



注意：如果 Φ=X，则可以不在乎时基相位偏移寄存器的值。

图 124 控制四个降压级的降压波形（$F_{PWM1}=F_{PWM2}$）



注意：每个圆点产生一个 ADC 转换。

控制峰值电流控制模式 Buck 转换器

峰值电流控制技术可以用于自动过电流限制，快速校正输入电压变化和降低磁饱和。PWM1A 与片上模拟比较器的 buck 变换器拓扑结构框图以及波形图如下所示。通过电流检测电阻检测输出电流并馈送到片上比较器的同相端。内部可编程的 12 位 DAC 可以将该输入连接到外部参考或可以提供比较器反相端的参考峰值电流。比较器的输出用作 DC 子模块的输入。如果检测到的电流达到参考峰值，则所述 PWM 模块配置为跳闸 PWM1A 输出。使用了逐周期式的跳闸机制。

图 125 峰值电流控制模式 Buck 转换器结构框图

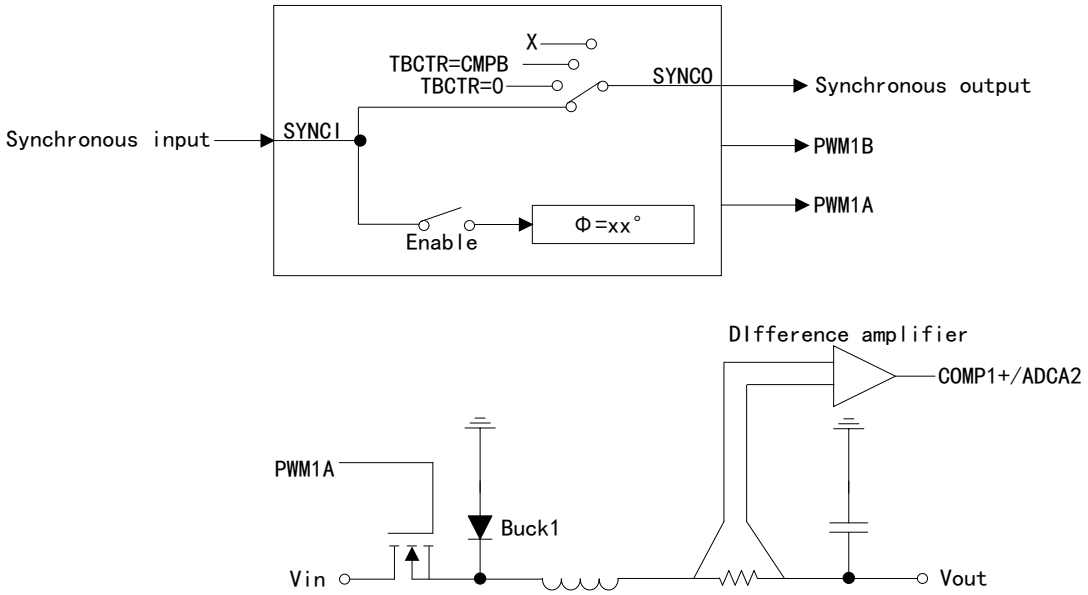
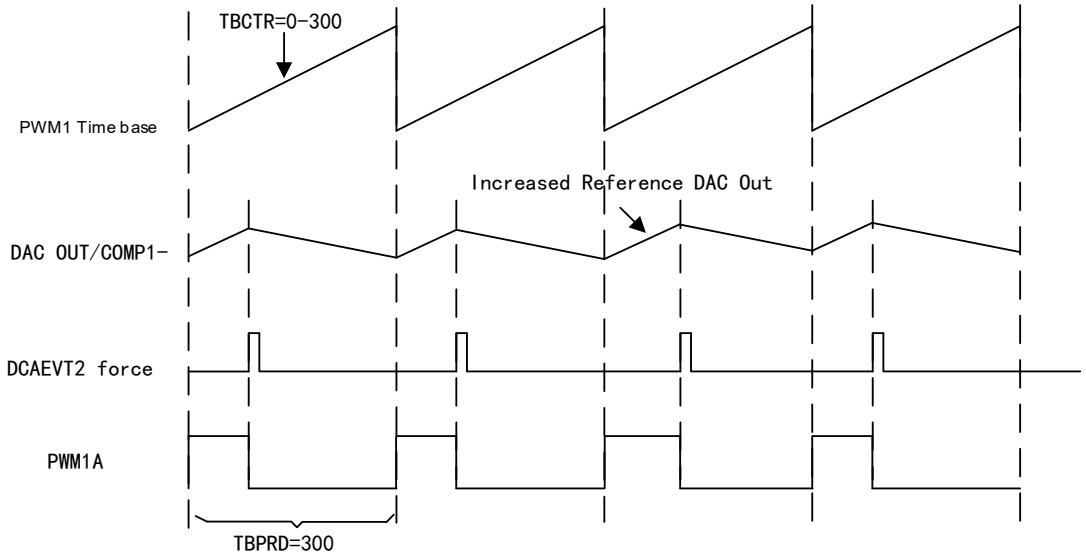


图 126 峰值电流控制模式 Buck 转换器波形图（只展示了三个波形）



31.5.12.4 半 H 桥转换器

相同的 PWM 模块可以用于控制多个开关元件的拓扑和使用单个 PWM 模块控制半 H 桥级，可以将这种控制扩展到多个级。控制 2 个同步半 H 桥的结构框图以及波形图如下所示，其中 PWM2 在 PWM1 的 N（整数）倍频率运行，即 $F_{PWM2}=N * F_{PWM1}$。

PWM2（从模式）配置为同步信号透传，该配置允许 PWM3 控制第 3 个半 H 桥且可以与 PWM1（主模式）保持同步。

图 127 控制 2 个同步半 H 桥的结构框图

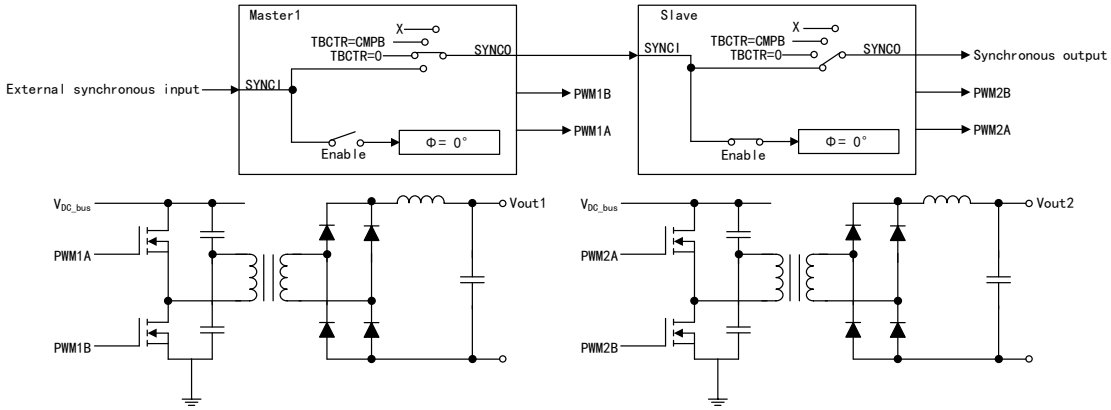
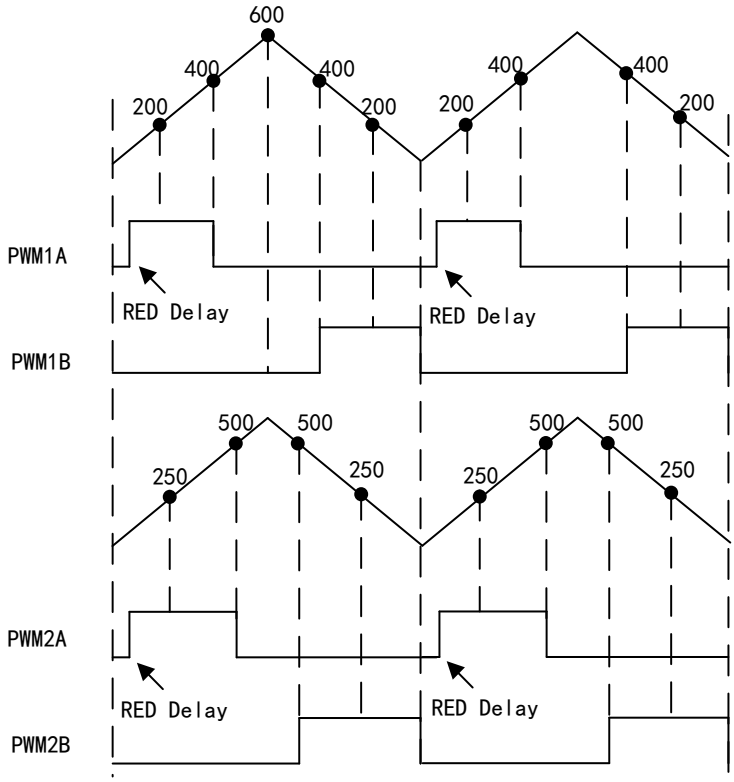


图 128 2 个同步半 H 桥波形图（F_{PWM1}=F_{PWM2}）



31.5.12.5 双三相逆变器

将多个模块控制单个功率级扩展三相逆变器的情况下，使用三个 PWM 模块控制六个开关元件，每个模块控制逆变器的一个桥臂。为了确保所有桥臂以相同的频率进行切换且所有桥臂保持同步，可以通过使用主从设置，即一个主模块和两个从模块实现。6 个 PWM 模块控制两个独立的三相逆变器（每个逆变器驱动一个电机）结构框图以及波形图如下所示。

如 Buck 转换器章节所示，可以选择以下方式运行逆变器：

- 将两个逆变器同步运行
  - PWM1 为主模块，其他模块为从模块
- 以不同频率运行每个逆变器

- PWM1 和 PWM4 为主模块，其中 FPWM1=FPWM2=FPWM3，FPWM4=FPWM5=FPWM6
- PWM4/5/6 在 PWM1/2/3 的 N（整数）倍频率运行，即 FPWM4/5/6 = N * FPWM1/2/3

图 129 双三相逆变器级结构框图

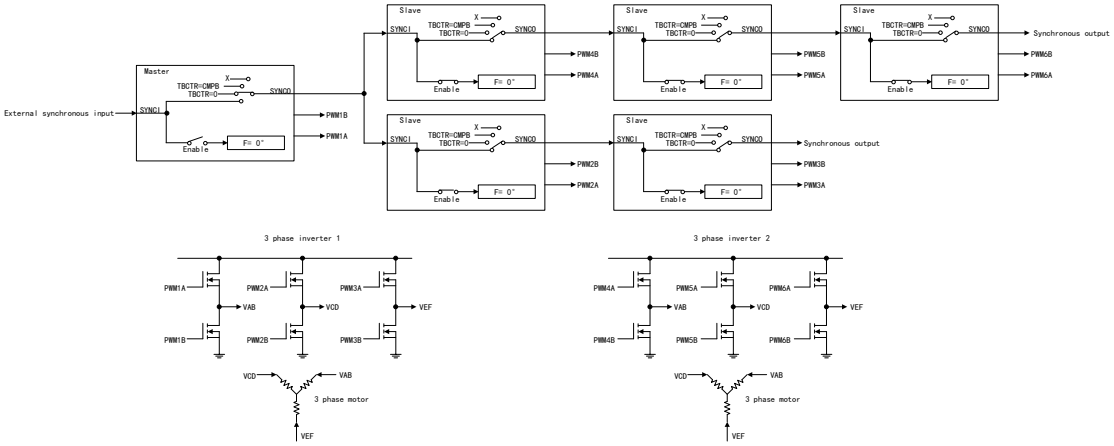
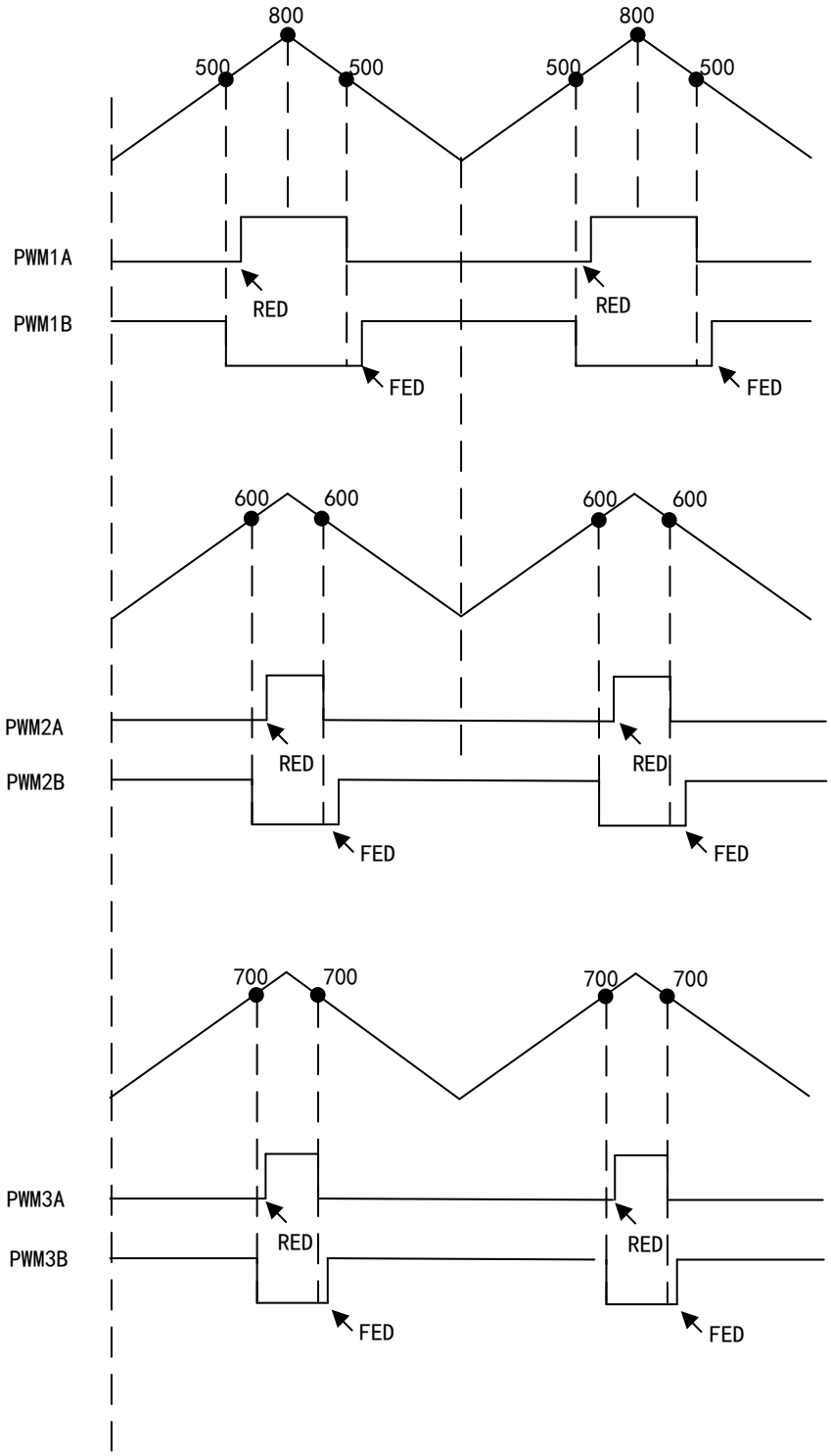


图 130 双三相逆变器的逆变波形图（只展示了一个逆变器）



31.5.12.6 三相交错 DC/DC 转换器

利用三个 PWM 模块控制三相交错 DC/DC 转换器的结构框图以及波形图如下所示，其中 PWM1 作为主模块，其他模块作为从模块。由于相邻模块之间的相位关系必须为 120°才能工作，这需要通过从模块的 TBPHS 寄存器设置为 TBPRD 的 1/3 和 2/3。

N 个相位的 TBPHS 值计算公式为：TBPHS（N，M）=（TBPRD/N）*（M-1），其中 N 为相位数，M 为 PWM 模块编号。该概念可以通过适当设置 TBPHS 值可以扩展到四相或更多相位。例如，如果 TBPRD=600 且 N=3，则 PWM2 的相位值 TBPHS（3，2）=200，PWM3 的相位值 TBPHS（3，3）=400，这两个从模块都同步到 PWM1 模块。

图 131 三相交错 DC/DC 转换器结构框图

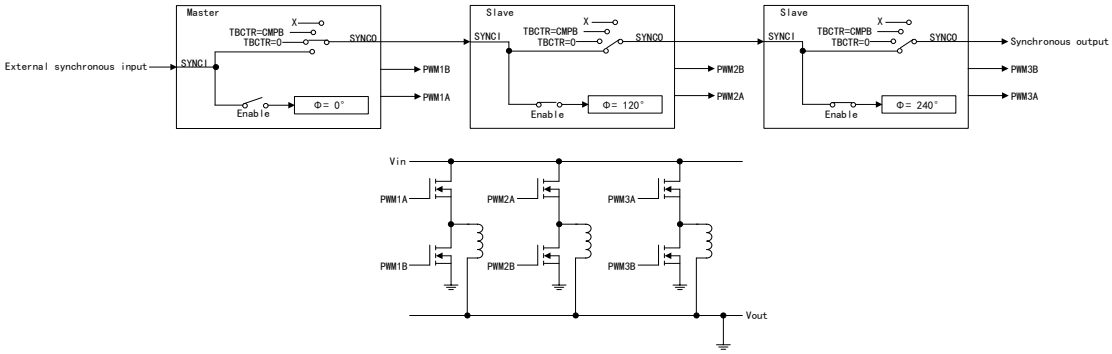
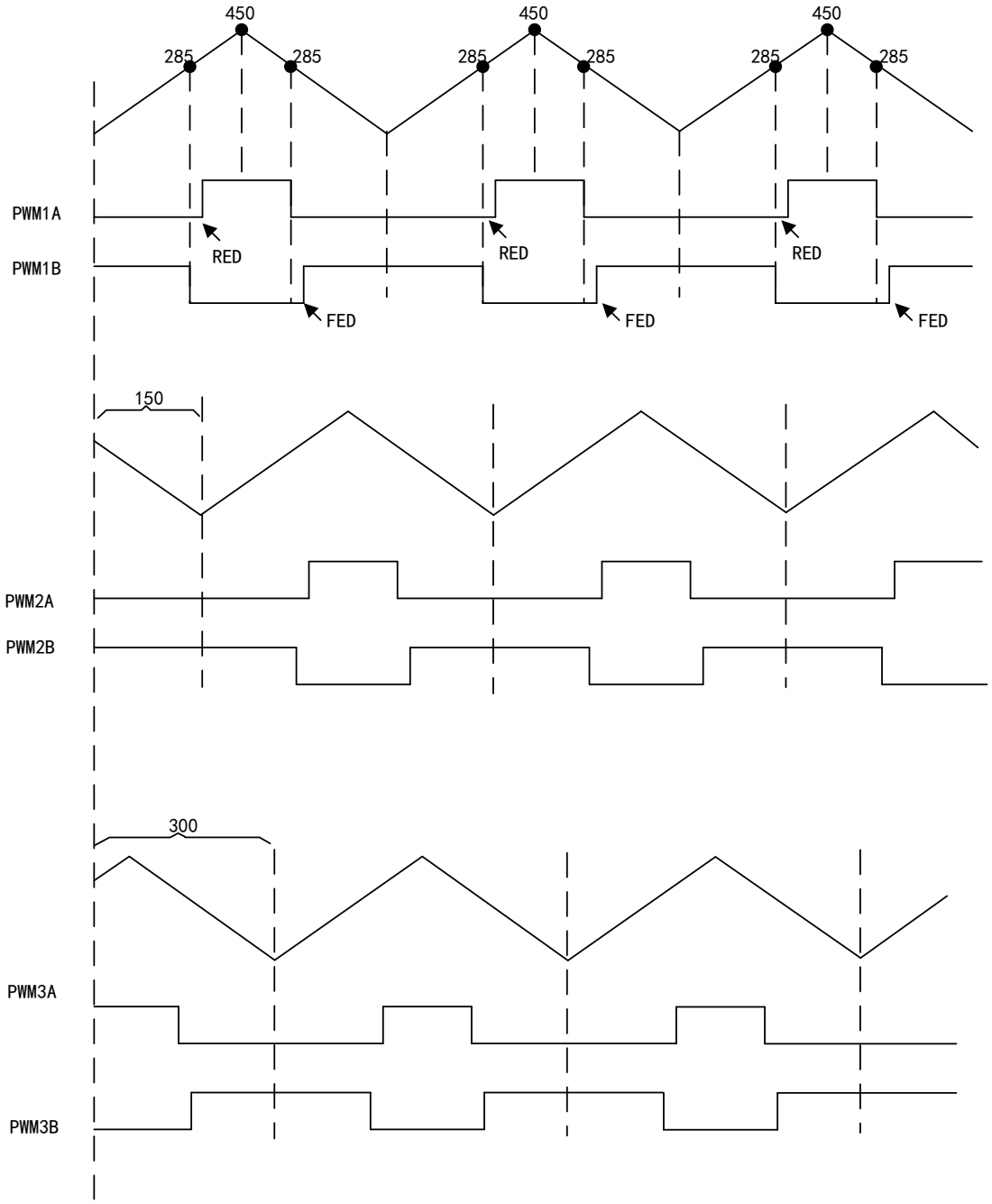


图 132 三相交错 DC/DC 转换器波形图



31.5.12.7 相位控制

为了确保正确的操作，通过设置 TBPHS 寄存器的相应数值，使多个 PWM 模块之间可以解决另一类依赖于桥臂之间相位关系的电源拓扑。为了同步输入脉冲可以将 TBPHS 寄存器的值装载到 TBCTR 寄存器中，可以配置 PWM 模块，详情请参见时基子模块功能描述章节。

从模块超前于主模块，两个模块之间的相位关系为 120°以及该配置的波形图如下图所示，其中主从模块的周期值为 600，从模块的为 200。每当主模块产生 TBCTR=TBPRD 同步脉冲时，将相位值装载到从模块的 PWM_TBCTR 寄存器中，因此从模块的时基总是比主模块的时基超前 120°。

图 133 相位控制两个 PWM 模块结构框图

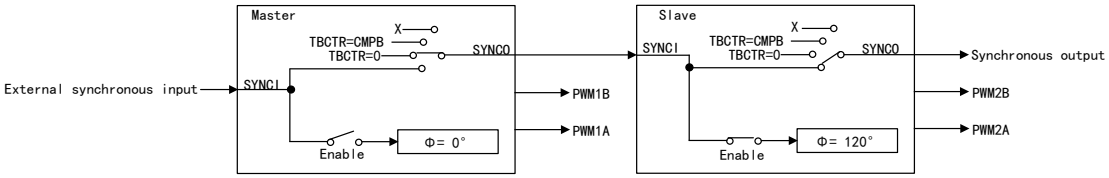
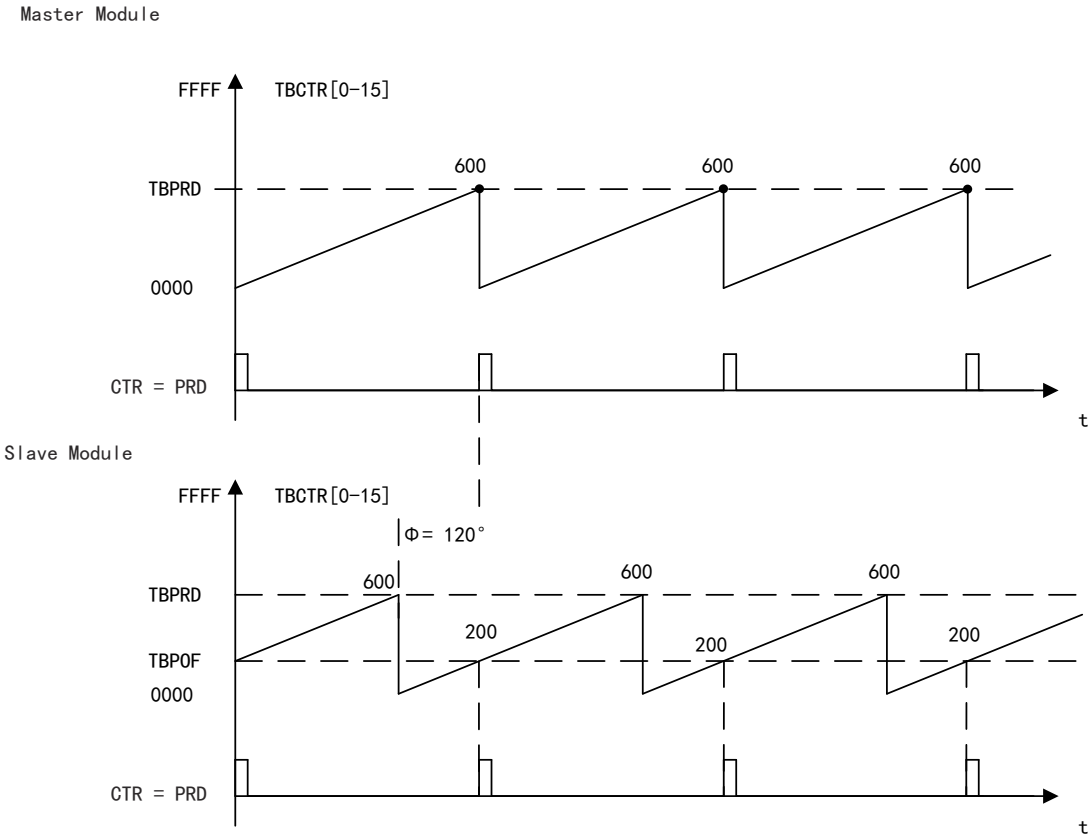


图 134 两个 PWM 模块之间相位同步的波形图

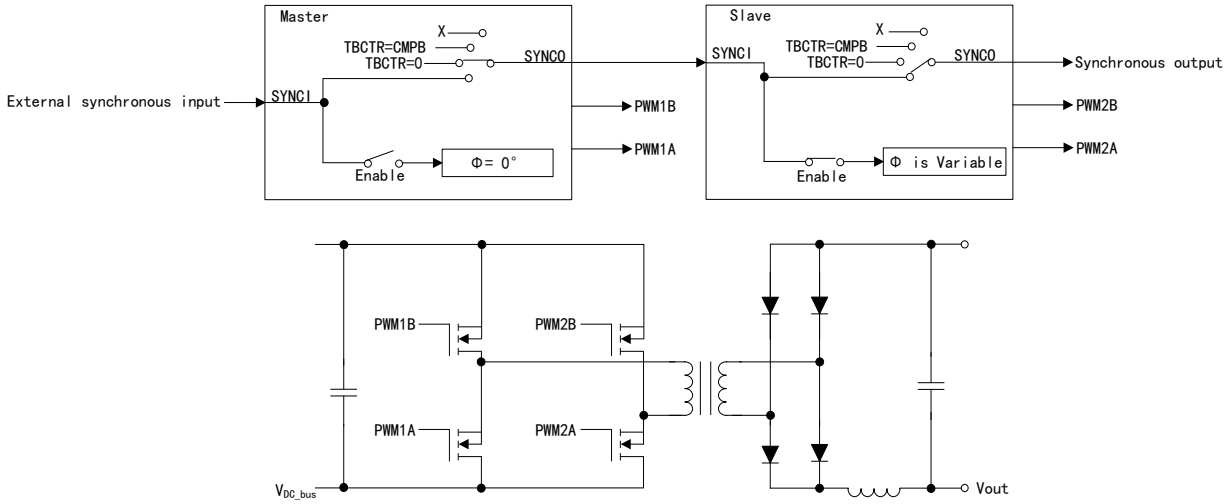


31.5.12.8 零电压开关全桥转换器

如全 H 桥控制结构框图所示，其中模块之间存在静态的相位关系。此时，可以通过调节占空比达到电源拓扑控制或通过逐周期来实现动态地改变相位关系。控制零电压开关全桥或相移全桥的电源结构符合这一特性。在该系统中：

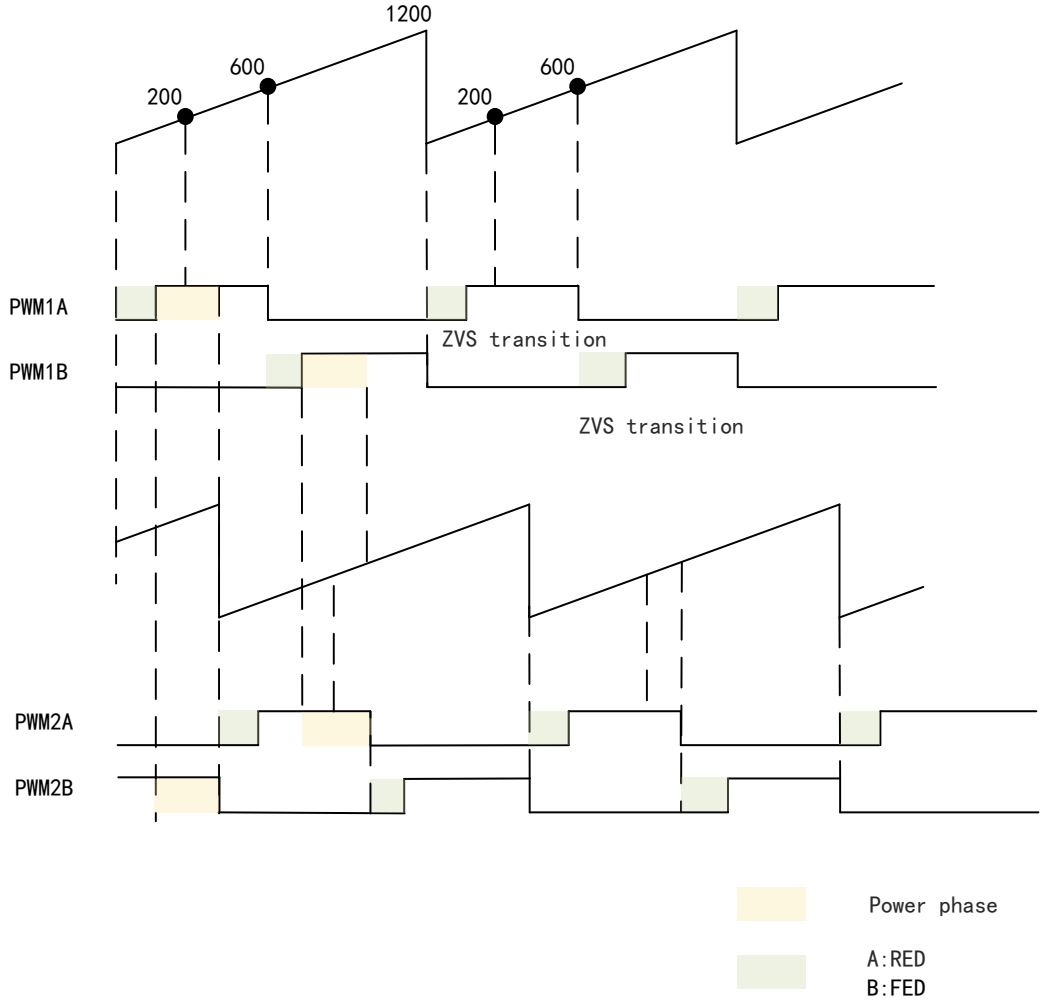
- 控制参数为模块之间的相位关系且占空比通常保持在约 50%
- 需要将两个 PWM 模块的资源分配给单个功率级，所以需要控制四个开关元件

图 135 全 H 桥控制结构框图（$F_{PWM1}=F_{PWM2}$）



使用主从模块组合以同步控制全 H 桥产生的波形图如下图所示，此时，主从模块都需要以相同的 PWM 频率进行切换。通过从模块 TBPHS 寄存器控制相位，如果没有使用主模块 TBPHS 寄存器，则将其初始化为 0。

图 136 全 H 桥波形图

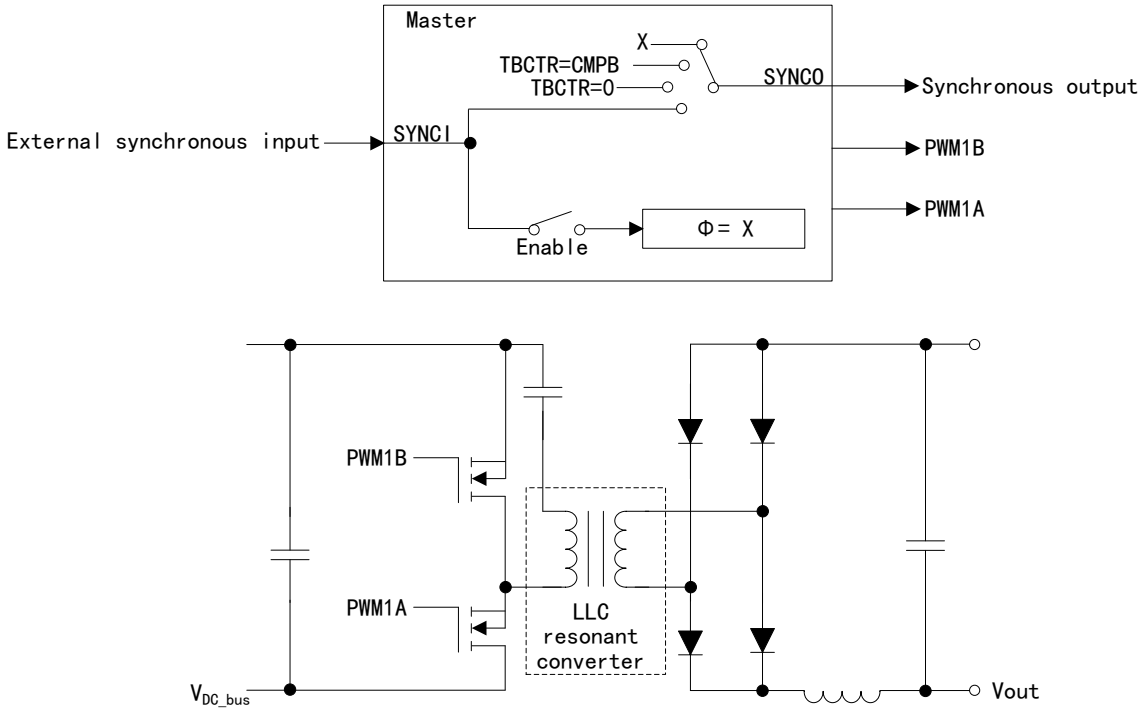


31.5.12.9 谐振变换器

在电力电子领域中，H 桥 LLC 谐振变换器拓扑结构也可以用于需要高效率和功率密度的消费电子应用中。两个谐振变换器结构框图以及 PWM 波形图如下图所示，PWM1 使用单通道配置，并可以扩展为多通道，此时：

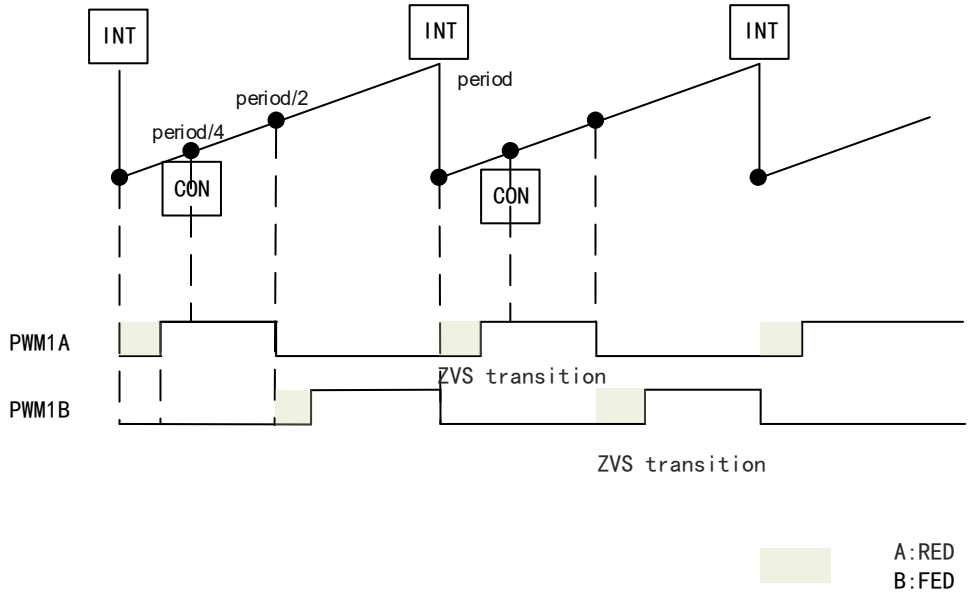
- 控制参数为频率且占空比通常保持在约 50%
- 死区不受控制且保持恒定为 300ns，但是用户可以实时更新死区
- 通过软件切换调整足够的延迟以提升效率

图 137 两个谐振变换器结构框图



注意：如果 Φ=X，则可以不在乎时基相位偏移寄存器的值。

图 138 谐振变换器 PWM 波形图



注意：

（1） INT 表示该事件触发中断，CON 表示该事件触发 ADC 开始转换。

（2） 其他未说明点为开始 ADC 转换。

31.5.13 寄存器锁存保护机制

为了保护关键的 PWM 寄存器，避免在代码失控时被意外损坏，增加了寄存器锁保护机制。在失控代码发生时，保护关键 PWM 寄存器不被意外写操作损坏。当 PWMLOCK 寄存器中 KEY 位域的值设置为 0xA5A5 时，才能成功写入此寄存器，详情请参见 PWMLOCK 寄存器描述。

PWMLOCK 寄存器定义的锁定位以及相应的寄存器，如下所示：

- HRLOCK：锁定 HRPWM 寄存器设置。锁定的寄存器为 HRCNFG、HRPWR、HRMSTEP 和 HRPCTL
- GLLOCK：锁定全局装载寄存器设置。锁定的寄存器为 GLDCTL 和 GLDCFG
- TZCFGLOCK：锁定跳闸区寄存器设置。锁定的寄存器为 TZSEL、TZDCSEL、TZCTL、TZCTL2、TZCTLDCx 和 TZEINT
- TZCLRLOCK：锁定跳闸区清除寄存器设置。锁定的寄存器为 TZCLR、TZCBCCLR、TZOSTCLR 和 TZFRC
- DCLOCK：锁定数字比较寄存器设置。锁定的寄存器为 DCTRIPSEL、DCxCTL、DCFCTL、DCCAPCTL、DCxHTRIPSEL 和 DCxLTRIPSEL

注意：由于同一个寄存器中 KEY 位域的存在，因此，当 KEY 匹配时，如果对该寄存器进行 16 位写操作，无论是写入该寄存器的上半部分或下半部分的 16 位，则都会被忽略，只有 32 位写操作才能成功。

31.6 寄存器组地址

表格 151 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Pwm1Regs	PWM_REGS	0x4000 0000	0x4000 03FF
Pwm2Regs	PWM_REGS	0x4000 0400	0x4000 07FF
Pwm3Regs	PWM_REGS	0x4000 0800	0x4000 0BFF
Pwm4Regs	PWM_REGS	0x4000 0C00	0x4000 0FFF
Pwm5Regs	PWM_REGS	0x4000 1000	0x4000 13FF
Pwm6Regs	PWM_REGS	0x4000 1400	0x4000 17FF
Pwm7Regs	PWM_REGS	0x4000 1800	0x4000 1BFF
Pwm8Regs	PWM_REGS	0x4000 1C00	0x4000 1FFF
SyncSocRegs	SYNC_SOC_REGS	0x4003_0C80	0x4003_0FFF

31.7 寄存器地址映射

表格 152 PWM_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
TBCTL	时基控制寄存器	0x00	-
TBCTL2	时基控制寄存器 2	0x02	-
TBCTR	时基计数器寄存器	0x08	-
TBSTS	时基状态寄存器	0x0A	-
CMPCTL	计数器比较控制寄存器	0x10	-
CMPCTL2	计数器比较控制寄存器 2	0x12	-
DBCTL	死区发生器控制寄存器	0x18	-
DBCTL2	死区发生器控制寄存器 2	0x1A	-
AQCTL	动作限定器控制寄存器	0x20	-
AQTSRCSEL	选择动作限定器触发器事件源寄存器	0x22	-
PCCTL	PWM 斩波控制寄存器	0x28	-
VCAPCTL	谷捕获控制寄存器	0x30	-
VCNTCFG	谷计数器配置寄存器	0x32	-
HRCNFG	HRPWM 配置寄存器	0x40	√
HRPWR	使能 HRP 校准逻辑寄存器	0x42	√
HRMSTEP	HRP 步骤数寄存器	0x4C	√
HRCNFG2	HRPWM 配置寄存器 2	0x4E	√

寄存器名称	寄存器描述	偏移地址	WRPRT
HRPCTL	高分辨率周期控制寄存器	0x5A	√
TRREM	HRPWM 转换高分辨率余数寄存器	0x5C	√
GLDCTL	全局 PWM 装载控制寄存器	0x68	√
GLDCFG	全局 PWM 装载事件配置寄存器	0x6A	√
PWMXLINK	PWMx 链接寄存器	0x70	-
AQCTLA	输出 A 的动作限定器控制寄存器	0x80	-
AQCTLA2	输出 A 的附加动作限定器控制寄存器	0x82	-
AQCTLB	输出 B 的动作限定器控制寄存器	0x84	-
AQCTLB2	输出 B 的附加动作限定器控制寄存器	0x86	-
AQSFRC	动作限定器软件强制寄存器	0x8E	-
AQCSFRC	动作限定器连续软件强制寄存器	0x92	-
DBREDHR	死区发生器上升沿延迟高分辨率寄存器	0xA0	-
DBRED	死区发生器上升沿延迟寄存器	0xA2	-
DBFEDHR	死区发生器下降沿延迟高分辨率寄存器	0xA4	-
DBFED	死区发生器下降沿延迟寄存器	0xA6	-
TBPHS	时基相位偏移寄存器	0xC0	-
TBPRDHR	高分辨率时基周期寄存器	0xC4	-
TBPRD	时基周期寄存器	0xC6	-
CMPA	计数比较模块 A 寄存器	0xD4	-
CMPB	计数比较模块 B 寄存器	0xD8	-
CMPC	计数比较模块 C 寄存器	0xDE	-
CMPD	计数比较模块 D 寄存器	0xE2	-
GLDCTL2	全局 PWM 装载控制寄存器 2	0xE8	-
SWVDELVAL	软件谷模式延迟寄存器	0xEE	-
TZSEL	跳闸区选择寄存器	0x100	√
TZDCSEL	跳闸区数字比较器配置寄存器	0x104	√
TZCTL	跳闸区控制寄存器	0x108	√
TZCTL2	跳闸区控制寄存器 2	0x10A	√
TZCTLDCA	跳闸区控制数字比较器 A 寄存器	0x10C	√
TZCTLDCB	跳闸区控制数字比较器 B 寄存器	0x10E	√
TZEINT	跳闸区中断使能寄存器	0x11A	√
TZFLG	跳闸区标志寄存器	0x126	-
TZCBCFLG	跳闸区 CCT 标志寄存器	0x128	-

寄存器名称	寄存器描述	偏移地址	WRPRT
TZOSTFLG	跳闸区 OST 标志寄存器	0x12A	-
TZCLR	清除跳闸区寄存器	0x12E	√
TZCBCCLR	清除跳闸区 CCT 寄存器	0x130	√
TZOSTCLR	清除跳闸区 OST 寄存器	0x132	√
TZFRC	强制跳闸区标志寄存器	0x136	√
ETSEL	事件触发选择寄存器	0x148	-
ETPS	事件触发预分配寄存器	0x14C	-
ETFLG	事件触发标志寄存器	0x150	-
ETCLR	清除事件触发寄存器	0x154	-
ETFRC	强制事件触发寄存器	0x158	-
ETINTPS	事件触发中断预分频寄存器	0x15C	-
ETSOCPS	事件触发 SOC 预分频寄存器	0x160	-
ETCNTINITCTL	事件触发计数器初始化控制寄存器	0x164	-
ETCNTINIT	事件触发计数器初始化寄存器	0x168	-
DCTRIPSEL	数字比较跳闸选择寄存器	0x180	√
DCACTL	数字比较 A 控制寄存器	0x186	√
DCBCTL	数字比较 B 控制寄存器	0x188	√
DCFCTL	数字比较滤波器控制寄存器	0x18E	√
DCCAPCTL	数字比较捕获控制寄存器	0x190	√
DCFOFFSET	数字比较滤波器偏移寄存器	0x192	-
DCFOFFSETCNT	数字比较滤波器偏移计数器寄存器	0x194	-
DCFWINDOW	数字比较滤波器窗口寄存器	0x196	-
DCFWINDOWCNT	数字比较滤波器窗口计数器寄存器	0x198	-
DCCAP	数字比较计数器捕获寄存器	0x19E	-
DCAHTRIPSEL	数字比较 A 高跳闸选择寄存器	0x1A4	√
DCALTRIPSEL	数字比较 A 低跳闸选择寄存器	0x1A6	√
DCBHTRIPSEL	数字比较 B 高跳闸选择寄存器	0x1A8	√
DCBLTRIPSEL	数字比较 B 低跳闸选择寄存器	0x1AA	√
PWMLOCK	PWM 锁定寄存器	0x1F4	-
HWVDELVAL	硬件谷模式延迟寄存器	0x1FA	-
VCNTVAL	硬件谷计数器寄存器	0x1FC	-

表格 153 SYNC_SOC_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
SYNCSELECT	同步输入和输出选择寄存器	0x00	√
ADCSOCOUTSELECT	外部 ADCSOC 选择寄存器	0x04	√
SYNCSOCLOCK	同步输入源和外部 ADCSOC 选择锁定寄存器	0x08	√

31.8 寄存器功能描述

31.8.1 时基控制寄存器（TBCTL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	CTRMODE	R/W	选择计数器模式（Counter Mode Select） TBCTR 在正常工作后不会更改工作模式。更改 TBCTR 工作模式后，将会在下一个 TBCLK 边沿生效且 TBCTR 值从更改前的值开始递增/递减。 00：向上计数模式 01：向下计数模式 10：中心对齐模式 11：冻结计数模式（默认）	3h
2	PHSEN	R/W	使能相位（Phase Enable） 0：禁止 1：使能，并允许 TBPRD 寄存器装载影子寄存器同步事件（出现软件强制同步信号或 PWMx 同步输入信号时）	0h
3	PRDLD	R/W	选择时基周期装载模式（Time-Base Period Load Mode Select） 0：影子模式。当 TBCTR =0 且/或 PRDLDSYNC=01 时， TBPRD 寄存器装载影子寄存器的值，读/写 TBPRD 寄存器实际访问的是影子寄存器 1：立即模式。立即绕过影子寄存器，读/写 TBPRD 寄存器实际访问的是活动寄存器	0h
5:4	SYNCOSEL	R/W	选择同步输出信号源（Sync Output Source Select） 00：PWMx 同步输入或 SWFSYNC 01：TBCTR =0 10：TBCTR =CMPB 11：由 TBCTL2[SYNCOSELX]位决定 PWMx 同步输出	0h
6	SWFSYNC	R/S	软件强制同步脉冲（Sync Pulse Forced by Software） 该位仅在 SYNCOUT=00 时才会影响到 PWMx 同步输出。 0：无效，读取返回 0 1：强制产生一个同步脉冲	0h

位/域	名称	R/W	描述	复位值
9:7	HSPCLKDIV	R/W	高速时基时钟预分频系数（High Speed Time Base Clock Pre-Scale Factor） TBCLK = PWMCLK / (HSPCLKDIV * CLKDIV) 000：分频系数为 1 001：分频系数为 2（默认） 010：分频系数为 4 011：分频系数为 6 100：分频系数为 8 101：分频系数为 10 110：分频系数为 12 111：分频系数为 14	1h
12:10	CLKDIV	R/W	配置时基时钟预分频系数（Time Base Clock Pre-Scale Factor Configure） TBCLK = PWMCLK / (HSPCLKDIV * CLKDIV) 000：分频系数为 1（默认） 001：分频系数为 2 010：分频系数为 4 011：分频系数为 8 100：分频系数为 16 101：分频系数为 32 110：分频系数为 64 111：分频系数为 128	0h
13	PHSDIR	R/W	配置计数方向（Count Direction Configure） 发生同步事件后，将 TBPOF 寄存器的新值装载到 TBCTR，并通过该位配置 TBCTR 计数方向，而与 TBCTR 在同步事件前的方向无关。 该位仅适用于 TBCTR 配置为中心对齐计数模式，故该位在向上计数模式和向下计数模式中会被忽略。 0：向下计数（同步事件后） 1：向上计数（同步事件后）	0h
15:14	FREE_SOFT	R/W	配置仿真模式（Emulation Mode Configure） 这些位域配置仿真事件期间 PWM 时基计数器的行为。 00：时基计数器在下一次递增或递减后停止 01：时基计数器在完成整个循环后停止 （1）向上计数模式：TBCTR 与周期相等时停止 （2）向下计数模式：TBCTR=0x00 时停止 （3）中心对齐计数模式：TBCTR=0x00 时停止 1x：自由运行	0h

31.8.2 时基控制寄存器 2（TBCTL2）

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
5:0	保留			0h

位/域	名称	R/W	描述	复位值
6	OSHTSYNCMODE	R/W	使能单次同步模式（Oneshot Sync Mode Enable） 0：禁止 1：使能	0h
7	OSHTSYNC	R/S	单次同步脉冲（Oneshot Sync Pulse） 0：无效，读取返回 0 1：允许通过一个同步脉冲	0h
11:8	保留			0h
13:12	SYNCOSELX	R/W	选择扩展同步输出信号（Extended SYNCOUT Select） 00：禁止 PWMx 同步输出信号 01：同步输出信号为 CMPC 10：同步输出信号为 CMPD 11：保留	0h
15:14	PRDLDSYNC	R/W	选择同步事件周期装载条件（Period Load Condition on SYNC event Select） 该位域仅在 PRDLD=0 时有效。 满足以下条件时， TBPRD 影子寄存器的值被装载到活动寄存器： 00：仅当 TBCTR =0 时 01：当 TBCTR =0 或发生同步事件时 10：仅发生同步事件时 11：保留	0h

31.8.3 时基计数器寄存器（TBCTR）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	TBCTR	R/W	时基计数器（Time Base Counter）	0h

31.8.4 时基状态寄存器（TBSTS）

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CTRDIR	R	计数方向标志（Count Direction Flag） 0：正在进行向下计数 1：正在进行向上计数	1h
1	SYNCI	R_W1C	外部同步事件标志（External Synchronization Event Flag） 写 1 清除锁存的事件。 0：写 0 无效，没有外部同步事件发生 1：有外部同步事件发生	0h

位/域	名称	R/W	描述	复位值
2	CTRMAX	R_W1C	时基计数器达到最大值标志（Time-base Counter Reached the Max Value Flag） 写 1 清除锁存的事件。 0：TBCTR 从未达到最大值 1：TBCTR 已未达到最大值 0xFFFF	0h
15:3	保留			0h

31.8.5 计数器比较控制寄存器（CMPCTL）

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	LOADAMODE	R/W	选择 CMPA 装载事件（Counter-Compare A (CMPA) Load Event Select） 该位域在 SHDWAMODE=1 时无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPA 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）	0h
3:2	LOADBMODE	R/W	选择 CMPB 装载事件（Counter-Compare B (CMPB) Load Event Select） 该位域在 SHDWBMODE=1 时无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPB 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）	0h
4	SHDWAMODE	R/W	CMPA 装载模式（Counter-Compare A (CMPA) Load Mode） 0：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 A。 1：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPA 以双缓冲区方式工作。	0h
5	保留			0h
6	SHDWBMODE	R/W	CMPB 装载模式（Counter-Compare B (CMPB) Load Mode） 0：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 B。 1：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPB 以双缓冲区方式工作。	0h
7	保留			0h

位/域	名称	R/W	描述	复位值
8	SHDWAFULL	R	CMPA 影子满状态（CMPA Shadow Full Status） 在发生装载脉冲时，该位将自动清零。 该位设置条件： （1）如果对 CMPA 寄存器进行 32-位写操作或对 CMPA 寄存器进行 16-位写操作，则设置该位。 （2）如果对 CMPA 寄存器进行 16-位写操作时，则该位不会被设置。 0：CMPA 影子寄存器未满 1：CMPA 影子寄存器已满，进行 CPU 写操作将会覆盖当前影子值	0h
9	SHDWBFULL	R	CMPB 影子满状态（CMPB Shadow Full Status） 在发生装载脉冲时，该位将自动清零。 0：CMPB 影子寄存器未满 1：CMPB 影子寄存器已满，进行 CPU 写操作将会覆盖当前影子值	0h
11:10	LOADASYNC	R/W	配置 CMPA 装载条件（CMPA Load Condition Configure） 该位只有在 SHDWAMODE=0 的情况下才有效。 在发生同步事件时，满足以下条件，可以将 CMPA 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADAMODE 位时装载 01：在设置 LOADAMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留	0h
13:12	LOADBSYNC	R/W	配置 CMPB 装载条件（CMPB Load Condition Configure） 该位仅在 SHDWBMODE=0 的条件下才有效。 在发生同步事件时，满足以下条件，可以将 CMPB 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADBMODE 位时装载 01：在设置 LOADBMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留	0h
15:14	保留			0h

31.8.6 计数器比较控制寄存器 2（CMPCTL2）

偏移地址：0x12

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	LOADCMODE	R/W	选择 CMPC 装载事件（Counter-Compare C (CMPC) Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPC 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）	0h
3:2	LOADDMODE	R/W	选择 CMPD 装载事件（Counter-Compare D (CMPD) Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将影子寄存器的值装载到活动 CMPD 寄存器中： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载操作（冻结）	0h
4	SHDWCMODE	R/W	CMPC 装载模式（Counter-Compare C (CMPC) Load Mode） 0：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 C。 1：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPC 以双缓冲区方式工作。	0h
5	保留			0h
6	SHDWDMODE	R/W	CMPD 装载模式（Counter-Compare D (CMPD) Load Mode） 0：立即模式。为了进行立即比较模式，所有写入和读取操作都可以直接访问活动寄存器。仅适用于活动比较寄存器 D。 1：影子模式。在该模式下，所有由 CPU 进行的写操作都会访问影子寄存器。CMPD 以双缓冲区方式工作。	0h
9:7	保留			0h
11:10	LOADCSYNC	R/W	配置 CMPC 装载条件（CMPC Load Condition Configure） 该位只有在 SHDWCMODE=0 的情况下才有效。 在发生同步事件时，满足以下条件，可以将 CMPC 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADCMODE 位装载 01：在设置 LOADCMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留	0h

位/域	名称	R/W	描述	复位值
13:12	LOADDSYNC	R/W	配置 CMPD 装载条件（CMPD Load Condition Configure） 该位只有在 SHDWDMODE=0 的情况下才有效。 在发生同步事件时，满足以下条件，可以将 CMPD 的值从影子寄存器装载到活动寄存器中： 00：在设置 LOADDMODE 位装载 01：在设置 LOADDMODE 位和发生同步事件时装载 10：仅接收到同步事件时装载 11：保留	0h
15:14	保留			0h

31.8.7 死区发生器控制寄存器（DBCTL）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	OUT_MODE	R/W	死区输出选择（Dead-Band Output Select） [0]：控制 S0 开关。 [1]：控制 S1 开关 00：忽略或完全禁止 DBM。在该情况下，POLSEL 位和 IN_MODE 位无效。 01：A 路径等于 A 信号路径的延迟旁路（InA），B 路径等于 B 信号路径中的下降沿延迟。 10：A 路径等于 A 信号路径中的上升沿延迟，B 路径等于 B 信号路径的延迟旁路（InB）。 11：完全使能 DBM，上升沿延迟和下降沿延迟处于活动状态。	0h
3:2	POLSEL	R/W	选择翻转的延迟信号（Invert Delayed Signal Select） [2]：控制 S2 开关。 [3]：控制 S3 开关。 通过该位域可以在 PWMA 延迟信号或 PWMB 延迟信号传输出死区发生器子模块前，对其进行翻转。 OUT_MODE=11 和 IN_MODE=00 对应数字电动机控制逆变器的一个支路上发现的典型上/下开关控制。其他增强模式不是典型使用模式。 00：高电平有效（AH）模式（默认）。未对 PWMA 延迟信号和 PWMB 延迟信号进行翻转。 01：低电平有效互补（ALC）模式。仅对 PWMA 延迟信号进行翻转。 10：高电平有效互补（AHC）模式。仅对 PWMB 延迟信号进行翻转。 11：低电平有效（AL）模式。都对 PWM 延迟信号 A 和 PWMB 延迟信号进行翻转。	0h
5:4	IN_MODE	R/W	死区输入选择（Dead-Band Input Select） [4]：控制 S4 开关。 [5]：控制 S5 开关。	0h

位/域	名称	R/W	描述	复位值
			该位域可以选择上升沿延迟和下降沿延迟的输入源。默认上升沿延迟和下降沿延迟的输入源为 PWMxA In 以便产生典型死区波形。 00：上升沿延迟和下降沿延迟的输入源为 PWMxA In 01：下降沿延迟的输入源为 PWMxA In，上升沿延迟输入源为 PWMxB In 10：上升沿延迟输入源为 PWMxA In，下降沿延迟的输入源为 PWMxB In 11：上升沿延迟和下降沿延迟的输入源为 PWMxB In	
7:6	LOADREDMODE	R/W	选择上升沿装载事件（Rising Edge Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将从影子寄存器装载到活动 DBRED 寄存器： 00：在计数器等于 0 时装载 01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）	0h
9:8	LOADFEDMODE	R/W	选择下降沿装载事件（Falling Edge Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将从影子寄存器装载到活动 DBFED 寄存器： 00：在计数器等于 0 时装载 01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）	0h
10	SHDWDBRED MODE	R/W	死区上升沿装载模式（RED Dead-Band Load Mode） 0：立即模式。为了立即进行下降沿延迟死区操作，所有通过 CPU 进行的写入或读取都可以直接访问活动寄存器。仅适用于活动 DBRED 寄存器。 1：影子模式（默认）。所有通过 CPU 的写操作都会访问影子寄存器。将以双缓冲区方式操作。	0h
11	SHDWDBFED MODE	R/W	死区下降沿装载模式（FED Dead-Band Load Mode） 0：立即模式。为了立即进行下降沿延迟死区操作，所有通过 CPU 进行的写入或读取都可以直接访问活动寄存器。仅适用于活动 DBFED 寄存器。 1：影子模式（默认）。所有通过 CPU 的写操作都会访问影子寄存器。将以双缓冲区方式操作。	0h
13:12	OUTSWAP	R/W	死区输出交换控制（Dead Band Output Swap Control） [12]：控制 S6 开关。 [13]：控制 S7 开关。 00：由 OUT_MODE 位域定义 PWMxA 输出信号和 PWMxB 输出信号 01：由 OUT_MODE 位域定义 PWMxA 输出信号为 A 路径。由 OUT_MODE 位域定义 PWMxB 输出信号为 A 路径，即等于 A 信号路径的上升沿延迟或延迟旁路。	0h

位/域	名称	R/W	描述	复位值
			10：由 OUT_MODE 位域定义 PWMxA 输出信号为 B 路径，即等于 B 信号路径的下降沿延迟或延迟旁路。由 OUT_MODE 位域定义 PWMxB 输出信号为 B 路径。 11：由 OUT_MODE 位域定义 PWMxA 输出信号为 B 路径，即等于 B 信号路径的下降沿延迟或延迟旁路。由 OUT_MODE 位域定义 PWMxB 输出信号为 A 路径，即等于 A 信号路径的上升沿延迟或延迟旁路。	
14	DEDB_MODE	R/W	死区双边延时 B 模式控制（Dead Band Dual-Edge Delay B Mode Control） 该位控制 S8 开关。 如果该位置位，用户则可以设置 OUT_MODE=01（使 A 路径等于 InA）或者设置 OUTSWAP=1x（使 PWMxA 输出信号为 B 路径），否则 PWMxA 无效。 0：上升沿延迟用于由 A 信号路径 DBCTL[4]位（S4 开关）选择的 InA/InB，下降沿延迟用于由 B 信号路径 DBCTL[5]位（S5 开关）选择的 InA/InB 1：上升沿延迟和下降沿延迟用于由 DBCTL[4]位（S4 开关）选择的输入源并输出到 B 信号路径	0h
15	HALFCYCLE	R/W	使能半周期时钟（Half Cycle Clocking Enable） 0：使能全周期时钟，此时，死区计数器时钟为 TBCLK 1：使能半周期时钟，此时，死区计数器时钟为 2*TBCLK	0h

31.8.8 死区发生器控制寄存器 2（DBCTL2）

偏移地址：0x1A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	LOADDBCTLMODE	R/W	选择 DBCTL 装载事件（DBCTL Load Event Select） 该位域在立即模式下无效。 00：在计数器等于 0 时装载 01：在计数器等于周期时装载 10：在计数器等于 0 或计数器等于周期时装载 11：禁止装载（冻结模式）	0h
2	SHDWDBCTLMODE	R/W	DBCTL 装载模式（DBCTL Load Mode） 0：立即模式。所有通过 CPU 写入或读取操作都可以直接访问活动寄存器。仅适用于活动 DBCTL 寄存器。 1：影子模式。除了对 DBCTL[5:0]的所有写入和读取操作都会传输到影子寄存器，其他位域仍然访问活动寄存器。	0h
15:3	保留			0h

31.8.9 动作限定器控制寄存器（AQCTL）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	LDAQAMODE	R/W	选择动作模块 A 装载事件（Action Qualifier A Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将从影子寄存器装载到活动 AQCTLA 寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载（冻结模式）	0h
3:2	LDAQBMODE	R/W	选择动作模块 B 装载事件（Action Qualifier B Load Event Select） 该位域在立即模式下无效。 满足以下条件时，将从影子寄存器装载到活动 AQCTLB 寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：禁止装载（冻结模式）	0h
4	SHDWAQAMODE	R/W	动作模块 A 装载模式（Action Qualifier A Load Mode） 0：影子模式。所有写操作由 CPU 通过访问影子寄存器进行。AQCTLA 寄存器用于双缓冲区操作。 1：立即模式。在该模式下，所有写入或读取都可以直接访问活动寄存器。仅适用于活动动作限定器寄存器。	0h
5	保留			0h
6	SHDWAQBMODE	R/W	动作模块 B 装载模式（Action Qualifier B Load Mode） 0：影子模式。所有写操作由 CPU 通过访问影子寄存器进行。AQCTLB 寄存器用于双缓冲区操作。 1：立即模式。在该模式下，所有写入或读取都可以直接访问活动寄存器。仅适用于活动动作限定器寄存器。	0h
7	保留			0h
9:8	LDAQASYNC	R/W	配置 AQCTLA 装载条件（AQCTLA Load Condition Configure） 该位只有在 SHDWAQAMODE=1 的情况下才有效。 满足以下条件下，将控制 AQCTLA 寄存器在何时进行影子寄存器装载到活动寄存器操作： 00：在设置 LDAQAMODE 位时装载 01：在设置 LDAQAMODE 位和发生同步事件时装载 10：在接收到同步事件时装载 11：保留	0h

位/域	名称	R/W	描述	复位值
11:10	LDAQBSYNC	R/W	配置 AQCTLB 装载条件（AQCTLB Load Condition Configure） 该位只有在 SHDWAQBMODE=1 的情况下才有效。 满足以下条件下，将控制 AQCTLB 寄存器在何时进行影子寄存器装载到活动寄存器操作： 00：在设置 LDAQBMODE 位时装载 01：在设置 LDAQBMODE 位和发生同步事件时装载 10：在接收到同步事件时装载 11：保留	0h
15:12	保留			0h

31.8.10 选择动作限定器触发器事件源寄存器（AQTSRCSEL）

偏移地址：0x22

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	T1SEL	R/W	配置 T1 事件源（T1 Event Source Configure） 0000：DCAEVT1 0001：DCAEVT2 0010：DCBEVT1 0011：DCBEVT2 0100：TZ1 0101：TZ2 0110：TZ3 0111：PWMx 同步输入 1000：DCEVTFILT 其他：保留	0h
7:4	T2SEL	R/W	配置 T2 事件源（T2 Event Source Configure） 0000：DCAEVT1 0001：DCAEVT2 0010：DCBEVT1 0011：DCBEVT2 0100：TZ1 0101：TZ2 0110：TZ3 0111：PWMx 同步输入 1000：DCEVTFILT 其他：保留	0h
15:8	保留			0h

31.8.11 PWM 斩波控制寄存器（PCCTL）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CHPEN	R/W	使能斩波功能（Chopping Function Enable） 0：禁止 1：使能	0h
4:1	OSHTWTH	R/W	配置单次脉冲宽度（One-Shot Pulse Width Configure） 0000：PWMCLK / 8 宽度*1，PWMCLK=100MHz 时，单次脉冲宽度为 80ns 0001：PWMCLK / 8 宽度*2，PWMCLK=100MHz 时，单次脉冲宽度为 160ns 0010：PWMCLK / 8 宽度*3，PWMCLK=100MHz 时，单次脉冲宽度为 240ns 0011：PWMCLK / 8 宽度*4，PWMCLK=100MHz 时，单次脉冲宽度为 320ns 0100：PWMCLK / 8 宽度*5，PWMCLK=100MHz 时，单次脉冲宽度为 400ns 0101：PWMCLK / 8 宽度*6，PWMCLK=100MHz 时，单次脉冲宽度为 480ns 0110：PWMCLK / 8 宽度*7，PWMCLK=100MHz 时，单次脉冲宽度为 560ns 0111：PWMCLK / 8 宽度*8，PWMCLK=100MHz 时，单次脉冲宽度为 640ns 1000：PWMCLK / 8 宽度*9，PWMCLK=100MHz 时，单次脉冲宽度为 720ns 1001：PWMCLK / 8 宽度*10，PWMCLK=100MHz 时，单次脉冲宽度为 800ns 1010：PWMCLK / 8 宽度*11，PWMCLK=100MHz 时，单次脉冲宽度为 880ns 1011：PWMCLK / 8 宽度*12，PWMCLK=100MHz 时，单次脉冲宽度为 9600ns 1100：PWMCLK / 8 宽度*13，PWMCLK=100MHz 时，单次脉冲宽度为 1040ns 1101：PWMCLK / 8 宽度*14，PWMCLK=100MHz 时，单次脉冲宽度为 1120ns 1110：PWMCLK / 8 宽度*15，PWMCLK=100MHz 时，单次脉冲宽度为 12000ns 1111：PWMCLK / 8 宽度*16，PWMCLK=100MHz 时，单次脉冲宽度为 1280ns	0h

位/域	名称	R/W	描述	复位值
7:5	CHPFREQ	R/W	斩波时钟分频（Chopping Clock Prescaler） 000：分频系数为 1，TBCLK=100MHz 时，斩波时钟为 12.5MHz 001：分频系数为 2，TBCLK=100MHz 时，斩波时钟为 6.25MHz 010：分频系数为 3，TBCLK=100MHz 时，斩波时钟为 4.16MHz 011：分频系数为 4，TBCLK=100MHz 时，斩波时钟为 3.12MHz 100：分频系数为 5，TBCLK=100MHz 时，斩波时钟为 2.50MHz 101：分频系数为 6，TBCLK=100MHz 时，斩波时钟为 2.08MHz 110：分频系数为 7，TBCLK=100MHz 时，斩波时钟为 1.78MHz 111：分频系数为 8，TBCLK=100MHz 时，斩波时钟为 1.56MHz	0h
10:8	CHPDUTY	R/W	配置斩波时钟周期占空比（Chopping Clock Duty Cycle Configure） 000：占空比为 12.5% 001：占空比为 25% 010：占空比为 37.5% 011：占空比为 50% 100：占空比为 62.5% 101：占空比为 75% 110：占空比为 87.5% 111：保留	0h
15:11	保留			0h

31.8.12 谷捕获控制寄存器（VCAPCTL）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	VCAPE	R/W	使能谷值捕获（Valley Capture Enable） 0：禁止 1：使能	0h
1	VCAPSTART	R/W	启动谷值捕获（Valley Capture Start） 在选择 TRIGSEL 位进行软件触发条件下，向该位写入 1 可以通过软件触发一个捕获序列。 0：无效 1：当 TRIGSEL=000 时，触发一次捕获序列	0h

位/域	名称	R/W	描述	复位值
4:2	TRIGSEL	R/W	选择捕获序列触发事件（Capture Sequence Trigger Event Select） 通过这些位域所选事件来触发谷捕获序列，如果发生所选事件，则捕获序列立即触发，且无论当前捕获状态都将进行重新触发捕获序列。 根据 DCFCTL[SRCSEL]位域所选事件进行捕获。 000：通过软件写入 TRIGSEL 位来触发捕获序列 001：TBCTR=0 事件 010：TBCTR=TBPRD 事件 011：TBCTR=0 或 TBCTR=TBPRD 事件 100：DCAEVT1 事件 101：DCAEVT2 事件 110：DCBEVT1 事件 111：DCBEVT2 事件 注意：TRIGSEL 位和 SRCSEL 位不能同时选择同一事件。	0h
6:5	保留			0h
9:7	VDELAYDIV	R/W	配置谷值延迟模式分频（Valley Delay Mode Divide Configure） 通过使用这些位域可以调整连续边沿捕获之间的延迟值。 000：硬件谷延迟值 = 软件谷延迟值 001：硬件谷延迟值 = 谷时基计数器 + 软件谷延迟值 010：硬件谷延迟值 = 谷时基计数器右移 1 位 + 软件谷延迟值 011：硬件谷延迟值 = 谷时基计数器右移 2 位 + 软件谷延迟值 100：硬件谷延迟值 = 谷时基计数器右移 4 位 + 软件谷延迟值 其他：保留	0h
10	EDGEFILTDLYSEL	R/W	边沿滤波器输出延迟（Edge Filter Output Delay） 0：无延迟 1：边沿滤波器输出 HWVDELVAL 延迟	0h
15:11	保留			0h

31.8.13 谷计数器配置寄存器（VCNTCFG）

偏移地址：0x32

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	STARTEDGE	R/W	选择启动计数的边沿（Start Count Edge Select） 当激活计数器操作且出现 TRIGSEL 位选择的触发事件，通过这些位域选择谷计数器开始计数的边沿。 0000：未开始计数 0001：第 1 个边沿开始计数 0010：第 2 个边沿开始计数 0011：第 3 个边沿开始计数 ... 1111：第 15 个边沿开始计数	0h
6:4	保留			0h
7	STARTEDGESTS	R	启动计数的边沿状态（Start Count Edge Status） 出现 TRIGSEL 位选择的触发事件时，该位进行复位。如果触发捕获序列且出现开始计数的边沿，则设置该位。 0：未出现开始计数的边沿 1：出现开始动计数的边沿	0h
11:8	STOPEDGE	R/W	选择停止计数的边沿（Stop Count Edge Select） 当激活计数器操作且出现 TRIGSEL 位选择的触发事件，通过这些位域选择谷计数器停止计数的边沿。 0000：未停止计数 0001：第 1 个边沿停止计数 0010：第 2 个边沿停止计数 0011：第 3 个边沿停止计数 ... 1111：第 15 个边沿停止计数	0h
14:12	保留			0h
15	STOPEDGESTS	R	停止计数的边沿状态（Stop Count Edge Status） 发生 TRIGSEL 位选择的触发事件时，该位进行复位。如果触发捕获序列且出现停止计数的边沿，则设置该位。 0：未出现停止计数的边沿 1：出现停止计数的边沿	0h

31.8.14 HRPWM 配置寄存器（HRCNFG）

偏移地址：0x40

复位类型：SYSRSn

该寄存器仅适用于具有 HRPWM 功能的 PWM 模块。

位/域	名称	R/W	描述	复位值
1:0	EDGMODE	R/W	选择 A 路信号边沿模式（PWMxA Edge Mode Select） 该位域用于选择由 HRP 逻辑控制 A 路信号边沿。 00：禁用 HRPWM 功能（默认） 01：HRP 控制 A 路信号的上升沿，此时，该上升沿由 CMPAHR 控制 10：HRP 控制 A 路信号的下降沿，此时，该下降沿由 CMPAHR 控制 11：HRP 同时控制 A 路信号的上升沿和下降沿，此时，两个边沿分别由 TBPRDHR 寄存器或 TBPHSHR 控制	0h

位/域	名称	R/W	描述	复位值
2	CTLMODE	R/W	选择 A 路信号控制模式（PWMxA Control Mode Select） 该位用于选择控制 A 路信号边沿位置的寄存器 0：周期或占空比控制模式（默认），即边沿位置由 TBPRDHR 寄存器或 CMPAHR 控制 1：相位控制模式，即边沿位置由 TBPHSHR 控制	0h
4:3	HRLOAD	R/W	选择 A 路信号影子值装载事件（PWMxA Shadow Value Load Event Select） 该位域用于选择将 CMPAHR 的值装载到活动寄存器的时间事件。 满足以下条件时，将 CMPAHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留	0h
5	SELOUTB	R/W	配置 B 路输出信号反相（PWMxB Output Invert Configure） 该位配置 PWMxB 通道的输出信号类型。在进行反相操作时，会考虑使用高分辨率模式，且反相后的 PWMxB 信号将包含由高分辨率模式下的任何变化。 0：B 路输出正常信号 1：B 路输出与 A 路相反的信号	0h
6	AUTOCONV	R/W	使能自动转换（Auto Convert Enable） 该位用于选择将 TBPHSHR 寄存器、TBPRDHR 寄存器或 CMPAHR 寄存器中周期、分数占空比或相位通过软件计算进行手动缩放或通过 HRMSTEP 寄存器的 HRP 比例系数进行自动缩放。 SFO 库函数会根据实际需求，计算合适的 HRP 比例系数并自动更新到 HRMSTEP 寄存器中。 在应用软件手动缩放分数占空比或相位时，必须禁用自动转换。 0：禁止 1：使能	0h
7	SWAPAB	R/W	交换 A 路和 B 路输出信号（PWM A & B Output Signals Swap） 该位可以交换 PWMxA 和 PWMxB 输出信号。 0：PWMxA 和 PWMxB 输出信号保持不变 1：交换 PWMxA 和 PWMxB 输出信号	0h
9:8	EDGMODEB	R/W	选择 B 路信号边沿模式（PWMxB Edge Mode Select） 该位域用于选择由 HRP 逻辑控制 B 路信号边沿。 00：禁用 HRPWM 功能（默认） 01：HRP 控制 B 路信号的上升沿，此时，该上升沿由 CMPBHR 寄存器控制 10：HRP 控制 B 路信号的下降沿，此时，该下降沿由 CMPBHR 寄存器控制 11：HRP 控制 B 路信号的上升沿和下降沿，此时，两个边沿由 TBPRDHR 寄存器或 TBPHSHR 寄存器控制	0h

位/域	名称	R/W	描述	复位值
10	CTLMODEB	R/W	选择 B 路信号控制模式（PWMxB Control Mode Select） 该位用于选择控制 B 路信号边沿位置的寄存器 0：周期或占空比控制模式（默认），即边沿位置由 TBPRDHR 寄存器或 CMPBHR 寄存器控制 1：相位控制模式，即边沿位置由 TBPHSHR 寄存器控制	0h
12:11	HRLOADB	R/W	选择 B 路信号影子值装载事件（PWMxB Shadow Value Load Event Select） 该位域用于选择将 CMPBHR 的值装载到活动寄存器的时间事件。 满足以下条件时，将 CMPBHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留	0h
15;13	保留			0h

31.8.15 使能 HRP 校准逻辑寄存器（HRPWR）

偏移地址：0x42

复位类型：SYSRSn

该寄存器仅适用于具有 HRPWM 功能的 PWM 模块。

位/域	名称	R/W	描述	复位值
14:0	保留			0h
15	CALPWRON	R/W	使能 HRP 校准逻辑（HRP Calibration Logic Enable） 该位仅适用于 PWM1。可以通过禁止 HRP 校准逻辑来降低功耗。 0：禁止 1：使能	0h

31.8.16 HRP 步骤数寄存器（HRMSTEP）

偏移地址：0x4C

复位类型：SYSRSn

该寄存器仅允许 16 位访问且仅适用于具有 HRPWM 功能的 PWM 模块。在 32 位访问模式下会导致调试器出现错误。

位/域	名称	R/W	描述	复位值
7:0	HRMSTEP	R/W	HRP 步进数（number of HRP steps） 在使能自动转换条件下，该位域自动将 CMPAHR 寄存器、CMPBHR 寄存器、DBFEDHR 寄存器、DBREDHR 寄存器、TBPRDHR 寄存器或 TBPHSHR 寄存器中的值转换成 HRPWM 输出上的缩放微边沿延迟。 在每次校准结束时，SFO 校准软件会将适当的值写入该位域。	0h
15:8	保留			0h

31.8.17 HRPWM 配置寄存器 2（HRCNFG2）

偏移地址：0x4E

复位类型：SYSRSn

该寄存器仅允许 16 位访问且仅适用于具有 HRPWM 功能的 PWM 模块。在 32 位访问模式下会导致调试器出现错误。

位/域	名称	R/W	描述	复位值
1:0	EDGMODEDB	R/W	选择死区边沿模式（Dead Band Edge Mode Select） 该位域用于选择由 HRP 逻辑控制死区边沿。 00：禁用 HRPWM 功能（默认） 01：HRP 控制死区的上升沿，此时，该上升沿由 DBREDHR 寄存器控制 10：HRP 控制死区的下降沿，此时，该下降沿由 PWM_DBREDHR 寄存器控制 11：HRP 同时控制死区的上升沿和下降沿，此时，两个边沿分别由 DBREDHR 寄存器和 DBFEDHR 寄存器控制	0h
3:2	CTLMODEDBRED	R/W	选择 DBREDHR 装载事件（DBREDHR Load Event Select） 该位域用于选择将 DBREDHR 的值装载到活动寄存器的时间事件。 满足以下条件时，将 DBREDHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留	0h
5:4	CTLMODEDBFED	R/W	选择 DBFEDHR 装载事件（DBFEDHR Load Event Select） 该位域用于选择将 DBFEDHR 的值装载到活动寄存器的时间事件。 满足以下条件时，将 DBFEDHR 的值装载到活动寄存器： 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：保留	0h
15:6	保留			0h

31.8.18 高分辨率周期控制寄存器（HRPCTL）

偏移地址：0x5A

复位类型：SYSRSn

该寄存器仅适用于具有 HRPWM 功能的 PWM 模块。

位/域	名称	R/W	描述	复位值
0	HRPE	R/W	使能高分辨率 PWM 周期（High Resolution Period Enable） 如果需要使用 HRPWM 周期功能，则需要将 CTRMODE 位设置为除向下计数模式之外的模式。 0：禁止，此时 PWM 类似与类型 4 PWM 1：使能，HRPWM 可以同时控制高分辨率的占空比和频率	0h
1	PWMSYNCSEL	R/W	选择 PWMSYNCPER 源（PWMSYNCPER Source Select） 通过该位选择用于驱动 COMP 和 GPDAC 模块的 PWMSYNCPER 信号源。 0：TBCTR=TBPRD 为 PWMSYNCPER 信号源 1：TBCTR=0 为 PWMSYNCPER 信号源	0h
2	TBPHSHRLOADE	R/W	使能高分辨率相位同步（High Resolution Phase Synchronization Enable） 该位可以通过同步输入、软件强制同步脉冲或数字比较事件进行 PWM 模块的高分辨率相位同步，以实现控制多个相同频率的 PWM 模块进行高分辨率相位对齐。 TBCTL[PHSEN]位与 TBPHSHRLOADE 位可以独立工作。如果要使用高分辨率相位和周期控制功能，则需要同时设置 TBCTL[PHSEN]位与 TBPHSHRLOADE 位。当高分辨率周期处于中心对齐计数模式时，如果 TBPHSHR 为 0，则 TBCTL[PHSEN]位与 TBPHSHRLOADE 位必须置位。如果只使能高分辨率占空比，则不需要设置该位。 0：禁止 1：使能，此时，通过使用 TBPHSHR 寄存器内容进行同步。	0h
3	保留			0h
6:4	PWMSYNCSELX	R/W	选择扩展的信号（Extended Signal Select） 000：通过 PWMSYNCSEL 位定义 PWMSYNCPER 信号（默认） 100：计数器的值等于 CMPC 寄存器的值并且向上计数 101：计数器的值等于 CMPC 寄存器的值并且向下计数 110：计数器的值等于 CMPD 寄存器的值并且向上计数 其他：保留	0h
15:7	保留			0h

31.8.19 HRPWM 转换高分辨率余数寄存器（TRREM）

偏移地址：0x5C

复位类型：SYSRSn

该寄存器仅适用于具有 HRPWM 功能的 PWM 模块。在使能同步输入、软件强制同步脉冲或数字比较事件的条件下，该寄存器的低 8 位可以自动通过 TBPHSHR

值进行初始化，用户也可以通过 CPU 写入一个值。在非对称模式下，如果未使用 TRREM[10]，则可强制为 0。

更新 TRREM 寄存器的优先级：

（1） 通过软件或硬件同步将 TBPHSHR 值传输到 TRREM 寄存器

（2） 通过 HRPWM 硬件更新 TRREM 寄存器

（3） 通过 CPU 写入 TRREM 寄存器

位/域	名称	R/W	描述	复位值
10:0	TRREM	R/W	转换器剩余部分(Translator Remainder Portion) 这些位域用于继续跟踪 HRPWM 算法计算的余数部分，该余数部分用于继续跟踪硬件计算的余数部分。 在向上计数模式下，TRREM 初始化为 0。 在中心对齐计数模式下，TRREM 初始化为 256。 在非对称模式下，TRREM[7:0] = TBPHSHR[15:8]，TRREM[10:8] = 000。 在对称模式下，TRREM[7:0] = TBPHSHR[15:8]，TRREM[10:8] = 001。	0h
15:11	保留			0h

31.8.20 全局 PWM 装载控制寄存器（GLDCTL）

偏移地址：0x68

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GLD	R/W	选择全局装载事件（Global Load Event Select） 该位用于控制影子寄存器到活动寄存器的装载行为。 0：所有影子寄存器装载行为按照各自指定的重新装载控制位进行 1：所有影子寄存器装载行为由 GLDMODE 位定义	0h
4:1	GLDMODE	R/W	选择全局装载脉冲（Global Load Pulse Select） 该位域用于选择在影子寄存器到活动寄存器模式重装载模式中期间何时进行全局装载脉冲操作。 0000：在计数器等于 0 时进行装载 0001：在计数器等于周期时进行装载 0010：在计数器等于 0 或计数器等于周期时进行装载 0011：在 SYNCEVT 信号时进行装载，SYNCEVT 信号为 PWMx 同步输入、软件强制同步脉冲、DCAEVT1.sync 和 DCBEVT1.sync 进行逻辑或操作 0100：在 SYNCEVT 信号或计数器等于 0 时进行装载 0101：在 SYNCEVT 信号或计数器等于周期时进行装载 0110：在 SYNCEVT 信号、计数器等于 0 或计数器等于周期时进行装载 1111：在写入 GFRCLD 位时进行装载 其他：保留	0h

位/域	名称	R/W	描述	复位值
5	OSHTMODE	R/W	使能单次装载模式（One Shot Load Mode Enable） 单次装载模式仅在 GLD=1 条件下使用。 0：禁止，所有选择的全局装载脉冲会连续进行影子寄存器到活动寄存器的装载行为 1：使能，直到 OSHTLD=1，否则所有全局装载脉冲一直被阻止	0h
6	保留			0h
9:7	GLDPRD	R/W	选择全局装载选通事件（Global Load Strobe Event Select） 该位域用于选择在发生特定数量的事件后，何时生成装载全局装载脉冲。 000：禁止计数器，不生成脉冲 001：在 GLDCNT=001 时生成脉冲 010：在 GLDCNT=010 时生成脉冲 011：在 GLDCNT=011 时生成脉冲 100：在 GLDCNT=100 时生成脉冲 101：在 GLDCNT=101 时生成脉冲 110：在 GLDCNT=110 时生成脉冲 111：在 GLDCNT=111 时生成脉冲	0h
12:10	GLDCNT	R	选择全局装载事件数（The Number of Global Load Events Select） 该位域选择已发生的选定事件数量。 000：无事件发生 001：发生 1 个事件 010：发生 2 个事件 011：发生 3 个事件 100：发生 4 个事件 101：发生 5 个事件 110：发生 6 个事件 111：发生 7 个事件	0h
15:13	保留			0h

31.8.21 全局 PWM 装载事件配置寄存器（GLDCFG）

偏移地址：0x6A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TBPRD_TBPRDHR	R/W	配置 TBPRD_TBPRDHR 全局装载事件（TBPRD_TBPRDHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h

位/域	名称	R/W	描述	复位值
1	CMPA_CMPAHR	R/W	配置 CMPA_CMPAHR 全局装载事件 （CMPA_CMPAHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
2	CMPB_CMPBHR	R/W	配置 CMPB_CMPBHR 全局装载事件 （CMPB_CMPBHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
3	CMPC	R/W	配置 CMPC 全局装载事件 （CMPC Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
4	CMPD	R/W	配置 CMPD 全局装载事件 （CMPD Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
5	DBRED_DBREDHR	R/W	配置 DBRED_DBREDHR 全局装载事件 （DBRED_DBREDHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
6	DBFED_DBFEDHR	R/W	配置 DBFED_DBFEDHR 全局装载事件 （DBFED_DBFEDHR Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
7	DBCTL	R/W	配置 DBCTL 全局装载事件 （DBCTL Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h

位/域	名称	R/W	描述	复位值
8	AQCTLA_AQCTLA2	R/W	配置 AQCTLA_AQCTLA2 全局装载事件 （AQCTLA_AQCTLA2 Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
9	AQCTLB_AQCTLB2	R/W	配置 AQCTLB_AQCTLB2 全局装载事件 （AQCTLB_AQCTLB2 Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
10	AQCSFRC	R/W	配置 AQCSFRC 全局装载事件 （AQCSFRC Global Load Event Configure） 0：即使 GLDCTL[GLD]=1，寄存器使用本地重新装载配置 1：当 GLDCTL[GLD]=1 和设置该位时，寄存器将使用全局装载配置	0h
15:11	保留			0h

31.8.22 PWMx 链接寄存器（PWMXLINK）

偏移地址：0x70

复位类型：SYSRSn

该寄存器用于配置 PWMx 模块之间的链接关系，且每个 PWMx 模块的默认复位值不同。为了防止无意中链接模块，复位值将每个 PWMx 模块链接到自身。

位/域	名称	R/W	描述	复位值
3:0	TBPRDLINK	R/W	TBPRD_TBPRDHR 链接 （TBPRD and TBPRDHR Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下位选择，写入 TBPRD:TBPRDHR 寄存器同时写入当前 PWMx 的 TBPRD_TBPRDHR 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	X

位/域	名称	R/W	描述	复位值
7:4	CMPALINK	R/W	CMPA_CMPAHR 链接 （CMPA and CMPAHR Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下选择，写入 CMPA_CMPAHR 寄存器同时写入当前 PWMx 的 CMPA 和 CMPAHR 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	X
11:8	CMPBLINK	R/W	CMPB_CMPBHR 链接 （CMPB and CMPBHR Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下选择，写入 CMPB_CMPBHR 寄存器同时写入当前 PWMx 的 CMPB 和 CMPBHR 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	X
15:12	CMPCLINK	R/W	CMPC 链接（CMPC Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下选择，写入 CMPC 寄存器同时写入当前 PWMx 的 CMPC 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	X
19:16	CMPDLINK	R/W	CMPD 链接（CMPD Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下选择，写入 CMPD 寄存器同时写入当前 PWMx 的 CMPD 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	X
27:20	保留			0h
31:28	GLDCTL2LINK	R/W	GLDCTL2 链接（GLDCTL2 Link） 该位域用于在所选的 PWMx 模块在进行写操作，并根据以下选择，写入 GLDCTL2 寄存器同时写入当前 PWMx 的 GLDCTL2 寄存器。 0000：PWM1 0001：PWM2 ... 直到最后一个 PWM 实例 其他：保留	0h

31.8.23 输出 A 的动作限定器控制寄存器（AQCTLA）

偏移地址：0x80

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	ZRO	R/W	配置当 TBCTR=0 时动作（Action When TBCTR = 0 Configure） 00：无操作，PWMxA 输出保持不变 01：清零，强制 PWMxA 输出为低电平 10：设置，强制 PWMxA 输出为高电平 11：翻转，强制 PWMxA 低电平输出信号切换为高电平，高电平输出信号切换为低电平 注意：在中心对齐计数模式下，当计数器等于 0 时，计数器方向定义为向上计数	0h
3:2	PRD	R/W	配置当 TBCTR=TBPRD 时动作（Action When TBCTR =TBPRD Configure） 参考 ZRO 描述	0h
5:4	CAU	R/W	向上计数时，配置当 TBCTR=CMPA 时动作（Action When TBCTR = CMPA On Up Count Configure） 参考 ZRO 描述	0h
7:6	CAD	R/W	向下计数时，配置当 TBCTR=CMPA 时动作（Action When TBCTR = CMPA On Down Count Configure） 参考 ZRO 描述	0h
9:8	CBU	R/W	向上计数时，配置当 TBCTR=CMPB 时动作（Action When TBCTR= CMPB On Up Count Configure） 参考 ZRO 描述	0h
11:10	CBD	R/W	向下计数时，配置当 TBCTR=CMPB 时动作（Action When TBCTR = CMPB On Down Count Configure） 参考 ZRO 描述	0h
15:12	保留			0h

31.8.24 输出 A 的附加动作限定器控制寄存器（AQCTLA2）

偏移地址：0x82

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	T1U	R/W	向上计数时，配置发生 T1 事件时动作（Action when event occurs on T1 in UP-Count Configure） 00：无操作，PWMxA 输出保持不变 01：清零，强制 PWMxA 输出为低电平 10：设置，强制 PWMxA 输出为高电平 11：翻转，强制 PWMxA 低电平输出信号切换为高电平，高电平输出信号切换为低电平 注意：在中心对齐计数模式下，当计数器等于 0 时，计数器方向定义为向上计数	0h
3:2	T1D	R/W	向下计数时，配置发生 T1 事件时动作（Action when event occurs on T1 in DOWN-Count Configure） 参考 T1U 描述	0h

位/域	名称	R/W	描述	复位值
5:4	T2U	R/W	向上计数时，配置发生 T2 事件时动作（Action when event occurs on T2 in UP-Count Configure） 参考 T1U 描述	0h
7:6	T2D	R/W	向下计数时，配置发生 T2 事件时动作（Action when event occurs on T2 in DOWN-Count Configure） 参考 T1U 描述	0h
15:8	保留			0h

31.8.25 输出 B 的动作限定器控制寄存器（AQCTLB）

偏移地址：0x84

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	ZRO	R/W	配置当 TBCTR=0 时动作（Action When TBCTR = 0 Configure） 00：无操作，PWMxB 输出保持不变 01：清零，强制 PWMxB 输出为低电平 10：设置，强制 PWMxB 输出为高电平 11：翻转，强制 PWMxB 低电平输出信号切换为高电平，高电平输出信号切换为低电平 注意：在中心对齐计数模式下，当计数器等于 0 时，计数器方向定义为向上计数	0h
3:2	PRD	R/W	配置当 TBCTR=TBPRD 时动作（Action When TBCTR =TBPRD Configure） 参考 ZRO 描述	0h
5:4	CAU	R/W	向上计数时，配置当 TBCTR=CMPA 时动作（Action When TBCTR = CMPA On Up Count Configure） 参考 ZRO 描述	0h
7:6	CAD	R/W	向下计数时，配置当 TBCTR=CMPA 时动作（Action When TBCTR = CMPA On Down Count Configure） 参考 ZRO 描述	0h
9:8	CBU	R/W	向上计数时，配置当 TBCTR=CMPB 时动作（Action When TBCTR = CMPB On Up Count Configure） 参考 ZRO 描述	0h
11:10	CBD	R/W	向下计数时，配置当 TBCTR=CMPB 时动作（Action When TBCTR = CMPB On Down Count Configure） 参考 ZRO 描述	0h
15:12	保留			0h

31.8.26 输出 B 的附加动作限定器控制寄存器（AQCTLB2）

偏移地址：0x86

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	T1U	R/W	向上计数时，配置发生 T1 事件时动作（Action when event occurs on T1 in UP-Count ）Configure 00：无操作，PWMxB 输出保持不变 01：清零，强制 PWMxB 输出为低电平 10：设置，强制 PWMxB 输出为高电平 11：翻转，强制 PWMxB 低电平输出信号切换为高电平，高电平输出信号切换为低电平 注意：在中心对齐计数模式下，当计数器等于 0 时，计数器方向定义为向上计数	0h
3:2	T1D	R/W	向下计数时，配置发生 T1 事件时动作（Action when event occurs on T1 in DOWN-Count Configure） 参考 T1U 描述	0h
5:4	T2U	R/W	向上计数时，配置发生 T2 事件时动作（Action when event occurs on T2 in UP-Count Configure） 参考 T1U 描述	0h
7:6	T2D	R/W	向下计数时，配置发生 T2 事件时动作（Action when event occurs on T2 in DOWN-Count Configure） 参考 T1U 描述	0h
15:8	保留			0h

31.8.27 动作限定器软件强制寄存器（AQSFRC）

偏移地址：0x8E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	ACTSFA	R/W	配置当发生一次软件强制事件后，A 路输出的动作（Action When One-Time Software Force A Is Invoked Configure） 00：无操作，A 路输出保持不变 01：清除，设置 A 路输出为低电平 10：设置，设置 A 路输出为高电平 11：翻转，将 A 路低电平输出信号切换为高电平或高电平输出信号切换为低电平 注意：这些动作不受计数器方向的影响。	0h
2	OTSFA	R/S	使能 A 路输出一次软件强制事件（One-Time Software Forced Event on Output A Enable） 使能 A 路输出一次软件强制事件后，将自动清除该位位域。该强制事件为一次性，A 路输出的后续事件可以覆盖该强制事件。 0：无效，读取返回 0。 1：使能	0h

位/域	名称	R/W	描述	复位值
4:3	ACTSFB	R/W	配置当发生一次软件强制事件后，B 路输出的动作（Action When One-Time Software Force B Is Invoked Configure） 00：无操作，B 路输出保持不变 01：清除，设置 B 路输出为低电平 10：设置，设置 B 路输出为高电平 11：翻转，将 B 路低电平输出信号切换为高电平或高电平输出信号切换为低电平 注意：这些动作不受计数器方向的影响。	0h
5	OTSFB	R/S	使能 B 路输出一次软件强制事件（One-Time Software Forced Event on Output B Enable） 使能 B 路输出一次软件强制事件后，将自动清除该位位域。该强制事件为一次性，B 路输出的后续事件可以覆盖该强制事件。 0：无效，读取返回 0 1：使能	0h
7:6	RLDCSF	R/W	选择 AQCSFRC 重装载模式（AQCSFRC Reload Mode Select） 该位域用于选择何时将 AQCSFRC 影子寄存器装载到活动寄存器的值。 00：在 TBCTR=0 时装载 01：在 TBCTR=TBPRD 时装载 10：在 TBCTR=0 或 TBCTR=TBPRD 时装载 11：立即装载，不使用影子寄存器，CPU 直接访问活动寄存器	0h
15:8	保留			0h

31.8.28 动作限定器连续软件强制寄存器（AQCSFRC）

偏移地址：0x92

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	CSFA	R/W	配置 A 路输出连续软件强制（Continuous Software Force on Output A Configure） 立即模式：在该模式下，连续软件强制在下一个 TBCLK 边沿生效。 影子模式：通过使用 RLDCSF 位配置影子模式。在该模式下，连续软件强制在影子装载到活动寄存器后的下一个 TBCLK 边沿生效。 00：禁止软件强制且对 A 路输出无影响 01：将 A 路输出连续软件强制设置为低电平 10：将 A 路输出连续软件强制设置为高电平 11：禁止软件强制且对 A 路输出无影响	0h

位/域	名称	R/W	描述	复位值
3:2	CSFB	R/W	配置 B 路输出连续软件强制（Continuous Software Force on Output B Configure） 立即模式：在该模式下，连续软件强制在下一个 TBCLK 边沿生效。 影子模式：通过使用 RLDCSF 位配置影子模式。在该模式下，连续软件强制在影子装载到活动寄存器后的下一个 TBCLK 边沿生效。 00：禁止软件强制且对 B 路输出无影响 01：将 B 路输出连续软件强制设置为低电平 10：将 B 路输出连续软件强制设置为高电平 11：禁止软件强制且对 B 路输出无影响	0h
15:4	保留			0h

31.8.29 死区发生器上升沿延迟高分辨率寄存器（DBREDHR）

偏移地址：0xA0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
8:0	保留			0h
15:9	DBREDHR	R/W	死区上升沿延迟高分辨率（Dead Band Rising Edge Delay High Resolution）	0h

31.8.30 死区发生器上升沿延迟寄存器（DBRED）

偏移地址：0xA2

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
13:0	DBRED	R/W	死区上升沿延迟值（Dead Band Rising Edge Delay Value）	0h
15:14	保留			0h

31.8.31 死区发生器下降沿延迟高分辨率寄存器（DBFEDHR）

偏移地址：0xA4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
8:0	保留			0h
15:9	DBFEDHR	R/W	死区下降沿延迟高分辨率（Dead Band Falling Edge Delay High Resolution）	0h

31.8.32 死区发生器下降沿延迟寄存器（DBFED）

偏移地址：0xA6

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
13:0	DBFED	R/W	死区下降沿延迟值（Dead Band Falling Edge Delay Value）	0h
15:14	保留			0h

31.8.33 时基相位偏移寄存器（TBPHS）

偏移地址：0xC0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	TBPHSHR	R/W	高分辨率时基相位偏移（Time Base Phase Offset High Resolution） 忽略 TBPHSHR 寄存器的低 8 位，忽略写入操作，读取操作返回 0。 必须通过 TRREM 寄存器来模拟 TBPHSHR 功能，而不是使用 TBPHSHR。	0h
31:16	TBPHS	R/W	时基相位偏移（Time Base Phase Offset） 该寄存器用于调整 PWM 相位，即设置所选 PWM 与提供同步输入信号的时基计数器之间的相位偏移量。 如果 TBCTL[PHSEN]=0，则 TBCTR 不会装载相位值并且忽略同步事件。 如果 TBCTL[PHSEN]=1，则在发生同步事件时（可以由软件强制同步或 PWMx 同步输入信号产生），TBCTR 装载相位值。	0h

31.8.34 高分辨率时基周期寄存器（TBPRDHR）

偏移地址：0xC4

复位类型：SYSRSn

忽略该寄存器的低 8 位，忽略写入操作，读取操作返回 0。该寄存器仅适用于高分辨率时基周期控制的 PWM 模块。仅在使能高分辨率时基周期功能时使用该寄存器。

位/域	名称	R/W	描述	复位值
15:0	TBPRDHR	R/W	高分辨率时基周期（Time Base Period High Resolution） 该寄存器用于存储高分辨率时基周期值的高 8 位，且不受 PRDLD 位的影响。对 TBPRDHR 进行读取和写入操作都是对影子寄存器进行的。	0h

31.8.35 时基周期寄存器（TBPRD）

偏移地址：0xC6

复位类型：SYSRSn

该寄存器的影子寄存器可以通过 TBCTL[PRDLD]位来使能/禁止，在默认情况下，该寄存器用于使能影子寄存器。

位/域	名称	R/W	描述	复位值
15:0	TBPRD	R/W	时基周期（Time Base Period） 该寄存器确定 TBCTR 周期，从而设置 PWM 频率。 如果 TBCTL[PRDLD]=0，则使能影子寄存器，任何读取/写入操作都将自动进入影子寄存器。此时，在 TBCTR=0 时，活动寄存器从影子寄存器中装载。 如果 TBCTL[PRDLD]=1，则禁止影子寄存器，任何读取/写入操作都将自动进入活动寄存器。 活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.36 计数比较模块 A 寄存器（CMPA）

偏移地址：0xD4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CMPAHR	R/W	计数比较 A 高分辨率扩展（Counter Compare A High Resolution Extension） 忽略 CMPAHR 寄存器的低 8 位。该寄存器高 8 位用于存储和操作计数器比较 A 值的高分辨率部分。 CMPA 寄存器可以通过单个 32-位的读取/写入操作来访问。与 CMPA 寄存器一样，通过 SHDWAMODE 位来使能影子寄存器，详细内容请参见 CMPA 寄存器描述。	0h
31:16	CMPA	R/W	计数比较模块 A（Counter Compare A） CMPA 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较模块 A 事件。将该事件发送到动作限定器进行限定和转换为一个或多个动作。 AQCTLA 寄存器和 AQCTLB 寄存器的配置决定这些动作可以应用于 PWMxA 输出信号或 PWMxB 输出信号。AQCTLA 寄存器和 AQCTLB 寄存器可以定义的动作如下所示： 无操作：忽略 TBCTR 等于计数比较模块 A 事件 清除：设置 PWMxA 或 PWMxB 输出信号为低电平 设置：设置 PWMxA 或 PWMxB 输出信号为高电平 翻转：将 PWMxA 或 PWMxB 输出信号的状态进行翻转 通过 SHDWAMODE 位来使能 CMPA 影子寄存器，在默认情况下使能影子模式。 当 SHDWAMODE=0 时，使能 CMPA 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADAMODE 位可以确定影子寄存器何时装载到活动寄存器中。 当 SHDWAMODE=1 时，禁止 CMPA 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。 在写入之前，可以根据 SHDWAFULL 位判断 CMPA 影子寄存器当前状态是否已满。 在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.37 计数比较模块 B 寄存器（CMPB）

偏移地址：0xD8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CMPBHR	R/W	计数比较 B 高分辨率扩展（Counter Compare B High Resolution Extension） 忽略 CMPBHR 寄存器的低 8 位。	0h

位/域	名称	R/W	描述	复位值
31:16	CMPB	R/W	计数比较模块 B（Counter Compare B） CMPB 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较模块 B 事件。将该事件发送到动作限定器进行限定和转换为一个或多个动作。 AQCTLA 寄存器和 AQCTLB 寄存器的配置决定这些动作可以应用于 PWMxA 输出信号或 PWMxB 输出信号。AQCTLA 寄存器和 AQCTLB 寄存器可以定义的动作如下所示： 无操作：忽略 TBCTR 等于计数比较模块 A 事件 清除：设置 PWMxA 或 PWMxB 输出信号为低电平 设置：设置 PWMxA 或 PWMxB 输出信号为高电平 翻转：将 PWMxA 或 PWMxB 输出信号的状态进行翻转 通过 SHDWBMODE 位来使能 CMPB 影子寄存器，在默认情况下使能影子模式。 当 SHDWBMODE=0 时，使能 CMPB 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADBMODE 位可以确定影子寄存器何时装载到活动寄存器中。 当 SHDWBMODE=1 时，禁止 CMPB 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。 在写入之前，可以根据 SHDWBFULL 位判断 CMPB 影子寄存器当前状态是否已满。 在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.38 计数比较模块 C 寄存器（CMPC）

偏移地址：0xDE

复位类型：SYSRSn

该寄存器的链接功能始终为 16 位访问。

位/域	名称	R/W	描述	复位值
15:0	CMPC	R/W	计数比较模块 C（Counter Compare C） CMPC 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较 C 事件。 通过 SHDWCMODE 位来使能 CMPC 影子寄存器，在默认情况下使能影子模式。 当 SHDWCMODE=0 时，使能 CMPC 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADCMODE 位可以确定影子寄存器何时装载到活动寄存器中。 当 SHDWCMODE=1 时，禁止 CMPC 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。 在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.39 计数比较模块 D 寄存器（CMPD）

偏移地址：0xE2

复位类型：SYSRSn

该寄存器的链接功能始终为 16-位访问。

位/域	名称	R/W	描述	复位值
15:0	CMPD	R/W	计数比较模块 D（Counter Compare D） CMPD 寄存器用于与 TBCTR 进行连续比较。如果两个值相等，则计数器比较模块就会生成一个 TBCTR 等于计数比较 D 事件。 通过 SHDWDMODE 位来使能 CMPD 影子寄存器，在默认情况下使能影子模式。 当 SHDWDMODE=0 时，使能 CMPD 影子寄存器，此时，任何写入或读取操作都会自动进入到影子寄存器。在该情况下，根据 LOADDMODE 位可以确定影子寄存器何时装载到活动寄存器中。 当 SHDWDMODE=1 时，禁止 CMPD 影子寄存器，此时，任何写入或读取操作将直接进入到活动寄存器。 在任何一种模式下，活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.40 全局 PWM 装载控制寄存器 2（GLDCTL2）

偏移地址：0xE8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	OSHTLD	R/S	使能单次模式下的重装载事件（Reload Event in One Shot Mode Enable） 该位允许一个装载脉冲事件并阻止后续的时间。 0：无效，读取返回 0 1：使能单次锁存条件。在选择的装载脉冲发生时，会发生一次影子到活动的重装载并清除锁存器。	0h
1	GFRCLD	R/S	使能单次模式下的强制装载事件（Force Load Event in One Shot Mode Enable） 该位用于在全局装载模式下，通过测试或软件强制来装载事件。 0：无效，读取返回 0 1：在事件预分频计数器的输入位置上强制一个装载事件	0h
15:2	保留			0h

31.8.41 软件谷模式延迟寄存器（SWVDELVAL）

偏移地址：0xEE

复位类型：SYSRSn

用户通过设置该寄存器的值，可以根据 VDELAYDIV 位定义硬件计算延迟值。

位/域	名称	R/W	描述	复位值
15:0	SWVDELVAL	R/W	软件谷延迟值（Software Valley Delay Value）	0h

31.8.42 跳闸区选择寄存器（TZSEL）

偏移地址：0x100

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CBC1	R/W	选择跳闸区 1 为逐周期跳闸源（TZ1 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ1。 0：不是 1：是	0h
1	CBC2	R/W	选择跳闸区 2 为逐周期跳闸源（TZ2 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ2。 0：不是 1：是	0h
2	CBC3	R/W	选择跳闸区 3 为逐周期跳闸源（TZ3 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ3。 0：不是 1：是	0h
3	CBC4	R/W	选择跳闸区 4 为逐周期跳闸源（TZ4 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ4。 0：不是 1：是	0h
4	CBC5	R/W	选择跳闸区 5 为逐周期跳闸源（TZ5 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ5。 0：不是 1：是	0h
5	CBC6	R/W	选择跳闸区 6 为逐周期跳闸源（TZ6 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 TZ6。 0：不是 1：是	0h
6	DCAEVT2	R/W	选择数字比较器 A 输出事件 2 为逐周期跳闸源（DCAEVT2 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 DCAEVT2。 0：不是 1：是	0h
7	DCBEVT2	R/W	选择数字比较器 B 输出事件 2 为逐周期跳闸源（DCBE2 as a CBC source Select） 通过该位选择逐周期跳闸源是否为 DCBE2。 0：不是 1：是	0h
8	OSHT1	R/W	选择跳闸区 1 为单次跳闸源（TZ1 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ1。 0：不是 1：是	0h

位/域	名称	R/W	描述	复位值
9	OSHT2	R/W	选择跳闸区 2 为单次跳闸源（TZ2 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ2。 0：不是 1：是	0h
10	OSHT3	R/W	选择跳闸区 3 为单次跳闸源（TZ3 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ3。 0：不是 1：是	0h
11	OSHT4	R/W	选择跳闸区 4 为单次跳闸源（TZ4 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ4。 0：不是 1：是	0h
12	OSHT5	R/W	选择跳闸区 5 为单次跳闸源（TZ5 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ5。 0：不是 1：是	0h
13	OSHT6	R/W	选择跳闸区 6 为单次跳闸源（TZ6 as a OST source Select） 通过该位选择单次跳闸源是否为 TZ6。 0：不是 1：是	0h
14	DCAEVT1	R/W	选择数字比较器 A 输出事件 1 为单次跳闸源（DCAEVT1 as a OSTsource Select） 通过该位选择单次跳闸源是否为 DCAEVT1。 0：不是 1：是	0h
15	DCBEVT1	R/W	选择数字比较器 B 输出事件 1 为单次跳闸源（DCBE1 as a OST source Select） 通过该位选择单次跳闸源是否为 DCBE1。 0：不是 1：是	0h

31.8.43 跳闸区数字比较器配置寄存器（TZDCSEL）

偏移地址：0x104

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	DCAEVT1	R/W	配置数字比较器 A 输出事件 1（Digital Compare Output A Event 1 Configure） 000：禁止数字比较器 A 输出事件 1 001：在 DCBH 信号为低电平时触发 010：在 DCBH 信号为高电平时触发 011：在 DCBL 信号为低电平时触发 100：在 DCBL 信号为高电平时触发 101：在 DCBL 信号为高电平，DCBH 信号为低电平时触发 110：保留 111：保留	0h
5:3	DCAEVT2	R/W	配置数字比较器 A 输出事件 2（Digital Compare Output A Event 2 Configure） 参考 DCAEVT1 描述	0h
8:6	DCBEVT1	R/W	配置数字比较器 B 输出事件 1（Digital Compare Output B Event 1 Configure） 参考 DCAEVT1 描述	0h
11:9	DCBEVT2	R/W	配置数字比较器 B 输出事件 2（Digital Compare Output B Event 2 Configure） 参考 DCAEVT1 描述	0h
15:12	保留			0h

31.8.44 跳闸区控制寄存器（TZCTL）

偏移地址：0x108

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	TZA	R/W	配置跳闸事件发生时 PWMxA 动作（Trip Event Occurs PWMxA Action Configure） 该位域定义在发生跳闸事件时对 PWMxA 输出采取的动作，通过 TZSEL 寄存器可以确定哪些跳闸区引脚可以引发事件。 00：PWMxA 为高阻态 01：强制 PWMxA 输出为高电平 10：强制 PWMxA 输出为低电平 11：PWMxA 保持原状态	0h
3:2	TZB	R/W	配置跳闸事件发生时 PWMxB 动作（Trip Event Occurs PWMxB Action Configure） 该位域定义在发生跳闸事件时对 PWMxB 输出采取的动作，通过 TZSEL 寄存器可以确定哪些跳闸区引脚可以引发事件。 00：PWMxB 为高阻态 01：强制 PWMxB 输出为高电平 10：强制 PWMxB 输出为低电平 11：PWMxB 保持原状态	0h

位/域	名称	R/W	描述	复位值
5:4	DCAEVT1	R/W	配置 PWMxA 的数字比较器 A 输出事件 1 动作（Digital Compare Output A Event 1 Action On PWMxA Configure） 00：PWMxA 为高阻态 01：强制 PWMxA 输出为高电平 10：强制 PWMxA 输出为低电平 11：禁止跳闸动作	0h
7:6	DCAEVT2	R/W	配置 PWMxA 的数字比较器 A 输出事件 2 动作（Digital Compare Output A Event 2 Action On PWMxA Configure） 参考 DCAEVT1 描述	0h
9:8	DCBEVT1	R/W	配置 PWMxA 的数字比较器 B 输出事件 1 动作（Digital Compare Output B Event 1 Action On PWMxA Configure） 00：PWMxB 为高阻态 01：强制 PWMxB 输出为高电平 10：强制 PWMxB 输出为低电平 11：禁止跳闸动作	0h
11:10	DCBEVT2	R/W	配置 PWMxA 的数字比较器 B 输出事件 2 动作（Digital Compare Output B Event 2 Action On PWMxA Configure） 参考 DCBEVT1 描述	0h
15:12	保留			0h

31.8.45 跳闸区控制寄存器 2（TZCTL2）

偏移地址：0x10A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	TZAU	R/W	计数方向向上时，配置跳闸事件发生时 PWMxA 动作（Trip Event Occurs PWMxA Action while Count direction is UP Configure） 000：PWMxA 输出为高阻态 001：强制 PWMxA 输出为高电平 010：强制 PWMxA 输出为低电平 011：将 PWMxA 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作	0h
5:3	TZAD	R/W	计数方向向下时，配置跳闸事件发生时 PWMxA 动作（Trip Event Occurs PWMxA Action while Count direction is Down Configure） 参考 TZAU 描述	0h

位/域	名称	R/W	描述	复位值
8:6	TZBU	R/W	计数方向向上时，配置跳闸事件发生时 PWMxB 动作（Trip Event Occurs PWMxB Action while Count direction is UP Configure） 000：PWMxB 输出为高阻态 001：强制 PWMxB 输出为高电平 010：强制 PWMxB 输出为低电平 011：将 PWMxB 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作	0h
11:9	TZBD	R/W	计数方向向下时，配置跳闸事件发生时 PWMxB 动作（Trip Event Occurs PWMxB Action while Count direction is Down Configure） 参考 TZBU 描述	0h
14:12	保留			0h
15	ETZE	R/W	使能 TZCTL2（TZCTL2 Enable） 0：通过 TZCTL 寄存器中定义的跳闸动作 1：通过 TZCTL2 寄存器、TZCTLDCA 寄存器和 TZCTLDCB 寄存器中定义的跳闸动作，忽略 TZCTL 寄存器中的设置。	0h

31.8.46 跳闸区控制数字比较器 A 寄存器（TZCTLDCA）

偏移地址：0x10C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	DCAEVT1U	R/W	计数方向向上时，配置 PWMxA 的数字比较器 A 输出事件 1 动作（Digital Compare Output A Event 1 Action On PWMxA Configure while Count direction is UP Configure） 000：PWMxA 输出为高阻态 001：强制 PWMxA 输出为高电平 010：强制 PWMxA 输出为低电平 011：将 PWMxA 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作	0h
5:3	DCAEVT1D	R/W	计数方向向下时，配置 PWMxA 的数字比较器 A 输出事件 1 动作（Digital Compare Output A Event 1 Action On PWMxA Configure while Count direction is Down Configure） 参考 DCAEVT1U 描述	0h
8:6	DCAEVT2U	R/W	计数方向向上时，配置 PWMxA 的数字比较器 A 输出事件 2 动作（Digital Compare Output A Event 1 Action On PWMxA Configure while Count direction is UP Configure） 参考 DCAEVT1U 描述	0h

位/域	名称	R/W	描述	复位值
11:9	DCAEVT2D	R/W	计数方向向下时，配置 PWMxA 的数字比较器 A 输出事件 2 动作（Digital Compare Output A Event 2 Action On PWMxA Configure while Count direction is Down Configure） 参考 DCAEVT1U 描述	0h
15:12	保留			0h

31.8.47 跳闸区控制数字比较器 B 寄存器（TZCTLDCB）

偏移地址：0x10E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	DCBEVT1U	R/W	计数方向向上时，配置 PWMxB 的数字比较器 B 输出事件 1 动作（Digital Compare Output B Event 1 Action On PWMxB Configure while Count direction is UP Configure） 000：PWMxB 输出为高阻态 001：强制 PWMxB 输出为高电平 010：强制 PWMxB 输出为低电平 011：将 PWMxB 输出状态进行翻转 100：保留 101：保留 110：保留 111：禁止跳闸动作	0h
5:3	DCBEVT1D	R/W	计数方向向下时，配置 PWMxB 的数字比较器 B 输出事件 1 动作（Digital Compare Output B Event 1 Action On PWMxB Configure while Count direction is Down Configure） 参考 DCBEVT1U 描述	0h
8:6	DCBEVT2U	R/W	计数方向向上时，配置 PWMxB 的数字比较器 B 输出事件 2 动作（Digital Compare Output B Event 1 Action On PWMxB Configure while Count direction is UP Configure） 参考 DCBEVT1U 描述	0h
11:9	DCBEVT2D	R/W	计数方向向下时，配置 PWMxB 的数字比较器 B 输出事件 2 动作（Digital Compare Output B Event 2 Action On PWMxB Configure while Count direction is Down Configure） 参考 DCBEVT1U 描述	0h
15:12	保留			0h

31.8.48 跳闸区中断使能寄存器（TZEINT）

偏移地址：0x11A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h

位/域	名称	R/W	描述	复位值
1	CBC	R/W	使能 CBC 中断（Trip-zone Cycle-by-Cycle Interrupt Enable） 逐周期跳闸事件将会导致 NVIC 中断。 0：禁止 1：使能	0h
2	OST	R/W	使能 OST 中断（Trip-zone One-Shot Interrupt Enable） 单次跳闸事件将会导致 NVIC 中断。 0：禁止 1：使能	0h
3	DCAEVT1	R/W	使能数字比较器 A 输出事件 1 中断（Digital Compare Output A Event 1 Interrupt Enable） 0：禁止 1：使能	0h
4	DCAEVT2	R/W	使能数字比较器 A 输出事件 2 中断（Digital Compare Output A Event 2 Interrupt Enable） 0：禁止 1：使能	0h
5	DCBEVT1	R/W	使能数字比较器 B 输出事件 1 中断（Digital Compare Output B Event 1 Interrupt Enable） 0：禁止 1：使能	0h
6	DCBEVT2	R/W	使能数字比较器 A 输出事件 2 中断（Digital Compare Output B Event 2 Interrupt Enable） 0：禁止 1：使能	0h
15:7	保留			0h

31.8.49 跳闸区标志寄存器（TZFLG）

偏移地址：0x126

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R	跳闸中断锁存状态（Trip Interrupt Latched Status） 未清除该位之前，不会再生成 NVIC 中断。当设置 CBC 位或 OST 位前设置此位，将会生成另一个中断脉冲。清除所有状态标志位可以防止进一步的中断。 通过将适当的值写入 TZCLR 寄存器来清除此位。 0：未产生中断 1：因为跳闸条件而产生 NVIC 中断	0h

位/域	名称	R/W	描述	复位值
1	CBC	R	逐周期跳闸事件锁存状态（Cycle-By-Cycle Trip Event Latched Status） 如果设置了该位，则会一直保持设置状态，直到手动清除该位。当清除此位但是仍然存在逐周期跳闸事件时，此位就会立即再次设置。当 TBCTR=0 时，若跳闸条件不再存在，则自动清除信号条件。无论在周期中任一位置，只有 TBCTR=0 时，才能清除信号条件。 通过将适当的值写入 TZCLR 寄存器来清除此位。 0：未发生逐周期跳闸事件 1：发生逐周期跳闸事件，该事件是由于选择一个信号作为逐周期跳闸源引起的	0h
2	OST	R	单次跳闸事件锁存状态（A One-Shot Trip Event Latched Status） 通过将适当的值写入 TZCLR 寄存器来清除此位。 0：未发生单次跳闸事件 1：在单次跳闸源引脚上发生单次跳闸事件	0h
3	DCAEVT1	R	数字比较器 A 输出事件 1 锁存状态（Digital Compare Output A Event 1 Latched Status） 0：DCAEVT1 未发生跳闸事件 1：DCAEVT1 发生跳闸事件	0h
4	DCAEVT2	R	数字比较器 A 输出事件 2 锁存状态（Digital Compare Output A Event 2 Latched Status） 0：DCAEVT2 未发生跳闸事件 1：DCAEVT2 发生跳闸事件	0h
5	DCBEVT1	R	数字比较器 B 输出事件 1 锁存状态（Digital Compare Output B Event 1 Latched Status） 0：DCBEVT1 未发生跳闸事件 1：DCBEVT1 发生跳闸事件	0h
6	DCBEVT2	R	数字比较器 B 输出事件 2 锁存状态（Digital Compare Output B Event 2 Latched Status） 0：DCBEVT2 未发生跳闸事件 1：DCBEVT2 发生跳闸事件	0h
15:7	保留			0h

31.8.50 跳闸区 CBC 标志寄存器（TZCBCFLG）

偏移地址：0x128

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CBC1	R	CBC1 锁存标志（CBC1 Latch Flag） 0：CBC1 未发生跳闸事件 1：CBC1 发生跳闸事件	0h
1	CBC2	R	CBC2 锁存标志（CBC2 Latch Flag） 0：CBC2 未发生跳闸事件 1：CBC2 发生跳闸事件	0h
2	CBC3	R	CBC3 锁存标志（CBC3 Latch Flag） 0：CBC3 未发生跳闸事件 1：CBC3 发生跳闸事件	0h

位/域	名称	R/W	描述	复位值
3	CBC4	R	CBC4 锁存标志（CBC4 Latch Flag） 0：CBC4 未发生跳闸事件 1：CBC4:发生跳闸事件	0h
4	CBC5	R	CBC5 锁存标志（CBC5 Latch Flag） 0：CBC5 未发生跳闸事件 1：CBC5 发生跳闸事件	0h
5	CBC6	R	CBC6 锁存标志（CBC6 Latch Flag） 0：CBC6 未发生跳闸事件 1：CBC6 发生跳闸事件	0h
6	DCAEVT2	R	数字比较器 A 输出事件 2 触发锁存标志（Digital Compare A Output Event 2 Trip Latch Flag） 0：DCAEVT2 未发生跳闸事件 1：DCAEVT2 发生跳闸事件	0h
7	DCBEVT2	R	数字比较器 B 输出事件 2 触发锁存标志 （Digital Compare B Output Event 2 Trip Latch Flag） 0：DCBEVT2 未发生跳闸事件 1：DCBEVT2 发生跳闸事件	0h
15:8	保留			0h

31.8.51 跳闸区 OST 标志寄存器（TZOSTFLG）

偏移地址：0x12A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	OST1	R	OST1 锁存标志（OST1 Latch Flag） 0：OST1 未发生跳闸事件 1：OST1 发生跳闸事件	0h
1	OST2	R	OST2 锁存标志（OST2 Latch Flag） 0：OST2 未发生跳闸事件 1：OST2 发生跳闸事件	0h
2	OST3	R	OST3 锁存标志（OST3 Latch Flag） 0：OST3 未发生跳闸事件 1：OST3 发生跳闸事件	0h
3	OST4	R	OST4 锁存标志（OST4 Latch Flag） 0：OST4 未发生跳闸事件 1：OST4 发生跳闸事件	0h
4	OST5	R	OST5 锁存标志（OST5 Latch Flag） 0：OST5 未发生跳闸事件 1：OST5 发生跳闸事件	0h
5	OST6	R	OST6 锁存标志（OST6 Latch Flag） 0：OST6 未发生跳闸事件 1：OST6 发生跳闸事件	0h
6	DCAEVT1	R	数字比较器 A 输出事件 1 锁存标志（Digital Compare A Output Event 1 Latch Flag） 0：DCAEVT1 未发生跳闸事件 1：DCAEVT1 发生跳闸事件	0h

位/域	名称	R/W	描述	复位值
7	DCBEVT1	R	数字比较器 B 输出事件 1 锁存标志（Digital Compare B Output Event 1 Latch Flag） 0：DCBEVT1 未发生跳闸事件 1：DCBEVT1 发生跳闸事件	0h
15:8	保留			0h

