M76E00_M73E00 系列

用户手册

Version 1.5

修订日期： 2025-05-29

版权声明

修订记录

表 0–1 修订记录

版本	章节	修订说明
V1.5	Analog–to–Digital Converter (ADC)	更正了不适用的寄存器描述。
V1.4	Clock Manage Unit (CMU)	更正了 CLK_PWM2 的偏移地址。
	Analog–to–Digital Converter (ADC)	删除了不适用的 12–bit ADC 模式描述。
	USB Full Speed On–The–Go (USB FS OTG)	删除了不适用的 DMA 相关功能。
V1.3	Direct Memory Access (DMA)	重命名了设备重要配置信息 中 ADC 设备名称。
	Thermal Sensor (THS)	更正了0x10C+n*0x20 THSn_DATA 的位宽，从 11:0 改为 13:0。
	Pulse Width Modulation Control System (PWMCS)	更正了 和 的位域值。
V1.2	Direct Memory Access (DMA)	更正了设备重要配置信息中 I2C 的位宽，从 8 位更正为 16 位。
	Mailbox (MBOX)	删除了不适用的 Mailbox 网络描述。
	SID 功能描述	• 新增 BROM_SECURE 区域定义章节。 • 修正了 CALI 区域定义的校准值。
	Pulse Width Modulation Control System (PWMCS)	QOUT_POS_SRC_SEL 中 0xD 值从 BIS_IF1 改为 BIS_IF0。
		更正了 0x060 VoltADC 寄存器的位域值信息。
	Pulse Width Modulation Control System (PWMCS)	在 EPWM_NRCFLT_INCTL 寄存器中删除了不适用的 EPWM_DBG_HOLD_EN 位域。
	Analog–to–Digital Converter (ADC)	• 新增了采样保持时间的计算公式。 • 新增了采样保持时间与 VCM_SET 的配置关系。 • 更新了采样转换时间的计算公式。 • 更新了 0x094 ADC_CTL bit 30 关于 VCM_SET 的描述。 • 更新了寄存器描述 • 删除了不适用的 12–bit ADC 模式。
	Parallel Bus (PBUS)	新增了 信号相位配置。
V1.1	Parallel Bus (PBUS)	更新了 PBUS 功能描述中的使用流程图。
V1.0	–	初版。

内容

1. 文档说明

1.1. 适用范围

用户手册适用于板级产品设计人员和产品软件开发商。 本手册假定读者具有计算机背景工程和/或软件工程，并了解数字化的概念、系统设计、 微处理器架构、输入/输出 (I/O) 设备、行业标准通信和设备接口协议。

1.2. 内容组成

用户手册在系统级别描述了芯片，并提供了架构概述。 它还描述了系统内存映射、系统级中断事件、外部引脚和引脚复用、外部存储器、 系统调试、系统启动、多媒体子系统、电源管理和系统安全。

1.3. 寄存器属性

术语	定义/寄存器类型
RO	只读（Read-only）
WO	只写（Write-only）
R/W	可读可写（Read/Write）
R/W1C	可读并写‘1’清零（Read/Write-‘1’-clear），常见于中断标志寄存器
R/W1T	可读并写‘1’反转（Read/Write-‘1’-toggle）
R/WAC	可读并写‘1’后自动清零（Read/Write-auto-clear），常见于模块的软复位寄存器
-	表示保留（Reserved），通常是不可写并且读返回‘0’
‘u’	寄存器值不确定，例如， 0x987a_bu65

1.4. 数值定义

类型	单位	大小
容量	1K	1024
	1M	1,048,576
	1G	1,073,741,824

类型	单位	大小
频率	1K	1000
	1M	1000,000
	1G	1000,000,000

1.5. 术语、缩写词

术语	全称	中文说明
ADC	Analog Digital Converter	模拟数字转换器
ADCIM	ADC Interface Management	模数转换管理模块
AES	Advanced Encryption Standard	一种对称分组密钥算法
API	Application Programming Interface	应用程序接口
BCLK	Bit Clock	比特时钟，每个时钟周期传输 1bit 音频数据
blit	Bit Block Transfer	位块搬移
BROM	Boot ROM	固化在芯片中的启动程序
CAN	Controller Area Network	控制器局域网络
CBC	Cipher Block Chaining	密码分组链模式
CE	Crypto Engine	硬件加解密算法引擎
CIR	consumer IR	消费者红外
CMA	Contiguous Memory Allocator	连续内存分配器
CMD	Command	SD/eMMC 协议的命令
CMU	Clock Manage Unit	时钟管理单元
CPHA	Clock Phase	时钟相位
CPOL	Clock Polarity	时钟极性
CTR	Counter	计数器模式

术语	全称	中文说明
CTS	Ciphertext Stealing	一种对称分组密码算法模式
DE	Display Engine	显示引擎
DES	Data Encryption Standard	一种对称分组密码算法
DF	Decimation Filter	抽值滤波器
DI	Display Interface	显示接口，包括 RGB/ LVDS/ DSI 等
DMA	Direct Memory Access	直接存储器访问
DMIC	Digital Microphone	数字麦克风
DRQ	DMA Request	指 DMA 请求的端口号
Dual SPI	Dual Serial Peripheral Interface	双路 SPI
DVP	Digital Video Port	数字视频接收端口
ECB	Electronic Code Book	电子密码本模式
EFF	Extended Frame Format	扩展帧
eFuse	Electric Fuse	电子熔丝
eMMC	Embedded Multi Media Card	内嵌（在板卡上）式多媒体卡
ESC	EtherCAT Slave Controller	一种实时工业以太网从机控制器
GE	Graphics Engine	2D 图形加速引擎
GIC	Generic Interrupt Controller	通用中断控制器
GPAI	General Purpose Analog Input Interface	通用模拟输入接口
GPIO	General Purpose Input Output	通用输入输出
HUK	Hardware Unique Key	硬件唯一密钥
I8080	Intel 8080 architecture	因特尔 8080 总线标准
ISP	Image Signal Processing	图像信号处理
LRCK	left right clock	左右声道时钟

术语	全称	中文说明
LVDS	Low Voltage Differential Signaling	低压差分信号的显示接口标准
MAC	Media Access Control	媒体访问控制，Ethernet 中的一个组成模块
MD5	Message Digest Algorithm	消息摘要算法
MDIO	Management Data Input/Output	PHY 的管理接口，用来读/写 PHY 的寄存器
MIPI DSI	Mipi Display Serial Interface	Mipi 组织定义的一个显示接口标准
MMC	Multi Media Card	多媒体卡
MPP	Media Process Platform	通用多媒体处理软件平台
MPP	Media Prosess Platform	多媒体处理平台
NEC	–	一种红外编码协议
PBUS	Parallel Bus	并行总线
PCM	Pulse Code Modulation	脉冲编码调制
PDM	Pulse Density Modulation	脉冲密度调制
PHY	Physic	物理接口收发器
PLL	Phase Locked Loop	锁相环时钟
PNK	Part Number Key	芯片型号密钥
PSK	Partner Secret Key	合作伙伴密钥
PWM	Pulse-Width Modulation	脉冲宽度控制，简称脉宽调制
QSPI	Quad Serial Peripheral Interface	四路 SPI，又称 QSPI
RC5	–	一种红外编码协议
RC6	–	一种红外编码协议
RGB	Red Green Blue	按红绿蓝三原色编码的显示接口标准
RLC	Run-Length Code	游程编码
RMII	Reduced MII	简化的媒体独立接口，使用 2 根线来传输数据

术语	全称	中文说明
RSA	Rivest–Shamir–Adleman	一种非对称密钥算法
RTC	Real Time Clock	实时时钟
RTR	Remote Transmission Request	远程请求帧
SD	Secure Digital	Flash 存储卡的一种标准，即常见的 SD 卡
SDCard	Secure Digital Memory Card	安全数码卡，同“SD 卡”
SDIO	Secure Digital Input and Output	安全数字输入输出接口
SDM	Sigma Delta Modulation	西格玛三角洲调解器
SDMC	SDCard & SDIO & eMMC Host Contollor	SD 卡/eMMC 主控制器
SHA	Secure Hash Algorithm	安全散列算法
SID	Secure ID	安全 ID 模块
SPI	Serial Peripheral Interface	串行外设接口
SSK	Symmetric Secret Key	对称密钥
TBC	Time Base Counter	时基计数器，用于产生锯齿波
TRNG	True Random Number Generator	真随机数生成器
TSensor/TSen	Thermal Sensor	温度传感器
V4L2	Video For Linux Two	Linux 中的视频框架第二版
VBI	Vertical Blanking Interval	垂直消隐期
VE	Video Engine	视频加速引擎
XTS	XEX Tweakable Block Ciphertext with Ciphertext Stealing	一种对称分组密码算法模式

2. 地址映射

模块名称	起始地址	结束地址	空间大小
DMA0	0x10000000	0x10000FFF	4 KB
DMA1	0x10001000	0x10001FFF	4 KB
DCE	0x10010000	0x10010FFF	4 KB
CE	0x10020000	0x1002FFFF	64 KB
USB_FS_DR	0x10240000	0x1024FFFF	64 KB
EMAC	0x10280000	0x1028FFFF	64 KB
CAN_FD0	0x102C0000	0x102C0FFF	4 KB
CAN_FD1	0x102C1000	0x102C1FFF	4 KB
SPI0	0x10400000	0x10400FFF	4 KB
SPI1	0x10401000	0x10401FFF	4 KB
SPI2	0x10402000	0x10402FFF	4 KB
SPI3	0x10403000	0x10403FFF	4 KB
SPI4	0x10404000	0x10404FFF	4 KB
SPI5	0x10405000	0x10405FFF	4 KB
SPI_ENC	0x1040F000	0x1040FFFF	4 KB
AHBCFG	0x104FE000	0x104FEFFF	4 KB
GPIO	0x10700000	0x10700FFF	4 KB
PBUS	0x107F0000	0x1080FFFF	128 KB
MBOX_C02C1	0x10B00000	0x10B00FFF	4 KB
MBOX_C12C0	0x10B03000	0x10B03FFF	4 KB
CORDIC	0x11000000	0x11000FFF	4 KB

模块名称	起始地址	结束地址	空间大小
PFM	0x11030000	0x11030FFF	4 KB
PWMCS	0x11040000	0x1104FFFF	64 KB
ADC0	0x11050000	0x11050FFF	4 KB
ADC1	0x11051000	0x11051FFF	4 KB
THS	0x11052000	0x11052FFF	4 KB
VTS	0x11053000	0x11053FFF	4 KB
FFT	0x11054000	0x11054FFF	4 KB
CPM	0x11060000	0x11060FFF	4 KB
PGA	0x11070000	0x11070FFF	4 KB
SDFM	0x110B0000	0x110B0FFF	4 KB
PWM0	0x110C0000	0x110C00FF	256 B
PWM1	0x110C0100	0x110C01FF	256 B
PWM2	0x110C0200	0x110C02FF	256 B
PWM3	0x110C0300	0x110C03FF	256 B
PWM4	0x110C0400	0x110C04FF	256 B
PWM5	0x110C0500	0x110C05FF	256 B
GPT0	0x110D0000	0x110D00FF	256 B
GPT1	0x110D0100	0x110D01FF	256 B
GPT2	0x110D0200	0x110D02FF	256 B
GPT3	0x110D0300	0x110D03FF	256 B
SYSCFG	0x18000000	0x18000FFF	4 KB
CMU	0x18020000	0x1802FFFF	64 KB
UART0	0x18710000	0x18710FFF	4 KB

模块名称	起始地址	结束地址	空间大小
UART1	0x18711000	0x18711FFF	4 KB
UART2	0x18712000	0x18712FFF	4 KB
UART3	0x18713000	0x18713FFF	4 KB
UART4	0x18714000	0x18714FFF	4 KB
UART5	0x18715000	0x18715FFF	4 KB
UART6	0x18716000	0x18716FFF	4 KB
UART7	0x18717000	0x18717FFF	4 KB
UART8	0x18718000	0x18718FFF	4 KB
I2C0	0x18720000	0x18720FFF	4 KB
I2C1	0x18721000	0x18721FFF	4 KB
I2C2	0x18722000	0x18722FFF	4 KB
I2C3	0x18723000	0x18723FFF	4 KB
WDOG0	0x19000000	0x19000FFF	4 KB
WDOG1	0x19001000	0x19001FFF	4 KB
WRI	0x1900F000	0x1900FFFF	4 KB
SID	0x19010000	0x19010FFF	4 KB
GTC	0x19050000	0x19051FFF	8 KB
CAN	0x19060000	0x19060FFF	4 KB
E907_CLINT	0x20000000	0x2000FFFF	64 KB
E907_CLIC	0x20800000	0x2080FFFF	64 KB
BROM	0x30000000	0x3000FFFF	64 KB
SRAM	0x30040000	0x3013FFFF	1024 KB
PSRAM	0x40000000	0x5FFFFFFF	512 MB

模块名称	起始地址	结束地址	空间大小
FLASH_XIP	0x60000000	0x7FFFFFFF	512 MB
RESERVED	0x80000000	0xFFFFFFFF	2 GB

注：

通过配置 SYSCFG SRAM_MAP_CFG[19:16]，可以将 SRAM 空间的前 384 KB（0x30040000~0x3009FFFF）切换为 TCM，提高访问速度。

3. 处理器和总线

3.1. Central Process Unit (CPU)

CPU 内核是一款基于 RV32IMAFDCP 指令架构的高性能嵌入式处理器。

3.1.1. 特性说明

- RV32IMAFDCP 指令集
- 双精度浮点单元
- 5 级单发按序执行流水线
- 一级哈佛结构，ICACHE 16 KB，DCACHE 16 KB，缓存行大小为 32 字节
- CPU0 支持紧耦合存储 (TCM) ，CPU1 不支持 TCM，I/DTC 可选 4 种灵活配置
- 物理内存保护 (PMP)
- 核内中断 CLINT 和中断控制器 CLIC，最大 32 级中断，支持中断嵌套
- 支持中断处理加速技术、支持 MCU 扩展特性
- 机器模式和用户模式
- 标准 2 线 JTAG 调试接口

3.1.2. 中断号

中断号	模块
7	CPU_TIMER
16	PBUS
19	PFM
20	CPM
21	SDFM
23	CORDIC
28	ADC

中断号	模块
31	DCE
32	DMA0
33	DMA1
39	GMAC
41	SPI_ENC
42	QSPI0
43	QSPI1
44	QSPI2
45	QSPI3
46	QSPI4
47	QSPI5
60	USB_FS_DR
64	WDG0
65	WDG1
68	GPIO_GRP_A
69	GPIO_GRP_B
70	GPIO_GRP_C
71	GPIO_GRP_D
72	GPIO_GRP_E
73	GPIO_GRP_F
74	GPIO_GRP_G
75	GPIO_GRP_H
76	GPIO_GRP_J

中断号	模块
77	GPIO_GRP_K
78	UART0
79	UART1
80	UART2
81	UART3
82	UART4
83	UART5
84	UART6
85	UART7
86	UART8
91	VTS
94	THS
96	PWM0
97	PWM1
98	PWM2
99	PWM3
100	PWM4
101	PWM5
110	GPT0
111	GPT1
112	GPT2
113	GPT3
118	MBOX_C02C1

中断号	模块
119	MBOX_C12C0
120	RDC
121	CE
126	I2C0
127	I2C1
128	I2C2
129	I2C3
135	CANFD0
136	CANFD1
137	CAN
141	PWMCS FAULT
142	PWMCS EPWM0
143	PWMCS EPWM1
144	PWMCS EPWM2
145	PWMCS EPWM3
146	PWMCS EPWM4
147	PWMCS EPWM5
148	PWMCS EPWM6
149	PWMCS EPWM7
150	PWMCS EPWM8
151	PWMCS EPWM9
152	PWMCS EPWM10
153	PWMCS EPWM11

中断号	模块
166	PWMCS CAP0
167	PWMCS CAP1
168	PWMCS CAP2
169	PWMCS CAP3
170	PWMCS CAP4
171	PWMCS CAP5
172	PWMCS QEP0
173	PWMCS QEP1
174	PWMCS QEP2
175	PWMCS QEP3
176	PWMCS QEP4
177	PWMCS QEP5
178	PWMCS QEP6
179	PWMCS QEP7
180	PWMCS QEP8
181	PWMCS QEP9
186	FFT

3.1.3. CLIC 中断控制器

核内中断控制器（以下简称 CLIC），仅用于对中断源进行采样，优先级仲裁和分发。其中 CLIC 支持中断嵌套，中断优先级最多为32级。每个中断目标拥有四个 memory-mapped 的控制寄存器，如下所示，用以配置中断目标各个属性:

- 0x1000+4*i 中断等待寄存器 CLICINTIP[i]
- 0x1001+4*i 中断使能寄存器 CLICINTIE[i]
- 0x1002+4*i 中断属性寄存器 CLICINTATTR[i]
- 0x1003+4*i 中断控制寄存器 CLICINTCTL[i]

注：

关于寄存器的详细配置说明，可查看 CLIC 寄存器描述。

3.1.4. CLIC 寄存器描述

CLIC 寄存器地址映射如下表（下表中的 i 与中断号相对应）：

地址	名称	描述
E907_CLIC + 0x0	CLICCFG	CLIC 配置寄存器
E907_CLIC + 0x8	MINTTHRESH	中断阈值寄存器
E907_CLIC + 0x1000 + 4 × i	CLICINTIP[i]	中断源 i 等待寄存器
E907_CLIC + 0x1001 + 4 × i[1]	CLICINTIE[i]	中断源 i 使能寄存器
E907_CLIC + 0x1002 + 4 × i[1]	CLICINTATTR[i]	中断源 i 属性寄存器
E907_CLIC + 0x1003 + 4 × i[1]	CLICINTCTL[i]	中断源 i 控制寄存器

(1) i 与中断号相对应

3.1.4.1. 0x0 配置寄存器 CLICCFG

默认值：0x07			
位域	类型	默认值	描述
7	–	0x0	–
6:5	R	0x0	nmbits 特权态有效位数，固定为 0。
4:1	R	0x5	nlbits 中断优先级有效位数。
0	R	0x1	nvbits 硬件矢量中断实现标志位，此位恒为 1。

3.1.4.2. 0x8 中断阈值寄存器 MINTTHRESH

默认值：0x00000000			
位域	类型	默认值	描述
31:24	R/W	0x0	mth 机器模式中断的阈值，此域定义了当前处于等待状态的中断请求能够向 CPU 流水线核心发起中断请求的优先级临界值。
23:0	–	0x0	–

3.1.4.3. 0x1000+4*i 中断等待寄存器 CLICINTIP[i]

1

默认值：0x00			
位域	类型	默认值	描述
7:1	–	0x0	–
0	R	0x0	IP 中断等待，此位为只读位域。此域的值需要通过对中断源进行操作来实现，中断源为高则 IP 为 1，中断源为低则 IP 为 0。

3.1.4.4. 0x1001+4*i 中断使能寄存器 CLICINTIE[i]

1

默认值：0x00			
位域	类型	默认值	描述
7:1	–	0x0	–
0	R/W	0x0	IE 中断使能，此位置 1 表示对应的中断源被使能。

3.1.4.5. 0x1002+4*i 中断属性寄存器 CLICINTATTR[i]

1

默认值：0xC0			
位域	类型	默认值	描述

默认值：0xC0			
7:6	R	0x3	mode 中断特权态，此域固定为 2'b11，表示机器模式中断。
5:3	–	0x0	–
2:1	R	0x0	trig 中断触发方式，此位固定为 0，代表电平中断。
0	R/W	0x0	shv 矢量中断使能，此位置 1 则使能硬件矢量中断。

3.1.4.6. 0x1003+4*i 中断控制寄存器 CLICINTCTL[i]

1

默认值：0x1F			
位域	类型	默认值	描述
7:3	R/W	0x0	int_ctl 参与仲裁优先级。
2:0	R	0x3	–

3.2. BUS

图 3–1 : 总线框图如下所示：

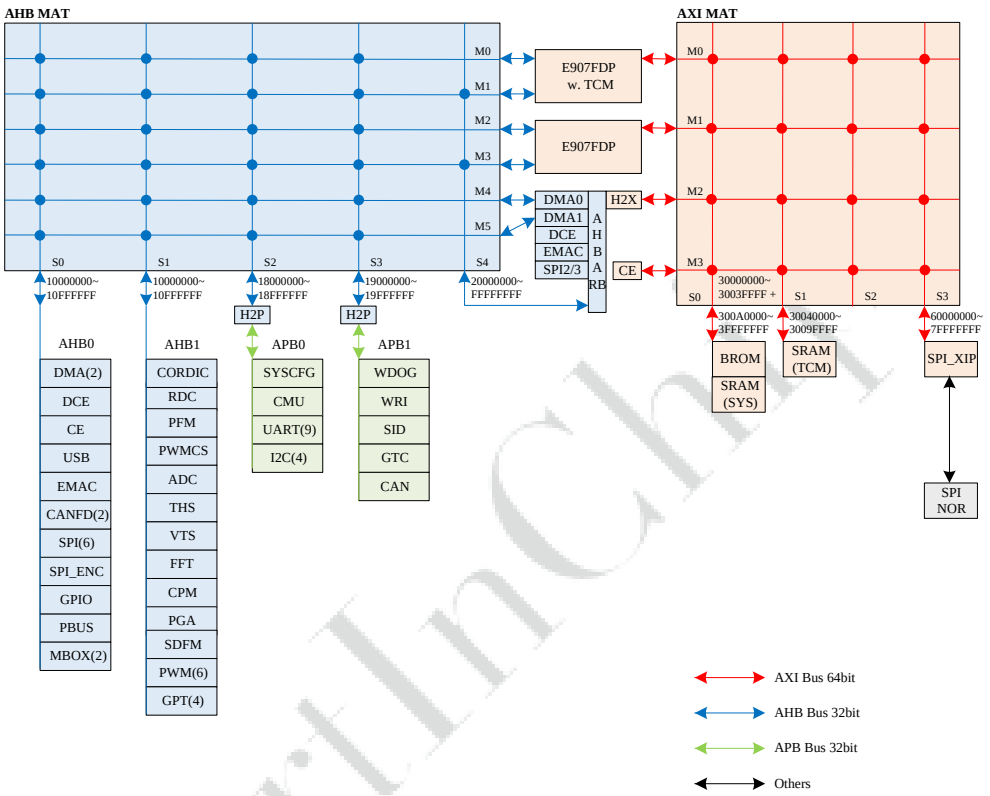


图 3–1 总线框图

> **注：**
>
> 通过配置 SYSCFG SRAM_MAP_CFG[19:16]，可以将 SRAM 空间的前 384 KB（0x30040000~0x3009FFFF）切换为 TCM，提高访问速度。

总线空间分配及内存属性 如下所示。

表 3–1 总线空间分配及内存属性

地址空间定义				CPU SYSMAP 硬件定义（机器模式内存属性）				
空间段	起始地址	结束地址	大小	变量	ADDR	SO	C	B
RESERVE0	0x00000000	0x0FFFFFFF	256 MB	SYSMAP_BASE_ADDR0	28'h10000	1	0	0
DEVICE	0x10000000	0x1FFFFFFF	256 MB	SYSMAP_BASE_ADDR1	28'h20000	1	0	0
E907	0x20000000	0x2FFFFFFF	256 MB	SYSMAP_BASE_ADDR2	28'h30000	1	0	0
BROM	0x30000000	0x3003FFFF	256 KB	SYSMAP_BASE_ADDR3	28'h30040	0	1	1
SRAM	0x30040000	0x37FFFFFF	127 MB	SYSMAP_BASE_ADDR5	28'h38000	0	1	1

表 3–1 总线空间分配及内存属性 (续)

地址空间定义				CPU SYSMAP 硬件定义（机器模式内存属性）				
空间段	起始地址	结束地址	大小	变量	ADDR	SO	C	B
PSRAM	0x40000000	0x5FFFFFFF	512 MB	SYSMAP_BASE_ADDR6	28'h60000	0	1	1
FLASH	0x60000000	0x7FFFFFFF	512 MB	SYSMAP_BASE_ADDR7	28'h80000	0	1	1

3.3. Direct Memory Access (DMA)

Direction memory access (DMA) 模块提供一种高效的数据传输技术，能够在不受 CPU 干预的情况下进行设备间的数据传输，从而减少 CPU 负荷。CPU 使能 DMA 之后，由 DMA 独立自主控制数据传输。此外，DMA 提供的数据交换具有高带宽和低延迟的优点。

3.3.1. 特性说明

- 每通道有 128 个源端和 128 个终端可选。
- 采用任务链表配置方式，寄存器描述通道状态。
- 设备位宽支持 32/ 64/ 128 位，兼容 8/ 16 位，突发长度支持 1/ 4/ 8/ 16 个。
- DMA 源端、终端地址 4 bytes 对齐。
- 2 个 DMA，每个 DMA 8 个通道

3.3.2. 功能描述

DMA 控制器通过任务链表管理多个数据传输任务，支持多种传输模式，并可以根据外设的 REQ/ ACK 信号进行数据传输。配置时需注意不同的外设支持不同的传输模式和数据宽度。

DMA 支持以下几种传输模式：

- 存储与储存传输。
- 外设到储存传输。
- 外设与外设的传输。

3.3.2.1. 设备类型及其 DMA 的交互机制

本节描述了下列设备类型与 DMA 的交互机制：

- **存储模块**：SRAM 或 PSRAM，与 DMA 交互时，无需单独的 REQ/ACK 机制。
- **非存储模块**：除 SRAM 与 PSRAM 之外的其它设备（外设），例如 UART 或 PSI，与 DMA 的交互采用 REQ/ACK 握手机制。

当 DMA 模式选择为握手模式时，设备需主动发起请求（读或写），DMA 再接收到请求后才会开始数据传输。这种机制是为了防止外设处理速度慢长时间占用总线。

3.3.2.2. 任务链表配置

DMA 通过任务链表来管理多个传输任务。每个任务单元包含多个字段，如任务 ID、配置信息、源地址、目标地址、传输长度等。当`下一任务高 8bit 地址 + 下一任务低 32bit 地址 = 0xff_fffffffc`时，表示 DMA 任务链表结束。

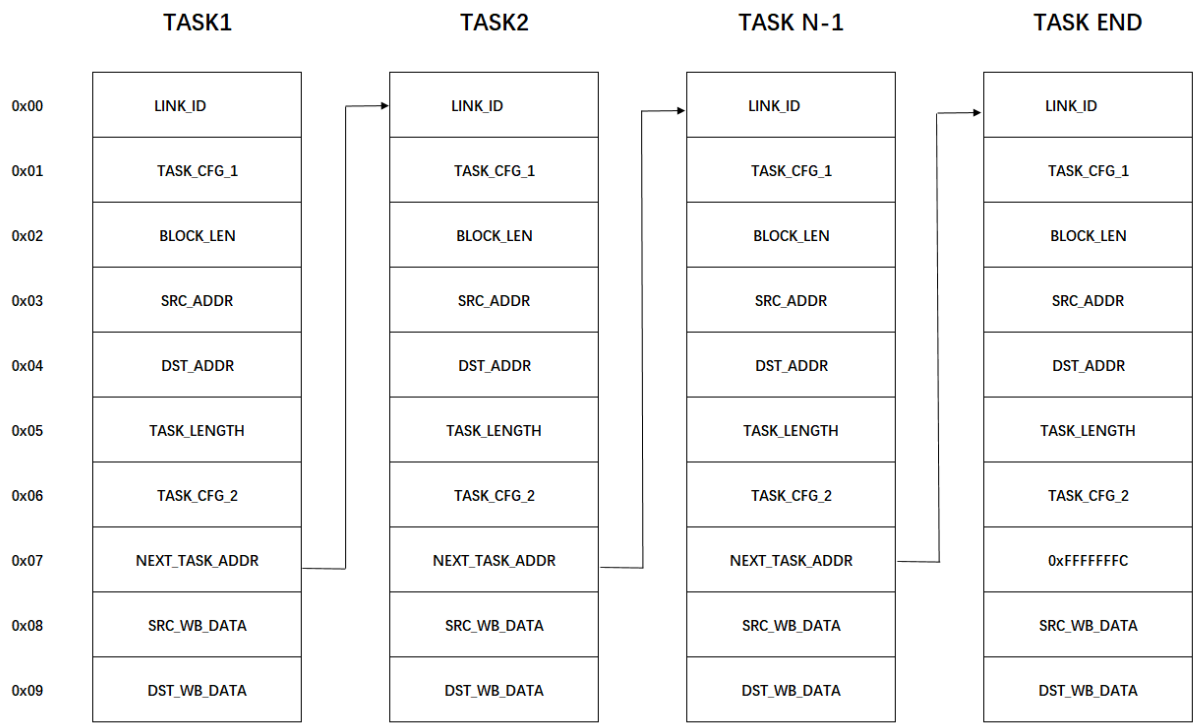


图 3–2 任务链表

多个任务组成任务链表，每个任务单元为 32 位，从低地址往高地址存放。DMA 在工作时会读出任务相关参数更新到对应寄存器中（软件可见）。

表 3–2 任务链表参数描述

参数	描述	备注
LINK ID	当前任务链表的 ID 号。	确保任务链表中的 LINK_ID 与 DMA_LINK_ID_SET 一致。
TASK_CFG_1	当前任务配置，包括源端和终端的 BURST 和 WIDTH 等。	外设模块的 WIDTH 需要与自身 FIFO 寄存器位宽对齐。
BLOCK_LEN	当前任务块长度设置。	–
SRC_ADDR	当前任务源端低 32bit 起始地址。	需要 4 byte 对齐。
DST_ADDR	当前任务终端低 32bit 起始地址。	需要 4 byte 对齐。

表 3–2 任务链表参数描述 (续)

参数	描述	备注
TASK_LENGTH	当前任务传输字节数，最大可设置为 32M–1 字节。	外设模块的总传输长度需要与 WIDTH 对齐，回写使能。
TASK_CFG_2	当前任务访问总线 qos、终端高 8bit 起始地址、源端高 8bit 起始地址、下一任务高 8bit 地址。	下一任务高 8bit 地址 + 下一任务低 32bit 地址 =0xff_ffffffc，表示 DMA Link 结束。
NEXT_TASK_ADDR	下一任务低 32bit 地址。	下一任务高 8bit 地址 + 下一任务低 32bit 地址 =0xff_ffffffc，表示 DMA Link 结束。
SRC_WB_DATA	源端回写的数据。	–
DST_WB_DATA	终端回写的数据。	–

3.3.2.3. 设备源端和终端传输类型

本节介绍了设备的源端 (Source Type) 和终端 (Destination type) 传输类型配置及其工作机制。源端对象和终端对象均可以是储存或外设。

下表分别列出了存储设备和非存储设备的源端和终端传输类型配置：

表 3–3 存储设备源端和终端传输类型配置说明

DMA_TASK_CFG_1 寄存器	含义	是否握手	设备类型
0x0: io_single	DMA 以 transfer（AHB 协议 ）进行传输，每次握手进行一次 single transfer。	需要	非存储设备
0x1: io_burst	DMA 以 transfer（AHB 协议）进行传输，一次握手进行多次 single transfer。	需要	非存储设备
0x2: memory	存储传输类型	不需要	存储设备
0x3: memoryset	存储传输类型 使用 memoryset 时，task 中源端与终端参数需配置一致。	不需要	存储设备
0x4: io_fast	DMA 以 transfer（AHB 协议 ）进行传输，一次握手进行多次 burst transaction。 外设支持 IO_FAST 模式需满足下列条件： • FIFO 地址入口是一个空间。 • 空间需要大于等于 Block length。	需要	非存储设备

注：

SRAM 与 DRAM 等存储设备仅能使用 Memory 或 memoryset 类型。对于非存储设备（外设），需按照 spec 规格选择上述适用的其中一种传输类型。

3.3.2.3.1. 数据传输的长度对齐与传输效率

对于非存储设备类型，IO_SINGLE、IO_BURST 和 IO_FAST 的传输效率为 IO_FAST > IO_BURST > IO_SINGLE：

- 当数据长度与 Block Length 对齐时

  如图所示，ADDR = 0x30、Date Width = 32、Type = io_burst、Burst = 4 且 Block_length = 16，DMA 传输数据总量 28（16+3*4）：

  - 由于 DMA 传输数据总量 (28) ≥ Block_length (16)，则第一次数据传输使用 burst 类型。
  - 由于剩余的 DMA 传输数据量 (12) < Block_length (16)，不满足 burst 长度需求，尽管 Type=io_burst，硬件会自动拆分为三次 single 类型进行传输。

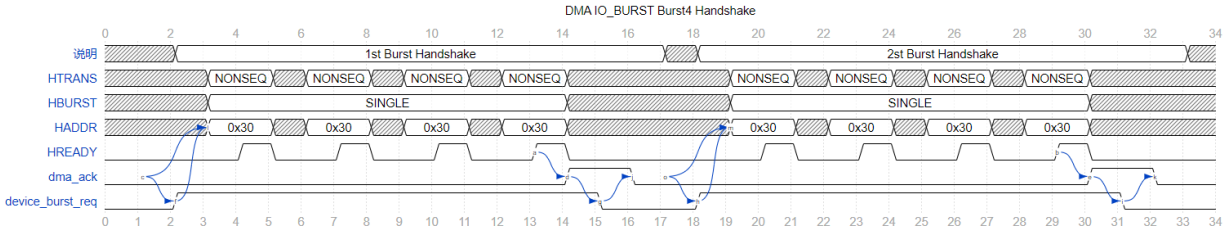


图 3–3 Type=IO_BURST 且数据长度与 Length 对齐

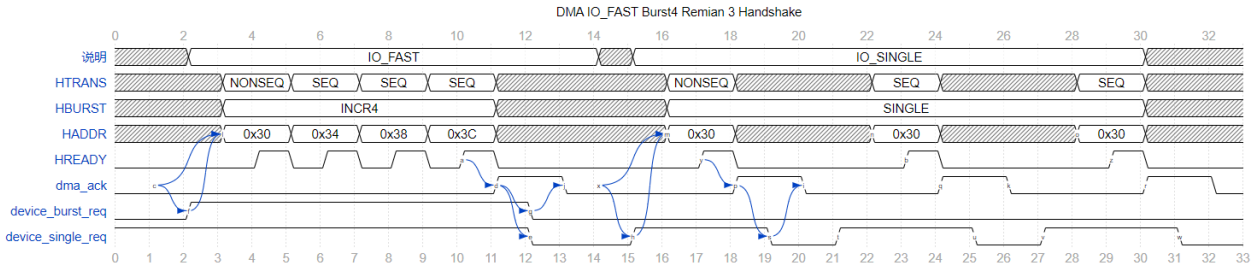


图 3–4 Type=IO_FAST 且数据长度与 Length 对齐

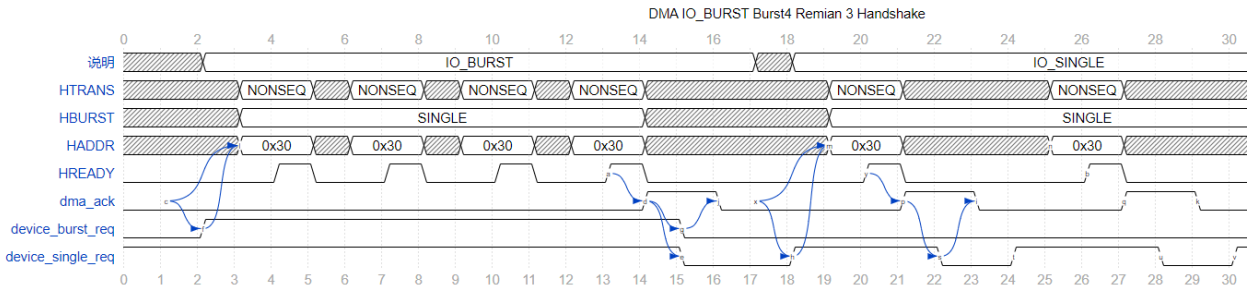


图 3–5 Type=IO_BURST 且数据长度与 Length 非对齐

- 当数据长度与 Block Length 非对齐，配置为 Type=IO_FAST 时

  如图所示，ADDR = 0x30、Date Width = 32、Type = io_fast、Burst = 4 且 Block_length = 16，DMA 传输数据总量 28（16+3*4）

- 由于 DMA 传输数据总量 (28) ≥ Block_length (16)， 第一次数据传输使用 fast 类型。
- 由于剩余的 DMA 传输数据量 (12) < Block_length (16)，不满足 fast 长度需求，尽管 Type=io_fast，硬件会自动拆分为三次 single 类型进行传输。

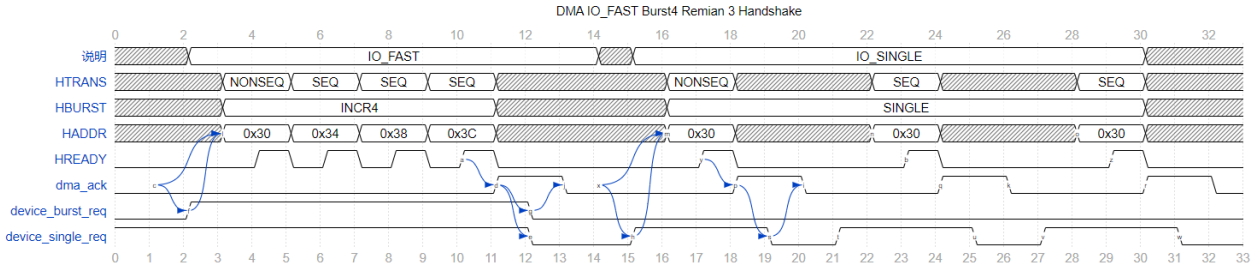


图 3–6 Type=IO_FAST 且数据长度与 Length 非对齐

3.3.2.4. 外设源端和终端传输模式

通过寄存器 DMA_CH_CTL_3 可以配置外设源端 (Source Mode Config) 和终端 (Destination Mode Config) 模式参数。

源端 (Source Mode Config) 和终端 (Destination Mode Config) 模式参数仅适用于外设，不适用于存储设备。

- 0x0： 使用握手模式，默认值

  推荐使用默认值。
- 0x1： 使用等待模式
- 0x2： 握手 + 等待模式

3.3.2.4.1. 使用握手模式

表 3–4 握手模式下外设源端和终端模式配置

	Burst Domain	Single Domain
使用方式	DMA 剩余传输数据量大于等于 Block Length	DMA 剩余传输数据量小于 Block Length
DMA 行为	会检测 REQ。 • 当检测到 device_burst_req，DMA 发起 transaction。 • 检测到总线最后一个 device 返回 ready，拉高 ack。 • 检测到 req 为低，拉低 ack。	会检测 REQ。 • 当检测到 device_single_req，DMA 发起 transaction。 • 检测到总线 device 返回 ready，拉高 ack。 • 检测到 req 为低，拉低 ack。
外设 FIFO	• 当外设 FIFO 到达水位并且 dma_ack 为低时，拉高 req。 • 当外设 FIFO 只要有数据（读）或空间（写）并且 ack 为底时，拉高 req。	

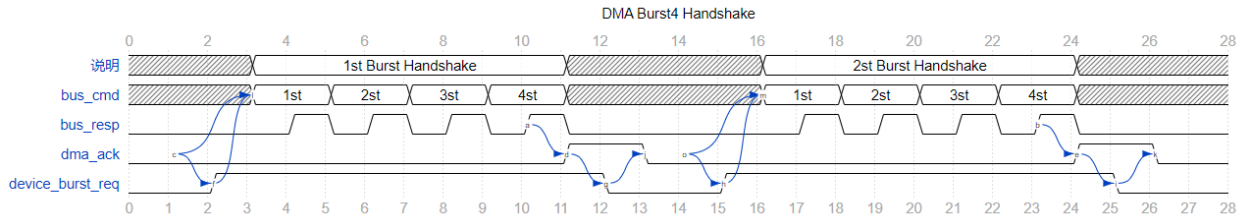


图 3–7 DMA Burst 4 Handshakes

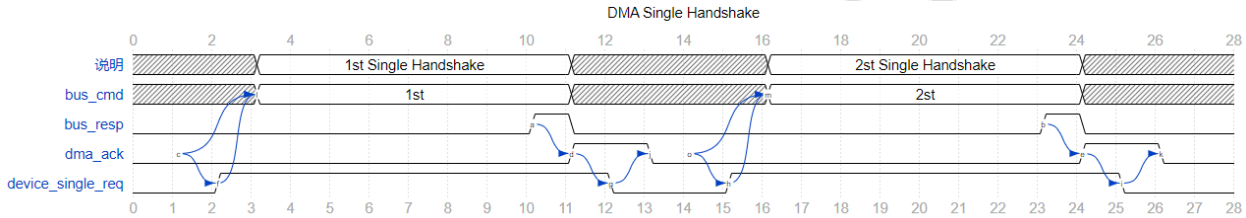


图 3–8 DMA Single Handshakes

3.3.2.4.2. 使用等待模式

表 3–5 等待模式下外设源端和终端模式配置

	Burst Domain	Single Domain
使用方式	DMA 剩余传输数据量大于等于 Block Length	DMA 剩余传输数据量小于 Block Length
DMA 行为	会检测 req。 • 当检测到 device_burst_req，DMA 发起 transaction。 • 检测到总线最后一个 device 返回 ready，拉高 ack。 • 延时 Wait*4 Cycle，拉低 ack。	会检测 req。 • 当检测到 device_single_req，DMA 发起 transaction。 • 检测到总线 device 返回 ready，拉高 ack。 • 延时 Wait*4 Cycle，拉低 ack。
外设 FIFO	• 当外设 FIFO 到达水位并且 dma_ack 为低时，拉高 req。 • 当外设 FIFO 只要有数据（读）或空间（写）并且 ack 为低时，拉高 req。	

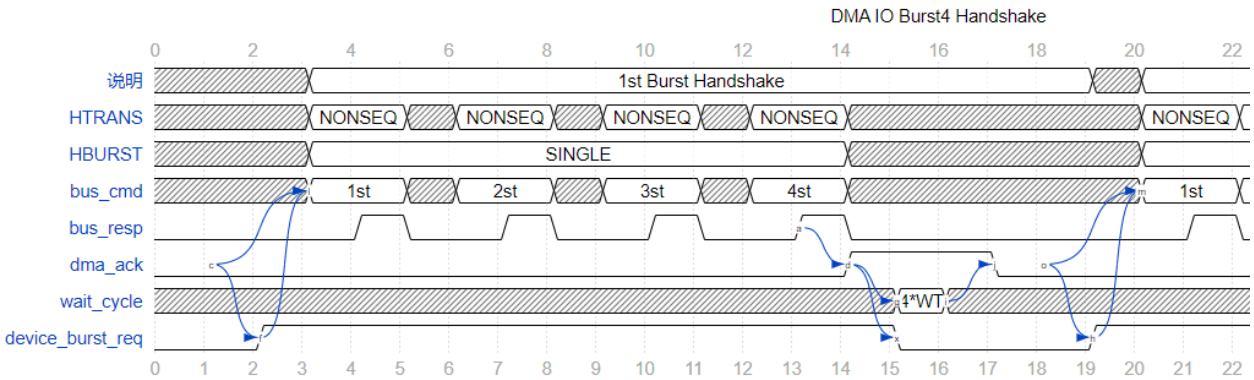


图 3–9 DMA Burst 4 Wait

3.3.2.4.3. 使用握手+等待模式

表 3–6 等待模式下外设源端和终端模式配置

	Burst Domain	Single Domain
使用方式	DMA 剩余传输数据量大于等于 Block Length	DMA 剩余传输数据量小于 Block Length
DMA 行为	会检测 req。 • 当检测到 device_burst_req，DMA 发起 transaction。 • 检测到总线最后一个 device 返回 ready，拉高 ack。 • 检测到 req 为低后，再延时 Wait*4 Cycle，拉低 ack。	会检测 req。 • 当检测到 device_single_req，DMA 发起 transaction。 • 检测到总线 device 返回 ready，拉高 ack。 • 检测到 req 为低之后，再延时 Wait*4 Cycle，拉低 ack。
外设 FIFO	• 当外设 FIFO 到达水位并且 dma_ack 为低时，拉高 req。 • 当外设 FIFO 只要有数据（读）或空间（写）并且 ack 为低时，拉高 req。	

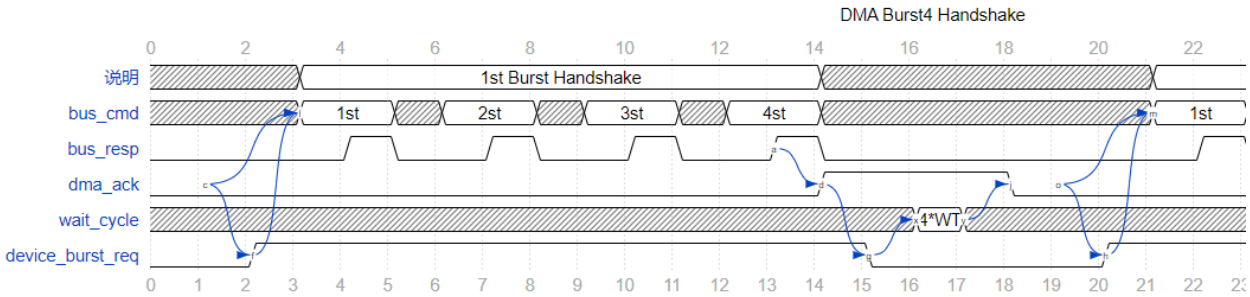


图 3–10 DMA Burst 4 Handshakes +Wait

3.3.2.5. 设备重要配置信息

本节描述了设备相关的重要配置信息，包括设备 ID、DMA 宽度以及 Type 类型。使用外设时，务必确保外设 ID 与地址设置正确，否则 DMA 通道可能卡住，从而无法产生错误状态。

表 3–7 设备重要配置信息

设备	ID	DMA_Width (bit)	Type
SRAM	0	32	Memory 或 Memset
PSRAM	1	32	Memory 或 Memset
CAN_FD0	4	8	IO_SINGLE 或 IO_BURST
CAN_FD1	5	8	IO_SINGLE 或 IO_BURST
PBUS	6	32	IO_SINGLE 或 IO_BURST

表 3–7 设备重要配置信息 (续)

设备	ID	DMA_Width (bit)	Type
CAN	7	32	IO_SINGLE 或 IO_BURST
SPI0	8	8、32	IO_SINGLE 或 IO_BURST
SPI1	9	8、32	IO_SINGLE 或 IO_BURST
SPI2	10	8、32	IO_SINGLE 或 IO_BURST
SPI3	11	8、32	IO_SINGLE 或 IO_BURST
SPI4	12	8、32	IO_SINGLE 或 IO_BURST
SPI5	13	8、32	IO_SINGLE 或 IO_BURST
UART0	16	8	IO_SINGLE 或 IO_BURST
UART1	17	8	IO_SINGLE 或 IO_BURST
UART2	18	8	IO_SINGLE 或 IO_BURST
UART3	19	8	IO_SINGLE 或 IO_BURST
UART4	20	8	IO_SINGLE 或 IO_BURST
UART5	21	8	IO_SINGLE 或 IO_BURST
UART6	22	8	IO_SINGLE 或 IO_BURST
UART7	23	8	IO_SINGLE 或 IO_BURST
UART8	24	8	IO_SINGLE 或 IO_BURST
ADC0_QUE1	32	32	IO_SINGLE 或 IO_BURST
ADC0_QUE2	33	32	IO_SINGLE 或 IO_BURST
ADC1_QUE1	34	32	IO_SINGLE 或 IO_BURST
ADC1_QUE2	35	32	IO_SINGLE 或 IO_BURST
SDFM0	36	32	IO_SINGLE 或 IO_BURST
SDFM1	37	32	IO_SINGLE 或 IO_BURST

表 3–7 设备重要配置信息 (续)

设备	ID	DMA_Width (bit)	Type
SDFM2	38	32	IO_SINGLE 或 IO_BURST
SDFM3	39	32	IO_SINGLE 或 IO_BURST
EPWM0	42	32	IO_SINGLE 或 IO_BURST
EPWM1	43	32	IO_SINGLE 或 IO_BURST
EPWM2	44	32	IO_SINGLE 或 IO_BURST
EPWM3	45	32	IO_SINGLE 或 IO_BURST
EPWM4	46	32	IO_SINGLE 或 IO_BURST
EPWM5	47	32	IO_SINGLE 或 IO_BURST
EPWM6	48	32	IO_SINGLE 或 IO_BURST
EPWM7	49	32	IO_SINGLE 或 IO_BURST
EPWM8	50	32	IO_SINGLE 或 IO_BURST
EPWM9	51	32	IO_SINGLE 或 IO_BURST
EPWM10	52	32	IO_SINGLE 或 IO_BURST
EPWM11	53	32	IO_SINGLE 或 IO_BURST
USB_FS0	58	32	IO_SINGLE 或 IO_BURST
USB_FS1	59	32	IO_SINGLE 或 IO_BURST
USB_FS2	60	32	IO_SINGLE 或 IO_BURST
USB_FS3	61	32	IO_SINGLE 或 IO_BURST
I2C0	72	16	IO_SINGLE 或 IO_BURST
I2C1	73	16	IO_SINGLE 或 IO_BURST
I2C2	74	16	IO_SINGLE 或 IO_BURST
I2C3	75	16	IO_SINGLE 或 IO_BURST

表 3–7 设备重要配置信息 (续)

设备	ID	DMA_Width (bit)	Type
PWM0	80	32	IO_SINGLE 或 IO_BURST
PWM1	81	32	IO_SINGLE 或 IO_BURST
PWM2	82	32	IO_SINGLE 或 IO_BURST
PWM3	83	32	IO_SINGLE 或 IO_BURST
PWM4	84	32	IO_SINGLE 或 IO_BURST
PWM5	85	32	IO_SINGLE 或 IO_BURST
CAP0	110	32	IO_SINGLE 或 IO_BURST
CAP1	111	32	IO_SINGLE 或 IO_BURST
CAP2	112	32	IO_SINGLE 或 IO_BURST
CAP3	113	32	IO_SINGLE 或 IO_BURST
CAP4	114	32	IO_SINGLE 或 IO_BURST
CAP5	115	32	IO_SINGLE 或 IO_BURST

3.3.3. 编程指南

注：

- 源端与终端地址需要 4Byte 对齐。
- 外设 WIDTH 设置需与外设模块寄存器位宽对齐。
- 外设 LENGTH 设置需要与 WIDTH 对齐。
- SPI Width 说明：DMA_Width=32，则 SPI.src_word_en=1。DMA_Width=8，则 SPI.src_word_en=0。需要配置 spi 模块内部 src_word_en 使用。
- UART Byte 传输说明：如 UART 需要实现 Byte 功能（即每接收到 1Byte 数据，马上写到内存中），则 Burst=1、Type=io_single、Block_length=1。

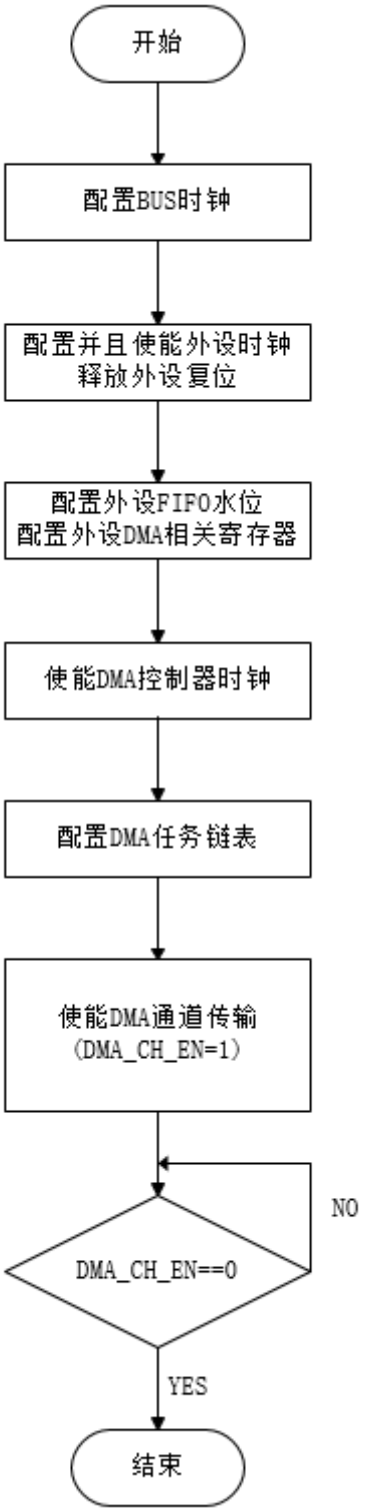


图 3–11 DMA 编程指南

3.3.3.1. 外设与外设传输

- **TASK_LENGTH**：需要与 max(src_width，dst_width)对齐。
- 外设模块内部水位：等于 **block_length**
- src_type=iob 与 dst_type=iob: `block_length=src_width*src_burst=dst_width*dst_burst`

- src_type=iob 与 dst_type=ios: `block_length=src_width*src_burst`
- src_type=ios 与 dst_type=iob: `block_length=dst_width*dst_burst`

以 SPI 为例，源端为 SPI0，终端为 SPI1。SPI 的 width 支持 8 或者 32。SPI 支持 burst 操作。

表 3–8 外设与外设传输编程示例

对象	参数	配置值例 1	配置值例 2	配置值例 3	配置值例 4	配置值例 5	配置值例 6
源端 SPI0	src_spi0						
	width	8	8	8	8	8	8
	burst	1	1	1	1	8	8
	type	ios	ios	ios	ios	iob (或 iof)	iob (或 iof)
终端 SPI1	dst_spi1						
	width	32	8	32	8	32	8
	burst	1	1	4	4	4	4
	type	ios	ios	iob (或 iof)	iob (者 iof)	iob (者 iof)	iob (或 iof)
要求	length 对齐要求	4Byte 对齐	1Byte 对齐	4Byte 对齐	1Byte 对齐	4Byte 对齐	1Byte 对齐
	block_length	4	1	16	4	16	8
	src_spi0 与 spi1 模块内部水位	4	1	16	4	16	8

3.3.3.2. DMA 编程说明

以下是以 SRAM 搬运数据到 SPI (spi0) 为例的 DMA 配置步骤：

1. 初始化参数。

表 3–9 DMA 编程初始化参数示例

源端	终端
SRAM （假设 src _width=32 ） • `src_width = 32` • `src_burst = 16` • `src_type = memory`	SPI (spi0, dma_id=8) • `dst_width = 32` • `dst_burst = 4` • `dst_type = io_burst`

2. 计算 Block_Length。

   由于对象包含外设并且 type=io_burst，则 block_length=32x4/8=16(Bytes)。

3. 初始化 SPI 进行，并且把 SPI 的水位设置为等于 length。此处设为 16，等于 length。
4. 使能 DMA 模块时钟、释放 DMA 复位并使能 DMA 中断。

5. 配置任务链表内容如下：

表 3–10 DMA 任务链表编程示例

地址	参数描述	备注
0xa1c8_6688	TASK_LINK_ID	TASK_LINK_ID 需要与 SET 一致。
0x2148_22c0	TASK_CFG_1	
0x0000_0010	BLOCK_LEN	
0x9004_7444	SRC_ADDR	源端地址：sram 首地址
0x913f_0200	DST_ADDR	终端：SPI0 TX_FIFO 地址
0x0000_00f4	TASK_LENGTH	
0x55ff_0000	TASK_CFG_2	使用推荐值 0x55ff_0000。
0xffff_fffc	下一 task 首地址	因为只有一个 task，故为 0xffff_fffc。

6. 把任务链表首地址（0x900406e0），写到 DMA 所对应通道的 DMA_CH_TASK_ADD_1 寄存器中。
7. 启动 DMA 传输，等待 DMA 传输完成中断。

3.3.4. 寄存器列表

表 3–11 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	DMA_IRQ_EN_SET_STATE_1	DMA 中断开启与开启状态 1	0x000 DMA_IRQ_EN_SET_STATE_1
0x004	DMA_IRQ_EN_SET_STATE_2	DMA 中断开启与开启状态 2	0x004 DMA_IRQ_EN_SET_STATE_2
0x020	DMA_IRQ_EN_CLR_1	DMA 中断关闭 1	0x020 DMA_IRQ_EN_CLR_1
0x024	DMA_IRQ_EN_CLR_2	DMA 中断关闭 2	0x024 DMA_IRQ_EN_CLR_2
0x040	DMA_IRQ_STA_1	DMA 中断状态 1	0x040 DMA_IRQ_STA_1
0x044	DMA_IRQ_STA_2	DMA 中断状态 2	0x044 DMA_IRQ_STA_2
0x54	DMA_BUS_ADDR_SEL	DMA 总线地址选择	0x54 DMA_BUS_ADDR_SEL
0x58	DMA_BUS_START_ADDR	DMA 总线起始地址	0x58 DMA_BUS_START_ADDR
0x5C+N*0x80	DMA_BUS_STOP_ADDR	DMA 总线结束地址	0x5C+N*0x80 DMA_BUS_STOP_ADDR
0x080	DMA_BUS_CFG	DMA 总线配置	0x080 DMA_BUS_CFG

表 3–11 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x088	DMA_LINK_ID_SET	DMA LINK ID 设置	0x088 DMA_LINK_ID_SET
0x090	DMA_FIFO_SIZE	DMA FIFO 大小	0x090 DMA_FIFO_SIZE
0x0a0	DMA_ALL_CH_STATE	DMA 所有通道状态	0x0a0 DMA_ALL_CH_STATE
0x100+N*0x80	DMA_CH_CTL_1	DMA 通道控制 1	0x100+N*0x80 DMA_CH_CTL_1
0x104+N*0x80	DMA_CH_CTL_2	DMA 通道控制 2	0x104+N*0x80 DMA_CH_CTL_2
0x108+N*0x80	DMA_CH_TASK_ADDR_1	DMA 通道任务地址 1	0x108+N*0x80 DMA_CH_TASK_ADDR_1
0x10c+N*0x80	DMA_CH_TASK_ADDR_2	DMA 通道任务地址 2	0x10c+N*0x80 DMA_CH_TASK_ADDR_2
0x110+N*0x80	DMA_CH_CTL_3	DMA 通道控制 3	0x110+N*0x80 DMA_CH_CTL_3
0x114+N*0x80	DMA_CH_CTL_4	DMA 通道控制 4	0x114+N*0x80 DMA_CH_CTL_4
0x118+N*0x80	DMA_MEM_SET_VAL	DMA 内存设置	0x118+N*0x80 DMA_MEM_SET_VAL
0x11c+N*0x80	DMA_TASK_BCNT	DMA 剩余传输字节数	0x11c+N*0x80 DMA_TASK_BCNT
0x120+N*0x80	DMA_LINK_ID	DMA 任务链表 ID	0x120+N*0x80 DMA_LINK_ID
0x124+N*0x80	DMA_TASK_CFG_1	DMA 任务配置	0x124+N*0x80 DMA_TASK_CFG_1
0x128+N*0x80	DMA_BLOCK_LEN	DMA 块长度	0x128+N*0x80 DMA_BLOCK_LEN
0x12c+N*0x80	DMA_SRC_ADDR	DMA 源端地址	0x12c+N*0x80 DMA_SRC_ADDR
0x130+N*0x80	DMA_DST_ADDR	DMA 终端地址	0x130+N*0x80 DMA_DST_ADDR
0x134+N*0x80	DMA_TASK_LEN	DMA 任务传输字节长度	0x134+N*0x80 DMA_TASK_LEN
0x138+N*0x80	DMA_TASK_CFG_2	DMA 已传任务数量	0x138+N*0x80 DMA_TASK_CFG_2
0x13c+N*0x80	DMA_NEXT_TASK_ADDR	DMA 下一任务地址	0x13c+N*0x80 DMA_NEXT_TASK_ADDR
0x140+N*0x80	DMA_SRC_WB_ADDR_SET	DMA 源端回写地址设置	0x140+N*0x80 DMA_SRC_WB_ADDR_SET
0x144+N*0x80	DMA_DST_WB_ADDR_SET	DMA 终端回写地址设置	0x144+N*0x80 DMA_DST_WB_ADDR_SET
0x148+N*0x80	DMA_SRC_WB_DATA	DMA 源端回写数据	0x148+N*0x80 DMA_SRC_WB_DATA

表 3–11 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x14c+N*0x80	DMA_DST_WB_DATA	DMA 终端回写数据	0x14c+N*0x80 DMA_DST_WB_DATA
0x150+N*0x80	DMA_TASK_CNT	DMA 已完成任务数	0x150+N*0x80 DMA_TASK_CNT
0x154+N*0x80	DMA_CFG_SEL	DMA 配置选择	0x154+N*0x80 DMA_CFG_SEL
0x160+N*0x80	DMA_DEBUG	DMA 调试（DMA DEBUG）	0x160+N*0x80 DMA_DEBUG
0x0fc	VERSION	版本号（VERSION）	0x0fc VERSION

3.3.5. 寄存器描述

注：

本节的变量 n=0~3。

3.3.5.1. 0x000 DMA_IRQ_EN_SET_STATE_1

重要：

对于本中断寄存器所有位域：

- 写 1 打开中断
- 写 0 忽略不作动作
- 关闭中断参考对应 DMA_IRQ_EN_CLR

默认值：0x00000000			DMA 中断开启与开启状态 1 (DMA Interrupt Request Set State 1)
位域	类型	默认值	描述
7+8*n	R/W1S	0x0	CHn_AXIWT_ILLEGAL_IRQ_SET 通道 AXI 写非法地址错误中断开启与中断开启状态
6+8*n	R/W1S	0x0	CHn_AXIRD_ILLEGAL_IRQ_SET 通道 AXI 读非法地址错误中断开启与中断开启状态
5+8*n	R/W1S	0x0	CHn_AHBWT_ILLEGAL_IRQ_SET 通道 AHB 写非法地址错误中断开启与中断开启状态

默认值：0x00000000			DMA 中断开启与开启状态 1 (DMA Interrupt Request Set State 1)
位域	类型	默认值	描述
4+8*n	R/W1S	0x0	CHn_AHBRD_ILLEGAL_IRQ_SET 通道 AHB 读非法地址错误中断开启与中断开启状态
3+8*n	R/W1S	0x0	CHn_LINKCFG_ERR_IRQ_SET 通道 LINK 地址错误与 LINK ID 错误中断开启与中断开启状态
2+8*n	R/W1S	0x0	CHn_LINK_FINISH_IRQ_SET 通道 n 完成链表中断开启与中断开启状态
1+8*n	R/W1S	0x0	CHn_TASK_FINISH_IRQ_SET 通道 n 完成一个任务中断开启与中断开启状态
0+8*n	R/W1S	0x0	CHn_TASK_HALF_IRQ_SET 通道 n 完成半个任务中断开启与中断开启状态

3.3.5.2. 0x004 DMA_IRQ_EN_SET_STATE_2

重要：

对于本中断寄存器所有位域：

- 写 1 打开中断
- 写 0 忽略不作动作
- 关闭中断参考对应 DMA_IRQ_EN_CLR

默认值：0x00000000			DMA 中断开启与开启状态 2(DMA Interrupt Request Set State 2)
位域	类型	默认值	描述
7+8*n	R/W1S	0x0	CH(n+4)_AXIWT_ILLEGAL_IRQ_SET 通道(n+4) AXI 写非法地址错误中断开启与中断开启状态
6+8*n	R/W1S	0x0	CH(n+4)_AXIRD_ILLEGAL_IRQ_SET 通道(n+4) AXI 读非法地址错误中断开启与中断开启状态
5+8*n	R/W1S	0x0	CH(n+4)_AHBWT_ILLEGAL_IRQ_SET 通道(n+4) AHB 写非法地址错误中断开启与中断开启状态
4+8*n	R/W1S	0x0	CH(n+4)_AHBRD_ILLEGAL_IRQ_SET

默认值：0x00000000			DMA 中断开启与开启状态 2(DMA Interrupt Request Set State 2)
位域	类型	默认值	描述
			通道(n+4) AHB 读非法地址错误中断开启与中断开启状态
3+8*n	R/W1S	0x0	CH(n+4)_LINKCFG_ERR_IRQ_SET 通道(n+4) LINK 地址错误与 ID 错误中断开启与中断开启状态
2+8*n	R/W1S	0x0	CH(n+4)_LINK_FINISH_IRQ_SET 通道(n+4)完成链表中断开启与中断开启状态
1+8*n	R/W1S	0x0	CH(n+4)_TASK_FINISH_IRQ_SET 通道(n+4)完成一个任务中断开启与中断开启状态
0+8*n	R/W1S	0x0	CH(n+4)_TASK_HALF_IRQ_SET 通道(n+4)完成半个任务中断开启与中断开启状态

3.3.5.3. 0x020 DMA_IRQ_EN_CLR_1

重要：

对于本中断寄存器所有位域：

- 写 1 关闭中断
- 写 0 忽略不作动作
- 打开中断参考对应 DMA_IRQ_EN_SET_STATE

默认值：0x00000000			DMA 中断关闭 1 (DMA Interrupt Request Clear 1)
位域	类型	默认值	描述
7+8*n	W1C	0x0	CHn_AXIWT_ILLEGAL_IRQ_CLR 通道 AXI 写非法地址错误中断关闭
6+8*n	W1C	0x0	CHn_AXIRD_ILLEGAL_IRQ_CLR 通道 AXI 读非法地址错误中断关闭
5+8*n	W1C	0x0	CHn_AHBWT_ILLEGAL_IRQ_CLR 通道 AHB 写非法地址错误中断关闭
4+8*n	W1C	0x0	CHn_AHBRD_ILLEGAL_IRQ_CLR 通道 AHB 读非法地址错误中断关闭

默认值：0x00000000			DMA 中断关闭 1 (DMA Interrupt Request Clear 1)
位域	类型	默认值	描述
3+8*n	W1C	0x0	CHn_LINKCFG_ERR_IRQ_CLR 通道 LINK 地址错误与 LINK ID 错误中断关闭
2+8*n	W1C	0x0	CHn_LINK_FINISH_IRQ_CLR 通道 n 完成链表中断关闭
1+8*n	W1C	0x0	CHn_TASK_FINISH_IRQ_CLR 通道 n 完成一个任务中断关闭
0+8*n	W1C	0x0	CHn_TASK_HALF_IRQ_CLR 通道 n 完成半个任务中断关闭

3.3.5.4. 0x024 DMA_IRQ_EN_CLR_2

重要：

对于本中断寄存器所有位域：

- 写 1 关闭中断
- 写 0 忽略不作动作
- 打开中断参考对应 DMA_IRQ_EN_SET_STATE

默认值：0x00000000			DMA 中断关闭 2(DMA Interrupt Request Clear 2)
位域	类型	默认值	描述
7+8*n	W1C	0x0	CH(n+4)_AXIWT_ILLEGAL_IRQ_CLR 通道(n+4) AXI 写非法地址错误中断关闭
6+8*n	W1C	0x0	CH(n+4)_AXIRD_ILLEGAL_IRQ_CLR 通道(n+4) AXI 读非法地址错误中断关闭
5+8*n	W1C	0x0	CH(n+4)_AHBWT_ILLEGAL_IRQ_CLR 通道(n+4) AHB 写非法地址错误中断关闭
4+8*n	W1C	0x0	CH(n+4)_AHBRD_ILLEGAL_IRQ_CLR 通道(n+4) AHB 读非法地址错误中断关闭
3+8*n	W1C	0x0	CH(n+4)_LINKCFG_ERR_IRQ_CLR

默认值：0x00000000			DMA 中断关闭 2(DMA Interrupt Request Clear 2)
位域	类型	默认值	描述
			通道(n+4) LINK 地址错误与 ID 错误中断关闭
2+8*n	W1C	0x0	CH(n+4)_LINK_FINISH_IRQ_CLR 通道(n+4)完成链表中断关闭
1+8*n	W1C	0x0	CH(n+4)_TASK_FINISH_IRQ_CLR 通道(n+4)完成一个任务中断关闭
0+8*n	W1C	0x0	CH(n+4)_TASK_HALF_IRQ_CLR 通道(n+4)完成半个任务中断关闭

3.3.5.5. 0x040 DMA_IRQ_STA_1

默认值：0x00000000			DMA 中断状态 1(DMA Interrupt States 1)
位域	类型	默认值	描述
7+8*n	R/W1C	0x0	CHn_AXIWT_ILLEGAL_STA 通道 AXI 写非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AXI 写指令，接收到对象返回的 ERROR，则产生中断状态。
6+8*n	R/W1C	0x0	CHn_AXIRD_ILLEGAL_STA 通道 AXI 读非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AXI 读指令，接收到对象返回的 ERROR，则产生中断状态。
5+8*n	R/W1C	0x0	CHn_AHBWT_ILLEGAL_STA 通道 AHB 写非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AHB 写指令，接收到对象返回的 ERROR，则产生中断状态。
4+8*n	R/W1C	0x0	CHn_AHBRD_ILLEGAL_STA 通道 AHB 读非法地址错误中断状态 0: 无中断 1: 有中断

默认值：0x00000000			DMA 中断状态 1(DMA Interrupt States 1)
位域	类型	默认值	描述
			DMA 发出 AHB 读指令，接收到对象返回的 ERROR，则产生中断状态。
3+8*n	R/W1C	0x0	CHn_TASKCFG_ERR_STA 通道 LINK 地址错误中断与 LINK ID 错误中断状态 0: 无中断 1: 有中断 Task 中，源端地址空间或终端地址空间存在跨 0x1_ffff_ffff 边界，则产生中断状态。 Task 中，设置的 ID 与 DMA_LINK_ID_SET 不一致，则产生中断状态。
2+8*n	R/W1C	0x0	CHn_LINK_FINISH_IRQ_STA 通道 n 完成链表中断状态 0: 无中断 1: 有中断 完成 LINK 中所有 Task 传输，则产生中断状态。
1+8*n	R/W1C	0x0	CHn_TASK_FINISH_IRQ_STA 通道 n 完成一个任务中断状态 0: 无中断 1: 有中断 每完成一个 Task 传输，则产生中断状态。
0+8*n	R/W1C	0x0	CHn_TASK_HALF_IRQ_STA 通道 n 完成半个任务中断状态 0: 无中断 1: 有中断 每完成半个 Task（传输剩余字节数 ≤1/2 Task Length）传输，则产生中断状态。

1：CHn_LINKID_ERR_STA 主要应用场景：TASK 未配置或 TASK 空间为非法空间，启动 DMA 后， DMA 先对 LINKID 进行对比校验，不一致产生错误提示，停止工作。正常情况下，TASK 空间有效，并且 TASK 正确设置，不会产生 _LINKID_ERR_STA 中断。

2：当 DMA 进入 CHn_TASK_ADDR_ERR_STA 或 CHn_LINKID_ERR_STA 状态下， DMA 会停止(DMA 会自动回到 Idle 状态 不需要软件介入)，产生中断状态。

3：Bus_stop_en=0。非 DDR 设备需要添加附加条件 AHB_resp_en=0。当 DMA 进入 CHn_AXI_ILLEGAL_STA 或 STA ，DMA 会继续完成传输，产生中断状态（推荐 enginer_clr_en=0）。

4：Bus_stop_en=0,AHB_resp_en=1。当 DMA 进入 STA ，DMA 会停止传输，产生中断状态，此时需要软件进行 DMA 通道复位(enginer_clr_en=1,dma_ch_en=0)。DDR 设备没有软件复位机制

5：Bus_stop_en=1,非 DDR 设备需要添加附加条件 AHB_resp_en=0。当 DMA 进入 CHn_AXI_ILLEGAL_STA,CHn_AHB_ILLEGAL_STA ，DMA 会停止传输，产生中断状态，DMA 通道会自动进行复位

3.3.5.6. 0x044 DMA_IRQ_STA_2

默认值：0x00000000			DMA 中断状态 2(DMA Interrupt States 2)
位域	类型	默认值	描述
7+8*n	R/W1C	0x0	CH(n+4)_AXIWT_ILLEGAL_STA 通道(n+4) AXI 写非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AXI 写指令，接收到对象返回的 ERROR，则产生中断状态
6+8*n	R/W1C	0x0	CH(n+4)_AXIRD_ILLEGAL_STA 通道(n+4) AXI 读非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AXI 读指令，接收到对象返回的 ERROR，则产生中断状态
5+8*n	R/W1C	0x0	CH(n+4)_AHBWT_ILLEGAL_STA 通道(n+4) AHB 写非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AHB 写指令，接收到对象返回的 ERROR，则产生中断状态
4+8*n	R/W1C	0x0	CH(n+4)_AHBRD_ILLEGAL_STA 通道(n+4) AHB 读非法地址错误中断状态 0: 无中断 1: 有中断 DMA 发出 AHB 读指令，接收到对象返回的 ERROR，则产生中断状态
3+8*n	R/W1C	0x0	CH(n+4)_TASKCFG_ERR_STA 通道(n+4) LINK 地址错误中断与 ID 错误中断状态 0: 无中断 1: 有中断 Task 中，源端地址空间或终端地址空间存在跨 0x1_ffff_ffff 边界，则产生中断状态 Task 中，设置的 ID 与 DMA_LINK_ID_SET 不一致，则产生中断状态
2+8*n	R/W1C	0x0	CH(n+4)_LINK_FINISH_IRQ_STA 通道(n+4)完成链表中断状态 0: 无中断 1: 有中断

默认值：0x00000000			DMA 中断状态 2(DMA Interrupt States 2)
位域	类型	默认值	描述
			完成 LINK 中所有 Task 传输，则产生中断状态
1+8*n	R/W1C	0x0	CH(n+4)_TASK_FINISH_IRQ_STA 通道(n+4)完成一个任务中断状态 0: 无中断 1: 有中断 每完成一个 Task 传输，则产生中断状态
0+8*n	R/W1C	0x0	CH(n+4)_TASK_HALF_IRQ_STA 通道(n+4)完成半个任务中断状态 0: 无中断 1: 有中断 每完成半个 Task（传输剩余字节数<=1/2 Task Length）传输，则产生中断状态

1：CHn_LINKID_ERR_STA 主要应用场景：TASK 未配置或 TASK 空间为非法空间，启动 DMA 后， DMA 先对 LINKID 进行对比校验，不一致产生错误提示，停止工作。正常情况下，TASK 空间有效，并且 TASK 正确设置，不会产生 _LINKID_ERR_STA 中断。

2：当 DMA 进入 CHn_TASK_ADDR_ERR_STA 或 CHn_LINKID_ERR_STA 状态下， DMA 会停止(DMA 会自动回到 Idle 状态 不需要软件介入)，产生中断状态。

3：Bus_stop_en=0。非 DDR 设备需要添加附加条件 AHB_resp_en=0。当 DMA 进入 CHn_AXI_ILLEGAL_STA 或 STA ，DMA 会继续完成传输，产生中断状态（推荐 enginer_clr_en=0）。

4：Bus_stop_en=0,AHB_resp_en=1。当 DMA 进入 STA ，DMA 会停止传输，产生中断状态，此时需要软件进行 DMA 通道复位(enginer_clr_en=1,dma_ch_en=0)。DDR 设备没有软件复位机制

5：Bus_stop_en=1,非 DDR 设备需要添加附加条件 AHB_resp_en=0。当 DMA 进入 CHn_AXI_ILLEGAL_STA,CHn_AHB_ILLEGAL_STA ，DMA 会停止传输，产生中断状态，DMA 通道会自动进行复位

3.3.5.7. 0x54 DMA_BUS_ADDR_SEL

默认值：0x00000001			DMA 总线地址选择(DMA BUS ADDR SELECT)
位域	类型	默认值	描述
31:1	_	_	_
0	R/W	0x1	DMA 总线地址选择： 0：DMA BUS START ADDR and STOP ADDR 设置是 AHB 范围，在地址范围内都会采用 AHB 访问（ START ADDR=< 范围 <STOP ADDR ） 1：DMA BUS START ADDR and STOP ADDR 设置是 AXI 范围，在地址范围内都会采用 AXI 访问（ START ADDR=< 范围 <STOP ADDR ）

3.3.5.8. 0x58 DMA_BUS_START_ADDR

默认值：0x00030000			DMA 总线起始地址(DMA BUS START ADDR)
位域	类型	默认值	描述
31:28	_	_	_
27:0	R/W	0x30000	DMA 总线起始地址(单位 4KBytes)

3.3.5.9. 0x5C+N*0x80 DMA_BUS_STOP_ADDR

默认值：0x000fffff			DMA 总线结束地址(DMA BUS STOP ADDR)
位域	类型	默认值	描述
31:28	_	_	_
27:0	R/W	0xfffff	DMA 总线结束地址(单位 4KBytes)

3.3.5.10. 0x080 DMA_BUS_CFG

默认值：0x00000000			DMA 总线配置(DMA BUS Config)
位域	类型	默认值	描述
31:7	_	_	_
6	R/W	0x0	AHB_hprot_auto_en 0: hprot 为 CFG_2 里面配置值 1: hprot 前面的 transaction 为 bufferable,最后一笔为 unbufferable
5	R/W	0x0	AHB_resp_enable 0: AHB 不需要等待 Signal 返回 Bus_stop_en=0:访问非法地址，会产生非法访问错误状态，DMA 通道会继续完成传输 Bus_stop_en=1:访问非法地址，会产生非法访问错误状态，DMA 通道会停止，自动复位通道。 1: AHB 需要等待 Signal 返回 访问非法地址，会产生非法访问错误状态，DMA 通道会停止。通过 Enable engine_clr_en, 并且关闭 EN，来对通道进行复位。 推荐使用复位值

默认值：0x00000000			DMA 总线配置(DMA BUS Config)
位域	类型	默认值	描述
4	–	–	–
3	R/W	0x0	engine_clr_en 0: Disable 1: Enable
2:0	–	–	–

3.3.5.11. 0x088 DMA_LINK_ID_SET

默认值：0xa1c86688			DMA LINK ID 设置(DMA LINK ID SET)
位域	类型	默认值	描述
31:0	R/W	0xa1c86688	LINK ID 设置，任务链表中 ID 需与此处设置值一致 每个通道中 task 内的 LINK ID 需要与此处一致，否则会产生 LIND_ID_ERROR ,DMA 无法正常工作。（不同通道中，不同 Task 的 LINK ID 是一样的，需要配置与此处值一致）

3.3.5.12. 0x090 DMA_FIFO_SIZE

默认值：0x80			DMA FIFO 大小(DMA FIFO SIZE)
位域	类型	默认值	描述
31:0	R	0x80	DMA 控制器 FIFO 大小（每个通道独立 FIFO 的大小） FIFO_SIZE = Wide x Len (单位 Bytes, Len=32)

3.3.5.13. 0x0a0 DMA_ALL_CH_STATE

默认值：0x00000000			DMA 所有通道状态(DMA All Channel States)
位域	类型	默认值	描述
31:0	R	0x00000000	CH(n) State （每个 bit 分别对应一个通道） 0: 空闲 1: 忙碌

3.3.5.14. 0x100+N*0x80 DMA_CH_CTL_1

默认值：0x00000000			DMA 通道控制 1(DMA Channel Control 1)
位域	类型	默认值	描述
31:5	–	–	–
4	R/WAC	0x0	DMA_CH_REEN DMA 重新使能 0：关闭 1：使能 当 DMA 正在传输时，使能 DMA_CH_REEN，控制器会中断当前正在传输的任务。重新获取描述符，开始新的传输，传输完成会自动清 0
3:1	–	–	–
0	R/WAC	0x0	DMA_CH_EN DMA 通道使能 0: 关闭 1: 打开 注意 DMA 传输完成后，该位会自动清零，可以通过读取该位表示 DMA 当前状态。每个 DMA 通道执行顺序，由 DMA 控制器自动决定。

3.3.5.15. 0x104+N*0x80 DMA_CH_CTL_2

默认值：0x00000000			DMA 通道控制 2(DMA Channel Control 2)
位域	类型	默认值	描述
31:3	–	–	–
2	R/WAC	0x0	CH_ABANDON 通道链表丢弃 0:通道链表丢弃完成 1:通道链表丢弃使能 DMA 立即停止（不需要等待 task 完成），并且丢弃所有未完成 Task 传输 DMA 丢弃完成之后，才能关闭其它设备。
1	R/W	0x0	CHn_TASK_PAUSE 通道任务暂停 0: 继续 1: 通道任务暂停使能

默认值：0x00000000			DMA 通道控制 2(DMA Channel Control 2)
位域	类型	默认值	描述
			DMA 完成当前 task 之后，才会暂停
0	R/W	0x0	CHn_PAUSE 通道链表暂停 0: 继续 1: 通道链表暂停使能 DMA 立即暂停（不需要等待 task 完成）

3.3.5.16. 0x108+N*0x80 DMA_CH_TASK_ADDR_1

默认值：0x00000000			DMA 通道任务地址 1(DMA Channel Task Addr 1)
位域	类型	默认值	描述
31:0	R/W	0x0	DMA_TASK_ADDR DMA 任务低 32bit 地址

3.3.5.17. 0x10c+N*0x80 DMA_CH_TASK_ADDR_2

默认值：0x00000000			DMA 通道任务地址 2(DMA Channel Task Addr 2)
位域	类型	默认值	描述
31:8	_	_	_
7:0	R/W	0x0	DMA_TASK_ADDR DMA 任务高 8 位地址

3.3.5.18. 0x110+N*0x80 DMA_CH_CTL_3

默认值：0x00000000			DMA 通道控制 3(DMA Channel Control 3)
位域	类型	默认值	描述
31:24	_	_	_
23:20	R/W	0x0	DMA 通道 源端阈值配置 如 Priority Config disable,则到达一定阈值，优先级自动+1 当 DMA 已进行了 n 次 AXI 读行为（n=AXI 读阈值配置,并且 n 次中无该通道）

默认值：0x00000000			DMA 通道控制 3(DMA Channel Control 3)
位域	类型	默认值	描述
			则 DMA 内部会对 Source Fixed Priority Config 或 Source Fixed Priority Config +1，一直到 Max，并且保持。如果期间该通道有被授予执行，则优先级恢复到 Config 或 Source Fixed Priority Config 所配置值。
19:16	R/W	0x0	DMA 通道终端阈值配置 如 Priority Config disable,则到达一定阈值，优先级自动+1 当 DMA 已进行了 n 次 AXI 写行为（n=AXI 写阈值配置,并且 n 次中无该通道） 则 DMA 内部会对 Source Fixed Priority Config 或 Source Fixed Priority Config +1，一直到 Max，并且保持。如果期间该通道有被授予执行，则优先级恢复到 Config 或 Source Fixed Priority Config 所配置值。
15	_	_	_
14:13	R/W	0x0	task_bcnt_sel DMA 剩余 Byte 数据（task_bcnt）功能选择 0x0:终端剩余字节数 0x1:源端剩余字节数 0x2:终端剩余字节数 0x3:源端剩余字节数
12	R/W	0x0	DMA FIX WORD EN 0:disable 1:enable 只有 Type=IO_FAST，并且 Width!=32,根据外设支持情况，才会 enable。 例如 Type=IO_FAST,Width=8。FIX WORD EN=0， 则 ADDR 按+1 累加,并且 AHB BUS HSIZE=0（8 bits）,传输次数=block length。 FIX WORD EN=1，则 ADDR 按+4 累加(0/4/8/c), 并且 AHB BUS HSIZE=2(32 bits),有效 data=8,传输次数=block length。
11：10	_	_	_
9	R/W	0x0	DMA 通道源端 Priority Config Enable 0: disable (到一定阈值，通道优先级自动提升功能开启) 1: enable (通道固定优先级参与 DMA 内部通道之间仲裁)
8	R/W	0x0	DMA 通道终端 Priority Config Enable 0: disable (到一定阈值，通道优先级自动提升功能开启) 1: enable (通道固定优先级参与 DMA 内部通道之间仲裁)

默认值：0x00000000			DMA 通道控制 3(DMA Channel Control 3)
位域	类型	默认值	描述
7	R/W	0x0	Bus Stop En 0: disable Bus Stop En=0：AHB 与 AXI 对象返回 Error，则 DMA 的行为由 AHB_resp_enable 或 enable 决定 1: enable Bus Stop En=1：AHB 与 AXI 对象返回 Error，则 DMA 自动停止，并且自动回到 Idle 状态（不需要软件介入）
6:5	R/W	0x0	DMA 通道内部 FIFO Priority 0x0:读优先级高 0x1:写优先级高 0x2:读写优先级循环 推荐使用默认值
4	R/W	0x0	CHn Write Back Enable DMA 通道回写使能 0: disable 1: enable 需要配合 Enable。只有同时有效，回写才会执行。 DMA 每完成一个 task 传输之后， 会自动读取 SET 与 DMA_SRC_WB_ADDR_SET 所配地址， 把该地址内容写到 Task 所对应 TASKn.DST_WB_DATA 与 DATA 空间，并且更新到通道寄存器 DMA_DST_WB_DATA 与 DATA。 只能同时使能 WB 与 WB。 不能单独使能 WB 或 WB。
3:2	R/W	0x0	Destination Mode Config 0x0:使用握手模式 0x1:使用等待模式 0x2:握手+等待模式 推荐使用默认值（此参数仅对外设有效，储存无效），具体区别参考功能描述.握手
1:0	R/W	0x0	Source Mode Config 0x0:使用握手模式 0x1:使用等待模式 0x2:握手+等待模式 推荐使用默认值（此参数仅对外设有效，储存无效），具体区别参考功能描述.握手

3.3.5.19. 0x114+N*0x80 DMA_CH_CTL_4

默认值：0x00000000			DMA 通道控制 4(DMA Channel Control 4)
位域	类型	默认值	描述
31:24	R/W	0x0	Destination Wait Cycle Set 当 Destination Mode Config=0x1,DMA 完成一笔传输之后，等待 Wait Cycle*4 周期之后，才重新开始一笔传输。 当 Destination Mode Config=0x2,DMA 完成一笔传输之后，等待握手完成之后，再等待 Wait Cycle*4 周期之后，才重新开始一笔传输。
23:16	R/W	0x0	Source Wait Cycle Set 当 Source Mode Config=0x1,DMA 完成一笔传输之后，等待 Wait Cycle*4 周期之后，才重新开始一笔传输。 当 Source Mode Config=0x2,DMA 完成一笔传输之后，等待握手完成之后，再等待 Wait Cycle*4 周期之后，才重新开始一笔传输。
15:14	_	_	_
13	R/W	0x0	fsm_frc_resp_memset 仅 debug 使用 0：memset 只有最后一笔等待 response 1：memset 每一笔传输都等待 Response
12	R/W	0x0	fsm_frc_resp_mem 仅 debug 使用 0：Memory 只有最后一笔等待 response，默认值 1：Memory 每一笔传输都等待 Response **注：** DMA UART 实现 Byte 模式，需要置 1，其它情况下，统一使用默认值 0。
11:9	_	_	_
8	R/W	0x0	fsm_bps_resp_wb 仅 debug 使用 0：回写使用 response 机制 1：回写屏蔽 wait response 机制
7	R/W	0x0	fsm_bps_resp_memset 仅 debug 使用 0：MemorySet 使用 response 机制 1：MemorySet 屏蔽 wait response 机制
6	R/W	0x0	fsm_bps_resp_mem 仅 debug 使用 0：Memory 使用 response 机制

默认值：0x00000000			DMA 通道控制 4(DMA Channel Control 4)
位域	类型	默认值	描述
			1：Memory 屏蔽 wait response 机制
5	R/W	0x0	fsm_bps_resp_iob_iof 仅 debug 使用 0：iob 与 iof 使用 wait response 机制 1：iob 与 iof 屏蔽 wait response 机制
4	R/W	0x0	fsm_bps_resp_ios 仅 debug 使用 0：ios 使用 response 机制 1：ios 屏蔽 wait response 机制
3:2	R/W	0x0	Destination Fixed Priority Config Destination 请求仲裁固定优先级配置 如 Priority Config enable,则通道以此处配置优先级申请请求。
1:0	R/W	0x0	Source Fixed Priority Config Source 请求仲裁固定优先级配置 如 Priority Config enable,则通道以此处配置优先级申请请求。

3.3.5.20. 0x118+N*0x80 DMA_MEM_SET_VAL

默认值：0x76543210			DMA 内存设置(DMA Memory Set Value)
位域	类型	默认值	描述
31:0	R/W	0x76543210	Memory Set 内容配置 当 Type=memoryset，DMA 会直接把该寄存器的值，写入到终端地址空间， 而不需要从源端读取。

3.3.5.21. 0x11c+N*0x80 DMA_TASK_BCNT

默认值：0x00000000			DMA 剩余传输字节数(DMA Task Byte Counter)
位域	类型	默认值	描述
31:25	–	–	–
24:0	R	0x0	DMA 任务的终端传输开始（或源端传输开始）剩余字节数 终端或源端的选择由 sel 进行决定

3.3.5.22. 0x120+N*0x80 DMA_LINK_ID

默认值：0x00000000			DMA 任务链表 ID(DMA Task Link ID)
位域	类型	默认值	描述
31:0	R	0x0	当前任务链表中的 LINK ID

3.3.5.23. 0x124+N*0x80 DMA_TASK_CFG_1

默认值：0x00000000			DMA 任务配置 (DMA Task Config 1)
位域	类型	默认值	描述
31	R	0x0	Destination Security
30:28	R	0x0	Destination data Width DMA 终端数据位宽（总线上有效数据的位宽） • 0x0: 8 位 • 0x1: 16 位 • 0x2: 32 位 • 0x3: 64 位 • 0x4: 128 位 根据 设备重要配置信息进行选择。
27:25	R	0x0	Destination Type • 0x0: io_single (需要握手) • 0x1: io_burst (需要握手) • 0x2: memory (不需要握手) • 0x3:memoryset (不需要握手) • 0x4:io_fast (需要握手) 关于不同设备的具体传输类型配置，可查看设备源端和终端传输类型。
24:23	R	0x0	Destination Burst DMA 终端突发数 • 0x0: 1 • 0x1: 4 • 0x2: 8 • 0x3: 16 当 Type=io_burst 或 fast 时，Burst 只能为 4, 8 或 16

默认值：0x00000000			DMA 任务配置 (DMA Task Config 1)
位域	类型	默认值	描述
			当 Type=io_single 时，Burst 只能为 1 当 Type=memory 或 memoryset，Burst 推荐为 16
22:16	R	0x0	Destination Device ID 根据 设备重要配置信息进行选择。
15	R	0x0	Source Security
14:12	R	0x0	Source data Width DMA 源端数据位宽（总线上有效数据的位宽） 0x0: 8 位 0x1: 16 位 0x2: 32 位 0x3: 64 位 0x4: 128 位 根据 设备重要配置信息进行选择。
11:9	R	0x0	Source Type 0x0:io_single (需要握手) 0x1:io_burst (需要握手) 0x2:memory (不需要握手) 0x3:memoryset (不需要握手) 0x4:io_fast(需要握手) 关于不同设备的具体传输类型配置，可查看设备源端和终端传输类型。
8:7	R	0x0	Source Burst DMA 源端突发数 0x0: 1 0x1: 4 0x2: 8 0x3: 16 当 Type=io_burst 或 fast 时，Burst 只能为 4, 8 或 16 当 Type=io_single 时，Burst 只能为 1 当 Type=memory 或 memoryset，Burst 推荐为 16
6:0	R	0x0	Source Device ID DMA 源端设备 ID 根据 设备重要配置信息进行选择。

3.3.5.24. 0x128+N*0x80 DMA_BLOCK_LEN

默认值：0x00000000			DMA 块长度(DMA Block Length)
位域	类型	默认值	描述
31:16	_	_	_
15:0	R	0x0	Block Length 块长度，单位 Byte • 当设备为存储模块时，推荐：`DMA_BLOCK_LEN = DMA_FIFO_SIZE/2` • 当设备为非存储模块时，推荐：`DEVICE_FIFO_LEVEL = DMA_BLOCK_LEN` • 当 Type = io_burst 或 fast 时，`DMA_BLOCK_LEN = Width * Burst / 8` • 当 Type = io_single 时，`DMA_BLOCK_LEN = Width * 1 / 8`

3.3.5.25. 0x12c+N*0x80 DMA_SRC_ADDR

默认值：0x00000000			DMA 源端地址 (DMA Source Addr)
位域	类型	默认值	描述
31:0	R	0x0	DMA 任务源端地址低 32 bits[1:0]=2'b00 (4byte 对齐)

3.3.5.26. 0x130+N*0x80 DMA_DST_ADDR

默认值：0x00000000			DMA 终端地址 (DMA Destination Addr)
位域	类型	默认值	描述
31:0	R	0x0	DMA 任务终端地址低 32 bits[1:0]=2'b00(4byte 对齐)

3.3.5.27. 0x134+N*0x80 DMA_TASK_LEN

默认值：0x00000000			DMA 任务传输字节长度 (DMA TASK LENGTH)
位域	类型	默认值	描述
31	R	0x0	Task Write Back Enable 0: Disable 1: Enable 要与 Enable 配合使用。只有同时有效，回写才会执行。

默认值：0x00000000			DMA 任务传输字节长度 (DMA TASK LENGTH)
位域	类型	默认值	描述
			DMA 每完成一个 task 传输之后，会自动读取 SET 与 DMA_SRC_WB_ADDR_SET 所配地址，把该地址内容写到 Task 所对应 TASKn.DST_WB_DATA 与 DATA 空间，并且更新到通道寄存器 DMA_DST_WB_DATA 与 DATA。 只能同时使能 WB 与 WB。 不能单独使能 WB 或 WB。
30	–	–	–
29:27	R	0x0	DMA 总线 qos,对应 AXI 接口 qos 信号
26	R	0x0	0：当终端 Type=io_fast,地址按照 width 进行累加跳转。 1：当终端 Type=io_fast，并且 FIX WORD EN=enable，地址按照 word 进行累加(0/4/8/C)跳转。
25	R	0x0	0：当源端 Type=io_fast,地址按照 width 进行累加跳转 1：当源端 Type=io_fast，并且 FIX WORD EN=enable，地址按照 word 进行累加(0/4/8/C)跳转。
24:0	R	0x0	DMA 当前 TASK 需要传输的数据长度单位 Bytes

3.3.5.28. 0x138+N*0x80 DMA_TASK_CFG_2

默认值：0x00000000			DMA 已传任务数量 (DMA TASK CONFIG 2)
位域	类型	默认值	描述
31:28	R	0x0	终端访问总线 hprot,对应 AHB 接口 hprot 信号
27:24	R	0x0	源端访问总线 hprot,对应 AHB 接口 hprot 信号
23:16	R	0x0	下一任务高 8bit 地址
15:8	R	0x0	终端高 8bit 起始地址
7:0	R	0x0	源端高 8bit 起始地址

3.3.5.29. 0x13c+N*0x80 DMA_NEXT_TASK_ADDR

默认值：0x00000000			DMA 下一任务地址 (Next Task Addr)
位域	类型	默认值	描述
31:0	R	0x0	下一任务低 32 bits 地址

3.3.5.30. 0x140+N*0x80 DMA_SRC_WB_ADDR_SET

默认值：0x00000000			DMA 源端回写地址设置 (Source Write Back Addr Set)
位域	类型	默认值	描述
31:0	R/W	0x0	SRC_WB_ADDR_SET 读取源端 Device 状态地址的设置 任务完成之后，DMA 会读取该地址，然后把该地址里面数据回写到链表空间 ,并且把数据更新到 DATA 寄存器中。

3.3.5.31. 0x144+N*0x80 DMA_DST_WB_ADDR_SET

默认值：0x00000000			DMA 终端回写地址设置 (Destination Write Back Addr Set)
位域	类型	默认值	描述
31:0	R/W	0x0	DST_WB_ADDR_SET 读取终端 Device 状态地址的设置 任务完成之后，DMA 会读取该地址，然后把该地址里面数据回写到链表空间 ,并且把数据更新到 DATA 寄存器中。

3.3.5.32. 0x148+N*0x80 DMA_SRC_WB_DATA

默认值：0x00000001			DMA 源端回写数据 (Source Write Back Data)
位域	类型	默认值	描述
31:0	R	0x1	SRC_WB_DATA 源端 Device 状态(回写的数据)

3.3.5.33. 0x14c+N*0x80 DMA_DST_WB_DATA

默认值：0x00000001			DMA 终端回写数据 (Destination Write Back Data)
位域	类型	默认值	描述
31:0	R	0x1	DST_WB_DATA 终端 Device 状态（回写的数据）

3.3.5.34. 0x150+N*0x80 DMA_TASK_CNT

默认值：0x00000000			DMA 已完成任务数 (DMA Task Counter)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	DMA 已完成任务数

3.3.5.35. 0x154+N*0x80 DMA_CFG_SEL

默认值：0x00000000			DMA 配置选择 (DMA CONFIG SELECT)
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	任务地址更新使能 0：关闭（保持初始值） 1：使能（更新为当前 task 的地址）
3:2	–	–	–
1	R/W	0x0	LINK 地址选择 0：选择从 memory 获取描述符 1：选择从寄存器获取描述符 回环功能：NEXT_TASK_ADDR=DMA_LINK_ID 的地址
0	–	–	–

3.3.5.36. 0x160+N*0x80 DMA_DEBUG

默认值：0x00000000			DMA 调试 (DMA DEBUG)
位域	类型	默认值	描述
31:26	_	_	_
25	R	0x0	终端 idle(1:idle 0:busy)
24	R	0x0	源端 idle(1:idle 0:busy)
23	_	_	_
22	R/W1C	0x0	Link_地址错误，高有效
21	R/W1C	0x0	终端状态机错误，高有效
20	R/W1C	0x0	源端状态机错误，高有效
19	R/W1C	0x0	LINK ID 错误，高有效
18	R/W1C	0x0	半个任务完成，高有效
17	R/W1C	0x0	任务完成，高有效
16	R/W1C	0x0	链表完成，高有效
15:14	_	_	_
13:8	R	0x0	终端状态机的状态
7:6	_	_	_
5:0	R	0x0	源端状态机的状态

3.3.5.37. 0x0fc VERSION

默认值：0x00000200			版本号 (VERSION)
位域	类型	默认值	描述
31:0	R	0x00000200	Version(模块版本号) 采用 BCD 码显示，V2.0

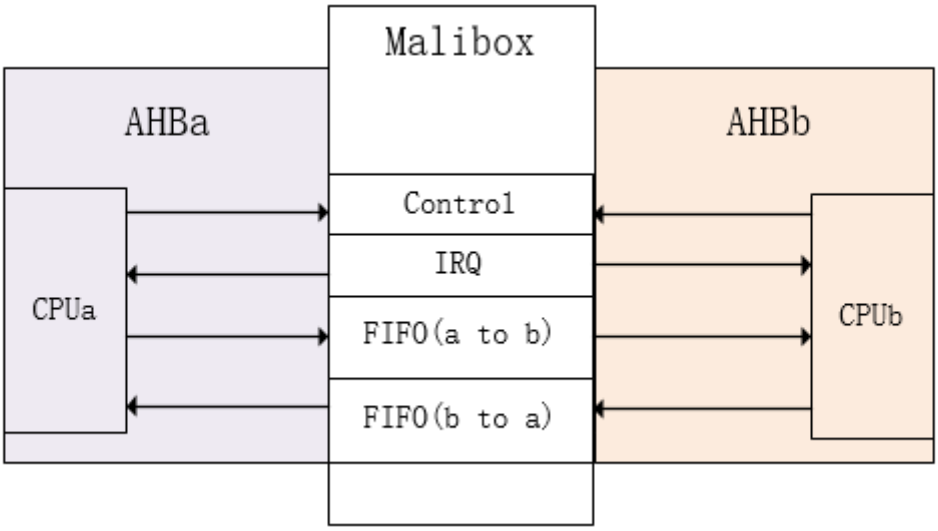
3.4. Mailbox (MBOX)

Mailbox (MBOX) 用于 CPU 间通信，可以作为发送者也可以是接收者。

3.4.1. 特性说明

- 支持 CPU 核读/写通道通信
- 支持每个 CPU 核独立的中断
- 32x32bit FIFO x2

3.4.2. 原理框图



对于 CPUa 来说，有其自己地址空间，同样 CPUb 也有其地址空间，因此，CPUa 以及 CPUb 需要独立配置相关中断使能以及 FIFO 设置等。

需要注意的是，虽然两个 CPU 所访问地址是不一样的，但 CPUa 的 TX FIFO 与 CPUb 的 RX FIFO 为同一个，反之亦然。

3.4.3. 功能描述

3.4.3.1. 发送数据

作为发送者时，TX_FIFO_FULL_EN 配置为 1，当 FIFO 达到 TX_FIFO_LEVEL 配置的水位，则 FULL 状态触发，产生 FIFO 为满中断信号，则停止进行写操作。每次写操作为一个 32bit message。TX_FIFO 为满，若进行写操作则会数据上溢出造成数据丢失。上溢出时数据覆盖或禁止写入可由 CTL 进行配置选择。

假设单次发送有 N 个 messages，当 N=1 时，只需要往 WCOMP 寄存器写 message 即可完成发送并且产生中断。当 N>1 时，需要往 WMSG 寄存器写 N–1 个 messages，往 WCOMP 寄存器写进最后一个 message 完成发送并且产生中断。

CPUa 作为发送方时 图 3–12 : CPUa 发送数据 示意图如下

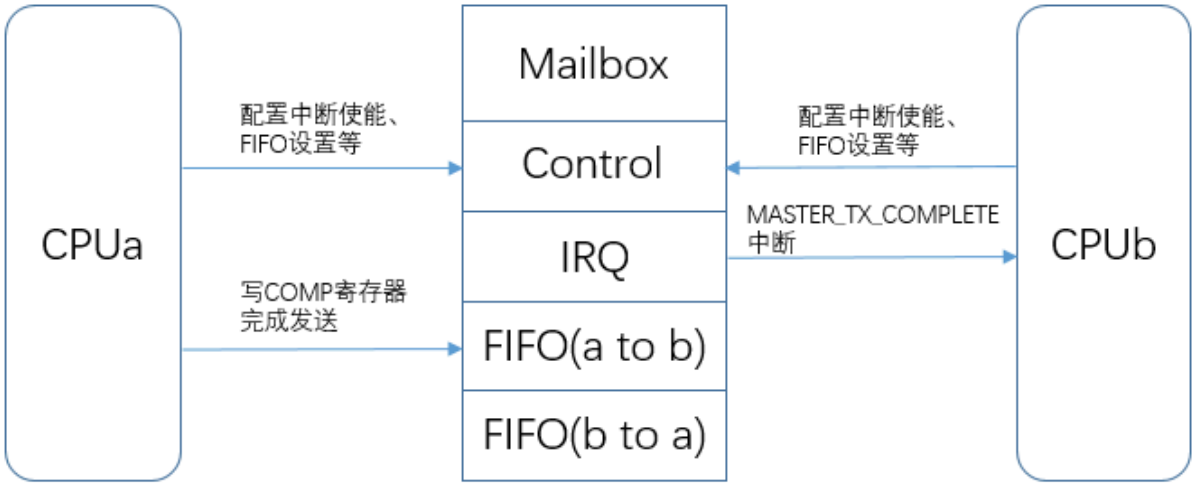


图 3–12 CPUa 发送数据

CPUb 作为发送方时 图 3–13 : CPUb 发送数据 示意图如下

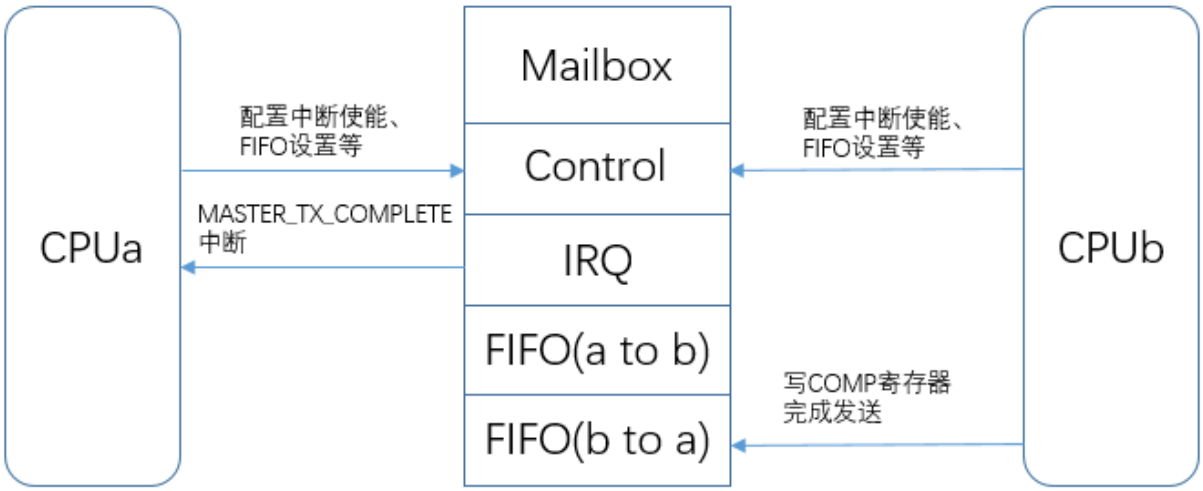


图 3–13 CPUb 发送数据

3.4.3.2. 接收数据

作为接收者时，RX_FIFO_FULL_EN 配置为 1，当 FIFO 达到 RX_FIFO_LEVEL 配置的水位，则 FULL 状态触发，产生 FIFO 为满中断信号，可进行读操作。每次读操作为一个 32bit message。RX_FIFO 为空，若继续进行读操作则造成下溢出，数据发生错误。

假设单次接收有 N 个 messages，当 N=1 时，只需要往 RCOMP 寄存器读 message 即可完成接收并且产生中断。当 N>1 时，需要往 RMSG 寄存器读 N–1 个 messages，读取 RCOMP 寄存器获得最后一个 message 完成接收并且产生中断。

CPUa 作为接收方时 图 3–14 : CPUa 接收数据 示意图如下

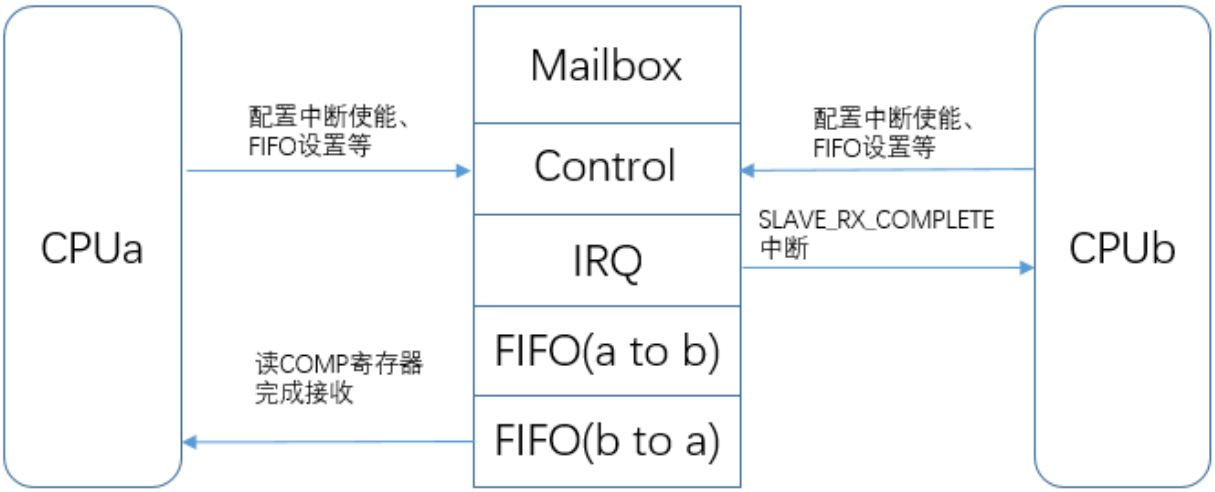


图 3–14 CPUa 接收数据

CPUb 作为接收方时 图 3–15 : CPUb 接收数据示意图如下

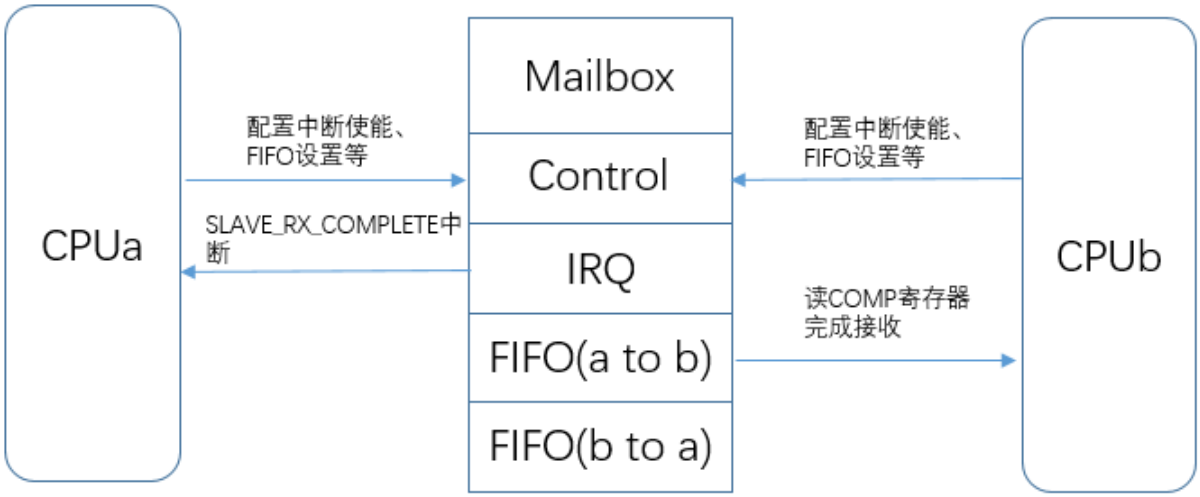


图 3–15 CPUb 接收数据

3.4.3.3. Debug 模式

配置为 debug 模式，则单个 CPU 核可以完成发送与接收数据。

CPUa Debug 模式数据路径： CPUa – FIFO(a to b)—FIFO(b to a) – CPUa

CPUb Debug 模式数据路径： CPUb – FIFO(b to a)—FIFO(a to b) – CPUb

3.4.4. 寄存器列表

表 3–12 MBOX 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	MB_CTL	MBOX 控制	0x000 MB_CTL
0x004	MB_FIFO_CSR	MBOX FIFO 控制及状态	0x004 MB_FIFO_CSR
0x008	MB_IRQ_EN	MBOX 中断使能	0x008 MB_IRQ_EN
0x00C	MB_IRQ_STAS	MBOX 中断状态	0x00C MB_IRQ_STAS
0x010	MB_WMSG	MBOX 写信息	0x010 MB_WMSG
0x014	MB_WCOMP	MBOX 写完成	0x014 MB_WCOMP
0x018	MB_RMSG	MB_RMSG	0x018 MB_RMSG
0x01C	MB_RCOMP	MB_RCOMP	0x01C MB_RCOMP
0xFFC	MB_VER	MBOX 版本号	0xFFC MB_VER

3.4.5. 寄存器描述

3.4.5.1. 0x000 MB_CTL

默认值：0x00000000			MBOX 控制（Mailbox Control）
位域	类型	默认值	描述
31:2	–	–	–
1	R/W	0x0	Complete_mode • 0：读写 Complete 寄存器产生中断 • 1：读写 Complete 寄存器不产生中断
0	R/W	0x0	Debug_mode_sel Debug 模式选择 • 0：不开启 debug 模式 • 1：独立完成发送与接收环路

3.4.5.2. 0x004 MB_FIFO_CSR

默认值：0x10001000			MBOX FIFO 控制及状态（Mailbox FIFO Control Status Register）
位域	类型	默认值	描述
31	R/W1C	0x0	RX_FIFO_RST 复位 RX_FIFO
30	–	–	–
29:24	R/W	0x10	RX_FIFO_LEVEL RX_FIFO 水位设置 RX FIFO 数据达到此水位时触发满中断默认水位为 16word • 0 无效 • 1 表示 1word（范围为 1–32）
23:16	R	0x0	RX_FIFO_CNT 接收 FIFO 数据量 单位为 word（范围为 0–32）
15	R/W1C	0x0	TX_FIFO_RST 复位 TX_FIFO
14	R/W	0x0	TX_FIFO_OVERFLOW_CTL • 0：TX_FIFO 上溢出覆盖数据 • 1：TX_FIFO 上溢出禁止写入
13:8	R/W	0x10	TX_FIFO_LEVEL TX_FIFO 水位 设置 FIFO 数据达到此水位时触发满中断 默认水位为 16word 配置为 0 时无效 1 表示 1word（范围为 1–32）
7:0	R	0x0	TX_FIFO_CNT 本字段为发送 FIFO 数据量单位为 word（范围为 0–32）

3.4.5.3. 0x008 MB_IRQ_EN

默认值：0x00000000			MBOX 中断使能（Mailbox Interrupt Quest Enable）
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	MASTER_TX_COMPLETE_EN 发送完成中断 • 0：禁止 • 1：使能 此位设为 1 时，作为接收方，当发送方完成 message 发送时，此中断状态触发，生成 IRQ
10	R/W	0x0	RX_FIFO_UNDERFLOW_EN RX FIFO 下溢出中断 • 0：禁止 1：使能 此位设为 1 时，若 FIFO 为空时，进行 FIFO 读取操作则生成 IRQ
9	R/W	0x0	RX_FIFO_FULL_EN RX_FIFO 为满中断 • 0：禁止 • 1：使能 此位设为 1 时，若此 FIFO 达到 RX_FIFO_LEVEL 设置的水位则生成 IRQ
8	R/W	0x0	RX_FIFO_EMPTY_EN RX_FIFO 为空中断 • 0：禁止 • 1：使能 此位设为 1 时，若 FIFO 为空时则生成 IRQ
7:4	–	–	–
3	R/W	0x0	SLAVE_RX_COMPLETE_EN 接收完成中断 • 0：禁止 • 1：使能 此位设为 1 时，作为发送方，若接收方完成 message 接收时，此中断状态触发，生成 IRQ
2	R/W	0x0	TX_FIFO_OVERFLOW_EN

默认值：0x00000000			MBOX 中断使能（Mailbox Interrupt Quest Enable）
位域	类型	默认值	描述
			TX FIFO 上溢出中断 • 0：禁止 • 1：使能 此位设为 1 时，若 FIFO 为 32word 时，继续往 FIFO 填充数据则生成 IRQ
1	R/W	0x0	TX_FIFO_FULL_EN TX_FIFO 为满中断 • 0：禁止 • 1：使能 此位设为 1 时，若此 FIFO 达到 TX_FIFO_LEVEL 设置的水位则生成 IRQ
0	R/W	0x0	TX_FIFO_EMPTY_EN TX_FIFO 为空中断 • 0：禁止 • 1：使能 此位设为 1 时，若 FIFO 为空时则生成 IRQ

3.4.5.4. 0x00C MB_IRQ_STAS

默认值：0x00000101			MBOX 中断状态（Mailbox Interrupt Quest Status）
位域	类型	默认值	描述
31:12	–	–	–
11	R/W1C	0x0	MASTER_TX_COMPLETE_STAS 发送完成中断 • 0：未触发 • 1：触发
10	R/W1C	0x0	RX_FIFO_UNDERFLOW_STAS RX FIFO 下溢出中断 • 0：未触发 • 1：触发
9	R/W1C	0x0	RX_FIFO_FULL_STAS RX_FIFO 为满中断

默认值：0x00000101			MBOX 中断状态（Mailbox Interrupt Quest Status）
位域	类型	默认值	描述
			• 0：未触发 • 1：触发
8	R/W1C	0x1	RX_FIFO_EMPTY_STAS RX_FIFO 为空中断 • 0：未触发 • 1：触发
7:4	–	–	–
3	R/W1C	0x0	SLAVE_RX_COMPLETE_STAS 接收完成中断 • 0：未触发 • 1：触发
2	R/W1C	0x0	TX_FIFO_OVERFLOW_STAS TX FIFO 上溢出中断 • 0：未触发 • 1：触发
1	R/W1C	0x0	TX_FIFO_FULL_STAS TX_FIFO 满中断 • 0：未触发 • 1：触发
0	R/W1C	0x1	TX_FIFO_EMPTY_STAS TX_FIFO 为空中断 • 0：未触发 • 1：触发

3.4.5.5. 0x010 MB_WMSG

默认值：0x00000000			MBOX 写信息（Mailbox Write Message）
位域	类型	默认值	描述
31:0	WO	0x0	WRITE_MESSAGE 写 messages 时：将 N–1（N 大于 1）个 message 写进本字段

3.4.5.6. 0x014 MB_WCOMP

默认值：0x00000000			MBOX 写完成（Mailbox Write Complete）
位域	类型	默认值	描述
31:0	WO	0x0	WRITE_COMPLETE 写 message 时：将最后一个 message 写进本字段完成传输

3.4.5.7. 0x018 MB_RMSG

默认值：0x00000000			MBOX 读信息（Mailbox Read Message）
位域	类型	默认值	描述
31:0	RO	0x0	READ_MESSAGE 读 message 时：读取本字段可获取 N–1（N 大于 1）个 message

3.4.5.8. 0x01C MB_RCOMP

默认值：0x00000000			MBOX 读完成（Mailbox Read Complete）
位域	类型	默认值	描述
31:0	RO	0x0	READ_COMPLETE 读 message 时：读取本字段可获取最后一个 message

3.4.5.9. 0xFFC MB_VER

默认值：0x00000100			MBOX 版本号（Version）
位域	类型	默认值	描述
31:0	RO	0x00000100	VERSION Mailbox 版本信息

4. 安全

4.1. Crypto Engine (CE)

Crypto Engine (CE) 是安全子系统中的加解密模块，用于安全算法的硬件加速以及其它安全机制的实现。

4.1.1. 特性说明

- 支持多种算法的加速
- 专用安全 SRAM
- 支持生成安全密钥
- 支持多种密钥输入方式
- 内部专用 DMA
- 支持多路并行处理

类别	算法
对称密钥算法	AES-ECB (128/ 192/ 256)
	AES-CBC (128/ 192/ 256)
	AES-CTR (128/ 192/ 256)
	TDES-EDE-ECB (64/ 128/ 192)
	TDES-EDE-CBC (64/ 128/ 192)
消息摘要算法	MD5
	SHA1/ 224/ 256/ 384/ 512
	HMAC-SHA1/ HMAC-SHA-256
	TRNG (256 bits)

4.1.2. 原理框图

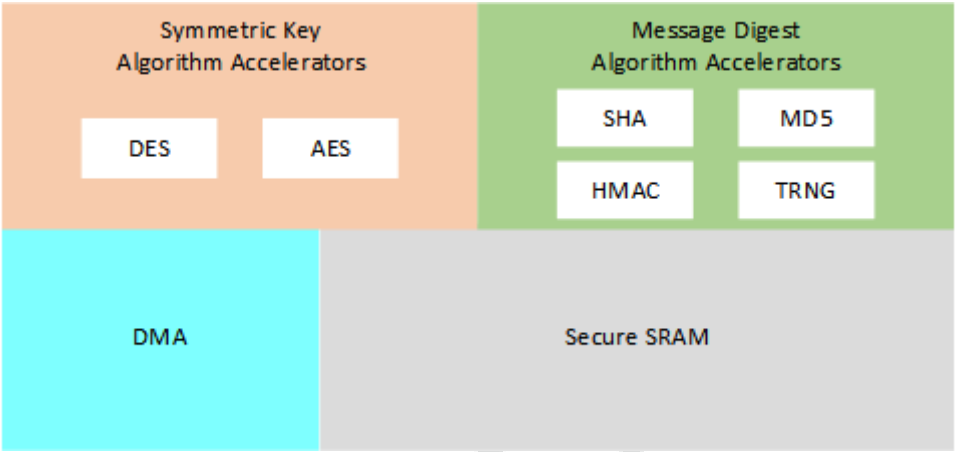


图 4–1 原理框图

4.1.3. 功能描述

CE 模块内部含有多个重要功能和组件，包括：

- 对称密钥算法加速单元：支持多种类型的对称密钥算法。
- 消息摘要算法加速单元：支持使用算法规定的默认初始化向量值，也支持用户配置初始化向量值。
- 专用的安全 SRAM，用于存放密钥。
- 支持以任务链的形式将一个运算请求分拆为多个链接在一起的子运算任务。一个任务链中的所有任务，算法类型必须相同。

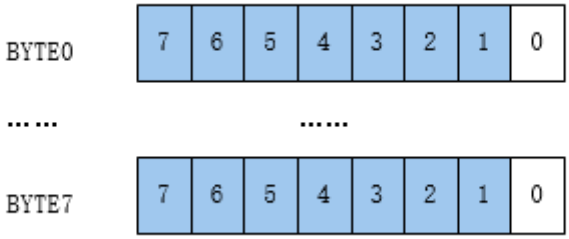
4.1.3.1. 对称密钥算法

CE 内部包含对称密钥算法加速单元，支持以下类型的对称密钥算法：

- AES 算法，支持下列模式：
  - ECB
  - CBC
  - CTR
- DES/ TDES 算法，支持下列模式：
  - ECB
  - CBC

  TDES 算法具体实现为 TDES–EDE，支持三种密钥长度，分别对应使用 1、2、3 个 64–bit 子密钥的场景：
  - 64 bits
  - 128 bits
  - 192 bits

TDES 算法的基础是 DES 算法，DES 算法使用 56–bit 长度的密钥，对于输入的 64–bit 密钥数据，每个字节只取高 7 位。



使用 TDES 算法时：

- 当输入 64–bit 长度的密钥时，TDES 算法单元实际执行经典 DES 算法运算。
- 当输入 128–bit 长度的密钥时，TDES 算法单元将输入密钥分为两个子密钥 K1 和 K2，并且使用 K1–K2–K1 分别执行 Encrypt–Decrypt–Encrypt 三次 DES 运算。
- 当输入 192–bit 长度的密钥时，TDES 算法单元将输入密钥分为三个子密钥 K1、K2 和 K3，并且使用 K1–K2–K3 分别执行 Encrypt–Decrypt–Encrypt 三次 DES 运算。

• SM4 算法，支持下列模式：

- ECB
- CBC
- CTR

4.1.3.2. 消息摘要算法

CE 内部的消息摘要算法加速单元包括：

• MD5 算法

• SHA 算法

• 使用 SHA1、SHA256 的 HMAC 算法

• SM3 算法

• TRNG 算法

消息摘要算法支持使用算法规定的默认初始化向量值，也支持用户配置初始化向量值。

警告：

• MD5 算法的初始化向量，输入时使用小端序；

• 其它算法的初始化向量，输入时使用大端序。

4.1.3.3. 任务和处理

CE 以任务的方式处理运算请求，支持以任务链的形式将一个运算请求分拆为多个链接在一起的子运算任务。

CE 内部的三个算法单元，可以并行处理各自的运算任务，但是一个算法单元每次仅接受一个任务链的处理请求，完成之后， 继续处理下一个任务链。因此软件在发起运算请求之前，需要检查对应的算法单元是否空闲。

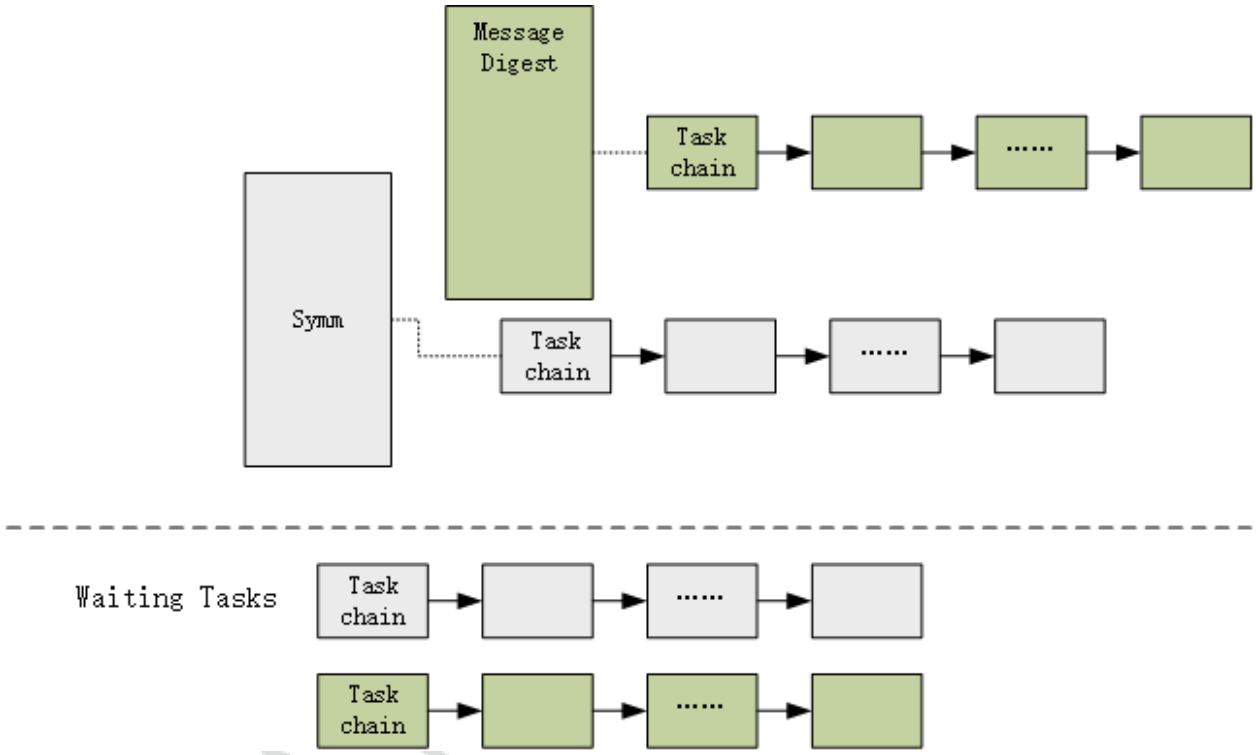


图 4–2 CE 对任务的处理

在 图 4–2 : CE 对任务的处理中，当 CE 的算法单元没有空闲时，对应的任务需要等待，由软件处理等待。

> **警告：**
>
> 一个任务链中的所有任务，算法类型必须相同。

4.1.3.4. 硬件安全密钥

CE 模块包含一块内部专用的安全 SRAM，用于存放密钥。保存在安全 SRAM 中的密钥称为硬件安全密钥 (HSK)。

安全 SRAM 的大小为 1 KB，由软件驱动完成具体空间的分配和管理，CE 模块仅使用该区域中的数据。安全 SRAM 只能被 CE 模块访问，因此保存在其中的密钥安全性高，外界无法窥探。

安全 SRAM 的使用规则如下：

- 必须以 64 bits 为单位分配。
- 仅 CE 可以访问，可以作为 CE 的输入密钥，也可以作为 CE 的输出。
- HSK 只能从其它密钥派生，由 CE 输出到安全 SRAM。
- 不能以任何方式导出安全 SRAM 中的 HSK。即 HSK 不能作为 CE 的输入数据，只能作为 CE 的密钥。
- 通过安全 SRAM 的地址索引使用 HSK。

模块名称	基地址	地址空间
Secure SRAM	0x1002_1000	1 KB

仅 CE 可以访问安全 SRAM 的地址空间，CPU 和其它模块无法访问。 CE 访问安全 SRAM 时，只需在任务描述符中直接指定使用的安全 SRAM 地址。

4.1.3.5. HSK 详细说明

4.1.3.5.1. HSK 的派生

HSK 只能从其它密钥派生。

- 派生 HSK 的密钥为父密钥。 父密钥可以为：
  - eFuse 安全密钥
  - 用户指定的其它密钥

    HSKM 可以是用户指定的任意数据，但输出的 HSK 最大只能为 2048 bit。
- 用于派生 HSK 的数据称为 HSK Material (HSKM)。

图 4–3 : eFuse 密钥和 HSKM 派生 HSK 的过程如下。

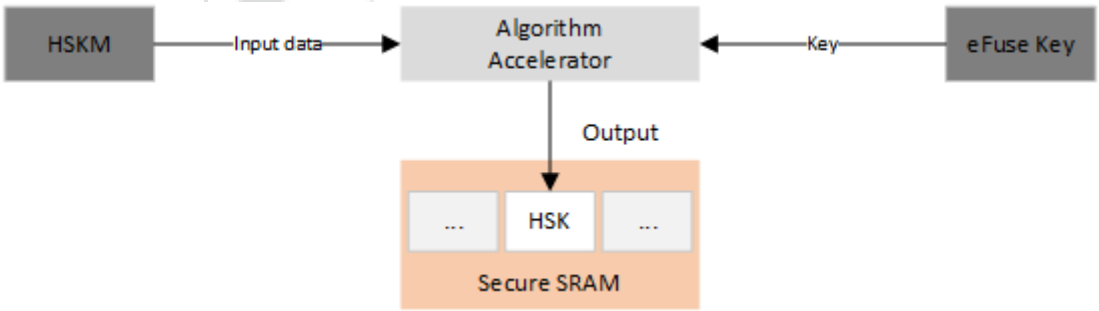


图 4–3 eFuse 密钥和 HSKM 派生 HSK 的过程

图 4–4 : HSK 的使用限制如下：

- 只能作为 CE 的密钥进行使用。
- 不能作为 CE 的数据输入。

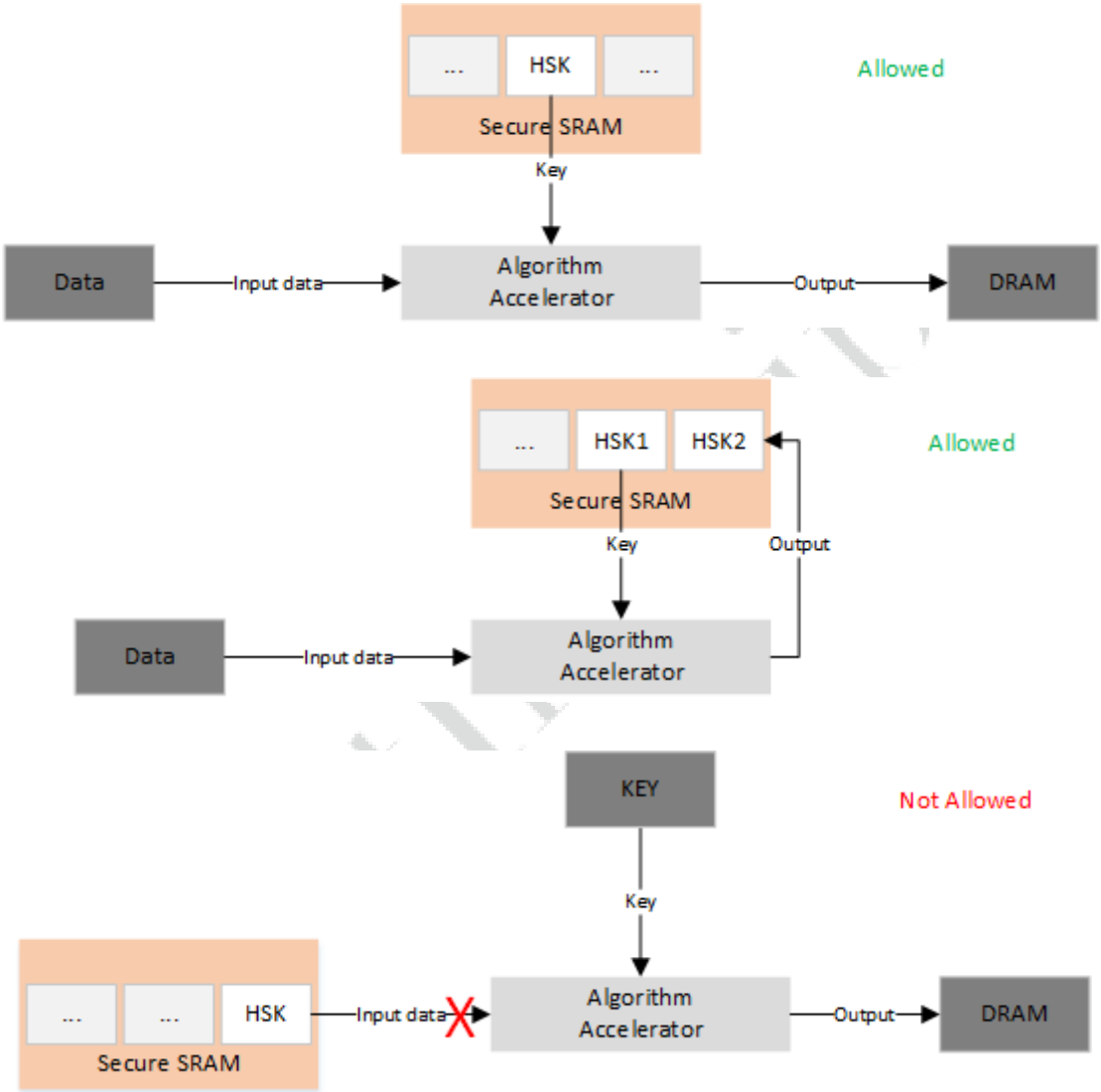


图 4–4 HSK 的使用限制

4.1.3.5.2. HSK 的功能

硬件安全密钥可以解决密钥的本地存储安全问题。

将关键密钥保存在设备中时，密钥的本地存储存在以下安全问题：

- 如以明文保存，密钥不安全，容易被窃取。
- 如烧录到 eFuse，则需要使用较多的安全 eFuse 空间。
- 在需要动态生成密钥的场景中，安全 eFuse 无法满足动态生成密钥的需求。

硬件安全密钥解决密钥本地存储安全问题的方式如下：

- 对于固定密钥：使用 eFuse 安全密钥，先对 HSK 进行加密，再将加密后的数据 (HSKM) 保存在设备本地。需要使用时，将 HSKM 解密输出到安全 SRAM，得到 HSK。

由于 eFuse 安全密钥仅 CE 可访问，软件以及其它硬件模块无法读写，因此本地设备保存 HSKM 不会导致 HSK 的泄露。

4.1.3.5.3. eFuse 安全密钥

eFuse 安全密钥是指 eFuse 中仅 CE 模块可访问，其它软硬件模块无法读写的 eFuse 密钥区域。

CE 模块支持下列 eFuse 安全密钥。

表 4–1 eFuse 安全密钥

名字	大小 (BIT)	描述
SSK/SPI_ENC_KEY	128	对称密钥，用于固件加密

4.1.3.5.4. 数据填充

CE 模块所支持的算法，多数是按照固定数据块的方式处理数据。如果实际待处理数据的长度不满足对齐要求时，需要进行数据填充。不同算法的填充处理方式不同，CE 仅支持部分算法进行硬件填充处理。

CE 模块中各算法对数据块要求和数据填充的支持情况如下：

表 4–2 CE 模块算法对应的数据填充要求

	数据块大小 (Bit)	对齐要求	填充支持
AES 算法	128	16 字节对齐	不支持 CE 硬件填充，需要用户进行数据填充
DES/TDES 算法	64	8 字节对齐	不支持 CE 硬件填充，需要用户进行数据填充
SHA/MD5 算法	非最后一笔数据，应 512bit/ 1024bit 对齐，具体看算法要求。		支持 CE 硬件填充。

4.1.4. 描述符说明

描述符是 CE 模块的软件编程接口。

4.1.4.1. 任务描述符

使用 CE 模块进行硬件加速时，软件需要以任务描述符的方式将详细的任务信息配置给 CE 模块。 多个相关的任务通过链表的方式构建任务链表 (Task List)，CE 模块则按照任务链表顺序逐个处理任务。

> **警告：**
>
> 当一个任务链表由多个任务描述符组成时，链表内的所有任务描述符必须使用相同的算法。

任务描述符的大小固定，组成部分如下所示：

表 4–3 任务描述符组成说明

任务描述符组成部分	大小/字节	说明
Algorithm descriptor	36	算法描述符
Data descriptor	24	数据描述符
Next	4	下一个任务描述符的地址

图 4–5 : 任务列表显示了多个任务描述符构成的任务列表示例：

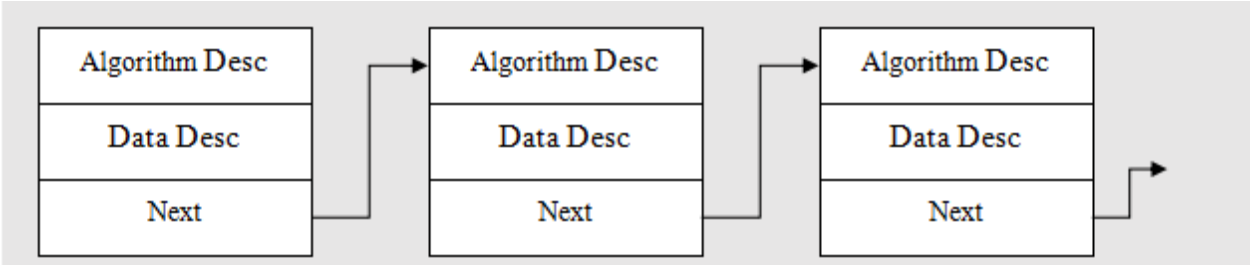


图 4–5 任务列表

以下是任务描述符的具体位域分配总览：

表 4–4 任务描述符的位域分配

字段		位域	备注
Algorithm Desc	Algorithm Config	31:24	–
		23:20	Key size，密钥大小
		19:16	Key source，密钥源选择
		15:12	–
		11:10	Counter width
		9	Hash IV flag
		8	Direction
		7:0	Algorithm tag，算法标识
	Key address	31:0	–
	IV address	31:0	–
	Counter output address	31:0	–
	Reserved	[31:0] × 5	20 字节
Data Desc	Data Info	31:2	–
		1	First flag，当前数据是否为开始数据块
		0	Last flag，当前数据是否为结束数据块
	Total length	31:0	–
	Input Address	31:0	当前输入数据地址

表 4–4 任务描述符的位域分配 (续)

字段		位域	备注
	Input Length	31:0	当前输入数据长度
	Output Address	31:0	当前输出数据地址
	Output Length	31:0	当前输出数据长度
Next		31:0	下一个任务描述符的地址

4.1.4.2. 算法描述符

算法描述符区域，用于描述当前任务的算法信息和参数。

以下以算法描述符的大小固定为 36 个字节为例。不同算法需要的输入参数各有不同，对于任务描述符中定义的算法描述符位域，具体的算法描述符根据需要给出具体定义。

表 4–5 算法标记值列表

算法标记值	描述
0x00	AES–ECB
0x01	AES–CBC
0x02	AES–CTR
0x03	–
0x04	–
0x10	DES–ECB
0x11	DES–CBC
0x20	TDES–ECB
0x21	TDES–CBC
0x40	SHA–1
0x41	SHA–224
0x42	SHA–256
0x43	SHA–384

表 4–5 算法标记值列表 (续)

算法标记值	描述
0x44	SHA–512
0x45	MD5
0x46	HMAC–SHA–1
0x47	HMAC–SHA–256
0x50	TRNG

4.1.4.2.1. AES–ECB 描述符

表 4–6 AES–ECB 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: AES–ECB 具体参考 算法标记值列表。

表 4–6 AES–ECB 描述符 (续)

字段	位域	描述
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址
Reserved	28 Bytes	–

4.1.4.2.2. AES–CBC 描述符

表 4–7 AES–CBC 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: AES–CBC 具体参考 算法标记值列表。
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址

表 4–7 AES–CBC 描述符 (续)

字段	位域	描述
IV address	31:0	初始化向量地址 地址应是 RAM 的地址。 地址应 8 字节对齐。
Reserved	24 Bytes	–

4.1.4.2.3. AES–CTR 描述符

表 4–8 AES–CTR 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: AES–CTR 具体参考 算法标记值列表。
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址

表 4–8 AES–CTR 描述符 (续)

字段	位域	描述
Counter input	31:0	初始化计数值地址 地址应是 RAM 的地址。 地址应 8 字节对齐。
Counter output	31:0	计数值输出地址 地址应是 RAM 的地址。 地址应 8 字节对齐。 输出下一个数据块的计数值。
Reserved	20 Bytes	–

4.1.4.2.4. DES–ECB 描述符

表 4–9 DES–ECB 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption

表 4–9 DES–ECB 描述符 (续)

字段	位域	描述
	7:0	Algorithm tag: DES–ECB 具体参考 算法标记值列表。
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址
Reserved	28 Bytes	–

4.1.4.2.5. DES–CBC 描述符

表 4–10 DES–CBC 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: DES–CBC 具体参考 算法标记值列表。

表 4–10 DES–CBC 描述符 (续)

字段	位域	描述
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址
IV address	31:0	初始化向量地址 地址应是 RAM 的地址。 地址应 8 字节对齐。
Reserved	24 Bytes	–

4.1.4.2.6. TDES–ECB 描述符

表 4–11 TDES–ECB 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: TDES–ECB 具体参考 算法标记值列表。

表 4–11 TDES–ECB 描述符 (续)

字段	位域	描述
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址
Reserved	28 Bytes	–

4.1.4.2.7. TDES–CBC 描述符

表 4–12 TDES–CBC 描述符

字段	位域	描述
Algorithm config	31:24	–
	23:20	Key size，密钥大小 • 0x00: 64（不支持） • 0x01: 128 • 0x02: 192 • 0x03: 256 • 0x04: 512（不支持） • 0x05: 1024（不支持） • 0x06: 2048（不支持）
	19:16	密钥源的选择 • 0x00: User/RAM • 0x01: eFuse SSK
	15:9	–
	8	Direction,运算方向 • 0x00: Encryption • 0x01: Decryption
	7:0	Algorithm tag: TDES–CBC 具体参考 算法标记值列表。
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址

表 4–12 TDES–CBC 描述符 (续)

字段	位域	描述
IV address	31:0	初始化向量地址 地址应是 RAM 的地址。 地址应 8 字节对齐。
Reserved	24 Bytes	–

4.1.4.2.8. Message Digest 描述符

表 4–13 Message Digest 描述符

字段	位域	描述
Algorithm config	31:12	–
	11	Input IV Flag，初始化向量标记 • 0x00: 使用内部默认的初始化向量值 • 0x01: 使用用户提供的初始化向量值
	10:8	–
	7:0	Algorithm tag 具体参考 算法标记值列表。
Key address	31:0	密钥地址 仅当密钥源类型为 RAM 时配置。 地址应是 RAM 或 Secure SRAM 的地址 仅 HMAC 算法需要配置，为任意长度数据。
IV address	31:0	初始化向量地址（Input IV Flag 为 1 时） 地址应是 RAM 的地址。 地址应 8 字节对齐。 初始化向量的长度，应参考对应算法的规定。
Reserved	24 Bytes	–

4.1.4.2.9. TRNG 描述符

表 4–14 TRNG 描述符

字段	位域	描述
Algorithm config	31:8	–
	7:0	Algorithm tag: TRNG 具体参考 算法标记值列表。
Reserved	32 Bytes	–

4.1.4.3. 数据描述符

表 4–15 数据描述符

字段	位域	描述
Data information	31:24	Hash data total length high8 数据摘要总数据长度的高 8 位，在数据摘要算法中使用
	23:2	–
	1	Last Flag, 结束数据块标记 • 0：当前数据数据非结束数据块 • 1：当前数据数据为结束数据块
	0	First Flag, 开始数据块标记 • 0：当前数据数据非开始数据块 • 1：当前数据数据为开始数据块
Hash data total length low32	31:0	数据摘要总数据长度的低 32 位 仅在数据摘要算法中使用，表示所有处理的数据长度
Input address	31:0	当前输入数据地址 必须是 RAM 的有效地址，不可以为 Secure SRAM 的地址。 地址应 8 字节对齐。
Input data length	31:0	当前输入数据长度 • 对 AES 算法，数据长度都必须为 16 字节对齐。 • CTR 模式向上取整到 16 字节对齐。 • DES/ TDES 算法，数据长度必须 8 字节对齐。 • 数据摘要算法对输入数据没有长度要求。

表 4–15 数据描述符 (续)

字段	位域	描述
Output address	31:0	当前输入数据地址 应为 RAM 或 Secure SRAM 的有效地址。 地址应 8 字节对齐。
Output data length	31:0	当前输出数据长度 • 对 AES 算法，数据长度都必须为 16 字节对齐。 • CTR 模式向上取整到 16 字节对齐。 • DES/ TDES 算法，数据长度都必须 8 字节对齐，与输入数据长度一致。

4.1.5. 寄存器列表

表 4–16 CE 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	CE_IRQ_CTL	中断控制	0x000 CE_IRQ_CTL
0x004	CE_IRQ_STA	中断状态	0x004 CE_IRQ_STA
0x008	CE_TSK_ADDR	任务描述符地址	0x008 CE_TSK_ADDR
0x00C	CE_TSK_CTL	任务控制	0x00C CE_TSK_CTL
0x010	CE_TSK_STA	任务状态	0x010 CE_TSK_STA
0x014	CE_TSK_ERR	任务错误	0x014 CE_TSK_ERR
0xFFC	CE_VER	版本	0xFFC CE_VER

4.1.6. 寄存器描述

4.1.6.1. 0x000 CE_IRQ_CTL

默认值：0x00000000			中断控制（Interrupt Request Control）
位域	类型	默认值	描述
31:2	–	–	–
1	R/W	0x0	数据摘要算法单元的中断使能位

默认值：0x00000000			中断控制（Interrupt Request Control）
位域	类型	默认值	描述
			0x0：关闭 0x1：打开
0	R/W	0x0	对称密钥算法单元的中断使能位 0x0：关闭 0x1：打开

4.1.6.2. 0x004 CE_IRQ_STA

默认值：0x00000000			中断状态（Interrupt Request Status）
位域	类型	默认值	描述
31:2	–	–	–
1	R/W1C	0x0	数据摘要算法单元的中断使能位 0x0：无中断 0x1：中断产生，表示使用该通道的任务处理完毕
0	R/W1C	0x0	对称密钥算法单元的中断使能位 0x0：无中断 0x1：中断产生，表示使用该通道的任务处理完毕

4.1.6.3. 0x008 CE_TSK_ADDR

默认值：0x00000000			任务描述符地址（Task Address Register）
位域	类型	默认值	描述
31:0	R/W	0x0	任务链表的首地址

4.1.6.4. 0x00C CE_TSK_CTL

默认值：0x00000000			任务控制（Task Control Register）
位域	类型	默认值	描述
31	R/W	0x0	任务加载 将该比特设置为 1，则 CE 开始加载。

默认值：0x00000000			任务控制（Task Control Register）
位域	类型	默认值	描述
			将 CE_TSK_ADDR 所指向的任务加载到指定算法所属的算法单元执行。任务开始加载时，该比特位自动清零。
30:8	–	–	–
7:0	R/W	0x0	任务对应的算法标记 指明当前任务所用的算法，CE 根据该算法标记将任务加载到对应的算法单元。具体值可参考 算法标记值列表。

4.1.6.5. 0x010 CE_TSK_STA

默认值：0x00FFFFFF			任务状态（Task Status Register）
位域	类型	默认值	描述
31:16	–	–	–
15:8	RO	0xFF	数据摘要算法单元正在执行的算法类型，具体值可参考算法标记值列表。
7:0	RO	0xFF	对称密钥算法单元正在执行的算法类型，具体值可参考算法标记值列表。

4.1.6.6. 0x014 CE_TSK_ERR

默认值：0x00000000			任务错误（Task Error Register）
位域	类型	默认值	描述
31:16	–	–	–
15:8	R/W1C	0x0	数据摘要算法单元的错误类型 0x01：算法不支持 0x02：数据长度错误 0x04：安全 SRAM 访问错误 0x08：地址非 8 字节对齐 0x10：密钥长度不正确
7:0	R/W1C	0x0	对称密钥算法单元的错误类型 0x01：算法不支持 0x02：数据长度错误 0x04：安全 SRAM 访问错误 0x08：地址非 8 字节对齐

默认值：0x00000000			任务错误（Task Error Register）
位域	类型	默认值	描述
			0x10：密钥长度不正确

4.1.6.7. 0xFFC CE_VER

默认值：0x00000101			版本（Version Register）
位域	类型	默认值	描述
31:0	RO	0x00000101	Version（模块版本号） 采用 BCD 码显示，V1.1

4.2. Serial Peripheral Interface Encryption (SPI ENC)

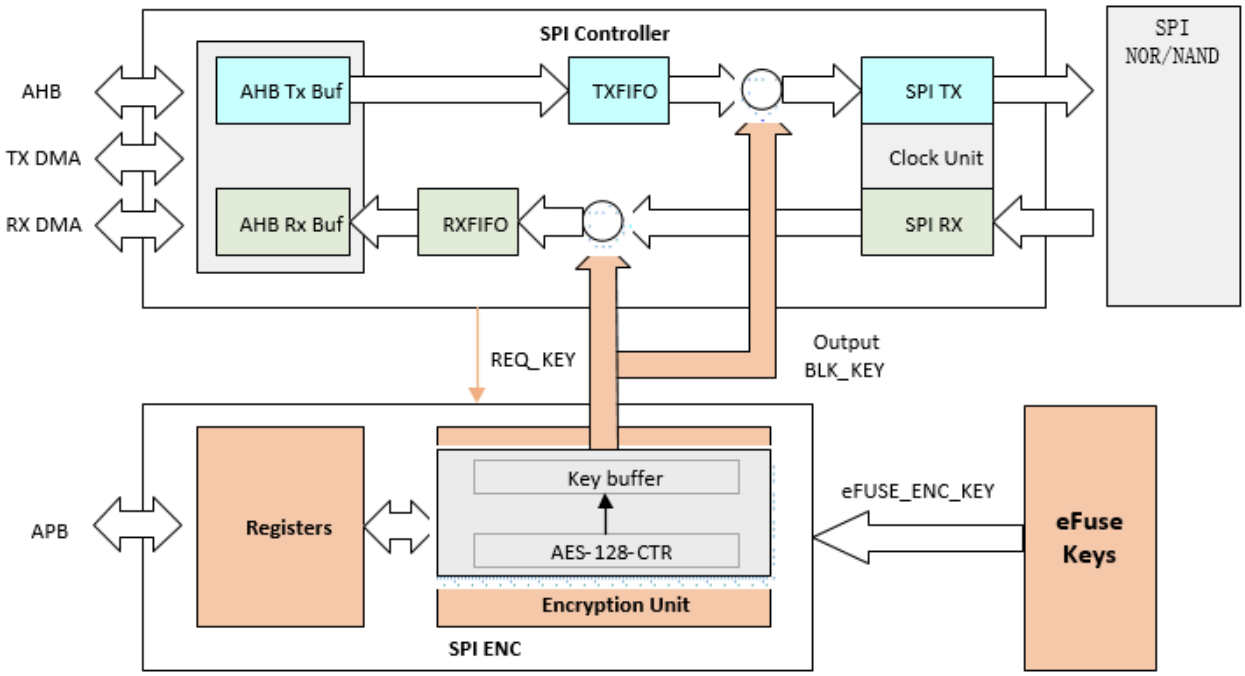
Serial Peripheral Interface Encryption (SPI ENC) 模块用于对 SPI 总线上的数据进行在线加密和解密，实现对 CPU 透明的 SPI 数据加密存储。即 CPU 对 SPI 存储设备的读写数据是明文，但存储在 SPI 设备上的数据是密文，数据的加密和解密操作在数据传输过程中完成。

SPI ENC 用于 SPI 存储设备，即 SPI NOR/ SPI NAND，需要配合 SPI 控制器使用。

4.2.1. 特性说明

- 支持 AES-128-CTR 加密和解密
- 通过配置连接到不同的 SPI 控制器
- 通过 eFuse 配置密钥
- 支持明文和密文混合传输
- 不影响 SPI 总线传输带宽
- 不支持 SPI 全双工模式
- 内置 64 字节分组密钥 Buffer
- 支持全 0xFF 空页检测

4.2.2. 原理框图



4.2.3. 功能描述

SPI 数据加密使用 AES-128-CTR 算法，加解密总体计算过程如下：

1. 以 16 字节为单位对数据进行分组，并且为每个数据分组产生对应的 Counter 值。

   关于详细的计算详细流程，可查看 计算分组密钥。

2. 使用一个 128-bit AES 密钥，对 Counter 值进行加密运算，产生数据分组的对应分组密钥。

   产生数据分组对应的 Counter 值，需要使用一个 Nonce，在 SPI ENC 使用过程中，Nonce 作为密钥数据的一部分提供。

3. 使用分组密钥与数据进行异或运算，得到密文。

   关于如何输出分组密钥的流程，可查看 输出分组密钥。

   关于异或运算的详细描述，可查看 数据加解密。

密钥数据的来源有两个：

- SPI 加密模块从 eFuse 中读取 AES KEY 和 Nonce。eFuse 中保存的密钥数据如下：

名称	长度	说明
SPI_ENC_KEY	128 bits	加密密钥，仅 SPI_ENC 模块可见。
SPI_ENC_NONCE	64 bits	生成 Counter，仅 SPI_ENC 模块可见。

- 用户通过相应的寄存器配置 AES KEY 和 Nonce，主要用于内部调试。

4.2.3.1. 计算分组密钥

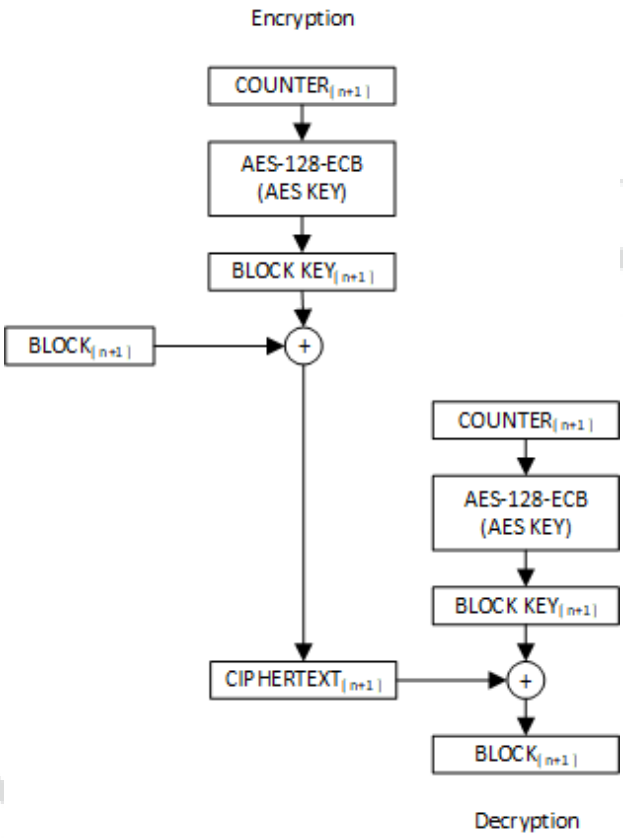


图 4–6 AES–128–CTR 算法原理

图 4–6 : AES–128–CTR 算法原理中，使用 AES–128–CTR 进行加密或者解密的过程分为三个步骤：

1. 生成数据分组的 Counter 值。

   Counter 值的产生方式如下：

```
ID = (SPIE_ADDR & 0xFFFFFFF0) / 16
Counter(ID) = (Nonce << 64) +(SPIE_TWEAK << 32) +  ID
```

2. 生成分组密钥。

   a. 使用数据地址寄存器 SPIE_ADDR 中的地址值，计算分组 ID。

   b. 分组 ID 与 Nonce 和 Tweak 值组成 Counter 值。

   c. 使用 AES–128–ECB 加密算法，用 AES KEY 对 Counter 值进行加密，得到分组密钥。

      SPI ENC 一次可产生 16 字节的分组密钥，输出到内部 Key buffer。

      模块开始工作后，只要 Key buffer 有空间，计算分组密钥就会继续，并且总是按地址顺序自动计算下一组密钥，直到控制寄存器 SPIE_CTL[0] 被设置为 0。

注：

如图 4–6 : AES–128–CTR 算法原理所示，数据的加密和解密的分组密钥计算流程完全一致。

3. 使用分组密钥对数据进行异或运算，产生加密数据。详细流程可查看 输出分组密钥。

4.2.3.2. 输出分组密钥

SPI ENC 模块仅计算 AES–128–CTR 的分组密钥，最后一步使用分组密钥对数据进行异或运算由 SPI 控制器完成。因此在启用 SPI ENC 的情况下，SPI 控制器需要为发送到总线或者从总线接收到的每一字节数据读取分组密钥。

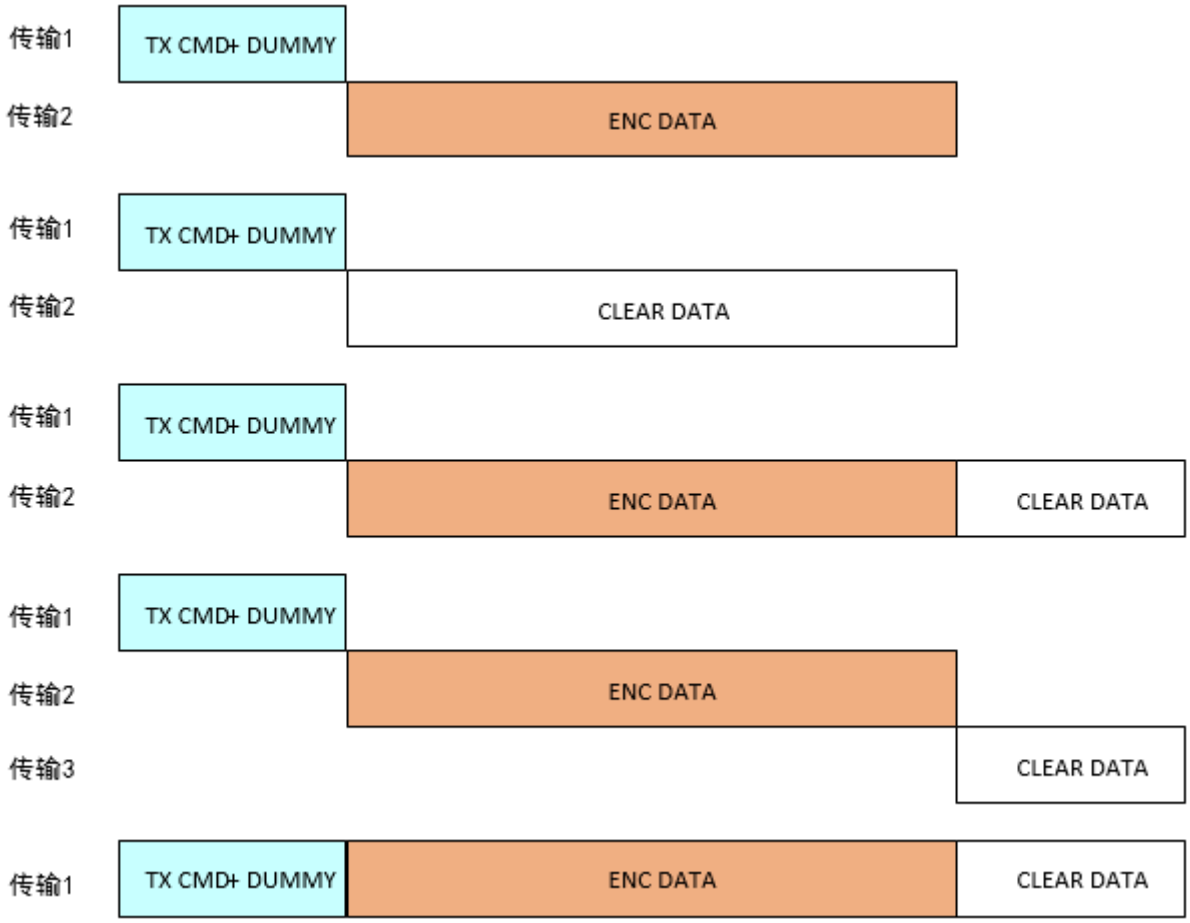


图 4–7 SPI 数据传输模式

如 图 4–7 : SPI 数据传输模式所示，在与 SPI NOR/ NAND 通信过程中，传输的数据可能出现明文和密文混合传输的情形。

SPI ENC 输出分组密钥的规则如下：

- SPI 控制器每收发一个字节数据，都要读取一个字节的分组密钥
- 明文数据位置对应的分组密钥，总是读取到 0；
- 密文数据位置对应的分组密钥，总是计算得到的分组密钥；

分组密钥的计算总是以 16 字节为单位，但用户访问的地址可能并不是 16 字节对齐，如从 0x0000_0005 地址开始读取数据。 这种情况下，SPI ENC 计算的第一个分组密钥仍然为 16 字节，只是对应的数据地址为 0x0000_0000 ~ 0x0000_000F；SPI ENC 输出数据分组密钥时，应输出 0x05~0x0F 数据对应的分组密钥值。

注：

SPI ENC 模块要准确的计算和输出分组密钥，依赖于用户开始传输前配置的数据地址、长度、加密数据的起始地址和长度等信息。

4.2.3.3. 数据加解密

使用分组密钥与对应的数据分组进行异或运算，即可完成数据的加密或者解密。

注：

由于 SPI ENC 输出的密钥，只能应用于接收的数据，或者发送的数据，不可以同时应用于两组数据，因此使能 SPI ENC 时，不支持 SPI 全双工模式，

- 对于发送数据，数据会先被写入 TXFIFO，SPI 控制器在将数据发送到 SPI 总线上之前执行：
  1. 读取一个字节的分组密钥数据。
  2. 将分组密钥数据与要发送的一个字节数据进行异或运算。
  3. 将结果发送到 SPI 总线。
- 对于接收数据：
  1. 从 SPI 总线读取到一个字节数据的同时，读取一个字节的分组密钥。
  2. 将分组密钥数据与接收的数据进行异或运算。
  3. 将解密后的数据，写入 RXFIFO。

4.2.4. 编程指南

开始配置之前，确保 SPI ENC 模块已经初始化。初始化步骤如下：

1. 根据系统配置，将 SPI ENC 模块连接到正确的 SPI 控制器。

   通过设置 SPIE_CTL[13:12] 选择连接的 SPI 控制器。
2. 设置密钥。默认使用 eFuse 中的密钥数据。
3. 通过设置 SPIE_ICR 寄存器使能相应中断。

SPI_ENC 模块的配置步骤如下：

1. 读写 SPI 存储，传输时确保 SPIE_CTL[0] = 0 (KEY_START = 0)。

   读写 SPI 存储时，不需要对命令和 SPI 器件的 OTP 读写操作进行加解密。
2. 设置以下参数进行 SPI 存储的加密数据传输。

- SPIE_ADDR：当前访问的数据地址
- SPIE_TWEAK：用于调整 Counter 值
- SPIE_CPOS, SPIE_CLEN：当前传输的密文数据位置

3. 设置 SPIE_CTL[0] = 1 (KEY_START = 1) 值，开始计算分组密钥。

   产生密钥 (SPIE_ISR[0])。

   确保密钥已经产生。如密钥产生，则 SPI 控制器开始传输数据。

4. 通过 SPIE_ISR 检查加解密是否完成，并查看读取的数据是否为空数据（全为 0xFF）。

4.2.5. 寄存器列表

表 4–17 SPIENC 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	SPIE_CTL	控制寄存器	0x000 SPIE_CTL
0x004	SPIE_ICR	中断控制寄存器	0x004 SPIE_ICR
0x008	SPIE_ISR	中断状态寄存器	0x008 SPIE_ISR
0x00C	SPIE_KCNT	密钥计数寄存器	0x00C SPIE_KCNT
0x010	SPIE_OCNT	输出计数寄存器	0x010 SPIE_OCNT
0x014	SPIE_ADDR	数据地址寄存器	0x014 SPIE_ADDR
0x018	SPIE_TWEAK	计数调整寄存器	0x018 SPIE_TWEAK
0x01C	SPIE_CPOS	密文位置寄存器	0x01C SPIE_CPOS
0x020	SPIE_CLEN	密文长度寄存器	0x020 SPIE_CLEN
0xFFC	SPIE_VER	SPIENC 版本	0xFFC SPIE_VER

4.2.6. 寄存器描述

4.2.6.1. 0x000 SPIE_CTL

默认值：0x00000000			控制寄存器（Control Register）
位域	类型	默认值	描述
31:17	–	–	–
16	R/W	0x0	XIP_EN SPI XIP 的总线数据加密使能 使能后，硬件自动连接 SPI 控制器。
15:14	–	–	–
13:12	R/W	0x0	SPI_SEL SPI 控制器选择 选择连接的总线，加密器对所选总线上的数据进行加解密。 • 0x0：BYPASS，不连接任何总线 • 0x1：连接 SPI0 • 0x2：连接 SPI1 XIP_EN 时，本字段设置无效。
11:1	–	–	–
0	R/W	0x0	KEY_START 开始计算分组密钥 数据加解密开始之前或者同时，必须先计算分组密钥，否则加解密单元会一直等待密钥生成，导致传输失败。传输完成，需要清零以停止密钥计算。 • 0：停止计算分组密钥 • 1：开始计算分组密钥 XIP_EN 时，本字段设置无效。

4.2.6.2. 0x004 SPIE_ICR

默认值：0x00000000			中断控制寄存器（Interrupt Control Register）
位域	类型	默认值	描述
31:6	–	–	–

默认值：0x00000000			中断控制寄存器（Interrupt Control Register）
位域	类型	默认值	描述
5	R/W	0x0	KEY_OVF 分组密钥 FIFO 上溢中断使能
4	R/W	0x0	KEY_UDF 分组密钥 FIFO 下溢中断使能
3	R/W	0x0	READ_HALF_EMPTY 读过半空数据中断使能
2	R/W	0x0	READ_EMPTY_DATA 读空数据中断使能
1	R/W	0x0	ENC_DEC_FINISHED 加解密完成中断使能
0	R/W	0x0	KEY_GEN 第一组分组密钥生成中断使能

4.2.6.3. 0x008 SPIE_ISR

默认值：0x00000000			中断状态寄存器（Interrupt Status Register）
位域	类型	默认值	描述
31:6	–	–	–
5	W1C	0x0	KEY_OVF 分组密钥 FIFO Overflow 中断状态。 • 0：分组密钥 FIFO 未发生上溢 • 1：分组密钥 FIFO 发生上溢
4	W1C	0x0	KEY_UDF 分组密钥 FIFO Underflow 中断状态。 • 0：分组密钥 FIFO 未发生上溢 • 1：分组密钥 FIFO 发生上溢
3	W1C	0x0	READ_HALF_EMPTY 读过半空数据中断，即读取的密文位置的数据，过半都是 0xFF。

默认值：0x00000000			中断状态寄存器（Interrupt Status Register）
位域	类型	默认值	描述
			• 0：中断未发生 • 1：当前读取的密文数据，超过一半都是 0xFF
2	W1C	0x0	READ_EMPTY_DATA 读空数据中断，即读取的密文位置的数据全部为 0xFF，即 Flash 擦除后的状态。 • 0：中断未发生 • 1：当前读取的密文数据，全部为 0xFF 软件可以根据该中断判断当前读取的 Page，是否为擦除后的数据。
1	W1C	0x0	ENC_DEC_FINISHED 加解密完成中断。 • 0：中断未发生 • 1：对密文位置的数据加密或者解密完成 若当前传输的密文长度为 0 时，该中断不会产生。
0	W1C	0x0	KEY_GEN 第一组分组密钥生成的中断状态。 • 0：中断未发生 • 1：第一组分组密钥已经产生 SPI 控制器可以开始传输数据。 在数据传输前，应确保该中断状态位为 1。 由于 ENC 模块产生密钥的速度比 SPI 数据传输的速度快，因此可以保证传输过程中，密钥总是能跟上数据传输的速度。

4.2.6.4. 0x00C SPIE_KCNT

默认值：0x00000000			密钥计数寄存器（Key Counter Register）
位域	类型	默认值	描述
31:0	RO	0x0	KEY_CNTSPI ENC 输出对应密文区域的分组密钥个数，单位为字节。

4.2.6.5. 0x010 SPIE_OCNT

默认值：0x00000000			输出计数寄存器（Output Counter Register）
位域	类型	默认值	描述
31:0	RO	0x0	OUTPUT_CNT

默认值：0x00000000			输出计数寄存器（Output Counter Register）
位域	类型	默认值	描述
			SPI ENC 输出的分组密钥个数，以字节为单位计数。 SPI ENC 为每一个传输的 SPI 数据输出分组密钥，但是如果对应输出的位置是明文，SPI ENC 输出的分组密钥为 0。所以此处计数是输出的 0 的个数和密文对应的分组密钥个数之和。

4.2.6.6. 0x014 SPIE_ADDR

默认值：0x00000000			数据地址寄存器（Data Address Register）
位域	类型	默认值	描述
31:0	R/W	0x0	DATA_ADDR 本次访问的 Flash 数据地址。从 0 开始，是相对 Flash 开始位置的访问偏移值，单位为字节。 该地址信息有两个用途： 1. 计算数据分组对应的 Counter 值 2. 输出分组密钥数据 XIP_EN 时，本字段设置无效。

4.2.6.7. 0x018 SPIE_TWEAK

默认值：0x00000000			计数调整寄存器（Tweak Register）
位域	类型	默认值	描述
31:0	R/W	0x0	CTR_TWEAK 用于产生数据分组的 Counter 值。数据分组的 Counter 值由 eFuse 中的 SPI_ENC_NONCE、SPIE_ADDR 以及 TWEAK 值共同产生，其中 NONCE 与数据的分组地址是固定的，TWEAK 值软件可调。 通过配置 TWEAK 值，软件可以根据实际情况调整 Counter 值，以免每次产生的分组密钥相同。

4.2.6.8. 0x01C SPIE_CPOS

默认值：0x00000000			密文位置寄存器（Ciphertext Position Register）
位域	类型	默认值	描述
31:0	R/W	0x0	CIPHERTEXT_POS

默认值：0x00000000			密文位置寄存器（Ciphertext Position Register）
位域	类型	默认值	描述
			本次传输的密文数据开始位置 用于指示本次传输的数据中密文数据所在的偏移位置。如果当前传输包含了发送和接收，则位置总是从发送开始计算。 值 M：表示本次传输的数据，从第 M 个字节开始是密文 XIP_EN 时，本字段设置无效。

4.2.6.9. 0x020 SPIE_CLEN

默认值：0x00000000			密文长度寄存器（Ciphertext Length Register）
位域	类型	默认值	描述
31:0	R/W	0x0	CIPHERTEXT_LEN 本次传输的密文数据长度。 值 N：表示本次传输的密文数据为 N 字节 XIP_EN 时，本字段设置无效。

4.2.6.10. 0xFFC SPIE_VER

默认值：0x00000101			SPIENC 版本（Version Register）
位域	类型	默认值	描述
31:0	RO	0x00000101	Version（模块版本号） 采用 BCD 码显示，V1.1

4.3. Secure ID (SID)

Secure ID (SID) 模块主要用于读写 eFuse，包括每颗芯片独立 ID、每颗芯片量产校准以及安全管理用到的 KEY 和 HASH 码等信息。

4.3.1. 特性说明

- 1Kbit eFuse 双备份实现 512bit 空间，软件读写按照1Kbit 空间，以 32 bits 为单位
- 512bit eFuse Buffer，上电复位放开后读取 eFuse Cell 到 eFuse Buffer
- 512bit 空间，以 32 bits 为单位，可烧写禁止读操作或写操作
- 提供 SPI_ENC 烧录的 KEY

- 支持烧写关闭 JTAG 功能，并可通过安全认证再打开 JTAG
- 提供 128-bit Chip ID，每颗芯片唯一
- 提供量产测试电阻电压校准值
- 每次烧写之间需要延迟至少 3 us 以上

4.3.2. 原理框图

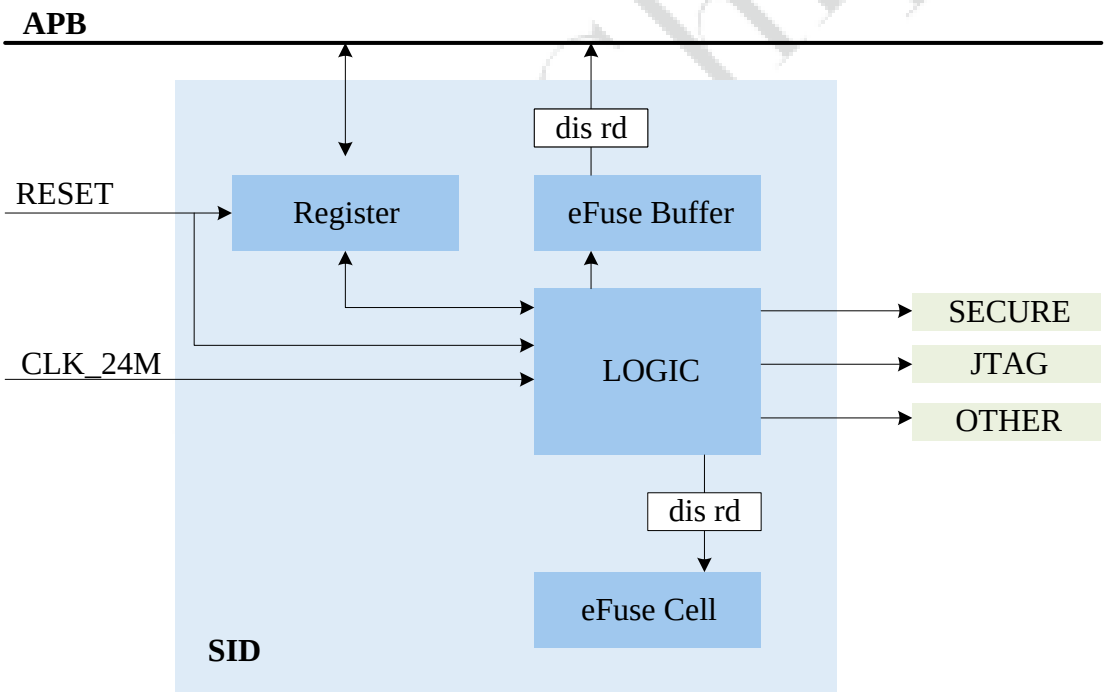


图 4-8 SID 原理框图

4.3.3. 功能描述

4.3.3.1. eFuse 和 SRAM 空间映射

软件读写 eFuse 空间，在 1Kbit 范围内以 32 bits 为单位读写。

为提升可靠性，采用双备份方案，在对 eFuse 进行烧写时，建议先后往 eFuse word[n] 和 word[64+n] 写入相同的值。

开机时，硬件自动读取 eFuse Cell 到 eFuse Buffer 空间，针对两个 word 空间进行按位或操作， 任意一位为 1 则 eFuse Buffer 对应位结果为 1，起到双备份的目的， 如 图 4-9 : eFuse Cell 和 eFuse Buffer 空间映射所示。

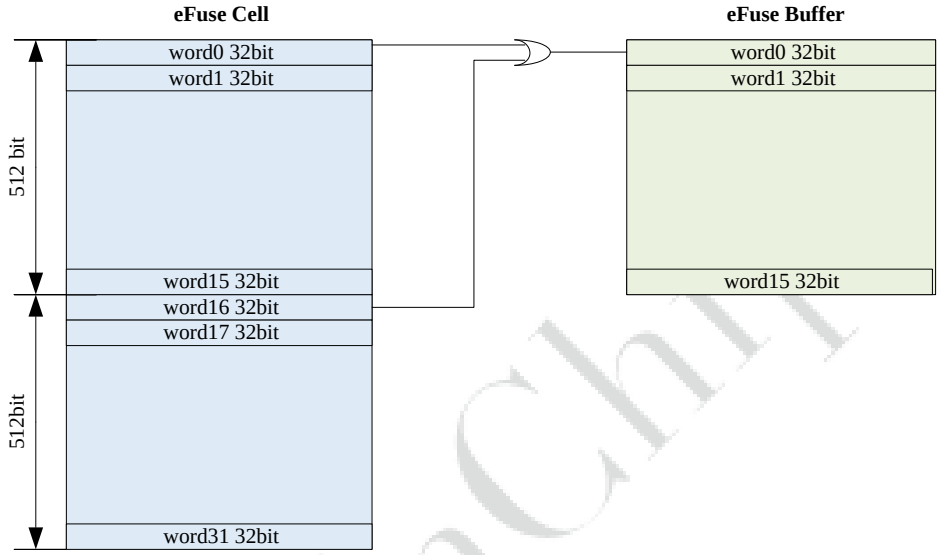


图 4–9 eFuse Cell 和 eFuse Buffer 空间映射

4.3.3.2. 512bit 空间映射定义

地址	用途	位数	禁止位	禁写	禁读	归属	备注
0~3	DIS	32	0	–	–	CSTM	15:00 eFuse 读禁止配置 31:16 eFuse 写禁止配置 每位对应 1 个 32bits 空间
4~7	BROM_SECURE	32	1	Y	–	CSTM	BROM 与安全启动配置
8~F	CALI_LOCK	64	2~3	Y	–	AIC	模拟电路校准参数与模块锁
10~1F	CHIP ID	128	4~7	Y	–	AIC	唯一芯片识别码
20~2F	SSK	128	8~11	Y	Y	CSTM	AES–128 对称加密密钥/SPI_ENC_KEY
30~37	CALI_PGA	64	12~13	–	–	AIC	模拟电路校准参数
38~3F	Reserved	64	14~15	–	–	CSTM	OEM 可自定义使用

4.3.3.3. CALI 区域定义

表 4–18 偏移地址: 0x04

比特位	名称	描述	备注
31:28	LDO18_CALI_VAL	LDO18 校准值	–
27:24	LDO11_CALI_VAL	LDO11 校准值	–
23:0	BROM_SECURE	BROM 和安全	–

表 4–19 偏移地址: 0x08

比特位	名称	描述	备注
31:28	THS_ENV_VAL	CP 校准记录环境温度: 25 ± THS_ENV_VAL	–
27:14	THS1_ADC_VAL	THS1 读得的数值	–
13:0	THS0_ADC_VAL	THS0 读得的数值	–

表 4–20 偏移地址: 0x0C

比特位	名称	描述	备注
31:15	LOCK	规格锁定使用	–
14:8	OSC24_OUT_TR	OSC24M 校准值	–
7:0	LDO25_BG_CTRL	LDO25 BG 校准值	–

表 4–21 偏移地址: 0x30

比特位	名称	描述	备注
31:24	PGA1_N_CALSEL	PGA1_N 校准值	–
23:16	PGA1_P_CALSEL	PGA1_P 校准值	–
24:8	PGA0_N_CALSEL	PGA0_N 校准值	–

表 4–21 偏移地址: 0x30 (续)

比特位	名称	描述	备注
7:0	PGA0_P_CALSEL	PGA0_P 校准值	–

表 4–22 偏移地址: 0x34

比特位	名称	描述	备注
31:24	PGA3_N_CALSEL	PGA3_N 校准值	–
23:16	PGA3_P_CALSEL	PGA3_P 校准值	–
24:8	PGA2_N_CALSEL	PGA2_N 校准值	–
7:0	PGA2_P_CALSEL	PGA2_P 校准值	–

4.3.3.4. BROM_SECURE 区域定义

偏移地址: 0x04

比特位	名称	描述
31:22	FOR_CALI	校准使用
21	SDMC_400K	设置 SDMC 使用 400 KHz 频率进行初始化，默认 100 KHz。
20:12	–	–
11	RSVD	预留
10	CHECKSUM_DIS	固件校验不打开 • 0：固件校验功能使能 • 1：固件校验功能关闭
9	RSVD	预留
8	PLL_FRA0_DIS	启动时不使能 PLL_FRA0
7	PLL_INT0_DIS	启动时不使能 PLL_INT0
6	SPI_DMA_DIS	从 SPI 设备启动时不使用 DMA
5	SDMC_DMA_DIS	从 SDMC 设备启动时不使用 DMA
4	SKIP_SD_UDISK	启动前跳过 SD 卡/U 盘

比特位	名称	描述
3	JTAG_DIS	彻底禁止 JTAG 调试功能
2	SPI_ENC_EN	使能 SPI 总线数据加密功能
1	ENCRYPT_BOOT_EN	使能固件加密启动功能
0	SECURE_BOOT_EN	使能安全启动功能（HMAC 签名校验）

4.3.4. 编程指南

4.3.4.1. 初始化流程

每次系统上电复位后，SID 自动读取所有 eFuse Cell 值到 eFuse Buffer，软件只需读取 eFuse Buffer 对应地址值即可。

> **注：**
>
> 第一次烧写后，在不断电的情况下，eFuse Buffer 值不会自动更新，需通过软件读操作将对应地址的值读到寄存器。

4.3.4.2. eFuse 烧写流程

1. 检查 eFuse 控制寄存器 0x000 EFUSE_CTL 状态位，若为非空闲则等待，若都为空闲则继续第 2 步。状态位包括：
   - EFUSE_WRITE_START
   - EFUSE_READ_START
   - EFUSE_AUTO_STATUS
2. 配置要烧写的 eFuse 地址到 eFuse 地址寄存器 0x004 EFUSE_ADDR。
3. 配置要烧写的 eFuse 数据到 eFuse 写数据寄存器 0x008 EFUSE_WDATA。
4. 在 eFuse 控制寄存器 0x000 EFUSE_CTL 中，配置 EFUSE_WRITE_START 为 1 且在 EFUSE_OP_CODE 位写入 0xA1C，启动烧写流程。
5. 等待 EFUSE_WRITE_START 自动清零，硬件自动关闭 LDO。

为达到双备份目的，要求针对备份地址，即偏移 512bit 地址，重复步骤 1~步骤 5 完成烧写。

4.3.4.3. eFuse 读取流程

1. 检查 eFuse 控制寄存器 0x000 EFUSE_CTL 状态位，若为非空闲则等待，若都为空闲则继续第 2 步。状态位包括：
   - EFUSE_WRITE_START
   - EFUSE_READ_START
   - EFUSE_AUTO_STATUS

2. 配置要读取的 eFuse_cell 地址到 eFuse 地址寄存器 0x004 EFUSE_ADDR 。
3. 在 eFuse 控制寄存器 0x000 EFUSE_CTL 中，配置 EFUSE_READ_START 为 1 且在 EFUSE_OP_CODE 位写入 0xA1C，启动读取流程。
4. 等待 EFUSE_READ_START 自动清零，即可读取 eFuse 读数据寄存器 (0x00C EFUSE_RDATA) 并获得 eFuse 的值。

4.3.5. 寄存器列表

表 4–23 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	EFUSE_CTL	eFuse 控制（eFuse Control）	0x000 EFUSE_CTL
0x004	EFUSE_ADDR	eFuse 地址（eFuse Address）	0x004 EFUSE_ADDR
0x008	EFUSE_WDATA	eFuse 写数据（eFuse Write Data）	0x008 EFUSE_WDATA
0x00C	EFUSE_RDATA	eFuse 读数据（eFuse Read Data）	0x00C EFUSE_RDATA
0x010	EFUSE_TIMING_LOW	eFuse 时序低位（eFuse Timing）	0x010 EFUSE_TIMING_LOW
0x020	EFUSE_TIMING_HIGH	eFuse 时序高位（eFuse Timing）	0x020 EFUSE_TIMING_HIGH
0x0FC	SID_VER	SID 版本（SID Version）	0x0FC SID_VER
0x200~0x23F	EFUSE_BUFFER	eFuse 数据 （EFUSE DATA）	0x200~0x23F EFUSE_BUFFER

4.3.6. 寄存器描述

4.3.6.1. 0x000 EFUSE_CTL

默认值：0x00000200			eFuse 控制（eFuse Control）
位域	类型	默认值	描述
31:30	–	–	–
29	W1C	0x0	SID_ERROR

默认值：0x00000200			eFuse 控制（eFuse Control）
位域	类型	默认值	描述
			SID 错误 • 0：正常 • 1：出现过 Busy 时启动读写
28	–	–	–
27:16	WO	0x0	EFUSE_OP_CODE eFuse 操作码 为防止误操作，设置为写入 0xA1C 后允许 eFuse 进行读写操作。
15:14	–	–	–
13	RO	0x1	EFUSE_DATA_READY eFuse 数据就绪 • 0：eFuse 数据读取中 • 1：eFuse 数据准备就绪
12:8	RO	0x2	EFUSE_STATUS eFuse 状态机状态 • 1：开机自动读取中 • 2：空闲中 • 4：烧录中 • 8：读取中 • 空闲状态建议同步查询 EFUSE_DATA_READY。 • 写结束状态建议查询 EFUSE_WRITE_START。 • 读结束状态建议查询 EFUSE_READ_START。
7:5	–	–	–
4	R/WAC	0x0	EFUSE_READ_START eFuse 读启动寄存器 • 0：无操作 • 1：启动 eFuse 读操作 写该位时与 EFUSE_OP_CODE = 0xA1C 同时写入。写 1 启动 eFuse 读取，读取完该位自动清零。
3:1	–	–	–

默认值：0x00000200			eFuse 控制（eFuse Control）
位域	类型	默认值	描述
0	R/WAC	0x0	EFUSE_WRITE_START eFuse 写启动寄存器： • 0：无操作 • 1：启动 eFuse 写操作 写该位时与 EFUSE_OP_CODE = 0xA1C 同时写入。 写 1 启动 eFuse 烧写，烧写完该位自动清零。期间会自动开关 LDO，无需配置开关 LDO。

4.3.6.2. 0x004 EFUSE_ADDR

默认值：0x00000000			eFuse 地址（eFuse Address）
位域	类型	默认值	描述
31:9	–	–	–
8:0	R/W	0x0	EFUSE_ADDR eFuse 读写地址寄存器 全空间读写，以 byte 为单位，由于读写操作按照 32 位进行，低两位需配置为 0。

4.3.6.3. 0x008 EFUSE_WDATA

默认值：0x00000000			eFuse 写数据（eFuse Write Data）
位域	类型	默认值	描述
31:0	R/W	0x0	EFUSE_WDATA eFuse 写数据寄存器 **注：** eFuse 为单次烧写，默认未烧写时为 0，烧写为 1，后无法再变回为 0。该字段对应的 bit 为 1，则对应进行烧写，对应的 bit 为 0 即不进行烧写， 32 bits 可以分多次针对不同 bits 进行烧写。

4.3.6.4. 0x00C EFUSE_RDATA

默认值：0x00000000			eFuse 读数据（eFuse Read Data）
位域	类型	默认值	描述
31:0	RO	0x0	EFUSE_RDATA eFuse 读数据寄存器

4.3.6.5. 0x010 EFUSE_TIMING_LOW

默认值：0x04026536			eFuse 时序低位（eFuse Timing）
位域	类型	默认值	描述
31:24	R/W	0x04	TAEN_RD 单次读数据处理时间所占的时钟周期数
23:16	R/W	0x02	TRD 单次读数据时高电平所占的时钟周期数
15:8	R/W	0x65	TAEN_PGM 单次写数据处理时间所占的时钟周期数
7:0	R/W	0x36	TPGM 单次写数据时高电平所占的时钟周期数

4.3.6.6. 0x020 EFUSE_TIMING_HIGH

默认值：0x00000100			eFuse 时序高位（eFuse Timing）
位域	类型	默认值	描述
31:16	–	–	–
9:8	R/W	0x1	TAEN_PGM 单次写数据处理时间所占的时钟周期数。以 EFUSE_TIMING_LOW 寄存器的 [15]–[8] 位为低位与该值进行拼接得到正确值，例如 TAEN_PGM 正确默认值为 0x11f。
7:2	–	–	–
1:0	R/W	0x0	TPGM 单次写数据时高电平所占的时钟周期数。以 EFUSE_TIMING_LOW 寄存器的 [7]–[0] 位为低位与该值进行拼接得到正确值，例如 TPGM 正确默认值为 0x0f1。

4.3.6.7. 0x0FC SID_VER

默认值：0x00000106			SID 版本（SID Version）
位域	类型	默认值	描述
31:0	RO	0x00000106	VERSION 模块版本。 V1.6

4.3.6.8. 0x200~0x23F EFUSE_BUFFER

默认值：0x00000000			eFuse 数据（EFUSE DATA）
位域	类型	默认值	描述
31:0	RO	0x0	EFUSE_BUFFER eFuse 数据 **芯片上电复位，会将 eFuse Cell 值同步到该区域。** **配置寄存器发起读操作，也会将 eFuse 值同步到该区域。**

4.4. Data Check Engine (DCE)

Data Check Engine (DCE) 模块是一种用于数据校验的硬件加速器，主要支持 CRC 校验。

4.4.1. 特性说明

- 支持可配置的 CRC 多项式
- 支持可配置的 CRC 初始值
- 支持可配置的 CRC 结果异或值
- 支持 CRC 输入输出数据反转可配置

4.4.2. 功能描述

4.4.2.1. CRC 校验算法配置

CRC 校验算法的配置流程如下：

1. 通过 CRC_POLY 寄存器配置 CRC 多项式。
2. 通过 CRC _INIT 寄存器配置 CRC 初始值。
3. 通过 CRC _XOROUT 寄存器配置 CRC 结果异或值，输出结果前会与该值进行异或。

4. 通过寄存器配置是否反转输入和输出数据，反转方式选择如下：

   • Bit In Word 每个 Word 里的 Bit 进行逆序处理。

注：

关于输入输出反转运算的详细说明，可查看 CRC 算法的反转运算说明。

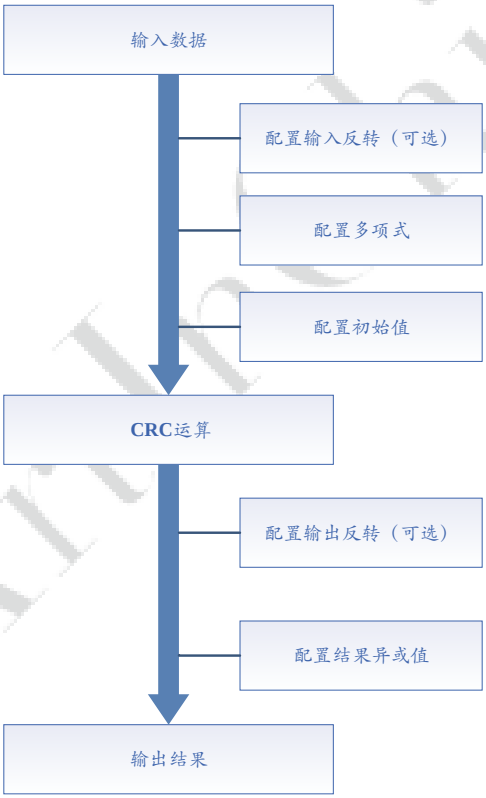


图 4–10 CRC 使用流程

4.4.2.2. CRC 算法的反转运算说明

CRC 算法涉及的反转运算的详细说明如下图所示：

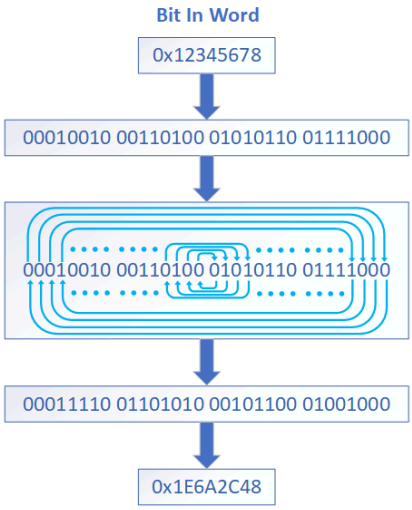


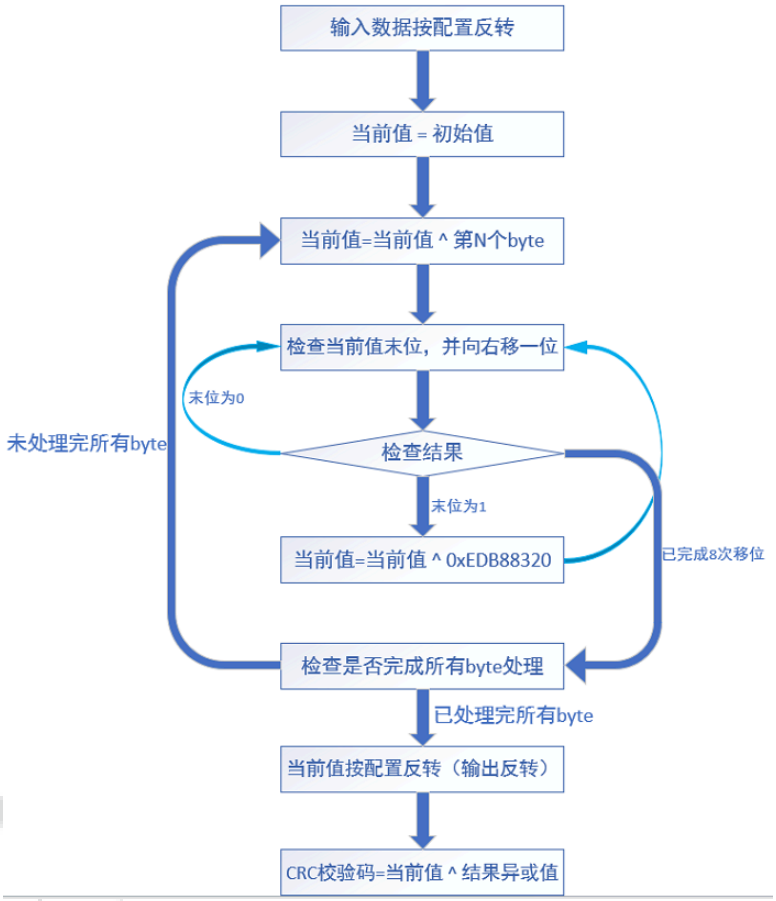
图 4–11 Bit In Word 反转

4.4.2.3. CRC 硬件运算流程

CRC 硬件运算流程如下所示：

1. 按照配置，选择对输入数据进行输入反转或不反转。
2. 设置 CRC 多项式和多项式长度。
3. 设置 CRC 初始值。初始值长度与多项式长度一致。
4. 将输入数据的第 n 个 Byte 与当前值进行异或，从第一个 Byte 开始。
5. 检查当前值最低 bit 是否为 1，并将该值向右移一位，末位丢弃，高位补零。
6. 如果最低 bit 为 0，重复 5 步；否则，将当前值与多项式 进行异或。
7. 重复 5 到 6 步，直到总共完成了 8 次移位。至此，输入数据的第一个 Byte 处理完毕。
8. 重复 5 – 7 步，对输入数据的下一个 Byte 进行相同的处理，直至所有数据全部处理完毕。
9. 所有数据处理完毕后，当前值按照配置进行输出反转或不反转。
10. 将经过上面所有步骤处理后的当前值与结果异或值进行异或运算得出的即为最终的 CRC 校验码。

以多项式为 0x4C11DB7 为例，图示如下：



4.4.3. 寄存器列表

表 4–24 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	DCE_CTL	DCE 控制（DCE_CTL）	0x000 DCE_CTL
0x004	DCE_CFG	DCE 配置（DCE_CFG）	0x004 DCE_CFG
0x008	IRQ_EN	中断控制寄存器（IRQ_EN）	0x008 IRQ_EN
0x00C	IRQ_STA	中断状态寄存器（IRQ_STA）	0x00C IRQ_STA
0x0014	DATA_LEN	数据长度配置寄存器（DATA_LEN）	0x0014 DATA_LEN
0x0018	CRC_POLY	CRC 多项式配置寄存器（CRC_POLY）	0x0018 CRC_POLY
0x001C	CRC_SIZE	CRC 多项式大小配置寄存器（CRC_SIZE）	0x001C CRC_SIZE

表 4–24 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0040	DATA_INPUT	数据输入寄存器（DATA_INPUT）	0x0040 DATA_INPUT
0x080	CRC_CFG	CRC 配置寄存器（CRC_CFG）	0x080 CRC_CFG
0x0084	CRC_INIT	初始值配置寄存器（CRC_INIT）	0x0084 CRC_INIT
0x0088	CRC_XOROUT	结果异或值配置寄存器（CRC_XOROUT）	0x0088 CRC_XOROUT
0x00C0	CRC_RESULT	结果存储寄存器（CRC_RESULT）	0x00C0 CRC_RESULT
0x0FFC	VERSION	DCE 版本号（VERSION）	0x0FFC VERSION

4.4.4. 寄存器描述

4.4.4.1. 0x000 DCE_CTL

默认值：0x00000000			DCE 控制（DCE_CTL）
位域	类型	默认值	描述
31:1	–	–	–
0	R/WAC	0x0	CAL_ENABLE 计算使能 • 0：关闭 • 1：计算使能 当计算完成并将结果存到 Result 寄存器后，该位自动清零。可以通过读取该位表示 DCE 模块当前状态。

4.4.4.2. 0x004 DCE_CFG

默认值：0x00000000			DCE 配置（DCE_CFG）
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	DAT_SRC_SEL

默认值：0x00000000			DCE 配置（DCE_CFG）
位域	类型	默认值	描述
			Data Source Select，数据源选择 • 0x0：Reserved • 0x1：读取寄存器 DATA_INPUT 的数据作为输入数据
3:0	R/W	0x0	MODE_SEL 算法选择 • 0x01：CRC

4.4.4.3. 0x008 IRQ_EN

默认值：0x0			中断控制寄存器（IRQ_EN）
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	CRC_FINISH_ENCRC 运算完成中断使能 • 0: 关闭中断 • 1: 打开中断

4.4.4.4. 0x00C IRQ_STA

默认值：0x0			中断状态寄存器（IRQ_STA）
位域	类型	默认值	描述
31:1	–	–	–
0	R/W1C	0x0	CRC_FINISH_STA CRC 运算完成中断状态 • 0: 无中断 • 1: 有中断，表示运算完成

4.4.4.5. 0x0014 DATA_LEN

默认值：0x0			数据长度配置寄存器（DATA_LEN）
位域	类型	默认值	描述
31:29	–	–	–
28:0	R/W	0x0	此位仅在 DAT_SRC_SEL = 0x1 时有效，单位为 bit，支持 0~32 bit。

4.4.4.6. 0x0018 CRC_POLY

默认值：0x4C11DB7			CRC 多项式配置寄存器（CRC_POLY）
位域	类型	默认值	描述
31:0	R/W	0x4C11DB7	CRC_POLY CRC Polynomial Set，CRC 多项式设置，计算 CRC 前配置此寄存器，为了避免 CRC 错误，请不要随意操作此寄存器。 举例，此处默认值为 0x4C11DB7，如果 POLY_SIZE 设置多项式为 32-bit，那么这里对应的 CRC 多项式为：$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X^{1}+X^{0}$

4.4.4.7. 0x001C CRC_SIZE

默认值：0x1F			CRC 多项式大小配置寄存器（CRC_SIZE）
位域	类型	默认值	描述
31:5	–	–	–
4:0	R/W	0x1F	POLY_SIZE CRC Polynomial Size Set，CRC 多项式大小设置。 • 0x0：1-bit 多项式 • 0x1：2-bit 多项式 • 0x2：3-bit 多项式 • ... • 0x1F：32-bit 多项式

4.4.4.8. 0x0040 DATA_INPUT

默认值：0x0			数据输入寄存器（DATA_INPUT）
位域	类型	默认值	描述
31:0	R/W	0x0	DATA_INPUT Data Input，输入数据寄存器，此位只有在 DAT_SRC_SEL = 0x1 的时候有效

4.4.4.9. 0x080 CRC_CFG

默认值：0x0000000b			CRC 配置寄存器（CRC_CFG）
位域	类型	默认值	描述
31:3	–	–	–
2	R/W	0x0	INPUT_BIT_IN_WORD_REV 输入数据反转，指输入数据每个 Word 里的 Bit 进行逆序处理 • 0：禁用 • 1：使能
1	–	–	–
0	R/W	0x1	OUTPUT_BIT_IN_WORD_REV 输出数据反转，指输出数据每个 Word 里的 Bit 进行逆序处理 • 0：禁用 • 1：使能

4.4.4.10. 0x0084 CRC_INIT

默认值：0xFFFFFFFF			初始值配置寄存器（CRC_INIT）
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	CRC 初始值

4.4.4.11. 0x0088 CRC_XOROUT

默认值：0xFFFFFFFF			结果异或值配置寄存器（CRC_XOROUT）
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	CRC 结果异或值

4.4.4.12. 0x00C0 CRC_RESULT

默认值：0x0			结果存储寄存器（CRC_RESULT）
位域	类型	默认值	描述
31:0	RO	0x0	CRC 运算结果

4.4.4.13. 0x0FFC VERSION

默认值：0x00000110			DCE 版本号（VERSION）
位域	类型	默认值	描述
31:0	RO	0x00000110	Version 模块版本号，采用 BCD 码显示，V1.1

5. 时钟和电源

5.1. Clock Manage Unit (CMU)

Clock Management Unit (CMU) 是一个关键的系统组件，用于配置系统时钟，包括 PLL 频率和展频、AXI/ AHB/ APB 总线时钟、各模块输入时钟、以及 IO 输出时钟。通过对系统时钟的精确控制和管理，包括时钟信号的频率与分配，CMU 确保了系统运行的效率与稳定性。

5.1.1. 特性说明

- PLL_INT 1 个 ，无小数分频和展频功能，可旁路直接输出 OSC _24M 时钟
- PLL_FRA 1 个 ，带小数分频和展频功能，可旁路直接输出 OSC _24M 时钟
- CPU 时钟可选 CLK24M 或 PLL _INT 或 PLL _FRA 进行 一到十六分频
- AXI/AHB 时钟为 CPU 时钟的 2 分频
- APB0 时钟为 AHB 的二分频，APB1 时钟为 CLK24M 时钟
- 每个模块总线时钟、模块时钟、复位三个开关可独立配置
- 两路可配置频率时钟输出，用作于外设的时钟输入

5.1.2. 原理框图

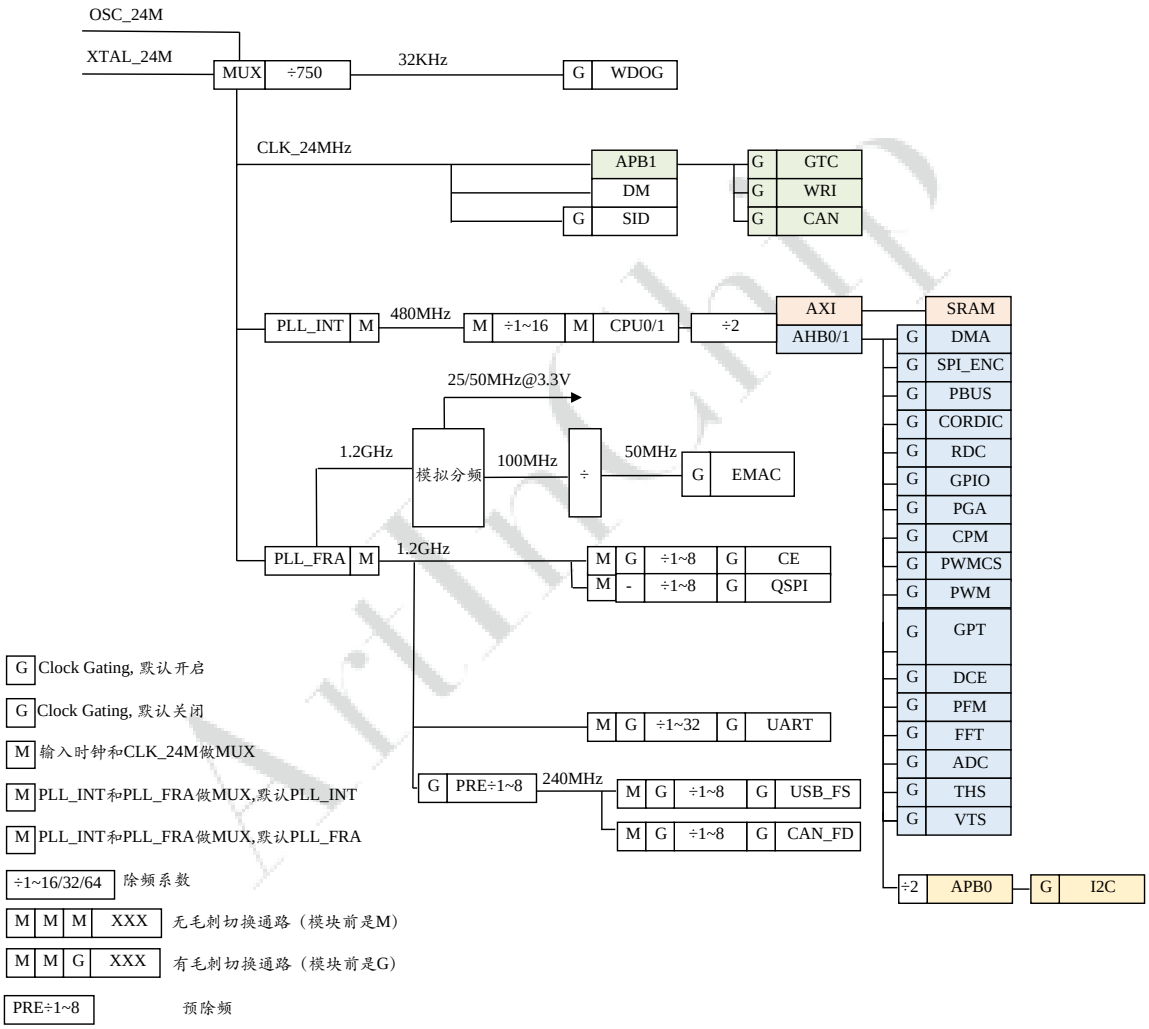


图 5–1 CMU 原理框图

5.1.3. 功能描述

5.1.3.1. PLL

在 CMU 中，有两个模拟电路 PLL 用于产生稳定且精确的时钟供给整个芯片，如下表所示：

表 5–1 PLL 用途及特性

名称	用途	典型频率	展频
PLL_INT	CPU/ AXI/ AHB/APB0/ DMA/ PWMCS/ SDFM/ PWM/ FFT	480 MHz	不支持
PLL_FRA	UART/ CE/ CANFD/ EMAC/ QSPI	1200 MHz	支持

PLL 内部结构如下图所示， PLL 频率计算公式为：PLL_OUT = 24MHz ÷ (P+1) × (N+1+(F ÷ (2^{17}−1))) ÷ (M+1)。

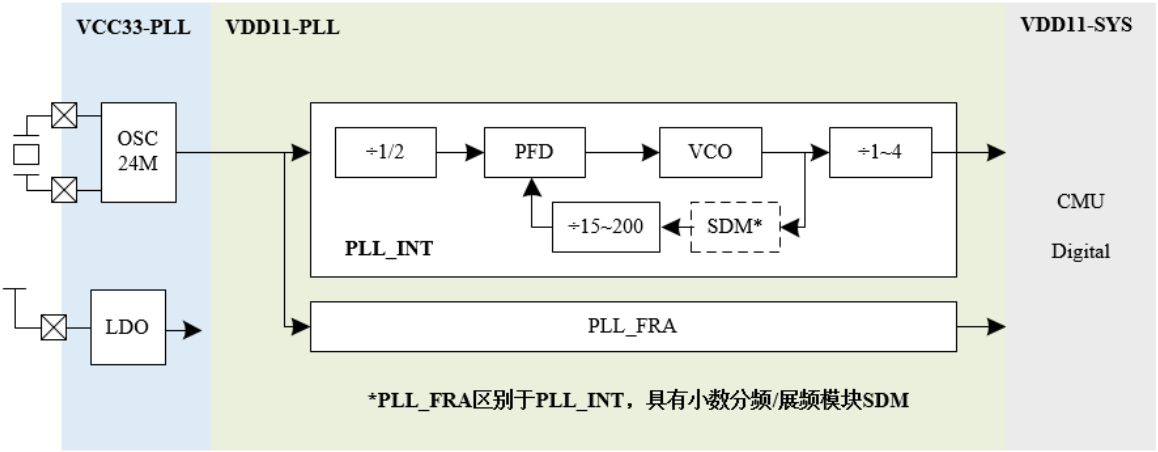


图 5–2 CMU PLL 内部框图

5.1.3.2. CLKOUT

时钟输出 (CLKOUT) 用于输出时钟给芯片外设使用， 总共两路 CLKOUT，可选来源为 PLL_INT 以及 PLL_FRA，可配置 1~258 除频， CLKOUT 通路如下图所示。

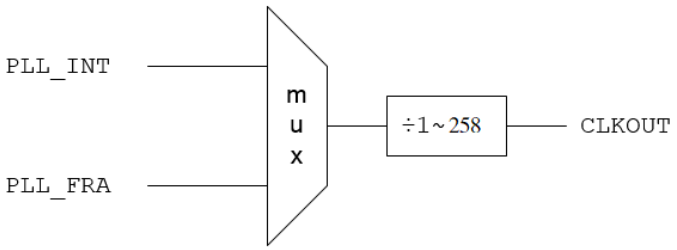


图 5–3 CMU CLKOUT 通路

5.1.3.3. OSC24M 和 XTAL

时钟源输入 CLK_24M 用于给芯片提供时钟，总共两个时钟源选择：

- 内部 OSC24M
- 外部 XTAL

正常使用模式： 软件先配置 xtal_swen 打开引脚连接，延迟 1 ms，再配置 xtal_sel 切换到 xtal 时钟。

强制切换模式： 通过引脚 XTAL_FORCE_N 拉低或者 ETC 配置 xtal_force=1 均可以强制打开引脚和切换时钟。

时钟源选择的规则如下：

- 当引脚 SYS_TEST 配置的值为 1 时，无论 BID_1 为何值，只能选择外部 XTAL 作为芯片时钟源输入。
- 当引脚 SYS_TEST 配置的值为 0 时，根据 BID_1 的值选择芯片的输入时钟源：

- 若 BID_1 值为 1，则选择外部 XTAL 时钟源。
- 若 BID_1 值为 0，则选择内部 OSC24M 时钟源。

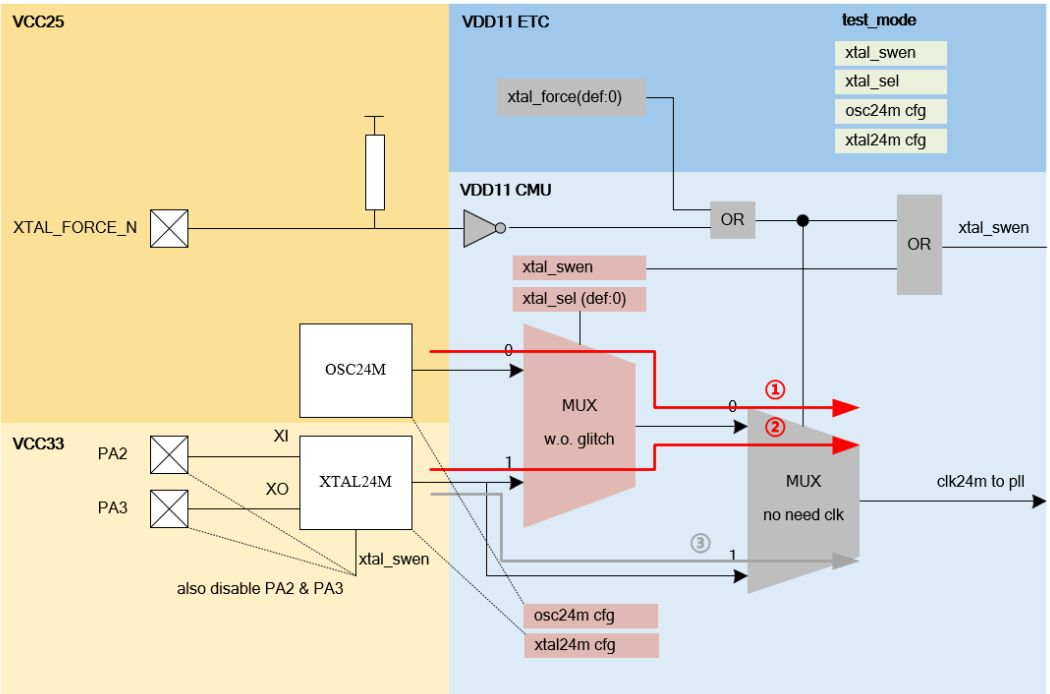


图 5–4 时钟源选择

5.1.3.4. 模块时钟

模块名称	总线时钟	模块时钟源	模块时钟极限频率	备注
E907 CORE	–	PLL_INT	480 MHz	–
E907 PLIC	–	PLL_INT ÷ 2	240 MHz	–
E907 CLINT	–	PLL_INT ÷ 2	240 MHz	–
E907 DM	–	CLK_24M	24 MHz	–
AXI	AXI	–	240 MHz	–
AHB0	AHB0	–	240 MHz	–
AHB1	AHB1	–	240 MHz	–
APB0	APB0	–	120 MHz	–
APB1	APB1	–	24 MHz	–

模块名称	总线时钟	模块时钟源	模块时钟极限频率	备注
BROM	AXI	–	–	–
SRAM	AXI	–	–	–
AHB Matrix	AHB	–	–	–
DMA0	AHB0	–	–	–
DMA1	AHB0	–	–	–
DCE	AHB0	–	–	–
CE	AHB0	PLL_FRA	200 MHz	–
EMAC	AHB0	PLL_FRA	(AHB)	–
SPI0	AHB0	PLL_FRA	266 MHz	–
SPI1	AHB0	PLL_FRA	266 MHz	–
SPI2	AHB0	PLL_FRA	266 MHz	–
SPI3	AHB0	PLL_FRA	266 MHz	–
SPI4	AHB0	PLL_FRA	266 MHz	–
SPI5	AHB0	PLL_FRA	266 MHz	–
PBUS	AHB0	–	–	–
CAN-FD0	AHB0	PLL_FRA	80 MHz	–
CAN-FD1	AHB0	PLL_FRA	80 MHz	–
GPIO	AHB0	–	–	–
USBFS	AHB0	PLL_FRA	48 MHz	48 MHz
PWM	AHB1	–	–	–
PWMCS	AHB1	–	–	–
ADC	AHB1	–	–	–
RDC	AHB1	–	–	–

模块名称	总线时钟	模块时钟源	模块时钟极限频率	备注
CORDIC	AHB1	–	–	–
SDFM	AHB1	–	–	–
THS	AHB1	–	–	–
CPM	AHB1	–	–	–
VTS	AHB1	–	–	–
FFT	AHB1	–	–	–
GPT	AHB1	–	–	–
PGA	AHB1	–	–	–
I2C0	APB0	–	–	–
I2C1	APB0	–	–	–
I2C2	APB0	–	–	–
I2C3	APB0	–	–	–
UART0	APB0	PLL_FRA	120 MHz	–
UART1	APB0	PLL_FRA	120 MHz	–
UART2	APB0	PLL_FRA	120 MHz	–
UART3	APB0	PLL_FRA	120 MHz	–
UART4	APB0	PLL_FRA	120 MHz	–
UART5	APB0	PLL_FRA	120 MHz	–
UART6	APB0	PLL_FRA	120 MHz	–
UART7	APB0	PLL_FRA	120 MHz	–
UART8	APB0	PLL_FRA	120 MHz	–
SYSCFG	APB0	CLK_24MHz	–	–
CMU	APB0	–	–	–

模块名称	总线时钟	模块时钟源	模块时钟极限频率	备注
SPI_ENC	APB0	–	–	–
WDOG	APB1	CLK_32K	32 KHz	–
WRI	APB1	CLK_24M	24 MHz	–
SID	APB1	CLK_24M	24 MHz	–
GTC	APB1	–	–	–

5.1.3.5. 模块开关控制和时序

USB

打开时序：

```
ctrlclk1->phyclk1->100us->phyrst1->ctrlrst1
```

关闭时序：

```
ctrlrst0->phyrst0->phyclk0->ctrlclk0
```

其它模块

打开时序：

```
modclk1->busclk1->rst1
```

关闭时序：

```
rst0->busclk0->modclk0
```

5.1.4. 寄存器列表

表 5–2 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	PLL_INT_GEN	PLL_INT 通用	0x000 PLL_INT_GEN
0x020	PLL_FRA_GEN	PLL_FRA 通用	0x020 PLL_FRA_GEN
0x040	PLL_INT_CFG	PLL_INT 配置	0x040 PLL_INT_CFG
0x060	PLL_FRA_CFG	PLL_FRA 配置	0x060 PLL_FRA_CFG

表 5–2 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x080	PLL_FRA_SDM	PLL_FRA 展频	0x080 PLL_FRA_SDM
0x0A0	PLL_COM	PLL 公共	0x0A0 PLL_COM
0x0A4	PLL_IN	PLL 输入	0x0A4 PLL_IN
0x0A8	PLL_OUT	PLL 输出	0x0A8 PLL_OUT
0x0D0	OSC_XTAL_CNT_CTL	OSC 和 XTAL 计数控制（OSC XTAL Counter Control）	0x0D0 OSC_XTAL_CNT_CTL
0x0D4	OSC_CNT	OSC 计数器（OSC Counter）	0x0D4 OSC_CNT
0x0D8	XTAL_CNT	XTAL 计数器（XTAL Counter）	0x0D8 XTAL_CNT
0x0DC	OSC_CNT_THR	计数器门限（Counter Threshold）	0x0DC OSC_CNT_THR
0x0E0	CLK_OUT0	CLK0 输出	0x0E0 CLK_OUT0
0x0E4	CLK_OUT1	CLK1 输出	0x0E4 CLK_OUT1
0x180	CLK_CPU0	CPU0 时钟	0x180 CLK_CPU0
0x184	RST_ENTRY_CPU0	CPU0 复位入口（CPU0 Reset Entry）	0x184 RST_ENTRY_CPU0
0x18C	DBG_CPU0	CPU0 调试（CPU0 Debug）	0x18C DBG_CPU0
0x190	CLK_CPU1	CPU1 时钟	0x190 CLK_CPU1
0x194	RST_ENTRY_CPU1	CPU1 复位入口（CPU1 Reset Entry）	0x194 RST_ENTRY_CPU1
0x1C0	CLK_PRE_DIV	预除频	0x1C0 CLK_PRE_DIV
0x20C	CLK_WDOG	WDOG 时钟	0x20C CLK_WDOG
0x244	CLK_PWM0	PWM0 时钟	0x244 CLK_PWM0

表 5–2 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x248	CLK_PWM1	PWM1 时钟	0x248 CLK_PWM1
0x24C	CLK_PWM2	PWM2 时钟	0x24C CLK_PWM2
0x250	CLK_PWM3	PWM3 时钟	0x250 CLK_PWM3
0x254	CLK_PWM4	PWM4 时钟	0x254 CLK_PWM4
0x258	CLK_PWM5	PWM5 时钟	0x258 CLK_PWM5
0x400	CLK_DMA0	DMA0 时钟	0x400 CLK_DMA0
0x404	CLK_DMA1	DMA1 时钟	0x404 CLK_DMA1
0x414	CLK_DCE	DCE 时钟	0x414 CLK_DCE
0x418	CLK_CE	CE 时钟	0x418 CLK_CE
0x438	CLK_USB_FS_DR	USB FS DR 时钟	0x438 CLK_USB_FS_DR
0x43C	CLK_USB_FS_DR_PHY	USB_FS_DR 时钟（USB_FS_DR_Clock）	0x43C CLK_USB_FS_DR_PHY
0x440	CLK_EMAC	EMAC 时钟	0x440 CLK_EMAC
0x444	CLK_EMAC1	EMAC1 时钟	#unique_60_Connect_42_section_y54_qqh_kfc
0x448	CLK_CANFD0	CANFD0 时钟	0x448 CLK_CANFD0
0x44C	CLK_CANFD1	CANFD1 时钟	0x44C CLK_CANFD1
0x450	CLK_CAN	CAN 时钟	0x450 CLK_CAN
0x45C	CLK_XSPI	XSPI 时钟	0x45C CLK_XSPI
0x460	CLK_QSPI0	QSPI0 时钟	0x460 CLK_QSPI0
0x464	CLK_QSPI1	QSPI1 时钟	0x464 CLK_QSPI1
0x468	CLK_QSPI2	QSPI2 时钟	0x468 CLK_QSPI2
0x46C	CLK_QSPI3	QSPI3 时钟	0x46C CLK_QSPI3
0x470	CLK_QSPI4	QSPI4 时钟	0x470 CLK_QSPI4

表 5–2 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x474	CLK_QSPI5	QSPI5 时钟	0x474 CLK_QSPI5
0x490	CLK_CORDIC	CORDIC 时钟	0x490 CLK_CORDIC
0x498	CLK_RDC	RDC 时钟	0x498 CLK_RDC
0x49C	CLK_PFM	PFM 时钟	0x49C CLK_PFM
0x4A0	CLK_PBUS	PBUS 时钟	0x4A0 CLK_PBUS
0x4A4	CLK_FFT	FFT 时钟	0x4A4 CLK_FFT
0x800	CLK_SYSCFG	SYSCFG 时钟	0x800 CLK_SYSCFG
0x810	CLK_SPI_ENC	SPI_ENC 时钟	0x810 CLK_SPI_ENC
0x814	CLK_PWMCS	PWMCS 时钟	0x814 CLK_PWMCS
0x818	CLK_ADC	ADC 时钟	0x818 CLK_ADC
0x83C	CLK_GPIO	GPIO 时钟	0x83C CLK_GPIO
0x840	CLK_UART0	UART0 时钟	0x840 CLK_UART0
0x844	CLK_UART1	UART1 时钟	0x844 CLK_UART1
0x848	CLK_UART2	UART2 时钟	0x848 CLK_UART2
0x84C	CLK_UART3	UART3 时钟	0x84C CLK_UART3
0x850	CLK_UART4	UART4 时钟	0x850 CLK_UART4
0x854	CLK_UART5	UART5 时钟	0x854 CLK_UART5
0x858	CLK_UART6	UART0 时钟	0x858 CLK_UART6
0x85C	CLK_UART7	UART7 时钟	0x85C CLK_UART7
0x860	CLK_UART8	UART8 时钟	0x860 CLK_UART8
0x87C	CLK_SDFM	SDFM 时钟	0x87C CLK_SDFM
0x904	CLK_SID	SID 时钟	0x904 CLK_SID

表 5–2 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x90C	CLK_GTC	GTC 时钟	0x90C CLK_GTC
0x960	CLK_I2C0	I2C0 时钟	0x960 CLK_I2C0
0x964	CLK_I2C1	I2C1 时钟	0x964 CLK_I2C1
0x968	CLK_I2C2	I2C2 时钟	0x968 CLK_I2C2
0x96C	CLK_I2C3	I2C3 时钟	0x96C CLK_I2C3
0x9AC	CLK_THS	THS 时钟	0x9AC CLK_THS
0x9B0	CLK_VTS	VTS 时钟	0x9B0 CLK_VTS
0x9E4	CLK_CPM	CPM 时钟	0x9E4 CLK_CPM
0x9F0	CLK_PGA	PGA 时钟	0x9F0 CLK_PGA
0xA00	CLK_GPT0	GPT0 时钟	0xA00 CLK_GPT0
0xA04	CLK_GPT1	GPT1 时钟	0xA04 CLK_GPT1
0xA08	CLK_GPT2	GPT2 时钟	0xA08 CLK_GPT2
0xA0C	CLK_GPT3	GPT3 时钟	0xA0C CLK_GPT3
0xFE8	REG_WR_CFG	寄存器写配置	0xFE8 REG_WR_CFG
0xFEC	REG_WR_DIS_STS	寄存器写失效状态	0xFEC REG_WR_DIS_STS
0xFFC	CMU_MOD_VER	CMU 模块版本	0xFFC CMU_MOD_VER

5.1.5. 寄存器描述

5.1.5.1. 0x000 PLL_INT_GEN

默认值：0x08002710			PLL_INT 通用 (PLL_INT General)
位域	类型	默认值	描述
31:29	–	–	–

默认值：0x08002710			PLL_INT 通用 (PLL_INT General)
位域	类型	默认值	描述
28:24	R/W	0x8	PLL_ICP 锁相环环路带宽调节 数值越小带宽越低
23:21	–	–	–
20	R/W	0x0	PLL_OUT_MUX 锁相环输出选择 • 0： OSC_24M 输出 • 1： PLL 时钟输出 该选择不受其它寄存器影响，PLL 损坏不工作时，时钟可旁路为 OSC_24M 时钟使用。在启动和休眠场景也可使用 OSC_24M 时钟。
19	R/W	0x0	FACTOR_M_EN 锁相环输出除频系数使能 • 0：关闭 • 1：打开
18	R/W	0x0	PLL_OUT_SYS 锁相环输出到系统使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	PLL_EN 锁相环电路使能 • 0：关闭锁相环电路 • 1：打开锁相环电路 锁相环输出频率 PLL_O = 24 / (P + 1) × (N + 1) / (M + 1) MHz，P、N、M 分别为 FACTOR_P、FACTOR_N、FACTOR_M。
15:8	R/W	0x27	FACTOR_N 锁相环倍频系数在应用中，N 配置范围为 14 到 199。
7:6	–	–	–
5:4	R/W	0x1	FACTOR_M 锁相环输出除频系数

默认值：0x08002710			PLL_INT 通用 (PLL_INT General)
位域	类型	默认值	描述
			FACTOR_M ≠0 时，需要配置 FACTOR_M_EN = 1 锁相环才有输出。
3:1	–	–	–
0	R/W	0	FACTOR_P 锁相环输入除频系数

5.1.5.2. 0x020 PLL_FRA_GEN

默认值：0x08003100			PLL_FRA 通用 (PLL_FRA General)
位域	类型	默认值	描述
31:29	–	–	–
28:24	R/W	0x8	PLL_ICP 锁相环环路带宽调节 数值越小带宽越低
23:21	–	–	–
20	R/W	0x0	PLL_OUT_MUX 锁相环输出选择 • 0： OSC_24M 输出 • 1： PLL 时钟输出 该选择不受其它寄存器影响，PLL 损坏不工作时，时钟可旁路为 OSC_24M 时钟使用。在启动和休眠场景也可使用 OSC_24M 时钟
19	R/W	0x0	FACTOR_M_EN 锁相环输出除频系数使能 • 0：关闭 • 1：打开
18	R/W	0x0	PLL_OUT_SYS 锁相环输出到系统使能 • 0：关闭 • 1：打开
17	–	–	–

默认值：0x08003100			PLL_FRA 通用 (PLL_FRA General)
位域	类型	默认值	描述
16	R/W	0x0	PLL_EN 锁相环电路使能 • 0：关闭锁相环电路 • 1：打开锁相环电路 锁相环输出频率 PLL_O = 24 / (P+1) × (N+1) / (M+1) MHz，P、N、M 分别为 FACTOR_P、FACTOR_N、FACTOR_M。
15:8	R/W	0x31	FACTOR_N 锁相环倍频系数在应用中，N 配置范围为 14 到 199。
7:6	–	–	–
5:4	R/W	0x0	FACTOR_M 锁相环输出除频系数 FACTOR_M ≠ 0 时，需要配置 FACTOR_M_EN = 1 锁相环才有输出。
3:1	–	–	–
0	R/W	0	FACTOR_P 锁相环输入除频系数

5.1.5.3. 0x040 PLL_INT_CFG

默认值：0x240C4010			PLL_INT 配置 (PLL_INT Configuration)
位域	类型	默认值	描述
31	–	–	–
30:28	R/W	0x2	PLL_LOCK_TIME
27	–	–	–
26:24	R/W	0x4	PLL_IVCO
23:22	–	–	–
21:20	R/W	0x0	PLL_VCO_SEL
19	R/W	0x1	PLL_VCO_RST

默认值：0x240C4010			PLL_INT 配置 (PLL_INT Configuration)
位域	类型	默认值	描述
18:16	R/W	0x4	PLL_VCO_GAIN
15	–	–	–
14:8	R/W	0x40	PLL_BINT
7	–	–	–
6:0	R/W	0x10	PLL_CINT PLL 初始振荡频率

5.1.5.4. 0x060 PLL_FRA_CFG

默认值：0x00000000			PLL_FRA 配置 (PLL_FRA Configuration)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	DITHER_EN 抖动使能 • 0：关闭 • 1：打开
23:21	–	–	–
20	R/W	0x0	FRA_EN 小数分频使能 • 0：关闭 • 1：打开
19:17	–	–	–
16:0	R/W	0x0	FRA_IN 小数分频系数

5.1.5.5. 0x080 PLL_FRA_SDM

默认值：0x00000000			PLL_FRA 展频 (PLL_FRA Spread Spectrum)
位域	类型	默认值	描述
31	R/W	0x0	SDM_EN 展频使能 • 0：关闭 • 1：打开
30:29	R/W	0x0	SDM_MODE 展频模式 • 0： DC=0 • 1： DC=1 • 2：三角波 • 3：预留
28:20	R/W	0x0	SDM_STEP 展频步进
19	–	–	–
18:17	R/W	0x0	SDM_FREQ 展频频率 • 0: 31.5 KHz • 1: 32.0 KHz • 2: 32.5 KHz • 3: 33.0 KHz
16:0	R/W	0x0	SDM_BOT 展频底线 该位域越小表示幅度越大

5.1.5.6. 0x0A0 PLL_COM

默认值：0x00000009			PLL 公共 (PLL Common)
位域	类型	默认值	描述
31:28	–	–	–

默认值：0x00000009			PLL 公共 (PLL Common)
位域	类型	默认值	描述
27:24	R/W	0x0	PLL_LDO_IBIAS_TUNING
23:21	R/W	0x0	PLL_BIAS_ATB_SELPLL BIAS 模拟测试选择
20	R/W	0x0	PLL_FORCE_LOCK PLL 锁定输出使能
19:17	R/W	0x0	PLL_ATBSEL PLL 测试信号选择
16	R/W	0x0	PLL_ATB_EN PLL 模拟测试使能 • 0：关闭 • 1：打开
3:1	R/W	0x4	LDO_VSET LDO 电压设置 • 0: 0.90 V • 1: 0.95 V • 2: 1.00 V • 3: 1.05 V • 4: 1.10 V • 5: 1.15 V • 6: 1.20 V • 7: 1.25 V
0	R/W	0x1	LDO_EN LDO 使能 • 0：关闭 • 1：打开

5.1.5.7. 0x0A4 PLL_IN

默认值：0xF1004003			PLL 输入 (PLL Input)
位域	类型	默认值	描述
31:30	R/W	0x3	XTAL_GM24M 晶体驱动调节值 越大表示驱动越强
29	R/W	0x1	XTAL_START24M 晶体启动 • 0：关闭 • 1：打开 该寄存器在晶体起振后可以关闭，节省功耗
28	R/W	0x1	XTAL_EN 振荡使能 • 0：关闭 • 1：打开
24	RO	0x1	PLL_IN_TYPE PLL 输入时钟类型 • 0：XTAL • 1：OSC24M
23:20	–	–	–
14:8	R/W	0x40	OSC_OUT_TR24M 振荡器频率配置 • 0：12 MHz • · · · • 64: 24 MHz • · · · • 127: 36 MHz **每一档对应 187.5 KHz。**
7	–	–	–
6:4	–	–	–
3:2	–	–	–

默认值：0xF1004003			PLL 输入 (PLL Input)
位域	类型	默认值	描述
1	R/W	0x1	OSC24M_ENOSC24M 振荡使能 • 0：关闭 • 1：打开
0	R/W	0x1	PLL_IN_SEL(CLK_24) 时钟选择 • 0：XTAL • 1：OSC24M 注： 使用外挂晶振时，需先配置 XTAL _EN=1，延迟 1 ms，再配置 PLL_IN_SEL=0

5.1.5.8. 0x0A8 PLL_OUT

默认值：0x00000002			PLL 输出 (PLL Output)
位域	类型	默认值	描述
31:7	–	–	–
6:4	R/W	0x0	ANA_PIN_DIV 模拟时钟驱动能力 • 0：180Ω • 1：90Ω • 2：60Ω • 3：45Ω • 4：36Ω • 5：30Ω • 6：26Ω • 7：23Ω
3	–	–	–
2	R/W	0x0	ANA_CLK_DIV_EN 模拟时钟前级使能

默认值：0x00000002			PLL 输出 (PLL Output)
位域	类型	默认值	描述
			• 0：关闭 • 1： 打开
1	R/W	0x1	ANA_CLK_DIV_PRE ANA 时钟前级除频 • 0：1 分频（FRA 为 600M 时） • 1：2 分频（FRA 为 1200M 时）
0	R/W	0x0	ANA_CLK_OUT_SEL ANA 输出时钟选择 • 0：25 MHz • 1：50 MHz

5.1.5.9. 0x0D0 OSC_XTAL_CNT_CTL

默认值：0x0000000C			OSC 和 XTAL 计数控制（OSC XTAL Counter Control）
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	COUNTER_MODE 计数器模式。测量模式，当启动 OSC 计数器时，两个计数器同时启动，并在 OSC 达到 CNT_THR 时同时停下。计数模式，两个计数器分别启动和停止，计数不受其它限制。 • 0： 测量模式 • 1： 计数模式
3	R/W1C	0x1	XTAL_COUNTER_RST XTAL 计数复位 • 0： 开始复位 • 1： 复位释放
2	R/W1C	0x1	OSC_COUNTER_RST OSC 计数复位 • 0： 开始复位 • 1： 复位释放
1	R/W	0x0	XTAL_COUNTER_EN

默认值：0x0000000C			OSC 和 XTAL 计数控制（OSC XTAL Counter Control）
位域	类型	默认值	描述
			XTAL 计数使能 • 0： 不使能 • 1： 使能。配置为测量模式时，在 OSC 达到 CNT_THR 时停下并清零
0	R/W	0x0	OSC_COUNTER_EN OSC 计数使能 • 0： 不使能 • 1： 使能。配置为测量模式时，在 OSC 达到 CNT_THR 时停下并清零

5.1.5.10. 0x0D4 OSC_CNT

默认值：0x00000000			OSC 计数器（OSC Counter ）
位域	类型	默认值	描述
31:0	RO	0x0	OSC_COUNTER_VALUE OSC 计数值

5.1.5.11. 0x0D8 XTAL_CNT

默认值：0x00000000			XTAL 计数器（XTAL Counter ）
位域	类型	默认值	描述
31:0	RO	0x0	XTAL_COUNTER_VALUE XTAL 计数值

5.1.5.12. 0x0DC OSC_CNT_THR

默认值：0x00005DC0			计数器门限（ Counter Threshold）
位域	类型	默认值	描述
31:0	W/R	0x5DC0	COUNTER_THRESHOLD 计数值门限。计数器在测量模式时，当 OSC 计数值等于门限值则停止计数。

5.1.5.13. 0x0E0 CLK_OUT0

默认值：0x00000000			CLK0 输出 (CLK0 Output)
位域	类型	默认值	描述
31:18	–	–	–
17	R/W	0x0	PRE_CLK_EN 前级时钟使能 • 0： 关闭 • 1： 打开
16	R/W	0x0	CLK_OUT_EN 时钟输出使能 • 0： 时钟输出关闭 • 1： 时钟输出打开 时钟输出频率 CLK_OUT = CLK_SRC / (DIV_N + 1)。 除频系数为 1~256，时钟输出为 50% 占空比。
15:14	–	–	–
13:12	R/W	0x0	CLK_SRC_SEL 时钟源选择 • 0：PLL_INT • 1：PLL_FRA • others：预留
11:8	–	–	–
7:0	R/W	0x0	CLK_DIV 时钟输出除频因子

5.1.5.14. 0x0E4 CLK_OUT1

默认值：0x00000000			CLK1 输出 (CLK1 Output)
位域	类型	默认值	描述
31:18	–	–	–
17	R/W	0x0	PRE_CLK_EN 前级时钟使能

默认值：0x00000000			CLK1 输出 (CLK1 Output)
位域	类型	默认值	描述
			• 0： 关闭 • 1： 打开
16	R/W	0x0	CLK_OUT_EN 时钟输出使能 • 0： 时钟输出关闭 • 1： 时钟输出打开 **时钟输出频率 `CLK_OUT = CLK_SRC / (DIV_N + 1)`。** **除频系数为 `1~256`，时钟输出为 `50%` 占空比。**
15:14	–	–	–
13:12	R/W	0x0	CLK_SRC_SEL 时钟源选择 • 0：PLL_INT • 1：PLL_FRA • others：预留
11:8	–	–	–
7:0	R/W	0x0	CLK_DIV 时钟输出除频因子

5.1.5.15. 0x180 CLK_CPU0

注：

对该寄存器写操作需要先配置 REG_WR_CFG 进行写授权申请。

默认值：0x00000000			CPU0 时钟 (CPU0 Clock)
位域	类型	默认值	描述
31:14	–	–	–
13	R/W	0x0	CLK_SRC_SEL CPU 时钟源选择 • 0：PLL_INT0/(DIV+1) • 1：PLL_FRA/(DIV+1)

默认值：0x00000000			CPU0 时钟 (CPU0 Clock)
位域	类型	默认值	描述
			50% 占空比，支持动态调频。
12	R/W	0x0	CLK_SRC_SEL_PLL 选择 PLL 时钟作为时钟源 • 0: CLK_24M • 1: CLK_SRC/(DIV+1) **50% 占空比，支持动态调频。**
11:4	–	–	–
3:0	R/W	0x0	CPU_CLK_DIV 输入时钟除频系数 0

5.1.5.16. 0x184 RST_ENTRY_CPU0

默认值：0x30000000			CPU0 复位入口（CPU0 Reset Entry）
位域	类型	默认值	描述
31:0	R/W	0x30000000	RST_ENTRY 复位入口地址

5.1.5.17. 0x18C DBG_CPU0

默认值：0x00000000			CPU0 调试（CPU0 Debug）
位域	类型	默认值	描述
31:29	–	–	–
28	RO	0x0	DFS_DONE 动态调频完成 **注：** CPU_DFS_BYPASS AHB_DFS_BYPASS AXI_DFS_BYPASS 任一位设为 0 时，启动 CPU_DFS 功能，如果发生 CPU 频率变更，完成频率变更后此位置 1。当下一次再有变更时，此位自动清零，直到完成变更后重新置 1
27	–	–	–

默认值：0x00000000			CPU0 调试（CPU0 Debug）
位域	类型	默认值	描述
26	R/W	0x1	CPU_CLK_SEL_DFS_BYPASS 此位设 0 打开 AXI _DFS 功能，变更 BUS _CLK_SEL 会有一个握手机制
25	R/W	0x1	CPU_CLK_SRC_DFS_BYPASS 此位设 0 打开 AXI _DFS 功能，变更 CLK _SRC_SEL 会有一个握手机制
24	R/W	0x1	CPU_CLK_DIV_DFS_BYPASS 此位设 0 打开 AXI _DFS 功能，变更 BUS _CLK_DIV 会有一个握手机制
23:0	–	–	–

5.1.5.18. 0x190 CLK_CPU1

注：

对该寄存器写操作需要先配置 REG_WR_CFG 进行写授权申请。

默认值：0x00000000			CPU1 时钟 (CPU1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	CPU_RSTN CPU 复位控制 • 0: 复位有效 • 1: 复位放开
23:17	–	–	–
16	R/W	0x0	CPU_CLK_EN CPU 时钟使能 • 0: 关闭 • 1: 打开
15:0	–	–	–

5.1.5.19. 0x194 RST_ENTRY_CPU1

默认值：0x30060000			CPU1 复位入口（CPU1 Reset Entry）
位域	类型	默认值	描述
31:0	R/W	0x30060000	RST_ENTRY 复位入口地址

5.1.5.20. 0x1C0 CLK_PRE_DIV

默认值：0x00020004			预除频 (CLK_PRE_DIV)
位域	类型	默认值	描述
31:18	–	–	–
17	R/W	0x1	CLK_EN_PRE 前级时钟使能 • 0：时钟关闭 • 1：时钟打开
16:4	–	–	–
3:0	R/W	0x4	MUX_DIMV MUX 除频系数

5.1.5.21. 0x20C CLK_WDOG

注：

对该寄存器写操作需要先配置 REG_WR_CFG 进行写授权申请。

默认值：0x00000000			WDOG 时钟 (WDOG Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位无效

默认值：0x00000000			WDOG 时钟 (WDOG Clock)
位域	类型	默认值	描述
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x0	MOD_CLK_SRC_SEL 模块时钟源选择 • 0：XTAL24M 750 分频 • 1：LPOSC24M_32K
11:0	–	–	–

5.1.5.22. 0x244 CLK_PWM0

默认值：0x00000000			XPWM0 时钟 (XPWM0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 注： **此信号同时复位模块和寄存器总线**

默认值：0x00000000			XPWM0 时钟 (XPWM0 Clock)
位域	类型	默认值	描述
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.23. 0x248 CLK_PWM1

默认值：0x00000000			XPWM1 时钟 (XPWM1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 注： **此信号同时复位模块和寄存器总线**
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.24. 0x24C CLK_PWM2

默认值：0x00000000			XPWM2 时钟 (XPWM2 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 注： **此信号同时复位模块和寄存器总线**
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.25. 0x250 CLK_PWM3

默认值：0x00000000			XPWM3 时钟 (XPWM3 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 注： **此信号同时复位模块和寄存器总线**
23:22	–	–	–

默认值：0x00000000			XPWM3 时钟 (XPWM3 Clock)
位域	类型	默认值	描述
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.26. 0x254 CLK_PWM4

默认值：0x00000000			XPWM4 时钟 (XPWM4 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 注： **此信号同时复位模块和寄存器总线**
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.27. 0x258 CLK_PWM5

默认值：0x00000000			XPWM5 时钟 (XPWM5 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0: 复位有效 • 1: 复位放开 **注：** **此信号同时复位模块和寄存器总线**
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.28. 0x400 CLK_DMA0

默认值：0x00000000			DMA0 时钟 (DMA0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23	–	–	–

默认值：0x00000000			DMA0 时钟 (DMA0 Clock)
位域	类型	默认值	描述
22	R/W	0x0	AXI_CLK_EN AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.29. 0x404 CLK_DMA1

默认值：0x00000000			DMA1 时钟 (DMA1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23	–	–	–
22	R/W	0x0	AXI_CLK_EN AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能

默认值：0x00000000			DMA1 时钟 (DMA1 Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.30. 0x414 CLK_DCE

默认值：0x00000000			DCE 时钟 (DCE Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.31. 0x418 CLK_CE

默认值：0x00001000			CE 时钟 (CE Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN

默认值：0x00001000			CE 时钟 (CE Clock)
位域	类型	默认值	描述
			模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23	–	–	–
22	R/W	0x0	AXI_CLK_EN AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 时钟源选择 • 0: PLL_INT • 1: PLL_FRA

默认值：0x00001000			CE 时钟 (CE Clock)
位域	类型	默认值	描述
15:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.32. 0x438 CLK_USB_FS_DR

默认值：0x00001004			USB FS DR 时钟 (CLK_USB_FS_DR)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块前级时钟使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能

默认值：0x00001004			USB FS DR 时钟 (CLK_USB_FS_DR)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x4	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.33. 0x43C CLK_USB_FS_DR_PHY

默认值：0x00000000			USB FS DR PHY 时钟 (CLK_USB_FS_DR_PHY)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:0	–	–	–

5.1.5.34. 0x440 CLK_EMAC

默认值：0x00000000			EMAC 时钟 (EMAC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:0	–	–	–

5.1.5.35. 0x448 CLK_CANFD0

默认值：0x00021000			CANFD0 时钟 (CANFD Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制

默认值：0x00021000			CANFD0 时钟 (CANFD Clock)
位域	类型	默认值	描述
			• 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x1	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.36. 0x44C CLK_CANFD1

默认值：0x00021000			CANFD1 时钟 (CANFD Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x1	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA

默认值：0x00021000			CANFD1 时钟 (CANFD Clock)
位域	类型	默认值	描述
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.37. 0x450 CLK_CAN

默认值：0x00000000			CAN 时钟 (CAN Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.38. 0x45C CLK_XSPI

注：

对该寄存器写操作需要先配置 REG_WR_CFG 进行写授权申请。

默认值：0x00001000			XSPI 时钟 (XSPI Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23	–	–	–
22	R/W	0x0	AXI_CLK_EN AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择

默认值：0x00001000			XSPI 时钟 (XSPI Clock)
位域	类型	默认值	描述
			• 0: PLL_INT • 1: PLL_FRA
11:4	–	–	–
3:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.39. 0x460 CLK_QSPI0

默认值：0x00001000			QSPI0 时钟 (QSPI0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23	–	–	–
22	R/W	0x0	AXI_CLK_EN AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN

默认值：0x00001000			QSPI0 时钟 (QSPI0 Clock)
位域	类型	默认值	描述
			模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.40. 0x464 CLK_QSPI1

默认值：0x00001000			QSPI1 时钟 (QSPI1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。

默认值：0x00001000			QSPI1 时钟 (QSPI1 Clock)
位域	类型	默认值	描述
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.41. 0x468 CLK_QSPI2

默认值：0x00001000			QSPI2 时钟 (QSPI2 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择

默认值：0x00001000			QSPI2 时钟 (QSPI2 Clock)
位域	类型	默认值	描述
			• 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.42. 0x46C CLK_QSPI3

默认值：0x00001000			QSPI3 时钟 (QSPI3 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–

默认值：0x00001000			QSPI3 时钟 (QSPI3 Clock)
位域	类型	默认值	描述
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.43. 0x470 CLK_QSPI4

默认值：0x00001000			QSPI4 时钟 (QSPI4 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能

默认值：0x00001000			QSPI4 时钟 (QSPI4 Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.44. 0x474 CLK_QSPI5

默认值：0x00001000			QSPI5 时钟 (QSPI5 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制

默认值：0x00001000			QSPI5 时钟 (QSPI5 Clock)
位域	类型	默认值	描述
			• 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
17	–	–	–
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:3	–	–	–
2:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.45. 0x490 CLK_CORDIC

默认值：0x00000000			CORDIC 时钟 (CORDIC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.46. 0x498 CLK_RDC

默认值：0x00000000			RDC 时钟 (RDC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–

默认值：0x00000000			RDC 时钟 (RDC Clock)
位域	类型	默认值	描述
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.47. 0x49C CLK_PFM

默认值：0x00000000			PFM 时钟 (PFM Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.48. 0x4A0 CLK_PBUS

默认值：0x00000000			PBUS 时钟 (PBUS Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.49. 0x4A4 CLK_FFT

默认值：0x00000000			FFT 时钟 (FFT Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
22	R/W	0x0	AXI_CLK_EN

默认值：0x00000000			FFT 时钟 (FFT Clock)
位域	类型	默认值	描述
			AXI 时钟使能 • 0：关闭 • 1：打开
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.50. 0x800 CLK_SYSCFG

默认值：0x00100000			SYSCFG 时钟 (SYSCFG Clock)
位域	类型	默认值	描述
31:21	–	–	–
20	R/W	0x1	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.51. 0x810 CLK_SPI_ENC

默认值：0x00000000			SPI_ENC 时钟 (SPI_ENC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开

默认值：0x00000000			SPI_ENC 时钟 (SPI_ENC Clock)
位域	类型	默认值	描述
			注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.52. 0x814 CLK_PWMCS

默认值：0x00000000			PWMCS 时钟 (PWMCS Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.53. 0x818 CLK_ADC

默认值：0x00000000			ADC 时钟 (ADC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.54. 0x820 CLK_I2S0

默认值：0x00001031			I2S0 时钟 (I2S0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–

默认值：0x00001031			I2S0 时钟 (I2S0 Clock)
位域	类型	默认值	描述
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:6	–	–	–
5:0	R/W	0x31	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.55. 0x824 CLK_I2S1

默认值：0x00001031			I2S1 时钟 (I2S1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制

默认值：0x00001031			I2S1 时钟 (I2S1 Clock)
位域	类型	默认值	描述
			• 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
11:6	–	–	–
5:0	R/W	0x31	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.56. 0x83C CLK_GPIO

默认值：0x01200000			GPIO 时钟 (GPIO Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x1	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x1	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.57. 0x840 CLK_UART0

默认值：0x00001000			UART0 时钟 (UART0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–

默认值：0x00001000			UART0 时钟 (UART0 Clock)
位域	类型	默认值	描述
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.58. 0x844 CLK_UART1

默认值：0x00001000			UART1 时钟 (UART1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制

默认值：0x00001000			UART1 时钟 (UART1 Clock)
位域	类型	默认值	描述
			• 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.59. 0x848 CLK_UART2

默认值：0x00001000			UART2 时钟 (UART2 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA

默认值：0x00001000			UART2 时钟 (UART2 Clock)
位域	类型	默认值	描述
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.60. 0x84C CLK_UART3

默认值：0x00001000			UART3 时钟 (UART3 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能

默认值：0x00001000			UART3 时钟 (UART3 Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.61. 0x850 CLK_UART4

默认值：0x00001000			UART4 时钟 (UART4 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–

默认值：0x00001000			UART4 时钟 (UART4 Clock)
位域	类型	默认值	描述
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.62. 0x854 CLK_UART5

默认值：0x00001000			UART5 时钟 (UART5 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。

默认值：0x00001000			UART5 时钟 (UART5 Clock)
位域	类型	默认值	描述
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x1	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x0	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.63. 0x858 CLK_UART6

默认值：0x00001000			UART0 时钟 (UART0 Clock)
位域	类型	默认值	描述
31:25	–	–	–

默认值：0x00001000			UART0 时钟 (UART0 Clock)
位域	类型	默认值	描述
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x1	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x0	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.64. 0x85C CLK_UART7

默认值：0x00001000			UART7 时钟 (UART7 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x1	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x0	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA

默认值：0x00001000			UART7 时钟 (UART7 Clock)
位域	类型	默认值	描述
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.65. 0x860 CLK_UART8

默认值：0x00001000			UART8 时钟 (UART8 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:18	–	–	–
17	R/W	0x0	MOD_CLK_PRE_EN 模块时钟预使能 • 0：关闭 • 1：打开
16	R/W	0x0	MOD_CLK_EN 模块时钟使能

默认值：0x00001000			UART8 时钟 (UART8 Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
15:13	–	–	–
12	R/W	0x1	MOD_CLK_SRC_SEL 模块时钟源选择 • 0: PLL_INT • 1: PLL_FRA
15:5	–	–	–
4:0	R/W	0x0	MOD_CLK_DIV 模块时钟除频系数，除频系数为 DIV + 1，50% 占空比。

5.1.5.66. 0x87C CLK_SDFM

默认值：0x00000000			SDFM 时钟 (SDFM Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.67. 0x904 CLK_SID

默认值：0x00110000			SID 时钟 (SID Clock)
位域	类型	默认值	描述
31:21	–	–	–
20	R/W	0x1	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:17	–	–	–
16	R/W	0x1	MOD_CLK_EN 模块时钟使能 • 0：关闭 • 1：打开
15:0	–	–	–

5.1.5.68. 0x90C CLK_GTC

默认值：0x01100000			GTC 时钟 (GTC Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x1	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x1	APB_CLK_EN APB 时钟使能

默认值：0x01100000			GTC 时钟 (GTC Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.69. 0x960 CLK_I2C0

默认值：0x00000000			I2C0 时钟 (I2C0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.70. 0x964 CLK_I2C1

默认值：0x00000000			I2C1 时钟 (I2C1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN

默认值：0x00000000			I2C1 时钟 (I2C1 Clock)
位域	类型	默认值	描述
			模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.71. 0x968 CLK_I2C2

默认值：0x00000000			I2C2 时钟 (I2C2 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能

默认值：0x00000000			I2C2 时钟 (I2C2 Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.72. 0x96C CLK_I2C3

默认值：0x00000000			I2C3 时钟 (I2C3 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–

5.1.5.73. 0x9AC CLK_THS

默认值：0x00000000			THS 时钟 (THS Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN

默认值：0x00000000			THS 时钟 (THS Clock)
位域	类型	默认值	描述
			模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.74. 0x9B0 CLK_VTS

默认值：0x00000000			VTS 时钟 (VTS Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能

默认值：0x00000000			VTS 时钟 (VTS Clock)
位域	类型	默认值	描述
			• 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.75. 0x9E4 CLK_CPM

默认值：0x00000000			CPM 时钟 (CPM Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:21	–	–	–
20	R/W	0x0	APB_CLK_EN APB 时钟使能 • 0：关闭 • 1：打开
19:0	–	–	–
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.76. 0x9F0 CLK_PGA

默认值：0x00000000			PGA 时钟 (PGA Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.77. 0xA00 CLK_GPT0

默认值：0x00000000			GPT0 时钟 (GPT0 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–

默认值：0x00000000			GPT0 时钟 (GPT0 Clock)
位域	类型	默认值	描述
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.78. 0xA04 CLK_GPT1

默认值：0x00000000			GPT1 时钟 (GPT1 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 **注：** 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.79. 0xA08 CLK_GPT2

默认值：0x00000000			GPT2 时钟 (GPT2 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.80. 0xA0C CLK_GPT3

默认值：0x00000000			GPT3 时钟 (GPT3 Clock)
位域	类型	默认值	描述
31:25	–	–	–
24	R/W	0x0	MOD_RSTN 模块复位控制 • 0：复位有效 • 1：复位放开 注： 此信号同时复位模块和寄存器总线。
23:22	–	–	–

默认值：0x00000000			GPT3 时钟 (GPT3 Clock)
位域	类型	默认值	描述
21	R/W	0x0	AHB_CLK_EN AHB 时钟使能 • 0：关闭 • 1：打开
20:0	–	–	–

5.1.5.81. 0xFE8 REG_WR_CFG

默认值：0x00000000			寄存器写配置 (Register Write Configuration)
位域	类型	默认值	描述
31:20	WO	0x0	AUTH_KEY 授权秘钥 • 0xA1C：申请写授权 • 0xD15：申请写失效 • Others：无效 需要写授权寄存器： • 0x000 PLL_INT_GEN • 0x020 PLL_FRA_GEN • 0x040 PLL_INT_CFG • 0x060 PLL_FRA_CFG • 0x080 PLL_FRA_SDM • 0x0A0 PLL_COM • 0x0A4 PLL_IN • 0x0A8 PLL_OUT • 0x180 CLK_CPU0 • 0x190 CLK_CPU1 • 0x20C CLK_WDOG 写授权操作流程： • 同时对 REG_ADDR_OFF 写入偏移地址和对 AUTH _KEY 写入 0xA1C • STS_WR_AUTH 置位表示申请成功 • 对 CMU 任何写操作都会将 STS _WR_AUTH 清除

默认值：0x00000000			寄存器写配置 (Register Write Configuration)
位域	类型	默认值	描述
			支持写失效的寄存器： • 0x20C CLK_WDOG 写失效操作流程： • 同时对 REG_ADDR_OFF 写入偏移地址和对 AUTH _KEY 写入 0xD15 • STS_WR_DIS 对应位置位表示失效生效，即无法对对应的寄存器进行写操作，注意复位是可以清除写失效
19:17	–	–	–
16	RO	0x0	STS_WR_AUTH 写授权状态 • 0：REG_ADDR_OFF 获得授权 • 1：REG_ADDR_OFF 获得授权
15:0	R/W	0x0	REG_ADDR_OFF 寄存器地址偏移

5.1.5.82. 0xFEC REG_WR_DIS_STS

默认值：0x00000000			寄存器写失效状态 (Register Write Disable Status)
位域	类型	默认值	描述
31:0	RO	0x0	STS_WR_DIS 写失效状态 • 0：寄存器可写 • 1：寄存器写失效 每位对应一个寄存器： • BIT0：CLK_WDOG • BIT31:1 预留

5.1.5.83. 0xFFC CMU_MOD_VER

默认值：0x00000103			CMU 模块版本 (CMU Module Version)
位域	类型	默认值	描述
31:0	RO	0x00000103	VERSION

默认值：0x00000103			CMU 模块版本 (CMU Module Version)
位域	类型	默认值	描述
			版本 V3.0

5.2. System Configuration (SYSCFG)

System Config (SYSCFG) 模块为系统域以及模块顶层提供丰富的基础配置选项，包括但不限于配置 LDO 的工作模式或电压范围、SRAM 的大小或速度、PRS 的频率或周期、以及 USB 工作模式选择等。

5.2.1. 特性说明

- 内置 LDO25，用于系统模拟及 ADC/ eFuse 供电
- 内置 LDO18，用于 PSRAM 颗粒供电
- 内置 LDO1x，用于 CORE 供电或外挂设备供电
- 提供 PRS 用于检测当前芯片的速度
- 提供参数配置到 SRAM，包含 CPU 等模块
- 提供模块顶层控制，包括 USB/EMAC 等
- 调试信息输出配置

5.2.2. 原理框图

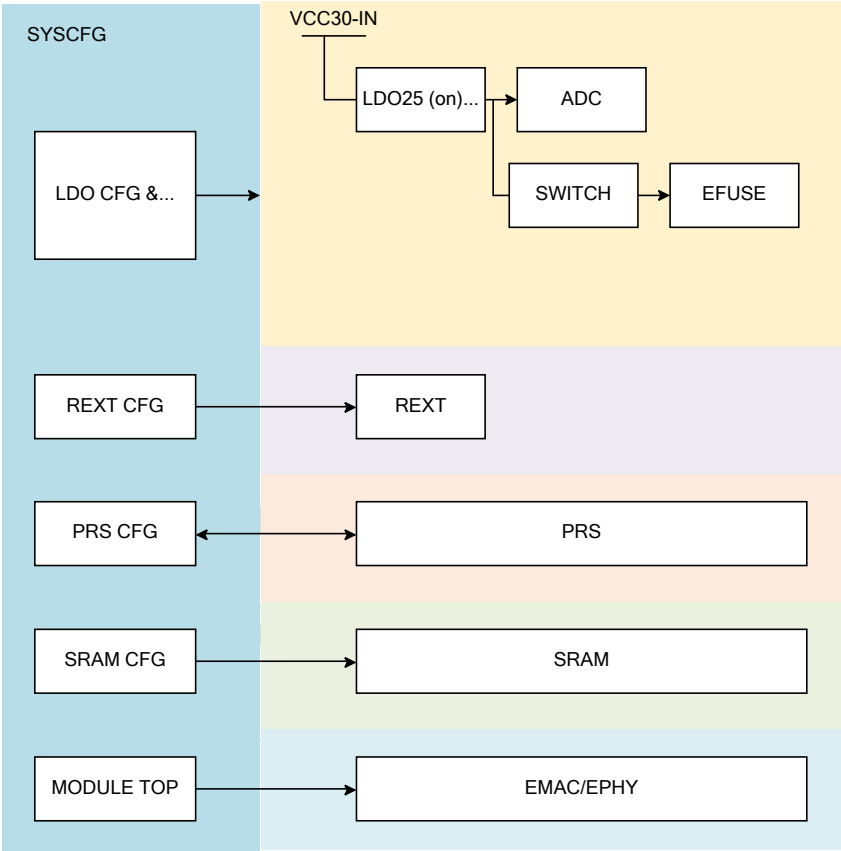


图 5–5 SYSCFG 原理框图

5.2.3. 寄存器列表

表 5–3 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x020	LDO25_CFG	LDO25 配置	0x020 LDO25_CFG
0x024	LDO18_CFG	LDO18 配置	0x024 LDO18_CFG
0x028	LDO1x_CFG	LDO1x 配置	0x028 LDO1x_CFG
0x0C0	PSEN_CFG	PSEN 配置	0x0C0 PSEN_CFG

表 5–3 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0C4	PSEN_CNT_VAL	PSEN 计数值	0x0C4 PSEN_CNT_VAL
0x100	SYS_SRAM_PAR	SYS SRAM 参数	0x100 SYS_SRAM_PAR
0x104	HIGH_SPEED_SRAM_PAR	高速 SRAM 参数	0x104 HIGH_SPEED_SRAM_PAR
0x140	SRAM_CLK_CFG	SRAM 时钟配置	0x140 SRAM_CLK_CFG
0x160	SRAM_MAP_CFG	SRAM 映射配置	0x160 SRAM_MAP_CFG
0x1F0	FLASH_CFG	FLASH 配置	0x1F0 FLASH_CFG
0x1F4	ENCODER_CFG	ENCODER 配置	#unique_63_Connect_42_section_vg2_qhj_syb
0x1F8	GPIO_CFG	GPIO 配置寄存器	0x1F8 GPIO_CFG
0x410	EMAC_CFG	EMAC配置	0x410 EMAC_CFG
0x414	EMAC_CFG1	EMAC 配置 1	0x414 EMAC_CFG1
0x500	JTAG_KSR0	JTAG 密钥保管寄存器 0	0x500 JTAG_KSR0
0x504	JTAG_KSR1	JTAG 密钥保管寄存器 1	0x504 JTAG_KSR1
0x508	JTAG_KSR2	JTAG 密钥保管寄存器 2	0x508 JTAG_KSR2
0x50C	JTAG_KSR3	JTAG 密钥保管寄存器 3	0x50C JTAG_KSR3
0x510	JTAG_KCR	JTAG 密钥控制寄存器	0x510 JTAG_KCR
0x520	JTAG_KIR0	JTAG 密钥输入寄存器 0	0x520 JTAG_KIR0
0x524	JTAG_KIR1	JTAG 密钥输入寄存器 1	0x524 JTAG_KIR1
0x528	JTAG_KIR2	JTAG 密钥输入寄存器 2	0x528 JTAG_KIR2
0x52C	JTAG_KIR3	JTAG 密钥输入寄存器 3	0x52C JTAG_KIR3
0xFFC	SYSCFG_VER	SYSCFG 版本	0xFFC SYSCFG_VER

5.2.4. 寄存器描述

5.2.4.1. 0x020 LDO25_CFG

默认值：0x00000011			LDO25 配置 (LDO25 Configuration)
位域	类型	默认值	描述
31:28	–	–	–
17:16	–	–	–
15:8	R/W	0x0	BG_CTRL BG 电压控制
7:5	–	–	–
4	R/W	0x1	LDO25_EN LDO25 使能 • 0: LDO 关闭 • 1: LDO 打开
3	–	–	–
2:0	R/W	0x1	LDO25_VAL LDO25 电压设置，为了避免影响 EFUSE 的烧写，此处配置为 2.50V。 • 0: 2.40 V • 1: 2.50 V • 2: 2.60 V • 3: 2.70 V • 4: 2.80 V • 5: 2.90 V • 6: 3.00 V • 7: 3.10 V

5.2.4.2. 0x024 LDO18_CFG

默认值：0x00030007			LDO18 配置 (dLDO18 Configuration)
位域	类型	默认值	描述
31	R/W	0x0	ATB1_ANA_EN

默认值：0x00030007			LDO18 配置 (dLDO18 Configuration)
位域	类型	默认值	描述
			模拟测试使能 • 0: 关闭 • 1: 使能
30:28	R/W	0x0	ATB1_ANA_SEL 模拟测试模块选择 • ATB1_INx 输出，其中 x 的范围为 0~7
27	R/W	0x0	ATB0_ANA_EN 模拟测试使能 • 0: 关闭 • 1: 使能
26:24	R/W	0x0	ATB0_ANA_SEL 模拟测试模块选择 • ATB0_INx 输出，其中 x 的范围为 0~7
23:19	–	–	–
18:16	R/W	0x3	LDO18_OSC24M_VAL 振荡器频率选择输出 • 0： 18 MHz • 1： 20 MHz • 2：22 MHz • 3：24 MHz • 4：26 MHz • 5：28 MHz • 6：30 MHz • 7：32 MHz
15:8	R/W	0x0	LDO18_RESERVED
7:6	–	–	–
5	R/W	0x0	LDO18_PD_FAST LDO18 快速掉电

默认值：0x00030007			LDO18 配置 (dLDO18 Configuration)
位域	类型	默认值	描述
			• 0: 正常掉电 • 1: 快速掉电
4	R/W	0x0	LDO18_EN LDO18 使能 • 0: LDO 关闭 • 1: LDO 打开
3:0	R/W	0x7	LDO18_VAL LDO18 电压设置 • 0: 1.71 V • 1: 1.74 V • 2: 1.77 V • 3: 1.80 V • 4: 1.83 V • 5: 1.86 V • 6: 1.89 V • 7: 1.92 V • 8: 1.95 V • 9: 1.98 V • 10: 1.01 V • 11: 2.04 V • 12: 2.07 V • 13: 2.10 V • 14: 2.13 V • 15: 2.16 V

5.2.4.3. 0x028 LDO1x_CFG

默认值：0x00000028			LDO1 配置 (LDO1 Configuration)
位域	类型	默认值	描述
31:13	–	–	–
12	R/W	0x0	BOR_EN

默认值：0x00000028			LDO1 配置 (LDO1 Configuration)
位域	类型	默认值	描述
			BOR 模块使能
11:8	R/W	0x0	BOR_HYSVAL_SEL BOR 阈值选择 • 0: 2.7 V • 1: 2.8 V • 2: 2.9 V • 3: 3.0 V • 8: 2.16 V • 9: 2.24 V • 10: 2.32 V • 11: 2.4 V • others: 无效
7	–	–	–
6	R/W	0x0	LDO1x_SOFT_EN LDO1x 软件使用 • 0: LDO 关闭 • 1: LDO 打开
5	R/W	0x1	LDO1_PD_FAST LDO1 快速掉电 • 0: 正常掉电 • 1: 快速掉电
4	R/W	0x0	LDO1x_EN LDO1x 使能 • 0: LDO 关闭 • 1: LDO 打开
3:0	R/W	0x8	LDO1x_VAL LDO1x 电压设置 • 0: 1.0 V • 1: 1.025 V • 2: 1.05 V

默认值：0x00000028			LDO1 配置 (LDO1 Configuration)
位域	类型	默认值	描述
			• 3: 1.075 V • 4: 1.1 V • 5: 1.125 V • 6: 1.15 V • 7: 1.175 V • 8: 1.2 V • 9: 1.225 V • 10: 1.25 V • 11: 1.275 V • 12: 1.3 V • 13: 1.325 V • 14: 1.35 V • 15: 1.375 V

5.2.4.4. 0x0C0 PSEN_CFG

默认值：0x00000000			PSEN 配置 (Process Sensor Configuration)
位域	类型	默认值	描述
31:16	R/W	0x0	CNT_TIME 计数时间 单位为 APB 总线时钟周期
15:5	–	–	–
4	R/W	0x0	PSEN_SEL PSEN 选择 • 0: 8T • 1: 12T
3:1	R/W	0x0	RO_SEL PSEN 选择 • 0: 无效 • 1: RVT_40 CELL • 2: LVT_40 CELL

默认值：0x00000000			PSEN 配置 (Process Sensor Configuration)
位域	类型	默认值	描述
			• 4: 无效 • 5: RVT_50 CELL • 6: LVT_50 CELL • others:无效
0	R/W	0x0	PSEN_START PSEN 开启 • 0: 关闭 • 1: 开启 **注：** PSEN 测试结束后，该位自动清零表示测试结束。

5.2.4.5. 0x0C4 PSEN_CNT_VAL

默认值：0x00000000			PSEN 计数值 (Process Sensor Counter Value)
位域	类型	默认值	描述
31:16	–	–	–
15:0	RO	0x0	CNT_VAL 计数值

5.2.4.6. 0x100 SYS_SRAM_PAR

默认值：0x00022202			SYS SRAM 参数 (System SRAM Parameter)
位域	类型	默认值	描述
31:0	R/W	0x00022202	SRAM_PAR SRAM 参数

5.2.4.7. 0x104 HIGH_SPEED_SRAM_PAR

默认值：0x55555555			高速 SRAM 参数 (High speed SRAM Parameter)
位域	类型	默认值	描述
31:0	R/W	0x55555555	SRAM_PAR SRAM 参数

5.2.4.8. 0x140 SRAM_CLK_CFG

默认值：0x00000000			SRAM 时钟配置 (SRAM Clock Configuration)
位域	类型	默认值	描述
31:18	–	–	–
17:0	R/W	0x0	SRAM_CLK_UNGATE SRAM 时钟不自动关闭 • 0: 自动关闭 • 1: 不自动关闭 每位对应一个模块，定义如下– • BIT24: FFT • BIT16: SDFM • BIT15: CANFD • BIT12: SPI • BIT11: USB • BIT10: UART • BIT9: SD • BIT8: CE • BIT3: GMAC • BIT2: DMA • BIT0: SYS

5.2.4.9. 0x160 SRAM_MAP_CFG

默认值：0x00000004			SRAM 映射配置 (SRAM Mapping Configuration)
位域	类型	默认值	描述
31:20	WO	0xa1c	AUTH_KEY 写授权 写入值匹配 0xa1c 才允许对整个寄存器写入。
19:7	–	–	–
6:5	R/W	0x0	CPU_TCM_SRAM_CFG CPU TCM SRAM 配置 • 0: I/D TCM 128KB/64KB • 1: I/D TCM 128KB/128KB • 2: I/D TCM 256KB/64KB • 3: I/D TCM 256KB/128KB
4	R/W	0x0	TCM_ENABLE TCM 使能 • 0: 不使能 TCM • 1: 使能 TCM
3	R/W	0x0	FFT_SRAM_CFG FFT SRAM 配置 • 0: 无专用 SRAM • 1: 分配专用 SRAM
2:0	–	–	–

5.2.4.10. 0x1F0 FLASH_CFG

默认值：0x00000000			FLASH 配置 (FLASH Configuration)
位域	类型	默认值	描述
31:15	–	–	–
14:12	R/W	0x0	FLASH_IOMAP_012 FLASH_IO[0:2] 映射

默认值：0x00000000			FLASH 配置 (FLASH Configuration)
位域	类型	默认值	描述
			• 0: WP – SO – CS • 1: WP – CS – SO • 2: SO – WP – CS • 3: SO – CS – WP • 4: CS – WP – SO • 5: CS – SO – WP • 6~7: 预留 **注：** BROM 启动后需要从 EFUSE 获取 FLASH_IOMAP 信息配置到该位域。
11	–	–	–
10:8	R/W	0x0	FLASH_IOMAP_345 FLASH_IO[3:5] 映射 • 0: HOLD – SCLK – SI • 1: HOLD – SI – SCLK • 2: SCLK – HOLD – SI • 3: SCLK – SI – HOLD • 4: SI – HOLD – SCLK • 5: SI – SCLK – HOLD • 6~7: 预留 **注：** BROM 启动后需要从 eFuse 获取 FLASH_IOMAP 信息配置到该位域。
7:2	–	–	–
1:0	R/W	0x0	FLASH_SRCSEL Flash 访问接口选择 • 0: NONE，SiP Flash 接口关闭，无法访问 • 1: PIN，SiP Flash 接口通过 PIN 映射的方式进行访问，可被外部设备访问 • 2: SPI0，SiP Flash 通过 SPI0 控制器访问 • 3: SPI1，SiP Flash 通过 SPI1 控制器访问

5.2.4.11. 0x1F8 GPIO_CFG

默认值：0x00000000			GPIO 配置寄存器
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	PK_FUNCTION PK 组功能选择 • 0: 预留 • 1: PINMUX

5.2.4.12. 0x410 EMAC_CFG

默认值：0x00000000			EMAC配置 (EMAC Configuration)
位域	类型	默认值	描述
31	–	–	–
30	R/W	0x0	RMII_RXCLK_INV RMII RX 时钟反向 • 0: 时钟正向 • 1: 时钟反向
29	R/W	0x0	REFCLK_INV 参考时钟反向 • 0: 时钟正向 • 1: 时钟反向
28:24	R/W	0x0	REFCLK_DLY_CHAIN_SEL 参考时钟延迟链选择
23	R/W	0x0	RXCLK_INV 接收时钟反向 • 0: 时钟正向 • 1: 时钟反向
22:18	R/W	0x0	RXCLK_DLY_SEL 接收时钟延迟链选择
17	R/W	0x0	TXCLK_INV

默认值：0x00000000			EMAC配置 (EMAC Configuration)
位域	类型	默认值	描述
			发送时钟反向 • 0: 时钟正向 • 1: 时钟反向
16:12	R/W	0x0	TXCLK_DLY_CHAIN_SEL 发送时钟延迟链选择
11:8	R/W	0x0	SW_TXCLK_DIV2 软件发送时钟分频 2
7:4	R/W	0x0	SW_TXCLK_DIV1 软件发送时钟分频 1
3	–	–	–
2	R/W	0x0	SW_TXCLK_DIV_EN 软件发送时钟分频使能 • 0: 分频固定 • 1: 分频可配置
1	R/W	0x0	RMII_EXTCLK_SEL RMII 外部时钟选择 • 0: 使用 INTCLK • 1: 使用 EXTCLK
0	–	–	–

5.2.4.13. 0x414 EMAC_CFG1

默认值：0x00000000			EMAC配置 1 (EMAC Configuration 1)
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	MII_TX_SHIFT MII 发送相位调节，可选 0/1/2/3，每个相位相差 0.5T
21	–	–	–

默认值：0x00000000			EMAC配置 1 (EMAC Configuration 1)
位域	类型	默认值	描述
20	R/W	0x0	MII_TX_SHIFT_MANUAL MII TX 手动相位调节 • 0: 禁用 • 1: 使能
19:18	–	–	–
17	R/W	0x0	MII_TX_CLK_OUT_EN MII TX 时钟输出使能 • 0: 禁用 • 1: 使能
16	R/W	0x0	PHY_MII_SEL PHY MII 接口选择 • 0: RMII • 1: MII
15:6	–	–	–
5:4	R/W	0x0	PTP_REFCLK_SEL PTP 时钟源选择 • 0: GMAC 内部 50M • 1: 外部输入 • 2: AHB 时钟 • 3:无效
3	–	–	–
2	R/W	0x0	RXCLK_DLY_CHAIN_EN 接收时钟延迟链使能 • 0: 旁路 • 1: 使能
1	R/W	0x0	TXCLK_DLY_CHAIN_EN 发送时钟延迟链使能 • 0: 旁路 • 1: 使能

默认值：0x00000000			EMAC配置 1 (EMAC Configuration 1)
位域	类型	默认值	描述
0	R/W	0x0	REFCLK_DLY_CHAIN_EN 参考时钟延迟链使能 • 0: 旁路 • 1: 使能

5.2.4.14. 0x500 JTAG_KSR0

默认值：0x00000000			JTAG 密钥保管寄存器 0 (JTAG Key Storage Register 0)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[31:0] 此寄存器保存 128-bit JTAG 密钥的一部分。当与输入密钥 JTAG_KIR0 进行对照时，如果匹配成功，则解锁 JTAG 功能。此寄存器默认可读写，但读写属性也受寄存器 JTAG_KCR 与上锁状态的控制。JTAG_KSR0~3 同时不为 0x00000000，JTAG 就会自动上锁，JTAG_LS 被置 1，并且 JTAG_KSR0~3 被写保护。

5.2.4.15. 0x504 JTAG_KSR1

默认值：0x00000000			JTAG 密钥保管寄存器 1 (JTAG Key Storage Register 1)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[63:32] 此寄存器保存 128-bit JTAG 密钥的一部分。当与输入密钥 JTAG_KIR0 进行对照时，如果匹配成功，则解锁 JTAG 功能。此寄存器默认可读写，但读写属性也受寄存器 JTAG_KCR 与上锁状态的控制。JTAG_KSR0~3 同时不为 0x00000000，JTAG 就会自动上锁，JTAG_LS 被置 1，并且 JTAG_KSR0~3 被写保护。

5.2.4.16. 0x508 JTAG_KSR2

默认值：0x00000000			JTAG 密钥保管寄存器 2 (JTAG Key Storage Register 2)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[95:64] 此寄存器保存 128-bit JTAG 密钥的一部分。当与输入密钥 JTAG_KIR0 进行对照时，如果匹配成功，则解锁 JTAG 功能。此寄存器默认可读写，但读写属性也受寄存器 JTAG_KCR 与上锁状态的控制。JTAG_KSR0~3 同时不为 0x00000000，JTAG 就会自动上锁，JTAG_LS 被置 1，并且 JTAG_KSR0~3 被写保护。

5.2.4.17. 0x50C JTAG_KSR3

默认值：0x00000000			JTAG 密钥保管寄存器 3 (JTAG Key Storage Register 3)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[127:96] 此寄存器保存 128-bit JTAG 密钥的一部分。当与输入密钥 JTAG_KIR0 进行对照时，如果匹配成功，则解锁 JTAG 功能。此寄存器默认可读写，但读写属性也受寄存器 JTAG_KCR 与上锁状态的控制。JTAG_KSR0~3 同时不为 0x00000000，JTAG 就会自动上锁，JTAG_LS 被置 1，并且 JTAG_KSR0~3 被写保护。

5.2.4.18. 0x510 JTAG_KCR

默认值：0x00080000			JTAG 密钥控制寄存器 (JTAG Key Control Register)
位域	类型	默认值	描述
31:24	RO	0x00000000	JTAG WC JTAG_KIR Write Count, JTAG KIR 寄存器写次数计数 此计数器用于记录对 JTAG_KIR0~3 寄存器的总写操作次数。每次对 JTAG_KIR0~3 寄存器进行写入操作时，计数器都会增加相应的次数。例如，对 JTAG_KIR0~3 分别各写一次，计数器会计数四次，对同一 JTAG_KIR 寄存器连续写多次也会导致计数器计数增加多次。 当本字段的计数值达到 JKIR_WC 字段的设置值，会导致 JTAG_KIR0~3 寄存器无法再被写入。 当计数器达到 JKIR_WL 限值且 JTAG_KIR0~3 被禁止写入后，如果还有对该地址段的写操作，计数器会继续计数。 计数器在达到 0xFF 时停止计数，以避免计数溢出。但是，这不影响 JTAG_KIR0~3 寄存器的写属性。如果 JKIR_WL=0x00，则继续可写；如果 JKIR_WL 为其它值，则应该禁止写入。
23:16	R/W	0x08	JKIR_WL JTAG_KIR Write Limit, JTAG_KIR 寄存器写次数限制 设置此字段可限制对 JTAG_KIR0~3 寄存器的总写次数。当 JKIR_WC 写入次数达到一旦达到此限值时，会导致 JTAG_KIR0~3 寄存器无法再被写入。 此字段为 0，会使写次数限制功能被取消。
15:13	–	–	–
12	R/WAC	0x0	JKIR_WCR JTAG_KIR Write Count Reset，KIR 寄存器写次数计数复位 通过往此字段写 0x1，可以把 JIKR_WC 计数器复位清零。 完成对 JKIR_WC 计数器的复位清零后，此字段由硬件自动清零。

默认值：0x00080000			JTAG 密钥控制寄存器 (JTAG Key Control Register)
位域	类型	默认值	描述
11:9	–	–	–
8	R/W1S	0x0	JKIR_RP JTAG_KIR Read Protect, JTAG_KIR 寄存器读保护 置位此字段，会导致 JTAG_KIR0–3 寄存器不可读（读回为全 0）。 此字段一旦被置位，不能被清零。
7:5	–	–	–
4	RO	0x0	JTAG_LS JTAG Lock Status, JTAG 上锁状态 如果 JTAG_KSR0~3 四个寄存器值均不为 0x00000000，JTAG 就会自动上锁，本字段被置为 0x1，同时对 JTAG_KSR0~3 进行写保护，禁止 JTAG_KSR0~3 密钥被修改。当 JTAG_KIR0~3 与 JTAG_KSR0~3 匹配后，本字段会被硬件清零表示 JTAG 解锁状态。
3:1	–	–	–
0	R/W1S	0x0	JKSR_RP JTAG_KSR Read Protect, JTAG_KSR 寄存器读保护 置位此字段，会导致 JTAG_KSR0–3 寄存器不可读（读回为全 0）。 此字段一旦被置位，不能被清零。

5.2.4.19. 0x520 JTAG_KIR0

默认值：0x00000000			JTAG 密钥输入寄存器 0 (JTAG Key Input Register 0)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[31:0] 通过此寄存器输入 128–bit JTAG 密钥中一部分，用于与保管密钥 JTAG_KSR0 进行对照解锁 JTAG。 此寄存器默认可读/可写，但读写属性也受寄存器 JTAG_KCR 控制。

5.2.4.20. 0x524 JTAG_KIR1

默认值：0x00000000			JTAG 密钥输入寄存器 1 (JTAG Key Input Register 1)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[63:32]

默认值：0x00000000			JTAG 密钥输入寄存器 1 (JTAG Key Input Register 1)
位域	类型	默认值	描述
			通过此寄存器输入 128-bit JTAG 密钥中一部分，用于与密钥保管 JTAG_KSR0 进行对照解锁 JTAG。 此寄存器默认可读/可写，但读写属性也受寄存器 JTAG_KCR 控制。

5.2.4.21. 0x528 JTAG_KIR2

默认值：0x00000000			JTAG 密钥输入寄存器 2 (JTAG Key Input Register 2)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[95:64] 通过此寄存器输入 128-bit JTAG 密钥中一部分，用于与保管密钥 JTAG_KSR0 进行对照解锁 JTAG。 此寄存器默认可读/可写，但读写属性也受寄存器 JTAG_KCR 控制。

5.2.4.22. 0x52C JTAG_KIR3

默认值：0x00000000			JTAG 密钥输入寄存器 3 (JTAG Key Input Register 3)
位域	类型	默认值	描述
31:0	R/W	0x00000000	JTAG Key[127:96] 通过此寄存器输入 128-bit JTAG 密钥中一部分，用于与保管密钥 JTAG_KSR0 进行对照解锁 JTAG。 此寄存器默认可读/可写，但读写属性也受寄存器 JTAG_KCR 控制。

5.2.4.23. 0xFFC SYSCFG_VER

默认值：0x00000105			SYSCFG 版本 (SYSCFG Version)
位域	类型	默认值	描述
31:0	RO	0x00000105	VERSION 版本 V1.5

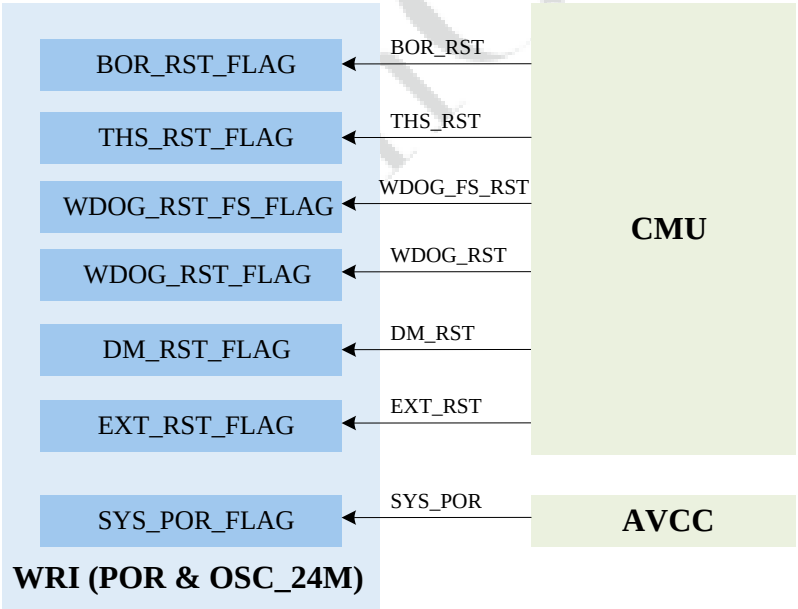
5.3. Warm Reset Information (WRI)

Warm Reset Info (WRI) 可以记录暖复位发生的状态信息， 提供引起死机重启的原因分析。暖复位包括有记录触发硬件复位的复位源等。

5.3.1. 特性说明

记录触发硬件复位的复位源。

5.3.2. 原理框图



5.3.3. 功能描述

暖复位，也称热复位 (Hot Reset)，相对于冷复位 (Cold Reset)，是指系统不做掉电直接复位。

WRI 模块可用于记录暖复位信息，在暖复位期间稳定工作，使用到的系统资源不受复位影响：

- WRI 模块复位源于 SYS_POR，不受其它复位源影响。
- 时钟来源于 CLK_24M，不受 CMU 时钟选择影响。

WRI 提供复位源记录功能，即每个复位源信号经过去抖后，使用其低电平置位对应标志位。对应标志位的值可进行写 1 清 0。每次启动后，建议检查复位源信息并进行清除，方便记录下次复位源信息。

5.3.4. 寄存器列表

表 5–4 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	RST_FLAG	复位标志	0x000 RST_FLAG
0x100	BOOT_INFO	启动信息	0x100 BOOT_INFO
0x104	SYS_BAK	系统备份	0x104 SYS_BAK
0xFFC	WRI_VER	WRI 版本	0xFFC WRI_VER

5.3.5. 寄存器描述

5.3.5.1. 0x000 RST_FLAG

默认值：0x00000001			复位标志 (Reset Flag)
位域	类型	默认值	描述
9	R/W1C	0x0	BOR_RST_FLAG 欠压复位标志 • 0: 没发生过复位 • 1: 发生过复位
8	R/W1C	0x0	THS_RST_FLAG 温度过高复位标志 • 0: 没发生过复位 • 1: 发生过复位
7	R/W1C	0x0	WDOG1_RST_FLAG 看门狗 1 复位标志 • 0: 没发生过复位 • 1: 发生过复位
6	R/W1C	0x0	WDOG0_RST_FLAG 看门狗 0 复位标志 • 0: 没发生过复位 • 1: 发生过复位

默认值：0x00000001			复位标志 (Reset Flag)
位域	类型	默认值	描述
5	R/W1C	0x0	WDOG1_SYS_FS_RST_FLAG 看门狗 1 强制系统复位标志 • 0: 没发生过复位 • 1: 发生过复位
4	R/W1C	0x0	WDOG0_SYS_FS_RST_FLAG 看门狗 0 强制系统复位标志 • 0: 没发生过复位 • 1: 发生过复位
3	R/W1C	0x0	CPU1_DM_RST_FLAG CPU1 调试复位标志 • 0: 没发生过复位 • 1: 发生过复位
2	R/W1C	0x0	CPU0_DM_RST_FLAG CPU0 调试复位标志 • 0: 没发生过复位 • 1: 发生过复位
1	R/W1C	0x0	EXT_RST_FLAG 外部输入复位标志 • 0: 没发生过复位 • 1: 发生过复位
0	R/W1C	0x1	SYS_POR_FLAG 系统上电复位标志 • 0: 没发生过复位 • 1: 发生过复位

5.3.5.2. 0x100 BOOT_INFO

默认值：0x00000000			启动信息 (Boot Info)
位域	类型	默认值	描述
31:8	R/W	0x0	保留位

默认值：0x00000000			启动信息 (Boot Info)
位域	类型	默认值	描述
7:4	R/W	0x0	REBOOT_REASON 启动/ 重启的原因记录寄存器
3:0	R/W	0x0	BOOT_DEV 启动介质记录寄存器，该信息由 BROM 设置。

5.3.5.3. 0x104 SYS_BAK

默认值：0x00000000			系统备份 (System Backup)
位域	类型	默认值	描述
31:0	R/W	0x0	SYS_BAK

5.3.5.4. 0xFFC WRI_VER

默认值：0x00000105			WRI 版本 (WRI Version)
位域	类型	默认值	描述
31:0	RO	0x00000105	VERSION 版本，V1.5

6. 计时器

6.1. Generic Timer Controller (GTC)

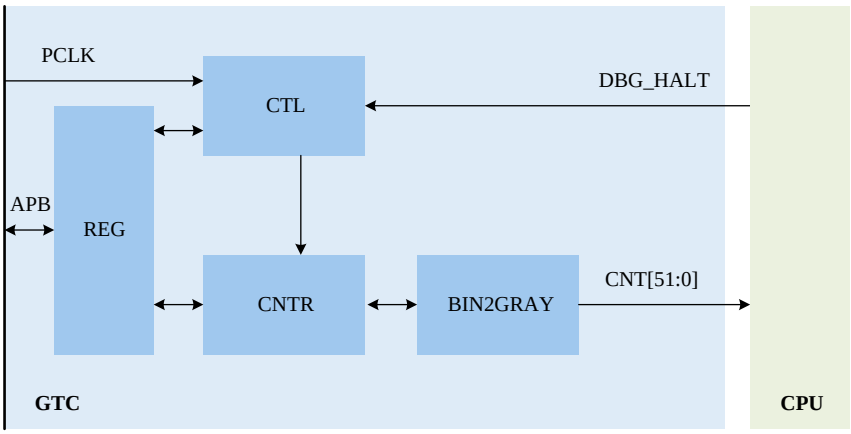
General Timer Counter (GTC) 通用计数器模块，为 CPU 提供基础时钟。GTC 模块输出计数值为 52 bits，计数的最高频率（基频）为 4 MHz，计数周期约为 35 年。

GTC 模块输出为 52 bits，但 CPU 输入为 64 bits，因此计数值高位寄存器的高 12 bits 固定为 0。

6.1.1. 特性说明

- 总线时钟 PCLK 的频率可选，但必须为 4 MHz 的整数倍频，例如 12 MHz/ 24 MHz/ 48 MHz/ 60 MHz 等
- 计数器最高计数频率（基频）为 4 MHz，计数精度为 0.25 MHz，即 250 ns
- 计数时钟频率误差不超过 10 s/24 h，约为 100 ppm
- 计数器位宽为 52 bits，计数周期约 35 年
- 计数频率可选，4 MHz/ 1 MHz/ 250 KHz
- 不同的计数频率，计数精度不变，例如 4 MHz 每周期增加 1，1 MHz 每周期增加 4，250 KHz 则每周期增加 16
- 52 bits 计数值以格雷码 (Gray Code) 编码方式输出
- 复位后模块计数从 0 开始，并且仅支持增量计数

6.1.2. 原理框图



6.1.3. 寄存器列表

表 6–1 GTC 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x0000	GTC_CNTCR	GTC 计数器控制	0x0000 GTC_CNTCR
0x0004	GTC_CNTSR	GTC 计数器控制	0x0004 GTC_CNTSR
0x0008	GTC_CNTVL	GTC 计数值低位	0x0008 GTC_CNTVL
0x000C	GTC_CNTVH	GTC 计数值高位	0x000C GTC_CNTVH
0x0020	GTC_CNTFID0	GTC 计数频率 ID0	0x0020 GTC_CNTFID0
0x0024	GTC_CNTFID1	GTC 计数频率 ID1	0x0024 GTC_CNTFID1
0x0028	GTC_CNTFID2	GTC 计数频率 ID2	0x0028 GTC_CNTFID2
0x002C	GTC_CNTFID3	GTC 计数频率 ID3	0x002C GTC_CNTFID3
0x00C0	GTC_CONFG	GTC 配置	0x00C0 GTC_CONFG
0x0FFC	GTC_VERSION	GTC 版本	0x0FFC GTC_VERSION

6.1.4. 寄存器描述

6.1.4.1. 0x0000 GTC_CNTCR

默认值：0x00000000			GTC 计数器控制 (GTC Counter Control)
位域	类型	默认值	描述
31:10	–	–	–
9:8	R/W	0x0	FCREQ 计数频率调整请求，本字段控制计数器计数频率选择： • 0：计数频率为基频 4 MHz • 1：计数频率为 1 MHz • 2：计数频率为 250 KHz • 3：未定义 写入新的值后，GTC 模块会切换计数频率，可通过查询状态寄存器 CNTSR 的 FCACK 确认是否切换完成。

默认值：0x00000000			GTC 计数器控制 (GTC Counter Control)
位域	类型	默认值	描述
1	R/W	0x0	CNT_ON_DBG 调试模式下继续计时 • 0：暂停计时 • 1：继续计时
0	R/W	0x0	EN 使能控制，仅控制计数器是否正常工作，不会复位计数器的其它配置。 • 0：计数器模块停止工作 • 1：计数器模块正常工作

6.1.4.2. 0x0004 GTC_CNTSR

默认值：0x00000000			GTC 计数器控制 (GTC Counter Control)
位域	类型	默认值	描述
31:10	–	–	–
9:8	RO	0x0	FCACK 当前计数频率 显示正在使用的计数频率点,可以通过查询本字段确定频率切换是否完成。 • 0：当前计数频率为基频 4 MHz • 1：当前计数频率为 1 MHz • 2：当前计数频率为 250 KHz • 3：未定义
7:0	–	–	–

6.1.4.3. 0x0008 GTC_CNTVL

默认值：0x00000000			GTC 计数值低位 (GTC Counter Value Low)
位域	类型	默认值	描述
31:0	R/W	0x00000000	CNTVL 计数值低位 显示计数器的低 32 bits。

默认值：0x00000000			GTC 计数值低位 (GTC Counter Value Low)
位域	类型	默认值	描述
			• 计数频率为 250 KHz 时，计数值保持低 4 bits 恒定为 0。 • 计数频率为 1 MHz 时，计数值保持低 2 bits 恒定为 0。 **计数器使能后，本寄存器禁止写操作。** **模块复位后，本寄存器清零。**

6.1.4.4. 0x000C GTC_CNTVH

默认值：0x00000000			GTC 计数值高位 (GTC Counter Value High)
位域	类型	默认值	描述
31:20	–	–	–
19:0	R/W	0x000000	CNTVH 计数值高位 本寄存器显示 52–bit 计数器的高 20 bits。 **计数器使能后，本寄存器禁止写操作。** **模块复位后，本寄存器清零。**

6.1.4.5. 0x0020 GTC_CNTFID0

默认值：0x003D0900			GTC 计数频率 ID0 (GTC Counter Frequencey ID0)
位域	类型	默认值	描述
31:0	RO	0x003D0900	CNTFID0 计数频率 0，单位为 Hz 保存计数器的计数频率 0 的频率值，即 4 MHz。

6.1.4.6. 0x0024 GTC_CNTFID1

默认值：0x000F4240			GTC 计数频率 ID1 (GTC Counter Frequencey ID1)
位域	类型	默认值	描述
31:0	RO	0x000F4240	CNTFID1 计数频率 1，单位为 Hz 保存计数器的计数频率 1 的频率值，即 1 MHz。

6.1.4.7. 0x0028 GTC_CNTFID2

默认值：0x0003D090			GTC 计数频率 ID2 (GTC Counter Frequencey ID2)
位域	类型	默认值	描述
31:0	RO	0x0003D090	CNTFID2 计数频率 2，单位为 Hz 保存计数器的计数频率 2 的频率值，即 250 KHz。

6.1.4.8. 0x002C GTC_CNTFID3

默认值：0x00000000			GTC 计数频率 ID3 (GTC Counter Frequencey ID3)
位域	类型	默认值	描述
31:0	RO	0x00000000	CNTFID3 计数频率 3

6.1.4.9. 0x00C0 GTC_CONFG

默认值：0x00000600			GTC 配置
位域	类型	默认值	描述
31:13	–	–	–
12:8	R/W	0x06	FDIV 总线时钟/计数基频，即 PCLK 与基频 4 MHz 的分频系数。 APB 总线时钟 PCLK 默认为 24 MHz，FDIV 可以采用默认值 6。
7:0	–	–	–

6.1.4.10. 0x0FFC GTC_VERSION

默认值：0x00000101			GTC 版本 (Version)
位域	类型	默认值	描述
31:0	R/W	0x00000101	VERSION 模块版本，V1.1

6.2. Watch Dog (WDOG)

Watch Dog (WDOG) 由一个 32 位计时器进行计数产生信号，用于监控系统。CPU 正常运行时，当计时器达到清零阈值后，可通过 CPU 操作定时清除 WDOG 计时器，保证 WDOG 模块不会超时并触发复位。当 WDOG 计时器先后达到中断和复位阈值时，WDOG 模块就会产生中断和复位。

6.2.1. 特性说明

- 固定 32K 时钟源输入，来源于 24 MHz 晶体，经过 750 分频
- 内部实现一个 32 位计时器，计时单位为一个 32K 时钟周期
- 实现两个信号输出，分别用于产生系统复位和中断信号
- 通过指令方式复位计时器和切换超时配置，支持写入窗口设置
- 调试模式下，可配置暂停计时或继续计时

6.2.2. 原理框图

6.2.3. 功能描述

6.2.3.1. WDOG 工作流程

图 6–1 : WDOG 工作流程涉及以下参数和步骤：

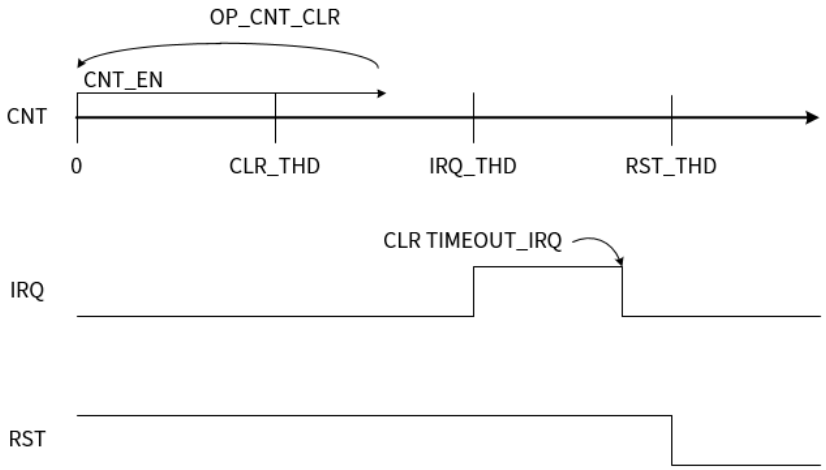


图 6–1 WDOG 工作流程

1. 计时器使能 (CNT_EN = 1) 并开始计时。计时以每个 32K 时钟周期为单位加一。

   使能计时器后，建议用户切换为写保护模式，具体配置可查看寄存器写保护功能。

2. 计时器达到清零阈值 (CNT > CLR_THD)。

- 如系统运行正常，即可通过 CPU 执行 OP_CNT_CLR 对计时器进行清零。此时不会产生中断或者复位。

  关于 OP_CNT_CLR 的操作说明，可查看 功能描述。

- 如系统出现异常，无法通过 CPU 对计时器进行清零，则计时器继续计时：

> **注：**
>
> 为了最大化防止以下中断或复位情况中 WDOG 功能被关闭，可设置寄存器写保护功能。

a. 计时器先达到中断阈值 (CNT > IRQ_THD)：计时器产生 CLR TIMEOUT_IRQ 中断，保存异常现场，该中断可被清除。

b. 计时器达到复位阈值 (CNT > RST_THD)：计时器产生复位，复位整个系统。

   如需实现强制产生复位信号，可以通过设置 FORCE_RST 寄存器，配置 0xFEDBCA98 进行强制复位。

6.2.3.2. 寄存器写保护功能

WDOG 具有寄存器写保护功能，针对异常状态下处理器的未知行为，以防止 WDOG 功能被关闭。寄存器写保护功能使能后，只有 OP 寄存器和 IRQ_STA 寄存器可被访问，只能执行 OP 指令和清中断操作。关于 OP 指令的详细说明，可查看 功能描述。

寄存器写保护功能具有多种模式，可通过 REG_WR_DIS 位域进行配置，具体模式和配置说明如下：

> **注：**
>
> 关于 REG_WR_DIS 的说明，可查看寄存器描述。

- 写允许 (REG_WR_DIS = 0)：正常写寄存器，在 WDOG 初始化时使用。

  设置 CNT_EN = 1 后建议切换为写失效模式。

- 写保护 (REG_WR_DIS = 1/2)：在需要动态配置 WDOG 的方案中使用。

  配置此寄存器前需先执行 OP_WR_EN，寄存器配置完成后再配置 REG_WR_DIS = 1/2，可以保护异常下不被误写。

- 写失效 (REG_WR_DIS = 3)：在无需动态配置 WDOG 方案中使用。

  配置完成后设置 CNT_EN = 1 && REG_WR_DIS = 3，运行期间配置不再被改写。

如有多场景需求且无需进行动态配置参数，可在初始化时完成多个场景配置后，设置 REG_WR_DIS = 3，应用过程只需要进行场景切换。 场景切换详细描述见场景切换。

6.2.3.3. 场景切换

WDOG 是基于 CLR_THD、IRQ_THD 和 RST_THD 三个参数为一组配置工作的，每个参数各有四个参数寄存器，用 n 进行区分，n = 0 ~ 3，详情可查看对应的寄存器描述。

通过设置参数寄存器，可组成不同的场景配置。WDOG 可支持保存最多四种组合场景，切换场景的 OP 指令分别为 OP_SW_CFG0、OP_SW_CFG1、OP_SW_CFG2 和 OP_SW_CFG3。

进行场景切换需要执行 OP 指令：

1. 执行 OP_SW_CFGn (n=0~3) 切换到配置好的第 n 个场景。

   若只需要一种场景，为了最大化防止异常情况下误切换了场景，可以把四个场景配置为相同参数。

2. 执行 OP_CNT_CLR 将计数器清零。

   新的一轮计数将按照新的场景参数运行。

关于 OP 指令的详细说明，可查看操作指令。

6.2.3.4. 操作指令

OP 寄存器是写失效下唯一可以进行写操作的寄存器， 通过连续两个写操作数与序列匹配，启动对应的指令。

现有支持指令如下表所示：

表 6–2 OP 指令定义

指令操作	指令名称	写操作数序列	备注
计时清零	OP_CNT_CLR	0xA1C55555 → 0xA1CAAAAA	需要满足 CNT > CLR_THD，操作才生效。
切换到配置 0	OP_SW_CFG0	0xA1C5A5A0 → 0xA1CA5A50	–
切换到配置 1	OP_SW_CFG1	0xA1C5A5A1 → 0xA1CA5A51	–
切换到配置 2	OP_SW_CFG2	0xA1C5A5A2 → 0xA1CA5A52	–
切换到配置 3	OP_SW_CFG3	0xA1C5A5A3 → 0xA1CA5A53	–
写使能	OP_WR_EN	0xA1C99999 → 0xA1C66666	在 REG_WR_DIS = 3 下无效。

6.2.4. 寄存器列表

表 6–3 WDOG 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	WDOG_CTL	看门狗控制	0x000 WDOG_CTL
0x004	WDOG_CNT	看门狗计时器	0x004 WDOG_CNT
0x008	IRQ_EN	中断使能	0x008 IRQ_EN
0x00C	IRQ_STA	中断状态	0x00C IRQ_STA
0x014	FORCE_RST	强制复位	0x014 FORCE_RST
0x040	CLR_THDn	清除阈值	0x040+n*0x10(n=0~3) CLR_THDn

表 6–3 WDOG 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x044	IRQ_THDn	中断阈值	0x044+n*0x10(n=0~3) IRQ_THDn
0x048	RST_THDn	复位阈值	0x048+n*0x10(n=0~3) RST_THDn
0x0E8	WDOG_OP	看门狗指令	0x0E8 WDOG_OP
0xFFC	WDOG_VER	看门狗版本	0xFFC WDOG_VER

6.2.5. 寄存器描述

6.2.5.1. 0x000 WDOG_CTL

默认值：0x00000000			看门狗控制 (Watch Dog Control)
位域	类型	默认值	描述
31:30	–	–	–
29:28	R/W	0x0	REG_WR_DIS 寄存器写失效 • 0：写允许 • 1：写保护。可通过 OP_WR_EN 回到允许写。除了 OP 寄存器，其它寄存器都不可写，可执行 OP 写使能将该寄存器清零。 • 2：写保护。可通过 OP_WR_EN 回到允许写。除了 OP 寄存器，其它寄存器都不可写，可执行 OP 写使能将该寄存器清零。 • 3：写失效，除了 OP 寄存器，其它寄存器都不可写。只能通过复位将寄存器清零，回到允许写。
27:24	RO	0x0	CFG_ID 配置场景 ID **配置当前使用的场景，复位后使用配置 0**
23:2	–	–	–
1	R/W	0x0	DBG_CNT_CONTINUE CPU 在调试模式下计时继续 • 0：计时暂停 • 1：计时继续

默认值：0x00000000			看门狗控制 (Watch Dog Control)
位域	类型	默认值	描述
			调试信号来源于处理器，如连接仿真器后进入调试模式。
0	R/W	0x0	CNT_EN 计时使能 • 0：计时暂停 • 1：计时启动

6.2.5.2. 0x004 WDOG_CNT

默认值：0x00000000			看门狗计时器 (Watch Dog Counter)
位域	类型	默认值	描述
31:0	R	0x0	WDOG_CNT 当前计时器值，单位为一个 32K 时钟周期 **注：** • CNT 每个计时时钟周期更新一次。 • 执行 OP_CLR_CNT 后需要等待一个 32K 时钟周期，才能保证该值被清零。

6.2.5.3. 0x008 IRQ_EN

默认值：0x00000000			中断使能 (Interrupt Request Enable)
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0	TIMEOUT_IRQ_EN 超时中断使能

6.2.5.4. 0x00C IRQ_STA

默认值：0x00000000			中断状态 (Interrupt Request Status)
位域	类型	默认值	描述
31:1	–	–	–

默认值：0x00000000			中断状态 (Interrupt Request Status)
位域	类型	默认值	描述
0	R/W1C	0	TIMEOUT_IRQ_STA 超时中断状态

6.2.5.5. 0x014 FORCE_RST

默认值：0x00000000			强制复位（Force Reset）
位域	类型	默认值	描述
31:0	R/W	0	Force Reset 0xFEDCBA98：强制复位 其它：Reserved

6.2.5.6. 0x040+n*0x10(n=0~3) CLR_THDn

默认值：0x00000000			清除阈值 (Clear Threshold)
位域	类型	默认值	描述
31:0	R/W	0	CLR_THDn 当计时大于该值，计时清零操作才能生效。

6.2.5.7. 0x044+n*0x10(n=0~3) IRQ_THDn

默认值：0x00000000			中断阈值 (Interrupt Request Threshold)
位域	类型	默认值	描述
31:0	R/W	0x27100	IRQ_THDn 当计时大于该值，产生中断信号。

6.2.5.8. 0x048+n*0x10(n=0~3) RST_THDn

默认值：0x00000000			复位阈值 (Reset Threshold)
位域	类型	默认值	描述
31:0	R/W	0x4e200	RST_THDn 当计时大于该值，产生系统复位。

6.2.5.9. 0x0E8 WDOG_OP

默认值：0x00000000			看门狗指令 (Watch Dog Operator)
位域	类型	默认值	描述
31:0	R/W	0x0	WDOG_OP 指令寄存器，详细说明见操作指令。

6.2.5.10. 0xFFC WDOG_VER

默认值：0x00000102			看门狗版本（Watch Dog Version）
位域	类型	默认值	描述
31:0	RO	0x00000102	VERSION 模块版本，V1.2

6.3. PWM

PWM 支持 PWM 输出和脉冲输出两种模式，主要应用于马达控制、高压控制、加热控制及其它 PWM 接口外设控制。

6.3.1. 特性说明

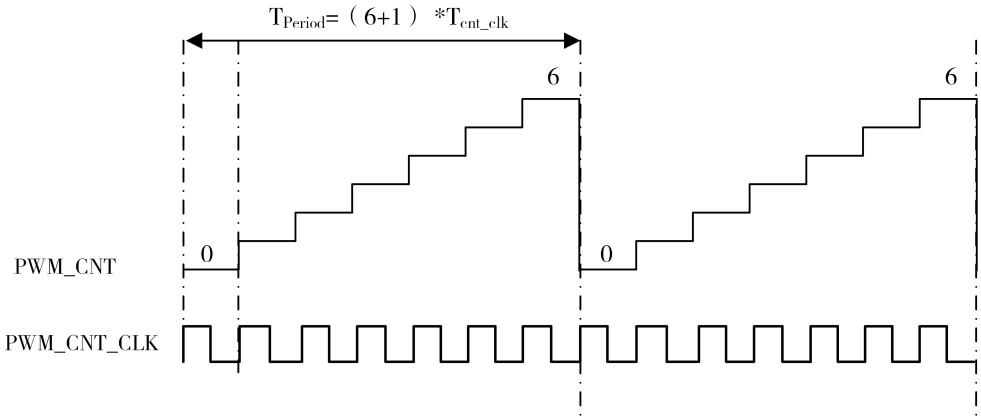
- 内置 32-bit 计数器
- 支持 PWM 输出模式和脉冲输出模式
- PWM 输出支持周期、占空比动态可调
- PWM 输出初始值可配置
- PWM 输出支持反相
- 脉冲输出模式支持 DMA 和中断
- 脉冲输出模式支持脉冲数量统计以及脉冲限制功能

6.3.2. 功能描述

6.3.2.1. 计数功能模块

PWM 计数器为 32-bit 的计数器，计数器计数时钟 PWM_CNT_CLK 由 SYS_CLK 进行分频得出，分频系数为 CLK_DIV，即 PWM_CNT_CLK = SYS_CLK / CLK_DIV。

另外，计数器采用递增的计数方式，通过设置周期 PWM_PRD 寄存器可以设置计数器的计数最大值，计数器达到 (PWM_PRD – 1) 值后重新复位到 0 开始递增计数，下图以 PWM_PRD = 7 为例演示计数功能：



6.3.2.2. PWM 输出模式

6.3.2.2.1. PWM 产生

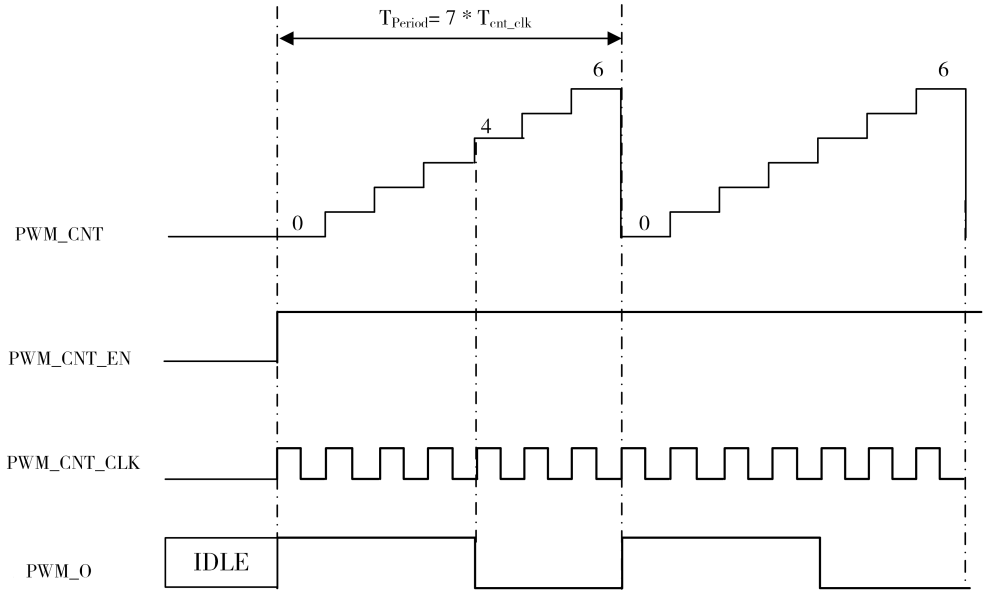
通过设置寄存器字段 PWM_MODE = 0x0，使 PWM 进入 PWM 输出模式。在该模式下，通过周期寄存器 PWM_PRD 和比较寄存器 PWM_CMP，可以调节 PWM 的频率以及占空比。

PWM 的生成方式，固定为在计数器 PWM_CNT = 0 的时刻，PWM 的输出为高电平。在 PWM_CNT = PWM_CMP 的时刻，PWM 的输出为低电平。

PWM 的输出频率和占空比计算方式如下：

- PWM_FRE = PWM_CNT_CLK / PWM_PRD
- PWM_DUTY = PWM_CMP / PWM_PRD
- PWM_CNT_CLK = SYS_CLK / DIV

下图以 PWM_PRD = 0x7，PWM_CMP = 0x4 的 PWM 输出为例，占空比为 4/7：



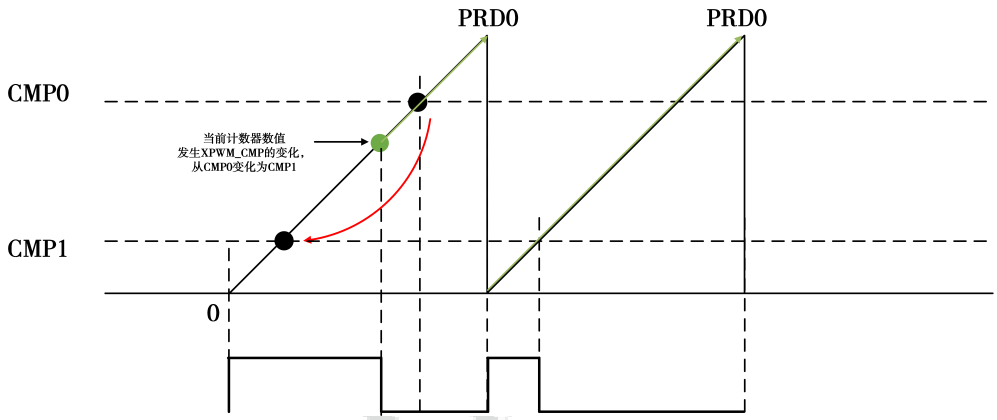
6.3.2.2.2. 动态更新

PWM 的周期值 PWM_PRD 和比较值 PWM_CMP，支持动态更新，更新的方式

- 立即更新：修改 PWM_PRD 和 PWM_CMP 寄存器后，PWM 立即生效。
- 影子更新：修改 PWM_PRD 和 PWM_CMP 寄存器后，在下一个周期进行 PWM 的更新，即下一个周期的 PWM_CNT_Z 对应的时刻进行更新。

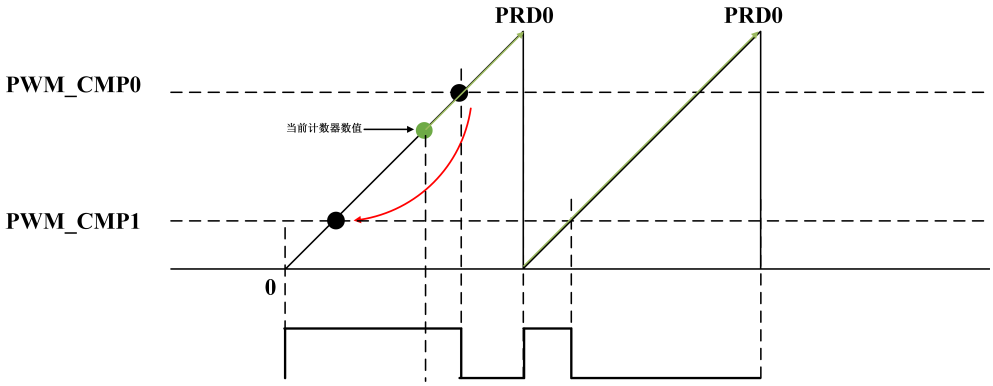
PWM_CMP 立即更新的举例：

PWM_CMP 从 CMP0 更新到 CMP1，其中 CMP1 小于 CMP0，也小于当前的计数器数值。



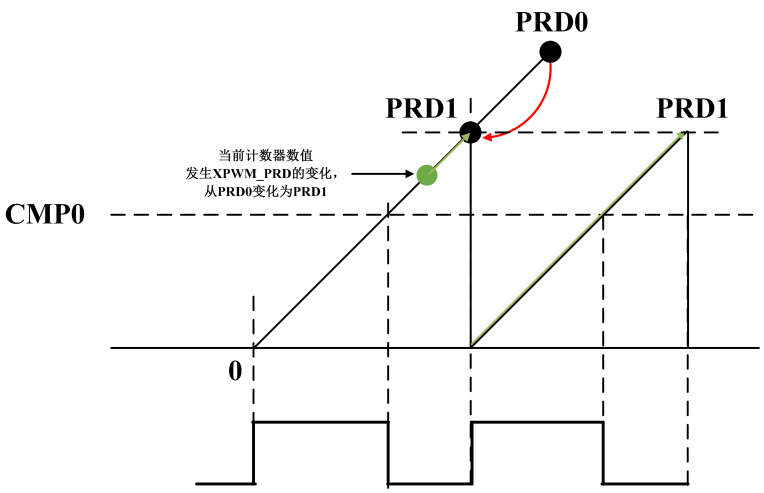
PWM_CMP 影子更新的举例

PWM_CMP 从 CMP0 更新到 CMP1，其中 CMP1 小于 CMP0，也小于当前的计数器数值。



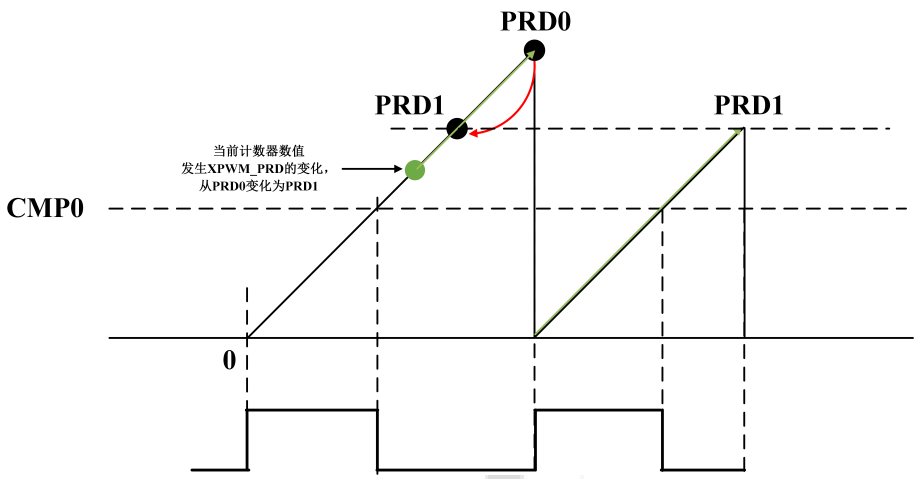
PWM_PRD 立即更新的举例：

PWM_PRD 从 PRD0 更新到 PRD1，其中 PRD1 小于 PRD0。



PWM_PRD 影子更新的举例：

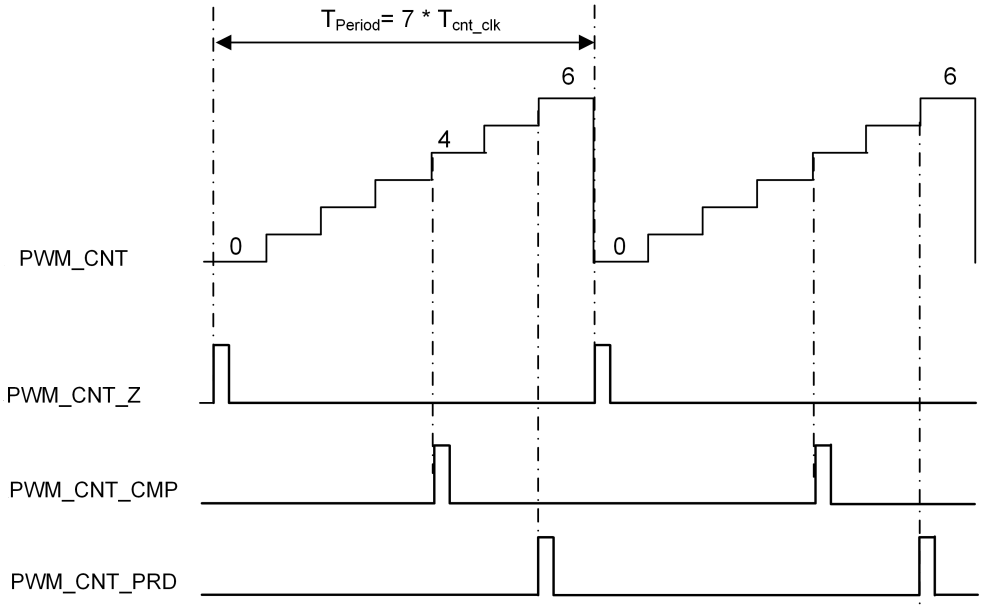
PWM_PRD 从 PRD0 更新到 PRD1，其中 PRD1 是小于 PRD0。



6.3.2.2.3. 中断信号

PWM 在 PWM 模式下，支持中断的产生，中断信号如下：

- PWM_CNT_Z：在计数器 PWM_CNT = 0 的时刻，对应产生的标记信号。
- PWM_CNT_CMP：在计数器 PWM_CNT = PWM_CMP 的时刻，对应产生的标记信号。
- PWM_CNT_PRD：在计数器 PWM_CNT =（PWM_CNT_PRD – 1）的时刻，对应产生的标记信号。



6.3.2.3. 脉冲输出模式

设置寄存器字段 PWM_MODE = 0x1，PWM 进入脉冲输出模式。在此模式下，通过配置以下寄存器来更新影子寄存器：

- PWM_PRD
- PWM_CMP
- 输出脉冲数量的寄存器 PUL_CNT

注：

通过上述寄存器更新影子寄存器需满足以下条件：

- **PUL_CNT 寄存器进行了写的操作。**
- **REG_UPDT 有效信号的到来。**

脉冲输出支持以下两种更新模式，通过配置 PWM_UPDATE_MODE 进行选择：

- 硬件更新：

  在 FIFO 模式下，FIFO 不为空时自动加载数据，非 FIFO 模式 IDLE 状态下写 PUL_CNT 会触发寄存器的影子更新。

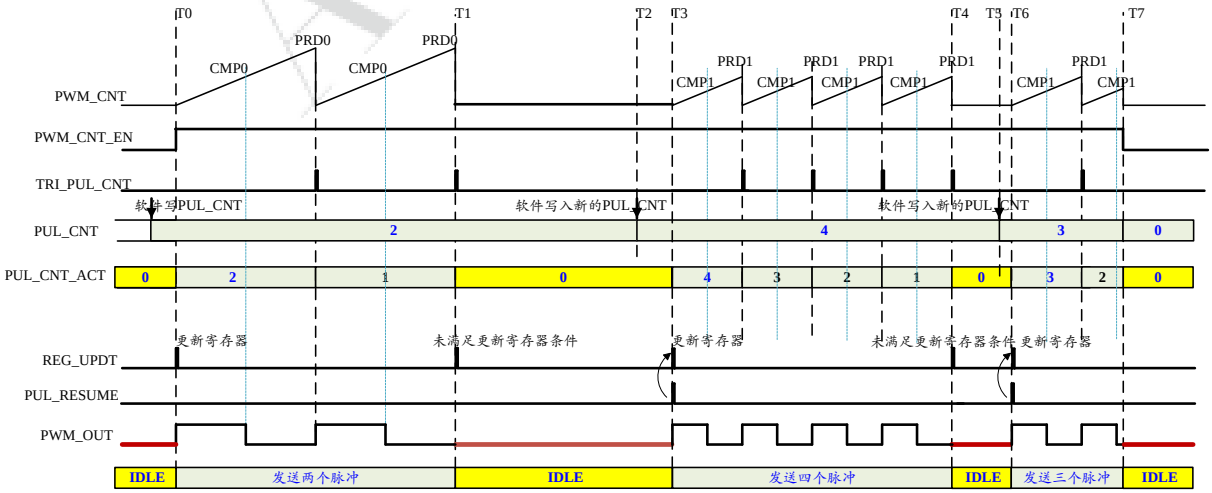
- 软件更新

  在 IDLE 下需要软件触发 resume 产生 REG_UPDT 信号，触发寄存器的影子更新。

脉冲输出模式支持三种方式实现脉冲输出的功能，分别是自由方式（未采用中断和 DMA）、持续方式（采用中断）、FIFO（采用中断或 DMA）方式。

6.3.2.3.1. 自由方式

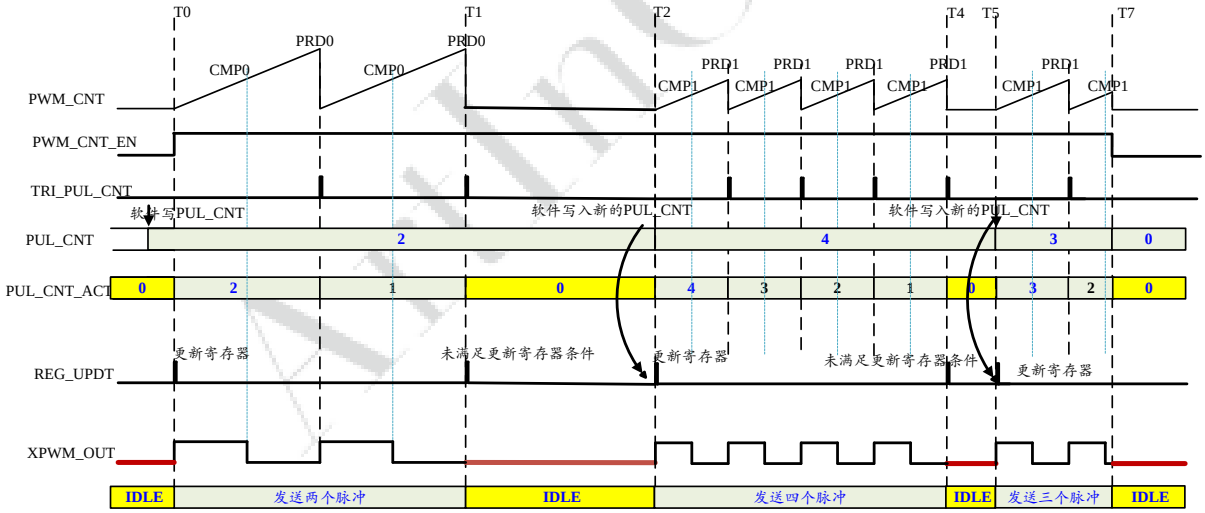
自由方式输出脉冲，每设置一次发送脉冲数量的数值，PWM 输出则发送相应数量的脉冲后则处于一直 IDLE 的状态（IDLE 状态可以设置为输出高电平或低电平）。以下为自由配置方式的脉冲输出例子：



如上图所示，为使用软件更新自由方式的脉冲输出，对应的说明如下：

1. 在脉冲输出的模式下，使能 PWM 的计数器，即 PWM_CNT_EN 从 0x0 跳变 0x1，此时产生一次的 REG_UPDT，即进行一次的寄存器更新。如上图的 T0 时刻。
2. 如上图，经过 T0 时刻的 REG_UPDT，PWM_PRD、PWM_CMP 和 PUL_CNT 加载影子寄存器的数值，即生效寄存器的情况为 PWM_PRD_ACT = PRD0、PWM_CMP_ACT = CMP0 和 PUL_CNT_ACT = 2。此时 PWM 的计数器开始计数，从零开始计数至 PRD0，在计数至 PRD0 触发一次 TRI_PUL_CNT，进而触发 PUL_CNT_ACT 自减一，此时 PWM 输出一个脉冲信号。如此重复，PWM 计数器再完成一次从零递增至 PRD0，PWM 再输出一个脉冲信号，并且此时 ACT 自减一，PUL_CNT_ACT = 0。

3. 在 PUL_CNT_ACT 跳变至零的时刻，产生一次 REG_UPDT 有效信号。如上图的例子，T1 时刻产生了一次有效的 REG_UPDT 信号，但是由于没有发生对 PUL_CNT 寄存器的写操作，此时并不会进行一次影子更新。
4. 在 PUL_CNT_ACT = 0 的状态下，进入 IDLE 状态，PWM 的计数器停止计数，上述例子中，PWM 的输出信号强制设置为低电平状态（此状态可以通过寄存器配置为高/低电平）。
5. 如上图示例，在 T3 时刻软件通过对寄存器字段 PUL_RESUME 写入 0x1，产生了一次的 REG_UPDT 有效信号，由于在 T2 时刻已对 PUL_CNT 寄存器进行写操作，所以此次的 REG_UPDT 会进行影子寄存器的更新。另外这里会将计数器重置为零开始计数。
6. 从 T4 时刻到 T6 时刻均为类似的过程。
7. 最后上图的 T7 时刻，将 PWM_CNT_EN 拉低为 0x0，PWM 计数器停止计数回到零，PWM_PRD、PWM_CMP 和 PUL_CNT 的影子寄存器均自动清零，PWM 的输出为 IDLE 状态。

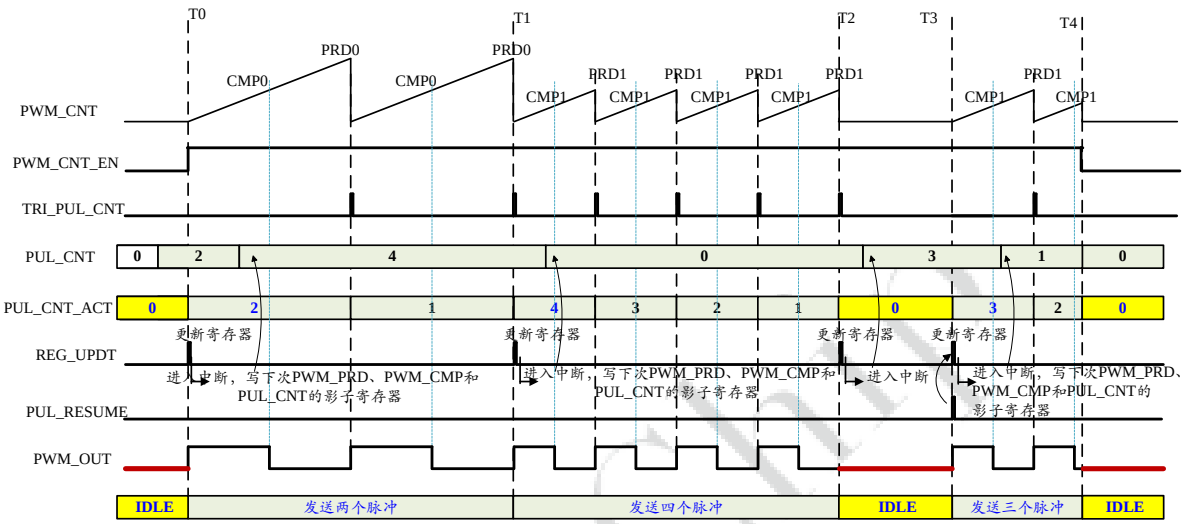


如上图所示，为使用硬件更新自由方式的脉冲输出，对应的说明如下：

1. 硬件更新和软件更新类似，在 T0 时刻使能 PWM，脉冲信号开始输出。
2. T2 时刻，在 IDLE 状态对 PUL_CNT 寄存器进行写操作，会触发影子寄存器的更新，脉冲信号开始输出。T5 时刻类似。
3. 最后 T7 时刻，PWM_CNT_EN 拉低为 0x0，PWM 计数器停止计数回零，PWM 的输出为 IDLE 状态。

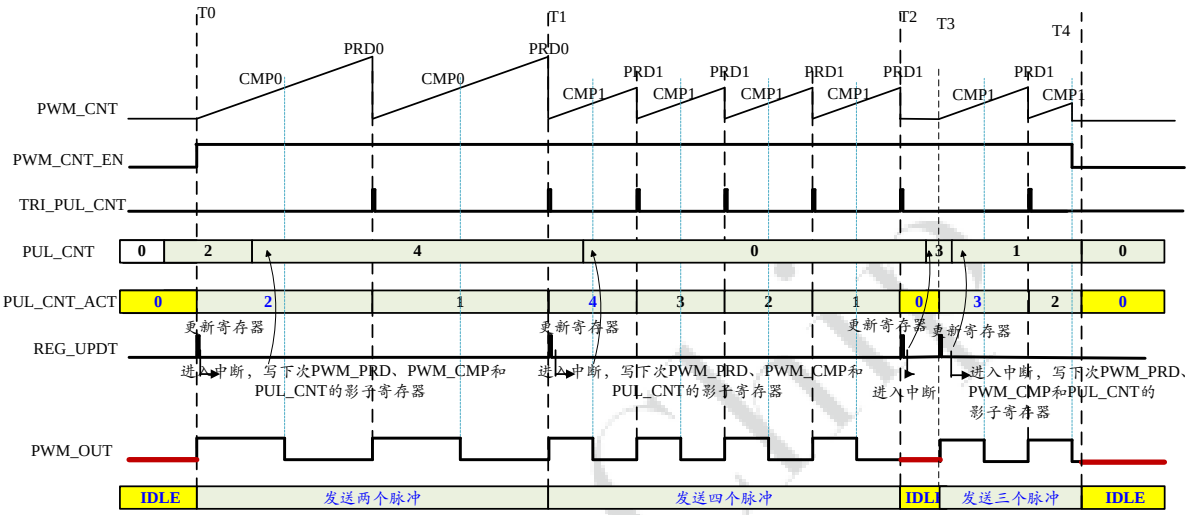
6.3.2.3.2. 持续方式

中断方式输出脉冲，即通过中断，每次中断函数设置下次发送脉冲数量的数值，PWM 输出则一直按照设置的数值持续发送脉冲。以下为持续方式的脉冲输出例子：



如上图所示，为使用软件更新持续方式的脉冲输出，其中 REG_UPDT 信号在更新寄存器后同时产生中断信号通知 CPU 进入中断函数，对应的说明如下：

1. 在 T0 时刻之前，PWM_CNT_EN = 0x0，写入一次 PRD、PWM_CMP 和 CNT 的影子寄存器数值，PWM_PRD = PRD0、PWM_CMP = CMP0 和 PUL_CNT = 2。
2. 在 T0 时刻使能 PWM 的计数器进行计数，PWM_CNT_EN = 0x1，这时产生一次 REG_UPDT，进行一次的寄存器更新，更新 T0 时刻之前写入的数值。更新的同时也产生一次中断信号进入中断函数，中断函数需要写入下一次的影子寄存器的数值，这里的设置是 PWM_PRD = PRD1、PWM_CMP = CMP1 和 PUL_CNT = 4。
3. 在 T1 时刻，PUL_CNT_ACT 会从 0x1 递减至 0x0，并且产生一次寄存器更新信号 REG_UPDT，加载了新的影子寄存器值。
4. 在 T 1 时刻更新寄存器的同时也会进入中断函数，软件会再写入下一次的影子寄存器值，PWM_PRD = PRD1、PWM_CMP = CMP1 和 PUL_CNT = 0。
5. 在 T2 时刻，更新了脉冲输出数为零的情况，此时 PWM 输出为 IDLE 状态。需等待 PUL_RESUME 信号的到来才可以继续进行脉冲输出的功能。
6. 在 T3 时刻，软件更新出现软件写 PUL_RESUME = 0x1，此时会产生一次的 REG_UPDT 有效信号。进行影子寄存器的更新并且计数器复位为零，这样脉冲输出的功能则按照之前的同样方式持续进行。
7. 最后在 T4 时刻，PWM_CNT_EN = 0x0，计数器停止计数，输出为 IDLE 状态。



如上图所示，为使用硬件更新持续方式的脉冲输出，其中 REG_UPDT 信号在更新寄存器后同时产生中断信号通知 CPU 进入中断函数，对应的说明如下：

1. T2 时刻前，硬件更新与软件更新一致，寄存器更新的同时也产生一次中断信号进入中断函数，中断函数需要写入下一次的影子寄存器的数值，脉冲信号持续输出。
2. T2 时刻，更新的脉冲输出数为零的情况，PWM 输出为 IDLE 状态。T3 时刻写入 PUL_CNT，会进行影子寄存器的更新并且计数器复位为零，这样脉冲输出的功能则按照之前的同样方式持续进行。
3. 最后在 T4 时刻，PWM_CNT_EN = 0x0，计数器停止计数，输出为 IDLE 状态。

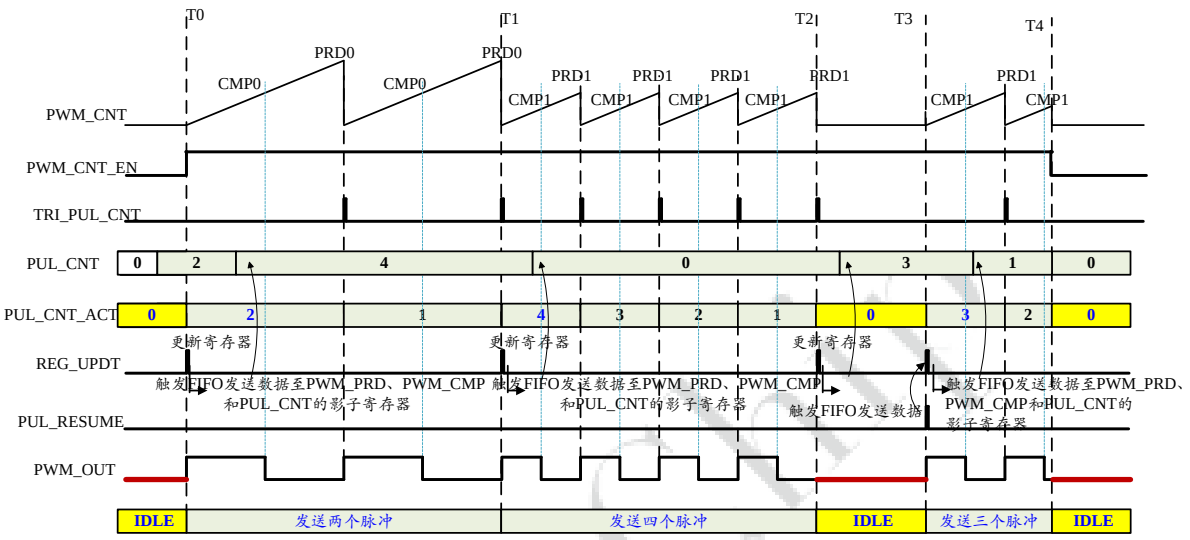
6.3.2.3.3. FIFO 方式

FIFO 方式输出脉冲，是利用内置的 FIFO 对影子寄存器更新的方式，实现持续发送脉冲。这样相对上面提到的持续方式，可以大大提高脉冲输出的效率，避免频繁进入中断函数。

针对 FIFO 方式的脉冲输出功能，内置了 24 * 32–bit 的 FIFO（最多写入八次的脉冲输出组合），用于存放 PUL_CNT、PWM_PRD、PWM_CMP。在触发信号 REG_UPDT 的作用下，FIFO 可以依次往 XPMW_PRD、PWM_CMP、PUL_CNT 进行写入新的影子寄存器值，因此 FIFO 写入也需按此顺序依次写入。

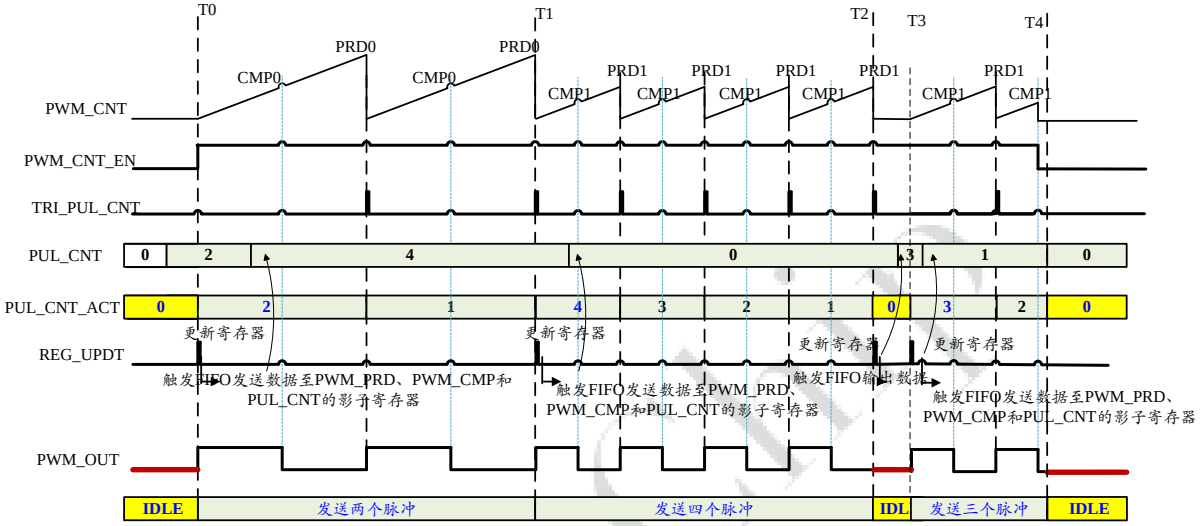
另外此 FIFO 支持 DMA 或中断的方式写入，通过设置 FIFO 的阈值，使能 DMA 请求或 FIFO 中断信号，即可进行数据的写入。

以下为 FIFO 方式的脉冲输出例子：



如上图所示，为软件更新 FIFO 方式的脉冲输出，其中 REG_UPDT 信号在更新寄存器后同时产生，触发 FIFO 进行更新影子寄存器的数值，对应的说明如下：

1. 在 T0 时刻之前，即 PWM_CNT_EN = 0x0，FIFO 需提前备好数据，可以通过中断或 DMA 进行写入。
2. 在 T0 时刻使能 PWM 的计数器进行计数，PWM_CNT_EN = 0x1，这时产生一次的 REG_UPDT，进行一次的寄存器更新，更新 T0 时刻之前写入的数值。更新的同时触发 FIFO 发送数据写入下一次的 PWM_PRD、PWM_CMP 和 PUL_CNT，这里的设置是 PWM_PRD = PRD1、PWM_CMP = CMP1 和 PUL_CNT = 4。
3. 在 T1 时刻，PUL_CNT_ACT 从 0x1 递减至 0x0，并且产生一次寄存器更新信号 UPDT，加载了新的影子寄存器值。
4. T1 时刻更新寄存器之后 ，FIFO 随后也写入新的影子寄存器的数值，PWM_PRD = PRD1、PWM_CMP = CMP1 和 PUL_CNT = 0。
5. 在 T2 时刻，更新了脉冲输出数为零的情况，此时 PWM 输出为 IDLE 状态。需等待 PUL_RESUME 信号的到来才可以继续进行脉冲输出的功能。
6. 在 T3 时刻，出现软件写 PUL_RESUME = 0x1，此时会重置计数器为零，并且产生一次 UPDT 有效信号，这样脉冲输出的功能则按照之前的同样方式持续进行。
7. 最后在 T4 时刻，PWM_CNT_EN = 0x0，计数器停止计数，输出为 IDLE 状态。



如上图所示，为硬件更新 FIFO 方式的脉冲输出，其中 REG_UPDT 信号在更新寄存器后同时产生，触发 FIFO 进行更新影子寄存器的数值，对应的说明如下：

T2 时刻前，硬件更新和软件更新一致。当 T2 时刻，PWM 输出为 IDLE 状态且此时 FIFO 不为空，硬件更新自动加载配置，更新影子寄存器继续输出脉冲信号。

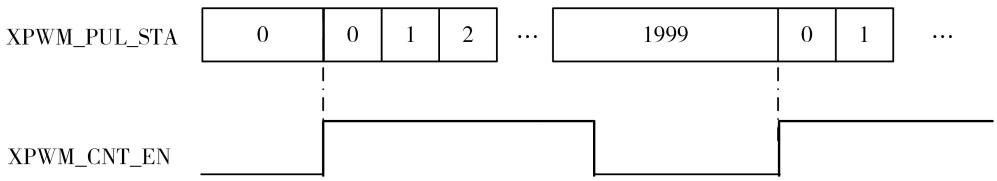
6.3.2.3.4. 软件触发 RESUME 功能

软件触发，可以通过对寄存器字段 PUL_RESUME 写 0x1，硬件产生一次 REG_UPDT 触发信号，该触发信号能够强制进行一次对 PUL_CNT、PWM_PRD、PWM_CPM 寄存器的影子更新，并且强制 CNT 复位至零。

6.3.2.3.5. 脉冲数量统计功能

在脉冲输出模式下，支持对发送脉冲数量的统计，统计的数量保存在寄存器 PWM_PUL_STA 中。

如下图所示，在 PWM_CNT_EN = 0x1 的时刻（即上升沿），PWM_PUL_STA 寄存器自动清零，从零开始统计发送的脉冲数量。在 PWM_CNT_EN 从 0x1 跳变为 0x0，此时 PWM_PUL_STA 寄存器保持当前的数据，只有在下一次的 PWM_CNT_EN 使能或对 PWM_PUL_STA_FLUSH 写 1，才会自清零开始下一次的脉冲数量统计。



6.3.2.3.6. 脉冲数量限制功能

在脉冲输出模式下，通过设置寄存器字段 PUL_LIMIT_EN = 0x1 可以使能脉冲数量限制的功能。脉冲数量限制功能是当前已发送脉冲数量 PWM_PUL_STA 等于所设置的脉冲阈值 PUL_THR 时，会触发中断状态 PWM_PUL_LIMIT。

当 PWM 达到 PUL_THR 后，PWM 的计数器停止计数，脉冲也会停止输出。如需继续按照之前的配置情况重新启动 PWM，则对寄存器字段 PUL_THR_RS 写 0x1，触发 PWM 的计数器继续进行计数。需注意此时需要配置新的 PUL_THR，否则会一直存在已发送脉冲数量 PWM_PUL_STA 大于 THR 的情况。

6.3.2.3.7. 强制脉冲输出功能

在脉冲输出模式下，对 PUL_CNT 写入 0xFFFF FFFF 后，脉冲输出进入强制输出模式，在此模式下，PWM 持续发送脉冲，发送脉冲的频率和占空比可以通过 PWM_PRD 和 PWM_CPM 寄存器动态调整。此时脉冲数量统计以及脉冲数量限制的功能仍然生效。

如需退出强制脉冲输出功能，则需通过对 PUL_CNT 写入非 0xFFFF FFFF 的脉冲数量值，然后在对寄存器 PUL_RESUME 写 0x1，触发一次软件更新信号，进行 CNT 的更新。

另外，在进入强制脉冲输出模式的时刻，支持触发中断信号 PWM_FRC_PUL。

6.3.2.3.8. 中断信号

在脉冲输出的模式下，支持的中断信号包括如下：

- REG_UPDT，更新寄存器信号中断
- FIFO_AVAL，FIFO Available 信号中断，即 FIFO 的数据低于设定阈值时产生的中断信号
- FIFO_EMPTY，FIFO 空信号中断，即此时 FIFO 无数据
- FIFO_FULL，FIFO 满信号中断，即此时 FIFO 已满
- PWM_FRC_PUL，输出进入了强制输出脉冲模式
- PWM_PUL_LIMIT，脉冲数量达到限制值 PUL_THR

6.3.2.4. 输出 IDLE

通过寄存器可以设置在 IDLE 状态时输出的电平，PWM_IDLE = 0x0 表示低电平，PWM_IDLE = 0x1 表示高电平。

在 PWM 输出的模式下，在 PWM_CNT_EN = 0x0 的时候，PWM 的输出为 IDLE 状态。而在脉冲输出的模式下，在 PUL_CNT = 0 的时候，PWM 的输出为 IDLE 状态。

6.3.2.5. 输出反相

PWM 的输出可以通过输出反相的功能对输出信号进行反相，此功能对于 PWM 输出模式以及脉冲输出模式均有效。输出反相功能为 PWM 的输出的最后一级功能，包括 IDLE 状态也会被反相。

6.3.3. 寄存器列表

表 6–4 PWM 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	PWM_CONF	PWM 配置寄存器	0x000 PWM_CONF
0x004	PWM_CH_CONF	PWM 通道控制寄存器	0x004 PWM_CH_CONF
0x008	PWM_STS_FLAG	PWM 状态标记寄存器	0x008 PWM_STS_FLAG
0x00C	PWM_INT_EN	PWM 中断使能寄存器	0x00C PWM_INT_EN

表 6–4 PWM 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x010	PWM_FIFO_FLUSH	PWM FIFO 清空寄存器	0x010 PWM_FIFO_FLUSH
0x014	PWM_RESUME	PWM 脉冲 RESUM 使能寄存器	0X014PWM_RESUME
0x018	PUL_THE_RSTHE_RS	PWM 脉冲重启使能寄存器	0x018 PUL_THR_RS
0x020	PWM_FIFO	PWM FIFO 寄存器	0x020 PWM_FIFO
0x024	PWM_PUL_STA	PWM 脉冲统计寄存器	0x24 PWM_PUL_STA
0x028	PWM_CNT_STA	PWM Counter 计数数值	0x28 PWM_CNT_STA
0x050	PWM_PRD	PWM Counter 计数周期	0x50 PWM_PRD
0x054	PUL_CNT	PWM Pulse 输出设置值	0x54 PUL_CNT
0x058	PUL_THR	PWM 脉冲阈值寄存器	0x58 PUL_THR
0x060	PWM_CMP	PWM Counter 计数比较值	0x60 PWM_CMP
0xFC	PWM_VERSION	PWM 模块版本寄存器	0xFC PWM_VERSION

6.3.4. 寄存器描述

6.3.4.1. 0x000 PWM_CONF

默认值：0x00000C00			PWM 配置寄存器 (PWM_CONF)
位域	类型	默认值	描述
31:26	–	–	–
25:16	R/W	0x0	CLK_DIV PWM Clock Divisor，PWM 计数器的时钟分频系数。计数器的计数时钟为 SYS_CLK / PWM_CLK_DIV。 • 0x0：1 • 0x1：2 • 0x2：3 • 0x3：4

默认值：0x00000C00			PWM 配置寄存器 (PWM_CONF)
位域	类型	默认值	描述
			• … • 0x3FF：1024
15:8	R/W	0xC	PWM_FIFO_TH PWM FIFO Threshold，FIFO 阈值设置。当 FIFO 的数据量低于这里设置的阈值，那么可以产生 DMA 请求或中断信号。 • 0x0：1 • 0x1：2 • 0x2：3 • 0x3：4 … • 0x17：24 • Others：24
7	R/W	0x0	PUL_LIMIT_EN Pulse Limit Enable，脉冲限制使能，使能后寄存器 THR 生效。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	PWM_DMA_EN PWM DMA Enable，PWM 的 DMA 使能，此位只有在 PWM_MOD = 0x1 的时候有效。使能后，可以通过 DMA 对 FIFO 的数据进行填充。 • 0x0：未使能 • 0x1：使能
5	–	–	–
4	R/W	0x0	PWM_FIFO_EN PWM FIFO Enable，FIFO 使能，此位只有在 PWM_MOD = 0x1 的时候有效。使能后，脉冲输出可以采用 FIFO 方式控制脉冲产生。 • 0x0：未使能 • 0x1：使能
3	–	–	–
2	R/W	0x0	PWM_UPDATE_MODE PWM UPDATE Mode，PWM 更新模式，此位只有在 PWM_MOD = 0x1 的时候有效。硬件更新在 FIFO 模式下自动加载配置。

默认值：0x00000C00			PWM 配置寄存器 (PWM_CONF)
位域	类型	默认值	描述
			• 0x0：硬件更新 • 0x1：软件更新
1	R/W	0x0	PWM_MOD PWM Operation Mode，PWM 工作模式 • 0x0：PWM 输出模式 • 0x1：脉冲输出模式
0	R/W	0x0	PWM_CNT_EN PWM Counter Enable，PWM 计数器使能。使能后 PWM 计数器开始计数。 • 0x0：未使能 • 0x1：使能

6.3.4.2. 0x004 PWM_CH_CONF

默认值：0x00000000			PWM 通道控制寄存器 (PWM_CH_CONF)
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	PWM_PUL_STA_MODE PWM Pulse Statistics Mode，脉冲输出统计模式 • 0x0：外部计数，利用脉冲边缘计数 • 0x1：内部计数，利用 PWM 内部计数器进行计数
3	R/W	0x0	PWM_ PUL_STA_POL PWM Pulse Statistic Polarity Set 脉冲输出统计极性控制，当 PWM_PUL_STA_MODE = 0x0 时有效 • 0x0：上升沿触发 • 0x1：下降沿触发
2	R/W	0x0	PWM_IDLE PWM Idle Status Set，PWM 在 IDLE 状态的输出电平设置。 • 0x0：IDLE 状态设置输出为低电平 • 0x1：IDLE 状态设置输出为高电平
1	R/W	0x0	PWM_INV_EN

默认值：0x00000000			PWM 通道控制寄存器 (PWM_CH_CONF)
位域	类型	默认值	描述
			PWM Output Invert Enable，PWM 模块输出反相使能。使能后 PWM 输出信号进行反相。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	PWM_IMD_UPDT PWM Immediate Update Enable，PWM 立即更新使能。此位只有在 PWM_MOD = 0x0 有效。此位使能后，PWM_PRD 以及 CMP 采用立即更新的方式。 • 0x0：未使能 • 0x1：使能

6.3.4.3. 0x008 PWM_STS_FLAG

默认值：0x00000000			PWM 状态标记寄存器 (PWM_STS_FLAG)
位域	类型	默认值	描述
31:9	–	0x0	–
8	R/W1C	0x0	CNT_CMP_FLG Counter Compare Flag，计数器数值达到 PWM_CMP 的标记信号。 • 0x0：无标记信号产生 • 0x1：产生标记信号
7	R/W1C	0x0	PWM_PUL_LIMIT _FLG PWM Pulse Limit Flag，PWM 脉冲输出达到输出阈值标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
6	R/W1C	0x0	PWM_FRC_PUL _FLG PWM Force Pulse Output Flag，PWM 进入强制脉冲输出标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
5	R/W1C	0x0	FIFO_OVFL_FLG FIFO Overflow Flag，FIFO 上溢出标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号，FIFO 出现上溢出的状态

默认值：0x00000000			PWM 状态标记寄存器 (PWM_STS_FLAG)
位域	类型	默认值	描述
4	R/W1C	0x0	FIFO_UDFL_FLG FIFO Underflow Flag，FIFO 下溢出标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号，FIFO 出现下溢出的状态
3	R/W1C	0x0	FIFO_AVAL_FLG FIFO Available Flag，FIFO 可用标记。在 FIFO 的数据量低于 FIFO 阈值时，此标记产生。 • 0x0：无标记信号产生 • 0x1：产生标记信号
2	R/W1C	0x0	REG_UPDT_FLG Register Update Flag，寄存器更新标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
1	R/W1C	0x0	CNT_PRD_FLG Counter Period Flag，计数器数值达到 PWM_PRD 的标记信号。 • 0x0：无标记信号产生 • 0x1：产生标记信号
0	R/W1C	0x0	CNT_ZRO_FLG Counter Zero Flag，计数器数值达到零的标记信号。 • 0x0：无标记信号产生 • 0x1：产生标记信号

6.3.4.4. 0x00C PWM_INT_EN

默认值：0x00000000			PWM 中断使能寄存器 (PWM_INT_EN)
位域	类型	默认值	描述
31:9	–	0x0	–
8	R/W	0x0	CNT_CMP_INT_EN Counter Compare Interrupt Enable，PWM 计数器达到 CMP 的标记中断使能。

默认值：0x00000000			PWM 中断使能寄存器 (PWM_INT_EN)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
7	R/W	0x0	PWM_PUL_LIMIT_ INT_EN PWM Pulse Limit Interrup Enable，PWM 脉冲输出达到输出阈值的中断使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	PWM_FRC_PUL_INT_EN PWM Fource Pulse Output Interrup Enable，PWM 进入强制脉冲输出的中断使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	FIFO_OVFL_INT_EN FIFO Overflow Interrupt Enable，FIFO 上溢出中断使能 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	FIFO_UDFL_INT_EN FIFO Underflow Interrupt Enable，FIFO 下 溢出中断使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	FIFO_AVAL_INT_EN FIFO Available Interrupt Enable，FIFO 中断使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	REG_UPDT_INT_EN Register Update Interrupt Enable，寄存器更新中断使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	CNT_PRD_INT_EN Counter Period Interrupt Enable，PWM 计数器达到 PWM_PRD 的标记中断使能。

默认值：0x00000000			PWM 中断使能寄存器 (PWM_INT_EN)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
0	R/W	0x0	CNT_ZRO_INT_EN Counter Zero Interrupt Enable，PWM 计数器达到零的标记中断使能。 • 0x0：未使能 • 0x1：使能

6.3.4.5. 0x010 PWM_FIFO_FLUSH

默认值：0x00000000			PWM FIFO 清空寄存器 (PWM_FIFO_FLUSH)
位域	类型	默认值	描述
31:24	–	0x0	–
23:16	R	0x0	FIFO_DATA_LEN FIFO Data Length，FIFO 缓存数据大小，指示当前 FIFO 缓存的数据量，单位是 32–bit。
15:1	–	0x0	–
0	W1AC	0x0	PWM_FIFO_FLUSH PWM FIFO Flush，清除 FIFO，此位写 1 自动清除 FIFO 的数据，完成后此位自动清 0。

6.3.4.6. 0X014PWM_RESUME

默认值：0x00000000			PWM 脉冲 RESUME 使能寄存器 （PWM_RESUME)
位域	类型	默认值	描述
31:5	–	–	–
4	W1C	0x0	PWM_PUL_STA_FLUSH PWM PUL STA FLUSH，脉冲输出统计清零，此位写 1 脉冲输出统计个数置为 0，完成后此位自动清 0。
3:1	–	–	–
0	W1AC	0x0	PUL_RESUME

默认值：0x00000000			PWM 脉冲 RESUME 使能寄存器（PWM_RESUME)
位域	类型	默认值	描述
			Pulse Resume，脉冲继续发送使能。写 1 使能后，产生一次的 UPDT 的有效信号，并且此位自动清 0。

6.3.4.7. 0x018 PUL_THR_RS

默认值：0x00000000			PWM 脉冲重启使能寄存器 (PUL_THR_RS)
位域	类型	默认值	描述
31:1	–	0x0	–
0	W1AC	0x0	PUL_THR_RS Pulse Threshold Restart，脉冲重启发送使能。写 1 使能后，重新启动计数器，触发脉冲继续输出，完成后此位自动清 0。此位用于在脉冲输出达到脉冲阈值后，重启计数器的操作。

6.3.4.8. 0x020 PWM_FIFO

默认值：0x00000000			PWM FIFO 寄存器 (PWM_FIFO)
位域	类型	默认值	描述
31:0	R/W	0x0	PWM_FIFO PWM FIFO Address Register，FIFO 地址寄存器，利用此地址写入 FIFO 数据。

6.3.4.9. 0x24 PWM_PUL_STA

默认值：0x00000000			PWM 脉冲统计寄存器 (PWM_PUL_STA)
位域	类型	默认值	描述
31:0	R	0x0	PWM_PUL_STA PWM Pulse Statistics，脉冲输出统计，用于统计脉冲发出的数量。

6.3.4.10. 0x28 PWM_CNT_STA

默认值：0x00000000			PWM Counter 计数数值 (PWM_CNT_STA)
位域	类型	默认值	描述
31:0	R	0x0	PWM_CNT_STA PWM Counter Status Value，PWM 计数器的计数值。

6.3.4.11. 0x50 PWM_PRD

默认值：0x00000000			PWM Counter 计数周期 (PWM_PRD)
位域	类型	默认值	描述
31:0	R/W	0x0	PWM_PRD PWM Counter Period Value，PWM 计数器的周期值。计数器计数从 0 计数到 PWM_PRD – 1，完成一个周期。

6.3.4.12. 0x54 PUL_CNT

默认值：0x00000000			PWM Pulse 输出设置值 (PUL_CNT)
位域	类型	默认值	描述
31:0	R/W	0x0	PUL_CNT Pulse Counter，脉冲输出设置值，此寄存器只有在脉冲输出模式有效，并且采用影子更新。

6.3.4.13. 0x58 PUL_THR

默认值：0xFFFFFFFF			PWM 脉冲阈值寄存器 (PUL_THR)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	PUL_THR Pulse Threshold，发送脉冲阈值，此域在 PUL_LIMIT_EN = 0x1 有效，即脉冲数量限制功能使能后有效。

6.3.4.14. 0x60 PWM_CMP

默认值：0x00000000			PWM Counter 计数比较值（PWM_CMP)
位域	类型	默认值	描述
31:0	R/W	0x0	PWM_CMP PWM Counter Compare Value，PWM 计数器的比较值。

6.3.4.15. 0xFC PWM_VERSION

默认值：0x00000100			PWM 模块版本寄存器 (PWM_VERSION)
位域	类型	默认值	描述
31:0	RO	0x00000100	Version （版本号)

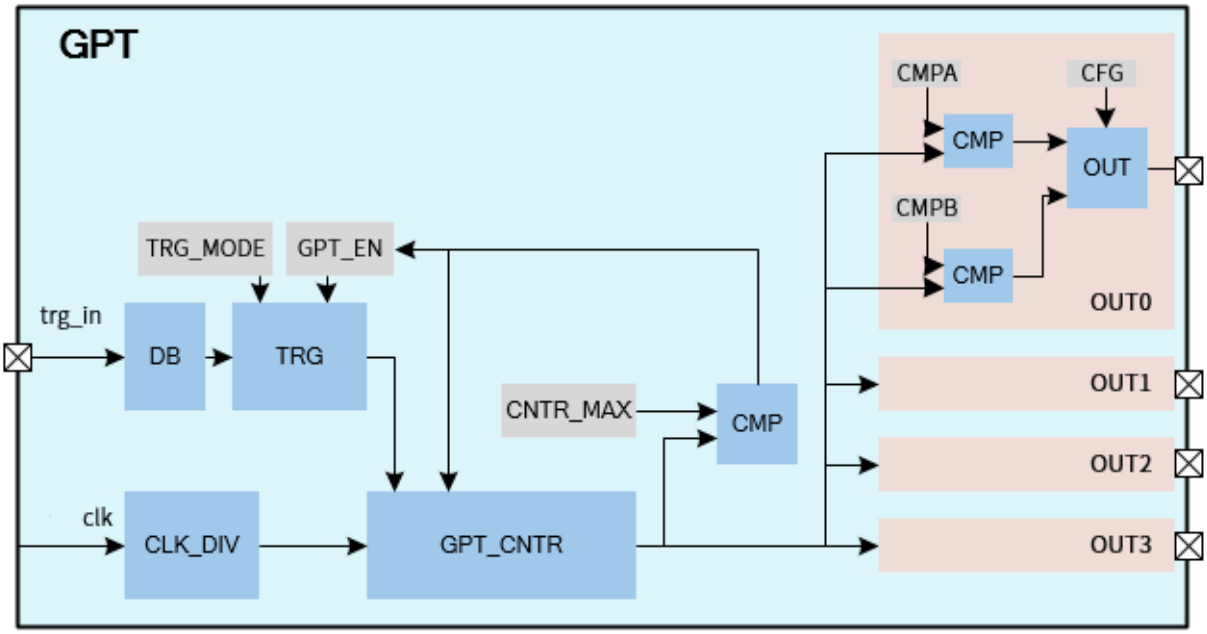
6.4. General Purpose Timer (GPT)

General Purpose Timer (GPT)，通用计数器，可实现触发模式以及更加精确的计时、计数功能。

6.4.1. 特性说明

- 内置 32-bit 计数器
- 计数器运行模式可选为单次模式和循环模式
- 计数器启动条件可选为自动触发和输入信号触发
- 四路输出比较信号，用于产生中断和引脚输出，可配置输出高低电平时间

6.4.2. 原理框图



6.4.3. 寄存器列表

表 6–5 GPT 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	GPT_CTL	通用计数器控制	0x000 GPT_CTL
0x004	GPT_CLK	通用计数器时钟	0x004 GPT_CLK
0x008	IRQ_EN	中断使能	0x008 IRQ_EN
0x00C	IRQ_STA	中断状态	0x00C IRQ_STA
0x010	CNTR_VAL	计数器值	0x010 CNTR_VAL
0x014	CNTR_MAX	计数器最大值	0x014 CNTR_MAX
0x01C	PWM_TBSTS	计数器清零	0x01C CNTR_CLR
0x040+N*0x10 (N=0~3)	OUTn_CMPA	输出 N 比较 A 寄存器	0x040+N*0x10 (N=0~3) OUTn_CMPA
0x044+N*0x10 (N=0~3) OUTn_CMPB	OUTn_CMPB	输出 N 比较 B 寄存器	0x044+N*0x10 (N=0~3) OUTn_CMPB

表 6–5 GPT 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x04C +N*0x10 (N=0~3)	OUTn_CFG	输出 N 配置寄存器	0x04C+N*0x10 (N=0~3) OUTn_CFG
0x0FC	GPT_VERSION	PWM 比较器 A 寄存器	0x0FC GPT_VERSION

6.4.4. 寄存器描述

6.4.4.1. 0x000 GPT_CTL

默认值：0x00000000			通用计数器控制（General Purpose Timer Control）
位域	类型	默认值	描述
31:29	–	–	–
28	R	0x0	TRG_LEVEL 输入触发引脚电平 • 0: 低电平 • 1: 高电平
27:24	–	–	–
23:20	R/W	0x0	TRG_DB1_DIV 第二级去抖采样除频除频系数为 2^DIV
19:16	R/W	0x0	TRG_DB1_POINT 第二级去抖采样点数配置为 0 则关闭本级去抖
15:12	R/W	0x0	TRG_DB0_POINT 第一级去抖采样点数配置为 0 则关闭本级去抖
11:10	–	–	–
9:8	R/W	0x0	TRG_MODE 触发模式 • 0：自动触发 GPT_EN=1 启动计数 • 1：引脚上升沿触发 GPT_EN=1 且输入引脚检测到上升沿启动计数

默认值：0x00000000			通用计数器控制（General Purpose Timer Control）
位域	类型	默认值	描述
			• 2：引脚下降沿触发 GPT_EN=1 且输入引脚检测到下降沿启动计数 • 3：引脚双沿触发 GPT_EN=1 且输入引脚检测到上升或下降沿启动计数 **注：** 每次触发 CNTR_VAL 同时被清为 0，即当 TRG_MODE ≠ 0 时再来触发，会重新计数。
7:5	–	–	–
4	R/W	0x0	RUN_MODE 运行模式 • 0：单次模式 当 CNTR_VAL 计数值达到 CNTR_MAX，GPT_EN 自动清零，CNTR_VAL 保持为 0。 • 1：循环模式 当 CNTR_VAL 计数值达到 CNTR_MAX，GPT_EN 保持为 1，CNTR_VAL 重新计数。
3:2	R/W	0x0	DBG_HALT_SEL 调试暂停触发条件 • 1: CPU0_HALT • 2: CPU1_HALT
1	R/W	0x0	DBG_HALT_BYPASS 调试暂停关闭 • 0：调试模式暂停计数 • 1：一直计数
0	R/W	0x0	GPT_EN 使能控制 • 0：计数器模块停止工作 • 1：计数器模块正常工作。 本使能字段仅控制计数器是否正常工作，其操作不会复位本模块的其它配置。

6.4.4.2. 0x004 GPT_CLK

默认值：0x00000000			通用计数器时钟（General Purpose Timer Clock）
位域	类型	默认值	描述
31:19	–	–	–
18:0	R/W	0x0	GPT_CLK_DIV 通用计数器时钟除频系数 配置为 n 则为 n+1。例如，当最大输入时钟为 200M 时，配置为 199，则输出计数频率为 1M。

6.4.4.3. 0x008 IRQ_EN

默认值：0x00000000			中断使能（Interrupt Request Enable）
位域	类型	默认值	描述
31:30	–	–	–
29	R/W	0x0	OUT3_CMPB_IRQ_EN 输出引脚 3 达到比较值 B 中断使能 • 0：中断关闭 • 1：中断使能
28	R/W	0x0	OUT3_CMPA_IRQ_EN 输出引脚 3 达到比较值 A 中断使能 • 0：中断关闭 • 1：中断使能
27:26	–	–	–
25	R/W	0x0	OUT2_CMPB_IRQ_EN 输出引脚 2 达到比较值 B 中断使能 • 0：中断关闭 • 1：中断使能
24	R/W	0x0	OUT2_CMPA_IRQ_EN 输出引脚 2 达到比较值 A 中断使能 • 0：中断关闭 • 1：中断使能

默认值：0x00000000			中断使能（Interrupt Request Enable）
位域	类型	默认值	描述
23:22	–	–	–
21	R/W	0x0	OUT1_CMPB_IRQ_EN 输出引脚 0 达到比较值 B 中断使能 • 0：中断关闭 • 1：中断使能
20	R/W	0x0	OUT1_CMPA_IRQ_EN 输出引脚 1 达到比较值 A 中断使能 • 0：中断关闭 • 1：中断使能
19:18	–	–	–
17	R/W	0	OUT0_CMPB_IRQ_EN 输出引脚 0 达到比较值 B 中断使能 • 0：中断关闭 • 1：中断使能
16	R/W	0x0	OUT0_CMPA_IRQ_EN 输出引脚 0 达到比较值 A 中断使能 • 0：中断关闭 • 1：中断使能
1	R/W	0x0	TRG_IN_IRQ_EN 输入引脚满足触发条件中断使能 • 0：中断关闭 • 1：中断使能
0	R/W	0x0	CNTR_MAX_IRQ_EN 计数器达到最大计数值中断使能 • 0：中断关闭 • 1：中断使能

6.4.4.4. 0x00C IRQ_STA

默认值：0x00000000			中断状态（Interrupt Request Status）
位域	类型	默认值	描述
31:30	–	–	–
29	R/W1C	0x0	OUT3_CMPB_IRQ_STA 输出引脚 3 达到比较值 B 中断状态 • 0：无中断 • 1：有中断
28	R/W1C	0x0	OUT3_CMPA_IRQ_STA 输出引脚 3 达到比较值 A 中断状态 • 0：无中断 • 1：有中断
27:26	–	–	–
25	R/W1C	0x0	OUT2_CMPB_IRQ_STA 输出引脚 2 达到比较值 B 中断状态 • 0：无中断 • 1：有中断
24	R/W1C	0x0	OUT2_CMPA_IRQ_STA 输出引脚 2 达到比较值 A 中断状态 • 0：无中断 • 1：有中断
23:22	–	–	–
21	R/W1C	0x0	OUT1_CMPB_IRQ_STA 输出引脚 0 达到比较值 B 中断状态 • 0：无中断 • 1：有中断
20	R/W1C	0x0	OUT1_CMPA_IRQ_STA 输出引脚 1 达到比较值 A 中断状态 • 0：无中断 • 1：有中断

默认值：0x00000000			中断状态（Interrupt Request Status）
位域	类型	默认值	描述
19:18	–	–	–
17	R/W1C	0x0	OUT0_CMPB_IRQ_STA 输出引脚 0 达到比较值 B 中断状态 • 0：无中断 • 1：有中断
16	R/W1C	0x0	OUT0_CMPA_IRQ_STA 输出引脚 0 达到比较值 A 中断状态 • 0：无中断 • 1：有中断
15:2	–	–	–
1	R/W1C	0x0	TRG_IN_IRQ_STA 输入引脚满足触发条件中断状态 • 0：无中断 • 1：有中断
0	R/W1C	0x0	CNTR_MAX_IRQ_STA 计数器达到最大计数值中断状态 • 0：无中断 • 1：有中断

6.4.4.5. 0x010 CNTR_VAL

默认值：0x00000000			计数器值 (Counter Value)
位域	类型	默认值	描述
31:0	R/W	0x00000000	CNTR_VAL 计数器值。读该寄存器返回当前计数器的值，写该寄存器更新当前计数器的值。

6.4.4.6. 0x014 CNTR_MAX

默认值：0x00000000			计数器最大值 (Counter Max)
位域	类型	默认值	描述
31:0	R/W	0x00000000	CNTR_MAX 计数器最大值 当 CNTR_VAL 计数值达到 CNTR_MAX，CNTR_VAL 自动清零。

6.4.4.7. 0x01C CNTR_CLR

默认值：0x0			计数器清零（Counter Clear）
位域	类型	默认值	描述
31:1	–	–	–
0	W1C	0x0	CNTR_CLR 计数器清零 • 0: 计数器计数 • 1: 计数器清零 **不论 GPT 处于何状态，该位写 1，计数器 CNTR_VAL 都会被清零。**

6.4.4.8. 0x040+N*0x10 (N=0~3) OUTn_CMPA

默认值：0x00000000			输出 N 比较 A 寄存器 (Output N Compare A Register）
位域	类型	默认值	描述
31:0	R/W	0x00000000	CMP_VAL 比较值

6.4.4.9. 0x044+N*0x10 (N=0~3) OUTn_CMPB

默认值：0x00000000			输出 N 比较 B 寄存器 (Output N Compare B Register）
位域	类型	默认值	描述
31:0	R/W	0x00000000	CMP_VAL 比较值

6.4.4.10. 0x04C+N*0x10 (N=0~3) OUTn_CFG

默认值：0x00000000			输出 N 配置寄存器 (Output N Configuration Register)
位域	类型	默认值	描述
31:29	–	–	–
28	R	0x0	OUTn_LEVEL 输出引脚 N 电平 • 0: 低电平 • 1: 高电平
27:14	–	–	–
13:12	R/W	0x0	OUTn_CMPB 计数值达到 CMPB 输出电平 • 0: 低电平 • 1: 高电平 • 2: 电平反转 • 3: 保持不变
11:9	–	–	–
9:8	R/W	0x0	OUTn_CMPA 计数值达到 CMPA 输出电平 • 0: 低电平 • 1: 高电平 • 2: 电平反转 • 3: 保持不变
7:5	–	–	–
4	R/W	0x0	OUTn_INIT 初始输出电平 • 0: 低电平 • 1: 高电平
3:1	–	–	–
0x0	R/W	0x0	OUTn_EN 输出使能

默认值：0x00000000			输出 N 配置寄存器 (Output N Configuration Register)
位域	类型	默认值	描述
			• 0: 引脚输出关闭 • 1: 引脚输出使能

6.4.4.11. 0x0FC GPT_VERSION

默认值：0x00000100			GPT 版本
位域	类型	默认值	描述
31:0	R/W	0x00000100	VERSION 模块版本，V1.0

7. 接口

7.1. USB Full Speed On-The-Go (USB FS OTG)

USB Full Speed On-The-Go (USB FS OTG) 模块为 USB Device 和 HOST 控制器，通过电平识别切换，实现 USB OTG 的功能，只支持 FS/LS，符合 Universal Serial Bus Specification Rev 1.1 协议。

7.1.1. 特性说明

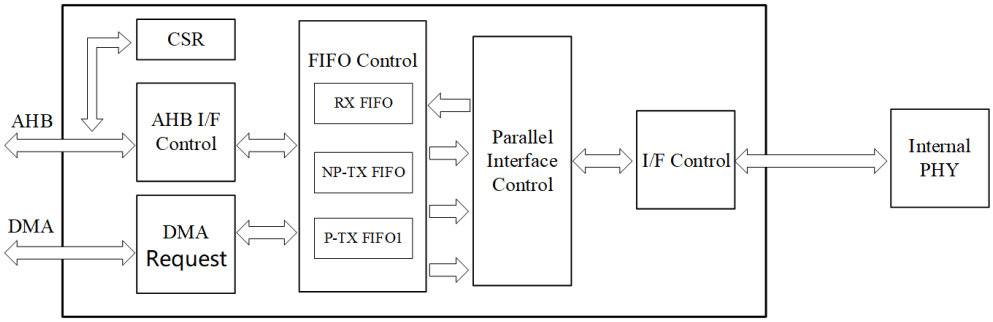
1. **Device**

   - 支持 FS/LS 两种 Device 配置
   - 1x 双向 Control EP0
   - 2x 双向 EP，用于 Bulk、Interrupt 或 Isochronous 传输类型
   - 1x Shared RXFIFO，用于所有的 OUT EP
   - 1x Non-periodic FIFO，用于所有的非周期性传输的 IN EP
   - 1x Periodic FIFO，用于周期性传输的 IN EP
   - 2K Byte FIFO 深度，支持动态调整
   - 支持外部系统 DMA

2. **Host**

   - 支持识别 FS/LS Device
   - 4x 双向 channal
   - 1x Shared RXFIFO，用于所有的 IN 传输
   - 1x Non-periodic FIFO，用于所有的非周期性传输的 OUT 传输
   - 1x Periodic FIFO，用于周期性传输的 OUT 传输
   - 2K Byte FIFO 深度，支持动态调整
   - 支持外部系统 DMA

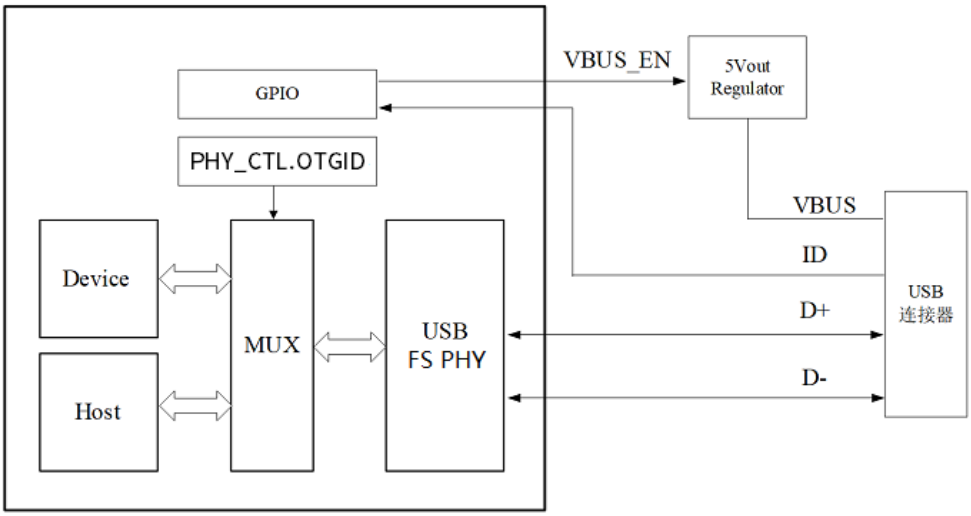
7.1.2. 原理框图



USB Device 模块结构框图

图 7–1 USB Host 控制器结构框图

7.1.3. USB FS OTG 实现方式



USB FS OTG 应用框图

切换的方式通过 GPIO 检测 USB 线缆的 ID 信号电平

1. GPIO 检测 ID 信号为高电平，则 USB FS OTG 则确定为 USB Device。
2. GPIO 检测 ID 信号为低电平，则 USB FS OTG 则确定为 USB Host，此时可以通过另一 GPIO 输出信号使能外部的 5V VBUS 电源输出，供电给连接的设备。并配置 PHY_CTL 寄存器的 OTGID 控制位为低。

7.1.4. 功能描述

7.1.4.1. PHY 接口选择

USB FS OTG 模块使用内部 PHY，且无需使用 UTMI/ULPI 接口，无需配置。

7.1.4.2. 端接电阻配置

1. 作 USB Device 模块支持作为 HS 或 LS 设备，先配置 PHY_CTL 寄存器的上拉电阻使能，然后通过 USB_DEV_SPD_SET 可进行速度配置，默认为 FS 设备。这里的配置需要在初始化完成，完成后，即可以实现相应的端接电阻配置。
2. 作 USB Host 模块，需配置 PHY_CTL 寄存器的下拉电阻使能。

7.1.4.3. 枚举地址设置

在 USB 的枚举过程，USB Host 会通过 Set Address 命令对 USB Device 的地址进行枚举，枚举地址需在 USB_DEV_ADDR 设置。对于 Host 来说，地址枚举完成后也许在 Channal 寄存器中配置相应的地址。

7.1.4.4. 中断

USB Device 模块支持中断：

- 对应的中断会有中断屏蔽控制，默认屏蔽所有中断。
  - 在 Interrupt Level 0 的 InEP 和 OutEP 中断，分别为所有 IN 和 OUT 端点的总中断状态。

    此位不能写 1 清除，如需清除此位，必须先清除所有端点产生的中断态。
  - Interrupt Level 1 是 InEP 和 OutEP 的下一层次逻辑，分别是InEP0/ InEP1/ …/ InEP4 和 OutEP0/ OutEP1/ …/ OutEP4， 表示各个端点的总中断态。默认屏蔽所有端点的中断态，通过屏蔽寄存器来使能某个端点的总中断态。

    此层的中断不能写 1 清 0，如需清除此层的中断，必须先清除下一层次的所有中断态。
  - Interrupt Level 2 是 IN/ OUT 端点的具体中断态，InEP 在此层次的中断屏蔽控制，共用一套屏蔽控制逻辑，由 INEP_INT_MSK 决定，默认屏蔽所有中断态。OutEP 在此层次的中断屏蔽控制，共用一套屏蔽控制逻辑，由 OUTEP_INT_MSK 决定，默认屏蔽所有中断态。
- 根据不同应用场景，选择是否使能中断信号。
- IN/ OUT 端点的中断信号存在层次结构的逻辑关系，具体可查看输入输出端点的终端结构图。

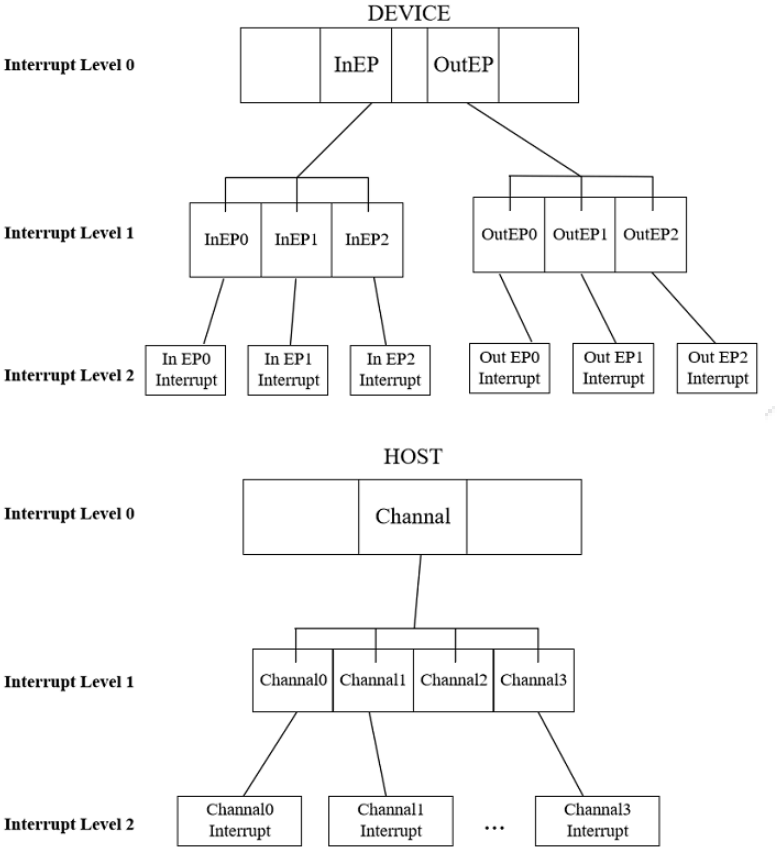


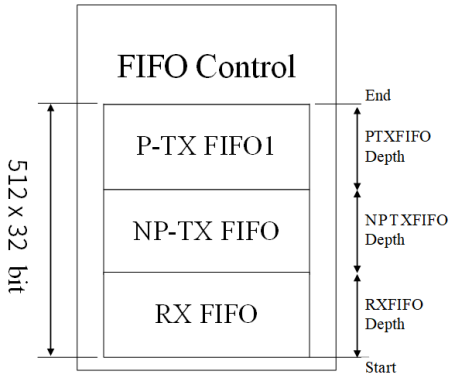
图 7–2 输入输出端点的中断结构层次

7.1.4.5. 数据 FIFO

USB Device 的数据 FIFO 总深度为 512 x 32 bits。图 7–3 : 数据 FIFO 组成如下：

- RXFIFO：FIFO 起始位 ，为所有 OUT EP 的共用 FIFO。
- NP–TXFIFO：非周期的 TXFIFO，为所有的 Control/ Bulk 类型的 Device IN EP 或 Host OUT 传输的共用 FIFO。
- P–TXFIFO：周期的 TXFIFO，Interrupt/ Isochronous 类型的 EP 或 Channal 对应 FIFO。无使用到此类型的 EP 或 Channal，可以设置该 FIFO 深度为 0。

通过 FIFO 对应的 FIFO 深度寄存器配置各个 FIFO 的大小，如果不需要，可配置为 0 深度。



数据 FIFO 组成

图 7–3 数据 FIFO 组成

7.1.4.6. Device 接收和发送控制

7.1.4.6.1. 接收控制

所有的 OUT 端点共用 RXFIFO。所有 OUT 端点的接收方式都相同，RXFIFO 的读取基地址为 BASE_USB0 + 0x1000。接收的流程如下：

1. 中断 RxFLvl 产生，指示 RXFIFO 接收了至少一个数据包。此中断状态位为只读。
2. 读取 RXFIFO_STS_POP 寄存器，将接收到的数据 POP 到端点相应的 FIFO 的起始地址，即 BASE_USB0 + 0x1000。

   其它非接收数据操作时，不允许对此寄存器进行读操作，避免造成接收数据错误。
3. 通过读取 RXFIFO_STS_POP 寄存器可知接收数据的状态、接收到数据的 byte 数量、端点号、数据 PID 类型等信息。

   如果接收数据的状态是接收到 OUT 数据包或 SETUP 数据包，可以读取相应端点的 FIFO 起始地址，每次读出 4 byte 数据，根据获取的 byte 总数决定读取的次数，每次读 FIFO 的起始地址即可。

7.1.4.6.2. 发送控制

注：

n = 1/ 2

控制类型端点的发送流程，即 IN EP0 的发送数据流程如下：

Bulk 类型端点的发送流程如下：

1. 初始化时，需对此端点的 RX_INTOKEN_EMPTY_INT 中断使能，即对应的端点收到 IN Token 后则产生中断。
2. 当 ITTXFE 中断到来，对端点相应的 FIFO 的起始地址进行写数据，写的次数取决于需要发送的数据长度。每个写操作可将 4 byte 数据写入 FIFO，每次只需对 FIFO 的起始地址 BASE_USB0 + 0x1000 * (n+1) 写即可。
3. 完成数据的写入后，对 RX_INTOKEN_EMPTY_INT 中断写 1 清除中断态，完成一次写数据。

7.1.4.6.3. 端点控制

通过寄存器 IN_EPn_CFG 以及 OUT_EPn_CFG 可对 IN/ OUT 端点进行以下的操作：

- 使能端点 EP。
- 设置 EP 的类型：Bulk/ Interrupt/ Iso。
- 在当前配置下激活端点 EP。
- 设置最大数据包大小。
- 选择 TXFIFO 的编号，针对 EP。
- 设置 Data0/ Data1 PID。
- 设置 NACK。
- 设置 STALL。
- 设置接收或发送 Transaction 时所对应的奇/偶帧。

7.1.4.7. Host 接收和发送控制

1. 通道初始化
   a. 使能相应通道的总中断
   b. 使能该次传输所需要的通道中断
   c. 设置该次传输的 PID、数据包数量、传输数据总长度
   d. 设置该次传输的最大包和传输方向
2. 发送 OUT 传输
   a. 通道初始化
   b. 使能通道
   c. 使能 TXFIFO 的空中断
   d. 当 TXFIFO 空中断到来，读取 TXFIFO_STS 状态寄存器
   e. 对通道相应的 FIFO 的起始地址写数据，每次写 4Byte，直到写完配置的数据传输总长度
   f. 完成写数据后屏蔽 TXFIFO 的空中断
3. 接收 IN 传输
   a. 通道初始化
   b. 使能通道
   c. 等待 RxFLvl 中断到来后，屏蔽 RxFLvl 中断
   d. 读取 RXFIFO_STS_POP 寄存器，将接收到的数据 POP 到端点相应的 FIFO 的起始地址（BASE_USB + 0x1000）。因此其它非接收数据操作时，不允许对此寄存器进行读操作，避免造成接收数据的错误。
   e. 通过读取 RXFIFO_STS_POP 寄存器可知接收数据的状态、接收到数据的 Byte 数量、通道号、数据 PID 类型等信息。

f. 读取通道相应的 FIFO 的起始地址即可获取数据，每次读 4Byte，直到读完所有数据

g. 使能 RxFLvl，等待下一次的传输

h. IN 传输完成后需要软件关闭通道并等待通道关闭的中断

7.1.5. 寄存器列表

表 7–1 USB FS OTG 寄存器地址映射表

Module Name	Mapping Size	Offset
USBFSOTG	33 KB	0x0000

表 7–2 USB FS OTG 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	OTG_CTL	OTG 基本配置寄存器	0x000 OTG_CTL
0x004	USB_GLB_INIT	USB Global 初始化寄存器	0x004 USB_GLB_INIT
0x008	USB_PHY_IF	USB PHY 接口寄存器	0x008 USB_PHY_IF
0x010	USB_INT_STS	USB 中断状态寄存器	0x010 USB_INT_STS
0x014	USB_INT_MSK	USB 中断屏蔽寄存器	0x014 USB_INT_MSK
0x018	RXFIFO_SIZ	RXFIFO SIZE 寄存器	0x018 RXFIFO_SIZ
0x01C	RXFIFO_STS_POP	RXFIFO 状态寄存器	0x01C RXFIFO_STS_POP
0x020	NPTXFIFO_SIZ	非周期性的 TXFIFO Size 寄存器	0x020 NPTXFIFO_SIZ
0x024	NPTXFIFO_STS	非周期性的 TXFIFO 状态寄存器	0x024 NPTXFIFO_STS
0x028	DEV_TXFIFO_SIZ	DEV TXFIFO SIZE 寄存器	0x028 DEV_TXFIFO_SIZ
0x02C	HOST_TXFIFO_SIZ	HOST TXFIFO SIZE 寄存器	0x02C HOST_TXFIFO_SIZ
0x030	RXFIFO_STS	RXFIFO 状态寄存器	0x030 RXFIFO_STS
0x040	PHY_CLK_CTL	PHY 时钟控制寄存器	0x040 PHY_CLK_CTL
0x100	USB_HOST_CONF	USB Host 配置寄存器	0x100 USB_HOST_CONF

表 7–2 USB FS OTG 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x104	HOST_FRAME	USB Host 帧间隔寄存器	0x104 HOST_FRAME
0x108	HOST_FRAME_NUM	USB Host 帧号寄存器	0x108 HOST_FRAME_NUM
0x110	HOST_PER_TXFIFO_STA	USB Host 周期 TXFIFO 状态寄存器	0x110 HOST_PER_TXFIFO_STA
0x114	HOST_ALLCHAN_INT	USB 通道中断寄存器	0x114 HOST_ALLCHAN_INT
0x118	HOST_ALLCHAN_INT_MSK	USB 通道中断屏蔽寄存器	0x118 HOST_ALLCHAN_INT_MSK
0x120	HOST_PORT_CTL	Host 端口控制和状态寄存器	0x120 HOST_PORT_CTL
0x130	HOST_CHAN_CHARn(n=0~3)	Host 通道 n 特征寄存器	0x130 HOST_CHAN_CHARn(n=0~3)
0x150	HOST_CHANn_INT(n=0~3)	Host 通道中断寄存器	0x150 HOST_CHANn_INT(n=0~3)
0x160	HOST_CHANn_INT_MSK(n=0~3)	Host 通道中断屏蔽寄存器	0x160 HOST_CHANn_INT_MSK(n=0~3)
0x170	HOST_CHANn_TXSIZ(n=0~3)	Host 通道大小控制寄存器	0x170 HOST_CHANn_TXSIZ(n=0~3)
0x200	USB_DEV_CONF	USB Device 配置寄存器	0x200 USB_DEV_CONF
0x204	USB_DEV_FUNC	USB Device 功能寄存器	0x204 USB_DEV_FUNC
0x208	USB_LINE_STS	USB Line 状态寄存器	0x208 USB_LINE_STS
0x20C	INEP_INT_MSK	输入端点中断屏蔽寄存器	0x20C INEP_INT_MSK
0x210	OUTEP_INT_MSK	输出端点中断屏蔽寄存器	0x210 OUTEP_INT_MSK
0x214	USB_EP_INT	USB 端点中断寄存器	0x214 USB_EP_INT
0x218	USB_EP_INT_MSK	USB 端点中断屏蔽寄存器	0x218 USB_EP_INT_MSK
0x220	IN_EP0_CFG	输入 EP0 配置寄存器	0x220 IN_EP0_CFG
0x0224	IN_EP1_CFG	IN EP1 配置寄存器	0x0224 IN_EP1_CFG
0x0228	IN_EP2_CFG	IN EP2 配置寄存器	0x0228 IN_EP2_CFG
0x240	OUT_EP0_CFG	OUT EP0 配置寄存器	0x240 OUT_EP0_CFG

表 7–2 USB FS OTG 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x244	OUT_EP1_CFG	OUT EP1 配置寄存器	0x244 OUT_EP1_CFG
0x248	OUT_EP2_CFG	OUT EP2 配置寄存器	0x248 OUT_EP2_CFG
0x260	IN_EP0_INT	IN EP0 中断状态寄存器	0x260 IN_EP0_INT
0x264	IN_EP1_INT	IN EP1 中断状态寄存器	0x264 IN_EP1_INT
0x268	IN_EP2_INT	IN EP2 中断状态寄存器	0x268 IN_EP2_INT
0x280	OUT_EP0_INT	OUT EP0 中断状态寄存器	0x280 OUT_EP0_INT
0x284	OUT_EP1_INT	OUT EP1 中断状态寄存器	0x284 OUT_EP1_INT
0x288	OUT_EP2_INT	OUT EP2 中断状态寄存器	0x288 OUT_EP2_INT
0x2A0	IN_EP0_TSF_SIZ	IN EP0 传输 Size 寄存器	0x2A0 IN_EP0_TSF_SIZ
0x2A4	IN_EP1_TSF_SIZ	IN EP1 传输 Size 寄存器	0x2A4 IN_EP1_TSF_SIZ
0x2A8	IN_EP2_TSF_SIZ	IN EP2 传输 Size 寄存器	0x2A8 IN_EP2_TSF_SIZ
0x2C0	OUT_EP0_TSF_SIZ	OUT EP0 传输 Size 寄存器	0x2C0 OUT_EP0_TSF_SIZ
0x2C4	OUT_EP1_TSF_SIZ	OUT EP1 传输 Size 寄存器	0x2C4 OUT_EP1_TSF_SIZ
0x2C8	OUT_EP2_TSF_SIZ	OUT EP2 传输 Size 寄存器	0x2C8 OUT_EP2_TSF_SIZ
0x340+n*0x4(n=1~2)	IN_EPn_TXFIFO_STA	IN EPn TXFIFO 状态寄存器	0x340+n*0x4(n=1~2) IN_EPn_TXFIFO_STA
0x360+(n−1)*0x4(n=1~2)	TKN_QUEUEn	TKN_QUEUEn 寄存器	0x360+(n−1)*0x4(n=1~2) TKN_QUEUEn
0xffc	USB_VERSION	USB FSOTG 版本寄存器	0xffc USB_VERSION
0x8000	USB_PHY_CTL	USB PHY 控制寄存器	0x8000 USB_PHY_CTL

7.1.6. 寄存器描述

7.1.6.1. 0x000 OTG_CTL

默认值：0x040D0000			OTG 基本配置寄存器 (OTG Contrl)
位域	类型	默认值	描述
31	R/W	0x0	保持默认值，不要改动
30	R/W	0x0	保持默认值，不要改动
29	R/W	0x0	保持默认值，不要改动
28	R/W	0x0	ASSB AHB Support Single Burst，AHB 支持 Single Burst 0x0：不支持，数据传输采用 INCR Burst 传输 0x1：支持，数据传输采用 Single Burst 传输
27	R/W	0x0	保持默认值，不要改动
26	R	0x1	AHB Master Idle，指示 AHB Master State Machine 处于 Idle 状态。
25:22	–	–	–
21	R	0x0	当前工作模式 0x0:Device 模式 0x1:Host 模式
20	R/W	0x0	OTG 版本 0x0:OTG 版本为 1.3 0x1:OTG 版本为 2.0
19	R	0x1	B–Session 有效标志，只用于 Device 模式，可用于判断 Device 是否连接 0x0:B–Session 无效 0x1:B–Session 有效
18	R	0x1	A–Session 有效标志，只用于 Host 模式 0x0:A–Session 无效 0x1:A–Session 有效
17	R	0x0	防抖时间设置，只用于 Host 模式 0x0：长防抖时间，用于物理连接(100 ms + 2.5 μs)

默认值：0x040D0000			OTG 基本配置寄存器 (OTG Contrl)
位域	类型	默认值	描述
			0x1：短防抖时间，用于软连接(2.5 μs)
16	R	0x1	连接线 ID 状态，表示模块工作模式 0x0：A–Device 模式，作为 Host 0x1：B–Device 模式，作为 Deivce
15	R/W	0x0	Bypass 防抖滤波器 0x0:关闭 0x1:使能
14:8	–	–	–
7	R/W	0x0	当 Bvalid 覆盖使能时 Bvalid 的值，只用于 Device 模式 0x0:Bvalid = 0 0x1:Bvalid = 1
6	R/W	0x0	Bvalid 覆盖使能，只用于 Device 模式 0x0:关闭 0x1:使能
5	R/W	0x0	当 Avalid 覆盖使能时 Avalid 的值，只用于 Host 模式 0x0:Avalid = 0 0x1:Avalid = 1
4	R/W	0x0	Avalid 覆盖使能，只用于 Host 模式 0x0:关闭 0x1:使能
3	R/W	0x0	当 VBUS valid 覆盖使能时 VBUS valid 的值，只用于 Host 模式 0x0:VBUS valid = 0 0x1:VBUS valid = 1
2	R/W	0x0	VBUS valid 覆盖使能，只用于 Host 模式 0x0:关闭 0x1:使能
1:0	–	–	–

7.1.6.2. 0x004 USB_GLB_INIT

默认值：0x00000000			USB Global 初始化寄存器 (USB Global Initial)
位域	类型	默认值	描述
31:20	–	–	–
19	R/W1C	0x0	Host Frame Counter Reset Host 帧计数器重置 写入此位以重置核心内的（微）帧编号计数器。当（微）帧计数器被重置时， 随后由核心发送的 SOF 具有 0 的（微）帧编号。 如果将 1 写入该位，可能无法读取回该值，因为它会在几个时钟周期内被核心清除。
18	R/W	0x0	强制选择 Device 模式 0x0:正常模式 0x1:强制选择 Device 模式
17	R/W	0x0	强制选择 Host 模式 0x0:正常模式 0x1:强制选择 Host 模式
16	R/W	0x0	PER_TFTH Periodic TxFIFO Threshold，周期的 TXFIFO 的阈值指示控制,只用于 Host 模式 0x0：周期 TXFIFO 半空则 PTXFIFO 空中断位生效 0x1：周期 TXFIFO 全空则 PTXFIFO 空中断位生效
15:11	–	–	–
10	R/W	0x0	NP_TFTH Non-Periodic TxFIFO Threshold，非周期的 TXFIFO 的阈值指示控制 0x0：非周期 TXFIFO 半空则 NPTXFIFO 空中断位生效 0x1：非周期 TXFIFO 全空则 NPTXFIFO 空中断位生效
9	R/W	0x0	GIE Global Interrupt Enable，全局中断使能位，使能 USB IP 的中断。 0x0：未使能 0x1：使能全局中断
8	R/WAC	0x0	C_IN_TKN_Q Clear IN Token Queue，清除 IN Token 的队列，写 1 清除队列，完成 自动清 0。
7:3	R/W	0x0	C_TX_NUM

默认值：0x00000000			USB Global 初始化寄存器 (USB Global Initial)
位域	类型	默认值	描述
			Clear TXFIFO Number，通过此域可以对 FIFO 进行清除。在 TxFFlsh bit 没有清 0 之前，此域不能及进行更改。 0x0：Non-periodic TXFIFO flush 0x1：Periodic TXFIFO 1 flush 0x2：Periodic TXFIFO 2 flush 0x10：Flush all TXFIFO Others：无效
2	R/WAC	0x0	C_TXFIFO Clear TXFIFO，此位写 1 进行 TXFIFO 清除（进行此操作前需确保 USB 没有进行数据传输），完成 TXFIFO 清除后自清 0。
1	R/WAC	0x0	C_RXFIFO Clear RxFIFO，此位写 1 进行 RxFIFO 清除（进行此操作前需确保 USB 没有进行数据传输），完成 RxFIFO 清除后自清 0。
0	R/WAC	0x0	Soft_RST Soft Reset，USB Device 软复位，此位写 1 进行软复位，复位完成自清 0。

7.1.6.3. 0x008 USB_PHY_IF

默认值：0x00001400			USB PHY 接口寄存器 (USB PHY Interface)
位域	类型	默认值	描述
31:6	–	–	–
5	R/W	0x0	选择单向或双向的 USB 1.1 全速串行收发器接口 0x0：6 引脚单向全速串行接口 0x1：3 引脚双向全速串行接口
4:3	–	–	–
2:0	R/W	0x0	TIMEOUT_CAL Timeout Calibration，USB Timeout 校准

7.1.6.4. 0x010 USB_INT_STS

默认值：0x04000020			USB 中断状态寄存器 (USB Interrupt Status)
位域	类型	默认值	描述
31	R/W1C	0x0	WAKEUP_INT Wake up Interrupt，唤醒产生中断，在 Suspend 状态下，Host 发出 Resume 进行唤醒。 0x0：无唤醒中断 0x1：发生唤醒中断
30	–	–	–
29	R/W1C	0x0	DisconnInt 指示一个设备断开连接，只用于 Host 模式
28	R/W1C	0x0	ConIDStsChng 指示一个设备连接 ID 状态发生变化
27	–	–	–
26	R	0x1	PER_TXFIFO_EMP periodic TXFIFO Empty，此位用于指示 periodic TxFIFO 半空或者全空(这里通过 PER_TFTH 决定) 0x0：PER TXFIFO 仍有数据需要传输 0x1：PER TXFIFO 全空或半空
25	R	0x0	HOST_CHAN_INT Host Channels Interrupt，指示存在 Channel 中断。此中断为 Channel 的总中断，需将各个 Channel 的中断清除才可以清除该中断。
24	R	0x0	HOST_PORT_INT 指示 Host 的端口发生变化，读取端口控制和状态 (HPRT) 寄存器以确定导致此中断的确切事件。须在端口控制和状态寄存器中清除相应的状态位以清除此位。
23:22	–	–	–
21	R/W1C	0x0	INCOMP_ISO_OUT_INT Incomplete Isochronous OUT Transfer Interrupt，此位中断用于指示至少一个 Isochronous OUT EP 当前帧里的 Transfer 不完整。
20	R/W1C	0x0	INCOMP_ISO_IN_INT Incomplete Isochronous IN Transfer Interrupt，此位中断用于

默认值：0x04000020			USB 中断状态寄存器 (USB Interrupt Status)
位域	类型	默认值	描述
			指示至少一个 Isochronous IN EP 当前帧里的 Transfer 不完整。
19	R	0x0	OUT_EP_INT OUT Endpoints Interrupt，指示存在 OUT EP 中断。此中断为 OUT EP 的总中断，需将各个 OUT EP 的中断清除才可以清除该中断。
18	R	0x0	IN_EP_INT IN Endpoints Interrupt，指示存在 IN EP 中断。此中断为 IN EP 的总中断，需将各个 IN EP 的中断清除才可以清除该中断。
17	R/W1C	0x0	EP_MIS_INT Endpoint Mismatch Interrupt，指示 Non-periodic TxFIFO 的数据与 IN Endpoint 不一致。 0x0：无中断产生 0x1：中断产生
16	–	–	–
15	R/W1C	0x0	EOP_FR_INT End of Periodic Frame Interrupt，此中断指示当前 Frame 达到 SET_PER_FR_INT 所设置的时间点。 0x0：无中断产生 0x1：中断产生
14	R/W1C	0x0	ISO_OUT_DROP_INT Isochronous OUT Packet Dropped Interrupt，此中断表示由于 RXFIFO 无足够空间存放 isochronous OUT EP 的一个 maximum size 的数据包，拒绝接收 isochronous OUT packet 0x0：无中断产生 0x1：中断产生
13	R/W1C	0x0	Enum_DONE Enumeration Done，此中断用于指示完成 USB 的速率枚举。 0x0：未完成速率枚举 0x1：完成速率枚举
12	R/W1C	0x0	USB_REST USB Reset，此中断用于指示 USB BUS 出现了 RESET 信号。 0x0：未出现 RESET

默认值：0x04000020			USB 中断状态寄存器 (USB Interrupt Status)
位域	类型	默认值	描述
			0x1：出现 RESET
11	R/W1C	0x0	USB_SUS USB Suspend，此中断用于指示 USB 进入 Suspend 状态。 0x0：未出现 Suspend 0x1：出现 Suspend
10	R/W1C	0x0	EARLY_SUS Early Suspend，此中断用于指示 USB 进入 idle 状态超过 3 ms 0x0：未出现 Early Suspend 0x1：出现 Early Suspend
9:8	–	–	–
7	R	0x0	OUT_NACK_EFF OUT Endpoint NAK Effective，此中断用于指示寄存器 SET_OUT_NACK 配置生效。此位清 0 可以通过设置 CLR_OUT_NACK。
6	R	0x0	IN_NACK_EFF IN Endpoint NAK Effective，此状态位用于指示寄存器 SET_NP_IN_NACK 配置生效。此位清 0 可以通过设置 CLR_NP_IN_NACK。
5	R	0x1	NP_TXFIFO_EMP Non-periodic TXFIFO Empty，此位用于指示 Non-periodic TxFIFO 半空或者全空(这里通过 NP_TFTH 决定) 0x0：NP TXFIFO 仍有数据需要传输 0x1：NP TXFIFO 全空或半空
4	R	0x0	RXFIFO_NO_EMP RXFIFO Non-Empty，此位指示 RxFIFO 里至少有一个 packet 需要读取 0x0：RXFIFO 空 0x1：RXFIFO 非空
3	R/W1C	0x0	RX_SOF Receive Start of (micro)Frame Token，此位指示 USB 接收到 SOF Token。 0x0：未接收到 SOF Token 0x1：接收到 SOF Token

默认值：0x04000020			USB 中断状态寄存器 (USB Interrupt Status)
位域	类型	默认值	描述
2	R/W1C	0x0	A–Device 超时中断 表示 Host 等待 Device 连接超时
1	R/W1C	0x0	模式冲突中断 指示 Host 模式下操作 Device 寄存器或 Device 模式下操作 Host 寄存器
0	R	0x0	表示当前工作模式 0x0:Device 模式 0x1:Host 模式

7.1.6.5. 0x014 USB_INT_MSK

默认值：0x00000000			USB 中断屏蔽寄存器 (USB Interrupt Mask)
位域	类型	默认值	描述
31	R/W	0x0	WAKEUP_INT_MSK Wake up Interrupt，唤醒产生中断，在 Suspend 状态下，Host 发出 Resume 进行唤醒。 0x0：屏蔽中断 0x1：不屏蔽中断
30	–	–	–
29	R/W	0x0	DisconnInt_MSK 指示一个设备断开连接，只用于 Host 模式 0x0：屏蔽中断 0x1：不屏蔽中断
28	R/W	0x0	ConIDStsChng_MSK 指示一个设备连接 ID 状态发生变化 0x0：屏蔽中断 0x1：不屏蔽中断
27	–	–	–
26	R/W	0x0	PER_TXFIFO_EMP_MSK periodic TXFIFO Empty，此位用于指示 periodic TxFIFO 半空或者全空(这里通过 PER_TFTH 决定)

默认值：0x00000000			USB 中断屏蔽寄存器 (USB Interrupt Mask)
位域	类型	默认值	描述
			0x0：屏蔽中断 0x1：不屏蔽中断
25	R/W	0x0	HOST_CHAN_INT_MSK Host Channels Interrupt，指示存在 Channel 中断。此中断为 Channel 的总中断，需将各个 Channel 的中断清除才可以清除该中断。 0x0：屏蔽中断 0x1：不屏蔽中断
24	R/W	0x0	HOST_PORT_INT_MSK 指示 Host 的端口发生变化，读取端口控制和状态 (HPRT) 寄存器以确定导致此中断的确切事件。须在端口控制和状态寄存器中清除相应的状态位以清除此位。 0x0：屏蔽中断 0x1：不屏蔽中断
23:22	–	–	–
21	R/W	0x0	INCOMP_ISO_OUT_INT_MSK Incomplete Isochronous OUT Transfer Interrupt，此位中断用于指示至少一个 Isochronous OUT EP 当前帧里的 Transfer 不完整。 0x0：屏蔽中断 0x1：不屏蔽中断
20	R/W	0x0	INCOMP_ISO_IN_INT_MSK Incomplete Isochronous IN Transfer Interrupt，此位中断用于指示至少一个 Isochronous IN EP 当前帧里的 Transfer 不完整。 0x0：屏蔽中断 0x1：不屏蔽中断
19	R/W	0x0	OUT_EP_INT_MSK OUT Endpoints Interrupt，指示存在 OUT EP 中断。此中断为 OUT EP 的总中断 0x0：屏蔽中断 0x1：不屏蔽中断
18	R/W	0x0	IN_EP_INT_MSK IN Endpoints Interrupt，指示存在 IN EP 中断。此中断为 IN EP 的总中断 0x0：屏蔽中断 0x1：不屏蔽中断

默认值：0x00000000			USB 中断屏蔽寄存器 (USB Interrupt Mask)
位域	类型	默认值	描述
17	R/W	0x0	EP_MIS_INT_MSK Endpoint Mismatch Interrupt，指示 Non-periodic TxFIFO 的数据与 IN Endpoint 不一致。 0x0：屏蔽中断 0x1：不屏蔽中断
16	–	–	–
15	R/W	0x0	EOP_FR_INT_MSK End of Periodic Frame Interrupt，此中断指示当前 Frame 达到 SET_PER_FR_INT 所设置的时间点。 0x0：屏蔽中断 0x1：不屏蔽中断
14	R/W	0x0	ISO_OUT_DROP_INT_MSK Isochronous OUT Packet Dropped Interrupt，此中断表示由于 RXFIFO 无足够空间存放 isochronous OUT EP 的一个 maximum size 的数据包，拒绝接收 isochronous OUT packet 0x0：屏蔽中断 0x1：不屏蔽中断
13	R/W	0x0	Enum_DONE_MSK Enumeration Done，此中断用于指示完成 USB 的速率枚举。 0x0：屏蔽中断 0x1：不屏蔽中断
12	R/W	0x0	USB_REST_MSK USB Reset，此中断用于指示 USB BUS 出现了 RESET 信号。 0x0：屏蔽中断 0x1：不屏蔽中断
11	R/W	0x0	USB_SUS_MSK USB Suspend，此中断用于指示 USB 进入 Suspend 状态。 0x0：屏蔽中断 0x1：不屏蔽中断
10	R/W	0x0	EARLY_SUS_MSK Early Suspend，此中断用于指示 USB 进入 idle 状态超过 3 ms

默认值：0x00000000			USB 中断屏蔽寄存器 (USB Interrupt Mask)
位域	类型	默认值	描述
			0x0：屏蔽中断 0x1：不屏蔽中断
9:8	–	–	–
7	R/W	0x0	OUT_NACK_EFF_MSK OUT Endpoint NAK Effective，此中断用于指示寄存器 SET_OUT_NACK 配置生效。 0x0：屏蔽中断 0x1：不屏蔽中断
6	R/W	0x0	IN_NACK_EFF_MSK IN Endpoint NAK Effective，此状态位用于指示寄存器 SET_NP_IN_NACK 配置生效。 0x0：屏蔽中断 0x1：不屏蔽中断
5	R/W	0x0	NP_TXFIFO_EMP_MSK Non-periodic TXFIFO Empty，此位用于指示 Non-periodic TxFIFO 半空或者全空 0x0：屏蔽中断 0x1：不屏蔽中断
4	R/W	0x0	RXFIFO_NO_EMP_MSK RXFIFO Non-Empty，此位指示 RxFIFO 里至少有一个 packet 需要读取 0x0：屏蔽中断 0x1：不屏蔽中断
3	R/W	0x0	RX_SOF_MSK Receive Start of (micro)Frame Token，此位指示 USB 接收到 SOF Token。 0x0：屏蔽中断 0x1：不屏蔽中断
2	–	–	–
1	R/W	0x0	模式冲突中断 指示 Host 模式下操作 Device 寄存器或 Device 模式下操作 Host 寄存器 0x0：屏蔽中断 0x1：不屏蔽中断

默认值：0x00000000			USB 中断屏蔽寄存器 (USB Interrupt Mask)
位域	类型	默认值	描述
0	–	–	–

7.1.6.6. 0x018 RXFIFO_SIZ

默认值：0x00000200			RXFIFO SIZE 寄存器 (RXFIFO SIZE)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x200	RXFIFO_SIZ RXFIFO Size，RxFIFO 空间大小，最小值 16，最大值 512

7.1.6.7. 0x01C RXFIFO_STS_POP

- Host 模式下

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status Pop)
位域	类型	默认值	描述
31:21	–	–	–
20:17	R	0x0	PKT_STS Received Packet Status ，此域用于指示接收的 packet 的状态。 0x2：收到 IN data packet 0x3：IN 传输完成(触发中断) 0x5：数据切换错误(触发中断) 0x7：通道停止(触发中断) Others：Reserved
16:15	R	0x0	DAT_PID Data PID，此域指示接收到的 packet 的 DATA PID 类型。 0x0：DATA0 0x1：DATA2 0x2：DATA1 0x3：MDATA
14:4	R	0x0	BYTE_CNT

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status Pop)
位域	类型	默认值	描述
			Byte Count，此域用于指示接收到的 data packet 的 Byte 计数
3:0	R	0x0	CHAN_NUM Channel Number，此域指示当前接收的 packet 的 Endpoint Number。

• Device 模式下

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status Pop)
位域	类型	默认值	描述
31:25	–	–	–
24:21	R	0x0	FR_NUM Frame Number，此域为 Frame Number 的低 4 位，只用于 Isochronous 类型的 OUT EPs。
20:17	R	0x0	RX_PKT_STS Received Packet Status ，此域用于指示接收的 packet 的状态。 0x1：OUT Token 回复 NACK (触发中断) 0x2：OUT Token 接收到数据包，软件可读取 RXFIFO 数据 0x3：OUT Token 传输完成(触发 OUT_TX_COMP_INT 中断) 0x4：SETUP Token 传输完成(触发 SETUP_TX_COMP _INT 中断) 0x6：SETUP Token 接收到数据包，软件可读取 RXFIFO 数据 Others：Reserved
16:15	R	0x0	OUT_DAT_PID Data PID，此域指示接收到的 OUT data packet 的 DATA PID 类型。 0x0：DATA0 0x1：DATA2 0x2：DATA1 0x3：MDATA
14:4	R	0x0	RXBYTE_CNT Byte Count，此域用于指示接收到的 data packet 的 Byte 计数
3:0	R	0x0	OUTEP_NUM Endpoint Number，此域指示当前接收的 packet 的 Endpoint Number。

7.1.6.8. 0x020 NPTXFIFO_SIZ

默认值：0x02000200			非周期性的 TXFIFO Size 寄存器 (NPTXFIFO SIZE)
位域	类型	默认值	描述
31:16	R/W	0x200	NPTXFIFO_SIZ Non-periodic TXFIFO Size，最小值 16，最大值 512
15:0	R/W	0x200	NPTXFIFO_START Non-periodic TXFIFO Start Address，非周期的 TxFIFO 的起始地址。

7.1.6.9. 0x024 NPTXFIFO_STS

默认值：0x00080200			非周期性的 TXFIFO 状态寄存器 (NPTXFIFO Status)
位域	类型	默认值	描述
31:24	R	0x0	NPTxQTop NPTXFIFO_QUE Non-periodic TXFIFO Request Queue，指示非周期发送请求队列的顶端位置，也就是正在处理的队列位置。 Bits [30:27]: 端点号/通道号 Bits [26:25]: – 0x0: IN/OUT token – 0x1: Zero-length transmit packet – 0x2/0x3: Reserved Bit [24]: Terminate (last Entry for selected Endpoint)
23:20	–	–	–
19:16	R	0x8	NP_QUE_LOC_AVA Non-periodic Transmit Request Queue Location Available， 指示目前非周期发送请求队列的可用位置 0x0：没有空闲位置 0x1：1 个位置空闲 … 0x8：8 个位置空闲 others：reserved

默认值：0x00080200			非周期性的 TXFIFO 状态寄存器 (NPTXFIFO Status)
位域	类型	默认值	描述
15:0	R	0x200	NP_TXFIFO_AVA Non-periodic TxFIFO Space Available，指示 Non-periodic TxFIFO 空间可用情况。 0x0：Non-periodic TxFIFO 满 0x1：1 word 空闲 … 0x400：1024 words 空闲 others：reserved

7.1.6.10. 0x028 DEV_TXFIFO_SIZ

默认值：0x02000400			DEV TXFIFO SIZE 寄存器 (TXFIFO1 Size)
位域	类型	默认值	描述
31:16	R/W	0x200	PTXFIFO_SIZ Periodic TxFIFO Size，周期性的 TXFIFO 大小，最小值 4。
15:0	R/W	0x400	PTXFIFO_START Periodic TxFIFO Start Address，周期性的 TXFIFO 的起始地址。

7.1.6.11. 0x02C HOST_TXFIFO_SIZ

默认值：0x03000B00			HOST TXFIFO SIZE 寄存器 (TXFIFO Size)
位域	类型	默认值	描述
31:16	R/W	0x300	PTXFIFO_SIZ Periodic TxFIFO Size，周期性的 TXFIFO 深度，最小值 4。
15:0	R/W	0xB00	PTXFIFO_START Periodic TxFIFO RAM Start Address，周期性的 TXFIFO 的起始地址。

7.1.6.12. 0x030 RXFIFO_STS

- Host 模式下

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status Pop)
位域	类型	默认值	描述
31:21	–	–	–
20:17	R	0x0	PKT_STS Received Packet Status ，此域用于指示接收的 packet 的状态。 0x2：收到 IN data packet 0x3：IN 传输完成(触发中断) 0x5：数据切换错误(触发中断) 0x7：通道停止(触发中断) Others：Reserved
16:15	R	0x0	DAT_PID Data PID，此域指示接收到的 packet 的 DATA PID 类型。 0x0：DATA0 0x1：DATA2 0x2：DATA1 0x3：MDATA
14:4	R	0x0	BYTE_CNT Byte Count，此域用于指示接收到的 data packet 的 Byte 计数
3:0	R	0x0	CHAN_NUM Channel Number，此域指示当前接收的 packet 的 Endpoint Number。

• Device 模式下

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status)
位域	类型	默认值	描述
31:25	–	–	–
24:21	R	0x0	FR_NUM Frame Number，此域为 Frame Number 的低 4 位，只用于 Isochronous 类型的 OUT EPs。
20:17	R	0x0	RX_PKT_STS Received Packet Status ，此域用于指示接收的 packet 的状态。 0x1：OUT Token 回复 NACK (触发中断) 0x2：OUT Token 接收到数据包，软件可读取 RXFIFO 数据

默认值：0x00000000			RXFIFO 状态寄存器 (RXFIFO Status)
位域	类型	默认值	描述
			0x3：OUT Token 传输完成(触发 OUT_TX_COMP_INT 中断) 0x4：SETUP Token 传输完成(触发 SETUP_TX_COMP _INT 中断) 0x6：SETUP Token 接收到数据包，软件可读取 RXFIFO 数据 Others：Reserved
16:15	R	0x0	OUT_DAT_PID Data PID，此域指示接收到的 OUT data packet 的 DATA PID 类型。 0x0：DATA0 0x1：DATA2 0x2：DATA1 0x3：MDATA
14:4	R	0x0	RXBYTE_CNT Byte Count，此域用于指示接收到的 data packet 的 byte 计数
3:0	R	0x0	OUTEP_NUM Endpoint Number，此域指示当前接收的 packet 的 Endpoint Number。

7.1.6.13. 0x040 PHY_CLK_CTL

默认值：0x00000000			PHY 时钟控制寄存器 (PHY Clock Control)
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	STP_PHY_CLK STOP PHY CLOCK，停止 PHY Clock，在 USB Suspend 状态下可以配置 此位，停止 PHY 的时钟，能够降低 USB PHY 的功耗。 0x0：无操作 0x1：停止 PHY 的时钟

7.1.6.14. 0x100 USB_HOST_CONF

默认值：0x00000000			USB Host 配置寄存器 (USB Host Configure)
位域	类型	默认值	描述
31	R/W	0x0	模式改变就绪计时器使能 (ModeChTimEn)

默认值：0x00000000			USB Host 配置寄存器 (USB Host Configure)
位域	类型	默认值	描述
			该位用于控制器在暂停或 LPM 后是否等待 200 个 PHY 时钟周期，再将 opmode 信号更改为 2’b00。 0x0：Host 恢复结束时等待 200 个 PHY 时钟周期或 SE0 线路状态，将 opmode 从 2’b10 更改为 2’b00 0x1：Host 在需恢复结束时等待 SE0 线路状态后，将 opmode 从 2’b10 更改为 2’b00。
30:16	–	–	–
15:8	R/W	0x0	唤醒检测 (ResValid) 当 HCFG.Ena32kHzS 使能时，会计算 ResValid 数量的时钟周期以检测有效的恢复
7	R/W	0x0	32kHz 休眠模式使能 (HCFG.Ena32kHzS) 系统会在挂起期间将 PHY 时钟从 48 MHz 切换到 32 kHz。
6:3	–	–	–
2	R/W	0x0	只支持 FS/LS 使能 (FSLSSupp) 0x0:支持 HS/FS/LS，取决于设备的最高速度 0x1:只支持 FS/LS，即使连接的设备支持 HS
1:0	R/W	0x0	FS/LS PHY 时钟选择(FSLSPclkSel) 当处于 FS 主机模式时： 0x0：PHY 时钟运行在 30/60 MHz 0x1：PHY 时钟运行在 48 MHz 其它：保留 当处于 LS 主机模式时： 0x0：PHY 时钟运行在 30/60 MHz。当 UTMI+/ULPI PHY 低功耗模式未被选中时，使用 30/60 MHz。 0x1：PHY 时钟运行在 48 MHz。当 UTMI+ PHY 低功耗模式被选中时，使用 48 MHz（如果 PHY 在 LS 模式下提供 48 MHz 时钟）。 0x2：PHY 时钟运行在 6 MHz。 在 USB 1.1 FS 模式下，当 UTMI+ PHY 低功耗模式被选中且 PHY 在 LS 模式下提供 6 MHz 时钟时， 使用 6 MHz。如果您在 LS 模式下选择了 6 MHz 时钟，必须进行软复位。 其它：保留

7.1.6.15. 0x104 HOST_FRAME

默认值：0x000017d7			USB Host 帧间隔寄存器 (USB Host Frame Interval)
位域	类型	默认值	描述
31:17	–	–	–
16	R/W	0x0	动态重新加载控制 (HFIRRldCtrl) 0x0:HFIR 不能动态重新加载 0x1:HFIR 可以在运行时动态重新加载 在初始配置期间需要对该位进行编程，且在运行时其值不能更改。
15:0	R/W	0x17d7	帧间隔设置 (FrInt) 如果没有编程值，会根据主机配置寄存器 (FSLSPclkSel) 中指定的 PHY 时钟计算值。 在初始配置之后，请勿更改此字段的值。

7.1.6.16. 0x108 HOST_FRAME_NUM

默认值：0x00000000			USB Host 帧号寄存器 (USB Host Frame Number)
位域	类型	默认值	描述
31:16	R	0x0	当前帧剩余时间(FrRem) 以 PHY 时钟 (PHY clocks) 为单位。该字段在每个 PHY 时钟上递减。 当它达到零时，该字段将重新加载帧间隔寄存器中的值，并在 USB 上发送新的 SOF。
15:0	R	0x0	帧号(FrNum) 字段在 USB 上发送新的 SOF 时递增，当它达到 0x3FFF 时重置为 0。

7.1.6.17. 0x110 HOST_PER_TXFIFO_STA

默认值：0x00080200			USB Host 周期 TXFIFO 状态寄存器 (USB Host Periodic Transmit FIFO/Queue Status)
位域	类型	默认值	描述
31	R	0x0	奇偶（微）帧 (PTxQTop) 0x0:在偶数（微）帧中发送 0x1:在奇数（微）帧中发送
30:27	R	0x0	通道/端点号
26:25	R	0x0	类型

默认值：0x00080200			USB Host 周期 TXFIFO 状态寄存器 (USB Host Periodic Transmit FIFO/Queue Status)
位域	类型	默认值	描述
			0x0：IN/OUT 0x1：零长度数据包 0x2：CSPLIT 0x3：禁用通道命令
24	R	0x0	终止（选定通道或端点的最后一个条目）
23:16	R	0x8	周期传输请求队列可用空间 (PTxQSpcAvail) 表示可以在周期传输请求队列中写入的空闲位置数量。此队列包含 IN 和 OUT 请求。 0x0：周期传输请求队列已满 0x1：有 1 个可用位置 0x2：有 2 个可用位置 n：有 n 个可用位置 (0 ≤n ≤16) 其它：保留
15:0	R	0x200	周期传输数据 FIFO 可用空间 (PTxFSpcAvail) 表示可以在周期 TxFIFO 中写入的空闲位置数量。值以 32 位为单位。 0x0：周期 TxFIFO 已满 0x1：有 1 个可用位置 0x2：有 2 个可用位置 0xn：有 n 个可用位置 (其中 0≤n≤32,768) 0x8000：有 32,768 个可用位置

7.1.6.18. 0x114 HOST_ALLCHAN_INT

默认值：0x00000000			USB 通道中断寄存器 (USB Channels Interrupt)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	HOST_CHAN_INT(HAINT) Host Channels Interrupt BIT，每位对应一个 Channel 的中断状态 此中断为该 Channel 的总中断，需将该 Channel 的所有中断清除才可以清除该中断。 Bit 0 for Channel 0 Bit 1 for Channel 1 Bit 2 for Channel 2 Bit 3 for Channel 3

默认值：0x00000000			USB 通道中断寄存器 (USB Channels Interrupt)
位域	类型	默认值	描述
			其它 bit 无效

7.1.6.19. 0x118 HOST_ALLCHAN_INT_MSK

默认值：0x00000000			USB 通道中断屏蔽寄存器 (USB Channels Interrupt Mask)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	HOST_CHAN_INT_MSK(HAINTMsk) Host Channels Interrupt Mask BIT，每位对应一个 OUT EP 的中断屏蔽位，0 表示屏蔽，1 表示不屏蔽。 Bit 0 for Channel 0 Bit 1 for Channel 1 Bit 2 for Channel 2 Bit 3 for Channel 3 其它 bit 无效

7.1.6.20. 0x120 HOST_PORT_CTL

默认值：0x00000000			Host 端口控制和状态寄存器 (Host Port Control and Status)
位域	类型	默认值	描述
31:19	–	–	–
18:17	R	0x0	端口速度(PrtSpd) 0x0:HS 0x1:FS 0x2:LS 0x3:Reserved
16:13	R	0x0	端口测试控制(PrtTstCtl) 0x0:Test mode 关闭 0x1: Test_J mode 0x2: Test_K mode 0x3: Test_SE0_NAK mode 0x4: Test_Packet mode

默认值：0x00000000			Host 端口控制和状态寄存器 (Host Port Control and Status)
位域	类型	默认值	描述
			0x5: Test_Force_Enable 其它 bit 无效
12	R/W	0x0	端口电源控制 (PrtPwr) 0x0:电源关闭 0x1:电源打开
11:10	R	0x0	端口 line status(PrtLnSts) 指示当前 USB 数据线上的逻辑状态 Bit 10 for D+ Bit 11 for D–
9	–	–	–
8	R/W	0x0	端口复位控制 (PrtRst) 使能复位后需等待规定时间后软件上将此位清除，FS/LS 下至少需要 10 ms 0x0:未处于复位状态 0x1:处于复位状态
7	–	–	–
6	R/W	0x0	端口恢复控制(PrtSusp) 0x0:端口未驱动恢复信号 0x1:端口驱动恢复信号
5	R/W1C	0x0	端口过流变化 (PrtOvrCurrChng) 表示 bit4 有变化，该 bit 可以写 1 清零
4	R	0x0	端口过流状态(PrtOvrCurrAct) 0x0:没有过流 0x1:过流
3	R/W1C	0x0	端口开关变化 (PrtEnChng) 表示 bit2 有变化，该 bit 可以写 1 清零
2	R/WA1C	0x0	端口开关控制 (PrtEna) 指示当遇到过流、断开连接或软件写 1 清除该 bit 时，port 会关闭 无法通过写该寄存器使能端口 0x0:端口关闭

默认值：0x00000000			Host 端口控制和状态寄存器 (Host Port Control and Status)
位域	类型	默认值	描述
			0x1:端口使能
1	R/W1C	0x0	端口连接检测 (PrtConnDet) 指示检测到设备连接
0	R	0x0	端口连接状态 (PrtConnSts) 0x0:端口上没有连接设备 0x1:端口上有连接设备

7.1.6.21. 0x130 HOST_CHAN_CHARn(n=0~3)

默认值：0x00000000			Host 通道 n 特征寄存器 (Host Channel–n Characteristics Register)
位域	类型	默认值	描述
31	R/WAC	0x0	通道使能 (ChEna)
30	R/WAC	0x0	通道关闭 (ChDis)
29	R/W	0x0	奇数帧 (OddFrm) 该字段由软件设置（重置）以指示 OTG 主机必须执行奇数（微）帧中的传输。 此字段仅适用于周期性（等时和中断）事务。 0x0:偶数（微）帧 0x1:奇数（微）帧
28:22	R/W	0x0	Device 地址(DevAddr) 这个字段选择作为数据源或接收器的具体设备。
21:20	R/W	0x0	Multi Count (MC) / Error Count (EC) 这个字段表示每个微帧中必须执行的事务数量 当 HCSPLTn.SpltEna 被设置(1’b1)时，此字段表示在事务错误上要执行的立即重试次数。 此字段必须设置为至少 2’b01。 0x0: 保留 0x1: 1 个事务 0x2: 每微帧为该端点发出 2 个事务 0x3: 每微帧为该端点发出 3 个事务
19:18	R/W	0x0	EP 类型选择 (EPType)

默认值：0x00000000			Host 通道 n 特征寄存器 (Host Channel–n Characteristics Register)
位域	类型	默认值	描述
			0x0: Control 0x1: Isochronous 0x2: Bulk 0x3: Interrupt
17	R/W	0x0	低速设备 (LSpdDev) 当与低速设备通信时，软件需将该位置 1
16	–	–	–
15	R/W	0x0	EP 方向 (EPDir) 0x0:OUT 0x1:IN
14:11	R/W	0x0	EP 号 (EPNum)
10:0	R/W	0x0	最大包长度 Maximum Packet Size (MPS)

7.1.6.22. 0x150 HOST_CHANn_INT(n=0~3)

默认值：0x00000000			Host 通道中断寄存器 (Host Channel–n Interrupt Register)
位域	类型	默认值	描述
31:11	–	–	–
10	R/W1C	0x0	数据切换错误 (DataTglErr)
9	R/W1C	0x0	帧 Overrun (FrmOvrun)
8	R/W1C	0x0	乱码错误 (BblErr)
7	R/W1C	0x0	传输事务错误 (XactErr) 表示在 USB 传输过程中发生了以下错误之一： ■ CRC 校验失败 ■ 超时 ■ 位填充错误 ■ 错误的 EOP (End of Packet)
6	R/W1C	0x0	NYET 响应接收中断 (NYET)

默认值：0x00000000			Host 通道中断寄存器 (Host Channel–n Interrupt Register)
位域	类型	默认值	描述
5	R/W1C	0x0	ACK 响应接收/发送中断 (ACK)
4	R/W1C	0x0	NAK 响应接收中断 (NAK)
3	R/W1C	0x0	STALL 响应接收中断 (STALL)
2	–	–	–
1	R/W1C	0x0	通道暂停 (ChHltd) 指示传输异常，原因可能是任何 USB 事务错误、应用程序的禁用请求或是由于已完成的传输。
0	R/W1C	0x0	传输完成 (XferCompl) 表示在没有任何错误的情况下正常完成传输。

7.1.6.23. 0x160 HOST_CHANn_INT_MSK(n=0~3)

默认值：0x00000000			Host 通道中断屏蔽寄存器 (Host Channel–n Interrupt Mask Register)
位域	类型	默认值	描述(0 表示屏蔽，1 表示不屏蔽)
31:11	–	–	–
10	R/W	0x0	数据切换错误 (DataTglErrMsk)
9	R/W	0x0	帧 Overrun (FrmOvrunMsk)
8	R/W	0x0	乱码错误 (BblErrMsk)
7	R/W	0x0	传输事务错误 (XactErrMsk) 表示在 USB 传输过程中发生了以下错误之一： ■ CRC 校验失败 ■ 超时 ■ 位填充错误 ■ 错误的 EOP (End of Packet)
6	R/W	0x0	NYET 响应接收中断 (NYETMsk)
5	R/W	0x0	ACK 响应接收/发送中断 (ACKMsk)
4	R/W	0x0	NAK 响应接收中断 (NAKMsk)

默认值：0x00000000			Host 通道中断屏蔽寄存器 (Host Channel–n Interrupt Mask Register)
位域	类型	默认值	描述(0 表示屏蔽，1 表示不屏蔽)
3	R/W	0x0	STALL 响应接收中断 (STALLMsk)
2	–	–	–
1	R/W	0x0	通道暂停 (ChHltdMsk) 指示传输异常，原因可能是任何 USB 事务错误、应用程序的禁用请求或是由于已完成的传输。
0	R/W	0x0	传输完成 (XferComplMsk) 表示在没有任何错误的情况下正常完成传输。

7.1.6.24. 0x170 HOST_CHANn_TXSIZ(n=0~3)

默认值：0x00000000			Host 通道大小控制寄存器 (Host Channel–n Transfer Size Register)
位域	类型	默认值	描述
31	R/W	0x0	Ping 传输 (DoPng) 仅用于输出传输。将此字段设置为 1，指示主机执行 PING 协议。 不要为输入传输设置此位。如果为此位设置了输入传输，它将禁用通道。
30:29	R/W	0x0	PID(Pid) 设置为初始事务要使用的 PID 类型 0x0: DATA0 0x1: DATA2 0x2: DATA1 0x3: MDATA（非控制）/ SETUP (控制）
28:19	R/W	0x0	Packet Count (PktCnt) 设置要发送 (OUT) 或接收 (IN) 的预期数据包数量。 主机在每次成功传输或接收一个 OUT/IN 数据包时递减此计数器。 一旦此计数器达到零，软件将被中断以指示正常完成
18:0	R/W	0x0	Transfer Size (XferSize) 对于 OUT 传输，它表示主机在传输过程中发送的数据字节数。 对于 IN 传输，它表示应用程序为传输预留的缓冲区大小。 软件应将此字段设置为 IN 事务 (周期性和非周期性）的最大数据包大小的整数倍。

7.1.6.25. 0x200 USB_DEV_CONF

默认值：0x08200000			USB Device 配置寄存器 (USB Device Configure)
位域	类型	默认值	描述
31:23	–	–	–
22:18	R/W	0x8	IN_EP_MIS_TH IN Endpoint Mismatch Threshold，触发 IN Endpoint Mismatch 中断的阈值。
17:16	–	–	–
15	R/W	0x0	MSK_Early_SUS_ERR_INT MSK Early Suspend Generate Erratic Error Interrupt，使能 Early Suspend 产生 Erratic Error 中断。 0x0：Early suspend 中断可以触发 erratic error 0x1：Early suspend 中断不触发 erratic error
14	R/W	0x0	EN_DLY_XCVR Enables or disables delay between xcvr_sel and txvalid during device chirp 0x0：No delay between xcvr_sel and txvalid during Device chirp 0x1：Enable delay between xcvr_sel and txvalid during Device chirp
13	–	–	–
12:11	R/W	0x0	SET_PER_FR_INT SET Periodic Frame Interval，设置 Periodic Frame 结束的时间点。 0x0：80% 0x1：85% 0x2：90% 0x3：95%
10:4	R/W	0x0	USB_DEV_ADDR USB Device Address，软件在 SetAddress command 时设置的 Device 地址。
3	–	–	–
2	R/W	0x0	OUT_RX_NZL_DATA_HDL OUT Transaction Recieve None-zero-length DATA Handle，设置在

默认值：0x08200000			USB Device 配置寄存器 (USB Device Configure)
位域	类型	默认值	描述
			OUT transaction 的 Status 阶段，接收到 non zero length 数据包时的处理方式。 0x0：发送接收到的 OUT packet 到应用层，同时根据 NAK 和 STALL Bit 的配置发送握手信号 0x1：发送 STALL 握手信号，并且不会发送接收到的 OUT packet 给应用层
1:0	R/W	0x0	USB_DEV_SPD_SET USB Device Speed Set，设置 USB Device 进行枚举的 USB 速度。 0x0：High speed，高速设备 0x1：Full speed，全速设备 others：Reserved

7.1.6.26. 0x204 USB_DEV_FUNC

默认值：0x00000002			USB Device 功能寄存器 (USB Device Function)
位域	类型	默认值	描述
31:17	–	–	–
16	R/W	0x0	EN_NACK_BBL_ERR Enable NACK Babble Error，使能 NACK 自动回复 Babble Error。 0x0：未使能 0x1：使能
15	R/W	0x0	IG_FR_NUM_ISO_EP Ignore Frame Number for Isochronous Endpoints，针对同步 EP 使用。
14:12	–	–	–
11	R/W	0x0	PWR_ON_DONE POWER-ON Programming Done，指示 Wakeup 后 Contrl、Status Register 已经 Ready。 0x0：无状态指示 0x1：指示 CSR Ready
10	W	0x0	CLR_OUT_NACK

默认值：0x00000002			USB Device 功能寄存器 (USB Device Function)
位域	类型	默认值	描述
			Cleare OUT Endpoint NACK，此位写 1 清除 OUT Endpoints NACK。
9	W	0x0	SET_OUT_NACK Set OUT Endpoint NACK，此位写 1 设置 OUT NACK。软件层通过此位设置，让所用的 OUT Endpoints 发送 NACK 握手信号。此位只有在 GOUTNakEFff 清 0 后才可以置 1。
8	W	0x0	CLR_NP_IN_NACK Clear Non-periodic IN Endpoint NACK，此位写 1 清除 Non-Periodic IN NACK。
7	W	0x0	SET_NP_IN_NACK Set Non-periodic IN Endpoint NACK，此位写 1 设置 Non-periodic IN NACK。软件通过此位设置，让所用的 Non periodic IN Endpoints 发送 NACK 握手信号。此位只有在 GINNakEFff 清 0 后才可以置 1。
6:4	R/W	0x0	TEST_MOD TEST MODE Select，测试模式选择。 0x0：Test mode disabled 0x1：Test_J mode 0x2：Test_K mode 0x3：Test_SE0_NAK mode 0x4：Test_Packet mode 0x5：Test_Force_Enable others：reserved
3	R	0x0	GLB_OUT_NACK_STS Global OUT NACK Status 0x0：根据 FIFO 的状态和 NACK、STALL bit 的设置发送了握手信号 0x1：不管空间情况，不进行 RxFIFO 的数据写入。除了 SETUP transaction，发送了 NAK 握手信号。所有的 Isochronous OUT packet 都丢弃。
2	R	0x0	GLB_IN_NACK_STS Global IN Non-periodic NAK Status 0x0：根据 TXFIFO 的数据使用情况，发送了握手信号 0x1：不管 TXFIFO 的数据使用情况，Non-periodic IN Eps 发送了 NAK 握手信号
1	R/W	0x1	SET_DEV_DISCON

默认值：0x00000002			USB Device 功能寄存器 (USB Device Function)
位域	类型	默认值	描述
			SET USB Device Disconnect，设置 Device 进行 disconnect 操作 0x0：正常工作 0x1：进行 disconnect 操作
0	R/W	0x0	EN_REM_WAKUP Enable Remot Wakeup，使能 USB Devcie 唤醒 USB Host 0x0：未使能 0x1：使能

7.1.6.27. 0x208 USB_LINE_STS

默认值：0x00000002			USB Line 状态寄存器 (USB Line Status)
位域	类型	默认值	描述
31:24	–	–	–
23:22	R	0x0	DAT_LINE_STS Data Line Status，指示目前 USB 数据线的逻辑电平，1 指示高电平；0 指示低电平。 bit23：D+线上的逻辑电平 bit22：D–线上的逻辑电平
21:8	R	0x0	RX_FR_NUM Number of the Received SOF，在 high speed 时，此域表示 micro frame number；在 full or low speed 时，此域表示 frame number。
7:4	–	–	–
3	R	0x0	ERRTIC_ERR Erratic Error，此位用于报告 UMTI+任意的错误。如果由于 erratic error 产生的 early suspend，软件只能通过实现 soft disconnect 恢复。
2:1	R	0x1	USB_ENUM_SPEED USB Enumerated Speed，指示经过速度的侦测最后 Device 得出的速率。 0x0：枚举为高速设备 0x1：枚举为全速设备 Others：Reserved

默认值：0x00000002			USB Line 状态寄存器 (USB Line Status)
位域	类型	默认值	描述
0	R	0x0	DET_SUS_STS Detect Suspend Status，USB 侦测到 Suspend Condition 时，此位置 1

7.1.6.28. 0x20C INEP_INT_MSK

默认值：0x00000000			输入端点中断屏蔽寄存器 (IN EP Interrupt Mask)
位域	类型	默认值	描述
31:14	–	–	–
13	R/W	0x0	NACK_INT_MSK NACK interrupt Mask 0x0：屏蔽 0x1：不屏蔽
12:7	–	–	–
6	R/W	0x0	IN_NACK_EFF_INT_MSK IN Endpoint NAK Effective Mask 0x0：屏蔽 0x1：不屏蔽
5	R/W	0x0	INTOKEN_MIS_INT_MSK IN Token received with EP Mismatch Mask 0x0：屏蔽 0x1：不屏蔽
4	R/W	0x0	RX_INTOKEN_EMPTY_INT_MSK IN Token Received When TxFIFO Empty Mask 0x0：屏蔽 0x1：不屏蔽
3	R/W	0x0	TIME_OUT_INT_MSK Timeout Interrupt Mask (Non–isochronous endpoints) 0x0：屏蔽 0x1：不屏蔽

默认值：0x00000000			输入端点中断屏蔽寄存器 (IN EP Interrupt Mask)
位域	类型	默认值	描述
2	–	–	–
1	R/W	0x0	DIS_EP_INT_MSK Endpoint Disabled Interrupt Mask 0x0：屏蔽 0x1：不屏蔽
0	R/W	0x0	TX_COMP_INT_MSK TX Transfer Completed Interrupt Mask 0x0：屏蔽 0x1：不屏蔽

7.1.6.29. 0x210 OUTEP_INT_MSK

默认值：0x00000000			输出端点中断屏蔽寄存器 (OUT EP Interrupt Mask)
位域	类型	默认值	描述
31:15	–	–	–
14	R/W	0x0	NYET_INT_MSK NYET interrupt Mask 0x0：屏蔽 0x1：不屏蔽
13	R/W	0x0	NACK_INT_MSK NAK interrupt Mask 0x0：屏蔽 0x1：不屏蔽
12	R/W	0x0	BABBLE_ERR_INT_MSK Babble Error Interrupt Mask 0x0：屏蔽 0x1：不屏蔽
11:7	–	–	–
6	R/W	0x0	B2B_SETUP_INT_MSK Back-to-Back SETUP Packets Received Mask

默认值：0x00000000			输出端点中断屏蔽寄存器 (OUT EP Interrupt Mask)
位域	类型	默认值	描述
			0x0：屏蔽 0x1：不屏蔽
5	R/W	0x0	STS_PHASE_RX_INT_MSK Status Phase Received Mask 0x0：屏蔽 0x1：不屏蔽
4	R/W	0x0	OUT_TOKEN_EP_DIS_INT_MSK OUT Token Received when Endpoint Disabled Mask 0x0：屏蔽 0x1：不屏蔽
3	R/W	0x0	SETUP_DONE_INT_MSK SETUP Phase Done Mask. Applies to control endpoints only. 0x0：屏蔽 0x1：不屏蔽
2	–	–	–
1	R/W	0x0	DIS_EP_INT_MSK Endpoint Disabled Interrupt Mask 0x0：屏蔽 0x1：不屏蔽
0	R/W	0x0	RX_COMP_INT_MSK Transfer Completed Interrupt Mask 0x0：屏蔽 0x1：不屏蔽

7.1.6.30. 0x214 USB_EP_INT

默认值：0x00000000			USB 端点中断寄存器 (USB EP Interrupt)
位域	类型	默认值	描述
31:16	R	0x0	OUT_EP_INT OUT Endpoint Interrupt BIT，每位对应一个 OUT EP 的中断状态 此中断为该 OUT EP 的总中断，需将该 OUT EP 的所有中断清除才可以清除该中断。

默认值：0x00000000			USB 端点中断寄存器 (USB EP Interrupt)
位域	类型	默认值	描述
			Bit 16 for OUT endpoint 0 Bit 17 for OUT endpoint 1 Bit 18 for OUT endpoint 2 其它 bit 无效
15:0	R	0x0	IN_EP_INT IN Endpoint Interrupt BIT，每位对应一个 IN EP 的中断状态 此中断为该 IN EP 的总中断，需将该 IN EP 的所有中断清除才可以清除该中断。 Bit 0 for IN endpoint 0 Bit 1 for IN endpoint 1 Bit 2 for IN endpoint 2 其它 bit 无效

7.1.6.31. 0x218 USB_EP_INT_MSK

默认值：0x00000000			USB 端点中断屏蔽寄存器 (USB EP Interrupt Mask)
位域	类型	默认值	描述
31:16	R/W	0x0	OUT_EP_INT_MSK OUT Endpoint Interrupt Mask，每位对应一个 OUT EP 的中断屏蔽位 ，1b’0 表示屏蔽，1b’1 表示不屏蔽。 Bit 16 for OUT endpoint 0 Bit 17 for OUT endpoint 1 Bit 18 for OUT endpoint 2 其它 bit 无效
15:0	R/W	0x0	IN_EP_INT_MSK IN Endpoint Interrupt Mask，每位对应一个 IN EP 的中断屏蔽位 ，1b’0 表示屏蔽，1b’1 表示不屏蔽。 Bit 0 for IN endpoint 0 Bit 1 for IN endpoint 1 Bit 2 for IN endpoint 2 其它 bit 无效

7.1.6.32. 0x220 IN_EP0_CFG

默认值：0x00008000			输入 EP0 配置寄存器 (IN EP0 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 IN EP0，将此位置 1，使能 EP0 发送数据。在 EP0 的 DIS_EP_INT 中断或 TX_COMP_INT 中断出现之前，此位会自清 0。
30	R/WAC	0x0	DIS_EP Disable Endpoint，关闭 IN EP0，需要在 EN_EP=1 的情况下，才可以对此位置 1。将此位置 1，EP0 停止发送数据，尽管 EP0 的 transfer 没有完成。需要接收到 EP0 的 DIS_EP_INT 中断，才可以认为 EP0 Disable。在 EP0 disabled 中断出现之前，此位会自清 0。
29:28	–	–	–
27	W	0x0	S_NACK Set NACK，配置 NACK，此位置 1 设置 EP0 的 NACK 0x0：无操作 0x1：Control EP0 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NACK，清除 NACK，此位置 1，清除 EP0 的 NAK 位 0x0：无操作 0x1：清除 Control EP0 的 NACK 位
25:22	R/W	0x0	TXFIFO_NUM TXFIFO Number，IN EP0 对应 Non-periodic TxFIFO，因此此域一直为 0，无需配置。
21	R/WAC	0x0	EN_STALL Enable STALL Handshake，此位置 1 发送 Stall 握手信号。Device core 会在收到 SETUP Token 后会将此位清 0。另外如果 NAK 与 Stall 同时置位，发送 Stall 握手信号的优先级更高。 0x0：无操作 0x1：发送 Stall 握手信号
20	–	–	–
19:18	R	0x0	EP0_TYPE Endpoint0 Type，EP0 的类型，这里为只读位，EP0 总是为 Control 类型的

默认值：0x00008000			输入 EP0 配置寄存器 (IN EP0 Configure)
位域	类型	默认值	描述
			端点。
017	R	0x0	IN_EP0_NACK_STS IN Endpoint0 NACK Status，IN EP0 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP0 发送非 NACK 握手信号 0x1：EP0 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	–	–	–
15	R	0x1	ACT_EP Active Endpoint，对于 EP0 此位一直为 1，表示 control IN EP0 一直处于活跃的状态。
14:2	–	–	–
1:0	R/W	0x0	MPS Maximum Packet Size，IN EP0 的最大 Packet Size 0x0：64bytes 0x1：32bytes 0x2：16bytes 0x3：8bytes

7.1.6.33. 0x0224 IN_EP1_CFG

默认值：0x00000000			IN EP1 配置寄存器 (IN EP1 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 IN EP1，将此位置 1，使能 EP1 发送数据。在 EP1 的 DIS_EP_INT 中断或 TX_COMP_INT 中断出现之前，此位会自清 0。
30	R/WAC	0x0	DIS_EP Disable Endpoint，关闭 IN EP1，需要在 EN_EP=1 的情况下，才可以对此位置 1。将此位置 1，EP1 停止发送数据，尽管 EP1 的 transfer 没有完成。 需要接收到 EP1 的 DIS_EP_INT 中断，才可以认为 EP1 Disable。在 EP1

默认值：0x00000000			IN EP1 配置寄存器 (IN EP1 Configure)
位域	类型	默认值	描述
			disabled 中断出现之前，此位会自清 0。
29	W	0x0	S_DATA1 Set DATA1 PID，只适用 interrupt/bulk 类型的 IN EPs，此位置 1， PID 域选择 DATA1。 0x0：无操作 0x1：PID 域选择 DATA1
29	W	0x0	S_ISO_ODD_FR Set Isochronous Odd (micro) frame，只适用 isochronous IN EPs，此位置 1，EO_FrNum 域选择奇数帧。 0x0：无操作 0x1：EO_FrNum 域选择奇数帧
28	W	0x0	S_DATA0 Set DATA0 PID，只适用 interrupt/bulk 类型的 IN EPs，此位置 1， PID 域选择 DATA0 0x0：无操作 0x1：PID 域选择 DATA0
28	W	0x0	SS_ISO_EVEN_FR Set Isochronous Even (micro) frame，只适用 isochronous IN EPs，此位置 1，EO_FrNum 域选择偶数帧。 0x0：无操作 0x1：EO_FrNum 域选择偶数帧
27	W	0x0	S_NACK Set NACK，配置 NACK，此位置 1 设置 EP1 的 NACK 0x0：无操作 0x1：EP1 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NAK，此位置 1，清除 EP1 的 NACK 握手信号 0x0：无操作 0x1：清除 IN EP1 的 NACK 握手信号
25:22	R/W	0x0	TXFIFO_NUM TXFIFO Number，IN EP 所对应采用的 TXFIFO。

默认值：0x00000000			IN EP1 配置寄存器 (IN EP1 Configure)
位域	类型	默认值	描述
			0x0：Non-periodic TxFIFO 0x1：Specified Periodic TxFIFO number 1 0x3：Specified Periodic TxFIFO number 2 Others：Reserved
21	R/W	0x0	EN_STALL Enable STALL Handshake，此位置 1 发送 Stall 握手信号，不适用 Isochronous 类型 Endpoint。Device core 会在收到 SETUP Token 后会将此位清 0。另外如果 NACK 与 Stall 同时置位，发送 Stall 握手信号的优先级更高。 0x0：无操作 0x1：发送 Stall 握手信号
20	–	–	–
19:18	R/W	0x0	EP_TYPE_SEL Endpoint Type Select，EP 类型选择 0x0：Control 0x1：Isochronous 0x2：Bulk 0x3：Interrupt
17	R	0x0	IN_EP_NACK_STS IN Endpoint NACK Status，IN EP 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP 发送非 NAK 握手信号 0x1：EP 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	R	0x0	DATA_PID_STS Endpoint Data PID Status，当 EP 类型为 interrupt/bulk EP。 0x0：DATA0 0x1：DATA1 EO_FRNUM_STS Even/Odd Frame Number Status，当 EP 类型为 isochronous EP。 0x0：偶数帧 0x1：奇数帧

默认值：0x00000000			IN EP1 配置寄存器 (IN EP1 Configure)
位域	类型	默认值	描述
15	R/WAC	0x0	ACT_EP Active Endpoint，EP 是否处于活跃工作状态，在 USB Data Line 出现 USB RESET 命令后，此位会自动清 0。 0x0：未活跃 0x1：活跃
14:11	–	–	–
10:0	R/W	0x0	MPS Maximum Packet Size，IN EP1 的最大 Packet Size，单位是 Byte。

7.1.6.34. 0x0228 IN_EP2_CFG

默认值：0x00000000			IN EP2 配置寄存器 (IN EP2 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 IN EP2，将此位置 1，使能 EP2 发送数据。在 EP2 的 DIS_EP_INT 中断或 TX_COMP_INT 中断出现之前，此位会自清 0。
30	R/WAC	0x0	DIS_EP Disable Endpoint，关闭 IN EP2，需要在 EN_EP=1 的情况下，才可以对此位置 1。将此位置 1，EP2 停止发送数据，尽管 EP2 的 transfer 没有完成。 需要接收到 EP2 的 DIS_EP_INT 中断，才可以认为 EP2 Disable。在 EP2 disabled 中断出现之前，此位会自清 0。
29	W	0x0	S_DATA1 Set DATA1 PID，只适用 interrupt/bulk 类型的 IN EPs，此位置 1，PID 域选择 DATA1。 0x0：无操作 0x1：PID 域选择 DATA1
29	W	0x0	S_ISO_ODD_FR Set Isochronous Odd (micro) frame，只适用 isochronous IN EPs，此位置 1，EO_FrNum 域选择奇数帧。 0x0：无操作

默认值：0x00000000			IN EP2 配置寄存器 (IN EP2 Configure)
位域	类型	默认值	描述
			0x1：EO_FrNum 域选择奇数帧
28	W	0x0	S_DATA0 Set DATA0 PID，只适用 interrupt/bulk 类型的 IN EPs，此位置 1，PID 域选择 DATA0 0x0：无操作 0x1：PID 域选择 DATA0
28	W	0x0	SS_ISO_EVEN_FR Set Isochronous Even (micro) frame，只适用 isochronous IN EPs，此位置 1，EO_FrNum 域选择偶数帧。 0x0：无操作 0x1：EO_FrNum 域选择偶数帧
27	W	0x0	S_NACK Set NACK，配置 NACK，此位置 1 设置 EP2 的 NACK 0x0：无操作 0x1：EP2 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NAK，此位置 1，清除 EP2 的 NACK 握手信号 0x0：无操作 0x1：清除 IN EP2 的 NACK 握手信号
25:22	R/W	0x0	TXFIFO_NUM TXFIFO Number，IN EP 所对应采用的 TXFIFO。 0x0：Non-periodic TxFIFO 0x1：Specified Periodic TxFIFO number 1 0x3：Specified Periodic TxFIFO number 2 Others：Reserved
21	R/W	0x0	EN_STALL Enable STALL Handshake，此位置 1 发送 Stall 握手信号，不适用 Isochronous 类型 Endpoint。Device core 会在收到 SETUP Token 后会将此位清 0。另外如果 NACK 与 Stall 同时置位，发送 Stall 握手信号的优先级更高。 0x0：无操作 0x1：发送 Stall 握手信号

默认值：0x00000000			IN EP2 配置寄存器 (IN EP2 Configure)
位域	类型	默认值	描述
20	–	–	–
19:18	R/W	0x0	EP_TYPE_SEL Endpoint Type Select，EP 类型选择 0x0：Control 0x1：Isochronous 0x2：Bulk 0x3：Interrupt
17	R	0x0	IN_EP_NACK_STS IN Endpoint NACK Status，IN EP 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP 发送非 NAK 握手信号 0x1：EP 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	R	0x0	DATA_PID_STS Endpoint Data PID Status，当 EP 类型为 interrupt/bulk EP。 0x0：DATA0 0x1：DATA1 EO_FRNUM_STS Even/Odd Frame Number Status，当 EP 类型为 isochronous EP。 0x0：偶数帧 0x1：奇数帧
15	R/WAC	0x0	ACT_EP Active Endpoint，EP 是否处于活跃工作状态，在 USB Data Line 出现 USB RESET 命令后，此位会自动清 0。 0x0：未活跃 0x1：活跃
14:11	–	–	–
10:0	R/W	0x0	MPS Maximum Packet Size，IN EP2 的最大 Packet Size，单位是 Byte。

7.1.6.35. 0x240 OUT_EP0_CFG

默认值：0x00008000			OUT EP0 配置寄存器 (OUT EP0 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 OUT EP0，将此位置 1，使能 EP0 接收数据。 在 OUT EP0 的 SETUP_DONE_INT 中断、DIS_EP_INT 中断或 RX_COMP_INT 中断出现之前，此位会自清 0。
30	R	0x0	DIS_EP Disable Endpoint Disable，关闭 OUT EP0，软件无法对此位进行写操作，无法关闭 OUT EP0。
29:28	–	–	–
27	W	0x0	S_NACK Set NACK，此位置 1，设置 OUT EP0 发送 NACK 0x0：无操作 0x1：Control EP0 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NACK，此位置 1，清除 EP0 NACK 握手信号 0x0：无操作 0x1：清除 Control EP0 的 NACK 握手信号
25:22	–	–	–
21	R/WAC	0x0	EN_STALL Enable STALL Handshake，此位置 1 发送 Stall 握手信号。 0x0：无操作 0x1：发送 Stall 握手信号 Device core 会在收到 SETUP Token 后会将此位清 0。如果 NACK 与 Stall 信号同时设置，发送 Stall 握手信号的优先级更高。另外，不管此位的设置情况，Device core 只要收到 SETUP 数据包，就会回复 ACK 握手信号。
20	R/W	0x0	EN_SNOOP Enable Snoop Mode，配置此位可以使得 EP0 进入 Snoop mode。在 Snoop 模式下，USB Device 不会检查接收的 OUT packet 是否正确。 0x0：未进入 Snoop mode 0x1：进入 Snoop mode

默认值：0x00008000			OUT EP0 配置寄存器 (OUT EP0 Configure)
位域	类型	默认值	描述
19:18	R	0x0	EP0_TYPE Endpoint0 Type，EP0 一直为 Control 类型，此 bit 不需配置。
17	R	0x0	OUT_EP0_NACK_STS OUT Endpoint0 NACK Status，OUT EP0 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP0 发送非 NACK 握手信号 0x1：EP0 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	–	–	–
15	R	0x1	ACT_EP Active Endpoint，对于 EP0 此位一直为 1，表示 control OUT EP0 一直处于活跃的状态。
14:2	–	–	–
1:0	R/W	0x0	MPS Maximum Packet Size，OUT EP0 的最大 Packet Size 0x0：64bytes 0x1：32bytes 0x2：16bytes 0x3：8bytes

7.1.6.36. 0x244 OUT_EP1_CFG

默认值：0x00000000			OUT EP1 配置寄存器 (OUT EP1 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 OUT EP1，将此位置 1，使能 EP1 接收数据。在 OUT EP1 的 SETUP_DONE_INT 中断、DIS_EP_INT 中断或 RX_COMP_INT 中断出现之前，此位会自清 0。
30	R/WAC	0x0	DIS_EP Disable Endpoint Disable，关闭 OUT EP0，软件无法对此位进行写

默认值：0x00000000			OUT EP1 配置寄存器 (OUT EP1 Configure)
位域	类型	默认值	描述
			操作，无法关闭 OUT EP0。
29	W	0x0	S_DATA1 Set DATA1 PID，只适用 interrupt/bulk 类型的 OUT EPs，此位置 1， PID 域选择 DATA1。 0x0：无操作 0x1：PID 域选择 DATA1
29	W	0x0	S_ISO_ODD_FR Set Isochronous Odd (micro) frame，只适用 isochronous OUT EPs，此位置 1，EO_FrNum 域选择奇数帧。 0x0：无操作 0x1：EO_FrNum 域选择奇数帧
28	W	0x0	S_DATA0 Set DATA0 PID，只适用 interrupt/bulk 类型的 OUT EPs，此位置 1 ，PID 域选择 DATA0 0x0：无操作 0x1：PID 域选择 DATA0
28	W	0x0	S_ISO_EVEN_FR Set Isochronous Even (micro) frame，只适用 isochronous OUT EPs，此位置 1，EO_FrNum 域选择偶数帧。 0x0：无操作 0x1：EO_FrNum 域选择偶数帧
27	W	0x0	S_NACK Set NACK，配置 NACK，此位置 1 设置 EP1 的 NACK 0x0：无操作 0x1：EP1 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NAK，此位置 1，清除 EP1 的 NACK 握手信号 0x0：无操作 0x1：清除 IN EP1 的 NACK 握手信号
25:22	–	–	–
21	R/W	0x0	EN_STALL

默认值：0x00000000			OUT EP1 配置寄存器 (OUT EP1 Configure)
位域	类型	默认值	描述
			Enable STALL Handshake，此位置 1 发送 Stall 握手信号，不适用 Isochronous 类型 Endpoint。Device core 会在收到 SETUP Token 后会将此位清 0。另外如果 NACK 与 Stall 同时置位，发送 Stall 握手信号的优先级更高。 0x0：无操作 0x1：发送 Stall 握手信号
20	R/W	0x0	EN_SNOOP Enable Snoop Mode，配置此位可以使得 EP1 进入 Snoop mode。在 Snoop 模式下，USB Device 不会检查接收的 OUT packet 是否正确。 0x0：未进入 Snoop mode 0x1：进入 Snoop mode
19:18	R/W	0x0	EP_TYPE_SEL Endpoint Type Select，EP 类型选择 0x0：Control 0x1：Isochronous 0x2：Bulk 0x3：Interrupt
17	R	0x0	OUT_EP_NACK_STS OUT Endpoint NACK Status，OUT EP 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP 发送非 NAK 握手信号 0x1：EP 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	R	0x0	DATA_PID_STS Endpoint Data PID Status，适用 interrupt/bulk OUT EPs。 0x0：DATA0 0x1：DATA1
16	R	0x0	EO_FRNUM_STS Even/Odd Frame Number Status，适用 isochronous OUT EPs。 0x0：偶数帧 0x1：奇数帧
15	R/WAC	0x0	ACT_EP

默认值：0x00000000			OUT EP1 配置寄存器 (OUT EP1 Configure)
位域	类型	默认值	描述
			Active Endpoint，EP 是否处于活跃工作状态，在 USB Data Line 出现 USB RESET 命令后，此位会自动清 0。 0x0：未活跃 0x1：活跃
14:11	–	–	–
10:0	R/W	0x0	MPS Maximum Packet Size，OUT EP1 的最大 Packet Size，单位是 Byte。

7.1.6.37. 0x248 OUT_EP2_CFG

默认值：0x00000000			OUT EP2 配置寄存器 (OUT EP2 Configure)
位域	类型	默认值	描述
31	R/WAC	0x0	EN_EP Enable Endpoint，使能 OUT EP2，将此位置 1，使能 EP2 接收数据。在 OUT EP2 的 SETUP_DONE_INT 中断、DIS_EP_INT 中断或 RX_COMP_INT 中断出现之前，此位会自清 0。
30	R/WAC	0x0	DIS_EP Disable Endpoint Disable，关闭 OUT EP0，软件无法对此位进行写操作，无法关闭 OUT EP0。
29	W	0x0	S_DATA1 Set DATA1 PID，只适用 interrupt/bulk 类型的 OUT EPs，此位置 1，PID 域选择 DATA1。 0x0：无操作 0x1：PID 域选择 DATA1
29	W	0x0	S_ISO_ODD_FR Set Isochronous Odd (micro) frame，只适用 isochronous OUT EPs，此位置 1，EO_FrNum 域选择奇数帧。 0x0：无操作 0x1：EO_FrNum 域选择奇数帧
28	W	0x0	S_DATA0 Set DATA0 PID，只适用 interrupt/bulk 类型的 OUT EPs，此位置 1