31.8.52 清除跳闸区寄存器（TZCLR）

偏移地址：0x12E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R/S	清除全局中断标志（Global Interrupt Flag Clear） 未清除 INT 位之前，不会再生成 NVIC 中断。清除 INT 位后，设置任何标志位将会生成另一个中断脉冲。清除所有状态标志位可以防止进一步的中断。 0：无效，读取返回 0 1：清除 INT 位	0h
1	CBC	R/S	清除 CBC 锁存标志（Cycle-By-Cycle Trip Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC 跳闸条件	0h
2	OST	R/S	清除 OST 锁存标志（One-Shot Trip Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST 跳闸条件	0h
3	DCAEVT1	R/S	清除数字比较器 A 输出事件 1 锁存标志（Digital Compare A Output Event 1 Latch Flag Clear） 0：无效，读取返回 0 1：清除 DCAEVT1 跳闸条件	0h
4	DCAEVT2	R/S	清除数字比较器 A 输出事件 2 锁存标志（Digital Compare A Output Event 2 Latch Flag Clear） 0：无效，读取返回 0 1：清除 DCAEVT2 跳闸条件	0h
5	DCBEVT1	R/S	清除数字比较器 B 输出事件 1 锁存标志（Digital Compare B Output Event 1 Latch Flag Clear） 0：无效，读取返回 0 1：清除 DCBEVT1 跳闸条件	0h
6	DCBEVT2	R/S	清除数字比较器 B 输出事件 2 锁存标志（Digital Compare B Output Event 2 Latch Flag Clear） 0：无效，读取返回 0 1：清除 DCBEVT2 跳闸条件	0h
13:7	保留			0h

位/域	名称	R/W	描述	复位值
15:14	CBCPULSE	R/W	清除 CBC 锁存脉冲（Cycle-By-Cycle Trip Latch Pulse Clear） 通过该位域可以选择清除 CBC 锁存的脉冲。 00：计数器等于 0 的脉冲 01：计数器等于周期的脉冲 10：计数器等于 0 或计数器等于周期的脉冲 11：不清除 CBC 锁存	0h

31.8.53 清除跳闸区 CBC 寄存器（TZCBCCLR）

偏移地址：0x130

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CBC1	R/S	清除 CBC1 锁存标志（Cycle-By-Cycle Trip (CBC1) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC1 位	0h
1	CBC2	R/S	清除 CBC2 锁存标志（Cycle-By-Cycle Trip (CBC2) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC2 位	0h
2	CBC3	R/S	清除 CBC3 锁存标志（Cycle-By-Cycle Trip (CBC3) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC3 位	0h
3	CBC4	R/S	清除 CBC4 锁存标志（Cycle-By-Cycle Trip (CBC4) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC4 位	0h
4	CBC5	R/S	清除 CBC5 锁存标志（Cycle-By-Cycle Trip (CBC5) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC5 位	0h
5	CBC6	R/S	清除 CBC6 锁存标志（Cycle-By-Cycle Trip (CBC6) Latch Flag Clear） 0：无效，读取返回 0 1：清除 CBC6 位	0h
6	DCAEVT2	R/S	清除 CBC 选择的数字比较器 A 输出事件 2 标志（Digital Compare Output A Event 2 selected for CBC Flag Clear） 0：无效，读取返回 0 1：清除 DCAEVT2 位	0h
7	DCBEVT2	R/S	清除 CBC 选择的数字比较器 B 输出事件 2 标志（Digital Compare Output B Event 2 selected for CBC Flag Clear） 0：无效，读取返回 0 1：清除 DCBEVT2 位	0h
15:8	保留			0h

31.8.54 清除跳闸区 OST 寄存器（TZOSTCLR）

偏移地址：0x132

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	OST1	R/S	清除 OST1 锁存标志（One-Shot Trip (OST1) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST1 位	0h
1	OST2	R/S	清除 OST2 锁存标志（One-Shot Trip (OST2) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST2 位	0h
2	OST3	R/S	清除 OST3 锁存标志（One-Shot Trip (OST3) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST3 位	0h
3	OST4	R/S	清除 OST4 锁存标志（One-Shot Trip (OST4) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST4 位	0h
4	OST5	R/S	清除 OST5 锁存标志（One-Shot Trip (OST5) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST5 位	0h
5	OST6	R/S	清除 OST6 锁存标志（One-Shot Trip (OST6) Latch Flag Clear） 0：无效，读取返回 0 1：清除 OST6 位	0h
6	DCAEVT1	R/S	清除 OST 选择的数字比较器 A 输出事件 1 标志（Digital Compare Output A Event 1 selected for OST Flag Clear） 0：无效，读取返回 0 1：清除 DCAEVT1 位	0h
7	DCBEVT1	R/S	清除 OST 选择的数字比较器 B 输出事件 1 标志（Digital Compare Output B Event 1 selected for OST Flag Clear） 0：无效，读取返回 0 1：清除 DCBEVT1 位	0h
15:8	保留			0h

31.8.55 强制跳闸区标志寄存器（TZFRC）

偏移地址：0x136

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	CBC	R/S	软件强制逐周期跳闸事件（Force a Cycle-by-Cycle Trip Event via Software） 0：无效，读取返回 0 1：强制触发逐周期跳闸事件并设置 CBC 位	0h

位/域	名称	R/W	描述	复位值
2	OST	R/S	软件强制单次跳闸事件（Force a One-Shot Trip Event via Software） 0：无效，读取返回 0 1：强制触发单次跳闸事件并设置 OST 位	0h
3	DCAEVT1	R/S	数字比较器 A 输出事件 1 强制标志（Digital Compare Output A Event 1 Forced Flag） 0：无效，读取返回 0 1：强制触发 DCAEVT1 跳闸条件并设置 DCAEVT1 位	0h
4	DCAEVT2	R/S	数字比较器 A 输出事件 2 强制标志（Digital Compare Output A Event 2 Forced Flag） 0：无效，读取返回 0 1：强制触发 DCAEVT2 跳闸条件并设置 DCAEVT2 位	0h
5	DCBEVT1	R/S	数字比较器 B 输出事件 1 强制标志（Digital Compare Output B Event 1 Forced Flag） 0：无效，读取返回 0 1：强制触发 DCBE1 跳闸条件并设置 DCBEVT1 位	0h
6	DCBEVT2	R/S	数字比较器 B 输出事件 2 强制标志（Digital Compare Output B Event 2 Forced Flag） 0：无效，读取返回 0 1：强制触发 DCBE2 跳闸条件并设置 DCBEVT2 位	0h
15:7	保留			0h

31.8.56 事件触发选择寄存器（ETSEL）

偏移地址：0x148

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	INTSEL	R/W	选择 PWM 中断（PWM Interrupt Select） 通过该位域可以选择生成 PWM 中断事件的条件。 000：保留 001：在 TBCTR=0 时生成 010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成	0h
3	INTEN	R/W	使能生成 PWM 中断（PWM Interrupt Generation Enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
4	SOCASELCMP	R/W	选择 PWMxSOCA 比较器（PWMxSOCA Compare Select） 该位所选事件将通过 SOCASEL 位进行处理。 0：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件。当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件 1：当计数器向上计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件。当计数器向下计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件	0h
5	SOCBSELCMP	R/W	选择 PWMxSOCB 比较器（PWMxSOCB Compare Select） 该位所选事件将通过 SOCBSEL 位进行处理。 0：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件。当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件 1：当计数器向上计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件。当计数器向下计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件	0h
6	INTSELCMP	R/W	选择 PWMxINT 比较器（PWMxINT Compare Select） 该位所选事件将通过 INTSEL 位进行处理。 0：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件。当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPB 时使能事件 1：当计数器向上计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件。当计数器向下计数时，在 TBCTR=CMPC 或 TBCTR=CMPD 时使能事件	0h
7	保留			0h
10:8	SOCASEL	R/W	选择生成 PWMxSOCA 的事件（Generate PWMxSOCA Event Select） 通过该位域可以选择在生成 PWMxSOCA 事件的条件。根据 SOCASELCMP 位选择 PWMxSOCA 比较器事件。 000：在 DCAEVT1.soc 事件时生成 001：在 TBCTR=0 时生成 010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成	0h

位/域	名称	R/W	描述	复位值
11	SOCAEN	R/W	使能 A 路触发 ADC 转换脉冲（PWMxSOCA）（the ADC Start of Conversion A (PWMxSOCA) Pulse Enable） 0：禁止 1：使能	0h
14:12	SOCBSEL	R/W	选择生成 PWMxSOCB 的事件（Generate PWMxSOCB Event Select） 通过该位域可以选择在生成 PWMxSOCB 事件的条件。根据 SOCBSELCMP 位选择 PWMxSOCB 比较器事件。 000：在 DCBE1.soc 事件时生成 001：在 TBCTR=0 时生成 010：在 TBCTR=TBPRD 时生成 011：在 TBCTR=0 或 TBCTR=TBPRD 时生成（用于中心对齐计数模式） 100：当计数器向上计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 101：当计数器向下计数时，在 TBCTR=CMPA 或 TBCTR=CMPC 时生成 110：当计数器向上计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成 111：当计数器向下计数时，在 TBCTR=CMPB 或 TBCTR=CMPD 时生成	0h
15	SOCBEN	R/W	使能 B 路触发 ADC 转换脉冲（PWMxSOCB）（the ADC Start of Conversion B (PWMxSOCB) Pulse Enable） 0：禁止 1：使能	0h

31.8.57 事件触发预分配寄存器（ETPS）

偏移地址：0x14C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	INTPRD	R/W	选择 PWM 中断周期（PWM Interrupt Period Select） 该位域用于确定生成中断前的条件。必须使 ETSEL[INT]=1 才能生成中断，未使用 ETSEL[INT]位清除 ETFLG[INT]标志位前不会产生中断。如果产生中断，则会自动清除 INTECNT 位。 00：当 INTECNT=00 时，不产生中断且忽略 ETFRC[INT]位 01：当 INTECNT=01 时，在第 1 个事件发生时产生中断 10：当 INTECNT=10 时，在第 2 个事件发生时产生中断 11：当 INTECNT=11 时，在第 3 个事件发生时产生中断	0h

位/域	名称	R/W	描述	复位值
3:2	INTCNT	R	PWM 中断事件计数器（PWM Interrupt Event Counter） 通过该位域选择已发生 INTSEL 事件的数量。该位域会在生成中断脉冲时自动清零，当 ETSEL[INT]=0 或 ETFLG[INT]=1 时，计数器在 ETPS[INTCNT] = ETPS[INTPRD]时停止计数。 00：无 INTSEL 事件发生 01：发生 1 个 INTSEL 事件 10：发生 2 个 INTSEL 事件 11：发生 3 个 INTSEL 事件	0h
4	INTPSSEL	R/W	选择 PWM 中断预分频（PWMxINTn Pre-Scale Select） 0：通过 ETPS [INTCNT, and INTPRD]位来确定事件频率为每 0...3 个事件产生一次脉冲 1：通过 ETINTPS [INTCNT2, and INTPRD2]位来确定事件频率为每 0...15 个事件产生一次脉冲	0h
5	SOCPSSEL	R/W	选择 PWMxSOC A/B 预分频（PWMxSOC A/B Pre-Scale Select） 0：通过 SOCACNT 位/ SOCBCNT 位和 SOCAPRD 位/ SOCBPRD 位来确定事件频率为每 0...3 个事件产生一次脉冲 1：通过 SOCACNT2 位/ SOCBCNT2 位和 SOCAPRD2 位/ SOCBPRD2 位来确定事件频率为每 0...15 个事件产生一次脉冲	0h
7:6	保留			0h
9:8	SOCAPRD	R/W	选择 PWMxSOCA 周期（PWM ADC Start-of-Conversion A Event (PWMxSOCA) Period Select） 该位域用于确定生成 PWMxSOCA 脉冲前的条件。必须使 ETSEL[SOCAEN]=1 才能生成脉冲，即使设置 ETFLG[SOCA]位，也将生成 PWMxSOCA 脉冲。此时，将会自动清除 ETPS[SOCACNT] 位。 00：SOCACNT=00，不产生脉冲 01：SOCACNT=01，在第 1 个事件发生时产生脉冲 10：SOCACNT=10，在第 2 个事件发生时产生脉冲 11：SOCACNT=11，在第 3 个事件发生时产生脉冲	0h
11:10	SOCACNT	R	PWMxSOCA 事件计数器（PWM ADC Start-of-Conversion A Event (PWMxSOCA) Counte）r 通过该位域选择已发生 ETSEL[SOCASEL]事件的数量。 00：无 SOCASEL 事件发生 01：发生 1 个 SOCASEL 事件 10：发生 2 个 SOCASEL 事件 11：发生 3 个 SOCASEL 事件	0h

位/域	名称	R/W	描述	复位值
13:12	SOCBPRD	R/W	选择 PWMxSOCB 周期（PWM ADC Start-of-Conversion B Event (PWMxSOCB) Period Select） 该位域用于确定生成 PWMxSOCB 脉冲前的条件。必须使 ETSEL[SOCBEN]=1 才能生成脉冲，即使设置 ETFLG[SOCB]位，也将生成 PWMxSOCB 脉冲。此时，将会自动清除 SOCBCNT 位。 00：SOCBCNT=00，不产生脉冲 01：SOCBCNT=01，在第 1 个事件发生时产生脉冲 10：SOCBCNT=10，在第 2 个事件发生时产生脉冲 11：SOCBCNT=11，在第 3 个事件发生时产生脉冲	0h
15:14	SOCBCNT	R	PWMxSOCB 事件计数器（PWM ADC Start-of-Conversion B Event (PWMxSOCB) Counter） 通过该位域选择已发生 ETSEL[SOCBSEL]事件的数量。 00：无 SOCBSEL 事件发生 01：发生 1 个 SOCBSEL 事件 10：发生 2 个 SOCBSEL 事件 11：发生 3 个 SOCBSEL 事件	0h

31.8.58 事件触发标志寄存器（ETFLG）

偏移地址：0x150

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R	PWM 中断锁存标志（PWM Interrupt (PWMx_INT) Latch Flag） 需要清除该位才能产生新的中断，且当 INT 位仍处于设置状态时，最多可以有一个中断挂起。如果有中断挂起，则会在清除 INT 位之后才能生成中断。 0：未产生中断 1：产生中断	0h
1	保留			0h
2	SOCA	R	PWMxSOCA 标志（PWM ADC Start-of-Conversion A (PWMxSOCA) Flag） 当设置该位时，在 PWMxSOCA 输出仍继续产生转换开始脉冲。 0：未产生脉冲 1：产生脉冲	0h
3	SOCB	R	PWMxSOCB 标志（PWM ADC Start-of-Conversion B (PWMxSOCB) Flag） 当设置该位时，在 PWMxSOCB 输出仍继续产生转换开始脉冲。 0：未产生脉冲 1：产生脉冲	0h
15:4	保留			0h

31.8.59 清除事件触发寄存器（ETCLR）

偏移地址：0x154

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R/S	清除 PWM 中断标志（PWM Interrupt (PWMx_INT) Flag Clear） 0：无效，读取返回 0 1：清除 INT 位	0h
1	保留			0h
2	SOCA	R/S	清除 PWMxSOCA 标志（PWM ADC Start-of-Conversion A (PWMxSOCA) Flag Clear） 0：无效，读取返回 0 1：清除 SOCA 位	0h
3	SOCB	R/S	清除 PWMxSOCB 标志（PWM ADC Start-of-Conversion B (PWMxSOCB) Flag Clear） 0：无效，读取返回 0 1：清除 SOCB 位	0h
15:4	保留			0h

31.8.60 强制事件触发寄存器（ETFRC）

偏移地址：0x158

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R/S	强制产生中断（Generate INT Force） 如果在 ETSEL 寄存器中使能该事件，则会产生中断。 0：无效，读取返回 0 1：产生中断并设置 INT 位	0h
1	保留			0h
2	SOCA	R/S	强制产生 SOCA 脉冲（Generates a Pulse on PWMxSOCA Force） 该位用于测试。 如果在 ETSEL 寄存器中使能该事件，则会产生 PWMxSOCA 脉冲。为了确定是否生成 PWMxSOCA 事件，无论是否生成脉冲都会设置 SOCA 位。 0：无效，读取返回 0 1：产生一个脉冲并设置 SOCA 位	0h
3	SOCB	R/S	强制产生 SOCB 脉冲（Generates a Pulse on PWMxSOCB Force） 该位用于测试。 如果在 ETSEL 寄存器中使能该事件，则会产生 PWMxSOCB 脉冲。为了确定是否生成 PWMxSOCB 事件，无论是否生成脉冲都会设置 SOCB 位。 0：无效，读取返回 0 1：产生一个脉冲并设置 SOCB 位	0h
15:4	保留			0h

31.8.61 事件触发中断预分频寄存器（ETINTPS）

偏移地址：0x15C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	INTPRD2	R/W	选择 PWM 中断周期 2（PWM Interrupt Period 2 Select） 当 INTPSSEL=1 时，该位域用于确定生成中断前的条件。 0000：INTCNT2=0000，不产生中断 0001：INTCNT2=0001，在第 1 个事件发生时产生中断 0010：INTCNT2=0010，在第 2 个事件发生时产生中断 0011：INTCNT2=0011，在第 3 个事件发生时产生中断 ... 1111：INTCNT2=1111，在第 15 个事件发生时产生中断	0h
7:4	INTCNT2	R	PWM 中断事件计数器 2（PWM Interrupt Event Counter 2） 当 INTPSSEL=1 时，该位用于选择已发生 INTSEL 事件的数量。 0000：无 INTSEL 事件发生 0001：发生 1 个 INTSEL 事件 0010：发生 2 个 INTSEL 事件 0011：发生 3 个 INTSEL 事件 ... 1111：发生 15 个 INTSEL 事件	0h
15:8	保留			0h

31.8.62 事件触发 SOC 预分频寄存器（ETSOCPS）

偏移地址：0x160

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	SOCAPRD2	R/W	选择 PWMxSOCA 周期 2（PWM ADC Start-of-Conversion A Event (PWMxSOCA) Period 2 Select） 当 SOCPSSEL=1 时，该位域用于确定生成 SOCA 脉冲前的条件。 0000：SOCACNT2=0000，不产生中断 0001：SOCACNT2=0001，在第 1 个事件发生时产生 SOCA 脉冲 0010：SOCACNT2=0010，在第 2 个事件发生时产生 SOCA 脉冲 0011：SOCACNT2=0011，在第 3 个事件发生时产生 SOCA 脉冲 ... 1111：SOCACNT2=1111，在第 15 个事件发生时产生 SOCA 脉冲	0h
7:4	SOCACNT2	R	PWMxSOCA 事件计数器 2（PWM ADC Start-of-Conversion A Event (PWMxSOCA) Counter 2） 当 SOCPSSEL=1 时，该位用于选择已发生 SOCASEL 事件的数量。 0000：无 SOCASEL 事件发生 0001：发生 1 个 SOCASEL 事件 0010：发生 2 个 SOCASEL 事件 0011：发生 3 个 SOCASEL 事件 ... 1111：发生 15 个 SOCASEL 事件	0h

位/域	名称	R/W	描述	复位值
11:8	SOCBPRD2	R/W	PWMxSOCB 事件计数器 2（PWM ADC Start-of-Conversion B Event (PWMxSOCB) Counter 2） 当 SOCPSSEL=1 时，该位域用于确定生成 SOCB 脉冲前的条件。 0000：SOCBCNT2=0000，不产生中断 0001：SOCBCNT2=0001，在第 1 个事件发生时产生 SOCB 脉冲 0010：SOCBCNT2=0010，在第 2 个事件发生时产生 SOCB 脉冲 0011：SOCBCNT2=0011，在第 3 个事件发生时产生 SOCB 脉冲 ... 1111：SOCBCNT2=1111，在第 15 个事件发生时产生 SOCB 脉冲	0h
15:12	SOCBCNT2	R	PWMxSOCB 事件计数器（PWM ADC Start-of-Conversion B Event (PWMxSOCB) Counter 2） 当 SOCPSSEL=1 时，该位用于选择已发生 SOCASEL 事件的数量。 0000：无 SOCASEL 事件发生 0001：发生 1 个 SOCASEL 事件 0010：发生 2 个 SOCASEL 事件 0011：发生 3 个 SOCASEL 事件 ... 1111：发生 15 个 SOCASEL 事件	0h

31.8.63 事件触发计数器初始化控制寄存器（ETCNTINITCTL）

偏移地址：0x164

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	保留			0h
10	INTINITFRC	R/S	强制 PWM 中断计数器 2 初始化（PWMxINT Counter 2 Initialization Force） 0：无效 1：强制 PWMxSOCB 计数器通过 INTINIT 内容进行初始化	0h
11	SOCAINITFRC	R/S	强制 PWMxSOCA 计数器 2 初始化 （PWMxSOCA Counter 2 Initialization Force） 0：无效 1：强制 PWMxSOCB 计数器通过 SOCAINIT 内容进行初始化	0h
12	SOCBINITFRC	R/S	强制 PWMxSOCB 计数器 2 初始化 （PWMxSOCB Counter 2 Initialization Force） 0：无效 1：强制 PWMxSOCB 计数器通过 SOCBINIT 内容进行初始化	0h

位/域	名称	R/W	描述	复位值
13	INTINITEN	R/W	使能 PWM 中断计数器 2 初始化（PWMxINT Counter 2 Initialization Enable） 0：无效 1：PWM 中断计数器在同步事件或软件强制时，通过 INTINITFRC 内容进行初始化	0h
14	SOCAINITEN	R/W	使能 PWMxSOCA 计数器 2 初始化（PWMxSOCA Counter 2 Initialization Enable） 0：无效 1：PWMxSOCA 计数器在同步事件或软件强制时，通过 SOCAINITFRC 内容进行初始化	0h
15	SOCBINITEN	R/W	使能 PWMxSOCB 计数器 2 初始化（PWMxSOCA Counter 2 Initialization Enable） 0：无效 1：PWMxSOCB 计数器在同步事件或软件强制时，通过 SOCBINITFRC 内容进行初始化	0h

31.8.64 事件触发计数器初始化寄存器（ETCNTINIT）

偏移地址：0x168

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	INTINIT	R/W	PWM 中断计数器 2 初始化（PWMxINT Counter 2 Initialization） 当发生同步事件或软件强制时，该寄存器内容被用于初始化 PWM 中断计数器 2。	0h
7:4	SOCAINIT	R/W	PWMxSOCA 计数器 2 初始化（PWMxSOCA Counter 2 Initialization Force） 当发生同步事件或软件强制时，该寄存器内容被用于初始化 PWMxSOCA 计数器 2。	0h
11:8	SOCBINIT	R/W	PWMxSOCB 计数器 2 初始化（PWMxSOCB Counter 2 Initialization Force） 当发生同步事件或软件强制时，该寄存器内容被用于初始化 PWMxSOCB 计数器 2。	0h
15:12	保留			0h

31.8.65 数字比较跳闸选择寄存器（DCTRIPSEL）

偏移地址：0x180

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	DCAHCOMPSEL	R/W	选择数字比较器 A 高位输入（Digital Compare A High Input Select） 0000：跳闸输入 1 0001：跳闸输入 2 0010：跳闸输入 3 0011：跳闸输入 4 ... 1100：保留 1101：跳闸输入 14 1110：跳闸输入 15 1111：跳闸输入组合。通过 DCAHTRIPSEL 寄存器所选的全部跳闸输入进行逻辑或运行。	0h
7:4	DCALCOMPSEL	R/W	选择数字比较器 A 低位输入（Digital Compare A Low Input Select） 参考 DCAHCOMPSEL 描述	0h
11:8	DCBHCOMPSEL	R/W	选择数字比较器 B 高位输入（Digital Compare B High Input Select） 参考 DCAHCOMPSEL 描述	0h
15:12	DCBLCOMPSEL	R/W	选择数字比较器 B 低位输入（Digital Compare B Low Input Select） 参考 DCAHCOMPSEL 描述	0h

31.8.66 数字比较 A 控制寄存器（DCACTL）

偏移地址：0x186

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	EVT1SRCSEL	R/W	选择 DCAEVT1 信号源（DCAEVT1 Source Signal Select） 0：DCAEVT1 信号 1：DCEVTFILT 信号	0h
1	EVT1FRCSYNCSEL	R/W	选择 DCAEVT1 强制同步信号（DCAE1 Force Synchronization Signal Select） 0：与 PWMCLK 同步 1：异步传递	0h
2	EVT1SOCE	R/W	使能 DCAEVT1 生成 SOC（DCAE1 Generate SOC Enabled） 0：禁止 1：使能	0h
3	EVT1SYNCE	R/W	使能 DCAEVT1 生成 SYNC（DCAE1 Generate SYNC Enabled） 0：禁止 1：使能	0h
7:4	保留			0h
8	EVT2SRCSEL	R/W	选择 DCAEVT2 信号源（DCAEVT2 Source Signal Select） 0：DCAEVT2 信号 1：DCEVTFILT 信号	0h

位/域	名称	R/W	描述	复位值
9	EVT2FRCSYNCSEL	R/W	选择 DCAEVT2 强制同步信号（DCAEVT2 Force Synchronization Signal Select） 0：与 PWMCLK 同步 1：异步传递	0h
15:10	保留			0h

31.8.67 数字比较 B 控制寄存器（DCBCTL）

偏移地址：0x188

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	EVT1SRCSEL	R/W	选择 DCBE1 信号源（DCBE1 Source Signal Select） 0：DCBE1 信号 1：DCEVTFILT 信号	0h
1	EVT1FRCSYNCSEL	R/W	选择 DCBE1 强制同步信号（DCBE1 Force Synchronization Signal Select） 0：与 PWMCLK 同步 1：异步传递	0h
2	EVT1SOCE	R/W	使能 DCBE1 生成 SOC（DCBE1 Generate SOC Enabled） 0：禁止 1：使能	0h
3	EVT1SYNCE	R/W	使能 DCBE1 生成 SYNC（DCBE1 Generate SYNC Enabled） 0：禁止 1：使能	0h
7:4	保留			0h
8	EVT2SRCSEL	R/W	选择 DCBE2 信号源（DCBE2 Source Signal Select） 0：DCBE2 信号 1：DCEVTFILT 信号	0h
9	EVT2FRCSYNCSEL	R/W	选择 DCBE2 强制同步信号（DCBE2 Force Synchronization Signal Select） 0：与 PWMCLK 同步 1：异步传递	0h
15:10	保留			0h

31.8.68 数字比较滤波器控制寄存器（DCFCTL）

偏移地址：0x18E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	SRCSEL	R/W	选择滤波块信号源（Filter Block Signal Source Select） 00：DCAEVT1 信号 01：DCAEVT2 信号 10：DCBEVT1 信号 11：DCBEVT2 信号	0h

位/域	名称	R/W	描述	复位值
2	BLANKE	R/W	使能空白窗口（Blanking Window Enable） 0：禁止 1：使能	0h
3	BLANKINV	R/W	翻转空白窗口（Blanking Window Inversion） 0：未翻转 1：翻转	0h
5:4	PULSESEL	R/W	选择空白和捕获对齐脉冲（Blanking & Capture Alignment Pulse Select） 00：TBCTR=TBPRD 01：TBCTR=0 10：TBCTR=0 或 TBCTR=TBPRD 11：保留	0h
6	EDGEFILTSEL	R/W	选择边沿滤波器（Edge Filter Select） 0：未选择 1：选择	0h
7	保留			0h
9:8	EDGEMODE	R/W	选择边沿模式（Edge Mode Select） 00：上升沿 01：下降沿 10：双边沿 11：保留	0h
12:10	EDGECOUNT	R/W	选择计数边沿数（Edge Count Select） 该位域用于选择在生成 DCEVTFILT 信号上的 TBCLK 宽脉冲之前要计算边缘数。 000：不计算边沿 001：计算 1 个边沿 010：计算 2 个边沿 011：计算 3 个边沿 100：计算 4 个边沿 101：计算 5 个边沿 110：计算 6 个边沿 111：计算 7 个边沿	0h
15:13	EDGESTATUS	R	边沿状态（Edge Status） 该位域反映当前捕获的边沿总数。当 EDGESTATUS 与 EDGECOUNT 匹配时，该位被设置为 0，并且在 DCEVTFILT 信号上生成一个 TBCLK 宽脉冲。边沿计数器可以通过将 000 写入 EDGECOUNT 值来复位。	0h

31.8.69 数字比较捕获控制寄存器（DCCAPCTL）

偏移地址：0x190

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	CAPE	R/W	使能 TBCTL 计数器捕获（TBCTL Counter Capture Enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
1	SHDWMODE	R/W	选择 TBCTR 计数器捕获影子模式（TBCTR Counter Capture Shadow Mode Select） 0：影子模式。当 TBCTR=0 或 TBCTR=TBPRD 时（由 PULSESEL 位定义），DCCAP 活动寄存器的值将被复制到影子寄存器。此时，从 DCCAP 寄存器中读取的值为影子寄存器值。 1：活动模式。在该模式下，禁用影子寄存器且从 DCCAP 寄存器中读取的值始终为活动寄存器值。	0h
12:2	保留			0h
13	CAPSTS	R	DC 捕获事件锁存标志（DC Capture Event Latched Flag） 0：未发生 1：发生	0h
14	CAPCLR	R/S	清除 DC 捕获锁存标志（DC Capture Latched Flag Clear） 0：无效 1：清除 CAPSTS 位	0h
15	CAPMODE	R/W	计数器捕获模式（Counter Capture Mode） 0：当 CAPE=1 且发生 DCEVTFILT 事件时，将当前计数器值捕获到活动寄存器中。如果发生跳闸事件，则在下一个 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）重新触发捕获机制前，将一直忽略进一步的捕跳闸/获事件。 影子模式：如果 SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子寄存器值。 活动模式：如果 SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。 1：当 CAPE=1 且发生 DCEVTFILT 事件时，将当前计数器值捕获到活动寄存器中。如果发生跳闸事件，则设置 CAPSTS 位并在清除该标志位前一直忽略进一步的捕跳闸/获事件。通过 CAPCLR 位可以清除该标志位并重新触发捕获机制。 影子模式：SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子寄存器值。 活动模式：SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。	0h

31.8.70 数字比较滤波器偏移寄存器（DCFOFFSET）

偏移地址：0x192

复位类型：SYSRSn

该寄存器为影子寄存器，活动寄存器在 PULSESEL 位定义的参考点装载。当活动寄存器装载时，初始化空白偏移计数器并开始向下计数。如果偏移计数器停止，则应用空白窗口。当空白窗口处于活动状态时，就会重新使能空白窗口计数器。

位/域	名称	R/W	描述	复位值
15:0	DCFOFFSET	R/W	空白窗口（Blanking Window Offset） 该位域用于指定空白窗口参考点（由 PULSESEL 位定义 TBCTR=0 或 TBCTR=TBPRD）到空白窗口应用点的 TBCLK 周期数。	0h

31.8.71 数字比较滤波器偏移计数器寄存器（DCFOFFSETCNT）

偏移地址：0x194

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DCFOFFSETCNT	R	空白偏移计数器（Blanking Offset Counter） 这 16 位位域用于表示当前偏移计数器的值。当该计数器向下计数达到 0 之后将会停止计数，直到下一个 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）重新装载计数器。即使设备由于仿真停止而暂停，该计数器也会继续向下计数，因此，偏移计数器不受自由或软件仿真位影响。	0h

31.8.72 数字比较滤波器窗口寄存器（DCFWINDOW）

偏移地址：0x196

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DCFWINDOW	R/W	空白窗口宽度（Blanking Window Width） 0000：无空白窗口生成 0001...FFFF：指定空白窗口宽度，单位为 TBCLK 周期。当偏移计数器达到指定值时，空白窗口开始。此时，空白窗口计数器装载并且开始向下计数。 需要注意避免当偏移计数器到达指定值并且空白窗口当前处于活动状态时，空白窗口计数器不会重新启动且空白窗口会提前结束的情况。空白窗口可以在 PWM 周期中的某个时间段内将信号暂时关闭。	0h

31.8.73 数字比较滤波器窗口计数器寄存器（DCFWINDOWCNT）

偏移地址：0x198

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DCFWINDOWCNT	R	空白窗口计数器（Blanking Window Counter） 这 16 位位域用于表示当前窗口计数器的值。该计数器向下计数达到 0 之后就会停止计数，直到偏移计数器再次达到 0 时重新装载。	0h

31.8.74 数字比较计数器捕获寄存器（DCCAP）

偏移地址：0x19E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	DCCAP	R	数字比较时基计数器捕获（Digital Compare Time-Base Counter Capture） 必须使 CAPE=1 才能使能 TBCTL 计数器捕获。当使能该功能时，在 DCEVTFLT 事件的上升沿触发时，会记录 TBCTR 的值。在 PULSESEL 位定义下一个 TBCTR=0 或 TBCTR=TBPRD 事件前，会忽略后面的捕获事件。 通过 SHDWMODE 位使能 DCCAP 寄存器的影子模式（默认）。 影子模式：SHDWMODE=0，当 TBCTR=0 或 TBCTR=TBPRD 事件（由 PULSESEL 位定义）时，活动寄存器值将被复制到影子寄存器中，则 CPU 读取该寄存器将返回影子寄存器值。 活动模式：SHDWMODE=1，则 CPU 读取该寄存器将返回活动寄存器值。 活动寄存器和影子寄存器的内存映射地址相同。	0h

31.8.75 数字比较 A 高跳闸选择寄存器（DCAHTRIPSEL）

偏移地址：0x1A4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TRIPINPUT1	R/W	TRIP 输入 1（TRIP Input 1） 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAH 多路复用器	0h
1	TRIPINPUT2	R/W	TRIP 输入 2（TRIP Input 2） 参考 TRIPINPUT1 描述	0h
2	TRIPINPUT3	R/W	TRIP 输入 3（TRIP Input 3） 参考 TRIPINPUT1 描述	0h
3	TRIPINPUT4	R/W	TRIP 输入 4（TRIP Input 4） 参考 TRIPINPUT1 描述	0h
4	TRIPINPUT5	R/W	TRIP 输入 5（TRIP Input 5） 参考 TRIPINPUT1 描述	0h
5	TRIPINPUT6	R/W	TRIP 输入 6（TRIP Input 6） 参考 TRIPINPUT1 描述	0h
6	TRIPINPUT7	R/W	TRIP 输入 7（TRIP Input 7） 参考 TRIPINPUT1 描述	0h
7	TRIPINPUT8	R/W	TRIP 输入 8（TRIP Input 8） 参考 TRIPINPUT1 描述	0h
8	TRIPINPUT9	R/W	TRIP 输入 9（TRIP Input 9） 参考 TRIPINPUT1 描述	0h
9	TRIPINPUT10	R/W	TRIP 输入 10（TRIP Input 10） 参考 TRIPINPUT1 描述	0h
10	TRIPINPUT11	R/W	TRIP 输入 11（TRIP Input 11） 参考 TRIPINPUT1 描述	0h
11	TRIPINPUT12	R/W	TRIP 输入 12（TRIP Input 12） 参考 TRIPINPUT1 描述	0h

位/域	名称	R/W	描述	复位值
12	保留			0h
13	TRIPINPUT14	R/W	TRIP 输入 14（TRIP Input 14） 参考 TRIPINPUT1 描述	0h
14	TRIPINPUT15	R/W	TRIP 输入 15（TRIP Input 15） 参考 TRIPINPUT1 描述	0h
15	保留			0h

31.8.76 数字比较 A 低跳闸选择寄存器（DCALTRIPSEL）

偏移地址：0x1A6

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TRIPINPUT1	R/W	TRIP 输入 1（TRIP Input 1） 0：未选择 1：选择作为组合逻辑或运算的输入到 DCAL 多路复用器	0h
1	TRIPINPUT2	R/W	TRIP 输入 2（TRIP Input 2） 参考 TRIPINPUT1 描述	0h
2	TRIPINPUT3	R/W	TRIP 输入 3（TRIP Input 3） 参考 TRIPINPUT1 描述	0h
3	TRIPINPUT4	R/W	TRIP 输入 4（TRIP Input 4） 参考 TRIPINPUT1 描述	0h
4	TRIPINPUT5	R/W	TRIP 输入 5（TRIP Input 5） 参考 TRIPINPUT1 描述	0h
5	TRIPINPUT6	R/W	TRIP 输入 6（TRIP Input 6） 参考 TRIPINPUT1 描述	0h
6	TRIPINPUT7	R/W	TRIP 输入 7（TRIP Input 7） 参考 TRIPINPUT1 描述	0h
7	TRIPINPUT8	R/W	TRIP 输入 8（TRIP Input 8） 参考 TRIPINPUT1 描述	0h
8	TRIPINPUT9	R/W	TRIP 输入 9（TRIP Input 9） 参考 TRIPINPUT1 描述	0h
9	TRIPINPUT10	R/W	TRIP 输入 10（TRIP Input 10） 参考 TRIPINPUT1 描述	0h
10	TRIPINPUT11	R/W	TRIP 输入 11（TRIP Input 11） 参考 TRIPINPUT1 描述	0h
11	TRIPINPUT12	R/W	TRIP 输入 12（TRIP Input 12） 参考 TRIPINPUT1 描述	0h
12	保留			0h
13	TRIPINPUT14	R/W	TRIP 输入 14（TRIP Input 14） 参考 TRIPINPUT1 描述	0h
14	TRIPINPUT15	R/W	TRIP 输入 15（TRIP Input 15） 参考 TRIPINPUT1 描述	0h
15	保留			0h

31.8.77 数字比较 B 高跳闸选择寄存器（DCBHTRIPSEL）

偏移地址：0x1A8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TRIPINPUT1	R/W	TRIP 输入 1（TRIP Input 1） 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBH 多路复用器	0h
1	TRIPINPUT2	R/W	TRIP 输入 2（TRIP Input 2） 参考 TRIPINPUT1 描述	0h
2	TRIPINPUT3	R/W	TRIP 输入 3（TRIP Input 3） 参考 TRIPINPUT1 描述	0h
3	TRIPINPUT4	R/W	TRIP 输入 4（TRIP Input 4） 参考 TRIPINPUT1 描述	0h
4	TRIPINPUT5	R/W	TRIP 输入 5（TRIP Input 5） 参考 TRIPINPUT1 描述	0h
5	TRIPINPUT6	R/W	TRIP 输入 6（TRIP Input 6） 参考 TRIPINPUT1 描述	0h
6	TRIPINPUT7	R/W	TRIP 输入 7（TRIP Input 7） 参考 TRIPINPUT1 描述	0h
7	TRIPINPUT8	R/W	TRIP 输入 8（TRIP Input 8） 参考 TRIPINPUT1 描述	0h
8	TRIPINPUT9	R/W	TRIP 输入 9（TRIP Input 9） 参考 TRIPINPUT1 描述	0h
9	TRIPINPUT10	R/W	TRIP 输入 10（TRIP Input 10） 参考 TRIPINPUT1 描述	0h
10	TRIPINPUT11	R/W	TRIP 输入 11（TRIP Input 11） 参考 TRIPINPUT1 描述	0h
11	TRIPINPUT12	R/W	TRIP 输入 12（TRIP Input 12） 参考 TRIPINPUT1 描述	0h
12	保留			0h
13	TRIPINPUT14	R/W	TRIP 输入 14（TRIP Input 14） 参考 TRIPINPUT1 描述	0h
14	TRIPINPUT15	R/W	TRIP 输入 15（TRIP Input 15） 参考 TRIPINPUT1 描述	0h
15	保留			0h

31.8.78 数字比较 B 低跳闸选择寄存器（DCBLTRIPSEL）

偏移地址：0x1AA

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TRIPINPUT1	R/W	TRIP 输入 1（TRIP Input 1） 0：未选择 1：选择作为组合逻辑或运算的输入到 DCBL 多路复用器	0h

位/域	名称	R/W	描述	复位值
1	TRIPINPUT2	R/W	TRIP 输入 2（TRIP Input 2） 参考 TRIPINPUT1 描述	0h
2	TRIPINPUT3	R/W	TRIP 输入 3（TRIP Input 3） 参考 TRIPINPUT1 描述	0h
3	TRIPINPUT4	R/W	TRIP 输入 4（TRIP Input 4） 参考 TRIPINPUT1 描述	0h
4	TRIPINPUT5	R/W	TRIP 输入 5（TRIP Input 5） 参考 TRIPINPUT1 描述	0h
5	TRIPINPUT6	R/W	TRIP 输入 6（TRIP Input 6） 参考 TRIPINPUT1 描述	0h
6	TRIPINPUT7	R/W	TRIP 输入 7（TRIP Input 7） 参考 TRIPINPUT1 描述	0h
7	TRIPINPUT8	R/W	TRIP 输入 8（TRIP Input 8） 参考 TRIPINPUT1 描述	0h
8	TRIPINPUT9	R/W	TRIP 输入 9（TRIP Input 9） 参考 TRIPINPUT1 描述	0h
9	TRIPINPUT10	R/W	TRIP 输入 10（TRIP Input 10） 参考 TRIPINPUT1 描述	0h
10	TRIPINPUT11	R/W	TRIP 输入 11（TRIP Input 11） 参考 TRIPINPUT1 描述	0h
11	TRIPINPUT12	R/W	TRIP 输入 12（TRIP Input 12） 参考 TRIPINPUT1 描述	0h
12	保留			0h
13	TRIPINPUT14	R/W	TRIP 输入 14（TRIP Input 14） 参考 TRIPINPUT1 描述	0h
14	TRIPINPUT15	R/W	TRIP 输入 15（TRIP Input 15） 参考 TRIPINPUT1 描述	0h
15	保留			0h

31.8.79 PWM 锁定寄存器（PWMLOCK）

偏移地址：0x1F4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	HRLOCK	R/WO	高精度 PWM 锁定（High Resolution PWM LOCK） HRPWM 寄存器地址映射范围为 0x40...0x5A。 0：HRPWM 寄存器受到 WRPRT 保护 1：锁定 HRPWM 寄存器且不可进行写操作	0h
1	GLLOCK	R/WO	全局 PWM 装载锁定（Global PWM Load LOCK） 全局 PWM 装载寄存器地址映射范围为 0x68...0x6A。 0：全局 PWM 装载寄存器受到 WRPRT 保护 1：锁定全局 PWM 装载寄存器且不可进行写操作	0h

位/域	名称	R/W	描述	复位值
2	TZCFGLOCK	R/WO	跳闸区配置锁定（Trip-zone Configure LOCK） 跳闸区配置寄存器地址映射范围为 0x100...0x11A。 0：跳闸区配置寄存器受到 WRPRT 保护 1：跳闸区配置寄存器且不可进行写操作	0h
3	TZCLRLOCK	R/WO	跳闸区清除锁定（Trip-zone Configure LOCK） 跳闸区清除寄存器地址映射范围为 0x12E...0x136。 0：跳闸区清除寄存器受到 WRPRT 保护 1：跳闸区清除寄存器且不可进行写操作	0h
4	DCLOCK	R/WO	数字比较锁定（Digital Compare LOCK） 数字比较寄存器地址映射范围为 0x180...0x1B2。 0：数字比较寄存器受到 WRPRT 保护 1：锁定数字比较寄存器且不可进行写操作	0h
15:5	保留			0h
31:16	KEY	R0/W	关键字（KEY） 只有将该位域的值设置为 0xA5A5 才能成功写入此寄存器。 当 KEY 匹配时，如果对该寄存器进行 16-位写操作，无论是该寄存器的[15:0]还是[31:16]，则都会被忽略，只有 32 位写操作才能成功。	0h

31.8.80 硬件谷模式延迟寄存器（HWVDELVAL）

偏移地址：0x1FA

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	HWVDELVAL	R	硬件谷延迟值（Hardware Valley Delay Value） 这些位域用于表示 VDELAYDIV 位中计算的硬件谷延迟值。当触发谷捕获序列并更新谷捕获 1/2 值时，该硬件谷延迟值也同时更新。	0h

31.8.81 硬件谷计数器寄存器（VCNTVAL）

偏移地址：0x1FC

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	VCNTVAL	R	谷时基计数器（Valley Time Base Counter） 这些位域用于表示当在 VCNTCFG 寄存器中选择了 STOPEDGE 位时，该寄存器记录并保存捕获的谷计数器值。	0h

31.8.82 同步输入和输出选择寄存器（SYNCSELECT）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	PWM4SYNCIN	R/W	选择 PWM4 同步输入源（PWM4 Sync Input Source Select） 000：PWM1 同步输出 001：PWM4 同步输出（保留） 010：PWM7 同步输出（保留） 011：保留 100：CAP1 同步输出（保留） 101：外部同步输入 1 110：外部同步输入 2 111：保留 注意：如果选择了保留位置，则同步复用器的输出将被驱动为低电平。	7h
5:3	PWM7SYNCIN	R/W	选择 PWM7 同步输入源（PWM7 Sync Input Source Select） 000：PWM1 同步输出 001：PWM4 同步输出 010：PWM7 同步输出（保留） 011：保留 100：CAP1 同步输出（保留） 101：外部同步输入 1 110：外部同步输入 2 111：CAP4 同步输出（保留） 注意：如果选择了保留位置，则同步复用器的输出将被驱动为低电平。	7h
8:6	保留			7h
11:9	CAP1SYNCIN	R/W	选择 CAP1 同步输入源（CAP1 Sync Input Source Select） 000：PWM1 同步输出 001：PWM4 同步输出 010：PWM7 同步输出 011：保留 100：CAP1 同步输出（保留） 101：外部同步输入 1 110：外部同步输入 2 111：CAP4 同步输出（保留） 注意：如果选择了保留位置，则同步复用器的输出将被驱动为低电平。	7h

位/域	名称	R/W	描述	复位值
14:12	CAP4SYNCIN	R/W	选择 CAP4 同步输入源（CAP4 Sync Input Source Select） 000：PWM1 同步输出 001：PWM4 同步输出 010：PWM7 同步输出 011：保留 100：CAP1 同步输出 101：外部同步输入 1 110：外部同步输入 2 111：CAP4 同步输出（保留） 注意：如果选择了保留位置，则同步复用器的输出将被驱动为低电平。	7h
17:15	CAP6SYNCIN	R/W	选择 CAP6 同步输入源（CAP6 Sync Input Source Select） 000：PWM1 同步输出 001：PWM4 同步输出 010：PWM7 同步输出 011：保留 100：CAP1 同步输出 101：外部同步输入 1 110：外部同步输入 2 111：CAP4 同步输出 注意：如果选择了保留位置，则同步复用器的输出将被驱动为低电平。	7h
26:18	保留			0h
28:27	SYNCOUT	R/W	选择同步输出源（Sync Output Source Select） 00：PWM1 同步输出信号 01：PWM4 同步输出信号 10：PWM7 同步输出信号 11：保留，默认设置为 PWM1 同步输出信号	0h
31:29	PWM1SYNCIN	R/W	选择 PWM1 同步输入源（PWM1 Sync Input Source Select） 000：EXTSYNCIN1 其他：保留 注意：只能选择一个有效选项 000，否则同步复用器的输出将被驱动为低电平。	7h

31.8.83 外部 ADCSOC 选择寄存器（ADCSOCOUTSELECT）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PWM1SOCAEN	R/W	选择 PWM1 SOCA 输出源（PWM1 SOCA Output Source Select） 0：未选择 1：选择	0h

位/域	名称	R/W	描述	复位值
1	PWM2SOCAEN	R/W	选择 PWM2 SOCA 输出源（PWM2 SOCA Output Source Select） 0：未选择 1：选择	0h
2	PWM3SOCAEN	R/W	选择 PWM3 SOCA 输出源（PWM3 SOCA Output Source Select） 0：未选择 1：选择	0h
3	PWM4SOCAEN	R/W	选择 PWM4 SOCA 输出源（PWM4 SOCA Output Source Select） 0：未选择 1：选择	0h
4	PWM5SOCAEN	R/W	选择 PWM5 SOCA 输出源（PWM5 SOCA Output Source Select） 0：未选择 1：选择	0h
5	PWM6SOCAEN	R/W	选择 PWM6 SOCA 输出源（PWM6 SOCA Output Source Select） 0：未选择 1：选择	0h
6	PWM7SOCAEN	R/W	选择 PWM7 SOCA 输出源（PWM7 SOCA Output Source Select） 0：未选择 1：选择	0h
7	PWM8SOCAEN	R/W	选择 PWM8 SOCA 输出源（PWM8 SOCA Output Source Select） 0：未选择 1：选择	0h
15:8	保留			0h
16	PWM1SOCBEN	R/W	选择 PWM1 SOCB 输出源（PWM1 SOCB Output Source Select） 0：未选择 1：选择	0h
17	PWM2SOCBEN	R/W	选择 PWM2 SOCB 输出源（PWM2 SOCB Output Source Select） 0：未选择 1：选择	0h
18	PWM3SOCBEN	R/W	选择 PWM3 SOCB 输出源（PWM3 SOCB Output Source Select） 0：未选择 1：选择	0h
19	PWM4SOCBEN	R/W	选择 PWM4 SOCB 输出源（PWM4 SOCB Output Source Select） 0：未选择 1：选择	0h

位/域	名称	R/W	描述	复位值
20	PWM5SOCBEN	R/W	选择 PWM5 SOCB 输出源（PWM5 SOCB Output Source Select） 0：未选择 1：选择	0h
21	PWM6SOCBEN	R/W	选择 PWM6 SOCB 输出源（PWM6 SOCB Output Source Select） 0：未选择 1：选择	0h
22	PWM7SOCBEN	R/W	选择 PWM7 SOCB 输出源（PWM7 SOCB Output Source Select） 0：未选择 1：选择	0h
23	PWM8SOCBEN	R/W	选择 PWM8 SOCB 输出源（PWM8 SOCB Output Source Select） 0：未选择 1：选择	0h
31:24	保留			0h

31.8.84 同步输入源和外部 ADCSOC 选择锁定寄存器（SYNCSOCLOCK）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SYNCSELECT	R/WO	SYNCSELECT 锁定（SYNCSELECT Lock） 该位用于保护 SYNCSELECT 寄存器的写操作。如果 SYNCSELECT 寄存器的任何位被设置为 1，只能通过 SYSRSn 来清除，且对该寄存器任何位写入 0 都无效。 锁存机制仅使用于写操作，但是仍然允许对其进行读取操作。 0：未锁定 1：锁定	0h
1	ADCSOCOUTSELECT	R/WO	ADCSOCOUTSELECT 锁定（ADCSOCOUTSELECT Lock） 该位用于保护 ADCSOCOUTSELECT 寄存器的写操作。如果 ADCSOCOUTSELECT 寄存器的任何位被设置为 1，只能通过 SYSRSn 来清除，且对该寄存器任何位写入 0 都无效。 锁存机制仅使用于写操作，但是仍然允许对其进行读取操作。 0：未锁定 1：锁定	0h
15:2	保留			0h

32 高分辨率脉冲宽度调制器（HRPWM）

32.1 术语全称、缩写描述

表格 154 术语全称、缩写描述

中文全称	英文全称	英文缩写
高分辨定位器	High Resolution Positioner	HRP
上升沿	Rising Edge	RE
下降沿	Falling Edge	FE
双边沿	Both Edges	BE
比例系数优化	Scale Factor Optimizing	SFO

32.2 简介

HRPWM 是一种用于扩展 PWM 的时间分辨率能力的技术，通常在 PWM 分辨率低于大约 9~10 位时使用。

有的 PWM 实例包括一个硬件扩展（HRPWM），允许更精确地控制 PWM 输出，详情请参见数据手册确定哪些 PWM 模块支持 HRPWM 功能。

典型低于 250kHzPWM 操作通常不需要 HRPWM。HRPWM 功能最适用于要求高频 PWM 的功率转换拓扑，例如单（多）相 Buck、Boost 和反激变换器、相移全桥和 D 类功率放大器的直接调制。

32.3 主要特征

（1） 可以用于控制 PWM 信号的占空比和相位

（2） 具有扩展时间分辨率能力

（3） 死区高分辨率控制
- 允许在半周期时钟运行时，对 RED 和 FED 进行控制

（4） 允许对 PWM 输出进行使能高分辨率切换

（5） 通过翻转 PWMxA 信号输出可以使能对 PWMxB 信号输出的高分辨率控制

（6） 应用在 PWMxA 和 PWMxB 输出（PWM 的 A 和 B 信号路径）上

（7） 自检诊断软件模式
- 该模块用于检查 HRP 逻辑的运行状况，即检查是否按设计运行

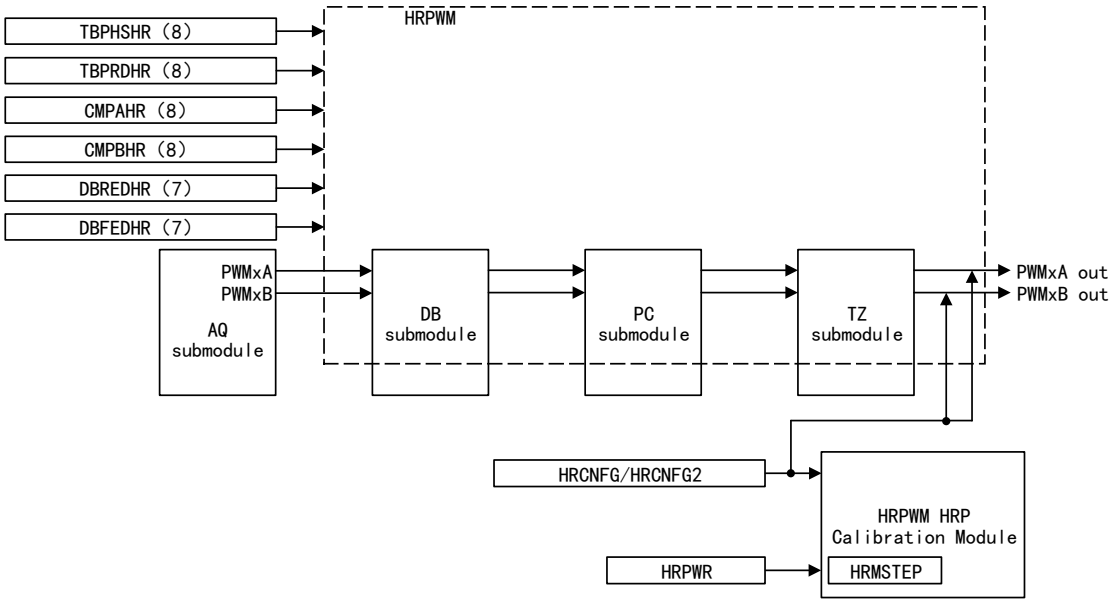
（8） 通过扩展 CMPA、CMPB 和 TBPHS 寄存器对 PWM 信号进行细致的边沿定位或时间粒度控制

（9） 如果设备有 PWM 模块，可以对 PWM 输出信号的周期、占空比和相位进行高分辨率控制

32.4 结构框图

32.4.1 HRPWM 结构框图

图 139 多个 PWM 模块结构框图



注意：TBPHSHR 和 TBPRDHR 来自 PWM TB 时基子模块，CMPAHR 和 CMPBHR 来自 PWM CC 计数器比较子模块，DBREDHR 和 DBFEDHR 来自 PWM DB 死区发生器子模块。

32.4.2 HRPWM 系统接口结构框图

HRPWM 系统接口结构框图请参见 PWM 主要信号章节中的 PWM 模块和关键内部信号互联结构框图。

32.5 功能描述

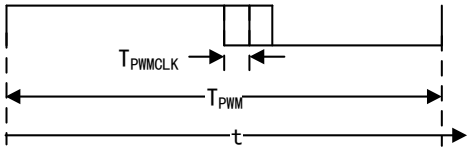
32.5.1 PWM 分辨率计算

PWM 外设用于实现数学上等价于数模转换器(DAC)的功能。常规生成的 PWM 时序图如图所示，其有效分辨率与 PWM 频率或周期和系统时钟频率有关。常规生成的 PWM 的分辨率计算：

$$\text{PWM 分辨率} = \frac{F_{PWM}}{F_{PWMCLK}} * 100\%$$

$$\text{PWM 分辨率} = log_2 \frac{T_{PWM}}{T_{PWMCLK}}$$

图 140 常规生成的 PWM 时序图



如果所需的 PWM 工作频率不能在 PWM 模式下提供足够的分辨率，则考虑使用 HRPWM。假设常规分辨率在 PWMCLK 为 100MHz 条件下且 HRP 步进为 180 ps，不同 PWM 频率下，以位数为单位的分辨率如下表格所示。HRP 的典型和最大性能指标请参见数据手册。

表格 155 不同频率下的 PWM 和 HRPWM 分辨率

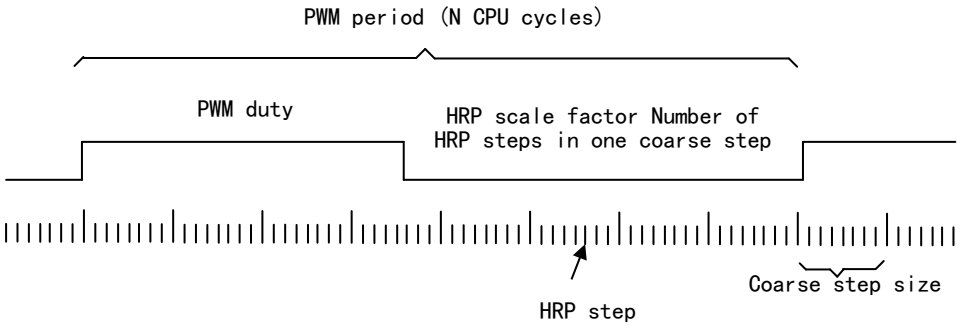
PWM 频率	PWM 位数	PWM 百分比	HRPWM 位数	HRPWM 百分比
20kHz	12.3	0.020%	18.1	0.000%
50kHz	11	0.050%	16.8	0.001%
100kHz	10	0.100%	15.8	0.002%
150kHz	9.4	0.150%	15.2	0.003%
200kHz	9	0.200%	14.8	0.004%
250kHz	8.6	0.250%	14.4	0.005%
1000kHz	6.6	1.000%	12.4	0.018%
1500kHz	6.1	1.500%	11.9	0.027%
2000kHz	5.6	2.00%	11.4	0.036%

32.5.2 HRP 技术

HRPWM 基于 HRP 技术，且 HRP 逻辑能够通过细分常规 PWM 发生器的粗系统时钟来非常精确地放置边沿，其时间步进精度为 150ps 数量级。特定设备上典型的 HRP 步进请参见数据手册。自检诊断软件模式可以检查 HRP 逻辑是否按照设计正常运行。有关软件诊断和函数的详细信息请参见 SDK 说明文档。

HRP 逻辑图如下所示，HRP 步进由 CMPAHR 寄存器的高 8 位控制且 CMPBHR 寄存器也适用同样的操作逻辑。控制 CMPA 寄存器值等于粗调步进的数量，CMPAHR 寄存器值等于微调步进的数量左移 8 位，然后将结果存储在该寄存器值的高 8 位中。

图 141 HRP 逻辑图



粗步进与 HRP 步进计算公式：

- 粗步进=int（PWM 周期 * PWM 占空比）
- HRP 步进= fraction（PWM 周期 * PWM 占空比） * HRP 比例系数 +0.5。其中 0.5 用于范围和四舍五入的调整（考虑 0x0080 的 Q8 格式）

需要配置 PWM 寄存器以产生给定频率和极性的常规 PWM 才能产生 HRPWM 波形。为了扩展边沿的分辨率，可以使 PWM 寄存器与 HRPWM 一同工作。在多种编程组合中，只有少数为所需且实用的编程组合，详情请参见 SDK 说明文档。

32.5.3 控制 HRPWM 功能

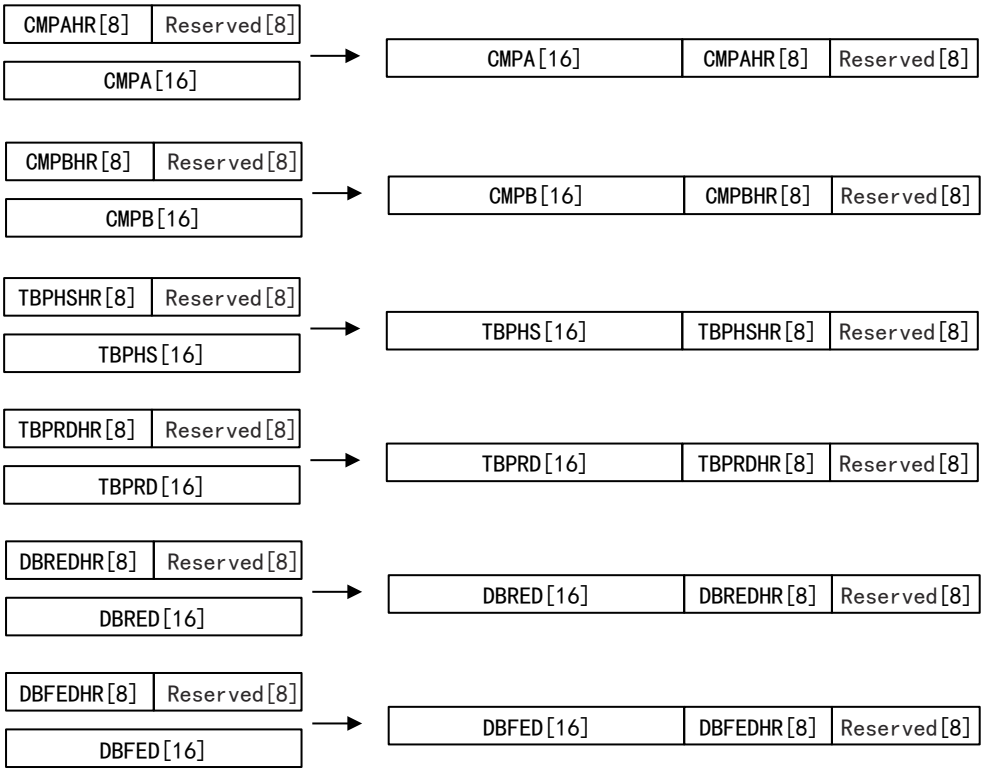
通过以下扩展寄存器控制 HRPWM 的 HRP，这些寄存器与用于控制 PWM 操作的 CMPA、CMPB、TBPHS、TBPRD、DBRED 和 DBFED 寄存器连接。

表格 156 HRPWM 扩展寄存器

寄存器名	描述
CMPAHR	计数比较 A 高分辨率扩展寄存器，该寄存器用于动作限定通道 A 的输出，与 CMPA 寄存器无关
CMPBHR	计数比较 B 高分辨率扩展寄存器，该寄存器用于动作限定通道 B 的输出，与 CMPB 寄存器无关
TBPHSHR	高分辨率时基相位偏移寄存器
TBPRDHR	高分辨率时基周期寄存器，只适用于部分器件
DBREDHR	死区上升沿延迟高分辨率寄存器
DBFEDHR	死区下降沿延迟高分辨率寄存器

注意：不能使用 TBPHSHR 寄存器，必须通过 TRREM 寄存器来模拟 TBPHSHR 功能。

图 142 HRPWM 扩展寄存器和内置配置



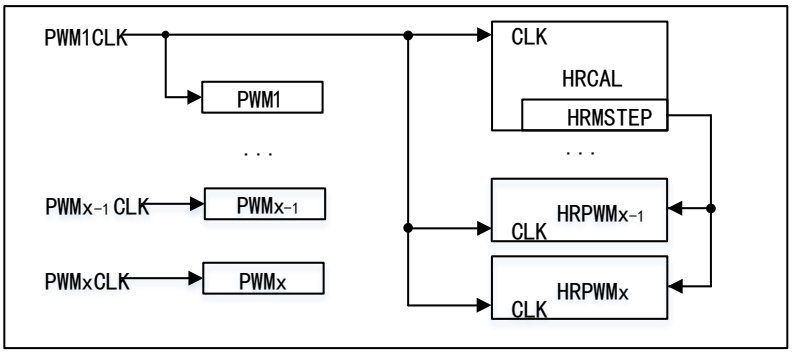
死区上升沿延迟和下降沿延迟的 HRPWM 能力仅适用于死区半周期时钟操作。HRP 步进的数量位[15:9]是占空比和相位高分辨率寄存器尺寸的一半。

HRPWM 功能是通过通道 A 和 B 的 PWM 信号路径进行控制。通过正确配置 HRCNFG2 寄存器，可以在死区信号路径上支持 HRPWM。HRPWM 与 8 位扩展寄存器的接口结构框图请参见 PWM 主要信号章节。

32.5.4 时钟源

通过 PWM1CLK 时钟来驱动 HRPWM 模块和 HRCAL 模块，因此，必须先使能 PWM1CLK 才能使能 HRCAL 或任何 HRPWM 模块。

图 143 HRPWM 和 HRCAL 时钟源



32.5.5 配置 HRPWM

如果将 PWM 配置为常规 PWM（提供给定频率和极性），可以通过编程

HRCNFG 寄存器来配置 HRPWM。HRCNFG 寄存器可以提供以下配置选项。

边沿模式

可以编程的 HRP 为上升沿（RE）、下降沿（FE）或双边沿（BE）提供了精确的位置控制。其中，FE 和 RE 由 CMPAHR 或 CMPBHR 寄存器控制，用于需要占空比控制的电源拓扑，BE 用于需要相移的拓扑，例如 TBPRDHR 或 TBPHSHR 寄存器控制的相移全桥。

控制模式

可以将 HRP 编程为以下两种选择：

- 在占空比控制情况下，由 CMPAHR 或 CMPBHR 寄存器控制
- 由 TBPHSHR 寄存器控制

CMPAHR 或 CMPBHR 寄存器可以与 RE 或 FE 控制模式同时使用。TBPHSHR 寄存器可以与 BE 控制模式同时使用。当在周期控制 TBPHSHR 寄存器情况下控制 HRP 时，也可以通过各自的高分辨率寄存器来控制占空比和相位。

影子模式

提供了与常规 PWM 模式相同的双缓冲/影子化选项，只有在由 CMPxHR 和 TBPHSHR 寄存器操作时有效且可以选择与 CMPA/CMPB 寄存器的常规加载选项相同。在使用 TBPHSHR 寄存器时无效。

控制高分辨率 B 信号

PWM 通道的 B 信号可通过 PWMxB 引脚输出 PWMxA 信号的反相版本来实现高分辨率输出。HRPWM 模块支持在独立于 A 信号的 B 信号路径上实现高分辨率功能。

交换高分辨率输出

该模式允许交换高分辨率 A 和 B 输出。模式可以选择以下方式：

- 允许 A 和 B 输出信号保持不变
- 交换 A 和 B 输出信号，即将 A 输出到 B，B 输出到 A

自动转换模式

自动转换模式只能与 SFO 软件一起使用。

使能自动转换：CMPAHR= [frac（PWM 周期 * PWM 占空比）*256]<<8

- 在后台代码中，SFO 软件将计算 HRP 比例系数，并将计算出的每粗步进的 HRP 步进数自动更新到 HRMSTEP 寄存器中。
- HRP 校准模块通过使用 HRMSTEP 和 CMPAHR 寄存器中的值自动计算合适的 HRP 步进数（表示为小数的占空比），并相应地移动高分辨率 PWM 信号边沿。

禁止自动转换：CMPAHR=[frac（PWM 周期 * PWM 占空比）* HRP 比例系数+

0.5]<<8

- 在该模式下，所有计算都需要由用户的代码执行且 HRMSTEP 寄存器将被忽略。
- 高分辨率周期与高分辨率占空比的自动转换行为相同。
- 如果使能高分辨率周期模式，则必须始终使能自动转换。

注意：当 HRPWM 模块配置为中心对齐计数模式时，必须将 HRPWM 寄存器的影子模式设置为在 TBCTR=0 和 TBCTR=TBPRD 时装载。只有当 TBCTR=0 时，用户的新值才会加载到影子寄存器中（寄存器的影子模式必须同时设置为 TBCTR=0 和 TBCTR=TBPRD）。TBCTR=TBPRD 事件用于 HRPWM 模块的特定内部逻辑。

自动转换模式执行 CMPBHR、DBREDHR 和 DBFEDHR 的计算：

使能自动转换：CMPAHR= [frac（PWM 周期 * PWM 占空比）*256]<<8

- 在后台代码中，SFO 软件将计算 HRP 比例系数，并将计算出的每粗步进的 HRP 步进数自动更新到 HRMSTEP 寄存器中。
- HRP 校准模块通过使用 HRMSTEP 和 CMPBHR / DBREDHR/DBFEDHR 寄存器中的值自动计算合适的 HRP 步进数（表示为小数的占空比），并相应地移动高分辨率 PWM 信号边沿。

禁止自动转换：CMPBHR 行为与 CMPAHR 一致，即 CMPBHR=[frac（PWM 周期 * PWM 占空比）* HRP 比例系数+ 0.5]<<8

32.5.6 工作原理

HRP 逻辑能够将边沿放置在 255 个（8 位）离散时间点之一上，典型 HRP 步进请参见数据手册。为了确保应用时间步进以及各种条件（宽范围的 PWM 频率和系统时钟频率等）下保持边沿的放置精度，HRP 与 TBM 和 CCM 寄存器一起工作。HRPWM 支持的典型频率范围如下所示。

表格 157 HRPWM 支持的典型频率范围

系统时钟（MHz）	最大分辨率（位数）(1)	每 PWMCLK 的 HRP 步进 (2) (3)	PWM 频率	
			最低（Hz）(4)	最高（MHz）
60.0	10.9	93.0	916.0	3.0
70.0	10.6	79.0	1068.0	3.5
80.0	10.4	69.0	1221.0	4.0
90.0	10.3	62.0	1373.0	4.5
100.0	10.1	56.0	1526.0	5.0

注意：

（1） 所给出以位数为单位的分辨率，是针对于所述的最大 PWM 频率

（2） TBCLK=PWMCLK

（3） 表中数据基于 180ps 的 HRP 时间分辨率，其中 HRP 步进=T_{PWMCLK}/180ps。HRP 的范围请参见数据手册。

（4） PWM 为非对称向上计数模式且 PWM 最低频率是基于一个最大周期值 65535。

边沿放置

数字控制器在一个典型的功率控制回路中发出一个以每单位或百分比表示的占空比命令。如果一个工作点要求占空比为 40.5%，要求变换器 PWM 频率为 1.25 MHz。在常规 PWM 生成中（系统时钟为 100MHz），占空比的选择大约是 40.5%左右。如下表所示，占空比最接近 40.5%数值的是在计数值 32，此时占空比为 40%，这相当于边沿放置在 320ns。

表格 158 CMPA 与占空比（左三列）以及[CMPA:CMPAHR]与占空比（右四列）

CMPA 计数值（1）（2）（3）	高电平时间	占空比	CCOMPACMPA 计数值	CCOMPAHRCMPAHR 计数值	高电平时间	占空比
28	280 ns	35.0%	32	18	323.24ns	40.405%
29	290 ns	36.3%	32	19	323.42ns	40.428%
30	300 ns	37.5%	32	20	323.60ns	40.450%
31	310 ns	38.8%	32	21	323.78ns	40.473%
32	320 ns	40.0%	32	22	323.96ns	40.495%
33	330 ns	41.3%	32	23	324.14ns	40.518%
34	340 ns	42.5%	32	24	324.32ns	40.540%
...	...	...	32	25	324.50ns	40.563%
32.4	324ns	40.5%	32	26	324.68ns	40.585%

注意：

（1） TBCLK 为 100MHz，10ns。

（2） HRP 的步进分辨率为 180ps，HRP 典型值和最大值请参见数据手册。

（3） 对于 PWM 周期寄存器值为 80，T_{PWM}=80×10ns=800ns，F_{PWM}=1/T_{PWM}=1.25MHz。

假设 HRP 的步进分辨率为 180ps，利用 HRP 可以得到更接近期望点 324ns 的边沿位置。在 CMPA 计数值之外，HRP 的 22 步进将边沿放置在 323.96ns，导致几乎零误差。

使用[CMPB:CMPBHR]进行占空比控制时，其操作方式以及公式与[CMPA:CMPAHR]寄存器组合一致。

缩放的考量

使用 CMPA 和 CMPAHR 寄存器资源来演示如何在时间上精确定位边沿的机制。在实际应用中，需要将小数的占空比写入到最终的整数（[CMPA:CMPAHR]寄存器组合）表示，并无缝地将该映射函数提供给 CPU。

要做到这一点，首先所涉及的映射或缩放步骤都需要进行检查。在控制软件中，通常使用每单位或百分比表示占空比。有利于所执行所有的数学计算都无需关心最终用时钟计数或 ns 表示高电平时间的绝对占空比。此外，它使代码更易于兼容不同 PWM 频率的多种转换器类型。

假设条件：

- TBCLK=10ns（100MHz）
- FPWM=1.25MHz
- PWM 占空比为 40.5%
- 粗步进的 PWM 周期为 80
- 每个粗步进的 HRP 步进为 180ps 时，HRP 比例系数为 55
- 用于将 CMPAHR 保持在 1-255 之间和小数舍入的常数为 0.5（默认值）

需要以下两步缩放过程便于实现映射方案：

- CMPA 寄存器的百分比整数部分占空比转换
  - CMPA 寄存器值 = int（PWM 周期 * PWM 占空比）=32
  - CMPAHR 寄存器值 = 32 = 0x20
- CMPAHR 寄存器小数部分转换
  - CMPAHR 寄存器值 = [frac（PWM 周期 * PWM 占空比）* HRP 比例系数+ 0.5]<<8 =22.5*256=5760，即 0x1680
  - CMPAHR 寄存器值 = 0x1680，也即 0x1600，因为硬件忽略低八位比特

当 AUTOCONV 置位且 HRP 比例系数在 HRMSTEP 寄存器中时，CMPxHR 寄存器值 = frac（PWM 周期 * PWM 占空比）。由硬件自动完成其余转换计算，且 PWM 通道将输出正确的经 HRP 缩放的信号边沿。当 AUTOCONV 未置位时，必须通过软件执行上述计算。

HRP 比例系数受系统时钟和 DSP 运行情况的影响。Geehy 提供的 HRP 比例系数 C 函数利用 HRPWM 的内置诊断返回最佳比例系数，因其变化缓慢，优化过程可以在后台循环中进行。

CPU 的 32 位数据能力可以将在内存中配置的 CMPx 和 CMPxHR 寄存器作为单个拼接的值来写入，即[CMPA:CMPAHR]、[CMPB:CMPBHR]等。

注意：

（1） 实际实现利用 Arm 的 32 位 CPU 体系结构，与上述步骤有所不同。

（2） 对于连一个周期都要考虑的时间关键控制循环，建议使用优化了周期的函数（汇编版本）。它接受一个 Q15 占空比值作为输入并写入单个[CMPA:CMPAHR]值。

限制 HRP 占空比

在高分辨率模式下，HRP 并非在 100%的 PWM 周期内都有效。HRP 可工作在以下情况：

（1） 未使能高分辨率 PWM 周期控制
- HRP 在周期开始的三个 PWMCLK 周期之后工作

（2） 使用 HRPCTL 寄存器使能高分辨率 PWM 周期控制
- 向上计数模式
  - HRP 工作于周期开始的三个 PWMCLK 周期后到周期结束的三个 PWMCLK 周期之间

- 中心对齐计数模式
  - 向上计数时，HRP 工作于 TBCTR=0 三个周期后到 TBCTR=TBPRD 三个周期之间
  - 向下计数时，HRP 工作于 TBCTR=TBPRD 三个周期后到 TBCTR=0 三个周期之间

（3） 使用 DBREDHR 或 DBFEDHR

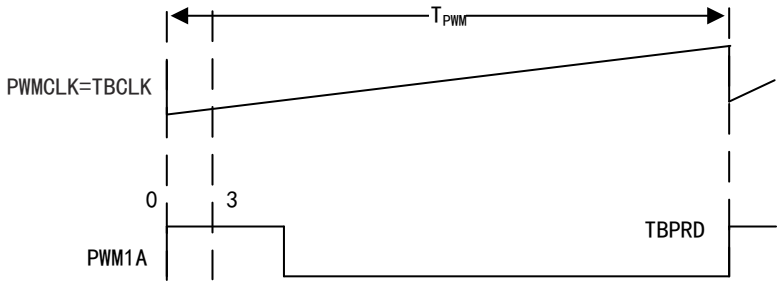
- DBRED 或 DBFED 对应边沿高分辨率放置的寄存器必须大于或等于 7

占空比范围限制如下图所示，它限制了 HRP 的占空比。例如，在占空比为 0%时，不可以使用精确的边缘控制。如果禁止高分辨率 PWM 周期控制，则可以使常规的 PWM 占空比控制完全工作到 0%占空比，即使 HRPWM 功能在前三个周期中不可用。但在大多数应用中，控制器调节点通常不会设计为接近 0%的占空比，所以可以避免该问题。如果使能高分辨率 PWM 周期控制，则占空比不能落入被限制的范围内，否则，可能在 PWMxA 输出上存在未定义的行为。

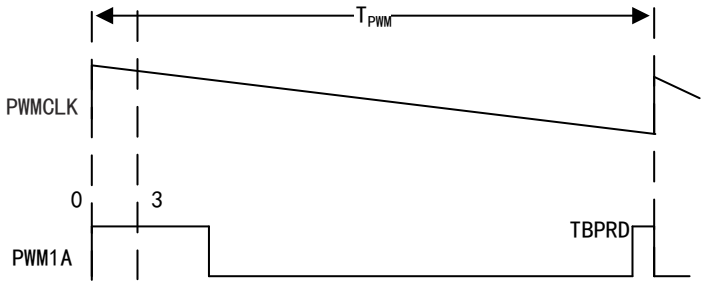
当应用要求 HRPWM 工作到最小占空比限制以下时，可以配置 HRPWM 工作在向下计数模式，可以在 HRPE=0 时，上升沿位置由 HRP 控制，如下图所示，可以在这种配置下解决最小占空比限制的问题，但存在一个相同百分比的最大占空比限制，三个周期的占空比范围限制如下表所示。

图 144 占空比范围限制

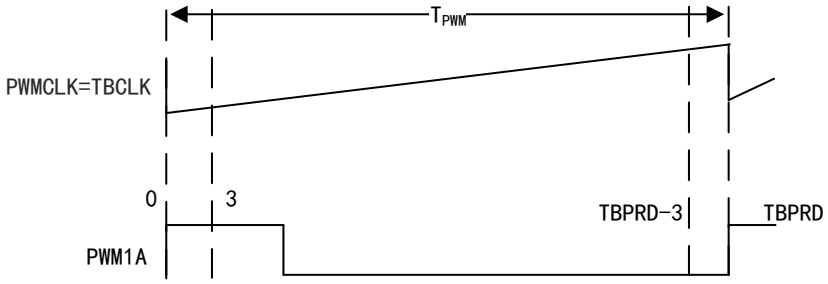
低%占空比范围限制(HRPE = 0)



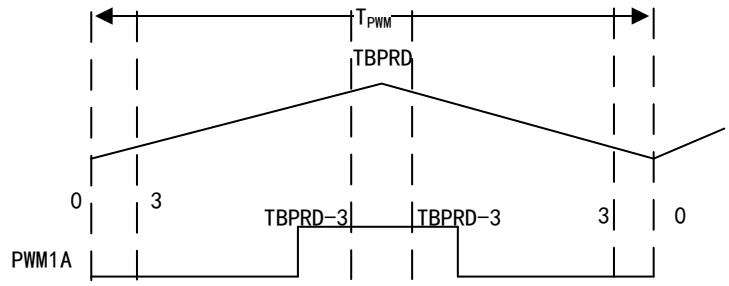
高%占空比范围限制(HRPE = 0)



上计数器占空比范围限制(HRPE = 1)



上下计数器占空比范围限制(HRPE = 1)



表格 159 三个周期的占空比范围限制

PWM 周期 (1)	占空比	
	最小值	最大值 (2)
200kHz	0.6%	99.4%
400kHz	1.2%	98.8%
600kHz	1.8%	98.2%
800kHz	2.4%	97.6%
1000kHz	3%	97%
1200kHz	3.6%	96.4%
1400kHz	4.2%	95.8%
1600kHz	4.8%	95.2%
1800kHz	5.4%	94.6%
2000kHz	6%	94%

注意：

（1） TBCLK=PWMCLK=100MHz

（2） 该限制仅适用于使能高分辨率 PWM 周期控制时

高分辨率周期

如果需要使用高分辨率 PWM 周期，则 PWM 模块必须按照特定顺序进行初始化。

将 CMPA 及其影子寄存器和相应的 HRCNFG 位域一起用于 PWMxA 上的高分辨率操作步骤如下所示。将 A 通道替换为 B 通道的对应寄存器即可进行 PWMxB 的高分辨率操作。

表格 160 高分辨率 PWM 周期操作步骤

步骤	描述
1	使能 PWM 时钟
2	使能 HRPWM 时钟
3	禁止同步 PWM 时钟（PWMCLKSYNC）
4	配置 PWMx 寄存器： （1）PWMx 仅可配置为向上或中心对齐计数模式。高分辨率 PWM 周期与向下计数模式不兼容。 （2）TBPRD 和 CCOMP 寄存器必须配置为影子装载。 （3）CMPCTL [LOADAMODE] 向上计数模式：LOADAMODE=01，在 TBCTR=TBPRD 时装载 中心对齐计数模式：LOADAMODE=10，在 TBCTR=0 或 TBCTR=TBPRD 时装载
5	配置 HRCNFG 寄存器： （1）HRLOAD=10，在 TBCTR=0 或 TBCTR=TBPRD 时装载

步骤	描述
	（2）使能自动转换（AUTOCONV=1） （3）HRP 逻辑同时控制 A 路信号两个边沿（AEMSEL=11）
6	高分辨率 PWM 周期的 TBPOF：TBPHSHR 同步，将 LDTBCNTVEN 位和 HRPHASYNCEN 位置位。在中心对齐计数模式下，不管 TBPHSHR 寄存器中任何内容，这些位段都必须置位。
7	使能高分辨率 PWM 周期控制（HRPE=1）
8	使能同步 PWM 时钟
9	软件强制产生一个同步脉冲（SPFSW=1）
10	HRMSTEP 在使能自动转换时必须包含精确的 HRP 比例系数，即每个 PWMCLK 粗步进的 HRP 步进数，该系数可以通过 SFO 库软件章节描述的 SFO()函数获得。
11	写入 PWM_TBPRDHR 寄存器可以控制高分辨率 PWM 周期

如果使能高分辨率周期模式，则 PWMx 同步脉冲，在向上计数模式下，给 PWM 带来±1 TBCLK 周期的抖动，在中心对齐计数模式下，给 PWM 带来±2 TBCLK 周期的抖动。所以不能将 TBCTR=0 或 TBCTR=CMPB 设置为同步输出信号源。当 TBCTL=0 时，软件同步脉冲只能在高分辨率周期初始化时发出一次；在 PWM 运行时应用同步脉冲会导致输出抖动。

若仅在 PWMxA 或 PWMxB 上使能高分辨率 PWM 周期控制，在向上计数模式下，非高分辨率输出具有±1 TBCLK 周期的抖动，在中心对齐计数模式下，具有±2 TBCLK 周期的抖动。

限制 HRP 占空比小节中描述的占空比换算程序也是用与高分辨率 PWM 周期。

假设条件：

- TBCLK=10ns（100MHz）
- FPWM=1.75MHz
- 周期为 571.482
- 每个粗步进的 HRP 步进为 180ps 时，HRP 比例系数为 55
- 用于将 TBPRDHR 保持在 1-255 之间和小数舍入的常数为 0.5（默认值）

表格 161 问题

模式	TBPRD	FPWM	TPWM
向上计数模式	570	175.13kHz	571*TTBCLK
	571	174.82kHz	572*TTBCLK
中心对齐计数模式	285	175.44kHz	285*2*TTBCLK
	286	174.82kHz	286*2*TTBCLK

解决方案（在 180ps 下，每粗步进有 55 HRP 步进时）：

（1） TBPRD 寄存器的百分比整数周期值转换

- 整数周期值= int（PWM 周期）* TTBCLK=571*TTBCLK
- 向上计数模式：TBPRD=周期值-1=570=0x23A

- 中心对齐计数模式：TBPRD=周期值/2=285=0x11D

（2） TBPRDHR 寄存器小数值转换

- 中心对齐计数模式：TBPRDHR 寄存器值 = frac（PWM 周期 * HRP 比例系数+ 0.5）
  - 当使能自动转换且 HRMSTEP =HRP 比例系数=55 时，TBPRDHR 寄存器值 = frac（PWM 周期）<<8，将值移动到 TBPRDHR 的高字节，即 TBPRDHR 寄存器值 = frac（571.428）<<8=0x6D
  - 自动转换将自动执行计算，从而 TBPRDHR HRP 延迟按硬件缩放到=[（TBPRDHR[15:0] >> 8 ）* HRMSTEP + 0x80] << 8 =0x6D * 55+ 0x80=0x17EB >> 8
  - 周期 HRP 延迟=0x17 HRP 步进
- 中心对齐计数模式：TBPRD= frac（PWM 周期 * HRP 比例系数+ 0.5）
  - 当使能自动转换且 HRMSTEP =HRP 比例系数=55 时，TBPRDHR 寄存器值 = frac（PWM 周期/2）<<8，将值移动到 TBPRDHR 的高字节，即 TBPRDHR 寄存器值 = frac（285.714）<<8=0xB6
  - 自动转换将自动执行计算，从而 TBPRDHR HRP 延迟按硬件缩放到=[（TBPRDHR[15:0] >> 8 ）* HRMSTEP + 0x80] << 8 =0xB6 * 55+ 0x80=0x279A >> 8
  - 周期 HRP 延迟=0x27 HRP 步进

32.5.7 高分辨率死区

配置高分辨率死区上升沿延迟和下降沿延迟

如果将 PWM 配置为提供给定频率、极性的常规 PWM，并在半周期时钟模式下使能死区，则通过在该 PWM 模块寄存器空间中对 HRCNFG2 寄存器进行编程，即可使能 DBRED 和 DBFED 的高分辨率操作。提供两种配置选项：

- 控制模式
  - 在高分辨率模式下，选择 DBRED 和 DBFED 在活动寄存器中加载影子值的时间事件
  - 选择脉冲以匹配 DBCTL[LOADREDMODE，LOADFEDMODE]位中的选择
- 边沿模式
  - HRP 可通过编程实现对 DBRED、DBFED 或同时对死区的双边沿，即 DBRED 信号上升沿和 DBFED 信号下降沿，进行精确的位置控制

高分辨率死区操作

在向上模式下，死区模块在使能任何高分辨率模式情况下都不可以使用。

假设条件：

- TBCLK=10ns（100MHz）
- 在半周期模式下，使能死区：TBCLK=PWMCLK
- F_{PWM}=1.33MHz

- 占空比为 50%
- 死区上升沿延迟为占空比之外 5%
- 以 ns 为单位时，死区上升沿延迟 = 0.05*375=18.75ns

注意：与使用 HRPWM 时占空比限制一样，高分辨率死区必须在 DBRED 和 DBFED 的值大于 3 时才能使用。

死区延迟值是 DBRED 和 DBFED 的函数：

- 当使能半周期时钟时，RED 和 FED 计算公式变为：
  - RED= TBCLK/2 * DBRED
  - FED=TBCLK/2 * DBFED

DBRED 和 DBFED 计数值：

- 以 ns 为单位时，DBRED=18.75ns
- 所需 DBRED = RED/（TBCLK/2）=3.75ns

解决方案（每个粗步进的 HRP 步进为 180ps 时）：

- DBRED 寄存器的整数死区值转换
  - 整数 DBRED 值= int [RED/（TBCLK/2）]=3
- DBREDHR 寄存器的小数值转换
  - DBREDHR 值 = frac[（所需 DBRED）*HRP 比例系数+0.5]<<8= frac（3.75）*55+0.5<<8 = 41.75*256 = 0x29C HRP 步进，硬件会忽略上述计算得到的低 9 位 DBREDHR 值

注意：当 AUTOCONV 置位且 HRP 比例系数在 HRMSTEP 寄存器中时，DBREDHR：DBRED =frac（所需 DBRED）<< 8。由硬件自动完成剩余的转换计算，PWM 输出通道会出现正确的 HRP 缩放后的信号边沿。当 AUTOCONV 未置位时，必须由软件实现上述计算。

32.5.8 比例系数优化软件

HRP 逻辑能够在 255 个离散时间点之一上放置边沿。这些步进的尺寸约为 150ps 数量级，器件的典型 HRP 步进请参见数据手册。HRP 步进受最坏情况的工艺参数、工作温度和电压的影响，HRP 步进随电压升高、温度降低而减小，反之则增大。HRP 步进根据 HRPWM 特性的应用程序可以使用 Geehy 提供的 HRP 比例系数软件函数。在 HRPWM 运行时，SFO 函数有助于每个 PWMCLK 周期的步进计算。

HRPWM 模块内置自检和诊断功能，以找到最佳 HRP 比例系数。Geehy 还提供了一个 C 可调用的 SFO 库，帮助确定最佳比例系数。

有关 SFO 库软件（SFO_Geehy_Build_V8.Lib）的详细信息，请参见 SFO 库软件章节。

32.5.9 SFO 库软件

SFO 库的特征如下 ：

- SFO_Geehy_Build_V8.Lib 完成检查，其单位为函数返回值
- 在重复调用且没有中断的情况下，SFO()更新 HRP 比例系数所需的典型周期为 130000 PWMCLK 周期。

比例系数优化函数

该程序通过驱动 HRP 校准模块运行 SFO 诊断，能够在任何时刻为设备确定合适的 HRP 比例系数，即每个 PWMCLK 粗步进的 HRP 步进。假设 PWMCLK 和 TBCLK 均为 100MHz，且 HRP 步进为 150ps，那么在 100MHz 下，典型的比例系数是在每 10ns（TBCLK 单元）中有 66 个 HRP 步进。

该函数返回 HRP 比例系数值为：HRP 比例系数= 每 PWMCLK 的 HRP 步进。

（1） int SFO()的使用限制

SFO()最低可以在 PWMCLK 和 TBCLK 均为 50MHz 时使用。因为 HRP 诊断逻辑使用 PWMCLK，所以 SFO()受 PWMCLK 的限制。如果低于 50MHz，由于器件工艺的变化，则 HRP 步进可能会在低温和高核心电压条件下降低从而导致 255 个 HRP 步进无法跨越整个 PWMCLK 周期。

在 HRP 校准模块上，在任何时候都可以调用 SFO()运行 SFO 诊断。

（2） int SFO()的用法

如果 PWM 通道以 HRPWM 模式运行，则可以在后台随时调用该函数。其作用是对 HRP 校准模块中的诊断逻辑进行利用，然后将当时得到的比例系数结果可以应用于所有运行在 HRPWM 模式下的 PWM 通道。

这个函数返回值来表示不同情况：

- 该程序返回 00：表示校准仍在运行
- 该程序返回 01：表示校准结束并计算出新的比例系数
- 该程序返回 10：表示出现错误，此时 HRP 比例系数大于每个粗步进 PWMCLK 周期的最大 255 个细步进。在此情况下，HRMSTEP 寄存器将维持最后一个小于 256 的 HRP 比例系数值以进行自动转换

如果不使能高分辨率 PWM 周期控制，则所有以 HRPWM 工作的 PWM 模块的最小占空比限制为 3 个 PWMCLK 周期。如果使能该控制，则在 PWM 周期结束前有一个额外的 3 个 PWMCLK 周期占空比限制，详情请参见限制 HRP 占空比章节。

比例系数结果可以被 SFO()用来更新 HRMSTEP 寄存器。当 AUTOCONV 置位时，应用软件只在后台运行该函数时负责设置 CMPAHR= [frac（PWM 周期 * PWM 占空比）*256]<<8 或 TBPRDHR = frac（PWM 周期）。HRP 校准模块将根据 HRMSTEP 和 CMPxHR/TBPRDHR 寄存器中的值自动计算适当的 HRP 步进，该步进由分数占空比或周期表示，并相应调整 HRPWM 信号的边沿位置。

当 AUTOCONV 位被清除时，HRMSTEP 寄存器将被忽略。应用程序需要手动进行以下计算：

- CMPAHR = frac（PWM 周期*PWM 占空比）*HRP 比例系数 <<8+0x80
- CMPBHR、DBREDHR/DBFEDHR 以及 TBPHSHR 的操作方式相似，使用 TBPRDHR 时，必须启用自动转换功能

（3） 根据 driverlib 函数来使用 HRPWM 占空比的程序

图 145 HRPWM 占空比程序

```
float32_t dutyFine = 85.62;
float32_t count = (dutyFine * (float32_t)(PWM_TIMER_TBPRD << 8))/100;
uint32_t compCount = (count);
HRPWM_setCounterCompareValue(PWM1_BASE, HRPWM_COUNTER_COMPARE_A, compCount);
HRPWM_setCounterCompareValue(PWM1_BASE, HRPWM_COUNTER_COMPARE_B, compCount);
```

该程序可以作为后台任务在较慢的循环中运行，所需的 CPU 周期可以忽略不计。执行 SFO 函数的重复频率取决于应用程序的工作环境。与所有数字 CMOS 器件一样，HRP 的运行受温度和电源电压的变化的影响。但是，这些参数在大多数应用中变化很慢，通常可以每 5 到 10 秒执行一次 SFO 函数。如果想要更快速的变化，则需要更频繁地执行以适应应用。需要注意的是，执行 SFO 函数的重复频率没有上限，因此，该函数可以根据后台循环的速度快速执行。

如果使能 HRPWM 特性，则在 PWM 周期的前 3 个 PWMCLK 周期内，HRPWM 逻辑是无效的；如果使用 TBPRDHR，那么 HRPWM 逻辑在 PWM 周期的最后 3 个 PWMCLK 周期内将无效。在使用该配置运行应用程序时，当 HRPE=0 且 CMPx 寄存器值小于 3 个周期时，必须清零 CMPx 寄存器。当 HRPE=1 时，为了避免 PWM 信号上的任何意外跳变，CMPA 寄存器值不能低于 3 或高于 TBPRD - 3。

32.6 寄存器

HRPWM 寄存器请参见 PWM 寄存器章节。

33 捕获器（CAP）

33.1 术语全称、缩写描述

表格 162 术语全称、缩写描述

中文全称	英文全称	英文缩写
辅助脉宽调制器	Auxiliary Pulse-width Modulator	APWM

33.2 简介

捕获器用于对精准计时外部事件的系统中。捕获器可以解码来自占空比编码电流/电压传感器的电压或电流幅度，能够测量脉冲序列信号的占空比和周期以及位置传感器脉冲之间的经过时间，支持通过霍尔传感器感应到的带齿链轮等方式来测量旋转机械的速度。

33.3 主要特征

（1） 128:1 输入多路复用器

（2） 事件过滤器预分频

（3） 4 个 32 位事件时间戳寄存器

（4） 单次捕获最多四个时间戳事件

（5） 通过连续模式捕获在四深度的循环缓冲区中的时间戳

（6） 最多可用于四个序列时间戳捕获事件的边沿极性选择

（7） 可以对四个事件中的任何一个响应并生成中断

（8） 差分（Delta）模式时间戳捕获

（9） 绝对时间戳捕获

（10） 独立 DMA 触发器

（11） CAP 可以配置为单通道 PWM 输出（CAP 未用于输入捕获时）

（12） 每个 CAP 通道具有以下功能：

- 128:1 输入多路复用器
- 使用输入 X-BAR 连接捕获输入
- 在 APWM 模式下，输出 X-BAR 配置为输出
- 输入捕获信号预分频
- 选择四个捕获事件的独立边沿极性（上升沿/下降沿）
- 四个 32 位时间戳寄存器
- 32 位计数器
- 与外部事件同步的四阶测序器（模 4 计数器），CAP 引脚上升/下降沿

- WRPRT 保护
- 单独的 DMA 触发器
- 四个捕获事件任意一个的中断功能
- 可以对模计数器、事件过滤器和中断标志进行复位
- 在四深度的环形缓冲区中，控制连续时间戳捕获
- 模计数器状态寄存器用于指示模计数器当前状态
- 一次触发比较寄存器（两位），冻结捕获后 1-4 时间戳事件

（13） APWM 模式功能说明：

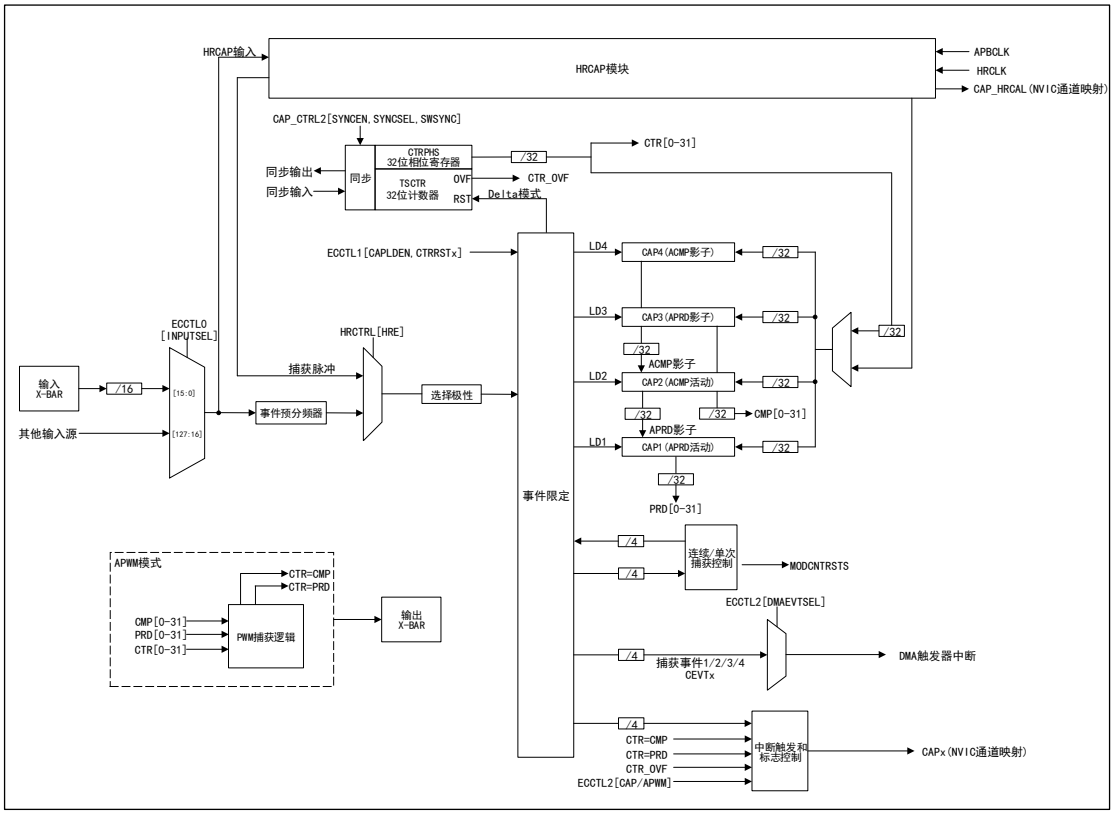
- 可以通过 2 个 32-位数字比较器来比较时间戳计数器总线
- 捕获寄存器 1/2 在 APWM 模式中，可以用作周期和比较值（未在 CAP 模式中使用捕获寄存器 1/2 时）
- 使用捕获寄存器 3/4（APRD 和 ACMP 影子寄存器）实现双缓冲，其内容可以在计数器等于周期触发时传输到捕获寄存器 1/2 或在写入时立即传输到捕获寄存器 1/2
- 在初始化期间，写入两个周期活动寄存器并进行比较，自动将初始值复制到阴影值中。使用影子寄存器可便于对运行过程中的后续进行比较更新
- 即时模式：在 APWM 模式中，写入活动寄存器（捕获寄存器 1/2）同时会将相同的值写入影子寄存器（捕获寄存器 3/4）。写入捕获寄存器 3/4 调用影子模式

（14） 中断事件：

- CTR=PRD
- CTR=CMP
- CEVTx（x=1...4）
- CTROVF

33.4 结构框图

图 146 CAP 结构框图



33.5 CAP 引脚功能描述

输入捕获

输入 X-BAR 将设备引脚连接到 CAP 模块作为输入，并且可以选择 CPU 上的任何 GPIO 作为输入源。可以在 GPxPUD 寄存器中配置内部上拉。可以通过设置 GPxQSELy 寄存器将 GPIO 输入配置为同步或异步模式。虽然同步输入可以抗噪，但是会影响 CAP±2 个周期的精度。在使用 GPIO 模式时，通过 GPxINV 寄存器可以对信号进行反转。

注意：为了能被 CAP 模块正确捕获，CAPxIN 必须有至少 2 个 APBCLK 循环宽度，否则 APBCLK 采样将错过输入脉冲。

多路复用器配置

类型 1 CAP 模块必须配置 128:1 输入复用器，如 CAP 结构框图所示。通过 ECCTL0[INPUTSEL]可以配置 CAP 输入源，详情请参见下面表格。

表格 163 多路复用器配置

选择多路复用器索引	配置 CAP 输入源
15:0	输入 X-BAR 的 INPUTn 信号（n=1...16）

选择多路复用器索引	配置 CAP 输入源
19:16	FLBx 的 OUTy 信号（x=1...4,y=14,15）
20	CANA_0 中断
21	CANB_0 中断
23:22	保留
31:24	输出 X-BAR 寄存器 OUTPUT 信号（x=1-8）
35:32	保留
39:36	ADCCINTx 信号（x=1...4）
43:40	ADCBINTx 信号（x=1...4）
47:44	ADCAINTx 信号（x=1...4）
63:48	保留
67:64	SD1FLTx 的 COMPL 信号（x=1...4）
71:68	保留
75:72	SD1FLTx 的 COMPZ 信号（x=1...4）
79:76	保留
83:80	SD1FLTx 的 COMPH 信号（x=1...4）
87:84	保留
91:88	SD1FLTx 的 COMPH 和 CTRIPL 相或运算信号（x=1...4）
95:92	保留
102:96	COMPx 的 CTRIPL 信号（x=1...7）
107:103	保留
114:108	COMPx 的 CTRIPH 信号（x=1...7）
119:115	保留
126:120	COMPx 的 CTRIPH 和 CTRIPH 相或运算信号（x=1...7）
127	保留

输出信号

输出 X-BAR 必须将输出信号连接到 OUTPUTXBARx 输出配置。GPIO 多路复用器必须配置为将 OUTPUTXBARx 连接到 GPIO 多路复用器的任意一个 IO 引脚。首先必须配置 GPyGMUX 寄存器，同时将 GPyMUX 寄存器中相应的位域保持为 0，然后再向 GPyGMUX 寄存器写入所需值。

有关 GPIO 多路复用器内容，详情请参见数据手册。有关 GPIO 设置和 X-BAR 配置内容，详情请参见 GPIO 章节。

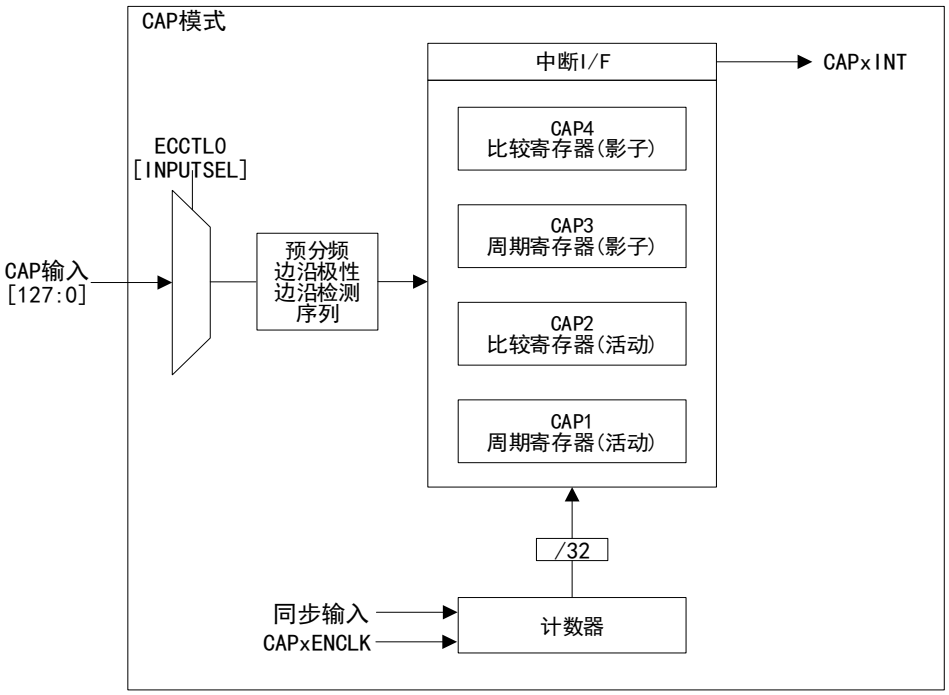
33.6 捕获描述功能描述

CAP1 寄存器成为活动周期寄存器，而 CAP3 寄存器成为其影子寄存器。CAP2 寄存器成为活动比较寄存器，而 CAP4 寄存器成为其影子寄存器。

注意：CAP 模式和 APWM 模式共享一个引脚。该引脚在捕获模式中为输入引脚。

33.6.1 CAP 模式结构框图

图 147 CAP 模式结构框图



33.6.2 事件过滤器预分频

可以通过 N＝2-62（2 的倍数）对输入捕获信号进行绕过预分频器或预分频，其结构框图与预分频函数时序图如下所示，通过 CTRFILTRESET 位对事件过滤器预分频进行复位。如果输入信号为较高频率信号，预分频器起到重要的作用。

图 148 事件过滤器预分频结构框图

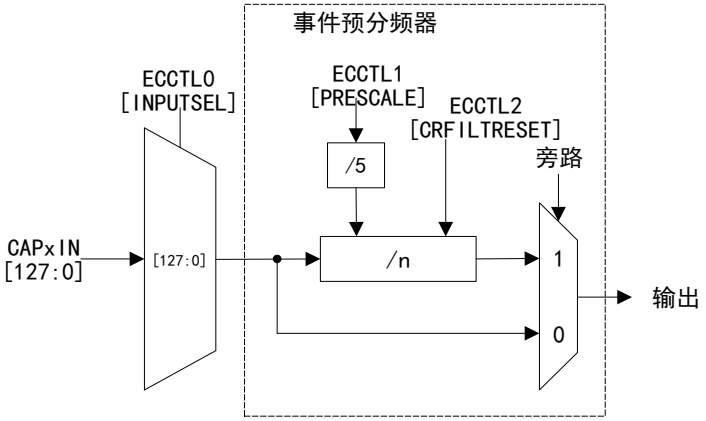
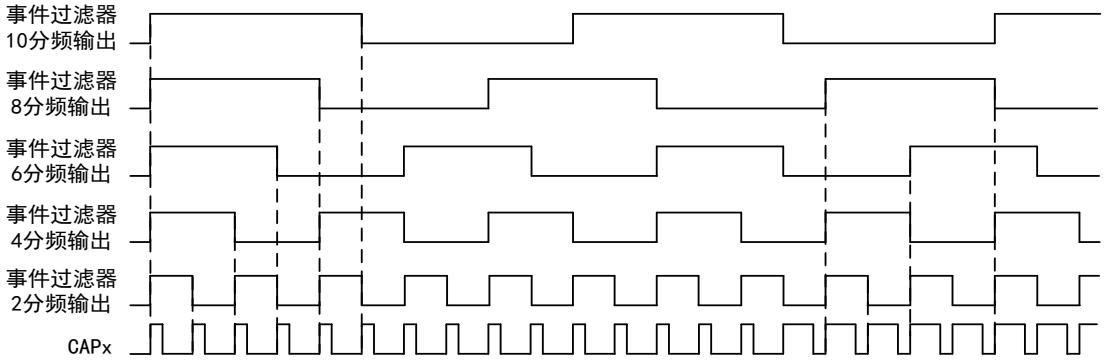


图 149 预分频函数时序图



如果 PRESCALE=00000 时，输入捕获信号则会绕过事件过滤器预分频。如果更改选择事件过滤器预分频配置，更改之后的第一个上升沿将会绕过预分频器，在更改之后的第二个上升沿才会生效。

33.6.3 选择边沿极性和事件限定

多路复用器由四个独立边沿极性（上升沿/下降沿）选择，用于四个捕获事件。在下降沿时，可以加载捕获寄存器，并通过模 4 计数器将边沿事件门控到各自的捕获寄存器。

模 4 序列器最多对四条边沿进行事件限定。

33.6.4 选择连续/单次模式

CAP 在连续/单次模式下的操作方式如下：

- 除非停止，否则模 4 计数器继续以 0->1->2->3->0 方式计数并循环
- 2 位模 4 计数器使用边沿限定事件 CEVTx（x=1...4）递增
- STOP_WRAP 在单次模式操作期间用于比较模 4 计数器输出，如果输出相等，则停止模 4 计数器并禁止捕获寄存器进一步装载。在该模式下，通过 CTRRSTx 位将时间戳计数器配置为在捕获事件 x 时复位计数器的情况下，如果计数器等于 STOP_WRAP 且未发生 REARM 置位，则仍会在捕获事件 x 时复位计数器。

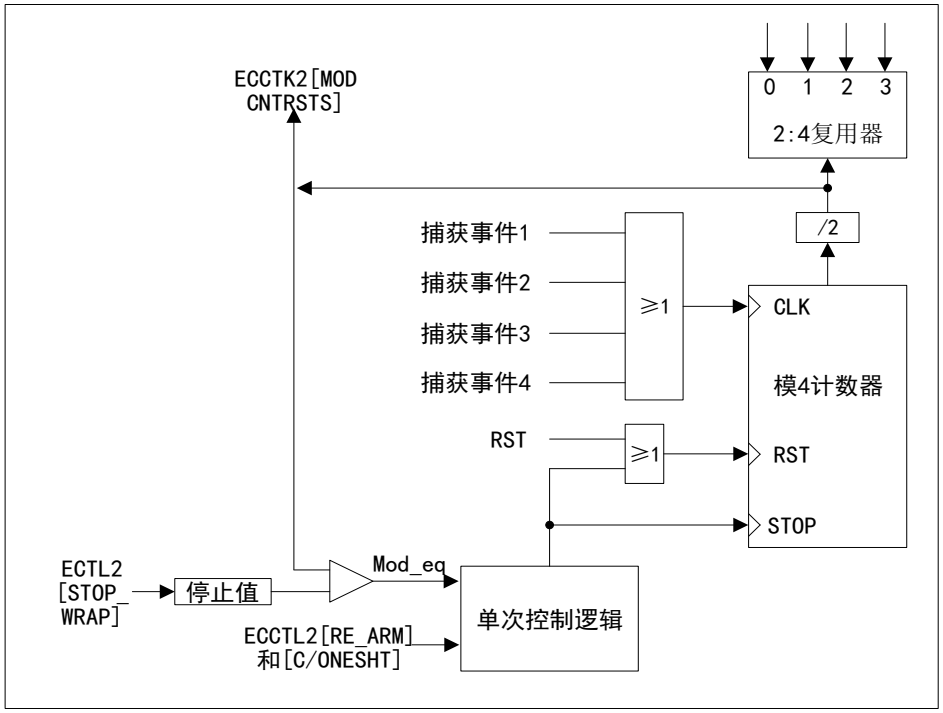
连续/单次块使用 mono-shot 类型动作来控制模 4 计数器的复位、启动和停止功能，可以通过停止值比较器来触发该动作，并使用软件控制重新加载。如果触发该动作，则 CAP 模块将会等待由停止值定义的捕获事件 1/2/3/4，并冻结捕获寄存器内容和模 4 计数器。

通过重新加载使 CAP 模块为捕获序列做好准备。当设置 CAPLDEN 位时，重新加载将使模 4 计数器清 0，并且允许在捕获事件上再次加载捕获寄存器。

在连续模式下，模 4 计数器继续以 0->1->2->3->0 方式运行，忽略单次模式操作，继续将捕获值以循环缓冲序列方式写入捕获寄存器。

注意：当用户在 CAP 初始化配置阶段开启计数后执行 reArm 操作时，如果此时存在外部信号输入，则通过中断读取数据可能导致比预期多捕获一次数据。这一现象与系统时钟、外部输入信号频率相关，建议用户在 CAP 初始化阶段避免执行 reArm 操作。

图 150 连续/单次块结构框图



33.6.5 计数器和相位寄存器

计数器使用 APB 时钟进行计时，且为捕获事件提供时基。

相位寄存器可以实现硬件和软件强制同步，以此实现与其他计数器的同步。

当使用 APWM 模式时，如果模块之间的相位需要偏移，可以使用这个功能。

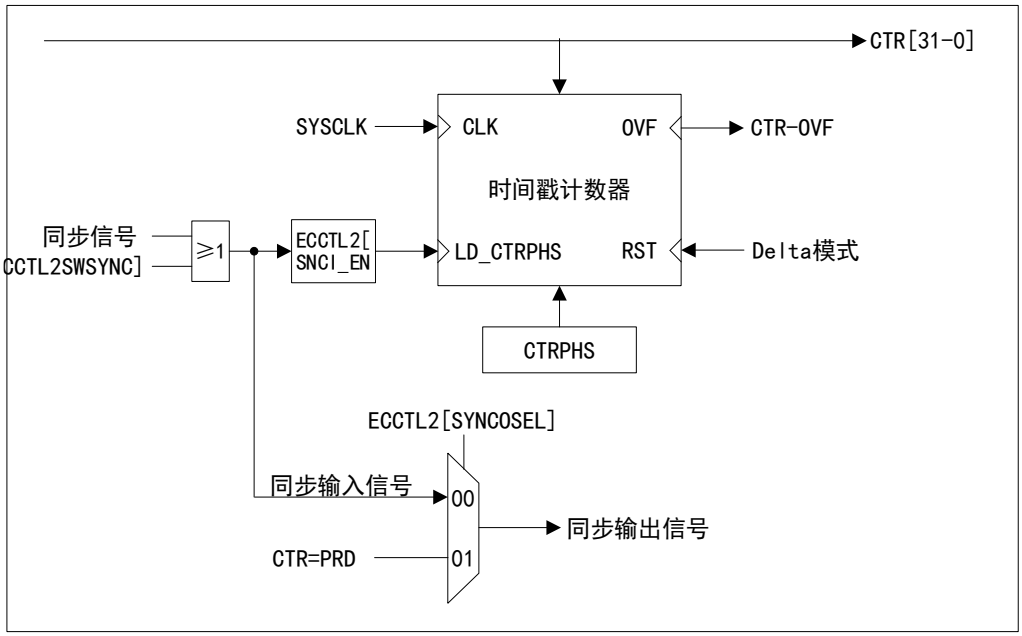
在四个事件装载中的任意一个，都可以对计数器进行复位，时差捕获可以使用这个功能。通过先捕获计数器值，然后使任何 LDx 信号将计数器值等于 0。

33.6.6 同步

CAP 模块通过选择公共同步输入源来实现模块之间相互同步。同步输入源可以选择来自 PWM、CAP 或 X-Bar 的外部同步输入信号或软件同步输入信号。如计数器和同步功能结构框图所示，CAP 模块的同步信号与 SWSYNC 需要进行逻辑“或”运算。通过 SYNCSELECT[CAPxSYNCIN]来选择同步信号。

注意：当从 CAPx 强制 SWSYNC 时，则同步脉冲需延迟一个时钟周期，用于接收同步脉冲 CAP。可以通过 CTRPHS 寄存器对同步脉冲接收 CAP 的周期延迟/滞后进行校正。

图 151 计数器和同步功能结构框图



同步信号

SWSYNC 与 CAP 模块配合使用操作步骤：

（1） 配置 CAP1 模块和 CAP3 模块的 SYNCO_SEL=0，选择同步输出信号为同步输入事件。

（2） 配置 CAP2 模块和 CAP3 模块的 SWSYNC=0，禁用 CAP2 到 CAP3 的软件同步。

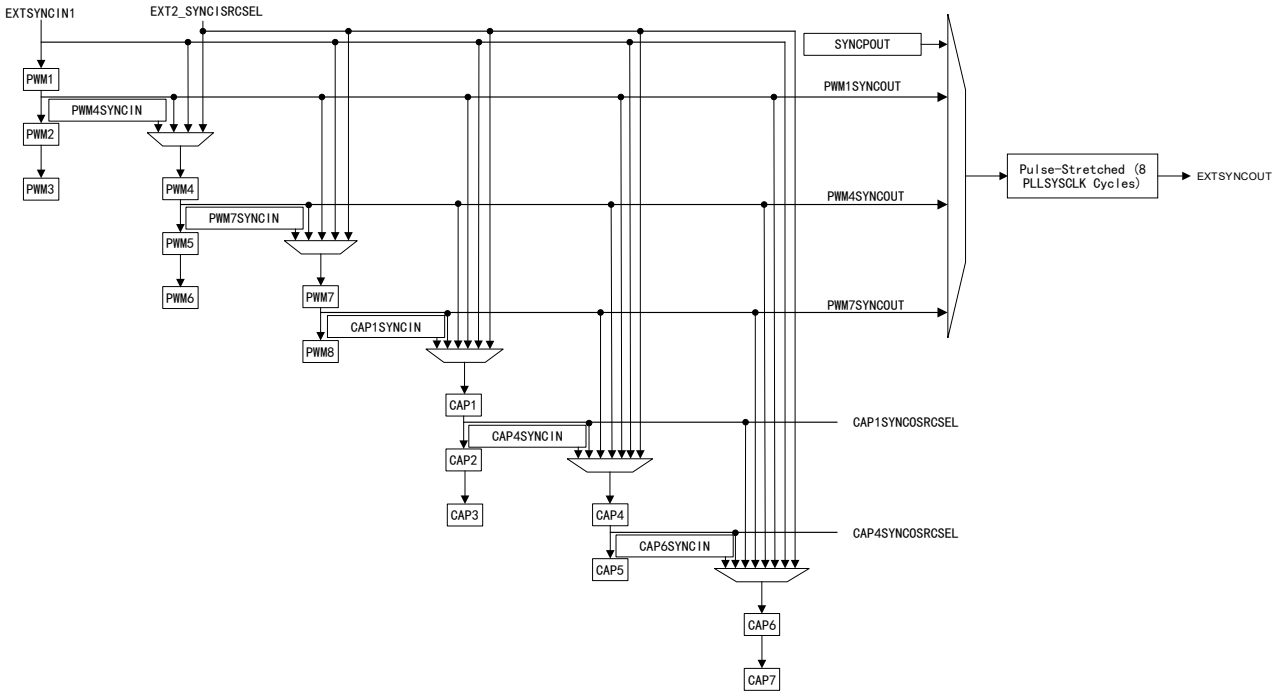
（3） 同步信号默认来自 PWM1，当配置 TBCTL[SYNCOSEL]不正确时，则会使 TSCTR 寄存器意外复位。在 INPUT5SELECT 寄存器中选择一个未使用的 GPIO，在输出模式下配置该 GPIO，并将 0 写入 GPxDAT 寄存器。由于它在默认情况下编程为 GPIO0，所以该引脚的任何活动都会导致 SWSYNC 出现错误。

（4） 配置 SYNCSEL[ECAP1SYNCIN]=101。使 CAPx.EXTSYNCIN 处于非活动状态。

（5） 配置 CAP1 模块的 SWSYNC=1，强制时间戳计数器软件同步。

如果要将其他 CAP 模块与 SWSYNC 一起使用，则需要确保上一个 CAP 链没有生成干扰软件同步的同步输出信号。

图 152 时基计数器同步方案



33.6.7 时间戳捕获寄存器

当时间戳捕获寄存器各自的 LD 输入被选通时，捕获寄存器由 32 位计数器定时器总线，CTR[0-31]和装载捕获时间戳提供。

CAPLDEN=0，禁用加载捕获寄存器。如果在单次模式操作期间发生停止条件，则自动清除 CAPLDEN 位，在捕获事件上禁止加载捕获寄存器且停止值=Mod4。

33.6.8 影子寄存器

在 APWM 模式下且影子装载处于活动状态时，允许以下选择：

- 周期相等时，PRD[31:0] = CTR[31:0]
- 将写入 APRD 或 ACMP 寄存器的新值立即转移到捕获寄存器 1 或捕获寄存器 2

在 CAP 模式下，该逻辑分别锁定来自 APRD 和 ACMP 寄存器中捕获寄存器 1 或捕获寄存器 2 的任何影子装载。

33.6.9 中断

产生中断的事件：

- 计数器溢出事件，即 FFFF FFFF->0000 0000
- APWM 事件（CTR=PRD，CTR=CMP）
- 捕获事件（CEVTx，CTROVF）

由于 G32R501 所使用 CPU 对于中断处理的特性，在 CAP 和 HRCAP 的中断程序里，不建议使用系统级 Clear Pending 语句，如果使用该语句，在高频捕获时可能造成程序运行中无法再次进入中断的问题，不使用则不会有该问题。

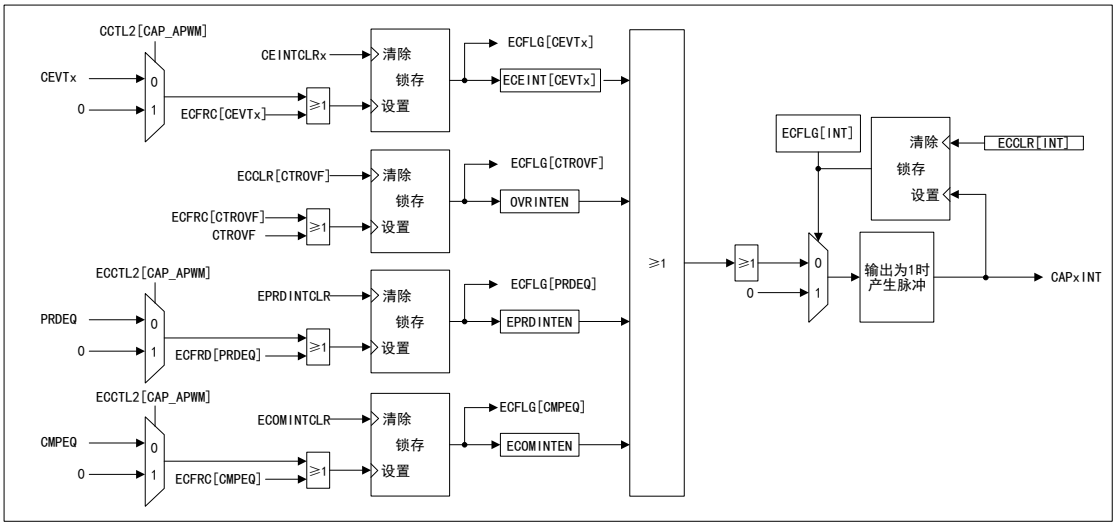
中断控制操作

根据时间排序，通过模 4 门控和极性选择对捕获事件进行边沿和序列器限定。可以选择来自 CAPx 模块任何一个事件作为中断源以此进入 NVIC。

只有使能任何中断事件，其 INT 为 0 且对应中断标志位为 1 时，才会使 NVIC 产生中断脉冲。中断服务程序必须在产生其他任何中断脉冲之前，通过使用 ECCLR 寄存器清除服务事件和全局中断标志位。在事件过滤器复位的情况下，可以通过 CTRFILTRESET 置位来清除所有中断标志。通过 ECFRC 寄存器可强制产生中断事件。