默认值：0x00000000			OUT EP2 配置寄存器 (OUT EP2 Configure)
位域	类型	默认值	描述
			，PID 域选择 DATA0 0x0：无操作 0x1：PID 域选择 DATA0
28	W	0x0	S_ISO_EVEN_FR Set Isochronous Even (micro) frame，只适用 isochronous OUT EPs，此位置 1，EO_FrNum 域选择偶数帧。 0x0：无操作 0x1：EO_FrNum 域选择偶数帧
27	W	0x0	S_NACK Set NACK，配置 NACK，此位置 1 设置 EP2 的 NACK 0x0：无操作 0x1：EP2 发送 NACK 握手信号
26	W	0x0	C_NACK Clear NAK，此位置 1，清除 EP2 的 NACK 握手信号 0x0：无操作 0x1：清除 IN EP2 的 NACK 握手信号
25:22	–	–	–
21	R/W	0x0	EN_STALL Enable STALL Handshake，此位置 1 发送 Stall 握手信号，不适用 Isochronous 类型 Endpoint。Device core 会在收到 SETUP Token 后会将此位清 0。另外如果 NACK 与 Stall 同时置位，发送 Stall 握手信号的优先级更高。 0x0：无操作 0x1：发送 Stall 握手信号
20	R/W	0x0	EN_SNOOP Enable Snoop Mode，配置此位可以使得 EP2 进入 Snoop mode。在 Snoop 模式下，USB Device 不会检查接收的 OUT packet 是否正确。 0x0：未进入 Snoop mode 0x1：进入 Snoop mode
19:18	R/W	0x0	EP_TYPE_SEL Endpoint Type Select，EP 类型选择

默认值：0x00000000			OUT EP2 配置寄存器 (OUT EP2 Configure)
位域	类型	默认值	描述
			0x0：Control 0x1：Isochronous 0x2：Bulk 0x3：Interrupt
17	R	0x0	OUT_EP_NACK_STS OUT Endpoint NACK Status，OUT EP 当前的回复 NACK 的状态 0x0：根据 FIFO 的状态，EP 发送非 NAK 握手信号 0x1：EP 回复 NACK 握手信号 另外需注意，不管此状态位如何，对于 SETUP 数据包，Device 总是回复 ACK 握手信号。
16	R	0x0	DATA_PID_STS Endpoint Data PID Status，适用 interrupt/bulk OUT EPs。 0x0：DATA0 0x1：DATA1
16	R	0x0	EO_FRNUM_STS Even/Odd Frame Number Status，适用 isochronous OUT EPs。 0x0：偶数帧 0x1：奇数帧
15	R/WAC	0x0	ACT_EP Active Endpoint，EP 是否处于活跃工作状态，在 USB Data Line 出现 USB RESET 命令后，此位会自动清 0。 0x0：未活跃 0x1：活跃
14:11	–	–	–
10:0	R/W	0x0	MPS Maximum Packet Size，OUT EP2 的最大 Packet Size，单位是 Byte。

7.1.6.38. 0x260 IN_EP0_INT

默认值：0x00000080			IN EP0 中断状态寄存器 (IN EP0 Interrupt Status)
位域	类型	默认值	描述
31:14	–	–	–
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。另外，如果 Isochronous IN EP 由于 TxFIFO 数据空间不足发送了 0 长度包，同样会产生此中断。
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error Interrupt，Babble 错误中断，当 EP0 接收到 Babble Error，此位置 1。
11:8	–	–	–
7	R/W1C	0x1	TXFIFO_EMP_INT TXFIFO Empty Interrupt，EP0 对应的 TXFIFO 空中断。此中断位在 TxFIFO 出现半空或全空时置 1。TXFIFO 空中断的阈值由 NP_TFTH 决定。
6	R/W1C	0x0	IN_NACK_EFF_INT IN Endpoint NACK Effective Interrupt，此位只适用 IN EP。 在对 C_NACK 置位前，此位需要先进行写 1 清 0 此中断位表示 IN EP 的 S_NACK 设置生效 此中断位不能保证 NACK 握手信号已发送到 USB 上。 STALL 位的优先级会比 NACK 的优先级高。
5	R/W1C	0x0	INTOKEN_MIS_INT IN Token Received with EP Mismatch，只适用于采用 Non-periodic FIFO 的 IN Eps。指示 Non-periodic TxFIFO 的数据与 EP 端点不匹配。
4	R/W1C	0x0	RX_INTOKEN_EMPTY_INT IN Token Received when TxFIFO is Empty，此中断位表示 EP 在收到 IN Token 的时候，对应的 TxFIFO 为空。
3	R/W1C	0x0	TIME_OUT_INT Timeout Interrupt，只适用于 non-isochronous IN EP。此位用于指示 EP 收到上次 IN Token 后，USB 检测到超时的状态出现。
2	–	–	–

默认值：0x00000080			IN EP0 中断状态寄存器 (IN EP0 Interrupt Status)
位域	类型	默认值	描述
1	R/W1C	0x0	DIS_EP_INT Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	TX_COMP_INT TX Transfer Completed Interrupt，该中断用于指示 TX Transfer 完成。

7.1.6.39. 0x264 IN_EP1_INT

默认值：0x00000080			IN EP1 中断状态寄存器 (IN EP1 Interrupt Status)
位域	类型	默认值	描述
31:14	–	–	–
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。另外，如果 Isochronous IN EP 由于 TxFIFO 数据空间不足发送了 0 长度包，同样会产生此中断。
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error Interrupt，Babble 错误中断，当 EP1 接收到 Babble Error，此位置 1。
11:8	–	–	–
7	R/W1C	0x1	TXFIFO_EMP_INT TXFIFO Empty Interrupt，EP1 对应的 TXFIFO 空中断。此中断位在 TxFIFO 出现半空或全空时置 1。TXFIFO 空中断的阈值由 NP_TFTH 决定。
6	R/W1C	0x0	IN_NACK_EFF_INT IN Endpoint NACK Effective Interrupt，此位只适用 IN EP。 在对 C_NACK 置位前，此位需要先进行写 1 清 0 此中断位表示 IN EP 的 S_NACK 设置生效 此中断位不能保证 NACK 握手信号已发送到 USB 上。 STALL 位的优先级会比 NACK 的优先级高。
5	R/W1C	0x0	INTOKEN_MIS_INT IN Token Received with EP Mismatch，只适用于采用 Non-periodic

默认值：0x00000080			IN EP1 中断状态寄存器 (IN EP1 Interrupt Status)
位域	类型	默认值	描述
			FIFO 的 IN Eps。指示 Non-periodic TxFIFO 的数据与 EP 端点不匹配。
4	R/W1C	0x0	RX_INTOKEN_EMPTY_INT IN Token Received when TxFIFO is Empty，此中断位表示 EP 在收到 IN Token 的时候，对应的 TxFIFO 为空。
3	R/W1C	0x0	TIME_OUT_INT Timeout Interrupt，只适用于 non-isochronous IN EP。此位用于指示 EP 收到上次 IN Token 后，USB 检测到超时的状态出现。
2	–	–	–
1	R/W1C	0x0	DIS_EP_INT Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	TX_COMP_INT TX Transfer Completed Interrupt，该中断用于指示 TX Transfer 完成。

7.1.6.40. 0x268 IN_EP2_INT

默认值：0x00000080			IN EP2 中断状态寄存器 (IN EP2 Interrupt Status)
位域	类型	默认值	描述
31:14	–	–	–
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。另外，如果 Isochronous IN EP 由于 TxFIFO 数据空间不足发送了 0 长度包，同样会产生此中断。
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error Interrupt，Babble 错误中断，当 EP2 接收到 Babble Error，此位置 1。
11:8	–	–	–
7	R/W1C	0x1	TXFIFO_EMP_INT TXFIFO Empty Interrupt，EP2 对应的 TXFIFO 空中断。此中断位在

默认值：0x00000080			IN EP2 中断状态寄存器 (IN EP2 Interrupt Status)
位域	类型	默认值	描述
			TxFIFO 出现半空或全空时置 1。TXFIFO 空中断的阈值由 NP_TFTH 决定。
6	R/W1C	0x0	IN_NACK_EFF_INT IN Endpoint NACK Effective Interrupt，此位只适用 IN EP。 在对 C_NACK 置位前，此位需要先进行写 1 清 0 此中断位表示 IN EP 的 S_NACK 设置生效 此中断位不能保证 NACK 握手信号已发送到 USB 上。 STALL 位的优先级会比 NACK 的优先级高。
5	R/W1C	0x0	INTOKEN_MIS_INT IN Token Received with EP Mismatch，只适用于采用 Non-periodic FIFO 的 IN Eps。指示 Non-periodic TxFIFO 的数据与 EP 端点不匹配。
4	R/W1C	0x0	RX_INTOKEN_EMPTY_INT IN Token Received when TxFIFO is Empty，此中断位表示 EP 在收到 IN Token 的时候，对应的 TxFIFO 为空。
3	R/W1C	0x0	TIME_OUT_INT Timeout Interrupt，只适用于 non-isochronous IN EP。此位用于指示 EP 收到上次 IN Token 后，USB 检测到超时的状态出现。
2	–	–	–
1	R/W1C	0x0	DIS_EP_INT Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	TX_COMP_INT TX Transfer Completed Interrupt，该中断用于指示 TX Transfer 完成。

7.1.6.41. 0x280 OUT_EP0_INT

默认值：0x00000000			OUT EP0 中断状态寄存器 (OUT EP0 Interrupt Status)
位域	类型	默认值	描述
31:16	–	–	–
15	R/W1C	0x0	SETUP_PKT_RX_INT Setup Packet Received Interrupt，只适用于 Control OUT EP。

默认值：0x00000000			OUT EP0 中断状态寄存器 (OUT EP0 Interrupt Status)
位域	类型	默认值	描述
			用于指示 buffer 收到了 8byte 的 setup 数据。OUT EP0 存在此 bit，其它的 OUT EP 无此 bit。
14	R/W1C	0x0	NYET_INT NYET interrupt，在非 isochronous OUT EP 发出 NYET 相应信号时，此位将置 1，产生中断信号。
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error interrupt，当 EP 收到 babble 信号，此位将置 1，产生中断信号。
11	R/W1C	0x0	PKT_DROP_STS_INT Packet Dropped Status Interrupt，此位置 1 用于指示存在 Isochronous OUT packet 丢失。此位无相应的 mask 位并且不会产生中断。
10:7	–	–	–
6	R/W1C	0x0	B2B_SETUP_INT Back-to-Back SETUP Packets Received，此位只适用 Control OUT EP。此位用于指示 device core 的 Control EP 收到了超过 3 个连续的 SETUP packets。
5	R/W1C	0x0	STS_PHASE_RX_INT Status Phase Received For Control Write，此中断位只适用于 Control OUT EP。此位用于指示进入 Control EP 的 Status Phase。
4	R/W1C	0x0	OUT_TOKEN_EP_DIS_INT OUT Token Received when Endpoint Disabled，此中断位表示 EP 在收到 OUT Token 的时候，相应的 EP 没有使能。
3	R/W1C	0x0	SETUP_DONE_INT SETUP Phase Done，只适用于 Control OUT EP。此位用于指示 Control EP 的 SETUP 阶段完成，可以对 SETUP 数据包进行解析。
2	–	–	–
1	R/W1C	0x0	DIS_EP_INT

默认值：0x00000000			OUT EP0 中断状态寄存器 (OUT EP0 Interrupt Status)
位域	类型	默认值	描述
			Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	RX_COMP_INT RX Transfer Completed Interrupt，该位用于表示 RX Transfer 完成。

7.1.6.42. 0x284 OUT_EP1_INT

默认值：0x00000000			OUT EP1 中断状态寄存器 (OUT EP1 Interrupt Status)
位域	类型	默认值	描述
31:16	–	–	–
15	R/W1C	0x0	SETUP_PKT_RX_INT Setup Packet Received Interrupt，只适用于 Control OUT EP。 用于指示 buffer 收到了 8byte 的 setup 数据。OUT EP0 存在此 bit，其它的 OUT EP 无此 bit。
14	R/W1C	0x0	NYET_INT NYET interrupt，在非 isochronous OUT EP 发出 NYET 相应信号时，此位将置 1，产生中断信号。
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error interrupt，当 EP 收到 babble 信号，此位将置 1，产生中断信号。
11	R/W1C	0x0	PKT_DROP_STS_INT Packet Dropped Status Interrupt，此位置 1 用于指示存在 Isochronous OUT packet 丢失。此位无相应的 mask 位并且不会产生中断。
10:7	–	–	–
6	R/W1C	0x0	B2B_SETUP_INT Back-to-Back SETUP Packets Received，此位只适用 Control OUT EP。此位用于指示 device core 的 Control EP 收到了超过 3 个连续的 SETUP packets。

默认值：0x00000000			OUT EP1 中断状态寄存器 (OUT EP1 Interrupt Status)
位域	类型	默认值	描述
5	R/W1C	0x0	STS_PHASE_RX_INT Status Phase Received For Control Write，此中断位只适用于 Control OUT EP。此位用于指示进入 Control EP 的 Status Phase。
4	R/W1C	0x0	OUT_TOKEN_EP_DIS_INT OUT Token Received when Endpoint Disabled，此中断位表示 EP 在 收到 OUT Token 的时候，相应的 EP 没有使能。
3	R/W1C	0x0	SETUP_DONE_INT SETUP Phase Done，只适用于 Control OUT EP。此位用于指示 Control EP 的 SETUP 阶段完成，可以对 SETUP 数据包进行解析。
2	–	–	–
1	R/W1C	0x0	DIS_EP_INT Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	RX_COMP_INT RX Transfer Completed Interrupt，该位用于表示 RX Transfer 完成。

7.1.6.43. 0x288 OUT_EP2_INT

默认值：0x00000000			OUT EP2 中断状态寄存器 (OUT EP2 Interrupt Status)
位域	类型	默认值	描述
31:16	–	–	–
15	R/W1C	0x0	SETUP_PKT_RX_INT Setup Packet Received Interrupt，只适用于 Control OUT EP。 用于指示 buffer 收到了 8byte 的 setup 数据。OUT EP0 存在此 bit，其它 的 OUT EP 无此 bit。
14	R/W1C	0x0	NYET_INT NYET interrupt，在非 isochronous OUT EP 发出 NYET 相应信号时， 此位将置 1，产生中断信号。
13	R/W1C	0x0	NACK_INT NACK Interrupt，NACK 中断，当 NAK 信号发送出去，此位置 1。

默认值：0x00000000			OUT EP2 中断状态寄存器 (OUT EP2 Interrupt Status)
位域	类型	默认值	描述
12	R/W1C	0x0	BABBLE_ERR_INT Babble Error interrupt，当 EP 收到 babble 信号，此位将置 1，产生中断信号。
11	R/W1C	0x0	PKT_DROP_STS_INT Packet Dropped Status Interrupt，此位置 1 用于指示存在 Isochronous OUT packet 丢失。此位无相应的 mask 位并且不会产生中断。
10:7	–	–	–
6	R/W1C	0x0	B2B_SETUP_INT Back-to-Back SETUP Packets Received，此位只适用 Control OUT EP。此位用于指示 device core 的 Control EP 收到了超过 3 个连续的 SETUP packets。
5	R/W1C	0x0	STS_PHASE_RX_INT Status Phase Received For Control Write，此中断位只适用于 Control OUT EP。此位用于指示进入 Control EP 的 Status Phase。
4	R/W1C	0x0	OUT_TOKEN_EP_DIS_INT OUT Token Received when Endpoint Disabled，此中断位表示 EP 在收到 OUT Token 的时候，相应的 EP 没有使能。
3	R/W1C	0x0	SETUP_DONE_INT SETUP Phase Done，只适用于 Control OUT EP。此位用于指示 Control EP 的 SETUP 阶段完成，可以对 SETUP 数据包进行解析。
2	–	–	–
1	R/W1C	0x0	DIS_EP_INT Endpoint Disabled Interrupt，EP 被禁止后产生的中断。
0	R/W1C	0x0	RX_COMP_INT RX Transfer Completed Interrupt，该位用于表示 RX Transfer 完成。

7.1.6.44. 0x2A0 IN_EP0_TSF_SIZ

默认值：0x00000000			IN EP0 传输 Size 寄存器 (IN EP0 Transfer Size)
位域	类型	默认值	描述
31:21	–	–	–
20:19	R/W	0x0	TX_PKT_CNT Transmit Packet Count，表示 EP0 的 Tansfer 包含的数据包的数量， 此域会在每次读取 TxFIFO 的情况下，逐次递减。
18:7	–	–	–
6:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP0 的 Transfer 的大小（以 Byte 为单位）。 此域会在每次 packet 从外部存储写入 TxFIFO 的情况下，逐次递减。 在发送数据的时候，配置 XferSize，相当于配置需要发送的数据长度。 若需要发送 N+1 个 Packet，那么 XferSize 配置为 N * MPS(Max packet size) + Short packet size TX_PKT_CNT 配置为 N + 1（N 个长包 + 1 个短包）

7.1.6.45. 0x2A4 IN_EP1_TSF_SIZ

默认值：0x00000000			IN EP1 传输 Size 寄存器 (IN EP1 Transfer Size)
位域	类型	默认值	描述
31	–	–	–
30:29	R/W	0x0	TX_MUTI_PKT Transmit Multiply Packet for IN EP，针对 Period-IN-EP，配置 每微帧 (micro frame) 必须发送的 packet 数目。 0x1：1 packet 0x2：2 packet 0x3：3 packet
28:19	R/W	0x0	PKT_CNT Packet Count，此域指示 EP 的组成 Transfer 的 data packet 的大小， PktCnt=XferSize/MPS。 This field is decremented every time a packet (maximum

默认值：0x00000000			IN EP1 传输 Size 寄存器 (IN EP1 Transfer Size)
位域	类型	默认值	描述
			size or short packet) is read from the TxFIFO
18:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP 的 Transfer 的大小。 在发送数据的时候，配置 XferSize，相当于配置需要发送的数据长度。 若需要发送 N+1 个 Packet，那么 XferSize 配置为 N * MPS(Max packet size) + Short packet size TX_PKT_CNT 配置为 N + 1（N 个长包 + 1 个短包） 在 XferSize 递减到 0 后，表示数据已写入 TxFIFO，硬件自动完成发送， 此时产生 XferComplete 中断。

7.1.6.46. 0x2A8 IN_EP2_TSF_SIZ

默认值：0x00000000			IN EP2 传输 Size 寄存器 (IN EP2 Transfer Size)
位域	类型	默认值	描述
31	–	–	–
30:29	R/W	0x0	TX_MUTI_PKT Transmit Multiply Packet for IN EP，针对 Period-IN-EP，配置 每微帧 (micro frame) 必须发送的 packet 数目。 0x1：1 packet 0x2：2 packet 0x3：3 packet
28:19	R/W	0x0	PKT_CNT Packet Count，此域指示 EP 的组成 Transfer 的 data packet 的大小， PktCnt=XferSize/MPS。 This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO
18:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP 的 Transfer 的大小。 在发送数据的时候，配置 XferSize，相当于配置需要发送的数据长度。 若需要发送 N+1 个 Packet，那么 XferSize 配置为 N * MPS(Max packet size) + Short packet size

默认值：0x00000000			IN EP2 传输 Size 寄存器 (IN EP2 Transfer Size)
位域	类型	默认值	描述
			TX_PKT_CNT 配置为 N + 1（N 个长包 + 1 个短包） 在 XferSize 递减到 0 后，表示数据已写入 TxFIFO，硬件自动完成发送， 此时产生 XferComplete 中断。

7.1.6.47. 0x2C0 OUT_EP0_TSF_SIZ

默认值：0x00000000			OUT EP0 传输 Size 寄存器 (OUT EP0 Transfer Size)
位域	类型	默认值	描述
31	–	–	–
30:29	R/W	0x0	STUP_PKT_CNT Setup Packet Count，表示能够接收连续的 SETUP 数据包数量 0x1：1 个数据包 0x2：2 个数据包 0x3：3 个数据包
28:20	–	–	–
19	R/W	0x0	PKT_CNT Packet Count，表示 EP0 的数据包的数量，此域会在每次写入 RxFIFO 的情况下，逐次递减。
18:7	–	–	–
6:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP0 的 Transfer 的大小（以 Byte 为单位）。此域 会在每次 packet 从 RxFIFO 读取数据包的情况下，逐次递减。 这里一般配置为 MPS (Max Packet Size) 即可，通过此域可以了解接 收到数据包的大小。 假设配置 PktCnt = 1，XferSize = MPS = 64byte，若接收到一个 8byte 的数据包，XferSize = 64 –8 = 56byte，PktCnt = 1 –1 =0 ，此时产生 Xfer Complete 中断。

7.1.6.48. 0x2C4 OUT_EP1_TSF_SIZ

默认值：0x00000000			OUT EP1 传输 Size 寄存器（OUT EP1 Transfer Size)
位域	类型	默认值	描述
31	–	–	–
30:29	R	0x0	RX_DTA_PID Received Data PID，此域只适用于 isochronous OUT EP，表示当前 EP 接收的 PID data 类型。 0x0: DATA0 0x1: DATA2 0x2: DATA1 0x3: MDATA
30:29	R/W	0x0	SUP_PKT_CNT SETUP Packet Count，此域只适用于 control OUT EP，表示 可以连续接收 SETUP Packet 的数目 0x1: 1 packet 0x2: 2 packets 0x3: 3 packets
28:19	R/W	0x0	PKT_CNT Packet Count，此域指示 EP 的组成 Transfer 的 data packet 的大小， PktCnt=XferSize/MPS。
18:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP 的 Transfer 的大小。 这里一般配置为 MPS (Max Packet Size 即可)，通过此域可以了解接收 到数据包的大小。 假设配置 PktCnt = 1，XferSize = MPS = 64byte，若接收到一个 8byte 的数据包，XferSize = 64 –8 = 56byte，PktCnt = 1 –1 =0 ，此时产生 Xfer Complete 中断。

7.1.6.49. 0x2C8 OUT_EP2_TSF_SIZ

默认值：0x00000000			OUT EP2 传输 Size 寄存器 (OUT EP2 Transfer Size)
位域	类型	默认值	描述
31	–	–	–

默认值：0x00000000			OUT EP2 传输 Size 寄存器 (OUT EP2 Transfer Size)
位域	类型	默认值	描述
30:29	R	0x0	RX_DTA_PID Received Data PID，此域只适用于 isochronous OUT EP，表示当前 EP 接收的 PID data 类型。 0x0: DATA0 0x1: DATA2 0x2: DATA1 0x3: MDATA
30:29	R/W	0x0	SUP_PKT_CNT SETUP Packet Count，此域只适用于 control OUT EP，表示 可以连续接收 SETUP Packet 的数目 0x1: 1 packet 0x2: 2 packets 0x3: 3 packets
28:19	R/W	0x0	PKT_CNT Packet Count，此域指示 EP 的组成 Transfer 的 data packet 的大小， PktCnt=XferSize/MPS。
18:0	R/W	0x0	XFR_SIZ Transfer Size，表示 EP 的 Transfer 的大小。 这里一般配置为 MPS (Max Packet Size 即可)，通过此域可以了解接收 到数据包的大小。 假设配置 PktCnt = 1，XferSize = MPS = 64byte，若接收到一个 8byte 的数据包，XferSize = 64 –8 = 56byte，PktCnt = 1 –1 =0 ，此时产生 Xfer Complete 中断。

7.1.6.50. 0x340+n*0x4(n=1~2) IN_EPn_TXFIFO_STA

默认值：0x00000000			IN EPn TXFIFO 状态寄存器 (In EPn Txfifo Status)
位域	类型	默认值	描述
31:16	–	–	–
15:0	RO	0	IN_EP_TXFIFO_STS

7.1.6.51. 0x360+(n−1)*0x4(n=1~2) TKN_QUEUEn

默认值：0x00000000			TKN_QUEUEn 寄存器
位域	类型	默认值	描述
31:0	RO	0	用于 Debug

7.1.6.52. 0xffc USB_VERSION

默认值：0x0000_0100			USB FSOTG 版本寄存器 (USB FSOTG Version)
位域	类型	默认值	描述
31:0	RO	0x0000_0100	版本寄存器 V1.0

7.1.6.53. 0x8000 USB_PHY_CTL

默认值：0x8000_0000			USB PHY 控制寄存器 (USB PHY Control)
位域	类型	默认值	描述
31	R/W	0x1	OTG ID PIN 选择 0x0: IDDIG 为低，用于 Host 模式 div 0x1: IDDIG 为高，用于 Device 模式
30	–	–	–
29	R/W	0x0	DP/DM 线下拉电阻使能，作 Host 时需使能 0x0: 下拉电阻断开 0x1: 下拉电阻使能
28:1	–	–	–
0	R/W	0x0	DP/DM 线上拉电阻使能 使能后控制器可根据配置的速度完成相应的电阻配置，作 Device 时需使能 0x0: 上拉电阻断开 0x1: 上拉电阻使能

7.2. Media Access Control (MAC)

EMAC 模块可通过 RMII 接口连接外部 PHY，实现兼容 IEEE802.3–2002 协议的以太网通讯。

7.2.1. 特性说明

1. MAC 支持特性：
   - 支持全双工/ 半双工
   - 支持 100/ 10 Mbps
   - 支持 MII/ RMII 接口
   - 支持内部 DMA
   - TXFIFO/ RXFIFO 的大小均为 2048 Byte，均支持阈值模式和存储–转发模式
   - 支持 MDIO 接口
   - 支持 Pause Frame/ 背压流控
   - 支持发送/ 接收 Jumbo Frame
   - 支持 VLAN 哈希过滤
   - 支持 64–bit 哈希地址过滤、完美地址过滤，支持八个地址寄存器
   - 支持源地址的插入/ 替换，VLAN 的插入/ 替换/ 删除，CRC 的插入/ 替换/ 删除
   - 支持接收校验和错误检测
   - 支持发送校验和计算
2. IEEE1588 特性支持：
   - 支持 IEEE1588–2002 时钟戳报文
   - 支持 IEEE1588–2008 时钟戳报文
   - 支持粗调和细调的校正方法
   - 支持 1x PPS 脉冲输出，脉冲支持间隔、脉宽可变
   - 支持 1x 时钟戳捕捉输入

7.2.2. 原理框图

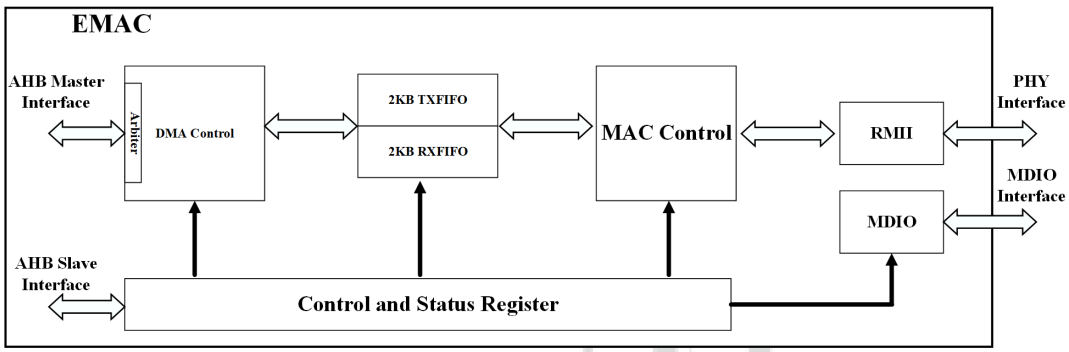


图 7–4 MAC 原理框图

7.2.3. 功能描述

7.2.3.1. RMII 接口时钟选择

RMII 支持内部参考时钟和外部参考时钟：

- 支持内部参考时钟，即 MAC 提供 50 MHz 时钟源至外部 PHY：
  - MAC 内部选通 PLL_EMAC_50M 作为 RMII 50 MHz 参考时钟源，通过 SYSCFG 的寄存器配置。
  - RMII_TX_CK 输出时钟，提供给外部 PHY 使用。
- 支持外部参考时钟，即 MAC 接收 PHY 提供的 RMII 50 MHz 参考时钟源：
  - REF_CLK 时钟使能，并且选通作为 RMII 50 MHz 参考时钟源，通过 SYSCFG 的寄存器配置。
  - RMII_TX_CK 不使能，无需提供时钟信号。

7.2.3.2. MDIO

通过 Management Data In/Out (MDIO) 接口，EMAC 可以对 PHY 的寄存器进行读写操作，进而控制 PHY 的工作模式以及获取 PHY 的工作状态。操作寄存器 MDIO_CTL 和 MDIO_DAT 可实现操作 MDIO 接口。

MDIO 接口由 MDC 以及 MDIO 实现：

- MDC：时钟输出信号，提供给 PHY。根据 AHB 总线的时钟来配置时钟分频系数获得，具体配置见寄存器 0x090 MDIO_CTL 描述。
- MDIO：数据输入输出信号。

  MDIO 的帧的格式：Idle + Preamble + Start + Opcode + Phy Addr + Reg Addr + TA + Data。

MDIO 帧域	描述
IDLE	IDLE 域
PREAMBLE	32 个连续的 bit 1

MDIO 帧域	描述
START	标志帧的起始位置，2'b10
OPCODE	操作标识域，2'b10 表示读，2'b01 表示写
PHY ADDR	5-bit 的 PHY 地址域
REG ADDR	PHY 寄存器地址
TA	Turnaround 域，读操作为 2'bZ0，写操作为 2'b10
DATA	16 bit 数据，读操作 MAC 为接收数据，写操作 MAC 为发送数据。

7.2.3.3. MAC 发送功能

1. SA 源地址自动插入或替换功能
   - 配置0x01C MAC_TX_FUNC 寄存器[14:12]，MAC TX 支持自动插入或替换发送帧的源地址，源地址可通过寄存器 MAC_ADDR0 或 MAC_ADDR1 配置。
2. TX Jabber Timer：MAC TX 支持 Jabber Timer，用于检测发送的帧长度。
   - 若关闭 Jabber Timer，发送帧的长度可以达到 16383 bytes。
   - 若未关闭 Jabber Timer 功能，最多只能发送 2048 bytes。使能 Frame 后，最大发送的长度为 10240 bytes），并且会在 TDES0 的 BIT14 置 1，指示发送的数据过长，发生 Jabber Timeout。
3. CRC 插入/替换
   - 配置 TX DMA 描述符 TDES0 [27]，可以对发送帧进行 CRC 的插入。
   - 配置 TX DMA 描述符 TDES0 [24]，可以对发送帧进行 CRC 的替换。此时需要设置 TDES0[27] = 1，关闭 CRC 的插入功能。
4. 自动补 0
   - 配置 TX DMA 描述符 TDES0 [27]，可以对长度低于 64 bytes 的发送帧进行自动补 0，并且此时会强制产生 4 bytes 的 CRC，即 short frame + 0 pad + CRC = 64 bytes。
   - 若不需要进行自动补 0，可以设置 TDES0[27] = 1。
5. TX IP 校验和插入：MAC 支持对发送帧的 IP 校验和进行计算和插入。
   a. 配置 MAC_CFG[7] = 1，打开 IP Offload 功能。
   b. 通过 TDES0[23:22] 选择 IP 检验和的计算类型。MAC 支持 IP header 校验和、IP Payload 校验和以及伪 header 校验的硬件加速计算。
   c. 打开 IP Offload 功能后，TDES0[16] 可以指示发送帧的 IP 校验和是否正确，置 1 表示发送帧存在 IP 校验和错误。

7.2.3.4. MAC 接收功能

1. 地址过滤。对于接收帧，MAC 可以进行地址过滤，包括：
    - 单播地址过滤
        - 源地址过滤
        - 反向源地址过滤
        - 目的地址过滤
        - 反向目的地址过滤
    - 广播地址过滤
    - 多播帧过滤
    - 控制帧过滤

    RDES0[30]、[13] 分别指示接收帧的目的地址、源地址过滤是否成功。
2. RX Watchdog：MAC RX 支持 RX Watchdog，用于检测接收的帧长度
    - 若关闭 RX Watchdog，发送帧的长度可以达到 16383 bytes
    - 若未关闭 RX Watchdog，最多只能接收 2048 bytes。使能 Jumbo Frame 后，最大接收的长度为 10240 bytes，并且会在 RDES0 [4] 置 1，指示接收的数据过长，发生 Watchdog Timeout。
3. CRC/ PAD 剥离：MAC 支持对接收帧进行 CRC/ PAD 剥离。
    - 配置 EN_STRIP_FCS = 1，可以对长度 ≥ 1536 bytes 的接收帧进行 CRC 剥离。
    - 配置 EN_STRIP_NML = 1，可以对长度 < 1536 bytes 的接收帧进行 CRC 以及 PAD 剥离。
4. RX IP 校验和计算：MAC 支持对接收帧的 IP 校验和进行计算。
    - 配置 MAC_CFG[7] = 1，打开 IP Offload 功能。此时需要设置 ALT_DES_SIZ = 0x1，激活 RDES4 的状态，软件通过 RDES4 可以获取接收帧 IP 检验和的校验结果，从而得知接收的 IP 帧的校验和结果。

7.2.3.5. 流控

流控功能可通过寄存器 0x034 FLOW_CTL 控制，包括：

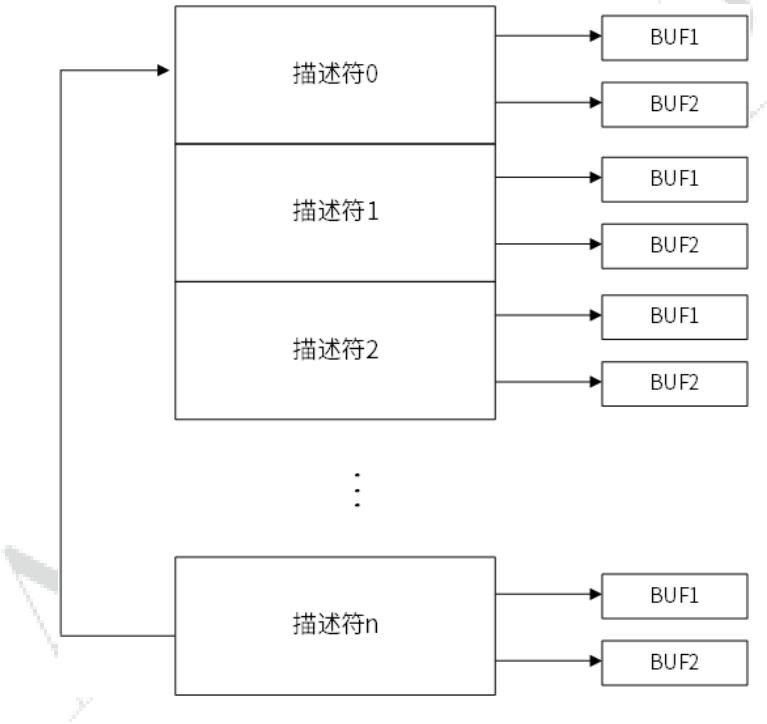
1. TX 流控
    - 在全双工模式下，支持 Pause 帧的发送。
    - 在半双工模式下，支持背压操作。
2. RX 流控
    - 支持 Pause 帧的解析，以及停止发送约定的 Pause 时间。
3. 单播 Pause 帧
    - MAC 模块支持单播的 Pause 帧。

7.2.3.6. DMA 结构

EMAC 内部具有 DMA，用于将发送 BUFFER（DRAM 地址）的数据传送到 TXFIFO， RXFIFO 接收到的数据传送到接收 BUFFER（DRAM 地址）。DMA 可以支持两种结构，分别是环型结构和链型结构。

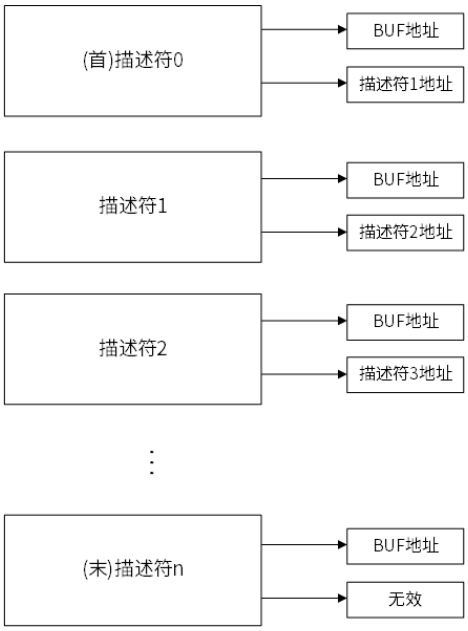
- 环型结构 (Ring) 的描述符，可以通过 DES_SKIP_LEN 来配置设定两两描述符之间的跳跃长度。

  通过描述符的 START_DES 和 END_DES 来决定环形结构的第一个描述符和最后一个描述符，从而形成环路。



- 链型结构 (Chain) 的描述符，则通过描述符的 Next Descriptor Address 来进行指向连接。

  在最后一个描述符的 Next Descriptor Address 域上设置为第一个描述符地址，可以形成环路。



7.2.3.7. DMA 发送描述符

发送描述符 TDES4~7 的应用，需要设置寄存器 0x004 DMA0_CONF 的 ALT_DES_SIZ = 1。其中，TDES4~5 Reserved，TDES6~7 时钟戳信息。

表 7–3 发送描述符 TDES0

Bit	域描述
31	TX_PRC，Transmit in Process，发送进行中。 • 0x0：DMA 空闲，DMA 完成当前的描述符的数据发送。 • 0x1：DMA 发送进行中。
30	EN_TX_COMP_INT，Enable Transmit Complete Interrupt，使能发送完成中断，此位只有在 BIT29 有效时才有效。 • 0x0：未使能。 • 0x1：使能，此位只有在 BIT29 有效
29	END_DES，End Descriptor，最后一个描述符。 • 0x0：无效。 • 0x1：有效，表示此描述符为最后一个描述符
28	START_DES，Start Descriptor，第一个描述符。 • 0x0：无效。 • 0x1：有效，表示此描述符为第一个描述符
27	DIS_CRC，Disable CRC，禁止 CRC 功能时生效，只需在第一个描述符设置此位即可。 • 0x0：无效。 • 0x1：有效，禁止 CRC
26	DIS_PAD，Disable Padding，禁止补 0 功能时生效，只需在第一个描述符设置此位即可。 • 0x0：无效。 • 0x1：有效，禁止对帧长低于 64 bytes 进行补 0
25	EN_TX_TMSTMP，Enable Transmit Timestamp，使能发送添加时间戳功能，只需在第一个描述符设置此位即可。 • 0x0：未使能 • 0x1：使能
24	EN_CRC_REP，Enable CRC Replacement，使能 CRC 替代功能。 必须保证软件发送帧的内容包含了 CRC，使能 CRC 替代功能后，会进行 CRC 重新计算并替代。另外为了避免 CRC 重复，需要 DIS_CRC =1 禁止 CRC 插入。在第一个描述符设置此位即可。

表 7–3 发送描述符 TDES0 (续)

Bit	域描述
	• 0x0：未使能 • 0x1：使能
23:22	EN_CIC，Enable Checksum Insertion Control，使能校验和插入功能，只需在第一个描述符设置此位即可。 • 0x0：禁止校验和插入。 • 0x1：开启 Header 校验和计算以及插入功能。 • 0x2：开启 Header、Payload 校验和的计算及插入功能，未开启伪 Header 的校验和计算。 • 0x3：开启 Header、Payload 校验和的计算及插入功能，并且开启伪 Header 的校验和计算。
21	TX_END_RNG，Transmit End of Ring，DMA 发送环的末端，表示此描述符为 DMA 发送环的最后一个描述符，结束后跳回起始描述符。 • 0x0：无效 • 0x1：生效
20	EN_DMA_CHN_MOD，Enable DMA Chain Mode，使能 Chain 模式。 • 0x0：未使能，采用 DMA Ring 模式 • 0x1：使能，采用 DMA Chain 模式
19:18	VLAN_TAG_CTL，VLAN Tag Control，VLAN 标签控制，只需在第一个描述符设置此位即可。 • 0x0：无效操作。 • 0x1：删除发送帧中的 VLAN 标签内容。 • 0x2：插入 VLAN 标签，标签内容。 • 0x3：替代 VLAN 标签。
17	TX_TMSTMP_STS，Transmit Timestamp Status，发送时间戳状态，此域只有在最后一个描述符有效。 • 0x0：无操作 • 0x1：完成发送时间戳的捕捉，时间戳写入在 TDES6 和 TDES7。
16	IP_HD_ERR，IP Header Error 此位置 1 表示 MAC 检测发送的 IP 报文的 Header 存在错误，此域只有在最后一个描述符有效。
15	TXERR_STS，Transmit Error Status，发送错误状态指示位。 • 0x0：无错误。 • 0x1：出现错误，错误源可能由于 BIT16/14/13/12/11/10/9/8/2/1 相应的错误状态引起
14	JAB_TO_ERR，Jabber Timout Error，发送 Jabber 超时错误，此位只有在 Jabber 检测功能未禁止的情况下有效。

表 7–3 发送描述符 TDES0 (续)

Bit	域描述
	• 0x0：无异常 • 0x1：出现发送超时错误
13	TXFIFO_FLSH_ERR，TXFIFO Flush Error，清除错误，出现软件清除 TXFIFO 的帧数据的操作导致的错误。 • 0x0：无异常 • 0x1：出现错误状态
12	IP_PL_ERR，IP Payload Error，此位置 1 指示 MAC 检测到发送的 TCP、UDP、ICMP IP 数据包 payload 存在错误。
11	LOS_CR，Loss Carrier，此位置 1 指示在帧发送发生了载波丢失，此位只有在半双工模式下有效。
10	NO_CR，No Carrier，此位置 1 指示在发送的过程中，PHY 的载波侦听 CS 信号并没有生效，即没有检测到载波。
9	LT_COL，Late Collision，此位置 1 指示由于碰撞导致发送终止，此碰撞发生在碰撞窗口之后。此位在 TXFIFO_UF_ERR = 1 时无效。
8	EXS_COL，Excessive Collision，此位置 1 指示当前帧尝试发送的过程中，由于连续的 16 次碰撞的产生终止发送帧。如果取消重传（DIS_RETRY = 1），只要出现 1 次的碰撞，此位则立即置 1。
7	VLAN_STS，VLAN Status，VLAN 帧状态。 • 0x0：发送帧为非 VLAN 帧 • 0x1：发送帧为 VLAN 帧
6:3	COL_CNT，Collision Count，此域指示在发送前发生多少次碰撞。此域在 TDES0. Bit8 =1 的时候无效。
2	EX_DEF_ERR，Excessive Deferral Error，发送终止由于回退时间过长，此位只有在 EN_DEF_CHECK = 1 的情况下有效。 • 0x0：未发生错误。 • 0x1：出现回退过量错误
1	TXFIFO_UF_ERR，TXFIFO Underflow Error，TXFIFO 下溢出错误，DMA 从 Buffer 地址读取帧数据错误。 • 0x0：未出现错误 • 0x1：发生下溢出错误
0	DEF_STS，Defer Status，回退状态，在发送前检测到载波而引起的回退，此位只对于半双工模式有效。

表 7–3 发送描述符 TDES0 (续)

Bit	域描述
	• 0x0：未出现回退 • 0x1：发送时出现回退

表 7–4 发送描述符 TDES1

Bit	域描述
31:29	TX_SA_CTL，Transmit Source Address Control，发送源地址控制。BIT31 = 1，表示采用 Address1 进行插入或替代。BIT31 = 0，表示采用 Address0 进行插入或替代。 Bit30:29 • 0x0：无操作 • 0x1：插入源地址 • 0x2：替代源地址 • 0x3：保留
28:16	TX_BUF2_SIZ，Transmit Buffer2 Size，发送缓存 2 的空间大小。
15:13	Reserved
12:0	TX_BUF1_SIZ，Transmit Buffer1 Size，接收缓存 1 的空间大小。

表 7–5 发送描述符 TDES2

Bit	域描述
31:0	BUF1_ADDR，Buffer1 Address，Buffer1 地址指针，Buffer1 的物理地址。

表 7–6 发送描述符 TDES3

Bit	域描述
31:0	BUF2_ADDR/NXT_DES_ADDR • DMA 采用 Ring 模式时，此域为 BUF2_ADDR BUF2_ADDR，Buffer2 Address，Buffer2 地址指针，Buffer2 的物理地址。

表 7–6 发送描述符 TDES3 (续)

Bit	域描述
	• 采用 Chain 模式时，此域为 NXT_DES_ADDR。 NXT_DES_ADDR，Next Descriptor Address，下一个描述符的地址，指向下一个描述符的物理地址，从而将描述符连接起来。

表 7–7 发送描述符 TDES6

Bit	域描述
31:0	TX_STMP_LO，Transmit Timestamp Low 32bit，发送时间戳的低 32 位，此域由 DMA 自动填写，与 TDES7 的高 32bit 组成完整的时间戳。

表 7–8 发送描述符 TDES7

Bit	域描述
31:0	TX_STMP_HI，Transmit Timestamp High 32bit，发送时间戳的高 32 位，此域由 DMA 自动填写，与 TDES6 的低 32bit 组成完整的时间戳。

注：

发送描述符 RDES4~7（RDES4 状态信息，RDES5 Reserved，RDES6~7 时钟戳信息）的应用，需要设置寄存器 DMA0/1_CONF 的 ALT_DES_SIZ = 1。

7.2.3.8. DMA 接收描述符

接收描述符 RDES4~7（RDES4 状态信息，RDES5 Reserved，RDES6~7 时钟戳信息）的应用，需要设置寄存器 DMA0/1_CONF 的 ALT_DES_SIZ = 1。

表 7–9 接收描述符 RDES0

Bit	域描述
31	RX_PRC，Receive in Process，接收进行中。 • 0x0：DMA 空闲，DMA 完成当前的描述符的数据接收 • 0x1：DMA 接收进行中
30	DA_FLT，Destination Address Filter Fail，目的地址过滤失败。 • 0x0：无效 • 0x1：指示接收帧的目的地址过滤失败
29:16	RX_FRM_LEN，Receive Frame Length，接收到的帧的长度，此域在最后一个描述符有效，并且无异常发生。

表 7–9 接收描述符 RDES0 (续)

Bit	域描述
15	RXERR_STS，Receive Error Status，接收错误状态指示位。 • 0x0：无错误 • 0x1：出现错误，错误源可能由于 RDES0 的 BIT14/11/7/6/4/3/1 或 RDES4 的[4:3] 相应的错误状态引起
14	RX_DES_ERR，Receive Descriptor Error，接收描述符错误，此域在最后一个描述符有效。 • 0x0：无效 • 0x1：由于描述符异常导致的接收中断
13	SA_FLT，Source Address Filter Fail，源地址过滤失败。 • 0x0：无效 • 0x1：指示接收帧的源地址过滤失败
12	RX_LEN_ERR，Receive Length Error，接收长度错误。 • 0x0：无效 • 0x1：指示接收长度错误，接收帧的实际长度与帧的长度域值不一致
11	RXFIFO_OF_ERR，RXFIFO Overflow Error，接收溢出错误。 • 0x0：无效 • 0x1：指示接收溢出错误，由于 RXFIFO 溢出，导致接收帧的损坏
10	VLAN_TAG，VLAN Tag，此位置 1 表示接收的帧位 VLAN 帧。
9	START_DES，Start Descriptor，第一个描述符。 • 0x0：无效 • 0x1：有效，表示此描述符为第一个描述符
8	END_DES，End Descriptor，最后一个描述符。 • 0x0：无效 • 0x1：有效，表示此描述符为最后一个描述符
7	TMSTMP_AVL/ IP_CHSM_ERR/ GNT_FRM 此位可指示三种状态： • TMSTMP_AVL，Timestamp Available，置 1 表示接收的帧已打时间戳，从 RDES6~7 可获得。 • IP_CHSM_ERR，IP Checksum Error，置 1 表示 IP 校验和错误或表示接收帧位非 IPv4 帧。 • GNT_FRM，Giant Frame，置 1 表示接收的帧位 Frame。
6	LT_COL_ERR，Late Collision Error，Late Collision 错误。

表 7–9 接收描述符 RDES0 (续)

Bit	域描述
	• 0x0：无效 • 0x1：接收发生 Collision 错误
5	RX_FRM_TYP，Receive Frame Type，接收帧的类型。 • 0x0：接收帧为 IEEE 802.3 类型的帧 • 0x1：接收帧为以太网类型的帧
4	RX_WD_TO_ERR，Receive Watchdog Timeout Error，接收看门狗的超时错误。 • 0x0：无效 • 0x1：发生接收看门狗的超时错误
3	RX_ERR，Receive Error，接收过程出现 ERR 信号。 • 0x0：无效 • 0x1：接收过程出现 ERR 信号
2	Reserved
1	RX_CRC_ER，RReceive CRC Error，接收 CRC 错误，此域只有在最后一个描述符有效。 • 0x0：无效 • 0x1：接收帧出现 CRC 错误
0	EXT_STS_AVA/ RX_MAC_ADDR • EXT_STS_AVA，Extend Status Available，当时钟戳或 IP 校验和 Offload 功能使能，此位置 1 表示 RDES4 生效。 • RX_MAC_ADDR，Rx MAC Address，当时钟戳、IP 校验和 Offload 功能均未使能，此位置 1 表示接收帧的目的地址匹配 MAC Address1~7 之一。此位清 0 表示接收帧的目的地址匹配 Address0。

表 7–10 接收描述符 RDES1

Bit	域描述
31	DIS_COMP_INT，Disable Complete Interrupt，禁止接收完成中断的产生。 • 0x0：未禁止 • 0x1：禁止
30:29	Reserved
28:16	RX_BUF2_SIZ，Receive Buffer2 Size，接收缓存 2 的空间大小。

表 7–10 接收描述符 RDES1 (续)

Bit	域描述
15	RX_END_RNG，Receive End of Ring，指示在 DMA 的 Ring 模式下，此描述为最后的描述符，完成后跳转回起始描述符。
14	RX_DMA_MOD，Receive DMA Mode，接收 DMA 的模式。 • 0x0：Ring 模式 • 0x1：Chain 模式
13	Reserved
12:0	RX_BUF1_SIZ，Receive Buffer1 Size，接收缓存 1 的空间大小。

表 7–11 接收描述符 RDES2

Bit	域描述
31:0	BUF1_ADDR，Buffer1 Address，Buffer1 地址指针，Buffer1 的物理地址。

表 7–12 接收描述符 RDES3

Bit	域描述
31:0	BUF2_ADDR/NXT_DES_ADDR • DMA 采用 Ring 模式时，此域为 BUF2_ADDR BUF2_ADDR，Buffer2 Address，Buffer2 地址指针，Buffer2 的物理地址。 • DMA 采用 Chain 模式时，此域为 NXT_DES_ADDR。NXT_DES_ADDR， Next Descriptor Address，下一个描述符的地址，指向下一个描述符的物理地址，从而将描述符连接起来。

表 7–13 接收描述符 RDES4

Bit	域描述
31:21	Reserved
20:18	VLAN_TAG_PRI_V，VLAN Tag Priority Value，此域内容填入接收到的包的 TAG 用户值域。
17	AV_TAG_PKT_RCD，AV Tagged Packet Received，此位置 1 表示接收到带 AV 标签的数据包。
16	AV_PKT_RCD，AV Packet Received，此位置 1 表示接收到 AV 包。
15	Reserved
14	TMSTMP_DRP，Timestamp Dropped，时钟戳丢失，由于 RXFIFO 溢出，导致时钟戳丢失。

表 7–13 接收描述符 RDES4 (续)

Bit	域描述
	• 0x0：无效 • 0x1：时钟戳丢失
13	PTP_VER，PTP Version，PTP 版本。 • 0x0：IEEE1588 • 0x1：IEEE1588V2
12	RX_PTP_FRM，Received PTP Frame Type，接收到 PTP 帧的类型。 • 0x0：接收到 PTP 报文通过 UDP–IPV4 或 IPV6 的形式传送 • 0x1：接收到 PTP 报文直接通过以太网帧形式传送
11:8	PTP_RX_MSG_TYP，Received PTP Message Type，接收到的 PTP 消息报文类型。 • 0x0：无 PTP 报文 • 0x1：SYNC 报文 • 0x2：Follow Up 报文 • 0x3：Delay_Req 报文 • 0x4：Delay_Resp 报文 • 0x5：Pdelay_Req 报文 • 0x6：Pdelay_Resp 报文 • 0x7：Pdelay_Resp_Follow_Up 报文 • 0x8：Announce 报文 • 0x9：Management 报文 • 0xA：Signaling 报文 • 0xB~0xE：Reserved • 0xF：PTP Packet with Reserved message type
7	RX_IPV6_PKT，Received IPV6 Packet，接收到 IPV6 包。 • 0x0：无效 • 0x1：接收到的 IP 包为 IPV6 包
6	RX_IPV4_PKT，Received IPV4 Packet，接收到 IPV4 包。 • 0x0：无效 • 0x1：接收到的 IP 包为 IPV4 包
5	DIS_IP_CHSUM，Disable IP Checksum Function，关闭 IP 校验和功能。

表 7–13 接收描述符 RDES4 (续)

Bit	域描述
	• 0x0：未关闭 • 0x1：关闭 IP 校验和功能
4	IP_PL_ERR，IP Payload Error，IP 包校验错误，接收到的 IP 包的 TCP/UDP/ICMP 的校验和错误，或长度错误。 • 0x0：无错误 • 0x1：出现 IP 包校验和错误
3	IP_HD_ERR，IP Header Error，接收到的 IP 包的首部校验和与计算的校验和不一致，或者 IP 版本与 Ethernet Type 域值不一致。 • 0x0：无错误 • 0x1：出现首部校验和错误
2:0	IP_PKT_TYP，IP Packet Type，IP 包类型，此域只有在 BIT6/7 生效时才有效。 • 0x0：未知类型 • 0x1：UDP 包 • 0x2：TCP 包 • 0x3：ICMP 包 • others：Reserved

表 7–14 接收描述符 RDES6

Bit	域描述
31:0	RX_STMP_LO，Receive Timestamp Low 32bit，接收时间戳的低 32 位，此域由 DMA 自动填写，与 TDES7 的高 32bit 组成完整的时间戳。

表 7–15 接收描述符 RDES7

Bit	域描述
31:0	RX_STMP_HI，Receive Timestamp High 32bit，发送时间戳的高 32 位，此域由 DMA 自动填写，与 TDES7 的低 32bit 组成完整的时间戳。

7.2.3.9. IEEE1588

7.2.3.9.1. 时钟戳收发

IEEE1588 时钟戳的收发，是指在 MAC 发送或接收到报文后，则触发时钟戳的捕捉，此时对应的发送或接收描述符会将相应的 Timestamp 信息记录在 TDES6、TDES7 和 RDES6、RDES7 中。

为了让 DMA 能够记录 Timestamp，需要配置以下信息：

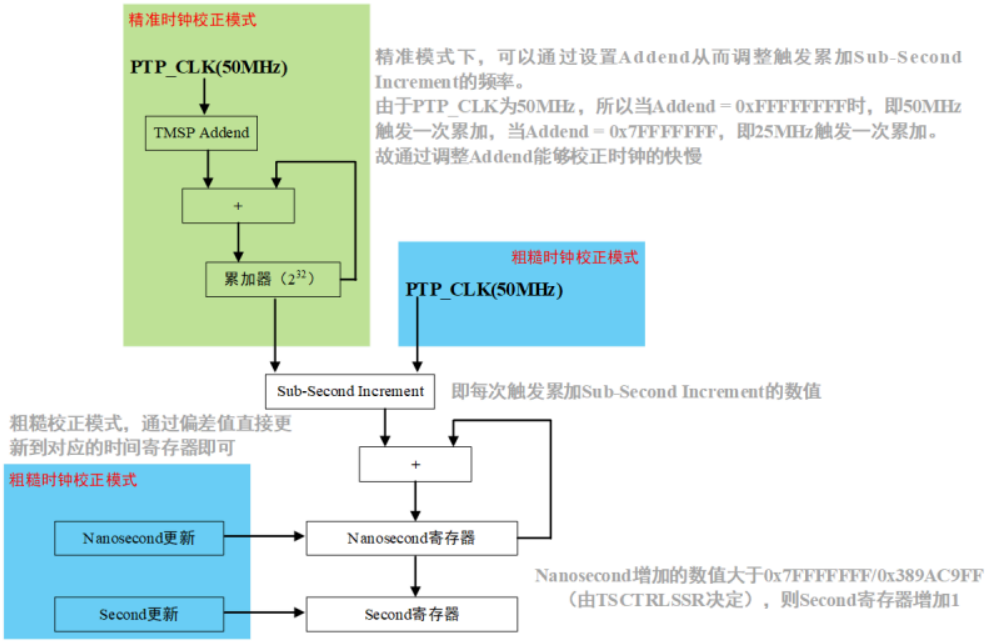
- 寄存器 ALT_DES_SIZ = 1
- 描述符的大小为 8 words
- 使能 TDES0 的 TTSE（Transmit Timestamp Enable）。

为了接收时间戳信息，需在 Timestamp 控制寄存器上使能对应的接收帧类型。

7.2.3.9.2. 时钟校准

IEEE1588 功能支持对内部系统时钟的校准：

- 粗糙校准：通过 PTP 协议报文的交互计算出时钟偏差，通过 Update 寄存器更新系统时间。
- 精准校准：通过 PTP 协议报文的交互计数时钟偏差的频率比，更新 Timestamp Addend 寄存器，实现同步时钟。



时钟校正模式

7.2.4. 寄存器列表

表 7–16 MAC 寄存器地址映射表

Module Name	Mapping Size	Offset
MAC	4 KB	0x000

表 7–17 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	MAC_CONF	MAC 配置寄存器	0x000 MAC_CONF
0x004	DMA0_CONF	DMA0 配置寄存器	0x004 DMA0_CONF

表 7–17 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x00C	DMA0_INT_STS	DMA0 中断状态寄存器	0x00C DMA0_INT_STS
0x010	DMA0_INT_EN	DMA0 中断使能寄存器	0x010 DMA0_INT_EN
0x01C	MAC_TX_FUNC	MAC 发送功能寄存器	0x01C MAC_TX_FUNC
0x020	MAC_RX_FUNC	MAC 接收功能寄存器	0x020 MAC_RX_FUNC
0x024	TX_DMA0_CTL	发送 DMA0 控制寄存器	0x024 TX_DMA0_CTL
0x028	RX_DMA0_CTL	接收 DMA0 控制寄存器	0x028 RX_DMA0_CTL
0x034	FLOW_CTL	流控寄存器	0x034 FLOW_CTL
0x038	VLAN_TAG	VLAN 标签寄存器	0x038 VLAN_TAG
0x03C	VLAN_FLT	VLAN 过滤寄存器	0x03C VLAN_FLT
0x040	MAC_FRM_FLT	MAC 帧过滤寄存器	0x040 MAC_FRM_FLT
0x044	HASH_TBL_HIGH	高位哈希列表寄存器	0x044 HASH_TBL_HIGH
0x048	HASH_TBL_LOW	低位哈希列表寄存器	0x048 HASH_TBL_LOW
0x050	MAC_ADDR0_HIGH	高位 MAC 地址 0 寄存器	0x050 MAC_ADDR0_HIGH
0x054	MAC_ADDR0_LOW	低位 MAC 地址 0 寄存器	0x054 MAC_ADDR0_LOW
0x058	MAC_ADDR1_HIGH	高位 MAC 地址 1 寄存器	0x058 MAC_ADDR1_HIGH
0x05C	MAC_ADDR1_LOW	低位 MAC 地址 1 寄存器	0x05C MAC_ADDR1_LOW
0x060	MAC_ADDR2_HIGH	高位 MAC 地址 2 寄存器	0x060 MAC_ADDR2_HIGH
0x064	MAC_ADDR2_LOW	低位 MAC 地址 2 寄存器	0x064 MAC_ADDR2_LOW
0x068	MAC_ADDR3_HIGH	高位 MAC 地址 3 寄存器	0x068 MAC_ADDR3_HIGH
0x06C	MAC_ADDR3_LOW	低位 MAC 地址 3 寄存器	0x06C MAC_ADDR3_LOW
0x074	MAC_ADDR4_LOW	低位 MAC 地址 4 寄存器	0x074 MAC_ADDR4_LOW
0x078	MAC_ADDR5_HIGH	高位 MAC 地址 5 寄存器	0x078 MAC_ADDR5_HIGH
0x07C	MAC_ADDR5_LOW	低位 MAC 地址 5 寄存器	0x07C MAC_ADDR5_LOW
0x080	MAC_ADDR6_HIGH	高位 MAC 地址 6 寄存器	0x080 MAC_ADDR6_HIGH
0x084	MAC_ADDR6_LOW	低位 MAC 地址 6 寄存器	0x084 MAC_ADDR6_LOW
0x088	MAC_ADDR7_HIGH	高位 MAC 地址 7 寄存器	0x088 MAC_ADDR7_HIGH
0x090	MDIO_CTL	MDIO 控制寄存器	0x090 MDIO_CTL
0x094	MDIO_DATA	MDIO 数据寄存器	0x094 MDIO_DATA
0x0A0	DEBUG_REG	Debug 寄存器	0x0A0 DEBUG_REG

表 7–17 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0B0	TXDMA0_DESC_START	TXDMA0 起始描述符地址寄存器	0x0B0 TXDMA0_DESC_START
0x0B4	RXDMA0_DESC_START	RXDMA0 起始描述符地址寄存器	0x0B4 RXDMA0_DESC_START
0x0B8	CUR_TXDMA0_DESC	TXDMA0 当前描述符地址寄存器	0x0B8 CUR_TXDMA0_DESC
0x0BC	CUR_RXDMA0_DESC	RXDMA0 当前描述符地址寄存器	0x0BC CUR_RXDMA0_DESC
0x0C0	CUR_TXDMA0_BUFADDR	TXDMA0 缓存地址寄存器	0x0C0 CUR_TXDMA0_BUFADDR
0x0C4	CUR_RXDMA0_BUFADDR	RXDMA0 缓存地址寄存器	0x0C4 CUR_RXDMA0_BUFADDR
0x0E8	DMA0_MISFRM_BUFOVF	DMA0 丢帧溢出寄存器	0x0E8 DMA0_MISFRM_BUFOVF
0x100	TMSTMP_CTL	时钟戳控制寄存器	0x100 TMSTMP_CTL
0x0104	SUB_SEC_INCR	亚秒递增寄存器	0x0104 SUB_SEC_INCR
0x0108	TMSMP_ADDEND	时钟戳加数寄存器	0x0108 TMSMP_ADDEND
0x010C	SYS_TIME_SEC	时间戳秒寄存器	0x010C SYS_TIME_SEC
0x0110	SYS_TIME_NANO_SEC	时间戳亚秒寄存器	0x0110 SYS_TIME_NANO_SEC
0x0114	UPDT_TIME_SEC	时间戳秒更新寄存器	0x0114 UPDT_TIME_SEC
0x0118	UPDT_TIME_NANO_SEC	时间戳亚秒更新寄存器	0x0118 UPDT_TIME_NANO_SEC
0x011C	TRGT_TIME_SEC	目标时间戳秒寄存器	0x011C TRGT_TIME_SEC
0x0120	TRGT_TIME_NANO_SEC	目标时间戳亚秒寄存器	0x0120 TRGT_TIME_NANO_SEC
0x0124	AUX_TMSMP_SEC	捕捉时钟秒寄存器	0x0124 AUX_TMSMP_SEC
0x0128	AUX_TMSMP_NANO_SEC	捕捉时钟亚秒寄存器	0x0128 AUX_TMSMP_NANO_SEC
0x012C	SYS_TIME_HI_WORD_SEC	高位时钟戳寄存器	0x012C SYS_TIME_HI_WORD_SEC
0x0130	TMSMP_STS	时钟戳状态寄存器	0x0130 TMSMP_STS
0x0134	PPS_CTL	脉冲控制寄存器	0x0134 PPS_CTL
0x0138	PPS0_INTERVAL	脉冲间隔寄存器	0x0138 PPS0_INTERVAL
0x013C	PPS0_WIDTH	脉冲宽度寄存器	0x013C PPS0_WIDTH
0x0FFC	VERSION	EMAC 版本号	0x0FFC VERSION

7.2.5. 寄存器描述

7.2.5.1. 0x000 MAC_CONF

默认值：0x00000000			MAC 配置寄存器 (MAC Configure)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	IP_CHKSUM_OL IP Layer Checksum Offload，IP 层的校验和 Offload • 0x0：未使能 • 0x1：使能
6:5	R/W	0x0	PRE_LEN_SET Preamble Length Set，发送帧的前导符长度设置 • 0x0：前导符为 7 字节 • 0x1：前导符为 5 字节 • 0x2：前导符为 3 字节 • 0x3：Reserved
4	R/W	0x0	DUP_MODE_SEL Duplex Mode Select，全/半双工模式选择 • 0x0：半双工模式 • 0x1：全双工模式
3	R/W	0x0	LP_MODE_EN Loop Mode Enable，选择回环模式，自发自收 • 0x0：未使能 • 0x1：使能回环模式
2:1	R/W	0x0	SPD SPEED，以太网速度配置 • 0x0：Reserved • 0x1：Reserved • 0x2：10M • 0x3：100M
0	R/WAC	0x1	SF_RST

默认值：0x00000000			MAC 配置寄存器 (MAC Configure)
位域	类型	默认值	描述
			Software Reset，软件复位，此位置 1，MAC 进行软件复位，复位所有的内部逻辑以及寄存器，完成复位后自动清 0。

7.2.5.2. 0x004 DMA0_CONF

默认值：0x00080400			DMA0 配置寄存器 (DMA0 Configure)
位域	类型	默认值	描述
31	R/W	0x0	RB_INCR_BURST Rebuild INCRx Burst，重新建立 INCR 突发模式 • 0x0：未使能 • 0x1：使能
30:	–	–	–
26	R/W	0x0	ADDR_ALN Address-Aligned Beats，地址对齐控制 • 0x0：未使能 • 0x1：使能
25	R/W	0x0	SEP_PBL Separate PBL，TX、RX 使用独立的 PBL 控制 • 0x0：未使能，TX、RX 均应用 LEN 配置控制 DMA 的 PBL • 0x1：使能，TX、RX 使用独立的 PBL 控制，TX 应用 LEN 配置控制 TX DMA 的 PBL，RX 应用 LEN 配置控制 RX DMA 的 PBL
24:19	R/W	0x1	RX_BURST_LEN Rx DMA Burst Length，此域表示在一次 DMA 传输中最大传输的 beats 的数量。此域设置为 1/2/4/8/16/32，其它值将会导致无法预知的行为。
18	R/W	0x0	MIX_BURST Mixed Burst，混合突发模式 • 0x0：未使能 • 0x1：使能
17	R/W	0x0	FIX_BURST Fixed Burst，固定突发模式

默认值：0x00080400			DMA0 配置寄存器 (DMA0 Configure)
位域	类型	默认值	描述
			• 0x0：非 Burst 模式，突发模式使用 SINGLE 和 INCR burst 进行传输 • 0x1：Fixed Burst 模式，突发模式使用 SINGLE、INCR4、INCR8、或 INCR16
16	R/W	0x0	EN_BL8_MOD Enable Burst Length x8 Mode，使能 Length 8 倍模式 • 0x0：未使能 • 0x1：使能
15:10	R/W	0x1	BURST_LEN Burst Length，DMA Burst 长度，表示在一次 DMA 传输最多可以传输的 beats 数量。 此域可设置为 1/2/4/8/16/32，其它的数值会导致不可预知的行为。配置更高的 Length 需要配合 EN_BL8_MOD=1，采用 8 倍 Length 模式，即在原有的基础上乘 8。
9	R/W	0x0	ALT_DES_SIZ Alternate Descriptor Size，描述符的大小配置 • 0x0：描述符大小为 4 DWORDS • 0x1：描述符大小为 8 DWORDS
8:4	R/W	0x0	DES_SKIP_LEN Descriptor Skip Length，描述符跳跃长度，此域决定了两个未串联的描述的跳跃长度（这里单位为 Dword，32–bit）。跳跃的长度为当前描述符的结束到下一个描述符的开始。当此域为 0 时，表示描述符为连续的。
3	R/W	0x0	PR_SET Priority SET，优先级的设置 • 0x0：RX DMA 优先级高于 TX DMA • 0x1：TX DMA 优先级高于 RX DMA
2:1	R/W	0x0	WEHT_SET Weight Set，这里设置 DMA 和 RX DMA 的权重比，此域只有加权轮询模式下有效。当 PR_SET = 1 时，这里的加权比为 TX：RX。当 PR_SET = 0 时，这里的优先级比为 RX：TX • 0x0：加权比为 1:1 • 0x1：加权比为 2:10x2：加权比为 3:1 • 0x3：加权比为 4:1
0	R/W	0x0	DMA_ARB_MODE DMA Arbitration Mode，DMA 仲裁模式

默认值：0x00080400			DMA0 配置寄存器 (DMA0 Configure)
位域	类型	默认值	描述
			• 0x0：加权轮询模式，根据加权比例 SET 和优先级 PR_SET 的设定，进行 TX 和 RX 轮询进行 • 0x1：固定优先级模式，只根据优先级 SET，固定优先进行 TX 或 RX

7.2.5.3. 0x00C DMA0_INT_STS

默认值：0x00000000			DMA0 中断状态寄存器 (DMA0 Interrupt Status)
位域	类型	默认值	描述
31:30	–	–	–
29	RO	0x0	TMSMP_INT Timestamp Interrupt，时钟戳中断 此中断出现的情况包括如下： • 系统时间达到或超过所设置的目标时间 • 系统秒时间寄存器溢出 • 辅助时钟捕捉事件发生 注： DMA1 无需此中断状态。
28:26	–	–	–
25:23	RO	0x0	BUS_ERR_TYP Bus Error Type，总线错误类型 • 0x0：在 RxDMA 写数据时出现错误 • 0x3：在 TxDMA 读数据时出现错误 • 0x4：在 RxDMA 描述符 write access 时出现错误 • 0x5：在 TxDMA 描述符 write access 时出现错误 • 0x6：在 RxDMA 描述符 read access 时出现错误 • 0x7：在 TxDMA 描述符 read access 时出现错误 • others：reserved
22:20	RO	0x0	TX_DMA_STS Transmit DMA Process Status，TXDMA 工作状态，这里不会产生中断。

默认值：0x00000000			DMA0 中断状态寄存器 (DMA0 Interrupt Status)
位域	类型	默认值	描述
			• 0x0: 停止，复位或停止发送命令出现 • 0x1： 运行中，获取发送描述符 • 0x2：运行中，等待获取状态 • 0x3：运行中，读取数据、传输至 TXFIFO • 0x4： 时钟戳写状态 • 0x5： Reserved • 0x6： 挂起状态，无效发送描述符或 Buffer 下溢出 • 0x7: 运行中，关闭发送描述符
19:17	RO	0x0	RX_DMA_STS Receive DMA Process Status，RXDMA 工作状态，这里不会产生中断 • 0x0：停止，复位或停止接收命令出现 • 0x1：运行中，获取接收描述符 0x2：Reserved • 0x3：运行中，等待接收包 • 0x4：挂起，无效接收描述符 • 0x5：运行中，关闭接收描述符 • 0x6：时钟戳写状态 • 0x7：运行中，传输数据，从 Buffer 搬运至 DRAM
16	R	0x0	NOR_INT_OR_SGNL Normal Interrupt Or Logic Signal，此位为下面中断信号的 Or 逻辑信号，只有将以下的中断状态清除，才可清除此 BIT 的状态。 • BIT[0]: TX_INT • BIT[2]: TX_BUF_UNAVL_INT • BIT[6]: RX_INT • BIT[14]: ERL_RX_INT
15	R	0x0	ERR_INT_OR_SGNL Error Interrupt Or Logic Signal，此位为下面异常中断信号的 Or 逻辑信号，只有将以下的中断状态清除，才可清除此 BIT 的状态。 • BIT [1]: TX_STP_INT • BIT [3]: TX_JAB_INT • BIT [4]: RX_OF_INT • BIT [5]: TX_UF_INT • BIT [7]: RX_BUF_UNAVL_INT

默认值：0x00000000			DMA0 中断状态寄存器 (DMA0 Interrupt Status)
位域	类型	默认值	描述
			• BIT [8]: RX_STP_INT • BIT [9]: RX_WTD_TO_INT • BIT [10]: ERL_TX _INT • BIT[13]: BUS_ERR_INT
14	R/W1C	0x0	ERL_RX_INT Early Receive Interrupt，提前接收中断，此中断指示 DMA 已将 RXFIFO 接收到的数据包填充到首个数据 buffer。
13	R/W1C	0x0	BUS_ERR_INT Bus Error Interrupt，总线异常错误，此位表示出现 bus 错误，具体错误描述见 bit[25:23]，属于异常中断。
12:11	–	–	–
10	R/W1C	0x0	ERL_TX _INT Early Transmit Interrupt，提前发送中断，此中断指示需要发送的帧已经从数据 buffer 写到 TXFIFO，属于异常中断。
9	R/W1C	0x0	RX_WTD_TO_INT Receive Watchdog Timeout Interrutp，接收看门狗超时中断，此中断指示 Rx 看门狗计时器超时，属于异常中断。
8	R/W1C	0x0	RX_STP_INT Receive Process Stopped Interrupt，接收暂停中断，此中断指示 Rx 过程进入停止状态，属于异常中断。
7	R/W1C	0x0	RX_BUF_UNAVL_INT Receive buffer Unavailable Interrupt，接收缓存无法获取中断，此中断指示 DMA 无法获取下一个接收描述符，属于异常中断。接收进入 Suspend 状态。
6	R/W1C	0x0	RX_INT Receive Interrupt，接收中断，此中断指示完成帧的接收。
5	R/W1C	0x0	TX_UF_INT Transmit Underflow Interrupt，发送下溢出中断，此中断指示在帧发送的过程中出现下溢出，属于异常中断。
4	R/W1C	0x0	RX_OF_INT Receive Overflow Interrupt，接收上溢出中断，此中断指示在帧接收的过程中出现上溢出，属于异常中断。
3	R/W1C	0x0	TX_JAB_INT

默认值：0x00000000			DMA0 中断状态寄存器 (DMA0 Interrupt Status)
位域	类型	默认值	描述
			Transmit Jabber Timeout Interrupt，发送 Jabber 超时中断，此中断指示发送产生 Jabber 计时器超时，发送的帧太大，属于异常中断。
2	R/W1C	0x0	TX_BUF_UNAVL_INT Transmit Buffer Unavailable Interrupt，发送无法获取缓存中断，此中断指示 DMA 无法获取下一个发送描述符，下一个描述由 CPU 占用。发送进入挂起状态。
1	R/W1C	0x0	TX_STP_INT Transmit Process Stopped Interrupt，发送停止中断，此中断指示发送停止，属于异常中断。
0	R/W1C	0x0	TX_INT Transmit Interrupt，发送中断，此中断指示帧发送完成。

7.2.5.4. 0x010 DMA0_INT_EN

默认值：0x00000000			DMA0 中断使能寄存器 (DMA0 Interrupt ，Enable)
位域	类型	默认值	描述
31:30	–	–	–
29	R/W	0x0	TTI_INT_EN Timestamp Interrupt ，Enable，时间戳中断使能 • 0x0：未使能 Timestamp 中断 • 0x1：使能 Timestamp 中断
28:26	–	–	–
25:17	–	–	–
16	R/W	0x0	EN_NOR_INT Enable Normal Interrupt，正常中断使能（其中正常中断具体包含的信号可见 SGNL 的描述），如果未使能此中断信号，那么无法产生正常类型的中断信号 • 0x0：未使能 • 0x1：使能
15	R/W	0x0	EN_ERR_INT Enable Error Interrupt，异常中断使能（其中异常中断具体包含的信号可见 SGNL 的描述），如果未使能此中断信号，那么无法产生异常类型的中断信号

默认值：0x00000000			DMA0 中断使能寄存器 (DMA0 Interrupt ，Enable)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
14	R/W	0x0	EN_ERL_RX_INT Enable Early Receive Interrupt，提前接收中断使能，此使能信号需 EN_NOR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
13	R/W	0x0	EN_BUS_ERR_INT Enable Bus Error Interrupt，总线异常中断使能， 此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
12:11	–	–	–
10	R/W	0x0	EN_ERL_TX _INT Enable Early Transmit Interrupt，提前发送中断使能， 此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	EN_RX_WTD_TO_INT Enable Receive Watchdog Timeout Interrupt，接收看门狗超时中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	EN_RX_STP_INT Enable Receive Process Stopped Interrupt，接收暂停中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	EN_RX_BUF_UNAVL_INT Enable Receive Buffer Unavailable Interrupt，接收缓存无法获取中断使能，此使能信号需 EN_ERR_INT=1 时才有效

默认值：0x00000000			DMA0 中断使能寄存器 (DMA0 Interrupt ，Enable)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
6	R/W	0x0	EN_RX_INT Enable Receive Interrupt，接收中断使能，此使能信号需 EN_NOR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	EN_TX_UF_INT Enable Transmit Underflow Interrupt，发送下溢出中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	EN_RX_OF_INT Enable Receive ，Overflow Interrupt，发送上溢出中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	EN_TX_JAB_INT Enable Transmit Jabber Timeout Interrupt，发送 Jabber 超时中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	EN_TX_BUF_UNAVL_INT Enable Transmit Buffer Unavailable Interrupt，发送缓存无法获取中断，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	EN_TX_STP_INT Enable Transmit Sopped Interrupt，发送暂停中断使能，此使能信号需 EN_ERR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			DMA0 中断使能寄存器 (DMA0 Interrupt ，Enable)
位域	类型	默认值	描述
0	R/W	0x0	EN_TX_INT Enable Transmit Interrupt，发送中断使能，此使能信号需 EN_NOR_INT=1 时才有效 • 0x0：未使能 • 0x1：使能

7.2.5.5. 0x01C MAC_TX_FUNC

默认值：0x00000000			MAC 发送功能寄存器 (MAC Transmit Function)
位域	类型	默认值	描述
31:15	–	–	–
14:12	R/W	0x0	TX_FR_SRC_ADR_SET TX Frame Source Address Set，TX 帧的源地址配置方式。 Bit14 • 0x0：使用 Addr0 寄存器作为源地址插入或替代 • 0x1：使用 Addr1 寄存器作为源地址插入或替代 BIT[13:12] • 0x0/0x1：Reserved • 0x2：插入源地址 • 0x3：替代源地址
11	–	–	–
10	R/W	0x0	EN_DEF_CHECK Enable Deferral Check，使能回退时间检测功能，该位只在半双工模式下有效 • 0x0：未使能 • 0x1：使能，对发送帧进行回退时间的检测
9	R/W	0x0	DIS_JAB Disable Jabber，禁止 Jabber，禁止后可以发送最多 16383 字节大小的帧 • 0x0：未禁止，发送帧为正常大小 • 0x1：禁止 Jabber
8	–	–	–
7:5	R/W	0x0	IFR_GAP Inter-Frame Gap，相邻帧的间隔时间配置

默认值：0x00000000			MAC 发送功能寄存器 (MAC Transmit Function)
位域	类型	默认值	描述
			• 0x0：96 bit times • 0x1：88 bit times • 0x2：80 bit times • … • 0x7：40 bit times
4	R/W	0x0	DIS_CRS Disable Carrier Sense During Transmission，在发送过程中禁止载波侦听，此位适合用在半双工模式下 • 0x0：未禁止 • 0x1：禁止载波侦听
3	R/W	0x0	DIS_RETRY Disable Retry，禁止重发功能，发送只进行一次的发送 • 0x0：未禁止 • 0x1：禁止重发
2:1	R/W	0x0	BK_OFF Back-Off Value，发送碰撞后回退值，此位只适合工作在半双工模式 • 0x0：k= min(n,10) • 0x1：k= min(n,8) • 0x2：k= min(n,4) • 0x3：k= min(n,1) 这里 n 为重传次数。
0	R/W	0x0	EN_TX Enable Transmit，使能 MAC 进行发送帧 • 0x0：未使能 • 0x1：使能

7.2.5.6. 0x020 MAC_RX_FUNC

默认值：0x00000000			MAC 接收功能寄存器 (MAC Receive Function)
位域	类型	默认值	描述
31:23	–	–	–

默认值：0x00000000			MAC 接收功能寄存器 (MAC Receive Function)
位域	类型	默认值	描述
22	R/W	0x0	EN_WD_TH Enable Watchdog Timeout Threshold，使能看门狗的超时阈值 • 0x0：未使能 • 0x1：使能，若看门狗未关闭，此时看门狗则依据 TH 的阈值进行工作。
21:8	R/W	0x0	WD_TH Watchdog Timeout Threshold，接收看门狗的超时阈值，当接收帧的长度大于这里的看门狗超时阈值，那么此帧终止并且报错。
7	–	–	–
6	R/W	0x0	DIS_WD Disable Watchdog，关闭接收看门狗 • 0x0：未关闭，最多只能接收 2048bytes（如果 EN_RX_JUMBO = 1，则最多接收 10240bytes） • 0x1：关闭看门狗，MAC 可以接收最大 16383bytes 的帧
5	R/W	0x0	EN_2K_FR Enable Receive 2000 bytes Frame，使能接收长度为 2000bytes 的帧 • 0x0：未使能 • 0x1：使能，长度为 2000bytes 的帧认为是正常的帧
4	R/W	0x0	EN_JUMBO Enable Jumbo Frame，使能巨型帧 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	DIS_RX_OWN Disable Receive Own，关闭自接收功能，此功能只适合用与半双工模式 • 0x0：未禁止，在半双工模式下能够接收所有的帧 • 0x1：禁止，在半双工模式下不接收帧
2	R/W	0x0	EN_STRIP_FCS Enable Strip FCS Field，使能对接收帧（Length/Type 域≥1536 的帧）去除 FCS 域的值 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	EN_STRIP_NML

默认值：0x00000000			MAC 接收功能寄存器 (MAC Receive Function)
位域	类型	默认值	描述
			Enable Strip Normal Frame，使能对接收帧（Length/Type 域<1536 的帧）自动去除 PAD 或 FCS 域的值 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	EN_RX Enable Receive，使能 MAC 进行接收帧 • 0x0：未使能 • 0x1：使能

7.2.5.7. 0x024 TX_DMA0_CTL

默认值：0x00000000			发送 DMA0 控制寄存器 (Transmit DMA0 Control)
位域	类型	默认值	描述
31:8	–	–	–
7	R/WAC	0x0	EN_TX_POLL Enable TX Poll，使能发送继续，DMA 读取当前的发送 DMA 描述符继续进行数据的发送。DMA 执行继续操作完毕后，自动清 0 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	EN_SEC_FR_TX Enabel Second Frame TX Mode，使能第二帧发送功能 • 0x0：未使能 • 0x1：使能，不管第一帧的发送情况，TX DMA 直接处理第二帧的数据
5	R/W	0x0	TX_MOD_SEL Transmit Mode Select，发送模式选择 • 0x0：阈值模式，根据 TH 阈值进行发送 • 0x1：存储模式，TXFIFO 存储完整的 Frame 才进行发送
4	R/W	0x0	C_TXFIFO Clear Transmit FIFO，清空发送 FIFO，写 1 进行 FIFO 清除，完成后此位自动清 0。
3:1	R/W	0x0	TX_TH

默认值：0x00000000			发送 DMA0 控制寄存器 (Transmit DMA0 Control)
位域	类型	默认值	描述
			Transmit Threshold，DMA 发送阈值，TX FIFO 数据高于此阈值则进行发送 • 0x0：64 • 0x1：128 • 0x2：192 • 0x3：256 • 0x4：40 • 0x5：32 • 0x6：24 • 0x7：16
0	R/W	0x0	EN_TX_DMA Enable DMA Transmit，使能 DMA 发送 • 0x0：停止 DMA 发送数据 • 0x1：启动 DMA 发送数据

7.2.5.8. 0x028 RX_DMA0_CTL

默认值：0x00000000			接收 DMA0 控制寄存器 (Receive DMA0 Control)
位域	类型	默认值	描述
31:24	–	–	–
23:16	R/W	0x0	RIWT（针对帧的数量的看门狗） RI Watchdog Timer Count，接收看门狗计数。
15:10	–	–	–
9	R/WAC	0x0	EN_RX_POLL Enable RX Poll，使能接收继续，DMA 读取当前的接收 DMA 描述符继续进行数据的接收。DMA 执行继续操作完毕后，自动清 0 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	DIS_DR_CS_ERR_FR Disable Dropping of TCP/IP Checksum Error Frames，禁止丢弃由于 TCP/IP 校验和错误的帧

默认值：0x00000000			接收 DMA0 控制寄存器 (Receive DMA0 Control)
位域	类型	默认值	描述
			• 0x0：未禁止 • 0x1：禁止
7	R/W	0x0	DIS_CL_RX_FR Disable Clear Received Frames，禁止清除接收帧，由于接收中断后重新进行接收，正常情况会对中断前的帧进行清除 • 0x0：未禁止 • 0x1：禁止清除 RX 帧
6	R/W	0x0	EN_KEEP_ERR_FR Enable Keep Error Frames，使能保留错误的帧 • 0x0：未使能，丢弃出现错误状态的帧 • 0x1：使能
5	R/W	0x0	EN_KEEP_UNSIZ_FR Enable Keep Undersized Good Frames，使能保留无错误的小帧（长度小于 64bytes） • 0x0：未使能，丢弃小帧 • 0x1：使能，另外若 TH 设置低于 64bytes 的阈值，那么此位无作用。
4	R/W	0x0	EN_DR_GI_FR Enable Drop Giant Frames，使能丢弃巨型帧 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	RX_MOD_SEL Receive Mode Select，接收模式选择 • 0x0：阈值模式，根据 TH 阈值进行接收 • 0x1：存储模式，RXFIFO 存储完整的 Frame 才进行接收
2:1	R/W	0x0	RX_TH Receive Threshold，MA 接收阈值，RX FIFO 数据高于此阈值则进行接收 • 0x0：64 • 0x1：32 • 0x2：96 • 0x3：128
0	R/W	0x0	EN_RX_DMA

默认值：0x00000000			接收 DMA0 控制寄存器 (Receive DMA0 Control)
位域	类型	默认值	描述
			Enable DMA Receive，使能 DMA 接收 • 0x0：停止 DMA 接收数据 • 0x1：启动 DMA 接收数据

7.2.5.9. 0x034 FLOW_CTL

默认值：0x00000000			流控寄存器 (Flow Control)
位域	类型	默认值	描述
31:16	R/W	0x0	PAU_TIM_SE Pause Time Set，设置 Pause 帧的 Pause Time 域的值。
15:8	–	–	–
7	R/W	0x0	DIS_ZQ_PAU Disable Zero-Quanta Pause，禁止 0 值 Pause 帧的生成 • 0x0：未使能 • 0x1：使能，禁止自动生成 Pause 帧
6	–	–	–
5:4	R/W	0x0	PAU_LO_TH Pause Low Threshold 此域用于配置 Pause 帧重发的计时器的阈值。
3	R/W	0x0	EN_UNI_PAUSE Enable Unicast Pause Frame Detect，使能单播地址的 Pauseframe 检测 • 0x0：MAC 只检测唯一的多播地址的 Pause 帧 • 0x1：MAC 除了检测多播地址的 Pause 帧外，支持检测单播地址的 Pause 帧，单播地址为 MAC ADDR0
2	R/W	0x0	EN_RX_FLOW Enable Receive Flow Control，使能接收流控，只适用于全双工模式下 • 0x0：未使能 • 0x1：使能，MAC 解析接收到 Pause 帧，禁止特定的时间内发送 Pause 帧
1	R/W	0x0	AL_TX_FLOW Allow Transmit Flow Control，使能允许发送的流控(全双工)/背压(半双工)

默认值：0x00000000			流控寄存器 (Flow Control)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能，MAC 允许发送流控(全双工)/背压(半双工)
0	R/W or R/WAC	0x0	EN_TX_FLOW Enable Transmit Flow Control，使能发送流控，全双工模式下，发送 Frame，半双工模式下，设置背压功能。在 EN_TX_FLOW = 1 允许发送流控时，此位用于初始化 Frame（全双工）或背压模式（半双工） • 0x0：未使能 • 0x1：使能，初始化 Frame（全双工）或背压模式（半双工）。另外，在 Frame 模式下，此位写 1 后完成 Frame 后自动清 0。

7.2.5.10. 0x038 VLAN_TAG

默认值：0x00000000			VLAN 标签寄存器 (VLAN TAG)
位域	类型	默认值	描述
31:20	–	–	–
19	R/W	0x0	CSVLC–VLAN or S–LAN. • 0x0: C–VLAN type (0x8100) is inserted or replaced in thetransmitted frames. • 0x1: S–VLAN type (0x88A8) is inserted or replaced in the13th and 14th bytes of transmitted frames.
18	R/W	0x0	VLP VLAN Priority Control. When this bit is set, the control Bits [17:16] are usedfor VLAN deletion, insertion, or replacement. When thisbit is reset, the mti_vlan_ctrl_i control input is used,and Bits [17:16] are ignored.
17:16	R/W	0x0	VLC VLAN Tag Control in Transmit Frames • 0x0：无 VLAN 标签的删除、插入或替代 • 0x1：VLAN 标签的删除 0x2：VLAN 标签的插入 • 0x3：VLAN 标签的替代
15:0	R/W	0x0	VLT VLAN Tag for Transmit Frames，此域包含用于插入或替代的 VLAN 标签。

默认值：0x00000000			VLAN 标签寄存器 (VLAN TAG)
位域	类型	默认值	描述
			• BIT[15:13]：用户优先级 • BIT[12]：Canonical Format Indicator(CFI) or Drop Eligible Indicator (DEI) • BIT[11:0]：VLAN 标签的 VID 域

7.2.5.11. 0x03C VLAN_FLT

默认值：0x00000000			VLAN 过滤寄存器 (VLAN Filter)
位域	类型	默认值	描述
31:20	–	–	–
19	R/W	0x0	EN_VL_HASH_TBL Enable VLAN Tag Hash Table Match • 0x0：未使能 • 0x1：使能，VLAN 标签的哈希列表过滤
18	R/W	0x0	EN_SVL Enable S-VLAN，使能接收 VLAN 帧 • 0x0：未使能 • 0x1：使能，MAC 对接收和发送的 type=0x88A8 的帧认为 VLAN 标签帧
17	R/W	0x0	EN_INV_TAG Enable Inverse VLAN TAG Match，使能反向的 TAG 匹配 • 0x0：未使能 • 0x1：使能，VLAN 标签反向匹配使能，即不符合 VLAN 标签的认为匹配。
16	R/W	0x0	EN_VID Enable 12-bit VLAN VID Comparison • 0x0：未使能 • 0x1：使能，只采用 12bit 的 VID 用于比较和过滤
15:0	R/W	0x0	VL_TAG_ID VLAN Tag Identifier for Receive Frames，虚拟网络标签的识别符 • BIT[15:13]：用户优先级 • BIT[12]：规范格式指示符

默认值：0x00000000			VLAN 过滤寄存器 (VLAN Filter)
位域	类型	默认值	描述
			• BIT[11:0]：VLAN 标签的 VID 域 • 在 Bit[16] ETV = 1，只有 VLAN 的 VID 域用于比较。

7.2.5.12. 0x040 MAC_FRM_FLT

默认值：0x00000000			MAC 帧过滤寄存器 (MAC Frame Filter)
位域	类型	默认值	描述
31	R/W	0x0	EN_RX_WI_STS Enable RX All Frame without RX Status，使能接收所有的帧且保留过滤的状态 • 0x0：MAC 接收经过 SA、DA 地址滤波的帧 • 0x1：MAC 接收所有的帧，不经过地址过滤
30:17	–	–	–
16	R/W	0x0	EN_VL_FLT Enable VLAN Tag Filter，使能 VLAN 标签的过滤器 • 0x0：未使能 • 0x1：使能，使能 MAC 的 Filter，丢弃 Tag 不匹配的帧。
15:11	–	–	–
10	R/W	0x0	EN_FR_FLT Enable Frame Filter，使能帧过滤器，进行完美滤波或哈希滤波 • 0x0：未使能 • 0x1：使能，地址滤波器允许通过符合完美滤波或哈希滤波的帧。
9	R/W	0x0	EN_SRC_ADDR_FLT Enable Source Address Filter，使能源地址的滤波器 • 0x0：未使能 • 0x1：使能，MAC 比对接收到的帧的源地址是否域使能的源地址寄存器一致，若不一致则丢弃
8	R/W	0x0	EN_INV_SA_FLT Enable Inverse SA Filtering，使能反向的源地址滤波

默认值：0x00000000			MAC 帧过滤寄存器 (MAC Frame Filter)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能，MAC 比对接收到的帧的源地址是否与使能的源地址寄存器一致，若一致则丢弃
7:6	R/W	0x0	CTL_FR_FLT_MOD Control Frames Fliter Mode，控制帧的过滤方式 • 0x0：MAC 过滤所有的控制帧 • 0x1：MAC 通过所有控制帧，除了 Pause 帧 0x2：MAC 通过所有控制帧 • 0x3：MAC 通过所有经过地址滤波器滤除的控制帧
5	R/W	0x0	DIS_BRD_FRM Disable Broadcast Frames，禁止广播帧 • 0x0：未使能，Address Filter Module 接收所有收到的广播帧 • 0x1：使能，Address Filter Module 阻止所有收到的广播帧
4	R/W	0x0	PAS_ALL_MUL Pass All Multicast，通过所有的多播帧 • 0x0：未使能，滤除所有的多播帧 • 0x1：使能，通过所有的多播帧
3	R/W	0x0	EN_INV_DA_FLT Enable Inverse DA Filter，使能反向的目的地址过滤 • 0x0：未使能 • 0x1：使能，MAC 比对接收到的帧的目的地址是否与使能的目的地址寄存器一致，若一致则丢弃
2	R/W	0x0	EN_HAS_MUL Enable Hash Multicast，使能哈希多播过滤 • 0x0：未使能 • 0x1：使能，MAC 对接收的多播帧进行哈希列表的目的地址滤除
1	R/W	0x0	EN_HAS_UNI Enable Hash Unicast，使能哈希单播过滤 • 0x0：未使能 • 0x1：使能，MAC 对接收的单播帧进行哈希列表的目的地址滤除
0	R/W	0x0	EN_RX_WO_STS Enable RX All Frame without RX Status，使能接收所有的帧且不保留过滤的状态

默认值：0x00000000			MAC 帧过滤寄存器 (MAC Frame Filter)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能，Address Filter Module 通过所有接收的帧，不管目的地址或源地址的过滤。此时源地址或目的地址的过滤失败状态会自动清除

7.2.5.13. 0x044 HASH_TBL_HIGH

默认值：0x00000000			高位哈希列表寄存器 (HASH Table High)
位域	类型	默认值	描述
31:0	RW	0x0	HAS_TBL_HI Hash Table High 此域为哈希表的高 32 位。

7.2.5.14. 0x048 HASH_TBL_LOW

默认值：0x00000000			低位哈希列表寄存器 (HASH Table Low)
位域	类型	默认值	描述
31:0	RW	0x0	HAS_TBL_LO Hash Table LOW 此域为哈希表的低 32 位。

7.2.5.15. 0x050 MAC_ADDR0_HIGH

默认值：0x8000FFFF			高位 MAC 地址 0 寄存器 (MAC Address0 high)
位域	类型	默认值	描述
31	RO	0x1	EN_ADDR Enable Address，此位总是 1，Address0 总是使能。
30:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.16. 0x054 MAC_ADDR0_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 0 寄存器 (MAC Address0 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.17. 0x058 MAC_ADDR1_HIGH

默认值：0x0000FFFF			高位 MAC 地址 1 寄存器 (MAC Address1 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDR Enable Address，地址使能 • 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤
30	R/W	0x0	SRC_ADDR_CMP Source Address Compare，是否源地址比对 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.18. 0x05C MAC_ADDR1_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 1 寄存器 (MAC Address1 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.19. 0x060 MAC_ADDR2_HIGH

默认值：0x0000FFFF			高位 MAC 地址 2 寄存器 (MAC Address2 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDR Enable Address，地址使能 • 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤
30	R/W	0x0	SRC_ADDR_CMP Source Address Compare，是否源地址比对 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.20. 0x064 MAC_ADDR2_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 2 寄存器 (MAC Address2 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.21. 0x068 MAC_ADDR3_HIGH

默认值：0x0000FFFF			高位 MAC 地址 3 寄存器 (MAC Address3 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDR Enable Address，地址使能 • 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤

默认值：0x0000FFFF			高位 MAC 地址 3 寄存器 (MAC Address3 high)
位域	类型	默认值	描述
30	R/W	0x0	SRC_ADDR_CMP Source Address Compare，是否源地址比对 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.22. 0x06C MAC_ADDR3_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 3 寄存器 (MAC Address3 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.23. 0x074 MAC_ADDR4_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 4 寄存器 (MAC Address4 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.24. 0x078 MAC_ADDR5_HIGH

默认值：0x0000FFFF			高位 MAC 地址 5 寄存器 (MAC Address5 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDR Enable Address，地址使能

默认值：0x0000FFFF			高位 MAC 地址 5 寄存器 (MAC Address5 high)
位域	类型	默认值	描述
			• 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤
30	R/W	0x0	SRC_ADDR_CMP Source Address Compare，是否源地址比对 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.25. 0x07C MAC_ADDR5_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 5 寄存器 (MAC Address5 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.26. 0x080 MAC_ADDR6_HIGH

默认值：0x0000FFFF			高位 MAC 地址 6 寄存器 (MAC Address6 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDR Enable Address，地址使能 • 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤
30	R/W	0x0	SRC_ADDR_CMP Source Address Compare，是否源地址比对 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对

默认值：0x0000FFFF			高位 MAC 地址 6 寄存器 (MAC Address6 high)
位域	类型	默认值	描述
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HI MAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.27. 0x084 MAC_ADDR6_LOW

默认值：0xFFFFFFFF			低位 MAC 地址 6 寄存器 (MAC Address6 low)
位域	类型	默认值	描述
31:0	R/W	0xFFFFFFFF	ADDR_LO MAC Address Low，低 MAC 地址，MAC Address0[31:0]，此域包含 MAC 地址的低 32–bits。

7.2.5.28. 0x088 MAC_ADDR7_HIGH

默认值：0x0000FFFF			高位 MAC 地址 7 寄存器 (MAC Address7 high)
位域	类型	默认值	描述
31	R/W	0x0	EN_ADDREnable Address，地址使能。 • 0x0：地址过滤模块无视此地址 • 0x1：地址过滤模块使用此地址进行完美过滤
30	R/W	0x0	SRC_ADDR_CMPSource Address Compare，是否源地址比对。 • 0x0：采用此地址对接收的帧的目的地址进行比对 • 0x1：采用此地址对接收的帧的源地址进行比对
29:16	–	–	–
15:0	R/W	0xFFFF	ADDR_HIMAC Address High，高 MAC 地址，MAC Address0[47:32]，此域包含 MAC 地址的高 16–bits。

7.2.5.29. 0x090 MDIO_CTL

默认值：0x00000000			MDIO 控制寄存器 (MDIO Control)
位域	类型	默认值	描述
31:16	–	–	–
15:11	R/W	0x0	PHY_ADDR Physical Layer Address，PHY 的地址。
10:6	R/W	0x0	PHY_REG_ADDR PHY Register Address，PHY 寄存器地址。
5:2	R/W	0x0	MDCIO_INT_CLK_SEL MDCIO Internal Clock Select，MDCIO 的内部时钟选择，根据 AHB 总线的时钟进行选择 • 0x0：60–100MHz • 0x1：100–150MHz • 0x2：20–35MHz • 0x3：35–60MHz • 0x4：150–250MHz • 0x5：50–300MHz • 0x6：2000 分频 • 0x7：4000 分频 当 Bit5 = 1，支持更高的 MDC 时钟，一般不做使用。
1	R/W	0x0	MDCIO_RW MDCIO Read/Write，MDCIO 接口进行读写操作选择 • 0x0：读操作 • 0x1：写操作
0	R/WAC	0x0	MDCIO_STS MDCIO Status，MDCIO 状态 • 0x0：空闲 • 0x1：繁忙

7.2.5.30. 0x094 MDIO_DATA

默认值：0x00000000			MDIO 数据寄存器 (MDIO Data)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	MDCIO_RW_DATA MDCIO Read/Write Data，MDCIO 完成的读写数据。

7.2.5.31. 0x0A0 DEBUG_REG

默认值：0x00000000			Debug 寄存器 (Debug)
位域	类型	默认值	描述
31:26	–	–	–
25	RO	0x0	TXSTSFSTSMTL TxStatus FIFO Full Status
24	RO	0x0	TXFSTSMTL Tx FIFO Not Empty Status
23	RO	0x0	/
22	RO	0x0	TWCSTSMTL Tx FIFO Write Controller Status
21:20	RO	0x0	TRCSTSMTL Tx FIFO Read Controller Status
19	RO	0x0	TXPAUSED MAC Transmitter in Pause
18:17	RO	0x0	TFCSTS MAC Transmit Frame Controller Status
16	RO	0x0	TPESTS MAC GMII or MII Transmit Protocol Engine Status
15:10	–	–	–
9:8	RO	0x0	RXFSTSMTL RxFIFO Fill–Level Status

默认值：0x00000000			Debug 寄存器 (Debug)
位域	类型	默认值	描述
7	–	–	–
6:5	RO	0x0	RRCSTSMTL RxFIFO Read Controller State
4	RO	0x0	RWCSTSMTL Rx FIFO Write Controller Active Status
3	–	–	–
2:1	RO	0x0	RFCFCSTS MAC Receive Frame FIFO Controller Status
0	RO	0x0	RPESTS MAC GMII or MII Receive Protocol Engine Status

7.2.5.32. 0x0B0 TXDMA0_DESC_START

默认值：0x00000000			TXDMA0 起始描述符地址寄存器 (TX DMA0 Descriptor Start)
位域	类型	默认值	描述
31:0	RW	0x0	TX_DMA_STR Start of Transmit DMA List，接收 DMA 描述符列表的第一个描述符的基地址。

7.2.5.33. 0x0B4 RXDMA0_DESC_START

默认值：0x00000000			RXDMA0 起始描述符地址寄存器 (RX DMA0 Descriptor Start)
位域	类型	默认值	描述
31:0	RW	0x0	RX_DMA_STR Start of Receive DMA List，接收 DMA 描述符列表的第一个描述符的基地址。

7.2.5.34. 0x0B8 CUR_TXDMA0_DESC

默认值：0x00000000			TXDMA0 当前描述符地址寄存器 (TX DMA0 Descriptor Current)
位域	类型	默认值	描述
31:0	RO	0x0	CUR_TX_DMA_ADDR Current Transmit DMA Descriptor Address Pointer，当前处理的发送描述符的地址指针。

7.2.5.35. 0x0BC CUR_RXDMA0_DESC

默认值：0x00000000			RXDMA0 当前描述符地址寄存器 (RX DMA0 Descriptor Current)
位域	类型	默认值	描述
31:0	RO	0x0	CUR_RX_DMA_ADDR Current Receive DMA Descriptor Address Pointer，当前处理的发送描述符的地址指针。

7.2.5.36. 0x0C0 CUR_TXDMA0_BUFADDR

默认值：0x00000000			TXDMA0 缓存地址寄存器 (TX DMA0 Buffer)
位域	类型	默认值	描述
31:0	RO	0x0	CUR_TX_BUF_ADDR Current Transmit DMA Buffer Address Pointer，当前处理的发送描述符的缓存地址指针。

7.2.5.37. 0x0C4 CUR_RXDMA0_BUFADDR

默认值：0x00000000			RXDMA0 缓存地址寄存器 (RX DMA0 Buffer)
位域	类型	默认值	描述
31:0	RO	0x0	CUR_RX_BUF_ADDR Current Receive DMA Buffer Address Pointer，当前处理的接收描述符的缓存地址指针。

7.2.5.38. 0x0E8 DMA0_MISFRM_BUFOVF

默认值：0x00000000			DMA0 丢帧溢出寄存器 (DMA0 Missframe Buffer Overflow)
位域	类型	默认值	描述
31:29	–	–	–

默认值：0x00000000			DMA0 丢帧溢出寄存器 (DMA0 Missframe Buffer Overflow)
位域	类型	默认值	描述
28	R/SS/RC	0x0	OVFCNTOVF Overflow Bit for FIFO，Overflow Counter • 0x0：计数器未溢出 • 0x1：计数器溢出
27:17	R/SS/RC	0x0	OVFFRMCNT Overflow Frame Counter，此计数器记录在应用过程中丢失的帧的数量。
16	R/SS/RC	0x0	MISCNTOVF Overflow Bit for Missed Frame Counter • 0x0：计数器未溢出 • 0x1：计数器溢出
15:0	R/SS/RC	0x0	MISFRMCNT Missed Frame Counter，此计数器记录由于接收 buffer 不可用导致的丢失帧的数量。

7.2.5.39. 0x100 TMSTMP_CTL

默认值：0x00002000			时钟戳控制寄存器 (Timestamp Contorl)
位域	类型	默认值	描述
31:26	–	–	–
25	R/W	0x0	EN_AUX_SNAP0 Enable Auxiliary Channel0 Snapshot，使能通道 0 辅助捕捉的功能 • 0x0：未使能 • 0x1：使能，ptp_aux_trig_i[0]输入触发时钟戳捕捉
24	R/W	0x0	SNAP_Clear Auxiliary Snapshot FIFO Clear，对 FIFO 的指针复位，FIFO 数据也相应清除 • 0x0：未使能 • 0x1：使能，复位 FIFO 的指针
23:19	–	–	–
18	R/W	0x0	EN_PTP_FLT Enable MAC address for PTP Frame Filtering，使能 PTP 帧过滤

默认值：0x00002000			时钟戳控制寄存器 (Timestamp Contorl)
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能，接收的帧的目的地址符合任意的 MAC 地址寄存器则过滤 PTP 帧
17:16	R/W	0x0	SNAP_SEL Snap Select，捕捉 PTP 报文的类型选择，需配合 MESGE 和 EN_SNAP_EVENT 共同选择。
15	R/W	0x0	EN_SNAP_MESGE Enable Snapshot for Messages Relevant to Master，使能对主节点相关的报文进行时间戳的捕捉 • 0x0：未使能，只对从节点相关的报文进行时间戳的捕捉 • 0x1：使能，只对主节点相关的报文进行时间戳的捕捉
14	R/W	0x0	EN_SNAP_EVENT Enable Timestamp Snapshot for Event Messages，使能事件报文进行时钟戳捕捉 • 0x0：未使能，时钟戳对除了 Announce、Management、Signaling 的所有报文进行捕捉 • 0x1：使能，时钟戳只针对事件报文捕捉
13	R/W	0x1	EN_PTP_IP4 Enable Processing of PTP Frames Sent over IPv4–UDP，使能处理包含 PTP 的 IPV4 UDP 帧 • 0x0：未使能 • 0x1：使能，MAC 接收机处理封装在 UDP 数包的 PTP 数据包
12	R/W	0x0	EN_PTP_IP6 Enable Processing of PTP Frames Sent over IPv6–UDP，使能处理包含 PTP 的 IPV6 UDP 帧 • 0x0：未使能 • 0x1：使能，MAC 接收机处理封装在 UDP 数包的 PTP 数据包
11	R/W	0x0	EN_ETH_FR Enable Processing of PTP over Ethernet Frames，使能处理包含 PTP 的 Ethernet 帧 • 0x0：未使能 • 0x1：使能，MAC 接收机处理封装在以太网帧的 PTP 数据包
10	R/W	0x0	EN_PTP_V2 Enable PTP packet Processing for Version 2 Format，使能处理 PTPV2 格式包 • 0x0：未使能 • 0x1：使能，采用 Version2 的格式处理 PTP 数据包

默认值：0x00002000			时钟戳控制寄存器 (Timestamp Contorl)
位域	类型	默认值	描述
9	R/W	0x0	NANO_SEC_REG_SET Timestamp Nano Second Register Set，设置 Register 的单位 bit 所代表的精度 • 0x0：0.46ns，Nano Second 寄存器最大为 0x7FFF_FFFF • 0x1：1ns，Nano Second 寄存器最大为 0x3B9A_C9FF
8	R/W	0x0	EN_RX_ALL_TMSMP Enable Timestamp for All Received Frames，使能对所有接收的帧的时钟捕捉 • 0x0：未使能 • 0x1：使能，MAC 能够对接收到的所有帧进行时钟戳捕捉
7:6	–	–	–
5	R/WAC	0x0	UPDT_ADD Update Addend，更新加数寄存器的值 • 0x0：未使能 • 0x1：使能，更新时钟戳加数寄存器，当更新完成，自动清 0
4	R/WAC	0x0	EN_TMSMP_TRI_INT Enable Timestamp Trigger Interrupt，使能时钟戳的触发中断 • 0x0：未使能 • 0x1：使能，在系统时间大于目标时间寄存器值时，触发中断
3	R/WAC	0x0	UPDT_TMSMP Timestamp Update，时间戳更新 • 0x0：未使能 • 0x1：使能，系统时钟利用 Update 值进行更新，完成后自动清 0
2	R/WAC	0x0	INIT_TMSMP Initialize Timestamp，初始化时钟戳 • 0x0：未使能 • 0x1：使能，系统时钟利用 Update 值进行初始化，完成后自动清 0
1	R/W	0x0	TTMSMP_UPT_MOD Timestamp Update Mode，时钟戳更新的方式选择 • 0x0：时钟戳以粗糙的方式更新 • 0x1：时钟戳以精细的方式更新

默认值：0x00002000			时钟戳控制寄存器 (Timestamp Contorl)
位域	类型	默认值	描述
0	R/W	0x0	EN_TMSMP Enable Timestamp，使能时钟戳功能 • 0x0：未使能 • 0x1：使能，时钟戳添加到接收和发送帧

7.2.5.40. 0x0104 SUB_SEC_INCR

默认值：0x00000000			亚秒递增寄存器 (Sub Second Increment)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	SUB_SEC_INC Sub-second Increment Register，亚秒递增寄存器。

7.2.5.41. 0x0108 TMSMP_ADDEND

默认值：0x00000000			时钟戳加数寄存器 (Timestamp Addend)
位域	类型	默认值	描述
31:0	R/W	0x0	ADDEND Timestamp Addend Register，时钟戳加数寄存器。

7.2.5.42. 0x010C SYS_TIME_SEC

默认值：0x00000000			时间戳秒寄存器 (Timestamp Second)
位域	类型	默认值	描述
31:0	RO	0x0	TMSMP_SEC Timestamp Second Register，时间戳秒寄存器。

7.2.5.43. 0x0110 SYS_TIME_NANO_SEC

默认值：0x00000000			时间戳亚秒寄存器 (Timestamp Sub Second Register)
位域	类型	默认值	描述
31	–	–	–
30:0	RO	0x0	TMSMP_SUB_SEC Timestamp Sub Second Register，时间戳亚秒寄存器。

7.2.5.44. 0x0114 UPDT_TIME_SEC

默认值：0x00000000			时间戳秒更新寄存器 (Update Timestamp Second)
位域	类型	默认值	描述
31:0	R/W	0x0	UPT_SEC Update Timestamp Second Register，时间戳秒更新寄存器。

7.2.5.45. 0x0118 UPDT_TIME_NANO_SEC

默认值：0x00000000			时间戳亚秒更新寄存器 (Update Timestamp Sub Second)
位域	类型	默认值	描述
31	R/W	0x0	UPT_SIGN Update Time Sign Bit，时间更新的符号位，对时间进行加法还是减法 • 0x0：时钟对更新寄存器的值做加法，即时钟加上更新寄存器的值。 • 0x1：时钟对更新寄存器的值做减法，即时钟减去更新寄存器的值。
30:0	R/W	0x0	UPT_SUB_SEC Update Timestamp Sub Second Register，时间戳亚秒更新寄存器

7.2.5.46. 0x011C TRGT_TIME_SEC

默认值：0x00000000			目标时间戳秒寄存器 (Target Time Second)
位域	类型	默认值	描述
31:0	R/W	0x0	TRG_SEC Target Time Second Register，目标时间戳秒寄存器。

7.2.5.47. 0x0120 TRGT_TIME_NANO_SEC

默认值：0x00000000			目标时间戳亚秒寄存器 (Target Time Sub Second)
位域	类型	默认值	描述
31	R/WAC	0x0	TRG_BUSY Target Time Register Busy，目标时钟寄存器繁忙状态指示，当此位指示繁忙时，不要进行目标时钟的更新。
30:0	R/W	0x0	TRG_NANO_SEC Target Timestamp Nano Second Register，目标时钟戳 NanoSecond 寄存器。

7.2.5.48. 0x0124 AUX_TMSMP_SEC

默认值：0x00000000			捕捉时钟秒寄存器 (Auxiliary Timestamp Second)
位域	类型	默认值	描述
31:0	RO	0x0	AUX_SEC Auxiliary Timestamp Second Register，输入触发捕捉到的时钟戳的 Second 寄存器。

7.2.5.49. 0x0128 AUX_TMSMP_NANO_SEC

默认值：0x00000000			捕捉时钟亚秒寄存器 (Auxiliary Timestamp Sub Second)
位域	类型	默认值	描述
31	–	–	–
30:0	RO	0x0	AUX_NANO_SEC Auxiliary Timestamp Nano Second Register，输入触发捕捉到的时钟戳的 Second 寄存器。

7.2.5.50. 0x012C SYS_TIME_HI_WORD_SEC

默认值：0x00000000			高位时钟戳寄存器 (Timestamp Higher Word)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	TMSMP_HI_SEC Timestamp Higher Word Register，时钟的高 16bit 时钟戳寄存器。

7.2.5.51. 0x0130 TMSMP_STS

默认值：0x00000000			时钟戳状态寄存器 (Timestamp Status)
位域	类型	默认值	描述
31:30	–	–	–
29:25	RO	0x0	AUX_SNAP_NUM Auxiliary Timestamp Snapshots Number，辅助时钟戳捕捉的次数，最大允许捕捉次数为 4。另外对 Clear 写 1，此域清 0。
24	RO	0x0	AUX_SNAP_ERR Auxiliary Timestamp Snapshot Error，辅助时钟戳捕捉 FIFO 已满却出现捕捉触发错误，此时则错过对此时钟戳捕捉 • 0x0：未丢失 • 0x1：丢失，表示 FIFO 满的情况下，存在时钟戳捕捉请求
23:4	–	–	–
3	R/SS/RC	0x0	TRG_TIME_ERR Timestamp Target Time Error，目标时间出错 • 0x0：未写入 • 0x1：配置的目标时间出错，当前时间已超过目标时间，此位读取后自动清 0
2	R/SS/RC	0x0	AUX_SNAP Auxiliary Timestamp Trigger Snapshot，辅助时钟戳捕捉发生 • 0x0：未写入 • 0x1：Auxiliary Snapshot 写入 FIFO，此位读取后自动清 0
1	R/SS/RC	0x0	TRG_REACH Timestamp Target Time Reached，目标时钟达到 • 0x0：未达到目标值 • 0x1：时间戳达到目标值，此位读取后自动清 0
0	R/SS/RC	0x0	SEC_OF Timestamp Seconds Overflow，时钟戳秒数溢出 • 0x0：未溢出 • 0x1：溢出，时钟戳秒数溢出，此位读取后自动清 0

7.2.5.52. 0x0134 PPS_CTL

默认值：0x00000000			脉冲控制寄存器 (Pulse Control)
位域	类型	默认值	描述
31:7	–	–	–
6:5	RW	0x0	REACH_TRG_PPS_MODE Target Time Register Mode for PPS0 Output，达到目标时间 PPS 的输出模式 • 0x0：达到目标时间，只产生内部中断，不产生 PPS 信号 • 0x1： Reserved • 0x2：达到目标时间，产生内部中断以及触发 PPS 发出或者暂停 • 0x3：达到目标时间，只触发 PPS 发出或者暂停，不产生中断。
4	RW	0x0	PPS_MODE PPS Output Mode Select，PPS 输出模式选择 • 0x0：Bit[3:0]为 PPSCTRL0，控制 PPS 作为时钟输出 • 0x1：Bit[3:0]为 PPSEN0，控制可变 PPS 的命令操作
3:0	RW	0x0	PPSCTRL0 PPS0 Output Frequency Control，PPS0 作为时钟输出。 • 0x1: The binary rollover is 2 Hz, and the digital rolloveris 1 Hz. • 0x2: The binary rollover is 4 Hz, and the digital rolloveris 2 Hz. • 0x3: The binary rollover is 8 Hz, and the digital rolloveris 4 Hz. • 0xF: The binary rollover is 32.768 KHz, and the digitalrollover is 16.384 KHz. PPSCMD0，Flexible PPS0 Output Control，控制可变 PPS0 的操作命令 • 0x0：无操作 • 0x1：当系统时间达到目标时间，发送单个的 x2：当系统时间达到目标时间，发送连续的 Pulse • 0x3：系统时间还没达到目标时间，取消命令 0x1/0x2 • 0x4：当系统时间达到目标时间，暂停发送连续的 Pulse • 0x5：在命令 0x2 使能的连续 Pulse 情况下，立即停止发送 Pulse • 0x6：取消命令 0x4 的操作 • others：Reserved

7.2.5.53. 0x0138 PPS0_INTERVAL

默认值：0x00000000			脉冲间隔寄存器 (Pulse Interval)
位域	类型	默认值	描述
31:0	R/W	0x0	GAP_PPS The Gap of PPS0 Output Signal，PPS0 的脉冲间隔。

7.2.5.54. 0x013C PPS0_WIDTH

默认值：0x00000000			脉冲宽度寄存器 (Pulse Width)
位域	类型	默认值	描述
31:0	R/W	0x0	WID_PPS The Width of PPS0 Output Signal，PPS0 的脉宽。

7.2.5.55. 0x0FFC VERSION

默认值：0x00000101			EMAC 版本号 (VERSION)
位域	类型	默认值	描述
31:0	RO	0x00000101	Version（模块版本号） 采用 BCD 码显示，V1.1

7.3. Quad Serial Peripheral Interface (QSPI)

Serial Peripheral Interface (SPI)，即串行外围设备接口，是一种高速、全双工、同步的串行时钟接口。SPI 可允许微控制器 (MCU) 与各种外围设备之间以串行方式和特定的传输速率进行高效的信息交换。

Quad Serial Peripheral Interface (QSPI) 作为 SPI 的扩展版本，提供了更高速的数据传输能力。标准 SPI 最大支持两线数据传输，而 QSPI 最大支持四线数据传输。与此同时，QSPI 控制器配备了一个 64 字节的接收缓存 (RX FIFO) 和一个 64 字节的发送缓存 (TX FIFO)，并通过 AHB 总线进行高效的数据访问，适用于需要快速数据处理和传输的典型场景。通过中断或 DMA 方式，QSPI 控制器可以实现快速的数据传输，无需 CPU 过多干预，来增强系统的运行效率。

SoC 集成多套 QSPI 控制器，常用的使用场景如下：

- 快速读写访问串行 Flash 设备：

  QSPI0/ 1 为 QSPI 接口，用于访问 SPI NOR/ NAND Flash 等。

- LCD 显示控制等通用 SPI 接口设备的访问：

  QSPI2/ 3 为 QSPI 接口，用于 QSPI LCD 屏幕的显示控制。

- 兼容多种应用领域的外设

  QSPI4/ 5 为标准 SPI 接口，用于电机编码器的控制以及其它标准 SPI 接口设备的访问。

7.3.1. 功能特性

- 全双工同步串行接口
- 支持主机和从机模式（仅 QSPI2/ 3 支持），从机模式支持 1/ 2/ 4 线数据传输
- 具有 64 字节的 FIFO 缓存用于发送和接收数据
- 片选信号和时钟的相位和极性可配
- 支持中断模式或 DMA 模式
- 支持以下传输模式：
  - mode0
  - mode1
  - mode2
  - mode3
- 支持可编程串行数据流长度配置：从 1 bit 到 32 bits
- 支持非标准 SPI 接口：MOSI 和 MISO 引脚复用
- 支持标准 SPI 接口，双路输出/双路输入 SPI 接口，双路 I/O SPI 接口，四路输出/四路输入 SPI 接口，QIO 和 QPI 接口
- QSPI0/ 1支持 SPI NOR 的 Fast Read DTR，Dual IO DTR 和 Quad IO DTR 模式。
- QSPI0/ 1/ 4/ 5支持双片选 CS0 和 CS1
- QSPI2/ 3支持使用控制器内部 DMA。
- QSPI0 和 QSPI1 支持引脚重映射，支持 12 种线序调整方式。
- QSPI0 支持 XiP 模式，通过 AXI 直接读取 SPI NOR Flash
- QSPI2/ 3 支持 QSPI LCD 刷屏模式

7.3.2. 原理框图

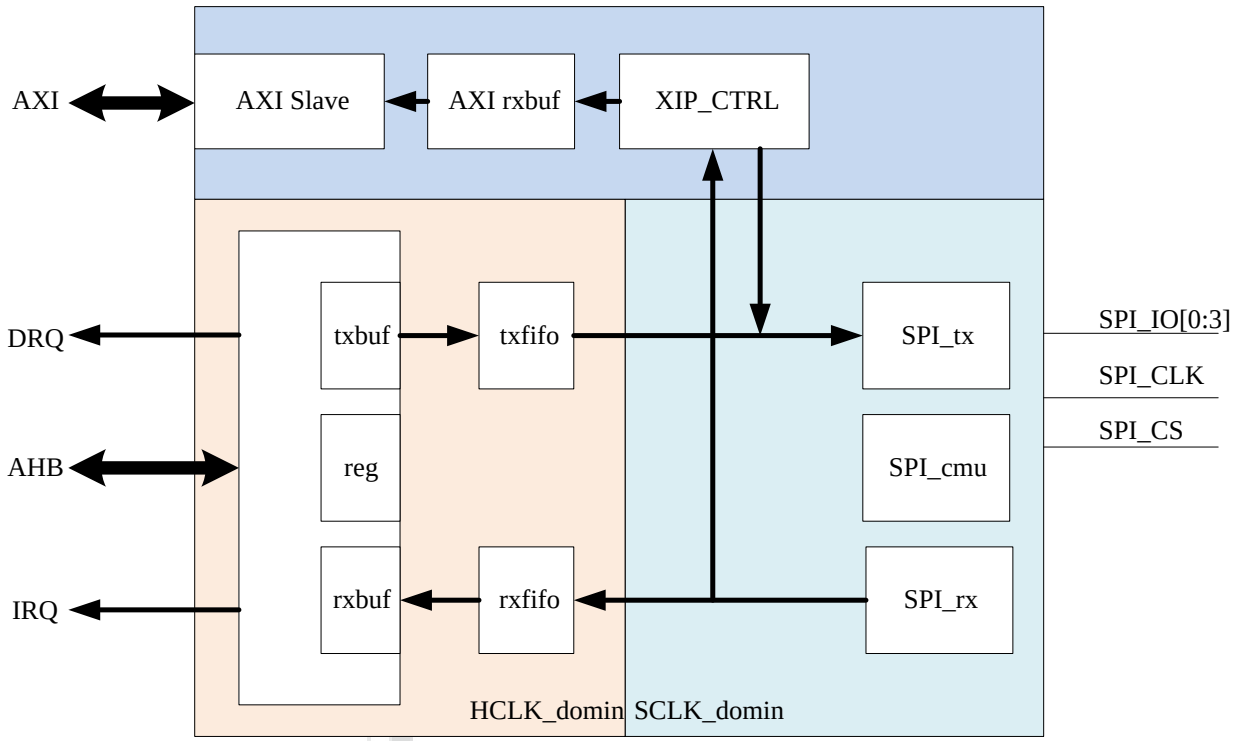


图 7–5 QSPI 原理框图

7.3.3. 工作模式

QSPI 控制器支持两种工作模式，用户根据实际工作需要选择主机模式或从机模式。

QSPI 控制器默认工作在从机模式下。配置 0x004 SPI_CFG 寄存器的 bit1 位为 1，选择主机模式。

- 主机模式下：
  1. 控制器产生 SPI_CLK 时钟发送给外部设备
  2. 数据从 TX FIFO 传送到 MOSI 引脚，MISO 引脚接收从机发来的数据传送给 RX FIFO。
  3. 在数据传输过程中，片选信号必须为低电平。

     片选信号可以通过寄存器配置为自动控制还是软件控制：

     - 当 SPI_TCFG.CS_CTL_SEL 字段为 0，控制器自动控制片选信号。
     - 当 SPI_TCFG.CS_CTL_SEL 字段为 1，则用软件手动控制片选信号。此时配置 SPI_TCFG.CS_LEVEL 决定片选信号的状态。
- 从机模式下：
  1. 接口工作时钟 SPI_CLK 由外部主机的时钟输入。
  2. 数据从 TX FIFO 传送到 MISO 引脚，从 MOSI 引脚接收来主机的数据传送给 RX FIFO。

3. 在数据传输过程中，片选信号由外部主机控制，不需要启动配置，只要有时钟输入，内部移位寄存器就会收发数据。
4. 从机模式默认工作在单线模式，可以配置为 Dual 或 Quad 模式，当 SPI_TCFG.SLV_OEN 字段为 1，配置 SPI 引脚输出使能。

7.3.4. 功能描述

主机模式

主机模式下，控制器支持的模式及其详细描述如下：

- 对于不同外设，可以选择相应的传输模式和总线模式：
  - 传输模式用于选择时钟信号的有效电平和采样相位。
  - 总线模式用于选择数据传输的有效位宽。
- 连续传输模式：通过寄存器配置突发传输的字节数，提高总线的传输效率。
- 多种不同的延时模式：满足不同传输路径引入的数据延时要求。
- 非字节对齐的位模式传输：可以 1~ 32 位任意配置，满足不同外设的传输协议要求。
- XIP 模式：对于 SPI NOR Flash，通过 AXI 总线直接访问 SPI NOR Flash 映射的物理地址，实现片上执行代码。
- 针对 SPI LCD 屏的图像刷屏模式：通过内部 DMA 模块，连续的发送图像数据，极大地提高 SPI LCD 屏的访问效率。

从机模式

从机模式下，也可以根据主机的要求选择相应的传输模式和总线模式。从机模式仅支持按字节对齐的传输，不支持按位对齐的传输。

如果发生了传输错误，控制器的状态寄存器相应的错误状态标志会置位， 并产生中断通知给 CPU 进行错误处理，保证数据传输的正确性。

7.3.4.1. 传输模式

QSPI 控制器支持多种不同的传输模式，外设使用的具体模式取决于其功能需求和性能特点。通过正确配置 SPI_CLK 信号的时钟极性 (SPI_TCFG.CPOL) 和时钟相位 (SPI_TCFG.CPHA)，可以实现不同的传输模式，从而确保数据传输的正确性。QSPI 控制器根据 SPI_CLK 信号从移位寄存器中读取或写入数据，在 SPI_CLK 时钟的上升沿或下降沿被采样。

CPOL 定义了 SPI_CLK 为空闲状态时的信号极性：

- 当 CPOL 为 1，SPI_CLK 为高电平。
- 当 CPOL 为 0，SPI_CLK 为低电平。

CPHA 的状态决定了 SPI_CLK 在时钟的奇数边沿是采样数据还是移位数据：

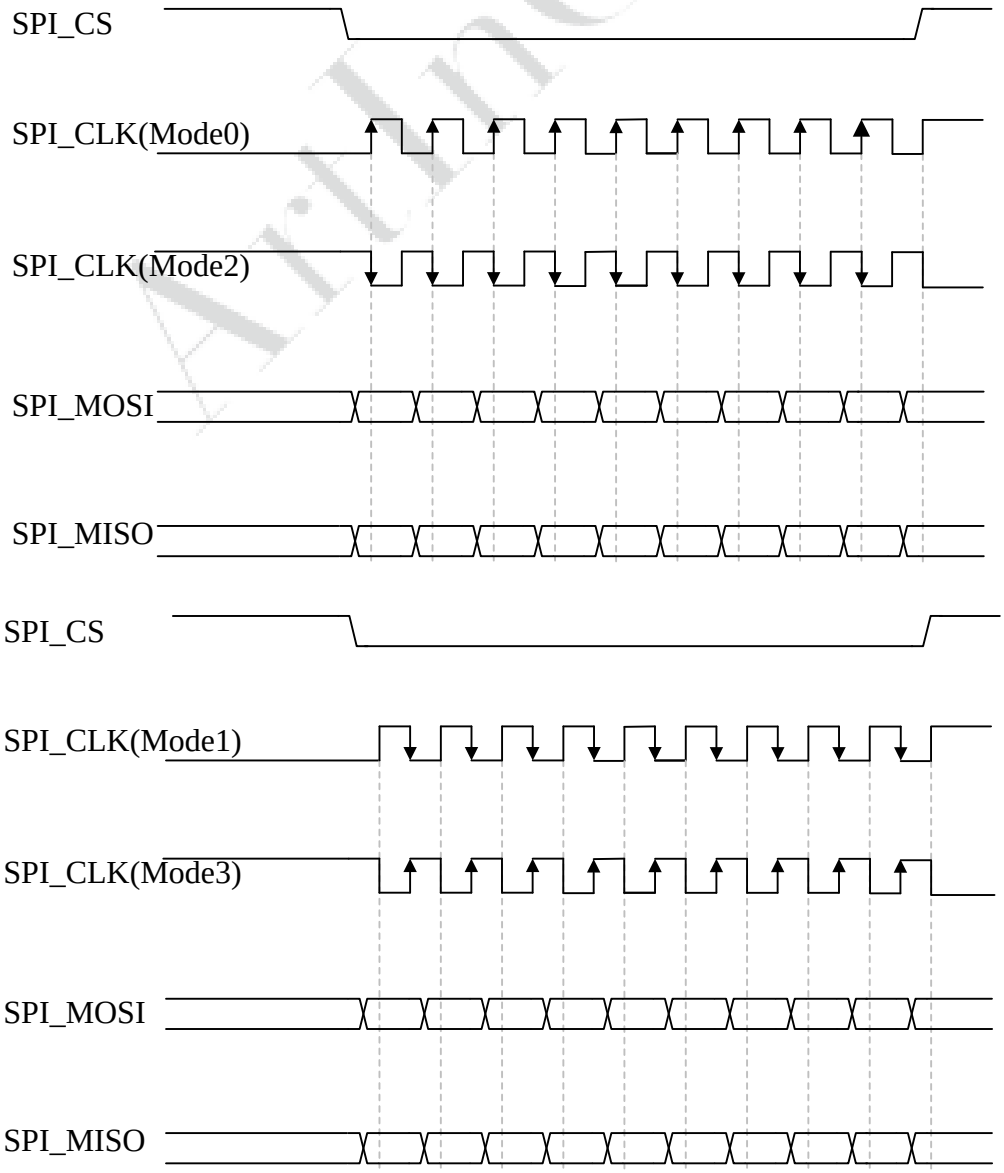
- 当 CPHA 为 1，在时钟的奇数边沿移位数据。
- 当 CPHA 为 0，在时钟的奇数边沿采样数据。

下表总结了几种不同的传输模式对应的极性、相位以及采样描述：

表 7–18 传输模式

SPI 传输模式	CPOL（极性）	CPHA（相位）	时钟奇数边沿	时钟偶数边沿
0	0	0	上升沿采样	下降沿移位
1	0	1	上升沿移位	下降沿采样
2	1	0	下降沿采样	上升沿移位
3	1	1	下降沿移位	上升沿采样

传输模式的流程图如下：



7.3.4.2. 总线模式

总线模式包含以下类型：

- SPI 标准模式
- SPI 三线模式
- SPI 双路输入/ 输出模式
- SPI 双路 I/O 模式
- SPI 四路输入 /输出模式
- SPI 四路 I/O 模式
- SPI QPI 模式
- SPI 突发传输配置

注：

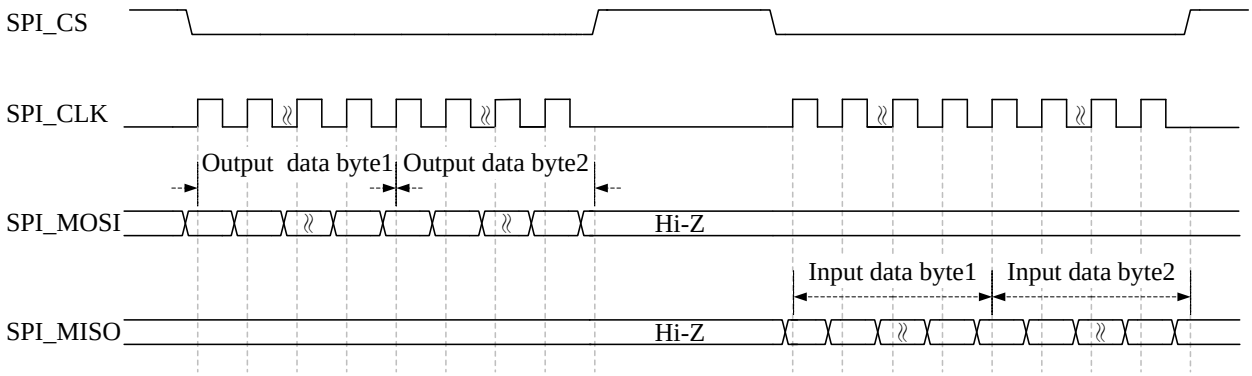
除了 SPI 标准模式和三线模式，在其它模式下，必须配置 SPI_TCFG.TXDLY_EN 位为 1，保证单线切换多线时命令或地址被正确采样。

7.3.4.2.1. SPI 标准模式

SPI 标准模式，是 SPI 默认的标准总线模式，使用以下信号线实现通信：

- SPI_CS：片选控制信号，在 SPI_CS 为低电平期间进行数据传输。
- SPI_CLK：时钟线。
- SPI_MOSI：作为主机时发送数据，作为从机时接收数据。
- SPI_MISO：作为主机时接收数据，作为从机时发送数据。

标准模式时序图如下：



注：

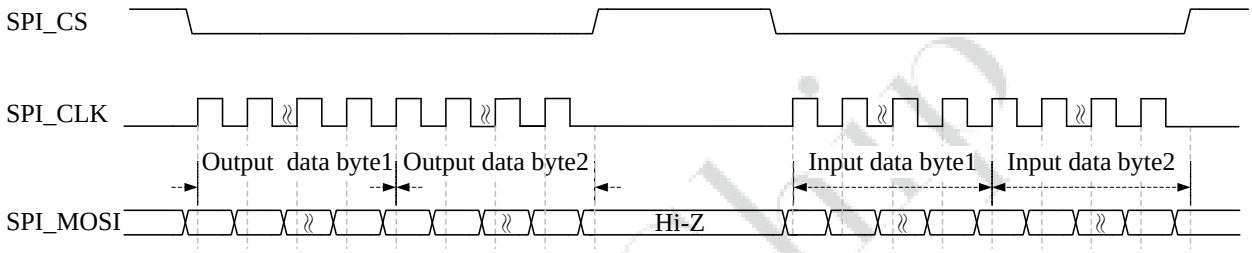
SPI 标准模式支持全双工，在发送数据的同时可以接收数据。

7.3.4.2.2. SPI 三线模式

SPI 三线模式是标准模式的变体，仅当 SPI 工作在主机模式时有效。SPI 三线模式使用以下信号线实现通信：

- SPI_CS：片选控制信号，在 SPI_CS 为低电平期间进行数据传输。
- SPI_CLK：时钟线。
- SPI_MOSI：数据的输入和输出的数据信号线 。

三线模式时序图如下：



通过配置寄存器 0x040 SPI_BMTC.BMOD_SEL 字段为 `0x2`，使用三线数据传输模式。

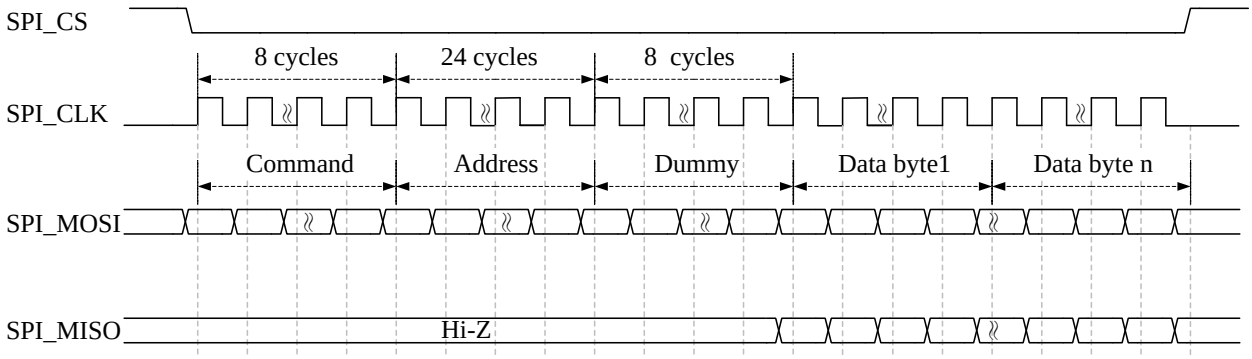
> **注：**
>
> SPI 三线模式默认是以位为单位的模式传输，通过配置 0x008 SPI_TCFG.3WIRE_EN，可以支持以字节为单位的模式传输。

7.3.4.2.3. SPI 双路输入/ 输出模式

SPI 双路输入/ 双路输出模式是指通过两根信号线 SPI_MOSI 和 SPI_MISO 进行发送或接收数据， 传输速率是标准单路 SPI 模式的两倍。

- SPI_CS：片选控制信号，在 SPI_CS 为低电平期间进行数据传输。
- SPI_CLK：时钟线。
- SPI_MOSI 和 SPI_MISO：命令/ 地址/ 填充数据通过 MOSI 线传输，有效数据通过 SPI_MOSI 和 SPI_MISO 两根信号线同时传输。
- 配置寄存器 0x038 SPI_TMC.DUAL_EN 位为 `1`，可选择 SPI 双路模式。

双路输入/ 输出模式时序图如下：

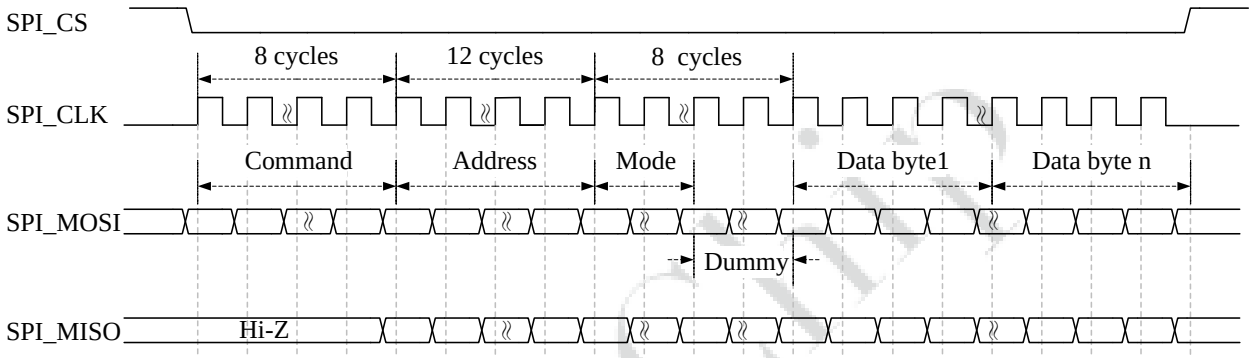


7.3.4.2.4. SPI 双路 I/O 模式

SPI 双路 I/O 模式的特点如下：

- 地址、填充数据和有效数据都通过两根信号线进行传输，命令通过 MOSI 进行单线传输。
- 配置寄存器 0x038 SPI_TMC.DUAL_EN 位为 1，可选择 SPI 双路 I/O 模式。

双路 IO 模式的时序图如下：

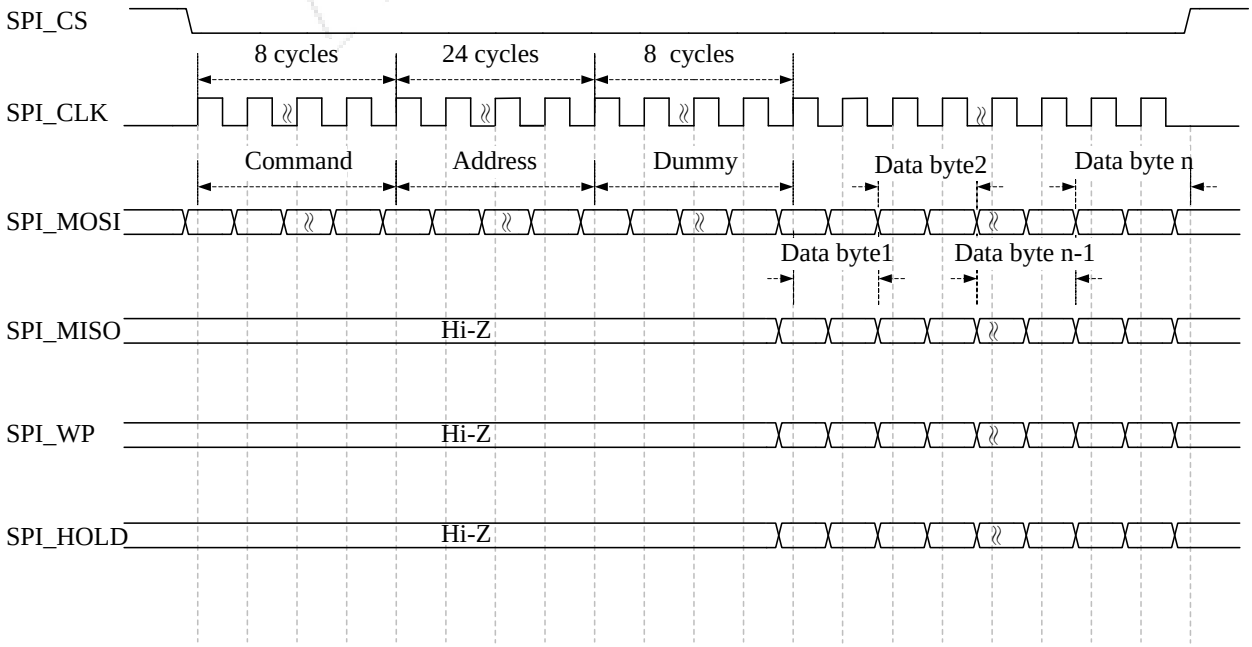


7.3.4.2.5. SPI 四路输入 /输出模式

SPI 四路输入输出模式是指通过四根信号线（MOSI/ MISO/ WP/ HOLD）进行发送或接收数据，此时传输速率是标准 SPI 模式的四倍。

- 命令/ 地址/ 填充数据通过 MOSI 单线传输，有效数据通过四根信号线同时传输。
- 通过配置寄存器 0x038 SPI_TMC.QUAD_EN 位为 1 进行选择，

四路输入/输出模式的时序图如下：

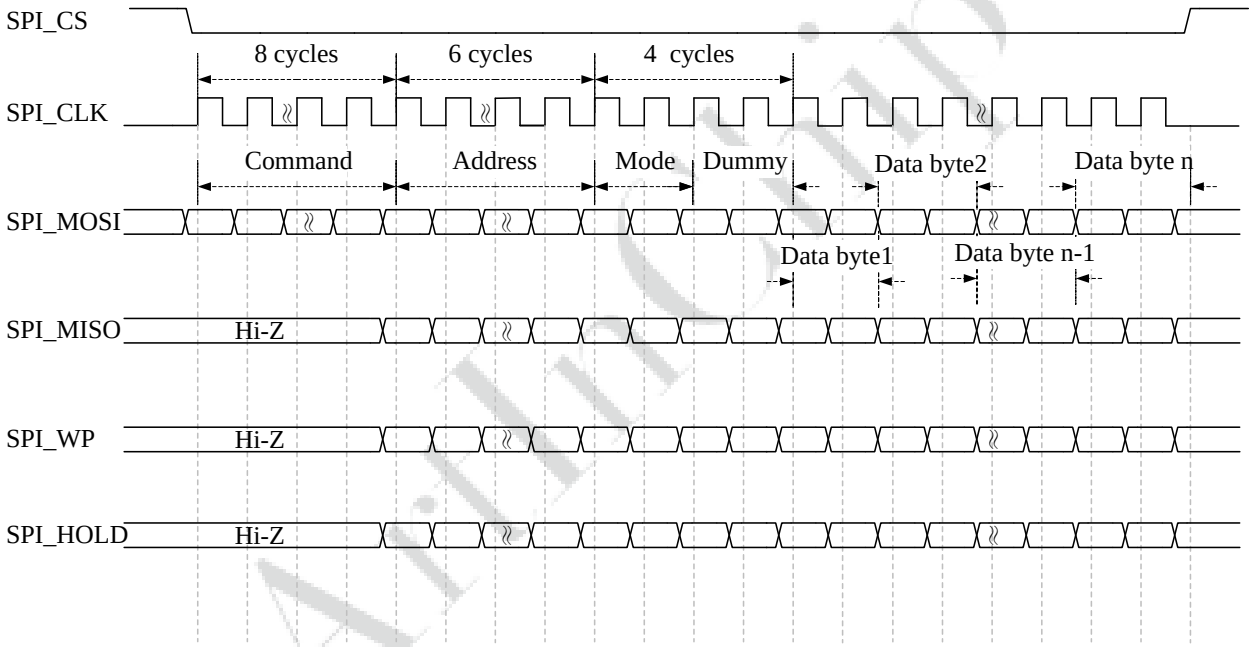


7.3.4.2.6. SPI 四路 I/O 模式

SPI 四路 I/O 模式的地址和数据信号全都使用 4 线传输。SPI 四路 I/O 模式下：

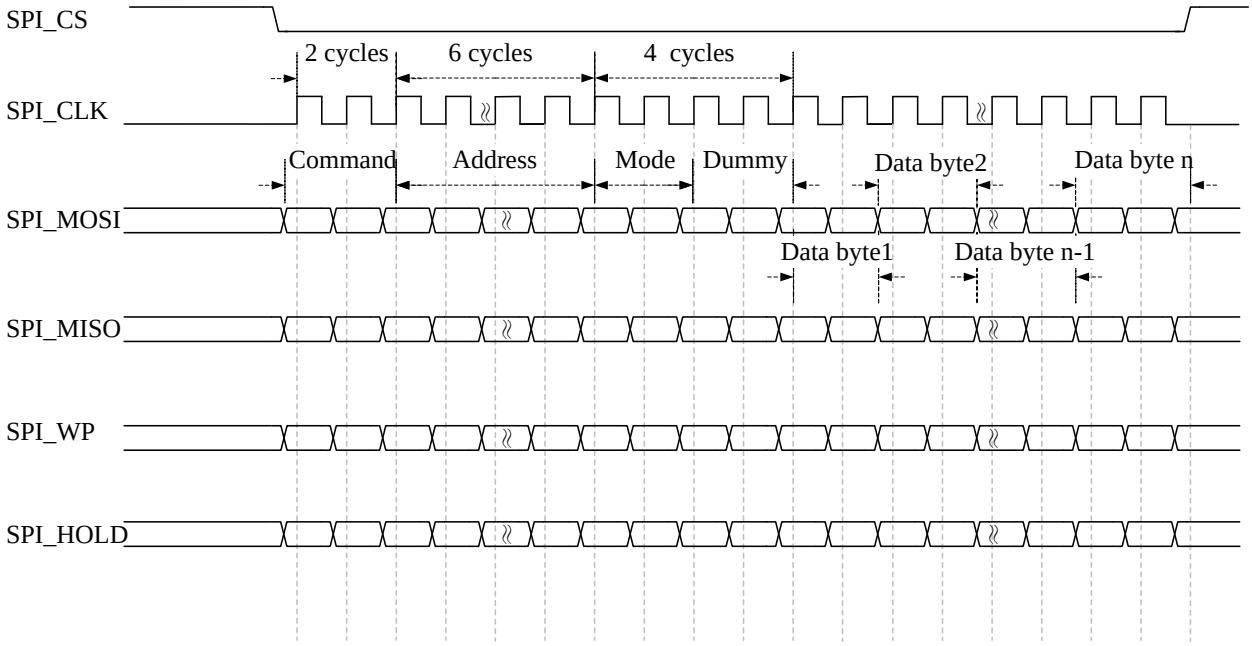
- 串行 Flash 设备支持快速读模式，即只需要发送一次读命令，后续操作可以跳过读命令，在 CS 信号拉低期间，仅发送地址即可读取数据。
- 支持带回环功能的突发读配置，支持的突发长度为 8/ 16/ 32/ 64 字节， 常用于 XIP 模式。
- 当 Mode 字节的 bit[5:4] = 2b'10 时，工作在读命令 bypass 模式， 即下一笔读操作不需要发送读命令。

四路 I/O 模式时序图如下：



7.3.4.2.7. SPI QPI 模式

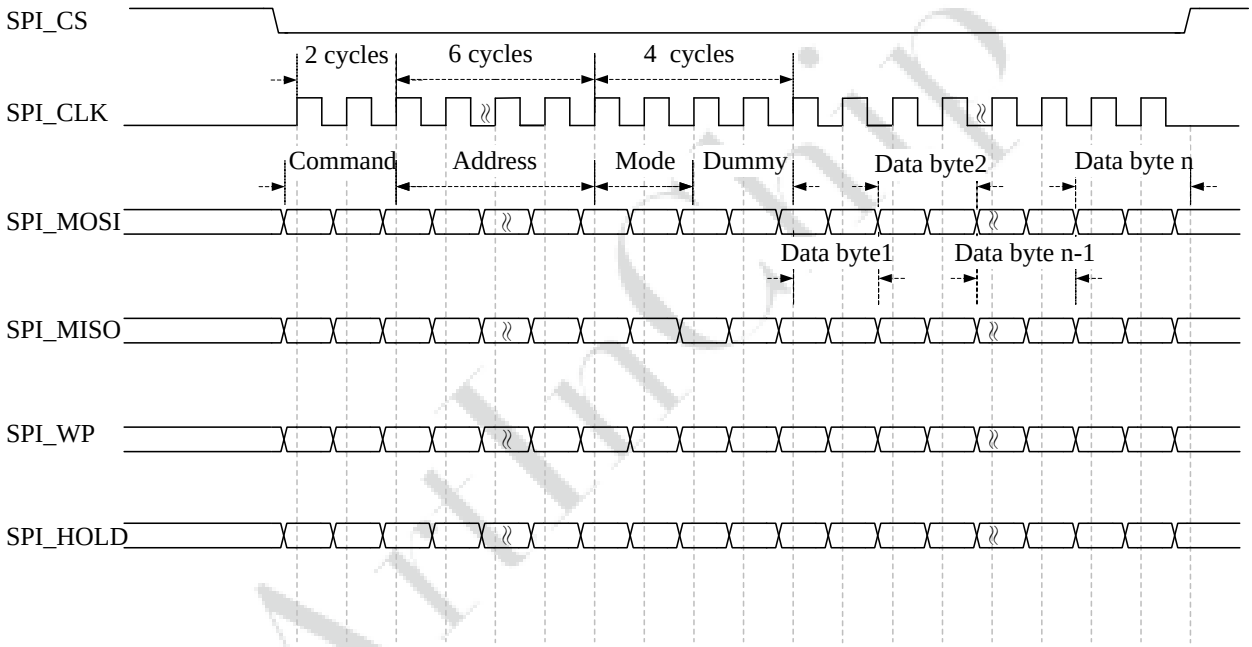
QPI 模式命令、地址和数据全部使用 4 线传输。QPI 模式不是标准的 SPI 模式，与标准 SPI 模式之间需要使用命令进行切换。QPI 模式也支持突发传输配置。时序图如下：



7.3.4.2.8. SPI DTR 模式

控制器支持 DTR 模式，即地址和数据按双边沿传输，命令按单边沿传输。SPI NOR 支持 Fast Read DTR，Dual I/ O DTR 和 Quad I/ O DTR 三种总线模式。配置 0x028 SPI_MISC.DTR_EN 位置 1，使能 DTR 模式。由于 DTR 模式 Dummy 是按照时钟周期数配置的，配置 0x028 SPI_MISC.DTR_DUMMY 位域选择对应频率的 Dummy 值。

SPI DTR 模式的时序图如下：



7.3.4.2.9. SPI 突发传输配置

当 SPI 工作在 Quad I/O 模式下，串行 Flash 设备支持带回环的突发传输 (Burst With Wrap)：

- 发送突发传输配置命令 0x77 或 0xC0，配置三个 dummy 字节。
- 配置 Mode 字节的 bit[6:4] 为相应的突发类型和突发长度。

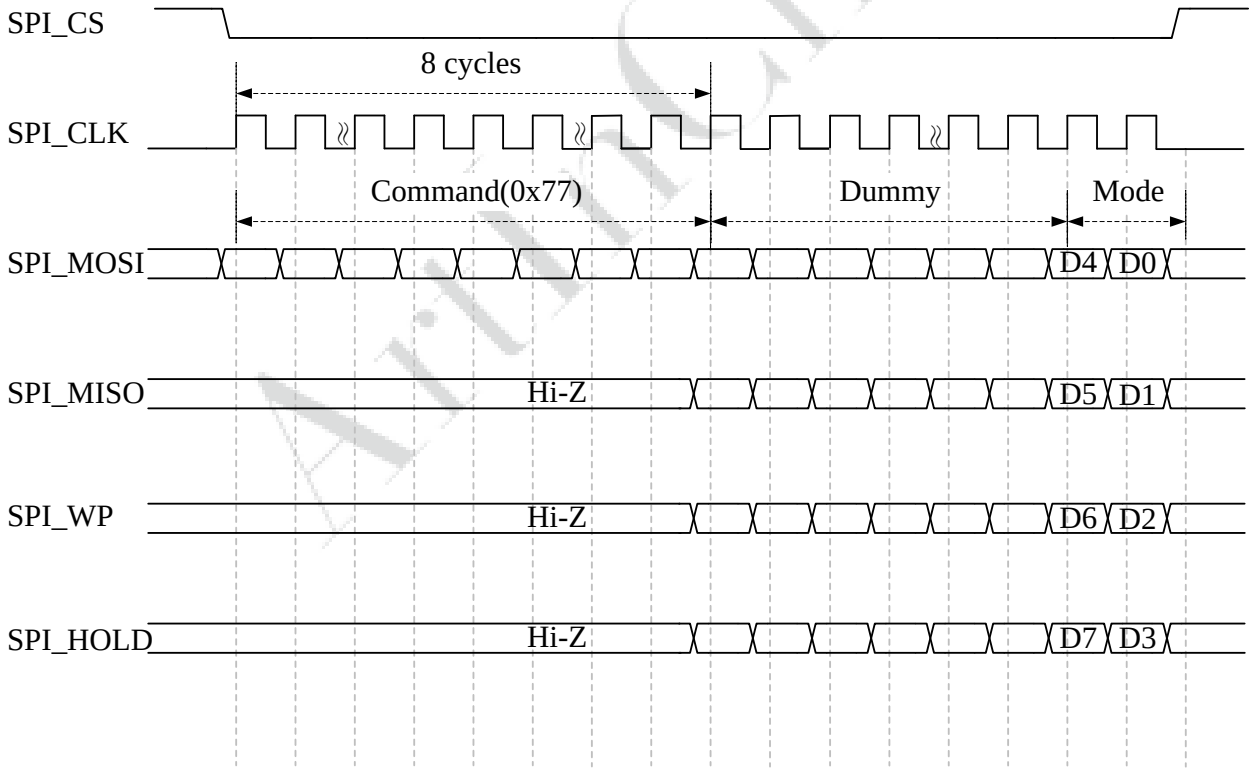
常用于 XIP 访问模式下，CPU 进行 Cache 操作，可以提高总线的传输效率，提高代码的执行性能。当 CPU 使用 Cache，并开启预取功能，突发传输类型配置为 WRAP，突发传输长度配置为 32 个字节。

在 XIP 模式下，配置 SPI_BTR.WRAP_EN 位为 1，使能回环模式，控制器在发送第一笔接收命令之前，会自动根据 SPI_BTR 寄存器的配置，自动发送突发传输配置命令。

Mode Bit[6:4]	突发类型	突发长度
3b’001	Linear	取决与 Flash 容量
3b’000	WRAP	8 个字节
3b’010	WRAP	16 个字节
3b’100	WRAP	32 个字节
3b’110	WRAP	64 个字节

当 SPI 工作在 QPI 模式，突发传输配置命令与标准模式下的命令不同，增加了频率相关的 dummy 字节数量的配置。对于 QPI 模式，不同厂家的突发传输配置方式或许不同，需要参考具体厂家的芯片规格书。举例 Winbond 的 W25Q128JW 芯片定义如下：

Mode Bit[5:4]	Dummy 时钟周期数	Mode Bit[1:0]	WRAP 长度
2b’01	4（80MHz）	2b’01	16 个字节
2b’10	6（104MHz）	2b’10	32 个字节
2b’11	8（133MHz）	2b’11	64 个字节



7.3.4.3. XIP 模式

在主机模式下，配置 QSPI0 控制器工作于 XIP 模式，可以通过 AXI 直接读取 SPI 的物理地址空间，就像访问内存空间一样，常用于 XIP 启动。在 XIP 模式下，SPI 的物理地址需要映射为 AXI 可访问的逻辑地址。

XIP 控制逻辑主要包括如下模块：

- AXI slave 模块：用于和 AXI master 进行通讯；
- XIP 控制器：将 AXI 的地址和数据信息转换为 SPI 协议规定的传输格式；
- AXI 数据缓存：TX buffer 用于缓存 AXI 发送到 SPI 的命令和地址信息，而 RX buffer 用于缓存从 SPI 接口接收到的数据。

XIP 模式传输：允许 CPU 或 其它 AXI Master 直接从 SPI NOR 获取数据，无需先将数据读取到内存中再获取。这是一种高效的数据访问方式，可以减小内存的使用空间。常用于获取较大的图像数据。

XIP 模式启动：CPU 通过 AXI 直接访问 QSPI 的映射地址读取指令。在系统引导启动阶段，通过 AHB 接口初始化 QSPI 控制器。完成相关配置后，CPU 跳转到 QSPI 映射的起始地址进行取指令，完成系统启动。

XIP 的详细传输流程如下：

1. 系统上电后，在引导程序中通过 AHB 访问的方式对 QSPI 设备进行初始化。

   在系统启动阶段，需要配置基本的 SPI NOR 初始化信息，配置 SPI NOR 配置寄存器的 QE 位为 1 ，确保其工作在 Quad 模式。

2. 通过 XIP 模式相关的寄存器，并使能 XIP 模式传输。配置流程如下：

   - 配置 SPI_RCM 寄存器，包括读命令指令值，dummy 数量，读命令 bypass 模式使能，Mode 字节等信息。
   - 配置 SPI_BTR 寄存器，包括突发传输配置命令指令值，传输位宽，dummy 数量，回环和线性模式指令等信息。
   - 配置 SPI_TMC 寄存器，配置 QADDR_EN 位置 1，选择 QIO 模式；配置 QPI_EN 位置 1，选择 QPI 模式。
   - 配置 SPI_CFG.AMOD 位为 1，使能 XIP 模式。

   注：

   如果 SPI NOR 使能读命令 bypass 模式，则除第一笔读操作外，后续的读操作将无需重复发送读命令。此配置可以简化读取流程，提升数据访问效率。

3. CPU 通过 AXI master 地址通道发起读请求，AXI slave 接收到读请求，获取地址信息、突发传输类型和突发长度。
4. XIP 控制器从寄存器 SPI_BTR 和 SPI_RCM 获取固定的数据传输格式，该方式仅支持 Quad I/O 和 QPI 模式。
5. 按照解析得到的传输格式，QSPI 控制器自动发送读命令，读取指令或数据信息并将其缓存到内部 RX buffer。
6. CPU 通过 AXI master 数据通道从 RX Buffer 中获取指令或数据。

   完成以上操作后，完整的读操作即告完成。

注：

如果开启 Cache 功能，CPU 每次读取 32 个字节数据，相当于一条 cache line 的大小，此时只需要配置一次突发长度即可。后续的 cache 读取都会使用这一固定的突发长度，且突发类型为 WRAP 模式，以此提高数据处理效率。

7.3.4.4. 连续传输

在主机模式下，SPI 可以进行连续数据传输，即发送或接收数据以字节为单位进行多个字节连续传输。详细功能配置如下：

- 发送数据的连续传输字节长度：通过寄存器 0x038 SPI_TMC.TXD_CNT 字段配置。
- 单路模式下，连续传输的长度：通过寄存器 0x038 SPI_TMC.STXD_CNT 字段配置。
- 控制器会自动发送填充数据，填充数据连续传输长度：通过寄存器 0x038 SPI_TMC.DMY_CNT 字段配置。
  - 如果不希望控制器自动传输填充数据，可以将填充数据作为发送数据一起写入寄存器0x038 SPI_TMC.TXD_CNT 字段。

在主机模式下，所有数据的突发传输长度配置在寄存器 0x030 SPI_TBC.TB_CNT 字段，当所有的发送数据和接收数据连续传输完成后，上述的配置字段都会被清零。

7.3.4.5. 延时模式

QSPI 控制器支持两种延时模式：

- 粗调模式：最小调节范围为 0.5 个时钟周期，可调范围为 0~2 个时钟周期。
- 细调模式：最小调节范围为 90 度相位，源时钟采用 2 倍时钟频率，可调范围为 0~3 个时钟周期。

QSPI 控制器默认工作在细调模式，对于 RX 数据延时半周期采样。配置以下位置 1，可以使控制器工作在粗调模式：

- BMRX_PHASE_MASK
- TX_PHASE_MASK
- RX_PHASE_MASK

粗调模式

QSPI 控制器的接口最大时钟频率为 100 MHz。在主机模式下，内部的 SPI 时钟频率与外部接口的时钟频率一致。SPI 时钟根据不同的时钟源工作在不同的模式下，有以下几种采样模式：

- 正常采样模式，不延时
- 延时半个时钟周期采样，QSPI 控制器默认工作在该采样模式
- 延时一个时钟周期采样
- 延时 1.5 个时钟周期采样
- 延时两个时钟周期采样

不同的采样模式配置见下表：

表 7–19 采样模式配置

采样模式	RXDLY_DIS 位	RXINDLY_EN 位	RXDLY_EXTD 位	时钟频率范围
正常模式	1	0	0	<= 24 MHz
延时半个时钟周期	0	0	0	24 ~ 60 MHz
延时一个时钟周期	0	1	0	60 ~ 100 MHz
延时 1.5 个时钟周期	0	0	1	大于 100 MHz
延时两个时钟周期	0	0	2	大于 100 MHz

细调模式

为了满足长路径传输的使用场景，且延时精度更高，控制器增加了细调模式，源时钟频率采用 2 倍频的输入模块时钟频率，控制器最大工作时钟频率为 200 MHz，QSPI 接口最大工作时钟频率为 100 MHz。

- 对于输出端口，时钟延时支持 0~360 度共 5 个相位调节，每个相位为 90 度。TX 数据支持 0 度和 180 度延时。
- 对于内部 RX 采样模式：

- 在 bit 传输模式下，时钟延时支持 0~720 度共 9 个相位调节，每个相位为 90 度。
- 在 byte 传输模式下，时钟延时支持 0~1080 度共 13 个相位调节，每个相位为 90 度。

详情参考 SPI_PHACTL 和 SPI_BMCLK 寄存器描述。

注：

采样模式与时钟频率的关系不是唯一确定的，取决于传输路径的延时时间，用户可根据实际情况选择合适的配置。

7.3.4.6. 采样模式

QSPI 控制器的接口最大时钟频率为 100 MHz。在主机模式下，内部的 SPI 时钟频率与外部接口的时钟频率一致。对于不同的时钟源，SPI 时钟可以工作在以下三种时钟模式下：

- 正常采样模式：采样时钟与输出时钟同相位，适用于标准的数据传输，没有额外的延时。
- 延时半周期采样模式：采样时钟相对于输出时钟延迟半个周期后进行数据采样，适用于当数据在半个时钟周期后达到稳定状态的场景，可以提供更准确的数据传输。

  QSPI 控制器默认工作在该采样模式。
- 延时一个周期采样模式：采样时钟相对于输出时钟延迟一个周期后进行数据采样，适用于数据线延时较大的场景，确保数据传输的完整性和准确性。

三种时钟的采样模式配置见下表：

表 7–20 采样模式配置

采样模式	RXDLY_DIS 位	RXINDLY_EN 位	时钟频率范围
正常模式	1	0	≤ 24 MHz
延时半个周期采样	0	0	24 ~ 60 MHz
延时一个周期采样	0	1	60 ~ 100 MHz

注：

采样模式与时钟频率的关系不是唯一确定的，取决于传输路径的延时时间，用户可根据实际情况选择合适的配置。

7.3.4.7. 位模式传输

位模式传输是以位为单位进行数据传输，用于对 SPI 接口的非 Flash 类型的外设进行访问，兼容更多的 SPI 设备。

位模式分为标准模式和三线模式。可通过下列步骤完成位模式传输配置：

1. 通过0x040 SPI_BMTC 寄存器的 BMOD_SEL 位域，选择两种不同的工作模式中的一种。

   注：

   关于标准模式和三线模式的详细说明，可查看总线模式 。

2. 通过 0x040 SPI_BMTC 寄存器选择位模式下的工作时钟。
3. 确定位模式的传输位数。位模式一次最大传输的位数为 32 bits：
   - 对于写操作，将待发送数据写入到 0x048 SPI_BMTXD 寄存器，并通过 0x040 SPI_BMTC.BM_TXCNT 配置写有效数据的长度。
   - 对于读操作，通过 0x040 SPI_BMTC.BM_RXCNT 配置读有效数据的长度， 读到的数据会自动存入 0x048 SPI_BMTXD 寄存器。
4. 使能 BM_START 寄存器启动数据传输，传输完成后 BM_TD 寄存器会置 1，也可以使能 BM_TD_INTE 完成中断，在中断服务程序中检查数据传输完成标志。

   注：

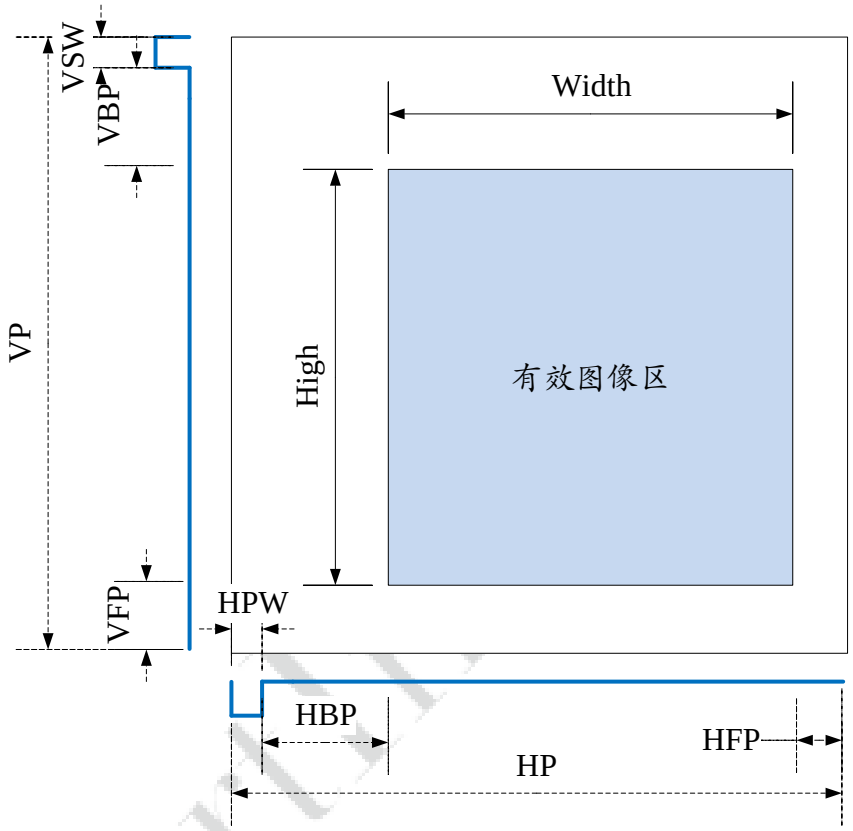
   位模式连续传输时，每笔传输结束后需要硬件复位一次，或者至少等待一个时钟周期，再传输下一笔数据。

7.3.4.8. 图像刷屏模式

QSPI 控制器支持 SPI LCD 屏的刷屏模式，允许在 SPI 屏初始化完成后，通过使能 SPI 刷屏模式，由硬件根据屏幕的配置信息和命令参数，使用内部 DMA 自动从指定的内存区域发送一帧 RGB 图像数据。这一过程利用内部 Direct Memory Access (DMA) 的功能，有效地从指定的内存区域传输数据，而无需增加 CPU 的额外负担。

LCD 屏幕包含以下关键区域，如图所示：

- VSW 区域：列同步宽度
- VBP 区域：列扫描后肩
- 有效区域：图像实际显示区域
- VFP 区域：列扫描前肩
- HSW 区域：行同步宽度
- HBP 区域：行扫描后肩
- HFP 区域：行扫描前肩



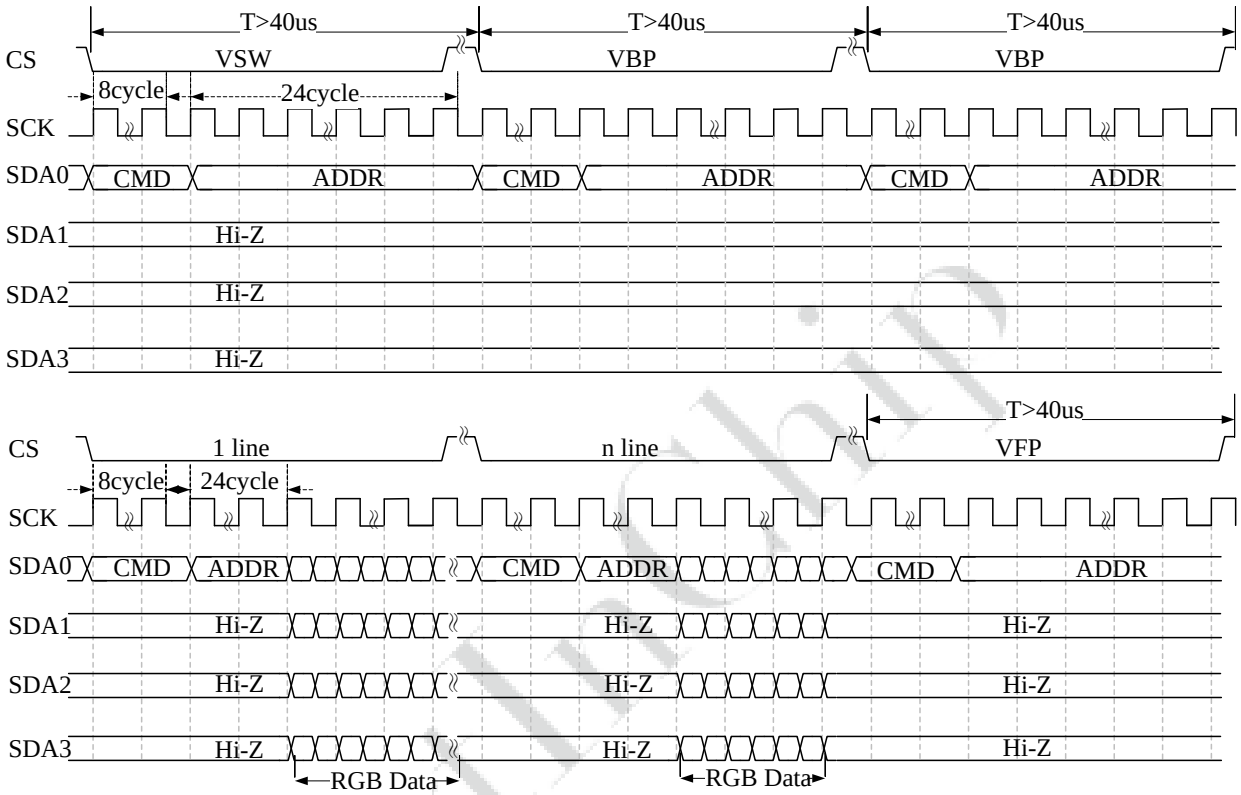
图像关键区域**时序配置如下：**

VSW、VBP、VFP 时序：按照屏的使用手册配置相关寄存器的行数量。

HSW、HBP、HFP 时序：由屏的驱动芯片自动控制。

图像传输顺序

QSPI 控制器按照 VSW 行 → VBP 行 → 有效图像行 → VFP 行的特定顺序传输每一帧图像，确保 LCD 屏幕正确显示图像。图像传输时序如下：



通过相应的命令寄存器配置行命令，行命令由 32 bits 组成，包含命令和地址。硬件自动将行命令填充到 FIFO：

- 对于 VSW、VBP、VFP 行，只需要传输行命令。
- 对于有效区域的行，需要传输行命令以及有效的图像数据：
  - 通过内部 DMA 自动将有效图像数据发送给 SPI 屏。
  - 通过 DUAL_EN 和 QUAD_EN，可 配置有效数据传输的线宽，支持 1/ 2/ 4 线传输。

7.3.4.8.1. 刷屏模式配置

SPI 刷屏模式配置流程如下：

1. 配置 SPI_CFG.DISP_EN 位置 1，使能刷屏模式。
2. 通过 SPI_DISP_HIGH 寄存器，配置 VBP、VFP 和有效图像区的行高，VSW 只有一行，不用配置高度。
3. 通过 SPI_DISP_WIDTH 寄存器，配置有效图像区的宽度。
4. 通过命令寄存器配置每个区域的行命令。

   行命令可以指导硬件如何正确地驱动显示屏。
5. 配置 SPI_DISP_RTIME 寄存器，设定 VSW、VBP、VFP 区行命令的附加时间值，该值应该大于每行的刷新时间。

   设定合理的附加时间值，可确保 VSW、VBP、VFP 区行命令的执行时间充足，避免因时序问题导致的画面显示异常。
6. 配置 IDMA 发送地址和长度寄存器，使能 IDMA 发送通道，详情可查看编程指南。

7. 设置 STRAT 位为 1，硬件自动开始传输一帧数据。每个区域的行传输完成后，DISP_DTO 中断状态位置 1。每一帧传输完成，TC 中断状态位置 1。
8. 传输完一帧数据后，重复配置步骤 2、3、6、7，软件控制帧连续传输。

7.3.4.8.2. 硬件连续帧传输模式

SPI 刷屏模式配置流程如下：

1. 配置 SPI_CFG.DISP_EN 位置 1，使能刷屏模式。
2. 通过 SPI_DISP_HIGH 寄存器，配置 VBP、VFP 和有效图像区的行高，VSW 只有一行，不用配置高度。
3. 通过 SPI_DISP_WIDTH 寄存器，配置有效图像区的宽度。
4. 通过命令寄存器配置每个区域的行命令。
5. 配置 SPI_DISP_RTIME 寄存器，设定 VSW、VBP、VFP 区行命令的附加时间值，该值应该大于每行的刷新时间。
6. 配置 0x0C0 SPI_DISP_FTIME 寄存器，设定每帧数据之间的等待时间，该值应参考屏规格书。
7. 配置 DISP_DMC 位域，对于 RGB565 格式的图像数据，Quad 模式下需要在 CMD 和 DATA 之间插入 dummy 时钟数。
8. 配置 IDMA 发送地址和长度寄存器，使能 IDMA 发送通道，详情可查看编程指南。
9. 设置 DISP_RESTART 位为 1，硬件执行连续帧传输。每个区域的行传输完成后，DISP_DTO 中断状态位置 1。每一帧传输完成，TC 中断状态位置 1。传输过程中可以随时修改 IDMA_TX_ADDR 的值，当前帧传输完成后，硬件会自动切换到目标图像缓存的地址。

在系统带宽紧张的场景下，可以选配控制器外部 2KB SRAM 作为发送缓存使用，在 DISP_RESTART 使能之前，按如下方式配置：

1. 配置 EXT_TXF_EN 位为 1，使能使用外部 SRAM；配置 DISP_URGENT_EN 位为 1，使能 urgent 信号。
2. 配置 DISP_VLDL_DRQ_SEL 位域为 1 ，推荐使用模式 1 ，在行传输间隔时间内控制器发送 IDMA 请求，当发送缓存数据量大于等于配置的水位值时，发起下一笔行传输。
3. 配置 DISP_TRANS_TRIG_WL 位域，选择合适的触发水位，当一笔行传输结束后，发送缓存的水位大于等于该值，发起新的行传输。
4. 配置 DISP_URGENT_THRL 和 DISP_URGENT_THRH 位域，选择 urgent 信号有效和无效的触发阈值。当发送缓存的数据量小于等于 DISP_URGENT_THRL 值，urgent 信号有效；当发送缓存的数据量大于等于 DISP_URGENT_THRL 值，urgent 信号无效。

7.3.4.9. 错误状态

QSPI 控制器在操作过程中可能会遇到多种错误状态，从而影响系统的正常运行和性能。对于 QSPI 控制器，如发生任何一种错误，硬件都会将 SPI 中断状态寄存器中对应的位置 1 并停止传输。以下是主要的错误状态描述及其对应的处理方法：

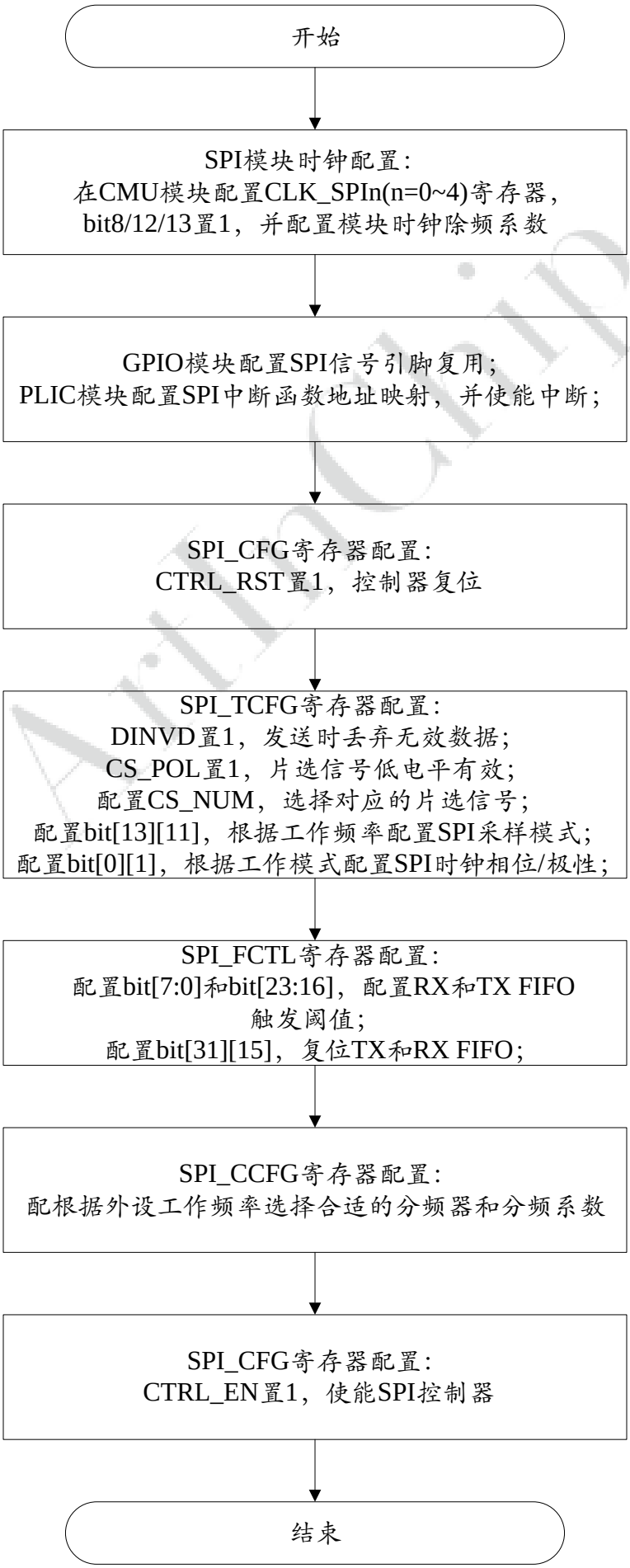
表 7–21 QSPI 控制器错误状态及处理方法

错误类型	原因	状态	处理方法及结果
TX_FIFO 下溢出	从空的 FIFO 中读取数据	控制器停止传输并将对应的中断状态位 TF_UDF 置 1。 此时控制器会产生一个硬件中断信号。	如果开始新的数据传输，需要在 0x004 SPI_CFG 寄存器中将 CTRL_RST 位置 1，将 FIFO 进行复位。处理完该异常后需要软件将该状态位清零。
TX_FIFO 上溢出	向满的 FIFO 中发送数据	控制器停止传输并将对应的中断状态位 TF_OVF 置 1。 此时控制器会产生一个硬件中断信号。	如果开始新的数据传输，需要在 0x004 SPI_CFG 寄存器中将 CTRL_RST 位置 1，将 FIFO 进行复位。处理完该异常后需要软件将该状态位清零。
RX_FIFO 下溢出	从空的 FIFO 中读取数据。	控制器停止传输并将对应的中断状态位 RF_UDF 置 1。 此时控制器会产生一个硬件中断信号。	如果开始新的数据传输，需要在 0x004 SPI_CFG 寄存器中将 CTRL_RST 位置 1，将 FIFO 进行复位。处理完该异常后需要软件将该状态位清零。
RX_FIFO 上溢出	向满的 FIFO 中发送数据	控制器停止传输并将对应的中断状态位 RF_OVF 置 1。 此时控制器会产生一个硬件中断信号	如果开始新的数据传输，需要在 0x004 SPI_CFG 寄存器中将 CTRL_RST 位置 1，将 FIFO 进行复位。处理完该异常后需要软件将该状态位清零。

7.3.5. 编程指南

7.3.5.1. 初始化流程

在进行数据传输之前，需要对 SPI 接口和控制器进行初始化。QSPI 控制器的初始化主要包括时钟配置，引脚配置，以及工作模式配置等，详情如下：



7.3.5.2. 字节模式传输流程

QSPI 在主机和从机之间传输串行数据，主要采用两种传输方式：

- CPU 模式
- DMA 模式

在数据传输过程中，QSPI 有两个数据通道：

- 发送通道：数据传输路径从 TX FIFO 到外部设备。

  发送数据时，CPU 或 DMA 将数据写入 TXD 寄存器，控制器自动将寄存器的数据搬到 TX FIFO 中，再由移位寄存器将 FIFO 中的数据发出。

- 接收通道：数据传输路径从外部设备到 RX FIFO。

  接收数据时，CPU 或 DMA 可以访问 SPI_RXD 寄存器获取接收数据，控制器自动将接收到的数据从 RX FIFO 搬到寄存器中。

每一笔数据传输完成后，QSPI 控制器都会发送一个传输完成的中断信号给处理器，此时 SPI 中断状态寄存器的 TD 位被置 1。

以下为不同模式下，QSPI 的数据传输流程：

• 通过 CPU 在中断服务程序中读写 FIFO，完成数据传输，流程图如下：

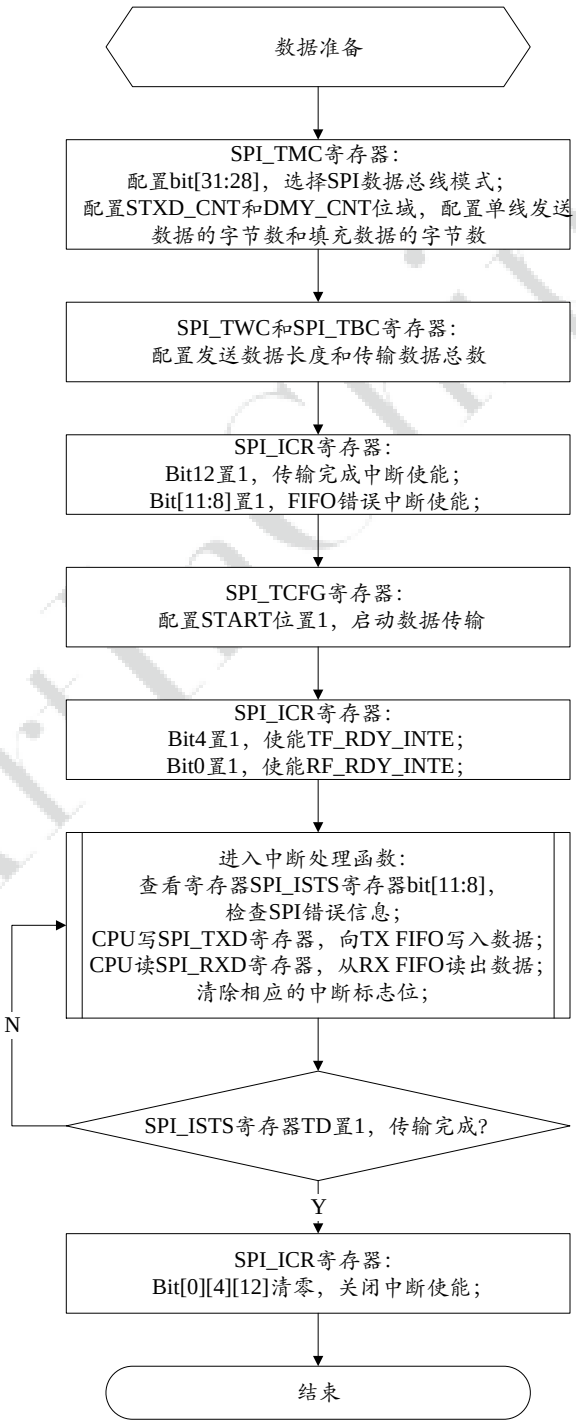


图 7–6 CPU 模式

• 通过 GPDMA 在内存和 FIFO 之间搬运数据，完成数据传输，流程图如下：

图 7–7 DMA 模式

QSPI 控制器同时支持使用内部 DMA 通过 AHB 主机接口搬运数据，只需要配置源地址和目标地址，以及数据长度即可。 相关的寄存器配置包括：

• SPI_IDMA_BTCFG

• SPI_IDMA_RXADDR

• SPI_IDMA_RXLEN

- SPI_IDMA_TXADDR
- SPI_IDMA_TXLEN

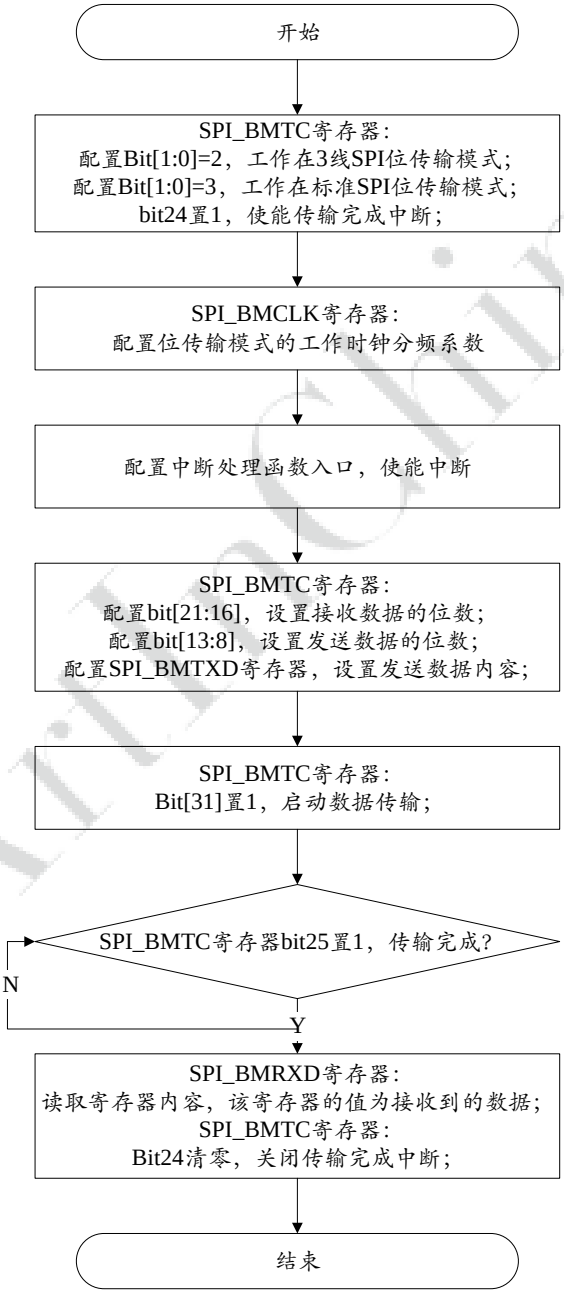
图 7–8 IDMA 模式的配置流程

7.3.5.3. 位模式传输流程

SPI 在主机和从机之间传输串行数据，支持按位模式访问，相关的寄存器配置包括：

- SPI_BMTC
- SPI_BMCLK
- SPI_BMTXD
- SPI_BMRXD

流程图如下：



7.3.6. 寄存器列表

表 7–22 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	SPI_VER	SPI 控制器版本	0x000 SPI_VER
0x004	SPI_CFG	SPI 配置	0x004 SPI_CFG
0x008	SPI_TCFG	SPI 传输配置	0x008 SPI_TCFG

表 7–22 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x00C	SPI_PHACTRL	SPI 相位控制	0x00C SPI_PHACTRL
0x010	SPI_ICR	SPI 中断控制	0x010 SPI_ICR
0x014	SPI_ISTS	SPI 中断状态	0x014 SPI_ISTS
0x018	SPI_FCTL	SPI FIFO 控制	0x018 SPI_FCTL
0x01C	SPI_FSTS	SPI FIFO 状态	0x01C SPI_FSTS
0x024	SPI_CCFG	SPI 时钟配置	0x024 SPI_CCFG
0x028	SPI_MISC	SPI 其它功能配置	0x028 SPI_MISC
0x02C	SPI_DDC	SPI 双边沿模式 Delay Chain 配置	0x02C SPI_DDC
0x030	SPI_TBC	SPI 总传输字节计数器	0x030 SPI_TBC
0x034	SPI_TWC	SPI 写发送计数器	0x034 SPI_TWC
0x038	SPI_TMC	SPI 传输控制	0x038 SPI_TMC
0x040	SPI_BMTC	SPI 位模式传输配置	0x040 SPI_BMTC
0x044	SPI_BMCLK	SPI 位模式时钟配置	0x044 SPI_BMCLK
0x048	SPI_BMTXD	SPI 位模式发送数据值	0x048 SPI_BMTXD
0x04C	SPI_BMRXD	SPI 位模式接收数据值	0x04C SPI_BMRXD
0x050	SPI_BTR	SPI 突发传输配置	0x050 SPI_BTR
0x054	SPI_RCM	SPI 读命令模式	0x054 SPI_RCM
0x060	SPI_WRAP_LEN	SPI 读命令模式	0x060 SPI_WRAP_LEN
0x08C	SPI_IDMA_TXLEN	SPI IDMA 写传输长度	0x08C SPI_IDMA_TXLEN
0x090	SPI_IDMA_RXLEN	SPI IDMA 读传输长度	0x090 SPI_IDMA_RXLEN
0x094	SPI_IDMA_TXADDR	SPI IDMA 发送源地址	0x094 SPI_IDMA_TXADDR
0x098	SPI_IDMA_RXADDR	SPI IDMA 接收目的地址	0x098 SPI_IDMA_RXADDR

表 7–22 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x09C	SPI_IDMA_BTCFG	SPI IDMA 突发传输配置	0x09C SPI_IDMA_BTCFG
0x0A0	SPI_DISP_CTRL	SPI 显示刷屏模式控制	0x0A0 SPI_DISP_CTRL
0x0A4	SPI_DISP_HIGH	SPI 显示图像高度配置	0x0A4 SPI_DISP_HIGH
0x0A8	SPI_DISP_WIDTH	SPI 显示图像宽度配置	0x0A8 SPI_DISP_WIDTH
0x0AC	SPI_DISP_VSW_CMD	SPI 显示 VSW 行命令	0x0AC SPI_DISP_VSW_CMD
0x0B0	SPI_DISP_VBP_CMD	SPI 显示 VBP 行命令	0x0B0 SPI_DISP_VBP_CMD
0x0B4	SPI_DISP_VLD_CMD	SPI 显示 VLD 行命令	0x0B4 SPI_DISP_VLD_CMD
0x0B8	SPI_DISP_VFP_CMD	SPI 显示 VFP 行命令	0x0B8 SPI_DISP_VFP_CMD
0x0BC	SPI_DISP_RTIME	SPI 显示行附加延时	0x0BC SPI_DISP_RTIME
0x0C0	SPI_DISP_FTIME	SPI 显示帧延时	0x0C0 SPI_DISP_FTIME
0x0C4	SPI_DISP_TXFC	SPI 显示发送缓存控制	0x0C4 SPI_DISP_TXFC
0x200	SPI_TXD	SPI 发送数据	0x200 SPI_TXD
0x300	SPI_RXD	SPI 接收数据	0x300 SPI_RXD

7.3.7. 寄存器描述

7.3.7.1. 0x000 SPI_VER

默认值：0x00000303			SPI 控制器版本 (SPI Controller Version)
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x303	VER 版本号 **注：** 不同控制器的版本号可能不同。

7.3.7.2. 0x004 SPI_CFG

默认值：0x00000080			SPI 配置 (SPI Configuration)
位域	类型	默认值	描述
31	R/WAC	0x0	CTRL_RST 控制器复位（Controller Reset） 置 1，SPI 控制器复位，完成后自动清零。
30:9	–	–	–
8	R/W	0x0	DISP_EN 显示模式使能（Display Mode Enable） • 0：正常传输模式 • 1：使能 LCD 显示刷屏模式，自动发送一帧数据
7	R/W	0x1	RXFULL_STOP 传输停止使能（RX FIFO Full Transfer Stop Enable） • 0：忽略 RXFIFO 状态 • 1：当 RXFIFO 满时停止传输数据
6:5	–	–	–
4	R/W	0x0	TXIDMA_EN TX 通道内部 DMA 使能（TX Inner DMA Enable）
3	R/W	0x0	RXIDMA_EN RX 通道内部 DMA 使能（RX Inner DMA Enable）
2	R/W	0x0	AMOD 访问模式选择（Access Mode） • 0：AHB 访问模式 • 1：AXI 访问模式 **注：** 使能 XIP 功能，该位必须置 1。退出 XIP 功能，该位必须清零。
1	R/W	0x0	CTRL_MODE_SEL 工作模式选择（Controller Work Mode Select） • 0：从机模式 • 1：主机模式

默认值：0x00000080			SPI 配置 (SPI Configuration)
位域	类型	默认值	描述
			注： 当 START = 1 时，不能写入。
0	R/W	0x0	CTRL_EN 控制器使能（Controller Enable） • 0：禁用 • 1：使能 注： 当传输形式从位模式切换到字节模式时，必须使能一次。

7.3.7.3. 0x008 SPI_TCFG

默认值：0x00000087			SPI 传输配置 (SPI Transfer Configuration)
位域	类型	默认值	描述
31	R/WAC	0x0	START 启动数据传输（Start Data Transfer） • 0：空闲状态 • 1：启动数据传输 置 1，开始执行 SPI 数据传输，完成后自动清零。 注： 当 START=1 时不能写入。
30:29	R/W	0x0	RXDLY_EXT 采样时钟扩展延时 • 0：不延时 • 1：扩展延时 1.5 个时钟周期 • 2：扩展延时 2 个时钟周期

默认值：0x00000087			SPI 传输配置 (SPI Transfer Configuration)
位域	类型	默认值	描述
			• 3：无效 注： 在粗调模式下，该位域与该寄存器的 bit11 和 bit13 位延时累加，最大延时 3 个时钟周期。
28:27	–	–	–
26	R/W	0x0	SLV_OEN 从机模式输出使能（Slave Output Enable） 从机模式下，Dual /Quad I/O 模式输出使能，单线模式不需要配置 • 0：不使能输出，默认为输入 • 1：输出使能
25	R/W	0x0	3WIRE_EN 三线 byte 模式传输使能（3–Wire Byte Mode Transfer Enable） • 0：标准模式，MOSI 与 MISO 引脚独立 • 1：使能三线 byte 模式传输，此时 MOSI 与 MISO 引脚复用
24:15	–	–	–
14	R/W	0x0	TXDLY_EN 发送数据延时模式使能（TX Data Delay Mode Enable） • 0：正常发送 • 1：使能延时发送 注： 在 Dual / Quad 模式下，该位必须置 1，发送数据延时半个时钟周期。
13	R/W	0x0	RXDLY_DIS 接收数据延时采样模式配置（RX Data Delay Sample Mode Disable） • 0：延时采样模式 • 1：正常采样模式 在正常采样模式下，SPI 在边沿采样数据。在延时采样模式下，SPI 延迟半个时钟后在下一个边沿采样。
12	R/W	0x0	LSB_EN 最低有效位优先传输使能（Low Significant Bit Transmit Enable）

默认值：0x00000087			SPI 传输配置 (SPI Transfer Configuration)
位域	类型	默认值	描述
			• 0：最高有效位先传输 • 1：最低有效位先传输 **注：** 当 START=1 时不能写入。
11	R/W	0x0	RXINDLY_EN 接收数据内部采样点模式配置（RX Data Inner Sample Delay Mode Enable） • 0：内部读数据采样点不用延迟 • 1：内部读数据的采样点延迟半个时钟周期
10	R/W	0x0	HSWM 高速写模式配置（High Speed Write Mode Configuration） • 0：正常写模式 • 1：高速写模式，提前半个时钟周期切换到下一个数据 **注：** 当 START=1 时不能写入。
9	R/W	0x0	DMY_VAL 填充数据值配置（Dummy Byte Value Configuration） • 0：填充数据的值为 0x00 • 1：填充数据的值为 0xFF **注：** 当 START=1 时不能写入。
8	R/W	0x0	DINVD 丢弃无效数据（Discard Invalid Data） 发送数据时，设备端会主动发无效数据回来，此时如果不能及时读取 FIFO 的数据，该位需要使能，自动丢弃读到的无效数据。 • 0：接收所有的无效数据 • 1：丢弃所有的无效数据 **注：** 当 START=1 时不能写入。

默认值：0x00000087			SPI 传输配置 (SPI Transfer Configuration)
位域	类型	默认值	描述
7	R/W	0x1	CS_LEVEL 片选信号电平配置（CS Level Configuration） • 0：低电平 • 1：高电平 **注：** 当 START=1 时不能写入。
6	R/W	0x0	CS_CTL_SEL 片选信号控制模式选择（CS Control Mode Select） • 0：SPI 控制器 • 1：软件控制 通常由控制器自动发送 CS 信号，当该位置 1，必须手动配置 LEVEL 的电平状态。 **注：** 当 START=1 时不能写入。
5:4	R/W	0x0	CS_NUM 选中外部 SPI 设备的 CS 引脚序号（CS Pin Number） • 0：SPI_CS0 有效 • 1：SPI_CS1 有效 • 2–3：保留 **注：** 当 START=1 时不能写入。
3	R/W	0x0	CS_VALID_CTL 片选信号有效控制（CS Valid Mode Control） • 0：在 SPI 数据传输期间，片选信号一直有效 • 1：在 SPI 数据传输期间，传输暂停时片选信号无效 该位用于选择片选信号的输出波形形式。 **注：** 当 START=1 时不能写入，当 CS_CTL_SEL=0 时该位才有效。
2	R/W	0x1	CS_POL

默认值：0x00000087			SPI 传输配置 (SPI Transfer Configuration)
位域	类型	默认值	描述
			片选信号极性控制(CS Polarity Control) • 0：高电平有效 • 1：低电平有效 注： 当 START=1 时不能写入。
1	R/W	0x1	CPOL 时钟极性控制（Clock Polarity Control） • 0：空闲时为低电平 • 1：空闲时为高电平 注： 当 START=1 时不能写入。
0	R/W	0x1	CPHA 时钟 / 数据相位控制（Clock Phase Control） • 0：时钟的第一个边沿采样数据 • 1：时钟的第二个边沿采样数据 注： 当 START=1 时不能写入。

7.3.7.4. 0x00C SPI_PHACTRL

默认值：0x00000087			SPI 相位控制 (SPI Phase Control)
位域	类型	默认值	描述
31:16	–	–	–
15	R/W	0x0	TX_PHASE_MASK 屏蔽 TX 细调延时模式（Mask TX Fine Phase Delay Mode） • 0：TX 使用细调延时模式 • 1：TX 使用粗调延时模式
14	R/W	0x0	TXD_PHASE_SET

默认值：0x00000087			SPI 相位控制 (SPI Phase Control)
位域	类型	默认值	描述
			TX DATA 相位调节（TX Data Phase Set） • 0：TX Data 没有延时输出 • 1：TX Data 延时半个时钟周期输出
13	R/W	0x0	RXCK_DLY_EN RX 采样时钟粗调模式延时使能 • 0：禁用粗调模式 • 1：使能粗调模式 **注：** 默认禁用粗调模式，SPI_TCFG 寄存器的 bit11 和 bit13 配置无效。
12	R/W	0x0	RX_PHASE_MASK 屏蔽 RX 细调延时模式（Mask RX Fine Phase Delay Mode） • 0：RX 使用细调延时模式 • 1：RX 使用粗调延时模式
11:8	–	–	–
7:4	R/W	0x2	RXCK_PHASE_SET RX 采样时钟相位延时配置（RX Sample Clock Phase Set） • 0：0 度 • 1：90 度 • 2：180 度 • 3：270 度 • 4：360 度 • 5：450 度 • 6：540 度 • 7：630 度 • 8：720 度 • 9：810 度 • 10：900 度 • 11：990 度 • 12：1080 度

默认值：0x00000087			SPI 相位控制 (SPI Phase Control)
位域	类型	默认值	描述
3	–	–	–
2:0	R/W	0x0	TXCK_PHASE_SET TX SCLK 输出时钟相位延时配置（TX Clock Phase Set） • 0：0 度 • 1：90 度 • 2：180 度 • 3：270 度 • 4：360 度

7.3.7.5. 0x010 SPI_ICR

默认值：0x00000000			SPI 中断控制 (SPI Interrupt Control)
位域	类型	默认值	描述
31:29	–	–	–
28	R/W	0x0	DISP_DLESS_INTE 显示刷屏模式行传输缺数中断使能
27	R/W	0x0	DISP_DTO_INTE 显示刷屏模式传输完成中断使能
26	R/W	0x0	DISP_ERR_INTE 显示刷屏模式错误中断使能
25:24	–	–	–
23	R/W	0x0	TXIDMA_DONE_INTE 内部 DMA TX 传输完成中断使能
22	R/W	0x0	RXIDMA_DONE_INTE 内部 DMA RX 传输完成中断使能
21	R/W	0x0	RXFIFO_UNALIGN_INTE 内部 DMA 模式下，RX FIFO 数据非 4byte 对齐中断使能
20	R/W	0x0	IDMA_ERROR_INTE 内部 DMA 总线错误中断使能

默认值：0x00000000			SPI 中断控制 (SPI Interrupt Control)
位域	类型	默认值	描述
19	R/W	0x0	AXI_TRAN_DONE_INTE AXI 传输完成中断使能
18	R/W	0x0	AXI_CFG_ERROR_INTE AXI 配置错误中断使能
17	R/W	0x0	AXI_TRAN_ERROR_INTE AXI 传输错误中断使能
16	R/W	0x0	AHB_TRAN_ERROR_INTE AHB 传输错误中断使能
15:14	–	–	–
13	R/W	0x0	CS_INTE CS 无效中断使能（CS Invalid Interrupt Enable） 片选信号从有效状态变到无效状态。 • 0：禁用 • 1：使能
12	R/W	0x0	TD_INTE 传输完成中断使能（Transfer Done Interrupt Enable） • 0：禁用 • 1：使能
11	R/W	0x0	TF_UDR_INTE TX FIFO 下溢出中断使能（TX FIFO Underrun Interrupt Enable） • 0：禁用 • 1：使能 当 CPU / DMA 读取空的 TX FIFO 会产生此中断。
10	R/W	0x0	TF_OVF_INTE TX FIFO 上溢出中断使能（TX FIFO Overflow Interrupt Enable） • 0：禁用 • 1：使能 当 CPU / DMA 写入满的 TX FIFO 会产生此中断。
9	R/W	0x0	RF_UDR_INTE RX FIFO 下溢出中断使能（RX FIFO Underrun Interrupt Enable）

默认值：0x00000000			SPI 中断控制 (SPI Interrupt Control)
位域	类型	默认值	描述
			• 0：禁用 • 1：使能 当 CPU / DMA 读取空的 RX FIFO 会产生此中断。
8	R/W	0x0	RF_OVF_INTE RX FIFO 上溢出中断使能（RX FIFO Overflow Interrupt Enable） • 0：禁用 • 1：使能 当 CPU / DMA 写入满的 RX FIFO 会产生此中断。
7	–	–	–
6	R/W	0x0	TF_FUL_INTE TX FIFO 满中断使能（TX FIFO Full Interrupt Enable） • 0：禁用 • 1：使能
5	R/W	0x0	TF_EMP_INTE TX FIFO 为空中断使能（TX FIFO Empty Interrupt Enable） • 0：禁用 • 1：使能
4	R/W	0x0	TF_RDY_INTE TX FIFO 准备好中断使能（TX FIFO Ready Interrupt Enable） • 0：禁用 • 1：使能
3	–	–	–
2	R/W	0x0	RF_FUL_INTE RX FIFO 满中断使能（RX FIFO Full Interrupt Enable） • 0：禁用 • 1：使能
1	R/W	0x0	RF_EMP_INTE RX FIFO 为空中断使能（RX FIFO Empty Interrupt Enable）

默认值：0x00000000			SPI 中断控制 (SPI Interrupt Control)
位域	类型	默认值	描述
			• 0：禁用 • 1：使能
0	R/W	0x0	RF_RDY_INTE RX FIFO 准备好中断使能（RX FIFO Ready Interrupt Enable） • 0：禁用 • 1：使能

7.3.7.6. 0x014 SPI_ISTS

默认值：0x00000032			SPI 中断状态 (SPI Interrupt Status)
位域	类型	默认值	描述
31:29	–	–	–
28	R/W	0x0	DISP_DLESS 显示刷屏模式行传输缺数状态
27	R/W	0x0	DISP_DTO 显示刷屏模式传输完成中断
26	R/W	0x0	DISP_ERR 显示刷屏模式错误中断
25:24	–	–	–
23	R/W1C	0x0	TXIDMA_DONE 内部 DMA TX 完成信号中断 置 1，表示内部 DMA 发送数据传输完成。
22	R/W1C	0x0	RXIDMA_DONE 内部 DMA RX 完成信号中断 置 1，表示内部 DMA 接收数据传输完成。
21	R/W1C	0x0	RXFIFO_UNALIGN RX FIFO 非对齐中断 内部 DMA 模式下，该位置 1，表示 RX FIFO 数据非 4byte 对齐。
20	R/W1C	0x0	IDMA_ERROR IDMA 错误中断

默认值：0x00000032			SPI 中断状态 (SPI Interrupt Status)
位域	类型	默认值	描述
			置 1，表示内部 DMA 发生总线错误状态。
19	R/W1C	0x0	AXI_TRAN_DONE AXI 传输完成中断 置 1，表示 AXI 传输完成。此时可结束 AXI 传输，开启 AHB 传输。
18	R/W1C	0x0	AXI_CFG_ERROR AXI 配置错误中断 置 1，表示 AXI 模式下寄存器配置错误。此时需立刻结束 AXI 传输，重新配置寄存器。
17	R/W1C	0x0	AXI_TRAN_ERROR AXI 传输错误中断 置 1，表示 AHB 正在传输时，发起 AXI 传输请求。此时需立刻结束 AHB 传输，启动 AXI 传输。
16	R/W1C	0x0	AHB_TRAN_ERROR AHB 传输错误中断 置 1，表示 AXI 正在传输时，发起 AHB 传输请求。此时需立刻结束 AXI 传输，启动 AHB 传输。
15:14	–	–	–
13	R/W1C	0x0	CS_INV 片选信号无效中断（CS Invalid Interrupt） 置 1，表示片选状态从有效变为无效。
12	R/W1C	0x0	TD 传输完成中断（Transfer Done Interrupt） • 0：正在传输 • 1：传输完成
11	R/W1C	0x0	TF_UDF TX FIFO 下溢出中断（TX FIFO Underrun Interrupt） • 0：TXFIFO 没有下溢出 • 1：TXFIFO 下溢出
10	R/W1C	0x0	TF_OVF TX FIFO 上溢出中断（TX FIFO Overflow Interrupt）

默认值：0x00000032			SPI 中断状态 (SPI Interrupt Status)
位域	类型	默认值	描述
			• 0：TXFIFO 没有上溢出 • 1：TXFIFO 上溢出
9	R/W1C	0x0	RF_UDF RX FIFO 下溢出中断（RX FIFO Underrun Interrupt） • 0：RXFIFO 没有下溢出 • 1：RXFIFO 下溢出
8	R/W1C	0x0	RF_OVF RX FIFO 上溢出中断（RX FIFO Overflow Interrupt） • 0：RXFIFO 没有上溢出 • 1：RXFIFO 上溢出
7	–	–	–
6	R/W1C	0x0	TF_FULL TX FIFO 满中断（TX FIFO Full Interrupt） • 0：TXFIFO 未满 • 1：TXFIFO 已满
5	R/W1C	0x1	TF_EMP TX FIFO 为空中断（TX FIFO Empty Interrupt） • 0：TXFIFO 非空 • 1：TXFIFO 为空
4	R/W1C	0x1	TF_READY TX FIFO 准备好中断（TX FIFO Ready Interrupt） • 0：TX_WL > TX_WATER_MARK • 1：TX_WL <= TX_WATER_MARK 该位置 1，CPU 可以向 TX FIFO 写入数据。
3	–	–	–
2	R/W1C	0x0	RF_FULL RX FIFO 满中断（RX FIFO Full Interrupt） • 0：RXFIFO 未满 1：RXFIFO 已满

默认值：0x00000032			SPI 中断状态 (SPI Interrupt Status)
位域	类型	默认值	描述
1	R/W1C	0x1	RF_EMP RX FIFO 为空中断（RX FIFO Empty Interrupt） • 0：RXFIFO 非空 • 1：RXFIFO 为空
0	R/W1C	0x0	RF_READY RX FIFO 准备好中断（RX FIFO Ready Interrupt） • 0：RX_WL < RX_WATER_MARK • 1：RX_WL >= RX_WATER_MARK 该位置 1，CPU 可以从 RX FIFO 读出数据。

7.3.7.7. 0x018 SPI_FCTL

默认值：0x00400001			SPI FIFO 控制 (SPI FIFO Control)
位域	类型	默认值	描述
31	R/WAC	0x0	TF_RST TX FIFO 复位，完成后自动清零。
30:29	–	–	–
28	R/W	0x0	DREQ_PAUSE DMA 请求暂停 **注：** 用于固定传输长度模式下，当收到 GPDMA 的传输完成信号，暂停 DREQ 请求，在下一笔启动传输后恢复发送请求。
27:25	–	–	–
24	R/W	0x0	TF_DREQ_EN TX FIFO DMA 请求使能 • 0：禁用 • 1：使能
23:16	R/W	0x40	TF_WATER_MARK

默认值：0x00400001			SPI FIFO 控制 (SPI FIFO Control)
位域	类型	默认值	描述
			TX FIFO 发出写请求信号的 FIFO 水位值，当 FIFO 水位小于该值，并且 TF_DREQ_EN=1, 向 DMA 发起写请求。
15	R/WAC	0x0	RF_RST RX FIFO 复位，完成后自动清零。
14:9	–	–	–
8	R/W	0x0	RF_DREQ_EN RX FIFO DMA 请求使能。 • 0：禁用 • 1：使能
7:0	R/W	0x1	RF_WATER_MARK RX FIFO 发出读请求信号的 FIFO 水位值，当 FIFO 水位大于该值，并且 RF_DREQ_EN=1, 向 DMA 发起读请求。

7.3.7.8. 0x01C SPI_FSTS

默认值：0x00000000			SPI FIFO 状态 (SPI FIFO Status)
位域	类型	默认值	描述
31	R	0x0	TF_WBUF_STS TX FIFO 写缓存状态（TX FIFO Write Buffer Status） 置 1，表示写缓存正在执行写操作。
30:24	R	0x0	TF_WBUF_CNT 写缓存计数器（TX FIFO Write Buffer Counter） 表示 TX FIFO 写缓存中字节的数量。 **注：** 注意：QSPI0/ 1/ 4/ 5 最大 4 个字节，QSPI2 /3 最大 64 个字节
24	–	–	–
23:16	R	0x0	TF_CNT TX FIFO 计数器（TX FIFO Counter） • 0：在 TX FIFO 中有 0 字节数据 • 1：在 TX FIFO 中有 1 字节数据

默认值：0x00000000			SPI FIFO 状态 (SPI FIFO Status)
位域	类型	默认值	描述
			• … • 64：在 TX FIFO 中有 64 字节数据
15	R	0x0	RF_RBUF_STS RX FIFO 读缓存状态（RX FIFO Read Buffer Status） 置 1，表示读缓存正在执行读操作。
14:8	R	0x0	RF_RBUF_CNT 读缓存计数器（RX FIFO Read Buffer Counter） 表示 RX FIFO 读缓存中字节的数量。
7:0	R	0x0	RF_CNT RX FIFO 计数器（RX FIFO Counter） • 0：在 RX FIFO 中有 0 字节 • 1：在 RX FIFO 中有 1 字节 • … • 64：在 RX FIFO 中有 64 字节

7.3.7.9. 0x024 SPI_CCFG

默认值：0x00000002			SPI 时钟配置 (SPI Clock Configuration)
位域	类型	默认值	描述
31:13	–	–	–
12	R/W	0x0	CKDIV_SEL 时钟分频器选择（Clock Divider Select） • 0：选择时钟分频器 1 • 1：选择时钟分频器 2
11:8	R/W	0x0	CKDIV1 时钟分频器 1（Clock Divider 1） SPI 接口时钟频率根据如下公式计算 SPI_CLK = 源时钟频率/(2^CKDIV1)
7:0	R/W	0x2	CKDIV2 时钟分频器 2（Clock Divider 2） SPI 接口时钟频率根据如下公式计算：

默认值：0x00000002			SPI 时钟配置 (SPI Clock Configuration)
位域	类型	默认值	描述
			SPI_CLK = 源时钟频率/(2*(CKDIV2 + 1))

7.3.7.10. 0x028 SPI_MISC

默认值：0x10000000			SPI 其它功能配置 (SPI Miscellaneous Confiuration)
位域	类型	默认值	描述
31:28	R/W	0x1	DTR_SDRC 双边沿传输模式下单沿传输的数据量
27:25	–	–	–
24	R/W	0x0	DTR_EN 使能双边沿传输模式
23:20	R/W	0x0	DTR_DUMMY 双边沿传输模式下 Dummy 时钟周期数
19:0	–	–	–

7.3.7.11. 0x02C SPI_DDC

默认值：0x00000002			SPI 双边沿模式 Delay Chain 配置(SPI DTR Delay Chain Configuration)
位域	类型	默认值	描述
31:17	–	–	–
16:12	R/W	0x0	SAMP_DLY 采样时钟 Delay Chain 配置
11:9	–	–	–
8:4	R/W	0x0	SCK_DLY 输出时钟 Delay Chain 配置
3:2	–	–	–
1	R/W	0x0	SAMP_DLYC_EN 采样时钟 Delay Chain 使能

默认值：0x00000002			SPI 双边沿模式 Delay Chain 配置(SPI DTR Delay Chain Configuration)
位域	类型	默认值	描述
0	R/W	0x0	SCK_DLYC_EN 输出时钟 Delay Chain 使能

7.3.7.12. 0x030 SPI_TBC

默认值：0x00000000			SPI 总传输字节计数器 (SPI Total Bytes Counter)
位域	类型	默认值	描述
31	R/W	0x0	MBC_DEC_EN 传输数据计数器递减使能 • 0：传输数据计数器寄存器值固定，下次传输无需配置 • 1：传输数据计数器寄存器递减为 0，下次传输需重新配置 **注：** 注意这里的传输数据计数器包括：TB_CNT, TXD_CNT, DMY_CNT, STXD_CNT。
30:24	–	–	–
23:0	R/W	0x0	TB_CNT 总传输字节计数器（Total Bytes Counter） 该值定义了传输的字节总数，包括发送，接收和填充数据。 • 0：共传输 0 个字节 • 1：共传输 1 个字节 • … • N：共传输 N 个字节 在 DTR 模式下，传输数据总数 = 发送数据 + 接收数据 **注：** 传输数据总数=发送数据+接收数据+填充数据

7.3.7.13. 0x034 SPI_TWC

默认值：0x00000000			SPI 写发送计数器 (SPI Transmit Write Counter)
位域	类型	默认值	描述
31:24	–	–	–
23:0	R/W	0x0	TXD_CNT 写数据计数器（TX Data Counter） 该值定义了在填充数据之前发送到 TX FIFO 的写数据字节总数，为了节约带宽，填充数据由控制器自动发送。 • 0：共发送 0 个字节 • 1：共发送 1 个字节 • … • N：共发送 N 个字节

7.3.7.14. 0x038 SPI_TMC

默认值：0x00000000			SPI 传输控制 (SPI Transmit Miscellaneous Control)
位域	类型	默认值	描述
31	R/W	0x0	QPI_EN QPI 模式使能（QPI Transfer mode Enable） • 0：命令使用 1 线模式传输 • 1：命令使用 4 线模式传输 QPI 模式下，命令/地址/数据都使用 4 线模式传输。 注： 仅当 QUAD_EN=1 时有效。
30	R/W	0x0	QADDR_EN QIO 模式使能（QIO Transfer mode Enable） • 0：地址使用 1 线模式传输 • 1：地址使用 4 线模式传输 QIO 模式下，该位置 1，命令使用 1 线传输，地址/数据使用 4 线传输。 注： 仅当 QUAD_EN=1 时有效。

默认值：0x00000000			SPI 传输控制 (SPI Transmit Miscellaneous Control)
位域	类型	默认值	描述
29	R/W	0x0	QUAD_EN 四线数据传输模式使能（Quad Transfer Mode Enable） • 0：4 线数据模式禁用 • 1：4 线数据模式使能 注： 4 线数据模式包括 4 线输入和 4 线输出。
28	R/W	0x0	DUAL_EN 双线数据传输模式使能（Dual Transfer Mode Enable） • 0：发送或接收数据使用单线数据传输模式 • 1：发送或接收数据使用双线数据传输模式 注： 当 START=1 时不能写入，仅当 QUAD_EN=0 时有效。
27:24	R/W	0x0	DMY_CNT 填充数据计数器（Dummy Bytes Counter） 定义了待发送的填充数据的字节数，用于双路模式和四路模式。 • 0：共发送 0 个字节填充数据 • 1：共发送 1 个字节填充数据 • … • N：共发送 N 个字节填充数据 在 DTR 模式下，该值为时钟周期数 注： 当 START=1 时不能写入。
23:0	R/W	0x0	STXD_CNT 单线发送数据计数器（Single TX Data Counter） 该值定义了需要使用单线模式发送的数据长度。 • 对于单线模式：包括命令、地址、数据。 • 对于双线和四线模式：仅包括命令、地址。 • 对于双线 IO 模式：仅包括命令。

默认值：0x00000000			SPI 传输控制 (SPI Transmit Miscellaneous Control)
位域	类型	默认值	描述
			• 0：单线共发送 0 个字节 • 1：单线共发送 1 个字节 • … • N：单线共发送 N 个字节 注： 当 START=1 时不能写入。

7.3.7.15. 0x040 SPI_BMTC

默认值：0x000000A0			SPI 位模式传输配置 (SPI Bit-Mode Transfer Configuration)
位域	类型	默认值	描述
31	R/WAC	0x0	BM_START 启动传输数据（Bit-Mode Start Data Transfer） • 0：空闲状态 • 1：启动传输数据 该字段置 1，启动发送串行数据流，数据存放在 (SPI_BMTXD 和 SPI_BMRXD 寄存器），传输完成后自动清零。
30	R/W	0x0	BM_SMP_SEL 采样模式选择（Bit-Mode Sample Mode Select） • 0：延时采样模式 • 1：标准采样模式 在标准采样模式下，SPI 主机在 SCLK 的上升边沿采样数据。 在延时采样模式下，SPI 主机延时半个时钟周期，在 SCLK 的上升边沿采样数据。
29:26	–	–	–
25	R/W1C	0x0	BM_TD 传输完成（Bit-Mode Transfer Done） • 0：忙等待 • 1：传输完成 置 1，表示最后一位数据传输完成，该字段写 1 清零。

默认值：0x000000A0			SPI 位模式传输配置 (SPI Bit-Mode Transfer Configuration)
位域	类型	默认值	描述
			注： 传输完成后，该位需要重复写 1 清零，检查该位是否为 0，直到清零后再启动下一笔传输。
24	R/W	0x0	BM_TD_INTE 传输完成中断使能（Bit-Mode Transfer Done Interrupt Enable）。 • 0：禁用 • 1：使能 **注：** 当总线模式选择 2 /3 时有效。
23	R/W	0x0	BM_CPOL 时钟极性控制（Clock Polarity Control） • 0：空闲时为高电平 • 1：空闲时为低电平 **注：** 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
22	R/W	0x0	BM_CPHA 时钟 / 数据相位控制（Clock Phase Control） • 0：时钟的第一个边沿采样数据 • 1：时钟的第二个边沿采样数据 **注：** 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
21:16	R/W	0x0	BM_RXCNT 接收串行数据长度（Bit-Mode RX Data Length） • 000000b：接收 0 比特 • 000001b：接收 1 比特 • … • 100000b：接收 32 比特

默认值：0x000000A0			SPI 位模式传输配置 (SPI Bit-Mode Transfer Configuration)
位域	类型	默认值	描述
			注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
15:14	–	–	–
13:8	R/W	0x00	BM_TXCNT 发送串行数据长度（Bit-Mode TX Data Length） • 000000b：发送 0 比特 • 000001b：发送 1 比特 • … • 100000b：发送 32 比特 注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
7	R/W	0x1	BMCS_LEVEL 片选信号电平配置 • 0：配置片选信号为低电平 • 1：配置片选信号为高电平 注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
6	R/W	0x0	BMCS_CTRL_SEL 片选信号输出控制方式选择 • 0：SPI 控制器 • 1：软件控制 通常控制器会自动控制片选信号高低电平。当该位置 1，则需要通过软件编程写入 SPI_TCFG.CS_LEVEL 的值，来控制片选信号。 注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
5	R/W	0x1	BMCS_POL 片选信号极性控制

默认值：0x000000A0			SPI 位模式传输配置 (SPI Bit-Mode Transfer Configuration)
位域	类型	默认值	描述
			• 0：高电平有效 • 1：低电平有效 注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
4	–	–	–
3:2	R/W	0x0	BMCS_NUM 片选信号引脚序号选择 • 0：SPI_CS0 选中 • 1–3：保留 注： 当总线模式选择 2 /3 时有效，当 BM_START=1 时不能写入。
1:0	R/W	0x0	BMOD_SEL 总线模式选择（Bus Mode Select） • 0：数据流按字节对齐，工作在标准 SPI，双路输入 /双路输出 SPI，双路 IO SPI，或四路输入/四路输出 SPI • 1：预留 • 2：数据流按位对齐，工作在三线模式下 • 3：数据流按位对齐，工作在标准 SPI 模式下

7.3.7.16. 0x044 SPI_BMCLK

默认值：0x02000000			SPI 位模式时钟配置 (SPI Bit-Mode Clock Configuration)
位域	类型	默认值	描述
31:29	–	–	–
28	R/W	0x0	BMRX_PHASE_MASK 位模式屏蔽 RX 细调延时模式（Mask RX Fine Phase Delay Mode） • 0：RX 使用细调延时模式 • 1：RX 使用粗调延时模式
27:24	R/W	0x2	BMRXCK_PHASE_SET

默认值：0x02000000			SPI 位模式时钟配置 (SPI Bit-Mode Clock Configuration)
位域	类型	默认值	描述
			位模式 RX 采样时钟相位延时配置（RX Sample Clock Phase Set） • 0：0 度 • 1：90 度 • 2：180 度 • 3：270 度 • 4：360 度 • 5：450 度 • 6：540 度 • 7：630 度 • 8：720 度
23	–	–	–
22:20	R/W	0x0	BMTXCK_PHASE_SET 位模式 TX 输出时钟相位配置（Bitmode TX Clock Phase Set） • 0：无延时 • 1：90 度 • 2：180 度 • 3：270 度 • 4：360 度
19	R/W	0x0	BMTXD_PHASE_SET 位模式 TX 数据输出相位配置（Bitmode TX Data Phase Set） • 0：无延时 • 1：延时 0.5 个时钟周期输出
18:8	–	–	–
7:0	R/W	0x0	BM_CKDIV 位模式时钟分频比例（Bit-Mode Clock Divide） 接口时钟频率计算公式如下： SPI_CLK = 源时钟/(2*(BM_CKDIV + 1))

7.3.7.17. 0x048 SPI_BMTXD

默认值：0x00000000			SPI 位模式发送数据值 (SPI Bit-Mode TX Data)
位域	类型	默认值	描述
31:0	R/W	0x0	BMTXD 位模式发送数据值 该寄存器用来存储发送的数据 注： 在传输过程中，最高有效位先发送。

7.3.7.18. 0x04C SPI_BMRXD

默认值：0x00000000			SPI 位模式接收数据值 (SPI Bit-Mode RX Data)
位域	类型	默认值	描述
31:0	R/W	0x0	BMRXD 位模式接收数据值 该寄存器用来存储接收的数据 注： 在传输过程中，最高有效位先接收。

7.3.7.19. 0x050 SPI_BTR

注：

该寄存器仅在 XIP 模式下配置有效。

默认值：0x77304010			SPI 突发传输配置 (SPI Burst Set)
位域	类型	默认值	描述
31:24	R/W	0x77	CMD_INDEX SPI 命令指令值 SPI 在 XIP 模式下，发送的 Wrap Burst 命令，支持 0x77 和 0xC0 两种典型值。当 QPI_EN=1，CMD 使用 4 线传输。否则，CMD 使用 1 线传输。

默认值：0x77304010			SPI 突发传输配置 (SPI Burst Set)
位域	类型	默认值	描述
			注： 0x77 命令仅支持 QIO 模式。0xC0 命令支持标准单线模式或 QPI 模式。
23:20	R/W	0x3	DUMMY_BYTE 传输配置填充数据字节数 定义 dummy 的字节数，为固定值或取决于工作时钟频率，可根据 WRAP 的值配置。
19	R/W	0x0	WIDTH 填充数据和有效数据位宽 Dummy 和数据位宽配置寄存器，在 QPI 模式下无效。 • 该位为 0，默认为 4–bit 位宽。 • 该位为 1，配置为 1–bit 位宽。 **注：** 当 burst 命令为 0xC0，工作在 QIO 模式下，该位必须配置为 1。 **工作在 QPI 模式下，该位不用配置。**
18	R/W	0x0	AUTO_WRAP_LEN 自动 Wrap 长度 • 0：wrap 长度为固定值，通过 WRAPPED 位域获取设定值。 • 1：wrap 长度可变，由硬件根据 SPI_WLEN 寄存器获取设定值，自动配置 wrap 长度。
17	–	–	–
16	R/W	0x0	WRAP_EN Wrap 模式使能 当系统开启 Cache 功能，该位必须使能。
15:8	R/W	0x40	SPI_BURST_WRAPPED Wrap 模式指令值 SPI 模式 XIP 使能时，若 AXI burst type 是 linear 切换 wrapped，触发此命令发送。Wrap 的值可以确定突发传输的长度和 dummy 的字节数，根据厂家的规格书进行设置。
7:0	R/W	0x10	SPI_BURST_LINEAR 线性模式指令值 SPI 模式 XIP 使能时，若 AXI burst type 是 wrapped 切换 linear，触发此命令发送

7.3.7.20. 0x054 SPI_RCM

注：

该寄存器仅在 XIP 模式下配置有效。

默认值：0xEB20A000			SPI 读命令模式 (SPI Read Command Mode)
位域	**类型**	**默认值**	**描述**
31:24	R/W	0xEB	CMD_INDEX 读命令指令值 SPI 在 XIP 模式下，定义读命令序号。
23:20	R/W	0x2	DUMMY_BYTE 读操作填充数据字节数 SPI 在 XIP 模式下，定义 dummy 时钟周期的个数。对于不同颗粒该值可能为固定值，或根据工作频率进行选择。
19	–	–	–
18	R/W	0x0	ADDR_4BYTE_EN4 字节地址模式使能 SPI 在 XIP 模式下，使能地址位采用 4 bytes 模式传输
17	R/W	0x0	READ_MODE_BYTE_ENSPI 读命令模式位使能 该位置 1，地址后面增加 1 byte 模式位，模式位的线宽与地址位一致，模式位的设定值由 RDCMD_BYPASS_EN 决定。 **注：** SPI 工作在 Dual I/O，Quad I/O，QPI 模式下时，用户可以配置该位使能，否则当作 dummy 处理。
16	R/W	0x0	RDCMD_BYPASS_ENSPI 读命令 bypass 模式使能 该位置 1，模式位的设定值为 RDCMD_BYPPASS_CODE，否则，模式位的设定值为 RDCMD_NORMAL_CODE。
15:8	R/W	0xA0	RDCMD_BYPASS_CODE 读命令 bypass 模式指令值 在 bypass 模式下，第一次需要发送读命令，之后的连续读操作不需要再次发送读命令。
7:0	R/W	0x00	RDCMD_NORMAL_CODE

默认值：0xEB20A000			SPI 读命令模式 (SPI Read Command Mode)
位域	类型	默认值	描述
			读命令正常模式指令值 在正常模式下，每次读操作都需要发送读命令。

7.3.7.21. 0x060 SPI_WRAP_LEN

注：

该寄存器仅在 XIP 模式下配置有效。该寄存器的值与 Wrap 命令的值有关，0x77 和 0xC0 命令对应的值不同，有些器件还包含 dummy 长度的配置信息，该寄存器需要在 XIP 传输之前配置好。

默认值：0x03020100			SPI 读命令模式 (SPI Wrap Length Config)
位域	类型	默认值	描述
31:24	R/W	0x03	WRAP64 Wrap 长度为 64 字节对应的命令参数
23:16	R/W	0x02	WRAP32 Wrap 长度为 32 字节对应的命令参数
15:8	R/W	0x01	WRAP16 Wrap 长度为 16 字节对应的命令参数
7:0	R/W	0x00	WRAP08 Wrap 长度为 8 字节对应的命令参数

7.3.7.22. 0x08C SPI_IDMA_TXLEN

注：

该寄存器仅在 IDMA 模式下配置有效。

默认值：0x00000000			SPI IDMA 写传输长度 (SPI Inner DMA Write Transaction Length)
位域	类型	默认值	描述
31:24	–	–	–
23:0	R/W	0x00	IDMA_TXLEN 内部 DMA 发送的数据长度

默认值：0x00000000			SPI IDMA 写传输长度 (SPI Inner DMA Write Transaction Length)
位域	类型	默认值	描述
			内部 DMA 通过 AHB 总线发送数据到 SPI 控制器的写缓存。

7.3.7.23. 0x090 SPI_IDMA_RXLEN

注：

该寄存器仅在 IDMA 模式下配置有效。

默认值：0x00000000			SPI IDMA 读传输长度 (SPI Inner DMA Read Transaction Length)
位域	类型	默认值	描述
31:24	–	–	–
23:0	R/W	0x00	IDMA_RXLEN 内部 DMA 接收的数据长度 内部 DMA 通过 AHB 总线从 SPI 控制器的读缓存接收数据。

7.3.7.24. 0x094 SPI_IDMA_TXADDR

注：

该寄存器仅在 IDMA 模式下配置有效。

默认值：0x00000000			SPI IDMA 发送源地址 (SPI Inner DMA TX Source Address)
位域	类型	默认值	描述
31:0	R/W	0x0	IDMA_TXADDR 内部 DMA 发送数据的源地址

7.3.7.25. 0x098 SPI_IDMA_RXADDR

注：

该寄存器仅在 IDMA 模式下配置有效。

默认值：0x00000000			SPI IDMA 接收目的地址 (SPI Inner DMA RX Destination Address)
位域	类型	默认值	描述
31：0	R/W	0x0	IDMA_RXADDR 内部 DMA 接收数据的目标地址

7.3.7.26. 0x09C SPI_IDMA_BTCFG

注：

该寄存器仅在 IDMA 模式下配置有效。

默认值：0x00004A04			SPI IDMA 突发传输配置 (SPI Inner DMA Burst Config)
位域	类型	默认值	描述
31:15	–	–	–
14	R/W	0x1	RX_UNALIGN_EN RX 非 4 字节对齐传输使能
13	R/W	0x0	RX_PRI_EN RX 优先传输使能
12	R/W	0x0	AUTO_LEN 自动突发传输长度配置使能 开启后控制器会根据 Buffer 空满状态自动调整突发传输长度。
11:10	R/W	0x2	RX_BURST_LEN RX 突发传输长度配置 • 0： 突发传输 4 字节 • 1： 突发传输 16 字节 • 2：突发传输 32 字节 • 3：突发传输 64 字节
9:8	R/W	0x2	TX_BURST_LEN TX 突发传输长度配置 • 0： 突发传输 4 字节 • 1： 突发传输 16 字节 • 2：突发传输 32 字节 • 3：预留

默认值：0x00004A04			SPI IDMA 突发传输配置 (SPI Inner DMA Burst Config)
位域	类型	默认值	描述
7:0	–	–	–

7.3.7.27. 0x0A0 SPI_DISP_CTRL

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示刷屏模式控制 (SPI Display Mode Control)
位域	类型	默认值	描述
31	R/W	0x0	DISP_RESTART 刷屏模式启动连续帧传输 • 0： 如果正在传输，则完成当前帧后，停止传输 • 1： 启动连续帧传输 注： 该位启动连续帧传输，硬件根据 IDMA_TXADDR 寄存器自动切换 buffer 地址。
30:24	–	–	–
23:20	R/W	0x0	DISP_DMC 刷屏模式插入的 dummy 字节数 注： 对于一些 QSPI LCD 屏，使用 RGB565 格式的图像数据工作在 Quad 模式时，CMD 和 DATA 之间需要插入指定数量的 dummy 时钟，该寄存器单位为字节数，对应 2 个 dummy 时钟。
19:16	R	0x0	DISP_STATE 刷屏模式传输状态机状态
15:12	–	–	–
11:10	R/W	0x0	DISP_VLDL_DRQ_SEL 刷屏模式有效行 IDMA 请求模式选择 • 0： 行间隔内不发送 IDMA 请求，不等待 txfifo 达到水位直接发起新的行传输 • 1： 行间隔内继续发送 IDMA 请求，等待 txfifo 达到水位后再发起新的行传输

默认值：0x00000000			SPI 显示刷屏模式控制 (SPI Display Mode Control)
位域	类型	默认值	描述
			• 2： 行间隔内不发送 IDMA 请求，等待 txfifo 达到水位后再发起新的行传输 • 3：保留值，配置无效
9	R/W	0x0	DISP_PARA_DEC_EN 行参数寄存器递减使能 • 0： 行参数寄存器值固定，不随计数器递减 • 1： 行参数寄存器随计数器递减到 0，如果使用 DISP_RESTART 启动刷屏模式，则下一帧传输时自动赋初值
8	R/W	0x0	DISP_MTW_EN 刷屏模式命令多线模式使能 • 0： 命令使用单线传输 • 1： 命令使用多线传输，与数据线宽一致
7:4	–	–	–
3:0	R/W	0x0	DISP_PROC_EN 刷屏模式传输行使能 每个位表示对应的传输行，置 1 使能该行传输。 bit0 对应 VSW 行传输使能。 bit1 对应 VBP 行传输使能。 bit2 对应有效行传输使能。 bit3 对应 VFP 行传输使能。

7.3.7.28. 0x0A4 SPI_DISP_HIGH

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示图像高度配置 (SPI Display High Configuration)
位域	类型	默认值	描述
31:24	R/W	0x0	DISP_VFP_HIGH 显示图像 VFP 区高度
23:16	R/W	0x0	DISP_VBP_HIGH 显示图像 VBP 区高度

默认值：0x00000000			SPI 显示图像高度配置 (SPI Display High Configuration)
位域	类型	默认值	描述
15:12	–	–	–
11:0	R/W	0x0	DISP_VLD_HIGH 显示图像有效区高度

7.3.7.29. 0x0A8 SPI_DISP_WIDTH

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示图像宽度配置 (SPI Display Width Configuration)
位域	类型	默认值	描述
31	–	–	–
30:16	R/W	0x0	DISP_STRIDE_LEN 显示图像对齐的行宽度，单位为字节，长度需要 4 字节对齐
15	–	–	–
14:0	R/W	0x0	DISP_LEN 显示图像行宽度，单位为字节，长度不需要 4 字节对齐

7.3.7.30. 0x0AC SPI_DISP_VSW_CMD

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示 VSW 行命令 (SPI Display VSW Row Command)
位域	类型	默认值	描述
31:0	R/W	0x0	VSW_CMD 表示每一帧开始行的 VSW 命令，硬件自动获取填入 FIFO

7.3.7.31. 0x0B0 SPI_DISP_VBP_CMD

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示 VBP 行命令 (SPI Display VBP Row Command)
位域	类型	默认值	描述
31:0	R/W	0x0	VBP_CMD 表示 VBP 区域行命令，硬件自动获取填入 FIFO

7.3.7.32. 0x0B4 SPI_DISP_VLD_CMD

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示 VLD 行命令 (SPI Display VLD Row Command)
位域	类型	默认值	描述
31:0	R/W	0x0	VLD_CMD 表示 VLD 区域行命令，硬件自动获取填入 FIFO

7.3.7.33. 0x0B8 SPI_DISP_VFP_CMD

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示 VFP 行命令 (SPI Display VFP Row Command)
位域	类型	默认值	描述
31:0	R/W	0x0	VFP_CMD 表示 VFP 区域行命令，硬件自动获取填入 FIFO

7.3.7.34. 0x0BC SPI_DISP_RTIME

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示行附加延时 (SPI Display Row Additional Time)
位域	类型	默认值	描述
31:0	R/W	0x0	DISP_RTIME 行附加时间 表示图像的 VSW /VBP /VFP 区域行命令之间需要填充的时间，该时间应大于等于有效图像区的行传输时间，单位为 AHB 的时钟周期。

7.3.7.35. 0x0C0 SPI_DISP_FTIME

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示帧延时 (SPI Display Frame Time)
位域	类型	默认值	描述
31:0	R/W	0x0	DISP_FTIME 帧间隔时间 刷屏模式图像帧之间的间隔时间，单位为 AHB 的时钟周期。

7.3.7.36. 0x0C4 SPI_DISP_TXFC

注：

该寄存器仅在 DISP 模式下配置有效。

默认值：0x00000000			SPI 显示发送缓存控制 (SPI Display TX Buffer Control)
位域	类型	默认值	描述
31	R/W	0x0	EXT_TXF_EN 外部发送缓存区使能 • 0: 使用控制器内部 SRAM 作为发送缓存 • 1: 使用控制器外部 SRAM 作为发送缓存
30	R/W	0x0	DISP_URTENT_EN 刷屏模式 urgent 信号使能 • 0: 不使用 urgent 机制 • 1: 使能 urgent 信号，控制器根据配置的水位是 urgent 信号有效

默认值：0x00000000			SPI 显示发送缓存控制 (SPI Display TX Buffer Control)
位域	类型	默认值	描述
29:27	–	–	–
26:24	R/W	0x0	DISP_TRANS_TRIG_WL 刷屏模式行传输触发水位 当数据量大于等于该值，开始启动新的行传输 • 0：0 字节 • 1：32 字节 • 2：64 字节 • 3：128 字节 • 4：256 字节 • 5：512 字节 • 6：1024 字节 • 7：DISP_LEN – 8
23:12	R/W	0x0	DISP_URTENT_THRL 刷屏模式 urgent 信号触发低阈值 当发送缓存的数量小于等于该值，urgent 信号有效。
11:0	R/W	0x0	DISP_URTENT_EN 刷屏模式 urgent 信号触发高阈值 当发送缓存的数量大于等于该值，urgent 信号无效。

7.3.7.37. 0x200 SPI_TXD

注：

该寄存器为只写。

默认值：0x00000000			SPI 发送数据 (SPI TX Data)
位域	类型	默认值	描述
31:0	WO	0x0	TXD 发送数据值 该寄存器可以通过 AHB 总线以字节，半字或字为单位进行访问：

默认值：0x00000000			SPI 发送数据 (SPI TX Data)
位域	类型	默认值	描述
			• 以字节访问时，每次写一笔数据到 TX FIFO，FIFO 深度加 1。 • 以半字访问时，每次写两笔数据到 TX FIFO，FIFO 深度加 2。 • 以字为单位访问时，每次写四笔数据到 TX FIFO，FIFO 深度加 4。

7.3.7.38. 0x300 SPI_RXD

注：

该寄存器为只读。

默认值：0x00000000			SPI 接收数据 (SPI RX Data)
位域	类型	默认值	描述
31:0	RO	0x0	RXD 接收数据值 该寄存器可以通过 AHB 总线以字节，半字或字为单位进行访问。 • 以字节访问时，每次从 RX FIFO 读一笔数据到寄存器，FIFO 深度减 1。 • 以半字访问时，每次从 RX FIFO 读两笔数据，FIFO 深度减 2。 • 以字为单位访问时，每次从 RX FIFO 读四笔数据，FIFO 深度减 4。

7.4. Universal Asynchronous Receiver/Transmitter (UART)

UART 是一种与外围设备、数据载体设备或者数据集等进行串行通信的接口。主处理器 (CPU) 通过 APB 总线将数据写到 UART，数据转换成串行形式后传输到目标设备。UART 也可以接收串行数据和存储，以供主处理器 (CPU) 读取。

- UART 包含了控制以下功能的寄存器：
  - 字符长度
  - 波特率
  - 奇偶校验
  - 中断控制

    尽管 UART 只有一个中断输出信号，但有几种优先级的中断类型可响应，每种中断类型都可以通过控制寄存器单独使能或未使能。
  - 中断可在发送缓冲/ FIFO、接收缓冲/ FIFO、调制解调器状态以及线路状态范围内生成。
- UART 具有以下两种工作模式，可与一系列标准软件驱动程序兼容。

- 16550 模式：数据的发送与接收都由 FIFO 缓存。
- 16450 模式：禁用 FIFO。

• UART 支持 5–8 位的字长，一个可选的奇偶校验位，1/ 1.5/ 2 个停止位，可通过 CPU APB 接口编程，包含 16 位可编程波特率发生器和 8 位暂存器，以及独立的发送和接收 FIFO。提供八条调制解调器控制线和调试回环模式。

7.4.1. 特性说明

• 兼容工业标准 16550 UART

• 128 x 8bit 发送与接收 FIFO

• 传输速度可达 3 Mbps

• 支持 5–8 数据位以及 1/ 1.5/ 2 停止位

• 支持奇校验，偶校验或者无奇偶校验

• 支持 DMA 控制器接口

• 支持软件/硬件流控

• 支持 9–bit RS–485 模式

• 支持 RS–485 硬件自动控制收发方向

• 支持 LIN 总线

• 支持 byte 或 word 收发模式

7.4.2. 原理框图

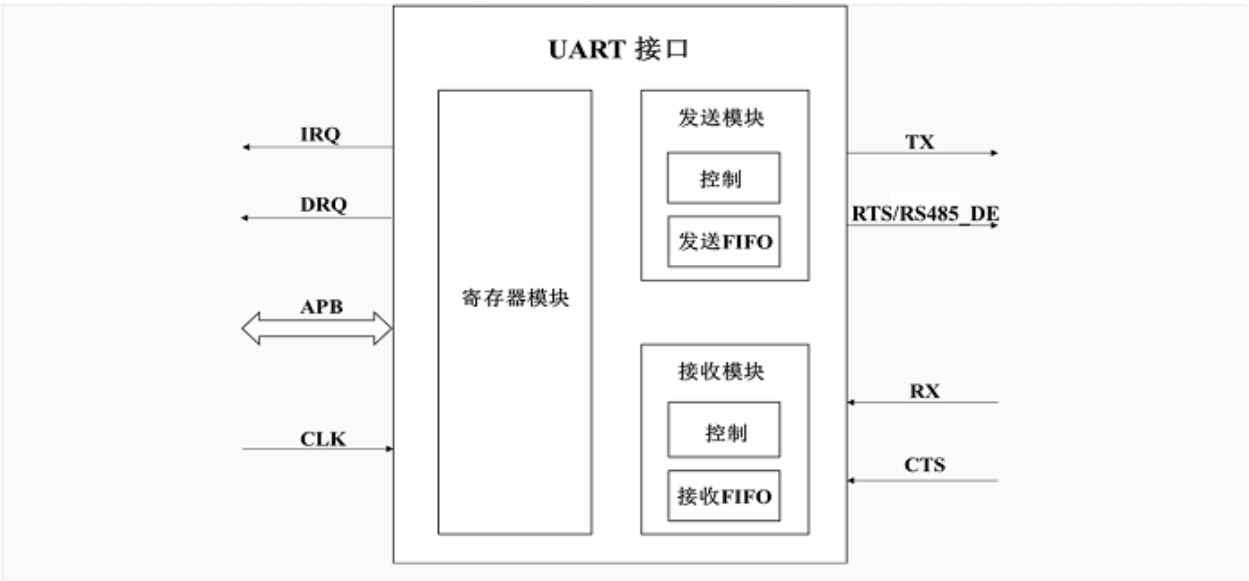


图 7–9 UART 原理框图

7.4.3. 功能描述

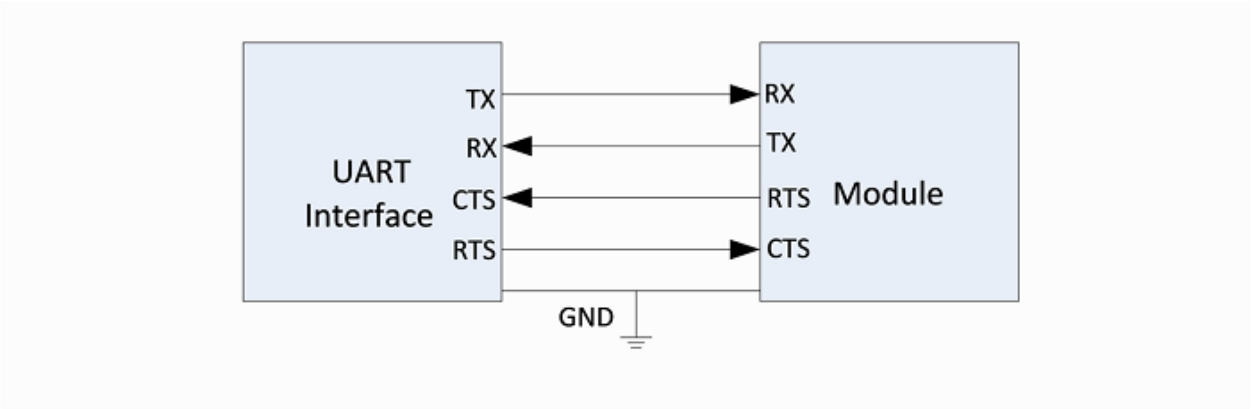
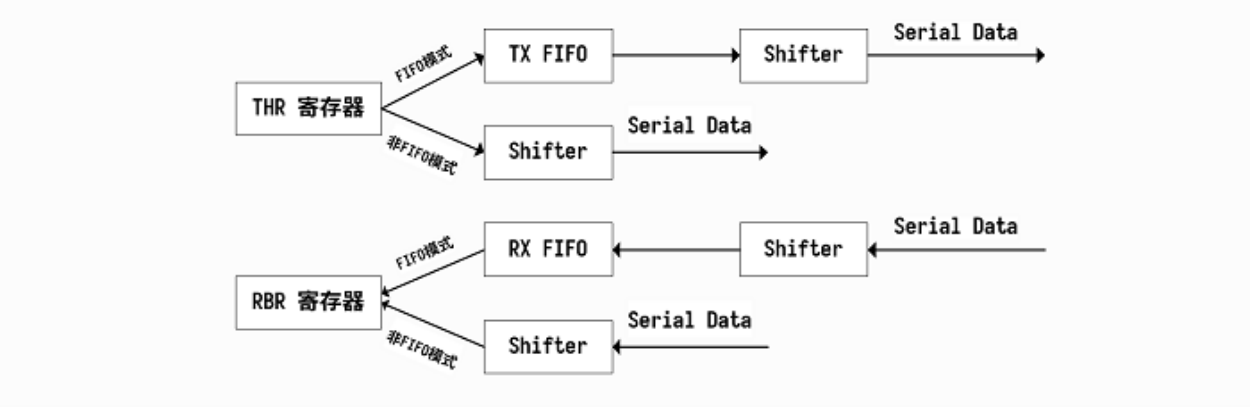


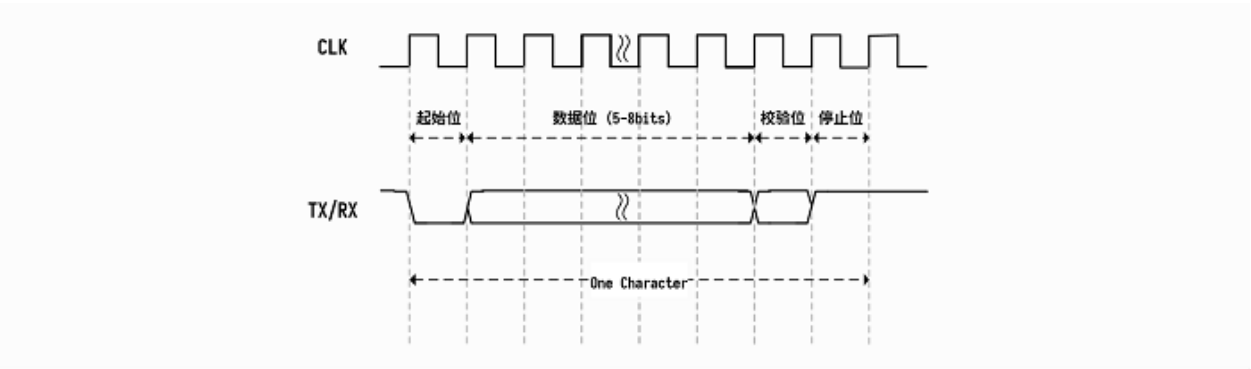
图 7–10 UART 典型应用

7.4.3.1. 功能实现

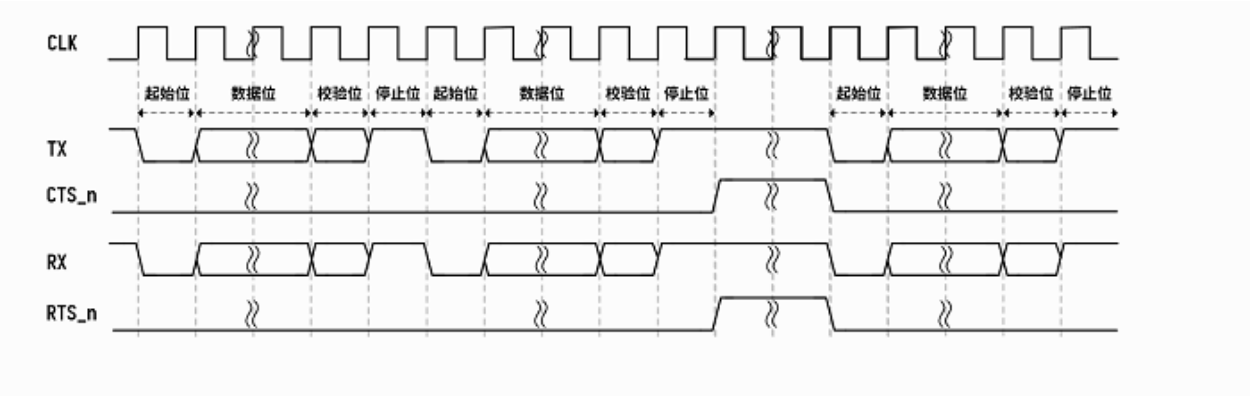
UART 数据流向：



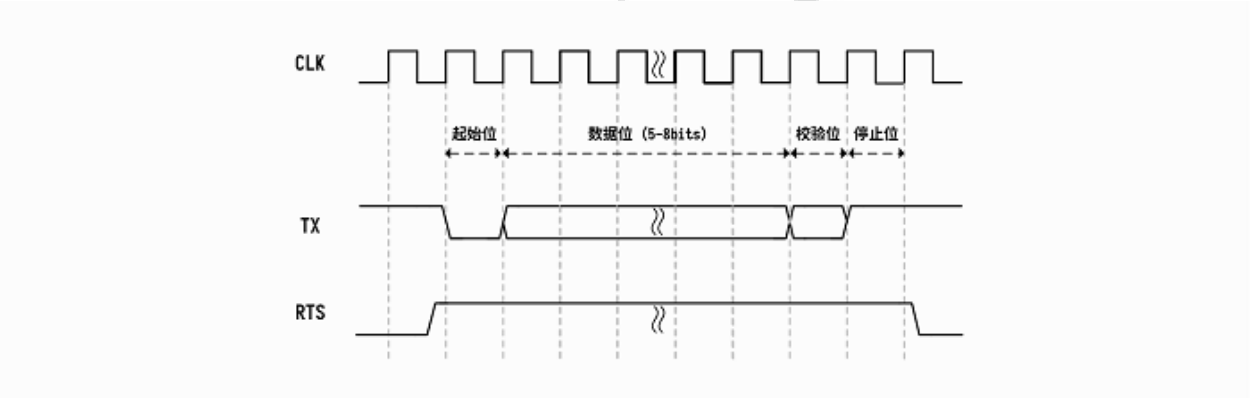
UART 串行数据格式：



UART 自动流控制时序：



UART RS-485 时序



UART 电气特性

UART 接收时序：

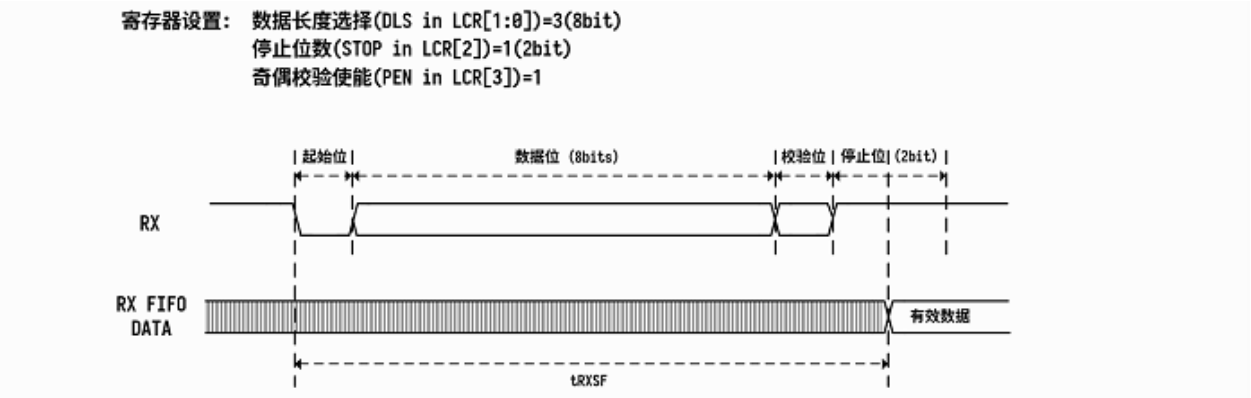


表 7-23 UART 接收时序参数

Parameter	Symbol	Min	Type	Max	Unit
RX start to RX FIFO	tRXSF	10.5 × BRP (1)	–	11 × BRP	ns

(1) BRP: Baud-Rate Period，波特率周期

UART nCTS 时序：

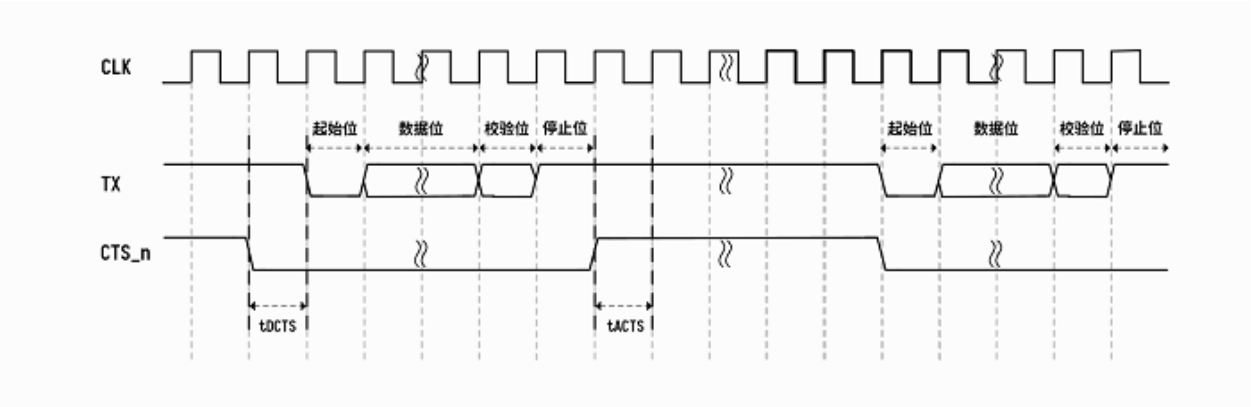


表 7–24 UART nCTS 时序参数

Parameter	Symbol	Min	Type	Max	Unit
Delay time of de–asserted nCTS to TX strat	tDCTS	–	–	BRP	ns
Step time of asserted nCTS to stop next transmission	tACTS	BRP /4	–	–	ns

UART nRTS 时序：当接收 FIFO 快满时 (FIFO 中的数据量 ≥ FIFO Depth – 2)，RTS_n 拉高。

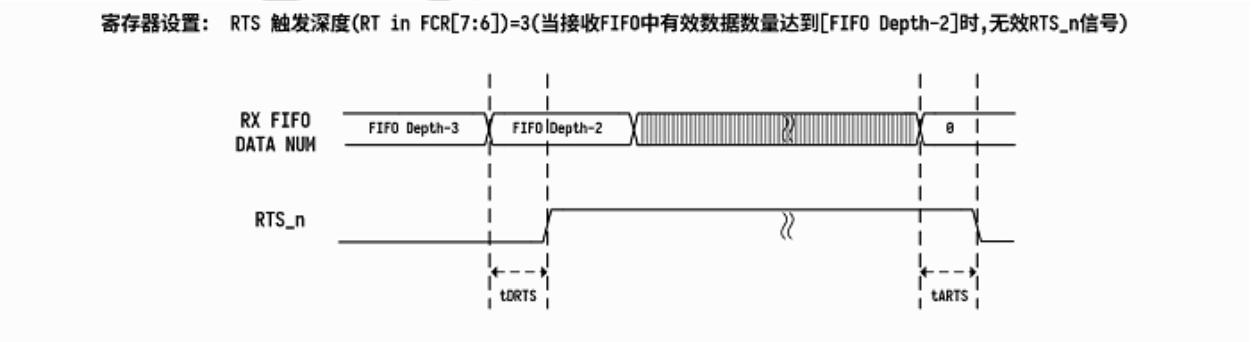


表 7–25 UART nRTS 时序参数

Parameter	Symbol	Min	Type	Max	Unit
Delay time of de–asserted nRTS	tDRTS	–	–	BRP	ns
Delay time of asserted nRTS	tARTS	–	–	BRP	ns

7.4.3.2. 操作模式

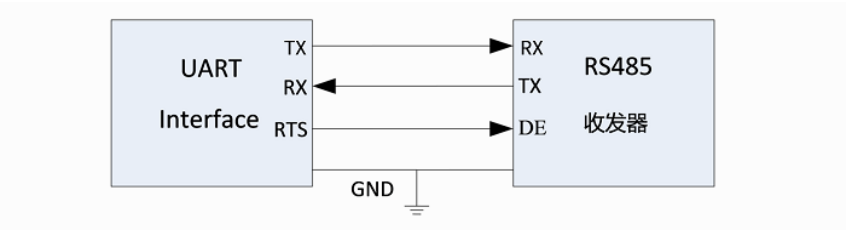
用户可以在 UART_MCR 寄存器中的 UART_FUNCTION (UART_MCR[7:6]) 位选择对应的操作模式。操作模式选择位可配置值及对应模式的描述如下：

- 0：基础 UART 模式，默认值。

  关于基础模式的详细配置，可查看 设置基础模式。

- 2：RS485 模式。

即使用 UART–TX (RS485–TX)、UART–RX (RS485–RX) 和 UART–RTS (RS485–DE) 三线制通信。RS485 模式，三线制典型应用：



- 3：RS485–2 模式。

  即使用 UART–TX (RS485–TX/RX) 和 UART–RX (RS485–DE) 两线制通信。RS485 为半双工通信，同一时段只能发送或接收，即使用两线制实现 RS485 通信，详细描述可见 RS485 半双工控制。

UART 模块同时支持 LIN 模式，可通过配置 UART_LIN_CSR 寄存器中的 LIN_MODE_EN 位选择关闭或使能 LIN 模式，详细描述见 LIN 模式。

以上操作模式均支持 byte 或 word 收发模式：

- 当通过 UART_RBR 寄存器进行数据接收时将以 byte 模式接收数据，即每次读取 1 byte 数据
- 当通过 UART_THR 寄存器进行数据发送时将以 byte 模式发送数据，即每次写入 1 byte 数据
- 当通过 UART_WRBR 寄存器进行数据接收时将以 word 模式接收数据，即每次读取 1 word 数据
- 当通过 UART_WTHR 寄存器进行数据发送时将以 word 模式发送数据，即每次写入 1 word 数据

为避免以 word 模式读取数据时出现取空、写入数据时出现溢出，可根据 FIFO 水位进行判断，也可根据UART_WINTR 寄存器 RWDR 位和 WTRR 位进行判断。

7.4.3.2.1. 设置基础模式

如在UART_MCR 寄存器中的 UART_FUNCTION (UART_MCR[7:6]) 位选择了基础模式，可以在 UART_LCR 寄存器中配置 UART 控制器的基本功能。可配置的基本参数有：

- 数据位宽 5–8 位
- 停止位个数 1/ 1.5/ 2
- 奇偶校验类型

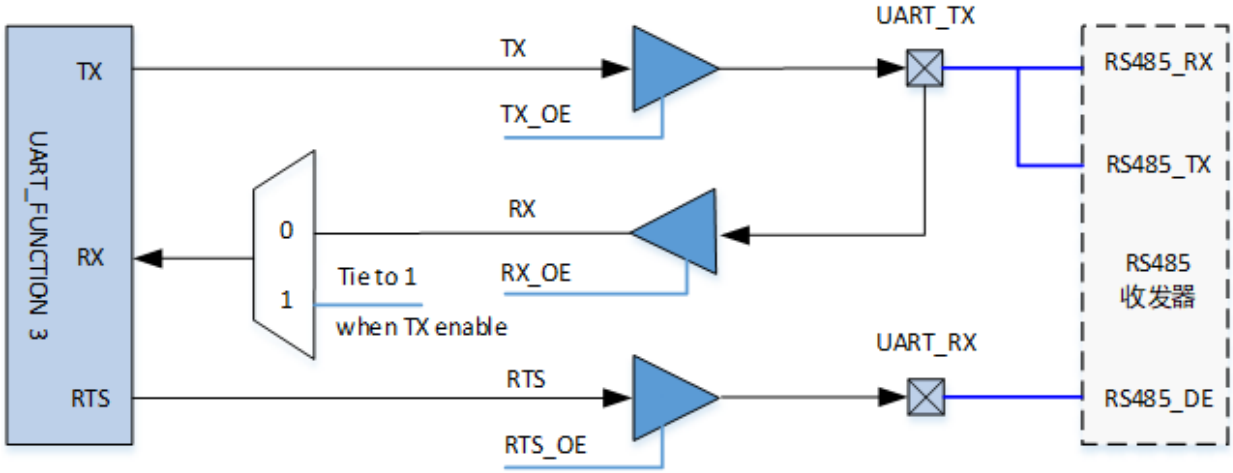
在基础模式下，UART 的一次帧传输主要包括起始信号、数据、校验位和结束信号，低位先传输 LSB。

- 起始信号 Start Bit：一个数据帧开始的标志，UART 协议规定 TX 信号出现一个低电平表示一个数据帧的开始。当 UART 没有传输数据时，保持高电平。
- 数据信号 Data Bit：数据宽度根据不同应用可进行调整，可以配置为 5/ 6/ 7/ 8 bits 数据位宽。
- 校验位 Parity Bit：校验位是 1 bit 纠错信号。校验位包括奇校验、偶校验，同时支持校验位的使能和禁止，可以通过 UART_LCR 寄存器设置。
- 结束信号 Stop Bit：数据位的停止位，支持 1/ 1.5/ 2 bit 停止位，可以通过 UART_LCR 寄存器设置。数据帧的结束信号就是把 TX 拉成高电平。

7.4.3.2.2. RS485 半双工控制

RS485 半双工控制即 RS485–2 线模式，可在 UART_MCR 寄存器中的 UART_FUNCTION (UART_MCR[7:6]) 功能选择位进行配置。

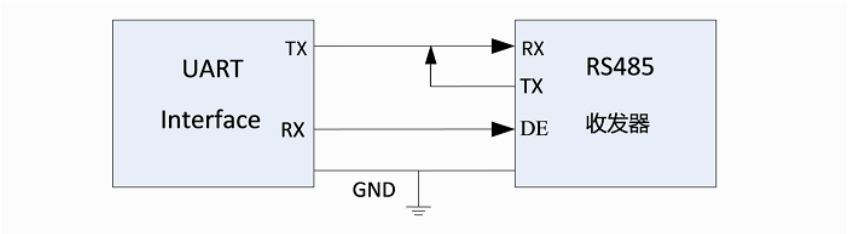
RS485–2 线模式原理框图如下：



在 RS485–2 线模式中：

- UART–TX 既可以发送数据也可以接收数据。
- UART–RX 在模块内部与 UART–RTS 复用，用于控制 DE 使能发送。
- 当进入 RS485–2 模式，RTS 复用至 UART_RX PIN 脚，TX 和 RX 共用 UART_TX PIN 脚。TX 发送使能时，RX 拉高进入空闲状态且 RX 接收通路关闭。TX 发送使能关闭后，开启 RX 接收通路。数据收发处理逻辑与常规 RS485 模式类似。

RS485 模式的两线制典型应用如下：



RS485–2 线模式相关信号的控制如下所示：

- RX 在数据发送时输出高电平：在软件控制模式下，通过配置 MCR[1]=0，DE (rts_n) 输出高电平。在硬件控制模式下，当数据发送正在进行时，硬件自动控制 RX 输出高电平。

> **注：**
>
> 若正在接收数据，必须等待当前数据接收完成。

- RX 在数据接收时输出低电平：在软件控制模式下，通过配置 MCR[1]=1，DE (rts_n) 输出低电平。在硬件控制模式下，当发送 FIFO 为空时，硬件自动控制 RX 输出低电平。

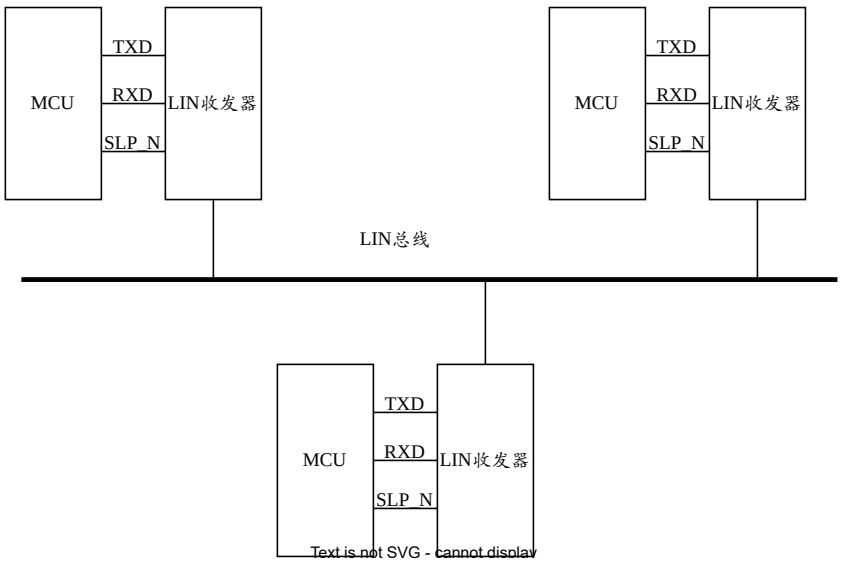
> **注：**
>
> 若正在发送数据，必须等待当前数据发送完成。

- UART_485_CTL[7]（RS485_CTL_MODE，RS485 控制模式）
  - RS485_CTL_MODE = 0：硬件控制模式。DE 信号 (RTS) 在数据发送时通过内部硬件逻辑进行自动控制，发送数据前自动使能 DE 为有效高电平，当发送移位寄存器为空且发送 FIFO 为空，则自动切换 DE 为无效低电平。
  - RS485_CTL_MODE = 1：软件控制模式。DE 信号 (RTS) 在数据发送时通过外部软件配置进行手动控制，配置 MCR[1]=0 则 DE 为有效高电平可发送数据，配置 MCR[1]=1 则 DE 为无效低电平可接收数据。
- UART_485_DE[3:0]（DE_AT，DE 有效时间 t1）
  - DE 信号上升沿到 TX 起始位之间的时间间隔，以串行时钟 (sclk) 周期为单位。
- UART_485_DE[7:4]（DE_DAT，DE 无效时间 t2）
  - TX 结束位到 DE 信号下降沿之间的时间间隔，以串行时钟 (sclk) 周期为单位。

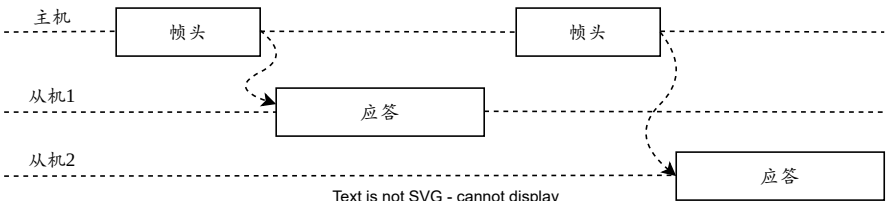
7.4.3.2.3. LIN 模式

Local Interconnect Network (LIN) 是面向汽车低端分布式应用的低成本、低速率（20 kbps）的串行通信总线。LIN 与传统 UART 的不同是 LIN 可以发送及检测 13 bits 长度的 break 信号，该信号表示 LIN 帧的开始，其它帧字节兼容串口标准。

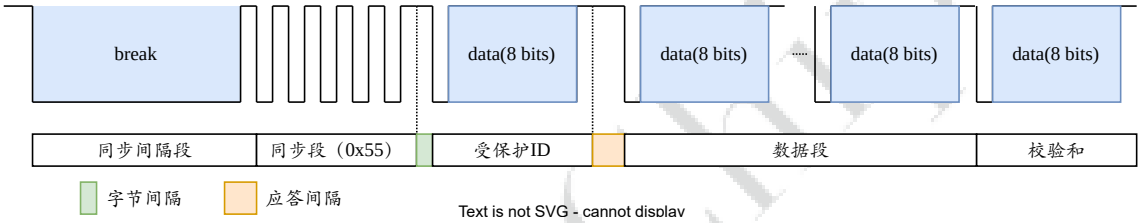
- LIN 是半双工通信，一般使用单线 12 V 电平，需要专用的收发器，LIN 的硬件连接如下图所示：



- LIN 的通信关系如下图所示，包含一个主机及多个从机（包含主机最多 16 个节点），主机主要发出帧头信息，从机根据帧头内容决定是否响应主机：



• LIN 的帧结构如下图所示：



7.4.3.3. 设置 DLAB

DLAB 控制位，即 UART_LCR[7]，是分频系数寄存器的访问控制位。

• 若 DLAB 为 0，偏移地址 0x00 为 TX 和 RX FIFO 寄存器，偏移地址 0x04 为 IER 寄存器。

• 若 DLAB 为 1，偏移地址 0x00 为 DLL 寄存器，偏移地址 0x04 为 DLH 寄存器。

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

7.4.3.4. 设置波特率

波特率计算公式：Baudrate = SCLK / (CMU_divisor * 16 * UART_divisor)，其中

• SCLK：时钟源，每个 UART 通道可单独在 CMU 配置各自时钟。

• CMU_divisor：CMU 模块的分频系数。

• UART_divisor ：UART 模块的分频系数，共有 16 位。

  寄存器 UART_DLL 为低 8 位，寄存器 UART_DLH 为高 8 位。

关于时钟源在不同频率下波特率的误差说明，详见Clock Manage Unit (CMU) 功能描述。

7.4.3.5. 设置 UART 繁忙

BUSY 标志位，即 UART_USR[0]，用于表示 UART 控制器是否繁忙。BUSY 标志位配置值如下：

• 0：空闲

• 1：繁忙

以下情况，BUSY 位将会自动置 1，表示 UART 控制器繁忙：

• TX 正在发送数据

• RX 正在接收数据

- TX FIFO 非空
- RX FIFO 非空

7.4.3.6. 设置 UART 接收器

对 UART 模块初始化配置完成后，需使能 UART 接收器，打开 RX 端口以接收数据，即 UART_RXCTL[0]，用于控制 UART 接收器是否使能。UART 接收器控制配置值如下：

- 0：RX 端口未使能
- 1：RX 端口使能

7.4.4. 编程指南

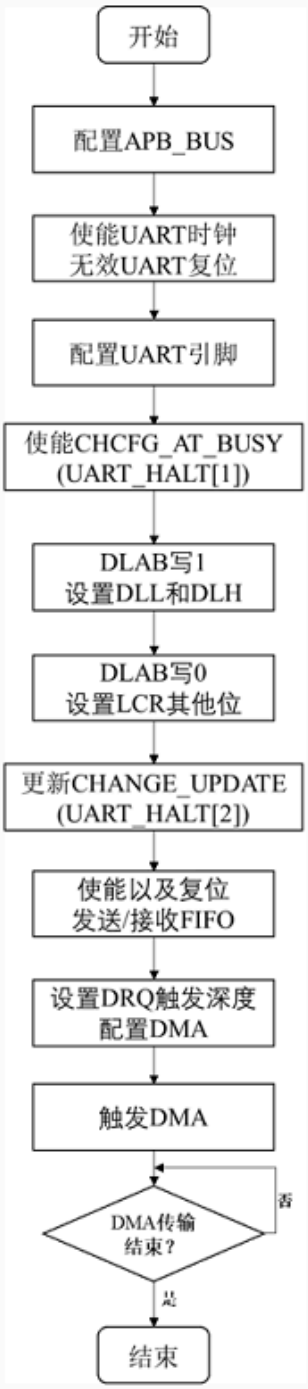


图 7–11 UART DRQ 流程

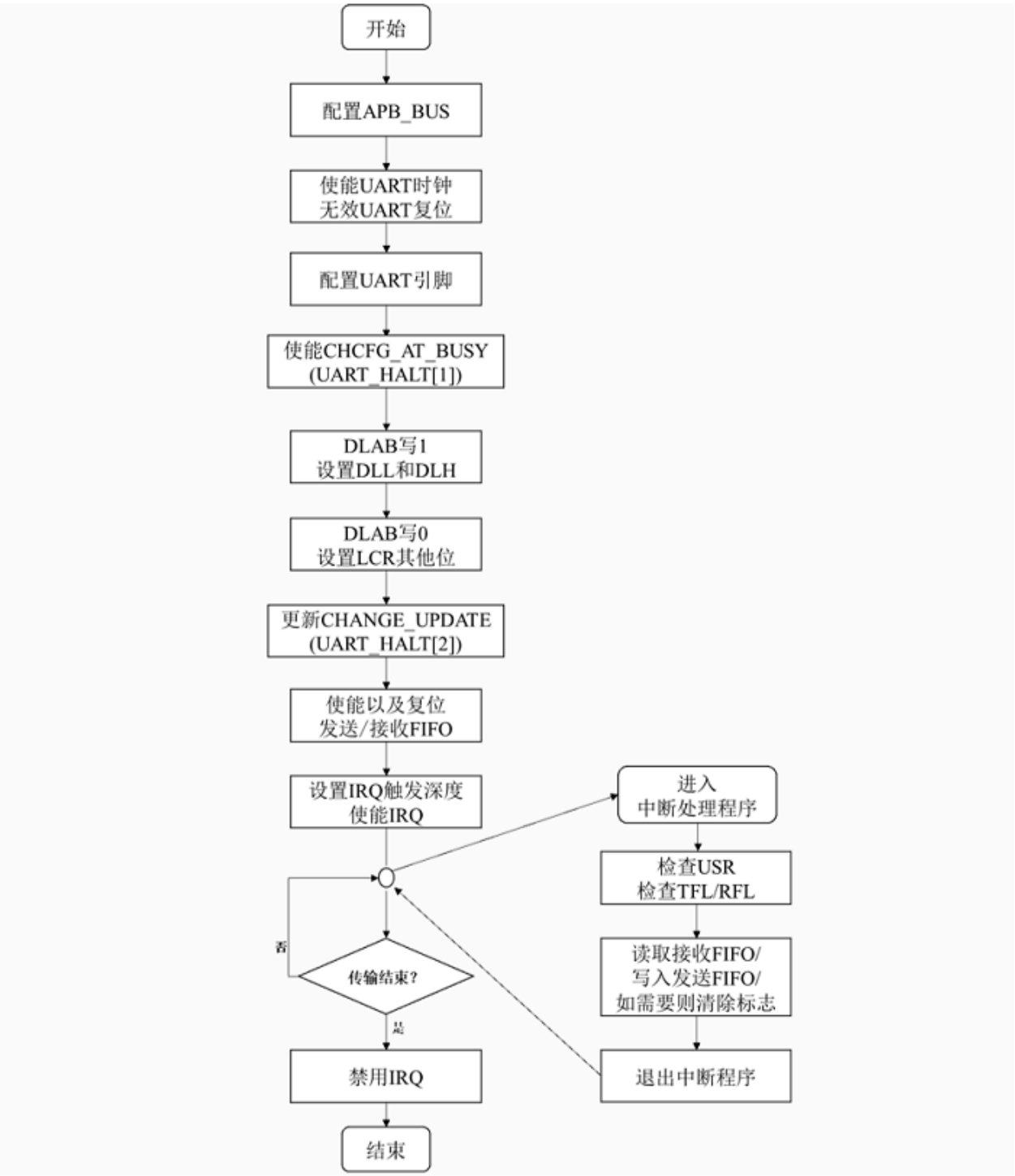


图 7–12 UART IRQ 流程

7.4.5. 寄存器列表

表 7–26 UART 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	UART_RBR	UART 数据接收寄存器	0x000 UART_RBR

表 7–26 UART 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x000	UART_THR	UART 数据发送寄存器	0x000 UART_THR
0x000	UART_DLL	UART 低位除数寄存器	0x000 UART_DLL
0x004	UART_DLH	UART 高位除数寄存器	0x004 UART_DLH
0x004	UART_IER	UART 中断使能寄存器	0x004 UART_IER
0x008	UART_IIR	UART 中断标志寄存器	0x008 UART_IIR
0x008	UART_FCR	UART FIFO 控制寄存器	0x008 UART_FCR
0x00C	UART_LCR	UART 线路控制寄存器	0x00C UART_LCR
0x010	UART_MCR	UART 调制解调器控制寄存器	0x010 UART_MCR
0x014	UART_LSR	UART 线路状态寄存器	0x014 UART_LSR
0x018	UART_MSR	UART 调制解调器状态寄存器	0x018 UART_MSR
0x01C	UART_SCH	UART 暂存器	0x01C UART_SCH
0x030	UART_WRBR	UART WORD 模式数据接收寄存器	0x030 UART_WRBR
0x030	UART_WTHR	UART WORD 模式数据发送寄存器	0x030 UART_WTHR
0x034	UART_WINTR	UART WORD 模式中断寄存器	0x034 UART_WINTR
0x07C	UART_USR	UART 状态寄存器	0x07C UART_USR
0x080	UART_TFL	UART 发送 FIFO 深度寄存器	0x080 UART_TFL
0x084	UART_RFL	UART 接收 FIFO 深度寄存器	0x084 UART_RFL
0x088	UART_HSK	UART DMA 握手配置寄存器	0x088 UART_HSK

表 7–26 UART 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x08C	UART_DMAW_CTL	UART DMA Word 模式控制寄存器	0x08C UART_DMAW_CTL
0x0A0	UART_RXCTL	UART 接收器控制寄存器	0x0A0 UART_RXCTL
0x0A4	UART_HALT	UART 停止发送寄存器	0x0A4 UART_HALT
0x0B0	UART_DBG_DLL	UART 低位除数调试寄存器	0x0B0 UART_DBG_DLL
0x0B4	UART_DBG_DLH	UART 高位除数调试寄存器	0x0B4 UART_DBG_DLH
0x0B8	UART_485_DE	UART RS485 DE 时间寄存器	0x0B8 UART_485_DE
0x0C0	UART_485_CTL	UART RS485 控制与状态寄存器	0x0C0 UART_485_CTL
0x0C4	RS485_ADDR_MATCH	UART RS485 地址匹配寄存器	0x0C4 RS485_ADDR_MATCH
0x0C8	BUS_IDLE_CHK	UART RS485 总线空闲查询寄存器	0x0C8 BUS_IDLE_CHK
0x0CC	TX_DLY	UART 发送延时寄存器	0x0CC TX_DLY
0x0D4	UART_TFAET	UART 发送 FIFO 快空触发	0x0D4 UART_TFAET
0x0D8	UART_RFAFT	UART 接收 FIFO 快空触发	0x0D8 UART_RFAET
0x0DC	UART_LIN_CSR	UART LIN 控制及状态寄存器	0x0DC UART_LIN_CSR
0x0FC	UART_VERSION	UART 版本号	0x0FC UART_VERSION

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

7.4.6. 寄存器描述

7.4.6.1. 0x000 UART_RBR

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

默认值：0x00000000			UART 数据接收寄存器 (Receive Buffer)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R	0x0	RBR Receive Buffer Register，数据接收寄存器 串行输入端口接收到的数据，仅当线路状态寄存器 (LSR) 中的数据就绪位 (DR bit) 被设为 1 时，该寄存器中的数据才有效。 使用此寄存器接收数据，UART 将以 byte 模式接收数据，即每次读取 1 byte 数据。 在 FIFO 模式中（FCR[0] 设为 1）读取该寄存器则访问接收 FIFO 的顶部（先进先出）。如果接收 FIFO 已满，并且在下一个数据到达之前没有读取该寄存器，FIFO 中已经存在的数据将被保留，但任何传入的数据都会丢失并发生溢出错误。

7.4.6.2. 0x000 UART_THR

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

默认值：0x00000000			UART 数据发送寄存器 (Transmit Holding)
位域	类型	默认值	描述
31:8	–	–	–
7:0	W	0x0	THR Transmit Holding Register，数据发送寄存器 串行输出端口发送的数据，仅当线路状态寄存器 (LSR) 中的 THRE 位 (LSR[5]) 被设为 1 时，数据才会写到该寄存器。 使用此寄存器发送数据，UART 将以 byte 模式发送数据，即每次写入 1 byte 数据。 在 FIFO 模式中（FCR[0] 设为 1）并且 THRE 为 1 时，数据可以在 FIFO 满之前写入该寄存器。当 FIFO 已满时，任何写入数据的尝试都会导致写入数据丢失。

7.4.6.3. 0x000 UART_DLL

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

默认值：0x00000000			UART 低位除数寄存器 (Divisor Latch Low)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	DLL Divisor Latch Low，低位除数寄存器 16 位 UART 波特率除数的低 8 位。仅当 DLAB 位 (LCR[7]) 为 1 并且 UART 空闲 (USR[0]) 时，该寄存器才能被访问。 输出波特率等于串行时钟频率除以波特率除数的 16 倍，如下： Baudrate = SCLK / (CMU_divisor * 16 * UART_divisor) 注： 当除数寄存器（DLL 与 DLH）设置为 0 时，波特时钟被禁用，并且没有串行通信发生。

7.4.6.4. 0x004 UART_DLH

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

默认值：0x00000000			UART 高位除数寄存器 (Divisor Latch High)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	DLH Divisor Latch High，高位除数寄存器 16 位 UART 波特率除数的高 8 位。仅当 DLAB 位 (LCR[7]）为 1 并且 UART 空闲 (USR[0]) 时，该寄存器才能被访问。 输出波特率等于串行时钟频率除以波特率除数的 16 倍，如下： Baudrate = SCLK / (CMU_divisor * 16 * UART_divisor)

默认值：0x00000000			UART 高位除数寄存器 (Divisor Latch High)
位域	类型	默认值	描述
			注： 当除数寄存器 (DLL 与 DLH) 设置为零时，波特时钟被禁用，并且没有串行通信发生。

7.4.6.5. 0x004 UART_IER

注：

UART 模块初始化时，如需设置分频系数，先要对 DLAB 写 1，才能访问寄存器 DLL 和 DLH。设置完成后，需对 DLAB 写 0，才能访问寄存器 TX、RX FIFO 以及 IER。

默认值：0x00000000			UART 中断使能寄存器 (Interrupt Enable)
位域	类型	默认值	描述
31:9	–	–	–
8	R/W	0x0	BUSY_DET_INT_EN Busy Detect Interrupt Enable，繁忙检测标志中断模式使能 • 0：未使能 • 1：使能
7	R/W	0x0	PTIME Programmable THRE Interrupt Mode Enable，THRE 中断模式使能 • 0：未使能。THRE 中断会在非空的条件清除。 • 1：使能。THRE 中断会在低于触发深度 FCR[5:4] 的条件清除。
6	–	–	–
5	R/W	0x0	Shifter_Reg_Empty_EN 发送移位寄存器为空中断使能，用于发送 FIFO 为空并且移位寄存器也为空的中断使能控制： • 0：未使能 • 1：使能
4	R/W	0x0	RS485_INT_EN RS485 Interrupt Enable，RS485 中断使能

默认值：0x00000000			UART 中断使能寄存器 (Interrupt Enable)
位域	类型	默认值	描述
			• 0：未使能 • 1：使能
3	R/W	0x0	EDSSI Enable Modem Status Interrupt，调制解调器状态中断使能 用于调制解调器状态 (MSR) 中断使能控制，为第四优先级中断。 • 0：未使能 • 1：使能
2	R/W	0x0	ELSI Enable Receive Line Status Interrupt，接收线路状态中断使能 用于接收线路状态中断使能控制，为最高级中断。 • 0：未使能 • 1：使能
1	R/W	0x0	ETBEI Enable Transmit Holding Register Empty Interrupt，数据发送寄存器为空中断使能 用于数据发送寄存器为空中断使能控制，为第三优先级中断。 • 0：未使能 • 1：使能
0	R/W	0x0	ERBFI Enable Receive Data Available Interrupt，接收数据可用中断使能 用于接收数据可用中断使能控制，以及字符超时中断控制（如果在 FIFO 模式并且 FIFO 使能），为第二优先级中断。 • 0：未使能 • 1：使能

7.4.6.6. 0x008 UART_IIR

默认值：0x00000001			UART 中断标志寄存器 (Interrupt Identity)
位域	类型	默认值	描述
31:8	–	–	–
7:6	R	0x0	FEFLAG FIFOs Enable Flag，FIFOs 使能标志

默认值：0x00000001			UART 中断标志寄存器 (Interrupt Identity)
位域	类型	默认值	描述
			表明 FIFO 是否使能。 • 0：未使能 • 3：使能
5	–	–	–
4	RC	0x0	Shifter_Reg_Empty_INT 发送移位寄存器为空中断状态 • 1：中断触发 • 0：中断未触发或被清除
3:0	R	0x1	IID Interrupt ID，中断 ID ，表示最高优先级的挂起中断，可以是以下类型之一： • 0x0：调制解调器状态 • 0x1：无中断 • 0x2：THR 为空 • 0x3：RS485 中断 • 0x4：接收数据可用 • 0x6：接收线路状态 • 0x7：繁忙检测 • 0xC：字符超时 **注：** bit3 为 FIFO 相关触发条件的中断标志位，仅在 FIFO 使能时有效。

中断 ID	优先级	中断类型	中断源	中断复位
0x1	–	–	–	–
0x6	最高级	接收线路状态	溢位/ 奇偶检验/ 帧错误或者 break 中断	读取线路状态寄存器。
0x3	第二级	RS485 中断	在 RS485 AAD 模式中，接收地址值与 ADDR_MATCH 值匹配。	地址标志写 1 复位。
0x4	第三级	接收数据可用	接收数据可用（非 FIFO 模式或者 FIFO 未使能）或达到接收 FIFO 触发水平（FIFO 模式和 FIFO 使能)。	读取接收缓冲寄存器（非 FIFO 模式或者 FIFO 未使能）或 FIFO 低于触发水平（FIFO 模式和 FIFO 使能）。

中断 ID	优先级	中断类型	中断源	中断复位
0xC	第四级	字符超时	在四个字符时间内，接收 FIFO 没有字符进出，并且在此期间 FIFO 中至少有一个字符。	读取数据接收寄存器。
0x2	第五级	数据发送寄存器为空	数据发送寄存器为空（THRE 模式未使能）或发送 FIFO 处于或者低于阈值（THRE 模式使能）。	根据情况选择对应方式： • 如果为中断源，则读取中断标志寄存器。 • 未选择或者未使能 FIFO 或者 THRE 模式时，写数据进数据发送寄存器。 • 选择并使能 FIFO 和 THRE 模式时，发送 FIFO 高于阈值。
0x0	第六级	调制解调器状态	CTS 如果处于自动流控，CTS 不会触发。	读取调制解调器状态寄存器。
0x7	第七级	繁忙检测标志	主处理器在 UART 繁忙时（USR[0]设置为 1）尝试写入线路控制寄存器。	读取 UART 状态寄存器。

7.4.6.7. 0x008 UART_FCR

默认值：0x00000000			UART FIFO 控制寄存器 (FIFO Control)
位域	类型	默认值	描述
31:8	–	–	–
7:6	W	0x0	RT RX Trigger，接收触发，用于在接收 FIFO 中选择接收数据可用中断的触发深度。 在自动流控制模式中，它用于确定何时使 rts_n (Request to Send) 信号无效,也用于确定 DRQ 信号在某些操作模式下何时有效。 • 0：FIFO 中一个字符 • 1：四分之一 FIFO • 2：二分之一 FIFO • 3：接近 FIFO 满状态 (FIFO – 2)
5:4	W	0x0	TFT TX Empty Trigger，发送为空触发，用于选择产生发送保持寄存器为空中断的空阈值电平，也用于确定 DRQ 信号在某些操作模式中何时有效。 • 0：FIFO 为空 • 1：FIFO 中两个字符 • 2：四分之一 FIFO • 3：二分之一 FIFO

默认值：0x00000000			UART FIFO 控制寄存器 (FIFO Control)
位域	类型	默认值	描述
3	W	0x0	DMAM DMA Mode，DMA 模式 • 0：模式 0 ◦ 如果 PTE (HALT[7]) 为 1 并且发送 FIFO 使能，当 TFL（发送 FIFO 深度）小于或者等于 FIFO 触发深度时，将发送 TX DMA 请求。 ◦ 如果 PTE 为 1 并且发送 FIFO 未使能，当 THRE 为 0 时，将发送 TX DMA 请求。 ◦ 如果 PTE 为 0，当发送 FIFO 为空时，将发送 TX DMA 请求。 ◦ 如果 DMA_PTE_RX 为 1 并且接收 FIFO 使能，当 RFL（接收 FIFO 深度）等于或者大于 FIFO 触发深度时将发送 RX_DRQ。 • 1：模式 1 ◦ 如果 PTE 为 1 并且发送 FIFO 使能，当 TFL 小于或者等于 FIFO 触发深度时，将发送 TX DMA 请求。 ◦ 如果 PTE 为低，TX DMA 请求将在 TX FIFO 为空时发送，并且仅当 TX FIFO 已满时停止请求。 ◦ 如果 RFL 等于或者大于 FIFO 触发深度，RX_DRQ 将会设为 1，在其它情况则为 0。
2	W1C	0x0	TXFIFOR TX FIFO Reset，发送 FIFO 复位 复位发送 FIFO 的控制并且将 FIFO 清空。也可以清除 DMA 发送请求。此位不需要清除，为自动清除。
1	W1C	0x0	RXFIFOR RX FIFO Reset，接收 FIFO 复位 复位接收 FIFO 的控制并且将 FIFO 清空。也可以清除 DMA 接收请求。此位不需要清除，为自动清除。
0	W	0x0	FIFOE FIFO Enable，FIFO 使能 使能/未使能发送以及接收 FIFO。每当该位的值改变时，FIFO 的发送以及接收控制部分都会复位。

7.4.6.8. 0x00C UART_LCR

默认值：0x00000000			UART 线路控制寄存器 (Line Control)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	DLAB Divisor Latch Access Bit，除数存取位，用于使能除数寄存器（DLL 和 DLH）的读写，以设置 UART 的波特率。 仅当 UART 空闲时（USR[0] 为 0）此位才可写。在初始化波特率设置后，必须清除此位，以便访问其它寄存器。 • 0：选择接收缓冲寄存器（RBR）、发送保持寄存器 (THR) 以及中断使能寄存器 (IER) • 1：选择低位除数寄存器 (DLL) 以及高位除数寄存器 (DLH)
6	R/W	0x0	BC Break Control Bit，中断信号控制位 产生一个中断条件传输到接收端。 若此位设 1，串行输出将强制为逻辑 0 状态。如果处于非回环模式中（取决于 MCR[4]），串行输出线强制为低电平，直到清除此位。
5:4	R/W	0x0	EPS Even Parity Select，偶校验选择 注： 仅当 PEN (LCR[3]) 设为 1 时才能使用偶校验选择。 仅当 UART 空闲时（USR[0] 为 0）此位可写。 当使能奇偶校验时（PEN 设为 1），此位用于选择奇性或者偶性校验。 • 0：奇性校验 • 1：偶性校验 • 2 或 3：反转 LCR[4] 并直接用于校验计算 对于 RS485 模式，数据包中的奇偶校验位（即第 9 位）用于指示该包是地址还是数据，而非奇偶校验： • 0 表示这是一个数据字节。 • 1 表示这是一个地址字节。
3	R/W	0x0	PEN Parity Enable，奇偶校验使能，用于选择是否使能发送接收串行字符中的奇偶校验生成与检测。 仅当 UART 空闲时（USR[0] 为 0）此位才可写。

默认值：0x00000000			UART 线路控制寄存器 (Line Control)
位域	类型	默认值	描述
			• 0：奇偶检验未使能 • 1：奇偶检验使能
2	R/W	0x0	STOP Number of Stop Bits，停止位数，用于选择外围设备发送和接收的每个字符的停止位数。 仅当 UART 空闲时（USR[0] 为 0）此位才可写。 **注：** 不管选择多少个停止位，接收器只检查第一个停止位。 • 0：1 个停止位 • 1：当 DLS (LCR[1：0]) 为 0 时，1.5 个停止位，否则 2 个停止位。当设为 1 且数据位设为 5 位时 (LCR[1：0])，传输 1.5 个停止位，否则传输 2 位停止位。
1:0	R/W	0x0	DLS Data Length Select，数据长度选择，用于选择外围设备发送和接收的每个字符的数据位数。 仅当 UART 空闲时（USR[0] 为 0）此位才可写。可选择的位数如下： • 0：5 位 • 1：6 位 • 2：7 位 • 3：8 位

7.4.6.9. 0x010 UART_MCR

默认值：0x00000000			UART 调制解调器控制寄存器 (Modem Control)
位域	类型	默认值	描述
31:8	–	–	–
7:6	R/W	0x0	UART_FUNCTION 选择 UART 或者 RS485 • 0：UART 模式 • 2：RS485 模式（常规 485，3 线模式，TX、RX 和 RTS） • 3：RS485–2 线模式（特殊 485，半双工，2 线模式，TX、RX，其中 TX 做数据输入输出，RX 做控制。）
5	R/W	0x0	AFCE

默认值：0x00000000			UART 调制解调器控制寄存器 (Modem Control)
位域	类型	默认值	描述
			Auto Flow Control Enable，自动流控制使能 当使能 FIFO 以及此位设 1 时，自动流控制功能使能。 • 0：自动流控制模式未使能 • 1：自动流控制模式使能
4	R/W	0x0	LOOP Loop Back Mode，回环模式，用于将 UART 置于调试模式以进行测试。 • 0：正常模式 • 1：回环模式，调制解调器控制输入 (cts_n) 断开，输出 (rts_n) 在内部回环输入。 在 UART 模式下运行（MCR[7:6] 设为 0），tx 线上的数据保持为高，而串行数据输出在内部回环到 rx 线路，所有中断均正常。
3:2	–	–	–
1	R/W	0x0	RTS Request To Send，发送请求，用于直接控制发送请求 (rts_n) 输出。 • 0：rts_n 信号无效状态（高电平） • 1：rts_n 信号有效状态（低电平） rts_n 输出用于通知调制解调器 UART 已准备好交换数据。 • 当自动流控制未使能（MCR[5]设为 0），则需通过将此位设为 1，使 rts_n 处于低电平。 • 在自动流模式中（MCR[5] 设为 1）并且 FIFO 使能（FCR[0] 设为 1），也需通过将此位设为 1，使 rts_n 处于低电平，但同时也受到接收 FIFO 触发深度门控（当超过阈值时 n 处于高电平无效状态）。 • 当 MCR[1] 设为 0 时 rts_n 无效。 **注：** 在回环模式中（MCR[4] 设为 1），当该位置的值在内部回环输入时，rts_n 输出保持高电平无效状态。在 RS485 或 RS485–2 模式中，RTS 外接 485 收发器方向控制脚 DE，需通过设置此位控制发送接收方向，设 1 则为接收方向，设 0 则为发送方向。
0	–	–	–

7.4.6.10. 0x014 UART_LSR

默认值：0x00000060			UART 线路状态寄存器 (Line Status)
位域	类型	默认值	描述
31:8	–	–	–
7	R	0x0	FIFOERR RX Data Error In FIFO，FIFO 接收数据错误 • 如果 FIFO 未使能，此位总为 0。 • 如果 FIFO 使能且接收 FIFO 中至少有 PE(LSR[2])、FE(LSR[3]) 和 BI(LSR[4]) 中一个，此位被置 1。 • 如果 FIFO 没有后续错误，则通过读取 LSR 寄存器清除此位。
6	R	0x1	TEMT Transmitter Empty，发送器为空 • 如果 FIFO 未使能，每当数据发送寄存器以及发送移位寄存器为空时，该位设 1。 • 如果 FIFO 使能，每当发送 FIFO 以及发送移位寄存器为空时，该位设 1。 • 以上任意情况下，当一个字符被写入发送数据信道时，该位被清除。
5	R	0x1	THRE TX Holding Register Empty，数据发送寄存器为空 • 如果 FIFO 未使能，每当数据发送寄存器为空并且准备接收新数据时，该位被设为 1。CPU 写入数据发送寄存器时，该位被清除。 • 如果 FIFO 使能，每当发送 FIFO 为空时，该位被设为 1。当至少有一个字符写入发送 FIFO 时，该位被清除。
4	RC	0x0	BI Break Interrupt，Break 中断，用于指示在串行输入数据上检测到中断序列。 break 条件会导致 UART 接收到仅且一个由 0 组成的字符。 • 在 FIFO 模式中，与 break 条件相关联的字符通过 FIFO 传输，当字符位于 FIFO 顶部时此位置 1，读取清除此位。 • 在非 FIFO 模式中，此位立即置 1 并持续到读取此位清除。
3	RC	0x0	FE Framing Error，帧错误，用于指示接收器中出现帧错误。 当接收器在接收的数据中没有检测到有效的停止位时，就会发生帧错误。在 FIFO 模式中，由于帧错误与接收到的字符相关，因此当具有帧错误的字符位于 FIFO 顶部时，才会显示该错误。帧错误出现时，UART 会尝试重新同步。接收器重新同步是通过假设错误是由下一个字符的起始位引起的，然后继续接收另一个数据位，以及奇偶检验位和停止位。 需要注意的是，如果发生 BI，则此位也置 1。

默认值：0x00000060			UART 线路状态寄存器 (Line Status)
位域	类型	默认值	描述
			• 0：无帧错误 • 1：帧错误，读取 LSR 清除此位
2	RC	0x0	PE Parity Error，奇偶校验错误 此位用于指示在接收中发生奇偶校验错误。 • 0：无奇偶校验错误 • 1：奇偶校验错误，读取 LSR 清除此位 注： 仅在使能奇偶校验 (PEN) 位 (LCR[3]) 功能时此位有效。
1	RC	0x0	OE Overrun Error，溢出错误 • 0：无溢出错误 • 1：溢出错误，读取 LSR 清除此位 如果在读取先前数据之前接收到新的数据字符，则出现此错误。 • 在非 FIFO 模式，当从 RBR 读取先前字符之前接收一个新字符，则此位被置 1，发生这种情况时，RBR 中的数据则会被覆盖。 • 在 FIFO 模式，当 FIFO 为满时接收到一个新的字符，则会发生此错误，此时 FIFO 中的数据被保留，而接收移位寄存器中的数据丢失。
0	R	0x0	DR Data Ready，数据就绪 用于指示在 RBR 或者接收 FIFO 中至少包含一个字符。 • 0：无数据就绪 • 1：数据就绪 当非 FIFO 模式中读取 RBR 或者 FIFO 模式中 FIFO 为空时清除此位。

7.4.6.11. 0x018 UART_MSR

默认值：0x00000010			UART 调制解调器状态寄存器 (Modem Status)
位域	类型	默认值	描述
31:5	–	–	–

默认值：0x00000010			UART 调制解调器状态寄存器 (Modem Status)
位域	类型	默认值	描述
4	R	0x1	CTS Line State Of Clear To Send，可发送状态 用于指示调制解调器控制线 cts_n 信号的当前状态。当 cts_n 有效时表示调制解调器已准备好 UART 交换数据。 • 0：cts_n 无效（高电平状态） • 1：cts_n 有效（低电平状态） **注：** 在回环模式中（MCR[4] 设为 1），CTS 内部连接 RTS (MCR[1])。
3:1	–	–	–
0	RC	0x0	DCTS Delta Clear To Send，表示自上次读取 MSR 后 cts_n 信号发生了改变。 • 0：上一次读取 MSR 后 cts_n 无变化 • 1：上一次读取 MSR 后 cts_n 有变化 读取 MSR 清除此位。在回环模式中 (MSR[4]=1)，DCTS 反映 MCR[1] (RTS) 的变化。 **注：** 如果未设置此位，并且 cts_n 信号有效（低电平）以及发生复位（软件复位或其它情况），如 cts_n 信号仍然保持有效，在移除复位时此位置 1。

7.4.6.12. 0x01C UART_SCH

默认值：0x00000000			UART 暂存器 (Scratch)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	SCRATCH_REG 暂存器，为用户提供临时存储空间，UART 中未定义用途。

7.4.6.13. 0x030 UART_WRBR

默认值：0x00000000			UART Word 模式数据接收寄存器 (Word Receive Buffer)
位域	类型	默认值	描述
31:0	R	0x0	WRBR Word Receive Buffer Register，Word 模式数据接收寄存器 串行输入端口接收到的数据，仅当 UART_WINTR 寄存器中的 Word 模式接收数据就绪位 (RWDR) 被设为 1 时，该寄存器中的数据才有效。 使用此寄存器接收数据时，UART 将以 word 模式接收数据，即每次读取 1 word 数据。 在 FIFO 模式中（FCR[0] 设为 1），读取该寄存器则依次访问接收 FIFO 顶部的 4 bytes 数据（先进先出）。如果接收 FIFO 已满并且在下一个数据到达之前没有读取该寄存器，FIFO 中已经存在的数据将被保留，但任何传入的数据都会丢失并发生溢出错误。

7.4.6.14. 0x030 UART_WTHR

默认值：0x00000000			UART Word 模式数据发送寄存器 (Word Transmit Holding)
位域	类型	默认值	描述
31:0	W	0x0	WTHR Word Transmit Holding Register，Word 模式数据发送寄存器 串行输出端口发送的数据，仅当 UART_WINTR 寄存器中的 Word 模式发送寄存器可用位 (WTRR) 被设为 1 时，数据才能写到该寄存器。 使用此寄存器发送数据时，UART 将以 word 模式发送数据，即每次写入 1 word 数据 在 FIFO 模式中 （FCR[0] 设为 1）并且 (WTRR) 为 1 时，数据可以在 FIFO 满之前写入该寄存器。当 FIFO 已满时，任何写入数据的尝试都会导致写入数据丢失。

7.4.6.15. 0x034 UART_WINTR

默认值：0x00000010			UART Word 模式中断状态寄存器 (Word Interrupt)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W1C	0x0	WRE Word Read Data Error，Word 模式读错误中断标志 当前要读取的一个 word 内，若某一字节存在（BI/FE/PE）错误，此时必须要对 RX FIFO 逐字节读取以判别具体错误，若存在 word 读取，只会读到无效数据，且会报错此位置 1 。

默认值：0x00000010			UART Word 模式中断状态寄存器 (Word Interrupt)
位域	类型	默认值	描述
			• 0：无 word 读数据错误 • 1：word 读数据错误
6	R/W	0x0	WRE_EN Word Read Data Error Enable，Word 模式读错误中断使能 • 0：未使能 • 1：使能
5	R	0x0	RWDR Receive Word Data Ready，Word 模式接收数据就绪 用于指示在 WRBR 或者接收 FIFO 中至少包含一个 word • 0：无数据就绪 • 1：数据就绪 当非 FIFO 模式中读取 WRBR 或者 FIFO 模式中 FIFO 中数据小于 4 bytes 时清除此位
4	R	0x1	WTRR Word Transmitter Register Ready，Word 模式发送寄存器可用 用于指示 WTHR 为空或者发送 FIFO 可以写入一个 word 以上的数据 • 0：发送寄存器不可用 • 1：发送寄存器可用 当非 FIFO 模式中 数据写入 WTHR 或者 FIFO 模式中 FIFO 不足以写入一个 word 以上的数据时清除此位
3	R/W1C	0x0	RFU RX FIFO Underrun，接收 FIFO 取空中断标志 • 0：接收 FIFO 无取空 • 1：接收 FIFO 取空
2	R/W1C	0x0	TFO TX FIFO Overflow，发送 FIFO 溢出中断标志 • 0：发送 FIFO 无溢出 • 1：发送 FIFO 溢出
1	R/W	0x0	RFU_EN RX FIFO Underrun Enable，接收 FIFO 取空中断使能 • 0：未使能 • 1：使能

默认值：0x00000010			UART Word 模式中断状态寄存器 (Word Interrupt)
位域	类型	默认值	描述
0	R/W	0x0	TFO_EN TX FIFO Overflow Enable，发送 FIFO 溢出中断使能 • 0：未使能 • 1：使能

7.4.6.16. 0x07C UART_USR

默认值：0x00000006			UART 状态寄存器 (Status)
位域	类型	默认值	描述
31:5	–	–	–
4	R	0x0	RFF Receive FIFO Full，接收 FIFO 为满 用于指示接收 FIFO 已完全满。 • 0：接收 FIFO 未满 • 1：接收 FIFO 已满当接收 FIFO 不再满时自动清零
3	R	0x0	RFNE Receive FIFO Not Empty，接收 FIFO 非空 用于指示接收 FIFO 包含一个或多个数据。 • 0：接收 FIFO 为空 • 1：接收 FIFO 非空当接收 FIFO 为空时自动清零
2	R	0x1	TFE Transmit FIFO Empty，发送 FIFO 为空 用于指示发送 FIFO 完全空。 • 0：发送 FIFO 非空。 • 1：发送 FIFO 为空。当发送 FIFO 不再空时自动清零。
1	R	0x1	TFNF Transmit FIFO Not Full，发送 FIFO 未满 用于指示发送 FIFO 未满。 • 0：发送 FIFO 已满 • 1：发送 FIFO 未满。当发送 FIFO 已满时自动清零。

默认值：0x00000006			UART 状态寄存器 (Status)
位域	类型	默认值	描述
0	R	0x0	BUSY UART Busy Bit，UART 繁忙位 • 0：空闲 • 1：繁忙

7.4.6.17. 0x080 UART_TFL

默认值：0x00000000			UART 发送 FIFO 深度寄存器 (Transmit FIFO Level)
位域	类型	默认值	描述
31:9	–	–	–
8:0	R	0x0	TFL Transmit FIFO Level，发送 FIFO 深度 用于指示发送 FIFO 中的数据数量

7.4.6.18. 0x084 UART_RFL

默认值：0x00000000			UART 接收 FIFO 深度寄存器 (Receive FIFO Level)
位域	类型	默认值	描述
31:9	–	–	–
8:0	R	0x0	RFL Receive FIFO Level，接收 FIFO 深度 用于指示接收 FIFO 中的数据数量

7.4.6.19. 0x088 UART_HSK

默认值：0x000000A5			UART DMA 握手配置寄存器 (DMA Handshake Configuration)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0xA5	HSK Handshake Configuration，握手配置

默认值：0x000000A5			UART DMA 握手配置寄存器 (DMA Handshake Configuration)
位域	类型	默认值	描述
			• 0xA5：DMA wait cycle 模式 • 0xE5：DMA handshake 模式

7.4.6.20. 0x08C UART_DMAW_CTL

默认值：0x00000000			UART DMA Word 模式控制寄存器 (DMA Word Control)
位域	类型	默认值	描述
31:2	–	–	–
1	R/W	0x0	DMA_WORD_REN DMA Word Read Enable，DMA Word 模式读使能 • 0：未使能 • 1：使能
0	R/W	0x0	DMA_WORD_WEN DMA Word Write Enable，DMA Word 模式写使能 • 0：未使能 • 1：使能

7.4.6.21. 0x0A0 UART_RXCTL

默认值：0x00000000			UART 接收器控制寄存器 (Receiver Control)
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	DLAB_AC_EN DLAB Auto Clean Enable，除数存取位自动清零使能 DLAB_AC_EN 与 CHCFG_AT_BUSY（繁忙状态更改配置）配合使用，当 DLAB_AC_EN 置 1，且 CHCFG_AT_BUSY 置 1，则更改配置完成时会将 DLAB 自动清零。 • 0：未使能 • 1：使能
4:1	–	–	–
0	R/W	0x0	RX_EN

默认值：0x00000000			UART 接收器控制寄存器 (Receiver Control)
位域	类型	默认值	描述
			UART 接收器使能 • 0：RX 端口未使能 • 1：RX 端口使能

7.4.6.22. 0x0A4 UART_HALT

默认值：0x00000000			UART 停止发送寄存器 (Halt TX)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	DMA_PTE_RX The Sending Of RX_DRQ，发送 RX_DRQ • 在 DMA1 模式中，当 RFL 大于或者等于触发深度/接收超时，发送 DRQ。 • 在 DMA0 模式中，如果 DMA_PTE_RX=1 并且 FIFO 使能，当 RFL 大于触发深度时，发送 DRQ。 • 在其它情况下，一旦接收数据有效，发送 DRQ。
6	R/W	0x0	PTE The Sending Of TX_REQ，发送 TX_REQ • 在 DMA1 模式（FIFO 使能）中： ◦ 如果 PTE 设为 1，当 TFL 小于触发深度时，发送 DMA 请求。 ◦ 如果 PTE 设为 0，当 FIFO 为空，发送 DMA 请求。当 FIFO 为满时，DMA 请求将停止。 • 在 DMA0 模式中： ◦ 如果 PTE 设为 1 并且 FIFO 使能，当 TFL 小于触发深度时，发送 DMA 请求。 ◦ 如果 PTE 设为 1 并且 FIFO 未使能，当 THRE 为空时，发送 DMA 请求。 ◦ 如果 PTE 设为 0，当 FIFO 为空时发送 DMA 请求。
5:1	–	–	–
0	R/W	0x0	HALT_TX 停止发送 此位用于停止传输以进行测试，以便在使能 FIFO 时，主控可以填充发送 FIFO。 • 0：停止发送未使能 • 1：停止发送使能

默认值：0x00000000			UART 停止发送寄存器 (Halt TX)
位域	类型	默认值	描述
			注： 如果 FIFO 未使能，设置此位对操作没有影响。

7.4.6.23. 0x0B0 UART_DBG_DLL

默认值：0x00000000			UART 低位除数调试寄存器 (DBG DLL)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	DEBUG DLL

7.4.6.24. 0x0B4 UART_DBG_DLH

默认值：0x00000000			UART 高位除数调试寄存器 (DBG DLH)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	DEBUG DLH

7.4.6.25. 0x0B8 UART_485_DE

默认值：0x000000ff			UART RS485 DE 时间寄存器 (RS485 DE Time)
位域	类型	默认值	描述
31:8	–	–	–
7:4	R/W	0xf	DE_DAT Driver Enable De-assertion Time，DE 无效时间 T2 串行数据结束位到 DE 信号下降沿之间的时间间隔，以串行时钟周期为单位。
3:0	R/W	0xf	DE_AT Driver Enable Assertion Time，DE 有效时间 T1 DE 信号上升沿到串行数据起始位之间的时间间隔，以串行时钟周期为单位

7.4.6.26. 0x0C0 UART_485_CTL

默认值：0x00000080			UART RS485 控制与状态寄存器 (RS485 Control and Status)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x1	RS485_CTL_MODE RS485 Control Mode，RS485 控制模式 • 0：硬件控制模式。DE 信号 (RTS) 在数据发送时通过内部硬件逻辑进行自动控制，发送数据前自动使能 DE 为有效高电平，发送移位寄存器为空且发送 FIFO 为空，则自动切换 DE 为无效低电平。 • 1：软件控制模式。DE 信号 (RTS) 在数据发送时通过外部软件配置进行手动控制，配置 MCR[1]=0 则 DE 为有效高电平可发送数据，MCR[1]=1 则 DE 为无效低电平可接收数据。
6	R/W1C	0x0	AAD_ADDR_F AAD 地址匹配标志 在 AAD 模式中，当 UART 接收到地址字节且与 RS485_ADDR_MATCH 相同时，该位将置为 1。如果 RS485 中断使能，则将产生 RS485 中断。 写 1 清除该位并且复位 RS485 中断。
5	R/W1C	0x0	RS485_ADDR_DET_F RS485 地址检测标志 此位为地址字节检测标志。当 UART 接收到地址字节时，此位将设为 1。如果 RS485 中断使能，则将产生 RS485 中断。 写 1 清除该位并且复位 RS485 中断。
4	–	–	–
3	R/W	0x0	RX_BF_ADDR 地址之前接收 在 NMM 模式中，如果此位设为 1，UART 将在接收到一个地址字节之前将所有字节接收到 FIFO 中。如果设为 0，则不进行此操作。 • 1：接收 • 0：不接收
2	R/W	0x0	RX_AF_ADDR 地址之后接收 • 1：接收 • 0：不接收 在 NMM 模式中：

默认值：0x00000080			UART RS485 控制与状态寄存器 (RS485 Control and Status)
位域	类型	默认值	描述
			• 如果此位设为 1，UART 将在接收到一个地址字节之后将所有字节接收到 FIFO 中。 • 如果设为 0，则不进行此操作。
1:0	R/W	0x0	RS485_SLAVE_MODE_SEL RS485 从机模式选择 RS485 从机模式： • 0：标准多点操作 (NMM) • 1：自动地址检测操作 (AAD) • 2：保留 • 3：保留

7.4.6.27. 0x0C4 RS485_ADDR_MATCH

默认值：0x00000000			UART RS485 地址匹配寄存器 (RS485 Address Match)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	ADDR_MATCH 地址匹配 在 AAD 模式下使用的匹配地址。 **注：** 仅在 AAD 模式下可用。

7.4.6.28. 0x0C8 BUS_IDLE_CHK

默认值：0x00000000			UART RS485 总线空闲查询寄存器 (RS485 Bus Idle Check)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	BUS_IDLE_CHK_EN 总线空闲查询使能 • 1：使能总线空闲查询功能。 • 0：未使能总线空闲查询功能

默认值：0x00000000			UART RS485 总线空闲查询寄存器 (RS485 Bus Idle Check)
位域	类型	默认值	描述
6	R	0x0	BUS_STATUS 总线状态标志 • 1：繁忙 • 0：空闲
5:0	R	0x0	ADJ_TIME 总线空闲时间 单位为 8 × 16 × T_{clk}

7.4.6.29. 0x0CC TX_DLY

默认值：0x00000000			UART 发送延时寄存器 (TX Delay)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	DLY 发送延时 最后一个停止位与下一个开始位之间的延迟时间。单位为 16 × T_{clk}。它用于控制数据发送中两个字节之间的间隔。

7.4.6.30. 0x0D0 UART_DBR

默认值：0x00000000			UART 调试寄存器 (Debug Register)
位域	类型	默认值	描述
31:6	–	–	–
5	R/W	0x0	TX_OUT_ONLY_MASK 屏蔽 TX 端口单向输出功能 RS485–2 线模式，该 bit 无效。非 RS485–2 线模式，该 bit 有效。建议软件配置成 1。 • 1：屏蔽 • 0：使能
4	W	0x0	NOR_ACT_485_2WIRE

默认值：0x00000000			UART 调试寄存器 (Debug Register)
位域	类型	默认值	描述
			• 0：普通模式 • 1：UART 2wire 模式（特殊 UART，2 线模式，TX/RX、RTS）
3	W	0x0	Bypass 预取调试
2:0	–	–	–

7.4.6.31. 0x0D4 UART_TFAET

默认值：0x00000000			UART 发送 FIFO 快空触发 (UART TX FIFO Almost Empty Trigger)
位域	类型	默认值	描述
31:8	–	–	–
7:0	WR	0x00	发送 FIFO 快空水位 • others：触发水位 • 0：使用 UART_FCR 的配置

7.4.6.32. 0x0D8 UART_RFAET

默认值：0x00000000			UART 接收 FIFO 快空触发 (UART RX FIFO Almost Empty Trigger)
位域	类型	默认值	描述
31:8	–	–	–
7:0	WR	0x00	接收 FIFO 快空水位 • others：触发水位 • 0：使用 UART_FCR 的配置

7.4.6.33. 0x0DC UART_LIN_CSR

默认值：0x00000000			UART LIN 控制及状态寄存器 (LIN Control and Status Register)
位域	类型	默认值	描述
31:6	–	–	–

默认值：0x00000000			UART LIN 控制及状态寄存器 (LIN Control and Status Register)
位域	类型	默认值	描述
5	WR	0x00	LIN_BREAK_DETECTION_FLAG • 1：检测到 break 信号，写 1 清零 • 0：未检测到 break 信号
4	WR	0x00	LIN_BREAK_DETECTION_INTERRUPT_EN • 1：break 中断使能 • 0：break 中断关闭
3:2	WR	0x00	LIN_BREAK_DETECTION_LENGTH • 11：13 bit • 10：12 bit • 01：11 bit • 00：10 bit
1	WR	0x00	LIN_SEND_BREAK_REQ • 1：请求发送 break 信号，完成自动清 0 • 0：break 发送完成
0	WR	0x00	LIN_MODE_EN • 1：使能 LIN 模式 • 0：关闭 LIN 模式

7.4.6.34. 0x0E0 UART_X8SR

默认值：0x00000000			UART 8 倍过采样使能寄存器 (x8 Sample Register)
位域	类型	默认值	描述
31:2	–	–	–
0	R/W	0x0	X8_SAMPLE_EN 8 倍过采样使能 • 1：使能 • 0：禁用

7.4.6.35. 0x0FC UART_VERSION

默认值：0x00000104			UART 版本号寄存器 (Version)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R	0x00000104	UART 版本号寄存器，V1.4

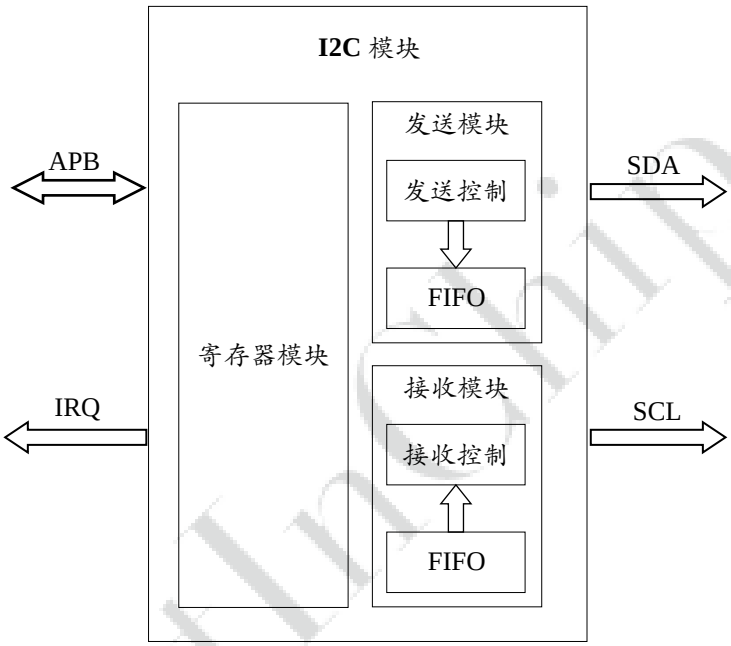
7.5. Inter Integrated Circuit (I2C)

I2C 模块是一种用于连接 CPU 主机与 I^2C 总线之间的接口，包括一根串行数据线 (SDA) 和一根串行时钟线 (SCL)，分别用于传输数据和时钟信号，是实现 I2C 通信的基础。在 I^2C 总线上，每个设备都有一个唯一的地址。当主设备 (master) 希望与从设备 (slave) 通信时，会通过发送从设备 (slave) 的地址来建立连接。这种寻址方式允许在同一总线上连接多个设备，并进行精确的数据传输。主设备负责发起数据传输并生成时钟信号，而被主设备寻址的设备则作为从设备进行响应。这种主从关系确保了通信的有序性和有效性。此外，I2C 支持多种传输模式，以满足不同的应用需求。

7.5.1. 特性说明

- 支持主和从模式
- 支持最高 400Kb/s 传输
- 支持 7 bits 和 10 bits 寻址
- 支持时钟同步
- 支持中断和轮询模式
- 可编程 SDA 保持时间
- 可编程防止 I^2C 通信异常总线挂死
- 支持 DMA 传输

7.5.2. 原理框图



7.5.3. 功能描述

7.5.3.1. 协议约定

I^2C 协议是一种应用广泛的串行通信协议，主要用于低速外设与处理器之间的数据传输。以下是对 I^2C 协议中一些关键特性和机制的详细解析：

表 7–27 协议约定

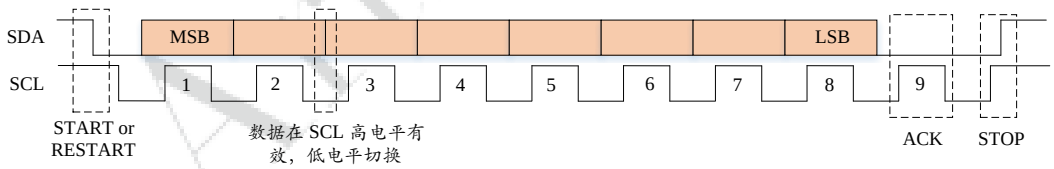
类型	描述
数据帧	大小固定为 8 bits（一字节）
传输方向	高位先发
总线释放状态	SCL 和 SDA 均为高电平，其中 SCL 一直由主设备控制。
启动信号（START）	SCL 高，SDA 由高转低
停止信号（STOP）	SCL 高，SDA 由低转高
应答信号	• ACK：主设备控制 SCL，发送完 8 bits 后释放 SDA，在第九个 SCL 波形时，从设备拉低 SDA。 • NACK：主设备控制 SCL，发送完 8 bits 后释放 SDA，在第九个 SCL 波形时，从设备不拉低 SDA。

表 7–27 协议约定 (续)

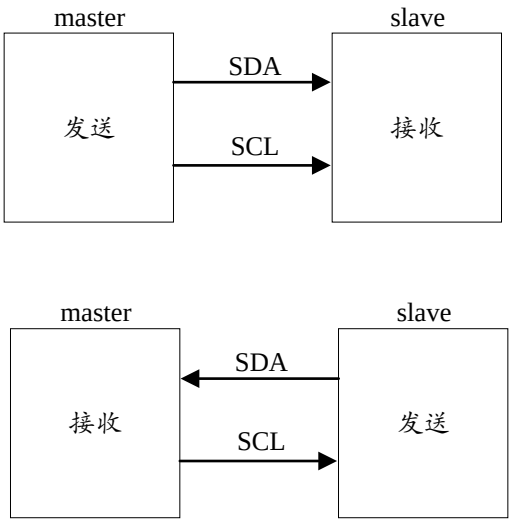
类型	描述
从设备地址	• 含 CPU 的智能器件，地址由软件初始化时定义，但不能与其它的器件有冲突。 • 不含 CPU 的非智能器件： ◦ 高 4 bits 由厂家在器件内部固化，不可改变。 ◦ 低 3 bits 由引脚连接电平决定，例如 eeprom，三个引脚常用名字 A1、A2、A3。
读写位	1 bit，以 7 位地址模式为例，与 从设备地址的 7 bits，组成读写数据帧的数据内容（从设备 7 bits + 1 bit），0 表示写，1 表示读。
空闲检测	主设备检测到 SDA、 SCL 均为高电平。

7.5.3.2. 主和从 模式

I^2C 协议：



主设备和从设备关系图：



主设备和从设备详细行为描述如下：

- 主设备发送操作：
  - 发起寻址：主设备通过 I2C_DATA_CMD[8] 写 0 进行发送请求。
  - 数据写入：主设备将数据写进 I2C_DATA_CMD[7:0]，然后发送到 I2C 总线。
  - 应答信号：响应的从设备从总线接收数据后，会向主设备发送 ACK 或 NACK 信号，以确认数据是否成功接收。
- 主设备接收操作：
  - 发起寻址：主设备通过 I2C_DATA_CMD[8] 写 1 进行接收请求。
  - 数据读取：响应的从设备将数据发送到 I2C 总线，由主设备从 I2C_DATA_CMD[7:0] 获取数据。
  - 应答信号：主设备在接收到数据后，会向从设备发送 ACK 或 NACK 信号。
- 从设备发送操作：
  - 接收请求：接收到主设备的接收请求后，从设备将数据写进 I2C_DATA_CMD[7:0]，然后发送到 I2C 总线。
  - 等待应答：从设备发送完数据后，等待主设备设备的应答信号。
- 从设备接收操作：
  - 接收数据：接收到主设备的发送请求后，从设备从 I2C_DATA_CMD[7:0] 获取数据。
  - 应答信号：从设备在接收数据后，向主设备发送 ACK 或 NACK 信号。
- 主设备发起寻址，发起 START 信号，生成 SCL，生成 STOP 信号，可发送和接收。从设备作为被寻址的一方，发起 ACK 或 NACK 信号，可发送和接收。

注：

I2C_DATA_CMD[8] 只能在主模式下配置，I2C 任何的通信行为都由主设备主动发起。

7.5.3.3. 7–bit & 10–bit 寻址

I^2C 协议的寻址方式主要分为 7 位寻址和 10 位寻址，两种方式各有其特点和应用场景，以适应不同规模和复杂度的系统需求。了解寻址方式有助于正确配置和使用 I^2C 设备，实现高效的数据传输和通信。

7-bit 寻址格式：允许主设备通过发送一个字节来寻址从设备，进行数据的读写操作。

在 7 位寻址模式下，I2C 设备的从机地址由 7 位组成，第 8 位是读写位（R/W），用于指示数据的方向：0 表示写操作，1 表示读操作。具体来说，I2C_TAR[6:0] 对应此 7 位地址，而 I2C_DATA_CMD[8] 则作为 R/W 位。

	MSB						LSB		
START	A6	A5	A4	A3	A2	A1	A0	R/W	ACK

注：

在某些情况下，从机设备可能支持 START 字节，即主设备先发起 0000 0001 字节（不需要 ACK），再发起 START 信号进行寻址。

10-bit 寻址格式：

在 10 位寻址模式下，寻址过程分为两个字节发送，确保了即使地址空间扩大到 10 位，也能保持与 7 位寻址的兼容性：

- 第一个字节：该字节的 bit[7:3] 向从设备指示这是一个 10bit 的地址。紧接着的 bit[2:1]为 I2C_TAR[9:8]，而 R/W 为 I2C_DATA_CMD[8]。

  这个字节用于告诉从机接下来的是一个 10 位地址。

- 第二个字节：这个字节直接包含 I2C_TAR[7:0]，即从机的后 8 位地址。

START	1	1	1	1	0	A9	A8	R/W	A7	A6	A5	A4	A3	A2	A1	A0	ACK

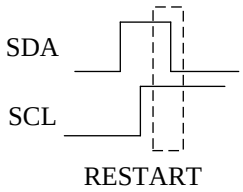
7.5.3.4. Restart 说明

在 I^2C 通信协议中，Restart（重复起始条件）是一种重要的信号，允许主设备在不产生停止条件的情况下继续与从设备进行通信。这种机制使得数据传输更加高效和连贯，尤其是在需要连续执行多次读写操作的场景中。

与起始条件类似，重复起始条件发生在停止条件之前。主设备在一个字节传输完成后，不产生停止条件而是发送重复起始条件来启动下一次传输，以便继续给从设备发送消息。

Restart 的时序要求与起始条件类似，都是在 SCL（串行时钟线）为高电平时，产生一个 SDA（串行数据线）的下降沿。具体来说，需要先令 SDA 和 SCL 都处于高电平状态，再令 SDA 拉低，以形成重复起始条件的时序。在 Restart 过程中，SDA 的下降沿必须在 SCL 为高电平时发生，这是判断是否为有效 Restart 的关键。

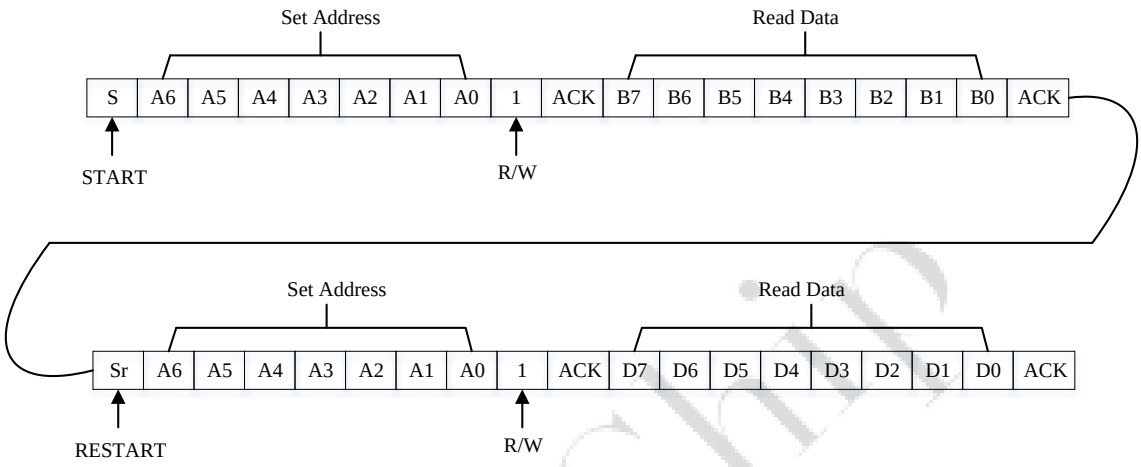
Restart 时序：



Restart 传输时序具有多种应用场景，示例如下：

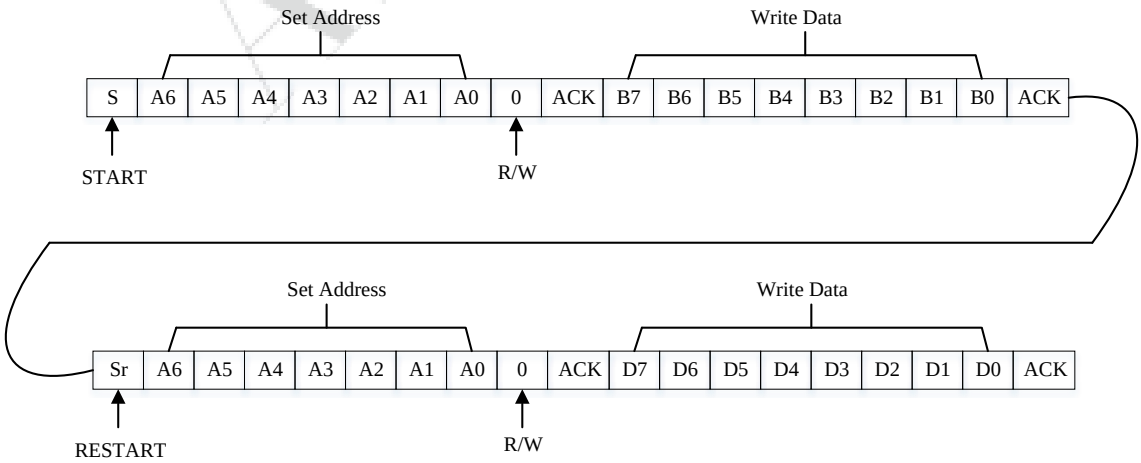
多次读数据：

- 当主设备需要从从设备的多个寄存器中读取数据时，可以使用 Restart 来避免在每次读取之间产生停止条件。
- 主设备首先发送起始条件和从设备地址，然后发送要读取的第一个寄存器的地址。在接收到从设备返回的数据后，主设备不产生停止条件，而是直接发起 Restart，再次发送起始条件和从设备地址，以及下一个要读取的寄存器的地址。如此重复，直到所有需要的数据都被读取完毕。



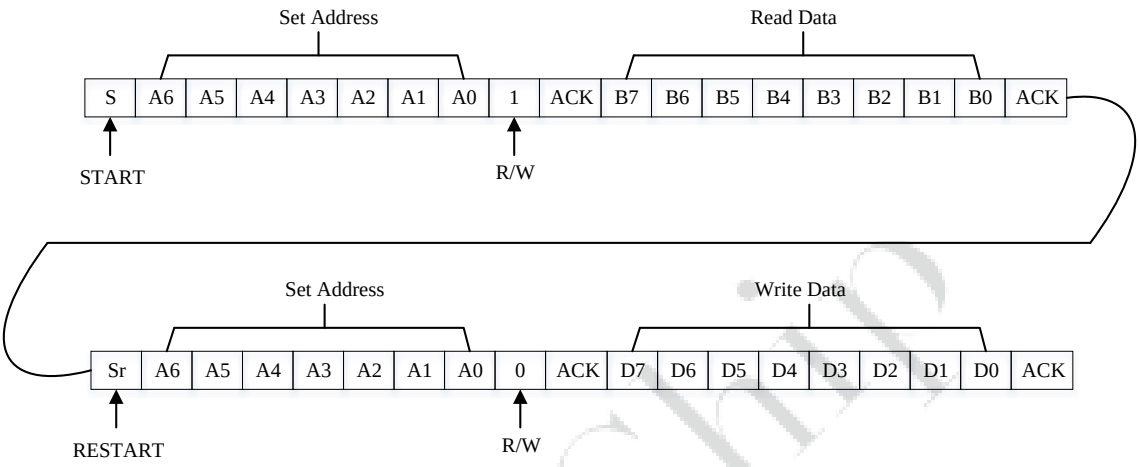
多次写数据：

- 当主设备需要从从设备的多个寄存器中写入数据时，可以使用 Restart 来避免在每次写入之间产生停止条件。
- 主设备首先发送起始条件和从设备地址，然后发送要写入的第一个寄存器的地址和数据。在完成这次写入后，主设备不产生停止条件，而是直接发起 Restart，再次发送起始条件和从设备地址，以及下一个要写入的寄存器的地址和数据。如此重复，直到所有需要的数据都被写入完毕。



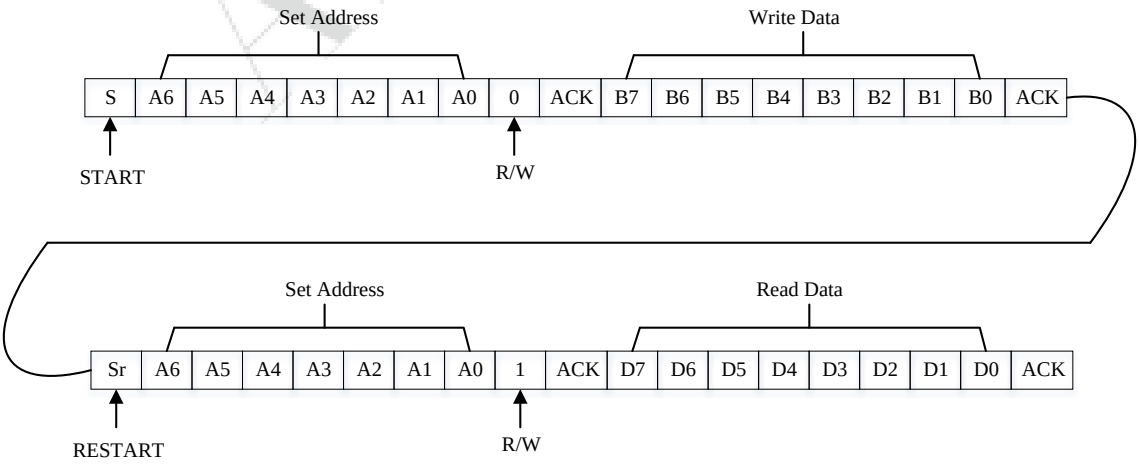
读数据转写数据：

- 在某些情况下，主设备可能需要先从从设备读取数据，然后根据读取到的数据来决定是否需要写入新的数据。
- 主设备首先发送起始条件和从设备地址，然后发送要读取的寄存器的地址。在接收到从设备返回的数据后，主设备可以根据需要决定是否发起 Restart 并切换到写模式。如果需要写入新的数据，则主设备再次发送起始条件和从设备地址，以及要写入的寄存器的地址和数据。



写数据转读数据：

- 某些情况下，主设备可能需要先向从设备写入数据，然后再读取从设备的状态或响应。
- 主设备首先发送起始条件和从设备地址，然后发送要写入的寄存器的地址和数据。在完成这次写入后，如果主设备需要读取从设备的状态或响应，则可以发起 Restart 并切换到读模式。具体来说，主设备再次发送起始条件和从设备地址，然后发送要读取的寄存器的地址。从设备会根据请求返回相应的数据或状态信息给主设备。



7.5.3.5. SDA_SETUP_TIME 和 SDA_HOLD_TIME 说明

在 I^2C 通信协议中，SDA_SETUP_TIME 和 SDA_HOLD_TIME 是两个至关重要的时序参数，确保了数据在传输过程中的准确性和稳定性。这两个参数主要涉及到 SDA（串行数据线）在 SCL（串行时钟线）上升沿或下降沿前后的状态保持时间。

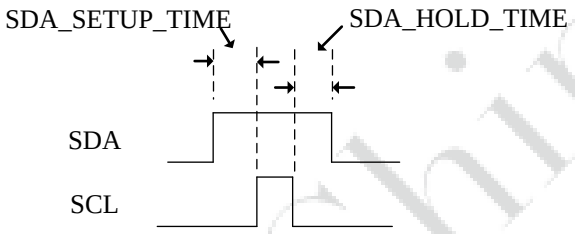
- SDA_SETUP_TIME：SCL 延迟时间，表示 SDA 在 SCL 上升沿到来之前需要保持稳定的时间，仅当 I2C 作为 从设备发送数据时使用。

  SDA_SETUP_TIME 确保了接收设备能够在 SCL 上升沿到来之前正确地读取 SDA 上的数据。如果 SDA 在 SCL 上升沿到来之前未能稳定，接收设备可能会读取到错误的数据。

- SDA_HOLD_TIME：SDA 保持时间，表示 SDA 在 SCL 上升沿到来之后需要保持稳定的时间。

  SDA_HOLD_TIME 确保了接收设备在 SCL 上升沿到来之后有足够的时间来读取并处理 SDA 上的数据。如果 SDA 在 SCL 上升沿到来之后未能保持稳定，接收设备可能无法正确读取数据。

SETUP 和 HOLD 说明：



7.5.3.6. START BYTE 机制

START BYTE 机制是一种用于 I^2C 通信中的技术，特别适用于片上系统没有专用的 I2C 模块的情况。这种机制通过延长开始条件的过程，使得处理器可以以较低的频率采样 SDA 线，从而降低功耗并提高系统效率。通过在生成 START 条件后，继续发送一个特定的字节信号 (0000_0001)，主设备持续的将 SDA 线拉低七个周期，使得处理器可以在低速下侦听总线，一旦检测到起始字节中的任一 0 位，便切换到高速模式进行后续的数据传输。

当 I2C 设为从设备时，无需使用 START BYTE 机制。

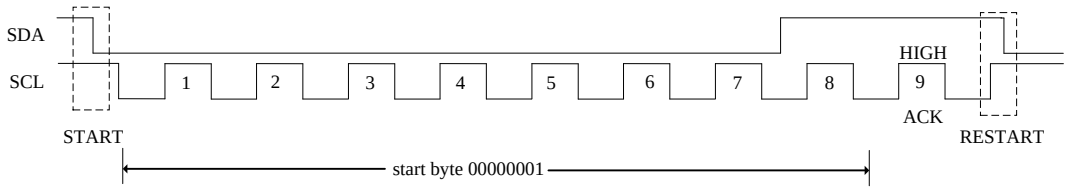
START BYTE 的过程：

1. 主设备先产生一个 START 条件。
2. 主设备 发送 byte（0000_0001），并发送 ACK 信号。

   主设备在整个字节发送期间保持 SDA 线为低电平。
3. 在此过程中，不能有任何从设备响应 ACK。
4. 主设备生成 restart 信号，标志着 START BYTE 机制的结束，随后进入正常的数据传输阶段。

START BYTE 时序要求如下：

- START 条件要求在 SCL 为高电平时，SDA 发生高到低的跳变；随后在 SCL 的第一个时钟周期内，SDA 必须保持低电平。
- 在发送特定字节信号时，主设备需确保在每个时钟周期内 SDA 线的正确电平状态，特别是要在整个字节期间保持 SDA 为低电平。
- RESTART 信号的产生时机是在特定字节信号发送完毕后，此时 SCL 应处于低电平状态，而 SDA 需由低电平跳变至高电平。



在整个 START BYTE 过程中，主设备需要严格控制时钟信号（SCL）和数据信号（SDA）的配合，确保时序的准确性和稳定性。

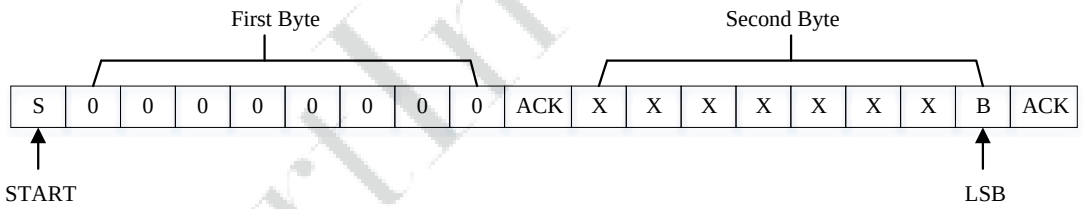
7.5.3.7. 通用广播地址

通用广播地址 (General call address) 是 I^2C 协议中的一种特殊机制，用于同时寻址总线上的所有设备。这种机制允许主设备通过发送特定的地址来广播消息或命令到所有连接在 I2C 总线上的从设备。

- 不需要使用广播功能的设备，可以不响应广播。
- 需要使用广播功能的设备，则在检测到广播地址后发送响应，并作为从设备接收读取总线上发送的数据。

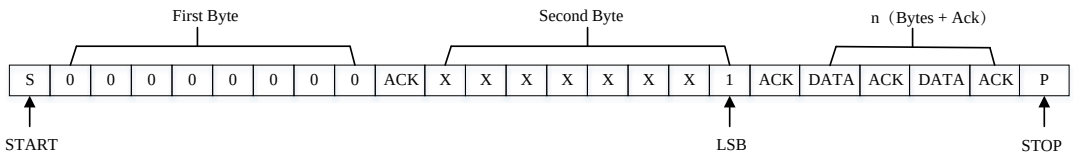
  主设备不知道总线上发送响应的从设备数量，总线上所有可以响应广播的从设备读取广播地址后的第二个以及后面的字节，不能处理广播数据的 从设备不发送响应来忽略广播。因此主设备检测不到未响应的从设备

广播信息的含义在第二个字节决定，如下所示：



GEN_CALL 格式有两种情况，详细说明如下：

1. 第二个 Byte 最低位 B 为 0：
   - 0000_0110(06h)：设备复位并通过硬件写数据到从设备的可编程部分。所有可以响应此类广播的从设备，收到此两个字节后，进行复位并进入其地址的可编程部分。
   - 0000_0100(04h)：通过硬件写内容到从设备的可编程部分，作用类似 (06h)，但设备不复位。
   - 0000_0000(00h)：不允许作为第二个字节使用。
2. 第二个 Byte 最低位 B 为 1：
   - 该两字节广播定义为硬件广播，由一个硬件 主设备器件发送，例如键盘扫描器。硬件广播不能编程，无法发送一个期望的从设备地址。由于硬件主设备预先不知道报文要传输给哪个器件，只能产生包含自身地址的硬件广播呼叫，便于系统识别。
   - 第二个字节中前 7 位是硬件主设备的地址，可由连接到总线的智能器件识别并指引硬件主设备的信息，例如，一个连接到总线的微控制器。如果硬件 主设备也可以作为从设备，其从设备地址和 主设备地址相同。
   - 硬件广播格式：



7.5.3.8. 总线挂死恢复机制

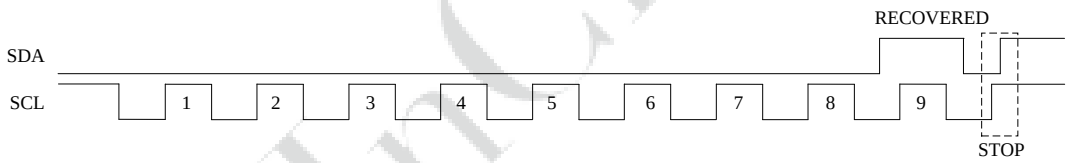
总线挂死恢复分为以下情况：

- **SDA 挂死恢复**：当检测到 SDA 挂死条件时，会触发恢复机制。

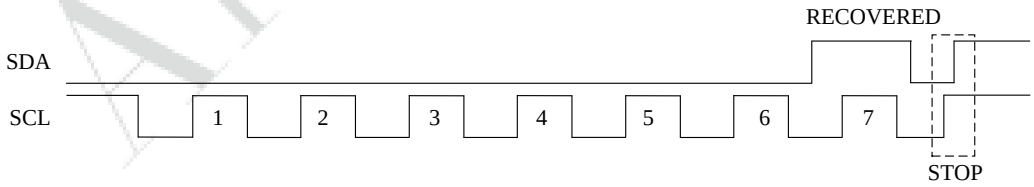
  I2C 通信异常可能会造成总线挂死情况，例如从设备应答 ACK 拉低 SDA 时出现异常。此时设备无法释放总线，造成主设备无法继续掌握总线进行通信。

  1. 主设备尝试在 SDA 发送逻辑 1 并且检查 SDA 是否恢复，最多连续发送 9 个时钟周期。
  2. 如果 SDA 在九个时钟周期内恢复，主设备则发送一个 STOP 信号释放总线。

     下图表示第九个时钟恢复：



     下图表示第七个时钟恢复



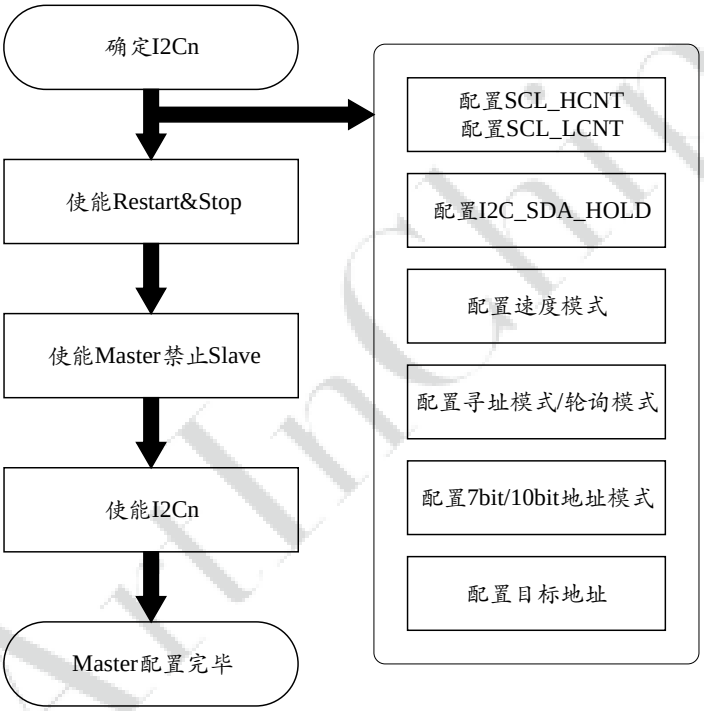
  3. 如果 SDA 在九个时钟周期内未恢复，则系统需要硬件复位。

- **SCL 挂死恢复**

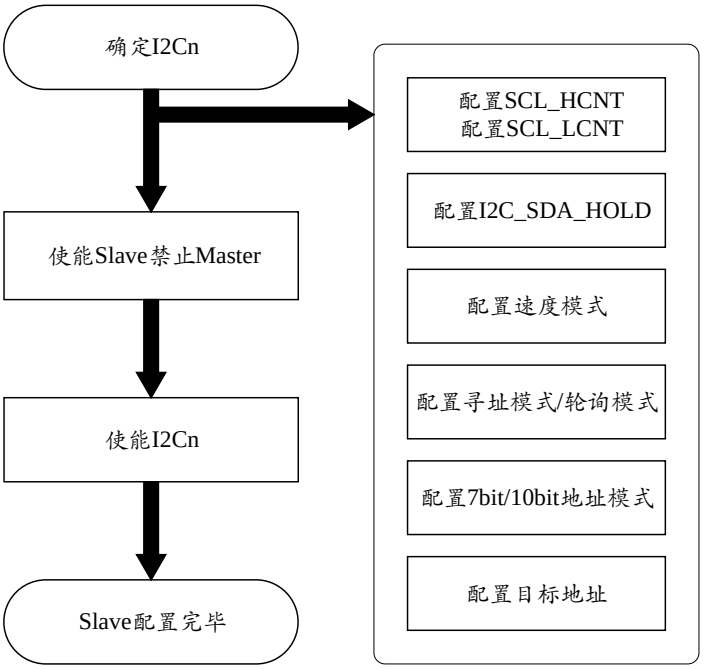
  通常只有在电路电气发生故障才存在 SCL 一直处于低电平，除了硬件复位系统外，没有有效的方法来解决此问题，若使能 SCL 挂死中断，可查看 I2C_INTR_STAT[14] 检查 SCL 状态。

7.5.4. 编程指南

7.5.4.1. 主模式初始化流程



7.5.4.2. 从模式初始化流程



7.5.5. 寄存器列表

表 7–28 I2C 寄存器列表

偏移地址	缩写	寄存器名称	章节
0x000	I2C_CTL	I2C 控制	0x000 I2C_CTL
0x004	I2C_TAR	I2C 目标设备	0x004 I2C_TAR
0x008	I2C_SAR	I2C 从机地址	0x008 I2C_SAR
0x00C	I2C_ACK_GEN_CALL	I2C 响应广播	0x00C I2C_ACK_GEN_CALL
0x010	I2C_DATA_CMD	I2C 数据命令	0x010 I2C_DATA_CMD
0x020	I2C_SS_SCL_HCNT	I2C 标准速度时钟高位计数	0x020 I2C_SS_SCL_HCNT
0x024	I2C_SS_SCL_LCNT	I2C 标准速度时钟低位计数	0x024 I2C_SS_SCL_LCNT
0x028	I2C_FS_SCL_HCNT	I2C 快速时钟高位计数	0x028 I2C_FS_SCL_HCNT
0x02C	I2C_FS_SCL_LCNT	I2C 快速时钟低位计数	0x02C I2C_FS_SCL_LCNT
0x030	I2C_SDA_HOLD	I2C SDA 保持时间	0x030 I2C_SDA_HOLD
0x034	I2C_SDA_SETUP	I2C SDA 建立时间	0x034 I2C_SDA_SETUP
0x038	I2C_INTR_MASK	I2C 中断屏蔽	0x038 I2C_INTR_MASK
0x03C	I2C_INTR_CLR	I2C 中断清除	0x03C I2C_INTR_CLR
0x040	I2C_RAW_INTR_STAT	I2C 中断状态	0x040 I2C_RAW_INTR_STAT
0x048	I2C_ENABLE	I2C 使能	0x048 I2C_ENABLE
0x04C	I2C_ENABLE_STATUS	I2C 使能状态	0x04C I2C_ENABLE_STATUS
0x050	I2C_STATUS	I2C 状态	0x050 I2C_STATUS
0x054	I2C_TX_ABRT_SOURCE	I2C 发送停止源	0x054 I2C_TX_ABRT_SOURCE
0x090	I2C_RX_TL	I2C 接收阈值	0x090 I2C_RX_TL
0x094	I2C_TX_TL	I2C 发送阈值	0x094 I2C_TX_TL
0x098	I2C_TXFLR	I2C 发送 FIFO 水位	0x098 I2C_TXFLR

表 7–28 I2C 寄存器列表 (续)

偏移地址	缩写	寄存器名称	章节
0x09C	I2C_RXFLR	I2C 接收 FIFO 水位	0x09C I2C_RXFLR
0x0A0	I2C_SCL_STUCK_TIMEOUT	I2C SCL 挂死超时	0x0A0 I2C_SCL_STUCK_TIMEOUT
0x0A4	I2C_SDA_STUCK_TIMEOUT	I2C SDA 挂死超时	0x0A4 I2C_SDA_STUCK_TIMEOUT
0x0B0	I2C_FS_SPIKELEN	I2C 快速模式毛刺抑制时长	0x0B0 I2C_FS_SPIKELEN
0x0FC	VERSION	I2C 版本号	0x0FC VERSION

7.5.6. 寄存器描述

7.5.6.1. 0x000 I2C_CTL

默认值：0x0000006F			I2C 控制（I2C Control）
位域	类型	默认值	描述
31:11	–	–	–
10	R/W	0x0	BUS_CLEAR_FEATURE_CTL 总线清除控制 • 0：BUS_CLEAR_FEATURE 禁止 • 1：BUS_CLEAR_FEATURE 使能 此位置 1，SDA 挂死机制使能以及 SCL_STUCK_AT_LOW 中断可触发 仅主模式有效
9	R/W	0x0	RX_FIFO_FULL_HLD_CTL RX_FIFO 满状态保持总线控制，仅在 主模式下： • 0：RX_FIFO 满状态时数据溢出 • 1：RX_FIFO 满状态时保持总线
8	R/W	0x0	TX_EMPTY_CTL TX_EMPTY 中断控制 • 0：TX_EMPTY 中断行为 0 • 1：TX_EMPTY 中断行为 1 （详细描述见 I2C_INTR_STAT[4]）

默认值：0x0000006F			I2C 控制（I2C Control）
位域	类型	默认值	描述
7	R/W	0x0	STOP_DET_IFADDRESSED STOP_DET 中断控制，在从模式下： • 0：不管被寻址与否，都发出 STOP_DET 中断 • 1：仅在被寻址时发出 STOP_DET 中断 **注：** 此位设为 1 时，且在 GEN_CALL 下，即使响应也不会发出 STOP_DET 中断
6	R/W	0x1	RESTART_ENABLE 重开始使能控制 • 0：主模式 RESTART 禁止 • 1：主模式 RESTART 使能 主设备 发送 RESTART 信号控制，一些旧从设备不支持处理 RESTART 信号。当 RESTART 禁止时，主设备 被禁止执行以下功能： • 发送 START 字节 • 使用 10bit 寻址模式进行读操作
5:4	R/W	0x2	SPEED_MODE_SELECT 速率模式选择 • 1：标准模式（100Kb/s） • 2：快速模式（<=400Kb/s） 此位只能写 1 或者 2。
3	R/W	0x1	10BITADDR_SELECT_SLAVE 从模式被寻址 bit 选择 • 0：7 bits 寻址模式 • 1：10 bits 寻址模式
2	R/W	0x1	10BITADDR_SELECT_MASTER 主模式寻址 bit 选择 • 0：7 bits 寻址模式 • 1：10 bits 寻址模式
1	R/W	0x1	SLAVE_MODE_DISABLE 从模式使能控制

默认值：0x0000006F			I2C 控制（I2C Control）
位域	类型	默认值	描述
			• 0：从模式使能 • 1：从模式禁止 当 I2C_CTL[0]设置为 1 时，此位也需设置为 1。I2C_CTL [0]设置为 0 时，此位也设置为 0。
0	R/W	0x1	MASTER_MODE_ENABLE 主模式使能控制 • 0：主模式禁止 • 1：主模式使能 当 I2C_CTL [1]设置为 1 时，此位也需设置为 1。I2C_CTL [1]设置为 0 时，此位也设置为 0。

7.5.6.2. 0x004 I2C_TAR

默认值：0x0000005A			I2C 目标设备（I2C Target）
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0	GEN_CALL_CTL • 0：GEN_CALL 模式禁止 • 1：GEN_CALL 模式使能 此位配 1，当 bit[10]为 0 时，主设备进行广播模式寻址，无视 bit[9:0]中的目标地址，进行 GEN_CALL 模式，GEN_CALL 模式只允许进行写操作，尝试读操作会导致 TX_ABRT（I2C_INTR_STAT[6]）中断产生。当 bit[10]为 1 时，start 字节使能时，主设备先发起 start 字节，再进行目标寻址。
10	R/W	0	START_BYTE_CTL Start 字节控制 • 0：start 字节禁止 • 1：start 字节使能
9:0	R/W	0x5A	I2C_TAR 目标地址 主设备依此目标地址寻址建立通信。如果 I2C_TAR 和 I2C_SAR 相同，则存在回环，但是 FIFO 在主设备与从设备之间共享，因此双向传输是不可行的，只能单向传输。主设备不能传输数据给自己，只能传输给从设备。 当进行 GEN_CALL 时，无视此地址。

7.5.6.3. 0x008 I2C_SAR

默认值：0x0000005A			I2C 从机地址（I2C Slave Address）
位域	类型	默认值	描述
31:10	–	–	–
9:0	R/W	0x5A	I2C_SAR 从模式被寻址的地址 当为 7bit 寻址时，只有 bit[6:0]有效。此位仅在 I2C 禁止时可写（I2C_ENABLE[0]写 0），其它时间写入无效。

7.5.6.4. 0x00C I2C_ACK_GEN_CALL

默认值：0x00000001			I2C 响应广播（I2C Acknowledge General Call）
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x1	ACK_GEN_CALL GEN_CALL 应答状态。 • 0：接收到 GEN_CALL 地址后不应答 • 1：接收到 GEN_CALL 地址后应答

7.5.6.5. 0x010 I2C_DATA_CMD

默认值：0x00000000			I2C 数据命令（I2C Data Command）
位域	类型	默认值	描述
31:11	–	–	–
10	W	0x0	RESTART 重开始行为控制 • 0：当传输方向与上一个传输命令的方向不同时。如果 RESTART_ENABLE(I2C_CTL[6]) 为 1，发出 RESTART 信号；如果 RESTART_ENABLE (I2C_CTL[6]) 为 0，则在发送一个 STOP 信号后接着发送一个 START 信号。 • 1：无论当前数据传输方向与上一个传输命令的方向是否相同。如果 RESTART_ENABLE(I2C_CTL[6]) 为 1，发出 RESTART 信号；如果 RESTART_ENABLE (I2C_CTL[6]) 为 0，则在发送一个 STOP 信号后接着发送一个 START 信号。

默认值：0x00000000			I2C 数据命令（I2C Data Command）
位域	类型	默认值	描述
9	W	0x0	STOP STOP 命令控制 • 0：一个 byte 传输完后不接着 STOP 信号 • 1：一个 byte 传输完后接着 STOP 信号
8	W	0x0	CMD 操作命令，仅在主模式下有效 • 0：主模式写操作命令 • 1：主模式读操作命令 注： 如果在收到 RD_REQ 请求后此位写 1，则产生 TX_ABRT 中断。
7:0	R/W	0x0	DAT 数据缓冲区 发送数据时，数据需写入此位段。接收数据时，数据亦接收到此位段。

7.5.6.6. 0x020 I2C_SS_SCL_HCNT

默认值：0x00000060			I2C 标准速度时钟高位计数（I2C Standard SCL High Count）
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x60	I2C_SS_SCL_HCNT 标准速率模式下 SCL 高电平持续时间计数。 此位仅在 I2C 禁止时可写（I2C_ENABLE[0] 写 0），其它时间写入无效。最小值为 0x6，写入小于 0x6 的数值无效。 计算方法： • 标准速率 SCL 高电平最小时间：4000 ns • 标准速率 SCL 低电平最小时间：4700 ns 以标准速度模式为例： • SCL 高电平时间 = (I2C_SS_SCL_HCNT + I2C_FS_SPIKELEN + 8) * I2C_clk_period • I2C_clk_period 为 I2C 模块输入时钟周期

7.5.6.7. 0x024 I2C_SS_SCL_LCNT

默认值：0x00000070			I2C 标准速度时钟低位计数（I2C Standard SCL Low Count）
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x70	I2C_SS_SCL_LCNT 标准速率模式下 SCL 低电平持续时间计数。 此位仅在 I2C 禁止时可写（I2C_ENABLE[0] 写 0），其它时间写入无效。最小值为 0x8，写入小于 0x8 的数值无效。 计算方法： • 标准速率 SCL 高电平最小时间：4000 ns • 标准速率 SCL 低电平最小时间：4700 ns 以标准速度模式为例 • SCL 低电平时间 = (I2C_SS_SCL_LCNT + 1) * I2C_clk_period • I2C_clk_period 为 I2C 模块输入时钟周期。

7.5.6.8. 0x028 I2C_FS_SCL_HCNT

默认值：0x0000000F			I2C 快速时钟高位计数（I2C Fast SCL High Count）
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0xF	I2C_FS_SCL_HCNT 快速模式下 SCL 高电平持续时间计数。 此位仅在 I2C 禁止时可写（I2C_ENABLE[0]写 0），其它时间写入无效。最小值为 0x6，写入小于 0x6 的数值无效。 计算方法： • 快速 SCL 高电平最小时间：600ns • 快速 SCL 低电平最小时间：1300ns 以快速模式为例 • SCL 高电平时间 = (I2C_FS_SCL_HCNT + I2C_FS_SPIKELEN + 8) * I2C_clk_period • I2C_clk_period 为 I2C 模块输入时钟周期

7.5.6.9. 0x02C I2C_FS_SCL_LCNT

默认值：0x0000001F			I2C 快速时钟低位计数（I2C Fast SCL Low Count）
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x1F	I2C_FS_SCL_LCNT 快速模式下 SCL 低电平持续时间计数。此位仅在 I2C 禁止时可写（I2C_ENABLE[0]写 0），其它时间写入无效。最小值为 0x8，写入小于 0x8 的数值无效。 计算方法： • 快速 SCL 高电平最小时间：600ns • 快速 SCL 低电平最小时间：1300ns 以快速模式为例 • SCL 低电平时间 = (I2C_FS_SCL_LCNT + 1) * I2C_clk_period • I2C_clk_period 为 I2C 模块输入时钟周期

7.5.6.10. 0x030 I2C_SDA_HOLD

默认值：0x00000001			I2C SDA 保持时间（I2C SDA Hold Time）
位域	类型	默认值	描述
31:24	–	–	–
23:16	R/W	0x0	I2C_SDA_RX_HOLD I2C 接收时设置所需的 SDA 保持时间，以 I2C_clk 周期为单位
15:0	R/W	0x1	I2C_SDA_TX_HOLD I2C 发送时设置所需的 SDA 保持时间，以 I2C_clk 周期为单位

7.5.6.11. 0x034 I2C_SDA_SETUP

默认值：0x00000002			I2C SDA 建立时间（I2C SDA Setup Time）
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x2	I2C_SDA_SETUP SDA 设置时间，数值为（该值–1），以 I2C_clk 周期为单位。

默认值：0x00000002			I2C SDA 建立时间（I2C SDA Setup Time）
位域	类型	默认值	描述
			如默认值为(2–1)个 cycle 建议如果需要延迟 1000ns，则对于 10Mhz 的 I2C，时钟频率此字段设为 11，此字段最小值设 2

7.5.6.12. 0x038 I2C_INTR_MASK

默认值：0x000048FF			I2C 中断屏蔽（I2C Interrupt Mask）
位域	类型	默认值	描述
31:15	–	–	–
14	R/W	0x1	M_SCL_STUCK_AT_LOW • 0：中断屏蔽 • 1：中断未屏蔽
13	R/W	0x0	M_MASTER_ON_HOLD • 0：中断屏蔽 • 1：中断未屏蔽
12	–	–	–
11	R/W	0x1	M_GEN_CALL • 0：中断屏蔽 • 1：中断未屏蔽
10	R/W	0x0	M_START_DET • 0：中断屏蔽 • 1：中断未屏蔽
9	R/W	0x0	M_STOP_DET • 0：中断屏蔽 • 1：中断未屏蔽
8	R/W	0x0	M_ACTIVITY • 0：中断屏蔽 • 1：中断未屏蔽
7	R/W	0x1	M_RX_DONE

默认值：0x000048FF			I2C 中断屏蔽（I2C Interrupt Mask）
位域	类型	默认值	描述
			• 0：中断屏蔽 • 1：中断未屏蔽
6	R/W	0x1	M_TX_ABRT • 0：中断屏蔽 • 1：中断未屏蔽
5	R/W	0x1	M_RD_REQ • 0：中断屏蔽 • 1：中断未屏蔽
4	R/W	0x1	M_TX_EMPTY • 0：中断屏蔽 • 1：中断未屏蔽
3	–	–	–
2	R/W	0x1	M_RX_FULL • 0：中断屏蔽 • 1：中断未屏蔽
1	–	–	–
0	R/W	0x1	M_RX_UNDER • 0：中断屏蔽 • 1：中断未屏蔽

7.5.6.13. 0x03C I2C_INTR_CLR

默认值：0x00000000			I2C 中断清除（I2C Interrupt Clear）
位域	类型	默认值	描述
31:15	–	–	–
14	R/W1C	0x0	CLR _SCL_STUCK_AT_LOW • 0：中断未激活 • 1：写 1 清零

默认值：0x00000000			I2C 中断清除（I2C Interrupt Clear）
位域	类型	默认值	描述
			详细描述见 I2C_INTR_STAT
13	R/W1C	0x0	CLR _M_MASTER_ON_HOLD • 0：中断未激活 • 1：写 1 清零
12	–	–	–
11	R/W1C	0x0	CLR _GEN_CALL • 0：中断未激活 • 1：写 1 清零 ACK 需持续一个 SCL 周期，在此期间无法对 GEN_CALL 中断清零详细描述见 I2C_INTR_STAT
10	R/W1C	0x0	CLR _START_DET • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT
9	R/W1C	0x0	CLR _STOP_DET • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT
8	R/W1C	0x0	CLR _ACTIVITY • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT
7	R/W1C	0x0	CLR _RX_DONE • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT
6	R/W1C	0x0	CLR _TX_ABRT • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT

默认值：0x00000000			I2C 中断清除（I2C Interrupt Clear）
位域	类型	默认值	描述
5	R/W1C	0x0	CLR _RD_REQ • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT
4:1	–	–	–
0	R/W1C	0x0	CLR _RX_UNDER • 0：中断未激活 • 1：写 1 清零 详细描述见 I2C_INTR_STAT

7.5.6.14. 0x040 I2C_RAW_INTR_STAT

默认值：0x00000000			I2C 中断状态（I2C Interrupt Status）
位域	类型	默认值	描述
31:15	–	–	–
14	R	0x0	SCL_STUCK_AT_LOW • 0：中断未触发 • 1：中断触发 指示 SCL 处于低电平的持续时间是否为 I2C_SCL_STUCK_TIMEOUT 设置的时间，需配置 I2C_CTL[10]为 1
13	R	0x0	MASTER_ON_HOLD • 0：中断未触发 • 1：中断触发 指示 主设备是否保持总线并且 TX FIFO 为空，硬件自动清零
12	–	–	–
11	R	0x0	GEN_CALL • 0：中断未触发 • 1：中断触发 仅在接收到 GEN_CALL 地址并且回复 ACK 时触发。
10	R	0x0	START_DET

默认值：0x00000000			I2C 中断状态（I2C Interrupt Status）
位域	类型	默认值	描述
			• 0：中断未触发 • 1：中断触发 指示主或者 从模式下，是否检测到一个 START 信号
9	R	0x0	STOP_DET • 0：中断未触发 • 1：中断触发 指示 主或者从 模式下，是否检测到一个 STOP 信号。 在 从模式下：当 I2C_CON[7]（STOP_DET_IFADDRESSED）设为 1，中断 STOP_DET 仅在寻址模式下触发，GEN_CALL 模式下不触发。当 I2C_CON[7]（STOP_DET_IFADDRESSED）设为 0，中断 STOP_DET 不管是否被寻址都会触发。
8	R	0x0	ACTIVITY • 0：中断未触发 • 1：中断触发 指示 I2C 捕获活动状态并且一直保持为 1 直到被清零，清除此位有 3 种方式： • 关闭 I2C • I2C_INTR_CLR[8]写 1 清零 • 系统复位 一旦此位被设为 1，即使 I2C 处于空闲状态，也会保持直到以上其中一种方式清零
7	R	0x0	RX_DONE • 0：中断未触发 • 1：中断触发 当 I2C 作为从设备发送数据时，若主设备不再回复一个 ACK，则此位设为 1。此中断在传输的最后一个字节发生，指示传输结束
6	R	0x0	TX_ABRT • 0：中断未触发 • 1：中断触发 指示 I2C 作为发送者无法完成对发送 FIFO 内容的预期操作。主和从模式都可发生此中断。当此中断发生时，I2C_TX_ABRT_SOURCE 寄存器指示发生停止发送的原因
5	R	0x0	RD_REQ • 0：中断未触发 • 1：中断触发

默认值：0x00000000			I2C 中断状态（I2C Interrupt Status）
位域	类型	默认值	描述
			当此位设为 1 时，表明在 从模式下，主设备在尝试读取 I2C 的数据。I2C 将总线保持为等待状态，直到此中断被服务，这意味这从设备已经被主设备寻址。当中断发生时，处理器必须响应此中断，将请求的数据写入 I2C_DATA_CMD 寄存器。
4	R	0x0	TX_EMPTY • 0：中断未触发 • 1：中断触发 TX_EMPTY_CTL 设为 0：当发送缓冲器小于等于 I2C_TX_TL 寄存器设定的阈值时，此位设为 1。 TX_EMPTY_CTL 设为 1：当发送缓冲器小于等于 I2C_TX_TL 寄存器设定的阈值时，并且最新命令的地址/数据从内部的移位寄存器传输完成，此位设为 1。 当缓冲器数据量超过 I2C_TX_TL 寄存器设定值硬件会自动清零。
3	–	–	–
2	R	0x0	RX_FULL • 0：中断未触发 • 1：中断触发 当接收缓冲器达到或超过 I2C_RX_TL 寄存器设定的阈值时，此位设 1。当缓冲器深度低于阈值，硬件自动清零。
1	–	–	–
0	R	0x0	RX_UNDER（接收下溢出） • 0：中断未触发 • 1：中断触发 如果接收缓冲器为空时，通过 I2C_DATA_CMD 读取数据则设为 1。

7.5.6.15. 0x048 I2C_ENABLE

默认值：0x00000000			I2C 使能（I2C Enable）
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	SDA_STUCK_RECOVERY_ENABLE SDA 挂死恢复使能控制 • 0: 主模式禁止 SDA 挂死恢复机制 • 1: 主模式使能 SDA 挂死恢复机制

默认值：0x00000000			I2C 使能（I2C Enable）
位域	类型	默认值	描述
			如果 SDA 挂死状态由 I2C_TX_ABRT_SOURCE[17] 指示的 TX_ABORT 中断，然后此位发起一个恢复机制，即发送 9 个 SCL 时钟和一个 STOP 信号释放 SDA 线，之后此位清零
2	–	–	–
1	R/W	0x0	ABORT， 传输停止控制 • 0：不发起传输停止或者传输停止已完成 • 1：停止正在进行的操作 软件设置此位可以停止主模式下的传输。只有在 ENABLE 设为 1 的情况下才能对此位操作，否则操作无效。一旦 ABORT 设为 1，软件则无法清除。响应一个 ABORT，控制器在当前传输完成后发出 STOP 信号并且冲刷 TX FIFO，然后在完成停止操作后设置 TX_ABORT 中断（I2C_INTR_STAT[6]）
0	R/W	0x0	ENABLE I2C 使能控制 • 0：I2C 禁止 • 1：I2C 使能

7.5.6.16. 0x04C I2C_ENABLE_STATUS

默认值：0x00000000			I2C 使能状态（I2C Enable Status）
位域	类型	默认值	描述
31:3	–	–	–
2	R	0x0	SLV_RX_DATA_LOST 从模式接收数据丢失。 • 0：从模式接收数据不丢失 • 1：从模式接收数据丢失 指示从模式接收下是否由于 I2C_ENABLE[0]由 1 设为 0 导致至少一个数据丢失。
1	R	0x0	SLV_DISABLED_WHILE_BUSY 从模式禁用时状态 • 0：从模式活跃时禁用 • 1：从模式空闲时禁用 指示由于 I2C_ENABLE[0]由 1 设为 0 时，导致活跃或者潜在的 从设备操作被停止。

默认值：0x00000000			I2C 使能状态（I2C Enable Status）
位域	类型	默认值	描述
0	R	0x0	I2C_EN_STATUS I2C 使能状态 • 0：I2C 禁止状态 • 1：I2C 使能状态

7.5.6.17. 0x050 I2C_STATUS

默认值：0x00000006			I2C 状态（I2C Status）
位域	类型	默认值	描述
31:12	–	–	–
11	R	0x0	SDA_STUCK_NOT_RECOVERED SDA 挂死未恢复 • 1：SDA 挂死触发恢复机制后未恢复 • 0：SDA 挂死触发恢复机制后已恢复 此位仅在主模式有效，从模式无效
10	R	0x0	SLV_HOLD_RX_FIFO_FULL 从设备因 RX FIFO 为满保持总线 • 0：从设备未保持总线或者总线保持非因 RX FIFO 为满 • 1：RX FIFO 为满，从设备保持总线 指示当 RX FIFO 为满并且接收额外的字节数据，从设备则保持总线停止接收。
9	R	0x0	SLV_HOLD_TX_FIFO_EMPTY 从设备因 TX FIFO 为空保持总线 • 0：从设备未保持总线或者总线保持非因 TX FIFO 为空 • 1：TX FIFO 为空，从设备保持总线 指示当从设备收到读请求而 TX FIFO 为空，从设备则保持总线直到 TX FIFO 有数据响应读请求
8	R	0x0	MST_HOLD_RX_FIFO_FULL 主设备因 RX FIFO 为满保持总线 • 0：主设备未保持总线或者总线保持非因 RX FIFO 为满 • 1：RX FIFO 为满，主设备保持总线 指示当 RX FIFO 为满并且接收额外的字节数据，主设备则保持总线停止接收。

默认值：0x00000006			I2C 状态（I2C Status）
位域	类型	默认值	描述
7	R	0x0	MST_HOLD_TX_FIFO_EMPTY 主设备因 TX FIFO 为空保持总线 • 0：主设备未保持总线或者总线保持非因 TX FIFO 为空 • 1：TX FIFO 为空，主设备保持总线 指示当 TX FIFO 为空并且最后一个字节数据没有 STOP 信号，主设备则保持总线停止发送。
6	R	0x0	SLV_ACTIVITY 从模式状态 • 0：从模式空闲状态 • 1：从模式活跃状态 当从模式处于活跃状态，此位置 1
5	R	0x0	MST_ACTIVITY 主模式状态 • 0：主模式空闲状态 • 1：主模式活跃状态 当主模式有限状态机处于活跃状态，此位置 1
4	R	0x0	RFF（RX FIFO FULL） 接收 FIFO 为满状态标志 • 0：接收 FIFO 非满 • 1：接收 FIFO 为满 当接收 FIFO 包含一个或多个数据时，此位置为 0。FIFO 为满时为 1
3	R	0x0	RFNE（RX FIFO NOT EMPTY） 接收 FIFO 非空状态标志 • 0：接收 FIFO 为空 • 1：接收 FIFO 非空 当接收 FIFO 包含一个或多个数据时，此位置 1。FIFO 为空时清零
2	R	0x1	TFE（TX FIFO EMPTY） 发送 FIFO 为空状态标志 • 0：发送 FIFO 非空 • 1：发送 FIFO 为空

默认值：0x00000006			I2C 状态（I2C Status）
位域	类型	默认值	描述
			当发送 FIFO 为空时，此位置 1。当 FIFO 包含一个或多个数据时清零。此位不请求中断
1	R	0x1	TFNF（TX FIFO NOT FULL） 发送 FIFO 非满状态标志 • 0：发送 FIFO 为满 • 1：发送 FIFO 非满 当发送 FIFO 包含一个或多个数据时，此位置 1。FIFO 为满时清零
0	R	0x0	ACTIVITY I2C 状态 • 0：I2C 处于空闲状态 • 1：I2C 处于活跃状态

7.5.6.18. 0x054 I2C_TX_ABRT_SOURCE

默认值：0x00000000			I2C 发送停止源（I2C Transport Abort Source）
位域	类型	默认值	描述
31:23	R	0x0	TX_FLUSH_CNT。 此位域指示由于 TX_ABRT 中断而冲刷的 TX FIFO 数据命令数。
22:18	–	–	–
17	R	0x0	ABRT_SDA_STUCK_AT_LOW • 0：不生成停止操作 • 1：生成停止操作。 仅在主模式有效，主设备检测 SDA 持续低电平的时间超过 I2C_SDA_STUCK_TIMEOUT 设置的阈值
16	R	0x0	ABRT_USER_ABRT • 0：不生成停止操作 • 1：生成停止操作。 仅在 主模式有效，主设备检测到传输中止 (I2C_ENABLE[1])，生成停止操作。
15	R	0x0	ABRT_SLVRD_INTX • 0：不生成停止操作 • 1：生成停止操作。

默认值：0x00000000			I2C 发送停止源（I2C Transport Abort Source）
位域	类型	默认值	描述
			从模式下正在进行发送数据给远程主设备，此时用户在 I2C_CMD_DATA[8] 写 1，则产生停止操作。
14	R	0x0	ABRT_SLV_ARBLOST • 0：不生成停止操作 • 1：生成停止操作。 指示在向远程主设备传输时，从模式丢失总线，同时 I2C_TX_ABRT_SOURCE[12] 设 1，生成停止操作。
13	R	0x0	ABRT_SLVFLUSH_TXFIFO • 0：不生成停止操作 • 1：生成停止操作。 指示从模式已接收到读取命令，并且某些数据存在于 TX FIFO 中，因此 从模式发出 TX_ABRT 中断以刷新 TX FIFO 中的旧数据，则生成停止操作。
12	R	0x0	ABRT_LOST • 0：不生成停止操作 • 1：生成停止操作。 指示主模式失去仲裁，或者设置了 I2C_TX_ABRT_SOURCE[14],则从模式失去仲裁，生成停止操作
11	R	0x0	ABRT_MASTER_DIS • 0：不生成停止操作 • 1：生成停止操作。 指示用户在主模式禁用的情况下发起一个 主模式操作，则生成停止操作
10	R	0x0	ABRT_10B_RD_NORSTRT • 0：不生成停止操作 • 1：生成停止操作。 指示当 RESTART 禁止时（I2C_CTL[6]设 0），主设备在 10bit 寻址模式发送一个读操作，则生成停止操作
9	R	0x0	ABRT_SBYTE_NORSTRT • 0：不生成停止操作 • 1：生成停止操作。 指示在 RESTART 禁止时（I2C_CTL[6]设 0）用户发送一个 START 字节清除此位，需修复造成此位生成停止操作的源，RESTART 必须先使能（I2C_CTL[6]设 1），I2C_TAR[11]或者 I2C_TAR[10]必须清除。一旦此位的源被修复，此位就可以读取清零。若在修复源之前尝试读取此位清零，则此位在一个周期内清除，随后重新赋值

默认值：0x00000000			I2C 发送停止源（I2C Transport Abort Source）
位域	类型	默认值	描述
8	–	–	–
7	R	0x0	ABRT_SBYTE_ACKDET • 0：不生成停止操作 • 1：生成停止操作。 指示 主设备发送一个 START 字节并且被应答（错误行为），则生成停止操作
6	–	–	–
5	R	0x0	ABRT_GCALL_READ • 0：不生成停止操作 • 1：生成停止操作。 指示 I2C 发送一个 GEN_CALL 但是紧接着用户进行一个读操作（I2C_DATA_CMD[8]写 1），则生成停止操作
4	R	0x0	ABRT_GCALL_NOACK • 0：不生成停止操作 • 1：生成停止操作。 指示主模式发出一个 GEN_CALL,而总线上没有从设备 应答，则生成停止操作
3	R	0x0	ABRT_TXDATA_NOACK • 0：不生成停止操作 • 1：生成停止操作。 仅在 主模式，当发送一个地址并且收到应答，接着发送数据而没有收到应答，则生成停止操作
2	R	0x0	ABRT_10ADDR2_NOACK • 0：不生成停止操作 • 1：生成停止操作。 仅在主模式，当 10bit 寻址模式发送第一个地址并且收到应答，接着发送第二个地址而没有收到应答，则生成停止操作
1	R	0x0	ABRT_10ADDR1_NOACK • 0：不生成停止操作 • 1：生成停止操作。 仅在主模式，当 10bit 寻址模式发送第一个地址没有收到应答，则生成停止操作
0	R	0x0	ABRT_7B_ADDR_NOACK

默认值：0x00000000			I2C 发送停止源（I2C Transport Abort Source）
位域	类型	默认值	描述
			• 0：不生成停止操作 • 1：生成停止操作。 仅在 主模式，当 7bit 寻址模式发送第一个地址没有收到应答，则生成停止操作

7.5.6.19. 0x084 I2C_DMA_CR

默认值：0x00000000			I2C DMA 控制（I2C DMA control）
位域	类型	默认值	描述
31:2	–	–	–
1	R/W	0x0	TDMAE 发送 DMA 使能 • 0：发送 FIFO 不使能发送 DMA 操作 • 1：发送 FIFO 使能 DMA 操作
0	R/W	0x0	RDMAE 接收 DMA 使能 • 0：接收 FIFO 不使能发送 DMA 操作 • 1：接收 FIFO 使能 DMA 操作

7.5.6.20. 0x088 I2C_DMA_TDLR

默认值：0x00000000			I2C DMA 发送阈值配置（I2C DMA Transmit Data Level）
位域	类型	默认值	描述
31:3	–	–	–
2:0	R/W	0x0	DMA TDL 发送 DMA 触发阈值配置 当 TDMAE = 1 时，用于控制触发 DMA 请求的控制逻辑，发送数据数量小于等于此值时触发。有效范围为 0–7。值 0 设置 0 项的阈值，值 7 设置 7 项的阈值

7.5.6.21. 0x08C I2C_DMA_RDLR

默认值：0x00000000			I2C DMA 接收阈值配置（I2C DMA Receive Data Level）
位域	类型	默认值	描述
31:3	–	–	–
2:0	R/W	0x0	DMA RDL 接收 DMA 触发阈值配置 当 RDMAE = 1 时，用于控制触发 DMA 请求的控制逻辑，接收数据数量大于等于此值时触发。有效范围为 0–7。值 0 设置 1 项的阈值，值 7 设置 8 项的阈值

7.5.6.22. 0x090 I2C_RX_TL

默认值：0x00000000			I2C 接收阈值（I2C Receive Threshold）
位域	类型	默认值	描述
31:3	–	–	–
2:0	R/W	0x0	RX_TL 接收 FIFO 阈值深度 控制触发 RX_FULL 中断的数据数量，接收数据数量大于等于此值时触发。有效范围为 0–7，硬件不允许该值设置为大于缓冲寄存器的最大深度。值 0 设置 1 项的阈值，值 7 设置 8 项的阈值

7.5.6.23. 0x094 I2C_TX_TL

默认值：0x00000000			I2C 发送阈值（I2C Transport Threshold）
位域	类型	默认值	描述
31:3	–	–	–
2:0	R/W	0x0	TX_TL 发送 FIFO 阈值深度 控制触发 TX_EMPTY 中断的数据数量，发送数据数量小于等于此值时触发。有效范围为 0–7，硬件不允许该值设置为大于缓冲寄存器的最大深度。值 0 设置 0 项的阈值，值 7 设置 7 项的阈值

7.5.6.24. 0x098 I2C_TXFLR

默认值：0x00000000			I2C 发送 FIFO 水位（I2C Transport FIFO Level）
位域	类型	默认值	描述
31:4	–	–	–
3:0	R	0x0	TXFLR，显示发送 FIFO 数据量

7.5.6.25. 0x09C I2C_RXFLR

默认值：0x00000000			I2C 接收 FIFO 水位（I2C Receive FIFO Level）
位域	类型	默认值	描述
31:4	–	–	–
3:0	R	0x0	RXFLR，接收 FIFO 数据量

7.5.6.26. 0x0A0 I2C_SCL_STUCK_TIMEOUT

默认值：0xFFFFFFFF			I2C SCL 挂死超时（I2C_SCL_STUCK_TIMEOUT）
位域	类型	默认值	描述
31:0	R/W	0xffff_ffff	I2C_SCL_STUCK_TIMEOUT SCL 挂死时间设置。 如果 I2C 检测 SCL 处于低的时间超过此位域设定的时间，则产生相应的中断（SCL_STUCK_AT_LOW）。仅在 I2C 禁止时（I2C_ENABLE[0] 设为 0）可配置，其它时间写入无效。

7.5.6.27. 0x0A4 I2C_SDA_STUCK_TIMEOUT

默认值：0xFFFFFFFF			I2C SDA 挂死超时（I2C_SDA_STUCK_TIMEOUT）
位域	类型	默认值	描述
31:0	R/W	0xffff_ffff	I2C_SDA_STUCK_TIMEOUT SDA 挂死时间设置。 如果 I2C 检测 SDA 处于低的时间超过此位域设定的时间，并且 I2C_ENABLE[3]设为 1，则触发 SDA 挂死恢复机制

7.5.6.28. 0x0B0 I2C_FS_SPIKELEN

默认值：0x00000002			I2C 快速模式毛刺抑制时长（I2C_FS_SPIKELEN）
位域	类型	默认值	描述
31:0	R/W	0x2	I2C_FS_SPIKELEN I2C FS 模式毛刺抑制寄存器。 此位设置 SCL 与 SDA 线持续抑制毛刺的最长时间，以 I2C_clk 周期为单位。 此位只有在 I2C_ENABLE[0]设为 0 时可配置，其它时间配置无效。此位最小值为 1，硬件会阻止写入小于此值的配置，即写 0 则此位会被设为 1

7.5.6.29. 0x0FC VERSION

默认值：0x00000200			I2C 版本号（VERSION）
位域	类型	默认值	描述
31:0	RO	0x00000200	Version (模块版本号) 采用 BCD 码显示，V2.0

7.6. Controller Area Network (CAN)

CAN 是一种广泛应用于汽车控制系统和一般工业环境中的区域网络总线。作为一款多主机、多广播的通信协议，CAN 以其高可靠性和卓越的错误检测能力而著称。

7.6.1. 特性说明

- 支持 CAN2.0A 和 CAN2.0B 协议
- 支持 11 位标准格式标识符和 29 位扩展格式标识符
- 可编程通信速率最高可达 1 Mbps
- 支持多种操作模式：正常模式、只听模式、自测模式、休眠模式、复位模式
- 支持接收过滤器，支持两种过滤模式
- 64 bytes 缓冲器
- 支持错误检测与处理：错误计数、错误报警阈值可配置、错误捕获、仲裁丢失捕获

7.6.2. 原理框图

CAN 控制器原理如下：

- 位时序逻辑 BTL：负责同步位时序，确保数据传输的稳定性，以预先配置的波特率发送和接受报文。
- 位流处理器 BSP：负责对发送数据进行位填充、附加 CRC 等处理，并为位时序逻辑模块生成位流。负责对接收的位流进行去填充、验证 CRC 等处理。负责检测总线上的错误并将错误报告给错误管理逻辑。
- 接收过滤器：是一个可编程的报文过滤单元，允许控制器根据报文的标识符接收或拒绝该报文，可以过滤掉总线发送的与节点无关的消息，使节点效率提高。
- 缓冲 FIFO：大小为 64 bytes，负责存储通过接收过滤器的报文。FIFO 中的第一条报文将被映射到 13 bytes 的接收缓冲寄存器中可被 CPU 访问，直到该报文被清除，空间被释放。随后，接收缓冲寄存器将继续映射 FIFO 中的下一条报文。
- 错误管理逻辑 EML：负责更新发送错误计数 TEC 和接收错误计数 REC 的数值，记录错误类型、错误位置，更新控制器错误状态，确保 BSP 模块发送正确的错误标志。还负责记录报文仲裁丢失时的 bit 位置。

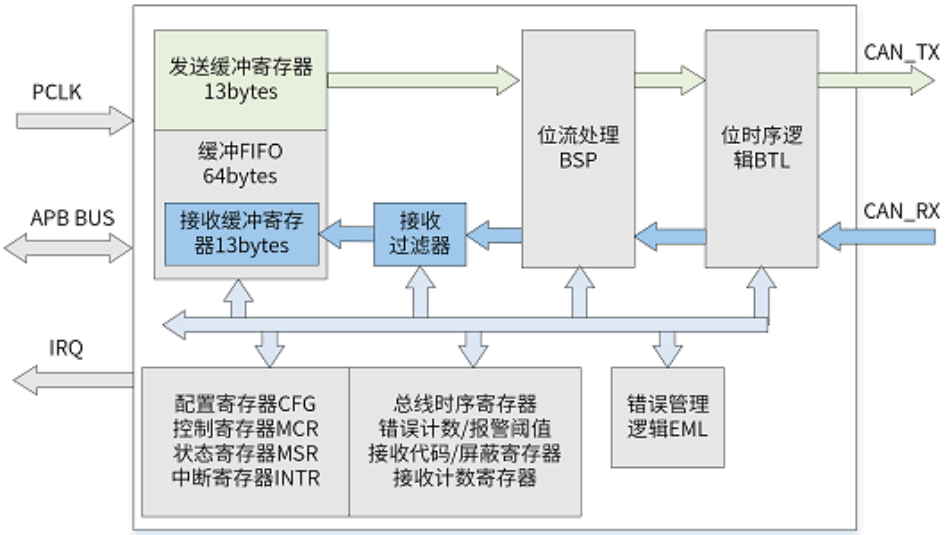
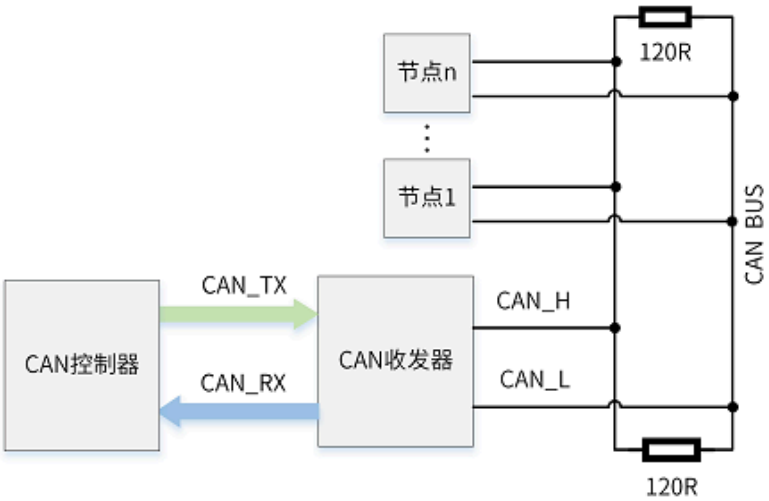


图 7–13 CAN 控制器原理框图

7.6.3. 典型应用

CAN 典型应用如图所示：



7.6.3.1. 运行模式

控制器包含五种运行模式，由模式寄存器 CAN_MODE 定义设置。

- 复位模式：Reset Mode。
  - 进入复位模式，控制器离线不参与总线任何活动，可修改 CAN 控制器的各种参数配置。
  - 退出复位模式，控制器需等待 11 个连续隐性位出现（等效于总线空闲），才能正常接收和发送报文。
- 正常模式： Normal Mode，CAN 控制器可以发送和接收包含错误信号在内的报文。
- 自测模式：Self-Test Mode，与正常模式相同，但在该模式下，CAN 控制器发送报文时，即使没有接收到应答，也不会产生应答错误。

  通常在控制器自测时使用。
- 只听模式： Listen Only Mode，CAN 控制器可以接收报文，但在 CAN 总线上保持完全被动。因此， CAN 控制器将无法发送任何报文、应答或错误信号，错误计数将保持冻结状态。

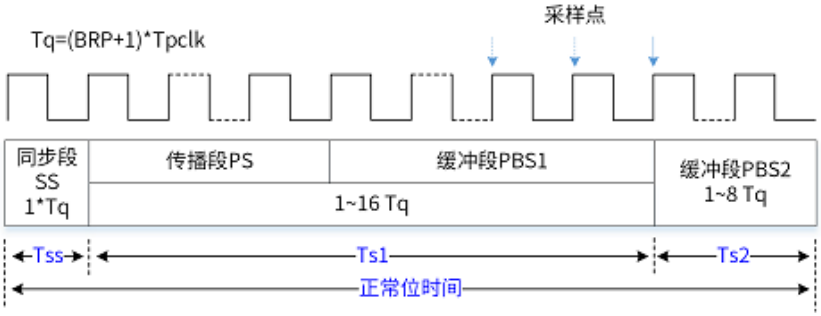
  通常用于 CAN 总线监控。
- 休眠模式：Sleep Mode，无总线活动及无中断发生，可进入低功耗休眠模式，该模式下时钟停止。

  SLEEP_MOD 为 0 或总线有信号输入时唤醒，产生唤醒中断。

7.6.3.2. 位时序

为了实现位同步，每个正常位时间划分为几个不重叠时间的片段，包括同步段 SS，传播段 PS，相位缓冲段 PBS1，相位缓冲段 PBS2，一个完整的位由 8~25 个时间定额 T_q 组成。

- 如图所示，将传播段 PS 和相位缓冲段 PBS1 合并形成新的时间段 T_{S1}。
- 在复位模式下，配置总线时序 0 寄存器 CAN_BTR0 和总线时序 1 寄存器 CAN_BTR1 中的 BRP、SJW、T_{S1}、T_{S2}、SAM 等参数，决定 CAN 的波特率。
- SJW 表示 CAN 波特率的容许偏差量，其大小与波特率值没有直接关系，SJW 值越大则总线获得更宽波特率容忍度。



位时序各参数计算关系如下：

- 时间定额 $T_q = 2 \times (BRP[5:0] + 1) \times T_{pclk}$，其中 T_{pclk} 为 APB 时钟（通常为 24 MHz）周期。
- 位时间 $T_{bit} = [1 + (T_{S1} + 1) + (T_{S2} + 1)] \times Tq$。
- 波特率 $Baud = 1 / T_{bit}$。
- 采样点 $SP = [1 + (T_{S1} + 1)]/ [1 + (T_{S1} + 1) + (T_{S2} + 1)]$。

位时序各参数计算

假设单次采样的参数值如下：

SAM = 0，且 SJW = 2，TS1 = 8，TS2 = 1，BRP = 0，则波特率为 24 MHz / [2 × (0 + 1) × (1 + 9 + 2)] = 1 Mbps，采样点为 (1 + 9) / (1 + 9 + 2) = 83%。

建议选取规则：

- $T_{S2} \geqslant SJW$
- BRP 尽量小、SJW 尽量大。
- 采样点选取范围 75%~85% 之间。
- 尽量采用单次采样。

7.6.3.3. 通信报文

CAN 通信报文类型包含数据帧、远程帧、错误帧、过载帧和间隔帧。

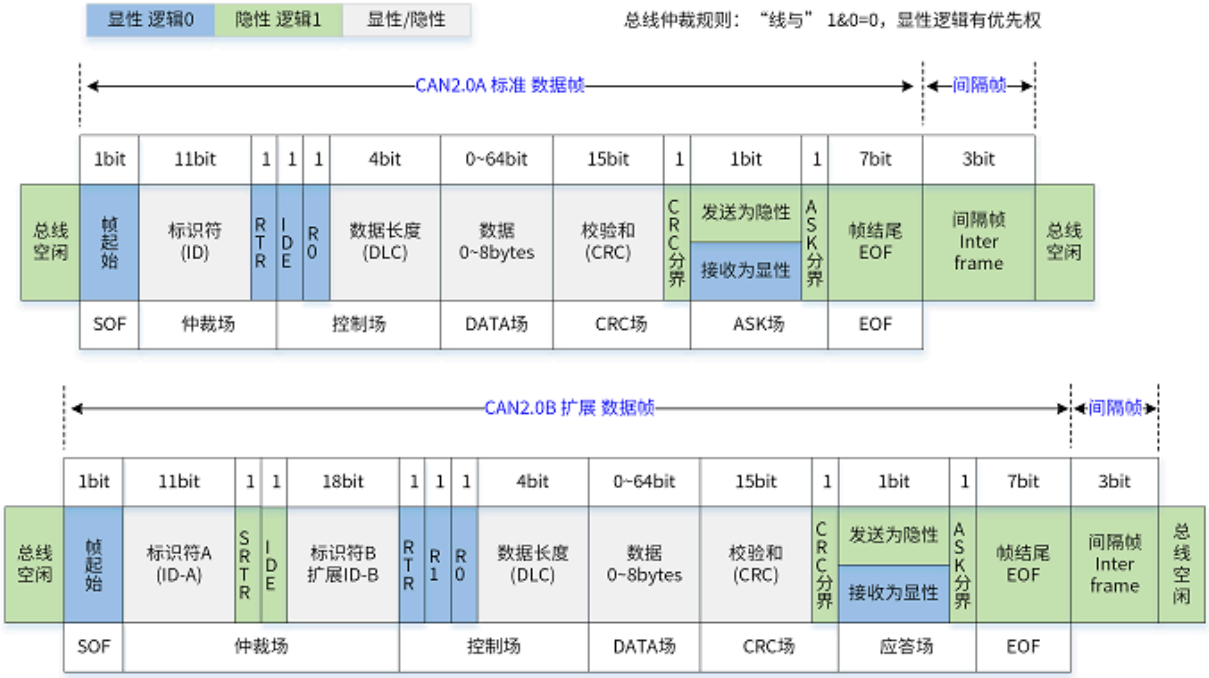


图 7–14 CAN 标准格式和扩展格式数据帧

图 7–15 CAN 标准格式和扩展格式远程帧

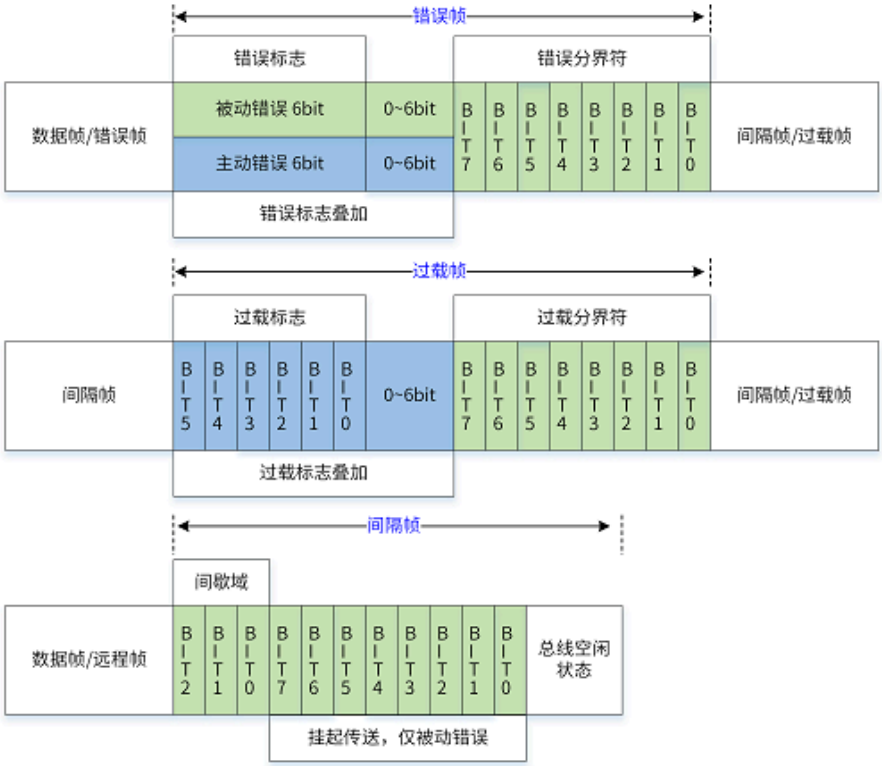


图 7–16 CAN 错误帧、过载帧、间隔帧

表 7–29 数据帧/远程帧描述

数据/远程帧	描述
SOF	帧起始，1 bit，用于同步总线上节点的单个显性位。
ID–A	标识符 A，11 bits，对应标准格式中的 11 位标识符（ID10~ ID0），或扩展格式中 29 位标识符的前 11–bit（ID28~ ID18）。
RTR	远程发送请求位，1 bit，显示当前报文是数据帧（显性 0）还是远程帧（隐性 1）。 当某个数据帧和远程帧有相同标识符时，数据帧始终优先于远程帧仲裁。
SRTR	代替远程发送请求位，1 bit，在扩展格式中以替代标准格式相同位置的 RTR 位。
IDE	标识符扩展位，1 bit，显示当前报文是标准格式（显性 0）还是扩展格式（隐性 1）。 当某标准帧和扩展帧有相同基标识符时，标准帧将始终优先于扩展帧仲裁。
ID–B	标识符 B，18 bits，扩展格式中 29 位标识符的剩余 18 bit（ID17 ~ ID0）。
R1	保留位，始终是显性位。
R0	保留位，始终是显性位。
DLC	数据长度代码， 4 bits，包含 0 ~ 8 中任一数值。

表 7–29 数据帧/远程帧描述 (续)

数据/远程帧	描述
	• 数据帧用于表示包含的数据字节数量。 • 远程帧用于表示从其它节点请求的数据字节数量。
数据字节	表示数据帧的数据负载量，该字节数量应与 DLC 的值匹配。首先发送数据字节 0，各数据字节优先发送最高有效位。
CRC	CRC 序列是一个 15–bit 的循环冗余校验码。具体实现方式为： 1. 被除多项式的系数由 SOF、仲裁场、控制场、数据场（若存在）及 15 位（最低系数）0 组成的未经填充的位流给定，而生成多项式为 $x^{15}+x^{14}+x^{10}+x^{8}+x^{7}+x^{4}+x^{3}+x^{0}$。 2. 被除多项式被生成多项式除（按模 2 计算），余数即为将要发送到总线上的 CRC 序列。 3. 发送节点和接收节点均采用相同方法生成 CRC 校验码，并与发送节点送出的 CRC 校验码进行比较： • 若出错，控制器依据仲裁原则及受损报文优先原则对已损坏报文自动重发。
CRC 分界符	分界符，1 bit，隐性位。
ASK 槽	应答槽，1 bit，用于接收节点，表示是否已成功接收数据帧或远程帧。发送节点将在应答槽中发送一个隐性位，若接收到的帧无错误，则接收节点用一个显性位以示应答。
ASK 分界符	分界符，1 bit，隐性位。
EOF	帧结束，7 bits，隐性位，标志着数据帧或远程帧的结束。

7.6.3.4. 中断管理

CAN 控制器提供八种中断，具体如下：

- 总线错误中断 ERRB_INT

  控制器在总线上检测到错误时触发此中断。错误类型和错误位置都将被捕获记录在状态寄存器 CAN_STAT 的 ERR_TYPE 和 ERR_CODE 中。状态被清除之前（通过 CPU 的读取），将不会再记录新的总线错误信息。

- 仲裁丢失中断 ARBLOST_INT

  控制器发送报文并且丢失仲裁时触发此中断。丢失仲裁位置将被捕获记录在状态寄存器 CAN_STAT 的 ARBLOST_CAP 中，状态被清除之前（通过 CPU 读取），将不会再记录新的仲裁丢失位置。

- 被动错误中断
- 唤醒中断
- 数据溢出中断
- 错误报警中断 ERRW_INT

状态寄存器 CAN_STAT 中 ERR_STAT 或 BUS_STAT 的位值发生改变（由 0 变 1 或由 1 变 0）都会触发此中断。触发时状态值可分以下几种：

- ERR_STAT = 0 与 BUS_STAT = 0：
  - 如果控制器处于主动错误状态，则表示 TEC 和 REC 的值都返回 ERRWT 所设阈值之下。
  - 如果控制器此前处于总线恢复状态，则表示此时总线恢复已成功完成。
- ERR_STAT = 1 与 BUS_STAT = 0：表示 TEC 或 REC 数值已超过 ERRWT 所设阈值。
- ERR_STAT = 1 与 BUS_STAT = 1：表示控制器已进入离线状态（TEC ≥ 255）。
- ERR_STAT = 0 与 BUS_STAT = 1：表示总线恢复期间，控制器 TEC 数值已低于 ERRWT 所设阈值。

- 发送中断 TX_INT

  当发送缓冲器空闲，将其它报文加载到缓冲器中等待发送时，都会触发此中断。

- 接收中断 RX_INT

  当接收 FIFO 中有待读取报文（接收数据计数寄存器 CAN_RXC 中 RXC > 0）时触发此中断。计数的报文数量包括有效报文和溢出报文。直到通过 RXB_REL 指令位清除所有挂起接收报文后，接收中断才会失效。

中断可由中断寄存器 CAN_INTR 定义，设置中断使能寄存器 CAN_INTEN 中相应的使能位，可以单独允许和禁用每个中断源。中断的工作原理和规则如下：

- 当有一个或多个中断触发，控制器 IRQ 有效，当所有中断位都被清除，控制器 IRQ 失效。
- 中断寄存器被读取后，除接收中断外，其中的中断位将自动清除。
- 接收中断直到通过 RXB_REL 指令位清除所有接收报文后，才能被清除，否则会影响接收缓冲状态 RXB_STAT。

7.6.3.5. 数据缓冲器

数据缓冲寄存器组是缓存发送和接收帧的存储空间，其访问地址范围为 0x40~0x70，读写共用相同的地址空间。

> **注：**
>
> 复位状态下不可访问。

表 7–30 数据缓冲寄存器组

寄存器	偏移地址	标准格式内容	扩展格式内容
CAN_BUF0	0x40	TX/RX 帧信息	TX/RX 帧信息
CAN_BUF1	0x44	TX/RX 标识符 1	TX/RX 标识符 1
CAN_BUF2	0x48	TX/RX 标识符 2	TX/RX 标识符 2
CAN_BUF3	0x4C	TX/RX 数据 1	TX/RX 标识符 3
CAN_BUF4	0x50	TX/RX 数据 2	TX/RX 标识符 4

表 7–30 数据缓冲寄存器组 (续)

寄存器	偏移地址	标准格式内容	扩展格式内容
CAN_BUF5	0x54	TX/RX 数据 3	TX/RX 数据 1
CAN_BUF6	0x58	TX/RX 数据 4	TX/RX 数据 2
CAN_BUF7	0x5C	TX/RX 数据 5	TX/RX 数据 3
CAN_BUF8	0x60	TX/RX 数据 6	TX/RX 数据 4
CAN_BUF9	0x64	TX/RX 数据 7	TX/RX 数据 5
CAN_BUF10	0x68	TX/RX 数据 8	TX/RX 数据 6
CAN_BUF11	0x6C	保留	TX/RX 数据 7
CAN_BUF12	0x70	保留	TX/RX 数据 8

- 发送缓冲寄存器：CPU 对数据缓冲寄存器组的写操作访问，配置待发送报文，指定报文的帧类型、帧格式、帧标识符和数据。CPU 通过控制寄存器 CAN_MCR 配置发送报文模式：
  - 正常报文发送：TX_REQ 置 1。
  - 自发自收：SELF_REQ 置 1。
  - 单次发送：TX_REQ 和 ABORT_REQ 同时置 1。
- 接收缓冲寄存器：CPU 对数据缓冲寄存器组的读取操作访问，接收缓冲寄存器映射到缓冲 FIFO 中的第一条报文，获取第一条报文的帧类型、帧格式、帧标识符和数据。

  读取完接收缓冲寄存器中的报文后，CPU 通过控制寄存器 CAN_MCR 中的 RXB_REL 置 1 来释放接收缓冲寄存器。

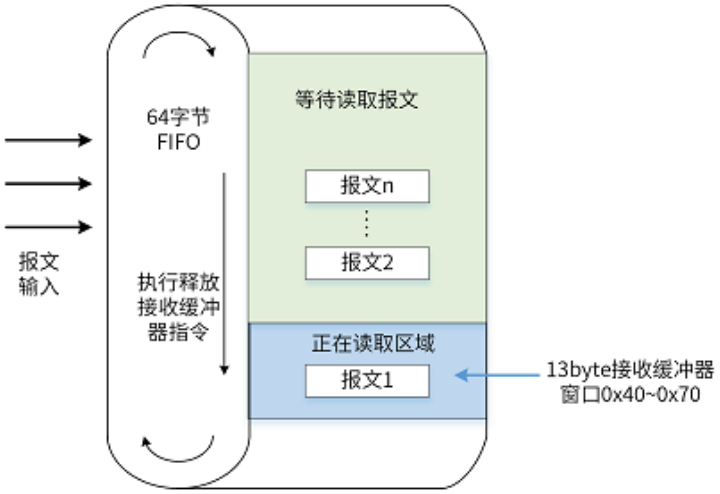
  若接收 FIFO 中仍有待处理的报文，按照接收报文的先后次序，将最早接收到的报文映射到接收缓冲寄存器中。
- 缓冲 FIFO： 是一个 64 字节大小的内部缓冲器，以先进先出原则存储接收到的报文。
  1. 接收缓冲寄存器为 FIFO 中可访问窗口，偏移地址 0x40~0x70，将被映射到 FIFO 中的第一条报文。

     一条报文可在 FIFO 中占 3 ~ 13 字节空间，其中字节序与接收缓冲寄存器相同。
  2. 当控制器接收到一条报文时，接收数据计数器 RXC 增加 1，最大值为 64。

     如果 FIFO 中有足够空间，报文内容将被写入到 FIFO 中。
  3. CPU 读取接收缓冲寄存器中的报文后，通过将 RXB_REL 置 1，释放 FIFO 中第一条报文所占的空间，RXC 将减小 1。
  4. 接收缓冲寄存器接着映射 FIFO 中的下一条报文。

- FIFO 数据溢出：任何情况下，FIFO 中可以存储的报文数量取决于各条报文的长度。当 FIFO 中没有足够空间完整地存储新报文时，控制器会产生数据溢出，通过状态位或数据溢出中断反馈给 CPU。

  FIFO 在内部将溢出报文标记为无效，后续接收到的溢出报文仍然将增加 RXC 到最大值 64。为了读取接收 FIFO 中的所有有效报文并清除所有溢出报文，应重复调用 RXB_REL，直到 RXC 为 0。



7.6.3.6. 接收过滤器

接收过滤器，是一个可编程的报文过滤单元，允许控制器根据报文的标识符、帧类型、第一个数据字节接收或拒绝该报文。接收过滤器由四个接收代码寄存器 CAN_RXCODE 和四个接收屏蔽寄存器 CAN_RXMASK 定义组成。

图 7–17 : 接收过滤器原理中，每条报文中的位必须匹配 RXCODE 值所指定的模式或者被 RXMASK 值屏蔽，才能使该报文通过过滤并存储到接收 FIFO 中。

接收代码寄存器、接收屏蔽寄存器，分别与缓冲 0~3 寄存器（偏移地址 0x40~0x4C）、缓冲 4~7 寄存器（偏移地址 0x50~5C）地址空间相同，因此只有控制器在复位模式时，才允许访问这两个寄存器。

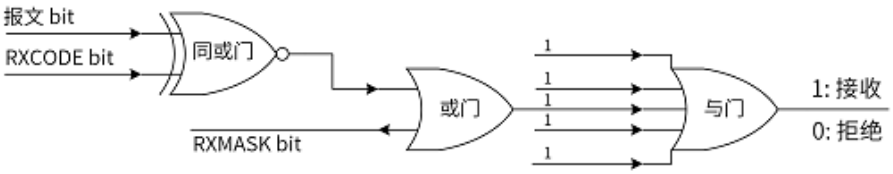


图 7–17 接收过滤器原理

- 单过滤模式：在模式寄存器 CAN_MODE 中将 FILTER_SEL 设置为 1 ，启动单过滤模式。

  根据 RXCODE/ RXMASK 的值定义单个过滤器，其定义及过滤报文格式如图所示。

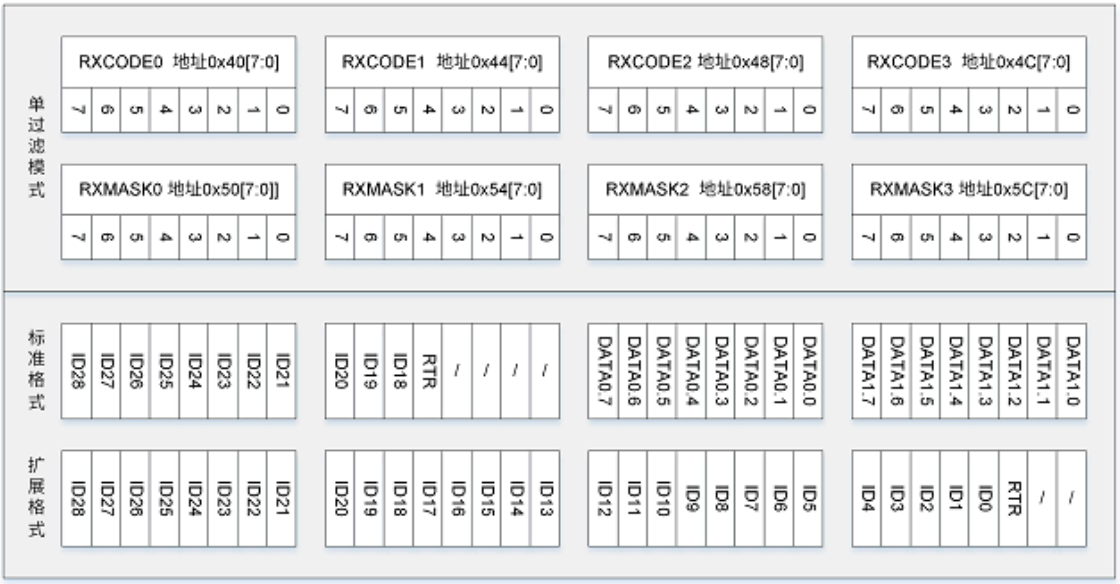


图 7–18 单过滤模式下单个过滤器定义及其可过滤报文格式

- 双过滤模式：在模式寄存器 CAN_MODE 中将 FILTER_SEL 设置为 0， 启动双过滤模式。

  根据 RXCODE/ RXMASK 的值将定义两个过滤器，分别为过滤器 0 和过滤器 1。两个过滤器的标准格式 (SFF) 和扩展格式 (EFF) 下的过滤报文定义分别如下图所示。双过滤模式下，如果报文通过至少一个滤波器，则表示报文成功通过过滤。

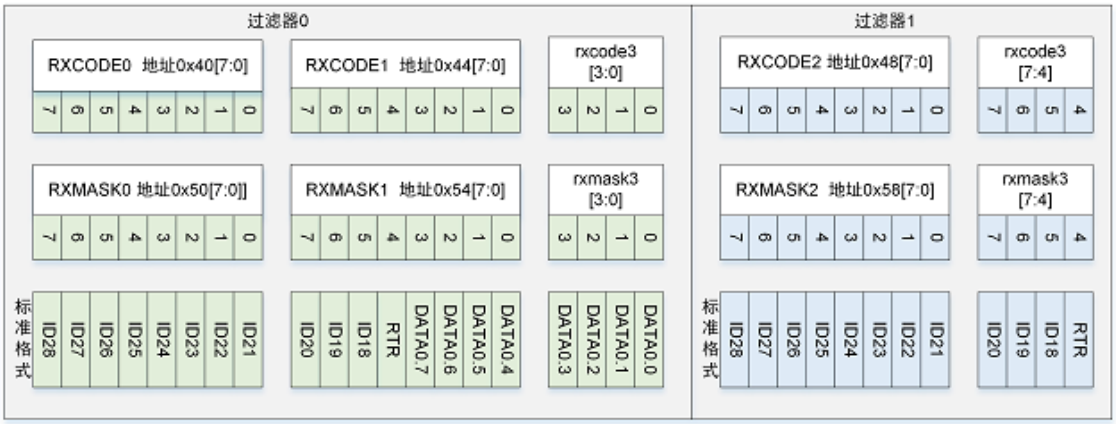


图 7–19 双过滤模式下 SFF 标准格式过滤报文定义

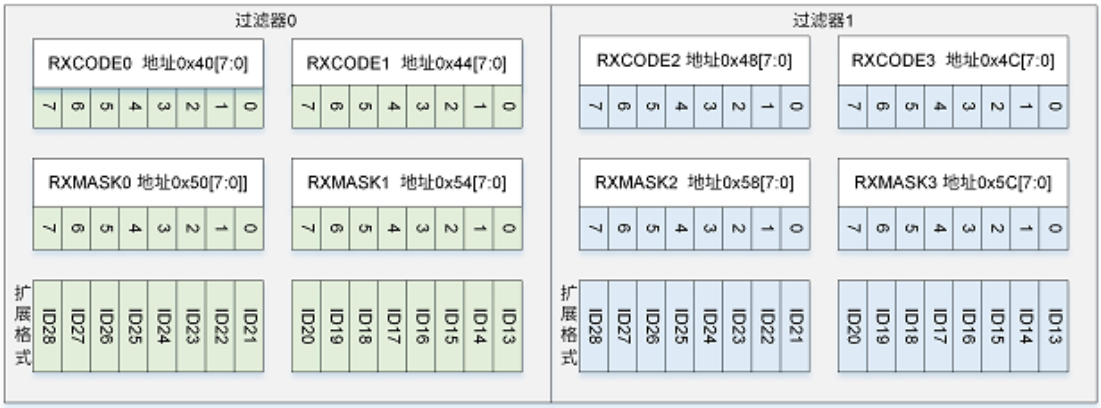


图 7–20 双过滤模式下 EFF 扩展格式过滤报文定义

7.6.3.7. 错误管理

错误计数包含发送错误计数 TEC 和接收错误计数 REC。错误计数决定 CAN 控制器当前的错误状态，例包括主动错误、被动错误或离线。

除了错误状态外，控制器还提供错误报警阈值 ERRWT 功能。ERRWT 功能可在控制器进入被动错误状态前，提醒用户当前发生的严重总线错误。

控制器当前的错误状态通过 TEC、REC、ERR_STAT 、BUS _STAT 和 ERRWT 体现，数值和状态位的变化可以触发中断，从而提醒用户当前的错误状态变化，详情可参考 中断管理。

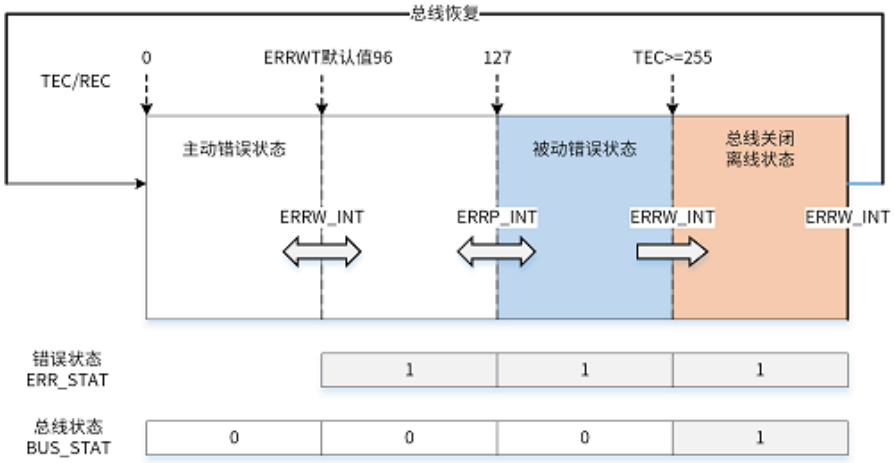


图 7–21 错误状态变化及中断关系

图 7–21 : 错误状态变化及中断关系表示了错误状态、计数数值、状态位以及相关中断之间的关系。

- 作为报警功能，错误报警阈值 ERRWT 可提示当前发生的总线错误，在控制器进入被动错误状态之前触发。在复位模式下，ERRWT 数值可在错误报警阈值寄存器 CAN_ERRWT 中进行配置。
  - 当 TEC 和 REC 中任意数值大于等于 ERRWT 时，ERR_STAT 被置 1。
  - 当 TEC 和 REC 数值都小于 ERRWT 时， ERR_STAT 被复位 0。
  - 只要 ERR_STAT 或 BUS_STAT 位值发生变化，便会触发错误报警中断 ERRW_INT。
- 当 TEC 或 REC 数值大于 127 时，节点进入被动错误状态。
- 当 TEC 和 REC 数值都小于等于 127 时，节点重新变为主动错误状态。

  节点在主动错误和被动错误状态之间切换时，都将触发被动错误中断 ERRP_INT。
- 当 TEC 数值大于 255 时，节点进入总线关闭离线状态。

  此时，控制器将 REC 数值置为 0，TEC 数值置为 127，BUS_STAT 位置 1，产生错误报警中断 ERRW_INT，控制器进入复位模式。
- 为了返回主动错误状态，必须进行离线恢复。
  1. 退出复位模式，进入正常操作模式。
  2. 要求节点在总线上检测到 128 次 11 个连续隐性位。

     每一次检测到 11 个连续隐性位时，TEC 数值都将减 1。
  3. 当离线恢复完成后（TEC 数值从 127 减小到 0），BUS_STAT 位复位为 0，从而触发错误报警中断 ERRW_INT。

7.6.3.8. 错误捕获

错误捕获功能允许控制器以错误代码的形式，记录总线错误类型和错误位置。

当检测到一个总线错误时，触发总线错误中断 ERRB_INT，状态寄存器 CAN_STAT 的 ERR_CODE 会记录相应的错误代码。在当前错误代码被读取前，后续的总线错误中断触发时，将不会再记录。

7.6.3.9. 仲裁丢失捕获

仲裁丢失捕捉功能允许控制器记录仲裁丢失的 bit 位置。当控制器丢失仲裁时，所丢失 bit 位置将被记录在状态寄存器 CAN_STAT 的 ARBLOST_CAP 中，同时触发仲裁丢失中断 ARBLOST_INT。在当前仲裁丢失捕获 bit 位置被读取前，后续的仲裁丢失中断触发时，将不会再记录。

7.6.4. 寄存器列表

表 7–31 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	CAN_MODE	CAN 模式寄存器	0x000 CAN_MODE
0x004	CAN_MCR	CAN 控制寄存器	0x004 CAN_MCR
0x008	CAN_STAT	CAN 状态寄存器	0x008 CAN_STAT
0x00C	CAN_INTR	CAN 中断寄存器	0x00C CAN_INTR
0x010	CAN_INTEN	CAN 中断使能寄存器	0x010 CAN_INTEN
0x018	CAN_BTR0	CAN 总线时序 0 寄存器	0x018 CAN_BTR0
0x01C	CAN_BTR1	CAN 总线时序 1 寄存器	0x01C CAN_BTR1
0x02C	CAN_ARBLOST	CAN 仲裁丢失捕获寄存器	0x02C CAN_ARBLOST
0x030	CAN_ERRCODE	CAN 错误代码捕获寄存器	0x030 CAN_ERRCODE
0x034	CAN_ERRWT	CAN 错误报警阈值寄存器	0x034 CAN_ERRWT
0x038	CAN_RXERR	CAN 接收错误计数寄存器	0x038 CAN_RXERR
0x03C	CAN_TXERR	CAN 发送错误计数寄存器	0x03C CAN_TXERR
0x040	CAN_BUF0	CAN 缓冲 0 寄存器	0x040 CAN_BUF0
0x044	CAN_BUF1	CAN 缓冲 1 寄存器	0x044 CAN_BUF1
0x048	CAN_BUF2	CAN 缓冲 2 寄存器	0x048 CAN_BUF2

表 7–31 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x04C	CAN_BUF3	CAN 缓冲 3 寄存器	0x04C CAN_BUF3
0x050	CAN_BUF4	CAN 缓冲 4 寄存器	0x050 CAN_BUF4
0x054	CAN_BUF5	CAN 缓冲 5 寄存器	0x054 CAN_BUF5
0x058	CAN_BUF6	CAN 缓冲 6 寄存器	0x058 CAN_BUF6
0x05C	CAN_BUF7	CAN 缓冲 7 寄存器	0x05C CAN_BUF7
0x060	CAN_BUF8	CAN 缓冲 8 寄存器	0x060 CAN_BUF8
0x064	CAN_BUF9	CAN 缓冲 9 寄存器	0x064 CAN_BUF9
0x068	CAN_BUF10	CAN 缓冲 10 寄存器	0x068 CAN_BUF10
0x06C	CAN_BUF11	CAN 缓冲 11 寄存器	0x06C CAN_BUF11
0x070	CAN_BUF12	CAN 缓冲 12 寄存器	0x070 CAN_BUF12
0x040	CAN_RXCODE0	CAN 接收代码 0 寄存器	0x040 CAN_RXCODE0
0x044	CAN_RXCODE1	CAN 接收代码 1 寄存器	0x044 CAN_RXCODE1
0x048	CAN_RXCODE2	CAN 接收代码 2 寄存器	0x048 CAN_RXCODE2
0x04C	CAN_RXCODE3	CAN 接收代码 3 寄存器	0x04C CAN_RXCODE3
0x050	CAN_RXMASK0	CAN 接收屏蔽 0 寄存器	0x050 CAN_RXMASK0
0x054	CAN_RXMASK1	CAN 接收屏蔽 1 寄存器	0x054 CAN_RXMASK1
0x058	CAN_RXMASK2	CAN 接收屏蔽 2 寄存器	0x058 CAN_RXMASK2
0x05C	CAN_RXMASK3	CAN 接收屏蔽 3 寄存器	0x05C CAN_RXMASK3
0x074	CAN_RXC	CAN 接收数据计数寄存器	0x074 CAN_RXC
0x078	CAN_RSADDR	CAN 接收缓冲起始地址寄存器	0x078 CAN_RSADDR
0x080	CAN_RXFIFO	CAN 接收 FIFO 寄存器	0x080 CAN_RXFIFO
0x180	CAN_TXB_RO	CAN 发送缓冲只读寄存器	0x180 CAN_TXB_RO

表 7–31 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x1C0	CAN_TX_DMA	CAN TX DMA 控制寄存器	0x1C0 CAN_TX_DMA
0x1C4	CAN_TX_DMA_THR	CAN TX DMA 水位寄存器	0x1C4 CAN_TX_DMA_THR
0x1C8	CAN_RX_DMA	CAN RX DMA 控制寄存器	0x1C8 CAN_RX_DMA
0x1CC	CAN_RX_DMA_THR	CAN RX DMA 水位寄存器	0x1CC CAN_RX_DMA_THR
0xFFC	CAN_VERSION	CAN 版本寄存器	0xFFC CAN_VERSION

7.6.5. 寄存器描述

7.6.5.1. 0x000 CAN_MODE

默认值：0x00000001			CAN 模式寄存器 (CAN_MODE)
位域	类型	默认值	描述
31:5	–	–	–
4	R/W	0x0	SLEEP_MOD 休眠模式 (Sleep Mode) • 0x0：正常操作，控制器唤醒 • 0x1：休眠模式，CAN 无总线活动和无中断发生 注： 仅复位状态可写
3	R/W	0x0	FILTER_MOD 接收过滤器模式 (Acceptance Filter Mode) • 0x0：双过滤模式，使用两个过滤器 • 0x1：单过滤模式，使用一个 4 字节过滤器
2	R/W	0x0	SELFTEST_MOD，。 自测模式 (Self Test Mode) • 0x0：正常操作，发送成功需有应答 • 0x1：自测模式，在 SELF_REQ 自接收请求指令下，无需其它节点参与也能完成整个节点测试，发送成功无需应答

默认值：0x00000001			CAN 模式寄存器 (CAN_MODE)
位域	类型	默认值	描述
1	R/W	0x0	LISTEN_MOD。 只听模式 (Listen Only Mode) • 0x0：正常操作，错误计数停止于当前值 • 0x1：只听模式，只接收总线上数据，不产生应答信号，也不更新接收错误计数
0	R/W	0x1	RST_MOD 复位模式 (Reset Mode) • 0x0：正常操作,控制器返回正常操作模式 • 0x1：复位模式,发送或接收中止，控制器处于可配置状态

7.6.5.2. 0x004 CAN_MCR

默认值：0x00000000			CAN 控制寄存器 (CAN_MCR)
位域	类型	默认值	描述
31:6	–	–	–
4	WO	0x0	SELF_REQ 自接收请求 (Self Reception Request) • 0x0：无动作 • 0x1：允许发送数据的同时接收总线上的数据
3	WO	0x0	CLR_OVF_FLAG 清除溢出标志 (Clear Overflow Flag) • 0x0：无动作 • 0x1：清除数据溢出标志位
2	WO	0x0	RXB_REL 释放接收缓冲器 (Release Receive Buffer) • 0x0：无动作 • 0x1：释放，缓冲器中存放信息的内存空间被释放
1	WO	0x0	ABORT_REQ 中止发送 (Abort Transmission) • 0x0：无动作 • 0x1：中止当前未开始的发送任务

默认值：0x00000000			CAN 控制寄存器 (CAN_MCR)
位域	类型	默认值	描述
0	WO	0x0	TX_REQ 发送请求 (Transmission Request) • 0x0：无动作 • 0x1：信息被发送

7.6.5.3. 0x008 CAN_STAT

默认值：0x0000003C			CAN 状态寄存器 (CAN_STAT)
位域	类型	默认值	描述
31:8	–	–	–
7	RO	0x0	BUS_STAT 总线状态 (Bus Status) • 0x0：正常状态,控制器参与总线活动 • 0x1：Bus Off 离线状态,控制器不参与总线活动。
6	RO	0x0	ERR_STAT 错误状态 (Error Status) • 0x0：所有错误计数器低于警报限制值 • 0x1：至少一个错误计数器高于警报限制值
5	RO	0x1	TX_STAT 发送状态 (Transmit Status) • 0x0：无任何发送 • 0x1：正在发送数据
4	RO	0x1	RX_STAT 接收状态 (Receive Status) • 0x0：无任何接收 • 0x1：正在接收数据
3	RO	0x1	TXC_STAT 发送完成状态 (Transmission Complete Status) • 0x0：没有完成发送 • 0x1：成功完成发送

默认值：0x0000003C			CAN 状态寄存器 (CAN_STAT)
位域	类型	默认值	描述
2	RO	0x1	TXB_STAT 发送缓冲状态 (Transmit Buffer Status) • 0x0：发送缓冲不可用，发送缓冲已锁定，不可访问 • 0x1：发送缓冲可用，发送缓冲已释放，可写入数据
1	RO	0x0	OVF_FLAG 溢出状态标志 (Overflow Flag) • 0x0：数据没有溢出 • 0x1：数据溢出
0	RO	0x0	RXB_STAT 接收缓冲状态 (Receive Buffer Status) • 0x0：接收缓冲为空，无数据可读 • 0x1：接收缓冲不为空，有数据可读

7.6.5.4. 0x00C CAN_INTR

默认值：0x00000000			CAN 中断寄存器 (CAN_INTR)
位域	类型	默认值	描述
31:8	–	–	–
7	R/RC	0x0	ERRB_INT 总线错误中断 (Bus Error Interrupt) 节点检测到总线上发生了错误 读中断寄存器 CAN_INTR 任意位，清除该中断
6	R/RC	0x0	ARBLOST_INT 仲裁丢失中断 (Arbitration Lost Interrupt) 发送节点仲裁丢失 读中断寄存器 CAN_INTR 任意位，清除该中断
5	R/RC	0x0	ERRP_INT 被动错误中断 (Error Passive Interrupt) 节点在主动错误与被动错误状态间发生了切换，触发 读中断寄存器 CAN_INTR 任意位，清除该中断

默认值：0x00000000			CAN 中断寄存器 (CAN_INTR)
位域	类型	默认值	描述
4	RC	0x0	WAKEUP 唤醒中断 (Wake-Up Interrupt) 读中断寄存器 CAN_INTR 任意位，清除该中断
3	R/RC	0x0	OVF_INT 溢出中断 (Overflow Interrupt) 当前接收缓冲中存储的报文为溢出失效报文，触发 读中断寄存器 CAN_INTR 任意位，清除该中断
2	R/RC	0x0	ERRW_INT 错误报警中断 (Error Warning Interrupt) 错误状态和总线状态发生变化，触发 读中断寄存器 CAN_INTR 任意位，清除该中断
1	R/RC	0x0	TX_INT 发送中断 (Transmit Interrupt) 数据发送任务结束，可以执行新的数据发送任务，触发 读中断寄存器 CAN_INTR 任意位，清除该中断
0	R	0x0	RX_INT 接收中断 (Receive Interrupt) RXB_STAT=1 接收缓冲不为空时触发，有数据待处理 对控制寄存器 CAN_MCR[2] 写 1，清除该中断

7.6.5.5. 0x010 CAN_INTEN

默认值：0x00000000			CAN 中断使能寄存器 (CAN_INTEN)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	ERRB_EN 总线错误中断使能 (Bus Error Interrupt Enable)
6	R/W	0x0	ARBLOST_EN 仲裁丢失中断使能 (Arbitration Lost Interrupt Enable)
5	R/W	0x0	ERRP_EN 被动错误中断使能 (Error Passive Interrupt Enable)

默认值：0x00000000			CAN 中断使能寄存器 (CAN_INTEN)
位域	类型	默认值	描述
4	R/W	0x0	WAKEUP_EN 唤醒中断使能 (Wake-Up Interrupt Enable)
3	R/W	0x0	OVF_EN 溢出中断使能 (Overflow Interrupt Enable)
2	R/W	0x0	ERRW_EN 错误报警中断使能 (Error Warning Interrupt Enable)
1	R/W	0x0	TXI_EN 发送中断使能 (Transmit Interrupt Enable)
0	R/W	0x0	RXI_EN 接收中断使能 (Receive Interrupt Enable)

7.6.5.6. 0x018 CAN_BTR0

默认值：0x00000000			CAN 总线时序 0 寄存器 (CAN_BTR0)
位域	类型	默认值	描述
30:8	–	–	–
7:6	R/W	0x0	SJW 重新同步跳转宽度 (Synchronization Jump Width) • 0x0：1Tq • 0x1：2Tq • 0x2：3Tq • 0x3：4Tq **注：** 仅复位模式可写
5:0	R/W	0x0	BRP 波特率预分频 (Baud Rate Prescaler)，$T_q = 2 \times (BRP[5:0] + 1) \times T_{pclk}$ **注：** 仅复位模式可写

7.6.5.7. 0x01C CAN_BTR1

默认值：0x00000000			CAN 总线时序 1 寄存器 (CAN_BTR1)
位域	类型	默认值	描述
30:24	–	–	–
7	R/W	0x0	SAM 采样次数 (Sample Time) • 0x0：采样 1 次，用于高速总线 • 0x1：采样 3 次，用于低和中速总线。 **注：** 仅复位模式可写
6:4	R/W	0x0	TS2 缓冲时间段 2 的宽度 [0x0~0x7]：1~8T_q **注：** 仅复位模式可写
3:0	R/W	0x0	TS1 缓冲时间段 1 的宽度 [0x0~0xF]：1~16T_q **注：** 仅复位模式可写

7.6.5.8. 0x02C CAN_ARBLOST

默认值：0x00000000			CAN 仲裁丢失捕获寄存器 (CAN_ARBLOST)
位域	类型	默认值	描述
31:5	–	–	–
4:0	RO	0x00	ARBLOST_CAP 仲裁丢失捕获 (Arbitration Lost Capture)，记录丢失 bit 的位置 • 0x00~0x0A：SFF ID10~ID0 或者 EFF ID28~ID18 仲裁丢失 • 0x0B：SRTR 仲裁丢失

默认值：0x00000000			CAN 仲裁丢失捕获寄存器 (CAN_ARBLOST)
位域	类型	默认值	描述
			• 0x0C：IDE 仲裁丢失 • 0x0D~0x1E：EFF ID17~ID0 仲裁丢失 • 0x1F：RTR 仲裁丢失

7.6.5.9. 0x030 CAN_ERRCODE

默认值：0x00000000			CAN 错误代码捕获寄存器 (CAN_ERRCODE)
位域	类型	默认值	描述
31:8	–	–	–
7:6	RO	0x0	ERR_TYPE 错误类型 (Error Type) • 0x0：位错误 • 0x1：格式错误 • 0x2：填充错误 • 0x3：其它类型错误
5	RO	0x0	ERR_DIR (Error Direction) 错误发生时数据传输方向 • 0x0：错误发生于发送 • 0x1：错误发生于接收
4:0	RO	0x00	ERR_CODE 错误发生段代码 (Error Segment Code) • 0x03：SOF，帧起始 • 0x02：ID28~ID21，标识符 • 0x06：ID20~ID18，标识符 • 0x04：SRTR，代替远程发送请求位 • 0x05：IDE，标识符扩展位 • 0x07：ID17~ID13，标识符 • 0x0F：ID12~ID5，标识符 • 0x0E：ID4~ID0，标识符 • 0x0C：RTR，远程发送请求位 • 0x0D：R1，保留位

默认值：0x00000000			CAN 错误代码捕获寄存器 (CAN_ERRCODE)
位域	类型	默认值	描述
			• 0x09：R0，保留位 • 0x0B：DLC，数据长度代码 • 0x0A：数据场 • 0x08：CRC，CRC 序列 • 0x18：CRC 分界符 • 0x19：ASK，应答槽 • 0x1B：ASK 分界符 • 0x1A：EOF，帧结尾 • 0x12：间隔帧 • 0x11：主动错误标志 • 0x16：被动错误标志 • 0x13：兼容显性位 • 0x17：错误分界符 • 0x1C：过载标志

7.6.5.10. 0x034 CAN_ERRWT

默认值：0x00000060			CAN 错误报警阈值寄存器 (CAN_ERRWT)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x60	ERRWT 错误报警阈值 (Error Warning Threshold) **注：** 正常模式为只读状态，复位模式下可读写。任一错误计数值超过该阈值，触发错误报警中断（使能信号有效情况下）

7.6.5.11. 0x038 CAN_RXERR

默认值：0x00000000			CAN 接收错误计数寄存器 (CAN_RXERR)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	REC 接收错误计数 (Receive Error Counter) **注：** 正常模式为只读状态，复位模式下可读写。

7.6.5.12. 0x03C CAN_TXERR

默认值：0x00000000			CAN 发送错误计数寄存器 (CAN_TXERR)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	TEC 发送错误计数 (Transmit Error Counter) **注：** 正常模式为只读状态，复位模式下可读写。

7.6.5.13. 0x040 CAN_BUF0

默认值：0x00000000			CAN 缓冲 0 寄存器 (CAN_BUF0)
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	IDE 标识符扩展位 (Identifier Extended)，用于明确报文格式 • 0x0：标准格式，SFF • 0x1：扩展格式，EFF

默认值：0x00000000			CAN 缓冲 0 寄存器 (CAN_BUF0)
位域	类型	默认值	描述
			注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。
6	R/W	0x0	RTR 远程发送请求位 (Remote Transmit Request) • 0x0：数据帧 • 0x1：远程帧 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。
5:4	–	–	–
3:0	R/W	0x0	DLC 数据长度代码 (Data Length Code) • 0x0：无数据内容 • 0x1~0x8：数据内容 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.14. 0x044 CAN_BUF1

默认值：0x00000000			CAN 缓冲 1 寄存器 (CAN_BUF1)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	ID • 标准格式标识符 [10:3] • 扩展格式标识符[28:21] 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.15. 0x048 CAN_BUF2

默认值：0x00000000			CAN 缓冲 2 寄存器 (CAN_BUF2)
位域	类型	默认值	描述
31:8	–	–	–
7:5	R/W	0x0	STID_EXID • 标准格式标识 [2:0] • 扩展格式标识符[20:18] 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。
4:0	R/W	0x0	EXID • 扩展格式标识符[17:13] 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.16. 0x04C CAN_BUF3

默认值：0x00000000			CAN 缓冲 3 寄存器 (CAN_BUF3)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA1_EXID • EXID[12:5]，扩展格式标识符[12:5] • STDATA1[7:0]，标准格式数据字节 1 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.17. 0x050 CAN_BUF4

默认值：0x00000000			CAN 缓冲 4 寄存器 (CAN_BUF4)
位域	类型	默认值	描述
31:8	–	–	–
7:3	R/W	0x0	STDATA2_EXID • EXID[4:0]，扩展格式标识符[4:0] • STDATA2[7:3]，标准格式数据字节 2[7:3] 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。
2:0	R/W	0x0	STDATA2 • STDATA2[2:0]，标准格式数据字节 2[2:0] 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.18. 0x054 CAN_BUF5

默认值：0x00000000			CAN 缓冲 5 寄存器 (CAN_BUF5)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA3_EXDATA1 • STDATA3[7:0]，标准格式数据字节 3 • EXDATA1[7:0]，扩展格式数据字节 1 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.19. 0x058 CAN_BUF6

默认值：0x00000000			CAN 缓冲 6 寄存器 (CAN_BUF6)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA4_EXDATA2 • STDATA4[7:0]，标准格式数据字节 4 • EXDATA2[7:0]，扩展格式数据字节 2 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.20. 0x05C CAN_BUF7

默认值：0x00000000			CAN 缓冲 7 寄存器 (CAN_BUF7)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA5_EXDATA3 • STDATA5[7:0]，标准格式数据字节 5 • EXDATA3[7:0]，扩展格式数据字节 3 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.21. 0x060 CAN_BUF8

默认值：0x00000000			CAN 缓冲 8 寄存器 (CAN_BUF8)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA6_EXDATA4 • STDATA6[7:0]，标准格式数据字节 6 • EXDATA4[7:0]，扩展格式数据字节 4

默认值：0x00000000			CAN 缓冲 8 寄存器 (CAN_BUF8)
位域	类型	默认值	描述
			注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.22. 0x064 CAN_BUF9

默认值：0x00000000			CAN 缓冲 9 寄存器 (CAN_BUF9)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA7_EXDATA5 • STDATA7[7:0]，标准格式数据字节 7 • EXDATA5[7:0]，扩展格式数据字节 5 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.23. 0x068 CAN_BUF10

默认值：0x00000000			CAN 缓冲 10 寄存器 (CAN_BUF10)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	STDATA8_EXDATA6 • STDATA8[7:0]，标准格式数据字节 8 • EXDATA6[7:0]，扩展格式数据字节 6 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.24. 0x06C CAN_BUF11

默认值：0x00000000			CAN 缓冲 11 寄存器 (CAN_BUF11)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	EXDATA7 • EXDATA7[7:0]，扩展格式数据字节 7 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.25. 0x070 CAN_BUF12

默认值：0x00000000			CAN 缓冲 12 寄存器 (CAN_BUF12)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	EXDATA8 • EXDATA8[7:0]，扩展格式数据字节 8 注： 对于接收缓冲只读，发送缓冲只写，复位模式时可读写。

7.6.5.26. 0x040 CAN_RXCODE0

默认值：0x00000000			CAN 接收代码 0 寄存器 (CAN_RXCODE0)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	RXCODE0 接收过滤代码 0 (Acceptance Code0)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.27. 0x044 CAN_RXCODE1

默认值：0x00000000			CAN 接收代码 1 寄存器 (CAN_RXCODE1)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	RXCODE1 接收过滤代码 1 (Acceptance Code1)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.28. 0x048 CAN_RXCODE2

默认值：0x00000000			CAN 接收代码 2 寄存器 (CAN_RXCODE2)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	RXCODE2 接收过滤代码 2 (Acceptance Code2)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.29. 0x04C CAN_RXCODE3

默认值：0x00000000			CAN 接收代码 3 寄存器 (CAN_RXCODE3)
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	RXCODE3 接收过滤代码 3 (Acceptance Code3)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.30. 0x050 CAN_RXMASK0

默认值：0x00000000			CAN 接收屏蔽 0 寄存器 (CAN_RXMASK0)
位域	类型	默认值	描述
7:0	R/W	0x0	RXMASK0 接收屏蔽 0 (Acceptance Mask0)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.31. 0x054 CAN_RXMASK1

默认值：0x00000000			CAN 接收屏蔽 1 寄存器 (CAN_RXMASK1)
位域	类型	默认值	描述
7:0	R/W	0x0	RXMASK1 接收屏蔽 1 (Acceptance Mask1)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.32. 0x058 CAN_RXMASK2

默认值：0x00000000			CAN 接收屏蔽 2 寄存器 (CAN_RXMASK2)
位域	类型	默认值	描述
7:0	R/W	0x0	RXMASK2 接收屏蔽 2 (Acceptance Mask2)，详细描述见 接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.33. 0x05C CAN_RXMASK3

默认值：0x00000000			CAN 接收屏蔽 3 寄存器 (CAN_RXMASK3)
位域	类型	默认值	描述
7:0	R/W	0x0	RXMASK3 接收屏蔽 3 (Acceptance Mask3)，详细描述见接收过滤器 注： 正常模式只读，复位模式时可读写。

7.6.5.34. 0x074 CAN_RXC

默认值：0x00000000			CAN 接收数据计数寄存器 (CAN_RXC)
位域	类型	默认值	描述
31:8	–	–	–
7:0	RO	0x0	RXC 接收数据计数 (Receive Message Counter)，存储接收缓冲中数据包个数

7.6.5.35. 0x078 CAN_RSADDR

默认值：0x00000000			CAN 接收缓冲起始地址寄存器 (CAN_RSADDR)
位域	类型	默认值	描述
31:6	–	–	–
5:0	R/W	0x0	RSADDR 接收缓冲器起始地址 (Receive Buffer Start Address) 注： 正常模式只读，复位模式时可读写。

7.6.5.36. 0x080 CAN_RXFIFO

默认值：0x00000000			CAN 接收 FIFO 寄存器 (RX FIFO)
位域	类型	默认值	描述
31:8	–	–	–
7:0	RO	0x0	RXFIFO 接收 FIFO (Receive FIFO)，0x80~0x17C 共 64 bytes 注： 正常模式只读，复位模式时可读写。

7.6.5.37. 0x180 CAN_TXB_RO

默认值：0x00000000			CAN 发送缓冲只读寄存器 (TX Buffer Read Only)
位域	类型	默认值	描述
31:8	–	–	–
7:0	RO	0x0	TXB_RO 发送缓冲只读寄存器 (Transmit Buffer Read Only) 每个寄存器只有低 8 位可有效访问，其余高位被读时返回 0。

7.6.5.38. 0x1C0 CAN_TX_DMA

默认值：0x00000000			CAN TX DMA 控制寄存器 (TX DMA CTL)
位域	类型	默认值	描述
31:6	–	–	–
5:4	R/W	0x0	TX_DMA_MODE TX DMA 模式 (Transmit DMA Mode) • 0x0：DMA 搬完数据后，需要 CPU 触发发送 • 0x1：DMA 搬完数据后，自动触发发送（正常模式） • 0x2：DMA 搬完数据后，自动触发发送（loopback 模式） • 0x3：DMA 搬完数据后，自动触发发送（单次发送）
3:1	–	–	–

默认值：0x00000000			CAN TX DMA 控制寄存器 (TX DMA CTL)
位域	类型	默认值	描述
0	R/W	0x0	TX_DMA_EN TX DMA 使能 (Transmit DMA Enable) • 0x0： 禁用 • 0x1： 使能

7.6.5.39. 0x1C4 CAN_TX_DMA_THR

默认值：0x00000000			CAN TX DMA 水位寄存器 (TX DMA THR)
位域	类型	默认值	描述
31:4	–	–	–
3:0	R/W	0x0	TX_DMA_THR TX DMA 水位 (Transmit DMA Threshold) 取值范围 0~15 byte

7.6.5.40. 0x1C8 CAN_RX_DMA

默认值：0x000000D0			CAN RX DMA 控制寄存器 (RX DMA CTL)
位域	类型	默认值	描述
31:8	–	–	–
7:4	R/W	0xD	RX DMA LEN RX DMA 任务传输长度（byte） 需要传输的帧长度应小于等于该长度。
3	–	–	–
2	R/W	0x0	RX_DMA_LEN_VLD RX DMA 传输长度有效 • 0x0： RX buffer 按照实际帧大小申请或者释放每一帧空间，dma 按照 rx_dma_len 来搬移数据，但比帧长度大的位置数据固定为 0。 • 0x1： RX buffer 按照 rx_dma_len 来申请或者释放每一帧空间，dma 按照 rx_dma_len 来搬移数据。
1	–	–	–

默认值：0x000000D0			CAN RX DMA 控制寄存器 (RX DMA CTL)
位域	类型	默认值	描述
0	R/W	0x0	RX_DMA_EN RX DMA 使能 (Receive DMA Enable) • 0x0： 禁用 • 0x1： 使能

7.6.5.41. 0x1CC CAN_RX_DMA_THR

默认值：0x00000000			CAN RX DMA 水位寄存器 (RX DMA THR)
位域	类型	默认值	描述
31:7	–	–	–
6:0	R/W	0x0	RX_DMA_THR RX DMA 水位 (Receive DMA Threshold) 取值范围 0~64 byte

7.6.5.42. 0xFFC CAN_VERSION

默认值：0x00000011			CAN 版本寄存器 (VERSION)
位域	类型	默认值	描述
31:0	RO	0x00000011	Version 版本号，采用 BCD 码显示，V1.1

7.7. Controller Area Network Flexible Data (CANFD)

可变数据控制器局域网 CANFD 是一种支持多主机的串行总线，广泛应用于汽车控制系统和一般工业环境中的区域网络控制，支持 CAN2.0B 和 CANFD 协议。

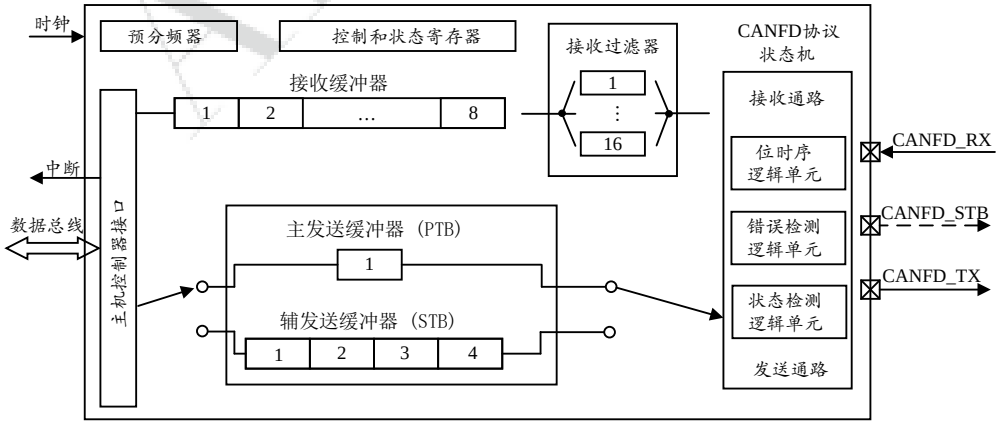
7.7.1. 特性说明

- 支持 CAN 2.0B（速率最高达 1 Mbit/s，最多 8 字节数据段）
- 支持 CANFD（速率最高达 10 Mbit/s，最多 64 字节数据段）
- 符合 ISO 11898-1:2015 或非 ISO Bosch 规范
- 支持一个八层接收缓冲器，一个单层主发送缓冲器 PTB，一个四层辅发送缓冲器 STB

- 支持 16 通道的独立和可编程内部 11 位或 29 位 ID 接收过滤器
- 支持多种运行模式：
  - 单发模式
  - 监听模式
  - 回环模式
  - 休眠模式
- 支持多种错误和中断检测
- 支持时间触发 CAN (TTCAN)，符合 ISO11898–4 协议等级 1 规范
- 支持 CiA603 时钟戳
- 兼容 AUTOSAR

7.7.2. 原理框图

图 7–22 CANFD 控制器结构



7.7.3. 功能描述

7.7.3.1. 典型应用

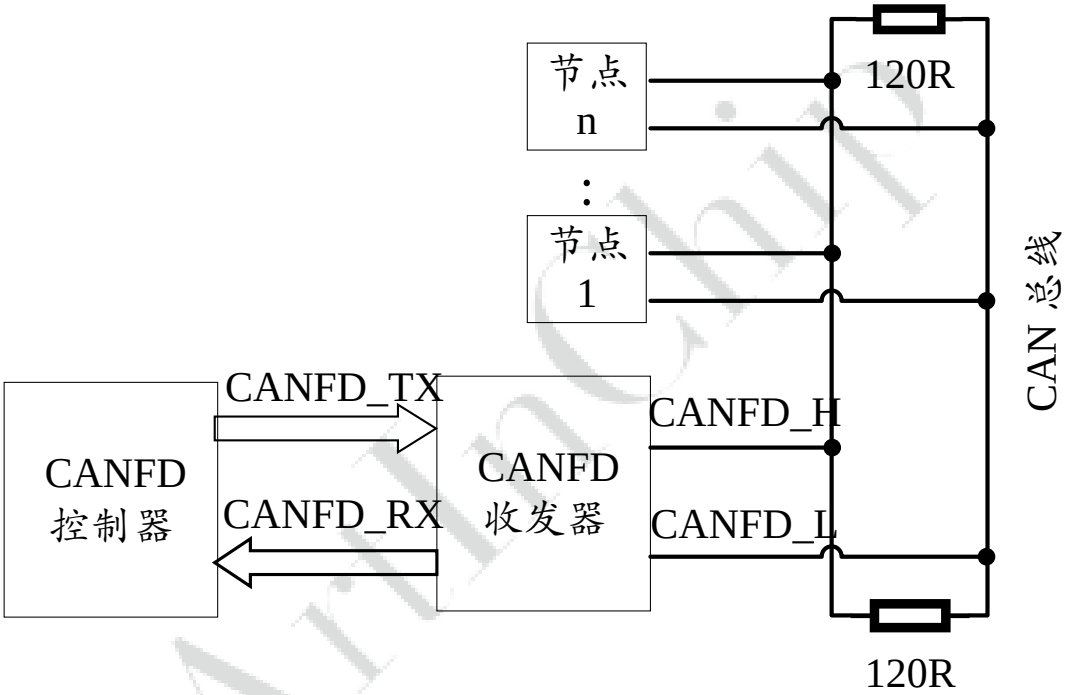


图 7–23 CANFD 典型应用

7.7.3.2. 运行模式

控制器包含四种运行模式：

- 单发模式：Transmission Primary Single Shot (TPSS) 或者 Transmission Secondary Single Shot mode（TSSS）

  对于信息发送，有时只需要单次发送，可以通过置位 TPSS（用于 PTB）或者置位 TSSS（用于 STB）实现。

  当传输不成功时，将发生以下情况：

  - 若 TPIE = 1，则 TPIF 被置位，相应的发送缓冲器数据被清除。
  - 若发生错误，KOER 和错误计数器相应更新。同时，若 BEIE = 1，则 BEIF 被置位，其它错误中断标志将相应地起作用。
  - 在仲裁丢失的情况下，若 ALIE = 1，则 ALIF 被置位。
  - 如果总线被另一个帧占用，则 CANFD 控制器等待直到总线空闲再尝试单发模式传输数据。

> **注：**
>
> 如果单发模式与 TSALL 一起使用并且 STB 中有一个以上的帧信息，则对每个帧信息进行一次单发传输，CANFD 控制器会前进到下一帧并在 STB 为空时停止。

- 监听模式：Listen Only Mode

  监听模式在不对总线产生任何影响的情况下监控 CAN 总线，CANFD 控制器的监听模式与 CANFD 规范中定义的总线监控功能兼容。监听模式可以用于自动比特率检测，能监视错误 (KOER、BEIF)。

  监听模式有以下特点：

  - 控制器无法将显性位写入总线，也不生成显性 ACK 响应，只生成隐性错误标志。
  - 无论发生任何错误情况，错误计数器都保持不变。
  - 若接收到的帧信息 ACK 错误后存在主动或被动错误标志，则监听模式下的控制器节点能够检测为 ACK 错误。

  注：

  应用监听模式过程中，需要注意：

  - 当传输处于活动状态时，不应激活监听模式。
  - 如果启用监听模式，则无法启动传输操作。
  - 外部回环模式 (LBME) 对监听模式的行为起着重要作用：
    - 如果禁用 LBME，则监听模式下节点不作任何传输操作，无法将任何显性位写入总线。
    - 如果使能 LBME，则允许节点传输帧。LBME 只允许传输包含可选 ACK 响应 (SACK) 的帧，但节点不会对来自其它节点的帧作出 ACK 响应，也不会生成错误或过载帧。

- 回环模式：Loop Back Mode

  CANFD 控制器支持两种回环模式，用于系统自测：

  - 内部回环模式 LBMI：CANFD 控制器与 CAN 总线断开连接，CANFD_TX 输出设置为隐性。输出数据流在内部反馈到输入。

    注：

    在 LBMI 模式下，节点生成 ACK 确认，以避免 ACK 错误。

  - 外部回环模式 LBME：CANFD 控制器可以通过以下两种方式获取自己发送出去的帧信息。
    1. 只需要短接控制器引出的 CANFD_TX 和 CANFD_RX 引脚，不需要外部收发器。
    2. 通过 CANFD_TX 和 CANFD_RX 引脚连接外部收发器芯片，收发器将 CANFD_TX 送过来的信号通过 CANFD_RX 回送给控制器。
       - 如果 SACK = 0，则不产生 ACK 确认，帧传输时会出现以下两种可能的结果，另一个节点也收到帧并生成 ACK：
         - 传输和接收均成功。
         - 没有其它节点连接到总线，导致 ACK 错误。
       - 如果 SACK = 1，产生 ACK 确认，没有其它节点连接到总线，导致 ACK 错误。

- 休眠模式：Standby Mode

  无总线活动，控制外部 Transceiver 收发器进入低功耗模式。

  使用寄存器位 STBY 驱动输出 CANFD_STB 引脚信号，可用于控制外部收发器的待机模式。

  - 传输激活（TPE、TSONE 或 TSALL 已置位）和休眠模式（置位 STBY），只能二选一。
  - 如果置位 STBY，收发器进入休眠模式。收发器无法全速接收帧，但会监控 CAN 总线的显性状态。
    - 如果显性状态在收发器数据表中定义的时间内处于活动状态，则收发器会将 CANFD_RX 信号拉低。
    - 如果 CANFD_RX 为低 CANFD 控制器自动将 STBY 清零，禁用收发器的休眠模式。操作过程不会中断 CANFD 控制器。

  收发器从休眠模式切换到工作模式需要时间，过程中无法成功接收初始唤醒帧。因此，最近处于待机状态的节点将不回应 ACK 响应。

  - 如果总线上没有 CAN 节点使用 ACK 响应唤醒帧，则会导致唤醒帧的发送节点出现 ACK 错误。此时，发送节点将自动重复帧发送。
  - 重复帧发送期间，收发器将回到工作模式，CANFD 控制器将接收帧并回以 ACK 响应。STBY 不受位 RESET 的影响。
  - 一个节点发送一个帧用于唤醒。如果所有其它节点都处于待机模式，则发送节点收到 ACK 错误并将自动重复该帧。

7.7.3.3. 位时序

在传输过程中，帧信息中的位具有以下段定义：

- SS 段 (SYNC SEG)：同步段，大小为 1 Time Quanta (TQ)。当总线上出现帧起始信号 (SOF) 时，其它节点上的控制器根据总线上的这个下降沿，对自己的位时序进行调整，把该下降沿包含到 SS 段内，根据起始帧来进行硬同步。
- PTS 段 (PROP SEG)：传播时间段，用于补偿网络的物理延时时间，包括发送单元的输出延迟、总线上信号的传播延迟和接收单元的输入延迟。
- PBS1 段 (PHASE SEG1)：相位缓冲段，主要用来补偿边沿阶段的误差，时间长度在重新同步的时候可以加长。
- PBS2 段 (PHASE SEG2)：相位缓冲段，用来补偿边沿阶段误差，时间长度在重新同步时可以缩短。
- SJW (reSynchronization Jump Width)：重新同步补偿宽度，即在重新同步的时候，PBS1 和 PBS2 段的允许加长或缩短的时间长度，SJW 加大后允许误差加大，但通信速度下降。SJW 为补偿此误差的最大值。