注意：CEINTFLG 仅在 CAP 模式（CAP_APWM=0）有效。CTR_PRD 和 CTR_CMP 仅在 APWM 模式（CAP_APWM=1）有效。CTROVF 在 CAP 模式和 APWM 模式下都有效。

图 153 中断



DMA

表格 164 类型 0 CAP 与类型 1 CAP 的 DMA 区别

CAP 类型	功能描述	区别
类型 0 CAP	CPU 需要使用 DMA 开始数据传输。	类型 1 CAP 增加了单独 DMA 触发器
类型 1 CAP	可以使用 DMA 触发器将捕获数据从 CAP 寄存器连续传输到片上存储器。 根据 DMAEVTSEL 可以选择四个中断事件 CEVTx（x=1...4）中任意一个作为 DMA 触发器的触发源。	

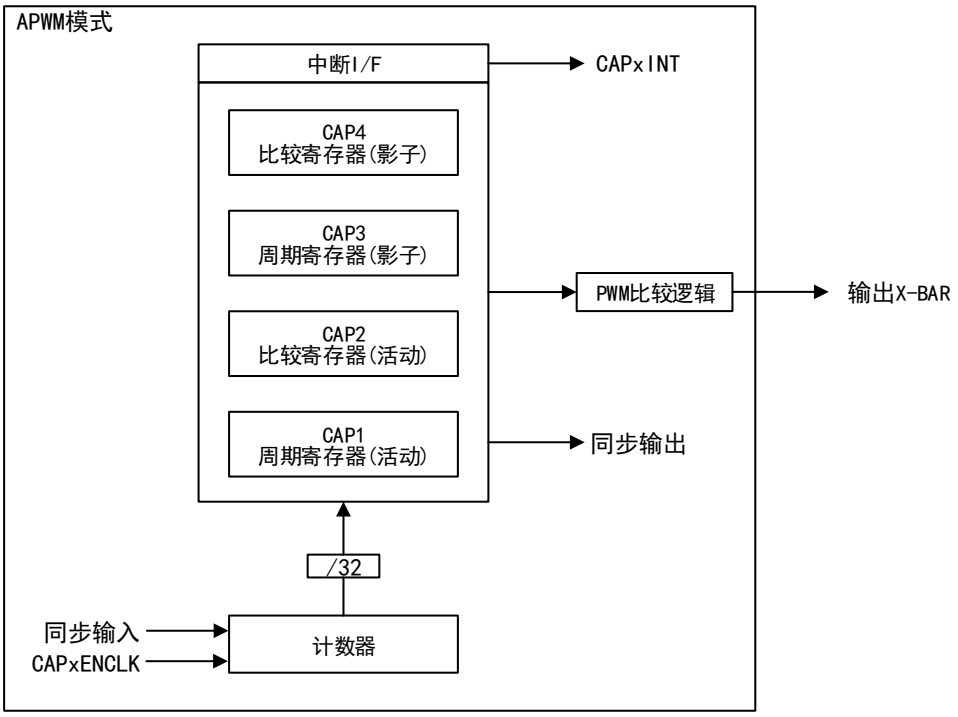
33.7 APWM 模式功能描述

33.7.1 APWM 结构框图

如果 CAP 模块未用作输入捕获，则可以将其应用产生一个单通道 PWM。当计数器处于向上计数模式时，为不对称 PWM 波形提供一个时基。写入捕获寄存器 3/4 可以调用影子模式。

注意：CAP 模式和 APWM 模式共享一个引脚。该引脚在 APWM 模式中为输出引脚。

图 154 APWM 结构框图



33.7.2 APWM 模式操作

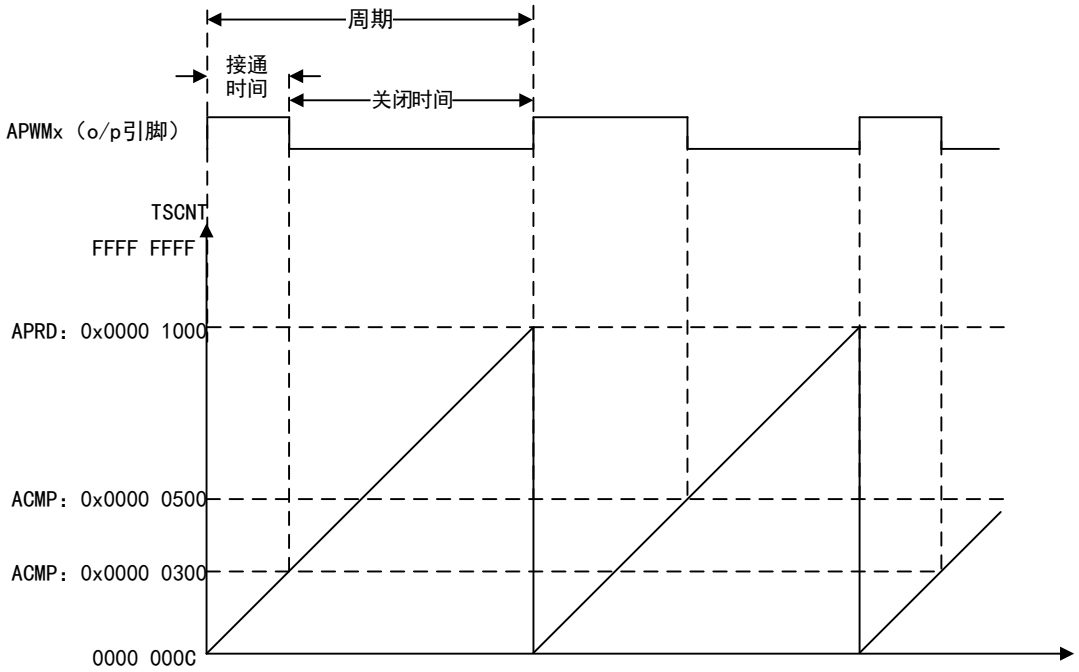
APWMx 输出引脚在 CAP 模块配置为 PWM 发生器时生成单通道 PWM 波形。当 APWMPOL=0，即 PWM 极性为高电平有效时，CAP2 比较寄存器的比较值表示该周期为高电平。当 APWMPOL=1 时，其比较值表示该周期为低电平。

表格 165 类型 0 CAP 与类型 1 CAP 的 DMA 区别

选择 APWM 输出极性	操作方式
APWMPOL=0	CMP=0x0000 0000，在一段周期内输出低电平，此时占空比为 0% CMP=0x0000 0001，输出高电平持续 1 个周期 CMP=0x0000 0002，输出高电平持续 2 个周期 CMP=周期，除了 1 个周期输出高电平，此时占空比小于 100% CMP=周期+1，整个周期都输出高电平，此时占空比为 100% CMP＞周期+1，整个周期都输出高电平
APWMPOL=1	CMP=0x0000 0000，在一段周期内输出高电平，此时占空比为 0% CMP=0x0000 0001，输出低电平持续 1 个周期 CMP=0x0000 0002，输出低电平持续 2 个周期 CMP=周期，除了 1 个周期输出低电平，此时占空比小于 100% CMP=周期+1，整个周期都输出低电平，此时占空比为 100% CMP＞周期+1，整个周期都输出低电平

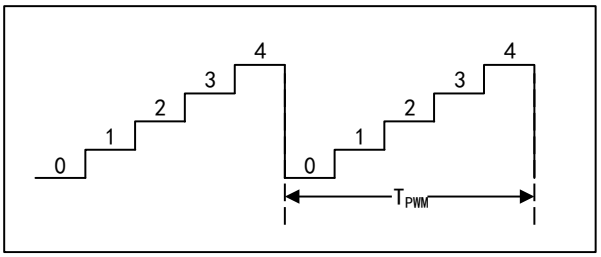
其 APWM 模式操作 PWM 波形图如下所示。

图 155 PWM 生成波形图



根据时基周期示意图，其计算时基频率以及周期公式为：

图 156 时基周期示意图



$$T_{PWM} = (CAP1 = 1) * T_{TSCNT}$$

$$F_{PWM} = \frac{1}{T_{PWM}}$$

33.7.3 计数器比较和 PRD 对 APWM 模式下 CAP 输出的影响

根据计数器比较和 PRD 结构框图，在 APWM 模式下 CAP 的输出波形如下图所示。

图 157 计数器比较和 PRD 结构框图

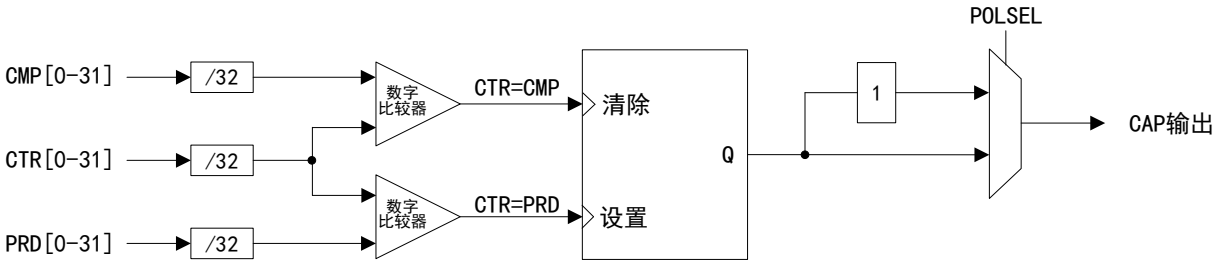
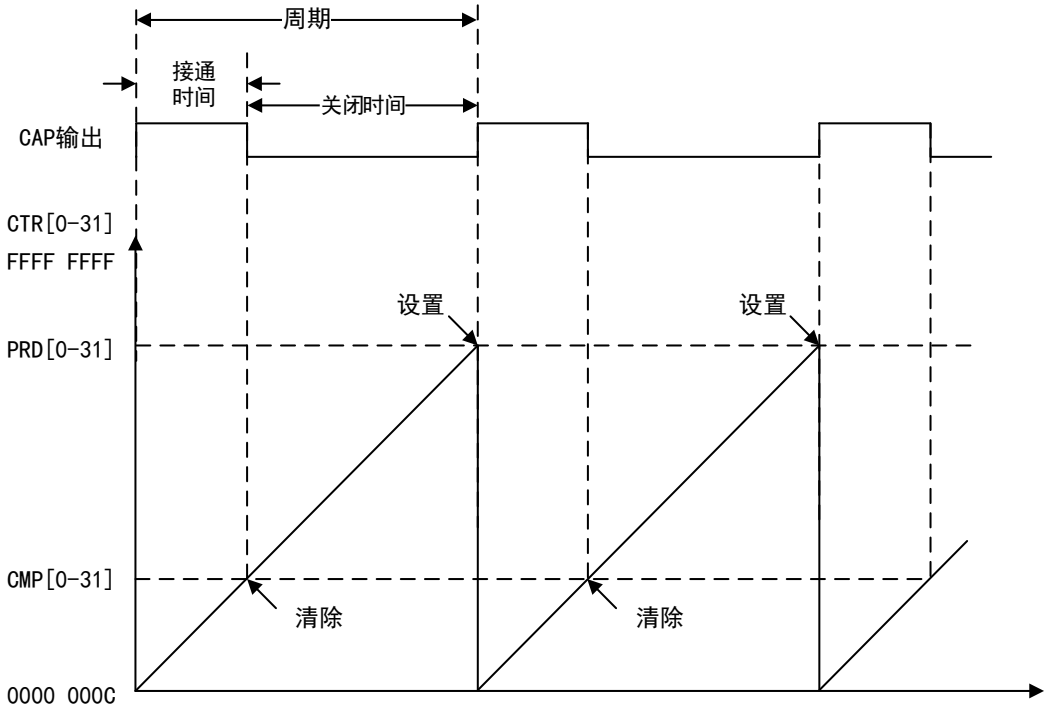


图 158 APWM 模式下 CAP 输出波形图



33.8 CAP 模块应用

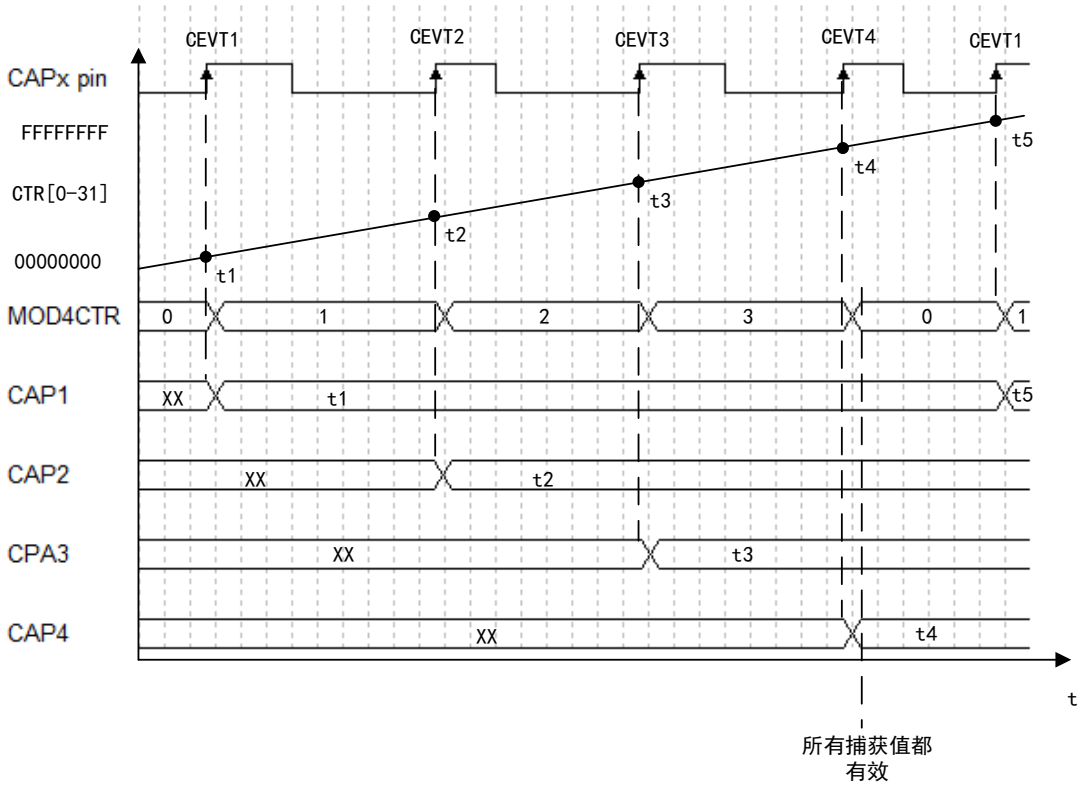
33.8.1 绝对时间戳操作

模 4 计数器循环的连续捕获操作示例如下。

当 TSCTR 处于向上计数（不复位的情况下），捕获事件仅限定为上升沿时，提供了频率和周期信息。

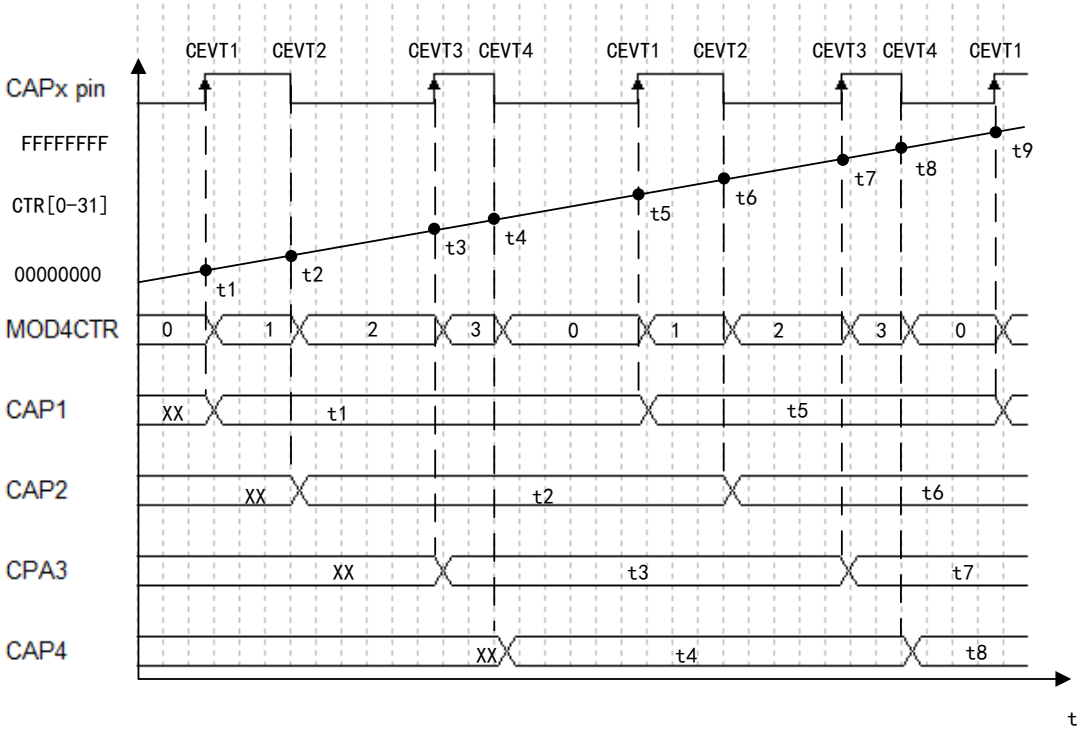
在一个捕获事件中，先捕获时间戳计数器内容，再使模 4 计数器递增到下一状态。如果 TSCTR 达到最大值 FFFF FFFF，则会循环到 0000 0000，此时就会设置 CTROVF 标志位，此时会产生一个中断（TSCTR 被使能时）。捕获的时间戳在第四个捕获事件后有效（如下图所示），因此捕获事件 4 可以用于触发中断并使 CPU 可以从捕获寄存器读取数据。

图 159 上升沿检测和绝对时间戳的捕获序列



当 TSCTR 处于向上计数（不复位的情况下），捕获事件限定为上升沿或下降沿时，提供了占空比和周期信息，即周期 1=t3-t1，占空比 1（接通时间%）=（t2-t1）/周期 1*100%，占空比 1（关闭时间%）=（t3-t2）/周期 1*100%等。

图 160 具有上升沿和下降沿检测的绝对时间戳捕获序列



33.8.2 时间差操作

CAP 模块可以用于在脉冲串波形中收集 Delta 定时数据。

当使用连续捕获模式时，TSCTR 处于向上计数（不复位的情况下）且模 4 计数器循环。捕获事件仅限定为上升沿。在 Delta-time 模式下，TSCTR 会在每一个有效事件复位为 0。

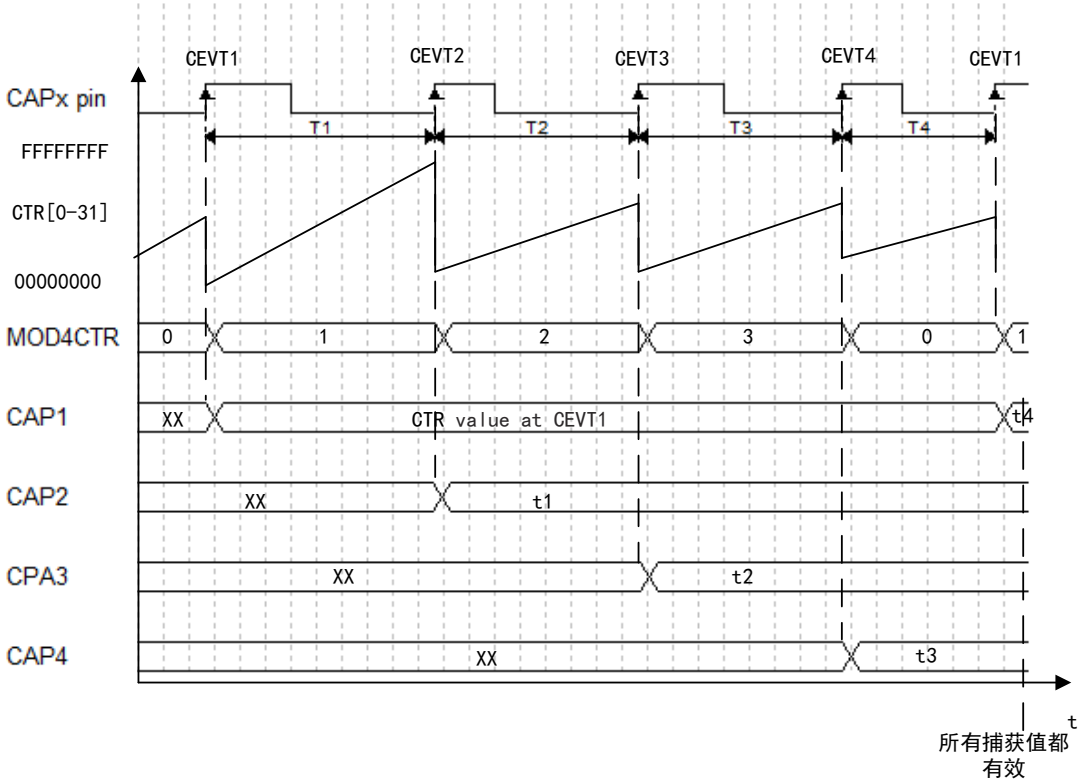
一个捕获事件顺序为：

- 对时间戳计数器内容进行捕获
- 使 TSCTR 复位
- 使模 4 计数器递增到下一状态。

在发生下一个事件之前，TSCTR 如果达到最大值 FFFF FFFF，那么将会循环到 0000 0000 并继续计数，此时就会设置 CTROVF 标志位，并在使能 TSCTR 时产生一个中断。

根据 T1=周期 1，T2=周期 2 等可知，Delta-time 模式方便 CAPx 内容无需 CPU 计算就可以直提供定时数据，如下图所示，读取定时数据的触发点是捕获事件 1，此时周期 1 至 4 都是有效的。

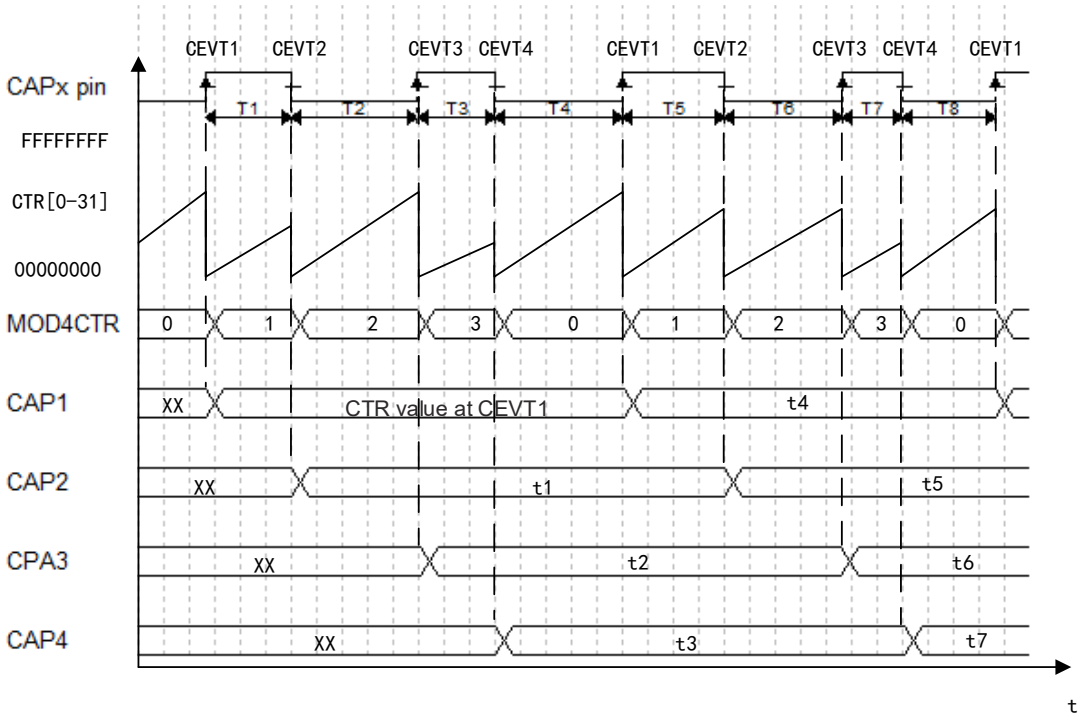
图 161 上升沿检测和 Delta 模式时间戳的捕获序列



捕获事件限定为上升沿或下降沿时，提供了占空比和周期信息，即 T1+T2=周期 1，占空比 1（接通时间%）=T1/周期 1*100%，占空比 1（关闭时间%）=T2/周期 1*100%以此类推。

在初始化期间，必须在两个周期内写入活动寄存器并进行比较，此时将会自动将初始值复制到影子值中。为了运行时的后续比较更新，必须使用影子寄存器。

图 162 具有上升沿和下降沿检测的 Delta 模式时间戳捕获序列



33.9 寄存器组地址

表格 166 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
CAP1Regs	CAP_REGS	0x4000 2000	0x4000 23FF
CAP2Regs	CAP_REGS	0x4000 2400	0x4000 27FF
CAP3Regs	CAP_REGS	0x4000 2800	0x4000 2BFF
CAP4Regs	CAP_REGS	0x4000 2C00	0x4000 2FFF
CAP5Regs	CAP_REGS	0x4000 3000	0x4000 33FF
CAP6Regs	CAP_REGS	0x4000 3400	0x4000 343F
CAP7Regs	CAP_REGS	0x4000 3800	0x4000 383F

33.10 寄存器地址映射

表格 167 CAP_REGS 寄存器组地址映射

寄存器名	描述	偏移地址	WRPRT
TSCTR	时间戳计数器寄存器	0x00	-
CTRPHS	计数器相位偏移值寄存器	0x04	-
CAP1	捕获寄存器 1	0x08	-
CAP2	捕获寄存器 2	0x0C	-

寄存器名	描述	偏移地址	WRPRT
CAP3	捕获寄存器 3	0x10	-
CAP4	捕获寄存器 4	0x14	-
ECCTL0	捕获控制寄存器 0	0x24	√
ECCTL1	捕获控制寄存器 1	0x28	√
ECCTL2	捕获控制寄存器 2	0x2A	√
ECEINT	使能捕获中断寄存器	0x2C	√
ECFLG	捕获中断标志寄存器	0x2E	-
ECCLR	清除捕获中断寄存器	0x30	-
ECFRC	强制捕获中断寄存器	0x32	√

33.11 寄存器功能描述

33.11.1 时间戳计数器寄存器（TSCTR）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	TSCTR	R/W	时间戳计数器（Time Stamp Counter） （1）不可写入该寄存器，可读取 HR 计数器值 （2）可以通过 CTRFILTRESET 位进行复位 （3）若未同步到 APBCLK 域，则会导致读取不准确	0h

33.11.2 计数器相位偏移值寄存器（CTRPHS）

偏移地址：0x04

复位类型：SYSRSn

计数器相位偏移值寄存器可以针对相位超前/滞后进行编程。该寄存器在同步输入事件或软件强制通过控制位加载到 TSCTR，以实现与 PWM 时基以及其他 CAP 的相位控制同步。CTRPHS 不适用于 HR 模式

位/域	名称	R/W	描述	复位值
31:0	CTRPHS	R/W	计数器相位偏移值（Counter Phase Offset Value）	0h

33.11.3 捕获寄存器 1（CAP1）

偏移地址：0x08

复位类型：SYSRSn

在 APWM 模式中，该寄存器成为活动周期寄存器。

位/域	名称	R/W	描述	复位值
31:0	CAP1	R/W	捕获 1（Capture 1） 可以通过以下方式写入（加载）该位域： （1）软件：可用于初始化或测试目的 （2）在 APWM 模式下使用的 CAP3 寄存器（ARPD 影子寄存器） （3）捕获事件期间的 TSCTR	0h

33.11.4 捕获寄存器 2（CAP2）

偏移地址：0x0C

复位值：0x0000 0000

复位类型：SYSRSn

在 APWM 模式中，该寄存器成为活动比较寄存器。

位/域	名称	R/W	描述	复位值
31:0	CAP2	R/W	捕获 2（Capture 2） 可以通过以下方式写入（加载）该位域： （1）软件：可用于测试目的 （2）在 APWM 模式下使用的 CAP4 寄存器（ACMP 影子寄存器） （3）在捕获事件期间的 TSCTR	0h

33.11.5 捕获寄存器 3（CAP3）

偏移地址：0x10

复位类型：SYSRSn

在 CMP 模式时，该寄存器为时间戳捕获寄存器。在 APWM 模式时，该寄存器为 APRD（CAP1 周期影子）寄存器，可用于更新 PWM 周期值。

位/域	名称	R/W	描述	复位值
31:0	CAP3	R/W	捕获 3（Capture 3） 在 APWM 模式下，CAP3 可以对 CAP1 进行影子处理。	0h

33.11.6 捕获寄存器 4（CAP4）

偏移地址：0x14

复位类型：SYSRSn

在 CMP 模式时，该寄存器为时间戳捕获寄存器。在 APWM 模式时，该寄存器为 ACMP（CAP2 比较影子）寄存器，可用于更新 PWM 比较值。

位/域	名称	R/W	描述	复位值
31:0	CAP4	R/W	捕获 4（Capture 4） 在 APWM 模式下，CAP4 可以对 CAP2 进行影子处理。	0h

33.11.7 捕获控制寄存器 0（ECCTL0）

偏移地址：0x24

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
6:0	INPUTSEL	R/W	选择捕获输入源（Capture input source select） 0000000：CAP 输入[0] 0000001：CAP 输入[1] 0000010：CAP 输入[2] … 11111111：CAP 输入[127]	7Fh
31:7	保留			0h

33.11.8 捕获控制寄存器 1（ECCTL1）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2（x-1）	CAPxPOL	R/W	选择捕获事件 x 的极性（Capture Event x Polarity select）（x=1...4） 0：上升沿触发的捕获事件 x 1：下降沿触发的捕获事件 x	0h
2x-1	CTRRSTx	R/W	捕获事件 x 的计数器复位（Capture Event x Counter Reset）（x=1...4） 0：捕获事件 x 的计数器不复位（绝对时间戳） 1：捕获事件 x 时间戳后复位计数器（用于差模操作）	0h
8	CAPLDEN	R/W	在捕获事件上使能加载 CAP1/2/3/4 寄存器（Enable load CAP1/2/3/4 register on capture event） 该位不禁止生成 CEVTx 事件。 0：禁止 1：使能	0h
13:9	PRESCALE	R/W	选择事件过滤器预分频（Event Filter prescale select） 00000：不分频 00001：2 分频 00010：4 分频 00011：6 分频 00100：8 分频 00101：10 分频 11110：60 分频 11111：62 分频	0h
15:14	FREE_SOFT	R/W	选择 TSCTR 计数器运行方式（TSCTR counter run mode Select） 00：仿真停止时，TSCTR 计数器立即停止 01：TSCTR 计数器一直运行到 TSCTR 为 0 其他：TSCTR 计数器自由运行，不受仿真挂起影响	0h

33.11.9 捕获控制寄存器 2（ECCTL2）

偏移地址：0x2A

复位类型：除 CTRFILTRESET，DMAEVTSEL 和 MODCNTRSTS 位域的复位类型为 CPU1. SYSRSn，其他位域复位类型为 SYSRSn

位/域	名称	R/W	描述	复位值
0	CONT_ONESHT	R/W	选择操作模式（Operate mode Select） 该位仅适用于 CAP 模式。 0：连续模式 1：单次模式	0h

位/域	名称	R/W	描述	复位值
2:1	STOP_WRAP	R/W	选择连续/单次模式下捕获事件后的运行方式（Select how to run after capture events in capture / one-shot mode） 捕获寄存器的数量为 1~4 之间，其中循环缓冲区循环并重新开始为连续模式的循环值。 在 CAP1/2/3/4 寄存器捕获序列停止（寄存器被冻结）前，允许发生 1~4 的捕获数量为单次模式的停止值。 当 STOP_WRAP 等于模 4 计数器时，将会产生以下动作： （1）禁止 CAP1/2/3/4 寄存器加载 （2）停止（冻结）模 4 计数器 阻止在单次模式中的中断事件，直到重新加载。 00：在连续模式下捕获事件 1 后循环，在单次模式下捕获事件 1 后停止 01：在连续模式下捕获事件 2 后循环，在单次模式下捕获事件 2 后停止 10：在连续模式下捕获事件 3 后循环，在单次模式下捕获事件 3 后停止 11：在连续模式下捕获事件 4 后循环，在单次模式下捕获事件 4 后停止	3h
3	REARM	R/S	重新控制（Re-Arming Control） 重新控制在连续或单次模式下有效。 0：无效，读取返回 0 1：根据以下方式使能单次序列： （1）使能 CAP1/2/3/4 寄存器加载 （2）将模 4 计数器复位为 0 （3）使模 4 计数器解冻	0h
4	TSCTRSTOP	R/W	时间戳计数器运行模式（TSCTR run mode） 0：停止 1：自由运行	0h
5	SYNCI_EN	R/W	使能同步输入模式（sync-in mode enable） 0：禁止 1：根据软件强制事件或同步信号，从 CTRPHS 寄存器加载使能 TSCTR	0h
7:6	SYNCO_SEL	R/W	选择同步输出信号（sync-out signal Select） 00：选择同步输出信号为同步输入事件 01：选择同步输出信号为 CTR=PRD 事件 其他：禁止同步输出信号	0h
8	SWSYNC	R/S	软件强制 TSCTR 同步（Software-forced TSCTR Synchronizer） 该位为用户提供软件生成同步脉冲的方法，同步脉冲在 APWM 模式中可以通过 CTR=PRD 事件生成。 0：无效，读取返回 0 1：强制当前 CAP 模块以及符合 SYNCO_SEL=00 向下流的任何 CAP 模块的都会执行 TSCTR 影子加载操作。在写入 1 之后，该位返回到 0	0h

位/域	名称	R/W	描述	复位值
9	CAP_APWM	R/W	选择 CAP/APWM 操作模式（CAP/APWM operating mode select） 0：CAP 模式。该模式将强制进行以下配置： （1）CAPx/APWMx 引脚作为捕获输入 （2）用户可以使能捕获寄存器加载 （3）抑制 CAP1 和 CAP2 寄存器的影子加载 （4）通过 CTR=PRD 事件抑制时间戳计数器复位 1：APWM 模式。该模式将强制进行以下配置： （1）CAPx/APWMx 引脚作为 APWM 输出 （2）禁止时间戳加载到捕获寄存器 （3）CAP1 和 CAP2 寄存器可以进行影子加载 （4）通过 CTR=PRD 事件进行时间戳计数器复位（周期边界）	0h
10	APWMPOL	R/W	选择 APWM 输出极性（APWM output polarity select） 该位仅适用于 APWM 模式。 0：高电平，比较值定义高电平持续时间 1：低电平，比较值定义低电平持续时间	0h
11	CTRFILTRESET	RC_W1	复位（Reset） 该位可以防止 CAP 配置时捕获到虚假输入，从已知状态使能捕获模块。 0：无影响 1：复位模计数器，TSCTR，事件过滤器，CEINTFLG 和 CTROVF，HR 错误标志	0h
13:12	DMAEVTSEL	R/W	选择 DMA 中断源（DMA interrupt source select） 00：捕获事件 1 01：捕获事件 2 10：捕获事件 3 11：捕获事件 4	0h
15:14	MODCNTRSTS	R	读取模数计数器的状态（reads modulo counter status） 该位域可以指示模计数器的当前状态。 00：CAP1 在下一个捕获事件中加载 01：CAP2 在下一个捕获事件中加载 10：CAP3 在下一个捕获事件中加载 11：CAP4 在下一个捕获事件中加载	0h

33.11.10 使能捕获中断寄存器（ECEINT）

偏移地址：0x2C

复位类型：SYSRSn

该寄存器用于使能/禁止单个中断事件源。

位/域	名称	R/W	描述	复位值
0	保留			0h
x	CEVTx	R/W	使能捕获事件 x 中断（Capture Event x Interrupt Enable）（x=1...4） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
5	CTROVF	R/W	使能计数器溢出中断（Counter Overflow Interrupt Enable） 0：禁止 1：使能	0h
6	CTR_EQ_PRD	R/W	使能计数器相等周期中断（Counter Equal Period Interrupt Enable） 0：禁止 1：使能	0h
7	CTR_EQ_CMP	R/W	使能计数器相等比较中断（Counter Equal Compare Interrupt Enable） 0：禁止 1：使能	0h
15:8	保留			0h

33.11.11 捕获中断标志寄存器（ECFLG）

偏移地址：0x2E

复位类型：SYSRSn

该寄存器用于指示是否已产生任何中断事件，包含全局中断标志位。

位/域	名称	R/W	描述	复位值
0	INT	R	全局中断标志（Global Interrupt Flag） 0：未产生中断 1：产生中断	0h
x	CEVTx	R	捕获事件 x 中断标志（Capture Event x Interrupt Flag）（x=1...4） 仅在捕获模式下，该标志位处于活动状态。 0：未产生中断 1：CAPx 引脚产生捕获事件 x 中断，x=1...4	0h
5	CTROVF	R	计数器溢出中断标志（Counter Overflow Interrupt Flag） 在 APWM 模式和捕获模式下，该标志位处于活动状态。 0：未产生中断 1：TSCTR 从 FFFFFFFF 转变为 00000000	0h
6	CTR_PRD	R	计数器相等周期中断标志（Counter Equal Period Interrupt Flag） 仅在 APWM 模式下，该标志位处于活动状态。 0：未产生中断 1：TSCTR 达到 APRD（周期寄存器值）并进行复位	0h
7	CTR_CMP	R	计数器相等比较中断标志（Counter Equal Compare Interrupt Flag） 仅在 APWM 模式下，该标志位处于活动状态。 0：未产生中断 1：TSCTR 达到 ACMP（比较寄存器值）	0h
15:8	保留			0h

33.11.12 清除捕获中断寄存器（ECCLR）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	RC_W1	清除全局中断标志（Global Interrupt Flag Clear） 0：无效，读取返回 0 1：清除 INT，任何事件标志置位时，将使能进一步中断产生功能。	0h
x	CEVTx	RC_W1	清除捕获事件 x 中断标志（Capture Event x Interrupt Flag Clear）（x=1...4） 0：无效，读取返回 0 1：清除 CEINTFLG 标志	0h
5	CTROVF	RC_W1	清除计数器溢出中断标志（Counter Overflow Interrupt Flag Clear） 0：无效，读取返回 0 1：清除 CTROVF 标志	0h
6	CTR_PRD	RC_W1	清除计数器相等周期中断标志（Counter Equal Period Interrupt Flag Clear） 0：无效，读取返回 0 1：清除 CTR_PRD 标志	0h
7	CTR_CMP	RC_W1	清除计数器相等比较中断标志（Counter Equal Compare Interrupt Flag Clear） 0：无效，读取返回 0 1：清除 CTR_CMP 标志	0h
31:8	保留			0h

33.11.13 强制捕获中断寄存器（ECFRC）

偏移地址：0x32

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
x	CEVTx	R-0/W1S	强制捕获事件 x 中断（Force Capture Event x Interrupt）（x=1...4） 0：无效，读取返回 0 1：设置 CEINTFLG	0h
5	CTROVF	R	强制计数器溢出中断（Force Counter Overflow Interrupt） 0：无效，读取返回 0 1：设置 CTROVF	0h
6	CTR_PRD	R	强制计数器相等周期中断（Force Counter Equal Period Interrupt） 仅在 APWM 模式下，该标志位处于活动状态。 0：无效，读取返回 0 1：设置 CTR_PRD	0h
7	CTR_CMP	R	强制计数器相等比较中断（Force Counter Equal Compare Interrupt） 仅在 APWM 模式下，该标志位处于活动状态。 0：无效，读取返回 0 1：设置 CTR_CMP	0h
31:8	保留			0h

34 高分辨率捕获器（HRCAP）

34.1 术语全称、缩写描述

表格 168 术语全称、缩写描述

中文全称	英文全称	英文缩写
高分辨率定位器	High Resolution Positioner	HRP

34.2 简介

根据 CAP 结构框图可知，HRCAP 子模块是类型 1 CAP 的一部分。高分辨率捕获（HRCAP）模块测量外部脉冲宽度的精度高于 CAP 模块。HRCAP 支持以下功能：

- 测量和扫描距离或者声纳
- 测量隔离边界上的电压
- 测量流量
- 测量瞬时速度和瞬时频率
- 测量脉冲序列周期的 HR 周期和占空比
- 电容式触摸应用

34.3 主要特征

（1） 绝对模式脉冲宽度捕获

（2） 捕获非高分辨率或高分辨率模式下的脉宽

（3） 捕获模式：
- 连续模式
- 单次模式

（4） 硬件校准逻辑（用于精确高分辨率捕获）

（5） 连续模式下，脉冲宽度捕获结果支持四级缓存

（6） 在上升沿或下降沿中产生中断

（7） 校准（无需脱机执行）

（8） 类型 0 HRCAP 的改进：
- HRCAP 用法与 CAP 一致
  - 在 CAP6/7 中增加了 HRCAP 增强功能，可以允许信号异步捕获到 APBCLK。如果使用 HRCAP 增强功能，可以访问所有 CAP 硬件，但是因为与 APBCLK 同步，则无法使用输入限定或事件筛选器。每个 HRCAP 子模块还包括一个捕获通道（除了硬件校准块）。
- 使用 HRCAP 增强功能可以访问所有 CAP 硬块
- 整数和小数部分打包为 32 位，减少计算小数的软件开销

- 简化校准方案
  - HRCAP 始终有效
  - 始终在后台运行校准且无需脱机执行校准
  - 减少了校准的软件开销

（9） 每个支持 HRCAP 通道具有以下功能：

- 专用校准中断
- HR 校准逻辑
- 相应 CAP 的所有硬件

34.4 结构框图

HRCAP 功能结构框图详情请参见 CAP 章节的 CAP 结构框图。

34.5 功能描述

在使用 HRCAP 增强功能前，需要设置 CAP 模块。就最小脉宽而言，所有 HRCAP 测量都是相对时间测量。只有在需要时间转换测量的情况下，才需要校准硬件和软件。如果已经提供了校准硬件以及软件功能，则以将相对时间测量转换为以秒为单位的时间转换测量。

34.5.1 时钟源

与以前的 0 型 HRCAP 模块不同，具有 HRCAP 功能的类型 1 CAP 只需一个 PLL，但是该模块仍然需要 APBCLK 和 HRCLK（第二个异步时钟源）。因为 HRCLK 对电压和温度的变化都很敏感，所以在使用时间转换测量时，需要进行周期性连续校准。

34.5.2 初始化

使用 HRCAP 进行相对时间测量的初始化步骤：

（1） 使用 HRCAP_enableHighResolutionClock() 函数内对应控制位使能 HRCLK

（2） 延时 1μs

（3） 使用 HRCAP_enableHighResolution() 函数内对应控制位使能 HR 模式

（4） 延时 1μs

（5） 根据需要配置 CAP 模块，包括中断

使用 HRCAP 进行时间转换测量的初始化步骤（HRCAP 初始化步骤）：

（1） 使用 HRCAP_enableHighResolutionClock() 函数内对应控制位使能 HRCLK

（2） 延时 1μs

（3） 使用 HRCAP_enableHighResolution() 函数内对应控制位使能 HR 模式

（4） 延时 1μs

（5） 根据需要配置 CAP 模块，包括中断

（6） 使用 HRCAP_setCalibrationPeriod() 函数内对应控制位设置校准周期

（7） 使用 HRCAP_setCalibrationMode() 函数内对应控制位使能连续校准

（8） 使用 HRCAP_enableCalibrationInterrupt() 函数内对应控制位使能中断

（9） 使用 HRCAP_startCalibration() 函数内对应控制位开始校准

34.5.3 校准

因为不需要对相对时间测量进行校准，所以校准功能仅适用于时间转换测量。

HRCAP 子模块有一个用于在计算 APBCLK 和 HRCLK 之间的比例系数时减少软件开销的校准块。HRCAP 子模块捕获的所有值都以 HRCLK 周期为单位。由于 HRCLK 速度随着电压和温度的而变化而变化，所以需要一个比例系数来将捕获值转换为 APBCLK 域，且需要定期重新计算比例系数。

注意：为了使 HRCAP 校准工作，系统需要在 APBCLK 频率大于 100 MHz 条件下运行。

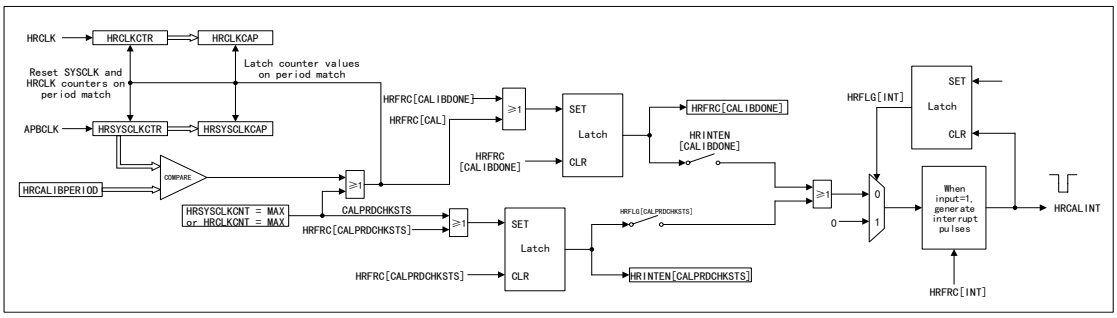
校准块主要资源

表格 169 校准块主要资源

主要资源	描述
PRD 寄存器	校准周期。当 HRSYSCLKCTR = PRD 时，停止校准。
HRSYSCLKCTR 寄存器	连接 APBCLK 的 32 位计数器。设置 CALIBSTART 位后，该计数器开始计数。
HRSYSCLKCAP 寄存器	在校准周期匹配时，HRSYSCLKCTR 被捕获到该寄存器中。
HRCLKCTR 寄存器	连接 HRCLK 的 32 位计数器。设置 CALIBSTART 位后，该计数器开始计数。
HRCLKCAP 寄存器	在校准周期匹配时，HRCLKCTR 被捕获到该寄存器中。
HRCALINT	在 HRSYSCLKCTR 和 HRCLKCTR 其中一个出现溢出情况时或校准周期匹配时发生的中断。

结构框图

图 163 HRCAP 校准结构框图



校准逻辑

校准逻辑由两个自由运行计数器 HRCLK 和 APBCLK 组成。当 HRSYSCLKCTR 等于 PRD 时，校准块将捕获两个计数器值并对其进行复位，然后触发中断，指示准备计算新的比例系数。当其中一个计数器出现溢出情况时，将设置 CALPRDCHKSTS 标志。比例系数计算公式：

$$\text{比例系数} = \frac{\text{HRSYSCLKCAP}}{\text{HRCLKCAP}}$$

根据 DriverLib 函数 HRCAP_getScaleFactor 可以确定比例系数，该函数可以在校准中断服务程序内部调用。

注意：

（1）即便已经进行了校准，1.1V 电源 VDD 上的噪声仍可能影响 HRCAP 子模块的标准偏差。因此，应注意确保 1.1V 电源清洁，并在使用 HRCAP 子模块时，尽量减少使能和禁用时钟树等内部噪声事件的发生。

（2）当 HRCLK 的频率超过 APBCLK 时，校准过程可能会过早停止。为了减轻校准不准确的风险，APBCLK 的频率必须至少设定为 100 MHz。

比例系数

DriverLib 函数 HRCAP_convertEventTimeStampNanoseconds() 可以用于将比例系数应用于捕获值。在不使用 DriverLib 函数的情况下，根据以下公式将原始计数转换为秒。

$$\text{Measurement(ns)} = \frac{\textit{捕获值} * \textit{比例系数}}{128} * \textit{系统时钟周期}$$

表格 170 公式参数说明

参数	描述
Measurement(ns)	信号转换为 ns
捕获值	从捕获寄存器 1/2/3/4 读取的值
比例系数 (1)	根据比例系数计算公式的计算值
128	由 HRCAP 子模块硬件决定常数

注意：（1）比例系数不会自动应用于捕获值，需要用户将比例系数应用于所有捕获的值。

34.5.4 中断

HRCAP 增强利用现有的 CAP 中断（硬件校准专用的 HRCALINT 除外），详情请参见 CAP 模块的中断章节。HRCALINT 可以在 HRSYSCLKCTR 或 HRCLKCTR 出现溢出时以及 HRSYSCLKCTR = HRCALIBPERIOD(CALPRD) 的任一条件下触发。

因为 G32R501 所使用 CPU 对于中断处理的特性，在 CAP 和 HRCAP 的中断程序里，不建议使用系统级 Clear Pending 语句，如果使用该语句，在高频捕获时可能造成程序运行中无法再次进入中断的问题，不使用则不会有该问题。

34.6 寄存器组地址

表格 171 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Hrcap6Regs	HRCAP_REGS	0x4000 3440	0x4000 37FF
Hrcap7Regs	HRCAP_REGS	0x4000 3840	0x4000 3BFF

34.7 寄存器地址映射

表格 172 HRCAP_REGS 寄存器组地址映射

寄存器名	描述	偏移地址	WRPRT
HRCTL	HR 控制寄存器	0x00	√
HRINTEN	使能 HR 校准中断寄存器	0x08	√
HRFLG	HR 校准中断标志寄存器	0x0C	-
HRCLR	清除 HR 校准中断寄存器	0x10	-
HRFRC	强制 HR 校准中断寄存器	0x14	√
HRCALPRD	校准周期寄存器	0x18	√
HRSYSCLKCTR	校准 APBCLK 计数器寄存器	0x1C	-
HRSYSCLKCAP	校准 APBCLK 捕获寄存器	0x20	-
HRCLKCTR	校准 HRCLK 计数器寄存器	0x24	-
HRCLKCAP	校准 HRCLK 捕获寄存器	0x28	-

34.8 寄存器功能描述

34.8.1 HR 控制寄存器（HRCTL）

偏移地址：0x00

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
0	HRE	R/W	使能高分辨率（High Resolution Enable） 使能该模块之前，必须先使 HRCLKE=1，并允许有一定的启动稳定期。 0：禁止 1：使能。使能高分辨率将连接边沿事件模式和 CAP1/2/3/4 寄存器，可以通过该功能访问 CAP。	0h
1	HRCLKE	R/W	使能高分辨率时钟（High Resolution Clock Enable） 在使能高分辨率功能之前先使能高分辨率时钟。 0：禁止（复位时默认） 1：使能	0h

位/域	名称	R/W	描述	复位值
2	PRDSEL	R/W	选择校准周期匹配（Calibration Period Match Select） 0：通过 APBCLK 计数器进行周期匹配（复位时默认） 1：保留	0h
3	CALIBSTART	R-0/W1S	开始校准（Calibration start） 0：无效 1：开始校准	0h
4	CALIBSTS	R	标准标志（Calibration Flag） 0：未进行校准周期 1：正在进行校准周期	0h
5	CALIBCONT	R/W	使能连续模式（Continuous mode Enable） 0：禁止 1：使能。在当前校准周期结束时，校准自动重新开始。	0h
31:6	保留			0h

34.8.2 使能 HR 校准中断寄存器（HRINTEN）

偏移地址：0x08

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	CALIBDONE	R/W	使能校准完成中断（Calibration done Interrupt Enable） 0：禁止 1：使能	0h
2	CALPRDCHKSTS	R/W	使能校准周期校验中断（Calibration Period Check Interrupt Enable） 0：禁止 1：使能	0h
31:3	保留			0h

34.8.3 HR 校准中断标志寄存器（HRFLG）

偏移地址：0x0C

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
0	CALIBINT	R	全局校准中断标志（Global calibration Interrupt Flag） 0：未生成中断 1：从 CALIBDONE 或 CALPRDCHKSTS 中生成中断	0h
1	CALIBDONE	R	校准完成中断标志（Calibration done Interrupt Flag） 该位将一直保持锁存状态，直到用户通过 CALIBDONE 位清除该位。 0：未完成校准周期 1：已完成校准周期	0h

位/域	名称	R/W	描述	复位值
2	CALPRDCHK STS	R	校准周期校验中断标志（Calibration Period Check Interrupt Flag） 该位将一直保持锁存状态，直到用户通过 CALPRDCHKSTS 位清除该位。 0：由于其中一个计数器溢出，校准在 PRDCHK 前结束 1：未发生任何校准事件	0h
31:3	保留			0h

34.8.4 清除 HR 校准中断寄存器（HRCLR）

偏移地址：0x10

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
0	CALIBINT	RC_W1	清除全局校准中断标志（Global calibration Interrupt Flag Clear） 0：无效 1：当设置了任何事件标志时，清除 CALIBINT 并允许生成进一步的中断	0h
1	CALIBDONE	RC_W1	清除校准完成中断标志（Calibration done Interrupt Flag Clear） 当用户尝试清除标志位且为所选位设置标志的同一周期发生事件时，H/W 优先级高于 CPU 写入。 0：无效 1：清除 CALIBDONE	0h
2	CALPRDCHK STS	RC_W1	清除校准周期校验中断标志（Calibration Period Check Interrupt Flag Clear:） 当用户尝试清除标志位且为所选位设置标志的同一周期发生事件时，H/W 优先级高于 CPU 写入。 0：无效 1：清除 CALPRDCHKSTS	0h
31:3	保留			0h

34.8.5 强制 HR 校准中断寄存器（HRFRC）

偏移地址：0x14

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	CALIBDONE	R/S	强制校准完成中断（Force Calibration done Interrupt） 0：无效 1：设置 CALIBDONE	0h
2	CALPRDCHK STS	R/S	强制使能校准周期校验中断（Force Calibration Period Check Interrupt） 0：无效 1：设置 CALPRDCHKSTS	0h
31:3	保留			0h

34.8.6 校准周期寄存器（HRCALPRD）

偏移地址：0x18

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
31:0	PRD	R/W	寄存器到程序的校准周期（Register to program calibration period） 在 HRSYSCLKCTR 与周期值匹配时会生成一个中断，并捕获 APBCLK 和 HRCLK 计数器值。	003F FFFFh

34.8.7 校准 APBCLK 计数器寄存器（HRSYSCLKCTR）

偏移地址：0x1C

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
31:0	HRSYSCLKCTR	R	校准 APBCLK 计数器值（Calibration SYSCLK Counter value）	0h

34.8.8 校准 APBCLK 捕获寄存器（HRSYSCLKCAP）

偏移地址：0x20

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
31:0	HRSYSCLKCAP	R	捕获 HRSYSCLKCTR（HRSYSCLKCTR captures） 在校准周期结束时，HRSYSCLKCTR 被捕获到该寄存器。	0h

34.8.9 校准 HRCLK 计数器寄存器（HRCLKCTR）

偏移地址：0x24

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
31:0	HRCLKCTR	R	校准 HRCLK 计数器值（calibration HRCLK counter value） 由于 HRCLK 未同步到 APBCLK 域，所以读取可能不准确。	0h

34.8.10 校准 HRCLK 捕获寄存器（HRCLKCAP）

偏移地址：0x28

复位类型：CPU1. SYSRSn

位/域	名称	R/W	描述	复位值
31:0	HRCLKCAP	R	捕获 HRCLKCTR（HRCLKCTR captures） 在校准周期结束时，HRCLKCTR 被捕获到该寄存器。由于 HRCLK 未同步到 APBCLK 域，所以读取可能不准确。	0h

35 正交编码器脉冲（QEP）

35.1 术语全称、缩写描述

表格 173 术语全称、缩写描述

中文全称	英文全称	英文缩写
正交模式适配器	Quadrature Mode Adapter	QMA
正交解码器单元	Quadrature decoder unit	QDU
位置计数器和控制单元	Position Counter and Control Unit	PCCU
正交边沿捕获单元	Quadrature Edge-capture Unit	QCAP
正交时钟	Quadrature-clock	QCLK
单元位置事件	Unit Position Event	UPE

35.2 简介

正交编码器脉冲（QEP）用于与旋转或线性增量编码器专用接口，获取来自旋转器的方向、速度和位置信息，用于位置控制和高性能运动系统。

35.3 主要特征

（1） 主要功能单元有：

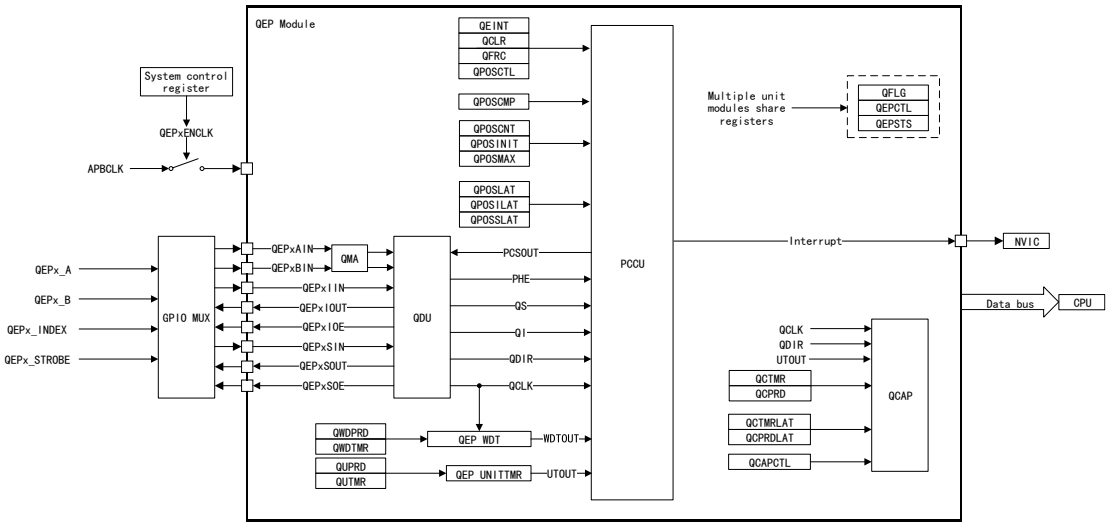
- QMA 模块
- 正交解码器单元
- 位置计数器和控制单元（用于位置测量）
- 正交边沿捕获单元（用于低速测量）
- 看门狗定时器（用于检测失速）
- 单元时基（用于测量速度/频率）
- 每个引脚的可编程输入确定

（2） 11 个中断事件：

- PCE
- PHE
- QDC
- WTO
- PCU
- PCO
- PCR
- PCM
- SEL
- IEL
- UTO

35.4 结构框图

图 164 QEP 结构框图



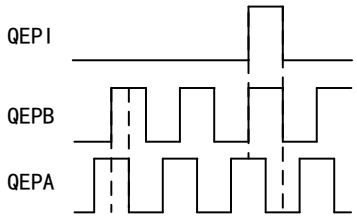
35.5 功能描述

35.5.1 增量编码器

编码器磁盘

如编码器磁盘平面图所示，增量编码器磁盘外围有一系列槽道，形成明暗交替图案。磁盘计数指每转线数出现明暗线对的数量。一个每转一次的 QEPI 索引信号通常通过添加第二个轨迹来产生，如 QEPI 波形图所示。QEPI 索引信号可以用于指示绝对位置。编码器制造商根据起始位置，零参考，索引和标记等术语来识别索引脉冲。

图 165 QEP 波形图



通过两个不同的光电元件读出磁盘上的线条可以获得方向信息，通过两个光电元件之间机械位移为 1/4 线对间距来观察磁盘图案。机械位移是通过标尺或掩模来测量，这将光电元件的视野限制在磁盘线所需部分。当磁盘旋转时，两个光电元件产生的正交 QEPA 和 QEPB 信号电相位相差 90 度。通常将 QEPA 通道在 QEPB 通道之前定义为编码器顺时针方向，反之亦然。如下图所示。

图 166 QEP 编码器输出信号顺时针运行方向

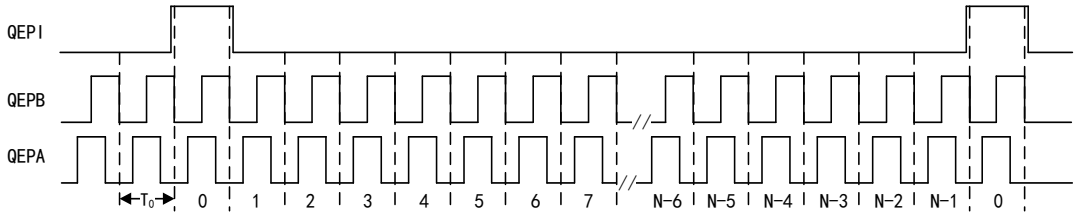
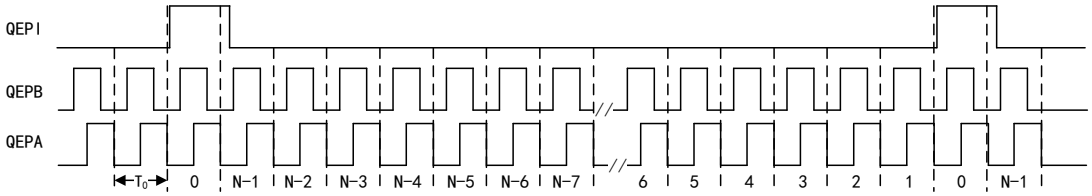


图 167 QEP 编码器输出信号逆时针运行方向



通常电动机每转一次，编码器轮也旋转一圈，或者编码器轮可以处于齿轮传动比（相对于电机）。因此，电动机的速度与 QEPA 和 QEPB 输出数字信号的频率按比例变化。CPU 可以根据测量 QEPA 或 QEPB 输出的频率来确定电机的速度。比如，一个 2000 线编码器直接耦合到以每分钟 5000 转运行的电机时，其产生频率为 166.6KHz。

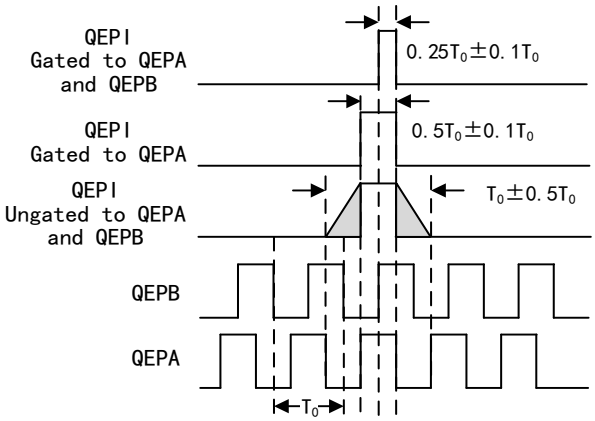
索引脉冲

正交编码器的索引脉冲在不同制造商中可以分为：

- 门控索引脉冲
- 非门控索引脉冲

非标准形式的索引脉冲是非门控索引脉冲。配置非门控时，索引边沿不一定与 QEPA 和 QEPB 信号一致。索引脉冲的宽度可以等于正交信号的整个周期、一半周期或四分之一周期，并且选通的索引脉冲与和四个正交边沿的其中一个对齐。

图 168 脉冲索引波形图



轴编码器

轴编码器的典型应用：

- 机器人
- 鼠标形式的计算机输入

可以在鼠标内部看到鼠标球旋转一对左/右轴和上/下轴的位置。因为这些轴连接到光学轴编码器，所以计算机可以有效地知道鼠标移动的速度和方向。

速度的不同近似值

在电机控制中有一种具有成本效益的策略，即估算来自数字位置传感器的速度。在低速和高速下都需要进行速度估算时，一种方法是在低速时，使用精确估算速度公式，并且在电机速度上升到某个特定阈值后，通过软件切换到传统估算速度公式。

（1） 传统估算速度公式为：

$$v(k) \approx \frac{\mathrm{x(k)-x(k-1)}}{\mathrm{T}} = \frac{\Delta X}{T}$$

表格 174 补充说明

参数	描述
v（k）	时刻 k 的速度
x（k）	时刻 k 的位置
x（k-1）	时刻 k-1 的位置
ΔX	单元时间内的增量位置移动
T	固定单元时间或速度计算速率的倒数

为了提供用于速度计算的单元时间事件，传统估算速度方法需要一个时基。通常来说，单元时间是速度计算速率的倒数。在每个单元时间事件中读取一次编码器计数。通过当前读数减去先前读数得出 x（k） - x（k-1），然后通过乘以已知单元时间事件之间的恒定时间 T 的倒数来计算速度。

传统估算速度方法具有与以下直接相关的固有精度限制：

- 单元时间段 T
- 位置传感器的分辨率

假设，考虑每转 500 线正交编码器，其速度计算速率为 400Hz。当该正交编码器用于定位时，其分辨率提高了四倍，则每转一次有 2000 次计数。所以检测到最小旋转为 0.0005 转，如果以 400Hz 采样，则速度分辨率为 12rpm。但是因为这种分辨率在低于 12rpm 的速度下会出现错误等于 0，所以这种方法只适用于中速或高速下。

（2） 精确估算速度公式为：

$$v(k) \approx \frac{\mathrm{X}}{t(k)-t(k-1)} = \frac{X}{\Delta T}$$

表格 175 补充说明

参数	描述
v（k）	时刻 k 的速度
t（k）	时刻 k 的时间
t（k-1）	时刻 k-1 的时间
X	固定单元位置
ΔT	单元位置移动所经过的增量时间

在低速时，通过上方公式可以更精确的估算速度。但是它需要一个位置传感器（比如正交编码器）来提供输出固定间隔脉冲串。由给定传感器分辨率的电机速度定义每个脉冲的宽度。精确估算速度方法通过测量连续正交脉冲边沿之间的经过时间来计算电动机速度。但是，该方法相较于传统估算速度方法有相反的限制。高传感器分辨率和相对较大的电机速度的结合会使时间间隔 ΔT 较小，而受计时器分辨率的影响更大。所以在高速下，使用精确估算速度方法误差增大。

35.5.2 引脚配置

GPIO 多路复用器寄存器必须配置为将 QEP 连接到设备引脚。首先必须配置 GPxGMUX 寄存器，同时将 GPxMUX 寄存器中相应的位域保持为 0，然后再向 GPxMUX 寄存器写入所需值，可以避免引脚出现故障。

为了使 QEP 模块正常工作，必须通过 GPxQSEL 寄存器将输入 GPIO 引脚配置为同步输入模式，并且 QEP 输入引脚不可以使用异步模式。可以在 GPxPUD 寄存器中配置内部上拉。

有关 GPIO 多路复用器内容，详情请参见数据手册。有关 GPIO 设置内容，详情请参见 GPIO 章节。

35.5.3 输入方式

QEP 输入方式有以下三种：

- 输入选通信号
- XCLK 和 XDIR 引脚
- 索引或 0 标记

QEP 模块要求 QEPA、QEPB 和 QEPI 输入在进入模块之前同步到 SYSCLK。在任何使能 QEP 的 GPIO 引脚上，应用程序代码可以使能同步 GPIO 输入功能，有关 GPIO 内容，详情请参见 GPIO 章节。

输入选通信号

在选通引脚上发生期望事件时，该通用选通信号（QEPS）可以初始化或锁存位置计数器。为了通知电机已到达特定位置，通常将该信号连接到传感器或限位开关。

XCLK 和 XDIR 引脚

XCLK 和 XDIR 引脚可以用于方向计数模式或正交时钟模式。

（1） 方向计数模式

在该模式下，直接从外部源提供时钟信号和方向，其中 QEPA 引脚提供时钟输入，QEPB 引脚提供方向输入。一些位置编码器不具有正交输出，但是具有这种类型的输出。

（2） 正交时钟模式

QEP 编码器提供两个方波信号（QEPA 和 QEPB），且 QEPA 和 QEPB 信号电相位相差 90 度。为了导出相对位置信息，该相位关系用于确定来自索引位置的 QEP 脉冲的数量和输入轴的旋转方向。正交解码器通过这两个输入信号来生成正交时钟和方向信号。在顺时针旋转或正向情况下，QEPA 信号领先于 QEPB 信号，反之亦然。

索引或 0 标记

QEP 编码器通过索引输入信号（QEPI）对绝对起始位置进行分配，从该绝对起始位置使用正交脉冲对位置信息进行递增编码。通过将该引脚连接到 QEP 编码器的索引输出，可以选择性地复位每转的位置计数器。在索引引脚上出现所需事件时，QEPI 可以用于初始化或锁存位置计数器。

极性选择

通过 QSP 位，QIP 位，QBP 位和 QAP 位来控制每个 QEP 输入进行反转。比如设置 QIP 位将反转 QEPI 输入。

35.5.4 定时器

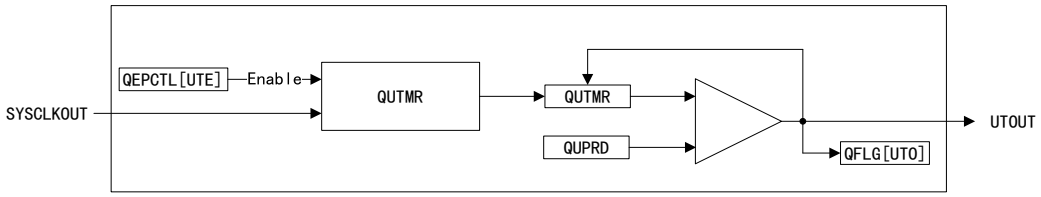
QEP 外设包含一个 32 位时基单元定时器和一个 16 位看门狗定时器。

32 位时基单元定时器

如 32 位时基单元定时器结构框图所示，32 位定时器时钟源为 SYSCLKOUT，可以生成速度计算的周期性中断。如果单元时间周期值与该定时器值匹配，则会对单元定时器进行复位并生成单元超时中断标志 UTO。每当 QUPRD 等于 QUTMR 时，就会复位 QUTMR。

QEP 为了将锁存值用于速度计算，该外设在单元超时事件上可以对位置计数器，捕获定时器值和捕获周期值进行锁存。

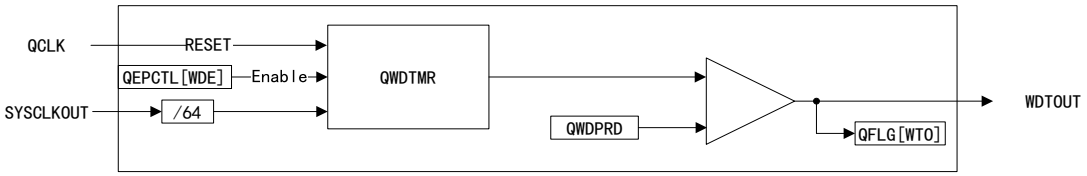
图 169 32 位时基单元定时器结构框图



16 位看门狗定时器

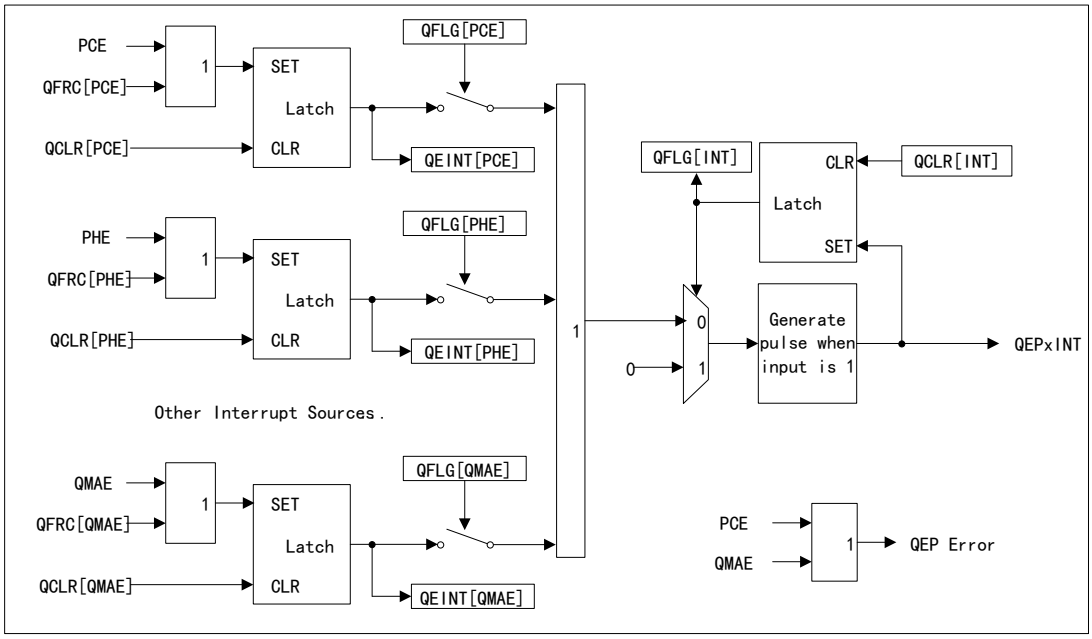
16 位看门狗定时器用来监测由正交时钟指示运动控制系统的正确操作。16 位看门狗定时器时钟源为 SYSCLKOUT/64，通过正交时钟事件（QWDTMR=QWDPRD）可以对看门狗定时器进行复位。当在周期匹配之前未检测到正交时钟事件时，看门狗定时器超时并设置看门狗超时中断标志 WTO。通过 QWDPRD 寄存器可以对其超时值进行编程。

图 170 16 位看门狗定时器结构框图



35.5.5 QEP 中断

图 171 QEP 中断结构框图



NVIC 产生中断脉冲条件

NVIC 在以下条件下会产生中断脉冲：

- 使能 QEP 事件中断（通过 QEINT 寄存器）
- 设置了 QEP 事件中断标志（通过 QFLG 寄存器）
- 对于先前生成的中断事件，全局中断标志位 INT 已被清除。中断服务程序需要在产生任何其他中断脉冲之前，通过 QCLR 寄存器清除服务事件和 INT 位。只要 QFLG 寄存器中有任一标志未被清除，NVIC 就不会因为新产生的中断事件而产生中断脉冲。通过 QFRC 寄存器可以强制产生中断事件。

35.6 正交模式适配器（QMA）功能描述

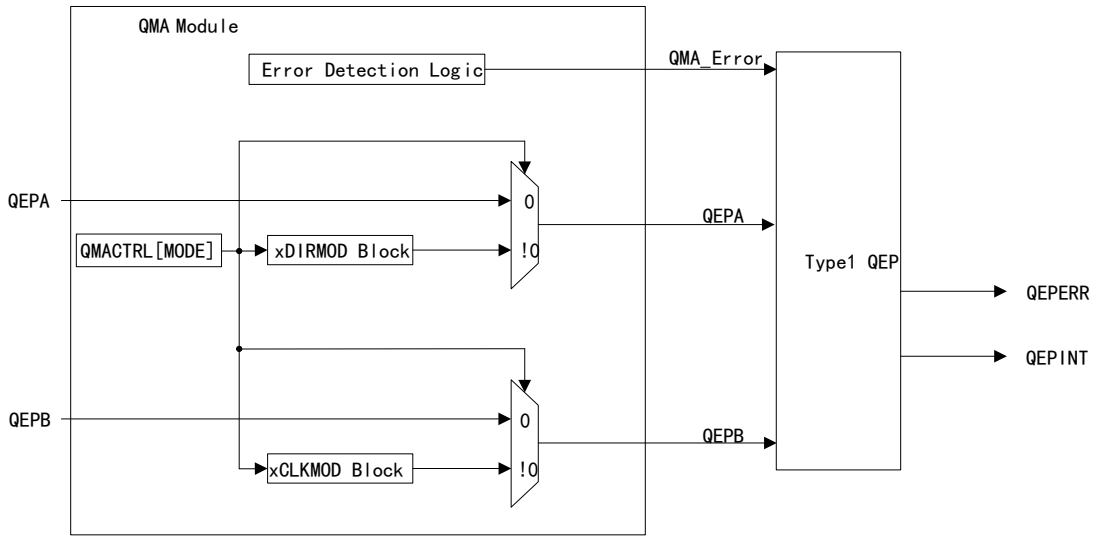
用于扩展 QEP 模块支持所描述的附加模式的功能。

当 QMA 复位时，会绕过 QMA 逻辑（默认情况下），引脚的 QEPA 和 QEPB 输入直接进入 QEP 模块。通过配置 MODE 位来使能 QMA 模块来处理 QEPA 和 QEPB 输入，并将修改后的 QEPA 和 QEPB 信号发送至 QEP 模块。为了正确操作预期模式，QMA 模块要求将 QEP 模块配置在方向计数模式下，根据需要在 QEPA 和 QEPB 输入上分别生成时钟信号和方向信号。

35.6.1 QMA 结构框图

通过 QMA 模块内的 xCLKMOD 块和 xDIRMOD 块可以查看外部 QEPA 和 QEPB 信号的转换，xCLKMOD 块在输入到 QEP 模块的 QEPA 上生成时钟信号。xDIRMOD 块在 QEPB 输入到 QEP 模块上生成方向信号。

图 172 QMA 结构框图



35.6.2 操作模式

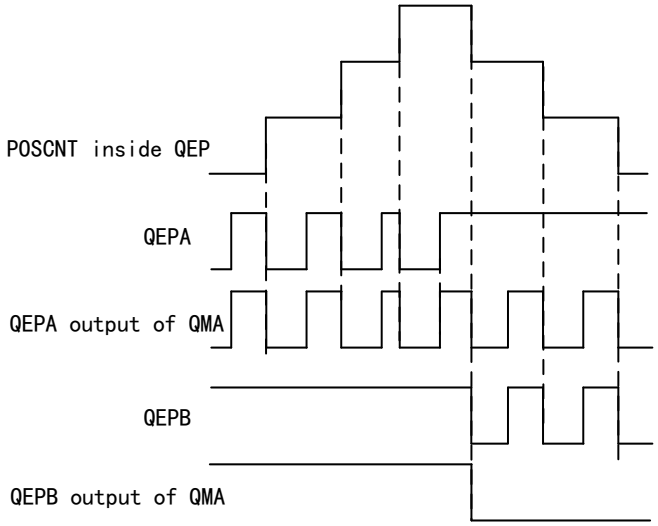
通过 QMACTRL 寄存器可以配置 QMA 模块的操作模式：

- MODE=001，选择操作模式 1
- MODE=010，选择操作模式 2

操作模式 1

在 QEPA 和 QEPB 输入信号默认状态为高时使用操作模式 1，此时，QMA 的输出如下图所示，其中 QEPA 输出是来自引脚的 QEPA 和 QEPB 输入的逻辑“与”运算，QEPB 输出是 QMA 基于 QEPA 和 QEPB 输入生成的方向信号。

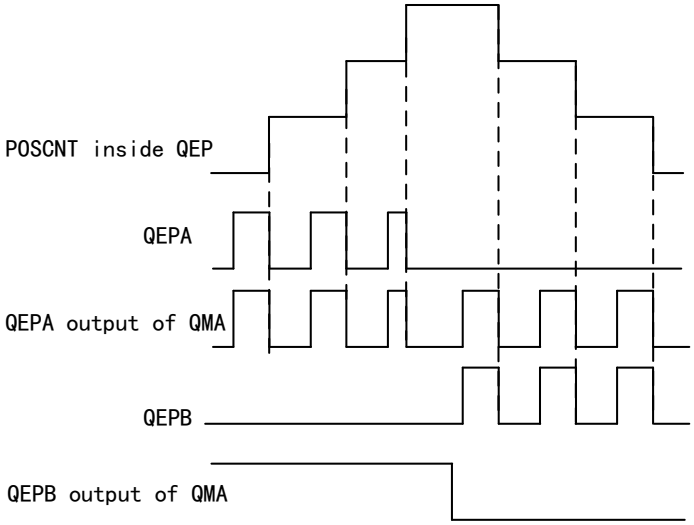
图 173 QMA 操作模式 1 输出



操作模式 2

在 QEPA 和 QEPB 输入信号默认状态为低时使用操作模式 2，此时，QMA 的输出如下图所示，其中 QEPA 输出是来自引脚的 QEPA 和 QEPB 输入的逻辑“或”运算，QEPB 输出是 QMA 基于 QEPA 和 QEPB 输入生成的方向信号。

图 174 QMA 操作模式 2 输出



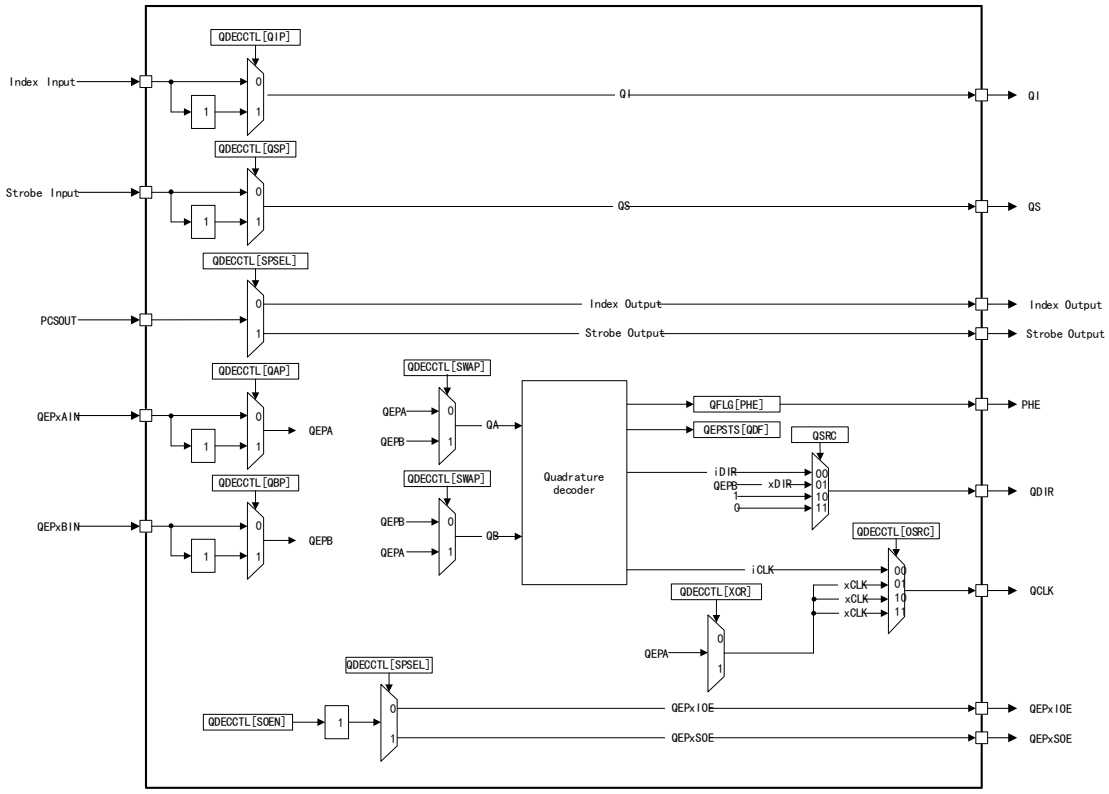
35.6.3 错误检测逻辑

QMA 模块具有错误检测逻辑，QEPA 和 QEPB 信号上的非法转换可以通过错误检测逻辑来检测并产生错误信号。该错误信号可用于生成错误输出和 QEP 中断，而 QMA 模块的错误和中断集成在 QEP 模块内，详情请参见 QEP 中断章节。

35.7 正交解码器单元（QDU）功能描述

35.7.1 QDU 结构框图

图 175 QDU 结构框图



35.7.2 输入模式

根据接口输入要求，通过 QSRC 位来选择位置计数器的方向输入和时钟：

- 向上计数模式
- 向下计数模式
- 正交计数模式
- 方向计数模式

向上计数模式

位置计数器用于测量 QEPA 输入的频率，计数器方向信号为硬连线用于向上计数。清除 XCR 位使时钟生成到 QEPA 输入两端的位置计数器，因此，将测量分辨率提高了 2 倍。

在向上计数模式下，应用程序应确保在 QEPB 输入上不生成信号边沿或不要将 QEPB 配置为 GPIO 复用选项。

向下计数模式

位置计数器用于测量 QEPA 输入的频率，计数器方向信号用于向下计数。清除 XCR 位使时钟生成到 QEPA 输入两端的位置计数器，因此，将测量分辨率提高了 2 倍。

在向下计数模式下，应用程序应确保在 QEPB 输入上不生成信号边沿或不要将 QEPB 配置为 GPIO 复用选项。

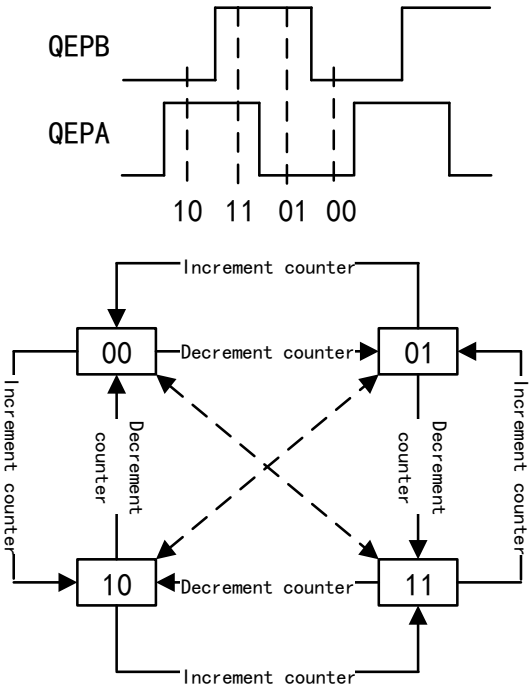
正交计数模式

在正交计数模式下，正交解码器生成到位置计数器的方向和时钟。

（1） 方向解码

由 QEP 电路的方向解码逻辑确定前导序列是 QEPA 还是 QEPB，并更新 QDF 位中相应的方向信息。正交解码逻辑状态机制图以及真值表格如下所示。

图 176 正交解码逻辑状态机制图



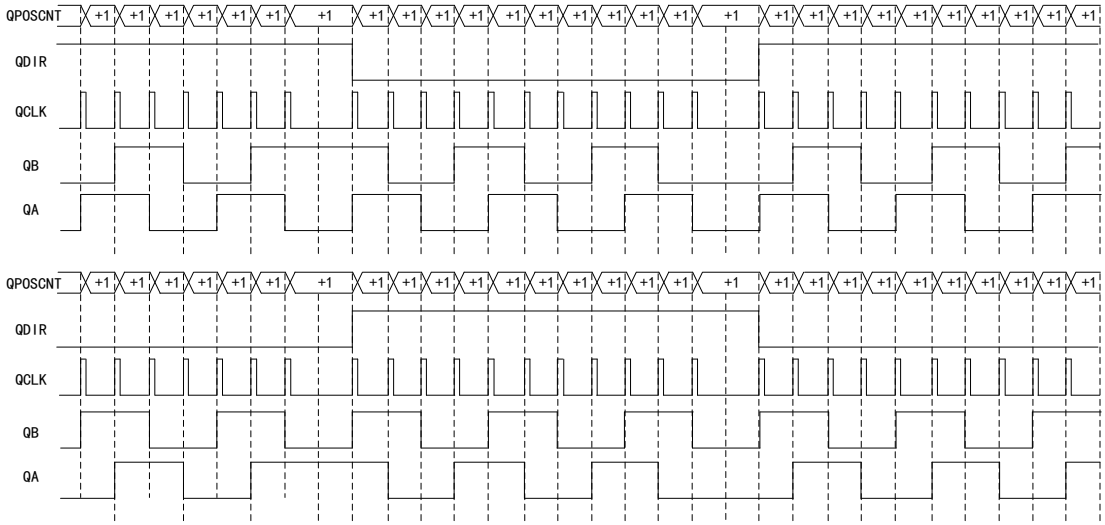
表格 176 正交解码逻辑真值

上一个边沿	当前边沿	计数方向	计数值（QPOSCNT）
QA 为上升沿	QA 为下降沿	切换	递增或递减
	QB 为上升沿	向上计数	递增
	QB 为下降沿	向下计数	递减
QA 为下降沿	QA 为上升沿	切换	递增或递减
	QB 为上升沿	向下计数	递减
	QB 为下降沿	向上计数	递增
QB 为上升沿	QA 为上升沿	向下计数	递减
	QA 为下降沿	向上计数	递增
	QB 为下降沿	切换	递增或递减
QB 为下降沿	QA 为上升沿	向上计数	递增

上一个边沿	当前边沿	计数方向	计数值（QPOSCNT）
	QA 为下降沿	向下计数	递减
	QB 为上升沿	切换	递增或递减

当检测到 QEPA 和 QEPB 信号的两个边沿时，将产生位置计数器的计数脉冲。QEP 输入信号的方向解码和正交时钟生成如下图所示，QEP 逻辑产生的时钟频率是每个输入序列的频率的四倍。

图 177 QEP 输入信号的方向解码和正交时钟



（2） 计数乘法

位置计数器通过在 QEPA 和 QEPB 输入时钟的上升沿/下降沿生成 QCLK，提供的分辨率是输入时钟的 4 倍，如 QEP 输入信号的方向解码和正交时钟图所示。

（3） 反向计数方向

在正常正交计数模式下，QEPA 输入被施加到正交解码器的 QA 输入，而 QEPB 输入被施加给正交解码器的 QB 输入。通过设置 QDECCTL[SWAP]位来使能交换正交时钟输入，从而实现反转计数方向。

（4） 正交相位错误标志

正交输入 QEPA 和 QEPB 相位相差 90 度（正常工作条件下）。正交相位错误标志设置在 QFLG 寄存器中，且 QPOSCNT 值可能不正确并且偏移 1 或 3 的倍数，即如果在 QEPA 和 QEPB 信号上同时检测到边沿转换，则可以选择性地生成中断。正交解码逻辑状态机制图中虚线标记的状态转换是产生正交相位错误的无效转换。

方向计数模式

当一些位置编码器提供方向和时钟输出而不是正交输出时，可以使用方向计数模式，其中 QEPA 输入将为位置计数器提供时钟，QEPB 输入将提供方向信息。如果方向输入为高，则位置计数器在 QEPA 输入的每个上升沿上递增，反之位置计数器递减。

35.8 位置计数器和控制单元（PCCU）功能描述

PCCU 提供 QEPCTL 和 QPOSCTL 两个配置寄存器用于设置位置计数器初始化，操作模式，锁存模式以及用于同步信号生成的位置比较逻辑。

35.8.1 初始化

可以通过软件初始化事件，选通事件和索引事件对位置计数器进行初始化。

表格 177 初始化位置计数器

事件	位域	描述
软件初始化	SWI	使 SWI 置位来使能软件初始化位置计数器，但是不能自动清除该位。当再次向该位写入 1 时（该位仍被设置），位置计数器会被重新初始化。
选通事件初始化	SEI	当 SEI=10 时，位置计数器在选通输入的上升沿时，使用 QPOSINIT 寄存器的值进行初始化。
		当 SEI=11 时，位置计数器在正向选通输入的上升沿和反向选通输入的下降沿时，使用 QPOSINIT 寄存器的值进行初始化。
索引事件初始化	IEI	QEPI 信号可用于在索引输入的上升沿/下降沿时触发位置计数器初始化。
		当 IEI=10 时，位置计数器在 QEPI 信号上升沿时，使用 QPOSINIT 寄存器的值进行初始化。
		当 IEI=11 时，位置计数器在 QEPI 信号下降沿时，使用 QPOSINIT 寄存器的值进行初始化。

35.8.2 配置位置计数器操作模式

根据 PCRM 位可以配置为以下模式来复位位置计数器：

- PCRM=00，索引事件
- PCRM=01，最大位置上
- PCRM=10，第一个索引事件
- PCRM=11，单元超时事件

在这四种模式中，位置计数器在上溢时会复位为 0，但是位置计数器在下溢时会复位为 QPOSMAX。如果在达到位置计数器最大值之后，位置计数器仍然继续向上计数时，则会发生上溢，并设置 PCO 位。如果位置计数器为 0 但是仍然继续向下计数时，则会发生下溢，并设置 PCU 位。

在索引事件上复位位置计数器

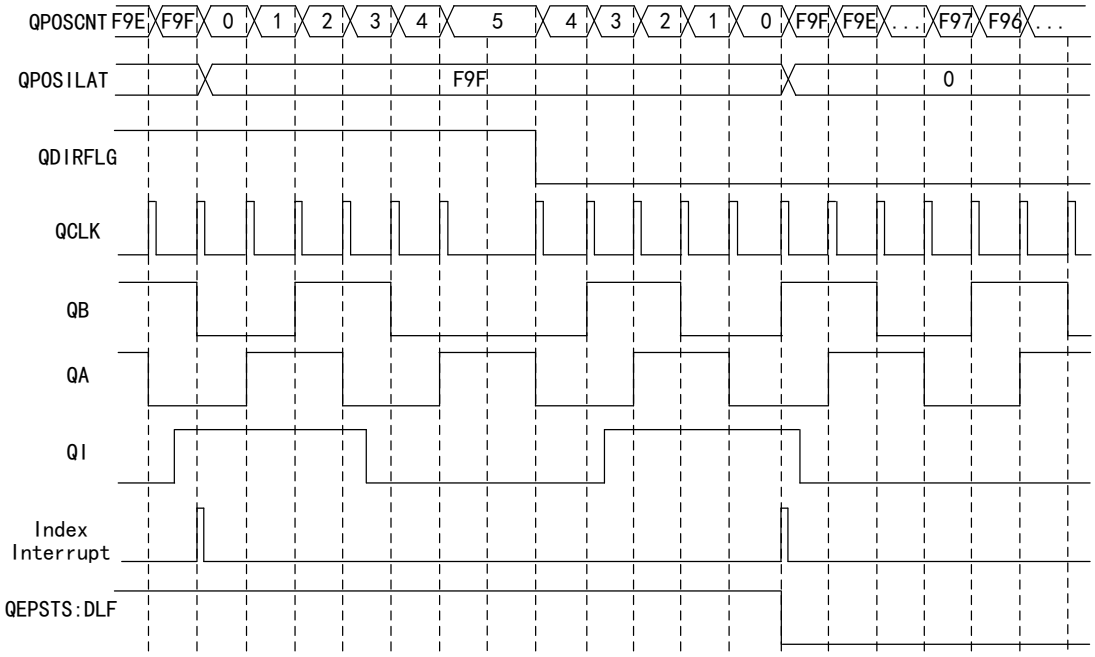
当索引事件发生在以下期间时，位置计数器将在下一个 QEP 时钟上进行复位：

- 顺时针方向期间，位置计数器复位为 0
- 逆时针方向期间，位置计数器复位为 QPOSMAX

将第一索引边沿之后的正交边沿定义为第一索引标记。QEPSTS 寄存器中出现的第一个索引标记 FIMF 和第一个索引标记上的方向 FIDF 由 QEP 外设记录。此

外，为了让相同的相对正交转换可以在索引事件上复位位置计数器，QEP 外设还标记在第一个索引标记上的正交边沿。当在顺时针方向 QEPB 下降沿发生第一次复位时，后续复位必须与顺时针方向 QEPB 下降沿对齐，逆时针方向则与 QEPB 上升沿对齐，如下图所示。

图 178 在索引事件上复位位置计数器



在每个索引事件标记的 QDLF 位中记录方向信息，并将 QPOSCNT 锁存到 QPOSILAT 寄存器。当锁存到 QPOSILAT 寄存器的值不等于 0 或 QPOSMAX 时，就会使 PCE（位置计数器错误中断标志）和 PCEF（位置计数器错误标志）置位。则只能通过软件清除 PCE 位（如果错误设置该位），并且在每个索引事件标记上更新 PCEF 位。

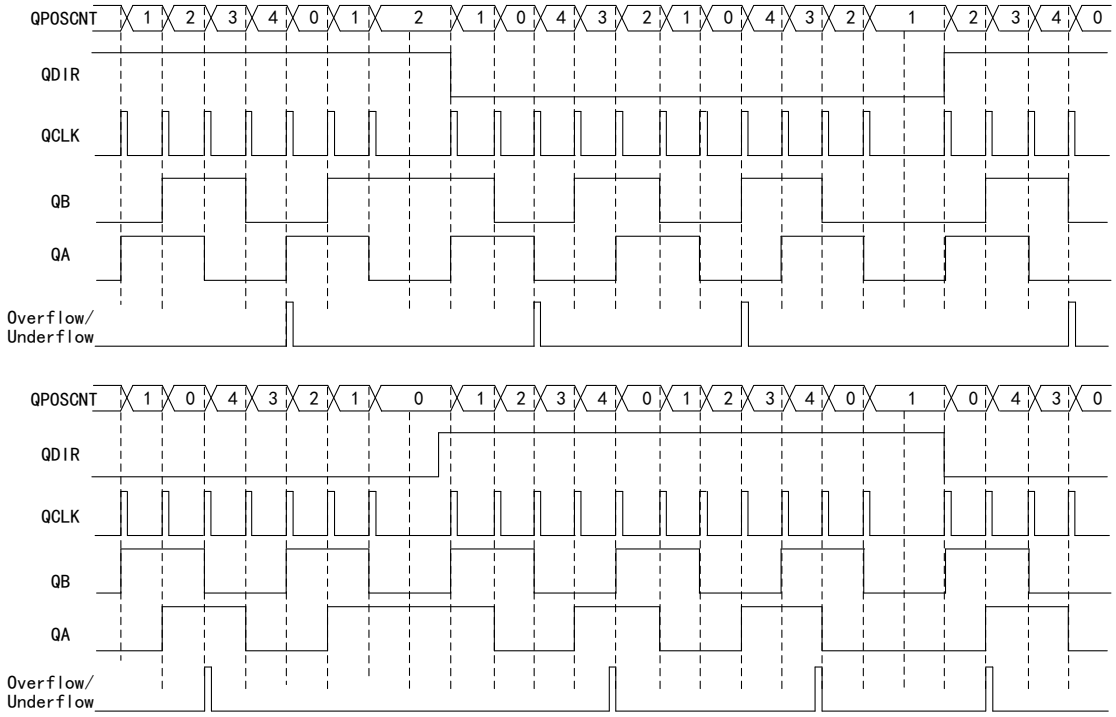
将 QPOSCNT 锁存到每个索引标记上的 QPOSILAT 寄存器中。如果仅在该模式下生成 PCE 位/PCEF 位和 PCRM=00，则索引事件锁存必须将 IEL 配置为 00 或 11。

在最大位置上复位位置计数器

第一个索引标记 FIMF 和 FIDF 不适用于此模式。

如下图所示在最大位置上复位位置计数器。如果 QPOSCNT=0，则位置计数器将会在下一个逆时针方向移动的 QEP 时钟上复位为 QPOSMAX，并使 PCU 置位。如果 QPOSCNT=QPOSMAX，则位置计数器将会在下一个顺时针方向移动 QEP 时钟上进行复位，并使 PCO 置位。

图 179 在 QPOSMAX=4 时复位位置计数器



在第一个索引事件上复位位置计数器

当第一个索引事件发生在以下期间时，位置计数器将在下一个 QEP 时钟上进行复位：

- 顺时针方向期间，位置计数器复位为 0
- 逆时针方向期间，位置计数器复位为 QPOSMAX

第一个索引标记 FIMF 和 FIDF 不适用于此模式。

注意：仅在第一个索引事件复位位置计数器，随后位置计数器将根据最大位置进行复位，详情请参见在最大位置上复位位置计数器章节。

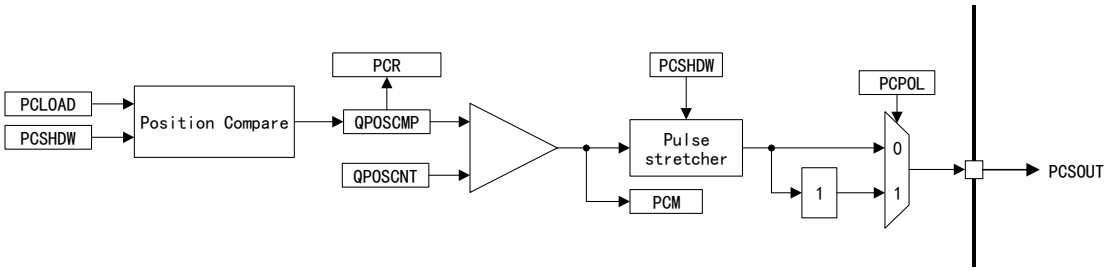
在单元超时事件上复位位置计数器

在该模式中，单元时间事件上 QSRC 位所选择的位置计数器模式决定 QPOSCNT=0 或 QPOSCNT=QPOSMAX。此时，该模式可以用于频率测量。

35.8.3 位置比较单元

在位置比较匹配时，位置比较单元用于产生同步输出和/或中断。位置比较单元结构框图如下所示，QPOSCMP 寄存器影子化，可以通过 QPOSCTL[PCSHDW]位来使能/禁用影子模式。CPU 将直接写入活动 QPOSCMP 寄存器（未使能影子模式时）。

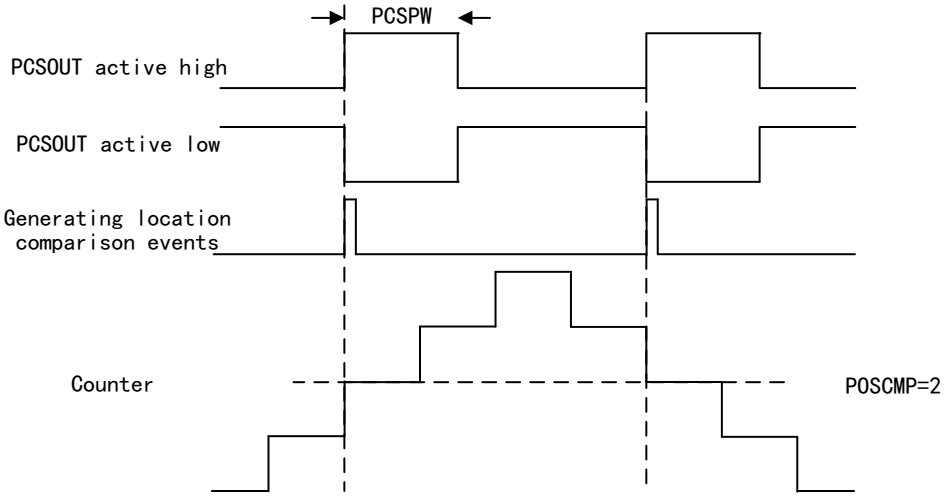
图 180 位置比较单元结构框图



为了在加载比较匹配事件和加载位置计数器 0 事件中将影子寄存器值加载到活动寄存器，并在加载后生成 PCR 中断，可以通过 PCLOAD 位配置位置比较单元（在影子模式下）。

当 QPOSCNT 与 QPOSCMP 匹配时，PCM 置位并且为了触发外部设备，在比较匹配时产生可编程脉冲宽度的位置比较同步输出。如下图所示，当 QPOSCMP=2 时，位置比较单元在 QEP 位置计数器向上计数从 1 转换到 2 以及在 QEP 位置计数器向下计数从 3 转换到 2 时生成一个位置比较事件。

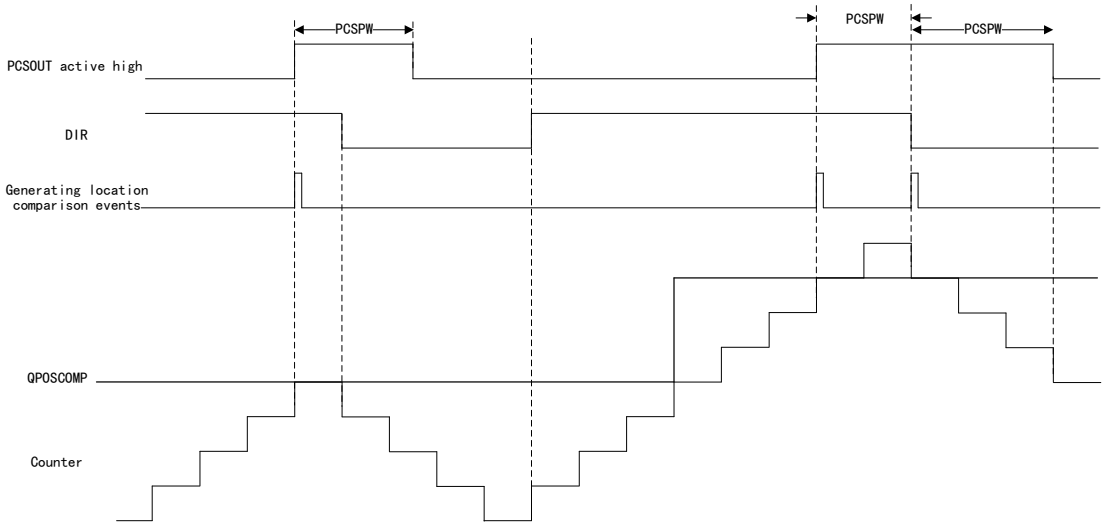
图 181 QEP 位置比较事件生成点



脉冲延展器逻辑

在位置比较匹配时，脉冲延展器逻辑就会产生一个可编程位置比较同步脉冲输出。如下图所示，在前一个位置比较脉冲仍然有效的情况下，当出现新的位置比较匹配时，脉冲延展器就会生成指定持续时间的脉冲（在新位置比较事件中生成）。

图 182 位置比较同步输出脉冲延展器



同步输出

在 QPOSCNT 寄存器和 QPOSCMP 寄存器进行比较匹配时，位置比较单元用于生成位置比较同步信号。该同步信号可以用于 QEP 外设的索引/选通引脚输出。可以通过置位 SOEN 来使能位置比较同步输出，通过 SPSEL 位选择同步输出引脚。

35.8.4 锁存模式

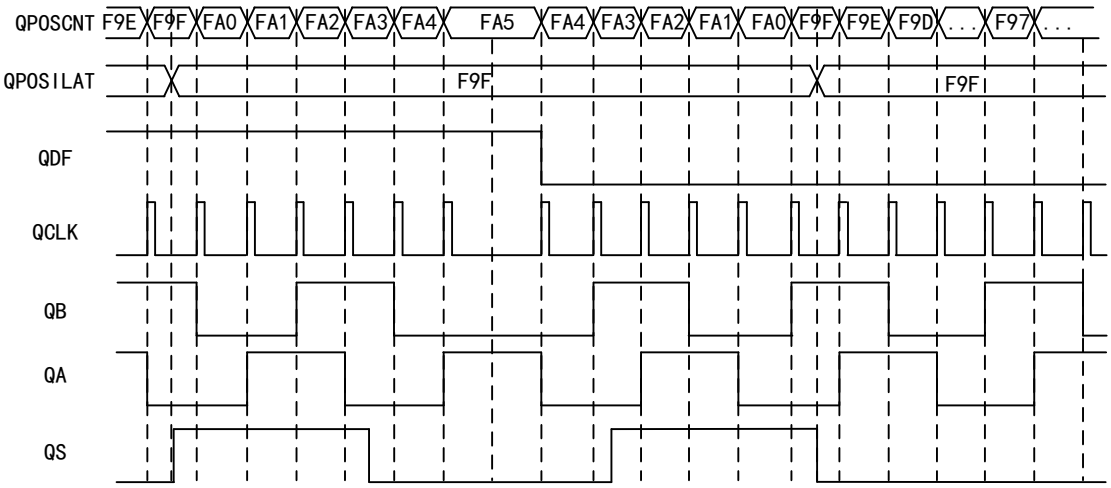
在索引引脚或选通引脚上发生确定事件时，QEP 索引和选通输入可以配置为将位置计数器分别锁存到 QPOSILAT 寄存器和 QPOSSLAT 寄存器中。

选通事件锁存

如选通事件锁存所示，如果设置了 SEL 位，则在顺时针方向时，QPOSCNT 在选通输入的上升沿被锁存到 QPOSSLAT 寄存器。在逆时针方向时，QPOSCNT 在选通输入的下降沿被锁存。清除 SEL 位时，QPOSCNT 在选通输入的上升沿被锁存到 QPOSSLAT 寄存器。

如果 QPOSCNT 锁存到 QPOSSLAT 寄存器，则设置 SEL 标志位。

图 183 选通事件锁存

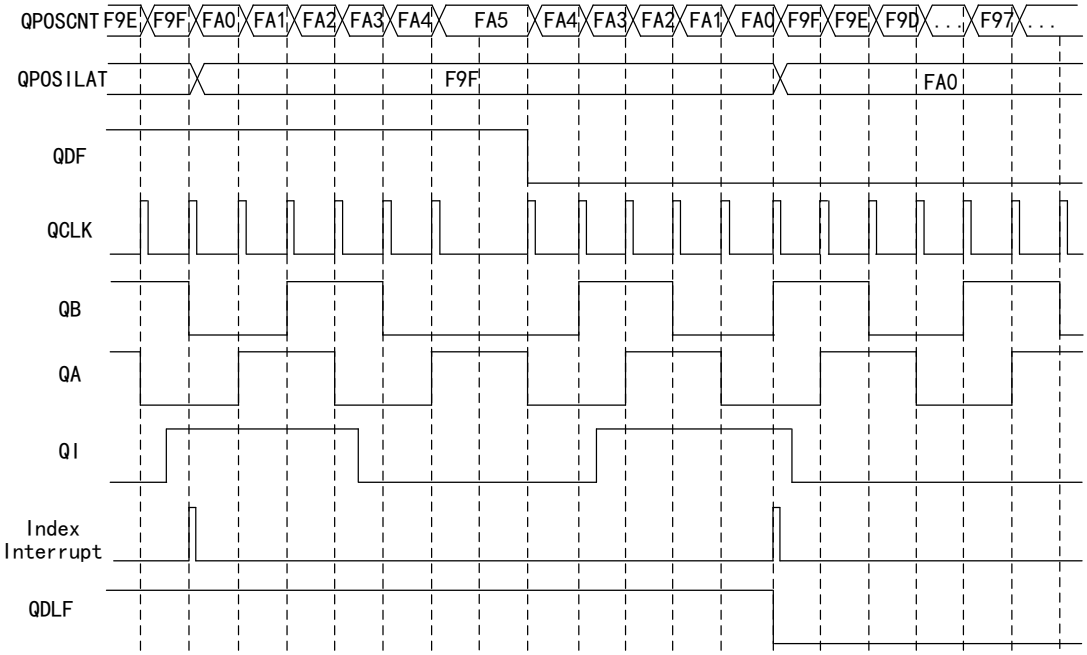


索引事件锁存

如果不想在每个索引事件上复位位置计数器，则需要在全 32 位模式下复位位置计数器，即设置 PCRM=01 和 PCRM=10。位置计数器根据 IEL 位可以选择为上升沿锁存（IEL=01），下降沿锁存（IEL=10）和软件索引标记锁存（IEL=11），根据 QDLF 位可以在每一个索引事件上记录方向信息，使用索引事件标记 PCRM 如图所示。

错误检测机制可以根据索引事件锁存来检查位置计数器是否累加正确计数（在索引事件之间）。如果 QPOSCNT 锁存到 QPOSILAT 寄存器，则设置 IEL 位。在 PCRM=00 时，将忽略 IEL 位。

图 184 1000 行编码器软件索引标记(QEPCTL[IEL] = 1)



上升沿锁存（IEL=01）

在索引输入的每个上升沿将 QPOSCNT 锁存到 QPOSILAT 寄存器。

下降沿锁存（IEL=10）

在索引输入的每个下降沿将 QPOSCNT 锁存到 QPOSILAT 寄存器。

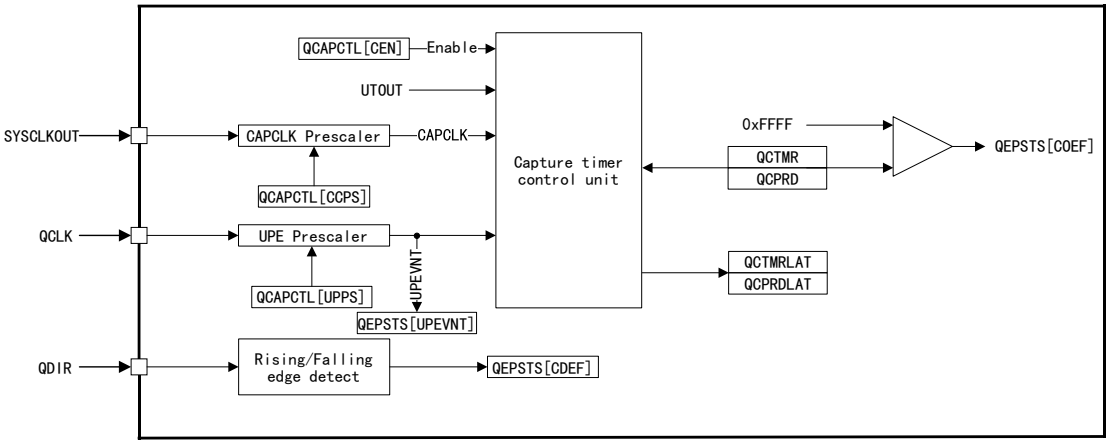
软件索引标记锁存（IEL=11）

将第一索引边沿之后的正交边沿定义为第一索引标记。QEPSTS 寄存器中出现的第一个索引标记 FIMF 和第一个索引标记上的方向 FIDF 由 QEP 外设记录。此外，为了使 IEL=11，即让相同的相对正交转换锁存位置计数器，QEP 外设还标记在第一个索引标记上的正交边沿。

35.9 边沿捕获单元功能描述

边沿捕获单元用于测量单元位置事件之间经过的时间，其结构框图如下所示。

图 185 边沿捕获单元结构框图



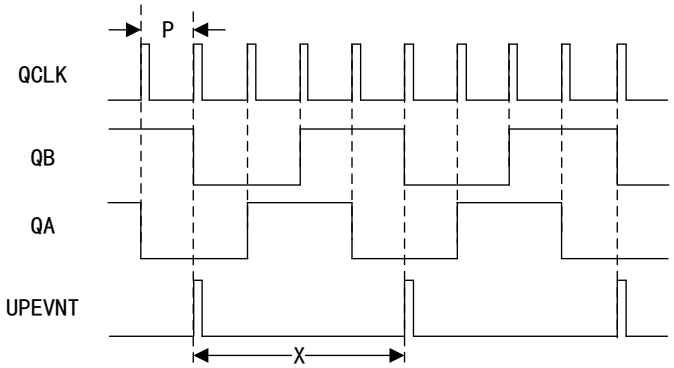
边沿捕获单元功能通常通过以下公式进行低速测量：

$$v(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T}$$

表格 178 补充说明

参数	描述
v（k）	时刻 k 的速度
t（k）	时刻 k 的时间
t（k-1）	时刻 k-1 的时间
X	由正交边沿的整数倍决定单元位置，如单元位置事件时序图所示
ΔT	单元位置事件的经过时间

图 186 单元位置事件时序图



X= P * Quadrature periods (UPPS bit selection)

从 SYSCLKOUT 运行捕获定时器 QCTMR，通过 CCPS 位域选择 QEP 捕获定时器时钟预分频。在每个单元位置事件中，将 QCTMR 锁存到 QCTMRLAT 寄存器，并对 QCTMR 进行复位，通过设置单元位置事件标志位 UPEVNT，指示将 QCTMR 新值锁存到 QCTMRLAT 寄存器。在读取进行低速测量的周期寄存器之前，软件可以检查 UPEVNT 位并通过写入 1 清除此位。

如果单元位置事件之间未发生方向变化或单元位置事件之间发生的计数小于 65535 次，则单元位置事件之间的时间测量具有准确性。

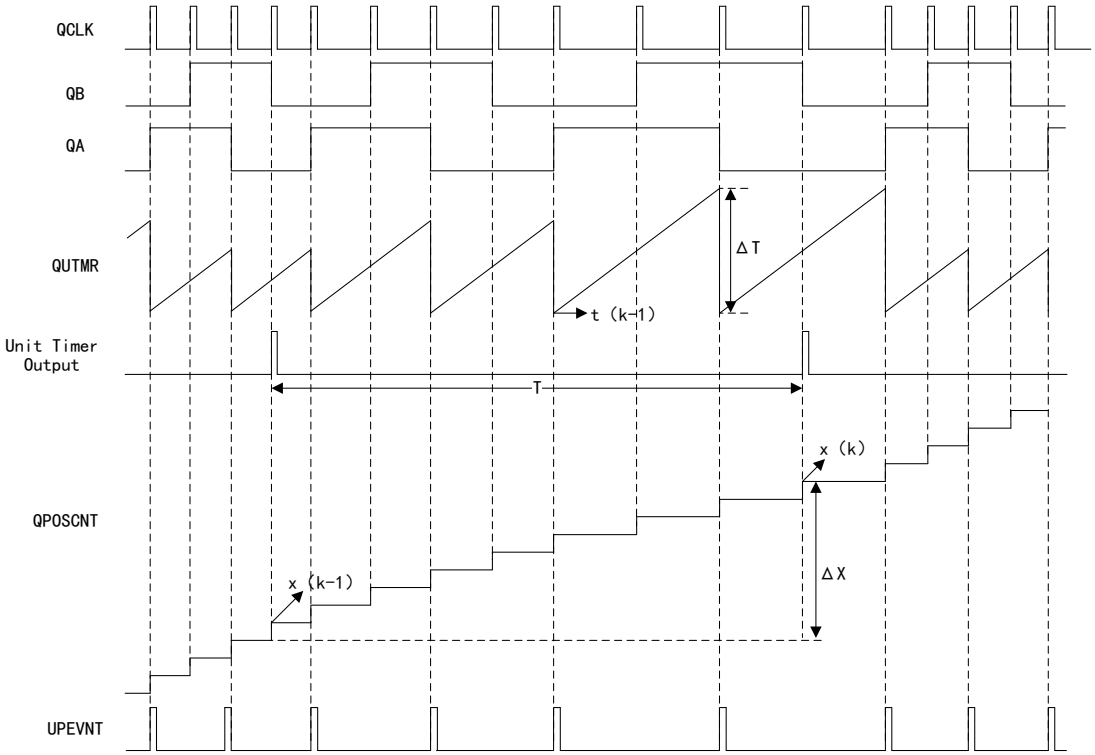
在单元位置事件之间发生捕获定时器溢出时，将 QCPRDLAT 寄存器设置为 FFFF 并置位 COEF。在单元位置事件之间发生方向变化时，将 QCPRDLAT 寄存器设置为 FFFF 并置位 CDEF。

QCTMR 寄存器和 QCPRD 寄存器可以配置为锁存单元超时事件和 CPU 读取 QPOSCNT 寄存器。

当清除了 QCLM 位且 CPU 读取 QPOSCNT 寄存器时，QCTMR 将锁存到 QCTMRLAT 寄存器中，而 QCPRD 将锁存到 QCPRDLAT 寄存器中。当清除了 QCLM 位且单元超时时，QPOSCNT，QCTMR 和 QCPRD 将分别锁存到 QPOSLAT 寄存器，QCTMRLAT 寄存器和 QCPRDLAT 寄存器中。

注意：为了避免预分频器出现错误，不允许动态修改 UPPS 位域，比如将 UPE 从 QCLK/4 切换到 QCLK/8。仅允许在禁用边沿捕获单元之后动态修改 CCPS 位域，比如将 CAPCLK 从 SYSCLK/4 切换到 SYSCLK/8。

图 187 边沿捕获单元时序图



表格 179 边沿捕获单元时序图补充说明

参数	描述	配置信息
v（k）	时刻 k 的速度	-
x（k）	时刻 k 的位置	-
x（k-1）	时刻 k-1 的位置	-
X	固定单元位置	固定单元位置由 UPPS 位域和传感器分辨率定义
ΔX	单元时间内的增量位置移动	增量位置=QPOSLAT（k）- QPOSLAT（k-1）
t（k）	时刻 k 的时间	-
t（k-1）	时刻 k-1 的时间	-
T	固定单元时间或速度计算速率的倒数	QUPRD 寄存器
ΔT	单元位置移动的增量时间	QCPRDLAT 寄存器

根据边沿捕获单元时序图，其速度计算公式为：

$$v(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T}$$

根据边沿捕获单元时序图补充说明表格可知，通过 UPPS 位域和 QUPRD 寄存器来配置单元周期和单元时间，而增量位置输出和增量时间输出由 QPOSLAT 寄存器和 QCPRDLAT 寄存器提供。

35.10 寄存器组地址

表格 180 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
Qep1Regs	QEP_REGS	0x4001_3800	0x4001_3BFF
Qep2Regs	QEP_REGS	0x4001_3C00	0x4001_3FFF

35.11 寄存器地址映射

表格 181 QEP_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
QPOSCNT	位置计数器寄存器	0x00	-
QPOSINIT	初始化位置计数器寄存器	0x04	-
QPOSMAX	最大位置计数器寄存器	0x08	-
QPOSCMP	位置比较寄存器	0x0C	-
QPOSILAT	索引位置锁存寄存器	0x10	-
QPOSSLAT	选通位置锁存寄存器	0x14	-
QPOSLAT	位置锁存寄存器	0x18	-
QUTMR	QEP 单元定时器寄存器	0x1C	-
QUPRD	QEP 单元周期寄存器	0x20	-
QWDTMR	QEP 看门狗定时器寄存器	0x24	-
QWDPRD	QEP 看门狗周期寄存器	0x26	-
QDECCTL	正交解码器控制寄存器	0x28	-
QEPCTL	QEP 控制寄存器	0x2A	-
QCAPCTL	正交捕获控制寄存器	0x2C	-
QPOSCTL	位置比较控制寄存器	0x2E	-
QEINT	使能 QEP 中断寄存器	0x30	-
QFLG	QEP 中断标志寄存器	0x32	-
QCLR	清除 QEP 中断寄存器	0x34	-
QFRC	强制 QEP 中断寄存器	0x36	-
QEPSTS	QEP 标志寄存器	0x38	-
QCTMR	QEP 捕获定时器寄存器	0x3A	-
QCPRD	QEP 捕获周期寄存器	0x3C	-
QCTMRLAT	QEP 捕获定时器锁存寄存器	0x3E	-
QCPRDLAT	QEP 捕获周期锁存寄存器	0x40	-

寄存器名称	寄存器描述	偏移地址	WRPRT
REV	QEP 修订编号寄存器	0x60	-
QEPSTROBESEL	选通源选择寄存器	0x64	-
QMACTRL	QMA 控制寄存器	0x68	-

35.12 寄存器功能描述

35.12.1 位置计数器寄存器（QPOSCNT）

偏移地址：0x00

复位类型：SYSRSn

根据方向输入，32 位位置计数器寄存器对每个 QEP 脉冲进行向上计数或向下计数。当计数器配置为向上计数模式或向下计数模式时，该寄存器为只读。如果计数器作为位置积分器，则选定参考点的位置与计数值成正比。

位/域	名称	R/W	描述	复位值
31:0	QPOSCNT	R/W	位置计数器（Position Counter） 注意：建议在初始化期间（即 QPEN=0 时）写入 QPOSCNT。如果在 QPEN=1 时写入 QPOSCNT，则会导致出现错误。	0h

35.12.2 初始化位置计数器寄存器（QPOSINIT）

偏移地址：0x04

复位类型：SYSRSn

对该寄存器的写入应该始终为 32 位写入。

位/域	名称	R/W	描述	复位值
31:0	QPOSINIT	R/W	初始化位置计数器（Position Counter initialize） 可以通过软件对位置计数器进行初始化。该寄存器包含位置值，可以用于根据外部选通事件或索引事件初始化位置计数器。	0h

35.12.3 最大位置计数器寄存器（QPOSMAX）

偏移地址：0x08

复位类型：SYSRSn

对该寄存器的写入应该始终为 32 位写入。

位/域	名称	R/W	描述	复位值
31:0	QPOSMAX	R/W	最大位置计数器（Maximum Position Counter） 该寄存器包含最大位置计数器值。	0h

35.12.4 位置比较寄存器（QPOSCMP）

偏移地址：0x0C

复位类型：SYSRSn

对该寄存器的写入应该始终为 32 位写入。

位/域	名称	R/W	描述	复位值
31:0	QPOSCMP	R/W	位置比较（Position Compare） 该寄存器包含位置比较值，将位置比较值与 QPOSCNT 进行比较，可以在比较匹配时产生中断和/或同步输出。	0h

35.12.5 索引位置锁存寄存器（QPOSILAT）

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	QPOSILAT	R	索引位置锁存（Index Position Latch） 在 IEL 位域定义的索引事件中，将位置计数器锁存到该寄存器中。	0h

35.12.6 选通位置锁存寄存器（QPOSSLAT）

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	QPOSSLAT	R	选通位置锁存（Strobe Position Latch） 在 SEL 位定义的选通事件中，将位置计数器锁存到该寄存器中。	0h

35.12.7 位置锁存寄存器（QPOSLAT）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	QPOSLAT	R	位置锁存（Position Latch） 在单元超时事件中，将位置计数器锁存到该寄存器中。	0h

35.12.8 QEP 单元定时器寄存器（QUTMR）

偏移地址：0x1C

复位类型：SYSRSn

对该寄存器的写入应该始终为 32 位写入。

位/域	名称	R/W	描述	复位值
31:0	QUTMR	R/W	QEP 单元定时器（QEP Unit Timer） 该寄存器可以作为单元时间事件产生的时基。在单元时间周期值与该定时器值匹配时，产生单元时间事件。	0h

35.12.9 QEP 单元周期寄存器（QUPRD）

偏移地址：0x20

复位类型：SYSRSn

对该寄存器的写入应该始终为 32 位写入。

位/域	名称	R/W	描述	复位值
31:0	QUPRD	R/W	QEP 单元周期（QEP Unit Period） 该寄存器包含产生周期性单元时间事件的单元计数器周期值。这些事件根据周期性间隔来锁定 QEP 位置，并可以选择产生中断。	0h

35.12.10 QEP 看门狗定时器寄存器（QWDTMR）

偏移地址：0x24

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	QWDTMR	R/W	QEP 看门狗定时器（QEP Watchdog Timer） 该寄存器可以作为看门狗检查电机失速产生的时基。在指示运动正交时钟边沿转换时，复位这个寄存器。在看门狗周期值与该定时器值匹配时，产生看门狗超时中断。	0h

35.12.11 QEP 看门狗周期寄存器（QWDPRD）

偏移地址：0x26

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	QWDPRD	R/W	QEP 看门狗周期（QEP Watchdog Period） 该寄存器包含 QEP 看门狗定时器的超时值。在看门狗周期值与该定时器值匹配时，产生看门狗超时中断。	0h

35.12.12 正交解码器控制寄存器（QDECCTL）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	保留			0h
5	QSP	R/W	反转 QEPS 输入（Negates QEPS input） 0：无影响 1：反转 QEPS 输入	0h
6	QIP	R/W	反转 QEPI 输入（Negates QEPI input） 0：无影响 1：反转 QEPI 输入	0h
7	QBP	R/W	反转 QEPB 输入（Negates QEPB input） 0：无影响 1：反转 QEPB 输入	0h
8	QAP	R/W	反转 QEPA 输入（Negates QEPA input） 0：无影响 1：反转 QEPA 输入	0h
9	IGATE	R/W	选择索引脉冲门控（Index pulse gating select） 0：禁止 1：选通索引引脚门控	0h
10	SWAP	R/W	使能交换正交时钟输入（swapped Quadrature-clock inputs enable） 0：禁止 1：使能	0h
11	XCR	R/W	选择外部时钟速率（External Clock Rate select） 0：2 倍分辨率（计算上升沿/下降沿） 1：1 倍分辨率（仅计算上升沿）	0h
12	SPSEL	R/W	选择同步输出引脚（Sync output pin select） 0：索引引脚 1：选通引脚	0h

位/域	名称	R/W	描述	复位值
13	SOEN	R/W	使能位置比较同步输出（position-compare sync output Enable） 0：禁止 1：使能	0h
15:14	QSRC	R/W	选择位置计数器模式（Position-counter mode select） 00：正交计数模式，QCLK=iCLK，QDIR=iDIR 01：方向计数模式，QCLK=xCLK，QDIR=xDIR 10：向上计数模式，QCLK=xCLK，QDIR=1 11：向下计数模式，QCLK=xCLK，QDIR=0	0h

35.12.13 QEP 控制寄存器（QEPCTL）

偏移地址：0x2A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	WDE	R/W	使能 QEP 看门狗（QEP watchdog enable） 0：禁止 1：使能	0h
1	UTE	R/W	使能 QEP 单元定时器（QEP unit timer enable） 0：禁止 1：使能	0h
2	QCLM	R/W	QEP 捕获锁存模式（QEP capture latch mode） 0：CPU 读取锁存位置计数器。如果 CPU 读取 QPOSCNT 寄存器，则将捕获定时器值和捕获周期值锁存到 QCTMRLAT 和 QCPRDLAT 寄存器中。 1：锁存单元超时。如果单元超时，则将捕获定时器、捕获周期值和位置计数器锁存到 QPOSLAT、QCTMRLAT 和 QCPRDLAT 寄存器中。	0h
3	QPEN	R/W	使能 QEP 位置计数器（Quadrature position counter enable） 0：禁止，QFLG 寄存器其中一些标志不会被清除或复位，并显示这些标志的实际状态。控制/配置寄存器不受软件复位干扰。复位 QEP 外设内部操作标志/只读寄存器。 1：使能	0h
5:4	IEL	R/W	选择位置计数器的索引事件锁存（position counter Index event latch select） 00：保留 01：上升沿锁存 10：下降沿锁存 11：软件索引标记锁存（适用于软件索引标记）。锁存索引事件标记上的正交方向标志（锁存到 QDLF 位）和位置计数器（锁存到 QPOSILAT 寄存器）。	0h

位/域	名称	R/W	描述	复位值
6	SEL	R/W	选择位置计数器的选通事件锁存（position counter Strobe event latch select） 0：如果 QPOSSLAT=QPOSCNT，则位置计数器在 QEPS 选通的上升沿被锁存。通过反转选通输入（使用 QSP 位）来实现在下降沿上锁存位置计数器。 1：顺时针方向时，位置计数器在 QEPS 选通的上升沿被锁存。逆时针方向时，位置计数器在 QEPS 选通的下降沿被锁存。	0h
7	SWI	R/W	使能软件初始化位置计数器（software initialize position counter Enable） 该位不会自动清除。 0：禁用 1：使能（QPOSCNT=QPOSINIT）	0h
9:8	IEI	R/W	选择位置计数器的索引事件初始化（Index event init of position count select） 00：禁用 01：禁用 10：QPOSCNT=QPOSINIT 时，在 QEPI 信号的上升沿时，对位置计数器进行初始化 11：QPOSCNT=QPOSINIT 时，在 QEPI 信号的下降沿时，对位置计数器进行初始化	0h
11:10	SEI	R/W	选择位置计数器的选通事件初始化（Strobe event init of position count select） 00：禁用 01：禁用 10：在 QEPS 信号的上升沿初始化位置计数器 11：顺时针方向时，在 QEPS 选通的上升沿时，对位置计数器进行初始化。逆时针方向时，在 QEPS 选通的下降沿时，对位置计数器进行初始化。	0h
13:12	PCRM	R/W	复位位置计数器（Position counter reset） 00：在索引事件上，复位位置计数器 01：在最大位置计数器上，复位位置计数器 10：在第一个索引事件上，复位位置计数器 11：在单元时间事件上，复位位置计数器	0h

位/域	名称	R/W	描述	复位值
15:14	FREE_SOFT	R/W	选择模拟模式（Emulation mode select） QPOSCNT 行为 00：模拟挂起时，位置计数器立即停止 01：位置计数器继续计数，直到翻转 10：位置计数器不受模拟挂起影响 11：位置计数器不受模拟挂起影响 QWDTMR 行为 00：看门狗定时器立即停止 01：看门狗定时器计数，直到 WDT 周期匹配翻转 10：看门狗定时器不受模拟挂起影响 11：看门狗定时器不受模拟挂起影响 QUTMR 行为 00：单元定时器立即停止 01：单元定时器计数，直到周期翻转 10：单元定时器不受模拟挂起的影响 11：单元定时器不受模拟挂起的影响 QCTMR 行为 00：捕获定时器立即停止 01：捕获定时器计数，直到下一个单元周期事件 10：捕获定时器不受模拟挂起的影响 11：捕获定时器不受模拟挂起的影响	0h

35.12.14 正交捕获控制寄存器（QCAPCTL）

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	UPPS	R/W	选择单元位置事件预分频（Unit position event prescale select） 0000：QCLK/1=UPE 0001：QCLK/2=UPE 0010：QCLK/4=UPE 0011：QCLK/8=UPE 0100：QCLK/16=UPE 0101：QCLK/32=UPE 0110：QCLK/64=UPE 0111：QCLK/128=UPE 1000：QCLK/256=UPE 1001：QCLK/512=UPE 1010：QCLK/1024=UPE 1011：QCLK/2048=UPE 其他：保留	0h

位/域	名称	R/W	描述	复位值
6:4	CCPS	R/W	选择 QEP 捕获定时器时钟预分频（QEP capture timer clock prescale select） 000：SYSCLKOUT/1=CAPCLK 001：SYSCLKOUT/2=CAPCLK 010：SYSCLKOUT/4=CAPCLK 011：SYSCLKOUT/8=CAPCLK 100：SYSCLKOUT/16=CAPCLK 101：SYSCLKOUT/32=CAPCLK 110：SYSCLKOUT/64=CAPCLK 111：SYSCLKOUT/128=CAPCLK	0h
14:7	保留			0h
15	CEN	R/W	使能 QEP 捕获（QEP capture Enable） 0：禁止 1：使能	0h

35.12.15 位置比较控制寄存器（QPOSCTL）

偏移地址：0x2E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
11:0	PCSPW	R/W	选择位置比较同步输出脉冲宽度（position compare sync output pulse width Select） 0000 0000 0000：1*SYSCLKOUT 循环*4 0000 0000 0001：2*SYSCLKOUT 循环*4 ... 1111 1111 1111：4096*SYSCLKOUT 循环*4	0h
12	PCE	R/W	使能位置比较（Position compare enable） 0：禁止 1：使能	0h
13	PCPOL	R/W	选择同步输出极性（sync output Polarity Select） 0：高脉冲 1：低脉冲	0h
14	PCLOAD	R/W	影子加载位置比较（Position compare of shadow load） 0：在 QPOSCNT=0 时加载 1：在 QPOSCNT=QPOSCMP 时加载	0h
15	PCSHDW	R/W	使能影子位置比较（Position compare of shadow enable） 0：禁止，立即加载 1：使能	0h

35.12.16 使能 QEP 中断寄存器（QEINT）

偏移地址：0x30

复位类型：SYSRSn

该寄存器用于使能/禁止各中断事件源。

位/域	名称	R/W	描述	复位值
0	保留			0h

位/域	名称	R/W	描述	复位值
1	PCE	R/W	使能位置计数器错误中断（Position counter error interrupt enable） 0：禁止 1：使能	0h
2	PHE	R/W	使能正交相位错误中断（Quadrature phase error interrupt enable） 0：禁止 1：使能	0h
3	QDC	R/W	使能正交方向改变中断（Quadrature direction change interrupt enable） 0：禁止 1：使能	0h
4	WTO	R/W	使能看门狗超时中断（Watchdog time out interrupt enable） 0：禁止 1：使能	0h
5	PCU	R/W	使能位置计数器下溢中断（Position counter underflow interrupt enable） 0：禁止 1：使能	0h
6	PCO	R/W	使能位置计数器上溢中断（Position counter overflow interrupt enable） 0：禁止 1：使能	0h
7	PCR	R/W	使能位置比较就绪中断（Position compare ready interrupt enable） 0：禁止 1：使能	0h
8	PCM	R/W	使能位置比较匹配中断（Position compare match interrupt enable） 0：禁止 1：使能	0h
9	SEL	R/W	使能选通事件锁存中断（Strobe event latch interrupt enable） 0：禁止 1：使能	0h
10	IEL	R/W	使能索引事件锁存中断（Index event latch interrupt enable） 0：禁止 1：使能	0h
11	UTO	R/W	使能单元超时中断（Unit time out interrupt enable） 0：禁止 1：使能	0h
12	QMAE	R/W	使能 QMA 错误中断（QMA Error Interrupt enable） 0：禁止 1：使能	0h
15:13	保留			0h

35.12.17 QEP 中断标志寄存器（QFLG）

偏移地址：0x32

复位类型：SYSRSn

该寄存器指示是否有锁存任何中断事件（包含全局中断标志）。

位/域	名称	R/W	描述	复位值
0	INT	R	全局中断标志（Global interrupt flag） 0：未产生中断 1：产生中断	0h
1	PCE	R	位置计数器错误中断标志（Position counter error interrupt flag） 0：未产生中断 1：产生中断	0h
2	PHE	R	正交相位错误中断标志（Quadrature phase error interrupt flag） 该位在 QEPA 和 QEPB 同时转换时设置。 0：未产生中断 1：产生中断	0h
3	QDC	R	正交方向改变中断标志（Quadrature direction change interrupt flag） 0：未产生中断 1：产生中断	0h
4	WTO	R	看门狗超时中断标志（Watchdog time out interrupt flag） 该位在看门狗超时时设置。 0：未产生中断 1：产生中断	0h
5	PCU	R	位置计数器下溢中断标志（Position counter underflow interrupt flag） 该位在位置计数器下溢时设置。 0：未产生中断 1：产生中断	0h
6	PCO	R	位置计数器上溢中断标志（Position counter overflow interrupt flag） 该位在位置计数器上溢时设置。 0：未产生中断 1：产生中断	0h
7	PCR	R	位置比较就绪中断标志（Position compare ready interrupt flag） 该位在将影子寄存器值传输到活动 QPOSCMP 寄存器之后设置。 0：未产生中断 1：产生中断	0h
8	PCM	R	位置比较匹配中断标志（Position compare match interrupt flag） 该位在位置比较匹配时设置。 0：未产生中断 1：产生中断	0h

位/域	名称	R/W	描述	复位值
9	SEL	R	选通事件锁存中断标志（Strobe event latch interrupt flag） 该位在将 QPOSCNT 锁存到 QPOSSLAT 之后设置。 0：未产生中断 1：产生中断	0h
10	IEL	R	索引事件锁存中断标志（Index event latch interrupt flag） 该位在将 QPOSCNT 锁存到 QPOSILAT 之后设置。 0：未产生中断 1：产生中断	0h
11	UTO	R	单元超时中断标志（Unit time out interrupt flag） 该位在单元定时器周期匹配时设置。 0：未产生中断 1：产生中断	0h
12	QMAE	R	QMA 错误中断标志（QMA Error Interrupt flag） 0：未产生中断 1：产生中断	0h
15:13	保留			0h

35.12.18 清除 QEP 中断寄存器（QCLR）

偏移地址：0x34

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INT	R/S	清除全局中断标志（Global interrupt flag Clear） 0：无效 1：清除 INT	0h
1	PCE	R/S	清除位置计数器错误中断标志（Position counter error interrupt flag Clear） 0：无效 1：清除 PCE	0h
2	PHE	R/S	清除正交相位错误中断标志（Quadrature phase error interrupt flag Clear） 0：无效 1：清除 PHE	0h
3	QDC	R/S	清除正交方向改变中断标志（Quadrature direction change interrupt flag Clear） 0：无效 1：清除 QDC	0h
4	WTO	R/S	清除看门狗超时中断标志（Watchdog time out interrupt flag Clear） 0：无效 1：清除 WTO	0h
5	PCU	R/S	清除位置计数器下溢中断标志（Position counter underflow interrupt flag Clear） 0：无效 1：清除 PCU	0h

位/域	名称	R/W	描述	复位值
6	PCO	R/S	清除位置计数器上溢中断标志（Position counter overflow interrupt flag Clear） 0：无效 1：清除 PCO	0h
7	PCR	R/S	清除位置比较就绪中断标志（Position compare ready interrupt flag Clear） 0：无效 1：清除 PCR	0h
8	PCM	R/S	清除位置比较匹配中断标志（Position compare match interrupt flag Clear） 0：无效 1：清除 PCM	0h
9	SEL	R/S	清除选通事件锁存中断标志（Strobe event latch interrupt flag Clear） 0：无效 1：清除 SEL	0h
10	IEL	R/S	清除索引事件锁存中断标志（Index event latch interrupt flag Clear） 0：无效 1：清除 IEL	0h
11	UTO	R/S	清除单元超时中断标志（Unit time out interrupt flag Clear） 0：无效 1：清除 UTO	0h
12	QMAE	R/S	清除 QMA 错误中断标志（QMA Error Interrupt flag Clear） 0：无效 1：清除 QMAE	0h
15:13	保留			0h

35.12.19 强制 QEP 中断寄存器（QFRC）

偏移地址：0x36

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	PCE	R/W	强制位置计数器错误中断（Force Position counter error interrupt） 0：无效 1：强制位置计数器错误中断	0h
2	PHE	R/W	强制正交相位错误中断（Force Quadrature phase error interrupt） 0：无效 1：强制正交相位错误中断	0h

位/域	名称	R/W	描述	复位值
3	QDC	R/W	强制正交方向改变中断（Force Quadrature direction change interrupt） 0：无效 1：强制正交方向改变中断	0h
4	WTO	R/W	强制看门狗超时中断（Force Watchdog time out interrupt） 0：无效 1：强制看门狗超时中断	0h
5	PCU	R/W	强制位置计数器下溢中断（Force Position counter underflow interrupt） 0：无效 1：强制位置计数器下溢中断	0h
6	PCO	R/W	强制位置计数器上溢中断（Force Position counter overflow interrupt） 0：无效 1：强制位置计数器上溢中断	0h
7	PCR	R/W	强制位置比较就绪中断（Force Position compare ready interrupt） 0：无效 1：强制位置比较就绪中断	0h
8	PCM	R/W	强制位置比较匹配中断（Force Position compare match interrupt） 0：无效 1：强制位置比较匹配中断	0h
9	SEL	R/W	强制选通事件锁存中断（Force Strobe event latch interrupt） 0：无效 1：强制选通事件锁存中断	0h
10	IEL	R/W	强制索引事件锁存中断（Force Index event latch interrupt） 0：无效 1：强制索引事件锁存中断	0h
11	UTO	R/W	强制单元超时中断（Force Unit time out interrupt） 0：无效 1：强制单元超时中断	0h
12	QMAE	R/W	强制 QMA 错误中断（Force QMA Error Interrupt） 0：无效 1：强制 QMA 错误中断	0h
15:13	保留			0h

35.12.20 QEP 标志寄存器（QEPSTS）

偏移地址：0x38

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PCEF	R	位置计数器错误标志（Position counter error flag） 该位会对每一个索引事件进行更新。 0：在上一次索引转换期间，未发生位置计数器错误 1：发生位置计数器错误	0h
1	FIMF	R/W	第一个索引标记标志（First index marker flag） 写入 1 可以清除此位。 0：未发生第一个索引脉冲 1：发生第一个索引脉冲时，进行设置	0h
2	CDEF	R/W	捕获方向错误标志（Capture direction error flag） 写入 1 可以清除此位。 0：未发生捕获方向错误 1：在捕获位置事件之间，方向发生变化	0h
3	COEF	R/W	捕获溢出错误标志（Capture overflow error flag） 写入 1 可以清除此位。 0：未发生捕获溢出 1：QEP 捕获定时器发生溢出	0h
4	QDLF	R	QEP 方向锁存标志（QEP direction latch flag） 0：逆时针旋转（在索引事件标记上） 1：顺时针旋转（在索引事件标记上）	0h
5	QDF	R	正交方向标志（Quadrature direction flag） 0：逆时针旋转 1：顺时针旋转	0h
6	FIDF	R	第一个索引标记上的方向（Direction on the first index marker） 0：逆时针旋转 1：顺时针旋转	0h
7	UPEVNT	R/W	单元位置事件标志（Unit position event flag） 写入 1 可以清除此位。 0：未发生单元位置事件 1：发生单元位置事件	1h
15:8	保留			0h

35.12.21 QEP 捕获定时器寄存器（QCTMR）

偏移地址：0x3A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	QCTMR	R/W	捕获定时器（Capture Timer） 该寄存器位边沿捕获单元提供时基。	0h

35.12.22 QEP 捕获周期寄存器（QCPRD）

偏移地址：0x3C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	QCPRD	R/W	捕获周期（Capture Period） 在最后连续的 QEP 位置事件之间，该寄存器保存周期计数值。	0h

35.12.23 QEP 捕获定时器锁存寄存器（QCTMRLAT）

偏移地址：0x3E

复位类型：SYSRSn

位/域	名称	R	描述	复位值
15:0	QCTMRLAT	R	捕获定时器锁存（Capture Timer latch） 在两个单元超时事件中，可以将 QEP 捕获定时器值锁存到该寄存器，并读取 QEP 位置计数器。	0h

35.12.24 QEP 捕获周期锁存寄存器（QCPRDLAT）

偏移地址：0x40

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	QCPRDLAT	R	捕获周期锁存（Capture Period latch） 在两个单元超时事件中，可以将 QEP 捕获周期值锁存到该寄存器，并读取 QEP 位置计数器。	0h

35.12.25 QEP 修订编号寄存器（REV）

偏移地址：0x60

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	MAJOR	R	主要修订编号（Major Revision number） 该位域指定 QEP IP 的主要修订编号。	1h
5:3	MINOR	R	次要修订编号（Minor Revision number） 该位域指定 QEP IP 的次要修订编号。	0h
31:6	保留			0h

35.12.26 选通源选择寄存器（QEPSTROBESEL）

偏移地址：0x64

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
1:0	STROBESEL	R/W	选通源选择（Strobe source select） 0x：在极性多路复用器后，QEP 选通 10：用 ADCSOCA 进行极性多路复用器逻辑“或”运算后，QEP 选通 11：用 ADCSOCB 进行极性多路复用器逻辑“或”运算后，QEP 选通	0h
31:2	保留			0h

35.12.27 QMA 控制寄存器（QMACTRL）

偏移地址：0x68

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	MODE	R/W	为 QMA 模式选择模式（ Select Mode for QMA mode） 000：QMA 模块被绕过 001：操作模式 1 010：操作模式 2 其他：保留（QMA 模块被绕过）	0h
31:3	保留			0h

36 串行外设接口（SPI）

36.1 术语全称、缩写描述

表格 182 术语全称、缩写描述

中文全称	英文全称	英文缩写
最高有效位	Most Significant Bit	MSB
最低有效位	Least Significant Bit	LSB
从机输入主机输出	Slave In Master Out	SIMO
从机输出主机输入	Slave Out Master In	SOMI
主机输出主机输入	Master Out Master In	MOMI
从机输入从机输出	Slave In Slave Out	SISO
低速外设时钟	Low-Speed Peripheral Clock	LPCLK

36.2 简介

SPI 是高速同步串行输入输出端口，SPI 接口用于 MCU 与外设或其他控制器之间的通信，包括通过使用 ADC 转换器，移位寄存器和显示驱动等对外设或外部 I/O 进行扩展。可以工作于主机或者从机模式，支持多设备通信，有 125 种可编程的传输波特率，传输的字符长度可设定为 1-16 位。为了减少占用 CPU，支持 16 级发送/接收 FIFO 功能。

36.3 主要特征

（1） 时钟的极性和相位可编程，支持四种时钟方案
- 无延迟上升沿
- 有延迟上升沿
- 无延迟下降沿
- 有延迟下降沿

（2） 125 种可编程传输波特率

（3） 支持 1-16 位传输数据

（4） 支持主/从模式

（5） 支持 FIFO 功能，16 个可编程中断级

（6） 支持 DMA 功能

（7） 支持传输延迟控制

（8） SPI 高速模式

（9） SPI 3 线模式

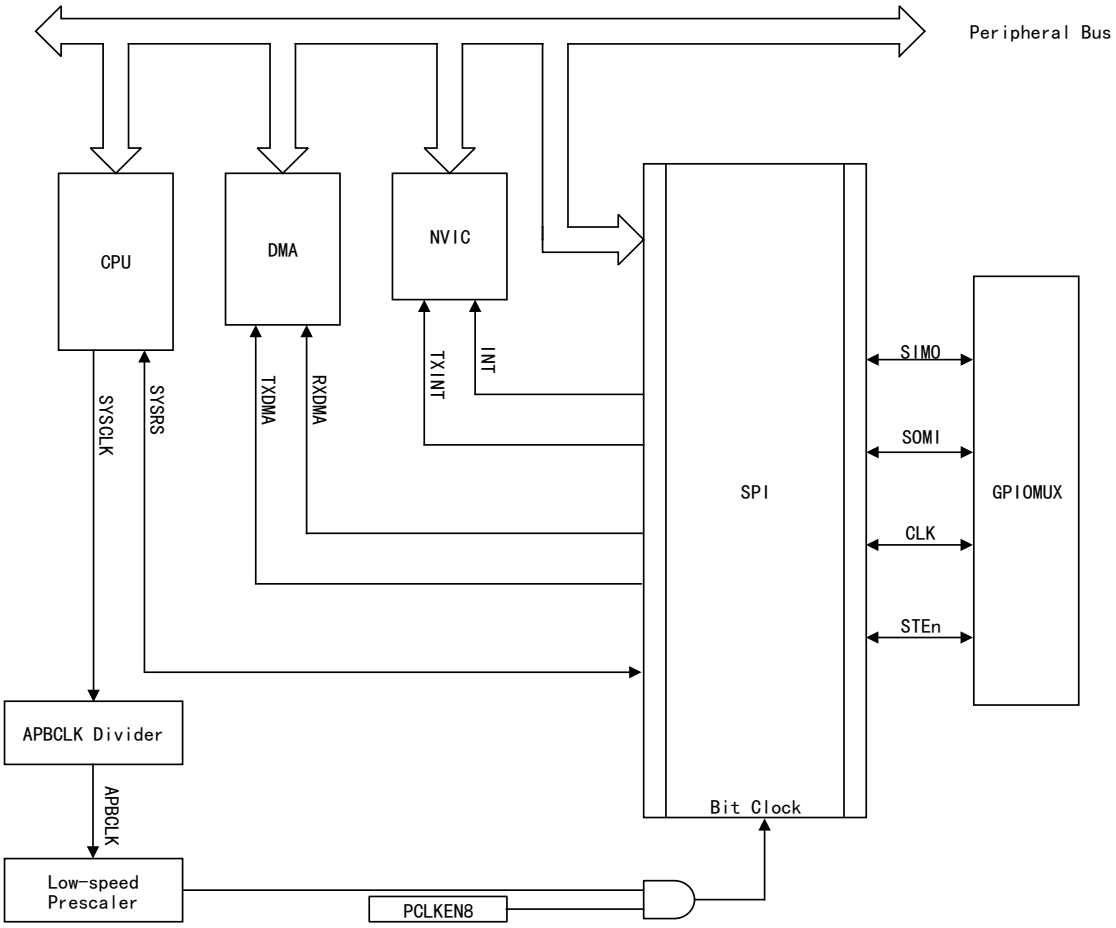
（10） 支持数字音频传输功能

（11） 同步收发操作（发送功能可以通过软件禁用）

（12） 通过中断驱动或轮询算法完成发送和接收操作

36.4 结构框图

图 188 SPI 的 CPU 接口结构框图



36.5 功能描述

36.5.1 SPI 信号线描述

表格 183 SPI 信号线描述

信号类别	引脚名称	描述
外部信号	SOMI	从机：输出引脚，发送数据 主机：输入引脚，接收数据 数据方向：从设备到主设备
	SIMO	从机：输入引脚，接收数据 主机：输出引脚，发送数据 数据方向：主设备到从设备

信号类别	引脚名称	描述
	CLK	主机：SPI 时钟输出 从机：SPI 时钟输入
	STEn	使能从机发送
中断信号	INT/RXINT	FIFO 模式下，接收数据时产生的中断 非 FIFO 模式下，接收/发送数据时产生的中断（称为 INT）。
	TXINT	FIFO 模式下，发送数据时产生的中断
DMA 触发信号	RXDMA	DMA 接收请求信号
	TXDMA	DMA 发送请求信号
控制信号	LPCLK	SPI 时钟速率

STEn 信号注意事项

当 SPI 从机丢失与主机的同步时，通过翻转 SPIRDY 的值，可以复位内部的位计数器和 SPI 中的各个状态标志位。位计数器复位后，下一个时钟信号传输的数据将被 SPI 确认为新数据的第一位。SPIRDY 复位的信息，详情请参见寄存器章节配置寄存器（SPI_CFG）。当 SPI 为从模式时，STEn 信号可以门控杂散时钟和数据脉冲。STEn 高电平时，禁止从机接收数据，以防止 SPI 从机丢失与主机的同步。因此，不建议经常激活 STEn。

模拟 STEn

通常情况下，一个 SPI 主机可以通过多个 STEn 引脚来连接多个 SPI 从机。但在本设备中，即使硬件上没有多个 STEn 信号来连接多个 SPI 从机，系统仍可以通过使用软件模拟多个 STEn 信号，方法如下：

使用 GPIO 代替 STEn 引脚。此时，SPI 必须配置为主机模式。在此配置下，每个 SPI 从机都通过一个专用的 GPIO 引脚来控制。这些 GPIO 引脚需要被配置为输出模式，而不使用 GPIO 复用选择 STEn。数据发送前，将目标 SPI 从机对应的 GPIO 引脚设置为激活状态，数据传输完成后，立即将该 GPIO 引脚设置为非激活状态。此操作会在多个共享同一主机 CLK，SIMO，SOMI 信号线的从机上反复发生。

36.5.2 引脚配置

通过配置 GPIO 章节的复用寄存器，选择对应的引脚将 SPI 与设备连接。必须按照 GPIO 章节中正确的步骤配置 GPIO 复用寄存器，即先配置 GPxGMUX 位段（此时对应的 GPxMUX 位段为默认值 0），再向 GPxMUX 位段写入所需值。

GPIO 寄存器中配置了一部分独立于 SPI 外设的 IO 功能。在 GPxPUD 寄存器中可以配置内部上拉电阻的状态。对于输入信号，可以通过配置对应的 GPxQSEL1/2[GPIOn]=11，将 GPIO 引脚配置为异步输入。更多 GPIO 复用的相关信息，详情请参见 GPIO 章节。

高速模式引脚配置

所有的 GPIO 都可复用为 SPI 高速模式。通过设置 SPI_CFG[HSEN]=1，以使能 SPI 高速模式。

注意：使用 SPI 高速模式时需确保引脚上电容性负载在数据手册中规定的值之内。当 SPI 工作在非高速模式或引脚上电容性负载超出数据手册中的值时，应将 SPI_CFG[HSEN]置 0。

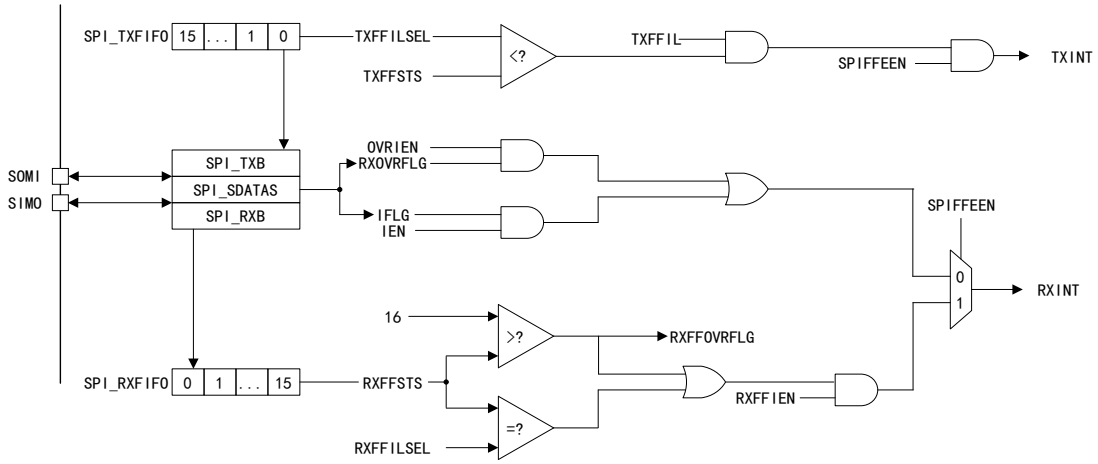
36.5.3 SPI 中断

SPI 中有两条中断线：INT/RXINT 和 TXINT。

（1） 在 FIFO 模式中，可以产生 RXINT 和 TXINT 中断；

（2） 在非 FIFO 模式中，所有可用中断一起路由以生成单个 INT 中断。

图 189 SPI 中断逻辑



表格 184 SPI 中断请求

模式	中断线	中断事件	中断标志	使能控制位
非 FIFO 模式（SPIFFEEN=0）	RXINT (INT)	发送缓冲区为空	IFLG	IEN
		接收缓冲区溢出	RXOVRFLG	OVRIEN
		数据接收	IFLG	IEN
FIFO 模式（SPIFFEEN=1）	RXINT	发送缓冲区为空	TXFFILSEL	TXFFIEN
	TXINT	FIFO 接收	RXFFILSEL	RXFFIEN

注意：

（1） 在非 FIFO 模式下，RXINT 和 INT 产生的中断统称为 INT。

36.5.3.1 INT/RXINT 中断

FIFO 模式

在 FIFO 模式下，SPI 通过判断接收 FIFO 的状态（RXFFSTS）和选择接收 FIFO 中断级（RXFFILSEL）的关系以产生 CPU 中断。当 RXFFSTS≥RXFFILSEL 时，置位接收 FIFO 中断标志（RXFFIFLG）。此时，若接收 FIFO 中断使能（RXFFIEN=1），则会在 NVIC 模块中触发 RXINT 中断。

非 FIFO 模式

在非 FIFO 模式下，所产生的中断统称为 INT；启用了 FIFO 中断增强功能后，所产生的中断称为 RXINT。在中断控制器（NVIC）中，这些中断共享相同的中断向量。

以下两个条件都可触发中断，并共享 INT 向量：

（1） 传输完成：传输完成表示 SPI 传输中，发送/接收的最后一位已完成，并且已为下一次的调用做好准备。此时中断标志位置位（IFLG=1），接收到的字符放入接收缓冲区（SPI_RXB）中。若使能了接收 FIFO 中断（RXFFIEN=1），IFLG 会在 INT 向量上产生一个中断。

（2） 接收溢出：接收溢出表示从缓冲区读取前一个字符前，已完成发送/接收操作，此时接收溢出标志位置位（RXOVRFLG）。若使能了溢出中断（OVRIEN=1），并且已清除之前的 RXOVRFLG 位，RXOVRFLG 会在 INT 向量上产生一个中断。

36.5.3.2 TXINT 中断

FIFO 模式

在 FIFO 模式下，产生 TXINT 中断的方式与产生 RXINT 中断的方式类似。SPI 通过判断发送 FIFO 的状态（TXFFSTS）和选择发送 FIFO 中断级（TXFFILSEL）的关系以产生 TXINT 中断。当 RXFFSTS≤RXFFILSEL 时，置位发送 FIFO 中断标志（TXFFIFLG）。此时，若发送 FIFO 中断使能（TXFFIEN=1），则会在 NVIC 模块中触发 TXINT 中断。

非 FIFO 模式

TXINT 中断不可用于非 FIFO 模式。

36.5.4 DMA 功能

36.5.4.1 DMA 事件

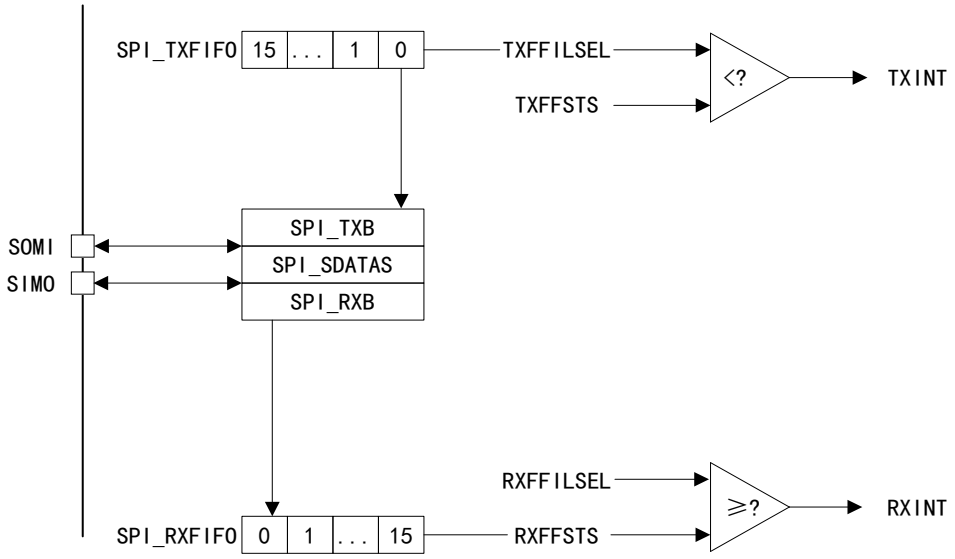
SPI 数据寄存器可以被 CPU 和 DMA 经由内部外设总线访问，并且只能通过 16 位寄存器读取/写入的方式访问。只有在使能了 FIFO 增强功能的情况下才可以生成 DMA 触发。每个 SPI 模块可以生成 SPITXDMA 和 SPIRXDMA 两个中断事件，根据以下配置以激活对应的 DMA 事件：

（1） SPITXDMA 事件：在 SPI_TXFIFO 寄存器中，当 TXFFSTS＜TXFFILSEL 时，激活 SPITXDMA。

（2） SPIRXDMA 事件：在 SPI_RXFIFO 寄存器中，当 RXFFSTS≥RXFFILSEL 时，激活 SPITXDMA。

SPI 触发 DMA 事件如下图所示：

图 190 SPI 触发 DMA 事件



36.5.4.2 **DMA 传输**

发送数据

在配置具有 DMA 功能的 TX FIFO 用于 SPI 通信时，为了避免 DMA 向一个已满的 FIFO 中继续写入数据而导致数据丢失，应确保 DMA 突发大小（BURST_SIZE[BURSTSIZE]）不超过（16- SPI_TXFIFO[TXFFILSEL]）。

数据发送步骤如下：

- 计算待发送的总数或待发送字的数量，用 WORDNUM 表示待发送数据的总字数；
- 在 SPI_TXFIFO[TXFFILSEL]中选择发送 FIFO 中断级；
- 在 TRANSFER_SIZE[TRANSFERSIZE]计算 DMA 传输的大小；
- 在 BURST_SIZE[BURSTSIZE]中计算 DMA 突发大小；
- 根据计算出的传输大小和突发大小，配置 DMA 以符合数据传输要求；
- 设置 SPI 和 FIFO：按照计算出的值调整 SPI 的相关参数；

注意：应避免将 SPI_TXFIFO[TXFFILSEL]配置为 0x00 或 0x10，以保证 DMA 配置的功能可以正常使用。

例如，要使用 DMA 发送一个大小为 128 字的数据，各参数配置如下表所示：

表格 185 128 字 DMA 数据发送示例

参数	计算过程	计算结果
WORDNUM	-	128
TXFFILSEL	-	8
TXSIZE	=(WORDNUM/TXFFILSEL)-1=(128/8)–1=15	15（16 次传输）
BURSTSIZE	(16- TXFFILSEL)-1=16-8–1=7	7(每 burst 8 个字)；

注意：应避免将 SPI_TXFIFO[TXFFILSEL]配置为 0x00 或 0x10，以保证 DMA 配置的功能可以正常使用。

接收数据

当配置具有 DMA 功能的 RX FIFO 用于 SPI 通信时，为了避免 DMA 从空的 FIFO 读取数据，应确保 DMA 突发大小（BURST_SIZE[BURSTSIZE]）不超过 SPI_RXFIFO[RXFFILSEL]。应配置 BURST_SIZE[BURSTSIZE]）=SPI_RXFIFO[RXFFILSEL]，或 SPI 传输总数的整除数，以保证来自 RX FIFO 的所有数据可以被 DMA 正确接收。

数据接收过程如下：

（1） 计算待接收的字的数量，用 WORDNUM 表示待接收的总字数；

- 在 SPI_RXFIFO[RXFFILSEL]中选择接收 FIFO 中断级；
- 在 TRANSFER_SIZE[TRANSFERSIZE]计算 DMA 传输的大小；
- 在 BURST_SIZE[BURSTSIZE]中计算 DMA 突发大小；
- 根据计算出的传输大小和突发大小，配置 DMA 以符合数据传输要求；
- 设置 SPI 和 FIFO：按照计算出的值调整 SPI 的相关参数；

注意：为了保证 DMA 的配置合理，避免将 SPI_TXFIFO[TXFFILSEL]配置为 0x00。

表格 186 200 字 DMA 数据接收示例

参数	计算过程	计算结果
WORDNUM	-	200
RXFFILSEL	-	4
TXSIZE	=(WORDNUM/RXFFILSEL)-1=(200/4)–1=49	49（50 次传输）
BURSTSIZE	RXFFILSEL-1=4–1=3	3(每 burst 4 个字);

36.5.5 SPI 模式

36.5.5.1 SPI 主机/从机连接

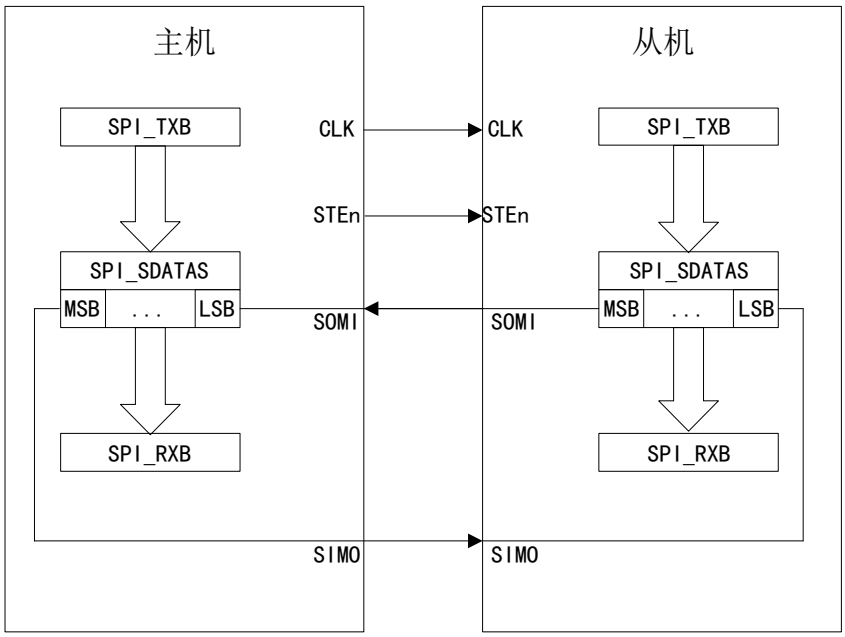
主机通过 CLK 信号传输数据。对于 SPI 主机和从机，当 SPI_CTRL[PHASEL]=0 时，在 CLK 时钟的一个边沿（上升沿/下降沿）从移位寄存器中移出数据，并在相对的另一个边沿（下降沿/上升沿）将数据锁存到移位寄存器中。当 SPI_CTRL[PHASEL]=1 时，数据的发送和接收过程发生在 CLK 信号转换的前半个周期。因此，SPI 会主机和从机同时发送和接收数据，需要通过应用软件判断发送的数据是否有效，数据传输方式一共有以下三种可能：

表格 187 数据传输方式

方式	主机	从机
1	发送有效数据	发送无效数据
2	发送有效数据	发送有效数据
3	发送无效数据	发送有效数据

由于 SPI 主机控制 CLK 信号，因此可以在任何时候启动数据传输，但是主机以什么方式检测从机是否准备好发送数据是由软件决定的。SPI 可以通过 SPI_CTRL[MSCFG]选择运行模式，即通过此位选择工作模式和 CLK 信号的源。SPI 主机/从机连接如下图所示：

图 191 SPI 主机/从机连接



36.5.5.2 SPI 主模式

配置 SPI_CTRL[MSCFG]=1，选择 SPI 运行模式为主模式。在主模式中，数据通过 SIMO 引脚输出，SOMI 引脚接收并锁存数据。SPI 通过 CLK 引脚为串行通信网络提供串行时钟。

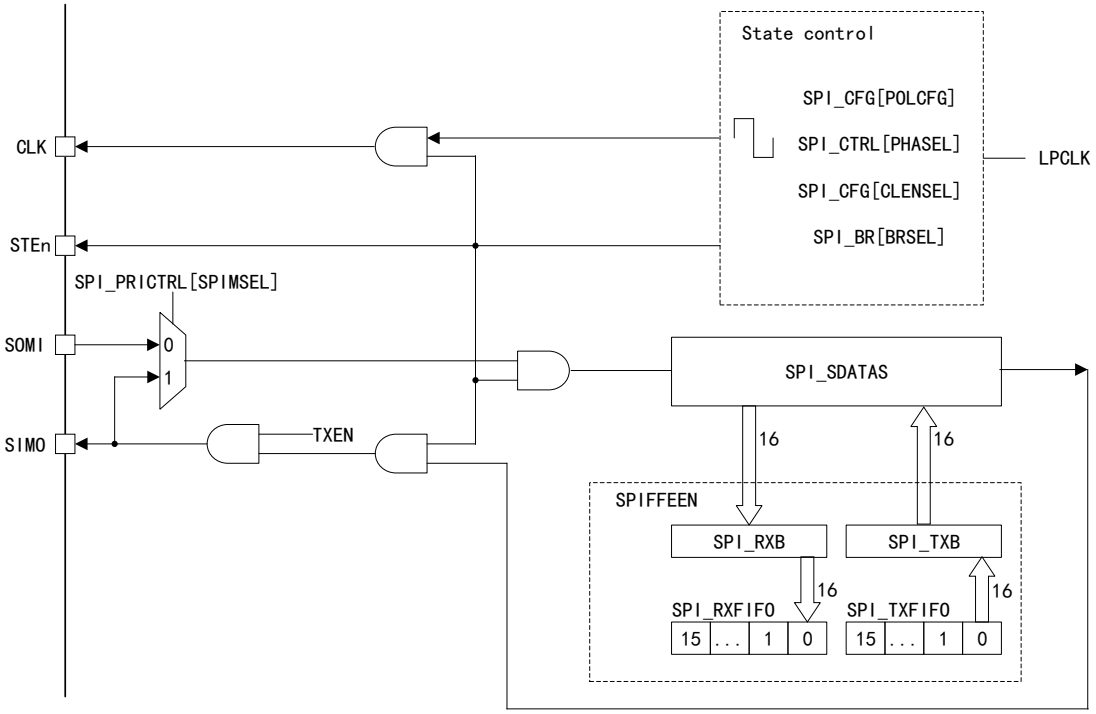
比特传输速率在波特率寄存器（SPI_BR）中配置，一共有 125 种数据传输速率可供选择。在串行数据移位寄存器（SPI_SDATAS）和发送缓冲区寄存器（SPI_TXB）中写入数据后，从最高有效位（MSB）开始依次从 SIMO 引脚上进行传输。在数据发送的同时，SOMI 引脚上接收的数据会按顺序填充到串行数据移位寄存器（SPI_SDATAS）最低有效位（LSB）中。在预定序列的数据传输完毕后，所接收到的数据会被转存到接收缓冲区寄存器（SPI_RXB）寄存器。在 SPI_RXB 中，这些数据按照右对齐的格式进行存储，以便 CPU 进行读取操作。

当设定数量的数据移入 SPI_SDATAS 中时，SPI_SDATAS 中的数据会移入 SPI_RXB 中，同时置位中断标志（IFLG=1）。此时若 SPI_TXB 中存在有效数据（即发送缓冲区满标志 TXBFFLG=1），那么该数据被移入 SPI_SDATAS 并发送。否则，当所有的数据位移出 SPI_SDATAS 后 CLK 停止。当使能 SPI 中断（即 IEN=1）时，产生 SPI 中断。

STEn 在典型应用中作为 SPI 从机片选使能引脚使用，在数据发送到主机前被主机驱动为低电平，在传输完成后驱动为高电平。

主模式下 SPI 的结构框图如下图所示：

图 192 SPI 主模式



36.5.5.3 SPI 从模式

配置 SPI_CTRL[MSCFG]=0，选择 SPI 运行模式为从模式。在从模式中，数据通过 SOMI 引脚输出和 SIMO 引脚输入。外部主机提供串行移位时钟，并通过 CLK 引脚输入。此外，串行移位时钟定义了传输速率，CLK 的输入频率不得超过 $f_{LPCLK}/4$。

从模式中，当检测到来自主机的 CLK 时钟的适当边缘跳变时，发送写入 SPI_SDATAS 和 SPI_TXB 的数据，根据 SPI_SDATAS 寄存器内的字符移出情况：

- SPI_SDATAS 内当前字符的所有数据位都已移出（即数据已全部发送），写入到 SPI_TXB 寄存器内的字符立即被复制到 SPI_SDATAS 寄存器中；
- SPI_SDATAS 内为空，即 SPI_SDATAS 内没有事先被复制的字符，写入到 SPI_TXB 寄存器内的字符立即被复制到 SPI_SDATAS 寄存器中；
- SPI_SDATAS 内不为空，即 SPI_SDATAS 内有事先被复制的字符，那么任何写入 SPI_TXB 的数据都不会被复制到 SPI_SDATAS，直到 SPI_SDATAS 中的当前字符被移出为止

当接收数据时，SPI 模块会等待主机的 CLK 时钟信号，接着将 SIMO 引脚上接收的数据移入 SPI_SDATAS 中。

当数据需要由从机同步发送时，若此时事先没有将数据装载到 SPI_SDATAS 中，那么必须在 CLK 信号开始之前将要发送的字符写入 SPI_TXB 中。

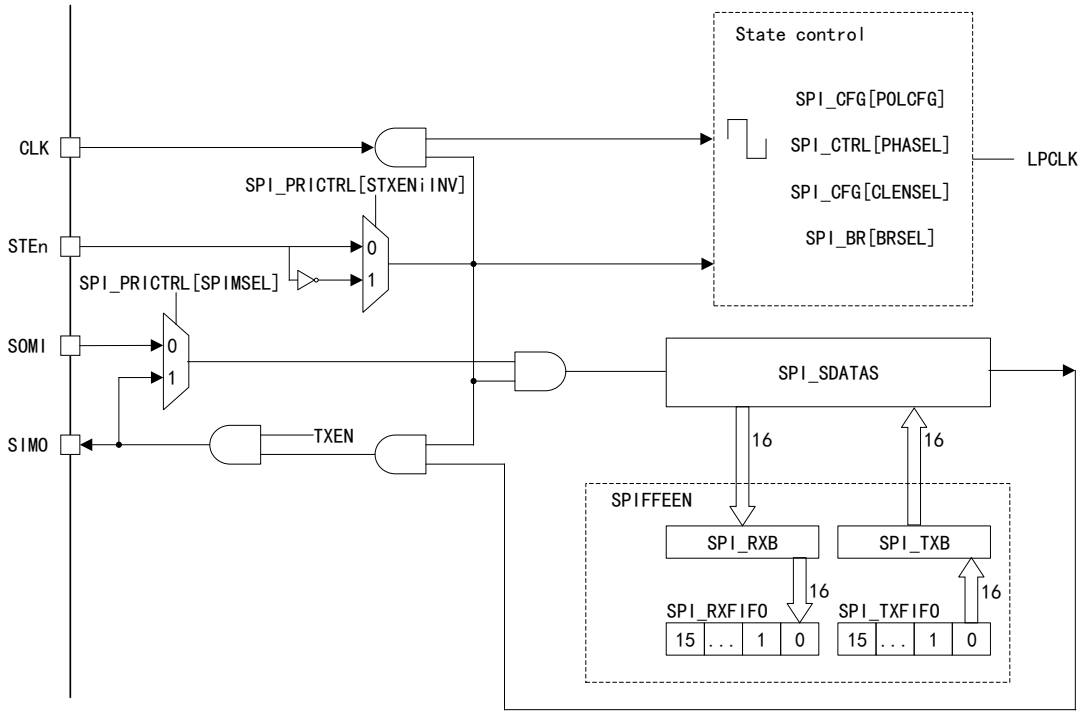
TXEN 位用于控制是否可以发送数据。当 TXEN 位清零后，通过将 SOMI 引脚置

于高阻态，禁用数据发送功能。在数据传输时清除 TXEN 位，正在进行传输的数据仍然会被完整传输。为了确保 SPI 模块可以正确接收到输入的数据，数据传输完成前，SPI 模块不会将 SOMI 强制设为高阻态，直到当前的数据传输完成后，才将 SOMI 强制设为高阻态。通过使用 TXEN 位可以控制哪个设备可以发送数据，可以在 SPI 网络中连接多个从设备，但在任何给定时刻只有一个设备被允许驱动 SOMI 引脚。

STEn 引脚是从机选择引脚。当 STEn 为低电平时，使能从机 SPI 将数据传输到串行数据线上；当 STEn 为高电平时，从机 SPI 串行移位寄存器（SPI_SDATAS）停止，输出引脚进入高阻态。尽管同时刻依然只能选择一个从设备进行通讯，但这让同一总线上连接多个设备成为可能。

从模式下 SPI 的结构框图如下图所示：

图 193 SPI 从模式



36.5.6 数据格式

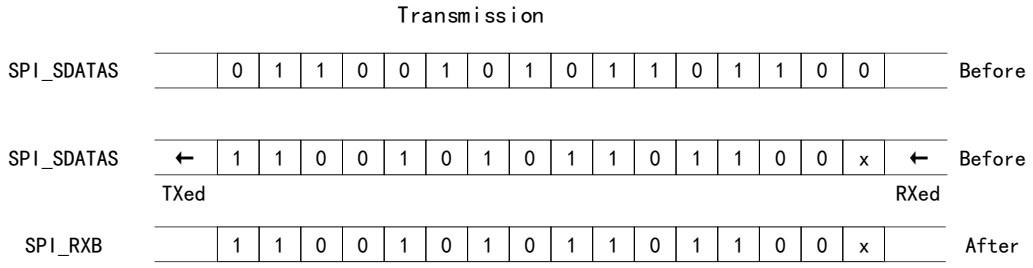
配置寄存器（SPI_CFG）中 CLENSEL 位段选择字符的长度（1 到 16 位）。CLENSEL 位段让状态控制逻辑单元计算传输的位数，以确定一个完整的字符被处理的时间。

对于小于 16 位的字符，数据必须以左对齐的方式写入 SPI_SDATAS 和 SPI_TXB，以右对齐的方式从 SPI_RXB 读出。此外，在 SPI_RXB 中包含最新接收的字符（右对齐），以及先前传输中剩余的移位到左边的字符。

SPI_RXB 中的位传输如下图所示。

假设传输字符长度为 1 位，SPI_SDATAS 中的当前值为 0x656C。

图 194 SPI_RXB 位传输示例



假设在主模式下，当 SOMI 信号为高电平时，x=1；当 SOMI 信号为低电平时，x=0。

36.5.7 传输波特率

SPI 有四种时钟方案可供选择，并支持 125 种不同的波特率。

（1） 在主模式中，SPI 产生 SPI 时钟，并通过 CLK 引脚输出，时钟频率不能大于 LPCLK/4；

（2） 在从模式中，SPI 时钟由外部源产生，并通过 CLK 引脚输入，时钟频率不能大于 LPCLK/4；

注意：配置波特率时，不能超过 GPIO 最大额定切换频率。有关 GPIO 最大额定切换频率，详情请参见数据手册。

选择传输波特率

- 当 SPI_BR=0/1/2：

$$\text{SPI 波特率} = \frac{\text{LPCLK}}{4}$$

- 当 SPI_BR=3~127：

$$\text{SPI 波特率} = \frac{LPCLK}{(SPI_BR + 1)}$$

其中：

LPCLK：低速外设时钟频率

SPI_BR：主机 SPI 中波特率寄存器的值

必须已知系统时钟频率（LPCLK）和操作的波特率，以确定写入到 SPI_BR 寄存器中的值。

计算传输波特率

以 LPCLK=60MHz，SPI_BR=5，SPI 工作在非高速模式（HSEN=0）下为例：

$$\text{SPI 波特率} = \frac{LSPCLK}{(SPI_BR + 1)}$$

$$= \frac{60 \times 10^6}{5 + 1}$$

$$= 10.0\text{Mbps}$$

36.5.8 时钟信号的相位和极性

时钟极性由 SPI_CFG[POLCFG]位控制，时钟相位 SPI_CTRL[PHASEL]位控制。

时钟极性是指 CLK 信号线的有效边沿。

- POLCFG=0 时，SPI 在时钟上升沿发送数据，下降沿接收数据。CLK 信号线在 SPI 空闲状态时为低电平。
- POLCFG=1 时，SPI 在时钟下降沿发送数据，上升沿接收数据。CLK 信号线在 SPI 空闲状态为高电平

时钟相位是指 CLK 信号是否延迟半个周期。

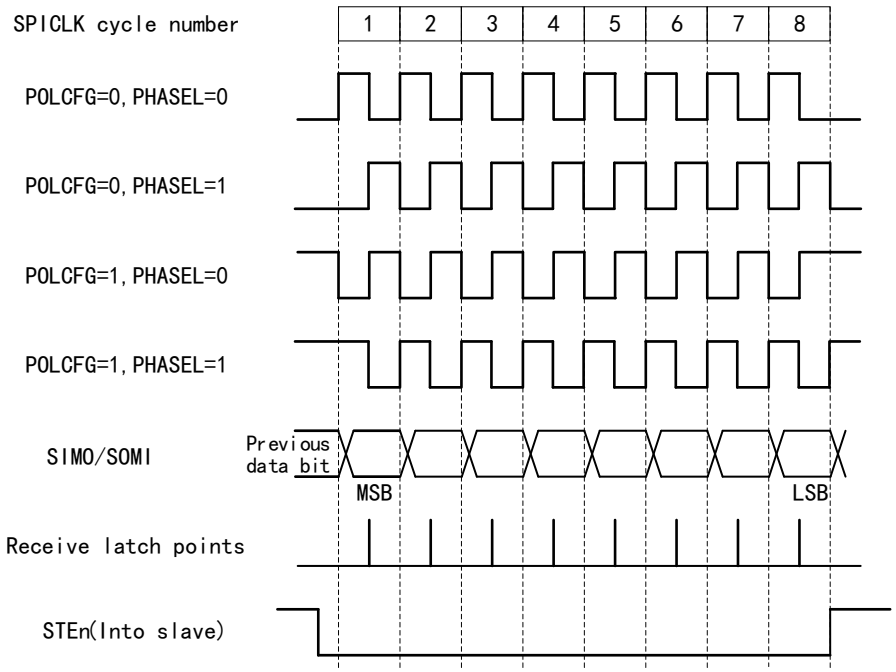
- PHASEL=0 时，无延迟。
- PHASEL=1 时，CLK 信号延迟半个周期。

根据时钟相位 POLCFG 和时钟极性 PHASEL 的不同状态，可以分成四种时钟方案。

表格 188 SPI 的四种时钟方案

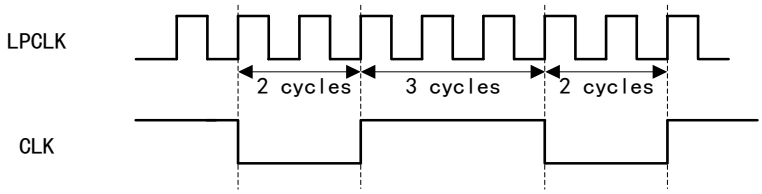
POLCFG	PHASEL	空闲时 CLK 时钟	延迟	时钟方案
0	0	低电平	无	无延迟上升沿。即 SPI 在时钟上升沿发送数据，下降沿接收数据。
0	1	低电平	有	有延迟上升沿。即 SPI 在时钟上升沿的前半周期发送数据，并在上升沿接收数据。
1	0	高电平	无	无延迟下降沿。即 SPI 在时钟下降沿发送数据，上升沿接收数据。
1	1	高电平	有	有延迟下降沿。即 SPI 在时钟下降沿的前半周期发送数据，并在下降沿接收数据。

图 195 SPI 信号选项



根据所需操作选择合适的波形，以确定时钟相位和极性的配置。

图 196 CLK-LPCLK 特性



如图所示，CLK 的对称性受到 SPI_BR 和 POLCFG 的影响：

（1） SPI_BR+1 为偶数值时，CLK 保留对称性

（2） SPI_BR+1>3 且为奇数值时，CLK 不对称，此时：

- 若 POLCFG=0，SPI 的低电平脉冲比高电平脉冲多一个 LPCLK 周期
- 若 POLCFG=1 时，SPI 的高电平脉冲比低电平脉冲长一个 LPCLK 周期。

36.5.9 FIFO 特性

（1） 使能 FIFO 模式：对 SPI_TXFIFO[SPIFFEEN]置 1 以使能 FIFO 模式，在 SPI 操作的任何时间，可以通过对 SPI_TXFIFO[SPIEN]置 1 以复位 FIFO 模式。使能 FIFO 模式后，所有 SPI 寄存器和 SPI FIFO 寄存器处于激活状态。

（2） SPI 复位：复位时，SPI 在标准模式下启动，FIFO 功能处于禁用状态，因此 FIFO 寄存器（SPI_TXFIFO，SPI_RXFIFO 和 SPI_FIFOCTRL）也处于禁用状态；

（3） FIFO 状态：SPI_TXFIFO[TXFFSTS]和 SPI_RXFIFO[RXFFSTS]位分别指示了发送和接收缓冲区中有效字的数量。当 SPI_TXFIFO[TXFFEN]和 SPI_RXFIFO[RXFFEN]置 1 时，将重置 FIFO 的指针；当 SPI_TXFIFO[TXFFEN]和 SPI_RXFIFO[RXFFEN]清零时，FIFO 开始复位操作。

（4） 缓冲区：在原有的发送缓冲区和接收缓冲区的基础上，分别增加了一个 16 字的 FIFO。标准 SPI 中，发送 FIFO 和移位寄存器使用大小为 1 个字的发送缓冲区寄存器（SPI_TXB）作为传输缓冲区。只有当移位寄存器中字符的最后一位被移出后，发送 FIFO 才会向 SPI_TXB 中装载数据。

（5） 中断：

- 在非 FIFO 模式中，使用 INT/RXINT 作为中断源。
- 在 FIFO 模式中，有两个中断源 TXINT 和 RXINT，分别用于发送和接收 FIFO。SPI FIFO 接收，接收错误和接收 FIFO 溢出共用 RXINT 中断。单个 INT 中断用于非 FIFO 模式中的发送和接收，在 FIFO 模式中被禁用，RXINT 作为 SPI FIFO 接收中断。有关中断的信息，详情请参见 SPI 中断章节。
- 中断级可编程。发送 FIFO 和接收 FIFO 都可以产生 DMA 触发中断和 CPU 中断。当 TXFFILSEL 位和 TXFFSTS 位匹配（即 TXFFSTS≤TXFFILSEL）时，将产生发送 FIFO 中断请求（TXINT），当 RXFFILSEL 位和 RXFFSTS 位匹配（即 RXFFSTS≥RXFFILSEL）时，将产生接收 FIFO 中断请求（RXINT）。以上操作为 SPI 的发送和接收提供了可编程的中断级。RXFFILSEL 默认配置为 0x11111，TXFFILSEL 默认配置为 0x00000。

（6） 传输延迟：可以调整从 FIFO 发送字到移位寄存器的速率。SPI_FFCTRL[FFTXDLYSEL]位定义了从 FIFO 发送缓冲区到发送移位寄存器之间的延迟。使用 SPI 时钟周期数定义此延迟。当没有延迟时，SPI 以连续模式发送数据，即一个接一个移出 FIFO 内的字。当存在 255 个时钟周期的延迟，SPI 以最大延迟模式发送数据。FIFO 中每个字相隔 255 个 CLK 周期的延迟从 FIFO 中移出。传输延迟有助于配合各种慢速 SPI 外设的无胶连接口，如 EEPROMs，ADC，DAC 等。

36.5.10 SPI 特殊模式

36.5.10.1 高速模式

通信速度

在 SPI 的高速模式中，能够以最高 LPCLK/4 的速率进行全双工数据传输。

在高速模式中，必须将 SPI 配置为单主机连接单从机模式，并且引脚上的负载在数据手册规定的值之内，以实现最大的全双工速度。有关最大额定速度的信息，详情请参见数据手册。

配置最大额定速度过程如下所示，假设在 100MHz 下运行：

（1） 设置 LPCLK=SYSCLK;

（2） 选择合适的 GPIO 复用引脚；

（3） 配置 SPI，其中：

- HSEN=1，使能高速模式;
- BRSEL=3，即 CLK=LPCLK/(BRSEL+1)=25;

高速模式中的其他配置过程与正常 SPI 模式配置过程相同，发送和接收数据，DMA 和中断方式也与正常 SPI 模式一致。有关 SPI 的配置过程，详情请参见 SPI 配置章节。

36.5.10.2 3 线模式

3 线/4 线模式差异

在正常模式中，SPI 通过四个引脚进行通信。SPI 的 3 线模式允许 SPI 通过三个引脚进行通信。

通过设置 SPIMSEL=1，使能 3 线模式。在主模式中，不再使用 SOMIx 引脚，SIMOx 作为双向 SPIMOMI 引脚。在从模式中，不再使用 SIMOx 引脚，SOMIx 作为双向 SPISISO 引脚。

3 线/4 线模式引脚功能差异如下表所示：

表格 189 3 线/4 线模式引脚功能差异

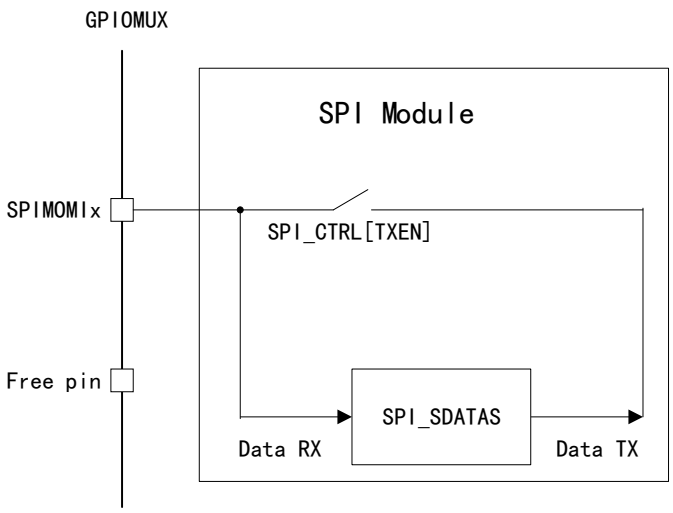
4 线模式	3 线模式（主模式）	3 线模式（从模式）
SOMIx	空闲	SISOx
SIMOx	MOMIx	空闲
CLKx	CLKx	CLKx
STEx	STEx	STEx

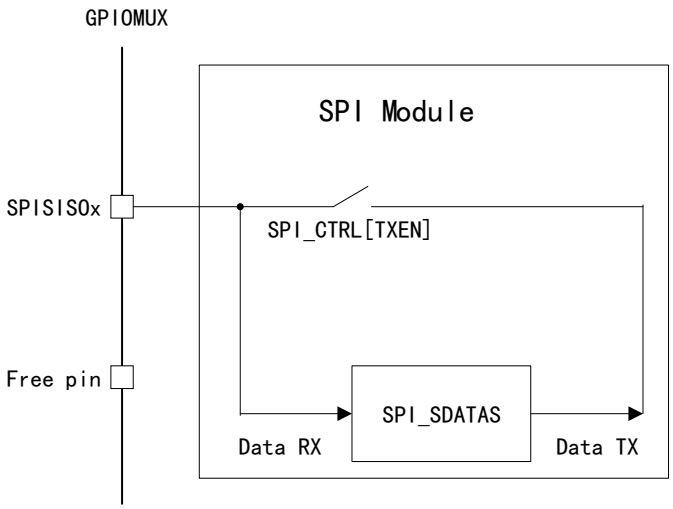
数据传输方式

在 3 线模式下，由于 SPI 内的接收和发送路径被连接在一起，因此 SPI 模块发送的任何数据也会被 SPI 自己接收。为了清除 SPI 数据寄存器中额外接收到的数据，软件层必须执行一次虚拟的读取。

3 线模式数据传输如下图所示：

图 197 SPI 3 线模式





将此 SPI_CTRL[TXEN]位置 1 以发送数据，在读取数据前清除此位以接收数据。在主模式下，为了启动读取操作，在从数据寄存器中读取数据之前，软件层必须在 TXEN 位被清零（没有任何数据被发送至 SPIMOMI）的情况下，将虚拟数据写入 SPI 数据寄存器（SPI_SDATAS 或 SPI_RXB）中。

当 SPI_CTRL[TXEN]置位或清零时，SPI 各模式引脚配置如下表所示：

表格 190 3 线 SPI 引脚配置

主/从模式	模式	SPI_PRICTRL[SPIMSEL]	SPI_CTRL[TXEN]	SIMO	SOMI
主模式	4 线模式	0	X	TX	RX
	3 线模式	1	0	RX	未使用
			1	TX/RX	
从模式	4 线模式	0	X	RX	TX
	3 线模式	1	0	未使用	RX
			1		TX/RX

3 线模式发送/接收示例

在 3 线模式中，除了一般的 SPI 初始化步骤外（详情请参见复位和初始化章节），还需要置位 SPI_PRICTRL[SPIMSEL]以配置 SPI 为 3 线模式。在 3 线模式下发送和接收数据时，主模式和从模式注意事项如下所示：

（1） 主模式发送：必须将 CLKx，STEx 和 SIMOx 引脚配置为 SPI 引脚，SOMIx 引脚可以配置为非 SPI 引脚。由于 SIMOx 和 SOMIx 在内部以 3 线模式连接，因此当主机发送数据时，会接收到自身发送的数据。根据以上特点，在每次发送数据时，必须清除 SPI_RXB 中接收到的垃圾数据。

（2） 主模式接收：必须将 CLKx，STEx 和 SIMOx 引脚配置为 SPI 引脚，SOMIx 引脚可以配置为非 SPI 引脚。此外，为了启动读取操作，主机必须对 TXEN 位被清零以关闭发送路径，然后向 SPI 数据寄存器中发送虚拟数据。与发送模式不同的是，由于 TXEN 位被清零，主机发送的虚拟数据不会传送到 SIMOx 引脚，也不会被主机本身接收。相反，主机会接收来自从机的数据。

（3） 从模式发送：必须将 CLKx，STEx 和 SOMIx 引脚配置为 SPI 引脚，SIMOx 引脚可以配置为非 SPI 引脚。由于 SOMIx 和 SIMOx 在内部以 3 线模式连接，因此当从机发送数据时，会接收到自身发送的数据。根据以上特点，在每次发送数据时，必须清除 SPI_RXB 中接收到的垃圾数据。

（4） 从模式接收：必须将 CLKx，STEx 和 SOMIx 引脚配置为 SPI 引脚，SIMOx 引脚可以配置为非 SPI 引脚。此外，为了启动读取操作，从机必须对 TXEN 位被清零以关闭发送路径，然后向 SPI 数据寄存器中发送虚拟数据。否则，从机将照常接收数据。

36.5.11 复位和初始化

系统复位时，强制 SPI 初始化为默认配置。默认配置如下：

（1） SPI 配置为从模式（MSCFG=0）；

（2） 禁用 SPI 中断功能；

（3） 禁用发送功能（TXEN=0）；

（4） 字符长度为 1 位（CLENSEL=0）；

（5） SPI_SDATAS 寄存器中的值为复位值 0x0000；

（6） 将数据锁存在 CLK 信号下降沿的输入端。

36.5.12 配置过程

在进行初始化更改前，必须将 SPIRDY 清零，以避免初始化期间或初始化后产生不可预见或不必要的错误。在初始化完成后再置位 SPIRDY。在 SPI 保持复位状态（即 SPIRDY=0）时，配置没有顺序要求。由于 SPI 寄存器可以通过单次 16 位的写操作来更新，因此除了需要特别注意 SPIRDY 的配置外，其他寄存器的配置顺序不需要特别注意。以下列表以逻辑顺序显示了 SPI 配置过程。

表格 191 SPI 配置过程

<table>
<tr><th>逻辑顺序</th><th colspan="3">配置过程</th><th>位域配置</th></tr>
<tr><td>1</td><td colspan="3">强制 SPI 为复位状态</td><td>SPIRDY=0</td></tr>
<tr><td rowspan="12">2</td><td rowspan="12">根据需要配置 SPI</td><td colspan="2">配置主从模式</td><td>MSCFG</td></tr>
<tr><td colspan="2">配置 CLK 极性</td><td>POLCFG</td></tr>
<tr><td colspan="2">选择 CLK 相位</td><td>PHASEL</td></tr>
<tr><td colspan="2">选择波特率</td><td>BRSEL</td></tr>
<tr><td colspan="2">（可选）使能高速模式</td><td>HSEN</td></tr>
<tr><td colspan="2">选择字符长度</td><td>CLENSEL</td></tr>
<tr><td colspan="2">清除 SPI 标志</td><td>IFLG, RXOVRFLG</td></tr>
<tr><td colspan="2">（可选）使能 STEn 反相</td><td>STEINV</td></tr>
<tr><td colspan="2">（可选）使能 3 线模式</td><td>SPIMSEL</td></tr>
<tr><td rowspan="4">（可选）使用 FIFO 增强功能</td><td>使能 SPI FIFO</td><td>SPIFFEEN</td></tr>
<tr><td>清除 FIFO 标志</td><td>TXFFICLR, RXFFOVRCLR, RXFFICLR</td></tr>
<tr><td>释放发送和接收 FIFO 复位</td><td>TXFFEN, RXFFEN</td></tr>
<tr><td>从复位中释放 SPI FIFO 通道</td><td>SPIEN</td></tr>
<tr><td rowspan="2">3</td><td rowspan="2">（可选）使用中断</td><td>FIFO 模式</td><td>设置发送和接收中断级，并使能中断</td><td>TXFFILSEL,RXFFILSEL,TXFFIEN,RXFFIEN</td></tr>
<tr><td>非 FIFO 模式</td><td>使能接收溢出和/或 SPI 中断</td><td>OVRIEN,IEN</td></tr>
<tr><td>4</td><td colspan="3">从复位状态释放 SPI</td><td>SPIRDY=1</td></tr>
</table>

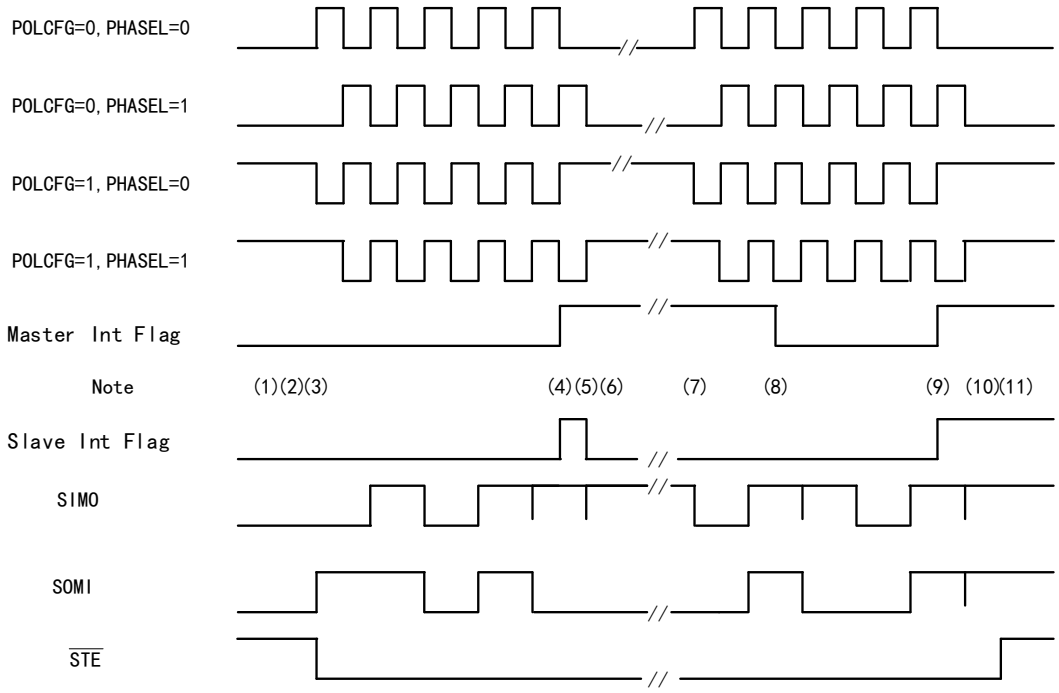
注意：在 SPI 进行通信时，禁止更改 SPI 配置。

36.5.13 SPI 传输时序

以字符长度为 5 位，CLK 信号对称的情况为例，SPI 主机和从机之间的数据传输如下图所示：

注意：下图仅适用于 8 位数据传输的 SPI，不适用于能够处理 16 位数据的 APM52 系列设备。此图仅供参考。

图 198 SPI 数据传输



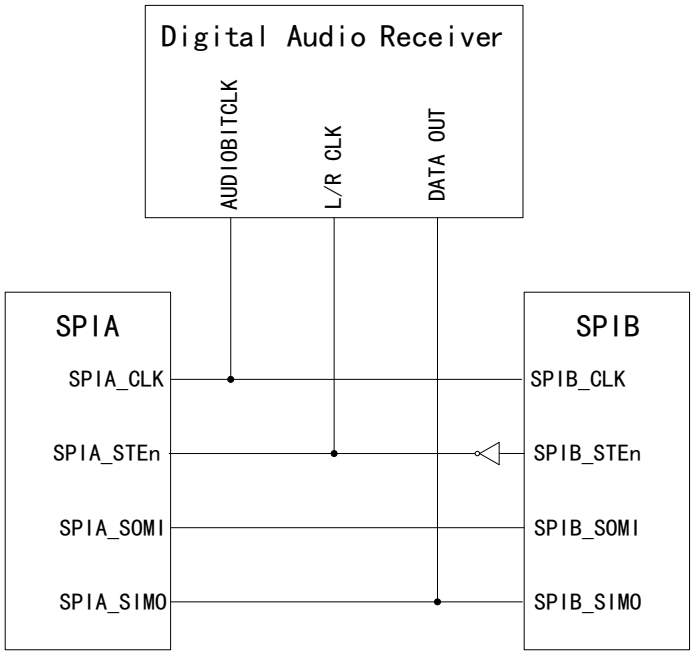
其中：

注释	描述
（1）	从机向 SPI_SDATAS 写入 0x0D0，并等待主机将数据移出
（2）	主机设置从机的 STEn 为低电平有效（激活）
（3）	主机向 SPI_SDATAS 写入 0x058，SPI_SDATAS 启动传输
（4）	完成第一个字节的传输，并设置中断标志
（5）	从机以右对齐的形式从 SPI_RXB 读取数据 0x0B
（6）	从机向 SPI_SDATAS 写入 0x04C，并等待主机将数据移出
（7）	主机向 SPI_SDATAS 写入 0x06C，SPI_SDATAS 启动传输
（8）	主机以右对齐的形式从 SPI_RXB 读取数据 0x01A
（9）	完成第二个字节的传输，并设置中断标志
（10）	从机读取数据 0x8D，主机读取数据 0x89。当未使用的位被软件屏蔽后，主机接收 0x09，从机接收 0x0D
（11）	主机清除从机的 STEn 信号，STEn 信号为高电平有效（失活）

36.5.14 数字音频传输功能

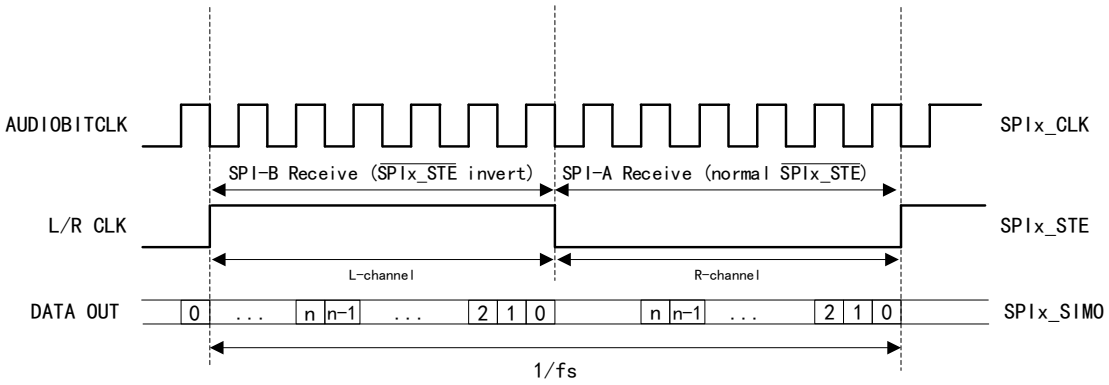
在有 2 个 SPI 外设的设备中，为了在从模式中可以接收左右声道的数字音频数据，对其中一个 SPI 外设使能 STEINV 位。右声道的数据在接收低电平 STEn 信号的时候存储，来自主机的左声道数据在接收相反的高电平 STEn 信号的时候存储。SPI 外设按照下图所示连接，以实现从数字音频接口接收音频数据：

图 199 数字音频接收



注意：此 STEINV 位仅用于 SPI 从模式，当 SPI 为主模式时，写入 STEINV 位不影响 STEn 引脚。

图 200 数字音频接口数据-标准右对齐格式



在配置 SPI 时钟极性 SPI_CFG[POLCFG]=0 和相位 SPI_CTRL[PHASEL]=1（即在时钟的上升沿锁存数据）的情况下，标准右对齐格式的数字音频接口数据如上图所示。

使用双 SPI 和 STEINV 位接收数字音频时，可接收的数字音频接口格式的数量受到 SPI 时序规范的限制。有关 SPI 时序规范，详情请参见数据手册。

36.6 寄存器组地址

表格 192 SPI 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
SpiaRegs	SPI_REGS	0x5000 1000	0x5000 13FF
SpibRegs	SPI_REGS	0x5010 0400	0x5010 07FF

36.7 寄存器地址映射

表格 193 SPI_REGS 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
SPI_CFG	配置寄存器	0x00	-
SPI_CTRL	控制寄存器	0x02	-
SPI_STS	状态寄存器	0x04	-
SPI_BR	波特率寄存器	0x08	-
SPI_EMUB	仿真缓冲区寄存器	0x0C	-
SPI_RXB	接收缓冲区寄存器	0x0E	-
SPI_TXB	发送缓冲区寄存器	0x10	-
SPI_SDATAS	串行数据移位寄存器	0x12	-
SPI_TXFIFO	发送 FIFO 寄存器 F	0x14	-
SPI_RXFIFO	接收 FIFO 寄存器	0x16	-
SPI_FFCTRL	FIFO 控制寄存器	0x18	-
SPI_PRICTRL	优先级控制寄存器	0x1E	-

36.8 寄存器功能描述

36.8.1 配置寄存器（SPI_CFG）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	CLENSEL	R/W	选择字符长度（Character Length select） 选择一个移位序列中单个字符移入或 CLENSEL[0]移出的位数。CLENSEL= Word length – 1 0000：1-bit word 0001：2-bit word … 1111：16-bit word	0h
4	LBEN	R/W	使能环回模式（Loopback Mode Enable） 环回模式仅在 SPI 主模式下有效。使能环回模式后，可以在设备测试期间进行模块验证。 0：禁用 1：使能，此时 SIMO 和 SOMI 是用于模块自检的内部连接	0h
5	HSEN	R/W	使能高速模式（High Speed Mode Enable） 使能高速模式时，应通过 GPIO 复用寄存器选择正确的 GPIO。 0：禁用 1：使能	0h

位/域	名称	R/W	描述	复位值
6	POLCFG	R/W	配置时钟极性（Clock Polarity Configure） 此位配置 CLK 信号的有效边沿。此位和 PHASEL 共同配置 CLK 引脚上的四种时钟方案。 0：SPI 在 CLK 上升沿发送数据，下降沿接收数据。CLK 信号线在 SPI 空闲状态时为低电平。PHASEL 位决定数据发送/接收边沿。 PHASEL=0，在 CLK 上升沿发送数据，在 CLK 的下降沿锁存接收的数据。 PHASEL=1，在 CLK 第一个上升沿的前半个周期以及后续的下降沿发送数据，在 CLK 的上升沿锁存接收的数据。 1：SPI 在 CLK 下降发送数据，上升沿接收数据。CLK 信号线在 SPI 空闲状态时为高电平。PHASEL 位决定数据发送/接收边沿。 PHASEL=0，在 CLK 下降沿发送数据，在 CLK 的上升沿锁存接收的数据。 PHASEL=1，在 CLK 第一个下降沿的前半个周期以及后续的上升沿发送数据，在 CLK 的下降沿锁存接收的数据	0h
7	SPIRDY	R/W	SPI 准备就绪（SPI ready） 当 SPIRDY=0，置位此位时写入发送器的字符不会移出，因此必须向串行数据移位寄存器写入新字符。置位后，CLK 将返回到非激活状态的一个周期中。 0：初始化 SPI 标志为复位条件，即清除 TXBFFLG，IFLG，RXOVRFLG 标志，将 CLK 立即驱动为低电平，而不考虑时钟的极性；将 STE 驱动为非激活状态。SPIRDY=0 时，保持 SPI 的配置不变。 1：SPI 准备就绪，SPI 已为发送/接收下一个字符做好准备 注意：应该在更改 SPI 配置前清除此位，并在恢复 SPI 操作前将此位置位。	0h
15:8	保留			0h

36.8.2 控制寄存器（SPI_CTRL）

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	IEN	R/W	使能 SPI 中断（SPI Interrupt Enable） 使能 SPI 发送/接收中断，中断标志（IFLG）位不受此位的影响。 0：禁用 1：使能	0h

位/域	名称	R/W	描述	复位值
1	TXEN	R/W	使能发送（Transmit Enable） 通过将串行输出数据高阻态以禁用数据传输（主模式/从模式）。当在传输期间禁用 TXEN 时，发送移位寄存器保持工作状态直到前一个字符被移除。当禁用此位时，SPI 仍然可以接收字符并更新对应的标志位。此位由系统复位清除。 0：禁用传输（在主模式中，若未配置 GPIO 引脚，SIMO 引脚将处于高阻抗；在从模式中，若未配置 GPIO 引脚，SOMI 引脚将处于高阻抗。） 1：使能传输（对于 4 线模式，需确保接收器的 STEn 输入引脚使能）	0h
2	MSCFG	R/W	配置主从模式（Master Slave mode configure） 0：从模式 1：主模式	0h
3	PHASEL	R/W	选择时钟相位（Clock Phase Select） 此位选择 CLK 信号的是否延迟半个周期。此位和 POLCFG 共同配置 CLK 引脚上的四种时钟方案。 0：无延迟。此时数据输入/输出具体取决于 POLCFG 1：CLK 信号延迟半个周期。此时数据输入/输出极性具体取决于 POLCFG。无论在何种 SPI 模式下，SPI（主机/从机）都会在写入 SPI_SDATAS 寄存器后以及 CLK 信号的第一个边沿前提供首位数据。	0h
4	OVRIEN	R/W	使能溢出中断（Overrun Interrupt Enable） 此位控制能否产生溢出中断。置位后，当硬件置位接收溢出标志（RXOVRFLG）位时产生中断。接收溢出标志（RXOVRFLG）和中断标志（IFLG）共享相同的中断向量 0：禁用 1：使能	0h
15:5	保留			0h

36.8.3 状态寄存器（SPI_STS）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	保留			0h
5	TXBFFLG	R	发送缓冲区满标志（Transmit Buffer Full Flag） 当 SPI 发送缓冲区写入字符时置位。当前一个字符移出 SPI 发送缓冲区并自动加载到 SPI_SDATAS 寄存器后自动清除此位。 0：发送缓冲区未满 1：发送缓冲区满	0h

位/域	名称	R/W	描述	复位值
6	IFLG	RC_R	中断标志（Interrupt Flag） 当 SPI 已完成发送/接收的最后一位并为下一次的操作做好准备时设置此位。同时，接收到的字符会放入接收缓冲区中。若 IEN=1，置位此位时会产生中断请求。通过系统复位，对 SPIRDY 位写 0 或读取接收缓冲区寄存器三种方式之一清除此位。 0：未接收或发送完整的字 1： SPI 已完成发送/接收的最后一位并为下一次的操作做好准备 注意：当启动了 FIFO 模式时禁用此位，接收到的字从 SPI 接收缓冲区复制到接收 FIFO 的内部过程将清除此位，类似功能在 FIFO 模式中通过 FIFO 状态或 FIFO 中断位实现。	0h
7	RXOVRFLG	RC_W1	接收溢出标志（Receiver Overrun Flag） 当接收/发送操作在前一个字符从缓冲区中读取之前完成时设置此位。通过系统复位，对此位写 1 或对 SPIRDY 位写 0 三种方式之一清除此位。若 OVRIEN=1，则 SPI 在第一次设置此位时仅产生一个中断请求。当 RXOVRFLG 已经置位，后续溢出时将不再产生其他中断请求。因此，必须在每次溢出发生后向此位写 1 来清除此位以允许产生新的溢出中断请求。否则，当中断服务例程未清除此位时，退出中断服务例程时不会立即重新输入另一个溢出中断请求。由于此位和 IFLG 位共享相同的中断向量，需在中断服务例程期间清除此位，在接收到下一个字节时可以排除中断源的可能性。 0：未发生接收溢出 1：最后一个接收到的字符被覆盖并丢失（在用户应用程序读取前一个字符之前，SPI 接收缓冲区被 SPI 模块重新写入）。	0h
15:8	保留			0h

36.8.4 波特率寄存器（SPI_BR）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
6:0	BRSEL	R/W	选择波特率（Baud Rate selcet） 当 SPI 作为从机时，SPI 通过 CLK 引脚从主机上接收时钟。此时，CLK 信号不受 BRSEL 位的影响。注意：从主机中接收到的时钟频率不得超过 SPI 从机 $f_{LPCLK}/4$。 当 SPI 作为主机时，此位域选择传输的比特传输速率。一共有 125 个数据传输速率可供选择，每一个传输速率都是 CPU 时钟 LPCLK 的函数。在 CLK 引脚的波特率时钟上输出 CLK 时钟，每个 CLK 周期移动一个数据位。 在主模式下，SPI 产生时钟并通过 CLK 引脚输出，传输的波特率计算公式详情请参见传输波特率章节。 000000：SPI 波特率=LPCLK/4 000001：SPI 波特率=LPCLK/4 000010：SPI 波特率=LPCLK/4 其他：SPI 波特率=LPCLK/（SPI_BR+1）	0h

位/域	名称	R/W	描述	复位值
15:7	保留			0h

36.8.5 仿真缓冲区寄存器（SPI_EMUB）

偏移地址：0x0C

复位类型：SYSRSn

建议在正常的仿真器运行模式中查看此寄存器。

位/域	名称	R/W	描述	复位值
15:0	EMUB	R	仿真缓冲区（Emulation Buffer） SPI_SDATAS 寄存器接收到完整的字符会移入 SPI_RXB 和 SPI_EMUB 寄存器中，因此可以在这两个寄存器中读取接收到的字符，移入的同时设置中断标志（IFLG）。SPI_EMUB 寄存器的功能与 SPI_RXB 寄存器几乎相同，但是读取 SPI_EMUB 寄存器中的内容时不会清除中断标志（IFLG）。实际上，SPI_EMUB 是一个虚拟地址，仿真器可以在不清除中断标志（IFLG）的情况下读取 SPI_RXB 寄存器中的内容。 此寄存器的作用是支持仿真。由于读取 SPI_RXB 寄存器的同时会清除中断标志（IFLG）。在正常进行仿真时，为了持续更新显示屏上的控制寄存器的内容将读取控制寄存器。通过读取此寄存器，仿真器可以正确更新显示屏的内容。即，通过读取此寄存器，允许仿真器更加准确地仿真 SPI 的真实环境。	0h

36.8.6 接收缓冲区寄存器（SPI_RXB）

偏移地址：0x0E

复位类型：SYSRSn

SPI_SDATAS 寄存器接收到完整的字符会移入此寄存器中，移入的同时设置中断标志（IFLG）。读取 SPI_RXB 寄存器的同时会清除中断标志（IFLG）。

当使能 FIFO 模式时，读取此寄存器还会使 SPI_RXFIFO[RXFFSTS]递减。

位/域	名称	R/W	描述	复位值
15:0	RXB	R	接收缓冲区（Received Buffer） 接收到的数据首先移入 SPI 的最高有效位，在此寄存器中以右对齐的方式存储数据。	0h

36.8.7 发送缓冲区寄存器（SPI_TXB）

偏移地址：0x10

复位类型：SYSRSn

下一个要发送的字符写入此寄存器中，同时设置 TXBFFLG 标志。如果当前的字符完成传输，将此寄存器的数据自动装载到 SPI_SDATAS 寄存器中，同时清除 TXBFFLG 标志。若当前没有传输处于活动状态，写入 SPI_TXB 中的数据将在不设置 TXBFFLG 标志的情况下传送到 SPI_SDATAS 寄存器。

在主模式中，若当前没有传输处于活动状态，那么写入此寄存器时启动传输的方式与写入 SPI_SDATAS 寄存器时启动传输的方式相同。

位/域	名称	R/W	描述	复位值
15:0	TXB	R/W	发送缓冲区（Transmit Buffer） 在此寄存器中以左对齐的方式写入数据。	0h

36.8.8 串行数据移位寄存器（SPI_SDATAS）

偏移地址：0x12

复位类型：SYSRSn

数据写入此寄存器后，在随后的 CLK 周期中，数据会从最高有效位（MSB）进行移位输出。对于每个移出的位（MSB），一个位会从最低有效位（LSB）移入寄存器。

位/域	名称	R/W	描述	复位值
15:0	SDATAS	R/W	串行数据移位（Serial Data Shift） 当 TXEN=1 时，串行输出引脚上输出的数据由此位提供； 当 SPI 在主模式时将启动数据传输。有关传输时钟的相位和极性信息，详情请参见 POLCFG 和 PHASEL 位。在主模式中，向此寄存器写入虚拟数据时会启动接收器序列。由于数据不是由硬件对齐，当传输的字符长度小于 16 位时，必须以左对齐的形式写入发送的数据，以右对齐的形式读取接收的数据。	0h

36.8.9 发送 FIFO 寄存器（SPI_TXFIFO）

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	TXFFILSEL	R/W	选择发送 FIFO 中断级（Transmit FIFO interrupt level Selcet） 当此位和 TXFFSTS 位匹配（即 TXFFSTS≤TXFFILSEL）时，将产生发送 FIFO 中断请求。 00000：发送缓冲区中没有剩余的字 00001：发送缓冲区中没有剩余的字或剩余 1 个字 00010：发送缓冲区中没有剩余的字或剩余 2 个以下的字 … 10000：发送缓冲区中没有剩余的字或剩余 16 个以下的字 其他：发送缓冲区中没有剩余的字或剩余 16 个以下的字	0h
5	TXFFIEN	R/W	使能发送 FIFO 中断（Transmit FIFO Interrupt Enable） 使能基于 TXFFILSEL 匹配的发送 FIFO 中断。 0：禁用 1：使能	0h
6	TXFFICLR	W	清除发送 FIFO 中断标志（Transmit FIFO Interrupt Flag Clear） 清除 TXFFIFLG 标志。 0：无影响，读为 0 1：清除 TXFFIFLG 标志	0h

位/域	名称	R/W	描述	复位值
7	TXFFIFLG	R	发送 FIFO 中断标志（Transmit FIFO interrupt flag） 0：未发生发送 FIFO 中断 1：发生发送 FIFO 中断	0h
12:8	TXFFSTS	R	发送 FIFO 的状态（ transmit FIFO status） 00000：发送缓冲区为空 00001：发送缓冲区中有 1 个字 00010：发送缓冲区中有 2 个字 … 10000：发送缓冲区中有 16 个字（最大值） 其他：发送缓冲区中有 16 个字（最大值）	0h
13	TXFFEN	R/W	使能发送 FIFO（transmit FIFO enable） 0：将 FIFO 指针复位为 0，并保持复位状态 1：从复位状态中释放发送 FIFO	1h
14	SPIFFEEN	R/W	使能 SPI FIFO（SPI FIFO Enhancements enable ） 0：禁用 1：使能	0h
15	SPIEN	R/W	使能 SPI（SPI Enable） 0：复位 SPI 发送和接收通道，并保持 SPI FIFO 寄存器的配置不变 1：恢复 SPI FIFO 的发送和接收功能，不影响 SPI 的寄存器。	1h

36.8.10 接收 FIFO 寄存器（SPI_RXFIFO）

偏移地址：0x16

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	RXFFILSEL	R/W	选择接收 FIFO 中断级（Receive FIFO interrupt level Selcet） 当此位和 RXFFSTS 位匹配（即 RXFFSTS≥RXFFILSEL）时，将产生接收 FIFO 中断请求。 00000：接收缓冲区中有 0 个字或 0 个以上字 00001：接收缓冲区中有 1 个字或 1 个以上字 00010：接收缓冲区中有 2 个字或 2 个以上字 … 10000：接收缓冲区中有 16 个字 其他：接收缓冲区中有 16 个字	1Fh
5	RXFFIEN	R/W	使能接收 FIFO 中断（Receive FIFO Interrupt Enable） 使能基于 RXFFILSEL 匹配的接收 FIFO 中断。 0：禁用 1：使能	0h

位/域	名称	R/W	描述	复位值
6	RXFFICLR	W	清除接收 FIFO 中断标志（Receive FIFO Interrupt Flag Clear） 清除 RXFFIFLG 标志。 0：无影响，读为 0 1：清除 RXFFIFLG 标志	0h
7	RXFFIFLG	R	接收 FIFO 中断标志（Receive FIFO interrupt flag） 0：未发生接收 FIFO 中断 1：发生接收 FIFO 中断	0h
12:8	RXFFSTS	R	接收 FIFO 的状态（Receive FIFO status） 00000：接收缓冲区为空 00001：接收缓冲区中有 1 个字 00010：接收缓冲区中有 2 个字 … 10000：接收缓冲区中有 16 个字（最大值） 其他：接收缓冲区中有 16 个字（最大值）	0h
13	RXFFEN	R/W	使能接收 FIFO（Receive FIFO enable） 0：将 FIFO 指针复位为 0，并保持复位状态 1：从复位状态中释放接收 FIFO，重新使能接收 FIFO	1h
14	RXFFOVRCLR	W	清除接收 FIFO 溢出标志（Receive FIFO Overflow Flag Clear） 清除 RXFFOVRFLG 标志。 0：无影响，读为 0 1：清除 RXFFOVRFLG 标志	0h
15	RXFFOVRFLG	R	接收 FIFO 溢出标志（Receive FIFO Overflow Flag） 0：未发生接收 FIFO 溢出 1：发生接收 FIFO 溢出，即 FIFO 中接收的字超过 16 个，且丢失接收的第一个字	0h

36.8.11 FIFO 控制寄存器（SPI_FFCTRL）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	FFTXDLYSEL	R/W	选择 FIFO 发送延迟（FIFO Transmit Delay Selcet） 此位域定义每次传输时从 FIFO 发送缓冲区到发送移位寄存器之间的延迟。使用 SPI 串行时钟周期数来定义此延迟，因此这个 8 位的寄存器定义的延迟范围为 0~255 个串行时钟周期。 在 FIFO 模式中，为了将传输之间的延迟传递到数据流中，位域移位寄存器和 FIFO 之间的发送缓冲区（SPI_TXB）应该在移位寄存器的最后一位移位完成后再填充。在 FIFO 模式中，不能将 SPI_TXB 当作一个额外的缓冲区级别。 0000 0000：在前一个字的传输完成后，FIFO 发送缓冲区的下一个字立即发送到 SPI_TXB 中 0000 0001：在前一个字的传输完成后，FIFO 发送缓冲区的下一个字在 1 个串行时钟周期后发送到 SPI_TXB 中 0000 0010：在前一个字的传输完成后，FIFO 发送缓冲区的下一个字在 2 个串行时钟周期后发送到 SPI_TXB 中 ….. 1111 1111：在前一个字的传输完成后，FIFO 发送缓冲区的下一个字在 255 个串行时钟周期后发送到 SPI_TXB 中	0h
15:8	保留			0h

36.8.12 优先级控制（SPI_PRICTRL）

偏移地址：0x1E

复位值：0x0000

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	SPIMSEL	R/W	选择 SPI 模式（SPI Mode Selcet） 在 3 线模式中，未使用的引脚将作为 GPIO 引脚。主模式下，SIMO 引脚作为 SPIMOMI（主机发送和主机接收）引脚，SOMI 提供给其他模块自由使用；从模式下，SOMI 引脚作为 SPISOSI（从机发送和主机接收）引脚，SIMO 提供给其他模块自由使用。 0：4 线模式（正常模式） 1：3 线模式	0h
1	STEINV	R/W	STE 反相（STEn Inversion） 此位仅用于 SPI 从模式，当 SPI 为主模式时写入此位无效。 在有 2 个 SPI 外设的设备中，为了接收左右声道的数字音频数据，需要对其中一个模块的 STE 信号进行反相。 0：STEn 信号低电平有效（正常） 1：STE 信号高电平有效（反相）	0h
3:2	保留			0h

位/域	名称	R/W	描述	复位值
4	EMUFREEEN	R/W	使能仿真自由运行（Emulation Free Run Enable） 此位配置当仿真挂起（例如调试器遇到断点）时发生的情况。在自由运行模式时，外设可以继续运行当前操作；在停止模式时，仿真挂起时立即停止或在当前操作（当前的发送/接收序列）完成后停止。 0：由 EMUSOFT 选择仿真模式 1：自由运行模式，外设继续运行当前操作	0h
5	EMUSOFT	R/W	仿真软件运行（Emulation Soft Run） 仅当 EMUFREEEN=0 时，此位有效。 传输开始的时间由所使用的波特率确定。在标准 SPI 模式中， 0：若 TSUSPEND 被断言，则在位流的中途停止传输。一旦 TSUSPEND 被取消断言，并且没有进行系统复位，数据缓冲区中待发送的剩余位将被移位。例如：如果 SPI_SDATAS 中已经移位了 8 位中的 3 位，则通信会在第三位处暂停。如果后续取消断言 TSUSPEND 并且没有进行 SPI 的复位，SPI 将从之前停止的位置（即第四位）开始传输，并从该点开始传输 8 位。 1：当传输开始前发生仿真挂起（即在第一个 CLK 脉冲之前），不会进行传输；当传输开始后发生仿真挂起时（即在第一个 CLK 脉冲之前），数据将被完整地移出 SPI_SDATAS。传输开始的时间由所使用的波特率确定。传输在数据从移位寄存器和缓冲区之间的移位完成后停止，即在标准的 SPI 模式中，当 SPI_TXB 和 SPI_SDATAS 都为空时停止传输；在 FIFO 模式中，当发送 FIFO 和 SPI_SDATAS 都为空时停止传输。	0h
15:6	保留			0h

37 通用异步收发器（UART）

37.1 术语全称、缩写描述

表格 194 术语全称、缩写描述

中文全称	英文全称	英文缩写
先进先出	first-in first-out	FIFO

37.2 简介

UART 是一个可以灵活地与外部设备进行全双工、半双工数据交换的串行通信设备，且同时满足外部设备对工业标准 NRZ 异步串行数据格式的要求。UART 还提供宽范围的波特率选择，且支持多处理器通信。

37.3 主要特征

（1） 全双工异步通信

（2） 单线半双工通信

（3） NRZ 标准格式

（4） 可编程的串口特性：
- 数据位：1 位-8 位
- 校验位：偶校验，奇校验
- 支持 1，2 个停止位

（5） 可以区分数据和地址的位

（6） 双缓冲接收和发送

（7） 独立的发送器和接收器使能位

（8） 独立的中断使能位

（9） 可通过基于中断或轮询算法的状态完成发送器和接收器操作

（10） 可切换 TX/RX 引脚

（11） 16 级发送/接收 FIFO

（12） 可编程的波特率发生器

（13） 波特率自动检测

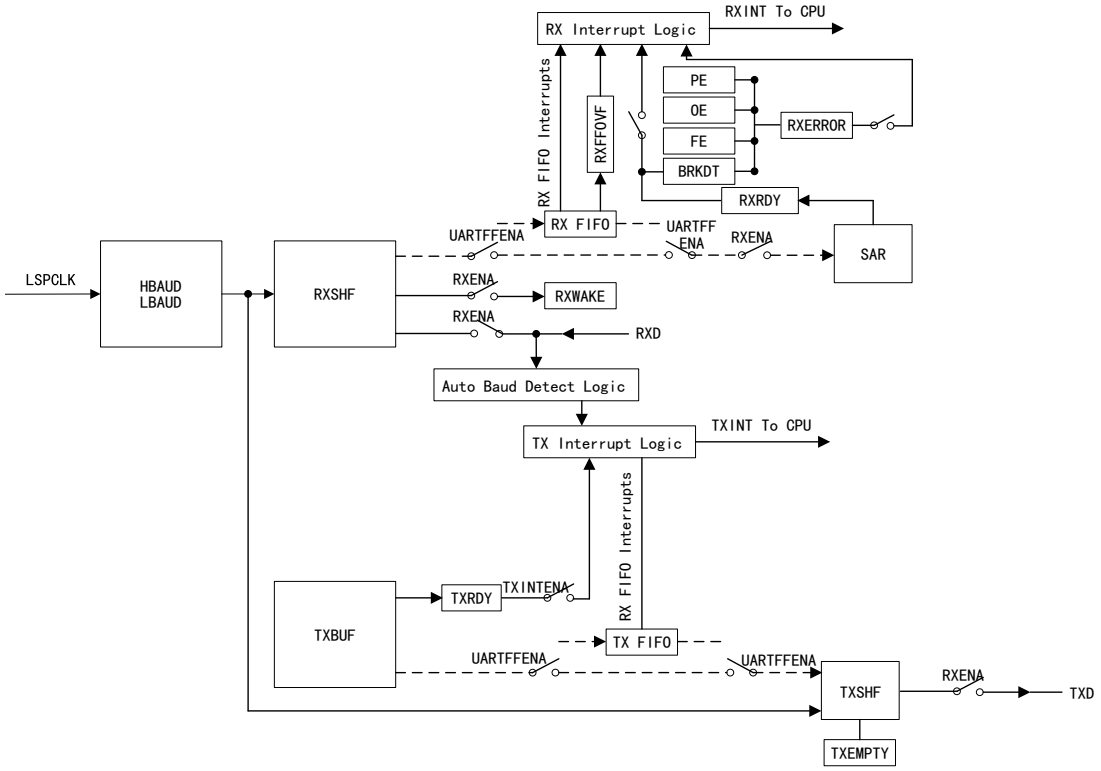
（14） 多处理器通信：
- 空闲线模式
- 地址位模式

（15） 状态标志位：

- 传输检测标志：发送寄存器为空，接收寄存器不为空
- 错误检测标志：溢出错误，中断检测，奇偶校验错误，帧错误

37.4 结构框图

图 201 结构框图



37.5 功能描述

表格 195 UART 信号描述

信号	类型	描述
波特率时钟	控制信号	LSPCLK 预分频时钟
TXD	外部信号	异步串口发送数据
RXD		异步串口接收数据
TXINT	中断信号	发送中断
RXINT		接收中断

37.5.1 配置引脚

表格 196 UART 引脚相关功能

操作	功能
配置 GPIO 多路复用寄存器	连接该外设到设备引脚
将适当的 GPIO 输入类型位设置为 11	GPIO 输入限定设置为异步模式

操作	功能
先配置 GPIO 高复用 n 位，再配置 GPIO 低复用 n 位	避免引脚上的毛刺

可以在 GPIO 上下拉 n 位中配置内部上拉电阻。部分 IO 功能由独立于该外设的 GPIO 寄存器设置定义。

37.5.2 帧格式

通过 UARTCCR 寄存器控制数据帧的帧格式

- UARTCHAR 位控制字符长度，可设置为 1 位-8 位
- PARITYENA 位控制是否开启校验位
- PARITY 位控制校验位为奇校验还是偶校验
- STOPBITS 位控制停止位长度，可设置为 1 个或 2 个

校验位

UARTCCR 的 PARITY 位决定奇偶校验位，当 PARITY=0 时，为奇校验，反之为偶校验。

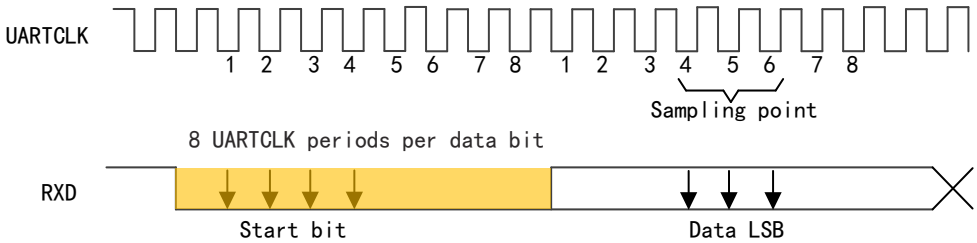
- 偶校验：帧数据和校验位‘1’的个数为偶数时，偶校验位为 0，否则为 1。
- 奇校验：帧数据和校验位‘1’的个数为偶数时，奇校验位为 1，否则为 0。

37.5.3 通信格式

使用单线或双线通信，每个数据位有 8 个 UARTCLK 周期。

在接收到四个连续内部 UARTCLK 周期的低电平后标识有效的起始位，此时接收器开始工作。如果没有位为零，则处理器重新寻找一个起始位。处理器通过在位的第四、第五和第六个 UARTCLK 周期上采样来确定起始位之后的位，位值取决于出现次数多的那个。由于接收器自动与帧同步，外部发送器和接收器不需要使用同步的串行时钟，所以时钟可以在本地生成。

图 202 异步通信格式

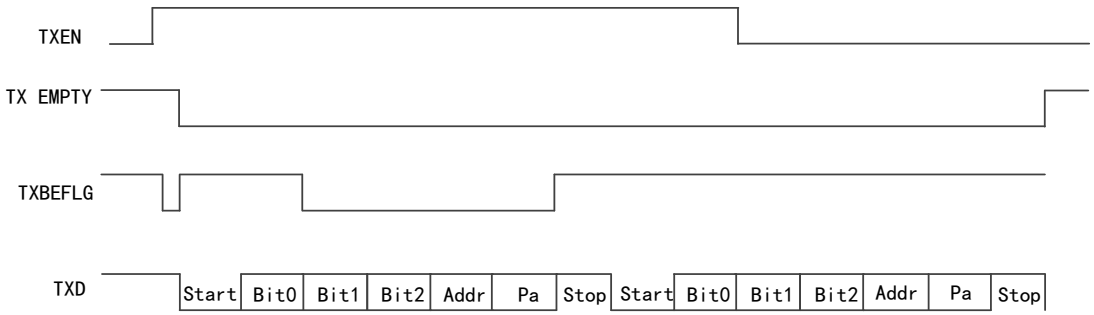


发送器信号

地址位唤醒模式时，TXENA 位置高可以使能发送器，变低则禁用，可以完成发送当前字符。第一个字符传输完成后第二个字符到移位寄存器 TXSHF 的传输开始。UART 将数据转移到 TXSHF。TXRDY 变高表示发送器已经准备好发送第二

个字符，并请求中断。TXRDY 变高后，程序向 UARTTXBUF 写入第二个字符。如果每个字符三位，发送器信号时序图如下：

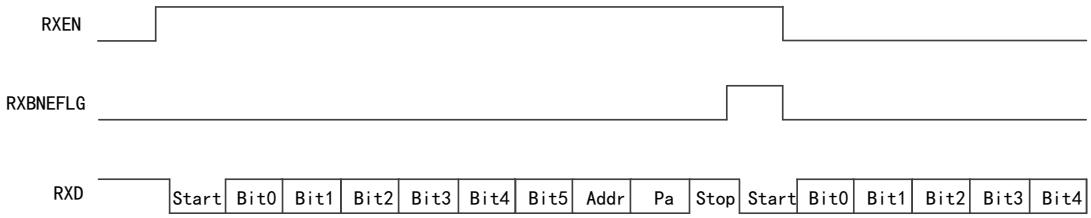
图 203 发送器信号时序



接收器信号

地址位唤醒模式时，RXENA 位置高可以使能接收器，变低则禁用，数据继续在 RXSHF 中组装，但不会传输到接收缓冲寄存器。当数据到达 RXD 引脚后会检测到起始位。数据从 RXSHF 移至 UARTRXBUF，需要请求一个中断。标志位 RXRDY 变高，表示已接收到新字符,读取 UARTRXBUF 时自动清除。如果每个字符六位，接收器信号时序图如下：

图 204 接收器信号时序



37.5.4 波特率

低速外设时钟和 UARTHBAUD，UARTLBAUD 寄存器决定内部生成的串行时钟。对于已经给定的 LSPCLK，UART 可以使用 UARTHBAUD，UARTLBAUD 寄存器的 16 位值来选择 64K 个不同串行时钟速率中的一个。根据计算异步波特率的公式。LSPCLK/16 是最大波特率。如果 LSPCLK 为 100 MHz，则最大波特率为 6.25 Mbps。

表格 197 异步波特率寄存器值

LSPCLK 时钟分频	100MHz 实际波特率	错误率
324	38400	0.16
650	19200	0.01
1301	9601	0.01
2603	4800	0
5207	2400	0

37.5.5 自动波特率

一般的 UART 不具备内置的自动波特率检测。这些 UART 模块集成了依赖 PLL 复位值的嵌入式控制器时钟频率。一般嵌入式控制器的时钟频率都会发生变化。

37.5.6 自动波特率检测序列

设置 UARTRST 位可以使能自动波特率。UARTFFCT 的 CDC 位和 ABD 位控制自动波特率。

如果 CDC=1 时设置 ABD，表示自动波特率对齐，此时 UART 将发送 FIFO 中断。在中断服务后，必须通过软件清除 CDC 位。如果 CDC 在中断服务后仍然置 1，将不会再次发生中断。

在较高的波特率下，发送器和连接器都可能会影响传入数据位的电压转换速率。如果超过 100k 波特率，可能会影响自动波特率检测的可靠性并导致自动波特率锁定功能失败。可以在主机和 UART bootloader 之间使用较低的波特率实现波特率锁定，然后主机可以将波特率寄存器设置为所需的更高的波特率来防止出现以上情况。

37.5.7 多处理器协议

多处理器协议允许多个处理器之间进行高效的数据传输。分为空闲线模式和地址位模式。

37.5.8 多处理器通信

多处理器通信允许一个处理器有效地向同一串行线上的其他处理器发送数据块。在一个串行线上，同一时刻只能有一个传输数据快。

睡眠模式

如果想要处理器只在检测到地址字节时被中断，串行线上的所有处理器都将设置 SLEEP。当处理器读取到与软件设置的 CPU 设备地址匹配的地址时，程序要清除 SLEEP 位，才能使 UART 在接收每个数据字节时生成中断。即使 SLEEP=1，接收器也可以工作，此时需要在检测到地址字节并且接收帧中的地址位为 1 时，接收器才会将 RXRDY，RXINT 或接收器错误状态位设置为 1。SLEEP 不会由 UART 主动设置，需要由软件设置。

地址字节

所有监听器都会读取地址字节，它包含在发送的信息块中的第一个字节。具有正确地址的监听器会被后续数据字节中断。错误地址的监听器直到下一个地址字节前不会中断。

识别地址字节

- 空闲线模式应用于典型的非多处理器 UART 通信。此模式会在地址字节前留一段静默时间。没有额外的地址/数据位，能更快的处理超过十个字节的数据块。

- 地址位模式在每个字节中添加了一个额外的地址位，用来区分地址和数据。因为不需要等待数据块之间的时间，所以此模式处理小数据块时效率更高。但是在高速传输速度下，程序不够快，无法避免传输流中的 10 位空闲时间。

发送接收

ADDRIDLE_MODE 位配置多处理器模式。这两种模式都使用 TXWAKE 位，RXWAKE 位和 SLEEP 位来控制这些模式的发送器和接收器功能。

接收顺序

接收到地址块时，UART 端口会唤醒并请求中断。它读取数据块中包含目的地址的第一个帧。通过中断进入软件程序并检查传入的地址。比较此地址字节与存储在内存中的设备地址字节。如果该数据块是发送给设备 CPU 的，则 CPU 会清除 SLEEP 位并读取数据块的其余部分。否则软件例程将退出，SLEEP 位仍然保持设置并且不会产生接收中断，直到下一个数据块开始之前。

37.5.9 空闲线模式

在此模式中，两个数据块之间的空闲时间比数据块内的两个帧之间的空闲时间长。一个帧后的十位或更多高电平位的空闲时间表示新数据块的开始。可以根据波特率计算传输一个位的时间。

步骤

（1） 接收到“块开始”信号后，UART 唤醒并请求中断。

（2） 处理器识别到下一个 UART 中断。

（3） 中断服务程序将会比较接收到的地址与自身地址。

（4） 如果 CPU 被寻址，服务程序清除 SLEEP 位并接收数据块的其余部分。反之 SLEEP 位保持不变。此时 CPU 可以继续执行其主程序，直到检测到下一个数据块的开始前都不会被 UART 端口中断。

在空闲线模式下，如果 UART 从 FIFO 读取所有 RXDATA 的时间大于 10 位，UART 可能会错过要检测的立即块开始。RXWAKE 在识别到 10 位 IDLE 周期时断言。如果读取 UARTRXBUF，即使在这之后仍然空闲也不会再次断言。

当 CPU 需要大于 10 位的 UART 时钟从 UARTRXBUF/ FIFO 读取数据时。使用以下两种方式：

- 如果在读取 UARTRXBUF 寄存器之前设置了 RXWAKE 位，则不能在 ISR 结束时设置 SLEEP 位。
- 设置 UARTCTL1。在 ISR 结束时读取所有 RX 数据后执行 SWRESET。

块开始信号

发送块开始信号的方法如下：

- 在上一个块中最后一帧数据的传输和新块的地址帧传输之间延迟十个或更多位的空闲时间。
- 先设置 TXWAKE 位，再写入 UARTTXBUF 寄存器。可以发送 11 位的空闲时间。此时串行通信线路的空闲时间不会超过必要的时间。

临时唤醒标志

临时唤醒标志是一个内部标志，与 TXWAKE 位双缓冲。当从 UARTTXBUF 加载 TXSHF 时，同时从 TXWAKE 加载 WUT，并将 TXWAKE 位清零。

发送一个块开始信号需要先设置 TXWAKE 位，再写入 UARTTXBUF 寄存器。当 TXSHF 再次空闲时，数据由 UARTTXBUF 内移至 TXSHF，TXWAKE 的值移至 WUT，然后清除 TXWAKE。设置 TXWAKE 会使起始位，数据位和奇偶校验位被一个 11 位的空闲时间替代，且在上一帧的停止位之后发送。向 UARTTXBUF 写入一个新的地址值。需要向 UARTTXBUF 寄存器写入一个数据字，TXWAKE 位的值才可以被移至 WUT。因为 TXSHF 和 WUT 是双缓冲的，在数据字被移至 TXSHF 寄存器后，可以再次写入 UARTTXBUF 和 TXWAKE。

接收器操作

接收器不受 SLEEP 位的影响。直到检测到地址帧之前，接收器既不设置 RXRDY 和错误状态位，也不请求接收中断。

37.5.10 地址位模式

设置 ADDRIDLE_MODE 位选择地址位模式，此时帧最后的数据位之后具有一个称为地址位的额外位。地址位在块的第一帧中设置为 1，在所有其他帧中设置为 0。不影响空闲周期时间。地址位格式通常用于 11 个字节或更少的数据帧，可以为所有发送的数据字节添加一个位。空闲格式通常用于 12 个或更多字节的数据帧。

发送地址

TXWAKE 位的值在地址位中。在发送过程中，当 UARTTXBUF 寄存器和 TXWAKE 分别加载到 TXSHF 寄存器和 WUT 时，TXWAKE 复位，WUT 变成当前帧的地址位的值。

要发送一个地址需要先设置 TXWAKE 位，并将地址值写入 UARTTXBUF 寄存器。当这个地址值被转移到 TXSHF 寄存器并被移出时，它的地址位被发送为 1。此时会提示串行线路上的其他处理器读取地址。加载 TXSHF 和 WUT 后，可以立即写入 UARTTXBUF 和 TXWAKE。随后设置 TXWAKE 位来发送块中的非地址帧。