对于 CANFD 控制器，如图 7–24 : CANFD 位时序所示：Tseg1 段包含帧信息 SS 段，PTS 段和 PBS1 段，而 Tseg2 段包含帧信息 PBS2 段。时钟 CLK 提供给整个控制器的系统时钟 Fclk，图 7–24 : CANFD 位时序示例中，PRESC 为 2， CAN 的 Time Quanta (TQ) 时间量子即一个 TQ 时长为 PRESC / Fclk。

Tseg1 + Tseg2 即为一个位的采样时长，也等于一个波特率时长。

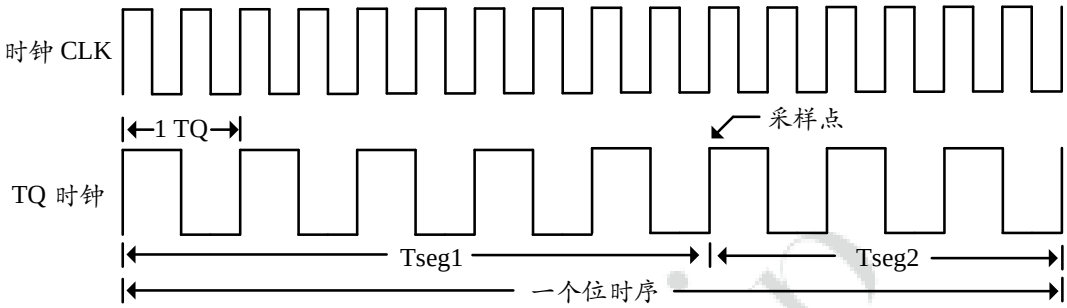


图 7–24 CANFD 位时序

控制器时序的建议设定范围如 CANFD 控制器时序建议设定范围所示：

表 7–32 CANFD 控制器时序建议设定范围

设置	建议设定范围	备注
Tseg1	[2…65] TQ	CAN 2.0B 速率
	[2…65] TQ	CANFD 普通速率
	[2…17] TQ	CANFD 高速速率
Tseg2	[1…8] TQ Tseg1 ≥ Tseg2+2	CAN 2.0B 速率
	[1…16] TQ Tseg1 ≥ Tseg2+2	CANFD 普通速率
	[1…8] TQ Tseg1 ≥ Tseg2+1	CANFD 高速速率
Tsjw	[1…16] TQ Tseg2 ≥ Tsjw	CAN 2.0B 速率
	[1…16] TQ Tseg2 ≥ Tsjw	CANFD 普通速率
	[1…8] TQ Tseg2 ≥ Tsjw	CANFD 高速速率

对于普通速率或者高速速率的帧波特率，对应的分频公式如普通速率/高速速率分频因子计算公式所示：

表 7–33 普通速率/高速速率分频因子计算公式

普通速率	高速速率
Tseg1 =（S_Seg_1 + 2）TQ	Tseg1 =（F_Seg_1 + 2）TQ
Tseg2 =（S_Seg_2 + 1）TQ	Tseg2 =（F_Seg_2 + 1）TQ
Tsjw =（S_SJW + 1）TQ	Tsjw =（F_SJW + 1）TQ
PRESC = S_PRESC + 1	PRESC = F_PRESC + 1

CANFD 控制器还提供了一个可选的 Transmitter delay compensation（发送延迟补偿 TDC）和 Receiver delay compensation（接收延迟补偿 RDC）配置功能。当 TDC 功能使能后，RDC 功能也自动使能。CANFD 有可能因数据通过外部收发器的延迟或者总线产生的延迟而超过了一个位的时长。这种情况就需要补偿。如果没有 TDC，CANFD 帧就有可能检测到一个位错误。

图 7–25：传输延迟给了一个大传输延迟的示例。Txd 发送数据流标识出了位 A 和位 B。每一个位时间包含了 Tseg1 = 5 TQ，以及 Tseg2 = 3 TQ。传输延迟超过了两个位时长。原始的采样点 SP 无法用于采样正确的位值，CANFD 规范定义了一个额外的第二采样点 SSP。如果 TDCEN = 1，那么 TDC 使能，CANFD 控制器自动确认传输延迟。SSP 的位置由传输延迟加上 SSP 偏移的时间（即 SSPOFF）。SSPOFF 建议设置为同样 Tseg1 的值。

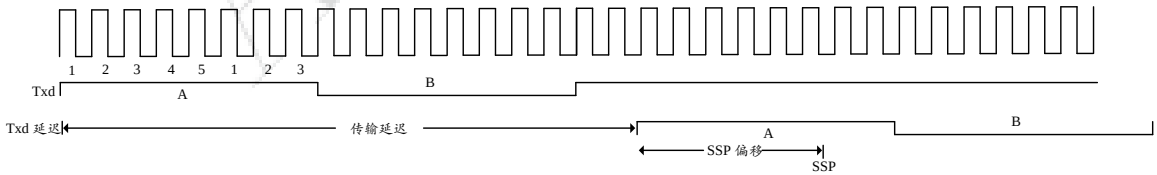


图 7–25 传输延迟

7.7.3.4. 通信报文

CANFD 通信报文类型包含数据帧、远程帧、错误帧、过载帧、间隔帧这五种类型。

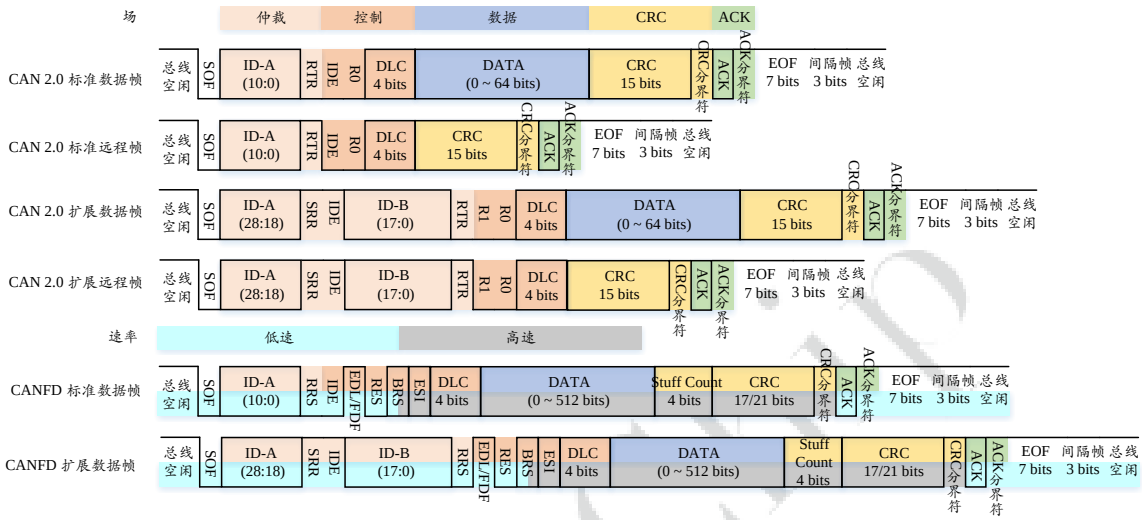


图 7–26 CANFD 通信报文帧格式解析

表 7–34 CANFD 帧位定义

数据/远程帧	描述
SOF	帧起始，一位长度，用于同步总线上节点的单个显性位。
ID–A	标识符 A，11 位长度，对应标准格式中的 11 位标识符 (ID10~ ID0)，或扩展格式中 29 位标识符的前 11 位 (ID28~ ID18)。
ID–B	标识符 B，18 位长度，扩展格式中 29 位标识符的剩余 18 位 (ID17 ~ ID0)。
RTR	远程发送请求位，1 位长度，显示当前报文是数据帧（显性 0）还是远程帧（隐性 1）。当某个数据帧和一个远程帧有相同标识符时，数据帧始终优先于远程帧仲裁。
SRR	代替远程发送请求位，1 位长度，在扩展格式中以替代标准格式相同位置的 RTR 位。
RRS	远程请求替换位，CANFD 取消了对远程帧的支持，用 RRS 位替换了 RTR 位，始终是显性位。
IDE	标识符扩展位，1 位长度，显示当前报文是标准格式（显性 0）还是扩展格式（隐性 1）。当某标准帧和扩展帧有相同基标识符时，标准帧将始终优先于扩展帧仲裁。
EDL	扩展数据长度位，始终是隐性位。Bosch 的 CANFD 规范（非 ISO ）使用名称 EDL，而 CANFD ISO 规范使用名称 FDF 。两个名字都是同义词。
FDF	灵活数据速率格式，始终是隐性位，意义同 EDL 位。标准 ISO 帧采用 FDF 对该位进行描述。
RES	保留位，始终是显性位。
R1	保留位，始终是显性位。
R0	保留位，始终是显性位。

表 7–34 CANFD 帧位定义 (续)

数据/远程帧	描述
BRS	位速率切换，这个额外的位允许 CANFD 帧以两种不同的速率发送。如果 BRS 位为显性被发送，则后续所有位发送速率不变。如果 BRS 位为隐性，则在此位后帧格式将使用较高的位速率，直到并包括 CRC 分界符。
ESI	错误状态指示器，通常显性发送为主。如果 CANFD 帧发送节点变为被动错误，则该位将被隐性发送，指示发送方节点存在重要的通信问题。
DLC	数据长度代码， 4 位长度，包含 0 ~ 15 中任一数值。
Stuff count	填充计数由以下两个元素组成： • 格雷码计算（前 3 位）：CRC 区域之前的填充位进行格雷码计算得到的值。 • 奇偶校验（最后 1 位）：通过格雷码计算后的值的奇偶校验（偶校验）。 对于 CANFD ISO 帧，填充计数作为 CRC 字段的一部分传输。对于 CANFD 非 ISO 帧，填充计数不是帧的一部分。此外，CRC 校验器对非 ISO 和 ISO 帧有不同的初始化。因此 ISO 和非 ISO 帧是不兼容的。
数据字节	表示数据帧的数据负载量，该字节数量应与 DLC 的值匹配。首先发送低位数据字节，各数据字节优先发送最高有效位。
CRC	对于 CAN 2.0，CRC 采用 15 位。 对于 CANFD，当传输数据为 16 字节或更少时，采用 17 位 CRC。当传输数据超过 16 字节时，采用 21 位 CRC。
CRC 分界符	分界符，1 位长度隐性位。
ACK 槽	应答槽，1 位长度，用于接收节点，表示是否已成功接收数据帧或远程帧。发送节点将在应答槽中发送一个隐性位，若接收到的帧无错误，则接收节点用一个显性位以示应答。
ACK 分界符	分界符，1 位长度隐性位。
EOF	帧结束，7 位长度隐性位，标志着数据帧或远程帧的结束。

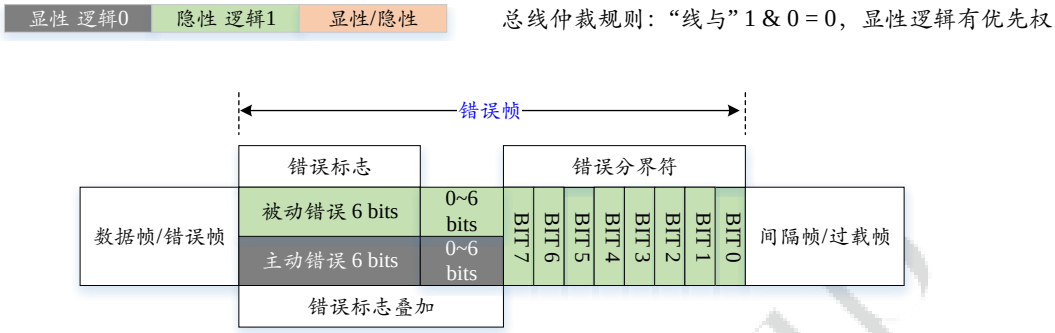


图 7–27 错误帧

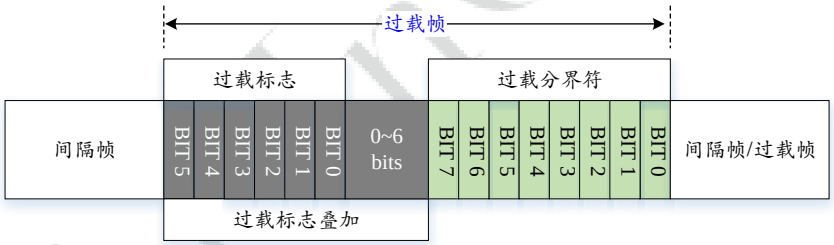


图 7–28 过载帧

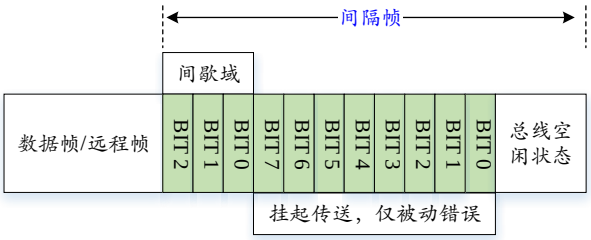


图 7–29 间隔帧

7.7.3.5. 数据缓冲器

CANFD 控制器包括一个八层（时隙）接收缓冲器，一个主发送缓冲器（简称 PTB ）和一个四层（时隙）辅发送缓冲器（简称 STB ）。

接收缓冲器指向入口为 RBUF，RBUF 具有 80 个字节长度，每一笔接收到的帧信息都先储存在这里，通过设置 RREL = 1 可以切换到接收缓冲器的下一层。RBUF 存储的数据格式区分标准帧和扩展帧，具体见RBUF（标准帧格式）和RBUF（扩展帧格式）。

- 通过设置接收过滤器，可以只接收特定的帧 ID 或者帧 ID 段范围，从而减少主机接收负荷。
- RBUF 只支持 32 位地址对齐模式访问，不支持 8 位/ 16 位地址对齐模式访问。

PTB 和 STB 构成了发送缓冲器，其指向入口为 TBUF。对于 STB，可以通过设置 TSNEXT = 1 切换到 STB 的下一层。TBUF 具有 72 个字节长度，每一笔要发送的帧都储存在这里。TBUF 存储的数据格式区分标准帧和扩展帧，具体见TBUF（标准帧格式）和TBUF（扩展帧格式）。

- PTB 优先级比 STB 高，无论 ID 值为何值，PTB 的帧总会比 STB 优先发送。
- 如果选择 PTB (TBSEL = 0)，则设置 TSNEXT = 0。
- 使用 PTB 发送时应置位 TPE 以启动传输。
- STB 具有 FIFO 和优先级两种工作模式。
  - 在 FIFO 模式下，STB 中的帧采用“先进先出”的工作模式发送。
  - 在优先级模式下，STB 会后台根据帧 ID 进行优先级排序，最高优先级的帧最优先发送。
- 使用 STB 发送，应置位 TSONE 或 TSALL 以启动传输。
- TBUF 和 RBUF 一样，只支持 32 位地址对齐模式访问，不支持 8 位/ 16 位地址对齐模式访问。

当传输完成后，以下传输中断将置位：

- 对于 PTB，如果使能了 TPIE，则 TPIF 置位。
- 对于使用 TSONE 的 STB，如果一个消息已经完成并且使能了 TSIE，则 TSIF 置位。
- 对于使用 TSALL 的 STB，如果所有消息都已完成并且使能了 TSIE，则 TSIF 置位。

注：

- 当 STB 为空时不允许置位 TSONE 或 TSALL。
- STB 优先级模式在对比时，将标准帧的 11 位依次往高位挪到与扩展帧 28:18 位逐位依次对比。挪动后标准帧对应于扩展帧 17:0 位补零。详见示例STB 优先级模式举例说明。

STB 优先级模式举例说明

为了便于理解，下面举几个例子说明，举例的 ID 值分别为 0，0x1235，0x2342，0x12345，0x34345，1，5，0x434234，0x1111111，0x2342342，0x111，0x125，0x222，0x300，0x12345678，原始的 ID 位排布如STB 优先级模式移位前的 ID 位排布：

表 7–35 STB 优先级模式移位前的 ID 位排布

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ID 值	ID 类型
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0x1235	扩展
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0x2342	扩展
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0x12345	扩展
0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0x34345	扩展
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	标准
0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0x434234	扩展
0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0x1111111	扩展
0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0x2342342	扩展
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0x111	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	1	0x125	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0x222	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0x300	标准
1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0x12345678	扩展
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0x555	标准
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0xC000000	扩展

移位过后各个位的排布如STB 优先级模式移位后的 ID 位排布：

表 7–36 STB 优先级模式移位后的 ID 位排布

28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	ID 值	ID 类型
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	标准
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	1	0x1235	扩展
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0x2342	扩展
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0x12345	扩展
0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0	0	1	0	1	0x34345	扩展
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	标准
0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	标准
0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0x434234	扩展
0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0x1111111	扩展
0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0	0	0	1	0	0x2342342	扩展
0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x111	标准
0	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x125	标准
0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x222	标准
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x300	标准
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0xC000000	扩展
1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	0x12345678	扩展
1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0x555	标准

从移位后的 ID 可以看出，优先级从 28 位开始逐位往下对比，位值为 0 的，比位值为 1 的优先级要高。特别要注意的是 0x300 和 0xC000000 ，两者调整过后的优先级一致，这种情况下标准帧优先于扩展帧发送。

7.7.3.5.1. RBUF 和 RBUF 格式

表 7–37 RBUF（标准帧格式）

地址	位位置								功能描述
–	7	6	5	4	3	2	1	0	–
RBUF + 0	ID(7:0)								ID
RBUF + 1	–					ID(10:8)			ID
RBUF + 2	–								ID
RBUF + 3	ESI	–							ID
RBUF + 4	IDE = 0	RTR	FDF	BRS	DLC(3:0)				控制
RBUF + 5	KOER			TX	–				状态
RBUF + 6	CYCLE_TIME(7:0)								TTCAN
RBUF + 7	CYCLE_TIME(15:8)								TTCAN
RBUF + 8	d1(7:0)								数据字节 1
RBUF + 9	d2(7:0)								数据字节 2
…	…								…
RBUF + 71	d64(7:0)								数据字节 64
RBUF + 72	RTS(7:0)								CiA603
…	…								…
RBUF + 79	RTS(63:56)								CiA603

表 7–38 RBUF（扩展帧格式）

地址	位位置								功能描述
–	7	6	5	4	3	2	1	0	–
RBUF + 0	ID(7:0)								ID

表 7–38 RBUF（扩展帧格式）(续)

地址	位位置								功能描述
RBUF + 1	ID(15:8)					ID(10:8)			ID
RBUF + 2	ID(23:16)								ID
RBUF + 3	ESI	–		ID(28:24)					ID
RBUF + 4	IDE = 1	RTR	FDF	BRS	DLC(3:0)				控制
RBUF + 5	KOER			TX	–				状态
RBUF + 6	CYCLE_TIME(7:0)								TTCAN
RBUF + 7	CYCLE_TIME(15:8)								TTCAN
RBUF + 8	d1(7:0)								数据字节 1
RBUF + 9	d2(7:0)								数据字节 2
…	…								…
RBUF + 71	d64(7:0)								数据字节 64
RBUF + 72	RTS(7:0)								CiA603
…	…								…
RBUF + 79	RTS(63:56)								CiA603

表 7–39 TBUF（标准帧格式）

地址	位位置								功能描述
–	7	6	5	4	3	2	1	0	–
TBUF + 0	ID(7:0)								ID
TBUF + 1	–					ID(10:8)			ID
TBUF + 2	–								ID
TBUF + 3	TTSEN	–							ID
TBUF + 4	IDE = 0	RTR	FDF	BRS	DLC(3:0)				控制

表 7–39 TBUF（标准帧格式）(续)

地址	位位置	功能描述
TBUF + 8	d1(7:0)	数据字节 1
TBUF + 9	d2(7:0)	数据字节 2
…	…	…
TBUF + 71	d64(7:0)	数据字节 64

表 7–40 TBUF（扩展帧格式）

地址	位位置								功能描述
–	7	6	5	4	3	2	1	0	–
TBUF + 0	ID(7:0)								ID
TBUF + 1	ID(15:8)								ID
TBUF + 2	ID(23:16)								ID
TBUF + 3	TTSEN	–		ID(28:24)					ID
TBUF + 4	IDE = 1	RTR	FDF	BRS	DLC(3:0)				控制
TBUF + 8	d1(7:0)								数据字节 1
TBUF + 9	d2(7:0)								数据字节 2
…	…								…
TBUF + 71	d64(7:0)								数据字节 64

7.7.3.6. 中止发送

如果发送缓冲器的消息由于优先级低而无法发送，将长时间阻塞缓冲器。为了避免这种情况，如果传输尚未开始，主机控制器可以通过分别置位 TPA 或 TSA 来撤回传输请求。

TPA 和 TSA 都使能一个中断标志 AIF。只有当 CANFD 控制器没有向 CAN 总线传输任何内容时，才会执行中止操作。

中止操作的执行会有以下情况：

- 如果节点丢失仲裁，将在仲裁之后执行中止操作。
- 在 CAN 节点未收到确认的不成功传输后，错误计数器会增加，并且将执行中止操作。
- 如果 STB 中至少还有一帧信息剩余，而主机已命令传输所有帧 (TSALL = 1)，控制器则会向主机发出完成的帧信息和中止信号。

中止传输需要一些时间，具体取决于 CAN 帧信息通信速度和帧长度。执行中止，则会导致以下结果：

- TPA 释放 PTB 导致 TPE = 0。释放 PTB 后，帧数据仍保存在 PTB 中。
- TSA 释放 STB 的单个消息或所有消息。这取决于是使用 TSONE 还是 TSALL 来启动传输。TSSTAT 将相应更新。

注：

不建议同时置位 TPA 和 TSA。如同时置位 TPA 和 TSA，则 AIF 将被置位，PTB 和 STB 的传输都可能被中止，以下中断标志可能被置位：

- AIF（每次 PTB 和 STB 传输中止触发一次）
- TPIF + AIF
- TSIF + AIF
- TPIF + TSIF（很少见，只有当控制器没有立即处理 TPIF 时才会发生）
- TPIF + TSIF + AIF（很少见，只有当控制器不立即处理 TPIF 和 TSIF 时才会发生）

7.7.3.7. 中断管理

CANFD 控制器提供一些中断，可通过设置对应的寄存器中相应的使能位，每个中断源都可以单独使能和禁用（AIF 和 TEIF 除外，没有对应的使能控制位）。

当有一个或多个中断触发，控制器中断请求有效，当所有中断位都被清除，控制器中断请求失效。向对应的中断标志位写入 1 可以清零对应的中断标志，写 0 无效。

7.7.3.8. Time–Trigger CAN (TTCAN)

Time–Trigger CAN（时间触发 CAN，TTCAN ）是一种根据 ISO 11898–4 标准协议定义的操作模式，将帧在预定义的时间窗口进行发送。一共有三种时间窗口类型：

- 独占时间窗口 (Exclusive time window)：同一时间内只允许一个节点发送定义好的 ID 的帧数据。
- 自由时间窗口 (Free time window)：保留为后续扩展用的时间窗口。
- 仲裁时间窗口 (Arbitrating time window)：多个节点占用并发生仲裁的事件窗口。

TTCAN 时序由系统管理员组织起一个系统矩阵，如图 7–30 : TTCAN 系统矩阵所示：

- 每一行称为基本周期，同时由参考信息发起。
- 参考信息只允许由时间主机发送。其它信息时间窗口无主从机限制。

- 信息的 SOF 时间称为同步标记 (Sync Mark) 。参考信息的同步标记称为参考标记 (Ref Mark) 。时序由一个 16 位定时器完成。
- 计时器和参考标记的偏差称为循环时间 (cycle time) 。换句话说，每个基本周期在参考标记处开始周期时间，循环时间存储于 RBUF 中作为一个时间戳 (time stamp)

图 7–30

TTCAN 系统矩阵

如果 CANFD 控制器需要作为一个时间主机，则参考信息需要放在 TBUF 中，并且当硬件触发器激活时可以被发送出去。时间从机和时间主机对参考消息的检测都是自动完成的。该过程可以配置正确的 REF_ID 和 REF_IDE 值来实现。在检测完成后，CANFD 控制器自动更新参考标记用于启动循环时间。

CANFD 控制器还提供了一个观测触发器用于辅助检查上一个参考信息到当前时间是否超时。主机应当设置观测触发器，用于周期场景确认。每两个基本周期之间，或者事件触发场景在当前与下一个事件触发周期之间，必须存在一个时间间隔。

当 TTTBM = 1，TBUF 在 TTCAN 模式下，STB 中的每层数据都可以由 TBPTR 寻址，并且可以通过 TBF 和 TBE 进行“满”或者“空”的标记，被标记为“满”的层会被写保护。TBSEL 和 TSNEXT 在此种模式下无效。PTB 可以被 TBPTR = 0 寻址，PTB 跟 STB 优先级相等，没有特殊优先级。此时 TBUF 中也没有 FIFO 或者优先级模式的处理方式。信息触发器定义了该信息需要被发送的时间点，并且通过 TTPTR 进行选取信息。如果触发器触发事件发生，则被选择的信息就启动发送。TPE，TSONE，TSALL，TPSS，以及 TPA 在 TTCAN 模式下无效。

当 TTTBM = 0，TBUF 在 TTCAN 模式下变成了事件驱动的 CAN 通讯模式，以及时间戳下的接收信息方式。如果 TTEN = 0 则 PTB 和 STB 仍然保持跟之前相同的行为，即 PTB 优先级高于 STB，以及 STB 可以工作于 FIFO 或者优先级模式。

如果 TTEN = 1，则 16 位定时器开始运行。如果参考信息被检测到或者如果时间主机成功发送了它的参考信息，CANFD 控制器会复制该信息的同步标记变成参考标记，并且设置周期时间为 0。对参考信息的成功接收会置位 RIF，而成功发送则会置位 TPIF 或者 TSIF。然后主机需要准备下一次触发行为。

7.7.3.9. 接收过滤器

为了减少主机控制器接收帧的负载，CANFD 控制器支持对接收的帧 ID 进行过滤，以决定该帧是否要存储进 RBUF 中。

CANFD 控制器一共支持 16 个通道的接收过滤器，每个通道通过 ACODE_x 和 AMASK_x 进行对应配置，ACODE_x 是参考目标 ID，AMASK_x 是掩码位控制，其中：

- 0 表示被接收帧的对应位必须与 ACODE_x 的对应位一致才可接收。
- 1 表示任意位均可（即不进行比对）。

需要注意的是， 16 个过滤通道是“或”的关系，即只需要满足任何一个通道的过滤条件，即可被正确接收。并且寄存器 CANFD_ACODEx 和 CANFD_MASKx 的值只能使用 32 位的访问模式，且必须在 RESET = 1 的情况下才可配置。

7.7.3.10. 错误管理

关于错误管理， CANFD 控制器具有以下功能：

- 进行自动错误处理，包括自动消息重传和自动删除收到的错误消息。
- 如果主机需要，CANFD 控制器可以选择提供有关错误的详细信息，并通过中断向主机发出每个错误信号。

  主机上可以运行诸如 CANFD 总线监视器之类的应用程序。

每个 CANFD 节点都有三种错误处理状态：

- 主动错误：节点在检测到错误时自动发送主动错误帧。
- 被动错误：节点在检测到错误时发送被动错误帧，即不会向总线传输显性值，而是期待其它 CANFD 节点传输的错误帧。
- 总线关闭：在出现太多错误后，节点会进行“总线关闭”，从而停止接触总线。

为了处理三种错误状态，每个 CANFD 节点都有两个错误计数器：发送错误计数器和接收错误计数器。两者都按照 CANFD 规范的定义递增和递减，并且节点在达到 CANFD 规范定义的计数器限定阈值时进入相应的错误状态。

- 如果有错误，错误计数器会增加。
- 有效帧传输或有效接收导致计数器递减。

CANFD 控制器如果接收到帧信息并且检测到错误，则丢弃接收到的帧信息，帧的发送方将重新传输该帧信息。RBUF 中的帧信息永远不会被有错误的帧信息覆盖。只有接收的有效帧信息才可能导致 RBUF 溢出。

- 如果 RBALL = 1 ，则所有接收到的数据帧都会被存储，即使是有错误的数据帧。

  此条件也适用于回环模式。

  RBUF 中只存储数据帧，不存储错误帧或过载帧。如果 CiA 603 时间戳已启用 (TIMEEN = 1) 并且为 EOF (TIMEPOS = 1) 配置了时间戳，则在出现错误时会在错误帧的开头获取时间戳。大多数可能的错误只有在节点是帧的发送者时才会发生。在这种情况下，如果回环模式被激活，帧将仅存储在 RBUF 中。
- 当 TECNT 或者 RECNT 的错误计数值达到 EWL 的配置值，则会触发 EIF 中断（当 EIE 使能）。

  EWL 的实际值应该为 (EWL + 1) × 8。
- 控制器识别 CAN 总线上的错误并将最后一个错误事件存储在 KOER 中。每个新的错误事件都会覆盖之前存储的 KOER 值，KOER 随每个新错误而更新。

KOER 错误类型如下所示：

表 7–41 KOER 错误类型

KOER	场景	描述
NO ERROR 没有错误	发送和接收方	成功接收
BIT ERROR 位错误	接收器	只能发生在 ACK 时隙。所有存储的数据均有效。
	发送器	有效载荷总是无效的。包括 ID 的标头位可能是有效的。 • 检测过程不包括仲裁阶段的错误位 。 • 帧的发送器在仲裁阶段检测到的填充位错误属于一个位错误，此时标头位无效，无法信任标头位。 但如果标头与帧的预期标头匹配，则可用于进一步调查。
FORM ERROR 格式错误	发送和接收器	仅包含数据框中的 FORM ERROR，包括 ： • CRC 分界符 • ACK 分界符 • EOF 位中的错误 所有存储的数据均有效。
STUFF ERROR 填充错误	接收器	STUFF ERROR 的位置未知。所有存储的数据均无效。
	发送器	只能在仲裁期间发生。所有存储的数据均无效。
ACK ERROR ACK 错误	接收器	仅当节点处于 LOM 中时才会发生。所有存储的数据均有效。
	发送器	发送器只能在没有自我确认的环回模式下发生。所有存储的数据均有效。
CRC ERROR CRC 错误	接收器	所有存储的数据均无效。

7.7.3.11. 仲裁丢失捕获

CANFD 控制器能够检测仲裁字段中仲裁丢失的确切位置。如果使能了 ALIE 中断，则 ALIF 中断信号自动置位。如果节点能够赢得仲裁，则 ALC 的值保持不变， ALC 持有上次仲裁丢失的旧值。ALC 的值定义如下：一帧以 SOF 位开始传输 ID 的第一位。第一个 ID 位的 ALC 值为 0，第二个 ID 位的 ALC 值为 1，依此类推。仲裁只允许在仲裁域中进行。因此 ALC 的最大值为 31，即扩展帧中的 RTR 位。

如果标准远程帧与扩展帧进行仲裁，则扩展帧在 IDE 位失去仲裁并且 ALC 将为 12。关于标准帧和扩展帧的详细说明，可查看 RBUF 和 RBUF 格式。

7.7.4. 编程指南

7.7.4.1. 控制器初始化

上电后，CANFD 控制器的初始化操作步骤如下：

1. 配置模块时钟源，并使能模块时钟，释放模块复位状态。
2. 配置 CANFD 引脚功能。
3. RESET = 1 将 CANFD 置于复位状态。
4. 配置所需的波特率分频因子。
5. RESET = 0 使 CANFD 退出复位状态。
6. 配置对应的中断使能控制位。
7. 发送模式选择，若选择单发模式，则置位 TPSS 或者 TSSS。否则，则清零 TPSS 和 TSSS。
8. 进入传输阶段，分别执行下列对应条件的相应操作：
    - 若选择 PTB 启动发送操作，则将帧数据填进 TBUF，置位 TPE，等待 TPE 执行完成。
    - 若选择 STB 启动发送操作，则将帧数据填进 TBUF，每写进一层，则需要置位 TSNEXT 切换到 STB 的下一层。以下配置二选一：
        - 置位 TSONE，等待 TSONE 完成。
        - 置位 TSALL，等待 TSALL 完成。
    - 若启动接收操作，则可以使能 RIE 、 RAFIE 或者 RFIE 中的任意一个，并在中断处理函数中接收数据。

> **注：**
>
> 主机通过 RBUF 获取收缓冲器或者每一时隙帧信息后，需要置位 RREL 切到接收缓冲器的下一时隙。

7.7.4.2. 波特率推荐计算

以 CANFD 帧格式举例，若用户只采用 CAN 2.0 帧格式，则对应的 CANFD 帧格式高速速率部分的系数配置可以忽略。关于 CANFD 控制器的波特率计算方法，可以参考下列步骤：

> **注：**
>
> CANFD 帧格式普速域波特率必须小于等于 1 Mbps，高速域波特率必须小于等于 10 Mbps，且普速域波特率必须小于高速域波特率。

1. 配置 CAN 2.0 帧格式以及 CANFD 帧格式的普速域的时钟分频系数，相关寄存器如下所示：
    - CANFD_SSEG1
    - CANFD_SSEG2

- CANFD_SSJW
- CANFD_SPRESC

2. 配置 CANFD 帧格式的高速域的时钟分频系数，相关寄存器如下所示：
   - CANFD_FSEG1
   - CANFD_FSEG2
   - CANFD_FSJW
   - CANFD_FPRESC
3. 选择合适的采样占空比，典型值是 80%，一般在 50% ~ 100% 之间。普速域占空比和高速域占空比可根据实际情况设置不同值。
4. 必须分开计算普速域波特率和高速域波特率的分频因子：
   a. 依次循环执行 S_PRESC 或者 F_PRESC 从 1 到 256。
   b. 对每一个 S_PRESC 或者 F_PRESC 值，计算每一个帧信息位包含的 TQ 数量。

      计算公式： nTQ = (clk_src / presc) / baudrate，其中

      - nTQ 表示每一个帧信息位包含的 TQ 数量。
      - clk_src 表示 CANFD 控制器的时钟源。
      - presc 即为 S_PRESC 或者 F_PRESC 值。
      - baudrate 即为对应的目标普速域或者高速域的波特率。

      计算得到的 nTQ 若不为整数，则进一步评估 nTQ + 1 和 nTQ，选择更靠近目标波特率的值作为 nTQ 的配置值。
   c. 计算采样点前的时间段 Tseg1。计算公式为 Tseg1 = nTQ × duty_cycle：
      - nTQ 表示每一个帧信息位包含的 TQ 数量。
      - duty_cycle 表示整个帧信息位的采样点的占空比

      若计算出来的 Tseg1 值不为整数，则进一步评估和比较 Tseg1 和 Tseg1 + 1 值，将更靠近目标占空比值作为 Tseg1 的配置值。

      。
   d. 计算采样点后的时间段 Tseg2。计算公式为 Tseg2 = nTQ – Tseg1。
   e. 在满足CANFD 控制器时序建议设定范围的约束情况下，计算并保存根据当前的 nTQ、Tseg1 和 Tseg2 值计算出来的跟目标波特率以及目标占空比的误差值。
   f. 计算 重新同步补偿宽度时间 Tsjw。计算公式为 Tsjw = Tseg2：

> **注：**
>
> 若有使用高速域波特率计算，则可以预配置 TDCEN = 1，SSPOFF = Tseg1。

g. 跳转到下一个 S_PRESC 或者 F_PRESC 值，重复执行步骤 b 到 步骤 f，分别计算 nTQ，Tseg1 和 Tseg2 值以及波特率和占空比误差值，并跟上一个 S_PRESC 或者 F_PRESC 值计算出来的波特率和占空比误差值作对比，选取最小的误差值的配置值作为最优解。

5. 确认此时 RESET = 1，即 CANFD 控制器置于复位状态。
6. 按照公式普通速率/高速速率分频因子计算公式将配置值更新到对应的寄存器中。

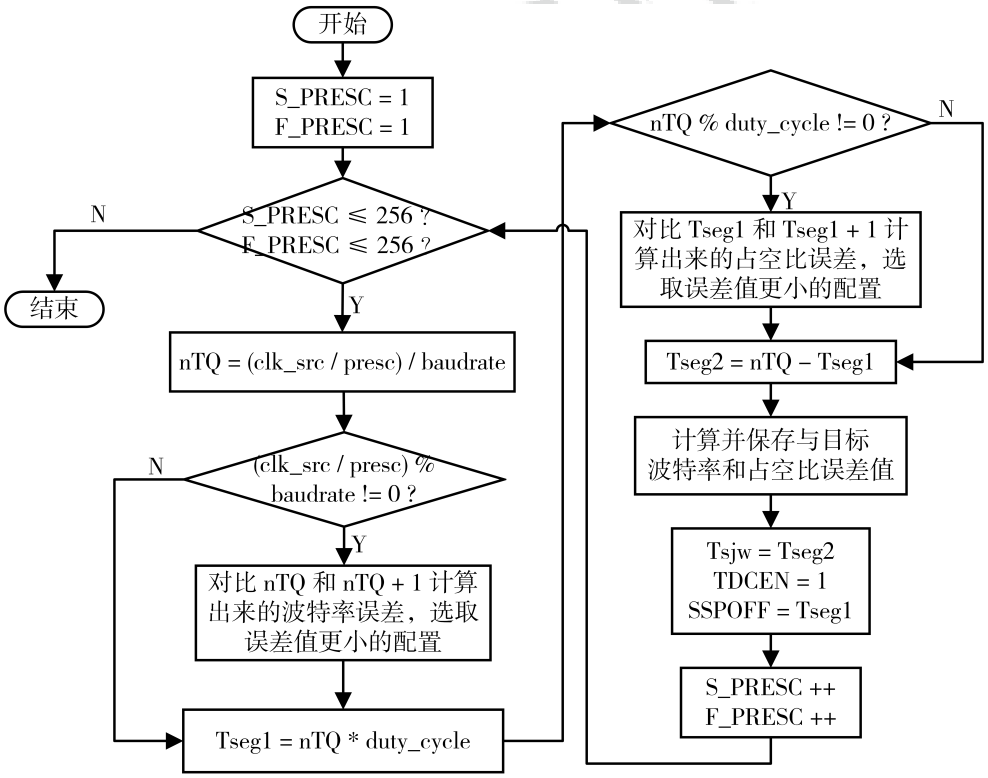


图 7–31 波特率计算

7.7.4.3. 接收过滤器

接收过滤器具体操作可以参考下列步骤：

1. 设置 SELMASK = 0，选择接收 ID 值模式。
2. 在 CANFD_ACFCTRL 中选择需进行过滤配置的对应通道序号。

   一共有十六个通道 (0 ~ 15) 。
3. 将参考目标 ID 写进CANFD_ACODEx 中。

> **注：**
>
> CANFD_ACODEx 必须在 RESET = 1 的情况下才可写入。

注：

CANFD_EN0 和 CANFD_EN1 对应使能通道 0 ~ 15，每个通道对应 1 位。通道 0 默认使能，并且 CANFD_MASKx 是全 1，表示默认接受任何帧数据。

4. 设置 SELMASK = 1，选择接收掩码模式。
5. 在 AIDEE 和 AIDE 位选择需要过滤的帧，包括下列选项：
   - 只过滤标准帧 ID
   - 只过滤扩展帧 ID
   - 两者都滤除
6. 将掩码值写进寄存器 CANFD_MASKx中。
   - 如果对应位是 1，表示该位不进行过滤，无论接收到的是 0 或 1 都可以被接收。
   - 如果对应位是 0，表示该位必须与 ACODE 中的值相同才可以被接收，否则被滤除。

注：

寄存器 CANFD_MASKx 必须在 RESET = 1 的情况下才可写入。

7.7.5. 寄存器列表

表 7–42 地址映射

模块名	基地址	地址空间
CANFD0	0x102A_0000	4 kB
CANFD1	0x102A_1000	4 kB

表 7–43 寄存器列表

偏移地址	缩写	章节
0x000	CANFD_RBUF0	0x000 CANFD_RBUF0
0x004	CANFD_RBUF1	0x004 CANFD_RBUF1
0x008~ 0x044	CANFD_RBUF2~17	0x008 ~ 0x044 CANFD_RBUF2~17
0x048~ 0x04C	CANFD_RBUF18~19	0x048 ~ 0x04C CANFD_RBUF18~19
0x050	CANFD_TBUF0	0x050 CANFD_TBUF0
0x054	CANFD_TBUF1	0x054 CANFD_TBUF1

表 7–43 寄存器列表 (续)

偏移地址	缩写	章节
0x058~ 0x094	CANFD_TBUF2~17	0x058 ~ 0x094 CANFD_TBUF2~17
0x098~ 0x09F	CANFD_TTS	0x098 ~ 0x09F CANFD_TTS
0x0A0	CANFD_CFG_STAT	0x0A0 CANFD_CFG_STAT
0x0A1	CANFD_TCMD	0x0A1 CANFD_TCMD
0x0A2	CANFD_TCTRL	0x0A2 CANFD_TCTRL
0x0A3	CANFD_RCTRL	0x0A3 CANFD_RCTRL
0x0A4	CANFD_RTIE	0x0A4 CANFD_RTIE
0x0A5	CANFD_RTIF	0x0A5 CANFD_RTIF
0x0A6	CANFD_ERRINT	0x0A6 CANFD_ERRINT
0x0A7	CANFD_LIMIT	0x0A7 CANFD_LIMIT
0x0A8	CANFD_SSEG1	0x0A8 CANFD_SSEG1
0x0A9	CANFD_ SSEG2	0x0A9 CANFD_ SSEG2
0x0AA	CANFD_SSJW	0x0AA CANFD_SSJW
0x0AB	CANFD_SPRESC	0x0AB CANFD_SPRESC
0x0AC	CANFD_FSEG1	0x0AC CANFD_FSEG1
0x0AD	CANFD_ FSEG2	0x0AD CANFD_ FSEG2
0x0AE	CANFD_FSJW	0x0AE CANFD_FSJW
0x0AF	CANFD_FPRESC	0x0AF CANFD_FPRESC
0x0B0	CANFD_EALCAP	0x0B0 CANFD_EALCAP
0x0B1	CANFD_TDC	0x0B1 CANFD_TDC
0x0B2	CANFD_RECNT	0x0B2 CANFD_RECNT
0x0B3	CANFD_TECNT	0x0B3 CANFD_TECNT

表 7–43 寄存器列表 (续)

偏移地址	缩写	章节
0x0B4	CANFD_ACFCTRL	0x0B4 CANFD_ACFCTRL
0x0B5	CANFD_TIMECFG	0x0B5 CANFD_TIMECFG
0x0B6	CANFD_EN0	0x0B6 CANFD_EN0
0x0B7	CANFD_EN1	0x0B7 CANFD_EN1
0x0B8	CANFD_ACODEx	0x0B8 CANFD_ACODEx
0x0B8	CANFD_MASKx	0x0B8 CANFD_MASKx
0x0BC~0x0BD	CANFD_VERSION	0x0BC~0x0BD CANFD_VERSION
0x0BE	CANFD_TBSLOT	0x0BE CANFD_TBSLOT
0x0BF	CANFD_TTCFG	0x0BF CANFD_TTCFG
0xC0~ 0xC3	CANFD_REFMSG[3:0]	0xC0 ~ 0xC3 CANFD_REFMSG[3:0]
0xC4	CANFD_TRIG_CFG0	0xC4 CANFD_TRIG_CFG0
0xC5	CANFD_TRIG_CFG1	0xC5 CANFD_TRIG_CFG1
0xC6~ 0xC7	CANFD_TT_TRIG [1:0]	0xC6 ~ 0xC7 CANFD_TT_TRIG [1:0]
0xC8~0xC9	CANFD_TT_WTRIG[1:0]	0xC8~ 0xC9 CANFD_TT_WTRIG[1:0]
0xD4	CANFD_DMAC	0xD4 CANFD_DMAC
0xD5	CANFD_DST_THR	0xD5 CANFD_DST_THR
0xD6	CANFD_SCR_EN	0xD6 CANFD_SCR_EN
0xD7	CANFD_SRC_THR	0xD7 CANFD_SRC_THR

7.7.6. 寄存器描述

7.7.6.1. 0x000 CANFD_RBUF0

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_RBUF0
位域	类型	默认值	描述
31	RO	0x0	ESI 错误状态指示位 对当前帧的错误状态进行识别 • 0：CANFD 节点处于主动错误模式 • 1：CANFD 节点处于被动错误模式
30:29	–	–	–
28:0	RO	0x0	ID[28:0] • 对于标准帧格式，ID[10:0] 为 11 位 ID，ID[28:11] 为保留位 • 对于扩展帧格式，为 29 位 ID

7.7.6.2. 0x004 CANFD_RBUF1

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_RBUF1
位域	类型	默认值	描述
31:16	RO	0x0	CYCLE_TIME[15:0] 时间戳循环时间 在 TTCAN 模式下，当前帧的时间戳循环时间将会存储在 RBUF 中。参考消息的循环时间总是 0。
15:13	RO	0x0	KOER 错误种类指示位 当 RBALL = 1 时，与0x0B0 CANFD_EALCAP 的 KOER 同义。

默认值：0x0000_0000			CANFD_RBUF1
位域	类型	默认值	描述
12	RO	0x0	TX 发送状态位 • 0：正常（非回环）传输模式 • 1：回环传输模式，且 CANFD 控制器接收自己发送的帧
11:8	–	–	–
7	RO	0x0	IDE ID 扩展指示位 • 0：标准帧格式，11 位 ID • 1：扩展帧格式，29 位 ID
6	RO	0x0	RTR 远程传输请求位 • 0：数据帧 • 1：远程帧 **注：** 在 CANFD 帧格式下，由于没有远程帧，RTR 位固定为 0。
5	RO	0x0	FDF CANFD 帧指示位 • 0：CAN 2.0 帧（数据段最多 8 个字节） • 1：CANFD 帧（数据段最多 64 个字节）
4	RO	0x0	BRS 位速率切换 • 0：整个帧保持正常/ 慢位速率 • 1：帧数据段和 CRC 段切换到快速率
3:0	RO	0x0	DLC 数据长度 表示当前帧的 DLC 值 对于 CAN 2.0 帧格式 • 0 ~ 7：表示当前帧包含 0 ~ 7 个数据字节 • 其它：表示当前帧包含 8 个数据字节

默认值：0x0000_0000			CANFD_RBUF1
位域	类型	默认值	描述
			对于 CANFD 帧格式 • 0 ~ 8：表示当前帧包含 0 ~ 8 个数据字节 • 9：表示当前帧包含 12 个数据字节 • 10：表示当前帧包含 16 个数据字节 • 11：表示当前帧包含 20 个数据字节 • 12：表示当前帧包含 24 个数据字节 • 13：表示当前帧包含 32 个数据字节 • 14：表示当前帧包含 48 个数据字节 • 15：表示当前帧包含 64 个数据字节

7.7.6.3. 0x008 ~ 0x044 CANFD_RBUF2~17

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_RBUF2~17
位域	类型	默认值	描述
511:0	RO	0x0	DATA 接收帧数据 当前接收帧数据，最多 64 个字节数据，按小端模式存储，低位字节偏移地址存储低位字节数据，高位字节偏移地址存储高位字节数据。

7.7.6.4. 0x048 ~ 0x04C CANFD_RBUF18~19

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x00000000			CANFD_RBUF18~19
位域	类型	默认值	描述
63:0	RO	0x0	RTS 接收时间戳

默认值：0x00000000			CANFD_RBUF18~19
位域	类型	默认值	描述
			CiA603 模式下，在 RBUF 地址范围尾部存储接收时间戳，按小端模式存储，低位字节偏移地址存储低位字节数据，高位字节偏移地址存储高位字节数据。

7.7.6.5. 0x050 CANFD_TBUF0

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_TBUF0
位域	类型	默认值	描述
31	R/W	0x0	TTSEN 发送时间戳使能 对于 CiA603 时间戳，选择是否获取时间戳 (TTS) • 0：对帧没有时间戳更新需求 • 1：TTS 保存最后传输帧时间戳
30:29	–	–	–
28:0	R/W	0x0	ID[28:0] • 对于标准帧模式，ID[10:0] 为 11 位 ID，ID[28:11] 为保留位 • 对于扩展帧模式，为 29 位 ID

7.7.6.6. 0x054 CANFD_TBUF1

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_TBUF1
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	IDE ID 扩展指示

默认值：0x0000_0000			CANFD_TBUF1
位域	类型	默认值	描述
			• 0：标准格式帧：11 位 ID • 1：扩展格式帧：29 位 ID
6	R/W	0x0	RTR 远程传输请求 • 0：数据帧 • 1：远程帧 **注：** 对于 CANFD 模式，由于没有远程帧，RTR 位固定配置 0。
5	R/W	0x0	FDF CANFD 帧指示 • 0：CAN 2.0 帧（数据段最多 8 个字节） • 1：CANFD 帧（数据段最多 64 个字节）
4	R/W	0x0	BRS 位速率切换 • 0：整个帧保持正常/慢位速率 • 1：帧数据段和 CRC 段切换到快速率
3:0	R/W	0x0	DLC 数据长度 表示当前帧的 DLC 值 对于 CAN 2.0 帧格式 • 0 ~ 7：表示当前帧包含 0 ~ 7 个数据字节 • 其它：表示当前帧包含 8 个数据字节 对于 CANFD 帧格式 • 0 ~ 8：表示当前帧包含 0 ~ 8 个数据字节 • 9：表示当前帧包含 12 个数据字节 • 10：表示当前帧包含 16 个数据字节 • 11：表示当前帧包含 20 个数据字节 • 12：表示当前帧包含 24 个数据字节 • 13：表示当前帧包含 32 个数据字节

默认值：0x0000_0000			CANFD_TBUF1
位域	类型	默认值	描述
			• 14：表示当前帧包含 48 个数据字节 • 15：表示当前帧包含 64 个数据字节

7.7.6.7. 0x058 ~ 0x094 CANFD_TBUF2~17

注：

只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_TBUF2~17
位域	类型	默认值	描述
511:0	R/W	0x0	DATA 发送帧数据 当前发送帧数据，最多 64 个字节数据，按小端模式存储，低位字节偏移地址存储低位字节数据，高位字节偏移地址存储高位字节数据。

7.7.6.8. 0x098 ~ 0x09F CANFD_TTS

默认值：0x00			CANFD_TTS
位域	类型	默认值	描述
63:0	RO	0x0	TTS 传输时间戳 TTS 用来保存 CiA 603 的最后传输帧的时间戳。 注： 如果 TTSEN = 1，每个新帧都会覆盖 TTS。

7.7.6.9. 0x0A0 CANFD_CFG_STAT

默认值：0x80			CANFD_CFG_STAT
位域	类型	默认值	描述
7	R/W	0x1	RESET 复位控制

默认值：0x80			CANFD_CFG_STAT
位域	类型	默认值	描述
			• 0：CANFD 控制器无本地复位 • 1：执行 CANFD 控制器的本地复位 **注：** 如果 RESET 设置为 1 并立即设置为 0，则需要一些时间才能将 RESET 读取为 0 并释放复位状态。
6	R/W	0x0	LBME Loop Back Mode External，外部回环 • 0：禁用 • 1：启用 当处于传输活动状态时，不能启用 LBME。
5	R/W	0x0	LBMI Loop Back Mode Internal，内部回环 • 0：禁用 • 1：启用 当处于传输活动状态时，不能启用 LBMI。
4	R/W	0x0	TPSS Transmission Primary Single Shot mode for PTB，PTB 单发模式功能 • 0：禁用 • 1：启用
3	R/W	0x0	TSSS Transmission Primary Single Shot mode for STB，STB 单发模式功能 • 0：禁用 • 1：启用
2	R	0x0	RACTIVE Reception ACTIVE，接收状态位 • 0：控制器当前没有接收活动 • 1：控制器当前正在接收帧
1	R	0x0	TACTIVE Transmission ACTIVE，发送状态位

默认值：0x80			CANFD_CFG_STAT
位域	类型	默认值	描述
			• 0：控制器当前无传输活动 • 1：控制器当前正在发送一个帧
0	R/W	0x0	BUSOFF Bus Off，总线断开 • 0：控制器总线状态连接 • 1：控制器总线状态断开 **注：** BUSOFF 置位将会清零 TECNT 和 RECNT 。

7.7.6.10. 0x0A1 CANFD_TCMD

注：

同时置位 TSONE 和 TSALL 是没有意义的，若同时置位 TSONE 和 TSALL ，则 TSALL 获胜并且 TSONE 被自动清零。

默认值：0x00			CANFD_TCMD
位域	类型	默认值	描述
7	R/W	0x0	TBSEL Transmit Buffer Select，发送缓冲区选择 • 0：选择 PTB (高优先级缓冲区) • 1：选择 STB 如果（TTEN = 1 且 TTTBM = 1），该位将重置为 0。
6	R/W	0x0	LOM Listen Only Mode，监听模式 • 0：禁用 • 1：启用

默认值：0x00			CANFD_TCMD
位域	类型	默认值	描述
			注： • 若已置位 TPE、TSONE 或 TSALL，则无法置位 LOM。 • LOM = 1 和 LBME = 0 禁用所有传输。 • LOM = 1 和 LBME = 1 禁用响应其它接收帧和错误帧的 ACK，但允许 ACK 响应传输自己的帧。
5	R/W	0x0	STBY Transceiver Standby Mode，收发器休眠模式 • 0：禁用 • 1：启用 当 STBY 模式启用时，外部 CANFD_STB 引脚输出高电平。 当 STBY 模式禁用时， 外部 STB 引脚输出低电平。 注： 若 TPE = 1，TSONE = 1 或者 TSALL = 1 的时候不能置位 STBY。
4	R/ WAC	0x0	TPE Transmit Primary Enable，使能 PTB 发送 • 0：PTB 无传输 • 1：PTB 传输使能 如果置位 TPE，则 PTB 将在当前传输完成后的下一个空闲时间点开始传输。 置位后，TPE 会等到消息已成功传输或使用 TPA 中止后才自动清零，控制器无法通过写 0 让其重置为 0。 如果 RESET = 1、STBY = 1、（LOM = 1 且 LBME = 0）或（TTEN = 1 且 TTTBM = 1），该位将重置为 0。
3	R/ WAC	0x0	TPA Transmit Primary Abort，中止当前 PTB 进行发送 • 0：不中止 • 1：中止来自 PTB 的传输，该传输已由 TPE = 1 请求但尚未开始发送。（消息仍保留在 PTB 中） 注： • 置位 TPA 会自动使 TPE 无效。 • 可以将 TPA 置位，但无法通过写 0 让其重置为 0。

默认值：0x00			CANFD_TCMD
位域	类型	默认值	描述
			• 若 RESET = 1 或（TTEN = 1 且 TTTBM = 1），该位将重置为 0。 • TPA 不应与 TPE 同时置位。
2	R/ WAC	0x0	TSONE Transmit Secondary ONE frame，使能 STB 发送单帧消息 • 0：STB 无传输 • 1：STB 中单帧传输使能。 ◦ 在 FIFO 模式下，发送最早的消息。 ◦ 在优先级模式下，发送具有最高优先级的消息。 置位后，TSONE 会等到消息成功传输或使用 TSA 中止后才自动清零。 控制器可以将 TSONE 置位，但不能将其重置为 0。 出现以下任意情况，该位将重置为 0： • RESET = 1 • STBY = 1 • LOM = 1 且 LBME = 0 • TTEN = 1 且 TTTBM = 1
1	R/ WAC	0x0	TSALL Transmit Secondary ALL frame，使能 STB 发送所有帧消息 • 0：STB 无传输 • 1：STB 中所有消息的传输使能 置位后，TSALL 会等直到所有消息都已成功传输或使用 TSA 中止后才自动清零。 控制器可以将 TSALL 置位，但不能将其重置为 0。#unique_153_Connect_42_p_kcy_dmx_dbc **注：** 如果在传输期间 STB 加载了新帧，则新帧也将被传输，表示当 STB 变空时，由 TSALL 发起的传输结束。
0	R/ WAC	0x0	TSA Transmit Secondary Abort，中止 STB 发送操作 • 0：不中止 • 1：中止已请求但尚未开始的 STB 的传输 对于 TSONE 传输，只有一帧被中止。而对于 TSALL 传输，所有帧都被中止，并且所有中止的消息都将丢失，TSSTAT 也会相应更新状态。

默认值：0x00			CANFD_TCMD
位域	类型	默认值	描述
			控制器可以将 TSA 置位，但不能将其重置为 0，置位 TSA 自动分别使 TSONE 或 TSALL 无效。 如果 RESET = 1，该位将被重置为 0。

7.7.6.11. 0x0A2 CANFD_TCTRL

默认值：0x90			CANFD_TCTRL
位域	类型	默认值	描述
7	R/W	0x1	FD_ISO CAN FD ISO mode CANFD 模式选择 • 0：选择 Bosch CAN FD （非 ISO）模式 • 1：选择 ISO CAN FD (ISO 11898–1:2015) 模式 ISO CANFD 模式具有不同的 CRC 初始化值和额外的填充位计数。 注： • 两种模式不兼容，不得在一个 CAN 网络中混合使用。 • 该位对 CAN2.0 帧格式没有影响。 • 该位仅在 RESET = 1 时可写。
6	R/ WAC	0x0	TSNEXT Transmit buffer Secondary NEXT，STB 下一个时隙选择 • 0：无动作 • 1：STB 时隙已满，选择下一个时隙 在帧的所有信息都写入 TBUF 寄存器后，控制器必须置位 TSNEXT 以发出该时隙已满的信号。由 CANFD 控制器内核将 TBUF 寄存器连接到下一个时隙。一旦一个时隙被标记为已满，就可以使用 TSONE 或 TSALL 开始传输。 • 若 TBSEL = 0，则置位 TSNEXT 不起任何作用，该位被自动清零。 • 若 TBSEL = 1，且 STB 的所有时隙都已满，则 TSNEXT 保持置位直到有时隙空闲。 • TSNEXT 在 TTCAN 模式下没有意义，固定为 0。
5	R/W	0x0	TSMODE Transmit buff Secondary operation mode，STB 操作模式 仅当 STB 为空时才能切换 TSMODE。

默认值：0x90			CANFD_TCTRL
位域	类型	默认值	描述
			• 0：FIFO 模式，帧按照写入 STB 的顺序传输。 • 1：优先级模式，STB 中具有最高优先级的帧自动首先传输。
4	R/W	0x1	TTTBM TTCAN Transmit Buffer Mode，TTCAN 发送缓存模式 0：分离 PTB 和 STB，行为由 TSMODE 定义。 1：完整的 TTCAN 支持。缓冲时隙可由 TBPTR 和 TTPTR 选择。 • 当 TTEN = 0 时，为非 TTCAN 场景，此时 TTTBM 无效，系统提供 PTB 和 STB，STB 的行为由 TSMODE 定义。 • 当 TTEN = 1，且需要完全支持包括时间触发传输在内的所有功能的 TTCAN 场景，则需要置位 TTTBM，此时所有 TB 时隙都可以使用 TTPTR 和 TBPTR 寻址。 • 当 TTEN = 1，且只支持接收时间戳的 TTCAN 场景，则可以清零 TTTBM，此时发送缓冲区就像在事件驱动模式下一样工作，并且可以通过 TSMODE 选择行为。 • 仅当 TBUF 为空时才切换 TTTBM 。
3:2	–	–	–
1:0	RO	0x0	TSSTAT Transmission Secondary STATUS bits，辅发送缓存器状态 如果 TTEN = 0 或者 TTTBM = 0: • 0x0：STB 为空 • 0x1：STB 少于或等于一半缓存空间 • 0x2：STB 大于一半缓存空间 • 0x3：STB 已满 如果 TTEN = 1 或者 TTTBM = 1: • 0x0：PTB 和 STB 为空 • 0x1：PTB 和 STB 非空也非满 • 0x3：PTB 和 STB 已满

7.7.6.12. 0x0A3 CANFD_RCTRL

默认值：0x00			CANFD_RCTRL
位域	类型	默认值	描述
7	R/W	0x0	SACK Self-ACKnowledge，自我应答

默认值：0x00			CANFD_RCTRL
位域	类型	默认值	描述
			• 0：无自我应答 • 1：当 LBME = 1 时产生自我应答
6	R/W	0x0	ROM Receive buff Overflow Mode，接收缓冲器溢出模式 若收到新消息时接收缓冲器已满，则接收缓冲器选择： • 0：最早的消息将被覆盖 • 1：新消息将被丢弃
5	R	0x0	ROV Receive buffer Overflow，接收缓冲器溢出状态 • 0：未溢出 • 1：已溢出 **注：** RREL 写 1 清零 ROV。
4	R/W	0x0	RREL Receive buffer Release，接收缓冲器释放 • 0：释放当前接收缓冲器的时隙 • 1：不释放当前接收缓冲器的时隙 主机读取接收缓冲器某时隙的帧消息后，置位 RREL 使控制器指向下一时隙帧消息，同时 RSTAT 更新状态。
3	R/W	0x0	RBALL Receive Buffer stores ALL data frames，接收缓冲器接收所有帧 • 0：正常接收（不包含有错误的帧消息） • 1：接收所有（包括有错误）的帧消息
2	–	–	–
1:0	R	0x0	RSTAT Receive buffer STATUS，接收缓冲器状态 • 0x0：空 • 0x1：非空，但是未达到 AFWL 设置的触发阈值

默认值：0x00			CANFD_RCTRL
位域	类型	默认值	描述
			• 0x2：大于等于 AFWL 设置的阈值，但还没满 • 0x3：已满

7.7.6.13. 0x0A4 CANFD_RTIE

默认值：0xFE			CANFD_RTIE
位域	类型	默认值	描述
7	R/W	0x1	RIE Receive Interrupt Enable，接收中断使能 • 0：禁用 • 1：启能
6	R/W	0x1	ROIE RB Overrun Interrupt Enable，接收缓冲器溢出中断使能 • 0：禁用 • 1：启能
5	R/W	0x1	RFIE RB Full Interrupt Enable，接收缓冲器已满中断使能 • 0：禁用 • 1：启能
4	R/W	0x1	RAFIE RB Almost Full Interrupt Enable，接收缓冲器快满中断使能 • 0：禁用 • 1：启能
3	R/W	0x1	TPIE Transmission Primary Interrupt Enable，PTB 传输中断使能 • 0：禁用 • 1：启能
2	R/W	0x1	TSIE Transmission Secondary Interrupt Enable，STB 传输中断使能

默认值：0xFE			CANFD_RTIE
位域	类型	默认值	描述
			• 0：禁用 • 1：启能
1	R/W	0x1	EIE Error Interrupt Enable，错误中断使能 • 0：禁用 • 1：启能
0	R	0x0	TSFF • 当 TTEN = 0 和 TTTBM = 0：Transmit Secondary buffer Full Flag，STB 满标志位 ◦ 0：STB 中未填充到最大数量的消息 ◦ 1：STB 中已填充最大数量的消息 • 当 TTEN = 1 和 TTTBM = 1：Transmit buffer Slot Full Flag，发送缓冲器时隙已满标志位 ◦ 0：TBPTR 指向的缓存时隙是空的 ◦ 1：TBPTR 指向的缓存时隙已填充满

7.7.6.14. 0x0A5 CANFD_RTIF

注：

标志位只有在对应的使能位开启后才能被写 1 清零。

默认值：0x00			CANFD_RTIF
位域	类型	默认值	描述
7	R/W1C	0x0	RIF Receive Interrupt Flag，接收缓冲器接收中断标志位 • 0：未收到帧消息 • 1：收到可用帧消息并已存在接收缓冲器中
6	R/W1C	0x0	ROIF RB Overrun Interrupt Flag，接收缓冲器溢出中断标志位 • 0：接收缓冲器未溢出，没有帧消息被覆盖 • 1：接收缓冲器已溢出，至少有一条接收到的帧消息被覆盖

默认值：0x00			CANFD_RTIF
位域	类型	默认值	描述
			注： 当 ROIF = 1 时，RFIF 也会置位
5	R/W1C	0x0	RFIF RB Full Interrupt Flag，接收缓冲器接收满中断标志位 • 0：接收缓冲器未满 • 1：接收缓冲器所有时隙都已满
4	R/W1C	0x0	RAFIF RB Almost Full Interrupt Flag，接收缓冲器接收阈值中断标志位 • 0：未超过 AFWL 设置的阈值 • 1：超过或等于 AFWL 设置的阈值
3	R/W1C	0x0	TPIF Transmission Primary Interrupt Flag，PTB 发送完成中断标志位 • 0：PTB 请求的发送未完成 • 1：PTB 请求的发送已完成 **注：** TTCAN 模式下，该位无效。
2	R/W1C	0x0	TSIF Transmission Secondary Interrupt Flag，STB 发送完成中断标志位 • 0：STB 请求的发送未完成 • 1：STB 请求的发送已完成 **注：** TTCAN 模式下，无论消息存储在什么位置，TSIF 都表示所有成功的传输。
1	R/W1C	0x0	EIF ERROR Interrupt Flag，错误中断标志位 • 0：未发生变化 • 1：超过错误警告阈值或者 BUSOFF 发生变化
0	R/W1C	0x0	AIF Abort Interrupt Flag，中止中断标志位

默认值：0x00			CANFD_RTIF
位域	类型	默认值	描述
			• 0：没有执行任何中止 • 1：发生 TPA 或者 TSA 中止 注： • AIF 没有对应的使能位。 • 不推荐同时置位 TPA 和 TSA，因为他们都使能 AIF。

7.7.6.15. 0x0A6 CANFD_ERRINT

注：

标志位只有在对应的使能位开启后才能被写 1 清零。

默认值：0x00			CANFD_ERRINT
位域	类型	默认值	描述
7	R	0x0	EWARN Error WARNING limit reached，错误警告达到门限 • 0：RECNT 或者 TECNT 未达到 EWL 门限。 • 1：RECNT 或者 TECNT 等于或超过 EWL。
6	R	0x0	EPASS Error Passive mode active，被动错误模式 • 0：未激活，主动错误模式 • 1：激活，被动错误模式
5	R/W	0x0	EPIE Error Passive Interrupt Enable，错误被动模式中断使能 • 0：禁用 • 1：启用
4	R/W1C	0x0	EPIF Error Passive Interrupt Flag，被动错误中断标志位 • 0：未发生中断 • 1：从主动错误变为被动错误或者从被动错误变为主动错误

默认值：0x00			CANFD_ERRINT
位域	类型	默认值	描述
3	R/W	0x0	ALIE Arbitration Lost Interrupt Enable，仲裁丢失中断使能 • 0：禁用 • 1：启用
2	R/W1C	0x0	ALIF Arbitration Lost Interrupt Flag，仲裁丢失中断标志位 • 0：未丢失 • 1：丢失
1	R/W	0x0	BEIE Bus Error Interrupt Enable，总线错误中断使能 • 0：禁用 • 1：启用
0	R/W1C	0x0	BEIF Bus Error Interrupt Flag，总线错误中断标志位 • 0：未发生错误 • 1：发生错误

7.7.6.16. 0x0A7 CANFD_LIMIT

默认值：0x1B			CANFD_LIMIT
位域	类型	默认值	描述
7：4	R/W	0x01	AFWL receive buffer Almost Full Warning Limit，接收缓冲器快满警告门限阈值 当接收缓冲器中的数据层数大于等于 AFWL 后会触发 RAFIF 中断。 • 0~1：AFWL = 1 • 2~7：AFWL = 2 ~ 7 • ≥8：AFWL = 8 **注：** 当 AFWL 设置为 8 的时候，实际上会触发 RFIF 中断。

默认值：0x1B			CANFD_LIMIT
位域	类型	默认值	描述
3：0	R/W	0x0B	EWL Programmable Error Warning Limit，可编程错误警告门限 • 错误警告门限 = (EWL + 1) × 8

7.7.6.17. 0x0A8 CANFD_SSEG1

默认值：0x03			CANFD_SSEG1
位域	类型	默认值	描述
7:0	R/W	0x3	S_Seg_1 Bit Timing Segment 1 慢速模式位时序段 1 实际采样时间点设置在帧开始后 Tseg1 = (S_Seg_1 + 2) × TQ (Time Quanta) **注：** 当 RESET = 1 时可写。 **注：** TQ=Prescaler/system_clock

7.7.6.18. 0x0A9 CANFD_ SSEG2

默认值：0x02			CANFD_ SSEG2
位域	类型	默认值	描述
7	–	–	–
6:0	R/W	0x02	S_Seg_2 Bit Timing Segment 2 慢速模式位时序段 2 采样点后 Tseg2 = (S_Seg_2 + 1) × TQ (Time Quanta) **注：** 当 RESET = 1 时可写。

7.7.6.19. 0x0AA CANFD_SSJW

默认值：0x02			CANFD_SSJW
位域	类型	默认值	描述
7	–	–	–
6:0	R/W	0x2	S_SJW Synchronization Jump Width，慢速模式重新同步调整宽度 Tsjw = (S_SJW + 1) × TQ (Time Quanta) 该时间表示用于重新同步的最大的加长或者缩短位时序时间 **注：** 当 RESET = 1 时可写。

7.7.6.20. 0x0AB CANFD_SPRESC

默认值：0x01			CANFD_SPRESC
位域	类型	默认值	描述
7:0	R/W	0x1	S_PRESC Prescaler，慢速模式下时钟预分频因子 实际分频因子 = S_PRESC + 1 **注：** 当 RESET = 1 时可写。

7.7.6.21. 0x0AC CANFD_FSEG1

默认值：0x03			CANFD_FSEG1
位域	类型	默认值	描述
7:5	–	–	–
4:0	R/W	0x3	F_Seg_1 Bit Timing Segment 1 快速模式位时序段 1 实际采样时间点设置在帧开始后 Tseg1 = (F_Seg_1 + 2) × TQ (Time Quanta)

默认值：0x03			CANFD_FSEG1
位域	类型	默认值	描述
			注： 当 RESET = 1 时可写。

7.7.6.22. 0x0AD CANFD_ FSEG2

默认值：0x02			CANFD_ FSEG2
位域	类型	默认值	描述
7:4	–	–	–
3:0	R/W	0x02	F_Seg_2 Bit Timing Segment 2 快速模式位时序段 2 采样点后 Tseg2 = (F_Seg_2 + 1) × TQ (Time Quanta) 注： 当 RESET = 1 时可写。

7.7.6.23. 0x0AE CANFD_FSJW

默认值：0x02			CANFD_FSJW
位域	类型	默认值	描述
7:4	–	–	–
3:0	R/W	0x2	F_SJW Synchronization Jump Width 快速模式重新同步调整宽度 Tsjw = (F_SJW + 1) × TQ (Time Quanta) 该时间表示用于重新同步的最大的加长或者缩短位时序时间 注： 当 RESET = 1 时可写。

7.7.6.24. 0x0AF CANFD_FPRESC

默认值：0x01			CANFD_FPRESC
位域	类型	默认值	描述
7:0	R/W	0x1	F_PRESC Prescaler 快速模式下时钟预分频因子 实际分频因子 = F_PRESC + 1 **注：** 当 RESET = 1 时可写。

7.7.6.25. 0x0B0 CANFD_EALCAP

默认值：0x00			CANFD_EALCAP
位域	类型	默认值	描述
7:5	R	0x0	KOER Kind of Error 错误类型 • 0x0：无错误 • 0x1：BIT ERROR 位错误 • 0x2：FORM ERROR 格式错误 • 0x3：STUFF ERROR 填充位错误 • 0x4：ACKNOWLEDGEMENT ERROR ACK 应答错误 • 0x5：CRC ERROR CRC 冗余校验位错误 • 0x6：其它错误 • 0x7：未使用 **注：** KOER 随每个新错误而更新。因此，当帧被成功传输或接收时，它保持不变。
4:0	R	0x0	ALC Arbitration Lost Capture 仲裁丢失捕捉位位置 • 0x00~0x0A：1~11 位标识符仲裁丢失，对应标准格式（ID10~ ID0）扩展格式（ID28~ ID18） • 0x0B：SRTR 仲裁丢失 • 0x0C：IDE 仲裁丢失

默认值：0x00			CANFD_EALCAP
位域	类型	默认值	描述
			• 0x0D~0x1E：12~29 位标识符仲裁丢失，对应扩展格式（ID17~ ID0） • 0x1F：RTR 仲裁丢失

7.7.6.26. 0x0B1 CANFD_TDC

注：

0x0B1 CANFD_TDC当 RESET = 1 时可写。

默认值：0x00			CANFD_TDC
位域	类型	默认值	描述
7	R/W	0x0	TDCEN Transmitter Delay Compensation Enable 发送延迟补偿使能 当在 CANFD 帧中 TDCEN = 1，BRS 处于活动状态后，则 TDC 功能将在数据阶段被激活。
6:0	R/W	0x0	SSPOFF Secondary Sample Point Offset 第二采样点偏移 传输器延迟加上 SSPOFF 后定义了第二采样点的时间。SSPOFF 以 TQ 的数量给出。

7.7.6.27. 0x0B2 CANFD_RECNT

默认值：0x00			CANFD_RECNT
位域	类型	默认值	描述
7:0	R	0x0	RECNT Receive Error Count，接收错误计数器 RECNT 按照 CAN 规范中的定义递增和递减。

7.7.6.28. 0x0B3 CANFD_TECNT

默认值：0x00			CANFD_TECNT
位域	类型	默认值	描述
7:0	R	0x0	TECNT Transmit Error Count，发送错误计数器

默认值：0x00			CANFD_TECNT
位域	类型	默认值	描述
			TECNT 按照 CAN 规范中的定义递增和递减。

7.7.6.29. 0x0B4 CANFD_ACFCTRL

默认值：0x00			CANFD_ACFCTRL
位域	类型	默认值	描述
7:6	–	–	–
5	R/W	0x0	SELMASK Select acceptance MASK 选择接收掩码 • 0：0x0B8 CANFD_ACODEx用作接收 ID 值 • 1：0x0B8 CANFD_MASKx用作接收掩码
4	–	–	–
3:0	R/W	0x0	ACFADR Acceptance filter address，接收过滤器地址 • 0 ~ 15 表示 0 ~ 15 一共 16 个通道接收过滤器

7.7.6.30. 0x0B5 CANFD_TIMECFG

默认值：0x02			CANFD_TIMECFG
位域	类型	默认值	描述
7:2	–	–	–
1	R/W	0x1	TIMEPOS TIME–stamping Position，CiA 603 时间戳位置 • 0：SOF • 1：EOF TIMEPOS 只能当 TIMEEN = 0 时可被更改。 **注：** 可以在同一个寄存器写操作时间点同时设置 TIMEEN = 1 和修改 TIMEPOS。
0	R/W	0x0	TIMEEN

默认值：0x02			CANFD_TIMECFG
位域	类型	默认值	描述
			TIME-stamping enable，CiA 603 时间戳使能 • 0：禁止 • 1：使能

7.7.6.31. 0x0B6 CANFD_EN0

默认值：0x01			CANFD_EN0
位域	类型	默认值	描述
7:0	R/W	0x01	AE_x Acceptance Filter Enable0，接收过滤器通道使能低 8 位 • 0：未使能 • 1：使能 每一位代表每一个接收过滤器通道，位 0 ~ 7 分别代表通道 0 ~ 7。

7.7.6.32. 0x0B7 CANFD_EN1

默认值：0x00			CANFD_EN1
位域	类型	默认值	描述
7:0	R/W	0x00	AE_x Acceptance Filter Enable0，接收过滤器通道使能高 8 位 • 0：未使能 • 1：使能 每一位代表每一个接收过滤器通道，位 0 ~ 7 分别代表通道 8 ~ 15。

7.7.6.33. 0x0B8 CANFD_ACODEx

注：

对于寄存器0x0B8 CANFD_ACODEx

- 当 RESET = 1 时可写。
- 只能是 32 位访问，不能使用 8 或者 16 位模式访问。

默认值：0x0000_0000			CANFD_ACODEx
位域	类型	默认值	描述
31:29	–	–	–
28:0	R/W	0x0	ACODE_x Acceptance CODE，接收过滤 ID 值 • 将接收到的帧 ID 值与对应位的 ID 值进行对比 ACODE_x[10:0] 用于标准帧。 ACODE_x[28:0] 用于扩展帧。

7.7.6.34. 0x0B8 CANFD_MASKx

注：

对于寄存器 0x0B8 CANFD_MASKx：

- 当 RESET = 1 时可写。
- 只能是 32 位访问，不能使用 8 或者 16 位模式访问。

。

默认值：0x0000_0000			CANFD_MASKx
位域	类型	默认值	描述
31	–	–	–
30	R/W	0x0	AIDEE Acceptance mask IDE bit check enable，IDE 检查使能 • 0：接收过滤器接收两种类型帧 • 1：接收过滤器接收 AIDE 指定的类型帧
29	R/W	0x0	AIDE Acceptance mask IDE bit value，IDE 掩码值 • 0：接收过滤器只接收标准帧 • 1：接收过滤器只接收扩展帧
28:0	R/W	0x0	AMASK_x Acceptance MASK，接收过滤器掩码值 • 0：该比特位需要与 ACODE_x 对应位对比 • 1：该比特位不需要与 ACODE_x 对应位对比

默认值：0x0000_0000			CANFD_MASKx
位域	类型	默认值	描述
			AMASK_x[10:0] 用于标准帧。 AMASK_x[28:0] 用于扩展帧。

7.7.6.35. 0x0BC~0x0BD CANFD_VERSION

默认值：0x0001			CANFD_VERSION
位域	类型	默认值	描述
15:8	R	0x01	MAJOR VERSION 主版本号
7:0	R	0x01	MINOR VERSION 次版本号

7.7.6.36. 0x0BE CANFD_TBSLOT

默认值：0x00			CANFD_TBSLOT
位域	类型	默认值	描述
7	R/W	0x0	TBE set TB slot to Empty，选择发送缓冲器时隙设置为空状态 • 0：无操作 • 1：选择发送缓冲器时隙标记为空状态 注： • 置位 TPE 后，若 TSFF = 0，则 TPE 位会自清零。 • 置位 TPE 后，若此时发送还在进行中，则控制器会等到发送完毕后，或者传输错误，或者仲裁丢失才自动清零。 • 如果 TBF 和 TBE 同时置位，则 TBE 生效而 TPF 无效。
6	R/W	0x0	TBF set TB slot to Filled，选择发送缓冲器时隙设置为满状态 • 0：无操作 • 1：选择发送缓冲器时隙标记为满状态

默认值：0x00			CANFD_TBSLOT
位域	类型	默认值	描述
			注： • 置位 TPF 后，若 TSFF = 1，则 TPF 位会自清零。 • 如果 TBF 和 TBE 同时置位，则 TBE 生效而 TPF 无效。
5:0	R/W	0x0	TBPTR Pointer to a TB message slot，指向发送缓冲器的缓冲区时隙指针 • 0：指向 PTB 中的时隙的指针 • 1 ~ 4：指向 STB 的时隙的指针 TBPTR 指向的消息时隙可以使用 TBUF 进行读写，可以通过 TBF 或 TBE 标记满或者空状态。 注： 若 TBPTR 大于等于 5，则 TPF 和 TPE 同时清零，并且不会发生任何动作。

7.7.6.37. 0x0BF CANFD_TTCFG

默认值：0x90			CANFD_TTCFG
位域	类型	默认值	描述
7	R/W	0x1	WTIE Watch Trigger Interrupt Enable，观测触发中断使能 • 0：未使能 • 1：使能
6	R/W1C	0x0	WTIF Watch Trigger Interrupt Flag，观测触发中断标志位 当周期计数到达 TT_WTRIG 定义的阈值以及 WTIE 置位时 WTIF 会置位。
5	R/W1C	0x0	TEIF Trigger Error Interrupt Flag，触发错误中断标志位。 注： TEIF 没有对应的使能或者禁用的中断使能位。
4	R/W	0x1	TTIE Time Trigger Interrupt Enable，时间触发中断使能

默认值：0x90			CANFD_TTCFG
位域	类型	默认值	描述
			• 0：未使能 • 1：使能
3	R/W1C	0x0	TTIF Time Trigger Interrupt Flag，时间触发中断标志位 当 TTIE = 1 时，若 cycle time 循环时间等于 TT_TRIG ，则将置位 TTIF。 注： TTIF 只会被置位 1 次，如果 TT_TRIG 没有更新，则在下一个基础周期内 TTIF 不会被再次置位。
2:1	R/W	0x0	T_PRESC TTCAN Time Prescaler，定时器预分频器 • 0：1 • 1：2 • 2：4 • 3：8 注： T_PRESC 只能在 TTEN = 0 时被修改，但可以在同一个写操作时间点同时修改 T_PRESC 和设置 TTEN = 1。
0	R/W	0x0	TTEN Time Trigger Enable，时间触发功能使能 • 0：未使能 • 1：TTCAN 使能，计时器运行

7.7.6.38. 0xC0 ~ 0xC3 CANFD_REFMSG[3:0]

默认值：0x0000_0000			CANFD_REFMSG[3:0]
位域	类型	默认值	描述
31	R/W	0x0	REF_IDE Reference message IDE bit，参考消息 IDE 位
30:29	–	–	–

默认值：0x0000_0000			CANFD_REFMSG[3:0]
位域	类型	默认值	描述
28:0	R/W	0x0	REF_ID Reference message Identifier，参考消息 ID 值 • 当 REF_IDE = 0，则 REF_ID[10:0] 有效（标准 ID ） • 当 REF_IDE = 1，则 REF_ID[28:0] 有效（扩展 ID ） REF_ID 用于在 TTCAN 模式下检测接收或者发送的参考消息。如果参考消息被检测到，并且没有错误，则该帧的 Sync_Mark 同步标记就会变成 Ref_Mark 参考标记。 REF_ID[2:0] 会被强制清零，表示可以支持 8 个时间主机。

7.7.6.39. 0xC4 CANFD_TRIG_CFG0

默认值：0x00			CANFD_TRIG_CFG0
位域	类型	默认值	描述
7:6	–	–	–
5:0	R/W	0x0	TTPTR Transmit Trigger TB Slot Pointer，发送触发 TB 时隙指针 • 0：指向 PTB 中的时隙的指针 • 1 ~ 4：指向 STB 的时隙的指针 注： • 如果 TTPTR 大于等于 5，则 TEIF 会被置位，并且 0xC5 CANFD_TRIG_CFG1写访问后无法激活新触发器。 • 若 TTPTR 指向一个空时隙，则当触发时间到达的那一刻，TEIF 也会置位。

7.7.6.40. 0xC5 CANFD_TRIG_CFG1

默认值：0x00			CANFD_TRIG_CFG1
位域	类型	默认值	描述
7:4	R/W	0x0	TEW Transmit Enable Window，发送使能窗口 对于单次发送触发器，允许帧开始的周期时间最多为 16 个节拍。 TEW + 1 定义了节拍数。

默认值：0x00			CANFD_TRIG_CFG1
位域	类型	默认值	描述
			注： TEW = 0 表示发送使能窗口缩短为 一个节拍。
3	–	–	–
2:0	R/W	0x0	TTYPE Trigger Type，触发器类型 • 0：立即触发器，用于立即发送 • 1：时间触发器，用于接收触发 • 2：单次发送触发器，用于独享时间窗口 • 3：发送开始触发器，用于合并仲裁时间窗口 • 4：发送停止触发器，用于合并仲裁时间窗口 • 其它：无操作 触发时间由 TT_TRIG 定义。 TTPTR 为发送触发器选择 TB 时隙。

7.7.6.41. 0xC6 ~ 0xC7 CANFD_TT_TRIG [1:0]

默认值：0x0000			CANFD_TT_TRIG [1:0]
位域	类型	默认值	描述
15:0	R/W	0x0	TT_TRIG Trigger Time，触发时间 TT_TRIG[15:0] 定义了触发器的循环时间。对于发送触发，最早的 SOF 发送时间点将在 TT_TRIG + 1。 对 0xC7 CANFD_TT_TRIG1 的写操作将会激活发送。激活后，下列寄存器将会被写锁住，直到 TT_TRIG 触发时间到达或者检测到错误。 • 0xC4 CANFD_TRIG_CFG0 • 0xC5 CANFD_TRIG_CFG1 • 0xC6 ~ 0xC7 CANFD_TT_TRIG [1:0]

7.7.6.42. 0xC8~ 0xC9 CANFD_TT_WTRIG[1:0]

默认值：0xFFFF			CANFD_TT_WTRIG[1:0]
位域	类型	默认值	描述
15:0	R/W	0x0	TT_WTRIG[15:0] Watch Trigger Time，观测触发时间 TT_WTRIG[15:0] 定义了观测触发的循环时间。

7.7.6.43. 0xD4 CANFD_DMAC

默认值：0x02			CANFD_DMAC
位域	类型	默认值	描述
7:6	–	–	–
4	R/W	0x0	ONE_BUF_EN 只使用一个 TBUF，仅适用 TTCAN 模式，使能后 TBUF 指针不再变化。 • 0：未使能 • 1：使能
3	–	–	–
2	R/W	0x0	TS SEL tsone/tsall 信号选择 • 0：tsone • 1：tsall
2	–	–	–
1	R/W	0x1	TS AUTO tsone 信号自动控制使能 • 0：未使能 • 1：使能
0	R/W	0x0	DMA DST EN DMA Destination Enable，发送 DMA 使能 • 0：未使能 • 1：使能

7.7.6.44. 0xD5 CANFD_DST_THR

默认值：0x0			CANFD_DST_THR
位域	类型	默认值	描述
7	–	–	–
6:0	R/W	0x0	DMA_DST_THR DMA destination threshold，触发水位，单位为 byte。 当 DMA DST EN 为 1 及 TX buffer 可以存入 1 帧数据，则发起 DMA 请求。如果可写数据量大于等于水位值，那么发起的请求为 burst 类型。

7.7.6.45. 0xD6 CANFD_SCR_EN

默认值：0x0			CANFD_SCR_EN
位域	类型	默认值	描述
7:1	–	–	–
0	R/W	0x0	DMA_SCR_EN DMA source en，source DMA 使能 • 0：未使能 • 1：使能

7.7.6.46. 0xD7 CANFD_SRC_THR

默认值：0x0			CANFD_SRC_THR
位域	类型	默认值	描述
7	–	–	–
6:0	R/W	0x0	DMA_SRC_THR DMA source threshold，触发水位，单位为 byte。 当 DMA SRC EN 为 1 及 RX buffer 可以读取 1 帧数据，则发起 DMA 请求。如果可写数据量大于等于水位值，那么发起的请求为 burst 类型。

7.8. Parallel Bus (PBUS)

Parallel Bus (PBUS) 模块功能主要实现了一组外部并行总线，可与外部 FPGA 和 SRAM 等元器件实现连接和高效通信，提高系统的灵活性和扩展性。此外，PBUS 模块支持 AHB 总线访问配置寄存器和外部设备，从而有效管理和控制与之相连的外部设备，确保数据的顺畅传输。通过合理的寄存器配置和使用约定的映射地址空间，PBUS 模块可提供更多的操作灵活性并提高系统性能。

7.8.1. 特性说明

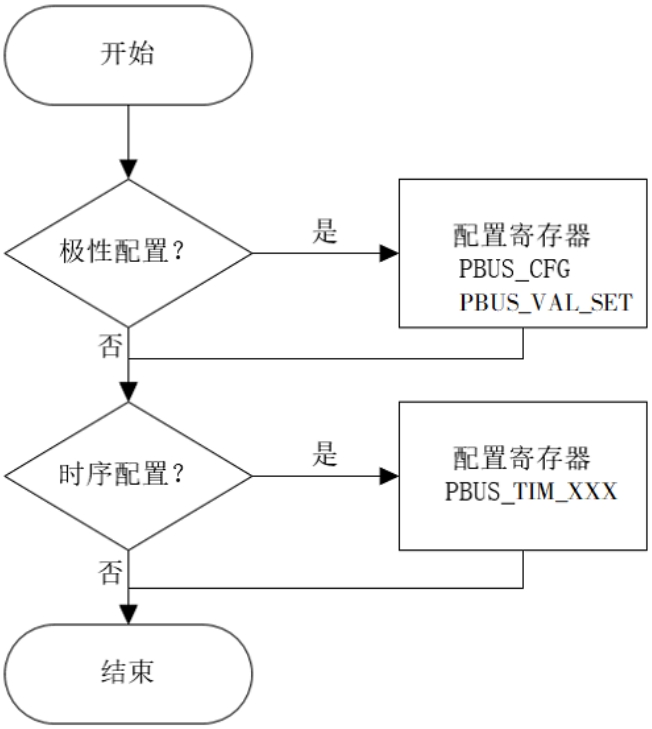
- 支持 AHB 总线访问配置寄存器和外部设备地址空间
- 16–bit 地址和数据总线复用
- 外部设备地址空间为 64 KB
- 每笔操作可实现 16 bit 数据读/ 写
- 可灵活配置 NCS/ NADV/ NWE/ NOE/ AD 信号时序
- 可支持 DMA 对外部设备地址空间进行读写访问
- 支持两路片选信号 CS0/1
- 支持最大 16 bit 独立地址线
- CPU 访问仅支持 Single 操作，DMA 访问支持 Burst 操作
- 支持 I8080/6800 接口

7.8.2. 功能描述

PBUS 可通过映射地址空间 0x1080_0000~0x1080_ffff 进行 PBUS 外设的读写访问：

- 如果直接使用寄存器默认值，则只需使能 CMU 模块时钟以及松开复位，即可实现基本操作。
- 用户也可以根据需要配置寄存器 0x000 PBUS_CFG 的相关参数，实现更多自定义的操作选项。

PBUS 详细使用流程如下图所示：



关于详细的时序配置描述，可查看总线时序与配置说明。

关于 PBUS 工作流程的详细说明，可查看 。

关于寄存器的参数配置，可查看寄存器描述。

7.8.2.1. 总线时序与配置说明

PBUS 模块的时钟周期，均指内部 PBUS CLK 周期。PBUS CLK 从 HCLK 时钟分频得到，通过配置 CLK_DIV 字段可以从 HCLK 进行 2/ 4/ 8 分频，从而获得灵活的时钟配置选项，来适应不同的操作需求。

PBUS 每个信号含义如下所示：

表 7–44 PBUS 信号描述

信号名称	输入/输出	信号描述
CLK	O	外部总线时钟信号被访问外设不需要时钟信号，工作在异步模式；被访问外设需要时钟信号，则工作在同步模式。
NCS	O	外设片选信号，默认低电平有效
NADV	O	总线地址有效信号，默认低电平有效。电平为低时，指示 AD[15:0] 为地址。
NWE	O	总线读写控制信号，默认低电平为写，高电平为读。
NOE	O	外设输出使能信号，默认低电平有效。有效时允许外设往 AD[15:0] 输出读数据信号。
AD[15:0]	I/O	地址/ 数据总线

表 7–44 PBUS 信号描述 (续)

信号名称	输入/输出	信号描述
		此总线 16–bit 位宽，地址与数据复用，并且可传输读写双向数据可访问地址空间为 64K，映射地址为 0x1080_0000~0x1080_ffff。
Adress[15:0]		地址总线 此总线 16 bits 宽，PBUS 模式可选地址/ 数据总线或者地址 + 数据总线 可访问地址空间为 64K，映射地址为 0x10800000~0x1080FFFF

7.8.2.1.1. 信号相位配置

并行信号传输中，信号翻转前后拆解成多个相位，并分别可配置持续长度以及高低电平

NCS 信号配置

NCS 的相位由 IDLE，PRE，PHA，POST 组成：

- 寄存器 0x10: 配置 PRE，PHA，POST 长度
- 寄存器 0x30: 配置 IDLE，PRE，PHA，POST 高低电平控制信号翻转，1 代表高电平，0 代表低电平

图 7–32 NCS 信号配置

NADV 信号配置

NADV 的相位由 IDLE，PRE，PHA，POST 组成：

- 寄存器 0x14: 配置 PRE，PHA，POST 长度
- 寄存器 0x30: 配置 IDLE，PRE，PHA，POST 高低电平控制信号翻转，1 代表高电平，0 代表低电平

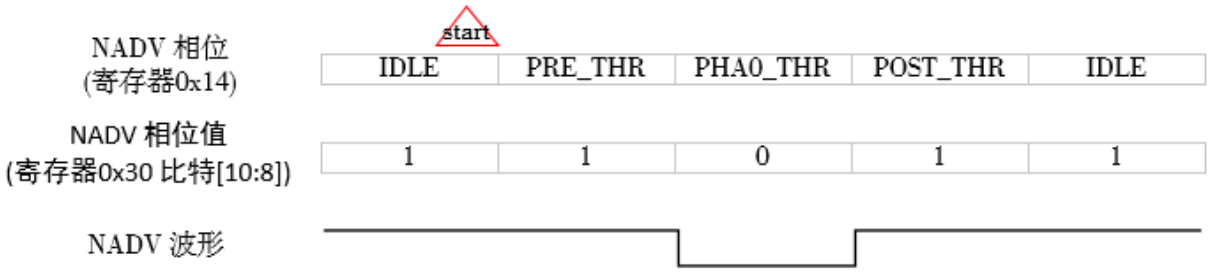


图 7–33 NADV 信号配置

NWE 信号配置

NWE 的相位由 IDLE，PRE，PHA0，WAIT，PHA1 组成：

- 寄存器 0x18: 配置 PRE，PHA0，WAIT，PHA1 长度
- 寄存器 0x30: 配置 IDLE，PRE，PHA0，WAIT，PHA1 高低电平控制信号翻转，1 代表高电平，0 代表低电平

图 7–34 NWE 信号配置

NOE 信号配置

NOE 的相位由 IDLE，PRE，PHA0，WAIT，PHA1 组成：

- 寄存器 0x1C: 配置 PRE，PHA0，WAIT，PHA1 长度
- 寄存器 0x30: 配置 IDLE，PRE，PHA0，WAIT，PHA1 高低电平控制信号翻转，1 代表高电平，0 代表低电平

图 7–35 NOE 信号配置

DATA 信号配置

DATA 的相位由 IDLE，PRE，PHA0，WAIT，PHA1 组成：

- 寄存器 0x20: 配置 PRE，PHA0，WAIT，PHA1 长度

图 7–36 DATA 信号配置

ADDR 信号配置

- ADDR 的相位由 IDLE，PRE，PHA0 组成：
  寄存器 0x24: 配置 PRE，PHA0 长度

图 7–37 ADDR 信号配置

7.8.2.1.2. 并行接口组合配置

地址/ 数据总线写时序

调整以下参数，可灵活配置具体的输出时序。图 7–38 : 地址/数据总线写时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 2,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 1,WE_PHA0 = 6,WE_WAIT = 0,WE_PHA1 = 0
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0
- ADDR_PRE = 1,ADDR_PHA0 = 2
- DATA_PRE = 1,DATA_PHA0 = 2

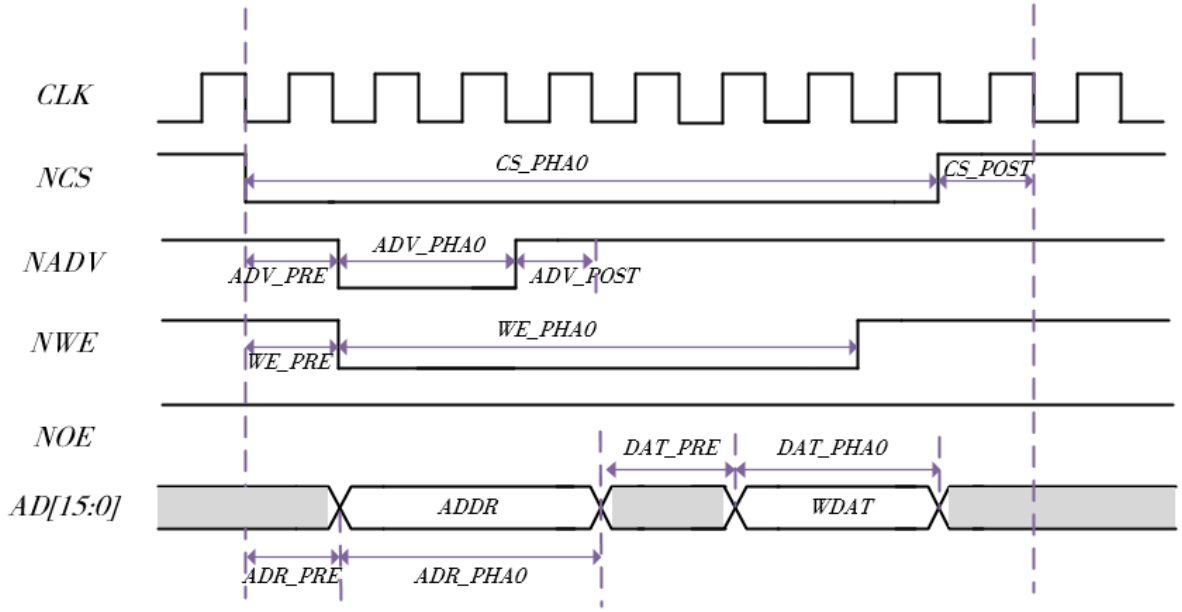


图 7–38 地址/数据总线写时序示例

地址/ 数据总线读时序

调整以下参数，可灵活配置具体的输出时序。图 7–39 : 地址/数据总线读时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 2,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- OE_PRE = 5,OE_PHA0 = 2,OE_WAIT = 0,OE_PHA1 = 0
- OE_IDLE_VAL = 1,OE_PRE_VAL = 1,OE_PHA0_VAL = 0
- ADDR_PRE = 1,ADDR_PHA0 = 3

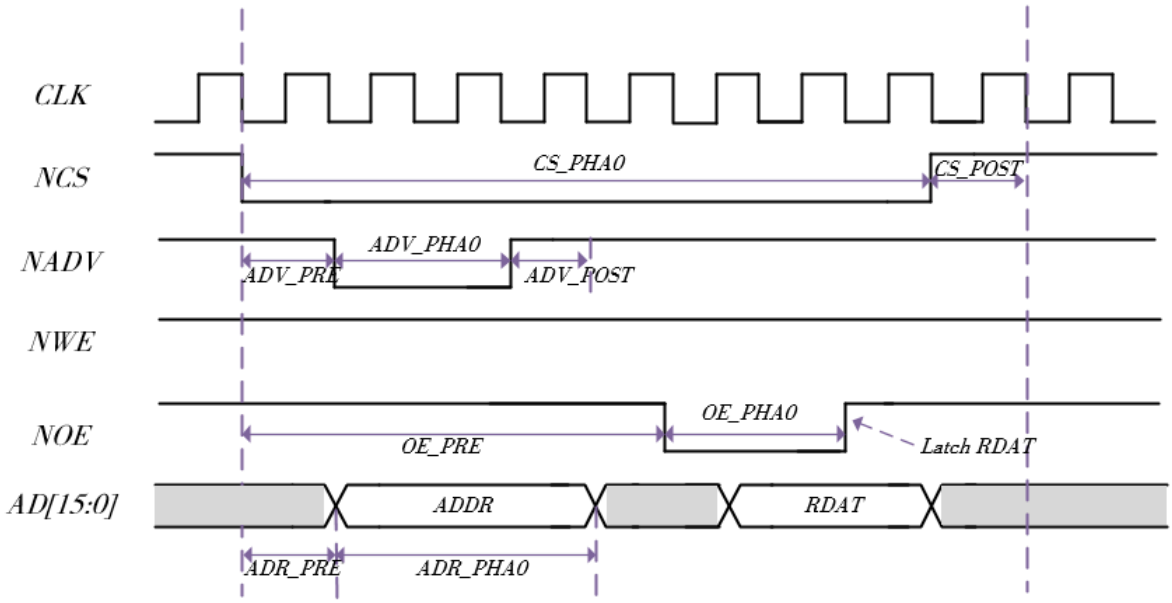


图 7–39 地址/数据总线读时序示例

地址 + 数据总线写时序

调整以下参数，可灵活配置具体的输出时序。图 7–40 : 地址 + 数据总线写时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 2,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 1,WE_PHA0 = 6,WE_WAIT = 0,WE_PHA1 = 0
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0
- ADDR_PRE = 1,ADDR_PHA0 = 2
- DATA_PRE = 1,DATA_PHA0 = 2

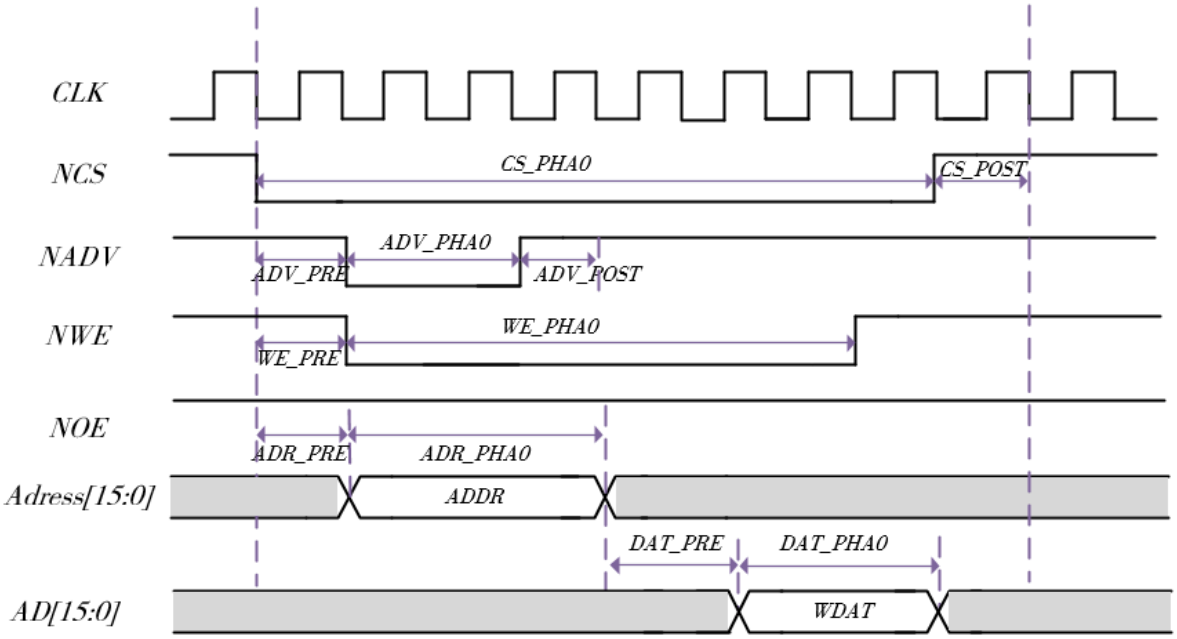


图 7–40 地址 + 数据总线写时序示例

地址 + 数据总线读时序

调整以下参数，可灵活配置具体的输出时序。图 7–41 : 地址+数据总线读时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 2,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- OE_PRE = 5,OE_PHA0 = 2,OE_WAIT = 0,OE_PHA1 = 0
- OE_IDLE_VAL = 1,OE_PRE_VAL = 1,OE_PHA0_VAL = 0
- ADDR_PRE = 1,ADDR_PHA0 = 3

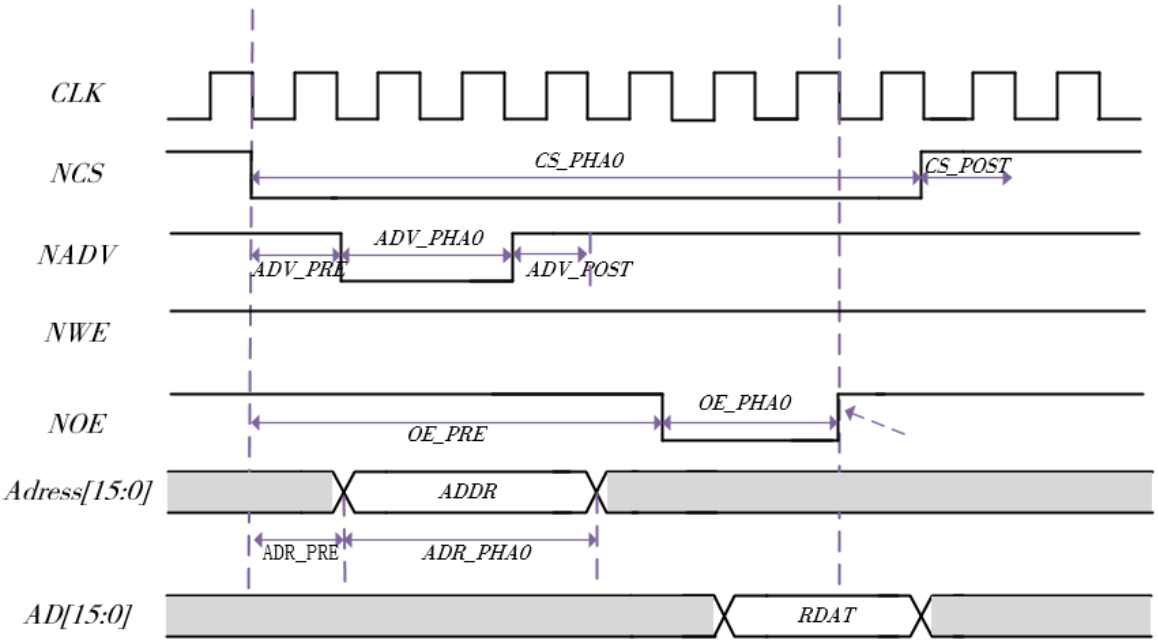


图 7–41 地址+数据总线读时序示例

I8080 总线写时序

调整以下参数，可灵活配置具体的输出时序。图 7–42 : I8080 总线写时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 9,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 3,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 1,WE_PHA0 = 2,WE_WAIT = 2,WE_PHA1 = 2
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0,WE_WAIT_VAL = 1,WE_PHA1_VAL = 0
- ADDR_PRE = 1,ADDR_PHA0 = 3
- DATA_PRE = 1,DATA_PHA0 = 3

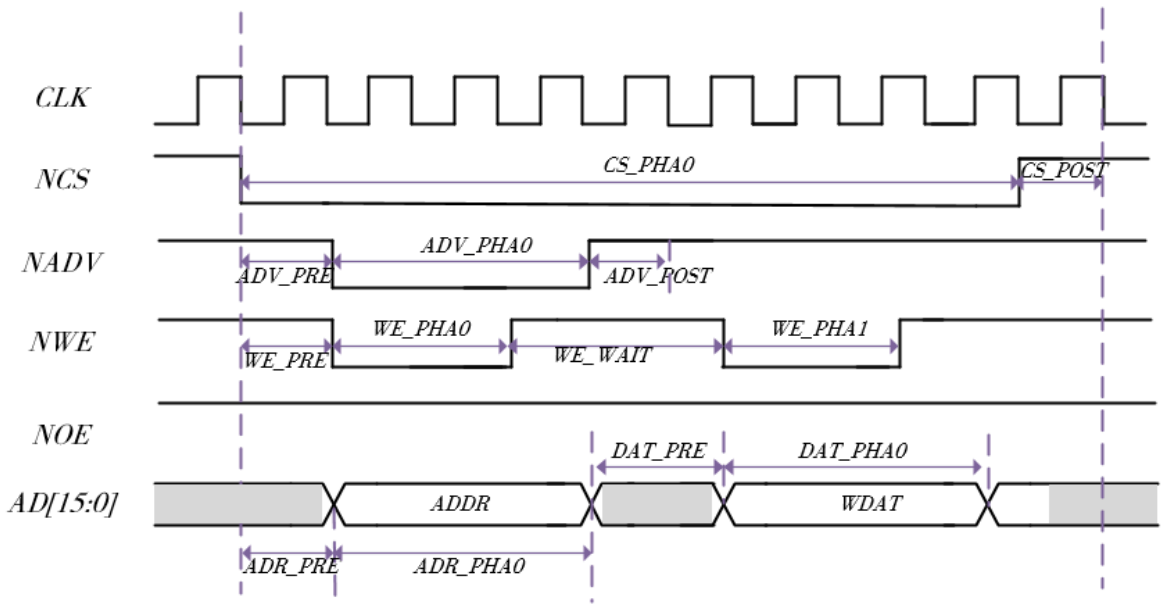


图 7–42 I8080 总线写时序示例

I6800 总线写时序

调整以下参数，可灵活配置具体的输出时序。图 7–43 : I6800 总线写时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 9,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 2,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 1,WE_PHA0 = 6,WE_WAIT = 0,WE_PHA1 = 0
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0
- OE_PRE = 1,OE_PHA0 = 2,OE_WAIT = 2,OE_PHA1 = 2
- OE_IDLE_VAL = 0,OE_PRE_VAL = 0,OE_PHA0_VAL = 1,OE_WAIT_VAL = 0,OE_PHA1_VAL = 1
- ADDR_PRE = 1,ADDR_PHA0 = 3
- DATA_PRE = 1,DATA_PHA0 = 3

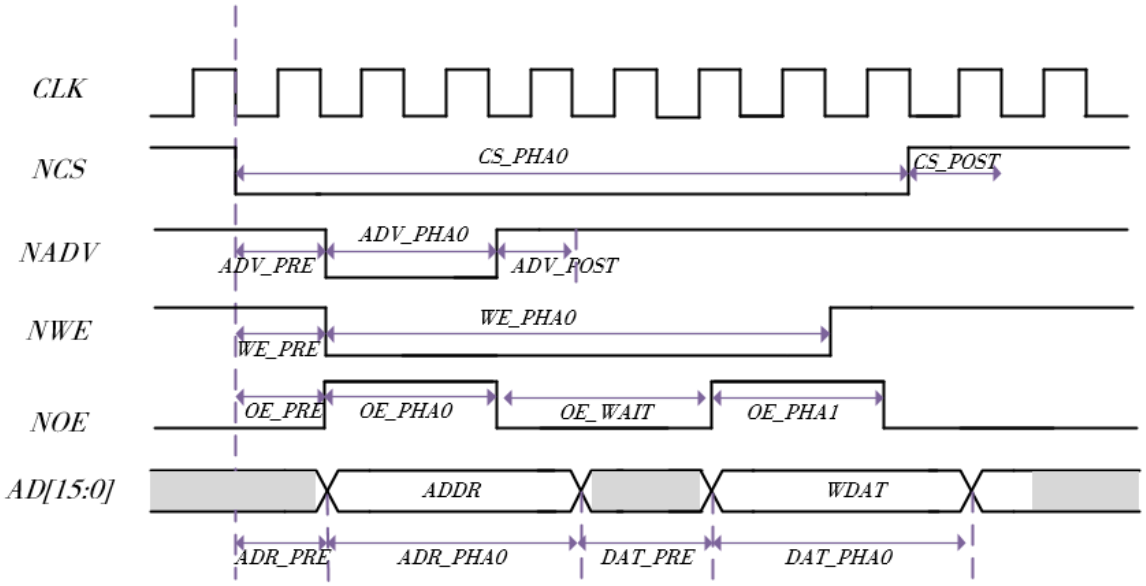


图 7–43 I6800 总线写时序示例

I8080 总线读时序

调整以下参数，可灵活配置具体的输出时序。图 7–44 : I8080 总线读时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 3,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 1,WE_PHA0 = 2,WE_WAIT = 0,WE_PHA1 = 0
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0,WE_WAIT_VAL = 1,WE_PHA1_VAL = 1
- OE_PRE = 5,OE_PHA0 = 2,OE_WAIT = 0,OE_PHA1 = 0
- WE_IDLE_VAL = 1,WE_PRE_VAL = 1,WE_PHA0_VAL = 0,WE_WAIT_VAL = 1,WE_PHA1_VAL = 1
- ADDR_PRE = 1,ADDR_PHA0 = 3

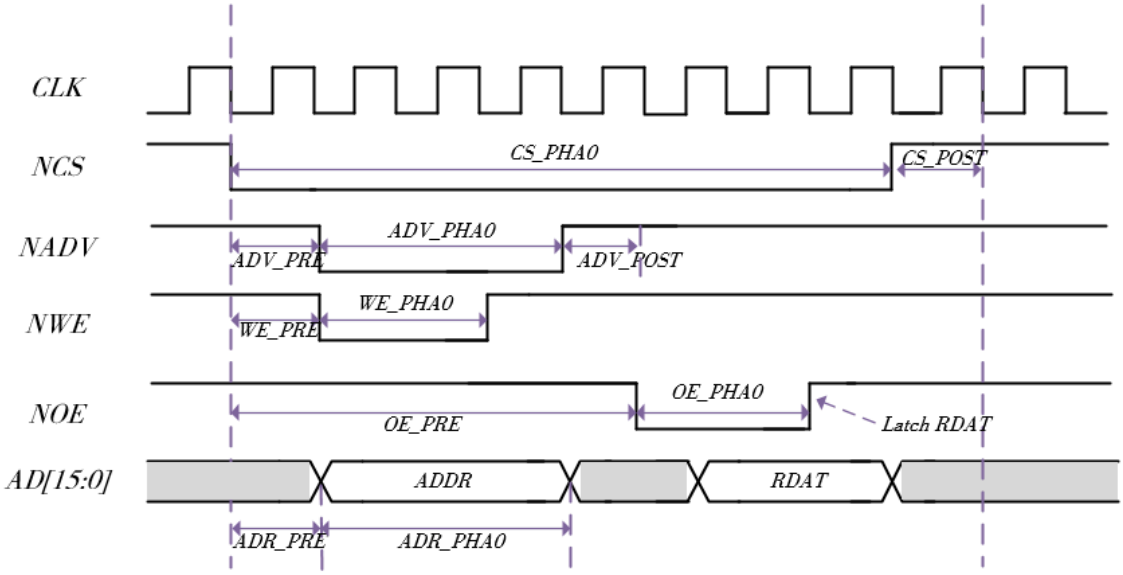


图 7–44 I8080 总线读时序示例

I6800 总线读时序

调整以下参数，可灵活配置具体的输出时序。图 7–45 : I6800 总线读时序示例中给出了一组参数配置示例：

- CS_PRE = 0,CS_PHA0 = 8,CS_POST = 1
- CS_IDLE_VAL = 1,CS_PRE_VAL = 1,CS_PHA0 = 0,CS_POST = 1
- ADV_PRE = 1,ADV_PHA0 = 3,ADV_POST = 1
- ADV_IDLE_VAL = 1,ADV_PRE_VAL = 1,ADV_PHA0 = 0,ADV_POST = 1
- WE_PRE = 6,WE_PHA0 = 3,WE_WAIT = 0,WE_PHA1 = 0
- WE_IDLE_VAL = 0,WE_PRE_VAL = 0,WE_PHA0_VAL = 1,WE_WAIT_VAL = 0,WE_PHA1_VAL = 0
- OE_PRE = 1,OE_PHA0 = 2,OE_WAIT = 3,OE_PHA1 = 2
- WE_IDLE_VAL = 0,WE_PRE_VAL = 0,WE_PHA0_VAL = 1,WE_WAIT_VAL = 0,WE_PHA1_VAL = 1
- ADDR_PRE = 1,ADDR_PHA0 = 3

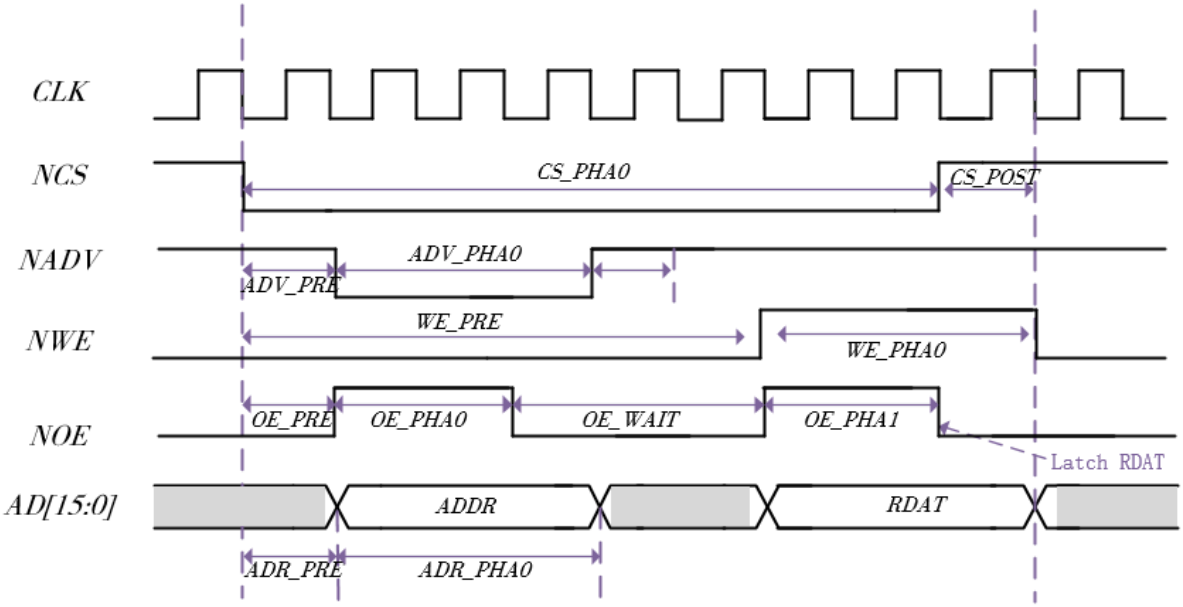


图 7–45 I6800 总线读时序示例

7.8.3. 寄存器列表

表 7–45 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	PBUS_CFG	PBUS 配置	0x000 PBUS_CFG
0x004	PBUS_CTL	PBUS 控制寄存器	0x004 PBUS_CTL
0x008	PBUS_FIFO_CFG	PBUS FIFO 配置	0x008 PBUS_FIFO_CFG
0x00C	PBUS_FIFO_STA	PBUS FIFO 状态	0x00C PBUS_FIFO_STA
0x010	PBUS_TIM_NCS	PBUS TIM 配置 0	0x010 PBUS_TIM_NCS
0x014	PBUS_TIM_NADV	PBUS TIM 配置 1	0x014 PBUS_TIM_NADV
0x018	PBUS_TIM_NWE	PBUS TIM 配置 2	0x018 PBUS_TIM_NWE
0x01C	PBUS_TIM_NOE	PBUS TIM 配置 3	0x01C PBUS_TIM_NOE
0x020	PBUS_TIM_DAT	PBUS TIM 配置 4	0x020 PBUS_TIM_DAT
0x024	PBUS_TIM_ADR	PBUS TIM 配置 5	0x024 PBUS_TIM_ADR
0x0028	PBUS_IRQ_EN	PBUS 中断使能寄存器	0x0028 PBUS_IRQ_EN

表 7–45 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x002C	PBUS_IRQ_STA	PBUS 中断状态寄存器	0x002C PBUS_IRQ_STA
0x0030	PBUS_VAL_SET	PBUS 配置值寄存器	0x0030 PBUS_VAL_SET
0x0034	PBUS_DATA_NUM	PBUS DMA DATA NUMBER	0x0034 PBUS_DATA_NUM
0x0038	PBUS_START	PBUS DMA START 寄存器	0x0038 PBUS_START
0x003C	PBUS_ADDRESS	PBUS 地址寄存器	0x003C PBUS_ADDRESS
0x0040	PBUS_DATA	PBUS DATA 寄存器	0x0040 PBUS_DATA
0xFFC	VERSION	PBUS 版本号	0xFFC VERSION

7.8.4. 寄存器描述

7.8.4.1. 0x000 PBUS_CFG

默认值：0x00000001			PBUS 配置 (PBUS_Configuration)
位域	类型	默认值	描述
31:21	–	–	–
20	R/W	0x0	PBUS_CS_SEL • 0: CS0 • 1: CS1
19	R/W	0x0	PBUS_DATA_WIDTH • 0: 16 位数据总线 • 1: 8 位数据总线
18:16	R/W	0x0	PBUS_MODE PBUS 模式 • 0: MANU 模式 • 1: 地址/数据总线 • 2: 地址+数据总线 • 3：I8080 总线

默认值：0x00000001			PBUS 配置 (PBUS_Configuration)
位域	类型	默认值	描述
			• 4：I6800 总线 • 5~7: 未定义
15	R/W	0	MANU_ADDR_INCR 地址信息是否自动加 1。 • 0: 地址保持寄存器配置值 • 1: 每传输一个数据，地址加 1
14	R/W	0	MANU_ADDR_FROM_FIFO FIFO 模式下，地址信息是否来源于 FIFO。 • 0: 地址来源于寄存器 • 1: 地址来源于 FIFO 的[31:16]
13	R/W	0	NCS_CHECK_DAT_EN NCS 信号回到 IDLE 状态前是否需要检查数据已完成。 • 0: 不检查 Data 信号是否完成。 • 1: NCS 等待 Data 信号完成后才回到 IDLE。
12	R/W	0	ADDR_BUS_EN 地址总线使能 若外设需要输出独立的的地址总线则使能该位。 • 0: 无独立的地址总线 • 1: 有地址总线（最大 16 比特位宽）
11:10	–	–	–
9:8	R/W	0x0	WORK_MODE DMA control 模式 • 0: CPU 直接访问模式，即 DMA 不工作，CPU 通过 AHB 直接读取或者写 PBUS；一次读写一个 WORD。 • 1: DMA 直接访问模式，即 CPU 将需要读写的 command 写入 FIFO，由外部 DMA 直接读取或者写入 PBUS，一次读写一个 WORD。 • 2：DMA FIFO read mode，即 CPU 不能直接访问 PBUS，通过外部 DMA 可以从 BPUS 上读连续数据。 • 3：DMA FIFO write mode，即 CPU 不能直接访问 PBUS，通过外部 DMA 可以从 BPUS 上写连续数据。
7	R/W	0	NOE_POL

默认值：0x00000001			PBUS 配置 (PBUS_Configuration)
位域	类型	默认值	描述
			外设输出使能极性控制，用于控制外设输出使能 NOE 的输出极性。 • 0: NOE 低电平有效 • 1: NOE 高电平有效
6	R/W	0	NWE_POL 读写指示信号极性控制，用于控制读写指示 NWE 的输出极性。 • 0: NWE 低电平为写，高电平为读 • 1: NWE 高电平为写，低电平为读
5	R/W	0	NADV_POL 地址有效信号极性控制，用于控制地址有效 NADV 的输出极性。 • 0: NADV 低电平有效 • 1: NADV 高电平有效
4	R/W	0	NCS_POL 外设片选信号极性控制，用于控制外设片选 NCS 的输出极性。 • 0: NCS 低电平有效 • 1: NCS 高电平有效
3	R/W	0	CLK_POL 总线时钟极性控制，用于控制 PBUS CLK 的输出极性。 • 0: 控制信号与地址/ 数据总线在下降沿发生跳变 • 1: 控制信号与地址/ 数据总线在上升沿发生跳变
2	R/W	0	CLK_OE 总线时钟输出使能 当总线工作在同步模式 (Sync)时，需要通过此字段使能总线时钟信号输出。 • 0：禁止时钟信号输出，被访问外设不需要时钟信号，工作在异步模式 • 1：使能时钟信号输出，被访问外设需要时钟信号，工作在同步模式
1:0	R/W	0x1	CLK_DIV 总线时钟分频 总线时钟 (PBUS CLK)从模块 AHB 总线的 HCLK 经过分频产生，此字段控制总线时钟的分频系数。 • 0: 未定义 • 1: HCLK/2

默认值：0x00000001			PBUS 配置 (PBUS_Configuration)
位域	类型	默认值	描述
			• 2: HCLK/4 • 3: HCLK/8

7.8.4.2. 0x004 PBUS_CTL

默认值：0x00000000			PBUS 控制寄存器 (PBUS_Control)
位域	类型	默认值	描述
31:1	–	–	–
0	W1C	0x0	PBUS_CLR PBUS 软件复位 写 1 则产生 PBUS 软复位，将清除 FIFO 以及内部各信号的状态机至 IDLE。

7.8.4.3. 0x008 PBUS_FIFO_CFG

默认值：0x00000008			PBUS FIFO 配置 (PBUS_FIFO_Configuration)
位域	类型	默认值	描述
31:5	–	–	–
4:0	R/W	0x8	FIFO_THR FIFO 水位设置，分两种情况： • FIFO 处于 TX 模式，如果 FIFO 中数据小于 FIFO_THR，则会发出 DMA_DST_DRQ_BURST • FIFO 处于 RX 模式，如果 FIFO 中数据大于等于 FIFO_THR，则会发出 DMA_SRC_DRQ_BURST

7.8.4.4. 0x00C PBUS_FIFO_STA

默认值：0x00000001			PBUS FIFO 状态 (PBUS_FIFO_Status)
位域	类型	默认值	描述
31:7	–	–	–
6:2	RO	0x0	FIFO_DLEN

默认值：0x00000001			PBUS FIFO 状态 (PBUS_FIFO_Status)
位域	类型	默认值	描述
			FIFO 数据量
1	RO	0x0	FIFO_FULL
0	RO	0x1	FIFO_EMPTY

7.8.4.5. 0x010 PBUS_TIM_NCS

表 7–46

默认值：0x00000000			PBUS TIM 配置 0 (PBUS_Time_Configuration_0)
位域	类型	默认值	描述
31:28	R/W	0x0	NCS_POST 外设片选信号的间隔时间 此字段规定两次片选信号间最少的间隔时间间，单位为 PBUS CLK 的周期数。
27:8	R/W	0x0	NCS_PHA 外设片选信号的有效保持时间 此字段规定了外设片选信号 (NCS)每次有效的保持时间，单位为 PBUS CLK 的周期数。
7:4	–	–	–
3:0	R/W	0x0	NCS PRE 外设片选信号的有效前的 Setup 时间 单位为 PBUS CLK 的周期数。

7.8.4.6. 0x014 PBUS_TIM_NADV

表 7–47

默认值：0x00000000			PBUS TIM 配置 1 (PBUS_Time_Configuration_1)
位域	类型	默认值	描述
31:12	–	–	–
11:8	R/W	0x0	NADV_PHA 外设 NADV 信号的有效保持时间 此字段规定了外设 NADV 每次有效的保持时间，单位为 PBUS CLK 的周期数。

表 7–47 (续)

默认值：0x00000000			PBUS TIM 配置 1 (PBUS_Time_Configuration_1)
位域	类型	默认值	描述
7:4	–	–	–
3:0	R/W	0x0	NADV_PRE NADV 信号的有效前的 Setup 时间 单位为 PBUS CLK 的周期数。

7.8.4.7. 0x018 PBUS_TIM_NWE

默认值：0x00000000			PBUS TIM 配置 2 (PBUS_Time_Configuration_2)
位域	类型	默认值	描述
31:28	–	–	–
27:24	R/W	0x0	NWE_PHA1 外设 NWE 信号 PHA1 的有效保持时间，单位为 PBUS CLK 的周期数。
23:20	–	–	–
19:16	R/W	0x0	NWE_WAIT 外设 NWE 信号 PHA0 和 PHA1 之间转换时间，单位为 PBUS CLK 的周期数。
15:12	–	–	–
11:8	R/W	0x0	NWE_PHA0 外设 NWE 信号 PHA0 的有效保持时间，单位为 PBUS CLK 的周期数。
7:4	–	–	–
3:0	R/W	0x0	NWE_PRE 外设 NWE 信号 PHA0 前的 Setup 时间,单位为 PBUS CLK 的周期数。

7.8.4.8. 0x01C PBUS_TIM_NOE

默认值：0x00000000			PBUS TIM 配置 3 (PBUS_Time_Configuration_3)
位域	类型	默认值	描述
31:28	–	–	–

默认值：0x00000000			PBUS TIM 配置 3 (PBUS_Time_Configuration_3)
位域	类型	默认值	描述
27:24	R/W	0x0	NOE_PHA1 外设 NOE 信号 PHA1 的有效保持时间，单位为 PBUS CLK 的周期数。
23:20	–	–	–
19:16	R/W	0x0	NOE_WAIT 外设 NOE 信号 PHA0 和 PHA1 之间转换时间，单位为 PBUS CLK 的周期数。
15:12	–	–	–
11:8	R/W	0x0	NOE_PHA0 外设 NOE 信号 PHA0 的有效保持时间，单位为 PBUS CLK 的周期数。
7:4	–	–	–
3:0	R/W	0x0	NOE_PRE 外设 NOE 信号 PHA0 前的 Setup 时间,单位为 PBUS CLK 的周期数。

7.8.4.9. 0x020 PBUS_TIM_DAT

默认值：0x00000000			PBUS TIM 配置 4 (PBUS_Time_Configuration_4)
位域	类型	默认值	描述
31:28	–	–	–
27:24	R/W	0x0	DAT_PHA1 外设 DAT 信号 PHA1 的有效保持时间，单位为 PBUS CLK 的周期数。
23:20	–	–	–
19:16	R/W	0x0	DAT_WAIT 外设 DAT 信号 PHA0 和 PHA1 之间转换时间，单位为 PBUS CLK 的周期数。
15:12	–	–	–
11:8	R/W	0x0	DAT_PHA0 外设 DAT 信号 PHA0 的有效保持时间，单位为 PBUS CLK 的周期数。
7:4	–	–	–
3:0	R/W	0x0	DAT_PRE

默认值：0x00000000			PBUS TIM 配置 4 (PBUS_Time_Configuration_4)
位域	类型	默认值	描述
			外设 DAT 信号 PHA0 前的 Setup 时间,单位为 PBUS CLK 的周期数。

7.8.4.10. 0x024 PBUS_TIM_ADR

默认值：0x00000000			PBUS TIM 配置 5 (PBUS_Time_Configuration_5)
位域	类型	默认值	描述
31:12	–	–	–
11:8	R/W	0x0	ADR_PHA 外设 ADR 信号 PHA 的有效保持时间，单位为 PBUS CLK 的周期数。
7:4	–	–	–
3:0	R/W	0x0	ADR_PRE 外设 ADR 信号 PHA0 前的 Setup 时间,单位为 PBUS CLK 的周期数。

7.8.4.11. 0x0028 PBUS_IRQ_EN

默认值：0x00000000			PBUS 中断使能寄存器 (PBUS IRQ Enable Register)
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	FIFO_IRQ_EN FIFO 中断使能 • 0：禁止 • 1：使能

7.8.4.12. 0x002C PBUS_IRQ_STA

默认值：0x00000000			PBUS 中断状态寄存器 (PBUS IRQ Status Register)
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	FIFO_IRQ_STA

默认值：0x00000000			PBUS 中断状态寄存器 (PBUS IRQ Status Register)
位域	类型	默认值	描述
			FIFO 中断状态 • 0：无中断 • 1：有中断 发送时，FIFO 数据低于 FIFO 设置水位，产生中断 接收时，FIFO 数据高于 FIFO 设置水位，产生中断

7.8.4.13. 0x0030 PBUS_VAL_SET

默认值：0x1B1B030B			PBUS 配置值寄存器 (PBUS Value Setting Register)
位域	类型	默认值	描述
31:29	–	–	–
28	R/W	0x1	NOE PHA1 VALUE
27	R/W	0x1	NOE WAIT VALUE
26	R/W	0x0	NOE PHA0 VALUE
25	R/W	0x1	NOE PRE VALUE
24	R/W	0x1	NOE IDLE VALUE
23:21	–	–	–
20	R/W	0x1	NWE PHA1 VALUE
19	R/W	0x1	NWE WAIT VALUE
18	R/W	0x0	NWE PHA0 VALUE
17	R/W	0x1	NWE PRE VALUE
16	R/W	0x1	NWE IDLE VALUE
15:11	–	–	–
10	R/W	0x0	NADV PHA0 VALUE
9	R/W	0x1	NADV PRE VALUE

默认值：0x1B1B030B			PBUS 配置值寄存器 (PBUS Value Setting Register)
位域	类型	默认值	描述
8	R/W	0x1	NADV IDLE VALUE
7:4	–	–	–
3	R/W	0x1	NCS POST VALUE
2	R/W	0x0	NCS PHA0 VALUE
1	R/W	0x1	NCS PRE VALUE
0	R/W	0x1	NCS IDLE VALUE

7.8.4.14. 0x0034 PBUS_DATA_NUM

默认值：0x00000000			PBUS DMA DATA NUMBER (PBUS DMA DATA NUMBER Register)
位域	类型	默认值	描述
31:10	–	–	–
9:0	R/W	0x0	DATA_NUM 需传输的 data 数量，单位是 word.

7.8.4.15. 0x0038 PBUS_START

默认值：0x00000000			PBUS DMA START 寄存器 (PBUS DMA START Register)
位域	类型	默认值	描述
31:1	–	–	–
0	W1C	0x0	DMA_START 写 1 启动 DMA。

7.8.4.16. 0x003C PBUS_ADDRESS

默认值：0x00000000			PBUS 地址寄存器 (PBUS FIFO Address Register)
位域	类型	默认值	描述
31:16	–	–	–
15:0	W/R	0x0	PBUS_Address 寄存器， 注： MANU_ADDR_FROM_FIFO 为 1 时，PBUS 的地址来源于 FIFO，该寄存器无效。

7.8.4.17. 0x0040 PBUS_DATA

默认值：0x00000000			PBUS DATA 寄存器 (PBUS FIFO DATA Register)
位域	类型	默认值	描述
31:16	–	–	–
15:0	W/R	0x0	PBUS_DATA 寄存器，FIFO 入口

7.8.4.18. 0xFFC VERSION

默认值：0x00000102			PBUS 版本号 (VERSION)
位域	类型	默认值	描述
31:0	RO	0x00000102	Version 模块版本号，采用 BCD 码显示，V1.2

7.9. General Purpose Input Output (GPIO)

General Purpose Input Output (GPIO)，通用引脚控制器，用于管理具有通用电气特性的引脚，区别于 USB、DDR 等带专用 PHY 的引脚。

7.9.1. 特性说明

- 八组引脚 (PA/PB/PC/PD/PE/PF/PG/PH)，支持 3.3 V 和 1.8 V 电平
- 每组引脚输入中断脉宽可配置，时间可配置为 10 ns~9.83 ms

- 每个引脚可单独配置为关闭，通用功能和最多七个专用功能
- 每个引脚可单独使能输出，可单独配置中断触发模式
- 每个引脚驱动能力单独配置，可选有八个档位
- 每个引脚上下拉单独配置，可选无上下拉/ 上拉 33 KΩ/ 下拉 33 KΩ
- 每个引脚可单独配置为输入，每个输入 IO 自带施密特触发器，配置输入后直接生效并且无法取消
- 关闭 IO 功能或配置为输入不使能上下拉时 IO 处于高阻态
- IO 不支持 keeper 功能，需要保持 IO 逻辑控制模块有电来保持电平，当前设计 IO 模块控制逻辑处于休眠域，休眠状态下，IO 状态得以保持，可通过中断服务程序唤醒系统

7.9.2. 原理框图

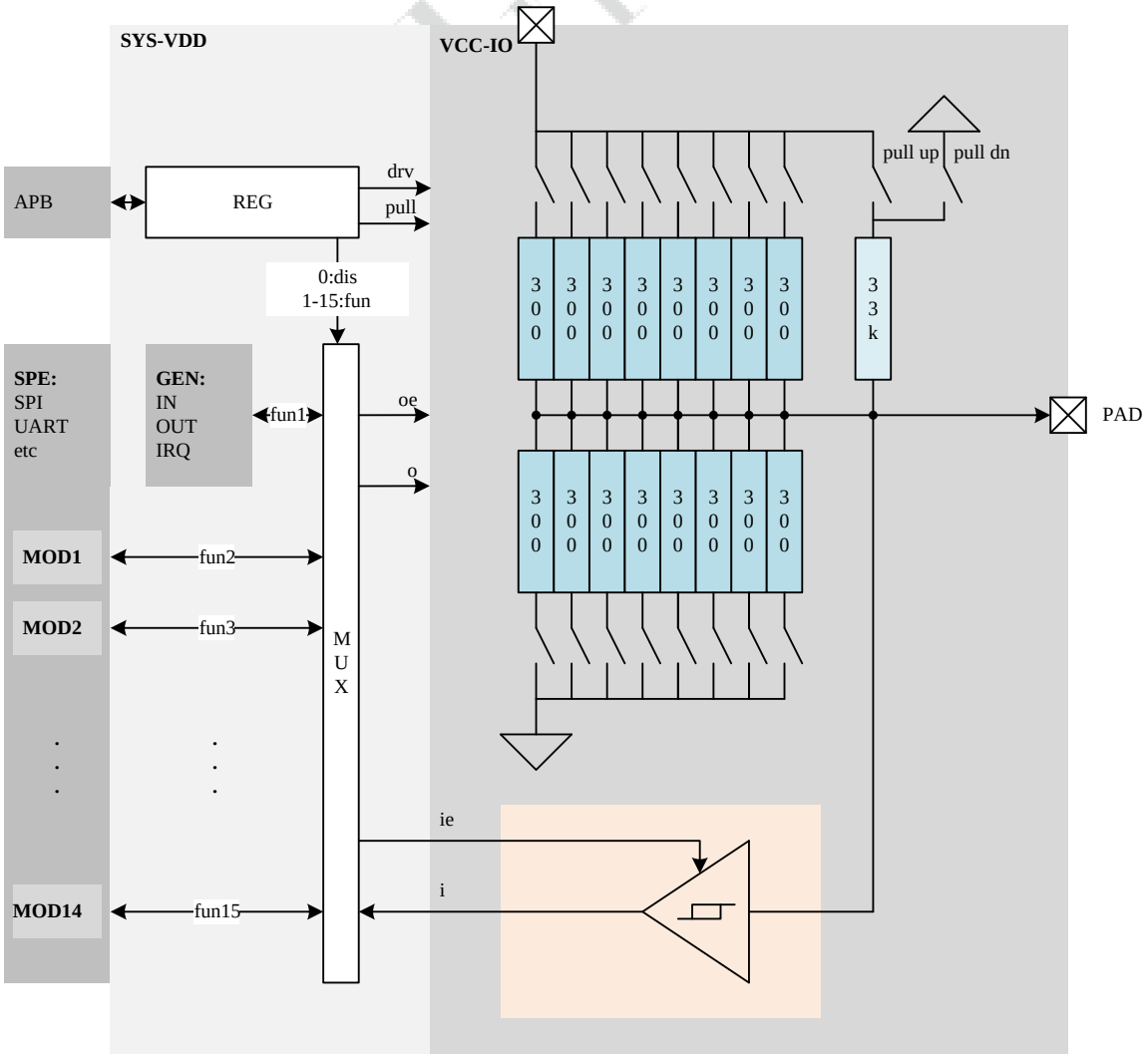


图 7–46 GPIO 原理框图

7.9.3. 功能描述

GPIO 有以下几种配置：

- 关闭
- 通用功能：在 GPIO 模块内部控制。通用功能又分为以下三个功能。
  - 读取输入电平：当引脚配置为通用功能后，使能 GEN_IE，即可在 GEN_IN_STA 对应的位读取到当前的输入电平。
  - 设置输出电平：当引脚配置为通用功能后，使能 GEN_OE，即可配置 GEN_OUT_CFG 使得对应的引脚输出电平。
  - 中断设置：当使用到中断检测功能，同样需要使能 GEN_IE，每个引脚可配置中断模式为：
    - 下降沿中断
    - 上升沿中断
    - 低电平中断
    - 高电平中断
    - 双沿中断（上升沿或下降沿）

  通用功能的输入，支持两级去抖，详情可查看通用输入去抖功能。
- 专用功能：指复用到其它模块使用，比如 UART、I2C 等。

7.9.3.1. 通用输入去抖功能

通用功能的输入，支持两级去抖，去抖后作用于寄存器和产生中断，如下所示：

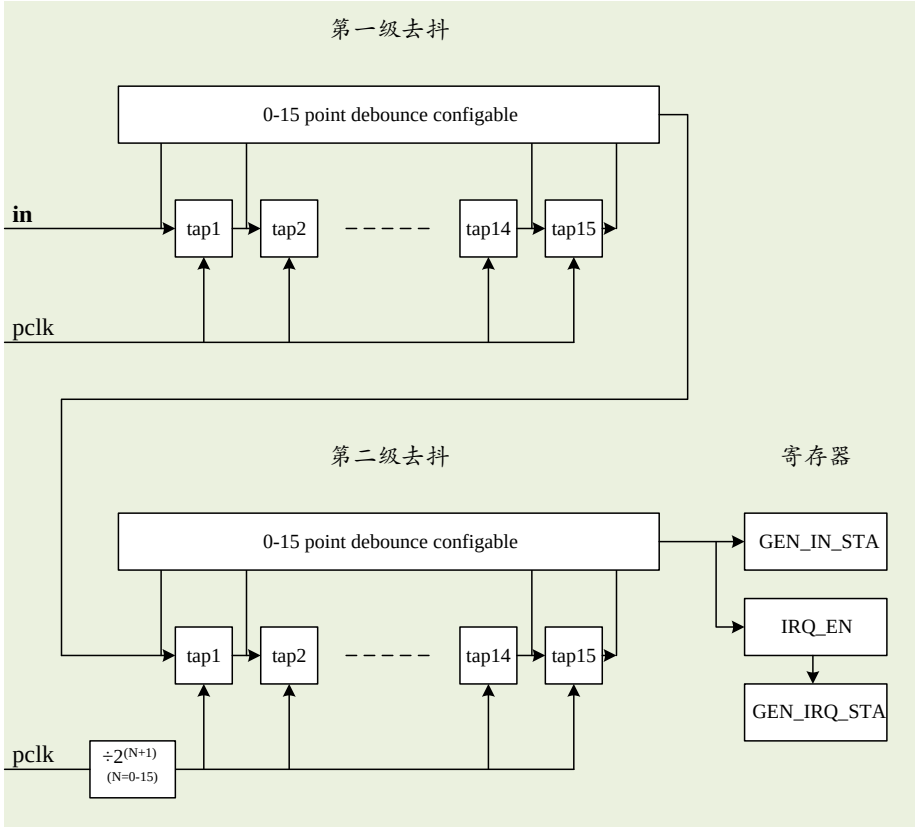


图 7–47 通用输入去抖功能示意图

- 第一级去抖用于滤除高频脉冲，去抖可配置使用 1–15 个参考点（配置为 0 表示不做去抖）。典型应用下（APB0 为 100 MHz 时钟），采样周期为 10 ns，则过滤脉冲最小为 10 ns，最大为 150 ns。
- 第二级去抖用于滤除宽脉冲，如机械按键抖动等，去抖可配置使用 1–15 个参考点（配置为 0 表示不做去抖）。典型应用下（APB0 为 100 MHz 时钟），采样周期可配置为 $2^{(N+1)}$ (N = 0 – 15)，则最大采样周期为 0.65 ms，最大过滤脉冲为 9.83 ms。

7.9.4. 寄存器列表

表 7–48 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x00+G*0x100	GEN_IN_STA	GPIO 通用输入状态寄存器	0x00+G*0x100 GEN_IN_STA
0x04+G*0x100	GEN_OUT_CFG	GPIO 通用输出控制寄存器	0x04+G*0x100 GEN_OUT_CFG
0x08+G*0x100	GEN_IRQ_EN	GPIO 通用中断使能寄存器	0x08+G*0x100 GEN_IRQ_EN
0x0C+G*0x100	GEN_IRQ_STA	GPIO 通用中断状态寄存器	0x0C+G*0x100 GEN_IRQ_STA
0x10+G*0x100	GEN_OUT_CLR	GPIO 通用输出清除寄存器	0x10+G*0x100 GEN_OUT_CLR

表 7–48 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x14+G*0x100	GEN_OUT_SET	GPIO 通用输出置位寄存器	0x14+G*0x100 GEN_OUT_SET
0x18+G*0x100	GEN_OUT_TOG	GPIO 反转寄存器	0x18+G*0x100 GEN_OUT_TOG
0x20+G*0x100	GEN_IRQ_EN_CLR	GPIO 通用中断使能清除寄存器	0x20+G*0x100 GEN_IRQ_EN_CLR
0x24+G*0x100	GEN_IRQ_EN_SET	GPIO 通用中断使能置位寄存器	0x24+G*0x100 GEN_IRQ_EN_SET
0x80+P*0x4+G*0x100	PIN_CFG	GPIO 引脚功能配置寄存器	0x80+P*0x4+G*0x100 PIN_CFG
0xFFC	VERSION	GPIO 版本寄存器	0xFFC VERSION

其中，P 和 G 取值范围如下：

表 7–49 GPIO P 和 G 取值范围 (BASE_ADDR: 0x10700000)

分组	偏移地址	G 取值	P 取值
PA	0x0000	0	0~23
PB	0x0100	1	0~6
PC	0x0200	2	0~18
PD	0x0300	3	0~29
PE	0x0400	4	0~23
PF	0x0500	5	0~9
PG	0x0600	6	0~22
PH	0x0700	7	0~5
PK	0x0A00	10	0~12

7.9.5. 寄存器描述

7.9.5.1. 0x00+G*0x100 GEN_IN_STA

注：

该寄存器在满足以下任一情况下才有效：

- PIN_FUN=1 且 GEN_IN_EN=1 时有效，或
- PIN_FUN = 2~15 且 SPE_IE_FORCE=1 时有效。

默认值：0x00000000			GPIO 通用输入状态寄存器
位域	类型	默认值	描述
31:0	RO	0x0	IN_STA 引脚输入状态 • 0: 表示输入为低电平 • 1: 表示输入为高电平 每位对应一个引脚。

7.9.5.2. 0x04+G*0x100 GEN_OUT_CFG

注：

该寄存器仅在 PIN_FUN=1 且 GEN_OUT_EN=1 时有效。

默认值：0x00000000			GPIO 通用输出控制寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	OUT_CFG 引脚输出配置 • 0: 输出设置为低电平 • 1: 输出设置为高电平 每位对应一个引脚

7.9.5.3. 0x08+G*0x100 GEN_IRQ_EN

注：

该寄存器仅在 PIN_FUN=1 且 GEN_IN_EN=1 时有效。

默认值：0x00000000			GPIO 通用中断使能寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	IRQ_EN 引脚中断使能 • 0: 关闭 • 1: 使能 每位对应一个引脚

7.9.5.4. 0x0C+G*0x100 GEN_IRQ_STA

注：

该寄存器仅在同时满足以下条件时才有效：

- PIN_FUN=1
- GEN_IN_EN=1
- IRQ_EN=1

默认值：0x00000000			GPIO 通用中断状态寄存器
位域	类型	默认值	描述
31:0	R/W1C	0x0	IRQ_STA 引脚中断状态，该寄存器写 1 清零，即对对应位写 1 为清除对应的寄存器 • 0: 无中断 • 1: 有中断 每位对应一个引脚

7.9.5.5. 0x10+G*0x100 GEN_OUT_CLR

默认值：0x00000000			GPIO 通用输出清除寄存器
位域	类型	默认值	描述
31:0	WO	0x0	OUT_CFG_CLR 输出控制寄存器清除 • 0: 对应引脚电平保持不变 • 1: 对应引脚设置为低电平 每位对应一个引脚

7.9.5.6. 0x14+G*0x100 GEN_OUT_SET

默认值：0x00000000			GPIO 通用输出置位寄存器
位域	类型	默认值	描述
31:0	WO	0x0	OUT_CFG_SET 输出控制寄存器置位 • 0: 对应引脚电平保持不变 • 1: 对应引脚设置为高电平 每位对应一个引脚

7.9.5.7. 0x18+G*0x100 GEN_OUT_TOG

默认值：0x00000000			GPIO 反转寄存器
位域	类型	默认值	描述
31:0	WO	0x0	OUT_CFG_TOG 输出控制寄存器置位 • 0: 对应引脚电平不变 • 1: 对应引脚电平反转 每位对应一个引脚

7.9.5.8. 0x20+G*0x100 GEN_IRQ_EN_CLR

默认值：0x00000000			GPIO 通用中断使能清除寄存器
位域	类型	默认值	描述
31:0	WO	0x0	IRQ_EN_CLR 引脚中断使能清除 • 0: IRQ_EN 对应位保持不变 • 1: IRQ_EN 对应位被清除 配置该寄存器清除对应的中断使能

7.9.5.9. 0x24+G*0x100 GEN_IRQ_EN_SET

默认值：0x00000000			GPIO 通用中断使能置位寄存器
位域	类型	默认值	描述
31:0	WO	0x0	IRQ_EN_SET 引脚中断使能置位 • 0: IRQ_EN 对应位保持不变 • 1: IRQ_EN 对应位被置位 配置该寄存器置位对应的中断使能

7.9.5.10. 0x80+P*0x4+G*0x100 PIN_CFG

默认值：PH0~5，PG21~22: 0x00000332 默认值：OTHERS: 0x00000000			GPIO 引脚功能配置寄存器
位域	类型	默认值	描述
31:28	R/W	0x0	GEN_IN_DB1_POINT 通用输入第二级去抖参考点数，仅在 PIN_FUN=1 且 GEN_IE=1 才有效。 该配置值为去抖采样点数，配置为 0 则关闭本级去抖。
27:24	R/W	0x0	GEN_IN_DB1_SAMP 通用输入第二级去抖采样除频，仅在 PIN_FUN=1 且 GEN_IE=1 才有效。 第二级去抖采样周期为 $2^{(SAMP+1)}$ 个 APB0 时钟 (100 MHz) 周期。
23:20	R/W	0x0	GEN_IN_DB0_POINT 通用输入第一级去抖采样点数，仅在 PIN_FUN=1 且 GEN_IE=1 才有效。 第一级去抖采样周期为一个 APB0 时钟 (100 MHz) 周期。 该配置值为去抖采样点数，配置为 0 则关闭本级去抖。
19	–	–	–
18	R/W	0x0	SPE_IE_FORCE 专用功能输入强制使能，仅在 PIN_FUN= 2~15才有效 • 0: 关闭 • 1: 打开 此字段用于调试使用，强制使能输入，并将输入状态存到通用输入状态寄存器
17	R/W	0x0	GEN_OE 通用功能输出使能，仅在 PIN_FUN=1 才有效。

默认值：PH0~5，PG21~22: 0x00000332 默认值：OTHERS: 0x00000000			GPIO 引脚功能配置寄存器
位域	**类型**	**默认值**	**描述**
			• 0: 关闭 • 1: 打开
16	R/W	0x0	GEN_IE 通用功能输入使能，仅在 PIN_FUN=1 才有效。 • 0: 关闭 • 1: 打开
15	–	–	–
14:12	R/W	0x0	GEN_IRQ_MODE 通用功能中断检测模式，仅在 PIN_FUN=1 且 GEN_IE=1 才有效： • 0: 下降沿 • 1: 上升沿 • 2: 低电平 • 3: 高电平 • 4: 双沿（上升沿或下降沿） • 5~7: 预留
11:10	–	–	–
9:8	R/W	PG21~22, PH0~5: 3 OTHERS: 0	PIN_PULL 引脚上下拉设置 • 0: 关闭，引脚处于高阻态 • 1: 预留 • 2: 下拉 33 KΩ 到 GND • 3: 上拉 33 KΩ 到 VCC–IO **注：** 引脚功能设置 PIN_FUN 配置为通用功能或专用功能，该寄存器都起作用。
7	–	–	–
6:4	R/W	PG21~22, PH0~5: 3 OTHERS: 0	PIN_DRV 引脚输出驱动能力

默认值：PH0~5，PG21~22: 0x00000332 默认值：OTHERS: 0x00000000			GPIO 引脚功能配置寄存器
位域	**类型**	**默认值**	**描述**
			<table><tr><th>–</th><th>3.3V</th><th>1.8V</th></tr><tr><td>0</td><td>180Ω</td><td>300Ω</td></tr><tr><td>1</td><td>90Ω</td><td>150Ω</td></tr><tr><td>2</td><td>60Ω</td><td>100Ω</td></tr><tr><td>3</td><td>45Ω</td><td>75Ω</td></tr><tr><td>4</td><td>36Ω</td><td>60Ω</td></tr><tr><td>5</td><td>30Ω</td><td>50Ω</td></tr><tr><td>6</td><td>26Ω</td><td>43Ω</td></tr><tr><td>7</td><td>23Ω</td><td>38Ω</td></tr></table> **注：** 引脚功能设置 PIN_FUN 配置为通用功能或专用功能，该寄存器都起作用。
3:0	R/W	PG21~22, PH0~5: 2 OTHERS: 0	PIN_FUN 引脚功能设置 • 0: 关闭 • 1: 通用功能 (GEN) • 2~15: 专用功能 (SPE)

7.9.5.11. 0xFFC VERSION

默认值：0x00000102			GPIO 版本寄存器
位域	**类型**	**默认值**	**描述**
31:0	RO	0x00000102	VERSION GPIO 版本，V1.2

8. 模拟

8.1. Thermal Sensor (THS)

THS 温度传感器模块，可以管理和读取芯片内置温度传感器，并基于读取的温度值进行高低温警告功能。

THS 温度传感器模块通过 ADC 来获取内部温度传感器温度值，因此在启动之前，需要先启动 Analog-to-Digital Converter (ADC)模块。

8.1.1. 特性说明

- 支持两路温度传感器
- ADC 与温度传感器捕获时间可配
- 支持单次读取和周期读取两种模式
- 支持一次多点读取，并进行算术平均滤波
- 基于周期读取模式时，支持高/ 低温报警（产生中断）

8.1.2. 原理框图

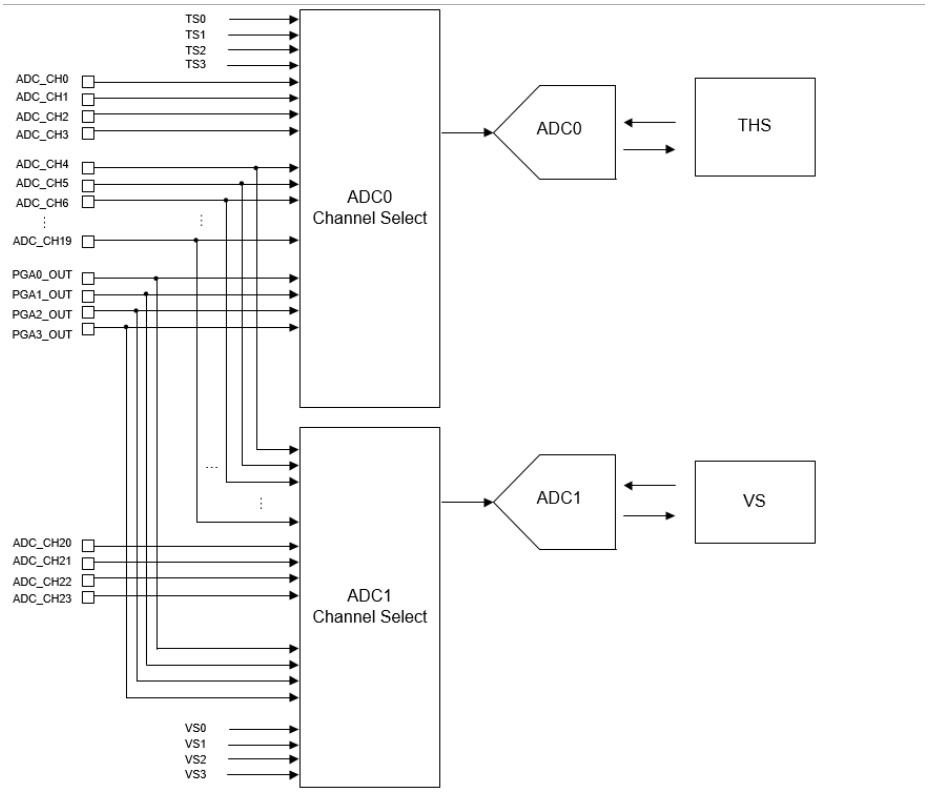


图 8-1 THS 接口模块与 ADC 连接关系

8.1.3. 功能描述

8.1.3.1. 中断与过温保护复位产生机制

THS 模块的中断输出，以及过温保护系统复位输出如下图所示。

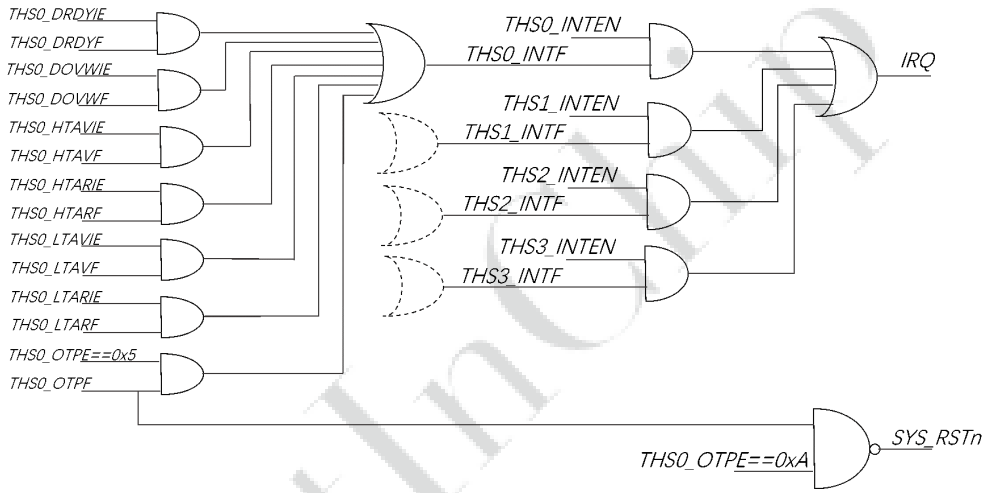


图 8–2 中断，以及过温保护系统复位输出

8.1.3.2. 高/ 低温报警的双阈值描述

由于温度在边界处反复低幅度变动时，单阈值容易产生频繁报警的问题，因此高/ 低温报警采用了双阈值，详情如下所示：

- 高温报警：
  - 高温报警有效阈值 (HTAVV)
  - 高温报警解除阈值 (HTARV)
- 低温报警：
  - 低温报警有效阈值 (LTAVV)
  - 低温报警解除阈值 (LTARV)

双阈值的使用示意图如下：

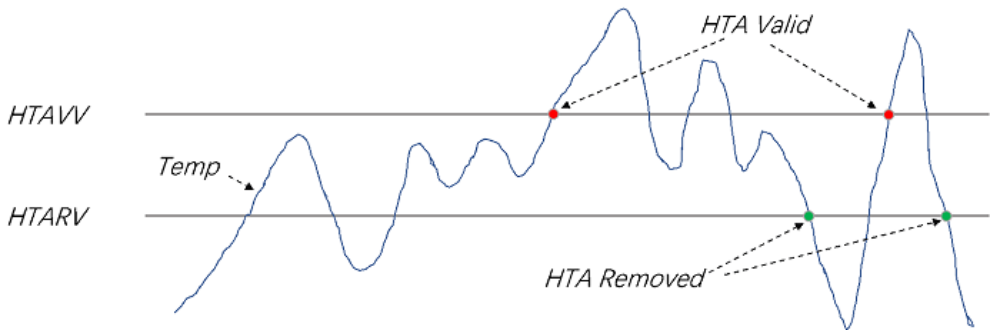


图 8–3 高温报警（HTA）双阈值判断示意图

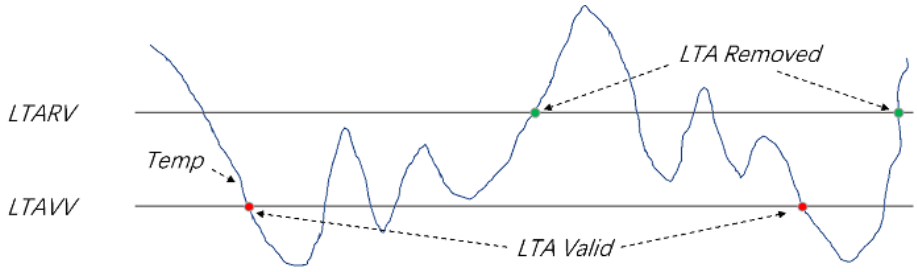


图 8–4 低温报警（LTA）双阈值判断示意图

注：

- 通常情况下，LTAVV < LTARV < HTARV < HTAVV。
- 在周期模式下，仅对采样后取平均值进行阀值判断，而不对原始数据进行阀值判断，避免某一次误触发。
- 温度越高，输入电压越高，ths 值越大

8.1.4. 寄存器列表

表 8–1 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	MCR	THS 模块控制	0x000 MCR
0x004	MINTR	THS 中断控制与状态	0x004 MINTR
0x008	THS_NUM	THS 数量	0x008 THS_NUM
0x100+n*0x20	THSn_CFG	THS 通道控制配置	0x100+n*0x20 THSn_CFG
0x104+n*0x20	THSn_ITV	THS 通道采样间隔配置	0x104+n*0x20 THSn_ITV
0x108+n*0x20	THSn_FIL	THS 采样滤波配置	0x108+n*0x20 THSn_FIL
0x10C+n*0x20	THSn_DATA	THS 数据	0x10C+n*0x20 THSn_DATA
0x110+n*0x20	THSn_INT	THS 中断状态	0x110+n*0x20 THSn_INT
0x114+n*0x20	THSn_HTAV	THS 高温报警阈值设置	0x114+n*0x20 THSn_HTAV
0x118+n*0x20	THSn_LTAV	THS 低温报警阈值设置	0x118+n*0x20 THSn_LTAV
0x11C+n*0x20	THSn_OTPV	THS 过温保护阈值设置	0x11C+n*0x20 THSn_OTPV
0xFFC	VERSION	THS 版本号	0xFFC VERSION

8.1.5. 寄存器描述

8.1.5.1. 0x000 MCR

默认值：0x00002F30			THS 模块控制 (THS Module Control Register)
位域	类型	默认值	描述
31:20	–	–	–
19	R/W	0	THS3_EN（THS3 Enable, THS3 传感器使能） • 0: THS3 禁止 • 1: THS3 使能
18	R/W	0	THS2_EN（THS2 Enable, THS2 传感器使能） • 0: THS2 禁止 • 1: THS2 使能
17	R/W	0	THS1_EN（THS1 Enable, THS1 传感器使能） • 0: THS1 禁止 • 1: THS1 使能
16	R/W	0	THS0_EN（THS0 Enable, THS0 传感器使能） • 0: THS0 禁止 • 1: THS0 使能
15:2	–	–	–
0	R/W	0	EN（模块使能） • 0: 本模块被禁止，本模块停止工作 • 1: 本模块被使能

8.1.5.2. 0x004 MINTR

默认值：0x00000000			THS 中断控制与状态 (THS Module Interrupt Register)
位域	类型	默认值	描述
31:20	–	–	–
19	RO	0	THS3_INTF THS3 Interrupt Flag，THS3 中断标志，只可读。

默认值：0x00000000			THS 中断控制与状态 (THS Module Interrupt Register)
位域	类型	默认值	描述
			只要清除了相应 THS 的中断标志，就会自动清零 • 0: THS3 无中断请求 • 1: THS3 有中断请求
18	RO	0	THS2_INTF（ ） THS2 Interrupt Flag，THS2 中断标志，只可读。 只要清除了相应 THS 的中断标志，就会自动清零 • 0: THS2 无中断请求 • 1: THS2 有中断请求
17	RO	0	THS1_INTF THS1 Interrupt Flag，THS1 中断标志，只可读。 只要清除了相应 THS 的中断标志，就会自动清零 • 0: THS1 无中断请求 • 1: THS1 有中断请求
16	RO	0	THS0_INTF（ ） THS0 Interrupt Flag，THS0 中断标志，只可读。 只要清除了相应 THS 的中断标志，就会自动清零 • 0: THS0 无中断请求 • 1: THS0 有中断请求
15:4	–	–	–
3	R/W	0	THS3_INTEN THS3 Interrupt Enable，THS3 中断使能信号 • 0：禁止 THS3 中断 • 1：使能 THS3 中断
2	R/W	0	THS2_INTEN THS2 Interrupt Enable，THS2 中断使能信号 • 0：禁止 THS2 中断 • 1：使能 THS2 中断
1	R/W	0	THS1_INTEN THS1 Interrupt Enable，THS1 中断使能信号

默认值：0x00000000			THS 中断控制与状态 (THS Module Interrupt Register)
位域	类型	默认值	描述
			• 0：禁止 THS1 中断 • 1：使能 THS1 中断
0	R/W	0	THS0_INTEN THS0 Interrupt Enable，THS0 中断使能信号 • 0：禁止 THS0 中断 • 1：使能 THS0 中断

8.1.5.3. 0x008 THS_NUM

默认值：0x00000004			THS 数量 (THS Number)
位域	类型	默认值	描述
31:3	–	–	–
2:0	RO	0x4	THS_MAX THS 最大编号，规定了 THSn 中 控制器通道数量。n 小于等于 4，即传感器数量最少为 1 个，最多为 4 个。

8.1.5.4. 0x100+n*0x20 THSn_CFG

(1)

默认值：0x000(n+8)2F00			THS 通道控制配置 (THS Channel N Config)
位域	类型	默认值	描述
31:28	–	0x0	–
27	R/W	0x0	Inverted Select • 0：取反 • 1：不变
26:5	–	–	–
4	R/W	0x0	ADC_PRI ADC Priority，ADC 采样的优先级控制

(1) n = 0~1 为 ths 通道号

默认值：0x000(n+8)2F00			THS 通道控制配置 (THS Channel N Config)
位域	类型	默认值	描述
			• 0：低优先级非实时采样，采样命令与其他非实时采样命令进入采样队列。 • 1：高优先级实时采样，采样命令会被直接处理，不进入采样队列。
3	R/W	0x0	–THS_CMD_TYPE THS Command Type • 0：主动模式，主动发送请求 ADC 采样的命令。 • 1：被动模式，通过 ADC 队列请求控制。
2	R	0x0	THS Channel State，工作状态： • 0: 当前 THS 通道处于空闲状态 • 1: 当前 THS 通道处于忙碌状态
1	R/W	0x0	PSEN Periodic Sample Enable，温度传感器周期采样使能。 • 0：停止相应温度传感器周期采样。 • 1：启动相应温度传感器周期采样。 采样间隔由 THSn_ITV 寄存器设定。如果在软件清零之前硬件已经启动了一次新的采样，新采样会继续完成，并最终返回数据。 **注：** 不允许同时使能周期采样与单次采样，即 SSEN 与 PSEN 不能同时有效。
0	R/WAC	0x0	SSEN Single Sample Enable，温度传感器单点采样使能。 • 0：相应传感器的温度单点采样已完成。 • 1：启动相应传感器的温度单点采样。 由硬件自动清零，表示相应传感器的温度单点采样已完成。 **注：** 尽管描述为单点采样，但为了避免单次采样误差，可能需要通过单点多次采样后的数据进行算术平均，得到一个相对稳定的温度值，因此单点采样实际上会包含多次采样。对应的由硬件向 ADC 接口控制器连续发出多个采样命令。连续发出采样命令的个数由滤波器配置 FIL 设定。例如滤波器配置为 0x1（2 点平均），则每次启动 SSEN，硬件会连续发出 2 个采样命令，接收到 2 个温度数据之后，硬件自动清零 SSEN。

8.1.5.5. 0x104+n*0x20 THSn_ITV

(2)

默认值：0x016EFFFF			THS 通道采样间隔配置 (THS Channel N Interval)
位域	类型	默认值	描述
31:16	R/W	0x016E	ITV Interval，采样间隔 周期采样的采样间隔，单位为 T(div_clk)，即采样间隔 = ITV* T(div_clk)。 假设 PCLK = 24 MHz，ITV = 0x16E = 366，DIV=65535，采样间隔 = 366 * (65535 + 1) * 41.67 ns = 1 s **注：** 不可为 0。硬件自动保护：当软件配置为 0 时，效果等同于 1。
15:0	R/W	0xFFFF	DIV Divide CLK，分频时钟 时钟 (div_clk) 分频系数，单位为 PCLKT (div_clk) = (DIV+1) *T (PCLK)

8.1.5.6. 0x108+n*0x20 THSn_FIL

(3)

默认值：0x00000000			THS 采样滤波配置 (THS Channel Filter)
位域	类型	默认值	描述
31:2	–	–	–
1:0	R/W	0x0	FIL Filter，滤波器配置 温度传感器采样数据算数平均滤波器的输入样本数。 • 0x0：无滤波 • 0x1：2 • 0x2：4 • 0x3：8 此字段同时定义了单次采样和周期采样每次的连续采样命令数量（采样点数），控制器接收对应的采样点数据后，进行算数平均，并把平均值通过本传感器的 THSn_DATA 寄存器返回。

(2) n = 0~1 为 ths 通道号

(3) n = 0~1 为 ths 通道号

8.1.5.7. 0x10C+n*0x20 THSn_DATA

(4)

默认值：0x00000000			THS 数据 (THS Channel N Data)
位域	类型	默认值	描述
31:14	–	–	–
13:0	RO	0x000	DATA ADC DATA,，ADC 采样数据 温度传感器的 ADC 采样数据结果。 • 如果 THSn_FIL.FIL 为 0x0，直接返回 ADC 的采样值。 • 否则返回的是多个采样点的算数平均值。 **注：** 模块无数据 FIFO，本寄存器只能暂存一次的采样结果，如果未被及时读取，则会被后来新的采样数据直接覆盖。

8.1.5.8. 0x110+n*0x20 THSn_INT

(5)

默认值：0x00000000			THS 中断状态 (THS Channel N Interrupt)
位域	类型	默认值	描述
31:29	–	–	–
28	R/W1C	0x0	OTPF Over-Temperature-Protected Flag，过温保护标志，仅周期采样模式（THSn_CFG.PSEN=1）时有效。 操作方式为硬件置位，软件写 1 清零。当温度传感器采样值超过 THSn_OTPV.OTPV，就会置为 1。 • 如果 OTPE=中断使能，则产生中断信号。 • 如果 OTPE=复位使能，则产生系统复位信号
27	R/W1C	0x0	LTARF Low-Temperature-Alarm Removed Flag，低温警告解除标志

(4) n = 0~1 为 ths 通道号

(5) n = 0~1 为 ths 通道号

默认值：0x00000000			THS 中断状态 (THS Channel N Interrupt)
位域	类型	默认值	描述
			• 仅周期采样模式 (THSn_CFG. PSEN = 1) 时有效。 • 硬件置位，软件写 1 清零。 • 当温度传感器采样值从低于 THSn_LTAV.LTAVV 值到高于 THSn_LTAV.LTARV 值，会置为 1。
26	R/W1C	0x0	LTAVF Low-Temperature-Alarm Valid Flag，低温警告有效标志 • 仅周期采样模式 (THSn_CFG. PSEN = 1) 时有效。 • 硬件置位，软件写 1 清零。 • 当温度传感器采样值从高于 THSn_LTAV.LTARV 值到低于 THSn_LTAV.LTAVV 值，会置为 1。
25	R/W1C	0x0	HTARF High-Temperature-Alarm Removed Flag，高温警告解除标志 • 仅周期采样模式（THSn_CFG. PSEN = 1）时有效。 • 硬件置位，软件写 1 清零。 • 当温度传感器采样值从高于 THSn_HTAV.HTAVV 值到低于 THSn_HTAV.HTARV 值，会置为 1。
24	R/W1C	0x0	HTAVF High-Temperature-Alarm Valid Flag，高温警告有效标志 • 仅周期采样模式（THSn_CFG. PSEN = 1）时有效。 • 硬件置位，软件写 1 清零。 • 当温度传感器采样值从低于 THSn_HTAV.HTARV 值到高于 THSn_HTAV.HTAVV 值，会置为 1。
23:18	–	–	–
17	R/W1C	0x0	DOVWF Data Overwrite Flag，数据覆盖标志 硬件置位，软件写 1 清零。当 DRDYF = 1（软件未清零数据有效标志），而且又有新的有效数据时，硬件置为 1，表示数据寄存器由于软件未及时读取，产生了上次有效数据被覆盖的情况。
16	R/W1C	0x0	DRDYF Data Ready Flag，数据有效标志 硬件置位，软件写 1 清零。当 THSn_DATA 寄存器有新的有效数据时，硬件置为 1，软件读取数据后，由软件操作清除。

默认值：0x00000000			THS 中断状态 (THS Channel N Interrupt)
位域	类型	默认值	描述
15:12	–	–	OTPIRE Over–Temperature–Protection Interrupt or Reset Enable，过温保护中断或复位使能 使能之后，当温度传感器采样值超过 THSn_OTPV.OTPV，THS 模块会输出中断或者系统复位（SYS_RSTn），从而引起 CPU 中断或者系统复位。 • 0x5: 过温保护中断使能 • 0xA: 过温保护复位使能 • Others: 过温保护禁止
11	R/W	0x0	LTARIE Low–Temperature–Alarm Removed Interrupt Enable，低温警告解除中断使能 当使能时，低温警告解除标志 (LTARF) 的状态就会传递到温度传感器的总中断。 • 0：禁止低温警告解除中断。 • 1：使能低温警告解除中断。
10	R/W	0x0	LTAVIE Low–Temperature–Alarm Valid Interrupt Enable，低温警告有效中断使能 当使能时，低温警告有效标志 (LTAVF) 的状态就会传递到温度传感器的总中断。 • 0：禁止低温警告有效中断。 • 1：使能低温警告有效中断。
9	R/W	0x0	HTARIE High–Temperature–Alarm Removed Interrupt Enable，高温警告解除中断使能 当使能时，高温警告解除标志 (HTARF) 的状态就会传递到温度传感器的总中断。 • 0：禁止高温警告解除中断。 • 1：使能高温警告解除中断。
8	R/W	0x0	HTAVIE High–Temperature–Alarm Valid Interrupt Enable，高温警告有效中断使能 当使能时，高温警告有效标志 (HTAVF) 的状态就会传递到温度传感器的总中断。 • 0：禁止高温警告有效中断。 • 1：使能高温警告有效中断。
7:2	–	–	–
1	R/W	0x0	DOVWIE Data Overwrite Interrupt Enable，数据覆盖中断使能 当使能时，数据覆盖标志 (DOVWF) 的状态就会传递到温度传感器的总中断。

默认值：0x00000000			THS 中断状态 (THS Channel N Interrupt)
位域	类型	默认值	描述
			• 0：禁止数据覆盖中断。 • 1：使能数据覆盖中断。
0	R/W	0x0	DRDYIE Data Ready Interrupt Enable，数据有效中断使能 当使能时，数据有效标志 (DRDYF) 的状态就会传递到温度传感器的总中断。 • 0：禁止数据有效中断。 • 1：使能数据有效中断。

8.1.5.9. 0x114+n*0x20 THSn_HTAV

默认值：0x00000000			THS 高温报警阈值设置 (Channel N High-Temperature Alarm Value)
位域	类型	默认值	描述
31	R/W	0	HTAE High-Temperature-Alarm Enable，高温警报使能 • 0：禁止高温警报功能，高温警报阈值无效。 • 1：使能高温警报功能，高温警报阈值有效。
30:28	–	–	–
27:16	R/W	0x000	HTARV High-Temperature-Alarm Remove Value，高温警报解除阈值 当温度从高于高温警报有效阈值 (HTAVV) 的数值，降到高温警报解除阈值（HTARV）以下，高温警报解除标志（HTARF）就会被硬件置位。 **注：** HTARV < HTAVV。
15:12	–	–	–
11:0	R/W	0x000	HTAVV High-Temperature-Alarm Valid Value，高温警报有效阈值 当温度从低于高温警报解除阈值（HTARV）的数值，升到高温警报有效阈值 (HTAVV) 以上，高温警报有效标志 (HTAVF) 就会被硬件置位。

默认值：0x00000000			THS 高温报警阈值设置 (Channel N High–Temperature Alarm Value)
位域	类型	默认值	描述
			注： HTARV < HTAVV。

8.1.5.10. 0x118+n*0x20 THSn_LTAV

默认值：0x00000000			THS 低温报警阈值设置 (Channel N Low–Temperature Alarm Value)
位域	类型	默认值	描述
31	R/W	0	LTAE Low–Temperature–Alarm Enable，低温警报使能 • 0：禁止低温警报功能，低温警报阈值无效。 • 1：使能低温警报功能，低温警报阈值有效。
30:28	–	–	–
27:16	R/W	0x000	LTARV Low–Temperature–Alarm Remove Value，低温警报解除阈值 当温度从低于低温警报有效阈值 (LTAVV) 的数值，升到低温警报解除阈值 (LTARV) 以上，低温警报解除标志 (LTARF) 就会被硬件置位。 注： LTAVV < LTARV。
15:12	–	–	–
11:0	R/W	0x000	LTAVV Low–Temperature–Alarm Valid Value，低温警报有效阈值 当温度从高于低温警报解除阈值 (LTARV) 的数值，降到低温警报有效阈值 (LTAVV) 以下，低温警报有效标志（LTAVF）就会被硬件置位。 注： LTAVV < LTARV。

8.1.5.11. 0x11C+n*0x20 THSn_OTPV

(6)

默认值：0x00000000			THS 过温保护阈值设置 (Channel N Over Temperature Protect Value)
位域	类型	默认值	描述
31	R/W	0	OTPE 过温保护使能 • 0：禁止过温保护功能，过温保护阈值无效。 • 1：使能过温保护功能，过温保护阈值有效。
30:12	–	–	–
11:0	R/W	0x000	OTPV 过温保护阈值，大于高温阀值 当温度上升到过温保护阈值 (OTPV) 以上，过温保护标志 (OTPF) 就会被硬件置位。

8.1.5.12. 0xFFC VERSION

默认值：0x00000110			THS 版本号 (THS Version)
位域	类型	默认值	描述
31:0	RO	0x00000110	Version（模块版本号） 采用 BCD 码显示，V1.1

8.2. Voltage Sensor (VTS)

Voltage Sensor (VTS)，电压传感器模块，主要用于监测芯片内部的电源电压，确保系统稳定运行。

VTS 模块通过 ADC 获取芯片内部电源的电压值。该模块能够实时监控多个关键电源电压，包括 VCC3V3、LDO25、LDO18 和 LDO1x，从而帮助开发者确保系统的电源管理处于最佳状态。在启动 VTS 模块前，需要先启动 ADC 模块。

8.2.1. 特性说明

- 支持多电压，能够监测 VCC3V3、LDO25、LDO18 和 LDO1x 四种内部电源电压。
- 支持灵活可配的 ADC 捕获时间。
- 支持单次读取和周期读取两种模式，适应不同的应用场景。
- 支持一次读取多个电压点，并进行算术平均滤波，提高数据准确性。

(6) n = 0~1 为 ths 通道号

• 支持高/低压报警，当检测到电压超出预设范围时，模块会触发中断，通知系统进行相应处理。

• 使用 ADC1 进行电压采集，确保数据采集的独立性和准确性。

8.2.2. 原理框图

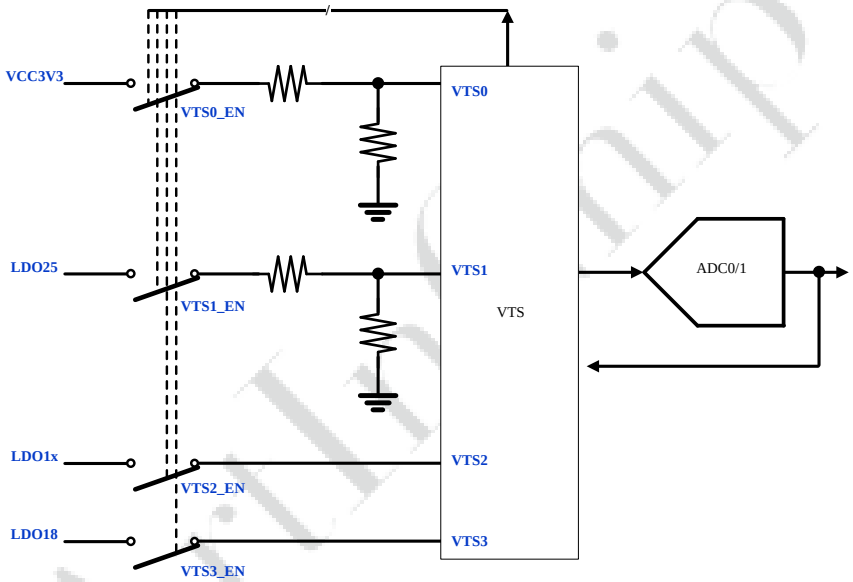


图 8–5 VTS 模块与 ADC 连接关系

VTS 模块通过内部连接与 ADC1 协同工作，实现对四路内置电源电压的检测。每路电压对应一个特定的 VTS 通道：

• VTS0：对应 VCC3V3/2 电压

• VTS1：对应 LDO25/2 电压

• VTS2：对应 LDO1x 电压

• VTS3：对应 LDO18 电压

8.2.3. 功能描述

8.2.3.1. 内部电源检测

VTS 模块通过内部连接与 ADC1 协同工作，实现对四路内置电源电压的检测。每路电压对应一个特定的 VTS 通道：

• VTS0：对应 VCC3V3/2 电压

• VTS1：对应 LDO25/2 电压

• VTS2：对应 LDO1x 电压

• VTS3：对应 LDO18 电压

VTS 模块对不同的内部电源电压采用不同的采样方式：

- **VCC3V3 和 LDO25**：需经过电阻分压（1/2）后输出给 ADC 进行采样转换。
- **LDO1x 和 LDO18**：无需电阻分压，直接输出给 ADC 进行采样转换。

内部电源电压 VCC3V3、LDO25、LDO1x 和 LDO18 的检测均可以通过寄存器控制是否使能。若不需要对某一路电源电压进行检测，需关闭对应的使能位，以避免分压电阻带来的额外功耗。

通过寄存器可以选择 ADC0 或 ADC1 进行电压的采样和转换。

8.2.3.2. 高低压报警阈值说明

为了防止电压在边界处反复波动时产生频繁报警，VTS 模块采用双阈值机制：

- 高压报警：
  - 高压报警有效阈值（HVAVV）：当电压超过此阈值时，触发高压报警。
  - 高压报警解除阈值（HVARV）：当电压低于此阈值时，解除高压报警。

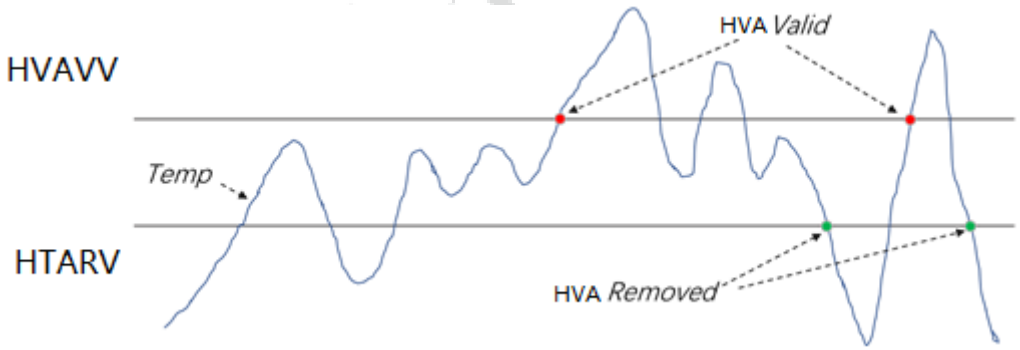


图 8–6 高压报警（HVA）双阈值判断

- 低压报警：
  - 低压报警有效阈值（LVAVV）：当电压低于此阈值时，触发低压报警。
  - 低压报警解除阈值（LVARV）：当电压高于此阈值时，解除低压报警。

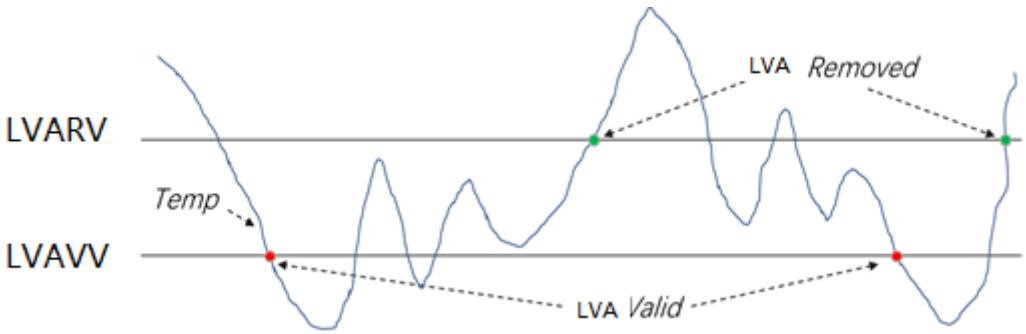


图 8–7 低压报警（LVA）双阈值判断

注：

- 通常情况下，LVAVV<LVARV<HVARV<HVAVV。
- 在周期读取模式下，仅对采样后的平均值进行阈值判断，而不对原始数据进行判断，以避免单次采样误触发。

VTS 使用配置流程

假设需要监测 VCC3V3 电压，并在电压超出 3.4V 范围时触发中断，3.2V 至 3.4V 波动触发一次中断：

1. 配置 ADC 模块并启动。
2. 设置 VTS 模块的控制寄存器：
   - 选择 VTS0 通道（对应 VCC3V3）。
   - 根据需求设置 ADC 捕获时间，例如 10 ms。
   - 启用高低压报警功能，设置 HVAVV 为 3.4 /2V，HVARV 为 3.2 /2V，

     注：

     关于详细说明，可查看高低压报警阈值说明。

   - 选择 ADC 通道：选择 ADC0 或 ADC1 进行电压采样和转换。
3. 根据应用需求，选择单次读取或周期读取模式，启动电压监测。例如，启动周期读取模式，每 100 ms 读取一次电压数据。
4. 如果启用了高低压报警功能，需配置中断服务程序以处理电压异常情况：
   - **高压报警**：记录日志或调整系统电源管理策略。
   - **低压报警**：记录日志或调整系统电源管理策略。

8.2.4. 寄存器列表

表 8–2 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	MCR	VTS 控制器模块控制寄存器	0x0000 MCR
0x100+n*0x20	VTSn_CFG	VTSn 配置寄存器	0x100+n*0x20 VTSn_CFG
0x104+n*0x20	VTSn_ITV	VTSn 周期采样的采样间隔寄存器	0x104+n*0x20 VTSn_ITV
0x108+n*0x20	VTSn_FIL	采样滤波配置寄存器	0x108+n*0x20 VTSn_FIL

表 8–2 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x10C+n*0x20	VTSn_DATA	VTSn 数据寄存器	0x10C+n*0x20 VTSn_DATA
0x110+n*0x20	VTSn_INT	VTSn 中断寄存器	0x110+n*0x20 VTSn_INT
0x114+n*0x20	VTSn_HTAV	VTSn 高压报警阈值寄存器	0x114+n*0x20 VTSn_HTAV
0x118+n*0x20	VTSn_LTAV	VTSn 低压报警阈值寄存器	0x118+n*0x20 VTSn_LTAV
0xFFC	VERSION	模块版本寄存器	0xFFC VERSION

8.2.5. 寄存器描述

8.2.5.1. 0x0000 MCR

默认值: 0x0000_0000			VTS 控制器模块控制寄存器
Bit	类型	默认值	描述
31:20	–	0x000	未定义
19	R/W	0	VTS3_EN，使能 VTS3，检测 LDO18 的内部电源电压 0：VTS3 禁止； 1：VTS3 使能。
18	R/W	0	VTS2_EN，使能 VTS2，检测 LDO1x 的内部电源电压 0：VTS2 禁止； 1：VTS2 使能。
17	R/W	0	VTS1_EN，使能 VTS1，检测 LDO25 的内部电源电压 0：VTS1 禁止； 1：VTS1 使能。
16	R/W	0	VTS0_EN，使能 VTS0，检测 VCC3V3 的内部电源电压 0：VTS0 禁止； 1：VTS0 使能。
15:1	–	0x0	未定义
0	R/W	0	EN（模块使能） 0：本模块被禁止，本模块停止工作；

默认值: 0x0000_0000			VTS 控制器模块控制寄存器
Bit	类型	默认值	描述
			1：本模块被使能。

8.2.5.2. 0x004 MINTR

默认值：0x00000000			VTS 中断控制与状态 (VTS Module Interrupt Register)
位域	类型	默认值	描述
31:20	–	–	–
19	RO	0	VTS3_INTF VTS3 Interrupt Flag，VTS3 中断标志，只可读。 只要清除了相应 VTS 的中断标志，就会自动清零 • 0: VTS3 无中断请求 • 1: VTS3 有中断请求
18	RO	0	VTS2_INTF（ ） VTS2 Interrupt Flag，VTS2 中断标志，只可读。 只要清除了相应 VTS 的中断标志，就会自动清零 • 0: VTS2 无中断请求 • 1: VTS2 有中断请求
17	RO	0	VTS1_INTF VTS1 Interrupt Flag，VTS1 中断标志，只可读。 只要清除了相应 VTS 的中断标志，就会自动清零 • 0: VTS1 无中断请求 • 1: VTS1 有中断请求
16	RO	0	VTS0_INTF（ ） VTS0 Interrupt Flag，VTS0 中断标志，只可读。 只要清除了相应 VTS 的中断标志，就会自动清零 • 0: VTS0 无中断请求 • 1: VTS0 有中断请求
15:4	–	–	–
3	R/W	0	VTS3_INTEN VTS3 Interrupt Enable，VTS3 中断使能信号

默认值：0x00000000			VTS 中断控制与状态 (VTS Module Interrupt Register)
位域	类型	默认值	描述
			• 0：禁止 VTS3 中断 • 1：使能 VTS3 中断
2	R/W	0	VTS2_INTEN VTS2 Interrupt Enable，VTS2 中断使能信号 • 0：禁止 VTS2 中断 • 1：使能 VTS2 中断
1	R/W	0	VTS1_INTEN VTS1 Interrupt Enable，VTS1 中断使能信号 • 0：禁止 VTS1 中断 • 1：使能 VTS1 中断
0	R/W	0	VTS0_INTEN VTS0 Interrupt Enable，VTS0 中断使能信号 • 0：禁止 VTS0 中断 • 1：使能 VTS0 中断

8.2.5.3. 0x008 VTS_NUM

默认值：0x00000004			VTS 数量 (VTS Number)
位域	类型	默认值	描述
31:3	–	–	–
2：0	RO	0x4	VTS_MAX VTS 最大编号，规定了 VTSn 中 控制器通道数量。n 小于等于 4，即传感器数量最少为 1 个，最多为 4 个。

8.2.5.4. 0x100+n*0x20 VTSn_CFG

默认值: 0x0000_2F00			VTSn 配置寄存器
Bit	读写类型	默认值	寄存器域描述
31:28	–	–	–
27	R/W	0x0	INVERTED SELECT

默认值: 0x0000_2F00			VTSn 配置寄存器
Bit	读写类型	默认值	寄存器域描述
			• 0：取反 • 1：不变
26:5	–	–	–
4	R/W	0x0	ADC_PRI ADC 采样优先级，本次 ADC 采样的优先级控制。 0 – 表示低优先级非实时采样，采样命令会与其它非实时采样命令进入采样队列； 1 – 表示高优先级实时采样，采样命令会被直接处理，不进入采样队列
3	R/W	0x0	VTS_CMD_TYP VTS Command Type，选择当前 VTS 的采样转换命令类型 0x0：主动模式，主动发送请求 ADC 转换的命令 0x1：被动模式，通过 ADC 的队列请求控制
2	R	0x0	VTS _STATE 0x0：空闲 0x1：工作
1	R/W	0x0	PSEN（Periodic Sample Enable，周期采样使能） 本字段功能为使能 VTS 传感器周期采样。 由软件置位，表示启动响应传感器的电压周期性采样，采样间隔由 VTSn_ITV 寄存器设定。软件清零表示停止周期性采样，如果在软件清零之前硬件已经启动了一次新的采样，那么这次采样会继续完成，并最终返回数据。 *注意：不允许同时使能周期采样与单次采样，即 SSEN 与 PSEN 不可以同时有效。*
0	R/WAC	0	SSEN（Single Sample Enable，单点采样使能） 本字段功能为使能 VTS 传感器单点采样。 由软件置位，表示启动相应传感器的电压采样，采样命令被 ADC 接收后，本字段就会由硬件自动清零。自动清零后，表示采样命令已经被接收，但并不代表采样数据已经返回，需要等到数据有效后，才意味着电压采样完成。 注意：尽管本字段描述为“单点”采样，但由于为了避免单次采样误差，可能需要通过单点多次采样后的数据进行算术平均，并得到一个相对稳定的电压值，所以本“单点”采样实际上会包含“多次”采样，对应的由硬件向 ADC 接口控制器连续发出多个采样命令，连续发出采样命令的个数由滤波器配置 VTSn_FIL 设定。 例如滤波器配置为 0x1（2 点平均），那么每次启动 SSEN，硬件会连续发出 2 个采样命令，2 个采样命令发出后，硬件自动清零 SSEN。

8.2.5.5. 0x104+n*0x20 VTSn_ITV

默认值: 0x016E_ffff			VTSn 周期采样的采样间隔寄存器
Bit	**读写类型**	**默认值**	**寄存器域描述**
31:16	R/W	0x016E	ITV（Interval，采样间隔） 此字段定义了周期采样的采样间隔。单位为 65536*T_{PCLK}，即 采样间隔=ITV* Multiple *T_{PCLK}。 假设 PCLK=24MHz，ITV=0x16E=366，采样间隔=366*65536*41.67ns=1s。 *注意：此字段不可为 0。硬件自动保护：当软件配置为 0 时，效果等同于 1。*
15:0	R/W	0xFFFF	Multiple:65536=0xffff+1，软件可配

8.2.5.6. 0x108+n*0x20 VTSn_FIL

默认值: 0x0000_0000			采样滤波配置寄存器
Bit	**读写类型**	**默认值**	**寄存器域描述**
31:2	–	0x0000 0000	未定义
1:0	R/W	0x0	FIL（Filter，滤波器配置） 本字段定义了电压传感器采样数据算数平均滤波器的输入样本数。 0x0：无滤波； 0x1：2 0x2：4 0x3：8 本字段也定义了单次采样和周期采样每次的连续采样命令数量（采样点数），本控制器接收对应的采样点数据后，进行算数平均，并把平均值通过本传感器的 VTSn_DATA 寄存器返回。

8.2.5.7. 0x10C+n*0x20 VTSn_DATA

默认值: 0x0000_0000			VTSn 数据寄存器
Bit	**读写类型**	**默认值**	**描述**
31:16	–	0x00000	未定义
15:0	RO	0x000	DATA（ADC 采样数据） 此字段为 VTS 的 ADC 采样数据结果。如果 VTSn_FIL.FIL 为 0x0，则直接返回 ADC 的采样值；否则返回的是多个采样点的算数平均值。

默认值: 0x0000_0000			VTSn 数据寄存器
Bit	读写类型	默认值	描述
			注意：本模块无数据 FIFO，本寄存器只能暂存一次的采样结果，如果未被及时读取，则会被后来新的采样数据直接覆盖。

8.2.5.8. 0x110+n*0x20 VTSn_INT

默认值: 0x0000_0000 0x20			VTSn 中断寄存器
Bit	读写类型	默认值	寄存器域描述
31:28	–	0x0	未定义
27	R/W1C	0	LVARF（Low-Voltage-Alarm Removed Flag，低压警告解除标志） 本字段为低压警告解除标志，**仅周期采样模式（VTSn_CFG.PSEN=1）时有效**。本字段的操作为硬件置位，软件写 1 清零。 当电压传感器采样值从低于 VTSn_LVAV.LVAVV 值到高于 VTSn_LVAV.LVARV 值，就会置位本字段。
26	R/W1C	0	LVAVF（Low-Voltage-Alarm Valid Flag，低压警告有效标志） 本字段为低压警告有效标志，**仅周期采样模式（VTSn_CFG. PSEN=1）时有效**。本字段的操作为硬件置位，软件写 1 清零。 当电压传感器采样值从高于 VTSn_LVAV.LVARV 值到低于 VTSn_LVAV.LVAVV 值，就会置位本字段。
25	R/W1C	0	HVARF（High-Voltage-Alarm Removed Flag，高压警告解除标志） 本字段为高压警告解除标志，**仅周期采样模式（VTSn_CFG. PSEN=1）时有效**。本字段的操作为硬件置位，软件写 1 清零。 当电压传感器采样值从高于 VTSn_HVAV.HVAVV 值到低于 VTSn_HVAV.HVARV 值，就会置位本字段。
24	R/W1C	0	HVAVF（High-Voltage-Alarm Valid Flag，高压警告有效标志） 本字段为高压警告有效标志，**仅周期采样模式（VTSn_CFG. PSEN=1）时有效**。本字段的操作为硬件置位，软件写 1 清零。 当电压传感器采样值从低于 VTSn_HVAV.HVARV 值到高于 VTSn_HVAV.HVAVV 值，就会置位本字段。
23:18	–	0x00	未定义
17	R/W1C	0	DOVWF（Data Overwrite Flag，数据覆盖标志） 本字段为数据覆盖标志。本字段的操作为硬件置位，软件写 1 清零。

默认值: 0x0000_0000 0x20			VTSn 中断寄存器
Bit	读写类型	默认值	寄存器域描述
			当 DRDYF=1（软件未清零数据有效标志），而且又有新的有效数据时，硬件置位本字段，表示数据寄存器由于软件未及时读取，产生了上次有效数据被覆盖的情况。
16	R/W1C	0	DRDYF（Data Ready Flag，数据有效标志） 本字段为数据有效标志。本字段的操作为硬件置位，软件写 1 清零。 当 VTSn_DATA 寄存器有新的有效数据时，硬件置位本字段，软件读取数据后，由软件操作清除本字段。
–	–	–	–
11	R/W	0	LVARIE（Low-Voltage-Alarm Removed Interrupt Enable，低压警告解除中断使能） 本字段为低压警告解除中断使能。当本字段使能时，低压警告解除标志（LVARF）的状态就会传递到本电压传感器的总中断标志。 0：禁止低压警告解除中断； 1：使能低压警告解除中断。
10	R/W	0	LVAVIE（Low-Voltage-Alarm Valid Interrupt Enable，低压警告有效中断使能） 本字段为低压警告有效中断使能。当本字段使能时，低压警告有效标志（LVAVF）的状态就会传递到本电压传感器的总中断标志。 0：禁止低压警告有效中断； 1：使能低压警告有效中断。
9	R/W	0	HVARIE（High-Voltage-Alarm Removed Interrupt Enable，高压警告解除中断使能） 本字段为高压警告解除中断使能。当本字段使能时，高压警告解除标志（HVARF）的状态就会传递到本电压传感器的总中断标志。 0：禁止高压警告解除中断； 1：使能高压警告解除中断。
8	R/W	0	HVAVIE（High-Voltage-Alarm Valid Interrupt Enable，高压警告有效中断使能） 本字段为高压警告有效中断使能。当本字段使能时，高压警告有效标志（HVAVF）的状态就会传递到本电压传感器的总中断标志。 0：禁止高压警告有效中断； 1：使能高压警告有效中断。
7:2	–	0x00	未定义
1	R/W	0	DOVWIE（Data Overwrite Interrupt Enable，数据覆盖中断使能）

默认值: 0x0000_0000 0x20			VTSn 中断寄存器
Bit	读写类型	默认值	寄存器域描述
			本字段为数据覆盖中断使能。当本字段使能时，数据覆盖标志（DOVWF）的状态就会传递到本电压传感器的总中断标志。 0：禁止数据覆盖中断； 1：使能数据覆盖中断。
0	R/W	0	DRDYIE（Data Ready Interrupt Enable，数据有效中断使能） 本字段为数据有效中断使能。当本字段使能时，数据有效标志（DRDYF）的状态就会传递到本电压传感器的总中断标志。 0：禁止数据有效中断； 1：使能数据有效中断。

8.2.5.9. 0x114+n*0x20 VTSn_HTAV

默认值: 0x0000_0000			VTSn 高压报警阈值寄存器
位域	类型	默认值	描述
31	R/W	0	HVAE（过压警报使能） 0：禁止高压警报功能，高压警报阈值无效； 1：使能高压警报功能，高压警报阈值有效。
30	R/W	–	未定义
29:16	R/W	0x000	HVARV（高压警报解除阈值） 此字段规定了高压警报解除阈值，当电压从高于高压警报有效阈值（HVAVV）的数值，降到高压警报解除阈值（HVARV）以下，高压警报解除标志（HVARF）就会被硬件置位。 *注意：HVARV< HVAVV。*
15:14	R/W	–	未定义
13:0	R/W	0x000	HVAVV（高压警报有效阈值） 此字段规定了高压警报有效阈值，当电压从低于高压警报解除阈值（HVARV）的数值，升到高压警报有效阈值（HVAVV）以上，高压警报有效标志（HVAVF）就会被硬件置位。 *注意：HVARV< HVAVV。*

8.2.5.10. 0x118+n*0x20 VTSn_LTAV

默认值: 0x0000_0000			VTSn 低压报警阈值寄存器
位域	类型	默认值	描述
31	R/W	0	LVAE（低压警报使能） 0：禁止低压警报功能，低压警报阈值无效； 1：使能低压警报功能，低压警报阈值有效。
30	–	–	未定义
29:16	R/W	0x000	LVARV（低压警报解除阈值） 此字段规定了低压警报解除阈值，当电压从低于低压警报有效阈值（LVAVV）的数值，升到低压警报解除阈值（LVARV）以上，低压警报解除标志（LVARF）就会被硬件置位。 *注意：LVAVV< HVARV。*
15:14	–	0x0	未定义
13:0	R/W	0x000	LVAVV（低压警报有效阈值） 此字段规定了低压警报有效阈值，当电压从高于低压警报解除阈值（LVARV）的数值，降到低压警报有效阈值（LVAVV）以下，低压警报有效标志（LVAVF）就会被硬件置位。 *注意：LVAVV< HVARV。*

8.2.5.11. 0xFFC VERSION

默认值: 0x0000_0100			模块版本寄存器
位域	类型	默认值	描述
31:0	RO	0x00000100	Version（模块版本号） 采用 BCD 码显示，V1.00

9. 伺服系统

9.1. Pulse Width Modulation Control System (PWMCS)

PWM Control System (PWMCS)，PWM 控制系统，由以下子系统组成：

- EPWM：用于产生 PWM 控制信号，具有 12x EPWM 模块 (EPWM0/1/.../11)
- CAP：用于捕捉输入信号或产生简易 PWM 信号。具有 6x CAP 模块（CAP0/1/…/5）
- QEP：用于接收增量编码器的信号，读取编码器位置数据和测量转速。具有 10x QEP 模块（QEP0/1/…/9）

9.1.1. 特性说明

EPWM 模块 Feature

- 16 比特的 PWM 计数器
- 支持递增、递减以及先递增后递减的计数方式
- 计数器支持同步输入信号以及同步输出信号
- PWM 输出脉宽可调节
- 每个模块均支持两路的 PWM 输出
- 支持软件强制控制 PWM 输出
- 支持死区时间控制
- 支持六路的故障指示输入
- 支持不可恢复以及可恢复的两种 PWM 保护功能响应故障
- 支持硬件触发 ADC 采样以及硬件触发编码器获取位置值
- 支持中断信号
- 支持高精度 PWM，精度为 130 ps
- 支持脉冲输出功能
- 支持 PWM 触发信号输出
- 支持 Debug Hold 功能
- 支持 DMA 功能

CAP 模块 Feature

- 每个模块支持 1x 捕获输入信号或 1x 普通 PWM 输出信号
- 32 比特的 CAP 计数器

- 支持 4x 捕获寄存器
- 可支持连续的捕获模式或单次的捕获模式
- 支持普通的 PWM 模式
- PWM 模式下支持相位的同步

QEP 模块 Feature

- 具有 32 比特的位置计数器、32 比特的定时器、32 比特的看门狗定时器、16 比特的 QEP 捕获寄存器
- 支持正交信号的解码
- 支持位置计数器的锁存
- 支持时间计数器的锁存
- 支持 M 法、T 法和 M/T 法测速

9.1.2. 实时控制外设互联图——EPWM

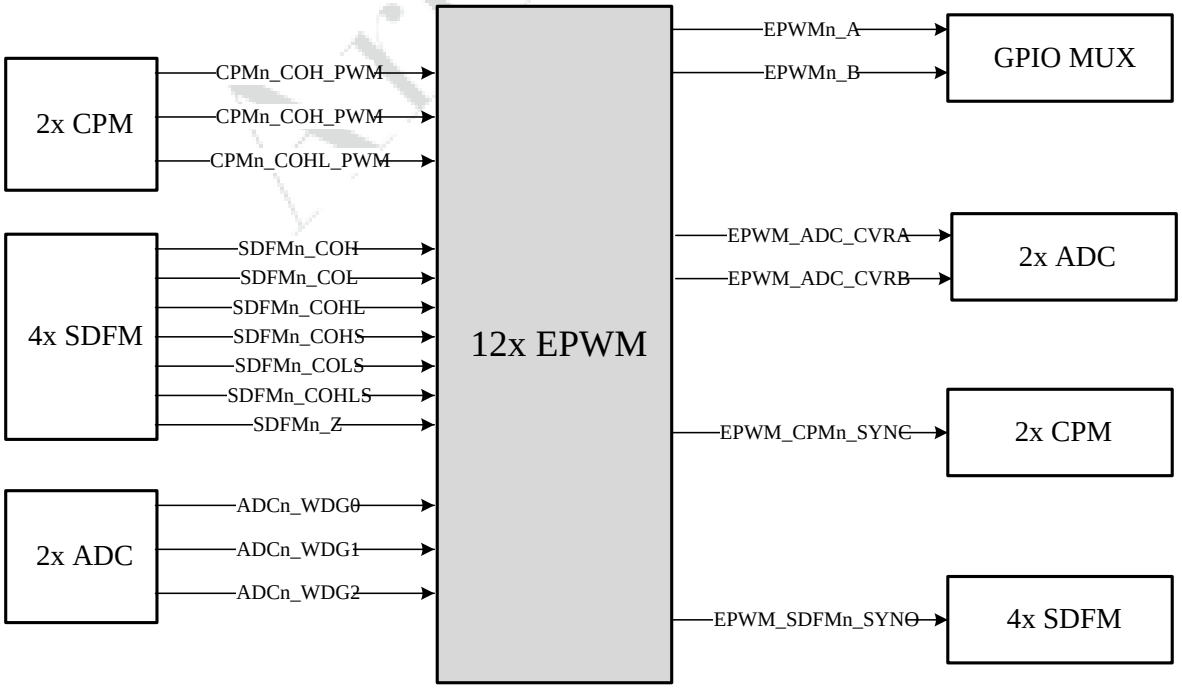


图 9–1 EPWM 模块与其它实时控制外设的连接关系

EPWM 的输入

- 比较器模块 CPM：输出的信号 CPMn_COH_PWM、CPMn_COL_PWM 和 CPMn_COHL_PWM（n=0/1），送到 EPWM 模块，可以用于触发 EPWM 的输出保护、触发 EPWM 的动作以及用于 EPWM 的同步网络输入。
- Sigma-Detla 滤波模块 SDFM：输出的 SDFMn_COH、SDFMn_COL 、 SDFMn_COHL、SDFMn_COHS、SDFMn_COLS、SDFMn_COHLS 和 SDFMn_Z（n=0/1/2/3），送到 EPWM 模块，可以用于触发 EPWM 的输出保护 。
- ADC：输出的 ADCn_WDG0、ADCn_WDG1 和 ADCn_WDG2（n=0/1），送到 EPWM 模块，可以用于触发 EPWM 的输出保护 。

EPWM 的输出

- EPWM 的输出信号 EPWMn_A 和 EPWMn_B 为 PWM 输出信号，直接通过 GPIO 复用传输到引脚。
- EPWM 的输出信号 EPWM_ADC_CVRA 和 EPWM_ADC_CVRB，传输到 ADC 模块，分别用于触发 ADC 的队列 0 和队列 1 工作的信号。
- EPWM 的输出信号 EPWM_CPMn_SYNC（n=0/1/2/…/11），传输到 CPM 模块，作为 CPM 模块的同步信号源。
- EPWM 的输出信号 EPWM_SDFMn_SYNC（n=0/1/2/…/11），传输到 SDFM 模块，作为 SDFM 模块的同步信号源。
- EPWM 的输出信号 EPWM_CVRC 和 EPWM_CVRD，传输到绝对值编码器 模块，用于触发绝对值编码器获取数据。

9.1.3. 实时控制外设互联图——CAP

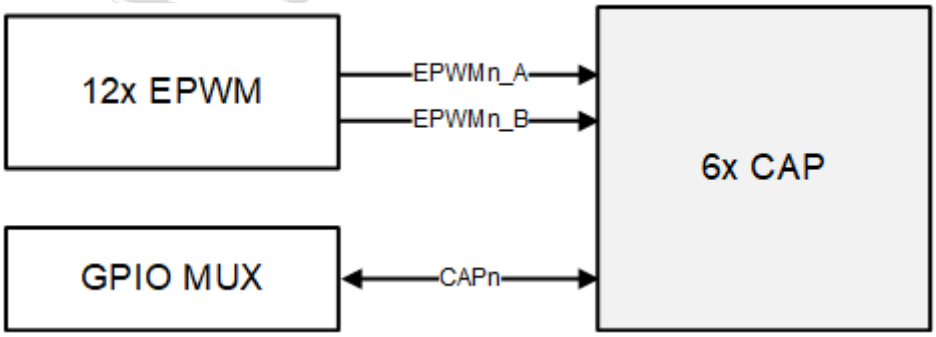


图 9–2 CAP 模块与其它实时控制外设的连接关系

CAP 的输入

- EPWM 模块的输出信号 EPWMn_A 和 EPWMn_B，能够传输到 CAP 模块，作为 CAP 模块输入信号源。
- CAP 的输入信号源，也可以选择来自引脚的 CAPn（n=0/1/…/5）信号。

9.1.4. 功能描述

9.1.4.1. EPWM 模块

EPWM 子模块 EPWM0/1/.../11为相同模块，各个子模块的信号都是相同的，标号 0/1/2… 用于区分不同模块的信号。例如 CNT_CLK0 代表 EPWM0 的计数器时钟信号 CNT_CLK。

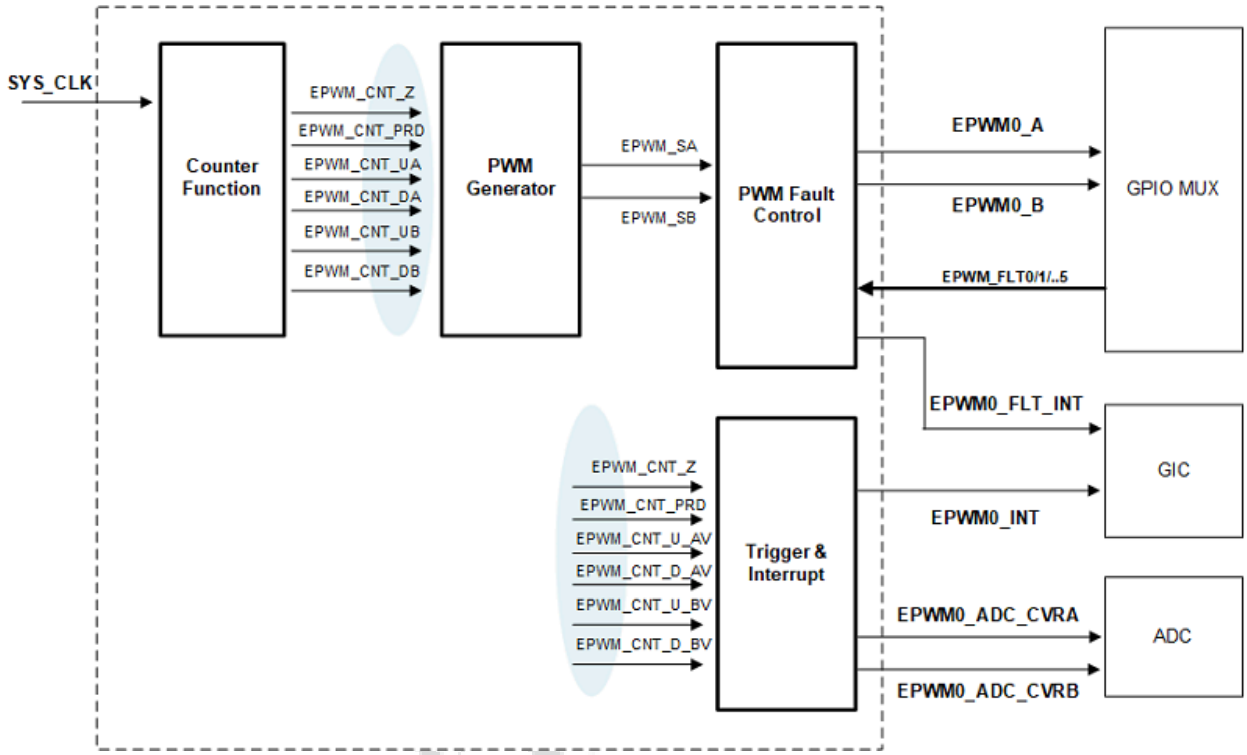


图 9–3 EPWM0 的结构框图

- 图 9–3 : EPWM0 的结构框图中，SYS_CLK 为工作时钟。
- EPWM_FLT0/1/2/3/4/5 输入信号由所有 EPWM 子模块共用，用于可恢复和不可恢复的故障输入指示。
- Counter Function 模块，为计数器的功能模块，负责计数器计数、计数器同步、计数器比较等功能。
- EPWM Generator 模块，为 EPWM 的生成模块，负责 EPWM 脉宽控制、EPWM 死区控制等功能。
- EPWM Fault 模块，为 EPWM 故障控制模块，负责在故障信号生效的时候实施保护功能以及产生中断。
- Trigger & Interrupt 模块，为 ADC 触发信号以及中断信号的产生模块。

9.1.4.1.1. 计数器功能模块

9.1.4.1.1.1. 计数时钟

计数器的时钟 EPWM_CNT_CLK 由 SYS_CLK 进行分频得出：

EPWM_CNT_CLK = SYS_CLK / (Divisor1 * Divisor2)

- *<Divisor1/2>* 为寄存器 EPWM_CLK_DIV1/2。
  - 除数 *<Divisor1>* 的可配置范围为 1/ 2/ 4/ 6/ 8/ 10/ 12/ 14。
  - 除数 *<Divisor2>* 的可配置范围为 1/ 2/ 4/ 8/ 16/ 32/ 64/ 128。
- 开启 EPWM 计数器的计数，需使能全局的控制信号 EPWM_CNT_GLB_EN。

9.1.4.1.1.2. 计数方式

寄存器字段 EPWM_CNT_MOD ，可以配置计数器的以下计数方式：

- 递增计数：计数器从 0 开始递增计数，计数至 EPWM_CNT_PRDV 值时，完成一个周期的计数，并且复位至 0 重新开始计数。

  图 9–4 : 计数器递增计数图示 中，设置 EPWM_CNT_PRDV = 6 的递增计数，此时开始从 0 到 6 的递增计数：

  - 在计数值为 0 时，信号 EPWM_CNT_Z = 1
  - 在计数值为 EPWM_CNT_PRDV 的时候，信号 EPWM_CNT_PRD = 1
  - 而指示计数方向的信号 EPWM_CNTD 一直为高电平，指示为递增计数。

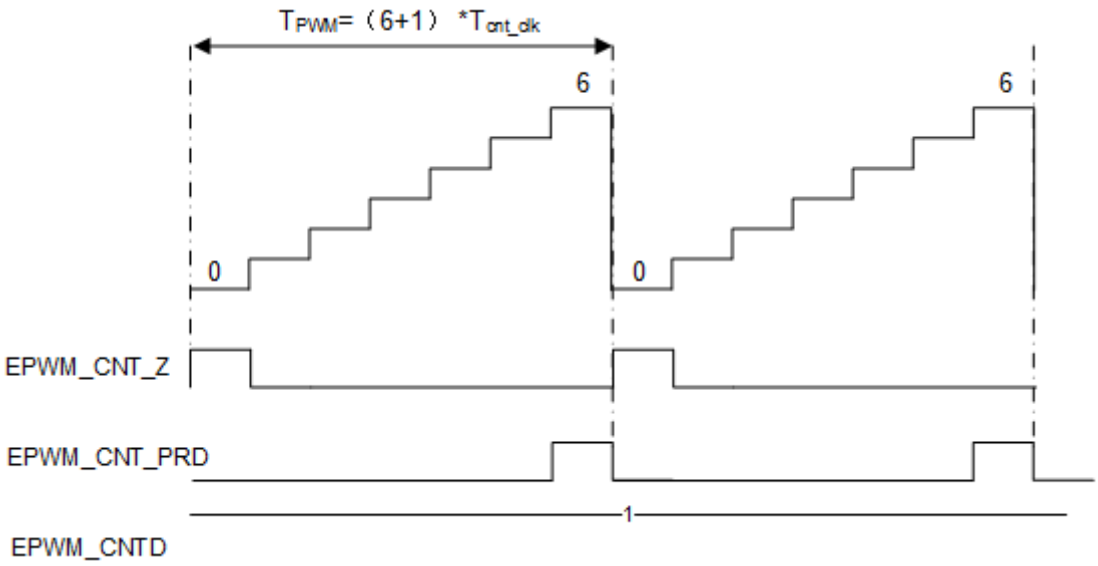


图 9–4 计数器递增计数图示

- 递减计数：计数器从 EPWM_CNT_PRDV 开始递减计数，计数至 0 时，完成一个周期的计数，并且复位至 EPWM_CNT_PRDV 重新开始计数。

  图 9–5 : 计数器递减计数图示中，设置 EPWM_CNT_PRDV = 6 的递减计数，此时进行从 6 到 0 递减计数：

  - 在计数值为 0 的时候，信号 EPWM_CNT_Z = 1。
  - 在计数值为 EPWM_CNT_PRDV 的时候，信号 EPWM_CNT_PRD = 1。
  - 指示计数方向的信号 EPWM_CNTD 一直为低电平，指示为递减计数。

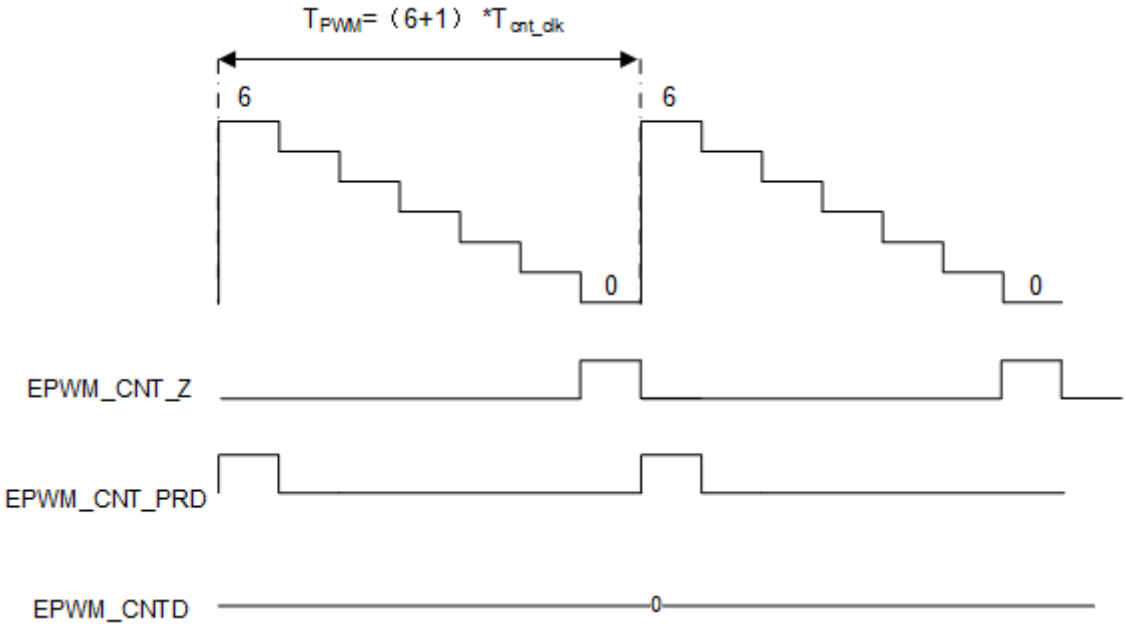


图 9–5 计数器递减计数图示

• 先递增后递减计数：计数器从 0 开始递增计数，计数至 EPWM_CNT_PRDV 后进行递减计数，当计数器计数为 1，完成一个周期的计数。下一个周期继续从 0 开始进行先递增后递减计数。

图 9–6 : 计数器先递增后递减计数示例中，设置 EPWM_CNT_PRDV = 6 的先递增再递减计数，此时进行先从 0 到 6 的递增计数再从 6 到 0 的递减计数：

- 在计数值为 0 的时候，信号 EPWM_CNT_Z = 1。
- 在计数值为 EPWM_CNT_PRDV 的时候，信号 EPWM_CNT_PRD = 1。
- 指示计数方向的信号 EPWM_CNTD 在前半周期为高电平指示此时为递增计数，后半周期指示为递减计数。

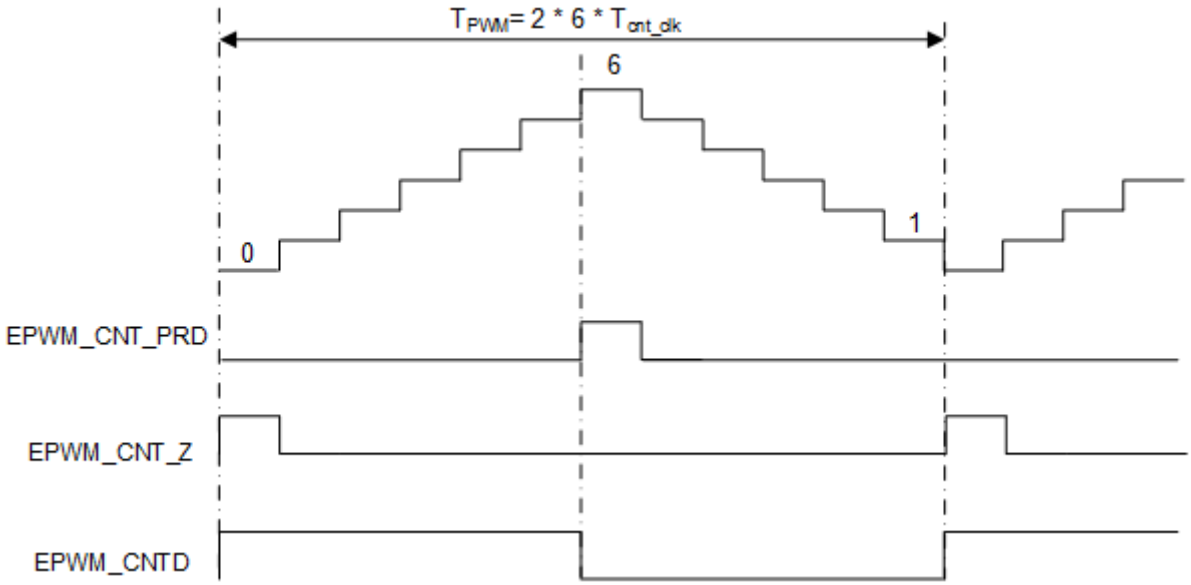


图 9–6 计数器先递增后递减计数示例

9.1.4.1.1.3. 计数比较

在计数数值达到预设的设定值时，即达到 EPWM_CNT_AV 与 EPWM_CNT_BV 时，计数器可以产生相应的信号 EPWM_CNT_A 与 EPWM_CNT_B。

为了区分计数的方向：

- 若为递增计数至 EPWM_CNT_AV 产生的信号，记为 EPWM_CNT_UA。
- 若为递减计数至 EPWM_CNT_AV 产生的信号，记为 EPWM_CNT_DA。
- 若为递增计数至 EPWM_CNT_BV 产生的信号，记为 EPWM_CNT_UB。
- 若为递减计数至 EPWM_CNT_BV 产生的信号，记为 EPWM_CNT_DB。

产生的信号输出至 EPWM 生成模块以及 Trigger & Interrupt 模块使用。具体时序图如下：

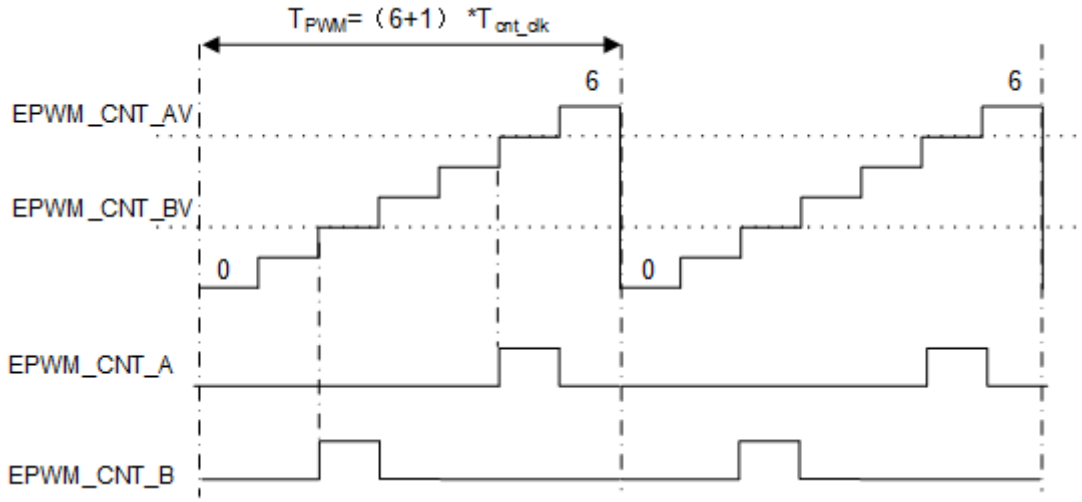


图 9–7 递增计数产生比较信号示例

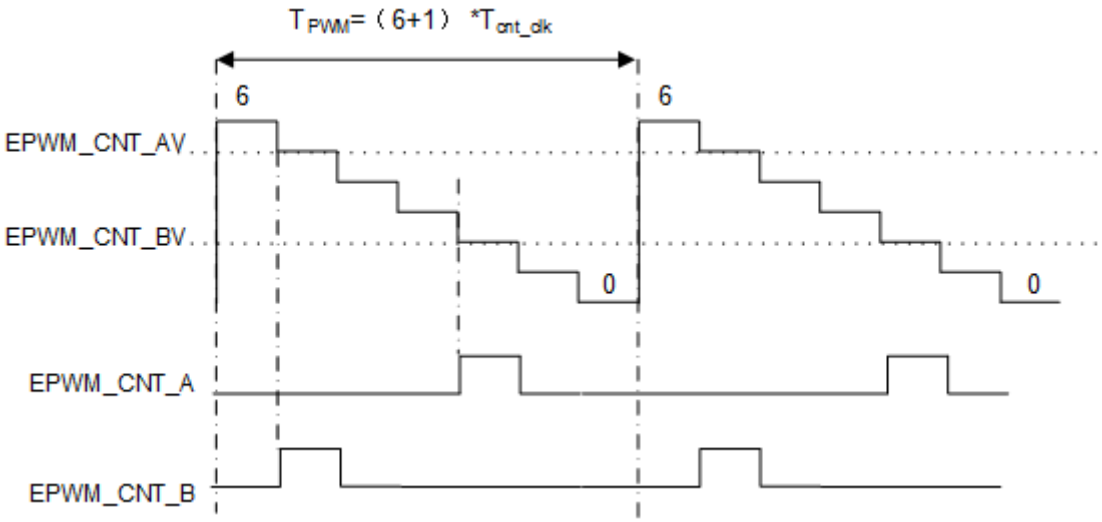


图 9–8 递减计数产生比较信号示例

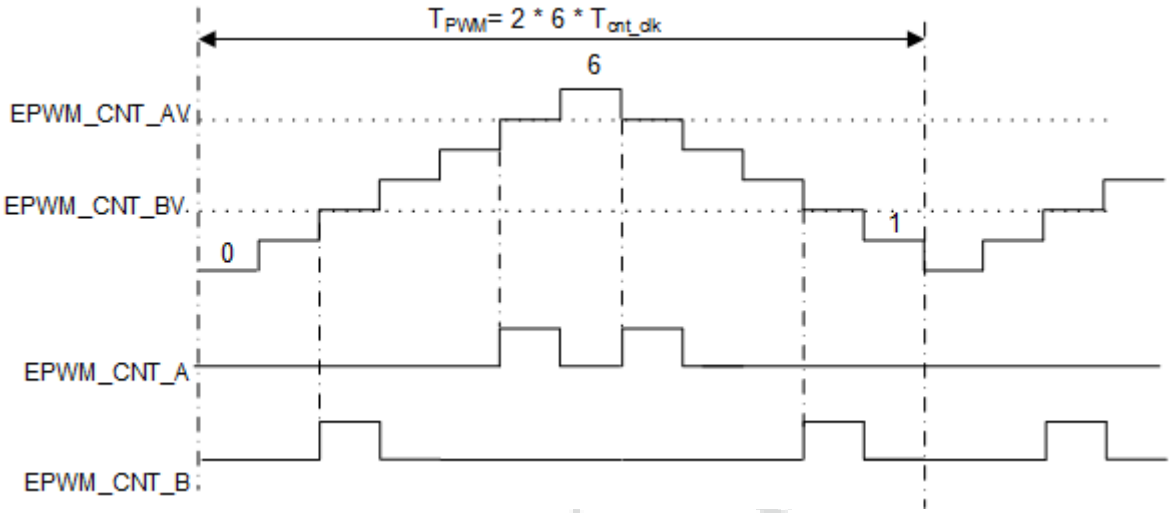


图 9–9 先递增后递减计数产生比较信号示例

9.1.4.1.1.4. 计数同步

计数器具有同步的功能，同步信号的使能后，可以使得计数器计数值跳至 EPWM_CNT_PHV 计数同步相位寄存器的数值。

计数器同步信号的来源有两个：

- IN_SYNC，对于 EPWM 子模块，是外部输入信号，各个 EPWM 子模块的 IN_SYNC 的信号来源可参考图 9–11 : 同步信号网络示例。
- SW_SYNC，为 EPWM 模块通过寄存器设置 EPWM_SW_FRC_SYNC 位而产生，即由软件控制产生。
- IN_SYNC 与 SW_SYNC 的信号经过或逻辑后，决定计数器的同步。
- ETRG0、ETRG1 均可作为相位同步的触发信号，如图 9–10 : 同步输出信号源所示。

计数器还会产生输出的同步信号 OUT_SYNC，通过寄存器设置可以选择不同的信号作为 OUT_SYNC。图 9–10 : 同步输出信号源 中展示了 OUT_SYNC 的来源。

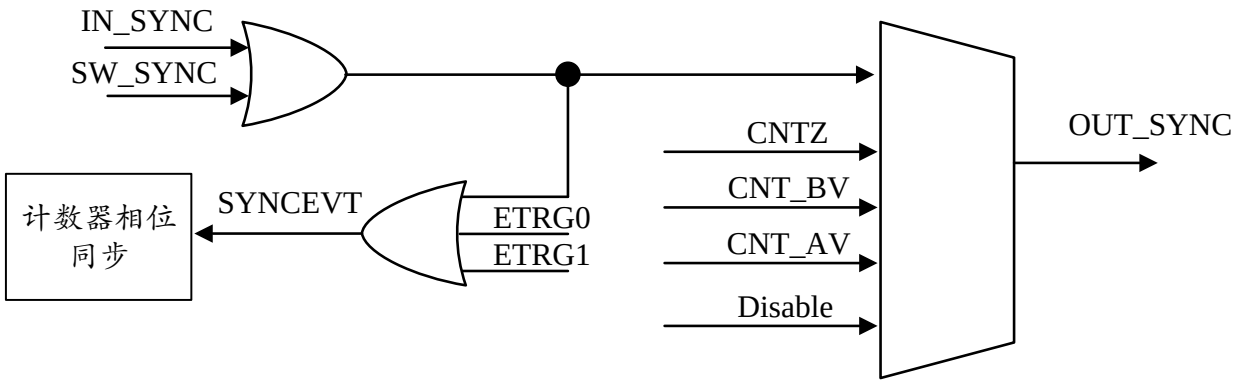


图 9–10 同步输出信号源

EPWM 模块利用输入同步信号以及输出同步信号，将所有子模块 EPWM0/1/.../11 进行计数器的相位同步。图 9–11 : 同步信号网络示例 展示了各个模块的同步连接方式。