37.5.11 中断

可以通过中断控制接收器和发送器。未设置中断使能位时，中断不会被断言，但是可以反应发送和接收状态。UARTCTL2 [TXRDY]表示中断条件置位情况，UARTRXST 的 BRKDT，RXRDY，和 RXERROR 位都是中断标志位。接收器和发送器有独立的外设中断向量。外设中断请求由外设输出到 NVIC 控制器的优先

级位来表示的，可以是高优先级或低优先级。当 TX 和 RX 同时发生具有相同的优先级的中断请求时，为了减少接收器溢出的可能，接收器的优先级比发送器更高。

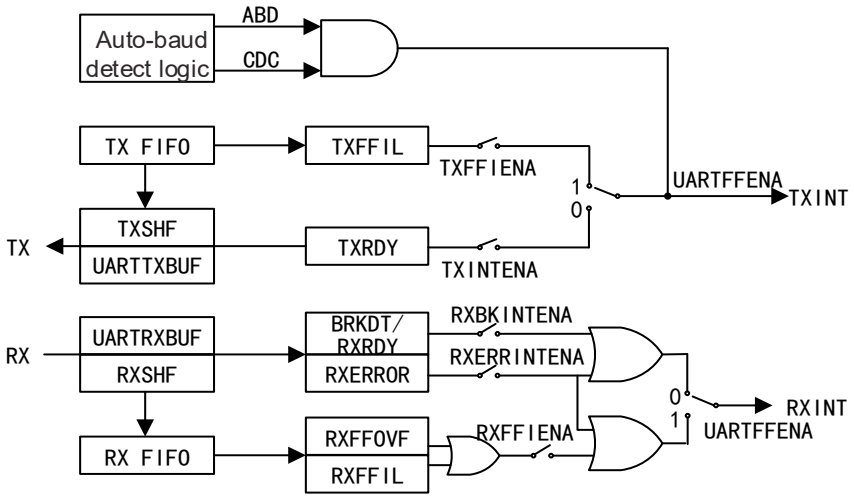
如果设置了 TXINTENA 位，当 UARTTXBUF 中的数据被传输到 TXSHF 寄存器时，就会断言发送器外设中断请求，此时 CPU 可以写入 UARTTXBUF，这将会设置 TXRDY 位并产生中断。

如果设置了 RXBKINTENA 位，当接收到完整的帧并将数据从 RXSHF 寄存器传输到 UARTRXBUF 寄存器，将会设置 RXRDY 位并产生中断。当发生中断检测条件将会设置 BRKDT 位并产生中断。

表格 198 UART 中断请求

中断事件	UARTFFENA 位	事件标志位	使能位
发送为空	0	TXRDY	TXINTENA
接收错误		RXERROR	RXERRINTENA
接收断开		BRKDT	RXBKINTENA
接收数据		RXRDY	
发送为空	1	TXFFIL	TXFFIENA
接收错误或接收断开		RXERROR	RXERRINTENA
接收 FIFO		RXFFIL	RXFFIENA
自动波特率检测	-	ABD	-

图 205 中断映射



37.5.12 FIFO

表格 199 功能及描述

功能	描述
复位	在复位时，以标准模式启动，并且禁用 FIFO 功能。

功能	描述
标准模式	标准模式一般将 TXINT/RXINT 中断作为模块的中断源。
使能 FIFO	设置 UARTFFTX [UARTFFENA]使能 FIFO 模式。UARTRST 位复位 FIFO 模式。
激活	激活所有寄存器。
中断	FIFO 模式的中断分别用于发送 FIFO 和接收 FIFO。在标准模式下禁用 TXINT，作为发送 FIFO 中断。RXINT 是接收错误，接收 FIFO 和接收 FIFO 溢出条件的通用中断。
可编程中断级别	发送和接收 FIFO 都可以生成 CPU 中断。当 TXFFST 位域与 TXFFIL 位域匹配时，将生成中断触发。接收 FIFO 的触发级别默认值是 0x11111，发送 FIFO 的是 0x00000。但它们是可编程的。
缓冲区	发送缓冲区和接收缓冲区有两个 16 级 FIFO。发送 FIFO 寄存器和接收 FIFO 寄存器的宽度分别为 8 位和 10 位。标准 UART 的 UARTTXBUF 在发送 FIFO 和移位寄存器之前作为过渡缓冲区。 当使能 FIFO 时： 因为 UARTTXBUF 不是额外的缓冲区，所以只有在移位寄存器的最后一位被移出后才加载到 FIFO 中。 TXSHF 在可选的延迟量之后直接从 FIFO 加载。 写入 UARTTXBUF 的字符被队列到 TXFIFO 中，并且可以通过 UARTRXBUF 读取到 RXFIFO 中接收到的字符。
延迟传输	FIFO 中的字传输速率可编程。FFTXDLY 可以定义 0-256 个波特率时钟周期的字传输之间的延迟。零延迟时，可以以连续模式传输数据，FIFO 中的字连续输出。在 256 个时钟周期的延迟后以最大延迟模式传输数据，FIFO 中的字之间相隔 256 个波特率时钟周期。可编程延迟可以与慢速 UART 通信，减少 CPU 干预。
FIFO 状态位	发送和接收 FIFO 分别有各自的状态位，用于随时定义 FIFO 中可用的字数。如果设置 TXFIFORESET 位和 RXFIFORESET 位，FIFO 将从头开始重新运行。清零时则会将 FIFO 指针重置为零。

37.6 寄存器组地址

表格 200 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
UartaRegs	UART_REGS	0x5000_0C00	0x5000_0FFF
UartbRegs	UART_REGS	0x5010_0000	0x5010_03FF

37.7 寄存器地址映射

表格 201 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
UARTCCR	通信控制寄存器	0x00	-
UARTCTL1	控制寄存器	0x02	-
UARTHBAUD	波特率高寄存器	0x04	-

寄存器名称	寄存器描述	偏移地址	WRPRT
UARTLBAUD	波特率低寄存器	0x06	-
UARTCTL2	控制寄存器 2	0x08	-
UARTRXST	接收状态寄存器	0x0A	-
UARTRXEMU	模拟器仿真器寄存器	0x0C	-
UARTRXBUF	接收缓冲器寄存器	0x0E	-
UARTTXBUF	发送缓冲器寄存器	0x12	-
UARTFFTX	发送 FIFO 寄存器	0x14	-
UARTFFRX	接收 FIFO 寄存器	0x16	-
UARTFFCT	FIFO/自动波特率控制寄存器	0x18	-
UARTPRI	选择模式寄存器	0x1E	-

37.8 寄存器功能描述

37.8.1 通信控制寄存器(UARTCCR)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	UARTCHAR	R/W	配置字符长度（Character length configure） 配置 1-8 位的字符长度。少于 8 位的字符在 UARTRXBUF 和 UARTRXEMU 中右对齐，在 UARTRXBUF 中需要用前导零填充，UARTRXEMU 则不需要。 000：1 位 001：2 位 … 111：8 位	0h
3	ADDRIDLE_MODE	R/W	选择多处理器模式（multiprocessor mode select） 多处理器通信与其他通信模式不同，它需要使用 SLEEP 和 TXWAKE。地址位模式在帧中增加了一个额外的位，空闲线路模式不增加这个额外的位，并且与 RS-232 类型的通信兼容。空闲线路模式通常用于正常通信。 0：空闲线路模式协议 1：地址位模式协议	0h
4	LOOPBKENA	R/W	使能环回测试模式（Loop Back test mode enable） 使能环回测试模式 Tx 引脚内部连接到 Rx 引脚。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
5	PARITYENA	R/W	使能检验（Parity Enable） 如果 ADDRIDLE_MODE=1，则地址位包含在奇偶校验计算中。少于 8 位的字符应在奇偶校验计算中屏蔽剩余未使用的位。 0：禁止 1：使能	0h
6	PARITY	R/W	选择奇偶校验（Odd/Even Parity Select） PARITYENA=1 时，此位表示奇偶校验。 0：奇校验 1：偶校验	0h
7	STOPBITS	R/W	配置停止位（STOP Bit Configure） 指定发送的停止位的个数。接收器只检查一个停止位。 0：1 个 1：2 个	0h
15:8	保留			0h

37.8.2 控制寄存器(UARTCTL1)

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	RXENA	R/W	使能接收（Receive Enable） 在 RXD 引脚上接收数据，然后经过接收器移位寄存器发送到接收器缓冲区。 0：阻止接收到的字符转移到 UARTRXBUF，UARTRXEMU 也不会产生接收器中断。但是接收器移位寄存器可以继续组装字符。 1：将接收到的字符发送到 UARTRXBUF，UARTRXEMU。	0h
1	TXENA	R/W	使能发送（Transmitte Enable） 禁止此位时写入 UARTTXBUF 的数据，即使后来使能也不会传输。设置此位时数据才通过 TXD 引脚传输。如果之前写入 UARTTXBUF 的数据都已发送，一旦复位，传输就将停止。 0：禁止 1：使能	0h
2	SLEEP	R/W	使能睡眠模式（Sleep mode enabled） 在多处理器配置中，控制接收器睡眠功能。清除此位将退出睡眠模式。SLEEP=1 时，接收器仍然可以操作，但是不更新 UARTRXST [6:2]，除非检测到地址字节。当检测到地址字节时，不清除 SLEEP。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
3	TXWAKE	R/W	选择数据传输特性（data-transmit feature select） 由 ADDRIDLE_MODE 位的设置决定。 0：不选择传输特性。 在空闲线路模式下：向此位写 1，然后向 UARTTXBUF 寄存器写入数据，产生 11 个数据位的空闲周期。 在地址位模式下：向此位写 1，然后将数据写入 UARTTXBUF，将该帧的地址位设置为 1。 1：选择传输特性取决于是什么模式，此位通过系统复位或转移到 WUT 标志来清除。	0h
4	保留			0h
5	SWRESET	R/W	软件复位（software reset） 此位不影响任何配置位。直到设置此位前，所有受影响的逻辑都保持在指定的复位状态。一旦此位被断言，直到解除断言之前，标志都将被冻结。 0：写 0 到这个位将初始化 UARTCTL2 和 UARTRXST 的标志位。 1：在系统复位后，需要写 1 到这个位重新使能 UART。	0h
6	RXERRINTENA	R/W	使能接收错误中断（receive error interrupt enable） 如果由于发生错误而设置了 RXERROR，设置此位可以使能接收错误中断。 0：禁止 1：使能	0h
15:7	保留			0h

37.8.3 波特率高寄存器(UARTHBAUD)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	BAUD	R/W	设置波特率的高位（High baud set） 内部生成的串行时钟由低速外设时钟信号和波特率寄存器决定。使用这些寄存器的 16 位值来选择 64K 串行时钟速率中的一个用于通信模式。 如果 0<BRR<65536： BRR = (UARTHBAUD<<8) + (UARTLBAUD) 波特率的计算公式如下： 异步波特率= LSPCLK / ((BRR+ 1)*8) BRR = LSPCLK / (UART 异步波特率*8)- 1 如果 BRR = 0： 异步波特率= LSPCLK / 16 BRR =波特率寄存器中的 16 位值	0h
15:8	保留			0h

37.8.4 波特率低寄存器(UARTLBAUD)

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	BAUD	R/W	设置波特率的低位（Low baud set） 参考上表的 BAUD 位。	0h
15:8	保留			0h

37.8.5 控制寄存器 2(UARTCTL2)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	TXINTENA	R/W	使能发送数据缓冲器为空中断（Transmit Data Buffer Empty interrupt enable） 在非 FIFO 模式中，为了产生第一次发送中断，必须将一个数据写入 UARTTXBUF。如果在向 UARTTXBUF 写入数据后使能此位，则不会产生中断。 0：禁止 1：使能	0h
1	RXBKINTENA	R/W	使能接收器缓冲区/断开中断（Receiver buffer/break interrupt enable） 0：禁止 1：使能	0h
5:2	保留			0h
6	TXEMPTY	R	发送器为空标志（Transmitter empty flag） 通过系统复位或使能 SWRESET 位设置为 1。不会引起中断请求。 0：发送缓冲区或移位寄存器不为空 1：发送缓冲区和移位寄存器空	0h
7	TXRDY	R	发送数据缓冲器为空标志（Transmit Data Buffer Empty Flag） 向 UARTTXBUF 写入数据会自动清除该此位。 0：UARTTXBUF 已满 1：UARTTXBUF 准备接收下一个字符。如果 TXINTENA=0，则该标志断言发送器中断请求。通过系统复位或使能 SWRESET 位设置为 1。	0h
15:8	保留			0h

37.8.6 接收状态寄存器(UARTRXST)

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h

位/域	名称	R/W	描述	复位值
1	RXWAKE	R	发生接收器唤醒标志（Receiver wake up Occur Flag） 在地址位多处理器模式下，此位反映 UARTRXBUF 中包含的字符的地址位值。在空闲线多处理器模式下，如果检测到 RXD 数据线为空闲，则设置此位。可以通过读取 UARTRXBUF，使能 SWRESET，UARTRST，系统复位或将地址字节后的第一个字节传输到 UARTRXBUF 来清除此位。 0：无接收器唤醒 1：检测到接收器唤醒	0h
2	PE	R	发生校验错误标志（parity error Occur Flag） 当接收到的字符与其奇偶校验位不匹配时，设置此位。如果不使能奇偶校验产生和检测，则禁用此位。可以通过使能 SWRESET，UARTRST 或系统复位来清除此位。 0：无校验错误或禁用校验 1：检测到校验错误	0h
3	OE	R	发生溢出错误标志（Overrun Error Occur Flag） 如果 UARTRXEMU 和 UARTRXBUF 寄存器中有字符时又转移了新的字符进来时设置此位，前一个字符被覆盖并丢失。可以通过使能 SWRESET，UARTRST 或系统复位来清除此位。 0：无溢出错误 1：检测到溢出错误	0h
4	FE	R	发生帧错误标志（Frame Error Occur Flag） 只检查第一个停止位。缺少停止位表明与起始位的同步已经丢失，并且发生帧错误。可以通过使能 SWRESET，UARTRST 或系统复位来清除此位。 注意：在睡眠模式下，当没有 RXWAKE 且 RXD 线的低电平大于 10 位时，只标记 BRKDT。除此之外，此位都将在 BRKDT 之前被标记。 0：无帧错误 1：检测到帧错误	0h
5	BRKDT	R	发生断开标志（Break Occur Flag） 当 RXD 连续保持低电平 9.625 位以上，从丢失第一个停止位开始，就会发生断开。如果 RX 线在 9.625 位期间的变高，那么将不会标记中断检测。 即使 SLEEP=1，也会发生 BRKDT 中断。如果 RXBKINTENA=1，则发生中断会导致接收器中断，但不会加载接收器缓冲区。 检测到中断后，可以通过使能 SWRESET，UARTRST 或系统复位来清除此位。此位和 RXRDY 位不会同时被设置，如果中断后没有复位则不会有进一步的中断。 0：无断开条件 1：发生断开条件	0h

位/域	名称	R/W	描述	复位值
6	RXRDY	R	接收数据缓冲器不为空标志（Receive Data Buffer Not Empty Flag） 如果 RXBKINTENA=1，则产生接收器中断。可以通过读取 UARTRXBUF 寄存器，使能 SWRESET，UARTRST 或系统复位来清除此位。 0：UARTRXBUF 为空 1：UARTRXBUF 不为空	0h
7	RXERROR	R	接收器发生错误标志（receiver Occur error flag） 表示 UARTRXST 中的一个错误标志被设置。这个位可以用于中断服务程序期间的快速错误状态检查。不能直接清除，可以通过使能 SWRESET，UARTRST 或系统复位来清除此位。 0：无错误标志 1：检测到错误标志	0h
15:8	保留			0h

37.8.7 模拟器仿真器寄存器(UARTRXEMU)

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	ERXDT	R	接收模拟器缓冲器（Receive emulation buffer） 一般的数据接收操作是从 UARTRXBUF 寄存器读取接收到的数据。而这个寄存器主要由模拟器使用，因为它可以连续地读取接收到的数据以更新。由系统复位来清除。这个寄存器需要在模拟器监视窗口中使用才能查看 UARTRXBUF 寄存器的内容。UARTRXEMU 只是一个不同的地址位置来访问 UARTRXBUF 寄存器而不清除 RXRDY 标志。	0h
15:8	保留			0h

37.8.8 接收缓冲器寄存器(UARTRXBUF)

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	SAR	R	接收字符位（Receive Character Bit）	0h
13:8	保留			0h
14	UARTFFPE	R	FIFO 校验错误标志（FIFO parity error flag） 如果接收到 7-0 位的字符时有奇偶校验错误，则会设置该标志。这个位与 FIFO 顶部的字符相关联。 0：没有校验错误 1：发生校验错误	0h
15	UARTFFFE	R	FIFO 帧错误标志（FIFO Framing error flag） 如果接收到 7-0 位的字符时有帧错误，则会设置该标志。这个位与 FIFO 顶部的字符相关联。 0：没有帧错误 1：发生帧错误	0h

37.8.9 发送缓冲器寄存器(UARTTXBUF)

偏移地址：0x12

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXDT	R/W	发送缓冲器（Transmit buffer） 如果字符的长度小于 8 位，最左边的位将被忽略。所以要传输的数据位写入 TXDT 时必须右对齐，将数据从这个寄存器传输到 TXSHF 发送器移位寄存器时会设置 TXRDY 标志。如果此时 TXINTENA=1，该数据传输也会导致中断。	0h
15:8	保留			0h

37.8.10 发送 FIFO 寄存器(UARTFFTX)

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	TXFFIL	R/W	发送 FIFO 中断等级（Transmit FIFO interrupt level） 每当 TXFFST 小于或等于 TXFFIL 时，发送 FIFO 产生中断。可以分配给这些位以产生中断的最大值不能超过 TX FIFO 的深度。复位后，这些位的默认值为 00000。	0h
5	TXFFIENA	R/W	使能发送 FIFO 中断（Transmit FIFO interrrupt enable） 0：禁止 1：使能。当 TXFFST 位与 TXFFIL 位匹配时，该中断被触发。	0h
6	TXFFINTCLR	R/W	清除发送 FIFO 中断（Transmit FIFO interrrupt clear） 置位此位可以清除 TXFFINT 标志。	0h
7	TXFFINT	R	发送 FIFO 中断（Transmit FIFO interrupt） 0：未发生中断 1：发生中断	0h
12:8	TXFFST	R	发送 FIFO 字数（Transmit FIFO Word Number） 00000：发送 FIFO 为空 00001：发送 FIFO 有 1 个字 00010：发送 FIFO 有 2 个字 … 10000：发送 FIFO 有 16 个字	0h
13	TXFIFORESET	R/W	重新发送 FIFO（Re-Transmit FIFO） 0：将 FIFO 指针复位为零并保持 1：重新发送 FIFO 操作	1h
14	UARTFFENA	R/W	使能 FIFO（FIFO enable） 0：禁止 1：使能	0h
15	UARTRST	R/W	重置 UART（UART Reset） 0：SWRESET，TXFFINT 和 RXFFINT 被复位，基本清除 TX/RX FIFO 内容。在写 1 之前，UART 将保持复位状态。它还重置 UARTRXST 寄存器，并将 TXEMPTY 和 TXRDY 位设置为 1。 1：FIFO 可以恢复发送或接收。即使自动波特逻辑工作，此位也为 1。	1h

37.8.11 接收 FIFO 寄存器(UARTFFRX)

偏移地址：0x16

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	RXFFIL	R/W	接收 FIFO 中断等级（Receive FIFO interrupt level） 每当 RXFFST 小于或等于 RXFFIL 时，发送 FIFO 产生中断。可以分配给这些位以产生中断的最大值不能超过 RX FIFO 的深度。复位后，这些位的默认值为 11111。	1Fh
5	RXFFIENA	R/W	使能接收 FIFO 中断（Receive FIFO interrrupt enable） 0：禁止 1：使能。当 RXFFST 位与 RXFFIL 位匹配时，该中断被触发。	0h
6	RXFFINTCLR	W	清除接收 FIFO 中断 （Receive FIFO interrrupt clear） 置位此位可以清除 RXFFINT 标志。RXFFINTCLR 和 RXFFOVRCLR 需要同时被清除，否则将阻止进一步的中断发生。	0h
7	RXFFINT	R	接收 FIFO 中断（Receive FIFO interrupt） 0：未发生中断 1：发生中断	0h
12:8	RXFFST	R	接收 FIFO 字数（Receive FIFO Word Number） 00000：接收 FIFO 为空 00001：接收 FIFO 有 1 个字 00010：接收 FIFO 有 2 个字 … 10000：接收 FIFO 有 16 个字	0h
13	RXFIFORESET	R/W	重新接收 FIFO（Re-Receive FIFO） 0：将 FIFO 指针复位为零并保持 1：重新接收 FIFO 操作	1h
14	RXFFOVRCLR	R/W	清除接收 FIFO 溢出标志（Receive FIFO overrun Flag clear） 置位此位可以清除 RXFFOVF 标志。RXFFINTCLR 和 RXFFOVRCLR 需要同时被清除，否则将阻止进一步的中断发生。	0h
15	RXFFOVF	R	接收 FIFO 溢出标志（Receive FIFO overrun Flag） 接收中断处于活动状态时设置。该位由 RXFFOVRCLR，UARTRST 或系统复位清除。 0：未溢出 1：溢出。超过 16 个字被接收到 FIFO 中，并且第一个字丢失	0h

37.8.12 FIFO/自动波特率控制寄存器(UARTFFCT)

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	FFTXDLY	R/W	FIFO 发送延迟（FIFO transfer delay） 从 FIFO 发送缓冲器到发送移位寄存器的每次传输之间的延迟，范围为 0-256 个波特时钟周期。在 FIFO 模式下，移位寄存器和 FIFO 之间的缓冲区需要在移位寄存器完成最后一位的移位后被填充。 在 FIFO 模式下，TXBUF 不能被视为一个额外的缓冲区级别。 当 UART 的停止位为一个时，FFTXDLY 插入的延迟是 FFTXDLY 的波特时钟周期数。当停止位为两个时，FFTXDLY 插入的延迟则需要-1。	0h
12:8	保留			0h
13	CDC	R/W	使能自动波特率对齐（auto-baud alignment Enable） 0：禁止 1：使能	0h
14	ABDCLR	W	清除自动波特率检测标志（Auto-baud detect Flag clear） 置位此位可以清除 ABD 标志。	0h
15	ABD	R	自动波特率检测标志（Auto-baud detect Flag） 0：未完成自动波特检测 1：完成自动波特检测	0h

37.8.13 选择模式寄存器(UARTPRI)

偏移地址：0x1E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	保留			0h
4:3	FREESOFT	R/W	选择当模拟事件挂起时的模式（Select the mode when a emulation event suspends） 00：挂起立即停止 01：停止前完成电流接收/发送序列 10：自由运行 11：自由运行	0h
15:5	保留			0h

38 内部集成电路接口（I2C）

38.1 术语全称、缩写描述

表格 202 术语全称、缩写描述

中文全称	英文全称	英文缩写
串行数据线	Serial Data	SDA
串行时钟线	Serial Clock	SCL
时钟	Clock	CLK
仲裁	Arbitration	ARB
无应答	No Acknowledgement	NACK

38.2 简介

提供了设备与符合 I2C 总线规范版本 2.1 的设备之间的接口，并通过 I2C 总线连接。连接到该 2 线串行总线的外部组件可以通过 I2C 模块向设备发送/接收 1 至 8 位数据。支持兼容 I2C 的主机或从机。

38.3 主要特征

（1） 可选择主机或从机模式
- 多主机发送和从机接收
- 多从机发送和主机接收
- 主机发送/接收和接收/发送模式

（2） 自由数据格式模式

（3） 8 位格式传输

（4） 7 位和 10 位寻址模式

（5） 广播通信

（6） 起始字节模式

（7） 数据传输速率：10 kbps~400 kbps

（8） 接收 FIFO 和发送 FIFO

（9） I2C 中断
- 发送数据准备就绪
- 接收数据准备就绪
- 寄存器访问准备就绪
- 作为从机被寻址
- 未收到应答

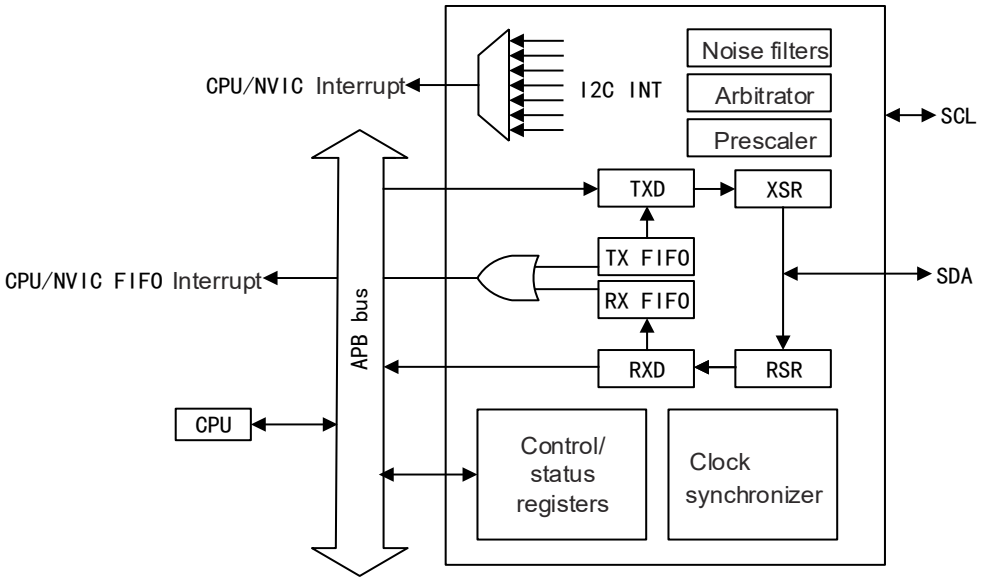
- 总线仲裁失败
- 检测到停止条件

（10） I2CFIFO 中断

- 发送 FIFO 中断
- 接收 FIFO 中断

38.4 结构框图

图 206 结构框图



38.5 功能描述

可编程预分频器将 SYSCLK 进行分频，以生成 I2C 模块的时钟（7~12 MHz），这决定了 I2C 模块的工作频率。进一步分频可以产生 SCL 引脚上的 I2C 主时钟。要先初始化 PSC 位域才能指定分频值。对应关系如下：

I2C Clock=SYSCLK/(PSC+1)

在 I2C 处于复位状态时才能对预分频器进行初始化。分频后的频率在使能 I2C 后生效。当 I2C 被配置为 I2C 总线上的主机时，SCL 引脚成为主机时钟的发送引脚。这个时钟控制 I2C 与从设备间的通信时序。

38.5.1 重置或禁用

I2CEN = 0 时，I2CSTS 寄存器中的所有状态位被强制设为它们的默认值，直到设置 I2CEN 位前 I2C 模块都将保持禁用状态。SDA 和 SCL 引脚处于高阻态。置低 XRS 引脚，将复位整个设备。释放 XRS 引脚时，所有 I2C 寄存器被复位为它们的默认值。I2CEN 位强制为 0，意味着 I2C 被重置，这可以节省电源并清除错误条件。当 I2CEN = 0 时才能配置或重新配置 I2C。

38.5.2 配置引脚

表格 203 I2C 引脚相关功能

操作	功能
配置 GPIO 多路复用寄存器	连接该外设到设备引脚
将适当的 GPIO 输入类型寄存器 n 位设置为 11	GPIO 输入限定设置为异步模式
先配置 GPIO 引脚复用高寄存器 n 位，再配置 GPIO 引脚复用低寄存器 n 位	避免引脚上的毛刺

可以在 GPIO 上下拉配置寄存器 n 位中配置内部上拉电阻。部分 IO 功能由独立于该外设的 GPIO 寄存器设置定义。

38.5.3 操作模式

输入与输出电平

主机为每个传输的数据位生成一个时钟脉冲。由于可以连接到 I2C 总线的各种不同技术设备，高/低电平的电平取决于相关的 VDD 电平。

数据有效性

SDA 的高/低电平状态只能在时钟信号 SCL 为低电平时改变。在时钟的高电平期间，SDA 上的数据必须保持稳定。

操作模式

有四种基本的操作模式，用于支持作为主机和从机的数据传输。如果 I2C 是主机，它将作为主机发送器，通常给特定的从机发送地址。要从从机接收数据，I2C 必须切换到主机接收器模式。如果 I2C 是从机，它将开始作为从机接收器，并在主机识别到其从机地址时发送确认信号。如果主机请求从 I2C 获取数据，则模块必须切换到从机发送器模式。

（1） 主机发送模式

I2C 作为主机发送控制信息和数据给从机。所有主机都从这个模式开始，7 位/10 位地址格式的数据在 SDA 上移出。数据位的移动与 SCL 上生成的时钟脉冲同步。在发送一个字节后，如果需要设备干预，禁止时钟脉冲并将 SCL 保持在低电平。

（2） 主机接收模式

I2C 作为主机从从机接收数据。在主机发送模式下，I2C 先向从机发送一个命令，才能进入此模式。当使用 7 位/10 位地址格式时，在传输从机地址字节和 R/W = 1 后，I2C 进入主机接收模式。

串行数据位根据 SCL 上的时钟脉冲移入 I2C。在接收到一个字节后，如果需要设备干预，禁止时钟脉冲并将 SCL 保持在低电平。

（3） 从机发送模式

I2C 作为从机发送数据。在从机接收模式下，I2C 先从主机接收到一个命令，才能进入此模式。当使用 7 位/10 位地址格式时，如果从机地址字节与其自身地址相同，并且主机已经传输了 R/W = 1，I2C 进入从机发送模式。

作为从机发送器，I2C 将通过主机生成的时钟脉冲将串行数据位移出到 SDA 上。向 TXD 寄存器写入数据以释放 SCL。作为从机时 I2C 不生成时钟信号，在发送一个字节后，如果需要设备干预，可以将 SCL 保持在低电平。

（4） 从机接收模式

I2C 作为从机接收数据。所有从机都从这个模式开始。在此模式下，主机生成的时钟脉冲将接收到的串行数据位移入。从 RXD 寄存器读取数据以释放 SCL。作为从机时 I2C 不生成时钟信号，在接收到一个字节后，如果需要设备干预，可以将 SCL 保持在低电平。

38.5.4 启动和停止

当 I2C 配置为主机时可以生成起始条件和停止条件。在 SCL 为高电平时，SDA 上发生由高到低的转变叫做起始条件。主机发送这个条件就表示数据传输的开始。在 SCL 为高电平时，SDA 上发生由低到高的转变叫做停止条件。主机发送这个条件就表示数据传输的结束。

在起始条件和下一个停止条件之间，I2C 总线为繁忙状态，此时 BBSYFLG = 1。在停止条件和下一个起始条件之间，I2C 总线为空闲状态，此时 BBSYFLG = 0。当 I2C 处于复位时，无法检测到起始条件或停止条件，BBSYFLG 位始终为 0。当 I2C 解除复位时，在检测到起始条件或停止条件前，BBSYFLG 位无法正确反映 I2C 总线的状态。设置 MSCFG 位和 STAGEN 位时，I2C 可以在开始数据传输时发送起始条件。设置 STOGEN 位时，I2C 在结束数据传输时会发送停止条件。当同时设置 BBSYFLG 位和 STAGEN 位时，会生成一个重复的起始条件。

在使用 I2C 进行第一次数据传输之前，需要进行以下操作：

（1） 设置 I2CEN 位可以使能 I2C，然后等待一个比最长数据传输时间更长的时间。这期间要确保至少发生了一个被 BBSYFLG 位捕获的起始条件或停止条件。之后 BBSYFLG 位将正确反映 I2C 总线的状态。

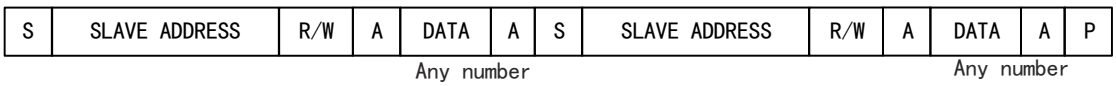
（2） 检查 BBSYFLG 位是否为 0，然后开始数据传输。

在传输时不复位 I2C 可以确保 BBSYFLG 位反映真实的 I2C 总线状态。如果必须在传输时复位 I2C，则每次解除复位时都需要重复执行以上步骤。

38.5.5 重复的起始条件

I2C 主机在每个数据的最后发送另一个起始条件就可以与多个从机地址进行通信。重复的起始条件可以与 7 位寻址，10 位寻址和自由数据格式一起使用。重复起始条件如下图所示：

图 207 重复起始条件



备注：

（1） S：起始信号

（2） SLAVE ADDRESS：从机地址

（3） R/W：发送方向选择位

（4） P：停止信号

38.5.6 重复与非重复

重复模式（REEN = 1）

（1） 软件控制要发送或接收的字节数。

（2） 每个字节的发送和接收结束时，设置 RAIFLG 位。

（3） 一旦开始 I2C 发送，直到完成 I2C 发送之前，不能切换到另一种模式。

非重复模式（REEN = 0）

（1） I2CCNT 寄存器决定要发送或接收的字节数。

（2） 如果 STOGEN = 0，则当内部数据计数器倒计数到 0 时，设置 RAIFLG 位。

（3） 如果 STOGEN = 1，当内部数据计数器倒计数到 0 时，I2C 会生成一个停止条件，不会设置 RAIREN 位。

（4） 一旦开始 I2C 发送，直到完成 I2C 发送之前，不能切换到另一种模式。

（5） 如果 I2CCNT 设置为 0，I2C 状态机期望发送或接收 65536 个字节，而不是 0 个字节。

38.5.7 寻址格式

7 位寻址

复位后默认为 7 位寻址格式。禁用扩展地址和自由数据格式可以使能此格式，此时起始条件后的第一个字节由 7 位从设备地址和一个 R/W 位组成。R/W 确定数据的方向：

- R/W = 0：I2C 主设备向寻址的从设备写入数据（RXTXCFG = 1）。
- R/W = 1：I2C 主设备从从设备读取数据（RXTXCFG = 0）。

在每个字节之后插入一个专用的 ACK 时钟周期。如果从设备在主设备的第一个字节之后插入了 ACK 位，则发送器会发送 DBSEL 位设置的数据。在数据位传输完成后，接收器插入一个 ACK 位。

10 位寻址

设置扩展地址并禁用自由数据格式可以使能 10 位寻址格式。与 7 位寻址格式类似，但主设备将从设备地址分为两个独立的字节传输：

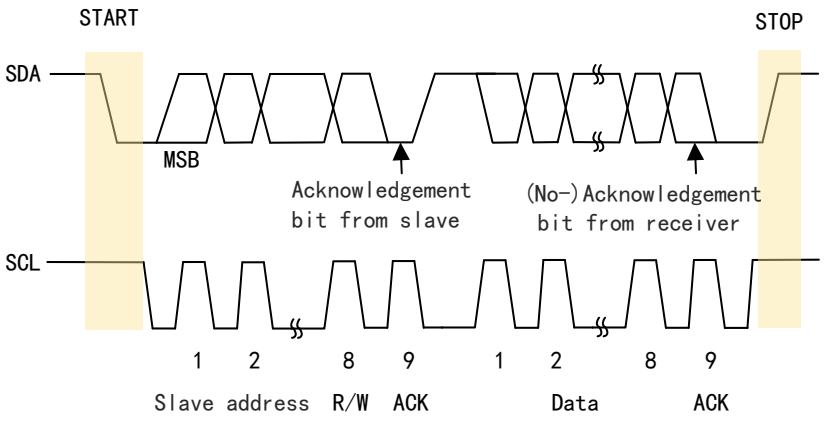
- 第一个字节由 11110，10 位从设备地址的两个最高有效位和 R/W 组成。
- 第二个字节是 10 位从设备地址的剩余 8 位。

从设备需要在每个字节传输后发送确认。一旦主设备将第二个字节写入从设备，主设备可以写入数据或使用重复的起始条件来改变数据方向。

38.5.8 串行数据格式

支持 1 到 8 位的数据值。在 SDA 线上的每个位相当于 SCL 线上的 1 个脉冲，数值以最高有效位优先进行传输。可以发送或接收的数据值数量不受限制。以下为 7 位寻址格式的示例：

图 208 串行数据格式



38.5.9 自由数据格式

FDFEN= 1 可以使能自由数据格式。此时在起始条件后的第一个字节是数据字节。该字节的位数由 DBSEL 位决定。在每个数据字节之后插入一个 ACK 位。不发送地址或数据方向位。所以发送器和接收器都支持自由数据格式，并且数据的方向在整个传输过程中保持不变。此格式不支持数字环回模式。

表格 204 配置 FDFEN 位和 MSCFG 位

FDFEN	MSCFG	I2C 状态	RXTXCFG 状态
0	0	非自由数据格式，从设备模式	不受影响。根据主设备的命令，I2C 会作为发送器或接收器进行响应。
1	0	自由数据格式，从设备模式	自由数据格式要求 I2C 在整个传输过程中保持为发送器或接收器。 RXTXCFG = 0：I2C 是接收器 RXTXCFG = 1：I2C 是发送器
0	1	非自由数据格式，主设备模式	RXTXCFG = 0：I2C 是接收器 RXTXCFG = 1：I2C 是发送器
1	1	自由数据格式，主设备模式	

38.5.10 时钟分频器寄存器

当 I2C 作为主机时，再次分频 I2C 时钟可以用作 SCL 引脚上的主时钟。主时钟的跳变由 CLKH 和 CLKL 决定。

主时钟周期与模块时钟周期的关系如下：

主时钟周期 = [(CLKH+d)+(CLKL+d)] / I2C 模块时钟

其中 d 位延迟，PSC 位域决定 d 的值，当 PSC 为 0 时，d 为 7。当 PSC 为 1 时，d 为 6。其余情况 d 都为 5。

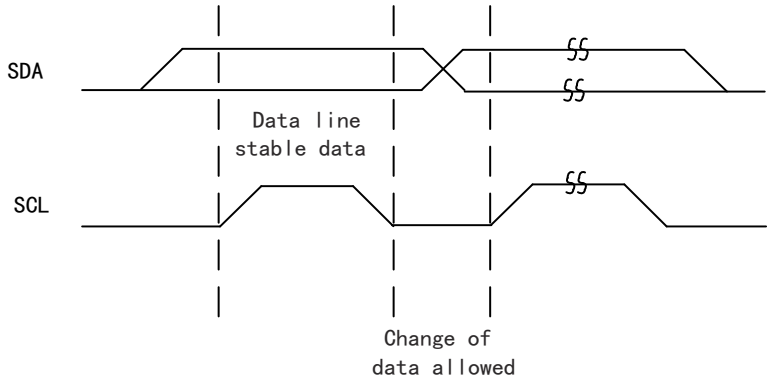
38.5.11 输入输出电平

主机为每个传输的数据位生成一个时钟脉冲。由于可以连接到 I2C 总线的各种不同技术设备，逻辑 0 和逻辑 1 的电平取决于相关的 VDD 电平。有关详细信息，请参阅数据手册。

38.5.12 数据有效性

在时钟的高电平期间，SDA 上的数据必须保持稳定。数据线 SDA 的高电平或低电平状态只能在时钟信号 SCL 为低电平时改变。

图 209 数据有效性

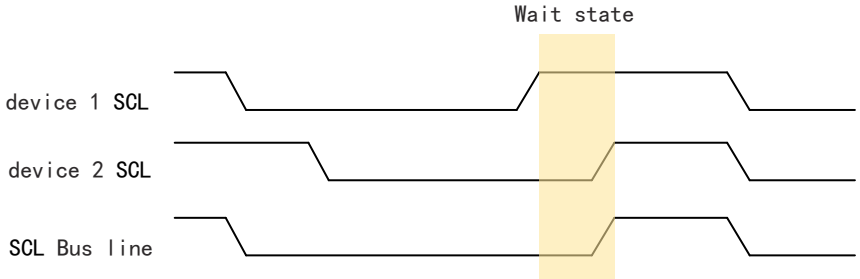


38.5.13 时钟同步

一般只有一个主设备会生成时钟信号 SCL。但是在仲裁过程中可能存在两个或多个主设备，所以需要同步时钟以便比较数据输出。

第一个在 SCL 上生成低电平的设备会覆盖其他设备。在从高电平到低电平的过渡中，强制其他设备的时钟发生器开始它们自己的低电平周期。低电平周期最长的设备会将 SCL 拉低。其他完成低电平周期的设备需要等 SCL 被释放后才能开始它们的高电平周期。如果一个设备将时钟线拉低的时间更长，所有的时钟发生器都要进入等待状态。所以从设备可以减慢高速主设备的速度，低速设备可以创建足够的时间来存储准备发送或已接收的字节。在 SCL 上获得同步的信号中，最慢的设备确定低电平周期的长度，最快的设备确定高电平周期的长度。

图 210 两个 I2C 时钟发生器同步



38.5.14 仲裁

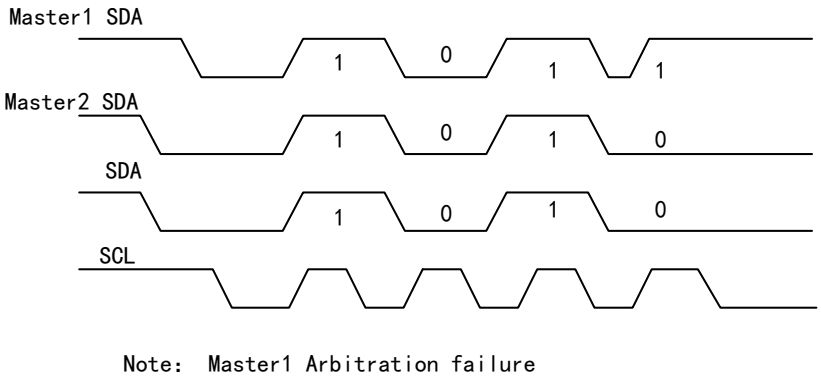
仲裁发生在两个或多个主机-发送器试图于相同的时间内在同一总线上启动传输时。当多个发送器竞争 SDA 的使用权时，仲裁机制决定传输的优先权。仲裁优先传输具有最低二进制值的数据流。将 SDA 拉低的主机-发送器会覆盖第一个释放 SDA 线为高电平的主机-发送器。如果两个或多个设备发送的第一个字节相同，仲裁将在后续的字节上继续进行。

如果 I2C 是仲裁失败的主机，它会切换到从机-接收器模式，同时设置 ARBLIFLG 位并生成仲裁丢失中断请求。如果在传输过程中仍在进行仲裁过程，但重复的起始条件或停止条件也传输到 SDA，那么相关主机-发送器必须在格式帧中的相同位置发送重复的起始条件或停止条件。

以下情况不能进行仲裁：

- 重复的起始条件和数据位间
- 重复的起始条件和停止条件间
- 停止条件和数据位间

图 211 仲裁程序



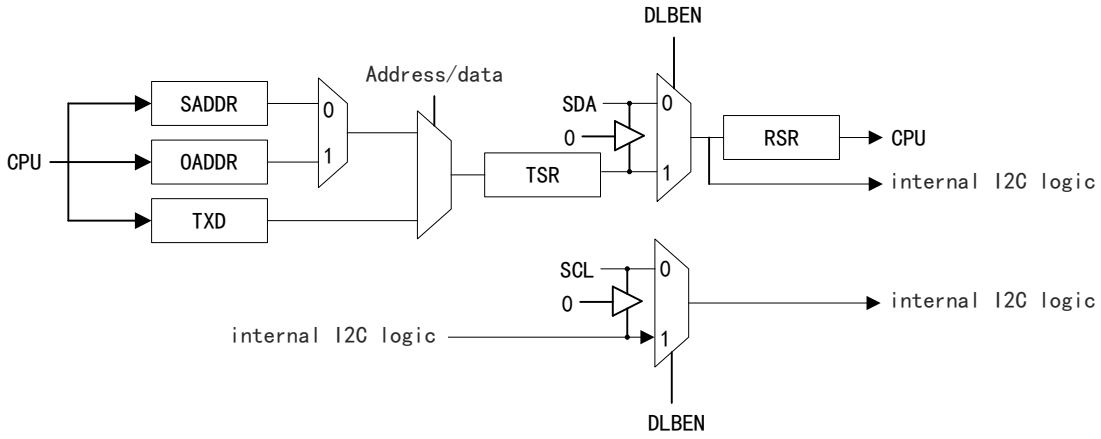
38.5.15 数字环回

置位 DLBEN 位可以使能数字环回模式。此时从 TXD 寄存器发出的数据沿着内部路径传输，并需要 n 个周期才能到达 RXD，n 由以下决定：

$$n = 8 * (SYSCLK) / (I2C\ 模块时钟)$$

发送时钟和接收时钟相同。在外部 SDA 引脚上看到的地址是 OADDR 寄存器中的地址。此模式下不支持自由数据格式。

图 212 数字环回模式



38.5.16 生成 NACK 位

当 I2C 作为接收器时，NACK 位可以确认或忽略由发送器发送的位。I2C 在总线上的确认周期内发送一个不应答位会忽略新的位。

各种模式生成 NACK 位的方式不相同，具体方式如下：

（1） 主机接收机模式和重复模式

- 重置模块
- 生成一个停止条件
- 在计划接收的最后一个数据位的上升沿之前设置 NACKCFG 位

（2） 主机接收机模式和非重复模式

- 重置模块
- 如果 STOGEN = 0，使 STOGEN = 1 以生成停止条件
- 如果 STOGEN = 1，允许内部数据计数器计数到 0 而强制生成停止条件
- 在计划接收的最后一个数据位的上升沿之前设置 NACKCFG 位

（3） 从机接收机模式

- 重置模块
- 允许溢出条件
- 在计划接收的最后一个数据位的上升沿之前设置 NACKCFG 位

38.5.17 中断请求

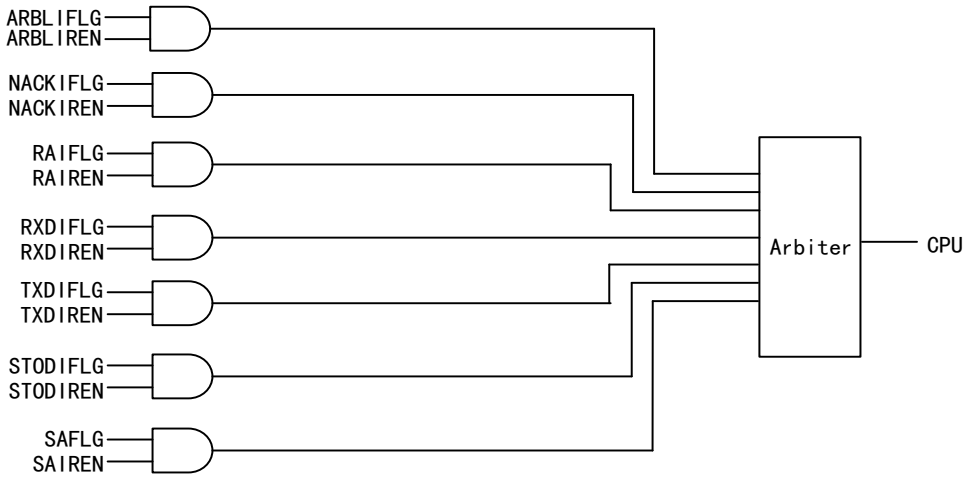
基本中断请求

所有请求都通过仲裁器多路复用到单个 I2C 中断请求发送到 CPU。每个中断请求都有一个使能位和标志位。当发生指定的事件时设置标志位。如果此时相应的使能位为 0，则中断请求阻塞。如果为 1，则请求作为 I2C 中断转发到 CPU。基本中断请求的优先级从高到低的顺序如下：

表格 205 I2C 中断请求

中断事件	事件标志位	中断控制位
仲裁丢失	ARBLIFLG	ARBLIREN
无应答	NACKIFLG	NACKIREN
寄存器访问就绪	RAIFLG	RAIREN
接收数据就绪	RXDIFLG	RXDIREN
发送数据就绪	TXDIFLG	TXDIREN
停止条件检测	STODIFLG	STODIREN
从机地址	SAFLG	SAIREN

图 213 I2C 中断映射

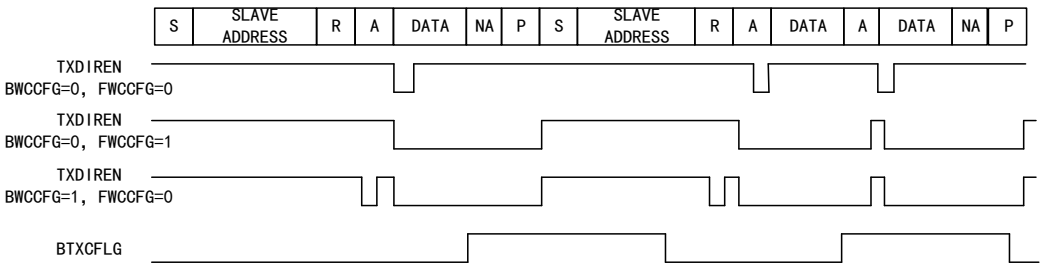


置位 FWCCFG 位可以避免普通发送中断时序可能导致在发送字节事务的中断被打断时，传输缓冲区中保留的是之前的旧数据。如果该位置位，当总线事务需要时才会生成传输数据准备中断。

- 在主机模式下，当接收到地址字节的 ACK 时首先生成中断。
- 在从机模式下，当地址匹配时首先生成中断。当数据被 ACK 时，产生进一步的中断。TXDIREN 与发送就绪中断同时断言。

EXT 寄存器中的 BWCCFG 位和 FWCCFG 位会影响 I2C 和中断。当配置为从机-发送器时，这两个位对 I2C 模块寄存器和中断的影响如下图所示：

图 214 向后兼容位和向前兼容位



备注：

（1） S：起始信号

（2） SLAVE ADDRESS：从机地址

（3） R/W：发送方向选择位

（4） A：应答位

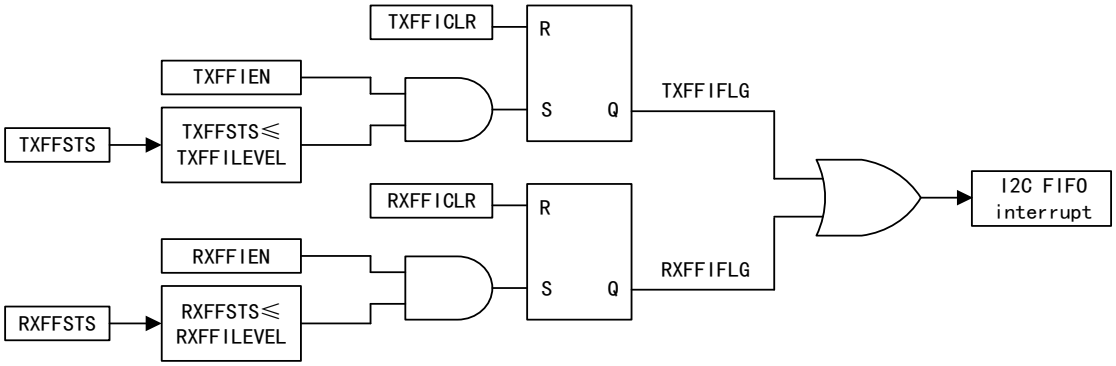
（5） NA：不应答位

（6） P：停止信号

FIFO 中断

除了基本 I2C 中断外，发送和接收 FIFO 都可以生成中断。可以配置为在发送/接收指定数量的字节后生成中断。这两个中断源被或运算到一起形成一个可屏蔽的 CPU 中断。

图 215 FIFO 中断



38.6 寄存器组地址

表格 206 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
I2cRegs	I2C_REGS	0x5000_1400	0x5000_17FF

38.7 寄存器地址映射

表格 207 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
OADDR	I2C 自身地址寄存器	0x00	-
IREN	使能中断请求寄存器	0x02	-
STS	状态寄存器	0x04	-
CLKL	时钟低分频器	0x06	-
CLKH	时钟高分频器	0x08	-

寄存器名称	寄存器描述	偏移地址	WRPRT
CNT	计数器	0x0A	-
RXD	接收数据寄存器	0x0C	-
SADDR	从机地址寄存器	0x0E	-
TXD	发送数据寄存器	0x10	-
CTRL	控制寄存器	0x12	-
ISRC	中断源寄存器	0x14	-
EXT	扩展寄存器	0x16	-
PSC	预分频寄存器	0x18	-
TXFIFO	发送 FIFO 寄存器	0x40	-
RXFIFO	接收 FIFO 寄存器	0x42	-

38.8 寄存器功能描述

38.8.1 I2C 自身地址寄存器(I2COADDR)

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	OADDR	R/W	I2C 自身地址（I2C own address） I2C 模块通过此寄存器定义自身的从机地址，用于在 I2C 总线上与其它从机设备区分 7 位寻址模式：位域 [6:0]提供 I2C 的 7 位从地址。 10 位寻址模式：位域 [9:0]提供 I2C 的 10 位从地址。	0h
15:10	保留			0h

38.8.2 使能中断请求寄存器(I2CIREN)

偏移地址：0x02

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ARBLIREN	R/W	使能仲裁丢失中断请求（Arbitration lost interrupt request enable） 0：禁止 1：使能	0h
1	NACKIREN	R/W	使能无应答中断请求（No acknowledgment interrupt request enable） 0：禁止 1：使能	0h
2	RAIREN	R/W	使能寄存器访问就绪中断请求（Register access ready interrupt request enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
3	RXDIREN	R/W	使能接收数据就绪中断请求（Receive data ready interrupt request enable） 使用 FIFO 模式时，不能设置此位。 0：禁止 1：使能	0h
4	TXDIREN	R/W	使能发送数据就绪中断请求（Transmit data ready interrupt request enable） 使用 FIFO 模式时，不能设置此位。 0：禁止 1：使能	0h
5	STODIREN	R/W	使能停止条件检测中断请求（Stop condition detected interrupt request enable） 0：禁止 1：使能	0h
6	SAIREN	R/W	使能从机地址中断请求（Slave Addressed interrupt request enable） 0：禁止 1：使能	0h
15:7	保留			0h

38.8.3 状态寄存器(I2CSTS)

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ARBLIFLG	R/W1C	仲裁丢失中断标志（Arbitration lost interrupt flag） 仅适用于 I2C 是主发送器时生效。当 I2C 检测到与其它主发送器同时开始传输导致仲裁失败，或者在总线忙（BBSYFLG = 1）时尝试启动传输时，该位便会置位。CPU 可以轮询这个标志位或触发中断请求来进行处理。 0：仲裁未丢失 1：仲裁丢失 以下任一操作均可清除该位： -往该位写 1 进行清除 -CPU 读取了 I2CISRC 寄存器 -I2C 模块复位	0h

位/域	名称	R/W	描述	复位值
1	NACKIFLG	R/W1C	无应答中断标志（No acknowledgment interrupt flag） 用于指示 I2C 主设备在传输过程中是否检测到从设备返回的 NACK 信号，如果硬件设备检测到从设备未应答（NACK 信号），该位便会置位。CPU 可以轮询这个标志查询 NACK 状态或者触发中断请求，通知 CPU 处理异常状态。 0：未收到 NACK 1：收到 NACK 以下任一操作可清除该位： -从设备发送 ACK -往该位写 1 进行清除 -CPU 读取了 I2CISRC 寄存器 -I2C 模块复位	0h
2	RAIFLG	R/W1C	寄存器访问就绪中断标志（Register access ready interrupt flag） 仅适用于 I2C 模块处于主模式时，用于指示寄存器是否可被访问。CPU 可以轮询该标志位或使用 RAIFLG 触发中断请求，通知 CPU 更新寄存器值。在非重复模式下如果 STOGEN = 0，则在内部数据计数器归零时此位置 1。在重复模式下 TXD 发送的每个字节的末尾此位置 1。 0：未就绪（传输进行中） 1：就绪（传输空闲） 以下任一操作可清除该位： -往该位写 1 进行清除 -2C 开始使用当前寄存器内容 -I2C 模块复位	0h
3	RXDIFLG	R/W1C	接收数据就绪中断标志（Receive data ready interrupt flag） 非 FIFO 模式时，当接收数据已经准备好被读取时，该位置 1。处于 FIFO 模式时，需要使用 RXFFIFLG 标志代替 RXDIFLG。CPU 可以轮询该标志位或使用 RXDIFLG 中断请求，通知 CPU 数据已经就绪，可以进行获取数据。 0：未就绪 1：就绪 以下任一操作可清除该位： -往该位写 1 进行清除 -CPU 读取数据寄存器 I2CRXD 后自动清除 -I2C 模块复位	0h

位/域	名称	R/W	描述	复位值
4	TXDIFLG	R	发送数据就绪中断标志（Transmit data ready interrupt flag） 非 FIFO 模式时，此位置 1 表示 TXD 寄存器可以接收新数据。处于 FIFO 模式时，使用 TXFFIFLG 标志代替 TXDIFLG。CPU 可以轮询该标志位或使用 TXDIFLG 中断请求，通知 CPU 发送数据寄存器已经就绪，可以进行写入发送数据。 0：未就绪 1：就绪 以下任一操作可清除该位： -CPU 读取数据寄存器 I2CRXD 后自动清除 -I2C 模块复位	1h
5	STODIFLG	R/W	停止条件中断标志（Stop condition detected interrupt flag） I2C 为主模式时，当主设备主动发送停止条件时置位。为从设备时，当从设备检测到总线上的停止条件时置位。当 STOGEN 置位时，模块会在该位置位后，再清除 STOGEN 位。 0：未检测到停止条件 1：检测到停止条件 以下任一操作可清除该位： -往该位写 1 进行清除 -CPU 读取到 I2CISR 的值为 110 时会清除此位 -I2C 模块复位	0h
6	BTXCFLG	R/W	字节发送完成标志（Byte transmit complete flag） 当主/从服务器在 SCL/SDA 上成功发送字节时设置此位。 0：字节发送未完成 1：字节发送完成 向此位写 1 或复位 I2C 模块会清除此位。	0h
7	保留			0h
8	ZEROAFLG	R	全零地址标志（Aall-zeros address flag） 0：未检测到全零地址 1：检测到全零地址	0h
9	SAFLG	R	从机地址匹配状态标志（Slave Addressed flag） 0：在 7 位寻址模式下，当接收到 NACK 信号，检测到 STOP 条件或检测到重复 START 条件时，清除此位。 在 10 位寻址模式中，当接收到 NACK 信号，检测到 STOP 条件或接收到与模块自身地址不匹配的从机地址时，清除此位。 1：I2C 已经识别出自己的从地址或全零地址	0h

位/域	名称	R/W	描述	复位值
10	TXSNEFLG	R	发送移位寄存器非空标志（Transmit shift register not empty flag） 当传输移位寄存器为空，但 TXD 寄存器未加载时，就会发生下溢。直到新数据在 TXD 中，才会发生下一次传输。如果新的数据没有及时传输，以前的数据可能会在 SDA 引脚上重新传输。 0：发送移位寄存器空 1：发送移位寄存器非空 写入 TXD 或复位 I2C 模块会设置此位。	1h
11	RXSFFLG	R	接收移位寄存器满标志（Receive shift register full flag） 如果还没有将移位寄存器中的旧数据读出又接收到新数据时，就会发生溢出。当新的位从 SDA 引脚到达时会覆盖接收移位寄存器中的位。在旧数据被读取之前，新数据不会被复制到 RXD 中。 0：未检测到溢出 1：检测到溢出 RXD 被 CPU 读取或复位 I2C 模块会清除此位。	0h
12	BBSYFLG	R	总线忙标志（Bus busy flag） 0：总线空闲 1：总线忙 I2C 发送/接收到 STOP 条件或复位 I2C 模块会清除此位。	0h
13	TXNACKFLG	R/W1C	发送无应答标志（Transmit no acknowledgment flag） 当 I2C 处于接收模式时使用此位指示是否已发送 NACK 状态。 0：未发送 NACK 1：已发送 NACK 向此位写 1 或复位 I2C 会清除此位。	0h
14	SFLG	R/W1C	从机标志（Slave flag） 该位用于指示 I2C 模块是否被总线上的主设备寻址为发送器。 0：I2C 为非从发送器模式 1：I2C 作为从发送器模式 以下任一操作可清除该位： -向该位写 1 -检测到总线上的 START 或 STOP 条件 -当启用数字环回环测试时，该位自动清除	0h
15	保留			0h

38.8.4 时钟低分频器(I2CCLKL)

偏移地址：0x06

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CLKL	R/W	低电平时间分频值（Clock low-time divide value） CLKL 是 I2C 模块中用于配置主时钟低电平持续时间的关键寄存器，其结合与分频器共同决定总线时钟的低电平周期。主时钟低电平持续时间为模块时钟的周期乘以(CLKL + d)，d 是基于预分频器的调整因子。 注意：必须为非零值才能正确生成 I2C 时钟。	0h

38.8.5 时钟高分频器(I2CCLKH)

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CLKH	R/W	高电平时间分频值（Clock high-time divide value） CLKH 是 I2C 模块中用于配置主时钟高电平持续时间的关键寄存器，其结合与预分频器共同决定总线时钟的高电平周期，需与 CLKL 协同工作已生成完整的时钟波形。主时钟高电平时间为模块时钟的周期乘以(CLKH + d)，d 是基于预分频器的调整因子。 注意：必须为非零值才能正确生成 I2C 时钟。	0h

38.8.6 计数器(I2CCNT)

偏移地址：0x0A

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	CNT	R/W	计数器值（Count value） 传输或接收的数据字节数。如果指定了一个 STOP 条件，那么 CNT 将在每个字节发送后减少，直到达到零，这反过来将生成一个 STOP 条件。 0000000000000000：计数值为 65536 0000000000000001：计数值为 1 0000000000000010：计数值为 2 … 1111111111111111：计数值为 65535	0h

38.8.7 接收数据寄存器(I2CRXD)

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	RXD	R	接收数据（Receive data） I2C 可以接收 1 ~ 8 位的数据字节。位的数量由 DBSEL 位决定。当接收到一个完整的数据字节时，I2C 将数据字节从接收移位寄存器复制到 RXD。CPU 无法直接访问接收移位寄存器。如果在 RXD 中存在小于 8 位的数据字节，则该数据值为右对齐，RXD 的其他位未定义。例如，DBSEL = 011，则接收数据为 RXD[2:0]，RXD[7:3]的内容未定义。当处于接收 FIFO 模式时，RXD 寄存器充当接收 FIFO 缓冲区。	0h
15:8	保留			0h

38.8.8 从机地址寄存器(I2CSADDR)

偏移地址：0x0E

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	SADDR	R/W	设置从机地址（Slave address Set） 在 7 位寻址模式下：SADDR [6:0]提供 I2C 在主发送模式下发送的 7 位从地址。 在 10 位寻址模式下 SADDR [9:0]提供 I2C 在主发送模式下发送的 10 位从地址。	3FFh
15:10	保留			0h

38.8.9 发送数据寄存器(I2CTXD)

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXD	R/W	发送数据（Transmit data） I2C 可以接受 1~8 位的数据字节。位的数量由 DBSEL 位决定。当写入少于 8 位的数据字节时，则该数据值为右对齐，TXD 的其它高位未定义。数据字节写入 TXD 后，I2C 将数据字节复制到传输移位寄存器。CPU 无法直接访问传输移位寄存器。当处于发送 FIFO 模式时，TXD 寄存器充当发送 FIFO 缓冲区。	0h
15:8	保留			0h

38.8.10 控制寄存器(I2CCTRL)

偏移地址：0x12

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	DBSEL	R/W	选择数据字节（Data byte select） I2C 要接收或发送的下一个数据字节中的位数。必须与另一个设备的数据大小匹配。当 DBSEL = 000 时，一个数据字节有 8 位。写入 TXD 的发送数据必须是右对齐。如果位数小于 8，则接收数据在 RXD[7:0]中为右对齐，其它高位未定义。 以下表示每数据字节分别有几位： 000：8 位 001：1 位 010：2 位 … 111：7 位	0h
3	FDFEN	R/W	使能自由数据格式模式（Free data format mode enable） 0：禁用，使用 7 位或 10 位寻址格式 1：使能，启用自由数据格式	0h

位/域	名称	R/W	描述	复位值
4	STABEN	R/W	使能开始字节模式（Start byte mode enable） 仅适用于 I2C 为主机时。START 字节可以帮助需要额外时间来检测 START 条件的从机。当 I2C 是从机时，会忽略来自主机的 START 字节。 0：禁止 1：使能。 当 STAGEN = 1 时，I2C 可以生成一个起始条件，一个起始字节（0000 0001b），一个虚拟应答时钟脉冲和重复 START 条件。然后正常工作，I2C 发送从设备地址及后续数据。	0h
5	I2CEN	R/W	I2C 模块复位控制位（I2C reset enable） 0：复位状态/禁用。清除此位时，所有的状态位都被设置为默认值 1：使能，工作状态。有释放 I2C 总线的效果	0h
6	DLBEN	R/W	使能数字环回模式（Digital loopback mode enable） 用于启用 I2C 模块内部数字环回模式，实现数据自发自收，适用于模块自检与通路链路调试。 0：禁止 1：使能（MSCFG=1） 发送时钟与接收时钟相同。在 SDA 引脚上传输的地址是 OADDR 中的地址。在此模式下，从 TXD 发出的数据经过 n 个延时周期后，通过一条内部路径被 RXD 接收，其中 n = ((I2C 输入时钟频率/模块时钟频率)× 8)	0h
7	REEN	R/W	使能重复模式（Repeat mode enable） 仅适用于 I2C 是主发送器或主接收器时。REEN，STOGEN 和 STAGEN 位决定了 I2C 模块开始和停止数据传输的时机。 0：非重复模式。I2CCNT 中的值决定 I2C 接收/发送的字节数 1：重复模式。在 STOGEN 位被手动设置前，每次 TXD 被写入时都会传输一个数据字节，会忽略 CNT 的值。通过 RAIFLG 位/中断可以确定 TXD 准备就绪的时机或数据已全部发送并且允许 CPU 写入 STOGEN 位。	0h
8	ADDRLEN	R/W	配置地址长度（Address length configure） 0：7 位寻址模式。I2C 发送 7 位从地址 1：10 位寻址模式。I2C 发送 10 位从地址	0h
9	RXTXCFG	R/W	配置收发模式（Receive transmit mode configure） 0：接收模式。I2C 是一个接收器，接收 SDA 引脚上的数据 1：发送模式。I2C 是一个发送器，通过 SDA 引脚发送数据	0h
10	MSCFG	R/W	配置主从模式（Master slave mode configure） 这是 I2C 模块主从模式的核心控制位，通过动态配置主/从角色实现总线控制权切换与时钟信号管理。当 I2C 主机产生 STOP 条件后自动切换为从模式以释放总线控制权。。 0：从模式。I2C 是从机，从主机接收串行时钟。 1：主模式。I2C 是主机，在 SCL 引脚上产生串行时钟。	0h

位/域	名称	R/W	描述	复位值
11	STOGEN	R/W	产生停止条件（stop condition generating） 仅适用于 I2C 为主机时。当 I2CEN = 0 时，该位不可写，需确保 I2C 处于工作状态。REEN，STOGEN 和 STAGEN 位决定了 I2C 模块开始和停止数据传输的时机，STOGEN 和 STAGEN 位可以终止重复模式。当处于非重复模式时，生成停止条件之前必须至少传输一个字节。直到设置 STODIFLG 位前，I2C 延迟清除该位。此位清除后再发起一条新消息可以避免中断 I2C 状态机。 0：产生停止条件后自动清除 1：当 I2C 内部计数器计数到 0 时产生停止条件	0h
12	保留			0h
13	STAGEN	R/W	产生开始条件（start condition generating） 仅适用于 I2C 为主机时。当 I2CEN = 0 时，该位不可写,。REEN，STOGEN 和 STAGEN 位决定了 I2C 模块开始和停止数据传输的时机，STOGEN 和 STAGEN 位可以终止重复模式。 0：产生开始条件后自动清除 1：在 I2C 总线上产生开始条件	0h
14	FREEEN	R/W	使能 I2C 自由运行（I2C free run enable） 0：当 I2C 为主机时：当发生断点时，如果 SCL 低，I2C 立即停止并保持低电平驱动 SCL。如果 SCL 高，I2C 模块等待，直到 SCL 变低，然后停止。 当 I2C 从机时：完成当前字节传输后，断点强制 I2C 停止。 1：断点触发后 I2C 自由运行	0h
15	NACKCFG	R/W	配置 NACK 模式（NACK mode configure） 仅适用于 I2C 作为接收器时。 0：主机-接收模式下：I2C 在每个应答周期内发送一个 ACK 位，直到内部计数器计数到 0，随后自动发送 NACK 终止传输。要使 NACK 位提前发送，必须设置 NACKCFG 位。 从机-接收模式：在每个字节传输后发送 ACK 信号，仅在软件强制置位 NACKCFG 时发送 NACK。 1：在从机-接收或主机-接收模式下：I2C 在总线上的下一个应答周期向发送端发送 NACK,随后自动清除 NACKCFG 位。 想要在下一个应答周期中发送 NACK 位，必须在最后一个数据位的上升沿之前完成置位，否则无法生效。	0h

38.8.11 中断源寄存器(I2CISRC)

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	IFLG	R	中断标志（Interrupt flag） 如果另一个较低优先级的中断处于挂起状态并使能，则将加载与该中断对应的值。否则，该值将保持清空状态。下面的中断事件按优先级降序列出。CPU 读操作可以清除此位。在仲裁丢失，检测到无应答条件或停止条件时，CPU 读操作还将清除 I2CSTS 寄存器中相关的位。 000：无 001：仲裁丢失中断	0h

位/域	名称	R/W	描述	复位值
			010：检测到无应答中断 011：寄存器访问就绪中断 100：接收数据准备就绪中断 101：发送数据准备就绪中断 110：检测到停止状态中断 111：被寻址为从机	
7:3	保留			0h
11:8	INTEST	R/W	内部测试（Internal testing） 保留的位应该总是写为零。	0h
15:12	保留			0h

38.8.12 扩展寄存器(I2CEXT)

偏移地址：0x16

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	BWCCFG	R/W	配置向后兼容模式（Backwards compatibility mode configure） 该模式位控制从机发送模式下 TXDIFLG 与 TXSNEFLG 状态位的时序。 0：单边沿同步 1：双边沿同步	1h
1	FWCCFG	R/W	配置向前兼容模式（Forward compatibility mode configure） 编程时这个位只在需要发送数据时才有发送请求的功能。为了避免影响 TXDIFLG 位，需要在 I2CEN = 0 后设置。 0：在缓冲区复制到移位寄存器或检测到总线起始条件（START）时，立即请求新数据。缓冲区的旧数据在非法启动、仲裁丢失、NACK 等异常等情况下被重用。 1：仅在 ACK 上请求数据的新功能是有效的。	0h
15:2	保留			0h

38.8.13 预分频寄存器(I2CPSC)

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	PSC	R/W	预分频值（Prescaler value） 决定 I2C 模块时钟，必须在 I2C 复位时初始化。 模块时钟频率= I2C 输入时钟频率/(PSC + 1)	0h
15:8	保留			0h

38.8.14 发送 FIFO 寄存器(I2CTXFIFO)

偏移地址：0x40

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	TXFFILEVEL	R/W	设置发送 FIFO 触发中断数据量（Transmit FIFO interrupt level set） 由于该设备上的 I2C 具有 16 级发送 FIFO，当 TXFFSTS≤TXFFILEVEL 时，发送 FIFO 中断标志位 TXFFIFLG 将置位。如果 TXFFIEN = 1，则会产生相应中断。可以根据实际应用场景，配置合适的 TXFFILEVEL 大小，确保及时填充新数据。	0h
5	TXFFIEN	R/W	使能发送 FIFO 中断（Transmit FIFO Interrupt Enable） 0：禁止 1：使能	0h
6	TXFFICLR	R/W	清除发送 FIFO 中断标志（Transmit FIFO Interrupt flag clear） 0：无效 1：清除 TXFFIFLG 位	0h
7	TXFFIFLG	R	发送 FIFO 中断标志（Transmit FIFO interrupt flag） 如果 TXFFIEN = 1，则该位置 1 时将产生中断。CPU 写 1 到 TXFFICLR 位可以清除此位。 0：未发生 1：发生	0h
12:8	TXFFSTS	R	发送 FIFO 的状态（Transmit FIFO status） I2C 模块退出复位时，发送 FIFO 状态位默认值为 00000，若此时 FIFO 操作已经启用，会立即触发 XTFFIFLG（发送 FIFO 中断标志），规避方法可以在退出复位后立即清除中断。 xxxxx：发送 FIFO 包含 xxxxx 字节,状态位显示当前 FIFO 中的字节数 00000：接收 FIFO 为空。	0h
13	TXFFEN	R/W	使能发送 FIFO（Transmit FIFO reset enable） 0：将发送 FIFO 指针复位为 0000，并将发送 FIFO 保持在复位状态 1：使能	0h
14	I2CFFEN	R/W	使能 I2C FIFO（I2C FIFO enable ） 使能此位才能使发送或接收 FIFO 正常工作。 0：禁止 1：使能	0h
15	保留			0h

38.8.15 接收 FIFO 寄存器(I2CRXFIFO)

偏移地址：0x42

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	RXFFILEVEL	R/W	设置接收 FIFO 触发中断数据量（Receive FIFO interrupt level set） 由于该设备上的 I2C 具有 16 级发送 FIFO，当 RXFFSTS≤RXFFILEVEL 时，接收 FIFO 中断标志位 RXFFIFLG 置 1。如果 RXFFIEN = 1，则会产生相应中断。为了避免由于这些位被复位为零，使能接收 FIFO 并且 I2C 从复位中取出，从未产生接收 FIFO 中断的情况，需要设置 RXFFEN 位之前，或者在相同的指令上修改这些位。	0h
5	RXFFIEN	R/W	使能接收 FIFO 中断（Receive FIFO Interrupt Enable） 0：禁止 1：使能	0h
6	RXFFICLR	R/W	清除接收 FIFO 中断标志（Receive FIFO Interrupt Flag Clear） 0：无效 1：清除 RXFFIFLG 位	0h
7	RXFFIFLG	R	接收 FIFO 中断标志（Receive FIFO interrupt flag） 如果 RXFFIEN = 1，则该位在置 1 时将产生中断。CPU 写 1 到 RXFFICLR 位可以清除此位。 0：未发生 1：发生	0h
12:8	RXFFSTS	R	接收 FIFO 的状态（ Receive FIFO status） xxxxx：接收 FIFO 包含 xxxxx 字节 00000：接收 FIFO 为空。	0h
13	RXFFEN	R/W	使能接收 FIFO（Receive FIFO enable） 0：将接收 FIFO 指针复位为 0000，并将接收 FIFO 保持在重置状态 1：使能	0h
15:14	保留			0h

39 电源管理总线（PMBus）

39.1 术语全称、缩写描述

表格 208 术语全称、缩写描述

中文全称	英文全称	英文缩写
电源管理总线	Power Management Bus	PMBus
系统管理总线	System Management Bus	SMBus
报文错误校验	Packet Error Checking	PEC
报文结束	End of Message	EOM

39.2 简介

PMBus 是一种基于 SMBus 的通信协议，用于微处理器和设备之间进行通信的接口，在物理层面上于 I2C 相似，该模块符合 SMI 论坛的 PMBus 规范第一部分 1.0 版和第二部分 1.1 版的要求。

39.3 主要特征

（1） 具备 PEC 功能，支持主从模式以及 I2C 模式

（2） 有四根信号线：ALERT、SCL、SDA、CONTROL

（3） 控制信号电平和时序、解析地址和缓冲数据

（4） 支持时钟高电平和低电平超时检测

（5） 带有发送和接收缓冲区，长达四字节

（6） 支持两种不同的通信速度，包括标准模式（最高 100kHz）和快速模式（最高 400kHz）

（7） 可处理主从模式下多种报文类型

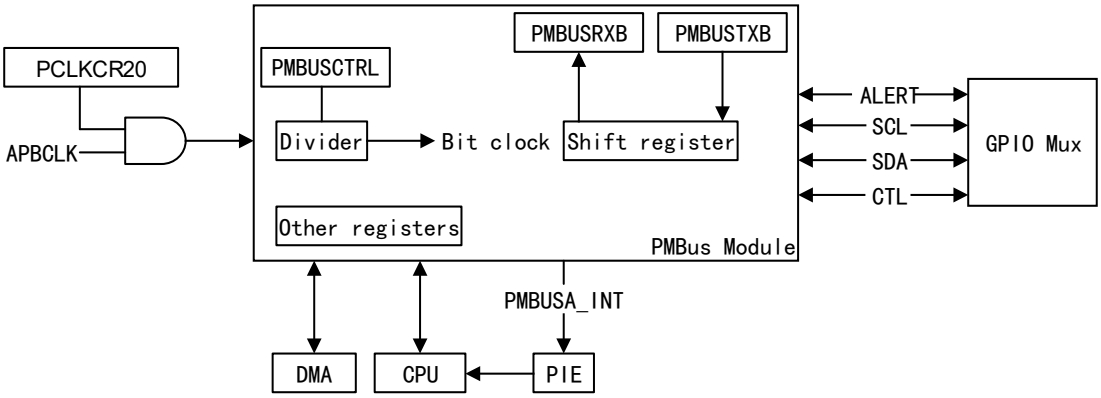
（8） 可配置为手动或自动确认设备地址和命令字节

（9） 报文传输结束发出 EOM 中断

（10） 触发可屏蔽中断的条件：
- 时钟低/高超时
- 总线空闲
- 发送缓冲区为空
- 警报输入断言
- 接收到数据就绪/从机地址
- 发出 EOM 中断

39.4 结构框图

图 216 PMBus 模块结构框图



该模块负责实现 PMBus 通信协议的底层部分。PMBus 模块包含以下四个信号线：

表格 209 PMBus 模块信号线

信号线	功能
ALERT	从机输出信号和主机输入信号，允许从机向主机发出关注的请求
SCL	总线时钟线，通常由主机控制，当从机需要更多时间来处理数据时，它可以暂时将 SCL 拉低，使得交易过程暂停
SDA	用于传输数据的线路，允许数据在主机和从机间双向流动
CONTROL	从机输入信号，可以关闭从机，也可以触发中断

39.5 功能描述

39.5.1 引脚以及模式配置

引脚配置

如果要将此外设连接到器件的引脚上，需要将 GPIO 引脚复用。有些 IO 功能是由 GPIO 寄存器设置定义的，该寄存器与当前外设无关。如果要避免引脚上产生毛刺，需要按照以下步骤进行配置：

（1） 设置 GPyGMUX 寄存器，此时对应的 GPyMUX 位域保持默认值零；

（2） 将 GPyMUX 寄存器设置为目标值。

如果要使输入信号不受时钟同步的影响，可以将 GPIO 输入限定配置为异步模式（GPxQSELn 寄存器相应的位设置为 0x11）；如果要配置内部上拉电阻，可以配置 GPyPUD 寄存器。有关 GPIO 复用和设置的细节，详情请参见 GPIO 章节。

主机模式配置

（1） 向 PMBUSCTRL [CLKDIV]位写入合适的时钟分频值，生成低于 10MHz 的位时钟频率；

（2） 置位 PMBUSCTRL[MASTEREN]位并清零 PMBUSCTRL [SLAVEEN]位，激活主机模式。

每次传输之前，PMBus 模块需要按照以下步骤进行配置和操作 PMBUSMCTRL 寄存器（在传输过程中，接收到的数据无需手动应答）：

表格 210 主机模式配置

配置	具体步骤
设置当前报文的从机地址	设置 SADDR 位为下一次传输设置正确的从机地址
使能 PEC 字节	总线要使用 PEC 时，设置 PECBYTEEN 位
配置命令代码使用的字节	设置 BYTECFG 位可以配置两个字节用于命令代码
使能在主启动报文上使用命令代码	设置 COMCEN 位可以在报文开始传输时发送一个命令字节
当前报文中传输的数据字节数	设置 BYTENUM 位确定要传输的数据字节数，其中不包括块长度字节（需要时自动生成）
使能发送群命令报文和使能发送调用报文	组命令过程中设置 GCOMEN 位使能发送群命令报文，过程调用过程中设置 TXPROMEN 位使能发送调用报文

将上述配置的选项值写入 PMBUSMCTRL 寄存器，启动传输。

从机模式配置

（1） 向 PMBUSCTRL[CLKDIV]位写入合适的时钟分频值，生成低于 10MHz 的位时钟频率；

（2） 置位 PMBUSCTRL[SLAVEEN]位，激活从机模式。

（3） 配置 PMBUSSCTRL 寄存器，定义与报文接收相关的参数：

表格 211 PMBUSSCTRL 寄存器配置

配置	具体步骤
设置从机地址和掩码	设置 SLAVEEN 位和 SMASKEN 位，配置从机接收报文时使用的地址和掩码
手动从机地址应答	使能 SADDRACKEN 位：允许由软件决定是否应答某个地址； 禁止 SADDRACKEN 位：根据预设的从机地址和掩码自动做出应答的决策
PEC 使能	如果使用 PEC，则设置 PECBYTEEN 位
手动命令字节应答	设置 COMC 位，允许由软件决定是否应答命令字节
自动应答的字节数	配置自动应答的接收缓冲区字节数，设置 AUTOACK 位，通常设置为最大值（除非需要更快地检测错误报文或不应答错误报文）

（1） 如果完整的报文长度（不包括地址）超过 4 个字节，则需要手动应答接收到的报文：

- 每接收到 4 个字节的数据包，通过向 PMBUSACK 寄存器写 1 手动应答（即使在自动应答模式下也是必要的）；
- PMBus 模块在等待应答信号时暂时拉低时钟线；
- 应答信号发出后模块会进一步拉低数据线并随后释放时钟线，以向总线主机发送应答信号；

（2） 如果完整的报文长度或报文的最后一部分不超过 4 字节或小于在 PMBUSSCTRL[AUTOACK]位中设置的限值，模块会自动应答，不需要写入 PMBUSACK 寄存器：

- 如果需要发送不应答信号，可以通过向 PMBUSACK 写 0（只能在模块等待应答时进行，在任何其他时候写零，会在接下来的报文中意外发送不应答信号造成干扰）。

39.5.2 主机模式的报文处理

39.5.2.1 Send Byte

设备地址，单个数据字节和可选的 PEC 字节构成了 Send Byte（发送字节），Send Byte 报文的步骤如下：

（1） 初始化 Send Byte 报文：将数据字节装载到 PMBUSTXB [7:0]位中，该数据字节发送到从机中；

（2） 设置设备地址，即配置 PMBUSMCTRL 寄存器。在编程地址时，如果将 PMBUSMCTRL[PECBYTEEN]位断言为高电平，则 PEC 字节可以在报文中发送；

（3） 在编程 PMBUSMCTRL 寄存器后，PMBus 模块将发送 Send Byte 报文；

（4） 报文发出后，模块发出 EOM 的中断，此时固件可以验证报文发送准确性；

（5） 收到结束中断的信号后，需要验证从机是否正确应答了发送的数据（通过读取 PMBUSSTS 寄存器）。

图 217 带 PEC 和不带 PEC 的 Send Byte 报文

S	Slave address	Wr	A	Byte #0	A	P

S	Slave address	Wr	A	Byte #0	A	PEC	A	P

39.5.2.2 Receive Byte

设备地址，单个数据字节和可选的 PEC 字节构成了 Receive Byte（接收字节）报文，Receive Byte 报文的步骤如下：

（1） 读取从机中的数据，对 Receive Byte 报文进行初始化；

（2） 编程 PMBUSMCTRL 寄存器，固件程序将 RW 位、设备地址以及可选的 PECBYTEEN 位编入，此时如果要指示报文类型，可以将 RW 位置为高电平，显示从从机发送到主机的数据；

（3） 在编程 PMBUSMCTRL 寄存器后，PMBus 模块发送 Receive Byte 报文；

（4） 报文发出后，模块发出 EOM 中断，此时固件可以验证报文的准确性；

（5） 收到结束报文中断后，需要确定 PMBUSRXB 寄存器是否有数据读取，以及验证从机地址是否正确确认。

接收来自于从机数据中应包含正确的 PEC 字节，如果该过程中断言了 PMBUSMCTRL[PECBYTEEN]位，那还需要检查 PMBUSSTS[PECVALID]位。

图 218 带 PEC 和不带 PEC 的 Receive Byte 报文

S	Slave address	Rd	A	Byte #0	NA	P

S	Slave address	Rd	A	Byte #0	A	PEC	NA	P

39.5.2.3 Write Byte 和 Write Word

设备地址、命令字节、被发送的数据字节和可选的 PEC 字节构成了 Write Byte（按字节写）和 Write Word（按字写），他们的发送过程与 Send Byte 协议类似，可以选择发送 1 个或 2 个字节（Write Byte 报文只发送 1 个字节，而 Write Word 报文可以发送 2 个字节）。Write Byte 和 Write Word 报文的写入步骤如下：

（1） 配置 PMBUSMCTRL 寄存器，如果要使能命令字节发送，需要置位可选的 PECBYTEEN 位以及 COMCEN 位；

（2） 使能命令字节发送后，固件将写入 PMBUSTXB 寄存器，其中位[7:0]写入发送给从机的命令字节，位[15:8]和位[23:16]写入数据字节（写入的格式与 Send Byte 协议不同）；

（3） 在编入 PMBUSMCTRL 寄存器后，PMBus 模块发送 Write Byte/Write Word 报文；

（4） 报文发出后，模块发出 EOM 的中断，此时固件可以验证报文的准确性，而 PMBUSSTS 寄存器会指示从机是否正确应答了报文。

图 219 带 PEC 和不带 PEC 的 Write Byte 和 Write Word 报文

S	Slave address	Wr	A	Command	A	Byte #0	A	P

S	Slave address	Wr	A	Command	A	Byte #0	A	PEC	A	P

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	P	

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	PEC	A	P

39.5.2.4 Read Byte 和 Read Word

设备地址、命令字节、接收从机发送的数据字节和可选的 PEC 字节构成了 Read Byte（按字节读）和 Read Word（按字读）报文。接收的报文是从机返回的，接

收过程与 Receive Byte 协议类似，可以选择接收 1 个或 2 个字节（Read Byte 报文接收 1 个字节，而 Read Word 报文协议支持接收 2 个字节），Read Byte 和 Read Word 报文的读取步骤如下：

（1） 配置 PMBUSMCTRL 寄存器，配置 PECBYTEEN 位读取附加到报文中的 PEC 字节（可以设置为接收或不接收 PEC 字节），并置位 COMCEN 位；

（2） 从机接收到报文后会进行处理，如果从机未能正确应答报文，或者从机在接收到预定数量的字节之后， PMBus 模块会自动终止报文；

此外，固件需要装载命令字节到 PMBUSRXB [7:0]位中，通过访问该寄存器获取从从机接收到的数据。

图 220 带 PEC 和不带 PEC 的 Read Byte 和 Read Word 报文

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	NA	P					
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	PEC	NA	P			
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P			
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	A	PEC	NA	P	

39.5.2.5 快速命令

在主模式下，配置 PMBUSMCTRL 寄存器可以触发快速命令，需要向该寄存器写入目标从机地址，发起快速命令的步骤如下：

（1） 配置字节计数 0 字节（向 PMBUSMCTRL [15:8]位写入全 0）；

（2） 发送设备地址，发送过程中，PMBus 模块会监视地址是否被从机应答：
- 如果从机未应答地址，PMBus 模块将终止报文，通过使能 PMBUSSTS[RXDFLG]位，然后在总线上触发发送一个停止状态；
- 如果从机应答地址，将向处理器发出数据请求；

（3） 接收到数据请求后，强制 PMBus 模块向总线写入一个停止状态来终止报文（固件向 PMBUSACK 位写入零）。

图 221 快速命令报文

S	Slave address	RW	A	P

39.5.2.6 扩展命令

扩展命令是在原命令的基础上，可以额外扩展 256 个命令码。PMBus 模块支持扩展命令，该命令可以添加到 Write Byte 和 Write Word，Read Byte 和 Read Word 协议中，带扩展命令的报文操作步骤如下：

（1） 将 PMBUSMCTRL[BYTECFG]位置为有效；

（2） 在扩展命令启用后，发送报文时会使用两个命令字节， 其中第二个命令字节不会单独发送，而是与后续数据字节合并在一起发送，在编程写数据或读报文的第一部分时，它们合并装载到 PMBUSTXB [15:8]位；

（3） 除了添加扩展命令外，Write Byte 和 Write Word 中需包含从机地址和重复起始条件，其他操作与 Send Byte 相同；

PMBus 模块会根据 BYTECFG 位的设置来调整协议的格式，此外还会自动处理 Write Byte 和 Write Word 中额外的字节（不需要固件支持）。.

图 222 带 PEC 和不带 PEC 的扩展命令 Read Byte 和 Read Word 报文

S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	NA	P				
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	PEC	NA	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	A	PEC	NA	P

图 223 带 PEC 和不带 PEC 的扩展命令 Write Byte 和 Write Word 报文

S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	P				
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	PEC	A	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	Byte #1	A	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	Byte #1	A	PEC	A	P

39.5.2.7 组命令

在组命令协议中，可以将多个设备的命令放在同一个报文中发送。发起组命令的操作步骤如下：

（1） 向 PMBUSMCTRL 写入报文，当报文中写入第一个设备的从机地址时，可以启用组命令模式（置位 PMBUSMCTRL[GCOMEN]位）；

（2） 报文的剩余部分依次写入，按照 Write Byte /Write Word 的报文方式；

（3） 组命令协议中报文被分成多个部分，每个部分对应一个设备的命令。当第一个设备的命令结束时，固件只要写入下一个设备地址，就会触发 PMBus 模块在总线上发送重复起始条件，并开始报文的下一部分。

（4） 固件在 GCOMEN 寄存器中写入报文的最后一个设备地址时，表示组命令结束，这时在写入 PMBUSMCTRL 寄存器之前禁用 GCOMEN 位。

注意：当总线上的设备接收到组命令报文时，它们会检测报文的结尾是否为停止状态。如果在报文的结尾检测到停止状态，设备将时执行接收到的命令，而不会等待其他设备的命令执行完毕。

图 224 带 PEC 和不带 PEC 的组命令报文

S	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A				
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A				
...														
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A	P			

S	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A				
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A				
...														
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A	PEC	A	P	

39.5.2.8 过程调用

Write Word 报文和 Read Word 报文构成了过程调用协议，该协议允许在单次通信过程中先写入数据然后紧接着读取数据。过程调用有一个可选功能：当从机发送读取数据给主机时，可以在数据之后附加一个 PEC 字节，验证接收到的数据是否准确。过程调用协议的执行步骤如下：

（1） 将命令字节装载到 PMBUSTXB [7:0]位，将数据字节装载到 23:8 位，完成 Write Word 报文；

（2） 写入 PMBUSMCTRL 寄存器，将 TXPROMEN 位设置为使能，过程调用报文开始发送，这时 PMBus 模块自动生成重复起始条件，并发起 Read Word 报文部分；

（3） 为了确定报文是否有效，当收到从机的应答和数据后，固件需等待 EOM 中断；

（4） 在接收到 EOM 时，通过 PMBUSSTS 寄存器可以查看从机对发送数据的应答状态，此外还可以指示是否接收 Read Word 部分的 2 个字节。如果需要判断 Read Word 部分接收来自从机的 PEC 字节是否正确，需要先使能了 PEC 处理，并通过 PMBUSSTS[PECVALID]位进行查看；

（5） 如果下一个要执行的操作不是过程调用报文，需要禁用 PMBUSMCTRL[TXPROMEN]位；

（6） 重复以上过程以执行下一条报文或其他操作。

注意：由于任意值写入 PMBUSMCTRL 寄存器都会触发报文，因此在固件发送新报文之前，请勿重新配置主机。

图 225 带 PEC 和不带 PEC 的过程调用报文

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	PEC	NA	P

39.5.2.9 Block Write

Block Write（按块写）协议用于发送超过 2 个数据字节的报文，并且可以在从机数据的结尾附加 PEC 字节。该协议结构上与 Write Word 协议相似，但包含一个额外的特性：命令字节后的首个数据字节用来指定随后的数据块长度。块长度表示从机在命令之后将接收到多少个数据字节，但不包括命令字节本身和块长度的首个数据字节。Block Write 的步骤如下：

（1） 发起 Block Write 报文，向 PMBUSMCTRL[BYTENUM]位写入块长度，其中当固件要发送的数据字节数超过 2 个时，PMBus 模块将自动向报文中插入块长度；

（2） 将初始的写数据装载到 PMBUSTXB 寄存器中，其中[7:0]位用来存放命令字节，然后存放块长度，剩余的 3 个字节用来存放块长度之后的前 3 个数据字节；

（3） 写入 PMBUSMCTRL 寄存器，发送 Block Write 报文：如果块长度超过 3 字节，PMBus 模块提供一个数据请求中断要求固件提供额外的数据字节。固件应遵循以下规则：

- 如果报文少于 4 个字节，则只将所需的数据字节写入 PMBUSTXB 寄存器；
- 如果报文超过 4 个字节，则使用全部 4 个字节写入 PMBUSTXB 寄存器；

（4） 报文完成后，PMBus 模块发出 EOM 中断，并且验证从机是否接受了写入数据块（检查 PMBUSSTS 寄存器）。

图 226 带 PEC 和不带 PEC 的 Block Write 报文

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	A	Byte #(N-1)	A	P	

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	A	Byte #(N-1)	A	PEC	A	P

39.5.2.10 Block Read

Block Read（按块读）协议可以从从设备中读取超过两个字节的数据，如果使能了 PEC 处理，则从机可以在报文的结尾附加一个 PEC 字节。Block Read 协议的结构类似于 Read Word 协议，从机发送的第一个数据字节表示从机正在写的数据的长度。Block Read 的步骤如下：

（1） 将发送给从机的命令字节写入 PMBUSTXB 的[7:0]位；

（2） 将块长度写入 PMBUSMCTRL[BYTENUM]位，发起 Block Read 报文，其中块长度计数不包含命令字节、任何从机地址及报文中的块长度字节；

（3） 发送 Block Read 报文，模块配置为在接收到 4 个数据字节后触发中断：

- 如果块长度是 3，EOM 中断与数据就绪中断同时接收；
- 如果块长度超过 3，由于固件每次可以读取的最大数据量是 4 个字节，因此只断言数据就绪中断；

（4） 检测到 EOM 中断后，固件可以验证接收到的 PEC。PMBUSRXB 寄存器存储的字节可能少于 4 个。通过读取 PMBUSSTS[RXBYTE]位的值，固件可以确定在最后一次数据传输中实际可用的字节数。

图 227 带 PEC 和不带 PEC 的 Block Read 报文

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	Byte #(N-1)	NA	P

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	Byte #(N-1)	A	PEC	NA	P

39.5.2.11 Block Write——Block Read 的过程调用

Block Write 和 Block Read 协议构成了 Block Write——Block Read 过程调用协议的，该协议允许在单次通信过程中先写入数据然后紧接着读取数据。Block Write——Block Read 过程调用的步骤如下：

（1） 主机进行 Block Write 操作，设置 PMBUSMCTRL[BYTENUM]位以指定写入的数据块的长度；

（2） 使能 PMBUSMCTRL[TXPROMEN]位发送数据；

（3） 在完成报文的 Block Write 后，PMBus 模块自动在总线上发出一个重复起始条件；

（4） 在重复起始条件之后，PMBus 模块开始进行发送 Block Read 报文，操作与 Block Read 报文相同，主机可以从从设备读取数据。

图 228 带 PEC 和不带 PEC 的 Block Write——Block Read 过程调用报文

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A

Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	NA	P	

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A

Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A	PEC	NA	P	

39.5.2.12 报警响应

在主机模式下，当主机检测到来自从机的警报条件时，会使用报警响应报文。该报文实质上是 Receive Byte 报文，不同之处是它专门为报警响应预留了地址，即将从设备地址设置为 0xC，并且禁用 PEC 功能。报警响应的步骤如下：

（1） 当 PMBus 模块在其输入端检测到警报条件时，它会通过中断信号通知固件，表明从机有一个警报条件的断言，也就是说，从机需要与主机通信；

（2） 为了发起报警响应报文，固件需要对 PMBUSMCTRL 寄存器进行编程，设置报警响应地址，并指定请求服务的从机地址；

（3） 在接收到 EOM 中断之后，固件可以从 PMBUSRXB 寄存器中读取到发送警报的从机地址。

图 229 报警响应报文

S	Alert response address	Rd	A	Slave address	NA	P

39.5.3 从机模式的报文处理

在从模式下，如果启用了自动地址和命令应答的最有效操作模式，并且开启了 PEC 功能，可以根据标志位来确定接收到的消息类型：

- 如果使能自动命令应答，则 DISDRDY 位设置为 1；
- 如果禁用自动地址应答，所有报文都需要手动设置 DISSADDRRDY 位为 1 启动报文发送。该模式下，读取和写入各有一个专用的命令。

如果报文没有 PEC，则它的有效数据长度会减少一个字节。例如，带有 PEC 的快速命令有一个字节，而没有 PEC 的快速命令则没有字节。

注意：

（1） 在特定条件下（即接收到停止信号、接收缓冲区已满或者发生故障），才会设置 PMBUSSTS 寄存器中的相应的位；

（2） 当接收到快速命令时，向 PMBUSACK 寄存器中写入 1 可以应答该报文；

（3） 在接收数据时，不会实时更新已经接收的字节数量。

39.5.3.1 **Send Byte**

设备地址、单个数据字节和可选的 PEC 字节构成了 Send Byte 报文。为了应答 PEC 字节，必须在 PMBUSSCTRL 寄存器中使能 PEC 处理，其中 PMBUSSTS[PECVALID]位表示接收到的 PEC 字节是否有效。应答模式分为以下两种：

- 自动地址应答模式：模块会自动应答数据字节和可选的 PEC 字节，无需固件干预。收到 Send Byte 报文后，模块产生一个 EOM 中断，读取并置位 PMBUSSTS[READDATA]；
- 手动模式：固件应答地址，模块应答剩余的数据字节和 PEC 字节。

以下是 PMBus 模块处理 Send Byte 报文的步骤：

（1） 在应答前，先从 PMBUSRXB 寄存器读取数据字节（接收到的第一个数据字节存储到 PMBUSRXB[7:0]位中）；

（2） 向 PMBUSACK 寄存器写 1，接收 Send Byte 报文；

（3） 当接收到 Send Byte 报文时，置位 PMBUSSTS [READDATA]位与 EOM。如果接收到 PEC 且有效，也将置位 PMBUSSTS[PECVALID]位；

（4） 读取 PMBUSSTS[RXBYTE]位：

- 如果没有使能 PEC 处理，PMBUSSTS[RXBYTE]将指示接收了一个字节；

- 如果使能 PEC 处理，当值为 2 时，则 PMBUSSTS[RXBYTE]指示接收了两个字节（数据字节和 PEC 字节），其中 PEC 字节存储到 PMBUSRXB [15:8]位中。

图 230 带 PEC 与不带 PEC 的 Send Byte 报文

S	Slave address	Wr	A	Byte #0	A	P

S	Slave address	Wr	A	Byte #0	A	PEC	A	P

39.5.3.2 Receive Byte

设备地址、单个数据字节和可选的 PEC 字节构成了 Receive Byte 报文。在自动地址应答模块下，处理 Receive Byte 报文的步骤如下：

（1） 检测到从机地址，并自动对主机进行地址确认；

（2） 固件接收到数据请求中断，将要发送到主机的数据字节存储在 PMBUSTXB [7:0]位中；

（3） 将 PMBUSSCTRL[VALBYTESEL]位设置为 1，发送一个字节。如果使能 PEC 处理，固件将同时设置 PMBUSSCTRL[TXPEC]位和 PMBUSSCTRL[PECPROEN]位；

（4） 如果使能了 PEC，从机在发送完数据字节后自动计算并添加 PEC 字节。

图 231 带 PEC 与不带 PEC 的 Receive Byte 报文

S	Slave address	Rd	A	Byte #0	NA	P

S	Slave address	Rd	A	Byte #0	A	PEC	NA	P

39.5.3.3 Write Byte 和 Write Word

从机地址、命令字、数据字节和可选的 PEC 字节构成了 Write Byte 和 Write Word 报文，其中 Write Byte 报文包含一个数据字节，Write Word 报文包含两个数据字节。

在自动地址应答模式下，模块自动应答数据字节和可选的 PEC 字节（无需固件干预），应答命令字需要配置 PMBUSSCTRL 寄存器。Write Byte 和 Write Word 处理报文的步骤如下：

（1） 对于 Write Byte 报文和不带 PEC 的 Write Word 报文：
- 当 PMBUSSTS[RXBYTE] 位的值为 3 时，表示接收到 3 个字节（命令字节、数据字节和 PEC 字节）；
- 发送结束，固件接收到 EOM 中断，此时读取并设置 PMBUSSTS[READDATA]位；
- 从 PMBUSRXB 寄存器读取数据，固件无需向主机发送应答。

（2） 对于带 PEC 字节的 Write Word 报文：

- 当 PMBUSSTS[RXBYTE] 位值为 4 时，表示接收到 4 个字节（命令字节、2 个数据字节和 PEC 字节）；
- 发送结束，使能 PMBUSSTS[READDATA]位中断；
- 固件从 PMBUSRXB 寄存器读取数据，写入 PMBUSACK 寄存器以应答主机；
- PMBus 模块保持 SCL 低电平，直到固件响应收到的数据。

图 232 带 PEC 与不带 PEC 的 Write Byte 和 Write Word 报文

S	Slave address	Wr	A	Command	A	Byte #0	A	P				
S	Slave address	Wr	A	Command	A	Byte #0	A	PEC	A	P		
S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	P		
S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	PEC	A	P

39.5.3.4 Read Byte 和 Read Word

从机地址、命令字、从从机地址接收的数据字节和可选的 PEC 字节构成了 Read Byte 和 Read Word 报文。响应 Read Byte 和 Read Word 报文的步骤如下：

（1） 配置 PMBUSSCTRL 寄存器以应答从机地址和命令字；

（2） 接收到命令字节后，将其存储于 PMBUSRXB [7:0]位中；

（3） 自动模式下，PMBus 模块在接收到重复起始条件和从机地址后，提供数据就绪中断和数据请求中断。

- 将 PMBUSSTS[RXBYTE]位设置为 1，触发数据就绪中断；

但此时模块无法区分即将接收的是组命令 Send Byte 报文还是 Read Byte 和 Read Word 报文：

- 如果使用相同的设备地址发送读取命令，断言数据请求位，表示从机应当发送数据到主机；
- 如果在接收到设备地址之前，数据已经写入 PMBUSTXB 寄存器，则不会断言数据请求位。如果固件希望能够响应 Send Byte，在接收到地址后，也应将 PMBUSSTS [RXBYTE]位设置为 1，为读取数据做准备；

（4） 发送完数据给主机后，等待并查看下一个事件是 EOM 还是数据请求：

- 如果事件是 EOM，作为 Send Byte 报文处理；
- 如果事件是数据请求，则处理为读取命令，其类型可能是 Read Byte、Word 或 Block；

（5） 识别为读取命令后，固件需进行响应：将数据写入 PMBUSTXB 寄存器，同时在 PMBUSSCTRL 寄存器中正确设置 PMBUSSCTRL[VALBYTESEL]位和 PEC 位（如果已经使能 PEC 处理）；

（6） 对于 Read Byte，设置发送字节计数为 1，如果要发送 PEC 字节，则置位 PMBUSSCTRL[TXPEC]位；

（7） 将所有字节同时写入 PMBUSTXB，启动发送；

（8） 主机接收到正确字节数的报文后，完成 Read Byte 报文，此时主机不会应答最后一个字节，然后置位 PMBUSSTS[ENDFLG]位通知固件；

注意：数据就绪位和数据请求位可能会在轮询间隔内变化，设计固件时应妥善处理这一情况。

图 233 带 PEC 与不带 PEC 的 Read Byte 和 Read Word 报文

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	NA	P				
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	PEC	NA	P		
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P		
S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	A	PEC	NA	P

39.5.3.5 快速命令

在从模式下，PMBus 模块接收到快速命令时，根据不同的模式对接收到的设备地址进行应答：

- 在自动地址应答模式下，它会自动对发送该命令的设备地址进行应答，无需固件干预。在接收到 EOM 后，固件可以选择读取 PMBUSHSADDR 寄存器，该寄存器包含了接收到的地址信息；
- 在手动地址应答模式下，固件需要写入 PMBUSACK 寄存器来应答地址。

图 234 快速命令报文

S	Slave address	RW	A	P

39.5.3.6 扩展命令

扩展命令是在原命令的基础上，额外扩展 256 个命令码。扩展命令中的数据字节和两个命令字节存储在 PMBUSRXB [15:0]位中。PMBus 模块处理扩展命令报文需要借助 PMBUSSTS[RXRESFLG]位和[RXBYTE]位。

（1） 处理扩展命令 Write Byte 和 Write Word 报文的步骤如下：

- 将两个命令字节存储在 PMBUSRXB [15:0]位中；
- 第一个命令字节（也就是命令代码中的第一个字节）必须含有一个特定的命令扩展代码，这标志着正在使用扩展命令协议；
- 一旦设备地址被重新发送，PMBUSSTS[RXRESFLG]位会被设置，表示一个扩展命令的开始；
- PMBUSSTS[RXBYTE]位用于区分写入操作的类型：
  - 如果其值等于 3，则代表接收的是扩展命令 Write Byte 报文；
  - 如果其值等于 4，代表接收的是扩展命令 Write Word 报文。

（2） 处理扩展命令 Read Byte 和 Read Word 报文的步骤如下：

- 当模块再次接收到设备地址后，它会触发数据就绪和数据请求中断，这准备发送数据给主机；
- 在 PMBUSRXB [15:0]位中，将可以找到两个命令字节；

- 将第一个命令字节会与命令扩展代码进行匹配，确保它是一个扩展命令；
- 为了完成读取操作并将数据返回给主机，固件必须加载要发送的数据。

图 235 带 PEC 和不带 PEC 的扩展命令 Write Byte 和 Write Word 报文

S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	P				
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	PEC	A	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	Byte #1	A	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Wr	A	Byte #0	A	Byte #1	A	PEC	A	P

图 236 带 PEC 和不带 PEC 的扩展命令 Read Byte 和 Read Word 报文

S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	NA	P				
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	PEC	NA	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P		
S	Slave address	Wr	A	Extended Command	A	Command	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	A	PEC	NA	P

39.5.3.7 组命令

PMBus 模块的组命令协议允许用户发送一条信息，同时向多个设备下达指令。发起组命令的步骤如下：

（1） 当设备检测到组命令报文末尾停止状态时，它们将协同执行接收到的指令；

（2） 在地址和命令得到应答后，如果模块检测到 4 个数据字节或者检测到总线上的重复起始信号，它会产生数据就绪中断；

（3） 接收到重复起始条件后，设置数据就绪位，将数据被读取到内存中；

（4） 等待 EOM 中断，因为所有的报文需要被完全接收，这是执行组命令的一个关键步骤；

（5） 设置 EOM 位后，固件对接收到的数据进行处理。

等待 EOM 中断是组命令与其他写入报文的唯一区别，在接收到组命令后，从机固件的其余处理流程与处理任何其他写入报文没有差别。

图 237 带 PEC 和不带 PEC 的组命令报文

S	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A
...										
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A	P

S	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A			
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A			
...													
Sr	Slave address	Wr	A	Command	A	Byte #0	A	...	Byte #N	A	PEC	A	P

39.5.3.8 过程调用

Write Word 报文和 Read Word 报文构成了过程调用协议，该协议允许在单次通信过程中先写入数据然后紧接着读取数据。过程调用协议的执行步骤如下：

（1） 配置 PMBUSSCTRL 寄存器以应答地址和命令；

（2） 在自动模式下，接收到重复起始条件和从机地址后，PMBus 模块提供数据就绪和数据请求中断；

（3） PMBUSSTS[RXRESFLG]被置位，表示已接收到过程调用的 Write Word 报文；

（4） 接收到的命令字节存储在 PMBUSRXB [7:0]位中，从主机接收到的两个数据字节存储在 PMBUSRXB [23:8]位中；

（5） 主机通过发送另一个重复起始信号紧跟着从机地址来发起读取请求；

（6） 接收到读取请求后，固件向 PMBUSTXB 写入 2 个数据字节，这些数据就绪发送回主机。如果使能了 PEC 处理，将设置 PMBUSSCTRL[TXPEC]位；

（7） 触发 EOM 中断，完成 Read Word 部分。

图 238 带 PEC 与不带 PEC 的过程调用报文

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	NA	P

S	Slave address	Wr	A	Command	A	Byte #0	A	Byte #1	A	Sr	Slave address	Rd	A	Byte #0	A	Byte #1	PEC	NA	P

39.5.3.9 **Block Write**

Block Write 协议可发送 2 个以上数据字节，结构与 Write Word 协议类似。Block

Write 协议的执行步骤如下：

（1） 接收到命令字节、块长度和 2 个数据字节后，PMBus 模块触发数据就绪中断，提示固件开始处理接收到的数据；

（2） 在等待固件读取并处理接收到的数据期间，模块通过将时钟线拉低暂停了总线，编程相应的应答寄存器。只有在固件处理完毕并准备好接收新数据时，总线才会恢复工作；

（3） 报文发送期间，接收的字节数存储于 PMBUSSTS 寄存器中。当累计接收到 4 个数据字节时，模块会再次触发数据就绪中断；

（4） 报文发送完成时，最后一组可能不足四个的数据字节会存入 PMBUSRXB 寄存器中；

（5） 固件检查 PMBUSSTS[PECVALID]位以确定接收到的 PEC 值是否正确。

图 239 带 PEC 与不带 PEC 的 Block Write 报文

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	A	Byte #(N-1)	A	P	

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	A	Byte #(N-1)	A	PEC	A	P	

39.5.3.10 Block Read

Block Read 协议用于发送超过 2 个数据字节的报文，结构上类似于 Read Word。Block Read 协议的执行步骤如下：

（1） 接收到重复起始条件和从机地址后，PMBus 模块产生数据就绪和数据请求中断。此时主机发送的命令字节被存放在 PMBUSRXB [7:0]位中；

（2） PMBus 模块会保持拉低 SCL 线，以暂停通信，直至固件准备发送数据；

（3） 固件向 PMBUSTXB 寄存器写入数据字节（第一次写入需要设置 PMBUSTXB [7:0]位以指定数据块长度）；

（4） 第一次数据发送时，PMBUSSCTRL [VALBYTESEL]位设置为 4，代表从机将发送四个字节的数据，但此时[TXPEC]位未置位，表示 PEC 字节不包括在此次发送中；

（5） 发送 4 个字节之后，固件等待数据请求并非 EOM，表示主机已经准备好接收更多的数据。此时模块会再次产生数据请求中断，并再次拉低 SCL，直到固件已经将更多数据写入 PMBUSTXB 寄存器中；

（6） 当发送最后四个或更少的数据字节时，固件会在 PMBUSSCTRL[VALBYTESEL]位中写入确切的字节数，并将这些字节写入到 PMBUSTXB 寄存器。此时固件置位[TXPEC]位，准备发送 PEC 字节；

（7） PMBus 模块将这些数据字节输出到总线上，然后是 PEC 字节。如果主机在接收到 PEC 后不进行应答，从机设置 EOM 位。

图 240 带 PEC 和不带 PEC 的 Block Read 报文

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	Byte #(N-1)	NA	P

S	Slave address	Wr	A	Command	A	Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	Byte #1	A	...	Byte #(N-1)	A	PEC	NA	P

39.5.3.11 Block Write——Block Read 的过程调用

Block-Write 和 Block Read 协议结合构成了 Block Write——Block Read 进程调用协议，允许在单次通信过程中先写入一段数据然后紧接着读取一段数据。该协议处理报文模式类似于过程调用报文的工作模式，Block Write——Block Read 过程调用的步骤如下：

（1） 配置 PMBUSSCTRL 寄存器以应答地址和命令；

（2） 在检测到 4 个数据字节或重复起始条件时，PMBus 模块生成数据就绪中断；

（3） 在接收到重复起始条件后，固件需要准备数据并加载到发送缓冲区以便发送给主机；

（4） 在准备发送数据时，固件必须在 PMBUSTXB [7:0]位中设置将要发送的数据块长度。

图 241 带 PEC 和不带 PEC 的 Block Write——Block Read 过程调用报文

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A

Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	NA	P	

S	Slave address	Wr	A	Command	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A

Sr	Slave address	Rd	A	BYTENUM=N	A	Byte #0	A	...	A	Byte #(N-1)	A	PEC	NA	P	

39.5.3.12 报警响应

当 PMBus 上的主机检测到从机的报警状态时，触发报警响应报文。处理报文的防止取决于 PMBus 模块处于自动地址确认模式还是手动地址确认模式：

（1） 自动地址应答模式：
- 检测到报警响应地址之后，PMBus 模块向主机发送应答信号；
- 将 PMBUSSCTRL 寄存器内编程的从机地址发送出去；
- 模块响应警报响应报文的前提是已经设置 PMBUSCTRL[ALERTLOW]位；
- 处理完报警响应报文后，模块清除报警状态，关闭 PMBUSCTRL 寄存器内的相关使能位。

（2） 手动地址应答模式：
- 固件读取 PMBUSRXB 寄存器已接收到的地址；
- 将目标从机地址发送给主机；

- 为了响应主机发起的警报响应报文，固件还必须重新配置 PMBUSCTRL 寄存器，以清除 ALERTLOW 位。

图 242 报警响应报文

S	Alert response address	Rd	A	Slave address	NA	P

39.6 寄存器组地址

表格 212 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
PmbusaRegs	PMBUS_REGS	0x5010 0800	0x5010 0BFF

39.7 寄存器地址映射

表格 213 PMBUS_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
PMBUSMCTRL	主机模式控制寄存器	0x00	√
PMBUSTXB	发送缓冲区寄存器	0x04	-
PMBUSRXB	接收缓冲区寄存器	0x08	-
PMBUSACK	应答寄存器	0x0C	-
PMBUSSTS	状态寄存器	0x10	-
PMBUSIMASK	中断屏蔽寄存器	0x14	√
PMBUSSCTRL	从机模式控制寄存器	0x18	√
PMBUSHSADDR	保持从机地址寄存器	0x1C	-
PMBUSCTRL	控制寄存器	0x20	√
PMBUSTIMCTRL	时序控制寄存器	0x24	√
PMBUSCLKTIM	时钟时序寄存器	0x28	√
PMBUSSTATIM	起始建立时间寄存器	0x2C	√
PMBUSBUSIDTIM	总线空闲时间寄存器	0x30	√
PMBUSCLKLTOTIM	时钟低电平超时值寄存器	0x34	√
PMBUSCLKHTOTIM	时钟高电平超时值寄存器	0x38	√

39.8 寄存器功能描述

39.8.1 主机模式控制寄存器（PMBUSMCTRL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	RWCFG	R/W	配置报文读写（Message Read and Write Configure） 0：写报文，主机发送数据给从机 1：读报文，从机发送数据给主机	0h
7:1	SADDR	R/W	当前报文的从机地址（Current Message Slave Address）	0h
15:8	BYTENUM	R/W	当前报文中传输的数据字节数（Number of Bytes of Data Transferred in the Current Message） 设备地址、命令字节或块报文中的块长度不在数据字节数的计数当中。通过 PMBus 接口进行字节写操作时，可以在单个报文中传输最多 255 个字节的数据。在块报文中，根据传输的数据字节数设置，PMBus 接口会自动插入响应的块长度到报文中，固件只需要加载传输地址、命令字节和数据。	0h
16	COMCEN	R/W	使能在主启动报文上使用命令代码（Use of Command Code on Master Initiated Messages Enable） 0：禁止 1：使能	0h
17	BYTECFG	R/W	配置命令代码使用的字节（ Use bytes for Command Code Configure） 0：1 字节 1：2 字节	0h
18	PECBYTEEN	R/W	使能 PEC 字节（PEC Byte Enables ） 0：禁止 PEC 处理 1：使能 PEC 发送/接收	0h
19	GCOMEN	R/W	使能发送群命令报文（Transmit of Group Command Message Enable） 0：报文默认状态（组命令报文除外） 1：使能发送组命令报文	0h
20	TXPROMEN	R/W	使能发送调用报文（Transmit Process Call Message Enable） 0：报文默认状态（过程调用报文除外） 1：使能发送过程调用报文	0h
31:21	保留			0h

39.8.2 发送缓冲区寄存器（PMBUSTXB）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	TXB	R/W	发送缓冲区（Transmit Buffer） [7:0]：字节 0（从发送缓冲区发送的第一个数据字节） [15:8]：字节 1（从发送缓冲区发送的第二个数据字节） [23:16]：字节 2（从发送缓冲区发送的第三个数据字节） [31:24]：字节 3（从发送缓冲区发送的最后一个数据字节）	0h

39.8.3 接收缓冲区寄存器（PMBUSRXB）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
31:0	RXB	R	接收缓冲区（Receive Buffer） [7:0]：字节 0（在接收缓冲区接收的第一个数据字节） [15:8]：字节 1（在接收缓冲区接收的第二个数据字节） [23:16]：字节 2（在接收缓冲区接收的第三个数据字节） [31:24]：字节 3（在接收缓冲区接收的最后一个数据字节）	0h

39.8.4 应答寄存器（PMBUSACK）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ACK	R/W	接收数据应答（Acknowledge Received Data） 0：接收数据未应答 1：接收数据应答（PMBus 发出应答信号该位清零）	0h
31:1	保留			0h

39.8.5 状态寄存器（PMBUSSTS）

偏移地址：0x10

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	RXBYTE	RC	接收到的字节（Receive Byte Selcet） 0：未接受到数据 1：接收到 1 个字节的数据（数据位于 LIN_RXB 寄存器[7:0]位） 2：接收到 2 个字节的数据（数据位于 LIN_RXB 寄存器[15:0]位） 3：接收到 3 个字节的数据（数据位于 LIN_RXB 寄存器[23:0]位） 4：接收到 4 个字节的数据（数据位于 LIN_RXB 寄存器[31:0]位）	0h
3	READDATA	RC	总线活动前读取数据（Read Data Prior to Bus Activity） 0：没有可供处理器读取的数据 1：读取数据，因为 PMBus 接口读取缓冲区已满。为了等待固件读取数据，PMBus 接口使能时钟拉伸功能来暂停总线上的其他活动	0h
4	ADITDATA	RC	请求附加数据（Request Additional Data） 0：不需要额外的数据 1：请求额外的数据，PMBus 接口使能时钟拉伸功能来暂停总线上的其他活动	0h
5	ENDFLG	RC	当前报文结束标志（Current Message End Flag） 0：PMBus 处于空闲状态或报文发送中 1：报文发送结束	0h
6	RXDFLG	RC	已接收数据标志（Receive Data Flag） 0：已接收 1：未接收	0h

位/域	名称	R/W	描述	复位值
7	PECVALID	RC	接收到的 PEC 有效（Received PEC is valid）. 0：无效（前提是已断言 EOM） 1：有效 注意：该位仅在 EOM 后才有效，报文发送期间忽略该位状态。	0h
8	CLKLTOFLG	RC	时钟低超时标志（Clock Low Timeout Flag） 0：未检测到时钟低超时标志 1：检测到时钟低超时标志，并且时钟低超时保持时间大于 35ms	0h
9	CLKHFLG	RC	时钟高标志（Clock High Flag） 0：未检测到时钟高标志 1：报文发送期间，时钟高超过了 50μs	0h
10	RDYRSADDR	RC	准备读取从机地址（Ready Read Slave Address） 0：无从机地址可以读取 1：准备从 LIN_RXB 寄存器 6:0 位中读取从机地址	0h
11	RXRESFLG	RC	接收到重复启动标志（Received Repeated Star Flag） 0：未接收 1：已接收	0h
12	BUSYFLG	RC	忙标志（Busy Flag） 0：PMBus 接口空闲，准备收发报文 1：PMBus 接口忙，处理当前报文	0h
13	AVAFLG	RC	可用标志（Available Flag） 0：不可用，PMBus 处理当前报文 1：可用于处理新报文	0h
14	LOSTCTRL	RC	主机失去对 PMBus 的控制（Master Lost Control of PMBus） 0：主机可控制 PMBus 1：主机仲裁失败，不可控制 PMBus	0h
15	MASTERFLG	RC	主模式标志（Master Mode FLag） 0：处于从模式或空闲模式 1：处于主模式	0h
16	ALERTTRAN	RC	警报引脚转换（Alert Pin Transitioned） 0：没有转换 1：已转换，由 PMBus 上的另一设备断言	0h
17	CTRLTRAN	RC	控制引脚转换（Control Pin Transitioned） 0：没有转换 1：已转换，由 PMBus 上的另一设备断言	0h
18	ALERTOLL	R	警报引脚在高逻辑电平（Alert Pin at Logic Level High） 0：在低电平时观察到该引脚 1：在高电平时观察到该引脚	1h
19	CTRLOLL	R	控制引脚在高逻辑电平（Control Pin at Logic Level High） 0：在低逻辑电平时观察到该引脚 1：在高逻辑电平时观察到该引脚	0h

位/域	名称	R/W	描述	复位值
20	SDAOLL	R	数据引脚在高逻辑电平（Data Pin at Logic Level High） 0：在低逻辑电平时观察到该引脚 1：在高逻辑电平时观察到该引脚	1h
21	SCLOLL	R	时钟引脚在高逻辑电平（Clock Pin at Logic Level High） 0：在低逻辑电平时观察到该引脚 1：在高逻辑电平时观察到该引脚	1h
31:22	保留			0h

39.8.6 中断屏蔽寄存器（PMBUSIMASK）

偏移地址：0x14

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	DISBUSIDLE	R/W	禁止总线空闲时产生中断（Disables Bus Idle Generation Interrupt） 0：断言总线空闲时产生中断 1：断言总线空闲时禁止产生中断	1h
1	DISCLKLTO	R/W	禁止时钟低超时时产生中断（Disables Clock Low Timeout Generation Interrupt） 0：断言时钟低超时标志时产生中断 1：断言时钟低超时标志时禁止产生中断	1h
2	DISDRDY	R/W	禁止数据就绪时产生中断（Disables Data Ready Generation Interrupt） 0：断言数据就绪时产生中断 1：断言数据就绪时禁止产生中断	1h
3	DISDREQ	R/W	禁止数据请求时产生中断（Disables Data Request Generation Interrupt） 0：断言数据请求时产生中断 1：断言数据请求时禁止产生中断	1h
4	DISSADDRRDY	R/W	禁止从机地址就绪时产生中断（Disables Slave Address Ready Generation Interrupt） 0：断言从机地址就绪时产生中断 1：断言从机地址就绪时禁止产生中断	1h
5	DISEND	R/W	禁止报文结束时产生中断（Disables Message End Generation Interrupt） 0：断言报文结束时产生中断 1：断言报文结束时禁止产生中断	1h
6	DISALERT	R/W	禁止警报时产生中断（Disables Alert Generation Interrupt） 0：断言警报时产生中断 1：断言警报时禁止产生中断	1h
7	DISCTRL	R/W	禁止控制时产生中断（Disables Control Generation Interrupt） 0：断言控制时产生中断 1：断言控制时禁止产生中断	1h