- EPWM0/3/6/9 以及 CAP0 的输出同步信号通过 GPIO_MUX 输出给外部器件，并且进行脉冲的 8*SYS_CLK 的长度延长。
- CPM0_COH 以及 CPM1_COH 分别为 CPM0 和 CPM1 输出的比较信号。

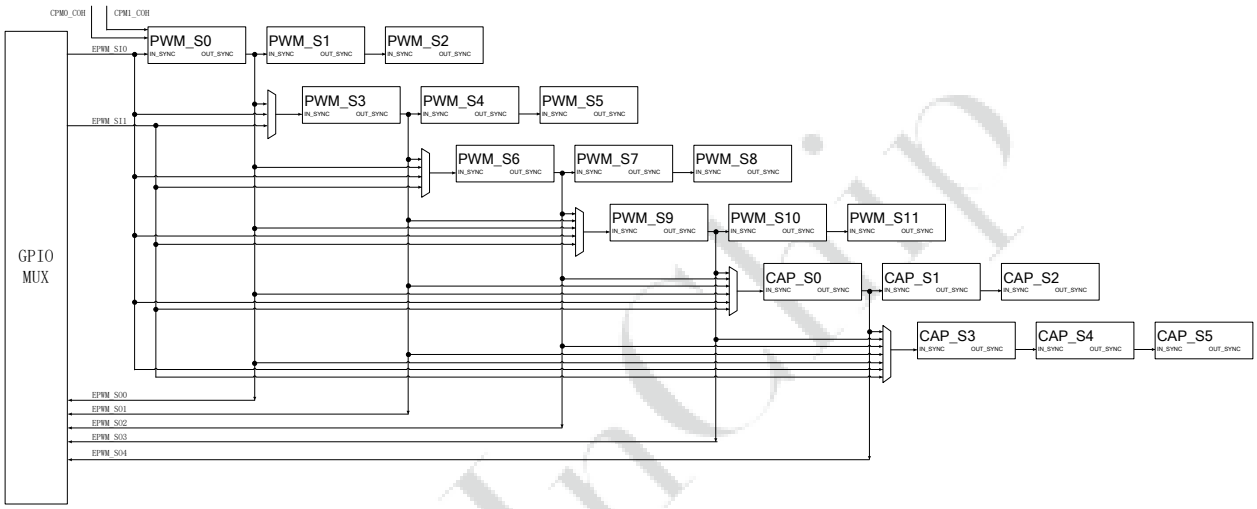


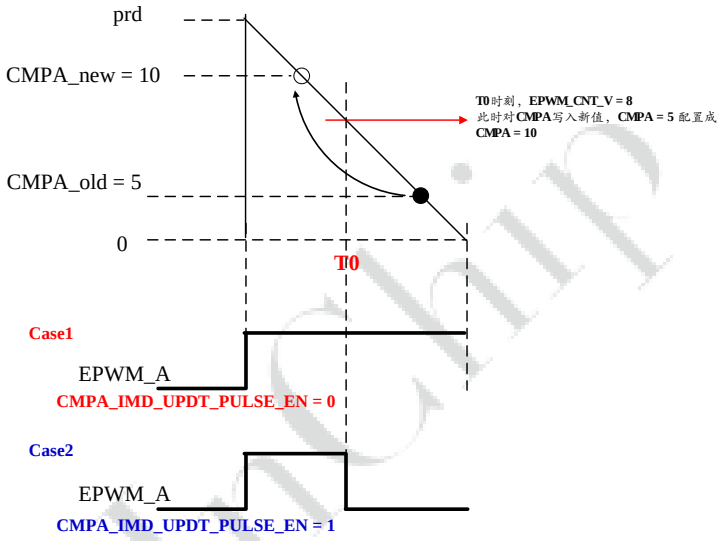
图 9–11 同步信号网络示例

9.1.4.1.1.5. 寄存器更新

寄存器 EPWM_CNT_PRDV、EPWM_CNT_AV 以及 EPWM_CNT_BV 支持以下更新模式，且各有独立控制的更新模式的控制位。

- 立即更新模式：
  - 对寄存器 EPWM_CNT_PRDV、EPWM_CNT_AV 以及 EPWM_CNT_BV 的数值写入，会立即生效，从而影响当前的 EPWM 生成。
  - 可能出现在当前的周期，新写入的数值与计数器的数值错过比较时刻，导致 AV 和 BV 的动作未发生，进而造成当前周期的输出 EPWM 占空比异常，例如 100% 或 0% 占空比。
  - 通过设置寄存器字段 EPWM_CMPA_IMD_UPDT_PULSE_EN = 1、EPWM_CMPB_IMD_UPDT_PULSE_EN = 1，可以避免立即更新带来的错误 EPWM 输出。举例如下图所示，CMPA 的更新采用立即更新，假设 EPWM_A 的输出在 EPWM_CNT_Z 点的动作为拉高动作，而在 EPWM_CNT_DA 的动作为拉低动作，在下图的 T0 时刻，计数器的数值为 8，而此时发生 CMPA 的数值修改，从 5 配置成 10。在 CMPA_IMD_UPDT_PULSE_EN = 0 的情况下（即下图的 Case1），那么 EPWMA 的输出就会持续拉高没有进行拉低的动作；而在

CMPA_IMD_UPDT_PULSE_EN = 1 的情况下（即下图的 Case2），EPWM 会在 T0 的时刻进行输出的拉低动作，避免了出现输出 100%占空比的情况。



• 影子寄存器更新模式：

◦ 对寄存器 EPWM_CNT_PRDV、EPWM_CNT_AV 以及 EPWM_CNT_BV 的数值的写入，暂时存入各自的影子寄存器，不会立即生效，只有当计数器计数值到指定加载点时才进行更新，影子寄存器的数值生效。

◦ 图 9–12 : EPWM_CNT_BV 影子寄存器更新示例中，当对 EPWM_CNT_BV 的数值进行写操作，将数值从 0x2 改变为 0x4 后，并没有立即生效，而是在计数值复位至 0 开始下一个周期的计数后，EPWM_CNT_BV 的数值才生效，从而改变了 EPWM 信号的占空比。

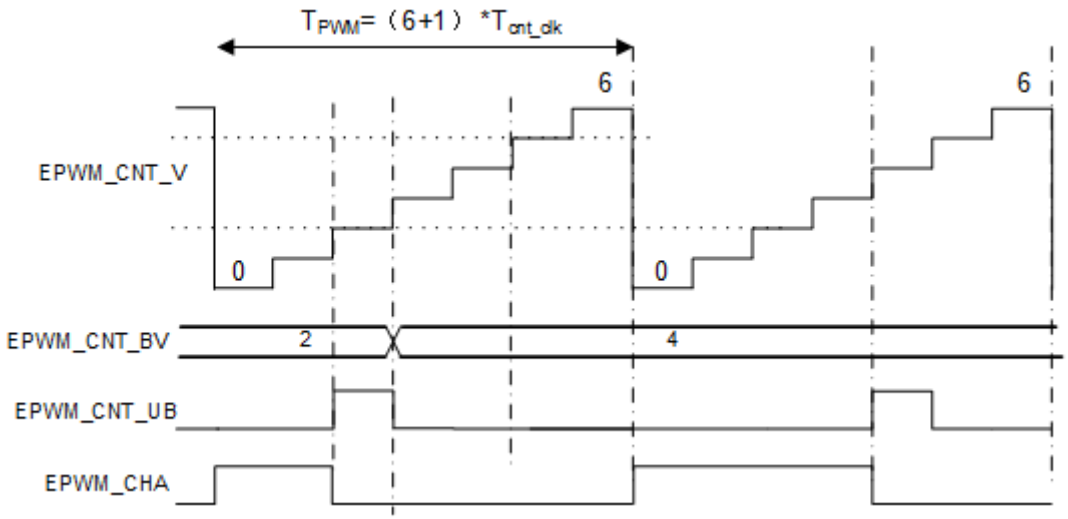


图 9–12 EPWM_CNT_BV 影子寄存器更新示例

• 多寄存器在同一时刻更新模式：

通过 EPWM_GLB_CTL 寄存器选择更新时刻，EPWM_GLB_CFG 寄存器选择需要全局更新的寄存器（如下所示），那么即可以实现在同一时刻对下述的并选中的寄存器，进行同时更新寄存器数值的功能。

- EPWM_CNT_PRD
- EPWM_CNT_CMPA
- EPWM_CNT_CMPB
- EPWMA_ACT
- EPWMB_ACT
- EPWM_ACT_SW_CT
- EPWM_DB_CTL
- EPWM_CNT_CMPC
- EPWM_CNT_CMPD
- HRPWM_VAL
- HRPWM_PRD_VAL
- EPWM_RE_DZCTL
- EPWM_FE_DZCTL

• 多个 EPWM 模块寄存器同时更新

EPWM 支持将寄存器链接到其它 EPWM，实现多个 EPWM 同时更新寄存器，支持的链接的寄存器如下所示。现在若需实现 EPWM0 和 EPWM1 的 EPWM_CNT_AV 的寄存器同时修改，只需通过配置寄存器 EPWM_LINK_CTL，将 EPWM1 的 EPWM_CNT_AV 链接到 EPWM0，那么在 EPWM0 修改寄存器 EPWM_CNT_AV 的时候，硬件也会自动对 EPWM1 的 EPWM_CNT_AV 寄存器进行修改。

- EPWM_CNT_PRD
- EPWM_CNT_CMPA
- EPWM_CNT_CMPB
- EPWM_RE_DZCTL
- EPWM_FE_DZCTL
- EPWM_CLB_CTL2
- HRPWM_PRD_VAL

9.1.4.1.1.6. 计数器单次计数模式

单次计数模式，通过寄存器 EPWM_CNT_SIG_EN = 0x1 设置，单次计数的方式如下：

• 如果是递增计数的方式，从 0 开始递增计数，计数至 EPWM_CNT_PRD 后，计数器不会继续计数，计数器数值维持 EPWM_CNT_PRD。

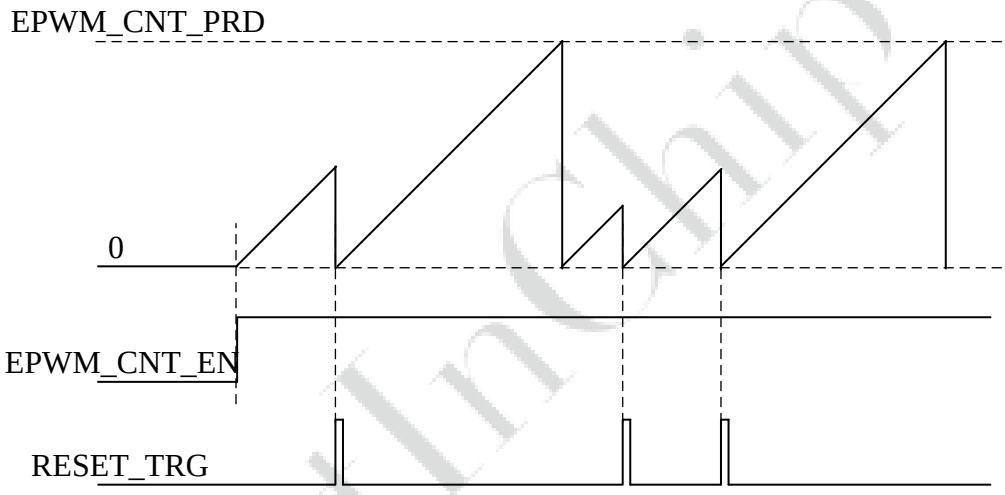
• 如果是递减计数的方式，从 EPWM_CNT_PRD 开始递减计数，计数至 0 后，计数器不会继续计数，计数器数值维持 0。

• 如果是先递增后递减计数的方式，从 0 开始递增计数，计数至 EPWM_CNT_PRD 后，递减计数到 0，然后不再计数，计数器数值维持 0。

9.1.4.1.1.7. 计数器计数复位

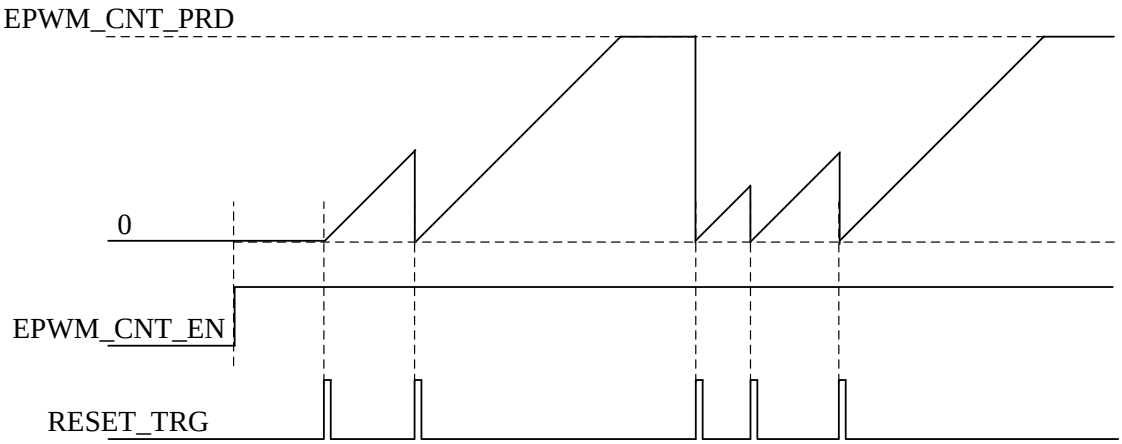
计数器在连续计数模式和单次计数模式下，均支持计数器的复位，复位信号为 EPWM_ETRG0/1 和软件复位信号。（其中 EPWM_ETRG0/ 1 信号的说明请查阅EPWM 外部触发信号）

连续计数模式下，计数器的复位情况如下：

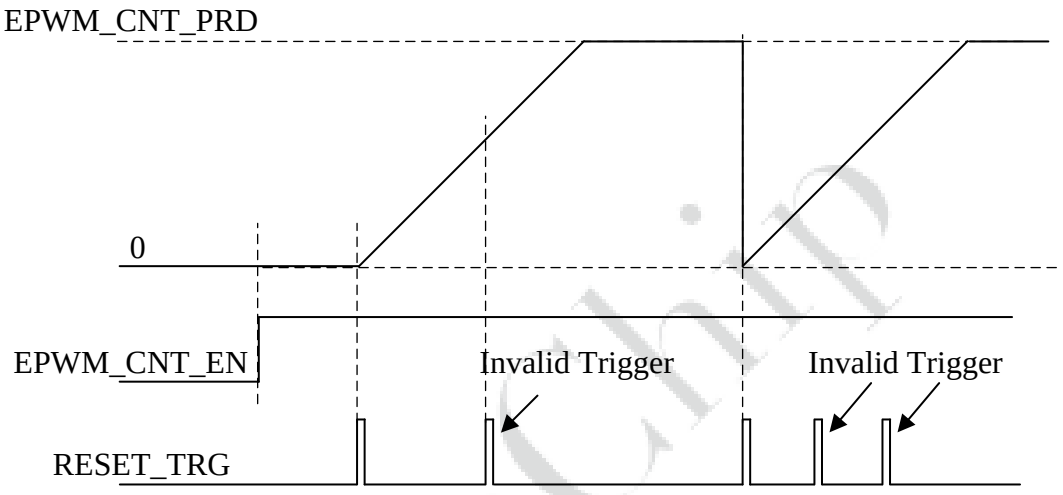


单次计数模式下，计数器的复位情况可通过寄存器选择以下触发类型：

- 可重复触发（寄存器 EPWM_CNT_RST_TRG = 0x1）：



• 不可重复触发（寄存器 EPWM_CNT_RST_TRG = 0x0）：



9.1.4.1.2. EPWM 生成模块

EPWM 生成模块，可以产生 A、B 两个通道的 EPWM 信号，分别为 EPWM _SA 和 EPWM_SB，具体需要配置以下信息：

• 下列信号产生的动作：

- EPWM_CNT_Z
- EPWM_CNT_PRD
- EPWM_CNT_UAV
- EPWM_CNT_DAV
- EPWM_CNT_UBV
- EPWM_CNT_DBV
- EPWM_ETRG0
- EPWM_ETRG1

• 寄存器中 EPWM 的输出模式。

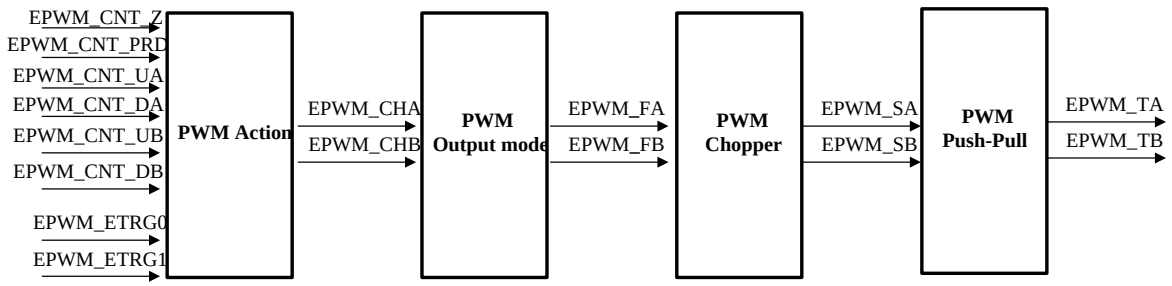


图 9–13 EPWM 生成模块的内部框图

如图 9–13 : EPWM 生成模块的内部框图所示，EPWM 生成模块的内部由以下模块组成：

- EPWM 动作控制模块：触发产生 EPWM 动作的信号，为计数器功能模块产生的信号。
- EPWM 输出模式：可以根据如下寄存器配置 EPWM 的输出。
  - EPWM_CH_SEL[1:0]
  - EPWM_POL_SET[1:0]
  - EPWM_BP_SET[1:0]
  - EPWM_SWAP_SET[1:0]
  - EPWM_DD_EN
- EPWM Chopper 模块
- 推挽输出模式

9.1.4.1.2.1. EPWM 动作控制模块

EPWM 动作控制模块，触发产生 EPWM 动作的信号为计数器功能模块产生的信号。所有的信号对 A、B 通道均有效。各信号描述如下：

- EPWM_CNT_Z，计数器数值达到 0 时产生。
- EPWM_CNT_PRD ，计数器数值达到 EPWM_CNT_PRDV 时产生。
- EPWM_CNT_UA，计数器递增计数达到 EPWM_CNT_AV 时产生。
- EPWM_CNT_DA，计数器递减计数达到 EPWM_CNT_AV 时产生。
- EPWM_CNT_UB，计数器递增计数达到 EPWM_CNT_BV 时产生。
- EPWM_CNT_DB，计数器递减计数达到 EPWM_CNT_BV 时产生。
- EPWM_ETRG0，外部触发信号 ETRG0 的有效信号。
- EPWM_ETRG1，外部触发信号 ETRG1 的有效信号。

当优先级高的信号所触发的动作为无动作时，控制器跳过该信号控制，由其它优先级的信号控制，如在 EPWM_CNT_UB/ DB 所触发的动作为无动作时，控制器则跳过 CMPB 的动作控制，进行 EPWM_CNT_UA/DA 的动作控制。各信号触发的动作优先级见动作控制信号优先级。

通过配置寄存器 EPWMA_ACT 和 EPWMB_ACT，可以在各信号上执行以下动作类型：

- 置 1：EPWM 输出信号设置为高电平。
- 置 0：EPWM 输出信号设置为低电平。
- 翻转：EPWM 输出信号进行翻转，若当前信号为高，动作执行将 EPWM 信号设置为低电平。
- 无操作：EPWM 输出信号无变化，维持现有的信号。

举例说明：假设 EPWM_CHA 动作满足以下条件，可以得出 图 9–14 : EPWM_CHA 信号生成示例：

- 计数器递增计数，EPWM_CNT_BV = 0x2，EPWM_CNT_PRD = 0x6
- EPWM_CHA 的动作设定：

- EPWMA_ACT_CNTZ = 0x2：置 1
- EPWMA_ACT_CNTUBV = 0x1：置 0
- 其它的动作设定则为无动作

图 9–14 : EPWM_CHA 信号生成示例中，生成的占空比则为 2/7 ≈ 28.5%

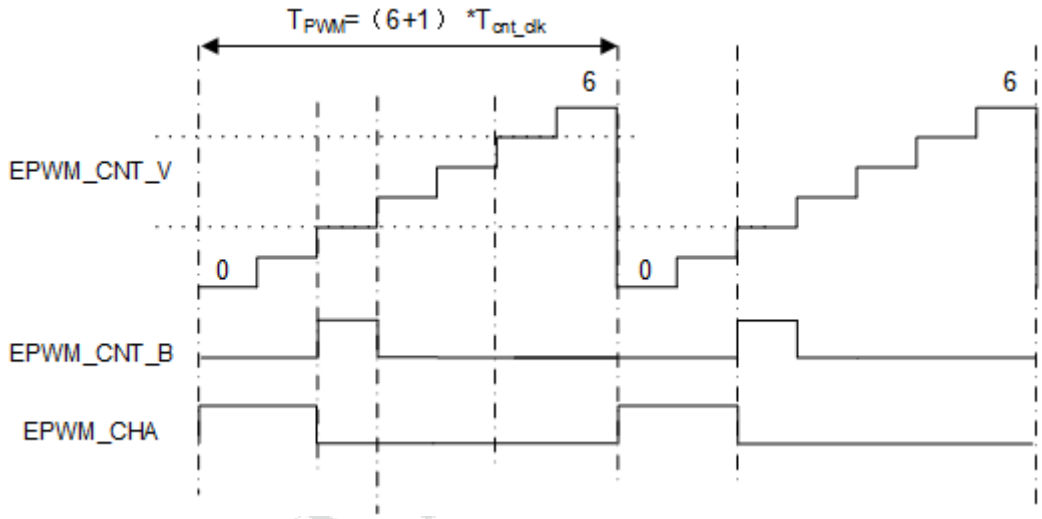


图 9–14 EPWM_CHA 信号生成示例

9.1.4.1.2.1.1. 软件强制控制信号

软件强制控制信号分为两种：

- 非持续的软件强制控制信号：信号非持续。若有下一个动作控制信号出现，则由下个控制信号控制 EPWM 的动作。
- 持续的软件强制控制信号：信号持续。即使下一个动作控制信号出现，由于软件强制控制信号优先级最高且信号持续，所以屏蔽后续的控制信号，仍然由持续的软件强制控制信号控制 EPWM 的动作。
  - 控制 EPWM 动作的类型包括置 1、置 0。
  - 如需由其它信号控制 EPWM 的动作，则关闭此功能。

9.1.4.1.2.1.2. 动作控制信号优先级

动作控制信号中，软件强制控制信号的优先级始终最高，其它信号的优先级在不同的计数模式下，排列会有所差异，动作控制信号优先级具体如下。当同时出现多个动作控制信号时，按照优先级最高的控制信号执行机制。

递增计数模式，优先级排列：

表 9–1 动作控制信号优先级

优先级	递增阶段
1	软件强制控制信号
2	EPWM_ETRG1

表 9–1 动作控制信号优先级 (续)

优先级	递增阶段
3	EPWM_ETRG0
4	EPWM_CNT_UB/ EPWM_CNT_DB
5	EPWM_CNT_UA/ EPWM_CNT_DA
6	EPWM_CNT_PRD
7	EPWM_CNT_Z

注：

EPWM_CNT_BV 对应产生的 EPWM_CNT_UB/ EPWM_CNT_DB 信号会影响 EPWM_CNT_Z 或 EPWM_CNT_PRD 上本应该发生的动作，因此软件需尽量避免 EPWM_CNT_BV = 0 或 PRD 值的情况 。

9.1.4.1.2.2. EPWM 输出模式

通过以下寄存器，可配置所需的 EPWM 输出：

- EPWM_CH_SEL[1:0]，通道选择控制信号，分别选择上升沿信号处理通道的输入源，以及下降沿信号处理通道的输入源。两路的信号处理通道都可以选择 CHA 或 CHB。

  注：

  通过对上升沿信号以及下降沿信号可以实现死区控制，具体配置可查看 EPWM 死区控制。

- EPWM_DD_EN，双边沿死区使能。
- EPWM_POL_SET[1:0]，极性选择控制信号，选择正极性或负极性的信号。
- EPWM_BP_SET[1:0]，旁通选择，EPWM_BP_SET[0]可以选择旁通 EPWM_CHA，EPWM_BP_SET[1] 可以选择旁通 EPWM_CHB。
- EPWM_SWAP_SEL[1:0]，互换选择，选择输出通路

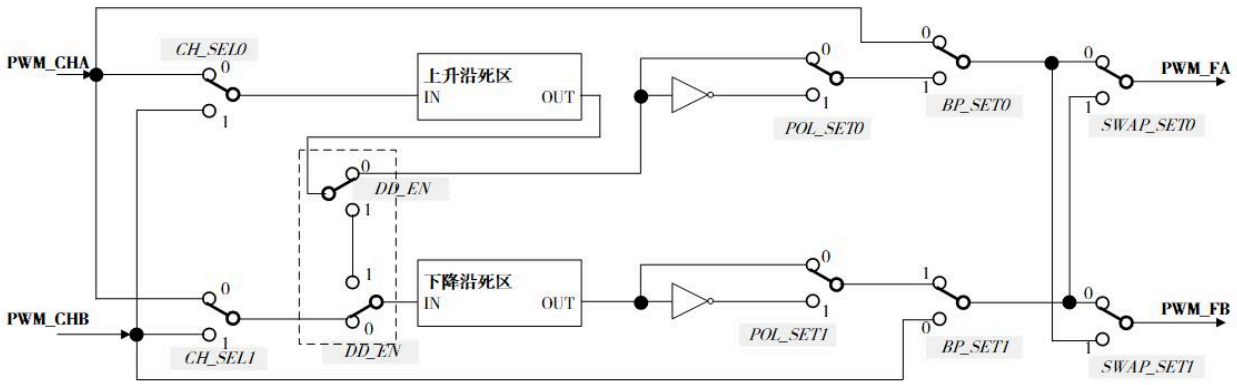


图 9–15 EPWM 输出模式的控制图示：

9.1.4.1.2.2.1. EPWM 死区控制

EPWM 死区控制，分别通过对上升/ 下降边沿的时间进行延迟，通过 EPWM_RE_DZCTL 和 EPWM_FE_DZCTL 寄存器的配置下列延迟时间参数（单位：EPWM_CNT_CLK)：

- EPWM_RE_DZCTL
- EPWM_FE_DZCTL

图 9–16 : EPWM_CHA 插入上升沿死区以及下降沿死区中，以通道 EPWM_CHA 的信号为例，分别设置 EPWM_RE_DZCTL = EPWM_FE_DZCTL = 1，得到如下的信号，实现上升沿/ 下降沿的死区时间控制为 1 * Tcnt_clk。

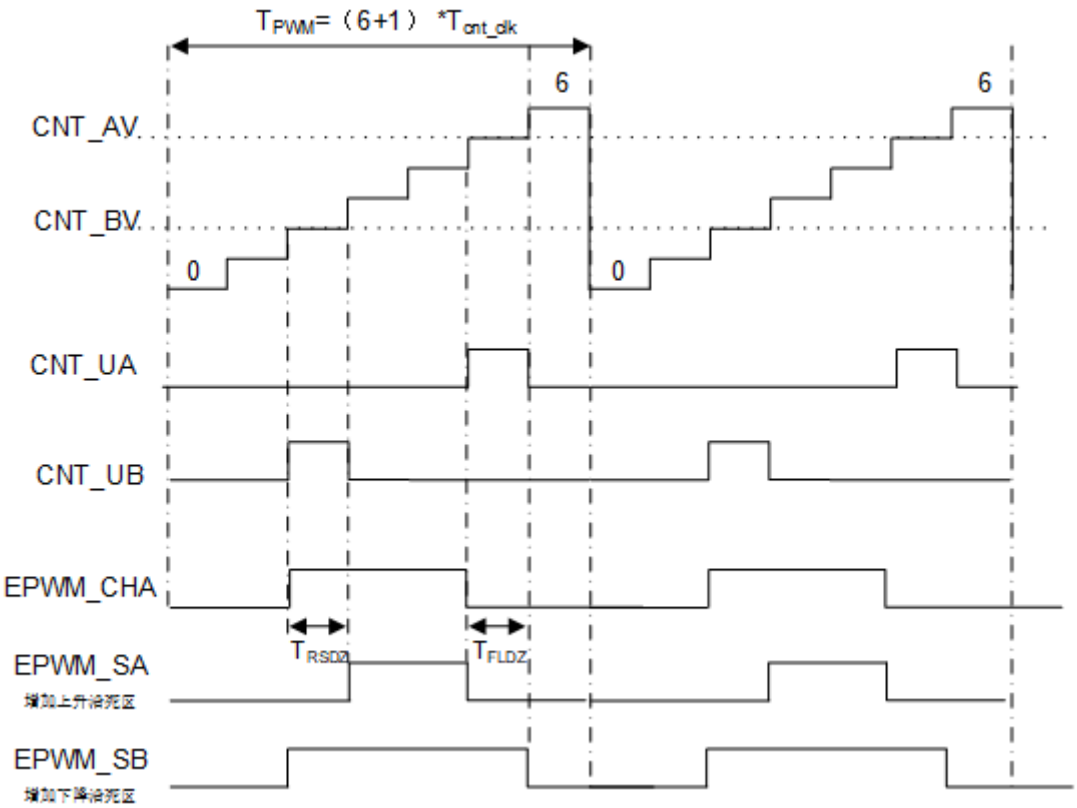


图 9–16 EPWM_CHA 插入上升沿死区以及下降沿死区

9.1.4.1.2.3. EPWM Chopper 模块

EPWM Chopper 功能如下所示：

- 寄存器 EPWM_CHOP_EN = 0x0：EPWM Chopper 模块旁通
- 寄存器 EPWM_CHOP_EN = 0x1：EPWM Chopper 模块工作

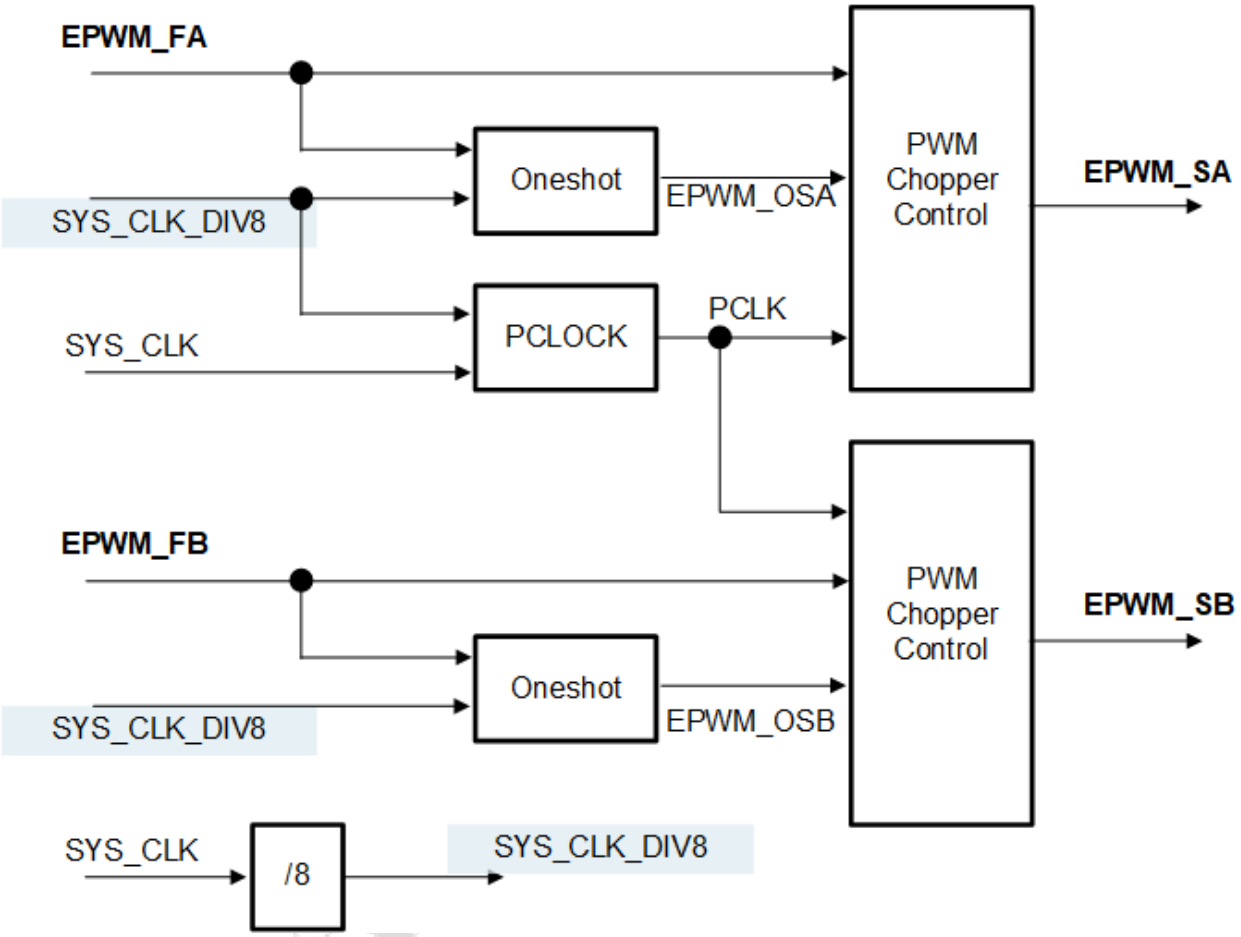


图 9–17 EPWM Chopper 功能框图

如 图 9–17 : EPWM Chopper 功能框图所示：

1. Oneshot 信号以 EPWM_FA/FB 信号的上升边沿为触发边沿，产生脉宽为 EPWM _CHOP_OS_WTH * PSCLK 的 Oneshot 信号。

   图 9–20 : PCLK 以及 Oneshot 调制图示 中，EPWM _CHOP_OS_WTH = 0x1，产生了 EPWM_OSA 信号，利用 EPWM_OSA 以及 PCLK 信号进行调制，得出 EPWM_SA。

2. PCLOCK 对 EPWM_FA 和 EPWM_FB 信号进行调制，生成 PCLK 信号。

   PClock 模块控制产生的 PCLK 信号的频率以及占空比。

   - 寄存器 EPWM_CHOP_FRE：配置分频系数
   - 寄存器 EPWM_CHOP_DUTY：调节和配置占空比，调节范围是 12.5%~87.5%，每档位 12.5%。

   图 9–18 : PCLK 调制 EPWM 信号中以互补的死区控制的一对信号，进行了 PCLK 信号的调制。

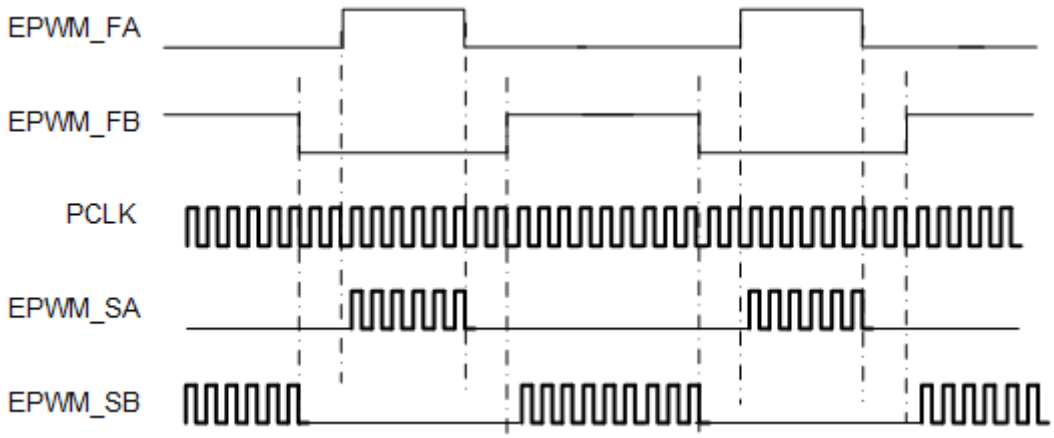


图 9–18 PCLK 调制 EPWM 信号

PCLK 的占空比调整则如图 9–19 : PCLK 的占空比所示。

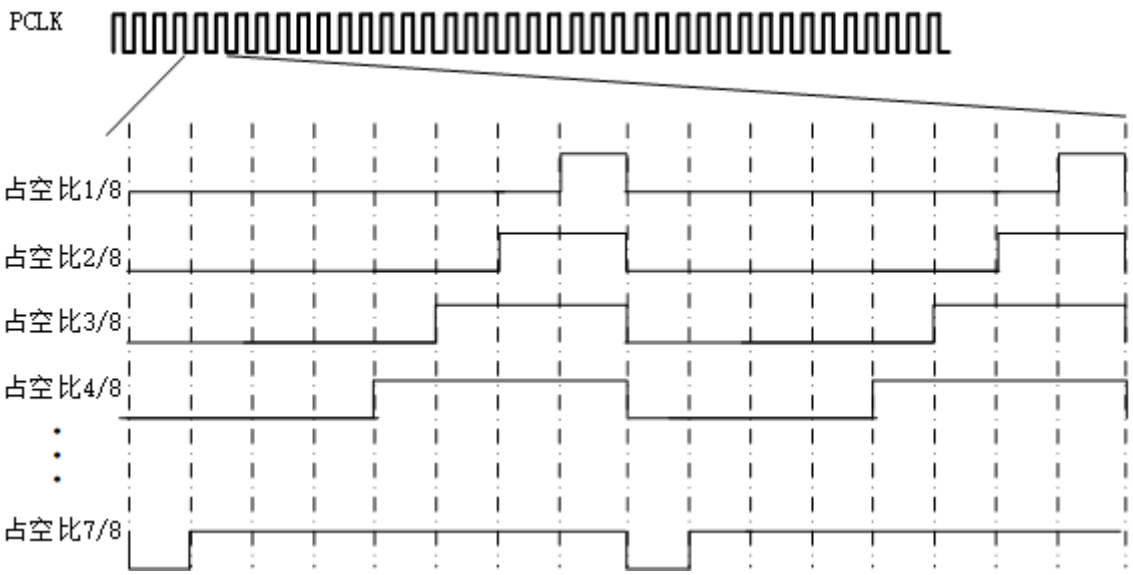


图 9–19 PCLK 的占空比

3. EPWM Chopper Control 利用 EPWM_OSA/ OSB 和 PCLK 信号输出调制信号 EPWM_SA 和 EPWM_SB。

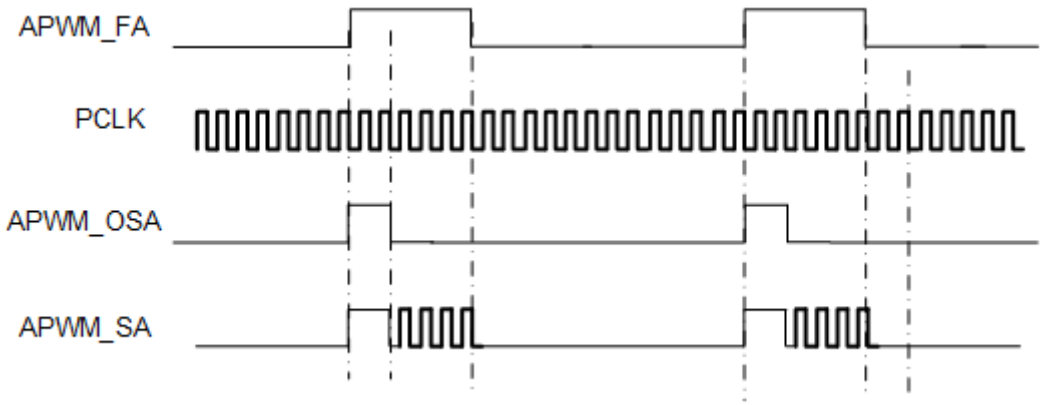


图 9–20 PCLK 以及 Oneshot 调制图示

9.1.4.1.2.4. 推挽输出模式

EPWM 的推挽输出模块功能如下：

- 寄存器 PSHPLL_EN = 0x0 时，EPWM 的推挽输出模块旁通。
- 寄存器 PSHPLL_EN = 0x1 时，EPWM 的推挽输出模块工作。

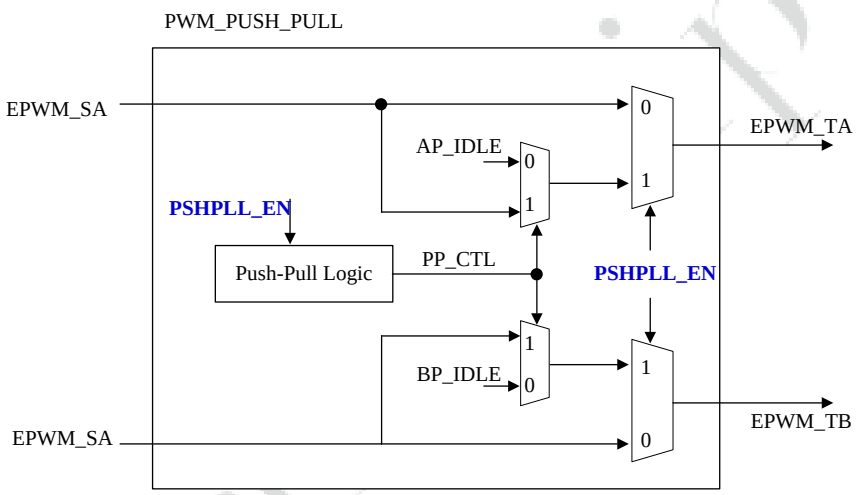
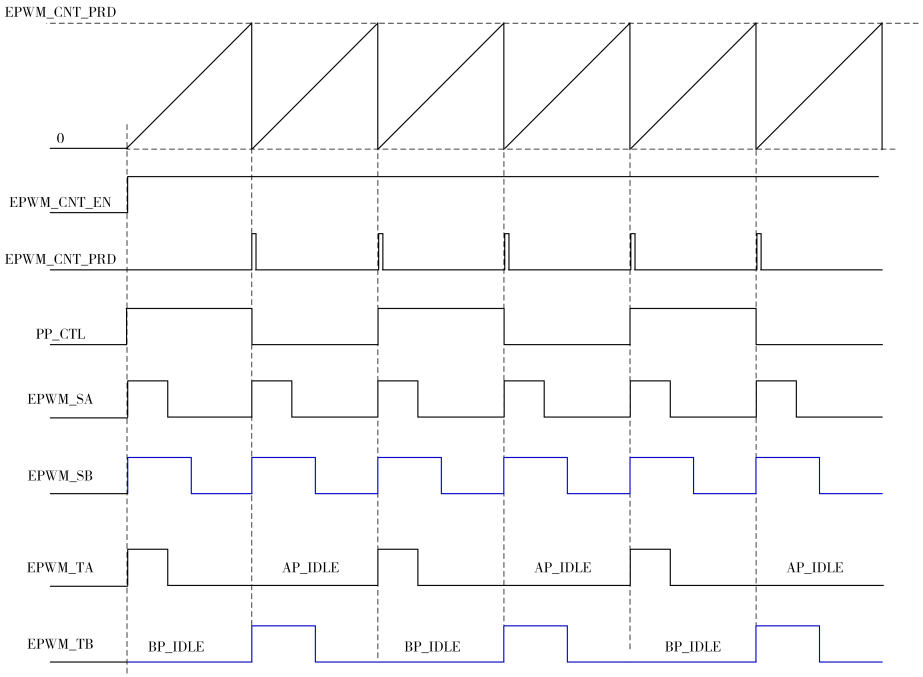


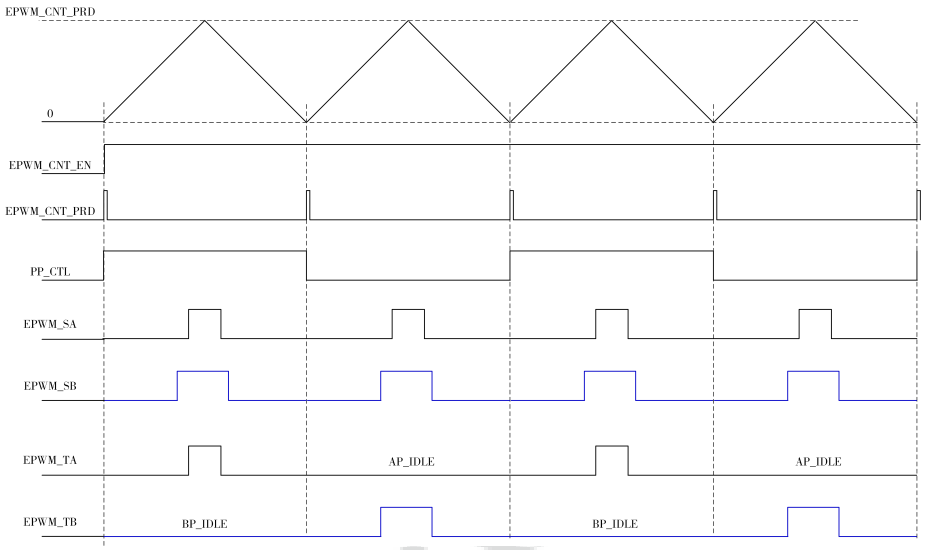
图 9–21 推挽输出模块框图

推挽输出模块的工作逻辑如下：

- Case1：计数器采用递增计数，PSHPLL_EN = 0x1，PP_CTL_INV_EN = 0x0



• Case2：计数器采用先递增后递减计数，PSHPLL_EN = 0x1，PP_CTL_INV_EN = 0x0



9.1.4.1.3. EPWM 保护模块

EPWM 保护模块提供了以下几大功能：

- EPWM 初始值配置
- 故障模式
- 故障指示输入
- 故障操作
- 故障中断

9.1.4.1.3.1. EPWM 初始值配置

通过寄存器可以配置 EPWM 的输出信号EPWM0/1/.../11_A 以及 EPWM0/1/.../11_B 在 EPWM 功能未使能的情况下的初始值，表示 EPWM 功能未使能时的初始状态。

以 EPWM0_A 信号为例，在 EPWM0 模块未使能的情况下，可以有以下的情况：

- 若 EPWM_A_INIT = 0x0，EPWM 模块未使能，EPWM0_A 输出低电平。
- 若 EPWM_A_INIT = 0x1，EPWM 模块未使能，EPWM0_A 输出高电平。
- 若 EPWM_A_INIT = 0x2/3，EPWM 模块未使能，EPWM0_A 输出为高阻态。

计数器停止计数时，通过配置以下寄存器，选择初始值控制：

- PAUSE_INI_SRC_EN，使能初始值控制全通路。

  使能时，EPWM 模块可以在计数器停止运行时，根据配置的初始值控制逻辑来设置输出信号的状态。

  - 0x0：停止计数时，初始值只控制输出。
  - 0x1：停止计数时，初始值控制全通路。

- PAUSE_INI_OUT_EN，停止计数是否插入初始值。

  该寄存器用于控制当计数器停止运行时，是否将配置的初始值插入到 EPWM 输出信号中。

- 0x0：未使能。
- 0x1：使能，初始值控制最后的输出。

表 9–2 计数器停止计数时初始值控制逻辑配置说明

场景	PAUSE_INI_SRC_EN	PAUSE_INI_OUT_EN	初始值控制逻辑
配置 1	0x0	0x0	EPWM_A_INIT 和 EPWM_B_INIT 寄存器配置值不生效。
配置 2	0x0	0x1	计数器停止计数时，EPWM_A_INIT 和 EPWM_B_INIT 控制输出。
配置 3	0x1	0x0	计数器停止计数时，PWM 输出取决于初始值以及输出模式配置。
配置 4	0x1	0x1	计数器停止计数时，PWM 输出取决于初始值配置，但当计数器恢复正常计数时，PWM 输出取决于初始值以及输出模式配置，直到下一次动作发生。

9.1.4.1.3.2. 故障模式

EPWM 保护模块可以同时支持两种类型的故障模式：

- 可恢复故障模式：
  - 当 EPWM_RFLT_ALL 有效，EPWM 模块立即执行故障操作，如果有使能中断模式，则产生 EPWM_RC_FALT_INT 中断。
  - 当计数器计数复位至 0，若 EPWM_RFLT_ALL 信号变回无效状态，EPWM 的故障操作解除，EPWM 可以继续正常工作。
- 不可恢复故障模式：
  - 当 EPWM_NRFLT_ALL 有效，EPWM 模块立即执行故障操作，如果有使能中断模式，则产生 EPWM_NRC_FALT_INT 中断。
  - 当出现不可恢复的故障时，EPWM 一直处于故障操作的状态，不能修改。

    不可恢复故障的消除，只能通过软件写寄存器消除此状态，才可以使得 EPWM 再次正常工作。

9.1.4.1.3.3. 故障指示输入

故障指示的输入源来自引脚 GPIO、内部比较模块 CPM 和内部 SDFM 模块ADC 看门狗输出信号。这类输入信号经过 OR 逻辑可以得到 EPWM_RFLT_ALL 和 EPWM_NRFLT_ALL 信号，分别触发进入可恢复故障模式和不可恢复故障模式。

> **注：**
>
> CPM 的故障输入以及 SDFM 的故障输入信号，固定为高电平有效，不可配置。

故障指示输入包括：

• 可恢复故障指示输入

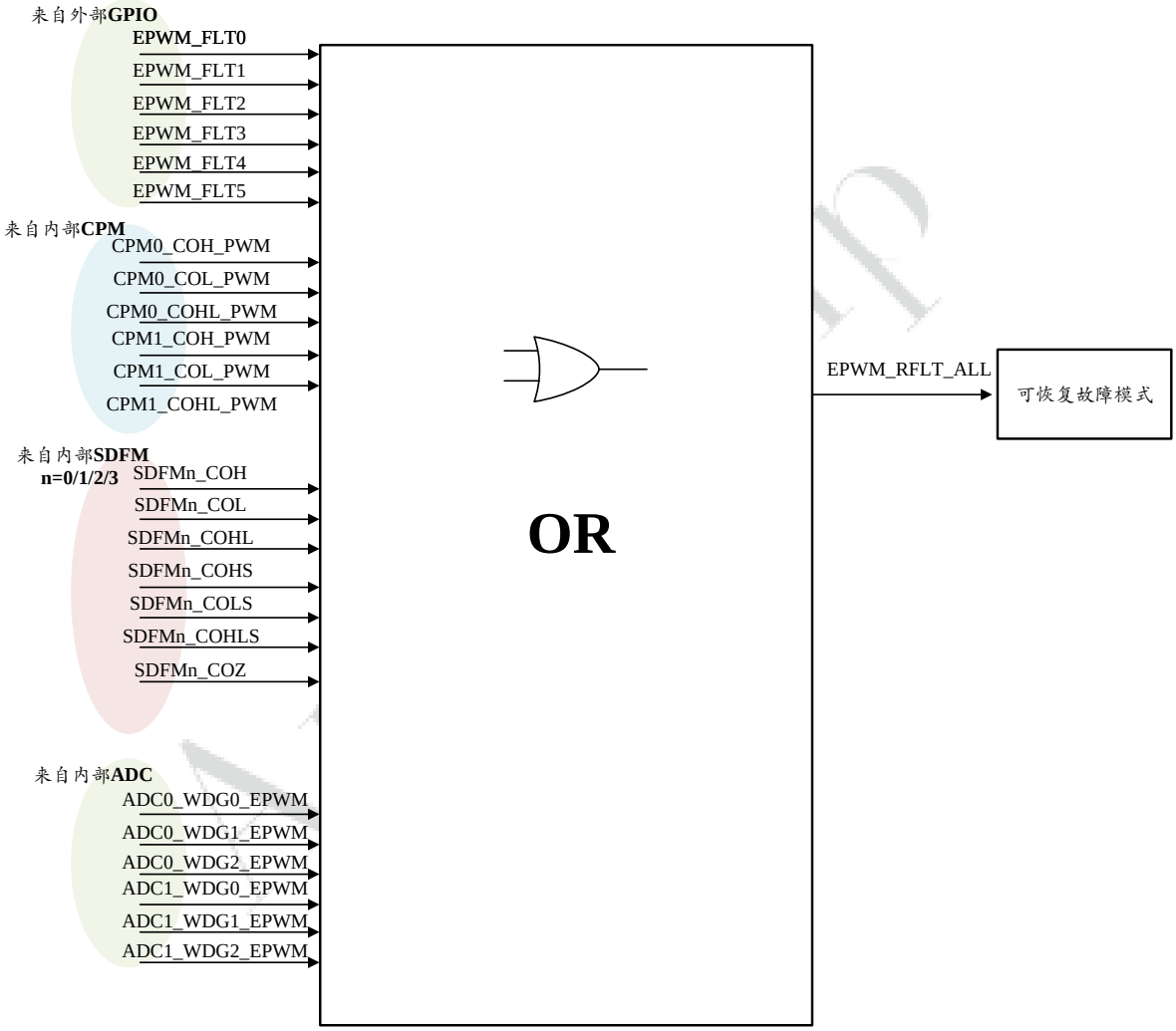


图 9–22 EPWM 可恢复故障指示输入

• 不可恢复故障指示输入：

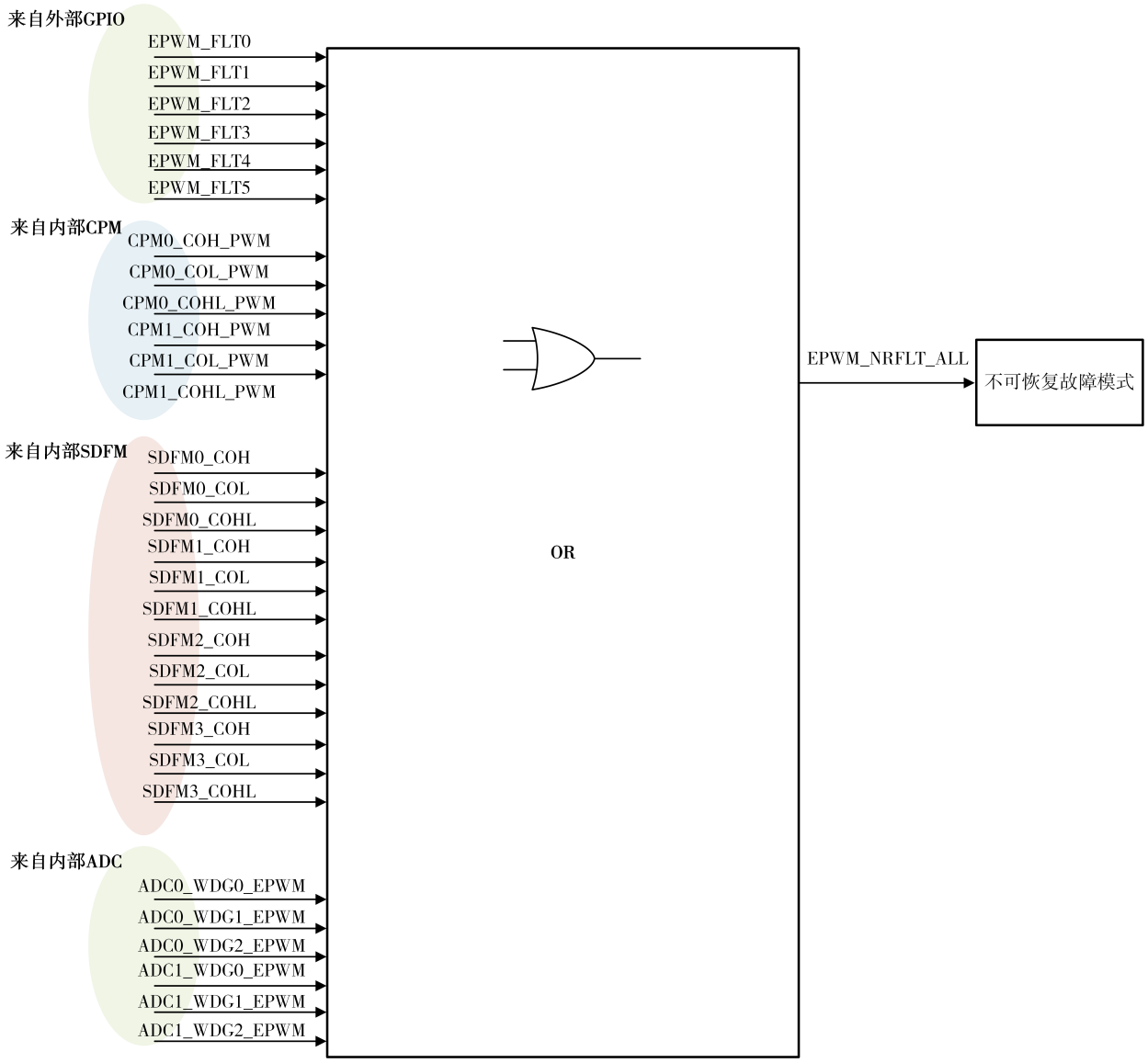


图 9–23 EPWM 不可恢复故障指示输入

9.1.4.1.3.4. 故障操作

当故障指示输入有效时，EPWM 强制执行故障操作，EPWM 输出信号的故障操作包括：

> **注：**
>
> 通过配置寄存器，软件可以强制 EPWM 输出执行故障操作。

• EPWM 强制输出高电平

• EPWM 强制输出低电平

• EPWM 强制输出高阻态

• 无操作

9.1.4.1.3.5. 故障延迟操作

EPWMA 和 EPWMB 的输出保护均支持保护延迟的操作，通过寄存器 EPWMA/B_FLT_PRTCT_DLY 控制，可以独立配置延迟 1/2/3/.../15 us。

9.1.4.1.3.6. 故障中断

故障中断，由可恢复故障中断 EPWM_RC_FALT_INT 与不可恢复中断 EPWM _NRC_FALT_INT 进行或逻辑所得 EPWM_FALT_INT 信号。

9.1.4.1.4. ADC 采样触发和中断模块

9.1.4.1.4.1. ADC 采样触发

EPWM 模块可以通过 ADC 采样触发的功能，产生 ADC 采样使能信号，直接触发 ADC 进行模拟信号的采集。具体流程如下：

1. 每个 EPWM 模块可以最多产生 2x ADC 采样触发信号，即输出 EPWM n_ADC_CVRA 和 EPWMn_ADC_CVRB。
2. EPWMn_ADC_CVRA 和 EPWMn_ADC_CVRB 分别经过 OR 逻辑后，对应生成 EPWM_ADC_CVRA 和 EPWM_ADC_CVRB，送到 ADC0/1，再分别作为 ADC0/1 的队列 1 和队列 2 的触发信号。

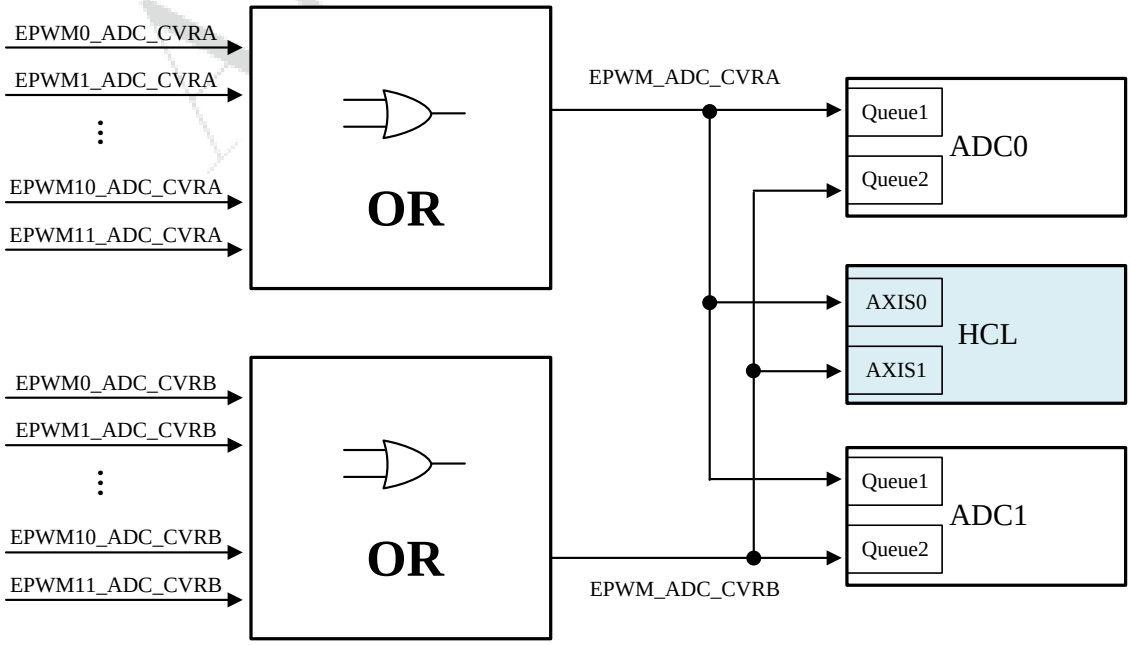


图 9–24 ADC 采样触发图示

3. 产生采样触发信号的输入源与控制 EPWM 动作的信号一致，可以选择以下信号作为触发信号：

> **注：**
>
> CPM 所有输出的信号的上升沿对应是触发的时刻。

- EPWM_CNT_Z
- EPWM_CNT_PRD
- EPWM_CNT_UA
- EPWM_CNT_DA
- EPWM_CNT_UB
- EPWM_CNT_DB
- CPM0_COH
- CPM0_COL
- CPM0_COHL
- CPM1_COH
- CPM1_COL
- CPM1_COHL
- EPWM_CNT_UC
- EPWM_CNT_DC
- EPWM_CNT_UD
- EPWM_CNT_DD
- ETRG0
- ETRG1

4. 通过寄存器字段 EPWM_ADC_CVRA/ B_DIV 可以配置 ADC 采样触发信号产生的频次。

   频次可配置为每 1/2/.../15 次产生触发信号。以 CNT_Z 信号为例，若配置频次为每三次，当出现三次 CNT_Z 有效信号，会产生一次 ADC 采样信号。

9.1.4.1.4.2. EPWM 中断

每个 EPWM 模块可以产生 1x EPWM 中断信号。此中断信号的产生与 EPWM 动作的信号一致。可以选择以下信号作为中断信号 EPWM_INT：

- EPWM_CNT_Z
- EPWM_CNT_PRD
- EPWM_CNT_UA
- EPWM_CNT_DA
- EPWM_CNT_UB
- EPWM_CNT_DB
- PUL_CNT_ZRO

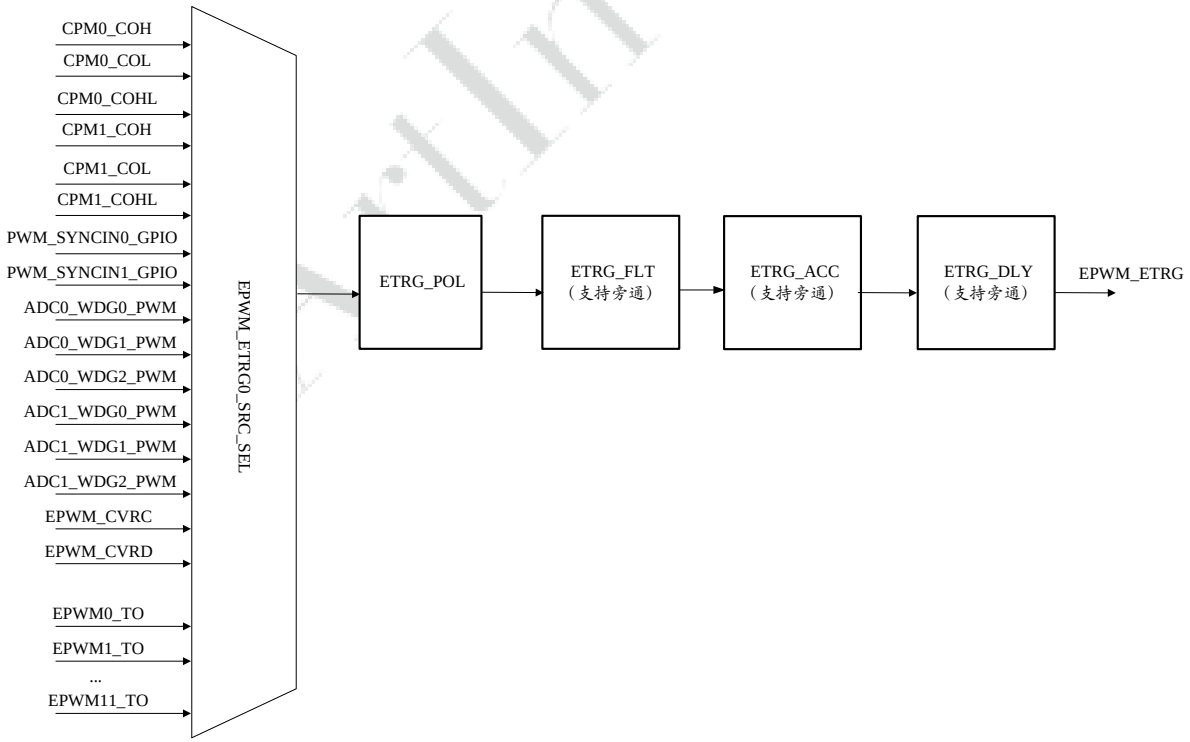
- EPWM_CNT_UC
- EPWM_CNT_DC
- EPWM_CNT_UD
- EPWM_CNT_DD

通过寄存器字段 EPWM_INT_DIV 可以配置 中断信号产生的频次。

频次可配置为每 1/2/.../15 次产生触发信号。以 CNT_Z 信号为例，若配置频次为每三次，当出现三次的 CNT_Z 有效信号，会产生一次 EPWM 中断信号。

9.1.4.1.5. EPWM 外部触发信号

EPWM 外部触发信号 EPWM_ETRG0/1 是按照下面的处理流程框图生成。触发信号 EPWM_ETRG0/1 可以用于对 EPWM 的计数器、对 EPWM 的输出动作、对 ADC 的采样触发信号以及对 EPWM 的同步信号均可以进行控制和修改。



9.1.4.1.5.1. ETRG0/ 1 信号源

EPWM 外部触发信号 EPWM_ETRG0/1 可以通过寄存器 EPWM_ETRG0/1_SRC_SEL 进行配置选择输入信号源，支持的输入信号源如下：

- 来自比较器的信号，CPM0_COH/ CPM0_COL/ CPM0_COHL/ CPM1_COH/ CPM1_COL/ CPM1_COHL
- 来自外部的引脚信号，PWM_SYNCIN0/1
- 来自 EPWM 的触发信号，EPWM_CVRC/D 和 EPWMn_TO

> **注：**
>
> EPWM 生成的输出信号 EPWM_CVRC/D 也可以作为外部触发信号。

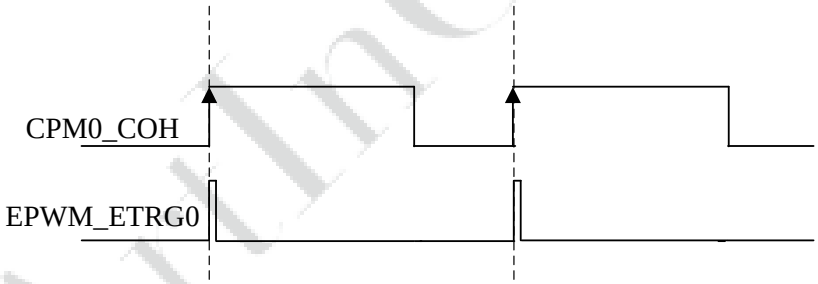
- 来自 ADC 的看门狗信号，ADC0_WDG0/1/2_PWM 和 ADC1_WDG0/1/2_PWM

9.1.4.1.5.2. ETRG0/ 1 信号极性

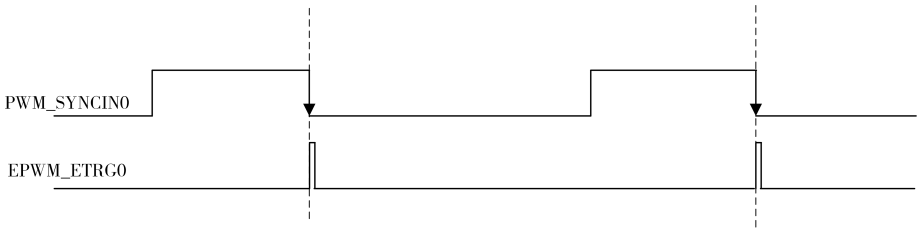
EPWM_ETRG0/1 选择输入信号源后，需要配置 EPWM_ETRG0_SRC_POL 进行源信号的极性设置。

假设选择了 PWM_SYNCIN0 作为 EPWM_ETRG0 的输入信号源：

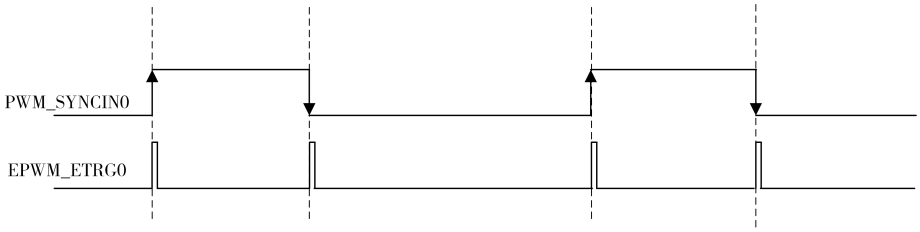
- 配置 EPWM_ETRG0_SRC_POL = 0x0 选择上升沿触发，PWM_SYNCIN0 信号的上升沿将会对应产生一次 EPWM_ETRG0 信号。



- 配置 EPWM_ETRG0_SRC_POL = 0x1 选择下降沿触发，PWM_SYNCIN0 信号的下降沿将会对应产生一次 EPWM_ETRG0 信号，如下图所示。



- 配置 EPWM_ETRG0_SRC_POL = 0x2/3 选择上下沿触发，对应的时序如下：



9.1.4.1.5.3. ETRG0/1 信号过滤

EPWM_ETRG0/1 信号支持内部过滤，从而限制 ETRG0/1 仅在特定的时间内生效。EPWM_ETRG0/1 信号过滤的方式有两种：

- 消隐模式，在定义时间段内屏蔽外部事件。
- 加窗模式，仅在定义的时间段内外部事件有效。

注：

关于两种信号过滤模式的配置示例，可分别查看 加窗模式配置和消隐模式配置。

定义时间段的方式：

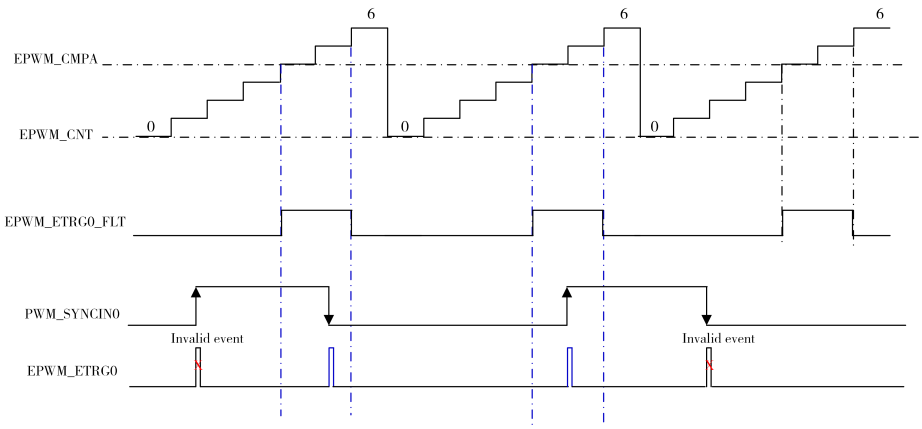
1. 设置时间的起点，时间的起点信号主要从下列信号中选择：
   - EPWM_CNT_Z
   - EPWM_CNT_PRD
   - EPWM_CNT_UA
   - EPWM_CNT_DA
   - EPWM_CNT_UB
   - EPWM_CNT_DB
   - EPWM_CNT_UC
   - EPWM_CNT_DC
   - EPWM_CNT_UD
   - EPWM_CNT_DD
2. 设置时间的长度，时间的长度通过寄存器 EPWM_ETRG0_FLT_LEN 配置，时间单位是 1x 计数器时钟。

加窗模式配置

加窗模式配置示例如下：

1. 选择 PWM_SYNCIN0 作为 EPWM_ETRG0 的输入信号源。
2. 配置 EPWM_ETRG0_SRC_POL = 0x2/3 选择上下沿触发，PWM_SYNCIN0 的信号的上下沿均会对应的产生一次 EPWM_ETRG0 信号。
3. 配置 EPWM_ETRG0_FLT_MOD = 0x3，选择加窗模式，即 ETRG0 仅在定义时间段内有效。
4. 配置 EPWM_ETRG0_FLT_ST_SET = 0x2，选择 EPWM_CNT_UA 作为滤波起点时刻。
5. 配置 EPWM_ETRG0_FLT_LEN = 0x2，定义从起点时刻持续 2x 计数器时钟的时间长度作为滤波时间。

实际效果图如下：

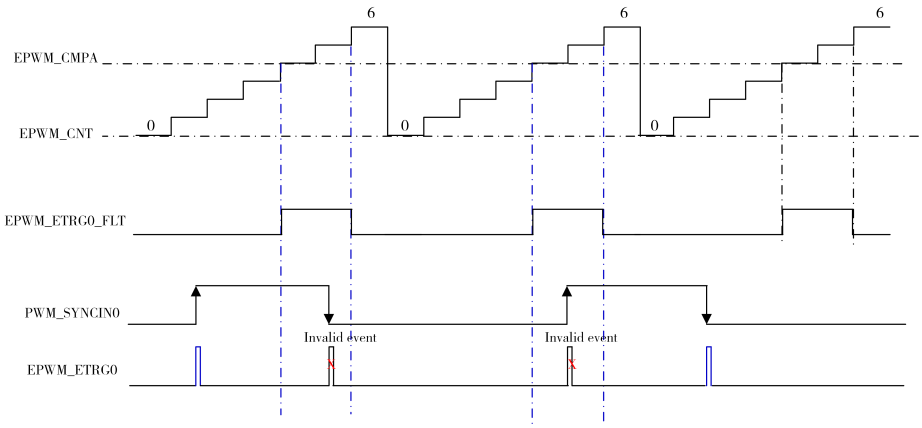


消隐模式配置

消隐模式配置示例如下：

1. 选择 PWM_SYNCIN0 作为 EPWM_ETRG0 的输入信号源。
2. 配置 EPWM_ETRG0_SRC_POL = 0x2/3 选择上下沿触发，则 PWM_SYNCIN0 信号的上下沿均会对应的产生一次 EPWM_ETRG0 信号。
3. 配置 EPWM_ETRG0_FLT_MOD = 0x2，选择消隐模式，即在定义时间段内 ETRG0 无效。
4. 配置 EPWM_ETRG0_FLT_ST_SET = 0x2，选择 EPWM_CNT_UA 作为滤波起点时刻。
5. 配置 EPWM_ETRG0_FLT_LEN = 0x2，定义从起点时刻持续 2x 计数器时钟的时间长度作为滤波时间。

实际效果如下图：



9.1.4.1.5.4. ETRG0/1 信号累积功能

EPWM_ETRG0/1 支持累积功能，即仅在 ETRG0/1 信号发生的事件数达到设定的阈值 EPWM_ETRG0_ACC_TH 后，才可以输出 ETRG0/1 信号。

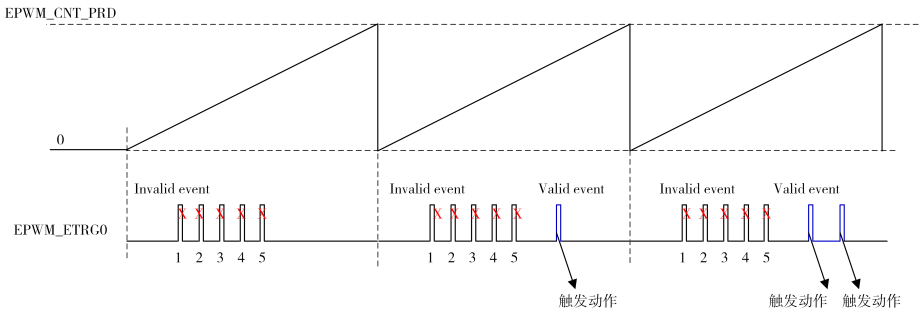
累积功能支持以下累积模式：

> **注：**
>
> EPWM_ETRG0_ACC_TH 采用影子更新，在每个 EPWM 周期完成后进行更新。

- 单周期的累积模式，EPWM_ETRG1_ACC_MOD = 0x0

  单周期模式下，每个 EPWM 周期内进行累积，在当前周期内 EPWM_TRG0/1 产生的有效事件数大于阈值后，EPWM_TRG0/1 才有有效，而在下一个新的 EPWM 周期，累积会清 0，需要重新累积超过阈值后，EPWM_TRG0/1 才有效。

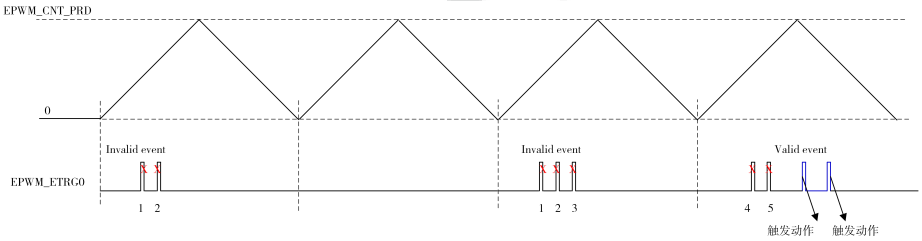
  下图为 EPWM_ETRG0 信号的单周期的累积模式，设置阈值 EPWM_ETRG0_ACC_TH = 5，累积功能的时序如下：



- 多周期的累积模式，EPWM_ETRG1_ACC_MOD = 0x1

  多周期模式下，EPWM_TRG0/1 事件会一直累积，只有在一个 EPWM 周期没有出现一次 EPWM_TRG0/1 时，会对累积进行清除。

  下图为 EPWM_ETRG0 信号的多周期的累积模式，设置阈值 EPWM_ETRG0_ACC_TH = 5，累积功能的时序如下：



9.1.4.1.6. 输入信号过滤

EPWM 保护模块的输入信号 EPWM_FLT0/1/2/3/4/5_IN，是经过输入信号过滤模块过滤的信号。输入信号过滤模块，用于滤除噪声信号，每路的输入信号都配置有对应的输入滤波模块：

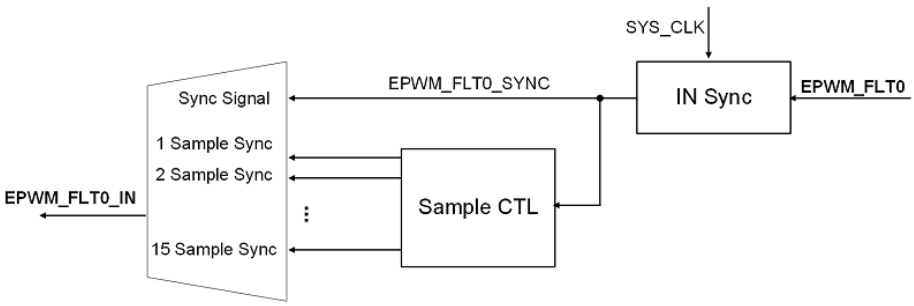


图 9–25 输入滤波模块功能框图

图 9–25 : 输入滤波模块功能框图中以 EPWM_FLT0 信号为例，输入滤波的功能可设置成以下模式：

- 同步信号输出：通过 SYS_CLK 对输入信号 EPWM_FLT0 进行同步。
- 1/2/…/15 采样点的同步信号输出：同步信号经过 Sample CTL 模块，判定 1/2/…/15 个采样点为相同值的信号认为有效信号，否则认为噪声信号进行滤除。

- Sample CTL 模块的输入信号是 IN Sync 模块的输出信号，即 SYS_SCLK 对 EPWM_FLT0 信号进行同步得出 EPWM_FLT0_SYNC 信号。
- 通过配置 Sample CTL 模块的寄存器，可以配置对 EPWM_FLT0_SYNC 信号的采样周期，采样周期为 1x/ 2x/ 4x/ 6x/ …/ 510x SYS_CLK。
- 通过寄存器可以配置 1/2/…/15 采样点的同步信号输出，即以 1/2/…/15 个采样点的时间长度作为一个处理周期。若这个周期的采样点均为相同值则认为是有效值，EPWM_FLT0_IN 的信号输出此有效值。

图 9–26 : 五采样点同步信号输出图示中，以采样周期配置为 2x SYS_CLK、5 采样点同步信号输出的配置为例，对 EPWM_FLT0_SYNC 信号进行处理。

1. 在 T0 时刻，对 EPWM_FLT0_SYNC 进行第一次信号采集，采集为 0 信号。
2. 接下来采集的四次信号均为 0 信号，因此在第五个采样点的时候，EPWM_FLT0_IN 进行了翻转，输出 0 信号，完成一个处理周期。
3. 在 T1 时刻，由于第一次采集信号为 0 信号，而第二次采集信号出现 1 信号，此次处理周期判定为无效值，EPWM_FLT0_IN 维持当前值。

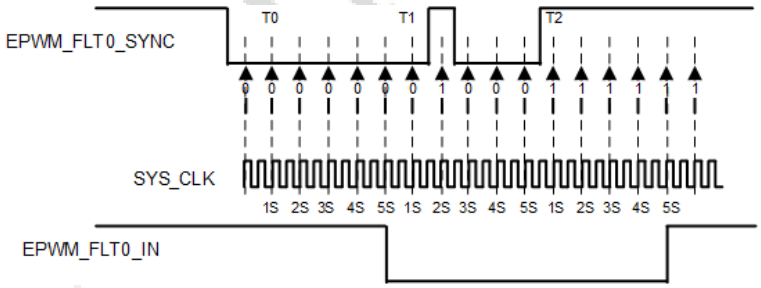


图 9–26 五采样点同步信号输出图示

9.1.4.1.7. 触发信号输出功能

EPWM 支持通过下列寄存器选择计数器在不同时刻的触发信号，通过外部 GPIO 输出：

- EPWM_CNT_Z
- EPWM_CNT_PRD
- EPWM_CNT_UAV
- EPWM_CNT_DAV
- EPWM_CNT_UBV
- EPWM_CNT_DBV
- EPWM_CNT_UCV
- EPWM_CNT_DCV
- EPWM_CNT_UDV
- EPWM_CNT_DDV

输出的触发信号，进行 8x SYS_CLK 的脉冲延长。

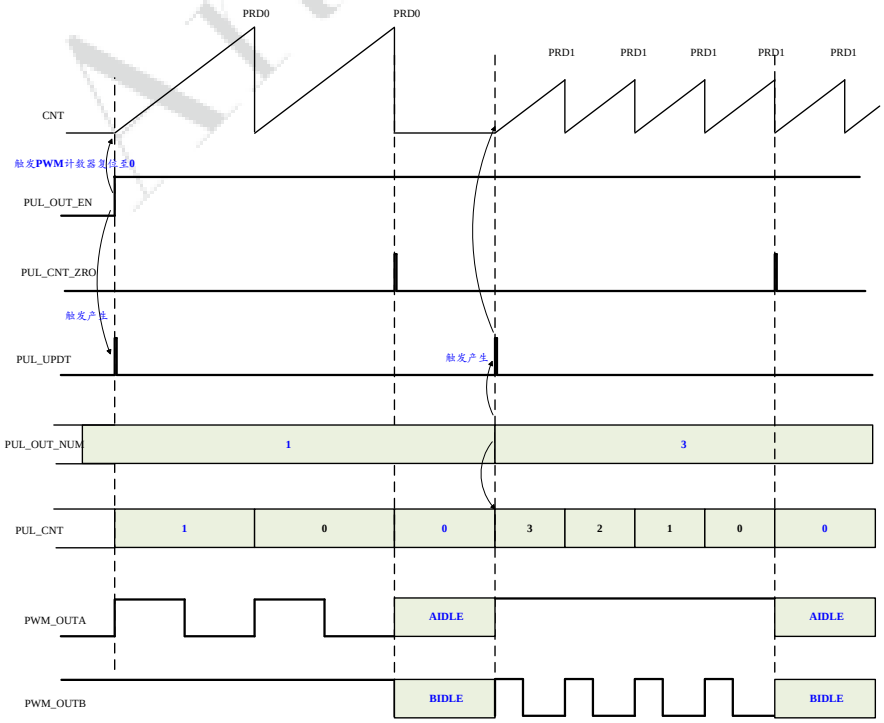
9.1.4.1.8. 脉冲输出功能

脉冲输出功能，为发送特定数量的脉冲信号的功能。

通过寄存器 PUL_OUT_EN= 0x1，进入脉冲输出模式，设置 PUL_OUT_NUM，可以配置发送脉冲的数量，发送的脉冲数量为 PUL_OUT_NUM+1。

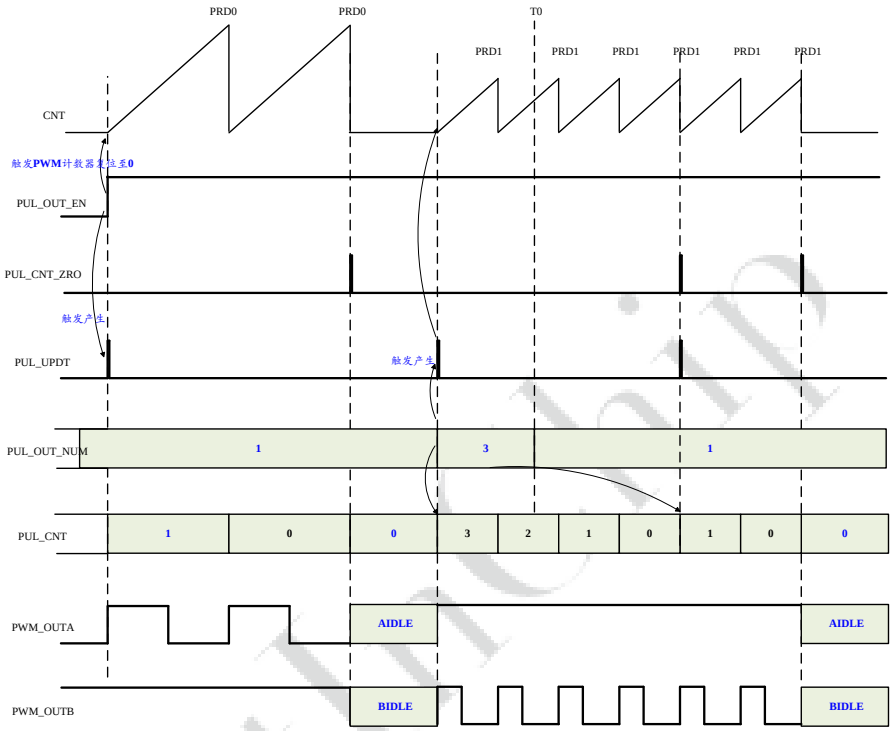
基本的脉冲输出功能的时序图如下：

1. PUL_OUT_EN = 0x1 的时刻，触发一次的脉冲输出，此时 EPWM 计数器需复位到 0，并且 PUL_OUT_NUM 的数值会传到 PUL_CNT。
2. 计数器每完成一次周期的计数后，会对 PUL_CNT 进行递减 1，在这里过程中，EPWM 的输出信号为正常输出的状态。
3. 在计数器完成了所设置的执行的周期数后，则进入了 IDLE 状态，EPWMA 通路为 AIDLE，EPWMB 通路为 BIDLE，A/BIDLE 的输出电平直接采用 AB 通道的初始值控制。
4. 如果在 IDLE 状态，对 PUL_OUT_NUM 寄存器进行一次写的操作，则触发新一轮的脉冲群的发送。相应的此时会复位计数器的计数到 0，重新开始新的一轮计数。如下图的对寄存器 PUL_OUT_NUM 写入数值 3。
5. 在完成上一步的操作，也即是完成 4 个脉冲的输出后，再次进入 IDEL 状态，EPWM 输出 IDLE 电平。

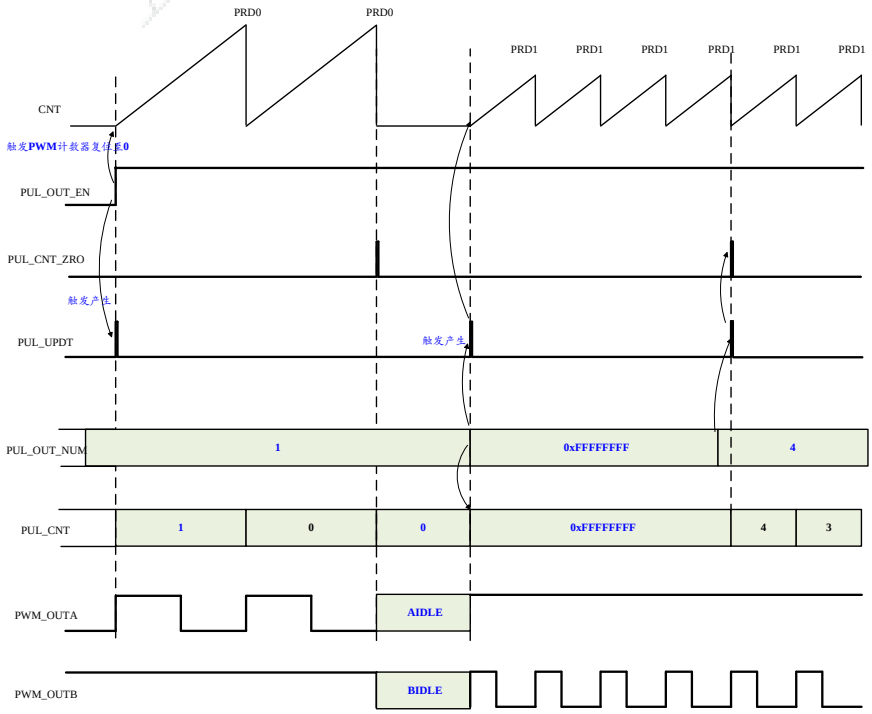


如果在脉冲群发送的过程中，修改了 PUL_OUT_NUM 的数值，那么执行的时序会如下图：

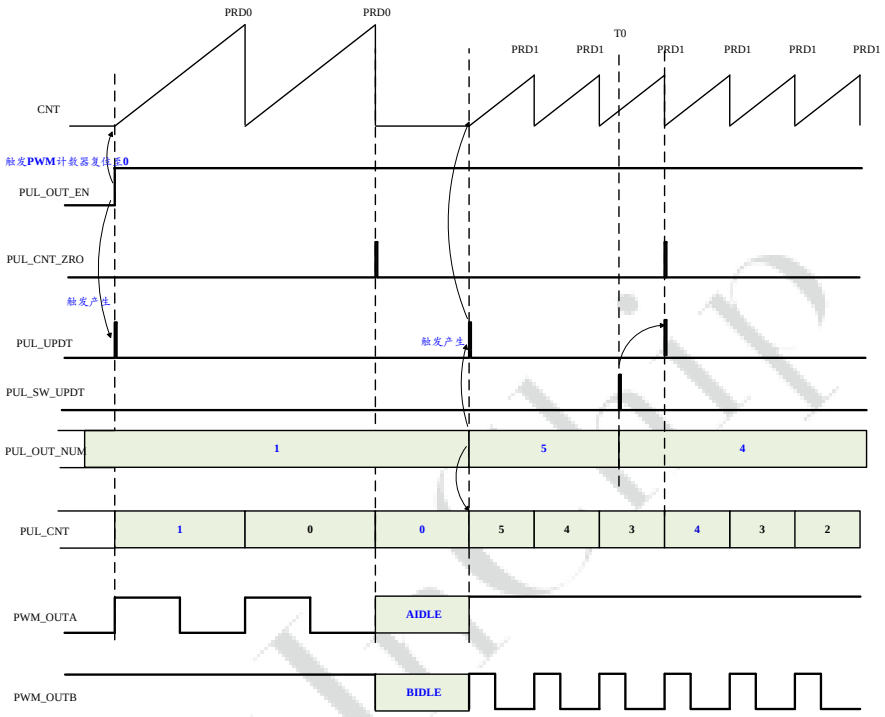
如下图的 T0 时刻对寄存器 PUL_OUT_NUM 写入数值 1 的时刻，此时不会复位计数器的计数，而是会在发送完当前的 4 个脉冲群之后才会执行此时写 PUL_OUT_NUM = 0x1 的脉冲群。



脉冲输出的功能支持强制持续输出脉冲群的功能，对 PUL_OUT_NUM 赋值 0xFFFFFFFF 即可以实现，如需退出持续输出脉冲群的状态，需要对 PUL_OUT_NUM 写入正常值，写入数值后在计数器完成一个周期的计数时更新写入的正常值，，具体如下图所示：



脉冲输出的支持软件强制更新功能，PUL_SW_UPDT 写 1 触发一次强制更新信号 PUL_UPDT，如下图的 T0 时刻。



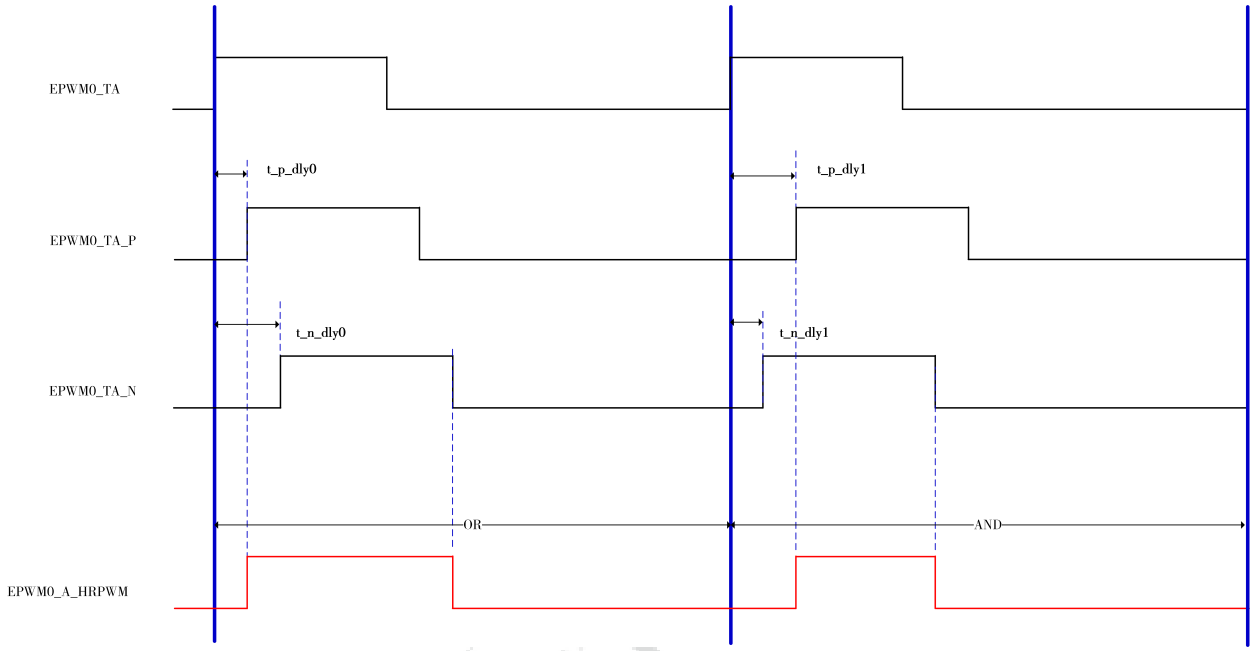
9.1.4.1.9. HRPWM 功能

HRPWM 的处理为 EPWM 最后一级处理模块，HRPWM 通过动态调整 EPWM 输出的上升沿和下降沿，调整的步进最低 130 ps（在 PWMCS 工作时钟设置为 240MHz 的情况），实现高精度 PWM 输出功能。

9.1.4.1.9.1. 输出处理

HRPWM 的输出整形控制上升沿的延迟的寄存器为 HRPWMA/B_P_DLY，控制下降沿的延迟的寄存器为 HRPWMA/B_N_DLY。

下图为 HRPWMA 的输出处理时序图，其中 t_p_dly0/1 为 HRPWMA_P_DLY 控制的延迟时间，t_n_dly0/1 为 HRPWMA_N_DLY 控制的延迟时间，EPWM0_TA 经过 HRPWMA_P_DLY 延迟控制的信号为 EPWM0_TA_P，EPWM0_TA 经过 HRPWMA_N_DLY 延迟控制的信号为 EPWM0_TA_N，最后经过逻辑处理后得出 EPWM0_A_HRPWM。如下图所示：



上图 EPWM0_TA_P、EPWM0_TA_N 经过的以下逻辑处理（HRPWM_AOUT_LOG = 0x0）得到 EPWM0_A_HRPWM：

- 在 t_p_dly <= t_n_dly 的时候，EPWM0_TA_P 与 EPWM0_TA_N 进行 OR 逻辑处理输出得到 EPWM0_A_HRPWM
- 在 t_p_dly > t_n_dly 的时候，EPWM0_TA_P 与 EPWM0_TA_N 进行 AND 逻辑处理输出得到 EPWM0_A_HRPWM

9.1.4.1.9.2. 互补输出

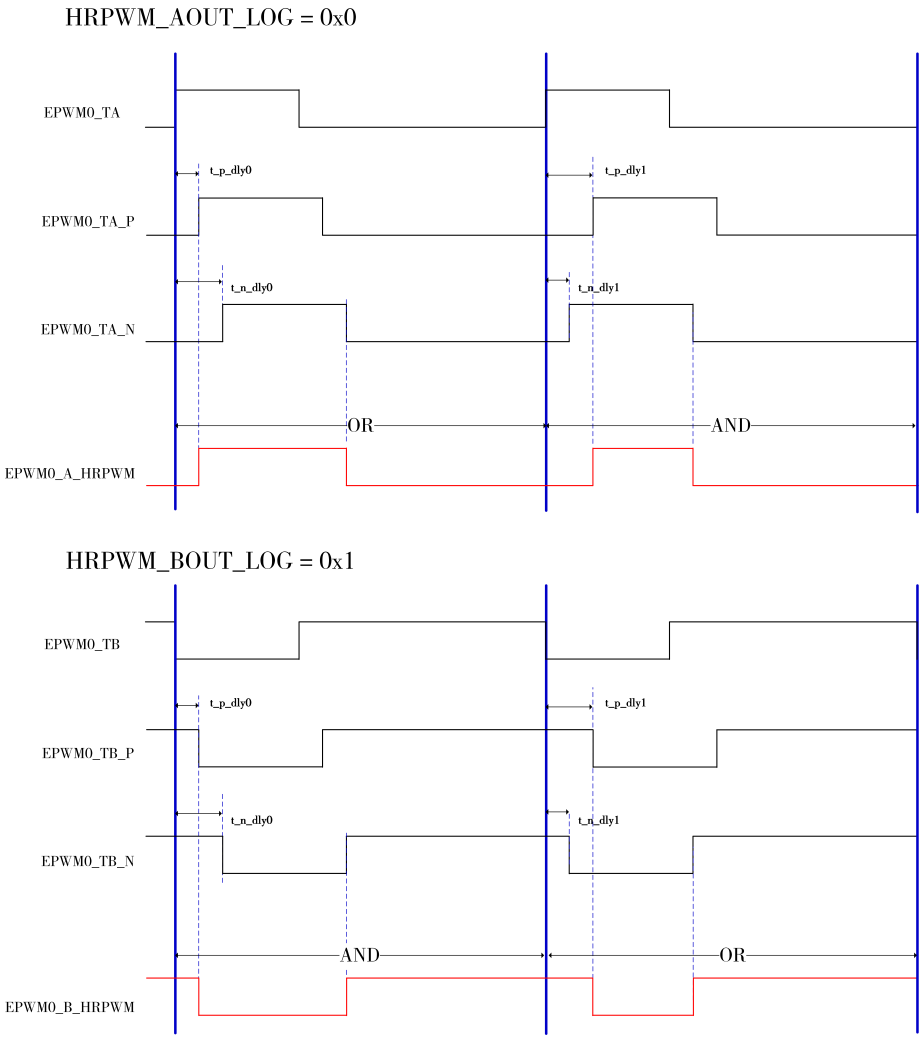


图 9–27 HRPWM 逻辑处理

EPWM_A_HRPWM 和 EPWM_B_HRPWM 的输出逻辑处理，可以分别通过 HRPWM_AOUT_LOG 和 HRPWM_BOUT_LOG 进行选择正常逻辑处理或互补逻辑处理。

互补逻辑处理方式与正常逻辑处理相反，具体如下：

- 在 t_p_dly <= t_n_dly 的时候，EPWM_TA/B_P 与 EPWM_TA/B_N 进行 AND 逻辑处理输出得到 EPWM0_A_HRPWM
- 在 t_p_dly > t_n_dly 的时候，EPWM_TA/B_P 与 EPWM_TA/B_N 进行 OR 逻辑处理输出得到 EPWM0_A_HRPWM

9.1.4.1.9.3. 高精度频率输出

高精度频率输出，是增加输出 EPWM 信号频率的可调精度的功能。

高精度的周期的功能通过设置寄存器 HPWM_PRD_EN = 0x1 后，高精度周期值生效，也就是 HRPWM_PRD_VAL 生效。寄存器字段 HRPWM_PRD_VAL 为影子更新，更新的时间点 可通过 HRPWM_PRD_VAL_LP 配置。

下面举例说明高精度周期的生效方式：

以 EPWM0 为例，计数器时钟为 240 MHz，设置 EPWM0 的计数器为递增计数，设置 PRD 值为 1，那么这里对应整数周期是 2 * 4.16 ns，而在高精度的 EPWM 周期值上设置为 8，即是小数周期实现 8 * 130 ps = 1.04 ns。所以经过整数和小数综合得到实现的周期为 2 * 4.16 ns + 8 * 130 ps = 9.36 ns。

注：

若需要同时进行高精度频率+占空比输出，动作的比较值只能采用 CMPA 进行。

9.1.4.1.10. Debug Hold 功能

调试进入断点模式，触发 EPWM 保护模式，进行 EPWM 的不可恢复输出保护，输出的电平由 EPWM 故障保护的寄存器决定。

调试进入断点模式，可通过 EPWM_EM_CTL 选择 EPWM 内部计数器的行为:

- 0x0：在调试断点停止计数
- 0x1：在调试断点时停止计数，直到计数完成当前周期
- 0x2/0x3：不停止计数

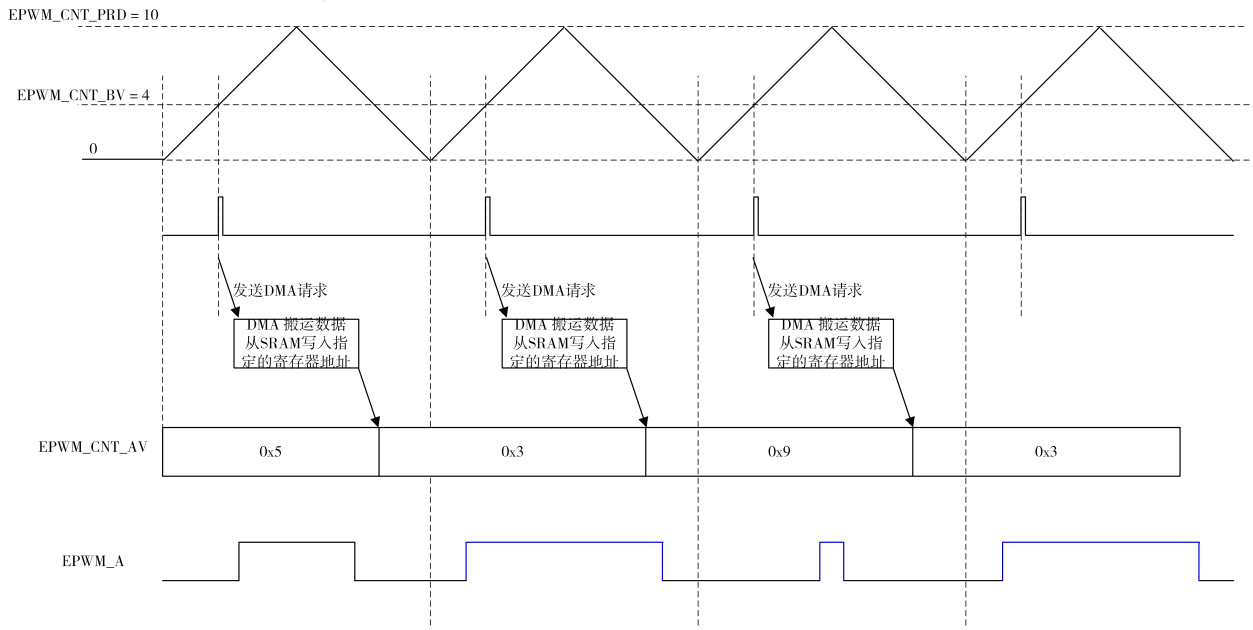
9.1.4.1.11. DMA 功能

EPWM 支持 DMA 的功能，使能 DMA 后，DMA 的请求信号可以通过寄存器配置，选择 EPWM_CNT_Z、EPWM_CNT_PRD、EPWM_CNT_UA、EPWM_CNT_DA、EPWM_CNT_UB、EPWM_CNT_DB、EPWM_CNT_UC、E

DMA 的搬运支持单次对一个寄存器进行写入，或者单次对多个寄存器写入的模式。

- 单次对一个寄存器写入，在 DMA 请求信号到来，DMA 根据 DMA 通道的配置，将 SRAM 的数据写到 EPWM 的某一寄存器地址中。下图举例为在 EPWM_CNT_UB 的时刻触发 DMA，对寄存器 EPWM_CNT_AV 的数值进行修改。

：



- 单次对多个寄存器写入，这里需要配置 EPWM 的寄存器 DMA_MAP_EN = 0x1，使能 DMA 的映射功能，并且通过寄存器 DMA_CYC_NUM 配置单次写入寄存器的数量，最多支持 8 个寄存器的写入。另外 DMA 的映射功能使能后，通过寄存器 EPWM_DMA_MAP0 和 EPWM_DMA_MAP1 进行映射表的设置。

  现在假设需要进行单次对两个寄存器的写入（分别是 EPWM_CNT_PRD 和 EPWM_CNT_AV），那么初始化的配置如下：

1、DMA_MAP_EN= 0x1，使能 DMA 的寄存器映射功能；

2、DMA_CYC_NUM= 0x1，设置 DMA 单次写入的寄存器的数量为 2；

3、设置 EPWM_DMA_MAP0 字段的 DMA_CNT0_MAP 和 DMA_CNT1_MAP ，DMA_CNT0_MAP 代表的是 DMA 写入第一个寄存器的偏移地址，这里设置 EPWM_CNT_PRD 的偏移地址 0x00，而 DMA_CNT1_MAP 代表的是第二个寄存器的偏移地址，这里设置 EPWM_CNT_AV 的偏移地址 0x14;

4、需要写入寄存器的数值，可以对寄存器地址 EPWM_DMA_ENTRY 依次写入；

5、通过上述的配置后，在 DMA 请求触发 DMA 动作后，DMA 会按照映射表将 EPWM_DMA_ENTRY 的内容依次写入到目标地址。

9.1.4.2. CAP 模块

CAP 模块，具有以下模式：

- 捕捉模式：一种输入模式，对输入的信号进行边沿的捕获，从而获得周期等时间信息。
- 普通 PWM 输出：一种输出模式，产生普通 PWM 信号。

注：

CAP 模块 CAP0/ 1/ .../ 5 均为相同模块，各个模块的信号均相同，但会增加标号 0/1/…/5，用于区分不同模块的信号。例如 CNT_CLK0 代表 CAP0 的计数器时钟信号 CNT_CLK。

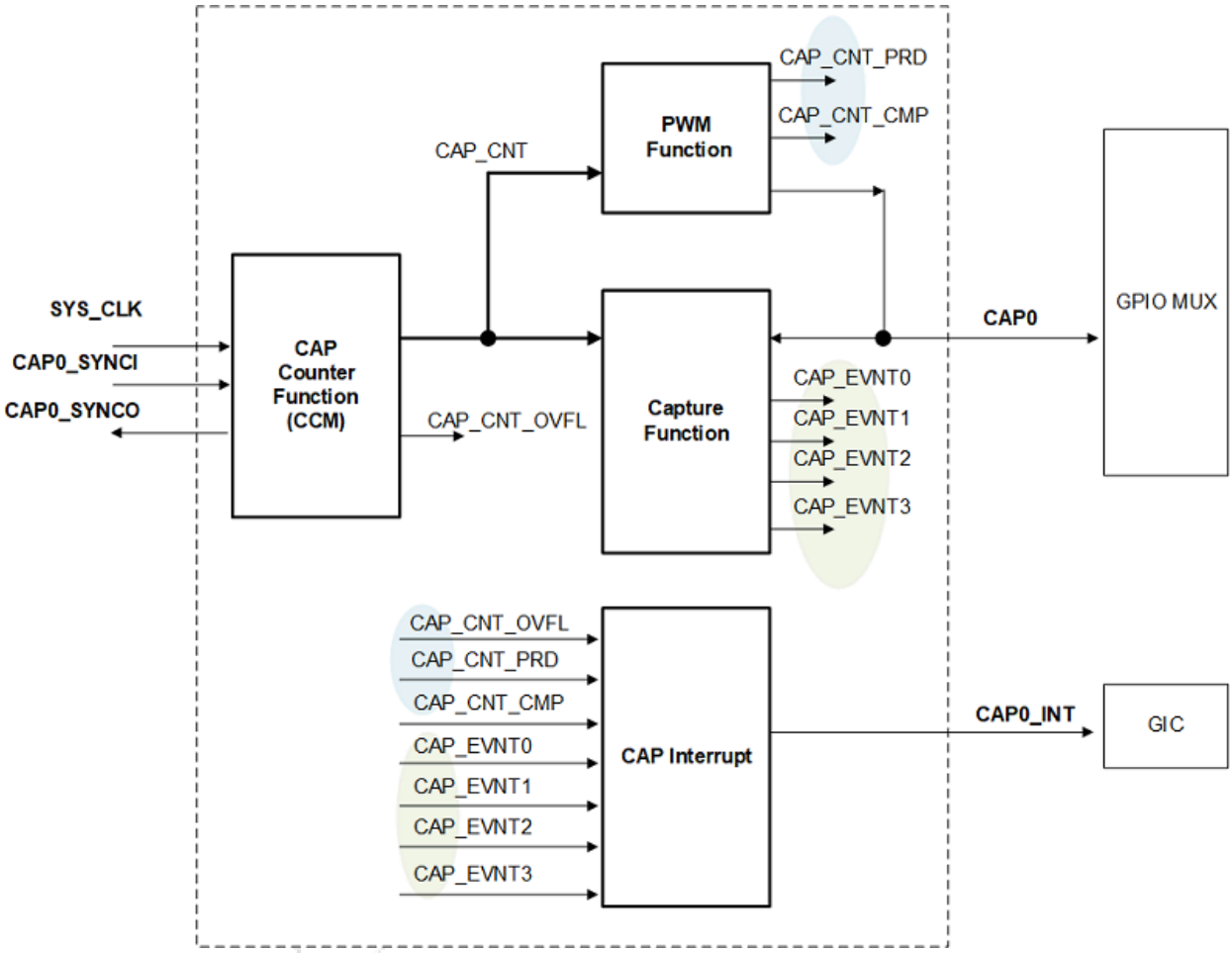


图 9–28 CAP0 的结构框图

图 9–28 : CAP0 的结构框图中：

- CAP0 与 CAP0/ 1/ .../ 5 的结构基本一致，SYS_CLK 为 CAP 模块工作时钟，典型值为 240MHz；
- CAP0 主要由计数功能模块、PWM 功能模块、捕捉功能模块和中断功能模块组成，其中 PWM 功能模块与捕捉功能模块是二选一功能。

9.1.4.2.1. 计数器功能

捕捉模式以及简单 PWM 输出模式，采用相同的计数器模块，使用 32–bit 计数器。

- **计数时钟**

  CAP 模块计数器的时钟无分频系数控制，直接由 SYS_CLK 驱动。

  计数器在计数值等于 CAP_CNT_PRDV（即 PWM 模式下寄存器 CAP_REG0）可产生 CAP_CNT_PRD 信号。

  计数器在计数值等于 CAP_CNT_CMPV（即 PWM 模式下寄存器 CAP_REG1）可产生 CAP_CNT_CMP 信号。

  在计数值达到 0xFFFF_FFFF 时可产生 CAP_CNT_OVFL 信号。

- **计数同步**

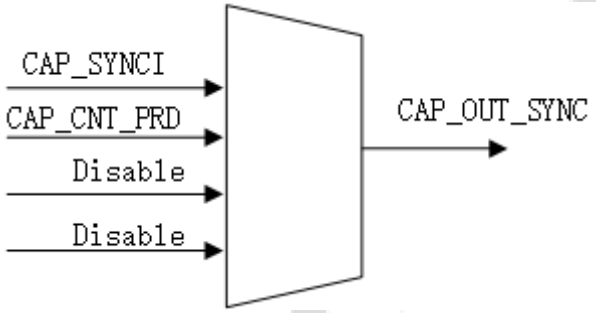
  CAP 模块的计数器支持外部输入信号同步计数器，并支持输出同步信号。

◦ 输入同步

当输入同步信号 CAP_SYNCI 有效，计数器将同步相位寄存器 CAP_CNT_PH 的数值加载至计数器，从而达到计数器相位的同步效果。另外 CAP0/ 1/ .../ 5 的输入同步信号的连接方式，可以参考 计数同步。

◦ 输出同步

通过寄存器 CAP_SYNC_OUT，可以选择 CAP_SYNCI 信号或 CAP_CNT_PRD 信号输出。



9.1.4.2.2. 捕捉输入功能

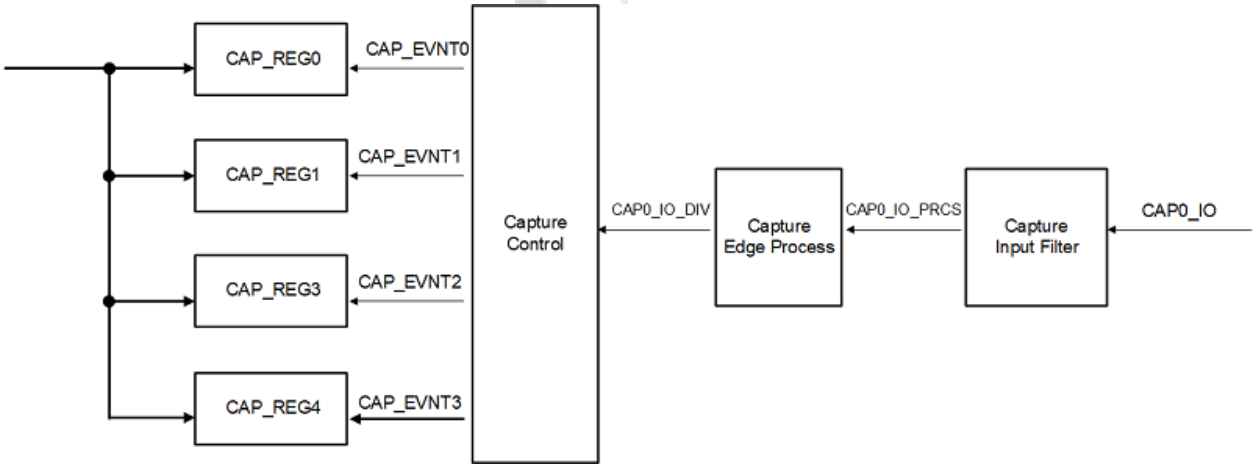


图 9–29 捕捉输入功能框图

- **输入捕捉信号选择**

输入捕捉信号除了支持外部的引脚信号捕捉之外，还可以支持内部的 EPWM 信号、CPM 的比较器输出信号和 EtherCat 的同步信号。

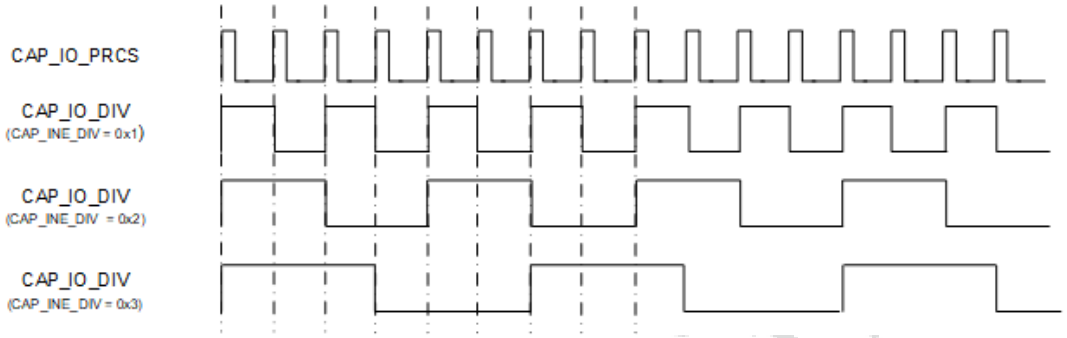
- **输入滤波**

CAP 模块在捕捉输入功能下，输入信号 CAP0/ 1/ .../ 5_IO 均需要经过输入滤波模块，产生 CAP0/ 1/ .../ 5_IO_PRCS 信号。

输入滤波功能，主要是用于滤除噪音信号，功能描述可以参考 EPWM 模块的输入滤波章节。

- **捕捉边沿预处理**

捕捉边沿预处理功能，主要用于对输入信号进行降低频率的处理。 通过配置寄存器字段 CAP_IN_EDG_EVN_DIV 对输入信号 CAP_IO 的边沿翻转进行处理。如设置 CAP_IN_EDG_EVN_DIV = 0x1（如下图中对应配置的例子），输入信号的每两次的翻转边沿才触发一次翻转，从而得到信号 CAP_IO_DIV，这样的预处理可以降低输入信号的翻转频次。



- **捕捉控制**

CAP 模块支持对四次捕捉事件，在捕捉事件到来后，捕捉控制模块会触发对应的捕捉寄存器 CAP_0/ 1/ 2/ 3 进行记录 CAP 的计数器的数值。

捕捉事件 EVENT0/ 1/ 2/ 3 可以根据需求，通过寄存器字段 CAP_EVNT0/ 1/ 2/ 3_POL，设置事件为上升沿捕捉或下降沿捕捉，以及通过寄存器字段 CAP_EVNT0/ 1/ 2/ 3_RST 配置发生捕捉事件后是否复位计数器，这样的配置方式可以计算出两次边沿的时间差值。

- **捕捉寄存器**

CAP 模块配有四个捕捉寄存器 CAP_REG0/ 1/ 2/ 3，每个寄存器对应一个捕捉事件 CAP_EVNT0/ 1/ 2/ 3，当输入信号满足捕捉事件的特性，则产生捕捉信号 CAP_EVNT0/ 1/ 2/ 3，对计数器的数值进行捕捉。

捕捉事件依次按照 EVENT0、EVEMT1、EVENT2、EVENT3 进行，所以捕捉到的计数器数据依次写入寄存器 CAP_REG0、CAP_REG1、CAP_REG2、CAP_REG3。

捕捉状态可以通过寄存器字段 CAP_OS_MOD_EN 配置为连续捕捉模式或单次捕捉模式。连续捕捉模式，那么从 EVENT0 执行到 EVETN3 后，继续从 EVENT0 执行到 EVENT3，按照顺序循环进行。

单次捕捉模式，通过配置寄存器字段 CAP_OS_EP 配置需要捕捉的事件次数，当从 EVENT0 执行到所需的事件次数后，CAP 模块不再进行捕获，CAP_REG0/ 1/ 2/ 3 寄存器的数据锁存当前数据。

- DMA 功能

CAP 支持 DMA 的功能，使能 DMA 后，DMA 的请求信号可以通过寄存器配置，选择 CAP_EVNT0/1/2/3。在每次 DMA 请求信号到来，DMA 根据 DMA 通道的配置，读取 CAP 的寄存器 CAP_REG0/1/2/3 的数值并且写到 SRAM。

9.1.4.2.3. 普通 PWM 输出功能

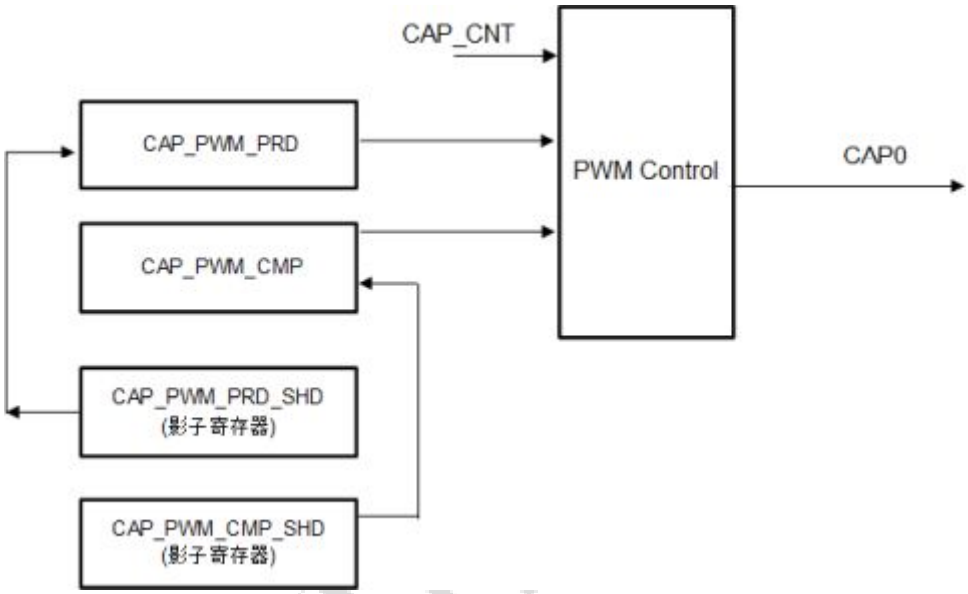


图 9–30 简易 PWM 输出功能框图

PWM 周期以及比较值寄存器

在普通 PWM 模式下，寄存器 CAP_REG0/ 1/ 2/ 3 的复用情况如下所示：

- CAP_REG0 作为 CAP_CNT_PRD 使用，用于设置 PWM 输出信号的周期值，并且在计数值达到此值时，PWM 输出的信号设置为高。
- CAP_REG1 作为 CAP_CNT_CMP 使用，用于设置 PWM 输出信号的比较值，并且在计数值达到此值时，PWM 输出的信号设置为低，用来调节 PWM 信号的脉宽。
- CAP_REG2 作为 CAP_CNT_PRD 的影子寄存器使用，影子寄存器在计数器的数值达到 PRD 时更新。
- CAP_REG3 作为 CAP_CNT_CMP 的影子寄存器使用，影子寄存器在计数器的数值达到 PRD 时更新。

因此对 CAP_CNT_PRD 写操作，CAP_CNT_PRD 直接更新，若对影子寄存器 CAP_CNT_PRD_SHD 写操作，则在计数器数值达到周期值时才进行更新。CAP_CNT_CMP 寄存器的更新方式也是同理。

另外，PWM 的输出极性可以利用寄存器 CAP_PWM_POL，配置 PWM 最终输出的极性，即有效电平为高还是低。

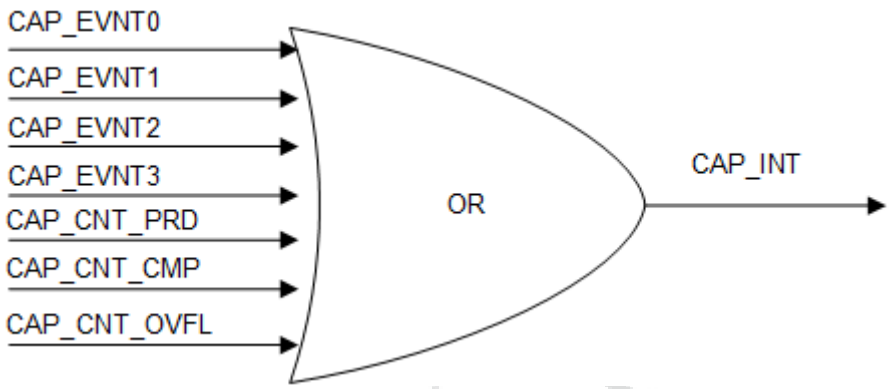
9.1.4.2.4. 中断信号

CAP 模块的中断信号 CAP_INT，由下列信号进行或逻辑产生：

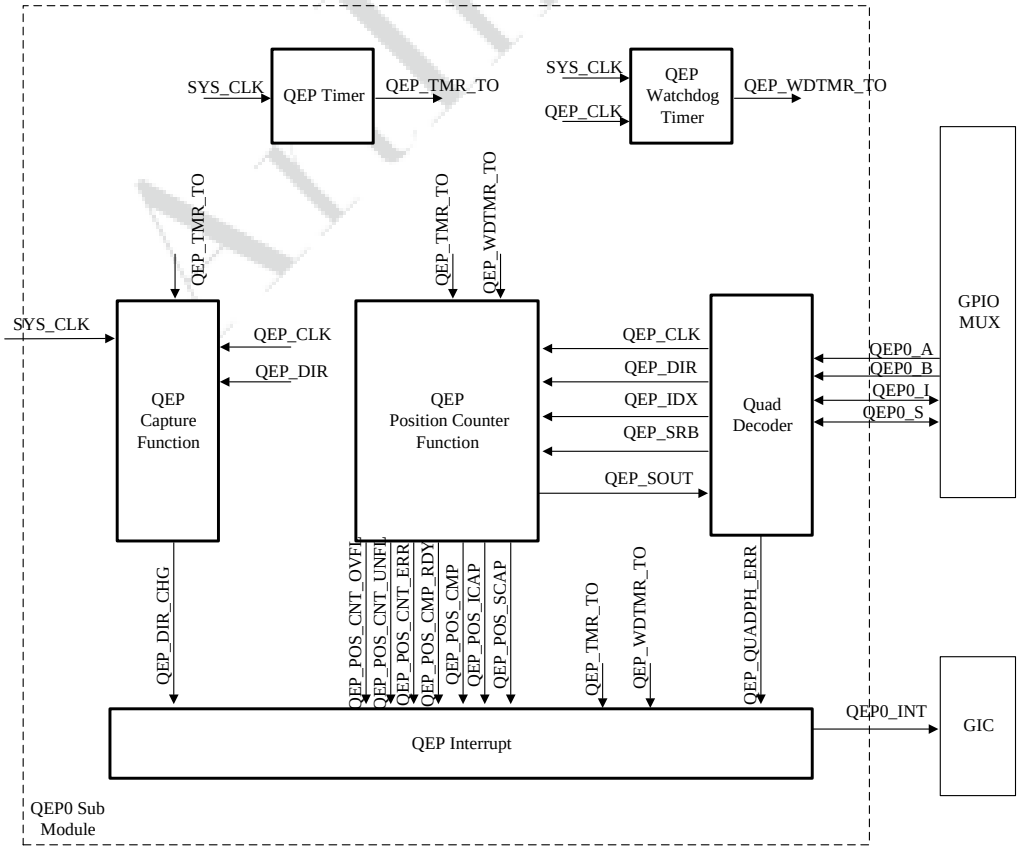
- CAP_EVNT0
- CAP_EVNT1
- CAP_EVNT2
- CAP_EVNT3
- CAP_CNT_PRD
- CAP_CNT_CMP
- CAP_CNT_OVFL

可以配置寄存器 CAP_INT_EN，选择是否使能对应的信号进行或逻辑运算。

CAP 模块的中断信号源或逻辑示例如下：



9.1.4.3. QEP 模块



QEP 模块主要由下列模块构成：

- Quad Decoder：正交解码模块，用于将正交信号 A 相、B 相进行正交信号的解码，获得后续 QEP Position Counter 需要的计数时钟 QEP_CLK 以及计数方向 QEP_DIR 等信息。
- QEP Position Counter Function：位置计数器功能模块，根据正交解码模块解析的信号进行位置计数器的计数。
- QEP Capture Function：QEP 捕捉功能，用于捕捉特定的位置距离所需要的时间，从而计算转速等信息。
- QEP Timer：QEP 定时器，用于产生单位时间间隔的信号。

- QEP Watchdog Timer：QEP 看门狗定时器。用于检测预设的时间内是否仍有有效 QEP_CLK 信号。
- QEP Interrupt：QEP 中断模块，用于产生中断信号。

SYS_CLK 为 QEP 的工作时钟，典型频率为 240 MHz。

9.1.4.3.1. 正交解码

通过寄存器字段 QEP_DEC_CNT_MODE，可配置解码器的以下几种解码方式：

- QEP_DEC _CNT_MODE = 0x0，正交计数解码
- QEP_DEC _CNT_MODE = 0x1，方向计数解码
- QEP_DEC _CNT_MODE = 0x2，递增计数解码
- QEP_DEC _CNT_MODE = 0x3，递减计数解码
- QEP_DEC _CNT_MODE = 0x4，高有效 CW/CCW 计数解码
- QEP_DEC _CNT_MODE = 0x5，低有效 CW/CCW 计数解码

9.1.4.3.1.1. 正交计数解码

正交计数解码方式，外部信号 QEP_A 和 QEP_B 输入为正交信号，根据 QEP_A 与 Q_EPB 的信号的相位信息解析出提供后续解码计数器的步进信号 QEP_CLK 以及步进方向信号 QEP_DIR，如图 9–31 : 正交计数解码所示。

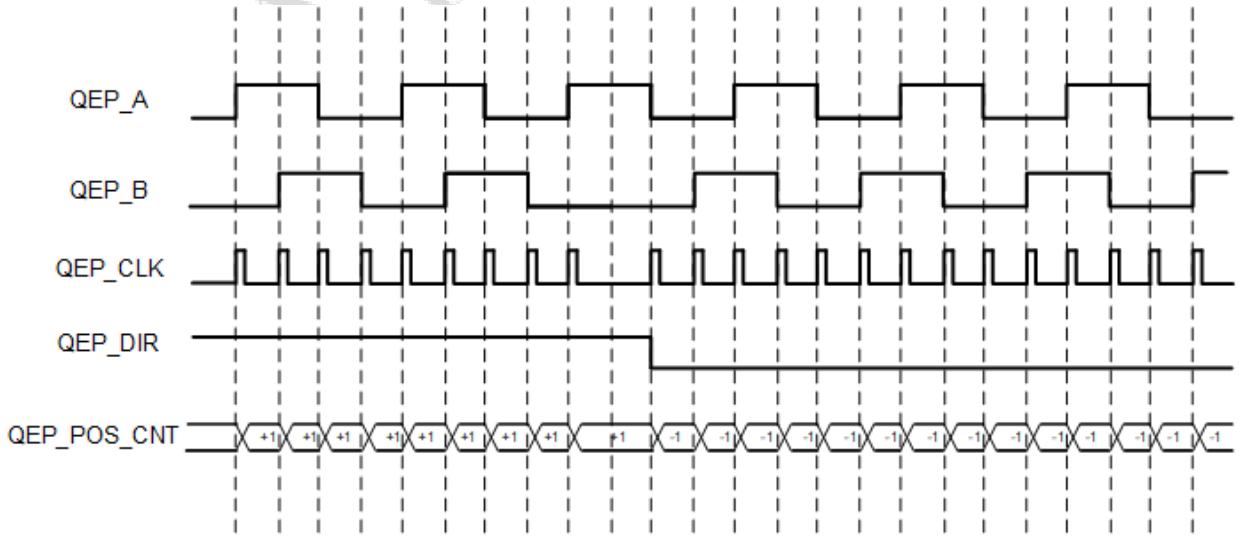


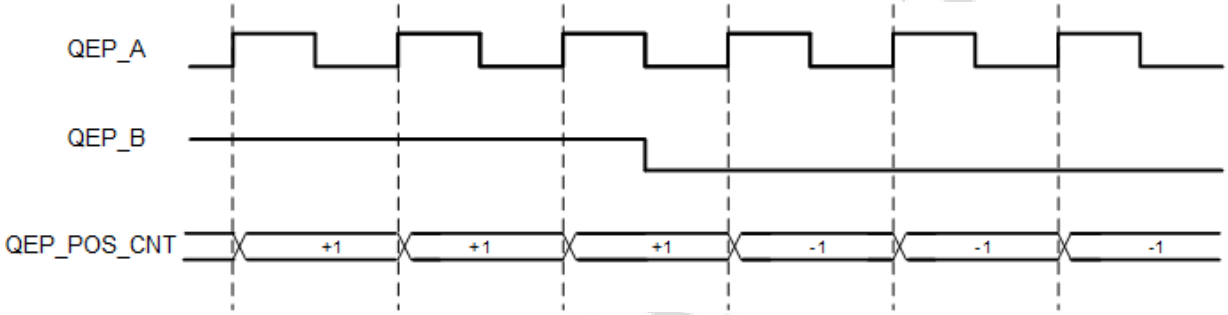
图 9–31 正交计数解码

另外正交计数解码还具有以下的功能：

- 相位异常检测：输入的 QEP_A 以及 QEP_B 的相位不符合正常的相位跳转模式，则产生正交信号异常的中断信号 QEP_QUADPH_ERR；
- 互换功能：通过寄存器字段 QEP_SWAP_AB 配置为 1， 可将 QEP_A 和 QEP_B 的输入信号互换。

9.1.4.3.1.2. 方向计数解码

方向计数解码方式，则无需经过正交信号的解码，外部信号 QEP_A 和 QEP_B 输入分别作为时钟信号 QEP_CLK 和方向信号 QEP_DIR，QEP_DIR 决定计数的方向，QEP_CLK 用于控制计数的步进。 默认的配置下(即输入信号的极性未进行方向反相)，QEP_DIR 为高电平的时候，表示正交计数器进行递增计数，QEP_DIR 为低电平的时候，表示正交计数器进行递减计数。 另外 QEP_A 的信号可以配置为上升沿触发一次 QEP_CLK 有效边沿，或上下沿均可触发一次的 CLK 有效边沿。



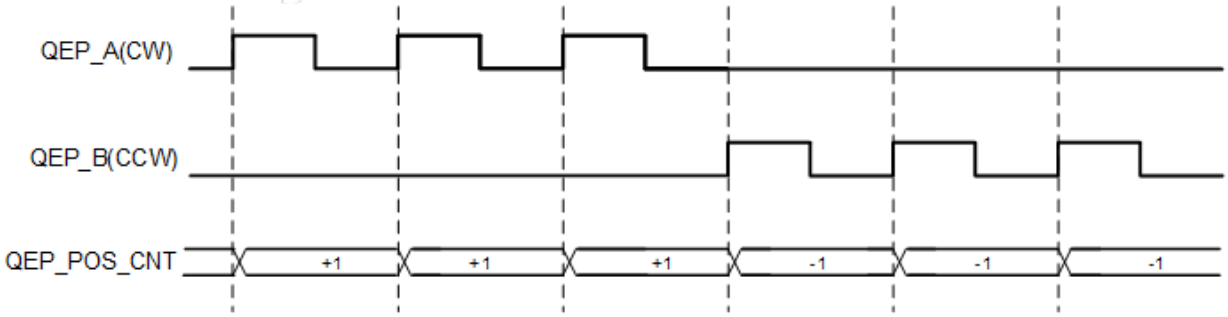
9.1.4.3.1.3. 递增/递减计数解码

内部信号 QEP_DIR 固定选定为递增/递减的方向，此时只需 QEP_A 的输入信号，而 QEP_B 信号无效。另外 QEP_A 的信号可以配置为上升沿触发一次 QEP_CLK 有效边沿，或上下沿均可触发一次的 QEP_CLK 有效边沿。

9.1.4.3.1.4. CW/ CCW 计数解码

CW/ CCW 计数解码方式包括高有效的 CW/ CCW 解码计数和低有效的 CW/ CCW 解码计数：

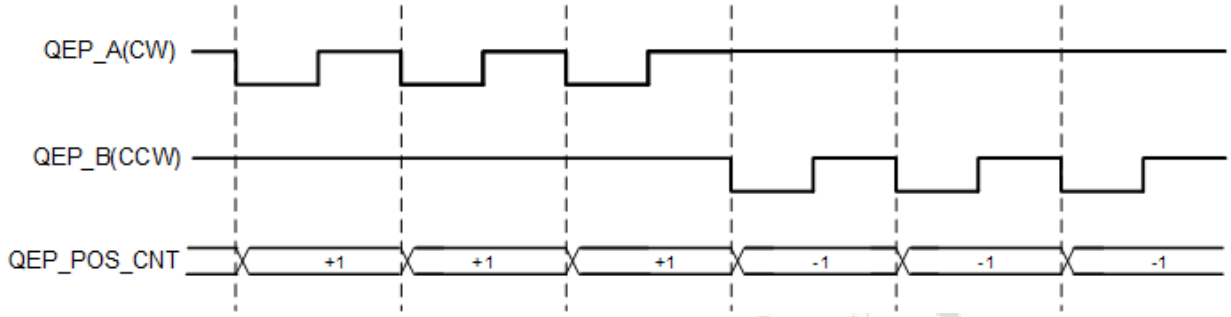
- 高有效的 CW/CCW 解码方式



高有效是指低电平为无效电平，高电平为有效电平，当从低电平跳至高电平产生的上升边沿为有效边沿。若此有效边沿出现在 QEP_A 信号，即 CW 信号，那么则位置计数器正向加 1；若此有效边沿出现在 QEP_B 信号，即 CCW 信号，那么则位置计数器反向减 1。

另外如若出现 QEP_A 和 QEP_B 同时处于有效电平高电平时，则产生 CW/ CCW 错误中断信号，QEP_CW_CCW_ERR_INT。

- 低有效的 CW/CCW 解码方式



低有效是指高电平为无效电平，低电平为有效电平，当从高电平跳至低电平产生的下降边沿为有效边沿。若此有效边沿出现在 QEP_A 信号，即 CW 信号，那么则位置计数器正向加 1；若此有效边沿出现在 QEP_B 信号，即 CCW 信号，那么则位置计数器反向减 1。

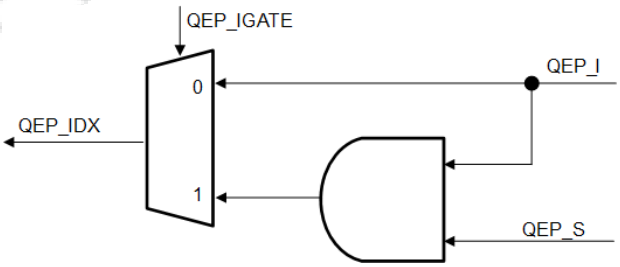
另外如若出现 QEP_A 和 QEP_B 同时处于有效电平低电平时，则产生 CW/CCW 错误中断信号，QEP_CW_CCW_ERR_INT。

9.1.4.3.1.5. 输入反相

QEP 模块的输入信号 QEP_A、QEP_B、QEP_I 和 QEP_S，均可以通过寄存器配置相应的输入反相器是否使能。使能后，输入信号为低有效信号。

9.1.4.3.1.6. 输入 QEP_IGATE

输入 QEP 模块的输入信号 QEP_I 可以通过寄存器字段 QEP_IGATE 配置是否需要使能信号 QEP_S 对 QEP_I 进行 Gate 处理，处理方式如下图。



9.1.4.3.1.7. 位置比较输出

QEP Position Counter Function 模块输出的信号 QEP_SOUT 信号，是位置计数器达到寄存器 QEP_POS_CNT_CMPV 时所产生的脉冲信号，该信号输出至正交解码模块，这里可以通过寄存器字段 QEP_SYNC_OUT_PIN 选择，使能输出为 QEP_I 或 QEP_S 信号。

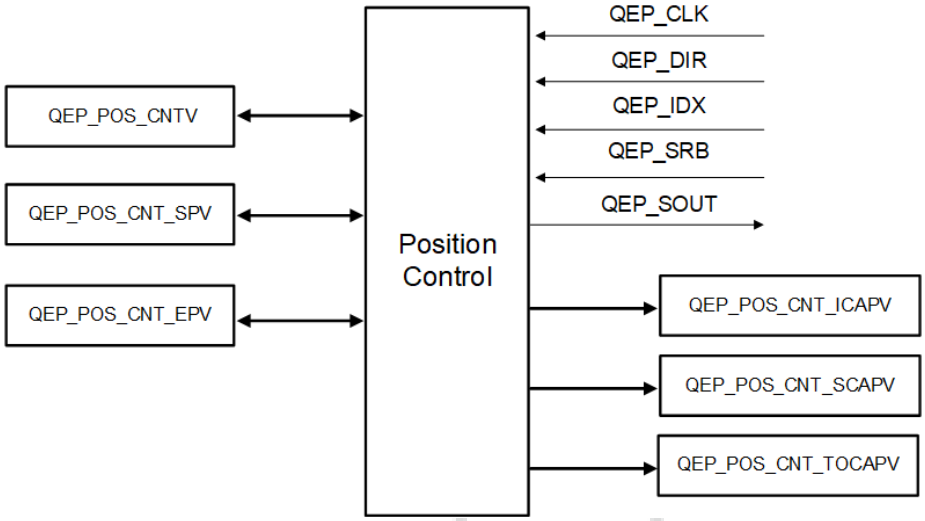
QEP 模块的信号 QEP_A、QEP_B 为外部输入信号，支持输入信号的滤波。而 QEP 模块的信号 QEP_I、QEP_S 为双向信号，当设置为输入信号时，支持输入信号的滤波。

9.1.4.3.1.8. 输入滤波

输入滤波功能，主要是用于滤除噪音信号，功能描述可以参考 EPWM 模块的输入滤波章节描述。

9.1.4.3.2. 正交位置计数功能

功能框图



正交位置计数器有四种复位模式，可通过寄存器字段 QEP_POS_CNT_RST 配置：

- 复位模式 1：QEP_IDX_MKR 信号复位位置计数器，即只要出现 QEP_IDX_MKR 信号，正交位置计数器则进行一次复位：

  注：

  复位模式 1 下，会进行锁存值的判断检测。

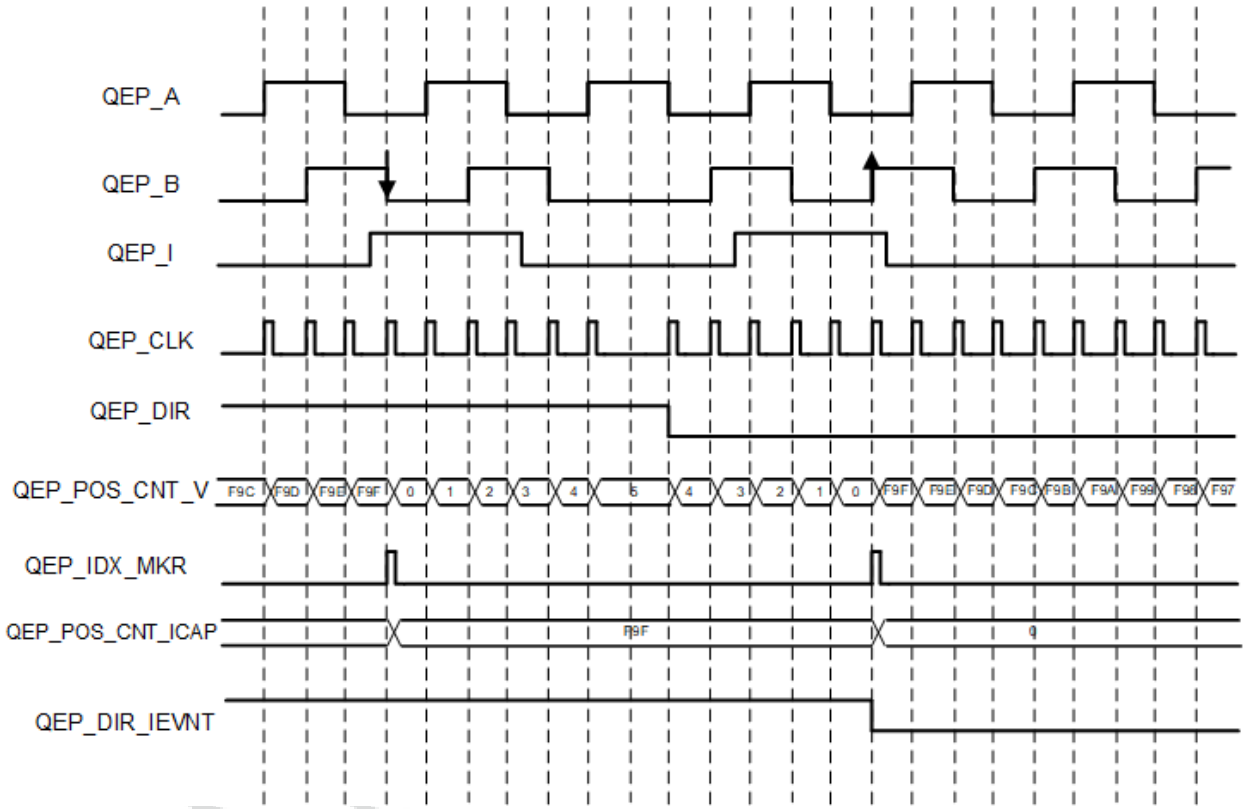
  - 若为正向前进，复位值为 0。

    锁存值的判断检测：在正向前进的情况下，如果 MKR 信号触发写入寄存器 QEP_POS_CNT_ICAPV 的数值与 QEP_POS_CNT_EPV 数值不一致，则 QEP_POS_CNT_ERR_FLG 置 1 且 QEP_POS_CNT_ERR_INT_FLG 置 1，表示产生位置计数的错误。

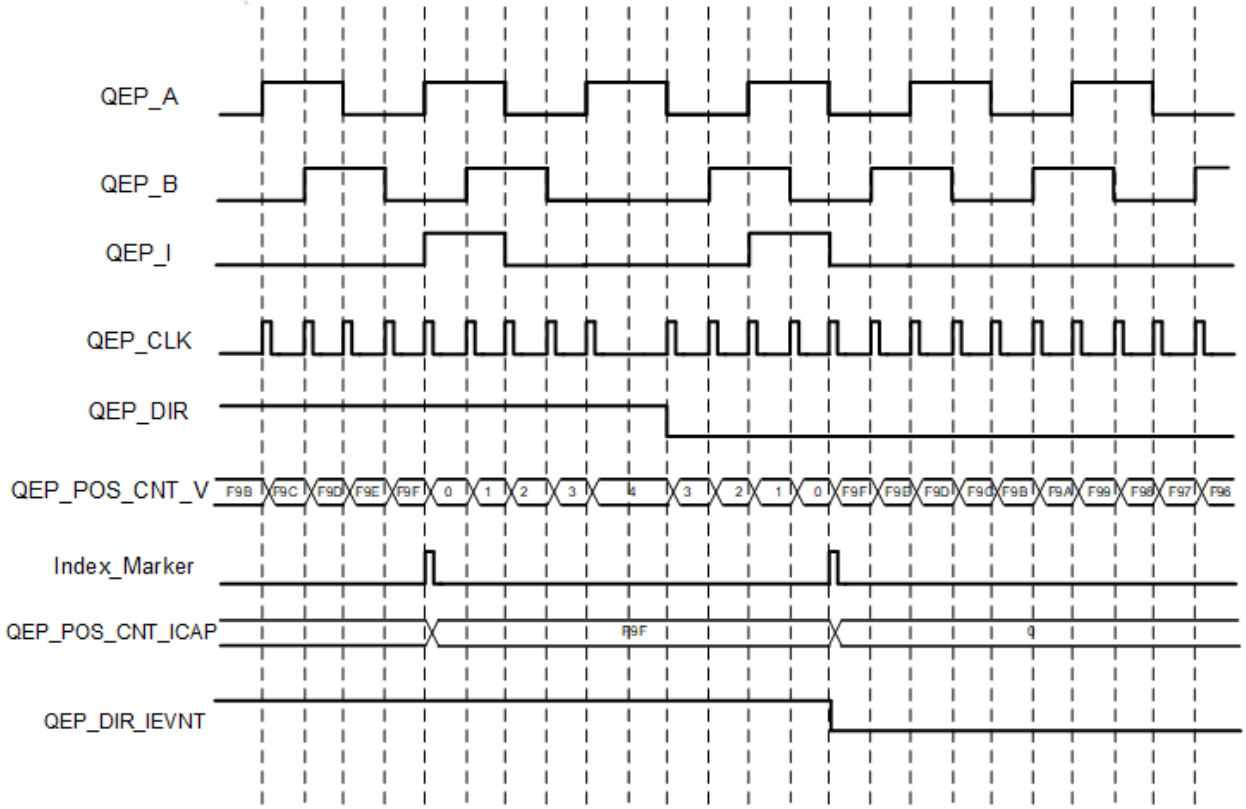
  - 若为反向前进，复位值为终点值 QEP_POS_CNT_EPV。

  QEP_IDX_MKR 信号是表示位置回到 0 点位置。QEP_IDX_MKR 信号的产生与输入管脚信号 QEP_I 有关。以下三张示例图片分别描述了三种不同 QEP_I 输入信号所对应的 QEP_IDX_MKR 复位位置寄存器，其中位置寄存器的计数采用十六进制表示。

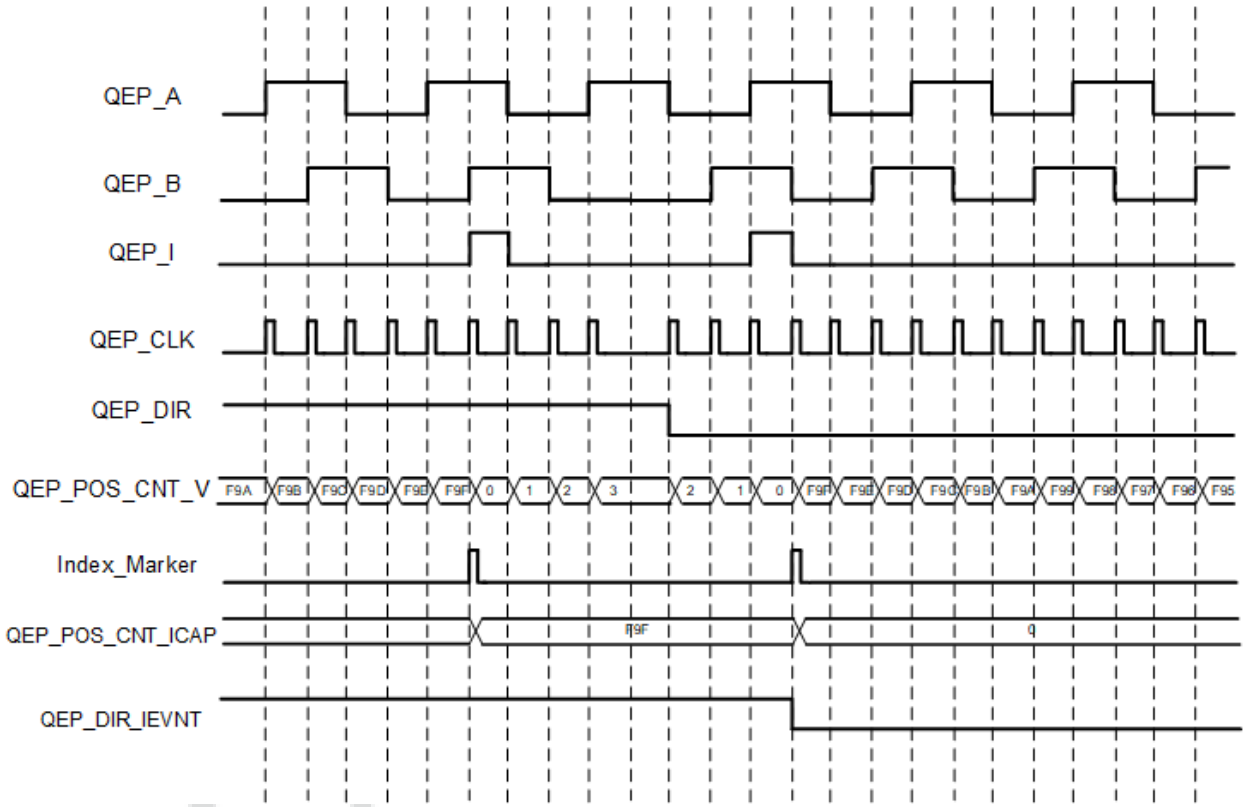
◦ QEP_IDX_MKR 信号复位位置计数：



◦ QEP_IDX_MKR 信号复位位置计数 (A Gated QEP_I)：



◦ QEP_IDX_MKR 信号复位位置计数 (A & B Gated QEP_I)



• 复位模式 2，QEP_POS_CNT_EP 信号复位位置计数。

QEP_POS_CNT_EP 信号表示位置计数器发生计数溢出的信号：

◦ 递增计数时，当计数达到 QEP_POS_CNT_EPV 时，复位计数器数值为 0，同时产生 QEP_POS_CNT_OVFL 中断信号，指示位置计数器计数上溢出。

◦ 递减计数时，当计数达到 0 时，复位计数器数值为 QEP_POS_CNT_EPV，同时产生 QEP_POS_CNT_UNFL 信号，指示计数器计数下溢出。

• 复位模式 3，QEP_FIDX_MKR 信号复位位置计数。QEP_FIDX_MKR 信号，为 QEP 模块使能后第一次检测到的 QEP_IDX_MKR 信号。

此模式下，位置计数器会在 QEP_FIDX_MKR 信号进行复位：

◦ 如果为正向前进，复位至 0。

◦ 如果为反向则复位至 QEP_POS_CNT_EPV。

◦ 由于后续不会再产生 QEP_FIDX_MKR 信号，后续的位置计数器则通过 QEP_POS_CNT_EP 信号复位，即按照模式 2 进行复位。

• 复位模式 4，QEP_TMR_TO 信号复位位置计数。QEP_TMR_TO 信号，为 QEP 定时器超时产生的信号。

此模式下，位置计数器会在 QEP_TMR_TO 超时信号进行复位：

◦ 如果为正向前进，复位至 0

◦ 如果为反向，复位至 QEP_POS_CNT_EPV。

正交位置计数器的捕获

位置计数器的捕获支持两种，Index 相关事件触发的捕获以及 Strobe 相关事件触发的捕获。

- Index 相关事件的触发捕获，通过寄存器可选择触发捕获位置计数器的 Index 事件。

  QEP_POS_ICAP_MOD = 0x1，使能 QEP_IDX 信号的上升沿触发捕获位置计数器，位置计数值 QEP_POS_CNT_V 写入 ICAPV。

  QEP_POS_ICAP_MOD = 0x2，使能 QEP_IDX 信号的下降沿触发捕获位置计数器，位置计数值 QEP_POS_CNT_V 写入 ICAPV。

  QEP_POS_ICAP_MOD = 0x3，使能 MKR 信号触发捕获位置计数器，位置计数值 QEP _POS_CNT_V 写入 ICAPV。

- Strobe 相关事件的触发捕获，通过寄存器可选择触发捕获位置计数器的 Strobe 事件。

  QEP_POS_SCAP_MOD = 0x0，使能 SRB 信号的上升沿触发捕获位置计数器，位置计数值 QEP_POS_CNT_V 写入 SCAPV。

  QEP_POS_SCAP_MOD = 0x1，使能根据方向进行捕获。正向前进时，以 SRB 上升沿进行捕获，反向前进时，以 SRB 下降沿进行捕获，位置计数值 QEP_POS_CNT_V 写入 SCAPV。

正交位置计数器的初始化

位置计数器的初始化，即更新当前位置计数器的计数值，支持三种模式的初始化：

- Index 相关事件的初始化，通过寄存器可选择触发位置计数器初始化的 Index 事件。

  QEP_POS_IDX_INIT = 0x2，使能 IDX 信号的上升沿触发位置计数器的初始化，位置起点值 SPV 写入位置计数器。

  QEP_POS_IDX_INIT = 0x3，使能 IDX 信号的下降沿触发位置计数器的初始化，位置起点值 SPV 写入位置计数器。

- Strobe 相关事件的初始化，通过寄存器可选择触发位置计数器初始化的 Strobe 事件。

  QEP_POS_SRB_INIT = 0x2，使能 SRB 信号的上升沿触发位置计数器的初始化，位置起点值 SPV 写入位置计数器。

  QEP_POS_SRB_INIT = 0x3，使能 SRB 信号的下降沿触发位置计数器的初始化，位置起点值 SPV 写入位置计数器。

- 软件初始化，寄存器 INIT 字段置 1 可触发位置计数器初始化，位置起点值 SPV 写入位置计数器。

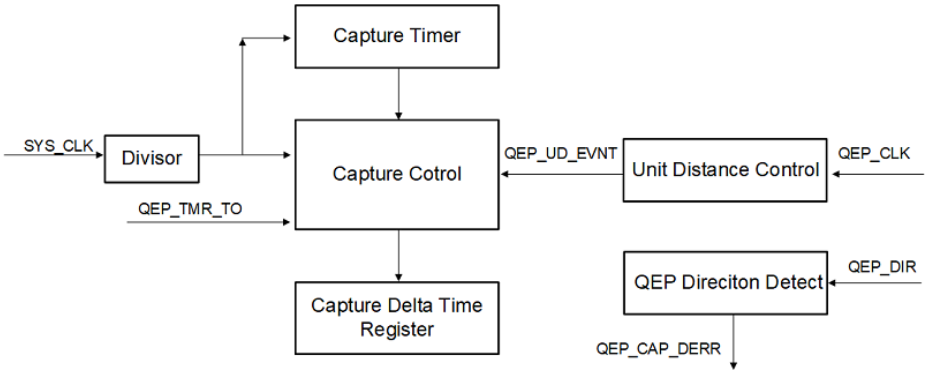
QEP_SOUT 信号

QEP_SOUT 信号，为位置计数器达到 QEP_POS_CNT_CMPV 的时候产生的信号，仅在 QEP_POS_CMP_EN = 1 时产生。QEP_SOUT 信号支持极性的控制，以及脉宽长度的调节。

寄存器 QEP_POS_CNT_CMPV 的更新方式支持立即更新以及影子更新模式。影子更新的加载点也可以通过寄存器配置，具体配置见 QEP_POS_CMP_LP 和 QEP_POS_CMP_UPDT_MOD 的描述。

QEP_SOUT 信号的输出方式见位置比较输出章节。

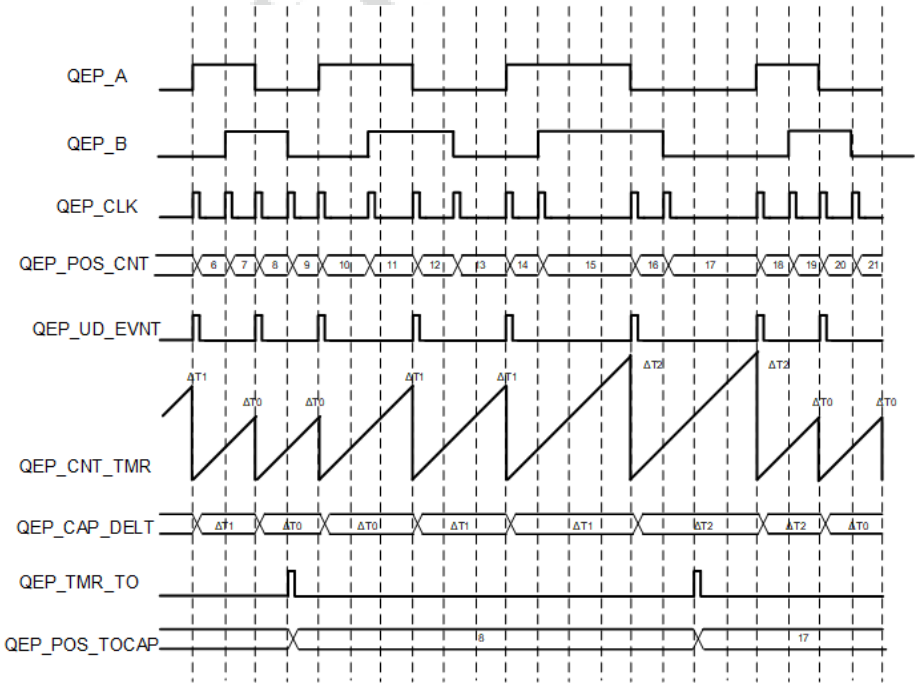
9.1.4.3.3. 正交时间捕捉功能



时间捕捉功能的框图如上图所示，具体的功能描述如下：

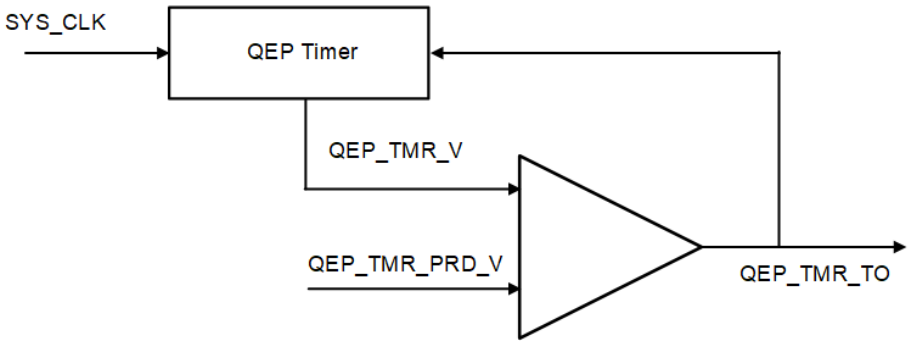
- SYS_CLK 信号，利用寄存器字段 DIV 进行分频，分频后提供给 Capture 定时器以及 Capture 控制。
- QEP_CLK 信号，表示 1 次位置计数器的计数，经过 Unit Distance Control 功能模块对 QEP_CLK 进行分频，对应就可以获得单位距离的信号 QEP_UD_EVNT，下图为设定 QEP_SET_UDIST = 0x1，即 2 次 CLK 的对应产生一次的 EVNT。
- QEP_UD_EVNT 信号，会触发 Capture 控制模块进行 Capture 定时器的数值捕获，进而计算与上次数值作差，写入 QEP_CAP_DELT 寄存器中，随后对 Capture 定时器进行复位至 0，如下图所示。
- 通过 QEP_UD_EVNT 信号触发获得 QEP_CAP_DELT，对于电机的应用，可以实现 T 法的速度计算。
- QEP_TMR_TO 信号，为 Timer 的超时信号（可参考章节 QEP Timer），也可以触发寄存器 QEP_CAP_TMRV 锁存至 QEP_CAP_TMR_LH，而 QEP_CAP_DELT 锁存至 QEP_CAP_DELT_LH。
- 通过 TO 信号触发读取 QEP_POS_TOCAP 的位置值，对于电机的应用，可以实现 M 法的速度计算。
- QEP_DIR 输入至方向检测模块，此模块检测在 EVNT 的有效边沿间隔中是否出现方向改变的状态，若出现方向改变，QEP_CAP_DERR 信号置 1。

QEP Capture Timer 时间捕获：



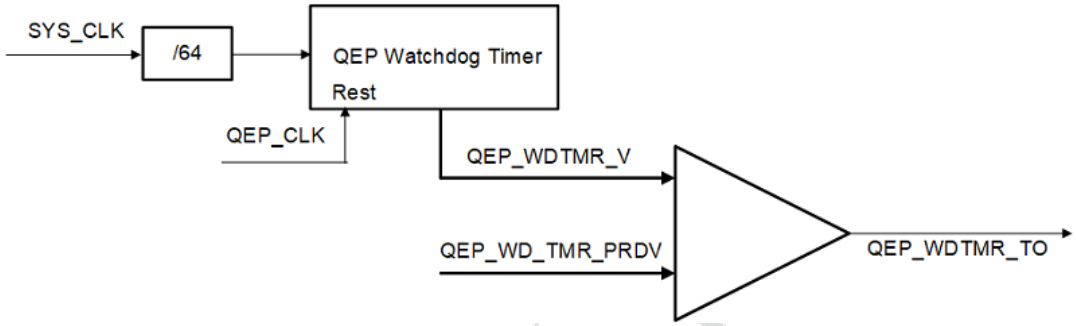
9.1.4.3.4. QEP Timer

QEP Timer 的功能框图如下所示，当寄存器 V 的数值大于 QEP_TMR_PRD_V 时，产生 TO 超时信号。



9.1.4.3.5. QEP Watchdog Timer

QEP Watchdog Timer 的功能框图如下所示，QEP_CLK 信号用于对看门狗定时器的复位，当无有效的 CLK 信号出现时，看门狗计时器一直计数，当 V 的数值大于 QEP_WDTMR_PRD_V 时，产生 TO 超时信号。



9.1.4.3.6. 中断

QEP 模块支持产生中断，通过寄存器可以使能以下的信号经过 OR 逻辑得出 QEP 模块的中断信号 QEP_INT：

- QEP_TMRTO：指示 QEP 计数器超时的信号
- QEP_ICAP：指示发生 Index 相关事件引起的捕获位置计数器的信号
- QEP_SCAP：指示发生 Strobe 相关事件引起的捕获位置计数器的信号
- QEP_POS_CMP：指示产生了位置比较信号
- QEP_POS_CMP_RDY：指示位置比较寄存器 QEP_POS_CNT_CMPV 的数值需要更新
- QEP_POS_CNT_OVFL：指示位置计数器发生上溢出的信号
- QEP_POS_CNT_UDFL：指示位置计数器发生下溢出的信号
- QEP_WDTO：指示看门狗计时器发生超时的信号
- QEP_DIR_CHG：指示 QEP 的正交解析的方向信号发生了改变
- QEP_QUADPH_ERR：指示 QEP 的正交解析的相位信号发生错误
- QEP_POS_CNT_ERR：指示 QEP 的位置计数器计数出错的信号
- QEP_CW_CCW_ERR_INT：指示 QEP 输入的 CW 和 CCW 信号出现异常

9.1.4.3.7. CVRA/ B 触发位置锁存

通过寄存器 QEP_CVRA/B_LATCH_EN 的配置，EPWM 模块产生的 CVRA、CVRB 可以用于触发 QEP 的位置数据的锁存，即在 CVRA 或 CVRB 的有效边沿时刻，QEP 的位置数据进行一次锁存，数据锁存在寄存器 QEP_CVRA_LATCH 和 QEP_CVRB_LATCH。

9.1.4.4. 全局控制

9.1.4.4.1. 全局时钟控制

PWMCS 控制系统的各个模块均具有对时钟源 SYS_CLK 的使能控制位，详细见寄存器 GLB_EPWM_CLK_CTL、GLB_CAP_CLK_CTL、GLB_QEP_CLK_CTL。

9.1.4.4.2. 全局中断

PWMCS 控制系统包括以下的全局中断：

- EPWM 全局中断状态

  通过寄存器 GLB_EPWM_INT_STS 可以查看各个 EPWM 对应的中断信号 EPWM0/1/.../11_INT 的状态。

- EPWM 全局故障中断状态

  通过寄存器 GLB_EPWM_FLT_INT_STS 可以查看各个 EPWM 模块对应的中断信号 EPWM0/1/.../11_FLT_INT 的状态。

- CAP 全局中断状态

  通过寄存器 GLB_CAP_INT_STS 可以查看各个 CAP 模块对应的中断信号 CAP0/1/.../5_INT 的状态。

- QEP 全局故障中断状态

  通过寄存器 GLB_QEP_INT_STS 可以查看各个 QEP 模块对应的中断信号 QEP0/1/.../9_INT 的状态。

9.1.5. 寄存器列表

9.1.5.1. 地址映射

模块名	基地址 (PWMCS_BAS 参考地址映射章节)	地址空间
EPWM 模块		
EPWM0	P0(PWMCS_BASE + 0x000)	256B
EPWM1	P1(PWMCS_BASE + 0x100)	256B
EPWM2	P2(PWMCS_BASE + 0x200)	256B
EPWM3	P3(PWMCS_BASE + 0x300)	256B
EPWM4	P4(PWMCS_BASE + 0x400)	256B
EPWM5	P5(PWMCS_BASE + 0x500)	256B
EPWM7	P7(PWMCS_BASE + 0x700)	256B
EPWM8	P8(PWMCS_BASE + 0x800)	256B
EPWM9	P9(PWMCS_BASE + 0x900)	256B
EPWM10	P10(PWMCS_BASE + 0xA00)	256B
EPWM11	P11(PWMCS_BASE + 0xB00)	256B

模块名	基地址 (PWMCS_BAS 参考地址映射章节)	地址空间
CAP 模块		
CAP0	C0(PWMCS_BASE+0x1000 + 0x000)	256B
CAP1	C1(PWMCS_BASE+0x1000 + 0x100)	256B
CAP2	C2(PWMCS_BASE+0x1000 + 0x200)	256B
CAP3	C3(PWMCS_BASE+0x1000 + 0x300)	256B
CAP4	C4(PWMCS_BASE+0x1000 + 0x400)	256B
CAP5	C5(PWMCS_BASE+0x1000 + 0x500)	256B
QEP 模块		
QEP0	Q0(PWMCS_BASE+0x2000 + 0x000)	256B
QEP1	Q1(PWMCS_BASE+0x2000 + 0x100)	256B
QEP2	Q2(PWMCS_BASE+0x2000 + 0x200)	256B
QEP3	Q3(PWMCS_BASE+0x2000 + 0x300)	256B
QEP4	Q4(PWMCS_BASE+0x2000 + 0x400)	256B
QEP5	Q5(PWMCS_BASE+0x2000 + 0x500)	256B
QEP6	Q6(PWMCS_BASE+0x2000 + 0x600)	256B
QEP7	Q7(PWMCS_BASE+0x2000 + 0x700)	256B
QEP8	Q8(PWMCS_BASE+0x2000 + 0x800)	256B
QEP9	Q9(PWMCS_BASE+0x2000 + 0x900)	256B
全局控制		
GLB_CTL	GB (PWMCS_BASE+0xF000)	4 KB

9.1.5.2. EPWM 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	EPWM_CNT_PRDV	EPWM Counter 计数周期	0x0000 EPWM_CNT_PRDV
0x0004	EPWM_CNT_PHV	EPWM Counter 计数同步相位	0x0004 EPWM_CNT_PHV
0x0008	EPWM_CNT_V	EPWM Counter 计数数值	0x0008 EPWM_CNT_V
0x000C	EPWM_CNT_CONF	EPWM Counter 配置寄存器	0x000C EPWM_CNT_CONF
0x0010	EPWM_CNT_STS	EPWM 计数状态寄存器	0x0010 EPWM_CNT_STS
0x0014	EPWM_CNT_AV	EPWM Counter 比较值 A	0x0014 EPWM_CNT_AV
0x0018	EPWM_CNT_BV	EPWM Counter 比较值 B	0x0018 EPWM_CNT_BV
0x001C	EPWM_UPDT_MOD	EPWM 更新模式寄存器	0x001C EPWM_UPDT_MOD
0x0020	EPWMA_ACT	EPWMA 动作寄存器	0x0020 EPWMA_ACT
0x0024	EPWMB_ACT	EPWMB 动作寄存器	0x0024 EPWMB_ACT
0x0028	EPWM_SW_ACT	EPWM 软件控制动作寄存器	0x0028 EPWM_SW_ACT
0x002C	EPWM_ACT_SW_CT	EPWM 软件持续控制动作寄存器	0x002C EPWM_ACT_SW_CT
0x0030	EPWM_OUT_CONF	EPWM 输出配置寄存器	0x0030 EPWM_OUT_CONF
0x0034	EPWM_RE_DZCTL	EPWM 上升沿死区控制寄存器	0x0034 EPWM_RE_DZCTL
0x0038	EPWM_FE_DZCTL	EPWM 下降沿死区控制寄存器	0x0038 EPWM_FE_DZCTL
0x003C	EPWM_CHOP_CONF	EPWM Chopper 控制寄存器	0x003C EPWM_CHOP_CONF
0x0040	EPWM_FLT_INPUT	EPWM 保护输入控制寄存器	0x0040 EPWM_FLT_INPUT
0x0044	EPWM_FLT_PRTCT	EPWM 保护控制寄存器	0x0044 EPWM_FLT_PRTCT
0x0048	EPWM_FLT_INT_EN	EPWM 故障中断控制寄存器	0x0048 EPWM_FLT_INT_EN

偏移地址	缩写	寄存器描述	章节
0x004C	EPWM_FLT_INT_FLG	EPWM 故障中断标记寄存器	0x004C EPWM_FLT_INT_FLG
0x0050	EPWM_FLT_INT_CLR	EPWM 故障中断标记清除寄存器	0x0050 EPWM_FLT_INT_CLR
0x0054	EPWM_SWFRC_FLT_INT	软件强制 EPWM 故障中断产生寄存器	0x0054 EPWM_SWFRC_FLT_INT
0x0058	EPWM_ADC_INT_CTL	ADC 采样触发以及中断控制寄存器	0x0058 EPWM_ADC_INT_CTL
0x005C	EPWM_ADC_INT_PRE	ADC 采样触发以及中断事件频次控制寄存器	0x005C EPWM_ADC_INT_PRE
0x0060	EPWM_EVNT_FLAG	EPWM 事件标记寄存器	0x0060 EPWM_EVNT_FLAG
0x0064	EPWM_EVNT_CLR	EPWM 事件标记清除寄存器	0x0064 EPWM_EVNT_CLR
0x0068	EPWM_SW_FRC_EVNT	软件强制 EPWM 事件产生寄存器	0x0068 EPWM_SW_FRC_EVNT
0x006C	EPWM_FLT_INCTL2	EPWM 内部触发故障保护寄存器	0x006C EPWM_FLT_INCTL2
0x0080	HRPWM_EN	高精度 PWM 使能寄存器	0x0080 HRPWM_EN
0x0084	HPWM_AV	HPWM 计数器比较 A 值寄存器	0x0084 HPWM_AV
0x0088	HPWM_BV	HPWM 计数器比较 B 值寄存器	0x0088 HPWM_BV
0x008C	HPWM_PRD_SET	HPWM 周期配置寄存器	0x008C HPWM_PRD_SET
0x0090	EPWM_CNT_CV	EPWM Counter 比较值 C	0x0090 EPWM_CNT_CV
0x0090	EPWM_CNT_DV	EPWM Counter 比较值 D	0x0094 EPWM_CNT_DV
0x0098	EPWM_CVRCD_CTL	CVRCD 信号控制寄存器	0x0098 EPWM_CVRCD_CTL
0x009C	EPWM_PSHPLL	EPWM Push-Pull 控制寄存器	0x009C EPWM_PSHPLL

偏移地址	缩写	寄存器描述	章节
0x00A0	EPWM_PUL_OUT_CFG	EPWM 脉冲输出配置寄存器	0x00A0 EPWM_PUL_OUT_CFG
0x00A4	EPWM_PUL_OUT_NUM	EPWM 脉冲输出数量寄存器	0x00A4 EPWM_PUL_OUT_NUM
0x00A8	DMA_CFG	DMA_CFG 寄存器	0x00A8 DMA_CFG
0x00B0	EPWM_NRCFLT_INCTL	EPWM 内部触发不可恢复故障保护寄存器	0x00B0 EPWM_NRCFLT_INCTL
0x00B4	EPWM_RCFLT_INCTL	EPWM 内部触发可恢复故障保护寄存器	0x00B4 EPWM_RCFLT_INCTL
0x00B8	EPWM_SC_SYNC_CTL	SDFM 以及 CPM 的同步控制寄存器	0x00B8 EPWM_SC_SYNC_CTL
0x00BC	EPWM_SC_SYNC_DIV	SDFM 以及 CPM 的同步分频寄存器	0x00BC EPWM_SC_SYNC_DIV
0x00C0	EPWM_ETRG0_SET	EPWM_ETRG0 配置寄存器	0x00C0 EPWM_ETRG0_SET
0x00C4	EPWM_ETRG1_SET	EPWM_ETRG1 配置寄存器	0x00C4 EPWM_ETRG1_SET
0x00C8	EPWM_ETRG_ACC	EPWM_ETRG 累加配置寄存器	0x00C8 EPWM_ETRG_ACC
0x00CC	EPWM_GLB_CTL	EPWM 全局控制寄存器	0x00CC EPWM_GLB_CTL
0x00D0	EPWM_GLB_CTL2	EPWM 全局控制寄存器 2	0x00D0 EPWM_GLB_CTL2
0x00D4	EPWM_GLB_CFG	EPWM 全局控制配置寄存器	0x00D4 EPWM_GLB_CFG
0x00D8	EPWM_LINK_CTL	EPWM 链接控制配置寄存器	0x00D8 EPWM_LINK_CTL
0x00DC	EPWM_ETRG_DLY	EPWM ETRG 延迟配置寄存器	0x00DC EPWM_ETRG_DLY
0x00E0	EPWM_DMA_ENTRY	EPWM DMA 输入入口	0x00E0 EPWM_DMA_ENTRY
0x00E4	EPWM_DMA_MAP0	EPWM DMA 映射配置寄存器 0	0x00E4 EPWM_DMA_MAP0

偏移地址	缩写	寄存器描述	章节
0x00E8	EPWM_DMA_MAP1	EPWM DMA 映射配置寄存器 1	0x00E8 EPWM_DMA_MAP1
0x00F0	EPWM_TRG_OUT	EPWM 触发输出寄存器	0x00F0 EPWM_TRG_OUT
0x0FC	EPWM_VER	EPWM 子模块版本寄存器	0x0FC EPWM_VER

9.1.5.3. CAP 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	CAP_CNT_V	CAP Counter 计数数值	0x0000 CAP_CNT_V
0x0004	CAP_CNT_PH	CAP Counter 计数同步相位	0x0004 CAP_CNT_PH
0x0008	CAP_REG0	CAP_REG0 寄存器	0x0008 CAP_REG0
0x000C	CAP_REG1	CAP_REG1 寄存器	0x000C CAP_REG1
0x0010	CAP_REG2	CAP_REG2 寄存器	0x0010 CAP_REG2
0x0014	CAP_REG3	CAP_REG3 寄存器	0x0014 CAP_REG3
0x0018	CAP_CONF1	CAP 配置寄存器 1	0x0018 CAP_CONF1
0x001C	CAP_CONF2	CAP 配置寄存器 2	0x001C CAP_CONF2
0x0020	CAP_INT_EN	CAP 中断使能寄存器	0x0020 CAP_INT_EN
0x0024	CAP_FLG	CAP 标记寄存器	0x0024 CAP_FLG
0x0028	CAP_FLG_CLR	CAP 标记清除寄存器	0x0028 CAP_FLG_CLR
0x002C	CAP_SW_FRC	软件强制 CAP 事件产生寄存器	0x002C CAP_SW_FRC
0x0030	CAP_IN_FLT	CAP 输入滤波寄存器	0x0030 CAP_IN_FLT
0x0034	CAP_IN_SRC	CAP 输入选择寄存器	0x0034 CAP_IN_SRC
0x0038	DMA_CFG	DMA 配置寄存器	0x0038 DMA_CFG
0x00FC	CAP_VER	CAP 子模块版本寄存器	0x00FC CAP_VER

9.1.5.4. QEP 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	QEP_POS_CNT_V	QEP Position Counter 计数值	0x0000 QEP_POS_CNT_V
0x0004	QEP_POS_CNT_SP	QEP Position Counter 起点值	0x0004 QEP_POS_CNT_SP
0x0008	QEP_POS_CNT_EP	QEP Position Counter 终点值	0x0008 QEP_POS_CNT_EP
0x000C	QEP_POS_CNT_CMPV	QEP Position Counter 比较值	0x000C QEP_POS_CNT_CMPV
0x0010	QEP_POS_CNT_ICAPV	QEP Position Counter Index 捕获值	0x0010 QEP_POS_CNT_ICAPV
0x0014	QEP_POS_CNT_SCAPV	QEP Position Counter Strobe 捕获值	0x0014 QEP_POS_CNT_SCAPV
0x0018	QEP_POS_CNT_TOCAPV	QEP Position Counter Timout 捕获值	0x0018 QEP_POS_CNT_TOCAPV
0x001C	QEP_TMR_V	QEP Timer 寄存器	0x001C QEP_TMR_V
0x0020	QEP_TMR_PRD_V	QEP Timer Period 寄存器	0x0020 QEP_TMR_PRD_V
0x0024	QEP_WD_TMRV	QEP Watchdog 定时器寄存器	0x0024 QEP_WD_TMRV
0x0028	QEP_WD_PRDV	QEP Watchdog 定时器周期寄存器	0x0028 QEP_WD_PRDV
0x002C	QEP_DEC_CONF	QEP Decoder 配置寄存器	0x002C QEP_DEC_CONF
0x0030	QEP_POS_CNT_CONF	QEP 位置计数器配置寄存器	0x0030 QEP_POS_CNT_CONF
0x0034	QEP_CAP_TMR_CONF	QEP Capture Timer 配置寄存器	0x0034 QEP_CAP_TMR_CONF
0x0038	QEP_POS_CNTCMP_CONF	QEP 位置计数器比较寄存器	0x0038 QEP_POS_CNTCMP_CONF
0x003C	QEP_INT_EN	QEP Interrupt 使能寄存器	0x003C QEP_INT_EN

偏移地址	缩写	寄存器描述	章节
0x0040	QEP_INT_FLG	QEP Interrupt 标志寄存器	0x0040 QEP_INT_FLG
0x0044	QEP_INT_CLR	QEP 中断清除寄存器	0x0044 QEP_INT_CLR
0x0048	QEP_SW_FRC_INT	软件强制 QEP 中断产生寄存器	0x0048 QEP_SW_FRC_INT
0x004C	QEP_STS	QEP 状态寄存器	0x004C QEP_STS
0x0050	QEP_CAP_TMRV	QEP Capture 定时器寄存器	0x0050 QEP_CAP_TMRV
0x0054	QEP_CAP_DELT	QEP Capture Delta T 寄存器	0x0054 QEP_CAP_DELT
0x0058	QEP_CAP_TMR_LH	QEP Capture 定时器锁存寄存器	0x0058 QEP_CAP_TMR_LH
0x005C	QEP_CAP_DELT_LH	QEP Capture Delta T 锁存寄存器	0x005C QEP_CAP_DELT_LH
0x0060	QEP_IN_FLT1	QEP 输入滤波寄存器 1	0x0060 QEP_IN_FLT1
0x0064	QEP_IN_FLT2	QEP 输入滤波寄存器 2	0x0064 QEP_IN_FLT2
0x0068	QEP_IN_CTL	QEP 输入控制	0x0068 QEP_IN_CTL
0x0084	QEP_CVRA	QEP CVRA 锁存位置寄存器	0x0084 QEP_CVRA
0x0088	QEP_CVRB	QEP CVRB 锁存位置寄存器	0x0088 QEP_CVRB
0x008C	QEP_INSRC	QEP 输入信号选择寄存器	0x008C QEP_INSRC
0x00FC	QEP_VER	QEP 子模块版本寄存器	0x00FC QEP_VER

9.1.5.5. Global 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0004	GLB_EPWM_INT_STS	Global EPWM 中断状态寄存器	0x0004 GLB_EPWM_INT_STS
0x0008	GLB_FLT_INT_STS	Global EPWM Fault Interrupt 状态寄存器	0x0008 GLB_FLT_INT_STS

偏移地址	缩写	寄存器描述	章节
0x000C	GLB_CAP_INT_STS	Global CAP Interrupt 状态寄存器	0x000C GLB_CAP_INT_STS
0x0010	GLB_QEP_INT_STS	Global QEP Interrupt 状态寄存器	0x0010 GLB_QEP_INT_STS
0x0014	GLB_EPWM_EN	Global EPWM 使能寄存器	0x0014 GLB_EPWM_EN
0x0018	GLB_EPWM_SI0_FLT	Global EPWM_SI0 输入滤波寄存器	0x0018 GLB_EPWM_SI0_FLT
0x001C	GLB_EPWM_SI1_FLT	Global EPWM_SI1 输入滤波寄存器	0x001C GLB_EPWM_SI1_FLT
0x0020	GLB_EPWM_CLK_CTL	Global EPWM 时钟控制寄存器	0x0020 GLB_EPWM_CLK_CTL
0x0024	GLB_CAP_CLK_CTL	Global CAP 时钟控制寄存器	0x0024 GLB_CAP_CLK_CTL
0x0028	GLB_QEP_CLK_CTL	Global QEP 时钟控制寄存器	0x0028 GLB_QEP_CLK_CTL
0x0050	GLB_EPWM_SYNC_IN_SEL	Global EPWM 同步输入选择寄存器	0x0050 GLB_EPWM_SYNC_IN_SEL
0x0054	SYNC_SRC_IN_SEL	同步信号输入源选择寄存器	0x0054 SYNC_SRC_IN_SEL
0x0058	EPWM_FLT_SRC_IN_SEL	故障输入源选择寄存器	0x0058 EPWM_FLT_SRC_IN_SEL
0x0060	EPWM_FALT_IN_FLT1	EPWM 故障输入滤波寄存器 1	0x0060 EPWM_FALT_IN_FLT1
0x0064	EPWM_FALT_IN_FLT2	EPWM 故障输入滤波寄存器 2	0x0064 EPWM_FALT_IN_FLT2
0x0068	EPWM_FALT_IN_FLT3	EPWM 故障输入滤波寄存器 3	0x0068 EPWM_FALT_IN_FLT3
0x0080	GLB_DLL_CFG	Global DLL 配置寄存器	0x0080 GLB_DLL_CFG
0x0088	GLB_FAULT_IN_STS	FAULT IN 状态寄存器	0x0088 GLB_FAULT_IN_STS

9.1.6. EPWM 模块寄存器描述

9.1.6.1. 0x0000 EPWM_CNT_PRDV

默认值：0x00000000			EPWM Counter 计数周期
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_PRDV EPWM Counter Period Value，EPWM 计数器的周期值。此寄存器的更新方式，支持立即更新以及影子更新的模式。

9.1.6.2. 0x0004 EPWM_CNT_PHV

默认值：0x00000000			EPWM Counter 计数同步相位
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_PHV EPWM Counter Phase Value，EPWM 计数器的相位值。当出现同步信号时，此时 EPWM 计数器进行同步，加载此寄存器的相位值进行计数。

9.1.6.3. 0x0008 EPWM_CNT_V

默认值：0x00000000			EPWM Counter 计数数值
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_V EPWM Counter Value，EPWM 计数器的计数值。对此字段进行读操作，可以读取此时 EPWM 计数器。对此字段进行写操作，可以设置 EPWM 计数器的计数值，EPWM 计数器立马更新此计数值，计数方式维持原有的设置进行。

9.1.6.4. 0x000C EPWM_CNT_CONF

默认值：0x00000083			EPWM Counter 配置寄存器
位域	类型	默认值	描述
31:30	R/W	0x0	EPWM_EM_CTL EPWM Emulation Control，调试模式控制 • 0x0：EPWM 内部计数器在调试断点停止计数 • 0x1：EPWM 内部计数器在调试断点时继续计数，直至计数完成当店周期 ◦ 递增模式，在计数器计数到 PRD 停止 ◦ 递减模式，在计数器计数到 0 停止 ◦ 先递增后递减模式，在计数器计数到 0 停止 • 0x2/0x3：EPWM 内部计数器不停止计数
29	–	–	–
28	R/W	0x0	EPWM_TRG1_RST_EN EPWM_TRG1 Reset Enable，使能 EPWM_TRG1 作为复位计数器的触发信号。 • 0x0：未使能 • 0x1：使能
27	R/W	0x0	EPWM_TRG0_RST_EN EPWM_TRG0 Reset Enable，使能 EPWM_TRG0 作为复位计数器的触发信号。 • 0x0：未使能 • 0x1：使能
26	W1AC	0x0	EPWM_CNT_RST EPWM Counter Reset，EPWM 计数器复位，对此位写 1 进行 EPWM 计数器的复位，写 1 后自动清 0。
25	R/W	0x0	EPWM_CNT_RST_RTG EPWM Counter Reset Retrigger Enable，EPWM 计数器的可重触发复位使能，此位只有在单次计数的模式下有效。 • 0x0：未使能，不可重触发复位 • 0x1：使能，可重触发复位
24	R/W	0x0	EPWM_CNT_SIG_EN EPWM Counter Single Mode Enable，EPWM 计数器的单次计数模式使能。 • 0x0：未使能，计数器为连续计数模式 • 0x1：使能，计数器为单次计数模式

默认值：0x00000083			EPWM Counter 配置寄存器
位域	类型	默认值	描述
23	–	–	–
22	R/W	0x0	PAUSE_INI_SRC_EN 初始值控制范围 • 0x0：停止计数时，初始值只控制输出 • 0x1：停止计数时，初始值控制全通路
21	R/W	0x0	PAUSE_INI_OUT_EN 停止计数插入初始值 • 0x0：未使能 • 0x1：使能，初始值控制最后的输出
20	R/W	0x0	EPWM _CMPB_IMD_UPDT_PULSE_EN EPWM COMPB Imediate Update Pulse Enable，CMPB 值在立即更新时是否补脉冲使能。此功能用于希望在 CMPB 值更新的时候，避免出现由于错过原来的 CMPB 值脉冲而导致的输出全周期为 1 或 0。 注： 此位需要在 CMPB 采用立即更新的方式才有效，即 EPWM_CNT_CMPB_UPDT = 0x1。
19	R/W	0x0	EPWM _CMPA_IMD_UPDT_PULSE_EN EPWM COMPA Imediate Update Pulse Enable，CMPA 值在立即更新时是否补脉冲使能。此功能用于希望在 CMPA 值更新的时候，避免出现由于错过原来的 CMPA 值脉冲而导致的输出全周期为 1 或 0。 注： 此位需要在 CMPA 采用立即更新的方式才有效，即 EPWM_CNT_CMPA_UPDT = 0x1。
18	R/W	0x0	EPWM_CMPB_SI_UPDT_PULSE_EN EPWM COMPB Syncin Update Pulse Enable，CMPB 值在 Syncin 更新时是否补脉冲使能。此功能用于希望在 CMPB 值更新的时候，避免出现由于错过原来的 CMPB 值脉冲而导致的输出全周期为 1 或 0。 注意，此位需要在 EPWM _CMPB_LP = 0x5 的时候才有效。
17	R/W	0x0	EPWM_CMPA_SI_UPDT_PULSE_EN EPWM COMPA Syncin Update Pulse Enable，CMPA 值在 Syncin 更新时是否补脉冲使能。此功能用于希望在 CMPA 值更新的时候，避免出现由于错过原来的 CMPA 值脉冲而导致的输出全周期为 1 或 0。

默认值：0x00000083			EPWM Counter 配置寄存器
位域	类型	默认值	描述
			注： 此位需要在 EPWM _CMPA_LP = 0x5 的时候才有效。
16:14	R/W	0x0	EPWM_SYNC_OUT EPWM Synchronization Output，EPWM 同步信号输出。 • 0x0：选择 EPWM_SYNC_IN 信号或软件产生的同步信号 • 0x1：EPWM_CNT_Z • 0x2：EPWM_CNT_CMPB • 0x3：EPWM_CNT_CMPA • 0x4：禁止产生同步信号 • others：Reserved
13	R/W	0x0	EPWM_CNT_PH_DIR EPWM Counter Phase Direction，EPWM 计数器同步相位后的计数方向，此位只有在先递增后递减计数模式下以及 EPWM_CNT_PH_EN = 0x1 才有效。 • 0x0：递减计数 • 0x1：递增计数
12:10	R/W	0x0	EPWM_CLK_DIV2 EPWM Clock Divisor2，EPWM 计数器的时钟除数 2，EPWM_CNT_CLK 由 SYS_CLK/ (Divisor1 * Divisor2)。 • 0x0：1 • 0x1：2 • 0x2：4 • 0x3：8 • 0x4：16 • 0x5：32 • 0x6：64 • 0x7：128
9:7	R/W	0x1	EPWM_CLK_DIV1 EPWM Clock Divisor1，EPWM 计数器的时钟除数 1。 • 0x0：1 • 0x1：2 • 0x2：4

默认值：0x00000083			EPWM Counter 配置寄存器
位域	类型	默认值	描述
			• 0x3：6 • 0x4：8 • 0x5：10 • 0x6：12 • 0x7：14
6	R/W	0x0	EPWM_SW_FRC_SYNC EPWM Software Force Synchronization，EPWM 软件强制产生同步。软件配置此位可以产生同步脉冲信号。 • 0x0：无效果 • 0x1：产生同步脉冲信号
5:4	R/W	0x0	EPWM_PRD_LD EPWM Period Shadow Load，EPWM 计数器的周期寄存器的影子更新信号。 • 0x0：CNT_Z(即在计数器计数复位至 0 时)，PWM 计数器的周期寄存器从影子寄存器进行更新 • 0x1： PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x2：IN_SYNC，外部的同步信号 • 0x3：CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新
3	R/W	0x0	EPWM_CNT_PRD _UPDT EPWM Counter Period Update Mode，EPWM 计数器的周期寄存器的更新模式。 • 0x0：影子寄存器更新模式，在计数器计数复位至 0 时，EPWM 计数器的周期寄存器从影子寄存器进行更新 • 0x1：直接更新模式，EPWM 计数器的周期寄存器直接更新。
2	R/W	0x0	EPWM_CNT_PH_EN EPWM Counter Phase Enable，EPWM 计数器相位使能。 • 0x0：未使能，EPWM 关闭同步功能。 • 0x1：使能，EPWM 打开同步功能。
1:0	R/W	0x3	EPWM_CNT_MOD EPWM Counter Mode，EPWM 计数器模式。 • 0x0：递增计数 • 0x1：递减计数

默认值：0x00000083			EPWM Counter 配置寄存器
位域	类型	默认值	描述
			• 0x2：先递增后递减计数 • 0x3：停止计数

9.1.6.5. 0x0010 EPWM_CNT_STS

默认值：0x00000000			EPWM 计数状态寄存器
位域	类型	默认值	描述
31:3	–	–	–
2	R	0x0	EPWM_CNT_OF_STS EPWM Counter Over Flow Status，EPWM 计数器上溢出状态。 • 0x0：无溢出 • 0x1：计数溢出，计数达到最大值 0xFFFF
1	R	0x0	EPWM_SYNC_IN_STS EPWM Synchronous Input Status，EPWM 同步输入状态。 • 0x0：无同步信号 • 0x1：指示接收到同步输入信号
0	R	0x0	EPWM_CNT_DIR_STS EPWM Counter Direction Status，EPWM 计数器的方向状态。 • 0x0：递减计数 • 0x1：递增计数

9.1.6.6. 0x0014 EPWM_CNT_AV

默认值：0x00000000			EPWM Counter 比较值 A
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_AV EPWM Counter A Value，EPWM 计数器比较值 A。当计数器达到此数值时可以产生脉冲信号，若是递增计数达到此值，产生 EPWM_CNT_U_AV 信号。若是递减计数到达此值，产生 EPWM_CNT_D_AV 信号。

9.1.6.7. 0x0018 EPWM_CNT_BV

默认值：0x00000000			EPWM Counter 比较值 B
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_BV EPWM Counter B Value，EPWM 计数器比较值 B。当计数器达到此数值时可以产生脉冲信号，若是递增计数达到此值，产生 EPWM _CNT_U_BV 信号。若是递减计数到达此值，产生 EPWM_CNT_D_BV 信号。

9.1.6.8. 0x001C EPWM_UPDT_MOD

默认值：0x00000000			EPWM 更新模式寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R	0x0	EPWM_CMPB_SHD_STS EPWM Compare B Shadow Value Status，EPWM 计数比较值 B 的影子寄存器状态。 • 0x0：影子寄存器为空 • 0x1：影子寄存器有待更新的数值，此时对 CMPB 寄存器写新的数值，会覆盖待更新的数值
8	R	0x0	EPWM_CMPA_SHD_STS EPWM Compare A Shadow Value Status，EPWM 计数比较值 A 的影子寄存器状态。 • 0x0：影子寄存器为空 • 0x1：影子寄存器有待更新的数值，此时对 CMPA 寄存器写新的数值，会覆盖待更新的数值
7	R/W	0x0	EPWM_CNT_CMPB_UPDT EPWM Counter Compare Value B Update，EPWM 计数器的比较 B 值的更新模式。 • 0x0：影子寄存器更新模式，EPWM_CNT_BV 的值需要在加载点有效，加载点通过 EPWM_CMPB_LP 配置 • 0x1：直接更新模式，EPWM_CNT_BV 的值直接生效
6	R/W	0x0	EPWM_CNT_CMPA_UPDT

默认值：0x00000000			EPWM 更新模式寄存器
位域	类型	默认值	描述
			EPWM Counter Compare Value A Update，EPWM 计数器的比较 A 值的更新模式。 • 0x0：影子寄存器更新模式，EPWM_CNT_AV 的值需要在加载点有效，加载点通过 EPWM_CMPA_LP 配置 • 0x1：直接更新模式，EPWM_CNT_AV 的值直接生效
5:3	R/W	0x0	EPWM_CMPB_LP EPWM Compare B Value Load Point，设置 EPWM_CNT_BV 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
2:0	R/W	0x0	EPWM_CMPA_LP EPWM Compare A Value Load Point，设置 EPWM_CNT_AV 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值

9.1.6.9. 0x0020 EPWMA_ACT

默认值：0x00000000			EPWMA 动作寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	EPWMA_ACT_ETRG1 EPWMA Action of ETRG1，EPWMA 通道在 EPWM_ETRG1 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
21:20	R/W	0x0	EPWMA_ACT_ETRG0 EPWMA Action of ETRG0，EPWMA 通道在 EPWM_ETRG0 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
19:16	–	–	–
15	R/W	0x0	EPWMA_ACT_UPDT EPWMA ACT Update，EPWMA 动作寄存器更新模式。 • 0x0：影子寄存器更新模式 • 0x1：直接更新模式
14:12	R/W	0x0	EPWMA_ACT_LP EPWMA Action of Load Point，EPWMA_ACT 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号

默认值：0x00000000			EPWMA 动作寄存器
位域	类型	默认值	描述
			• 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
11:10	R/W	0x0	EPWMA_ACT_CNTDBV EPWMA Action of CNTDBV，EPWMA 通道在 EPWM_CNT_D_BV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
9:8	R/W	0x0	EPWMA_ACT_CNTUBV EPWMA Action of CNTUBV，EPWMA 通道在 EPWM_CNT_U_BV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
7:6	R/W	0x0	EPWMA_ACT_CNTDAV EPWMA Action of CNTDAV，EPWMA 通道在 EPWM_CNT_D_AV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
5:4	R/W	0x0	EPWMA_ACT_CNTUAV EPWMA Action of CNTUAV，EPWMA 通道在 EPWM_CNT_U_AV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
3:2	R/W	0x0	EPWMA_ACT_CNTPRD

默认值：0x00000000			EPWMA 动作寄存器
位域	类型	默认值	描述
			EPWMA Action of CNTPRD，EPWMA 通道在 EPWM_CNT_PRD 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
1:0	R/W	0x0	EPWMA_ACT_CNTZ EPWMA Action of CNTZ，EPWMA 通道在 EPWM_CNT_Z 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转

9.1.6.10. 0x0024 EPWMB_ACT

默认值：0x00000000			EPWMB 动作寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	EPWMB_ACT_ETRG1 EPWMB Action of ETRG1，EPWMB 通道在 EPWM_ETRG1 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
21:20	R/W	0x0	EPWMB_ACT_ETRG0 EPWMB Action of ETRG0，EPWMB 通道在 EPWM_ETRG0 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转

默认值：0x00000000			EPWMB 动作寄存器
位域	类型	默认值	描述
19:16	–	–	–
15	R/W	0x0	EPWMB_ACT_UPDT EPWMB ACT Update，EPWMB 动作寄存器更新模式。 • 0x0：影子寄存器更新模式 • 0x1：直接更新模式
14:12	R/W	0x0	EPWMB_ACT_LP EPWMB Action of Load Point，EPWMB_ACT 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT (即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
11:10	R/W	0x0	EPWMB_ACT_CNTDBV EPWMB Action of CNTDBV，EPWMB 通道在 EPWM_CNT_D_BV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
9:8	R/W	0x0	EPWMB_ACT_CNTUBV EPWMB Action of CNTUBV，EPWMB 通道在 EPWM_CNT_U_BV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
7:6	R/W	0x0	EPWMB_ACT_CNTDAV

默认值：0x00000000			EPWMB 动作寄存器
位域	类型	默认值	描述
			EPWMB Action of CNTDAV，EPWMB 通道在 EPWM_CNT_D_AV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
5:4	R/W	0x0	EPWMB_ACT_CNTUAV EPWMB Action of CNTUAV，EPWMB 通道在 EPWM_CNT_U_AV 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
3:2	R/W	0x0	EPWMB_ACT_CNTPRD EPWMB Action of CNTPRD，EPWMA 通道在 EPWM_CNT_PRD 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
1:0	R/W	0x0	EPWMB_ACT_CNTZ EPWMB Action of CNTZ，EPWMA 通道在 EPWM_CNT_Z 信号有效时的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转

9.1.6.11. 0x0028 EPWM_SW_ACT

默认值：0x00000000			EPWM 软件控制动作寄存器
位域	类型	默认值	描述
31:9	–	–	–
8:6	R/W	0x0	EPWM_ACT_SW_CT_UPDT Register EPWM_ACT_SW_CT Update mode，寄存器 EPWM_ACT_SW_CT 的更新模式。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：立即更新寄存器值 • 0x4：PUL_UPDT (即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
5	R/WAC	0x0	EPWMB_ACT_SW_OS_EN Enable EPWMB Action of Software Oneshot，使能软件的单次控制，产生脉冲信号，此位写 1 后执行完自动清 0。
4:3	R/W	0x0	EPWMB_ACT_SW_OS EPWMB Action of Software Oneshot，软件单次控制 EPWMB 通道的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平 • 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转
2	R/WAC	0x0	EPWMA_ACT_SW_OS_EN Enable EPWMA Action of Software Oneshot，使能软件的单次控制，产生脉冲信号，此位写 1 后执行完自动清 0。
1:0	R/W	0x0	EPWMA_ACT_SW_OS EPWMA Action of Software Oneshot，软件单次控制 EPWMA 通道的动作。 • 0x0：无动作 • 0x1：执行清除操作，信号为低电平

默认值：0x00000000			EPWM 软件控制动作寄存器
位域	类型	默认值	描述
			• 0x2：执行置位操作，信号为高电平 • 0x3：执行翻转操作，信号进行高低翻转

9.1.6.12. 0x002C EPWM_ACT_SW_CT

默认值：0x80800000			EPWM 软件持续控制动作寄存器
位域	类型	默认值	描述
31:16	R/W	0x8080	维持默认值
15:4	–	–	–
3:2	R/W	0x0	EPWMB_ACT_SW_CT EPWMB Action of Software Oneshot，软件持续控制 EPWMB 通道的动作。 • 0x0：软件无持续动作 • 0x1：软件控制 EPWM 持续执行低电平 • 0x2：软件控制 EPWM 持续执行高电平 • 0x3：软件无持续动作
1:0	R/W	0x0	EPWMA_ACT_SW_CT EPWMA Action of Software Oneshot，软件持续控制 EPWMA 通道的动作。 • 0x0：软件无持续动作 • 0x1：软件控制 EPWM 持续执行低电平 • 0x2：软件控制 EPWM 持续执行高电平 • 0x3：软件无持续动作

9.1.6.13. 0x0030 EPWM_OUT_CONF

默认值：0x00000000			EPWM 输出配置寄存器
位域	类型	默认值	描述
31: 19	–	–	–
18:17	R/W	0x0	EPWM_DB_CTL_LP EPWM Deadband Control Load Point，EPWM 死区控制寄存器加载点，此域在 EPWM_DB_CTL_UPDT = 0x1 生效。

默认值：0x00000000			EPWM 输出配置寄存器
位域	类型	默认值	描述
			• 0x0：在 CNT = 0 的时刻加载 • 0x1：在 CNT = PRD 的时刻加载 • 0x2：在 CNT = 0 或 CNT = PRD 的时刻加载 • 0x3：不更新
16	R/W	0x0	EPWM_DB_CTL_UPDT EPWM Deadband Control Update，EPWM 死区控制寄存器的更新模式，包括 EPWM_BP_SET[1:0]、EPWM_POL_SET[1:0]、EPWM_CH_SEL[1:0]、EPWM_SWAP_SET[1:0 和 EPWM_DD_EN 寄存器 • 0x0：立即更新 • 0x1：影子更新
15	–	–	–
14	R/W	0x0	EPWM_DD_EN EPWM Dual Edge Deadband Enable，双边沿死区控制使能。
13:12	R/W	0x0	EPWM_SWAP_SEL EPWM Swap Setting，输出通道交换，具体配置说明见 EPWM 输出模式的描述。
11	R/W	0x0	EPWM_FE_UPDT EPWM Falling Edge Update，下降沿死区时间 EPWM_FE_DZCTL 更新模式。 • 0x0：立即更新 • 0x1：影子更新
10	R/W	0x0	EPWM_RE_UPDT EPWM Rising Edge Update，上升沿死区时间 EPWM_RE_DZCTL 更新模式。 • 0x0：立即更新 • 0x1：影子更新
9:8	R/W	0x0	EPWM_FE_LP EPWM Falling Edge Load Point，下降沿死区时间 EPWM_FE_DZCTL 加载点。 • 0x0：在 CNT = 0 的时刻加载 • 0x1：在 CNT = PRD 的时刻加载 • 0x2：在 CNT = 0 或 CNT = PRD 的时刻加载 • 0x3：不更新
7:6	R/W	0x0	EPWM_RE_LP

默认值：0x00000000			EPWM 输出配置寄存器
位域	类型	默认值	描述
			EPWM Rising Edge Load Point，上升沿死区时间 EPWM_RE_DZCTL 加载点。 • 0x0：在 CNT = 0 的时刻加载 • 0x1：在 CNT = PRD 的时刻加载 • 0x2：在 CNT = 0 或 CNT = PRD 的时刻加载 • 0x3：不更新
5:4	R/W	0x0	EPWM_CH_SEL[1:0] EPWM Channel Select，EPWM 通道选择信号。具体的配置说明见 EPWM 输出模式章节的描述。 EPWM_CH_SEL[1:0] = bit[5:4] • bit[4]为 EPWM_A 输出信号源的选择，若 bit[4] = 0，则选择 A 通道信号。相反则选择 B 通道信号 • bit[5]为 EPWM_B 输出信号源的选择，若 bit[5] = 0，则选择 A 通道信号。相反则选择 B 通道信号
3:2	R/W	0x0	EPWM_POL_SET[1:0] EPWM 极性设置信号，具体的配置说明见 EPWM 输出模式章节 的描述。 EPWM_POL_SEL[1:0] = bit[3:2] • bit[2]为 EPWM_A 输出的极性控制，若 bit[2] = 1，则 EPWM_A 输出经过信号反相。 • bit[3]为 EPWM_B 输出的极性控制，若 bit[3] = 1，则 EPWM_B 输出经过信号反相。
1:0	R/W	0x0	EPWM_BP_SET[1:0] EPWM Bypass Set，EPWM 旁通控制信号,具体的配置说明见 EPWM 输出模式章节 的描述。EPWM_BP_SET[1:0] = bit[1:0] • bit[0] 为 EPWM_A 输出的旁通功能选择，若 bit[0] = 0，则 EPWM_A 直接输出 A 通道的信号，旁通了 EPWM_CH_SEL 以及 EPWM _POL_SET 的配置。 • bit[1] 为 EPWM_B 输出的旁通功能选择，若 bit[1] = 0，则 EPWM_B 直接输出 B 通道的信号，旁通了 EPWM_CH_SEL 以及 EPWM _POL_SET 的配置。

9.1.6.14. 0x0034 EPWM_RE_DZCTL

默认值：0x00000000			EPWM 上升沿死区控制寄存器
位域	类型	默认值	描述
31: 15	–	–	–

默认值：0x00000000			EPWM 上升沿死区控制寄存器
位域	类型	默认值	描述
14:0	R/W	0x0	EPWM_RE_DZCTL EPWM Rising Edge Deadzone Control，EPWM 的上升边沿的死区时间控制。 死区延迟的时间为 EPWM_RE_DZCTL * EPWM _CNT_CLK。

9.1.6.15. 0x0038 EPWM_FE_DZCTL

默认值：0x00000000			EPWM 下降沿死区控制寄存器
位域	类型	默认值	描述
31: 15	–	–	–
14:0	R/W	0x0	EPWM_FE_DZCTL EPWM Falling Edge Deadzone Control，EPWM 的下降边沿的死区时间控制。死区延迟的时间为 EPWM _FE_DZCTL * EPWM_CNT_CLK。

9.1.6.16. 0x003C EPWM_CHOP_CONF

默认值：0x00002000			EPWM Chopper 控制寄存器
位域	类型	默认值	描述
31:14	–	–	–
13	R/W	0x1	EPWM_CHOP_ALIGN EPWM Chopper Align Enable，Chopper 对齐使能。 • 0x0：未使能 • 0x1：使能
12	R/W	0x0	EPWM_OSB_EN EPWM Oneshot B Module Enable，EPWM Chopper Oneshot B 模块使能。 • 0x0：未使能 • 0x1：EPWM_FB 进行 Oneshot 调制
11	R/W	0x0	EPWM_OSA_EN EPWM Oneshot BAModule Enable，EPWM Chopper Oneshot A 模块使能。 • 0x0：未使能 • 0x1：EPWM_FA 进行 Oneshot 调制

默认值：0x00002000			EPWM Chopper 控制寄存器
位域	类型	默认值	描述
10:8	R/W	0x0	EPWM_CHOP_DUTY EPWM Chopper Duty，EPWM Chopper 占空比。 • 0x0： 1/8 • 0x1： 2/8 • 0x2： 3/8 • 0x3： 4/8 • 0x4： 5/8 • 0x5： 6/8 • 0x6： 7/8 • 0x7： Reserved
7:5	R/W	0x0	EPWM_CHOP_FRE EPWM Chopper Frequency，EPWM Chopper 频率。 • 0x0： SYS_CLK / 8 • 0x1： SYS_CLK / 16 • 0x2： SYS_CLK / 24 • 0x3： SYS_CLK / 32 • 0x4： SYS_CLK / 40 • 0x5： SYS_CLK / 48 • 0x6： SYS_CLK / 56 • 0x7： SYS_CLK / 64
4:1	R/W	0x0	EPWM_CHOP_OS_WTH EPWM Chopper One-shot Pulse Width，Choopper One-shot 脉宽。 • 0x0：1 x SYS_CLK / 8 • 0x1：2 x SYS_CLK / 8 • 0x2：3 x SYS_CLK / 8 • 0x3：4 x SYS_CLK / 8 • … • 0xE：15 x SYS_CLK / 8 • 0xF：16 x SYS_CLK / 8
0	R/W	0x0	EPWM_CHOP_EN EPWM Chopper Function Enable，EPWM chopper 功能使能。

默认值：0x00002000			EPWM Chopper 控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能，此模块旁通 • 0x1：使能 Chopper 模块

9.1.6.17. 0x0040 EPWM_FLT_INPUT

默认值：0x00000000			EPWM 保护输入控制寄存器
位域	类型	默认值	描述
31:22	–	–	–
21	R/W	0x0	EPWM_FLT5_IN_POL EPWM Fault5 Input Polarity，EPWM 故障输入信号极性。 • 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
20	R/W	0x0	EPWM_FLT4_IN_POL EPWM Fault4 Input Polarity，EPWM 故障输入信号极性。 • 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
19	R/W	0x0	EPWM_FLT3_IN_POL EPWM Fault3 Input Polarity，EPWM 故障输入信号极性。 • 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
18	R/W	0x0	EPWM_FLT2_IN_POL EPWM Fault2 Input Polarity，EPWM 故障输入信号极性。 • 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
17	R/W	0x0	EPWM_FLT1_IN_POL EPWM Fault1 Input Polarity，EPWM 故障输入信号极性。 • 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
16	R/W	0x0	EPWM_FLT0_IN_POL EPWM Fault0 Input Polarity，EPWM 故障输入信号极性。

默认值：0x00000000			EPWM 保护输入控制寄存器
位域	类型	默认值	描述
			• 0x0：低有效，低电平触发保护 • 0x1：高有效，高电平触发保护
15:14	–	–	–
13	R/W	0x0	EPWM_REC_FLT5_EN EPWM Recoverable Fault5 Enable，使能 EPWM 可恢复的故障信号 5。 • 0x0：不使能 • 0x1：使能故障信号 5 作为可恢复故障信号的输入
12	R/W	0x0	EPWM_REC_FLT4_EN EPWM Recoverable Fault4 Enable，使能 EPWM 可恢复的故障信号 4。 • 0x0：不使能 • 0x1：使能故障信号 4 作为可恢复故障信号的输入
11	R/W	0x0	EPWM_REC_FLT3_EN EPWM Recoverable Fault3 Enable，使能 EPWM 可恢复的故障信号 3。 • 0x0：不使能 • 0x1：使能故障信号 3 作为可恢复故障信号的输入
10	R/W	0x0	EPWM_REC_FLT2_EN EPWM Recoverable Fault2 Enable，使能 EPWM 可恢复的故障信号 2。 • 0x0：不使能 • 0x1：使能故障信号 2 作为可恢复故障信号的输入
9	R/W	0x0	EPWM_REC_FLT1_EN EPWM Recoverable Fault1 Enable，使能 EPWM 可恢复的故障信号 1。 • 0x0：不使能 • 0x1：使能故障信号 1 作为可恢复故障信号的输入
8	R/W	0x0	EPWM_REC_FLT0_EN EPWM Recoverable Fault0 Enable，使能 EPWM 可恢复的故障信号 0。 • 0x0：不使能 • 0x1：使能故障信号 0 作为可恢复故障信号的输入
7:6	–	–	–
5	R/W	0x0	EPWM_NREC_FLT5_EN

默认值：0x00000000			EPWM 保护输入控制寄存器
位域	类型	默认值	描述
			EPWM Non-recoverable Fault5 Enable，使能EPWM不可恢复的故障信号 5。 • 0x0：不使能 • 0x1：使能故障信号 5 作为不可恢复故障信号的输入
4	R/W	0x0	EPWM_NREC_FLT4_EN EPWM Non-recoverable Fault4 Enable，使能EPWM不可恢复的故障信号 4。 • 0x0：不使能 • 0x1：使能故障信号 4 作为不可恢复故障信号的输入
3	R/W	0x0	EPWM_NREC_FLT3_EN EPWM Non-recoverable Fault3 Enable，使能EPWM不可恢复的故障信号 3。 • 0x0：不使能 • 0x1：使能故障信号 3 作为不可恢复故障信号的输入
2	R/W	0x0	EPWM_NREC_FLT2_EN EPWM Non-recoverable Fault2 Enable，使能EPWM不可恢复的故障信号 2。 • 0x0：不使能 • 0x1：使能故障信号 2 作为不可恢复故障信号的输入
1	R/W	0x0	EPWM_NREC_FLT1_EN EPWM Non-recoverable Fault1 Enable，使能EPWM不可恢复的故障信号 1。 • 0x0：不使能 • 0x1：使能故障信号 1 作为不可恢复故障信号的输入
0	R/W	0x0	EPWM_NREC_FLT0_EN EPWM Non-recoverable Fault0 Enable，使能EPWM不可恢复的故障信号 0。 • 0x0：不使能 • 0x1：使能故障信号 0 作为不可恢复故障信号的输入

9.1.6.18. 0x0044 EPWM_FLT_PRTCT

默认值：0x00000000			EPWM 保护控制寄存器
位域	类型	默认值	描述
31:24	R/W	0xC8	UNIT_US_THR

默认值：0x00000000			EPWM 保护控制寄存器
位域	类型	默认值	描述
			此处默认 200，对应 PWMCS 模块时钟 200MHz，如果模块时钟不是 200MHz，此处需要根据配置的时钟进行修改，以保证保护延迟基本单位 1us 的准确度。
23:20	–	–	–
19:18	R/W	0x0	EPWM_B_INIT EPWM Output B Init Value，EPWM _B 输出管脚信号的初始值，即在 EPWM 功能未使能时的输出值。 • 0x0：输出低电平 • 0x1：输出高电平 • 0x2/0x3：输出高阻
17:16	R/W	0x0	EPWM_A_INIT EPWM Output A Init Value，EPWM _A 输出管脚信号的初始值，即在 EPWM 功能未使能时的输出值。 • 0x0：输出低电平 • 0x1：输出高电平 • 0x2/0x3：输出高阻
15:12	R/W	0x0	EPWMB_FLT_PRTCT_DLY EPWMB Fault Protect Delay，EPWMB 故障保护延迟。 • 0x0：无延迟，立即保护 • 0x1：延迟 1 us • 0x2：延迟 2 us • ... • 0xF：延迟 15 us
11:8	R/W	0x0	EPWMA_FLT_PRTCT_DLY EPWMA Fault Protect Delay，EPWMA 故障保护延迟。 • 0x0：无延迟，立即保护 • 0x1：延迟 1 us • 0x2：延迟 2 us • ... • 0xF：延迟 15 us
7:4	–	–	–

默认值：0x00000000			EPWM 保护控制寄存器
位域	类型	默认值	描述
3:2	R/W	0x0	EPWMB_FLT_PRTCT EPWMB Fault Protect mode，EPWM B 的故障保护措施。 • 0x0：执行高阻态 • 0x1：强制输出为高 • 0x2：强制输出为低 • 0x3：无操作
1:0	R/W	0x0	EPWMA_FLT_PRTCT EPWMA Fault Protect mode，EPWM A 的故障保护措施。 • 0x0：执行高阻态 • 0x1：强制输出为高 • 0x2：强制输出为低 • 0x3：无操作

9.1.6.19. 0x0048 EPWM_FLT_INT_EN

默认值：0x00000000			EPWM 故障中断控制寄存器
位域	类型	默认值	描述
31:3	–	–	–
2	R/W	0x0	EPWM_NREC_FLT_INT_EN EPWM Non-Recoverable Fault Interrupt Enable，EPWM 不可恢复故障中断使能。 • 0x0：不使能 • 0x1：使能
1	R/W	0x0	EPWM_REC_FLT_INT_EN EPWM Recoverable Fault Interrupt Enable，EPWM可恢复故障中断使能。 • 0x0：不使能 • 0x1：使能
0	–	–	–

9.1.6.20. 0x004C EPWM_FLT_INT_FLG

默认值：0x00000000			EPWM 故障中断标记寄存器
位域	类型	默认值	描述
31:3	–	–	–
2	R	0x0	EPWM_NREC_FLT_FLAG EPWM Non-Recoverable Fault Flag，EPWM不可恢复故障信号标记。 • 0x0：无故障信号产生 • 0x1：产生了不可恢复故障信号信号
1	R	0x0	EPWM_REC_FLT_FLAG EPWM Recoverable Fault Flag，EPWM 可恢复故障信号标记。 • 0x0：无故障信号产生 • 0x1：产生了可恢复故障信号信号
0	R	0x0	EPWM_FLT_INT_FLAG EPWM Fault Interrupt Flag，EPWM 故障中断事件标记。 • 0x0：无中断信号产生 • 0x1：产生中断信号

9.1.6.21. 0x0050 EPWM_FLT_INT_CLR

默认值：0x00000000			EPWM 故障中断标记清除寄存器
位域	类型	默认值	描述
31:3	–	–	–
2	R/WAC	0x0	EPWM_NREC_FLT_CLR EPWM Non-Recoverable Fault Flag Clear，EPWM 不可恢复故障信号标记清除。 • 0x0：无操作 • 0x1：清除不可恢复故障信号，执行完自动清 0
1	R/WAC	0x0	EPWM_REC_FLT_CLR EPWM Recoverable Fault Flag Clear，EPWM 可恢复故障信号标记清除。 • 0x0：无操作 • 0x1：清除可恢复故障信号，执行完自动清 0
0	R/WAC	0x0	EPWM_FLT_CLR

默认值：0x00000000			EPWM 故障中断标记清除寄存器
位域	类型	默认值	描述
			EPWM Fault Interrupt Flag Clear，EPWM 故障中断标记清除。 • 0x0：无操作 • 0x1：清除故障信号，执行完自动清 0

9.1.6.22. 0x0054 EPWM_SWFRC_FLT_INT

默认值：0x00000000			软件强制 EPWM 故障中断产生寄存器
位域	类型	默认值	描述
31:3	–	–	–
2	R/WAC	0x0	EPWM_SW_FRC_NREC_FLT EPWM Software Force Non-Recoverable Fault，软件强制 EPWM 不可恢复故障信号的产生。 • 0x0：无操作 • 0x1：产生不可恢复故障信号
1	R/WAC	0x0	EPWM_SW_FRC_REC_FLT EPWM Software Force Recoverable Fault，软件强制 EPWM 可恢复故障信号的产生。 • 0x0：无操作 • 0x1：产生可恢复故障信号
0	–	–	–

9.1.6.23. 0x0058 EPWM_ADC_INT_CTL

默认值：0x00000000			ADC 采样触发以及中断控制寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:20	R/W	0x0	CVRB_CPM_SRC_SEL Conversion B CPM Source Select，CPM 输出信号作为 CVRB 信号的源选择（这里的信号，需要在 CPM 模块使能比较信号输出至 EPWM 的功能）。Converision B Source Select，CVRB 信号的源选择。

默认值：0x00000000			ADC 采样触发以及中断控制寄存器
位域	类型	默认值	描述
			• 0x0：CPM0_COH • 0x1：CPM0_COL • 0x2：CPM0_COHL • 0x3：CPM1_COH • 0x4：CPM1_COL • 0x5：CPM1_COHL • 0x6：EPWM_CNT_U_CV • 0x7：EPWM_CNT_D_CV • 0x8：EPWM_CNT_U_DV • 0x9：EPWM_CNT_D_DV • 0xA：ETRG0 • 0xB：ETRG1 • others：Reserved
19:16	R/W	0x0	CVRA_CPM_SRC_SEL Conversion A CPM Source Select，CPM 输出信号作为 CVRA 信号的源选择（这里的信号，需要在 CPM 模块使能比较信号输出至 EPWM 的功能）。Converision A Source Select，CVRA 信号的源选择。 • 0x0：CPM0_COH • 0x1：CPM0_COL • 0x2：CPM0_COHL • 0x3：CPM1_COH • 0x4：CPM1_COL • 0x5：CPM1_COHL • 0x6：EPWM_CNT_U_CV • 0x7：EPWM_CNT_D_CV • 0x8：EPWM_CNT_U_DV • 0x9：EPWM_CNT_D_DV • 0xA：ETRG0 • 0xB：ETRG1 • others：Reserved
15	R/W	0x0	EPWM_ADC_CVRB_EN EPWM ADC Conversion B Trigger Enable，CVRB 触发信号使能。

默认值：0x00000000			ADC 采样触发以及中断控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
14:12	R/W	0x0	EPWM_ADC_CVRB_SEL EPWM ADC Conversion B Trigger Selection，EPWM ADC 转换触发 B 信号选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：CPM_SRC（通过 bit[19:16]选择）
11	R/W	0x0	EPWM_ADC_CVRA_EN EPWM ADC Conversion A Trigger Enable，CVRA 触发信号使能。 • 0x0：未使能 • 0x1：使能
10:8	R/W	0x0	EPWM_ADC_CVRA_SEL EPWM ADC Conversion A Trigger Selection，EPWM ADC 转换触发 A 信号选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：CPM_SRC（通过 bit[19:16]选择）
7:5	–	–	–
4	R/W	0x0	EPWM_INT_EN EPWM Interrupt Enable，使能 EPWM 中断信号。

默认值：0x00000000			ADC 采样触发以及中断控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
3:0	R/W	0x0	EPWM_INT_SEL EPWM Interrupt Select，EPWM 中断源选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z 或 EPWM _CNT_PRD • 0x7：PUL_CNT_ZRO（脉冲输出为 0） • 0x8：EPWM_CNT_U_CV • 0x9：EPWM_CNT_D_CV • 0xA：EPWM_CNT_U_DV • 0xB：EPWM_CNT_U_DV • others：Reserved

9.1.6.24. 0x005C EPWM_ADC_INT_PRE

默认值：0x00000000			ADC 采样触发以及中断事件频次控制寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:20	R	0x0	EPWM_ADC_CVRB _CNT EPWM ADC Conversion B Count，ADC 转换触发信号源产生的计数，表示目前已经产生了多少次的源的信号。
19:16	R	0x0	EPWM_ADC_CVRA _CNT EPWM ADC Conversion A Count，ADC 转换触发信号源产生的计数，表示目前已经产生了多少次的源的信号。
15:12	R/W	0x0	EPWM_ADC_CVRB_DIV

默认值：0x00000000			ADC 采样触发以及中断事件频次控制寄存器
位域	类型	默认值	描述
			EPWM ADC Conversion A Divisor，ADC 转换触发信号 B 的信号源（由 EPWM_ADC_CVRB_SEL 选择）的除数，设置此除数可以设定 ADC 转换触发信号 B 的产生频次。 • 0x0：无信号产生 • 0x1：源信号产生一次则对应产生一次 ADC 转换触发信号 • 0x2：源信号产生两次则对应产生一次 ADC 转换触发信号 • 0x3：源信号产生三次则对应产生一次 ADC 转换触发信号 • ... • 0xF：中断源信号产生 15 次则对应产生一次 ADC 转换触发信号
11:8	R/W	0x0	EPWM_ADC_CVRA_DIV EPWM ADC Conversion A Divisor，ADC 转换触发信号 A 的信号源（由 EPWM_ADC_CVRA_SEL 选择）的除数，设置此除数可以设定 ADC 转换触发信号 A 的产生频次。 • 0x0：无信号产生 • 0x1：源信号产生一次则对应产生一次 ADC 转换触发信号 • 0x2：源信号产生两次则对应产生一次 ADC 转换触发信号 • 0x3：源信号产生三次则对应产生一次 ADC 转换触发信号 • ... • 0xF：中断源信号产生 15 次则对应产生一次 ADC 转换触发信号
7:4	R	0x0	EPWM_INT_CNT EPWM Interrupt Count，EPWM 中断信号源产生的计数，表示目前已经产生了多少次的 EPWM 中断源的信号。
3:0	R/W	0x0	EPWM_INT_DIV EPWM Interrupt Divisor，EPWM 中断信号源（由 EPWM_INT_SEL 选择）的除数，设置此除数可以设定 EPWM 中断信号产生频次。 • 0x0：无中断信号产生 • 0x1：中断源信号产生一次则对应产生一次中断 • 0x2：中断源信号产生两次则对应产生一次中断 • 0x3：中断源信号产生三次则对应产生一次中断 • ... • 0xF：中断源信号产生 15 次则对应产生一次中断

9.1.6.25. 0x0060 EPWM_EVNT_FLAG

默认值：0x00000000			EPWM 事件标记寄存器
位域	类型	默认值	描述
31: 6	–	–	–
5	R	0x0	EPWM_CVRD_FLAG EPWM Conversion D Flag，转换触发信号 D 的标记。 • 0x0：无转换信号产生 • 0x1：产生转换信号
4	R	0x0	EPWM_CVRC_FLAG EPWM Conversion C Flag，转换触发信号 C 的标记。 • 0x0：无转换信号产生 • 0x1：产生转换信号
3	R	0x0	EPWM_ADC_CVRB_FLAG EPWM ADC Conversion B Flag，ADC 转换触发信号 B 的标记。 • 0x0：无 ADC 转换信号产生 • 0x1：产生 ADC 转换信号
2	R	0x0	EPWM_ADC_CVRA_FLAG EPWM ADC Conversion A Flag，ADC 转换触发信号 A 的标记。 • 0x0：无 ADC 转换信号产生 • 0x1：产生 ADC 转换信号
1	–	–	–
0	R	0x0	EPWM_INT_FLAG EPWM Interrupt Flag，EPWM 中断事件标记。 • 0x0：无中断信号产生 • 0x1：产生中断信号

9.1.6.26. 0x0064 EPWM_EVNT_CLR

默认值：0x00000000			EPWM 事件标记清除寄存器
位域	类型	默认值	描述
31: 6	–	–	–

默认值：0x00000000			EPWM 事件标记清除寄存器
位域	类型	默认值	描述
5	R/WAC	0x0	EPWM_CVRD_CLR EPWM Conversion D Flag Clear，转换触发信号 D 的标记清除。 • 0x0：无操作 • 0x1：清除转换信号，执行完自动清零
4	R/WAC	0x0	EPWM_CVRC_CLR EPWM Conversion C Flag Clear，转换触发信号 C 的标记清除。 • 0x0：无操作 • 0x1：清除转换信号，执行完自动清零
3	R/WAC	0x0	EPWM_ADC_CVRB_CLR EPWM ADC Conversion B Flag Clear，ADC 转换触发信号 B 的标记清除。 • 0x0：无操作 • 0x1：清除 ADC 转换信号，执行完自动清 0
2	R/WAC	0x0	EPWM_ADC_CVRA_CLR EPWM ADC Conversion A Flag Clear，ADC 转换触发信号 A 的标记清除。 • 0x0：无操作 • 0x1：清除 ADC 转换信号，执行完自动清 0
1	–	–	–
0	R/WAC	0x0	EPWM_INT_CLR EPWM Interrupt Flag Clear，EPWM 中断事件标记清除。 • 0x0：无操作 • 0x1：清除中断信号，执行完自动清 0

9.1.6.27. 0x0068 EPWM_SW_FRC_EVNT

默认值：0x00000000			软件强制 EPWM 事件产生寄存器
位域	类型	默认值	描述
31: 6	–	–	–
5	R/WAC	0x0	EPWM_SW_FRC_CVRD EPWM Software Conversion D，软件强制转换触发信号 D 的产生。

默认值：0x00000000			软件强制 EPWM 事件产生寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：产生转换触发信号
4	R/WAC	0x0	EPWM_SW_FRC_CVRC EPWM Software Conversion C，软件强制转换触发信号 C 的产生。 • 0x0：无操作 • 0x1：产生转换触发信号
3	R/WAC	0x0	EPWM_SW_FRC_ADC_CVRB EPWM Software ADC Conversion B，软件强制 ADC 转换触发信号 B 的产生。 • 0x0：无操作 • 0x1：产生 ADC 转换触发信号
2	R/WAC	0x0	EPWM_SW_FRC_ADC_CVRA EPWM Software ADC Conversion A，软件强制 ADC 转换触发信号 A 的产生。 • 0x0：无操作 • 0x1：产生 ADC 转换触发信号
1	–	–	–
0	R/WAC	0x0	EPWM_SW_FRC_INT EPWM Software Force Interrupt，软件强制 EPWM 中断信号产生。 • 0x0：无操作 • 0x1：产生 EPWM 中断信号

9.1.6.28. 0x006C EPWM_FLT_INCTL2

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
31	R/W	0x0	EPWM_NREC_SDFM3_Z_EN EPWM Non-Recoverable SDFM3 Z Enable，使能 SDFM3_Z 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
30	R/W	0x0	EPWM_NREC_SDFM3_COHLS_EN

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			EPWM Non-Recoverable SDFM3 COHLS Enable，使能 SDFM3_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
29	R/W	0x0	EPWM_NREC_SDFM3_COLS_EN EPWM Non-Recoverable SDFM3 COLS Enable，使能 SDFM3_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
28	R/W	0x0	EPWM_NREC_SDFM3_COHS_EN EPWM Non-Recoverable SDFM3 COHS Enable，使能 SDFM3_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
27	R/W	0x0	EPWM_NREC_SDFM2_Z_EN EPWM Non-Recoverable SDFM2 Z Enable，使能 SDFM2_Z 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
26	R/W	0x0	EPWM_NREC_SDFM2_COHLS_EN EPWM Non-Recoverable SDFM2 COHLS Enable，使能 SDFM2_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
25	R/W	0x0	EPWM_NREC_SDFM2_COLS_EN EPWM Non-Recoverable SDFM2 COLS Enable，使能 SDFM2_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
24	R/W	0x0	EPWM_NREC_SDFM2_COHS_EN EPWM Non-Recoverable SDFM2 COHS Enable，使能 SDFM2_COH 作为 EPWM 不可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能
23	R/W	0x0	EPWM_NREC_SDFM1_Z_EN EPWM Non-Recoverable SDFM1 Z Enable，使能 SDFM1_Z 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
22	R/W	0x0	EPWM_NREC_SDFM1_COHLS_EN EPWM Non-Recoverable SDFM1 COHLS Enable，使能 SDFM1_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
21	R/W	0x0	EPWM_NREC_SDFM1_COLS_EN EPWM Non-Recoverable SDFM1 COLS Enable，使能 SDFM1_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
20	R/W	0x0	EPWM_NREC_SDFM1_COHS_EN EPWM Non-Recoverable SDFM1 COHS Enable，使能 SDFM1_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
19	R/W	0x0	EPWM_NREC_SDFM0_Z_EN EPWM Non-Recoverable SDFM0 Z Enable，使能 SDFM0_Z 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
18	R/W	0x0	EPWM_NREC_SDFM0_COHLS_EN EPWM Non-Recoverable SDFM0 COHLS Enable，使能 SDFM0_COHL 作为 EPWM 不可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能
17	R/W	0x0	EPWM_NREC_SDFM0_COLS_EN EPWM Non-Recoverable SDFM0 COLS Enable，使能 SDFM0_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
16	R/W	0x0	EPWM_NREC_SDFM0_COHS_EN EPWM Non-Recoverable SDFM0 COHS Enable，使能 SDFM0_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
15	R/W	0x0	EPWM_REC_SDFM3_Z_EN EPWM Recoverable SDFM3 Z Enable，使能 SDFM3_Z 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
14	R/W	0x0	EPWM_REC_SDFM3_COHLS_EN EPWM Recoverable SDFM3 COHLS Enable，使能 SDFM3_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
13	R/W	0x0	EPWM_REC_SDFM3_COLS_EN EPWM Recoverable SDFM3 COLS Enable，使能 SDFM3_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
12	R/W	0x0	EPWM_REC_SDFM3_COHS_EN EPWM Recoverable SDFM3 COHS Enable，使能 SDFM3_COH 作为 EPWM 可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能
11	R/W	0x0	EPWM_REC_SDFM2_Z_EN EPWM Recoverable SDFM2 Z Enable，使能 SDFM2_Z 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
10	R/W	0x0	EPWM_REC_SDFM2_COHLS_EN EPWM Recoverable SDFM2 COHLS Enable，使能 SDFM2_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
9	R/W	0x0	EPWM_REC_SDFM2_COLS_EN EPWM Recoverable SDFM2 COLS Enable，使能 SDFM2_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
8	R/W	0x0	EPWM_REC_SDFM2_COHS_EN EPWM Recoverable SDFM2 COHS Enable，使能 SDFM2_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
7	R/W	0x0	EPWM_REC_SDFM1_Z_EN EPWM Recoverable SDFM1 Z Enable，使能 SDFM1_Z 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
6	R/W	0x0	EPWM_REC_SDFM1_COHLS_EN EPWM Recoverable SDFM1 COHLS Enable，使能 SDFM1_COHL 作为 EPWM 可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能
5	R/W	0x0	EPWM_REC_SDFM1_COLS_EN EPWM Recoverable SDFM1 COLS Enable，使能 SDFM1_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
4	R/W	0x0	EPWM_REC_SDFM1_COHS_EN EPWM Recoverable SDFM1 COHS Enable，使能 SDFM1_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
3	R/W	0x0	EPWM_REC_SDFM0_Z_EN EPWM Recoverable SDFM0 Z Enable，使能 SDFM0_Z 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
2	R/W	0x0	EPWM_REC_SDFM0_COHLS_EN EPWM Recoverable SDFM0 COHLS Enable，使能 SDFM0_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
1	R/W	0x0	EPWM_REC_SDFM0_COLS_EN EPWM Recoverable SDFM0 COLS Enable，使能 SDFM0_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
0	R/W	0x0	EPWM_REC_SDFM0_COHS_EN EPWM Recoverable SDFM0 COHS Enable，使能 SDFM0_COH 作为 EPWM 可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能

9.1.6.29. 0x0080 HRPWM_EN

默认值：0x00000000			高精度 PWM 使能寄存器
位域	类型	默认值	描述
31	R/W	0x0	HRPWM_FRA_CNT_SET 此位写 1 将 bit[28:24]写入小数计数器中。
30:29	–	–	–
28:24	R/W	0x0	HRPWM_FRA_CNT 此字段用于设置小数部分的计数器，bit31 写 1 生效。通过此方式可以设置初始计数器数值。
23:20	R/W	0x0	HRPWM_PRD_VAL_LP HRPWM Period Value Load Point，设置寄存器 HRPWM_PRD_FRA_CFG 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：在计数器完成一个周期计数后更新影子寄存器 • 0x4：PUL_CNT_ZRO（即脉冲数计数器等于零时）加载 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 IN_SYNC，均可以触发影子寄存器更新 • 0x7：不更新寄存器
19	R/W	0x0	HRPWM_PRD_UPDT_MOD HRPWM Period Value Update Mode，设置寄存器控制位 HRPWM_PRD_FRA_CFG 的更新模式。 • 0x0：影子更新 • 0x1：立即更新
18	R/W	0x0	HRPWM_COMPB_EN 占空比 CMPB 的补偿使能，若需要使能高精度周期，此位需写 1。

默认值：0x00000000			高精度 PWM 使能寄存器
位域	类型	默认值	描述
17	R/W	0x0	HRPWM_COMPA_EN 占空比 CMPA 的补偿使能，若需要使能高精度周期，此位需写 1。
16	R/W	0x0	HRPWM_ FRA_EN 周期小数使能
15:13	–	–	–
12:8	R/W	0x0	HRPWM_PRD_FRA_RPT
7:4	R/W	0x0	HRPWM_VAL_LP HRPWM Value Load Point，设置 HRPWMA_P_DLY、HRPWMA_N_DLY 和 HRPWMB_P_DLY、HRPWMB_N_DLY 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPMW_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPMW_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_CNT_ZRO(即脉冲计数器 = 0 时)加载 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 IN_SYNC，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
3	R/W	0x0	HRPWM_VAL_UPDT_MOD HRPWM Value Update Mode，设置 HRPWMA_P_DLY、HRPWMA_N_DLY 和 HRPWMB_P_DLY、HRPWMB_N_DLY 的更新模式 • 0x0：影子更新 • 0x1：立即更新
2	R/W	0x0	HRPWM_BOUT_LOG B 输出的逻辑设置 • 0x0：正常逻辑 • 0x1：互补逻辑
1	R/W	0x0	HRPWM_AOUT_LOG A 输出的逻辑设置 • 0x0：正常逻辑 • 0x1：互补逻辑

默认值：0x00000000			高精度 PWM 使能寄存器
位域	类型	默认值	描述
0	R/W	0x0	HRPWM_EN High Resolution PWM Enable，高精度 PWM 功能使能。 • 0x0：未使能 • 0x1：使能

9.1.6.30. 0x0084 HPWM_AV

默认值：0x00000000			HPWM 计数器比较 A 值寄存器
位域	类型	默认值	描述
31: 13	–	–	–
12:8	R/W	0x0	HRPWMA_N_DLY HRPWMA N Delay，HPWM_A 的 N 延迟控制，延迟时间单位精度为 130 ps。 • 0x0：HPWM 值为 0，即对应 0x 130 ps • 0x1：HPWM 值为 1，即对应 1x 130 ps • 0x2：HPWM 值为 2，即对应 2x 130 ps • ... • 0x1F：HPWM 值为 31，即对应 31x 130 ps
7:5	–	–	–
4:0	R/W	0x0	HRPWMA_P_DLY HRPWMA P Delay，HPWM_A 的 P 延迟控制，延迟时间单位精度为 130 ps。 • 0x0：HPWM 值为 0，即对应 0x 130 ps • 0x1：HPWM 值为 1，即对应 1x 130 ps • 0x2：HPWM 值为 2，即对应 2x 130 ps • ... • 0x1F：HPWM 值为 31，即对应 31x 130 ps

9.1.6.31. 0x0088 HPWM_BV

默认值：0x00000000			HPWM 计数器比较 B 值寄存器
位域	类型	默认值	描述
31:13	–	–	–
12:8	R/W	0x0	HRPWMB_N_DLY HRPWMB N Delay，HPWM_B 的 N 延迟控制，延迟时间单位精度为 130 ps。 • 0x0：HPWM 值为 0，即对应 0x 130 ps • 0x1：HPWM 值为 1，即对应 1x 130 ps • 0x2：HPWM 值为 2，即对应 2x 130 ps • ... • 0x1F：HPWM 值为 31，即对应 31x 130 ps
7:5	–	–	–
4:0	R/W	0x0	HRPWMB_P_DLY HRPWMB P Delay，HPWM_B 的 P 延迟控制，延迟时间单位精度为 130 ps。 • 0x0：HPWM 值为 0，即对应 0x 130 ps • 0x1：HPWM 值为 1，即对应 1x 130 ps • 0x2：HPWM 值为 2，即对应 2x 130 ps • ... • 0x1F：HPWM 值为 31，即对应 31x 130 ps

9.1.6.32. 0x008C HPWM_PRD_SET

默认值：0x00000000			HPWM 周期配置寄存器
位域	类型	默认值	描述
31:5	–	–	–
4:0	R/W	0x0	HRPWM_PRD_FRA_CFG 周期小数部分

9.1.6.33. 0x0090 EPWM_CNT_CV

默认值：0x00000000			EPWM Counter 比较值 C
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_CV EPWM Counter C Value，EPWM 计数器比较值 C。当计数器达到此数值时可以产生脉冲信号，若是递增计数达到此值，产生 EPWM_CNT_U_CV 信号。若是递减计数到达此值，产生 EPWM_CNT_D_CV。

9.1.6.34. 0x0094 EPWM_CNT_DV

默认值：0x00000000			EPWM Counter 比较值 D
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	EPWM_CNT_DV EPWM Counter D Value，EPWM 计数器比较值 D。当计数器达到此数值时可以产生脉冲信号，若是递增计数达到此值，产生 EPWM_CNT_U_DV 信号。若是递减计数到达此值，产生 EPWM_CNT_D_DV。

9.1.6.35. 0x0098 EPWM_CVRCD_CTL

默认值：0x00000000			CVRCD 信号控制寄存器
位域	类型	默认值	描述
31	R/W	0x0	EPWM_CNT_CMPD_UPDT EPWM Counter Compare Value D Update，EPWM 计数器的比较值 D 值的更新模式。 • 0x0：影子寄存器更新模式，EPWM_CNT_DV 的值需要在加载点更新，加载点通过 EPWM_CMPD_LP 配置 • 0x1：直接更新模式，EPWM_CNT_DV 的值直接生效
30	R/W	0x0	EPWM_CNT_CMPC_UPDT EPWM Counter Compare Value C Update，EPWM 计数器的比较值 C 值的更新模式。

默认值：0x00000000			CVRCD 信号控制寄存器
位域	类型	默认值	描述
			• 0x0：影子寄存器更新模式，EPWM_CNT_CV 的值需要在加载点更新，加载点通过 EPWM_CMPC_LP 配置 • 0x1：直接更新模式，EPWM_CNT_CV 的值直接生效
29:27	R/W	0x0	EPWM_CMPD_LP EPWM Compare D Value Load Point，设置 EPWM_CNT_DV 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
26:24	R/W	0x0	EPWM_CMPC_LP EPWM Compare C Value Load Point，设置 EPWM_CNT_CV 的影子寄存器的加载点。 • 0x0：在 EPWM_CNT_Z 信号有效时加载 • 0x1：在 EPWM_CNT_PRD 信号有效时加载 • 0x2：在 EPWM_CNT_Z 信号或 EPWM_CNT_PRD 信号有效时加载 • 0x3：不更新寄存器值 • 0x4：PUL_UPDT(即进行新一轮脉冲群输出的起始时刻)，从影子寄存器进行更新 • 0x5：IN_SYNC，外部的同步信号 • 0x6：EPWM_CNT_Z 与 EPWM _SYNCI，均可以触发影子寄存器更新 • 0x7：不更新寄存器值
23:20	R/W	0x0	EPWM_CVRD_DIV EPWM Conversion D Divisor， 转换触发信号 D 的信号源的除数，设置此除数可以设定转换触发信号 D 的产生频次。 • 0x0：无信号产生 • 0x1：源信号产生一次则对应产生一次转换触发信号 • 0x2：源信号产生两次则对应产生一次转换触发信号

默认值：0x00000000			CVRCD 信号控制寄存器
位域	类型	默认值	描述
			• 0x3：源信号产生三次则对应产生一次转换触发信号 • ... • 0xF：源信号产生十五次则对应产生一次转换触发信号
19:16	R/W	0x0	EPWM_CVRC_DIV EPWM Conversion C Divisor，转换触发信号 C 的信号源的除数，设置此除数可以设定转换触发信号 C 的产生频次。 • 0x0：无信号产生 • 0x1：源信号产生一次则对应产生一次转换触发信号 • 0x2：源信号产生两次则对应产生一次转换触发信号 • 0x3：源信号产生三次则对应产生一次转换触发信号 • ... • 0xF：源信号产生十五次则对应产生一次转换触发信号
15	R/W	0x0	EPWM_CVRD_EN EPWM Conversion D Trigger Enable • 0x0：未使能 • 0x1：使能
14:12	–	–	–
11:8	R/W	0x0	EPWM_CVRD_SEL EPWM Conversion D Trigger Selection，EPWM 转换触发 D 信号选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：EPWM_CNT_U_CV • 0x8：EPWM_CNT_D_DV • 0x9：EPWM_CNT_U_DV • 0xA：EPWM_CNT_D_DV • others：Reserved

默认值：0x00000000			CVRCD 信号控制寄存器
位域	类型	默认值	描述
7	R/W	0x0	EPWM_CVRC_EN EPWM Conversion C Trigger Enable • 0x0：未使能 • 0x1：使能
6:4	–	–	–
3:0	R/W	0x0	EPWM_CVRC_SEL EPWM Conversion C Trigger Selection，EPWM 转换触发 C 信号选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：EPWM_CNT_U_CV • 0x8：EPWM_CNT_D_DV • 0x9：EPWM_CNT_U_DV • 0xA：EPWM_CNT_D_DV • others：Reserved

9.1.6.36. 0x009C EPWM_PSHPLL

默认值：0x00000000			EPWM Push-Pull 控制寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	BP_IDLE BP Idle Status，BP Idle 状态 • 0x0：低电平 • 0x1：高电平
2	R/W	0x0	AP_IDLE

默认值：0x00000000			EPWM Push-Pull 控制寄存器
位域	类型	默认值	描述
			AP Idle Status，AP Idle 状态 • 0x0：低电平 • 0x1：高电平
1	R/W	0x0	PP_CTL_INV_EN PP_CTL Invert Enable，PP_CTL 信号反相使能 • 0x0：未使能 • 0x1：使能，PP_CTL 信号反相
0	R/W	0x0	EPWM_PHSPLL_EN EPWM Push-Pull Enable，EPWM 推挽输出使能 • 0x0：未使能 • 0x1：使能，推挽输出功能生效

9.1.6.37. 0x00A0 EPWM_PUL_OUT_CFG

默认值：0x00000000			EPWM 脉冲输出配置寄存器
位域	类型	默认值	描述
31:5	–	–	–
4	R/WAC	0x0	EPWM_PUL_SW_UPDT Pulse Software Force Update，脉冲强制更新，该信号触发加载影子寄存器数值，以及产生一次中断。 • 0x0：无动作 • 0x1：产生一次软件强制触发更新信号，完成后自动清 0
3:1	–	–	–
0	R/W	0x0	EPWM_PUL_OUT_EN Pulse Ouput Enable，脉冲输出功能使能，使能后脉冲计数器开始计数。 • 0x0：未使能 • 0x1：使能，使能后脉冲计数器开始计数

9.1.6.38. 0x00A4 EPWM_PUL_OUT_NUM

默认值：0x00000000			EPWM 脉冲输出数量寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	PUL_OUT_NUM Pulse Ouput Number，脉冲输出数量，若需要发送 N 个脉冲，则配置为 N–1。

9.1.6.39. 0x00A8 DMA_CFG

默认值：0x00000000			DMA_CFG 寄存器
位域	类型	默认值	描述
31:11	–	–	–
10:8	R/W	0x0	DMA_CYC_NUM DMA Cycle Number，DMA 单周期搬运数据数量，单位 32bits • 0x0：1x • 0x1：2x • 0x2：3x • ... • 0x7：8x
7:4	R/W	0x0	DMA_REQ_SRC_SEL DMA REQ Source Select，DMA 请求信号源选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_U_CV • 0x7：EPWM_CNT_D_DV • 0x8：EPWM_CNT_U_DV • 0x9：EPWM_CNT_D_DV • 0xA：PUL_CNT_ZRO（即脉冲计数器等于零时） • others：Reserved

默认值：0x00000000			DMA_CFG 寄存器
位域	类型	默认值	描述
3:2	–	–	–
1	R/W	0x0	DMA_MAP_EN DMA MAP Enable，DMA MAP 使能控制 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	DMA_EN DMA Enable，DMA 使能控制 • 0x0：未使能 • 0x1：使能

9.1.6.40. 0x00B0 EPWM_NRCFLT_INCTL

默认值：0x00000000			EPWM 内部触发不可恢复故障保护寄存器
位域	类型	默认值	描述
31	–	–	–
30	R/W	0x0	EPWM_NREC_ADC1_WDG2_EN EPWM Non-recoverable ADC1_WDG2 Enable，使能 ADC1_WDG2 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
29	R/W	0x0	EPWM_NREC_ADC1_WDG1_EN EPWM Non-recoverable ADC1_WDG1 Enable，使能 ADC1_WDG1 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
28	R/W	0x0	EPWM_NREC_ADC1_WDG0_EN EPWM Non-recoverable ADC1_WDG0 Enable，使能 ADC1_WDG0 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能

默认值：0x00000000			EPWM 内部触发不可恢复故障保护寄存器
位域	类型	默认值	描述
27	R/W	0x0	EPWM_NREC_ADC0_WDG2_EN EPWM Non-recoverable ADC0_WDG2 Enable，使能 ADC0_WDG2 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
26	R/W	0x0	EPWM_NREC_ADC0_WDG1_EN EPWM Non-recoverable ADC0_WDG1 Enable，使能 ADC0_WDG1 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
25	R/W	0x0	EPWM_NREC_ADC0_WDG0_EN EPWM Non-recoverable ADC0_WDG0 Enable，使能 ADC0_WDG0 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
24:23	–	–	–
22	R/W	0x0	EPWM_NREC_SDFM3_COHL_EN EPWM Non-recoverable SDFM3_COHL Enable，使能 SDFM3_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
21	R/W	0x0	EPWM_NREC_SDFM3_COL_EN EPWM Non-recoverable SDFM3_COL Enable，使能 SDFM3_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
20	R/W	0x0	EPWM_NREC_SDFM3_COH_EN EPWM Non-recoverable SDFM3_COH Enable，使能 SDFM3_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能

默认值：0x00000000			EPWM 内部触发不可恢复故障保护寄存器
位域	类型	默认值	描述
19	–	–	–
18	R/W	0x0	EPWM_NREC_SDFM2_COHL_EN EPWM Non-recoverable SDFM2_COHL Enable，使能 SDFM2_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
17	R/W	0x0	EPWM_NREC_SDFM2_COL_EN EPWM Non-recoverable SDFM2_COL Enable，使能 SDFM2_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
16	R/W	0x0	EPWM_NREC_SDFM2_COH_EN EPWM Non-recoverable SDFM2_COH Enable，使能 SDFM2_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
15	–	–	–
14	R/W	0x0	EPWM_NREC_SDFM1_COHL_EN EPWM Non-recoverable SDFM1_COHL Enable，使能 SDFM1_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
13	R/W	0x0	EPWM_NREC_SDFM1_COL_EN EPWM Non-recoverable SDFM1_COL Enable，使能 SDFM1_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
12	R/W	0x0	EPWM_NREC_SDFM1_COH_EN EPWM Non-recoverable SDFM1_COH Enable，使能 SDFM1_ COH 作为 EPWM 不可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发不可恢复故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能
11	–	–	–
10	R/W	0x0	EPWM_NREC_SDFM0_COHL_EN EPWM Non-recoverable SDFM0_COHL Enable，使能 SDFM0_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
9	R/W	0x0	EPWM_NREC_SDFM0_COL_EN EPWM Non-recoverable SDFM0_COL Enable，使能 SDFM0_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
8	R/W	0x0	EPWM_NREC_SDFM0_COH_EN EPWM Non-recoverable SDFM0_COH Enable，使能 SDFM0_ COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
7	–	–	–
6	R/W	0x0	EPWM_NREC_CPM1_COHL_EN EPWM Non-recoverable CPM1_COHL Enable，使能 CPM1_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
5	R/W	0x0	EPWM_NREC_CPM1_COL_EN EPWM Non-recoverable CPM1_COL Enable，使能 CPM1_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
4	R/W	0x0	EPWM_NREC_CPM1_COH_EN

默认值：0x00000000			EPWM 内部触发不可恢复故障保护寄存器
位域	类型	默认值	描述
			EPWM Non-recoverable CPM1_COH Enable，使能 CPM1_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
3	–	–	–
2	R/W	0x0	EPWM_NREC_CPM0_COHL_EN EPWM Non-recoverable CPM0_COHL Enable，使能 CPM0_COHL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
1	R/W	0x0	EPWM_NREC_CPM0_COL_EN EPWM Non-recoverable CPM0_COL Enable，使能 CPM0_COL 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
0	R/W	0x0	EPWM_NREC_CPM0_COH_EN EPWM Non-recoverable CPM0_COH Enable，使能 CPM0_COH 作为 EPWM 不可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能

9.1.6.41. 0x00B4 EPWM_RCFLT_INCTL

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
31	–	–	–
30	R/W	0x0	EPWM_REC_ADC1_WDG2_EN EPWM Recoverable ADC1_WDG2 Enable，使能 ADC1_WDG2 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
29	R/W	0x0	EPWM_REC_ADC1_WDG1_EN EPWM Recoverable ADC1_WDG1 Enable，使能 ADC1_WDG1 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
28	R/W	0x0	EPWM_REC_ADC1_WDG0_EN EPWM Recoverable ADC1_WDG0 Enable，使能 ADC1_WDG0 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
27	R/W	0x0	EPWM_REC_ADC0_WDG2_EN EPWM Recoverable ADC0_WDG2 Enable，使能 ADC0_WDG2 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
26	R/W	0x0	EPWM_REC_ADC0_WDG1_EN EPWM Recoverable ADC0_WDG1 Enable，使能 ADC0_WDG1 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
25	R/W	0x0	EPWM_REC_ADC0_WDG0_EN EPWM Recoverable ADC0_WDG0 Enable，使能 ADC0_WDG0 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
24:23	R/W	0x0	Debug Hold Source Select • 0x0：来自 cpu0 或 cpu1 • 0x1：来自 cpu0 • 0x2：来自 cpu1 • 0x3：来自 cpu0 或 cpu1
22	R/W	0x0	EPWM_REC_SDFM3_COHL_EN

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
			EPWM Recoverable SDFM3_COHL Enable，使能 SDFM3_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
21	R/W	0x0	EPWM_REC_SDFM3_COL_EN EPWM Recoverable SDFM3_COL Enable，使能 SDFM3_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
20	R/W	0x0	EPWM_REC_SDFM3_COH_EN EPWM Recoverable SDFM3_COH Enable，使能 SDFM3_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
19	–	–	–
18	R/W	0x0	EPWM_REC_SDFM2_COHL_EN EPWM Recoverable SDFM2_COHL Enable，使能 SDFM2_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
17	R/W	0x0	EPWM_REC_SDFM2_COL_EN EPWM Recoverable SDFM2_COL Enable，使能 SDFM2_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
16	R/W	0x0	EPWM_REC_SDFM2_COH_EN EPWM Recoverable SDFM2_COH Enable，使能 SDFM2_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
15	–	–	–

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
14	R/W	0x0	EPWM_REC_SDFM1_COHL_EN EPWM Recoverable SDFM1_COHL Enable，使能 SDFM1_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
13	R/W	0x0	EPWM_REC_SDFM1_COL_EN EPWM Recoverable SDFM1_COL Enable，使能 SDFM1_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
12	R/W	0x0	EPWM_REC_SDFM1_COH_EN EPWM Recoverable SDFM1_COH Enable，使能 SDFM1_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
11	–	–	–
10	R/W	0x0	EPWM_REC_SDFM0_COHL_EN EPWM Recoverable SDFM0_COHL Enable，使能 SDFM0_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
9	R/W	0x0	EPWM_REC_SDFM0_COL_EN EPWM Recoverable SDFM0_COL Enable，使能 SDFM0_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
8	R/W	0x0	EPWM_REC_SDFM0_COH_EN EPWM Recoverable SDFM0_COH Enable，使能 SDFM0_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
7	–	–	–
6	R/W	0x0	EPWM_REC_CPM1_COHL_EN EPWM Recoverable CPM1_COH Enable，使能 CPM1_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
5	R/W	0x0	EPWM_REC_CPM1_COL_EN EPWM Recoverable CPM1_COL Enable，使能 CPM1_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
4	R/W	0x0	EPWM_REC_CPM1_COH_EN EPWM Recoverable CPM1_COH Enable，使能 CPM1_COH 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
3	–	–	–
2	R/W	0x0	EPWM_REC_CPM0_COHL_EN EPWM Recoverable CPM0_COHL Enable，使能 CPM0_COHL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
1	R/W	0x0	EPWM_REC_CPM0_COL_EN EPWM Recoverable CPM0_COL Enable，使能 CPM0_COL 作为 EPWM 可恢复故障的输入信号源。 • 0x0：不使能 • 0x1：使能
0	R/W	0x0	EPWM_REC_CPM0_COH_EN EPWM Recoverable CPM0_COH Enable，使能 CPM0_COH 作为 EPWM 可恢复故障的输入信号源。

默认值：0x00000000			EPWM 内部触发可恢复故障保护寄存器
位域	类型	默认值	描述
			• 0x0：不使能 • 0x1：使能

9.1.6.42. 0x00B8 EPWM_SC_SYNC_CTL

默认值：0x00000000			SDFM 以及 CPM 的同步控制寄存器
位域	类型	默认值	描述
31:16	–	–	–
15	R/W	0x0	EPWM_SDFM_SYNC_EN EPWM SDFM Synchronization Enable，SDFM 同步使能。 • 0x0：未使能 • 0x1：使能
14:12	R/W	0x0	EPWM_SDFM_SRC_SEL EPWM SDFM Synchronization Source Selection，SDFM 同步信号选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：EPWM_CNT_U_AV or EPWM_CNT_D_BV
11	R/W	0x0	EPWM_CPM_SYNC_EN EPWM Comparator Synchronization Enable，比较模块 CPM 同步使能。 • 0x0：未使能 • 0x1：使能
10:8	R/W	0x0	EPWM_CPM_SRC_SEL EPWM Comparator Synchronization Source Selection，比较模块 CPM 的同步源选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD

默认值：0x00000000			SDFM 以及 CPM 的同步控制寄存器
位域	类型	默认值	描述
			• 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：EPWM_CNT_U_AV or EPWM_CNT_D_BV
7:0	–	–	–

9.1.6.43. 0x00BC EPWM_SC_SYNC_DIV

默认值：0x00000000			SDFM 以及 CPM 的同步分频寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:6	R	0x0	EPWM_SDFM_SYNC_CNT EPWM SDFM Synchronization Count，SDFM 同步信号源产生的计数，表示目前已经产生了多少次的 SDFM 同步信号。
5:4	R/W	0x0	EPWM_SDFMM_SYNC_DIV EPWM SDFM Synchronization Divisor，SDFM 同步信号分频。 • 0x0：无同步信号产生 • 0x1：同步信号产生一次则对应产生一次有效同步信号 • 0x2：同步信号产生两次则对应产生一次有效同步信号 • 0x3：同步信号产生三次则对应产生一次有效同步信号
3:2	R	0x0	EPWM_CPM_SYNC_CNT EPWM Comparator Synchronization Count，CPM 同步信号源产生的计数，表示目前已经产生了多少次的 CPM 同步信号。
1:0	R/W	0x0	EPWM_CPM_SYNC_DIV EPWM Comparator Synchronization Divisor，CPM 同步信号分频。 • 0x0：无同步信号产生 • 0x1：同步信号产生一次则对应产生一次有效同步信号 • 0x2：同步信号产生两次则对应产生一次有效同步信号 • 0x3：同步信号产生三次则对应产生一次有效同步信号

9.1.6.44. 0x00C0 EPWM_ETRG0_SET

默认值：0x00000000			EPWM_ETRG0 配置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	EPWM_ETRG0_FLT_LEN EPWM ETRG0 Filter Length，ETRG0 的滤波长度设置。这里的时间单位是 1x 计数器计数时钟。
15:12	R/W	0x0	EPWM_ETRG0_FLT_ST_SET EPWM ETRG0 Filter Start Point Set，滤波起点时刻选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_U_CV • 0x7：EPWM_CNT_D_CV • 0x8：EPWM_CNT_U_DV • 0x9：EPWM_CNT_D_DV • others：Reserved
11:10	–	–	–
9:8	R/W	0x0	EPWM_ETRG0_FLT_MOD EPWM ETRG0 Filter Mode Set，ETRG0 滤波模式设置。 • 0x0/0x1：未对 ETRG0 进行过滤 • 0x2：消隐模式，在定义时间段内 ETRG0 无效 • 0x3：加窗模式，只在定义时间段内 ETRG0 无效
7	R/W	0x0	EPWM_ETRG0_SYNC_EN ETRG0 Sync Enable，ETRG0 同步使能，使能后 ETRG0 可以作为 EPWM 的相位同步信号。 • 0x0：未使能 • 0x1：使能
6:5	R/W	0x0	EPWM_ETRG0_SRC_POL

默认值：0x00000000			EPWM_ETRG0 配置寄存器
位域	类型	默认值	描述
			• 0x0：源信号的上升沿触发一次 ETRG0 信号 • 0x1：源信号的下降沿触发一次 ETRG0 信号 • 0x2/0x3：源信号的上下沿均会触发一次 ETRG0 信号
4:0	R/W	0x0	EPWM_ETRG0_SRC_SEL EPWM ETRG0 Source Select，ETRG0 的信号源选择。 • 0x0：未选择信号源，ETRG0 未生效 • 0x1：CPM0_COH • 0x2：CPM0_COL • 0x3：CPM0_COHL • 0x4：CPM1_COH • 0x5：CPM1_COL • 0x6：CPM1_COHL • 0x7：EPWM_SYNCIN0 • 0x8：EPWM_SYNCIN1 • 0x9：ADC0_WDG0_PWM • 0xA：ADC0_WDG1_PWM • 0xB：ADC0_WDG2_PWM • 0xC：ADC1_WDG0_PWM • 0xD：ADC1_WDG1_PWM • 0xE：ADC1_WDG2_PWM • 0xF：EPWM_CVRC • 0x10：EPWM_CVRD • 0x11：EPWM0_TO • 0x12：EPWM1_TO • 0x13：EPWM2_TO • 0x14：EPWM3_TO • 0x15：EPWM4_TO • 0x16：EPWM5_TO • 0x17：EPWM6_TO • 0x18：EPWM7_TO • 0x19：EPWM8_TO • 0x1A：EPWM9_TO

默认值：0x00000000			EPWM_ETRG0 配置寄存器
位域	类型	默认值	描述
			• 0x1B：EPWM10_TO • 0x1C：EPWM11_TO • others：无效

9.1.6.45. 0x00C4 EPWM_ETRG1_SET

默认值：0x00000000			EPWM_ETRG1 配置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	EPWM_ETRG1_FLT_LEN EPWM ETRG1 Filter Length，ETRG1 的滤波长度设置。这里的时间单位是 1x 计数器计数时钟。
15:12	R/W	0x0	EPWM_ETRG1_FLT_ST_SET EPWM ETRG1 Filter Start Point Set，滤波起点时刻选择。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_U_CV • 0x7：EPWM_CNT_D_DV • 0x8：EPWM_CNT_U_DV • 0x9：EPWM_CNT_D_DV • others：Reserved
11:10	–	–	–
9:8	R/W	0x0	EPWM_ETRG1_FLT_MOD EPWM ETRG1 Filter Mode Set，ETRG1 滤波模式设置。 • 0x0/0x1：未对 ETRG1 进行过滤 • 0x2：消隐模式，在定义时间段内 ETRG1 无效 • 0x3：加窗模式，只在定义时间段内 ETRG1 无效
7	R/W	0x0	EPWM_ETRG1_SYNC_EN

默认值：0x00000000			EPWM_ETRG1 配置寄存器
位域	类型	默认值	描述
			ETRG1 Sync Enable，ETRG1 同步使能，使能后 ETRG1 可以作为 EPWM 的相位同步信号。 • 0x0：未使能 • 0x1：使能
6:5	R/W	0x0	EPWM_ETRG1_SRC_POL • 0x0：源信号的上升沿触发一次 ETRG1 信号 • 0x1：源信号的下降沿触发一次 ETRG1 信号 • 0x2/0x3：源信号的上下沿均会触发一次 ETRG1 信号
4:0	R/W	0x0	EPWM_ETRG1_SRC_SEL EPWM ETRG1 Source Select，ETRG1 的信号源选择。 • 0x0：未选择信号源，ETRG1 未生效 • 0x1：CPM0_COH • 0x2：CPM0_COL • 0x3：CPM0_COHL • 0x4：CPM1_COH • 0x5：CPM1_COL • 0x6：CPM1_COHL • 0x7：EPWM_SYNCIN0 • 0x8：EPWM_SYNCIN1 • 0x9：ADC0_WDG0_PWM • 0xA：ADC0_WDG1_PWM • 0xB：ADC0_WDG2_PWM • 0xC：ADC1_WDG0_PWM • 0xD：ADC1_WDG1_PWM • 0xE：ADC1_WDG2_PWM • 0xF：EPWM_CVRC • 0x10：EPWM_CVRD • 0x11：EPWM0_TO • 0x12：EPWM1_TO • 0x13：EPWM2_TO • 0x14：EPWM3_TO • 0x15：EPWM4_TO

默认值：0x00000000			EPWM_ETRG1 配置寄存器
位域	类型	默认值	描述
			• 0x16：EPWM5_TO • 0x17：EPWM6_TO • 0x18：EPWM7_TO • 0x19：EPWM8_TO • 0x1A：EPWM9_TO • 0x1B：EPWM10_TO • 0x1C：EPWM11_TO • others：无效

9.1.6.46. 0x00C8 EPWM_ETRG_ACC

默认值：0x00000000			EPWM_ETRG 累加配置寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	EPWM_ETRG1_ACC_TH EPWM ETRG1 Accumulate Threshold，累积阈值设置，当 ETRG1 的源信号发生的有效事件数大于累积阈值的时候，ETRG1 才输出有效的事件信号。 **注：** 此字段为影子更新，每周期更新一次。
23:18	–	–	–
17	R/W	0x0	EPWM_ETRG1_ACC_MOD EPWM ETRG1 Accumulate Mode Set，累积功能模式选择，此位只有在累积功能使能后才有效。 • 0x0：单周期累积 • 0x1：多周期累积
16	R/W	0x0	EPWM_ETRG1_ACC_EN EPWM ETRG1 Accumulate Enable，ETRG1 累积功能使能。 • 0x0：未使能 • 0x1：使能
15:8	R/W	0x0	EPWM_ETRG0_ACC_TH EPWM ETRG0 Accumulate Threshold，累积阈值设置，当 ETRG0 的源信号发生的有效事件数大于累积阈值的时候，ETRG0 才输出有效的事件信号。

默认值：0x00000000			EPWM_ETRG 累加配置寄存器
位域	类型	默认值	描述
			注： 此字段为影子更新，每周期更新一次。
7:2	–	–	–
1	R/W	0x0	EPWM_ETRG0_ACC_MOD EPWM ETRG0 Accumulate Mode Set，累积功能模式选择，此位只有在累积功能使能后才有效。 • 0x0：单周期累积 • 0x1：多周期累积
0	R/W	0x0	EPWM_ETRG0_ACC_EN EPWM ETRG0 Accumulate Enable，ETRG0 累积功能使能。 • 0x0：未使能 • 0x1：使能

9.1.6.47. 0x00CC EPWM_GLB_CTL

默认值：0x00000000			EPWM 全局控制寄存器
位域	类型	默认值	描述
31:13	–	–	–
12:10	R	0x0	EPWM_GLB_LP_CNT EPWM Global Loadpoint Counter，EPWM 全局更新加载信号的计数器。 • 0x0：没有发生全局更新加载信号 • 0x1：发生 1 次全局更新加载信号 • ... • 0x7：发生 7 次全局更新加载信号
9:7	R/W	0x0	EPWM_GLB_LP_DIV EPWM Global Loapoint Divisor，EPWM 全局更新的信号分频系数。 • 0x0：不产生全局更新信号 • 0x1：每发生一次的更新加载信号，产生一次的全局更新 • 0x2：每发生两次的更新加载信号，产生一次的全局更新

默认值：0x00000000			EPWM 全局控制寄存器
位域	类型	默认值	描述
			• ... • 0x7：每发生七次的更新加载信号，产生一次的全局更新
6	–	–	–
5	R/W	0x0	EPWM_GLB_OS_EN EPWM Global Oneshot Update Enable，EPWM 全局更新单次更新使能。此位只有在 EPWM_GLB_UPDT_EN = 0x1 的时候生效。 如果单次更新使能后，寄存器的更新只有在 EPWM_GLB_OS_UPDT = 0x1 的时候按照选择的更新信号进行更新。 • 0x0：未使能，全局更新一直有效 • 0x1：使能，使能后全局更新只有在 EPWM_GLB_OS_UPDT = 0x1 的时候生效
4:1	R/W	0x0	EPWM_GLB_LP EPWM Global Loadpoint，EPWM 全局更新加载点。 • 0x0：CNT=0 时刻更新 • 0x1：CNT=PRD 时刻更新 • 0x2：CNT=0 or CNT=PRD 时刻更新 • 0x3：SYNCEVT 时刻更新 • 0x4：SYNCEVT 时刻或 CNT=0 时刻更新 • 0x5：SYNCEVT 时刻或 CNT=PRD 时刻更新 • 0x6：SYNCEVT 时刻或 CNT=0 or CNT=PRD 时刻更新 • Others：Reserved • 0xF：软件写 EPWM_SE_FRC_GLB_UPDT=0x1 时刻更新 **注：** SYNCEVT 时刻可参考图 9–10 : 同步输出信号源。
0	R/W	0x0	EPWM_GLB_UPDT_EN EPWM Global Upadte Enable，EPWM 全局更新使能 • 0x0：未使能 • 0x1：使能，使能后，设定的寄存器按照配置的全局更新信号 EPWM_GLB_LP 进行统一更新。

9.1.6.48. 0x00D0 EPWM_GLB_CTL2

默认值：0x00000000			EPWM 全局控制寄存器 2
位域	类型	默认值	描述
31:2	–	–	–
1	W1AC	0x0	EPWM_SW_GLB_UPDT EPWM Software Fource Global Update，EPWM 强制全局更新一次，此位写 1 后在强制产生一次有效的全局更新后自动清 0。
0	W1AC	0x0	EPWM_GLB_OS_UPDT EPWM Global Oneshot Update，使能 EPWM 进行单次全局更新，发生一次有效的全局更新后自动清 0。

9.1.6.49. 0x00D4 EPWM_GLB_CFG

默认值：0x00000000			EPWM 全局控制配置寄存器
位域	类型	默认值	描述
31:13	–	–	–
12	R/W	0x0	EPWM_FE_DZCTL_UPDT_GLBE EPWM_FE_DZCTL Global Update Enable，寄存器 EPWM_FE_DZCTL 全局更新使能。 • 0x0：未使能 • 0x1：使能
11	R/W	0x0	EPWM_RE_DZCTL_UPDT_GLBE EPWM_RE_DZCTL Global Update Enable，寄存器 EPWM_RE_DZCTL 全局更新使能。 • 0x0：未使能 • 0x1：使能
10	R/W	0x0	HRPWM_PRD_VAL_UPDT_GLBE HRPWM_PRD_VAL Global Update Enable，寄存器 HRPWM_PRD_VAL 全局更新使能。 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	HRPWM_VAL_UPDT_GLBE HRPWM_VAL Global Update Enable，寄存器 HRPWM_VAL 全局更新使能。

默认值：0x00000000			EPWM 全局控制配置寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
8	R/W	0x0	EPWM_CNT_CMPD_UPDT_GLBE EPWM_CNT_CMPD Global Update Enable，寄存器 EPWM_CNT_CMPD 全局更新使能。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	EPWM_CNT_CMPC_UPDT_GLBE EPWM_CNT_CMPC Global Update Enable，寄存器 EPWM_CNT_CMPC 全局更新使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	EPWM_DB_CTL_UPDT_GLBE EPWM_DB_CTL Global Update Enable，寄存器 EPWM_DB_CTL 全局更新使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	EPWM_ACT_SW_CT_UPDT_GLBE EPWM_ACT_SW_CT Global Update Enable，寄存器 EPWM_ACT_SW_CT 全局更新使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	EPWMB_ACT_UPDT_GLBE EPWMB_ACT Global Update Enable，寄存器 EPWMB_ACT 全局更新使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	EPWMA_ACT_UPDT_GLBE EPWMA_ACT Global Update Enable，寄存器 EPWMA_ACT 全局更新使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	EPWM_CNT_CMPB_UPDT_GLBE

默认值：0x00000000			EPWM 全局控制配置寄存器
位域	类型	默认值	描述
			EPWM_CNT_CMPB Global Update Enable，寄存器 EPWM_CNT_CMPB 全局更新使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	EPWM_CNT_CMPA_UPDT_GLBE EPWM_CNT_CMPA Global Update Enable，寄存器 EPWM_CNT_CMPA 全局更新使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	EPWM_CNT_PRD_UPDT_GLBE EPWM_CNT_PRD Global Update Enable，寄存器 EPWM_CNT_PRD 全局更新使能。 • 0x0：未使能 • 0x1：使能

9.1.6.50. 0x00D8 EPWM_LINK_CTL

默认值：0x00FFFFFF			EPWM 链接控制配置寄存器
位域	类型	默认值	描述
31:28	–	–	–
27:24	R/W	0xF	HPWM_PRD_SET_LINK HPWM_PRD_SET Register Link，当前 EPWM 的 HPWM_PRD_SET 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 HPWM_PRD_SET 寄存器的时候，同时也会写当前 EPWM 的 HPWM_PRD_SET 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
23:20	R/W	0xF	EPWM_GLB_CTL2_LINK

默认值：0x00FFFFFF			EPWM 链接控制配置寄存器
位域	类型	默认值	描述
			EPWM_GLB_CTL2 Register Link，当前 EPWM 的 EPWM_GLB_CTL2 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_GLB_CTL2 寄存器的时候，同时也会写当前 EPWM 的 EPWM_GLB_CTL2 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
19:16	R/W	0xF	EPWM_FE_DZCTL_LINK EPWM_FE_DZCTL Register Link，当前 EPWM 的 EPWM_FE_DZCTL 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_FE_DZCTL 寄存器的时候，同时也会写当前 EPWM 的 EPWM_FE_DZCTL 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
15:12	R/W	0xF	EPWM_RE_DZCTL_LINK EPWM_RE_DZCTL Register Link，当前 EPWM 的 EPWM_RE_DZCTL 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_RE_DZCTL 寄存器的时候，同时也会写当前 EPWM 的 EPWM_RE_DZCTL 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
11:8	R/W	0xF	EPWM_CNT_BV_LINK EPWM_CNT_BV Register Link，当前 EPWM 的 EPWM_CNT_BV 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_CNT_BV 寄存器的时候，同时也会写当前 EPWM 的 EPWM_CNT_BV 寄存器。

默认值：0x00FFFFFF			EPWM 链接控制配置寄存器
位域	类型	默认值	描述
			• 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
7:4	R/W	0xF	EPWM_CNT_AV_LINK EPWM_CNT_AV Register Link，当前 EPWM 的 EPWM_CNT_AV 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_CNT_AV 寄存器的时候，同时也会写当前 EPWM 的 EPWM_CNT_AV 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接
3:0	R/W	0xF	EPWM_CNT_PRDV_LINK EPWM_CNT_PRDV Register Link，当前 EPWM 的 EPWM_CNT_PRDV 寄存器链接配置。举例若当前 EPWM 选择链接 EPWM1，那么在 EPWM1 写 EPWM_CNT_PRDV 寄存器的时候，同时也会写当前 EPWM 的 EPWM_CNT_PRDV 寄存器。 • 0x0：链接到 EPWM0 • 0x1：链接到 EPWM1 • ... • 0xB：链接到 EPWM11 • Others：无效，未产生链接

9.1.6.51. 0x00DC EPWM_ETRG_DLY

默认值：0x00000000			EPWM ETRG 延迟配置寄存器
位域	类型	默认值	描述
31:26	–	–	–
25:16	R/W	0x0	ETRG1_DLY_SET ETRG1 Delay Set，ETRG1 信号延迟设置

默认值：0x00000000			EPWM ETRG 延迟配置寄存器
位域	类型	默认值	描述
			• 0x0：无延迟，延迟模块不生效 • 0x1：延迟 1xSYS_CLK • ... • 0x3FF：延迟 1023xSYS_CLK
15:10	–	–	–
9:0	R/W	0x0	ETRG0_DLY_SET ETRG0 Delay Set，ETRG0 信号延迟设置 • 0x0：无延迟，延迟模块不生效 • 0x1：延迟 1xSYS_CLK • ... • 0x3FF：延迟 1023xSYS_CLK

9.1.6.52. 0x00E0 EPWM_DMA_ENTRY

默认值：0x00000000			EPWM DMA 输入入口
位域	类型	默认值	描述
31:0	R/W	0x0	EPWM_DMA_ENTRY

9.1.6.53. 0x00E4 EPWM_DMA_MAP0

默认值：0x00000000			EPWM DMA 映射配置寄存器 0
位域	类型	默认值	描述
31:24	R/W	0x0	DMA_CNT3_MAP DMA Counter3 Map Offset，DMA 第四个映射地址。
23:16	R/W	0x0	DMA_CNT2_MAP DMA Counter2 Map Offset，DMA 第三个映射地址。
15:8	R/W	0x0	DMA_CNT1_MAP DMA Counter1 Map Offset，DMA 第二个映射地址。
7:0	R/W	0x0	DMA_CNT0_MAP DMA Counter0 Map Offset，DMA 第一个映射地址。

9.1.6.54. 0x00E8 EPWM_DMA_MAP1

默认值：0x00000000			EPWM DMA 映射配置寄存器 1
位域	类型	默认值	描述
31:24	R/W	0x0	DMA_CNT7_MAP DMA Counter7 Map Offset，DMA 第八个映射地址。
23:16	R/W	0x0	DMA_CNT6_MAP DMA Counter6 Map Offset，DMA 第七个映射地址。
15:8	R/W	0x0	DMA_CNT5_MAP DMA Counter5 Map Offset，DMA 第六个映射地址。
7:0	R/W	0x0	DMA_CNT4_MAP DMA Counter4 Map Offset，DMA 第五个映射地址。

9.1.6.55. 0x00F0 EPWM_TRG_OUT

默认值：0x0000000F			EPWM 触发输出寄存器
位域	类型	默认值	描述
31:3	–	–	–
4	R/W	0x0	EPWM_TRG_OUT_INV_EN EPWM Trigger Output Invertor Enable，EPWM Trigger Out 输出信号的反相器使能。 • 0x0：未使能 • 0x1：使能
3:0	R/W	0xF	EPWM_CNT_TRIG_SEL EPWM Trigger Set，EPWM 触发信号源设置。 • 0x0：EPWM_CNT_Z • 0x1：EPWM_CNT_PRD • 0x2：EPWM_CNT_U_AV • 0x3：EPWM_CNT_D_AV • 0x4：EPWM_CNT_U_BV • 0x5：EPWM_CNT_D_BV • 0x6：EPWM_CNT_Z or EPWM _CNT_PRD • 0x7：EPWM_CNT_U_CV

默认值：0x0000000F			EPWM 触发输出寄存器
位域	类型	默认值	描述
			• 0x8：EPWM_CNT_D_CV • 0x9：EPWM_CNT_U_DV • 0xA：EPWM_CNT_D_DV • 0xF：无信号输出

9.1.6.56. 0x0FC EPWM_VER

默认值：0x00000220			EPWM 子模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0220	版本寄存器 V2.2

9.1.7. CAP 模块寄存器描述

9.1.7.1. 0x0000 CAP_CNT_V

默认值：0x00000000			CAP Counter 计数数值
位域	类型	默认值	描述
31:0	R/W	0x0	CAP_CNT_V CAP Counter Value，CAP 计数器的计数值。 对此字段进行读操作，可以读取此时 CAP 计数器对此字段进行写操作，可以设置 CAP 计数器的计数值，CAP 计数器立马更新此计数值，计数方式维持原有的设置进行

9.1.7.2. 0x0004 CAP_CNT_PH

默认值：0x00000000			CAP Counter 计数同步相位
位域	类型	默认值	描述
31:0	R/W	0x0	CAP_CNT_PH CAP Counter Phase，CAP 计数器的同步相位。 当出现同步信号时，此时 CAP 计数器进行同步，加载此寄存器的相位值进行计数。

9.1.7.3. 0x0008 CAP_REG0

默认值：0x00000000			CAP_REG0 寄存器
位域	类型	默认值	描述
31:0	R or R/W	0x0	CAP_REG0， CAP Register 0，CAP 寄存器 0。 在 Capture 模式下，此寄存器为只读，用于捕捉事件 0 的捕捉寄存器，捕捉计数器数值。在普通 PWM 模式下，此寄存器为可读写，此寄存器用于计数器周期值 CAP_CNT_PRDV。

9.1.7.4. 0x000C CAP_REG1

默认值：0x00000000			CAP_REG1 寄存器
位域	类型	默认值	描述
31:0	R or R/W	0x0	CAP_REG1 CAP Register 1，CAP 寄存器 1。 在 Capture 模式下，此寄存器用于捕捉事件 1 的捕捉寄存器，捕捉计数器数值。在普通 PWM 模式下，此寄存器用于计数器比较值 CAP_CNT_CMPV。

9.1.7.5. 0x0010 CAP_REG2

默认值：0x00000000			CAP_REG2 寄存器
位域	类型	默认值	描述
31:0	R or R/W	0x0	CAP_REG2 CAP Register 2，CAP 寄存器 2。 在 Capture 模式下，此寄存器用于捕捉事件 2 的捕捉寄存器，捕捉计数器数值。在普通 PWM 模式下，此寄存器用于计数器周期值 CAP_CNT_PRDV 的影子寄存器。

9.1.7.6. 0x0014 CAP_REG3

默认值：0x00000000			CAP_REG3 寄存器
位域	类型	默认值	描述
31:0	R or R/W	0x0	CAP_REG3 CAP Register 3，CAP 寄存器 3。

默认值：0x00000000			CAP_REG3 寄存器
位域	类型	默认值	描述
			在 Capture 模式下，此寄存器用于捕捉事件 3 的捕捉寄存器，捕捉计数器数值。在普通 PWM 模式下，此寄存器用于计数器比较值 CMPV 的影子寄存器。

9.1.7.7. 0x0018 CAP_CONF1

默认值：0x00000000			CAP 配置寄存器 1
位域	类型	默认值	描述
31:16	–	–	–
15:14	R/W	0x0	CAP_EM_CTL CAP Emulation Control，调试模式控制 • 0x0：CAP 内部计数器在调试断点停止计数 • 0x1：CAP 内部计数器在调试断点时继续计数，直至计数到零时停止计数 • other：CAP 内部计数器不停止计数
13:9	R/W	0x0	CAP_IN_EDG_EVN_DIV Capture Input Edge Event Divisor，输入捕捉信号的边沿除数，经过此除数可以将翻转频次较高的信号处理为翻转频次较低的信号。 • 0x0：输入信号 1 次翻转边沿进行 1 次的翻转处理，即跟随输入源信号 • 0x1：输入信号 2 次翻转边沿进行 1 次的翻转处理 • 0x2：输入信号 4 次翻转边沿进行 1 次的翻转处理 • 0x3：输入信号 6 次翻转边沿进行 1 次的翻转处理 • 0x4：输入信号 8 次翻转边沿进行 1 次的翻转处理 • 0x5：输入信号 10 次翻转边沿进行 1 次的翻转处理 • ... • 0x1E：输入信号 60 次翻转边沿进行 1 次的翻转处理 • 0x1F：输入信号 62 次翻转边沿进行 1 次的翻转处理
8	R/W	0x0	CAP_REG_LD_EN Capture Register Load Enable，捕捉寄存器加载使能，即发生捕捉事件 0/1/2/3 时，使能捕捉寄存器 0/1/2/3 进行计数器数值的捕捉。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	CAP_EVNT3_RST

默认值：0x00000000			CAP 配置寄存器 1
位域	类型	默认值	描述
			Capture Event3 Reset，捕捉事件 3 触发后进行 CAP 计数器的复位。 • 0x0：无操作 • 0x1：复位计数器至 0
6	R/W	0x0	CAP_EVNT3_POL Capture Event3 Polarity Set，捕捉事件 3 的极性控制。 • 0x0：上升沿触发 • 0x1：下降沿触发
5	R/W	0x0	CAP_EVNT2_RST Capture Event2 Reset，捕捉事件 2 触发后进行 CAP 计数器的复位。 • 0x0：无操作 • 0x1：复位计数器至 0
4	R/W	0x0	CAP_EVNT2_POL Capture Event1 Polarity Set，捕捉事件 1 的极性控制。 • 0x0：上升沿触发 • 0x1：下降沿触发
3	R/W	0x0	CAP_EVNT1_RST Capture Event1 Reset，捕捉事件 1 触发后进行 CAP 计数器的复位。 • 0x0：无操作 • 0x1：复位计数器至 0
2	R/W	0x0	CAP_EVNT1_POL Capture Event1 Polarity Set，捕捉事件 1 的极性控制。 • 0x0：上升沿触发 • 0x1：下降沿触发
1	R/W	0x0	CAP_EVNT0_RST Capture Event0 Reset，捕捉事件 0 触发后进行 CAP 计数器的复位。 • 0x0：无操作 • 0x1：复位计数器至 0
0	R/W	0x0	CAP_EVNT0_POL Capture Event0 Polarity Set，捕捉事件 0 的极性控制。

默认值：0x00000000			CAP 配置寄存器 1
位域	类型	默认值	描述
			• 0x0：上升沿触发 • 0x1：下降沿触发

9.1.7.8. 0x001C CAP_CONF2

默认值：0x00000000			CAP 配置寄存器 2
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	CAP_PWM_OUT_INIT Capture PWM Output Initial Value，PWM 输出初始值，即在 PWM 模式下，未使能 CAP Counter 时的输出值。 • 0x0：输出为低电平 • 0x1：输出为高电平
10	R/W	0x0	CAP_PWM_POL Capture PWM Output Polarity，PWM 的输出极性。 • 0x0：输出高有效 • 0x1：输出低有效
9	R/W	0x0	CAP_MODE_SEL Capture Mode Select，捕捉模式选择。 • 0x0：捕捉模式 • 0x1：PWM 模式
8	R/WAC	0x0	CAP_SW_FRC_SYNC Capture Software Force Synchronous Signal，软件产生强制的同步信号。 • 0x0：无操作 • 0x1：产生同步信号
7:6	R/W	0x0	CAP_SYNC_OUT Capture Synchronous Output Select，同步输出信号选择。 • 0x0：选择 CAP_SYNC_IN 或者软件产生的同步信号 0x1：CAP_CNT_PRD 信号

默认值：0x00000000			CAP 配置寄存器 2
位域	类型	默认值	描述
			0x2：无同步输出信号 0x3：无同步输出信号
5	R/W	0x0	CAP_SYNC_IN_EN Capture Synchronous Input Enable，同步输入使能，使能 CAP 计数器进行同步，同步值为 CAP_CNT_PH，同步使能信号为 CAP_SYNC_IN 或者软件产生的强制同步信号。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	CAP_CNT_EN Capture Counter Enable，捕捉计数器使能。 • 0x0：未使能 • 0x1：使能，计数器开始计数，且捕获或普通 PWM 功能启动
3	R/WAC	0x0	CAP_EVNT_RST Capture Event Reset，复位捕捉事件的起点，从捕捉事件 0 重新开始新的捕捉。 • 0x0：无操作 • 0x1：复位捕捉事件的起点
2:1	R/W	0x0	CAP_OS_EP Capture One-shot End Point，单次捕捉终点设置。 • 0x0：完成捕捉事件 0 结束 • 0x1：完成捕捉事件 1 结束 • 0x2：完成捕捉事件 2 结束 • 0x3：完成捕捉事件 3 结束
0	R/W	0x0	CAP_OS_MOD_EN Capture One-shot Mode Enable，单次捕捉使能。 • 0x0：连续捕捉 • 0x1：单次捕捉

9.1.7.9. 0x0020 CAP_INT_EN

默认值：0x00000000			CAP 中断使能寄存器
位域	类型	默认值	描述
31:8	–	–	–
7	R/W	0x0	CAP_CNT_CMP_INT_EN Capture Counter Compare Interrupt Enable，捕捉计数器计数达到比较值的中断使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	CAP_CNT_PRD_INT_EN Capture Counter Period Interrupt Enable，捕捉计数器计数达到周期值的中断使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	CAP_CNT_OVFL_INT_EN Capture Counter Overflow Interrupt Enable，捕捉计数器上溢出中断使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	CAP_EVNT3_INT_EN Capture Event3 Interrupt Enable，捕捉事件 3 的中断使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	CAP_EVNT2_INT_EN Capture Event2 Interrupt Enable，捕捉事件 2 的中断使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	CAP_EVNT1_INT_EN Capture Event1 Interrupt Enable，捕捉事件 1 的中断使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	CAP_EVNT0_INT_EN Capture Event0 Interrupt Enable，捕捉事件 0 的中断使能。

默认值：0x00000000			CAP 中断使能寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
0	–	–	–

9.1.7.10. 0x0024 CAP_FLG

默认值：0x00000000			CAP 标记寄存器
位域	类型	默认值	描述
31:8	–	–	–
7	R	0x0	CAP_CNT_CMP_FLG Capture Counter Compare Flag，捕捉计数器计数达到比较值的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
6	R	0x0	CAP_CNT_PRD_FLG Capture Counter Period Flag，捕捉计数器计数达到周期值的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
5	R	0x0	CAP_CNT_OVFL_FLG Capture Counter Overflow Flag，捕捉计数器上溢出状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
4	R	0x0	CAP_EVNT3_FLG Capture Event3 Flag，捕捉事件 3 的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
3	R	0x0	CAP_EVNT2_FLG Capture Event2 Flag，捕捉事件 2 的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号

默认值：0x00000000			CAP 标记寄存器
位域	类型	默认值	描述
2	R	0x0	CAP_EVNT1_FLG Capture Event1 Flag，捕捉事件 1 的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
1	R	0x0	CAP_EVNT0_FLG Capture Event0 Flag，捕捉事件 0 的状态标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号
0	R	0x0	CAP_INT_FLG Capture Interrupt Flag，捕捉模块的中断信号标记。 • 0x0：无标记信号产生 • 0x1：产生标记信号

9.1.7.11. 0x0028 CAP_FLG_CLR

默认值：0x00000000			CAP 标记清除寄存器
位域	类型	默认值	描述
31:8	–	–	–
7	R/WAC	0x0	CAP_CNT_CMP_CLR Capture Counter Compare Flag Clear，捕捉计数器计数达到比较值的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
6	R/WAC	0x0	CAP_CNT_PRD_CLR Capture Counter Period Flag Clear，捕捉计数器计数达到周期值的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
5	R/WAC	0x0	CAP_CNT_OVFL_CLR Capture Counter Overflow Flag Clear，捕捉计数器上溢出状态标记清除。

默认值：0x00000000			CAP 标记清除寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：清除标记信号
4	R/WAC	0x0	CAP_EVNT3_CLR Capture Event3 Flag Clear，捕捉事件 3 的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
3	R/WAC	0x0	CAP_EVNT2_CLR Capture Event2 Flag Clear，捕捉事件 2 的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
2	R/WAC	0x0	CAP_EVNT1_CLR Capture Event1 Flag Clear，捕捉事件 1 的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
1	R/WAC	0x0	CAP_EVNT0_CLR Capture Event0 Flag Clear，捕捉事件 0 的状态标记清除。 • 0x0：无操作 • 0x1：清除标记信号
0	–	–	–

9.1.7.12. 0x002C CAP_SW_FRC

默认值：0x00000000			软件强制 CAP 事件产生寄存器
位域	类型	默认值	描述
31:8	–	–	–
7	R/WAC	0x0	CAP_SW_FRC_CNT_CMP Capture Software Force Counter Compare，强制捕捉计数器计数达到比较值的信号产生。 • 0x0：无操作 • 0x1：强制产生信号

默认值：0x00000000			软件强制 CAP 事件产生寄存器
位域	类型	默认值	描述
6	R/WAC	0x0	CAP_SW_FRC_CNT_PRD Capture Software Force Counter Period，强制捕捉计数器计数达到周期值的信号产生。 • 0x0：无操作 • 0x1：强制产生信号
5	R/WAC	0x0	CAP_SW_FRC_CNT_OVFL Capture Software Force Counter Overflow，强制计数器上溢出信号的产生。 • 0x0：无操作 • 0x1：强制产生信号
4	R/WAC	0x0	CAP_SW_FRC _EVNT3 Capture Software Force Event3，强制捕捉事件的 EVNT3 的产生。 • 0x0：无操作 • 0x1：强制产生信号
3	R/WAC	0x0	CAP_SW_FRC _EVNT2 Capture Software Force Event2，强制捕捉事件的 EVNT2 的产生。 • 0x0：无操作 • 0x1：强制产生信号
2	R/WAC	0x0	CAP_SW_FRC _EVNT1 Capture Software Force Event1，强制捕捉事件的 EVNT1 的产生。 • 0x0：无操作 • 0x1：强制产生信号
1	R/WAC	0x0	CAP_SW_FRC_EVNT0 Capture Software Force Event0，强制捕捉事件的 EVNT0 的产生。 • 0x0：无操作 • 0x1：强制产生信号
0	–	–	–

9.1.7.13. 0x0030 CAP_IN_FLT

默认值：0x00000000			CAP 输入滤波寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:8	R/W	0x0	CAP_SMP_CTL_PRD Capture Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	CAP_IN_PRE_FLT， CAP Input Pre Filter Enable，CAP 输入信号的预滤波使能，预滤波采样周期为 SYS_CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	CAP_IN_FLT_SEL， Capture Input Filter Seclect，输入滤波功能选择，此字段只有在输入捕捉模式有效。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.7.14. 0x0034 CAP_IN_SRC

默认值：0x00000000			CAP 输入选择寄存器
位域	类型	默认值	描述
31:18	–	–	–
17:16	R	0x0	CAP_IND_NXT_EVNT Indicate Next Capture Event，下一个捕获事件的指示。 • 0x0：捕获事件 0 • 0x1：捕获事件 1 • 0x2：捕获事件 2 • 0x3：捕获事件 3
15:6	–	–	–
5:0	R/W	0x0	CAP_IN_SRC_SEL， Capture Input Source Seclect，输入信号源选择，此字段只有在输入捕捉模式有效。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_A • 0x2：EPWM1_A • 0x3：EPWM2_A • … • 0xC：EPWM11_A • 0x11：EPWM0_B • 0x12：EPWM1_B • … • 0x1C：EPWM11_B • 0x20：CPM0_COH • 0x21：CPM0_COL • 0x22：CPM0_COHL • 0x23：CPM1_COH • 0x24：CPM1_COL • 0x25：CPM1_COHL • 0x30：ECAT_SYNC0

默认值：0x00000000			CAP 输入选择寄存器
位域	类型	默认值	描述
			• 0x31：ECAT_SYNC1 • others：Reserved

9.1.7.15. 0x0038 DMA_CFG

默认值：0x00000000			DMA 配置寄存器
位域	类型	默认值	描述
31:17	–	–	–
16	R	0x0	DMA_ERR_STS DMA Error Status，DMA 异常状态 • 0x0：无状态发生 • 0x1：发生异常状态，当前 DMA 请求未完成出现新的 DMA 请求
15:4	–	–	–
3:1	R/W	0x0	DMA_REQ_SRC_SEL DMA REQ Source Select，DMA 请求信号源选择。 • 0x0：CAP_EVNT0 • 0x1：CAP_EVNT1 • 0x2：CAP_EVNT2 • 0x3：CAP_EVNT3 • Others：无效
0	R/W	0x0	DMA_EN DMA Enable，DMA 使能控制。 • 0x0：未使能 • 0x1：使能

9.1.7.16. 0x00FC CAP_VER

默认值：0x00000110			CAP 子模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0110	版本寄存器 V1.1

默认值：0x00000110			CAP 子模块版本寄存器
位域	类型	默认值	描述

9.1.8. QEP 模块寄存器描述

9.1.8.1. 0x0000 QEP_POS_CNT_V

默认值：0x00000000			QEP Position Counter 计数值
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_POS_CNT_V QEP Position Counter Value，QEP 位置计数器的计数值。只要 QEP 的位置计数器开始了计数，此字段只能读操作，不可写操作。

9.1.8.2. 0x0004 QEP_POS_CNT_SP

默认值：0x00000000			QEP Position Counter 起点值
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_POS_CNT_SPV QEP Position Counter Start Point Value，QEP 位置计数器的起点值。 设置此字段表示外部 QEP_S 或 QEP_I 信号所对应的位置。

9.1.8.3. 0x0008 QEP_POS_CNT_EP

默认值：0x00000000			QEP Position Counter 终点值
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_POS_CNT_EPV QEP Position Counter End Point Value，QEP 位置计数器的终点值。 设置此字段表示外部编码器的最大旋转的位置。

9.1.8.4. 0x000C QEP_POS_CNT_CMPV

默认值：0x00000000			QEP Position Counter 比较值
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_POS_CNT_CMPV QEP Position Counter Compare，QEP 位置计数器的比较值。 当 QEP 位置计数器计数值达到这里设定的比较值，可以产生信号 QEP_POS_CNT_CMP_S。 此信号可以用于生成 QEP 模块的中断信号或输出的同步信号 QEP_SYNC_OUT。

9.1.8.5. 0x0010 QEP_POS_CNT_ICAPV

默认值：0x00000000			QEP Position Counter Index 捕获值
位域	类型	默认值	描述
31:0	R	0x0	QEP_POS_CNT_ICAPV QEP Position Counter Index Capture，QEP 位置计数器的 Index 触发捕获值。当发生 Index 信号相关的事件（可通过寄存器配置），触发捕获 QEP 位置计数器的计数值。

9.1.8.6. 0x0014 QEP_POS_CNT_SCAPV

默认值：0x00000000			QEP Position Counter Strobe 捕获值
位域	类型	默认值	描述
31:0	R	0x0	QEP_POS_CNT_SCAPV QEP Position Counter Strobe Capture Value，QEP 位置计数器的 Strobe 触发捕获值。 当发生 Strobe 信号相关的事件（可通过寄存器配置），触发捕获 QEP 位置计数器的计数值。

9.1.8.7. 0x0018 QEP_POS_CNT_TOCAPV

默认值：0x00000000			QEP Position Counter Timout 捕获值
位域	类型	默认值	描述
31:0	R	0x0	QEP_POS_CNT_TOCAPV

默认值：0x00000000			QEP Position Counter Timout 捕获值
位域	类型	默认值	描述
			QEP Position Counter Timeout Capture Value，QEP 位置计数器的 Timeout 触发捕获值。 当 Timer 的 Timeout 信号有效，触发捕获 QEP 位置计数器的计数值。

9.1.8.8. 0x001C QEP_TMR_V

默认值：0x00000000			QEP Timer 寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_TMR_V， QEP Timer Value，定时器 Timer 数值。 当 TMR 的数值达到 QEP_TMR_PRD 的数值，则产生，QEP Timer 的 Timeout 信号。

9.1.8.9. 0x0020 QEP_TMR_PRD_V

默认值：0x00000000			QEP Timer Period 寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_TMR_PRD_V QEP Timer Period Value，定时器 Timer 周期数值。 当 TMR 的数值达到 QEP_TMR_PRD 的数值，则产生，QEP Timer 的 Timeout 信号。

9.1.8.10. 0x0024 QEP_WD_TMRV

默认值：0x00000000			QEP Watchdog 定时器寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_WD_TMRV QEP Watchdog Timer Value，看门狗定时器的数值。 当 TMR 的数值达到 QEP_WD_PRD 的数值，则产生，QEP Watchdog Timer 的 Timeout 中断。

9.1.8.11. 0x0028 QEP_WD_PRDV

默认值：0x00000000			QEP Watchdog 定时器周期寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_WD _PRDV QEP Watchdog Timer Period Value，看门狗定时器的周期值。 当 TMR 的数值达到 QEP_WD_PRD 的数值，则产生，QEP Watchdog Timer 的 Timeout 中断。

9.1.8.12. 0x002C QEP_DEC_CONF

默认值：0x00000000			QEP Decoder 配置寄存器
位域	类型	默认值	描述
31:17	–	–	–
16:14	R/W	0x0	QEP_CNT_MODE，QEP Counter Mode，QEP 计数模式选择。 • 0x0：正交计数模式，QEP_A 与 B 正交 • 0x1：方向计数，QEP_A 作为 QCLK，QEP_B 作为 QDIR • 0x2：递增计数模式，QEP_A 作为 QCLK ，QEP_B 无效 • 0x3：递减计数模式，QEP_A 作为 QCLK ，QEP_B 无效 • 0x4：高有效 CW/CCW 计数方式，QEP_A 为 CW 信号，QEP_B 为 CCW 信号，高有效 • 0x5：低有效 CW/CCW 计数方式，QEP_A 为 CW 信号，QEP_B 为 CCW 信号，低有效 • Others：正交计数模式
13	R/W	0x0	QEP_SOUT_EN QEP Strobe OUTPUT Enable，使能 QEP_SOUT 输出（即 QEP 位置计数器达到比较值的信号）。 • 0x0：未使能。 • 0x1：使能
12	R/W	0x0	QEP_SRB_OUT_PIN QEP Strobe Output PIN Select，QEP_SOUT 输出信号的管脚输出选择。 • 0x0：QEP_I • 0x1：QEP_S
11	R/W	0x0	QEP_QA_DEDG_EN

默认值：0x00000000			QEP Decoder 配置寄存器
位域	类型	默认值	描述
			QEP QA Double Edge Enable，使能 QEP_A 输入信号上下边沿有效，此功能只针对在 QEP_A 作为 QCLK 使用的计数模式下。 • 0x0：未使能，QEP_A 的上升沿有效 • 0x1：QEP_A 双沿有效
10	R/W	0x0	QEP_SWAP_AB QEP SWAP AB，交换 QEP_A 与 QEP_B 信号。 • 0x0：无操作 • 0x1：QEP_A 与 QEP_B 信号交换
9	R/W	0x0	QEP_IGATE QEP Index Gate Set，设置 Index Gate，表示是否需要使能信号 QEP_S 对 QEP_I 进行 Gate 处理： • 0x0：使能 • 0x1：不使能
8	R/W	0x0	QEP_A_INV_EN QEP_A Inverse Enable，QEP_A 信号反相使能。 • 0x0：无操作 • 0x1：信号反相
7	R/W	0x0	QEP_B_INV_EN QEP_B Inverse Enable，QEP_B 信号反相使能。 • 0x0：无操作 • 0x1：信号反相
6	R/W	0x0	QEP_I_INV_EN QEP_I Inverse Enable，QEP_I 信号反相使能。 • 0x0：无操作 • 0x1：信号反相
5	R/W	0x0	QEP_S_INV_EN QEP_S Inverse Enable，QEP_S 信号反相使能。 • 0x0：无操作 • 0x1：信号反相
4:0	–	–	–

9.1.8.13. 0x0030 QEP_POS_CNT_CONF

默认值：0x00000000			QEP 位置计数器配置寄存器
位域	类型	默认值	描述
31:18	–	–	–
17	R/W	0x0	QEP_CVRB_LATCH_EN QEP CVRB Latch Positon Data Enable，CVRB 信号触发锁存位置数据使能，使能后寄存器 QEP_CVRB_LATCH 生效。 • 0x0：未使能 • 0x1：使能
16	R/W	0x0	QEP_CVRA_LATCH_EN QEP CVRA Latch Positon Data Enable，CVRA 信号触发锁存位置数据使能，使能后寄存器 QEP_CVRA_LATCH 生效。 • 0x0：未使能 • 0x1：使能
15:14	–	–	–
13:12	R/W	0x0	QEP_POS_CNT_RST QEP Position Counter Reset，位置计数器复位方式。 • 0x0：复位模式 1，QEP_IDX_MKR 信号复位位置计数器 • 0x1：复位模式 2，QEP_POS_CNT_EP 信号位置计数器复位 • 0x2：复位模式 3，QEP_FIDX_MKR 信号位置计数器复位 • 0x3：复位模式 4，QEP_TMR_TO 信号位置计数器复位
11:10	R/W	0x0	QEP_POS_SRB_INIT QEP Position SRB Initialization，QEP_SRB 初始化位置计数器 • 0x0：无操作 • 0x1：无操作 • 0x2：以 QEP_SRB 的上升沿进行初始化位置计数器，QEP_POS_CNT_SPV 写入位置计数器 • 0x3：正向前进时，以 QEP_SRB 上升沿进行初始化，反向前进时，以 QEP_SRB 下降沿进行初始化
9:8	R/W	0x0	QEP_POS_IDX_INIT QEP Position IDX Initialization，QEP_IDX 初始化位置计数器。 • 0x0：无操作 • 0x1：无操作

默认值：0x00000000			QEP 位置计数器配置寄存器
位域	类型	默认值	描述
			• 0x2：以 QEP_IDX 的上升沿进行初始化位置计数器，QEP_POS_CNT_SPV 写入位置计数器 • 0x3：以 QEP_IDX 的下降沿进行初始化位置计数器，QEP_POS_CNT_SPV 写入位置计数器
7	R/WAC	0x0	QEP_POS_SW_INIT，QEP Position Software Initialization，软件初始化位置计数器 • 0x0：无操作 • 0x1：进行软件的初始化位置计数器，完成自动清 0
6	R/W	0x0	QEP_POS_SCAP_MOD QEP Position Strobe Trigger Capture Mode，Strobe 触发捕获模式，位置计数器的数值捕获至寄存器 QEP_POS_CNT_SCAP。 • 0x0：QEP_SRB 信号的上升沿进行捕获（将 QEP_SRB 信号反相即可以实现下降沿捕获）。 • 0x1：正向前进时，以 QEP_SRB 上升沿进行捕获，反向前进时，以 QEP_SRB 下降沿进行捕获
5:4	R/W	0x0	QEP_POS_ICAP_MOD QEP Position Index Trigger Capture Mode，Index 触发捕获模式，位置计数器的数值捕获至寄存器 QEP_POS_CNT_ICAP • 0x0：Reserved • 0x1：在 QEP_IDX 的上升沿捕获 QEP 位置计数器 • 0x2：在 QEP_IDX 的下降沿捕获 QEP 位置计数器 • 0x3：QEP_IDX_MKR 捕获 QEP 位置计数器
3	R/W	0x0	QEP_POS_CNT_EN QEP Position Counter Enable，位置计数器使能 • 0x0：位置计数器复位 • 0x1：使能位置计数器
2	R/W	0x0	QEP_CAP_LH_MOD QEP Capture Latch Mode Select，QEP 捕获锁存模式选择 • 0x0：在 CPU 读取 QEP_POS_CNT_V 时，进行下面的操作： QEP_CAP_TMRV 的数值锁存 QEP_CAP_TMR_LH QEP_CAP_DELT 的数值锁存在 QEP_CAP_DELT_THT_TH • 0x1：在 QEP 定时器产生 Timeout 信号时，进行下面的操作： QEP_POS_CNT_V 的数值锁存在 QEP_POS_CNT_TOCAPV

默认值：0x00000000			QEP 位置计数器配置寄存器
位域	类型	默认值	描述
			QEP_CAP_TMRV 的数值锁存 QEP_CAP_TMR_LH QEP_CAP_DELT 的数值锁存在 QEP_CAP_DELT_TH
1	R/W	0x0	QEP_TMR_EN QEP Timer Enable，QEP 定时器使能 • 0x0：关闭 QEP 定时器 • 0x1：使能 QEP 定时器
0	R/W	0x0	QEP_WD_TMR_EN QEP Watchdog Timer Enable，QEP 看门狗定时器使能 • 0x0：关闭看门狗定时器。 • 0x1：使能看门狗定时器

9.1.8.14. 0x0034 QEP_CAP_TMR_CONF

默认值：0x00000000			QEP Capture Timer 配置寄存器
位域	类型	默认值	描述
31:9	–	–	–
15	R/W	0x0	QEP_CAP_TMR_EN QEP Capture Timer Enable，QEP 捕捉定时器使能。 • 0x0：关闭定时器。 • 0x1：使能定时器
14:7	–	–	–
6:4	R/W	0x0	QEP_CAP_TMR_CLK_DIV QEP Capture Timer Clock Division，捕获定时器的时钟分频系数 • 0x0：1，SYS_CLK/1 • 0x1：2，SYS_CLK/2 • 0x2：4，SYS_CLK/4 • 0x3：8，SYS_CLK/8 • 0x4：16，SYS_CLK/16 • 0x5：32，SYS_CLK/32

默认值：0x00000000			QEP Capture Timer 配置寄存器
位域	类型	默认值	描述
			• 0x6：64，SYS_CLK/64 • 0x7：128，SYS_CLK/128
3:0	R/W	0x0	QEP_SET_UDIST QEP Set Unit Distance，设置单位的位置距离 • 0x0：1 次有效的 QCLK，即 1x 最小位移 • 0x1：2 次有效的 QCLK，即 2x 最小位移 • 0x2：4 次有效的 QCLK，即 4x 最小位移 • 0x3：8 次有效的 QCLK，即 8x 最小位移 • … • 0xB：2048 次有效的 QCLK，即 2048x 最小位移 • Others：Reserved

9.1.8.15. 0x0038 QEP_POS_CNTCMP_CONF

默认值：0x00000000			QEP 位置计数器比较寄存器
位域	类型	默认值	描述
31:16	–	–	–
15	R/W	0x0	QEP_POS_CMP_UPDT_MOD QEP Position Compare Register Update Mode，位置比较寄存器的更新模式。 • 0x0：立即更新模式。 • 0x1：影子更新模式
14	R/W	0x0	QEP_POS_CMP_LP QEP Position Compare Register Load Point，位置比较寄存器影子更新的加载点。 • 0x0：在位置计数器值为 0 的时候加载 • 0x1：在位置计数器值为 QEP_POS_CNT_QEP_POS_CNT_CMP 的时候加载
13	R/W	0x0	QEP_SOUT_POL QEP SOUT Polarity，SOUT 输出信号的极性。 • 0x0：高有效脉冲 • 0x1：低有效脉冲
12	R/W	0x0	QEP_POS_CMP_EN

默认值：0x00000000			QEP 位置计数器比较寄存器
位域	类型	默认值	描述
			QEP Position Compare Enable，QEP 位置计数比较使能。 • 0x0：未使能 • 0x1：使能
11:0	R/W	0x0	QEP_SOUT_PW， QEPS OUT Pulse Width，SOUT 输出信号的输出脉宽信号。 • 0x0：1 x 4 x SYS_CLK • 0x1：2 x 4 x SYS_CLK • 0x2：3 x 4 x SYS_CLK • 0x3：4 x 4 x SYS_CLK • … • 0xFFF：4096 x 4 x SYS_CLK

9.1.8.16. 0x003C QEP_INT_EN

默认值：0x00000000			QEP Interrupt 使能寄存器
位域	类型	默认值	描述
31:13	–	–	–
12	R/W	0x0	QEP_CW_CCW_ERR_INT_EN QEP CW/CCW Signal Error Interrupt Enable，CW/CCW 信号异常中断使能。 • 0x0：未使能 • 0x1：使能
11	R/W	0x0	QEP_TMROT_INT_EN QEP Timer Timeout Interrupt Enable，发生 Timer 超时事件的中断使能。 • 0x0：未使能 • 0x1：使能
10	R/W	0x0	QEP_ICAP_INT_EN QEP Index Capture Interrupt Enable，发生 QEP_I 信号触发捕获位置计数器的事件的中断使能。 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			QEP Interrupt 使能寄存器
位域	类型	默认值	描述
9	R/W	0x0	QEP_SCAP_INT_EN QEP Strobe Capture Interrupt Enable，发生 QEP_S 信号触发捕获位置计数器的事件的中断使能。 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	QEP_POS_CMP_INT_EN QEP Position Compare Interrupt Enable，发生位置计数器达到 QEP_POS_CNT_CMP 的事件值的中断使能。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	QEP_POS_CMP_RDY_INT_EN QEP Position Compare Ready Interrupt Enable，指示位置比较寄存器的影子寄存器更新的中断使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	QEP_POS_CNT_OVFL_INT_EN QEP Position Counter Overflow Interrupt Enable，位置计数器上溢出中断使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	QEP_POS_CNT_UDFL_INT_EN QEP Position Counter Underflow Interrupt Enable，位置计数器下溢出中断使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	QEP_WDTO_INT_EN QEP Watchdog Timeout Interrupt Enable，看门狗超时中断使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	QEP_DIR_CHG_INT_EN QEP Direction Change Interrupt Enable，方向改变中断使能。 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			QEP Interrupt 使能寄存器
位域	类型	默认值	描述
2	R/W	0x0	QEP_QUADPH_ERR_INT_EN QEP Quadrature Phase Error Interrupt Enable，正交相位错误中断使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	QEP_POS_CNT_ERR_INT_EN QEP Position Counter Error Interrupt Enable，位置计数器计数错误中断使能。 • 0x0：未使能 • 0x1：使能
0	–	–	–

9.1.8.17. 0x0040 QEP_INT_FLG

默认值：0x00000000			QEP Interrupt 标志寄存器
位域	类型	默认值	描述
31:13	–	–	–
12	R	0x0	QEP_CW_CCW_ERR_INT_FLG QEP CW/CCW Signal Error Interrupt Flag，发送 CW/CCW 信号异常中断标记。 • 0x0：未产生 • 0x1：产生标记信号
11	R	0x0	QEP_TMROT_INT_FLG QEP Timer Timeout Interrupt Flag，发生，QEP Timer 超时事件的中断标记。 • 0x0：未产生 • 0x1：产生标记信号
10	R	0x0	QEP_ICAP_INT_FLG QEP Index Capture Interrupt Flag，发生 QEP_I 信号触发捕获位置计数器的事件的中断标记。 • 0x0：未产生 • 0x1：产生标记信号
9	R	0x0	QEP_SCAP_INT_FLG

默认值：0x00000000			QEP Interrupt 标志寄存器
位域	类型	默认值	描述
			QEP Strobe Capture Interrupt Flag，发生 QEP_S 信号触发捕获位置计数器的事件的中断标记。 • 0x0：未产生 • 0x1：产生标记信号
8	R	0x0	QEP_POS_CMP_INT_FLG QEP Position Compare Interrupt Flag，发生位置计数器达到 QEP_POS_CNT_CMP 的事件值的中断标记。 • 0x0：未产生 • 0x1：产生标记信号
7	R	0x0	QEP_POS_CMP_RDY_INT_FLG QEP Position Compare Ready Interrupt Flag，指示位置比较寄存器的影子寄存器更新的中断标记。 • 0x0：未产生 • 0x1：产生标记信号
6	R	0x0	QEP_POS_CNT_OVFL_INT_FLG QEP Position Counter Overflow Interrupt Flag，位置计数器上溢出中断标记。 • 0x0：未产生 • 0x1：产生标记信号
5	R	0x0	QEP_POS_CNT_UDFL_INT_FLG QEP Position Counter Underflow Interrupt Flag，位置计数器下溢出中断标记。 • 0x0：未产生 • 0x1：产生标记信号
4	R	0x0	QEP_WDTO_INT_FLG QEP Watchdog Timeout Interrupt Flag，看门狗超时中断标记。 • 0x0：未产生 • 0x1：产生标记信号
3	R	0x0	QEP_DIR_CHG_INT_FLG QEP Direction Change Interrupt Flag，方向改变中断标记。 • 0x0：未产生 • 0x1：产生标记信号

默认值：0x00000000			QEP Interrupt 标志寄存器
位域	类型	默认值	描述
2	R	0x0	QEP_QUADPH_ERR_INT_FLG QEP Quadrature Phase Error Interrupt Flag，正交相位错误中断标记。此位只有在正交计数下有效，其它计数模式此位无意义。 • 0x0：未产生 • 0x1：产生标记信号
1	R	0x0	QEP_POS_CNT_ERR_INT_FLG QEP Position Counter Error Interrupt Flag，位置计数器计数错误中断标记。此状态只有在 QEP_POS_CNT_RST = 0x0 的时候有效，其它模式下可忽略此状态。 • 0x0：未产生 • 0x1：产生标记信号
0	R	0x0	QEP_INT_FLG，QEP 中断标记。 • 0x0：未产生 • 0x1：产生标记信号

9.1.8.18. 0x0044 QEP_INT_CLR

默认值：0x00000000			QEP 中断清除寄存器
位域	类型	默认值	描述
31:13	–	–	–
12	R/WAC	0x0	QEP_CW_CCW_ERR_INT_CLR QEP CW/CCW Signal Error Interrupt Clear，发送 CW/CCW 信号异常中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
11	R/WAC	0x0	QEP_TMROT_INT_CLR QEP Timer Timeout Interrupt Clear，发生，QEP Timer 超时事件的中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
10	R/WAC	0x0	QEP_ICAP_INT_CLR QEP Index Capture Interrupt Clear，发生 QEP_I 信号触发捕获位置计数器的事件的中断清除。

默认值：0x00000000			QEP 中断清除寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
9	R/WAC	0x0	QEP_SCAP_INT_CLR QEP Strobe Capture Interrupt Clear，发生 QEP_S 信号触发捕获位置计数器的事件的中断清除 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
8	R/WAC	0x0	QEP_POS_CMP_INT_CLR QEP Position Compare Interrupt Clear，发生位置计数器达到 QEP_POS_CNT_CMP 的事件值的中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
7	R/WAC	0x0	QEP_POS_CMP_RDY_INT_CLR QEP Position Compare Ready Interrupt Clear，指示位置比较寄存器的影子寄存器更新的中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
6	R/WAC	0x0	QEP_POS_CNT_OVFL_INT_CLR QEP Position Counter Overflow Interrupt Clear，位置计数器上溢出中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
5	R/WAC	0x0	QEP_POS_CNT_UDFL_INT_CLR QEP Position Counter Underflow Interrupt Clear，位置计数器下溢出中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
4	R/WAC	0x0	QEP_WDTO_INT_CLR QEP Watchdog Timeout Interrupt Clear，看门狗超时中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
3	R/WAC	0x0	QEP_DIR_CHG_INT_CLR QEP Direction Change Interrupt Clear，方向改变中断清除。

默认值：0x00000000			QEP 中断清除寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
2	R/WAC	0x0	QEP_QUADPH_ERR_INT_CLR QEP Quadrature Phase Error Interrupt Clear，正交相位错误中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
1	R/WAC	0x0	QEP_POS_CNT_ERR_INT_CLR QEP Position Counter Error Interrupt Clear，位置计数器计数错误中断清除。 • 0x0：无操作 • 0x1：清除 FLAG，完成后自动清 0
0	–	–	–

9.1.8.19. 0x0048 QEP_SW_FRC_INT

默认值：0x00000000			软件强制 QEP 中断产生寄存器
位域	类型	默认值	描述
31:12	–	–	–
12	R/WAC	0x0	QEP_SW_FRC_CW_CCW_ERR_INT QEP Software Force CW/CCW Signal Error Interrupt，软件强制发生 CW/CCW 信号异常中断。 • 0x0：无操作 • 0x1：软件强制产生信号
11	R/WAC	0x0	QEP_SW_FRC_TMROT_INT QEP Software Force Timer Timeout Interrupt，软件强制发生，QEP Timer 超时事件的中断。 • 0x0：无操作 • 0x1：软件强制产生信号
10	R/WAC	0x0	QEP_SW_FRC_ICAP_INT QEP Software Force Index Capture Interrupt，软件强制发生 QEP_I 信号触发捕获位置计数器的事件的中断。

默认值：0x00000000			软件强制 QEP 中断产生寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：软件强制产生信号
9	R/WAC	0x0	QEP_SW_FRC_SCAP_INT QEP Software Force Strobe Capture Interrupt，软件强制发生 QEP_S 信号触发捕获位置计数器的事件的中断。 • 0x0：无操作 • 0x1：软件强制产生信号
8	R/WAC	0x0	QEP_SW_FRC_POS_CMP_INT QEP Software Force Position Compare Interrupt，软件强制发生位置计数器达到 QEP_POS_CNT_CMP 的事件值的中断。 • 0x0：无操作 • 0x1：软件强制产生信号
7	R/WAC	0x0	QEP_SW_FRC_POS_CMP_RDY_INT QEP Software Force Position Compare Ready Interrupt，软件强制发生指示位置比较寄存器的影子寄存器已更新的中断。 • 0x0：无操作 • 0x1：软件强制产生信号
6	R/WAC	0x0	QEP_SW_FRC_POS_CNT_OVFL_INT QEP Software Force Position Counter Overflow Interrupt，软件强制发生位置计数器上溢出中断。 • 0x0：无操作 • 0x1：软件强制产生信号
5	R/WAC	0x0	QEP_SW_FRC_POS_CNT_UDFL_INT QEP Software Force Position Counter Underflow Interrupt，软件强制发生位置计数器下溢出中断。 • 0x0：无操作 • 0x1：软件强制产生信号
4	R/WAC	0x0	QEP_SW_FRC_WDTO_INT QEP Software Force Watchdog Timeout Interrupt，软件强制发生看门狗超时中断。

默认值：0x00000000			软件强制 QEP 中断产生寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：软件强制产生信号
3	R/WAC	0x0	QEP_SW_FRC_DIR_CHG_INT QEP Software Force Direction Change Interrupt，软件强制发生方向改变中断。 • 0x0：无操作 • 0x1：软件强制产生信号
2	R/WAC	0x0	QEP_SW_FRC_QUADPH_ERR_INT QEP Software Force Quadrature Phase Error Interrupt，软件强制发生正交相位错误中断。 • 0x0：无操作 • 0x1：软件强制产生信号
1	R/WAC	0x0	QEP_SW_FRC_POS_CNT_ERR_INT QEP Software Force Position Counter Error Interrupt，软件强制产生位置计数器计数错误中断。 • 0x0：无操作 • 0x1：软件强制产生信号
0	–	–	–

9.1.8.20. 0x004C QEP_STS

默认值：0x00000000			QEP 状态寄存器
位域	类型	默认值	描述
31:8	–	–	–
7	R/W1C	0x0	QEP_UD_EVNT_FLG QEP Unit Distance Event Flag，单位位置距离的状态标志。 • 0x0：未产生标志 • 0x1：产生标志
6	R	0x0	QEP_FIDX_MKR_DIR_FLG QEP First Index Marker Direction Flag，第一 Index 标记的方向状态标志。此状态只有在 QEP_POS_CNT_RST = 0x0/0x2 的时候有效，其它模式下可忽略此状态。

默认值：0x00000000			QEP 状态寄存器
位域	类型	默认值	描述
			• 0x0：逆时针/反向旋转 • 0x1：顺时针/正向旋转
5	R	0x0	QEP_DIR_FLG QEP Direction Flag，正交信号解析的方向标记。 • 0x0：逆时针/反向旋转 • 0x1：顺时针/正向旋转
4	R	0x0	QEP_DIR_IEVNT_FLG QEP Direciton Flag of Index Event，QEP_INDX_MKR 有效边沿对应的方向状态标记。此状态只有在 QEP_POS_CNT_RST = 0x0 的时候有效，其它模式下可忽略此状态。 • 0x0：逆时针/反向旋转 • 0x1：顺时针/正向旋转
3	R/W1C	0x0	QEP_CAP_TMR_OVFL_FLG QEP Capture Timer Overflow Flag，捕捉功能模块的捕捉定时器出现上溢出状态。 • 0x0：未发现溢出 • 0x1：捕捉定时器出现上溢出
2	R/W1C	0x0	QEP_CAP_DERR_FLG QEP Capture Direction Error Flag，捕捉功能模块发现方向错误状态。 • 0x0：未发现错误 • 0x1：捕捉功能发现在捕捉期间出现了方向的改变
1	R/W1C	0x0	QEP_FIDX_MKR_FLG QEP First Index Marker Flag，QEP 模块使能后接收的第一个 Index 有效信号的标记。若此位置 1 后被清 0，不会再变化，只有重新复位可以再次置 1。此状态只有在 QEP_POS_CNT_RST = 0x0/0x2 的时候有效，其它模式下可忽略此状态。 • 0x0：第一个 Index 标记未发生 • 0x1：发生第一个 Index 标记
0	R	0x0	QEP_POS_CNT_ERR_FLG QEP Position Counter Error Flag，位置计数器的错误标记。此标记状态在每次的 QEP_INDX_EVNT 进行更新。此状态只有在 QEP_POS_CNT_RST = 0x0 的时候有效，其它模式下可忽略此状态。

默认值：0x00000000			QEP 状态寄存器
位域	类型	默认值	描述
			• 0x0：无错误发生 • 0x1：发生计数错误

9.1.8.21. 0x0050 QEP_CAP_TMRV

默认值：0x00000000			QEP Capture 定时器寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	QEP_CAP _TMRV QEP Capture Timer，QEP 捕捉定时器的数值。

9.1.8.22. 0x0054 QEP_CAP_DELT

默认值：0x00000000			QEP Capture Delta T 寄存器
位域	类型	默认值	描述
31:17	–	–	–
16	R	0x0	QEP_CAP_DELT_ERR QEP Capture Delta Time ERR，指示时间差值是否有错误，即发生了方向的改变。 • 0x0：时间差值正确，无发生方向的改变 • 0x1：时间差值错误，发生了方向的改变
15:0	R	0x0	QEP_CAP_DELT QEP Capture Delta Time，单位位移内的 QEP 捕捉定时器的时间差值。单位位移需要通过寄存器配置。完成一次的时间差捕捉，QEP 捕捉定时器复位至 0。

9.1.8.23. 0x0058 QEP_CAP_TMR_LH

默认值：0x00000000			QEP Capture 定时器锁存寄存器
位域	类型	默认值	描述
31:16	–	–	–

默认值：0x00000000			QEP Capture 定时器锁存寄存器
位域	类型	默认值	描述
15:0	R	0x0	QEP_CAP _TMR_LH QEP Capture Timer Latch，QEP 捕捉定时器的锁存数值。当 Timer 出现 timeout 信号或软件读取 QEP 位置计数器时，这里锁存 QEP_CAP_TMR 的定时器数值。

9.1.8.24. 0x005C QEP_CAP_DELT_LH

默认值：0x00000000			QEP Capture Delta T 锁存寄存器
位域	类型	默认值	描述
31:17	–	–	–
16	R	0x0	QEP_CAP_DELT_ERR QEP Capture Delta Time ERR，指示时间差值是否有错误，即发生了方向的改变。 • 0x0：时间差值正确，无发生方向的改变 • 0x1：时间差值错误，发生了方向的改变
15:0	R	0x0	QEP_CAP_DELT_LH QEP Capture Delta Time Latch，单位位移内的 QEP 捕捉定时器的时间差锁存。当 Timer 出现 timeout 信号或软件读取 QEP 位置计数器时，这里锁存 QEP_CAP_DELT 的寄存器数值。

9.1.8.25. 0x0060 QEP_IN_FLT1

默认值：0x00000000			QEP 输入滤波寄存器 1
位域	类型	默认值	描述
31:24	R/W	0x0	QEP_B_SMP_CTL_PRD QEP B Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK

默认值：0x00000000			QEP 输入滤波寄存器 1
位域	类型	默认值	描述
23:20	R/W	0x0	QEP_B_IN_PRE_FLT QEP B Input Pre Filter Enable，QEPB 输入信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果。 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
19:16	R/W	0x0	QEP_B_IN_FLT_SEL QEP B Input Filter Seclect，输入滤波功能选择，此字段只有在输入捕捉模式有效。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
15:8	R/W	0x0	QEP_A _SMP_CTL_PRD QEP A Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	QEP_A_IN_PRE_FLT QEP A Input Pre Filter Enable，QEPA 输入信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果。 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号

默认值：0x00000000			QEP 输入滤波寄存器 1
位域	类型	默认值	描述
			• 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	QEP_A_IN_FLT_SEL QEP A Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.8.26. 0x0064 QEP_IN_FLT2

默认值：0x00000000			QEP 输入滤波寄存器 2
位域	类型	默认值	描述
31:24	R/W	0x0	QEP_S_SMP_CTL_PRD QEP S Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
23:20	R/W	0x0	QEP_S_IN_PRE_FLT QEP S Input Pre Filter Enable，QEP S 输入信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号

默认值：0x00000000			QEP 输入滤波寄存器 2
位域	类型	默认值	描述
			• … • 0xF：15 采样点同步信号
19:16	R/W	0x0	QEP_S _IN_FLT_SEL QEP S Input Filter Seclect，输入滤波功能选择，此字段只有在输入捕捉模式有效。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
15:8	R/W	0x0	QEP_I _SMP_CTL_PRD QEP I Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	QEP_I_IN_PRE_FLT QEP I Input Pre Filter Enable，QEP I 输入信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	QEP_I_IN_FLT_SEL QEP I Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号

默认值：0x00000000			QEP 输入滤波寄存器 2
位域	类型	默认值	描述
			• 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.8.27. 0x0068 QEP_IN_CTL

默认值：0x00000002			QEP 输入控制
位域	类型	默认值	描述
31:26	–	–	–
25:16	R/W	0x0	QEP_SMP_WIN_TH QEP sample window threshold，QEP 输入信号的采样窗口的宽度阈值，单位为 clk。 • 0x0：0 x SYS_CLK（等同于无窗口） • 0x1：1 x SYS_CLK • 0x2：2 x SYS_CLK • 0x3：3x SYS_CLK • … • 0x3ff：1023 x SYS_CLK
15:10	–	–	–
9	R/W	0x0	QEP_S_OP_EN QEP S Output Enable，QEP S 信号使能作为输出信号。 • 0x0：QEP_S 为输入信号 • 0x1：QEP_S 为输出信号
8	R/W	0x0	QEP_I_OP_EN QEP I Output Enable，QEP I 信号使能作为输出信号。 • 0x0：QEP_I 为输入信号 • 0x1：QEP_I 为输出信号
7:4	–	–	–
3	R/W	0x0	QEP_SMP_WIN_EN

默认值：0x00000002			QEP 输入控制
位域	类型	默认值	描述
			QEP sample window enable，使能 QEP 输入信号的采样窗口，使能窗口后，窗口宽度内的 QEP 输入信号的变化认为是相同时刻的变化。 • 0x0：未使能 • 0x1：使能，窗口的宽度通过 Bit[25:16]确定
2	R/W	0x0	QEP_I_TYP QEP_I Sigal Type，QEP_I 的输入信号的类型，此信号只有在 QEP_I_INT_GATEA = 0 的时候才需要配置，否则无需对此位进行配置。 • 0x0：QEP_I 为未与 QEP_A 或 QEP_B 进行 Gate 处理的信号 • 0x1： QEP_I 为与 QEP_A 或 QEP_B 进行 Gate 处理的信号
1	R/W	0x1	QEP_I_INT_GATEA QEP_I Internal Gate A Disable，禁止 QEP_I 与 QEP_A 在滤波前进行内部的 Gate 处理。 • 0x0：QEP_I 与 QEP_A 在输入滤波前没有进行内部的 Gate 处理 • 0x1：QEP_I 与 QEP_A 在输入滤波前进行内部的 Gate 处理
0	R/W	0x0	QEP_IN_FLT_EN QEP Input Filter Enable，QEP 输入信号（QEP_A、QEP_B、QEP_I、QEP_S）的使能信号。使能后，各个输入信号才根据对应的输入滤波配置进行同步或滤波。 • 0x0：未使能，输入信号无效 • 0x1：使能，QEP 的输入信号的滤波模块使能

9.1.8.28. 0x0084 QEP_CVRA

默认值：0x00000000			QEP CVRA 锁存位置寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_CVRA_LATCH QEP CVRA Latch Position，EPWM CVRA 触发锁存的位置数据值。

9.1.8.29. 0x0088 QEP_CVRB

默认值：0x00000000			QEP CVRB 锁存位置寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	QEP_CVRB_LATCH QEP CVRB Latch Position，EPWM CVRB 触发锁存的位置数据值。

9.1.8.30. 0x008C QEP_INSRC

默认值：0x00000000			QEP 输入信号选择寄存器
位域	类型	默认值	描述
31:22	–	–	–
21:16	R/W	0x0	QEP_I_IN_SRC_SEL QEP I Input Source Seclect，QEP I 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_A • 0x2：EPWM1_A • 0x3：EPWM2_A • ... • 0xC：EPWM11_A • 0x11：EPWM0_B • 0x12：EPWM1_B • ... • 0x1C：EPWM11_B • 0x20：QOUT0_Z • 0x21：QOUT1_Z • Others：Reserved
15:14	–	–	–
13:8	R/W	0x0	QEP_B_IN_SRC_SEL QEP B Input Source Seclect，QEP B 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_B • 0x2：EPWM1_B

默认值：0x00000000			QEP 输入信号选择寄存器
位域	类型	默认值	描述
			• ... • 0xC：EPWM11_B • 0x20：QOUT0_B • 0x21：QOUT1_B • Others：Reserved
7:6	–	–	–
5:0	R/W	0x0	QEP_A_IN_SRC_SEL QEP A Input Source Seelect，QEP A 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_A • 0x2：EPWM1_A • 0x3：EPWM2_A • ... • 0xC：EPWM11_A • 0x20：QOUT0_A • 0x21：QOUT1_A • Others：Reserved

9.1.8.31. 0x00FC QEP_VER

默认值：0x00000120			QEP 子模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x00000120	版本寄存器 V1.2V1.2

9.1.9. Global 寄存器描述

9.1.9.1. 0x0004 GLB_EPWM_INT_STS

默认值：0x00000000			Global EPWM 中断状态寄存器
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	EPWM11_INT_STS EPWM11 Interrupt Status，EPWM11 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
10	R/W	0x0	EPWM10_INT_STS EPWM10 Interrupt Status，EPWM10 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
9	R/W	0x0	EPWM9_INT_STS EPWM9 Interrupt Status，EPWM9 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
8	R/W	0x0	EPWM8_INT_STS EPWM8 Interrupt Status，EPWM8 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
7	R/W	0x0	EPWM7_INT_STS EPWM7 Interrupt Status，EPWM7 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
6	R/W	0x0	EPWM6_INT_STS EPWM6 Interrupt Status，EPWM6 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
5	R/W	0x0	EPWM5_INT_STS

默认值：0x00000000			Global EPWM 中断状态寄存器
位域	类型	默认值	描述
			EPWM5 Interrupt Status，EPWM5 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
4	R/W	0x0	EPWM4_INT_STS EPWM4 Interrupt Status，EPWM4 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
3	R/W	0x0	EPWM3_INT_STS EPWM3 Interrupt Status，EPWM3 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
2	R/W	0x0	EPWM2_INT_STS EPWM2 Interrupt Status，EPWM2 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
1	R/W	0x0	EPWM1_INT_STS EPWM1 Interrupt Status，EPWM1 中断状态。 • 0x0：未发生中断 • 0x1：发生中断
0	R/W	0x0	EPWM0_INT_STS EPWM0 Interrupt Status，EPWM0 中断状态。 • 0x0：未发生中断 • 0x1：发生中断

9.1.9.2. 0x0008 GLB_FLT_INT_STS

默认值：0x00000000			Global EPWM Fault Interrupt 状态寄存器
位域	类型	默认值	描述
31:6	–	–	–
11	R	0x0	EPWM11_FLT_INT_STS

默认值：0x00000000			Global EPWM Fault Interrupt 状态寄存器
位域	类型	默认值	描述
			EPWM11 Fault Interrupt Status，EPWM11 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
10	R	0x0	EPWM10_FLT_INT_STS EPWM10 Fault Interrupt Status，EPWM10 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
9	R	0x0	EPWM9_FLT_INT_STS EPWM9 Fault Interrupt Status，EPWM9 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
8	R	0x0	EPWM8_FLT_INT_STS EPWM8 Fault Interrupt Status，EPWM8 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
7	R	0x0	EPWM7_FLT_INT_STS EPWM7 Fault Interrupt Status，EPWM7 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
6	R	0x0	EPWM6_FLT_INT_STS EPWM6 Fault Interrupt Status，EPWM6 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
5	R	0x0	EPWM5_FLT_INT_STS EPWM5 Fault Interrupt Status，EPWM5 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
4	R	0x0	EPWM4_FLT_INT_STS EPWM4 Fault Interrupt Status，EPWM4 故障中断状态。

默认值：0x00000000			Global EPWM Fault Interrupt 状态寄存器
位域	类型	默认值	描述
			• 0x0：无中断产生 • 0x1：产生中断信号
3	R	0x0	EPWM3_FLT_INT_STS EPWM3 Fault Interrupt Status，EPWM3 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
2	R	0x0	EPWM2_FLT_INT_STS EPWM2 Fault Interrupt Status，EPWM2 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
1	R	0x0	EPWM1_FLT_INT_STS EPWM1 Fault Interrupt Status，EPWM1 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
0	R	0x0	EPWM0_FLT_INT_STS EPWM0 Fault Interrupt Status，EPWM0 故障中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号

9.1.9.3. 0x000C GLB_CAP_INT_STS

默认值：0x00000000			Global CAP Interrupt 状态寄存器
位域	类型	默认值	描述
31:6	–	–	–
5	R	0x0	CAP5_INT_STS CAP5 Interrupt Status，CAP5 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
4	R	0x0	CAP4_INT_STS CAP4 Interrupt Status，CAP4 中断状态。

默认值：0x00000000			Global CAP Interrupt 状态寄存器
位域	类型	默认值	描述
			• 0x0：无中断产生 • 0x1：产生中断信号
3	R	0x0	CAP3_INT_STS CAP3 Interrupt Status，CAP3 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
2	R	0x0	CAP2_INT_STS CAP2 Interrupt Status，CAP2 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
1	R	0x0	CAP1_INT_STS CAP1 Interrupt Status，CAP1 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
0	R	0x0	CAP0_INT_STS CAP0 Interrupt Status，CAP0 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号

9.1.9.4. 0x0010 GLB_QEP_INT_STS

默认值：0x00000000			Global QEP Interrupt 状态寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R	0x0	QEP9_INT_STS QEP9 Interrupt Status，QEP9 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
8	R	0x0	QEP8_INT_STS QEP8 Interrupt Status，QEP8 中断状态。

默认值：0x00000000			Global QEP Interrupt 状态寄存器
位域	类型	默认值	描述
			• 0x0：无中断产生 • 0x1：产生中断信号
7	R	0x0	QEP7_INT_STS QEP7 Interrupt Status，QEP7 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
6	R	0x0	QEP6_INT_STS QEP6 Interrupt Status，QEP6 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
5	R	0x0	QEP5_INT_STS QEP5 Interrupt Status，QEP5 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
4	R	0x0	QEP4_INT_STS QEP4 Interrupt Status，QEP4 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
3	R	0x0	QEP3_INT_STS QEP3 Interrupt Status，QEP3 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
2	R	0x0	QEP2_INT_STS QEP2 Interrupt Status，QEP2 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号
1	R	0x0	QEP1_INT_STS QEP1 Interrupt Status，QEP1 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号

默认值：0x00000000			Global QEP Interrupt 状态寄存器
位域	类型	默认值	描述
0	R	0x0	QEP0_INT_STS QEP0 Interrupt Status，QEP0 中断状态。 • 0x0：无中断产生 • 0x1：产生中断信号

9.1.9.5. 0x0014 GLB_EPWM_EN

默认值：0x00000000			Global EPWM 使能寄存器
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	GLB_EPWM_EN EPWM Global Enable，EPWM 模块的全局使能，此位置 1 后 EPWM 所有的子模块的功能才启动。 • 0x0：未使能 • 0x1：使能

9.1.9.6. 0x0018 GLB_EPWM_SI0_FLT

默认值：0x00000000			Global EPWM_SI0 输入滤波寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	–
15:8	R/W	0x0	EPWM_SI0_SMP_CTL_PRD EPWM Synchronous Input0 Sample Control Period，同步信号 0 的输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	EPWM_SI0_IN_PRE_FLT

默认值：0x00000000			Global EPWM_SI0 输入滤波寄存器
位域	类型	默认值	描述
			EPWM Synchronous Input0 Pre Filter Enable，同步输入信号 0 的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	EPWM_SI0_IN_FLT_SEL EPWM Synchronous Input0 Filter Seclect，同步信号 0 的输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.9.7. 0x001C GLB_EPWM_SI1_FLT

默认值：0x00000000			Global EPWM_SI1 输入滤波寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	–
15:8	R/W	0x0	EPWM_SI1_SMP_CTL_PRD EPWM Synchronous1 Input Sample Control Period，同步信号 1 的输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	EPWM_SI1_IN_PRE_FLT

默认值：0x00000000			Global EPWM_SI1 输入滤波寄存器
位域	类型	默认值	描述
			EPWM Synchronous1 Input Pre Filter Enable，同步输入信号 1 的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	EPWM_SI1_IN_FLT_SEL EPWM Synchronous1 Input Filter Seclect，同步信号 1 的输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.9.8. 0x0020 GLB_EPWM_CLK_CTL

默认值：0x00000000			Global EPWM 时钟控制寄存器
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	EPWM11_CLK_EN EPWM11 Clock Enable，EPWM11 时钟使能。 • 0x0：未使能 • 0x1：使能
10	R/W	0x0	EPWM10_CLK_EN EPWM10 Clock Enable，EPWM10 时钟使能。 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			Global EPWM 时钟控制寄存器
位域	类型	默认值	描述
9	R/W	0x0	EPWM9_CLK_EN EPWM9 Clock Enable，EPWM9 时钟使能。 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	EPWM8_CLK_EN EPWM8 Clock Enable，EPWM8 时钟使能。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	EPWM7_CLK_EN EPWM7 Clock Enable，EPWM7 时钟使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	EPWM6_CLK_EN EPWM6 Clock Enable，EPWM6 时钟使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	EPWM5_CLK_EN EPWM5 Clock Enable，EPWM5 时钟使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	EPWM4_CLK_EN EPWM4 Clock Enable，EPWM4 时钟使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	EPWM3_CLK_EN EPWM3 Clock Enable，EPWM3 时钟使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	EPWM2_CLK_EN EPWM2 Clock Enable，EPWM2 时钟使能。

默认值：0x00000000			Global EPWM 时钟控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
1	R/W	0x0	EPWM1_CLK_EN EPWM1 Clock Enable，EPWM1 时钟使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	EPWM0_CLK_EN EPWM0 Clock Enable，EPWM0 时钟使能。 • 0x0：未使能 • 0x1：使能

9.1.9.9. 0x0024 GLB_CAP_CLK_CTL

默认值：0x00000000			Global CAP 时钟控制寄存器
位域	类型	默认值	描述
31:6	–	–	–
5	R/W	0x0	CAP5_CLK_EN CAP5 Clock Enable，CAP5 时钟使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	CAP4_CLK_EN CAP4 Clock Enable，CAP4 时钟使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	CAP3_CLK_EN CAP3 Clock Enable，CAP3 时钟使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	CAP2_CLK_EN CAP2 Clock Enable，CAP2 时钟使能。

默认值：0x00000000			Global CAP 时钟控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
1	R/W	0x0	CAP1_CLK_EN CAP1 Clock Enable，CAP1 时钟使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	CAP0_CLK_EN CAP0 Clock Enable，CAP0 时钟使能。 • 0x0：未使能 • 0x1：使能

9.1.9.10. 0x0028 GLB_QEP_CLK_CTL

默认值：0x00000000			Global QEP 时钟控制寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R/W	0x0	QEP9_CLK_EN QEP9 Clock Enable，QEP9 时钟使能。 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	QEP8_CLK_EN QEP8 Clock Enable，QEP8 时钟使能。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	QEP7_CLK_EN QEP7 Clock Enable，QEP7 时钟使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	QEP6_CLK_EN QEP6 Clock Enable，QEP6 时钟使能。

默认值：0x00000000			Global QEP 时钟控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
5	R/W	0x0	QEP5_CLK_EN QEP5 Clock Enable，QEP5 时钟使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	QEP4_CLK_EN QEP4 Clock Enable，QEP4 时钟使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	QEP3_CLK_EN QEP3 Clock Enable，QEP3 时钟使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	QEP2_CLK_EN QEP2 Clock Enable，QEP2 时钟使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	QEP1_CLK_EN QEP1 Clock Enable，QEP1 时钟使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	QEP0_CLK_EN QEP0 Clock Enable，QEP0 时钟使能。 • 0x0：未使能 • 0x1：使能

9.1.9.11. 0x0030 GLB_EPWM_RST_CTL

默认值：0x00000fff			Global EPWM 复位控制寄存器
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x1	EPWM11_RST EPWM11 Reset，EPWM11 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
10	R/W	0x1	EPWM10_RST EPWM10 Reset，EPWM10 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
9	R/W	0x1	EPWM9_RST EPWM9 Reset，EPWM9 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
8	R/W	0x1	EPWM8_RST EPWM8 Reset，EPWM8 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
7	R/W	0x1	EPWM7_RST EPWM7 Reset，EPWM7 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
6	R/W	0x1	EPWM6_RST EPWM6 Reset，EPWM6 复位控制。

默认值：0x00000fff			Global EPWM 复位控制寄存器
位域	类型	默认值	描述
			• 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
5	R/W	0x1	EPWM5_RST EPWM5 Reset，EPWM5 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
4	R/W	0x1	EPWM4_RST EPWM4 Reset，EPWM4 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
3	R/W	0x1	EPWM3_RST EPWM3 Reset，EPWM3 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
2	R/W	0x1	EPWM2_RST EPWM2 Reset，EPWM2 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
1	R/W	0x1	EPWM1_RST EPWM1 Reset，EPWM1 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
0	R/W	0x1	EPWM0_RST EPWM0 Reset，EPWM0 复位控制。

默认值：0x00000fff			Global EPWM 复位控制寄存器
位域	类型	默认值	描述
			• 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。

9.1.9.12. 0x0034 GLB_CAP_RST_CTL

默认值：0x0000003f			Global CAP 复位控制寄存器
位域	类型	默认值	描述
31:6	–	–	–
5	R/W	0x1	CAP5_RST CAP5 Reset，CAP5 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
4	R/W	0x1	CAP4_RST CAP4 Reset，CAP4 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
3	R/W	0x1	CAP3_RST CAP3 Reset，CAP3 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
2	R/W	0x1	CAP2_RST CAP2 Reset，CAP2 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
1	R/W	0x1	CAP1_RST CAP1 Reset，CAP1 复位控制。

默认值：0x0000003f			Global CAP 复位控制寄存器
位域	类型	默认值	描述
			• 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
0	R/W	0x1	CAP0_RST CAP0 Reset，CAP0 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。

9.1.9.13. 0x0038 GLB_QEP_RST_CTL

默认值：0x000003ff			Global QEP 复位控制寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R/W	0x1	QEP9_RST QEP9 Reset，QEP9 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
8	R/W	0x1	QEP8_RST QEP8 Reset，QEP8 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
7	R/W	0x1	QEP7_RST QEP7 Reset，QEP7 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
6	R/W	0x1	QEP6_RST QEP6 Reset，QEP6 复位控制。

默认值：0x000003ff			Global QEP 复位控制寄存器
位域	类型	默认值	描述
			• 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
5	R/W	0x1	QEP5_RST QEP5 Reset，QEP5 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
4	R/W	0x1	QEP4_RST QEP4 Reset，QEP4 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
3	R/W	0x1	QEP3_RST QEP3 Reset，QEP3 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
2	R/W	0x1	QEP2_RST QEP2 Reset，QEP2 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
1	R/W	0x1	QEP1_RST QEP1 Reset，QEP1 复位控制。 • 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。
0	R/W	0x1	QEP0_RST QEP0 Reset，QEP0 复位控制。

默认值：0x000003ff			Global QEP 复位控制寄存器
位域	类型	默认值	描述
			• 0x0：模块复位 • 0x1：放开复位 若需要复位模块，则将此比特置零，使用前将此比特置 1。

9.1.9.14. 0x0050 GLB_EPWM_SYNC_IN_SEL

默认值：0x00000000			Global EPWM 同步输入选择寄存器
位域	类型	默认值	描述
31:29	–	–	–
28	R/W	0x0	CAP0_SYNCO_EN CAP0_SYNCO 输出使能。 • 0x0：未使能 • 0x1：使能
27	R/W	0x0	EPWM9_SYNCO_EN EPWM9_SYNCO 输出使能。 • 0x0：未使能 • 0x1：使能
26	R/W	0x0	EPWM6_SYNCO_EN EPWM6_SYNCO 输出使能。 • 0x0：未使能 • 0x1：使能
25	R/W	0x0	EPWM3_SYNCO_EN EPWM3_SYNCO 输出使能。 • 0x0：未使能 • 0x1：使能
24	R/W	0x0	EPWM0_SYNCO_EN EPWM0_SYNCO 输出使能。 • 0x0：未使能 • 0x1：使能
23:19	–	–	–

默认值：0x00000000			Global EPWM 同步输入选择寄存器
位域	类型	默认值	描述
18:16	R/W	0x0	CAP3_SYNC_IN_SEL， CAP3 Synchronous Input Select，CAP3 同步输入信号源的选择 • 0x0：EPWM_SYNCI0 • 0x1：EPWM_SYNCI1 • 0x2：EPWM0_SYNCO • 0x3：EPWM3_SYNCO • 0x4：EPWM6_SYNCO • 0x5：EPWM9_SYNCO • 0x6：CAP0_SYNCO • 0x7：Reserved
15	–	–	–
14:12	R/W	0x0	CAP0_SYNC_IN_SEL CAP0 Synchronous Input Select，CAP0 同步输入信号源的选择。 • 0x0：EPWM_SYNCI0 • 0x1：EPWM_SYNCI1 • 0x2：EPWM0_SYNCO • 0x3：EPWM3_SYNCO • 0x4：EPWM6_SYNCO • 0x5：EPWM9_SYNCO • others：Reserved
11:9	–	–	–
8:6	R/W	0x0	EPWM9_SYNC_IN_SEL EPWM9 Synchronous Input Select，EPWM9 同步输入信号源的选择。 • 0x0：EPWM_SYNCI0 • 0x1：EPWM_SYNCI1 • 0x2：EPWM0_SYNCO • 0x3：EPWM3_SYNCO • 0x4：EPWM6_SYNCO • others：Reserved
5:4	R/W	0x0	EPWM6_SYNC_IN_SEL

默认值：0x00000000			Global EPWM 同步输入选择寄存器
位域	类型	默认值	描述
			EPWM6 Synchronous Input Select，EPWM6 同步输入信号源的选择。 • 0x0：EPWM_SYNCI0 • 0x1：EPWM_SYNCI1 • 0x2：EPWM0_SYNCO • 0x3：EPWM3_SYNCO
3:2	R/W	0x0	EPWM3_SYNC_IN_SEL EPWM3 Synchronous Input Select，EPWM3 同步输入信号源的选择。 • 0x0：EPWM_SYNCI0 • 0x1：EPWM_SYNCI1 • 0x2：EPWM0_SYNCO • 0x3：Reserved
1:0	R/W	0x0	EPWM0_SYNC_IN_SEL EPWM0 Synchronous Input Select，EPWM0 同步输入信号源的选择。 • 0x0：EPWM_SYNCI0 • 0x1：CPM0_COH • 0x2：CPM1_COH • 0x3：Reserved

9.1.9.15. 0x0054 SYNC_SRC_IN_SEL

默认值：0x00000000			同步信号输入源选择寄存器
位域	类型	默认值	描述
31:14	–	–	–
13:8	R/W	0x0	SYNCI1_IN_SRC_SEL SYNCI1 Input Source Select，SYNCI1 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO

默认值：0x00000000			同步信号输入源选择寄存器
位域	类型	默认值	描述
			• 0x20：CPM0_COH • 0x21：CPM0_COL • 0x22：CPM0_COHL • 0x23：CPM1_COH • 0x24：CPM1_COL • 0x25：CPM1_COHL • 0x30：ECAT_SYNC0 • 0x31：ECAT_SYNC1 • others：Reserved
7:6	–	–	–
5:0	R/W	0x0	SYNCI0_IN_SRC_SEL SYNCI0 Input Source Select，SYNCI0 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x20：CPM0_COH • 0x21：CPM0_COL • 0x22：CPM0_COHL • 0x23：CPM1_COH • 0x24：CPM1_COL • 0x25：CPM1_COHL • 0x30：ECAT_SYNC0 • 0x31：ECAT_SYNC1 • others：Reserved

9.1.9.16. 0x0058 EPWM_FLT_SRC_IN_SEL

默认值：0x00000000			故障输入源选择寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:25	R/W	0x0	EPWM_FALT5_IN_SRC_SEL EPWM_FALT5 Input Source Select，EPWM_FALT5 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved
24:20	R/W	0x0	EPWM_FALT4_IN_SRC_SEL EPWM_FALT4 Input Source Select，EPWM_FALT4 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved

默认值：0x00000000			故障输入源选择寄存器
位域	类型	默认值	描述
19:15	R/W	0x0	EPWM_FALT3_IN_SRC_SEL EPWM_FALT3 Input Source Select，EPWM_FALT3 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved
14:10	R/W	0x0	EPWM_FALT2_IN_SRC_SEL EPWM_FALT2 Input Source Select，EPWM_FALT2 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved
9:5	R/W	0x0	EPWM_FALT1_IN_SRC_SEL EPWM_FALT1 Input Source Select，EPWM_FALT1 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO

默认值：0x00000000			故障输入源选择寄存器
位域	类型	默认值	描述
			• 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved
4:0	R/W	0x0	EPWM_FALT0_IN_SRC_SEL EPWM_FALT0 Input Source Select，EPWM_FALT0 输入信号源选择。 • 0x0：GPIO，即管脚上的信号 • 0x1：EPWM0_TO • 0x2：EPWM1_TO • 0x3：EPWM2_TO • ... • 0xC：EPWM11_TO • 0x11：EPWM0_A • 0x12：EPWM1_A • 0x13：EPWM2_A • ... • 0x1C：EPWM11_A • others：Reserved

9.1.9.17. 0x0060 EPWM_FALT_IN_FLT1

默认值：0x00000000			EPWM 故障输入滤波寄存器 1
位域	类型	默认值	描述
31:24	R/W	0x0	EPWM_FALT1_SMP_CTL_PRD EPWM Fault1 Input Sample Control Period，输入采样周期。

默认值：0x00000000			EPWM 故障输入滤波寄存器 1
位域	类型	默认值	描述
			• 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
23:20	R/W	0x0	EPWM_FALT1_PRE_FLT EPWM Fault1 Input Filter Enable，EPWM故障输入 1 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
19:16	R/W	0x0	EPWM_FALT1_IN_FLT_SEL EPWM Fault1 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
15:8	R/W	0x0	EPWM_FALT0_SMP_CTL_PRD EPWM Fault0 Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK

默认值：0x00000000			EPWM 故障输入滤波寄存器 1
位域	类型	默认值	描述
7:4	R/W	0x0	EPWM_FALT0_PRE_FLT EPWM Fault0 Input Filter Enable，EPWM故障输入 0 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	EPWM_FALT0_IN_FLT_SEL EPWM Fault0 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.9.18. 0x0064 EPWM_FALT_IN_FLT2

默认值：0x00000000			EPWM 故障输入滤波寄存器 2
位域	类型	默认值	描述
31:24	R/W	0x0	EPWM_FALT3_SMP_CTL_PRD EPWM Fault3 Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
23:20	R/W	0x0	EPWM_FALT3_PRE_FLT

默认值：0x00000000			EPWM 故障输入滤波寄存器 2
位域	类型	默认值	描述
			EPWM Fault3 Input Filter Enable，EPWM故障输入 3 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
19:16	R/W	0x0	EPWM_FALT3_IN_FLT_SEL EPWM Fault3 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
15:8	R/W	0x0	EPWM_FALT2_SMP_CTL_PRD EPWM Fault2 Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	EPWM_FALT2_PRE_FLT EPWM Fault2 Input Filter Enable，EPWM故障输入 2 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号

默认值：0x00000000			EPWM 故障输入滤波寄存器 2
位域	类型	默认值	描述
			• … • 0xF：15 采样点同步信号
3:0	R/W	0x0	EPWM_FALT2_IN_FLT_SEL EPWM Fault2 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

9.1.9.19. 0x0068 EPWM_FALT_IN_FLT3

默认值：0x00000000			EPWM 故障输入滤波寄存器 3
位域	类型	默认值	描述
31:24	R/W	0x0	EPWM_FALT5_SMP_CTL_PRD EPWM Fault5 Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
23:20	R/W	0x0	EPWM_FALT5_PRE_FLT EPWM Fault5 Input Filter Enable，EPWM故障输入 5 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号

默认值：0x00000000			EPWM 故障输入滤波寄存器 3
位域	类型	默认值	描述
19:16	R/W	0x0	EPWM_FALT5_IN_FLT_SEL EPWM Fault5 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
15:8	R/W	0x0	EPWM_FALT4_SMP_CTL_PRD EPWM Fault4 Input Sample Control Period，输入采样周期。 • 0x0：1x SYS_CLK • 0x1：2x SYS_CLK • 0x2：4x SYS_CLK • 0x3：6x SYS_CLK • … • 0xFF：510x SYS_CLK
7:4	R/W	0x0	EPWM_FALT4_PRE_FLT EPWM Fault4 Input Filter Enable，EPWM故障输入 4 信号的预滤波使能，预滤波采样周期为 CLK。一般使用，无需使能，默认旁通即可。 • 0x0：旁通，无预滤除效果 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号 • … • 0xF：15 采样点同步信号
3:0	R/W	0x0	EPWM_FALT4_IN_FLT_SEL EPWM Fault4 Input Filter Seclect，输入滤波功能选择。 • 0x0：同步信号，SYS_CLK 对输入信号进行同步 • 0x1：1 采样点同步信号 • 0x2：2 采样点同步信号 • 0x3：3 采样点同步信号

默认值：0x00000000			EPWM 故障输入滤波寄存器 3
位域	类型	默认值	描述
			• … • 0xF：15 采样点同步信号

9.1.9.20. 0x0080 GLB_DLL_CFG

默认值：0x00000000			Global DLL 配置寄存器
位域	类型	默认值	描述
31:30	–	–	–
29	R/W	0x0	REG_DLL1_DCC_EN 占空比自动校准电路使能 • 0x0：未使能 • 0x1：使能
28	R/W	0x0	REG_DLL1_LPC_EN 防错锁电路使能 • 0x0：未使能 • 0x1：使能
27:26	R/W	0x0	REG_ICP1
25:24	R/W	0x0	REG_DLY1
23:22	R/W	0x0	REG_ATB_SEL1
21	R/W	0x0	EN_ATB_SEL1
20	R/W	0x0	EN_LVS1
19	R/W	0x0	EN_CP1
18	R/W	0x0	EN_VCDL1
17	R/W	0x0	EN_CLK1
16	R/W	0x0	DLL1_LDO_EN DLL1 LDO Enable，DLL 1LDO 使能，在使用高精度 PWM 的情况下需要使能此比特。

默认值：0x00000000			Global DLL 配置寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
15:14	–	–	–
13	R/W	0x0	REG_DLL_DCC_EN 占空比自动校准电路使能 • 0x0：未使能 • 0x1：使能
12	R/W	0x0	REG_DLL_LPC_EN 防错锁电路使能 • 0x0：未使能 • 0x1：使能
11:10	R/W	0x0	REG_ICP
9:8	R/W	0x0	REG_DLY
7:6	R/W	0x0	REG_ATB_SEL
5	R/W	0x0	EN_ATB
4	R/W	0x0	EN_LVS，此位需首先使能，然后依次是 EN、EN_CLK、EN_VCDL、EN_CP。每个使能信号依次间隔 100us 打开。
3	R/W	0x0	EN_CP
2	R/W	0x0	EN_VCDL
1	R/W	0x0	EN_CLK
0	R/W	0x0	DLL_LDO_EN DLL LDO Enable，DLL LDO 使能，在使用高精度 PWM 的情况下需要使能此比特。 • 0x0：未使能 • 0x1：使能

9.1.9.21. 0x0088 GLB_FAULT_IN_STS

默认值：0x00000000			FAULT IN 状态寄存器
位域	类型	默认值	描述
31:22	–	–	–
21	R/W	0x0	FALT5_EGE_DET Fault Input 5 Edge Detect，故障输入 5 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
20	R/W	0x0	FALT4_EGE_DET Fault Input 4 Edge Detect，故障输入 4 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
19	R/W	0x0	FALT3_EGE_DET Fault Input 3 Edge Detect，故障输入 3 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
18	R/W	0x0	FALT2_EGE_DET Fault Input 2 Edge Detect，故障输入 2 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
17	R/W	0x0	FALT1_EGE_DET Fault Input 1 Edge Detect，故障输入 1 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
16	R/W	0x0	FALT0_EGE_DET Fault Input 0 Edge Detect，故障输入 0 边沿检测选择。 • 0x0：上升沿 • 0x1：下降沿
15:14	–	–	–
13	R/W1C	0x0	FALT5_STS_LATCH Fault Input 5 Status，故障输入 5 发生了一次的边沿跳变，边沿跳变方向由 FALT5_EGE_DET 决定。

默认值：0x00000000			FAULT IN 状态寄存器
位域	类型	默认值	描述
			• 0x0：无跳变 • 0x1：发生了跳变
12	R/W1C	0x0	FALT4_STS_LATCH Fault Input 4 Status，故障输入 4 发生了一次的边沿跳变，边沿跳变方向由 FALT4_EGE_DET 决定。 • 0x0：无跳变 • 0x1：发生了跳变
11	R/W1C	0x0	FALT3_STS_LATCH Fault Input 3 Status，故障输入 3 发生了一次的边沿跳变，边沿跳变方向由 FALT3_EGE_DET 决定。 • 0x0：无跳变 • 0x1：发生了跳变
10	R/W1C	0x0	FALT2_STS_LATCH Fault Input 2 Status，故障输入 2 发生了一次的边沿跳变，边沿跳变方向由 FALT2_EGE_DET 决定。 • 0x0：无跳变 • 0x1：发生了跳变
9	R/W1C	0x0	FALT1_STS_LATCH Fault Input 1 Status，故障输入 1 发生了一次的边沿跳变，边沿跳变方向由 FALT1_EGE_DET 决定。 • 0x0：无跳变 • 0x1：发生了跳变
8	R/W1C	0x0	FALT0_STS_LATCH Fault Input 0 Status，故障输入 0 发生了一次的边沿跳变，边沿跳变方向由 FALT0_EGE_DET 决定。 • 0x0：无跳变 • 0x1：发生了跳变
7:6	–	–	–
5	R	0x0	FALT5_STS Fault Input 5 Status，故障输入 5 状态 • 0x0：低电平 • 0x1：高电平

默认值：0x00000000			FAULT IN 状态寄存器
位域	类型	默认值	描述
4	R	0x0	FALT4_STS Fault Input 4 Status，故障输入 4 状态 • 0x0：低电平 • 0x1：高电平
3	R	0x0	FALT3_STS Fault Input 3 Status，故障输入 3 状态 • 0x0：低电平 • 0x1：高电平
2	R	0x0	FALT2_STS Fault Input 2 Status，故障输入 2 状态 • 0x0：低电平 • 0x1：高电平
1	R	0x0	FALT1_STS Fault Input 1 Status，故障输入 1 状态 • 0x0：低电平 • 0x1：高电平
0	R	0x0	FALT0_STS Fault Input 0 Status，故障输入 0 状态 • 0x0：低电平 • 0x1：高电平

9.2. CORDIC

CORDIC 数学算术加速模块能够对 Sine、Cosine 等数学计算函数进行加速。对于需要大量数学算术的应用，CORDIC 加速模块能够释放 CPU 更多的资源。

9.2.1. 特性说明

- 支持 Sin、Cos、Arctan、Atan2、Phase、Modulus、Square root、Sinh、Cosh 和 Atanh
- 支持结果直接读取无需轮询或中断

9.2.2. 功能描述

9.2.2.1. CORDIC 函数列表

CORDIC 数学加速模块主要支持下表的数学函数，其中参数 1 和参数 2 为函数的输入参数，结果 1 和结果 2 为函数输出的计算结果，部分函数可以输出两个结果，可以减小冗余的计算。

表 9–3 CORDIC 函数列表

Function	参数 1	参数 2	结果 1	结果 2
Cos 余弦	角度 θ	模 A	A × cos θ	A × sin θ
Sin 正弦	角度 θ	模 A	A × sin θ	A × cos θ
Phase 相位	x	y	Arctan (y ÷ x)	$\sqrt{x^2+y^2}$
Mod 求模	x	y	$\sqrt{x^2+y^2}$	Arctan (y ÷ x)
Arctan 反正切	x	–	Arctan (x)	–
Cosh 双曲余弦	x	–	Cosh (2 × x) ÷ 2	–
Sinh 双曲正弦	x	–	Sinh (2 × x) ÷ 2	–
Atanh 双曲反正切	x	–	Atanh (2 × x) ÷ 2	–
Square root 开方	a	–	$\sqrt{a}$	–
Atan2	x	y	Atanh (y,x)	$\sqrt{x^2+y^2}$

9.2.2.2. Cosine 函数

Cosine 函数的输入输出参数如下表：

表 9–4 Cosine 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	角度 θ（弧度，除 π）	[–1, 0.999969482]
参数 2	16 位定点数 Q15	模 A	[0, 0.999969482]
结果 1	16 位定点数 Q15	A × cos (θ × π)	[–0.999969482, 0.999969482]
结果 2	16 位定点数 Q15	A × sin (θ × π)	[–0.999969482, 0.999969482]

9.2.2.3. Sine 函数

Sine 函数的输入输出参数如下表：

表 9–5 Sine 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	角度 θ（弧度，除 π）	[–1, 0.999969482]
参数 2	16 位定点数 Q15	模 A	[0, 0.999969482]
结果 1	16 位定点数 Q15	A × sin (θ × π)	[–0.999969482, 0.999969482]
结果 2	16 位定点数 Q15	A × cos (θ × π)	[–0.999969482, 0.999969482]

9.2.2.4. Phase 函数

Phase 函数的输入输出参数如下表：

表 9–6 Phase 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–1, 0.999969482]
参数 2	16 位定点数 Q15	y	[–1, 0.999969482]
结果 1	16 位定点数 Q15	arctan(y/x)（弧度，除 π）	[–0.5, 0.5]
结果 2	16 位定点数 Q15	$sqrt(x^2 + y^2)$	[0, 0.999969482]

9.2.2.5. Mod 函数

Mod 函数的输入输出参数如下表：

表 9–7 Mod 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–1, 0.999969482]
参数 2	16 位定点数 Q15	y	[–1, 0.999969482]
结果 1	16 位定点数 Q15	$sqrt(x^2 + y^2)$	[0, 0.999969482]
结果 2	16 位定点数 Q15	arctan(y/x)（弧度，除 π）	[–0.5, 0.5]

9.2.2.6. Arctan 函数

Arctan 函数的输入输出参数如下表：

表 9–8 Arctan 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–1, 0. 999969482]
参数 2	16 位定点数 Q15	–	–
结果 1	16 位定点数 Q15	arctan(128 × x)（弧度，除 π）	[–0.5, 0.5]
结果 2	–	–	–

为了统一输入参数的格式，Arctan 函数输入参数为 x，采用 Q15 的格式，范围为[–1, 0.999969482]，Arctan 实施的运算为 Arctan(128 × x)。以计算 64 的反正切值为例，输入参数 x = 64 / 128 = 0.5（128 为正切值的最大值），只要将 x = 0.5 转换成 Q15 格式写入输入参数 1 的寄存器即可。

9.2.2.7. Atan2 函数

Atan2 函数的输入输出参数如下表：

表 9–9 Atan2 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–1, 0.999969482]
参数 2	16 位定点数 Q15	y	[–1, 0.999969482]
结果 1	16 位定点数 Q15	atan2(y ÷ x)（弧度，除 π）	[–1, 0.999969482]
结果 2	16 位定点数 Q15	sqrt($x^2 + y^2$)	[0, 0.999969482]

9.2.2.8. Sinh 函数

Sinh 函数的输入输出参数如下表：

表 9–10 Sinh 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–0.559, 0.559]
参数 2	–	–	–
结果 1	16 位定点数 Q15	sinh(2 × x) / 2	[–0.683, 0.683]

表 9–10 Sinh 函数的输入输出参数 (续)

参数	数据格式	描述	范围
结果 2	16 位定点数 Q15	cosh(2 × x) / 2	[0.5, 0.846]

9.2.2.9. Cosh 函数

Cosh 函数的输入输出参数如下表：

表 9–11 Cosh 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–0.559, 0.559]
参数 2	–	–	–
结果 1	16 位定点数 Q15	cosh(2 × x) / 2	[0.5, 0.846]
结果 2	16 位定点数 Q15	sinh(2 × x) / 2	[–0.683, 0.683]

9.2.2.10. Atanh 函数

Atanh 函数的输入输出参数如下表：

表 9–12 Atanh 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	x	[–0.403, 0.403]
参数 2	–	–	–
结果 1	16 位定点数 Q15	Atanh(2 × x) / 2	[–0.559, 0.559]
结果 2	–	–	–

9.2.2.11. Square Root 函数

Square Root 函数的输入输出参数如下表：

表 9–13 Square Root 函数的输入输出参数

参数	数据格式	描述	范围
参数 1	16 位定点数 Q15	a	[0.027, 0.75]

表 9–13 Square Root 函数的输入输出参数 (续)

参数	数据格式	描述	范围
参数 2	–	–	–
结果 1	16 位定点数 Q15	sqrt(a)	[0.164, 0.866]
结果 2	–	–	–

9.2.2.12. 输入输出格式

CORDIC 模块的函数的输入为定点格式 Q1.15 (Q15)，输出可通过寄存器 COR_OUT_FMT 选择定点格式 Q15 或定点格式 Q24。Q1.15 定点格式，代表 1 bit 的符号位，15 bits 的小数位，数值表示范围是 $-1 \sim 1-2^{-15}$。

- 如只需一个输入参数，只需配置寄存器 COR_IN1 即可。
- 如需两个输入参数，需先配置寄存器 IN_PARA2，再配置寄存器 IN_PARA1。

  当输出包含两个结果，则寄存器 OUT_RES1 为输出结果 1，OUT_RES2 为输出结果 2。

9.2.2.13. CORDIC 操作

软件进行 CORDIC 的操作步骤如下：

1. 在 COR_CFG 寄存器中，选择功能函数以及迭代次数。
2. 若需两个输入参数的，则先写入 IN_PARA2，再写入 IN_PARA1（写入 IN_PARA1 后自动触发一次 CORDIC 的计算）。
3. 软件可以直接读取结果寄存器，无需读取 CORDIC 的工作状态。

9.2.2.14. 中断

CORIDC 模块返回的结果支持使用中断的方式读取，当 CORDIC 模块计算完成，产生中断，通知 CPU 读取结果。

9.2.3. 寄存器列表

表 9–14 CORDIC 寄存器列表

偏移	缩写	寄存器名称	章节
0x0000	COR_CTL	CORDIC 控制寄存器	0x0000 COR_CTL
0x0004	COR_STS	CORDIC 状态寄存器	0x0004 COR_STS
0x0008	COR_INT_EN	CORDIC 中断使能寄存器	0x0008 COR_INT_EN
0x000C	COR_CFG	CORDIC 配置寄存器	0x000C COR_CFG

表 9–14 CORDIC 寄存器列表 (续)

偏移	缩写	寄存器名称	章节
0x0010	COR_IN1	CORDIC 输入参数 1 寄存器	0x0010 COR_IN1
0x0014	COR_IN2	CORDIC 输入参数 2 寄存器	0x0014 COR_IN2
0x0030	COR_OUT1	CORDIC 输出结果 1 寄存器	0x0030 COR_OUT1
0x0034	COR_OUT2	CORDIC 输出结果 2 寄存器	0x0034 COR_OUT2
0x0FFC	COR_VER	CORDIC 模块版本寄存器	0x0FFC COR_VER

9.2.4. 寄存器描述

9.2.4.1. 0x0000 COR_CTL

默认值：0x00000000			CORDIC 控制寄存器
位域	类型	默认值	描述
31:2	–	–	–
1	R/WAC	0x0	COR_FSM_CLR CORDIC FSM Clear，CORDIC 状态清除，回到 IDLE 状态。 • 0x0：无操作 • 0x1：进行 FSM 状态清除，FSM 回到 IDLE 状态
0	–	–	–

9.2.4.2. 0x0004 COR_STS

默认值：0x00000000			CORDIC 状态寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:4	R	0x0	COR_FSM CORDIC FSM，CORDIC FSM 状态信息，用于 Debug。
3:1	–	–	–

默认值：0x00000000			CORDIC 状态寄存器
位域	类型	默认值	描述
0	R/WAC	0x0	RDY_FLG Result Ready Flag，计算结果完成标志。 • 0x0：没有新的计算结果输出 • 0x1：已完成计算，产生新的计算结果

9.2.4.3. 0x0008 COR_INT_EN

默认值：0x00000000			CORDIC 中断使能寄存器
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	RDY_IEN Result Ready Interrup Enable，计算结果完成中断使能。 • 0x0：未使能 • 0x1：使能

9.2.4.4. 0x000C COR_CFG

默认值：0x00010700			CORDIC 配置寄存器
位域	类型	默认值	描述
31:20	–	–	–
19	R/W	0x0	COR_OUT_FMT Cordic Output Format，Cordic 输出格式 • 0x0：Q15 • 0x1：Q24
18:17	–	–	–
16	R/W	0x1	RD_AUTO_CLR Read Result Auto Clear Result Ready Flag，读取计算结果自动清除 RDY_FLG。 • 0x0：不自动清除标志位 RDY_FLG，需要软件清除 • 0x1：自动清除标志位 RDY_FLG

默认值：0x00010700			CORDIC 配置寄存器
位域	类型	默认值	描述
15:12	–	–	–
11:8	R/W	0xC	NM_ITER Number of iterations，迭代次数。 • 0x0：0 • 0x1：2 • 0x2：4 • 0x3：6 • ... • 0xC：24
7:4	–	–	–
3:0	R/W	0x0	FUNC_SEL Function Select，函数选择。 • 0x0：Cosine 余弦函数 • 0x1：Sine 正弦函数 • 0x2：Phase 相位函数 • 0x3：Modulus 模函数 • 0x4：Arctangent 反正切函数 • 0x5：Cosh 双曲正弦函数 • 0x6：Sinh 双曲余弦函数 • 0x7：Atanh 双曲反正切函数 • 0x8：Square Root 开方函数 • 0x9：Arctan2 • 0xa：Modulus 模函数 • others：Reserved

9.2.4.5. 0x0010 COR_IN1

默认值：0x00000000			CORDIC 输入参数 1 寄存器
位域	类型	默认值	描述
31:16	–	–	–

默认值：0x00000000			CORDIC 输入参数 1 寄存器
位域	类型	默认值	描述
15:0	R/W	0x0	IN_PARA1 Input Parameter 1，输入参数 1，软件写入参数 1 后，将自动触发 CORDIC 模块进行一次算术计算。

9.2.4.6. 0x0014 COR_IN2

默认值：0x00007FFF			CORDIC 输入参数 2 寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x7FFF	IN_PARA2 input Parameter 2，输入参数 2。

9.2.4.7. 0x0030 COR_OUT1

默认值：0x00000000			CORDIC 输出结果 1 寄存器
位域	类型	默认值	描述
31:0	R	0x0	OUT_RES1 Output Result 1，输出结果 1 • 若输出为 Q15 格式，直接读取 RES1[15:0]，获得结果（ 1 bit 的符号位，15 bits 的小数位 ） • 若输出为 Q24 格式，直接读取 RES1[24:0]，获得结果（ 1 bit 的符号位，24 bits 的小数位 ）

9.2.4.8. 0x0034 COR_OUT2

默认值：0x00000000			CORDIC 输出结果 2 寄存器
位域	类型	默认值	描述
31:0	R	0x0	OUT_RES2 Output Result 2，输出结果 2。

默认值：0x00000000			CORDIC 输出结果 2 寄存器
位域	类型	默认值	描述
			1. 若输出为 Q15 格式，直接读取 RES1[15:0]，获得结果（ 1 bit 的符号位，15 bits 的小数位 ） 2. 若输出为 Q24 格式，直接读取 RES1[24:0]，获得结果（ 1 bit 的符号位，24 bits 的小数位 ）

9.2.4.9. 0x0FFC COR_VER

默认值：0x00000101			CORDIC 模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x00000101	版本寄存器 V1.1

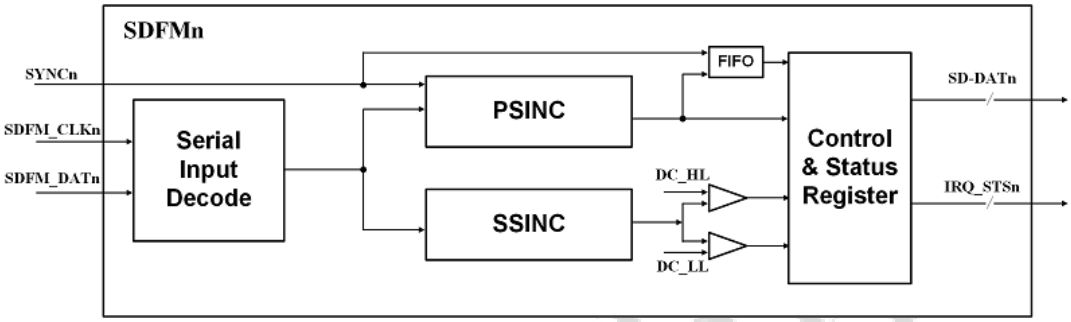
9.3. Sigma Delta Filter Module (SDFM)

SDFM，全称 Sigma Delta Filter Module，用于对 1-bit Sigma-Delta 调制的信号进行低通数字处理，进而还原真实信号。常见的比如隔离式 ADC，需要通过 SDFM 进行低通滤波还原获得数字信号，提供给 MCU 进行信号处理。

9.3.1. 特性说明

- 外部输入信号 SDIN_DATn、输入输出信号SDIN_CLKn（ n = 0/1/2/3 ）
- 对输入信号支持 4 种解码模式
- 每个 SDFM 模块内部包含两路滤波器：
  - PSINC：解析的数据用于用户算法计算
  - SSINC：解析的数据用于内部比较，产生触发信号，触发 EPWM 输出保护
- PSINC 支持 OSR 配置(1~256)，滤波类型配置 (Sinc1/ Sinc2/ Sinc3)
- PSINC 输出为 16 bits/ 32 bits
- SSINC 支持 OSR 配置 (1~32)，滤波类型配置 (Sinc1/ Sinc2/ Sinc3)
- SSINC 支持两组高低双阈值以及 Z 阈值比较
- 支持 FIFO 模式（ 64 x 32-bit ）

9.3.2. 原理框图



- Serial Input Decode 模块：用于解析串行数据。
- PSINC 滤波器模块：即 Primary Sinc Filter，主滤波器用于串行数据的数字处理。
- SSINC 滤波器模块：即 Secondary Sinc Filter，次滤波器用于快速比较，产生比较信号。
- SYNCn 信号来自 EPWM 的同步信号。

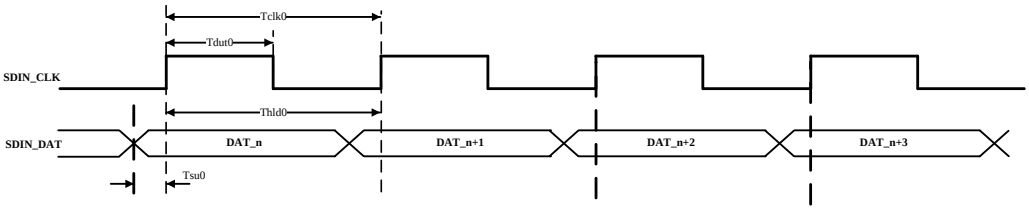
9.3.3. 功能描述

9.3.3.1. 串行输入解码

SDFM 支持 4 种串行输入解析方式，可以根据外部连接器件，配置寄存器以适配对应器件的工作模式。

Mode0

Mode0 支持的串行输入信号时序如下，SDIN_CLK 的每个上升沿进行一次 SDIN_DAT 的数据读取，SDIN_CLK 的频率与 SDIN_DAT 的速率一致。



Mode0 的输入时序要求如下表所示：

- SYSCLK 典型值设置为 240 MHz。
- TSYSCLK 为 SYSCLK 的周期。

表 9–15 Mode0 的输入时序要求

参数	描述	MIN	TYP	MAX	UNIT
Tclk0	Mode0 的输入时钟频率	40	–	256 * TSYSCLK	ns
Tdut0	Mode0 的输入时钟的高占空比	10	–	Tclk0 – 10	ns

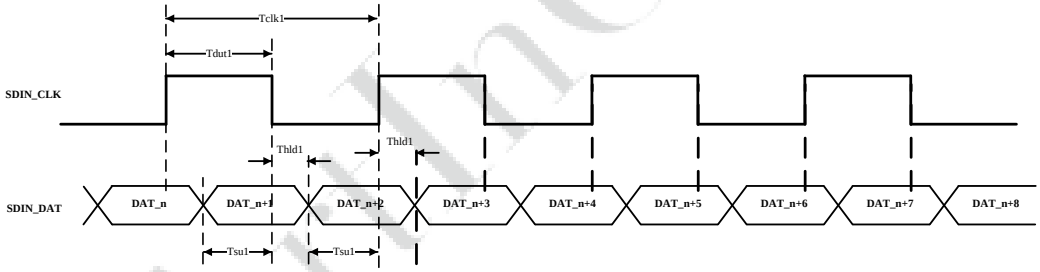
表 9–15 Mode0 的输入时序要求 (续)

参数	描述	MIN	TYP	MAX	UNIT
Tsu0	Mode0 的数据建立时间	5	–	–	ns
Thld0	Mode0 的数据保持时间	5	–	–	ns

Mode1

Mode1 支持的串行输入信号时序如下图所示

- SDIN_CLK 的上升沿或下降沿均进行一次 SDIN_DAT 的数据读取。
- SDIN_CLK 的频率是 SDIN_DAT 的速率的 1/2 倍。



Mode1 的输入时序要求如下表所示：

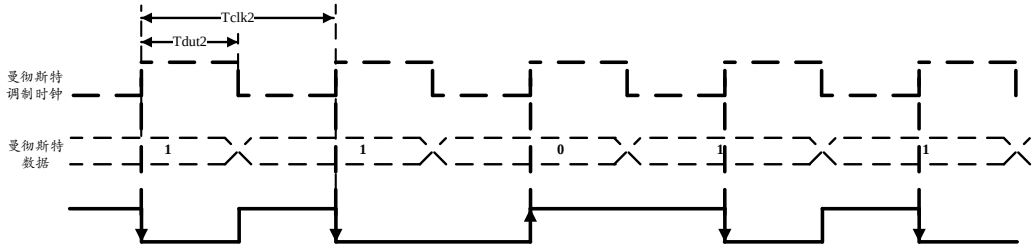
表 9–16 Mode1 的输入时序要求

参数	描述	MIN	TYP	MAX	UNIT
Tclk1	Mode1 的输入时钟频率	80	–	256 * TSYSCLK	ns
Tdut1	Mode1 的输入时钟的高占空比	10	–	Tclk0 – 10	ns
Tsu1	Mode1 的数据建立时间	5	–	–	ns
Thld1	Mode1 的数据保持时间	5	–	–	ns

Mode2

Mode2 支持的串行输入信号时序如下图所示：

- 串行输入信号 SDIN_DAT 为曼彻斯特信号，无需 SDIN_CLK 输入。
- SDFM 内部硬件自动恢复时钟。



Mode2 的输入时序要求如下表所示：

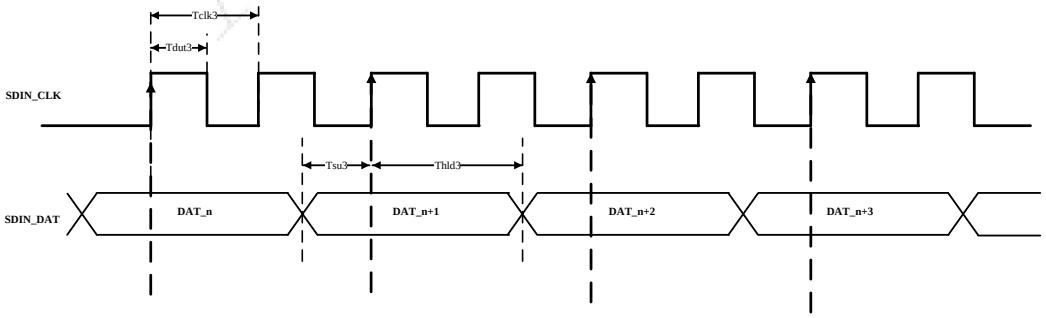
表 9–17 Mode2 的输入时序要求

参数	描述	MIN	TYP	MAX	UNIT
Tclk2	Mode2 的输入时钟频率	8*SYSCLK	–	20*SYSCLK	ns
Tdut2	Mode2 的输入时钟的高占空比	10	–	–	ns

Mode3

Mode3 支持的串行输入信号时序如下图所示：

- SDIN_CLK 的每隔一个上升沿进行一次的 SDIN_DAT 的数据读取。
- SDIN_CLK 的频率是 SDIN_DAT 的速率的 2 倍。



Mode3 的输入时序要求如下表所示：

表 9–18 Mode3 的输入时序要求

参数	描述	MIN	TYP	MAX	UNIT
Tclk3	Mode3 的输入时钟频率	40	–	256 * TSYSCLK	ns
Tdut3	Mode3 的输入时钟的高占空比	10	–	Tclk3 – 5	ns
Tsu3	Mode3 的数据建立时间	5	–	–	ns
Thld3	Mode3 的数据保持时间	5	–	–	ns

9.3.3.2. SINC 滤波器

SDFM 内置下列两套 SINC 滤波器，即 PSINC 和 SSINC：

- PSINC，Primary SINC 滤波器，用于精确数据计算
- SSINC，Secondary SINC 滤波器，用于数据比较

SINC 滤波器实现的功能框图如下：

- N：SINC 滤波器阶数，可通过寄存器配置为 1/2/3
- OSR：过采样率，可通过寄存器配置
- PSINC：配置 N = 1/2/3，OSR = 1/2/3/…/256
- SSINC：配置 N = 1/2/3，OSR = 1/2/3/…/32

滤波延迟

SINC 滤波器的延迟时间的计算公式如下：

T_latency = (N * OSR) / R_mod

- N：SINC 滤波器阶数，可通过寄存器配置为 1/2/3
- OSR：过采样率，可通过寄存器配置
- R_mod：串行输入调制数据的速率

假设 SINC 滤波器的阶数 N = 3，过采样率 OSR = 128，输入信号的速率为 20 MHz，则 T_lantency = 19.2 us，表示 SINC 滤波器的输出延迟为 19.2 us。

9.3.3.2.1. PSINC 输出

PSINC 输入输出具有以下特点：

- 低电平输入代表 –1
- 高电平输入代表 +1
- 采用互补的二进制格式

下表展示了 SINC1/ 2/ 3 在不同 OSR 模式下滤波器计算过程中的数据范围，其中：

- 输出最大的是 SINC3 滤波器和 OSR = 256 的情况下，需要 25 bits 的有符号整型数，数据范围是 $-2^{24} \sim +2^{24}$ 。

OSR	SINC1	SINC2	SINC3
x	–x ~ x	$-x^2 \sim x^2$	$-x^3 \sim x^3$
4	–4 ~ 4	–16 ~ 16	–64 ~ 64

OSR	SINC1	SINC2	SINC3
8	–8 ~ 8	–64 ~ 64	–512 ~ 512
16	–16 ~ 16	–256 ~ 256	–4096 ~ 4096
32	–32 ~ 32	–1024 ~ 1024	–32768 ~ 32768
64	–64 ~ 64	–4096 ~ 4096	–262144 ~ 262144
128	–128 ~ 128	–16384 ~ 16384	–2097152 ~ 2097152
256	–256 ~ 256	–65536 ~ 65536	–16777216 ~ 16777216

PSINC 的输出支持配置以下输出场景：

- 32–bit 输出：无需对过程结果数据进行任何操作，因为 32–bit 位宽完全满足过程数据最大 25–bit 的需求。
- 16–bit 输出：由于过程数据输出的范围可能超过 16–bit 可表示的范围，即超过 -2^{15}~$(2^{15}-1)$，所以需要对过程数据进行右移，从而保证输出结果有效。推荐的数据右移配置如下表所示：

OSR	SINC1	SINC2	SINC3
1~31	0	0	0
32~40	0	0	1
41~50	0	0	2
51~63	0	0	3
64~80	0	0	4
81~101	0	0	5
102~127	0	0	6
128~161	0	0	7
162~181	0	0	8
182~203	0	1	8
204~255	0	1	9
256	0	2	9

9.3.3.2.2. PSINC 同步

SDFM 模块的 PSINC 滤波器支持同步信号。当同步信号生效，SINC 滤波器重新开始进行计算。同步机制，可以保证两个或多个 SDFM 的 PSINC 同步开始计算，达到数据处理的时刻对齐。

基于 SINC 的工作原理，在 SINC 使能开始工作的时刻以及同步信号生效的时刻，SINC 输出存在若干个不正确的数据点。具体如下：

- SINC1（N=1）滤波器，任何时刻都无错误数据输出。
- SINC2（N=2）滤波器，输出的起始存在第一个数据点错误。
- SINC3（N=3）滤波器，输出的起始存在前两个数据点错误。

由于 SDFM 输出数据时，会自动丢弃起始的错误数据点，因此在上述 N=2 和 N=3 的情况下，输出延迟会相应的增加所丢弃的数据点的转换时间。

9.3.3.2.3. SSINC 输出

SSINC 输出具有以下特点：

- 低输入代表 0
- 高输入代表 1
- 采用 16–bit 无符号数据表示

下表展示了 SINC1/ 2/ 3 在不同 OSR 模式下滤波器计算过程中的数据范围：

OSR	SINC1	SINC2	SINC3
x	0 ~ x	0 ~ x^2	0 ~ x^3
4	0 ~ 4	0 ~ 16	0 ~ 64
8	0 ~ 8	0 ~ 64	0 ~ 512
16	0 ~ 16	0 ~ 256	0 ~ 4096
32	0 ~ 32	0 ~ 1024	0 ~ 32768

9.3.3.2.4. SSINC 高低阈值比较

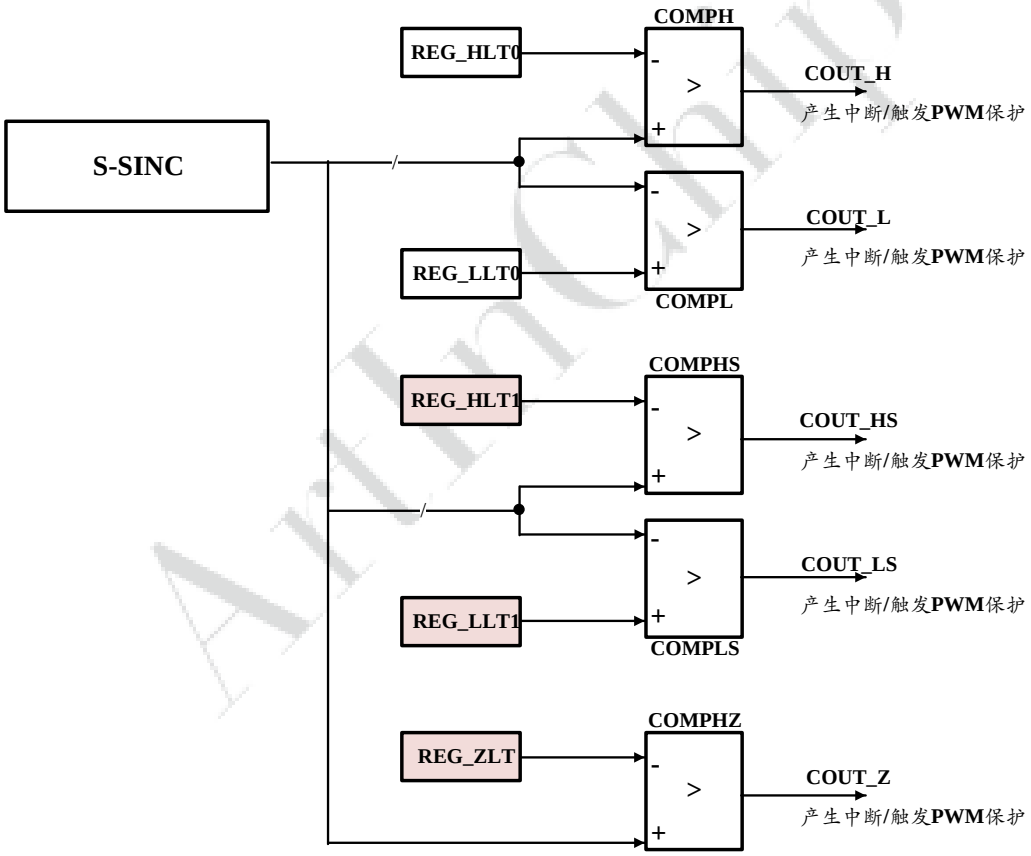
SSINC 的输出可以进行高低阈值的比较，高低阈值可以通过寄存器设置。如下图所所示：

- 对于高位比较器：
  - 当 SSINC 滤波器的输出结果大于高阈值 时，高位比较器的输出信号 则为高电平。
  - 当 SSINC 滤波器的输出结果低于高阈值 时，高位比较器的输出信号 则为低电平。
- 对于低位比较器：

- 当 SSINC 滤波器的输出结果小于低阈值 时，低位比较器的输出信号 则为高电平。
- 当 SSINC 滤波器的输出结果高于低阈值 时，低位比较器的输出信号 则为低电平。

• 输出信号 可以作为触发 EPWM 保护的信号或者产生中断的信号。

如下图所示，共有三个高位比较器，高阈值分别为 HLT0，HLT1 以及 ZLT；两个低位比较器，低阈值分别为 LLT0 和 LLT1。



如需监控电压，可分别设置高低阈值 ，对输入电压进行过压以及欠压检测。

9.3.3.3. 中断信号

SDFM 支持中断，对应有以下的中断源：

- COMP_L_FLG：低阈值状态标记，表示 SSINC 滤波器的输出值小于低阈值。
- COMP_H_FLG：高阈值状态标记，表示 SSINC 滤波器的输出值大于高阈值。
- MOD_FAIL_FLG：调制器异常状态标记，表示在默认的无时钟异常阈值时间内，SDIN_CLK 不再翻转。

  无时钟异常阈值可通过 SDIN_CLK_THR 配置，默认值为 64，单位为 SYSCLK。

  调制器异常状态标记属于异常状态。
- DAT_DONE_FLG：数据完成中断状态标记，表示 PSINC 完成一次数据生成。

  此状态可以通过写 1 清 0。
- FIFO_FLG：FIFO 状态标记，指示 FIFO 接收到有效数据（即 FIFO 的数据多于阈值 FIFO_RXTH）时产生的标记，此时也会相应的产生 FIFO_DRQ 信号。

- FIFO_OF_FLG：FIFO 出现上溢出状态标记。
- FIFO_EMPTY_FLG：FIFO 空状态标记。
- COMP_LS_FLG：低阈值状态标记，表示 SSINC 滤波器的输出值小于低阈值。
- COMP_HS_FLG：高阈值状态标记，表示 SSINC 滤波器的输出值大于高阈值。
- COMP_Z_FLG：Z 阈值状态标记，表示 SSINC 滤波器的输出值大于阈值 ZLT。

9.3.3.4. FIFO 模式

SDFM 模块支持 FIFO 模式，内置 64 * 32–bit FIFO，支持 DMA 传输。用户可通过寄存器配置 FIFO 模式：

- 配置 FIFO 的阈值 FIFO_RXTH：当 FIFO 数据大于 FIFO_RXTH 则产生中断状态 FLG，此中断状态告知 CPU 进行数据的读取。
- 在 FIFO 模式下，如使能 SYNC 功能，则只有在 FIFO 检测到 SYNC 同步信号后，才开始进行存入数据。如未使能 SYNC 功能，则只要滤波器输出有效数据，即存入 FIFO。

  在使能了 SYNC 功能后，可以选择是否在 FIFO 数据达到阈值时，清除 FIFO 数据，默认不清除，保持当前采集到的数据。

9.3.3.5. 时钟输出

为了节省硬件的管脚资源，通过寄存器配置，SDFM0/1/2/3 可以选择外部的管脚输入或者内部的时钟分频模块，如图 9–32 : SDFM 时钟信号所示：

- CLK_DIV 模块采用 SDFM 的系统时钟作为时钟信号源。
- 使能 CLK_DIV 模块后，CLK_DIV 的输出时钟信号除了提供给内部 SDFM 之外，同时也可以通过 SD_CLK0/1/2/3管脚提供给外部器件。

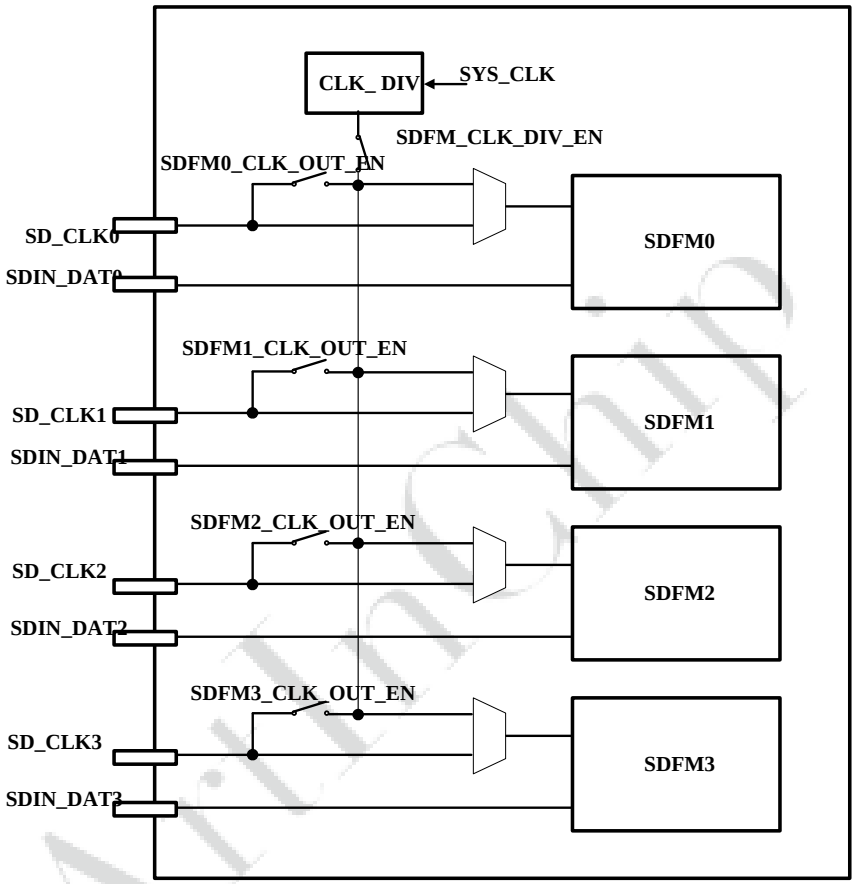
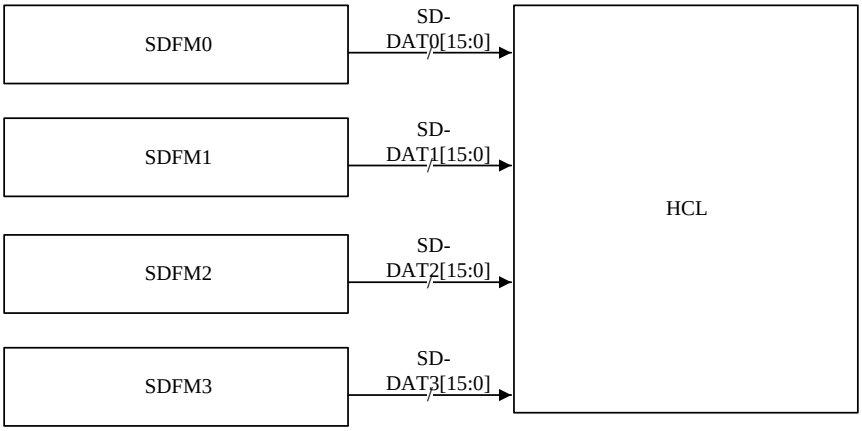


图 9–32 SDFM 时钟信号

9.3.3.6. SDFM 结果输出 HCL

HCL 算法计算时需要使用 SDFM0/1/2/3 的数据，分别作为 U 相的电流数据和 V 相的电流数据。

在 SDFM 的数据作为 HCL 电流数据源的情况下，需要配置 SDFM 输出为 16–bit 的数据，HCL 将 SDFM 的结果数据（即寄存器 SD_DATn 的低 16 bits）以 Q15 的格式进行读取。



9.3.4. 寄存器列表

表 9–19 地址映射

模块名	基地址	地址空间
GLB	GB（SDFM_BASE）	64 B
SDFM0	SD0（SDFM_BASE + 0x40）	64 B
SDFM1	SD1（SDFM_BASE + 0x80）	64 B
SDFM2	SD2（SDFM_BASE + 0xC0）	64 B
SDFM3	SD3（SDFM_BASE + 0x100）	64 B

表 9–20 SDFM 寄存器列表

偏移	缩写	寄存器名称	章节
GLB 寄存器			
0x0000	GLB_EN	全局使能寄存器	0x0000 GLB_EN
0x0004	GLB_IRQ_STS	全局中断状态寄存器	0x0004 GLB_IRQ_STS
0x0008	GLB_SDFM_CLK_SET	SDFM 时钟配置寄存器	0x0008 GLB_SDFM_CLK_SET
0x0010	GLB_SDFM_SMP_SET	SDFM 采样时刻配置寄存器	0x0010 GLB_SDFM_SMP_SET
SDFM 寄存器			
0x0000	SDFM_FLG	SDFM 标志寄存器	SDn + 0x00 SDFM_FLG
0x0004	SDFM_INT_EN	SDFM 中断使能寄存器	SDn + 0x04 SDFM_INT_EN
0x0008	SDFM_CTL	SDFM 控制寄存器	SDn + 0x08 SDFM_CTL
0x000C	PSINC_OUT	PSINC 输出寄存器	SDn + 0x0C PSINC_OUT
0x0010	PSINC_SET	PSINC 设置寄存器	SDn + 0x10 PSINC_SET
0x0014	SSINC_TH	SSINC 比较阈值寄存器	SDn + 0x14 SSINC_TH
0x0018	SSINC_SET	SSINC 设置寄存器	SDn + 0x18 SSINC_SET
0x001C	SD_DAT	SINC 滤波数据寄存器	SDn + 0x1C SD_DAT

表 9–20 SDFM 寄存器列表 (续)

偏移	缩写	寄存器名称	章节
0x0020	FIFO_CTL	FIFO 控制寄存器	SDn + 0x20 FIFO_CTL
0x0028	FIFO_DAT	FIFO 数据寄存器	SDn + 0x28 FIFO_DAT
0x0030	SSINC_TH1	SSINC 比较阈值 1 寄存器	SDn + 0x30 SSINC_TH1
0x0034	SSINC_ZLT	SSINC ZLT 阈值寄存器	SDn + 0x34 SSINC_ZLT
版本寄存器			
0xFFC	VERSION	SDFM 模块的版本寄存器	0x0FFC SDFM_VER

9.3.5. 寄存器描述

9.3.5.1. 地址映射

SDFM 模块最多支持 4 套 SDFM 接口，即 SDFM0/1/2/3，对应的寄存器地址映射关系如下。GLB 模块为对 SDFM0/1/2/3 的全局控制。

模块名	基地址	地址空间
GLB	GB (SDFM_BASE)	64 B
SDFM0	SD0 (SDFM_BASE + 0x40)	64 B
SDFM1	SD1 (SDFM_BASE + 0x80)	64 B
SDFM2	SD2 (SDFM_BASE + 0xC0)	64 B
SDFM3	SD3 (SDFM_BASE + 0x100)	64 B

9.3.5.2. 0x0000 GLB_EN

默认值：0x00000000			全局使能寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	SDFM3_EN SDFM3 Enable，SDFM3 使能。

默认值：0x00000000			全局使能寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
2	R/W	0x0	SDFM2_EN SDFM2 Enable，SDFM2 使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	SDFM1_EN SDFM1 Enable，SDFM1 使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	SDFM0_EN SDFM0 Enable，SDFM0 使能。 • 0x0：未使能 • 0x1：使能

9.3.5.3. 0x0004 GLB_IRQ_STS

默认值：0x00000000			全局中断状态寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	SDFM3_IRQ_STS SDFM3 IRQ Status，SDFM3 的中断状态。 • 0x0：无中断状态产生 • 0x1：出现中断状态
2	R/W	0x0	SDFM2_IRQ_STS SDFM2 IRQ Status，SDFM2 的中断状态。 • 0x0：无中断状态产生 • 0x1：出现中断状态
1	R/W	0x0	SDFM1_IRQ_STS SDFM1 IRQ Status，SDFM1 的中断状态。

默认值：0x00000000			全局中断状态寄存器
位域	类型	默认值	描述
			• 0x0：无中断状态产生 • 0x1：出现中断状态
0	R/W	0x0	SDFM0_IRQ_STS SDFM0 IRQ Status，SDFM0 的中断状态。 • 0x0：无中断状态产生 • 0x1：出现中断状态

9.3.5.4. 0x0008 GLB_SDFM_CLK_SET

默认值：0x00000000			SDFM 时钟配置寄存器
位域	类型	默认值	描述
31:28	–	–	–
27	R/W	0x0	SDFM3_CLK_IN_EN SDFM3 Clock Input Enable，SDFM3 时钟输入使能。 • 0x0：未使能，SDFM3_CLK 为输出 IO • 0x1：使能，SDFM3_CLK 为输入 IO
26	R/W	0x0	SDFM2_CLK_IN_EN SDFM2 Clock Input Enable，SDFM2 时钟输入使能。 • 0x0：未使能，SDFM2_CLK 为输出 IO • 0x1：使能，SDFM2_CLK 为输入 IO
25	R/W	0x0	SDFM1_CLK_IN_EN SDFM1 Clock Input Enable，SDFM1 时钟输入使能。 • 0x0：未使能，SDFM1_CLK 为输出 IO • 0x1：使能，SDFM1_CLK 为输入 IO
24	R/W	0x0	SDFM0_CLK_IN_EN SDFM0 Clock Input Enable，SDFM0 时钟输入使能。 • 0x0：未使能，SDFM0_CLK 为输出 IO • 0x1：使能，SDFM0_CLK 为输入 IO
23:16	R/W	0x0	SDFM_CLK_DIV

默认值：0x00000000			SDFM 时钟配置寄存器
位域	类型	默认值	描述
			SDFM Clock Divisor，SDFM 时钟分频系数。需先使能 SDFM_CLK_DIV_EN 后此位才能生效。 • 0x0：除 1 • 0x1：除 2 • 0x2：除 3 • ... • 0xFF：除 256
15:9	–	–	–
8	R/W	0x0	SDFM_CLK_DIV_EN SDFM Clock Divisor Enable，SDFM 时钟分频使能，此位使能后 SDFM_CLK_DIV 生效。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	SDFM3_CLK_OUT_EN SDFM3 Clock Output Enable，SDFM3 时钟输出使能。 • 0x0：未使能，SDFM3_CLK 输入时钟信号 • 0x1：使能，SDFM3_CLK 输出时钟信号
6	R/W	0x0	SDFM2_CLK_OUT_EN SDFM2 Clock Output Enable，SDFM2 时钟输出使能。 • 0x0：未使能，SDFM2_CLK 输入时钟信号 • 0x1：使能，SDFM2_CLK 输出时钟信号
5	R/W	0x0	SDFM1_CLK_OUT_EN SDFM1 Clock Output Enable，SDFM1 时钟输出使能。 • 0x0：未使能，SDFM1_CLK 输入时钟信号 • 0x1：使能，SDFM1_CLK 输出时钟信号
4	R/W	0x0	SDFM0_CLK_OUT_EN SDFM0 Clock Output Enable，SDFM0 时钟输出使能。 • 0x0：未使能，SDFM0_CLK 输入时钟信号 • 0x1：使能，SDFM0_CLK 输出时钟信号
3	R/W	0x0	SDFM3_CLK_SEL

默认值：0x00000000			SDFM 时钟配置寄存器
位域	类型	默认值	描述
			SDFM3 Clock Select，SDFM3 时钟选择。 • 0x0：选择 SDFM3 管脚输入的时钟信号 • 0x1：选择内部时钟分频的时钟信号，此位使能后，需要使能 SDFM_CLK_DIV_EN
2	R/W	0x0	SDFM2_CLK_SEL SDFM2 Clock Select，SDFM2 时钟选择。 • 0x0：选择 SDFM2 管脚输入的时钟信号 • 0x1：选择内部时钟分频的时钟信号，此位使能后，需要使能 SDFM_CLK_DIV_EN
1	R/W	0x0	SDFM1_CLK_SEL SDFM1 Clock Select，SDFM1 时钟选择。 • 0x0：选择 SDFM1 管脚输入的时钟信号 • 0x1：选择内部时钟分频的时钟信号，此位使能后，需要使能 SDFM_CLK_DIV_EN
0	R/W	0x0	SDFM0_CLK_SEL SDFM0 Clock Select，SDFM0 时钟选择。 • 0x0：选择 SDFM0 管脚输入的时钟信号 • 0x1：选择内部时钟分频的时钟信号，此位使能后，需要使能 SDFM_CLK_DIV_EN

9.3.5.5. 0x0010 GLB_SDFM_SMP_SET

默认值：0x00000000			SDFM 采样时刻配置寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:12	R/W	0x0	SDFM3_SMP_PH_SET SDFM3 Sample Shase Set，SDFM3 采样相位采集设置。 • 0x0：0 相位 • 0x1：1 相位 • 0x2：2 相位

默认值：0x00000000			SDFM 采样时刻配置寄存器
位域	类型	默认值	描述
			• ... • 0xF：15 相位
11:8	R/W	0x0	SDFM2_SMP_PH_SET SDFM2 Sample Shase Set，SDFM2 采样相位采集设置。 • 0x0：0 相位 • 0x1：1 相位 • 0x2：2 相位 • ... • 0xF：15 相位
7:4	R/W	0x0	SDFM1_SMP_PH_SET SDFM1 Sample Shase Set，SDFM1 采样相位采集设置。 • 0x0：0 相位 • 0x1：1 相位 • 0x2：2 相位 • ... • 0xF：15 相位
3:0	R/W	0x0	SDFM0_SMP_PH_SET SDFM0 Sample Shase Set，SDFM0 采样相位采集设置。 • 0x0：0 相位 • 0x1：1 相位 • 0x2：2 相位 • ... • 0xF：15 相位

9.3.5.6. SDn + 0x00 SDFM_FLG

默认值：0x00000000			SDFM 标志寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R/W1C	0x0	COMP_Z_FLG Z Comparator Flag，Z 阈值比较器的状态标记。

默认值：0x00000000			SDFM 标志寄存器
位域	类型	默认值	描述
			• 0x0：无状态产生 • 0x1：产生标记，表示 SSINC 输出数据大于阈值 ZLT
8	R/W1C	0x0	COMP_LS_FLG Second Low Comparator Flag，低位比较器的状态标记。 • 0x0：无状态产生 • 0x1：产生标记，表示 SSINC 输出数据小于低阈值 LLT1
7	R/W1C	0x0	COMP_HS_FLG Second High Comparator Flag，高位比较器的状态标记。 • 0x0：无状态产生 • 0x1：产生标记，表示 SSINC 输出数据大于高阈值 HLT1
6	R/W1C	0x0	FIFO_EMPTY_FLG FIFO Empty Flag，FIFO 空状态标记。 • 0x0：无状态标记 • 0x1：存在状态标记
5	R/W1C	0x0	FIFO_OF_FLG FIFO Overflow Flag，FIFO 上溢出标记 • 0x0：无状态标记 • 0x1：存在溢出状态标记
4	R/W1C	0x0	FIFO_FLG FIFO Flag，FIFO 状态标记，指示 FIFO 收到的数据大于阈值 FIFO_RXTH。 • 0x0：无状态标记 • 0x1：存在状态标记
3	R/W1C	0x0	DAT_DONE_FLG Data Done Flag，数据转换完成标记。 • 0x0：无新数据产生 • 0x1：PSINC 产生了一次新的数据
2	R/W1C	0x0	MOD_FAIL_FLG Modulator Fail Flag，调制器错误标记。 • 0x0：无异常发生 • 0x1：出现异常，SDIN_CLK 输入异常

默认值：0x00000000			SDFM 标志寄存器
位域	类型	默认值	描述
1	R/W1C	0x0	COMP_L_FLG Low Comparator Flag，低位比较器的状态标记。 • 0x0：无状态产生 • 0x1：产生标记，表示 SSINC 输出数据小于低阈值 LLT0
0	R/W1C	0x0	COMP_H_FLG High Comparator Flag，高位比较器的状态标记。 • 0x0：无状态产生 • 0x1：产生标记，表示 SSINC 输出数据大于高阈值 HLT0

9.3.5.7. SDn + 0x04 SDFM_INT_EN

默认值：0x00000000			SDFM 中断使能寄存器
位域	类型	默认值	描述
31:10	–	–	–
9	R/W	0x0	COMP_Z_INT_EN Z Comparator Flag Interrupt Enable，Z 阈值比较器中断状态使能。 • 0x0：未使能 • 0x1：使能
8	R/W	0x0	COMP_LS_INT_EN Second Low Comparator Flag Interrupt Enable，低位比较器中断状态使能。 • 0x0：未使能 • 0x1：使能
7	R/W	0x0	COMP_HS_INT_EN Second High Comparator Flag Interrupt Enable，高位比较器中断状态使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	FIFO_EMPTY_INT_EN FIFO Empty Interrupt Enable，FIFO 空中断状态使能。 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			SDFM 中断使能寄存器
位域	类型	默认值	描述
5	R/W	0x0	FIFO_OF_INT_EN FIFO Overflow Interrupt Enable，FIFO 上溢出中断状态使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	FIFO_INT_EN FIFO Interrupt Enable，FIFO 中断状态使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	DAT_DONE_INT_EN Data Done Interrupt Enable，数据转换完成中断状态使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	MOD_FAIL_INT_EN Modulator Fail Interrupt Enable，调制器错误中断状态使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	COMP_L_INT_EN Low Comparator Flag Interrupt Enable，低位比较器中断状态使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	COMP_H_INT_EN High Comparator Flag Interrupt Enable，高位比较器中断状态使能。 • 0x0：未使能 • 0x1：使能

9.3.5.8. SDn + 0x08 SDFM_CTL

默认值：0x00004000			SDFM 控制寄存器
位域	类型	默认值	描述
31:16	–	–	–

默认值：0x00004000			SDFM 控制寄存器
位域	类型	默认值	描述
15:8	R/W	0x40	SDIN_CLK_THR，Clock Threshold，无时钟异常阈值。
7:5	–	–	–
4	R/W	0x0	SDIN_CLK_CAP，CLK 异常检测方式选择，默认值即可，无需设置。
3	R/W	0x0	SDIN_DAT_INV Data Input Invert，串行输入数据 DAT 进行反相。 • 0x0：无操作 • 0x1：SDIN_DAT 反相
2	R/W	0x0	SDIN_CLK_INV Clock Input Invert，串行输入时钟 CLK 进行反相。 • 0x0：无操作 • 0x1：SDIN_CLK 反相
1:0	R/W	0x0	SDIN_MOD Serial Data Input Mode Set，串行数据输入模式设置。 • 0x0：Mode0 • 0x1：Mode1 • 0x2：Mode2 • 0x3：Mode3

9.3.5.9. SDn + 0x0C PSINC_OUT

默认值：0x00000000			PSINC 输出寄存器
位域	类型	默认值	描述
31:6	–	–	–
5:1	R/W	0x0	DAT_OUT_SHIFT Data Shift Control，输出数据右移位数控制。 • 0x0：无右移 • 0x1：右移 1 位 • 0x2：右移 2 位 • …

默认值：0x00000000			PSINC 输出寄存器
位域	类型	默认值	描述
			• 0x9：右移 9 位 • others：右移 9 位
0	R/W	0x0	DAT_OUT_WDTH Data Output Width，数据输出位宽。 • 0x0：输出 16–bit 数据 • 0x1：输出 32–bit 数据

9.3.5.10. SDn + 0x10 PSINC_SET

默认值：0x00000000			PSINC 设置寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:12	R/W	0x0	SYNC_SRC_SEL PSINC Filter Synchronous Source Select，PSINC 滤波器同步信号源选择。 • 0x0：EPWM0_SDFM_SYNC • 0x1：EPWM1_SDFM_SYNC • 0x2：EPWM2_SDFM_SYNC • … • 0xB：EPWM11_SDFM_SYNC • others：Reserved
11	R/W	0x0	PSINC_SYNC_EN PSINC Filter Synchronous Enable，PSINC 滤波器同步使能。 • 0x0：未使能 • 0x1：使能
10:9	R/W	0x0	PSINC_TYP PSINC Filter Type Set，PSINC 滤波器类型设置。 • 0x0/0x1：SINC1 • 0x2：SINC2 • 0x3: SINC3
8	R/W	0x0	PFEN

默认值：0x00000000			PSINC 设置寄存器
位域	类型	默认值	描述
			PSINC Filter Enable，PSINC 滤波器使能。 • 0x0：未使能 • 0x1：使能
7:0	R/W	0x0	POSR PSINC Filter Oversampling Ratio，PSINC 滤波器过采样率。 • 0x0：过采样率为 1 倍 • 0x1：过采样率为 2 倍 • 0x2：过采样率为 3 倍 • … • 0xFF：过采样率为 256 倍

9.3.5.11. SDn + 0x14 SSINC_TH

默认值：0x00000000			SSINC 比较阈值寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	LLT0 Low-level Threshold，SSINC 低位比较阈值0。
15:0	R/W	0x0	HLT0 High-level Threshold，SSINC 高位比较阈值0。

9.3.5.12. SDn + 0x18 SSINC_SET

默认值：0x00000000			SSINC 设置寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:6	R/W	0x0	SSINC_TYP SSINC Filter Type Set，SSINC 滤波器类型设置。 • 0x0/0x1：SINC1 • 0x2：SINC2 • 0x3: SINC3

默认值：0x00000000			SSINC 设置寄存器
位域	类型	默认值	描述
5	R/W	0x0	SFEN SSINC Filter Enable，SSINC 滤波器使能。 • 0x0：未使能 • 0x1：使能
4:0	R/W	0x0	SOSR SSINC Filter Oversampling Ratio，SSINC 滤波器过采样率。 • 0x0：过采样率为 1 倍 • 0x1：过采样率为 2 倍 • 0x2：过采样率为 3 倍 • … • 0x1F：过采样率为 32 倍

9.3.5.13. SDn + 0x1C SD_DAT

默认值：0x00000000			PSINC 设置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	HDAT16 High 16–bit Data，高 16–bit 数据，在使能 PSINC 输出位宽为 32–bit 模式下，此域生效。
15:0	R/W	0x0	LDAT16 Low 16–bit Data，低 16–bit 数据，在使能 PSINC 输出位宽为 16–bit 模式下，只需读取 LDAT16 即可。

9.3.5.14. SDn + 0x20 FIFO_CTL

默认值：0x00000000			FIFO 控制寄存器
位域	类型	默认值	描述
31:14	–	–	–
13:8	R/W	0x0	FIFO_RXTH

默认值：0x00000000			FIFO 控制寄存器
位域	类型	默认值	描述
			FIFO Receive Threshold，FIFO 接收数据的阈值设置（单位 32 bits），当 FIFO 接收到的数据多于阈值 FIFO_RXTH 时，硬件产生 FIFO_FLG 中断状态。若同时 DMA 使能位 DMA_EN = 1，则产生 DMA 请求。
7:5	–	–	–
4	R/W	0x0	DMA_EN DMA Enable，DMA 使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	FIFO_SYNC_CLR FIFO SYNC Clear，在 FIFO_WAIT_SYNC = 1 的情况下，此位选择是否在 FIFO 达到阈值时清除 FIFO 数据。 • 0x0：未使能，保持原有数据 • 0x1：使能，覆盖原有数据
2	R/W	0x0	FIFO_WAIT_SYNC FIFO Wait for SYNC，FIFO 等待 SYNC 信号，仅当 SYNC 信号出现后，FIFO 才开始进行存入数据。 • 0x0：未使能 • 0x1：使能
1	R/W1AC	0x0	FIFO_FLUSH FIFO Flush，擦除 FIFO，写 1 擦除 FIFO 数据，完成后自动清 0。
0	R/W	0x0	FIFO_EN FIFO Enable，FIFO 使能。 • 0x0：未使能 • 0x1：使能

9.3.5.15. SDn + 0x28 FIFO_DAT

默认值：0x00000000			FIFO 数据寄存器
位域	类型	默认值	描述
31:0	R	0x0	FIFO_DAT FIFO Data，FIFO 数据，通过读取此寄存器可获得存取的数据。

9.3.5.16. SDn + 0x30 SSINC_TH1

默认值：0x00000000			SSINC 比较阈值 1 寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	LLT1 Low-level Threshold，SSINC 低位比较阈值 1。
15:0	R/W	0x0	HLT1 High-level Threshold，SSINC 高位比较阈值 1。

9.3.5.17. SDn + 0x34 SSINC_ZLT

默认值：0x00000000			SSINC ZLT 阈值寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	ZLT Zero-level Threshold，ZLT 比较阈值。

9.3.5.18. 0x0FFC SDFM_VER

默认值：0x00000100			SDFM 模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0100	版本寄存器 V1.0

9.4. Analog-to-Digital Converter (ADC)

ADC 模块用于对外部电流、电压、以及其它模拟量信号的采样。配合 EPWM 系统输出的采样触发信号，能够精准控制 ADC 的采样时刻，适合应用于多种实时控制场景。

9.4.1. 特性说明

- 最多支持 24 个外部模拟信号输入通道采集，以及对 4 路 PGA 的输出信号采集
- 支持 EPWM CVRA/ CVRB 硬件触发采样
- 支持 14-bit 模式，采样率最高为 2 Msps
- 支持外部信号触发采样

- 支持软件触发采样
- 支持双队列模式和单队列模式
- 支持 1/2/4/8/16/32 倍过采样模式
- 支持外部输入参考电压
- 队列 1 FIFO 为 36*22 bits，队列 2 FIFO 为 18*22 bits
- 常规队列支持单次模式和周期模式

9.4.2. 原理框图

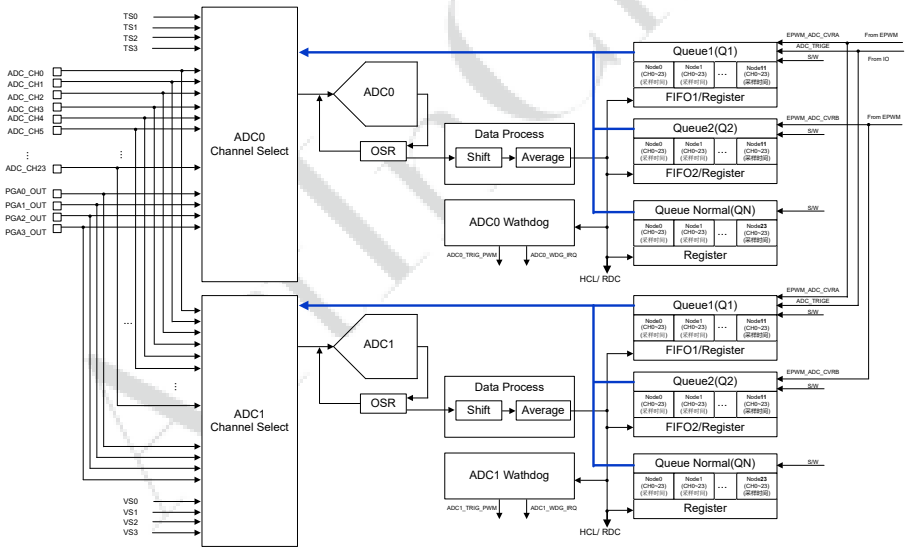


图 9–33 ADC 模块原理图框图

ADC0 和 ADC1 的输入通道的选择，是基于各自队列 1 和队列 2 的配置。当有效的队列触发信号到来时，ADC 会依次对队列节点所对应的通道进行采样转换，并将转换后的数字数据保存在队列的 FIFO 中。

以下是关于队列 1 和队列 2 的触发信号的详细说明：

- **队列 1 对应的触发信号**
  - EPWM_ADC_CVRA：由 EPWM 模块产生的触发信号。
  - ADC_TRIGE：外部输入信号（上升沿触发），需要通过管脚输入。
  - SW：软件触发信号，通过软件写寄存器产生。
- **队列 2 对应的触发信号**
  - EPWM_ADC_CVRB：由 EPWM 模块产生的触发信号。
  - SW：软件触发信号，通过软件写寄存器产生。

9.4.3. 功能描述

9.4.3.1. ADC 队列

在 ADC 模块中，采用队列方式进行采样转换。通过寄存器设置队列的采样点数以及每个节点对应的通道，可以精确控制采样过程。

当触发信号到来时，ADC 会根据队列的设置，从节点 0 开始，依次对每个节点进行采样转换，直至完成队列中所有节点的采样转换。

ADC 内置实时队列和常规队列，特点如下：

表 9–21 ADC 内置队列特性描述

队列模式	优先级	特性	触发信号
实时队列 (Q1, Q2, QC)	优先级最高，可打断常规队列 QN 的当前请求。	实时队列包括队列 Q1、Q2： • Q1、Q2 或 QC 在触发信号作用下，根据队列长度及设置完成采样转换。 • 队列配置不支持动态写入，仅在任务完成后可修改。 • 支持过采样功能，以提高采样精度。 • 结果数据可通过 FIFO 或寄存器读取。	• Q1: EPWM_ADC_CVRA, ADC_TIRGE, S/W • Q2: EPWM_ADC_CVRB, S/W • QC: EPWM_ADC_CVRA, EPWM_ADC_CVRB, ADC_TIRGE, S/W
常规队列 (QN)	优先级低于实时队列。 当实时队列有采样请求时，常规队列的任务会被暂时打断，直到实时队列的任务完成后，常规队列才会从被打断的地方继续执行。	• 支持单次或连续执行，连续模式下可配置间隔时间。 • 最多支持 24 个节点，每个节点的采样时间可配置。 • 不支持过采样模式。 • 队列配置支持动态写入，可在每次任务完成后更新配置。	S/W

9.4.3.1.1. 实时队列配置

实时队列支持下列模式：

- 双队列模式：Q1 和 Q2 独立工作，各支持最多 12 个节点。
- 合并队列模式：Q1 和 Q2 合并成 QC，合并后形成的 QC 队列最多支持 24 个节点。

  需对 Q1 和 Q2 的节点寄存器进行配置，以确定每个节点对应的通道。

两种模式均支持以下配置：

表 9–22 实时队列模式通用配置说明

参数	描述	备注
Q1/2_TRIG_CNT	队列采样节点数	–
ADC_Q1_NODE	各节点对应的采样通道	–

表 9–22 实时队列模式通用配置说明 (续)

参数	描述	备注
ADC_Q2_NODE	各节点对应的采样通道	–
ADC_ACQ	采样时间	各个节点可以通过寄存器选择四种采样时间，从而满足不同节点的不同采样时间需求，详情可查看ADC 采样率设置。
ADC_ACQ_1	采样时间	
ADC_ACQ_2	采样时间	
ADC_ACQ_3	采样时间	
Q1_SMP_SETn	过采样设置，各节点默认采用单点采集。	通过寄存器可以选择进行 2/ 4/ 8/ 16/ 32 倍的过采样。
Q2_SMP_SETn	过采样设置，各节点默认采用单点采集。	
ADC_RES_FLTn	对 ADC 结果寄存器的数据支持进行平均算法滤波。	如不需要额外 ADC 数据处理，使用默认值即可。关于详细说明，可查看ADC 数据处理。
ADC_OFF_EN	偏置调整，调整 ADC 转换结果的偏置。	

如图 9–34 : 实时队列配置示例所示，队列 Q1 设置采样节点数为 3，节点 1 对 CH5 进行 4 倍过采样，节点 2 对 CH1 进行单点采样，节点 3 对 CH2 进行 2 倍过采样。

节点 1 在完成 4 倍过采样后，即进行连续 4 次采样转换并结果求平均后，视为此节点任务完成，平均后的结果即为过采样的结果。

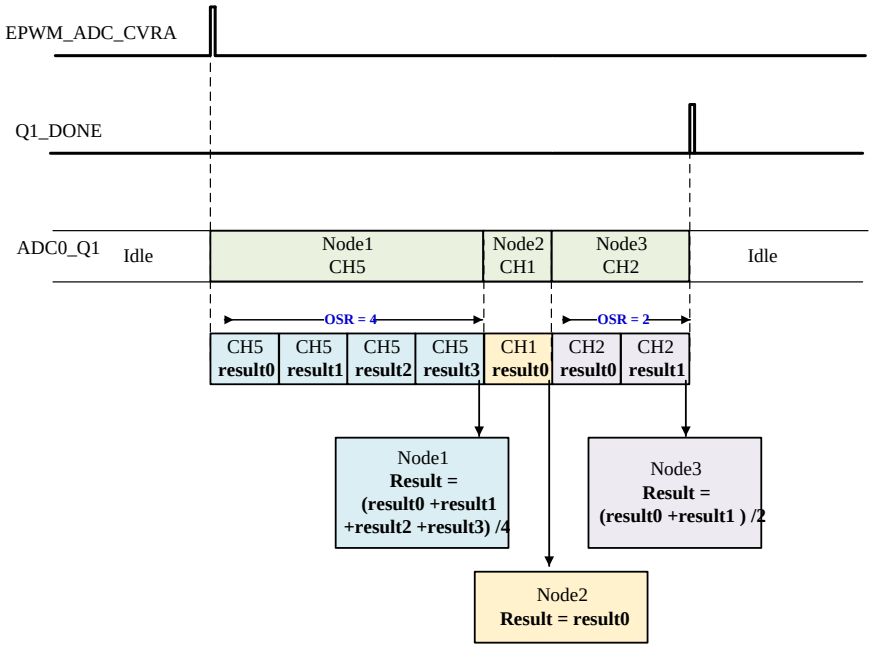


图 9–34 实时队列配置示例

9.4.3.1.2. 双队列模式配置

在实时队列下，可配置 ADC 为双队列模式，即队列 1（Q1）和队列 2（Q2）独立工作，各自最大可支持 12 个节点。

表 9–23 双队列模式配置说明

参数	描述	备注
Q1/2_TRIG_CNT	队列采样节点数	–
ADC_Q1_NODE	各节点对应的采样通道	–
ADC_Q2_NODE	各节点对应的采样通道	–
ADC_ACQ	采样时间	各个节点可以通过寄存器选择四种采样时间，从而满足不同节点的不同采样时间需求，详情可查看ADC 采样率设置。
ADC_ACQ_1	采样时间	
ADC_ACQ_2	采样时间	
ADC_ACQ_3	采样时间	
Q1_SMP_SETn	过采样设置，各节点默认采用单点采集。	通过寄存器可以选择进行 2/ 4/ 8/ 16/ 32 倍的过采样。
Q2_SMP_SETn	过采样设置，各节点默认采用单点采集。	–
ADC_RES_FLTn	对 ADC 结果寄存器的数据支持进行平均算法滤波。	如不需要额外 ADC 数据处理，使用默认值即可。关于详细说明，可查看ADC 数据处理。
ADC_OFF_EN	偏置调整，调整 ADC 转换结果的偏置。	
EPWM_ADC_CVRA	EPWM 产生的触发信号 A，此信号为所有 CVRA 的 OR 逻辑信号（n=0/1/2/…/11）。	Q1 支持的触发信号。
EPWM_ADC_CVRB	EPWM 产生的触发信号 B，此信号为所有 CVRB 的 OR 逻辑信号（n=0/1/2/…/11）。	Q2 支持的触发信号。
ADC_TRIGE	外部的触发信号，通过外部管脚输入。	Q1 支持的触发信号。
SW_TRIG	软件触发信号，通过软件写寄存器字段 TRIGS 产生。	Q1 和 Q2 支持的触发信号。

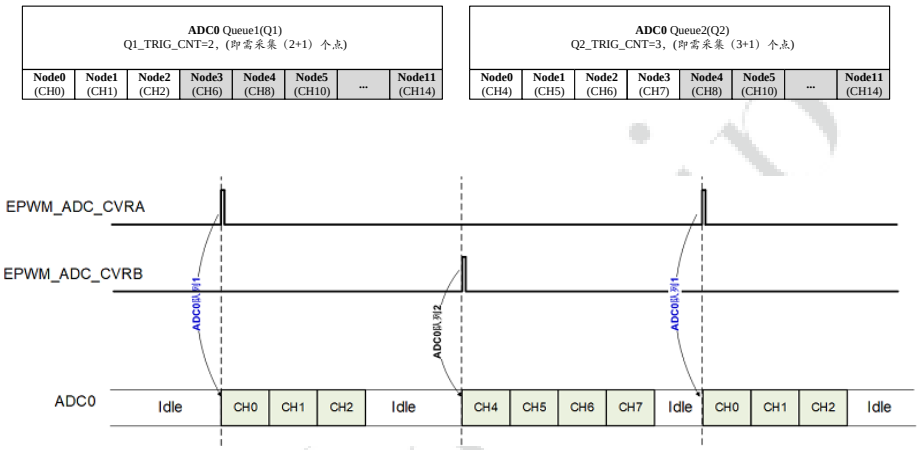
在双队列模式下，Q1 和 Q2 是独立的两个队列，为了避免 Q1 和 Q2 互相干扰，设置各自的触发信号时需保证 ADC 完成某一个队列后再完成下一个队列。以 ADC0 举例说明队列 1 和队列 2 的使用方法：

- 队列 1 设置 3 个采样节点 (Node0~Node2)，依次对 CH0、CH1、CH2 进行采样。当接收到队列 1 的有效触发信号后，按顺序完成这三个通道的采样转换，完成后队列 1 任务结束，ADC0 进入空闲状态。

  由于队列 1 里设置 3 个采样节点，所以队列 1 中的 Node3~Node11 都不会生效。

- 队列 2 设置 4 个采样节点 (Node0~Node3)，依次对 CH4、CH5、CH6、CH7 进行采样。当接收到队列 2 的有效触发信号后，按顺序完成这四个通道的采样转换，完成后队列 2 任务结束，ADC0 进入空闲状态。

  由于队列 2 设置 4 个采样点，所以队列 2 中的 Node4~node11 都不会生效。



9.4.3.1.3. 合并队列模式

在实时队列下，可配置 ADC 为单队列模式，即 Q1 和 Q2 进行合并而成的 QC 队列，最大支持 24 个节点。

表 9–24 双队列模式配置说明

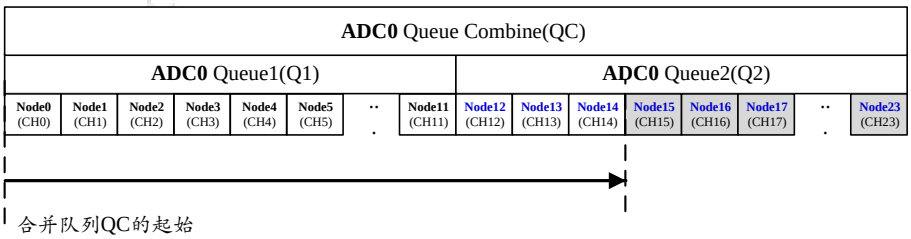
参数	描述	备注
Q1_TRIG_CNT	队列采样节点数	–
ADC_Q1_NODE	0~11 节点各自对应的采样通道	–
ADC_Q2_NODE	12~23 节点各自对应的采样通道	–
ADC_ACQ	采样时间	各个节点可以通过寄存器选择四种采样时间，从而满足不同节点的不同采样时间需求，详情可查看ADC 采样率设置。
ADC_ACQ_1	采样时间	
ADC_ACQ_2	采样时间	
ADC_ACQ_3	采样时间	
Q1_SMP_SETn	过采样设置，各节点默认采用单点采集。	通过寄存器可以选择进行 2/ 4/ 8/ 16/ 32 倍的过采样。
Q2_SMP_SETn	过采样设置，各节点默认采用单点采集。	–
ADC_RES_FLTn	对 ADC 结果寄存器的数据支持进行平均算法滤波。	如不需要额外 ADC 数据处理，使用默认值即可。关于详细说明，可查看ADC 数据处理。
ADC_OFF_EN	偏置调整，调整 ADC 转换结果的偏置。	

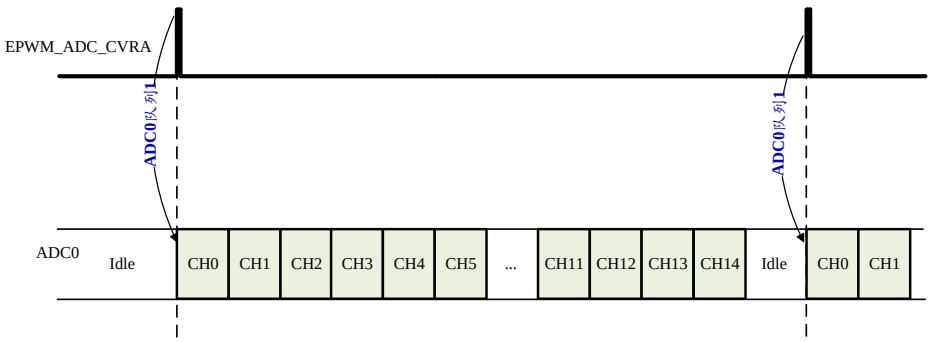
表 9–24 双队列模式配置说明 (续)

参数	描述	备注
EPWM_ADC_CVRA	EPWM 产生的触发信号 A，此信号为所有 CVRA 的 OR 逻辑信号（n=0/1/2/…/11）。	QC 支持的触发信号。
EPWM_ADC_CVRB	EPWM 产生的触发信号 B，此信号为所有 CVRB 的 OR 逻辑信号（n=0/1/2/…/11）。	
ADC_TRIGE	外部的触发信号，通过外部管脚输入。	
SW_TRIG	软件触发信号，通过软件写寄存器字段 TRIGS 产生。	

以 ADC0 举例说明合并队列 QC（采用 15 个采样节点）的使用方法：

- 设置采用合并队列 QC，并设置 15 个采样节点（Node0~Node14）。
- 对 Q1 的节点寄存器依次配置为 CH0、CH1、CH2 等表示 Node0 – Node11 节点。
- 对 Q2 的节点寄存器依次配置为 CH12、CH13、CH14，表示 Node12~Node14。
- ADC 在接收到合并队列的有效触发信号后，依次对通道进行一次采样转换，完成后 ADC0 进入空闲状态。由于合并队列 QC 里设置 15 个采样节点，所以队列中的 Node15~Node23 都不会生效.