位/域	名称	R/W	描述	复位值
8	DISLOSTARB	R/W	禁止失去仲裁时产生中断（Disables Lost Arbitration Generation Interrupt） 0：断言失去仲裁时产生中断 1：断言失去仲裁时禁止产生中断	1h
9	DISCLKH	R/W	禁止时钟高时产生中断（Disables Clock High Generation Interrupt） 0：断言时钟高时产生中断 1：断言时钟高时禁止产生中断	1h
31:10	保留			0h

39.8.7 从机模式控制寄存器（PMBUSSCTRL）

偏移地址：0x18

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
6:0	SDEVADDR	R/W	设置从机的当前设备地址（Slave Current Device Address Set） 从机设置为默认模式，即自动从机地址应答模式。在接收到设备地址后，PMBus 接口将经过掩码过滤后是接收地址与从机预设地址相比较，如果匹配从机会自动应答。	7Ch
7	SADDRACKEN	R/W	使能手动确认从机地址（Manual Slave Address Acknowledgement Enable） 0：从机自动应答 SDEVADDR 位指定的设备地址 1：手动设置从机应答地址，尤其是当 PMBUSCTRL[MODECFG]位为 1 时。该模式下固件需要检查接收到的地址并对报文进行确认。	0h
14:8	SMASKEN	R/W	使能从机掩码（Slave Mask Enable） 检测地址时，从机使用掩码可以对多个设备地址进行确认。 （1）上电后从机掩码默认为 7Fh，此时只有当接收地址与从机地址的[6:0]位完全一致时，从机才会应答； （2）掩码所有位为 0 时，可应答任意设备地址； （3）掩码中任意一位为 0 时，在该位上从机地址为 0/1 都可以应答。	7Fh
15	PECPROEN	R/W	使能 PEC 处理（PEC Processing Enable） 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
18:16	VALBYTESEL	R/W	选择有效字节（Valid Bytes Selcet） 000：无有效字节 001：一个有效字节，LIN_TXB 寄存器[7:0]位：Byte#0 010：两个有效字节，LIN_TXB 寄存器[15:0]位：Byte#0 和 Byte#1 011：三个有效字节，LIN_TXB 寄存器[23:0]位：Byte#0、Byte#1 和 Byte#2 100：四个有效字节，LIN_TXB 寄存器[31:0]位：Byte#0 、Byte#1、Byte#2 和 Byte#3	0h
19	TXPEC	R/W	在报文末尾发送 PEC 字节（Transmit a PEC Byte at the End of the Message） 该位置为有效时将在报文末尾发送 PEC 字节。PMBus 接口根据 VALBYTESEL 位指定的数据字节数进行发送，发送完毕后再发送 PEC 字节。	0h
20	COMC	R/W	收到命令代码后生成的数据请求标志（Data Request Flag Generated after Receive of Command Code） 0：接收到命令代码后从机自动应答 1：接收到命令代码后从机生成数据请求标志，等待固件发出应答信号，才可以继续报文发送	0h
22:21	AUTOACK	R/W	配置从模式自动确认的数据字节（slave mode automatic acknowledge data bytes Configure） 00：从机接收到 1 个字节，硬件不自动应答接收字节，固件手动应答每个接收到的字节 01：从机接收到 2 个字节，硬件自动应答接收到的第一个字节，固件手动应答第二个字节 10：从机接收到 3 个字节，硬件自动应答接收到的前两个字节，固件手动应答第三个字节 11：从机接收到 4 个字节，硬件自动应答接收到的前三个字节，固件手动应答第四个字节	3h
31:23	保留			0h

39.8.8 保持从机地址寄存器（PMBUSHSADDR）

偏移地址：0x1C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ADDRRW	R	地址存储读写位（Address Stored R/W Bit ） 0：写访问 1：读访问	0h
7:1	DEVADDR	R	存储设备地址（Stored Device Address ）	0h
31:8	保留			0h

39.8.9 控制寄存器（PMBUSCTRL）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	STSMRSTEN	R/W	使能状态机复位（State Machines Reset Enable） 0：默认状态，状态机不复位 1：重置状态机为初始状态 注意：软件重置不会清除 PMBUSSTS 寄存器，需要手动读取该寄存器才可以清除。	0h
1	ALERTLOW	R/W	PMBus 报警被从机压低（PMBus Alert Driven Low By Slave） 0：PMBus 报警通过上拉电阻保持在高电平状态，不受从机控制 1：PMBus 报警被从机压低到低电平	0h
2	CLKLTOCFG	R/W	配置时钟低超时的中断产生（Clock Low Timeout Generated Interrupt Configure） 0：时钟低超时的上升沿产生中断 1：时钟低超时的下降沿产生中断	0h
3	FASTEN	R/W	使能快速模式（Fast Mode Enable ） 0：使能标准 100KHz 模式 1：使能快速 400KHz 模式	0h
4	保留			0h
5	CTRLCFG	R/W	配置控制的中断产生（Control Generated Interrupt Configure） 0：控制的上升沿产生中断 1：控制的下降沿产生中断	0h
6	ALERTCFG	R/W	配置警报引脚（Alert Pin Configure） 0：配置为功能模式 1：配置为 GPIO 引脚	0h
7	ALERTDRVCFG	R/W	配置 GPIO 模式的警报引脚驱动（GPIO Mode Alert Pin Driven Configure） 0：低电平 1：高电平	0h
8	ALERTDIRCFG	R/W	配置警报引脚方向（Alert Pin Direction Configure ）. 0：输出 1：输入	0h
9	PINCFG	R/W	配置控制引脚（Control Pin Configure） 0：默认功能模式 1：GPIO 引脚	0h
10	CTRLDRVCFG	R/W	配置 GPIO 模式的控制引脚驱动（GPIO Mode Control Pin Driven Configure） 0：低电平 1：高电平	0h
11	CTRLDIRCFG	R/W	配置控制引脚方向（Control Pin Direction Configure） 0：输出 1：输入	0h

位/域	名称	R/W	描述	复位值
12	SDACFG	R/W	配置数据引脚（Data Pin Configure） 0：低电平 1：高电平	0h
13	SDADRVCFG	R/W	配置 GPIO 模式的数据引脚驱动（GPIO Mode Data Pin Driven Configure） 0：低电平 1：高电平	0h
14	SDADIRCFG	R/W	配置数据引脚方向（Data Pin Direction Configure） 0：输出 1：输入	0h
15	SCLCFG	R/W	配置时钟引脚（Clock Pin Configure） 0：功能模式 1：GPIO 引脚	0h
16	SCLDRVCFG	R/W	配置 GPIO 模式的时钟引脚驱动（GPIO Mode Clock Pin Driven Configure） 0：低电平 1：高电平	0h
17	SCLDIRCFG	R/W	配置时钟引脚方向（Clock Pin Direction Configure） 0：输出 1：输入	0h
18	THRUENA	R/W	使能电流源 PMBus 地址检测通过 ADC（Current Source for PMBus Address Detection thru ADC Enable） 0：禁止 1：使能	0h
19	THRUENB	R/W	使能电流源 PMBus 地址检测通过 ADC（Current Source for PMBus Address Detection thru ADC Enable） 0：禁止 1：使能	0h
20	DISCLKLTO	R/W	禁止时钟低超时（Clock Low Timeout Disabled） 0：使能 1：禁止	0h
21	SLAVEEN	R/W	使能 PMBus 从机（PMBus Slave Enables） 0：禁止 1：使能	1h
22	MASTEREN	R/W	使能 PMBus 主机（PMBus Master Enables） 0：禁止 1：使能	0h
27:23	CLKDIV	R/W	SYSCLK 时钟分频（SYSCLK Clock Divider） PMBus 发送/接收 FMS 的时钟（FSMCLK）是通过将 SYSLCK 分频得到，其中 FMSCLK≤10MHz，公式如下： FMSCLK 频率 = SYSCLK 频率 /（CLKDIV+1）	0h

位/域	名称	R/W	描述	复位值
30:28	保留			0h
31	MODECFG	R/W	配置模式（Mode Configure） 0：PMBus 模式 1：I2C 模式	0h

39.8.10 时序控制寄存器（PMBUSTIMCTRL）

偏移地址：0x24

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	PARCFG	R/W	设置 FSM 参数（FSM Parameters Configure） 当该位设置为 1 时，FSM 使用 PMBUSCLKTIM、PMBUSSTATIM、PMBUSBUSIDTIM、PMBUSCLKLTOTIM 和 PMBUSCLKHTOTIM 寄存器的设置；默认情况下，FSM 使用预设的定时参数。	0h
31:1	保留			0h

39.8.11 时钟时序寄存器（PMBUSCLKTIM）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	CLKHIGHP	R/W	设置 FSM 输入时钟在 PMBus 主时钟高脉冲中的个数（Number of FSM Input Clock in the PMBUS Master Clock High Pulse Set） 一个时钟高脉冲中 FSM 时钟的个数 = (CLKHIGHP +3)	2Fh
15:8	保留			0h
23:16	CLKTIM	R/W	设置 PMBus 主时钟周期内 FSM 输入时钟的个数（Number of FSM Input Clock in the PMBUS Master Clock Period） 一个时钟周期内的 FSM 时钟数量 = (CLKTIM +4)	60h
31:24	保留			0h

39.8.12 起始建立时间寄存器（PMBUSSTATIM）

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	STATIM	R/W	设置 PMBus 主时钟最后一个上升沿到下一个开始沿之间的时间（The Time between the Last Rising Edge of the PMBus Master Clock and the Next Starting Edge） 该位定义了 PMBus FSM 的时钟周期。	2Fh
31:8	保留			0h

39.8.13 总线空闲时间寄存器（PMBUSBUSIDTIM）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	BUSIDTIM	R/W	设置总线空闲时间（Bus Idle Time Set） 该位的值表示连续逻辑 1 的持续时间，单位是 PMBus 的时钟周期数。当 PMBus 的时钟和数据线都保持逻辑 1 状态一段时间后，可以认定总线处于空闲状态。	1F3h
31:10	保留			0h

39.8.14 时钟低电平超时值寄存器（PMBUSCLKLTOTIM）

偏移地址：0x34

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
19:0	CLKLTOTIM	R/W	设置时钟低超时时间（Clock Low Timeout Time） 该位表示连续低电平的持续时间，单位是 PMBus FSM 的时钟周期数。当 PMBus 的时钟线保持低电平的时间超过 CLKLTOTIM 值时，将会发生时钟低超时。	0005 572Fh
31:20	保留			0h

39.8.15 时钟高电平超时值寄存器寄存器（PMBUSCLKHTOTIM）

偏移地址：0x38

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
9:0	CLKHTOTIM	R/W	设置时钟高超时时间（Clock High Timeout Time） 该位表示连续高电平的持续时间，单位是 PMBus FSM 的时钟周期数。当 PMBus 的时钟线保持低电平的时间超过 CLKHTOTIM 值时，将会发生时钟高超时。	1F3h
31:10	保留			0h

40 控制器局域网（CAN）

40.1 术语全称、缩写描述

表格 214 术语全称、缩写描述

中文全称	英文全称	英文缩写
数据长度码	Data Length Code	DLC
总线关闭状态	Bus-off State	Bus-off
总线自动开启	Auto-Bus-on	ABO
位流处理器	Bit Stream Processor	BSP
模 2 求和	Mod-2 sum	-
信息处理时间	Information Processing Time	IPT

40.2 简介

CAN 是控制器局域网络 Controller Area Network 的缩写，是一种串行通信协议，符合 ISO11898-1（CAN 2.0B）协议规范。CAN 模块支持的比特率高达 1 Mbit/s，支持具有高可靠性的分布式实时控制。

40.3 主要特征

（1） 通信比特率最高为 1Mbps

（2） 符合 ISO11898-1 协议规范（CAN 协议 2.0A 和 2.0B）

（3） 两个中断线：CANINT0 和 CANINT1

（4） 支持 DMA 功能

（5） 支持多种时钟源

（6） 支持软件模块复位

（7） 支持报文 RAM 奇偶校验机制

（8） 支持挂起模式，用于调试操作

（9） 支持可编程环回模式，用于自检操作

（10） 进入 Bus-off 状态后，由 32 位可编程定时器自动恢复到 bus-on

（11） 32 个邮箱，每个邮箱的特性如下：

- 支持发送或接收功能
- 支持数据帧和远程帧两种帧类型
- 支持 11 位标准标识符或 29 位扩展标识符，标识符接收掩码可配置
- 可容纳 0-8 个字节的数据

- 支持数据 RAM 和奇偶校验配置
- 每个邮箱都有一个独立的标识符掩码
- 支持可编程的 FIFO 模式

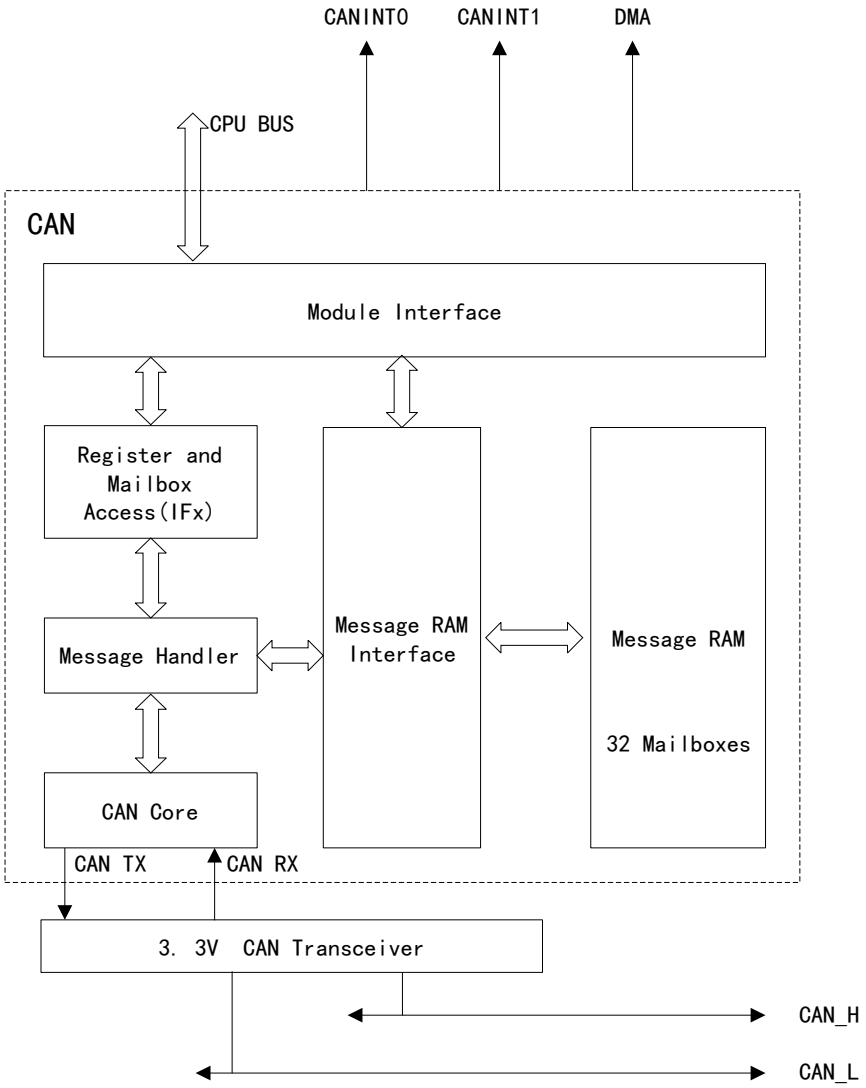
注意：

（1） 根据使用的时序设置，片上内置晶振的精度可能不符合 CAN 协议的要求，此时必须使用外部时钟源，片上内置晶振的精度详情请参见数据手册。

（2） 对于 100 MHz 的 CAN 位时钟，可用的最小位速率为 3.90625 kbps。

40.4 结构框图

图 243 CAN 结构框图

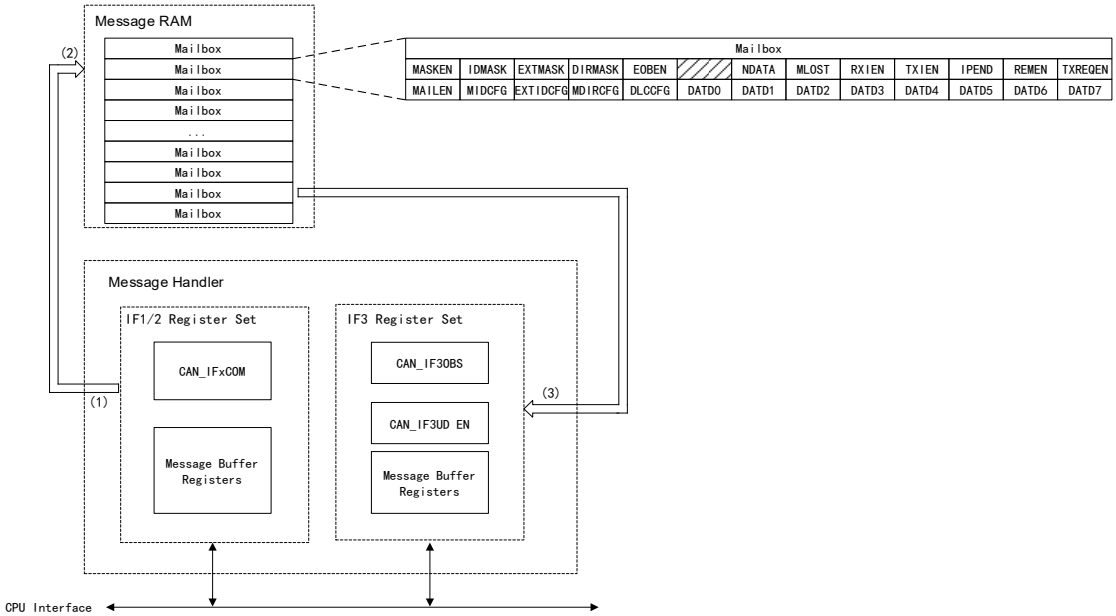


40.4.1 内部结构

表格 215 CAN 内部结构

内部结构	功能
CAN 内核	由 Rx/Tx 移位寄存器和 CAN 协议控制器组成，负责处理所有与 ISO 11898-1 协议相关的功能
报文 RAM	存储 32 条报文。在专用测试模式下，消息 RAM 被映射到内存中，因此可以直接访问。
报文处理器	(1)控制单端口报文 RAM 和 CAN 内核的 Rx/Tx 移位寄存器之间的数据传输的状态机 (2)根据控制寄存器中的配置，处理接受过滤和产生相应的中断请求信号
寄存器和邮箱访问（IFx）	正常运行时，所有 CPU 和 DMA 通过 IFx 对报文 RAM 的访问进行间接访问，以确保数据在读写过程中的一致性。IFx 寄存器可以被看作间接访问邮箱的窗口。CPU 对报文 RAM 的读写操作由三组接口寄存器控制，IF1 和 IF2 寄存器组可以进行读写操作，IF3 寄存器组仅用于读操作。IFx 的字长与报文 RAM 相同。

图 244 通过 IFx 寄存器间接访问邮箱



访问步骤如下：

（1） 在 IF1/2 寄存器组的 CAN_IFxCOM 寄存器中选择传输方向和待传输的寄存器；

（2） 在 IF1/2 寄存器组的 CAN_IFxCOM 寄存器中选择传输的邮箱并进行传输

（3） 当使能自动更新操作时，通过 IF3 寄存器组的 CAN_IF3OBS 和 CAN_IF3UDEN 寄存器进行自动更新

40.5 功能描述

如结构框图所示，CAN 模块通过一个 CAN 收发器芯片连接到 CAN 总线（物理层）。

可以为 CAN 网络上的通信配置独立的邮箱，这些邮箱及其标识符掩码被存储在报文 RAM 中。

报文处理器负责执行所有与报文处理相关的功能，包括：

- 处理传输/DMA 请求，并产生中断
- 在 CAN 内核和报文 RAM 之间传输报文
- 接收过滤

CAN 内核和报文处理器的控制和配置，访问报文 RAM 的功能在 CAN 寄存器组中实现，CPU 可以通过模块接口直接访问 CAN 寄存器组。

40.5.1 引脚配置

通过配置 GPIO 章节的复用寄存器，选择对应的引脚将 CAN 与设备连接。必须按照 GPIO 章节中正确的步骤配置 GPIO 复用寄存器，即先配置 GPxGMUX 位段（此时对应的 GPxMUX 位段为默认值 0），再向 GPxGMUX 位段写入所需值，以避免引脚上产生毛刺。GPIO 寄存器中配置了一部分独立于 CAN 的 IO 功能。在 GPxPUD 寄存器中可以配置内部上拉电阻的状态。对于输入信号，可以通过配置对应的 GPxQSEL1/2[GPIOn]=11，将 GPIO 引脚配置为异步输入。更多 GPIO 复用的相关信息，详情请参见 GPIO 章节。

40.5.2 寻址方案

一般情况下，建议使用含有__byte_peripheral_32() 函数的 HWREG_BP() 指令对 CAN 寄存器进行 32 位访问，但 CAN 其实也支持 16 位的字节访问，只要将低字节写入寄存器地址，将高字节写入“寄存器地址+2”的地址就能实现。由于总线桥接的存在，通过 Keil 等 IDE 内存窗口查看 CAN 模块寄存器空间时，所见的地址可能与实际的硬件寻址并不一致。

- 当查看模式为 32 位或 16 位时，忽略奇地址，复制偶地址；
- 当查看模式为 8 位时，来自 CAN 模块内部的偶地址会被复制到 CCS 内存窗口的奇数地址中，模块中的奇地址则不会显示。

表格 216 软件访问 CAN 寄存器

软件访问			CAN 寄存器空间		
格式	访问	数据	名称	地址	数据
8 位	__byte((int *)0x00,0)	0x0000	CAN_CTRL	0x00	0x33221100
	__byte((int *)0x01,0)	0x0011	CAN_EFLG	0x04	0x33221100
	__byte((int *)0x02,0)	0x0022	CAN_ECNT	0x08	0x33221100
	__byte((int *)0x03,0)	0x0033	CAN_BTIM	0x0C	0x33221100
	__byte((int *)0x04,0)	0x0044			

软件访问			CAN 寄存器空间		
	__byte((int *)0x05,0)	0x0055			
	__byte((int *)0x06,0)	0x0066			
	__byte((int *)0x07,0)	0x0077			
	__byte((int *)0x08,0)	0x0088			
	__byte((int *)0x09,0)	0x0099			
	__byte((int *)0x0A,0)	0x00AA			
	__byte((int *)0x0B,0)	0x00BB			
	__byte((int *)0x0C,0)	0x00CC			
	__byte((int *)0x0D,0)	0x00DD			
	__byte((int *)0x0E,0)	0x00EE			
	__byte((int *)0x0F,0)	0x00FF			
16 位	(*((short *)(0x00)))	0x1100	CAN_CTRL	0x00	0x33221100
	(*((short *)(0x01)))	0x1111	CAN_EFLG	0x04	0x33221100
	(*((short *)(0x02)))	0x3322	CAN_ECNT	0x08	0x33221100
	(*((short *)(0x03)))	0x3322	CAN_BTIM	0x0C	0x33221100
	(*((short *)(0x04)))	0x5544			
	(*((short *)(0x05)))	0x5544			
	(*((short *)(0x06)))	0x7766			
	(*((short *)(0x07)))	0x7766			
	(*((short *)(0x08)))	0x9988			
	(*((short *)(0x09)))	0x9988			
	(*((short *)(0x0A)))	0xBBAA			
	(*((short *)(0x0B)))	0xBBAA			
	(*((short *)(0x0C)))	0xDDCC			
	(*((short *)(0x0D)))	0xDDCC			
	(*((short *)(0x0E)))	0xFFEE			
	(*((short *)(0x0F)))	0xFFEE			
32 位	(*((long *)(0x00)))	0x33221100	CAN_CTRL	0x00	0x33221100
	(*((long *)(0x01)))	0x33221100	CAN_EFLG	0x04	0x33221100
	(*((long *)(0x02)))	0x33221100	CAN_ECNT	0x08	0x33221100
	(*((long *)(0x03)))	0x33221100	CAN_BTIM	0x0C	0x33221100
	(*((long *)(0x04)))	0x77665544			
	(*((long *)(0x05)))	0x77665544			

软件访问			CAN 寄存器空间		
	(*((long *)(0x06)))	0x77665544			
	(*((long *)(0x07)))	0x77665544			
	(*((long *)(0x08)))	0xBBAA9988			
	(*((long *)(0x09)))	0xBBAA9988			
	(*((long *)(0x0A)))	0xBBAA9988			
	(*((long *)(0x0B)))	0xBBAA9988			
	(*((long *)(0x0C)))	0xFFEEDDCC			
	(*((long *)(0x0D)))	0xFFEEDDCC			
	(*((long *)(0x0E)))	0xFFEEDDCC			
	(*((long *)(0x0F)))	0xFFEEDDCC			

表格 217 Keil 等 IDE 访问 CAN 寄存器

格式	访问	显示的数据
8 位	0x00	0x00
	0x01	0x00
	0x02	0x22
	0x03	0x22
	0x04	0x44
	0x05	0x44
	0x06	0x66
	0x07	0x66
	0x08	0x88
	0x09	0x88
	0x0A	0xAA
	0x0B	0xAA
	0x0C	0xCC
	0x0D	0xCC
	0x0E	0xEE
	0x0F	0xEE
16 位	0x00	0x1100
	0x01	0x1100
	0x02	0x3322
	0x03	0x3322
	0x04	0x5544

格式	访问	显示的数据
	0x05	0x5544
	0x06	0x7766
	0x07	0x7766
	0x08	0x9988
	0x09	0x9988
	0x0A	0xBBAA
	0x0B	0xBBAA
	0x0C	0xDDCC
	0x0D	0xDDCC
	0x0E	0xFFEE
	0x0F	0xFFEE
32 位	0x00	0x11001100
	0x02	0x33223322
	0x04	0x554455544
	0x06	0x77667766
	0x08	0x99889988
	0x0A	0xBBAABBAA
	0x0C	0xDDCCDDCC
	0x0E	0xFFEEFFEE

40.5.3 工作模式

40.5.3.1 初始化模式

通过以下三种方式之一可以进入初始化模式：

- 硬件复位
- 软件复位：置位 CAN_CTRL[INIT]
- 进入 Bus-off 状态

当 CAN_CTRL[INIT]=1 时，停止 CAN 总线之间的报文传输。此外，CAN 错误计数器不会更新，CAN_TX 输出为隐性电平（即高电平）。置位 CAN_CTRL[INIT]不会影响其他配置寄存器。

初始化 CAN 时，CPU 必须配置 CAN 位时序，并且 CAN 通信中必须使用邮箱。通过清除对应的 MAILEN 位，以停用 CAN 通信中未使用的邮箱。

当置位 CAN_CTRL[INIT]和 CAN_CTRL[CFGWACCEN]时，可以访问位时序寄存器以配置时序。

通过清除 CAN_CTRL[INIT]位完成软件初始化。此后，为了与 CAN 总线上的数据传输进行同步，BSP 需要等待 11 个连续的高电平（即总线空闲状态）的出现

后，参与总线活动并开始报文传输，有关位时序的信息，详情请参见位时序章节。

邮箱的初始化不受 CAN_CTRL[INIT]的影响。然而，所有的邮箱在开始报文传输前都必须配置为特定的标识符，或设置为“无效”状态。

在 CPU 正常运行期间，可以改变邮箱的配置。在配置并随后将邮箱从接口寄存器传输到报文 RAM 后，如果修改的邮箱编号与先前找到的邮箱相同或更小时，将对此邮箱进行接收过滤。这样即使在更改邮箱时，例如存在待处理的 CAN 帧接收时，也可以确保数据的一致性。

40.5.3.2 正常模式

CAN 初始化并将 CAN_CTRL[INIT]复位后，CAN 内核将与 CAN 总线同步，并准备进行报文传输。

当接收到的报文通过了接收过滤，则整个报文都将被存储到对应的邮箱中，包括 MSGID，DLC 和最多 8 个数据字节。因此，举例来说，假设使用了标识符掩码，允许一些位在改变，而不会这些变动影响到报文的接收。

由于报文处理器可以在并发访问的情况下确保数据一致性，因此 CPU 可以在任意时间通过接口寄存器读取或写入每条报文。

CPU 可以更新待发送的报文。当报文具有永久邮箱（即在配置时设置了 MSGID 和控制位，并设置为用于多个 CAN 传输）时，则可以仅更新数据字节。当需要将多个待发送的报文分配给同一个邮箱时，必须先配置整个邮箱，再发出发送报文请求。多个邮箱的发送请求可以同时被响应。然后，根据其内部优先级，逐个发送邮箱。即使发送请求被挂起，报文也可以随时进行更新或设置为“无效”。然而，如果在挂起的发送请求开始传输前更新了报文，则将丢弃报文的数据字节。根据邮箱的配置，当接收到与其标识符匹配的远程帧时，可以自动请求传输。

40.5.4 总线自动开启功能

当总线进入 Bus-off 状态后，CPU 通过复位 CAN_CTRL[INIT]以启动 Bus-off 恢复序列，否则，CAN 模块将一直处于 Bus-off 状态。通过置位 CAN_CTRL[ABOEN]，使能自动总线启动功能，此时，CAN 自动启动 Bus-off 恢复序列。用户可以定义 Bus-off 恢复序列启动的延迟时钟周期数，以控制序列的启动时间。

注意：当 CAN 模块由于多次总线错误而进入 Bus-Off 状态，停止所有总线活动的同时自动置位 CAN_CTRL[INIT]。在应用程序清除 CAN_CTRL[INIT]，或使能自动总线启动功能后，在等待 129 个总线空闲位（即 129 * 11 个连续的隐性位）后，设备才会恢复正常操作。设置或清除 CAN_CTRL[INIT]无法缩短 Bus-off 恢复序列的时间。此外，错误计数器在 Bus-off 恢复序列的末尾复位。

为了使 CPU 能够检测 CAN 总线是否持续受到干扰，或停滞在显性位，并监控 Bus-off 恢复序列的进展，在 CAN_CTRL[INIT] 复位后，每当检测到 11 个连续的隐性位（即逻辑“1”）时，会向错误标志寄存器（CAN_EFLG）写入 Bit 0 错误代码。

40.5.5 禁用自动重传

自动重传功能在默认情况下为使能状态，通过设置 CAN_CTRL[DISARETX]以禁用自动重传。

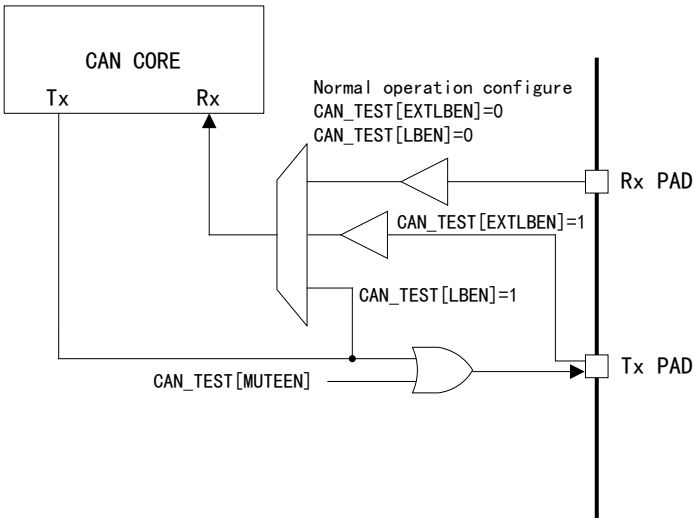
基于 CAN 规范，自动重传机制用于重新传输仲裁失败或受到干扰错误的帧。直到成功完成传输前，CAN 不会向用户提供帧传输是否成功的确认信息。更多此模式的信息，详情请参见事件驱动报文发送章节。

40.5.6 测试模式

测试模式主要用于 CAN 模块的自检，对于所有的测试模式，需要先置位 CAN_CTRL[TESTEN]，以使能对测试寄存器（CAN_TEST）的写操作。

各测试模式的操作如下图所示，需要注意的是，模块的门级精确功能不在下图中体现，此图仅展示模块的操作，不包括 GPIO 复用或 I/O 缓冲区模块。

图 18 CAN_MUX



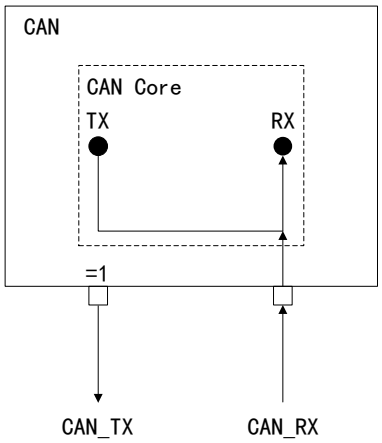
40.5.6.1 静默模式

置位 CAN_TEST[SLTEN]，使能静默模式。

此模式下，只能向 CAN 总线发送隐性位（逻辑“1”），不能发送显性位（逻辑“0”），可以从总线接收有效的数据帧和远程帧，并将接收到的帧路由到 CAN 内核处理。由于不能通过发送显性位（例如确认位，过载标志，主动错误标志）来影响 CAN 总线，因此在静默模式下可以在不影响 CAN 总线的情况下分析 CAN 总线上的信息流

静默模式下 CAN_TX 和 CAN_RX 与 CAN 内核的连接情况如下图所示：

图 245 CAN 工作在静默模式



在 ISO 11898-1 中，静默模式被称为总线监视模式。

40.5.6.2 环回模式

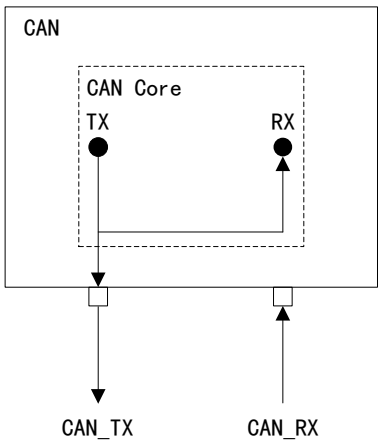
置位 CAN_TEST[LBEN]，使能环回模式。

此模式下，在 CAN 内部，CAN 内核将 Tx 输出连接到 Rx 输入，发送的报文直接传到输入端接收，被当做接收到的报文。当报文通过接收过滤时，可以存储到邮箱中。CAN 内核忽略外部 CAN_RX 输入的实际值，在 CAN_TX 引脚上可以监控到发送的报文

环回模式主要用于硬件自检功能。在环回模式下，应答错误（即在数据帧/远程帧的应答时隙中采样到的隐性位）会被 CAN 内核忽略。

环回模式下 CAN_TX 和 CAN_RX 与 CAN 内核的连接情况如下图所示：

图 246 CAN 工作在环回模式



注意：在环回模式下，会忽略 CAN 内核与 Tx 引脚之间的信号路径，以及 Tx 引脚自身。当需要测试以上内容时，详情请参见环回静默模式章节。

40.5.6.3 环回静默模式

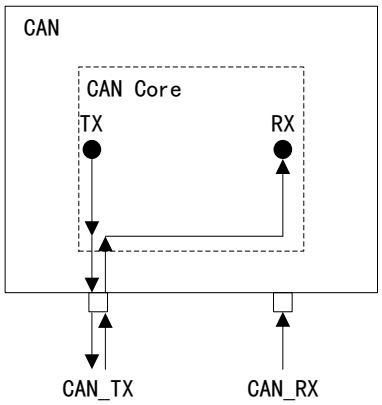
同时置位 CAN_TEST[LBEN]和 CAN_TEST[SLTEN]，使能环回静默模式。

此模式下，发送的数据直接传到输入端接收，CAN_RX 引脚与 CAN 内核未连

接，不能从 CAN 总线上接收数据。CAN_TX 引脚只能向总线发送隐性位（逻辑“1”），不能发送显性位（逻辑“0”）。环回静默模式可以在不干扰 CAN 网络正常运行的情况下测试 CAN 硬件。

环回静默模式下 CAN_TX 和 CAN_RX 与 CAN 内核的连接情况如下图所示：

图 247 CAN 工作在环回静默模式



40.5.6.4 外部环回模式

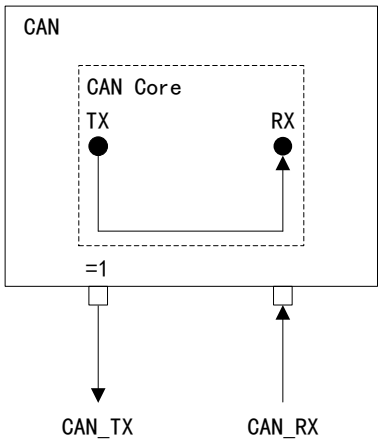
置位 CAN_TEST[EXTLBEN]，使能外部环回模式。在环回模式激活时（即置位 CAN_TEST[LBEN] 时），CAN_TEST[EXTLBEN]的置位会被系统忽略。

此模式下 CAN 内核与 CAN_TX 引脚的输入缓冲区连接，包含从数据从 CAN 内核发出到 CAN_TX 引脚，再由 TX 引脚返回到 CAN 内核的完整信号路径。

外部环回模式与环回模式类似，用于测试 Tx 引脚的 IO 电路是否正常工作。

外部环回模式下 CAN_TX 和 CAN_RX 与 CAN 内核的连接情况如下图所示：

图 248 CAN 工作在外部环回模式



40.5.7 时钟源

通常情况下，系统时钟（SYSCLK）提供 CAN 位时序时钟。此外，如有需求，外部时钟源（XTAL）也可以提供 CAN 位时钟。有关配置 CAN 模块时钟源的信息，详情请参见系统控制中断章节和数据手册。

配置 CAN 内核时，必须确保每个位时间至少为 8 个时钟周期。为了达到 1Mbps

的传输速率，至少需要使用 8MHz 或更高频率的振荡器。

40.5.8 中断

40.5.8.1 中断线

CAN 模块有两个中断线：CANINT0 和 CANINT1。

置位 CAN_CTRL[IEN0]，使能 CANINT0 中断线；置位 CAN_CTRL[IEN1]，使能 CANINT1 中断线。

CAN 模块配备了模块级的中断使能和确认功能。置位 CAN_GLBINTEN [GLBINT0EN]，使能 CANINT0 全局中断；置位 CAN_GLBINTEN [GLBINT1EN]，使能 CANINT1 全局中断。处理某个中断时，在使用 CAN_GLBINTCLR 和 NVIC_ACKn 确认中断前，优先清除单个报文或状态改变标志。

40.5.8.2 中断源类型

CAN 可分为以下三类中断源：

- 错误中断
- 邮箱中断
- 状态改变中断

通过中断寄存器（CAN_INT）中的 INT0 和 INT1 以识别中断源。INT0 和 INT1 置 0 时，表示当前没有挂起的中断。当触发中断线 CANINT0 或 CANINT1 时，中断线将保持激活状态，直到满足以下任一条件：

（1） 中断寄存器（CAN_INT）中的对应位 INT0 或 INT1 重新置 0（表示已处理触发中断的事件）；

（2） 对应位 CAN_CTRL[IEN0]或 CAN_CTRL[IEN1]复位。

当 CAN_INT[INT0]=0x8000 时，表示由于 CAN 内核更新了错误标志寄存器，产生了一个待处理的中断，且此中断的优先级最高。注意，此更新操作并不一定表示错误标志寄存器中的值改变。

通过读取错误标志寄存器，CPU 可以更新（或复位）RXCFLG，TXCFLG 和 LEC 标志位。当 CPU 对错误标志寄存器进行写入操作时，不会产生或复位中断。当 CAN_INT[INT0]或 CAN_INT[INT1]的值在 0x0001-0x0020 时，此值与邮箱编号对应，表示对应编号的邮箱产生了中断。CAN_INT[INT0]和 CAN_INT[INT1]的值指向挂起的优先级最高的中断。邮箱的编号越小，其优先级越高；反之，编号越大，优先级越低。

中断服务程序在读取中断源消息时，还能够读取报文并通过清除 IFx 命令寄存器（CAN_IFxCOM）（x=1/2）复位邮箱的 IPEND 位。当清除 IPEND 位时，中断寄存器将指向下一个中断挂起的邮箱。

错误中断

通过置位 CAN_CTRL[ERRIEN]，使能错误中断组。错误中断仅路由到 CANINT0

中断线中，置位 CAN_CTRL[IEN0]，使能 CANINT0 中断线。

错误中断包含以下三种事件：

- 错误警告（EWFLG）
- Bus-off（BUSOFLG）
- 奇偶校验错误（PEFLG）

邮箱中断

邮箱中断可以路由到 CANINT0 和 CANINT1 中断线。

置位 CAN_IFxMCTRL[IPEND]可以强制产生中断。

邮箱中断由邮箱事件产生，由邮箱结构中的 IPEND，TXIEN 和 RXIEN 标志控制。

状态改变中断

通过置位 CAN_CTRL[SCHIEN]，使能状态改变中断组。置位后，每个 CAN 帧都会产生一个状态改变中断，而与有效的 CAN 通信，总线错误和报文 RAM 的配置都无关。

状态改变中断仅路由到 CANINT0 中断线中，置位 CAN_CTRL[IEN0]，使能 CANINT0 中断线。

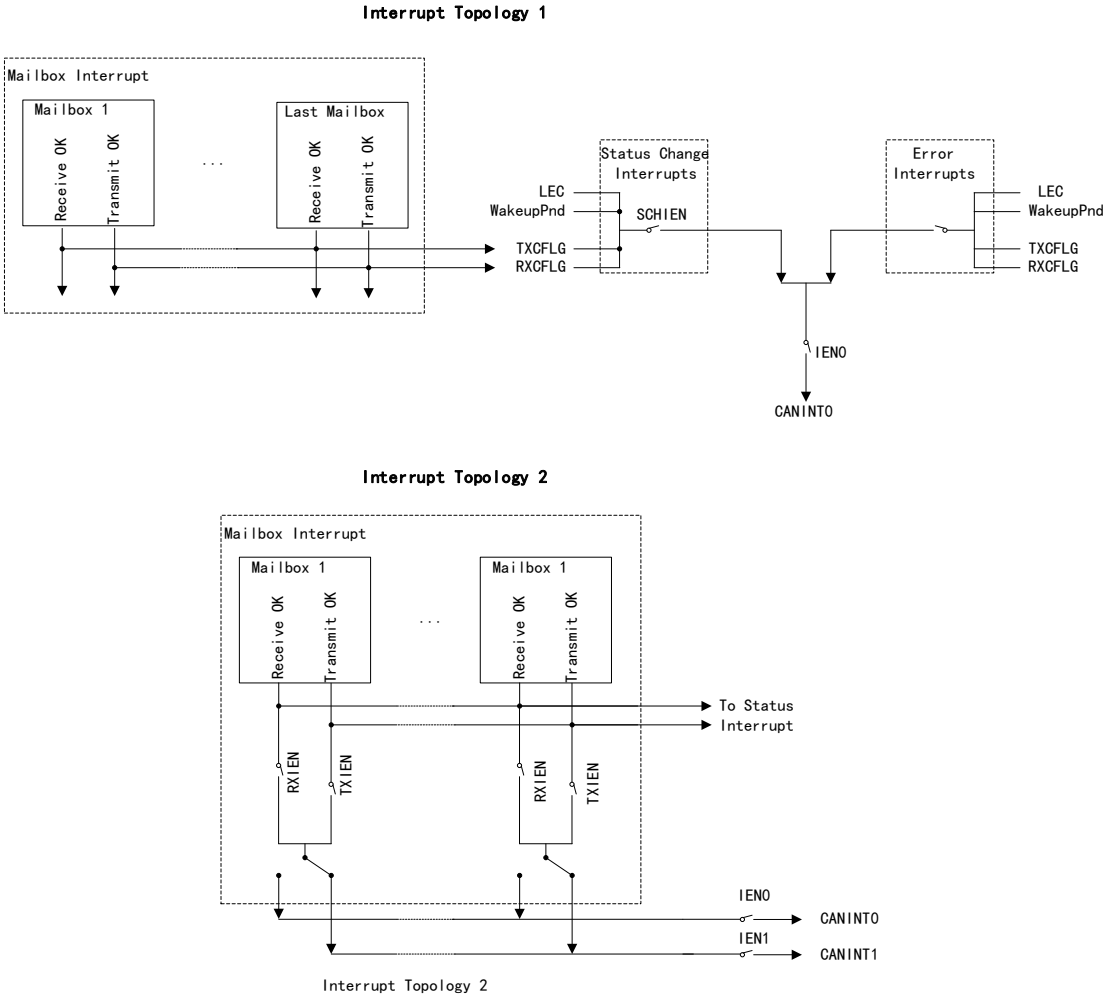
状态改变中断包含以下三种事件：

- 最后一个错误类型（LEC）
- 发送完成（TXCFLG）
- 接收完成（RXCFLG）

40.5.8.3 中断拓扑结构

CAN 中断拓扑结构如下图所示：

图 249 中断拓扑结构



如上图所示，错误中断和状态改变中断只能路由到 CANINT0 中断线，但用于发送和接收操作的邮箱中断可以路由到 CANINT0 和 CANINT1 中断线。

40.5.8.4 NVIC 中断命名

在 CANA 和 CANB 中，CAN 模块中断线的 NVIC 命名如下表所示：

表格 218 中断的 NVIC 命名

中断	CANINT0	CANINT1
CANA	CANA_0	CANA_1
CANB	CANB_0	CANB_1

40.5.9 DMA 功能

CAN 模块的 DMA 功能通过接口寄存器 IFx 实现。IF1/2/3 三个接口寄存器分别对应 CAN 模块的 3 个 DMA 触发输出，通过置位 CAN_CTRL[IFDENx](x=1..3)，使能接口寄存器对应的 DMA 请求。

完成 CAN 内部 IFx 更新后，会触发并保持一个 DMA 请求，当对 IFx 寄存器组的首次读写操作会清除 DMA 请求。

对于 IF1 和 IF2 寄存器，通过对 IF1 和 IF2 命令寄存器进行写操作以启动更新 IF1 和 IF2 寄存器。启动后，将置位 IF1/2 命令寄存器的 REQDMAIFx，当对应接口变为可用时，触发一次 DMA 请求。对于 IF3 寄存器，当某个邮箱被配置为自动更新 IF3，并且接收到 CAN 报文时，IF3 寄存器的内容会自动更新。若 IF3 的 DMA 请求被使能，则每次 IF3 更新都将触发一个 DMA 请求。

40.5.10 奇偶校验机制

为了确保报文 RAM 数据的完整性，CAN 提供了奇偶校验机制。每 32 位数据（1 个字）会计算出一个对应的奇偶校验位。

配置 CAN_CTRL[PEN]=1，使能奇偶校验机制。此时，数据写入报文 RAM 中时，奇偶校验位也一同写入，在读取数据时，将对比存储在报文 RAM 中的奇偶校验位以检查数据的完整性。当使能奇偶校验机制时，CAN 模块将自动生成并检查奇偶校验位。奇偶校验位的生成遵循模 2 求和的规则，即如果数据位的模 2 求和为 1，则设置奇偶校验位。换句话说，当设置奇偶校验位时，那么在数据中 1 的数量是奇数个。

配置 CAN_CTRL[PEN]=0，禁用奇偶校验机制。此时，对报文 RAM 写入数据时，报文 RAM 中的奇偶校验位保持不变，读取数据时，不会对比奇偶校验位。

奇偶校验错误的处理机制

对报文 RAM 的每次读操作都会对邮箱进行奇偶校验，例如，在 CAN 帧传输开始时进行奇偶校验。当检测到奇偶校验错误时，置位 CAN_EFLG[PEFLG]，以指示发生了奇偶校验错误。此时将清除邮箱的 MVAL 位，以避免在 CAN 总线上传输无效的数据。此外，若使能了错误中断，在检测到奇偶校验错误的同时还会产生一个中断。

由于即使存在奇偶校验错误，CPU 仍然可以读取邮箱中的数据，因此，应用程序必须确保 CPU 读取的是有效数据。例如，可以通过在奇偶校验错误中断发生时立即检查奇偶校验错误寄存器（CAN_PE）

40.5.11 调试模式

当外部调试器连接到 CAN 模块且暂停 CAN 内核时，将自动进入调试模式。CAN 模块的调试模式提供了暂停 CAN 活动，使调试器可以访问报文 RAM 内容等功能，用于支持外部调试单元的使用。

系统会等待当前进行中的发送操作启动或接收操作完成，或检测到总线空闲状态后，再进入调试模式。当置位 CAN_CTRL[IDBGSEN]时，调试器可以立即中断当前正在进行的发送或接收。进入调试模式后，将置位 CAN_CTRL[DBGFLG]以指示当前 CAN 模块处于调试模式。

此模式下，可以访问所有 CAN 的寄存器，但是对保留位进行操作是无效的。外部调试单元可以读取内存映射的信息 RAM，相关内容请参见“调试模式下报文 RAM 表示”章节。在调试模式中，将禁用错误标志寄存器和 IF1/1F2 命令寄存器的自动清除功能。

注意：

（1） 在调试模式下，对控制寄存器的写入可能会影响 CAN 状态机，并且进一步影响报文的处理。

（2） 由于在调试模式下，报文 RAM 直接被内存映射，从而允许调试器直接访问，此时不能通过接口寄存器组来访问消息 RAM。

40.5.12 CAN 初始化

硬件复位后 CAN 的初始状态

在硬件复位之后，将禁用所有的 CAN 协议功能，并置位 CAN_CTRL[INIT]。必须先配置位时序和邮箱，再使能 CAN 协议功能。

配置 CAN 位时序

配置位时序时，需要同时置位 CAN_CTRL[INIT]和 CAN_CTRL[CFGWACCEN]。

当清除 CAN_CTRL[INIT]位时，CAN 内核的协议控制器状态机和消息处理器状态机开始控制 CAN 的内部数据流，表示 CAN 模块由初始化模式过渡到正常运行模式。待发送的报文将被存储在 CAN 内核的移位寄存器中，并按照 CAN 协议，通过 CAN 总线发送。通过接收过滤的报文将被存储在报文 RAM 中。

有关位时序配置的信息，详情请参见位时序配置章节。

配置 CAN 邮箱

配置邮箱时，不需要额外置位 CAN_CTRL[CFGWACCEN]。

硬件复位将邮箱的 MVAL，NDATA，IPEND 和 TXREQ 都复位为 0。通过设置 IF1/IF2 接口寄存器组的掩码寄存器，仲裁寄存器，报文控制寄存器和数据寄存器，以实现对邮箱的配置。将邮箱编号写入到对应 IF1/IF2 命令寄存器的 VALMNUM 位，即可将 IF1/IF2 寄存器中的设定值加载到报文 RAM 相应的邮箱中。

有关邮箱配置的信息，详情请参见邮箱配置章节。

中断使能

CPU 清除 CAN_CTRL[INIT]和 CAN_CTRL[CFGWACCEN]后，可以同时使能 CANINT0 和 CANINT1 中断线，即同时置位 CAN_CTRL[IEN0]和 CAN_CTRL[IEN1]。此外，还可以同时使能错误中断和错误中断，即同时置位 CAN_CTRL[ERRIEN]和 CAN_CTRL[SCHIEN]。

中断驱动/轮询模式初始化

CAN 通信支持中断驱动和轮询两种模式进行数据传输控制。

（1） 在中断驱动模式下，当邮箱的状态发生变化时（如接收到新报文或报文发送完成），会设置相应邮箱的中断挂起（IPEND）位为 1。此时，中断寄

存器会指向那些 IPEND 位为 1 的邮箱。即使禁用中断线 CANINT0 和 CANINT1（即 CAN_CTRL[IEN0]和 CAN_CTRL[IEN1]=0），中断寄存器仍然会更新。

（2） 在轮询模式下，CPU 轮询所有邮箱的新数据寄存器（CAN_NDATA）的 NDATA 位和发送请求寄存器（CAN_TXREQ）的 TXREQ 位。为了简化轮询操作，建议将所有的发送邮箱分配给较低编号的组中，并将所有的接收邮箱分配到较高编号的组中。

40.5.13 报文 RAM

报文 RAM 用于存储邮箱及邮箱的奇偶校验位，特点如下：

（1） 邮箱数量：一共有 32 个邮箱存储在报文 RAM 中；

（2） 间接访问：在正常操作中，CPU 不能直接访问报文 RAM，只能通过接口寄存器组对报文 RAM 进行间接访问；

（3） 调试模式访问：报文 RAM 只能在调试模式下被访问。

（4） 报文 RAM 的基地址=CAN 外设基地址+0x400

（5） 接口寄存器组：

- IF1 和 IF2：这两组寄存器支持 CPU 对报文 RAM 进行间接读/写操作，并缓冲将要传输到邮箱或从邮箱中发送的控制信息和用户数据；
- IF3：IF3 寄存器组在接收到 CAN 报文并且邮箱更新后，自动从报文 RAM 接收控制信息和用户数据。CPU 无需启动从报文 RAM 到 IF3 寄存器组的传输。

（6） 避免冲突：报文处理程序解决了并发访问报文 RAM 和 CAN 帧接收/发送之间潜在的冲突；

40.5.13.1 邮箱寻址

报文 RAM 中，某个邮箱的起始地址=邮箱基地址+邮箱编号*0x20。

例如：

邮箱 1 起始地址=邮箱基地址+1*0x20，即邮箱 1 的偏移地址为 0x20

邮箱 2 起始地址=邮箱基地址+2*0x20，即邮箱 2 的偏移地址为 0x40

注意：

（1） 邮箱编号为 1-32， 0 是无效的邮箱编号

（2） 邮箱的编号越低，优先级越高；反之，邮箱的编号越高，优先级越低。因此，在地址 0x00 处的邮箱 32（编号最大的邮箱）的优先级最低，邮箱 1 的优先级最高。

（3） 如果向尚未分配或者未被系统定义为有效邮箱的报文 RAM 地址中写入数据，可能会覆盖已实现的邮箱的数据。

表格 219 调试模式下报文 RAM 寻址

调试模式	邮箱编号	偏移地址	字编号
Parity	1	0x20	1
MXtd,MDir,Mask		0x24	2
Xtd,Dir,ID		0x28	3
Ctrl		0x2C	4
Data Bytes 3-0		0x30	5
Data Bytes 7-4		0x34	6
Parity	2	0x40	1
MXtd,MDir,Mask		0x44	2
Xtd,Dir,ID		0x48	3
Ctrl		0x4C	4
Data Bytes 3-0		0x50	5
Data Bytes 7-4		0x54	6
…	…	…	…
Parity	31	0x3E0	1
MXtd,MDir,Mask		0x3E4	2
Xtd,Dir,ID		0x3E8	3
Ctrl		0x3EC	4
Data Bytes 3-0		0x3F0	5
Data Bytes 7-4		0x3F4	6
Parity	32	0x00	1
MXtd,MDir,Mask		0x04	2
Xtd,Dir,ID		0x08	3
Ctrl		0x0C	4
Data Bytes 3-0		0x10	5
Data Bytes 7-4		0x014	6

注意：邮箱 32 为实现的最大邮箱编号。

40.5.13.2 调试模式下报文 RAM 表示

在调试模式中，由于报文 RAM 是内存映射的，因此外部调试单元可以直接访问报文 RAM。此时，不能通过 IFx 寄存器组访问报文 RAM。

图 250 调试模式下报文 RAM 结构

Message RAM

31	30	...	18	17	16
15	14	...	2	1	0

Message Address + 0x00

Reserved	
Reserved	Parity[4:0]

Message Address + 0x04

EXTMASK	DIRMASK	Reserved	IDMASK[28:16]
IDMASK[15:0]			

Message Address + 0x08

Reserved	EXTIDCFG	MDIRCFG	MIDCFG[28:16]
MIDCFG[15:0]			

Message Address + 0x0C

Reserved										
Reserved	MLOST	Reserved	MASKEN	TXIEN	RXIEN	REMEN	Reserved	EOBEN	Reserved	DLCCFG[3:0]

Message Address + 0x10

DATD3	DATD2
DATD1	DATD0

Message Address + 0x14

DATD7	DATD6
DATD5	DATD4

40.5.14 邮箱结构

邮箱结构如下图所示，其中 NDATA，IPEND，TXREQEN，MAILEN 由专用寄存器表示。

图 251 邮箱结构

Mailbox												
MASKEN	IDMASK	EXTMASK	DIRMASK	EOBEN		NDATA	MLOST	RXIEN	TXIEN	IPEND	REMEN	TXREQEN
MAILEN	MIDCFG	EXTIDCFG	MDIRCFG	DLCCFG	DATD0	DATD1	DATD2	DATD3	DATD4	DATD5	DATD6	DATD7

Unused

MASKEN –使能掩码

该位指示 Mask 位段（IDMASK，EXTMASK，DIRMASK）是否用于接收过滤。当此位置 1 时，必须在邮箱初始化期间配置邮箱的 MASK 位段，再置位 CAN_MVAL[MVAL]。

0：被忽略，并且不用于接收过滤。

1：用于接收过滤。

IDMASK –标识符掩码

该位指示邮箱标识符中的对应位是否用于接收过滤。

0：不用于接收过滤。

1：用于接收过滤。

EXTMASK –扩展标识符掩码

该位指示扩展标识符（CAN_IFxARB[EXTIDCFG]）是否用于接收过滤。

当邮箱使用 11 位的标准格式的标识符时，接收到的数据帧的 11 位标识符被放置在 MIDCFG[28:18]。在接收过滤的过程中，仅考虑 MIDCFG[28:18]和对应的 IDMASK[28:18]位。

0：不用于接受过滤。

1：用于接受过滤。

DIRMASK–报文方向掩码

该位指示报文方向位（CAN_IFxARB[MDIRCFG]）是否用于接收过滤。

0：不用于接收过滤。

1：用于接收过滤。

EOBEN –使能块结束

该位用于将多个邮箱级联起来构建一个 FIFO 缓冲区。当作为独立邮箱时，此位必须始终置 1。

0：当前的邮箱属于 FIFO 缓冲区块的一部分，并且不是这个 FIFO 缓冲区块的最后一个邮箱

1：当前的邮箱是一个独立的邮箱，不属于任何 FIFO 缓冲区块或当前的邮箱是 FIFO 缓冲区块中的最后一个邮箱

NDATA –新数据

该位用于指示邮箱的数据字节是否更新。

0：数据未更新（上次清除此标志位后，没有新数据被报文处理器写入到此邮箱的数据部分）

1：此邮箱的数据部分有新数据（由 CPU 或报文处理器）写入

MLOST –报文丢失

该位仅适用于传输方向设置为接收的邮箱。

0：CPU 上次复位此位后，没有报文丢失

1：发生了报文丢失，即报文处理器在 CAN_NDATA[NDATA]置位时接收到了新报文，并且将其存储在了此邮箱中，导致之前的报文被新报文覆盖。

RXIEN –使能接收中断

该位指示成功接收到帧数据后，是否触发 IPEND。

0：未触发 IPEND。

1：触发 IPEND。

TXIEN –使能发送中断

该位指示成功发送帧数据后，是否触发 IPEND。

0：未触发 IPEND。

1：触发 IPEND。

IPEND –中断挂起

该位指示邮箱中是否有挂起的中断。

0：此邮箱中没有挂起的中断。

1：邮箱中有挂起的中断。当没有其他优先级更高的中断挂起时，中断寄存器中的中断标识符将指向此邮箱

REMEN –使能远程帧

该位指示接收到远程帧后，是否置位 TXREQEN。

0：接收到远程帧后，不更改 TXREQEN

1：接收到远程帧后，置位 TXREQEN

TXREQEN –使能发送请求

该位指示邮箱中是否有待处理的发送请求。

0：此邮箱没有待处理的发送请求

1：此邮箱有待处理的发送请求并且未完成发送

MAILEN –使能邮箱

此位指示报文处理器是否使用此邮箱。禁用邮箱时，报文处理器忽略此邮箱，它不会被报文处理器考虑用于接收或发送 CAN 报文；使能邮箱时，它会被报文处理器用于接收或发送操作。

注意：即使标识符位段的 MIDCFG，EXTIDCFG，MDIRCFG，或 DLCCFG 改变，此位也可能保持高电平。

0：禁用，报文处理器忽略此邮箱

1：使能，报文处理器使用此邮箱

MIDCFG–邮箱标识符

该位指示邮箱的标识符。

11 位标识符-标准帧：MIDCFG[28:18]

29 位标识符-扩展帧：MIDCFG[28:0]

EXTIDCFG–扩展标识符

该位指示邮箱使用的标识符是标准格式还是扩展格式。

0：邮箱使用 11 位标准标识符

1：邮箱使用 29 位扩展标识符

MDIRCFG –报文有效

该位指示报文的传输方向。

0：接收，当 TXREQEN=1 时，发送远程帧，远程帧携带具有与此邮箱相匹配的标识符；当接收到数据帧，且此数据帧的标识符与邮箱的标识符匹配时，该数据帧将被存储到邮箱中

1：发送，当 TXREQEN=1 时，发送数据帧；当接收到远程帧，且此远程帧的标识符与邮箱的标识符匹配时，如果 REMEN=1，设置此邮箱的 TXREQEN=1

DLCCFG –数据长度码

该位指示数据帧的数据长度。

注意：具有相同标识符的不同节点的邮箱需要设置相同的数据长度码。当报文处理器接收到一个数据帧时，它将根据接收到的报文设置相应邮箱中的 DLC。

0000-1000：数据帧具有 0-8 个数据字节

1001-1111：数据帧具有 8 个数据字节

DATDx –数据帧的第 x 个数据字节

该位表示数据帧的数据部分。

数据字节按照接收期间移位到 CAN 内核的移位寄存器的顺序排列，即 DATD0 是第一个字节，DATD7 是最后一个字节。报文处理器存储数据帧时，将 8 个数据字节全部写入邮箱中，若不足 8 个数据字节，邮箱中剩余的字节部分可能写入未定义的值。

40.5.15 邮箱配置

在初始化结束前应完成对整个报文 RAM 的配置，但在 CAN 通信期间可以改变邮

箱的配置。

40.5.15.1 发送邮箱配置

发送邮箱的各位段初始化配置如下表所示：

表格 220 发送邮箱初始化配置

位段	值
MAILEN	1
Aarbitration	由用户程序指定
Data	由用户程序指定
Mask	由用户程序指定
EOBEN	1
MDIRCFG	1
NDATA	0
MLOST	0
RXIEN	0
TXIEN	由用户程序指定
IPEND	0
REMEN	由用户程序指定
TXREQEN	0

（1） Aarbitration 位段：此位段定义了发出报文的标识符和类型，包含 MIDCFG 和 EXTIDCFG，由用户程序指定此位段的值。当使用标准帧，即 11 位的标识符时，即 EXTIDCFG=0，忽略 MIDCFG[17:0]，11 位标识符被写入 MIDCFG[28:18]中。

（2） Data 位段：此位段定义数据寄存器信息，包含 DLCCFG 和 DATD0-7，由用户程序指定此位段的值。应先设置数据有效，再置位 REMEN 和 TXREQEN

（3） Mask 位段：此位段用于接收具有相同标识符的数据帧组以置位 TXREQEN，包含 IDMASK，MASKEN，EXTMASK 和 DIRMASK。使能此位段后（即 MASKEN=1），可以接收具有相同标识符的数据帧组。不可以屏蔽 MDIRCFG 位。当禁止使用远程帧置位 TXREQEN 时，此位段也被禁用，即当 REMEN=0 时，MASKEN=0。更多信息，详情请参见接收数据帧/远程帧章节。

（4） TXIEN 位段：置位后，在邮箱成功发送数据后，将置位 IPEND。

（5） REMEN 位段：置位后，当 CAN 接收到一个匹配远程帧后，将置位 TXREQEN。CAN 将自动生成一个数据帧，并发送此数据帧作为对远程帧的响应。

发送邮箱远程帧配置

当发送远程帧时，不需要配置发送邮箱。为接收邮箱置位 TXREQEN 时，发送邮箱会自动发送一个与此接收邮箱的数据帧具有相同标识符的远程帧。

40.5.15.2 单个接收邮箱配置-接收数据帧

当报文处理器存储邮箱中的数据帧时，会存储接收到的数据长度码和与数据长度码对应的字节数。若数据长度码小于 8，邮箱中的其余字节会被未定义值填充。当配置接收邮箱用于接收数据帧时，各位段初始化配置如下表所示：

表格 221 接收邮箱-接收数据帧初始化配置

位段	值
MAILEN	1
Aarbitration	由用户程序指定
Data	由用户程序指定
Mask	由用户程序指定
EOBEN	1
MDIRCFG	0
NDATA	0
MLOST	0
RXIEN	由用户程序指定
TXIEN	0
IPEND	0
REMEN	0
TXREQEN	0

（1） Aarbitration 位段：此位段定义了接收数据帧的标识符和类型，包含 MIDCFG 和 EXTIDCFG，由用户程序指定此位段的值。当使用标准帧，即 11 位的标识符时，即 EXTIDCFG=0，忽略 MIDCFG[17:0]，11 位标识符被写入 MIDCFG[28:18]中。当接收到带有 11 位标识符的数据帧时，MIDCFG[17:0]将被设置为 0。

（2） Mask 位段：此位段用于接收具有相同标识符的数据帧组，包含 IDMASK，MASKEN，EXTMASK 和 DIRMASK。使能此位段后（即 MASKEN=1），可以接收具有相同标识符的数据帧组。在典型的应用中，不可以屏蔽 MDIRCFG 位。当 Mask 位段的某些位被设置为“无效”时，仲裁寄存器中的相应位将被存储的数据帧的位所覆盖

（3） RXIEN 位段：置位后，在邮箱接收到数据帧并存储后，将置位 IPEND。

（4） TXREQEN 位段：置位后，将发送一个远程帧，此远程帧的标识符与仲裁位段中实际存储的标识符匹配。当使能掩码时用于接收过滤时，即 MASKEN=1，可能会改变 Aarbitration 位段中的值。

40.5.15.3 单个接收邮箱配置-接收远程帧

接收邮箱用于接收远程帧时，可监控 CAN 总线上的远程帧，能够扩展成一个 FIFO 缓冲区。有关缓冲区的信息，详情请参见 FIFO 缓冲区配置章节。接收邮箱中存储的远程帧不会触发数据帧的发送。当配置接收邮箱用于接收远程帧时，各位段初始化配置如下表所示：

表格 222 接收邮箱-接收远程帧初始化配置

位段	值
MAILEN	1
Aarbitration	由用户程序指定
Data	由用户程序指定
Mask	由用户程序指定
EOBEN	1
MDIRCFG	1
NDATA	0
MLOST	0
RXIEN	由用户程序指定
TXIEN	0
IPEND	0
REMEN	0
TXREQEN	0

（1） Aarbitration 位段：此位段定义了接收远程帧的标识符和类型，包含 MIDCFG 和 EXTIDCFG，由用户程序指定此位段的值。当使用标准帧，即 11 位的标识符时，即 EXTIDCFG=0，忽略 MIDCFG[17:0]，11 位标识符被写入 MIDCFG[28:18]中。当接收到带有 11 位标识符的数据帧时，MIDCFG[17:0]将被设置为 0。

（2） Mask 位段：此位段用于接收具有相同标识符的远程帧组，包含 IDMASK，MASKEN，EXTMASK 和 DIRMASK。在这些位中，必须置位 MASKEN。在典型的应用中，不可以屏蔽 MDIRCFG 位。可以将 Mask 位段设置为“必须匹配”或“无效”，以允许对具有相同标识符的远程帧组进行接收过滤。更多信息，详情请参见接收数据帧/远程帧章节。

（3） RXIEN 位段：置位后，在邮箱接收到远程帧并存储后，将置位 IPEND。

（4） 数据长度码（DLCCFG）：当报文处理器存储邮箱中接收到的远程帧时，数据长度码也一同被保存，邮箱中的数据字节将保持不变。数据长度码由用户程序指定。

40.5.15.4 FIFO 缓冲区配置

连接多个邮箱到同一个 FIFO 缓冲区时，需要确保这些邮箱的标识符和掩码（如

果使用）被编程为匹配的值。由于邮箱的编号和优先级的隐藏关系，编号最小的邮箱将作为 FIFO 缓冲区的第一个邮箱。

对于 FIFO 缓冲区的接收邮箱配置，除 EOBEN 位外，其配置与单个接收邮箱相同。FIFO 缓冲区中的最后一个邮箱必须配置为 1，以表示块结束，而除最后一个邮箱外的其他所有邮箱的 EOBEN 位都必须配置为 0。

40.5.16 报文处理器

报文处理

CAN 模块初始化完成后，能够立即与 CAN 总线上的其他设备同步通信。接收到的报文在 CAN 中进行接收过滤，当报文通过接收过滤后，被存储在指定的邮箱中，用户可以读取这些报文。

用户必须更新待发送的报文数据，使能发送并触发发送请求。当接收到与标识符匹配的远程帧时，CAN 将自动请求发送报文。

如果在用户读取邮箱当前的报文前，此邮箱又接收到了新的报文，那么新报文覆盖未读取的报文。

可以通过以下两种方式读取报文：

- 通过中断驱动读取
- 轮询 NDATA 后读取

报文处理器

报文处理器状态机（以下简称报文处理器），用于控制 CAN 内核的移位寄存器与报文 RAM 之间的数据传输。

报文处理器负责执行以下任务：

（1） 数据传输
- 接收报文，从 CAN 内核到报文 RAM
- 待发送报文，从报文 RAM 到 CAN 内核
- 从 CAN 内核到接收过滤单元

（2） 接收过滤，通过扫描报文 RAM，寻找匹配的邮箱

（3） 当接口寄存器（IF1 和 IF2）修改了邮箱，那么该邮箱只有在其优先级未降低（即保持不变或提高）时，才会在再次扫描时被报文处理器识别和处理。

（4） 处理中断标志和发送请求标志

在报文处理器的寄存器中，包含所有邮箱的传输请求标志，新数据标志，中断挂起标志和报文有效标志四类状态标志，只能对这些寄存器进行读访问。

报文处理寄存器使用户能够高效地查看所有邮箱的状态信息，而不必使用 IFx 寄存器去逐个收集每一个邮箱的各项状态信息。比如，通过报文处理寄存器可以一次性查看所有待发送请求。

40.5.17 事件驱动报文发送

当满足以下两个条件时，CAN 模块会评估邮箱有效寄存器（CAN_MVAL）中的 MVAL 位和发送请求寄存器（CAN_TXREQ）中的 TXREQEN 位。

（1） CAN 内核的移位寄存器准备好加载数据；

（2） 在 IFx 寄存器和报文 RAM 之间没有数据传输。

当有多个待发送请求时，报文处理器会选择优先级最高的有效邮箱加载到移位寄存器中并启动发送。邮箱的 NDATA 位在报文被加载到移位寄存器后复位。

- 如果报文发送成功，并且自发送开始以来没有新数据写入此邮箱中，即 NDATA 始终为 0，发送请求位将被复位；
- 如果置位 TXIEN，那么在发送成功后，将置位 IPEND；
- 如果在发送过程中，CAN 发生错误或仲裁丢失，那么此报文将会在 CAN 总线再次空闲时重新发送。
- 如果在报文等待发送期间，有优先级更高的报文请求发送，则将按照报文的优先级顺序发送这些报文。

如果禁用了自动重传模式（即 CAN_CTRL[DISARETX]=1），那么接口寄存器组中的 IFx 报文控制寄存器（CAN_IFxMCTRL）中的 TXREQEN 和 NDATA 位配置如下：

表格 223 TXREQEN 和 NDATA 位配置

状态	TXREQEN	NDATA
启动发送	复位	不变
发送成功	不变	复位
发送失败（仲裁丢失或出错）	再次复位以重新启动发送	不变

此外，当接收到远程帧时，不需要为远程帧分配接收邮箱。此时，如果与远程帧匹配的发送邮箱中的 REMEN 置位，会自动触发的数据帧的发送。

40.5.18 发送/接收优先级

发送/接收优先级

在 CAN 模块中，邮箱编号决定邮箱的发送/接收优先级，CAN 标识符与优先级无关。邮箱的编号越小，优先级越高，反之，编号越大，优先级越低。因此邮箱 1 的优先级最高，邮箱 32 的优先级最低。

当存在多个发送请求时，CAN 模块会根据邮箱的优先级决定处理发送请求。因此，优先级最高的报文会被存储在编号最小的邮箱中。

接收过滤优先级

邮箱编号也决定了接收过滤的优先级。当对接收到的数据帧或远程帧进行接收过滤时，根据邮箱的编号，将对编号较小的邮箱进行接收过滤。因此，一旦数据帧或远程帧被一个邮箱接受，它将不会被另一个编号更大的其他邮箱接受。

可以将最后一个邮箱配置为接受任何未被其他邮箱接受的数据帧或远程帧，以确保 CAN 总线上的有需求的节点的报文通信能够完整记录。

40.5.19 数据发送

40.5.19.1 发送邮箱更新/更改

更新

更新发送邮箱时，需要注意以下几点：

（1） 发送邮箱中的数据字节可以被 CPU 通过 IF1 和 IF2 寄存器组随时更新，更新数据前不需要复位报文有效位 MVAL 和发送请求使能位 TXREQEN。

（2） 若只更新数据字节，可以按照以下步骤操作：
- 将 0x87 写入 IFx 命令寄存器（CAN_IFxCOM）的[23:16]位
- 将邮箱编号写入 IFx 命令寄存器（CAN_IFxCOM）的[7:0]位
- 更新数据字节并置位 TXREQEN 和 NDATA

（3） 即使只更新部分数据字节， IF1/IF2 数据寄存器 CAN_IFxDATAA 或 CAN_IFxDATAB（x=1/2）寄存中的四个字节在传输到邮箱之前都必须是有效的。为了保证数据的有效性可以通过以下两种方式写入：
- CPU 直接向 IF1/IF2 数据寄存器中写入所有四个字节
- 将邮箱当前的数据传输到 IF1/IF2 数据寄存器后， CPU 再写入新的数据字节到寄存器中

（4） 在更新数据时，必须同时置位 NDATA 和事件驱动的 CAN 通信中的 TXREQEN，以避免在更新过程中正在进行的发送操作中的 TXREQEN 被复位。当 NDATA 和 TXREQEN 都处于置位状态时，一旦启动新的发送过程，将会复位 NDATA。有关信息，详情请参见事件驱动报文发送章节。

更改

当可用的邮箱数量不足以仅用作永久邮箱时，可以对发送邮箱进行动态管理。

CPU 可以向接口寄存器中写入整个报文，包括仲裁，控制和数据字段。通过将 0xB7 写入 IFx 命令寄存器（CAN_IFxCOM）的[23:16]位，可以将整个报文的内容传输到邮箱中，并且操作前不需要复位 TXREQEN 和 NDATA。

当一个邮箱的发送请求已经在进行中且尚未完成，使用上述方式更改邮箱时，此邮箱会继续进行当前的发送过程；当发送过程由于受到干扰而中断时，不会重新启动发送过程。

若只更新当前正在发送报文的数据字节，通过将 0x87 写入 IFx 命令寄存器（CAN_IFxCOM）的[23:16]位。

注意：发送邮箱更新之后，邮箱中已更新和未更新的内容都会在接口寄存器组中进行备份。

40.5.20 数据接收

40.5.20.1 接收过滤

在向 CAN 内核的移位寄存器移入报文的仲裁位和控制位，即标识符，扩展标识符，远程传输请求和数据长度码后，报文处理器开始扫描报文 RAM，寻找与报文匹配的有效邮箱，按照以下步骤：

（1） 加载移位寄存器中的仲裁位：接收过滤单元从 CAN 内核的移位寄存器中加载仲裁位；

（2） 加载邮箱 1 的仲裁位和掩码位：接收过滤单元加载邮箱 1 的仲裁位和掩码位，包括 MVAL，MASKEN，NDATA 和 EOBEN。

（3） 比较前两个步骤中加载的位。

上述步骤将在后续的邮箱中重复进行，直到找到匹配的邮箱，或直到扫描到报文 RAM 的末尾。当找到匹配的邮箱后，停止扫描，报文处理器会根据接收到的帧类型（数据帧/远程帧）继续操作。

40.5.20.2 接收数据帧/远程帧

数据帧

报文处理器负责将 CAN 内核移位寄存器中的报文的内容存储到报文 RAM 中的对应邮箱中，包括数据字节，仲裁位和数据长度码。这种操作可以确保即使在使用了仲裁掩码寄存器的情况下，数据字节也能够与标识符关联。

接收到新数据（CPU 尚未读取的数据）后，置位 NDATA，在 CPU 读取邮箱后复位此位。当接收到新数据时，NDATA 已置位，则意味着先前的数据（CPU 尚未读取的数据）已经丢失，此时置位 MLOST 以指示报文丢失。如果接收中断已使能（置位 RXIEN），接收到数据时会置位 IPEND，中断寄存器会指向此邮箱从而触发中断。

为了防止在刚接收到所请求的数据帧时发送远程帧，接收到数据帧后，会复位此邮箱的 TXREQEN。

远程帧

接收远程帧时，匹配邮箱的三种配置情况如下表所示：

MDIRCFG 位用于配置报文方向，REMEN 位用于使能远程帧，MASKEN 位用于使能掩码。

表格 224 匹配邮箱配置

MDIRCFG	REMEN	MASKEN	操作
1	1	1/0	当接收到匹配的远程帧时，置位邮箱的 TXREQEN，邮箱的其余部分保持不变。
1	0	0	忽略接收到的远程帧，邮箱保持不变。

MDIRCFG	REMEN	MASKEN	操作
1	0	1	当接收到远程帧时，处理方式与数据帧相同。即接收到匹配的远程帧时，置位邮箱的 TXREQEN。CAN 内核的移位寄存器中的仲裁位和控制位，即标识符，扩展标识符，远程传输请求和数据长度码被存储在报文 RAM 对应的邮箱中，并置位此邮箱的 NDATA 位，邮箱的数据字节保持不变。

40.5.20.3 读取接收报文

报文处理器负责保证读取操作中数据的一致性，CPU 可以在任何时间通过 IFx 接口寄存器读取接收到的报文。将接收到的整个报文从报文 RAM 传输到接口寄存器组的典型读取步骤如下：

- 将 0x7F 写入 IFx 命令寄存器（CAN_IFxCOM）的[23:16]位
- 将邮箱编号写入 IFx 命令寄存器（CAN_IFxCOM）的[7:0]位

此外，将清除报文 RAM 中的 NDATA 位和 IPEND 位（接口寄存器中的 NDATA 位和 IPEND 位不受影响）。报文控制寄存器中这些位的值始终保持复位前的状态。当邮箱使用掩码进行接收过滤时，仲裁位会显示哪些匹配的报文已被接收。

NDATA 的实际值和 MLOST 实际值的含义：

（1） NDATA 实际值：自上次读取此邮箱后，是否接收到新的报文；

（2） MLOST 实际值：自上次读取此邮箱后，是否接收到多条报文，从而导致报文丢失。MLOST 不会自动复位。

40.5.20.4 新数据接收请求

通过置位邮箱的 TXREQEN，CPU 可以向其他 CAN 节点发送一个远程帧来请求接收新数据。此远程帧携带接收邮箱的标识符，会触发另一个 CAN 节点发送具有与标识符匹配的数据帧的传输。FIFO 在远程帧发送前，已经接收到匹配的数据帧，则自动复位邮箱的 TXREQEN。

通过将 0x84 写入 IFx 命令寄存器（CAN_IFxCOM）的[23:16]位，可以在不改变邮箱内容的情况下置位 TXREQEN。

40.5.21 FIFO 缓冲区

若干个邮箱可以组成一个 FIFO 缓冲区，也可以组成多个 FIFO 缓冲区。

存储接收报文

每个 FIFO 被用于存储接收到的具有特定标识符的报文。在同一个 FIFO 缓冲区中，所有邮箱的仲裁和掩码寄存器配置相同。FIFO 缓冲区中的最后一个邮箱必须配置为 1，以表示块结束，而除最后一个邮箱外的其他所有邮箱的 EOBEN 位都必须配置为 0。

当接收报文与 FIFO 缓冲区中的标识符匹配时，此接收报文将按照邮箱编号从小到大的顺序依次存储在 FIFO 缓冲区的邮箱中。存储到邮箱的同时，置位此邮箱的 NDATA，表示接收到了新数据。在邮箱的 EOBEN 为 0（即此邮箱不是 FIFO

缓冲区的最后一个邮箱）时置位 NDATA，直到 CPU 清除 NDATA 位，将锁定此邮箱以禁止报文处理器进行后续的写入操作

FIFO 按照邮箱编号从小到大的顺序存储报文，直到报文填充到此缓冲区的最后一个邮箱。如果所有的邮箱都处于锁定状态(即 NDATA 置位)，由于 FIFO 缓冲区的最后一个邮箱的 EOBEN 为 1，表示该邮箱是最后一个可用邮箱，因此随后的报文只能被写入 FIFO 缓冲区的最后一个邮箱中并覆盖此邮箱中之前存储的（尚未被 CPU 读取的）报文。

读取 FIFO 缓冲区

在一组邮箱构成的一个 FIFO 缓冲区中，这组邮箱可以存储多条报文，用户程序需要清空缓冲区以避免数据丢失。长度为 N 的 FIFO 缓冲区中存储了两部分内容：

- N-1 条报文
- 自动上次清空以来，接收的最后一条报文

从 FIFO 缓冲区的邮箱中读取报文并复位 NDATA 位的操作方式与单个邮箱相同。在一个子程序中，按照邮箱编号从小到大的顺序，通过读取和并复位所有邮箱的 NDATA 位，以清空 FIFO 缓冲区。中断驱动的 CPU 处理 FIFO 缓冲区的子程序流程步骤如下：

（1） 收到邮箱中断，启动流程；

（2） 读取中断标识符；

（3） 根据中断标识符值进行条件判断：
- 值为 0x0000，流程结束
- 值为 0x0800，进入状态改变中断处理程序
- 其他值，进行下一步；

（4） 将 0x007F 写入 CAN_IFxCOM[31:16]，邮箱编号等于中断标识符的值；

（5） 将邮箱编号写入 CAN_IFxCOM 寄存器，触发报文传输到 IF1/IF2 寄存器组，并清除 NDATA 和 IPEND。

（6） 读取 IF1/IF2 报文控制字段

（7） 判断 NDATA 值
- 若 NDATA=1，表示由待读取的新数据，进行下一步
- 若 NDATA=0，表示没有新数据，从步骤（2）重新开始

（8） 读取 IF1/IF2 中的数据寄存器 CAN_IFxDATAA 和 CAN_IFxDATAB 中的数据

（9） 判断 EOBEN 值
- 若 EOBEN=1，表示当前数据块的所有数据已读取，从步骤（2）重新开始读取中断标识符
- 若 EOBEN=0，表示还有当前数据块还有待读取的数据，返回步骤（5）开始进行下一个邮箱的读取操作

注意：必须首先读取并清除 FIFO 缓冲区中的所有邮箱，然后再存储下一批报文。若不按照这个顺序操作，由于缓冲区中已被读取的邮箱将根据优先级从高到低进行填充，因此无法正确实现 FIFO 功能。

40.5.22 位时序

CAN 支持 1kBit/s 到 1000kBit/s 的比特率。CAN 模块的时钟信号通常由晶体振荡器提供，每个 CAN 节点都有自己的时钟信号。因此，每个 CAN 节点的位时序参数可以单独配置。即使 CAN 节点的振荡器周期(F_{osc})可能不同，也可以产生相同的比特率。当温度或电压的改变以及恶化的元件都会引起振荡器频率微小的波动，因此，振荡器频率不是绝对稳定的。如果变化的范围比较小（不超过容忍范围 df），CAN 节点是可以通过重新同步到位流来满足位速率的需求。

CAN 位同步可以纠正 CAN 位时序的配置错误，降低错误帧的产生频率。但是，在仲裁时，如果一个帧被两个或两个以上节点同时发送时，错位的采样点可能会导致其中一个发送节点产生被动错误。

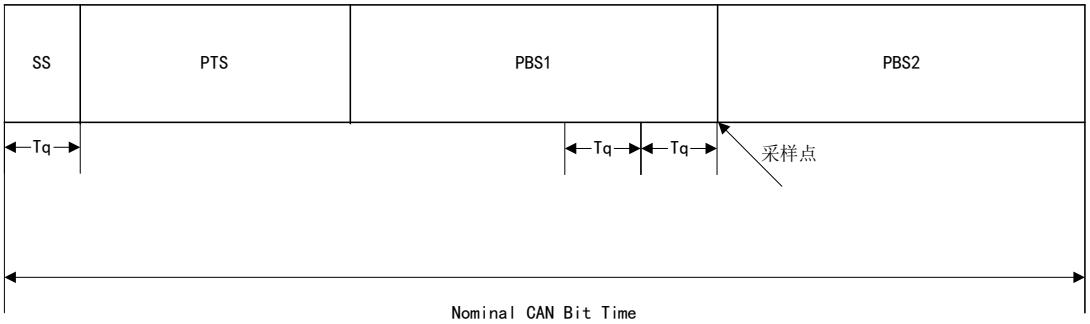
在许多情况下，CAN 位同步将修正 CAN 位时序的错误配置，使其只在偶尔产生错误帧。然而，在仲裁的情况下，当两个或多个 CAN 节点同时试图传输一个帧时，一个错位的采样点可能会导致其中一个发送器成为错误的隐性。

深入了解 CAN 节点内部的 CAN 位同步以及 CAN 节点在 CAN 总线上的交互有助于分析这种偶发错误。即使 CAN 位时序配置中存在微小的错误，也可能会对 CAN 网络的性能产生重大影响，尽管这种微小的错误不会立即导致故障。

位时序构成

位时序的基本时间单位是时间量子（Tq）。根据 CAN 协议规范，位时序细分为同步段（SS），传播时间段（PTS），相位缓冲段 1（PBS1）和相位缓冲段 2（PBS2），每个时间段都由特定数量的时间量子构成。同步段的长度固定，为一个时间量子，其他时间段的时间量子数可配置。如下图所示：

图 252 位时序



时间量子（Tq）的计算公式如下：

$$\mathrm{Tq} = \frac{(\text{波特率预分频})}{f_{CANCLK}}$$

其中：

波特率预分频：由 BRPSCEXT 和 BRPSC 构成，可配置 1~1024 之间的分频。

f_{CANCLK}：CAN 时钟频率。

对于给定的比特率，可能有多种位时序配置满足要求。位时序中各时间段的最小可配置范围如下表所示：

表格 225 位时序可配置范围

参数	描述	可配置范围
同步段	用于同步总线输入到 CAN_CLK	1* Tq（）
传播时间段	用于补偿物理延迟时间	（1~8）* Tq
相位缓冲段 1	同步过程中可以暂时延长	（1~8）* Tq
相位缓冲段 2	同步过程中可以暂时缩短	（1~8）* Tq
同步跳变宽度	此参数的时间不得大于任何一个相位缓冲段	（1~4）* Tq

注意：为了确保 CAN 网络稳定运行，必须考虑振荡器的容差范围和物理延迟时间。

40.5.22.1 同步段（SS）

在 CAN 位时序中，CAN 总线电平边沿跳变预期发生在同步段。当边沿跳变发生在同步段以外时，那么它与同步段之间的距离称为该边沿跳变的相位误差。

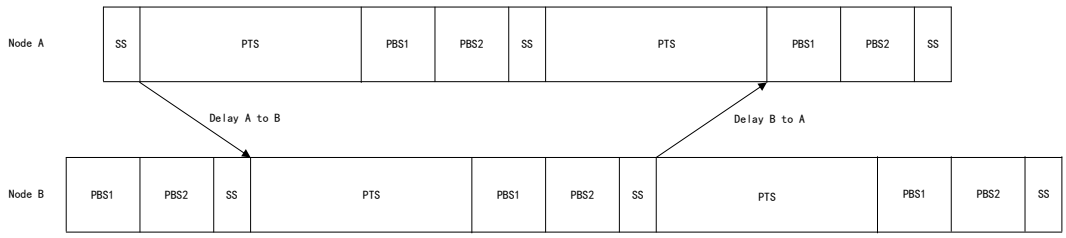
40.5.22.2 传播时间段（PTS）

在 CAN 位时序中，传播时间段用于补偿 CAN 网络的物理延迟时间，包括 CAN 节点内部的延迟时间和总线上信号传播时间。

由于两个节点之间的距离不同，信号传播到各个节点的时间也不同。因此，任何与 CAN 总线比特流同步的 CAN 节点都可能与发送比特流的节点不同相。

在 CAN 协议中，非破坏性的位级仲裁机制以及接收节点所提供的显性应答位要求，每个发送比特流的 CAN 节点，也必须能够接收其他节点发送的，与此比特流同步的显性位。两个 CAN 节点之间的相移和传播时间示例下图所示：

图 253 传播时间段



其中：

A 到 B 的延迟：Delay A to B≥节点 A 输出延迟+（A 到 B）总线延迟+节点 B 输入延迟；

传播时间段≥Delay A to B+ Delay B to A

示例中的具体分析情况如下：

（1） 延迟和总线仲裁

- 节点 A 和节点 B 在 CAN 总线上执行仲裁，并同时作为数据的发送方；
- 节点 A 比节点 B 发送起始帧（显性位）的时间稍早，稍早的时间小于一个位时序；
- 节点 B 接收到来自节点 A 的隐性位到显性位的边沿跳变时，立即进行同步；
- 由于节点 B 是在节点 A 发送显性位后，才接收到边沿跳变延迟（Delay A to B），因此节点 B 的位时序段相较于节点 S 发生了偏移；
- 当节点 B 发送一个具有较高优先级的标识符时，如果恰好在节点 A 发送隐性位的同时节点 B 发送显性位，节点 B 将在此标识符位上赢得仲裁。此时节点 B 发送的显性位经过延迟（Delay B to A）后被节点 A 接收。

（2） 振荡器容差

- 由于振荡器的容差，节点 A 的采样点实际位置可能位于其相位缓冲段的标称范围内的任何位置；
- 因此，为了确保节点 A 能够正确采样到节点 B 发送的位，节点 B 发送的跳变边沿必须在节点 A 的相位缓冲段 1 开始之前到达节点 A。因此，传播时间长度必须满足上述要求。

（3） 采样错误

- 如果节点 B 发送的边沿跳变在节点 A 的相位缓冲段 1 开始后才到达，那么节点 A 在采样点可能会错误地采样到一个隐性位，进而导致位错误，并产生一个错误标志破坏当前帧。
- 仅当两个节点都在总线上尝试仲裁时才会产生上述错误，并且这两个节点的振荡器频率在容差范围的两端，通过一个较长的总线隔开。因此，位时序中，可能会由于传播时间（PTS）配置太短而导致偶发的总线错误。

40.5.22.3 相位缓冲段和同步跳变宽度

振荡器容差通过相位缓冲段和同步跳变宽度进行补偿。

相位缓冲段（PBS1 和 PBS2）

相位缓冲段位于采样点两侧，在同步过程中，相位缓冲段可以根据需要被延长或缩短

同步跳变宽度(SJW)

同步跳变宽度用于定义重新同步机制可以在相位缓冲段内调整采样点位置的最大幅度，以补偿边沿相位误差。

边沿检测

为了检测边沿，需要在每个时间量子对总线电平进行采样，并将当前采样的总线电平与上一个采样的总线电平进行比较。同步过程发生在隐性位到显性位的边沿跳变时，用于控制边沿到采样点之间的距离。因此，只有在上一个采样点采样到隐性电平，并且当前时间量子的总线电平为显性时，才能进行同步过程。

同步边沿与相位误差

（1） 当在同步段内产生边沿时，这个边沿被定义为同步边沿；

（2） 当在同步段外产生边沿时，这个边沿到同步段之间的距离被定义为边沿相位误差，以时间量子为单位测量。

- 相位误差为正：边沿产生在同步段之后
- 相位误差为负：边沿产生在同步段之前

硬同步和重新同步

同步方式分为硬同步和重新同步。

（1） 硬同步：硬同步通常发生在帧传输开始前。在硬同步后，不考虑边沿相位误差，同步段结束后会重新启动位时序，这意味着强制触发硬同步的边沿位于重新启动的位时序的同步段内。

（2） 重新同步：在帧传输时，只允许触发重新同步。重新同步通过缩短或延长位时间以调整采样点相对于边沿的距离。

- 相位误差为正：延长 PBS1。若相位误差小于同步跳变宽度，则 PBS1 延长相位误差的宽度，反之，PBS1 延长同步跳变的宽度；
- 相位误差为负：缩短 PBS2。若相位误差小于同步跳变宽度，则 PBS2 缩短相位误差的宽度，反之，PBS2 缩短同步跳变的宽度；

相位误差和同步跳变宽度的关系对同步的影响：

- 当相位误差的幅度≤同步跳变宽度时，硬同步和重新同步的结果相同。
- 当相位误差的幅度＞同步跳变宽度时，重新同步的结果不能完全补偿相位误差，此时仍然存在大小为(相位误差 - 同步跳变宽度)的误差。

在两个采样点之间，只能进行一次同步。同步操作使边沿与采样点之间保持最小距离，这为总线电平稳定提供了时间，并且过滤掉了那些短于（PTS 第一相位段 PBS1）的尖峰噪声。

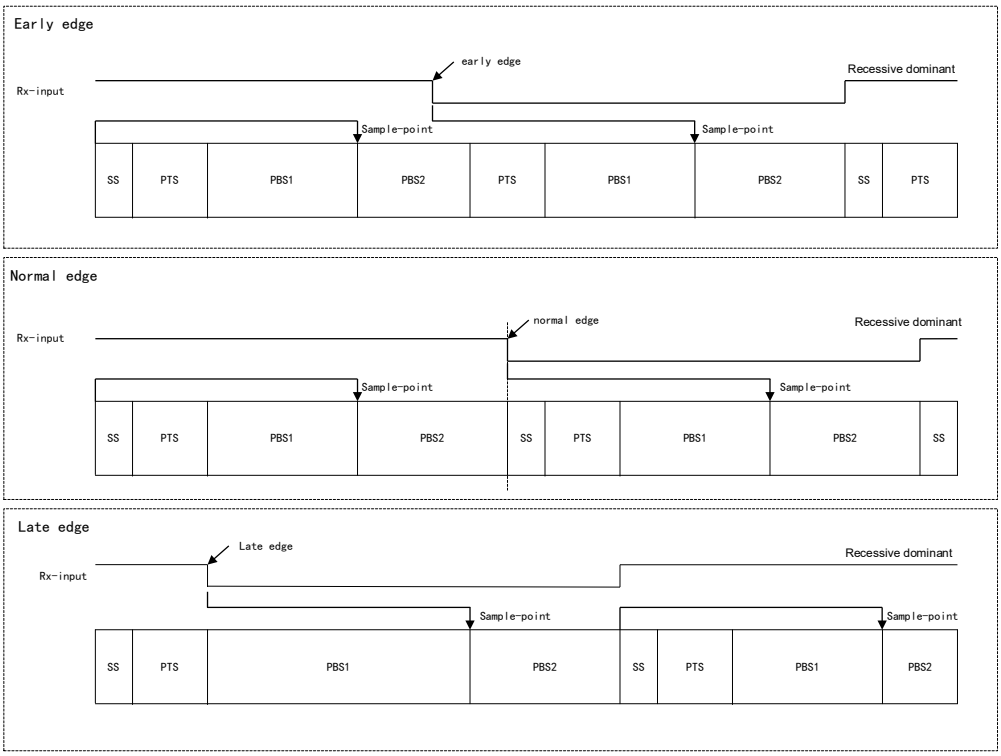
产生同步的原因

仲裁会产生同步，这是再大多数情况下，除了尖锐噪声外产生同步的原因。所有发送数据的节点会在最先开始发送的节点的跳变边沿上会产生硬同步。但是由于传输会延迟，实际上所有节点不会做到同步。最先开始发送的节点（“领先”节点）不一定会赢得仲裁，因此节点的接收器必须将“领先”节点与随后“领先”的其他节点同步，这些发送节点与之前“领先”节点不同步。应答域：在应答域中也会发生与仲裁类似的情况：发送节点和一些接收节点必须同步到“领先”的，发送显性应答位的接收节点。

振荡器容差：仲裁结束后，振荡器容差会触发同步过程。由于在两次同步之间的时间内（最多 10 位），发送节点和接收器的振荡器时钟周期的差异会累加起来。这些累积的差异不能超过同步跳变宽度，从而限制了振荡器的容差范围。

使用相位缓冲段补偿相位误差示例如下图所示：

图 254 在“早”边沿和“晚”边沿上的同步



三个图分别表示“早”边沿，无同步和“晚”边沿上的同步。同步时，只是临时对相位缓冲段进行延长或缩短，在下一个位时序中，同步段的长度将返回预设的标称值。

“早”边沿同步

如上图所示，（从隐性到显性的）边沿跳变发生在相位缓冲段 2（PBS2）上，早于同步段（SS），因此是“早”边沿。当检测到“早”边沿时，为了使边沿跳变到采样点的距离与没有发生边沿跳变时从同步段（SS）到采样点的距离相同，将缩短 PBS2 并忽略 SS。由于此“早”边沿的相位误差小于同步跳变宽度，因此可以被完全补偿。

“晚”边沿同步

如上图所示，（从隐性到显性的）边沿跳变发生在传播时间段（PTS）的末尾，晚于同步段（SS），因此是“晚”边沿。当检测到“晚”边沿时，为了使边沿跳变到采样点的距离与没有发生边沿跳变时从同步段（SS）到采样点的距离相同，将延长 PBS1。由于此“晚”边沿的相位误差小于同步跳变宽度，因此可以被完全补偿。由于这种补偿，在下一个 SS（一个标称位时间长度）中将产生的，从显性到隐性的边沿跳变。

在上述示例中，从 CAN 实现的状态机的角度分析位时序，其中位时间从采样点开始到采样点结束。在进行“早”边沿同步时，将忽略 SS。因为一旦在 PBS2 中检测到边沿跳变，状态机无法将该时间量子重新指定为 SS。

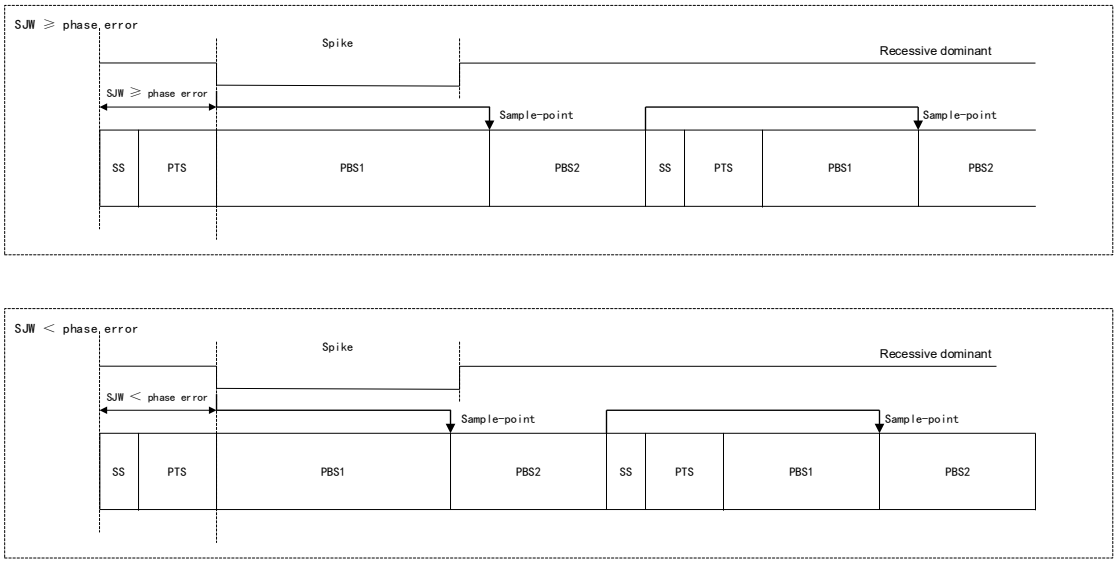
由于状态机无法在随后将边沿发生的那个相位缓冲段 2 的时间量子重新定义为同

步段。

显性尖峰过滤

使用同步过滤短尖峰噪声如下图所示，其中尖峰始于 PTS 末尾，长度为（PTS+PBS1）

图 255 短显性尖峰过滤



（1） 当 SJW≥相位误差：采样点被移动到尖峰后，采样到的电平是隐性的（“1”）。

（2） 当 SJW＜相位误差：由于采样点不能被移动到足够远的地方，因此采样到的实际总线电平为显性尖峰。

40.5.22.4 振荡器容差

振荡器容差范围（振荡器频率 f_{osc} 相较于标称频率 f_{nom} 的范围）如公式所示：

$$(df) \cdot f_{nom} \leq f_{osc} \leq (1 + df) \cdot f_{nom} \tag{1}$$

其中：

df：容差范围

f_{osc}：振荡器频率

f_{osc}：标称频率

最大公差取决于相位缓冲段 1，相位缓冲段 2，同步跳变宽度和位时间的比例，必须同时满足以下两个要求

$$df \leq \frac{\min(T_{PBS1}, T_{PBS2})}{2\big((13 \times bit\ time) - T_{PBS2}\big)} \tag{2}$$
$$df \leq \frac{SJW}{20 \times bit\ time}$$

SJW：同步跳变宽度

T_{PBS1}，T_{PBS2}：相位缓冲段时间长度

注意：

（1） SJW 必须小于等于相位缓冲段中较小的部分。

（2） 可用于相位缓冲段的位时间受到传播时间段的限制。

例如，当 PTS=1，PBS1= PBS2=SJW=4 时，根据公式（2），振荡器容差最大值为 1.58%。在这种配置中，传播时间段仅占位时序的 10%，因此不适用于位时间较短的情况。这种配置适用于高达 125 kBit/s（位时间为 8μs）的位速率，且总线长度为 40 米的使用场景。

40.5.22.5 位时序配置

CAN 模块中，位时序配置通过位时序寄存器（CAN_BTIM）中两个字节实现：

（1） TSSPVP1 和 TSSPVP2 构成一个字节，TSSPVP1 表示 PTS 和 PBS1 的总和，TSSPVP2 表示 PBS2

（2） SJWP 和 BRPSC 构成一个字节。此外，位时序寄存器还提供了用于波特率预分频扩展的字节 BRPSCEXT，与 SJWP 和 BRPSC 共同构成一个字节。

位时序寄存器中的数据用于配置 CAN 协议控制器的输入，其中， TSSPVP1，TSSPVP2 和 SJWP 配置的位时序逻辑电路定义了位时序中的时间量子数量，BRPSC 和 BRPSCEXT 构成的波特率预分频用于定义时间量子的长度，时间量子是位时序的基本时间单位。

在位时序寄存器中，TSSPVP1，TSSPVP2，SJWP 和 BRPSC 的位域值与功能值的对应关系如下表所示：

表格 226 位域值与功能值对应关系

功能值	位域值
1	0
2	1
…	…
n	n-1

因此，位域值的范围为 0~n-1，而不是 1~n。举例来说，当 SJWP 的功能值的范围为 1~4 时，位域值范围为 0~3，只需用两位表示 SJWP 的位域值。

位时间长度的写入值和功能值分别如下所示：

（1） （写入值）位域值：[TSSPVP1+TSSPVP2+3]*Tq

（2） 功能值：[SS+PTS+PBS1+PBS2] *Tq

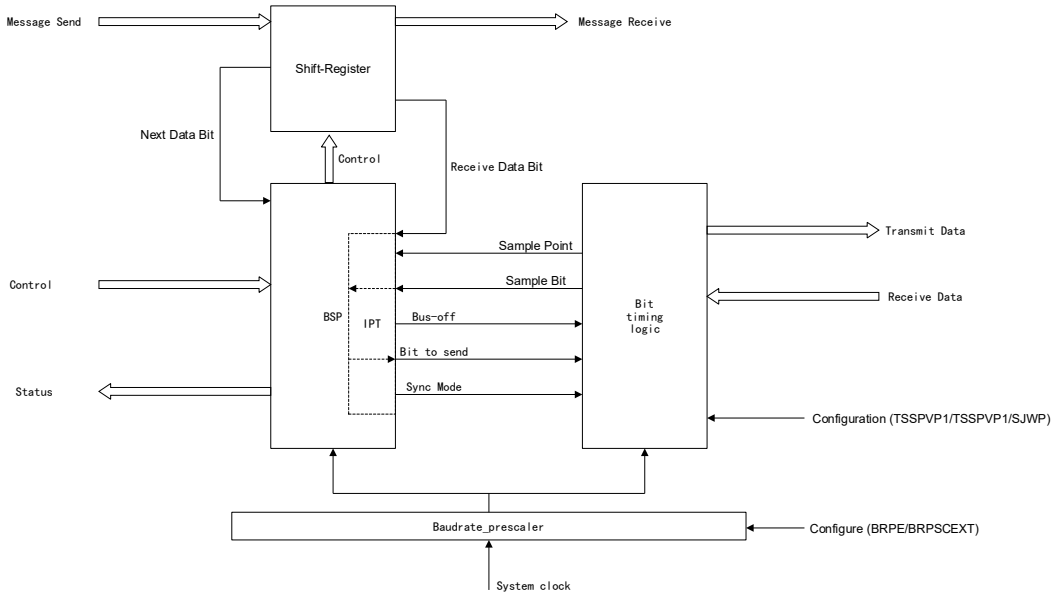
CAN 协议控制器的位时序状态机负责控制位时间的处理， 计算采样点的位置和

间歇性的同步。

位流处理器（BSP）

CAN 协议控制器的位流处理器（BSP）状态机在每个位时间的采样点计算一次。

图 256　CAN 协议控制器结构



BSP 的功能如下：

（1）　帧转换：BSP 可以将报文转换成帧，也可以将帧转换成报文；

（2）　管理固定格式位：BSP 能够生成或丢弃数据的固定格式的边界位；

（3）　填充位插入或提取：BSP 在必要时插入填充位，并在接收时识别并提取这些填充位；

（4）　计算和校验 CRC 码：BSP 负责计算发送时的 CRC 校验码，并在接收时检查 CRC 校验码，以确保数据的完整性；

（5）　执行错误管理；

（6）　选择同步类型：BSP 决定使用哪种类型的同步方式来保持时钟同步

（7）　处理总线输入位：BSP 在采样点进行评估，并处理采样的总线输入位。

（8）　计算信息处理时间(IPT) ：在采样点后，BSP 需要计算发送下一个位（例如数据位，CRC 位，填充位，错误标志或空闲状态）所需的时间，即信息处理时间（IPT），在 CAN 中，这个时间被定义为 0 时间量子（Tq）。

IPT 的值对于不同的 CAN 控制器可能会有所不同，但其长度不得大于 2*Tq。此外，IPT 的长度是 PBS2 所配置长度的下限，在进行同步时，PBS2 的实际值可能会小于 IPT，但总线时序不会受到影响。

40.5.22.6 位时序各参数计算

计算位时序配置通常从所需的位速率或位时间开始，计算得出的位时间必须是 CAN 时钟周期的整数倍。位时间的范围为 8~25 个时间量子（Tq），时间量子的计算公式详情请参见位时序章节。位时间与位速率的关系如以下公式所示：

$$\text{位时间} = \frac{1}{\text{位速率}}$$

可以通过多种组合实现所需的位时间，并可以通过以下步骤进行重复迭代：

（1） 计算传播时间段 （PTS）长度：PTS 长度由系统中测量的延迟时间决定，对于可扩展 CAN 总线系统，必须定义最大总线长度和最大节点延迟时间。计算出的 PTS 长度必须转化成 Tq 的整数倍（向上取整）。

（2） 计算同步段（SS）长度：SS 长度为固定值 1*Tq。

（3） 计算相位缓冲段（PBS1 和 PBS2）长度：相位缓冲段总长度为（位时间-PTS- SS）*Tq。

- 剩余值为偶数个 Tq 时：PBS1=PBS2
- 剩余值为奇数个 Tq 时：PBS2=PBS1+1

此外，由于 PBS2 必须大于等于网络中任何 CAN 控制器的信息处理时间，信息处理时间由设备决定，因此需考虑 PBS2 的最小标称长度范围，即（0~2）*Tq。

（1） 计算同步跳变宽度（SJW）长度：必须将 SJW 长度设置为最大值，即 PBS1 长度和 4 中的最小值。

（2） 根据以上步骤得出的各参数，计算振荡器容差范围，详情请参考振荡器容差章节。当位时间可以通过多种组合实现时，建议选择振荡器容差最大的组合。所有节点中，振荡器容差范围最小的节点限制了 CAN 系统的振荡器容差范围。

当 CAN 节点的时钟不同时，需要采用不同的位时序配置，以实现相同的位速率。在 CAN 网络中，基于最大延迟的节点计算 CAN 网络的延迟时间。在整个 CAN 网络中，只进行一次延迟时间计算，

以上计算表明，必须降低位速率或减小总线长度或提高振荡器频率的稳定性以实现符合协议的 CAN 位时序。将通过以上步骤得出的各参数配置，写入位时序寄存器：

(PBS2-1)&(PBS1+PTS-1)&(SJW-1)&(PSC-1)

注意：CAN 运行在 1MBit/s 速率时所需的最低 CAN 时钟频率为 8MHz。

计算示例-高波特率

此示例中，假设 f_{CANCLK}=10MHz，BRPSC =0，位速率=1 MBit/s。

表格 227 高波特率位时序配置示例

参数	计算公式	计算结果
Tq	tCANCLK	100 ns
总线延迟（40m）	-	220 ns
总线驱动延迟	-	90 ns
接收电路延迟	-	40 ns
tPTS	2*延迟 =2*（220+90+40）ns=700ns=7* Tq	700 ns
tSS	1* Tq	100 ns
tSJW	1* Tq	100 ns
tPBS1	tPTS+ tSJW	800ns
tPBS2	tIPT+1* Tq	100ns
位时间	tSS + t PBS1 + t PBS1	1000ns
振荡器容差	min(tPBS1 , tPBS2)/2((13×bit time)-tPBS2)=100ns/2(13*1000ns-100ns)	0.35%

因此，写入位时序寄存器中的时间参数为$(4-1)_3$&$(8-1)_4$&$(1-1)_2$&$(1-1)_6$，即向位时序中写入0x0000 0700。

计算示例-低波特率

此示例中，假设 f_{CANCLK}=2 MHz，BRPSC =1，位速率=100 KBit/s。

表格 228 低波特率位时序配置示例

参数	计算公式	计算结果
Tq	2*tCANCLK	1 μs
总线延迟（40m）	-	80 μs
总线驱动延迟	-	200 μs
接收电路延迟	-	220 μs
tPTS	1* Tq	1 μs
tSS	1* Tq	1 μs
tSJW	4* Tq	4 μs
tPBS1	tPTS+ tSJW	5 μs
tPBS2	tIPT+4* Tq	4 μs
位时间	tSS + t PBS1 + t PBS1	10 μs
振荡器容差	min(tPBS1 , tPBS2)/2((13×bit time)- tPBS2)= 4 μs /2(13*10μs -4μs)	1.58%

因此，写入位时序寄存器中的时间参数为$(4-1)_3$&$(5-1)_4$&$(4-1)_2$&$(2-1)_6$，即向位时序中写入0x0000 34C1。

40.5.23 报文接口寄存器组

CAN 中一共有三个接口寄存器组，用于控制 CPU 对报文 RAM 的读取访问和写入访问。其中：

- IF1 和 IF2 寄存器组用于读取访问和写入访问
- IF3 寄存器组仅用于读取访问

由于报文 RAM 的结构限制，不能直接修改邮箱中的单个位或字节。当需要更新邮箱中的部分内容时，必须对报文 RAM 中的整个邮箱进行访问。因此，报文处理器需要对从 IF1/IF2 寄存器到报文 RAM 之间的数据传输执行读取-修改-写入操作。对于邮箱中需要保持不变的部分，将这部分内容从报文 RAM 读取到接口寄存器组 IF1 和 IF2 中，并在更新之后将接口寄存器组的全部内容写入邮箱中。

部分写入：在对邮箱进行部分写入后，接口寄存器组中未被命令寄存器选择的部分将设置为所选邮箱的实际内容。

部分读取：在对邮箱进行部分读取入后，接口寄存器组中未被命令寄存器选择的部分将保持不变。

数据缓存

为了避免并发的 CAN 接收和发送报文和 CPU 访问报文 RAM 之间的冲突，对待传输的数据进行缓存。在报文 RAM 和 IF1/IF2 寄存器组之间的一次传输中，可以完整传输一个邮箱的全部内容或邮箱中的部分内容。通过并行传输所有选定部分的数据，从而保证 CAN 报文中的数据保持一致。

写访问丢失

原因：在接收同一邮箱的报文时，主机报文处理器的读-修改-写访问之间发生了 IFx 寄存器对报文 RAM 的写入。因此，当被访问邮箱的 MAILEN=1（表示此邮箱使能），并且 CAN 通信正在进行时，则从 IFx 寄存器到报文 RAM 的之间数据传输可能会丢失。

如何避免写访问丢失：

- 对于接收邮箱，在更改 MIDCFG，EXTIDCFG，MDIRCFG，DLCCFG，RXIEN，TXIEN，REMEN，EOBEN，MASKEN，IDMASK，EXTMASK 和 DIRMASK 字段前复位 MsgVal
- 对于发送邮箱，在更改 MDIRCFG，RXIEN，TXIEN，REMEN，EOBEN，MASKEN，IDMASK，EXTMASK 和 DIRMASK 字段前复位 MsgVal

除上述列出的字段外，更改其他字段时不会发生对报文 RAM 的写访问丢失。

40.5.23.1 IF1/2

IF1 和 IF2 寄存器组可以与邮箱进行数据交互。这些寄存器接口可以传输整个邮箱或邮箱中的各个部分。

寄存器组中的 IFx 命令寄存器（CAN_IFxCOM）负责控制数据传输的方向：

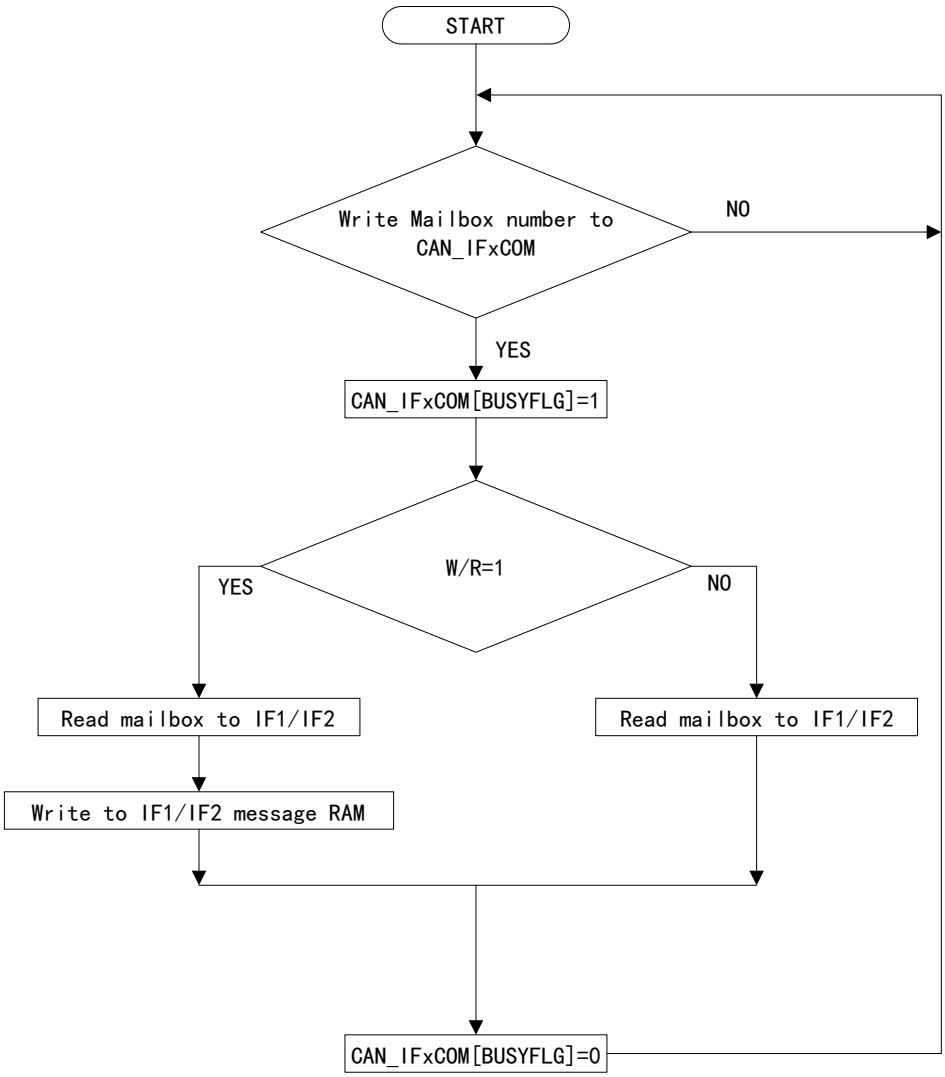
（1） 若 CAN_IFxCOM 寄存器被设置为读，则被 CAN_IFxCOM 寄存器选中的邮箱字段将从邮箱复制到其他 IFx 寄存器；

（2） 若 CAN_IFxCOM 寄存器被设置为写，则被 CAN_IFxCOM 寄存器选中的邮箱字段将被来自其他 IFx 寄存器的数值覆盖。

将目标邮箱的编号写入 CAN_IFxCOM 寄存器的[7:0]以启动数据传输。

当 CPU 启动 IF1/IF2 寄存器和报文 RAM 之间的数据传输时，报文处理器会将相应 CAN_IFxCOM [BUSYFLG]置 1。一旦传输完成，CAN_IFxCOM [BUSYFLG]位将被重新置 0，如下图所示：

图 257 数据交互流程



40.5.23.2 IF3

IF3 寄存器中，可以为每个邮箱配置自动更新功能（详情请参见 CAN_IF3UDEN 寄存器）。IF3 寄存器组可以在不需要由 CPU 启动报文 RAM 传输的情况下，将接收到的邮箱自动更新到寄存器中。

在报文 RAM 的所有有效邮箱中，当使能邮箱的自动更新功能时，将检查这些邮箱中的 NDATA 标志是否被激活。若检测到 NDATA 标志激活的邮箱，且没有正在

进行的 DMA 传输时，那么此邮箱将会被传输到 IF3 寄存器中，由 IF3 观察寄存器（CAN_IF3OBS）进行控制。若检测到多个 NDATA 标志激活的邮箱，自动更新的优先级与邮箱编号有关，邮箱编号越低，优先级越高，反之，邮箱编号越高，优先级越低。邮箱的传输到 IF3 寄存器后，邮箱中的 NDATA 标志会复位。

通过置位 CAN_CTRL[IFDEN3]，以使能 CAN 的 DMA 功能。CAN 内部的 IF3 寄存器组更新完成后，会触发 DMA 请求，并且 DMA 请求会一直保持激活状态，直到首次对 IF3 寄存器中的任一寄存器进行读取访问。

注意： IF3 寄存器组仅用于读取访问，不能用于将数据发送到邮箱中。

40.6 寄存器组地址

表格 229 CAN 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
CanaRegs	CAN_REGS	0x5000 2000	0x5000 2FFF
CanbRegs	CAN_REGS	0x5010 1800	0x5010 1FFF

40.7 寄存器地址映射

表格 230 CAN_REGS 偏移地址

寄存器名称	寄存器描述	偏移地址	WRPRT
CAN_CTRL	控制寄存器	0x00	-
CAN_EFLG	错误标志寄存器	0x04	-
CAN_ECNT	错误计数寄存器	0x08	-
CAN_BTIM	位时序寄存器	0x0C	-
CAN_INT	中断寄存器	0x10	-
CAN_TEST	测试寄存器	0x14	-
CAN_PE	奇偶校验错误寄存器	0x1C	-
CAN_RAMINIT	RAM 初始化寄存器	0x40	-
CAN_GLBINTEN	全局中断使能寄存器	0x50	-
CAN_GLBINTFLG	全局中断标志寄存器	0x54	-
CAN_GLBINTCLR	全局中断清除寄存器	0x58	-
CAN_ABOTMR	Auto-Bus-On 定时寄存器	0x80	-
CAN_TXREQFLG	发送请求标志寄存器	0x84	-
CAN_TXREQ	发送请求寄存器	0x88	-
CAN_NDATAFLG	新数据标志寄存器	0x98	-
CAN_NDATA	新数据寄存器	0x9C	-
CAN_IPENDFLG	中断挂起标志寄存器	0xAC	-

寄存器名称	寄存器描述	偏移地址	WRPRT
CAN_IPEND	中断挂起寄存器	0xB0	-
CAN_MVALFLG	邮箱有效标志寄存器	0xC0	-
CAN_MVAL	邮箱有效寄存器	0xC4	-
CAN_INTMAIL	中断邮箱寄存器	0xD8	-
CAN_IFxCOM（x=1…2）	IFx 命令寄存器	0x100+(x-1)*0x40	-
CAN_IFxMASK（x=1…3）	IFx 掩码寄存器	0x104+(x-1)*0x40	-
CAN_IFxARB（x=1…3）	IFx 仲裁寄存器	0x108+(x-1)*0x40	-
CAN_IFxMCTRL（x=1…3）	IFx 报文控制寄存器	0x10C+(x-1)*0x40	-
CAN_IFxDATAA（x=1..3）	IFx 数据 A 寄存器	0x110+(x-1)*0x40	-
CAN_IFxDATAB（x=1…3）	IFx 数据 B 寄存器	0x114+(x-1)*0x40	-
CAN_IF3OBS	IF3 观察寄存器	0x140	-
CAN_IF3UDEN	IF3 更新使能寄存器	0x160	-

40.8 寄存器功能描述

40.8.1 控制寄存器（CAN_CTRL）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	INIT	R/W	初始化模式（Initialization Mode） 在 Bus-off 事件期间自动设置此位，此位用于在报文 RAM 初始化和位时序配置期间，控制 CAN 模块保持非活动状态。总线恢复时间不会因清除此位而缩短。 0：正常处理报文 1：忽略总线活动	0h
1	IEN0	R/W	使能中断线 0（Interrupt line 0 Enable） 使能后，当有中断产生时，CANINT0 将置 1 并保持活动状态，直到处理完中断事件。 0：禁用 CANINT0 1：使能 CANINT0	0h
2	SCHIEN	R/W	使能状态改变中断（Status Change Interrupt Enable） 该位指示 TXCFLG，RXCFLG 和 LEC 能否在 CANINT0 中断线上产生中断。 0：禁用，不可以在 CANINT0 上产生中断 1：使能，此时可以在 CANINT0 上产生中断	0h
3	ERRIEN	R/W	使能错误中断（Error Interrupt Enable） 该位指示 PEFLG，BUSOFLG 和 EWFLG 能否在 CANINT0 中断线上产生中断。 0：禁用，不可以在 CANINT0 上产生中断 1：使能，此时可以在 CANINT0 上产生中断并影响 CAN_INT 寄存器	0h

位/域	名称	R/W	描述	复位值
4	保留			0h
5	DISARETX	R/W	禁止自动重传（Disable Automatic ReTransmit） 0：使能自动重传，报文将一直自动重传直到发送成功 1：禁止自动重传	0h
6	CFGWACCEN	R/W	使能配置写访问（Configure write access Enable） 0：CPU 不可以对配置寄存器进行写入访问 1：当 INIT 置 1 时，CPU 可以对配置寄存器进行写入访问	0h
7	TESTEN	R/W	使能测试模式（Test Mode Enable） 0：禁用（正常操作） 1：使能	0h
8	IDBGSEN	R/W	使能中断调试支持（Interrupt Debug Support Enable） 该位指示当接收到调试模式请求时，是否中断正在进行的发送或接收。 0：等待当前启动的发送或接收完成，再进入调试模式 1：中断正在进行的任何发送或接收，并立即进入调试模式	0h
9	ABOEN	R/W	使能 Auto-Bus-On（Auto-Bus-On Enable） 0：禁用 1：使能	0h
13:10	PEN	R/W	使能奇偶校验（Parity Enable） 0101：禁用 其他：使能	0h
14	保留			0h
15	SWRSTEN	RC_W1	使能软件复位（Software Reset Enable） 软件复位步骤：设置 INIT 位以关闭 CAN 通信，然后设置 SWRSTEN 位。置位后，此位将在执行软件复位的一个时钟周期后自动清除。 注意：该位由 INIT 位进行写保护。当使用 SWRSTEN 位进行模块复位，用户配置不会丢失，只有状态位以及需要在下一次 CAN 事务中复位的逻辑会被复位。当使用 SOFTPRESx 寄存器进行模块复位时，包括配置寄存器在内的整个 CAN 模块都将被复位。 0：正常操作 1：强制 CAN 模块执行软件复位	0h
16	DBGFLG	R	调试模式标志（Debug Mode Flag） 此位指示调试范文的内部初始化状态。 0：不处于调试模式，或已请求调试模式但尚未进入调试模式 1：已请求调试模式且 CAN 模块已准备好进行调试访问	0h

位/域	名称	R/W	描述	复位值
17	IEN1	R/W	使能中断线 1（Interrupt line 1 Enable） 使能后，当有中断产生时，CANINT1 将置 1 并保持活动状态，直到处理完中断事件。 0：禁用 CANINT1 1：使能 CANINT1	0h
18	IFDEN1	R/W	使能 IF1 的 DMA 请求线（IF1 DMA request line Enable） 注意：当此位置 1 时，只有当对 IF1 寄存器的首次访问发生时，挂起的 DMA 请求才会被激活。 0：禁用 1：使能	0h
19	IFDEN2	R/W	使能 IF2 的 DMA 请求线（IF2 DMA request line Enable） 注意：当此位置 1 时，只有当对 IF2 寄存器的首次访问发生时，挂起的 DMA 请求才会被激活。 0：禁用 1：使能	0h
20	IFDEN3	R/W	使能 IF3 的 DMA 请求线（IF3 DMA request line Enable） 注意：当此位置 1 时，只有当对 IF2 寄存器的首次访问发生时，挂起的 DMA 请求才会被激活。 0：禁用 1：使能	0h
31:21	保留			0h

40.8.2 错误标志寄存器（CAN_EFLG）

偏移地址：0x04

复位类型：SYSRSn

此寄存器用于指示 CAN 模块的错误类型。当设置 CAN_CTRL[SCHIEN]时，TXCFLG，RXCFLG 和 LEC 位可以产生中断；当设置 CAN_CTRL[ERRIEN]时，PEFLG，BUSOFLG 和 EWFLG 位可以产生中断。EPFLG 位的改变不会产生中断。下一个较低优先级的中断值将替换中断寄存器中的状态中断值（0x8000）。

读取此寄存器后，将自动清除 TXCFLG，RXCFLG 和 PEFLG 位，并将 LEC 复位为 0x07。在调试模式或暂停模式下，将禁用此寄存器的自动清除功能（即读取寄存器时自动清除标志位）。

位/域	名称	R/W	描述	复位值
2:0	LEC	R	最后一个错误类型（Last Error Code） 此位域指示 CAN 总线上最后一个错误的类型。当一个报文在无错误的情况下被传输（接收或发送）时，此位域将被清 0。每当 CPU 读取寄存器时，此位域将被复位为 0x07。 000：无错误 001：填充错误，在接收的报文中检测到超过五个连续相等的位（不允许出现此情况） 010：格式错误，接收帧固定格式部分的格式错误 011：应答错误，CAN 内核发送的报文未被另一个节点响应 100：Bit 1 错误，设备想在报文发送期间（除了仲裁域之外）发送一个隐性电平（逻辑值为“1”的位），但被监测到的总线电平是显性的 101：Bit 0 错误，设备想在发送报文（或应答位，主动错误标志或过载标志）期间发送一个显性电平（逻辑值为“0”），但被监测到的总线电平是隐性的。在总线恢复期间，当检测到 11 个连续的隐性位序列时，设置此状态。因此 CPU 能够监控总线恢复序列的进程（指示总线未卡在显性位或受到持续干扰） 110：CRC 错误：在接收到的报文中，CRC 校验和错误（即接收到的报文的 CRC 与接收数据的计算 CRC 不匹配） 111：从上次 CPU 读取 CAN_EFLG 寄存器依赖，没有 CAN 总线事件被检测到。对 CAN_EFLG 寄存器的任何读取访问都会将此位复位为 111	7h
3	TXCFLG	R	发送完成标志（Transmit completion flag） 此位指示发送状态。此位不会被 CAN 的内部事件复位。 0：从上次 CPU 读取 TXCFLG 以来，没有任何报文被成功传输 1：从上次 CPU 读取访问清除 TXCFLG 位后，已成功发送一条报文（无错误），并至少得到一个节点的应答	0h
4	RXCFLG	R	接收完成标志（Reception completion flag） 此位指示接收状态。此位不会被 CAN 的内部事件复位。 0：从上次 CPU 读取 RXCFLG 以来，没有任何报文被成功接收 1：从上次 CPU 读取访问清除 RXCFLG 位后，已成功接收到报文。而无论接收过滤的结果如何，成功接收到报文后都将设置此位	0h
5	EPFLG	R	被动错误标志（Error Passive Flag） 0：在 CAN 总线出错时，CAN 可以发送主动错误帧 1：CAN 内核处于 CAN 规范中定义的被动错误状态	0h
6	EWFLG	R	错误警告标志（Error Warning Flag） 此位指示错误警告状态，即指示错误计数器是否达到 96 的错误警告限制。 0：两个错误计数器都没有达到错误警告限制 1：至少一个错误计数器达到错误警告限制	0h

位/域	名称	R/W	描述	复位值
7	BUSOFLG	R	Bus-off 标志（Bus-off Flag） 此位指示 CAN 模块是否处于 Bus-off 状态。 0：未处于 Bus-off 状态 1：处于 Bus-off 状态	0h
8	PEFLG	R	奇偶校验错误标志（Parity Error Flag） 此位指示是否检测到奇偶校验错误。 0：从上次 CPU 读取访问以来，未检测到奇偶校验错误 1：在报文 RAM 中，奇偶校验机制检测到奇偶校验错误	0h
31:9	保留			0h

40.8.3 错误计数寄存器（CAN_ECNT）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXECNT	R	发送错误计数器（Transmit Error Counter） 此位域指示发送错误计数器的实际值（0-255）。	0h
14:8	RXECNT	R	接收错误计数器（Receive Error Counter） 此位域指示接收错误计数器的实际值（0-127）。	0h
15	RXEP	R	接收被动错误（Receive Error Passive） CAN 规范中定义了错误被动等级，此位指示接收错误计数器的值是否超过被动错误等级。 0：接收错误计数器的值低于被动错误等级 1：接收错误计数器的值达到被动错误等级	0h
31:16	保留			0h

40.8.4 位时序寄存器（CAN_BTIM）

偏移地址：0x0C

复位类型：SYSRSn

此位域用于配置 CAN 的位时序参数。只有 CAN_CTRL[INIT]和[CFGWACCEN]设置时，才可以对此寄存器进行写入访问。

CAN 位时间可配置值：8-25 个位时间量子。

CAN 位时间量子可配置值：1-1024 个 CAN_CLK 周期。

位/域	名称	R/W	描述	复位值
5:0	BRPSC	R/W	波特率预分频（Baud Rate Prescaler） 此位用于生成位时间量子，CAN_CLK 除以 BRPSC 的值等于位时间量子。位时间是位时间量子的倍数。BRPSC 有效编程值为 0-63。 BRPSC 实际值= BRPSC 编程值+1 注意：此位受 CAN_CTRL[CFGWACCEN]位的写使能保护。	1h

位/域	名称	R/W	描述	复位值
7:6	SJWP	R/W	同步跳变宽度有效编程值（Synchronization Jump Width Valid programmed values） 此位用于配置同步跳变宽度，有效编程值为 0-3。 SJWP 实际值= SJWP 编程值+1 注意：此位受 CAN_CTRL[CFGWACCEN]位的写使能保护。	0h
11:8	TSSPVP1	R/W	采样点前的时间段 1 有效编程值（Time segment 1 before the sample point Valid programmed values） 此位用于配置采样点前的时间段 1 有效编程值，有效编程值为 1-15。 TSSPVP1 实际值= TSSPVP1 编程值+1 注意：此位受 CAN_CTRL[CFGWACCEN]位的写使能保护。	3h
14:12	TSSPVP2	R/W	采样点前的时间段 2 有效编程值（Time segment 2before the sample point Valid programmed values） 此位用于配置采样点前的时间段 2 有效编程值，有效编程值为 0-7。 TSSPVP2 实际值= TSSPVP2 编程值+1 注意：此位受 CAN_CTRL[CFGWACCEN]位的写使能保护。	2h
15	保留			0h
19:16	BRPSCEXT	R/W	扩展波特率预分频（Baud Rate Prescaler Extension） 此位用于配置波特率预分频扩展有效编程值，有效编程值为 0-15。扩展后，波特率预分频值可以达到 1024。 BRPSCEXT 实际值= BRPSCEXT 编程值+1 注意：此位受 CAN_CTRL[CFGWACCEN]位的写使能保护。	0h
31:17	保留			0h

40.8.5 中断寄存器（CAN_INT）

偏移地址：0x10

复位类型：SYSRSn

此寄存器用于识别中断源。

位/域	名称	R/W	描述	复位值
15:0	INT0	R	中断 0（Interrupt 0） 此位域用于指示挂起的中断类型。当有多个中断挂起时，CAN_INT 寄存器将指向优先级最高的中断挂起。 注意：状态中断的优先级最高，报文中断中，邮箱中断的优先级随着邮箱编号的增加而递减。CANINT0 中断线将保持活动状态直到 CAN_INT[INT0]=0 或 CAN_CTRL[IEN0]被清除。 0x0000：无挂起的中断 0x0001-0x0020：产生中断的邮箱编号 0x8000：CAN_EFLG 寄存器的值不为 0x07 其他：保留	0h

位/域	名称	R/W	描述	复位值
23:16	INT1	R	中断 1（Interrupt 1） 此位域用于指示挂起的中断类型。当有多个中断挂起时，CAN_INT 寄存器将指向优先级最高的中断挂起。 注意：报文中断中，邮箱中断的优先级随着邮箱编号的增加而递减。通过清除 CAN_IFxMCTRL[IPEND]位以清除报文中断。CANINT1 中断线将保持活动状态直到 CAN_INT[INT1]=0 或 CAN_CTRL[IEN1]被清除。 0x00：无挂起的中断 0x01-0x20：产生中断的邮箱编号 其他：保留	0h
31:24	保留			0h

40.8.6 测试寄存器（CAN_TEST）

偏移地址：0x14

复位类型：SYSRSn

如果要对 CAN 进行测试，可以配置此寄存器。测试开始前，需要使能 CAN_CTRL[TESTEN]位，才能访问此寄存器（除 RXMON 外，此位只读，用于监控 CANRX 引脚）。注意：

（1） 激活内部环回模式时（即 LBEN=1），将忽略 EXTLBEN 位的设置；

（2） 将 TXCTRL 配置为 00 以外的值时会干扰报文传输。

位/域	名称	R/W	描述	复位值
2:0	保留			0h
3	SLTEN	R/W	使能静默模式（Silent Mode Enable） 0：禁用 1：使能	0h
4	LBEN	R/W	使能环回模式（ Loop Back Mode Enable） 0：禁用 1：使能	0h
6:5	TXCTRL	R/W	控制发送引脚（TX pin Control） 此位用于控制发送引脚的操作。 00：正常操作，CAN 内核控制发送引脚 CANTX 01：可以在 CANTX 引脚上检测到采样点 10：CANTX 引脚驱动为显性电平（逻辑“0”） 11：CANTX 引脚驱动为隐性电平（逻辑“1”）	0h
7	RXMON	R	监控接收引脚（RX pin Monitors） 此位用于监控接收引脚的实际值。 0：CAN 总线为显性电平（逻辑“0”） 1：CAN 总线为隐性电平（逻辑“1”）	0h
8	EXTLBEN	R/W	使能外部回环模式（External Loop Back Mode Enable） 0：禁用 1：使能	0h
9	RDAEN	R/W	使能 RAM 直接访问（RAM Direct Access Enable） 0：正常操作 1：使能对 RAM 的直接访问	0h
31:10	保留			0h

40.8.7 奇偶校验错误寄存器（CAN_PE）

偏移地址：0x1C

复位类型：SYSRSn

此寄存器指示检测到奇偶校验错误的字编号/邮箱编号。

若产生奇偶校验错误，则 CAN_EFLG[PEFLG]置位。CAN_EFLG[PEFLG]不会被奇偶校验机制复位，必须通过读取 CAN_EFLG 寄存器的方式复位。

此外，此寄存器指示产生奇偶校验错误的内存区域。当多个有奇偶校验错误的字被检测到时，此寄存器中只显示编号值最大的字。一旦检测到奇偶校验错误，直到系统断电前，最后一个错误信息将始终保留在此寄存器中。

位/域	名称	R/W	描述	复位值
7:0	MAILNUM	R	校验错误邮箱号（parity error Mailbox number ） 0x01-0x21：邮箱产生奇偶校验错误，位域值表示产生奇偶校验错误的邮箱编号。	0h
10:8	WORDNUM	R	校验错误字号（parity error Word number ） 0x01-0x21：字产生奇偶校验错误，位域值表示产生奇偶校验错误的字编号。根据报文 RAM 在 RAM 直接访问（RDA）模式中的表示，邮箱的 RDA 字的编号为 1-5。	1h
31:11	保留			0h

40.8.8 RAM 初始化寄存器（CAN_RAMINIT）

偏移地址：0x40

复位类型：SYSRSn

此寄存器用于初始化邮箱 RAM，初始化过程中，包括 MVAL 位在内的整个邮箱 RAM 都被清除。

位/域	名称	R/W	描述	复位值
0	KEY0	R/W	关键字 0（Key 0） 只有当 KEY3- KEY0 设置为 1010 时，对此寄存器的写入访问有效。 注意：当 CAN RAM 初始化完成后，KEY3- KEY0 将恢复到复位值。	0h
1	KEY1	R/W	关键字 1（Key 1） 详情请参见 KEY0。	0h
2	KEY2	R/W	关键字 2（Key 2） 详情请参见 KEY0。	1h
3	KEY3	R/W	关键字 3（Key 3） 详情请参见 KEY0。	0h
4	RAMINIT	R/W	初始化 CAN 邮箱 RAM 初始化（Initiate CAN Mailbox RAM initialization） 此位用于读取 CAN 邮箱 RAM 初始化状态并触发初始化。 读操作： 0：已完成初始化或未启动初始化 1：正在进行初始化 写操作： 0：无任何操作 1：启动 CAN 邮箱 RAM 的初始化。在初始化过程完成后，RAMINIT 位会被自动清 0	0h

位/域	名称	R/W	描述	复位值
5	RAMINITCFLG	R	CAN 邮箱 RAM 初始化完成标志（CAN Mailbox RAM initialization completion flag） 0：正在进行初始化或未启动初始化 1：已完成初始化	0h
31:6	保留			0h

40.8.9 全局中断使能寄存器（CAN_GLBINTEN）

偏移地址：0x50

复位类型：SYSRSn

此寄存器用于使能 CANINT0 和 CANINT1 全局中断。

位/域	名称	R/W	描述	复位值
0	GLBINT0EN	R/W	使能 CANINT0 全局中断（CANINT0 Global Interrupt Enable） 0：CANINT0 不会向 NVIC 的发出中断请求 1：当检测到中断条件时，CANINT0 向 NVIC 发出中断请求	0h
1	GLBINT1EN	R/W	使能 CANINT1 全局中断（CANINT1 Global Interrupt Enable） 0：CANINT1 不会向 NVIC 的发出中断请求 1：当检测到中断条件时，CANINT1 向 NVIC 发出中断请求	0h
31:2	保留			0h

40.8.10 全局中断标志寄存器（CAN_GLBINTFLG）

偏移地址：0x54

复位类型：SYSRSn

此寄存器用于指示 CANINT0 和 CANINT1 的中断状态。

位/域	名称	R/W	描述	复位值
0	GLBINT0FLG	R	CANINT0 全局中断标志（CANINT0 Global Interrupt Flag） 0：无 CANINT0 中断 1：产生 CANINT0 中断	0h
1	GLBINT1FLG	R	CANINT1 全局中断标志（CANINT1 Global Interrupt Flag） 0：无 CANINT1 中断 1：产生 CANINT1 中断	0h
31:2	保留			0h

40.8.11 全局中断清除寄存器（CAN_GLBINTCLR）

偏移地址：0x58

复位类型：SYSRSn

此寄存器用于清除 CANINT0 和 CANINT1 的在 CAN_GLBINTFLG 寄存器中的全局中断标志。

位/域	名称	R/W	描述	复位值
0	GLBINT0CLR	W	清除 CANINT0 全局中断标志（CANINT0 Global Interrupt Flag Clear） 0：无作用 1：清除 CAN_GLBINTFLG[GLBINT0FLG]，并允许 NVIC 从 CANINT0 中接收另一个中断	0h
1	GLBINT1CLR	W	清除 CANINT1 全局中断标志（CANINT1 Global Interrupt Flag Clear） 0：无作用 1：清除 CAN_GLBINTFLG[GLBINT1FLG]，并允许 NVIC 从 CANINT1 中接收另一个中断	0h
31:2	保留			0h

40.8.12 Auto-Bus-On 定时寄存器（CAN_ABOTMR）

偏移地址：0x80

复位类型：SYSRSn

此寄存器用于在 Bus-off 恢复序列启动前引入延迟，延迟值为变量。

位/域	名称	R/W	描述	复位值
31:0	ABOTMR	R/W	Auto-Bus-On 定时器（Auto-Bus-On Timer） 通过清除 INIT 位以启动 Bus-off 恢复序列，此位域用于在此序列前增加时钟周期数。时钟指 CAN 模块的输入时钟。 通过置位 CAN_CTRL[ABOEN]以使能 Auto-Bus-On 定时器功能，Auto-Bus-On 定时器由 32 位计数器实现。当 CAN 进入 Bus-off 状态时，计数器开始从预装载值倒计时至 0。倒计时结束后，计数器将重新加载 CAN_ABOTMR 的预装载值。 注意： （1）如果 CAN 模块处于调试模式，ABO 定时器将暂停； （2）如果在 ABO 定时器运行时对 CAN_CTRL 寄存器进行写操作，ABO 程序将被中止。	0h

40.8.13 发送请求标志寄存器（CAN_TXREQFLG）

偏移地址：0x84

复位类型：SYSRSn

此寄存器用于指示 CAN_TXREQ 寄存器中的一个或多个位是否置位，此寄存器中一个 Bit 表示 CAN_TXREQ 寄存器中的一个字节（即 8 个邮箱一组），当 TXREQ 至少一个位被置位时，CAN_TXREQFLG 寄存器中的对应位也置位。

位/域	名称	R/W	描述	复位值
1:0	TXREQ1FLG	R	发送请求寄存器 1 标志（Transmit Request Register 1 flag） [0]：表示 CAN_TXREQ 寄存器中的第一个字节（byte 0），当 CAN_TXREQ 寄存器中的 byte 0 中至少有一个位被置位时，此位也置位 [1]：表示 CAN_TXREQ 寄存器中的第二个字节（byte 1），当 CAN_TXREQ 寄存器中的 byte 1 中至少有一个位被置位时，此位也置位	0h

位/域	名称	R/W	描述	复位值
3:2	TXREQ2FLG	R	发送请求寄存器 2 标志（Transmit Request Register 2 flag） [2]：表示 CAN_TXREQ 寄存器中的第三个字节（byte 2），当 CAN_TXREQ 寄存器中的 byte 2 中至少有一个位被置位时，此位也置位 [3]：表示 CAN_TXREQ 寄存器中的第四个字节（byte 3），当 CAN_TXREQ 寄存器中的 byte 3 中至少有一个位被置位时，此位也置位	0h
31:4	保留			0h

40.8.14 发送请求寄存器（CAN_TXREQ）

偏移地址：0x88

复位类型：SYSRSn

此寄存器用于指示邮箱是否有待处理的发送请求。

设置/复位特定邮箱的 TXREQ 位方式：

（1） CPU 通过使用 IF1/2 报文接口寄存器来设置或复位

（2） 在成功传输后，使用报文处理器进行设置或复位

（3） 在接收到远程帧后，使用报文处理器进行设置或复位

位/域	名称	R/W	描述	复位值
31:0	TXREQ	R	发送请求（Transmit Request） Bit 0 对应邮箱 1 Bit 1 对应邮箱 2 … Bit 31 对应邮箱 32 0：此邮箱没有发送请求 1：此邮箱有待处理的发送请求并且未完成发送	0h

40.8.15 新数据标志寄存器（CAN_NDATAFLG）

偏移地址：0x98

复位类型：SYSRSn

此寄存器用于指示 CAN_NDATA 寄存器中的一个或多个位是否置位，此寄存器中一个 Bit 表示 CAN_NDATA 寄存器中的一个字节（即 8 个邮箱一组），当 NDATA 中至少一个位被置位时，CAN_ NDATAFLG 寄存器中的对应位也置位。

位/域	名称	R/W	描述	复位值
1:0	NDATA1FLG	R	新数据寄存器 1 标志（New Data Register 1 flag） [0]：表示 CAN_DATA 寄存器中的第一个字节（byte 0），当 CAN_DATA 寄存器中的 byte 0 中至少有一个位被置位时，此位也置位 [1]：表示 CAN_DATA 寄存器中的第二个字节（byte 1），当 CAN_DATA 寄存器中的 byte 1 中至少有一个位被置位时，此位也置位	0h

位/域	名称	R/W	描述	复位值
3:2	NDATA2FLG	R	新数据寄存器 2 标志（New Data Register 2 flag） [2]：表示 CAN_DATA 寄存器中的第三个字节（byte 2），当 CAN_DATA 寄存器中的 byte 2 中至少有一个位被置位时，此位也置位 [3]：表示 CAN_DATA 寄存器中的第四个字节（byte 3），当 CAN_DATA 寄存器中的 byte 3 中至少有一个位被置位时，此位也置位	0h
31:4	保留			0h

40.8.16 新数据寄存器（CAN_NDATA）

偏移地址：0x9C

复位类型：SYSRSn

此寄存器用于指示邮箱的哪些数据字节已更新。

设置/复位特定邮箱的 NDATA 位方式：

（1） CPU 通过使用 IFx 报文接口寄存器来设置或复位

（2） 在成功传输后，使用报文处理器进行设置或复位

（3） 在接收到远程帧后，使用报文处理器进行设置或复位

位/域	名称	R/W	描述	复位值
31:0	NDATA	R	新数据（New Data） Bit 0 对应邮箱 1 Bit 1 对应邮箱 2 … Bit 31 对应邮箱 32 0：数据未更新（上次清除此标志位后，没有新数据被报文处理器写入到此邮箱的数据部分） 1：此邮箱的数据部分有新数据（由 CPU 或报文处理器）写入	0h

40.8.17 中断挂起标志寄存器（CAN_IPENDFLG）

偏移地址：0xAC

复位类型：SYSRSn

此寄存器用于指示 CAN_IPEND 寄存器中的一个或多个位是否置位，此寄存器中一个 Bit 表示 CAN_IPEND 寄存器中的一个字节（即 8 个邮箱一组），当 NDATA 中至少一个位被置位时，CAN_IPENDFLG 寄存器中的对应位也置位。

位/域	名称	R/W	描述	复位值
1:0	IPEND1FLG	R	中断挂起寄存器 1 标志（Interrupt Pending Register 1 flag） [0]：表示 CAN_IPEND 寄存器中的第一个字节（byte 0），当 CAN_IPEND 寄存器中的 byte 0 中至少有一个位被置位时，此位也置位 [1]：表示 CAN_IPEND 寄存器中的第二个字节（byte 1），当 CAN_IPEND 寄存器中的 byte 1 中至少有一个位被置位时，此位也置位	0h

位/域	名称	R/W	描述	复位值
3:2	IPEND2FLG	R	中断挂起寄存器 2 标志（Interrupt Pending Register 2 flag） [2]：表示 CAN_IPEND 寄存器中的第三个字节（byte 2），当 CAN_IPEND 寄存器中的 byte 2 中至少有一个位被置位时，此位也置位 [3]：表示 CAN_IPEND 寄存器中的第四个字节（byte 3），当 CAN_IPEND 寄存器中的 byte 3 中至少有一个位被置位时，此位也置位	0h
31:4	保留			0h

40.8.18 中断挂起寄存器（CAN_IPEND）

偏移地址：0xB0

复位类型：SYSRSn

此寄存器用于指示哪些邮箱的中断挂起。

设置/复位特定邮箱的 IPEND 位方式：

（1） CPU 通过使用 IFx 报文接口寄存器来设置或复位

（2） 在成功传输后，使用报文处理器进行设置或复位

（3） 在接收到远程帧后，使用报文处理器进行设置或复位

位/域	名称	R/W	描述	复位值
31:0	IPEND	R/W	中断挂起（Interrupt Pending） Bit 0 对应邮箱 1 Bit 1 对应邮箱 2 … Bit 31 对应邮箱 32 0：此邮箱中没有挂起的中断 1：此邮箱有挂起的中断	0h

40.8.19 邮箱有效标志寄存器（CAN_MVALFLG）

偏移地址：0xC0

复位类型：SYSRSn

此寄存器用于指示 CAN_MVAL 寄存器中的一个或多个位是否置位，此寄存器中一个 Bit 表示 CAN_MVAL 寄存器中的一个字节（即 8 个邮箱一组），当 MVAL 中至少一个位被置位时，CAN_MVALFLG 寄存器中的对应位也置位。

位/域	名称	R/W	描述	复位值
1:0	MVAL1FLG	R	消息有效寄存器 1 标志（Message Valid Register 1 flag） [0]：表示 CAN_MVAL 寄存器中的第一个字节（byte 0），当 CAN_MVAL 寄存器中的 byte 0 中至少有一个位被置位时，此位也置位 [1]：表示 CAN_MVAL 寄存器中的第二个字节（byte 1），当 CAN_MVAL 寄存器中的 byte 1 中至少有一个位被置位时，此位也置位	0h

位/域	名称	R/W	描述	复位值
3:2	MVAL2FLG	R	消息有效寄存器 2 标志（Message Valid Register 2 flag） [2]：表示 CAN_MVAL 寄存器中的第三个字节（byte 2），当 CAN_MVAL 寄存器中的 byte 2 中至少有一个位被置位时，此位也置位 [3]：表示 CAN_MVAL 寄存器中的第四个字节（byte 3），当 CAN_MVAL 寄存器中的 byte 3 中至少有一个位被置位时，此位也置位	0h
31:4	保留			0h

40.8.20 邮箱有效寄存器（CAN_MVAL）

偏移地址：0xC4

复位类型：SYSRSn

此寄存器用于指示邮箱是否有效。

CPU 通过使用 IF1/2 报文接口寄存器来设置或复位特定邮箱的 MVAL 位。

位/域	名称	R/W	描述	复位值
31:0	MVAL	R	邮箱有效（Mailbox Valid） Bit 0 对应邮箱 1 Bit 1 对应邮箱 2 … Bit 31 对应邮箱 32 0：此邮箱被报文处理器忽略 1：此邮箱已配置并由报文处理器处理	0h

40.8.21 中断邮箱寄存器（CAN_INTMAIL）

偏移地址：0xD8

复位类型：SYSRSn

此寄存器用于确定每个邮箱置位 CAN_IPEND[IPEND]时断言的中断线（CANINT0 或 CANINT1）。当置位 CAN_CTRL[IEN0]时，全局使能 CANINT0，当清除 CAN_CTRL[IEN0]=0 时，全局禁用 CANINT0，CANINT1 同理。

CAN_INT 寄存器中的中断。当中断线 CANINT0 或 CANINT1 被触发时，CAN_INT[IINT0]或 CAN_INT[IINT1]也会被设置为与邮箱编号对应的值。

位/域	名称	R/W	描述	复位值
31:0	INTMAIL	R/W	中断邮箱（Interrupt mailbox） Bit 0 对应邮箱 1 Bit 1 对应邮箱 2 … Bit 31 对应邮箱 32 0：当对应的 CAN_IPENDFLG[IPENDFLG]=1 时，断言 CANINT0 中断线 1：当对应的 CAN_IPENDFLG[IPENDFLG]=1 时，断言 CANINT1 中断线	0h

40.8.22 IFx 命令寄存器（CAN_IFxCOM）（x=1…2）

偏移地址：0x100+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于配置并启动报文 RAM 与 IF1/IF2 寄存器组之间的数据传输，可以对需要被传输的邮箱进行配置。

启动传输：当 CPU 向 CAN_IFxCOM[VALMNUM]写入邮箱编号时，会启动一个传输。这个写操作会自动置位 CAN_IFxCOM[BUSYFLG]，以表示传输正在进行中。

传输完成：启动传输后，报文 RAM 与接口寄存器之间的传输将在 4-14 个时钟周期后完成，并清除 CAN_IFxCOM[BUSYFLG]位。当报文传输与 CAN 报文的发送，接收过滤或报文存储重合时，所需的时钟周期数最大。

连续写入：当向 CAN_IFxCOM 寄存器连续写入邮箱编号时（即在第一次传输仍在进行时请求第二次传输），那么第二次传输将在第一次完成后开始

注意：

（1） 应以 32 位的形式写入此寄存器或先写入高 16 位再写入低 16 位。

（2） 禁止向整个寄存器写入零。

（3） 调试支持：在调试模式或暂停模式下，将禁用此寄存器的自动清除功能（即读取/写入寄存器时自动清除 REQDMAIFx 位）。

（4） 写保护：当置位 CAN_IFxCOM[BUSYFLG]时，不能向 IF1/IF2 寄存器组写入数据（写保护）。

（5） 无效邮箱编号：如果向 CAN_IFxCOM[VALMNUM]写入无效的邮箱编号，报文处理器可能会访问已实现（有效的）邮箱。

位/域	名称	R/W	描述	复位值
7:0	VALMNUM	R/W	有效的邮箱编号（Valid Mailbox Number） 此位域指示报文 RAM 中用于数据传输的邮箱编号。 此位受 BUSYFLG 位写保护。 0x01-0x20：进行数据传输的有效邮箱编号 其他：无效的邮箱编号	1h
13:8	保留			0h
14	REQDMAIFx	R/W	IFx 更新后请求 DMA（Request DMA after IFx update） 此位域指示 IFx 更新后 DMA 请求的触发状态。当此位置 1 时，写入 VALMNUM 位不会取消 DMA 请求的活跃状态。 此位受 BUSYFLG 位写保护。 自动复位特性：此位具有自动复位功能。即每次 DMA 请求完成后，该位将被自动复位为 0，因此需要在每个 DMA 周期中单独设置此位。 0：没有激活的 IFx DMA 请求 1：当报文 RAM 与 IFx 之间的传输完成后，激活 DMA 请求。一旦 DMA 请求被激活，它将保持活跃状态，直到对 IFx 寄存器的第一次读写操作	0h

位/域	名称	R/W	描述	复位值
15	BUSYFLG	R	忙碌标志（Busy Flag） 当 CPU 向 VALMNUM 写入邮箱编号后，会自动设置此位，并且对 IFx 寄存器组进行写保护，在传输完成后自动清除此位。 0：报文 RAM 与 IFx 寄存器组之间没有正在进行的数据传输 1：报文 RAM 与 IFx 寄存器组之间正在进行数据传输	0h
16	DATABYTEB	R/W	访问数据字节 4-7（Access Data Bytes 4-7） 此位受 BUSYFLG 位写保护。 注意：报文传输的持续时间与传输的字节数量无关。 0：无操作 1：读操作：数据字节 4-7 将从写入 VALMNUM 位的邮箱编号寻址的邮箱传输到相应的 IFx 寄存器组；写操作：数据字节 4-7 将从 IFx 寄存器组传输到写入 VALMNUM 位的邮箱编号寻址的邮箱	0h
17	DATABYTEA	R/W	访问数据字节 0-3（Access Data Bytes 0-3） 此位受 BUSYFLG 位写保护。 注意：报文传输的持续时间与传输的字节数量无关。 0：无操作 1：读操作：数据字节 0-3 将从写入 VALMNUM 位的邮箱编号寻址的邮箱传输到相应的 IFx 寄存器组；写操作：数据字节 0-3 将从 IFx 寄存器组传输到写入 VALMNUM 位的邮箱编号寻址的邮箱	0h
18	TXREQCFG	R/W	配置发送请求/新数据 （Transmit Request/ New Data Configure） 此位受 BUSYFLG 位写保护。 注意：读取邮箱时，可以与控制位 CAN_IPEND[IPEND]和 CAN_NDATA[NDATA]的复位结合。传输到 IFx 报文控制寄存器的这些位的值始终反映复位前的状态。当通过设置此寄存器中的 TXREQCFG 来请求 CAN 传输时，邮箱中的 CAN_TXREQ[TXREQ]/CAN_NDATA[NDATA]位将被置 1，不管 IFx 报文控制寄存器中的值如何。 读操作： 0：无操作 1：清除邮箱中的 CAN_NDATA[NDATA]位 写操作： 0：根据 ACCCTRL 位的配置处理 CAN_TXREQ[TXREQ]/CAN_NDATA[NDATA]位 1：设置邮箱中的 CAN_TXREQ[TXREQ]/CAN_NDATA[NDATA]位	0h
19	IPENDCLR	R/W	清除中断挂起（Interrupt Pending Clear） 此位受 BUSYFLG 位写保护。 0：无操作 1：读操作：清除邮箱中的 CAN_IPEND[IPEND]位；写入：忽略此位	0h

位/域	名称	R/W	描述	复位值
20	ACCCTRL	R/W	访问控制（Access control） 当置位 TXREQCFG 时，将忽略 CAN_IFxMCTRL[TXREQEN]和 CAN_IFxMCTRL[NDATA]位 此位受 BUSYFLG 位写保护。 0：无操作 1：读操作：CAN_IFxMCTRL 中的位将从写入 VALMNUM 位的邮箱编号寻址的邮箱传输到相应的 IFx 寄存器组；写操作：CAN_IFxMCTRL 中的位将从 IFx 寄存器组传输到写入 VALMNUM 位的邮箱编号寻址的邮箱	0h
21	ACCARB	R/W	访问仲裁（Access Arbitration） 此位受 BUSYFLG 位写保护。 0：无操作 1：读操作：CAN_IFxARB 中的位将从写入 VALMNUM 位的邮箱编号寻址的邮箱传输到相应的 IFx 寄存器组；写操作：CAN_IFxARB 中的位将从 IFx 寄存器组传输到写入 VALMNUM 位的邮箱编号寻址的邮箱	0h
22	ACCMASK	R/W	访问屏蔽（Access Mask） 此位受 BUSYFLG 位写保护。 0：无操作 1：读操作：CAN_IFxMASK 中的位将从写入 VALMNUM 位的邮箱编号寻址的邮箱传输到相应的 IFx 寄存器组；写操作：CAN_IFxMASK 中的位将从 IFx 寄存器组传输到写入 VALMNUM 位的邮箱编号寻址的邮箱	0h
23	RWCFG	R/W	配置读写功能（Read/Write function Configure） 此位指示数据的传输方向，CAN_IFxCOM 寄存器其他位的功能与数据传输方向有关。 此位受 BUSYFLG 位写保护。 0：进行读操作，即数据从邮箱传输到选定的 IFx 报文缓冲寄存器中 1：进行写操作，即数据从 IFx 报文缓冲寄存器传输到写入 VALMNUM 位的邮箱编号寻址的邮箱中	0h
31:24	保留			0h

40.8.23 IFx 掩码寄存器（CAN_IFxMASK）（x=1…2）

偏移地址：0x104+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于配置邮箱的掩码。

注意：当 CAN_IFxCOM[BUSYFLG]=1（x=1…2）时，不可以对 IFx（x=1…2）寄存器组进行写入访问。

位/域	名称	R/W	描述	复位值
28:0	IDMASK	R/W	标识符掩码（Identifier Mask） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：邮箱标识符中的对应位不用于接收过滤（即可以为任意值） 1：邮箱标识符中的对应位用于接收过滤	1FFFFFFFh
29	保留			1h

位/域	名称	R/W	描述	复位值
30	DIRMASK	R/W	报文方向掩码（Message Direction Mask） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：CAN_IFxARB[MDIRCFG]位不用于接收过滤 1：CAN_IFxARB[MDIRCFG]位用于接收过滤	1h
31	EXTMASK	R/W	扩展标识符掩码（Extended Identifier Mask） 当邮箱使用 11 位的标准格式的标识符时，接收到的数据帧的 11 位标识符被放置在 MIDCFG[28:18]。在接收过滤的过程中，仅考虑 MIDCFG[28:18]和对应的 IDMASK[28:18]位。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：CAN_IFxARB[EXTIDCFG]位不用于接收过滤 1：CAN_IFxARB[EXTIDCFG]位用于接收过滤	1h

40.8.24 IFx 仲裁寄存器（CAN_IFxARB）（x=1…2）

偏移地址：0x108+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于配置邮箱的仲裁信息。MIDCFG，MDIRCFG 和 EXTIDCFG 位定义要发送的报文的类型和标识符，并与 CAN_IFxMASK 寄存器中的对应位一起用于传入报文的接收过滤。

当接收到报文存储在具有匹配标识符的有效邮箱中，如果传输方向为接收，则接收到的是数据帧（Data Frame）；如果传输方向为发送，则收到的是远程帧（Remote Frame）。

当接收到的报文（数据帧或远程帧）同时匹配多个有效邮箱时，报文将被存储到编号最低的邮箱中。

注意：当 CAN_IFxCOM[BUSYFLG]=1（x=1...2）时，不可以对 IFx（x=1...2）寄存器组进行写入访问。

位/域	名称	R/W	描述	复位值
28:0	MIDCFG	R/W	配置报文标识符（Message Identifier Configure） 11 位标识符-标准帧：MIDCFG[28:18] 29 位标识符-扩展帧：MIDCFG[28:0] 此位受 CAN_IFxCOM[BUSYFLG]位写保护。	0h
29	MDIRCFG	R/W	配置报文方向（Message Direction Configure） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：接收，当 TXREQEN=1 时，发送远程帧，远程帧携带具有与此邮箱相匹配的标识符；当接收到数据帧，且此数据帧的标识符与邮箱的标识符匹配时，该数据帧将被存储到邮箱中 1：发送，当 TXREQEN=1 时，发送数据帧；当接收到远程帧，且此远程帧的标识符与邮箱的标识符匹配时，如果 REMEN=1，设置此邮箱的 TXREQEN=1	0h
30	EXTIDCFG	R/W	配置扩展标识符（Extended Identifier Configure） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：邮箱使用 11 位标准标识符 1：邮箱使用 29 位扩展标识符	0h

位/域	名称	R/W	描述	复位值
31	MAILEN	R/W	使能邮箱（mailbox enable） 禁用邮箱时，报文处理器忽略此邮箱，它不会被报文处理器考虑用于接收或发送 CAN 报文；使能邮箱时，它会被报文处理器用于接收或发送操作。 在 CAN 初始化期间，CPU 应该将所有未使用邮箱的 CAN_MVAL[MVAL]位复位为 0。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：禁用 1：使能	0h

40.8.25 IFx 报文控制寄存器（CAN_IFxMCTRL）（x=1…2）

偏移地址：0x10C+(x-1)*0x40

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	DLCCFG	R/W	配置数据长度码（Data Length Code Configure） 注意：具有相同标识符的不同节点的邮箱需要设置相同的数据长度码。当报文处理器接收到一个数据帧时，它将根据接收到的报文设置相应邮箱中的 DLC。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0000-1000：数据帧具有 0-8 个数据字节 1001-1111：数据帧具有 8 个数据字节	0h
6:4	保留			0h
7	EOBEN	R/W	使能块结束（End of Block Enable） 此位域用于将多个邮箱级联起来构建一个 FIFO 缓冲区。当作为独立邮箱时，此位必须始终置 1。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：当前的邮箱属于 FIFO 缓冲区块的一部分，并且不是这个 FIFO 缓冲区块的最后一个邮箱 1：当前的邮箱是一个独立的邮箱，不属于任何 FIFO 缓冲区块或当前的邮箱是 FIFO 缓冲区块中的最后一个邮箱	0h
8	TXREQEN	R/W	使能发送请求（Transmit Request Enable） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：此邮箱没有待处理的发送请求 1：此邮箱有待处理的发送请求并且未完成发送	0h
9	REMEN	R/W	使能远程（Remote Enable） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：接收到远程帧后，不更改 TXREQEN 1：接收到远程帧后，置位 TXREQEN	0h
10	RXIEN	R/W	使能接收中断（Receive Interrupt Enable） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：成功接收到帧数据后，不触发 IPEND 1：成功接收到帧数据后，触发 IPEND	0h
11	TXIEN	R/W	使能发送中断（Transmit Interrupt Enable） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：成功发送帧数据后，不触发 IPEND 1：成功发送帧数据后，触发 IPEND	0h

位/域	名称	R/W	描述	复位值
12	MASKEN	R/W	使能掩码（Mask Enable） 当此位置 1 时，必须在邮箱初始化期间配置邮箱的 MASK 位段（IDMASK，EXTMASK，DIRMASK），再置位 CAN_MVAL[MVAL]。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：忽略掩码 1：使用掩码（即 CAN_IFxMASK 寄存器中的位域）进行接收过滤	0h
13	IPEND	R/W	中断挂起（Interrupt Pending） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：此邮箱中没有挂起的中断 1：此邮箱有挂起的中断。当没有此邮箱中断源的优先级最高时，中断寄存器的中断标识符将指向此邮箱	0h
14	MLOST	R/W	报文丢失（Message Lost ） 此位域仅适用于传输方向设置为接收的邮箱。 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：CPU 上次复位此位后，没有报文丢失 1：发生了报文丢失，即报文处理器在 CAN_NDATA[NDATA]置位时接收到了新报文，并且将其存储在了此邮箱中，导致之前的报文被新报文覆盖。	0h
15	NDATA	R/W	新数据（New Data） 此位受 CAN_IFxCOM[BUSYFLG]位写保护。 0：CPU 上次清除此位后，此邮箱的数据部分没有新数据写入 1：有新数据写入，即此邮箱的数据部分有新数据写入（由消息处理器或 CPU 写入）	0h
31:16	保留			0h

40.8.26 IFx 数据 A 寄存器（CAN_IFxDATAA）（x=1…2）

偏移地址：0x110+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于提供访问 CAN 报文数据字节的窗口。数据字节被按照 CAN 数据帧的发送和接收顺序排列。即 DATD0 是第一个字节，DATD7 是最后一个字节。在 CAN 的串行比特流中，最先发送每个字节的最高有效位（MSB）。

注意：当 CAN_IFxCOM[BUSYFLG]=1（x=1...2）时，不可以对此寄存器中的所有位进行写入访问。

位/域	名称	R/W	描述	复位值
7:0	DATD0	R/W	数据字节 0（Data Byte 0）	0h
15:8	DATD1	R/W	数据字节 1（Data Byte 1）	0h
23:16	DATD2	R/W	数据字节 2（Data Byte 2）	0h
31:24	DATD3	R/W	数据字节 3（Data Byte 3）	0h

40.8.27 IFx 数据 B 寄存器（CAN_IFxDATAB）（x=1…2）

偏移地址：0x114+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于提供访问 CAN 报文数据字节的窗口。数据字节被按照 CAN 数据帧的发送和接收顺序排列。即 DATD0 是第一个字节，DATD7 是最后一个字节。在 CAN 的串行比特流中，最先发送每个字节的最高有效位（MSB）。

注意：当 CAN_IFxCOM[BUSYFLG]=1（x=1...2）时，不可以对此寄存器中的所有位进行写入访问。

位/域	名称	R/W	描述	复位值
7:0	DATD4	R/W	数据字节 4（Data Byte 4）	0h
15:8	DATD5	R/W	数据字节 5（Data Byte 5）	0h
23:16	DATD6	R/W	数据字节 6（Data Byte 6）	0h
31:24	DATD7	R/W	数据字节 7（Data Byte 7）	0h

40.8.28 IF3 观察寄存器（CAN_IF3OBS）

偏移地址：0x140

复位类型：SYSRSn

IF3 寄存器组自动更新新接收到的邮箱，而无需 CPU 初始化从报文 RAM 到 IF3 寄存器组的传输。

此寄存器的 Bit[4:0]为观察标志位，用于确定需要读取 IF3 寄存器组的哪些数据部分，以完成一个 DMA 读周期。当所有被标记的数据部分被读取后，使能 CAN 并使用新数据更新 IF3 寄存器组。Bit[12:8]用于观察当前读周期的状态。应用程序可以利用 Bit[12:8]和 UDDATA 在轮询模式或中断模式下获得有关新 IF3 内容的通知。

对 IF3 寄存器的写入访问将中断一个正在进行的 DMA 周期，通过复位 DMA 请求线，并使得 IF3 接口寄存器组可以用新数据进行更新。为了保证数据一致，在重新配置 CAN_IFxOBS 寄存器之前禁用 DMA。当使用 IF3 更新使能机制且没有置位观察标志时，相应的邮箱将被复制到 IF3，且无需激活 DMA 请求线，也无需等待 DMA 读取访问。

任意顺序的单字节或半字都可以进行访问。在使用字节或半字访问时，如果读取了一个数据部分的所有字节，该数据部分被标记为已完成。

位/域	名称	R/W	描述	复位值
0	MASKRD	R/W	读取掩码数据（Mask data read） 0：不读取掩码数据 1：读取掩码数据，以进行下一次 IF3 更新	0h
1	ARBRD	R/W	读取仲裁数据（Arbitration data read） 0：不读取仲裁数据 1：读取仲裁数据，以进行下一次 IF3 更新	0h
2	CTRLRD	R/W	读取控制位（Control bits read） 0：不读取控制位数据 1：读取控制位数据，以进行下一次 IF3 更新	0h
3	DATAARD	R/W	读取数据 A（Data A read） 0：不读取数据 A 1：读取数据 A，以进行下一次 IF3 更新	0h

位/域	名称	R/W	描述	复位值
4	DATABRD	R/W	读取数据 B（Data B read） 0：不读取数据 B 1：读取数据 B，以进行下一次 IF3 更新	0h
7:5	保留			0h
8	MASKRDACC	R	掩码数据读访问状态（Mask data read access status） 0：已读取所有掩码数据字节，或掩码数据字节未标记为待读取 1：掩码数据字节中有待读取的数据	0h
9	ARBRDACC	R	仲裁数据读访问状态（Arbitration data read access status） 0：已读取掩码的所有数据字节，或掩码的数据字节未标记为待读取 1：掩码的数据字节中有待读取的数据	0h
10	CTRLRDACC	R	控制位数据读访问状态（Control bits read access status） 0：已读取控制位所有数据字节，或控制位的数据字节未标记为待读取 1：控制位的数据字节中有待读取的数据	0h
11	DATAARDACC	R	数据 A 读访问状态（Data A read access status） 0：已读取数据 A 的所有字节，或数据 A 的字节未标记为待读取 1：数据 A 的字节中有待读取的数据	0h
12	DATABRDACC	R	数据 B 读访问状态（Data B read access status） 0：已读取数据 B 的所有字节，或数据 B 的字节未标记为待读取 1：数据 B 的字节中有待读取的数据	0h
14:13	保留			0h
15	UDDATA	R	更新数据（Update Data） 此位域自从上一次读取 IF3 寄存器组之后，是否有新的数据被加载到 IF3 寄存器中 0：未加载新数据 1：已加载新数据	0h
31:16	保留			0h

40.8.29 IF3 掩码寄存器（CAN_IF3MASK）

偏移地址：0x144

复位类型：SYSRSn

此寄存器用于配置邮箱的掩码。

位/域	名称	R/W	描述	复位值
28:0	IDMASK	R/W	标识符掩码（Identifier Mask） 0：邮箱标识符中的对应位不用于接收过滤（即可以为任意值） 1：邮箱标识符中的对应位用于接收过滤	1FFFFFFFh
29	保留			1h

位/域	名称	R/W	描述	复位值
30	DIRMASK	R/W	报文方向掩码（Message Direction Mask） 0：CAN_IF3ARB[MDIRCFG]位不用于接收过滤 1：CAN_IF3ARB[MDIRCFG]位用于接收过滤	1h
31	EXTMASK	R/W	扩展标识符掩码（Extended Identifier Mask） 当邮箱使用 11 位的标准格式的标识符时，接收到的数据帧的 11 位标识符被放置在 MIDCFG[28:18]。在接收过滤的过程中，仅考虑 MIDCFG[28:18]和对应的 IDMASK[28:18]位。 0：CAN_IF3ARB[EXTIDCFG]位不用于接收过滤 1：CAN_IF3ARB[EXTIDCFG]位用于接收过滤	1h

40.8.30 IF3 仲裁寄存器（CAN_IF3ARB）

偏移地址：0x148

复位类型：SYSRSn

此寄存器用于配置邮箱的仲裁信息。

位/域	名称	R/W	描述	复位值
28:0	MIDCFG	R/W	配置报文标识符（Message Identifier Configure） 11 位标识符-标准帧：MIDCFG[28:18] 29 位标识符-扩展帧：MIDCFG[28:0]	0h
29	MDIRCFG	R/W	配置报文方向（Message Direction Configure） 0：接收，当 TXREQEN=1 时，发送远程帧，远程帧携带具有与此邮箱相匹配的标识符；当接收到数据帧，且此数据帧的标识符与邮箱的标识符匹配时，该数据帧将被存储到邮箱中 1：发送，当 TXREQEN=1 时，发送数据帧；当接收到远程帧，且此远程帧的标识符与邮箱的标识符匹配时，如果 REMEN=1，设置此邮箱的 TXREQEN=1	0h
30	EXTIDCFG	R/W	配置扩展标识符（Extended Identifier Configure） 0：邮箱使用 11 位标准标识符 1：邮箱使用 29 位扩展标识符	0h
31	MAILEN	R/W	使能邮箱（mailbox enable） 禁用邮箱时，报文处理器忽略此邮箱，它不会被报文处理器考虑用于接收或发送 CAN 报文；使能邮箱时，它会被报文处理器用于接收或发送操作。 在 CAN 初始化期间，CPU 应该将所有未使用邮箱的 CAN_MVAL[MVAL]位复位为 0。 0：禁用 1：使能	0h

40.8.31 IF3 报文控制寄存器（CAN_IF3MCTRL）

偏移地址：0x14C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
3:0	DLCCFG	R/W	配置数据长度码（Data Length Code Configure） 注意：具有相同标识符的不同节点的邮箱需要设置相同的数据长度码。当报文处理器接收到一个数据帧时，它将根据接收到的报文设置相应邮箱中的 DLC。 0000-1000：数据帧具有 0-8 个数据字节 1001-1111：数据帧具有 8 个数据字节	0h
6:4	保留			0h
7	EOBEN	R/W	使能块结束（End of Block Enable） 此位域用于将多个邮箱级联起来构建一个 FIFO 缓冲区。当作为独立邮箱时，此位必须始终置 1。 0：当前的邮箱属于 FIFO 缓冲区块的一部分，并且不是这个 FIFO 缓冲区块的最后一个邮箱 1：当前的邮箱是一个独立的邮箱，不属于任何 FIFO 缓冲区块或当前的邮箱是 FIFO 缓冲区块中的最后一个邮箱	0h
8	TXREQEN	R/W	使能发送请求（Transmit Request Enable） 0：此邮箱没有待处理的发送请求 1：此邮箱有待处理的发送请求并且未完成发送	0h
9	REMEN	R/W	使能远程（Remote Enable） 0：接收到远程帧后，不更改 TXREQEN 1：接收到远程帧后，置位 TXREQEN	0h
10	RXIEN	R/W	使能接收中断（Receive Interrupt Enable） 0：成功接收到帧数据后，不触发 IPEND 1：成功接收到帧数据后，触发 IPEND	0h
11	TXIEN	R/W	使能发送中断（Transmit Interrupt Enable） 0：成功发送帧数据后，不触发 IPEND 1：成功发送帧数据后，触发 IPEND	0h
12	MASKEN	R/W	使能掩码（Mask Enable） 当此位置 1 时，必须在邮箱初始化期间配置邮箱的 MASK 位段（IDMASK，EXTMASK，DIRMASK），再置位 CAN_MVAL[MVAL]。 0：忽略掩码 1：使用掩码（即 CAN_IF3MASK 寄存器中的位域）进行接收过滤	0h
13	IPEND	R/W	中断挂起（Interrupt Pending） 0：此邮箱中没有挂起的中断 1：此邮箱有挂起的中断。当没有此邮箱中断源的优先级最高时，中断寄存器的中断标识符将指向此邮箱	0h
14	MLOST	R/W	报文丢失（Message Lost ） 此位域仅适用于传输方向设置为接收的邮箱。 0：CPU 上次复位此位后，没有报文丢失 1：发生了报文丢失，即报文处理器在 CAN_NDATA[NDATA]置位时接收到了新报文，并且将其存储在了此邮箱中，导致之前的报文被新报文覆盖。	0h

位/域	名称	R/W	描述	复位值
15	NDATA	R/W	新数据（New Data） 0：CPU 上次清除此位后，此邮箱的数据部分没有新数据写入 1：有新数据写入，即此邮箱的数据部分有新数据写入（由消息处理器或 CPU 写入）	0h
31:16	保留			0h

40.8.32 IF3 数据 A 寄存器（CAN_IF3DATAA）

偏移地址：0x110+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于提供访问 CAN 报文数据字节的窗口。

位/域	名称	R/W	描述	复位值
7:0	DATD0	R/W	数据字节 0（Data Byte 0）	0h
15:8	DATD1	R/W	数据字节 1（Data Byte 1）	0h
23:16	DATD2	R/W	数据字节 2（Data Byte 2）	0h
31:24	DATD3	R/W	数据字节 3（Data Byte 3）	0h

40.8.33 IF3 数据 B 寄存器（CAN_IF3DATAB）

偏移地址：0x114+(x-1)*0x40

复位类型：SYSRSn

此寄存器用于提供访问 CAN 报文数据字节的窗口。

位/域	名称	R/W	描述	复位值
7:0	DATD4	R/W	数据字节 4（Data Byte 4）	0h
15:8	DATD5	R/W	数据字节 5（Data Byte 5）	0h
23:16	DATD6	R/W	数据字节 6（Data Byte 6）	0h
31:24	DATD7	R/W	数据字节 7（Data Byte 7）	0h

40.8.34 IF3 更新使能寄存器（CAN_IF3UDEN）

偏移地址：0x160

复位类型：SYSRSn

此寄存器用于为每个邮箱单独使能 IF3 寄存器组的自动更新功能。当邮箱对应的 UDEN=1 时，此邮箱自动更新 IF3 寄存器组。此时，如果一个邮箱的 NDATA 位被激活（例如，由于接收到一个 CAN 帧被激活），这将触发将整个邮箱自动复制到 IF3 寄存器组的操作。

注意：应避免为发送邮箱使能 IF3 更新功能。

位/域	名称	R/W	描述	复位值
31:0	UDEN	R/W	使能 IF3 更新（IF3 Update Enable） 此位域用于为每个邮箱单独使能 IF3 寄存器组的自动更新功能。 0：禁用邮箱的 IF3 寄存器组的自动更新功能 1：使能邮箱的 IF3 寄存器组的自动更新功能	0h

41 局域互联网络（LIN）

41.1 术语全称、缩写描述

表格 231 术语全称、缩写描述

中文全称	英文全称	英文缩写
局域互联网络	Local Interconnect Network	LIN
通用异步收发器	Universal Asynchronous Transceiver	UART
不归零	Non-return to Zero	NRZ
有限状态机	Finite State Machine	FSM
UART 发送移位寄存器/ UART 接收移位寄存器	UART TX Shift Register/ UART RX Shift Register	UARTTXSHF/ UARTRXSHF

41.2 简介

LIN 可以用作串行通信接口，在此兼容模式下，LIN 功能与其他独立的串口模块（UART）兼容，但是寄存器和代码等不同。模块作为串口时，处于兼容模式。

通过配置，此模块可以用作 UART 或 LIN，G32R501 为了兼容 LIN 的功能，增强了硬件特性。

模块符合 LIN 规范包中规定的 LIN2.1 协议，LIN 标准基于 UART 串行数据链路格式，用于在网络节点之间进行多播传输。

41.3 UART 主要特征

（1） 支持全双工或半双工

（2） 标准 UART 通信，异步通信模式（不支持同步），标准 NRZ 格式

（3） 多缓冲接收和发送单元，支持双缓冲收发

（4） 帧格式中，每个字符中的 3~13 位均可配置，条件如下
- 停止位可配置为 1 或 2 位
- 奇偶校验位可配置为零或 1 位（奇校验或偶校验）
- 地址位模式下需要附加地址位
- 数据字长可配置为 1~8 位

（5） 独立使能的 2 个中断线，分别为 level 0 和 level 1

（6） 支持休眠模式，在多处理器通信期间能释放 CPU 资源，唤醒，接收消息

（7） 2 种多处理器通信格式，允许多个设备之间通信

（8） 支持 2^{24} 种高精度波特率
- 在 100MHz 时钟下，最大波特率为 3.125Mbits/s

- 24 位可编程波特率

（9） 能够通过 DMA 发送和接收数据

（10） 5 个错误标志和 7 个状态标志

（11） LINTX 和 LINRX 作为外部引脚，不支持 UART 硬件控制（可以通过软件实现）

41.4 LIN 主要特征

（1） 符合 LIN1.3，2.0 和 2.1 协议

（2） 多缓冲接收和发送单元

（3） 主报头能够自动生成，包括以下字段
- 同步字段
- 可编程同步中断字段
- 标识符字段

（4） 标识掩码，用于过滤消息

（5） 2 个有优先级编码中断线，用于接收，发送，ID，错误和状态

（6） 支持自动唤醒，能够生成唤醒信号，可配置超时时间

（7） 支持自动总线空闲检测，LIN2.0 校验和

（8） 支持错误检测，包括：
- 奇偶校验错误
- 同步字段错误
- 校验和错误
- 无响应错误
- 总线错误
- 位错误

（9） 支持 2^{31} 可编程传输速率（7 个小数位）

（10） 波特率最高为 20kpbs

（11） 能够通过 DMA 发送和接收数据

（12） LINTX 和 LINRX 作为外部引脚，LINRX 主电平被收发器唤醒

（13） 从机同步功能，如下：
- 同步验证
- 同步中断检测
- 更新波特率（可选）

（14） 更新唤醒/休眠

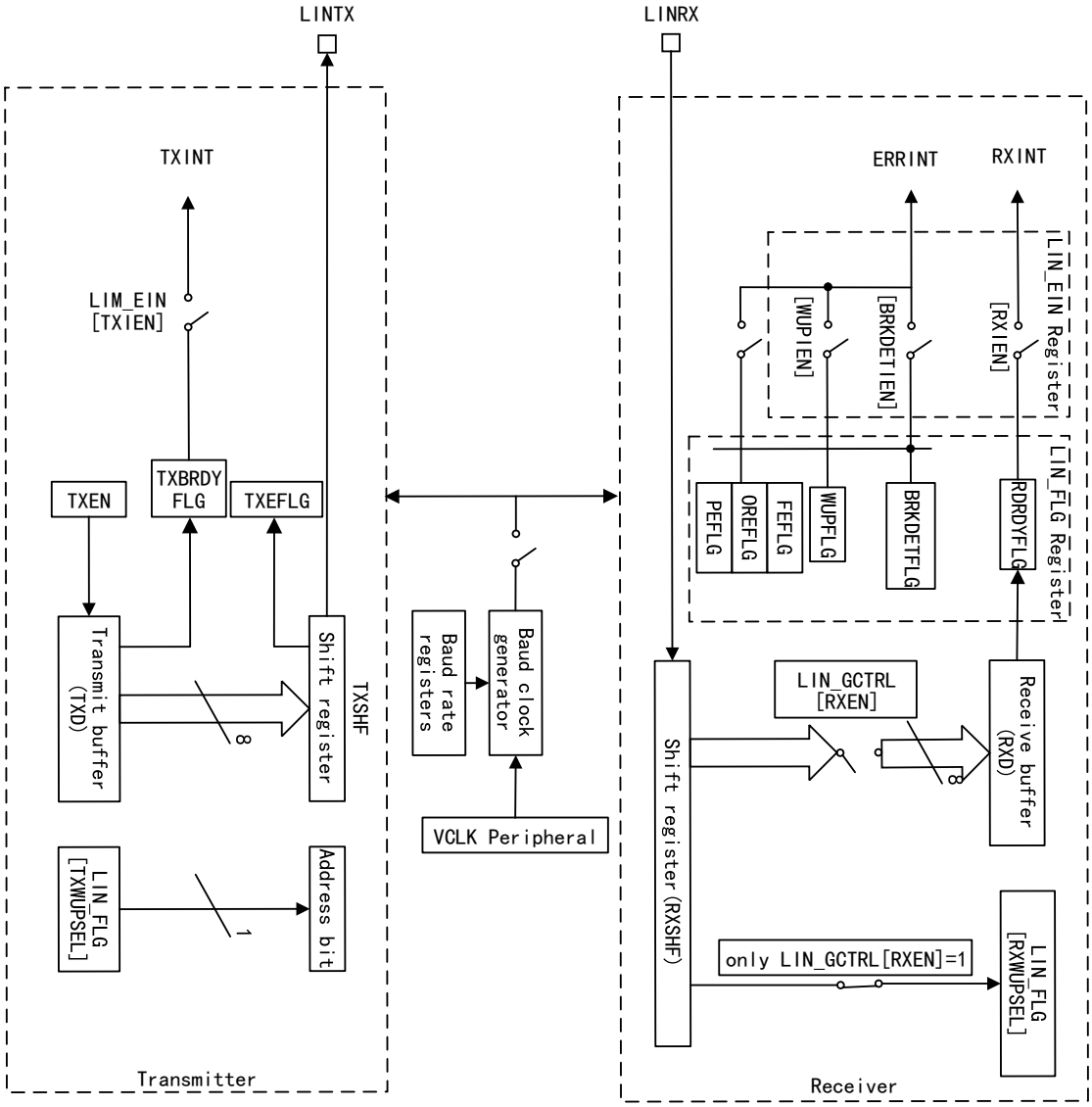
（15） 增强项：
- 同步器有限状态机（FSM）帧，支持帧处理

- 处理扩展帧
- 波特率生成器

41.5 结构框图

串口的结构框图如下所示。

图 258 串口结构框图



结构包括三个主模块，分别是接收器、发送器、波特率时钟发生器。

（1） 接收器（RX）

- 移位寄存器通过 LINRX 引脚接收数据，一次只接收一位
- 移位寄存器对接收到的数据进行移位操作，并将完整的数据发送给接收数据缓冲器
- 独立的使能位和中断
- 可以独立工作，或者在全双工模式下与发送器一起工作

（2） 发送器（TX）

- CPU 的数据被加载到发送数据缓冲器，发送数据缓冲器将数据发送给移位寄存器
- 移位寄存器对数据进行移位操作，并将移位后的数据通过 LINTX 引脚发送给外部，一次只发送一位
- 独立的使能位和中断
- 可以独立工作，或者在全双工模式下与接收器一起工作

（3） 波特率时钟发生器

- 输入时钟 VCLK 向波特率时钟发生器提供时钟源，$f_{VCLK}=f_{SYSCLK}$
- 可编辑波特率时钟发生器对时钟信号进行缩放，然后向模块提供波特率时钟

串口通过检查数据是否有中断、奇偶、溢出和帧错误来保证数据是完整的。

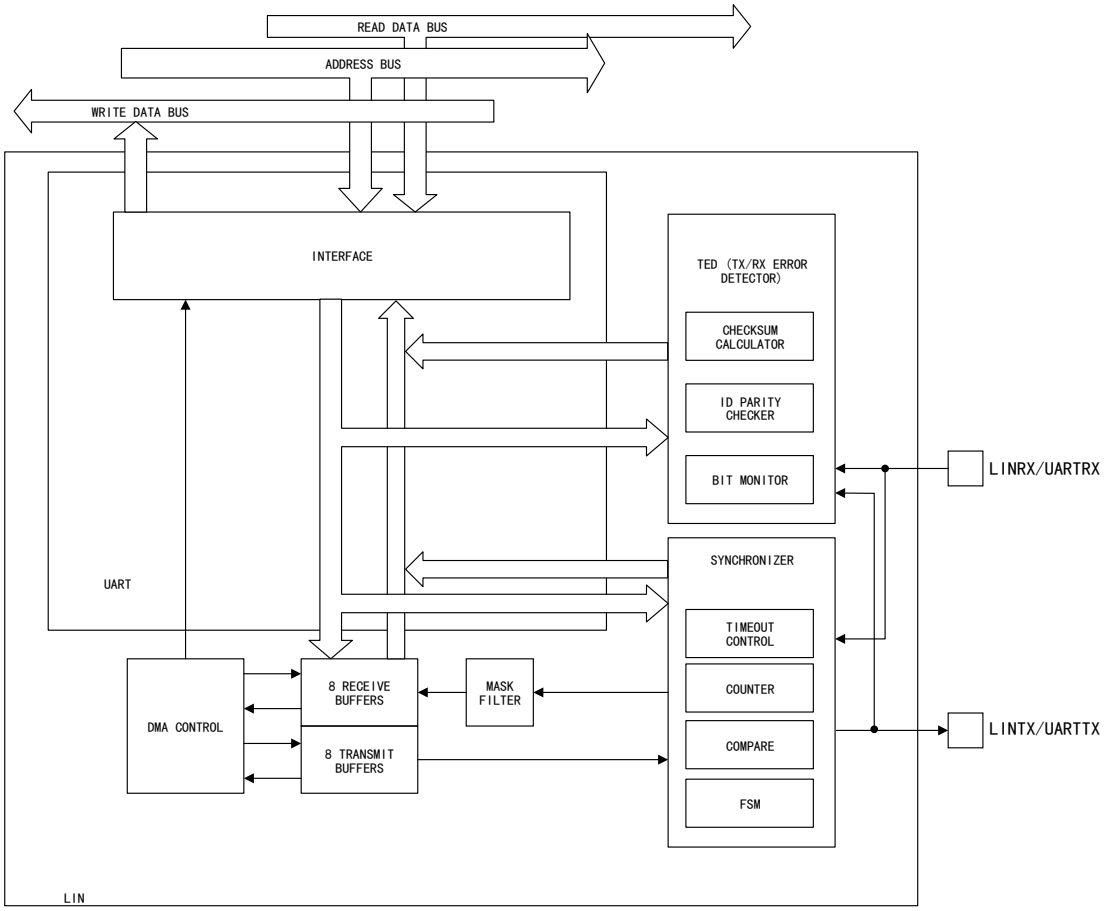
比特率可以通过 24 位波特选择寄存器进行配置，可配置成 1600 万种以上的速率。

UART/LIN 模块与 UART 模块的区别在于，前者比后者增加了以下部分：

- 掩码滤波器
- 错误监测器，包括奇偶校验计算器、校验和计算器、位监视器
- 同步器
- 多缓冲接收器
- 多缓冲发送器

为了实现 LIN 的兼容，增强了硬件部分，包括 UART 接口，DMA 控制子模块和波特率发生器，UART/LIN 块状结构框图如下图所示。

图 259 UART/LIN 块状结构框图



41.6 LIN 功能描述

41.6.1 LIN 标准

LIN 模块包含了在硬件 LIN 规范版本 1.3 和版本 2.0 中定义的 CPU 性能消耗特性。此模块的主模式兼容 LIN2.1 标准。

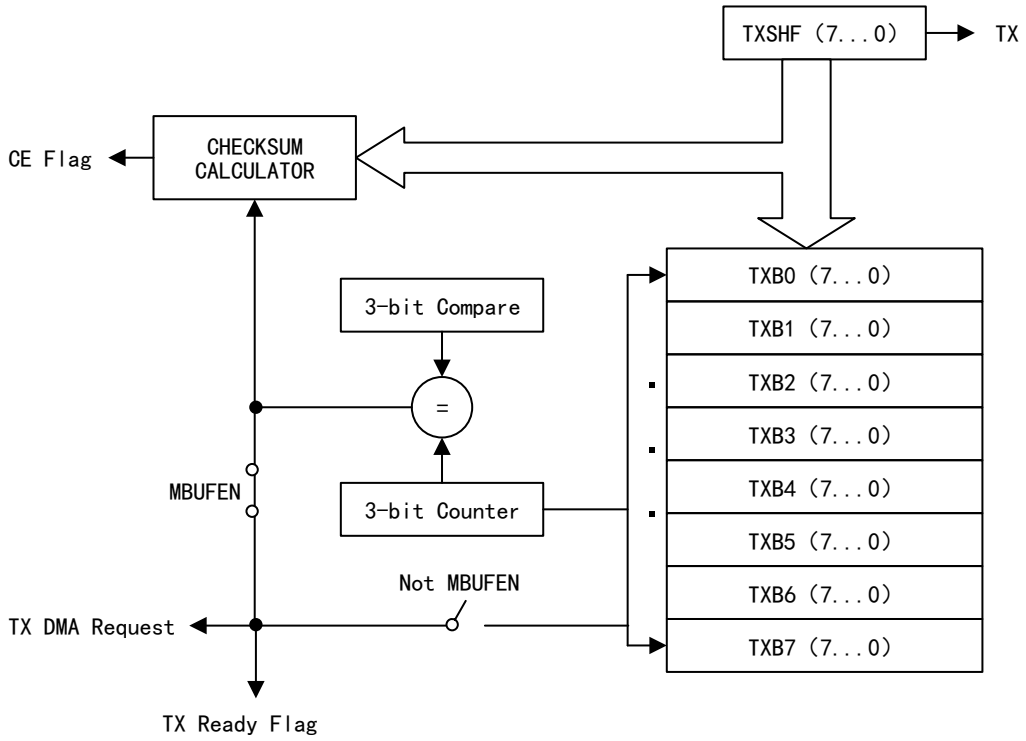
为了与 LIN2.0 标准兼容，在 LIN1.3 上实现以下功能：

- 更新唤醒/进入休眠
- 增强同步器 FSM 对帧处理的支持
- 增强对扩展帧的处理
- 增强波特率生成器
- 支持 LIN2.0 校验和

41.6.2 发送数据缓冲区

为了减轻 CPU 负担，UART/LIN 模块提供了 8 个发送缓冲区（LINTXB0 和 LINTXB1 寄存器），可用于传输 LIN N-字节（N = 1-8）响应。整个 LIN 响应字段可以预加载到发送缓冲区中。此外，如果不选择使用多缓冲区模式（MBUFEN 位清 0），也可以按字节进行 DMA 传输。

图 260 发送缓冲区



在 Multi-buffer（多缓冲区）模式下，多缓冲区的 3 位计数器用于计算从发送缓冲区寄存器传输的数据字节数量；否则，计数器将计算从 TXB0 到 TXSHF 传输的数据字节数。比较寄存器中包含预期要传输的数据字节数。ID 字段可以用于传递消息长度（LIN 协议使用标识符中的奇偶校验位，控制长度位是可编程的），如果不使用 ID 传递长度，那就使用 FLCFG 值表示期望的长度，并用于加载比较寄存器。通过配置 COMCFG 位选择使用长度控制字段或 FLCFG 值。

发送响应后，在使能相应功能的位的情况下，可以发生发送中断和发送就绪标志以及 DMA 请求。根据是否启用多缓冲区模式（MBUFEN 位），可以为每个传输字节或整个响应生成 DMA 请求。

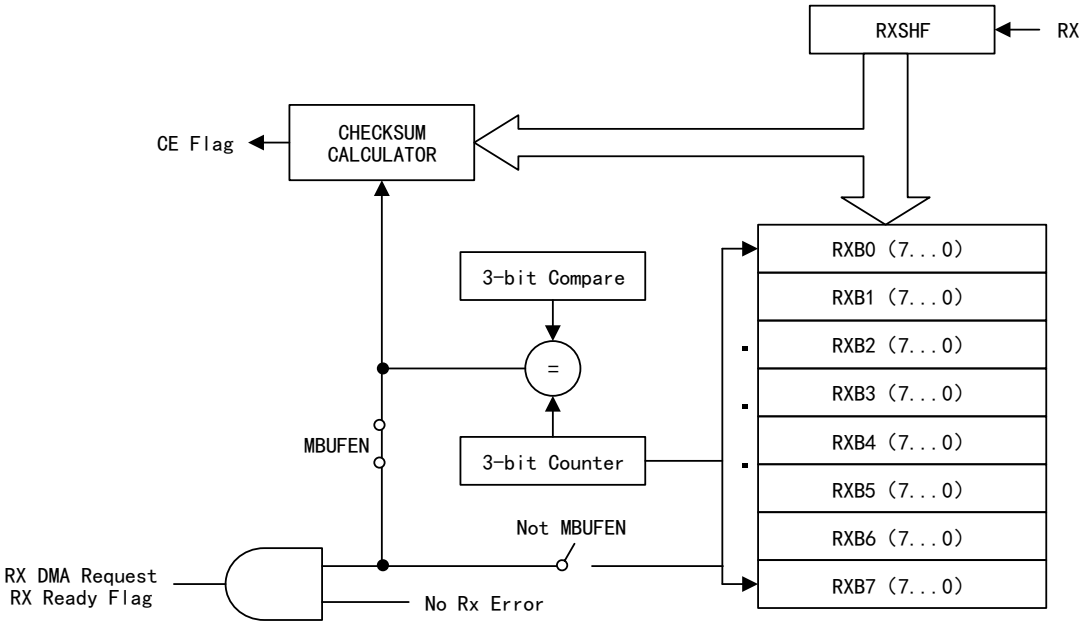
校验和字节由校验和计算器自动生成，并在数据字段传输完成后发送。多缓冲区 3 位计数器对从 TXBy 缓冲区传输到 TXSHF 寄存器的数据字节进行计数。

注意：通过使用 LINICLR 寄存器禁用相应的中断或使用 TXEN 位禁用发送器，可以消除发送中断请求，直到下一系列数据被写入发送缓冲区 LINTXB0 和 LINTXB1。

41.6.3 接收数据缓冲区

为了在中断模式或 DMA 模式下接收 LIN N-字节（N = 1-8）响应时减少 CPU 负载，UART/LIN 模块有 8 个接收缓冲区。这些缓冲区可以在 RXBy 接收缓冲区中存储整个 LIN 响应。接收缓冲区如下图所示。

图 261 接收缓冲区



校验和字节由内部计算器进行检验，如果检测出错误，校验和错误标志位会被置起，如果相应中断被使能，设备就会产生校验和错误中断。

与发送模式相反，在 Multi-buffer（多缓冲区）模式下，多缓冲区的 3 位计数器用于计算从接收移位寄存器传输到缓冲区的数据字节数量；否则，计数器将计算传输到 TXB 的数据字节数。

但与发送模式相同的是，比较寄存器中包含预期要接收的数据字节数。ID 字段可以用于传递消息长度（LIN 协议使用标识符中的奇偶校验位，控制长度位是可编程的），如果不使用 ID 传递长度，那就使用 FLCFG 值表示期望的长度，并用于加载比较寄存器。通过配置 COMCFG 位选择使用长度控制字段或 FLCFG 值。接收响应后，在使能相应功能的位的情况下，可以发生接收中断和接收就绪标志以及 DMA 请求。根据是否启用多缓冲区模式（MBUFEN 位），可以为每个接收字节或整个响应生成 DMA 请求。在确认接收前，校验和计算器会检测校验和字节。

注意：在多缓冲区模式下，RXRDYFLG 不能通过 LININTVECT0/1 寄存器来清除，而是通过 LINRXBx 寄存器清除，当 FLCFG 大于 4 时，用 LINRXB1 寄存器清除，否则用 LINRXB0 寄存器清除。

41.6.4 通信格式

UART/LIN 模块可以在 LIN 模式和 UART 模式下使用。增强的波特生成、DMA 控制和 LIN 模式操作所需的额外接收/发送缓冲区也是增强的缓冲 UART 模块的一部分。通过使能 LINEN 位来选择 LIN 模式。

UART/LIN 是围绕 UART 平台构建的，并使用类似的采样方案：每个比特 16 个样本，对样本 8、9 和 10 进行多数投票。对于 START 位，使用前三个样本。

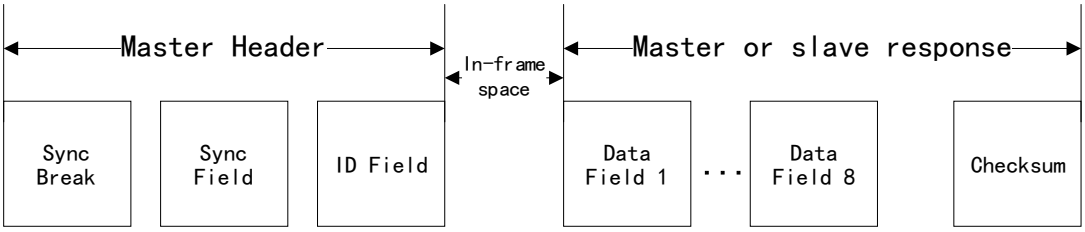
41.6.5 帧格式

LIN 协议中的消息帧如下图所示。每个帧含有：一个主报头，一个帧内响应空间（可以为 0），一个响应和字节间空间（可以为 0）。

由于 LIN 协议中没有仲裁机制，所以响应报头的多个从节点可能被认为是错误。

LIN 总线是单通道的有线与门，显性电平用二进制的 0 表示；隐形电平用二进制的 1 表示。

图 262 帧格式：主报头和响应

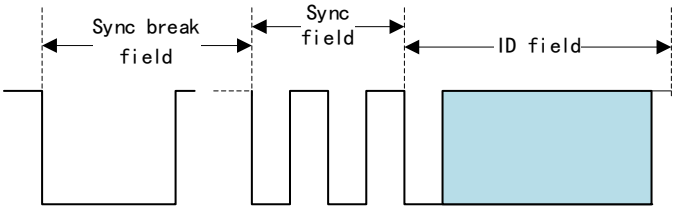


41.6.5.1 报头

消息头由主机发起（如下图所示），由三个字段序列组成:

- 同步中断字段表示消息的开始
- 传输 LIN 总线比特率信息的同步字段
- 标识字段表示消息的内容

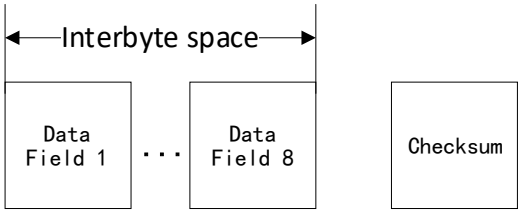
图 263 报头 3 字段：同步中断，同步和 ID



41.6.5.2 响应

响应的格式如下图所示。响应中有两种类型的字段:数据和校验和。数据字段由一个数据字节、一个起始位和一个停止位组成，总共为 10 位。LSB 先发送。校验和字段由一个校验和字节、一个起始位和一个停止位组成。校验和字节是响应数据字段中所有数据字节的 256 模的和。

图 264 LIN 消息框的响应格式



响应的格式是一个包含 N 个数据字段和一个校验和字段的流。通常 N 的取值范围是 1 到 8，除了扩展命令帧。响应的长度 N 可以通过 ID 字段的可选长度控制位

来表示（这在 LIN 1.x 之前的标准中使用），如表 27-8 所示；或通过 LINLCFG[18:16]寄存器的 FLCFG 值表示，参见下表。UART/LIN 模块支持响应长度从 1 到 8 字节，符合 LIN2.0。

表格 232 V1.3 以前的 LIN 标准使用 IDBYTE 字段位[5:4]的响应长度信息

ID5	ID4	数据字节数
0	0	2
0	1	2
1	0	4
1	1	8

表格 233 LINLCFG[18:16]配置的响应长度信息

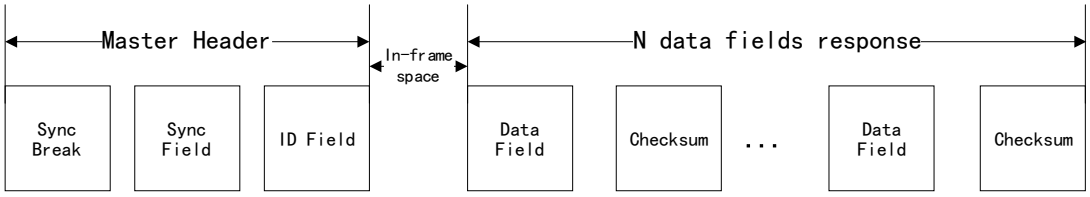
LINLCFG[18:16]	字节数
000	1
001	2
010	3
011	4
100	5
101	6
110	7
111	8

41.6.6 扩展帧处理

LIN2.0 及之前的版本中，消息帧还包括两个扩展帧，标识符为 62（用户可编程）和 63（保留扩展）。用户可编程的帧（标识符为 62 或 3Eh）的响应数据不受限制，在网络配置时就要设置好长度，并与 LIN 总线节点共享。

在接收到保留扩展帧（标识符为 63 或 3Fh）时将触发扩展帧通信，帧格式如下图所示。设置 STOEXTCOM 位可以停止扩展帧通信，需要在发出另一个标头前置位。

图 265 扩展帧响应中的可选嵌入式校验和



接收 ID 62（0x3E）时将生成 ID 中断（如果启用并且有匹配）。这个中断允许 CPU 使用一个软件计数器来跟踪发送出去的字节，并决定何时计算和插入校验和字节（建议以周期性速率）。要处理这个过程，设置 TXCHAEN 位，将启动自动

发送校验和字节。最后一个数据字段应该始终是符合 LIN 协议的校验和。

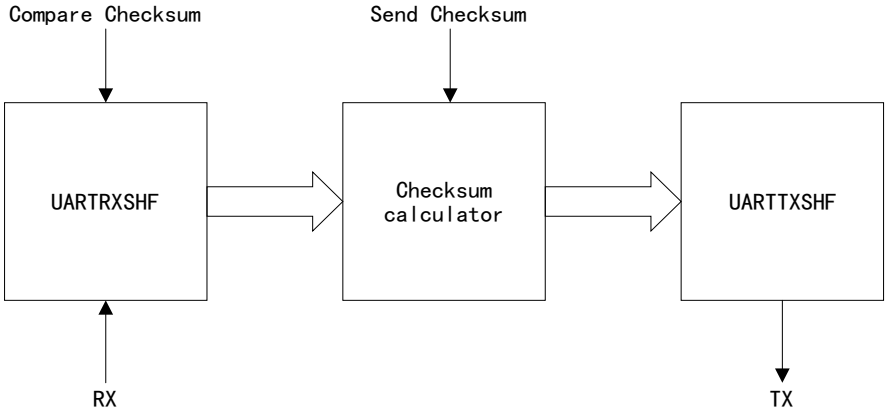
接收节点使用在网络配置时定义的校验和插入的周期性来评估正在进行的消息的校验和，并增强可靠性。

对于发送节点，每次设置发送校验和位 TXCHAEN 时，自动嵌入校验和。

对于接收节点，每次设置比较校验和位 COMPCHASUM 时比较校验和，如下图所示。

注意：LIN 2.0 增强的校验和不适用于保留的标识符。保留的标识符总是使用经典的校验和。

图 266 扩展帧的校验和比较和发送



41.6.7 同步器

在主节点和从节点之间的消息传递中，同步器有三个作用：

（1） 生成主报头数据流

（2） 与 LIN 总线同步以进行响应

（3） 在本地检测超时。

为了匹配 LIN 描述文件中指示的 LIN_speed 值，使用 BR 寄存器中的预分频器对比特率进行编程。

LIN 同步器能够在任何时候检测传入中断并初始化通信，它的功能主要有：

（1） 主报头信号生成

（2） 从检测

（3） 同步到具有可选波特率调整的消息报头，响应传输定时和超时控制

41.6.8 波特率

CPU 会在初始配置时，设置所有节点的传输波特率，定义位时间 T_{bit}。

位时间可以通过波特率寄存器 BR 中的 PSCSELL/H 位（P）和 FDIVSEL（M）位进行配置，而 SFDIVSEL（U）位可以配置额外的 3 位小数分频值，进一步微

调数据字段波特率。

BR 寄存器中预分频器值都是可编程额的，三个值的范围如下表所示：

表格 234 预分频值的范围

位域名	代表的分频值	范围
PSCSELL/H	P	0，1，2，3，4，……，2^{24}-1
FDIVSEL	M	0，1，2，3，4，……，15
SFDIVSEL	U	0，1，2，3，4，……，7

PSCSELL/H 和 FDIVSEL 分频器可用于 UART 模式和 LIN 模式来选择波特率。SFDIVSEL 值用于决定“aTVCLK”（a = 0,1）被添加到每个 T_{bit} 上，具体请参见“超小数分频器”章节。

如果在 LIN 从机处于自适应波特率模式下，ABEN 位被设置，那么在测量同步字段时，在报头接收期间自动获得所有分频器值。

LIN 协议中定义了波特率范围：1 kHz ≤ F_{LINCLK} ≤ 20 kHz

所有传输的位在 T_{bit} 周期内移进移出。

41.6.8.1 小数分频器

M 的增量为 P 的 1/16，如果要微调波特率，可以配置 M 位，以此修改整数预分频器 P。

比特时间 T_{bit} 用 VCLK 周期 T_{VCLK} 表示如下：

（1） P 不等于 0，M 为任意值：

$$T_{bit} = 16\left(P + 1 + \frac{M}{16}\right)T_{VCLK}$$

（2） P 等于 0：

$$T_{bit} = 32T_{VCLK}$$

因此，LINCLK 的频率：

（3） P 不等于 0：

$$F_{LINCLK} = \frac{F_{VCLK}}{16\left(P + 1 + \frac{M}{16}\right)}$$

（4） P 等于 0：

$$F_{CLK} = \frac{F_{VCLK}}{32}$$

41.6.8.2 超小数分频器

在 LIN 的主模式和从模式下都能使用超小数分频器进行配制。

- 主模式的字段：同步字段+标识字段+响应字段+校验和字段
- 从模式的字段：响应字段+校验和字段

基于 4 位小数分频器 M，超小数分频器使用额外的 3 位调制值，如下表所示。

其中，Data 字段可以被看作是同步字段（0x55），标识字段和响应字段。位是 1 就表示在其 T_{bit} 上增加额外的 VCLK 周期。

表格 235 超小数位调制

LINBR [30:28]	起始位	Data[0:7]								停止位
		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	
0h	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0
2h	1	0	0	0	1	0	0	0	1	0
3h	1	0	1	0	1	0	0	0	1	0
4h	1	0	1	0	1	0	1	0	1	0
5h	1	1	1	0	1	0	1	0	1	1
6h	1	1	1	0	1	1	1	0	1	1
7h	1	1	1	1	1	1	1	0	1	1

在 LIN 主模式下，同步字段+标识字段+响应字段采用位调制。

在 LIN 从模式下，标识字段+响应字段采用位调制。

波特率将在 LIN 数据字段中变化，根据 LINBR [30:28]值的 d 部分外设内部时钟的平均值：0<d<1。

平均位时间用 T_{VCLK} 表示如下:

（1） P 不等于 0，以及任意的 M 和 d （0<d<1）：

$$T_{bit}^{a} = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right] \mathrm{T}_{VCLK}$$

（2） P 等于 0：

$$T_{bit} = 32T_{VCLK}$$

瞬时位时间用 T_{VCLK} 表示如下：

（3） P 不等于 0，以及任意的 M 和 d （d=0,1）：

$$T_{bit}^{i} = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right] \mathrm{T}_{VCLK}$$

（4） P 等于 0：

$$T_{bit} = 32T_{VCLK}$$

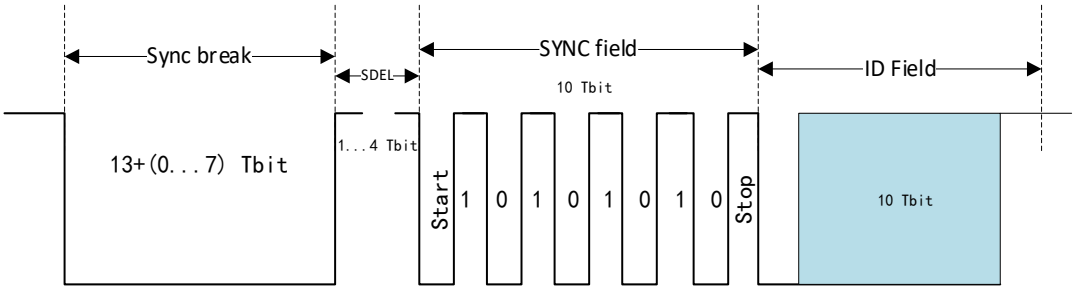
41.6.9 消息头生成

支持自动生成 LIN 协议头数据流，不需要 CPU 交互，只需要由 CPU 和 DMA 触发生成，LIN 可以自己进行生成过程。触发可以通过 CPU 或 DMA 写入 LINID[IDBYTE]实现。

报头由主机发送来发起 LIN 通信，它由同步中断字段、同步字段和识别字段组成如下图所示。

LIN 协议使用标识符中的奇偶校验位，在协议中，长度位是可编程的。

图 267 以 T_{bit} 为单位的消息头



BREAK 字段由两个部分组成：同步中断（SYNCH BREAK）和同步分隔符（SDEL）。

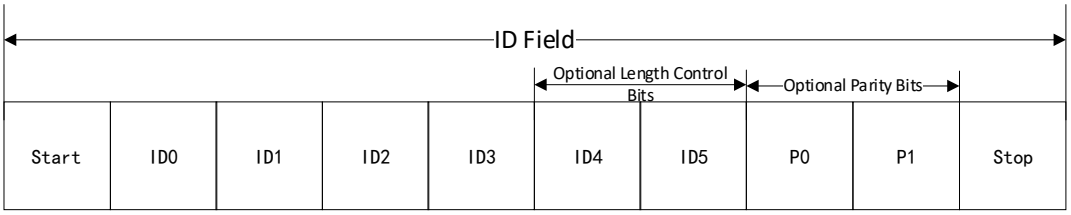
如图所示，SYNCH BREAK 可以有 13bit~20bit 的主导位。同步中断长度可以从最小值扩展到 LINCOMP 寄存器中的 3 位 SBEXTSEL 值。

SDEL 由最少 1 位（隐性）高位到最多 4 位（隐性）高位组成。分隔符标记同步中断字段的结束。同步分隔符的长度取决于 LINCOMP 寄存器中的 2 位 SDCOMPSEL 值。

同步字段（SYNCH FIELD）用于传输 T_{bit} 信息和重新同步 LIN 总线节点。由一个起始位、字节 0x55 和一个停止位组成。

标识符字段 ID 字节使用 6 个位作为标识符，并且支持可选的长度控制和两个可选的奇偶校验标识符。如果奇偶校验位被使能，那么会使用奇偶标识符并对其进行检查，此时如果不使用长度控制位，那么标识符会有 64 个，后面加上奇偶校验位；如果这两个位都不使用，那么标识符最多可以有 256 个。

图 268 ID 字段



注意：控制长度位只适用于 LIN1.3 之前的 LIN 标准。如果符合 LIN1.3 之前的标准，IDBYTE 字段将传递响应长度信息。LINLCFG 寄存器为以后版本的 LIN 协议存储响应的长度。

如果在多缓冲区模式下配置为从属的 LIN 模块正在传输数据，而新报头传入，则

该模块可能最终使用来自先前中断响应的数据（而不是与新 ID 对应的数据）进行响应。为了避免这种情况，可以执行以下步骤：

（1） 检查响应传输过程中的位错误（BE）。如果 BE 标志被设置，这表明 LIN 总线上发生了冲突（由于新的同步中断）。

（2） 在位错误 ISR 中，将 TXB0 和 TXB1 寄存器配置为下一组要在 TX 匹配上传输的数据，用于传入 ID。为了避免 TXB0/TXB1 可能会为一个 ID 写两次，在写入 TXB0/TXB1 之前，要确保没有因为位错误而进行更新。

（3） 一旦接收到完整的 ID，基于匹配，新配置的数据将由节点传输。

41.6.9.1 事件触发帧处理

LIN 2.0 协议使用事件触发的帧，这可能会导致冲突。事件触发的帧必须在软件中处理。

NRE 表示无响应错误，表示没有从机响应事件触发帧报头，此时会将 NREFLG 位置 1，并产生 NRE 中断（如果使能了中断）。如果发生了冲突，可能会先产生帧错误和校验和错误，并可能产生相应中断。帧错误（FE）和校验和错误（CSE）取决于响应从机的行为和同步。如果从服务器完全同步并在发生冲突后停止传输，那么尽管发生了冲突，也可能只标记 NRE 错误。为了检测在标记 NRE 错误之前是否已经接收到一个字节，可以使用总线忙碌标志（BUSYFLG）作为指示器。

BUSYFLG 在接收报头的第一个位时置位，并保持设置直到报头接收完成，并且再次在接收响应的第一个位时置位。如果发生碰撞，则在设置 NREFLG 标志的同一周期内清除该标志。

软件可以实现以下时序：

- 一旦接收报头完成（轮询 RXID 标志），等待 BUSYFLG 被置位或 NREFLG 标志被置位。
- 如果 BUSYFLG 没有在 NREFLG 标志之前置位，则出现个真正的无响应情况（没有数据被传输到总线上）。
- 如果设置了 BUSYFLG，则等待 NREFLG 标志设置或接收成功。如果置位了 NREFLG 标志，那么在这种情况下，总线上发生了冲突。

即使在冲突的情况下，接收到的（损坏的）数据也可以在 RX 缓冲区中访问。

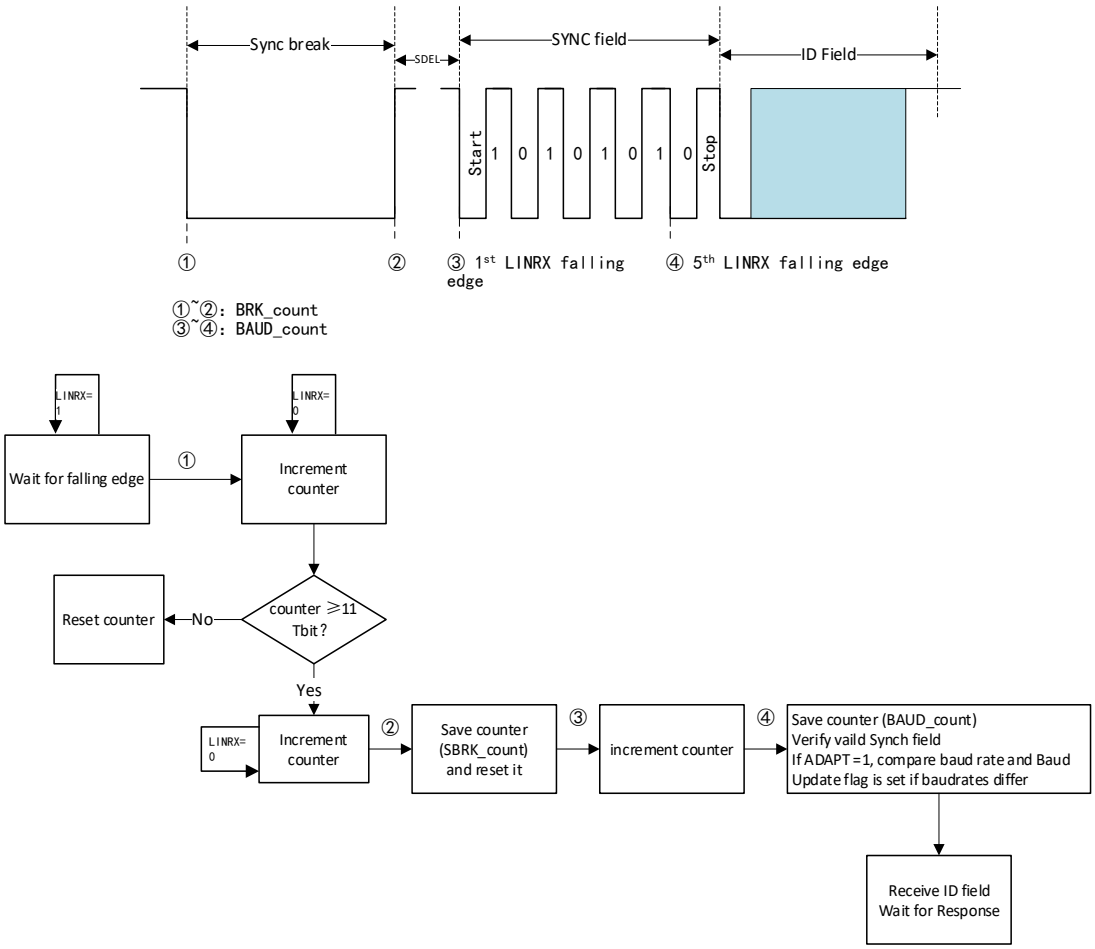
41.6.9.2 报头接收和自适应波特率

从节点波特率可以选择性地调整为检测到的比特率。

通过设置 ABEN 位来启用自适应波特率选项。在报头接收期间，从机在同步场检测期间测量波特率。如果设置了 ABEN 位，则将测量波特率与从节点编程波特率进行比较，并在必要时调整为 LIN 总线波特率。

LIN 同步器确定两个测量值：BRK_count 和 BAUD_count。在报头接收同步字段验证期间计算这些值。

图 269 同步测量，同步验证过程及波特率调整



同步器计数器以 VCLK 作为时基，用于测量相对于检测节点 Tbit 的同步中断。如上图所示，通过 BRK_count 和 BAUD_count 两个计数器，可以检测到有效的同步中断序列，在下降沿开始计数，在上升沿结束计数。

根据 LIN 协议，从节点在接收同步中断时需要使用 11Tbit 的阈值。在 LINRX 引脚的下降沿触发同步器计数器，用于检测同步中断的主导数据流。如果检测到同步分隔符，那么 LIN 会先保存计数器的数值再重置计数器。

如图中的波形图所示，在检测到 4 个连续的下降沿后，在紧接着的第 5 个下降沿会开始测量 BAUD_count。位定时计算和所需精度的一致性是根据 LIN 2.0 的建议实现的。

从设备节点的单个 Tbit 时间可以通过 BAUD_count/8 来计算。

BAUD_count 计数器右移 3 位，并舍入第一个无关位以获得一个 Tbit 单位。如果设置了 ABEN 位，则开始比较检测到的波特率与编程波特率。

提供以下公式用于评估检测到的边缘的一致性，BAUD_count1 表示 BAUD_count 右移 2 位，BAUD_count1 表示 BAUD_count 右移 3 位：

$$BAUD_count + BAUD_count1 + BAUD_count2 \leq BRK_count$$

在接收报头的过程中，会判断 BRK_count 的值是否小于 $11T_{bit}$，如果结果为“是”，那么同步中断无效。如果设置了 ABEN 位，那么将使用 LINMBRPSC 寄存器测量两个同步器计数器的值，并自动调整 LIN 总线速率。

注意：在自适应模式下，LINMBRPSC 所设置的波特率，必须小于 LIN 预期的波特率的 10%，否则可能会导致将 0x00 的数字字节被误认为是同步中断。

相对于从设备节点的断开阈值为 $11T_{bit}$，而在 LIN V1.3 中指定的断点为 13 个 T_{bit}。这意味着在设计和调整 LIN 时，需要注意波特率的设定，以防止误检测和确保通信的可靠性。如果在给定的公差范围内未检测到同步字段，则将设置不一致同步字段错误标志。如果设置在 LINIEN 寄存器中对应的位，则会产生 ISFE 中断。ID 字节应该在同步字段验证成功后接收。任何时候检测到一个有效的中断（大于 $11T_{bit}$），接收方状态机应该重置为接收这个新帧。此重置条件仅在响应状态期间有效，在报头接收期间发生额外的同步中断时无效。

注意：当发生不一致同步字段错误时，建议应用程序的操作是重置 RDY 位，并置位 RDY 位，以确保内部状态机恢复到正常状态。

41.6.10 超时控制

任何侦听总线并期望从主节点发起响应的 LIN 节点都可以标记一个无响应错误超时事件。

LIN 协议的超时事件都由硬件处理，超时事件有四种类型：

- 无响应超时错误
- 总线空闲检测
- 唤醒信号后超时
- 三个唤醒信号后超时

41.6.10.1 无响应错误

当任何期望响应的节点等待 TFRAME_MAX 时间，并且消息帧没有在允许的最大长度 TFRAME_MAX 内完全完成时，就会发生无响应错误。在此之后，在 LINFLG 寄存器的 NREFLG 位中标记无响应错误（NRE）。如果启用，将触发中断。

在 LIN 1.3 标准中规定，传输一帧的最小时间为（其中 N =数据字段数）：

$$T_{FRAME_MIN} = T_{HEADER_MIN} + T_{DATA_FIELD} + T_{CHECKSUM_FIELD} = 44 + 10N$$

最大时间范围为:

$$T_{FRAME_MAX} = T_{FRAME_MIN} * 1.4 = （44 + 10n） * 1.4$$

超时值 T_{FRAME_MAX} 来自于 N 个数据字段的值，参见下表。

表格 236 T_{bit} 单位的超时值

N	T_{DATA_FIELD}	T_{FRAME_MIN}	T_{FRAME_MAX}
1	10	54	76
2	20	64	90
3	30	74	104

N	T_{DATA_FIELD}	T_{FRAME_MIN}	T_{FRAME_MAX}
4	40	84	118
5	50	94	132
6	60	104	146
7	70	114	160
8	80	124	174

N 值要么嵌入到消息的报头 ID 字段中，要么是描述文件的一部分。在后一种情况下，是 LINLCFG 寄存器中的 CLCFG 值表示 N 的值。

注意：ID 字段的长度编码不适用于两个扩展的帧标识符，0x3E（62）和 0x3F（63）的 ID 字段。在这些情况下，ID 字段后面可以跟着任意数量的数据字节字段。此外，LIN 2.0 协议规范提到不能使用 ID 字段 0x3F（63）。对于这两种情况，NRE 将不会由 LIN 硬件处理。

41.6.10.2 总线空闲检测

总线空闲检测超时事件发生在节点检测到非活动 LIN 总线时：总线上没有检测到隐性值和显性值之间的转换。

这至少在 4 秒后发生（80000 个 F_{LINCLK} 周期，最快总线速率为 20kbps）。如果节点在设置 TIMEOUT 位时检测到总线中没有任何活动，则可以假定 LIN 总线处于休眠模式。软件可以使用 Timeout 标志来确定 LIN 总线何时处于非活动状态，并通过写入 LPREQ 位将 LIN 置于睡眠模式。

注意：标记超时后，应该在进入低功耗模式之前断言 RDY。如果在空闲时间之前总线上有不完整的帧，则需要重置接收器。

41.6.10.3 唤醒信号后超时和三个唤醒信号后超时

这两种超时情况与唤醒信号有关。发起唤醒的节点应该在定义的唤醒信号超时时间内从主服务器获得一个报头，更多情况请参见“唤醒超时”章节。

41.6.11 TX/RX 错误检测器（TED）

TED 逻辑由位监视器、ID 奇偶校验器和校验和错误组成。TED 能检测到的错误源有：

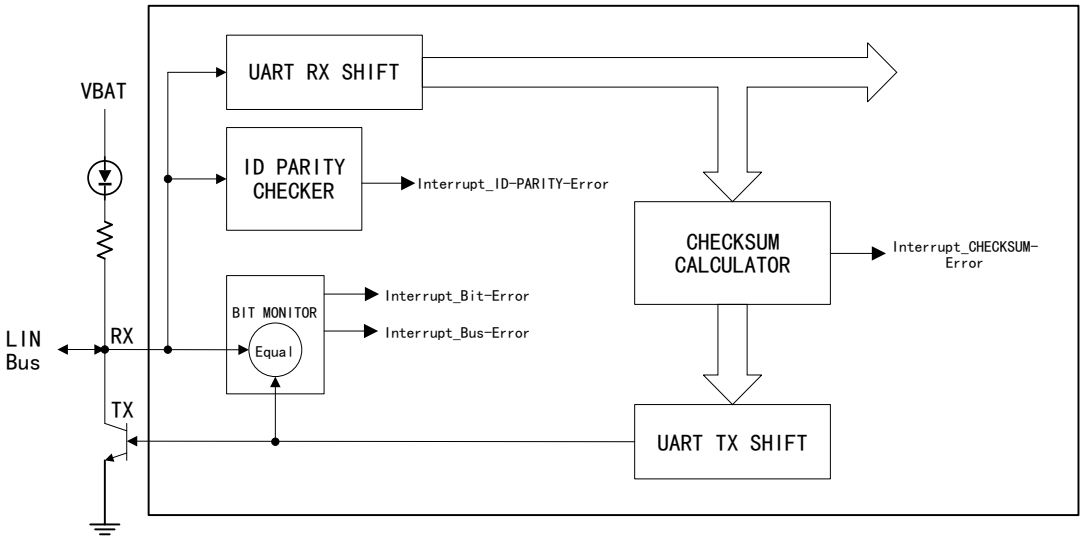
- 物理总线错误（PBEFLG）
- 标识奇偶校验错误（PEFLG）
- 位错误（BEFLG）
- 校验和错误（CSEFLG）

设置相应中断使能位后，才能生成中断。如果在帧结束前没有检测到这些错误，那么发送数据和接收数据都被视为有效。

41.6.11.1 位错误（Bit Error）

当监测到的比特值与发送的比特值不一致时，在比特时间检测到位错误。向 BE 发送信令后，不迟于下一个字节终止传输。位监视器通过回读 LINRX 引脚来确保 LINTX 中传输的位是 LIN 总线上的正确值，如下图所示：

图 270 TX/RX 错误检测器



如果在从响应期间由于接收报头而发生位，则不会为新帧设置 NREFLG/TIMEOUT 标志。

41.6.11.2 物理总线错误（Physical Bus Error）

如果在总线上不能生成有效的消息（总线短接到 GND 或 VBAT），则必须由主设备检测到物理总线错误。位监视器在报头传输过程中检测到 PBE，如果不能产生同步中断（例如，因为到 VBAT 的总线短缺）或如果不能产生同步中断分隔符（例如，因为到 GND 的总线短缺）。一旦同步分隔符被验证，所有其他在监控和发送的比特值之间的偏差都被标记为该帧的位错误。

41.6.11.3 ID 奇偶校验错误（ ID Parity Error）

如果使能了奇偶校验，当发送的 ID 字节的两个校验位中有一个不等于接收节点计算出的奇偶校验时，就会检测到 ID 奇偶校验错误 PEFLG （ID parity error）。这两个奇偶校验位使用以下混合奇偶校验算法生成：

$$P0 = ID0 \oplus ID1 \oplus ID2 \oplus ID4 \text{ (even Parity)}$$

$$P1 = ID1 \oplus ID3 \oplus ID4 \oplus ID5 \text{ （odd Parity）}$$

如果检测到 PEFLG，则标记 PEFLG，并且接收到的 ID 无效。详情请参见“消息过滤和验证”章节。

41.6.11.4 校验和错误（Checksum Error）

如果所有接收到的数据字节（包括增强校验和类型的 ID 字节）加上校验和字节的计算模 256 之和没有产生 0xFF，则检测到校验和错误并在接收端标记。模数为 256 的和是通过进位加法对每个字节进行计算的，其中每个加法的进位被加到其结果和的 LSB 上。

对于发送节点，在消息末尾发送的校验和字节是所有数据字节的倒和（见“传输节点的经典验证和生成”图），这是经典的校验和实现。校验和字节是标识符字节和所有数据字节的倒和（见“传输节点 LIN2.0 兼容校验和生成”图），用于 LIN 2.0 兼容的增强校验和实现。经典的校验和实现应该始终用于保留标识符 60 到 63。

因此，在这种情况下，CHASUMCFG 位将被覆盖。对于信号携带帧标识符（0 到 59），使用的校验和类型取决于 CHASUMCFG 位。

图 271 传输节点的经典验证和生成

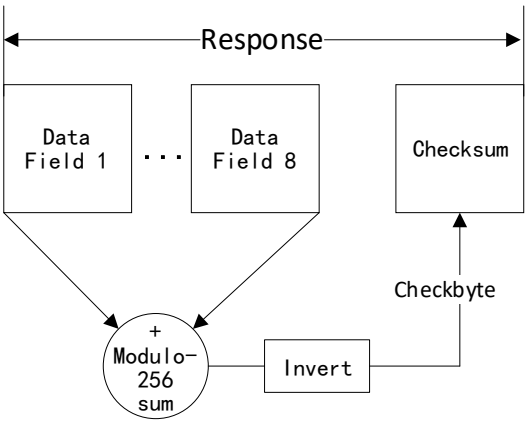
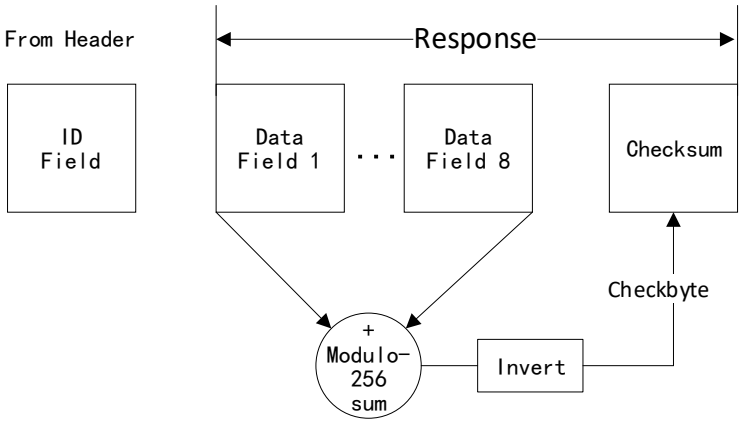


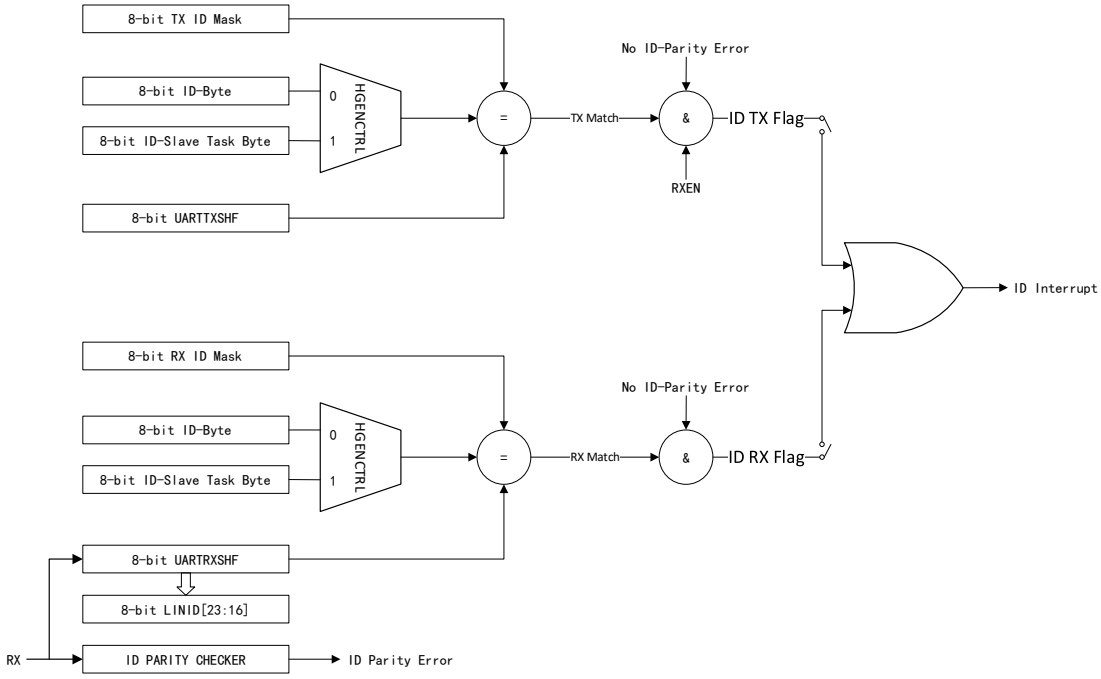
图 272 传输节点 LIN2.0 兼容校验和生成



41.6.12 消息过滤和验证

消息过滤使用整个标识符来确定参与响应的节点，接收或传输响应。因此，使用两个接收掩码，如下图所示。

图 273 ID 接收，过滤和验证



在接收报头期间，所有节点会对 ID 字段进行过滤，节点需要确定响应的类型是发送还是接收，对此，消息 ID 过滤分别有一个接收掩码和一个发送掩码。所有节点在接收到 ID 字段后，与存储在 LINID 寄存器中的 ID-SlaveTask 的标识符作对比，并且通过 LINIDMASK 的发送或接收掩码位来过滤无效标识符。

如果有一个 RX 匹配没有奇偶校验错误且 RXEN 位被置位，使能后将有一个 ID RX 标志和中断将被触发。如果有一个没有奇偶校验错误的 TX 匹配且 TXENA 位被置位，则会有一个 ID TX 标志，如果在 LINIEN 寄存器中使能则会触发中断。

被屏蔽的位变得不在乎比较。要为一组标识符构建掩码，可以使用 XOR 函数。

例如，使用 LINID[7:0] = 0x20 建立一个掩码来接受 id 0x26 和 0x25，即比较 5 个最高有效位并过滤 3 个最低有效位，则接受掩码可以为：

$$(0x26 + 0x25) \oplus 0x20 = 0x07$$

HGENCTRL 位的值将使不同的掩码对接收到的标识符与 IDBYTE 作不同的处理。当 HGENCTRL=0 时，LIN 节点将两者做对比，并用 LINMASK 寄存器的掩码过滤掉不需要比较的标识符；当 HGENCTRL=1 时， 全 1 掩码（0XFF）会过滤掉所有的接收标识符，无论 IDBYTE 的值是多少，都会有匹配的 ID。

注意：如果有一个 RX 匹配没有奇偶校验错误和 RXEN 位被置位，将有一个 ID RX 标志和中断将被触发，如果启用。一个全 0 的掩码将把移位寄存器中接收到的标识符的所有位与 LINID[7:0]中的 ID-BYTE 字段进行比较。全部为 1 的掩码将过滤接收到的标识符的所有位，并且没有匹配。

- 如果 HGENCTRL = 1
  - 使用 RXID 掩码和 TXID 掩码，将接收 ID 与 ID-Slave-Task 字节进行比较
  - 全 1 的掩码总是会导致匹配
  - 全零掩码意味着所有位必须相同才能匹配

- 如果掩码有一些位是 1，那么这些位将不会用于过滤标准

- 如果 HGENCTRL = 0
  - 使用 RXID 掩码和 TXID 掩码将接收到的 ID 与 ID 字节进行比较
  - 全 1 的掩码将导致没有匹配
  - 全零掩码意味着所有位必须相同才能匹配
  - 如果掩码有一些位是 1，那么这些位将不会用于过滤标准

在接收报头时，接收到的标识符被复制到接收 ID 字段 LINID[23:16]。如果没有奇偶校验错误，并且存在 TX 匹配或 RX 匹配，则设置相应的 TX 或 RX ID 标志。如果使能 ID 中断，则生成 ID 中断。

ID 中断产生后，CPU 可能会读取接收 ID 字段 LINID[23:16]，并决定将什么响应加载到发送缓冲区中。

注意：当字节 0 写入 TXB0 （LINTXB0[31:24]）时，自动生成响应传输。

在多缓冲区模式下：

- TXBRDYFLG 标志被置位时，表示发送准备完成，意味着所有响应数据字节和校验和字节已经被加载到移位寄存器。
- TXEFLG 标志被置位时，意味着发送缓冲区和移位寄存器被清空，校验和字节已经被发送。

而在非多缓冲区模式下：

- 每次将一个字节复制到移位寄存器时，TXBRDYFLG 标志就会被置位，并且在校验和字节复制到 UARTTXSHF 寄存器后，才设置帧的最后一个字节。
- 每次清空 TXD0 和 UARTTXSHF 寄存器时，TXEFLG 标志就会被置位，除了必须要传输校验和字节的帧的最后一个字节。

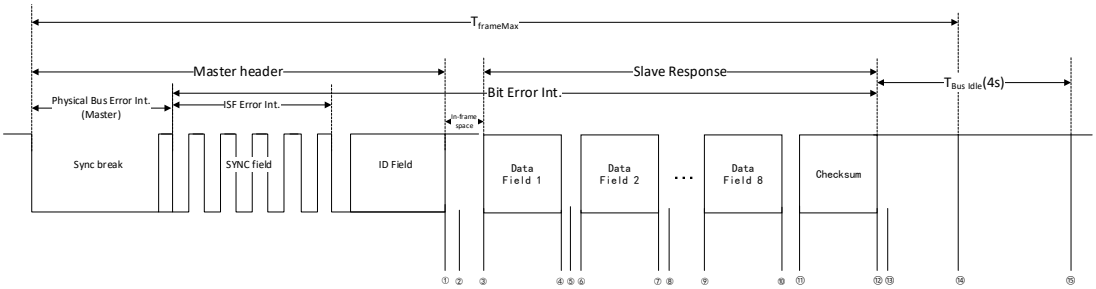
所有从接收节点将使用 ID 字节的 8 位都用来验证标识符（如果使能了奇偶校验）。如果检测出错误，UART/LIN 将会对错误的标识符进行标记。

41.6.13 LIN 中断

LIN 和 UART 模式有一个共同的中断块，如 41.5 节所述。UART/LIN 模块中有 16 个中断源，其中 8 个仅为 LIN 模式，如表 241 所示。

下图显示了一个指示 LIN 中断的时序和顺序的消息框。

图 274 显示 LIN 中断时间和顺序的 LIN 消息框



①Parity Error Int. ID interrupt
②FE
③TX Int.(Single buffer)
④RX Int.(Single buffer) Overrun Error Int.
⑤FE
⑥TX Int.(Single buffer)
⑦RX Int.(Single buffer)
⑧FE
⑨TX Int.(Single buffer)
⑩Frame Error Int.(FE)
⑪TX Int.(Multi buffer/last byte single buffer)
⑫Checksum Error Int. RX Int.(Multi-buffer/last byte single buffer)
⑬FE
⑭No Response Error Int.
⑮Timeout-Bus Idle(4s) Int.