9.4.3.1.4. ADC0/1 实时队列采样同步

ADC0 和 ADC1 各自拥有实时队列 Q1 和 Q2，用于存储 ADC 转换的结果数据。ADC0 的 Q1 和 Q2 与 ADC1 的 Q1 和 Q2 对应相同的触发信号，这意味着当触发信号到来时，两台 ADC 将同时开始转换过程。通过配置 ADC0 的寄存器字段 ADC_SMP_SYC_EN = 0x1（仅 ADC0 拥有此字段），可以启用 ADC0 和 ADC1 之间的同步功能。

使能同步后，ADC0 的 Q1 与 ADC1 的 Q1 进行同步，同时 ADC0 的 Q2 与 ADC1 的 Q2 也进行同步。两个 ADC 在相同的时间点上对输入信号进行采样和转换，确保数据的一致性和准确性。

9.4.3.2. ADC 采样率设置

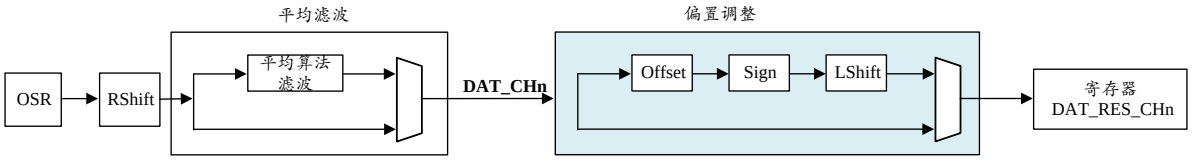
ADC 将模拟信号转换为数字信号，其采样率决定了单位时间内的采样次数。采样率越高，对信号的细节捕捉能力越强，但也可能增加数据处理的难度和存储需求。ADC 采样率相关参数及描述如下所示：

表 9–25 ADC 采样率配置描述

参数	描述	备注
ADC_ACQ	采样保持时间 （ADC_ACQ+1）* T(ADC_ANACNT_DIVCLK)	• 采样保持时间与 VCM_SET 的关系： ◦ VCM_SET = 1 时，采样保持时间 > 500 ns ◦ VCM_SET = 8 时，采样保持时间 > 250 ns ◦ VCM_SET = 15 时，采样保持时间 > 125 ns • 通过控制 ADC 采样保持时间，可以设置 ADC 的转换时间。 • ADC 的转换时间: 15 * T（ADC_CMP_CLK）+（ADC_ACQ +2）* T(ADC_ANACNT_DIVCLK)+2 * T（AHB_CLK）。 • 举例： AHB = 240 MHz ADC_ACQ=34 CLK_HIGH_CYCLE=3 CLK_LOW_CYCLE=1 ADC_ANACNT_DIVC=2 ADC 的转换时间为 0.83 us，最大转化率为1.2 MSPS。
ADC_ACQ_1	采样保持时间 （ADC_ACQ+1）* T(ADC_ANACNT_DIVCLK)	
ADC_ACQ_2	采样保持时间 （ADC_ACQ+1）* T(ADC_ANACNT_DIVCLK)	
ADC_ACQ_3	采样保持时间 （ADC_ACQ+1）* T(ADC_ANACNT_DIVCLK)	
ADC_CMP_CLK	ADC 比较时钟： F(ADC_CMP_CLK) = AHB / (CLK_LOW_CYCLE + CLK_HIGH_CYCLE + 2) 建议设置为 40 MHz（即 AHB=240 MHz,CLK_HIGH_CYCLE=3，CLK_LOW_CYCLE=1）。	
ADC_ANACNT_DIVCLK	ADC 模拟计数分频时钟： F（ADC_ANACNT_DIVCLK）= AHB_CLK / (ADC_ANACNT_DIVCLK+1) 建议设置为 80 MHz（即 AHB = 240 MHz, ADC_ANACNT_DIVCLK=2）。	

9.4.3.3. ADC 数据处理

ADC 数据处理包含以下关键步骤：



- 右移处理，对 ADC 队列 Q1/2/C/N 的结果进行右移处理，通过寄存器 ADC_RES_SHIFT 配置，可以选择右移 0、1、2 或 3 位。右移处理在过采样处理后进行，用于调整数据精度。
- 平均处理，对 ADC 结果寄存器的数据支持进行平均算法滤波。若采用平均算法滤波，则寄存器 DAT_RES_CHn 中的数据为经过平均算法滤波后的结果。平均处理是在过采样和右移处理后进行。
- 偏置调整，调整 ADC 转换结果的偏置。使能后，可进行如下偏置运算：

```
DAT_RES_CHn = ((DAT_CHn - DAT_OFFn) * (-1)^SIGN_EN) << LS
```

  - DAT_OFFn：寄存器配置的偏置值，与 ADC 的转换结果位宽相同。

    例如，对于 14–bit 的 ADC 结果，则软件中 `DAT_OFFn` 应设置为 14–bit 的无符号数。经过偏置模块计算后，结果转换为有符号数。
  - SIGN_EN：符号翻转使能。
  - LS：对最终数据进行左移的位数，需根据 ADC 转换结果（DAT_CHn) 的位宽设置。例如，对于 14–bit 的 ADC 结果，若配置 LS = 4，则对数据进行左移四位，输出为有符号的 Q15 格式结果。

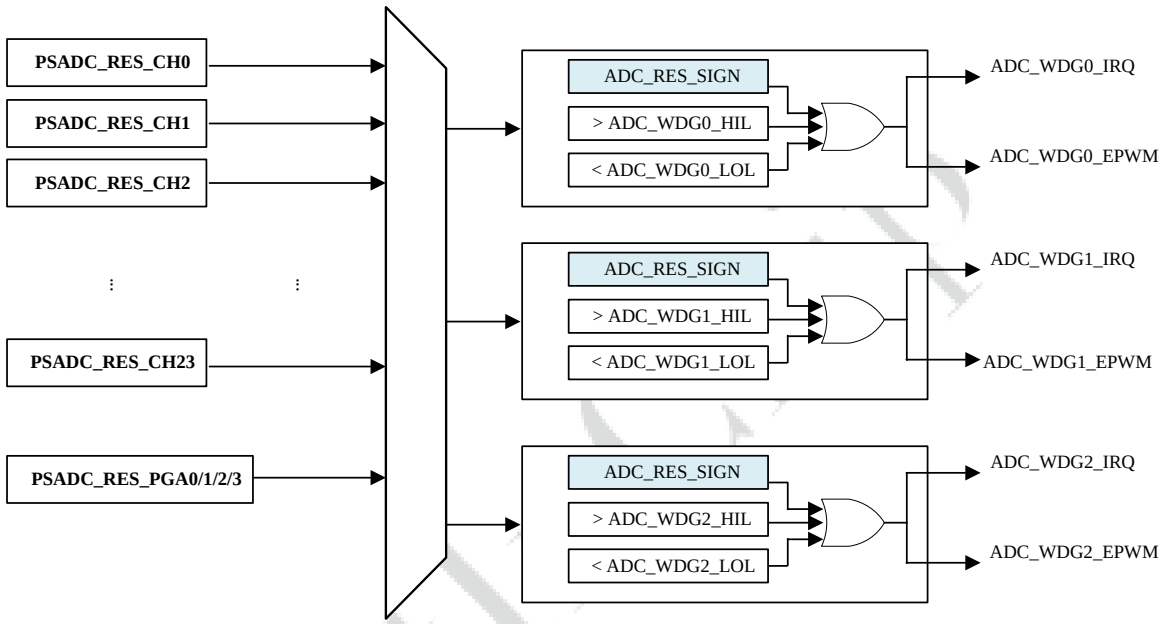
数据处理完成后，支持下列读取模式：

- 寄存器读取模式：ADC 各个通道的采样转换结果数据存储在对应的通道结构数据寄存器 DAT_RES_CHn（n=0/1/2/.../23）和 DAT_RES_PGAn（n=0/1/2/3）中。这些寄存器只有在每次完成通道信号的采样转换后才进行更新。
- FIFO 读取模式：ADC 采样转换的数据按照队列的顺序依次保存在队列的 FIFO 中。

  对于队列 1，假设设置 3 个采样节点，依次采样 CH0、CH1 和 CH2 的通道信号。当 ADC 完成队列 1 的采样转换后，会产生 Q1_INT 状态（指示完成了一次队列 1 的任务）。软件可以通过轮询或中断的方式读取队列 1 的 FIFO 获取转换结果。 第一次读取得到 CH0 的转换数据，第二次读取得到 CH1 的转换数据，第三次读取得到 CH2 的转换数据。

  此外，FIFO 数据的读取支持 DMA 操作，需要设置 DMA 使能以及 FIFO 阈值寄存器，才可以实现 DMA 自动获取 ADC 结果数据。

9.4.3.4. ADC 看门狗



9.4.3.4.1. ADC 看门狗特性

- 最多支持 3 路看门狗
- 看门狗的输入检测数据可以选择 ADC_RES_CHn
- 支持中断和 EPWM 保护功能
- 支持高低阈值检测和过 0 检测

9.4.3.4.2. Q15 格式数据处理

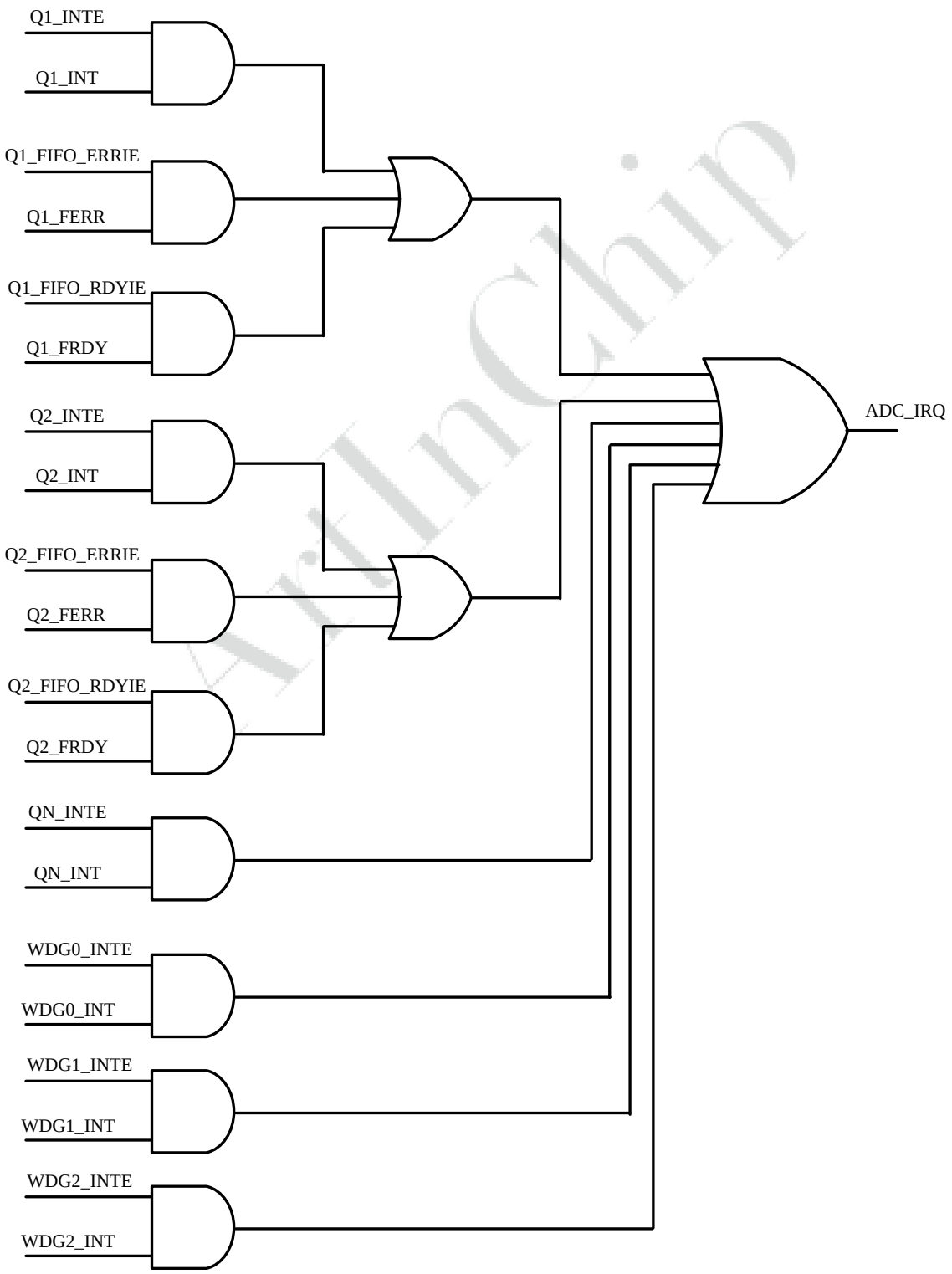
经过偏置调整模块的处理后，所得的 ADC 结果转换为 Q15 的有符号数。因此，在 ADC 看门狗模块的数据也是 Q15 格式，且相应配置的高低阈值需要设置为 Q15 格式。

9.4.3.4.3. ADC_RES_SIGN（过 0）检测功能

ADC_RES_SIGN（过 0）检测功能：

- 支持正向过 0 检测，即结果数据从正数变为负数
- 支持反向过 0 检测，即结果数据从负数变为正数
- 支持正反向过 0 检测，无论是正向还是反向过 0 检测都可以触发看门狗事件

9.4.3.5. 中断产生机制



ADC 模块支持产生中断信号，如上图所示，根据用户的需要，配置寄存器使能相应的信号源作为中断信号即可。其中中断信号源依次为如下：

- Q1_INT：指示完成队列 1（或队列 QC）所有节点的采样转换的标记信号
- Q1_FRDY：指示队列 1（或队列 QC）的 FIFO 达到了阈值的标记信号
- Q1_FERR：指示队列 1（或队列 QC）的 FIFO 出现异常的标记信号
- Q2_INT：指示完成队列 2 所有节点的采样转换的标记信号
- Q2_FRDY：指示队列 2 的 FIFO 达到了阈值的标记信号
- Q2_FERR：指示队列 2 的 FIFO 出现异常的标记信号
- QN_INT：指示完成队列 N 所有节点的采样转换的标记信号
- WDG0/1/2_INT：指示看门狗标记信号

9.4.4. 寄存器列表

表 9–26 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x000	ADC_MCR	ADC 模块控制	0x000 ADC_MCR
0x004	ADC_TCR	ADC 队列采样节点数	0x004 ADC_TCR
0x010	ADC_MSR	ADC 模块状态	0x010 ADC_MSR
0x014	ADC_CALCSR	ADC 校准控制寄存器	0x014 ADC_CALCSR
0x018	ADC_GAINCSR	增益控制寄存器	0x018 ADC_GAINCSR
0x01C	ADC_TRIGE_FILTER	TRIGE 输入滤波控制	0x01C ADC_TRIGE_FILTER
0x020	ADC_Q1FCR	ADC 队列 1 FIFO 控制	0x020 ADC_Q1FCR
0x024	ADC_Q2FCR	ADC 队列 2 FIFO 控制	0x024 ADC_Q2FCR
0x028	ADC_SMP_CFG	ADC 采样时间配置寄存器	0x028 ADC_SMP_CFG
0x02C	ADC_SHIFT	ADC 移位控制寄存器	0x02C ADC_SHIFT
0x040	ADC_Q1FDR	ADC 队列 1 FIFO 数据	0x040 ADC_Q1FDR
0x080	ADC_Q2FDR	ADC 队列 2 FIFO 数据	0x080 ADC_Q2FDR
0x090	ADC_HCL_CTL	ADC 硬件电流环控制寄存器	0x090 ADC_HCL_CTL
0x094	ADC_CTL	ADC 控制寄存器	0x094 ADC_CTL

表 9–26 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x098	ADC_CTL1	ADC 控制寄存器 1	0x098 ADC_CTL1
0x09C	ADC_CAL_RS	校准结果寄存器	0x09C ADC_CAL_RS
0x0A0	ADC_Q1_NOD03	采样队列 1 节点寄存器	0x0A0 ADC_Q1_NOD03
0x0A4	ADC_Q1_NOD47	采样队列 1 节点寄存器	0x0A4 ADC_Q1_NOD47
0x0A8	ADC_Q1_NOD811	采样队列 1 节点寄存器	0x0A8 ADC_Q1_NOD811
0x0B0	ADC_Q2_NOD03	采样队列 2 节点寄存器	0x0B0 ADC_Q2_NOD03
0x0B4	ADC_Q2_NOD47	采样队列 2 节点寄存器	0x0B4 ADC_Q2_NOD47
0x0B8	ADC_Q2_NOD811	采样队列 2 节点寄存器	0x0B8 ADC_Q2_NOD811
0x0C0	ADC_Q1_SMP_SET	采样队列 1 采样时间寄存器	0x0C0 ADC_Q1_SMP_SET
0x0C4	ADC_Q2_SMP_SET	采样队列 2 采样时间寄存器	0x0C4 ADC_Q2_SMP_SET
0x0C8	ADC_Q1_SMP_SET0	采样队列 1 过采样配置 0 寄存器	0x0C8 ADC_Q1_SMP_SET0
0x0CC	ADC_Q1_SMP_SET1	采样队列 1 过采样配置 0 寄存器	0x0CC ADC_Q1_SMP_SET1
0x0D0	ADC_Q2_SMP_SET0	采样队列 2 过采样配置 0 寄存器	0x0D0 ADC_Q2_SMP_SET0
0x0D4	ADC_Q2_SMP_SET1	采样队列 1 过采样配置 0 寄存器	0x0D4 ADC_Q2_SMP_SET1
0x0D8	ADC_RES_FLT0	通道结果滤波器 0 寄存器	0x0D8 ADC_RES_FLT0
0x0DC	ADC_RES_FLT1	通道结果滤波器 1 寄存器	0x0DC ADC_RES_FLT1
0x0E0	ADC_RES_FLT2	通道结果滤波器 2 寄存器	0x0E0 ADC_RES_FLT2
0x0100+0x4*n(n=0/1/.../23)	ADC_RES_CHn	通道 n 寄存器	0x0100+0x4*n(n=0/1/.../23) ADC_RES_CHn

表 9–26 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0180+0x4*n(n=0/1/.../23)	ADC_OFF_CHn	通道 n 偏置调整寄存器	0x0180+0x4*n(n=0/1/.../23) ADC_OFF_CHn
0x0220	ADC_WDG0_CFG	看门狗 0 配置寄存器	0x0220 ADC_WDG0_CFG
0x0224	ADC_WDG0_LVL	看门狗 0 阈值配置寄存器	0x0224 ADC_WDG0_LVL
0x0230	ADC_WDG1_CFG	看门狗 1 配置寄存器	0x0230 ADC_WDG1_CFG
0x0234	ADC_WDG1_LVL	看门狗 1 阈值配置寄存器	0x0234 ADC_WDG1_LVL
0x0240	ADC_WDG2_CFG	看门狗 2 配置寄存器	0x0240 ADC_WDG2_CFG
0x0244	ADC_WDG2_LVL	看门狗 2 阈值配置寄存器	0x0244 ADC_WDG2_LVL
0x0248	ADC_WDG_STA	看门狗状态寄存器	0x0248 ADC_WDG_STA
0x0264	ADC_HCL_CFG	_HCL_CFG 寄存器	#unique_13_Connect_42_section_ik3_mr2_h2c
0x0300	ADC_QN_MCR	常规队列控制寄存器	0x0300 ADC_QN_MCR
0x304	ADC_QN_TCR	常规队列采样节点数寄存器	0x304 ADC_QN_TCR
0x308	ADC_PSI	常规队列采样间隔寄存器	0x308 ADC_PSI
0x030C	ADC_QN_MSR	常规队列状态寄存器	0x030C ADC_QN_MSR
0x0310	ADC_QN_NOD0	采样队列 N 节点寄存器	0x0310 ADC_QN_NOD0
0x0314	ADC_QN_NOD1	采样队列 N 节点寄存器	0x0314 ADC_QN_NOD1
0x0318	ADC_QN_NOD2	采样队列 N 节点寄存器	0x0318 ADC_QN_NOD2
0x031C	ADC_QN_NOD3	采样队列 N 节点寄存器	0x031C ADC_QN_NOD3
0x0320	ADC_QN_NOD4	采样队列 N 节点寄存器	0x0320 ADC_QN_NOD4
0x0324	ADC_QN_NOD5	采样队列 N 节点寄存器	0x0324 ADC_QN_NOD5
0x0328	ADC_QN_SMP_SET0	采样队列 N 采样时间寄存器	0x0328 ADC_QN_SMP_SET0

表 9–26 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x032C	ADC_QN_SMP_SET1	采样队列 N 采样时间寄存器	0x032C ADC_QN_SMP_SET1
0x0330	ADC_QNFCR	ADC 队列 N FIFO 控制寄存器	0x0330 ADC_QNFCR
0x0334	ADC_QNFDR	常规队列 FIFO 数据寄存器	0x0334 ADC_QNFDR
0x0380	ADC_CAPCALCSR	常规队列 FIFO 数据寄存器	0x0380 ADC_CAPCALCSR
0x0384	ADC_DCALC10	数据校准配置寄存器	0x0384 ADC_DCALC10
0x0388	ADC_DCALC32	数据校准配置寄存器	0x0388 ADC_DCALC32
0x038C	ADC_DCALC54	数据校准配置寄存器	0x038C ADC_DCALC54
0x0390	ADC_DCALC76	数据校准配置寄存器	0x0390 ADC_DCALC76
0x0394	ADC_DCALC98	数据校准配置寄存器	0x0394 ADC_DCALC98
0x0398	ADC_DCALC1110	数据校准配置寄存器	0x0398 ADC_DCALC1110
0x039C	ADC_DCALC1312	数据校准配置寄存器	0x039C ADC_DCALC1312
0x03A0	ADC_DCALC1514	数据校准配置寄存器	0x03A0 ADC_DCALC1514
0x03A4	ADC_DCALC1716	数据校准配置寄存器	0x03A4 ADC_DCALC1716
0x03A8	ADC_DCALCx18	数据校准配置寄存器	0x03A8 ADC_DCALCx18
0xFFC	VERSION	ADC 模块版本	0xFFC VERSION

9.4.5. 寄存器描述

9.4.5.1. 0x000 ADC_MCR

默认值：0x00002F00			ADC 模块控制 (ADC Module Control Register)
位域	类型	默认值	描述
31	–	–	–

默认值：0x00002F00			ADC 模块控制 (ADC Module Control Register)
位域	类型	默认值	描述
30	R/WAC	0x0	Q2_TRIGS Queue 2 Trigger Status，队列 2 的状态，同时软件也可以通过本字段触发 ADC 队列 2 进行采样。 • 0x0：显示 Q2 处于空闲状态，此状态下，软件可以通过置位本字段触发 Q2 进行采样 • 0x1：显示 Q2 处于工作状态，采样的触发有可能来自于任何触发源（包括 TRIGB/TRIGS），一旦采样完成，本字段会被硬件自动清零。
29	–	–	–
28	R/W	0x0	Q2_TRIGB Queue 2 Trigger B Enable，队列 2 EPWM 采样触发 B 使能信号 • 0x0：禁止 EPWM_ADC_CVRB 触发队列 2 • 0x1：使能 EPWM_ADC_CVRB 触发队列 2。 关于 EPWM_ADC_CVRB 触发信号所代表的含义，请参考 EPWM 模块的配置。
27	–	–	–
26	R/W	0x0	Q2_INTE Queue 2 Interrupt Enable，队列 2 中断使能。 • 0x0：禁止 Q2_INT 中断 • 0x1：使能 Q2_INT 中断
25	R/W	0x0	Q2_INTM Queue 2 Interrupt Mode，队列 2 中断模式选择。 • 0x0：每次队列执行结束都会产生一次 Q2_INT 中断 • 0x1：每隔一次（每两次）队列执行结束就会产生一次 Q2_INT 中断
24	R/WAC	0x0	Q2_RST Queue 2 Reset，队列 2 复位信号。 • 0x0：无使能 • 0x1：立刻复位队列 2 到 Q2 的节点 0，下一次触发发生后，从节点 0 开始执行采样，复位后现有未执行的采样命令立刻放弃，并等待已执行的采样命令完成后，硬件自动清零本字段。
23	R/W	0x0	QC_TRIGB Queue of Combination Trigger B，合并队列的 EPWM 采样触发 B 使能信号。

默认值：0x00002F00			ADC 模块控制 (ADC Module Control Register)
位域	类型	默认值	描述
			• 0x0：禁止 EPWM_ADC_CVRB 触发 QC • 0x1：使能 EPWM_ADC_CVRB 触发 QC 关于 PWM_ADC_CVRB 触发信号所代表的含义，请参考 EPWM 模块的配置
22	R/WAC	0x0	Q1_TRIGS Queue 1 Trigger Status，队列 1 的状态，同时软件也可以通过本字段触发 ADC 队列 1 进行采样。另外当使能了合并队列 QUE_COMB = 0x1，这里反馈的是合并队列的状态。 • 0x0：显示 Q1 或 QC 处于空闲状态，此状态下，软件可以通过置位本字段触发 Q1/QC 进行采样。 • 0x1：显示 Q1 或 QC 处于工作状态，采样的触发有可能来自于任何触发源（包括 TRIGA/TRIGB/TRIGE/TRIGS），一旦采样完成，本字段会被硬件自动清零。
21	R/W	0x0	Q1_TRIGE Queue 1 Trigger E Enable，外部触发信号（TRIGE）对 ADC 队列 1 的采样触发使能。另外当使能了合并队列 QUE_COMB = 0x1，这里对应控制的是合并队列 QC。 • 0x0：禁止 TRIGE 触发 Q1 或 QC • 0x1：使能 TRIGE 触发 Q1 或 QC
20	R/W	0x0	Q1_TRIGA Queue 1 Trigger A Enable，队列 1 EPWM 采样触发 A 使能信号。另外当使能了合并队列 QUE_COMB = 0x1，这里对应控制的是合并队列 QC。 • 0x0：禁止 EPWM_ADC_CVRA 触发 Q1 或 QC • 0x1：使能 EPWM_ADC_CVRA 触发 Q1 或 QC。关于 EPWM_ADC_CVRA 触发信号所代表的含义，请参考 EPWM 模块的配置。
19	–	–	–
18	R/W	0x0	Q1_INTE Queue 1 Interrupt Enable，队列 1 中断使能。另外当使能了合并队列 QUE_COMB = 0x1，这里对应控制的是合并队列 QC。 • 0x0：禁止 Q1/QC_INT 中断 • 0x1：使能 Q1/QC_INT 中断
17	R/W	0x0	Q1_INTM Queue 1 Interrupt Mode，队列 1 中断模式选择。另外当使能了合并队列 QUE_COMB = 0x1，这里对应控制的是合并队列 QC。

默认值：0x00002F00			ADC 模块控制 (ADC Module Control Register)
位域	类型	默认值	描述
			• 0x0：每次队列执行结束都会产生一次 Q1/QC_INT 中断 • 0x1：每隔一次（每两次）队列执行结束就会产生一次 Q1/QC_INT 中断。
16	R/WAC	0x0	Q1_RST Queue 1 Reset，队列 1 复位信号。另外当使能了合并队列 QUE_COMB = 0x1，这里对应控制的是合并队列 QC。 • 0x0：无使能 • 0x1：立刻复位 Q1/QC 到 Q1 的节点 0，下一次触发发生后，从节点 0 开始执行采样，复位后现有未执行的采样命令立刻放弃，并等待已执行的采样命令完成后，硬件自动清零本字段。
15:8	R/W	0x2F	ADC_ACQ ADC Acquirement Time，ADC 捕获时间。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，T_{ADC_ACQ} = （ADC_ACQ+1）*$T_{ADC\_ANACNT\_DIVCLK}$。假设 ADC_ANACNT_DIVCLK=24MHz，ADC_ACQ=0x2F，T_{ADC_ACQ} = (47+1) × 41.67 ns = 2 us。 注意：VTS与THS固定使用ADC_ACQ 注意：ADC采集VTS或外部电压： VCM=1,T_{ADC_ACQ}需大于500ns。 VCM=8,T_{ADC_ACQ}需大于250ns。 VCM=15,T_{ADC_ACQ}需大于125ns。 注意：ADC采集THS：T_{ADC_ACQ}需大于2000ns。 推荐ADC_ACQ留给VTS与THS使用。 推荐PSADC采集外部电压使用ADC_ACQ_1、ADC_ACQ_2、ADC_ACQ_3，不受THS T_{ADC_ACQ}的需求影响
7:4	–	–	–
3	R/W	0x0	ADC_SMP_SYC_EN ADC Queue Sample Sync Enable，ADC 队列采样同步使能。此字段只有 ADC0 拥有。 • 0x0：未使能 • 0x1：使能，使能后 ADC0 的 Q1 和 ADC1 的 Q1 进行同步，ADC0 的 Q2 和 ADC1 的 Q2 进行同步，需要确保 ADC0 的的队列长度大于 ADC1 的队列长度。
2	R/W	0x0	QUE_WRAP Queue Wrap Around Mode，队列回环模式

默认值：0x00002F00			ADC 模块控制 (ADC Module Control Register)
位域	类型	默认值	描述
			• 0x0：回环从节点 0 到节点 Q1/Q2_TRIG_CNT • 0x1：回环从节点 0 到最大节点
1	R/W	0x0	QUE_COMB Queue Combine，合并队列使能。 • 0x0：两个采样队列相互独立，队列 1（Q1）和队列 2（Q2）各有 12 个节点 • 0x1：合并两个队列，Q1 与 Q2 合并后，统一作为 Q1 进行控制，变成单个 24 个节点的队列（QC）
0	R/W	0x0	ADC_EN ADC Enable，ADC 使能。 • 0x0：ADC 被禁止 • 0x1：ADC 被使能

9.4.5.2. 0x004 ADC_TCR

默认值：0x00000000			ADC 队列采样节点数 (ADC Trigger Count Register)
位域	类型	默认值	描述
31:16	–	–	–
15:8	R/W	0x0	Q2_TRIG_CNT（队列 2 触发采样节点数） 本字段设定了队列 2 的每次触发信号发生后，采样队列的连续节点数，实际采样节点数为 Q2_TRIG_CNT+1。 • 0x0：每次触发执行 1 个采样节点 • 0x1：每次触发执行 2 个采样节点 • … • 0xB：每次触发执行 12 个采样节点 • others：每次触发执行 12 个采样节点
7:0	R/W	0x0	Q1_TRIG_CNT（队列 1 触发采样节点数） 本字段设定了队列 1（包括合并队列）的每次触发信号发生后，采样队列的连续节点数，实际采样节点数为 Q1_TRIG_CNT+1。 • 0x0：每次触发执行 1 个采样节点 • 0x1：每次触发执行 2 个采样节点 • …

默认值：0x00000000			ADC 队列采样节点数 (ADC Trigger Count Register)
位域	类型	默认值	描述
			• 0x17：每次触发执行 24 个采样节点 • others：每次触发执行 24 个采样节点

9.4.5.3. 0x010 ADC_MSR

默认值：0x00000000			ADC 模块状态 (ADC Module State Register)
位域	类型	默认值	描述
31:26	RO	0x0	QX_CNT 本字段显示内部状态机的当前队列剩余采样节点数的计数值。 –
25:21	RO	0x0	Q2_STATE Queue 2 State，队列 2 状态。 本字段显示当前队列 2 所处的采样节点位置。初始为 0，显示队列 2 处于 NODE0 的位置，随着触发采样的递进，Q2_STATE 会不断往前计数，例如 Q2_STATE=3，就表示下一次会对 NODE3 描述的通道号进行采样。
20:16	RO	0x0	Q1_STATE Queue 1 State，队列 1 状态。 本字段显示当前队列 1（包括合并队列）所处的采样节点位置。初始为 0，显示队列 1 处于 NODE0 的位置，随着触发采样的递进，Q1_STATE 会不断往前计数，例如 Q1_STATE=5，就表示下一次会对 NODE5 描述的通道号进行采样。
15:12	–	–	–
11	RO	0x0	Q2_EOQ Queue 2 End of Queue，队列 2 结束翻转标志。 本字段仅在中断模式 1（Q2_INTM=1）有效，在中断模式 0（Q2_INTM=0）下，本字段固定为 0。在中断模式 1 下，每次按照 Q2_TRIG_CNT 计数完成相应的采样节点数后就会自动翻转一次（由 0 翻转为 1，或者由 1 翻转为 0）。
10	RO	0x0	Q1_EOQ Queue 1 End of Queue，队列 1 结束翻转标志。 本字段仅在中断模式 1（Q1_INTM=1）有效，在中断模式 0（Q1_INTM=0）下，本字段固定为 0。在中断模式 1 下，每次按照 Q1_TRIG_CNT 计数完成相应的采样节点数后就会自动翻转一次（由 0 翻转为 1，或者由 1 翻转为 0）。
9	RO	0x0	Q2_BUSY Queue 2 Busy Flag 有，队列 2 忙状态标志。

默认值：0x00000000			ADC 模块状态 (ADC Module State Register)
位域	类型	默认值	描述
			• 0x0：Q2 处于空闲状态，等待触发信号 • 0x1：Q2 处于采样过程中
8	RO	0x0	Q1_BUSY Queue 1 Busy Flag，队列 1 忙状态标志。 • 0x0：Q1（或 QC）处于空闲状态，等待触发信号 • 0x1：Q1（或 QC）处于采样过程中
7	–	–	–
6	RO	0x0	Q2_FERR Queue 2 FIFO Error Flag，队列 2 FIFO 异常标志。
5	RO	0x0	Q2_FRDY Queue 2 FIFO Ready Flag，队列 2 FIFO 数据就绪标志。
4	R/W1C	0x0	Q2_INT Queue 2 Interrupt Flag，队列 2 完成事件标志。 • 0x0：无完成事件 • 0x1：有完成事件，本字段可以通过软件写‘1’清零
3	–	–	–
2	RO	0x0	Q1_FERR Queue 1 FIFO Error Flag，队列 1 FIFO 异常中断标志。在合并队列的情况下，本字段为 QC 队列的状态。 本字段为只读，FIFO 上/下溢出状态有效（FIFO_OFS 或 UFS），硬件就会自动置位本字段。当 FIFO 的上/下溢出状态消除，此位自动清 0。
1	RO	0x0	Q1_FRDY Queue 1 FIFO Ready Flag，队列 1 FIFO 数据就绪标志。在合并队列的情况下，本字段为指示队列 QC 的 FIFO 数据就绪标志。 本字段为只读， FIFO 数据就绪，硬件就会置位本字段。当 FIFO 有效数据量低于 FIFO 数据就绪阈值（FIFO_DCNT<FIFO_DRTH），此位自动清 0。
0	R/W1C	0x0	Q1_INT Queue 1 Interrupt Flag，队列 1 完成事件标志。在合并队列的情况下，本字段为指示队列 QC 的完成事件标志。 • 0x0：无完成事件 • 0x1：有完成事件，本字段可以通过软件写‘1’清零

9.4.5.4. 0x014 ADC_CALCSR

默认值：0x20002F00			ADC 校准控制寄存器
位域	类型	默认值	描述
31	R/WAC	0x0	CALVAL_UPD 通过软件设置 CALVAL 的同时，置位本字段，表示软件需要更新 CALVAL 值，本字段被置位后会触发内部更新的动作，硬件更新完毕之后，会自动清零本字段。
30	–	–	–
29:16	R/W	0x2000	CALVAL Calibration Value，ADC 校准输出值。 在校准过程中，ADC 会采样芯片内部参考电压的电平，获得一个原始值，保存在本字段。硬件使用这个值对所有的采样数据进行校准，以便消除生产工艺引起的偏移误差。 如果希望使用一个非硬件自动校准取得的值对 ADC 数据进行校准，可以通过软件设置一个合适的值，硬件就会参考本字段，对 ADC 采样数据进行自动校准。
15:8	R/W	0x2F	ADC_ACQ_CAL ADC Acquirement Time，用于 ADC 校准的捕获时间。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，$T_{ADC\_ACQ\_CAL}$ = (ADC_ACQ_CAL +1) *$T_{ADC\_ANACNT\_DIVCLK}$。假设 ADC_ANACNT_DIVCLK=24 MHz，ADC_ACQ_CAL= 0x2F，$T_{ADC\_ACQ\_CAL}$ = (47+1) × 41.67 ns = 2 us。
7:5	–	–	–
4	R/W	0x0	ADC_CAL_MODE • 0x0：方式1 • 0x1：方式2（推荐值）
3:2	–	–	–
1	R/W	0x0	DCALMASK Data Calibration Mask，采样数据校准屏蔽。 • 0x0：硬件默认对采样数据进行校准后送给 ADC 应用功能模块 • 0x1：屏蔽采样数据校准功能，本模块直接把来自 ADC 模拟电路的采样数据不经校准地送给 ADC 应用功能模块
0	R/WAC	0x0	CALEN Calibration Enable，ADC 校准使能

默认值：0x20002F00			ADC 校准控制寄存器
位域	类型	默认值	描述
			本使能信号由软件置位，使能 ADC 校准，ADC 校准完后，对此控制信号自动清零。软件查询到此信号被清零后，意味着校准已完成。

9.4.5.5. 0x018 ADC_GAINCSR

默认值：0x00000000			增益控制寄存器
位域	类型	默认值	描述
31	–	–	–
30	R/W	0x1	ADC_GAIN_BYPASS • 0x0：增益计算使能 • 0x1：增益计算旁通
29	R/W	0x1	VTS_GAIN_BYPASS • 0x0：增益计算使能 • 0x1：增益计算旁通
28	R/W	0x1	THS_GAIN_BYPASS • 0x0：增益计算使能 • 0x1：增益计算旁通
27:17	–	–	–
16:0	R/W	0x10000	GAIN_SET 对 ADC 数据进行增益补偿，通过软件设置一个合适的值，硬件就会参考本字段，对 ADC 采样数据进行增益补偿。 硬件计算公式：(ADC_VALUE*GAIN_SET)>>16

9.4.5.6. 0x01C ADC_TRIGE_FILTER

默认值：0x00000000			TRIGE 输入滤波控制(TRIGE Filter)
位域	类型	默认值	描述
31:16	–	–	–
15:8	R/W	0x0	DEBNCE_DIV_CLK

默认值：0x00000000			TRIGE 输入滤波控制(TRIGE Filter)
位域	类型	默认值	描述
			Second Debounce Clock Divison，二级去抖分频器。仅对 ADC_TRIG 有效（pad 输入） 本计数单位为 TPCLK。当本字段为 0（即 0x0000）时，去抖功能无效。本字段定义的去抖时间范围为 0 ~（2^8–1）*TPCLK。
7:4	R/W	0x0	DEBNCE_CNT_FST First Debounce Counter，第一级去抖计数器。仅对 ADC_TRIG 有效（pad 输入） 本计数单位为 divclk。当本字段为 0（即 0x0000）时，去抖功能无效。本字段定义的去抖时间范围为 0 ~（2^4–1）*divclk。
3:0	R/W	0x0	DEBNCE_CNT_SEC Second Debounce Counter，第二级去抖计数器。仅对 ADC_TRIG 有效（pad 输入） 本计数单位为 divclk。当本字段为 0（即 0x0000）时，去抖功能无效。本字段定义的去抖时间范围为 0 ~（2^4–1）*divclk。

9.4.5.7. 0x020 ADC_Q1FCR

默认值：0x00000000			ADC 队列 1 FIFO 控制(ADC Queue 1 FIFO Control Register)
位域	类型	默认值	描述
31:30	–	–	–
29:24	RO	0x0	Q1_FIFO_DCNT Queue 1 FIFO Data Count，队列 1 的 FIFO 有效数据量。Q1 FIFO 的总深度为 36，因此 0x0~0x24 为有效。在使能合并队列后，这里指示的合并队列 QC 的状态。
23:18	–	–	–
17	R/W1C	0x0	Q1_FIFO_UFS Queue 1 FIFO Underflow Status，队列 1 FIFO 下溢出状态，高电平有效。本字段由硬件置位，并由软件写 1 清零。在使能合并队列后，这里指示的合并队列 QC 的状态。
16	R/W1C	0x0	Q1_FIFO_OFS Queue 1 FIFO Overflow Status，队列 1 FIFO 上溢出状态，高电平有效。本字段由硬件置位，并由软件写 1 清零。在使能合并队列后，这里指示的合并队列 QC 的状态。

默认值：0x00000000			ADC 队列 1 FIFO 控制(ADC Queue 1 FIFO Control Register)
位域	类型	默认值	描述
15:14	–	–	–
13:8	R/W	0x0	Q1_FIFO_DRTH Queue 1 FIFO Data Ready Threshold，队列 1 FIFO 数据就绪阈值。 当 FIFO 的数据量达到了本阈值后，就有会产生数据就绪中断或者 DMA 请求。0x01~0x24 表示有效阈值范围。其它为非法配置，会导致无中断/DMA 请求。
7:4	–	–	–
3	R/W	0x0	Q1_FIFO_ERRIE Queue 1 FIFO Error Interrupt Enable，队列 1 的 FIFO 异常中断使能。 • 0x0：禁止产生中断 • 0x1：允许产生中断。
2	R/W	0x0	Q1_FIFO_DRQE Queue 1 FIFO DMA Request Eanble，队列 1 的 DMA 数据请求使能。 • 0x0：禁止向 DMA 发出 DRQ • 0x1：允许向 DMA 发出 DRQ。
1	R/W	0x0	Q1_FIFO_RDYIE Queue 1 FIFO Ready Interrupt Enable，队列 1 的 FIFO 数据就绪中断使能。 • 0x0：禁止产生中断 • 0x1：允许产生中断。
0	R/WAC	0x0	Q1_FIFO_FLUSH Queue 1 FIFO Flush，队列 1 FIFO 清空操作。 本字段由软件置位硬件清零。软件置位启动 FIFO 清空操作，FIFO 清空操作完成后，由硬件清零。

9.4.5.8. 0x024 ADC_Q2FCR

默认值：0x00000000			ADC 队列 2 FIFO 控制(ADC Queue 2 FIFO Control Register)
位域	类型	默认值	描述
31:29	–	–	–
28:24	RO	0x0	Q2_FIFO_DCNT

默认值：0x00000000			ADC 队列 2 FIFO 控制(ADC Queue 2 FIFO Control Register)
位域	类型	默认值	描述
			Queue 2 FIFO Data Count，队列 2 的 FIFO 有效数据量。Q2 FIFO 的总深度为 16，因此 0x0~0x10 为有效。
23:18	–	–	–
17	R/W1C	0x0	Q2_FIFO_UFS Queue 2 FIFO Underflow Status，队列 2 FIFO 下溢出状态，高电平有效。 本字段由硬件置位，并由软件写 1 清零。
16	R/W1C	0x0	Q2_FIFO_OFS Queue 2 FIFO Overflow Status，队列 2 FIFO 上溢出状态，高电平有效。 本字段由硬件置位，并由软件写 1 清零。
15:12	–	–	–
11:8	R/W	0x0	Q2_FIFO_DRTH Queue 2 FIFO Data Ready Threshold，队列 2 FIFO 数据就绪阈值。当 FIFO 的数据量达到了本阈值后，就有会产生数据就绪中断或者 DMA 请求。 0x1~0x10 表示有效阈值范围。其它为非法配置，会导致无中断/DMA 请求。
7:4	–	–	–
3	R/W	0x0	Q2_FIFO_ERRIE Queue 2 FIFO Error Interrupt Enable，队列 2 的 FIFO 异常中断使能。 • 0x0：禁止产生中断 • 0x1：允许产生中断。
2	R/W	0x0	Q2_FIFO_DRQE Queue 2 FIFO DMA Request Eanble，队列 2 的 DMA 数据请求使能。 • 0x0：禁止向 DMA 发出 DRQ。 • 0x1：允许向 DMA 发出 DRQ。
1	R/W	0x0	Q2_FIFO_RDYIE Queue 2 FIFO Ready Interrupt Enabel，队列 2 的 FIFO 数据就绪中断使能。 • 0x0：禁止产生中断。 • 0x1：允许产生中断。
0	R/WAC	0x0	Q2_FIFO_FLUSH Queue 2 FIFO Flush，队列 2 FIFO 清空操作。

默认值：0x00000000			ADC 队列 2 FIFO 控制(ADC Queue 2 FIFO Control Register)
位域	类型	默认值	描述
			本字段由软件置位硬件清零。软件置位启动 FIFO 清空操作，FIFO 清空操作完成后，由硬件清零。

9.4.5.9. 0x028 ADC_SMP_CFG

默认值：0x002F2F2F			ADC 采样时间配置寄存器
位域	类型	默认值	描述
31:24	R/W	0x3	ADC_CLR_THR 本字段定义复位ADC内部信号的时间长度。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，$T_{ADC\_CLR\_THR}$ = （ADC_CLR_THR +1）*$T_{ADC\_ANACNT\_DIVCLK}$。 注意：ADC_CLR_THR需要小于ADC_ACQ
23:16	R/W	0x2F	ADC_ACQ_3 本字段定义捕获时间长度。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，$T_{ADC\_ACQ\_3}$ = （ADC_ACQ_3+1）*$T_{ADC\_ANACNT\_DIVCLK}$。假设 ADC_ANACNT_DIVCLK = 24 MHz，ADC_ACQ_3 = 0x2F，$T_{ADC\_ACQ\_3}$ = (47+1) × 41.67 ns = 2 us。
15:8	R/W	0x2F	ADC_ACQ_2 本字段定义捕获时间长度。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，$T_{ADC\_ACQ\_2}$ = （ADC_ACQ_2+1）*$T_{ADC\_ANACNT\_DIVCLK}$。假设 ADC_ANACNT_DIVCLK = 24 MHz，ADC_ACQ_2 = 0x2F，$T_{ADC\_ACQ\_2}$ = (47 + 1)*41.67 ns = 2 us。
7:0	R/W	0x2F	ADC_ACQ_1 本字段定义捕获时间长度。本计数值单位为 ADC_ANACNT_DIVCLK 的周期数，$T_{ADC\_ACQ\_1}$ = （ADC_ACQ_1+1）*$T_{ADC\_ANACNT\_DIVCLK}$。假设 ADC_ANACNT_DIVCLK = 24 MHz，ADC_ACQ_1 = 0x2F，$T_{ADC\_ACQ\_1}$ = (47 + 1) * 41.67 ns = 2 us。

9.4.5.10. 0x02C ADC_SHIFT

默认值：0x00000000			ADC 移位控制寄存器
位域	类型	默认值	描述
31:4	–	–	–
3:0	R/W	0x0	ADC_SHIFT

默认值：0x00000000			ADC 移位控制寄存器
位域	类型	默认值	描述
			本字段定义 ADC 的右移位数，此字段对于 ADC 的队列 Q1、Q2、QC 和 QN 的转换数据进行控制。 • 0x0：无右移 • 0x1：右移 1 位 • 0x2：右移 2 位 • ... • 0x8：右移 8 位 • others：无效

9.4.5.11. 0x040 ADC_Q1FDR

默认值：0x00000000			ADC 队列 1 FIFO 数据 (ADC Queue 1 FIFO Data Register)
位域	类型	默认值	描述
31:22	–	–	–
21:16	RO	0x0	Q1_CHNUM ADC Sample Channel Number，ADC 队列 1 采样通道号。 本字段为 DATA 对应的通道号，与 NODEx 中的数值对应。本字段的主要是目的是 Debug 使用，接收到数据后可以与 NODEx 中的数值进行比较，判断控制器是否出错。
15:0	RO	0x0	Q1_DATA ADC Sample Q1 Data，ADC 采样队列 1 数据。本字段为 12–bit ADC 采样后转化为数字量的数据。

9.4.5.12. 0x080 ADC_Q2FDR

默认值：0x00000000			ADC 队列 2 FIFO 数据 (ADC Queue 2 FIFO Data Register)
位域	类型	默认值	描述
31:16	–	–	–
21:16	RO	0x0	Q2_CHNUM ADC Sample Channel Number，ADC 队列 2 采样通道号。

默认值：0x00000000			ADC 队列 2 FIFO 数据 (ADC Queue 2 FIFO Data Register)
位域	类型	默认值	描述
			本字段为 DATA 对应的通道号，与 NODEx 中的数值对应。本字段的主要是目的是 Debug 使用，接收到数据后可以与 NODEx 中的数值进行比较，判断控制器是否出错。
15:0	RO	0x0	Q2_DATA ADC Sample Q2 Data，ADC 采样队列 2 数据。本字段为 12-bit ADC 采样后转化为数字量的数据。

9.4.5.13. 0x090 ADC_HCL_CTL

默认值：0x00000000			ADC 硬件电流环控制寄存器
位域	类型	默认值	描述
31:1	–	–	–
0	R/W	0x0	HCL_RD_EN Hardware Current Loop Read Enable，使能硬件电流环读取 ADC 数据。 • 0x0：未使能 • 0x1：使能

9.4.5.14. 0x094 ADC_CTL

默认值：0x2fff0100			ADC 控制寄存器
位域	类型	默认值	描述
31:30	R/W	0x2	ADC_V165_SEL • bit[31] 为内部 1.65V 电压驱动能力，若 bit[31] = 0，驱动能力最弱100uA，否则驱动能力最强300uA。 • bit[30] 为 Vcm sel： ◦ 0：VCM 由状态机动态控制开关，值由 VCM_SET 控制。 ◦ 1：VCM 常开，值由 VCM_SET 控制。
29:28	R/W	0x0	THS_BIAS THS运放偏置电流设置，(单位uA) I_{THS_BIAS}= THS_BIAS*0.5+3
27:16	R/W	0xfff	THS_ACQ THS 温度电压转换时间 = (THS_ACQ+1) × ADC_ANACNT_DIVCLK

默认值：0x2fff0100			ADC 控制寄存器
位域	类型	默认值	描述
			注意：温度电压转换时间需要大于 1.2 us。
15	–	–	–
14	R/W	0x0	EN_CAP_CAL • 0x0：关闭电容阵列补偿 • 0x1：开启电容阵列补偿 推荐开启，能消除由于生产工艺引起的电容失配，开启能提升ENOB
13	R/W	0x0	Vin_compen • 0x0：关闭输入信号比较器 • 0x1：开启输入信号比较器
12:9	R/W	0x8	VCM_SET 共模电压驱动能力设置（单位 uA） I_{Vref} = VCM_SET*100+400
8:5	R/W	0x8	ADC_BIAS ADC 运放偏置电流设置（单位 uA） 0~7：I_{ADC_BIAS} = ADC_BIAS 8~15：I_{ADC_BIAS} = ADC_BIAS –1
4:0	–	–	–

9.4.5.15. 0x098 ADC_CTL1

默认值：0x04000000			ADC 控制寄存器 1
位域	类型	默认值	描述
31:29	–	–	–
28	R	0x0	PREAMP_DONE 运放预先计数器是否完成 • 0x0：未完成 • 0x1：完成
27:16	R/W	0x400	ADC_PREAMP_SET ADC 运放预先计数时间=(ADC_PREAMP_SET+1)*ADC_ANACNT_DIVCLK

默认值：0x04000000			ADC 控制寄存器 1
位域	类型	默认值	描述
			硬件自动实现一个计数功能，当计数器到达设定值，硬件自动把PREAMP_DONE置1，此时才可以触发ADC采样。 ADC 运放预先计数时间需大于10us（即ADC相关寄存器配置完成之后，ADC至少需要延时10us,才可以开始采样）
15:12	R/W	0x1	CLK_LOW_CYCLE sar 比较阶段的时钟 (ADC_COMP) 周期及占空比配置：时钟周期低电平持续时间单位 ahb_clk T=(CLK_LOW_CYCLE +1) * Tahb_clk
11:8	R/W	0x2	CLK_HIGH_CYCLE sar 比较阶段的时钟 (ADC_COMP) 周期及占空比配置：时钟周期高电平持续时间单位 ahb_clk T=(CLK_HIGH_CYCLE +1) * Tahb_clk
7:5	–	–	–
4:1	R/W	0x8	ADC_ANACNT_DIVCLK ADC 模拟计数分频时钟 = AHB_CLK / (ADC_ANACNT_DIVCLK + 1) ADC_ACQ、THS_ACQ、PREAMP SET 均采用 ADC_ANACNT_DIVCLK 来计数
0	–	–	–

9.4.5.16. 0x09C ADC_CAL_RS

默认值：0x00000000			校准结果寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:16	R	0x0	Vrefp CAL Result Vrefp 校准结果
15:14	–	–	–
13:0	R/W	0x0	Vrefn CAL Result Vrefn 校准结果

9.4.5.17. 0x0A0 ADC_Q1_NOD03

默认值：0x00000000			采样队列 1 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q1_NODE3 本字段定义了 Q1 节点 3 或 QC 节点 3 的 ADC 输入通道号。
23:16	R/W	0x0	Q1_NODE2 本字段定义了 Q1 节点 2 或 QC 节点 2 的 ADC 输入通道号。
15:8	R/W	0x0	Q1_NODE1 本字段定义了 Q1 节点 1 或 QC 节点 1 的 ADC 输入通道号。
7:0	R/W	0x0	Q1_NODE0 本字段定义了 Q1 节点 0 或 QC 节点 0 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道 • 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0 • 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.18. 0x0A4 ADC_Q1_NOD47

默认值：0x00000000			采样队列 1 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q1_NODE7 本字段定义了 Q1 节点 7 或 QC 节点 7 的 ADC 输入通道号。

默认值：0x00000000			采样队列 1 节点寄存器
位域	类型	默认值	描述
23:16	R/W	0x0	Q1_NODE6 本字段定义了 Q1 节点 6 或 QC 节点 6 的 ADC 输入通道号。
15:8	R/W	0x0	Q1_NODE5 本字段定义了 Q1 节点 5 或 QC 节点 5 的 ADC 输入通道号。
7:0	R/W	0x0	Q1_NODE4 本字段定义了 Q1 节点 4 或 QC 节点 4 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道 • 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0 • 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.19. 0x0A8 ADC_Q1_NOD811

默认值：0x00000000			采样队列 1 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q1_NODE11 本字段定义了 Q1 节点 11 或 QC 节点 11 的 ADC 输入通道号。
23:16	R/W	0x0	Q1_NODE10 本字段定义了 Q1 节点 10 或 QC 节点 10 的 ADC 输入通道号。
15:8	R/W	0x0	Q1_NODE9 本字段定义了 Q1 节点 9 或 QC 节点 9 的 ADC 输入通道号。

默认值：0x00000000			采样队列 1 节点寄存器
位域	类型	默认值	描述
7:0	R/W	0x0	Q1_NODE8 本字段定义了 Q1 节点 8 或 QC 节点 8 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道 • 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0 • 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.20. 0x0B0 ADC_Q2_NOD03

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q2_NODE3 本字段定义了 Q2 节点 3 或 QC 节点 15 的 ADC 输入通道号。
23:16	R/W	0x0	Q2_NODE2 本字段定义了 Q2 节点 2 或 QC 节点 14 的 ADC 输入通道号。
15:8	R/W	0x0	Q2_NODE1 本字段定义了 Q2 节点 1 或 QC 节点 13 的 ADC 输入通道号。
7:0	R/W	0x0	Q2_NODE0 本字段定义了 Q2 节点 0 或 QC 节点 12 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
			• ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道 • 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0 • 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.21. 0x0B4 ADC_Q2_NOD47

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q2_NODE7 本字段定义了 Q2 节点 7 或 QC 节点 19 的 ADC 输入通道号。
23:16	R/W	0x0	Q2_NODE6 本字段定义了 Q2 节点 6 或 QC 节点 18 的 ADC 输入通道号。
15:8	R/W	0x0	Q2_NODE5 本字段定义了 Q2 节点 5 或 QC 节点 17 的 ADC 输入通道号。
7:0	R/W	0x0	Q2_NODE4 本字段定义了 Q1 节点 4 或 QC 节点 16 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
			• 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0 • 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.22. 0x0B8 ADC_Q2_NOD811

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	Q2_NODE11 本字段定义了 Q2 节点 11 或 QC 节点 23 的 ADC 输入通道号。
23:16	R/W	0x0	Q2_NODE10 本字段定义了 Q2 节点 10 或 QC 节点 22 的 ADC 输入通道号。
15:8	R/W	0x0	Q1_NODE9 本字段定义了 Q2 节点 9 或 QC 节点 21 的 ADC 输入通道号。
7:0	R/W	0x0	Q2_NODE8 本字段定义了 Q2 节点 8 或 QC 节点 20 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18：选择 PGA0 输入通道 • 0x19：选择 PGA1 输入通道 • 0x1A：选择 PGA2 输入通道 • 0x1B：选择 PGA3 输入通道 • 0x1C~0x1F：无效 • 0x20：TS0/VTS0，ADC0 对应 TS0，ADC1 对应 VTS0

默认值：0x00000000			采样队列 2 节点寄存器
位域	类型	默认值	描述
			• 0x21：TS1/VTS1，ADC0 对应 TS1，ADC1 对应 VTS1 • 0x22：TS2/VTS2，ADC0 对应 TS2，ADC1 对应 VTS2 • 0x23：TS3/VTS3，ADC0 对应 TS3，ADC1 对应 VTS3 • Others：无效

9.4.5.23. 0x0C0 ADC_Q1_SMP_SET

默认值：0x00000000			采样队列 1 采样时间寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	Q1_NODE11_SMP_SET Q1 节点 11 或 QC 节点 11 的采样时间配置。
21:20	R/W	0x0	Q1_NODE10_SMP_SET Q1 节点 10 或 QC 节点 10 的采样时间配置。
19:18	R/W	0x0	Q1_NODE9_SMP_SET Q1 节点 9 或 QC 节点 9 的采样时间配置。
17:16	R/W	0x0	Q1_NODE8_SMP_SET Q1 节点 8 或 QC 节点 8 的采样时间配置。
15:14	R/W	0x0	Q1_NODE7_SMP_SET Q1 节点 7 或 QC 节点 7 的采样时间配置。
13:12	R/W	0x0	Q1_NODE6_SMP_SET Q1 节点 6 或 QC 节点 6 的采样时间配置。
11:10	R/W	0x0	Q1_NODE5_SMP_SET Q1 节点 5 或 QC 节点 5 的采样时间配置。
9:8	R/W	0x0	Q1_NODE4_SMP_SET Q1 节点 4 或 QC 节点 4 的采样时间配置。
7:6	R/W	0x0	Q1_NODE3_SMP_SET Q1 节点 3 或 QC 节点 3 的采样时间配置。
5:4	R/W	0x0	Q1_NODE2_SMP_SET

默认值：0x00000000			采样队列 1 采样时间寄存器
位域	类型	默认值	描述
			Q1 节点 2 或 QC 节点 2 的采样时间配置。
3:2	R/W	0x0	Q1_NODE1_SMP_SET Q1 节点 1 或 QC 节点 1 的采样时间配置。
1:0	R/W	0x0	Q1_NODE0_SMP_SET Q1 节点 0 或 QC 节点 0 的采样时间配置。 • 0x0：采样时间 1，根据寄存器 ADC_ACQ 去配置采样时间 • 0x1：采样时间 2，根据寄存器 ADC_ACQ_1 去配置采样时间 • 0x2：采样时间 3，根据寄存器 ADC_ACQ_2 去配置采样时间 • 0x3：采样时间 4，根据寄存器 ADC_ACQ_3 去配置采样时间

9.4.5.24. 0x0C4 ADC_Q2_SMP_SET

默认值：0x00000000			采样队列 2 采样时间寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	Q2_NODE11_SMP_SET Q2 节点 11 或 QC 节点 23 的采样时间配置。
21:20	R/W	0x0	Q2_NODE10_SMP_SET Q2 节点 10 或 QC 节点 22 的采样时间配置。
19:18	R/W	0x0	Q2_NODE9_SMP_SET Q2 节点 9 或 QC 节点 21 的采样时间配置。
17:16	R/W	0x0	Q2_NODE8_SMP_SET Q2 节点 8 或 QC 节点 20 的采样时间配置。
15:14	R/W	0x0	Q2_NODE7_SMP_SET Q2 节点 7 或 QC 节点 19 的采样时间配置。
13:12	R/W	0x0	Q2_NODE6_SMP_SET Q2 节点 6 或 QC 节点 18 的采样时间配置。
11:10	R/W	0x0	Q2_NODE5_SMP_SET Q2 节点 5 或 QC 节点 17 的采样时间配置。

默认值：0x00000000			采样队列 2 采样时间寄存器
位域	类型	默认值	描述
9:8	R/W	0x0	Q2_NODE4_SMP_SET Q2 节点 4 或 QC 节点 16 的采样时间配置。
7:6	R/W	0x0	Q2_NODE3_SMP_SET Q2 节点 3 或 QC 节点 15 的采样时间配置。
5:4	R/W	0x0	Q2_NODE2_SMP_SET Q2 节点 2 或 QC 节点 14 的采样时间配置。
3:2	R/W	0x0	Q2_NODE1_SMP_SET Q2 节点 1 或 QC 节点 13 的采样时间配置。
1:0	R/W	0x0	Q2_NODE0_SMP_SET Q2 节点 0 或 QC 节点 12 的采样时间配置。 • 0x0：采样时间 1，根据寄存器 ADC_ACQ 去配置采样时间 • 0x1：采样时间 2，根据寄存器 ADC_ACQ_1 去配置采样时间 • 0x2：采样时间 3，根据寄存器 ADC_ACQ_2 去配置采样时间 • 0x3：采样时间 4，根据寄存器 ADC_ACQ_3 去配置采样时间

9.4.5.25. 0x0C8 ADC_Q1_SMP_SET0

默认值：0x00000000			采样队列 1 过采样配置 0 寄存器
位域	类型	默认值	描述
31	–	–	–
30:28	R/W	0x0	Q1_NODE7_OSR Q1 节点 7 或 QC 节点 7 的过采样配置。
27	–	–	–
26:24	R/W	0x0	Q1_NODE6_OSR Q1 节点 6 或 QC 节点 6 的过采样配置。
23	–	–	–
22:20	R/W	0x0	Q1_NODE5_OSR Q1 节点 5 或 QC 节点 5 的过采样配置。

默认值：0x00000000			采样队列 1 过采样配置 0 寄存器
位域	类型	默认值	描述
19	–	–	–
18:16	R/W	0x0	Q1_NODE4_OSR Q1 节点 4 或 QC 节点 4 的过采样配置。
15	–	–	–
14:12	R/W	0x0	Q1_NODE3_OSR Q1 节点 3 或 QC 节点 3 的过采样配置。
11	–	–	–
10:8	R/W	0x0	Q1_NODE2_OSR Q1 节点 2 或 QC 节点 2 的过采样配置。
7	–	–	–
6:4	R/W	0x0	Q1_NODE1_OSR Q1 节点 1 或 QC 节点 1 的过采样配置。
3	–	–	–
2:0	R/W	0x0	Q1_NODE0_OSR Q1 节点 0 或 QC 节点 0 的过采样配置。 • 0x0：OSR = 1，单点采样 • 0x1：OSR = 2，两倍过采样 • 0x2：OSR = 4，四倍过采样 • 0x3：OSR = 8，八倍过采样 • 0x4：OSR = 16，十六倍过采样 • 0x5：OSR = 32，三十二倍过采样 • Others：无效

9.4.5.26. 0x0CC ADC_Q1_SMP_SET1

默认值：0x00000000			采样队列 1 过采样配置 0 寄存器
位域	类型	默认值	描述
31:15	–	–	–

默认值：0x00000000			采样队列 1 过采样配置 0 寄存器
位域	类型	默认值	描述
14:12	R/W	0x0	Q1_NODE11_OSR Q1 节点 11 或 QC 节点 11 的过采样配置。
11	–	–	–
10:8	R/W	0x0	Q1_NODE10_OSR Q1 节点 10 或 QC 节点 10 的过采样配置。
7	–	–	–
6:4	R/W	0x0	Q1_NODE9_OSR Q1 节点 9 或 QC 节点 9 的过采样配置。
3	–	–	–
2:0	R/W	0x0	Q1_NODE8_OSR Q1 节点 8 或 QC 节点 8 的过采样配置。 • 0x0：OSR = 1，单点采样 • 0x1：OSR = 2，两倍过采样 • 0x2：OSR = 4，四倍过采样 • 0x3：OSR = 8，八倍过采样 • 0x4：OSR = 16，十六倍过采样 • 0x5：OSR = 32，三十二倍过采样 • Others：无效

9.4.5.27. 0x0D0 ADC_Q2_SMP_SET0

默认值：0x00000000			采样队列 2 过采样配置 0 寄存器
位域	类型	默认值	描述
31	–	–	–
30:28	R/W	0x0	Q2_NODE7_OSR Q1 节点 7 或 QC 节点 19 的过采样配置。
27	–	–	–
26:24	R/W	0x0	Q2_NODE6_OSR

默认值：0x00000000			采样队列 2 过采样配置 0 寄存器
位域	类型	默认值	描述
			Q1 节点 6 或 QC 节点 18 的过采样配置。
23	–	–	–
22:20	R/W	0x0	Q2_NODE5_OSR Q1 节点 5 或 QC 节点 17 的过采样配置。
19	–	–	–
18:16	R/W	0x0	Q2_NODE4_OSR Q1 节点 4 或 QC 节点 16 的过采样配置。
15	–	–	–
14:12	R/W	0x0	Q2_NODE3_OSR Q1 节点 3 或 QC 节点 15 的过采样配置。
11	–	–	–
10:8	R/W	0x0	Q2_NODE2_OSR Q1 节点 2 或 QC 节点 14 的过采样配置。
7	–	–	–
6:4	R/W	0x0	Q2_NODE1_OSR Q1 节点 1 或 QC 节点 13 的过采样配置。
3	–	–	–
2:0	R/W	0x0	Q2_NODE0_OSR Q1 节点 0 或 QC 节点 12 的过采样配置。 • 0x0：OSR = 1，单点采样 • 0x1：OSR = 2，两倍过采样 • 0x2：OSR = 4，四倍过采样 • 0x3：OSR = 8，八倍过采样 • 0x4：OSR = 16，十六倍过采样 • 0x5：OSR = 32，三十二倍过采样 • Others：无效

9.4.5.28. 0x0D4 ADC_Q2_SMP_SET1

默认值：0x00000000			采样队列 1 过采样配置 0 寄存器
位域	类型	默认值	描述
31:15	–	–	–
14:12	R/W	0x0	Q2_NODE11_OSR Q2 节点 11 或 QC 节点 23 的过采样配置。
11	–	–	–
10:8	R/W	0x0	Q2_NODE10_OSR Q2 节点 10 或 QC 节点 22 的过采样配置。
7	–	–	–
6:4	R/W	0x0	Q2_NODE9_OSR Q2 节点 9 或 QC 节点 21 的过采样配置。
3	–	–	–
2:0	R/W	0x0	Q2_NODE8_OSR Q2 节点 8 或 QC 节点 20 的过采样配置。 • 0x0：OSR = 1，单点采样 • 0x1：OSR = 2，两倍过采样 • 0x2：OSR = 4，四倍过采样 • 0x3：OSR = 8，八倍过采样 • 0x4：OSR = 16，十六倍过采样 • 0x5：OSR = 32，三十二倍过采样 • Others：无效

9.4.5.29. 0x0D8 ADC_RES_FLT0

默认值：0x00000000			通道结果滤波器 0 寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:28	R/W	0x0	RES_CH7_FLT_SET 通道 7 滤波算法设置。

默认值：0x00000000			通道结果滤波器 0 寄存器
位域	类型	默认值	描述
27:26	–	–	–
25:24	R/W	0x0	RES_CH6_FLT_SET 通道 6 滤波算法设置。
23:22	–	–	–
21:20	R/W	0x0	RES_CH5_FLT_SET 通道 5 滤波算法设置。
19:18	–	–	–
17:16	R/W	0x0	RES_CH4_FLT_SET 通道 4 滤波算法设置。
15:14	–	–	–
13:12	R/W	0x0	RES_CH3_FLT_SET 通道 3 滤波算法设置。
11:10	–	–	–
9:8	R/W	0x0	RES_CH2_FLT_SET 通道 2 滤波算法设置。
7:6	–	–	–
5:4	R/W	0x0	RES_CH1_FLT_SET 通道 1 滤波算法设置。
3:2	–	–	–
1:0	R/W	0x0	RES_CH0_FLT_SET 通道 0 滤波算法设置。 • 0x0：无滤波 • 0x1：两点平均 • 0x2：四点平均 • 0x3：八点平均

9.4.5.30. 0x0DC ADC_RES_FLT1

默认值：0x00000000			通道结果滤波器 1 寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:28	R/W	0x0	RES_CH15_FLT_SET 通道 15 滤波算法设置。
27:26	–	–	–
25:24	R/W	0x0	RES_CH14_FLT_SET 通道 14 滤波算法设置。
23:22	–	–	–
21:20	R/W	0x0	RES_CH13_FLT_SET 通道 13 滤波算法设置。
19:18	–	–	–
17:16	R/W	0x0	RES_CH12_FLT_SET 通道 12 滤波算法设置。
15:14	–	–	–
13:12	R/W	0x0	RES_CH11_FLT_SET 通道 11 滤波算法设置。
11:10	–	–	–
9:8	R/W	0x0	RES_CH10_FLT_SET 通道 10 滤波算法设置。
7:6	–	–	–
5:4	R/W	0x0	RES_CH9_FLT_SET 通道 9 滤波算法设置。
3:2	–	–	–
1:0	R/W	0x0	RES_CH8_FLT_SET 通道 8 滤波算法设置。

默认值：0x00000000			通道结果滤波器 1 寄存器
位域	类型	默认值	描述
			• 0x0：无滤波 • 0x1：两点平均 • 0x2：四点平均 • 0x3：八点平均

9.4.5.31. 0x0E0 ADC_RES_FLT2

默认值：0x00000000			通道结果滤波器 2 寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:28	R/W	0x0	RES_CH23_FLT_SET 通道 23 滤波算法设置。
27:26	–	–	–
25:24	R/W	0x0	RES_CH22_FLT_SET 通道 22 滤波算法设置。
23:22	–	–	–
21:20	R/W	0x0	RES_CH21_FLT_SET 通道 21 滤波算法设置。
19:18	–	–	–
17:16	R/W	0x0	RES_CH20_FLT_SET 通道 20 滤波算法设置。
15:14	–	–	–
13:12	R/W	0x0	RES_CH19_FLT_SET 通道 19 滤波算法设置。
11:10	–	–	–
9:8	R/W	0x0	RES_CH18_FLT_SET 通道 18 滤波算法设置。

默认值：0x00000000			通道结果滤波器 2 寄存器
位域	类型	默认值	描述
7:6	–	–	–
5:4	R/W	0x0	RES_CH17_FLT_SET 通道 17 滤波算法设置。
3:2	–	–	–
1:0	R/W	0x0	RES_CH8_FLT_SET 通道 16 滤波算法设置。 • 0x0：无滤波 • 0x1：两点平均 • 0x2：四点平均 • 0x3：八点平均

9.4.5.32. 0x0100+0x4*n(n=0/1/.../23) ADC_RES_CHn

默认值：0x00000000			通道 n 寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	ADC_RES_CHn 本字段定义为通道 n 的采样转换结果。每完成一次本通道的采样转换后，此处更新结果。

9.4.5.33. 0x0180+0x4*n(n=0/1/.../23) ADC_OFF_CHn

默认值：0x00000000			通道 n 偏置调整寄存器
位域	类型	默认值	描述
31:21	–	–	–
20	R/W	0x0	ADC_OFF_EN ADC Offset Enable，偏置使能。

默认值：0x00000000			通道 n 偏置调整寄存器
位域	类型	默认值	描述
			• 0x0：未使能，ADC 通道的数据未经过偏置、符号和移位的操作，数据为无符号数 • 0x1：使能，ADC 通道的数据经过偏置、符号和移位的操作，数据为有符号数
19:17	R/W	0x0	RES_LS_BIT_SET Result Left Shift Bit Set，结果左移处理。 • 0x0：无操作 • 0x1：左移 1 位 • 0x2：左移 2 位 • 0x3：左移 3 位 • 0x4：左移 4 位 • others：Reserved
16	R/W	0x0	SIGN_EN Sign Inverse Enable，符号翻转使能。
15:0	R/W	0x0	ADC_OFF_CHn 本字段定义为通道 n 的偏置值，用于偏置调整。

9.4.5.34. 0x0220 ADC_WDG0_CFG

默认值：0x00000000			看门狗 0 配置寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:16	R/W	0x0	ADC_WDG0_SRC_SEL ADC Watchdog0 Source Select，ADC 看门狗 0 的输入源选择。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • others：Reserved
15:6	–	–	–

默认值：0x00000000			看门狗 0 配置寄存器
位域	类型	默认值	描述
5	R/W	0x0	ADC_WDG0_NZD_EN ADC Watchdog0 Negative Zero Detect Enable，看门狗反向过 0 检测使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	ADC_WDG0_PZD_EN ADC Watchdog0 Positive Zero Detect Enable，看门狗正向过 0 检测使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	ADC_WDG0_IRQ_EN ADC Watchdog0 IRQ Enable，看门狗检测中断使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	ADC_WDG0_LOL_EN ADC Watchdog0 Lowlevel Enable，看门狗检测低阈值使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	ADC_WDG0_HIL_EN ADC Watchdog0 Highlevel Enable，看门狗检测高阈值使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	ADC_WDG0_EN ADC Watchdog0 Enable，ADC 看门狗 0 使能。 • 0x0：未使能 • 0x1：使能

9.4.5.35. 0x0224 ADC_WDG0_LVL

默认值：0x00000000			看门狗 0 阈值配置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	ADC_WDG0_HIL

默认值：0x00000000			看门狗 0 阈值配置寄存器
位域	类型	默认值	描述
			ADC Watchdog0 Highlevel，ADC 看门狗 0 的高阈值设置。
15:0	R/W	0x0	ADC_WDG0_LOL ADC Watchdog0 Lowlevel，ADC 看门狗 0 的低阈值设置。

9.4.5.36. 0x0230 ADC_WDG1_CFG

默认值：0x00000000			看门狗 1 配置寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:16	R/W	0x0	ADC_WDG1_SRC_SEL ADC Watchdog1 Source Select，ADC 看门狗 1 的输入源选择。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • others：Reserved
15:6	–	–	–
5	R/W	0x0	ADC_WDG1_NZD_EN ADC Watchdog1 Negative Zero Detect Enable，看门狗反向过 1 检测使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	ADC_WDG1_PZD_EN ADC Watchdog1 Positive Zero Detect Enable，看门狗正向过 1 检测使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	ADC_WDG1_IRQ_EN ADC Watchdog1 IRQ Enable，看门狗检测中断使能。 • 0x0：未使能 • 0x1：使能

默认值：0x00000000			看门狗 1 配置寄存器
位域	类型	默认值	描述
2	R/W	0x0	ADC_WDG1_LOL_EN ADC Watchdog1 Lowlevel Enable，看门狗检测低阈值使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	ADC_WDG1_HIL_EN ADC Watchdog1 Highlevel Enable，看门狗检测高阈值使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	ADC_WDG1_EN ADC Watchdog1 Enable，ADC 看门狗 1 使能。 • 0x0：未使能 • 0x1：使能

9.4.5.37. 0x0234 ADC_WDG1_LVL

默认值：0x00000000			看门狗 1 阈值配置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	ADC_WDG1_HIL ADC Watchdog1 Highlevel，ADC 看门狗 1 的高阈值设置。
15:0	R/W	0x0	ADC_WDG1_LOL ADC Watchdog1 Lowlevel，ADC 看门狗 1 的低阈值设置。

9.4.5.38. 0x0240 ADC_WDG2_CFG

默认值：0x00000000			看门狗 2 配置寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:16	R/W	0x0	ADC_WDG2_SRC_SEL ADC Watchdog2 Source Select，ADC 看门狗 2 的输入源选择。

默认值：0x00000000			看门狗 2 配置寄存器
位域	类型	默认值	描述
			• 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • others：Reserved
15:6	–	–	–
5	R/W	0x0	ADC_WDG2_NZD_EN ADC Watchdog2 Negative Zero Detect Enable，看门狗反向过 2 检测使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	ADC_WDG2_PZD_EN ADC Watchdog2 Positive Zero Detect Enable，看门狗正向过 2 检测使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	ADC_WDG2_IRQ_EN ADC Watchdog2 IRQ Enable，看门狗检测中断使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	ADC_WDG2_LOL_EN ADC Watchdog2 Lowlevel Enable，看门狗检测低阈值使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	ADC_WDG2_HIL_EN ADC Watchdog2 Highlevel Enable，看门狗检测高阈值使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	ADC_WDG2_EN ADC Watchdog2 Enable，ADC 看门狗 2 使能。 • 0x0：未使能 • 0x1：使能

9.4.5.39. 0x0244 ADC_WDG2_LVL

默认值：0x00000000			看门狗 2 阈值配置寄存器
位域	类型	默认值	描述
31:16	R/W	0x0	ADC_WDG2_HIL ADC Watchdog2 Highlevel，ADC 看门狗 2 的高阈值设置。
15:0	R/W	0x0	ADC_WDG2_LOL ADC Watchdog2 Lowlevel，ADC 看门狗 2 的低阈值设置。

9.4.5.40. 0x0248 ADC_WDG_STA

默认值：0x00000000			看门狗状态寄存器
位域	类型	默认值	描述
31:20	–	–	–
19	R/W1C	0x0	ADC_WDG2_NZD_STATE • 0x0：无负向过 0 报警 • 0x1：有负向过 0 报警
18	R/W1C	0x0	ADC_WDG2_PZD_STATE • 0x0：无正向过 0 报警 • 0x1：有正向过 0 报警
17	R/W1C	0x0	ADC_WDG2_LOL_STATE • 0x0：无低阈值报警 • 0x1：有低阈值报警
16	R/W1C	0x0	ADC_WDG2_HIL_STATE • 0x0：无高阈值报警 • 0x1：有高阈值报警
15:12	–	–	–
11	R/W1C	0x0	ADC_WDG1_NZD_STATE • 0x0：无负向过 0 报警 • 0x1：有负向过 0 报警
10	R/W1C	0x0	ADC_WDG1_PZD_STATE

默认值：0x00000000			看门狗状态寄存器
位域	类型	默认值	描述
			• 0x0：无正向过 0 报警 • 0x1：有正向过 0 报警
9	R/W1C	0x0	ADC_WDG1_LOL_STATE • 0x0：无低阈值报警 • 0x1：有低阈值报警
8	R/W1C	0x0	ADC_WDG1_HIL_STATE • 0x0：无高阈值报警 • 0x1：有高阈值报警
7:4	–	–	–
3	R/W1C	0x0	ADC_WDG0_NZD_STATE • 0x0：无负向过 0 报警 • 0x1：有负向过 0 报警
2	R/W1C	0x0	ADC_WDG0_PZD_STATE • 0x0：无正向过 0 报警 • 0x1：有正向过 0 报警
1	R/W1C	0x0	ADC_WDG0_LOL_STATE • 0x0：无低阈值报警 • 0x1：有低阈值报警
0	R/W1C	0x0	ADC_WDG0_HIL_STATE • 0x0：无高阈值报警 • 0x1：有高阈值报警

9.4.5.41. 0x0300 ADC_QN_MCR

默认值：0x00000000			常规队列控制寄存器
位域	类型	默认值	描述
31	R/WAC	0x0	QN_SS_EN 软件通过本字段触发 ADC 队列 N 进行采样。

默认值：0x00000000			常规队列控制寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能，使能队列单次采样
30:6	–	–	–
5	R/W	0x0	QN_INTE 本字段设定了队列 N 的中断使能。 • 0x0：禁止 QN_INT 中断，尽管 QN_INT 中断标志置位了，但是仍然不会产生中断 • 0x1：使能 QN_INT 中断，只要 QN_INT 中断标志被置位，就会产生中断，直至 QN_INT 标志被软件清除
4	R/W	0x0	QN_INTM 本字段设定了队列 N 的中断模式。 • 0x0：每次队列执行结束都会产生一次 QN_INT 中断 • 0x1：每隔一次（每两次）队列执行结束就会产生一次 QN_INT 中断
3	R/WAC	0x0	QN_RST 常规队列复位。 • 0x0：无使能 • 0x1：立即复位常规队列到常规队列的节点 0，下一次触发发生后，从节点 0 开始执行采样，复位后现有未执行的采样命令立刻放弃，并等待已执行的采样命令后，硬件自动清零本字段
2	R/W	0x0	QN_PS_EN Queue Normal Periodic Sample Mode Enable，常规队列周期采样模式使能。 • 0x0：未使能，关闭队列周期采样 • 0x1：使能，触发队列进行周期采样，周期采样间隔为 PSI 控制
1	R/W	0x0	QN_Override • 0x0：回环从节点 0 到节点 TRIG_CNT • 0x1：回环从节点 0 到最大节点
0	R/W	0x0	QN_EN Queue Normal Enable，常规队列使能。 • 0x0：未使能 • 0x1：使能

9.4.5.42. 0x304 ADC_QN_TCR

默认值：0x00000000			常规队列采样节点数寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:0	R/W	0x0	QN_TRIG_CNT（常规队列触发采样节点数） 本字段设定了队列 N 的每次触发信号发生后，采样队列的连续节点数，实际采样节点数为 QN_TRIG_CNT + 1。 • 0x0：每次触发执行 1 个采样节点 • 0x1：每次触发执行 2 个采样节点 • 0x2：每次触发执行 3 个采样节点 • … • 0x17：每次触发执行 24 个采样节点 • Others：每次触发执行 24 个采样节点

9.4.5.43. 0x308 ADC_PSI

默认值：0x00005DC0			常规队列采样间隔寄存器
位域	类型	默认值	描述
31:28	–	–	–
27:0	R/W	0x5DC0	PSI Periodic Sample Interval，周期性采样间隔，本字段为 0 表示连续采样，两次采样循环之间没有 Delay，仅周期采样模式（QN_PS_EN = 0x1) 下有效。 本字段定义周期性采样的采样间隔，采样间隔为 PSI × AHB_CLK。例如，假设 AHB_CLK = 24 MHz，PSI = 0x5DC0 = 24000，采样周期 TPSI = 24000/24000000 = 0.001 s = 1 ms。

9.4.5.44. 0x030C ADC_QN_MSR

默认值：0x00000000			常规队列状态寄存器
位域	类型	默认值	描述
31:26	R0	0x0	QN_CNT 本字段显示内部状态机的当前 QN 队列剩余采样节点数的计数值。
25:21	–	–	–

默认值：0x00000000			常规队列状态寄存器
位域	类型	默认值	描述
20:16	RO	0x0	QN_STATE Queue N State，队列 N 状态。 本字段显示当前队列 N 所处的采样节点位置。初始为 0，显示队列 N 处于 NODE0 的位置，随着触发采样的递进，QN_STATE 会不断往前计数。例如 QN_STATE = 5，表示下一次会对 NODE5 描述的通道号进行采样。
15:3	–	–	–
2	RO	0x0	QN_EOQ Queue N End of Queue，队列 N 结束翻转标志。 本字段仅在中断模式 1（QN_INTM=1）有效，在中断模式 0（QN_INTM=0）下，本字段固定为 0。在中断模式 1 下，每次按照 QN_TRIG_CNT 计数完成相应的采样节点数后就会自动翻转一次（由 0 翻转为 1，或者由 1 翻转为 0）。
1	RO	0x0	QN_BUSY • 0x0：队列 N 处于空闲状态，等待触发信号 • 0x1：队列 N 处于采样过程中
0	R/W1C	0x0	QN_INT Queue N Interrupt Flag，队列 N 完成事件标志。 • 0x0：无完成事件 • 0x1：有完成事件，本字段可以通过软件写 1 清零

9.4.5.45. 0x0310 ADC_QN_NOD0

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE3 本字段定义了 QN 节点 3 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE2 本字段定义了 QN 节点 2 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE1 本字段定义了 QN 节点 1 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE0 本字段定义了 QN 节点 0 的 ADC 输入通道号。

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
			• 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0 • 0x21：TS1 • 0x22：TS2 • 0x23：TS3 • 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1 • 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.46. 0x0314 ADC_QN_NOD1

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE7 本字段定义了 QN 节点 7 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE6 本字段定义了 QN 节点 6 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE5 本字段定义了 QN 节点 5 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE4 本字段定义了 QN 节点 4 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ...

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
			• 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0 • 0x21：TS1 • 0x22：TS2 • 0x23：TS3 • 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1 • 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.47. 0x0318 ADC_QN_NOD2

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE11 本字段定义了 QN 节点 11 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE10 本字段定义了 QN 节点 10 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE9 本字段定义了 QN 节点 9 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE8 本字段定义了 QN 节点 8 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
			• 0x21：TS1 • 0x22：TS2 • 0x23：TS3 • 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1 • 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.48. 0x031C ADC_QN_NOD3

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE15 本字段定义了 QN 节点 15 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE14 本字段定义了 QN 节点 14 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE13 本字段定义了 QN 节点 13 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE12 本字段定义了 QN 节点 12 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0 • 0x21：TS1 • 0x22：TS2 • 0x23：TS3

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
			• 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1 • 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.49. 0x0320 ADC_QN_NOD4

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE19 本字段定义了 QN 节点 19 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE18 本字段定义了 QN 节点 18 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE17 本字段定义了 QN 节点 17 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE16 本字段定义了 QN 节点 16 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0 • 0x21：TS1 • 0x22：TS2 • 0x23：TS3 • 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
			• 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.50. 0x0324 ADC_QN_NOD5

默认值：0x00000000			采样队列 N 节点寄存器
位域	类型	默认值	描述
31:24	R/W	0x0	QN_NODE23 本字段定义了 QN 节点 23 的 ADC 输入通道号。
23:16	R/W	0x0	QN_NODE22 本字段定义了 QN 节点 22 的 ADC 输入通道号。
15:8	R/W	0x0	QN_NODE21 本字段定义了 QN 节点 21 的 ADC 输入通道号。
7:0	R/W	0x0	QN_NODE20 本字段定义了 QN 节点 20 的 ADC 输入通道号。 • 0x0：选择 ADC 输入通道 0 • 0x1：选择 ADC 输入通道 1 • ... • 0x17：选择 ADC 输入通道 23 • 0x18~0x1F：无效 • 0x20：TS0 • 0x21：TS1 • 0x22：TS2 • 0x23：TS3 • 0x24~0x2F：无效 • 0x30：VTS0 • 0x31：VTS1 • 0x32：VTS2 • 0x33：VTS3 • others：无效

9.4.5.51. 0x0328 ADC_QN_SMP_SET0

默认值：0x00000000			采样队列 N 采样时间寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	QN_NODE11_SMP_SET QN 节点 11 的采样时间配置。
21:20	R/W	0x0	QN_NODE10_SMP_SET QN 节点 10 的采样时间配置。
19:18	R/W	0x0	QN_NODE9_SMP_SET QN 节点 9 的采样时间配置。
17:16	R/W	0x0	QN_NODE8_SMP_SET QN 节点 8 的采样时间配置。
15:14	R/W	0x0	QN_NODE7_SMP_SET QN 节点 7 的采样时间配置。
13:12	R/W	0x0	QN_NODE6_SMP_SET QN 节点 6 的采样时间配置。
11:10	R/W	0x0	QN_NODE5_SMP_SET QN 节点 5 的采样时间配置。
9:8	R/W	0x0	QN_NODE4_SMP_SET QN 节点 4 的采样时间配置。
7:6	R/W	0x0	QN_NODE3_SMP_SET QN 节点 3 的采样时间配置。
5:4	R/W	0x0	QN_NODE2_SMP_SET QN 节点 2 的采样时间配置。
3:2	R/W	0x0	QN_NODE1_SMP_SET QN 节点 1 的采样时间配置。
1:0	R/W	0x0	QN_NODE0_SMP_SET QN 节点 0 的采样时间配置。 • 0x0：采样时间 1，根据寄存器 ADC_ACQ 去配置采样时间 • 0x1：采样时间 2，根据寄存器 ADC_ACQ_1 去配置采样时间

默认值：0x00000000			采样队列 N 采样时间寄存器
位域	类型	默认值	描述
			• 0x2：采样时间 3，根据寄存器 ADC_ACQ_2 去配置采样时间 • 0x3：采样时间 4，根据寄存器 ADC_ACQ_3 去配置采样时间

9.4.5.52. 0x032C ADC_QN_SMP_SET1

默认值：0x00000000			采样队列 N 采样时间寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:22	R/W	0x0	QN_NODE23_SMP_SET QN 节点 23 的采样时间配置。
21:20	R/W	0x0	QN_NODE22_SMP_SET QN 节点 22 的采样时间配置。
19:18	R/W	0x0	QN_NODE21_SMP_SET QN 节点 21 的采样时间配置。
17:16	R/W	0x0	QN_NODE20_SMP_SET QN 节点 20 的采样时间配置。
15:14	R/W	0x0	QN_NODE19_SMP_SET QN 节点 19 的采样时间配置。
13:12	R/W	0x0	QN_NODE18_SMP_SET QN 节点 18 的采样时间配置。
11:10	R/W	0x0	QN_NODE17_SMP_SET QN 节点 17 的采样时间配置。
9:8	R/W	0x0	QN_NODE16_SMP_SET QN 节点 16 的采样时间配置。
7:6	R/W	0x0	QN_NODE15_SMP_SET QN 节点 15 的采样时间配置。
5:4	R/W	0x0	QN_NODE14_SMP_SET QN 节点 14 的采样时间配置。
3:2	R/W	0x0	QN_NODE13_SMP_SET

默认值：0x00000000			采样队列 N 采样时间寄存器
位域	类型	默认值	描述
			QN 节点 13 的采样时间配置。
1:0	R/W	0x0	QN_NODE12_SMP_SET QN 节点 0 的采样时间配置。 • 0x0：采样时间 1，根据寄存器 ADC_ACQ 去配置采样时间 • 0x1：采样时间 2，根据寄存器 ADC_ACQ_1 去配置采样时间 • 0x2：采样时间 3，根据寄存器 ADC_ACQ_2 去配置采样时间 • 0x3：采样时间 4，根据寄存器 ADC_ACQ_3 去配置采样时间

9.4.5.53. 0x0330 ADC_QNFCR

默认值：0x00000000			ADC 队列 N FIFO 控制寄存器
位域	类型	默认值	描述
31:30	–	–	–
29:24	RO	0x0	QN_FIFO_DCNT Queue N FIFO Data Count，队列 N 的 FIFO 有效数据量。QN FIFO 的总深度为 24，因此 0x0~0x18 为有效。
23:22	–	–	–
21	RO	0x0	QN_FERR 常规队列的 FIFO 异常中断标志，FIFO 上/下溢出状态有效（FIFO_OFS ‖ FIFO_UFS），硬件会自动置为本字段。清零本字段的方法：清零 FIFO 上/下溢出状态（FIFO_OFS ‖ FIFO_UFS=0）。
20	RO	0x0	QN_FRDY 常规队列的 FIFO 数据就绪中断标志，FIFO 数据就绪，硬件就会置位本字段。当 FIFO 有效数据量低于 FIFO 数据就绪阈值（FIFO_DCNT<FIFO_DRTH）硬件自动清零。
19:18	–	–	–
17	R/W1C	0x0	QN_FIFO_UFS Queue N FIFO Underflow Status，队列 N FIFO 下溢出状态，高电平有效。 本字段由硬件置位，并由软件写 1 清零。
16	R/W1C	0x0	QN_FIFO_OFS Queue N FIFO Overflow Status，队列 N FIFO 上溢出状态，高电平有效。

默认值：0x00000000			ADC 队列 N FIFO 控制寄存器
位域	类型	默认值	描述
			本字段由硬件置位，并由软件写 1 清零。
15:13	–	–	–
12:8	R/W	0x0	QN_FIFO_DRTH Queue N FIFO Data Ready Threshold，队列 N FIFO 数据就绪阈值。当 FIFO 的数据量达到了本阈值后，就有会产生数据就绪中断。 0x1~0x18 表示有效阈值范围。其它为非法配置，会导致无中断。
7:4	–	–	–
3	R/W	0x0	QN_FIFO_ERRIE Queue N FIFO Error Interrupt Enable，队列 N 的 FIFO 异常（上/下溢出）中断使能。 • 0x0：禁止产生中断 • 0x1：允许产生中断。
2	–	–	–
1	R/W	0x0	QN_FIFO_RDYIE Queue N FIFO Ready Interrupt Enabel，队列 N 的 FIFO 数据就绪中断使能。 • 0x0：禁止产生中断。 • 0x1：允许产生中断。
0	R/WAC	0x0	QN_FIFO_FLUSH Queue N FIFO Flush，队列 N FIFO 清空操作。 本字段由软件置位硬件清零。软件置位启动 FIFO 清空操作，FIFO 清空操作完成后，由硬件清零。

9.4.5.54. 0x0334 ADC_QNFDR

默认值：0x00000000			常规队列 FIFO 数据寄存器
位域	类型	默认值	描述
31:22	–	–	–
21:16	RO	0x0	CHNUM DATA 对应的通道号，与 QN_NODEx 中的数值对应。
15:0	RO	0x0	DATA

默认值：0x00000000			常规队列 FIFO 数据寄存器
位域	类型	默认值	描述
			本字段为 ADC 采样后转化为数字量的数据。

9.4.5.55. 0x0380 ADC_CAPCALCSR

默认值：0x08000000			常规队列 FIFO 数据寄存器
位域	类型	默认值	描述
31	R/WAC	0x0	CAP_CALVAL_UPD Capacitor Calibration Value Update，ADC 电容校准值更新。通过软件设置 CAP_OFFSET_CALVAL与DCAL_CFG ，置位本字段，表示软件需要更新 CAP_OFFSET_CALVAL与DCAL_CFG 值，本字段置位后会触发内部更新的动作，硬件更新完毕后，会自动清零。
30:28	–	–	–
27:16	R/W	0x200	CAP_OFFSET_CALVAL Capacitor OFFSET Calibration Value，ADC电容偏移失配校准值。
15:6	–	–	–
5:4	R/W	0x0	CAP_CAL_MODE 0：电容失配校准方式1 1：电容失配校准方式2 2：电容失配校准方式3（推荐方式3）
3:1			
0	R/WAC	0x0	CAP_CALEN Capacitor Calibration Enable，ADC电容失配校准使能，由软件置位，使能 ADC 电容失配校准后，校准完成之后，对此控制信号自动清零。软件查询到此信号被清零后，意味着校准已完成。

9.4.5.56. 0x0384 ADC_DCALC10

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG1

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
			DATA Calibrate Config 1，数据校准配置 1。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG0 DATA Calibrate Config 0，数据校准配置 0。

9.4.5.57. 0x0388 ADC_DCALC32

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG3 DATA Calibrate Config 3，数据校准配置 3。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG2 DATA Calibrate Config 2，数据校准配置 2。

9.4.5.58. 0x038C ADC_DCALC54

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG5 DATA Calibrate Config 5，数据校准配置 5。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG4 DATA Calibrate Config 4，数据校准配置 4。

9.4.5.59. 0x0390 ADC_DCALC76

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG7 DATA Calibrate Config 7，数据校准配置 7。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG6 DATA Calibrate Config 6，数据校准配置 6。

9.4.5.60. 0x0394 ADC_DCALC98

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG9 DATA Calibrate Config 9，数据校准配置 9。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG8 DATA Calibrate Config 8，数据校准配置 8。

9.4.5.61. 0x0398 ADC_DCALC1110

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG11 DATA Calibrate Config 11，数据校准配置 11。
15:11	–	–	–

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
10:0	R/W	0x0	DCAL_CFG10 DATA Calibrate Config 10，数据校准配置 10。

9.4.5.62. 0x039C ADC_DCALC1312

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG13 DATA Calibrate Config 13，数据校准配置 13。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG12 DATA Calibrate Config 12，数据校准配置 12。

9.4.5.63. 0x03A0 ADC_DCALC1514

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG15 DATA Calibrate Config 15，数据校准配置 15。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG14 DATA Calibrate Config 14，数据校准配置 14。

9.4.5.64. 0x03A4 ADC_DCALC1716

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:27	–	–	–
26:16	R/W	0x0	DCAL_CFG17 DATA Calibrate Config 17，数据校准配置 17。
15:11	–	–	–
10:0	R/W	0x0	DCAL_CFG16 DATA Calibrate Config 16，数据校准配置 16。

9.4.5.65. 0x03A8 ADC_DCALCx18

默认值：0x00000000			数据校准配置寄存器
位域	类型	默认值	描述
31:11	–	–	–
10:0	R/W	0x0	DCAL_CFG18 DATA Calibrate Config 18，数据校准配置 18。

9.4.5.66. 0xFFC VERSION

默认值：0x00000100			ADC 模块版本 (ADC Version)
位域	类型	默认值	描述
31:0	RO	0x00000100	Version（模块版本号） 采用 BCD 码显示，V1.0。

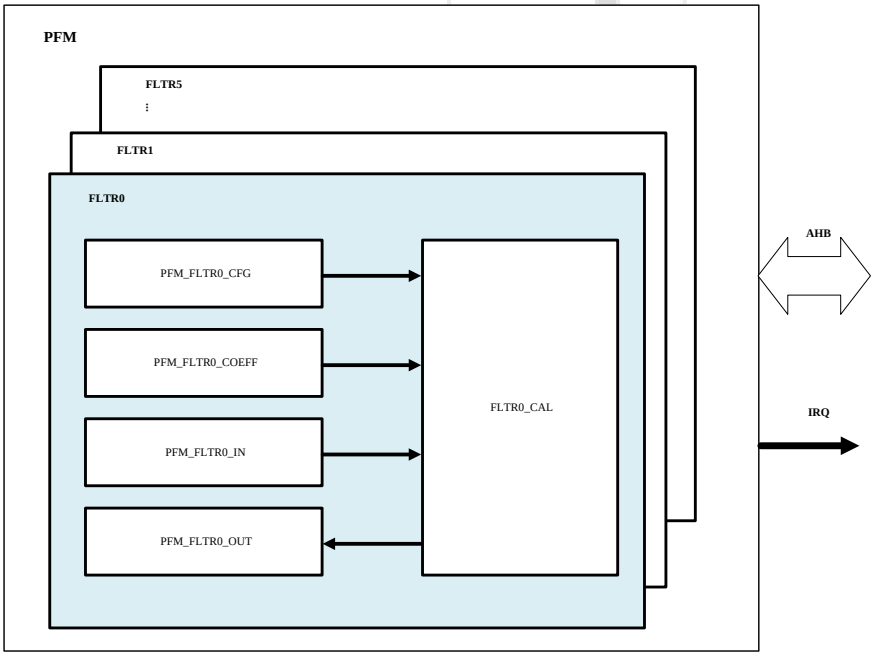
9.5. Programable Filter Module (PFM)

Programable Filter Module (PFM) 是一种可编程滤波单元，通常用于信号处理应用中，以对输入信号进行滤波和处理，并根据需要选择不同的滤波类型来去除噪音或增强特定频率的信号。

9.5.1. 特性说明

- 最高可支持 32–bit 的 Q31 定点数的计算输入和计算输出
- 最多支持 6 x FIR/ IIR 滤波器（FLTR0/ 1/ 2/ 3/ 4/ 5）同时工作
- 每个滤波器均可以独立配置系数，最高可以实现 4 阶滤波器
- 每个滤波器内置 64 x 32 乘法器、96 位加法器

9.5.2. 原理框图



9.5.3. 功能描述

滤波器的实现和计算过程涉及以下步骤：

1. 设置滤波器参数：
   - 阶数：确定滤波器的阶数，即滤波器的复杂程度。
   - 配置系数 A 和 系数 B：决定滤波器的频率响应特性，可根据滤波器设计方法计算并设置滤波器的系数。系数将分别存储在两组寄存器中，详情可查看系数寄存器配置。
   - 配置控制寄存器 FLTR_COEFF_SEL 来选择当前使用的系数组。
2. 数据处理（可选）。
   - 缩放处理（SCALA 和 SCALB 模块）：如果系数较大或较小，可以使用缩放模块调整系数，以避免数值溢出或精度损失。
   - 限值处理（LIM 模块）：在部分应用中，可能需要对输出数据进行限值处理，以确保数据在预期范围内。
3. 系数切换（可选）。

- 如果需要切换系数组，首先等待滤波器完成当前计算（检查 FLTR_DNE 标志位）。
- 更新控制寄存器 FLTR_COEFF_SEL 以选择新的系数组。

9.5.3.1. 滤波阶数配置

滤波阶数配置是理解滤波器工作原理的关键。在 图 9–35 : FLTR0/ 1/ 2/ 3/ 4/ 5 的滤波器结构中：

- x[n] 为输入数据
- y[n] 为输出数据
- A[1]~A[4] 为系数 A，即 A[k]
- B[0]~B[4] 为系数 B，即 B[k]
- SCALA 和 SCALB 模块均为缩小模块，支持对数据进行最多 2^{31} 倍的缩小
- LIM 模块为限值模块，支持设定最大 64 bits 的限值

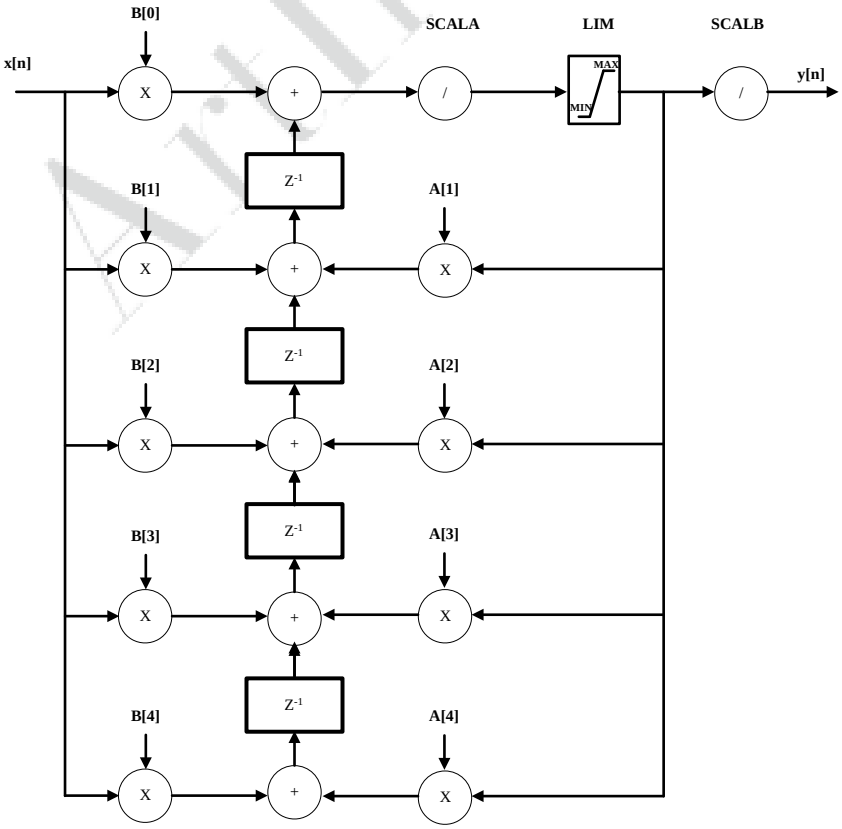


图 9–35 FLTR0/ 1/ 2/ 3/ 4/ 5 的滤波器结构

滤波器的阶数 N 可通过寄存器 FLTR_ODR 配置，最高可以支持四阶滤波器 (N = 1、2、3、4)。

9.5.3.2. 系数寄存器配置

在需要动态调整滤波参数的应用场景中，可以使用滤波器的系数切换功能。在本芯片设计中，每个滤波器都有独立的两套系数寄存器，通过寄存器 FLTR_COEFF_SEL 控制系数的切换。

- 第一组系数寄存器为 FLTRn_FGP_A1\A2\A3\A4 和 FLTRn_FGP_ B0\B1\B2\B3\B4，用于存储第一组滤波器系数。
- 第二组系数寄存器为 FLTRn_SGP_A1\A2\A3\A4 和 FLTRn_SGP_ B0\B1\B2\B3\B4，用于存储第二组滤波器系数。

寄存器 FLTR_COEFF_SEL 为影子更新，即只有在每次滤波器完成计算（FLTR_DNE 标志位被置位）后才更新切换控制值。

9.5.3.3. 输出数据 (y[n]) 计算

根据当前选择的系数组和滤波器的阶数，使用下列滤波器公式计算输出数据：

$$y[n]=\sum_{k=0}^{N}B[k]*x[n-k]+\sum_{k=1}^{N}A[k]*y[n-k]$$

9.5.3.4. 滤波器计算流程

如图 9–36 : 滤波器计算流程所示，为了保证每次获取滤波器结果的速度，滤波器的计算可以分成下列部分：

> **注：**
>
> 在触发滤波器计算后，滤波器计算结果的读取支持零等待模式，无需查询状态而是直接读取计算结果。若使能了串联结构，则不支持零等待。

- BlockA 部分：利用当前输入的 x[n] 以及上次计算积累的 last[n] 计算当前滤波的结果 y[n]。

  在触发滤波器的进行计算后，BlockA 部分的计算属于第一阶段 (Phase1)。计算完成后，滤波器的结果会被更新到结果寄存器中，软件可通过读取结果寄存器获得结果。

- BlockB 部分：计算积累部分，用于下一次的滤波计算使用。

  在触发滤波器的进行计算后，BlockB 部分的计算属于第二阶段 (Phase2)。计算完成后，视为完成此次滤波器计算，并产生 FLTR _DNE 信号。

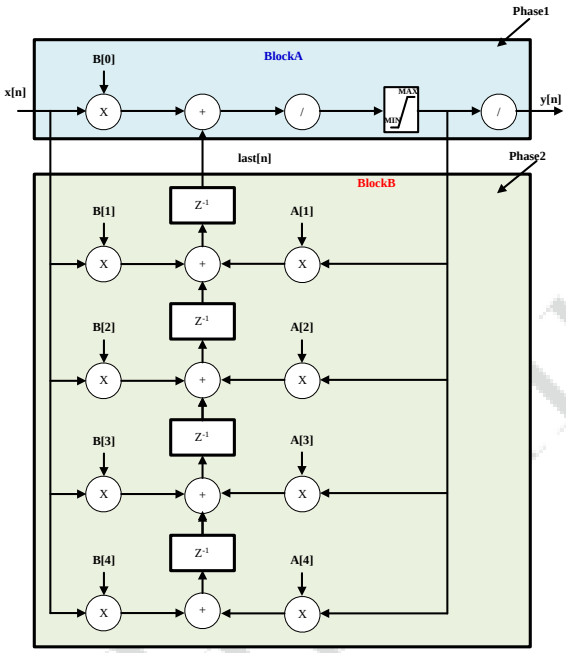


图 9–36 滤波器计算流程

9.5.3.5. 多滤波器级联设置

通过将多个滤波器串联起来，可以增强滤波效果或实现复杂的滤波功能。以下是关于如何设置多滤波器级联的详细步骤和注意事项。

1. 确定需要级联的滤波器数量及其顺序。
2. 配置每个滤波器的触发信号和数据源。

   如图所示：

   - 通过寄存器选择每个滤波计算的 START 触发信号，默认选择各自滤波器的触发信号。
   - 通过寄存器 FLTR_IN_SRC_SEL 选择输入的数据源，默认选择各自滤波器的输入数据寄存器。

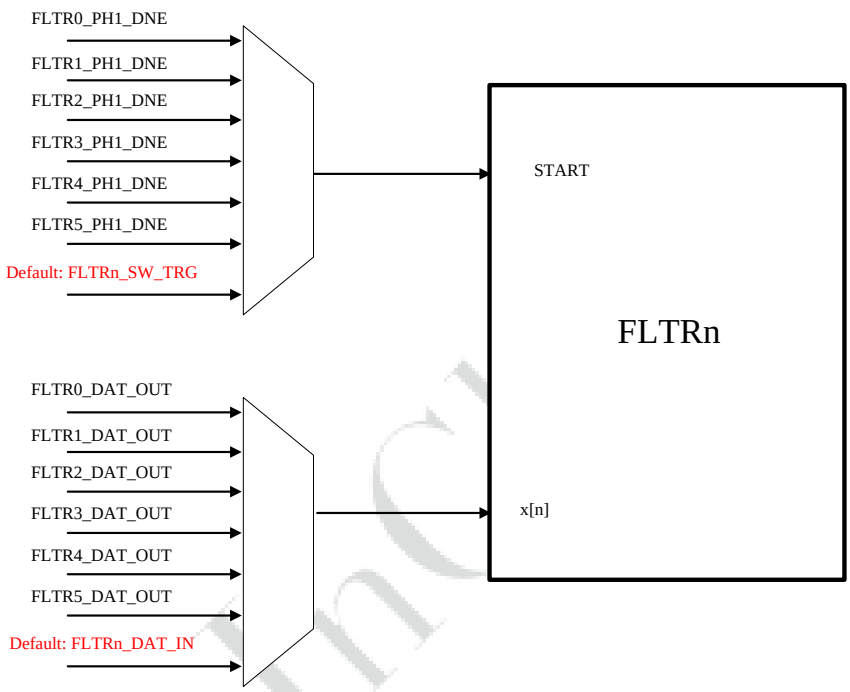
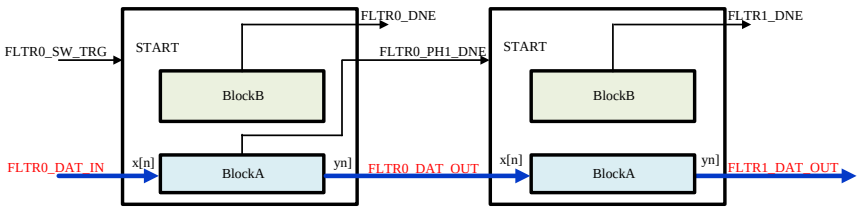


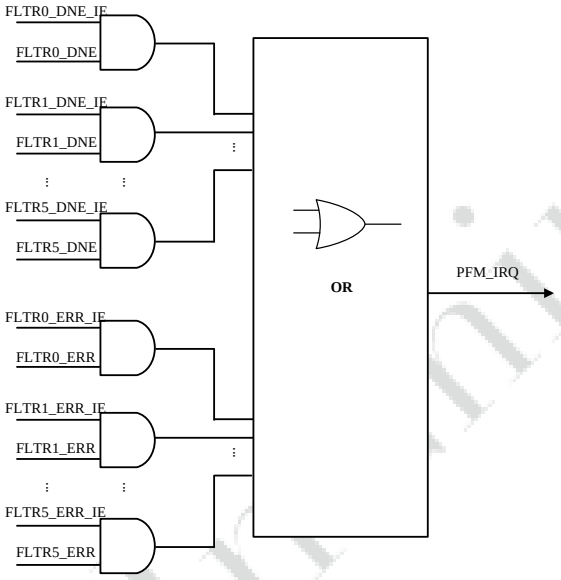
表 9–27 多滤波器级联配置示例

滤波器	触发信号 START	数据源	实现效果
FLTR0	选择默认值，即使用 FLTR0 自身的触发信号。	选择默认值，即 FLTR0 的输入数据寄存器 FLTR0_DAT_IN。	• FLTR0_DAT_IN 的输入为级联滤波器 FLTR0 + FLTR1 的数据输入。 • FLTR1_DAT_OUT 的输出为级联滤波器 FLTR0 + FLTR1 的数据输出。 • 软件在判断 FLTR0_DNE 和 FLTR1_DNE 的信号都有效后，才视为级联滤波器 FLTR0 + FLTR1 的本次计算结果结束，且可以进行下一次的计算触发。
FLTR1	选择 FLTR0 的第一阶段完成信号 FLTR0_PH1_DNE。	选择 FLTR0 的输出数据寄存器 FLTR0_DAT_OUT。	

以下是上述配置达到的效果图示：



9.5.3.6. 中断



PFM 模块支持产生中断信号，如图所示，根据用户的需要，通过寄存器使能相应的信号源作为中断信号即可。

其中，中断信号源说明如下：

- FLTRn_DNE（n=0/1/2/.../5)：指示滤波器完成计算的标记信号
- FLTRn_ERR（n=0/1/2/.../5)：指示滤波器发生错误的标记信号

9.5.4. 寄存器列表

PFM 模块最多支持六套 PFM 同时工作，即 PFM0/ 1/ 2/ 3/ 4/ 5，对应的寄存器地址映射关系如下。PFM 寄存器部分为对 PFM0/ 1/ 2/ 3/ 4/ 5 的全局控制。

表 9–28 地址映射

名称	基地址	地址空间
PFM	PFM_BASE	256 B
FLTR0	FLTR0(PFM_BASE+0x100)	256 B
FLTR1	FLTR1(PFM_BASE+0x200)	256 B
FLTR2	FLTR2(PFM_BASE+0x300)	256 B
FLTR3	FLTR3(PFM_BASE+0x400)	256 B
FLTR4	FLTR4(PFM_BASE+0x500)	256 B

表 9–28 地址映射 (续)

名称	基地址	地址空间
FLTR5	FLTR5(PFM_BASE+0x600)	256 B

表 9–29 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	PFM_STS	–	0x0000 PFM_STS
0x0004	PFM_IE	–	0x0004 PFM_IE
FLTRn+0x000	FLTRn_CTL	–	FLTRn+0x000 FLTRn_CTL
FLTRn+0x004	FLTRn_TRG	–	FLTRn+0x004 FLTRn_TRG
FLTRn+0x008	FLTRn_SCAL	–	FLTRn+0x008 FLTRn_SCAL
FLTRn+0x00C	FLTRn_DAT_IN	–	FLTRn+0x00C FLTRn_DAT_IN
FLTRn+0x010	FLTRn_DAT_OUT	–	FLTRn+0x010 FLTRn_DAT_OUT
FLTRn+0x014	FLTRn_OUT_MIN_H32	–	FLTRn+0x014 FLTRn_OUT_MIN_H32
FLTRn+0x018	FLTRn_OUT_MIN_L32	–	FLTRn+0x018 FLTRn_OUT_MIN_L32
FLTRn+0x01C	FLTRn_OUT_MAX_H32	–	FLTRn+0x01C FLTRn_OUT_MAX_H32
FLTRn+0x020	FLTRn_OUT_MAX_L32	–	FLTRn+0x020 FLTRn_OUT_MAX_L32
FLTRn+0x050	FLTRn_FGP_A1	–	FLTRn+0x050 FLTRn_FGP_A1
FLTRn+0x054	FLTRn_FGP_A2	–	FLTRn+0x054 FLTRn_FGP_A2
FLTRn+0x058	FLTRn_FGP_A3	–	FLTRn+0x058 FLTRn_FGP_A3
FLTRn+0x05C	FLTRn_FGP_A4	–	FLTRn+0x05C FLTRn_FGP_A4
FLTRn+0x060	FLTRn_FGP_B0	–	FLTRn+0x060 FLTRn_FGP_B0
FLTRn+0x064	FLTRn_FGP_B1	–	FLTRn+0x064 FLTRn_FGP_B1
FLTRn+0x068	FLTRn_FGP_B2	–	FLTRn+0x068 FLTRn_FGP_B2
FLTRn+0x06C	FLTRn_FGP_B3	–	FLTRn+0x06C FLTRn_FGP_B3

表 9–29 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
FLTRn+0x070	FLTRn_FGP_B4	–	FLTRn+0x070 FLTRn_FGP_B4
FLTRn+0x074	FLTRn_SGP_A1	–	FLTRn+0x074 FLTRn_SGP_A1
FLTRn+0x078	FLTRn_SGP_A2	–	FLTRn+0x078 FLTRn_SGP_A2
FLTRn+0x07C	FLTRn_SGP_A3	–	FLTRn+0x07C FLTRn_SGP_A3
FLTRn+0x080	FLTRn_SGP_A4	–	FLTRn+0x080 FLTRn_SGP_A4
FLTRn+0x084	FLTRn_SGP_B0	–	FLTRn+0x084 FLTRn_SGP_B0
FLTRn+0x088	FLTRn_SGP_B1	–	FLTRn+0x088 FLTRn_SGP_B1
FLTRn+0x08C	FLTRn_SGP_B2	–	FLTRn+0x08C FLTRn_SGP_B2
FLTRn+0x090	FLTRn_SGP_B3	–	FLTRn+0x090 FLTRn_SGP_B3
FLTRn+0x094	FLTRn_SGP_B4	–	FLTRn+0x094 FLTRn_SGP_B4
0x0FC	PFM_VER	–	0x0FC PFM_VER

9.5.5. 寄存器描述

9.5.5.1. 0x0000 PFM_STS

默认值：0x00000000			PFM 状态寄存器
位域	类型	默认值	描述
31:30	–	–	–
29	R	0x0	FLTR5_STS Filter5 Status，Filter5 的工作状态。 • 0x0：空闲状态 • 0x1：繁忙状态
28	R	0x0	FLTR4_STS Filter4 Status，Filter4 的工作状态。

默认值：0x00000000			PFM 状态寄存器
位域	类型	默认值	描述
			• 0x0：空闲状态 • 0x1：繁忙状态
27	R	0x0	FLTR3_STS Filter3 Status，Filter3 的工作状态。 • 0x0：空闲状态 • 0x1：繁忙状态
26	R	0x0	FLTR2_STS Filter2 Status，Filter2 的工作状态。 • 0x0：空闲状态 • 0x1：繁忙状态
25	R	0x0	FLTR1_STS Filter1 Status，Filter1 的工作状态。 • 0x0：空闲状态 • 0x1：繁忙状态
24	R	0x0	FLTR0_STS Filter0 Status，Filter0 的工作状态。 • 0x0：空闲状态 • 0x1：繁忙状态
23:22	–	–	–
21	R/WAC	0x0	FLTR5_ERR Filter5 Error Status，Filter5 报错状态，此状态置 1 表示 Filter5 还在忙碌过程中触发了新的计算或者 Filter5 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
20	R/WAC	0x0	FLTR4_ERR Filter4 Error Status，Filter4 报错状态，此状态置 1 表示 Filter4 还在忙碌过程中触发了新的计算或者 Filter4 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
19	R/WAC	0x0	FLTR3_ERR

默认值：0x00000000			PFM 状态寄存器
位域	类型	默认值	描述
			Filter3 Error Status，Filter3 报错状态，此状态置 1 表示 Filter3 还在忙碌过程中触发了新的计算或者 Filter3 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
18	R/WAC	0x0	FLTR2_ERR Filter2 Error Status，Filter2 报错状态，此状态置 1 表示 Filter2 还在忙碌过程中触发了新的计算或者 Filter2 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
17	R/WAC	0x0	FLTR1_ERR Filter1 Error Status，Filter1 报错状态，此状态置 1 表示 Filter1 还在忙碌过程中触发了新的计算或者 Filter1 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
16	R/WAC	0x0	FLTR0_ERR Filter0 Error Status，Filter0 报错状态，此状态置 1 表示 Filter0 还在忙碌过程中触发了新的计算或者 Filter0 在未使能的情况下触发计算。 • 0x0：无状态 • 0x1：产生有效状态
15:6	–	–	–
5	R/WAC	0x0	FLTR5_DNE Filter5 Done Status，Filter5 完成状态，此状态置 1 表示 Filter5 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态
4	R/WAC	0x0	FLTR4_DNE Filter4 Done Status，Filter4 完成状态，此状态置 1 表示 Filter4 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态
3	R/WAC	0x0	FLTR3_DNE

默认值：0x00000000			PFM 状态寄存器
位域	类型	默认值	描述
			Filter3 Done Status，Filter5 完成状态，此状态置 1 表示 Filter3 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态
2	R/WAC	0x0	FLTR2_DNE Filter2 Done Status，Filter2 完成状态，此状态置 1 表示 Filter2 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态
1	R/WAC	0x0	FLTR1_DNE Filter1 Done Status，Filter1 完成状态，此状态置 1 表示 Filter1 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态
0	R/WAC	0x0	FLTR0_DNE Filter0 Done Status，Filter0 完成状态，此状态置 1 表示 Filter0 完成一次的算法计算。 • 0x0：无状态 • 0x1：产生有效状态

9.5.5.2. 0x0004 PFM_IE

默认值：0x00000000			PFM 中断使能寄存器
位域	类型	默认值	描述
31:22	–	–	–
21	R/W	0x0	FLTR5_ERR_IE Filter5 Error IRQ Enable，Filter5 错误中断使能。 • 0x0：未使能 • 0x1：使能
20	R/W	0x0	FLTR4_ERR_IE Filter4 Error IRQ Enable，Filter4 错误中断使能。

默认值：0x00000000			PFM 中断使能寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
19	R/W	0x0	FLTR3_ERR_IE Filter3 Error IRQ Enable，Filter3 错误中断使能。 • 0x0：未使能 • 0x1：使能
18	R/W	0x0	FLTR2_ERR_IE Filter2 Error IRQ Enable，Filter2 错误中断使能。 • 0x0：未使能 • 0x1：使能
17	R/W	0x0	FLTR1_ERR_IE Filter1 Error IRQ Enable，Filter1 错误中断使能。 • 0x0：未使能 • 0x1：使能
16	R/W	0x0	FLTR0_ERR_IE Filter0 Error IRQ Enable，Filter0 错误中断使能。 • 0x0：未使能 • 0x1：使能
15:6	–	–	–
5	R/W	0x0	FLTR5_DNE_IE Filter5 Done IRQ Enable，Filter5 完成中断使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	FLTR4_DNE_IE Filter4 Done IRQ Enable，Filter4 完成中断使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	FLTR3_DNE_IE Filter3 Done IRQ Enable，Filter3 完成中断使能。

默认值：0x00000000			PFM 中断使能寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
2	R/W	0x0	FLTR2_DNE_IE Filter2 Done IRQ Enable，Filter2 完成中断使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	FLTR1_DNE_IE Filter1 Done IRQ Enable，Filter1 完成中断使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	FLTR0_DNE_IE Filter0 Done IRQ Enable，Filter0 完成中断使能。 • 0x0：未使能 • 0x1：使能

9.5.5.3. FLTRn+0x000 FLTRn_CTL

默认值：0x00000000			FLTRn 控制寄存器
位域	类型	默认值	描述
31:28	R/W	0x0	FLTR_YIN_SRC_SEL Filter Y Input Souce Select，滤波器 Y 输入信号源选择。 • 0x0：选择本滤波器的输出 • 0x1：选择 FLTR0 输出 • 0x2：选择 FLTR1 输出 • 0x3：选择 FLTR2 输出 • 0x4：选择 FLTR3 输出 • 0x5：选择 FLTR4 输出 • 0x6：选择 FLTR5 输出
27:24	R/W	0x0	FLTR_ZIN_SRC_SEL Filter Z Input Souce Select，滤波器 Z 输入信号源选择。

默认值：0x00000000			FLTRn 控制寄存器
位域	类型	默认值	描述
			• 0x0：滤波器独立或串行 • 0x1：选择 FLTR0 输出 • 0x2：选择 FLTR1 输出 • 0x3：选择 FLTR2 输出 • 0x4：选择 FLTR3 输出 • 0x5：选择 FLTR4 输出 • 0x6：选择 FLTR5 输出
23:20	R/W	0x0	FLTR_START_TRG_SEL Filter Start Trigger Select，滤波器启动计算触发信号源选择。 • 0x0：选择 FLTR_SW_TRG 作为触发源 • 0x1：选择 FLTR0 _PH1_DNE • 0x2：选择 FLTR1 _PH1_DNE • 0x3：选择 FLTR2 _PH1_DNE • 0x4：选择 FLTR3 _PH1_DNE • 0x5：选择 FLTR4 _PH1_DNE • 0x6：选择 FLTR5 _PH1_DNE • others：Reserved
19:16	R/W	0x0	FLTR_IN_SRC_SEL Filter Data Input Souce Select，滤波器输入信号源选择。 • 0x0：选择寄存器 FLTRn_DAT_IN • 0x1：选择 xin[0] • 0x2：选择 xin[1] • 0x3：选择 xin[2] • 0x4：选择 xin[3] • 0x5：选择 xin[4] • 0x6：选择 xin[5] • 0x7：Reserved • 0x8：选择 FLTR0_DAT_OUT • 0x9：选择 FLTR1_DAT_OUT • 0xA：选择 FLTR2_DAT_OUT • 0xB：选择 FLTR3_DAT_OUT • 0xC：选择 FLTR4_DAT_OUT

默认值：0x00000000			FLTRn 控制寄存器
位域	类型	默认值	描述
			• 0xD：选择 FLTR5_DAT_OUT • others：Reserved
15:10	–	–	–
9	R/W	0x0	FLTR_BP_Z0_EN Filter Bypass Z0 Enable，旁通 Z0 级的运算 ，主要适用与并行运算模式下。 • 0x0：未使能 • 0x1：使能
8	–	–	–
7	R/W	0x0	FLTR_SCALB_EN Filter Scale B Enable，滤波器缩小模块 B 使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	FLTR_SCALA_EN Filter Scale A Enable，滤波器缩小模块 A 使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	FLTR_LIMIT_EN Filter Limit Enable，滤波器限幅使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	FLTR_COEFF_SEL Filter Coeff Select，滤波器系数选择，此控制位为影子更新，在每次计算完成后进行更新。 • 0x0：选择第一组系数 FGP • 0x1：选择第二组系数 SGP
3:1	R/W	0x0	FLTR_ODR Filter Order Set，滤波器阶数设置 • 0x0：1 阶 • 0x1：2 阶 • 0x2：3 阶

默认值：0x00000000			FLTRn 控制寄存器
位域	类型	默认值	描述
			• 0x3：4 阶 • others：无效
0	R/W	0x0	FLTR_EN Filter Enable，滤波器使能 • 0x0：未使能 • 0x1：使能

9.5.5.4. FLTRn+0x004 FLTRn_TRG

默认值：0x00000000			FLTRn 启动寄存器
位域	类型	默认值	描述
31:16	WAC	0x0	FLTR_RESET 此域写 A1C8，进行滤波器的复位，完成复位后，此域自动清 0 。
15:1	–	–	–
0	W1AC	0x0	FLTR_SW_TRG Filter Software Trigger，滤波器软件触发使能，此位写 1 后自动清 0 . • 0x0：未使能 • 0x1：使能，写 1 自动清 0

9.5.5.5. FLTRn+0x008 FLTRn_SCAL

默认值：0x00000000			FLTRn 输出缩写寄存器
位域	类型	默认值	描述
31:14	–	–	–
13:8	R/W	0x0	FLTR_SCALB_SET Filter Scale B Set，滤波器缩小模块 B 设置 。 • 0x0：不缩小 • 0x1：2^1 缩小 • 0x2：2^2 缩小 • ...

默认值：0x00000000			FLTRn 输出缩写寄存器
位域	类型	默认值	描述
			• 0x1F：2^31 缩小 • ... • 0x3F：2^63 缩小
7:6	–	–	–
5:0	R/W	0x0	FLTR_SCALA_SET Filter Scale A Set，滤波器缩小模块 A 设置 。 • 0x0：不缩小 • 0x1：2^1 缩小 • 0x2：2^2 缩小 • ... • 0x1F：2^31 缩小 • ... • 0x3F：2^63 缩小

9.5.5.6. FLTRn+0x00C FLTRn_DAT_IN

默认值：0x00000000			FLTRn 输入数据寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_DAT_IN FLTR Data Input，滤波器数据输入寄存器，这里最高可以支持 Q31 的定点数输入 。

9.5.5.7. FLTRn+0x010 FLTRn_DAT_OUT

默认值：0x00000000			FLTRn 输出数据寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_DAT_OUT FLTR Data Output，滤波器数据输出寄存器，最高支持 Q31 的定点数输出 。

9.5.5.8. FLTRn+0x014 FLTRn_OUT_MIN_H32

默认值：0x00000000			FLTRn 输出 MIN 值高 32 位寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_OUT_MIN_H32 FLTR Minimum High 32bits，滤波器最小限制值的高 32 位 。此字段仅在 FLTR _LIMIT_EN = 0x1 时有效 。

9.5.5.9. FLTRn+0x018 FLTRn_OUT_MIN_L32

默认值：0x00000000			FLTRn 输出 MIN 值低 32 位寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_OUT_MIN_L32 FLTR Minimum Low 32bits，滤波器最小限制值的低 32 位 。此字段仅在 FLTR _LIMIT_EN = 0x1 时有效 。

9.5.5.10. FLTRn+0x01C FLTRn_OUT_MAX_H32

默认值：0x00000000			FLTRn 输出 MAX 值高 32 位寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_OUT_MAX_H32 FLTR Maximum High 32bits，滤波器最大限制值的高 32 位 。此字段只有在 FLTR _LIMIT_EN = 0x1 有效 。

9.5.5.11. FLTRn+0x020 FLTRn_OUT_MAX_L32

默认值：0x00000000			FLTRn 输出 MAX 值低 32 位寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_OUT_MAX_L32 FLTR Maximum Low 32bits，滤波器最大限制值的低 32 位 。此字段只有在 FLTR _LIMIT_EN = 0x1 有效 。

9.5.5.12. FLTRn+0x050 FLTRn_FGP_A1

默认值：0x00000000			FLTRn FGP A1 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_A1 FLTR First Group A1，滤波器的第一组系数的 A1 寄存器，有符号 32-bit 整数。

9.5.5.13. FLTRn+0x054 FLTRn_FGP_A2

默认值：0x00000000			FLTRn FGP A2 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_A2 FLTR First Group A2，滤波器的第一组系数的 A2 寄存器，有符号 32-bit 整数。

9.5.5.14. FLTRn+0x058 FLTRn_FGP_A3

默认值：0x00000000			FLTRn FGP A3 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_A3 FLTR First Group A3，滤波器的第一组系数的 A3 寄存器，有符号 32-bit 整数。

9.5.5.15. FLTRn+0x05C FLTRn_FGP_A4

默认值：0x00000000			FLTRn FGP A4 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_A4 FLTR First Group A4，滤波器的第一组系数的 A4 寄存器，有符号 32-bit 整数。

9.5.5.16. FLTRn+0x060 FLTRn_FGP_B0

默认值：0x00000000			FLTRn FGP B0 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_B0 FLTR First Group B0，滤波器的第一组系数的 B0 寄存器，有符号 32-bit 整数。

9.5.5.17. FLTRn+0x064 FLTRn_FGP_B1

默认值：0x00000000			FLTRn FGP B1 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_B1 FLTR First Group B1，滤波器的第一组系数的 B1 寄存器，有符号 32-bit 整数。

9.5.5.18. FLTRn+0x068 FLTRn_FGP_B2

默认值：0x00000000			FLTRn FGP B2 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_B2 FLTR First Group B2，滤波器的第一组系数的 B2 寄存器，有符号 32-bit 整数。

9.5.5.19. FLTRn+0x06C FLTRn_FGP_B3

默认值：0x00000000			FLTRn FGP B3 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_B3 FLTR First Group B3，滤波器的第一组系数的 B3 寄存器，有符号 32-bit 整数。

9.5.5.20. FLTRn+0x070 FLTRn_FGP_B4

默认值：0x00000000			FLTRn FGP B4 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_FGP_B4 FLTR First Group B4，滤波器的第一组系数的 B4 寄存器，有符号 32-bit 整数。

9.5.5.21. FLTRn+0x074 FLTRn_SGP_A1

默认值：0x00000000			FLTRn SGP A1 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_A1 FLTR Second Group A1，滤波器的第二组系数的 A1 寄存器，有符号 32-bit 整数。

9.5.5.22. FLTRn+0x078 FLTRn_SGP_A2

默认值：0x00000000			FLTRn SGP A2 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_A2 FLTR Second Group A2，滤波器的第二组系数的 A2 寄存器，有符号 32-bit 整数。

9.5.5.23. FLTRn+0x07C FLTRn_SGP_A3

默认值：0x00000000			FLTRn SGP A3 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_A3 FLTR Second Group A3，滤波器的第二组系数的 A3 寄存器，有符号 32-bit 整数。

9.5.5.24. FLTRn+0x080 FLTRn_SGP_A4

默认值：0x00000000			FLTRn SGP A4 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_A4 FLTR Second Group A4，滤波器的第二组系数的 A4 寄存器，有符号 32-bit 整数。

9.5.5.25. FLTRn+0x084 FLTRn_SGP_B0

默认值：0x00000000			FLTRn SGP B0 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_B0 FLTR Second Group B0，滤波器的第二组系数的 B0 寄存器，有符号 32-bit 整数。

9.5.5.26. FLTRn+0x088 FLTRn_SGP_B1

默认值：0x00000000			FLTRn SGP B1 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_B1 FLTR Second Group B1，滤波器的第二组系数的 B1 寄存器，有符号 32-bit 整数。

9.5.5.27. FLTRn+0x08C FLTRn_SGP_B2

默认值：0x00000000			FLTRn SGP B2 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_B2 FLTR Second Group B2，滤波器的第二组系数的 B2 寄存器，有符号 32-bit 整数。

9.5.5.28. FLTRn+0x090 FLTRn_SGP_B3

默认值：0x00000000			FLTRn SGP B3 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_B3 FLTR Second Group B3，滤波器的第二组系数的 B3 寄存器，有符号 32-bit 整数。

9.5.5.29. FLTRn+0x094 FLTRn_SGP_B4

默认值：0x00000000			FLTRn SGP B4 系数寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FLTR_SGP_B4 FLTR Second Group B4，滤波器的第二组系数的 B4 寄存器，有符号 32-bit 整数。

9.5.5.30. 0x0FC PFM_VER

默认值：0x00000000			PFM 版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0100	版本寄存器 V1.0

9.6. Fast Fourier Transform (FFT)

Fast Fourier Transform （FFT，快速傅里叶变换）模块实现基于 FFT 算法的硬件加速，可应用于伺服驱动及相似领域所需的频率分析等场景。采用 FFT 模块进行傅里叶变换的计算，能够大幅度降低 CPU 的计算负载，并快速获取计算结果。

9.6.1. 特性说明

- 支持 8、16、32、64、128、256、512、1024、2048、4096 等多种点数的复数或实数信号的快速傅里叶变换 (FFT) 和逆变换 (IFFT)。
- 支持定点数 Q15 和 Q31 的输入计算。
- 支持浮点数据的输入输出转换。
- 支持多次命令计算，通过预先建立的描述符链表自动加载下一次命令的配置，实现连续计算。

9.6.2. 功能描述

9.6.2.1. FFT 工作流程

FFT 操作流程如下所示：

1. 通过对 SFTRST 位写 1 进行软件复位，完成复位后该位自动清零。

   如果在 FFT 模块计算过程中，对 SFTRST 位写 1，会强制停止 FFT 的计算，并且进行 FFT 模块复位。

2. 配置基本控制寄存器，包括所需状态的中断使能。

   中断源主要是计算饱和截止、总线错误、执行结束。需要相应的中断使能被允许，且相应的状态位置 1 时才会提起中断。这些中断共用一根中断线，所以需要读取相应状态位和中断允许位来决定中断原因。

3. 配置当前命令的计算参数：

   - NXT_CMD_EN = 0x0，即单次命令计算，表示完成一次命令计算即结束。

     如图 9–37 : fft 命令计算流程所示：

     a. 将对应的输入数据地址写入寄存器 FFT_INBUF。

     b. 将对应的输出数据地址写入寄存器 FFT_OUTBUF。

     c. 通过寄存器字段 FFT_LEN 选择 FFT 计算长度。

     d. 通过寄存器字段 IFFT_EN 选择使用 FFT 或 FFT 逆变换计算。

     e. 通过寄存器 OP_CMD 选择输入输出的数据类型。

        关于数据格式及配置说明，可查看数据输入说明及配置和数据输出说明及配置

     f. 通过对 CAL_EN 位写 1 启动 FFT 计算，计算完成后 CAL_EN 自动清零。

   - NXT_CMD_EN = 0x1，即多次命令计算，表示完成当前的计算命令后需要根据 NXT_ADDR 的地址获取下一次的命令的配置参数。

     多次命令计算是通过预先建立的描述符链表的方式将一系列的命令链接在一起。如图 9–37 : fft 命令计算流程所示。

     通过配置 NXT_CMD_EN = 0x1 和命令地址 NXT_CMD_ADDR，硬件自动加载下一次的命令配置，并且触发计算。只要最后一次的命令计算，配置 NXT_CMD_EN = 0x0，则硬件在完成此次计算后就会完成多次命令计算的操作。

4. 如有需要，处理中断。

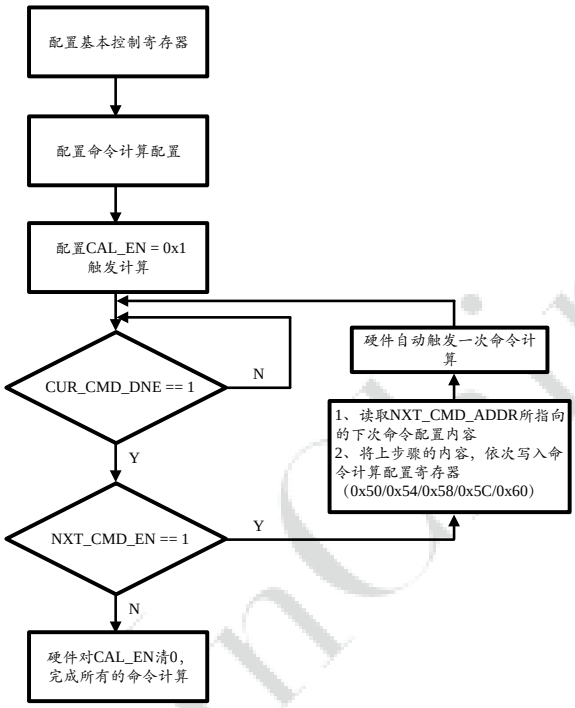


图 9–37 fft 命令计算流程

9.6.2.2. 数据输入说明及配置

在执行实数或复数的 FFT 计算时，FFT 模块的数据输入 x[n] 要求如下：

注：

在进行复数的 FFT 计算时，输入的数据需按照实部数据在前、虚部数据在后的格式填充，即 x[0]_real、x[0]_img、x[1]_real、x[1]_img…

表 9–30 FFT 模块的数据输入 x[n] 要求

定点数	输入数据范围	数值范围	保存方式
Q15	$[-2^{15}, 2^{15}-1]$	$[-1, (1-2^{-15})]$	采用 1 个 32–bit word 保存，保存在低 16 位。
Q31	$[-2^{31}, 2^{31}-1]$	$[-1, (1-2^{-31})]$	采用 1 个 32–bit word 保存，保存在整个 word，实部和虚部各占一个 word。

9.6.2.2.1. 输入配置

> **注：**
>
> 关于单精度数据的详细描述，可查看单精度浮点。

表 9–31 输入配置

序号	IND_TYP	FLOAT_IN_CVR	输入格式
1	0x0	0x0	Q31 定点数，数据为全实部
2	0x1	0x0	Q15 定点数，数据为全实部
3	0x2	0x0	Q31 定点数，数据为实部+虚部，按照实部在前、虚部在后的方式输入
4	0x3	0x0	Q15 定点数，数据为实部+虚部，按照实部在前、虚部在后的方式输入
5	0x1/0x3	0x1	单精度数据，需转换成 Q15 定点数，数据构成方式由 IND_TYP 决定
5	0x0/0x2	0x2	单精度数据，需转换成 Q31 定点数，数据构成方式由 IND_TYP 决定

9.6.2.3. 数据输出说明及配置

在实际应用中，FFT 计算的数据结果输出通常是复数形式，表示为实部和虚部。对于实数输入的 FFT，输出结果具有对称性，因此只需输出前一半的数据点加上中间的一个点。而对于复数输入的 FFT，输出结果不具有对称性，需要输出所有数据点。

- FFT 计算的每个输出点均先输出实部数据（即 y[n]_real）再输出虚部数据（即 y[n]_img）。
- 对于实数输入的 FFT，只输出 0 到 (N/2) 个点的数据，输出的数据长度为（N/2 + 1），输出的复数为 y[0]_real、y[0]_img、y[1]_real、y[1]_img、…、y[N/2 –1]_real、y[N/2 –1]_img、y[N/2]_real、y[N/2]_img。
- 对于复数输入的 FFT，需输出 N 个点复数的数据，对应的输出的复数为 y[0]_real、y[0]_img、y[1]_real、y[1]_img、…、y[N–1]_real、y[N–1]_img。

在硬件实现中，输出结果采用 Q15 或 Q31 的定点数表示，详细说明如下。输出结果需要进行缩放，输出缩放系数为 1/N。

表 9–32 FFT 模块的数据输出 y[n] 说明

定点数	数值范围	输出缩放系数	保存方式
Q15	[–1 , +1）	1/N	采用 1 个 32–bit word 保存，保存在低 16 位。
Q31	[–1 , +1）	1/N	采用 1 个 32–bit word 保存，保存在整个 word，实部和虚部各占一个 word。

9.6.2.3.1. 输出配置

> **注：**
>
> 关于单精度数据的详细描述，可查看单精度浮点。

表 9–33 输出配置

序号	OUT_TYP	ABS_OUT_EN	FLOAT_OUT_CVR	输出格式
1	0x0/0x2	0x0	0x0	Q31 定点数，按照实部在前、虚部在后的方式输出
2	0x1/0x3	0x0	0x0	Q15 定点数，按照实部在前、虚部在后的方式输出
3	any	0x0	0x1	单精度浮点数，按照实部在前、虚部在后的方式输出
4	0x0/0x1	0x1	any	Q31 定点数，按照幅值在前、相位在后的方式输出
5	0x2/0x3	0x1	any	Q15 定点数，按照幅值在前、相位在后的方式输出

9.6.2.3.2. 输出数据的幅值和相位计算

输出数据支持进行幅值和相位计算，即硬件对输出的复数结果进行幅值和相位的计算，减轻后续软件的计算负担。详细说明如下：

- 配置寄存器 ABS_OUT_EN = 0x1 使能幅值输出。使能后，输出结果为 N 个点实数的数据，即 y[0]_abs、y[0]_ph、y[1] _abs、y[1] _ph、…、y[N−1]_abs、y[N−1]_ph。
- 幅值的计算方式为 $y[n]\_abs = \sqrt{(y[n]\_real)^2 + (y[n]\_img)^2}$
- 相位的计算方式为 $y[n]\_ph = Atan2(y[n]\_img, y[n]\_real)$

9.6.2.4. 单精度浮点

如图所示，单精度浮点标准格式 IEEE754 由三个部分组成：

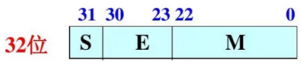


表 9–34 单精度浮点格式说明

组成部分	长度 (bit)	含义
符号位 (S)	1	• 0：正数 • 1：负数
阶码 (E)	8	以 2 为底数的指数，实际存储值 = 阶码真值 + 127 全 0 或全 1，需保留代表其它的意义

表 9–34 单精度浮点格式说明 (续)

组成部分	长度 (bit)	含义
尾数 (M)	23	采用隐含最高位为 1 的表示方法，实际尾数是 24 位。尾数真值 = 1 + 尾数

单精度浮点标准格式的非 0 浮点数真值为 $(-1)^{符号} * 2^{(阶码-127)} * (1+（尾数/2^{23})）$：

- 当阶码 E 的真值 = 阶码 E–127 且 0 < E < 255，表示的范围如下所示：
  - 正数范围： $+2^{+127} * (1+1-2^{-23})$ 到 $+2^{-126} * (1+0)$
  - 负数范围： $-2^{+127} * (1+1-2^{-23})$ 到 $-2^{-126} * (1+0)$
- 当 阶码 E = 0 或 E = 255，在 IEEE754 中表示特殊的数。特殊情况的含义如下所示：
  - 当阶码 (E) 为全 0，且尾数 (M) 不全为 0 时，表示该浮点数的尾数真值无需 + 1，即次正规化数，表示形式为 0.XXXXX 而不是 1.XXXXX。
  - 当阶码 (E) 为全 1，且尾数 (M) 全为 0 时，表示无穷大。正或负取决于符号位，如果符号位 (S) 为 0，表示正无穷大；如果符号位 (S) 为 1，表示负无穷大。
  - 当阶码 (E) 为全 1，且尾数 (M) 不全为 0 时，表示非数值数据（NaN），即产生 NaN 错误状态。使用此场景作为 NaN 错误检测。

    对于输入的浮点数，支持对 NaN 错误检测。

9.6.2.4.1. 浮点和定点数据转换

FFT 模块的输入输出数据支持浮点，但内部仍以定点数方式处理数据。因此，需要在浮点和定点数据之间进行转换，具体流程如下：

- 将浮点输入转换成 Q15 或 Q31 定点数
  - **使能**：通过寄存器 FLOAT_IN_CVR，使能对输入浮点数进行 Q15 或 Q31 定点数的转换。
  - **饱和处理**：FFT 输入输出处理的 Q15 或 Q31 定点数的表示的数值范围是 [–1,1)，浮点数转换也基于此范围。当输入的浮点数据存在超出 [–1,1) 的情况，转换的数据直接进行饱和截止输出。
    - 如果是浮点输入转 Q15，出现正向饱和则截止为 0x7FFF（对应 $2^{15}-1$），出现负饱和则截止为 0x8000（对应-2^{15}）。
    - 如果是浮点输入转 Q31，出现正向饱和则截止为 0x7FFFFFFF（对于 $2^{31}-1$），出现负向饱和则截止为 0x80000000（对于-2^{31}）。
- 将 Q15 或 Q31 定点数转换成浮点输出
  - **使能**：通过寄存器 FLOAT_OUT_CVR，使能对输出 Q15 或 Q31 定点数进行浮点数的转换。
  - **饱和处理**：FFT 输出的结果的范围为 [–1,1)，定点数转浮点数的过程中不存在饱和的情况。

9.6.2.5. SRAM 空间分配

FFT 的计算需要的数据 Buffer 空间，可以通过寄存器配置选择 SRAM_B0、SRAM_B1、SRAM_B2 和 SRAM_B3 作为 FFT 的输入输出 Buffer 空间。

表 9–35 FFT 的输入输出 Buffer 空间配置说明

寄存器配置	地址空间	空间大小 (KB)	配置建议
SRAM_B0	0x30138000 ~ 0x30140000	32	为避免 SRAM 空间浪费，按照下列建议选择： • 所需空间不超过 32 KB 时，使用 SRAM_B0 • 所需空间不超过 64 KB 时，使用 SRAM_B0 和 SRAM_B1 • 所需空间不超过 128 KB 时，使用 SRAM_B0、SRAM_B1、SRAM_B2 和 SRAM_B3
SRAM_B1	0x30130000 ~ 0x30138000	32	
SRAM_B2	0x30128000 ~ 0x30130000	32	
SRAM_B3	0x30120000 ~ 0x30128000	32	

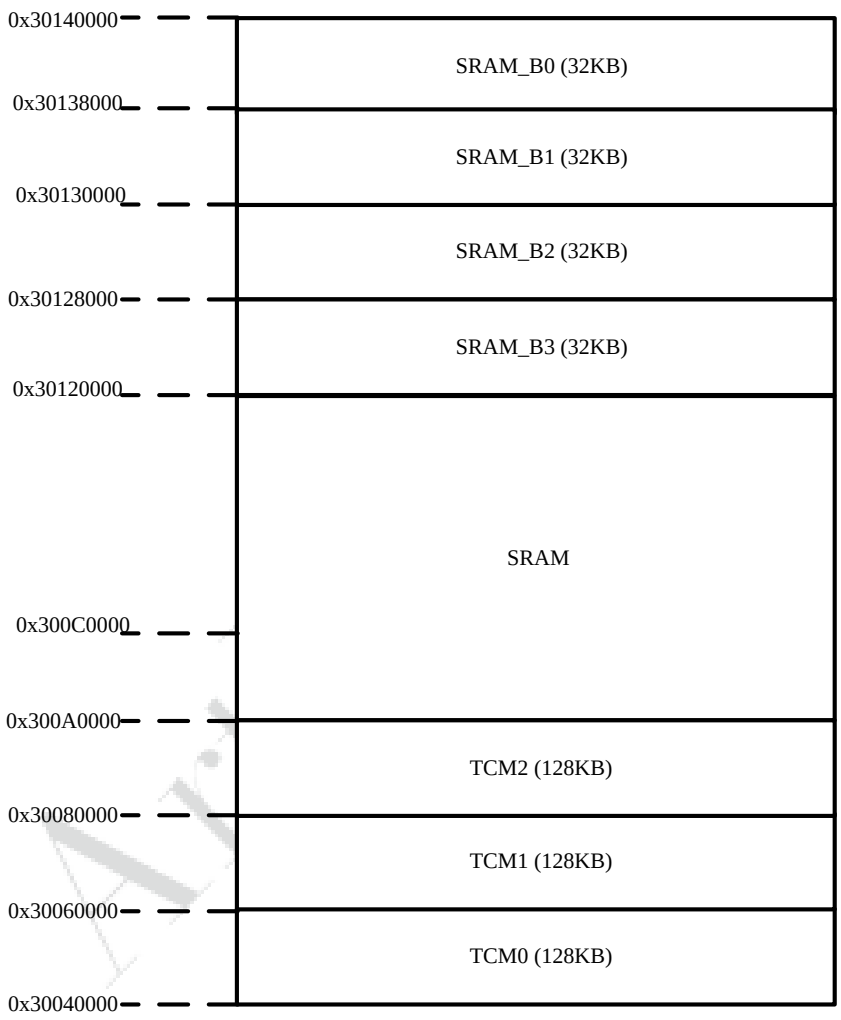


图 9–38 SRAM 空间分布

9.6.2.5.1. 单片空间使用

单片空间使用，就是将分配给 FFT 的 SRAM 空间当作是一片完整的 FFT_SRAM。

单片空间使用情况下，SRAM 空间分配建议如下：

- 使用 SRAM_B0，即单片 32 KB 空间
- 使用 SRAM_B0 + B1，即单片 64 KB 空间
- 使用 SRAM_B0 + B1 + B2 + B3，即单片 128 KB 空间

单片空间详细使用说明如下：

1. 每次触发 FFT 计算之前，将此片 FFT_SRAM 归属给 CPU，由 CPU 写入 FFT 计算命令所要的数据。
2. 输入数据准备完成后，切换 FFT_SRAM 的归属到 FFT，由 FFT 进行计算。

> **警告：**
>
> 在 FFT 计算的过程中，切勿操作 FFT_SRAM 的归属控制，只有在 FFT 完成计算后才可以切换 FFT_SRAM 的归属。

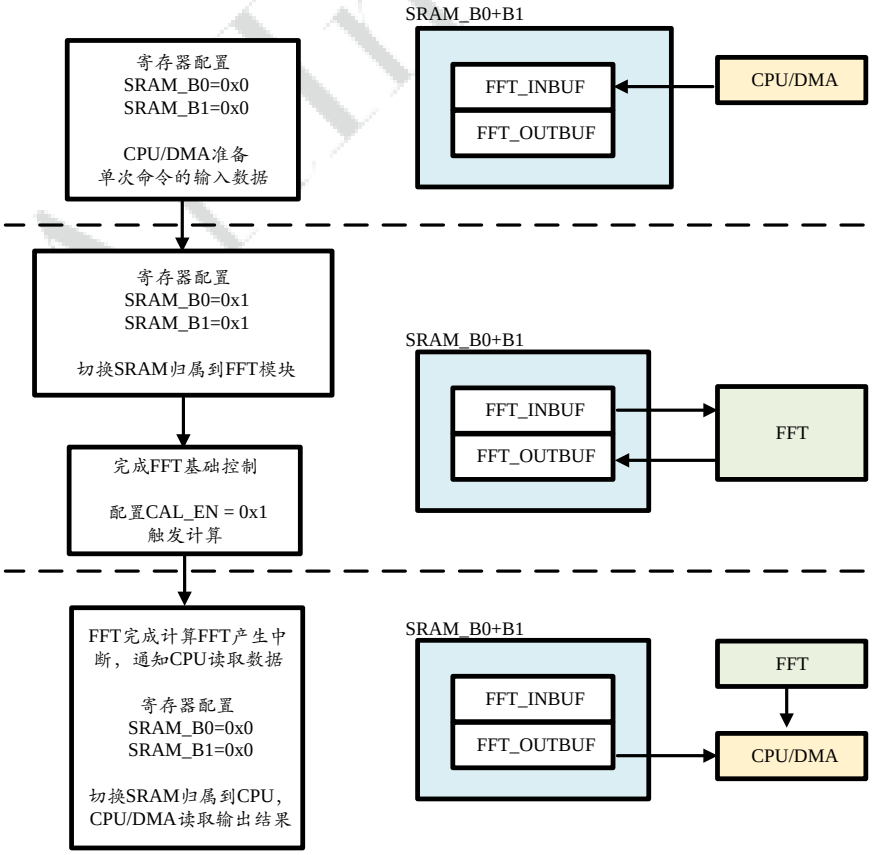
3. 完成计算后，产生中断信号，通知 CPU。
4. CPU 收到 FFT 计算完成后，再次切换 FFT_SRAM 归属到 CPU。

   此时，CPU/ DMA 就可以访问 FFT 计算完成的结果。

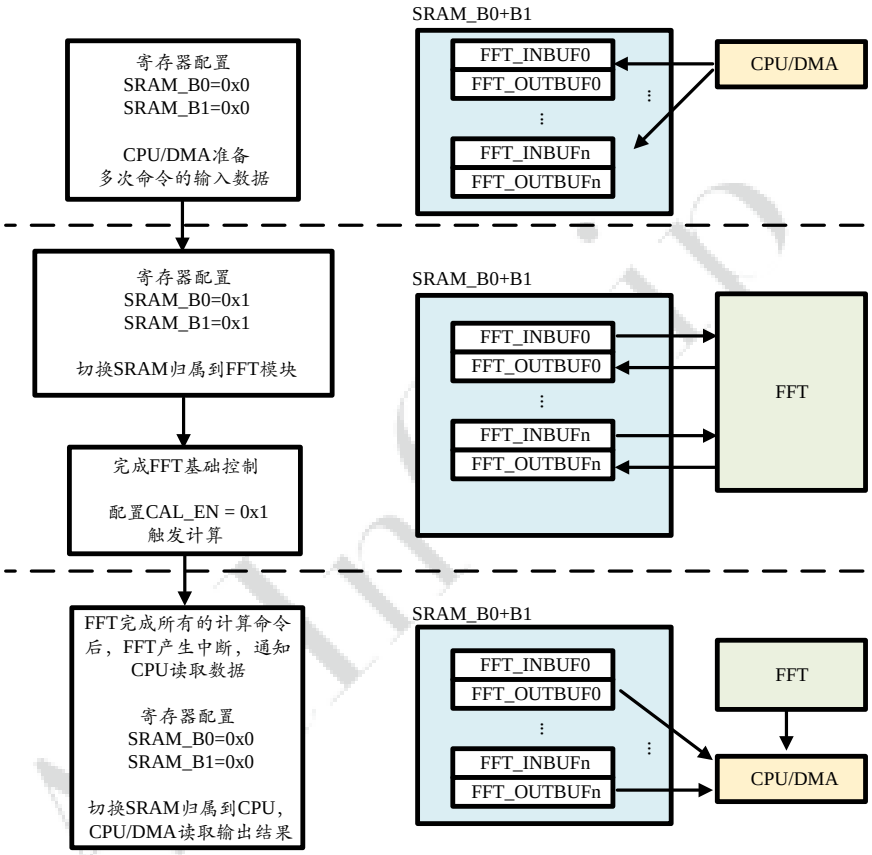
单片空间使用示例

以选择 SRAM_B0/ B1 用于 FFT 计算为例，单片空间操作流程如下：

- 单次命令：



• 多次命令：



9.6.2.5.2. 多片空间使用

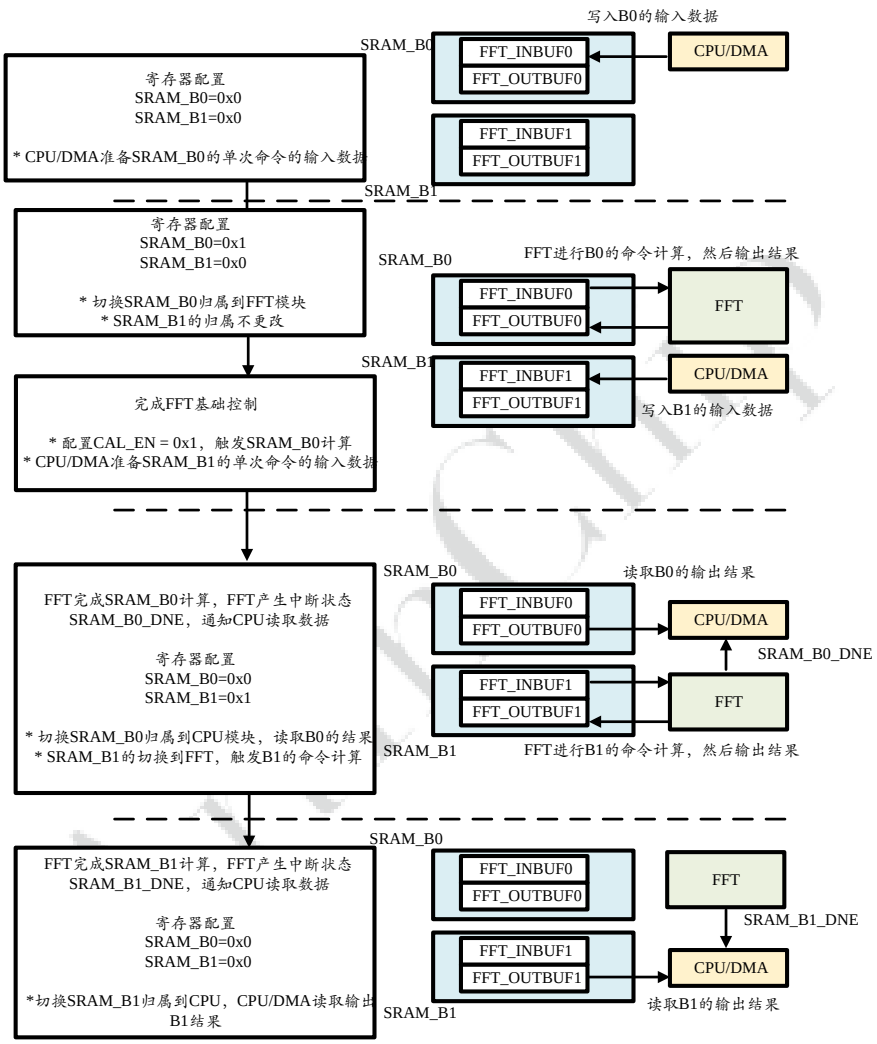
多片空间使用便于灵活地输入每次 FFT 计算的数据，可以更加高效的完成需要实时处理的数据。在此模式下，当其中一片空间归属于 FFT 进行计算的过程时，CPU 可以对其它空间写入下一笔 FFT 计算的输入数据。

多片空间使用的情况下，SRAM 空间分配建议如下：

- SRAM_B0 搭配 SRAM_B1，即 32KB 搭配 32 KB 的使用场景
- SRAM_B0 + B1 搭配 SRAM_B2+B3，即 64 KB 搭配 64 KB 的使用场景

如下图所示选择 SRAM_B0 和 SRAM_B1 两片空间用于 FFT 计算时：

- 当 SRAM_B0 空间处于 FFT 计算的过程中，SRAM_B1 的空间可供 CPU 进行数据准备。
- 当 SRAM_B0 的计算命令完成后，可以切换 SRAM_B0 的归属为 CPU，SRAM_B1 的归属为 FFT。CPU 在读取 SRAM_B0 结果的同时 FFT 也在进行 SRAM_B1 的计算。
- 当 SRAM_B0/1/2/3 的命令完成后，产生 CMD_SRAM_B0/1/2/3_DNE 的中断状态，用于区分不同空间的命令完成状态。



9.6.3. 寄存器列表

表 9–36 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	FFT_CTL	FFT 控制寄存器	0x0000 FFT_CTL
0x0004	FFT_STS	FFT 状态寄存器	0x0004 FFT_STS
0x0008	FFT_INT_EN	FFT 状态寄存器	0x0008 FFT_INT_EN
0x0010	SRAM_CFG	SRAM 配置寄存器	0x0010 SRAM_CFG
0x0014	SRAM_BASE	SRAM 基地址配置寄存器	0x0014 SRAM_BASE
0x0050	OP_CTL	OP_CTL 寄存器	0x0050 OP_CTL
0x0054	OP_CMD	OP_CMD 寄存器	0x0054 OP_CMD

表 9–36 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0058	FFT_CFG	FFT_CFG 寄存器	0x0058 FFT_CFG
0x05C	FFT_INBUF	FFT_INBUF 寄存器	0x05C FFT_INBUF
0x060	FFT_OUTBUF	FFT_OUTBUF 寄存器	0x060 FFT_OUTBUF
0x0FC	FFT_VER	FFT 版本寄存器	0x0FC FFT_VER

9.6.4. 寄存器描述

9.6.4.1. 0x0000 FFT_CTL

默认值：0x00000000			FFT 控制寄存器
位域	类型	默认值	描述
31	R/WAC	0x0	SFT_RST Soft Reset，软复位，此位置 1 复位，完成复位后自动清 0。
30:1	–	–	–
0	R/WAC	0x0	CAL_EN Calculation Enable，使能计算，使能计算后（包括所有的继发命令），此处自动清 0。

9.6.4.2. 0x0004 FFT_STS

默认值：0x00000000			FFT 状态寄存器
位域	类型	默认值	描述
31:24	–	–	–
23:16	R	0x0	CMD_DONE_CNT Command Done Counter，完成命令计数，每完成一次命令的计算，计数器进行自增 1。计数器只有在 CAL_EN = 0x1 的时候才开始计数。 当 CAL_EN 的数值从 0 变成 1 的时刻，会对此计数器进行清 0；当 CAL_EN 的数值从 1 变成 0 的时刻，此计数器停止计数。
15	R/W1C	0x0	CMD_SRAM_B3_DNE Comand in SRAM B3 Done，在 SRAM_B3 的计算命令完成。

默认值：0x00000000			FFT 状态寄存器
位域	类型	默认值	描述
			在使能 CMD_EN =0x1 时，判断 FFT_INBUF 的地址，如果地址属于 SRAM_B3，那么在 CAL_EN 拉低时，此位拉高，表示 SRAM_B3 计算命令完成。
14	R/W1C	0x0	CMD_SRAM_B2_DNE Comand in SRAM B2 Done，在 SRAM_B2 的计算命令完成。 在使能 CMD_EN =0x1 时，判断 FFT_INBUF 的地址，如果地址属于 SRAM_B2，那么在 CAL_EN 拉低时，此位拉高，表示 SRAM_B2 计算命令完成。
13	R/W1C	0x0	CMD_SRAM_B1_DNE Comand in SRAM B1 Done，在 SRAM_B1 的计算命令完成。 在使能 CMD_EN =0x1 时，判断 FFT_INBUF 的地址，如果地址属于 SRAM_B1，那么在 CAL_EN 拉低时，此位拉高，表示 SRAM_B1 计算命令完成。
12	R/W1C	0x0	CMD_SRAM_B0_DNE Comand in SRAM B0 Done，在 SRAM_B0 的计算命令完成。 在使能 CMD_EN =0x1 时，判断 FFT_INBUF 的地址，如果地址属于 SRAM_B0，那么在 CAL_EN 拉低时，此位拉高，表示 SRAM_B0 计算命令完成。
11:7	–	–	–
6	R/W1C	0x0	FP_NAN_ERR Float Point NAN Error，浮点数据出现 NAN 错误。
5	R/W1C	0x0	FP_SAT_ERR Float Point Saturation Error，浮点转定点时出现饱和错误。
4	R/W1C	0x0	AXI_NXT_CMD_ERR AXI Bus Next Command Error，继发命令 AXI 总线读错误。
3	R/W1C	0x0	AXI_BUS_ERR AXI Bus Error，AXI 总线读错误。
2	R/W1C	0x0	CMD_END Command End，命令计算结束，在完成所有的命令计算后产生此状态。
1	R/W1C	0x0	NXT_CMD_DNE Next Command Read Done，继发命令的数据已全部读入。
0	R/W1C	0x0	CUR_CMD_DNE

默认值：0x00000000			FFT 状态寄存器
位域	类型	默认值	描述
			Current Command Read Done，当前命令的数据已全部读入。

9.6.4.3. 0x0008 FFT_INT_EN

默认值：0x00000000			FFT 状态寄存器
位域	类型	默认值	描述
31:16	–	–	–
15	R/W	0x0	CMD_SRAM_B3_DNE_IE CMD_SRAM_B3_DNE Interrupt Enable，CMD_SRAM_B3_DNE 中断使能。 • 0x0：未使能 • 0x1：使能
14	R/W	0x0	CMD_SRAM_B2_DNE_IE CMD_SRAM_B2_DNE Interrupt Enable，CMD_SRAM_B2_DNE 中断使能。 • 0x0：未使能 • 0x1：使能
13	R/W	0x0	CMD_SRAM_B1_DNE_IE CMD_SRAM_B1_DNE Interrupt Enable，CMD_SRAM_B1_DNE 中断使能。 • 0x0：未使能 • 0x1：使能
12	R/W	0x0	CMD_SRAM_B0_DNE_IE CMD_SRAM_B0_DNE Interrupt Enable，CMD_SRAM_B0_DNE 中断使能。 • 0x0：未使能 • 0x1：使能
11:7	–	–	–
6	R/W	0x0	FP_NAN_ERR_IE FP_NAN_ERR Interrupt Enable，FP_NAN_ERR 中断使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	FP_SAT_ERR_IE FP_SAT_ERR Interrupt Enable，FP_SAT_ERR 中断使能。

默认值：0x00000000			FFT 状态寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
4	R/W	0x0	AXI_NXT_CMD_ERR_IE AXI_NXT_CMD_ERR Interrupt Enable，AXI_NXT_CMD_ERR 中断使能。 • 0x0：未使能 • 0x1：使能
3	R/W	0x0	AXI_BUS_ERR_IE AXI_BUS_ERR Interrupt Enable，AXI_BUS_ERR 中断使能。 • 0x0：未使能 • 0x1：使能
2	R/W	0x0	CMD_END_IE CMD_END Interrupt Enable，CMD_END 中断使能。 • 0x0：未使能 • 0x1：使能
1	R/W	0x0	NXT_CMD_DNE_IE NXT_CMD_DNE Interrupt Enable，NXT_CMD_DNE 中断使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	CUR_CMD_DNE_IE CUR_CMD_DNE Interrupt Enable，CUR_CMD_DNE 中断使能。 • 0x0：未使能 • 0x1：使能

9.6.4.4. 0x0010 SRAM_CFG

默认值：0x00000000			SRAM 配置寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	SRAM_B3_CFG SRAM_B3 Configure，SRAM_B3 配置。

默认值：0x00000000			SRAM 配置寄存器
位域	类型	默认值	描述
			• 0x0：SRAM_B3 的空间归属 CPU 读写 • 0x1：SRAM_B3 的空间归属 FFT 读写
2	R/W	0x0	SRAM_B2_CFG SRAM_B2 Configure，SRAM_B2 配置。 • 0x0：SRAM_B2 的空间归属 CPU 读写 • 0x1：SRAM_B2 的空间归属 FFT 读写
1	R/W	0x0	SRAM_B1_CFG SRAM_B1 Configure，SRAM_B1 配置。 • 0x0：SRAM_B1 的空间归属 CPU 读写 • 0x1：SRAM_B1 的空间归属 FFT 读写
0	R/W	0x0	SRAM_B0_CFG SRAM_B0 Configure，SRAM_B0 配置。 • 0x0：SRAM_B0 的空间归属 CPU 读写 • 0x1：SRAM_B0 的空间归属 FFT 读写

9.6.4.5. 0x0014 SRAM_BASE

默认值：0x00000000			SRAM 基地址配置寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	SRAM_BASE SRAM Base Address，SRAM 基地址。 如果使用 B0 空间，此处设置为 0x30138000 如果使用 B0+B1 空间，此处设置为 0x30130000 如果使用 B0+B1+B2+B3 空间，此处设置为 0x30120000

9.6.4.6. 0x0050 OP_CTL

默认值：0x00000000			OP_CTL 寄存器
位域	类型	默认值	描述
31:2	R/W	0x0	NXT_CMD_ADDR

默认值：0x00000000			OP_CTL 寄存器
位域	类型	默认值	描述
			继发命令指针，（当前命令执行完后读取）,此地址为高 30–bit 地址，低 2–bit 地址默认数值为 0。
1	–	–	–
0	R/W	0x0	NXT_CMD_EN Next Command Enable，继发命令使能。 • 0x0：未使能，表示当前的计数命令是最后一次的计算命令 • 0x1：使能，表示完成当前的计算命令后需要根据 NXT_ADDR 的地址获取下一次的命令的配置参数

9.6.4.7. 0x0054 OP_CMD

默认值：0x00000000			OP_CMD 寄存器
位域	类型	默认值	描述
31:9	–	–	–
8	R/W	0x0	ABS_OUT_EN Absolut Ouput Enable，复数绝对值输出。（ABS_OUT_EN 使能后，输出结果的固定位幅值 0、相位 0、幅值 1、相位 1、…、赋值 N、相位 N 的方式输出） • 0x0：未使能 • 0x1：使能
7:6	R/W	0x0	OUTD_TYP Output Data Type，输出数据格式。 • 0x0/0x2：输出 FFT 计算的实部+虚部，采用 Q31 的格式输出 • 0x1/0x3：输出 FFT 计算的实部+虚部，采用 Q15 的格式输出
5:4	R/W	0x0	IND_TYP Input Data type，输入数据格式。 • 0x0：实数 Q31 FFT 计算 • 0x1：实数 Q15 FFT 计算 • 0x2：复数 Q31 FFT 计算 • 0x3：复数 Q15 FFT 计算
3:2	R/W	0x0	FLOAT_OUT_CVR

默认值：0x00000000			OP_CMD 寄存器
位域	类型	默认值	描述
			Float Output Convert Type，浮点输出数据转换类型。 • 0x0：输出为定点数，无需转换 • 0x1：Q15 定点数转换为单精度浮点数输出 • 0x2：Q31 定点数转换为单精度浮点数输出 • 0x3：Reserved
1:0	R/W	0x0	FLOAT_IN_CVR Float Input Convert Type，浮点输入数据转换类型。 • 0x0：输入为定点数，无需转换 • 0x1：输入为单精度浮点数，需转换成 Q15 定点数 • 0x2：输入为单精度浮点数，需转换成 Q31 定点数 • 0x3：Reserved

9.6.4.8. 0x0058 FFT_CFG

默认值：0x00000000			FFT_CFG 寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:4	R/W	0x0	FFT_LEN FFT Length，FFT 计算长度。 • 0x0：8 点 FFT/IFFT 计算 • 0x1：16 点 FFT/IFFT 计算 • 0x2：32 点 FFT/IFFT 计算 • 0x3：64 点 FFT/IFFT 计算 • ... • 0x9：4096 点 FFT/IFFT 计算 • others：Reserved
3:1	–	–	–
0	R/W	0x0	IFFT_EN Invert FFT Enable，FFT 逆变换使能。

默认值：0x00000000			FFT_CFG 寄存器
位域	类型	默认值	描述
			• 0x0：未使能，进行 FFT 变换计算 • 0x1：使能，进行 FFT 逆变换计算

9.6.4.9. 0x05C FFT_INBUF

默认值：0x00000000			FFT_INBUF 寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FFT_INBUF FFT Input Buffer Address，FFT 输入数据的起始地址。

9.6.4.10. 0x060 FFT_OUTBUF

默认值：0x00000000			FFT_OUTBUF 寄存器
位域	类型	默认值	描述
31:0	R/W	0x0	FFT_OUTBUF FFT Output Buffer Address，FFT 输出数据的起始地址。

9.6.4.11. 0x0FC FFT_VER

默认值：0x00000000			FFT 版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0100	版本寄存器 V1.0

9.7. Programmable Gain Amplifier (PGA)

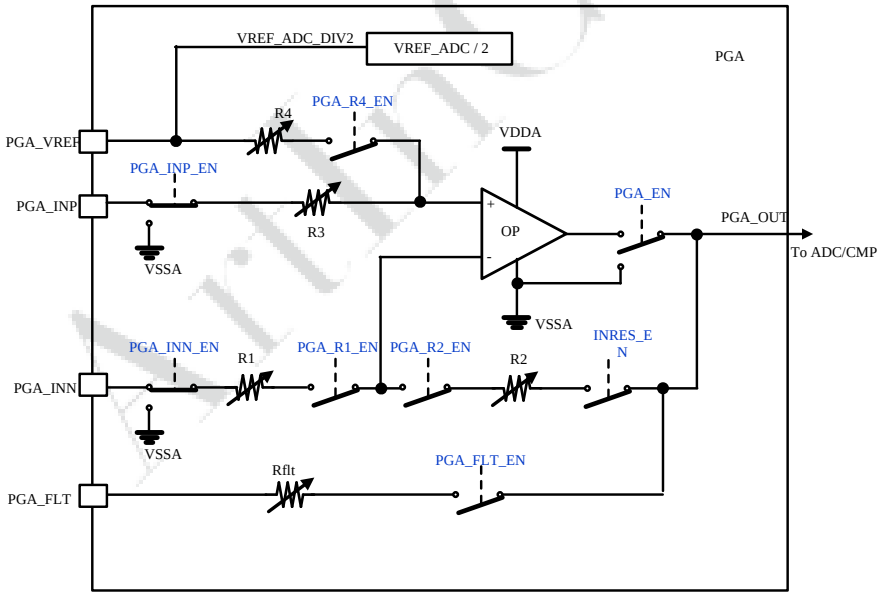
Programmable Gain Amplifier（PGA，增益可调放大器），对外部模拟小信号的放大后，输出至内部 ADC 或 CPM 模块，实施采样转换或电压比较。

M7000 系列芯片最多支持 4 x PGA 模块，分别为 PGA0/ 1/ 2/ 3。

9.7.1. 特性说明

- 支持源随模式、正向放大、反向放大和运放模式
- 增益放大模式最高支持 64 倍的放大
- 支持内部或外部参考电压设置
- 支持输出信号输出到 ADC 或 CPM 模块的输入端口
- 支持输出 RC 滤波

9.7.2. 原理框图



9.7.3. 功能描述

9.7.3.1. 典型工作模式

本节介绍 PGA 的几种典型工作模式及其初始化配置说明。

表 9–37 PGA 典型工作模式及配置说明

初始化配置参数	模式 1	模式 2	模式 3	模式 4	模式 5
	源跟随器	增益放大	增益放大	负增益放大	运放模式
PGA_INP_EN	0x1	0x1	0x1	0x0	0x1
PGA_R4_EN	0x0	0x0	0x1	0x1	0x0
PGA_R1_EN	0x0	0x1	0x1	0x1	0x1

表 9–37 PGA 典型工作模式及配置说明 (续)

初始化配置参数	模式 1	模式 2	模式 3	模式 4	模式 5
	源跟随器	增益放大	增益放大	负增益放大	运放模式
PGA_R2_EN	0x1	0x1	0x1	0x1	0x0
INRES_EN	0x1	0x1	0x1	0x1	0x0
PGA_INN_EN	–	0x0	0x1	0x1	0x1
PGA_EN	0x1	0x1	0x1	0x1	0x1
R4	–	–	可编程电阻，R2 = R4	可编程电阻，R3=R4	短路
R3	短路	短路	可编程电阻，R1 = R3	可编程电阻，R3=R4	短路
R2	短路	可编程电阻	可编程电阻，R2 = R4	可编程电阻，R1=R2	短路
R1	–	可编程电阻	可编程电阻，R1 = R3	可编程电阻，R1=R2	短路

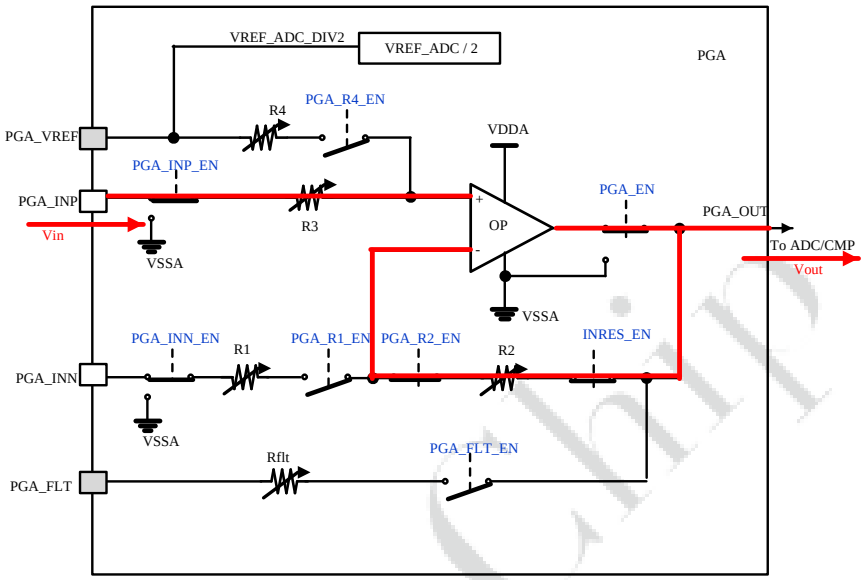
9.7.3.1.1. 模式 1：源跟随器

在此模式下，输入信号直接连接到输出端，通过单位增益放大器实现信号的传递。由于没有额外的增益或衰减，因此输出信号与输入信号相同。

初始化配置如下：

- PGA_INP_EN = 0x1，使能，开关选通 PGA_INP 输入通路
- PGA_R4_EN = 0x0，未使能，开关处于断开
- PGA_R1_EN = 0x0，未使能，开关处于断开
- PGA_R2_EN = 0x1，使能，开关处于闭合
- INRES_EN = 0x1，使能，开关处于闭合
- PGA_EN = 0x1，使能，开关选通运放的输出
- R3 配置为短路、R2 配置为短路

得出如下的拓扑关系，实现的功能为源跟随器 Vout = Vin。

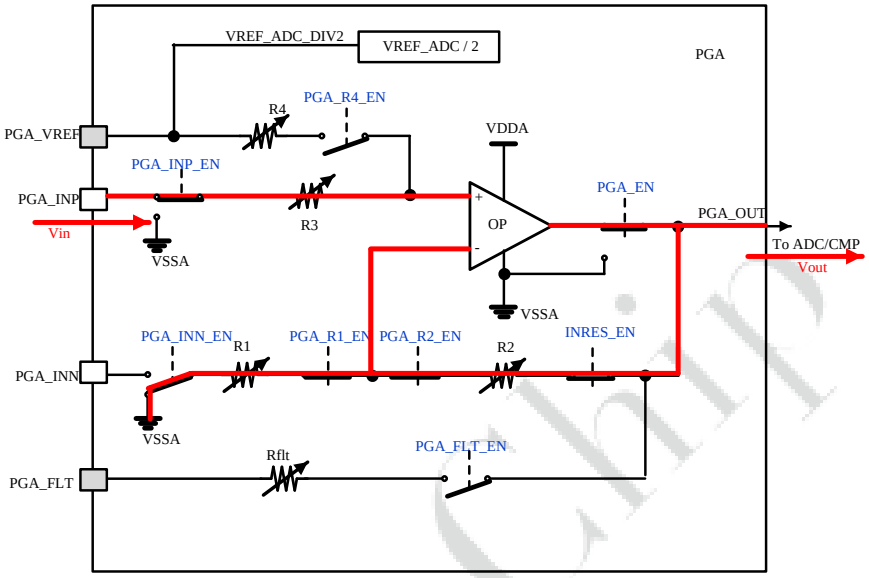


9.7.3.1.2. 模式 2：增益放大

在此模式下，通过设置 R1 和 R2 的阻值比来实现所需的增益。例如，如果 R1 = R2，则增益为 2。用户可以根据实际需求选择合适的电阻值来获得所需的增益。初始化配置：

- PGA_INP_EN = 0x1，使能，开关选通 PGA_INP 输入通路
- PGA_R4_EN = 0x0，未使能，开关处于断开
- PGA_R1_EN = 0x1，使能，开关处于闭合
- PGA_R2_EN = 0x1，使能，开关处于闭合
- INRES_EN = 0x1，使能，开关处于闭合
- PGA_INN_EN = 0x0，未使能，开关导通至模拟地 VSSA
- PGA_EN = 0x1，使能，开关选通运放的输出
- R3 配置为短路状态，R1、R2 配置为可编程电阻，阻值根据应用选择

得出如下的拓扑关系，实现的功能为增益放大，放大公式为：Vout = Vin *（R1+ R2）/ R1



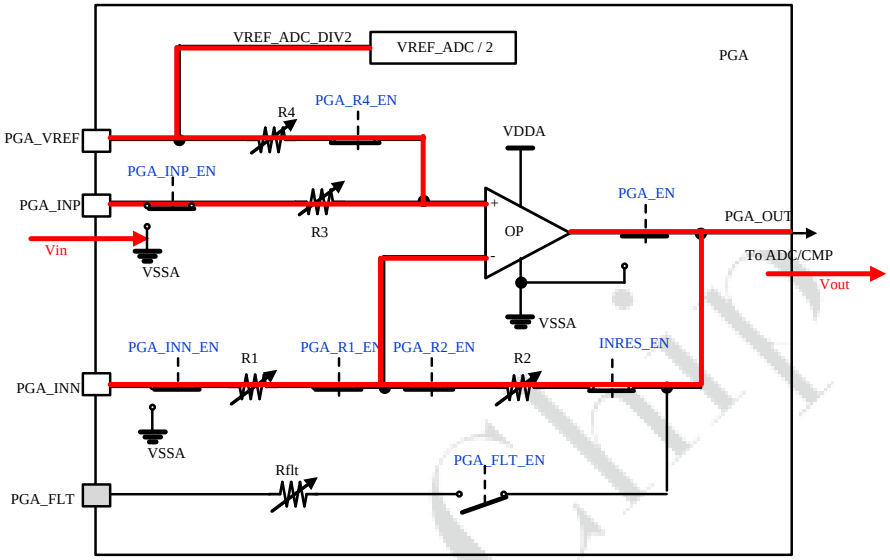
9.7.3.1.3. 模式 3：增益放大

在此模式下，通过设置 R1、R2、R3 和 R4 的阻值比来实现所需的差分增益。例如，如果 R1 = R3 且 R2 = R4，则差分增益为 1。

初始化配置：

- PGA_INP_EN = 0x1，使能，开关选通 PGA_INP 输入通路
- PGA_R4_EN = 0x1，使能，开关处于闭合
- PGA_R1_EN = 0x1，使能，开关处于闭合
- PGA_R2_EN = 0x1，使能，开关处于闭合
- INRES_EN = 0x1，使能，开关处于闭合
- PGA_INN_EN = 0x1，使能，开关导通至输入 PGA_INN
- PGA_EN = 0x1，使能，开关选通运放的输出
- R1、R2、R3、R4 配置为可编程电阻，阻值根据应用选择，设置 R1 = R3，R2 = R4

得出如下的拓扑关系，实现的功能为增益放大，放大公式为：Vout = Vref + Vin *（R2/R1）



注：

关于 Vref 的描述，可查看参考电压配置。

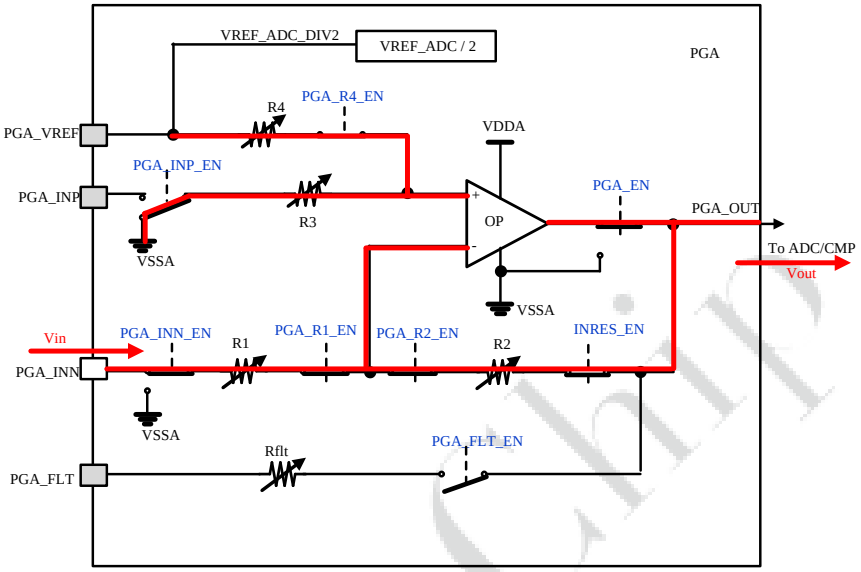
9.7.3.1.4. 模式 4：负增益放大模式

在此模式下，通过设置 R1、R2 和 R3、R4 阻值比来实现所需的反相增益。例如，如果 R1 = R2，则反相增益为 –1。

初始化配置：

- PGA_INP_EN = 0x0，未使能，开关选通模拟地 VSSA
- PGA_R4_EN = 0x1，使能，开关处于闭合
- PGA_R1_EN = 0x1，使能，开关处于闭合
- PGA_R2_EN = 0x1，使能，开关处于闭合
- INRES_EN = 0x1，使能，开关处于闭合
- PGA_INN_EN = 0x1，使能，开关导通至输入通路 PGA_INN
- PGA_EN = 0x1，使能，开关选通运放的输出
- R1、R2 和 R3、R4 配置为可编程电阻，R1=R2，R3=R4

得出如下的拓扑关系，实现的功能为增益放大，放大公式为： Vout = Vin*（– R2/R1 ）+ Vref



注：

关于 Vref 的描述，可查看参考电压配置。

9.7.3.1.5. 模式 5：运放模式

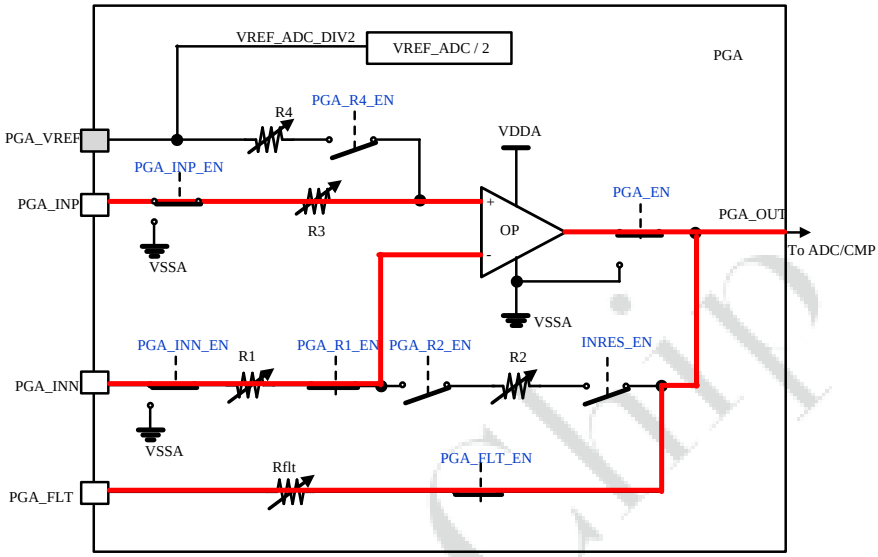
在此模式下，PGA 内部的四个电阻都被短路，用户可以根据自己的需求连接外部电阻和其它元件来构建特定的电路拓扑结构。由于没有内部增益或衰减，因此用户可以完全控制电路的行为。

PGA 的内置运放支持 Normal 和 High Speed 两种模式，High Speed 模式相对 Normal 模式提供更高的动态响应的性能。

初始化配置：

- PGA_INP_EN = 0x1，使能，开关选通 PGA_INP 输入通路
- PGA_R4_EN = 0x0，未使能，开关处于断开
- PGA_R1_EN = 0x1，使能，开关处于闭合
- PGA_R2_EN = 0x0，未使能，开关处于断开
- INRES_EN = 0x0，未使能，开关处于断开
- PGA_INN_EN = 0x1，使能，开关导通至 PGA_INN 输入通路
- PGA_EN = 0x1，使能，开关选通运放的输出
- R1、R2、R3、R4 配置短路状态

此模式为运放模式，输入输出均开放到管脚，用于实现其它功能。



9.7.3.2. 可编程电阻配置

在 PGA 中，R1、R2、R3 和 R4 均为可编程电阻，并且均可以配置为短路状态，即阻值为 0 Ω。

可编程电阻允许用户根据具体的应用需求来调整电路的性能。

表 9–38 电阻设置范围

电阻	阻值 (kΩ)
R1	0/ 10
R2	0/ 10/ 20/ 30/ 40/ 50/ 60/ 70/ 80/ 150/ 160/ 310/ 320/ 630/ 640
R3	0/ 10
R4	0/ 10/ 20/ 30/ 40/ 50/ 60/ 70/ 80/ 150/ 160/ 310/ 320/ 630/ 640

9.7.3.3. 参考电压配置

在 PGA 模块中，可通过寄存器选择不同的参考电压方式：

- 采用内部参考电压：通过模块 VREF_ADC/2 实现。

  VREF_ADC/2 模块对 ADC 的参考电压进行 1/2 分压后提供 VREF_ADC_DIV2 电压，确保内部参考电压的稳定性和一致性，同时也简化了电路设计。

- 采用外部参考电压：参考电压通过管脚 PGA_VREF 提供，用户可根据不同的应用需求调整参考电压。

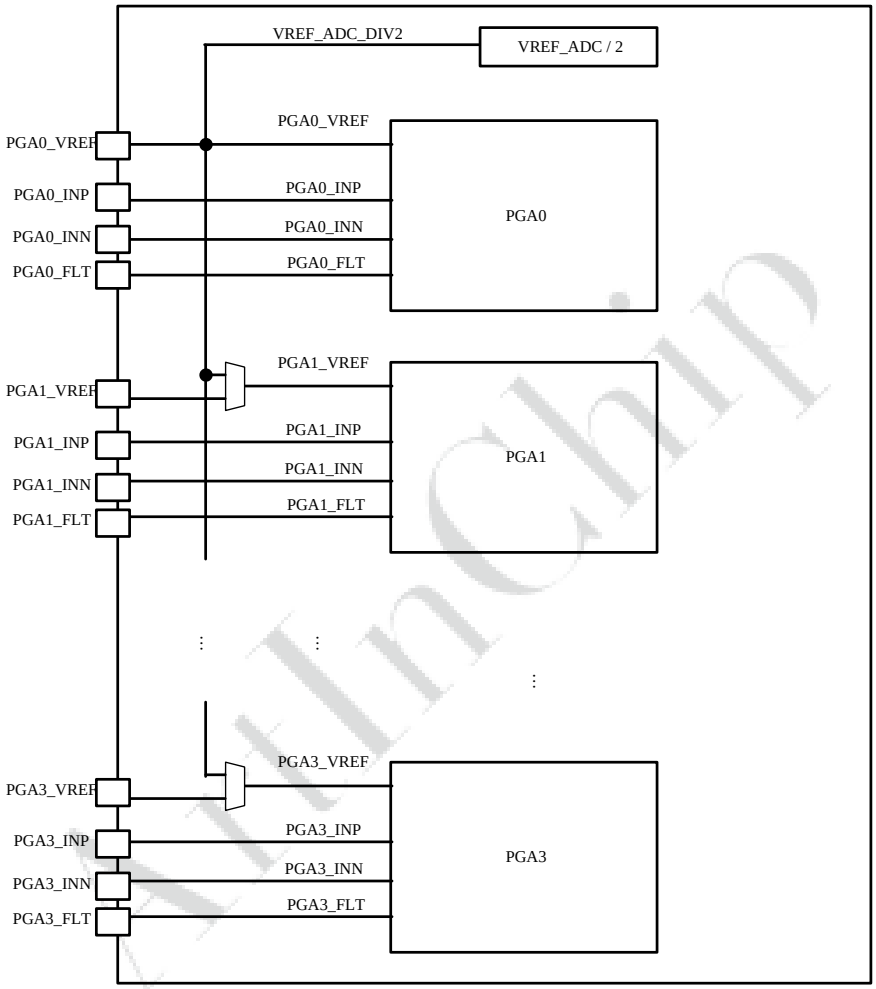


图 9–39 内部参考电压具体连接关系

9.7.3.4. 低通滤波功能

PGA 模块支持 RC 低通滤波功能。PGA 模块内置低通滤波的电阻 Rflt，只需外挂滤波电容 Cflt 即可实现低通滤波功能。连接方式参考如下：

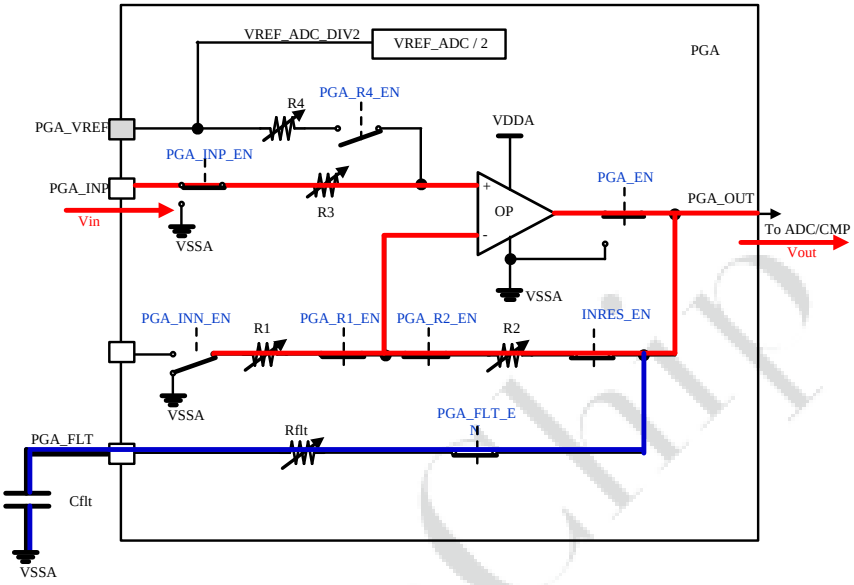


图 9–40 增益放大模式下增加低通滤波通路示例

以一个典型的滤波应用为例，按照下列流程配置低通滤波功能：

1. 将外挂滤波电容 Cflt 连接到适当的位置。
2. 通过寄存器使能低通滤波器的通路。

低通滤波器的截止频率计算公式为 $F_{cutoff} = 1/(2\Pi * Rflt * Cflt)$。其中，Rflt 是一个可编程电阻，其值可以根据具体需求进行设置。

9.7.4. 寄存器列表

表 9–39 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	PGA0_CTL	PGA0 控制寄存器	0x0000 PGA0_CTL
0x0004	PGA1_CTL	PGA1 控制寄存器	0x0004 PGA1_CTL
0x0008	PGA2_CTL	PGA2 控制寄存器	0x0008 PGA2_CTL
0x000C	PGA3_CTL	PGA3 控制寄存器	0x000C PGA3_CTL
0x0020	PGA0_RES_SET	PGA0_RES_SET 寄存器	0x0020 PGA0_RES_SET
0x0024	PGA1_RES_SET	PGA1_RES_SET 寄存器	0x0024 PGA1_RES_SET
0x0028	PGA2_RES_SET	PGA2_RES_SET 寄存器	0x0028 PGA2_RES_SET
0x002C	PGA3_RES_SET	PGA3_RES_SET 寄存器	0x002C PGA3_RES_SET

表 9–39 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
0x0040	PGA_REF_CTL	PGA_REF_CTL 寄存器	0x0040 PGA_REF_CTL
0x00FC	PGA_VER	PGA 模块版本寄存器	0x00FC PGA_VER

9.7.5. 寄存器描述

9.7.5.1. 0x0000 PGA0_CTL

默认值：0x80800000			PGA0 控制寄存器
位域	类型	默认值	描述
31:24	R/W	0x80	PGA0_P_CALSEL
23:16	R/W	0x80	PGA0_N_CALSEL
15:13	–	–	–
12:11	R/W	0x0	PGA0_ATB_SEL PGA0_ATB Select，测试信号选择。
10	R/W	0x0	PGA0_EN_VCM PGA0 Enabel VCM，PGA0 使能选择内部的 VCM 作为 PGA0 REF 输入。 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	PGA0_EN_ATB PGA0 Enable ATB，测试信号选择使能。
8	R/W	0x0	PGA_OP_MODE PGA Operator Mode，PGA 运放模式。 • 0x0：Normal Mode • 0x1：High Speed Mode
7	R/W	0x0	PGA_FLT_EN PGA Filter Enable，PGA 滤波使能。 • 0x0：使能 • 0x1：未使能
6	R/W	0x0	PGA_R4_EN

默认值：0x80800000			PGA0 控制寄存器
位域	类型	默认值	描述
			PGA R4 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
5	R/W	0x0	PGA_R2_EN PGA R2 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
4	R/W	0x0	PGA_R1_EN PGA R1 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
3	R/W	0x0	PGA_INRES_EN PGA Internal Resistor Net Enable，PGA 内部反馈电阻网络使能。 • 0x0：使能 • 0x1：未使能
2	R/W	0x0	PGA_INN_EN PGA Negative Input Enable，PGA 负相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INN
1	R/W	0x0	PGA_INP_EN PGA Positive Input Enable，PGA 正相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INP
0	R/W	0x0	PGA_EN PGA Enable，PGA 使能。 • 0x0：PGA 未使能，未使能运放进入停止工作，降低功耗 • 0x1：PGA 使能

9.7.5.2. 0x0004 PGA1_CTL

默认值：0x80800000			PGA1 控制寄存器
位域	类型	默认值	描述
31:24	R/W	0x80	PGA1_P_CALSEL
23:16	R/W	0x80	PGA1_N_CALSEL
15:13	–	–	–
12:11	R/W	0x0	PGA1_ATB_SEL PGA1_ATB Select，测试信号选择。
10	R/W	0x0	PGA1_EN_VCM PGA1 Enabel VCM，PGA1 使能选择内部的 VCM 作为 PGA1 REF 输入。 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	PGA1_EN_ATB PGA1 Enable ATB，测试信号选择使能。
8	R/W	0x0	PGA_OP_MODE PGA Operator Mode，PGA 运放模式。 • 0x0：Normal Mode • 0x1：High Speed Mode
7	R/W	0x0	PGA_FLT_EN PGA Filter Enable，PGA 滤波使能。 • 0x0：使能 • 0x1：未使能
6	R/W	0x0	PGA_R4_EN PGA R4 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
5	R/W	0x0	PGA_R2_EN PGA R2 开关使能。 • 0x0：开关断开 • 0x1：开关闭合

默认值：0x80800000			PGA1 控制寄存器
位域	类型	默认值	描述
4	R/W	0x0	PGA_R1_EN PGA R1 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
3	R/W	0x0	PGA_INRES_EN PGA Internal Resistor Net Enable，PGA 内部反馈电阻网络使能。 • 0x0：使能 • 0x1：未使能
2	R/W	0x0	PGA_INN_EN PGA Negative Input Enable，PGA 负相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INN
1	R/W	0x0	PGA_INP_EN PGA Positive Input Enable，PGA 正相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INP
0	R/W	0x0	PGA_EN PGA Enable，PGA 使能。 • 0x0：PGA 未使能，未使能运放进入停止工作，降低功耗 • 0x1：PGA 使能

9.7.5.3. 0x0008 PGA2_CTL

默认值：0x80800000			PGA2 控制寄存器
位域	类型	默认值	描述
31:24	R/W	0x80	PGA2_P_CALSEL
23:16	R/W	0x80	PGA2_N_CALSEL
15:13	–	–	–
12:11	R/W	0x0	PGA2_ATB_SEL

默认值：0x80800000			PGA2 控制寄存器
位域	类型	默认值	描述
			PGA2_ATB Select，测试信号选择。
10	R/W	0x0	PGA2_EN_VCM PGA2 Enabel VCM，PGA2 使能选择内部的 VCM 作为 PGA2 REF 输入。 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	PGA2_EN_ATB PGA2 Enable ATB，测试信号选择使能。
8	R/W	0x0	PGA_OP_MODE PGA Operator Mode，PGA 运放模式。 • 0x0：Normal Mode • 0x1：High Speed Mode
7	R/W	0x0	PGA_FLT_EN PGA Filter Enable，PGA 滤波使能。 • 0x0：使能 • 0x1：未使能
6	R/W	0x0	PGA_R4_EN PGA R4 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
5	R/W	0x0	PGA_R2_EN PGA R2 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
4	R/W	0x0	PGA_R1_EN PGA R1 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
3	R/W	0x0	PGA_INRES_EN PGA Internal Resistor Net Enable，PGA 内部反馈电阻网络使能。

默认值：0x80800000			PGA2 控制寄存器
位域	类型	默认值	描述
			• 0x0：使能 • 0x1：未使能
2	R/W	0x0	PGA_INN_EN PGA Negative Input Enable，PGA 负相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INN
1	R/W	0x0	PGA_INP_EN PGA Positive Input Enable，PGA 正相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INP
0	R/W	0x0	PGA_EN PGA Enable，PGA 使能。 • 0x0：PGA 未使能，未使能运放进入停止工作，降低功耗 • 0x1：PGA 使能

9.7.5.4. 0x000C PGA3_CTL

默认值：0x80800000			PGA3 控制寄存器
位域	类型	默认值	描述
31:24	R/W	0x80	PGA3_P_CALSEL
23:16	R/W	0x80	PGA3_N_CALSEL
15:13	–	–	–
12:11	R/W	0x0	PGA3_ATB_SEL PGA3_ATB Select，测试信号选择。
10	R/W	0x0	PGA3_EN_VCM PGA3 Enabel VCM，PGA3 使能选择内部的 VCM 作为 PGA3 REF 输入。 • 0x0：未使能 • 0x1：使能
9	R/W	0x0	PGA3_EN_ATB

默认值：0x80800000			PGA3 控制寄存器
位域	类型	默认值	描述
			PGA3 Enable ATB，测试信号选择使能。
8	R/W	0x0	PGA_OP_MODE PGA Operator Mode，PGA 运放模式。 • 0x0：Normal Mode • 0x1：High Speed Mode
7	R/W	0x0	PGA_FLT_EN PGA Filter Enable，PGA 滤波使能。 • 0x0：使能 • 0x1：未使能
6	R/W	0x0	PGA_R4_EN PGA R4 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
5	R/W	0x0	PGA_R2_EN PGA R2 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
4	R/W	0x0	PGA_R1_EN PGA R1 开关使能。 • 0x0：开关断开 • 0x1：开关闭合
3	R/W	0x0	PGA_INRES_EN PGA Internal Resistor Net Enable，PGA 内部反馈电阻网络使能。 • 0x0：使能 • 0x1：未使能
2	R/W	0x0	PGA_INN_EN PGA Negative Input Enable，PGA 负相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INN
1	R/W	0x0	PGA_INP_EN

默认值：0x80800000			PGA3 控制寄存器
位域	类型	默认值	描述
			PGA Positive Input Enable，PGA 正相输入使能。 • 0x0：未使能，开关选通模拟地 • 0x1：使能，开关选通 PGA_INP
0	R/W	0x0	PGA_EN PGA Enable，PGA 使能。 • 0x0：PGA 未使能，未使能运放进入停止工作，降低功耗 • 0x1：PGA 使能

9.7.5.5. 0x0020 PGA0_RES_SET

默认值：0x00000000			PGA0_RES_SET 寄存器
位域	类型	默认值	描述
31:27	R/W	0x0	PGA0_RSV
26:24	R/W	0x0	PGA_RFLT_SET PGA Resistor Filter Set，PGA 滤波电阻设置。 • 0x0：短路 • 0x1：50 Ω • 0x2：80 Ω • 0x3：100 Ω • 0x4：130 Ω • 0x5：160 Ω • 0x6：200 Ω • others：短路
23:20	–	–	–
19:16	R/W	0x0	PGA_R4_SET PGA Resistor 4 Set，PGA 电阻 4 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ

默认值：0x00000000			PGA0_RES_SET 寄存器
位域	类型	默认值	描述
			• 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
15:13	–	–	–
12	R/W	0x0	PGA_R3_SET PGA Resistor 3 Set，PGA 电阻 3 设置。 • 0x0：短路 • 0x1：10 kΩ
11:8	–	–	–
7:4	R/W	0x0	PGA_R2_SET PGA Resistor 2 Set，PGA 电阻 2 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ

默认值：0x00000000			PGA0_RES_SET 寄存器
位域	类型	默认值	描述
			• 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
3:1	–	–	–
0	R/W	0x0	PGA_R1_SET PGA Resistor 1 Set，PGA 电阻 1 设置。 • 0x0：短路 • 0x1：10 kΩ

9.7.5.6. 0x0024 PGA1_RES_SET

默认值：0x00000000			PGA1_RES_SET 寄存器
位域	类型	默认值	描述
31:27	R/W	0x0	PGA1_RSV
26:24	R/W	0x0	PGA_RFLT_SET PGA Resistor Filter Set，PGA 滤波电阻设置。 • 0x0：短路 • 0x1：50 Ω • 0x2：80 Ω • 0x3：100 Ω • 0x4：130 Ω • 0x5：160 Ω • 0x6：200 Ω • others：短路
23:20	–	–	–
19:16	R/W	0x0	PGA_R4_SET PGA Resistor 4 Set，PGA 电阻 4 设置。 • 0x0：短路 • 0x1：10 kΩ

默认值：0x00000000			PGA1_RES_SET 寄存器
位域	类型	默认值	描述
			• 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
15:13	–	–	–
12	R/W	0x0	PGA_R3_SET PGA Resistor 3 Set，PGA 电阻 3 设置。 • 0x0：短路 • 0x1：10 kΩ
11:8	–	–	–
7:4	R/W	0x0	PGA_R2_SET PGA Resistor 2 Set，PGA 电阻 2 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ

默认值：0x00000000			PGA1_RES_SET 寄存器
位域	类型	默认值	描述
			• 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
3:1	–	–	–
0	R/W	0x0	PGA_R1_SET PGA Resistor 1 Set，PGA 电阻 1 设置。 • 0x0：短路 • 0x1：10 kΩ

9.7.5.7. 0x0028 PGA2_RES_SET

默认值：0x00000000			PGA2_RES_SET 寄存器
位域	类型	默认值	描述
31:27	R/W	0x0	PGA2_RSV
26:24	R/W	0x0	PGA_RFLT_SET PGA Resistor Filter Set，PGA 滤波电阻设置。 • 0x0：短路 • 0x1：50 Ω • 0x2：80 Ω • 0x3：100 Ω • 0x4：130 Ω • 0x5：160 Ω • 0x6：200 Ω • others：短路
23:20	–	–	–
19:16	R/W	0x0	PGA_R4_SET

默认值：0x00000000			PGA2_RES_SET 寄存器
位域	类型	默认值	描述
			PGA Resistor 4 Set，PGA 电阻 4 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
15:13	–	–	–
12	R/W	0x0	PGA_R3_SET PGA Resistor 3 Set，PGA 电阻 3 设置。 • 0x0：短路 • 0x1：10 kΩ
11:8	–	–	–
7:4	R/W	0x0	PGA_R2_SET PGA Resistor 2 Set，PGA 电阻 2 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ

默认值：0x00000000			PGA2_RES_SET 寄存器
位域	类型	默认值	描述
			• 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
3:1	–	–	–
0	R/W	0x0	PGA_R1_SET PGA Resistor 1 Set，PGA 电阻 1 设置。 • 0x0：短路 • 0x1：10 kΩ

9.7.5.8. 0x002C PGA3_RES_SET

默认值：0x00000000			PGA3_RES_SET 寄存器
位域	类型	默认值	描述
31:27	R/W	0x0	PGA3_RSV
26:24	R/W	0x0	PGA_RFLT_SET PGA Resistor Filter Set，PGA 滤波电阻设置。 • 0x0：短路 • 0x1：50 Ω • 0x2：80 Ω • 0x3：100 Ω • 0x4：130 Ω • 0x5：160 Ω • 0x6：200 Ω • others：短路

默认值：0x00000000			PGA3_RES_SET 寄存器
位域	类型	默认值	描述
23:20	–	–	–
19:16	R/W	0x0	PGA_R4_SET PGA Resistor 4 Set，PGA 电阻 4 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ • 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
15:13	–	–	–
12	R/W	0x0	PGA_R3_SET PGA Resistor 3 Set，PGA 电阻 3 设置。 • 0x0：短路 • 0x1：10 kΩ
11:8	–	–	–
7:4	R/W	0x0	PGA_R2_SET PGA Resistor 2 Set，PGA 电阻 2 设置。 • 0x0：短路 • 0x1：10 kΩ • 0x2：20 kΩ • 0x3：30 kΩ

默认值：0x00000000			PGA3_RES_SET 寄存器
位域	类型	默认值	描述
			• 0x4：40 kΩ • 0x5：50 kΩ • 0x6：60 kΩ • 0x7：70 kΩ • 0x8：80 kΩ • 0x9：150 kΩ • 0xA：160 kΩ • 0xB：310 kΩ • 0xC：320 kΩ • 0xD：630 kΩ • 0xE：640 kΩ • others：短路
3:1	–	–	–
0	R/W	0x0	PGA_R1_SET PGA Resistor 1 Set，PGA 电阻 1 设置。 • 0x0：短路 • 0x1：10 kΩ

9.7.5.9. 0x0040 PGA_REF_CTL

默认值：0x00000000			PGA_REF_CTL 寄存器
位域	类型	默认值	描述
31:7	–	–	–
6	R/W	0x0	ENVCM_ATB
5:4	R/W	0x0	PGA_VCM_ATBSEL
3	R/W	0x0	PGA3_REF_SEL PGA3 Reference Select，PGA3 参考电压选择。 • 0x0：选择与 PGA0 一致的参考电压 • 0x1：选择由 PGA3_VREF 管脚提供参考电压

默认值：0x00000000			PGA_REF_CTL 寄存器
位域	类型	默认值	描述
2	R/W	0x0	PGA2_REF_SEL PGA2 Reference Select，PGA2 参考电压选择。 • 0x0：选择与 PGA0 一致的参考电压 • 0x1：选择由 PGA2_VREF 管脚提供参考电压
1	R/W	0x0	PGA1_REF_SEL PGA1 Reference Select，PGA1 参考电压选择。 • 0x0：选择与 PGA0 一致的参考电压 • 0x1：选择由 PGA1_VREF 管脚提供参考电压
0	R/W	0x0	PGA0_REF_SEL PGA0 Reference Select，PGA0 参考电压选择。 • 0x0：选择由内部 VCM 提供参考电压 • 0x1：选择由 PGA0_VREF 管脚提供参考电压

9.7.5.10. 0x00FC PGA_VER

默认值：0x00000100			PGA 模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0100	版本寄存器 V1.0

9.8. Comparator Module (CPM)

Comparator Module (CPM)，可用于比较外部输入电压，也可以与内部 DAC 进行比较。比较的结果可以用来产生中断状态，或者直接控制 EPWM 进入保护状态。M7000 最多内置两个 CPM 模块，分别是 CPM0 和 CPM1。

9.8.1. 特性说明

- 内置 12-bit DAC 参考的窗口比较
- 支持外部电压信号的比较
- 输出支持 IRQ 和触发 EPWM 保护
- 支持同步信号

9.8.2. 原理框图

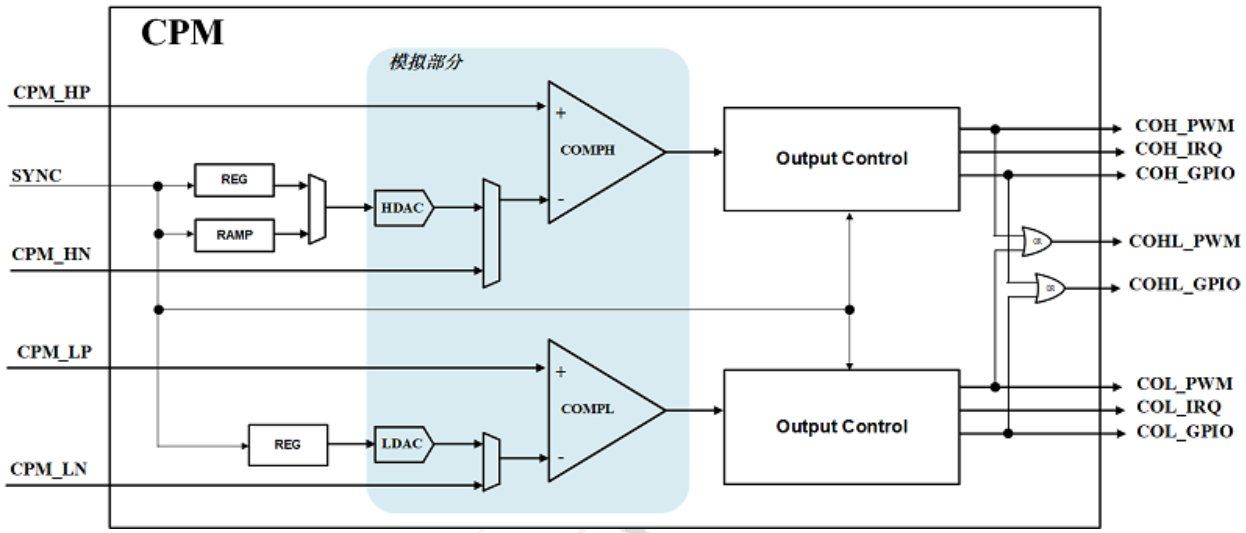


图 9–41 CPM 模块原理框图

图 9–41 : CPM 模块原理框图 中：

- COMPH：高位比较器，负端带有一个 DAC，即高位 HDAC。HDAC 支持寄存器和斜坡模块 RAMP 的控制
- COMPL：低位比较器，负端带有一个 DAC，即低位 LDAC。LDAC 仅支持寄存器控制。
- Output Control：输出控制模块，用于对比较器输出信号的输出处理。

输入输出信号说明如下：

信号	类型	描述
CPM_HP	输入	来自管脚的输入，作为高位比较器的正端输入
CPM_HN	输入	来自管脚的输入，作为高位比较器的负端输入
CPM_LP	输入	来自管脚的输入，作为低位比较器的正端输入
CPM_LN	输入	来自管脚的输入，作为低位比较器的负端输入
SYNC	输入	来自 EPWM 模块输出的信号，用于同步 CPM 模块
COH_PWM	输出	高位比较器的输出，用于作为 EPWM 模块的保护触发信号
COH_IRQ	输出	高位比较器的输出，用于作为 CPM 模块的中断触发信号
COH_GPIO	输出	高位比较器的输出，用于管脚信号的输出
COL_PWM	输出	低位比较器的输出，用于作为 EPWM 模块的保护触发信号
COL_IRQ	输出	低位比较器的输出，用于作为 CPM 模块的中断触发信号

信号	类型	描述
COL_GPIO	输出	低位比较器的输出，用于管脚信号的输出
COHL_PWM	输出	高低位比较器的输出的 OR 逻辑信号，用于作为 EPWM 模块的保护触发信号
COHL_GPIO	输出	高低位比较器的输出的 OR 逻辑信号，用于管脚信号的输出

9.8.3. 功能描述

9.8.3.1. CPM 内置比较器

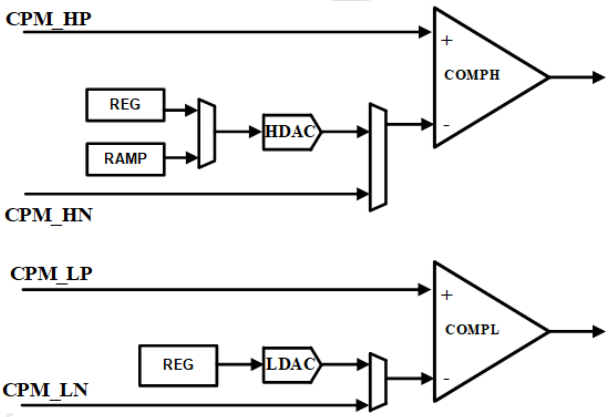


图 9–42 CPM 内置比较器结构

如 图 9–42 : CPM 内置比较器结构所示，CPM 内置两路比较器：

表 9–40 高位比较器与低位比较器特性对比

比较器	正端电压输入	负端电压输入	备注
高位比较器 (`COMPH`)	外部输入信号 `CPM_HP`	可选择外部输入信号 `CPM_HN` 或 `HDAC` 的信号。`HDAC` 的输出信号可以通过软件配置寄存器控制，或由内置的斜波功能模块控制。	当比较器的正端输入电压大于负端输入电压时，比较器输出高电平，否则输出低电平。
低位比较器 (`COMPL`)	外部输入信号 `CPM_LP`	可选择外部输入信号 `CPM_LN` 或 `LDAC` 的信号。`LDAC` 的输出信号只能通过软件配置寄存器控制。	

9.8.3.2. DAC 更新模式

高位和低位的比较器各自对配备了一个内置的 12–bit DAC，分别称为 HDAC 和 LDAC。HDAC 与 LDAC 规格基本相同。

以 HDAC 为例，通过配置输出寄存器 DACH_VALS，HDAC 会输出相应的电压 HDAC_OUT。输出电压的计算公式为：HDAC_OUT = VCC33_IO ×（DACH_VALS ÷ 4096），其中 VCC33_IO 是 IO 输入电压。

• HDAC 的输出电压的更新，支持两种模式：寄存器更新和斜坡模块控制更新，如图 9–43 : HDAC 结构框图所示。在寄存器更新模式中，可以通过时钟信号 SYS_CLK 或同步信号 SYNC 触发读取影子寄存器更新。斜坡模块的更新方式可参考斜坡模块配置。

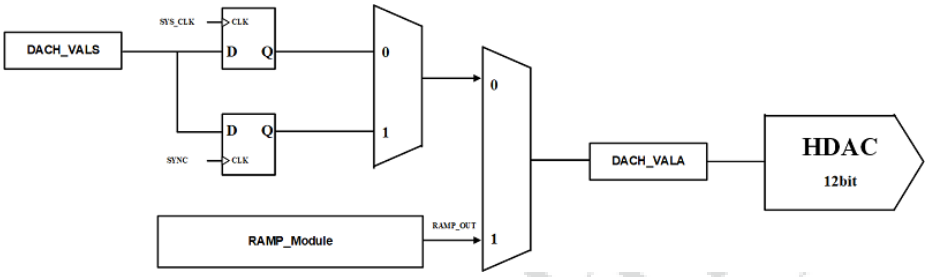


图 9–43 HDAC 结构框图

• LDAC 的输出电压更新仅支持寄存器更新模式，不支持斜坡模块更新，寄存器的更新模式与 HDAC 的一致。

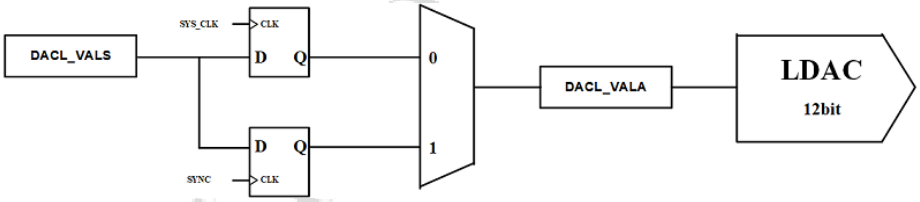


图 9–44 LDAC 结构框图

9.8.3.3. 斜坡模块配置

斜坡模块 (RAMP_Module) 用于生成递减的斜坡信号 RAMP_OUT，控制 HDAC 的输出电压。CPM 可以利用此功能生成与输入电压相关的 PWM 信号。例如，在电源应用中，它可以用来控制反馈电压的 PWM 占空比。

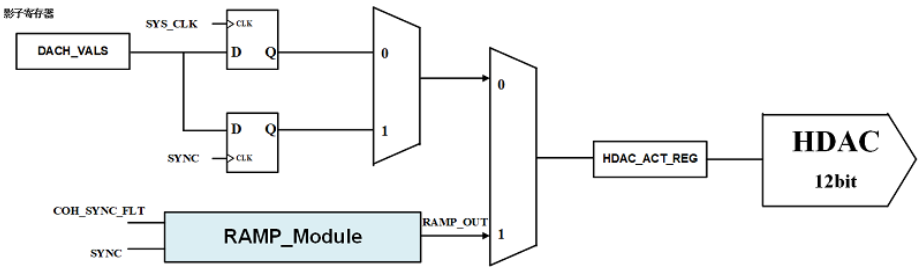


图 9–45 HDAC 斜坡模块

如图 9–45 : HDAC 斜坡模块所示，斜坡模块可以通过以下寄存器配置并结合 SYNC 信号触发，生成精确控制的递减斜坡信号 RAMP_OUT。

表 9–41 斜坡模块配置寄存器描述

寄存器名称	功能描述	更新方式	更新触发信号	备注
RAMP_MAX_REFS	设置斜坡信号的起始点。斜坡信号从该寄存器设置的数值开始递减。	影子更新	SYNC	无单位，具体数值由应用决定

表 9–41 斜坡模块配置寄存器描述 (续)

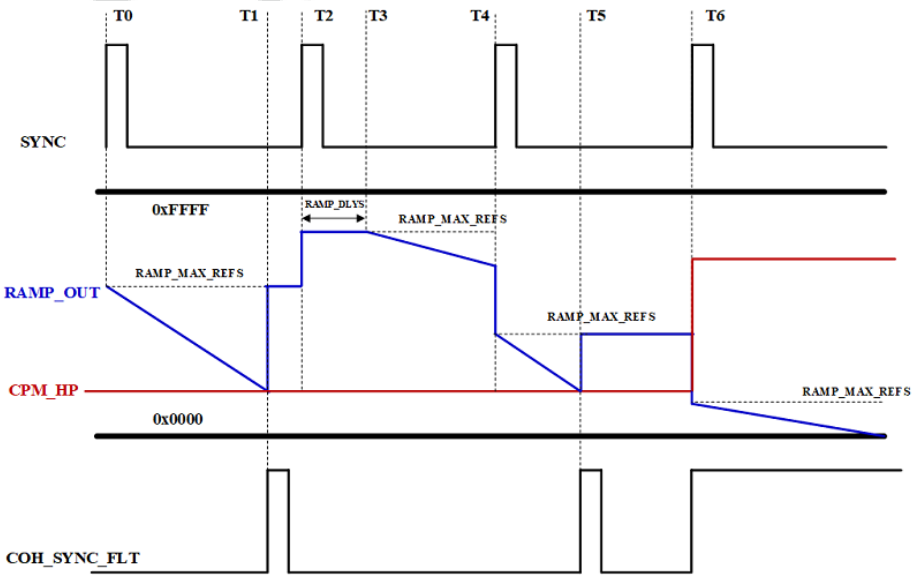
寄存器名称	功能描述	更新方式	更新触发信号	备注
RAMP_DEC_VALS	设置斜坡信号每次递减的幅度值。	影子更新	SYNC	无单位，具体数值由应用决定
RAMP_DLYS	设置斜坡模块启动的延迟时间。	影子更新	SYNC	延迟时间为 SYS_CLK 周期数

注：

- SYS_CLK：斜坡信号每次递减的时钟频率，由 CPM 模块提供，单位 Hz。
- 影子更新：表示寄存器的更新不会立即生效，而是在同步信号 SYNC 到来时，影子寄存器中的值才会被应用到实际寄存器中。
- SYNC：同步信号，用于触发影子寄存器的更新。每次 SYNC 到来时，影子寄存器会更新，使用当前的 RAMP_MAX_REFS、RAMP_DEC_VALS 和 RAMP_DLYS 值来控制斜坡信号的生成。

斜坡信号 RAMP_OUT 典型工作时序

以下是斜坡信号 RAMP_OUT 的典型工作时序示例：



时间点	事件描述
T0	SYNC 信号到来，触发 RAMP_MAX_REFS、RAMP_DEC_VALS 和 RAMP_DLYS 影子寄存器更新，斜坡模块启动（RAMP_DLYS = 0）。 由于 RAMP_DLYS 设置为 0，RAMP_OUT 从当前的 RAMP_MAX_REFS 开始递减。
T1	RAMP_OUT 控制 HDAC 的输出低于比较器阈值 CPM_HP，CPMH 的输出比较信号 COH_SYNC_FLT 发生翻转，斜坡模块停止递减并将输出恢复到当前的 RAMP_MAX_REFS 值。 HDAC 的输出也回到起始值，比较器的输出再次反转。

时间点	事件描述
T2	SYNC 信号再次到来，影子寄存器更新，斜坡模块会延迟启动（RAMP_DLYS > 0）。延迟时间为 (RAMP_DLYS × TSYS_CLK)。
T3	斜坡模块启动，RAMP_OUT 开始递减。 在 T3 ~ T4 时间段，RAMP_OUT 尚未低于 CPM_HP 的电压水平，比较器未发生反转，斜坡信号仍在持续递减。
T4	SYNC 信号到来，影子寄存器更新，斜坡模块继续递减。
T5	比较器信号翻转，斜坡模块停止并恢复到 RAMP_MAX_REFS 起始值。
T6	输入信号 CPM_HP 跳变，导致比较器信号反转。SYNC 信号同时到来，斜坡模块继续按照新一轮寄存器值递减。

注：

斜坡控制模块的递减计数数据位宽为 16 位。数字处理时，最终会截取高 12 位作为 RAMP_OUT 信号输出给 HDAC。

9.8.3.4. 数字输出控制

CPM 模块的数字输出控制逻辑用于处理比较器输出信号，经过多级逻辑操作后，生成可用于 **PWM 保护**、**中断触发**或 **GPIO 输出**的信号。高位比较器和低位比较器的处理逻辑相同。

9.8.3.4.1. 数字输出控制逻辑流程

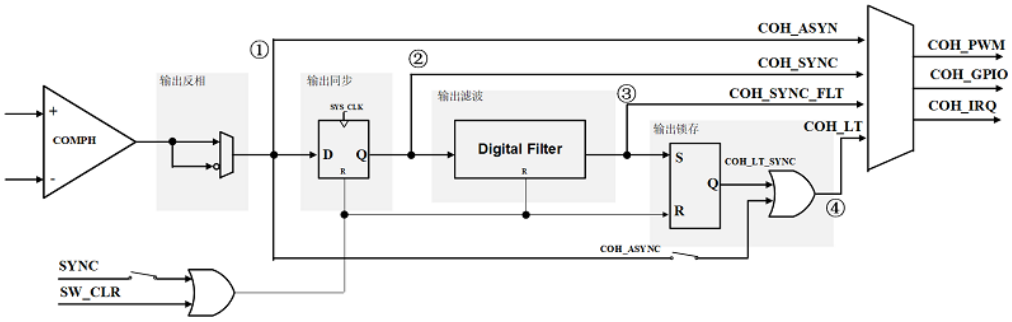


图 9–46 数字输出控制逻辑流程

以高位比较器的输出为例，数字输出控制逻辑分为四级操作，依次如下：

1. 输出异步，此时的信号为异步信号 COH_ASYNC，该通路可以支持输出反相功能。
2. 输出同步，采用 SYS_CLK 对比较器的输出信号进行同步，得到同步信号 COH_SYNC。
3. 输出滤波，采用数字滤波器，对上级信号进行数字滤波，得到信号 COH_SYNC_FLT。
4. 输出锁存，采用 SR 锁存器，锁存输出结果后与 COH_ASYNC 进行 OR 逻辑，得到信号 COH_LT。

9.8.3.4.2. 数字滤波器

数字滤波器通过采样窗口内的数据判断是否翻转输出信号，等效如下的 C 代码实现：

```
//当输出为 0 时，检查采样窗口中 1 的数量

if (FILTER_OUTPUT == 0) {

  //当采样窗口里数据为 1 的数量大于阈值，则输出信号翻转

  if (Num_1s_in_SAMPWIN >= THRESH) {

        FILTER_OUTPUT = 1;
  }
}

//当输出为 1 时，检查采样窗口中 0 的数量

else {

    //当采样窗口里数据为 0 的数量大于阈值，则输出信号翻转

    if (Num_0s_in_SAMPWIN >= THRESH) {

        FILTER_OUTPUT = 0;
    }
}
```

表 9–42 滤波器参数说明

参数名称	描述
`FILTER_OUTPUT`	滤波器的当前输出状态（0 或 1）
`Num_1s_in_SAMPWIN`	采样窗口中值为 1 的数量
`Num_0s_in_SAMPWIN`	采样窗口中值为 0 的数量
`THRESH`	翻转输出的阈值 例如：采样窗口中 1 或 0 的数量超过该值时翻转输出

9.8.3.5. 中断信号

通过寄存器 COH_IRQ_EN/ COL_IRQ_EN 使能，可选择比较器的输出信号 COH_IRQ、COL_IRQ 触发 CPM 中断的产生。通过配置寄存器的模式位，可以选择具体的触发模式：

- 电平模式：支持高/ 低电平触发中断

  当比较器输出信号保持高电平或低电平时，将持续触发中断。

- 边沿模式：支持上升沿/ 下降沿/ 双沿触发中断

  在信号由低到高或由高到低的瞬间触发一次中断。

9.8.3.6. 比较输入选择

CPM 模块支持高低位比较器的输入信号选择。高低位比较器的正相输入端，可以通过寄存器选择使用 GPIO 信号或者 PGA0/1/.../5 的信号。

9.8.4. 寄存器列表

M7000 最多内置 2x CPM 模块，分别为 CPM0/1，对应的寄存器地址映射关系如下。GLB 寄存器部分为对 CPM0/1 的全局控制。

表 9–43 地址映射

模块名	基地址	地址空间
GLB	CPM_BASE	16 B
CPM0	BC0 (CPM_BASE)	256 B
CPM1	BC1 (CPM_BASE + 0x100)	256 B

表 9–44 寄存器列表

偏移地址	缩写	寄存器描述	章节
0x0000	GLB_EN	全局使能寄存器	0x0000 GLB_EN
0x0004	GLB_IRQ_STS	全局中断状态寄存器	0x0004 GLB_IRQ_STS
BC0/1+0x10	COMP_CTL	比较器控制寄存器	BC0/1+0x10 COMP_CTL
BC0/1+0x014	COMP_OUT	比较器输出寄存器	BC0/1+0x014 COMP_OUT
BC0/1+0x18	COMP_HYSCTL	比较器迟滞控制寄存器	BC0/1+0x18 COMP_HYSCTL
BC0/1+0x1C	COMP_STS	比较器状态寄存器	BC0/1+0x1C COMP_STS
BC0/1+0x20	COMP_STS_CLR	比较器状态清除寄存器	BC0/1+0x20 COMP_STS_CLR
BC0/1+0x24	COMP_DAC_CTL	SSINC 比较阈值寄存器	BC0/1+0x24 COMP_DAC_CTL
BC0/1+0x28	DACH_VALS	高位 DAC 数值影子寄存器	BC0/1+0x28 DACH_VALS
BC0/1+0x2C	DACH_VALA	高位 DAC 数值生效寄存器	BC0/1+0x2C DACH_VALA
BC0/1+0x30	RAMP_MAX_REFA	斜坡最大值生效寄存器	BC0/1+0x30 RAMP_MAX_REFA
BC0/1+0x34	RAMP_MAX_REFS	斜坡最大值影子寄存器	BC0/1+0x34 RAMP_MAX_REFS

表 9–44 寄存器列表 (续)

偏移地址	缩写	寄存器描述	章节
BC0/1+0x38	RAMP_DEC_VALA	斜坡递减值生效寄存器	BC0/1+0x38 RAMP_DEC_VALA
BC0/1+0x3C	RAMP_DEC_VALS	斜坡递减值影子寄存器	BC0/1+0x3C RAMP_DEC_VALS
BC0/1+0x40	RAMP_DLYA	斜坡延迟值生效寄存器	BC0/1+0x40 RAMP_DLYA
BC0/1+0x44	RAMP_DLYS	斜坡延迟值影子寄存器	BC0/1+0x44 RAMP_DLYS
BC0/1+0x48	RAMP_STS	斜坡状态值寄存器	BC0/1+0x48 RAMP_STS
BC0/1+0x4C	DACL_VALS	低位 DAC 数值影子寄存器	BC0/1+0x4C DACL_VALS
BC0/1+0x50	DACL_VALA	低位 DAC 数值生效寄存器	BC0/1+0x50 DACL_VALA
BC0/1+0x54	LFIL_CTL	低位滤波器控制寄存器	BC0/1+0x54 LFIL_CTL
BC0/1+0x58	LFIL_CLK_CTL	低位滤波器时钟控制寄存器	BC0/1+0x58 LFIL_CLK_CTL
BC0/1+0x5C	HFIL_CTL	高位滤波器控制寄存器	BC0/1+0x5C HFIL_CTL
BC0/1+0x60	HFIL_CLK_CTL	高位滤波器时钟控制寄存器	BC0/1+0x60 HFIL_CLK_CTL
BC0/1+0x70	IN_SRC_SEL	输入源选择寄存器	BC0/1+0x70 IN_SRC_SEL
0x0FFC	CPM_VER	CPM 模块版本寄存器	0x0FFC CPM_VER

9.8.5. 寄存器描述

9.8.5.1. 0x0000 GLB_EN

默认值：0x00000000			全局使能寄存器
位域	类型	默认值	描述
31:2	–	–	–
1	R/W	0x0	CPM1_EN CPM1 Enable，CPM1 使能。 • 0x0：未使能 • 0x1：使能
0	R/W	0x0	CPM0_EN CPM0 Enable，CPM0 使能。 • 0x0：未使能 • 0x1：使能

9.8.5.2. 0x0004 GLB_IRQ_STS

默认值：0x00000000			全局中断状态寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W1C	0x0	CPM1_COL_IRQ CPM1 Low Comparator Output IRQ，低位比较器中断。 • 0x0：无中断状态产生 • 0x1：出现中断状态
2	R/W1C	0x0	CPM1_COH_IRQ CPM1 High Comparator Output IRQ，高位比较器中断。 • 0x0：无中断状态产生 • 0x1：出现中断状态
1	R/W1C	0x0	CPM0_COL_IRQ CPM0 Low Comparator Output IRQ，低位比较器中断。 • 0x0：无中断状态产生 • 0x1：出现中断状态
0	R/W1C	0x0	CPM0_COH_IRQ CPM0 High Comparator Output IRQ，高位比较器中断。 • 0x0：无中断状态产生 • 0x1：出现中断状态

9.8.5.3. BC0/1+0x10 COMP_CTL

默认值：0x00000000			比较器控制寄存器
位域	类型	默认值	描述
31:29	–	–	–
28:27	R/W	0x0	COL_IRQ_SEL Low Comparator COL_IRQ source select，低位比较器输出至中断的信号源选择。 • 0x0：同步信号输出 COL_SYNC • 0x1：同步信号输出 COL_SYNC

默认值：0x00000000			比较器控制寄存器
位域	类型	默认值	描述
			• 0x2：同步后进行数字滤波的信号输出 COL_SYNC_FLT • 0x3：锁存信号输出 COL_LT
26	R/W	0x0	COMPL_EN Low Comparator Enable，低位比较器使能。 • 0x0：未使能 • 0x1：使能
25:24	–	–	–
23	R/W	0x0	LDAC_EN Low Comparator DAC enable，LDAC 使能。 • 0x0：未使能 • 0x1：使能
22	R/W	0x0	COL_LT_ASYNC_EN Low Comparator COL_LT select asynchronous source，低位输出比较器的锁存信号 COL_LT 采用异步信号源使能。 • 0x0：未使能，COL_LT 采用同步信号源 • 0x1：使能，COL_LT 采用异步信号源
21:20	R/W	0x0	COL_GPIO_SEL Low Comparator COL_GPIO source select，低位比较器输出至 GPIO 的信号源选择。 • 0x0：异步信号输出 COL_ASYN • 0x1：同步信号输出 COL_SYNC • 0x2：同步后进行数字滤波的信号输出 COL_SYNC_FLT • 0x3：锁存信号输出 COL_LT
19:18	R/W	0x0	COL_PWM_SEL Low Comparator COL_PWM source select，低位比较器输出至 EPWM 的信号源选择。 • 0x0：异步信号输出 COL_ASYN • 0x1：同步信号输出 COL_SYNC • 0x2：同步后进行数字滤波的信号输出 COL_SYNC_FLT • 0x3：锁存信号输出 COL_LT

默认值：0x00000000			比较器控制寄存器
位域	类型	默认值	描述
17	R/W	0x0	COMPL_INV Low Comparator output invert enable，低位比较器输出反相使能。 • 0x0：未使能 • 0x1：使能，比较器输出信号反相
16	R/W	0x0	COMPL:_NSRC Low Comparator Negative Input Source，低位比较器的负端输入源选择。 • 0x0：选择内部 DAC 信号 • 0x1：选择外部管脚信号 COMPL_N
15:13	–	–	–
12:11	R/W	0x0	COH_IRQ_SEL High comparator COH_IRQ source select，高位比较器输出至中断的信号源选择。 • 0x0：同步信号输出 COH_SYNC • 0x1：同步信号输出 COH_SYNC • 0x2：同步后进行数字滤波的信号输出 COH_SYNC_FLT • 0x3：锁存信号输出 COH_LT
10	R/W	0x0	COMPH_EN High Comparator Enable，高位比较器使能。 • 0x0：未使能 • 0x1：使能
9:8	–	–	–
7	R/W	0x0	HDAC_EN High Comparator DAC enable，HDAC 使能。 • 0x0：未使能 • 0x1：使能
6	R/W	0x0	COH_LT_ASYNC_EN High comparator COH_LT select asynchronous source，高位输出比较器的锁存信号 COH_LT 采用异步信号源使能。 • 0x0：未使能，COH_LT 采用同步信号源 • 0x1：使能，COH_LT 采用异步信号源

默认值：0x00000000			比较器控制寄存器
位域	类型	默认值	描述
5:4	R/W	0x0	COH_GPIO_SEL High comparator COH_GPIO source select，高位比较器输出至 GPIO 的信号源选择。 • 0x0：异步信号输出 COH_ASYN • 0x1：同步信号输出 COH_SYNC • 0x2：同步后进行数字滤波的信号输出 COH_SYNC_FLT • 0x3：锁存信号输出 COH_LT
3:2	R/W	0x0	COH_PWM_SEL High Comparator COH_PWM source select，高位比较器输出至 EPWM 的信号源选择。 • 0x0：异步信号输出 COH_ASYN • 0x1：同步信号输出 COH_SYNC • 0x2：同步后进行数字滤波的信号输出 COH_SYNC_FLT • 0x3：锁存信号输出 COH_LT
1	R/W	0x0	COMPH_INV High Comparator output invert enable，高位比较器输出反相使能。 • 0x0：未使能 • 0x1：使能，比较器输出信号反相
0	R/W	0x0	COMPH_NSRC High Comparator Negative Input Source，高位比较器的负端输入源选择。 • 0x0：选择内部 DAC 信号 • 0x1：选择外部管脚信号 COMPH_N

9.8.5.4. BC0/1+0x014 COMP_OUT

默认值：0x00000000			比较器输出寄存器
位域	类型	默认值	描述
31:30	–	–	–
29	R/W	0x0	COHL_GPIO_EN COHL GPIO output Enable，高低位比较器输出进行 OR 逻辑作为 GPIO 输出使能。

默认值：0x00000000			比较器输出寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
28	R/W	0x0	COHL_PWM_EN COHL PWM Enable，高低位比较器输出进行 OR 逻辑作为 EPWM 保护使能。 • 0x0：未使能 • 0x1：使能
27:23	–	–	–
22	R/W	0x0	COL_PWM_EN Low Comparator PWM Enable，低位比较器输出 EPWM 保护使能。 • 0x0：未使能 • 0x1：使能
21	R/W	0x0	COL_GPIO_EN Low Comparator GPIO output Enable，GPIO 输出使能。 • 0x0：未使能 • 0x1：使能
20	R/W	0x0	COL_IRQ_LVL Low Comparator Interrupt Level Mode Select，低位比较器中断电平模式选择。 • 0x0：高电平 • 0x1：低电平
19:18	R/W	0x0	COL_IRQ_EDG Low Comparator Interrupt Edge Mode Select，低位比较器中断边沿模式选择。 • 0x0：上升沿 • 0x1：下降沿 • 0x2/0x3：双沿
17	R/W	0x0	COL_IRQ_MOD Low Comparator Interrupt Mode Select，低位比较器中断模式选择。 • 0x0：电平模式 • 0x1：边沿模式
16	R/W	0x0	COL_IRQ_EN Low Comparator Interrupt Enable，低位比较器中断使能。

默认值：0x00000000			比较器输出寄存器
位域	类型	默认值	描述
			• 0x0：未使能 • 0x1：使能
15:7	–	–	–
6	R/W	0x0	COH_PWM_EN High Comparator PWM Enable，高位比较器输出 EPWM 保护使能。 • 0x0：未使能 • 0x1：使能
5	R/W	0x0	COH_GPIO_EN High Comparator GPIO output Enable，GPIO 输出使能。 • 0x0：未使能 • 0x1：使能
4	R/W	0x0	COH_IRQ_LVL High Comparator Interrupt Level Mode Select，高位比较器中断电平模式选择。 • 0x0：高电平 • 0x1：低电平
3:2	R/W	0x0	COH_IRQ_EDG High Comparator Interrupt Edge Mode Select，高位比较器中断边沿模式选择。 • 0x0：上升沿 • 0x1：下降沿 • 0x2/0x3：双沿
1	R/W	0x0	COH_IRQ_MOD High Comparator Interrupt Mode Select，高位比较器中断模式选择。 • 0x0：电平模式 • 0x1：边沿模式
0	R/W	0x0	COH_IRQ_EN High Comparator Interrupt Enable，高位比较器中断使能。 • 0x0：未使能 • 0x1：使能

9.8.5.5. BC0/1+0x18 COMP_HYSCTL

默认值：0x00000000			比较器迟滞控制寄存器
位域	类型	默认值	描述
31:9	–	–	–
8	R/W	0x00	COMP_IBIAS_EN Comparator Ibias Enable，比较器偏置电流使能。 • 0x0：未使能 • 0x1：使能
7	–	–	–
6:4	R/W	0x0	COMPL_HYS_EN Low Comparator Hysteresis Enable，低位比较器迟滞电压使能。 • 0x0：无迟滞 • 0x1：1x 迟滞电压，迟滞电压典型为 1x 12 LSB • 0x2：2x 迟滞电压，迟滞电压典型为 2x 12 LSB • 0x3：3x 迟滞电压，迟滞电压典型为 3x 12 LSB • ... • 0x7：7x 迟滞电压，迟滞电压典型为 7x 12 LSB LSB 为 DAC 的 LSB，在 3.3V 供电的情况下，LSB = 3.3V/4096。
3	–	–	–
2:0	R/W	0x0	COMPH_HYS_EN High Comparator Hysteresis Enable，高位比较器迟滞电压设置。 • 0x0：不使能，比较器无迟滞 • 0x7：使能，比较器迟滞电压使能，迟滞为 +38mV • others：无效

9.8.5.6. BC0/1+0x1C COMP_STS

默认值：0x00000000			比较器状态寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R	0x0	COL_LT_SYNC

默认值：0x00000000			比较器状态寄存器
位域	类型	默认值	描述
			Latched Value of ，Low Comparator Digital Filter Output，低位比较器经过锁存的信号。 • 0x0：低电平 • 0x1：高电平
2	R	0x0	COL_SYNC_FLT Low Comparator Digital Filter Output，低位比较器经过数字滤波后的输出信号。 • 0x0：低电平 • 0x1：高电平
1	R	0x0	COH_LT_SYNC Latched Value of ，High Comparator Digital Filter Output，高位比较器经过锁存的信号。 • 0x0：低电平 • 0x1：高电平
0	R	0x0	COH_SYNC_FLT High Comparator Digital Filter Output，高位比较器经过数字滤波后的输出信号。 • 0x0：低电平 • 0x1：高电平

9.8.5.7. BC0/1+0x20 COMP_STS_CLR

默认值：0x00000000			比较器状态清除寄存器
位域	类型	默认值	描述
31:4	–	–	–
3	R/W	0x0	COL_LT_SYNCLR_EN Low Comparator SYNCLR Enable，使能允许 SYNCLR 进行低位比较器的锁存信号清 0。 • 0x0：未使能 • 0x1：使能
2	R/W1AC	0x0	COL_LT_CLR Low Comparator Latch Clear，低位比较器的输出锁存信号进行清 0。

默认值：0x00000000			比较器状态清除寄存器
位域	类型	默认值	描述
			• 0x0：无操作 • 0x1：对锁存信号清 0，完成后此位自动清 0
1	R/W	0x0	COH_LT_SYNCLR_EN High comparator SYNCLR Enable，使能允许 SYNCLR 进行高位比较器的锁存信号清 0。 • 0x0：未使能 • 0x1：使能
0	R/W1AC	0x0	COH_LT_CLR High Comparator Latch Clear，高位比较器的输出锁存信号进行清 0。 • 0x0：无操作 • 0x1：对锁存信号清 0，完成后此位自动清 0

9.8.5.8. BC0/1+0x24 COMP_DAC_CTL

默认值：0x00000000			SSINC 比较阈值寄存器
位域	类型	默认值	描述
31:8	–	–	–
7:4	R/W	0x0	SYNC_SRC Synchronous Source Select，同步信号源选择，此同步信号用于控制 HDAC 和 LDAC 的影子寄存器的更新，HDAC 的斜波信号模块的启动，以及输出锁存状态清 0 操作。 • 0x0：EPWM0_CPM_SYN • 0x1：EPWM1_CPM_SYN • 0x2：EPWM2_CPM_SYN • … • 0xB：EPWM11_CPM_SYN • others：Reserved
3	–	–	–
2	R/W	0x0	DAC_LOAD_SEL DAC Load Select，DAC 数值的影子寄存器更新至立即寄存器的触发信号选择。

默认值：0x00000000			SSINC 比较阈值寄存器
位域	类型	默认值	描述
			• 0x0：采用 SYS_CLK • 0x1：采用 SYNC
1	R/W	0x0	RAMP_LOAD_SEL Ramp Load Select，选择 RAMPSTS 的更新数据源。 • 0x0：更新源选择 RAMP_MAX_REFA • 0x1：更新源选择 RAMP_MAX_REFS
0	R/W	0x0	HDAC_SRC HDAC source select，高位 DAC 输出控制源选择。 • 0x0：HDAC_VALA 通过影子寄存器 HDAC_VALS 更新 • 0x1：HDAC_VALA 通过斜坡信号模块更新 注： 此位从 0x0 切换到 0x1，自动触发一次斜坡模块的影子寄存器更新到立即寄存器，即 RAMP_DLYS 更新到 RAMP_DLYA，RAMP_DEC_VALS 更新到 RAMP_DEC_VALA，RAMP_MAX_REFS 更新到 RAMP_MAX_REFA。

9.8.5.9. BC0/1+0x28 DACH_VALS

默认值：0x00000000			高位 DAC 数值影子寄存器
位域	类型	默认值	描述
31:12	–	–	–
11:0	R/W	0x0	DACH_VALS High DAC Shadow Value，HDAC 的影子寄存器。

9.8.5.10. BC0/1+0x2C DACH_VALA

默认值：0x00000000			高位 DAC 数值生效寄存器
位域	类型	默认值	描述
31:12	–	–	–
11:0	R	0x0	DACH_VALA

默认值：0x00000000			高位 DAC 数值生效寄存器
位域	类型	默认值	描述
			High DAC Active Value，HDAC 的当前生效寄存器。

9.8.5.11. BC0/1+0x30 RAMP_MAX_REFA

默认值：0x00000000			斜坡最大值生效寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	RAMP_MAX_REFA Ramp Maximum Reference Active Value，RAMP_MAX_REF 的当前生效值。

9.8.5.12. BC0/1+0x34 RAMP_MAX_REFS

默认值：0x00000000			斜坡最大值影子寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	RAMP_MAX_REFS Ramp Maximum Reference Shadow Value，RAMP_MAX_REF 的影子值。

9.8.5.13. BC0/1+0x38 RAMP_DEC_VALA

默认值：0x00000000			斜坡递减值生效寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	RAMP_DEC_VALA Ramp Decrement Active Value，RAMP_DEC_VAL 的当前生效值。

9.8.5.14. BC0/1+0x3C RAMP_DEC_VALS

默认值：0x00000000			斜坡递减值影子寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	RAMP_DEC_VALS Ramp Decrement Value Active，RAMP_DEC_VAL 的影子值。

9.8.5.15. BC0/1+0x40 RAMP_DLYA

默认值：0x00000000			斜坡延迟值生效寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	RAMP_DLYA Ramp Delay Active Value，RAMP_DLY_VAL 的当前生效值。

9.8.5.16. BC0/1+0x44 RAMP_DLYS

默认值：0x00000000			斜坡延迟值影子寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R/W	0x0	RAMP_DLYS Ramp Delay Shadow Value，RAMP_DLY_VAL 的影子值。

9.8.5.17. BC0/1+0x48 RAMP_STS

默认值：0x00000000			斜坡状态值寄存器
位域	类型	默认值	描述
31:16	–	–	–
15:0	R	0x0	RAMP_STS Ramp Status，RAMP 状态值。

9.8.5.18. BC0/1+0x4C DACL_VALS

默认值：0x00000000			低位 DAC 数值影子寄存器
位域	类型	默认值	描述
31:12	–	–	–
11:0	R/W	0x0	DACL_VALS Low DAC Shadow Value，LDAC 的影子寄存器。

9.8.5.19. BC0/1+0x50 DACL_VALA

默认值：0x00000000			低位 DAC 数值生效寄存器
位域	类型	默认值	描述
31:12	–	–	–
11:0	R	0x0	DACL_VALA Low DAC Active Value，LDAC 的当前生效寄存器。

9.8.5.20. BC0/1+0x54 LFIL_CTL

默认值：0x00000000			低位滤波器控制寄存器
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	FIL_EN Low Filter Enable，低位滤波器使能。 • 0x0：未使能 • 0x1：使能
10:6	R/W	0x0	THRESH Low Filter Majority Voting Threshold，低位滤波器的阈值，生效值为 THRESH + 1。
5:0	R/W	0x0	SAMP_WIN Low Filter Sample Window Size，低位滤波器的采样窗口大小，生效值为 SAMP_WIN + 1。

9.8.5.21. BC0/1+0x58 LFIL_CLK_CTL

默认值：0x00000000			低位滤波器时钟控制寄存器
位域	类型	默认值	描述
31:10	–	–	–
9:0	R/W	0x0	CLK_PRE_SCALE Low filter sample clock prescale，采样时钟的分频比。

9.8.5.22. BC0/1+0x5C HFIL_CTL

默认值：0x00000000			高位滤波器控制寄存器
位域	类型	默认值	描述
31:12	–	–	–
11	R/W	0x0	FIL_EN High Filter Enable，高位滤波器使能。 • 0x0：未使能 • 0x1：使能
10:6	R/W	0x0	THRESH High Filter Majority Voting Threshold，高位滤波器的阈值，生效值为 THRESH + 1。
5:0	R/W	0x0	SAMP_WIN High Filter Sample Window Size，高位滤波器的采样窗口大小，生效值为 SAMP_WIN + 1。

9.8.5.23. BC0/1+0x60 HFIL_CLK_CTL

默认值：0x00000000			高位滤波器时钟控制寄存器
位域	类型	默认值	描述
31:10	–	–	–
9:0	R/W	0x0	CLK_PRE_SCALE High Filter Sample Clock Prescale，采样时钟的分频比。

9.8.5.24. BC0/1+0x70 IN_SRC_SEL

默认值：0x00000000			输入源选择寄存器
位域	类型	默认值	描述
31:12	–	–	–
11:9	R/W	0x0	COMP_ATB_SEL COMP_ATB Select，测试信号选择。 • 0x0：comph 的 vbn 信号 • 0x1：comph 的 vbp 信号 • 0x2：comph 的 vip 信号 • 0x3：comph 的 vin 信号 • 0x4：compl 的 vbn 信号 • 0x5：compl 的 vbp 信号 • 0x6：compl 的 vip 信号 • 0x7：compl 的 vin 信号
8	R/W	0x0	COMP_ATB_EN COMP_ATB Enable，测试信号选择使能。 • 0x0：未使能 • 0x1：使能
7:5	R/W	0x0	COMPL_P_SEL COMPL_P Source Select，COMPL_P 输入源选择。 • 0x0：PGA0_OUT • 0x1：PGA1_OUT • 0x2：PGA2_OUT • 0x3：PGA3_OUT • 0x4：GPIO • others：Reserved
4	R/W	0x0	COMPL_P_EN COMPL_P Source Enable，COMPL_P 输入信号使能。 • 0x0：未使能 • 0x1：使能
3:1	R/W	0x0	COMPH_P_SEL COMPH_P Source Select，COMPH_P 输入源选择。

默认值：0x00000000			输入源选择寄存器
位域	类型	默认值	描述
			• 0x0：PGA0_OUT • 0x1：PGA1_OUT • 0x2：PGA2_OUT • 0x3：PGA3_OUT • 0x4：GPIO • others：Reserved
0	R/W	0x0	COMPH_P_EN COMPH_P Source Enable，COMPH_P 输入信号使能。 • 0x0：未使能 • 0x1：使能

9.8.5.25. 0x0FFC CPM_VER

默认值：0x00000100			CPM 模块版本寄存器
位域	类型	默认值	描述
31:0	R	0x0000_0100	版本寄存器 V1.0