41.6.13.1 处理 LIN 中断

当处理中断时，在清除全局中断标志（LIN_GLB_INT_CLR）之前，清除标志寄存器（LINFLG）中的相应标志。ISR 应遵循以下指导。这将防止任何虚假或重复的中断发生。

- 清除 LINFLG 寄存器中的 LIN 中断标志。
- 读取 LIN 中断状态寄存器以确保该标志被清除。
- 清除 LIN_GLB_INT_CLR 中的全局中断标志位。

41.6.14 LIN DMA 接口

LIN DMA 接口使用 UART DMA 接口逻辑。接收（RXDMA 请求）和发送（TXDMA 请求）的 DMA 请求可用于 UART/LIN 模块。根据是否使用多缓冲区启用控制位（MBUFEN）启用多缓冲区模式，DMA 传输有两种模式。

注意：不要使用 DMA 向多个从 ID 传输数据。写入 LINID 寄存器将启动一个新的传输。在 LIN 状态机准备好接受新的 ID 之前，DMA 将写入 LINID 寄存器。这样做将导致 LIN 错过此传输。

41.6.14.1 发送 DMA 请求

在启用了多缓冲区选项的 LIN 模式中，在一次传输之后（在 LINTXBx 寄存器中最多 8 个数据字节存储在传输缓冲区中），生成一个 DMA 请求，以便为下一次传输重新加载传输缓冲区。如果禁用多缓冲区选项，则 DMA 请求将以每个字节为基础生成，直到传输所有字节。此 DMA 功能分别使用 TXDREQEN 和 TXDREQCLR 位使能和禁用。

41.6.14.2 接收 DMA 请求

当启用多缓冲区选项时，如果将接收到的响应（最多 8 个数据字节）传输到接收缓冲区（RXBy），则生成 DMA 请求。如果禁用了多缓冲区选项，那么 DMA 请求将以字节为单位生成，直到接收到所有预期的响应数据字段。此 DMA 功能分别使用 RXDREQEN 和 RXDREQCLR 位使能和禁用。

41.6.15 发送/接收数据配置

在以 LIN 模式传输或接收数据之前，软件应该执行的配置步骤如下所示。寄存器编程的顺序是可以调整的，条件是：在配置 LIN 的整个过程中，LINGCTRL1 寄存器中的 RDY 位被清除为 0。

（1） 通过置位 RST 位使能 LIN。

（2） 配置 LIN 前，将 SWnRST 设置为 0。

（3） 通过置位 RXPEN 和 TXPEN 位来启用 LINRX 和 LINTX 引脚。

（4） 通过设置 LIN 模式位选择 LIN 模式。

（5） 通过设置时钟位选择主或从模式。

（6） 通过设置 LINGCTRL1 选择所需的帧格式（校验和，奇偶校验，长度控制）。

（7） 通过设置 MBUFEN 位选择多缓冲区模式。

（8） 通过设置 LINBR 寄存器选择需要进行通信的波特率。

（9） 通过设置 LINMBRPSC 寄存器设置通信使用的最大波特率。

（10） 设置 CONTSUS 位，使 LIN 在当前接收或传输完成之前不会停止模拟断点（此位仅在仿真环境中使用）。

（11） 如果需要，设置 LBEN 位将发射器连接到接收器内部（此功能用于执行自检）。

（12） 如果要接收数据，请选择接收使能 RXEN 位。

（13） 如果要传输数据，请选择发送使能 TXEN 位。

（14） 选择 LINIDMASK 寄存器中的 RXIDMASK 和 TXIDMASK 字段。

（15） 配置 LIN 后，将 SWnRST 配置为 1。

（16） 执行发送或配置数据（参见“消息过滤和验证”章节、“发送数据配置”章节和“配置数据配置”章节）。

使能发送中断并释放软件复位后，LIN 只能生成一个 DMA 请求，硬件不会主动触发传输中断，只能由软件将数据写入到相关寄存器，再传输到 LIN TX 来进行第一次传输。

41.6.15.1 发送数据配置

如果 TXPEN 位和 TXEN 位都设置为 1，则 LIN 发送器启用。如果未设置 TXPEN 位，则 LINTX 引脚作为通用 I/O 引脚而不是 LIN 函数引脚。在 TXEN 位设置为 1 之前，任何写入 TXB0 的值都不会被传输。这两个控制位都允许 LIN 发送器独立于接收器保持非活动状态。

TXIDFLG 是在使用 TX Match 接收到有效的 LIN ID 后设置的。如果启用，将生成 ID 中断。

以单缓冲区模式发送数据

当 MBUFEN 位设置为 0 时，表示选择单缓冲区模式。等待发送的数据写入 TXB0（TXBRDYFLG 被置位）后，被传输到移位寄存器，再由 LIN 传输数据。如果缓冲区和移位寄存器是空的，那么 TXEFLG 位会被置位。

发送数据有以下几种方式：

（1） 轮询发送就绪标志

（2） 发送中断

（3） 直接存储器存取（DMA）

在轮询方法中，在将数据写入 TXB0 之前，软件可以轮询 TXBRDYFLG 位走高。通过选择轮询方法，CPU 不必要地过载。为了避免这种情况，可以使用中断或 DMA 方法。要使用中断方法，需要设置 TXIEN 位。要使用 DMA 方法，需要设置 TXDREQEN 位。当设置 TXBRDYFLG 位时，会产生中断或 DMA 请求。当 LIN 完成所有挂起帧的传输时，UARTTXSHF 寄存器和 TXB0 为空，TXBRDYFLG 位被设置，启用则生成中断/DMA 请求。所有数据都已传输完成后，中断/DMA 请求应该停止。这可以通过禁用发送中断（TXICLR） / DMA 请求（TXDREQCLR 位）或禁用发送器（清除 TXEN 位）来实现。如果将 CONTSUS （Send checksum）位设置为 1 使能校验和方案，则校验和字节将在当前字节传输后发送。SC 位将在校验和字节传输后被清除。

TXBRDYFLG 标志不能通过读取 LINIVOx 寄存器中相应的中断偏移量来清除。

以多缓冲模式发送数据

当 MBUFEN 位设置为 1 时，表示选择多缓冲区模式。与单缓冲区模式类似，可以使用轮询、DMA 或中断方法来写入要传输的数据。传输的数据必须根据字节数写入 LINTXB0 和 LINTXB1 寄存器。LIN 等待数据写入 LINTXB0 寄存器的字节 0（TXB0），并将编程的字节数传输给 TXSHF，以便自动逐个传输。如果通过将 CONTSUS 位设置为 1 来启用校验和方案，则校验和将在程序数据字节数（由 FLCFG 字段表示）的最后一个字节传输后发送。CONTSUS 位将在校验和字节传输后被清除。

41.6.15.2 接收数据配置

如果 RXPEN 位和 RXEN 位都设置为 1，则 LIN 接收器可以接收消息。如果没有置位 RXPEN 位，则 LINRX 引脚作为通用 I/O 引脚而不是 LIN 函数引脚。

RXIDFLG 是在使用 RX Match 接收到有效的 LIN ID 后置位的。如果启用，将生成 ID 中断。

以单缓冲区模式接收数据

当 MBUFEN 位被清除为 0 时，选择单缓冲区模式。在这种模式下，LIN 在将新接收到的数据 RXSHF 传输到 RXB0 时设置 RXRDYFLG 位。当 RXRDYFLG 中的新数据被读取后，UART 会清除 RXRDYFLG 位。此外，当数据从 RXSHF 传

输到 RD0 时，如果在接收到的数据中检测到任何这些错误条件，LIN 将设置 FEFLG、OEFLG 或 PEFLG 标志。可配置中断能力支持这些错误条件。

接收数据有以下几种方式：

（1） 轮询接收就绪标志

（2） 接收中断

（3） 直接存储器存取

在轮询方法中，软件可以轮询 RXRDYFLG 位，一旦 RXRDYFLG 位设置高，就从 LINRXB0 寄存器的 RXB0 字节读取数据。通过选择轮询方法，CPU 不必要地过载。为了避免这种情况，您可以使用中断或 DMA 方法。要使用中断方法，需要设置 RXIEN 位。要使用 DMA 方法，必须设置 RXDREQEN 位。当 RXRDYFLG 位被设置时，会产生一个中断或一个 DMA 请求。如果通过将比较校验和（COMPCHASUM）位设置为 1 来启用校验和方案，则将在当前接收的字节上比较校验和，该字节预计是校验和字节。一旦接收到校验和，COMPCHASUM 位将被清除。如果存在校验和错误，CSEFLG 将立即被标记。

以多缓冲区模式接收数据

当 MBUFEN 位设置为 1 时，表示选择多缓冲区模式。在这种模式下，LIN 在接收缓冲区和校验和字段中接收到编程后设置 RXRDYFLG 位，即完整帧。错误条件检测逻辑类似于单缓冲区模式，除了它监视完整的帧。与单缓冲区模式类似，您可以使用轮询、DMA 或中断方法来读取数据。接收到的数据必须根据字节数从 LINRD0 和 LINRXB1 寄存器中读取。对于小于或等于 4 的 FLCFG，从 LINRD0 寄存器中读取将清除 RXRDYFLG 标志。对于大于 4 的 FLCFG，从 LINRXB1 寄存器中读取将清除 RXRDYFLG 标志。如果通过在接收数据期间将比较校验和（COMPCHASUM）位设置为 1 来启用校验和方案，那么在接收到由 FLCFG 字段指示的编程数据字节数之后接收到的字节将被视为校验和字节。一旦校验和完成，COMPCHASUM 位将被清除。

41.7 低功耗模式

UART/LIN 模块可置于局部或全局低功耗模式。全局低功耗模式由系统启用 ，不受 UART/LIN 模块控制。在全局低功耗模式下，所有到 UART/LIN 的时钟都关闭，因此模块完全不活动。如果接收器在接收数据时请求全局低功耗模式，则 UART/LIN 完成当前接收，然后进入低功耗模式，即只有清除 Busy 位（LINFLG[3]）后模块才进入低功耗模式。

当 LINRX 引脚上没有活动超过 4s（这可以是一个恒定的隐性或显性水平）或当接收到睡眠命令帧时，LIN 模块可以进入低功耗模式。一旦设置 Timeout 标志（LINFLG[4]）或一旦收到 Sleep 命令，应用软件必须设置 LPREQ 位以使模块进入局部低功耗模式。唤醒信号将终止 LIN 总线的休眠模式。

在接收和发送时启用局部低功耗模式时，如果唤醒中断被启用，并且在接收器接收数据时请求低功耗模式，那么 UART/LIN 立即产生一个唤醒中断来清除掉电

位。因此，阻止 UART/LIN 进入低功耗模式，完成电流接收。否则，如果唤醒中断被禁用，则 UART/LIN 完成当前接收，然后进入低功耗模式。

41.7.1 休眠模式

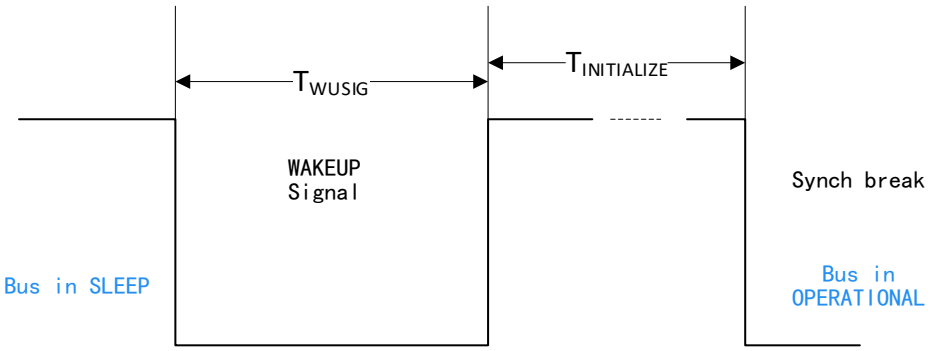
在 LIN 协议中，使用 Sleep 命令可以让总线处于休眠模式。Sleep 命令的诊断主请求帧的标识符为 0x3C，第一个数据段为 0x00。

通过设置 LINCTRL2[LPREQ]位，UART/LIN 模块可以自己进入局部低功耗模式，原理是使时钟源不作用于模块。但是，在局部掉电的情况下，时钟可以通过特定的方式进入 UART，因此所有寄存器都是可以被访问的。

41.7.2 唤醒

唤醒中断的使能或禁用决定了接收引脚上的低电平能否使模块自动退出低功耗模式。如果设备生成了进入低功耗模式的请求，并且唤醒中断被使能，那么在接收引脚检测到低电平时可以触发唤醒，并清除 LPREQ 位。在 LIN 模式下，任何节点都可以通过发送唤醒信号来终止休眠模式，如下图所示。从节点检测到总线处于休眠模式，并且唤醒请求挂起，将发送唤醒信号。唤醒信号是 T_{WUSIG} 在 LIN 总线上的主导值；这至少是 5 个 T_{bits} 的 LIN 总线波特率。唤醒信号是通过发送一个 0xF0 字节来生成的，其中包含 5 个显性 Tbits 和 5 个隐性 T_{bits}。

图 275 唤醒信号的产生



理想状态下（没有噪声和负载），向缓冲区写入 0xF0 会加载到发送器，满足 T_{WUSIG} 的唤醒信号时序要求。将 WUPGEN 位置 1 后，设备将缓冲区的数值读取出来，然后传输唤醒信号。

需要注意的是，只有在处于软件复位且节点处于掉电模式时，才能对 WUPGEN 位进行设置或复位的操作，清除的条件是检测到有效的同步中断。主机发送唤醒请求，然后在检测到唤醒脉冲后退出掉电模式，并且清除 WUPGEN 位，使得主机不会再次发出唤醒请求。

LIN 收发器在接收到唤醒信号后，将其转换为微控制器进行在 RX 引脚上具有主导电平或信号到稳压器的唤醒。置位 LPREQ 位时，如果 LIN 模块检测到 RX 引脚上的隐性到主导沿（下降沿），且使能了 LINIEN 寄存器，它将产生唤醒中断。

唤醒请求被定义为 LIN 收发器在总线上检测到的超过 150ms 的电平。LIN 从机准备在唤醒信号结束后 100ms 内监听总线。

41.7.3 唤醒超时

唤醒信号在没有等到回应后，在 1.5s 内暂停发送唤醒请求的行为，LIN 协议定义为超时，具体规则如下：在唤醒信号发送后，所有节点等待主节点发送报头。如果在唤醒信号后 150ms 内未检测到同步信号，发起唤醒请求的节点将重新发送唤醒信号，如果三次唤醒尝试均未成功，1.5s 内将不会再次发起请求。

为了使 LIN2.0 能够兼容 LIN1.3 的超时条件，触发唤醒信号的节点可以通过设置 LINMBRPSC 寄存器调整位时基，以满足目标时间：128Tbits×编程预分频器。

LIN 协议的唤醒超时处理由硬件实现。

41.8 仿真模式

在仿真模式下，程序被挂起时，UART/LIN 模块的操作由 CONTSUS 位决定。

在调试模式下，CONTSUS 位被置起或被清除，决定了计数器是继续工作还是停止。

仿真时，LINFLG 寄存器的标志位不会被读操作所影响。

41.9 UART 功能描述

41.9.1 收发格式

由于通信格式因具体应用而异，UART/LIN 的许多属性都是用户可配置的。UART 配置选项有：

- UART 框架格式
- UART 定时模式
- UART 波特率
- UART 多处理器模式

41.9.2 框架格式

UART 采用可编程帧格式。所有框架包括以下内容:

- 一个起始位
- 1 ~ 8 位数据位
- 零个或一个地址位
- 零个或一个奇偶校验位
- 一个或两个停止位

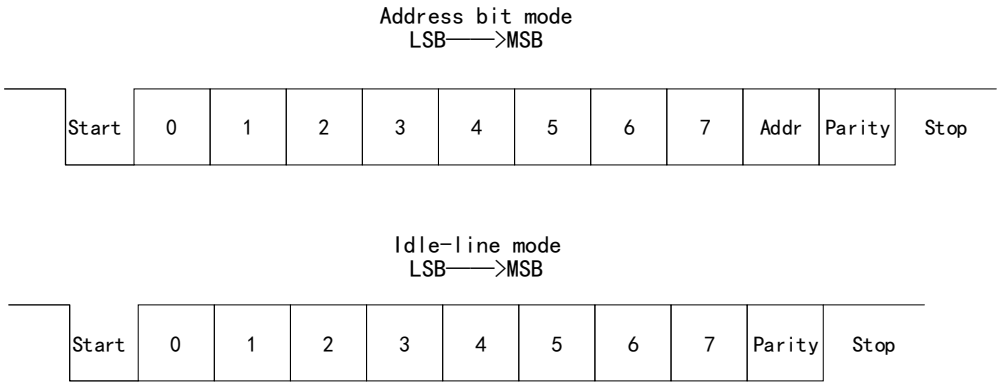
发送和接收的数据帧可以通过 LINGCR1 寄存器中的位进行编程。数据的传输采用非归零（NRZ）格式，这表示在空闲状态下，发送和接收线路呈现逻辑高电平。每一帧的传输以一个起始位开始，此时发送器将 UART 线拉低（逻辑低）。在起始位之后，数据帧的最低有效位（LSB）将首先被发送和接收。

当 UART 被设置为地址位模式时，每个数据帧中都会包含一个地址位；而在空闲线模式下，数据帧不带地址位。带有和不带地址位的帧格式如下图所示。

如果使能了奇偶校验位，那么每个数据帧中都会有一个奇偶校验位。这个奇偶校验位的值取决于帧中 1 的数量，以及选择的奇偶校验类型。下中展示的两个示例均设定了奇偶校验。

所有数据帧的末尾都会包含一个停止位，该停止位始终为高电平。帧末的高电平用于标识帧的结束，以确保通信设备之间保持同步。如果在 LINGCR1 寄存器中配置了停止位，则会发送两个停止位。下方示例中的每个帧使用的是一个停止位。

图 276 典型的 UART 数据帧格式



41.9.3 异步定时格式

可以通过设置 LINGCTRL1[1]使 UART 进入异步时序模式。

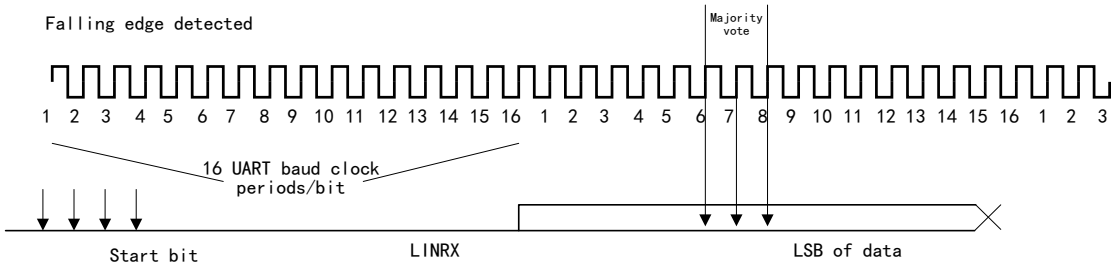
在异步定时模式下，UART 只需要连接接收数据线和发送数据线。每个时钟周期采样一次，而每个比特在帧中占用 16 个 UART 波特时钟周期，因此每个比特由 16 次采样组成。如果参与通信的设备的波特率不匹配，容易导致接收错误。

当接收器接收到 LINRX 引脚上的下降沿信号后，如果随后的四个采样均为逻辑电平 0，UART 就会识别为有效的起始位。一旦检测到下降沿，UART 会假设接收帧的过程开始，并将自身与总线进行同步。

LINRX 在至少连续四个 UART 波特时钟周期内保持低电平时，才会被认为是有效起始位，否则总线将处于空闲状态。然后，UART 在第 7,8,9 个周期对 LINRX 进行采样，以获取每个位的值，最终确定存储在接收移位寄存器中的值。在此模式中，UART 通过在位中间采样的方式，有效减少了由于传输延迟、上升和下降时间以及噪声而产生的误差。

发送器在每个比特的持续时间上同样为 16 个 UART 波特时钟周期。在一个比特的首个时钟周期，发送器会将该比特的值传送到 LINTX 引脚上，并在接下来的 16 个 UART 波特时钟周期内保持 LINTX 上的值不变。

图 277 异步通信位定时



41.9.4 波特率

UART/LIN 内部会生成一个串行时钟，由 VALK 和预分频系数 P、M 决定，

在异步定时模式下，UART 根据以下公式产生一个波特时钟：

$$\text{UARTCLK Frequency} = \frac{\text{VCLK Frequency}}{P + 1 + \frac{M}{16}}$$

$$\text{Asynchronous baud value} = \frac{\text{UARTCLK Frequency}}{16}$$

当 P=0 时，

$$\text{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{32}$$

超小数分频

在异步模式（空闲行和地址位模式）中，UART 可以使用超小数分频器调制更精细的波特率。表格（普通配置）中带有 1 的位将在其 Tbit 上添加一个额外的 VCLK 周期。对于字符长度超过 10 的情况，那么调制表是原始的翻转版本，见（最大配置）表。

在字段中，波特率会根据 BR 寄存器的值而变化，由 d、U、T_{VCLK}、P、M 等系数决定。计算方式如下：

0<d<1

对于 P≠0，以及所有 M 和 d（0 或 1），T_{VCLK} 表示瞬时位时间：

$$T^{i}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right] T_{VCLK}$$

对于 P=0，T_{VCLK} 表示瞬时位时间：

$$T_{bit}=32T_{VCLK}$$

对于 P≠0，以及所有 M 和 d （0<d<1），T_{VCLK} 表示平均位时间：

$$T^{a}bit = \left[16\left(P + 1 + \frac{M}{16}\right) + d\right] T_{VCLK}$$

对于 P=0，T_{VCLK} 表示平均位时间：

$T_{bit}=32T_{VCLK}$

表格 237 UART 模式的超分数比特调制（普通配置）

正常配置=起始位+ 8 位数据位+停止位										
LINBR[30:28]	起始位	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	停止位
0h	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0
2h	1	0	0	0	1	0	0	0	1	0
3h	1	0	1	0	1	0	0	0	1	0
4h	1	0	1	0	1	0	1	0	1	0
5h	1	1	1	0	1	0	1	0	1	1
6h	1	1	1	0	1	1	1	0	1	1
7h	1	1	1	1	1	1	1	0	1	1

表格 238 UART 模式的超分数比特调制（最大配置）

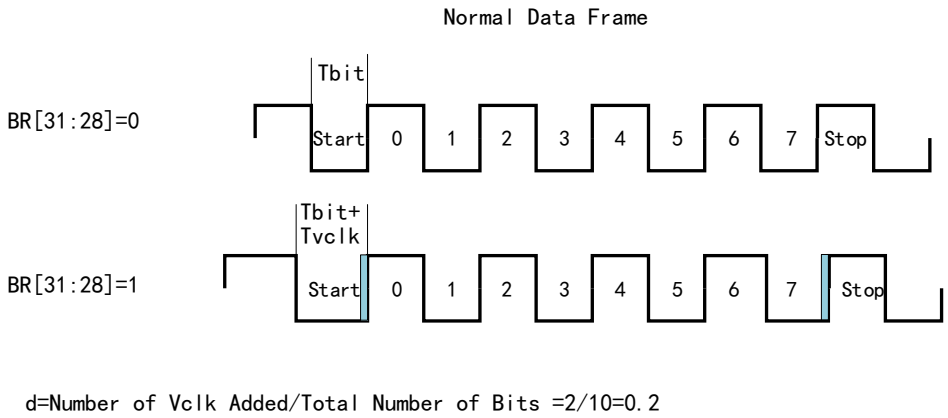
最大配置=起始位+ 8 位数据位+ 地址位+奇偶校验位+停止位 0 +停止位 1													
LINBR [30:28]	起始位	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	地址位	奇偶校验位	停止位 0	停止位 1
0h	0	0	0	0	0	0	0	0	0	0	0	0	0
1h	1	0	0	0	0	0	0	0	1	0	0	0	0
2h	1	0	0	0	1	0	0	0	1	0	0	0	1
3h	1	0	1	0	1	0	0	0	1	0	1	0	1
4h	1	0	1	0	1	0	1	0	1	0	1	0	1
5h	1	1	1	0	1	0	1	0	1	1	1	0	1
6h	1	1	1	0	1	1	1	0	1	1	1	0	1
7h	1	1	1	1	1	1	1	0	1	1	1	1	1

表格 239 UART 模式的超分数比特调制（最小配置）

最小配置=起始位+ 1 数据位+停止位			
LINBR[30:28]	起始位	D[0]	停止位
0h	0	0	0
1h	1	0	0
2h	1	0	0
3h	1	0	1
4h	1	0	1

最小配置=起始位+ 1 数据位+停止位			
5h	1	1	1
6h	1	1	1
7h	1	1	1

图 278 超分数除法示例



表格 240 不同 P 值的比较波特值，异步模式

VCLK = 50 MHz				
24 位寄存器值		波特率选择		百分误差
十进制	十六进制	理想值	实际值	
26	00001A	115200	115740	0.47
53	000035	57600	57870	0.47
80	000050	38400	38580	0.47
162	0000A2	19200	19172	-0.15
299	00012B	10400	10417	0.16
325	000145	9600	9586	-0.15
399	00018F	7812.5	7812.5	0.00
650	00028A	4800	4800	0.00
15624	003BA0	200	200	0.00
624999	098967	5	5	0.00

41.9.5 多处理器通信模式

UART 可以连接多个串行通信设备进行数据传输。在多处理器环境中，UART 可以通过寄存器选择空闲线模式或地址位模式，并且可以将几个数据帧发送到单个或所有设备上。在向单个设备发送数据时，接收设备需要识别何时对其进行寻址。当接收到的消息不针对某个设备时，该设备可以选择忽略后续的数据。当 UART 网络中仅存在两台设备时，则不需要进行寻址，因此也不需要复杂的多处理器通信方案。

如果不在多处理器环境中，所有帧都可以被软件认为是数据帧。此时，空闲线模式和地址位模式是有区别的，地址位协议使每个帧存在一个额外的地址位。

虽然 UART 支持全双工通信，但是如果 UART 使用的协议在同一时间只有一个设备进行传输，那么 UART 不进行仲裁。

41.9.5.1 空闲线多处理器模式

地址帧与数据帧的区别是帧前面的空闲周期数量，地址帧有大于等于 10 个空闲 bit，数据帧则少于 10 个。下图展示了两种帧的格式。

空闲线模式有两种传输地址帧的方法：

- 人为地在编程时，在前一个块的最后一个数据帧与新块的地址帧之间插入一个空闲周期。此方法只能通过延迟环路来实现。
- 配置 UART 使其在前一个块的最后一个数据帧与新块的地址帧之间自动发送一个空闲周期。此方法可以通过使用发送缓冲区和 TXWUPSEL 位来实现。步骤如下：

（1） TXWUPSEL 位写 1

（2） 将虚拟数据值写入 LINTXD 寄存器，发送器移位寄存器为空时触发 UART 开始空闲周期

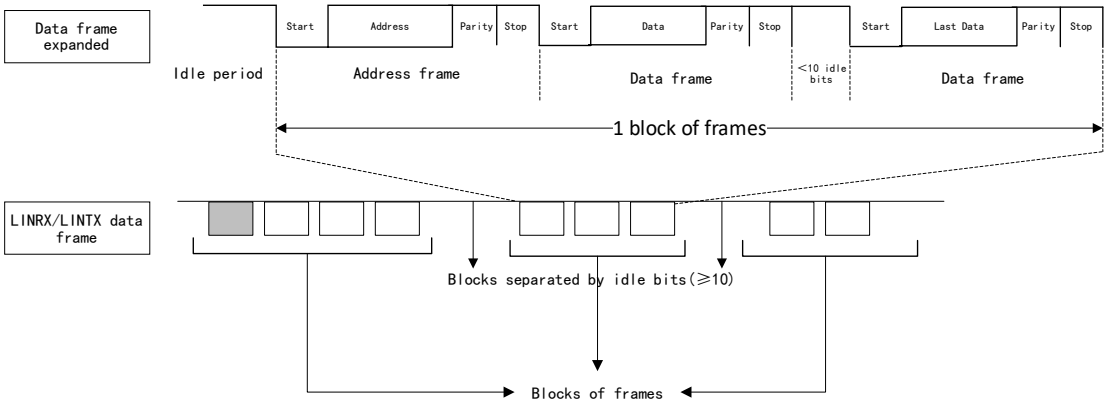
（3） UART 清除 TXWUPSEL 标志

（4） 将地址值写入 LINTXD

在步骤 3 时，那么 UART 清除了 TXWUPSEL 位的同时，设置 TXRDTFLG 后，如果 TX 中断被使能，会中断由 LINTXD 向 UARTTXSHF 的传输，所以不要用软件轮询等待 UART 清除的 TXWUPSEL 位。

使用空闲线多处理器通信时，必须要在地址帧前有 10 个以上的空闲周期，并且为了避免帧之间有 10 个周期的延迟，数据帧需要快速写入发送器。否则可能会导致设备出现错误。

图 279 空闲行多处理器通信格式



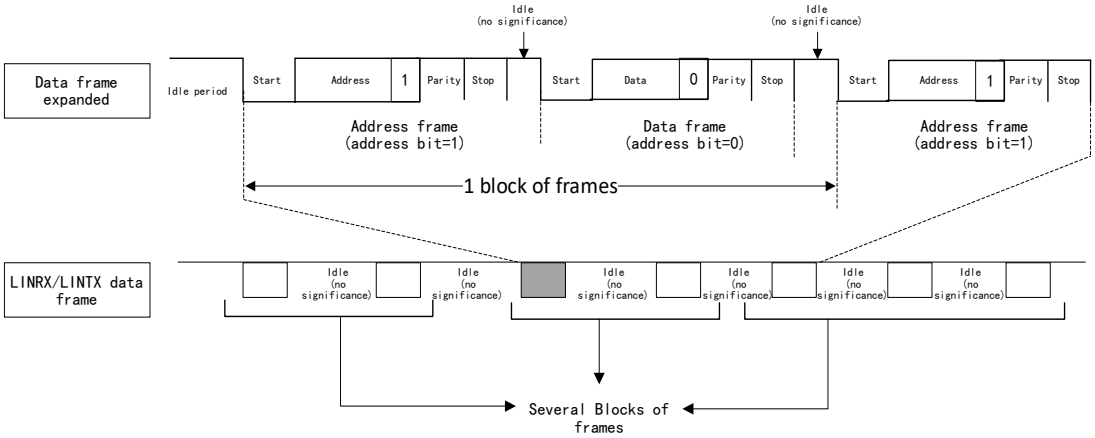
41.9.5.2 地址位多处理器模式

在地址位协议中，每个帧的数据段后都有一个地址位，地址位的值用于决定这个

帧是地址帧或是数据帧，如果是 1，那么这个帧称之为地址帧，如果是 0，则称之为数据帧。这种模式下，空闲时间的时序变得不重要。

地址位模式下的帧格式如下图所示：

图 280 地址位多处理器通信格式



在地址位模式中，不需要提前对 LINTXD 进行虚拟写操作，在 TXWUPSEL 被软件写入 1 后会被当做地址位发送。当 LINTXD 的内容从 TXWUPSEL 寄存器移位后，此位会被清除，因此除了地址位（TXWUPSEL）之外，其他的帧都可以作为数据帧。

41.9.6 发送数据

UART 发送数据有两种模式：单缓冲区模式和多缓冲区模式。通过配置 LINGCTRL1[MBUFEN]位选择模式。

发送功能由 LIN_PCTRL0[TXPEN]位和 LIN_GCTRL1[TXEN]控制，TXPEN 位使能发送引脚，只有被设置为 1 时，LINTX 引脚才能被用作 UART 引脚。TXEN 使能发送，被设置为 1 时，才开始传输缓冲区的值。当两个位都是 0 时，发送器独立于接收器。

以单缓冲区模式发送数据

当发送器准备好将数据写入 TXD 时，TXBRDYFLG 位被置位，写入后，数据被传输到 TXSHF，然后发送数据。当 TXD 和 TXSHF 都是空的时，TXEFLG 被置位。

传输数据有以下几种方式：

（1） 轮询：数据被写入 TXD 之前，CPU 会不断访问 TXBRDYFLG 位，造成运行过载。

（2） 接收中断：需要设置 TXIEN 位，使能中断。当传输完所有挂起帧时，TXD 和 TXSHF 都为空，TXBRDYFLG 被置位，此时会产生中断。传输完成后，应该通过设置 TXICLR 位，禁用中断。

（3） DMA：需要设置 TXDREQEN 位，使能 DMA 请求。当传输完所有挂起帧时，TXD 和 TXSHF 都为空，TXBRDYFLG 被置位，此时会产生 DMA 请求。传输完成后，应该通过清除 TXEN 位，禁用发送器。

注意：不能通过 LINIVOx 寄存器清除 TXBRDYFLG 位。

以多缓冲区模式发送数据

与单缓冲区模式类似，多缓冲区模式传输数据的方式相同，但是数据必须被写入 TXD 寄存器，在 UART 等待数据被写入的过程中，还要将字节数传输给 TXSHF 逐个传输。

41.9.7 接收数据

UART 接收数据有两种模式：单缓冲区模式和多缓冲区模式。通过配置 LINGCTRL1[MBUFEN]位选择模式。

接收功能由 LINPCTRL0[RXPEN]位和 LINGCTRL1[RXEN]控制，RXPEN 位使能接收引脚，只有被设置为 1 时，LINRX 引脚才能被用作 UART 引脚。RXEN 使能接收，被设置为 1 时，才开始传输缓冲区的值。

与发送引脚不同，接收引脚可以在检测到有效空闲时间并有数据到达引脚时自动接收数据。

以单缓冲区模式接收数据

当 RXD 将接收到的数据写入 RXD 时，RXBRDYFLG 位被置位，在传输过程中，如果出现帧错误、溢出错误、校验错误，LINFLG 寄存器里的相应标志位会被置位。在 RXD 的数据被读取后，标志位被清除。

发生帧错误、溢出错误或校验错误时，WUPFLG 和 BRKDETFLG 也会被置位，但是不一定会在数据被写入 RXD 的过程中发生。

传输数据有以下几种方式：

（1） 轮询：RXD 的数据被读取之前，CPU 会不断访问 RXBRDYFLG 位，造成运行过载。

（2） 接收中断：需要设置 RXIEN 位，使能中断。传输完成后，应该通过设置 RXICLR 位，禁用中断。

（3） DMA：需要设置 RXDREQEN 位，使能 DMA 请求。

以多缓冲区模式接收数据

与单缓冲区模式类似，多缓冲区模式传输数据的方式相同，但是错误检测逻辑中不包括完整帧。

41.9.8 多缓冲模式

UART 通过多缓冲区来减少 CPU 负载（配置 MBUFEN）。UART 的发送和接收都有 8 个独立的缓冲区。

有一个 3-bit 计数器计算寄存器到缓冲区或缓冲区到寄存器的数据字节数，通过 LINLCFG[FLCFG]加载比较器包含预期发送或接收的字节数量。

在接收到一个无错误的响应后，接收中断可以通过 LINIVOx 寄存器被使能，RXRDYFLG 位可以通过标志寄存器被置位，可以生成接收 DMA 请求。

在发送一个响应后，发送中断可以被使能，RXRDYFLG 位可以通过标志寄存器被置位，可以生成发送 DMA 请求。接收和发送多缓冲区的功能框图分别如下图所示。

图 281 接收缓冲区

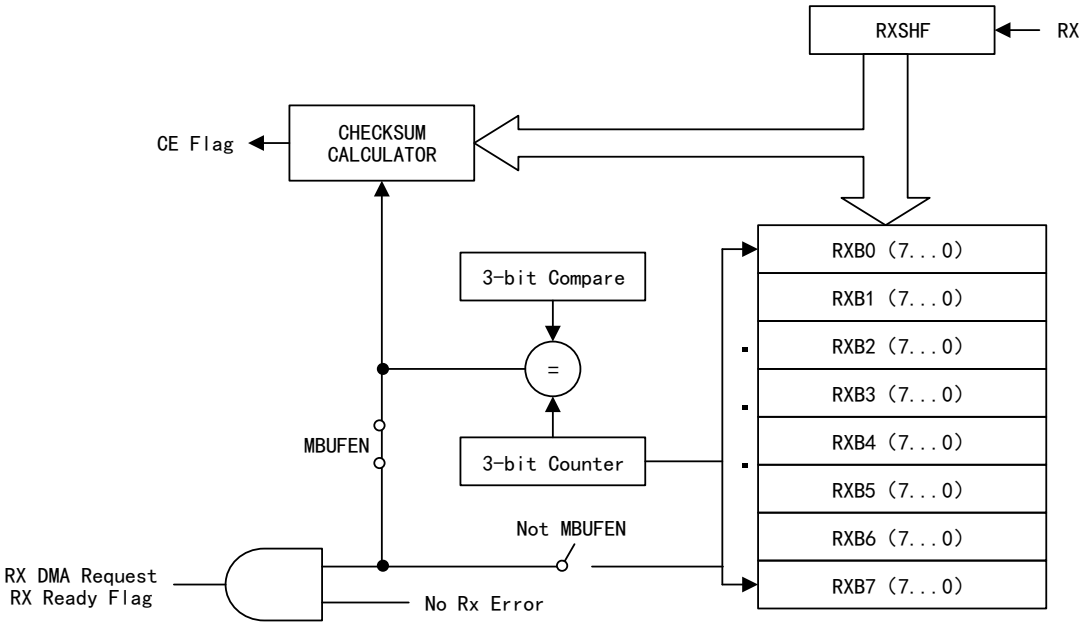
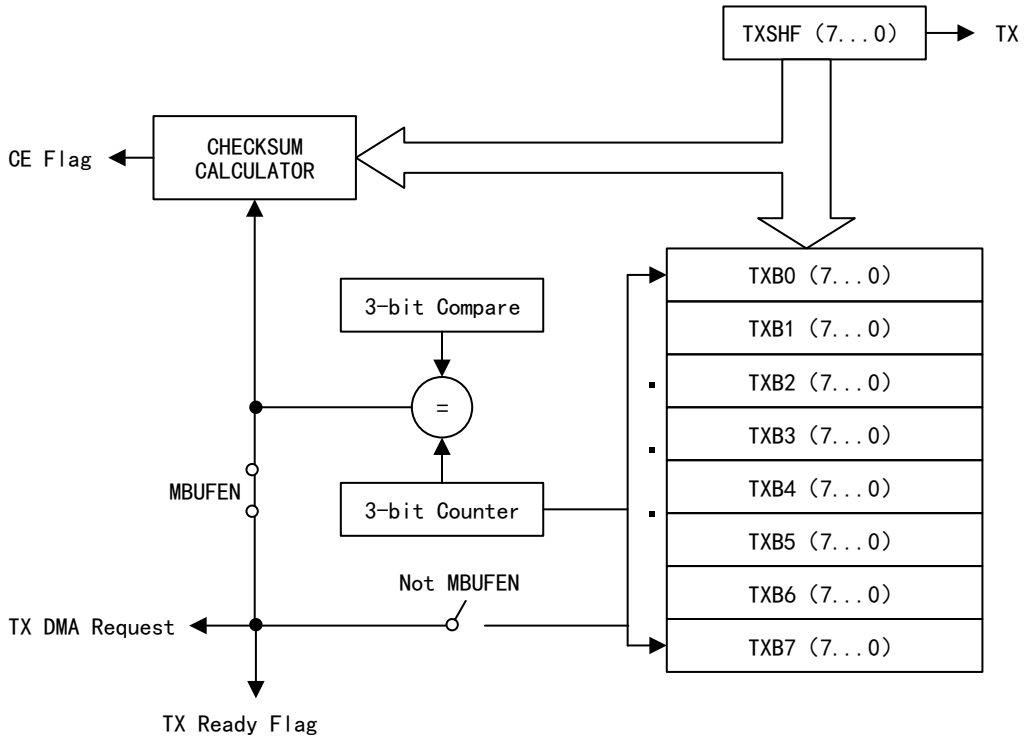


图 282 发送缓冲区



41.9.9 中断

UART 有两个中断线 INT0 和 INT1，用于产生可配置优先级的中断。通过矢量中断管理器控制。管理结构框图如下图所示。

图 283 中断管理结构框图

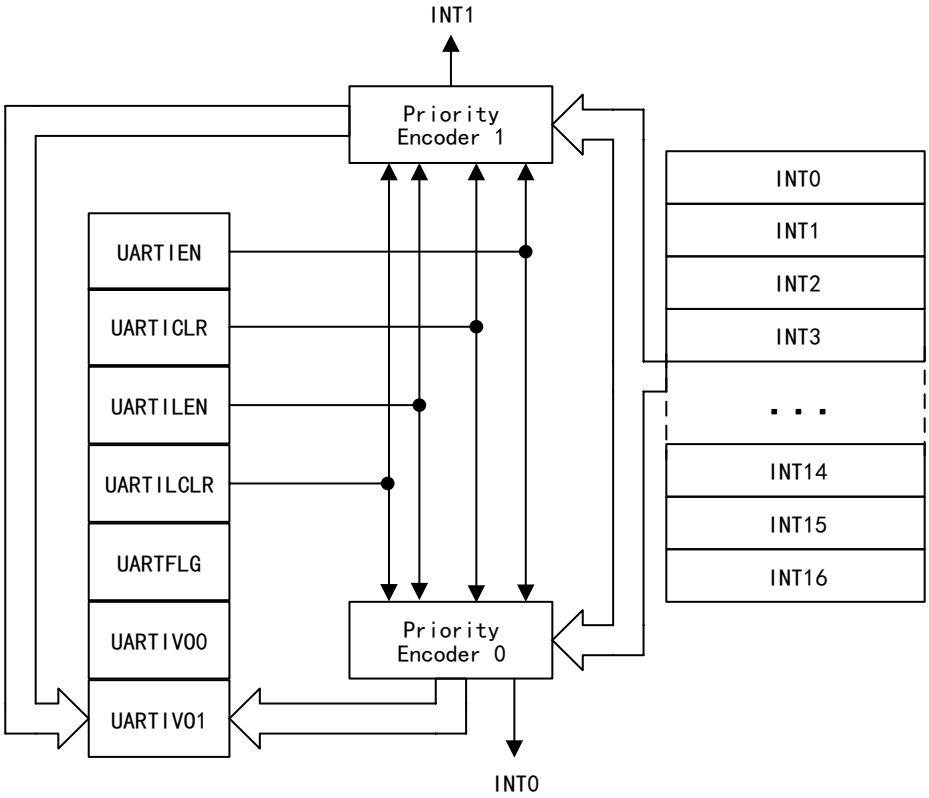
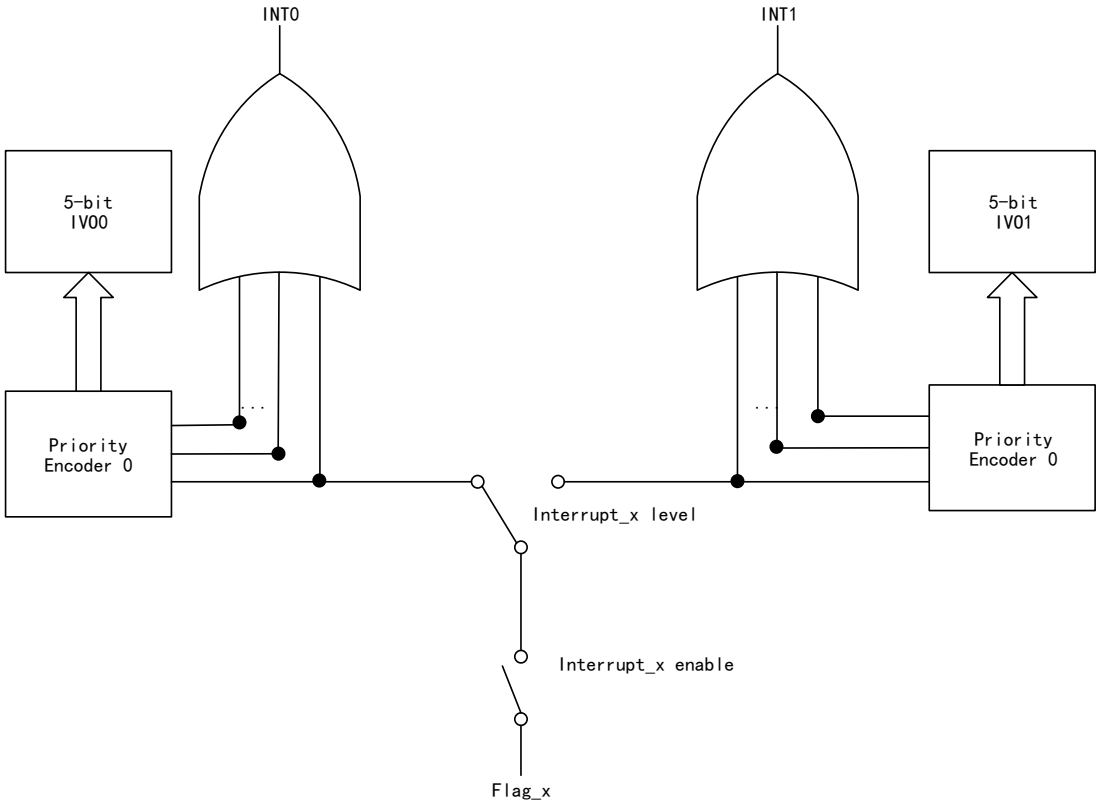


图 284 为特定标志产生中断



LINIVOx 寄存器根据各自的优先级编码器确定触发的中断源。中断向量寄存器返

回挂起中断线的向量。如果有多个挂起的中断，寄存器只保留优先级最高的中断，确保系统能够按照优先级从高到低处理中断。

通过配置 LINIEN 和 LINICLR 寄存器使能和禁用中断。

每个中断的级别默认为 0，可以通用配置 LINILEN 寄存器配置为级别 1，或通过配置 LINILCLR 寄存器配置为级别 0。

发送中断

当 TXIEN=1，TXDREQEN=0 时，才能使用传输中断功能。

在“发送数据”章节中可以了解到，TXIEN 与 TXBRDYFLG 被设置时，在第一次从 TXD 传输到 TXSHF 后才会产生传输中断，即，在生成任何中断之前，必须向 TXD 写入第一个数据，所以，在传输中断服务程序中向 TXD 写入数据，可以传输更多数据。

在使用中断进行传输数据时，TXD 被写入数据时会清除 TXBRDYFLG 位，TXSHF 被写入数据时会再次设置 TXBRDYFLG 位。过程中，可以通过配置 TXICLR 位禁用中断请求，但是当 TXIEN 被使能时，将再次产生中断，TXBRDYFLG 位被设置。

要清除中断请求，可以使用以下方式：

- 清除 TXEN 位
- 软件复位
- 硬件复位

但是当下一个数据被写入 TXD 时，将重新产生中断。

接收中断

当数据从 RXSHF 传输到 RXD 时，会设置 RXBRDYFLG 位，此时 UART 将会读取接收到的数据。设置 RXIEN 位可以使能接收中断，如果 RXBRDYFLG 也被置位，那么就会产生接收中断。

接收的数据可以在 ISR 中读取。

当 UART 和 DMA 同时存在于一个设备时，如果要使用中断，必须清除 SETRXDMA。

唤醒中断

如果 RX 线路上的总线活动阻止进入掉电模式，或者 RX 线路活动导致退出掉电模式，则 UART 设置 WUPFLG 标志。如果启用（WUPIEN），则设置 WUPFLG 标志后触发唤醒中断。

错误中断

以下错误检测是由 UART 模块中断支持的:

- 奇偶校验错误（PEFLG）

- 比特错误（BEFLG）
- 帧错误（FEFLG）
- 溢出错误（OEFLG）
- 中断检测错误（BRKDETFLG）

如果没有检测到错误，那么发送设备和接收设备都会认为信息有效。

在 UART/LIN 模块中共有 16 个中断源，而在 UART 只支持 8 个中断，如下表所示。

表格 241 UART/LIN 中断

偏移量/优先级	中断	适用于 UART	适用于 LIN
0	无中断	-	-
1	唤醒	是	是
2	不同步字段错误	否	是
3	奇偶校验错误	是	是
4	ID	否	是
5	物理总线错误	否	是
6	帧错误	是	是
7	中断信号检测	是	否
8	校验和错误	否	是
9	超速错误	是	是
10	字节错误（BEFLG）	是	是
11	接收	是	是
12	发送	是	是
13	无响应错误	否	是
14	唤醒信号后超时时间（150ms）	否	是
15	3 次唤醒信号后超时时间（1.5 s）	否	是
16	超时（总线空闲，4s）	否	是

表格 242 UART 接收标志位

UART 标志	寄存器	位	复位值
CSEFLG	LINFLG	29	0
ISFEFLG	LINFLG	28	0
NREFLG	LINFLG	27	0
FEFLG	LINFLG	26	0
OEFLG	LINFLG	25	0
PEFLG	LINFLG	24	0

UART 标志	寄存器	位	复位值
RXWUPSEL	LINFLG	12	0
RXRDYFLG	LINFLG	9	0
BUSYFLG	LINFLG	3	0
RXIDLEFLG	LINFLG	2	1
WUPFLG	LINFLG	1	0
BRKDETFLG	LINFLG	0	0

当 RDY = 0 时，这些标志位被冻结，其复位值均为 0。

表格 243 UART 发送标志位

UART 标志	寄存器	位	复位值
BEFLG	LINFLG	31	0
PBEFLG	LINFLG	30	0
TXWUPSEL	LINFLG	10	0
TXEFLG	LINFLG	11	1
TXBRDYFLG	LINFLG	8	1

当 RDY = 0 时，这些标志位被冻结，其复位值均为 0。

41.9.10 DMA

UART 和 LIN 模块都可以使用 DMA 请求（发送和接收），DMA 传输方式取决于是否使用多缓冲模式。

DMA 的请求由 CPU 决定，通过设置 TXDREQEN/TXDREQCLR 位来使能或禁用发送请求；通过设置 RXDREQEN/RXDREQCLR 位来使能或禁用接收请求。

发送 DMA 请求

在多缓冲模式中，发送准备就绪或在传输设定的数据后，就会生成一个 TX DMA 请求，设备会将下一次传输的数据加载到缓冲区。如果禁用了多缓冲模式，那么 DMA 请求将以字节为单位生成。

接收 DMA 请求

在多缓冲模式中，接收器将接收到的字符加载到缓冲区。当设定的最后一个字符被接收且加载到缓冲区时，就会生成一个 RX DMA 请求。如果禁用了多缓冲模式，那么 DMA 请求将以字节为单位生成。

在多处理器模式且禁用了 SLEEP 位的情况下，通过设置 ADRXDREQEN 位，UART 可以对地址帧和数据帧产生 DMA 请求，或者对地址帧产生接收中断，对数据帧产生 DMA 请求。如果设置了 SLEEP 位，那么 UART 就不会为接收到的数据帧产生 DMA 请求。

41.9.11 UART 的低功耗模式

全局低功耗模式，也就是当整个系统处于低功耗模式下，UART/LIN 模块的时钟被关闭，所以这两个模块将不会进行操作。

通过设置 LINCTRL2[LPREQ]位，UART/LIN 模块可以自己进入局部低功耗模式，原理是使时钟源不作用于模块。但是，在局部掉电的情况下，时钟可以通过特定的方式进入 UART，因此所有寄存器都是可以被访问的。

通过清除 LPREQ 位可以退出低功耗模式，当检测到 LINRX 处于低电平时，唤醒中断可以使 UART 自动清除 LPREQ 位。如果唤醒中断被禁用，当请求生成时，UART/LIN 会立即进入低功耗模式，并且 LINRX 引脚上的任何活动都不会使模块退出低功耗模式。

接收器在接收数据的过程中，如果设备生成了进入低功耗模式的请求：

（1） 如果使能了唤醒中断，UART 会立刻产生中断，清除 LPREQ 位，将不会进入低功耗模式，并且会完成接收

（2） 如果禁用了唤醒中断，UART 会在接收完成后，进入低功耗模式

睡眠模式下的多处理器模式

UART 在接收数据的过程中，数据从移位寄存器加载到 LINRXD 中，并且接收准备标志位被置位时会产生接收中断（如果使能了中断），此时 CPU 会在接收完成前读取新的帧。在多处理器通信模式中，这种行为会变得高效，以此提供新数据的选择性指示。

休眠模式适用于空闲线模式和地址位多处理器模式。休眠模式可以让 UART 忽略掉不匹配的地址帧后面的数据，直到匹配到正确的地址帧，才开始接收数据。

睡眠模式中，由于 UART 没有硬件支持，所以必须由软件会读取缓冲区，并比较地址帧与已经在内存中存储的地址来确定 UART 是否被寻址，以下举例不同的确认结果。

如果没有被寻址：

（1） UART 已完成配置，使能睡眠模式和接收

（2） 接收端接收到一个地址帧并产生一个接收中断

（3） 软件将接收到的地址帧与已经在内存中存储的地址进行对比，确定 UART 没有被寻址，SLEEP 位的值不会改变

（4） 数据帧被加载到 UARTRXSHF 进行组装，没有被转移到 LINRXD，也没有接收中断产生

如果确认被寻址：

（1） UART 已完成配置，使能睡眠模式和接收

（2） 接收到一个新的地址帧并产生一个接收中断

（3） 软件将接收到的地址帧与已经在内存中存储的地址进行对比，确定 UART 正在被寻址，清除 SLEEP 位

（4） 转移到 UARTRXSHF 中的数据被加载到 LINRXD 中，每接收到一个数据帧后都会产生一个接收中断

（5） 在每个中断例程中，软件检查 RXWUPSEL 以确定当前帧是否为地址帧

（6） 接收到另一个地址帧，设置 RXWUPSEL，软件确定 UART 没有被寻址，将 SLEEP 位设置为 1。在此地址帧之后的数据帧不会产生接收中断

通过以上事例可以看出，用休眠模式忽略掉无效的数据帧，可以减少中断的产生，这使得可以释放更多的 CPU 资源。

当 UART 处于休眠状态时，除了接收准备标志，其他标志位都会保持更新，这样程序能够及时检测到错误并作出纠正措施。UART 在需要时使用轮询的方式查询接收准备标志，一旦被置位，软件会接收到新的地址并对比，如果 UART 没有被寻址，软件会继续轮询，直到 UART 被寻址，软件改变 SLEEP 位，然后接收数据。

41.10 寄存器组地址

表格 244 LIN 寄存器组地址

设备寄存器	寄存器组	起始地址	结束地址
LINA	LIN_REGS	0x5000 0000	0x5000 03FF

41.11 寄存器地址映射

表格 245 LIN_REGS 寄存器地址映射

寄存器名称	寄存器描述	偏移地址	WRPRT
LINGCTRL0	全局控制寄存器 0	0x00	-
LINGCTRL1	全局控制寄存器 1	0x04	-
LINGCTRL2	全局控制寄存器 2	0x08	-
LINIEN	使能中断寄存器	0x0C	-
LINICLR	清除中断寄存器	0x10	-
LINILEN	使能中断等级寄存器	0x14	-
LINILCLR	清除中断等级寄存器	0x18	-
LINFLG	标志寄存器	0x1C	-
LINIVO0	中断向量偏移寄存器 0	0x20	-
LINIVO1	中断向量偏移寄存器 1	0x24	-
LINLCFG	配置长度寄存器	0x28	-
LINBR	波特率寄存器	0x2C	-

寄存器名称	寄存器描述	偏移地址	WRPRT
LINRXED	接收器仿真数据寄存器	0x30	-
LINRXD	接收数据缓冲寄存器	0x34	-
LINTXD	发送数据缓冲寄存器	0x38	-
LINPCTRL0	控制引脚寄存器 0	0x3C	-
LINPCTRL2	控制引脚寄存器 2	0x44	-
LINCOMP	比较寄存器	0x60	-
LINRXB0	接收缓冲区寄存器 0	0x64	-
LINRXB1	接收缓冲区寄存器 1	0x68	-
LINIDMASK	屏蔽 ID 寄存器	0x6C	-
LINID	ID 寄存器	0x70	-
LINTXB0	发送缓冲区寄存器 0	0x74	-
LINTXB1	发送缓冲区寄存器 1	0x78	-
LINMBRPSC	最大波特率预分频寄存器	0x7C	-
LINEET	模拟错误和测试寄存器	0x90	-
LINGIEN	全局中断寄存器	0xE0	-
LINGIFLG	全局中断标志寄存器	0xE4	-
LINGICLR	清除全局中断标志寄存器	0xE8	-

41.12 寄存器功能描述

41.12.1 全局控制寄存器 0（LINGCTRL0）

偏移地址：0x00

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	RST	R/W	复位 USART/LIN（USART/LIN Reset） 0：禁用 1：使能	0h
31:1	保留			0h

41.12.2 全局控制寄存器 1（LINGCTRL1）

偏移地址：0x04

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	COMCFG	R/W	配置 USART/LIN 通信模式（USART/LIN Communication Mode Configure） 在兼容模式下，它选择 UART 通信模式。在 LIN 模式下，它选择 ID 字段位 ID4 和 ID5 的长度控制选项。 0：UART 兼容模式：使用空闲线模式。 LIN 模式：ID4 和 ID5 不用于长度控制。 1：UART 兼容模式：使用地址位模式。 LIN 模式：ID4 和 ID5 用于长度控制。	0h
1	TIM	R/W	USART 定时模式（USART timing mode） 这个位仅在 UART 兼容模式下有效。在使用 UART 模式时，必须将这个位设置为 1。这个位配置 UART 进行异步操作。 0：保留。 1：当模块配置为 UART 操作时，必须设置为 1。	0h
2	PARITYEN	R/W	使能 USART/LIN 校验（USART/LIN Parity Enable） 使能或禁用奇偶校验功能。 0：UART 兼容模式：禁用校验位，在传输过程中不生成校验位，接收过程中也不期望校验位。 LIN 模式：ID 校验位验证被禁用。 1：UART 兼容模式：使能校验位，传输过程中生成校验位，在接收过程中期望校验位。 LIN 模式：使能 ID 校验位验证。	0h
3	PARITY	R/W	选择 USART 奇偶校验位（USART Odd/Even Parity Select） 这个位仅在 UART 兼容模式下有效（写入）。如果 PARITY ENA 位（UARTGCR1.2）被设置，PARITY 指定奇校验或偶校验。校验位是根据每帧中的数据位和地址位（在地址位模式下）计算得出的。帧中的起始位和停止位不包括在校验计算中。 0：使用奇校验。UART 发送和期望接收的校验位值使得帧中值为 1 的总位数为奇数。 1：使用偶校验。UART 发送和期望接收的校验位值使得帧中值为 1 的总位数为偶数。	0h
4	STOPCFG	R/W	配置 USART 停止位（USART STOP Bit Configure） 仅在 UART 兼容模式下有效（写入）。 注意：接收器仅检查一个停止位。然而，在空闲线模式下，如果 STOP = 1，接收器会等到第二个停止位结束后开始检查空闲时间。 0：使用一个停止位。 1：使用两个停止位。	0h
5	CLK_MSCFG	R/W	使能 USART 内部时钟/配置 LIN 主从模式（USART internal clock enable/LIN Master Slave configure） 在 UART 模式下，这个位启用 UART 模块的时钟。在 LIN 模式下，这个位确定 LIN 节点是从节点还是主节点。 0：UART 兼容模式：保留。 LIN 模式：模块处于从节点模式。 1：UART 兼容模式：启用 UART 模块的时钟。 LIN 模式：节点处于主节点模式。	0h

位/域	名称	R/W	描述	复位值
6	LINEN	R/W	使能 LIN 模式（LIN mode Enable） 如果使能 LIN 模式，则禁用 UART 兼容模式；反之，则使能 UART 兼容模式。 0：使能 LIN 模式 1：禁止 LIN 模式	0h
7	RDY	R/W	USART/LIN 就绪（USART/LIN Ready） 这个位在 LIN 或兼容 UART 模式下有效。在 RDY = 0 时才能配置 UART/LIN。 只有以下配置位可以在运行时更改（即在 RDY = 1 时）： 停止扩展帧（LINGCR1[13]） COMPCHASUM 位 SC 位 0：UART/LIN 处于复位状态，不会传输或接收数据。将此位写为 0 会初始化 UART/LIN 状态机和操作标志。所有受影响的逻辑保持在复位状态，直到向此位写入 1。 1：UART/LIN 处于准备状态，可以进行传输和接收。将此位设置为 1 后，模块的配置不应更改。	0h
8	SLEEPEN	R/W	使能 USART 睡眠模式（USART Sleep mode enable） 仅在 UART 兼容模式下有效（可写）。在多处理器配置中，这个位控制接收休眠功能。清除此位将使 UART 退出休眠模式。 当 SLEEP 位被设置时，接收器仍在操作，但只有在检测到地址帧时，RXRDYFLG 才会更新，LINRXD 才会加载新数据。其余接收器状态标志将被更新，并且如果对应的中断使能位被设置，将请求错误中断，不论 SLEEP 位的值如何。通过这种方式，如果在 UART 处于休眠状态时检测到接收数据线上的错误，软件可以及时处理错误情况。当检测到地址字节时，SLEEP 位不会自动清除。 0：禁止 1：使能	0h
9	ABEN	R/W	使能 LIN 波特率自动调节（ LIN Automatic baudrate adjustment enable） 这个模式只在 LIN 模式下有效。 这个位在检测同步字段时起作用。根据节点能力文件定义，可以通过这个位启用两种 LIN 协议比特率模式：自动或选择。软件和网络配置将决定前两种模式中的哪种。当清除这个位时，应使用 LIN 2.0 协议的固定比特率。如果设置了 ADAPT 位，检测波特率的 LIN 从节点将比较其与 BRSR 寄存器中的分频器，并在它们不同时更新它。BRSR 寄存器将使用新值进行更新。如果未设置此位，则不会调整 BRSR 寄存器。这个字段仅在 LIN 模式下可写。 0：禁止 1：使能	0h
10	MBUFEN	R/W	使能 USART/LIN 多缓冲区模式（USART/LIN multi-buffer mode enable） 这个位在 LIN 或兼容 UART 模式下有效。此位控制接收/发送缓冲区的使用，即是使用 RX/TX 多缓冲区还是单个寄存器 RD0/TXB0。 0：禁止 1：使能	0h

位/域	名称	R/W	描述	复位值
11	CHASUMCFG	R/W	配置 LIN 校验和（LIN Checksum Configure） 这个位只在 LIN 模式下有效。此位控制要使用的校验和类型：经典或增强。 0：使用经典校验和。 这种校验和与 LIN 1.3 从节点兼容。经典校验和包含对所有数据字节进行累加取模 256 的结果。使用标识符 60（0x3C）到 63（0x3F）发送的帧必须始终使用经典校验和。 1：使用增强校验和。 增强校验和与 LIN 2.0 及更新的从节点兼容。增强校验和包含对所有数据字节和受保护的标识符进行累加取模 256 的结果。	0h
12	HGENCTRL	R/W	控制 LIN 的 HGEN（LIN HGEN control） 这个位只在 LIN 模式下有效。此位控制掩码过滤比较的类型。 0：使用 ID-Byte 进行 ID 过滤。 LINID 寄存器中的 RECEIVEDID 和 IDBYTE 字段用于检测匹配（使用 TX/RXMASK 值）。LINMASK 寄存器中掩码为 0xFF 将导致无匹配。 1h（读/写）= 使用 ID-SLAVETask 字节进行 ID 过滤（推荐）。 LINID 寄存器中的 RECEIVEDID 和 IDSLAVETASKBYTE 字段用于检测匹配（使用 TX/RXMASK 值）。LINMASK 寄存器中掩码为 0xFF 将导致始终匹配。	0h
13	STOEXTCOM	R/W	停止 LIN 扩展帧通信（Stop LIN extended frame communication） 这个位只在 LIN 模式下有效。这个位只能在扩展帧通信期间写入。当扩展帧通信停止时，此位会自动清除。 0：无效果 1：一旦当前帧的传输/接收完成，扩展帧通信将被停止。	0h
15:14	保留			0h
16	LBEN	R/W	使能 USART/LIN 回环模式（USART/LIN Loopback Mode Enable） 这个位在 LIN 或 UART 兼容模式下有效。UART/LIN 的自检选项可以通过这个位选择。如果 LINTX 和 LINRX 引脚配置了 UART/LIN 功能，则 LINTX 引脚在内部连接到 LINRX 引脚。在外部，在环回操作期间，LINTX 引脚输出高电平，LINRX 引脚处于高阻态。如果在 UART/LIN 正在传输或接收数据时改变此位的值，可能会导致错误。 0：禁用 1：使能	0h

位/域	名称	R/W	描述	复位值
17	CONTSUS	R/W	USART/LIN 挂起后继续工作（USART/LIN Continue work on suspend） 这个位只在使用仿真器调试程序时才会起作用，它决定了在程序暂停时 UART/LIN 的运行方式。这个位会影响 LIN 计数器。当设置了这个位时，在调试过程中计数器不会停止。当清除这个位时，在调试过程中计数器会停止。 0：进入调试模式时，UART/LIN 状态机被冻结。传输和 LIN 计数器被停止，当退出调试模式时恢复。 1：进入调试模式时，UART/LIN 会继续运行直到当前的发送和接收功能完成。	0h
23:18	保留			0h
24	RXEN	R/W	使能 USART/LIN 接收（USART/LIN Receive enable） 此位控制允许接收器将数据从移位缓冲区传输到接收缓冲区或多缓冲区。在 LIN 或 UART 兼容模式下有效。 注意： （1）清除 RXENA 会阻止接收到的字符被传输到接收缓冲区或多缓冲区，防止接收数据更新 RX 状态标志（见表 7），并禁止接收和错误中断。但是，移位寄存器会继续组装数据，而不受 RXENA 状态的影响。 （2）如果在接收帧完成之前清除 RXENA，则帧中的数据不会被传输到接收缓冲区。 （3）如果在接收帧完成之前设置了 RXENA，则帧中的数据将被传输到接收缓冲区。如果在 UARTRXSHF 正在组装帧的过程中设置 RXENA，则不能保证状态标志对于该帧是准确的。为确保状态标志正确反映在特定帧期间总线上检测到的内容，应在检测到该帧之前设置 RXENA。 复位类型：SYSRSn 0：不允许 1：允许	0h
25	TXEN	R/W	使能 USART/LIN 发送（USART/LIN Transmit enable） 位域在 LIN 和 UART 模式下有效。 只有在 TXENA 位被设置时，数据才会从 LIN 模式下的 LINTXD 或 TXBy（其中 y=0, 1,...7）缓冲区传输到 UARTTXSHF 移位寄存器中。注意：在设置 TXENA 之前写入的数据不会被传输到 LINTXD 或传输多缓冲区。如果在传输过程中清除 TXENA，则之前写入 LINTXD 的数据将被发送（包括 LIN 模式下的校验和字节）。 复位类型：SYSRSn 0：禁用从 LINTXD 或 TXBy 到 UARTTXSHF 的传输 1：使能从 LINTXD 或 TXBy 到 UARTTXSHF 的数据传输	0h
31:26	保留			0h

41.12.3 全局控制寄存器 2（LINGCTRL2）

偏移地址：0x08

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	LPREQ	R/W	请求低功耗模式（ low-power mode Request） 这个位在 LIN 或兼容 UART 模式下有效。当设置了电源关闭位时，UART/LIN 模块会尝试进入本地低功耗模式。如果在接收器正在主动接收数据且唤醒中断被禁用时设置了 LPREQ 位，则 UART/LIN 将延迟进入低功耗模式，直到接收完成。在 LIN 模式下，用户可以在接收到休眠命令或检测到总线空闲（超过 4 秒，即 20kHz 下的 80,000 个周期）时设置 LPREQ 位。 0：正常操作 1：请求本地低功耗模式	0h
7:1	保留			0h
8	WUPGEN	R/W	产生唤醒信号（Generate wakeup signal） 这个位控制通过发送 TXB0 缓冲区值生成唤醒信号。在接收到有效的同步中断时，这个位将被清除。 0：无效 1：用于唤醒发送 TXB0。这个位将在 RDY 时被清除。	0h
15:9	保留			0h
16	TXCHAEN	R/W	使能发送校验字节（Transmit Checksum Byte Enable） 这个模式只在 LIN 模式下有效（写入）。这个位由传输器与扩展帧一起用来发送一个校验字节。在非多缓冲区模式下，校验字节将在当前字节传输后发送。在多缓冲区模式下，校验字节将在最后一个字节计数（由 LINLCFG[18:16]指示）后发送。 0：不发送校验字节。 1：将发送一个校验字节。在发送校验字节后，此位将自动清除。如果在发送第一个字节之前设置了这个位（即在帧间空隙期间），则不会发送校验和。	0h
17	COMPCHASUM	R/W	比较校验和（Compare Checksum） 这个模式只在 LIN 模式下有效。接收器用于在扩展帧中触发校验和比较的位。用户将通过向这个位写入 1 来启动此事务。 在非多缓冲区模式下，一旦设置了 COMPCHASUM 位，将在当前正在接收的字节上比较校验和，预期是校验字节。 在多缓冲区模式下，以下是与 COMPCHASUM 位相关的情况： 如果在接收数据时设置了 COMPCHASUM 位，则在接收由 LINLCFG[18:16]指示的编程数据字节数后接收的字节将被视为校验和字节。 如果在空闲期间（即在帧间间隔期间）设置了 COMPCHASUM 位，则下一个立即字节将被视为校验和字节。 如果存在校验和错误，则将立即标记为 CSEFLG。 一旦成功比较了校验和，这个位将被自动清除。 0：无效 1：在预期的校验字节上比较校验和	0h
31:18	保留			0h

41.12.4 使能中断寄存器（LINIEN）

偏移地址：0x0C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	BRKDETIEN	R/W	使能断开检测中断（break detect interrupt Enable） 这个位仅在 UART 兼容模式下有效（写入）。设置此位使 UART/LIN 在 LINRX 引脚上检测到中断条件时生成中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
1	WUPIEN	R/W	使能唤醒中断（wake up interrupt Enable） 这个位在 LIN 或 UART 兼容模式下有效。设置此位使 UART/LIN 生成唤醒中断，从而退出低功耗模式。唤醒中断在唤醒脉冲的下降沿被断言。如果启用，当在接收器忙碌时请求本地低功耗模式或在低功耗模式期间检测到 LINRX 引脚上的低电平时，唤醒中断将被断言。如果模块不处于掉电模式，则在唤醒脉冲时不会断言唤醒中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
3:2	保留			0h
4	TOIEN	R/W	使能超时中断（timeout interrupt Enable） 这个位仅在 LIN 模式下有效（写入）。设置此位使 UART/LIN 在至少 4 秒内没有 LIN 总线活动（总线空闲）时生成中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
5	保留			0h
6	TOWUPSIEN	R/W	使能唤醒信号后超时中断（Timeout After Wakeup Signal interrupt Enable） 这个位仅在 LIN 模式下有效（写入）。设置此位使 UART/LIN 在发送了一个唤醒信号后发生超时时生成中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
7	TO3WUPSIEN	R/W	使能 3 个唤醒信号后超时中断（Timeout After 3 Wakeup Signal interrupt Enable） 这个位仅在 LIN 模式下有效（写入）。设置此位使 UART/LIN 在发送了 3 个唤醒信号后发生超时时生成中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
8	TXIEN	R/W	使能发送器中断（Transmitter interrupt Enable） 设置此位使 UART/LIN 能够在数据从 LINTXD 传输到 UARTTXSHF 并且 TXBRDYFLG 位被设置时生成发送中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h

位/域	名称	R/W	描述	复位值
9	RXIEN	R/W	使能接收器中断（Receiver interrupt Enable） 设置此位使 UART/LIN 能够在完全接收一个帧并将数据从 UARTRXSHF 传输到 UARTRXD 时生成接收中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
12:10	保留			0h
13	IDIEN	R/W	使能识别中断（Identification interrupt Enable） 这个位仅在 LIN 模式下有效。设置此位后，在接收到有效匹配标识符时发生中断。 0：禁止。将此位写为 0 不会产生任何效果。 1：使能	0h
15:14	保留			0h
16	TXDREQEN	R/W	使能发送 DMA 请求（ transmit DMA request enable） 这个位在 LIN 或 UART 兼容模式下有效。要使能发送器的 DMA 请求，必须设置该位。如果清除该位，则根据 SETTXINT 设置会生成中断请求。 0：发送器 DMA 请求被禁用。将此位写为 0 不会产生任何效果。 1：使能发送器 DMA 请求。	0h
17	RXDREQEN	R/W	使能接收 DMA 请求（ Receive DMA request enable） 这个位在 LIN 或 UART 兼容模式下有效。要使能接收器的 DMA 请求，必须设置该位。如果清除该位，则根据 SETRXINT 设置会生成中断请求。 0：接收器 DMA 请求被禁用。将此位写为 0 不会产生任何效果。 1：使能接收器 DMA 请求。	0h
18	ADRXDREQEN	R/W	为地址和数据帧使能接收器 DMA 请求（Address & Data frames Receive DMA request enable） 设置接收器 DMA 用于地址帧和数据帧。这个位在 LIN 或兼容 UART 模式下有效。要启用地址帧和数据帧的 RX DMA 请求，必须设置该位。如果清除该位，则对于地址帧会生成 RX 中断请求，对于数据帧会生成 DMA 请求。 0：对于地址帧，接收器 DMA 请求被禁用（对于地址帧启用 RX 中断请求）。将此位写为 0 不会产生任何效果。 1：对于地址和数据帧，使能接收器 DMA 请求	0h
23:19	保留			0h
24	PIEN	R/W	使能奇偶校验中断（parity interrupt Enable） 这个位在 LIN 模式或 UART 模式下有效。 使能后，UART/LIN 模块在出现奇偶校验错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h

位/域	名称	R/W	描述	复位值
25	OREIEN	R/W	使能过载错误中断（overrun error interrupt Enable） 这个位在 LIN 模式或 UART 模式下有效。 使能后，UART/LIN 模块在出现过载错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
26	FEIEN	R/W	使能帧错误中断（framing error interrupt Enable） 这个位在 LIN 模式或 UART 模式下有效。 使能后，UART/LIN 模块在出现帧错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
27	NREIEN	R/W	使能无响应错误中断（no response error interrupt Enable） 这个位只在 LIN 模式下有效（写入）。 使能后，UART/LIN 模块在出现无响应错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
28	ISFEIEN	R/W	使能不一致的同步字段错误中断（inconsistent sync field error interrupt Enable） 这个位只在 LIN 模式下有效（写入）。 使能后，UART/LIN 模块在出现不一致的同步字段错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
29	CSEIEN	R/W	使能校验和错误中断（checksum-error Interrupt Enable） 这个位只在 LIN 模式下有效（写入）。 使能后，UART/LIN 模块在出现校验和错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
30	PBEIEN	R/W	使能物理总线错误中断（ physical bus error interrupt Enable） 这个位只在 LIN 模式下有效（写入）。 使能后，UART/LIN 模块在出现物理总线错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h
31	BEIEN	R/W	使能位错误中断（bit error interrupt Enable） 这个位只在 LIN 模式下有效（写入）。 使能后，UART/LIN 模块在出现位错误时生成中断。 0：禁用。将此位写为 0 不会产生任何效果。 1：使能	0h

41.12.5 清除中断寄存器（LINICLR）

偏移地址：0x10

复位类型：SYSRSn

此寄存器用于清除 LINIEN 寄存器中提到的相应中断或请求（除了“为地址和数据帧使能接收器 DMA 请求”），设置相应位为 1 进行清除。

位的有效模式和写入条件（仅在 LIN 模式下或 LIN/UART 模式下）与 LINIEN 寄存器一致。

所有位写 0 无效，写 1 置位。

位/域	名称	R/W	描述	复位值
0	BRKDETICLR	RC_W1	清除断开检测中断（break detect interrupt Clear）	0h
1	WUPICLR	RC_W1	清除唤醒中断（wake up interrupt Clear）	0h
3:2	保留			0h
4	TOICLR	RC_W1	清除超时中断（ timeout interrupt Clear）	0h
5	保留			0h
6	TOWUPSICLR	RC_W1	清除唤醒信号后超时中断（Timeout After Wakeup Signal interrupt Clear）	0h
7	TO3WUPSICLR	RC_W1	清除 3 个唤醒信号后超时中断（Timeout After 3 Wakeup Signal interrupt Clear）	0h
8	TXICLR	RC_W1	清除发送器中断（Transmitter interrupt Clear）	0h
9	RXICLR	RC_W1	清除接收器中断（Receiver interrupt Clear）	0h
12:10	保留			0h
13	IDICLR	RC_W1	清除识别中断（Identification interrupt Clear）	0h
15:14	保留			0h
16	TXDREQCLR	RC_W1	清除发送 DMA 请求（ transmit DMA request Clear）	0h
17	RXDREQCLR	RC_W1	清除接收 DMA 请求（ Receive DMA request Clear）	0h
23:18	保留			0h
24	PICLR	RC_W1	清除奇偶校验中断（parity interrupt Clear）	0h
25	OREICLR	RC_W1	清除过载错误中断（overrun error interrupt Clear）	0h
26	FEICLR	RC_W1	清除帧错误中断（framing error interrupt Clear）	0h
27	NREICLR	RC_W1	清除无响应错误中断（no response error interrupt Clear）	0h
28	ISFEICLR	RC_W1	清除不一致的同步字段错误中断（inconsistent sync field error interrupt Clear）	0h
29	CSEICLR	RC_W1	清除校验和错误中断（checksum-error Interrupt Clear）	0h
30	PBEICLR	RC_W1	清除物理总线错误中断（ physical bus error interrupt Clear）	0h
31	BEICLR	RC_W1	清除位错误中断（bit error interrupt Clear）	0h

41.12.6 使能中断等级寄存器（LINILEN）

偏移地址：0x14

复位类型：SYSRSn

此寄存器用于设置 LINIEN 寄存器中提到的相应中断或请求的等级，写 0 表示设置为 INT0，写 1 表示设置为 INT1。

位的有效模式和写入条件（仅在 LIN 模式下或 LIN/UART 模式下）与 LINIEN 寄存器一致。

位/域	名称	R/W	描述	复位值
0	BRKDETILEN	R/W	使能断开检测中断级别 1（break detect interrupt level 1 Enable）	0h
1	WUPILEN	R/W	使能唤醒中断级别 1（wake up interrupt level 1 Enable）	0h
3:2	保留			0h
4	TOILEN	R/W	使能超时中断级别 1（ timeout interrupt level 1 Enable）	0h
5	保留			0h
6	TOWUPSILEN	R/W	使能唤醒信号后超时中断级别 1（Timeout After Wakeup Signal interrupt level 1 Enable）	0h
7	TO3WUPSILEN	R/W	使能 3 个唤醒信号后超时中断级别 1（Timeout After 3 Wakeup Signal interrupt level 1 Enable）	0h
8	TXILEN	R/W	使能发送器中断级别 1（Transmitter interrupt level 1 Enable）	0h
9	RXILEN	R/W	使能接收器中断级别 1（Receiver interrupt level 1 Enable）	0h
12:10	保留			0h
13	IDILEN	R/W	使能识别中断级别 1（Identification interrupt level 1 Enable）	0h
23:14	保留			0h
24	PILEN	R/W	使能奇偶校验中断级别 1（parity interrupt level 1 Enable）	0h
25	OREILEN	R/W	使能过载错误中断级别 1（overrun error interrupt level 1 Enable）	0h
26	FEILEN	R/W	使能帧错误中断级别 1（framing error interrupt level 1 Enable）	0h
27	NREILEN	R/W	使能无响应错误中断级别 1（no response error interrupt level 1 Enable）	0h
28	ISFEILEN	R/W	使能不一致的同步字段错误中断级别 1（inconsistent sync field error interrupt level 1 Enable）	0h
29	CSEILEN	R/W	使能校验和错误中断级别 1（checksum-error Interrupt level 1 Enable）	0h
30	PBEILEN	R/W	使能物理总线错误中断级别 1（ physical bus error interrupt level 1 Enable）	0h
31	BEILEN	R/W	使能位错误中断级别 1（bit error interrupt level 1 Enable）	0h

41.12.7 清除中断等级寄存器（LINILCLR）

偏移地址：0x18

复位类型：SYSRSn

此寄存器用于清除 LINILEN 寄存器中提到的相应中断或请求的等级，写 0 表示设置为 INT0，写 1 表示设置为 INT0 并清除该位。

位的有效模式和写入条件（仅在 LIN 模式下或 LIN/UART 模式下）与 LINILEN 寄存器一致。

位/域	名称	R/W	描述	复位值
0	BRKDETILCLR	RC_W1	清除断开检测中断级别 1（break detect interrupt level 1 Clear）	0h
1	WUPILCLR	RC_W1	清除唤醒中断级别 1（wake up interrupt level 1 Clear）	0h
3:2	保留			0h
4	TOILCLR	RC_W1	清除超时中断级别 1（ timeout interrupt level 1 Clear）	0h
5	保留			0h
6	TOWUPSILCLR	RC_W1	清除唤醒信号后超时中断级别 1（Timeout After Wakeup Signal interrupt level 1 Clear）	0h
7	TO3WUPSILCLR	RC_W1	清除 3 个唤醒信号后超时中断级别 1（Timeout After 3 Wakeup Signal interrupt level 1 Clear）	0h
8	TXILCLR	RC_W1	清除发送器中断级别 1（Transmitter interrupt level 1 Clear）	0h
9	RXILCLR	RC_W1	清除接收器中断级别 1（Receiver interrupt level 1 Clear）	0h
12:10	保留			0h
13	IDILCLR	RC_W1	清除识别中断级别 1（Identification interrupt level 1 Clear）	0h
23:14	保留			0h
24	PILCLR	RC_W1	清除奇偶校验中断级别 1（parity interrupt level 1 Clear）	0h
25	OREILCLR	RC_W1	清除过载错误中断级别 1（overrun error interrupt level 1 Clear）	0h
26	FEILCLR	RC_W1	清除帧错误中断级别 1（framing error interrupt level 1 Clear）	0h
27	NREILCLR	RC_W1	清除无响应错误中断级别 1（no response error interrupt level 1 Clear）	0h
28	ISFEILCLR	RC_W1	清除不一致的同步字段错误中断级别 1（inconsistent sync field error interrupt level 1 Clear）	0h
29	CSEILCLR	RC_W1	清除校验和错误中断级别 1（checksum-error Interrupt level 1 Clear）	0h
30	PBEILCLR	RC_W1	清除物理总线错误中断级别 1（ physical bus error interrupt level 1 Clear）	0h
31	BEILCLR	RC_W1	清除位错误中断级别 1（bit error interrupt level 1 Clear）	0h

41.12.8 标志寄存器（LINFLG）

偏移地址：0x1C

复位类型：SYSRSn

需要注意，不同的位域在不同的模式下有效，仅在特定的模式下才可写。

位/域	名称	R/W	描述	复位值
0	BRKDETFLG	RC_W1	断开检测标志（break detect flag） 此位仅在 UART 兼容模式下有效。 当 UART 在 LINRX 引脚上检测到断开条件时，此位被设置。断开条件发生在 LINRX 引脚在缺失第一个停止位后（即帧错误后）连续保持低电平至少 10 个位时。如果 BRKDETIEN 位被设置，检测到断开条件会导致 UART 生成一个错误中断。BRKDETIEN 位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：未检测到断开条件 1：检测到断开条件	0h
1	WUPFLG	RC_W1	唤醒标志（Wake up flag） 此位仅在 LIN 模式下有效（写入）。 当接收器或发送器活动使模块退出降功耗模式时，此位由 UART/LIN 设置。如果 WUPIEN 位被设置，会生成一个中断。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：不从降功耗模式唤醒。 1：从降功耗模式唤醒。	0h
2	RXIDLEFLG	R	接收器空闲标志（receiver idle flag） 此位仅在 UART 兼容模式下有效（写入）。 在此位被设置时，UART 会寻找一个空闲期以重新同步自身与位流。当此位被设置时，接收器不会接收任何数据。总线必须空闲 11 个位周期才能清除此位。 UART 进入此状态的情况有： ● 系统复位后 ● 设置 RDY 位 ● 从降功耗模式恢复后 0：检测到空闲期，UART 准备接收 1：未检测到空闲期，UART 将不会接收任何数据	1h

位/域	名称	R/W	描述	复位值
3	BUSYFLG	R	总线忙碌标志（Bus BUSY flag） 此位在 LIN 模式和 UART 兼容模式下有效（写入）。 此位指示接收器是否正在接收帧。当接收器检测到起始位的开始时，此位被设置为 1。当帧接收完成时，BUSY 位被清除。如果 WUPIEN 被设置且在此位被设置时请求降功耗模式，UART/LIN 会自动防止进入低功耗模式并生成唤醒中断。BUSY 位由 UART 接收器直接控制，但可以通过以下方式清除： ● 设置 RDY 位 ● RST 位 ● 系统复位 0：接收器当前未接收帧 1：接收器当前正在接收帧	0h
4	LINIDLEFLG	RC_W1	LIN 总线空闲标志（ LIN bus idle flag） 此位仅在 LIN 模式下有效（写入）。 如果在至少 4 秒内没有 LIN 总线活动，此位将被设置。LIN 总线活动指从隐性到显性的转换。 此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：未检测到总线空闲 1：检测到 LIN 总线空闲	0h
5	保留			0h
6	TOWUPSFLG	RC_W1	唤醒信号后超时标志（Timeout After Wakeup Signal flag） 此位仅在 LIN 模式下有效（写入）。 如果在发送唤醒信号后未接收到同步中断，此位将被设置。在发出另一唤醒信号之前，使用至少 150 毫秒的超时时间。 此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：一个唤醒信号（150 ms）后未超时 1：一个唤醒信号后超时	0h

位/域	名称	R/W	描述	复位值
7	TO3WUPSFLG	RC_W1	3 个唤醒信号后超时标志（Timeout After 3 Wakeup Signal flag） 此位仅在 LIN 模式下有效（写入）。 如果在发送 3 个唤醒信号后和经过 1.5 秒的时间未接收到同步中断，此标志将被设置。这样的超时时间在发出另一轮唤醒信号之前使用。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：3 个唤醒信号后未超时 1：3 个唤醒信号和 1.5 秒后超时	0h
8	TXBRDYFLG	R	发送器缓冲区就绪标志（Transmitter buffer ready flag） 当设置此位时，表示发送缓冲寄存器（兼容模式下为 LINTXD，MBUF 模式下为 LINTXB0、LINTXB1）已准备好从 CPU 写入另一个字符。 在兼容模式下，向 LINTXD 写入数据会自动清除此位。在 LIN 模式下，一旦字节 0（TXB0）被写入 LINTXB0，此位将被清除。当 TX 缓冲区的数据被移入 UARTTXSHF 寄存器后，此位将被设置。如果 DMA 使能位被设置，此事件可以触发发送 DMA 事件。此位被以下方式设置为 1： ● RST 位 ● 设置 RDY ● 系统复位 注意：通过读取 LINIVO0/1 寄存器中对应的中断偏移不能清除 TXRDY 标志。 注意：通过在 LINICLR 寄存器中禁用相应的中断或通过 TXENA 位禁用发送器，可以消除发送中断请求，直到写入下一系列数据到发送缓冲区 LINTXB0 和 LINTXB1。 0： 兼容模式：LINTXD 已满。 LIN 模式：多缓冲区已满。 1： 兼容模式：LINTXD 已准备好接收下一个字符。 LIN 模式：多缓冲区已准备好接收下一个字符。	1h

位/域	名称	R/W	描述	复位值
9	RXRDYFLG	RC_W1	接收器就绪标志（Receiver ready flag） 在 UART 兼容模式下，接收器设置此位以指示 LINRXD 包含新数据并准备好由 CPU 读取。在 LIN 模式下，一旦在多缓冲模式中接收到有效帧，RXRDYFLG 将被设置，有效帧是指没有错误的消息帧。在非多缓冲模式下，每接收到一个字节 RXRDYFLG 将被设置，如果没有错误，将为帧的最后一个字节设置此位。如果中断使能位被设置，当 RXRDYFLG 标志位被设置时，UART/LIN 将生成接收中断。RXRDYFLG 可以通过以下方式清除： ● RST 位 ● 设置 RDY ● 系统复位 ● 向此位写入 1 ● 在 UART 兼容模式下读取 LINRXD ● 在 LIN 模式下读取响应的最后一个数据字节 RXBy 注意：通过读取 LINIVO0/1 寄存器中对应的中断偏移不能清除 RXRDYFLG 标志。 0：LINRXD/LINRXBy 中没有新数据。 1：新数据已准备好从 LINRXD 读取。	0h
10	TXWUPSEL	R/W	选择发送器唤醒方式（transmitter wakeup mode select） 此位仅在 UART 兼容模式下有效。 TXWUPSEL 位控制 LINTXD 中的数据是否应使用多处理器通信格式作为地址或数据帧发送。此位在字节写入 LINTXD 之前由软件设置为 1 或 0，当数据从 LINTXD 传输到 UARTTXSHF 或系统复位时由 UART 清除。TXWUPSEL 不会被发送准备位清除。 0：地址位模式：要传输的帧将是数据（地址位=0）。 空闲线模式：要传输的帧将是数据。 1：地址位模式：要传输的帧将是地址（地址位=1）。 空闲线模式：接下来的帧将是地址（将此位写为 1 然后将虚拟数据写入 LINRXD，将导致在下一个帧传输前有 11 个位周期的空闲期）。	0h

位/域	名称	R/W	描述	复位值
11	TXEFLG	R	发送器空标志（Transmitter Empty flag） 该标志的值指示发送器的缓冲寄存器（LINTXD/TXBy）和移位寄存器（UARTTXSHF）的内容。在多缓冲模式下，该标志指示 TXBx 寄存器和移位寄存器（UARTTXSHF）的值。在非多缓冲模式下，该标志指示 LINTXB0（字节）和移位寄存器（UARTTXSHF）的值。 此位可以通过以下方式设置： ● RST 位 ● 设置 RDY 位 ● 系统复位 注意：此位不会引起中断请求。 0： 兼容模式或无多缓冲的 LIN 模式： 发送缓冲器或移位寄存器（或两者）已加载数据。 在使用多缓冲模式的 LIN 模式下： 多缓冲器或移位寄存器（或所有）已加载数据。 1： 兼容模式或无多缓冲的 LIN 模式： 发送缓冲器和移位寄存器均为空。 在使用多缓冲模式的 LIN 模式下： 多缓冲器和移位寄存器均为空。	1h
12	RXWUPSEL	R	选择接收器唤醒方式（Receiver wakeup mode select） 此位仅在 UART 兼容模式下有效（写入）。 UART 设置此位以指示 LINRXD 中的当前数据是一个地址。 此位可以通过以下方式清除： ● RST 位 ● 设置 RDY 位 ● 系统复位 ● 接收数据帧 0：LINRXD 中的数据不是地址 1：LINRXD 中的数据是地址 有关更多关于在睡眠模式中使用 RXWUPSEL 位的信息，请参见“多处理器通信的睡眠模式”。	0h

位/域	名称	R/W	描述	复位值
13	TXIDFLG	RC_W1	发送识别符标志（Transmit Identifier Flag） 此位仅在 LIN 模式下有效（写入）。 当接收到一个具有 TX 匹配且无 ID 奇偶校验错误的标识符时，此标志会被设置。详情请参见“消息过滤和验证”部分。当此标志被设置时，表示在 TX 匹配上接收到一个新的有效标识符。 此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● RST 位 ● 设置 RDY ● 系统复位 ● 读取 LINID 寄存器 ● 向此位写入 1 0：未接收到有效 ID。 1：在 TX 匹配上接收到的 LINID[23:16]中包含有效 ID。	0h
14	RXIDFLG	RC_W1	接收识别符标志（Receive Identifier Flag） 此位仅在 LIN 模式下有效（写入）。 当接收到一个具有 RX 匹配且无 ID 奇偶校验错误的标识符时，此标志会被设置。详情请参见“消息过滤和验证”部分。当此标志被设置时，表示在 RX 匹配上接收到一个新的有效标识符。 此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 读取 LINID 寄存器 ● 向此位写入 1 0：未接收到有效 ID。 1：在 RX 匹配上接收到的 LINID[23:16]中包含有效 ID。	0h
23:15	保留			0h

位/域	名称	R/W	描述	复位值
24	PEFLG	RC_W1	奇偶校验错误标志（Parity error flag） 此位在 LIN 或 UART 兼容模式下有效。 当在接收数据中检测到奇偶校验错误时，此位会被设置。在 UART 地址位模式下，奇偶校验基于接收帧的数据和地址字段进行计算。在空闲线模式下，仅使用数据来计算奇偶校验。 当接收到的字符中 1 的数量与其奇偶校验位不匹配时，会产生错误。有关奇偶校验的更多信息，请参见“UART 全局控制寄存器 1”描述。如果奇偶校验功能被禁用，则 PEFLG 标志被禁用并读取为 0。 检测到奇偶校验错误会导致 LIN 生成错误中断（如果 PIEN 位= 1）。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 接收新字符（UART 兼容模式）或帧（LIN 模式） ● 向此位写入 1 0：未检测到奇偶校验错误或奇偶校验被禁用。 1：检测到奇偶校验错误。	0h
25	OREFLG	RC_W1	过载错误标志（Overrun error flag） 此位在 LIN 或 UART 兼容模式下有效（写入）。当数据从 UARTRXSHF 传输到 LINRXD 时覆盖了 LINRXD 或 RXBy 缓冲器中未读的数据时，此位会被设置。如果 OREIEN 位为 1，检测到溢出错误会导致 LIN 生成错误中断。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 0：未检测到溢出错误。 1：检测到溢出错误。	0h

位/域	名称	R/W	描述	复位值
26	FEFLG	RC_W1	帧错误标志（Framing error flag） 此位在 LIN 或 UART 兼容模式下有效（写入）。当预期的停止位未找到时，此位会被设置。在 UART 兼容模式下，仅检查第一个停止位。缺少停止位表明与起始位的同步已丢失，字符帧错误。如果设置 FEIEN 位，检测到帧错误会导致 UART 生成错误中断。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 ● 接收新字符（UART 兼容模式），或新帧（LIN 模式） 在多缓冲模式下，帧由 UARTFORMAT 寄存器定义。 0：未检测到帧错误。 1：检测到帧错误。	0h
27	NREFLG	RC_W1	无响应错误标志（No Response Error Flag） 此位仅在 LIN 模式下有效（写入）。当主设备的头部在 TFRAME_MAX 内未完成时，此位会被设置。此超时时间适用于未知长度的消息帧（标识符为 0 到 61）。该错误由模块的同步器检测到。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 ● 接收新的同步中断 0：未检测到无响应错误。 1：检测到无响应错误。	0h
28	ISFEFLG	RC_W1	不一致的同步字段错误标志（Inconsistent Sync Field Error Flag） 此位仅在 LIN 模式下有效（写入）。当同步器在头部接收过程中检测到不一致的同步字段错误时，此位会被设置。有关更多信息，请参见“头部接收和自适应波特率”部分。此位可以通过以下方式清除： ● 读取 LINIVO0/1 寄存器中对应的中断偏移 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向此位写入 1 ● 接收新的同步中断 0：未检测到不一致的同步字段错误。 1：检测到不一致的同步字段错误。	0h

位/域	名称	R/W	描述	复位值
29	CSEFLG	RC_W1	校验和错误标志（Checksum Error Flag） 这个位仅在 LIN 模式下有效（写入）。当接收节点检测到校验和错误时会设置此位。要使用的校验和类型取决于 LINGCTRL1.CHASUMCFG 位。 以下情况会清除这个位： ● 读取 LINIVO0/1 寄存器中相应的中断偏移量 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向该位写入 1 ● 接收到新的同步中断 0：未检测到校验和错误 1：检测到校验和错误。	0h
30	PBEFLG	RC_W1	物理总线错误标志（Physical Bus Error Flag） 这个位仅在 LIN 模式下有效（写入）。当出现物理总线错误时会设置此位。这是由 TED 中的位监视器检测到的。 以下情况会清除这个位： ● 读取 LINIVO0/1 寄存器中相应的中断偏移量 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向该位写入 1 ● 接收到新的同步中断 注意：只有在无法生成同步中断（由于到 VBAT 的总线短缺）或者无法生成同步中断分隔符（由于到 GND 的总线短缺）时，才会标记 PBEFLG。 0：未检测到物理总线错误。 1：检测到物理总线错误。	0h
31	BEFLG	RC_W1	位错误标志（Bit Error Flag） 这个位仅在 LIN 模式下有效。当出现位错误时会设置此位。这是通过内部位监视器中的位监视器检测到的。 以下情况会清除这个位： ● 读取 LINIVO0/1 寄存器中相应的中断偏移量 ● 设置 RDY 位 ● RST 位 ● 系统复位 ● 向该位写入 1 ● 接收到新的同步中断 这个字段只能在 LIN 模式下写入。 0：未检测到位错误 1：检测到位错误	0h

41.12.9 中断向量偏移寄存器 0（LINIVO0）

偏移地址：0x20

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	IVO0	R	INT0 的中断向量偏移（INT0 Interrupt vector offset ） 该寄存器表示中断线 INT0 的偏移量。读取该寄存器会将其值更新为 LINFLG 中下一个最高优先级的挂起中断，并清除与读取的偏移量对应的标志。 注意：接收（LINFLG[9]）和发送（LINFLG[8]）中断的标志不能通过读取该寄存器中的对应偏移向量来清除（详见 LINFLG 寄存器的详细描述）。	0h
31:5	保留			0h

41.12.10 中断向量偏移寄存器 1（LINIVO1）

偏移地址：0x24

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
4:0	IVO1	R	INT1 的中断向量偏移（INT1 Interrupt vector offset ） 该寄存器表示中断线 INT1 的偏移量。读取该寄存器会将其值更新为 LINFLG 中下一个最高优先级的挂起中断，并清除与读取的偏移量对应的标志。 注意：接收（LINFLG[9]）和发送（LINFLG[8]）中断的标志不能通过读取该寄存器中的对应偏移向量来清除（详见 LINFLG 寄存器的详细描述）。	0h
31:5	保留			0h

41.12.11 配置长度寄存器（LINLCFG）

偏移地址：0x28

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	CLCFG	R/W	配置字符长度（Character length Configure） 这些位仅在 UART 兼容模式下有效。这些位设置 UART 字符长度为 1 到 8 位。 注意：在兼容模式或缓冲 UART 模式下，当接收的数据长度少于八个比特位时，它会在 LINRXD/RXBy 中左对齐，并用尾部的零填充。从 LINRXD 读取的数据应通过软件右移使接收到的数据右对齐。 注意：写入 LINTXD 的数据应右对齐，但不需要在前面加零填充。 这些位仅在 UART 模式下可写。 000：字符长度为 1 位。 001：字符长度为 2 位。 010：字符长度为 3 位。 011：字符长度为 4 位。 100：字符长度为 5 位。 101：字符长度为 6 位。 110：字符长度为 7 位。 111：字符长度为 8 位。	0h
15:3	保留			0h

位/域	名称	R/W	描述	复位值
18:16	FLCFG	R/W	配置帧长度（Frame length Configure） 在 LIN 模式下，这些位表示响应字段中的字节数，从 1 到 8 字节。在缓冲 UART 模式下，这些位表示字符数。当这些位用于表示 LIN 响应长度（LINGCTRL1[0] = 1）时，当有 ID RX 匹配时，该值应更新为响应的预期长度。在缓冲 UART 模式下，这些位表示具有 LINLCFG[2:0]每字符位数的字符数。即，这些位表示发送器/接收器格式的字符数：1 到 8。最多可以有八个字符，每个字符八位。 000：响应字段有 1 字节/字符。 001：响应字段有 2 字节/字符。 010：响应字段有 3 字节/字符。 011：响应字段有 4 字节/字符。 100：响应字段有 5 字节/字符。 101：响应字段有 6 字节/字符。 110：响应字段有 7 字节/字符。 111：响应字段有 8 字节/字符。	0h
31:19	保留			0h

41.12.12 波特率寄存器（LINBR）

偏移地址：0x2C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
15:0	PSCSELL	R/W	选择预分频低（Prescaler Selcet Low） UART/LIN 24 位整数预分频选择。 这些位用于选择 UART/LIN 模块的波特率。这些位在 LIN 模式和 UART 兼容模式下有效。UART/LIN 具有由 LIN 模块输入时钟和该寄存器中的预分频器 P 和 M 决定的内部生成串行时钟。UART/LIN 使用 24 位整数预分频器 P 值来选择超过 16,700,000 个可用波特率中的一个。附加的 4 位分数预分频器 M 细化了波特率选择。	0h
23:16	PSCSELH	R/W	选择预分频高（Prescaler Selcet High） UART/LIN 24 位整数预分频选择。 这些位用于选择 UART/LIN 模块的波特率。这些位在 LIN 模式和 UART 兼容模式下有效。UART/LIN 具有由 LIN 模块输入时钟和该寄存器中的预分频器 P 和 M 决定的内部生成串行时钟。UART/LIN 使用 24 位整数预分频器 P 值来选择超过 16,700,000 个可用波特率中的一个。附加的 4 位分数预分频器 M 细化了波特率选择。	0h
27:24	FDIVSEL	R/W	选择 4 位分数分频（4 bit Fractional Divider Selcet） 这些位在 LIN 或 UART 异步模式下有效。它们用于选择 UART/LIN 模块的波特率，并且是波特率规范的分数部分。M 分频器允许对 P 整数预分频器进行微调，每个 P 整数值有 15 个附加的中间值。	0h
30:28	SFDIVSEL	R/W	选择超分数分频（Superfractional Divider Selcet） 这些位是波特率规范的附加分数部分。这些位允许对分数波特率进行超精细调节，每个 M 分数分频器值有 7 个更多的中间值。详见超分数分频器部分。	0h
31	保留			0h

41.12.13 接收器仿真数据寄存器（LINRXED）

偏移地址：0x30

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	RXED	R	接收器仿真数据（Receiver Emulation Data） 该位仅在 UART 兼容模式下有效。读取此位不会清除 RXRDYFLG 标志。该寄存器应仅由必须连续读取数据缓冲区而不影响 RXRDYFLG 标志的仿真器使用。	0h
31:8	保留			0h

41.12.14 接收数据缓冲寄存器（LINRXD）

偏移地址：0x34

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	RXD	R	接收数据（Received Data） 该位仅在 UART 兼容模式下有效。当一个帧完全接收后，帧中的数据从接收移位寄存器 UARTRXSHF 转移到该寄存器。当这种转移发生时，RXRDYFLG 标志被设置，并且如果 RX 中断使能被设置，则会生成接收中断。当从 LINRXD 读取数据时，RXRDYFLG 标志会自动清除。 当 UART 接收到少于八个比特位长度的数据时，它会将数据以左对齐格式加载到该寄存器中，并用尾部的零填充。因此，软件应对数据进行逻辑移位，以将其右对齐到正确的位置。	0h
31:8	保留			0h

41.12.15 发送数据缓冲寄存器（LINTXD）

偏移地址：0x38

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXD	R/W	发送数据（Transmit Data） 该位仅在 UART 兼容模式下有效。要传输的数据写入该寄存器。从该寄存器向传输移位寄存器 UARTTXSHF 的传输会设置 TXBRDYFLG 标志（LINFLG[23]），这表示 LINTXD 已准备好加载另一个字节的数据。注意：如果 TXIEN（LINIEN[8]）被设置，此数据传输也会引发中断。 注意：写入 LINRXD 寄存器的数据长度少于八个比特位时必须右对齐，但不需要在前面加零填充。	0h
31:8	保留			0h

41.12.16 控制引脚寄存器 0（LINPCTRL0）

偏移地址：0x3C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h

位/域	名称	R/W	描述	复位值
1	RXPEN	R/W	使能接收引脚（Receive pin Enable） 该位在 LIN 或 UART 模式下有效。该位定义了 LINRX 引脚的功能。 0：禁用 LINRX 引脚。 1：启用 LINRX 引脚。	0h
2	TXPEN	R/W	使能发送引脚（Transmit pin Enable） 该位在 LIN 或 UART 模式下有效。该位定义了 LINTX 引脚的功能。 0：禁用 LINTX 引脚。 1：启用 LINTX 引脚。	0h
31:3	保留			0h

41.12.17 控制引脚寄存器 2（LINPCTRL2）

偏移地址：0x44

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	保留			0h
1	RXPVAL	R	接收引脚当前值（Receive pin current value ） 该位在 LIN 或 UART 兼容模式下有效。该位包含 LINRX 引脚上的当前值。	0h
2	TXPVAL	R	发送引脚当前值（Transmit pin current value ） 该位在 LIN 或 UART 兼容模式下有效。该位包含 LINTX 引脚上的当前值。	0h
31:3	保留			0h

41.12.18 比较寄存器（LINCOMP）

偏移地址：0x60

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
2:0	SBEXTSEL	R/W	选择同步中断扩展（Sync Break extend Selcet） 仅适用于 LIN 模式。这些位用于配置同步中断的 Tbit 数量，以将同步字段中的最小 13 Tbit 扩展到最多 20 Tbit。 同步中断的时间延迟计算公式为： TSYNBRK = 13Tbit + SBREAK x Tbit 这些位仅在 LIN 模式下可写。 000：同步中断无额外 Tbit。 001：同步中断有 1 个额外 Tbit。 010：同步中断有 2 个额外 Tbit。 011：同步中断有 3 个额外 Tbit。 100：同步中断有 4 个额外 Tbit。 101：同步中断有 5 个额外 Tbit。 110：同步中断有 6 个额外 Tbit。 111：同步中断有 7 个额外 Tbit。	0h
7:3	保留			0h

位/域	名称	R/W	描述	复位值
9:8	SDCOMPSEL	R/W	选择同步分隔符（Sync Delimiter compare Selcet） 这些位仅在 LIN 模式下有效。这些位用于配置同步字段中同步定界符的 Tbit 数量。 同步定界符的时间延迟计算公式为： TSDEL = （SDEL + 1）Tbit 这些位仅在 LIN 模式下可写。 00：同步定界符有 1 个 Tbit。 01：同步定界符有 2 个 Tbit。 10：同步定界符有 3 个 Tbit。 11：:同步定界符有 4 个 Tbit。	0h
31:10	保留			0h

41.12.19 接收缓冲区寄存器 0（LINRXB0）

偏移地址：0x64

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	RXB3	R	接收缓冲区 3（Receive Buffer 3） 每个在 UARTRXSHF 寄存器中接收的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h
15:8	RXB2	R	接收缓冲区 2（Receive Buffer 2） 每个在 UARTRXSHF 寄存器中接收的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h
23:16	RXB1	R	接收缓冲区 1（Receive Buffer 1） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h
31:24	RXB0	R	接收缓冲区 0（Receive Buffer 0） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。 读取此字节会清除 RXDY 字节。 注意：RXB<x-1> 等同于 LIN 帧的数据字节<x>。	0h

41.12.20 接收缓冲区寄存器 1（LINRXB1）

偏移地址：0x68

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	RXB7	R	接收缓冲区 7（Receive Buffer 7） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h
15:8	RXB6	R	接收缓冲区 6（Receive Buffer 6） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h
23:16	RXB5	R	接收缓冲区 5（Receive Buffer 5） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h

位/域	名称	R/W	描述	复位值
31:24	RXB4	R	接收缓冲区 4（Receive Buffer 4） 每个在 UARTRXSHF 寄存器中接收到的响应数据字节都会根据接收的字节数转移到相应的 RXBy 寄存器。	0h

41.12.21 屏蔽 ID 寄存器（LINIDMASK）

偏移地址：0x6C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXIDMASK	R/W	屏蔽发送 ID（Transmit ID mask） 该字段仅在 LIN 模式下有效。该 8 比特位掩码用于过滤传入的 ID 消息并将其与 ID 字节进行比较。当接收到的 ID 与 TX ID 掩码匹配时，将设置 TXIDFLG，并在启用时触发 ID 中断。掩码中的 0 位表示该位与 ID 字节进行比较。掩码中的 1 位表示该位被过滤，因此不用于比较。当 HGENCTRL 设置为 1 时，如果需要比较完整的 ID，则该字段必须设置为 0xFF。	0h
15:8	保留			0h
23:16	RXIDMASK	R/W	屏蔽接收 ID（Receive ID mask） 该字段仅在 LIN 模式下有效。该 8 比特位掩码用于过滤传入的 ID 消息并将其与 ID 字节进行比较。当接收到的 ID 与 RX ID 掩码匹配时，将设置 ID RX 标志，并在启用时触发 ID 中断。掩码中的 0 位表示该位与 ID 字节进行比较。掩码中的 1 位表示该位被过滤，因此不用于比较。当 HGENCTRL 设置为 1 时，如果需要比较完整的 ID，则该字段必须设置为 0xFF。	0h
31:24	保留			0h

41.12.22 ID 寄存器（LINID）

偏移地址：0x70

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	IDBYTE	R/W	设置 ID 字节（ID byte Set） 该字段仅在 LIN 模式下有效。此字节是 LIN 模式消息 ID。在主节点上，CPU 写入该寄存器会启动报头传输。对于从任务，当 HGENCTRL（LINGCTRL1[12]）为 '0' 时，此字节用于消息过滤。 这些位仅在 LIN 模式下可写。	0h
15:8	IDSTBYTE	R/W	ID 从机任务字节（ID Slave Task byte） 该字段仅在 LIN 模式下有效。此字节包含一个标识符，用于将接收到的报头 ID 进行比较，以决定 LIN 节点是否需要执行 RX 接收响应、TX 发送响应或不采取任何动作。 这些位仅在 LIN 模式下可写。	0h

位/域	名称	R/W	描述	复位值
23:16	RXID	R	接收 ID（Received ID） 该位仅在 LIN 模式下有效。此字节包含当前消息标识符。在报头接收期间，如果没有 ID 奇偶校验错误且存在 RX/TX 匹配，接收到的 ID 将从 UARTRXSHF 寄存器复制到此字节。 注意：如果在 ID 接收期间检测到帧错误（FEFLG），接收到的 ID 也不会被复制到 LINID 寄存器。	0h
31:24	保留			0h

41.12.23 发送缓冲区寄存器 0（LINTXB0）

偏移地址：0x74

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXB3	R/W	发送缓冲区 3（Transmit Buffer 3） 第 3 个要传输的字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
15:8	TXB2	R/W	发送缓冲区 2（Transmit Buffer 2） 第 2 个要传输的字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
23:16	TXB1	R/W	发送缓冲区 1（Transmit Buffer 1） 第 1 个要传输的字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
31:24	TXB0	R/W	发送缓冲区 0（Transmit Buffer 0） 第 0 个要传输的字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。当第 0 个字节被写入 TXBO 缓冲区后，传输将被启动。	0h

41.12.24 发送缓冲区寄存器 1（LINTXB1）

偏移地址：0x78

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
7:0	TXB7	R/W	发送缓冲区 7（Transmit Buffer 7） 要传输的第 7 个字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
15:8	TXB6	R/W	发送缓冲区 6（Transmit Buffer 6） 要传输的第 6 个字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
23:16	TXB5	R/W	发送缓冲区 5（Transmit Buffer 5） 要传输的第 5 个字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h
31:24	TXB4	R/W	发送缓冲区 4（Transmit Buffer 4） 要传输的第 4 个字节被写入此寄存器，然后复制到 UARTTXSHF 进行传输。	0h

41.12.25 最大波特率预分频寄存器（LINMBRPSC）

偏移地址：0x7C

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
12:0	MBRPSC	R/W	最大波特率预分频（Maximum Baud Rate Prescaler） 此字段仅在 LIN 模式下有效。如果 ADAPT 位被设置，该 13 位预分频器将在从模块的同步阶段（参见“头接收和自适应波特率”部分）中使用。通过这种方式，使用自动或选择比特率模式的 UART/LIN 从设备可以自动检测到任何 LIN 总线的合法速率。 MBR 值应编程为允许最大波特率不超过 LIN 网络中预期工作波特率的 10%。否则，0x00 数据字节可能会被错误地检测为同步中断。 默认值基于 70MHz 的 LINCLK（0xDAC）。此 MBR 预分频器由唤醒和空闲时间计数器使用，以相对于 20kHz 速率保持恒定的到期时间。	DACh
31:13	保留			0h

41.12.26 模拟错误与测试寄存器（LINEET）

偏移地址：0x90

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	ALBCFG	R/W	配置模拟回环（Analog loopback Configure） 该位定义了在模拟回环模式下，传输引脚或接收引脚的 I/O 缓冲区是否包含在通信路径中。 0：通过传输引脚使能模拟回环。 1：通过接收引脚使能模拟回环。	0h
1	LBCFG	R/W	配置回环模式（Loopback Mode Configure） 在模拟回环模式下，可以测试通过 I/O 的完整通信路径，而在数字回环模式下，I/O 缓冲区被排除在此路径之外。 0： 启用数字回环。 1：在模块 I/O DFT 模式下启用模拟回环（当 IODFTEN = 1010 时）。	0h
7:2	保留			0h

位/域	名称	R/W	描述	复位值
11:8	IODFTEN	R/W	使能 IO DFT（IO DFT Enable） 该字段用于使能 UART/LIN 模块的 IODFT 模式以进行测试。 0000：禁用 0001：禁用 0010：禁用 0011：禁用 0100：禁用 0101：禁用 0110：禁用 0111：禁用 1000：禁用 1001：禁用 1010：使能 1011：禁用 1100：禁用 1101：禁用 1110：禁用 1111：禁用	5h
15:12	保留			0h
18:16	TXDLYCFG	R/W	配置发送延迟（Transmit Delay Configure） 这些位定义了 LINTX 上的值被延迟的时间，以使 LINRX 上的值变为异步。（不适用于起始位） 000：无延迟 001：延迟 1 个 SCLK 010：延迟 2 个 SCLK 011：延迟 3 个 SCLK 100：延迟 4 个 SCLK 101：延迟 5 个 SCLK 110：延迟 6 个 SCLK 111：延迟 7 个 SCLK	0h
20:19	PSMASK	R/W	屏蔽引脚采样（Pin sample mask） 这些位定义了在传输过程中 TX 引脚的值将被反转的采样编号，以验证接收引脚是否正确使用多数检测电路进行采样。 注意：在针对引脚采样掩码进行 IODFT 模式测试期间，预分频器 P 必须被编程为大于 2。 00 ：无掩码 01：在 TBIT_CENTER 处反转 TX 引脚的值 10：TBIT_CENTER + 1 个 SCLK 处反转 TX 引脚的值 11：在 TBIT_CENTER + 2 个 SCLK 处反转 TX 引脚的值	0h
23:21	保留			0h

位/域	名称	R/W	描述	复位值
24	BRKEEN	R/W	使能 BRKDT 错误（BRKDT error Enable） 此位仅在兼容 UART 模式下有效。该位用于产生 BRKDT 错误（仅 UART 模式）。当该位被设置时，帧的停止位与 '0' 进行 AND 运算并传递给 RSM，使得发生帧错误。然后 RX 引脚被强制为连续低电平 10 个 Tbits，以产生 BRKDT 错误。	0h
25	PEEN	R/W	使能奇偶校验错误（ Parity error Enable） 此位仅在兼容 UART 模式下有效。该位用于产生奇偶校验错误。当该位被设置时，在兼容模式下，接收到的奇偶校验位将被切换，从而产生奇偶校验错误。	0h
26	FEEN	R/W	使能帧错误（ Frame error Enable） 此位用于产生帧错误。 此位仅在兼容 UART 模式下有效。当此位被设置时，接收到的停止位与 '0' 进行 AND 运算并传递给停止位检查电路。	0h
27	保留			0h
28	ISFEEN	R/W	使能不一致的同步字段错误（Inconsistent Sync Field Error Enable） 不一致同步字段错误使能位。 此位仅在 LIN 模式下有效。该位用于产生 ISF 错误。当此位被设置时，同步字段中的位宽会被改变，从而使 ISF 检查失败并设置错误标志。	0h
29	CSEEN	R/W	使能校验和错误（Checksum Error Enable） 此位仅在 LIN 模式下有效。该位用于产生校验和错误。当此位被设置时，接收校验和计算器中的 CTYPE（校验和类型）的极性会被改变，从而产生校验和错误。	0h
30	PBEEN	R/W	使能物理总线错误（Physical Bus Error Enable） 此位仅在 LIN 模式下有效。该位用于产生物理总线错误。当此位被设置时，同步中断字段传输期间接收到的位与 1 进行 OR 运算并传递给位监控电路。	0h
31	BEEN	R/W	使能位错误（Bit Error Enable） 此位仅在 LIN 模式下有效。该位用于产生位错误。当此位被设置时，接收到的位与 1 进行 OR 运算并传递给位监控电路。	0h

41.12.27 全局中断寄存器（LINGIEN）

偏移地址：0xE0

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GIEN0	R/W	为 LIN INT0 使能全局中断（Global Interrupt Enable for LIN INT0） 此位决定 INT0 中断线是否会向 PIE 产生中断。 0：LIN INT0 线不会向 PIE 产生中断。 1：如果发生已启用的中断条件，LIN INT0 线会向 PIE 产生中断。	0h

位/域	名称	R/W	描述	复位值
1	GIEN1	R/W	为 LIN INT1 使能全局中断（Global Interrupt Enable for LIN INT1） 此位决定 INT1 中断线是否会向 PIE 产生中断。 0：LIN INT1 线不会向 PIE 产生中断。 1：如果发生已启用的中断条件，LIN INT1 线会向 PIE 产生中断。	0h
31:2	保留			0h

41.12.28 全局中断标志寄存器（LINGIFLG）

偏移地址：0xE4

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GIFLG0	R	LIN INT0 全局中断标志（Global Interrupt Flag for LIN INT0） 此位指示是否由于 INT0 中断线上的已启用中断而向 PIE 产生了中断。有关产生中断的条件，请参阅 LIN 中断状态寄存器。 通过向 LINGICLR 寄存器中的相应位写入 1 可以清除此位。 0：INT0 线上没有活动的中断。 1：由于 INT0 中断线上已启用的中断而产生了中断。	0h
1	GIFLG1	R	LIN INT1 全局中断标志（Global Interrupt Flag for LIN INT1） LIN INT1 的全局中断标志。 此位指示是否由于 INT1 中断线上的已启用中断而向 PIE 产生了中断。有关产生中断的条件，请参阅 LIN 中断状态寄存器。 通过向 LINGICLR 寄存器中的相应位写入 1 可以清除此位。 0：INT1 线上没有活动的中断。 1：由于 INT1 中断线上已启用的中断而产生了中断。	0h
31:2	保留			0h

41.12.29 清除全局中断标志寄存器（LINGICLR）

偏移地址：0xE8

复位类型：SYSRSn

位/域	名称	R/W	描述	复位值
0	GICLR0	R/W1C	清除 LIN INT0 全局中断标志（Global Interrupt Flag clear for LIN INT0） 此位用于清除 LINGIFLG 寄存器中的相应位。写入 1 以清除 GIFLG0 位。写入 0 没有效果。	0h
1	GICLR1	R/W1C	清除 LIN INT1 全局中断标志（Global Interrupt Flag clear for LIN INT1） 此位用于清除 LINGIFLG 寄存器中的相应位。写入 1 以清除 GIFLG1 位。写入 0 没有效果。	0h
31:2	保留			0h

42 四线串行外设接口（QSPI）

本模块的描述请参见《QSPI 用户手册》 。

43 灵活逻辑块（FLB）

本模块的描述请参见《FLB 用户手册》 。

44 版本历史

表格 246 文件版本历史

日期	版本	变更历史
2025.2	1.0	新建
2025.2	1.1	• 勘误总线架构描述 • 勘误模拟子系统框图 • 勘误时钟连接表格 • 勘误部分寄存器内容，修改笔误
2025.4	1.2	• 勘误 CPU1 邮箱状态寄存器的描述 • 勘误 SPI 的 CPU 接口结构框图 • 更新 CFGSMS 逻辑块描述 • 更新 I2C 的寄存器功能描述
2025.6	1.3	• 更新模拟量输出的描述术语 • 更新比较器的描述术语 • 增加模拟量比较器的输入引脚图 • 勘误 GPBDSEL1_GPIO32 和 GPBFEN_GPIO32 的描述
2025.8	1.4	• 修改 WRPRT 章节的描述，将指令“WRPRT”和“WRPRTDIS”分别改成“WRPRT_DISABLE”和“WRPRT_ENABLE” • 40.6 章节中，将 CanbRegs 的起始地址由“0x50104000”改成“0x50101800”，结束地址由“0x50104FFF”改成“0x50101FFF”
2025.12	1.5	• 修改图 25 X-BAR 输入源的四个“与门”为“或门”
2026.9	1.6	• 修改 SPI 寄存器命名 • 修改 GPIO 地址

声明

本手册由珠海极海半导体有限公司（以下简称“极海”）制订并发布，所列内容均受商标、著作权、软件著作权相关法律法规保护，极海保留随时更正、修改本手册的权利。使用极海产品前请仔细阅读本手册，一旦使用产品则表明您（以下称“用户”）已知悉并接受本手册的所有内容。用户必须按照相关法律法规和本手册的要求使用极海产品。

1、权利所有

本手册仅应当被用于与极海所提供的对应型号的芯片产品、软件产品搭配使用，未经极海许可，任何单位或个人均不得以任何理由或方式对本手册的全部或部分内容进行复制、抄录、修改、编辑或传播。

本手册中所列带有“®”或“™”的“极海”或“Geehy”字样或图形均为极海的商标，其他在极海产品上显示的产品或服务名称均为其各自所有者的财产。

2、无知识产权许可

极海拥有本手册所涉及的全部权利、所有权及知识产权。

极海不应因销售、分发极海产品及本手册而被视为将任何知识产权的许可或权利明示或默示地授予用户。

如果本手册中涉及任何第三方的产品、服务或知识产权，不应被视为极海授权用户使用前述第三方产品、服务或知识产权，也不应被视为极海对第三方产品、服务或知识产权提供任何形式的保证，包括但不限于任何第三方知识产权的非侵权保证，除非极海在销售订单或销售合同中另有约定。

3、版本更新

用户在下单购买极海产品时可获取相应产品的最新版的手册。

如果本手册中所述的内容与极海产品不一致的，应以极海销售订单或销售合同中的约定为准。

4、信息可靠性

本手册相关数据经极海实验室或合作的第三方测试机构批量测试获得，但本手册相关数据难免会出现笔误或因测试环境差异所导致的误差，因此用户应当理

解，极海对本手册中可能出现的该等错误无需承担任何责任。本手册相关数据仅用于指导用户作为性能参数参照，不构成极海对任何产品性能方面的保证。

用户应根据自身需求选择合适的极海产品，并对极海产品的应用适用性进行有效验证和测试，以确认极海产品满足用户自身的需求、相应标准、安全或其它可靠性要求；若因用户未充分对极海产品进行有效验证和测试而致使用户损失的，极海不承担任何责任。

5、合规要求

用户在使用本手册及所搭配的极海产品时，应遵守当地所适用的所有法律法规。用户应了解产品可能受到产品供应商、极海、极海经销商及用户所在地等各国有关出口、再出口或其它法律的限制，用户（代表其本身、子公司及关联企业）应同意并保证遵守所有关于取得极海产品及/或技术与直接产品的出口和再出口适用法律与法规。

6、免责声明

本手册由极海“按原样”（as is）提供，在适用法律所允许的范围内，极海不提供任何形式的明示或暗示担保，包括但不限于对产品适销性和特定用途适用性的担保。

极海产品并非设计、授权或担保适合用于军事、生命保障系统、污染控制或有害物质管理系统中的关键部件，亦非设计、授权或担保适合用于在产品失效或故障时可导致人员受伤、死亡、财产或环境损害的应用。

如果产品未标明“汽车级”，则表示不适用于汽车应用。如果用户对产品的应用超出极海提供的规格、应用领域、规范，极海不承担任何责任。

用户应该确保对产品的应用符合相应标准以及功能安全、信息安全、环境标准等要求。用户对极海产品的选择和使用负全部的责任。对于用户后续在针对极海产品进行设计、使用的过程中所引起的任何纠纷，极海概不承担责任。

7、责任限制

在任何情况下，除非适用法律要求或书面同意，否则极海和/或以“按原样”形式提供本手册及产品的任何第三方均不承担损害赔偿责任，包括任何一般、特殊因使用或无法使用本手册及产品而产生的直接、间接或附带损害（包括但不限于数据丢失或数据不准确，或用户或第三方遭受的损失），这涵盖了可能导致的人身安全、财产或环境损害等情况，对于这些损害极海概不承担责任。

8、适用范围

本手册的信息用以取代本手册所有早期版本所提供的信息。

9、语言版本和人工智能

本手册以多种语言提供。如果中文版本与翻译后的信息存在不一致，以中文版本为准。对于因第三方人工智能系统调用、解析本手册内容而产生的任何误解、偏差或损失，极海不承担任何法律责任。

©2026 珠海极海半导体有限公司 – 保留所有权利